Министерство образования и науки Украины
Запорожский национальный технический университет
Открытое акционерное общество «Мотор Сич»

В.И. Дубровин
С.А. Субботин
А.В. Богуслаев
В.К. Яценко

# ИНТЕЛЛЕКТУАЛЬНЫЕ СРЕДСТВА ДИАГНОСТИКИ И ПРОГНОЗИРОВАНИЯ НАДЕЖНОСТИ АВИАДВИГАТЕЛЕЙ

Запорожье
2003

ББК 32.97
Д79
УДК 61.2:004.93:007.52

**Дубровин В.И., Субботин С.А., Богуслаев А.В., Яценко В.К.**

Д79 Интеллектуальные средства диагностики и прогнозирования надежности авиадвигателей: Монография.-Запорожье: ОАО «Мотор-Сич», 2003.- 279 с.

ISBN 966-7108-59-7

Ил. 76 Табл. 22 Библиогр.: 144 назв.

Рассмотрены экспериментально-статистические методы диагностики и прогнозирования надежности авиадвигателей в процессе производства и в эксплуатации, содержащие элементы искусственного интеллекта. Значительное внимание уделено решению практических задач.

Предназначено для научных и инженерно-технических работников, аспирантов, а также может использоваться студентами различных специальностей для приобретения практических навыков решения задач управления качеством.

Рекомендовано к печати Научно-техническим советом ОАО "Мотор Сич"

Рецензенты:

Долматов Анатолий Иванович, доктор технических наук, профессор, заведующий кафедрой Национального аэрокосмического университета им. Н.Е. Жуковского "Харьковский авиационный институт"

Кривов Георгий Алексеевич, доктор технических наук, профессор, директор ОАО "Украинский научно-исследовательский институт авиационной технологии"

Valeriy Dubrovin, Sergey Subbotin, Alexander Boguslayev and Viktor Yatzenko
Intelligent Means of Diagnostics and Prediction of Reliability of Airengines: Monograph.-Zaporozhye, Joint-Stock Company “Motor Sich”, 2003.-279 p. (ISBN 966-7108-59-7)

# СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

Обеспечение высокого уровня качества, надежности и долговечности производимых и эксплуатируемых изделий обуславливает необходимость разработки и внедрения в производстве и в эксплуатации автоматизированных систем управления качеством, одной из важнейших составных частей которых являются системы диагностики и прогнозирования надежности изделий.

Разработка автоматизированных систем диагностики и прогнозирования в свою очередь предполагает создание математического, алгоритмического, программного и информационного обеспечения для сбора, хранения, обработки и анализа диагностической информации.

Для решения задач, связанных с анализом данных при наличии случайных и непредсказуемых воздействий, необходимо использовать арсенал методов математической статистики и теории принятия решения. Эти методы позволяют выявлять закономерности на фоне случайностей, делать обоснованные выводы и прогнозы.

Использование методов многомерной статистики предполагает обращение к системному анализу рассматриваемого явления, основных его составляющих и их связей, принятие решения о характере установленных закономерностей. Кроме того, программно-алгоритмическое беспечение такого анализа имеет отношение к методам искусственного интеллекта.

Настоящая книга ставит целью дать обзор современных технологий решения на компьютере задач обработки и интерпретации экспериментальных данных, а также рассмотреть методологию разработки программного обеспечения систем поддержки принятия решений для автоматизации диагностики и прогнозирования надежности сложных технических объектов и процессов.

Особое внимание в книге уделяестя рассмотрению нейросетевых методов моделирования многомерных нелинейных зависимостей как весьма перспективному направлению искусственного интеллекта.

В книге детально описаны и проанализированы примеры применения методов обработки и анализа данных при решении практических задач диагностики и прогнозирования надежности авиадвигателей.

Авторы выражают искреннюю благодарность своим коллегам, принимавшим участие в отдельных исследованиях, результаты которых приведены в данной книге, рецензентам, а также тем, кто оказывал дружескую поддержку авторам при написании книги.

# ГЛАВА 1. ОСНОВНЫЕ ЗАДАЧИ И ПРИНЦИПЫ ИНТЕЛЛЕКТУАЛЬНОЙ ДИАГНОСТИКИ

## 1.1 Определение и цель диагностики

**Техническая диагностика** — наука о распознавании состояния технической системы, включающая широкий круг проблем, связанных с получением и оценкой диагностической информации.

Термин «**диагностика**» происходит от греческого слова «дйбгнщуфйкб», что означает распознавание, определение. В процессе диагностики устанавливается диагноз, т. е. определяется состояние состояние системы.

Техническая диагностика изучает методы получения и оценки диагностической информации, диагностические модели и алгоритмы принятия решений.

**Целью технической диагностики** является повышение надежности и ресурса технических систем.

Как известно, наиболее важным показателем надежности является отсутствие отказов во время функционирования (работы) технической системы. Техническая диагностика благодаря раннему обнаружению дефектов и неисправностей позволяет устранить отказы в процессе технического обслуживания, что повышает надежность и эффективность эксплуатации, а также дает возможность эксплуатации технических систем ответственного назначения по состоянию.

В практике ресурс таких систем определяется по наиболее «слабым» экземплярам изделий. При эксплуатации по состоянию каждый экземпляр эксплуатируется до предельного состояния в соответствии с рекомендациями системы технической диагностики.

## 1.2 Структура технической диагностики. Интеллектуальная диагностика

На рис. 1.1 показана структура технической диагностики. Она характеризуется двумя взаимопроникающими и взаимосвязанными направлениями: теорией контролеспособности и теорией распознавания.

**Теория контролеспособности** включает разработку средств и методов получения диагностической информации, автоматизированный контроль и поиск неисправностей. Техническую диагностику следует рассматривать как раздел **общей теории надежности**.

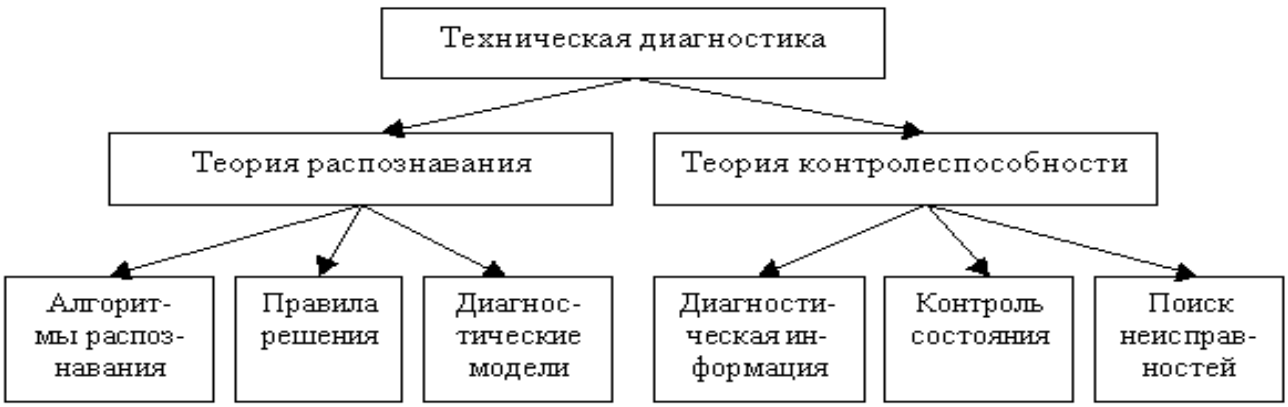



Рис. 1.1 - Структура технической диагностики

Использование физических средств неразрущающего контроля и методов первичной статистической обработки данных оказывается недостаточным при построении надежных автоматизированных систем принятия решений в задачах технической диагностики, поскольку эти средства не позволяют строить высокоточные многомерные диагностические модели нелинейных процессов, а также извлекать знания из экспериментальных данных и адаптироваться к

изменениям во внешней и внутренней среде диагностируемого объекта.

**Интеллектуальная диагностика** представляет собой совокупность средств, позволяющих строить надежные и адекватные модели диагностируемых сложных технических объектов и процессов по экспериментальным данным, обладающие при этом низкой избыточностью, высокой эффективностью и способностью адаптироваться к изменениям во внешней и внутренней средах диагностируемого объекта (процесса), что достигается обучением (переобучением).

Инструментальным базисом для осуществления интеллектуальной диагностики является теория распознавания образов и методы нейроинформатики.

**Теория распознавания** содержит разделы, связанные с построением алгоритмов распознавания, решающих правил и диагностических моделей.

### 1.3. Задачи интеллектуальной диагностики

Интеллектуальная диагностика решает обширный круг задач, многие из которых являются смежными с задачами других научных дисциплин.

**Основной задачей интеллектуальной диагностики** является распознавание состояния технической системы в условиях ограниченной информации. Техническую диагностику иногда называют **безразборной диагностикой**, т.е. диагностикой, осуществляемой без разборки изделия. Анализ состояния проводится в условиях эксплуатации, при которых получение информации крайне затруднено. Часто не представляется возможным по имеющейся информации сделать однозначное заключение и приходится использовать статистические методы.

Теоретическим фундаментом для решения основной задачи интеллектуальной диагностики следует считать общую **теорию распознавания образов**. Эта теория, составляющая важный раздел технической кибернетики, занимается распознаванием образов любой природы (геометрических, звуковых и т.п.), машинным распознаванием речи, печатного и рукописного текстов и т.д. Интеллектуальная диагностика изучает алгоритмы распознавания применительно к задачам диагностики, которые обычно могут рассматриваться как задачи классификации.

Алгоритмы распознавания в технической диагностике частично основываются на диагностических моделях, устанавливающих связь между **состояниями технической системы** и их отображениями в пространстве **диагностических сигналов**. Важной частью проблемы распознавания являются **правила принятия решений** (**решающие правила**). Решение диагностической задачи (отнесение изделия к исправным или неисправным) всегда связано с риском ложной тревоги или пропуска цели. Для принятия обоснованного решения целесообразно привлекать методы теории статистических решений, разработанные впервые в радиолокации.

**Решение задач** диагностики всегда связано с **прогнозированием надежности** на ближайший период эксплуатации (до следующего технического осмотра). Здесь решения должны основываться на **моделях отказов**, изучаемых в **теории надежности**.

Вторым важным направлением диагностики является **теория контролеспособности**.

**Контролеспособностью** называется свойство изделия обеспечивать достоверную оценку его технического состояния и раннее обнаружение неисправностей и отказов. Под **ранним обнаружением отказов и неисправностей** понимается выявление дефекта или неисправности в начальной стадии, при которой еще не проявляются отрицательные последствия для надежности или работоспособности изделия. Контролеспособность создается конструкцией изделия и принятой системой технической диагностики.

Важное значение контролеспособность имеет для радиоэлектронных систем, для которых теория автоматического контроля и поиска неисправностей составляет самостоятельный раздел технической диагностики.

Контролеспособность в первую очередь зависит от качества и объема диагностической информации, которая может быть получена при эксплуатации изделия и его техническом обслуживании, а также при специальных диагностических испытаниях (диагностических тестах). Поэтому, крупной **задачей теории контролеспособности** является изучение средств и методов получения

диагностической информации. В сложных технических системах используется автоматизированный контроль состояния, которым предусматривается обработка диагностической информации и формирование управляющих сигналов. Методы проектирования автоматизированных систем контроля составляют одно из направлений теории контролеспособности. Наконец, очень важные задачи теории контролеспособности связаны с разработкой алгоритмов поиска неисправностей, разработкой диагностических тестов, минимизацией процесса установления диагноза.

### 1.4. Прикладные вопросы интеллектуальной диагностики

В механических и радиоэлектронных системах основное назначение диагностики - повышение надежности и ресурса изделий с помощью раннего обнаружения дефектов и оптимизации процессов технического обслуживания. Интеллектуальная диагностика сложных систем представляет собой систему, которая должна иметь информационное, техническое и математическое обеспечение.

**Информационное обеспечение** включает способы получения диагностической информации, ее хранение и систематизацию. Информационное обеспечение содержит необходимый массив восполняемых технических сведений (обучающие последовательности)

**Техническое обеспечение** представляет собой совокупность устройств получения и обработки информации (диагностические приборы, датчики, сигнализаторы и т. п.). Важную часть технического обеспечения современных систем диагностики составляют ЭВМ, устройства типа «аналог—код» и др.

**Математическое обеспечение** включает математические методы, алгоритмы и программы распознавания и численного моделирования.

Техническая диагностика как система включает также и **коллектив специалистов**, ответственных за принятие решении.

## 1.5 Процесс и этапы интеллектуальной диагностики

**Процесс интеллектуальной диагностики** в общем случае состоит из ряда **этапов**, выполнение которых позволяет получить математическую модель объекта или процесса, необходимую для осуществления диагностических мероприятий.

Укрупненная схема процесса интеллектуальной диагностики представлена на рис. 1.2. На схеме изображены только те этапы, автоматизация которых является возможной на данном этапе развития науки и техники. Рассмотрим процесс диагностики по этапам.

**Этап построения априорной концептуальной модели объекта (процесса)** необходим для получения начальной (предварительной) информации о любом объекте (процессе). Этот этап поддерживается **теорией подобия**, указывающей целесообразную структуру параметров модели, и **теорией аналогий**, изучающей связь моделей физических систем различных видов. Отдельные вопросы концептуального моделирования встречаются в теориях геометрических, динамических, статистических измерений, идентификации, планирования эксперимента. Однако в целом теория такого моделирования не построена.

Заслуживают внимания, в частности: возможность описания одних и тех же объектов моделями разного вида; связи между моделями, описывающими различные стороны объекта; использование человеком иерархий моделей, различающихся степенью наглядности (и вообще понятие наглядной модели). Интерес, по-видимому, представляет также дуализм между структурой модели и ее параметрами. Недостаточно изученным представляется вопрос о классах моделей, присущих тем или иным видам познавательных процедур. Наконец, мало внимания уделяется общему подходу к неопределенности моделей.

Трудность формализации данного этапа и его неопределенность выводит рассмотрение этого этапа за пределы данной книги. Отметим лишь, что в результате выполнения этапа построения априорной концептуальной модели объекта должен быть получен перечень параметров, предположительно характеризующих данный объект, а также для каждого из параметров должен быть известен тип (целое число,

вещественное число, качественный показатель и т.п.) и, по возможности, границы диапазона значений .

После получения априорной концептуальной модели объекта (процесса) необходимо ее конкретизировать для получения математического описания объекта – его математической модели. Сделать это можно выполнив некоторую **познавательную процедуру**. Все познавательные процедуры, требующие обращения к объекту познания, должны включать функцию, которую обычно называют **восприятием**, хотя с точки зрения биологической аналогии правильнее говорить о **рецепции**. В технике рецепция есть реализация некоторого процесса, общего для объекта и технической системы, несущего необходимую информацию об объекте.

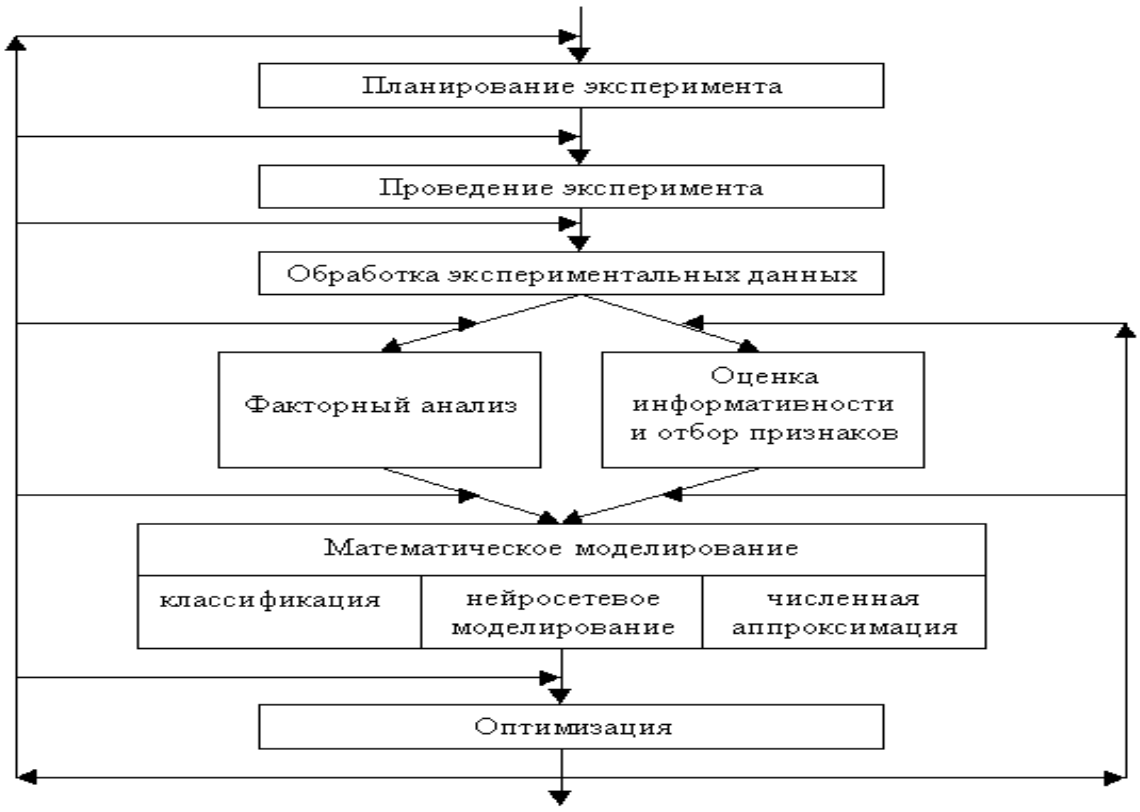



Рис. 1.2 - Схема процесса интеллектуальной диагностики

Во многих случаях для осуществления рецепции требуется активное воздействие на объект – его **стимуляция**, или, по крайней мере, задание **условий его функционирования** (**факторов**).

Этап **планирования эксперимента** ставит своей задачей получение модели эксперимента (испытаний), позволяющей при минимальном количестве испытаний изделий на надежность, долговечность и т.п. получить экспериментальные данные, характеризующие свойства объекта, на основе которых можно построить его математическую модель, удовлетворяющую определенным требованиям, главнейшим из которых является **адекватность модели**.

После построения плана эксперимента логичным является осуществление **этапа проведения эксперимента**. На котором осуществляются испытания объектов и собираются эксериментальные данные, характеризующие эти объекты.

После сбора экспериментальных данных об объекте познания эти данные необходимо **обработать** определенным образом, чтобы исключить (или по крайней мере уменьшить) погрешности и другие негативные факторы, оказавшие влияние на результаты измерений. Для этого используют **фильтрацию** и **сглаживание** данных, то есть осуществляют **селекцию** или **очистку информации**. На физическом уровне очистка связана с выбором принципа рецепции, на сигнальном – с организацией инвариантных структур и калибровок, на уровне первичных данных и более высоких уровнях – с введением поправок. Соответственно вопросы селекции затрагиваются в большинстве теорий, относящихся к познавательным процедурам. По-видимому, недостает лишь общего подхода к селекции, который объединил бы все ее возможные уровни.

Кроме фильтрации и сглаживания данных, на практике часто необходимо найти значения некоторых функций, характеризующих определенные свойства данных, (например, математическое ожидание, дисперсию), которые будут необходимы на следующих этапах диагностики, то есть необходимо выполнить **статистическую обработку экспериментальных данных**.

Имея обработанные экспериментальные данные, далеко не всегда сразу можно построить адекватную модель объекта познания, поскольку не все параметры с помощью которых охарактеризован данный объект на этапе построения априорной концептуальной модели имеют значение для построения модели, кроме того, некоторые параметры могут оказывать негативное влияние на модель.

Поэтому на следующем этапе необходимо **оценить информативность** (значимость) параметров (признаков) объекта познания и **отобрать** наиболее значимые из них для использования на следующих этапах диагностики. Выполнение этого этапа позволяет не только улучшить модель, исключив вредные и ненужные параметры, но и упростить и удешевить процесс диагностики за счет сокращения измерений параметров.

К сожалению, при построении концептуальной модели объекта познания не всегда удается выбрать такой набор признаков, который был бы тесно связан с теми параметрами модели, моделирование которых необходимо осуществить. Зачастую некоторые параметры, тесно связанные с моделируемыми параметрами объекта познания, в набор признаков непосредственно не включаются, а присутствуют в нем косвенно через другие параметры, связанные с ними. Поэтому для выяснения наличия таких скрытых параметров и определения их связи с параметрами, входящими в концептуальную модель необходимо осуществлять **этап факторного анализа**.

После обработки и анализа экспериментальных данных, а также выполнения отбора информативных признаков можно приступать к **математическому моделированию** объекта познания, которое может заключаться в моделировании качественной либо количественной (численной) связи между определенными параметрами объекта познания, для чего необходима **локализация** познавательной процедуры, т.е. задание (или нахождение) той области во времени и в физическом пространстве или в структуре объекта, к которой должен быть отнесен результат процедуры. В теориях таких познавательных процедур как поиск и диагностика, операции сканирования, задачи датирования результатов измерений и т.п. эта функция является важнейшей. Отметим, что в ряде случаев стремятся к тому, чтобы

область локализации была возможно меньшей, но есть ситуации, когда ее желательно максимально расширить (как, например, при статистическом анализе стационарных процессов).

Получив модель объекта познания, эту модель, как правило необходимо **оптимизировать** с целью того, чтобы она как можно ближе характеризовала объект познания.

После построения и **оптимизации** математической модели объекта познания необходимо оценить ее достоверность – **адекватность** объекту познания при определенных допущениях.

Однако, выполнение всех данных этапов последовательно друг за другом далеко не всегда приводит к получению адекватной модели и на практике, как правило, необходимо многократно повторять отдельные этапы процесса диагностики для получения приемлемой модели.

### 1.6 Диагностика и прогнозирование

В связи с возрастающей ролью автоматических и автоматизированных систем возрастает значение предвидения их состояния. Без предвидения нельзя управлять состоянием системы, нельзя своевременно предупреждать аварийные ситуации. **Теория прогнозирования технического состояния** располагается на стыке ряда научных дисциплин и теорий, таких, как теория надежности, техническая диагностика, основы технических измерений и других. Прогнозирование технического состояния непосредственно примыкает к **теории надежности**, так как главная **цель прогнозирования** заключается в своевременном обнаружении неблагоприятного состояния системы (изделия) и разработке рекомендаций, которые, в конечным счете, направлены на повышение его надежности и эффективности.

Использование теории и методов прогнозирования для анализа надежности изделий создает возможность существенно повысить эффективность оценки надежности их на различных этапах разработки, изготовления и эксплуатации

(сокращение объема и времени испытаний, повышение достоверности расчетов).

В процессе создания изделия, его производства и настройки, а также подготовки к эксплуатации и самой эксплуатации очень важно уметь определять его **техническое состояние**, т.е. знать, какими характеристиками обладает изделие в данный момент времени. Эта задача решается средствами обычного **технического контроля**, позволяющего получать данные об измеряемых технических параметрах в момент их измерения.

С появлением технических систем, выполняющих ответственные функции, возрастает роль **предвидения технического состояния** в некоторый будущий отрезок времени, с тем чтобы можно было своевременно принять меры по предотвращению отказов. В процессе развития техники возникла **задача управления техническим состоянием** больших систем путем своевременного переключения на резерв, своевременного перехода на новые рабочие режимы и т. п. Но управлять без прогнозирования ожидаемого состояния нельзя. Таким образом, новые этапы развития техники вызвали к жизни новую техническую проблему - **проблему прогнозирования технического состояния**.

Для изделий важно установить не только то, что они исправны в данный момент времени (в период контроля), но и то, что они будут продолжать оставаться исправными на протяжении некоторого будущего интервала времени. В дальнейшем оказалось, что прогнозирование технического состояния важно не только для периода эксплуатации, но и для периода проектирования изделий и для процесса производства.

При **проектировании** следует высказать обоснованное предположение о технических характеристиках, которыми будет обладать будущее проектируемое изделие.

В процессе **производства** по результатам испытаний ограниченного объема (малых выборок, небольших продолжительностях и т. д.) делается предположение о технических характеристиках и работоспособности больших партий на больших временных интервалах. По результатам ускоренных испытаний делается прогноз о предполагаемом состоянии изделий в нормальных условиях. Словом, и в процессе

производства имеет место прогнозирование технического состояния изделий.

**Прогнозировать событие** — значит предвидеть, предсказать будущее событие на основании изучения таких факторов, от которых оно зависит или которые ему сопутствуют. Научное прогнозирование основывается на изучении объективных закономерностей, которым подчиняются интересующие нас процессы и события. При этом используются две группы закономерностей: — закономерности случайных событий или вероятностные (стохастические) и закономерности детерминированные.

При прогнозировании события можно выделить два характерных подхода к решению поставленной задачи:

- прогнозирование будущего состояния данного события на основании изучения закономерности изменения данного события;

- прогнозирование будущего состояния данного события на основании изучения другого события (или группы других событий), связанного с данным.

Все сказанное о прогнозировании в общем плане имеет непосредственное отношение к прогнозированию технического состояния и надежности изделий.

**Техническое состояние изделия** определяется значением технических параметров, от которых зависит его работоспособность. Изменение этих параметров обычно вызывается многими причинами, поэтому исключается возможность установить однозначную связь между изменением параметра и причинами, вызывающими такое изменение.

Прогнозирование надежности, основанное на наблюдении прямых или косвенных прогнозирующих параметров, позволяет исследовать надежность конкретных изделий в процессе их работы. Это обстоятельство приобретает особую важность для изделий, которые изготавливаются в небольшом числе экземпляров и выполняют ответственные функции. Для них может оказаться совершенно недопустимой ориентация на оценку надежности по числу зафиксированных отказов, так как главным требованием может быть предупреждение отказов.

Прогнозирование технического состояния и надежности можно осуществлять на различных стадиях создания и использования изделий: на стадии

проектирования, производства и эксплуатации. На этих стадиях математические основы прогнозирования сохраняются общими, однако конкретные методики и алгоритмы различны.

На **стадии проектирования изделий** исходными данными являются предполагаемые характеристики проектируемого изделия, рабочие режимы и предполагаемые условия работы. Целевая направленность прогнозирования на этом этапе - создание конструкции, которая наилучшим образом удовлетворяет предполагаемым условиям работы.

На **стадии эксплуатации изделий** исходными данными являются предполагаемые закономерности изменения технических параметров реального изделия. Целью прогнозирования технического состояния при эксплуатации является своевременное предупреждение отказов и применение таких рабочих условий и обслуживания изделий, которые наилучшим образом отвечают задаче обеспечения заданной надежности и эффективности.

### 1.7 Основные направления прогнозирования

На **принцип прогнозирования** влияют различные факторы, но основные из них - совокупность имеющихся параметров, целевая направленность поставленной задачи и рабочий алгоритм. Совокупность прогнозируемых параметров $x_1$, $x_2$, ..., $x_n$, определяющих состояние аппаратуры, можно представить различным образом: значениями параметров в моменты времени t, распределениями параметров, комплексными показателями и т. д.

**Процесс прогнозирования** преследует различные цели. Он позволяет определить: 1) протекание процесса на протяжении будущего отрезка времени в конкретной размерности; 2) ожидаемую вероятность того, что исследуемый процесс не выйдет за установленные допусковые границы; 3) к какому классу по долговечности следует отнести исследуемый процесс. В зависимости от прогнозируемых параметров и целевой направленности прогнозирования выбираются имеющиеся методы и математический аппарат.

Сформулируем **задачу прогнозирования**, подходя к этому с позиции **первого направления**. Пусть контролируемый процесс, характеризующий состояние, можно представить в виде многомерной функции $y(x_1, x_2, ..., x_k)$, которая наблюдается в период времени от 0 до $t_n$, вследствие чего известны значения этой функции $y(t_0)$, $y(t_1)$, ..., $y(t_n)$ соответственно в моменты времени $t_0$, $t_1$, ..., $t_n \in T_1$. Необходимо определить значения этой функции $y(t_{n+1})$, $y(t_{n+2})$, ..., $y(t_{n+m})$ в моменты времени $t_{n+l}$, $t_{n+2}$, ..., $t_{n+m} \in T_2$.

Подобную задачу можно решить как в явном виде, определяя непосредственно y(x, t), так и косвенным путем, находя сначала каждый параметр $x_s$, а затем уже y(x, t). Подобная постановка задачи справедлива в предположении, что значения $y(x, t_0)$, ..., $y(x, t_n)$ предопределяют величины $y(x, t_{n+1})$, ..., $y(x, t_{n+m})$, иными словами, что процесс «информативен» во времени. Возможность подобного допущения зависит от степени изученности прогнозируемого процесса, т. е. объема данных о процессе, полученных в период времени $T_1$ от 0 до $t_n$. Идеальным случаем при этом является получение аналитического выражения для функции состояния y(x,t). Задачу прогнозирования в подобной постановке можно решить различными методами, отличающимися применяемым математическим аппаратом и называемыми методами **аналитического прогнозирования**.

**Второе направление** прогнозирования связано с определением вероятности невыхода процесса за установленные ограничения. Эту задачу можно сформулировать следующим образом: пусть известны значения параметров $x_s$ (s = l, 2, ..., k), полученные в моменты времени $t_i$ (i = 0, 1, 2,..., n), и в каждый момент $t_i$ функция состояния $y(x,t_i)$ полностью характеризуется функцией распределения $F_i(y)$. Необходимо по известным значениям $x_s(t_i)$, $y(x_s, t_i)$, $F_i(y)$, $t_i \in [0 ... t_n]$ вычислить

$$F_{n+j}(\varepsilon) = P\{|y(x, t_{n+j}) - y_н(x)| < \varepsilon\},$$

где $\varepsilon = y^*(x_s) - y_н(x_s)$, $y_н(x_s)$ - номинальное (оптимальное), а $y^*(x_s)$ - допустимое значение функции y(x, t) в области $t_{n+1} ... t_{n+m}$ для значений $t_{n+j}$ (j = 1, 2, ..., m). Методы, основанные на таком решении задачи прогнозирования, назовем **методами вероятностного прогнозирования**.

**Третье направление прогнозирования** предусматривает отнесение

контролируемого (диагностируемого) объекта к одному из временных классов. Задача прогнозирования формулируется следующим образом: пусть в момент времени $t_0$ или в ограниченный начальный период времени получены значения параметров диагностируемого объекта $x_1$, $x_2$, ..., $x_k$, характеризующих функцию состояния y(x). Необходимо по совокупности параметров $x_s$ координат многомерной функции y(x) принять решение о принадлежности объекта к тому или иному классу $K_q$, где $K_q$ могут быть параметрическими, временными и другими.

Множество и размер классов определяются специфическими техническими особенностями прогнозируемых объектов. Методы, основанные на отнесении исследуемых объектов к одному из классов, будем называть методами **статистической классификации**. В них используется аппарат **теории распознавания образов**, а также **теории искусственных нейронных сетей**.

### 1.8 Способы прогнозирования

В рамках рассмотренных направлений существуют разновидности основных постановок задачи прогнозирования, которые получили название **способов прогнозирования**. При этом наиболее часто используется **первая группа способов**, т. е. решается прямая (**прямое прогнозирование**) или обратная (**обратное прогнозирование**) задача.

**1). Прямое прогнозирование.** В этом случае при аналитическом прогнозировании, предполагая наличие связей между характеристиками процесса $y(x_1, x_2, . . ., x_k, t_i)$, $t_i \in T_1$; i = 0,1, ..., n и $y(x_1, x_2, . . ., x_k, t_{n+j})$; $t_{n+j} \in T_2$; j = 1, 2, ..., m, причем $T_1 \cup T_2$, и получая из эксперимента или расчетным путем значение $y(x_s, t_i)$, находят аналитическое выражение зависимости $y(x,t_{n+j})=\varphi[y(x,t_i)]$, которое позволяет определить значение процесса для любого момента времени $t_{n+j} \in T_2$, j = 1, 2, ..., m. При вероятностном решении задачи прямое прогнозирование предусматривает получение зависимости, аналогичной: $P_y(t_{n+j}) = \varphi_1[f_{ti}(x_1)\ f_{ti}(x_2)\ ...\ f_{ti}(x_k)] = \varphi_2[f_{ti}(y)]$, где $P_y(t_{n+j})$ - прогнозируемая вероятность; $f_{ti}(y)$, $f_{ti}(x)$ – плотности распределения вероятностей значений процесса y и его координат $x_s$; $\varphi_1$, $\varphi_2$ -

соответствующие функциональные зависимости, выражающие характер связей.

**2). Обратное прогнозирование**. Идея обратной задачи заключается в определении времени $t_ж = t^*$ (долговечности или времени «жизни» изделия), когда характеристика процесса y(x, t) или вероятности P(y) достигают предельных значений, задаваемых наложенными ограничениями.

При **аналитическом обратном прогнозировании** в выражение $y(x,t_{n+j})=\varphi[y(x,t_i)]$ вводится предельное значение $y^*(x)$ и полученное уравнение решается относительно $t_{n+j} = t^*$, т. е. находится $t^*$ в явном виде. Таким образом, величину $t^*$, как результат вероятностного обратного прогнозирования, можно найти из следующего выражения:

$$P\{|y(x_s, t^*) - y_н(x_s)|< E\} = P^*(y),$$

где $P^*(y)$ - допустимая вероятность нахождения функции y в заданной области. При статистической классификации процессов и образовании временных классов $K_q = T_{q-1}...T_q$ (q = l, 2,...) возникает задача, относящаяся также к категории задач обратного прогнозирования, которые могут быть решены методами теории распознавания образов. Примерами, которые иллюстрируют необходимость решения обратных задач, может служить определение долговечности, сроков профилактических работ, сроков выполнения контроля и т. п.

**Другая группа способов** классифицируется по **направлению аргумента** при осуществлении прогнозирования. Она объединяет три способа, которые получили название **прогнозирования вперед**, в настоящем и назад (генетическое).

**1). Прогнозирование вперед.** В подавляющем большинстве практических случаев прогнозирование связано с определением состояния в последующие значения аргумента в области будущих моментов времени, т. е. на основе предыстории определяется предстоящая ситуация. В этом случае для временного аргумента $y(x, t_i)$ и $y(x, t_{n+j})$ должно соблюдаться условие: $t_0<t_1<...<t_{n+m}$, т. е. аргумент всегда возрастает. Такое прогнозирование можно определить как **перспективное**.

**2). Прогнозирование в настоящем.** Этот способ соответствует задаче прогнозирования по множеству, сформулировать которую можно следующим образом. Пусть в результате контроля получена ограниченная информация

(выборка) $\{x\}_l$ о состоянии диагностируемого множества (генеральной совокупности) $\{x\}_L$. Необходимо, зная состояние или свойства (уровень качества, степень работоспособности и т. д.) $\{x\}_l$, оценить состояние всего множества $\{x\}_L$. В данном случае необходимо осуществить **экстраполяцию** (распространение) свойств выборки на свойства генеральной совокупности.

**3). Прогнозирование назад.** В некоторых случаях требуется оценить процесс в прошлом по информации, полученной в определенный интервал времени. Такие задачи возникают тогда, когда по техническим или другим причинам нельзя определить величину y(x, 0), а знание ее необходимо. Отличие в решении подобных задач заключается в том, что необходимо переставить местами области $[0 \ldots t_n]$ и $[t_{n+1} \ldots t_{n+m}]$, при этом значения аргумента не возрастают, а убывают. Подобная постановка задачи имеет много сходства с генезисом и поэтому удобно назвать решение такого варианта задачи **генетическим прогнозированием**.

Принципиально важными для практики являются способы **индивидуального** и **группового прогнозирования**.

**1). Индивидуальное прогнозирование.** Особенность решения подобной задачи наиболее удобно объяснить на примере прогнозирования изменения состояния технических изделий. В этом случае для получения прогноза экспериментально исследуется функция состояния y(x, t) индивидуального образца технического изделия в области $T_1$ и осуществляется оценка поведения этой функции в области $T_2$, причем, как и раньше, $T_1 \cup T_2$.

**2). Групповое прогнозирование.** При этом рассматривается целая группа однородных процессов (например, изменение состояния целой группы технических изделий), получаются и анализируются их статистические характеристики (средние значения, элементы ковариационных матриц), полученные в области $T_1$.

Множество методов решения задачи прогнозирования имеет одну **общую идею**: обнаружение **экстраполяционных** связей, существующих между прошлым и будущим, между информацией о процессе в контролируемый период времени и характером протекания процесса в последующем. Очевидно, что характер экстраполяционных связей будет определять аппарат решения задачи

прогнозирования, а от того, насколько точно описаны рассматриваемые связи, будет зависеть точность прогнозирования. Поскольку эти связи могут быть **детерминированными**, **квазидетерминированными**, **вероятностно-детерминированными** и т. п., то часто задачи более эффективно решаются при комбинировании методов и математического аппарата различных направлений прогнозирования. Так, достаточно перспективным является совместное использование статистической классификации и аналитического прогнозирования. **Статистическую классификацию** можно рассматривать как «грубое» прогнозирование, определяющее временной класс обычно величиной в несколько тысяч часов, к которому относится диагностируемое изделие, а **аналитическое прогнозирование** указывает конкретную величину, например, функцию состояния внутри соответствующего класса, т. е. уточняет результат **предварительного прогноза**.

**Успешность решения задачи прогнозирования** зависит от ряда условий: объема и качества информации о прогнозируемом процессе; правильности формулировки задачи прогнозирования и обоснованности выбора метода ее решения; наличия вычислительных средств и вычислительного аппарата для решения задачи в соответствии с выбранным методом. Отсутствие любого из этих условий может сделать невозможным прогнозирование.

Важнейшим из них является **формулировка задачи**, так как она определяет требования к объему и качеству информации, математический аппарат и точность прогноза. **Объем и качество информации**, естественно, обусловливают успех прогнозирования. Информация о прогнозируемом объекте (процессе) получается из результатов контроля. **Контроль** может быть непрерывным, периодическим и однократным.

## 1. 9 Показатели качества прогнозирования

Чтобы обосновать выбор того или другого метода прогнозирования, необходимо иметь возможность количественно оценить **качество прогнозирования** на основе данного метода. Каждый метод прогнозирования желательно сопровождать своим вполне определенным значением **показателя качества**, изменяющимся в зависимости от формулировки задач и условий ее решения. Но это чрезвычайно трудно. В каждом конкретном случае прогнозирования возможны различные методы и каждый из них характеризуется не одним показателем качества, а набором показателей, изменяющихся при изменении формулировки задачи и условий ее решения. К числу наиболее важных показателей качества прогнозирования относятся:

**1). Точность прогнозирования** $K_T$, которая характеризуется степенью соответствия величины, полученной в результате прогноза, и величины действительной. Она измеряется величиной ошибки $\Delta\varphi$, равной разности между величиной $\varphi_{пр}$, полученной в результате прогноза, и действительной, истинной величиной $\varphi$: $\Delta\varphi = \varphi_{пр} - \varphi$. Если осуществляется вероятностное прогнозирование, ошибка $\Delta\varphi$ носит случайный характер и представляется двумя показателями: средним значением $M_{\Delta\varphi}$ и дисперсией $D_{\Delta\varphi}$. В инженерной практике часто бывает удобно оценивать точность прогнозирования возможным интервалом значений прогнозируемой величины (**точность оценки**) и вероятностью того, что именно в этот интервал попадет истинное значение прогнозируемой величины (**достаточность оценки**).

**2). Достоверность прогнозирования** $K_D$, которая совпадает с понятием достоверности оценки, полученной в результате прогнозирования. Точность и достоверность - взаимосвязанные понятия. Часто под достоверностью прогнозирования понимают **надежность прогнозирования**.

**3). Быстродействие прогнозирования**, измеряемое затратами времени на процесс прогнозирования $K_б$. Разновидностью этого показателя является отношение

времени прогнозирования к времени, на которое распространяется прогнозирование.

**4). Стоимость прогнозирования** $K_C$, измеряемая затратами материальных средств на операцию прогнозирования, т. е. на создание специальной аппаратуры и на эксплуатацию этой аппаратуры.

**5). Информационный показатель качества прогнозирования**, который указывает, насколько увеличилась информация об исследуемом объекте в результате прогнозирования: $K_{и} = \sum_{i=1}^{n}(H_{0i} - H_i) \Big/ \sum_{i=1}^{n} H_{0i}$, где $H_{0i}$ и $H_i$ - начальная и конечная энтропии по i-му параметру соответственно.

Энтропия характеризует меру неопределенности состояния объекта: $H(x) = -\sum_{i=1}^{n} p_i \log p_i$, где $p_i$ - вероятность возможного i-го состояния объекта, n — число всех возможных состояний.

**6). Показатель полноты прогнозирования** $K_п$, который представляет собой отношение числа параметров, охваченных контролем, к общему числу параметров, определяющих работоспособность изделия: $K_п = n / N$.

**7). Показатель эффективности прогнозирования** $K_э$, который показывает, насколько улучшились эксплуатационные характеристики исследуемого изделия в результате прогноза, и является обобщенным показателем качества прогноза. Смысл показателя $K_э$ будет различным для различных объектов. В случае, когда целью прогнозирования является повышение надежности объекта, показателем эффективности будет абсолютное, либо относительное изменение показателя надежности в результате прогноза. В частном случае это может быть относительное изменение коэффициента готовности изделия $K_г$:

$K_э = (K_{г2} - K_{г1}) / K_{г1}$, $K_г = T_0 / (T_0 + \tau)$.

Время восстановления $\tau$ существенно сокращается при проведении прогнозирования и поэтому $K_{г2}$ может становиться близким к единице, а относительный показатель эффективности будет тем больше, чем меньше начальное значение коэффициента готовности $K_{г1}$.

# ГЛАВА 2. ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

## 2.1 Сглаживание сигналов

Во многих случаях кривая изменения диагностического параметра существенно искажается за счет неизбежных ошибок измерений. Это свойственно параметрам, записываемым вручную по показаниям стрелочных приборов или при недостаточной точности измерений и т. п. В таких случаях целесообразно проводить анализ предварительно сглаженных кривых.

Существуют два основных метода сглаживания: метод наименьших квадратов и метод преобразования.

Выбор метода сглаживания и весовых коэффициентов определяется особенностями поведения кривой x(t), характером случайных отклонений, задачами диагностики и осуществляется на основании практического опыта. Метод сглаживания должен исключить случайные погрешности, но сохранить общую тенденцию изменения параметра.

По **методу наименьших квадратов** кривая x(t) на участке от $t_1$ до $t_2$ заменяется полиномом $x(t) \approx a_0+a_1t+a_2t^2+...$, где $a_0$, $a_1$, ... — параметры, подлежащие определению. Не рекомендуется применять полиномы степени выше третьей, что делает аппроксимацию слишком «жесткой», лучше уменьшить длину участка, в пределах которого осуществляется аппроксимация. В таком случае кривая x(t) заменяется полигональной кривой из отрезков прямых или парабол - **метод сплайнов**.

**Метод преобразования** состоит в преобразовании исходных значений $x_j$ в другие «сглаженные» значения $x^*_j$.

Часто применяется **метод скользящего среднего**. По этому методу величина $x^*_j$, представляет собой среднее нескольких значений, непосредственно примыкающих к измерению при $t_1$:

$$x_j^* = \frac{1}{2n+1}\sum_{p=j-n}^{j+n} x_p \ . \qquad (2.1)$$

Практически осреднение проводится не более чем для 10 соседних значений параметра. Естественно, что в начале и конце общего интервала времени для n точек сглаженные значения не могут быть получены. Это обстоятельство либо не принимается во внимание, либо кривая сглаженных значений по касательной экстраполируется на граничные точки.

Применяется **способ повторного сглаживания**, в результате которого находятся значения $x^{**}_j$. Это эквивалентно тому, что в равенство (2.1) значения $x_j$ входят с другими весовыми коэффициентами:

$$x_j^{**} = \frac{1}{2r+1}\sum_{p=j-r}^{j+r} x_p^* = \frac{1}{2r+1}\sum_{p=j-r}^{j+r}\frac{1}{2n+1}\sum_{p=j-n}^{j+n} x_p \ .$$

Сопоставляя с формулой (2.1), находим, что по мере удаления от $x_j$ весовые коэффициенты уменьшаются, что характерно для **процесса релаксационного сглаживания**. Сумма весовых коэффициентов при релаксационном сглаживании всегда равна единице. Последнее очевидно, если применить формулу (2.1) для параметра, имеющего постоянное значение. Одна из простых процедур релаксационного сглаживания может быть получена по формуле:

$$x_j^* = \alpha x_j + (1-\alpha)x_{j-1}^* ,$$

где параметр релаксации $0<\alpha<1$. При $\alpha=1$ сглаживания не происходит, при $\alpha=0$ сглаженная функция получает постоянное значение. В практических задачах используется $0{,}1 < \alpha < 0{,}5$.

**Метод экспоненциального сглаживания** можно представить в виде процедуры:

$$x_j^* = \frac{1}{N}\sum_{p=0}^{j}\left(\frac{N-1}{N}\right)^{n-p} x_p ,$$

где N - некоторый параметр сглаживания.

Мерой качества предварительной обработки сигнала является **эффективность обработки** $Э_о$, которая непосредственно влияет на качество распознавания (прогнозирования) и будет тем выше, чем больше подавлены составляющие помехи. Количественно эффективность обработки оценивается как отношение

$$Э_{о} = \frac{D(x^{*}(t))/D(z^{*}(t))}{D(x(t))/D(z(t))},$$

где D – дисперсия соотвествующих составляющих, x(t) – исходный сигнал, $x^{*}(t)$ – обработанный (сглаженный) сигнал, z(t) и $z^{*}(t)$ – помехи, содержащиеся в исходном и обработанном сигналах, соответственно.

## 2.2 Нормирование и масштабирование сигналов

При обработке наборов признаков методами распознования и прогнозирования часто приходится сталкиваться с тем, что признаки имеют не только разные физические размерности но и разный масштаб, что может оказать существенное влияние на эффективность построения модели. Поэтому для снижения негативного влияния этих факторов перед обучением рапознаванию и перед распознаванием значения признаков нормируют и/или масштабируют.

Правила нормирования и масштабирования на практике подбирают на основе эвристических соображений так, чтобы максимально повысить эффективность соотвествующих методов.

В качестве правила нормирования можно использовать формулу:

$$x^{s} = \frac{x^{s} - Min(x)}{Max(x) - Min(x)},$$

где $x^{s}$ – s-ое значение переменной x, Min(x) – минимальное значение переменной x, Max(x) – максимальное значение переменной x.

Разнормирование переменной x производят по формуле:

$$x^{s} = x^{s}(Max(x) - Min(x)) + Min(x).$$

## 2.3 Квантование сигналов

На практике большинство сигналов являются непрерывными, в то время, как некоторые методы распознавания и прогнозирования способны работать только с конечным множеством значений, и, в таких случаях, возникает задача преобразования непрерывных сигналов в дискретный набор отсчетов мгновенных значений уровня сигнала – задача **квантования сигналов по времени**.

Пусть имеется непрерывный сигнал x(t), который можно измерить на интервале $[t_1,t_2]$ в N равноотстоящих друг от друга точках $x_i=x(t_1+i\Delta t)$, i=1,2,...,N, с интервалом времени $\Delta t = ( t_2 - t_1) / N$. Тогда, допуская некоторую ошибку $o(\Delta t)$ сигнал x(t) на интервале $[t_1,t_2]$ можно заменить на набор дискретных отсчетов $x=\{x_1, x_2, ..., x_i, ..., x_N\}$.

Однако, для использования ряда методов распознавания квантования сигнала по времени бывает недостоточно. Это касается тех методов, которые способны работать только с дискретной информацией. В этом случае сигнал необходимо **квантовать** еще и **по уровню**.

Пусть на интервале $[t_1,t_2]$ в точке i сигнал $x(t_1+i\Delta t)$ может принимать вещественные значения $x_i$ в диапазоне $[a_1,a_2]$. Тогда, задав шаг $\Delta a$ и допуская некоторую ошибку $o(\Delta a)$, будем заменять $x_i$ набором Z дискретных отсчетов $\{x_{ij}\}$, j = 1, ..., Z,

где $x_{ij} = \begin{cases} 0, \text{если } x_i < j\Delta a, \\ 1, \text{если } x_i \geq j\Delta a, \end{cases}$,

$Z = (a_2-a_1) / \Delta a$ .

# ГЛАВА 3. СОКРАЩЕНИЕ РАЗМЕРНОСТИ ДИАГНОСТИЧЕСКОЙ ИНФОРМАЦИИ

## 3.1 Задача отбора и критерии оценки информативности признаков

В настоящее время признано, что распознавание сложных образов наиболее целесообразно проводить на основе их относительного описания (описания в пространстве признаков). Выбор информативной системы признаков - наиболее важная задача теории распознавания. Однако удовлетворительного решения, определяющего порядок автоматического отыскания последних посредством переработки информации, получаемой на уровне абсолютного описания объекта, пока не найдено. Поэтому основным решением остается автоматизированный выбор наиболее информативных признаков из некоторого исходного множества (ансамбля) свойств, задаваемого эвристически.

От успешного решения данной задачи зависят процент ошибок на этапе экзамена и распознавания, быстродействие и объем памяти распознающего устройства. В настоящее время при решении этой задачи в лучшем случае формализуется лишь процедура выбора наиболее информативных признаков из заранее заданного ансамбля свойств. Иногда, правда, формализуется и процедура наращивания этого ансамбля в случае необходимости. Однако при этом, как правило, реализуется лишь получение заранее предписанных свойств.

До настоящего времени отсутствует формальная постановка задачи отбора признаков. В неформальных постановках задачи определение информативных признаков преследует:

1) уменьшение до минимума количества необходимых для описания классов признаков без существенного увеличения вероятности ошибки распознавания;

2) возможность использования относительно простых алгоритмов распознавания;

3) уменьшение вероятности ошибки распознавания.

С решением этой задачи обычно связаны вопросы упрощения распознающей системы и повышения качества ее работы.

К построению информативной системы признаков может быть два подхода.

Первый подход заключается в том, что с самого начала берется постановка на отыскание малого числа признаков большой информативности. Однако все используемые при этом методы до сих пор основаны на эвристике и эмпирике, т. е. выбор признаков определяется интуицией, опытом и воображением разработчика. Как бы удачна ни была сконструированная система признаков, нельзя доказать, что она лучше некоторой другой.

Второй подход заключается в том, что из большого числа исходных признаков согласно некоторому критерию информативности признаков выбирается как можно меньшее число наиболее полезных для распознавания признаков.

Оба рассмотренных подхода к построению информативной системы признаков требуют определения критериев оценки информативности признаков.

К настоящему времени разработаны разнообразные критерии информативности признаков, основанные на методах математической статистики и теории информации. Наибольшее признание из них получили критерии, отражающие расстояния между распределениями классов.

Из рассмотренных выше подходов к построению информативной системы признаков второй подход является более конструктивным, чем первый. На его основе можно функционально связать критерий информативности признаков с вероятностью ошибки распознавания.

Полезность некоторого признака в исходной совокупности n признаков определим по приращению полной вероятности ошибки $\Delta P_{ош}$ при исключении этого признака из исходной совокупности:

$$\Delta P_{ош} = P_{ош} - P'_{ош},$$

где $P_{ош}$ — полная вероятность ошибки распознавания классов $K_1$ и $K_2$ для исходной совокупности n признаков; $P'_{ош}$ — полная вероятность ошибки распознавания классов $K_1$ и $K_2$ при исключении k-го признака из исходной совокупности. В зависимости от знака приращения $\Delta P_{ош}$ могут иметь место

следующие случаи:

$_{\Delta P_{ош}}<$ 0 - k-ый признак полезен, так как его исключение из исходного описания приводит к увеличению вероятности ошибки;

$_{\Delta P_{ош}}=$ 0 - k-ый признак бесполезен, так как его исключение из исходного описания не изменяет вероятности ошибки;

$_{\Delta P_{ош}}>$ 0 - k-ый признак вреден, так как без него вероятность ошибки распознавания уменьшается.

Такой подход к определению критерия полезности признаков предполагает использование конкретного решающего правила, поскольку только в его рамках имеет смысл ошибка распознавания.

Если существование полезных или бесполезных признаков не вызывает никаких сомнений, так как подтверждается большим количеством легко конструируемых примеров, то концепция «вредности» признаков на первый взгляд кажется спорной. Однако она не противоречит утверждению, что вредной информации не существует. Информация о вредности признака есть полезная информация; весь вопрос о том, правильно ли она используется.

Трудность восприятия и осознания концепции вредности признака заключается в том, что она возникает в чистом виде только при различении двух классов. В случае большего числа классов «абсолютной» вредности признака, как правило, не бывает: вредности признака при различении одних пар классов противостоит его полезность при различении других пар.

«Исчезновение» вредных признаков при различении более чем двух классов только кажущееся. Оно происходит за счет усреднения информативности признаков по всем парам классов в условиях преобладающего числа полезных признаков. Отрицательное же влияние признаков, вредных для различения тех или иных классов, не исчезает и выражается в увеличении вероятности ошибки распознавания этих классов, а, следовательно, и суммарной ошибки распознавания. При распознавании более чем двух классов может возникнуть «порочный» круг: включение некоторого признака в описание классов окажется полезным для различения одних пар классов, но вредным для различения других; исключение

этого признака из описаний классов, наоборот, окажется вредным для различения первых классов и полезным для различения вторых. Следствие такого противоречия - обязательное возрастание количества ошибок распознавания с увеличением размера алфавита классов при любых решающих правилах, использующих один эталон на класс.

Только на основе анализа полезности, бесполезности или вредности некоторого признака при разделении каждой из пар классов заданного алфавита можно решить альтернативу включения или исключения этого признака из исходного описания с точки зрения минимизации ошибки распознавания.

### 3.2 Выбор прогнозирующих параметров

Выбор прогнозирующих параметров технических изделий в настоящее время редставляется весьма трудной задачей. В каждом конкретном случае приходится решать задачу их выбора своим оригинальным путем, особенно если задачи прогнозирования решаются различными методами. Однако можно сформулировать некоторые рекомендации (методы) по выбору прогнозирующих параметров применительно к различным техническим изделиям.

Решение проблемы повышения качества изделий, весьма актуальной для современной промышленности, связано с рассмотрением множества задач, среди которых важное место занимает задача принятия решения о качестве изделия. Объективность принимаемого решения зависит от источников информации о качестве изделия. Информацию о состоянии изделия несут различные признаки (факторы), и если учесть, что понятие качества – комплексно и многогранно, то количество показателей, описывающих все стороны качества изделия, может достигать нескольких десятков и даже нескольких сотен. Очевидно, не все из них равнозначны, не все содержат необходимое количество информации о качестве изделия. Естественно, возникает задача: сколько признаков и какие признаки нужно контролировать, чтобы принять решение о качестве изделия, удовлетворяющее заданному критерию, то есть необходимо осуществлять отбор наиболее важных для

оценки качества показателей или признаков, которые называются информативными. При этом значимость или информативность признаков нужно оценивать количественно, так как только в этом случае можно отдать предпочтение тому или иному признаку, что трудно сделать при качественном подходе к отбору признаков.

Следует отметить, что физический подход к выявлению информативных признаков, основанный на анализе физической модели изделия, не противопоставляется математическому, зависящему от конкретно поставленной задачи. На первом этапе используется физический анализ, на основе которого отбирается вся совокупность измеряемых признаков (в общем случае избыточная по заданному критерию), в той или иной степени характеризующих качество изделий. На втором этапе применяется математический подход, когда из всей избыточной совокупности отбираются наиболее информативные (значимые).

Задачу, решаемую на втором этапе можно кратко сформулировать следующим образом. Пусть имеется множество признаков $x = \{x_1, x_2, ..., x_n\}$, которое представляет собой избыточную совокупность. Необходимо из n признаков отобрать v наиболее информативных по заданному критерию, причем $v \le n$. Не уточняя критерия информативности, можно выдвинуть ряд общих требований к выделенной совокупности информативных признаков:

- совокупность $x = \{x_1, x_2, ..., x_v\}$, должна обладать **достаточной информативностью**, т.е. добавление к ней одного любого признака не должно приводить к значимому изменению заданного критерия;

- совокупность выбранных признаков должна содержать возможно меньшее количество признаков с целью минимизации экономических затрат и технической трудоемкости;

- выделенная совокупность признаков может состоять из количественных и качественных признаков, но должна допускать количественную оценку (ранжировку).

Совокупность признаков, удовлетворяющая выше перечисленным требованиям, называется **оптимальной совокупностью признаков**.

Задача выбора информативных признаков может возникнуть в процессе разработки, изготовления и эксплуатации изделий. При этом в связи со специфичностью задач в каждом случае могут применяться различные методы.

Необходимость определения информативных признаков из имеющейся совокупности обусловлена также тем, что качество прогноза не инвариантно к системе используемых признаков. В практических случаях характеристики классов и прогнозирующие правила определяются по экспериментальным данным ограниченного объема, поэтому добавление неинформативных признаков приводит к более сильному пересечению представителей классов в пространстве признаков, что может ухудшить качество прогноза.

**Информативность признаков** – это величина, количественно характеризующая пригодность признаков или их набора для распознавания классов отказавших и не отказавших изделий. С физической точки зрения, информативность отражает степень взаимосвязи выбранных признаков с процессами, приводящими к отказу.

Информативность набора признаков равна сумме информативности отдельных признаков только при их независимости. В этом случае на основании информативности отдельных признаков можно составлять наиболее информативный набор.

Если признаки зависимы друг от друга, то информативность набора не выражается через информативность отдельных признаков. В этом случае выбор наиболее информативных наборов признаков является самостоятельной задачей.

Исследования показали, что в задачах прогнозирования, в основном, приходится иметь дело с системой статистически зависимых признаков. Поэтому в таких задачах для определения информативной системы целесообразно использовать один из методов отбора информативных признаков.

Система информативных признаков определяется с помощью метода **распознавания образов**. Такая задача интерпретируется как задача определения системы признаков, которые при использовании в алгоритме обучения обеспечивают наилучшее разделение классов обучающей выборки. Желание

оптимизировать набор признаков так, чтобы обеспечить наилучшее разделение классов, при выбранном алгоритме обучения и заданной обучающей выборке приводит к необходимости полного перебора всех подмножеств исходной системы признаков. В большинстве случаев для этой цели требуется чрезмерно большой объем машинного времени из-за того, что для каждого подмножества признаков требуется произвести процесс обучения.

В настоящее время существуют различные подходы к построению информативной системы признаков. Среди них особо следует выделить эвристический, информационный, статистический, вероятностный и нейросетевой подходы. Рассмотрим их подробнее.

### 3.3 Эвристический подход

Для получения «наилучшего» (по выбранному критерию) набора признаков можно рекомендовать следующее правило. Из всех признаков выбирается один (или несколько) наиболее информативный («ценный»); далее, к первому признаку добавляется такой признак из n-1 оставшихся, чтобы информативность пары признаков для прогнозирования была наибольшей; затем к полученной паре признаков добавляется наилучшим образом новый признак и так далее. Процесс заканчивается тогда, когда информативность некоторой совокупности признаков незначительно превосходит информативность совокупности, полученной на предыдущем шаге, или когда достигнут требуемый уровень информативности (или требуемый уровень точности распознавания). Процедура такого отбора признаков называется **алгоритмом сокращенного перебора с добавлением признаков**.

Процесс отбора признаков можно вести и в обратном порядке: сначала выбирается наиболее «информативное» подмножество (n-1) признаков при исходной совокупности n признаков, затем из этого подмножества отбирается наиболее «информативное» подмножество (n-2) признаков из всех возможных подмножеств и так далее. Процедура такого отбора признаков называется **алгоритмом сокращенного перебора с исключением**

**признаков**.

Иногда для упрощения правила отбора признаков рекомендуется использовать такие свойства признаков, как статистическая независимость. При этом различают случаи, когда признаки статистически независимы при объединении классов и когда признаки независимы внутри каждого класса.

В случае статистической независимости «объединенных» признаков (при объединении классов) рекомендуется в качестве критерия для отбора признаков использовать **нформационную меру Шеннона**. Тогда процедура отбора сводится к упорядочению признаков по убыванию количества обеспечиваемой ими информации при прогнозировании (или, что то же самое, возрастанию неопределенности решения - **энтропии**).

Объединяя идеи алгоритмов перебора с сокращением и добавлением признаков, получаем **алгоритм комбинированного перебора**, схематическое описание которого представлено на рис. 3.1.


Рис. 3.1. - Схематическое описание комбинированного метода

В начале работы алгоритма комбинированного перебора поочередно оценивается эффективность классификации для каждого признака в отдельности. После перебора всех признаков в промежуточный набор добавляется тот признак, для которого эффективность классификации окажется наибольшей. Затем к промежуточному набору добавляют поочередно по одному признаку из тех, которые еще не вошли в набор и оценивают их эффективность классификации. Окончательно в набор заносят тот признак, при добавлении которого эффективность классификации окажется наибольшей по сравнению с эффективностью при его отсутствии. Описанный процесс продолжается до тех пор, пока не окажется, что среди оставшихся признаков нет ни одного, включение которого улучшило бы результат классификации.

По окончании процесса добавления признаков начинает работать блок исключения признаков из набора. Из промежуточного набора исключается признак и выясняется, не приводит ли это к увеличению эффективности классификации. Окончательно из набора выбрасывается тот признак, исключение которого максимально улучшает или, в крайнем случае, не ухудшает классификацию. Процесс исключения повторяется до тех пор, пока из промежуточного набора признаков окажется невозможным исключить ни одного, не ухудшая эффективности классификации.

По окончании процесса исключения признаков снова делается попытка добавить к набору признаков такие признаки, которые улучшили бы эффективность классификации. Процедура отбора признаков заканчивается, когда к системе отобранных признаков оказывается невозможным ни добавить ни одного, улучшающего классификацию, ни исключить ни одного признака, не ухудшая классификации объектов обучающей выборки.

В отличие от рассмотренных выше алгоритмов **алгоритм полного перебора** всех возможных комбинаций признаков является наиболее точной процедурой выбора информативных признаков, но его целесообразно применять, если исходная совокупность признаков невелика (меньше 10).

Пусть требуется выбрать из заданных n признаков наиболее информативный набор, состоящий не более чем из v признаков. Составляются все возможные наборы из 1, 2, ... , v признаков. Для каждого набора строятся прогнозирующие правила и прогнозируется оценка достоверности прогнозирования. Вычисленные значения сравниваются и выбирается наиболее информативный набор.

Достоинством эвристического подхода отбора признаков является относительная простота реализации эвристических процедур.

Недостатками этого подхода являются отсутствие строгого математического обоснования эвристических процедур и длительность процесса итеративного перебора признаков.

## 3.4 Информационный подход

### 3.4.1 Отбор признаков по количеству вносимой информации

Сравнительная оценка информативности признаков $x_k$ и $x_s$, k, s = l, ..., n, может быть произведена также на основе определения количества информации, которое получает система в процессе распознавания объектов в результате определения каждого из этих признаков.

Пусть распознаваемый объект может принадлежать лишь одному из m классов, априорные вероятности отнесения этого объекта к определенному классу обозначим $P(K_j)$, j=1, 2, ..., m, а условные плотности распределения значений признаков $f_j(x_k)$, $f_j(x_s)$.

Если признак $x_k$ принимает дискретные значения с вероятностями

$$P\left(x_{k_q}\right)=\sum_{j=1}^{m}P(K_j)P\left(x_{k_q}\mid K_j\right),\ q = l, ..., n_k,$$

то информативность признака $x_k$ составит:

$$J_{x_k}=-\sum_{i=1}^{m}P(K_i)\log P(K_i)+\sum_{j=1}^{m}P(K_j)\sum_{i=1}^{m}\sum_{q=1}^{n_k}P\left(x_{k_q}\mid K_j\right)P\left(K_i\mid x_{k_q}\right)\log P\left(K_i\mid x_{k_q}\right),$$

Если признак $x_k$ - непрерывный и его совместная плотность распределения $f(x_k)=\sum_{j=1}^{m}P(K_j)f_j(x_k)$, где $f_i(x_k)$ – условная плотность распределения признака $x_k$ в $K_j$-ом классе, то информативность признака $x_k$ составит:

$$J_{x_k}=-\sum_{i=1}^{m}P(K_i)\log P(K_i)+\sum_{j=1}^{m}P(K_j)\sum_{i=1}^{m}\int_{G_i(x_k)}f_j(x_k)P(K_i\mid x_k)\log P(K_i\mid x_k)dx_k\ ,$$

где $G_i$ - область изменения признака $x_k$ в $K_i$-ом классе.

Аналогичные выражения могут быть получены и для признака $x_s$. При этом будем полагать, что информативность признака $x_k$ выше, чем информативность признака $x_s$, в случае, если информативность признака $x_k$, больше, чем информативность признака $x_s$, т. е. $J_{xk} > J_{xs}$. При этом имеем в виду следующее. Информативность признаков не является постоянной величиной и не представляет

собой безусловной величины, а наоборот, в общем случае количество информации, получаемое системой распознавания в результате измерения каждого данного признака, зависит от того, какие признаки были определены ранее и какие значения они приняли. Это в равной мере относится как к статистически зависимым, так и к статистически независимым признакам.

### 3.4.2 Отбор признаков по минимуму энтропии

Контроль с информационной точки зрения позволяет снять неопределенность состояния объекта, которая количественно характеризуется энтропией этого состояния. Пусть координаты $x_s$ (s = 1, 2, ..., k) вектора состояния X независимые величины. Тогда, используя известное свойство, заключающееся в том, что энтропия совокупности независимых величин равна сумме энтропии этих величин, можно записать

$$H_\Sigma = \sum_{s=1}^{k} H_s(x),$$

где $H_\Sigma$ - энтропия состояния диагностируемого изделия, $H_s(x)$-безусловная энтропия s-го признака. Количество информации $J_s(x_s, \Sigma)$, которую несет признак о состоянии изделия, можно оценить выражением

$J_s(x_s, \Sigma) = H_\Sigma - H_s(\Sigma / x_s)$,

где $H_s(\Sigma / x_s)$ - условная энтропия состояния изделия после контроля признака $x_s$.

Выбор признаков следует начинать с признака $x_s$, несущего максимальное количество информации $J_{smax}$. Энтропию по s-му признаку $H_s(x)$ можно вычислить с помощью следующего выражения

$$H_s(x) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

где $p_i$—вероятность попадания признака $x_s$ в i-й интервал диапазона его изменения.

Признаки можно выбирать и по критерию минимума величины H(x):

$$H_s(x) = \log_2 \sqrt{2\pi e D(x_s)}$$

где $D(x_s)$ - дисперсия распределения признака $x_s$, e=2.7182...- число Эйлера, $\pi$= 3.1415...

Таким образом, упорядочение признаков по степени информативности можно осуществлять по величине дисперсии распределения признака. Это можно объяснить иначе: чем меньше дисперсия признака, тем плотнее распределение и тем больше вероятность того, что изделия принадлежат к одному классу, который характеризуется определенной степенью работоспособности или сроком службы. И наоборот, чем больше D(x), тем менее однородной является партия изделий. Этот метод можно рекомендовать при вероятностном прогнозировании, когда вычисляются и анализируются величины дисперсий компонентов прогнозируемого процесса.

### 3.4.3 Диагностическая ценность обследования

**Частная диагностическая ценность обследования.** Условимся считать **диагностической ценностью** обследования по признаку $x_j$ для класса (диагноза) $K_i$ величину информации, носимую всеми реализациями признака $x_j$ в установление класса $K_i$.

Для m-разрядного признака

$$Z_{K_i}(x_j) = \sum_{s=1}^{m} P(x_{js}/K_i) Z_{K_i}(x_{js}),$$

где $x_{js}$- s-ый разряд j-го признака.

Диагностическая ценность обследования учитывает все возможные реализации признака и представляет собой математическое ожидание величины информации, вносимой отдельными реализациями. Так как величина $Z_{K_i}(x_j)$ относится только к одному классу $K_i$, то будем называть ее частной диагностической ценностью обследования по признаку $x_j$.

Следует также отметить, что $Z_{K_i}(x_j)$ определяет независимую диагностическую ценность обследования. Она характерна для случая, когда обследование проводится

первым или когда результаты других обследований неизвестны. Величина $Z_{K_i}(x_j)$ может быть записана в трех эквивалентных формах:

$$Z_{K_i}(x_j)=\sum_{s=1}^{m}P(x_{js}/K_i)\log_2\frac{P(x_{js}/K_i)}{P(x_{js})}$$

$$Z_{K_i}(x_j)=\sum_{s=1}^{m}P(x_{js}/K_i)\log_2\frac{P(K_i/x_{js})}{P(K_i)}$$

$$Z_{K_i}(x_j)=\sum_{s=1}^{m}P(x_{js}/K_i)\log_2\frac{P(K_i x_{js})}{P(K_i)P(x_{js})}.$$

**Общая диагностическая ценность обследования.** Известно, что обследование, обладающее небольшой диагностической ценностью для одного класса, может иметь начительную ценность для другого. Введем понятие общей диагностической ценности

обследования по признаку $x_j$ для всей системы классов K, определив ее как количество информации, вносимое обследованием в систему классов:

$$Z_K(x_j)=\sum_{i=1}^{q}P(K_i)Z_{K_i}(x_j)=\sum_{i=1}^{q}\sum_{s=1}^{m}P(K_i)P(x_{js}/K_i)\log_2\frac{P(x_{js}/K_i)}{P(x_{js})},$$

где q-количество классов (диагнозов).

Величина $Z_K(x_j)$ представляет собой ожидаемое (среднее) значение информации, которое может быть внесено обследованием в установление неизвестного заранее класса, принадлежащего рассматриваемой системе (совокупности) классов. В другой форме

$$Z_K(x_j)=\sum_{i=1}^{q}\sum_{s=1}^{m}P(K_i x_{is})\log_2\frac{P(K_i x_{js})}{P(K_i)P(x_{js})}.$$

**Общая диагностическая ценность одновременного обследования по комплексу признаков.** Для комплекса v признаков общая диагностическая ценность составит

$$Z_K(X)=Z_K(x_1x_2\ldots x_v)=\sum_{i=1}^{q}\sum_{s=1}^{m_1}\sum_{p=1}^{m_2}\ldots\sum_{y=1}^{m_v}P(K_i x_{1s}\ldots x_{vy})\log_2\frac{P(K_i x_{1s}x_{2p}\ldots x_{vy})}{P(K_i)P(x_{1s}x_{2p}\ldots x_{vy})},$$

где

v-количество признаков,

$m_v$-количество разрядов v-го признака.

Аналогичные соотношения справедливы и для частной диагностической ценности комплекса признаков:

$$Z_{K_i}(x_1 x_2 \ldots x_v) = \sum_{s=1}^{m_1}\sum_{p=1}^{m_2}\ldots\sum_{y=1}^{m_v} P(x_{1s} x_{2p} \ldots x_{vy} / K_i) \log_2 \frac{P(x_{1s} x_{2p} \ldots x_{vy} / K_i)}{P(x_{1s} x_{2p} \ldots x_{vy})}.$$

**Общая диагностическая ценность комплекса признаков при последовательном проведении обследования.** Если проведено обследование по комплексу признаков $X^{(n)}$ и требуется выбрать новый комплекс признаков для одновременного обследования $X^{(m)}$ с наибольшей диагностической ценностью, то следует исходить из величины:

$$Z_K(X^{(m)}/X^{(n)}) = Z_K(x_1 / X^{(n)}) + Z_K(x_2 / X^{(n)} x_1) + \ldots + Z_K(x_m / X^{(n)} x_1 x_2 .. x_{m-1}).$$

Достоинством информационого подхода является наличие строгого математического обоснования оценки информативности признаков. Недостатком – зависимость многих методов данного подхода от разрядности признаков (необходимость квантовать значения признаков на фиксированное количество дискретных разрядов).

## 3.5 Статистический подход

### 3.5.1 Выбор признаков методами статистической классификации. Метод весовых коэффициентов

Для оценки информативности (значимости) признаков можно ввести некоторое множество чисел, каждое из которых будет характеризовать «полезность» отдельного признака. Такие числа называются «весами», потому что они описывают как бы веса признаков в общей оценке работоспособности. Эти числа образуют поле положительных действительных чисел, и признаку, имеющему наибольшую информативность, должен быть приписан наибольший вес. Допустим, имеется множество N объектов. И пусть для простоты пояснения это множество состоит только из двух классов объектов $K_1$ и $K_2$. Каждый объект множества K описывается одним и тем же набором

n признаков: $x_1$, $x_2$, ..., $x_n$, значения которых в совокупности и определяют принадлежность объекта к своему классу. Если пространство признаков рассматривать как линейное метрическое, то набор признаков порождает две суммы, характеризующие близость объектов к классу $K_1$ и $K_2$:

$$S_1 = \sum_{s=1}^{n} a_s(\bar{x}_{1s} - x_s),\ S_2 = \sum_{s=1}^{n} a_s(\bar{x}_{2s} - x_s),$$

где $\bar{x}_{1s}$ и $\bar{x}_{2s}$ - средние значения s-го признака, определенные по совокупности экземпляров 1-го и 2-го классов соответственно;

$a_s$ - весовой коэффициент s-го признака.

Задача сводится к нахождению такой совокупности весовых коэффициентов, которая позволила бы уменьшить расстояние между объектами внутри одного класса и увеличить расстояние между объектами различных классов. Тогда принадлежность L-го объекта к одному из классов можно оценить с помощью разности $y = S_1 - S_2$, причем $x_L \in K_1$, если $S_1 < W$, $S_2 > W$; $x_L \in K_2$, если $S_1 > W$, $S_2 < W$, где W - некоторый порог, относительно которого оценивается принадлежность L-го объекта к соответствующему классу. Величина этого порога зависит от степени перекрытия классов, т. е. от множества $G=\{\Omega_L\}$ объектов $\Omega_L \in K_1 \underset{N}{\cap} K_2$, образующих пересечение классов. Таким образом, необходимо найти совокупность весовых коэффициентов, входящих в соотношение $y = s_1 - s_2$, таких, чтобы максимально уменьшить пересечение классов.

Пусть множество G есть множество объектов $\Omega_L$, принадлежащих пересечению классов $K_1$ и $K_2$. Каждый элемент множества можно характеризовать суммой

$$S_L = \sum_{s=1}^{n} E_{sL} a_s$$

Средняя величина таких сумм для объектов $x_L \in G$ равна

$$S = \frac{1}{m}\sum_{L=1}^{m} s_L = \frac{1}{m}\sum_{L=1}^{m}\sum_{s=1}^{n} E_{sL} a_s = \sum_{s=1}^{n} \bar{E}_s a_s\ ,$$

где $\bar{E}_s$ - среднее значение s-го признака по всем объектам множества G.

Для того чтобы можно было максимально точно разделить объекты множества

G, принадлежащие различным классам, необходимо в пространстве признаков найти направление максимальной дисперсии объектов. Очевидно, что в этом направлении должен быть расположен вектор, компонентами которого являются весовые коэффициенты $a_s$.

Тогда эти весовые коэффициенты будут порождать такие значения $s_L$, что множество G будет представлено в виде двух групп, за исключением объектов, для которых $s_L$ имеет значение, близкое или равное $\overline{S}$.

Составим выражение вида

$$y = \frac{\sum_{L=1}^{m}(s_L - \overline{S})^2}{\sum_{s=1}^{n} a_s^2}.$$

Сумма $\sum_{L=1}^{m}(s_L - \overline{S})^2$ в этом выражения будет тем больше, чем меньше сумма квадратов $a_s$, при этом будет иметь место наилучшее разделение объектов.

Выражение для у можно записать в матричной форме: у= аАа’/аБа', где а - вектор строки, а’ - вектор столбца, Б - единичная матрица, а А - матрица, пропорциональная выборочной ковариационной матрице, элементами которой являются числа

$$a_{sj} = \sum_{L=1}^{m}(E_{sL} - \overline{E}_s)(E_{jL} - \overline{E}_s).$$

Поскольку матрица А задана в области вещественных чисел и является симметричной, то все ее собственные значения являются действительными числами. Дифференцируя выражение для у по каждому $a_s$, получаем систему уравнений: |уБ - А|а’ = 0.

Для ненулевых $a_s$ система уравнений разрешима только в том случае, если ее определитель |уБ - А| = 0. Если раскрыть это выражение, то получим характеристический многочлен матрицы А, корни которого образуют спектр матрицы $y_1, y_2, ..., y_n$. Среди всех собственных чисел матрицы имеется максимальное $y_s$, собственный вектор, соответствующий этому числу, является тем вектором, компоненты которого образуют искомый набор весовых коэффициентов.

Собственный вектор будем искать методом итераций. Так как A - действительная симметричная матрица порядка n, то собственные значения этой матрицы есть действительные числа, а собственные векторы $a_1, a_2, ..., a_n$ образуют ортогональный базис. Тогда любой произвольный n-мерный вектор f можно единственным образом разложить в этом базисе: $f=a_1a^*_1+a_2a^*_2+ ... +a_na^*_n$, где $a^*_s$— собственные векторы матрицы A.

Умножая f на $A_m$, получаем $A_mf = a_1y_ma_1+...$ Если $y_1$ является наибольшим собственным числом матрицы A, а m достаточно велик, то членами, не содержащими $y_1$, можно пренебречь. Тогда среди слагаемых, обозначенных многоточием, содержатся лишь члены, имеющие сомножителями $y_{m2}$, $my_{(m-l)2}$, ..., $y_{m3}$ и т. д. В этом случае $A_mf/y_{m1}= =a_1a^*_1+...$, где невыписанные члены содержат $y_{m2}/y_{m1}$ ... и т. д.

Так как $|y_1| > |y_2|, |y_1|>|y_3|,$ то при $m \to \infty$

$$\lim_{m\to\infty} \frac{A_m f}{y_{m1}} = a_1 a_1 .$$

Обозначая L-й компонент вектора $A_mf$ в произвольном фиксированном базисе через $V_{mL}$, а L-й компонент вектора a в этом же базисе через $a_L$ и определяя пределы

$$\lim_{m\to\infty} \frac{a_1 V_{(m-1)L}}{y_{(m-1)l}} = a_1 a_L , \quad \lim_{m\to\infty} \frac{a_1 V_{mL}}{y_{m1}} = a_1 a_L ,$$

можно записать

$$\lim_{m\to\infty} \frac{V_{mL}}{V_{(m-1)L}} = y_1 .$$

Таким образом, числа $V_{m1}, V_{m2}, ..., V_{mn}$ при больших m пропорциональны числам $a_1, a_2, ..., a_k$, а вектор $A_mf$ приблизительно пропорционален собственному вектору, соответствующему собственному числу $y_1$.

Таким образом, **метод нахождения необходимого набора весовых коэффициентов** состоит в следующем. Берем произвольный вектор f и составляем последовательность $A_1f; A_2f; A_3f; ..., A_mf, ...$ Начиная с некоторого m строка $A_mf$ приблизительно пропорциональна строке $A_{m-1}f$.

Коэффициент пропорциональности есть собственное число $y_1$, а сам вектор $A_mf$— собственный вектор, компоненты которого и есть искомыйнабор весовых

коэффициентов.

Располагая коэффициенты в порядке убывания $|a_1| > |a_2| > \ldots >$, упорядочиваем выбранную систему признаков по значимости.

### 3.5.2 Статистическая оценка информативности и отбор признаков

При наличии достаточно большого объема статистических данных производится дополнительная обработка с целью количественной оценки информативности совокупности имеющихся начальных признаков.

Оценка информативности начальных признаков о потенциальной стабильности параметров-критериев годности (параметры-критерии не выходят за установленные пределы) производится с помощью аппарата **линейного регрессионно-корреляционного анализа** путем подсчета коэффициентов парной и / или множественной корреляции между экстремальными значениями параметра-критерия в процессе испытания (или значениями параметра-критерия в конце испытаний) и начальными признаками изделий.

Информативность признаков оценивается путем анализа полученных коэффициентов парной корреляции. Из начальных признаков выбираются те, которые имеют значимые коэффициенты парной корреляции с параметром-критериам. Затем из этих признаков отбираются те, которые имеют наименьшие коэффициенты парной корреляции между собой. По совокупности отобранных признаков подсчитываются коэффициенты линейной регрессии и коэффициент множественной корреляции данной совокупности начальных признаков с параметром–критерием.

Коэффициент множественной корреляции является мерой информативности данной совокупности.

При наличия нормального распределения начальних признаков в каждом классе их частная информативность оценивается по формуле:

$$J_j = \frac{\left(\bar{x}_j^{А} - \bar{x}_j^{Б}\right)^2}{y^2}, \text{ если } y = y_{x_j^{А}} = y_{x_j^{Б}};$$

$$J_j = 0.5\left(y_{x_j^А} - y_{x_j^Б}\right)^2\left(\frac{1}{\left(y_{x_j^Б}\right)^2} - \frac{1}{\left(y_{x_j^А}\right)^2}\right) +$$

$$+ 0.5\left(\frac{1}{\left(y_{x_j^А}\right)^2} + \frac{1}{\left(y_{x_j^Б}\right)^2}\right)\left(\overline{x}_j^А - \overline{x}_j^Б\right)^2, \text{ если } y_{x_j^А} \neq y_{x_j^Б},$$

где

$J_j$- частная мера информативности j-го признака;

$\overline{x}_j$ - среднее значение j-го признака;

$\sigma_{x_j}$ - среднеквадратическое отклонение j-го признака.

Индексы А и Б подразумевают вычисление соответствующей величины только для экземпляров соответствующего класса.

При тех же условиях информативность совокупности признаков в случае многомерного нормального распределения оценивается по формуле:

$$J = 0.5\mathrm{tr}\left[\left(V^А - V^Б\right)\left(V^{А^{-1}} - V^{Б^{-1}}\right)\right] + 0.5\mathrm{tr}\left[\left(V^{А^{-1}} + V^{Б^{-1}}\right)\left(\overline{X}^А - \overline{X}^Б\right)\left(\overline{X}^А - \overline{X}^Б\right)^T\right],$$

где

J – мера информативности совокупности признаков (чем больше, тем выше информативность);

tr – след матрицы (сумма диагональных элементов);

V-ковариационная матрица вида

$$\begin{pmatrix} V_{11} & V_{12} & \cdots & V_{1j} & \cdots & V_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ V_{k1} & V_{k2} & \cdots & V_{kj} & \cdots & V_{kn} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ V_{n1} & V_{n2} & \cdots & V_{nj} & \cdots & V_{nn} \end{pmatrix},$$

где $V_{kj}$ – общий элемент матрицы, определяется по формуле:

$$V_{kj} = M\left[\left(x_k - \overline{x}_k\right)\left(x_j - \overline{x}_j\right)\right],$$

где M-оператор математического ожидания,

$V^{-1}$ – матрица, обратная к V;

$\overline{X} = \left(\overline{x}_1, \overline{x}_2, ..., \overline{x}_j, ..., \overline{x}_n\right)$ - вектор средних значений признаков;

$^{T}$ – символ транспонирования.

Выбор наиболее информативной совокупности v признаков из числа имеющихоя n признаков ($1 \le v < n$) при небольшом n производится следующим образом: вычисляется информативность n совокупноотей из (n-1) признаков; выбирается совокупность с наибольшей информативностью и вычисляется информативность всех (n-1) входящих в нее совокупноотей из (n-2) признаков и т.д.

Таким образом, получаем оптимальные (в смысле информативности) совокупности из n-1, n-2, **...** 3,2,1 признаков.

Для распознавания из этих совокупностей выбирается совокупность минимальной размерности, удовлетворяющая требованию:

$J \ge J_{min}$, где $J_{min}$ - минимальное значение J, при котором общий процент правильного распознавания P превышает заданный уровень $P_{min}$. При этом для $P_{min}$ = 70% $J_{min} \sim 3$, для $P_{min}$ = 75% $J_{min} \sim 4$, для $P_{min}$ = 80% $J_{min} \sim 6$, для $P_{min}$ = 90% $J_{min} \sim 10$.

### 3.5.3 Метод случайного баланса

Задача отыскания наиболее существенных признаков, влияющих на качественные показатели изделия, может быть решена с помощью так называемых отсеивающих экспериментов.

Если число признаков велико и они коррелированны, то для выделения наиболее информативных признаков целесообразно использовать метод случайного баланса. Мерой информативности при этом служит степень связи признаков с выходным параметром y, которая оценивается по величине информативности признака:

$$b = \frac{m^{(+)} - m^{(-)}}{2},$$

где b – информативность (эффект) признака, $m^{(+)}$ - математическое ожидание y при условии, что данный признак находится на верхнем уровне диапазона своего

изменения, а остальные признаки изменяются случайным образом, $m^{(-)}$ - математическое ожидание y при условии, что данный признак находится на нижнем уровне диапазона своего изменения.

Поэтому весь диапазон изменения признаков $x_i$ делится на два уровня. Для этого рассчитываются оценка математического ожидания каждого признака $m_i$ и его выборочная дисперсия $S_i^2$:

$$m_i = \frac{1}{n}\sum_{j=1}^{N} x_{ij},$$

$$S_i^2 = \frac{1}{n-1}\sum_{j=1}^{N} (x_{ij}-m_i)^2,$$

где N – объем выборки приборов, j изменяется от 1 до N.

Верхний уровень составляют значения $x_i^{(+)}$, лежащие в диапазоне:

$$m_i + S_i \le x_i^{(+)} \le m_i + 3S_i$$

Нижний уровень $x_i^{(-)}$: $m_i - 3S_i \le x_i^{(-)} \le m_i + S_i$

Строится диаграмма рассеяния. На графике над отдельными признаками для верхнего и нижнего уровней наносятся значения y и определяются медианы выходного параметра для обоих уровней признака.

**Медиана** – это такое число X, что $x_i$ принимает с вероятностью 0.5 как значения, большие X, так и значения меньшие X.

Чем больше разность между медианами, тем сильнее связь соответствующего признака с выходным параметром y.

Выбирается наиболее информативный признак $x^1$.

Для расчета информативности признака определяют среднее значение y, для которого признак $x^1$ находится на верхнем уровне - $m^{(+)}$, и вычитают из него среднее значение y на нижнем уровне - $m^{(-)}$. Оценка информативности равна:

$$\delta^{(1)} = \frac{m^{(+)} - m^{(-)}}{2}.$$

Для нахождения других существенных признаков необходимо устранить влияние признака $x^1$ на y. С этой целью вычитают величину $2\delta^{(1)}$ из всех y, для которых $x^1$ находится на верхнем уровне.

Затем строят диаграмму рассеяния, в которой признак $x^1$ уже не оказывает влияния на y. Этот процесс осуществляется для всех признаков. Значимость рассчитанных коэффициентов проверяется с помощью критерия Стьюдента. Условие значимости имеет вид:

$$\left|\theta_i\right| \geq t_{1-\alpha} S_i ,$$

где

$$S_i^2 = \frac{(g_i^2)^{(+)} f_i^{(+)} + (g_i^2)^{(-)} f_i^{(-)}}{f_i^{(+)} + f_i^{(-)}},$$

$$(g_i^2)^{(+)} = \frac{1}{k-1} \sum_{i=1}^{k} (y_i - m_i^{(+)})^2 ,$$

где $(g_i^2)^{(+)}$ - дисперсия выходного параметра при условии, что признак $x_i$ находится на верхнем уровне, k – число значений выходного параметра при условии, что $x_i = x_i^{(+)}$, $m_i^{(+)}$ - среднее значение выходного параметра y при условии, что $x_i = x_i^{(+)}$, $f_i$ - число степеней свободы:

$$f_i = f_i^{(+)} + f_i^{(-)} ,$$

$$f_i^{(+)} = k - 1 .$$

Величина $(g_i^2)^{(-)}$ определяется аналогично $(g_i^2)^{(+)}$.

В оптимальную совокупность включаются признаки со значимыми коэффициентами.

### 3.5.4. Пошаговая регрессия

Еще одним методом отбора информативных признаков является **пошаговая регрессия** (прямая), когда независимые переменные одна за другой включаются в подмножество согласно предварительно заданному критерию. В то же время некоторая переменная может быть заменена другой переменной, не входящей в набор, или удалена из него. Совокупность критериев, определяющих, какие переменные включать, заменять и удалять, называется пошаговой процедурой

Предположим, что имеются набор независимых переменных $X_1,...,X_p$, которые являются кандидатами на роль предикторов Y, и случайная выборка объема N.

Стандартная пошаговая процедура состоит из правила включения переменных и правила исключения переменных (замена переменных не входит в состав стандартной пошаговой процедуры).

**Стандартная пошаговая процедура (F-метод)**. Включение и удаление переменных осуществляется с помощью статистики критерия, именно t-критерия для проверки равенства нулю частного коэффициента корреляции. В большинстве пакетов статистических программ вычисляется квадрат этой статистики, который имеет F-распределение и называется либо F-включения, либо F-удаления.

Более точно, предположим, что в набор *c* уже включено k переменных, k=0,1,..., p-1. Тогда значение F-включения для переменной X (не входящей в *c*) вычисляется по формуле:

$$F_{yx\cdot c} = \frac{r^2_{yx\cdot c} \cdot (N - k - 2)}{1 - r^2_{yx\cdot c}},$$

где $r^2_{yx\cdot c}$ – частный коэффициент корреляции между X и Y при фиксированных переменных из *c*. Его вычисление будет рассмотрено дальше.

Эта величина (F-включения) служит статистикой критерия для проверки гипотезы о том, что предсказание Y значимо не улучшается при включении X в набор *c*, если эта гипотеза верна, то распределена по закону F(1, N– k–2).

Аналогично, величина F-удаления для какой-либо переменной X из *c* служит статистикой критерия для проверки гипотезы о том, что набор *c*′, получающийся из *c* при удалении X и содержащий k′=k-1 переменных, предсказывает Y “так же хорошо”, как и набор *c*. Статистикой критерия является величина F-удаления, распределенная по закону F(1, N – k′ - 2 ),

$$F_{yx\cdot c'} = \frac{r^2_{yx\cdot c'}(N - k' - 2)}{1 - r^2_{yx\cdot c'}},$$

где $r^2_{yx\cdot c}$ – частный коэффициент корреляции между X и Y при фиксированных переменных из $c'$.

Правило остановки, обычно используемое в стандартной процедуре, основано на задании допустимого минимума F-включения (или, что эквивалентно, максимума уровня значимости $\alpha$). В некоторых пакетах по умолчанию предполагается, что

минимум F-включения равен 4.0. Для удаляемых переменных также выбирается допустимый минимум F-удаления (эта величина должна быть меньше минимума F-включения; в некоторых пакетах по умолчанию принимается, что минимум F-удаления равен 3.9). Рассмотрим теперь подробно шаги стандартной процедуры.

Шаг 0. Вычисляются простые коэффициенты корреляции $r_{yx_i}$ и величины F-включения $F_{yx_i}$ для i=1, ..., p. (Заметим, что простой коэффициент корреляции есть частный коэффициент корреляции при k=0 и пустом наборе *c*). Статистика критерия дается выражением:

$$F_{yx_i} = r^2_{yx_i} \frac{N-2}{1-r^2_{yx_i}},$$

$$r_{yx_i} = \frac{\sum_{j=1}^{N}(x_{ij}-\bar{x}_i)\cdot(y_j-\bar{y})}{\left[\sum_{j=1}^{N}(x_{ij}-\bar{x}_i)^2 \cdot \sum_{j=1}^{N}(y_j-\bar{y})^2\right]^{1/2}}.$$

Величина $F_{yx_i}$ имеет F-распределение с 1 и N-2 степенями свободы.

Шаг 1. Переменная $X_{i_1}$, которой отвечает наибольшее значение F-включения (или, что эквивалентно, наибольшая величина квадрата коэффициента корреляции с Y), выбирается как наилучшие предиктор для Y. Вычисляются соответствующее уравнение наименьших квадратов, таблица дисперсионного анализа (см. табл. 3.1) и множественный коэффициент корреляции $r_{y\cdot x_{i_1}} = \left| r_{yx_{i_1}} \right|$.

Таблица 3.1 – Таблица дисперсионного анализа для модели множественной линейной регрессии

| Источник дисперсии | Сумма квадратов | Степени свободы | Средний квадрат | F-отно-шение |
|---|---|---|---|---|
| Регрессия | $SS_D = \sum_{i=1}^{p} b_i \sum (x_{ij}-\bar{x}_i) y_i$ | $v_D = p$ | $MS_D = \frac{SS_D}{v_D}$ | $F = \frac{MS_D}{MS_R}$ |
| Отклонение от регрессии | $SS_R = SS_T - SS_D$ | $v_R = N-p-1$ | $MS_R = \frac{SS_R}{v_R}$ | |
| Полная | $SS_T = \sum_{i=1}^{N}(y_i-\bar{y})^2$ | $v_T = N-1$ | | |

Величина F-удаления для $X_{i_1}$ в этом случае совпадает с величиной F-включения. Далее вычисляются коэффициенты частной корреляции $r_{yx_i \cdot x_{i_1}}$ и значение F-включения:

$$r_{yx_i \cdot x_{i_1}} = 1 - \frac{1 - r^2_{y \cdot x_{i_1} x_{i_2}}}{1 - r^2_{y \cdot x_{i_1}}},$$

$$F_{yx_i \cdot x_{i_1}} = r^2_{yx_i \cdot x_{i_1}} \frac{N - 3}{1 - r^2_{yx_i \cdot x_{i_1}}},$$

где $r_{y \cdot x_{i_1} x_{i_2}}$, $r^2_{y \cdot x_{i_1}}$ - множественные коэффициенты корреляции, n – объем выборки; i=1, ..., p, $i \neq i_1$, т.е. для каждой переменной не вошедшей в уравнение регрессии.

Эта статистика имеет 1 и N–3 степеней свободы. Если все вычисленные значения F-включения меньше установленного минимума, то далее выполняется шаг S. В противном случае происходит переход на шаг 2.

Шаг 2. Переменная $X_{i_2}$, имеющая наибольшее значение F-включения (или, что эквивалентно, наибольший квадрат частного коэффициента корреляции с Y, при фиксированном значении $X_{i_1}$), выбирается как наилучший предиктор для Y при условии, что уже выбрана переменная $X_{i_1}$. Вычисляются уравнение наименьших квадратов, таблица дисперсионного анализа, множественный коэффициент корреляции $r_{y \cdot x_{i_1} x_{i_2}}$ и значения F-удаления $F_{yx_{i_1} \cdot x_{i_2}}$ и $F_{yx_{i_2} \cdot x_{i_1}}$. Эти статистики имеют 1 и N–3 степеней свободы и определяются выражениями

$$F_{yx_{i_1} \cdot x_{i_2}} = \frac{r^2_{yx_{i_1} \cdot x_{i_2}} \cdot (N - 3)}{1 - r^2_{yx_{i_1} \cdot x_{i_2}}},$$

$$F_{yx_{i_2} \cdot x_{i_1}} == \frac{r^2_{yx_{i_2} \cdot x_{i_1}} \cdot (N - 3)}{1 - r^2_{yx_{i_2} \cdot x_{i_1}}},$$

наконец вычисляются частный коэффициент корреляции $r_{yx_i \cdot x_{i_1} x_{i_2}}$ и значение F-включения $F_{yx_i \cdot x_{i_1} x_{i_2}} = \frac{r^2_{yx_i \cdot x_{i_1} x_{i_2}} \cdot (N - 4)}{1 - r^2_{yx_i \cdot x_{i_1} x_{i_2}}}$ с 1 и n – 4 степенями свободы при i=1, ..., p, $i \neq i_1$, $i \neq i_2$.

Если все значения F-включения меньше установленного минимума, то далее выполняется шаг S. В противном случае происходит переход на шаг 3.

Шаг 3. а) Пусть L обозначает набор из *l* независимых переменных, которые включены в уравнение регрессии. Если какое-либо из значений F-удаления для переменных из L меньше, чем соответствующий минимум, то переменная, которой соответствует наименьшее значение F-удаления, удаляется из набора и выполняется шаг 3, б) с заменой *l* на *l* – 1. Если для всех переменных, не входящих в L, значение F-включения меньше установленного минимума, то выполняется шаг S. В противном случае в набор L добавляется переменная, которой соответствует максимальное значение F-включения, и *l* заменяется на *l* + 1. б) Вычисляются уравнение наименьших квадратов, таблица дисперсионного анализа и множественный коэффициент корреляции $r_{y\cdot l}$ между Y и переменными из L, а также значения F-удаления $F_{yx_{i_j}\cdot(l-1)}$ между Y и переменной $X_{i_j}$ из L при заданных остальных переменных из L. Каждая из этих величин имеет 1 и n – *l* – 1 степеней свободы. Наконец, определяются величина частного коэффициента корреляции $r_{yx_i\cdot l}$ и значение F-включения между $F_{yx_i\cdot l}$ Y и каждой переменной $X_i$, не входящей в L, при данных переменных из L. Эта статистика имеет *1* и n – *l* – 2 степеней свободы.

Шаги 4, 5...Рекуррентно повторяется шаг 3. Шаг S выполняется а) если F-включения для всех переменных, не входящих в L, меньше установленного минимума, б) если для всех переменных из L значение F-удаления больше установленного минимума или в) число включенных переменных равно p.

Шаг S. Суммарная таблица печатается, как правило, по запросу пользователя. Для каждого шага печатается номер шага, номер включенных и удаленных переменных, значения F-включения и F-удаления и множественного коэффициента корреляции между Y и включенными переменными.

Стандартное правило, которое реализовано в большинстве программ пошаговой регрессии, осуществляет контроль числа переменных с помощью величины, называемой допустимый минимум F-включения. Как указывалось выше, величине минимума F-включения соответствует величина максимума уровня значимости $\alpha$, что в символьных обозначения выглядит так: min F-включения = $F_1$.

$_{\alpha}(1,\nu)$ для некоторого числа степеней свободы $\nu$. Обычно полагают $\nu = N - p - 1$, а рекомендуемое значение $\alpha$ составляет 0.15.

Стандартное правило остановки. Значения F-включения одно за другим сравниваются с величиной минимума F-включения. Набор H буде определен, когда все вычисляемые значения F-включения станут меньше заданного минимума. Рассмотрим этот процесс по шагам:

а) На шаге 1 включается переменная $x_{i_1}$. Если соответствующее значение F-включения незначимо, т.е. F-включения < min F-включения, то считается, что регрессия бессмысленна и пользователь должен обратиться к другим методам анализа своих данных. В противном случае $H=\{ x_{i_1} \}$.

б) На шаге 2 была добавлена переменная $x_{i_2}$. Если для нее F-включения < min F-включения, то H состоит только из переменной $x_{i_1}$ и наилучшая регрессия получена на шаге 1. В противном случае $H=\{ x_{i_1}, x_{i_2} \}$.

в) Для каждого дальнейшего шага при удалении переменной из H происходит переход на следующий шаг. С другой стороны, при включении некоторой переменной производится сравнение значения F-включения с порогом. Если величина F-включения значима, H расширяется добавлением этой переменной и происходит переход на следующий шаг. В противном случае происходит остановка процедуры, а наилучшим будет набор, полученный на предыдущем шаге.

Достоинством статистического подхода является учет статистической связи между признаками и выходным параметром, а также учет характеристик распределений данных величин.

Недостатком статистического подхода является относительная сложность реализации и длительность работы отдельных методов, связанные с необходимостью вычисления большого количества различных статистических характеристик.

## 3.6 Вероятностный подход

### 3.6.1 Сравнение апостериорных вероятностей

Пусть задан алфавит классов $K_i$, i = 1, ..., m, выбран априорный словарь признаков $x_j$, j=l, ..., n, известны условные плотности распределений $f_i(x_j)$ и априорные вероятности $P(K_i)$. Требуется произвести сравнительную оценку признаков $x_k$ и $x_s$, k, s=l, ..., n, иначе - определить, какой из этих признаков обладает лучшими разделительными свойствами. Разделим диапазон изменения признака $x_k$ на интервалы $\Delta_i^1(x_k); \Delta_i^2(x_k); ...; \Delta_i^m(x_k)$, на которых отличны от нуля соответственно одна функция $f_i(x_k)$; две функции $f_i(x_k)$, ... ; m функций $f_i(x_k)$. То же проделаем и с признаком $x_s$, т. е. определим интервалы $\Delta_i^1(x_s); \Delta_i^2(x_s); ...; \Delta_i^m(x_s)$.

Вероятность получить m-значное решение вида «класс 1, или класс 2, ..., или класс m» равна

$$P_m = \sum_{i=1}^{m} P(K_i) P\left[x_k \in \Delta_i^m(x_k) \mid i\right] = \sum_{i=1}^{m} P(K_i) \int_{\Delta_i^m(x_k)} f_i(x_k) dx_k ,$$

где $\Delta_i^m(x_k)$- совокупность интервалов, на которых отличны от нуля все m функций $f_i(x_k)$, i = 1, 2, ..., m.

Обозначив $M(z) = \sum_{z=1}^{m} z P_z$ - математическое ожидание случайной величины z, которая может принимать значения z=1, 2,... ..., m с вероятностями $P_z$, определим указанное математическое ожидание для первого и второго признаков, т. е. $M_{x_k}(z)$ и $M_{x_s}(z)$. Если $M_{x_k}(z) > M_{x_s}(z)$, то признак $x_s$ обладает лучшими разделительными свойствами; если $M_{x_s}(z) > M_{x_k}(z)$, то признак $x_k$ обладает лучшими разделительными свойствами. Будем полагать, что в первом случае выше информативность признака $x_s$, а во втором случае - признака $x_k$.

### 3.6.2 Сравнение вероятностных характеристик признаков

Сравнительная оценка информативности признаков может быть произведена и в случае, когда условные плотности распределений $f_i(x_j)$ неизвестны, однако известны первые и вторые моменты этих распределений, т. е. $m_{ji}$ и $D_{ji}$. Оценка, основанная на использовании этих данных, возможна в связи с тем, что признаки $x_j$ могут быть условно подразделены на две группы.

К первой группе относятся признаки, значения которых незначительно изменяются при переходе от одного объекта данного класса к другому объекту и весьма заметно изменяются при переходе от объекта одного класса к объектам других классов.

Ко второй группе относятся признаки, значения которых чувствительны к переходам от одного объекта данного класса к другому объекту и лишь незначительно изменяются при переходах от объектов одного класса к объектам других классов.

Признаки, относящиеся к первой группе, полезней признаков, относящихся ко второй группе. Количественная оценка информативности признаков $x_j$, j = 1, 2, ..., n, может быть произведена следующим образом.

Пусть некоторый механизм вырабатывает значения j-го признака с вероятностями, равными априорным вероятностям $P(K_i)$, i = l, ..., m. Определим математическое ожидание некоторой фиктивной случайной величины, принимающей значения $m_{ji}$ с вероятностями $P(K_i)$. т. е.

$$M[m_{ji}] = \sum_{i=1}^{m} m_{ji} P(K_i),$$

а также математическое ожидание дисперсии j-го признака по классам:

$$M[D_{ji}] = \sum_{i=1}^{m} D_{ji} P(K_i).$$

Если $M[D_{ki}]<M[D_{si}]$, k, s=1, ..., n, то при прочих равных условиях информативность признака $x_k$ выше, чем информативность признака $x_s$, так как вдоль оси признака $x_k$ объекты располагаются компактней, чем вдоль оси признака $x_s$.

Дисперсия математического ожидания распределений признаков при переходе от класса к классу $\overline{D}_{ji} = M\left\{\left[m_{ji} - M(m_{ji})\right]^2\right\}$. Если $\overline{D}_{ki} > \overline{D}_{si}$, то при прочих равных условиях информативность признака $x_s$ выше, чем информативность признака $x_k$, так как вдоль оси признака $x_k$ объекты, относящиеся к разным классам, располагаются на удалениях больших, чем вдоль оси признака $x_s$.

В качестве критерия сравнительной оценки признаков целесообразно использовать величину $Z_i=M[D_{ji}] \ / \ \overline{D}_{ji}$. Будем полагать, что если $Z_k<Z_s$, то информативность признака $x_k$ выше, чем информативность признака $x_s$, при этом наилучший признак тот, который реализует $\min\limits_j K_j = \min\limits_j \left\{M\left[D_{ji}\right] / \overline{D}_{ji}\right\}$.

Достоинством вероятностного подхода является учет вероятностной связи между признаками и выходным параметром. Недостатком рассмотренного подхода является необходимость знания априорных вероятностей классов и попадания значений признаков в определенные интервалы.

## 3.7 Нейросетевой подход

Весьма перспективными для решения задачи оценки информативности и отбора признаков являются нейронные сети (НС), которые способны извлекать знания из данных.

Наиболее известными и часто используемыми моделями НС для решения задач распознавания образов и количественной аппроксимации зависимостей являются перцептроны.

### 3.7.1 Оценка информативности признаков на основе однослойного перцептрона

Благодаря своим адаптивным способностям однослойный перцептрон может использоваться для аппроксимации линейных и квазилинейных функций, а также

для оценки информативности (значимости) признаков, если предполагается, что зависимость между признаками и выходом перцептрона близка к линейной.

Под количественной оценкой информативности (значимости) признаков будем понимать степень их влияния на изменение выходного сигнала перцептрона. Качественно будем различать информативные (значимые) признаки и неинформативные (малозначимые) признаки.

Из описания процесса работы перцептрона следует, что выходной сигнал y определяется уровнями входных сигналов $x_i$, их весовыми коэффициентами $w_i$, значением порога $\theta$, а также видом функции активации $\psi$. Так как функция активации $\psi$ и величина порога $\theta$ одинаковы для всех $x_i$, то при оценке значимости признаков $x_i$ их можно не рассматривать.

Значения входных сигналов $x_i$, в общем случае, могут находиться в интервале $(-\infty,+\infty)$, однако на практике значения сигналов $x_i$ предварительно нормируют, отображая их в интервал [0,1], что, как показывают результаты экспериментов, позволяет существенно ускорить процесс обучения НС и даже достигнуть большей точности. Далее будем полагать $\forall x_i \in [0,1]$.

Количественно информативность (значимость) признака $x_i$ (степень влияния $x_i$ на y) будем оценивать долей, которую $x_i$ вносит в y. Очевидно, эта доля $z(x_i^L)$ для i-го признака L-го экземпляра $x_i^L$ обучающей выборки определяется из выражения:

$$z\left(x_i^L\right)=\frac{|w_i x_i^L|}{\sum_{i=1}^{N}|w_i x_i^L|},$$

где N - количество признаков.

Тогда для всей обучающей выборки, в среднем, информативность (значимость) i-го признака $z(x_i)$ будет определяться выражением:

$$z(x_i)=\frac{\sum_{L=1}^{s}\frac{|w_i x_i^L|}{\sum_{i=1}^{N}|w_i x_i^L|}}{s},$$

где s - размер обучающей выборки, L-номер текущего экземпляра обучающей выборки.

Так как на практике для отбора признаков зачастую вполне достаточно иметь лишь приближенную относительную оценку информативности (значимости) признаков, рассмотрим возможность упрощения выражений для $z(x_i)$. Наложив на $x_i$ ограничение $\forall x_i \in [0,1]$ и приняв 1 в качестве максимально возможного значения признаков $x_i$, оценим информативность (значимость) i-го признака $z(x_i^*)$ для гипотетического экземпляра, у которого $\forall x_i = 1$:

$$z\left(x_i^*\right) = \frac{|w_i|}{\sum_{i=1}^{N} |w_i|}.$$

Такая оценка, очевидно, будет содержать определенную ошибку, но для многих практических задач она может оказаться вполне удовлетворительной для грубой оценки информативности (значимости) призаков в первом приближении.

Количественные оценки информативности (значимости) признаков $z(x_i^L)$, $z(x_i)$ и $z(x_i^*)$ будут лежать в интервале [0,1] и будут являться своего рода аналогами модуля коэффициента корреляции i-го признака $x_i$ и выхода перцептрона y.

Для качественной оценки i-го признака $x_i$, будем получать количественную оценку его значимости. Если она не будет превышать некоторый заранее заданный порог $\lambda$, то будем считать признак незначимым, в противном случае – значимым.

Определив качественные оценки признаков, в отношении каждого из них принимаем решение “исключить i-й признак” или “оставить (принять) i-й признак”.

Значение порога $\lambda$ следует подбирать экспериментально, но при этом необходимо учитывать следующие закономерности.

1) С увеличением значения $\lambda$ будет увеличиваться вероятность исключения значимых признаков и уменьшаться вероятность принятия незначимых признаков.

2) С уменьшением значения $\lambda$ будет увеличиваться вероятность принятия незначимых признаков и уменьшаться вероятность исключения значимых признаков.

### 3.7.2 Оценка информативности признаков на основе многослойного перцептрона

Как было отмечено выше, степень влияния i-го признака на выходной сигнал y для однослойного перцептрона определяется, в основном, значениями весов и признаков. Для МНС, имеющей один выход, то есть содержащей на последнем слое только один нейрон, степень влияния i-го признака $x_i$, значения которого поступают на входы нейронов первого слоя МНС, на выходной сигнал y также будут, в основном, определяться значениями весов и признаков МНС.

Количественная оценка информативности (значимости) i-го признака будет определяться как сумма информативностей i-ых входов нейронов первого слоя МНС. Для оценки информативности входов МНС необходимо оценить не только их значимость относительно нейрона, но и значимость выхода нейрона как входа нейронов следующего слоя и так далее до нейронов последнего слоя. Очевидно, что реализовать подобную процедуру проще в обратном порядке, двигаясь от последнего слоя к первому, поскольку вычисление значимости входов нейрона последнего слоя будет производиться аналогично однослойному перцептрону, а значимости входов нейронов предыдущего слоя можно будет определить как произведение частных значимостей входов относительно выходов и значимостей выходов, как входов нейронов следующего слоя.

Для реализации такого подхода будем использовать процедуру, подобную применяемой в широко известном алгоритме обратного распространения ошибки (Error Backpropagation Method), используемом при обучении МНС.

**Алгоритм обратной оценки значимости признаков на основе МНС**, разработанный авторами, применительно к оценке информативности признаков для L-го экземпляра обучающей выборки, будет иметь вид.

Шаг 1. Установить счетчик слоев q = M.

Шаг 2. Для всех j-ых нейронов q-го слоя МНС определить частные значимости их i-ых входов относительно их выходов ( эти значимости будем называть

частными, поскольку они не учитывают информативность выхода данного нейрона как входа нейронов следующего слоя):

$$z^*\left(x_i^{(q,j)[L]}\right) = \frac{|\, w_i^{(q,j)}\, x_i^{(q,j)[L]}\,|}{\sum_{d=1}^{N_q} |\, w_d^{(q,j)} x_d^{(q,j)[L]}\,|}$$

Шаг 3. Для всех нейронов q-го слоя определяют значимости их входов относительно выходов с учетом значимости выхода нейрона для нейронов следующего слоя:

$$z\left(x_i^{(q,j)[L]}\right) = z^*\left(x_i^{(q,j)[L]}\right)\sum_{p=1}^{N_{q+1}} z\left(x_j^{(q+1,p)[L]}\right).$$

Для нейронов М-го слоя $\sum_{p=1}^{N_{q+1}} z\left(x_j^{(q+1,p)[L]}\right)$ принимается равной 1.

Шаг 4. Если q>1, то установить q=q-1 и перейти на шаг 2, в противном случае – перейти на шаг 5.

Шаг 5. Для всех признаков $x_i$ i = 1,2,...,N определить оценки их значимости:

$$z\left(x_i^{[L]}\right) = \sum_{j=1}^{N_1} z\left(x_i^{(1,j)[L]}\right).$$

Аналогично однослойному перцептрону можно определить средние оценки значимости признаков для всей обучающей выборки. В этом случае алгоритм обратной оценки значимости признаков на основе МНС примет следующий вид.

Шаг 0. Установить счетчик экземпляров L=1.

Шаг 1. Установить счетчик слоев q = M.

Шаг 2. Для всех нейронов q-го слоя МНС определить частные значимости их входов относительно их выходов ( эти значимости будем называть частными, поскольку они не учитывают информативность выхода данного нейрона как входа нейронов следующего слоя):

$$z^*\left(x_i^{(q,j)[L]}\right) = \frac{|\, w_i^{(q,j)}\, x_i^{(q,j)[L]}\,|}{\sum_{d=1}^{N_q} |\, w_d^{(q,j)} x_d^{(q,j)[L]}\,|}$$

Шаг 3. Для всех нейронов q-го слоя определяют значимости их входов относительно выходов с учетом значимости выхода нейрона для нейронов следующего слоя:

$$z\left(x_i^{(q,j)[L]}\right) = z^*\left(x_i^{(q,j)[L]}\right)\sum_{p=1}^{N_{q+1}} z\left(x_j^{(q+1,p)[L]}\right).$$

Для нейронов М-го слоя $\sum_{p=1}^{N_{q+1}} z\left(x_j^{(q+1,p)[L]}\right)$ принимается равной 1.

Шаг 4. Если q>1, то установить q=q-1 и перейти на шаг 2, в противном случае – перейти на шаг 5.

Шаг 5. Для всех признаков $x_i$, i = 1,2,...,N определить оценки их значимости:

$$z\left(x_i^{[L]}\right) = \sum_{j=1}^{N_1} z\left(x_i^{(1,j)[L]}\right).$$

Шаг 6. Если L<s, где s – размер обучающей выборки, то установить L=L+1 и перейти на шаг 1.

Шаг 7. Определить средние оценки значимости признаков:

$$z\left(x_i\right) = \frac{1}{s}\sum_{L=1}^{s} z\left(x_i^{[L]}\right), \quad i = 1,2,...,N.$$

Однако, такая вычислительная процедура будет достаточно сложной, так как будет содержать 4 вложенных цикла (перебор экземпляров обучающей выборки, слоев МНС, нейронов слоев и входов нейронов) и будет достаточно медленной. Поэтому, на практике для ускорения вычислений, как и в случае однослойного перцептрона, можно ограничиться грубой оценкой значимости признаков в первом приближении, приняв все $x_i^{(q,j)}$ равными 1.

В этом случае алгоритм обратной оценки значимости признаков на основе МНС примет следующий вид.

Шаг 1. Установить счетчик слоев q = M.

Шаг 2. Для всех нейронов q-го слоя МНС определить частные значимости их входов относительно их выходов ( эти значимости будем называть частными, поскольку они не учитывают информативность выхода данного нейрона как входа нейронов следующего слоя):

$$z^*\left(x_i^{(q,j)*}\right) = \frac{|w_i^{(q,j)}|}{\sum_{d=1}^{N_q} |w_d^{(q,j)}|}$$

Шаг 3. Для всех нейронов q-го слоя определяют значимости их входов относительно выходов с учетом значимости выхода нейрона для нейронов следующего слоя:

$$z\left(x_i^{(q,j)*}\right) = z^*\left(x_i^{(q,j)*}\right)\sum_{p=1}^{N_{q+1}} z\left(x_j^{(q+1,p)*}\right).$$

Для нейронов М-го слоя $\sum_{p=1}^{N_{q+1}} z\left(x_j^{(q+1,p)*}\right)$ принимается равной 1.

Шаг 4. Если q>1, то установить q=q-1 и перейти на шаг 2, в противном случае – перейти на шаг 5.

Шаг 5. Для всех признаков $x_i$ i = 1,2,...,N определить оценки их значимости: $z\left(x_i^*\right) = \sum_{j=1}^{N_1} z\left(x_i^{(1,j)*}\right)$.

Оценки $z(x_i^*)$ в общем будут содержать определенную ошибку и нет гарантии, что эти оценки всегда будут близкими к $z(x_i)$. Однако, можно предположить, что на практике оценки $z(x_i^*)$ позволят правильно определять качественную оценку значимости признаков, чего вполне может быть достаточно для многих приложенией.

## 3.8 Когнитивный анализ и отбор информативных признаков

Для исследования возможности классификации объектов по набору признаков целесообразно применять методы когнитивной компьютерной графики, которые представляют собой совокупность различных способов представления трудно воспринимаемых человеком данных в легко воспринимаемых формах с целью оказания человеку помощи для извлечения знаний из данных. Среди этих способов особо следует отметить визуализацию числовых данных в графической форме.

Для визуализации больших наборов признаков одинаковой природы, характеризующих моделируемый объект, можно производить их отображение на графиках в различных координатных системах (декартовой, логарифмической, тангенциальной, гиперболической тангенциальной и др.), а также отображение на графиках в преобразованном виде: в виде графиков кумулятивных сумм, графиков быстрого дискретного смещенного преобразования Фурье, а также в виде графиков лапласианов и др. При этом на графиках следует отображать наборы признаков, характеризующие все экземпляры обучающей выборки, выделяя значения признаков экземпляров, принадлежащих к различным классам разными цветами. Затем среди всех построенных графиков необходимо отобрать те графики, на которых классы хорошо разделяются хотя бы на одном участке.

Помимо выяснения принципиальной возможности классификации объектов по конкретному набору признаков, когнитивную графику можно использовать для визуального выделения (отбора) наиболее информативных групп признаков. При этом информативным следует считать те области (группы признаков) на графике, где наблюдается хорошая разделимость классов.

Помимо выяснения принципиальной возможности классификации образов по конкретному набору признаков, когнитивные методы можно использовать для выделения (отбора) наиболее информативных признаков или их групп.

Достоинствами рассмотренного подхода являются его простота и наглядность, а недостатком – низкая точность и большая зависимость от человека. Поэтому необходимо определить процедуру автоматического выделения информативных

групп признаков, а также для каждой из выделенных групп признаков подобрать свертку, наилучшим образом извлекающую наиболее ценную информацию.

Для решения данной задачи предлагается использовать **эвристические алгоритмы выделения информативных групп признаков и подбора свертки.**

Пусть мы имеем обучающую выборку экземпляров, характеризующихся наборами однородных признаков, и среди этих признаков выделены группы (для лопаток из вышерассмотренной задачи – участки спектра), где обеспечивается хорошее разделение классов. Тогда, если мы заранее определим некоторый набор функций-сверток и метод классификации, мы сможем для каждого участка подобрать свою свертку на основе **эвристического алгоритма итеративного подбора свертки** для заданной группы признаков, который имеет следующий вид.

Шаг 1. Инициализация. Задается набор сверток $s=\{s_1,s_2,....,s_q\}$, где q-количество заданных сверток. Вычисляются значения всех сверток для заданной группы признаков всех экземпляров обучающей выборки.

Шаг 2. Для экземпляров из обучающей и/или контрольной выборки производится одномерная классификация по текущей группе признаков (для лопаток – по текущему участку спектра), при этом в качестве признака последовательно используются свертки $s_k$, k=1,2,...,q.

Шаг 3. Вычисляются риски потребителя, производителя и общая ошибка классификации для каждой свертки. Та свертка, которая наилучшим образом удовлетворит целевую функцию, принимается в качестве базовой.

Алгоритм итеративного подбора сверток требует, чтобы информативные группы признаков были заранее определены, что требует участия человека. Для устранения этого недостатка мы предлагаем использовать **алгоритм сжатия набора признаков**, который позволяет автоматически выделять информативные группы признаков и подбирать для них свертки.

Шаг 1. Инициализация. Задать набор сверток $s=\{s_1,s_2,....,s_q\}$, где q-количество заданных сверток, и функцию оценки информативности признаков $f(x_i)$, где $x_i$ – i-ый признак, i=1,2,...,N, N – количество текущих признаков. В качестве набора текущих признаков принять исходные признаки. Задать значение порога значимости (информативности) признаков P и значение коэффициента редукции $б$, $(0 < б < 1)$.

Шаг 2. Для всех текущих признаков оценить информативности $f(x_i)$, i=1,2,...,N.

Шаг 3. Если оценки информативности всех текущих признаков меньше порога значимости $f(x_i)<P$, i=1,2,...,N, то перейти на шаг 4, иначе перейти на шаг 6.

Шаг 4. Установить $N = \text{round}(\text{б}N)$, где round – функция округления. Разбить набор текущих признаков на N групп и для каждого такого интервала вычислить значения всех сверток, произвести одномерную классификацию по каждой из сверток и закрепить за интервалом ту свертку, которая будет лучше всего удовлетворять целевой функции (например, минимуму ошибки). Перейти на шаг 2.

Шаг 5. Исключить из набора текущих признаков те признаки $x_j$, для которых $f(x_j)<P$. Уменьшить N на число удаленных признаков.

Шаг 6. Конец.

Заметим, что в качестве функции оценки информативности признаков в простейшем случае можно использовать отношение:

$$f(x_i) = \frac{|r_{x_i,y}|}{\sum_{i=1}^{N} |r_{x_i,y}|},$$

где $|r_{x_i,y}|$ - коэффициент корреляции i-го признака и номера класса y.

### 3.9 Комбинированная оценка информативности признаков

Подходы к отбору информативных признаков, рассмотренные в данной главе могут быть проанализированы на основе ряда критериев. Сравнительная характеристика рассмотренных подходов приведенна в табл. 3.2.

Таблица 3.2 – Сравнительная характеристика подходов к отбору информативных признаков

| Критерии сравнения | Подходы к отбору признаков | | | | |
|---|---|---|---|---|---|
| | эвристи-ческий | информа-ционный | статисти-ческий | вероят-ностный | нейро-сетевой |
| математическое обоснование | низкое | высокое | среднее | среднее | низкое |
| относительная сложность реализации | низкая | средняя | высокая | низкая | высокая |
| относительная скорость работы процедур | низкая | высокая | низкая | высокая | средняя |
| требования к данным | низкие | высокие | низкие | низкие | средние |

Как видно из табл. 5.6, ни один из рассмотренных подходов не обладает абсолютным преимуществом перед всеми остальными. Поэтому на практике следует использовать тот подход, который лучше соответствует наиболее важному требованию (критерию), предъявляемому к решаемой задаче.

### 3.10 Разбиение исходной выборки на обучающую и тестовую

При построении моделей сложных объектов и процессов по точечным данным на основе теории распознавания образов, нейронных сетей и др. методов возникает задача разбиения исходной выборки данных на обучающую и контрольную.

Традиционно **обучающую** и **контрольную выборки** выделяют с помощью случайных чисел или линейным разбиением исходной выборки. Однако при этом в контрольную выборку могут попасть такие экземпляры, которые сильно удалены в метрическом пространстве признаков от экземпляров, попавших в обучающую

выборку. В этом случае модель, построенная по обучающей выборке, будет плохо работать даже для контрольной выборки, не говоря уже об адекватности этой модели исследуемому объекту.

Поэтому возникает задача разработки алгоритма, позволяющего разбивать исходную выборку таким образом, чтобы обучающая выборка содержала все экземпляры, находящиеся в узловых точках, представленные в исходной выборке, а контрольная выборка содержала экземпляры близкие (в смысле расстояния) к соответствующим экземплярам обучающей выборки.

С другой стороны, при обучении нейронных сетей одной из важнейших задач является сокращение времени обучения сети при обеспечениии заданного уровня точности прогнозирования, что может быть достигнуто за счет сокращения обучающей выборки путем удаления из нее экземпляров, не находящихся в узловых точках (т.е. избыточных примеров).

Для решения обеих вышеописанных задач предлагается использовать следующий **алгоритм разбиения исходной выборки на обучающую и тестовую**.

Шаг 1. Инициализация параметров алгоритма разбиения исходной выборки. Задать исходную выборку экземпляров $x_{исх.}$ и сопоставленные им номера классов или значения прогнозируемого параметра $t_{исх.}$, а также L - количество разбиений исходной выборки. Занести в переменную N количество признаков, характеризующих экземпляры, а в переменную M – количество экземпляров исходной выборки. Для задач классификации принять ширину допустимого интервала вариации прогнозируемого параметра dt=0, для задач численной оценки прогнозируемого параметра принять dt = | max($t_{исх.}$) - min($t_{исх.}$) | / L, где min(a) и max(a) – минимальное и максимальное значения вектора a, соответсвенно. Установить счетчик newind=1.

Шаг 2. Вычислить расстояния между экземплярами исходной выборки.

$$R(p,q)=\begin{cases}\sqrt{\sum_{i=1}^{N}\left(x_i^p-x_i^q\right)^2}, p\neq q; \\ RealMax, p=q,\end{cases} \qquad p=1,...,M,\ q=1,..,M$$

где RealMax - максимально представимое в ЭВМ число, $x^p_i$- значение i-го признака p-го экземпляра.

Шаг 3. Найти в матрице расстояний R минимальный элемент minx и его индексы q и p, а также максимальный элемент maxx, при условии, что при нахождении минимума и максимума здесь и далее игнорируются элементы равные RealMax.

Шаг 4. Принять a=| maxx-minx | / (2L).

Шаг 5. Если minx<Realmax, то перейти на шаг 6, в противном случае – перейти на шаг 13.

Шаг 6 Принять: $x_{об.}(newind) = x_{исх.}(q)$, $t_{об.}(newind) = t_{исх.}(q)$, где $x_{об.}$ и $t_{об.}$ – массивы экземпляров обучающей выборки и сопоставленных им прогнозируемых значений, соответственно. Установить: newind=newind+1, значение текущего минимального элемента в строке teck=R(q,p). Найти минимальный элемент mminx и его индексы mqmin и mpmin среди элементов q-ой строки матрицы R.

Шаг 7. Если mminx<Realmax, то перейти на шаг 8, в противном случае – перейти на шаг 11.

Шаг 8. Установить значение указателя удаленного экземпляра из столбца deleted = 0 (в матрице R нумерация строк и столбцов должна начинаться с 1).

Шаг 9. Если | teck - R(q,mpmin) | ≤ a, то перейти на шаг 10, иначе принять: deleted = mpmin, R(q,mpmin) = Realmax, R(mpmin,q) = Realmax и перейти на шаг 11.

Шаг 10. Если $|(t_{исх.}(mpmin)-t_{исх.}(q)| \le dt$, то принять: R(v, mpmin) = realmax, R(mpmin, v) = realmax, v=1,..,M, в противном случае – принять: $x_{об.}(newind) = x_{исх.}(mpmin)$, $t_{об.}(newind) = t_{исх.}(mpmin)$, newind = newind+1, R(v, mpmin) = Realmax, R(mpmin, v) = Realmax, v=1,..,M.

Шаг 11. Найти минимальный элемент mminx и его индексы mqmin и mpmin среди элементов q-ой строки матрицы R.

Шаг 12. Перейти на шаг 7.

Шаг 13. Принять: R(v, q) = Realmax, R(q, v) = Realmax, v = 1,..,M, указатель удаленного экземпляра из строки dstr = q. Найти в матрице расстояний R минимальный элемент minx и его индексы q и p.

Шаг 14. Перейти на шаг 5.

Шаг 15. Если (deleted $\neq$ dstr) и (deleted>0), тогда принять: $x_{об.}$(newind) = $x_{исх.}$(deleted). $t_{об.}$(newind) = $t_{исх.}$(deleted).

Шаг 16. Останов.

В результате выполнения данного алгоритма для исходной выборки $x_{исх.}$ и сопоставленного ей набора значений $t_{исх.}$ мы получим обучающую выборку $x_{об.}$ и сопоставленный ей набор значений $t_{об.}$ Остаток экземпляров из $x_{исх.}$ и $t_{исх.}$, не вошедших в $x_{об.}$ и $t_{об.}$ составит контрольную выборку.

# ГЛАВА 4. МЕТОДЫ И АЛГОРИТМЫ ПОСТРОЕНИЯ ДИАГНОСТИЧЕСКИХ МОДЕЛЕЙ

## 4.1 Основные понятия теории распознавания образов

**Распознавание образов** - процесс, при котором на основании многочисленных характеристик (признаков) некоторого объекта определяется одна или несколько наиболее существенных, но недоступных для непосредственного определения, его характеристик, в частности, его принадлежность к определенному классу объектов. Решить задачу распознавания - значит найти на основании косвенных данных правила, по которым каждому набору значений признаков некоторого объекта ставится в соответствие одно из заданного множества возможных решений, определяющих существенные характеристики этого объекта.

В каждой задаче распознавания исходными данными являются результаты некоторых наблюдений или непосредственных измерений. Их называют **первичными признаками**, а совокупность всех первичных признаков - **входным сигналом**.

Результатом единичного **акта распознавания** является **решение**, а результатом решения задачи распознавания - **решающее правило** (или **алгоритм принятия решения**, или **решающая функция**), которое определяет отображение множества сигналов на множество решений, т. е. для каждого сигнала указывает определенное решение. Если множество решений дискретно и число различных решений невелико, то распознавание можно рассматривать как **классификацию**. Решающая функция в этом случае делит множество сигналов на подмножества, называемые **классами**, так что каждому классу соответствует одно определенное решение. В тех случаях, когда множество сигналов является топологическим пространством, т. е., когда целесообразно говорить о близости двух сигналов, **границы классов** называют **разделяющими поверхностями** (в частности, это могут быть гиперплоскости).

В большинстве случаев существует некоторая объективная классификация

сигналов, которая, в принципе, может быть известна, если доступны некоторые дополнительные (по отношению к входному сигналу) сведения. Однако возможны случаи, когда такая объективная классификация не существует. Объективную классификацию можно описать с помощью некоторого дискретного параметра, называемого искомым параметром. Тогда сигнал следует считать зависящим от искомого параметра, В общем случае может быть несколько искомых параметров, и они могут быть непрерывными.

Методы распознавания образов могут найти применение для решения следующих практических задач: 1) распознавание букв и цифр с целью ввода данных в ЭВМ; 2) распознавание слов устной речи с целью ввода данных в ЭВМ или управления автоматами; 3) диагностика болезней, где непрерывное множество решений представляет собой множество способов лечения; 4) диагностика неисправностей машин и отдельных их деталей; 5) обработка данных геологической разведки, при которой решения принимаются относительно наличия определенных ископаемых; 6) обработка радиолокационных сигналов с принятием решений относительно наличия определенных обнаруживаемых объектов, а также относительно значений параметров, характеризующих эти объекты; 7) автоматическая классификация живых клеток, напр., кровяных телец, наблюдаемых под микроскопом; 8) обработка фотографий следов частиц в физических экспериментах с целью определения параметров частиц и отбора снимков, содержащих интересующие физика события; 9) распознавание фраз или слов в тексте, написанном на формальном или естественном языке; 10) распознавание алгебр, выражений определенных типов при выполнении формальных преобразований над формулами с помощью ЭВМ.

Эти задачи существенно отличаются по своей природе. В первых двух необходимо найти такой способ классификации входных сигналов, который как можно точнее соответствовал бы классификации, осуществляемой человеком. Это обусловлено тем, что различные варианты написания букв и произнесения слов приспособлены к человеческому восприятию.

В задачах 3) — 8) существуют некие объективно правильные решения,

которые, в принципе. можно узнать, располагая дополнительными (по отношению к входному сигналу) данными. В этих случаях решающая функция должна как можно точнее воспроизводить эти правильные решения. В задаче 10) предполагается известным формальное определение класса алгебраических. выражений и задача распознавания заключается в преобразовании такого определения в правило принятия решения о принадлежности к классу. Такое преобразование иногда трудно осуществить.

Среди перечисленных выше задач распознавания только задача 10) и, иногда, 9) имеет с самого начала формальную математическую постановку. Однако и многие из остальных задач допускают формальную постановку. Она базируется на более или менее обоснованных гипотезах о процессах, определяющих зависимость первичных признаков от тех величин или параметров, относительно значений которых необходимо принимать решения. Эти гипотезы могут относиться к свойствам различных подмножеств или к свойствам решающих функций, или к характеру процессов, порождающих наблюдаемые сигналы.

Различают четыре типа задач, относящихся к проблеме распознавания образов и отличающихся постановками. Ниже приводятся несколько упрощенные постановки этих задач.

**а) Задача классификации.** Дано распределение вероятностей сигнала, зависящее от некоторого дискретного параметра, называемого искомым, или некоторые условия, тоже зависящие от параметра, которым должен удовлетворять сигнал. Указан некоторый критерий, называемый **риском распознавания**, характеризующий качество решающей функции для различных значений параметра (в среднем или для «наихудшего» значения параметра). Можно сказать, что критерий характеризует степень соответствия получаемых решений истинным значением параметра, т. о. «правильность» решений. Требуется найти наилучшую (в смысле этого критерия) решающую функцию. В случае, когда дано распределение вероятностей, распознавание сводится к одной из задач теории статистических решений. Случай, когда заданы условия, определяющие непересекающиеся подмножества значений сигнала для каждого значения искомого параметра, на

первый взгляд представляется тривиальным, поскольку решение содержится в условиях задачи. Однако это далеко не всегда так, потому что условия, совершенно точно определяющие подмножества, иногда очень трудно непосредственно проверить. В таких случаях необходимо найти эффективный способ проверки условии.

**б) Задача описания.** Дано множество некоторых элементарных сигналов и правила составления сложного сигнала из элементарных (**правила синтеза**). Требуется найти **правила анализа**, т. е. правила, по которым, имея реализацию сложного сигнала, можно найти те элементарные сигналы, из которых он составлен, а также указать использованные при его составлении правила синтеза. Например, изображение буквы можно рассматривать как сложное изображение, составленное из таких элементарных частей, как отрезки прямых линий и дуг окружностей. Правила синтеза определяют выбор нужных отрезков и порядок их соединения между собой. Описание данного изображения буквы состоит в перечислении входящих в ее состав отрезков и в указании их взаимного расположения. Задача описания усложняется, если определенные правила синтеза можно указать лишь для некоторых идеализированных сигналов, называемых **эталонами**, а наблюдаемые сигналы отличаются от эталонов наличием случайных помех. В этом случае либо должны быть известны статистические свойства помех, либо должны быть приняты определенные допущения об этих свойствах. Решить задачу описания в этом случае означает указать правила нахождения такого эталона, который составлен по заданным правилам синтеза и одновременно является при данном сигнале наиболее правдоподобным, т. е. в определенном смысле наиболее близким к данному сигналу.

**в) Задача обучения** возникает в тех случаях, когда в условии одной из задач типа а) при б) присутствует, кроме искомого параметра, некоторый другой неизвестный параметр, т. е. постоянный параметр, о котором известно только, что он сохраняет постоянное значение. Т. о., распределение вероятностей. или условия, задающие подмножества сигналов. или множество допустимых эталонов определены не полностью. Дана также обучающая выборка, представляющая собой последовательность наблюдавшихся в этих условиях сигналов, для каждого из

которых указано правильное решение. Требуется построить решающую функцию. В случае обучения условия задачи определяют не единственную решающую функцию, а целое семейство таких функций.

С помощью обучающей выборки и заданного критерия качества распознавания (риска) можно выбрать наилучшую в смысле этого критерия решающую функцию из семейства.

**г) Задача самообучения**. Постановка этой задачи подобна предыдущей и отличается только тем, что обучающая выборка содержит лишь последовательность сигналов без указания правильных решений.

В процессе распознавания образов одной из основных задач является **задача классификации**, т. е. разделения множества исходных данных на однородные в некотором смысле подмножества. Критерии такого разделения далеко не всегда могут быть точно и непосредственно формализованы.

Рассмотрим пространство признаков, которые выбраны адекватно поставленной задаче. Первый этап решения состоит в том, чтобы отобразить в этом пространстве «облако» точек (или, может быть, единственную точку), связанных с одним и тем же классом, а затем определить один или несколько прототипов, представляющих будущие классы. Эти представители классов вовсе не обязательно должны совпадать с какими-то конкретными реализациями, по которым получены результаты экспериментальных измерений или наблюдений. Скорее наоборот, имея некоторые первоначальные сведения о прототипах, можно управлять процессом измерений или иным методом получения исходных данных. Обычно прототипы называют именами представляемых ими классов. На последующих этапах они используются для распознавания неизвестного образа, исходные данные о котором получены таким же методом, как и данные об уже известных образах. Основу рассматриваемого здесь способа классификации составляет **процесс обучения**, в задачу которого входит постепенное усовершенствование алгоритма разделения предъявляемых объектов на классы. Этот процесс обычно стремятся, по возможности, автоматизировать. С этой целью отбирают часть предъявляемых объектов и используют их, в процессе обучения для «тренировки» системы. Массив

исходных данных в обучаемой системе состоит из двух частей: **обучающей выборки** и **тестовой выборки**, используемой в процессе испытаний (или **экзамена**). При этом, естественно, возникает вопрос о представительности выбранного множества.

Если совокупность классов известна заранее, то обучение называют **контролируемым** (**«обучение с учителем»**). Роль разработчика заключается в выработке наилучших критериев классификации, учитывающих различия между признаками, характерными для отдельных классов. Главной задачей становится в этом случае поиск оптимальных методов разделения.

Если классы, составляющие обучающую выборку, не известны заранее, до начала процедуры классификации, то обучение называют **неконтролируемым,** или «**обучением без учителя**». Решение задачи при таких условиях значительно более сложно, чем в предыдущем случае.

**а) Обучение с учителем.** Главная особенность контролируемого метода классификации заключается в непременном наличии «априорных» сведений о принадлежности к определенному классу каждого вектора измерений, входящего в обучающую выборку. Например, если с большого расстояния требуется отличить поле, засеянное зерновыми, от леса или пустыни, то заранее проводят измерения известных участков земли, на которых есть поля, леса, пустыни и т. д. Таким образом, получают множество векторов измерений от источников, принадлежность которых к определенному классу заранее известна.

Роль обучающего состоит в том, чтобы создать такую систему, которая позволила бы каждый вектор измерений, источник которого неизвестен, отнести к одному из уже известных классов.

Задача заключается в уточнении и оптимизации процедуры принятия решений, которая часто строится интерактивным методом с помощью обучающей выборки. В основу процедуры положено понятие расстояния от рассматриваемой точки до границы, отделяющей пространство, характеризуемое определенными признаками. В зависимости от размерности пространства, от его формы граница может описываться по-разному: линия, гиперплоскость, потенциальный рельеф и т. д.

Одним из основных путей оптимизации служит метод градиентного спуска в пространстве параметров. Системы такого рода обладают всеми достоинствами автоматических систем: высокой скоростью выполнения операций, точностью, надежностью, постоянством характеристик и т. д.

**б) Обучение без учителя.** В процессе неконтролируемого обучения (без учителя) автоматическое устройство самостоятельно устанавливает классы, на которые подразделяется исходное множество, и одновременно определяет присущие им признаки.

Роль разработчика заключается лишь в том, чтобы сообщить машине критерии, в соответствии с которыми должно выполняться разделение на классы.

Неконтролируемое обучение представляет собою значительно более сложную операцию, чем контролируемое. Действительно, здесь в большинстве случаев не известны ни число классов, ни характеристики каждого из них. Поэтому процесс организуют так, чтобы среди всех возможных вариантов группирования найти такой, при котором группы обладали бы наибольшей компактностью. Структурная схема алгоритма, изображенная на рис. 4.1, иллюстрирует метод последовательных приближений к искомому решению.

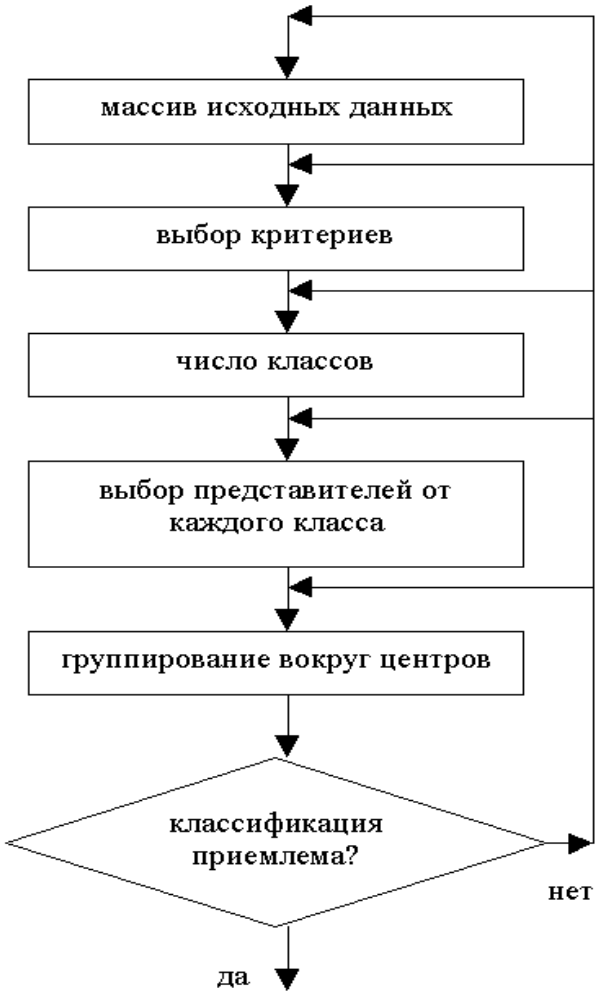



Рис. 4.1 - Схема процесса обучения без учителя.

Если полученная на каком-то этапе классификация по каким-либо причинам неудовлетворительна, то следует вернуться к одному из предыдущих этапов с учетом имеющейся, но не использованной ранее информации. Каждое возвращение на очередной виток цикла характеризуется определенной ценой, в качестве которой можно принять расстояние, отделяющее конечный этап от того, с которого начинается новый цикл. Но необходимо иметь в виду, что такая процедура не обязательно будет сходящейся.

**Классификация систем распознавания** основывается на использовании в качестве классификационного принципа свойства информации, используемой в процессе распознавания.

Возможная классификация систем распознавания показана на рис 4.2.

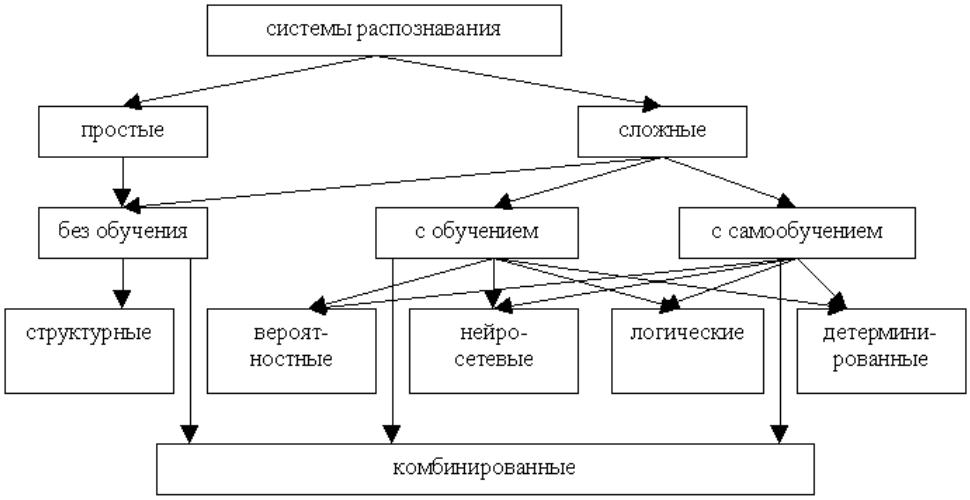



Рис. 4.2 - Классификация систем распознавания.

Системы распознавания можно подразделить на простые и сложные в зависимости от того, физически однородная или физически неоднородная информация используется для описания распознаваемых объектов, имеют ли признаки, на языке которых произведено описание алфавита классов, единую или различную физическую природу.

**а) Простые системы распознавания.** К ним относят, например, читающие автоматические распознающие устройства, в которых признаки **рабочего словаря** представляют собой лишь те или иные линейные размеры распознаваемых объектов; автоматы для размена монет, где в качестве признака, используемого при распознавании монет, берется их масса; автоматические устройства, предназначенные для отбраковки деталей, в которых в качестве признаков, применяемых для описания классов бракованных и небракованных деталей, используются либо некоторые линейные размеры, либо масса и т. д.

**б) Сложные системы распознавания.** К ним относят, например, системы

медицинской диагностики, в которых в качестве признаков (симптомов) могут использоваться данные анализа крови и кардиограмма, температура и динамика кровяного давления и т. п.; системы, предназначенные для распознавания образцов геологической разведки, в которых в качестве признаков берутся различные физические и химические свойства, или образцов военной техники вероятного противника и т. д.

Если в качестве принципа классификации использовать способ получения апостериорной информации, то сложные системы можно подразделить на одноуровневые и многоуровневые.

**а) Одноуровневые сложные системы.** В этих системах апостериорная информация о признаках распознаваемых объектов определяется прямыми измерениями непосредственно на основе обработки результатов экспериментов.

**б) Многоуровневые сложные системы.** В этих системах апостериорная информация о признаках распознаваемых объектов определяется на основе косвенных измерений.

Если в качестве принципа классификайии избрать количество первоначальной априорной информации о распознаваемых объектах, то системы распознавания можно подразделить на системы без обучения, обучающиеся и самообучающиеся.

**а) Системы без обучения.** В этих системах первоначальной априорной информации достаточно для того, чтобы определить априорный алфавит классов, построить априорный словарь признаков и на основе непосредственной обработки исходных данных произвести описание каждого класса на языке этих признаков, т. е. в первом приближении достаточно определить решающие границы, решающие правила. Будем считать, что для построения этого класса систем необходимо располагать полной первоначальной априорной информацией.

**б) Обучающиеся системы.** В этих системах первоначальной априорной информации достаточно для того, чтобы определить априорный алфавит классов и построить априорный словарь признаков, но недостаточно (либо ее по тем или другим соображениям нецелесообразно использовать) для описания классов на языке признаков. Исходная информация, необходимая для построения обучающихся

систем распознавания, позволяет выделить конкретные объекты, принадлежащие различным классам. Эти объекты представляют собой обучающие объекты (обучающая последовательность, обучающая выборка). Цель процедуры обучения - определение разделяющих функций путем многократного предъявления системе распознавания различных объектов с указанием классов к которым эти объекты принадлежат. Системы распознавания, обучающиеся на стадии формирования, работают с «учителем». Эта работа заключается в том, что «учитель» многократно предъявляет системе обучающие объекты всех выделенных классов и указывает, к каким классам они при надлежат. Затем «учитель» начинает «экзаменовать» систем распознавания, корректируя ее ответы до тех пор, пока количество ошибок в среднем не достигнет желаемого уровня.

**в) Самообучающиеся системы.** В этих системах первоначальной априорной информации достаточно лишь для определения словаря признаков, но недостаточно для проведения классификации объектов. На стадии формирования системы ей предъявляют исходную совокупность объектов, заданных значениями своих признаков, однако из-за ограниченного объема первоначальной информации система при этом не получает указаний о том, к какому классу объекты исходной совокупности принадлежат. Эти указания заменяются набором правил, в соответствии с которыми на стадии самообучения система распознавания сама вырабатывает классификацию, которая может отличаться от естественной.

Если в качестве принципа классификации использовать характер информации о признаках распознаваемых объектов, которые подразделили на детерминированные, вероятностные, логические и структурные, то в зависимости от того, на языке каких признаков производится описание этих объектов, иначе - в зависимости от того, какой алгоритм распознавания реализован, системы распознавания могут быть подразделены на детерминированные, вероятностные, логические, структурные и комбинированные.

**а) Детерминированные системы.** В этих системах для построения алгоритмов распознавания используются «геометрические» меры близости, основанные на измерении расстояний между распознаваемым объектом и эталонами

классов. В общем случае применение детерминированных методов распознавания предусматривает наличие координат эталонов классов в признаковом пространстве либо координат объектов, принадлежащих соответствующим классам.

**б) Вероятностные системы.** В данных системах для построения алгоритмов распознавания используются вероятностные методь распознавания, основанные на теории статистических решений В общем случае применение вероятностных методов распознавания предусматривает наличие вероятностных зависимостей между признаками распознаваемых объектов и классами, к которым эти объекты относятся.

**в) Логические системы.** В этих системах для построения алгоритмов распознавания используются логические методы распознавания, основанные на дискретном анализе и базирующемся на нем исчислении высказываний. В общем случае применени логических методов распознавания предусматривает наличие логических связей, выраженных через систему булевых уравнений в которой переменные - логические признаки распознаваемых объектов, а неизвестные величины - классы, к которым эти объекты относятся.

**г) Нейросетевые системы.** Эти системы основаны на использовании моделей и методов вычислительных структур, подобных в некотором смысле биологическим нейронным сетям. Достоинством этих систем являются высокие адаптивные и аппроксимационные способности.

**д) Структурные (лингвистические) системы.** В этих системах для построения алгоритмов распознавания используются специальные грамматики, порождающие языки, состоящие из предложений, каждое из которых описывает объекты, принадлежащие конкретному классу. Применение структурных методов распознавания требует наличия срвокупностей предложений, описывающих все множество объектов, принадлежащих всем классам алфавита классов системы распознавания. При этом множество предложений должно быть подразделено на подмножества по числу классов системы. Элементами подмножеств и являются предложения, описывающие объекты, принадлежащие данному подмножеству (классу). Таким образом, априорными описаниями классов являются совокупности

предложений, каждое из которых соответствует конкретному объекту, принадлежащему данному классу.

**е) Комбинированные системы.** В этих системах для построения алгоритмов распознавания используется специально разработанный метод вычисления оценок. Такие алгоритмы распознавания называют алгоритмами вычисления оценок (АВО). Их применение требует наличия таблиц, где содержатся объекты, принадлежащие соответствующим классам, а также значения признаков, которыми характеризуются эти объекты. Признаки могут быть детерминированными, логическими, вероятностными и структурными.

### 4.2 Индивидуальное прогнозирование по признакам

Индивидуальное прогнозирование является одним из перспективных направлений в проблеме повышения качества продукции. Возможности прогнозирования велики и они привлекают в настоящее время широкие круги специалистов как в области исследования, разработки и теоретического обоснования методов прогнозирования, так и в области их практического применения.

Основная цель прогнозирования - предсказание будущего состояния объекта на основе изучения таких факторов, от которых оно зависит или которые ему просто сопутствуют. При прогнозировании информацию, полученную об объекте, используют для того, чтобы дать количественную или качественную характеристику состояния объекта, процесса или явления в будущем. Прогнозирование основывается на изучении объективных закономерностей, которым подчиняются процессы в исследуемом объекте.

В процессе изготовления какого-либо изделия его элементы и материалы подвергаются как необходимым технологическим воздействиям, так и некоторым случайным воздействиям. Это приводит к тому, что параметры схем и конструкций, полученных после изготовления, являются случайными величинами. Усиленный контроль всех технологических операций и параметров изделий при изготовлении

приводит к неоправданным затратам материалов и трудовых ресурсов. В то же время весьма желательно знать не только средние показатели надежности выпускаемых изделий, но и для каждого отдельного экземпляра. Индивидуальное прогнозирование предназначено именно для этого.

Использование индивидуального прогнозирования в производстве позволяет устранить потенциально ненадежные изделия из готовой продукции, что само по себе очень важно, так как появляется возможность научно обоснованного управления качеством выпускаемой продукции за счет введения обратной связи от прогнозирования к производству.

Индивидуальное прогнозирование может применяться также и в эксплуатации. Цель индивидуального прогнозирования в эксплуатации предотвращение отказов и увеличение сроков между профилактическими работами путем выявления и исключения из эксплуатации потенциально ненадежных экземпляров с ухудшенными значениями параметров и интенсивным старением.

Чтобы применение прогнозирования оправдывало себя, необходимо выполнять следующие требования:

- точность (вероятность ошибочного прогнозирования должна быть достаточно малой);

- затраты времени на прогнозирование должны быть минимальными;

- оборудование для целей прогнозирования должно быть как можно проще и дешевле.

**Индивидуальное прогнозирование с качественной оценкой прогнозируемого параметра** - такое прогнозирование, в результате которого должно быть указано, в каком интервале значений находится величина прогнозируемого параметра каждого экземпляра к моменту $t_{пр}$. Также такое прогнозирование можно назвать прогнозированием с классификацией. В практических приложениях этого вида прогнозирования совокупность изделий бывает необходимо разделить, как правило, на два класса: годных и дефектных.

**Прогнозирование по признакам** называется также прогнозированием на основе теории распознавания образов. При таком прогнозировании начальное

состояние каждого экземпляра оценивается по значениям некоторых информативных признаков параметров изделия (признаков), вероятностно связанных с прогнозируемым параметром, и на основе этой информации определяется состояние прогнозируемого параметра каждого экземпляра в будущем к моменту времени прогнозирования ($t_{пр}$).

Требуется, используя информацию об этих наблюдениях, найти такой оператор $H_{хкл}$, используя который можно по совокупности значений признаков каждого экземпляра оценить его принадлежность к тому или иному классу. Если число классов равно двум, то для определения номера класса может быть найдено одно пороговое значение П для оператора $H_{хкл}$, которое разделит все множество значений оператора $H_{хкл}$ на две области. Тогда экземпляр будет отнесен к первому классу, если $H_{хкл} \leq П$, и ко второму классу, если $H_{хкл} > П$.

Прогнозирование состоит из четырех этапов:

- обучающего эксперимента для получения набора исходных данных;

- обучения для получения оценки класса экземпляров из рабочей выборки (с известными классами);

- экзамена для оценки ошибок классификации данным оператором прогнозирования (по результатам обучения) и улучшения оператора прогнозирования или подбора порога;

- прогнозирования для определения классов новых экземпляров.

Для решения задач индивидуального прогнозирования на основе теории распознавания образов с классификацией необходимо иметь массив исходных данных следующего состава: информация о состоянии изделия в начальный момент времени представляется значениями признаков каждого экземпляра, а состояние каждого экземпляра ко времени $t_{пр}$ определяется по тому, в каком из интервалов значений находится прогнозируемый параметр. Решение об отнесении экземпляра к конкретному классу принимается по результатам прогнозирования в зависимости от соотношения между величиной порога П и значением оператора прогнозирования. Для оптимизации оператора прогнозирования используются различные критерии,

тем или иным образом связанные с уменьшением вероятностей ошибочных решений, заключающихся в переименовании классов.

Эти вероятности находятся по данным обучающего эксперимента и обучения путем определения числа верных и ошибочных решений по каждому классу, то есть в результате экзамена. Если вычисленное значение вероятности ошибочных решений не превышает заданных допустимых значений, полученный оператор можно рекомендовать для прогнозирования класса новых экземпляров, не участвовавших в обучающем эксперименте.

Качество прогнозирования может быть улучшено как выбором соответствующего оператора прогнозирования, так и подбором более информативных признаков.

Под **эвристическим прогнозированием** понимается искусство суждения о развитии и исходе событий на основе субъективно взвешенного набора фактов, большая часть которых носит качественный характер.

Алгоритм прогнозирования не вытекает из строгих положений теории, а в значительной степени основан на интуиции и опыте исследователя.

Такие методы могут давать удовлетворительные результаты и при ограниченной информации о вероятностных характеристиках признаков и прогнозируемого параметра. Для применения этих методов необходимо иметь набор признаков, сильно коррелированных с прогнозируемым параметром, и необязательно знать вид их условных плотностей распределения. Естественно, методы индивидуального прогнозирования, основанные на использовании эвристических алгоритмов, в отличие от методов оптимального оценивания не всегда приводят к оптимальным решениям. Однако для их применения на практике достаточно, чтобы ошибка прогнозирования не превышала допустимого значения, а этого можно добиться, например, подбором более информативных признаков, применением соответствующих способов улучшения оператора прогнозирования.

**Методы оптимального оценивания класса** прогнозируемого экземпляра основаны на теории статистических оценок. Такие методы довольно сложно реализуются на практике, требуют для своего применения информацию о

совместной плотности распределения признаков и прогнозируемого параметра, однако они позволяют получить оптимальное решение.

**Вероятностное прогнозирование** основывается на использовании вероятностных моделей. Эти методы прогнозирования являются методами, адекватными природе изучаемых объектов, и их использование не может считаться лишь промежуточным этапом в исследованиях. Необходимо подчеркнуть, что использование методов вероятностного прогнозирования возможно лишь при выполнении требования статистической устойчивости параметров объекта. Смысл этого заключается в том, что в процессе производства и эксплуатации вероятностные характеристики параметров исследуемого объекта от экземпляра к экземпляру обладают определенной однородностью, как в отношении начального разброса параметров, так и при изменении внешних воздействий во времени. Это означает, что выборочные значения исследуемого параметра принадлежат одной генеральной совокупности.

Индивидуальное прогнозирование основывается на изучении вероятностной связи между значением прогнозируемого параметра данного экземпляра по истечению времени, называемом временем прогнозирования, и начальным состоянием этого экземпляра.

### 4.3 Алгоритм оптимальных классификаций

Рассмотрим решение задачи индивидуального прогнозирования методами теории статистической классификации, для чего нужно располагать условными многомерными плотностями распределения признаков для каждого класса. Задача индивидуального прогнозирования заключается в отыскании способа принятия оптимального решения о принадлежности проверяемого экземпляра к тому или иному классу в условиях неопределенности. То есть в условиях действия случайных факторов, которые маскируют связь между признаками и классом экземпляра.

Проверяемый экземпляр принадлежит к классу $К_1$, если значение прогнозируемого параметра у к моменту времени $t_{пр}$ будет больше некоторого граничного значения $y_{гр}$. Такие изделия считаются годными. А если $y<y_{гр}$ - экземпляр принадлежит к классу $К_2$, дефектных изделий.

Пусть начальное состояние изделия характеризуется k признаками, каждый из которых является случайной величиной. Совокупность k случайных величин обозначим как $\{\tilde{x}_i\}=\tilde{x}_1,\tilde{x}_2,...,\tilde{x}_k$.

По конкретным значениям признаков $\{x_i^{(j)}\}$ j-го экземпляра необходимо принять решение об отнесении этого экземпляра к классу $К_1$ или $К_2$. В таком виде задачу целесообразно ставить лишь в том случае, когда между классом, к которому принадлежит j-ый экземпляр и значениями его признаков существует какая-либо связь.

В задачах прогнозирования предполагается наличие вероятностей связи между классом и признаками j-го экземпляра. Степень тесноты этой связи полностью определяется видом условных совместных плотностей распределения признаков $\{\tilde{x}_i\}$ при условии, что экземпляр принадлежит к классу $К_1$ и к классу $К_2$, соответственно:

$$W(x_1,x_2,...,x_k,t_1/K_1,t_{пр}), \quad W(x_1,x_2,...,x_k,t_1/K_2,t_{пр}).$$

Далее будем обозначать их как $W(x_1,x_2,...x_k/K_1)$ и $W(x_1,x_2,...x_k/K_2)$. Эти плотности могут быть получены соответствующей обработкой результатов эксперимента.

Совместная плотность и условная плотность связаны соотношениями:

$$W(x_1,x_2,...,x_k/K_1)=C_1\int_{y_{гр}}^{\infty}W(y,x_1,x_2,...,x_k)dy, \qquad (4.1)$$

$$W(x_1,x_2,...,x_k/K_2)=C_2\int_{-\infty}^{y_{гр}}W(y,x_1,x_2,...,x_k)dy \qquad (4.2)$$

где $C_1,C_2$ - нормирующие коэффициенты.

$$C_1=\frac{1}{P(K_1)}=[\int_{y_{гр}}^{\infty}\int_{-\infty}^{\infty}...\int_{-\infty}^{\infty}W(y,x_1,x_2,...,x_k)dydx_1...dx_k]^{-1},$$

$$C_1=\frac{1}{P(K_2)}=[\int_{-\infty}^{y_{гр}}\int_{-\infty}^{\infty}...\int_{-\infty}^{\infty}W(y,x_1,x_2,...,x_k)dydx_1...dx_k]^{-1}$$

Введем следующие обозначения:

$P(\text{реш}K_2/K_1)$ - условная вероятность принятия решения об отнесении экземпляра к классу $К_2$ при условии, что он фактически принадлежит к классу $К_1$.

$P(\text{реш}K_1/K_2)$ - условная вероятность принятия решения об отнесении экземпляра к классу $К_1$ при условии, что он фактически принадлежит к классу $К_2$.

$P(K_1/\text{реш}K_2)$ - условная вероятность того, что экземпляр фактически является годным (принадлежит к классу $К_1$) при условии, что принято решение считать его дефектным, то есть отнести его к классу $К_2$. Эта вероятность определяет, насколько рискует изготовитель, когда верит прогнозированию. Эту вероятность называют риском изготовителя.

$P(K_2/\text{реш}K_1)$ - условная вероятность того, что экземпляр в действительности принадлежит к классу $К_2$ (является дефектным) при условии, что принято решение считать его годным, то есть отнести его к классу $К_1$. Эта вероятность определяет, насколько рискует потребитель, когда верит прогнозирования. Эту вероятность называют риском потребителя.

$P(\text{реш}K_1)$- априорная вероятность принятия решения об отнесении экземпляра к классу $К_1$, то есть вероятность отнесения к годным любого взятого наугад экземпляра.

$P(\text{реш}K_2)$- априорная вероятность принятия решения об отнесении экземпляра к классу $К_2$, то есть вероятность отнесения к дефектным любого взятого наугад экземпляра.

Вероятность ошибки в переименовании класса экземпляра из $К_1$ в $К_2$, то есть вероятность того, что экземпляр является фактически годным (класс $К_1$) и относительного его принято решение об отнесении к классу $К_2$ равна в соответствии с теоремой умножения вероятностей:

$$P(K_1 \times \text{реш}K_2) = P(\text{реш}K_2/K_1)\, P(K_1) = P(K_1/\text{реш}K_2)\, P(\text{реш}K_2). \qquad (4.3)$$

Аналогично вероятность ошибки в переименовании класса экземпляра из $К_2$ в $К_1$, то есть вероятность отнесения фактически дефектных экземпляров к годным, равна:

$$P(K_2 \times \text{решK}_1) = P(\text{решK}_1/K_2)\,P(K_2) = P(K_2/\text{решK}_1)\,P(\text{решK}_1). \qquad (4.4)$$

Обозначим потери связанные с переименованием класса экземпляра из $К_1$ в $К_2$, то есть цену такого переименования $Ц_{1->2}$, а цену переименования класса экземпляра из $К_2$ в $К_1$ как $Ц_{2->1}$. Тогда величина среднего риска (средних потерь) при многократном распознавании равна, с учетом выражений (4.3) и (4.4):

$$\rho = P(K_1 \times \text{решK}_2)Ц_{1->2} + P(K_2 \times \text{решK}_1)Ц_{2->1},$$
$$\rho = P(\text{решK}_2/K_1)P(K_1)Ц_{1->2} + P(\text{решK}_1/K_2)P(K_2)Ц_{2->1}. \qquad (4.5)$$

В качестве критерия оптимальности естественно взять **минимум среднего риска**, то есть: $\rho \to \min$. Это наиболее распространенный критерий, его называют **критерием Байеса**.

В выражении (4.5) вероятности $P(K_1)$ и $P(K_2)$ известны и не зависят от процедуры прогнозирования, так как их величина определяется фактической долей годных и дефектных экземпляров среди изделий данного типа. Значит, минимизация среднего риска может быть достигнута путем изменения вероятностей $P(\text{решK}_2/K_1)$ и $P(\text{решK}_1/K_2)$. Эти вероятности могут быть определены по известным условным совместным плотностям распределения признаков (4.1) и (4.2).

$$P(\text{решK}_2/K_1) = \int_{V_2}\!\!\cdots\!\int W(x_1, x_2, \ldots, x_k/K_1)\,dx_1 dx_2 \ldots dx_k \qquad (4.6)$$

$$P(\text{решK}_1/K_2) = \int_{V_1}\!\!\cdots\!\int W(x_1, x_2, \ldots, x_k/K_2)\,dx_1 dx_2 \ldots dx_k \qquad (4.7)$$

где $V_1, V_2$ - область знаний признаков $(x_1, x_2, \ldots, x_k)$, при которых принимается решение об отнесении экземпляра к классу $К_2$ или $К_1$ соответственно. Подставляя выражения (4.6) и (4.7) в (4.5) получим:

$$\rho = \int_{V_2}\!\!\cdots\!\int P(K_1)Ц_{1\to 2}W(x_1, x_2, \ldots, x_k/K_1)\,dx_1 dx_2 \ldots dx_k +$$
$$+ \int_{V_1}\!\!\cdots\!\int P(K_2)Ц_{2\to 1}W(x_1, x_2, \ldots, x_k/K_2)\,dx_1 dx_2 \ldots dx_k$$

Для того, чтобы избежать интегрирования многомерных плотностей, приведем два многомерных интеграла к одному используя соотношение:

$$P(\text{реш}K_2 / K_1) = 1 - P(\text{реш}K_1 / K_2) = 1 - \int_{V_1} ... \int W(x_1, x_2, ..., x_k / K_1) dx_1 dx_2 ... dx_k$$

Тогда получим:

$$\rho = P(K_1)Ц_{1\to 2} - \int_{V_1} ... \int [P(K_1)Ц_{1\to 2} W(x_1, ..., x_k / K_1) - P(K_2)Ц_{2\to 1} W(x_1, ..., x_k / K_2)] dx_1 dx_2 ... dx \qquad (4.8)$$

Для минимизации величины среднего риска следует так выбрать область решения по классу $K_1$, то есть $V_1$, чтобы интеграл выражения (4.8) принял наибольшее положительное значение. Для этого необходимо так выбрать область $V_1$, чтобы подынтегральное выражение в (4.8) было положительным во всей этой области и отрицательно вне этой области, то есть всюду в области $V_1$ и только в ней должно выполнятся условие.

$$P(K_1)Ц_{1\to 2} W(x_1, ..., x_k / K_1) - P(K_2)Ц_{2\to 1} W(x_1, ..., x_k / K_2) > 0 . \qquad (4.9)$$

Значит решение об отнесении j-го экземпляра к классу $K_1$ принимается тогда, когда совокупность значений его признаков $\{x_i^{(j)}\}$ удовлетворяет неравенству (4.9). Преобразуя это выражение получим, что средний риск минимален, если решение об отнесении экземпляра к классу $K_1$ принимается когда:

$$\frac{W(x_1, x_2, ..., x_k / K_1)}{W(x_1, x_2, ..., x_k / K_2)} > \frac{P(K_2)Ц_{2\to 1}}{P(K_1)Ц_{1\to 2}} .$$

Обозначим отношение

$$\frac{W(x_1, x_2, ..., x_k / K_1)}{W(x_1, x_2, ..., x_k / K_2)} = \lambda(x_1, ..., x_k) .$$

Его называют **отношением правдоподобия**. Величина, которая определяет пороговое значение отношения правдоподобия и не зависит от значений признаков, равна

$$\frac{P(K_2)Ц_{2\to 1}}{P(K_1)Ц_{1\to 2}} = П .$$

Обозначим $\lambda^{(j)}$ - отношение правдоподобия найденное для j-го экземпляра:

$$\lambda^{(j)} = \frac{W(x_1^{(j)}, x_2^{(j)}, ..., x_k^{(j)} / K_1)}{W(x_1^{(j)}, x_2^{(j)}, ..., x_k^{(j)} / K_2)} .$$

Тогда алгоритм оптимальной классификации формулируется: если $\lambda^{(j)} \geq П$, то принимается решение об отнесении j-го экземпляра к классу $K_1$; если $\lambda^{(j)} < П$ -

принимается решение об отнесении j-го экземпляра к классу $К_2$. Если k=1, то есть задача классификации решается по одному признаку, то классификацию можно проводить по пороговому значению $X_{кл}$ самого признака, который находится из уравнения:

$$\lambda(X_{кл}) = П .$$

Алгоритм оптимальной классификации дает минимальную величину среднего риска, то есть потери вызванные ошибочными решениями будут минимальными. А любой другой алгоритм классификации не приведет к меньшим потерям. Однако на практике метод оптимальной классификации используется редко поскольку многомерные плотности распределения признаков и прогнозируемого параметра, как правило, неизвестны.

### 4.4 Классификация на основе эвристических алгоритмов

При прогнозировании по признакам с классификацией задача состоит в разделении исследуемой совокупности изделий на классы и нет необходимости в оценке конкретного значения прогнозируемого параметра. В большинстве практических приложений этого метода число классов равно двум. Так бывает, например, когда исследуемую совокупность необходимо по заданному правилу разделить на класс годных и дефектных изделий.

Для решения задачи прогнозирования методами теории статистической классификации необходимо располагать условными многомерными плотностями распределения признаков для каждого класса, тогда задача заключается в отыскании способа принятия оптимального решения о принадлежности проверяемого экземпляра к тому или иному классу в условиях неопределенности, т.е. в условиях действия случайных факторов, маскирующих связь между признаками и классом экземпляра.

Классические статистические методы дают оптимальное решение задачи прогнозирования. Однако практическое применение этих методов возможно, если проведен специальный эксперимент по сбору и такой обработке статистических данных о прогнозируемом параметре и признаках, в результате которой найдены

подходящие аналитические модели условных многомерных плотностей распределения прогнозируемого параметра и признаков. Однако в реальных задачах исследователь сталкивается здесь с рядом проблем, поэтому реализовать классические статистические методы не всегда возможно.

Во-первых, для реальных изделий даже при известной совокупности информативных признаков (выявление которых представляет весьма трудоемкую задачу) не всегда изучены многомерные условные плотности распределения признаков и прогнозируемого параметра.

Во-вторых, получение аналитических моделей этих условных плотностей распределения представляет трудоемкий процесс и может быть поставлено только отдельной самостоятельной задачей для каждого типа изделий и для данных условий эксплуатации.

В-третьих, даже если такие аналитические модели получены, необходимые при этих методах прогнозирования аналитические преобразования достаточно сложны. Задача относительно легко решается аналитически, если многомерные условные плотности подчиняются нормальному закону, что в действительности имеет место далеко не всегда.

В связи со сказанным выше представляет интерес применение методов решения задач прогнозирования, основанных на эвристических алгоритмах. Смысл понятия “эвристический алгоритм” состоит в том, что в этом случае алгоритм прогнозирования не вытекает из строгих положений теории, а в значительной степени основан на интуиции и опыте исследователя.

Такие методы могут давать удовлетворительные результаты и при ограниченной исходной информации о вероятностных характеристиках признаков и прогнозируемого параметра. Так, для применения этих методов для прогнозирования по признакам необходимо иметь набор признаков, сильно коррелированных с прогнозируемым параметром, и необязательно знать вид их условных плотностей распределения.

Следует сказать, что методы прогнозирования, основанные на использовании эвристических алгоритмов, не всегда приводят к оптимальным решениям. Однако

для их применения на практике достаточно, чтобы ошибка прогнозирования не превышала допустимого значения, а этого можно добиться, например, подбором более информативных признаков, применением соответствующих способов улучшения оператора прогнозирования.

### 4.4.1 Эвристический алгоритм классификации

Пусть мы имеем обучающую выборку $x=\{x^1,x^2,...,x^S\}$, состоящую из S экземпляров, характеризующихся N признаками $x^q_i$, где q - номер экземпляра обучающей выборки, i - номер признака. Каждому экземпляру обучающей выборки сопоставлен номер класса $y^q$, $\forall y^q \in \{K_0, K_1\}$, где $K_0$ и $K_1$ - условные обозначения разных классов экземпляров. Условимся, что проверяемый экземпляр принадлежит к классу $K_0$, если значение прогнозируемого параметра к моменту времени прогнозирования будет больше некоторого граничного значения; будем считать такие изделия годными. В противном случае экземпляр принадлежит к классу $K_1$ (дефектных).

В общем случае, экземпляры характеризуются достаточно большим количеством признаков, имеющих разную (как правило, относительно небольшую) информативность. Признаки зачастую связаны с прогнозируемым номером класса экземпляра нелинейными связями, что не позволяет в большинстве случаев построить линейную модель для классификации по одному признаку, удовлетворяющую заданным требованиям достоверности прогнозирования. Поэтому метод классификации должен быть многомерным (учитывать все признаки) и должен учитывать информативность (значимость) признаков.

Наиболее простым способом реализации многомерной классификации, очевидно, будет объединение результатов одномерных классификаций с учетом значимости признаков.

Если $\psi(x_i)$ – результат одномерной классификации по i-му признаку, то y - номер класса экземпляра можно представить как округленную взвешенную сумму:

$$y = \text{round}\left(\sum_{i=1}^{N} б_i ш(x_i)\right), \qquad (4.10)$$

где $\alpha_i$ – коэффициент, учитывающий значимость результата одномерной классификации по i-му признаку (доля i-го признака в формировании значения y).

Если значение $\alpha_i$ известно, то задача сводится к разработке правила одномерной классификации для i-го признака относительно порога классификации $\theta_i$. Это правило должно быть нелинейным, а также должно учитывать степень близости экземпляров к центрам классов. В качестве такого правила предлагается использовать сигмоидую функцию:

$$ш(x_i) = \frac{1}{1 + e^{(x_i - и_i)}},$$

которая будет тем ближе к 1, чем ближе экземпляр к центру класса $К_1$, и наоборот, тем ближе к 0, чем ближе экземпляр к центру класса $К_0$. Это правило предполагает, что центр класса $К_0$ находится левее центра класса $К_1$, что на практике происходит далеко не всегда, поэтому введем в правило одномерной классификации параметр $\beta_i$, учитывающий наиболее вероятное размещение центров классов относительно порога $\theta_i$:

$$ш(x_i) = \frac{1}{1 + e^{в_i(x_i - и_i)}}. \qquad (4.11)$$

Подставив (4.11) в выражение (4.10), получим правило классификации:

$$y = \text{round}\left( \sum_{i=1}^{N} б_i \frac{1}{1 + e^{в_i(x_i - и_i)}} \right). \qquad (4.12)$$

Для нахождения значений параметров $\alpha_i$, $\beta_i$ и $\theta_i$ предлагается использовать следующий алгоритм.

Шаг 1. Вычислить:

$м_{x_i}$ - математическое ожидание i-го признака $x_i$:

$$м_{x_i} = \frac{1}{S} \sum_{q=1}^{S} x_i^q, \quad i=1,2,...,N$$

где S – количество экземпляров обучающей выборки, $x^q_i$-значение i-го признака q-го экземпляра обучающей выборки.

$м_{x_i}^{K_1}$ - математическое ожидание i-го признака для экземпляров обучающей выборки, принадлежащих к классу $К_1$:

$$M_{x_i}^{K_1} = \frac{1}{S^{K_1}} \sum_{q=1}^{S^{K_1}} x_i^q, x_i^q \in K_1,\ i=1,2,...,N$$

где $S^{K_1}$ - количество экземпляров обучающей выборки, принадлежащих к классу $K_1$.

$M_{x_i}^{K_0}$ - математическое ожидание i-го признака для экземпляров обучающей выборки, принадлежащих к классу $K_0$:

$$M_{x_i}^{K_0} = \frac{1}{S^{K_0}} \sum_{q=1}^{S^{K_0}} x_i^q, x_i^q \in K_0,\ i=1,2,...,N$$

где $S^{K_0}$ - количество экземпляров обучающей выборки, принадлежащих к классу $K_0$.

$M_y$ - математическое ожидание номера класса:

$$M_y = \frac{1}{S} \sum_{q=1}^{S} y^q,$$

где $y^q$ - номер класса q-го экземпляра обучающей выборки.

Если необходимо для нахождения порога $\theta_i$, вычислить:

Дисперсии признаков:

$$D_{x_i} = \frac{1}{S} \sum_{q=1}^{S} \left(x_i^q - M_{x_i}\right)^2,\ i = 1,2,...,N.$$

Дисперсии признаков экземпляров, относящихся к классу $K_1$:

$$D_{x_i}^{K_1} = \frac{1}{S^{K_1}} \sum_{q=1}^{S^{K_1}} \left(x_i^q - M_{x_i}^{K_1}\right)^2, x_i^q \in K_1,\ i = 1,2,...,N.$$

Дисперсии признаков экземпляров, относящихся к классу $K_0$:

$$D_{x_i}^{K_0} = \frac{1}{S^{K_0}} \sum_{q=1}^{S^{K_0}} \left(x_i^q - M_{x_i}^{K_0}\right)^2, x_i^q \in K_0,\ i = 1,2,...,N.$$

Дисперсию номера класса:

$$D_y = \frac{1}{S} \sum_{q=1}^{S} \left(y^q - M_y\right)^2.$$

Шаг 2. Вычислить коэффициенты корреляции каждого i-го признака и номера класса:

$$r_{x_i y} = \frac{\sum_{q=1}^{S} \left(x_i^q - M_{x_i}\right)\left(y^q - M_y\right)}{\sqrt{\sum_{q=1}^{S} \left(x_i^q - M_{x_i}\right)^2 \sum_{q=1}^{S} \left(y^q - M_y\right)^2}}.$$

Шаг 3. Вычислить степень (долю) влияния i-го признака на номер класса экземпляра:

$$б_i = \frac{|r_{x_i y}|}{\sum_{j=1}^{N} |r_{x_j y}|}, \quad i=1,2,...,N,$$

где j – номер текущего признака .

Вычислить коэффициент $\beta_i$, учитывающий наиболее вероятное размещение полюсов (центров сосредоточения экземпляров) классов при одномерной классификации по i-му признаку:

$$в_i = \text{sign}\left(M_{x_i}^{K_1} - M_{x_i}^{K_0}\right).$$

Этот коэффициент будет равен:

+1, если полюс класса $K_0$ расположен левее полюса класса $K_1$ по оси зачений i-го признака;

–1, если полюс класса $K_0$ расположен правее полюса класса $K_1$ по оси зачений i-го признака;

0, если полюса классов совпадают.

Вычислить значение порога, относительно которого будем осуществлять одномерную классификацию экземпляров по i-му признаку. Для нахождения значения порога можно предложить достаточно много различных способов. Рассмотрим наиболее быстрые и простые.

$$и_i = \frac{|M_{x_i}^{K_1} - M_{x_i}^{K_0}|}{2} + \min\left(M_{x_i}^{K_1}, M_{x_i}^{K_0}\right), \; i=1,2,...,N, \qquad (4.13)$$

$$и_i = M_{x_i} + \frac{D_{x_i}(0.5 - M_y)}{r_{x_i y} D_y}, \; i=1,2,...,N, \qquad (4.14)$$

$$и_i = \max\left(M_{x_i}^{K_1}, M_{x_i}^{K_0}\right) - \frac{|M_{x_i}^{K_1} - M_{x_i}^{K_0}|}{1 + \frac{D_{x_i}^{K_0}}{D_{x_i}^{K_1}}}, \quad i=1,2,...,N. \qquad (4.15)$$

Шаг 4. Оценить вероятности ошибки переименования классов для экземпляров обучающей и (или) контрольной выборок и сделать вывод о применимости данного алгоритма для решения поставленной задачи. Для этого следует найти значения номеров классов для экземпляров обучающей и (или) контрольной выборок по

правилу классификации (4.12), после чего определить количество неверных решений $S_{ош}$ и оценить вероятность принятия ошибочных решений $P_{ош}=S_{ош}/(S+S_к)$, где S и $S_к$ – размер обучающей и контрольной выборок, соответственно.

### 4.4.2 Нейросетевая интерпретация эвристического алгоритма

Из приведенного эвристического алгоритма можно видеть, что его правило классификации делится на два этапа: на первом этапе осуществляется одномерная классификация на основе N признаков, на втором этапе осуществляется объединение результатов, полученных на первом этапе, с учетом значимости признаков. Такой метод подобен классификации на основе двуслойного перцептрона, являющегося частным случаем многослойной нейронной сети.

Нейросетевая интерпретация эвристического алгоритма классификации представлена на рис.4.3.

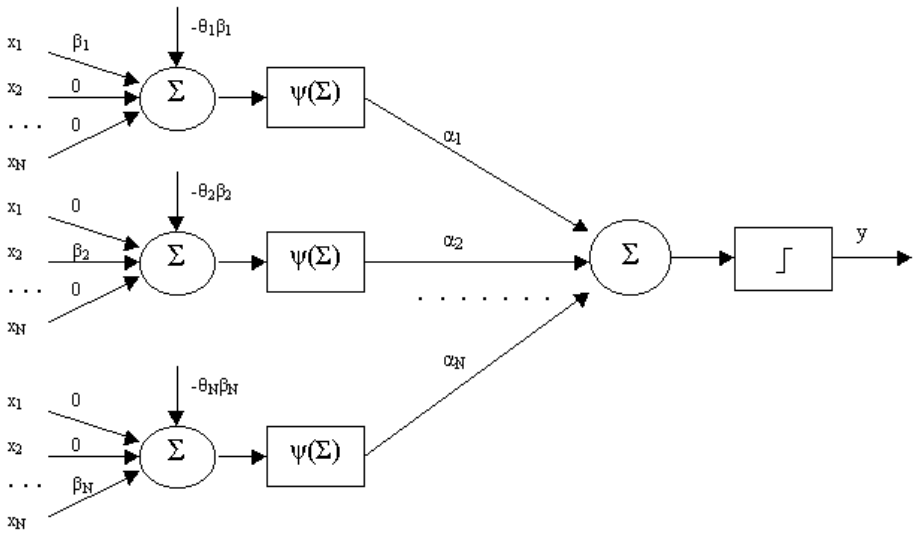



Рис 4.3 - Нейросетевая интерпретация эвристического алгоритма классификации.

Правила вычисления параметров эвристического алгоритма в этом случае останутся неизменными, а параметры и функции активации НС необходимо будет определить на их основе по следующим правилам.

Функция активации $ш^{(м,i)}$ i-го нейрона м-го слоя:

$$ш^{(1,i)}(\vartheta) = \frac{1}{1+e^{-\vartheta}}, \quad \forall i = \overline{1,N},$$

$$ш^{(2,1)}(\vartheta) = \begin{cases} 0, \vartheta \le 0, \\ 1, \vartheta > 0. \end{cases}$$

Весовой коэффициент $w_j^{(м,i)}$ j-го входа i-го нейрона м-го слоя:

$$w_j^{(м,i)} = \begin{cases} 0, & i \ne j,\ j > 0,\ м = 1, \\ в_i, & i = j,\ j > 0,\ м = 1, \\ -и_i в_i, & j = 0,\ м = 1, \\ б_i, & м = 2, \end{cases}$$

i = 1,2,...,N, j = 0,1,2,...,N.

### 4.4.3 Алгоритм классификации и обучения двухслойного персептрона

Пусть имеется обучающая выборка, содержащая S экземпляров, характеризующихся N признаками $\{x^s_j\}$, s = 1, 2, ..., S; j = 1, 2, ..., N. Каждому экземпляру обучающей выборки $x^s$ сопоставлен бинарный номер класса $y^s$, к которому относится экземпляр.

Представим бинарный номер класса экземпляра как бинарную функцию суммы частных одномерных классификаций по N признакам:

$$y^s = ш\left(\sum_{j=1}^{N} y^s_j\right),$$

где $y^s_j$ - результат (бинарный номер класса) одномерной классификации s-го экземпляра обучающей выборки по j-му признаку, а бинарная функция задается формулой:

$$ш(x)=\begin{cases}0, x \le 0;\\ 1, x > 0.\end{cases}$$

Одномерные классификации по признакам будем осуществлять следующим образом.

Вначале найдем средние значения (оценки математического ожидания) признаков для экземпляров каждого класса по отдельности:

$$C_j^0 = \frac{1}{S^0}\sum_{s=1}^{S} x_j^s, y^s = 0;$$

$$C_j^1 = \frac{1}{S^1}\sum_{s=1}^{S} x_j^s, y^s = 1;$$

j = 1, 2, ..., N,

где $S^0$ и $S^1$ - количество экземпляров, принадлежащих к классам 0 и 1, соответственно.

Затем для учета рассеяния значений признаков экземпляров по классам около математических ожиданий признаков по классам определим координаты центров классов:

$$C_j^{0*} = C_j^0 + ky_j^0;$$

$$C_j^{1*} = C_j^1 + ky_j^1;$$

j = 1, 2, ..., N,

где k - некоторый коэффициент ($k \ge 0$), а $y_j^0$ и $y_j^1$ - средние квадратические отклонения j-го признака для экземпляров классов 0 и 1, соответственно.

Средние квадратические отклонения найдем по формулам:

$$y_j^0 = \sqrt{\frac{1}{S^0}\sum_{s=1}^{S}\left(x_j^s - C_j^0\right)^2}, y^s = 0,$$

$$y_j^1 = \sqrt{\frac{1}{S^1}\sum_{s=1}^{S}\left(x_j^s - C_j^1\right)^2}, y^s = 1,$$

j = 1, 2, ..., N,

где $S^0$ и $S^1$ - количество экземпляров обучающей выборки, принадлежащих к классам 0 и 1, соотвественно.

Зная координаты центров классов с учетом рассеяния их значений, определим результаты одномерных классификаций $y_j^s$:

$$y_j^s = \begin{cases} -1, x_j \le C_j^{0*}; \\ 0, C_j^{0*} < x_j < C_j^{1*}; \\ 1, x_j \ge C_j^{1*}. \end{cases}$$

Рассмотренный алгоритм можно использовать для обучения двухслойного персептрона, первый слой которого будет осуществлять одномерную классификацию, а второй слой - объединять результаты одномерных классификаций по признакам.

Схема персептрона, обученного на основе рассмотренного алгоритма, приведена на рис. 4.4.

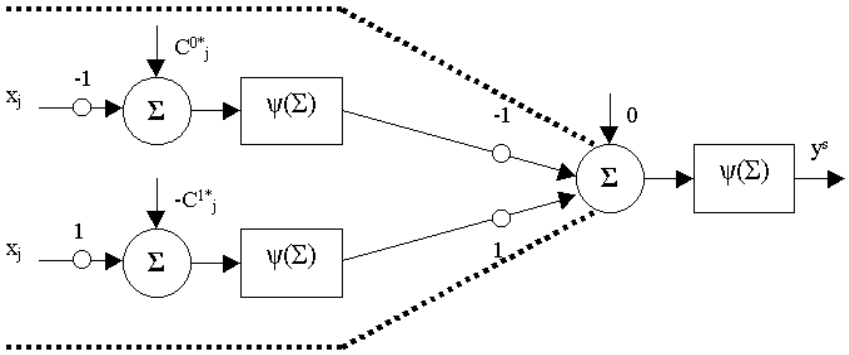



Рис. 4.4 - Схема двухслойного персептрона

Нейроны обоих слоев сетии имеют пороговую функцию активации:

$$ш(x) = \begin{cases} 0, x \le 0; \\ 1, x > 0. \end{cases}$$

Весовые коэффициенты нейронов определяются по формуле:

$$w_j^{(м,i)} = \begin{cases} -1, м = 1, i = 1,3,5,...,(2N-1), j = 1; \\ 1, м = 1, i = 2,4,6,...,2N, j = 1; \\ C_j^{0*}, м = 1, i = 1,3,5,...,(2N-1), j = 0 \\ -C_j^{1*}, м = 1, i = 2,4,6,...,2N, j = 0; \\ -1, м = 2, i = 1, j = 2N-1; \\ 1, м = 2, i = 1, j = 2N; \\ 0, м = 2, i = 1, j = 0, \end{cases}$$

где $w_j^{(м,i)}$ - вес j-го входа i-го нейрона $м$-го слоя сети.

### 4.4.4 Алгоритм классификации и обучения пятислойного персептрона

В алгоритме, рассмотренном выше, мы осуществляли одномерные классификации относительно центров сосредоточения экземпляров по отдельным признакам, которые затем объединяли.

В случае, когда классы сложно разделимы, рассмотренный метод будет плохо работать. Поэтому необходимо дополнить его таким образом, чтобы классы лучше разделялись, т.е. необходимо ввести некоторую свертку, которая сжимала бы экземпляры внутри классов, а разные классы удаляла бы друг от друга.

Разделим набор признаков на две группы.

К первой группе отнесем те признаки, у которых

$C^{0*}_j$ - координата центра класса 0, меньше $C^{1*}_j$ - сответствующей координаты центра класса 1.

Ко второй группе отнесем те признаки, у которых: $C^{1*}_j < C^{0*}_j$.

Свертку для разделения экземпляров определим как соотношение сумм признаков, соответствующих групп:

$$C(s) = \frac{\sum_{j=1}^{N} в_j x_j^s}{\sum_{j=1}^{N} (1 - в_j) x_j^s},$$

где $в_j$ - коэффициент принадлежности j-го признака к группе:

$$в_j = \begin{cases} 1, C_j^{0*} > C_j^{1*}; \\ 0, C_j^{0*} \le C_j^{1*}. \end{cases}$$

В выражении для свертки C(s) в числитель попадут признаки, относящиеся к первой группе, в знаменатель - ко второй группе.

Аналогичным образом для центров классов свертки будут иметь вид:

$$C^0 = \frac{\sum_{j=1}^{N} в_j C_j^{0*}}{\sum_{j=1}^{N} (1 - в_j) C_j^{0*}},$$

$$C^1 = \frac{\sum_{j=1}^{N} (1 - в_j) C_j^{1*}}{\sum_{j=1}^{N} в_j C_j^{1*}}.$$

Для классификации экземпляров будем применять следующее правило: s-ый экземпляр $x^s$ будем относить к тому классу, расстояние от свертки C(s) до свертки центра которого будет меньше: если $(C(s) - C^0)^2 < (C(s) - C^1)^2$, тогда установить: $y^{s*} = 0$, в противном случае - установить: $y^{s*} = 1$, где $y^{s*}$ - расчетный номер класса для s-го экземпляра.

Как и в предыдущем случае, данный алгоритм может иметь нейросетевую интерпретацию в виде пятислойного персептрона (рис. 4.5).

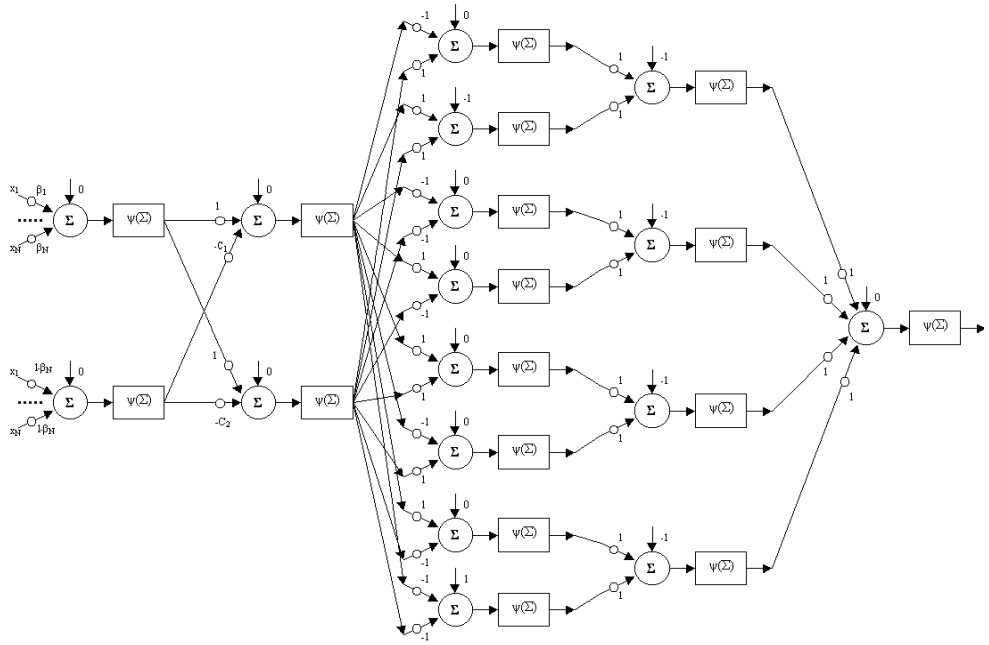


Рис. 4.5 - Схема пятислойного персептрона

Функция активации i-го нейрона м-го слоя сети определяется по формулам:

$$ш(x)^{(1,\forall i)} = ш(x)^{(2,\forall i)} = x,$$

$$ш(x)^{(3,\forall i)} = ш(x)^{(4,\forall i)} = ш(x)^{(5,\forall i)} = \begin{cases} 0, x \le 0; \\ 1, x > 0. \end{cases}$$

Весовой коэффициент j-го входа i-го нейрона м-го слоя $w_j^{(м,i)}$ определяется по формуле:

$$w_j^{(м,i)} = \begin{cases} в_j, м = 1, i = 1, \forall j > 0; \\ 0, м = 1, \forall i, j = 0; \\ 1 - в_j, м = 1, i = 1, \forall j > 0; \\ 0, м = 2, i = 2, \forall j > 0; \\ 1, м = 2, \forall i, j = 0; \\ - C_1, м = 2, i = 1, j = 2; \\ - C_2, м = 2, i = 2, j = 2; \\ 0, м = 3, i = 1,3,4,5,6,7, j = 0; \\ - 1, м = 3, i = 2, j = 0; \\ 1, м = 3, i = 8, j = 0; \\ - 1, м = 3, i = 3,8, \forall j > 0; \\ 1, м = 3, i = 2,5, \forall j > 0; \\ - 1, м = 3, i = 1,6, j = 1; \\ 1, м = 3, i = 1,6, j = 2; \\ - 1, м = 3, i = 2,7, j = 2; \\ 1, м = 3, i = 4,7, \forall j = 1; \\ 1, м = 4, \forall i, j > 0; \\ - 1, м = 4, \forall i, j = 0; \\ 1, м = 5, i = 1, j > 0; \\ 0, м = 5, i = 1, j = 0. \end{cases}$$

## 4.5 Метод стохастической аппроксимации

Пусть существует некоторая E - решающая плоскость с параметрами e и $e_0$, а $D_1$ и $D_2$ - множества объектов обучающей выборки из классов $K_1$ и $K_2$ соответственно. Зная E, $D_1$ и $D_2$, можно определить положение каждой точки обучающей выборки относительно решающей плоскости E и по выбранному виду показателя качества обучения $\varphi = \varphi(E, D_1, D_2)$ определить его значение.

Для нахождения экстремума величины φ необходимо отыскать такую плоскость $E^*$, для которой extr φ = φ ( $E^*$, $D_1$,$D_2$ ).

Существует множество алгоритмов приближенного решения этой задачи для различных видов показателей качества обучения.

В частности, для решения этой задачи можно использовать метод стохастической аппроксимации или градиентные алгоритмы оптимизации.

Метод стохастической аппроксимации, развитый Я. З. Цыпкиным применительно к проблеме распознавания и ряду смежных проблем, позволяет оптимизировать процесс разделения в пространстве признаков. Разделение на два класса в пространстве признаков может быть сведено к построению разделяющей функции f(x) и использованию правила решения:

$x \in K_1$, если $f(x)>0$;

$x \in K_2$, если $f(x)<0$.

В общем виде разделяющая функция определяется по формуле:

$$f(x) = \sum_{i=1}^{v} \lambda_i \varphi_i = \lambda\varphi ,$$

где $\lambda$- весовой вектор; $\varphi = (\varphi_1(x), \varphi_2(x), ..., \varphi_v(x))$ - вектор-функция. Построение разделяющей функции сводится к нахождению или выбору класса функций $\{\varphi_i(x)\}$ и определению коэффициентов разложения $\lambda_i$. Таким образом, одним из методов решения задачи распознавания является **метод аппроксимации**.

Будем считать, что класс функций $\{\varphi_i(x)\}$ выбран надлежащим образом и задача состоит в определении коэффициентов $\lambda_i$. Разделяющую функцию будем обозначать f (x, $\lambda$), подчеркивая зависимость от вектора $\lambda$. Если $f^*(x)$ — точное значение разделяющей функции, то погрешность аппроксимации можно определить как квадратичную погрешность

$$J(\lambda) = \int_{x \in K} \left(f^*(x) - f(x,\lambda)\right)^2 dx$$

где $x \in K$ означает, что интегрирование проводится по всей области изменения x, а dx - элемент области. Однако в реальных задачах различные значения x (объекты, имеющие набор признаков x) имеют разную вероятность появления и

потому более эффективным критерием точности будет среднеквадратичная погрешность:

$$J(\lambda) = \int_{x \in K} \left(f^*(x) - f(x,\lambda)\right)^2 p(x)dx$$

где p (x) — плотность совместного распределения параметров $x_1$, $x_2$, ..., $x_N$.

В дальнейшем погрешность будем записывать в более общей форме:

$$J(\lambda) = \int_{x \in K} F(x,\lambda)p(x)dx\,, \qquad (4.16)$$

где $F(x,\lambda)$- **функция потерь** или **штрафная функция**. Можно принять

$$F(x,\lambda) = \Phi\left(f^*(x) - f(x,\lambda)\right),$$

где Φ(u) - положительная, монотонно возрастающая функция разности точного и приближенного решений, обращающаяся в нуль при их совпадении.

В общем случае функция потерь должна обращаться в нуль при совпадении точного и приближенного решений и быть положительной при их несовпадении.

Построение разделяющей функции, минимизирующей погрешность приближенного решения, является оптимизацией процесса разделения в пространстве признаков. Однако применение метода минимальной погрешности в его классической форме встречает серьезные затруднения. Часть из них связана с тем, что плотность распределения p(x) обычно неизвестна и имеются только отдельные значения $x_j$, входящие в обучающую последовательность. В такой ситуации оказывается целесообразным применение метода стохастической аппроксимации. Рассмотрим функционал, представляющий собой среднее значение (математическое ожидание) некоторой случайной функции, зависящей от x и вектора коэффициентов $\lambda$:

$$J(\lambda) = \int_{x \in K} F(x,\lambda)p(x)dx = M_x\left[F(x,\lambda)\right],$$

где $M_x$[ ] - знак математического ожидания (усреднения по пространству признаков x), p(x) - плотность вероятности значений x в данной точке пространства. Предположим сначала, что величина p(x) является заданной и тогда условия экстремума функционала будут такими:

$$\frac{\partial J}{\partial \lambda_i} = 0\,,\ i=1,2,...,\ N.$$

Эти условия можно записать в векторной форме:

$$\text{grad}J(\lambda)=\left(\frac{\partial J(\lambda)}{\partial\lambda_1},\frac{\partial J(\lambda)}{\partial\lambda_2},\ldots,\frac{\partial J(\lambda)}{\partial\lambda_N}\right)=0. \qquad (4.17)$$

Условие равенства нулю градиента функционала дает систему уравнений для определения значения вектора $\lambda$, при котором $J(\lambda)$ достигает экстремума. Для пространства большой размерности часто более эффективным является применение итеративных методов (методов последовательных приближений). Представим равенство (4.17) в эквивалентной форме:

$\lambda=\lambda-\|\gamma\|\ \text{grad}\ J(\lambda)$,

где $\|\gamma\|$ - матрица скалярных коэффициентов, детерминант которой отличен от нуля:

$$\|\gamma\|=\begin{Vmatrix}\gamma_{11} & \cdots & \gamma_{1N}\\ \cdots & \cdots & \cdots\\ \gamma_{N1} & \cdots & \gamma_{NN}\end{Vmatrix}.$$

Эта форма записи дает естественный **алгоритм последовательных приближений**:

$$\lambda_{(n+1)}=\lambda_{(n)}-\|\gamma\|_{n+1}\ \text{grad}\ J(\lambda_{(n)}),$$

где $\text{grad}\ J(\lambda)=\int\limits_{x\in K}\text{grad}_\lambda F(x,\lambda)p(x)dx=M_x[\text{grad}_\lambda F(x,\lambda)]$.

Алгоритм последовательных приближений можно обобщить на случай, когда среднее значение (математическое ожидание) $\text{grad}_\lambda F(x,\lambda)$ неизвестно, но известны отдельные его реализации. Они используются как оценки среднего значения, что приводит к следующей процедуре:

$$\lambda_{(n+1)}=\lambda_{(n)}-\|\gamma\|_{n+1}\ \text{grad}_\lambda\ F(x_{(n+1)},\lambda_{(n)}). \qquad (4.18)$$

В этом алгоритме для построения (n+1)-го приближения вектора $\lambda$ необходимо знать предыдущее значение $\lambda$ и значение (реализацию) x на (n+1)-ом шаге. При некоторых условиях, накладываемых на матрицу «стягивающих коэффициентов» $\|\gamma\|$ и grad F, алгоритм последовательных приближений является сходящимся. В дальнейшем ограничимся рассмотрением случая, когда матрица стягивающих коэффициентов пропорциональна единичной матрице I:

$\|\gamma\|_{n+1}=\|\gamma\|_n I$.

В этом случае равенство (4.18) будет таким:

$$\lambda_{(n+1)} = \lambda_{(n)} - \gamma_{n+1} \operatorname{grad}_{\lambda} F(x_{(n+1)}, \lambda_{(n)}),$$

где $\gamma_{n+1}$ - скалярный множитель.

Рассматриваемое ранее условие оптимальности состояло в достижении минимума потерь (4.16) для различных реализаций комплексов признаков x. Это условие недостаточно полно отражает процесс рапознавания, так как не учитывает классы объектов, различную стоимость ошибочных решений. Введем вторую (дискретную) переменную y, описывающую состояние (класс) объекта, и будем считать

$$y = \begin{cases} y_1, \text{для класса } K_1; \\ y_2, \text{для класса } K_2. \end{cases}$$

Часто принимают $y_1 = 1$, $y_2 = -1$.

Искомый весовой вектор $\lambda$ определим из условия минимума функционала

$$J(\lambda) = \iint\limits_{x,y \in K} F(x, y, \lambda) p(x, y) dx dy,$$

где интеграл распространяется на всю область изменения переменных, плотность вероятности по y представляет собой дельтафункцию.

Процедура нахождения весового вектора $\lambda$ сохраняет прежний вид, так как изменение относится только к расширению области осреднения

$$\lambda_{(n+1)} = \lambda_{(n)} - \gamma_{n+1} \operatorname{grad}_{\lambda} F(x_{(n+1)}, y_{(n+1)}, \lambda_{(n)}).$$

В зависимости от выбора штрафной функции $F(x,y,\lambda)$ могут быть получены различные алгоритмы нахождения весового вектора.

## 4.6 Метод потенциальных функций

### 4.6.1 Общая идея метода потенциальных функций

Рассмотрим процесс обучения, при котором ЭВМ предъявляются образы, о которых заранее известно, что они принадлежат к одному из двух классов – либо к А (класс годных в задаче диагностики), либо к В (класс негодных). При этом должно соблюдаться условие разделимости классов. То, что мы ограничиваемся лишь двумя классами, не принципиально: число классов можно наращивать по необходимости.

В начале система предварительной обработки выделяет надлежащим образом выбранные признаки. Набор признаков позволяет построить вектор $x_i$ измерений для i-го образа.

На первом этапе задача состоит в нахождении разделяющей функции, позволяющей, исходя из обучающей выборки, определить границу, разделяющую два класса.

На втором этапе (распознавание неизвестных предъявляемых образов) эта функция используется для классификации.

Идея метода потенциальных функций состоит в том, что каждый экземпляр обучающей выборки представляется точкой в пространстве признаков. Если представить, что в каждой такой точке помещен заряд, то в некоторой произвольной точке совокупность всех зарядов создает потенциал Ф, являющийся суммой отдельных потенциалов $Y_i$, создаваемых каждым отдельным зарядом. Потенциал выражается функцией, симметричной относительно точки, в которой помещен заряд - в этой точке потенциал равен бесконечности. Линии, соединяющие точки равного потенциала (эквипотенциали), будут образовывать рельеф. Облако точек, отображающих некоторый класс будет выглядеть как потенциальное плато, отделенное от другого плато, отображающего другой класс, глубокой долиной, потенциал в которой минимален. Определение минимальной эквипотенциали позволит найти границу раздела между классами.

Назовем потенциальной функцию $Y(x,x_i)$, центрированную относительно $x_i$. Для любой точки x и для любого $x_i$ можно выбрать некоторое $Y(x,x_i)$, имеющее вид

$$Y(x, x_i) = \sum_{k=1}^{\infty} л_k^2 \varphi_k(x) \varphi_k(x_i),$$

где $\lambda_k$, k=1,2,... выбраны такими, чтобы удовлетворялись граничные условия, а функции $\varphi_k(x)$ представляют собой элементы последовательностей ортонормированных функций.

Другой метод создания потенциальных функций состоит в использовании симметрии относительно $x_i$, то есть $Y(x,x_i) = Y(x_i,x)$. Например, можно использовать такие потенциальные функции:

a) $Y(x, x_i) = e^{-нR^2}$,

b) $Y(x, x_i) = \frac{1}{1 + нR^2}$,

c) $Y(x, x_i) = \frac{\sin нR^2}{нR^2}$,

где ν – положительная константа.

Эти функции уменьшаются по мере увеличения расстояния R, где $R^2 = ||x-x_i||^2$.

При выборе конкретной потенциальной функции для практической реализации метода потенциальных функций следует учитывать скорость убывания потенциальной функции при увеличении расстояния $R^2$, регулируемую параметром ν. Чем сложнее и “вычурнее” разделяющая функция, тем быстрее должен убывать потенциал $Y(x,x_i)$.

С увеличением значения коэффициента ν скорость убывания потенциальной функции Y возрастает, а с уменьшением – понижается. Значение коэффициента ν при решении практических задач следует выбирать таким, чтобы скорость убывания потенциальной функции, регулируемая им, обеспечивала бы минимальную ошибку классификации. Подбор оптимального значения ν представляет собой оптимизационную задачу.

Разделяющая функция находится с помощью суммарного потенциала Φ(x), вычисляемого как сумма частных потенциалов $Y(x,x_i)$, связанных с каждым новым предьявляемым i-м экземпляром: $\Phi_{i+1}(x) = \Phi_i(x) + c_{i+1} Y(x, x_{i+1})$.

Корректирующий член $\rho_{i+1}$ удовлетворяет следующим условиям:

$$c_{i+1} = \begin{cases} +1, \text{если } x_{i+1} \in A \text{ и } \Phi_i(x_{i+1}) \le 0, \\ -1, \text{если } x_{i+1} \in B \text{ и } \Phi_i(x_{i+1}) > 0, \\ 0, \text{иначе (при правильной классификации)}. \end{cases}$$

Правильная классификация соответствует случаям, когда $\Phi(x)>0$ при $x \in A$ и $\Phi(x)<0$ при $x \in B$. Поэтому можно использовать $\Phi_i(x)$ как разделяющую функцию, вычисляемую рекуррентно:

$$\Phi_{i+1} = \sum_{k=1}^{\infty} c_k(i+1)\varphi_k(x) = R_{i+1}(x),$$

$$R_{i+1}(x) = R_i(x) + c_{i+1}Y(x, x_{i+1}),$$

где $c_k(i+1)$ – некоторые коэффициенты:

$$c_k(i+1) = c_k(i) + c_{i+1}л_k^2\varphi_k(x_{i+1}).$$

### 4.6.2 Нерекуррентный метод потенциальных функций

Нерекуррентный алгоритм классификации на основе потенциальных функций состоит из следующих шагов.

Шаг 1. Инициализация. Задаются пары наборов значений признаков экземпляров обучающей выборки и сопоставленных им классов соответствующих экземпляров (годный / негодный).

Шаг 2. Параметры нормируются по формуле:

$$x_{j\ норм} = \frac{x_{j\ max} - M(x_j)}{y_{x_j}},$$

где $x_j$ – значение j-го параметра экземпляра обучающей выборки, $M(x_j)$ – математическое ожидание j-го признака, $x_{j\ max}$ – максимальное значение j-го признака, $y_{x_j}$ - среднеквадратическое отклонение j-го признака.

Шаг 3. Производится проверка качества обучения путем распознавания каждого экземпляра, участвующего в обучении. Для этого последовательно для каждого экземпляра обучающей выборки находятся расстояния этого экземпляра от экземпляров класса годных и экземпляров класса негодных:

$$R^2_{A_i} = \sum_{j=1}^{n} \left( x_{j\ норм} - x^{A}_{ij\ норм} \right)^2,\ i=1,2,...,N^A,$$

$$R^2_{B_i} = \sum_{j=1}^{n} \left( x_{j\ норм} - x^{B}_{ij\ норм} \right)^2,\ i=1,2,...,N^B,$$

где n-число признаков, $N^A$- число годных экземпляров, $N^B$- число негодных экземпляров.

Затем находится значение потенциала, создаваемого i-ым экземпляром класса годных обучающей выборки во вновь введенном экземпляре:

$$Y_{A_i} = \frac{1}{1 + \nu R^2_{A_i}},\ i=1,2,...,N^A,$$

где $\nu$ – параметр, регулирующий скорость убывания потенциальной функции $Y_{Ai}$.

Аналогично вычисляется значение потенциала, создаваемого i-ым экземпляром класса негодных обучающей выборки во вновь введенном экземпляре:

$$Y_{B_i} = \frac{1}{1 + нR^2_{B_i}},\ i=1,2,...,N^B.$$

После чего вычисляются средний потенциал класса годных во вновь введенном экземпляре:

$$\Phi_A = \frac{1}{N^A} \sum_{i=1}^{N^A} Y_{A_i}$$

и средний потенциал класса негодных во вновь введенном экземпляре:

$$\Phi_B = \frac{1}{N^B} \sum_{i=1}^{N^B} Y_{B_i}\ .$$

Введенный экземпляр относят к классу годных если $\Phi_A > \Phi_B$, иначе – к классу негодных.

Шаг 4. По заданному критерию (риску потребителя, риску изготовителя или общей ошибке) произвести оптимизацию оператора прогнозирования путем подбора

коэффициента $\nu$ в выражениях для потенциалов. Кроме того, качество прогнозирования можно повысить введением весовых множителей $\gamma_j$ (j=1,2,...,n); $\Sigma\gamma_j=1$ в выражения расстояний.

Шаг 5. После проверки всех экземпляров и оцеки качества обучения приступают к распознаванию экземпляров контрольной выборки, которое осуществляется подобно шагу 3.

### 4.6.3 Модифицированный метод потенциальных функций

Пусть задана обучающая выборка, состоящая из s экземпляров $x^q$, q=1,2,...,s, характеризующихся N признаками $x^q_i$, i=1,2,...,N, q=1,2,...,s, и каждому $x^q$ сопоставлен класс A или B.

Исходя из предположения, что экземпляры одного класса вероятнее всего будут расположены ближе в пространстве признаков, определим для каждого класса и каждого признака координаты центров сосредоточения (центров тяжести) экземпляров.

Координата центра сосредоточения экземпляров, принадлежащих к классу A, по i-му признаку $C^A_{x_i}$ будет определяться из выражения:

$$C^A_{x_i} = \frac{1}{N^A}\sum_{q=1}^{N^A} x_i^q,\ x_i^q \in A,$$

где $N^A$- количество экземпляров, принадлежащих к классу A.

Аналогично, координата центра сосредоточения экземпляров, принадлежащих к классу B, по i-му признаку $C^B_{x_i}$ будет определяться из выражения:

$$C^B_{x_i} = \frac{1}{N^B}\sum_{q=1}^{N^B} x_i^q,\ x_i^q \in B,$$

где $N^B$- количество экземпляров, принадлежащих к классу B.

Зная координаты центров сосредоточения экземпляров обучающей выборки для обоих классов, можно осуществлять классификацию по расстоянию нового экземпляра от этих центров в N-мерной системе координат, где N – число признаков, аналогично методу потенциальных функций.

Для этого для каждого нового экземпляра $x^q$ последовательно находятся расстояния этого экземпляра от центров сосредоточения экземпляров для каждого из классов:

$$R_A^2 = \sum_{j=1}^{N} \left(C_{x_j}^A - x_j^q\right)^2 ,$$

$$R_B^2 = \sum_{j=1}^{N} \left(C_{x_j}^B - x_j^q\right)^2 .$$

После чего сразу можно найти значения суммарных потенциалов, создаваемых всеми экземплярами того или иного класса в новом экземпляре:

$$\Phi_A = \frac{1}{N^A} Y\left(R_A^2\right),$$

$$\Phi_B = \frac{1}{N^B} Y\left(R_B^2\right),$$

где $Y(R^2)$ –некоторая потенциальная функция, например:

$$Y\left(R^2\right) = \frac{1}{1 + нR^2},$$

где $н$-некоторая константа.

Новый экземпляр относят к классу А, если $\Phi_A > \Phi_B$, в противном случае – к классу В.

Так, как потенциальная функция Y для потенциалов обоих классов будет одной и той же, а $R^2_A$ и $R^2_B$ будут неотрицательными величинами, то сравнение потенциалов можно заменить сравнением расстояний: новый экземпляр относят к классу А, если $R^2_A > R^2_B$, в противном случае – к классу В.

Такой алгоритм, по сравнению с методом потенциальных функций, будет обладать рядом преимуществ. Он не будет требовать наличия обучающей выборки в памяти ЭВМ после обучения, будет существенно быстрее работать и позволит избавиться от достаточно большого количества вычислений, то есть будет более оптимальным с вычислительной точки зрения.

## 4.7 Алгоритм многомерной классификации

### 4.7.1 Общая идея алгоритма многомерной классификации

Модифицированный метод потенциальных функций предполагает сравнение расстояний нового экземпляра от центров сосредоточения экземпляров по всем координатам (признакам) сразу. При этом не учитывается информация о значимости признаков и информация об их взаимосвязанности. Для устранения этих недостатков представим класс экземпляра K как функцию суммы результатов частной классификации по i-му и j-му признакам $K_{ij}$ с учетом их значимостей $б_{ij}$.

$K = ш\left(\sum_{i,j} б_{ij} K_{ij}\right)$, где $ш(x) = \begin{cases} 1, x > 0, \\ 0, x \le 0. \end{cases}$

Условимся, что класс A будет кодироваться значением 0, а класс B – значением 1.

Для определения результатов частной классификации по i-му и j-му признакам $K_{ij}$ для экземпляра $x^q$ найдем расстояния этого экземпляра от центров сосредоточения классов на плоскости, образованной i-ым и j-ым признаками:

$$R^2_{C^A_{ij}}(x^q) = \left(C^A_{x_i} - x^q_i\right)^2 + \left(C^A_{x_j} - x^q_j\right)^2,$$

$$R^2_{C^B_{ij}}(x^q) = \left(C^B_{x_i} - x^q_i\right)^2 + \left(C^B_{x_j} - x^q_j\right)^2.$$

Если $R^2_{C^A_{ij}}(x^q) < R^2_{C^B_{ij}}(x^q)$, то будем считать, что на плоскости (i,j) $x^q \in A$, иначе $x^q \in B$.

То есть, если $R^2_{C^A_{ij}}(x^q) < R^2_{C^B_{ij}}(x^q)$, то $K_{ij} = 0$, иначе $K_{ij} = 1$.

Для нахождения значимостей $б_{ij}$ результатов частной классификации по i-му и j-му признакам для всех экземпляров обучающей выборки определим количество ошибочных решений при двумерной классификации по i-му и j-му признакам:

$$N_{ош K_{ij}} = \sum_{q=1}^{s} \left(\left|K^q_{ij} - K^q\right|\right),$$

где $K^q$-значение, сопоставленное классу q-го экземпляра, $K^q_{ij}$-результат двумерной классификации q-го экземпляра по i-му и j-му признакам.

Затем найдем значимости $б_{ij}$ результатов частной классификации по i-му и j-му признакам:

$$б_{ij} = \frac{1 - \frac{N_{ошK_{ij}}}{s}}{\sum_{i,j} 1 - \frac{N_{ошK_{ij}}}{s}}.$$

Для упрощения вычислений можно предложить альтернативный вариант установки значений $б_{ij}$ :

$$б_{ij} = \frac{1}{\sum_{i,j} 1}.$$

Такой вариант несколько ускорит работу алгоритма, но при этом значимости частных результатов классификации учитываться не будут.

Так, как результаты классификации $K_{ij} = K_{ji}$, то для упрощения и ускорения процесса обучения и распознавания зададим области определения для i и j: i, $j \in [1,2,...,N]$, $\forall i \le j$. В этом случае при вычислении результата классификации будут использоваться частные результаты двумерных классисифкаций по i-му и j-му признакам ($i \neq j$) и результаты одномерной классификации по i-му (j-му) признаку (i = j).

Очевидно, если положить: i, j $\in [1,2,...,N]$, $\forall i<j$, то будут учитываться только частные результаты двумерной классификации по i-му и j-му признакам ($i \neq j$).

В свою очередь, если положить: i, j $\in [1,2,...,N]$, $\forall i = j$, то будут учитываться только частные результаты одномерной классификации по i-му (j-му) признаку (i = j).

### 4.7.2 Нейросетевая интерпретация алгоритма многомерной классификации

Вышеописанный алгоритм многомерной классификации может иметь нейросетевую интерпретацию на основе трехслойного перцептрона, являющегося частным случаем МНС.

Для нейросетевой реализации сравнения расстояний и определения значения $K_{ij}$ можно использовать следующее выражение:

$$K_{ij} = ш\left( R^2_{C^B_{ij}}\left(x^q\right) - R^2_{C^A_{ij}}\left(x^q\right)\right),$$

где $ш(x)$ - логистическая функция.

Если функция ш(x) будет дискретной, например, пороговой: $ш(x)=\begin{cases}1, x>0,\\ 0, x\le 0.\end{cases}$, то $K_{ij}$ будет принимать значение 0 или 1. Если функция ш(x) будет вещественной, например, сигмоидной: $ш(x)=\frac{1}{1+e^{-x}}$, то $K_{ij}$ будет принимать значения на интервале [0,1]: чем ближе значение этой функции будет к 0, тем ближе экземпляр будет к классу 0, и, соответственно, наоборот, чем ближе значение этой функции будет к 1, тем ближе экземпляр будет к классу 1. Использование сигмоидной функции может быть более предпочтительным на практике, поскольку она позволяет не только определить к какому классу ближе экземпляр, но и на сколько ближе.

Для вычисления разности расстояний $R^2_{C^B_{ij}}(x^q)-R^2_{C^A_{ij}}(x^q)$ подставим соответствующие выражения:

$$R^2_{C^B_{ij}}(x^q)-R^2_{C^A_{ij}}(x^q)=\left(C^B_{x_i}-x^q_i\right)^2+\left(C^B_{x_j}-x^q_j\right)^2-\left(C^A_{x_i}-x^q_i\right)^2-\left(C^A_{x_j}-x^q_j\right)^2,$$

раскроем скобки, сгруппируем члены по i и j и приведем подобные. После несложных математических преобразований получим:

$$R^2_{C^B_{ij}}(x^q)-R^2_{C^A_{ij}}(x^q)=\tilde{i}+\tilde{j},$$

где $\tilde{i}=\left(C^B_{x_i}\right)^2-\left(C^A_{x_i}\right)^2+2x^q_i\left(C^A_{x_i}-C^B_{x_i}\right)$, $\tilde{j}=\left(C^B_{x_j}\right)^2-\left(C^A_{x_j}\right)^2+2x^q_j\left(C^A_{x_j}-C^B_{x_j}\right)$.

Легко видеть, что выражения для $\tilde{i}$ и $\tilde{j}$ могут быть вычислены на основе формального нейрона, имеющего один вход, на который подается значение $x_i$ или $x_j$, вес которого равен $2\left(C^A_{x_i}-C^B_{x_i}\right)$ или $2\left(C^A_{x_j}-C^B_{x_j}\right)$, соответственно. Порог нейрона (нулевой вес) в этом случае будет равен $\left(C^B_{x_i}\right)^2-\left(C^A_{x_i}\right)^2$ или $\left(C^B_{x_j}\right)^2-\left(C^A_{x_j}\right)^2$, соответственно.

Нейросетевая интерпретация алгоритма многомерной классификации представлена на рис. 4.6.

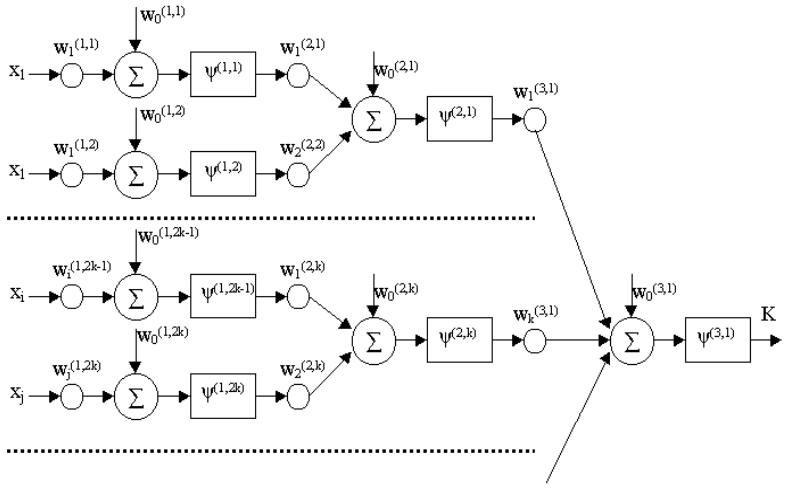



Рис 4.6 - Нейросетевая интерпретация алгоритма многомерной классификации.

Правила вычисления параметров алгоритма многомерной классификации в этом случае останутся неизменными, а параметры и функции активации НС необходимо будет определить на их основе по следующим правилам.

Функция активации $ш^{(м,k)}$ k-го нейрона м-го слоя:

$$ш^{(м,k)}(x) = \frac{1}{1+e^{-x}}, \quad м = 1,2; \forall k$$

$$ш^{(3,1)}(x) = \begin{cases} 0, x \le 0, \\ 1, x > 0, \end{cases}$$

Весовой коэффициент $w_p^{(м,k)}$ p-го входа k-го нейрона м-го слоя:

$$w_p^{(м,k)} = \begin{cases} б_i, м = 3, p = j + (i-1)(N - 0.5i), \\ 0, м = 3, k = 1, p = 0, \\ 0, м = 2, \forall k, p = 0, \\ 1, м = 2, \forall k, p = 1,2, \\ 2\left(C_{x_i}^A - C_{x_i}^B\right), м = 1, k = 2(j + (i-1)(N - 0.5i)) - 1, \ p = i, \\ 2\left(C_{x_j}^A - C_{x_j}^B\right), м = 1, k = 2(j + (i-1)(N - 0.5i)), \ p = j, \\ \left(C_{x_i}^B\right)^2 - \left(C_{x_i}^A\right)^2, м = 1, k = 2(j + (i-1)(N - 0.5i)) - 1, \ p = 0, \\ \left(C_{x_j}^B\right)^2 - \left(C_{x_j}^A\right)^2, м = 1, k = 2(j + (i-1)(N - 0.5i)), \ p = 0, \\ 0, м = 1, k = 2(j + (i-1)(N - 0.5i)) - 1, \ p > 0, p \ne i, \\ 0, м = 1, k = 2(j + (i-1)(N - 0.5i)), \ p > 0, p \ne j, \end{cases}$$

$\forall i, j: \ i = 1,2,....,N; \ j = i,(i+1),...,N$.

## 4.8 Метод дискриминантных функций

Одним из основных результатов, получаемых в процессе обучения распознаванию образов является построение разделяющей функции (поверхности), которая разграничивает выделенные класы $K_\lambda$.

Разделяющие поверхности в любом случае при классификации объектов можно полностью определить скалярными функциями $g_1(x),..., g_m(x)$, где $x=\{x_1, x_2, ..., x_k\}$ - вектор состояния диагностируемого объекта. Эти функции, названные дискриминантными, выбираются так, чтобы для всех $x \in K_\lambda$ выполнялось условие $g_\lambda(x) > g_r(x)$ при $\lambda, r = 1, ..., m, \lambda \ne r$.

Иначе говоря, $\lambda$-я дискриминантная функция в области $K_\lambda$ принимает

наибольшее значение по сравнению с другими дискриминантными функциями.

Если предположить, что дискриминантные функции непрерывны на разделяющих поверхностях, то поверхность, разделяющая смежные классы $K_\lambda$ и $K_r$ определяется уравнением:

$g_\lambda(x) - g_r(x) = 0$

С помощью дискриминантных функций можно получить стандартный и удобный для практического использования метод задания разделяющих поверхностей. В достаточно распространенном на практике случае, когда $\lambda = 2$, при осуществлении классификации необходимо определить, какое из чисел $g_1(x)$ и $g_2(x)$ больше, а это можно сделать, определив знак дискриминантной функции $g(x) = g_1(x) - g_2(x)$. Если $g(x)>0$, то объект принадлежит к классу $K_1$, если $g(x)<0$, то к $K_2$. Соотношение $g(x) = 0$ характеризует в этом случае разделяющую поверхность. Таким образом, при $\lambda=m$ можно найти m-дискриминантные функции $g_1,..., g_m$, при $\lambda =2$ - только одну функцию $g(x)$.

Дискриминантные функции можно выбирать различными способами. Иногда функции точно определяются на основе полной априорной информации об объектах, подлежащих лассификации. В других случаях можно сделать предположения, опираясь на качественные характеристики объектов. В каждом из этих случаев, особенно во втором, может оказаться необходимым корректировать функции для получения приемлемой точности прогнозирования. Адаптация и корректировка нужна и тогда, когда в объект вносятся конструктивные и технологические изменения.

Как уже упоминалось, дискриминантные функции определяются на стадии обучения. При использовании параметрических и непараметрических методов обучения построение дискриминантных функций будет различным.

Параметрические методы целесообразно применять в тех случаях, когда априорно известно, что каждый класс объектов $\lambda$ ($\lambda =1, 2, ..., m$) характеризуется системой параметров, причем значения некоторых из них могут быть неизвестными. При этом обучающая выборка объектов используется для определения параметров, по которым затем находятся дискриминантные функции.

Пусть, например, необходимо разделить объекты на два класса $K_1$ и $K_2$, которые

описываются векторами $x_1$ и $x_2$ с координатами ($x_{11}$, $x_{12}$, ..., $x_{1k}$) и ($x_{21}$, $x_{22}$, ..., $x_{2k}$). Дискриминантную функцию, которая разделяла бы эти два класса, можно представить в виде:

$$g(x) = (x_1 - x_2)\, x + 0{,}5\, |x_2|^2 - 0{,}5\, |x_1|^2$$

где $(x_1 - x_2)\, x$ - скалярное произведение векторов, $|x|^2$ – квадрат модуля вектора x.

Параметрический метод в этом случае используется для определения $x_1$ и $x_2$. Если в $K_1$ находится $N_1$ объектов, а в $K_2$ – $N_2$, то соответствующие средние могут служить разумными оценками $x_1$ и $x_2$. Затем оценку этих средних можно использовать для определения g(x). На этом процесс обучения заканчивается.

Когда же нельзя сделать каких-либо предположений относительно параметров, характеризующих классы, наиболее уместным является применение непараметрических методов обучения. При этом на форму дискриминантных функций накладывают ограничения. Например, предполагают, что эти функции являются либо линейными, либо квадратичными, либо кусочно-линейными. Эти функциональные формы содержат неизвестные коэффициенты, подбираемые таким образом, чтобы дискриминантные функции осуществляли требуемое разделение на обучаемой выборке.

При использовании линейной дискриминантной функции вида

$$g(x) = b_1x_1+b_2x_2+...+b_kx_k$$

необходимо в процессе обучения определить коэффициенты b таким об разом, чтобы величина g(x) могла быть использована для классификации объектов. Обозначим дискриминантные функции для обоих классов $K_1$ и $K_2$ как $g_1(x)$ и $g_2(x)$. Взаимное распределение величин g(x)[$f_1(g)$ для $g_1(x)$ и $f_2(g)$ для $g_2(x)$] позволяет сделать вывод об эффективности дискриминантной функции.

Если распределения значительно перекрывают друг друга, то применение дискриминантной функции не эффективно. Ее целесообразно использовать в случае, когда разность [$g_1(x)$ - $g_2(x)$] велика по сравнений с дисперсиями распределений $f_1(g)$ и $f_2(g)$. Комбинация изменений величин g(x) для обоих классов объектов определяется выражением

$$\sum_{\lambda=1}^{2}\sum_{j=1}^{n_\lambda}\left(g_{\lambda j}-\overline{g}_\lambda\right)^2,$$

где $\overline{g}_\lambda$ - среднее значение распределения λ-го класса, $g_{\lambda j}$ – значения величин g(x) внутри λ-го класса, $n_\lambda$ - количество экземпляров в λ-ом классе.

Наилучшее разделение получается при максимизации выражения:

$$G=\frac{\left(\overline{g}_1-\overline{g}_2\right)^2}{\sum_{\lambda=1}^{2}\sum_{j=1}^{n_\lambda}\left(g_{\lambda j}-\overline{g}_\lambda\right)^2},$$

что приводит к следующим уравнениям:

$b_1 s_{r1} + b_2 s_{r2} + .... + b_m s_{rm} = C\, d_r,$

$C = ( \lambda_1 d_1 + ... + \lambda_m d_m) / G,$

$$s_{pq} = s_{qp} = \sum_{\lambda=1}^{2}\sum_{j=1}^{n_\lambda}\left(x_{p\lambda j}-\overline{x}_{p\lambda}\right)\left(x_{q\lambda j}-\overline{x}_{q\lambda}\right),$$

$$x_{p\lambda} = \frac{1}{n_\lambda}\sum_{j=1}^{n_\lambda} x_{p\lambda j},$$

$$d_p = \overline{x}_{p1} - \overline{x}_{p2}.$$

Индексы p, q, r обозначают специфические параметры и лежат в пределах от 1 до m. Точное значение C несущественно и на практике можно считать C=1. полученная система состоит из m независимых уравнений и позволяет найти m неизвестных b.

При сильном смешивании классов необходим вероятностный подход к построению дискриминантных функций. В этом случае дискриминантная функция будет зависеть от вероятности $P(x/K_\lambda)$ появления объекта x при условии, что он принадлежит к классу $K_\lambda$ и априорной вероятности $P(K_\lambda)$ каждого класса.

В частности, дискриминантные функции можно выразить в виде:

$g_\lambda(x) = P(x/K_\lambda)P(K_\lambda)$, $\lambda=1,2,..,m$

или, что то же самое,

$g_\lambda(x) = \lg P(x/K_\lambda) + \lg P(K_\lambda)$, $\lambda=1,2,..,m.$

При m=2:

$$g(x) = \lg\left(\frac{P(x/K_1)}{P(x/K_2)}\right) + \lg\left(\frac{P(K_1)}{1-P(K_2)}\right).$$

Определение на стадии обучения соответствующих вероятностей позволяет построить вероятностную дискриминантную функцию.

## 4.9 Метод последовательного анализа

Метод последовательного анализа, предложенный Вальдом, применяется для дифференциальной диагностики (распознавания двух состояний). В отличие от метода Байеса, число обследований заранее не устанавливается, их проводится столько, сколько необходимо для принятия решения с определенной степенью риска.

В методе последовательного анализа отношения вероятностей признаков (отношения правдоподобия) составляются не сразу, а в последовательном порядке; поэтому, как правило, требуется меньшее число обследований, чем в методе Байеса.

Будем считать, что признаки являются независимыми. Пусть проведено v - 1 обследований, которые еще не дали возможности принятия решения,

$$B < \frac{P(x_1/K_2)}{P(x_1/K_1)} \cdots \frac{P(x_k/K_2)}{P(x_k/K_1)} < A \text{ , } k = 1,2,..., v\text{-}1,$$

где A и B – верхняя и нижняя границы принятия решения, соотвествтенно, $x_k$ – k-ый признак,

но после v-го обследования

$$\frac{P(x_1/K_2)}{P(x_1/K_1)} \cdots \frac{P(x_k/K_2)}{P(x_k/K_1)} < A .$$

Тогда принимается решение об отнесении объекта к диагнозу $K_2$.

Если после v-го обследования

$$\frac{P(x_1/K_2)}{P(x_1/K_1)} \cdots \frac{P(x_k/K_2)}{P(x_k/K_1)} < B ,$$

то объект относится к диагнозу $K_1$. Для сокращения объема обследований следует вначале проводить обследование по наиболее информативным признакам.

Отметим, что этот метод пригоден и для непрерывно распределенных диагностических параметров, но вместо вероятностей признаков в соотвествующие отношения входят плотности вероятностей параметров.

При распознавании могут быть ошибки двоякого рода. Ошибка, относящаяся к диагнозу D1 (принимается решение о наличии диагноза D2, когда в

действительности объект принадлежит диагнозу D1), называется ошибкой первого рода. Ошибка, относящаяся к диагнозу D2 (принимается решение в пользу диагноза D1, когда справедлив диагноз D2), называется ошибкой второго рода.

Считая состояние D1 исправным, а состояние D2 дефектным, легко понять, что ошибка первого рода является «ложной тревогой», а ошибка второго рода «пропуском дефекта». Обозначим вероятность ошибки первого рода $\alpha$ а, второго рода $\beta$. Допустим, принимается решение в пользу диагноза D2. Вероятность того, что это решение будет справедливым, равна 1-$\beta$. Вероятность принадлежности объекта с данной реализацией признаков к диагнозу D1 составляет $\alpha$. С другой стороны, вероятность диагноза D2, по крайней мере, в А раз больше, чем диагноза D1, т.е. $\frac{1-\beta}{\alpha} \geq A$. Аналогично, $B \geq \frac{\beta}{1-\alpha}$.

В практических расчетах часто принимают $\alpha = \beta = 0{,}05$ или $\alpha = \beta = 0{,}10$.

### 4.10 Метрические методы

В большинстве методов распознавания делается естественное предположение, что изображения объектов одного класса (образа) более близки друг другу, чем изображения разных классов. Метрические методы основаны на количественной оценке этой близости. В качестве изображения объекта принимается точка в пространстве признаков, мерой близости считается расстояние между точками.

**Метрическим пространством** называется множество объектов над которыми определена метрика.

**Метрика** – правило, с помощью которого вводится расстояние d(a,b) между элементами пространства и удовлетворяющее условиям:

1) $d(a,b) \geq 0$, причем d(a,b)=0 тогда и только тогда, когда a=b;

2) d(a,b)=d(b,a);

3) $d(a,b) \leq$ d(a,c) d(b,c).

Рассмотрим наиболее часто используемые метрики.

1. **Расстояние Хэмминга (расстояние по Манхэттену, метрика городских кварталов)** между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k:

$$d = \sum_{s=1}^{k} \left| x_{qs} - x_{gs} \right| .$$

2. **Евклидово расстояние** между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k:.

$$d = \left( \sum_{s=1}^{k} \left( x_{qs} - x_{gs} \right)^2 \right)^{\frac{1}{2}} .$$

3. **Расстояние Минковского** (**обобщенное расстояние**) между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k:

$$d = \left( \sum_{s=1}^{k} \left| x_{qs} - x_{gs} \right|^{v} \right)^{\frac{1}{v}},$$

где v – целое число.

4. **Диагностическая мера расстояния** между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k:

$$d = \left( \sum_{s=1}^{k} \left| x_{qs} - x_{gs} \right|^{v} \right)^{\frac{\mu}{v}},$$

где v, $\mu$ – некоторые числа.

5. **Расстояние в неизотропном пространстве признаков** между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k во многих случаях дает повышенную точность классификации:

$$d = \sum_{s=1}^{k} (\alpha_s)^2 \left( x_{qs} - x_{gs} \right)^2 ,$$

где $\alpha_s$ - коэффициент значимости s-го признака.

Так как для распознавания важен относительный вес, то можно использовать условие нормирования в виде

$$\sum_{s=1}^{k} \alpha_s = 1 .$$

Введение весовых коэффициентов деформирует пространство признаков. Если поставить условие, чтобы при подобных деформациях сохранился объем областей

диагнозов, то условие нормирования можно принять таким:

$$\prod_{s=1}^{k}\alpha_s = 1.$$

6. **Обобщенное расстояние в пространстве признаков**. Выражения для расстояния в неизотропном пространстве устанавливает «неравноправие» отдельных координат в пространстве признаков, но оно не учитывает взаимосвязь координаты $x_s$ и класса $K_i$. Значение признаков различно для различных классов и расстояние точки (экземпляра) $x_q$ до точки $x_g$, принадлежащей к классу $K_i$.

$$d_i = \left( \sum_{s=1}^{k} \alpha_{is}^{v} \left| x_{qs} - x_{gs} \right|^{v} \right)^{\frac{\mu}{v}}$$

где v, $\mu$ – некоторые числа.

Часто оказывается целесообразным принять: $\alpha_{is} = \frac{c_{is}}{\sigma_{is}}$,

где $\sigma_{is}$ - среднеквадратичное отклонение признака (параметра) $x_s$ для экземпляров, относящихся к классу $K_i$., $c_{is}$ - безразмерный коэффициент характеризующий ценность признака.

Для дискретного признака xs, имеющего m дискретных значений $x_{s1}, x_{s2}, \ldots,$ можно принять:

$$c_{is} = \sum_{p=1}^{m_j} P\left(x_{sp} / K_i\right) \log_2 \left( P\left(x_{sp} / D_i\right) P\left(x_{sp}\right) \right).$$

Для непрерывно распределенных признаков $x_s$ вероятность дискретных значений заменяется плотностью вероятности, суммирование - интегрированием по области значений $x_s$. В тех случаях, когда отсутствуют статистические сведения, величины $c_{is}$ могут быть назначены на основании экспертной оценки и т. п.

Условия нормирования:

$$\sum_{s=1}^{k}\alpha_{is} = 1,\ i=1,2\ldots,\ n.$$

$$\prod_{s=1}^{k}\alpha_{is} = 1,\ i=1,2\ldots,\ n.$$

7. **Расстояние в нелинейном пространстве** между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k во многих случаях дает повышенную точность классификации:

$$d = \sum_{s=1}^{k} (\alpha_s)^p \left( (x_{qs})^p - (x_{gs})^p \right),$$

где p- степень нелинейности, предназначенная уменьшить ошибку классификации (обычно p=2,3), $\alpha_s$ - коэффициент значимости s-го признака.

8. **Обобщенное (взвешенное) расстояние Махалонобиаса** между объектами $x_q$ и $x_g$, характеризующимися векторами признаков $\bar{x}_q$ и $\bar{x}_g$:

$$d = \sqrt{(\bar{x}_q - \bar{x}_g)^T \Lambda^T \Sigma^{-1} \Lambda (\bar{x}_q - \bar{x}_g)},$$

где $\Sigma$ - ковариационная матрица генеральной совокупности признаков, $\Lambda$ - некоторая симметричная неотрицательно определенная матрица весовых коэффициентов, которая чаще всего выбирается диагональной.

9. **Расстояние Камберра** между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k:

$$d = \sum_{s=1}^{k} \frac{|x_{qs} - x_{gs}|}{|x_{qs} + x_{gs}|}.$$

10. **Расстояние Чебышева** между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k:

$$d = \operatorname{Max}_s |x_{qs} - x_{gs}|.$$

11. **Расстояние «$\chi^2$»** между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k:

$$d = \sum_{s=1}^{k} \frac{1}{x_{*s}} \left( \frac{x_{qs}}{x_{q*}} - \frac{x_{gs}}{x_{g*}} \right)^2,$$

где $x_{*p} = \sum_{i=1}^{k} x_{ip}$, $x_{p*} = \sum_{i=1}^{k} x_{pi}$.

12. **Скалярные произведения при различных способах нормирования** между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k:

$$d = \sum_{s=1}^{k} x_{qs} x_{gs} \quad \text{или} \quad d = \frac{\sum_{s=1}^{k} x_{qs} x_{gs}}{\sqrt{\sum_{s=1}^{k} x_{qs}^2 \sum_{s=1}^{k} x_{gs}}} \quad \text{или} \quad d = \frac{\sum_{s=1}^{k} x_{qs} x_{gs}}{\sum_{s=1}^{k} (x_{qs} \vee x_{gs})^2_{max}}.$$

13. **Корреляционный метод** нахождения расстояния между объектами $x_q$ и $x_g$, характеризующимися k признаками $x_{qs}$ и $x_{gs}$, s=1,2,...,k:

$$d = \left(\sum_{s=1}^{k} x_{qs} x_{gs}\right) - \frac{1}{k}\left(\sum_{s=1}^{k} x_{qs}\right)\left(\sum_{s=1}^{k} x_{gs}\right).$$

14. **Угловое расстояние.** Близость вектора $x_q$ к эталонному вектору $x_g$ можно охарактеризовать с помощью угла между векторами. Более удобно ввести в рассмотрение косинус угла между векторами, определяя его с помощью скалярного произведения:

$$\cos\gamma = \frac{x_q x_g}{|x_q||x_g|} = \frac{\sum_{s=1}^{k} x_{qs} x_{gs}}{\sqrt{\left(\sum_{s=1}^{k} x_{qs}^2\right)\left(\sum_{s=1}^{k} x_{gs}^2\right)}}.$$

### 4.10.1 Классификация по расстоянию в пространстве признаков

Рассматриваемые методы подразделяются на две группы: классификация по расстоянию до эталона и по расстоянию до множества.

В **методе классификации по расстоянию до эталона (методе эталонов)** отнесение предъявленного для распознавания объекта к одному из n классов совершается по наименьшему расстоянию до эталона. В качестве эталона для класса $K_i$ принимается типичный объект, относящийся к классу $K_i$. Наиболее естественный выбор эталона состоит в использовании средних значений параметров в области класса. Если известны $M_i$ объектов, относящихся к классу $K_i$, то в качестве эталона класса $K_i$ можно принять:

$$x_i^* = \frac{1}{M_i}\sum_{g=1}^{M_i} x_g ,$$

где $x_g$ – экземпляр, относящийся к классу $K_i$, (экземпляр с верифицированным диагнозом). Это равенство определяет эталон как центр тяжести области класса. Координаты вектора $x_i^*$ равны средним значениям координат векторов, входящих в обучающую последовательность.

Допустим, что в пространстве признаков используется диагностическая мера расстояния L и предъявлен для диагностики объект $x_q$. Для отнесения объекта $x_q$ к одному из n диагнозов определяются расстояния L до эталонных точек $x_1^*$, $x_2^*$, . . .,

$x_n^*$. Объект $x_q$ относят к классу $K_i$, если мера расстояния между точками $x_q$, $x_i^*$ минимальна:

если $L_i = \min$, то $x_q \in K_i$ или в другой форме

$x_q \in K_i$, если $L_i < L_s$ $s= 1, 2, \ldots, n$; $s \neq i$.

В некоторых случаях это условие принимается в более строгой форме

$L_i - L_s > \varepsilon$,

где $\varepsilon$ — порог распознавания ($\varepsilon > 0$).

**Классификация по расстоянию до множества** может осуществляться на основе следующих методов.

**Метод среднего расстояния.** В этом методе оценивается расстояние не от одной точки — эталона, а расстояния от точки $x_q$ (объекта, предъявленного для распознавания) до всех точек множества, относящихся к данному классу. Расстояние до множества оценивается как среднее расстояние, но возможны и другие способы этой оценки.

Пусть для класса $K_i$ группа экземпляров, относящихся к данному классу, содержит $M_i$ образцов. Допустим, что выбрана диагностическая мера расстояния, и тогда расстояние от точки $x_q$ до точки $x_g$, входящей в группу верифицированных образцов (при квадратичной мере),

$$L_{ig} = \sum_{s=1}^{k} \alpha_{is}^2 \left(x_{qs} - x_{gs}\right)^2 .$$

Можно определить среднее расстояние от точки x до точек обучающей последовательности, принадлежащей классу $K_i$:

$$L_i = \frac{1}{M_i} \sum_{g=1}^{M_i} L_{ig} = \frac{1}{M_i} \sum_{g=1}^{M_i} \sum_{s=1}^{k} \alpha_{is}^2 \left(x_{qs} - x_{gs}\right)^2$$

Процедура классификации остается такой же, как при определении расстояния до эталона. Принимается решение $x_q \in K_i$,если $L_i<L_s$ ($s= 1, 2,\ldots,k$; $s \neq i$), или, что эквивалентно, $x_q \in K_i$,если $L_i = \min$.

**Метод минимального расстояния до множества.** Ранее использовалось «среднее расстояние» до точек диагноза. Возьмем теперь в качестве расстояния до множества минимальное расстояние среди всех расстояний от точки $x_q$ до точек, входящих в группу класса $K_i$:

$$L_i = \min_{x_{gi} \in K_i} L_i^g .$$

Алгоритм распознавания состоит в следующем. Определяется расстояние от точки xq (объекта, предъявленного для классификации) до всех точек, входящих в область данного класса (точки обучающей группы) и запоминается минимальное расстояние. Принимается решение

$$x_q \in K_i \text{ , если } L_i = \min\left( \min_{x_{gs} \in K_s} L_s^g \right), \; s=1,2,....,k; \; s \neq i,$$

где $L_s = \min_{x_{gs} \in K_s} L_s^g$ - минимальное расстояние до точек класса $K_s$.

Таким образом, решение здесь принимается по близости к прецеденту, а не ко всей совокупности случаев с данным диагнозом.

**Алгоритм максимина** имеет целью отыскивание представительных элементов каждого класса исходя из произвольного выбора.

Пусть каждый экземпляр $x_i$, характеризуется k признаками $x_{si}$, s=1,...,k. Алгоритм максимина состоит из нескольких этапов.

Шаг 1. Выбор начального ядра: произвольно выбираем первый экземпляр $N_1 = x_1$ из множества экземпляров $x = \{x_1, x_2, ...,x_v\}$, где v – количество экземпляров. Затем следует определить другие ядра $N_2$, $N_3$, ..., $N_m$, число m которых заранее неизвестно.

Находим такой элемент, который удален из N1 на наибольшее расстояние (в метрике, выбранной из условий удобства вычислений и простоты реализации).

Шаг 2. Вычисление расстояний $d_{1i}(N_1, x_i) \; \forall i \neq 1$.

Шаг 3. Определение максимального расстояния $d_{1q}$, по которому можно найти ядро $N_2$:

$N_2=x_q$, где $d_{1q} = \text{Max } d_{1i}(N_1 x_i)$.

Шаг 4. Вычисление расстояний между имеющимися ядрами и оставшимися точками

$d_{si} = d(N_s \; x_i)$. s=l, 2; i-= 1, 2, ..., v, среди которых находится наименьшее из них $\delta_{si} = \text{Min } (d_{si})$, s = 1, 2.

Шаг 5. Поиск максимального среди всех $\delta_{si}$ значения, которое обозначим через

$D_{sp}$ (отсюда становится понятным название алгоритма: максимум среди минимумов). Если это расстояние больше половины расстояния, разделяющего $N_1$ и $N_2$, то создают новое ядро, дополнительное к имеющимся:

$N_3=x_p$, такое, что $D_{sp} = Max\ \delta_{si}$, s = 1, 2, . . ., где $D_{sp} \geq D_{12}$.

Шаг 6. Повторение процедуры с новым ядром. Изменяется только предыдущий этап за счет того, что сравнение производится с половиной средней величины расстояний между ядрами. Процедура заканчивается, если все максимальные значения минимальных расстояний ниже этого порога. К этому моменту выявляется число классов q и их ядра, представляющие соответственно каждый класс: $N_1$, $N_2$, ..., $N_q$.

Процедура прекращается, когда число классов (а значит, и ядер) остается неизменным.

**Алгоритм К средних** отличается от алгоритма максимина начальными условиями и числом центров. Обычно используют К первых элементов из списка данных или К первых сигналов от наблюдаемых объектов. Процедура состоит из следующих операций.

Фиксируют число ядер К.

Выбирают первые элементы из каждого ядра:

$N_{11}$; ...; $N_{s1}$.

Каждому из этих векторов приводят в соответствие свою область пространства. Формируют эти области, связывая векторы измерений с К ядрами согласно правилу минимального расстояния. На r-ом этапе вектор $x_p$ связывают с ядром $N_{ir}$, если удовлетворяется следующее неравенство:

$\| x_p - N_{ir} \| < \| x_p - N_{jr} \|, \ \forall j \neq i$,

тогда $x_p$ принадлежит к области $N^*_{ir}$. Таким образом, на r-ом этапе $N^*_{ir}$ представляет область, связанную с ядром $N_{ir}$. Определяют новые элементы, характеризующие новые ядра $N_{i(r+1)}$. За их значение принимают x, обеспечивающее минимум среднеквадратичного отклонения:

$$J_i = \sum_{x_p \in N^*_{ir}} \left\| x_p - N_{i(r+1)} \right\|^2 , \ i=1,2,...,K.$$

Действительно, $J_i$ принимает минимальное значение лишь при одном x, равном среднему арифметическому векторов, принадлежащих одной области $N^*_i$. Тогда новое ядро будет:

$$N^*_{i(r+1)} = \frac{1}{\text{Card } N^*_i} \sum_{x_p \in N^*_i} x_p \text{ , } i=1,2,...,K.$$

Видно, что для нового определения необходимо вычислить К средних значений (отсюда и происходит название этого алгоритма).

Процедура заканчивается, если положение центров $N_{i(r+1)}$ по отношению к положению предыдущих центров $N_{ir}$ не изменяется. В противном случае ее повторяют заново, начиная с третьего этапа, путем формирования новых областей вокруг новых центров.

**Алгоритм ISODATA (**в пер. с англ. «Итеративная самоорганизующаяся система анализа данных») представляет собой вполне определенную, гибкую последовательность операций. Их итеративное выполнение приводит к тому, что основные элементы классификации вырабатываются непосредственно в процессе работы. В частности, это относится и к числу ядер, количество которых априори не было определено. Алгоритм состоит из следующих этапов. Исходное расположение центров выбирают произвольно. Опыт показывает, что окончательный результат почти не зависит от первоначального выбора. Определяют области, в которые входят точки, близкие (в ев клидовом геометрическом смысле) к исходным центрам.

Делят на две каждую группу, внутри которой среднее расстояние между точками превышает порог $\theta_1$.

Определяют новые «средние» точки каждой области с учетом вновь появившихся областей. Вычисляют расстояния между каждой парой средних точек. Объединяют области, связанные со средними точками, расстояние между которыми меньше некоторого порога $\theta_2$.

Выполняют процедуру заново.

Существенное отличие алгоритма ISODATA состоит в том, что здесь нет необходимости вычислять все расстояния между каждой парой точек на втором этапе (определение областей). Оператор может произвольно выбирать значения

порогов $\theta_1$ и $\theta_2$. Область применения этого алгоритма не ограничивается только распознаванием образов. Благодаря тому что с его помощью можно легко обрабатывать большие массивы исходных данных, его используют также в метеорологии, социологии и других областях.

### 4.10.2 Классификация по принадлежности экземпляра к области класса

Метрические методы относят экземпляр к тому классу, расстояние к центру которого меньше. Такое разделение будет безошибочным, если классы хорошо разделяются и имеют сферическую группировку вокруг центров классов.

В ситуации же, изображенной на рис. 4.7 экземпляры классов A и B сгруппированы вокруг центров классов, но области этих группировок вытянутее.

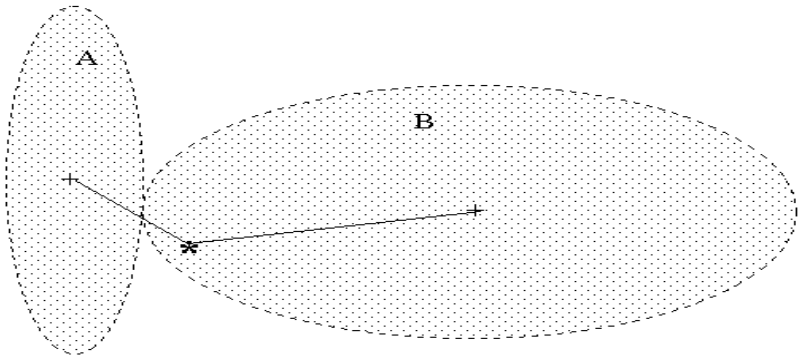



Рис. 4.7 - Частный случай ошибочной классификации на основе метричеких методов.

В этом случае экземпляр, принадлежащий классу B (обозначен “*”), метрическая классификация будет ошибочно относить к классу A, поскольку по расстоянию он ближе к A.

Однако, легко видеть, что данный экземпляр находится внутри области, граница которой может быть аппроксимирована по обучающим данным и этот экземпляр можно было бы вполне уверенно отнести к классу B, если бы классификация осуществлялась по принадлежности экземпляра к области класса

(фигуре, ограниченной некоторой поверхностью, аппроксимирующей границу области класса по точечным данным обучающего эксперимента).

Пусть в k-мерном признаковом пространстве имеется односвязная область с центром в точке C, ограниченная поверхностью L(x), тогда точка $x_q$ будет находиться на поверхности L(x), если $L(x_q) = 0$; будет находиться внутри области с центром в точке C, если $L(x)<0$; и вне области с центром в точке C, если $L(x)>0$.

Примем, что $x_q$ принадлежит области C, если $L(x_q) \le 0$. Тогда, если известны уравнения гиперплоскостей $L^A(x)$ и $L^B(x)$, аппроксимирующих границы областей классов $C^A$ и $C^B$, то

$x_q \in C^A$, если $L^A(x_q) \le 0$, $L^B(x_q)>0$;

$x_q \in C^B$, если $L^B(x_q) \le 0$, $L^A(x_q)>0$;

$x_q \notin C^A$, $x_q \notin C^B$, если ($L^A(x_q) >0$ и $L^B(x_q)>0$) или ($L^A(x_q) <0$ и $L^B(x_q)<0$) или ($L^A(x_q) = 0$ и $L^B(x_q) = 0$).

Такие правила будем называть уверенной классификацией, поскольку они однозначно определяют принадлежность экземпляра.

Выражения гиперповерхностей границ областей классов, как правило, неизвестны. Поэтому их необходимо аппроксимировать на основе точечных значений признаков экземпляров обучающей выборки, принадлежащих соответствующим классам.

Кроме того, на практике известны на все возможные экземпляры, принадлежащие $C^A$ и $C^B$, а лишь некоторое небольшое подмножество таких экземпляров (обучающая выборка). Поэтому $L^A$ и $L^B$, построенные на основе обучающей выборки, будут ограничивать не области классов для генеральной совокупности экземпляров, а области классов для экземпляров обучающей выборки и обеспечивать уверенную классификацию только в этих областях. Вне данных областей классификация на основе вышеописанных правил будет невозможна (отказ от классификации).

Если ввести приоритеты для классов, определив, что экземпляр $x_q$ относится к классу A, если $L^A(x_q) \le 0$ и к классу B – в противном случае, то это правило позволит осуществлять классификацию во всем признаковом пространстве, но при этом

определенная часть экземпляров, которые принадлежат к классу A, но не попадают в область $C^A$ (т.е. не входят в обучающую выборку) могут быть ошибочно отнесены к классу B.

Выбор выражения для $L^p(x)$, где p={A,B}– обозначение класса, производится с учетом предварительной информации о характере сосредоточения экземпляров.

Например, для $L^p(x)$ можно использовать уравнение гиперэллипсоида ( частным случаем которого является гиперсфера):

$$L^p(x) = \sum_{i=1}^{k} \frac{(x_i)^2}{(\alpha_i^p)^2} - 1,$$

где $\alpha_i^p$- некоторые константы, определяющие полурадиусы эллипсоида. Константы $\alpha_i^p$ следует находить по данным обучающей выборки, например, по такому правилу:

$$(\alpha_i^p)^2 = \frac{Max(x_{qi}) - Min(x_{qi})}{2}, \forall x_q \in C^p.$$

Так, как центры эллипосидов, аппроксимирующих границы областей классов, находятся не в начале координат, а смещенны в центры сосредоточения классов, то в $L^p(x)$ необходимо учесть это смещение, которое по i-ой координате будет равно i-ой координате центра p-ой области $C^p_i$:

$$L^p(x) = \sum_{i=1}^{N} \frac{(x_i - C_i^p)^2}{(\alpha_i^p)^2} - 1.$$

### 4.10.3 Комбинированнный метод классификации

Из вышесказанного следует, что если новый экземпляр попадет в области, где имеются экземпляры обучающей выборки, то для него целесообразно проводить классификацию по принадлежности к области и такая классификация будет уверенной. Если же экземпляр находится вне областей, где имеются экземпляры обучающей выборки или на пересечении областей, то для него целесообразно применять метрическую классификацию, которая будет неуверенной, поскольку не сможет гарантировать, что экземпляр в действительности принадлежит данному

классу. Это приводит к следующим алгоритмам комбинированного метода классификации.

**Обучение.**

Шаг 1. Инициализация. Задается вид $L^A(x)$, $L^B(x)$, а также метрика d.

Шаг 2. Для каждого i-го признака экземпляров p-го класса находятся:

а) координаты центров сосредоточения экземпляров (центров областей):

$$C_i^p = \frac{1}{N_p}\sum_{q=1}^{N_p} x_{qi}, \forall x_q \in C^p ,$$

где $N_p$ – количество экземпляров обучающей выборки, принадлежащих к области $C^p$.

б) минимальное и максимальное значения i-го признака p-го класса $Max(x_i)$, $Min(x_i), \forall x \in C^p$.

Шаг 3. Для всех экземпляров $x_q$ из обучающей выборки определяются коэффициенты корреляции $r_{x_i,p^*}$ между i-ым признаком и номером класса p* (p* = 0 - для класса A и p*=1 – для класса B).

Шаг 4. На основе $C^p_i$, $Min(x_i)$, $Max(x_i)$, i=1,..,k, $\forall x \in C^p$ определяются параметры $L^p(x)$.

Шаг 5. На основе $r_{x_i,p^*}$ находятся коэффициенты метрики, учитывающие значимость признаков:

$$\beta_i = \frac{\left|r_{x_i,p^*}\right|}{\sum_{i=1}^{N}\left|r_{x_i,p^*}\right|} .$$

**Распознавание.**

Шаг 1. Предъявляется экземпляр $x_q$.

Шаг 2. Вычисляются $L^A(x_q)$ и $L^B(x_q)$.

Шаг 3. Если $L^A(x_q) \le 0$, $L^B(x_q)>0$, то $x_q \in C^A$ и переход на шаг 7.

Шаг 4. Если $L^B(x_q) \le 0$, $L^A(x_q)>0$, то $x_q \in C^B$ и переход на шаг 7.

Шаг 5. Вычисляются $d(x_q, C^A)$ и $d(x_q, C^B)$ – расстояния от экземпляра $x_q$ до центров соотвествующих классов.

Шаг 6. Если $d(x_q, C^A) > d(x_q, C^B)$, то $x_q \in C^A$, иначе - $x_q \in C^B$.

Шаг 7. Останов.

## 4.11 Алгоритм классификации с оценкой значимости признаков

Признаки, характеризующие моделируемый объект (процесс), будем условно делить на значимые и незначимые. К значимым будем относить те признаки, которые тесно связаны с выходным параметром и пренебрежение которыми может существенно ухудшить модель, к незначимым - те признаки, которые слабо связаны (или не вообще не связаны) с выходным параметром и пренебрежение которыми не ухудшает модель, либо ухудшает, но не намного.

Традиционно для оценки степени связи параметров используют различные критерии, наиболее известным представителем которых является коэффициент корреляции. Однако коэффициент корреляции в основном применим лишь для оценки взаимосвязи вещественных параметров, в то время как во многих задачах диагностики необходимо получать оценку взаимосвязи вещественного (признак) и дискретного (номер класса) параметров. С другой стороны, при построении диагностических моделей желательно заранее знать, сколько потребуется разделяющих плоскостей для осуществления классификации, что нельзя оценить, исходя из коэффициента корреляции.

Объединяя вышеизложенные соображения, в качестве меры связи признака и выходного параметра (меры влияния признака на выходной параметр) будем использовать количество интервалов, на которые разбивается диапазон значений признака, таких, что экземпляры, со значением признака, попавшим в один интервал, относятся к одному и тому же классу, а экземпляры смежных интервалов относятся к разным классам. Очевидно, что такая мера позволит не только оценить значимость признака (чем меньше количество интервалов, тем больше значимость и наоборот), но и оценить необходимое количество разделяющих прямых для классификации по данному признаку. Одновременно с оценкой значимости признаков представляется возможным для каждого интервала найти граничные значения признака для экземпляров обучающей выборки, которые можно будет использовать при классификации.

Обобщая вышеизложенное, сформулируем **алгоритм оценки значимости признаков и расчета параметров решающего правила**.

Шаг 1. Инициализация. Задать обучающую выборку экземпляров, представленную в виде массива данных p, в котором признаки линеаризованы по строкам, а экземпляры - по столбцам, а также соответствующий массив t, содержащий номера классов, сопоставленные экземплярам обучающей выборки (0 или 1). Создать массив nx, равный по размеру количеству признаков N, элементы которого будут содержать число интервалов для каждого признака. Установить nx(i) = 0, i = 1,...,N, где i - номер текущего признака. Занести количество экземпляров обучающей выборки в переменную S. Установить номер текущего признака i=1.

Шаг 2. Если $i \le N$, тогда перейти на шаг 3, в противном случае - перейти на шаг 12.

Шаг 3. Занести в буфер признака x вектор значений i-го признака из обучающей выборки: x(j) = p(i,j), j=1,...,S; занести в буфер класса y копию массива t: y(j) = t(j), j=1,...,S.

Шаг 4. Отсортировать массивы x и y в порядке возрастания массива x (шаги 4.1-4.7 реализуют простейший алгоритм пузырьковой сортировки, который можно заменить на практике более быстродействующим алгоритмом).

Шаг 4.1 Установить номер текущего экземпляра обучающей выборки j=1.

Шаг 4.2 Если $j \le S$, тогда перейти на шаг 4.3, в противном случае - перейти на шаг 5.

Шаг 4.3 Установить номер текущего экземпляра: k = j+1.

Шаг 4.4 Если $k \le S$, тогда перейти на шаг 4.5, в противном случае - перейти на шаг 4.7.

Шаг 4.5 Если x(j)>x(k), тогда установить: tmpx = x(j), x(j) = x(k), x(k) = tmpx, tmpy = y(j), y(j) = y(k), y(k) = tmpy, где tmpx и tmpy - буферные переменные.

Шаг 4.6 Установить: k=k+1. Перейти на шаг 4.4.

Шаг 4.7 Установить: j=j+1. Перейти на шаг 4.2.

Шаг 5. Установить: s = 1, k = 1.

Шаг 6. Если $s \le S$, тогда установить tempa=x(s), где tempa - буфер для хранения левой границы k-го интервала i-го признака, и перейти на шаг 7, в противном случае - перейти на шаг 11.

Шаг 7. Пока (s<S)и(y(s) = y(s+1)) выполнять: s=s+1.

Шаг 8. Если (s = S)и(y(s) = y(s-1), тогда установить: Kx(i,k) = y(s), Ax(i,k) = tempa, Bx(i,k) = x(s), k = k+1, s = s+1, перейти на шаг 10. Здесь Kx(i,k) - номер класса сопоставленный экземплярам обучающей выборки, значение i-го признака которых попадает внуть k-го интервала; Ax(i,k) и Bx(i,k) - левая и правая границы k-го интервала i-го признака, соотвественно.

Шаг 9. Если (s < S)и($y(s) \neq y(s+1)$), тогда установить: Kx(i,k) = y(s), Ax(i,k) = tempa, Bx(i,k) = x(s), k = k+1, s = s+1, nx(i) = nx(i)+1, в противном случае - установить: Kx(i,k) = y(s), Ax(i,k) = x(s), Bx(i,k) = x(s), k = k+1, s = s+1.

Шаг 10 Перейти на шаг 6.

Шаг 11 Установить: i=i+1, перейти на шаг 2.

Шаг 12 Останов.

В результате выполнения шагов 1- 12 для обучающей пары {p,t} мы получим массив nx, содержащий для каждого признака количество интервалов на которые он разбивается (для оценки информативности признаков необходимо принять NNx(i) = min(nx)/nx(i), i=1,...,N), а также массивы Ax, Bx и Kx, содержащие информацию о границах интервалов и номерах классов, сопоставленных им для всех признаков. На основе этих массивов будем осуществлять классификацию.

Одномерную классификацию по i-му признаку будем осуществлять следующим образом. Найдем интервал, в который попадает значение признака и отнесем экземпляр по данному признаку к классу, номер которого сопоставлен интервалу, в который попало значение признака. Если значение признака не попадает ни в один интервал из определенных в массивах Ax и Bx, тогда отнесем экземпляр по данному признаку к классу, сопоставленному экземплярам ближайшего интервала.

Произведя одномерные классификации экземпляра по всем признакам, отобразим результаты классификаций с интервала [0,1] на интервал [-1,1] и найдем их сумму, взвешенную с помощью коэффициентов NNx(i). Очевидно, что

результаты одномерной классификации для значимых признаков в этом случае будут вносить больший вклад в сумму, чем результаты классификации по малозначимым признакам. Рассчитав взвешенную сумму, отобразим ее на интервал [0,1], что и будет итоговым результатом классификации экземпляра по всем признакам.

Для оценки относительной надежности классификации для нескольких экземпляров разделим модуль взвешенной суммы результатов классификации (без отображения на интервал [0,1]) на максимальное по модулю значение взвешенной суммы для данных экземпляров.

Обобщая вышесказанное, запишем **алгоритм классификации**.

Шаг 1. Инициализация. Задать массивы p, nx, Ax, Bx и Kx.

Шаг 2. Найти оценки значимости для признаков:

NNx(i) = min(nx(i))/nx(i), i=1,...,N.

Шаг 3. Установить номер текущего экземпляра j=1.

Шаг 4. Если $j \le S$, тогда перейти на шаг 5, в противном случае - перейти на шаг 15

Шаг 5. Установить значение взвешенной суммы результатов одномерных классификаций j-го экземпляра rj = 0, номер текущего признака j-го экземпляра: i=1.

Шаг 6. Если $i \le N$, тогда перейти на шаг 7, в противном случае - перейти на шаг 12

Шаг 7. Установить результат классификации для j-го экземпляра по i-му признаку r(i)=0;

Шаг 8 Определить интервал, к которому относится j-ый экземпляр по i-му признаку и номер класса, сопоставленный данному интервалу.

Шаг 8.1 Если p(i,j)<Ax(i,1), тогда r(i)=Kx(i,1), перейти на шаг 9, в противном случае - перейти на шаг 8.2.

Шаг 8.2 Если p(i,j)>Bx(i,nx(i)+1), тогда установить r(i)=Kx(i,nx(i)+1) и перейти на шаг 9, в противном случае - перейти на шаг 8.3.

Шаг 8.3 Установить: k=1.

Шаг 8.4 Если k ≤ nx(i)+1, тогда перейти на шаг 8.5, в противном случае - перейти на шаг 9.

Шаг 8.5 Если (p(i,j)≥Ax(i,k))и(p(i,j) ≤Bx(i,k)), тогда установить r(i)=Kx(i,k) и перейти на шаг 9, в противном случае - перейти на шаг 8.6.

Шаг 8.6 Если (k<nx(i)+1) и (p(i,j)>Bx(i,k)) и (p(i,j)<Ax(i,k+1)), тогда перейти на шаг 8.7, в противном случае - перейти на шаг 8.8.

Шаг 8.7 Если (Ax(i,k+1)-p(i,j))<(p(i,j)-Ax(i,k+1)), тогда установить r(i) = Kx(i,k+1), в противном случае - установить r(i) = Kx(i,k).

Шаг 8.8 Установить: k=k+1, перейти на шаг 8.4

Шаг 9. Если r(i)>0, тогда установить r(i) = 1, в противном случае установить r(i) = -1.

Шаг 10. Установить: rj = rj+r(i)NNx(i).

Шаг 11. i=i+1, перейти на шаг 6.

Шаг 12 Установить pr(j)=rj, где pr(j) -массив, содержащий оценки относительной уверенности (надежности) классификации.

Шаг 13. Если rj>0, тогда установить t(j)=1, в противном случае - установить t(j)=0. Здесь t - массив результатов классификации.

Шаг 14. Установить: j=j+1, перейти на шаг 4.

Шаг 15. Установить: pr = |pr|/max(|pr|).

Рассмотренный алгоритм классификации является неитеративным и может быть использован для построения и настройки весов нейронной сети прямого распространения - многослойного персептрона.

Для этого функции активации для всех нейронов сети следует задать как:

$$\psi(x)=\begin{cases}0, x \le 0;\\ 1, x > 0.\end{cases}$$

Весовой коэффициент q-го входа ρ-го нейрона μ-го слоя установить в соответствии с формулой:

$$w_q^{(\mu,\rho)} = \begin{cases} NNx\ (i), \mu = 6, \rho = 1, q = 2i - 1; \\ -NNx\ (i), \mu = 6, \rho = 1, q = 2i; \\ 0, \mu = 6, \rho = 1, q = 0; \\ 0, \mu = 5, \forall \rho, q = 0; \\ 1, \mu = 5, \rho = 2i - 1, q = 1; \\ -1, \mu = 5, \rho = 2i, q = 1; \\ 0, \mu = 4, \forall \rho, q = 0; \\ 1, \mu = 4, \forall \rho, q > 0; \\ 1, \mu = 3, \forall \rho, q = 1; \\ Kx\ (i,k), \mu = 3, \forall \rho, q = 0; \\ -1, \mu = 2, \forall \rho, q = 0; \\ 1, \mu = 2, \forall \rho, q > 0; \\ 1, \mu = 1, \rho = 2i - 1, q = 1; \\ -1, \mu = 1, \rho = 2i, q = 1; \\ -Ax\ (i,k), \mu = 1, \rho = 2i - 1, q = 0; \\ Bx\ (i,k), \mu = 1, \rho = 2i, q = 0. \end{cases}$$

k=1,..,nx(i); i=1,...,N.

Схема персептрона, веса которого настроены по предложенной формуле представлена на рис. 4.8.

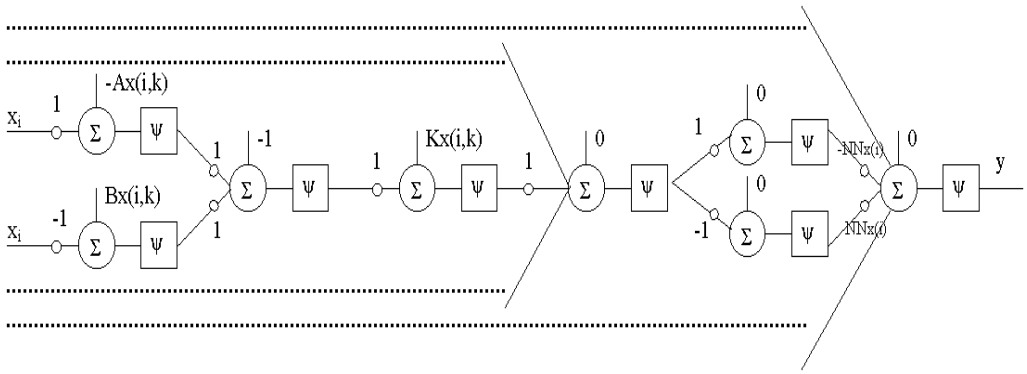



Рис. 4.8 - Схема персептрона.

Такая схема будет обеспечивать только жесткую классификацию, т.е. экземпляр может быть классифицирован только в том случае, если значения всех его признаков попадают в интервалы, значения классов для которых сопоставлены алгоритмом рачета параметров решающего правила.

Для того, чтобы осуществлять более гибкую классификацию перед созданием ее нейросетевой реализации для каждого признака следует найти все те интервалы,

которым не сопоставлены номера классов, разделить эти интервалы пополам и изменить значения правой границы предыдущего и левой границы последующего смежных интервалов, которым сопоставлены номера классов, чтобы поглотить этот интервал. Например, если правая граница предыдущего интервала равна b, а левая граница последующего интервала равна a, тогда установить: b=b+(a-b)/2, a=a-(a-b)/2. После чего произвести построение нейросетевой реализации.

# ГЛАВА 5. НЕЙРОСЕТЕВЫЕ МЕТОДЫ РАСПОЗНАВАНИЯ И АППРОКСИМАЦИИ

## 5.1 Принципы организации и классификация нейронных сетей

При построениии адаптивных систем диагностики перспективным является использование искусственных нейронных сетей (НС), которые обладают такими свойствами, как обучаемость, универсальность и способность аппроксимировать любые вычислимые функции. Это позволяет использовать их для классификации, оценки значений параметров и построения математических моделей сложных процессов и объектов даже в тех случаях, когда другими способами это сделать затруднительно.

Кроме вышеперечисленных достоинств НС характеризуются высокой надежностью и устойчивостью к негативным внешним воздействиям (высокими робастными свойствами), обладают способностями самостоятельно извлекать знания из данных в процессе обучения, а также способны самостоятельно решать некоторые оптимизационные задачи.

Все модели искусственных НС представляют собой множество нейронов (или нейроподобных элементов), связанных определенным способом между собой. Основными отличиями моделей НС являются способы связи нейронов между собой, механизмы и направления распространения сигналов по сети, а также ограничения на используемые функции активации.

Одно из важнейших свойств НС - способность к самоорганизации и самоадаптации с целью улучшения качества функционирования. Это достигается обучением НС, алгоритм которого задается набором обучающих правил. Обучающие правила определяют, каким образом изменяются связи в ответ на входное воздействие. Обучение основано на увеличении силы связи (веса синапса) между одновременно активными нейронами. Таким образом, часто используемые связи усиливаются, что объясняет феномен обучения путем повторения и привыкания.

В настоящее время не существует единой стандартной классификации НС, поскольку нейроинформатика является относительно новой областью науки и

терминология здесь еще не устоялась. Поэтому рассмотрим классификацию НС только по некоторым базовым характеристикам.

В зависимости от типа функции активации НС подразделяют на дискретные, вещественные (непрерывные) и дискретно-непрерывные.

В зависимости от направления распространения сигналов НС подразделяют на сети прямого распространения, сети обратного распространения и двунаправленные НС.

В зависимости от количества и структуры связей НС подразделяют на полносвязные (все нейроны связаны со всеми) и неполносвязные.

В зависимости от количества слоев нейронов НС подразделяют на однослойные и многослойные. Иногда особо выделяют двуслойные и трехслойные НС.

## 5.2 Формальный нейрон. Однослойный персептрон

Нервная система человека состоит из клеток, называемых нейронами, и имеет ошеломляющую сложность: около $10^{11}$ нейронов участвуют в порядка $10^{15}$ передающих связях, имеющих длину метр и более. Каждый нейрон обладает многими качествами, общими с другими клетками, но его уникальной способностью является прием, обработка и передача электрохимических сигналов по нервным путям, которые образуют коммуникационную систему мозга.

На рис. 5.1 показана структура биологического нейрона. Дендриты идут от тела нервной клетки к другим нейронам, где они принимают сигналы в точках соединения, называемых синапсами.

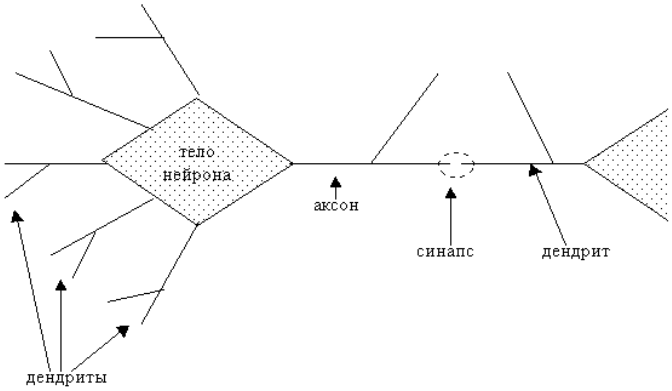



Рис. 5.1 - Структура биологического нейрона

Принятые синапсом входные сигналы подводятся к телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, другие – воспрепятствовать его возбуждению. Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону сигнал другим нейронам. У этой основной функциональной схемы много усложнений и исключений, тем не менее большинство искусственных нейронных сетей моделируют лишь эти простые свойства.

Итак, нейрон (формальный нейрон, нейроподобный элемент), представляющий собой примитивное вычислительное устройство, имеющее несколько входов и один выход является основным вычислительным элементом НС.

Однослойный персептрон является одним из самых простых вариантов НС (рис. 5.2) и содержит вего один нейрон. Являясь самостоятельной моделью НС с одной стороны, формальный нейрон (однослойный персептрон) является основным конструкционным элементов для большинства моделей НС, с другой стороны.

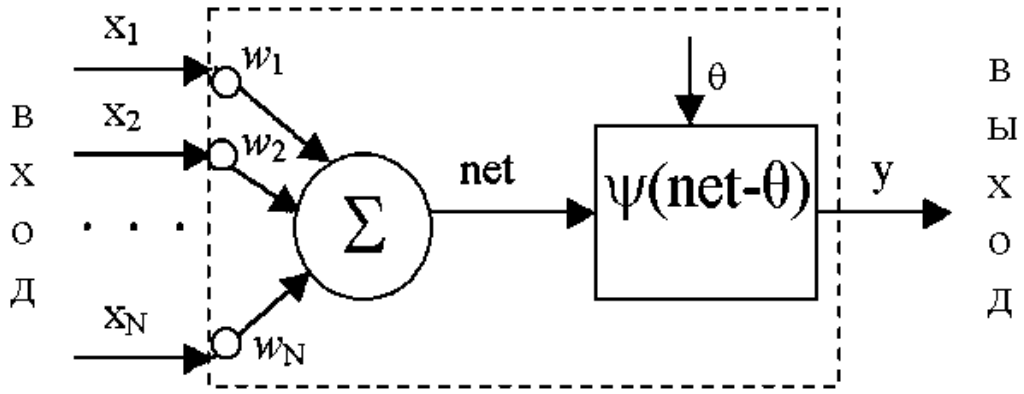



Рис. 5.2 - Однослойный персептрон (формальный нейрон)

На вход нейроподобного элемента (формального нейрона) поступает набор входных сигналов $x_1$, $x_2$, ..., $x_N$ или входной вектор x. Каждый входной сигнал умножается на соответствующий вес связи $w_1$, $w_2$, ..., $w_N$ - аналог эффективности синапса (межнейронного контакта).

Вес связи является скалярной величиной, положительной для возбуждающих и отрицательной для тормозящих связей. Взвешенные весами связей входные сигналы поступают на блок суммации, соответствующий телу клетки, где осуществляется их алгебраическая суммация и определяется уровень возбуждения нейроподобного элемента net:

$$net = \sum_{i=1}^{N} w_i x_i \quad .$$

Выходной сигнал нейрона y определяется путем пропускания уровня возбуждения net через нелинейную функцию активации ψ:

y = ψ(net-θ), где θ - некоторое постоянное смещение (аналог порога нейрона). Обычно используются простейшие нелинейные функции: бинарная (пороговая):

$$ш(x) = \begin{cases} 1, \text{при } x > 0, \\ 0, \text{при } x < 0, \end{cases}$$

или сигмоидная: $\psi(x) = 1/(1+e^{-x})$.

В зависимости от типа функции активации различают дискретные персептроны, использующие пороговую функцию активации, и вещественные – использующие вещественные функции активации, например сигмоидную функцию.

Каждый нейрон обладает небольшой памятью, реализуемой весовыми коэффициентами входных синапсов (межнейронных контактов) и порогом нейрона. Поэтому нейроны можно рассматривать как запоминающие устройства. В то же время нейроны могут рассматриваться как примитивные процессоры, осуществляющие вычисление значения функции актиивации на основе разности взвешенной суммы входных сигналов и порога.

**Алгоритм обучения однослойного дискретного персептрона** имеет вид.

Шаг 1. Весам $w_i(0)$ (i=1, ... ,N) и порогу θ(0) присваиваются случайные значения (через $w_i(t)$ обозначен весовой коэффициент i-го входа персептрона в момент времени t, через θ(t) обозначена величина смещения (порога) нейрона в момент времени t).

Шаг 2. Предъявляются очередной входной вектор $x=\{x_1,...,x_N\}^T$ из обучающего множества и желаемый выход y*(t) (y*(t) = 1, если x(t) относится к классу A, y*(t) = 0, если x(t) относится к классу B).

Шаг 3. Вычисляется реальное значение на выходе персептрона по формулам:

$$net = \sum_{i=1}^{N} w_i(t)x_i(t),$$

$$y(t) = ш(net - и(t)).$$

Шаг 4. Корректируются веса согласно равенствам:

$w_i(t+1) = w_i(t) + \eta(y^*(t) - y(t))x_i(t)$, $i = 1,2,...,N$,

$\theta(t+1) = \theta(t) + \eta(y^*(t) - y(t))$,

где $\eta$ - положительное корректирующее приращение.

Шаг 5. Если достигнута сходимость, то процедура обучения заканчивается; в противном случае - переход к шагу 2.

Согласно данному алгоритму сначала производится инициализация параметров персептрона случайными значениями. Затем поочередно предъявляются образы с известной классификацией, выбранные из обучающего множества, и корректируются веса в соответствии с формулами шагов 3 и 4. Величина коррекции определяется положительным корректирующим приращением $\eta$ конкретное значение которого выбирается достаточно большим, чтобы быстрее производилась коррекция весов, и в то же время достаточно малым, чтобы не допустить чрезмерного возрастания значений весов.

Процедура обучения продолжается до тех пор, пока не будет достигнута сходимость, то есть пока не будут получены веса, обеспечивающие правильную классификацию для всех образов из обучающего множества.

В том случае, когда обучающие выборки разделить гиперплоскостью невозможно для обучения персептрона можно использовать **алгоритм Уидроу-Хоффа**, минимизирущий среднеквадратическую ошибку между желаемыми и реальными выходами сети для обучающих данных. Этот алгоритм также можно применять для обучения однослойного вещественного персептрона. Алгоритм Уидроу-Хоффа можно записать в том же виде, что и вышеописанный алгоритм, предполагая что в узлах персептрона нелинейные элементы отсутствуют, а корректирующее приращение $\eta$ в процессе итераций постепенно уменьшается до нуля.

Если для решения задачи распознавания образов используется дискретный персептрон, решающее правило относит входной образ к классу A, если на выход персептрона равен 1, и к классу B – в противном случае.

В случае, если для решения задач распознавания образов используется вещественный персептрон, решающее правило относит входной образ к классу A, если выход сети больше 0.5, и к классу B в противном случае.

Однослойный персептрон может использоваться как **гауссовский классификатор максимального правдоподобия**. В этом случае предполагается, что функции правдоподобия входных векторов имеют гауссовскую форму и каждый класс задается своей ковариационной матрицей и вектором средних. Входной вектор в этом случае причисляется к тому классу, для которго его функция правдоподобия максимальна.

Для упрощения вычислений вместо самой функции правдоподобия используют ее логарифм (делать это позволяет тот факт, что логарифмическая функция является возрастающей). Если составляющие входного вектора не коррелируют, то эти логарифмы могут быть вычислены узлами однослойного персептрона, у которых отсутствуют нелинейные элементы. Действительно, предположив, что функции распределения входных векторов для различных классов различаются лишь средними, для логарифмов функций правдоподобия можно записать:

$$\ln p_j(x) = a - \sum_{i=1}^{N} \frac{x_i^2}{2y_i^2} + \sum_{i=1}^{N} \frac{m_{ji} x_i}{y_i^2} - \sum_{i=1}^{N} \frac{m_{ji}^2}{2y_i^2},\ j=1,...,M,$$

где $p_j(x)$ — функция правдоподобия входного вектора $x=\{x_1,x_2,....,x_N\}^T$ для j-го класса (то есть плотность вероятности вектора x при условии, что он относится к j-му классу); M – количество классов; $m_{ji}$ - математическое ожидание составляющей $x_i$ для j-го класса; $y_i^2$ - дисперсия величины $x_i$ (в нашем случае одинаковая для каждого класса); a - некоторая постоянная.

Так как первые два слагаемых в данном выражении являются постоянными для всех классов, то при классификации они могут быть отброшены, а оставшиеся два слагаемых могут быть вычислены линейными узлами персептрона, если принять веса и пороги равными

$$w_{ji} = \frac{m_{ji}^2}{y_i^2} \text{ и } и_i = \sum_{i=1}^{N} \frac{m_{ji}^2}{2y_i^2},\ i=1,2,...,N,\ j=1,2,....,M,$$

где $w_{ji}$ - весовой коэффициент i-го входа j-го узла; $и_j$ - порог j-го узла.

## 5.3 Многослойный персептрон

### 5.3.1 Модель сети

Основным вычислительным элементом многослойного персептрона или многослойной нейронной сети (МНС) прямого распространения является формальный нейрон. Он выполняет параметрическое нелинейное преобразование входного вектора x в скалярную величину y. Нейроны образуют сеть, которая характеризуется следующими параметрами и свойствами: M - число слоев сети, $N_\mu$ - число нейронов μ-го слоя, связи между нейронами в слое отсутствуют.

Выходы нейронов μ-го слоя, μ = 1,2,...,M-1 поступают на входы нейронов только следующего μ+1-го слоя. Внешний векторный сигнал x поступает на входы нейронов только первого слоя, выходы нейронов последнего M-го слоя образуют вектор выходов сети $y^{(M)}$.

Структура сети показана на рис. 5.3. Каждый i-й нейрон μ-го слоя (μi-й нейрон) преобразует входной вектор $x^{(\mu,i)}$ в выходную скалярную величину $y^{(\mu,i)}$. Это преобразование состоит из двух этапов: вначале вычисляется дискриминантная функции $net^{(\mu,i)}$, которая далее преобразуется в выходную величину $y^{(\mu,i)}$.

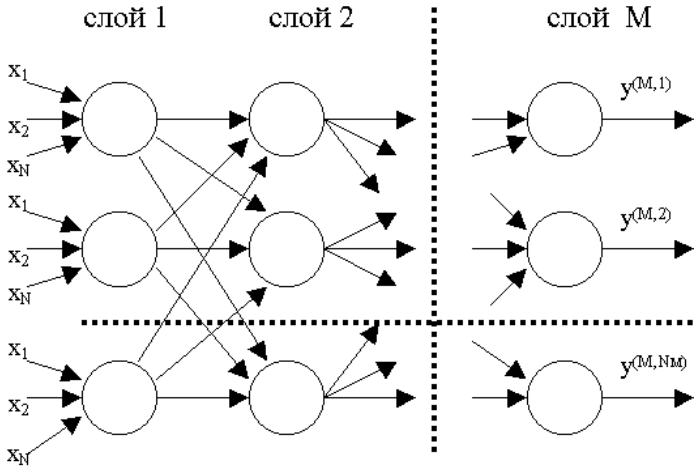



Рис. 5.3 - Структура многослойной нейронной сети

Дискриминантная функция представляет собой отрезок многомерного ряда Тейлора. Коэффициенты разложения отрезка многомерного ряда Тейлора образуют вектор весовых коэффициентов $w^{(\mu,i)}$, или память нейрона. Дискриминантная функция нейрона имеет вид:

$$net^{(м,i)} = w_0^{(м,i)} + \sum_{j=1}^{N} w_j^{(м,i)} x_j^{(м,i)},$$

где $w^{(\mu,i)}=(w_0^{(\mu,i)}, w_1^{(\mu,i)}, \ldots, w_N^{(\mu,i)})^T$ - вектор весовых коэффициентов нейрона; $x_j^{(\mu,i)}$ - j-я компонента N-мерного входного вектора $x^{(\mu,i)}$.

Нелинейное преобразование $y^{(\mu,i)} = \psi(net^{(\mu,i)})$ задается функцией активации, которая является монотонной и ограниченной. В частности, при неотрицательных выходах нейрона такой функцией может быть сигмоидная функция $\psi(x) = 1/(1+e^{-x})$.

Обозначим через $y^{(\mu)} = ( y^{(\mu,1)}, y^{(\mu,2)}, \ldots, y^{(\mu,N\mu)})^T$ вектор выхода нейронов μ-го слоя.

### 5.3.2 Обучение многослойного персептрона

Процесс обучения МНС осуществляется в результате минимизации целевой функции - некоторого критерия качества F(w), который характеризует интегральную меру близости выходов сети $y^{(M)}(k)$ и указаний учителя y*(k):

$$F(w)=\frac{1}{k}\sum_{m=1}^{k} Q(e(w,m)),$$

где k – номер текущего цикла обучения НС; m=1, 2,..., k – номера предыдущих циклов обучения НС; w - составной вектор-столбец весовых коэффициентов сети, составляющими которого являются вектор-столбцы $w^{(\mu)} =( w^{(\mu,1)T}, w^{(\mu,2)T}, \ldots, w_N^{(N\mu)T})^T$, μ=M,M-1,...,1 каждого слоя. Мгновенный критерий качества Q(ε(w,k)), входящий в интегральный критерий качества F(w), зависит от вектора ошибки сети Q(ε(w,m)): $\varepsilon(w,m) = y^{(M)}(m)-y^*(m)$.

Для каждого входного вектора x из обучающего множества должен быть определен вектор желаемых выходов сети y*. Если обучаемая МНС используется в качестве классификатора, то обычно желаемые выходы имеют низкий уровень (0 или меньше 0,1), кроме выхода узла, соответствующего классу, к которому относится x; этот выход в данном случае имеет высокий уровень (1 или больше 0,9).

Градиентные методы обучения МНС основаны на использовании градиента целевой функции F(w). Эти методы носят итеративный характер, так как компоненты градиента оказываются нелинейными функциями. Все далее рассмотренные методы основаны на итерационной процедуре, реализуемой в соответствии с формулой:

$w_{k+1} = w_k + \alpha_k s(w_k)$,

где $w_k$ , $w_{k+1}$ – текущее и новое приближения значений весов и порогов НС к оптимальному решению, соответственно, $\alpha_k$ – шаг сходимости, $s(w_k)$-направление поиска в N-мерном пространстве весов. Способ определения $s(w_k)$ и $\alpha_k$ на каждой итерации зависит от особенностей конкретного метода.

**Обобщенный градиентный алгоритм** применительно к задаче **обучения МНС** имеет следующий вид.

Шаг 1. Инициализация: Задаются параметры МНС: N-число входов, M – число слоев, начальные веса и пороги w. Задаются параметры алгоритма обучения: максимальное допустимое число циклов обучения Epochs, параметр сходимости алгоритма $\varepsilon_1$ - цель обучения (в качестве нее обычно выступает максимальная допустимая среднеквадратическая ошибка), $\varepsilon_2$ – параметр сходимости вдоль прямой (для простоты можно полагать $\varepsilon_2 = \varepsilon_1$).

Шаг 2. Положить счетчик итераций k=0.

Шаг 3. Вычислить компоненты $\frac{\partial Q(e(w_{k-1}, k))}{\partial w}$.

Шаг 4. Выполняется ли равенство $\left\| \frac{\partial Q(e(w_{k-1}, k))}{\partial w} \right\| \le e_1$?

Да: Сходимость достигнута. Перейти на шаг 13.

Нет: перейти на шаг 5.

Шаг 5. Выполняется ли неравенство k>Epochs?

Да: Достигнуто максимальное число циклов обучения, сходимость не достигнута. Перейти на шаг 13.

Нет: перейти на шаг 6.

Шаг 6. Вычислить $s(w_k)$.

Шаг 7. Выполняется ли неравенство $\frac{\partial Q(e(w_{k-1}, k))^T}{\partial w} s(w_k) < 0$?

Да: перейти на шаг 9.

Нет: положить: $s(w_k) = -\frac{\partial Q(e(w_{k-1}, k))}{\partial w}$. Перейти на шаг 9.

Шаг 8. Найти такое значение $\alpha_k$, при котором $F(w_k + \alpha_k s(w_k)) \to \min$, используя параметр $\varepsilon_2$.

Шаг 9. Положить $w_{k+1} = w_k + \alpha_k s(w_k)$.

Шаг 10. Выполняется ли неравенство $F(w_{k+1}) < F(w_k)$ ?

Да: перейти на шаг 11.

Нет: Перейти на шаг 13.

Шаг 11. Выполняется ли неравенство $\frac{\|Дw\|}{\|w_k\|} \le e_1$?

Да: Окончание поиска: нет продвижения к решению. Перейти на шаг 13.

Нет: перейти на шаг 12.

Шаг 12. Положить k=k+1. Перейти на шаг 3.

Шаг 13. Останов.

Следует заметить, что в процессе одномерного поиска следует по возможности избегать точных вычислений, так как эксперименты показывают, что на выполнение операций поиска вдоль прямой тратится весьма значительная часть общего времени вычислений.

В вышеописанном обобщенном градиентном алгоритме можно использовать различные градиентные методы путем определения соответствующих направлений поиска на шаге 6. Определяя соответствующим образом $s(w_k)$, можно трансформировать рассмотренный обобщенный градиентный алгоритм практически в любой градиентный алгоритм обучения МНС, что позволяет существенно упростить разработку нейросетевых систем без жесткой привязки к определенному градиентному алгоритму.

Рассмотрим некоторые частные случаи обобщенного градиентного алгоритма.

**Метод Коши** (метод наискорейшего спуска, в обучении нейронных сетей известный, как алгоритм обратного распространения ошибки первого порядка или Backpropagation) заключается реализации правила (шаг 6 обобщенного градиентного алгоритма):

$$s(w_k) = -\gamma \nabla_w Q(e(w_{k-1}, k)) = -\gamma \frac{\partial Q(e(w_{k-1}, k))}{\partial w}.$$

Обозначив текущий градиент $g = \frac{\partial Q}{\partial w}$, получим:

$s(w_k) = -\gamma_k g_k,$

где $\gamma_k$ –скорость обучения.

Величина $\gamma$ либо полагается постоянной, и тогда обычно последовательность $w_k$ сходится в окрестность оптимального значения w, либо она является убывающей функцией времени так, как это делается в стохастических алгоритмах оптимизации и адаптации.

Данная процедура может выполняться до тех пор, пока значения управляемых переменных не стабилизируются или пока ошибка не уменьшится до приемлемого значения. Следует отметить, что данную процедуру характеризует медленная скорость сходимости и возможность попадания в локальные минимумы функционала.

Для **метода Ньютона** или алгоритма обратного распространения ошибки второго порядка шаг 6 обобщенного градиентного алгоритма имеет вид:

$$s(w_k) = -\left( \sum_{m=1}^{k} \frac{\partial}{\partial w} \left( \frac{\partial Q(m)}{\partial w} \right)^{\Phi} \right)^{-1} g_k.$$

**Алгоритмы сопряженных градиентов** представляют собой подкласс квадратично сходящихся методов. Для алгоритмов сопряженных градиентов шаг 6 обобщенного градиентного алгоритма имеет вид:

$s(w_k) = -g_k + \beta_k s(w_{k-1}).$

В большинстве алгоритмов сопряженных градиентов размер шага корректируется при каждой итерации, в отличие от других алгоритмов, где обучающаяся скорость используется для определения размера шага. Различные версии алгоритмов сопряженных градиентов отличаются способом, по которому вычисляется константа $\beta$.

Для **алгоритма сопряженных градиентов Флетчера-Ривса** правило вычисление константы $\beta$ имеет вид:

$$в_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}},$$

где β - отношение квадрата нормы текущего градиента к квадрату нормы предыдущего градиента.

В обобщенном градиентном алгоритме отсутствует процедура возврата к начальной итерации для метода Флетчера-Ривса, но вместе с тем тесты, включенные в алгоритм, обеспечивают обнаружение любых трудностей, ассоциированных с необходимостью возврата при расчетах по методу сопряженных градиентов.

Для **алгоритма сопряженных градиентов Полака-Рибьера** правило вычисление константы β имеет вид:

$$в_k = \frac{\Delta g_{k-1}^T g_k}{g_{k-1}^T g_{k-1}},$$

где β - внутреннее произведение предыдущего изменения в градиенте и текущего градиента, деленное на квадрат нормы предыдущего градиента.

**Алгоритм Левенберга-Марквардта** требует наличия информации о значениях производных целевой функции. В алгоритме Левенберга-Марквардта используется метод обратного распространения ошибки первого порядка, чтобы вычислить якобиан J целевой функции относительно весов и порогов сети.

Для алгоритма Левенберга-Марквардта шаг 6 обобщенного градиентного алгоритма имеет вид:

$s(w_k) = -[H_k + \eta I]^{-1} g_k$,

где $H = J^T J$, J - якобиан, $g_k = J^T e$ – текущий градиент, e - вектор ошибок, η - скаляр, I – единичная матрица.

Адаптивное значение η увеличивается в $\eta^+$ раз до тех пор, пока значение целевой функции не уменьшится. После чего изменения вносятся в сеть и η уменьшается в $\eta^-$ раз.

### 5.3.3 Следящий алгоритм обучения МНС

В том случае, когда целевая функция обучения МНС является достаточно гладкой, не содержит сложных изгибов и большого количества локальных минимумов, традиционно применяемые градиентные алгоритмы обучения НС позволяют достигнуть цели обучения, обеспечивая достаточно быструю сходимость.

Когда же целевая функция является существенно нелинейной и содержит большое количество локальных минимумов, а объем обучающих данных велик, градиентные алгоритмы обеспечивают быструю сходимость только в начале обучения, после чего во многих случаях сходимость обучения существенно замедляется, что можно объяснить попаданием сети в локальные минимумы и недостаточно большим корректирующим приращением, чтобы быстро выбраться из локального минимума.

Поэтому необходимо использовать алгоритм, позволяющий ускорить сходимость градиентных алгоритмов обучения МНС в случае попадания сети в локальные минимумы.

Для ускорения работы градиентных алгоритмов при попадании НС в локальный минимум необходимо решить две задачи:

1) обнаружить, что сеть попала в локальный минимум и скорость сходимости является низкой;

2) вывести сеть из локального минимума.

Для решения первой задачи предлагается следить за процессом обучения МНС, запоминая значения целевой функции на каждом цикле обучения сети, и в случае, если в течение заданного количества циклов целевая функция уменьшилась менее чем на заданную величину, выполнять процедуру вывода сети из локального минимума.

Вывод сети из локального минимума предлагается осуществлять, изменяя значения весов и порогов сети на некоторые относительно небольшие величины таким образом, чтобы сеть сохранила весь предыдущий положительный опыт и в то

же время могла изменить свое положение в многомерном пространстве весов и порогов.

Заметим, что коррекция весов сети даже на достаточно малую величину может приводить к существенным изменениям, причем, не только положительным, но и к отрицательным, когда следящий алгоритм ухудшает сходимость, достигнутую градиентным алгоритмом. Поэтому результаты изменений весов МНС на основе следящего алгоритма должны приниматься только в том случае, когда они улучшают сходимость, в противном случае должны восстанавливаться значения весов, полученные после применения корректирующего правила градиентного алгоритма.

Обобщая вышесказанное, запишем следящий алгоритм.

Шаг 1. Инициализация параметров градиентного алгоритма обучения МНС. Инициализация параметров следящего алгоритма: шага $\alpha$, размера окна слежения $\Delta t$ (в циклах), критерия целесообразности применения алгоритма слежения на данном этапе $\xi$ и указателя ячейки окна слежения $p_t$: $p_t=1$. Резервирование памяти для окна слежения Err($\Delta t$), здесь Err($\Delta t$) – массив $\Delta t$ элементов.

Шаг 2. Установить счетчик циклов обучения epoch=0.

Шаг 3. Если epoch>Epochs, где Epochs – заданное максимальное допустимое количество циклов обучения НС, тогда перейти на шаг 11, в противном случае – перейти на шаг 4.

Шаг 4. Вычислить значение perf целевой функции обучения МНС.

Шаг 5. Если номер ячейки памяти $p_t \le \Delta t$, тогда принять $Err(p_t) = perf$, $p_t = p_t+1$; в противном случае, принять $p_t=1$, $Err(p_t) = perf$.

Шаг 6. Проверить критерий останова для градиентного алгоритма. Если градиентный алгоритм должен прекратить работу, то перейти на шаг 11, в противном случае выполнить коррекцию весов для данного цикла обучения на основе градиентного алгоритма.

Шаг 7. Если epoch> $\Delta t$ и $| perf - Err(mod(epoch, \Delta t) + 1) | / \Delta t < \xi$, где mod(a,b) – остаток от целочисленного деления a на b, тогда перейти на шаг 8, в противном случае – перейти на шаг 10.

Шаг 8. Принять: $w^*=R(w)$, где w и $w^*$ - наборы значений весов и порогов МНС, соответственно, до и после применения корректирующего правила весов НС следящего алгоритма R(w), и вычислить значение целевой функции $perf^*$ после применения правила R(w).

Шаг 9. Если $perf^*< perf$, где perf – значение целевой функции до применения корректирующего правила весов следящего алгоритма, тогда установить: w = w*.

Шаг 10. Установить: epoch=epoch+1. Перейти на шаг 3.

Шаг 11. Останов.

В качестве правила R(w) предлагается использовать следующие выражения:

$$R(w) = \alpha w, \quad (5.1)$$

$$R(w) = w + \alpha w; \quad (5.2)$$

$$R(w) = w + \alpha rand, \quad (5.3)$$

где rand – случайное число в диапазоне [0,1].

### 5.4 Радиально-базисные нейронные сети

Радиально-базисная НС (РБНС) состоит из двух слоев. Соединительные весовые векторы слоев будем обозначать $w^{(\mu,j)}$, где μ-номер слоя (μ=1,2), j – номер нейрона (узла) в слое. Базисные (или ядерные) функции в первом слое производят локализованную реакцию на входной стимул. Выходные узлы сети формируют взвешенную линейную комбинацию из базисных функций, вычисленных узлами первого слоя.

Выходные узлы соответствуют выходным классам, в то время, как узлы первого слоя представляют собой кластера (количество кластеров m задается пользователем), на которые разбивается входное пространство. Обозначим $x = (x_1, ..., x_i, ..., x_N)$ и $y = (y_1, ..., y_i, ..., y_K)$ - вход и выход сети, соответственно. Здесь N – количество признаков, а K-число классов.

Выход $u_j$ j-го узла первого слоя, используя ядерную функцию Гауссиан как базисную, определяется по формуле:

$$u_j = \exp\left[-\frac{\left(x - w^{(1,j)}\right)^T\left(x - w^{(1,j)}\right)}{2\sigma_j^2}\right],\ j=1,2,...,m,$$

где x - входной образ (экземпляр), $w^{(1,j)}$ - его входной весовой вектор (то есть центр Гауссиана для узла j) и $\sigma^2_j$ - параметр нормализации j-го узла, такой что $0 < u_j < 1$ (чем ближе вход к центру Гауссиана, тем сильнее реакция узла).

Выход $y_j$ j-го узла второго слоя определяется из выражения:

$y_j = w^{(2,j)T}u,\ j = 1, 2, ..., K;$

где $w^{(2,j)}$ – весовой вектор для j-го узла второго слоя и u - вектор выходов первого слоя.

Сеть выполняет линейную комбинацию нелинейных базисных функций. Задача обучения сети состоит в минимизации ошибки:

$$E = \frac{1}{2}\sum_{s=1}^{S}\sum_{j=1}^{K}\left(y_j^s - y_j^{s*}\right)^2,$$

где $y^{s*}_j$ и $y^s_j$ - желаемое и расчетное значения выхода j-го узла выходного слоя для s-го экземпляра, S - размер набора данных (количество экземпляров), и K - число выходных узлов (число классов). Далее для наглядности верхний индекс s опущен.

Обучение РБНС может выполняться двумя различными способами.

Первый способ заключается в том, что алгоритмом кластеризации формируется фиксированное множество центров кластеров. Затем минимизацией квадратичной ошибки, то есть минимизацией E, получают ассоциации центров кластеров с выходом.

Второй способ заключается в том, что центры кластеров могут быть также обучены наряду с весами от первого слоя до выходного слоя методом градиентного спуска. Однако, обучение центров наряду с весами может привести к попаданию сети в локальные минимумы.

Пусть фиксированное множество центров кластеров сформировано на основе первого способа, а центры кластеров будут обозначены $w^{(1,j)}$, j = 1, ..., m. Параметр нормализации $\sigma_j$ представляет меру распределения данных, ассоциируемых с каждым узлом.

Обучение в выходном слое выполняется после того, как определены параметры

базисных функций. Веса обычно обучают, используя алгоритм среднеквадратических отклонений:

$$\Delta w^{(\mu,j)} = -\eta e_j u,$$

где $e_j = y_j — y^*_j$ и $\eta$ - коэффициент скорости обучения.

## 5.5 Нейронные сети Хопфилда

### 5.5.1 Модель сети Хопфилда

НС Хопфилда (псевдоинверсная НС) задается четверкой net=(N, w, θ, x), где N — число нейронов в сети, $\theta = \{\theta_1,\theta_2,...,\theta_N\}$-вектор внешних воздействий. Нейроны связаны по принципу «все со всеми», это значит, что в сети NxN связей. Связь между i-ым и j-ым нейронами обозначается $w_{ij}$. Величина $w_{ij}$ называется весом связи и может быть нулем, положительным или отрицательным числом. Веса связей задаются матрицей $w = \{w_{ij}\}$, i, j= 1,.., N. В модели Хопфилда связи симметричные, т. е. $w_{ij} = w_{ji}$. Состояние сети определяется вектором состояний нейронов $x = \{x_1, ..., x_N\}$.

Нейрон рассматривается как двустабильный пороговый элемент (модель Мак-Каллока - Питтса). Состояние $x_i$ нейрона i может иметь два значения 0 и 1 или 1 и -1. Нейрон i имеет внешний вход θ, входы от других нейронов $x_j$ и один ветвящийся выход, равный $x_i$. Вход в нейрон i (постсинаптический потенциал) определяется суммой взвешенных состояний, связанных с ним нейронов:

$$net_i = \sum_{j=1}^{N} w_{ij} x_j + и_i \cdot$$

В зависимости от величины входа $net_i$ нейрон i изменяет свое состояние или остается в прежнем в соответствии с пороговым правилом $net_i^{k+1}=\psi(net_i^k)$, где k, k+1— номера старого и нового состояний нейрона i, а ψ(x) – функция активации нейрона:

$$ш(x) = \begin{cases} 0, x \geq 0; \\ 1, x < 0; \end{cases}$$ (пороговая) или $$ш(x) = \frac{1}{1+e^{-x}}$$ (сигмоидная).

Сеть может изменять свое состояние синхронным или асинхронным способом.

В **синхронном случае** все нейроны одновременно изменяют свои состояния. Аналитическое выражение перехода сети из состояния $x_k$ в $x_{k+1}$ записывается в матричной форме: $net_k = wx_k + \theta_k$, $x_{k+1} = \psi(net_k)$, где $x_k=\{x_1^k,x_2^k,...,x_N^k\}$, $net_k=\{net_1^k,net_2^k,...,net_N^k\}$ Функция $\psi$ применяется к вектору $net_k$ поэлементно.

В **асинхронном случае** каждый нейрон может изменять свое состояние случайно, при этом он использует информацию об обновленных состояниях других нейронов. Аналитическая запись перехода сети из состояния $x_k$ в $x_{k+1}$ в асинхронном случае, когда нейрон m изменяет свое состояние, имеет вид $net_m^k = w_m x_k + \theta_k$, $x_{k+1}=\{x_1^k,...,\psi(net_m^k),...,x_N^k\}$, где $w_m$ — строка матрицы w с номером m.

Начиная с начального состояния $x_0$ и работая синхронно или асинхронно, сеть генерирует последовательность состояний $x_0$, $x_1$, ..., $x_M$, которая в благоприятных случаях заканчивается устойчивым состоянием, в неблагоприятных случаях могут возникнуть колебания.

Основной операцией, производимой нейронной сетью, является умножение матрицы на вектор (в синхронном случае) или вектора на вектор (в асинхронном случае) с последующим вычислением нелинейной функции. Однако, благодаря массовости связей большого числа нейронов при такой достаточно простой операции сеть обладает способностью решать сложные задачи.

### 5.5.2 Обучение сети Хопфилда распознаванию образов

Одной из задач, решаемых с помощью НС, является задача распознавания образов. Сеть из N нейронов может восстанавливать образы размера N, запомненные в сети, по ключу, т. е. по неполной или неточной информации об этих образах. Это позволяет создавать на основе НС блоки ассоциативной памяти в вычислительных системах. Эталонные образы кодируются словами длины N, состоящими из 1 и -1. Переход нейрона из состояния 1 в -1 (и наоборот) будем считать ступенчатым.

Работу НС, содержащей нейроны, состояние которых определяется действием

внешних стимулов, и эффекторы, выходы которых являются реакцией на стимул, можно рассматривать как пофрагментную классификацию действующих стимулов. Множество предсинаптических потенциалов, соответствующее позитивной реакции нейрона $x_i(t) = 1$, создает компактную область вокруг вектора $w_{ij}$.

Эта область не пустая лишь при выполнении условия:

$$и_i(t) \le \left( N \sum_{j=1}^{N} w_{ij}^2 \right)^{0.5}.$$

Комбинируя значения весовых коэффициентов $w_{ij}$ и порогов $\theta_i$ можно создавать НС, способные к классификации образов без каких-либо ограничений. Нахождение значений величин $w_{ij}$ и $\theta_i(t)$, которые обеспечивают необходимые реакции на заданном множестве стимулов, составляет задачу обучения НС.

Ассоциирование (или распознавание) образа достигается сетью путем эволюции из начального состояния, соответствующему введенному образу, в конечное состояние, которым является ассоциация образа с запомненным ранее образом. Основная задача построения сети состоит в наделении ее распознающими свойствами, проявляющимися в том, что при подаче на вход сети возмущенной версии образа, сеть на выходе способна восстанавливать оригинал из шума. То есть, имея M образов-эталонов необходимо найти такую матрицу связей w, которая заставляла бы сеть проявлять распознающие свойства относительно этих эталонов. Нахождение такой матрицы составляет процесс обучения НС, а правило вычисления матрицы w является обучающим правилом. Продуктивность НС определяется количеством эталонов M, которые могут быть запомнены и распознаны сетью, состоящей из N нейронов, и тем, насколько хорошо происходит распознавание, то есть отделение эталонов от шума при данном количестве эталонов M.

Заметным шагом на пути к увеличению продуктивности псевдоинверсных НС было предложение Хопфилда рассматривать их с энергетической точки зрения. Процесс распознавания можно представить как "сдвиг" сети в минимумы некоторой энергетической функции E в пространстве состояний. Предложенное Хопфилдом определение этой функции имеет вид:

$$E(t) = - 0{,}5\ x^T(t)(wx(t)-\theta(t)),$$

где x(t) - вектор состояния системы, $^{T}$ – знак транспонирования, θ(t) - внешнее поле, определяющее порог чувствительности, w - оператор, учитывающий расстояние между состояниями системы.

**Проекционный алгоритм обучения** (**псевдоинверсный алгоритм**) НС Хопфилда заключается в том, что мы осуществляем проекцию обучающего множества на множество весов НС, что позволяет обучать сеть один раз перед использованием, и не требует выполнения итеративного процесса коррекции весов, как, например, в алгоритмах обучения персептронов.

Пусть имеется M эталонных образов $x_k$, k = 1, ..., M. Положив $\theta_i=0$, i=1,...,N, можем записать выражение для нахождения весов сети:

$$w_{ij} = \sum_{k=1}^{M} x_i^k x_j^k \cdot$$

Начиная работу в состоянии $x_0$, сеть может попасть в одно из следующих трех положений:

- прийти в устойчивое состояние, соответствующее эталонному образу, который по хэмминговому расстоянию (по числу компонентов, в которых различаются два вектора) является наиболее близким к $x_0$;

- прийти в устойчивое состояние, не соответствующее никакому эталону, т. е. к ложному образу;

- оказаться вовлеченной в колебательный процесс.

Результаты моделирования сети показывают, что сеть работает хорошо, т. е. без ошибок восстанавливает эталонные образы из случайных, если в нее записывается не более, чем 0,15N эталонных образов.

Для большинства задач возможности одноразового внесения информации в ассоциативную память оказывается недостаточно, т.к. требуется в процессе работы добавлять в память новую информацию - проекционный алгоритм (см. выше) оказывается непригодным. Для решения таких задач используют **итеративный проекционный алгоритм обучения**.

Пусть мы имеем обученную для M наборов значений НС Хопфилда, тогда для внесения в память M+1 набора значений, значения весов можно установить:

$$w_{ij}^{M+1} = w_{ij}^{M} + \frac{(x_i^M - y_i)(x_j^M - y_j)}{\sum_{j=1}^{N} x_j^{M+1}(x_j^{M+1} - y_j)}, \text{ где } y_i = \sum_{j=1}^{N} w_{ij}^{M} x_j^{M+1}.$$

Для упрощения вычислительной процедуры предложены и другие правила установки весов, среди которых особо следует выделить:

$$w_{ij}^{M+1} = w_{ij}^{M} + \frac{(x_i^{M+1} - y_i)x_j^{M+1}}{N}.$$

### 5.5.3 Эффект разнасыщения

Из-за того, что НС Хопфилда позволяли создавать только дискретные ассоциативные устройства, характеризовавшиеся небольшим объемом запоминаемой информации, им долгое время уделялось крайне мало внимания. Лишь в начале 90-х годов в Институте проблем математических машин и систем НАН Украины ими заинтересовалась группа ученых. В 1995 г. Д.О. Городничий, изучая НС с перенасыщенной памятью, обнаружил, что при уменьшении веса связей, замыкающих выход нейрона на его вход, НС способна вспоминать образы, которые казались безнадежно утерянными. Этот “**эффект разнасыщения**” был положен в основу разработки нового метода повышения объема ассоциативной памяти НС Хопфилда, что позволило почти в два раза увеличить теоретический предел для объема памяти и сделало возможным регулировать количество запоминаемых образов.

Теоретическая граница объема памяти для псевдоинверсного обучающего правила составляет 50% от количества нейронов сети, но практически она является недостижимой. При обучении на основе псевдоиверсного обучающего правила поведение сети существенно зависит от уровня обратной связи нейронов. Уменьшение обратной связи будет благоприятствовать улучшению характеристики обучающего правила НС.

Теоретический анализ этой методики и ее экспериментальная проверка путем программного моделирования, показывают, что объем памяти модифицированной псевдоинверсной НС может не только достигать теоретической границы 50% от количества нейронов, но и значительно превышать ее. Предложенную методику назвали **разнасыщением сети**, а сеть, построенную по этой методике, - **разнасыщенной псевдоинверсной** НС.

Для псевдоинверсного обучающего правила при увеличении заполнения памяти НС (M/N) диагональные элементы синаптической матрицы начинают доминировать над остальными ее элементами, а с увеличением веса диагональных элементов уменьшается вероятность попадания сети в локальные минимумы. Эти два факта позволили предложить такую модификацию псевдоинверсного обучающего правила:

После того, как значения синаптической матрицы найдены, все диагональные элементы этой матрицы необходимо частично уменьшить по правилу:

$w_{ii}^{*} = Dw_{ii}$ , $0 < D < 1$.

Такое сокращение ослабляет уровень отрицательной обратной связи нейронов, что приводит к определенной дестабилизации поведения НС. Уменьшая диагональные элементы матрицы, мы уменьшаем величину соотношения $w_{ii} / w_{ij}$, что дает эффект, похожий на сокращение количества запомненных эталонов, то есть сокращение уменьшения насыщения памяти сети. Поэтому псевдоинверсное обучающее правило с сокращенными обратными связями называют **разнасыщенным псевдоинверсным правилом**. Полную НС, построенную по этому правилу, назвали **разнасыщенной сетью**. Синаптическая матрица $w^{*}$ для разнасыщенного псевдоинверсного правила может быть определена как

$w^{*} = w - (1 - D)I = xx^{T} - (1 - D)I,$

где D – коэффициент разнасыщения, $0< D <1$, оптимальное значение коэффициента D лежит в пределах 0,1-0,2; I - единичная матрица.

При разнасыщении после обучения НС диагональные элементы синаптической матрицы умножаются на положительный коэффициент D<l. При экзамене такой сети, постсинаптический потенциал каждого нейрона получает приращение:

$d_i = (D-1)w_{ii}x_i$.

Поскольку D < 1, а $w_{ii} > 0$, приращение имеет знак, противоположный выходу нейрона, и действует как дестабилизирующий фактор. Критический объем памяти сети при деформации увеличивается, удваиваясь при D = 0.

Экспериментально подтверждена возможность значительного (в 2-3 раза) увеличения объема ассоциативной памяти НС при ослаблении диагонали в 3-5 раз (D = 0,2-0,3). Дальнейшее уменьшение веса диагональных элементов синаптической матрицы при параллельной организации нейровычислений увеличивает риск потери стабильности сети.

### 5.6 Нейронная сеть Хэмминга

Когда на выходе НС достаточно получать номер образца, ассоциативную память успешно реализует сеть Хэмминга. Данная разновидность нейросети характеризуется, что видно по ее структуре, изображенной на рис. 5.4, в сравнении с чаще используемой сетью Хопфилда, снижением затрат на память и также снижением общего объема вычислений.

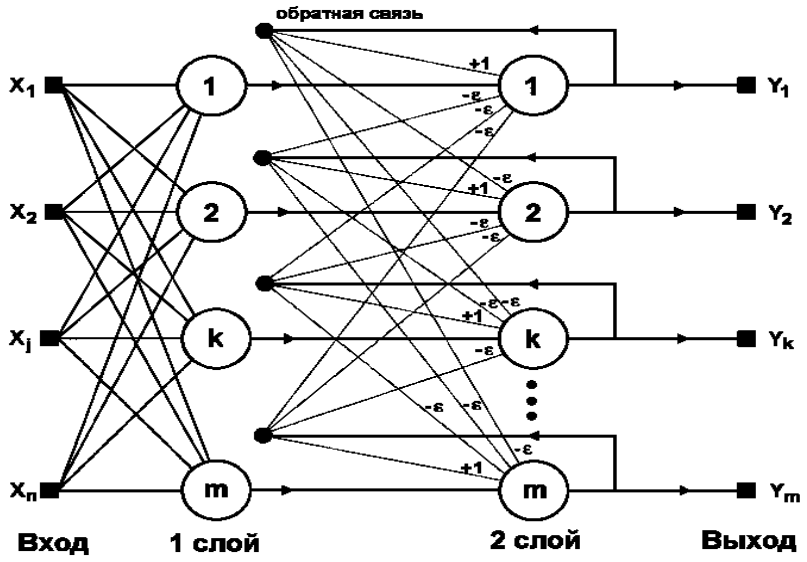



Рис. 5.4 - Структура нейронной сети Хемминга.

Идея данной конфигурации сети состоит в нахождении расстояния Хэмминга от тестируемого образца до всех образцов. При этом, расстоянием Хэмминга принято считать число отличающихся битов в двух бинарных векторах.

Сеть выбирает образец с минимальным расстоянием Хэмминга до неизвестного входного сигнала, в результате чего будет активизирован только один выход сети, соответствующий этому образцу.

На стадии инициализации весовым коэффициентам первого слоя и порогу активационной функции присваиваются следующие значения:

$w_{ik} = 0{,}5x_i^k$, i = 0,..., N-1, k = 0, ..., m-1,

$\theta_k$ = N / 2, k = 0, ..., m-1,

где $x_i^k$ – i-ый элемент k-го образца, n - количество признаков, m – количество нейронов в слое.

Весовые коэффициенты тормозящих синапсов во втором слое берут равными некоторой величине $0 < \varepsilon < 1/m$. Синапс нейрона, связанный с его же аксоном имеет вес +1.

Алгоритм функционирования сети Хэмминга следующий:

1. На входы сети подается неизвестный вектор $x = \{x_i\}$, i = 0, ..., N-1, исходя из которого рассчитываются состояния нейронов первого слоя (верхний индекс указывает номер слоя):

$$y^{(1,j)} = net^{(1,j)} = \sum_{i=0}^{N-1} w_{ij} x_j + и_j, \; j=0, ..., m-1.$$

После этого полученными значениями инициализируются значения аксонов второго слоя:

$y^{(2,j)} = y^{(1,j)}$, j = 0, ..., m-1.

2. Вычислить новые состояния нейронов второго слоя:

$$net^{(2,j)}(p+1) = y^{(j)}(p) - e\sum_{k=0}^{m-1} y^{(2,k)}(p), \; k \neq j, \; j = 0,...,m-1$$

и значения их аксонов:

$$y^{(2,j)}(p+1) = ш\left[net^{(2,j)}(p+1)\right], \; j = 0,..., m-1.$$

Активационная функция $\psi$ имеет вид порога, причем границы порога должны быть достаточно велики, чтобы любые возможные значения аргумента не приводили к насыщению.

3. Проверить, изменились ли выходы нейронов второго слоя за последнюю итерацию. Если да – перейди к шагу 2. Иначе – завершение алгоритма.

При внимательной оценке алгоритма видно, что роль первого слоя весьма условна: воспользовавшись один раз на шаге 1 значениями его весовых коэффициентов, сеть больше не обращается к нему, поэтому первый слой может быть вообще исключен из сети (заменен на матрицу весовых коэффициентов).

### 5.7 Машина Больцмана

Машина Больцмана задается четверкой {N, E, $x_0$, w}, где N - число нейронов, E = {(i, j)} - множество связей между нейронами, i, j= l,...,N при этом все автосвязи принадлежат этому множеству, то есть {(i,i)} - подмножество E. Каждый нейрон может иметь состояние 0 или 1. Состояние $x_k$ машины Больцмана определяется состояниями нейронов $x_k= (x_{k1}, ..., x_{kN})$, $x_0$ - начальное состояние.

Каждая связь (i, j) имеет вес $w_{ij}$ — вещественное число, множество весов связей обозначается w. Связь (i, j) называется активной в состоянии $x_k$, если $x_{ki}x_{kj} = 1$ Вес связи (i, j) интерпретируется как количественная мера желательности, чтобы эта связь была активной. Если $w_{ij}>>0$, то считается, что активность связи очень желательна, если $w_{ij}<<0$, то очень не желательна. Как и в модели Хопфилда связи в машине Больцмана симметричны, то есть $w_{ij} = w_{ji}$.

Для состояния $x_{ij}$ машины Больцмана вводится понятие консенсуса:

$$C_k = \sum_{(i,j)} w_{ij} x_i^k x_j^k \cdot$$

Консенсус $C_k$ интерпретируется как количественная мера желательности, чтобы все связи (i, j) в состоянии $x_k$ были активными.

Для состояния $x_k$ определяется множество соседей x(k). Соседнее состояние $x_k(i)$, принадлежащее x(k), получается из $x_k$ при изменении состояния нейрона i:

$$x_j^{k(i)} = \begin{cases} x_j^k, j \neq i, \\ 1 - x_j^k, j = i. \end{cases}$$

Разница консенсусов соседних состояний $x_k$ и $x_k(i)$ равна:

$$ДС_{kk(i)} = C_{k(i)} - C_k = \left(1 - 2x_i^k\right)\left(\sum_{(i,j)\in E_{(i)}} w_{ij}x_i^k + w_{ii}\right),$$

где E(i) - множество связей нейрона i.

Эволюция состояний от начального $x_0$ с максимизацией консенсуса приводит машину к финальному состоянию, имеющему локальный или глобальный максимум консенсуса и соответствующему решению задачи, близкому к оптимальному или оптимальному.

Переход машины Больцмана из состояния в состояние с максимизацией консенсуса происходит путем выполнения пошаговой процедуры. На каждом шаге ее выполняется испытание, состоящее из двух частей:

1) для данного состояния $x_k$ генерируется соседнее $x_k(i)$,

2) оценивается, может ли быть принято состояние $x_k(i)$, если может, то результат испытания— $x_k(i)$, иначе $x_k$.

Состояние $x_k(i)$ принимается с вероятностью:

$$A_{kk(i)}(t) = \frac{1}{1 + e^{\frac{+ДС_{kk(i)}}{t}}},$$

где t - управляющий параметр (положительное вещественное число).

Процесс максимизации консенсуса начинается с высокого значения $t_0$ параметра t и случайно выбранного начального состояния $x_0$. В течение процесса параметр t уменьшается от $t_0$ до 0. По мере того как t приближается к нулю нейроны все реже изменяют свои состояния и, наконец, машина Больцмана стабилизируется в финальном состоянии.

Асимптотически машина Больцмана способна прийти к финальному состоянию, соответствующему оптимальному решению задачи. Но практически сходимость машины к состоянию с максимальным консенсусом гарантирована быть не может, машина Больцмана стабилизируется в состоянии, соответствующем локальному максимуму консенсуса, который близок (или равен) глобальному.

Сходимостью машины Больцмана управляют следующие параметры:

1) начальное значение параметра t для каждого нейрона i:

$$t_0^{(i)} = \sum_{(i,j)\in E_{(i)}} |w_{ij}| + |w_{ii}|,$$

2) правило затухания t:

$$t_{j+1}^{(i)} = \text{б}t_j^{(i)},$$

где б - положительное число меньшее единицы, но близкое к ней;

3) число L испытаний, которые проводятся без изменения t (L - функция от N),

4) число M последовательных испытаний, не приводящих к изменению состояния машины (M - также функция от N), как критерий завершения процесса.

Описанный процесс максимизации консенсуса является последовательным. Максимизация может быть выполнена параллельно синхронным или асинхронным способом.

Для выполнения **синхронного процесса** все множество нейронов разбивается на непересекающиеся подмножества $\{W_1, ..., W_m\}$, такие, что нейроны, попавшие в одно подмножество, не связаны один с другим. Тогда на каждом такте синхронизации элементы случайно выбранного подмножества $W_i$ могут одновременно изменить свои состояния в соответствии с вероятностью.

В **асинхронном** параллельном процессе все нейроны могут изменять свои состояния только в зависимости от величины вероятности. Практически асинхронный параллелизм может быть выполнен следующим образом. Случайно выбирается подмножество W, содержащее q нейронов. Для каждого нейрона из этого подмножества устанавливается состояние в соответствии с выражением. Получившееся в результате состояние $x_i$ есть результат одного асинхронного шага. Число q выбирается равным 2/3.

### 5.8 Двунаправленная ассоциативная память

**Двунаправленная ассоциативная память** (ДАП – Bi-directional Associative Memory, **BAM, НС Коско**) относится к гетероассоциативной памяти. Входной вектор поступает на один набор нейронов, а соответствующий выходной вектор

вырабатывается на другом наборе нейронов. Входные образы ассоциируются с выходными.

Для сравнения: сеть Хопфилда является автоассоциативной. Входной образ может быть восстановлен или исправлен сетью, но не может быть ассоциирован с другим образом. В сети Хопфилда используется одноуровневая структура ассоциативной памяти, в которой выходной вектор появляется на выходе тех же нейронов, на которые поступает входной вектор.

ДАП, как и сеть Хопфилда, способна к обобщению, вырабатывая правильные выходные сигналы, несмотря на искаженные входы. Кроме того, могут быть реализованы адаптивные версии ДАП, выделяющие эталонный образ из зашумленных экземпляров. Эти возможности сильно напоминают процесс мышления человека и позволяют искусственным нейронным сетям сделать шаг в направлении моделирования мозга.

На рис. 5.5 приведена базовая конфигурация ДАП. Она выбрана таким образом, чтобы подчеркнуть сходство с сетями Хопфилда и предусмотреть увеличения количества слоев.

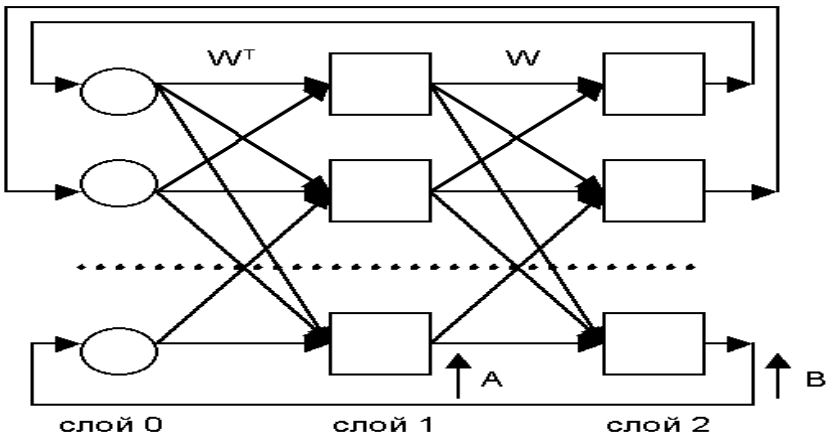



Рис. 5.5 - Структура двунаправленной ассоциативной памяти

На рис. 5.5 входной вектор A обрабатывается матрицей весов w сети, в результате чего вырабатывается вектор выходных сигналов нейронов B. Вектор B затем обрабатывается транспонированной матрицей $w^T$ весов сети, которая вырабатывает новые выходные сигналы, представляющие собой новый входной

вектор A. Этот процесс повторяется до тех пор, пока сеть не достигнет стабильного состояния, в котором ни вектор A, ни вектор B не изменяются.

Нейроны в слоях 1 и 2 функционируют, как и в других моделях НС, вычисляя сумму взвешенных входов и вычисляя по ней значение функции активации ψ. Этот процесс может быть выражен следующим образом:

$b_i = ш(\sum_j a_j w_{ij})$ или в векторной форме: B = ψ(Aw),

где B – вектор выходных сигналов нейронов слоя 2, A – вектор выходных сигналов нейронов слоя 1, w – матрица весов связей между слоями 1 и 2, ψ – функция активации.

Аналогично: $A = \psi(Bw^T)$, где $W^T$ является транспозицией матрицы W.

В качестве функции активации используется сигмоидная функция.

Слой 0 не производит вычислений и не имеет памяти. Он является только средством распределения выходных сигналов слоя 2 к элементам матрицы $w^T$.

Формула для вычисления значений синаптических весов:

$$w = \sum_i A_i^T B_i ,$$

где $A_i$ и $B_i$ - входные и выходные сигналы обучающей выборки.

Весовая матрица вычисляется как сумма произведений всех векторных пар обучающей выборки.

Системы с обратной связью имеют тенденцию к колебаниям. Они могут переходить от состояния к состоянию, никогда не достигая стабильности. Доказано, что ДАП безусловно стабильна при любых значениях весов сети.

Емкость ДАП жестко ограничена. Если n - количество нейронов в меньшем слое, то число векторов, которые могут быть запомнены в сети не превышает $L = n / 2 \log_2 n$.

ДАП обладает некоторой непредсказуемостью в процессе функционирования, возможны ложные ответы.

По сравнению с автоассоциативной памятью (например, сетью Хопфилда), двунаправленная ассоциативная память дает возможность строить ассоциации между векторами A и B, которые в общем случае имеют разные размерности. За

счет таких возможностей гетероассоциативная память имеет более широкий класс приложений, чем автоассоциативная память. Процесс формирования синаптических весов простой и быстрый. Сеть быстро сходится в процессе функционирования.

Рассмотрим модификации ДАП.

Негомогенная двунаправленная ассоциативная память, в которой пороговые значения подбираются отдельно для каждого нейрона (в исходной модели ДАП все нейроны имеют нулевые пороговые значения). Емкость негомогенной сети выше, чем исходной модели. Сигналы в сети могут быть как дискретными, так и непрерывными. Для обоих случаев доказана стабильность сети.

Адаптивная ДАП (ДАП с обучением без учителя) изменяет свои веса в процессе функционирования. Это означает, что подача на вход сети обучающего набора входных векторов заставляет ее изменять энергетическое состояние до получения резонанса. Постепенно кратковременная память превращается в долговременную память, настраивая сеть в результате ее функционирования. В процессе обучения векторы подаются на слой А, а ассоциированные векторы на слой В. Один из них или оба вектора могут быть зашумленными версиями эталона; сеть обучается исходным векторам, свободным от шума. В этом случае она извлекает сущность ассоциаций, обучаясь эталонам, хотя «видела» только зашумленные аппроксимации.

Так как доказано, что непрерывная ДАП является стабильной независимо от значения весов, ожидается, что медленное изменение ее весов не должно нарушить этой стабильности.

Простейший обучающий алгоритм использует правило Хэбба, в котором изменение веса пропорционально уровню активации его нейрона-источника и уровню активации нейрона-приемника. Символически это можно представить следующим образом:

$\Delta w_{ij} = \eta(net_i\ net_j)$,

где $\Delta w_{ij}$ – изменение веса связи нейрона i с нейроном j в матрицах w или $w^T$, $net_i$ – выход нейрона i слоя 1 или 2; $\eta$ – положительный нормирующий коэффициент обучения, меньший 1.

Введение латеральных связей внутри слоя дает возможность реализовать конкурирующую ДАП. Веса связей формируют матрицу с положительными значениями элементов главной диагонали и отрицательными значениями остальных элементов. Теорема Кохена-Гроссберга показывает, что такая сеть является стабильной.

### 5.9 Нейросетевой селектор максимума

В задачах классификации часто требуется выделить вход, имеющий максимальное значение. Для выполнения данной операции могут использоваться НС с латеральным торможением.

Алгоритм селекции максимума, выполняемый сетью, приведен в ниже.

Шаг 1. Сети предъявляются N входных величин $x_1,x_2,...,x_N$, из которых она должна выбрать максимальную, и инициализируются ее выходы $м_i(0) = x_i,\ 1 \le i \le N$, где $м_i(t)$- выход i-го узла в момент времени t.

Шаг 2. Производятся итеративные вычисления по правилу

$$м_j(t+1) = f\left( м_j(t) - e\sum_{k \ne i} з_k(t) \right), 1 \le j,\ k \le N,\ 0 < e < 1/N ,$$

где f - кусочно-линейная функция.

Итерации продолжаются до тех пор, пока не будет достигнута сходимость, посте которой положительным остается выход только одного узла сети.

После инициализации выходов сети входными значениями начинаются итерационные вычисления, которые продолжаются до тех пор, пока не будет достигнута сходимость, после чего оставшийся положительным выход будет соответствовать максимальному входному значению. Если используемая в алгоритме величина $e$ меньше 1/N, то данный алгоритм будет сходиться и положительным останется лишь один выход.

## 5.10 Карта признаков самоорганизации Кохонена

### 5.10.1 Формирование сети Кохонена

Карта признаков самоорганизации Кохонена (Kohonen Self-organizing Map – КПСК, SOM) является НС с латеральным торможением и относится к классификаторам, для обучения которых используются выборки образов с заранее не заданной классификацией.

Задачей сети является определение принадлежности входного вектора признаков s-го экземпляра выборки $x^s = \{x^s_1, x^s_2, ..., x^s_N\}^T$ к одному из L возможных кластеров, представленных векторными центрами $w_j = \{w_{j1}, w_{j2}, ..., w_{jN}\}^T$, j=1,2, ..., L , где $^T$ –символ транспонирования.

Обозначим i-ю компоненту входного вектора $x^s$ в момент времени t как $x^s_i(t)$, а вес i-го входа j-го узла в момент времени t как $w_{ij}(t)$.

Если узлы КПСК являются линейными, а вес i-го входа j-го узла равен $w_{ij}$, i=1,.2,...,N, j=1,2,...,L, то, очевидно, что при соответствующих значениях порогов каждый i-й выход сети с точностью до несущественных постоянных будет равен евклидовому расстоянию $d_j$ между предъявленным входным вектором $x^s_i$ и j-м центром кластера.

Считается, что вектор $x^s$ принадлежит к j-му кластеру, если расстояние $d_j$ для j-го центра кластера $w_j$ минимально, то есть если $d_j \le d_k$ для каждого $k \neq j$.

При обучении НС предъявляются входные векторы без указания желаемых выходов и корректируются веса согласно алгоритму, который предложил Теуво Кохонен. **Алгоритм Кохонена**, формирующий карты признаков, требует, чтобы возле каждого узла было определено поле NE, размер которого с течением времени постоянно уменьшается.

Шаг 1. Инициализируются веса входов узлов малыми случайными значениями. Устанавливается начальный размер поля NE.

Шаг 2. Предъявляется новый входной вектор $x^s$.

Шаг 3. Вычисляется расстояние (метрика) $d_j$ между входным вектором и

каждым выходным узлом j:

$$d_j = \sum_{i=1}^{N} (x_i^s(t) - w_{ji}(t))^2 \quad .$$

Шаг 4. Определяется узел j* с минимальным расстоянием $d_j$.

Шаг 5. Корректируются веса входов узлов, находящихся в поле $NE_j(t)$ узла j*, таким образом, чтобы новые значения весов были равны

$$w_{ji}(t+1) = w_{ji}(t) + з(t)(x_i^s - w_{ji}(t)) \ , \ j \in NE_j(t), \ i = 1,2,...,N.$$

При этом корректирующее приращение $\eta(t)$ ($0< \eta(t)<1$) должно убывать с ростом t.

Шаг 6. Если сходимость не достигнута, то перейти к шагу 2.

Сходимость считается достигнутой, если веса стабилизировались и корректирующее приращение $\eta$ в шаге 5 снизилось до нуля.

Если число входных векторов в обучающем множестве велико по отношению к выбранному числу кластеров, то после обучения веса сети будут определять центры кластеров, распределенные в пространстве входов таким образом, что функция плотности этих центров будет аппроксимировать функцию плотности вероятности входных векторов. Кроме того, веса будут организованы таким образом, что топологически близкие узлы будут соответствовать физически близким (в смысле евклидова расстояния) входным векторам.

Из выше изложенного следует, что КПСК способны разделять экземпляры по степени близости их признаков. Это позволяет применять КПСК для выделения центров сосредоточения экземпляров, что может быть использовано при планировании обучающего эксперимента, в случае, когда большое количество опытов ставить затруднительно, например, по причине дороговизны или уникальности изделий. Планирование обучающего эксперимента в этом случае может быть проведено следующим образом: на основании значений признаков всех экземпляров обучающей выборки производится формирование КПСК, а затем для экземпляров, которые наиболее близки к сформированным векторным центрам КПСК, проводятся эксперименты по определению фактических классов.

### 5.10.2 Интерпретация результатов классификации НС Кохонена

Важно отметить, что при классификации с помощью КПСК, номер узла, к которому отнесен экземпляр, и фактический номер его класса в общем случае не совпадают - разделяя экземпляры, КПСК производит субъективную классификацию, не имеющую того реального физического смысла, которым мы наделяем классы.

Результаты классификации КПСК могут быть наделены фактическим смыслом путем постановки в соответствие номеру каждого узла КПСК номера того фактического класса, к которому относится большая часть экземпляров обучающей выборки, отнесенных КПСК к данному узлу. Для осуществления такой постановки можно предложить использовать простой способ, основанный на использовании ассоциативного запоминающего устройства (АЗУ).

**Алгоритм обучения системы КПСК-АЗУ** имеет вид:

Шаг 1. Реализуется обучающий эксперимент и определяются фактические классы экземпляров. Производится обучение КПСК для всех экземпляров обучающей выборки

Шаг 2. Для каждого узла КПСК подсчитывается число экземпляров, относящихся к каждому из фактических классов.

Шаг 3. Каждому узлу КПСК ставится в соответствие тот фактический класс, к которому относится большая часть экземпляров, отнесенных КПСК к данному узлу. Постановка соответствия производится путем записи пары (кортежа) <номер узла КПСК, номер класса> в АЗУ. В качестве АЗУ может быть использован как блок линейной или динамической памяти, обслуживаемый соответствующей процедурой, так и нейросетевая ассоциативная память:

а) для системы с двумя классами - однослойный дискретный персептрон;

б) для системы с большим числом классов – многослойная нейронная сеть или комбинация ассоциативной памяти на основе НС Хопфилда с нейросетевым селектором максимума. При этом на соответствующие входы НС Хопфилда подаются сигналы от каждого из узлов КПСК, а на выходе получают 0, если номер узла КПСК не сопоставлен данному классу и 1 – если сопоставлен. Нейросетевой

селектор максимума определяет номер узла НС Хопфилда (т.е. номер фактического класса), для которого выход равен 1, для всех остальных узлов КПСК выход НС Хопфилда будет равен 0.

Блок КПСК-АЗУ может быть рекомендован для использования в системах классификации в случае, когда:

а) реальный принцип деления экземпляров на классы совпадает или близок к методу классификации КПСК.

б) размер обучающей выборки, то есть совокупности значений признаков экземпляров и сопоставленных им номеров классов, недостаточен для классификации другими методами (статистическими, нейросетевыми), а экземпляры одного и того же класса имеют близкие значения признаков, то есть классы хорошо разделяются и имеют центры, вокруг которых достаточно плотно сосредоточены экземпляры, относящиеся к данному классу.

### 5.10.3 Выбор метрики и учет информативности признаков

Евклидово расстояние, является частным случаем метрики Минковского:

$$d_j = \left( \sum_{i=1}^{N} | x_i^s(t) - w_{ji}(t) |^{л} \right)^{1/л}, \ \lambda \in R.$$

Так, как метрика $d_j$ вычисляется для всех узлов по одной и той же формуле, то вычисление корня λ-й степени можно опускать. Изменяя λ, можно получить неограниченное число других метрик. Однако на практике следует ограничиваться легко вычислимыми метриками, топологически близкими к фактическому разделению классов. Очевидно, чем ближе выбранная метрика к реальной закономерности разделения на классы, тем точнее и с меньшими потерями будет производиться классификация. При построении систем диагностики возникает задача автоматического выбора наилучшей метрики из множества заданных для каждого конкретного набора классов изделий.

Пусть D = $\{d^p\}$ - множество заданных метрик $d^p$, p=1,2,..$N_p$, где $N_p$-число заданных метрик. Тогда, очевидно, метрика $d^q$ для данного класса изделий является наилучшей на этом множестве в смысле точности классификации, если число принимаемых ошибочных решений при этой метрике минимально.

**Алгоритм итеративного подбора метрики** имеет вид:

Шаг 1. Установить счетчик p=1.

Шаг 2. Принять в качестве текущей метрики d метрику $d^p$: DEF FN $d=d^p$.

Здесь DEF FN – означает определение/переопределение функции.

Шаг 3. Произвести обучение блока КПСК-АЗУ на всей выборке X

Шаг 4. Определить число ошибочных решений $N_{ош}[p]$ для p-й метрики в отношении экземпляров, о которых известен фактический номер класса.

Шаг 5. Если $p \geq N_p$ – перейти на шаг 6, иначе увеличить счетчик p : p=p+1 и перейти на шаг 2.

Шаг 6. В качестве лучшей принимается та метрика $d^p$, для которой $N_{ош}[p]$ – минимально.

Выше рассмотренные метрики, как правило, предполагают, что все признаки, по которым производится классификация, являются одинаково значимыми. Однако на практике признаки разделяются на значимые и незначимые. В отношении некоторых признаков, может быть заранее известно или предполагаться, что они являются наиболее или наименее значимыми, но степень такой значимости точно неизвестна и не позволяет производить классификацию только на основе данного признака или исключить данный признак в случае его малозначимости.

Если использовать такую априорную информацию при обучении НС, то, очевидно, классификация изделий будет производиться с меньшим числом ошибок.

Для КПСК задача состоит в том, чтобы использовать априорную информацию о значимости признаков при вычислении метрики. Для примера, далее будем рассматривать в качестве метрики евклидово расстояние (1). Априорная информация о значимости признаков в КПСК может быть использована путем введения в формулу метрики положительно определенной коэффициентной функции a(i). В этом случае метрика может быть задана в одной из следующих форм:

$$d_j = \sum_{i=1}^{N} (a(i) x_i^s(t) - w_{ji}(t))^2 \quad \text{или} \quad d_j = \sum_{i=1}^{N} a(i)(x_i^s(t) - w_{ji}(t))^2 .$$

Чем информативнее i-й признак, тем меньшее значение должна принимать функция a(i) и, наоборот, чем менее значимый признак – тем большее. Вариантов задания такой функции может быть достаточно много, рассмотрим те из них, что имеют наибольшее практическое значение.

**Вариант 1.** Если априорно известно о значимости только некоторых признаков, то для тех признаков, которые предположительно значимее других, a(i) полагают равной значению a, $0 \le a < 1$. Если о i-ом признаке известно или предполагается, что он – незначимый или малозначимый, то такой признак либо вообще исключается, либо функцию a(i) полагают равной значению b, $b > 1$. Для признаков, о которых неизвестна априорная информация о значимости функцию a(i) полагают равной 1.

**Вариант 2.** Значимость признаков определяется степенью их влияния на фактический номер класса, к которому относится экземпляр. В этом случае,

функцию a(i) можно положить равной дополнению до единицы модуля коэффициента корреляции i-го признака и номера фактического класса y*:

$\forall\ x^s$: $a(i) = 1 - |r_{x_i, y^*}|$ или использовать алгоритмы определения информативности признаков.

### 5.11 Квантование обучающих векторов

Одной из моделей НС, перспективных для диагностики и прогнозирования состояния технических процессов и объектов, является квантование обучающих векторов (Learning Vector Quantization -LVQ).

В основе алгоритмов LVQ лежит механизм обучения слоя конкурирующих нейронов (конкурирующего слоя), контролируемый учителем. Конкурирующий слой может автоматически обучаться классифицировать входные векторы. По-сути, он представляет собой карту признаков самоорганизации Кохонена. Разделение классов, которые определяет карта признаков самоорганизации Кохонена, основано только на расстоянии между входными векторами. Если два входных вектора очень близки, то конкурирующий слой, весьма вероятно, отнесет их к одному классу. Однако, разделение на классы, производимое конкурирующим слоем, как правило, не совпадает с тем, что определяет учитель. Возникает задача разработки механизма, который бы позволял карте признаков самоорганизации Кохонена осуществлять классификацию, близкую к заданной учителем.

Для решения этой задачи могут быть использованы различные методы. В разделе 5.10 рассмотрен метод конструирования и обучения НС, состоящей из карты Кохонена и нейросетевого ассоциативного запоминающего устройства (АЗУ). Достоинствами данного метода являются высокая степень самоорганизации и простота практической реализации. Недостатком является то, что при формировании карты Кохонена не учитываются указания учителя, то есть информация о фактическом разделении классов, что часто приводит к формированию карты Кохонена неоптимальной структуры и иногда не позволяет обучить НС достоверной классификации.

Альтернативным методом, лишенным недостатков этого метода, позволяющим решать поставленную задачу, является LVQ.

### 5.11.1 Модель сети

Нейронная сеть LVQ (рис. 5.6) состоит из двух последовательно соединенных слоев нейронов: конкурирующего слоя и линейного слоя. Оба слоя НС LVQ содержат по одному конкурирующему и одному линейному нейрону на каждый подкласс / целевой класс. Обозначим $S^1$ – количество подклассов, $S^2$– количество целевых классов ($S^1$ всегда будет больше, чем $S^2$).

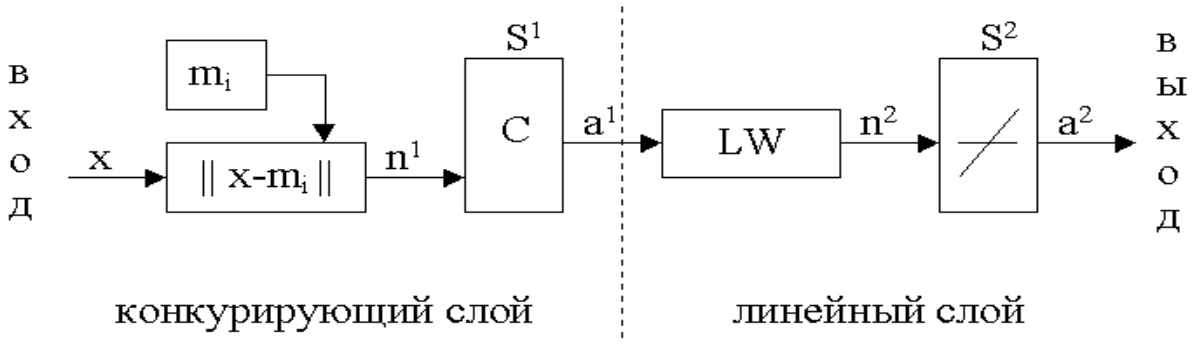



Рис. 5.6 - Схема LVQ - нейронной сети

Конкурирующий слой производит разделение входных векторов x на классы, выделяя центры сосредоточения входных векторов $m_i$. Для этого определяются расстояния $n^1=||x-m_i||$ между входными векторами x и начальными значениями центров сосредоточения векторов $m_i$, i=1,2,..,q, где q-количество входных векторов.

Линейный слой преобразует класс входного вектора, определенный конкурирующим слоем -подкласс $a^1$, в класс, определенный пользователем - целевой класс $a^2$, путем умножения $a^1$ на значения весов LW линейных нейронов, которые устанавливается равными 1, если целевой класс и подкласс совпадают и 0 – в противном случае. Соответствующие произведения $n^2=a^1LW$ подаются на выходы всех линейных нейронов, образуя двоичный вектор $a^2$, все элементы которого равны 0, за исключением элемента, который соответствует целевому классу (этот элемент равен 1).

### 5.11.2 Алгоритм обучения LVQ1

Пусть некоторое количество векторов со свободными параметрами $m_i$ помещено во входное пространство для аппроксимации различных областей входного вектора x их квантованными значениями. Каждому классу значений x назначается несколько векторов со свободными параметрами, и затем принимается решение об отнесении x к тому классу, к которому принадлежит самый близкий вектор $m_i$. Пусть индекс c определяет самый близкий к x вектор $m_i$, обозначенный далее как $m_c$:

$$c = \arg\min_i \{\|x - m_i\|\} .$$

Значения для $m_i$, минимизирующие ошибку классификации, могут быть найдены как асимптотические значения в следующем процессе обучения. Пусть x(t) – входная выборка, $m_i(t)$ -представление последовательности $m_i$, дискретизированной по времени. Начиная с правильно определенных начальных значений, основной процесс алгоритма LVQ1 определяют следующие выражения:

$m_c(t + 1) = m_c(t) + \alpha(t)[x(t) - m_c(t)]$,

если x и $m_c$ принадлежат одному и тому же классу;

$m_c(t + 1) = m_c(t) - \alpha(t)[x(t) - m_c(t)]$,

если x и $m_c$ принадлежат разным классам;

$m_i(t + 1) = m_i(t)$ , $\forall i \neq c$.

Здесь $0 < \alpha(t) < 1$, $\alpha(t)$ может быть константой или монотонно уменьшаться со временем. Для алгоритма LVQ1 рекомендуется, чтобы первоначальное значение $\alpha$ было меньше 0.1.

### 5.11.3 Алгоритм обучения LVQ2

Решение задачи классификации в алгоритме LVQ2 идентично алгоритму LVQ1. Однако в процессе обучения LVQ2 два вектора со свободными параметрами $m_i$ и $m_j$, являющиеся самыми близкими соседями x, модифицируются одновременно. Один из них должен принадлежать к классу 1, а другой - к классу 2. Кроме того, x должен находиться в зоне значений, называемой “окном”, которое определено вокруг середины плоскости, образуемой векторами $m_i$ и $m_j$. Пусть $d_i$ и $d_j$ – эвклидовы расстояния x от $m_i$ и $m_j$, тогда x определенно попадет в “окно” относительной ширины w, если

$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s$, где $s = \frac{1-w}{1+w}$.

Рекомендуется, чтобы значения относительной ширины “окна” w находились в пределах от 0.2 до 0.3.

Алгоритм обучения LVQ2 имеет вид:

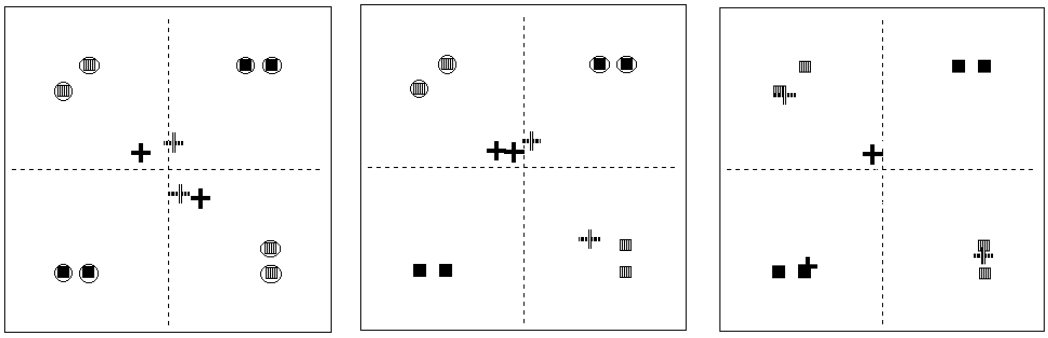
$m_i(t + 1) = m_i(t) - \alpha(t)[x(t) - m_i(t)]$,

$m_j(t + 1) = m_j(t) + \alpha(t)[x(t) - m_j(t)]$,

где $m_i$ и $m_j$ - два самых близких к x вектора со свободными параметрами, причем x и $m_j$ принадлежат к одному и тому же классу, в то время как x и $m_i$ принадлежат различным классам, кроме того, x должен попадать в “окно”.

На рис. 5.7 квадратами обозначены значения признаков экземпляров, а крестами – значения весов НС LVQ. Штриховка фигур обозначает их принадлежность к классу (сплошная – класс 1,

пунктирная – класс 2). Ошибочно классифицированные экземпляры выделены окружностями.



а) б) в)

Рис. 5.7 - Процесс квантования обучающих векторов

На рис 5.7 а) показано начальное расположение весов НС LVQ - все экземпляры классифицированы неверно. На рис 5.7 б) для тех же экземпляров показано положение весов НС LVQ после одной итерации обучения с помощью алгоритма LVQ 2 – одновременно модифицированы значения двух векторов, количество неправильно классифицированных экземпляров уменьшилось. На рис 5.7 в) показано положение весов НС LVQ после 10 итераций обучения – все экземпляры классифицированы верно.

### 5.11.4 Алгоритм обучения LVQ3

Алгоритм обучения LVQ2 основан на идее дифференциального смещения границ решения относительно Байесовских пределов, при этом не учитывается то, что может произойти с положением $m_i$ в случае достаточно продолжительной работы алгоритма. Следовательно, необходимо внести изменения, которые гарантировали бы, что $m_i$ хотя бы грубо продолжит аппроксимацию распределения классов. При объединении этих идей, мы получаем улучшенный алгоритм - LVQ3:

$m_i(t + 1) = m_i(t) - \alpha(t)[x(t) - m_i(t)]$,

$m_j(t + 1) = m_j(t) + \alpha(t)[x(t) - m_j(t)]$,

Для $k \in \{i,j\}$, если x, $m_i$ и $m_j$ принадлежат одному и тому же классу:

$m_k(t + 1) = m_k(t) + \varepsilon\alpha(t)[x(t) - m_k(t)]$,

где $m_i$ и $m_j$ - два самых близких к x вектора со свободными параметрами, причем x и $m_j$ принадлежат к одному и тому же классу, в то время как x и $m_i$ принадлежат различным классам, кроме того, x должен попадать в “окно”.

В результате ряда экспериментов было установлено, что значения ε должны находиться между 0.1 и 0.5. Оптимальное значение ε, возможно, зависит от размера наименьшего окна.

Этот алгоритм является самостабилизирующимся, то есть оптимальное размещение $m_i$ не изменяется при продолжительном обучении.

### 5.11.5 Алгоритм обучения OLVQ1

Алгоритм обучения OLVQ1 (Optimized-learning-rate LVQ1) представляет собой алгоритм LVQ1 модифицированный таким образом, чтобы каждому $m_i$ была назначена индивидуальная скорость обучения $\alpha_i(t)$. Таким образом, мы получаем следующий дискретизированный по времени процесс обучения.

Пусть c определяется уравнением:

$$c = \arg\min_i \{\|x - m_i\|\} \quad .$$

Тогда:

$m_c(t + 1) = m_c(t) + \alpha_c(t)[x(t) - m_c(t)]$,

если x классифицирован правильно;

$m_c(t + 1) = m_c(t) - \alpha_c(t)[x(t) - m_c(t)]$,

если x классифицирован неправильно;

$m_c(t + 1) = m_i(t)$ , $\forall i \neq c$.

Рассмотрим способ определения оптимального $\alpha_i(t)$ для наиболее быстрой сходимости OLVQ1. Выразим рассмотренные уравнение в форме

$m_c(t + 1) = [1 - s(t)\alpha_c(t)]m_c(t) + s(t)\alpha_c(t)x(t)$,

где $s(t) = +1$, если классификация правильная и $s(t) = -1$, если классификация неправильная.

Отмечено, что $m_c(t)$ статистически не зависит от x(t), и статистическая точность полученных значений векторов со свободными параметрами оптимальна, если результаты исправлений сделаны в разное время.

Заметим, что $m_c(t + 1)$ содержит след x(t) через последний член в последнем уравнении и прослеживает предыдущие x(t'), t' = 1, 2, ..., t-1 через $m_c(t)$.

Абсолютная величина последнего следа x(t) масштабируется коэффициентом $\alpha_c(t)$ и, в свою очередь, след x(t - 1) масштабируется коэффициентом [1 - s(t) $\alpha_c(t)$] $\alpha_c(t - 1)$.

Теперь предусмотрим, чтобы оба эти масштабирования были идентичны и применим данное условие для всех t. Тогда “следы” всех предыдущих x, собранные до времени t, будут в конце масштабироваться одним числом, и, следовательно, “оптимальные” значения $\alpha_i(t)$ определяются рекурсивно:

$$б_c(t) = \frac{б_c(t-1)}{1 + s(t) б_c(t)}.$$

На практике можно убедиться, что это правило обеспечивает быструю сходимость.

Однако, заметим, что $\alpha_c(t)$ может также увеличиваться, и важно, чтобы значение $\alpha_c(t)$ не превышало 1. Начальные значения $\alpha_i(t)$ могут быть выбраны довольно высокими (например, 0.3), благодаря чему обучение значительно ускоряется (особенно в начале) и приближенные асимптотические значения $m_i$ находятся довольно быстро.

Следует обратить внимание на то, что рассмотренное выражение не применимо для LVQ2, так как $\alpha_i$, в среднем, не будет уменьшаться и процесс не будет сходиться.

## 5.12 Контрастирование нейронных сетей

Известно, что НС с минимальным количеством нейронов должна более гладко аппроксимировать функцию, но выяснение этого минимального количества требует больших затрат времени и ресурсов ЭВМ. Если число нейронов избыточно, то можно получить результат с первой попытки, но существует риск построить “плохую” аппроксимацию. Поэтому нужно выбирать число нейронов большим, чем необходимо, но не намного. Наиболее надежным способом оценки минимального числа нейронов является использование процедуры контрастирования, предполагающей выявление и удаление малоинформативных избыточных связей.

При постановке задачи для НС не всегда удается точно определить сколько и каких входных данных нужно подавать на вход. Поэтому применение методов, позволяющих осуществлять отбор информативных признаков представляется достаточно важным. Одним из таких методов является контрастирование.

После осуществления контрастирования НС ее стуктура будет более простой и менее избыточной, что на практике позволяет ускорить работу НС, а также упростить извлечение знаний из сети в удобной для человека форме.

### 5.12.1 Контрастирование на основе показателей значимости

С помощью этой процедуры можно контрастировать, как входные сигналы, так и параметры сети. Далее в данном разделе будем предполагать, что контрастируются параметры сети. При контрастировании входных сигналов процедура остается той же, но вместо показателей значимости параметров сети используются показатели значимости входных сигналов. Обозначим через $\chi_p$ – показатель значимости p-го параметра; через $w_p^0$ – текущее значение p-го параметра; через $w_p^*$ – ближайшее выделенное значение для p-го параметра.

Используя введенные обозначения процедуру контрастирования можно записать следующим образом:

1. Вычислить показатели значимости.

2. Найти минимальный среди показателей значимости – $\chi_{p^*}$.

3. Заменить соответствующий этому показателю значимости параметр $w_{p^*}^0$ на $w_{p^*}^*$, и исключаем его из процедуры обучения.

4. Предъявить сети все примеры обучающего множества. Если сеть не допустила ни одной ошибки, то перейти ко второму шагу процедуры.

5. Обучить полученную сеть. Если сеть обучилась безошибочному решению задачи, то перейти к первому шагу процедуры, в противном случае перейти к шестому шагу.

Восстановить сеть в состояние до последнего выполнения третьего шага. Если в ходе выполнения шагов со второго по пятый был отконтрастирован хотя бы один параметр, (число обучаемых параметров изменилось), то перейти к первому шагу. Если ни один параметр не был отконтрастирован, то получена минимальная сеть.

### 5.12.2 Контрастирование без ухудшения

Пусть нам дана только обученная нейронная сеть и обучающее множество. Допустим, что вид функции оценки и процедура обучения нейронной сети неизвестны. В этом случае так же возможно контрастирование сети. Предположим, что данная сеть идеально решает задачу. В этом случае возможно контрастирование сети даже при отсутствии обучающей выборки, поскольку ее можно сгенерировать используя сеть для получения ответов. Задача не ухудшающего контрастирования ставится следующим образом: необходимо так провести контрастирование параметров, чтобы выходные сигналы сети при решении всех примеров изменились не более чем на заданную величину. Для решения задача редуцируется на отдельный адаптивный сумматор: необходимо так изменить параметры, чтобы выходной сигнал адаптивного сумматора при решении каждого примера изменился не более чем на заданную величину.

Обозначим через $x_p^q$ - p-ый входной сигнал сумматора при решении q-го примера; через $f^q$ – выходной сигнал сумматора при решении q-го примера; через

$w_p$ – вес p-го входного сигнала сумматора; через $\varepsilon$ – требуемую точность; через n – число входных сигналов сумматора; через m – число примеров. Очевидно, что при решении примера выполняется равенство $f^q = \sum_{p=1}^{n} w_p x_p^q$.

Требуется найти такой набор индексов $I = \{i_1, \ldots, i_k\}$, что $\left\| f - \sum_{p \in I} \alpha_p x_p \right\| < \varepsilon$, где $\alpha_p$ – новый вес p-го входного сигнала сумматора. Набор индексов будем строить по следующему алгоритму.

Положим $f^{(0)} = f$, $x_p^* = x_p$, $I^{(0)} = \varnothing$, $J^{(0)} = \{1, \ldots, n\}$, k=0.

Для всех векторов $x_p^*$ таких, что $p \in J^{(k)}$, проделаем следующее преобразование: если $\left\| x_p^* \right\| << \varepsilon$, то исключаем p из множества обрабатываемых векторов – $J^{(k)} = J^{(k)} / \{p\}$, в противном случае нормируем вектор $x_p^*$ на единичную длину – $x_p^{(k)} = x_p^* \Big/ \left\| x_p^* \right\|$.

Если $\left\| f^{(k)} \right\| < \varepsilon$ или $J^{(0)} = \varnothing$, то переходим к шагу 10.

Находим $i_{k+1}$ – номер вектора, наиболее близкого к $f^{(k)}$ из условия $\left| \left( f^{(k)}, x_{i_{k+1}}^{(k)} \right) \right| = \min_{p \in J^{(k)}} \left| \left( f^{(k)}, x_p^{(k)} \right) \right|$.

Исключаем $i_{k+1}$ из множества индексов обрабатываемых векторов: $J^{(k+1)} = J^{(k)} / \{i_{k+1}\}$.

Добавляем $i_{k+1}$ в множество индексов найденных векторов: $I^{(k+1)} = I^{(k)} \cup \{i_{k+1}\}$

Вычисляем не аппроксимированную часть (ошибку аппроксимации) вектора выходных сигналов: $f^{(k+1)} = f^{(k)} - \left( f^{(k)}, x_{i_{k+1}}^{(k)} \right) x_{i_{k+1}}^{(k)}$

Преобразуем обрабатываемые вектора к промежуточному представлению – ортогонализуем их к вектору $x_{i_{k+1}}^{(k)}$, для чего каждый вектор $x_p^{(k)}$, у которого $p \in J^{(k)}$ преобразуем по следующей формуле: $x_p^* = x_p^{(k)} - \left( x_p^{(k)}, x_{i_{k+1}}^{(k)} \right) x_{i_{k+1}}^{(k)}$.

Увеличиваем k на единицу и переходим к шагу 2.

Если $k = 0$, то весь сумматор удаляется из сети и работа алгоритма завершается.

Если $k = n+1$, то контрастирование невозможно и сумматор остается неизменным.

В противном случае полагаем $I = I^{(k)}$ и вычисляем новые веса связей $\alpha_p \left( p \in I \right)$ решая систему уравнений $f - f^{(k)} = \sum_{p \in I} \alpha_p x_p$.

Удаляем из сети связи с номерами $p \in J$, веса оставшихся связей полагаем равными $\alpha_p (p \in I)$.

### 5.12.3 Гибридная процедура контрастирования

Можно упростить процедуру контрастирования без ухудшения. Предлагаемая процедура годится только для контрастирования весов связей нейронов по отдельности.

Для работы нейрона наименее значимым будем считать тот вес, который при решении примера даст наименьший вклад в сумму. Обозначим через $x_p^q$ входные сигналы рассматриваемого адаптивного сумматора при решении q-го примера. Показателем значимости веса назовем следующую величину: $\chi_p^q = \left| \left( w_p - w_p^* \right) \cdot x_p^q \right|$. Усредненный по всем примерам обучающего множества показатель значимости имеет вид $\chi_p = \left| \left( w_p - w_p^* \right) \right| \cdot \max_q \left| x_p^q \right|$. Производим контрастирование на основе показателей значимости. В самой процедуре контрастирования есть только одно отличие – вместо проверки на наличие ошибок при предъявлении всех примеров проверяется, что новые выходные сигналы сети отличаются от первоначальных не более чем на заданную величину.

### 5.12. 4 Определение показателей значимости

Нейронная сеть двойственного функционирования может вычислять градиент функции оценки по входным сигналам и обучаемым параметрам сети

Показателем значимости параметра при решении q-го примера будем называть величину, которая показывает насколько изменится значение функции оценки решения сетью q-го примера если текущее значение параметра $w_p$ заменить на выделенное значение $w_p^*$. Точно эту величину можно определить произведя замену и вычислив оценку сети. Однако учитывая большое число параметров сети

вычисление показателей значимости для всех параметров будет занимать много времени. Для ускорения процедуры оценки параметров значимости вместо точных значений используют различные оценки. Рассмотрим простейшую и наиболее используемую линейную оценку показателей значимости. Разложим функцию оценки в ряд Тейлора с точностью до членов первого порядка:

$H_q(w^*) = H_q^0 + \sum_p \frac{\partial H_q}{\partial w_p}(w_p - w_p^*)$, где $H_q^0$ – значение функции оценки решения q-го примера при $w^* = w$. Таким образом показатель значимости p-го параметра при решении q-го примера определяется по следующей формуле:

$$\chi_p^q = \left| \frac{\partial H_q}{\partial w_p}\left(w_p - w_p^*\right) \right|.$$

Этот показатель значимости может вычисляться для различных объектов.

Показатель значимости параметра $\chi_p^q$ зависит от точки в пространстве параметров, в которой он вычислен и от примера из обучающего множества. Существует два принципиально разных подхода для получения показателя значимости параметра, не зависящего от примера. При первом подходе считается, что в обучающей выборке заключена полная информация о всех возможных примерах. В этом случае, под показателем значимости понимают величину, которая показывает насколько изменится значение функции оценки по обучающему множеству, если текущее значение параметра $w_p$ заменить на выделенное значение $w_p^*$. Эта величина вычисляется по следующей формуле:

$$\chi_p = \left| \frac{\partial H_{мМ}}{\partial w_p}\left(w_p - w_p^*\right) \right|.$$

В рамках другого подхода обучающее множество рассматривают как случайную выборку в пространстве входных параметров. В этом случае показателем значимости по всему обучающему множеству будет служить результат некоторого усреднения по обучающей выборке.

Существует множество способов усреднения. Рассмотрим два из них. Если в результате усреднения показатель значимости должен давать среднюю значимость, то такой показатель вычисляется по следующей формуле:

$$\chi_p = \frac{1}{m}\sum_{q=1}^{m}\chi_p^q .$$

Если в результате усреднения показатель значимости должен давать величину, которую не превосходят показатели значимости по отдельным примерам (значимость этого параметра по отдельному примеру не больше чем $\chi_p$), то такой показатель вычисляется по следующей формуле:

$$\chi_p = \max_q \chi_p^q .$$

## 5.13 Гибридные интеллектуальные системы

При построении систем распознавания образов, к которым относятся и диагностические системы, необходимо решить три главных задачи:

- отбор информативных признаков (традиционно – оффлайновые методы);
- моделирование объекта (процесса);
- принятие решения – выдача определенного управляющего воздействия.

Традиционно большинство систем распознавания для решения конкретной задачи используют какой-либо один метод классификации. Однако ни один метод классификации не является универсальным и абсолютно надежным, и разные методы, как правило, основываются на разной философии. Вследствие этого, методы, обеспечивающие одинаковую или близкую надежность классификации для всей обучающей и/или контрольной выборки в целом, для отдельных экземпляров могут давать разные результаты (это особенно характерно для экземпляров, находящихся на стыках границ классов).

Естественно, что подобные системы, построенные на основе одного классификатора, не будут удовлетворять высоким требованиям надежности, предъявляемым во многих областях промышленности. Поэтому крайне важно создавать гибридные системы классификации, интегрирующие различные модели и методы классификации.

В распознавании образов объединение множества классификаторов различной природы рассматривается как новое направление для разработки высоконадежных

систем распознавания. При этом исходят из того, что комбинация нескольких классификаторов может давать гораздо лучший результат по сравнению с классификацией на основе какого-нибудь одного классификатора - для отдельных прикладных задач каждый из классификаторов может достигать различной степени успеха, но ни один из них не может быть совершенным или хотя бы удовлетворительным для всех приложений. Результаты работы нескольких классификаторов могут быть объединены для улучшения качества распознавания общей системы классификации.

Важность интеграции классификаторов может быть определена, исходя из следующих соображений:

- задача распознавания на основе интегрированной системы классификации может быть рассмотрена с различных сторон, что обеспечивает получение многостороннего (более объективного) результата;

- объединение классификаторов позволяет разделять многомерный входной вектор на несколько векторов меньшей размерности. Классификаторы по отдельности могут обрабатывать соответствующие векторы малой размерности параллельно. Результаты работы отдельных классификаторов затем могут быть объединены для получения конечного результата.

Метод интегрированной классификации эффективен для решения задач, которые содержат большое количество шумовых данных или имеют большую размерность. Интеграция различных классификаторов особенно полезна при решении задач диагностики, которые требуют высокой надежности распознавания и нечувствительности к шуму.

Но и этого не достаточно, поскольку построение эффективной модели объекта возможно только при наличии оптимального набора признаков, причем, для каждой модели, в общем случае, оптимальные наборы будут различными и одного только предварительного задания набора признаков на основе оффлайновых методов (предварительный физический анализ, планирование и т.п.) будет недостаточно. Поэтому необходимо многократное итерационное повторение комбинации “отбор признаков – построение модели ”.

Однако и этого не достаточно, поскольку, как правило, классификаторы содержат достаточное количество параметров, которые необходимо настроить вручную (для нейросетей – это количество слоев, количество нейронов в слое, структура связей, параметры алгоритма обучения). Для автоматизации этого процесса необходима разработка процедур развития (роста) модели.

Получив систему, состоящую из классификаторов, имеющих оптимальное для данного классификатора количество входов, необходимо оптимизировать их структуру (для НС – удалить (уменьшить) избыточные малоинформативные и усилить информативные связи при помощи алгоритма контрастирования).

Рассмотрев основные этапы диагностики, можно прийти к выводу, что их раздельная реализация неэффективна и для решения большинства прикладных задач целесообразно использовать некую гибридную интегрированную систему, включающую реализации различных блоков для решения рассмотренных задач. Структура такой системы может быть аналогична приведенной на рис. 5.8.

Как видно из рис. 5.12, процесс гибридной диагностики должен включать следующие этапы.

1. Предварительный отбор признаков на основе оффлайновых методов и статистического анализа.

2. Построение модели объекта.

2.1 Создание и обучение многоклассификаторной модели объекта.

2.2 Модификация параметров отдельных классификаторов для улучшения надежности классификации (выращивание НС).

2.3 Оптимизация нейросетевых моделей (отбор признаков и контрастирование).

2.4 Создание и обучение блока-интегратора.

3. Создание и обучение блока принятия решений.

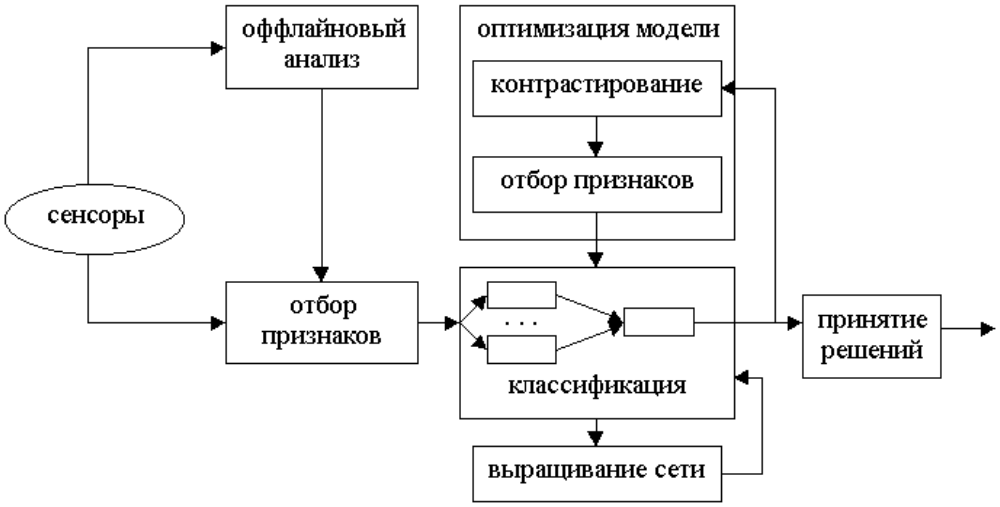



Рис. 5.8 - Обобщенная схема гибридной диагностики.

Для построения гибридной системы диагностики, реализующей рассмотренные этапы и содержащей нейросетевые классификаторы предлагается использовать следующие алгоритмы.

**Алгоритм функционирования гибридной системы диагностики.**

Шаг 1. Задать набор признаков, характеризующих моделируемый объект и обучающую выборку.

Шаг 2. Произвести предварительный отбор информативных признаков.

Шаг 3. Построить и оптимизировать модель объекта.

Шаг 4. Проверить адекватность модели.

Шаг 5. Если модель адекватна, то перейти на шаг 6, в противном случае – изменить параметры модели и перейти на шаг 3 или, если количество итераций шагов 3-4 превышает установленный предел, перейти на шаг 7, выдав сообщение о невозможности построения адекватной модели.

Шаг 6.Построить блок принятия решений.

Шаг 7. Останов.

**Алгоритм построения и оптимизации модели объекта.**

Шаг 0. Задать набор классификаторов: определить используемые модели, начальные параметры моделей и методы обучения.

Шаг 1. Обучить все классификаторы на основе обучающей выборки.

Шаг 2. Для всех экземпляров обучающей выборки последовательно подать на входы классификаторов значения признаков для текущего экземпляра $x^s$, s=1,...,S, где S-количество экземпляров в обучающей выборке, и определить $K_k^s$ - класс s-го экземпляра для k-го классификатора, k=1,..,G, где G – количество классификаторов.

Шаг 3. Набор результатов классификации $K_{вх}=\{K_k^s\}$ для всех экземпляров и классификаторов и набор фактических значений классов $K=\{K^s\}$ для всех экземпляров подать для обучения на входы и выход блока-интегратора, соответственно. Если в качестве блока-интегратора используется НС Кохонена, то подается только набор $K_{вх}$.

Шаг 4. Обучить интегратор моделированию зависимости между классами на выходах классификаторов и фактическим классом.

Шаг 5. Вычислить $I_k$ - вклад k-го классификатора в формирование общего результата, k=1,2,...,G. Если в качестве блока-интегратора используется НС Кохонена, то $I_k$ могут быть определены как информативности входов НС на основе алгоритма отбора информативных признаков.

Шаг 6. Для классификаторов, у которых $I_k > \frac{\sum_{k=1}^{G} I_k}{G}$, выполнить процедуру оптимизации структуры (контрастировать связи и / или оценить информативность и осуществить отбор признаков).

Для классификаторов, у которых $I_k \le \frac{\sum_{k=1}^{G} I_k}{G}$, если возможно, выполнить процедуру выращивания, иначе исключить классификатор из системы.

Шаг 7. Если значимость классификатора в процессе P повторений шагов 1-6 меньше определенного заданного числа, то исключить классификатор из системы, иначе – перейти на шаг 1.

Выход из цикла, определяемого шагами 1-7, осуществляется, если будет выполняться хотя бы одно из условий:

1) количество классификаторов $G \le 2$;

2) достигнуты пределы увеличения / сокращения структуры для всех классификаторов (дальнейшее применение алгоритмов контрастирования и выращивания нецелесообразно);

3) требуемая точность классификации достигнута;

требуемая точность классификации недостигнута в течение заданного времени.

**Алгоритм контрастирования многослойной нейронной сети.**

Шаг 0. Сохранить текущий набор значений весов и порогов сети.

Шаг 1. Задать или вычислить по определенному правилу граничное значение значимости весов $\overline{w}$. В качестве такого значения, например, можно использовать среднее арифметическое значение модулей весов и порогов сети.

Шаг 2. Все веса $|w_j^{(\mu,i)}| < \overline{w}$, где $w_j^{(\mu,i)}$ – вес j-го входа i-го нейрона $\mu$-го слоя НС, уменьшить на основе правила $w_j^{(\mu,i)} = R^-(w_j^{(\mu,i)})$, где $\mu$, i, j удовлетворяют заранее определенным правилам и ограничениям.

Шаг 3. Все веса $|w_j^{(\mu,i)}| \ge \overline{w}$ увеличить на основе правила $w_j^{(\mu,i)} = R^+(w_j^{(\mu,i)})$, где $\mu$, i, j удовлетворяют заранее определенным правилам и ограничениям.

Шаг 4. Для всех экземляров обучающей выборки $x^s$, s=1,...,S вычислить значение суммарной ошибки сети Error.

Шаг 5. Если Error<MaxError, где MaxError – некоторое заданное максимальное граничное значение ошибки сети, то перейти на шаг 0, в противном случае – перейти на шаг 6.

Шаг 6. Восстановить предыдущие значения весов и порогов сети.

Шаг 7. Останов.

В качестве правил R(w) предлагается использовать:

$$R^-(w) = \begin{cases} w - \alpha, \text{если } w > 0, w \ge \alpha; \\ w + \alpha, \text{если } w < 0, w \le -\alpha; \\ 0, \text{если } (w = 0) \text{ или } (w > 0 \text{ и } w < \alpha) \text{ или } (w < 0 \text{ и } w > -\alpha). \end{cases}$$

$$R^+(w) = \begin{cases} w + \alpha, \text{если } w > 0; \\ w - \alpha, \text{если } w < 0; \\ 0, \text{если } w = 0. \end{cases}$$

или

$$R^-(w)=\begin{cases} w\alpha, & 0<\alpha<1, \text{если } w>\xi; \\ 0, & \text{если } w\le\xi. \end{cases} \quad R^+(w)=w/\alpha, \ 0<\alpha<1,$$

или

$$R^-(w)=w(1-P(w)),\ R^+(w)=w/P(w),$$

где $$P(w)=\begin{cases} -\frac{2}{\pi}\operatorname{arctg}\frac{1}{w}, & \text{если } w<0; \\ \frac{2}{\pi}\operatorname{arctg}\frac{1}{w}, & \text{если } w>0; \\ 1, & \text{если } w=0, \end{cases}$$

где $\alpha$, $\xi$ - заранее заданные константы.

При задании этих констант следует руководствоваться следующими соображениями:

1) слишком большие значения констант могут существенно ухудшить результат и затруднить процесс обучения;

2) слишком малые значения констант могут привести к существенному увеличению времени работы алгоритма контрастирования;

3) чем меньше значения констант, тем меньше будет дополнительная ошибка классификации, возникающая после выполнения алгоритма контрастирования.

В отличие от большинства других алгоритмов контрастирования, предполагающих удаление малоинформативных связей, данный алгоритм осуществляет последовательное увеличение больших весов и уменьшение малых, что позволяет более гибко осуществлять процесс контрастирования.

**Алгоритм выращивания сети.**

Шаг 0. Инициализация: $\mu=1$.

Шаг 1. Если количество нейронов в $\mu$-ом слое сети $N_\mu$ меньше определенной величины $\beta$, то увеличить количество нейронов в $\mu$-ом слое в соответствии с правилом $N_\mu = N^+(N_\mu)$, где $N^+(N_\mu)$ может определяться одним из выражений:

$$N^+(N_\mu)=\begin{cases} N_\mu+\gamma, & \text{если } (N_\mu+\gamma)<\beta; \\ \beta, & \text{если } (N_\mu+\gamma)\ge\beta. \end{cases}$$

$$N^{+}(N_{\mu}) = \begin{cases} \text{round}(N_{\mu}\gamma), \gamma > 1, \text{ если round}(N_{\mu}\gamma) < \beta; \\ \beta, \text{ если round}(N_{\mu}\gamma) \geq \beta. \end{cases}$$

где $\beta$ и $\gamma$ - заранее заданные константы, и перейти на шаг 4, в противном случае – перейти на шаг 2.

Шаг 2. Если $\mu$<M, то положить: $\mu=\mu+1$ и перейти на шаг 4, в противном случае – на шаг 3.

Шаг 3. Если $\mu$=M и M<$M_{max}$, где $M_{max}$ – максимально допустимое количество слоев НС, то положить: M=M+1, $N_M$=1 и перейти на шаг 2, в противном случае – на шаг 4.

Шаг 4. Останов.

При задании констант $\beta$ и $\gamma$ следует учитывать следующие соображения:

1) при увеличении значения $\gamma$ сеть будет расти быстрее;

2) значения $\beta$ и $\gamma$ должны быть больше нуля;

3) константа $\beta$ должна быть целым числом;

4) если константа $\gamma$ - не является целым числом, то значение $N^{+}(N_{\mu})$ должно округляться.

5) с увеличением значения $\beta$ снижаются возможности роста сети.

**Построение блока принятия решений.**

Основная задача блока принятия решений – обработка результата классификации для данного экземпляра и выдача соответствующих управляющих сигналов, определенных для данной ситуации.

Блок принятия решений можно строить различными способами:

1) на основе жестко заданных правил «Если (условие), то (действие)».

2) на основе нечеткой логики.

3) на основе нейросетевых моделей.

Последний вариант является более предпочтительным в тех случаях, когда нет заранее формализованных правил принятия решений, а есть только некоторое множество кортежей (пар данных), состоящих из описания ситуации и решения, которое целесообразно принять в такой ситуации.

Схема интегрированной системы нейросетевой классификации показана на рис. 5.9 Она состоит из подсистемы классификации и блока интеграции.

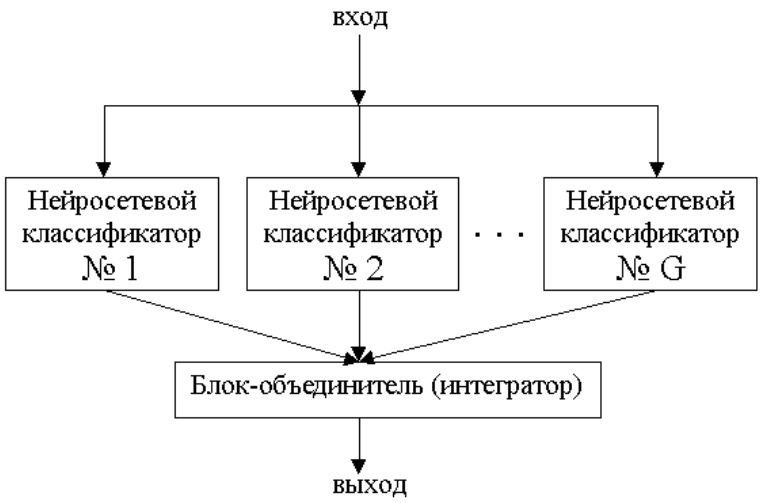



Рис. 5.9 - Интегрированная система классификации.

На входы классификаторов подаются входные наборы данных, на выходы классификаторов – выходные наборы данных. Для каждого классификатора производится обучение, после которого по данным на входе он должен на выходе выдавать значения незначительно отличающиеся от желаемых. Выходы отдельных обученных классификаторов подаются на входы блока-объединителя. Количество возможных классификаторов определяется входной размерностью блока-объединителя.

В качестве блока-интегратора можно использовать нейросетевые, статистические, эвристические и другие модели. Рассмотрим наиболее просто реализуемые и, в тоже время, наиболее обоснованные модели.

Карта признаков самоорганизации Кохонена имеет ряд свойств, которые позволяют использовать ее как блок интеграции в методе интегрированной классификации. Использование самоорганизующейся карты Кохонена обусловлено подобием ее работы некоторым механизмам функционирования человеческого мозга. Мозг организован так, что различные сенсорные входы представляются топологически упорядоченными вычислительными картами. В частности, сенсорные входы осязательного, визуального и акустического типа отображаются на

различные области коры головного мозга топологически упорядоченным способом. Таким образом, карта Кохонена представляет собой базовый конструктивный блок в инфраструктуре обработки данных возбужденной системы.

Как известно, обработка информации в человеческом мозге связана с извлечением значений из данных внешнего мира и последующим их объединением. Например, распознавание объекта часто определяется интеграцией систем зрения, осязания, слуха и обоняния. Карта Кохонена подобна биохимической отображающей модели мозга. Доказано, что объединение с использованием НС Кохонена соответствует человеческому способу распознавания. Следовательно, карты Кохонена целесообразно использовать в качестве блока-интегратора.

Карты Кохонена обладают рядом свойств, желательных для блока-интегратора:

- они способны аппроксимировать входное пространство;
- карты Кохонена осуществляют топологическое упорядочение данных;
- НС Кохонена учитывает статистические характеристики, присущие данным.

Последнее свойство особено важно для решения задачи интеграции классификаторов. НС Кохонена объединяет выходы отдельных классификаторов согласно распределениям их вероятностей.

Использование НС Кохонена в качестве блока-интегратора делает метод интегрированной классификации нечувствительным к корреляции отдельных классификаторов, что объясняется конкурирующим принципом работы НС.

Другой моделью для блока-интегратора может служить усредненная взвешенная сумма выходов классификаторов, учитывающая некоторым образом их значимость (чем выше безошибочность работы соответствующего классификатора, тем выше его значимость). Если на выходах классификаторы будут выдавать значения 0 или 1, то класс на выходе блока-интегратора K можно будет определить по формуле:

$$K = \mathrm{round}\left(\beta \frac{\sum_{i=1}^{G} \alpha_i K_i}{\sum_{i=1}^{G} \alpha_i}\right),$$

где G - количество классификаторов, $\alpha_i$ - значимость выхода i-го классификатора, $K_i$ - значение на выходе i-го классификатора, $\beta$ - некоторый коэффициент, позволяющий регулировать влияние малозначимых классификаторов на общий результат, round – функция округления.

Значения $\alpha_i$ могут быть найдены разными способами. Например, в качестве $\alpha_i$ можно использовать вероятности правильной классификации:

$$\alpha_i = \frac{N_{прав.}}{s},$$

где s-количество экземпляров обучающей выборки; $N_{прав.}$ – количество экземпляров обучающей выборки, для которых произведена правильная классификация.

При выборе значения коэффицента $\beta$ следует учитывать следующие соображения:

1) значения коэффициента $\beta$ должны быть больше нуля;

2) в простейшем случае, когда нет необходимости регулировать влияние малозначимых классификаторов, коэффициент $\beta$ можно принять равным единице;

3) с увеличением коэффициента $\beta$ вклад малозначимых классификаторов будет существенно снижаться.

# ГЛАВА 6. ПРОГРАММНЫЕ СРЕДСТВА ДИАГНОСТИКИ И ПРОГНОЗИРОВАНИЯ

## 6.1 Автоматизированная система «Диагностика»

Автоматизированная система (АС) «Диагностика» представляет собой комплекс программ, предназначенных для автоматизации отдельных этапов обработки диагностической информации, и включает в себя подсистемы:

- предобработки и визуализации данных;
- сокращения размерности данных;
- топологической диагностики;
- нейросетевой диагностики;
- обучения теории диагностики.

Все подсистемы АС "Диагностика" представляют собой самостоятельные программы и могут использоваться как по отдельности, так и совместно в любом сочетании. Это позволяет существенно экономить ресурсы компьютера, поскольку на каждом этапе диагностики может использоваться всего лишь одна из подсистем, а остальные можно не устанавливать.

Особенностью АС "Диагностика" является ее модульность и открытость, что позволяет добавлять новые методы обработки диагностической информации.

Входные данные программы сохраняются в файлах различных форматов (текстовом, таблиц Exel, баз данных Database, Paradox, Microsoft Access и др.). Это позволяет снять ограничение размера файлов данных, стандартизировать процесс ввода данных, позволяет методами SQL запросов проводить выборки данных по заданным параметрам, а также даёт возможность использовать, при необходимости, данные, подготовленные в других программах.

При создании подсистем АС "Диагностика" использовались различные языки программирования и среды разработки интерфейса, однако все программы совместимы между собой по данным.

Благодаря наличию в АС "Диагностика" специальных средств преобразования данных из различных форматов она может работать совместно с математическим пакетом MATLAB версии 5.0 (и выше) фирмы MathWorks Inc. (США), аппаратно-

программным измерительным комплексом ПОС "Вояж" НПП "Мера" (Россия), а также теми пакетами, которые поддерживают хотя бы один из популярных форматов баз данных или текстовый формат.

### 6.1.1 Подсистема предобработки и визуализации данных

Подсистема предобработки и визуализации данных АС "Диагностика" обеспечивает представление на экране монитора данных в трех видах:

- визуализация признаков экземпляров с однотипными упорядоченными признаками;
- визуализация признаков нескольких экземпляров с однотипными упорядоченными признаками;
- плоскостная визуализация признаков экземпляров с разнотипными признаками.

Исходя из этого в программе были разработаны три модуля, которые отвечают за отображение разных типов графиков на диаграмме (рис.6.1).

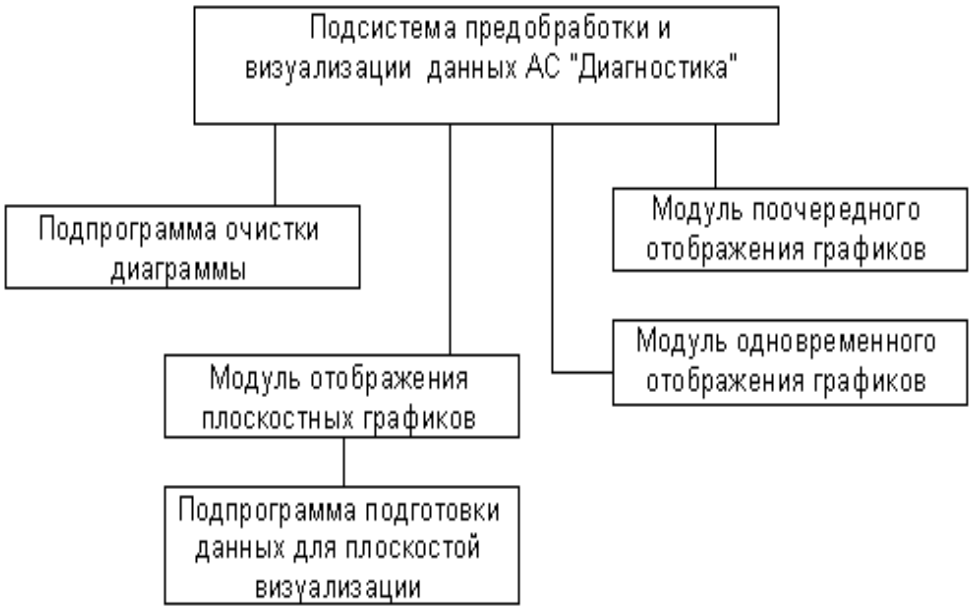



Рис. 6.1 – Структура подсистемы предобработки и визуализации данных АС "Диагностика"

За визуализацию признаков экземпляров с однотипными упорядоченными признаками отвечает модуль поочередного отображения графиков. Визуализацию признаков нескольких экземпляров выполняет модуль отображения всех графиков. А плоскостную визуализацию признаков экземпляров с разнотипными признаками выполняет модуль отображения плоскостных графиков.

Также в программу входит подпрограмма очистки диаграммы, которая выполняет очистку диаграммы при переходе от одного типа визуализации к другому. В модуль отображения плоскостных графиков входит подпрограмма подготовки данных для плоскостной визуализации, задача которой создавать массив из необходимых признаков экземпляров.

Входной информацией для программы служат файлы баз данных. Для того, чтобы программный комплекс мог работать с файлами баз данных, необходимо наличие на ЭВМ установленной системы Borland Database Engine (Borland Database Engine), которая поддерживает высокопроизводительный 32-разрядный доступ к базам данных dBASE, Paradox: Sybase. Oracle, DB2, Microsoft SQL Server, Informix, InterBase и Local InterBase.

Функциональная схема программы представлена на рис. 6.2.

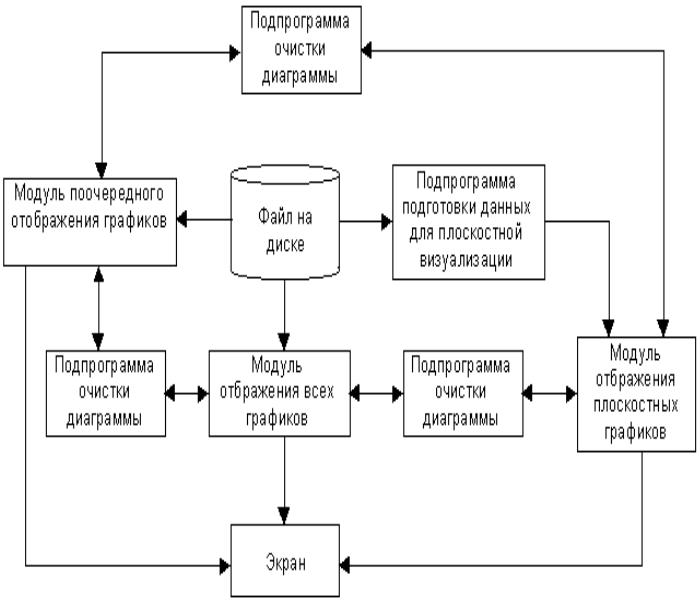



Рис. 6.2 - Функциональная схема программы.

При запуске программы до появлению главного окна запускается диалоговое окно открытия файла базы данных. После выбора файла активизируется таблица, что дает возможность обращаться к ней для считывания или записи данных.

Команда "Открыть базу данных" в меню "Файл" выполняет подпрограмму очистки диаграммы и запускает диалоговое окно открытия файла базы данных.

Команда "Выход" в меню "Файл" закрывает таблицу и саму программу.

Модуль поочередного отображения графиков рисует график упорядоченных однотипных признаков для одного экземпляра на диаграмме.

Модуль отображения всех графиков работает также, как и модуль поочередного отображения графиков, только он на диаграмме одновременно отображает графики для всех экземпляров с однотипными упорядоченным признаками.

Модуль отображения плоскостных графиков и запускает подпрограмму очистки диаграммы.

Потом модуль создает новую таблицу и выполняет подпрограмму подготовки данных для плоскостной визуализации.

Подпрограмма подготовки данных для плоскостной визуализации проводит подготовку данных (на основе выбранных пользователем признаков заполняет таблицу созданную модулем отображения плоскостных графиков).

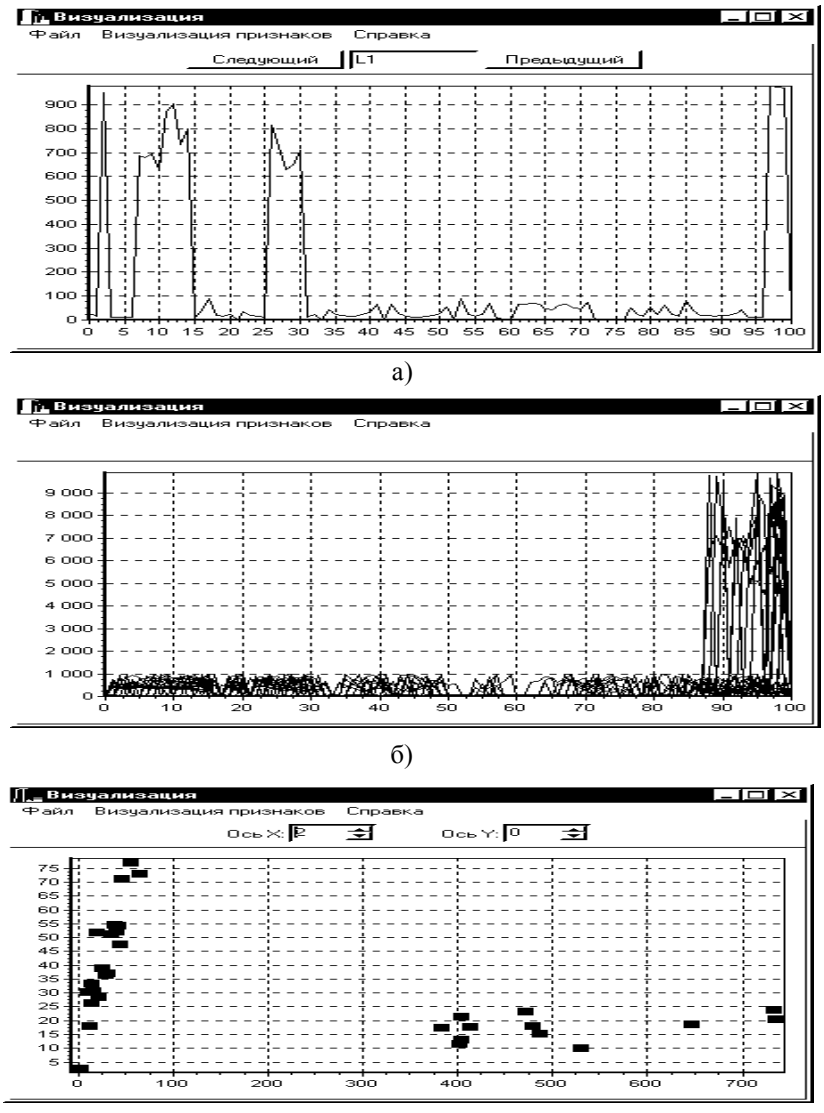
Сразу после запуска программы появляется диалоговое окно "Открытие базы данных". В нем пользователь должен выбрать файл базы данных, с которым будет работать программа.

Выбрав файл базы данных, пользователь должен нажать кнопку "Открыть". После этого появится главное окно программы. В пункте меню "Визуализация признаков" пользователь может выбрать одну из трех представленных там команд, которая запустит соответствующий модуль отображения графиков.

Если при запуске программы в диалоговом окне будет нажата кнопка "Отмена" то при выборе вида визуализации программа автоматически предложит открыть файл базы данных.

По окончании работы с программой пользователь должен выбрать команду "Выход" в пункте меню "Файл"

На рис. 6.3 изображены интерфейсные формы для различных видов визуализации данных.




а)

б)

в)

Рис. 6.3 - Интерфейсные формы: а) визуализация экземпляра с однотипными упорядоченными признаками; б) визуализация нескольких экземпляров с однотипными упорядоченными признаками; в) плоскостная визуализация признаков экземпляров.

### 6.1.2. Подсистема сокращения размерности данных

Подсистема сокращения размерности данных АС "Диагностика" содержит процедуры, реализующие методы оценки информативности и отбора признаков, а также алгоритм разбиения исходной выборки на обучающую и тестовую.

Структурная схема подсистемы представлена на рис. 6.4.

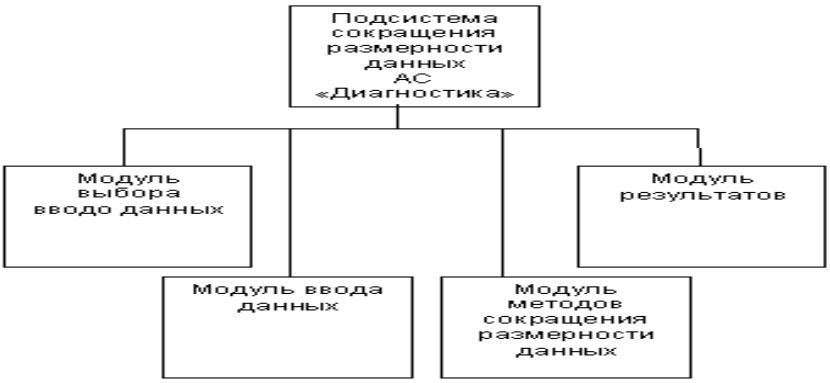



Рис. 6.4 – Структура подсистемы сокращения размерности.

Модули «Выбор ввода данных», «Ввод данных» и «Результаты» содержат средства организации пользовательского интерфейса.

Модуль «Методы сокращения размерности данных» содержит процедуры, реализующие алгоритмы оценки информативности и отбора признаков, а также алгоритм формирования обучающей и тестовой выборок.

Функциональная схема подсистемы показана на рис. 6.5.

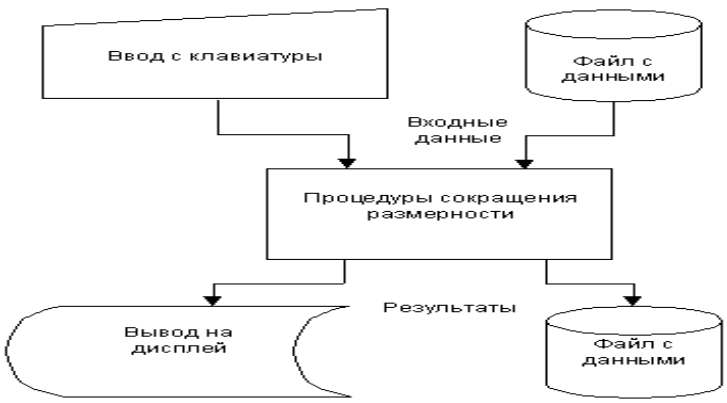



Рис. 6.5 – Функциональная схема подсистемы сокращения размерности.

### 6.1.3 Подсистема топологической диагностики

Подсистема топологической диагностики АС “Диагностика” реализует методы потенциальных функций и метрической классификации.

Структура программы показана на рис. 6.6.


Рис. 6.6 - Структура подсистемы топологической диагностики

Модуль обучения предназначен для выполнения процесса обучения по выборки данных. Он ориентирует всю программу на определенный объект системы при помощи расчета координат центров каждого класса и сохраняет во временный файл Temp.ini.

Модуль распознавания предназначен для классификации экземпляров выборки. Он находит расстояния от экземпляров выборки до центров каждого класса и значения суммарных потенциалов для них (используя сохраненные в координаты центров классов во временном файле).

Модуль отчета предназначен для формирования отчетов. Полученные результаты этот модуль может сохранить в файл или вывести на печать – по желанию пользователя.

Функциональная схема подсистемы представлена на рис. 6.7.

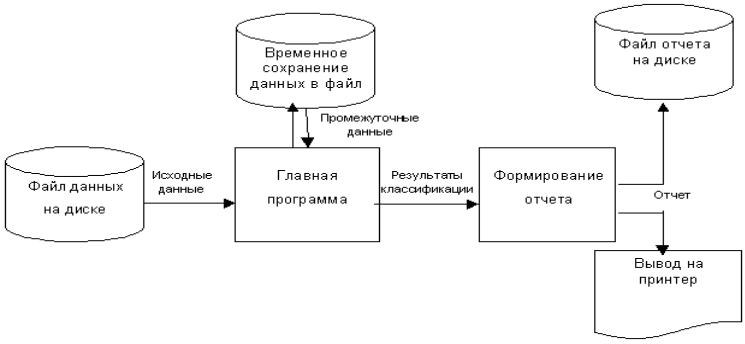



Рис. 6.7 - Функциональная схема подсистемы топологической диагностики.

Данные представляют собой выборку, которая поступает в главную программу подсистемы. В результате обучения формируются координаты центров сосредоточения экземпляров обучающей выборки. Эти данные сохраняются во временный файл и используются в дальнейшем при распознавании. После этого результаты диагностики формируются в отчеты и сохраняются на диске в виде файлов или выводятся на печать.

Пользовательский интерфейс реализован с учетом эргономических требований посредством экранных форм, созданных в среде Delphi. Главная экранная форма представлена на рис.6.8.

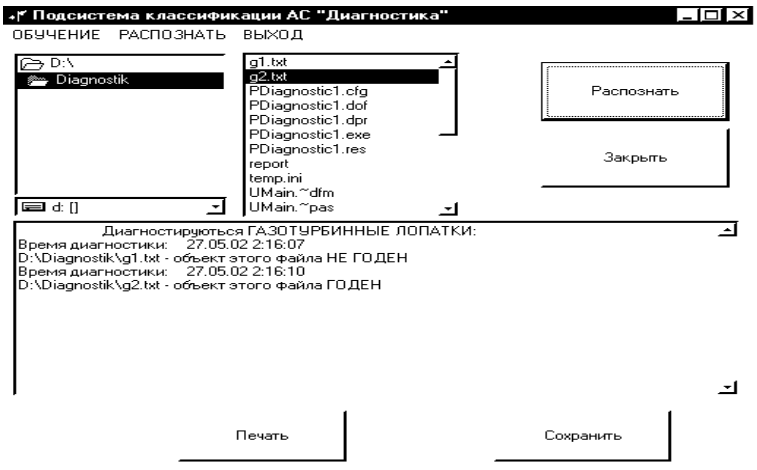



Рис. 6.8 - Главная экранная форма (процесс распознавания экземпляров).

Главное меню программы Diagnostic.exe содержит следующие пункты:

ОБУЧЕНИЕ – выполняет процедуру соответствующую названию. При помощи удобного и доступного интерфейса выбирается файл для обучения данной выборки. Этот файл должен быть сформирован заранее и содержать данные классов с их признаками.

РАСПОЗНАНИЕ – выполняет процесс диагностики на пригодность объектов. При помощи удобного и доступного интерфейса выбирается файл для распознавания. Этот файл должен содержать набор признаков по которым и будет сделан вывод о пригодности экземпляра. Результат тут же отобразиться на экране и может быть сохранен в файл или же выведен на печать.

ВЫХОД – осуществляет выход из программы.

### 6.1.4 Подсистема нейросетевой диагностики

Подсистема нейросетевой диагностики АС “Диагностика” из 5 модулей. Структура подсистемы показана на рис. 6.9.

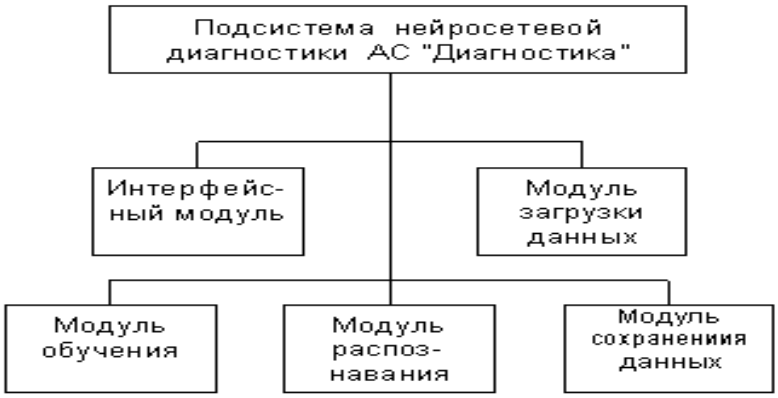



Рис. 6.9 – Структура подсистемы нейросетевой диагностики

Интерфейсный модуль представляет собой набор процедур для работы пользователя с экранными формами. Модуль обучения производит построение и обучение нейросетевой модели.

Модуль распознавания проводит расчет прогнозируемых параметров по обученной нейросетевой модели.

Модуль сохранения данних сохраняет полученные данные в текстовом файле, а модуль загрузки данных в свою очередь– загружает.

Работа пользователя с программой осуществляется в интерактивном режиме, с использованием многоуровневого меню. Каждый пункт меню объединяет определенный набор функций, которые реализуют общую задачу.

### 6.1.5. Подсистема обучения теории диагностики

Подсистема обучения теории диагностики сочетает в себе электронный учебник по теории технической диагностики, средства тестирования знаний и электронный справочник по методам, реализованным в АС "Диагностика" и ее подсистемах.

Подсистема представляет собой универсальное средство для построения обучающих систем, поскольку она без переделки может быть настроена пользователем на различные предметные области, а также предусматривает легкое дополнение и модификацию содержащегося материала.

Учебный материал представляется в виде кадров (небольших порций информации), которые хранятся в виде файлов в формате Microsoft Word 97 для Windows. Это позволяет легко редактировать и дополнять информацию, а также за счет средств пакета MS Office включать внутрь учебного материала не только текст, но и формулы, рисунки, графики, таблицы, звуковые и видео-объекты, а также объекты других приложений.

### 6.1.6 Диагностический программный комплекс

Исторически созданию АС "Диагностика" предшествовала многолетняя работа авторов по разработке программных средств, реализующих отдельные этапы обработки диагностической информации. Совокупность разработанных ранее программных средств диагностики и прогнозирования составила диагностический программный комплекс.

Особенностью диагностического программного комплекса является его модульность и открытость, что позволяет добавлять в него новые методы диагностики. Учитывая автономность и мобильность подсистем диагностического программного комплекса их можно также рассматривать как подсистемы АС "Диагностика".

Классы решаемых задач и структура диагностического программного комплекса показаны на рис. 6.10.

Процесс диагностики в программном комплексе разбит на отдельные этапы: планирование эксперимента, отбор информативных признаков, факторный анализ, построение решающего (или аппроксимирующего) правила, оптимизация решающего (аппроксимирующего) правила.

Для повышения эффективности комплекса каждый этап диагностики можно выполнить различными методами, при разумном подборе которых можно достигнуть высокой точности диагностики и (или) минимальных временных затрат.

Схема фукнкционирования диагностического программного комплекса показана на рис. 6.11

Обучающая выборка поступает на вход диагностического программного комплекса, далее производится отбор информативных признаков, после чего выполняется построение решающего (аппроксимирующего) правила, которое оптимизируется и проверяется на адекватность моделируемой зависимости. Построенная модель может быть использована для прогнозирования параметров или классификации объектов диагностики.

Входные данные могут быть представлены в файлах следующих форматов: текстовом, таблиц Exel, баз данных Database, Paradox, Microsoft Access и др.

Выходные данные представляются в текстовом формате, а также в форматах тех же баз данных, что и входные данные. Кроме того выходные данные (отчеты) могут быть скопированы в область обмена ОС Windows.

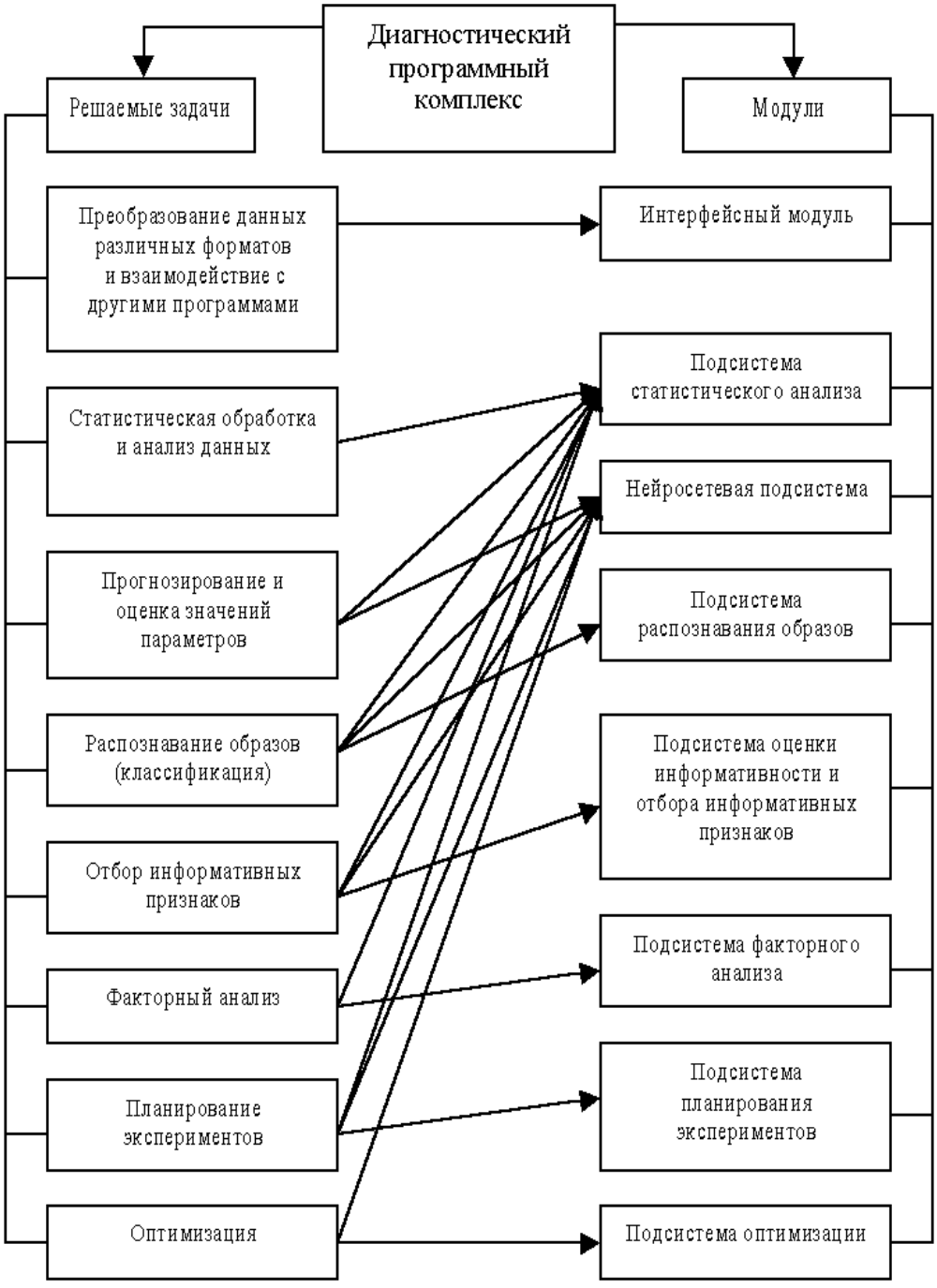



Рис. 6.10 - Классы решаемых задач и структура диагностического программного комплекса.

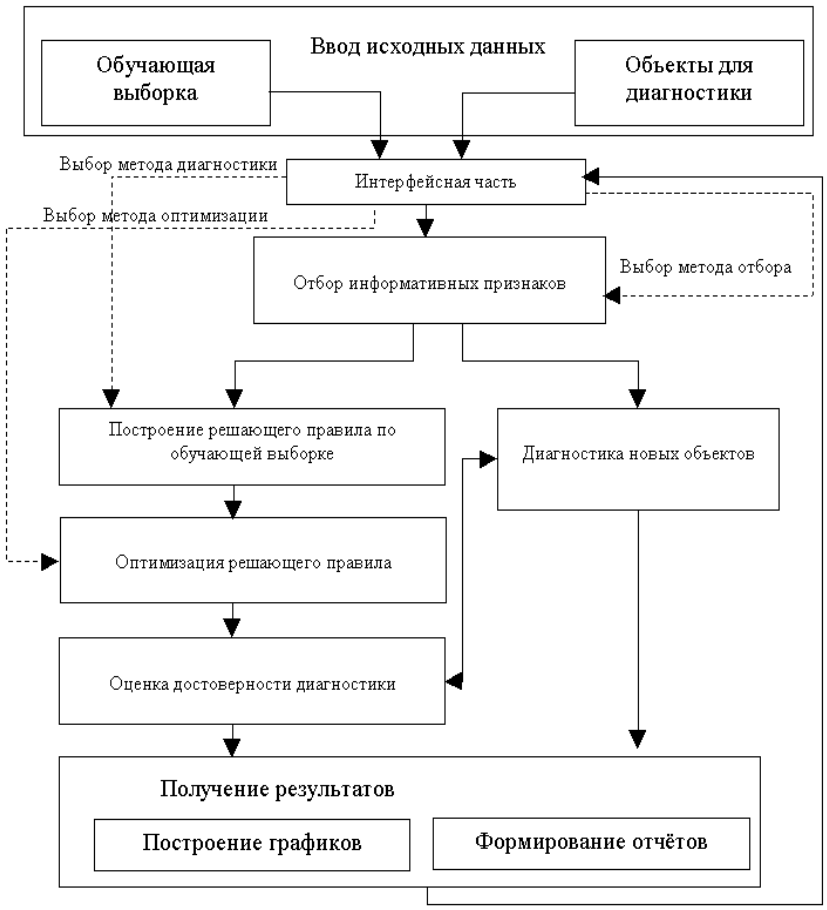



Рис 6.11 - Схема функционирования диагностического программного комплекса.

Подсистемы диагностического программного комплекса представляют собой отдельные программы, которые могут работать независимо друг от друга. При этом они совместимы по данным не только между собой, но и с АС "Диагностика" и могут рассматриватьтся как ее подсистемы. Поэтому представляется целесообразным привести обзор основных возможностей и характеристик подсистем диагностического программного комплекса.

**Подсистема планирования экспериментов** содержит статистические процедуры, позволяющие оказывать помощь в планировании активного полного и дробного факторного экспериментов, строить на основе экспериментальных данных статистические модели и проверять адекватность полученных моделей.

Подсистема отбора информативных признаков **предназначена для определения информативности признаков, характеризующих сложные**

**объекты и процессы, и получения набора признаков, обладающих наибольшей информативностью.**

В программе реализованы следующие методы отбора информативных признаков: метод полного перебора, метод сокращенного перебора с добавлением признаков, метод сокращенного перебора с исключением признаков, метод комбинироавнного перебора, метод случайного баланса и пошаговая регрессия. алгоритм полного перебора;

Алгоритм полного перебора всех возможных комбинаций признаков является наиболее точной процедурой выбора информативных признаков, но его целесообразно применять, если исходная совокупность признаков невелика (меньше 10).

Если число признаков велико и они коррелированны, то для выделения наиболее информативных признаков целесообразно использовать метод случайного баланса.

Еще одним методом отбора информативных признаков является пошаговая регрессия, когда независимые переменные одна за другой включаются в подмножество согласно предварительно заданному критерию. В то же время некоторая переменная может быть удалена из набора. Совокупность критериев, определяющих, какие переменные включать, заменять и удалять, называется пошаговой процедурой.

Подсистема отбора признаков состоит из целого ряда алгоритмов, отвечающих за выполнение отдельных методов отбора информативных признаков и реализацию сервисных функций; и выполняет следующие функции:

- ввод входных данных (файла базы данных, выбор множества признаков, выбор вектора класса или параметра);
- предоставляет средства для просмотра файла базы данных и редактирования его записей;
- осуществляет выбор одного или нескольких методов отбора наиболее важных признаков;

-удаляет по желанию пользователя неинформативные признаки из базы данных;

-сравнивает различные методы отбора по скорости вычисления;

-просматривает созданные файлы отчетов;

-выводит на печать файлы отчетов.

Подсистема отбора признаков может работать как самостоятельно, так и в составе других программных комплексов, например, в составе диагностического программного комплекса.

На рис. 6.12 приведена структура системы отбора информативных признаков.

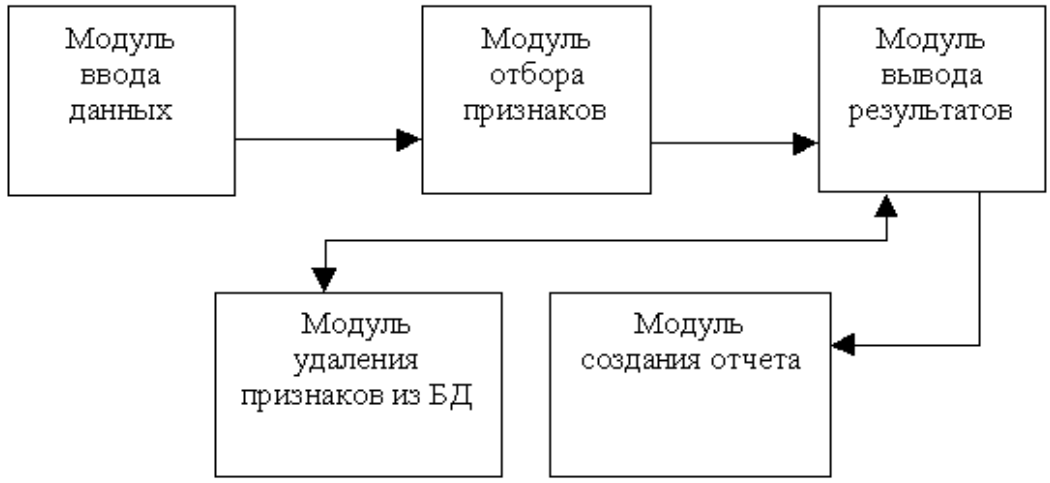



Рис. 6.12 – Структура подсистемы отбора информативных признаков

Модуль ввода данных осуществляет загрузку исходных данных:

-выбор файла базы данных;

-выбор множества признаков;

-выбор вектора класса или параметра (в зависимости от используемого метода);

-задание граничного значения параметра .

В каждом файле данных хранится информация об обучающей выборке для изделий одного типа. Программа позволяет просматривать и редактировать существующие файлы баз данных.

Алгоритмы отбора информативных признаков объединены в один модуль. Это позволяет использовать следующие преимущества:

- при выборе нескольких алгоритмов отбора, пользователю достаточно один раз указать файл данных;
- для выбранного файла данных один раз указать вектора признаков, класса и параметров;
- один раз просчитываются необходимые для всех методов дисперсии, математические ожидания, среднеквадратические отклонения, что позволяет сократить время расчета;
- отсутствие передачи данных из программы в программу сокращает время работы методов.

Программа предоставляет возможность выбора одного или нескольких методов отбора признаков одновременно. Объединением методов в одном модуле достигается наибольшая эффективность отбора, а также использование одних и тех же данных для различных методов.

Данная программа не ограничивает количество методов отбора признаков, входящих в систему. Подключения новых методов осуществляется с помощью файла конфигурации System.cfg, который находится в основном каталоге системы. Файл конфигурации представляет собой текстовый файл, состоящий из строк вида:

“Название метода – расположение на жестком диске”.

Вся входная информация для новых методов отбора хранится в текстовом файле “Input_file.txt”. Этот файл содержит сведения о :

- выбранном файле базы данных;
- номерах столбцов выбранных пользователем признаков;
- номере столбца вектора класса;
- номере столбца вектора параметра;
- граничном значении параметра.

Всю выходную информацию методы, не входящие в данную систему, должны хранить в текстовом файле “Output_file.txt”. При этом первые строки содержат информацию об информативных признаках и времени выполнения метода в формате ЧЧ:ММ:СС.

После выполнения отбора признаков, можно удалить неинформативные признаки из базы данных.

По желанию пользователя, результаты работы программы могут быть зафиксированы в файле отчета с расширением *.txt, структура которого имеет следующий вид:

- название метода;
- имя файла, с которым работал данный метод;
- время работы метода;
- список выбранных признаков;
- список неинформативных признаков, отобранных данным методом или (в случае выбора метода полного перебора) информативные наборы с вероятностями принятия ошибочных решений;
- информация о том удалены ли неинформативные признаки из базы данных.

Основное окно программы приведено на рис.6.13.

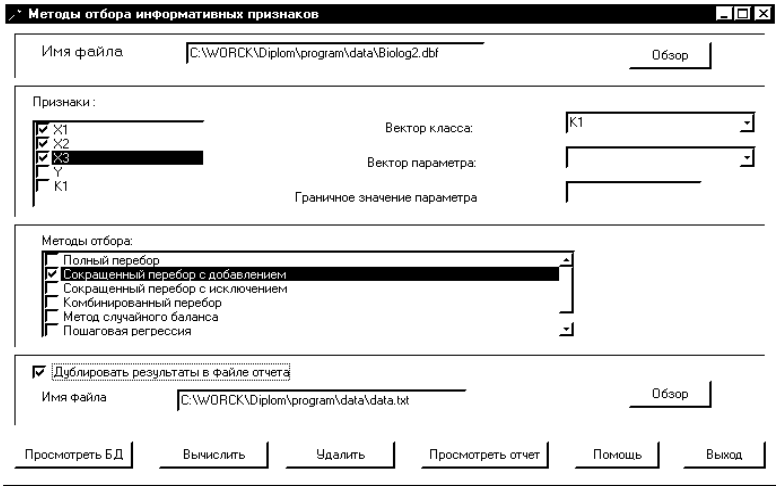



Рис. 6.13 – Основное окно программы.

С помощью клавиатуры или кнопки “Обзор” можно выбрать файл базы данных. Введенный файл можно просмотреть, нажав на кнопку “Просмотреть БД”. В этом же окне можно отредактировать таблицу, не изменяя ее структуры.

После ввода файла необходимо указать признаки, с которыми продолжать дальнейшую работу. Выбрать один или несколько методов отбора. Задать вектор класса в том случае, если выбран хотя бы один из следующих методов: метод полного перебора, метод сокращенного перебора с исключением параметров, метод сокращенного перебора с добавлением параметров, комбинированный метод перебора. Для выше перечисленных методов вектор класса можно заменить вектором параметра, но при этом необходимо указать граничное значение параметра. Если выбран метод случайного баланса или пошаговая регрессия, то необходимо указать вектор параметра. Если для выбранного метода недостаточно введенной информации (не задан вектор класса или граничное значение параметра), то программа выдаст сообщение об ошибке.

Вычисления начинаются лишь после нажатия кнопки “Вычислить”. При этом на экране появляется сообщение, которое указывает вычисляемый в данный момент метод и текущий процент проделанной работы от длительности всего вычисления.

Если выбран только один метод отбора, то программа выведет окно результата, представленное на рис. 6.14, с запросом: удалять ли неинформативные признаки из БД. Если будет нажата кнопка “Да”, то программа удалит эти признаки, создав при этом копию исходного файла, дописав в конец его имени “_bak” (например, был задан файл Primer.dbf, его копией будет файл Primer_bak.dbf).

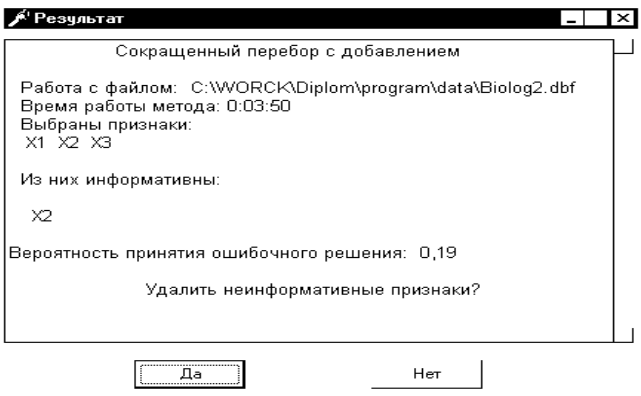



Рис. 6.14 – Окно результата в случае выбора одного метода отбора.

Если будет создаваться файл – отчет, то в него будет помещена информация о том, удалены ли признаки из файла данных или нет.

По желанию пользователя, результат работы программы сохраняется в файле отчета с расширением *.txt (имя файла указывает пользователь).

Для этого надо до вычислений указать программе на необходимость “Дублировать результаты в файле отчета” и с помощью клавиатуры или кнопки “Обзор” указать имя файла отчета с полным путем

**Подсистема факторного анализа** предназначена для проведения факторного анализа над статистическими данными, представленными в виде файлов на диске.

Выходными данными являются результаты вычислений методами факторного анализа: матрица корреляций, факторное отображение, остаточная матрица корреляций, факторная структура до вращения, факторная структура после вращения, факторные оценки, а также графическая информация, выдаваемая по данным факторной структуры до вращения.

Результатом работы программы является отчёт по результатам факторного анализа в виде таблиц и графиков.

На рис. 6.15 представлена схема функционирования программы.

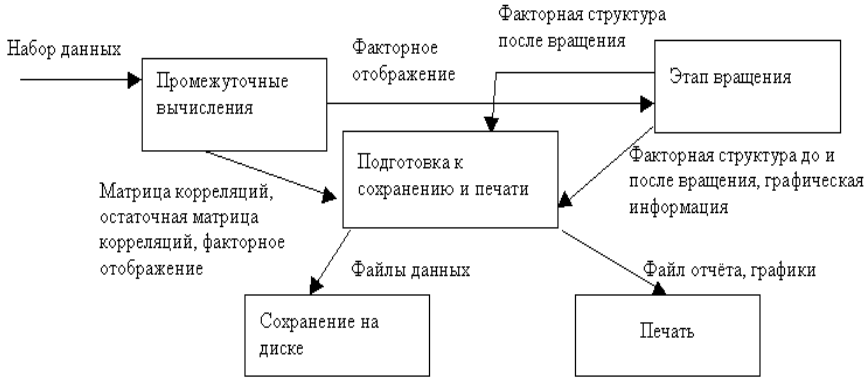



Рис. 6.15 - Схема функционирования программы.

**Общий алгоритм подсистемы, реализованной в ходе выполнения данной работы, можно представить в таком виде:**

**-производится настройка на формат данных;**

**-выбираются данные, по которым будут производиться вычисления (это обеспечивается набором команд в меню основного модуля);**

**-производится настройка параметров и выбор переменных для проведения факторного анализа (интерфейс настройки параметров реализован в модуле “Факторный анализ (входные данные)”);**

**-вычисляется матрица корреляций (эти и последующие вычисления реализованы в математическом модуле “Mathematic.pas”);**

**-рассчитываются общности;**

**-составляется редуцированная матрица корреляций**

**-выделяется набор факторов, достаточно хорошо описывающих весь набор выбранных переменных;**

**-производится этап вращения;**

**-результаты вычислений сохраняются на диске или выдаются на печать (за просмотр, сохранение и распечатку выходных данных, а также создание отчёта по ним отвечают модули “Факторный анализ (выходные данные)” и “Сохранение данных”).**

Сеанс работы с подсистемой можно описать следующим образом: настройка на конкретный формат данных, открытие или создание новых баз данных и их редактирование, настройка входных параметров факторного анализа, получение промежуточных данных, запуск этапа вращения для факторного анализа, создание отчёта по полученным результатам, сохранение их на жёстком диске или распечатка отчёта и графической информации.

Вызов основных действий при работе осуществляется при помощи системы меню и экранных кнопок.

Остановимся более подробно на основном пункте главного меню “Выполнить”. Здесь находятся пункты создания и редактирования форматов данных и настройки системы на текущий формат, а также пункт запуска факторного анализа “Факторный анализ”.

При выборе пунктов “Формат данных/Новый формат” и “Формат

данных/Редактор формата” появляется окно, в котором пользователь может отредактировать выбранный формат данных и сохранить его на диске (кнопка “Сохранить”). Для получения более подробных сведений о работе в редакторе форматов можно воспользоваться помощью, нажав кнопку “Помощь”.

Настройка системы на текущий формат (пункт “Формат данных/Сменить формат”) позволяет настроить систему на любой формат данных.

При выборе в подменю “Выполнить” пункта “Факторный анализ” открывается окно “Факторный анализ (входные данные)”. В нём находятся две закладки. В первой из них пользователь может с помощью клавиш “»”, “«”,“>” и “<” выбрать переменные для вычисления (расчёты будут проводиться по данным, которые находятся в окне, выделенном до запуска факторного анализа), а во второй – параметры факторного анализа. Здесь пользователь может выбрать параметры, которые можно использовать: переменные, факторы, неизвестные параметры (панель “Учитывать”); методы выделения факторов, вычисления общностей и вращения (одноимённые панели); максимальное число выделяемых факторов.

После настройки параметров факторного анализа для начала вычислений следует нажать клавишу “Вычисления” на нижней панели окна. Здесь находятся также кнопки “Отмена” для возврата в редактируемые окна данных и “Помощь”. При нажатии кнопки “Вычисления” выдаётся окно с промежуточными выходными данными на отдельных закладках: матрица корреляций, факторное отображение, остаточная матрица корреляций, факторная структура до вращения, факторная структура после вращения, факторные оценки. Здесь расположена также закладка “Вращение”, в котором пользователь может продолжить вычисления на этапе вращения. Для этого на панели “факторы на осях” необходимо выбрать из списка полученных факторов два (нагрузки переменных на эти факторы будут отображены на графике, и более подробные данные по ним можно будет получить, кликнув левой кнопкой мыши на конкретной точке графика). Кроме этого для этапа вращения необходимо выбрать угол наклона оси. На панели “вращать” находятся два пункта: “автоматически” (программа сама выберет угол) и ”вручную”. Если выбран второй пункт, пользователю предоставляется два способа выбора: набрать

его значение в соответствующем редакторном окне или выбрать его с помощью мыши (клик правой кнопкой на графике – начать выбор, клик левой – зафиксировать угол).

***После того как параметры настроены, можно начинать этап вращения. Для этого необходимо нажать кнопку "Вращение" на нижней панели окна. Этап вращения является итеративным процессом, его можно повторять несколько раз, пока результаты вращения не станут приемлемыми. После каждого шага итерации графическую информацию можно отправить на печать, нажав кнопку "Печать графика".***

Более подробную помощь можно получить по ходу выполнения программы, нажав кнопку "Помощь" на нижней панели окна.

Табличные выходные данные также можно распечатать в виде отчёта, а также сохранить их в отдельных файлах на диске для дальнейшей работы с ними. Для этого необходимо нажать кнопку "Сохранить" или "Отчёт" на нижней панели окна. Пользователю будет предложен перечень выходных данных. Из них необходимо выбрать те, которые нужно сохранить или вставить в отчёт и нажать кнопку подтверждения выбора на этом окне.

Для завершения факторного анализа и возврата в редактор данных нужно закрыть все окна с помощью кнопок "Отмена" на каждом из них.

**Подсистема распознавания образов** содержит программные реализации следующих алгоритмов:

- метод оптимальной оценки значения диагностируемого параметра;
- метод оптимальной классификации;
- метод дискриминантных функций;
- метод потенциальных функций.

Диагностика и прогнозирование производятся методом классификации по признакам (с использованием теории распознавания образов).

Для выбора наиболее информативных признаков используются методы последовательного исключения признаков, последовательного добавления признаков и комбинированный.

В качестве входных данных для всех алгоритмов диагностики выступают данные, которые несут информацию об экземплярах обучающей выборки.

Для работы с программой необходимо средствами Windows запустить управляющую программу. После запуска этой программы на экране появится главное меню, содержащее пункт "Методы".

Пункт "Методы" содержит названия всех методов прогнозирования и диагностики, присутствующих в системе. Содержание этого пункта может меняться в зависимости от того, сколько новых методов будет подключаться к системе в процессе ее эксплуатации. Для проведения диагностики одним из методов необходимо выбрать его имя из данного пункта меню.

В качестве примера работы с программным комплексом рассмотрим работу метода прогнозирования с использованием эвристического алгоритма.

Для этого необходимо из пункта "Методы" главного меню пакета выбрать подпункт "Эвристический алгоритм прогнозирования".

После этого откроется окно (рис 6.16), в котором нужно выбрать имя файла из предложенного

списка, щелкнув левой кнопкой мыши на кнопке "Обзор", или ввести имя файла с полным путем вручную.

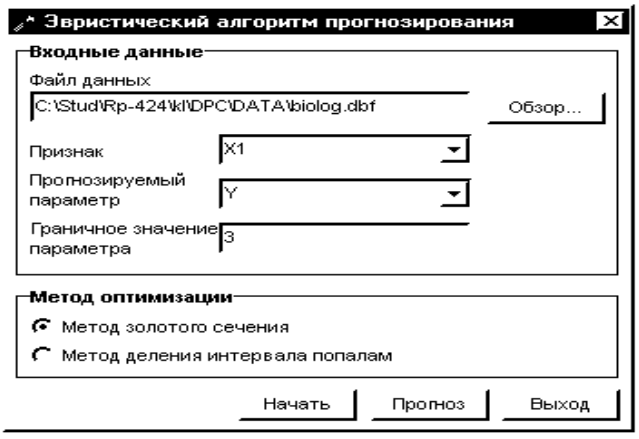



Рис. 6.16 – Основное окно метода прогнозирования с использованием эвристического алгоритма.

В каждом файле данных хранится информация об экземплярах обучающей выборки для одного конкретного типа объектов. Содержимое выбранного файла данных можно просмотреть, "щелкнув" правой кнопкой мыши на форме, появится всплывающее меню, где нужно выбрать пункт "Просмотр файла", либо нажав комбинацию горячих клавиш Ctrl+V.

Так как структура файла данных может быть произвольной, то перед началом выполнения метода прогнозирования необходимо выбрать по какому признаку и параметру будет вестись прогнозирование. Для этого из предложенного списка имен полей базы данных необходимо выбрать нужное. Также необходимо ввести граничное значение прогнозируемого параметра и выбрать один из предложенных методов оптимизации.

Для продолжения работы необходимо нажать кнопку "Начать", для выхода выберите кнопку "Отмена". После получения данных программа произведет все необходимые подсчеты.

На экране в отдельном окне появится характеристика полученной математической модели (рис. 6.17), т.е. вероятность принятия ошибочных решений, риск потребителя и риск изготовителя.

Полученную информацию можно сохранить в файле отчета. Если Вас не устраивают полученные параметры, попробуйте выбрать другой метод прогнозирования и загрузить в него эти

исходные данные.

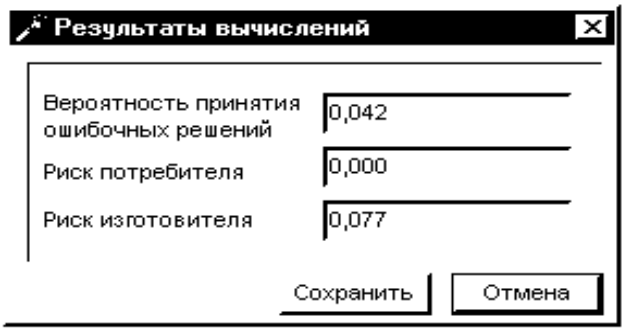



Рис. 6.17 – Окно результата.

Если Вы удовлетворены результатами экзамена полученного оператора прогнозирования, Вы можете приступить к анализу экземпляров не участвующих в обучающей выборке. Для этого необходимо "щелкнуть" на кнопке "Прогноз" и в раскрывшемся окне ввести файл данных, где хранится информация об экземплярах не участвующих в обучающей выборке, выбрать по какому признаку будет вестись прогнозирование и в каком поле базы данных сохранить полученный

класс экземпляра. Далее "щелкнув" на кнопке "Начать", в нижней части окна появится таблица (рис. 6.18).

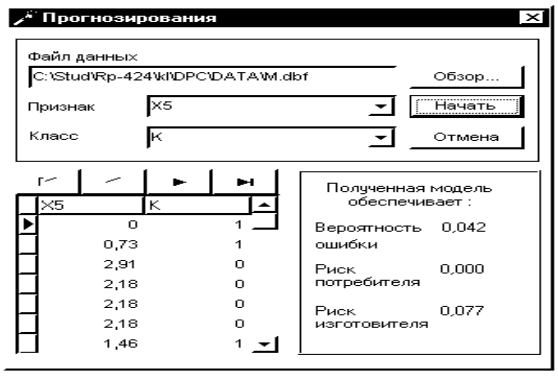



Рис. 6.18 – Окно прогнозирования.

Таблица содержит признаки, по которым велось прогнозирование, и класс, к которому был отнесен экземпляр не участвующий в обучающем эксперименте.

По кнопке "Отмена" можно вернутся в предыдущее окно.

**Подсистема численной аппроксимации и оценивания** реализует модели и методы, позволяющие получать количественную оценку прогнозируемого (оцениваемого) параметра.

В подсистеме численной аппроксимации и оценивания реализованы следующие методы:

- интерполяции Лагранжа по Эйтекену;
- рациональной интерполяции с помощью непрерывных дробей;
- интерполяции по Ньютону;
- аппроксимации кривыми Безье;
- аппроксимации сплайнами;

- аппроксимации полиномами Чебышева;

- интерполяции тригонометрическими полиномами;

- линейного одномерного и многомерного регрессионного анализа;

- нелинейного регрессионного анализа;

\- индивидуального прогнозирования по признакам с оценкой значения прогнозируемого параметра на основе теории статистических оценок.

Подсистема представляет собой Windows – приложение, использующее при работе процедуры, входящие в сотав других модулей диагностического программного комплекса.

**Подсистема нейросетевой диагностики** позволяет решать задачи диагностики на основе нейронных сетей (НС), важнейшими свойствами которых являются универсальность, высокая степень адаптации, а также способность к аппроксимации многомерных функций.

Нейросетевая подсистема состоит из трех Windows-приложений. Такое разделение на программы является обоснованным, поскольку каждая программа может быть использована отдельно, что позволяет существенно снизить требования программного комплекса к ресурсам ЭВМ.

Функциональная схема нейросетевой подсистемы диагностического программного комплекса показана на рис. 6.19.

В состав нейросетевой подсистемы входят программы:

- визуальный конструктор НС – редактор программ на встроенном языке макросов;
- эмулятор-отладчик программ на встроенном языке макросов, позволяющий в режиме интерпретатора пошагово просмотреть результаты выполнения команд;
- программа построения и печати отчетов, графиков, создания, преобразования и визуализации данных.

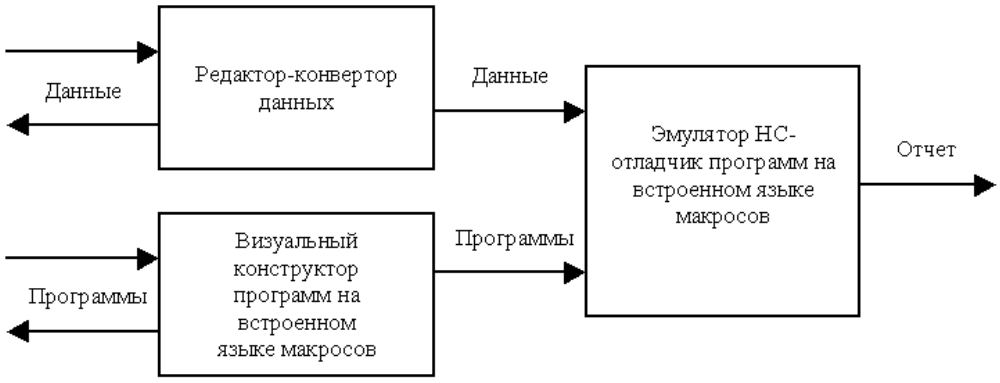



Рис. 6.19 - Функциональная схема нейросетевой подсистемы.

Модели НС, реализованные в нейросетевой подсистеме, а также классы задач, решаемых на их основе, показаны на рис. 6.20.

В нейросетевой подсистеме реализованы следующие алгоритмы:

- алгоритм обучения однослойного перцептрона Уидроу-Хоффа;
- градиентные алгоритмы обучения многослойных нейронных сетей (алгоритм обратного распространения ошибки первого порядка, алгоритм Ньютона, алгоритмы сопряженных градиентов Флетчера-Ривса и Полака-Рибьера, алгоритм Левенберга-Марквардта);
- алгоритмы оценки информативности и отбора информативных признаков для перцептронов;
- алгоритмы обучения НС Хопфилда, машины Больцмана, НС LVQ;
- алгоритмы решения задач комбинаторной оптимизации на основе НС Хопфилда и машины Больцмана;
- алгоритмы формирования карты признаков самоорганизации Кохонена (КПСК), планирования экспериментов на основе КПСК, а также обучения системы КПСК-АЗУ;
- алгоритмы обучения радиально-базисных НС.

Нейросетевая подсистема имеет открытую архитектуру и позволяет пользователю не только выбирать запрограммированные функции и параметры, но и самостоятельно задавать их. Например, пользователь имеет возможность задавать собственные функции активации формальных нейронов.

Программа «эмулятор-отладчик» функционирует под управлением программ на встроенном языке макросов. Встроенный язык макросов предназначен для создания макросов, позволяющих хранить информацию о структуре разработанных пользователем моделей НС, а также для управления процессом работы нейросетевой подсистемы диагностического программного комплекса.

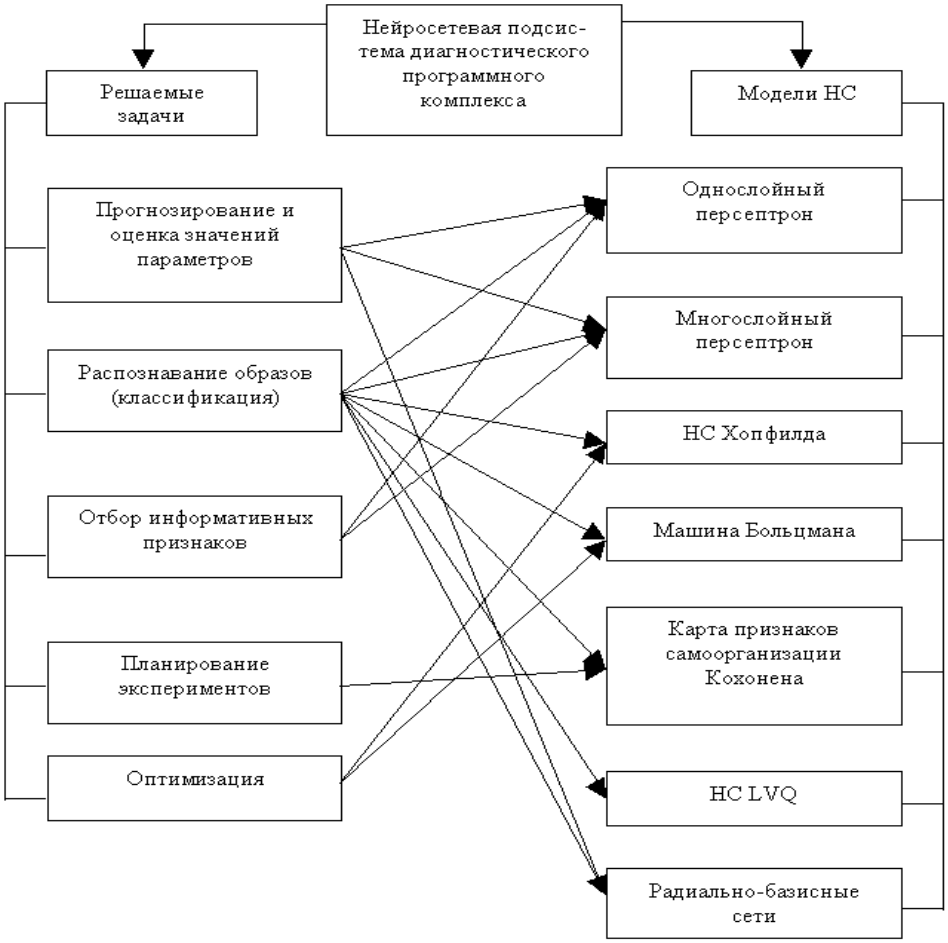



Рис. 6.20 - Модели НС и задачи, решаемые нейросетевой подсистемой на их основе.

Синтаксис этого языка достаточно прост и похож на синтаксис Бейсика, что делает его волне доступным для пользователей. Алфавит языка включает в себя буквы латинского алфавита (заглавные и строчные), цифры 0-9, а также знаки “,”, “.”, “%”, “;”, “+”, “_”, “-”. Язык содержит относительно небольшое количество макросов.

Описание основных макросов приведено в таблице 6.1. Параметры макросов, задаваемые пользователем, набраны курсивом.

Таблица 6.1 – Основные макросы нейросетевой подсистемы

| **Формат макроса** | **Назначение** |
|---|---|
| SET_DATA_SOURCE *путь и имя файла* | Установить источник данных |
| SET_DATA_DESTINATION *путь и имя файла* | Установить приемник данных |
| SET_*внутренняя переменная значение* | Присвоить внутренней переменной значение |
| MATLAB_EXEC *путь и имя MATLAB-файла* | Запустить на выполнение MATLAB-файл |
| WIN_EXEC *путь и имя Windows-приложния* | Запустить на выполнение Windows-приложние |
| ; или % | комментарий (действует в пределах данной строки) |
| STOP или END | Останов - окончание программы |
| *модель*_CREATE | Создать НС |
| *модель* _DATA_RANDOM | Создать обучающую выборку из псевдослучайных чисел |
| *модель* _DATA_LOAD | Загрузить обучающую выборку |
| *модель* _TRAIN | Обучить НС |
| *модель* _SIM | Эмулировать работу НС |
| *модель* _FREE | Уничтожить НС |

Внешние данные (данные на магнитных дисках), используемые нейросетевой подсистемой, представляют собой текстовые файлы в ASCII/OEM кодировке. Для

программы преобразования данных внешние данные могут быть представлены также в виде файлов баз данных, поддерживаемых Borland Database Engine.

В качестве входных данных в нейросетевой подсистеме используются:

- таблицы данных для обучения НС или классификации;
- программы на внутреннем языке программного комплекса (макросы).

Внутренние данные представлены в виде записей. При этом те поля записей, которые содержат наборы данных, представлены в виде динамических массивов, что позволяет с одной стороны эффективно использовать память ЭВМ, а с другой хранить в памяти достаточно большое число данных, ограниченное лишь размером памяти ЭВМ и разрядностью адреса.

Выходными данными нейросетевой подсистемы являются представленные в виде текстовых файлов в ASCII/OEM кодировке:

- результаты обучения или работы НС (отчет);
- текст программы на внутреннем языке программного комплекса.

В нейросетевой подсистеме реализован стандартный интерфейс для приложений Windows'95. Нейросетевая подсистема осуществляет взаимодействие с пользователем посредством многоуровневого меню и кнопок на панелях экранных форм. Экранные формы главных модулей

программ, входящих в нейросетевую подсистему показаны на рис. 6.21 и 6.22.

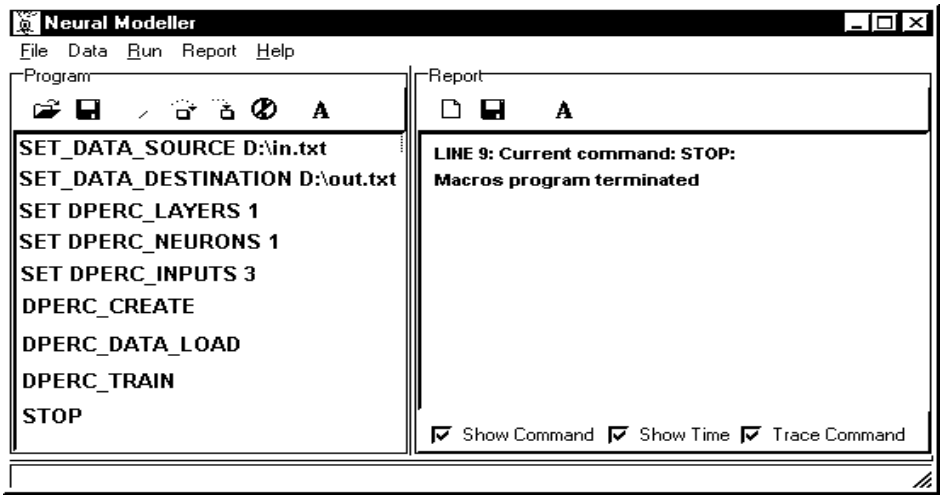



Рис. 6.21 - Главная экранная форма редактора-конструктора программ.

Нейросетевая подсистема обеспечивает вывод на экран (по желанию пользователя) промежуточной информации в виде отчета, что способствует формированию у пользователя доверия к программе и позволяет ему глубже понять основные механизмы программы. Таким образом пользователь всегда может проследить, каким образом было получено то или иное решение, проверить правильность расчетов.

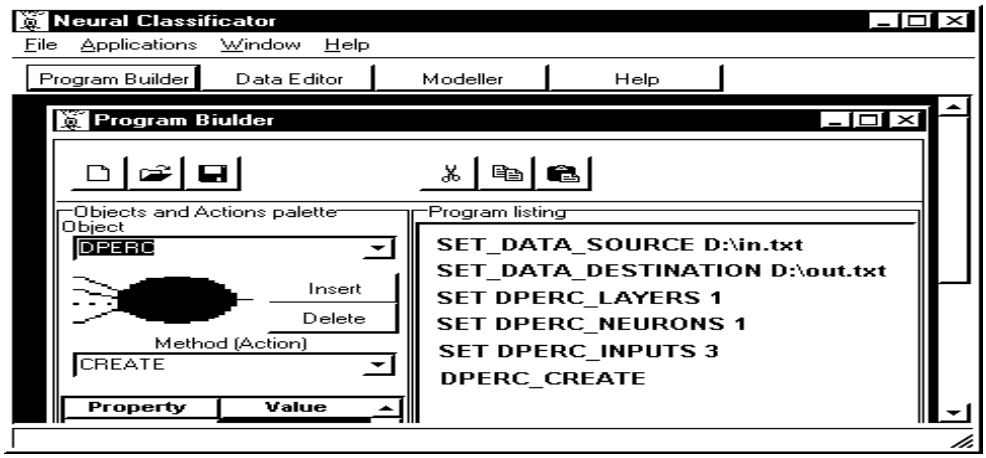



Рис. 6.22 - Главная экранная форма интерпретатора-отладчика

Программы отслеживают корректность вводимых пользователем данных. В случае ввода некорректных данных на экран выдается сообщение об ошибке и ее характере.

Результаты проделанной работы отображаются подробно. Пользователь имеет возможность регулировать объем (степень детализации) информации, выдаваемой в виде отчета. Всегда можно просмотреть входные данные, обработка которых привела к получению именно таких результатов.

Благодаря интерфейсному модулю диагностического программного комплекса нейросетевая подсистема может совместно работать с программным комплексом MATLAB 5.2 фирмы MathWorks, при этом достигается совместимость по данным, а также возможно осуществлять MATLAB-вставки внутрь программы на языке

макросов нейросетевой подсистемы. Для возможности исполнения MATLAB-вставок в программах на встроенном языке макросов необходимо наличие на ЭВМ установленной программы MATLAB 5.2.

**Подсистема оптимизации** представляет собой программный модуль, используемый подсистемами, входящими в состав диагностического программного комплекса. Модуль оптимизации реализован на языке Inprise (Borland) Delphi 5.0 и содержит процедуры, реализующие следующие алгоритмы:

- алгоритмы одномерной оптимизации:
- деления интервала пополам;
- Ньютона-Рафсона;
- Фибоначчи;
- золотого сечения;
- квадратичной аппроксимации;
- кубической аппроксимации с использованием производных;
- средней точки;
- скользящих средних.
- алгоритмы многомерной оптимизации:
    - градиентный метод Коши;
    - Ньютона;
    - сопряженных градиентов Флетчера-Ривса;
    - сопряженных градиентов Полака-Рибьера;
    - партан-метод;
    - Левенберга-Марквардта;
    - Давидона-Флетчера-Пауэлла;
    - обобщенный градиентный;
    - Хука-Дживса;
    - Нелдера-Мида;
    - штрафных функций;
    - линейного программирования.

Процедуры, входящие в состав подсистемы оптимизации могут быть вызваны пользователем посредством интерфейсного модуля.

В процессе работы процедур оптимизации генерируется отчет, который хранится во внутренней переменной модуля оптимизации Report. Этот отчет доступен как другим подсистемам, использующим модуль оптимизации, так и пользователю при работе через интерфейсный модуль.

## 6.2 Программно-аппаратный комплекс ПОС «Вояж» НПП «Мера»

Программно-аппаратный переносной многоканальный измерительный комплекс «Портативный анализатор сигналов – пакет обработки сигналов (ПОС)», разработанный научно-производственным предприятием "Мера" совместно с ЗАО "L-card", построен на базе переносного или портативного компьютера (рис. 6.23). Это позволяет использовать весь спектр программ, работающих на компьютере типа IBM PC.

«Портативный анализатор сигналов - ПОС» представляет собой многофункциональный прибор, который может использоваться как:

- многоканальное устройство ввода в компьютер внешних сигналов, позволяющее одновременно вводить сигналы самых разнообразных типов с установленного набора датчиков;
- прибор для мобильных систем мониторинга и контроля;
- универсальный прибор с большими возможностями для исследовательских лабораторий, позволяющий решать широкий спектр задач;
- прибор для вибрационного контроля в эксплуатации;
- прибор с возможностью реализации сложного и адаптивного алгоритма для диагностики конкретного дефекта.

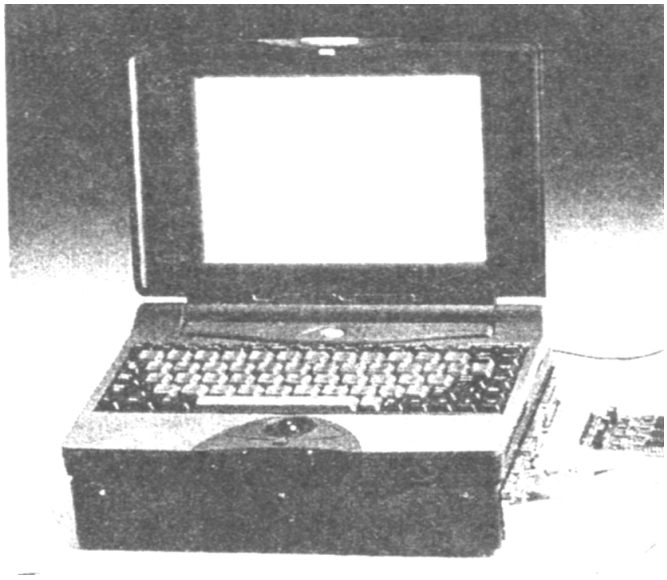


Рис. 6.23 - Внешний вид программно-аппаратного комплекса ПОС

В одном корпусе с анализатором можно получить целый комплект полноценных измерительных приборов и средств контроля. Это возможно благодаря специализированному программному обеспечению, реализующему функции этих устройств. Эти воплощения физических устройств в программном виде называются виртуальными приборами. В настоящее время реализованы: сборщик, динамический анализатор, магнитограф, интеллектуальный осциллограф, виброметр, прибор для расчета собственных частот объекта, тензоанализатор, прибор для расчета колебаний лопаток, балансировщик, виброанализатор спектра.

В руках опытного исследователя или инженера анализатор становится удобным и мощным средством для исследований и анализа. ПОС предоставляет для этого все возможности:

- управление в диалоговом режиме аппаратными средствами, их программирование и тестирование;
- представление измерительной информации в графическом и табличном виде (двухмерная и трехмерная графика, декартовые, полярные и логарифмические координаты);
- алгоритмы обработки: цифровая фильтрация, (взаимно-) спектральный.

(взаимно-) корреляционный, статистический, третьоктавный анализ, расчет огибающей, интегрирование, дифференцирование и другие;

-виброанализ: следящий анализ (расчет АФЧХ) на режимах выбега и разбега. расчет вибропаспорта и многое другое;

-документирование изображения с экрана на принтер/диск;

-командный режим, позволяющий пользователю путем написания в нем программ быстро и удобно реализовать свои протоколы обработки;

-архивацию измерительной информации и результатов обработки.

-Командный режим ПОС - инструмент для реализации уникальных методик и алгоритмов. Макроязык ПОС, включающий более 230 процедур (список может быть расширен самим пользователем) позволяет создавать свои виртуальные приборы.

-Измерительная часть (модуль АЦП-ЦОС и до 6 модулей УСО серии LM) размещается в специализированном крейте с жестким креплением к Notebook и обладает следующими техническими характеристиками в штатной поставке:

-частота ввода - до 250 кГц;

-питание: от сети или автономное от встроенных аккумуляторов до 1.5 часов, включая питание датчиков;

-число каналов - 16 аналоговых (до 112 во внештатной поставке), входной диапазон ±5В, 8 виброканалов (усилитель+ФНЧ+УВХ);

-вес - 4.5 кг;

-Notebook с цветным TFT дисплеем, процессором Pentium – 120, ОЗУ объемом - 8 Мб и жестким диском объемом 1 Гб.

Главной составляющей портативного анализатора является ПОС, представляющий собой открытую программную платформу, которая превращает ПЭВМ в удобный инструмент для полного цикла работ с измерительной информацией. Пакет имеет более чем 5-ти летний опыт эксплуатации как в качестве самостоятельного программного продукта, так и в составе программно-аппаратных комплексов в различных областях.

Программно-аппаратный комплекс ПОС в составе ПЭВМ, ПОС и плат АЦП-ЦАП фирмы “L-card” имеет утвержденную Госстандартом России программу метрологической аттестации.

Набор основных алгоритмов ПОС включает в себя алгоритмы:

- цифровой фильтрации (Наряду со встроенными фильтрами Баттерворта, Чебышева и Кауэра пользователь может сам легко синтезировать фильтры с требуемыми частотными характеристиками и тут же увидеть их амплитудные и фазовые характеристики);
- спектрального анализа (Расчет амплитудного спектра, спектра мощности, спектральной плотности мощности, спектральной плотности энергии: расчет комплексного спектра и представление его в виде модуля и фазы либо в виде вещественной и мнимой части. При этом пользователю доступны: применение весовых окон Ханнинга, Блэкмана-Хэрриса, треугольного, Flat Top, линейное усреднение с возможностью наложения 25%, 50% и 75%, выбор размера порции БПФ от 64 до 20480 точек);
- взаимно-спектральный анализ (Расчет взаимного спектра мощности, его модуля и фазы либо вещественной и мнимой частей. Реализованы алгоритмы вычисления передаточной функции, функции когерентности (некогерентности), когерентной (некогерентной) выходной мощности, отношения сигнал/шум);
- кепстральный анализ;
- (взаимно-) корреляционный анализ;
- статистический анализ с расчетом первых четырех моментов и плотности распределения вероятности;
- третьоктавный анализ;
- преобразование Гильберта и расчет огибающей;
- интегрирование;
- дифференцирование;
- арифметические операции над наборами данных, их нормирование и центрирование, линейная и квадратичная тарировка измерительной

информации.

Однако, комплекс ПОС имеет и недостатки, среди которых особо следует выделить:

- отсутствие средств преобразования данных из / в файлов различных форматов, в том числефайлов баз данных и электронных таблиц;
- неполнота и ограниченность средств аппроксимации численных зависимостей
- отсутствие современных интеллектуальных средств моделирования, таких как нейронные сети (НС).

### 6.3 Интегрированная система диагностики

Портативный анализатор сигналов ПОС сосредотачивает в себе мощность современного компьютера, надежность высоко-технологичных средств и многофункциональность обширного набора методик исследований и анализа сигналов, что дает возможность использовать его вместо дорогостоящего, зачастую громоздкого, отечественного и зарубежного измерительного оборудования. В то же время, как было отмечено ранее, ПОС не позволяет осуществлять построение моделей многомерных зависимостей с заданной точностью, что необходимо для автоматизации процесса диагностики лопаток.

В свою очередь нейросетевая подсистема диагностического программного комплекса, обладающая разнообразными средствами моделирования многомерных зависимостей и преобразования данных, не может непосредственно управлять средствами измерительной аппаратуры.

Очевидно, что интеграция обоих программных комплексов в единую систему позволит не только избавиться от недостатков и ограничений, имеющихся у комплексов по отдельности, но и предоставит возможность автоматизации рабочих мест контролера качества в производстве и

обслуживающего техника в эксплуатации двигателей. Предлагаемая схема интегрированной системы диагностики изображена на рис. 6.24.

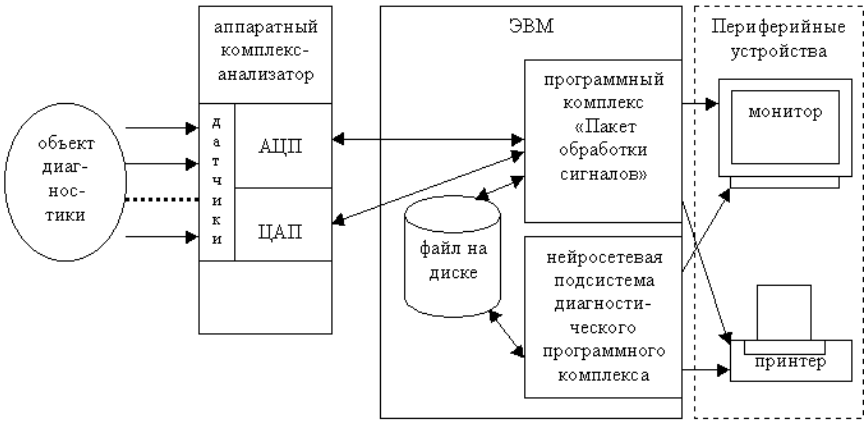



Рис. 6.24 - Схема интегрированной системы диагностики.

Как видно из рис. 6.24, в начале процесса диагностики определенные параметры диагностируемого объекта измеряются с помощью датчиков, сигналы от которых поступают в аппаратный анализатор. Здесь сигналы преобразуются из аналоговой формы в цифровую с помощью ЦАП и передаются в ЭВМ для дальнейшей обработки программному комплексу ПОС, с помощью которого данные проходят предварительную обработку (фильтрация, нормирование, преобразование сигнала в другую форму и т.д.) и затем сохраняются на диске в текстовом формате. Далее нейросетевая подсистема диагностического программного комплекса на основе предварительно обученной НС осуществляет классификацию объекта.

После проведения расчетов нейросетевая подсистема диагностического программного комплекса генерирует подробный отчет, содержащий результаты классификации и промежуточные данные. Этот отчет может быть отображен на экране монитора или выведен на печать с помщью принтера.

## 6.4 Пакет Matlab

MATLAB (MATrix LABoratory – МАТричная ЛАБоратория) представляет собой математический пакет, предназначенный для решения задач вычислительной математики, математической физики и построения численных моделей сложных объектов и процессов.

Пакет MATLAB состоит из интерпретатора - модельной среды, имеющего терминальный интерфейс, ядра (набора простейших стандартных операций, функций и процедур для вычислений), а также библиотек функций (Toolbox).

Достоинствами пакета являются: богатые графические возможности, обширный набор математических функций, простота встроенного языка MATLAB, возможность автоматического преобразования текстов программ на языке MATLAB в тексты программ на языке Си, а также то, что тексты библиотечных функций поставляются в исходном виде.

К недостаткам пакета следует отнести отсутствие дружелюбного пользовательского интерфейса и низкую скорость работы.

Для моделирования нейронных сетей с помощью пакета MATLAB необходимо установить и использовать библиотеку **Neural Network Toolbox**. Рассмотрим примеры построения моделей и обучения НС на языке пакета MATLAB.

Моделирование и обучение НС Хопфилда

| | |
|---|---|
| x = [-1 1; -1 –1; 1 1]; | Задаем значения признаков экземпляров обучающей выборки: 2<br>экземпляра (столбцы), 3 признака (строки). |
| net = newhop(x); | Создаем и обучаем НС Хопфилда. |

Моделирование и обучение персептрона

| | |
|---|---|
| x = [0.1 0.5 0.2 0.4 0.3 0.9;<br>0.9 0.5 0.8 0.6 0.7 0.1;<br>0.3 0.0 0.6 0.1 0.2 0.9]; | Задаем значения признаков экземпляров обучающей выборки: 6 экземпляров (столбцы), 3 признака (строки). |
| y = [ 1 0 1 0 1 0]; | Задаем номера классов для 6 экземпляров обучающей выборки. |
| net=newff(repmat([0 1], 3,1),[2,1],{'logsig', 'logsig'}, 'trainlm'); | Создаем нейронную сеть net и определяем ее топологию: диапазон изменения значений признаков [0 1], количество признаков – 3, количество выходных переменных – 1, на первом слое – 2 нейрона, на втором слое 1 – нейрон, нейроны 1 и 2 слоев имеют сигмоидные функции активации (logsig), в качестве метода обучения сети используется метод Левенберга-Марквардта (trainlm). |
| net.trainparam. show=25; | Задаем период отображения информации о процессе обучения на экране в циклах (эпохах) обучения. |
| net.trainparam.lr= 0.01; | Задаем шаг обучения. |
| net.trainparam. epochs=500; | Задаем максимально допустимое количество циклов обучения (эпох). |
| net.trainparam. goal=0.01; | Задаем максимально допустимое значение критерия обучения (ошибки обучения). |
| ct=cputime; | Определяем и запоминаем текущее значение счетчика времени в переменной ct. |
| net=train(net, x,y); | Обучаем нейронную сеть net на основе обучающей выборки, представленной набором значений признаков экземпляров x и набором значений соответствующих им номеров классов y. |
| ct=cputime-ct | Определяем текущее значение счетчика времени, вычитаем из него значение переменной ct – определяем время обучения НС, которое заносим в переменную ct и выдаем на экран (признак печати на экране – отсутствие символа “;” в конце оператора. |
| a=round(sim(net, x)); | Вычисляем по обученной сети net номера классов для экземпляров, характеризующихся набором значений признаков x. |

## Моделирование и обучение НС LVQ

| | |
|---|---|
| x=[1 –2 2 0 4 –5 3]; | Задаем значения признаков экземпляров обучающей выборки: 7 экземпляров (столбцы), 1 признак (строки). |
| y=[1 2 1 2 1 2 1]; | Задаем номера классов для 7 экземпляров обучающей выборки. |
| yc=ind2vec(y); | Преобразуем номера классов во внутренний формат. |
| net=newlvq(minmax(x),4, [0.6 0.4], 'learnlv1'); | Создаем нейронную сеть net и определяем ее топологию: диапазон изменения значений признаков определяется функцией minmax, количество скрытых нейронов (кластеров) – 4, априорная вероятность отнесения экземпляров к первому классу – 0.6, ко второму – 0.4, в качестве метода обучения сети используем метод LVQ1. |
| net.trainParam.epochs= 1000; | Задаем максимально допустимое количество циклов обучения (эпох). |
| net.trainParam.show= 100; | Задаем период отображения информации о процессе обучения на экране в циклах (эпохах) обучения. |
| net.trainParam.lr=0.05; | Задаем шаг обучения. |
| net=train(net,x,yc); | Обучаем нейронную сеть net на основе обучающей выборки, представленной набором значений признаков экземпляров x и набором значений соответствующих им номеров классов y. |
| a=sim(net,x); | Вычисляем по обученной сети net номера классов для экземпляров, характеризующихся набором значений признаков x. |
| ac=vec2ind(a) | Преобразуем номера кластеров в удобный для восприятия формат и выдаем на экран. |

## Моделирование и обучение радиально-базисных НС

| | |
|---|---|
| x = [1 2 3]; | Задаем значения признаков экземпляров обучающей выборки: 3 экземпляра (столбцы), 1 признак (строки). |
| y = [2.0 4.1 5.9]; | Задаем значения прогнозируемого параметра для 3 экземпляров обучающей выборки. |
| net = newrb(x,y); | Создаем и обучаем радиально-базисную НС |
| a = sim(net,x) | Вычисляем по обученной сети net значения прогнозируемого параметра для экземпляров, характеризующихся набором значений признаков x. |

Моделирование и обучение НС SOM

| | |
|---|---|
| x = rand(1,400); | Задаем набор значений для одного признака 400 экземпляров с помощью датчика случайных чисел |
| net=newsom( minmax(x), [2 5]); | Создаем нейронную сеть net и определяем ее топологию: диапазон изменения значений признаков определяется функцией minmax, в первом слое массив нейронов – 2x5. |
| net=train(net, x); | Обучаем нейронную сеть net (формируем кластеры) на основе обучающей выборки, представленной набором значений признаков экземпляров x. |
| y=sim(net,x); | Вычисляем по обученной сети net номера кластеров, сопоставленных нейронам выходного слоя НС для экземпляров, характеризующихся набором значений признаков x. |
| yc=vec2ind(y) | Преобразуем номера кластеров в удобный для восприятия формат и выдаем на экран. |

Кроме рассмотренных средств моделирования НС пакет MATLAB, начиная с версии 6.0, содержит визуальный интерфейсный модуль **nntool**, который входит в библиотеку Neural Network Toolbox.

Использование nntool позволяет более удобными средствами, чем написание программы вручную, строить нейросетевые модели технических объектов и процессов. Рассмотрим некоторые основные возможности и приемы работы со средством nntool.

После запуска Matlab.exe в командном окне для начала работи с nntool нужно ввести: nntool. После этого загрузится средство nntool (рис. 6.25).

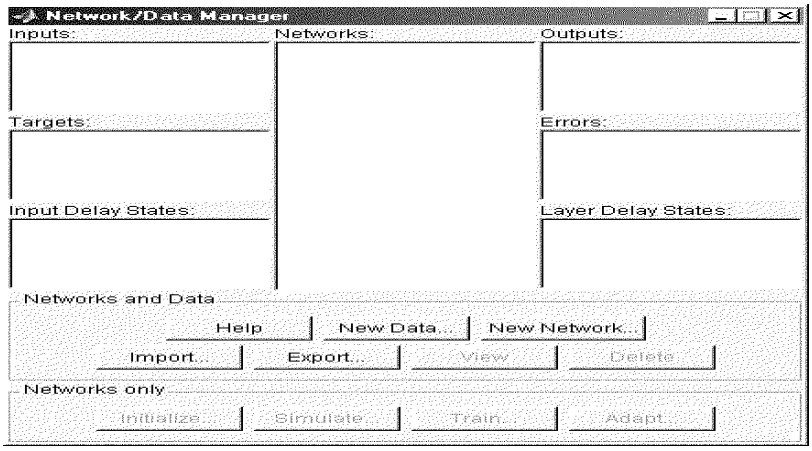



Рис. 6.25 - Главная диалоговая форма средства nntool

На панели Network and Data ("Нейросети и данные") пользователь должен нажать кнопки для задания выходных данных для построения нейросетевой модели.

Кнопка New Data ("Новые данные") вызывает редактор для создания новых данных (рис. 6.26).

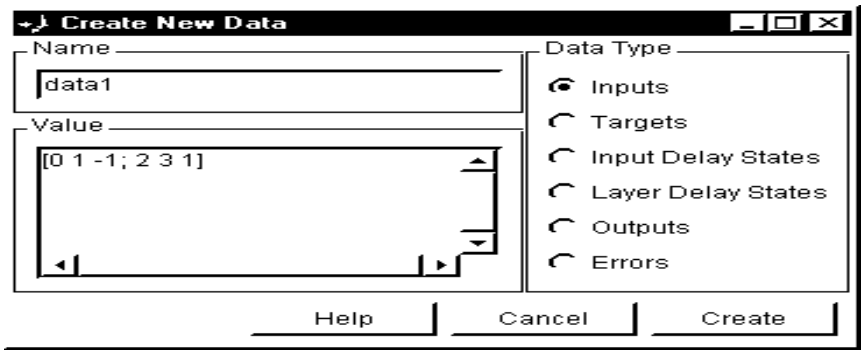



Рис. 6.26 - Редактор данных средства nntool

Поле Name ("Имя") задает имя новой переменной среды MATLAB, в которую сохраняется массив новых данных, которые вводятся в поле Value ("Значение"). Панель Data Type ("Тип данных") определяет назначение введенных данных: Inputs - входы сети, Targets - целевые значения выходов сети, Input Delay States и Layer Delay States - описание задержек на входах и в слоях сети, Outputs - расчетные значения на выходах сети, Errors - ошибки сети.

Если данные уже существуют в виде внешних файлов илии содержатся в среде MATLAB в виде переменных, они могут быть импортированы с пмощью кнопки Import ("Импорт"). При этом появляется диалоговая форма (рис. 6.27).

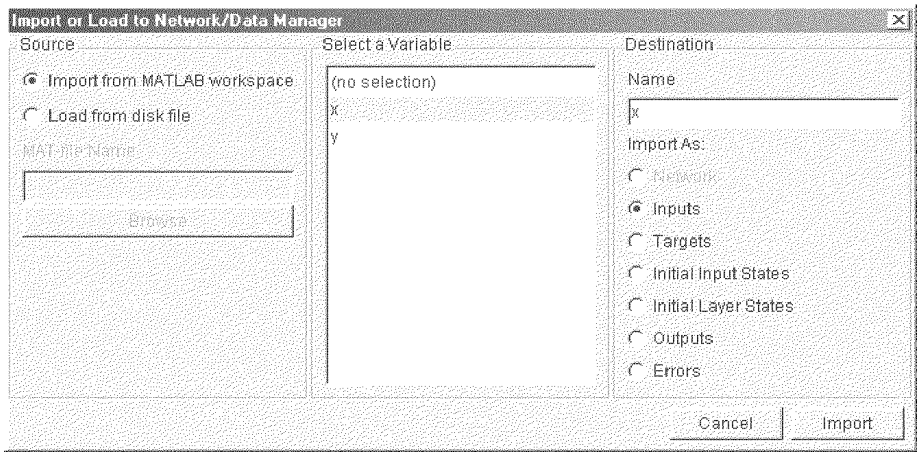



Рис. 6.28 - Диалоговая форма импорта данных.

Поле Source ("Источник") позволяет выбрать источник ввода данных: Import from Matlab workspace (импорт данных из среды MATLAB) или Load from disk file (загрузка данных из файла на диске). Кнопка Browse позволяет выбратт необходимый файл.

Поле Select a Variable ("Выбор переменной") позволяет уазать средству nntool, какую переменную нужно использовать для импорта данных.

Панель Destination ("Приемник") позволяет задать переменную для приема импортируемых данных. Ее имя указывается в поле Name ("Имя"), а назначение (Import as) выбирается из приведенного меню.

Кнопка Export ("Экспорт") главной диалоговой формы позволяет сохранить данные из среды nntool в файле на диске или передать их в среду MATLAB.

Кнопка "New Network" ("Новая сеть") вызывает диалоговую форму для конструирования нейросети и определения ее параметров (рис. 6.29).

Поле Network Name ("Имя сети") определяет имя переменной, где хранится сеть. Список выбора Network Type ("Тип сети") позволяет выбрать тип архитектуры сети (например, Feed-forward backprop - многослойная НС прямого распространения), поля Training Function, Adaptation Learning Function, Performance Function и Number of Layers определяют, соответственно, тип алгоритма обучения

сети, тип алгоритма адаптации весов сети, целевую функцию и количество слоев сети.

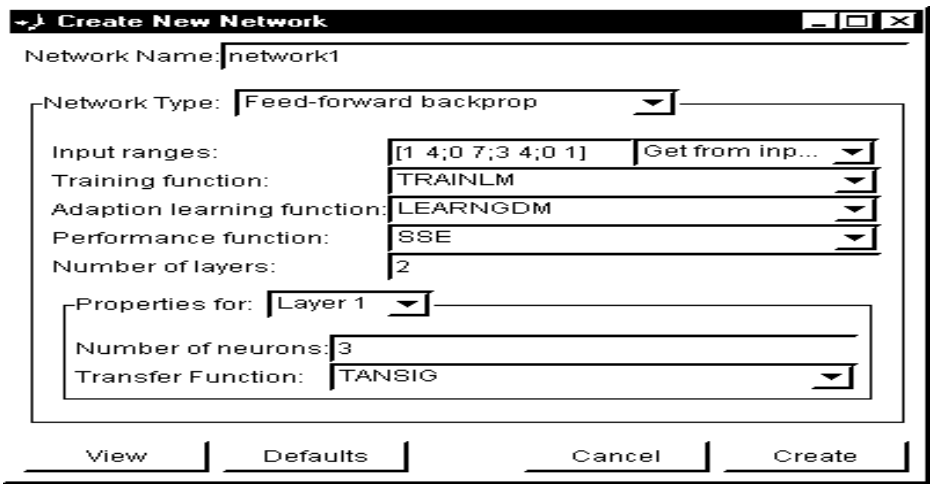



Рис. 6.29 - Форма конструирования нейросети.

Панель Properties for Layer K позволяет задать свойства для нейронов K-го слоя сети. В поле Number of Neurons указывают количество нейронов для текущего слоя, а в поле Transfer Function - тип функции активации нейронов текущего слоя сети.

Кнопка "View" позволяет получить графическое изображение схемы построенной нейросети (рис. 6.30).

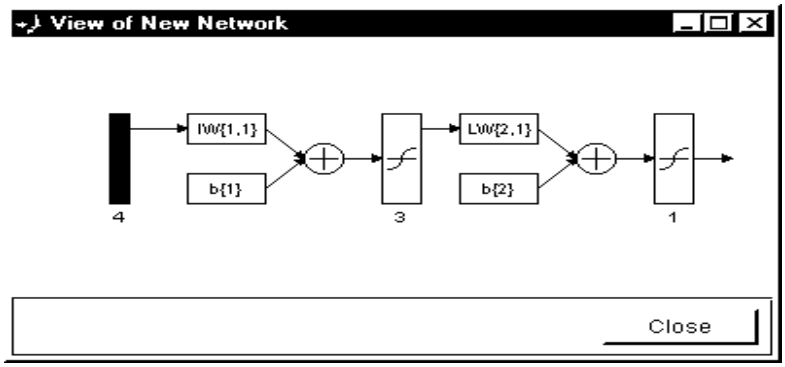



Рис. 6.30 - Пример изображения формы нейросети, построенной с помощью средства nntool.

Кнопка "Delete" главной формы позволяет удалить ненужный элемент данных (переменную или сеть).

Кнопка "Help" позволяет вызвать справочную службу MATLAB с описанием необходимых компонентов и пояснением их использования.

После построения НС в нижней части главной диалоговой формы nntool становится доступной панель Networks only, которая предназначена для работи с построенной сетью (рис. 6.31).

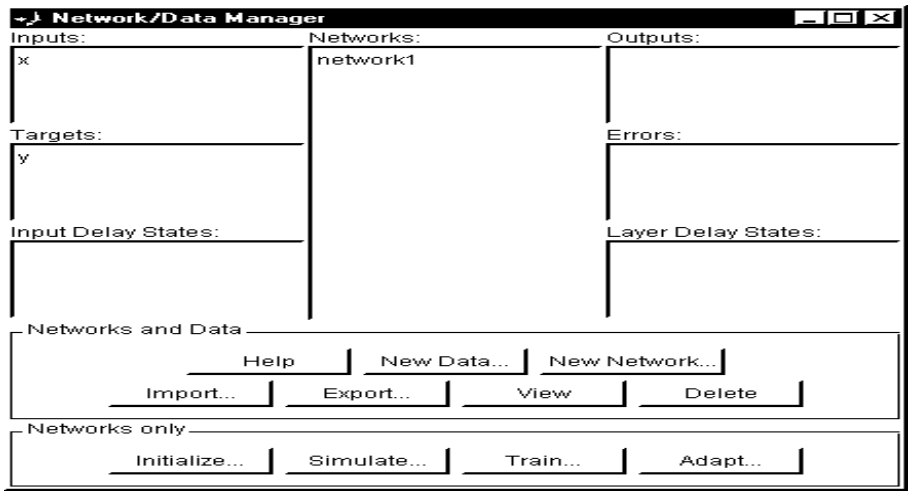



Рис. 6.31 - Главная диалоговая форма nntool после построения сети

При нажатии любой кнопки этой панели вызывается диалоговая форма Network ("Сеть"), которая содержит набор закладок-панелей для работи с сетью (рис. 6.32-6.35).

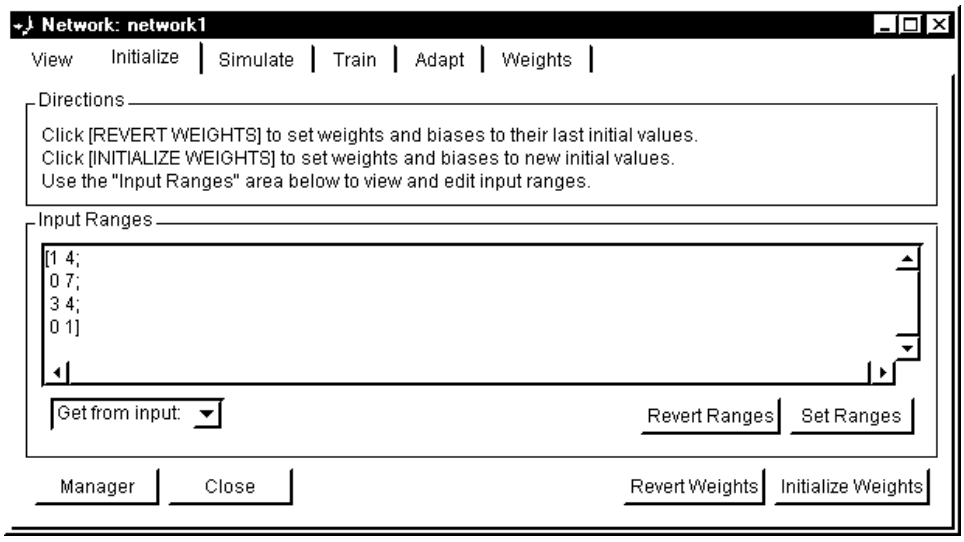



Рис. 6.32 - Форма Network: закладка Initialize

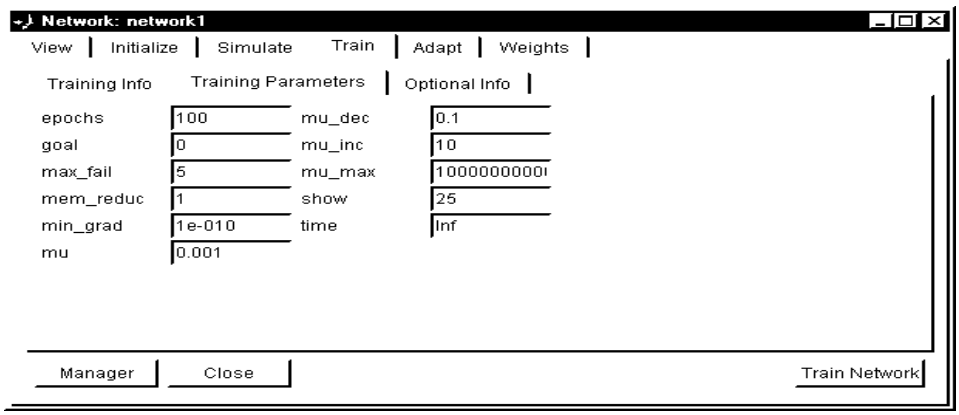



Рис. 6.33 - Форма Network: закладка Train.

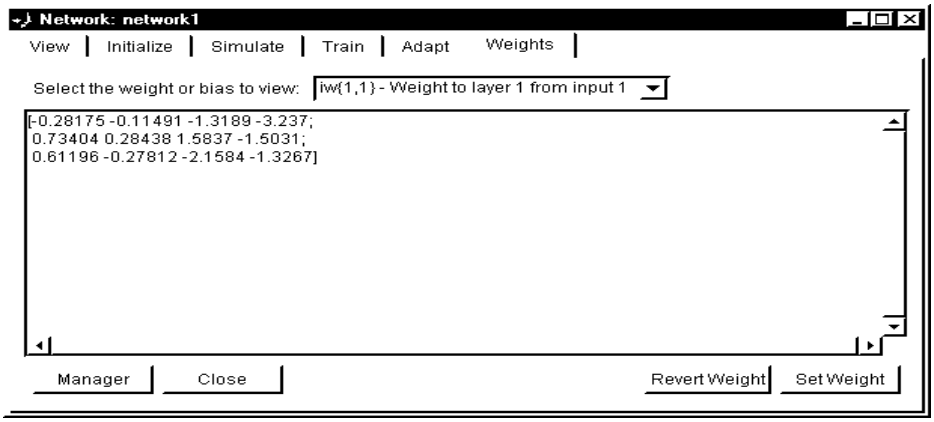



Рис. 6.34 - Форма Network: закладка Weights.

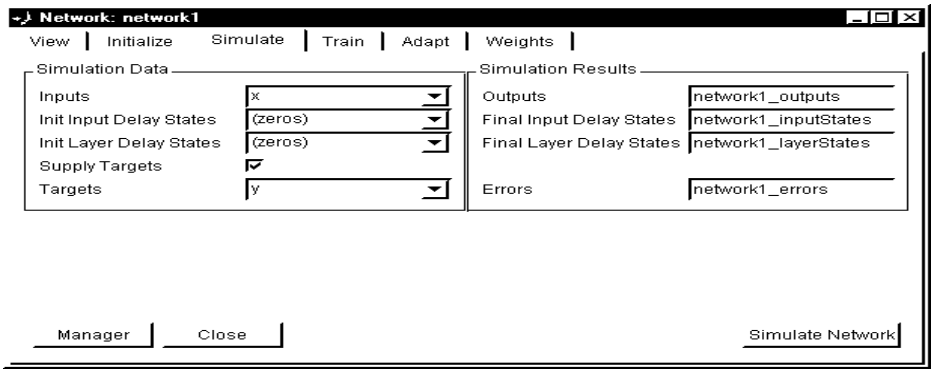



Рис. 6.35 - Форма Network: закладка Simulate.

Закладка Initialize ("Инициализация") позволяет задать границы, в которых изменяются входные данные и рассчитать на их основе начальные значения весов сети.

Инициализирования сеть может быть обучена с помощью закладки Train ("Тренировка, обучения"). Среди параметров обучения, доступных на этой панели обязательно следует задать: goal - максимально допустимое значение целевой функции, epochs - максимальное допустимое количество циклов обучения сети, show - шаг вывода на экран информации об обучении сети, задается в циклах обучения. В процессе обучения среда MATLAB строит график изменения значения целевой функции по эпохам - циклам обучения (рис. 6.36).

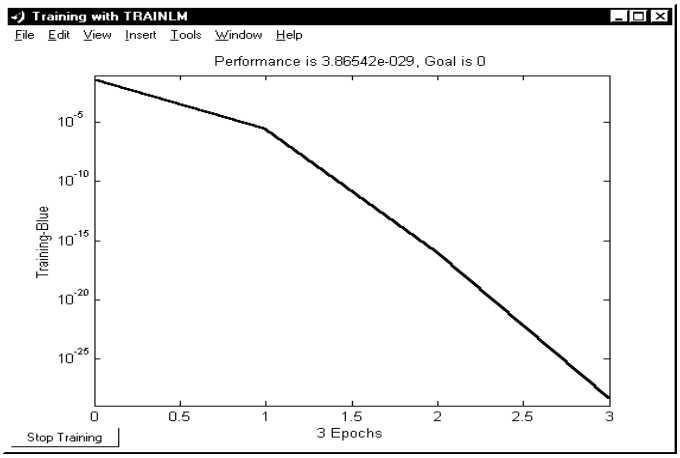



Рис. 6.36 - График изменения значения целевой функции в процессе обучения.

Веса обученной сети можно просмотреть используя закладку Weights ("Веса").

После того, как сеть обучилась, ее можно использовать для распознавания с помощью закладки Simulate ("Моделирование").

# ГЛАВА 7. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ ПО ДИАГНОСТИКЕ И ПРОГНОЗИРОВАНИЮ НАДЕЖНОСТИ АВИАДВИГАТЕЛЕЙ

## 7.1 Диагностика лопаток газотурбинных авиадвигателей

### 7.1.1 Постановка задачи

Высокие требования к качеству и надежности изделий предъявляются в моторостроении. Отказ авиационного двигателя в полетных условиях, связанный с их разрушением, может привести к катастрофическим последствиям. Поэтому важно своевременно выявить и устранить дефекты и причины их возникновения в процессе эксплуатации двигателей летательных аппаратов.

Наиболее нагруженными деталями газотурбинных двигателей являются лопатки. Наилучшая работоспособность нагруженных деталей газотурбинных двигателей обеспечивается при сочетании высоких пределов текучести, длительной прочности и высокой деформационной способности (пластичности) материала. Однако в большинстве случаев такое сочетание прочностных и пластических свойств не может быть достигнуто.

В связи с тем, что лопатки работают при значительных вибрационных нагрузках, основным требованием к их материалам является высокое сопротивление усталости.

Усталость представляет собой крайне опасный вид разрушения деталей машин из-за фактора внезапности и полного выхода их из строя.

Под усталостью понимается процесс, происходящий в детали при циклическом нагружении и приводящий в конечном счете к разрушению при напряжениях, безопасных при статической нагрузке. Количественно усталость описывается зависимостью между накопленным повреждением и числом циклов или длительностью нагружения. Сложный процесс усталости может быть разделен на

ряд более простых процессов, некоторые из которых непосредственно ответственны за усталость, а другие являются не столько причиной, сколько следствием усталости и могут быть использованы для ее изучения. Основными среди усталостных явлений являются локальная пластическая деформация, циклическое упрочнение и разупрочнение, зарождение, рост и слияние микротрещин.

Проблема усталости может быть решена только в том случае, если будут разработаны достаточно надежные методы, позволяющие прогнозировать зарождение усталостной трещины, описать процесс ее развития и предсказать момент окончательного разрушения с учетом влияния основных конструкционных, технологических и эксплуатационных факторов. В то же время в большинстве исследований по многоцикловой усталости за критерий разрушения принимается полное разрушение.

Трещины в лопатках турбины, возникающие в процессе эксплуатации, являются одним из наиболее характерных и распространенных дефектов. Этот вид дефекта, как правило, появляется и развивается в течение определенного временного промежутка в процессе выработки двигателем его ресурса. Это дает реальную возможность осуществлять диагностические мероприятия для своевременного выявления дефектных лопаток. Однако, обычно эта процедура, осуществляемая традиционными методами и средствами, является весьма трудоемкой для обслуживающего

персонала. Поэтому автоматизация процесса выявления дефектных лопаток крайне важна для упрощения и ускорения процесса обслуживания авиадвигателей, повышения надежности их диагностирования.

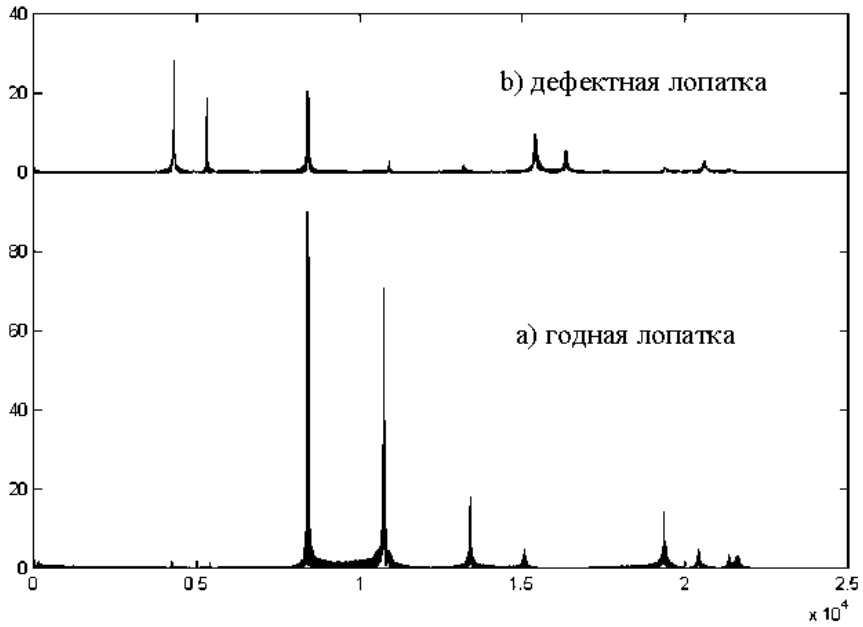



Рис. 7.1 - Спектры свободных затухающих колебаний годной (a) и дефектной (b) лопаток авиадвигателя после ударного возбуждения.

Одним из возможных методов диагностирования является метод измерения параметров свободных затухающих колебаний лопаток в процессе их широкополосного импульсного возбуждения путем простукивания. Для определения дефектов собираются данные - спектры свободных затухающих колебаний лопаток после ударного возбуждения (рис.7.1) или разности полупериодов затухающих колебаний лопаток после ударного возбуждения (рис. 7.2). На основе этих данных необходимо уметь осуществлять классификацию лопаток на группы кондиционных и дефектных.

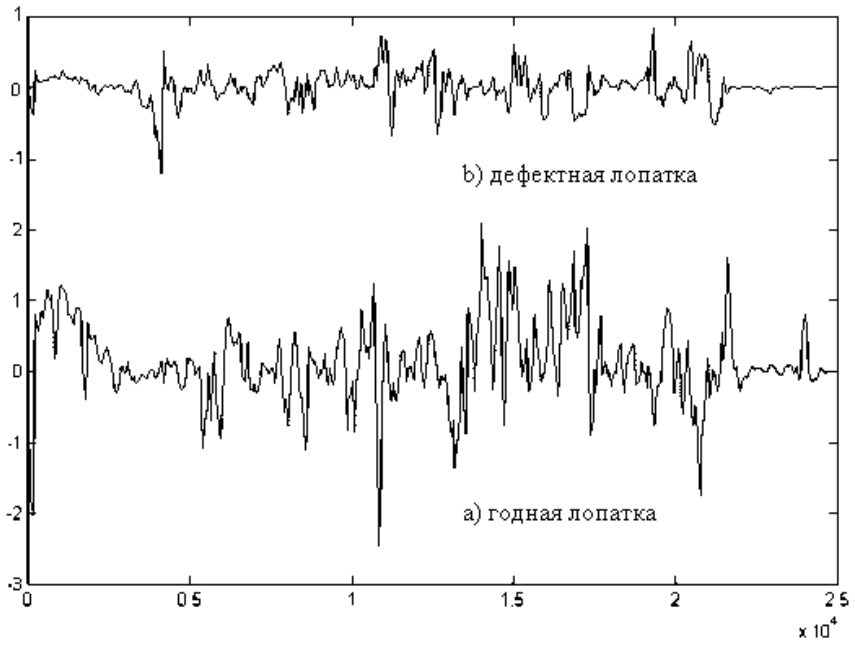



Рис. 7.2 - Разности полупериодов затухающих колебаний годной (a) и дефектной (b) лопаток после ударного возбуждения.

Для исследования взаимосвязи между параметрами и классом лопаток были проведены эксперименты, в результате которых были получены наборы значений параметров усредненных спектров мощности свободных колебаний для кондиционных лопаток и лопаток с трещинами.

Эксперименты проводились для рабочих лопаток первой ступени турбины высокого давления двигателя. Спектры свободных затухающих колебаний содержали 10240 спектральных линий в частотном диапазоне до 25000 Гц, с разрешением по частоте 2.44 Гц. Значения спектров характеризуют усредненный частотный состав свободных затухающих колебаний в виде спектральной плотности мощности амплитуд виброускорения.

### 7.1.2 Сокращение размерности данных

Поскольку набор признаков в данной задаче имеет чрезвычайно большой размер (порядка $10^5$) для построения эффективной модели, а также для сокращения затрат времени и ресурсов памяти ЭВМ перед построением диагностической модели качества лопаток необходимо сократить количество признаков путем отбора наиболее информативных.

Поскольку в данной задаче мы имеем дело с наборами однотипных упорядоченных признаков, их можно сжать с помощью свертки, например, заменять группы смежных признаков на их максимальные, средние, или минимальные значения. также эффективным на практике оказывается вычисление интегральной суммы в области группы смежных признаков. В экспериментах, проведенных нами, исходный набор данных был сжат до 100 интегральных сумм групп смежных признаков. Несмотря на значительное (более чем в 100 раз) сокращение размера набора признаков для полученного набора из 100 признаков задача сокращения размерности путем отбора информативных признаков.

Одним из способов, позволяющих быстро и наглядно решить поставленную задачу является одновременная визуализация значений признаков лопаток на графиках.

На на рис. 7.3-7.12 изображены графики признаков лопаток в различных координатных системах. Графически значения для годных (кондиционных) лопаток выделены черным цветом, а негодных (дефектных, некондиционных) - серым. На оси абсцисс размещены номера признаков, на оси ординат - значения признаков в соотвтетсвующей системе координат.

Как видно из рис. 7.3 – 7.12, спектры мощности и разности полупериодов колебаний лопаток разных классов на отдельных участках графиков имеют разделимость, в некоторых случаях даже линейную, что свидетельствует о принципиальной возможности классификации лопаток по этим признакам.

Кроме графиков и преобразований рис. 7.3-7.12, при отборе информативных признаков также можно использовать графики модулей коэффициентов корреляции признаков и классов (рис. 7.13, 7.14).

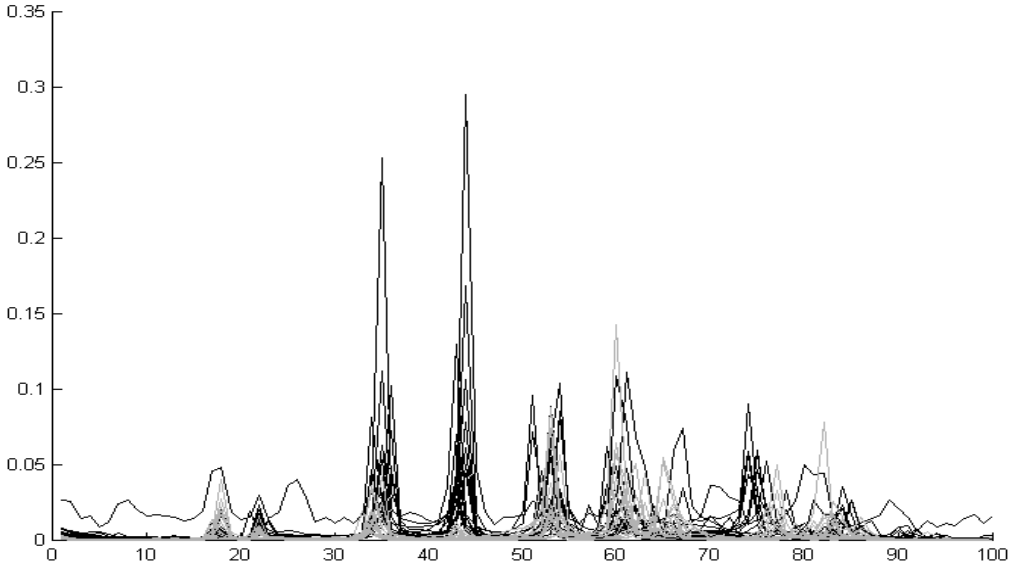



Рис. 7.3 – Спектры свободных затухающих колебаний лопаток после ударного возбуждения в декартовой системе координат.

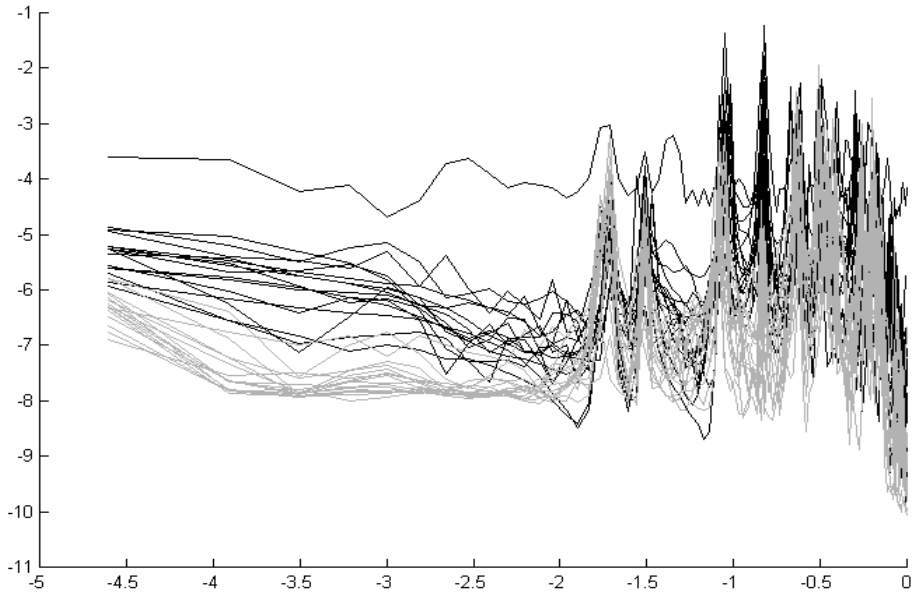



Рис. 7.4 – Спектры свободных затухающих колебаний лопаток после ударного возбуждения в логарифмической системе координат.

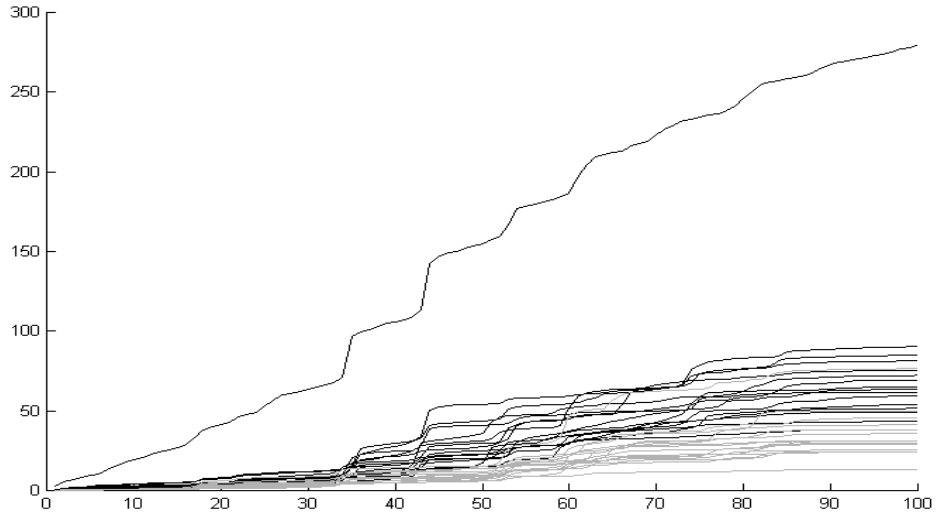



Рис. 7.5 – Кумулятивные суммы мощностей спектров свободных затухающих колебаний лопаток после ударного возбуждения.

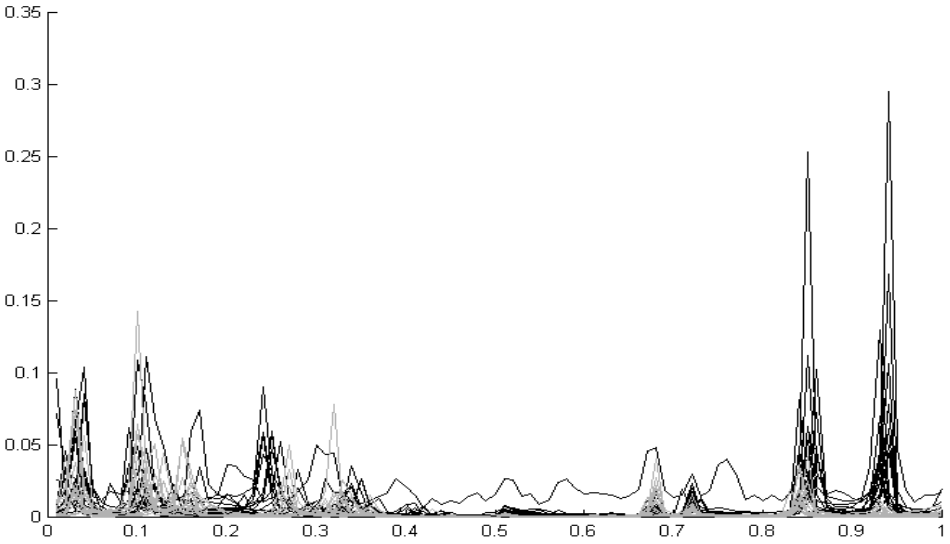



Рис. 7.6 – Спектры свободных затухающих колебаний лопаток после ударного возбуждения в системе координат смещенного дискретного быстрого преобразования Фурье.

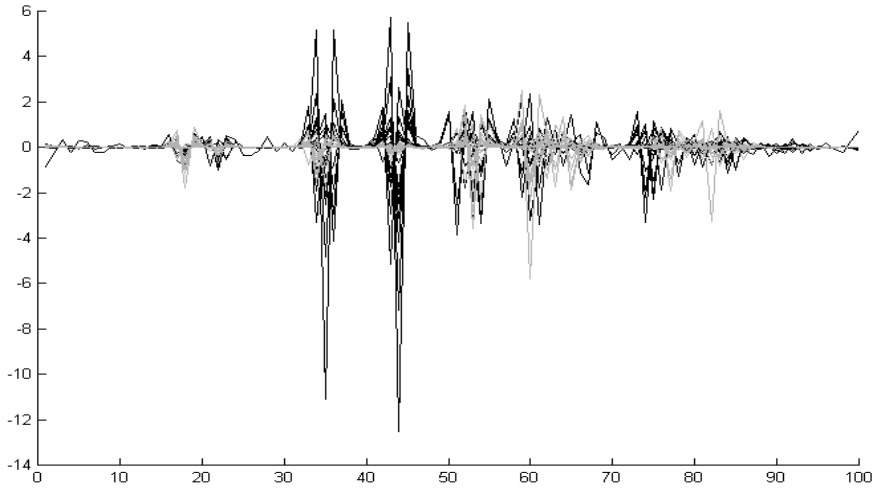



Рис. 7.7 – Лапласианы спектров свободных затухающих колебаний лопаток после ударного возбуждения.

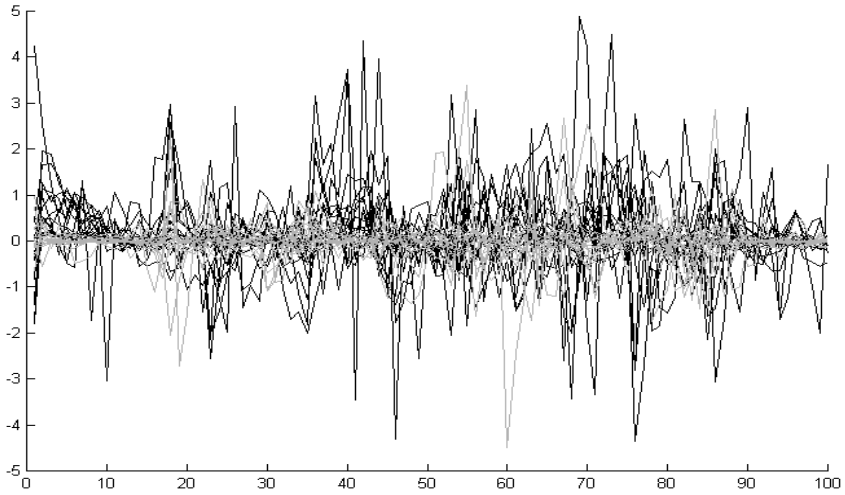



Рис. 7.8 – График разностей полупериодов свободных затухающих колебаний лопаток после ударного возбуждения в декартовой системе координат.

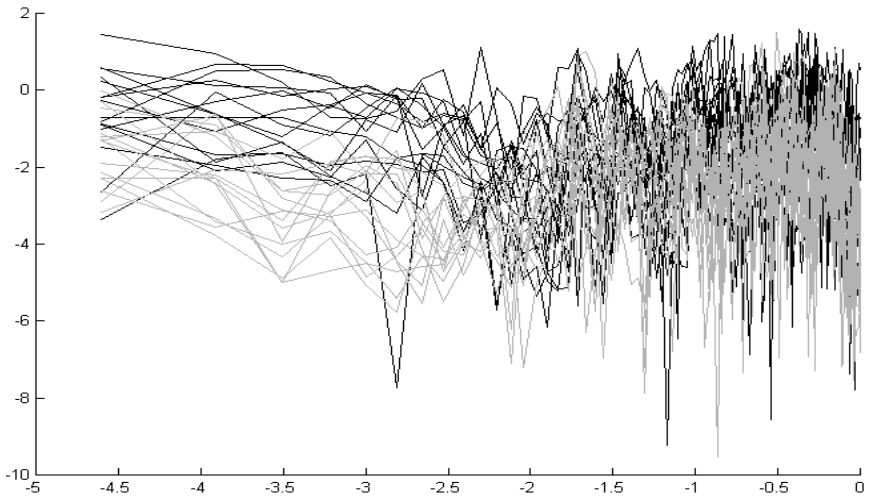



Рис. 7.9 – График разностей полупериодов свободных затухающих колебаний лопаток после ударного возбуждения в логарифмической системе координат.

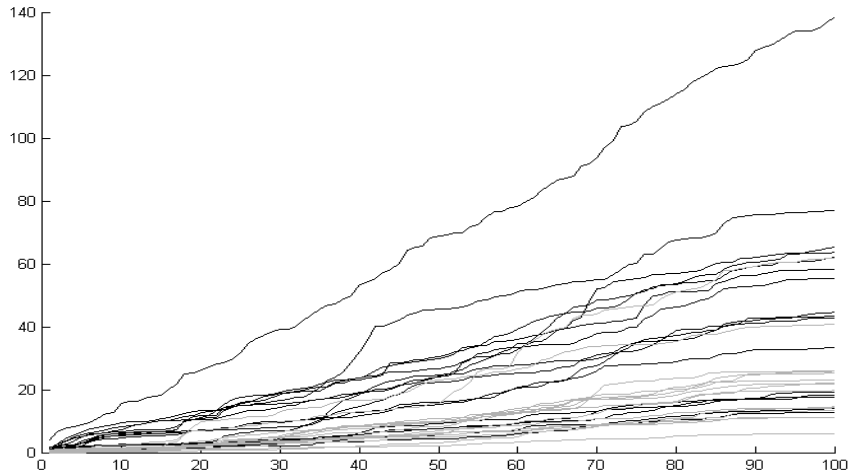



Рис. 7.10 – Кумулятивные суммы амплитуд разностей полупериодов свободных затухающих колебаний лопаток после ударного возбуждения.

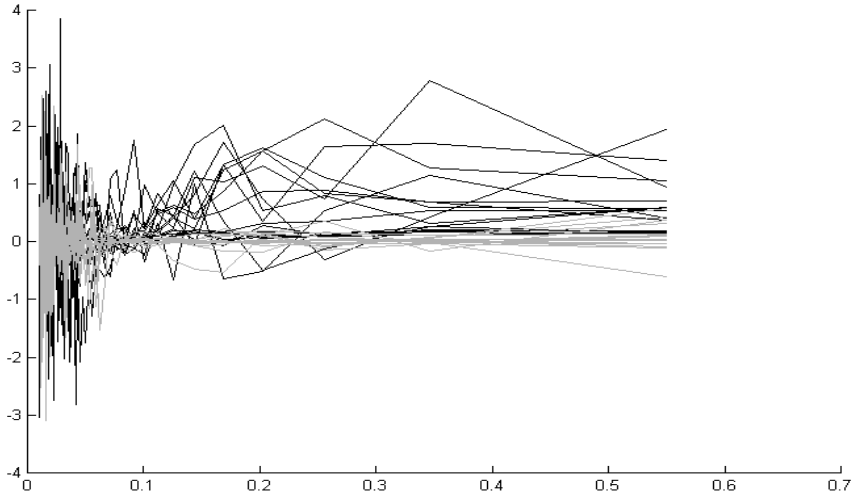



Рис. 7.11 – График разностей полупериодов свободных затухающих колебаний лопаток после ударного возбуждения в тангенциальной системе координат.

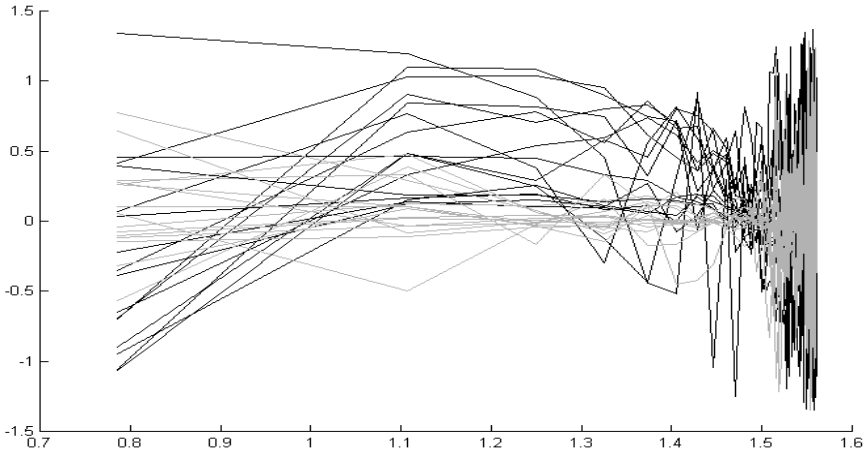



Рис. 7.12 – График разностей полупериодов свободных затухающих колебаний лопаток после ударного возбуждения в гиперболической тангенциальной системе координат.

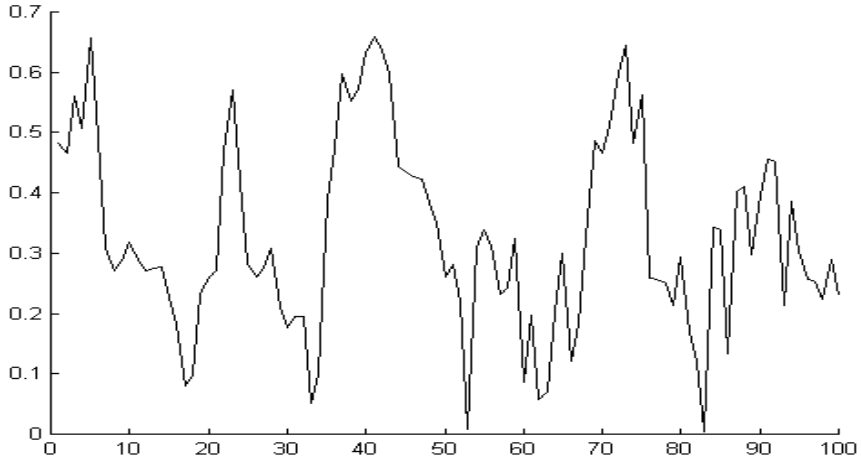



Рис. 7.13 – График модулей коэффициентов корреляции спектров свободных затухающих колебаний лопаток после ударного возбуждения и номера класса.

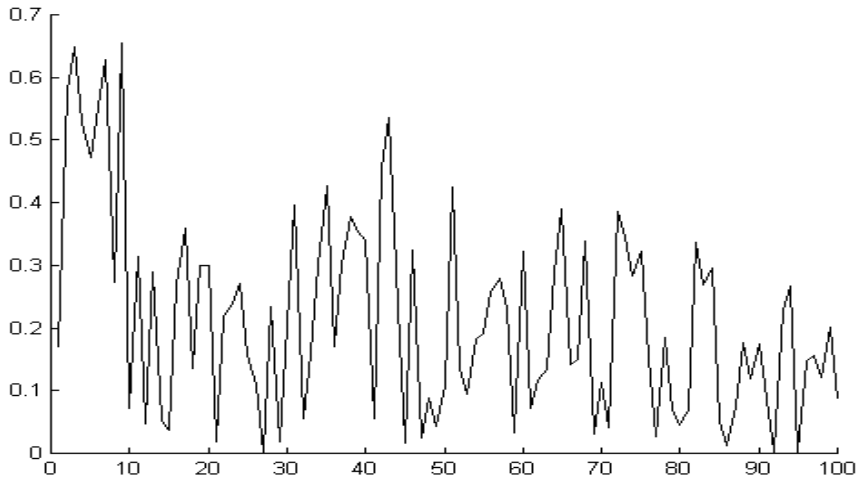



Рис. 7.14 – График модулей коэффициентов корреляции амплитуд разностей полупериодов свободных затухающих колебаний лопаток после ударного возбуждения и номера класса.

В табл. 7.1 приведены результаты выделения человеком информативных участков графиков признаков лопаток в различных координатных системах. Информативными считаются участки графиков, где годные и дефектные изделия хорошо разделимы.

Таблица 7.1 – Информативные области графиков, выделенные в результате анализа признаков

| координатная система или график преобразования | информативные области графиков в соответствующих координатных системах, масштабах и единицах | |
|---|---|---|
| | спектры колебаний | разности полупериодов колебаний |
| декартова | 20..24, 33..37, 41..46 | 0..15, 38..43 |
| логарифмическая | -4,6..-2, -1,25..-0,75 | -4..-3 |
| тангенциальная | - | 0,08..0,12, 0,3..0,5 |
| гиперболическая тангенциальная | - | 1,1..1,3 |
| кумулятивные суммы | 40..100 | 50..100 |
| быстрое дискретное смещенное преобразование Фурье | 31..38, 40..47 | - |
| лапласиан | 0,2..0,27, 0,7..0,75, 0,82..0,88, 0,9..0,97 | - |
| модули коэффициентов корреляции признаков и классов (в декартовой системе) | 3..6, 22..24, 40..46, 70..75 | 2..9, 30..39, 42..44 |

**После когнитивного выделения информативных групп признаков человеком, а также задания набора сверток, на основе алгоритма итеративного подбора свертки для каждого участка выбиралась наилучшая свертка, после**

**чего значения сверток для всех экземпляров использовались для многомерной классификации на основе многослойного персептрона – нейронной сети прямого распространения, обучавшейся с помощью алгоритма Левенберга-Марквардта.**

Эксперименты показали, что алгоритм итеративного подбора свертки способен подобрать наилучшие свертки из заданного набора для заданных участков спектра, но качество классификации в этом случае будет сильно зависеть от правильности задания информативных участков спектра, то есть основным возмущающим фактором здесь будет человек.

В другой серии экспериментов задавался только набор сверток, а для выделения информативных участков использовался алгоритм сжатия набора признаков. После отбора информативных участков и подбора сверток для них осуществлялась многомерная классификация лопаток так же, как и в предыдущем случае.

Эксперименты второй серии показали, что алгоритм сжатия набора признаков способен существенно сократить количество признаков путем выделения информативных участков спектров и подобрать для этих участков свертки, обеспечивающие требуемый уровень надежности диагностики лопаток.

Нейросетевая классификация на основе выделенных признаков, сжатых с помощью подобранных сверток, была безошибочной для обучающей и тестовой выборок, что свидетельствует о том, что выбранные признаки обладают достаточной информативностью.

На ряду с методами сжатия данных на основе сверток для сокращения размерности диагностической информации можно применять методы отбора информативных признаков, среди которых следует особо выделить методы оценки информативности и отбора признаков на основе нейронных сетей.

Эксперименты по нейросетевому отбору признаков для диагностики лопаток показали, что исходный набор признаков может быть сокращен более чем в 5 раз без потери точности распознавания, при этом время распознавания также сокращается в 5 раз.

Качественные и количественные оценки значимости признаков, полученные на основе рассмотренных алгоритмов вполне соответствовали физическому смыслу.

Результаты экспериментов показали, что однослойный персептрон имеет смысл использовать для оценки значимости признаков только, если связи между входами НС и ее выходом близки к линейным. Если связи – существенно нелинейные, то целесообразно использовать МНС.

Кроме вышерассмотренных методов для сокращения размерности обучающих данных при построении диагностических моделей целесообразно применять алгоритм разбиения исходной выборки на обучающую и тестовую.

В частности, для задачи диагностики лопаток авиадвигателей исходная выборка, содержавшая 32 экземпляра (100 признаков) была разбита на обучающую (13 экз.) и контрольную (19 экз.) выборки, что позволило обучить нейронную сеть с такой же точностью, как и для 32 экз., но за существенно меньшее время.

### 7.1.3 Классификация лопаток

Для классификации лопаток могут применяться различные методы теории распознавания образов, а также нейронные сети. Однако из-за малого объема обучающей выборки применение вероятностных и статистических методов в данном случае затруднено. Поэтому для построения диагностических моделей лопаток будем применять топологические и нейросетевые методы, которые менее требовательны к обучающим данным.

Рассмотрим особенности применения МНС для классификации лопаток. Поскольку обучающий набор данных имеет большую размерность при построении нейросетевых моделей качества лопаток возникает необходимость выбора наиболее быстрого и точного алгоритма обучения МНС. Поэтому представляет интерес исследовать, какой частный случай обобщенного градиентного алгоритма (то есть какой градиентный алгоритм обучения МНС) является лучшим.

Из анализа рассмотренных градиентных алгоритмов обучения МНС следует, что их сходимость зависит, кроме обучающих данных и начальных весов, еще и от

задаваемых параметров: максимальной допустимой общей ошибки обучения (цель обучения) и количества допустимых циклов обучения (критерий скорости обучения) для всей выборки. Эти два критерия целесообразно использовать для сравнения методов обучения МНС. Зафиксировав поочередно каждый из них, можно исследовать, как для данного алгоритма обучения меняются значения другого критерия при одинаковых наборах различных обучающих выборок и одинаковых начальных весах.

Результаты экспериментов следует оценивать по таким критериям, как время работы алгоритма обучения и затраченное количество циклов обучения. Фрагменты результатов экспериментов по сравнению методов обучения МНС приведены в табл. 7.2 и 7.3, а также на рис. 7.15.

Как видно из рис. 7.15 и табл. 7.2 и 7.3, наилучшим среди всех методов обучения МНС оказался алгоритм Левенберга-Марквардта. Он является наиболее быстрым среди всех алгоритмов и менее сложным с вычислительной точки зрения, чем метод Ньютона.

Метод Ньютона является наиболее сложным с вычислительной точки зрения и самым требовательным к условиям применения, но, вместе с тем, он превосходит алгоритмы сопряженных градиентов как по скорости обучения, так и по точности.

Алгоритмы сопряженных градиентов Полака-Рибьера и Флетчера-Ривса являются конкурирующими и уступают как по скорости, так и по точности методу Ньютона и алгоритму Левенберга-Марквардта.

Таблица 7.2 - Сравнительная характеристика алгоритмов обучения МНС при фиксированной максимальной точности

| Алгоритм | Фиксированная максимальная точность (ошибка) | | | | | |
|---|---|---|---|---|---|---|
| | $10^{-1}$ | | $10^{-3}$ | | $10^{-6}$ | |
| | Время обуче-ния, с | Число циклов обуче-ния | Время обуче-ния, с | Число циклов обуче-ния | Время обуче-ния, с | Число циклов обуче-ния |
| Ньютона | 2.03 | 3 | 8.68 | 26 | 9.78 | 45 |
| Флетчера-Ривса | 1.87 | 2 | 10.55 | 40 | 27.9 | 172 |
| Полака-Рибьера | 1.87 | 2 | 10.93 | 41 | 26.04 | 157 |
| Левенберга – Марквардта | 1.91 | 3 | 5.01 | 13 | 5.91 | 14 |

Таблица 7.3 - Сравнительная характеристика алгоритмов обучения МНС при фиксированном максимальном числе циклов обучения

| Алгоритм | Фиксированное максимальное число циклов обучения | | | | | |
|---|---|---|---|---|---|---|
| | 10 | | | 100 | | |
| | Время обуче-ния, с | Достиг-нутая точность | Затрачен-ное число циклов обучения | Время обуче-ния, с | Достиг-нутая точность | Затрачен-ное число циклов обучения |
| Ньютона | 3.29 | 0.007 | 10 | 20.49 | $7.81 \cdot 10^{-8}$ | 100 |
| Флетчера-Ривса | 3.3 | 0.01 | 10 | 17.63 | $1.11 \cdot 10^{-4}$ | 100 |
| Полака-Рибьера | 2.9 | 0.01 | 10 | 17.31 | $3.79 \cdot 10^{-5}$ | 100 |
| Левенберга – Марквардта | 2.63 | $3.01 \cdot 10^{-8}$ | 10 | 4.66 | $2.25 \cdot 10^{-11}$ | 16 |

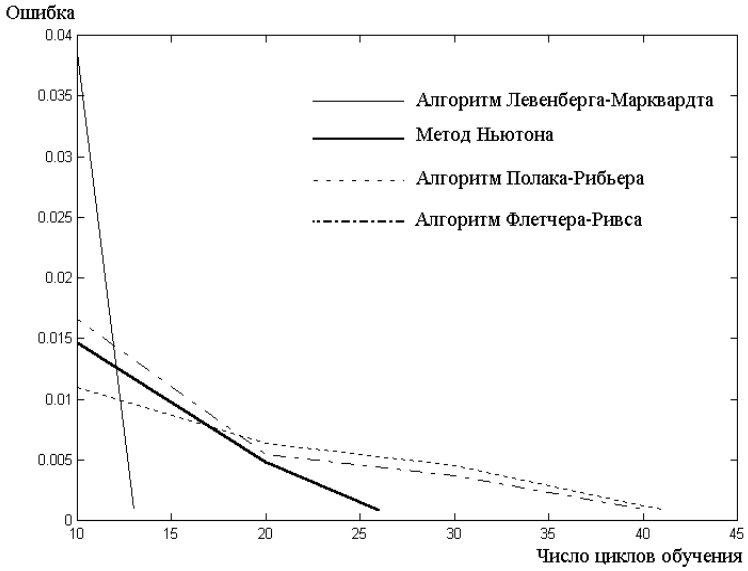



Рис. 7.15 - График сходимости алгоритмов обучения многослойных нейронных сетей

Результаты экспериментов свидетельствуют о работоспособности обобщенного градиентного алгоритма, о его больших адаптивных возможностях и позволяют рекомендовать обобщенный градиентный алгоритм обучения МНС для внедрения и использования на практике.

Несмотря на то, что алгоритм Левенберга-Марквардта является одним из наиболее быстрых методов обучения МНС, при построении нейросетевых моделей многомерных нелинейных зависимостей по выборкам большой размерности скорость работы алгоритма Левенберга-Марквардта может оказаться довольно низкой. Поэтому для ускорения процесса сходимости алгоритма Левенберга - Марквардта целесообразно применение следящего алгоритма.

Для сравнения следящего алгоритма и наиболее быстрого градиентного алгоритма Левенберга-Марквардта на основе МНС решались задачи классификации объектов и прогнозирования значений параметров сложных изделий по реальным данным. Результаты обучения представлены на рис. 7.16 и рис. 7.17.

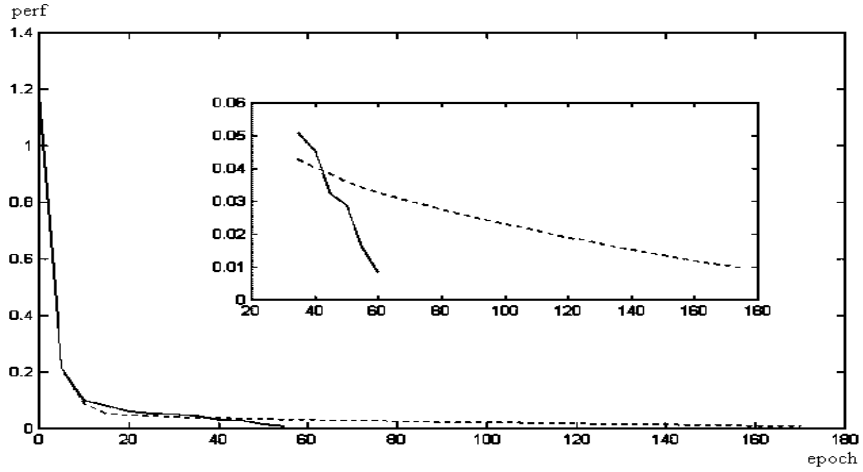



Рис. 7.16 - Графики убывания ошибки perf при обучении НС классификации (пунктирной линией выделен график ошибки алгоритма Левенберга-Марквардта, сплошной линией – следящего алгоритма).

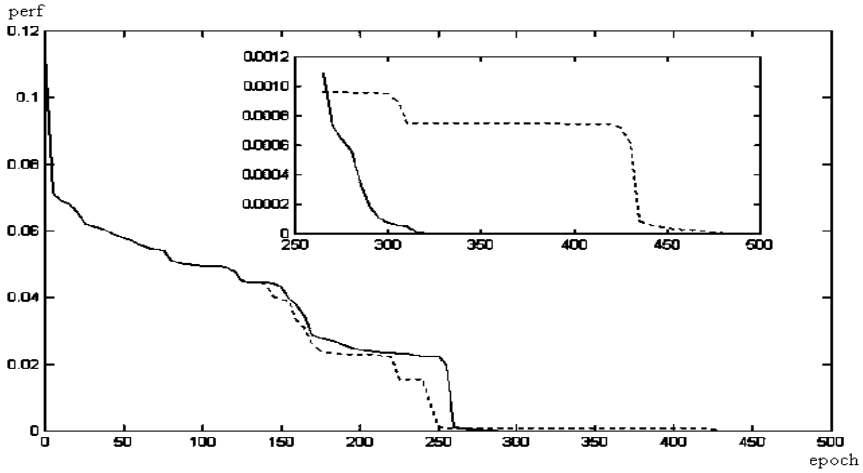



Рис. 7.17 - Графики убывания ошибки perf при обучении НС численной аппроксимации функций (пунктирной линией выделен график ошибки алгоритма Левенберга-Марквардта, сплошной линией – следящего алгоритма).

Как видно из рис. 7.16 и рис.7.17, следящий алгоритм позволяет существенно сократить время обучения МНС и увеличить скорость сходимости градиентного алгоритма. В разных экспериментах время обучения МНС с использованием следящего алгоритма сокращалось на 20 – 60 %, что является очень хорошим результатом.

Анализируя графики сходимости градиентного и следящего алгоритмов, можно сделать следующие выводы:

1) в начале оба алгоритма обеспечивают одинаковую сходимость, что объясняется двумя причинами: следящий алгоритм включается только после выполнения первых $\Delta t$ циклов обучения и сходимость, обеспечиваемая градиентным алгоритмом, в начале обучения довольно высока и не требует включения следящего алгоритма;

2) после прохождения начальной фазы обучения градиентный алгоритм в течение некоторого времени обеспечивает более быструю сходимость, чем следящий.

Это, по-видимому, объясняется тем, что даже изменения, уменьшающие значения целевой функции, вносимые следящим алгоритмом, могут иногда негативно влиять на общий процесс сходимости, т.е. мешают работе градиентного алгоритма на данном этапе.

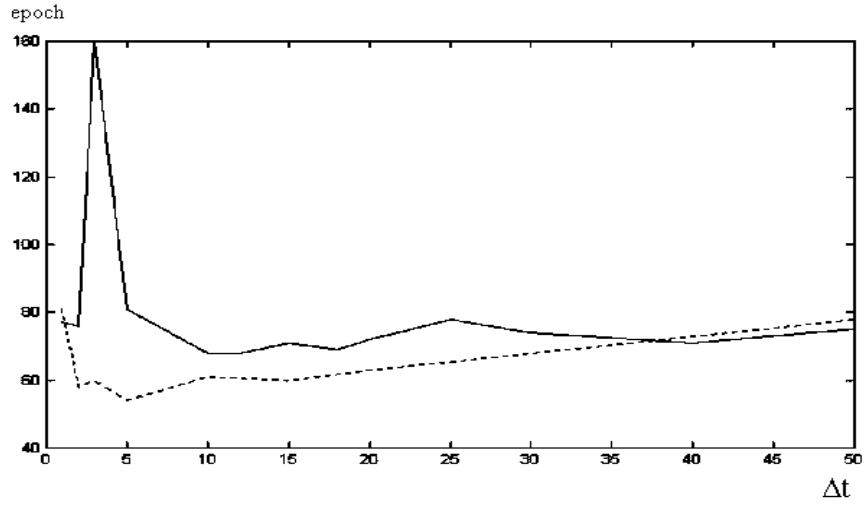
3) на завершающей стадии обучения, когда градиентный алгоритм обучения НС начинает блуждать по локальным минимумам целевой функции и обеспечивает очень медленную сходимость, следящий алгоритм обеспечивает существенно более быструю сходимость и превосходит по скорости градиентный алгоритм.

Следует отметить, что применение следящего алгоритма для решения задач классификации легко разделимых образов может не давать выигрыша в скорости обучения, поскольку из-за высокой скорости сходимости градиентного алгоритма следящий алгоритм в работу не будет вступать.

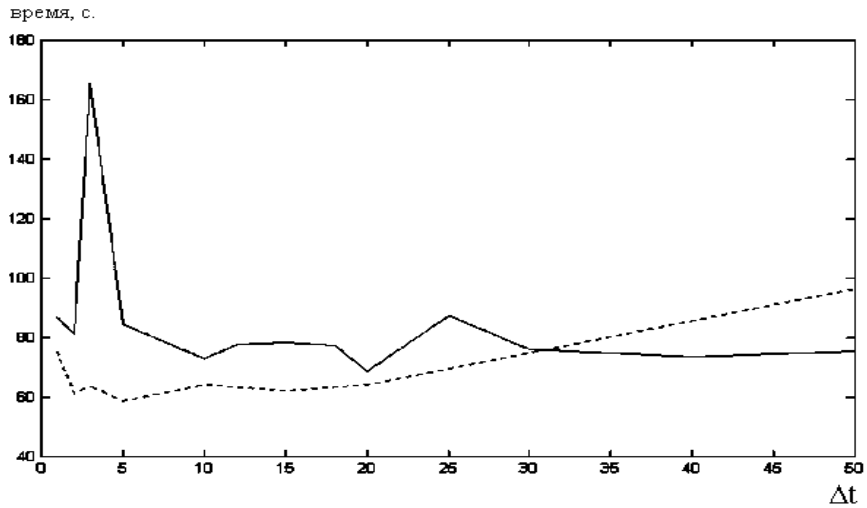
Эксперименты по обучению МНС решению практических задач на основе следящего алгоритма позволяют рекомендовать правила (5.1) и (5.2) для обучения НС классификации, а правило (5.3) – для обучения аппроксимации. Применение правил (5.1) и (5.2) для обучения аппроксимации нецелесообразно. Это объясняется тем, что данные правила способны вносить довольно сильные возмущения в набор весов сети.

Из описания следящего алгоритма следует, что, кроме выбора соответствующего правила R(w), необходимо задавать значения определенных

параметров. Для того, чтобы выяснить, какие значения наиболее приемлемы для соответствующих параметров следящего алгоритма проводились эксперименты, фрагменты результатов которых представлены на рис. 7.18 и 7.19.




a)




b)

Рис. 7.18 - Графики зависимостей количества циклов (a) и времени (b) обучения МНС от параметра $\Delta t$ для следящего алгоритма при фиксированном параметре $\alpha$ (сплошной линией выделены графики при $\alpha$=1.01, пунктирной линией – при $\alpha$=1.03).

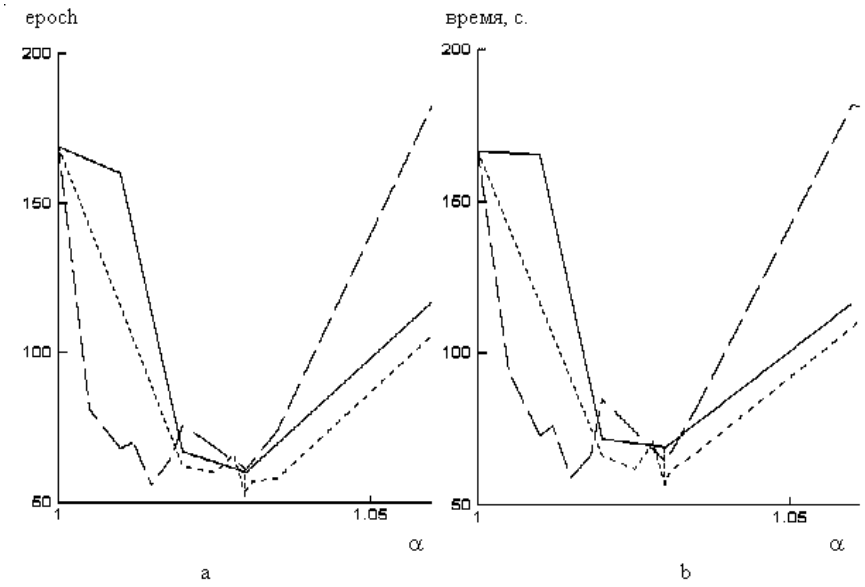



Рис. 7.19 – Графики зависимостей количества циклов (a) и времени (b) обучения МНС от параметра $\alpha$ для следящего алгоритма при фиксированном параметре $\Delta t$ (сплошной линией выделены графики при $\Delta t$=3, штриховой – при $\Delta t$=5, пунктирной – при $\Delta t$= 10).

Из рис. 7.18 и 7.19 легко видеть, что приемлемыми для большинства случаев могут быть следующие значения параметров следящего алгоритма: $\alpha$ = 1.03, $\Delta t$ = 6-20. Значение параметра $\xi$ следует принять равным 0.1.

Представляет интерес исследовать, как влияет размер окна слежения $\Delta t$ на количество успешных выполнений шага 9 следящего алгоритма. Результаты экспериментов представлены на рис. 7.20

Как видно из рис. 7.20, с уменьшением размера окна слежения следящий алгоритм применяется чаще, однако, как следует из рис. 7.17, при этом скорость сходимости снижается.

С другой стороны, на интервале значений $\Delta t \in [6,20]$ количество успешных выполнений шага 9 следящего алгоритма является средним и обеспечивает достаточно быструю сходимость. Это может служить дополнительным подтверждением обоснованности выбора вышеприведенных значений параметров следящего алгоритма.

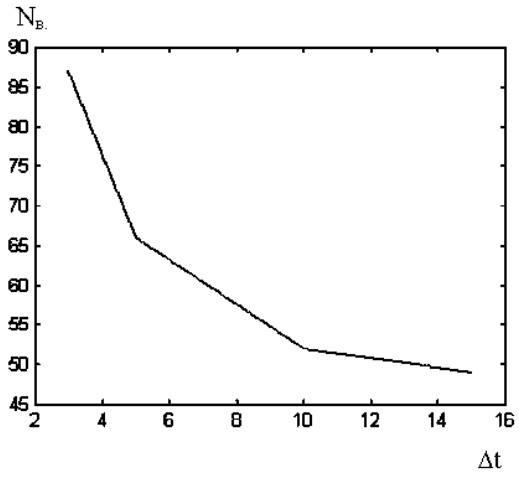



Рис. 7.20 - График зависимости количества успешных выполнений шага 9 $N_{в.}$ следящего алгоритма от размера окна слежения при фиксированном шаге $\alpha$=1.01.

Не смотря на высокую точность классификации, обеспечиваемую градиентными алгоритмами обучения МНС, для настройки весовых коэффифиентов МНС на практике имеет смысл применять эвристические алгоритмы в нейросетевой интерпретации.

Для эвристического алгоритма классификации (алгоритма обучения двухслойного персептрона) была выполнена программная реализация на языке макросов пакета MATLAB 5.2. Для сравнения разработанного алгоритма с алгоритмом обучения МНС Левенберга-Марквардта использовался модуль Neural Toolbox пакета MATLAB 5.2. При этом в качестве модели НС, обучавшейся на основе алгоритма Левенберга-Марквардта, использовался двуслойный персептрон, содержавший на первом слое столько нейронов, сколько и признаков, а на втором слое – 1 нейрон. В качестве цели обучения была задана среднеквадратическая ошибка 0.01, максимальное количество циклов обучения – 500.

**Результаты экспериментов показывают, что время обучения НС на основе алгоритма Левенберга-Марквардта, в целом, существенно больше чем время обучения НС на основе алгоритма эвристической классификации, но при этом алгоритм Левенберга-Марквардта обеспечивает гораздо меньшие вероятности принятия ошибочных решений. Поэтому на практике применять**

**разработанный метод эвристической классификации следует тогда, когда вероятность принятия ошибочных решений не будет превышать заданное значение.**

**Эвристический алгоритм является пригодным для решения многих практических задач (обладает универсальностью), хотя его конкретные реализации, зависящие от правила вычисления $\theta_i$, могут иметь более узкие области применения, что объясняется сложностью и несоответствием фактического разделения классов и разделения классов, формируемого правилом классификации эвристического алгоритма.**

**Эксперименты показывают, что надежность классификации на основе эвристического алгоритма тем выше, чем информативнее признаки. Скорость работы алгоритма тем больше, чем меньше признаков. Если количество признаков велико и все признаки обладают незначительно отличающейся информативностью, то применение данного алгоритма нецелесообразно.**

В результате можно сделать вывод о том, что эвристический алгоритм целесообразно применять в тех случаях, когда признаки достаточно информативны и граница между классами не очень сложная.

Для распознавания лопаток весьма эффективнымим на практике могут быть разновидности метода потенциальных функций, которые просты в реализации и не содержат сложных вычислений. Сравнительная характеристика методов приведена в табл. 7.4. и табл. 7.5.

Таблица 7.4 – Сравнительная характеристика методов классификации

| Метод классификации | время обучения, с. | время работы, с. |
|---|---|---|
| нерекуррентный метод потенциальных функций | 18.71 | 25.1 |
| модифицированный метод потенциальных функций | 0.68 | 0.82 |
| алгоритм многомерной классификации | 141.21 | 86.4 |
| классификация на основе нейронной сети, обученной на основе алгоритма Левенберга-Марквардта | 156.01 | 0.32 |

Как видно из табл. 7.4, наиболее скоростным методом обучения является модифицированный метод потенциальных функций, наименее скоростным – алгоритм многомерной классификации. Наиболее быстро работающим является метод нейросетевой классификации на основе НС, обученной с помощью алгоритма Левенберга-Марквардта, самым медленно работающим является метод многомерной классификации.

**Таблица 7.5 – Сравнительная характеристика методов классификации по качеству классификации**

| Метод классификации | $P_{ош.}$ обуч. | $P_{ош.}$ тест. |
|---|---|---|
| нерекуррентный метод потенциальных функций | 0.12 | 0.15 |
| модифицированный метод потенциальных функций | 0.12 | 0.15 |
| алгоритм многомерной классификации | 0.12 | 0.15 |
| классификация на основе нейронной сети, обученной на основе алгоритма Левенберга-Марквардта | 0 | 0.06 |

Как видно из табл. 7.5, наиболее точным является метод нейросетевой классификации (НС, обученная на основе алгоритма Левенберга-Марквардта), наименее точными являются методы потенциальных функций. Алгоритм многомерной классификации занимает промежуточное положение. Это объясняется тем, что НС обладают более сильными аппроксимационными способностями, чем другие рассмотренные методы, а алгоритм многомерной классификации, в отличие от методов потенциальных функций, учитывает значимость признаков посредством учета частных значимостей двупризнаковых классификаций.

Результаты экспериментов позволяют предложить следующие рекомендации по выбору метода классификации для использования на практике:

1) Алгоритм многомерной классификации следует использовать для классификации образов, характеризующихся не очень большим количеством признаков (от 3 до 20) там, где желательно добиться более высокой (в крайнем случае такой же) точности классификации чем методы потенциальных функций за

конечное заранее известное количество итераций в процессе обучения, чего нельзя сделать методом нейросетевой классификации.

2) Классификацию на основе НС, обученнной с помощью алгоритма Левенберга-Марквардта следует применять тогда, когда важно добиться заранее заданного уровня ошибки (например, в задаче диагностики лопаток ГТД).

3) Методы потенциальных функций следует использовать для задач большой размерности (более 30 признаков) там, где необходима высокая скорость обучения и работы. При этом нерекуррентный метод потенциальных функций, как правило, можно заменить модифицированным нерекуррентным методом потенциальных функций, который работает значительно быстрее, обеспечивая тот же уровень ошибок, что и метод потенциальных функций.

Для исследования практической применимости комбинированного метода классификациина его основе, а также (для сравнения) на основе метрической классификаци решалась задача диагностики лопаток газотурбинных авиадвигателей.

Результаты экспериментов приведены в табл. 7.6.

Таблица 7.6 – Сравнительная характеристика методов классификации при решении задачи классификации лопаток авиадвигателей

| Метод классификации | Время обучения, с. | Время классификации, с. | Вероятность принятия ошибочных решений |
|---|---|---|---|
| метрической классификации | 0.44 | 0.05 | 0.16 |
| комбинированный метод | 0.61 | 0.05 | 0.08 |

Как видно из табл. 7.6, оба метода обеспечивают приблизительно одинаковый процент правильного распознавания, но при этом комбинированный метод в некоторых случаях работает точнее, но в тоже время и немного медленее, чем метрический метод.

Кроме характеристик методов, приведенных в таблице, в экспериментах исследовалось, каково соотношение вероятностей выполнения шагов 3-4 и 5-6 на этапе распознавания для комбинированного метода. Как показывают результаты экспериментов соотношение классификации по принадлежности к области (шаги 3-4) и метрической классификации (шаги 5-6) для тестовой выборки составляет при классификации лопаток – 1:3.

Из этого можно сделать следующие выводы. Для повышения точности классификации необходимо увеличить долю уверенной классификации - классификации по принадлежности к области, для чего, в свою очередь, необходимо находить такие выражения для L(x) и их параметры, чтобы аппроксимация границ областей классов была бы близка к оптимальной, и предварительно нормировать значения признаков таким образом, чтобы области классов хорошо разделялись и были компактными. Чем ближе будет обучающая выборка к генеральной совокупности (по количеству экземпляров и репрезентативности), тем более надежным будет процесс обучения.

Результаты проведенных экспериментов позволяют рекомендовать применение предложенного комбинированного метода на практике.

Поскольку различные методы классификации обеспечивают разную точность при решении различных задач эффективным может быть объединение нескольких классификаторов в единую систему, что может быть осуществлено на основе рассмотренных построения гибридных классификаторов.

**Для апробации метода построения гибридных систем на основе нейронных сетей проводился ряд экспериментов по классификации многомерных наборов данных. Фрагменты результатов экспериментов приведены в табл. 7.7.**

Таблица 7.7 – Фрагмент результатов экспериментов по построению и обучению гибридной системы классификаторов

| классификатор | вероятность правильной классификации в разных опытах (в %) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| двухслойный персептрон | 100 | 100 | 85 | 97 | 98 | 100 | 82 | 97 |
| радиально-базисная сеть | 100 | 100 | 100 | 100 | 98 | 100 | 100 | 93 |
| сеть LVQ | 100 | 93,7 | 93,7 | 95 | 96 | 85 | 93,7 | 95 |
| интегрированная классификация на основе НС Кохонена | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

Как видно из табл. 7.7, метод интегрированной классифкации позволяет автоматически выделять наиболее значимые классификаторы, а также способен учитывать статистические характеристики входных данных.

Из таблицы также следует, что интеграция различных классификаторов с помощью НС Кохонена позволяет обеспечить на практике довольно высокий уровень надежности классификации.

Предложенная методология построения диагностических систем на практике позволяет обеспечивать достаточно надежную классификацию для многих приложений. Наличие методов, позволяющих наращивать и сокращать структуру основных блоков системы в зависимости от сложности задачи, позволяет системе гибко адаптироваться к изменению задачи и разделения классов.

## 7.2 Моделирование коэффициента упрочнения деталей авиадвигателей при алмазном выглаживании

Одной из важных задач при расчете запаса прочности деталей газотурбинных авиадвигателей (ГТД) и внедрении нового технологического процесса является предварительная оценка коэффициента упрочнения $\beta^{y}$ - отношения пределов выносливости упрочненной детали $\sigma_{-1}^{y}$ и детали, окончательно обработанной по серийной технологии шлифованием или полированием $\sigma_{-1}$: $\beta^{y} = \sigma_{-1}^{y} / \sigma_{-1}$.

Для определения коэффициента упрочнения необходимо провести испытания на усталость определенного числа деталей, что на стадии проектирования является дорогостоящей и трудновыполнимой задачей. В настоящее время расчет запаса прочности деталей выполняется по результатам испытания на усталость стандартных образцов с различными концентраторами напряжений. В этом случае не всегда соблюдается подобие напряженного состояния в зоне контакта при деформационном упрочнении и изменение коэффициента упрочнения $\beta^{y}$ при переходе от упрочненного образца к детали.

Эффективность алмазного выглаживания, которое нашло применение в авиадвигателестроении, в значительной мере зависит от выбранных режимов, физико-механических и геометрических характеристик упрочняемых деталей и деформирующего инструмента.

В задачу предыдущих исследований для решения данной задачи входило получение с помощью теории подобия и анализа размерностей математической модели коэффициента упрочнения с участием параметров процесса алмазного выглаживания, физико-механических характеристик материалов деталей и инструмента с учетом изменения эффективности упрочнения при наличии концентрации напряжений и масштабного фактора деталей на этапе проектирования и внедрения технологического процесса.

В качестве факторов, наиболее полно отражающих процесс алмазного выглаживания деталей, предлагается использовать:

1). НВ, МПа – твердость материала;

2). $q_{max}$, МПа – среднее контактное давление в зоне деформирования;

3). a, мм –полуось эллипса касания в зоне упругого контакта;

4). s, мм / об – подача при выглаживании;

5). $\sigma_B$, МПа – предел прочности;

6). $\sigma_{0.2}$, МПа – предел текучести материала;

7). n – показатель деформационного упрочнения;

8). $\alpha_\sigma^{техн}$ – теоретический коэффициент концентрации напряжений от следов обработки;

9). $R_{a1}$, мкм – параметр исходной шероховатости детали;

10). $P_y$, Н – сила выглаживания;

11). $R_{сф}$, мм – радиус сферы алмазного инструмента;

12). $R_{a2}$, мкм – параметр шероховатости после выглаживания детали;

13). $\alpha_\sigma$ – теоретический коэффициент концентрации напряжений натурной детали (образца);

14). d, мм – диаметр образца в опасном сечении;

15). r, мм – радиус скругления галтели или надреза;

16). $\overline{G}$, мм$^{-1}$ – относительный градиент первого главного напряжения;

На основе полученных экспериментальных данных в предыдущих исследованиях строились статистические модели коэффициента упрочнения, которые в отдельных случаях допускали погрешность при расчете $\beta^y$ свыше 10 %, что является недостаточно точной оценкой $\beta^y$.

Задачей настоящего исследования было получение более точной модели коэффициента упрочнения деталей при алмазном выглаживании, а также оценка значимости (информативности) факторов, используемых при построении модели.

Для построения модели коэффициента упрочнения предлагается использовать искусственные НС, обладающие способностью к аппроксимации многомерных функциональных зависимостей по точечным данным. После построения нейросетевой модели можно осуществить анализ информативности факторов, на основе которых осуществляется оценка значения коэффициента упрочнения, с

целью упрощения модели и повышения ее достоверности. Для этого предлагается использовать комбинацию корреляционного анализа с нейросетевой оценкой информативности признаков.

Для моделирования коэффициента упрочнения были использованы результаты испытаний на усталость 57 партий образцов диаметром от 7.5 до 60 мм, изготовленных из высоколегированных сталей и сплавов марок 40ХН2МАШ, 13Х11Н2В2МФШ, 12ХН3А, 14Х17Н2Ш, ХН77ТЮР, Х12НМБФШ, 18Х15Н3МАШ.

Алмазное выглаживание образцов производилось инструментами с радиусами сферы от 0.8 до 3 мм. Твердость исследуемых материалов находилась в пределах НВ = 2350-3300 МПа, предел прочности $\sigma_В$ = 950 - 1150 МПа, предел текучести $\sigma_{0.2}$ = 600 - 1000 МПа, показатель деформационного упрочнения n = 0,103 - 0,131, сила выглаживания $P_y$ = 100 - 500 Н, подача s = 0,03 - 0,16 мм/об., относительный градиент первого главного напряжения $\overline{G}$ = 0.43 - 2.51 мм$^{-1}$. Для гладких образцов $\overline{G}_0$ = 0,5 мм$^{-1}$ (d =7,5 мм и r = 10 мм).

Испытания на усталость проводились на электромагнитной установке в режиме резонансных колебаний ($\nu$ = 310 - 320 Гц) при плоском знакопеременном изгибе консольно закрепленного образца и при чистом изгибе с вращением на машине МВП-10000 ($\nu$ = 50 Гц). Для каждой партии (10 - 12 образцов) определяли пределы выносливости упрочненных образцов $\sigma^y_{-1}$ и исходных образцов $\sigma_{-1}$ для вероятности разрушения Р=50%.

Фрагмент результатов испытаний на усталость и значения факторов представлены в табл. 7.8.

Таблица 7.8 - Фрагмент экспериментальных данных

| номер экземп-ляра | Номер признака | | | | | | | | | | | | | | | | $\beta^y$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |
| 1 | 2350 | 3390 | 0.14 | 0.08 | 900 | 650 | 0.116 | 1.45 | 1.1 | 100 | 3 | 0.25 | 1 | 10 | 10 | 0.43 | 1.16 |
| 6 | 3300 | 5630 | 0.219 | 0.085 | 1150 | 1000 | 0.126 | .35 | 0.85 | 400 | 3 | 0.12 | 1 | 10 | 10 | 0.43 | 1.39 |
| 10 | 2700 | 5570 | 0.22 | 0.08 | 950 | 600 | 0.103 | .35 | 0.85 | 400 | 3 | 0.6 | 1 | 10 | 10 | 0.43 | 1.51 |
| 12 | 3300 | 5110 | 0.199 | 0.06 | 1150 | 1000 | 1.126 | 1.35 | 0.85 | 300 | 3 | 0.1 | 1 | 10 | 10 | 0.43 | 1.38 |
| 20 | 2300 | 3900 | 0.126 | 0.08 | 950 | 850 | 0.106 | 1.25 | 0.23 | 110 | 3 | 0.14 | 1 | 7.5 | 10 | 0.50 | 1.27 |
| 21 | 3100 | 5500 | 0.202 | 0.08 | 960 | 850 | 0.131 | 1.45 | 1.1 | 300 | 2.5 | 0.19 | 1 | 10 | 10 | 0.43 | 1.45 |
| 22 | 2850 | 4860 | 0.196 | 0.08 | 1000 | 700 | 0.103 | 1.35 | 0.85 | 250 | 2.5 | 0.12 | 1 | 10 | 10 | 0.43 | 1.50 |
| 25 | 2600 | 3970 | 0.125 | 0.08 | 1000 | 800 | 0.126 | 1.3 | 0.7 | 110 | 3 | 0.07 | 1 | 7.5 | 10 | 0.5 | 1.44 |
| 27 | 3300 | 5630 | 0.219 | 0.08 | 1150 | 1000 | 0.126 | 1.45 | 1.15 | 400 | 3 | 0.14 | 1.15 | 10 | 6 | 0.575 | 1.29 |
| 30 | 2350 | 3140 | 0.40 | 0.08 | 950 | 850 | 0.106 | 1.25 | 0.63 | 200 | 2.5 | 0.40 | 1.45 | 60 | 10 | 0.30 | 1.57 |
| 32 | 2550 | 5330 | 0.224 | 0.08 | 960 | 770 | 0.115 | 1.25 | 0.65 | 400 | 3 | 0.11 | 1.0 | 10 | 10 | 0.43 | 1.47 |
| 33 | 3300 | 4810 | 0.176 | 0.085 | 1150 | 1000 | 0.126 | 1.8 | 2.8 | 200 | 2.5 | 1.0 | 1.0 | 10 | 10 | 0.43 | 1.43 |
| 38 | 2700 | 4780 | 0.177 | 0.08 | 950 | 600 | 0.103 | 1.8 | 2.8 | 200 | 2.5 | 0.35 | 1.0 | 10 | 10 | 0.43 | 1.49 |
| 44 | 2350 | 4620 | 0.18 | 0.08 | 900 | 650 | 0.116 | 1.8 | 2.8 | 200 | 2.5 | 0.35 | 1.0 | 10 | 10 | 0.43 | 1.66 |
| 50 | 2300 | 3900 | 0.126 | 0.08 | 950 | 850 | 0.106 | 1.7 | 2.5 | 110 | 3.0 | 0.6 | 1.0 | 7.5 | 10 | 0.5 | 1.20 |
| 52 | 2600 | 3970 | 0.125 | 0.1 | 1000 | 800 | 0.126 | 1.65 | 2.4 | 110 | 3 | 0.30 | 1.0 | 7.5 | 10 | 0.5 | 1.13 |
| 54 | 3300 | 3480 | 0.139 | 0.08 | 1180 | 1000 | 0.126 | 1.65 | 2.4 | 100 | 3 | 1.2 | 2.52 | 10 | 0.25 | 8.2 | 1.83 |
| 58 | 2550 | 5330 | 0.21 | 0.10 | 960 | 770 | 0.115 | 1.65 | 2.3 | 400 | 3 | 0.25 | 1.0 | 10 | 10 | 0.43 | 1.35 |

Моделирование коэффициента упрочнения осуществлялось с помощью двуслойного персептрона, первый слой которого содержал 4 нейрона, а второй слой – 1 нейрон. Все нейроны имели сигмоидную функцию активации $\psi(x)=1/(1+e^{-x})$.

На входы НС подавались значения факторов. На выход НС подавалось значение коэффициента упрочнения для соответствующего образца.

В качестве целевой функции при обучении использовался минимум среднеквадратической ошибки сети для всей выборки goal=$10^{-6}$.

Обучение НС производилось на основе алгоритма Левенберга-Марквардта. При обучении НС значение $\eta$ полагалось равным 0.9, шаг обучения 0.00001, максимальное число циклов обучения НС epochs=500.

Матрица весовых коэффициентов, полученная в результате обучения НС, представлена в табл. 7.9.

Таблица 7.9 - Матрица весовых коэффициентов НС $w_j^{(\mu,i)}$ - параметров нейросетевой модели коэффициента упрочнения

| j – номер входа нейрона | i – номер нейрона в слое | | | | μ - номер слоя |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| 0 | 14,1894 | -5,1831 | -10,8277 | -5,3176 | 1 |
| 1 | -21,1447 | 4,6313 | -0,7001 | 0,9456 | |
| 2 | -7,8715 | -3,2767 | 0,4041 | 3,4354 | |
| 3 | -11,5879 | 35,883 | 6,7379 | 5,2636 | |
| 4 | -2,8764 | 15,19 | -1,1862 | 2,0939 | |
| 5 | 34,8108 | -4,2685 | 0,3106 | -2,1849 | |
| 6 | -14,0461 | 11,9728 | -0,4762 | 2,2898 | |
| 7 | -1,3948 | 28,2757 | -36,5509 | 19,7894 | |
| 8 | -4,4415 | -0,8422 | -1,7184 | 3,4117 | |
| 9 | -5,6667 | -8,1141 | 13,0482 | 1,4346 | |
| 10 | 6,1819 | -7,5646 | -2,5819 | -1,7134 | |
| 11 | -1,5746 | -10,854 | -3,4903 | 2,2829 | |
| 12 | 10,664 | 4,4055 | 5,3613 | -8,4166 | |
| 13 | -0,7028 | 0,5237 | -1,0671 | 1,2218 | |
| 14 | -21,3477 | 15,2453 | -8,8919 | 5,6718 | |
| 15 | -0,9858 | 7,8709 | 15,4614 | -2,6364 | |
| 16 | 0,4963 | -7,2036 | 30,9704 | 0,4219 | |
| 0 | -34,3184 | | | | 2 |
| 1 | 37,3173 | | | | |
| 2 | -32,6482 | | | | |
| 3 | 29,7206 | | | | |
| 4 | 63,3830 | | | | |

Результаты нейросетевого моделирования коэффициента упрочнения приведены в табл. 7.10. Здесь $\beta^{у}_{эксп.}$-значение коэффициента упрочнения, полученное экспериментально, $\beta^{у}_{расч.}$ – расчетное значение коэффициента упрочнения, полученное с помощью НС.

Таблица 7.10 - Результаты нейросетевого моделирования

| номер экземпляра | $\beta^{у}_{эксп.}$ | $\beta^{у}_{расч.}$ | номер экземпляра | $\beta^{у}_{эксп.}$ | $\beta^{у}_{расч.}$ | номер экземпляра | $\beta^{у}_{эксп.}$ | $\beta^{у}_{расч.}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1,16 | 1,16 | 21 | 1,45 | 1,45 | 41 | 1,58 | 1,58 |
| 2 | 1,27 | 1,27 | 22 | 1,5 | 1,5 | 42 | 1,6 | 1,6 |
| 3 | 1,38 | 1,38 | 23 | 1,61 | 1,61 | 43 | 1,56 | 1,56 |
| 4 | 1,54 | 1,54 | 24 | 1,64 | 1,64 | 44 | 1,66 | 1,66 |
| 5 | 1,46 | 1,46 | 25 | 1,44 | 1,44 | 45 | 1,6 | 1,6 |
| 6 | 1,35 | 1,35 | 26 | 1,13 | 1,1304 | 46 | 1,59 | 1,59 |
| 7 | 1,39 | 1,39 | 27 | 1,29 | 1,29 | 47 | 1,55 | 1,55 |
| 8 | 1,35 | 1,35 | 28 | 1,29 | 1,29 | 48 | 1,56 | 1,56 |
| 9 | 1,21 | 1,21 | 29 | 1,32 | 1,32 | 49 | 1,53 | 1,53 |
| 10 | 1,51 | 1,51 | 30 | 1,42 | 1,42 | 50 | 1,2 | 1,2 |
| 11 | 1,51 | 1,51 | 31 | 1,57 | 1,57 | 51 | 1,19 | 1,19 |
| 12 | 1,38 | 1,38 | 32 | 1,47 | 1,47 | 52 | 1,15 | 1,1499 |
| 13 | 1,37 | 1,37 | 33 | 1,43 | 1,43 | 53 | 1,13 | 1,1306 |
| 14 | 1,21 | 1,21 | 34 | 1,4 | 1,4 | 54 | 1,83 | 1,83 |
| 15 | 1,38 | 1,38 | 35 | 1,48 | 1,48 | 55 | 1,75 | 1,75 |
| 16 | 1,38 | 1,38 | 36 | 1,52 | 1,52 | 56 | 1,23 | 1,23 |
| 17 | 1,38 | 1,38 | 37 | 1,48 | 1,48 | 57 | 1,22 | 1,22 |
| 18 | 1,37 | 1,37 | 38 | 1,49 | 1,49 | 58 | 1,39 | 1,39 |
| 19 | 1,33 | 1,33 | 39 | 1,45 | 1,45 | 59 | 1,35 | 1,35 |
| 20 | 1,27 | 1,27 | 40 | 1,6 | 1,6 | | | |

Время обучения НС составило 106.7 с, количество затраченных циклов обучения 339, среднеквадратическая ошибка $9.94262 \cdot 10^{-7}$.

После получения нейросетевой модели коэффициента упрочнения осуществлялась нейросетевая оценка информативности факторов, а также были найдены коэффициенты корреляции факторов и коэффициента упрочнения. Результаты оценки информативностей факторов приведены в табл. 7.11.

Таблица 7.11 – Результаты оценки информативности факторов

| номер признака | Коэффициенты корреляции факторов и прогнозируемого параметра | Нейросетевая оценка информативности фактора | Решение об уровне информативности фактора |
|---|---|---|---|
| 1 | -0.088 | 0.0456 | малоинформативный |
| 2 | 0.288 | 0.0380 | информативный |
| 3 | 0.3076 | 0.1033 | информативный |
| 4 | -0.0769 | 0.0374 | малоинформативный |
| 5 | -0.199 | 0.0736 | информативный |
| 6 | -0.2947 | 0.0512 | информативный |
| 7 | -0.042 | 0.2096 | информативный |
| 8 | 0.1979 | 0.0314 | информативный |
| 9 | 0.1665 | 0.0459 | малоинформативный |
| 10 | 0.2139 | 0.0330 | информативный |
| 11 | -0.4946 | 0.0345 | информативный |
| 12 | 0.2442 | 0.0814 | информативный |
| 13 | 0.0428 | 0.0108 | малоинформативный |
| 14 | 0.1308 | 0.0993 | информативный |
| 15 | -0.0919 | 0.0491 | малоинформативный |
| 16 | 0.2875 | 0.0558 | информативный |

На основе полученных значений коэффициентов корреляции и оценок информативностей факторов принимались решения о разделении факторов на две

группы: информативные и малоинформативные. К информативным относились те факторы, информативности и коэффициенты корреляции которых превышали определенные пороговые значения, а к малоинформативным относились те факторы, информативности и коэффициенты корреляции которых были меньше определенных пороговых значений. При принятии решения об информативности признаков принимались следующие пороги значимости: для коэффициентов корреляции = 0.1979, для нейросетевых оценок информативностей признаков = 0.0625.

После принятия решений об информативности признаков из обучающего множества были исключены малоинформативные признаки (1,4,9,13,15). Затем осуществлялось повторное моделирование коэффициента упрочнения на основе НС. При этом все параметры НС и процесса обучения были такими же, как и в предыдущем случае, за исключением максимального числа циклов обучения, которое было увеличено epochs=1000.

Матрица весовых коэффициентов, полученная в результате обучения НС, представлена в табл. 7.12. Результаты нейросетевого моделирования коэффициента упрочнения приведены в табл. 7.13.

Таблица 7.12 - Матрица весовых коэффициентов НС $w_j^{(\mu,i)}$ - параметров нейросетевой модели коэффициента упрочнения после исключения малоинформативных факторов

| j – номер входа нейрона | i – номер нейрона в слое | | | | μ - номер слоя |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| 0 | -5,542 | -2,2031 | 13,5044 | -8,3408 | 1 |
| 1 | 5,3587 | 0,5508 | 43,6335 | 13,2709 | |
| 2 | -7,8206 | 4,0778 | 10,5403 | -25,7152 | |
| 3 | 3,1711 | 0,6497 | 6,5934 | 6,4903 | |
| 4 | -3,7075 | -0,6608 | -18,2564 | -7,8339 | |
| 5 | 2,8267 | 2,7226 | -0,4401 | 4,6906 | |
| 6 | 0,3677 | -0,4689 | -7,5889 | 1,222 | |
| 7 | -6,3121 | -2,4655 | -8,942 | -11,2141 | |
| 8 | 5,202 | 0,6805 | -30,9099 | 9,7942 | |
| 9 | -1,9664 | -0,2499 | -1,1736 | -3,1426 | |
| 10 | 11,0803 | -2,8783 | 29,7395 | 31,9021 | |
| 11 | 2,3601 | 0,5349 | -4,6556 | 3,7072 | |
| 0 | -1,4457 | | | | 2 |
| 1 | 213,0422 | | | | |
| 2 | -86,3955 | | | | |
| 3 | 6,6419 | | | | |
| 4 | -70,9734 | | | | |

Время обучения НС составило 190.87 с. для 1000 затраченных циклов обучения (для 500 циклов – 95.44 с.), среднеквадратическая ошибка $3{,}98 \cdot 10^{-4}$.

Как видно из табл. 7.10 и 7.13, погрешность расчета коэффициента упрочнения по сравнению с предыдущим случаем несколько увеличилась, что связано с уменьшением памяти НС за счет сокращения весов удаленных признаков, а также

удаления из обучающей выборки информации, содержавшейся в удаленных признаках.

С другой стороны, полученный результат является вполне приемлемым. Отметим также, что в последнем случае скорость обучения при фиксированном количестве циклов обучения и скорость работы НС повысились по сравнению с предыдущим случаем.

Коэффициент множественной корреляции для данных из табл. 6 составил 0.99(9), в то время, как для статистических моделей коэффициента упрочнения, полученных в предыдущих исследованиях он не превышал 0.95. Это свидетельствует о том, что полученная нейросетевая модель является более точной по сравнению со статистическими моделями.

Таблица 7.13 - Результаты нейросетевого моделирования после исключения малоинформативных факторов

| номер экземпляра | $\beta^{у}_{эксп.}$ | $\beta^{у}_{расч.}$ | номер экземпляра | $\beta^{у}_{эксп.}$ | $\beta^{у}_{расч.}$ | номер экземпляра | $\beta^{у}_{эксп.}$ | $\beta^{у}_{расч.}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1,16 | 1,1594 | 21 | 1,45 | 1,4503 | 41 | 1,58 | 1,5853 |
| 2 | 1,27 | 1,2706 | 22 | 1,5 | 1,5006 | 42 | 1,6 | 1,6004 |
| 3 | 1,38 | 1,38 | 23 | 1,61 | 1,61 | 43 | 1,56 | 1,5593 |
| 4 | 1,54 | 1,5396 | 24 | 1,64 | 1,6404 | 44 | 1,66 | 1,6572 |
| 5 | 1,46 | 1,4602 | 25 | 1,44 | 1,4398 | 45 | 1,6 | 1,6006 |
| 6 | 1,35 | 1,3499 | 26 | 1,13 | 1,1343 | 46 | 1,59 | 1,5912 |
| 7 | 1,39 | 1,3907 | 27 | 1,29 | 1,2901 | 47 | 1,55 | 1,5497 |
| 8 | 1,35 | 1,3492 | 28 | 1,29 | 1,29 | 48 | 1,56 | 1,5651 |
| 9 | 1,21 | 1,2103 | 29 | 1,32 | 1,3196 | 49 | 1,53 | 1,5251 |
| 10 | 1,51 | 1,5098 | 30 | 1,42 | 1,42 | 50 | 1,2 | 1,196 |
| 11 | 1,51 | 1,51 | 31 | 1,57 | 1,57 | 51 | 1,19 | 1,1929 |
| 12 | 1,38 | 1,38 | 32 | 1,47 | 1,47 | 52 | 1,15 | 1,1508 |
| 13 | 1,37 | 1,37 | 33 | 1,43 | 1,4283 | 53 | 1,13 | 1,1302 |
| 14 | 1,21 | 1,2094 | 34 | 1,4 | 1,4015 | 54 | 1,83 | 1,83 |
| 15 | 1,38 | 1,3805 | 35 | 1,48 | 1,4825 | 55 | 1,75 | 1,75 |
| 16 | 1,38 | 1,3805 | 36 | 1,52 | 1,5167 | 56 | 1,23 | 1,2295 |
| 17 | 1,38 | 1,3805 | 37 | 1,48 | 1,4809 | 57 | 1,22 | 1,2207 |
| 18 | 1,37 | 1,3683 | 38 | 1,49 | 1,4903 | 58 | 1,39 | 1,3919 |
| 19 | 1,33 | 1,3303 | 39 | 1,45 | 1,4499 | 59 | 1,35 | 1,3479 |
| 20 | 1,27 | 1,2704 | 40 | 1,6 | 1,5951 | | | |

Высокая точность, обеспечиваемая при моделировании коэффициента упрочнения на основе НС, позволяет рассчитывать предел выносливости деталей на стадии разработки технологического процесса. Результаты моделирования коэффициента упрочнения деталей ГТД на основе НС являются вполне приемлимыми для применения на практике.

Комбинированный анализ информативности факторов на основе НС и коэффициентов корреляции позволяет упростить и оптимизировать модель коэффициента упрочнения, а также повысить достоверность получаемых результатов.

## 7.3 Моделирование коэффициента упрочнения деталей авиадвигателей при обкатке

В задачу ранее проводившихся исследований входило получение с помощью теории подобия и анализа размерностей математической модели коэффициента упрочнения с участием параметров обкатки шариками и роликами, физико-механических характеристик материалов деталей и инструмента с учетом изменения эффективности упрочнения при наличии концентрации напряжений и масштабного фактора деталей на этапе проектирования и внедрения технологического процесса.

В качестве факторов, наиболее существенно влияющих на коэффициент упрочнения деталей при обкатке, предлагается использовать:

$x_1$ - $P_y$, Н – сила обкатывания.

$x_2$ - $R_{пр.}$, мм – профильный радиус ролика.

$x_3$ - q, МПа – среднее контактное давление, рассчитанное по формулам теории упругости.

$x_4$ - НВ, МПа – твердость материала.

$x_5$ - а, мм –полуось эллипса касания в зоне упругого контакта.

$x_6$ - $\sigma_в$, МПа – предел прочности.

$x_7$ - $\sigma_{0.2}$, МПа – предел текучести материала.

$x_8$ - n – показатель деформационного упрочнения.

$x_9$ - $\overline{G}$, мм$^{-1}$ – относительный градиент первого главного напряжения.

В предыдущих работах на основе полученных экспериментальных данных строились статистические модели коэффициента упрочнения, которые допускали среднюю погрешность при расчете $\beta^y$ свыше 10 %, что приводит к снижению точности расчета запаса прочности. Поэтому необходимо осуществить моделирование коэффициента упрочнения на основе более точной методики.

Для построения более точной модели коэффициента упрочнения в настоящей работе предлагается использовать искусственные нейронные сети (НС). В качестве наиболее часто используемой на практике модели НС выступает многослойная НС (МНС), которая способна обучаться аппроксимации многомерных функций по точечным данным.

Однако обучение МНС аппроксимации сложных существенно нелинейных зависимостей представляет собой достаточно длительный процесс, который при большом объеме многомерных экспериментальных данных может занимать продолжительное время даже на современных ЭВМ. Поэтому представляется достаточно актуальным использование методов, позволяющих увеличить скорость обучения МНС.

Для построения модели коэффициента упрочнения были статистически обработаны результаты испытаний на усталость различных образцов и деталей, изготовленных из сталей и сплавов: сталь 45, 40ХН, сталь 40, 45ХН2МФА, 2Х13, сталь 40Х, 13Х11Н2В2МФШ, 18Х2Н4ВА, ВТ3-1, 34ХН3М.

Упрочнение образцов и деталей производили обкатыванием роликами. Испытания на усталость производили при плоском изгибе и изгибе с вращением. Пределы выносливости исследуемых материалов были определены для вероятности разрушения Р=50 %. Твердость образцов и деталей находилась в пределах НВ=1100-5350 МПА, предел прочности $\sigma_B$= 410-2080 МПА, предел текучести $\sigma_{0.2}$ = 230-1900 МПа, показатель деформационного упрочнения n = 0.06-0.22, сила обкатывания $P_y$ =

200-6000 Н, подача инструмента s =0.08-0.21 мм/об., относительный градиент первого главного напряжения $\overline{G}$ = 0.20-21.9 мм$^{-1}$.

Фрагмент результатов испытаний на усталость и значения факторов представлены в табл. 7.14, здесь $\beta^{y}_{эксп.}$-значение коэффициента упрочнения, полученное экспериментально.

Моделирование коэффициента упрочнения осуществлялось с помощью двухслойного персептрона, первый слой которого содержал 5 нейронов, а второй слой – 1 нейрон. Все нейроны имели сигмоидную функцию активации $\psi(x)=1/(1+e^{-x})$. На входы НС подавались значения факторов. На выход НС подавалось значение коэффициента упрочнения для соответствующего образца. Моделирование коэффициента упрочнения осуществлялось в двух экспериментах.

В первом случае обучение МНС осуществлялось на основе алгоритма Левенберга-Марквардта. В качестве целевой функции при обучении использовался минимум среднеквадратической ошибки сети для всей выборки goal=$10^{-6}$. При обучении НС коэффициент редукции $\eta$ для алгоритма Левенберга-Марквардта полагался равным 0.9, шаг обучения 0.00001, максимальное число циклов обучения НС epochs=500. Время обучения МНС составило 197.6 с., количество итераций обучения 500, средняя погрешность полученной модели 2,6 % (для статистической модели в предыдущих работах - более 10%), максимальная погрешность 12,1 %, среднеквадратическая ошибка обучения МНС составила $4.37\cdot10^{-3}$.

Во втором случае обучение МНС осуществлялось на основе следящего алгоритма с использованием алгоритма Левенберга-Марквардта, параметры которого задавались так же, как и в предыдущем случае. Параметры следящего алгоритма задавались следующим образом: $\alpha$ = 1.0001, $\Delta t$ = 25; $\xi$ = 0.1.

Таблица 7.14 - Фрагмент результатов экспериментов по моделированию коэффициента упрочнения

| № | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $\beta^y_{эксп.}$ | $\beta^y_{расч.}$ | $\beta^{y\,*}_{расч.}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1500 | 0.0225 | 18500 | 1670 | 0.478 | 657 | 356 | 0.33 | 4.63 | 1.456 | 1.4568 | 1.4561 |
| 2 | 1500 | 0.0133 | 2690 | 2860 | 0.565 | 860 | 583 | 0.124 | 10.53 | 5.70 | 5.7006 | 5.7000 |
| 3 | 1500 | 0.0133 | 2690 | 1830 | 0.565 | 724 | 268 | 0.199 | 10.53 | 2.63 | 2.6391 | 2.6288 |
| 4 | 1500 | 0.0133 | 2690 | 1780 | 0.565 | 661 | 266 | 0.200 | 10.53 | 3.35 | 3.3417 | 3.3504 |
| 5 | 1500 | 0.0133 | 2690 | 1950 | 0.565 | 682 | 351 | 0.169 | 10.53 | 3.31 | 3.3088 | 3.3112 |
| 6 | 1500 | 0.0133 | 2690 | 3490 | 0.565 | 1170 | 1050 | 0.087 | 10.53 | 6.26 | 6.2600 | 6.2593 |
| 7 | 1500 | 0.0133 | 2690 | 1920 | 0.565 | 648 | 289 | 0.190 | 10.53 | 1.97 | 1.9699 | 1.9699 |
| 8 | 1000 | 0.6 | 5750 | 2350 | 0.327 | 750 | 600 | 0.122 | 0.3 | 1.213 | 1.1920 | 1.2187 |
| 9 | 1000 | 0.0217 | 2290 | 2350 | 0.148 | 750 | 600 | 0.122 | 4.25 | 1.61 | 1.6089 | 1.6095 |
| 10 | 65000 | 0.0852 | 23540 | 1800 | 2.15 | 650 | 380 | 0.161 | 0.261 | 1.54 | 1.5406 | 1.5400 |
| 11 | 1000 | 0.0273 | 12410 | 2070 | 0.465 | 748 | 394 | 0.157 | 3.33 | 1.91 | 1.9097 | 1.9102 |
| 12 | 850 | 0.25 | 5030 | 2770 | 0.284 | 985 | 875 | 0.097 | 0.34 | 1.08 | 1.1705 | 1.0692 |
| 13 | 1700 | 0.25 | 6343 | 2770 | 0.357 | 985 | 875 | 0.097 | 0.34 | 1.12 | 1.1746 | 1.1449 |
| 14 | 2800 | 0.25 | 7490 | 2770 | 0.422 | 985 | 875 | 0.097 | 0.34 | 1.18 | 1.1853 | 1.2197 |
| 15 | 1100 | 0.25 | 5490 | 3490 | 0.309 | 1200 | 1100 | 0.084 | 0.34 | 1.06 | 1.1811 | 1.0882 |
| 16 | 2500 | 0.25 | 7210 | 3410 | 0.406 | 1200 | 1100 | 0.084 | 0.34 | 1.12 | 1.2219 | 1.1649 |
| 17 | 3600 | 0.25 | 8140 | 3410 | 0.459 | 1200 | 1100 | 0.084 | 0.34 | 1.29 | 1.3042 | 1.2271 |
| 18 | 1100 | 0.25 | 5490 | 3750 | 0.309 | 1460 | 1310 | 0.076 | 0.34 | 1.18 | 1.1809 | 1.1847 |
| 19 | 2500 | 0.25 | 7210 | 3750 | 0.406 | 1460 | 1310 | 0.076 | 0.34 | 1.23 | 1.2284 | 1.2495 |
| 20 | 3600 | 0.25 | 8140 | 3750 | 0.459 | 1460 | 1310 | 0.076 | 0.34 | 1.29 | 1.3196 | 1.2923 |
| 21 | 250 | 0.1 | 4360 | 2550 | 0.165 | 950 | 850 | 0.126 | 0.467 | 1.11 | 1.1790 | 1.1139 |
| 22 | 350 | 0.1 | 4880 | 2550 | 0.194 | 950 | 850 | 0.126 | 0.467 | 1.093 | 1.1765 | 1.1007 |
| 23 | 1000 | 0.5 | 5070 | 3400 | 0.306 | 1150 | 1000 | 0.089 | 0.53 | 1.21 | 1.2490 | 1.2375 |
| 24 | 500 | 0.1 | 5090 | 3400 | 0.236 | 1150 | 1000 | 0.089 | 0.867 | 1.272 | 1.2880 | 1.2708 |
| 25 | 1000 | 0.1 | 5070 | 3400 | 0.306 | 1150 | 1000 | 0.089 | 8.2 | 2.13 | 2.1271 | 2.1303 |
| 26 | 1850 | 0.6 | 7100 | 3850 | 0.394 | 1434 | 1171 | 0.086 | 0.32 | 1.46 | 1.4596 | 1.4508 |
| 27 | 2000 | 0.1923 | 11850 | 1630 | 1.608 | 580 | 263 | 0.201 | 0.479 | 1.24 | 1.2429 | 1.2409 |
| 28 | 1000 | 0.05 | 13640 | 3000 | 0.245 | 1300 | 1190 | 0.08 | 2.94 | 2.0 | 2.0002 | 2.0002 |
| 29 | 3400 | 0.5333 | 5580 | 3000 | 0.776 | 1140 | 1050 | 0.087 | 0.467 | 1.31 | 1.2398 | 1.3131 |
| 30 | 3400 | 0.5333 | 5580 | 3000 | 0.776 | 1370 | 1320 | 0.076 | 0.467 | 1.15 | 1.2001 | 1.1495 |
| 31 | 1700 | 0.5333 | 4420 | 3000 | 0.616 | 1370 | 1320 | 0.76 | 0.467 | 1.08 | 1.1688 | 1.0811 |
| 32 | 1000 | 0.5 | 5460 | 2250 | 0.308 | 820 | 560 | 0.127 | 0.33 | 1.22 | 1.1722 | 1.2176 |
| 33 | 1700 | 0.25 | 6340 | 3520 | 0.357 | 1340 | 1230 | 0.079 | 0.34 | 1.19 | 1.1895 | 1.1951 |
| 34 | 2500 | 0.25 | 7210 | 3520 | 0.406 | 1340 | 1230 | 0.079 | 0.34 | 1.33 | 1.2167 | 1.2752 |
| 35 | 3600 | 0.25 | 8140 | 3520 | 0.459 | 1340 | 1230 | 0.079 | 0.34 | 1.33 | 1.2919 | 1.3227 |
| 36 | 4500 | 0.25 | 877 | 3520 | 0.494 | 1340 | 1230 | 0.079 | 0.34 | 1.33 | 1.3260 | 1.3295 |
| 37 | 2500 | 0.25 | 7210 | 4150 | 0.406 | 1530 | 1470 | 0.071 | 0.34 | 1.26 | 1.2558 | 1.2367 |
| 38 | 3600 | 0.25 | 8140 | 4150 | 0.459 | 1530 | 1470 | 0.071 | 0.34 | 1.27 | 1.2671 | 1.3190 |
| 39 | 200 | 0.125 | 6385 | 1770 | 0.191 | 624 | 375 | 0.162 | 0.34 | 1.19 | 1.1689 | 1.1752 |
| 40 | 500 | 0.125 | 10090 | 1770 | 0.302 | 624 | 375 | 0.162 | 0.34 | 1.27 | 1.1725 | 1.2535 |
| 41 | 900 | 0.125 | 13540 | 1770 | 0.405 | 624 | 375 | 0.162 | 0.34 | 1.34 | 1.3578 | 1.3492 |

За те же 500 итераций обучения, как и в предыдущем случае, и время обучения 276 с. средняя погрешность полученной модели составила 1 % (для статистической модели - более 10%), максимальная погрешность - 6,3 %, среднеквадратическая ошибка обучения МНС - $0.82 \cdot 10^{-3}$. Среднеквадратическая ошибка менее $4.37 \cdot 10^{-3}$ следящим алгоритмом была достигнута за 101.2 с. Таким образом, следящий алгоритм позволил существенно сократить время обучения МНС и увеличить скорость сходимости градиентного алгоритма для достижения тех же результатов, что и алгоритм Левенберга-Марквардта.

Результаты экспериментов представлены в табл. 7.14, здесь $\beta^{у}_{расч.}$ – расчетное значение коэффициента упрочнения, полученное с помощью НС, обученной на основе алгоритма Левенберга-Марквардта, $\beta^{у}_{расч.*}$ – расчетное значение коэффициента упрочнения, полученное с помощью НС, обученной на основе следящего алгоритма. Матрица весовых коэффициентов, полученная в результате обучения НС на основе следящего алгоритма, представлена в табл. 7.15.

Таблица 7.15 - Матрица весовых коэффициентов МНС

| номер нейрона в 1 слое | | | | | | номер нейрона во 2 слое | |
|---|---|---|---|---|---|---|---|
| номер входа нейрона | 1 | 2 | 3 | 4 | 5 | номер входа нейрона | 1 |
| 0 | 5.5188 | -2.7160 | 5.5295 | 4.5764 | 29.5621 | 0 | -12.7605 |
| 1 | -3.4531 | 33.7441 | -10.7443 | 5.838 | 8.8652 | 1 | 39.2546 |
| 2 | -8.1986 | 2.7885 | -8.6723 | -8.4272 | 9.2333 | 2 | 25.3902 |
| 3 | -10.1509 | 11.4559 | -10.5799 | -6.6921 | 30.7992 | 3 | -18.1889 |
| 4 | -15.5475 | 26.9654 | -0.6976 | 6.7071 | 9.2096 | 4 | 1.9189 |
| 5 | -14.2435 | 17.1615 | 1.2639 | 0.512 | -44.7011 | 5 | -14.9947 |
| 6 | -3.248 | 21.0626 | 1.5073 | -28.685 | -0.042 | | |
| 7 | 15.4715 | -43.8403 | -2.8917 | -3.6671 | -38.2587 | | |
| 8 | 1.5678 | -39.0633 | -10.1356 | -4.8526 | -48.464 | | |
| 9 | -0.173 | -0.0155 | -0.3095 | 4.2307 | -1.2678 | | |

Высокая точность, обеспечиваемая при моделировании коэффициента упрочнения на основе НС, позволяет рассчитывать предел выносливости деталей на стадии разработки технологического процесса. Результаты моделирования коэффициента упрочнения деталей ГТД на основе НС являются вполне приемлемыми для применения на практике.

## 7.4 Моделирование коэффициента упрочнения деталей авиадвигателей при повышенных температурах

Важным вопросом при проектировании технологического процесса упрочнения деталей ГТД является прогнозирование эффективности упрочнения и обоснование выбранных режимов с учетом условий эксплуатации.

Показателем эффективности упрочнения алмазным выглаживанием деталей ГТД, работающих при циклических нагрузках, является коэффициент упрочнения, который обычно определяется по результатам испытания на усталость стандартных образцов. Проведение испытаний на усталость деталей является дорогостоящей и не всегда осуществимой задачей.

В предыдущих исследованиях осуществлялось моделирование коэффициента упрочнения деталей после алмазного выглаживания при нормальных условиях. Однако при повышенных температурах изменяются значения физико-механических характеристик деталей, таких, как предел прочности, предел текучести, модуль упругости, удельная теплоемкость, коэффициенты линейного расширения и теплопроводности и др., оказывающих определенное влияние на сопротивление усталости.

Поэтому достаточно актуальным является моделирование коэффициента упрочнения деталей, работающих при повышенных температурах.

В качестве факторов, отражающих процесс алмазного выглаживания деталей, работающих при повышенных температурах, предлагается использовать:

$x_1$ - число проходов;

$x_2$ - радиус сферы алмазного инструмента, $R_{сф.}$, мм;

$x_3$ - твердость деталей по Бринелю, НВ, МПа;

$x_4$ - сила выглаживания, Р, Н;

$x_5$ - температура испытаний, Т, $^{o}C$;

$x_6$ - коэффициент линейного удлинения, $\alpha,1\cdot10^{6}/^{o}C$;

$x_7$ - предел прочности, определенный при повышенных температурах, $\sigma_{в}$, МПа;

$x_8$ - предел текучести, определенный при повышенных температурах, $\sigma_{0,2}$, МПа;

$x_9$ - модуль упругости, определенный при повышенных температурах, Е, ГПа;

$x_{10}$ - диаметр образца в опасном сечении, d, мм (здесь и далее d = 7.5 мм = const);

$x_{11}$ - подача при выглаживании, s, мм/об. (здесь и далее s = 0,03 мм/об. = const).

На основе экспериментально полученных данных для образцов, изготовленных из материалов: ХН77ТЮР, 13ХПН2В2М ФШ и 40ХН2МАШ осуществлялось нейросетевое моделирование коэффициента упрочнения деталей, работающих при повышенных температурах.

Исходные данные приведены в табл. 7.16.

Таблица 7.16 - Исходные данные и результаты моделирования коэффициента упрочнения деталей авиадвигателей после алмазного выглаживания, работающих при повышенных температурах

| i / j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | $\beta^{у}_{эксп.}$ | $\beta^{у}_{расч.}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 240 | 15 | 20 | 12,4 | 1000 | 700 | 215 | 1,60 | 1,60 |
| 2 | 1 | 3 | 240 | 15 | 300 | 15,1 | 950 | 650 | 191 | 1,51 | 1,51 |
| 3 | 1 | 3 | 240 | 15 | 600 | 17,8 | 900 | 600 | 161,5 | 1,40 | 1,40 |
| 4 | 1 | 3 | 240 | 15 | 780 | 19,4 | 550 | 420 | 135 | 1,21 | 1,21 |
| 5 | 1 | 3 | 260 | 20 | 20 | 11 | 1150 | 1000 | 200 | 1,30 | 1,30 |
| 6 | 1 | 3 | 260 | 20 | 200 | 12 | 1120 | 970 | 182 | 1,14 | 1,14 |
| 7 | 1 | 3 | 260 | 20 | 400 | 13,3 | 1020 | 900 | 165 | 1,27 | 1,27 |
| 8 | 1 | 2,5 | 305 | 15 | 20 | 12,8 | 1090 | 970 | 195 | 1,20 | 1,20 |
| 9 | 1 | 2,5 | 305 | 15 | 200 | 14 | 1030 | 850 | 180 | 1,15 | 1,15 |
| 10 | 1 | 2,5 | 305 | 15 | 400 | 14,8 | 970 | 790 | 168 | 1,26 | 1,26 |
| 11 | 2 | 2,5 | 305 | 15 | 20 | 12,8 | 1090 | 970 | 195 | 1,26 | 1,26 |
| 12 | 2 | 2,5 | 305 | 15 | 200 | 14 | 1030 | 850 | 180 | 1,25 | 1,25 |
| 13 | 2 | 2,5 | 305 | 15 | 400 | 14,8 | 970 | 790 | 168 | 1,28 | 1,28 |
| $e_i$ | 1 | 2,5 | 240 | 15 | 20 | 11 | 550 | 420 | 135 | 1,14 | |
| $f_i$ | 2 | 3,0 | 305 | 20 | 780 | 19,4 | 1150 | 1000 | 215 | 1,60 | |
| $a_i$ | 0 | 0 | 0 | 0 | 0,022 | 0,044 | 0,044 | 0,044 | 0,044 | | |
| $b_i$ | 0,413 | 0,717 | 0,636 | 0,413 | 4,296 | 21,0 | 21,52 | 4,763 | 13,48 | | |
| $c_i$ | 0,188 | 0,214 | 0,231 | 0,165 | 1,111 | 5,268 | 5,965 | 1,652 | 2,957 | | |
| $z_i$ | 0,45 | 0,38 | 0,58 | 0,1 | 0,53 | 1,0 | 0,83 | 0,41 | 0,79 | | |
| $p_i$ | + | - | + | - | + | + | + | - | + | | |

Здесь i - номер фактора, j - номер образца, $\beta^{y}_{эксп.}$ - экспериментально полученное значение коэффициента упрочения деталей после алмазного выглаживания, работающих при повышенных температурах, $\beta^{y}_{расч}$ - расчетное значение коэффициента упрочения деталей после алмазного выглаживания, работающих при повышенных температурах.

Моделирование коэффициента упрочнения осуществлялось в два этапа на основе двухслойного персептрона, содержавшего в первом слое 2 нейрона, а во втором слое 1 нейрон. Все нейроны первого слоя имели сигмоидную функцию активации: $\Psi(a)=\frac{1}{1+e^{-a}}$, нейрон второго слоя имел линейную функцию активации: $\Psi(a)=a$.

На первом этапе на вход персептрона подавались нормированные значения признаков $x_1$-$x_9$, на выход персептрона подавалось нормированное значение $\beta^{y}_{эксп.}$. Факторы $x_{10}$ и $x_{11}$ при моделировании не учитывались, поскольку они являются константами.

Значения факторов нормировались по формуле:

$$x^{j}_{i,норм.}=\frac{x^{j}_{i}-Min(x_i)}{Max(x_i)-Min(x_i)},$$

где $x^{j}_{i}$ - значение i-го фактора j-го образца, $x^{j}_{i,норм.}$ - нормированное значение i-го фактора j-го образца, $Min(x_i)$ и $Max(x_i)$ - минимальное и максимальное значения i-го признака для всех образцов, соответственно.

Значения $\beta^{y}_{эксп.}$ нормировались по формуле:

$$\beta^{y(j)}_{эксп.,норм.}=\frac{\beta^{y(j)}_{эксп.}-Min(\beta^{y}_{эксп.})}{Max(\beta^{y}_{эксп.})-Min(\beta^{y}_{эксп.})},$$

где $в^{y(j)}_{эксп.}$ - значение $\beta^{y}_{эксп.}$ для j-го образца, $\beta^{y(j)}_{эксп.,норм.}$ - нормированное значение $\beta^{y}_{эксп.}$ для j-го образца, $Min(\beta^{y}_{эксп.})$ и $Max(\beta^{y}_{эксп.})$ - минимальное и максимальное значения $\beta^{y}_{эксп.}$ для всех образцов, соответственно.

Минимальные и максимальные значения факторов и коэффициента упрочнения приведены в таблице, где обозначены $e_i$ и $f_i$, соответственно.

Обучение персептрона производилось в разных экспериментах на основе алгоритма Левенберга-Марквардта, а также на основе следящего алгоритма, имплементированного в алгоритм Левенберга-Марквардта.

Среднее время обучения персептрона в разных экспериментах для алгоритма Левенберга-Марквардта составило 38,3 с., а для следящего алгоритма - 28,6 с. Средняя среднеквадратическая ошибка для алгоритма Левенберга-Марквардта составила $4{,}33 \cdot 10^{-4}$, а для следящего алгоритма - $2{,}8 \cdot 10^{-4}$. Наименьшая среднеквадратическая ошибка для алгоритма Левенберга-Марквардта составила $1{,}8 \cdot 10^{-9}$, а для следящего алгоритма - $9{,}9 \cdot 10^{-11}$. Таким образом, следящий алгоритм в среднем на 25% работает быстрее, достигая при этом на 35% большей точности, чем алгоритм Левенберга-Марквардта.

После получения нейросетевой модели коэффициента упрочнения осуществлялись оценка информативности и отбор наиболее информативных факторов.

Для этого была оценена относительная нейросетевая информативность $z_i$ для каждого i-го фактора, определяемая на основе весов персептрона, обученного аппроксимации $\beta^{y}_{эксп.}$ по набору значений факторов $x_i$, i=1,2,..,N, где N-количество факторов (в данном случае N=9).

Кроме того были рассчитаны $a_i = \mathrm{Min}\left(\left|\Delta\beta^{y}_{эксп.,норм.} / \Delta x_{i,норм.}\right|\right)$, $b_i = \mathrm{Max}\left(\left|\Delta\beta^{y}_{эксп.,норм.} / \Delta x_{i,норм.}\right|\right)$ и $c_i = \mathrm{Mid}\left(\left|\Delta\beta^{y}_{эксп.,норм.} / \Delta x_{i,норм.}\right|\right)$ - минимальная, максимальная и средняя скорости изменения нормированного $в^{y}_{эксп.}$ при изменении нормированного $x_i$, соответственно, при условии, что значения нормированного $x_i$ упорядочены по возрастанию, а значения нормированного $\beta^{y}_{эксп.}$ упорядочены в соответствии с отсортированным нормированным $x_i$. Эти скорости предлагается использовать в качестве характеристик информативности факторов: чем больше скорости, тем сильнее $\beta^{y}_{эксп.,норм.}$

реагирует на изменение значений соответствующего фактора, тем более сильно влияние фактора на $\beta^{y}_{эксп.,норм.}$ и, следовательно, на $\beta^{y}_{эксп.}$.

На основе $a_i$, $b_i$, $c_i$ и $z_i$ принимались $p_i$ - решения об уровне значимости (информативности) i-го фактора: "+" - информативный (значимый), "-" - малоинформативный (малозначимый).

После принятия решений о значимости факторов малоинформативные факторы были исключены из обучающего множества и моделирование осуществлялось уже для 6 факторов: $x_1$, $x_3$, $x_5$, $x_6$, $x_7$ и $x_9$.

Обучение персептрона осуществлялось на основе следящего алгоритма. В результате обучения персептрона была получена нейросетевая модель коэффициента упрочнения, которая в разнормированном виде может быть представлена в следующей форме:

$$в^{y}_{расч.} = 1{,}14 + 0{,}46(2{,}1035 - 1{,}0511Ш(1{,}7992 - 0{,}584x^{j}_{1,норм.} - 3{,}0818x^{j}_{3,норм.} -$$

$$- 5{,}3577x^{j}_{5,норм.} + 2{,}8570x^{j}_{6,норм.} - 11{,}1865x^{j}_{7,норм.} + 18{,}8947x^{j}_{9,норм.}) -$$

$$- 1{,}8387Ш(2{,}2165 - 0{,}2304x^{j}_{1,норм.} + 2{,}6034x^{j}_{3,норм.} + 6{,}7875x^{j}_{5,норм.} -$$

$$- 7{,}0878x^{j}_{6,норм.} + 0{,}5920x^{j}_{7,норм.} - 5{,}0090x^{j}_{9,норм.})), \text{ где } \Psi(a) = \frac{1}{1+e^{-a}}.$$

Среднеквадратическая ошибка для полученной модели составила $3{,}9 \cdot 10^{-11}$, время обучения - 6,4 с.

Полученная модель может быть использована для расчета коэффициента упрочнения деталей, изготовленных из материалов ХН77ТЮР, 13ХПН2В2М ФШ и 40ХН2МАШ, характеризующихся следующими свойствам: число проходов 1-2, радиус сферы алмазного инструмента $R_{сф.}$= 2,5 - 3 мм, твердость деталей по Бринелю НВ = 240 - 305 МПа, сила выглаживания Р = 15 - 20 Н, температура испытаний Т = 20 - 780 $^{\circ}C$, коэффициент линейного удлинения $\alpha = 11 - 19{,}4\frac{1 \cdot 10^{6}}{^{\circ}C}$, предел прочности, определеннный при повышенных температурах, $\sigma_в$ = 550 - 1150 МПа, предел текучести, определеннный при повышенных температурах, $\sigma_{0,2}$= 420 - 1000 МПа, модуль упругости, определенный при

повышенных температурах, E = 135 - 215 ГПа, диаметр образца в опасном сечении d = 7.5 мм, подача при выглаживании, s = 0,03 мм/об.

Для сравнения полученных результатов с результатами ранее проводившегося статистического моделирования, был рассчитан коэффициент множественной корреляции, который составил 0,9(9), в то время, как для статистических моделей наибольший коэффициент множественной корреляции составляет 0,947. Погрешность для лучшей статистической модели составляет 7,9 %, в то время, как полученная нейросетевая модель обладает практически нулевой погрешностью для обучающих данных.

Для рассчетов по статистическим моделям для каждой детали необходимо измерять значения всех 11 факторов, в то время, как для полученной нейросетевой модели за счет исключения малозначимых факторов необходимо иметь значения всего 6 факторов.

Таким образом на основании выше изложенного можно заключить, что предложенная в настоящей работе нейросетевая модель является более точной и менее требовательной к измерительным ресурсам по сравнению со статистическими моделями.

Использование следящего алгоритма для обучения нейронных сетей моделированию многомерных зависимостей является оправданным и позволяет сократить время построения моделей и быстрее достигать требуемой точности по сравнению с алгоритмом Левенберга-Марквардта.

Отбор информативных признаков позволяет упростить модель, снизить требования к ресурсам ЭВМ и уменьшить затраты на измерения за счет исключения неинформативных факторов.

## 7.5 Моделирование коэффициента упрочнения деталей авиадвигателей шариками в ультразвуковом поле

На ряду с вышеописанными методами упрочениня для поверхностной обработки деталей применяют упрочение шариками в ультразвуковом поле. В

данном случае также актуальна задача моделирования коэффициента упрочнения деталей авиадвигателей.

Для обоснования физических факторов, определяющих величину коэффициента упрочнения шариками в ультразвуковом поле, были проведены исследования, анализ результатов которых позволил остановиться на следующих факторах, оказывающих наиболее существенное влияние на коэффициент упрочнения деталей авиадвигателей шариками в ультразвуковом поле $\beta^{y}$:

$x_1$ - $\overline{G}$, мм$^{-1}$ – относительный градиент первого главного напряжения.

$x_2$ - $\sigma_{0.2}$, МПа – предел текучести материала.

$x_3$ - НВ, МПа – твердость материала по Бринелю.

$x_4$ - $v$ - скорость соударения шарика с упрочняемым объектом, м/с;

$x_5$ - D - диаметр шариков, мм;

$x_6$ - t - время упрочнения, с;

$x_7$ - M - масса шариков, кг (далее M = const = 0,4 кг);

$x_8$ - $\sigma_{B}$, МПа - предел прочности (далее $\sigma_{B}$ = const = 1100 МПа).

$x_9$ - V - объем камеры для крепления упрочняемых деталей, м$^3$ (далее V = const = 0,2м$^3$).

В предыдущих работах для построения модели коэффициента ультразвукового упрочнения по вышеперечисленным факторам была построена статистическая модель по выборке, содержавшей 40 экземпляров, соответствовавших различным образцам и лопаткам. Данная выборка из-за ограниченного объема использовалась как в процессе построения модели, так и для проверки модели, что не позволило достаточно полно оценить адекватность полученной модели.

Поэтому задачами экспериментальных исследований в настоящей работе были: разбиение исходной выборки на обучающую и тестовую, а также построение высокоточной модели коэффициента упрочнения.

Достаточно эффективным средством для построения моделей сложных объектов и процессов по выборке небольшого объема являются искусственные НС, обладающие высокими адаптивными и аппроксимационными способностями. В качестве базовой модели НС в настоящей работе предлагается использовать радиально-базисную НС, которая обладает относительно несложной архитектурой и способна обучаться быстрее, чем ряд других моделей НС.

Для проверки адекватности полученной модели предлагается разделить исходную выборку на обучающую и контрольную выборки таким образом, чтобы в обучающей выборке присутствовали только те экземпляры, которые находятся на границах областей сосредоточения экземпляров в метрическом пространстве, а остальные экземпляры (избыточные примеры) были отнесены к контрольной выборке. Для реализации подобного разделения экземпляров предлагается использовать алгоритм планирования обучающего эксперимента.

Для построения НС-модели коэффициента упрочнения использовались результаты испытаний на усталость образцов из сплава ВТ8, стали ЭП718 и лопаток из сплава ВТ8, которые приведены в табл. 7.17.

Моделирование коэффициента упрочнения осуществлялось с помощью радиально-базисной НС, первый слой которой содержал 38 нейронов, а второй слой – 1 нейрон. Все нейроны первого слоя имели радиально-симметричную функцию активации, нейрон последнего слоя имел линейную функцию активации $\psi(x)=x$.

Для оценки эффективности метода планирования обучающего эксперимента проводились 2 опыта.

В первом случае в качестве обучающей и тестовой выборок использовалась вся исходная выборка экземпляров. Матрица весовых коэффициентов, полученная в результате обучения НС, представлена в табл. 7.18. Результаты нейросетевого моделирования коэффициента упрочнения приведены в табл. 7.17. Здесь $\beta^{у}_{эксп.}$ - значение коэффициента упрочнения, полученное экспериментально, $\beta^{у}_{расч.}$ – расчетное значение коэффициента упрочнения, полученное с помощью радиально-базисной НС. Средняя погрешность полученной модели составила 2,1 % (в

предыдущих исследованиях - 3%), максимальная погрешность составила 9.3 %. Это подтверждает обоснованность применения НС для моделирования коэффициента упрочнения.

Во втором случае перед обучением НС осуществлялось разбиение исходной выборки на обучающую и тестовую. В результате такого разбиения к обучающей выборке были отнесены 20 экземпляров, а к тестовой 18 экземпляров. На основе полученной обучающей выборки была построена нейросетевая модель коэффициента упрочнения, весовые коэффициенты которой приведены в табл. 7.19.

Результаты нейросетевого моделирования коэффициента упрочнения для обучающей и тестовой выборок представлены в табл. 7.17. Здесь $\beta^{y}_{расч.*}$ – расчетное значение коэффициента упрочнения, полученное с помощью радиально-базисной НС, обученной на основе только обучающей выборки.

Таблица 7.17 - Результаты моделирования коэффициента упрочнения

| № | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $\beta^{y}_{эксп.}$ | $\beta^{y}_{расч}$ | $\beta^{y}_{расч}{}^{*}$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.9 | 700 | 3250 | 28 | 2.35 | 300 | 1.15 | 1.17 | 1.15 |
| 2 | 0.9 | 700 | 3250 | 28 | 2.35 | 300 | 1.18 | 1.17 | 1.15 |
| 3 | 1.43 | 700 | 3250 | 19 | 1.3 | 300 | 1.19 | 1.19 | 1.11 |
| 4 | 1.43 | 700 | 3250 | 19 | 1.3 | 300 | 1.19 | 1.19 | 1.11 |
| 5 | 1.43 | 700 | 3250 | 19 | 1.3 | 300 | 1.19 | 1.19 | 1.11 |
| 6 | 0.9 | 700 | 3250 | 19 | 1.3 | 300 | 1.09 | 1.12 | 1.1 |
| 7 | 0.9 | 700 | 3250 | 19 | 1.3 | 300 | 1.12 | 1.12 | 1.11 |
| 8 | 0.9 | 700 | 3250 | 19 | 1.3 | 300 | 1.15 | 1.12 | 1.11 |
| 9 | 0.66 | 700 | 3250 | 28 | 2.35 | 300 | 1.16 | 1.16 | 1.18 |
| 10 | 0.66 | 700 | 3250 | 28 | 2.35 | 300 | 1.16 | 1.16 | 1.18 |
| 11 | 1.43 | 700 | 3250 | 28 | 2.35 | 300 | 1.16 | 1.11 | 1.06 |
| 12 | 1.43 | 700 | 3250 | 28 | 2.35 | 300 | 1.06 | 1.11 | 1.06 |
| 13 | 1.43 | 950 | 3600 | 19 | 1.3 | 300 | 1 | 1.1 | 1.1 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 14 | 1.43 | 950 | 3600 | 19 | 1.3 | 300 | 1.05 | 1.1 | 1.1 |
| 15 | 1.43 | 950 | 3600 | 19 | 1.3 | 300 | 1.07 | 1.1 | 1.1 |
| 16 | 1.43 | 950 | 3600 | 19 | 1.3 | 300 | 1.12 | 1.1 | 1.1 |
| 17 | 1.43 | 950 | 3600 | 19 | 1.3 | 300 | 1.15 | 1.1 | 1.1 |
| 18 | 1.43 | 950 | 3600 | 19 | 1.3 | 300 | 1.17 | 1.1 | 1.1 |
| 19 | 1.0 | 950 | 3600 | 28 | 2.35 | 300 | 0.93 | 0.94 | 0.93 |
| 20 | 1.0 | 950 | 3600 | 28 | 2.35 | 300 | 0.96 | 0.94 | 0.93 |
| 21 | 1.0 | 950 | 3600 | 28 | 2.35 | 300 | 0.93 | 0.94 | 0.93 |
| 22 | 1.0 | 950 | 3600 | 19 | 1.3 | 300 | 1.07 | 1.08 | 1.08 |
| 23 | 1.0 | 950 | 3600 | 19 | 1.3 | 300 | 1.09 | 1.08 | 1.08 |
| 24 | 1.0 | 950 | 3600 | 19 | 1.3 | 300 | 1.09 | 1.08 | 1.08 |
| 25 | 0.66 | 950 | 3600 | 28 | 2.35 | 300 | 0.93 | 0.96 | 0.93 |
| 26 | 0.66 | 950 | 3600 | 28 | 2.35 | 300 | 0.93 | 0.96 | 0.93 |
| 27 | 0.66 | 950 | 3600 | 28 | 2.35 | 300 | 0.98 | 0.96 | 0.93 |
| 28 | 0.66 | 950 | 3600 | 28 | 2.35 | 300 | 0.98 | 0.96 | 0.93 |
| 29 | 0.66 | 950 | 3600 | 19 | 1.3 | 300 | 1.03 | 1.04 | 1.08 |
| 30 | 0.66 | 950 | 3600 | 19 | 1.3 | 300 | 1.05 | 1.04 | 1.08 |
| 31 | 0.66 | 950 | 3600 | 19 | 1.3 | 300 | 1.08 | 1.04 | 1.08 |
| 32 | 0.66 | 950 | 3600 | 19 | 1.3 | 300 | 0.98 | 1.04 | 1.08 |
| 33 | 3.05 | 950 | 3600 | 19 | 1.3 | 300 | 1.23 | 1.23 | 1.23 |
| 34 | 3.05 | 950 | 3600 | 28 | 2.35 | 300 | 1.13 | 1.13 | 1.13 |
| 35 | 3.05 | 950 | 3600 | 19 | 1.3 | 600 | 1.12 | 1.12 | 1.12 |
| 36 | 3.05 | 950 | 3600 | 28 | 2.35 | 600 | 1.11 | 1.11 | 1.11 |
| 37 | 3.05 | 950 | 3600 | 19 | 1.3 | 900 | 1.22 | 1.22 | 1.22 |
| 38 | 3.05 | 950 | 3600 | 28 | 2.35 | 900 | 1.04 | 1.04 | 1.04 |

Таблица 7.18 - Матрица весовых коэффициентов НС, обученной на всей исходной выборке экземпляров

| номер слоя | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | 2 | |
| номер нейрона в 1 слое | номер входа нейрона | | | | | | | номер входа нейрона | веса входов нейрона |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 0 | 0.9197 |
| 1 | 0.8326 | 1.4 | 700 | 3250 | 19 | 1.3 | 300 | 1 | 0.3269 |
| 2 | 0.8326 | 0.9 | 700 | 3250 | 28 | 2.4 | 300 | 2 | 0.2749 |
| 3 | 0.8326 | 1.0 | 950 | 3600 | 28 | 2.4 | 300 | 3 | -0.1362 |
| 4 | 0.8326 | 3.0 | 950 | 3600 | 19 | 1.3 | 900 | 4 | 0.3003 |
| 5 | 0.8326 | 3.0 | 950 | 3600 | 19 | 1.3 | 600 | 5 | 0.2003 |
| 6 | 0.8326 | 3.0 | 950 | 3600 | 28 | 2.4 | 600 | 6 | 0.1903 |
| 7 | 0.8326 | 3.0 | 950 | 3600 | 28 | 2.4 | 300 | 7 | 0.2147 |
| 8 | 0.8326 | 3.0 | 950 | 3600 | 28 | 2.4 | 900 | 8 | 0.1203 |
| 9 | 0.8326 | 3.0 | 950 | 3600 | 19 | 1.3 | 300 | 9 | 0.3342 |
| 10 | 0.8326 | 0.7 | 950 | 3600 | 19 | 1.3 | 300 | 10 | -0.6205 |
| 11 | 0.8326 | 0.9 | 700 | 3250 | 19 | 1.3 | 300 | 11 | -0.0688 |
| 12 | 0.8326 | 0.7 | 950 | 3600 | 28 | 2.4 | 300 | 12 | 0.1569 |
| 13 | 0.8326 | 1.4 | 700 | 3250 | 28 | 2.4 | 300 | 13 | -0.0360 |
| 14 | 0.8326 | 1.0 | 950 | 3600 | 19 | 1.3 | 300 | 14 | 1.1111 |
| 15 | 0.8326 | 1.4 | 950 | 3600 | 19 | 1.3 | 300 | 15 | -0.4466 |

Таблица 7.19 - Матрица весовых коэффициентов НС, обученной на основе обучающей выборки

| номер слоя | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | 2 | |
| номер нейрона в 1 слое | номер входа нейрона | | | | | | | номер входа нейрона | веса входов нейрона |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 0 | 1.1051 |
| 1 | 0.8326 | 1.0 | 950 | 3600 | 28 | 2.4 | 300 | 1 | -0.0990 |
| 2 | 0.8326 | 0.9 | 700 | 3250 | 28 | 2.4 | 300 | 2 | 0.2543 |
| 3 | 0.8326 | 3.0 | 950 | 3600 | 19 | 1.3 | 900 | 3 | 0.1149 |
| 4 | 0.8326 | 3.0 | 950 | 3600 | 19 | 1.3 | 600 | 4 | 0.0149 |
| 5 | 0.8326 | 3.0 | 950 | 3600 | 28 | 2.4 | 300 | 5 | 0.0319 |
| 6 | 0.8326 | 3.0 | 950 | 3600 | 28 | 2.4 | 900 | 6 | -0.0651 |
| 7 | 0.8326 | 3.0 | 950 | 3600 | 19 | 1.3 | 300 | 7 | 0.1256 |
| 8 | 0.8326 | 0.7 | 950 | 3600 | 28 | 2.4 | 300 | 8 | -0.0844 |
| 9 | 0.8326 | 1.4 | 700 | 3250 | 28 | 2.4 | 300 | 9 | -0.2544 |
| 10 | 0.8326 | 1.0 | 950 | 3600 | 19 | 1.3 | 300 | 10 | -0.0351 |
| 11 | 0.8326 | 1.4 | 950 | 3600 | 19 | 1.3 | 300 | 11 | 0.0074 |

Средняя погрешность полученной модели составила 3 %, как и в предыдущих исследованиях, максимальная погрешность составила 10%. Снижение точности по сравнению с предыдущим случаем объясняется тем, что в последнем случае использовалась лишь часть исходной выборки, содержавшая меньше информации. В то же время полученный результат не хуже результатов ранее проводившихся исследований, но является более объективным по сравнению с ними, т.к. его адекватность проверяется с помощью экземпляров, не использовавшихся при обучении.

Анализируя результаты, полученные в настоящей работе, можно отметить, что относительно невысокая точность модели коэффициента упрочнения объясняется

тем, что при обучении использовались экземпляры с одинаковыми значениями факторов, но с существенно разными значениями $\beta^y$ (например, экземпляры 1,2 и 6-8 в табл. 7.15).

Результаты, полученные в настоящей работе, позволяют рекомендовать применение радиально-базисных НС в задачах аппроксимации сложных зависимостей.

# ЗАКЛЮЧЕНИЕ

Методы и алгоритмы обработки экспериментальных данных, отбора информативных признаков и распознавания образов, рассмотренные в данной книге, составляют теоретический базис для разработки программного обеспечения в области математического моделирования и его приложений в задачах диагностики и прогнозирования надежности изделий.

На ряду с хорошо известными классическими методами и алгоритмами в книге приведено описание оригинальных авторских разработок, которые свободны от ряда недостатков и сложностей применения, присущих классическим методам.

Описание современных программных средств диагностики и прогнозирования позволяет прикладным специалистам обучиться работе с ними и эффективно использовать их при решении практических задач.

Примеры решения практических задач диагностики и прогнозирования надежности авиадвигателей доказывают эффективность и целесообразность применения рассмотренных методов, а также служат своеобразным примером методлологии применения рассмотренных методов и алгоритмов на практике.

Обобщая материал, изложенные в настоящей книге, можно заключть, что применение методов вычислительной математики, основанных на использовании технологий искусственного интеллекта, при построении диагностических моделей сложных объектов и процессов является необходимым условием для обеспечения и повышения их надежности, устойчивости и работоспособности.

# ЛИТЕРАТУРА


1. Boseniuk T., van der Meer M., Poschel T. A Multiprocessor system for high speed simulation of neural networks // Journal of New Generation Computer Systems. – 1990. - № 3. - P. 65-71.
2. Dubrovin V., Morshchavka S., Piza D., Subbotin S. Plant recognition by genetic algorithm based back-propagation // Proceedings, Remote Sensing 2000: from spectroscopy to remotely sensed spectra. Soil Science Society of America, Bouyocos Conference, Corpus Christi, Texas, October 22-25, 2000.-P. 47- 54.
3. Dubrovin V., Subbotin S. Choice of neuron transfer functions and research of their influence for learning quality of neural networks // Proceedings of International Conference on Modern Problem of Telecommunications, Computer Science and Engineers Training TCSET’2000, February 14-19, 2000, Lviv-Slavsko, pp. 114-115.
4. Dubrovin V., Subbotin S. Model of Magnetic Heads Audio Characteristics // The Experience of Designing and Application of CAD Systems in Microelectronics: Proceedings of the VI International Conference CADSM 2001.-Lviv: Publishing House of LPNU, 2001.-P. 232-233.
5. Dubrovin V., Subbotin S. The Quick Method of Neural Network Training // Proceedings of International Conference on Modern Problems of Telecommunications, Computer Science and Engineers Training TCSET'2002.-Lviv-Slavsko: NU"Lvivska Politechnica", pp.266-267.
6. Dubrovin V., Subbotin S., Morshchavka S., Piza D. The plant recognition on remote sensing results by the feed-forward neural networks // Smart Engineering System Design, 2001, № 3, P. 251-256.
7. Dubrovin V.I., Subbotin S.A., Morshchavka S.V., Piza D.M. The plant recognition on remote sensing results by the feed-forward neural networks // Smart Engeneering Systems Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems, ANNIE 2000: the 10-th Anniversary edition / ed. C. H. Dagli et al.-Missouri-Rolla:ASME Press, 2000, vol.10, P. 697-701.

8. Dubrovin V.I., Subbotin S.A., Morshchavka S.V., Piza D.M., Adamenko V.A. Object recognition by hybrid neural network classifier // Труды Международной научно-практической конференции "Знание - Диалог - Решение" -KDS-2001.-СПб.:С-ЗГЗТУ - "Лань", 2001.-С. 194-200.
9. Dubrovin V.I., Subbotin S.A., Yatzenko V.K. Neural network model of hardening coefficient of airengine details // Smart Engeineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems / ed.: C.H. Dagli, et al.- New York: ASME press, 2001, vol. 11, P. 939-944.
10. Kohonen T. Self-organization and associative memory. - Berlin: Springer, 1984.- 255 p.
11. Kohonen T., Kangas J., Laaksonen J., Torkkola K. LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms // Proceedings of the International Joint Conference on Neural Networks. - Baltimore: IEEE, June 1992.- vol I. - P. 725-730
12. LVQ_PAK: The Learning Vector Quantization Program Package / Kohonen T., Hynninen J., Kangas J. and others. - Helsinki: Helsinki University of Technology, 1995.-30 p.
13. Neural Network Toolbox User Guide / Beale M., Demuth H. - Natick: Mathworks, 1997. - 700 p.
14. Абу-Мустафа Я.С., Псалтис Д. Оптические нейронно-сетевые компьютеры // В мире науки, 1987. N 5. С. 42-50.
15. Аведьян Э.Д. Алгоритмы настройки многослойных нейронных сетей // Автоматика и телемеханика. – 1995. - № 4. - С. 106-118
16. Адаменко В.А., Басов Ю.Ф., Дубровин В.И., Субботин С.А. Нейросетевая обработка сигналов в задачах диагностики газотурбинных авиадвигателей // Цифровая обработка сигналов и ее применение: 3-я Международная конференция и выставка.-М.:РНТОРЭС им. А.С. Попова, 2000.-С. 40-45.
17. Адаменко В.А., Дубровин В.И., Жеманюк П.Д., Субботин С.А. Диагностика лопаток авиадвигателей по спектрам свободных затухающих колебаний после ударного возбуждения // Автоматика-2000. Міжнародна конференція з

автоматичного управління, Львів, 11-15 вересня 2000: Праці у 7-ми томах.-Т. 5.- Львів: Державний НДІ інформаційної інфраструктури, 2000.- С. 7-13.

18. Адаменко В.А., Дубровин В.И., Жеманюк П.Д., Субботин С.А. Диагностика усталостных трещин в деталях газотурбинных авиадвигателей // Надійність машин та прогнозування їх ресурсу / Доповіді міжнародної науково-технічної конференції.-В 2-х томах. Том 1.-Івано-Франківськ: ІФДТУНГ-Факел, 2000.- С. 151 – 158.
19. Адаменко В.А., Дубровин В.И., Субботин С.А. Диагностика лопаток авиадвигателей по спектрам затухающих колебаний после ударного возбуждения на основе нейронных сетей прямого распространения // Нові матеріали і технології в металургії та машинобудуванні, 2000, № 1, С. 91-96.
20. Адаменко В.А., Дубровин В.И., Субботин С.А. Нейросетевая диагностика деталей энергетических установок, работающих при циклических нагрузках // Новые технологии, методы обработки и упрочнения деталей энергетических установок: Тез. докл. Международной конференции “Новые технологии, методы обработки и упрочнения деталей энергетических установок” / Отв. ред. В.К. Яценко.-Запорожье: ЗГТУ, 2000.-С. 4-6.
21. Айвазян С.А., Бежаева З.И., Староверов О.В. Классификация многомерных наблюдений.- М.: Статистика, 1974.- 240 с.
22. Айзерман М.А., Браверман Э.М., Розоноэр Л.И. Метод потенциальных функций в теории обучения машин. М.: Наука, 1970.- 383 с.
23. Алгоритмы и программы восстановления зависимостей / Под ред. В.Н. Вапника.- М.: Наука, 1984.-816 с.
24. Андерсон Т. Введение в многомерный статистический анализ.- М.: Физматгиз, 1963. 500 с.
25. Анисимов Б.В., Курганов В.Д., Злобин В.К. Распознавание и цифровая обработка изображений: Учеб. пособие для студентов вузов.-М.: Высшая школа, 1983.-295 с.
26. Аркадьев А.Г., Браверман Э.М. Обучение машины классификации объектов.- М.: Наука, 1971.- 172 с.

27. Афифи А., Эйзен С. Статистический анализ: Подход с использованием ЭВМ.: Пер. с англ. – М.: Мир, 1982. – 488 с.
28. Ачасова С.М. Вычисления на нейронных сетях // Программирование. – 1991. - № 2. - С. 40-52.
29. Банди Б. Методы оптимизации. Вводный курс: Пер с англ..-М.: Радио и связь, 1988.-128 с.
30. Биргер И.А. Техническая диагностика.-М.: Машиностроение, 1978.- 240 с.
31. Бовель Е.И., Паршин В.В. Нейронные сети в системах автоматического распознавания речи // Зарубежная радиоэлектроника. - 1998. - №4. - С. 50-57.
32. Богуслаев А.В., Дубровин В.И., Субботин С.А., Яценко В.К. Моделирование коэффициента ультразвукового упрочнения деталей авиадвигателей // Нові матеріали і технології в металургії та машинобудуванні, 2001, №2, С. 87-90.
33. Богуслаев А.В., Дубровин В.И., Субботин С.А., Яценко В.К. Модель коэффициента упрочнения деталей ГТД // Технологические системы, 2001, № 3.- С.42-45.
34. Богуслаев В.А., Яценко В.К., Притченко В.Ф. Технологическое обеспечение и прогнозирование несущей способности деталей ГТД.-К.:Издательская фирма «Манускрипт», 1993.-332 с.
35. Браверман Э.М., Мучник И.Б. Структурные методы обработки эмпирирических данных. - М.: Наука. Главная редакция физико-математической литературы. 1983. - 464 с.
36. Вапник В.Н. Задача обучения распознаванию образов.-М.: Знание, 1971.-60 с.
37. Вапник В.Н., Червоненкис А.Ф. Теория распознавания образов. - М.: Наука, 1974.
38. Васильев В.И. Проблема обучения распознаванию образов.-К.: Выща школа, 1989.-64 с.
39. Васильев В.И. Распознающие системы (Справочник).- К.:Наукова думка, 1983.- 422 с.
40. Внуков Ю.Н., Дубровин В.И. Алгоритм классификации с использованием дискриминантных функций. // Высокие технологии в машиностроении. Сборник научных трудов ХГПУ.-Харьков, 1998, С.64-66.

41. Внуков Ю.Н., Дубровин В.И. Методики прогнозирования с использованием теории статистических оценок и статистической классификации // Высокие технологии в машиностроении: диагностика процессов и обеспечение качества / Материалы VI международного научно-технического семинара.- Харьков: ХГПУ, 1996, С.26-27.
42. Галушкин А.И. Синтез многослойных систем распознавания образов. -М.: Энергия, 1974.-368 с.
43. Гаскаров Д.В., Голинкевич Т.А., Мозгалевский А.В. Прогнозирование технического состояния и надежности радиоэлектронной аппаратуры./ Под ред. Т.А. Голинкевича-М.: Советское радио, 1974.- 234 с.
44. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. М.: Мир,1985.-509 с.
45. Горбань А.Н. Нейрокомпьютер, или аналоговый ренессанс // Мир ПК, 1994, № 10.- С. 126-130.
46. Горбань А.Н. Обучение нейронных сетей. М.: СП "ParaGraph", 1990.- 160 с.
47. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере.- Новосибирск: Наука, 1996.- 276 с.
48. Горелик А.Л., Скрипкин В.А. Методы распознавания: Учеб. пособие для вузов.- 3-е изд. перераб. и доп.-М.: Высшая школа, 1989.-232 с.
49. Дубровин В.И. Идентификация и оптимизация сложных технических процессов и объектов.-Запорожье: ЗГТУ, 1997.- 92 с.
50. Дубровин В.И. Эвристические алгоритмы классификации // Машиностроитель, 1998, №7, С.6–9.
51. Дубровин В.И., Емчицкий В.Т., Онацко А.Г. Автоматизированная система факторного анализа. // Електротехніка та електроенергетика, 1999, № 2, С. 62-67.
52. Дубровин В.И., Корецкий Н.Х. Методика оптимальной классификации // Методологические проблемы качества обучения и обучения качеству / Материалы научно-методической конеренции.- Харьков: ХАИ, 1997, С.34-35.
53. Дубровин В.И., Морщавка С.В., Пиза Д. М., Субботин С.А. Нейросетевая идентификация объектов по спектрам // Труды международной конференции

“Идентификация систем и задачи управления” SICPRO’ 2000.-М.: ИПУ РАН, 2000.-С. 1190-1204 (CD-ROM).

54. Дубровин В.И., Морщавка С.В., Пиза Д. М., Субботин С.А. Применение радиально-базисных нейронных сетей для обработки данных дистанционного зондирования растений // Цифровая обработка сигналов и ее применение: 3-я Международная конференция и выставка.-М.:РНТОРЭС им. А.С. Попова, 2000.-С.48-53.
55. Дубровин В.И., Морщавка С.В., Пиза Д. М., Субботин С.А. Распознавание растений по результатам дистанционного зондирования на основе многослойных нейронных сетей // Математичні машини і системи, 2000, № 2-3, С. 113-119.
56. Дубровин В.И., Субботин С.А. Алгоритм классификации с оценкой значимости признаков // Радіоелектроніка. Інформатика. Управління, 2001, № 2, С. 145-150.
57. Дубровин В.И., Субботин С.А. Алгоритм многомерной классификации и его нейросетевая интерпретация // Радіоелектроніка. Інформатика. Управління, 2000, № 2, С. 49 –54.
58. Дубровин В.И., Субботин С.А. Алгоритм настройки весов трехслойного перцептрона // Труды VII Всероссийской конференции “Нейрокомпьютеры и их применение” НКП-2001 с международным участием , Москва, 14-16 февраля, 2001 г.- М.: ИПУ РАН, С. 552-555.
59. Дубровин В.И., Субботин С.А. Алгоритм нейросетевого отбора признаков // Матеріали міжнародної конференції з автоматичного управління "Автоматика-2001", 10-14 вересня 2001 р.-Одеса: ОДПУ, 2001.-Т.2, С. 88-89 .
60. Дубровин В.И., Субботин С.А. Алгоритм ускорения процесса обучения нейронных сетей // Научно-технический калейдоскоп. Серия "Приборостроение, радиотехника и информационные технологии" / Под. ред. Л.И. Волгина. - Ульяновск: Научно-производственный журнал, 2001, № 2.-С. 49-55.
61. Дубровин В.И., Субботин С.А. Алгоритм ускоренного обучения нейросетей // Нейроинформатика и ее приложения / Материалы IX Всероссийского семинара, 5-7 октября 2001 г. / Под ред. А.Н.Горбаня. Отв. за выпуск Г.М.Цибульский.-Красноярск:КГТУ, 2001.-С. 63-64.

62. Дубровин В.И., Субботин С.А. Алгоритм ускоренного обучения персептронов // Сборник научных трудов 4-й Всероссийской научно-технической конференции "Нейроинформатика-2002".-М.:МИФИ, 2002.-Ч. 2.-С.106-112.
63. Дубровин В.И., Субботин С.А. Алгоритмы редукции набора признаков для диагностики и прогнозирования // Проектирование и технология электронных средств, 2002, № 2, С. 19-23.
64. Дубровин В.И., Субботин С.А. Выбор информативных признаков при диагностике лопаток ГТД // Новые технологии, методы обработки и упрочнения деталей энергетических установок: Тез. докл. Международной конференции “Новые технологии, методы обработки и упрочнения деталей энергетических установок / Отв. ред. В.К. Яценко.-Запорожье: ЗГТУ, 2000.-С.25-27.
65. Дубровин В.И., Субботин С.А. Диагностика на основе эвристических алгоритмов в условиях ограниченного объема обучающей выборки // Proceedings of International conference “Soft computing and measurement” SCM-2000, 27-30 June 2000.-Saint-Petersburg: Saint-Petersburg State Electrotechnical University (LETI), 2000.-CD-ROM
66. Дубровин В.И., Субботин С.А. Диагностика состояния технических процессов и объектов на основе нейросетевого квантования обучающих векторов // Проектирование и технология электронных средств, 2001, № 4, С. 20-27.
67. Дубровин В.И., Субботин С.А. Индивидуальное прогнозирование надежности изделий электронной техники на основе нейронных сетей // Труды VII Всероссийской конференции “Нейрокомпьютеры и их применение” НКП-2001 с международным участием , Москва, 14-16 февраля, 2001 г.- М.: ИПУ РАН, С. 228-231.
68. Дубровин В.И., Субботин С.А. Интегрированные многоклассификаторные нейросетевые системы диагностики // Електротехніка та електроенергетика, 2001, № 1, С. 38-43.
69. Дубровин В.И., Субботин С.А. Когнитивный анализ и отбор информативных признаков при решении задач неразрушающей диагностики лопаток ГТД // Нові матеріали і технології в металургії та машинобудуванні, 2000, №2, С. 91-97.

70. Дубровин В.И., Субботин С.А. Комбинированный метод классификации // Електротехніка та електроенергетика, 2000, № 2, С. 55-59.
71. Дубровин В.И., Субботин С.А. Комбинированный метод классификации // Реляторные, непрерывнологические и нейронные сети и модели: Труды международной конференции "Континуальные логико-алгебраические исчисления и нейроматематика в науке, технике и экономике". - Ульяновск: УлГТУ, 2001. Том 2.-С. 94-96.
72. Дубровин В.И., Субботин С.А. Методы повышения эффективности процедур нейросетевой диагностики // Нейрокомпьютеры: разработка, применение, 2002, № 3, С. 3-9.
73. Дубровин В.И., Субботин С.А. Неитеративный алгоритм обучения двухслойного персептрона // Труды VIII Всероссийской конференции "Нейрокомпьютеры и их применение" НКП-2002 с международным участием. Москва, 21-22 марта 2002 г. / Под редакцией проф. А.И. Галушкина. М.: Институт проблем управления им. В.А. Трапезникова РАН, 2002.-С. 964-971.-CD-ROM (ISBN 5-201-14935-9).
74. Дубровин В.И., Субботин С.А. Нейронная сеть LVQ в задачах технической диагностики // Вычислительная техника и новые информационные технологии: Межвуз. науч. сб.-Уфа:УГАТУ, 2001, вып. 4.-С. 49-57.
75. Дубровин В.И., Субботин С.А. Нейросетевая диагностика в управлении качеством // Управление в технических системах – XXI век: сборник научных трудов III Международной научно-технической конференции.-Ковров: КГТА, 2000.-С. 136-138.
76. Дубровин В.И., Субботин С.А. Нейросетевая диагностика газотурбинных лопаток // Оптические, радиоволновые и тепловые методы и средства контроля качества материалов, промышленных изделий и окружающей среды / Тезисы докладов VIII международной научно-технической конференции.-Ульяновск: УлГТУ, 2000.-С.121-124.
77. Дубровин В.И., Субботин С.А. Нейросетевая диагностика лопаток энергетических установок // Датчики и преобразователи информации систем измерения, контроля и управления / Сборник материалов XII научно-

технической конференции с участием зарубежных специалистов. Под ред. проф. В.Н. Азарова. М.: МГИЭМ, 2000.-С. 240-242.

78. Дубровин В.И., Субботин С.А. Нейросетевая интерпретация алгоритма многомерной классификации // Труды III Всероссийской научно-технической конференции "Нейроинформатика-2001".-М.: МИФИ, 2001.-Ч.1 С.38-46.
79. Дубровин В.И., Субботин С.А. Нейросетевая оценка информативности и отбор признаков // Реляторные, непрерывнологические и нейронные сети и модели: Труды международной конференции "Континуальные логико-алгебраические исчисления и нейроматематика в науке, технике и экономике". - Ульяновск: УлГТУ, 2001. Том 2.- С. 91-93.
80. Дубровин В.И., Субботин С.А. Нейросетевая подсистема диагностического программного комплекса // Нейрокомпьютеры: разработка и применение, 2001, №2, С. 55-62.
81. Дубровин В.И., Субботин С.А. Нейросетевое моделирование и оценка параметров нелинейных регрессий // Нейрокомпьютеры и их применение / Сборник докладов 6-ой Всероссийской конференции, Москва 16-18 февраля 2000.-М.:Издательское предприятие журнала “Радиотехника”, 2000.- С. 118-120.
82. Дубровин В.И., Субботин С.А. Обобщенный градиентный алгоритм обучения многослойных нейронных сетей //Електротехніка та електроенергетика, 2000, № 1, С. 17-22.
83. Дубровин В.И., Субботин С.А. Онлайновые методы управления качеством: гибридная диагностика на основе нейронных сетей // Радіоелектроніка. Інформатика. Управління, 2001, № 1, С. 158 –163.
84. Дубровин В.И., Субботин С.А. Оценка значимости признаков с фиксацией значений // Нейронные сети и модели в прикладных задачах науки и техники: Труды международной конференции КЛИН-2002.- Ульяновск: УлГТУ, 2002. Т.3.- С.101-102.
85. Дубровин В.И., Субботин С.А. Подсистема нейросетевой диагностики // Нейроинформатика и ее приложения: Материалы VIII Всероссийского семинара

6-8 октября 2000 года, Красноярск / Под общей ред. А.Н. Горбаня; Отв. за вып. Г.М. Цыбульский.-Красноярск:ИПЦ КГТУ, 2000.- С. 63-64.

86. Дубровин В.И., Субботин С.А. Построение систем диагностики на основе карт самоорганизации Кохонена // Нейрокомпьютеры и их применение / Сборник докладов 6-ой Всероссийской конференции, Москва 16-18 февраля 2000.-М.:Издательское предприятие журнала “Радиотехника”, 2000, С. 464-467.
87. Дубровин В.И., Субботин С.А. Прогнозирование отказов деталй ГТД в процессе эксплуатации // Моделирование неравновесных систем-2000: Материалы III Всероссийского семинара / Под .ред. А.Н. Горбаня.-Красноярск: ИПЦ КГТУ, 2000.-С. 84.
88. Дубровин В.И., Субботин С.А. Программный комплекс нейросетевой диагностики // Программные продукты и системы, 2000, № 3, С. 21-23.
89. Дубровин В.И., Субботин С.А. Следящий алгоритм обучения нейронных сетей // Вимірювальна та обчислювальна техніка в технологічних процесах: Збірник наукових праць.-Хмельницький: ТУП, 2001.-С. 88-91.
90. Дубровин В.И., Субботин С.А. Эвристический алгоритм классификации и его нейросетевая интерпретация //Радіоелектроніка. Інформатика. Управління, 2000, № 1, С. 72-76.
91. Дубровин В.И., Субботин С.А., Адаменко В.А., Басов Ю.Ф. Диагностика авиадвигателей на основе нейросетевого гибридного классификатора // Труды Международной конференции "Параллельные вычисления и задачи управления" (PACO'2001).-М.: ИПУ РАН, 2001.-Т.5, С. 53-73 (CD-ROM, ISBN 5-201-09559-3).
92. Дубровин В.И., Субботин С.А., Адаменко В.А., Басов Ю.Ф. Построение гибридных систем диагностики деталей энергетических установок на основе нейронных сетей // Modelling and Analysis of Safety, Risk and Quality in Computer Systems / Proceedings of the International Scientific School MA SRQ - 2001.-СПб.: ООО НПО "Омега", 2001.-С. 236-239.
93. Дубровин В.И., Субботин С.А., Кривенко В.И., Евченко Л.Н. Сокращение объема данных в задачах распознавания и дигностики // Труды VIII Всероссийской конференции "Нейрокомпьютеры и их применение" НКП-2002 с международным

участием. Москва, 21-22 марта 2002 г. / Под редакцией проф. А.И. Галушкина. М.: Институт проблем управления им. В.А. Трапезникова РАН, 2002.- С. 954-963.-CD-ROM (ISBN 5-201-14935-9).

94. Дубровин В.И., Субботин С.А., Согорин А.А. Радиально-базисные нейронные сети в задачах технической диагностики // Інтернет, освіта, наука, друга міжнарожна конференція ІОН-2000, 10-12 жовтня. Збірник матеріалів конференції.-Вінниця:Універсум-Вінниця, 2000.-С. 303-306.
95. Дубровин В.И., Субботин С.А., Яценко В.К. Методика оценки коэффициента упрочнения деталей газотурбинных авиадвигателей // Техническая диагностика и неразрушающий контроль, 2001, № 3.-С. 42-45.
96. Дубровин В.И., Субботин С.А., Яценко В.К. Нейросетевая методика расчета коэффициента упрочнения деталей авиадвигателей // Техника машиностроения, 2001, № 4, С. 46-52.
97. Дубровин В.И., Субботин С.А., Яценко В.К. Нейросетевые технологии в задачах моделировния коэффициента упрочнения деталей авиадвигателей // Труды VIII Всероссийской конференции "Нейрокомпьютеры и их применение" НКП-2002 с международным участием. Москва, 21-22 марта 2002 г. / Под редакцией проф. А.И. Галушкина. М.: Институт проблем управления им. В.А. Трапезникова РАН, 2002.-С.572-591.-CD-ROM (ISBN 5-201-14935-9).
98. Дубровин В.И., Субботин С.А., Яценко В.К. Построение нейросетевой модели коэффициента упрочнения при обкатке деталей энергетических установок // Електротехніка та електроенергетика, 2001, № 2, С. 38-42.
99. Дубровин В.И., Субботин С.А., Яценко В.К. Прогнозирование запаса прочности деталей авиадвигателей // Моделирование неравновесных систем - 2001 / Материалы IV Всероссийского семинара - Красноярск: ИВМ СО РАН, 2001.-С. 44-45.
100. Дубровин В.И., Субботин С.А.. Построение адаптивных систем классификации на основе нейронных сетей с латеральным торможением // Радіоелектроніка. Інформатика. Управління, 1999, №2, С. 110-114.

101. Дубровін В.І., Субботін С.О. Вибір функцій активації формального нейрона та дослідження їх впливу на якість навчання нейронних мереж // Вісник Національного університету “Львівська політехніка” “Комп’ютерні системи проектування. Теорія і практика”, № 398, 2000.-С.12-17.

102. Дуда Р., Харт П. Распознавание образов и анализ сцен.М.: Мир, 1976.- 512 с.

103. Загоруйко Н.Г. Методы распознавания и их применение.-М.: Советское радио, 1972.-208 с.

104. Загоруйко Н.Г., Елкина В.Н., Лбов Г.С. Алгоритмы обнаружения эмпирических закономерностей.- Новосибирск: Наука, 1985.- 110 с.

105. Заявка № 2001118028 Україна, Спосіб побудови і навчання нейронної мережі з латеральним гальмуванням / Басов Ю.Ф., Дубровін В.І., Піза Д.М., Субботін С.О. Заявлено 23.11.2001 р.

106. Заявка № 2001118029 Україна, Спосіб настроювання вагових коефіцієнтів двошарового персептрона для рішення задач розпізнавання образів і діагностики / Внуков Ю.М., Дубровін В.І., Жеманюк П.Д., Субботін С.О. Заявлено 23.11.2001 р.

107. Заявка № 2001129102 Україна, Спосіб прискореного навчання багатошарових нейронних мереж / Богуслаєв В.О., Дубровін В.І., Субботін С.О. Заявлено 27.12.2001 р.

108. Заявка № 2002010338 Україна, Спосіб навчання шестишарового персептрона класифікації та діагностиці виробів / Богуслаєв О.В., Дубровін В.І., Субботін С.О., Яценко В.К. Заявлено 14.01.2002 р.

109. Заявка № 2002107024 Украіна, Спосіб настроювання вагових коефіцієнтів тришарового персептрона для рішення задач розпізнавання образів і діагностики / Дубровін В.І., Лук'янов В.С., Субботін С.О. Заявлено 16.10.2001р.

110. Ивахненко А.Г. Персептрон - система распознавания образов.-К.: Наукова думка, 1975. - 431 с.

111. Ивахненко А. Г. Персептроны. - Киев: Наукова думка, 1974.

112. Ивахненко А.Г. Самообучающиеся системы распознавания и автоматического регулирования.- Киев: Техника, 1969.- 392 с.

113. Ивахненко А.Г., Мюллер И. Самоорганизация прогнозирующих моделей, Берлин,1985

114. Итоги науки и техники. Сер. "Физ. и Матем. модели нейронных сетей" /Под ред. А.А.Веденова. - М.: Изд-во ВИНИТИ, 1990-92 - Т. 1-5.

115. Корнеев В.В. Параллельные вычислительные системы .-М.: Нолидж, 1999.-320 с.

116. Круг Г.К., Кабанов В.А., Фомин Г.А., Фомина Е.С. Планирование эксперимента в задачах нелинейного оценивания и распознавания образов.-М.: Наука, 1981.-172 с.

117. Кузин Л.Т. Основы кибернетики: В 2-х тт. Учеб. пособие для вузов. – М: Энергия, 1976.

118. Минский М., Пейперт С. Персептроны. М.: Мир, 1971, 261 с.

119. Миркин Б.Г. Анализ качественных признаков и структур.-М.: Статистика, 1980.- 319 с.

120. Мкртчян С.О. Проектирование логических устройств ЭВМ на нейронных элементах. - М.: Энергия, 1977.

121. Мктрчян С.О. Нейроны и нейроподобные сети, М.,1971

122. Мостеллер Ф., Тьюки Дж. Анализ данных и регрессия.- М.: Финансы и статистика, 1982.- 239 с.

123. Нейроинтеллект: от нейрона к компьютеру / Соколов Е.Н., Вайткявичюс Г.Г.- М., Наука,1989.-238с.

124. Нейроинформатика / А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес, А.Ю.Новоходько, Д.А.Россиев, С.А.Терехов, М.Ю.Сенашова, В.Г.Царегородцев. Новосибирск: Наука, Сибирская издательская фирма РАН, 1998.- 296 с.

125. Нейрокомпьютеры и интеллектуальные роботы / Амосов Н.М., Байдык Т.Н., Гольцев А.Д. и др.; под ред. Амосова Н.М. - Киев: Наукова думка, 1991. - 272 с.

126. Основы технической кибернетики. Учеб. пособие для вузов.-М.: Высшая школа, 1970.- 464 с.

127. Патент № 44662А Україна, Спосіб розрахунку коефіцієнта зміцнення деталей після алмазного вигладжування / Богуслаєв О.В., Дубровін В.І., Субботін С.О., Яценко В.К. Заявлено 16.10.2001 р. Опубл. 15.02.2002 р., бюл. № 2 "Промислова власність".

128. Распознавание образов: состояние и перспективы: Пер. с англ./ К.Верхаген, Р. Дейн, Ф. Грун и др.-М.: Радио и связь, 1985.-104 с.

129. Резник А.М. Итеративный проекционный алгоритм обучения нейронных сетей // Кибернетика и системный анализ. – 1993. - №6. - С.131-141.

130. Резник А.М., Городничий Д.О., Сычев А.С. Регулирование локальной обратной связи в нейронных сетях с проекционным алгоритмом обучения // Кибернетика и системный анализ. – 1996. - №6. - С. 153-162.

131. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике: В 2-х кн. Пер с англ.– М.:Мир, 1986.-Кн. 1: 349 с., Кн. 2: 320 с.

132. Розенблат Ф. Принципы нейродинамики. Перцептрон и теория механизмов мозга. М.: Мир, 1965. 480 с.

133. Субботін С.О. Нейронні мережі керують якістю // Пульсар, 1999, № 12, С. 8-10.

134. Субботин С.А. Алгоритм планирования он-лайнового эксперимента в нейросетевой диагностике // Нейроинформатика и ее приложения / Материалы IX Всероссийского семинара, 5-7 октября 2001 г. / Под ред. А.Н.Горбаня. Отв. за выпуск Г.М.Цибульский.- Красноярск:КГТУ, 2001.-С.180-181.

135. Субботин С.А. Нейрокибернетика в СССР-СНГ: аналитический обзор изобретений и патентов // Сборник научных трудов 4-й Всероссийской научно-технической конференции "Нейроинформатика - 2002".-М.:МИФИ, 2002.-Ч. 1.-С.48-54.

136. Терехов В.А., Ефимов Д.В., Тюкин И.Ю., Антонов В.Н. Нейросетевые системы управления.-СПб.: Изд-во С.-Петербургского ун-та, 1999.- 265 с.

137. Трикоз Д.В. Нейронные сети: как это делается?// Компьютеры + программы - 1993 - N 4(5) - с. 14-20.

138. Тэнк Д.У., Хопфилд Д.Д. Коллективные вычисления в нейроноподобных электронных схемах.//В мире науки. 1988. N 2. С.44-53.

139. Уоссермен Ф. Нейрокомпьютерная техника.- М.: Мир, 1992.

140. Фукунга К. Введение в статистическую теорию распознавания образов.- М.: Наука, 1979.- 367 с.

141. Химмельблау Д. Прикладное нелинейное программирование. М.: Мир, 1975.- 534 с.

142. Хинтон Дж. Е. Как обучаются нейронные сети// В мире науки, 1992, № 11-12, С. 103-107

143. Хинтон Дж.Е. Обучение в параллельных сетях // Реальность и прогнозы искусственного интеллекта.- М.: Мир, 1987.- С. 124-136.

144. Цыпкин Я.З. Основы теории обучающихся систем. М.: Наука, 1970. 252 с.

# ПРИЛОЖЕНИЕ. ПУТЕВОДИТЕЛЬ ПО СПИСКУ ЛИТЕРАТУРЫ

| тема | номер источника |
|---|---|
| Теория технической диагностики | 30, 34, 39, 43, 49 |
| Обработка экспериментальных данных | 23, 24, 27, 35, 43, 49, 51, 81, 104, 116, 122 |
| Сокращение размерности диагностической информации | 30, 43, 56, 59, 63, 64, 69, 79, 84, 93, 119, 134 |
| Распознавание образов | 21-23, 25-27, 30, 36- 43, 48-50, 52, 56-58, 70, 71, 90, 102-104, 112, 113, 116, 117, 126, 128, 140, 144 |
| Нейроинформатика | 1-20, 28, 29, 31-33, 42, 44-47, 53-58, 60-62, 65-68, 72-78, 81-83, 86, 89-92, 94-101, 105-111, 114, 115, 118, 120, 121, 123-125, 129-133, 135-139, 141-143 |
| Программные средства технической диагностики | 11, 12, 13, 23, 80, 85, 88 |
| Прикладные задачи диагностики | 2, 4, 6, 7, 9, 16-20, 32-34, 43, 49, 53-55, 64, 67, 69, 76, 77, 87, 91, 91, 95-99, 127 |

Часть работ, приведенных в списке литературы, размещена в сети Интернет на веб-сайтах: http://csit.narod.ru/people/Subbotin.htm и http://www.zntu.edu.ua/RIC.

Valeriy Dubrovin, Sergey Subbotin,
Alexander Boguslayev and Viktor Yatzenko

# INTELLIGENT MEANS OF DIAGNOSTICS AND PREDICTION OF RELIABILITY OF AIRENGINES

## SUMMARY

The monography contains the review of modern methods of a construction of diagnostic models. The large attention in the book is given to application of methods of an artificial intelligence to problem solving of diagnostics and reliability prediction.

The book contains the description of experiments under the solution by the authors of practical problems of diagnostics and reliability prediction of airengines.

The introduction contains the substantation of a urgency and purposes of the book.

**Chapter 1. The primal problems and principles of technical diagnostics** contains definition and purpose of diagnostics, description of a structure, problems, process and stages of technical diagnostics and prediction.

**Chapter 2. The preprocessing of the experimental data** contains the description of methods of smoothing, normalization, scaling and quantization of signals.

**Chapter 3. The reduction of dimensionality of the diagnostic information** contains the description statistical, probabilistic, heuristic, information, neural network and cognitive methods of informative features selection, and also a method of a partition of initial set tolearning and test sets.

**Chapter 4. The methods and algorithms of a construction of diagnostic models** contains the description of statistical, probabilistic and heuristic methods of the a pattern recognition theory.

**Chapter 5. Neural network methods of diagnostics and prediction** contains a general characteristic and description of basic models and methods of training of neural networks.

**Chapter 6. The software of diagnostics and prediction** contains the description of an automated system "Diagnostica", hardware-software complex POS "Vojag", integrated system of diagnostics and description of a package Matlab.

**Chapter 7. The experimental researches on diagnostics and reliability prediction of airengines** contains the description and results of experiments on problem solving of diagnostics of airengine blades, modelling of a hardening coefficient of details of airengines at a diamond smoothness and at a rolling, modelling of a hardening coefficient of details of airengines at the high temperatures, and also modelling of coefficient of hardening of airengines details by balls in a ultrasonic field.

**Conclusion** contains main conclusions and recommendations for application of intelligentl means of diagnostics.