

Міністерство освіти і науки України
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних і самостійних робіт
з дисципліни
«СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ»
для студентів напрямів підготовки
6.050103 “Програмна інженерія” та
6.050101 “Комп’ютерні науки”
(усіх форм навчання)

Методичні вказівки до виконання лабораторних і самостійних робіт з дисципліни "Системи штучного інтелекту" для студентів напрямів підготовки 6.050103 “Програмна інженерія” та 6.050101 “Комп’ютерні науки” (усіх форм навчання) / Уклад.: С.О. Субботін, Федорченко Є.М. – Запоріжжя: ЗНТУ, 2010. – 30 с.

Автори: Сергій Олександрович Субботін, к.т.н.,
доцент кафедри програмних засобів
Євген Миколайович Федорченко, асистент

Рецензент: А.О. Олійник, кандидат технічних наук

Відповідальний
за випуск: А.В. Притула, зав. каф. програмних засобів

Затверджено
на засіданні кафедри
програмних засобів

Протокол № 10
від 15.06.2010 р.

ЗМІСТ

Загальні положення.	4
1. Лабораторна робота № 1. Розпізнавання образів на основі метричної класифікації.	5
2. Лабораторна робота № 2. Методи відбору ознак для побудови розпізнаючих моделей.	8
3. Лабораторна робота № 3. Розпізнавання на основі нейронних мереж	10
4. Самостійна робота № 1. Програмні засоби для побудови інтелектуальних систем	13
5. Самостійна робота № 2. Самоорганізація та навчання без учителя. Кластер-аналіз	14
6. Самостійна робота № 3. Гібридні нейро-нечіткі системи.	16
Література.	18
Додаток А. Моделювання нейронних мереж у середовищі MATLAB	23
Додаток Б. Кластер-аналіз у середовищі MATLAB.	33

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Дане видання призначене для вивчення та практичного освоєння студентами усіх форм навчання основ створення систем штучного інтелекту.

Відповідно до графіка студенти перед виконанням лабораторної або самостійної роботи повинні ознайомитися з конспектом лекцій та рекомендованою літературою.

Для одержання заліку по кожній роботі студент здає викладачу цілком оформлений звіт, а також 3,5-дюймову дискету у форматі MS – DOS / Windows, перевірену на відсутність вірусів, з текстами розроблених програм, файлами програм, що виконуються, файлами даних і текстом звіту.

Звіт (приклад оформлення - див. додаток А) має містити:

- титульний аркуш (на ньому вказують назву міністерства, назву університету, назву кафедри, номер, вид і тему роботи, виконавця та особу, що приймає звіт, рік);
- мету, варіант і завдання роботи;
- лаконічний опис теоретичних відомостей;
- текст програми, що обов'язково містить коментарі;
- вхідні та вихідні дані програми;
- змістовний аналіз отриманих результатів та висновки.

Звіт виконують на білому папері формату А4 (210 x 297 мм). Текст розміщують тільки з однієї сторони листа. Поля сторінки з усіх боків – 20 мм. Аркуші скріплюють за допомогою канцелярських скріпок. Для набору тексту звіту використовують редактор MS Word 97: шрифт Times New Roman, 12 пунктів. Міжрядковий інтервал: полуторний – для тексту звіту, одинарний – для листингів програм, таблиць і роздруківок даних.

Під час співбесіди студент повинний виявити знання про мету роботи, по теоретичному матеріалу, про методи виконання кожного етапу роботи, по змісту основних розділів розробленого звіту з демонстрацією результатів на конкретних прикладах. Студент повинний вміти правильно аналізувати отримані результати. Для самоперевірки при підготовці до виконання і здачі роботи студент повинний відповісти на контрольні питання, приведені наприкінці опису відповідної роботи. Загальний залік студент одержує після виконання і здачі останньої роботи.

1 Лабораторна робота № 1

РОЗПІЗНАВАННЯ ОБРАЗІВ НА ОСНОВІ МЕТРИЧНОЇ КЛАСИФІКАЦІЇ

Мета роботи: вивчити та засвоїти на практиці метричні методи розпізнавання образів у просторі ознак, навчитися створювати програмні засоби, що реалізують методи метричної класифікації.

Завдання до роботи

1. Ознайомитися з конспектом лекцій та рекомендованою літературою. На алгоритмічній мові програмування пакету MATLAB написати програму, що реалізує процедури для навчання та емуляції за методом еталонів.

2. Згідно з номером студента за журналом для відповідного номера варіанта V сформувати навчаючу вибірку $\langle x, y \rangle$ обсягом S екземплярів x^s , $s=1, 2, \dots, S$, що характеризуються N ознаками x_j^s , $j=1, 2, \dots, N$, та зіставити кожному екземпляру значення цільової ознаки y^s :

$$x_j^s = \begin{cases} jV - 0,1s, & j = 1, 5, 9, \dots, \\ 0,01jV^{-1} + 0,3s, & j = 2, 4, 6, \dots, \\ j\text{rand}, & j = 3, 7, 11, \dots; \end{cases} \quad y^s = \begin{cases} 0, (x_1^s)^2 + (x_2^s)^2 < V^2 + 0,04S^2, \\ 1, (x_1^s)^2 + (x_2^s)^2 \geq V^2 + 0,04S^2; \end{cases}$$

$$S = \begin{cases} 10V, & V < 10, \\ 5V, & 10 \leq V < 20, \\ 3V, & V \geq 20; \end{cases} \quad N = \begin{cases} 5V, & V < 7, \\ 4V, & 7 \leq V < 10, \\ 3V, & 10 \leq V < 20, \\ 2V, & V \geq 20; \end{cases}$$

де rand - випадкове число в діапазоні $[0, 1]$.

3. Виконати нормування навчаючої вибірки даних.

4. На основі пронормованої вибірки побудувати розпізнаючу модель за методом еталонів, тобто визначити координати центрів класів (еталонів) у просторі ознак C_j^q , де q - номер класу, j - номер ознаки. Зафіксувати час навчання.

5. На основі побудованої моделі для екземплярів навчаючої вибірки виконати розпізнавання, тобто визначити розрахункові номери класів y^{s*} . Зафіксувати час розпізнавання.

6. Обчислити помилку розпізнавання, визначити ймовірність прийняття правильного рішення та ймовірність прийняття помилкового рішення для побудованої моделі.

7. У попередньо сформованій вибірці залишити тільки одну ознаку, номер якої у попередній вибірці дорівнює V , а також у центрів еталонів класів залишити тільки V -ту координату C^q_V .

8. Для нової вибірки та нових еталонів виконати розпізнавання, тобто визначити розрахункові номери класів y^{s*} . Зафіксувати час розпізнавання.

9. Обчислити помилку розпізнавання, визначити ймовірність прийняття правильного рішення та ймовірність прийняття помилкового рішення для нової моделі.

10. Порівняти результати проведених експериментів, зробити висновки щодо впливу параметрів навчаючої вибірки на характеристики процесів навчання та розпізнавання.

Зміст звіту

1. Мета роботи.

2. Короткі теоретичні відомості до роботи. У звіті не слід наводити докладне викладення теоретичного матеріалу, необхідно виділити лише найголовніші ідеї, формули, що необхідні для пояснення суті методу, моделі тощо.

3. Текст розробленої програми.

4. Сформована навчаюча вибірка $\langle x, y \rangle$.

5. Побудована модель - координати еталонів (центрів зосередження екземплярів) класів. Час навчання.

6. Результати розпізнавання: розраховані значення номеру класу для екземплярів y^s , помилка розпізнавання, імовірності прийняття правильного та помилкового рішень, час розпізнавання.

7. Навчаюча вибірка із однією ознакою.

8. Еталони класів з однією ознакою.

9. Результати розпізнавання скороченої вибірки за скороченими еталонами: розраховані значення номеру класу для екземплярів y^{s*} , помилка розпізнавання, імовірності прийняття правильного та помилкового рішень, час розпізнавання.

10. Висновки. У висновках треба проаналізувати результати роботи, а також лаконічно відповісти на контрольні питання.

Контрольні питання

1. Задача розпізнавання образів.
2. Основні поняття теорії розпізнавання образів.
3. Розбиття вихідної вибірки на навчаючу та тестову.
4. Навчання з учителем.
5. Методи метричної класифікації.
6. Навчання без учителя. Кластер-аналіз.
7. Лінійна роздільність і лінійна нерозділеність класів.
8. помилка навчання / класифікації, час навчання / класифікації, цільова функція навчання.
9. Чи впливає кількість використаних ознак на швидкість навчання перцептрона? Відповідь обґрунтуйте теоретично та доведіть експериментально.
10. Що таке генеральна сукупність, вибірка, екземпляр, ознака?
11. Вимоги до навчаючих вибірок даних.
12. Що таке репрезентативна вибірка даних?
13. Чи повинна навчальна вибірка бути репрезентативною?
14. Чи повинна тестова вибірка бути репрезентативною?
15. Чи впливає обсяг навчаючої вибірки на швидкість навчання?
16. Чи впливає репрезентативність навчаючої вибірки на точність класифікації екземплярів тестової вибірки?
17. Чи впливає репрезентативність тестової вибірки на точність класифікації екземплярів тестової вибірки?
18. Чи впливає репрезентативність тестової вибірки на точність навчання перцептрона за навчаючої вибіркою?
19. Чи залежить якість навчання від якості та обсягу навчаючої вибірки?

2 Лабораторна робота № 2

МЕТОДИ ВІДБОРУ ОЗНАК ДЛЯ ПОБУДОВИ РОЗПІЗНАЮЧИХ МОДЕЛЕЙ

Мета роботи: вивчити та засвоїти на практиці методи оцінювання інформативності та відбору ознак, для побудови розпізнаючих моделей.

Завдання до роботи

1. Ознайомитися з конспектом лекцій та рекомендованою літературою.

2. На алгоритмічній мові програмування пакету MATLAB написати програму, що реалізує методи оцінювання інформативності ознак: на основі модуля коефіцієнта парної кореляції, коефіцієнта кореляції знаків, коефіцієнта кореляції Фехнера, кількості інтервалів зміни номера класу, інформаційного підходу, статистичного підходу.

3. Згідно з номером студента за журналом для відповідного номера варіанта V сформувати навчаючу вибірку $\langle x, y \rangle$ за формулами, наведеними у завданні попередньої роботи. Також для екземплярів вибірки визначити значення другої цільової ознаки y_2^s :

$$y_2^s = 2x_1^s + 0,1x_2^s, s=1,2,\dots,S.$$

4. На основі сформованої вибірки по відношенню до дискретного виходу y^s оцінити інформативність ознак екземплярів вибірки за допомогою коефіцієнта кореляції знаків, коефіцієнта кореляції Фехнера, кількості інтервалів зміни номера класу, інформаційного підходу, статистичного підходу.

5. На основі сформованої вибірки по відношенню до дійсного виходу y_2^s оцінити інформативність ознак екземплярів вибірки за допомогою на основі модуля коефіцієнта парної кореляції, кількості інтервалів зміни номера класу (для цього попередньо дискретизувати y_2^s).

6. Побудувати таблицю з оцінками інформативності ознак відносно дискретного y^s , стовпці якої повинні мати назви: номер ознаки, коефіцієнт кореляції знаків, коефіцієнт кореляції Фехнера, оцінка за

кількістю інтервалів зміни номера класу, оцінка за інформаційним підходом, оцінка на основі статистичного підходу.

7. Побудувати таблицю з оцінками інформативності ознак відносно дійсного y^s , стовпці якої повинні мати назви: номер ознаки, модуль коефіцієнта парної кореляції, оцінка за кількістю інтервалів зміни номера класу

8. Проаналізувати за побудованими таблицями оцінки інформативності ознак. Зробити висновки щодо важливості ознак окремо для y^s та y^s_2 .

Зміст звіту

1. Мета роботи.
2. Короткі теоретичні відомості до роботи.
3. Текст розробленої програми.
4. Сформована навчаюча вибірка $\langle x, y, y_2 \rangle$.
5. Таблиці з оцінками інформативності ознак по відношенню до y^s та y^s_2 .
6. Рішення щодо важливості ознак окремо для y^s та y^s_2 .
7. Висновки. У висновках треба проаналізувати результати роботи, а також лаконічно відповісти на контрольні питання.

Контрольні питання

1. Задача відбору ознак.
2. Критерії оцінювання інформативності ознак на основі евристичного, інформаційного, статистичного та імовірнісного підходів.
3. Оцінювання ознак за допомогою коефіцієнта кореляції знаків, коефіцієнта кореляції Фехнера, кількості інтервалів зміни номера класу.
4. Комбінована оцінка інформативності ознак.
5. Що таке генеральна сукупність, вибірка, екземпляр, ознака?
6. Вимоги до навчаючих вибірок даних.
7. Що таке репрезентативна вибірка даних?

3 Лабораторна робота № 3

РОЗПІЗНАВАННЯ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

Мета роботи: Вивчити архітектури формального нейрона та багатошарової нейромережі, а також методи їхнього навчання; ознайомитися з програмними продуктами, що моделюють персептрони.

Завдання до роботи

1. Ознайомитися з конспектом лекцій та рекомендованою літературою. Вивчити архітектури формального нейрона, багатошарового персептрона, а також методи їхнього навчання.

2. Використовуючи додаток А ознайомитися з засобами для моделювання нейромереж у пакеті MATLAB.

3. Сформувати навчаючу вибірку на основі формул лабораторної роботи № 1.

4. Визначити за табл. 3.1 згідно з номером варіанта студента V кількості шарів та нейронів у шарах для побудови багатошарових персептронів, а також функцію активації нейронів.

5. Використовуючи пакет MATLAB для відповідних варіанту студента вхідних даних (навчаючої вибірки) на основі одношарового та багатошарового персептронів побудувати розпізнаючі моделі.

6. Змінюючи значення кроку навчання, для одношарового та багатошарового персептронів з заданою функцією активації, що відповідає номеру варіанту студента, дослідити, як впливає величина кроку навчання на час навчання. Побудувати графіки залежності часу навчання персептронів від величини кроку навчання.

7. Зберегти у файлі на диску початкові параметри нейромоделей, результати їхнього навчання (матрицю ваг та структуру з зазначенням функцій активації і кількості нейроелементів у шарах) та роботи для навчаючої вибірки (помилку класифікації), значення на входах і виході, час навчання, час класифікації).

8. Результати виконання роботи занести в таблицю, стовпці якої повинні мати назви: назва архітектури нейромережі, кількість шарів, кількість нейронів у шарах, функції активації нейронів у шарах, метод навчання, час навчання, час класифікації навченої мережі, помилка класифікації.

Таблиця 3.1 - Параметри нейромереж для варіантів

V	Багатошаровий персептрон		Назва функції активації нейронів
	Кількість шарів	Кількості нейронів у шарах	
1	2	3-1	логістична сигмоїдна
2	3	3-3-1	тангенційна сигмоїдна
3	4	3-3-3-1	логістична сигмоїдна
4	2	2-1	тангенційна сигмоїдна
5	3	2-2-1	логістична сигмоїдна
6	4	2-2-2-1	тангенційна сигмоїдна
7	2	5-1	логістична сигмоїдна
8	3	5-5-1	тангенційна сигмоїдна
9	4	5-5-5-1	логістична сигмоїдна
10	2	4-1	тангенційна сигмоїдна
11	3	4-4-1	логістична сигмоїдна
12	4	4-4-4-1	тангенційна сигмоїдна
13	2	6-1	логістична сигмоїдна
14	3	6-3-1	тангенційна сигмоїдна
15	4	6-4-3-1	логістична сигмоїдна
16	2	7-1	тангенційна сигмоїдна
17	3	7-7-1	логістична сигмоїдна
18	4	7-3-4-1	тангенційна сигмоїдна
19	2	8-1	логістична сигмоїдна
20	3	8-8-1	тангенційна сигмоїдна
21	4	8-7-6-1	логістична сигмоїдна
22	2	2-1	тангенційна сигмоїдна
23	3	2-3-1	логістична сигмоїдна
24	4	2-4-4-1	тангенційна сигмоїдна
25	2	3-1	логістична сигмоїдна
26	3	6-2-1	тангенційна сигмоїдна
27	4	7-6-3-1	логістична сигмоїдна
28	2	4-1	тангенційна сигмоїдна
29	3	9-2-1	логістична сигмоїдна
30	4	9-3-2-1	тангенційна сигмоїдна

9. Порівняти одношаровий та багатошаровий персептрони за часом навчання і точністю класифікації; принципами побудови архітектури.

10. Відповісти на контрольні питання.

Зміст звіту

1. Мета роботи.
2. Короткі теоретичні відомості, що містять стислий опис архітектур формального нейрона та багат шарового перцептрона, а також методів їхнього навчання.
3. Опис процесу виконання роботи.
4. Тексти програм, розроблених (модифікованих) студентом.
5. Вхідні дані та результати роботи програм, узагальнюючі таблиці, графіки.
6. Висновки, що містять відповіді на контрольні питання, а також відображують результати виконання роботи та їх критичний аналіз.

Контрольні питання

1. Поняття: формальний нейрон, синапс, поріг, дискримінантна (вагова) функція, функція активації, цільова функція навчання.
2. Загальне уявлення про навчання нейромереж.
3. Характеристики процесу навчання нейромереж.
4. Нейронні мережі у пакеті MATLAB.
5. Метод Уїдрой-Хоффа. Чи завжди збігаються алгоритми навчання одношарового перцептрона?
6. Чи доцільно застосовувати одношаровий перцептрон для класифікації складно (нелінійно) роздільних образів?
7. Чи впливає величина кроку навчання на час навчання одношарового перцептрона, багат шарової мережі? Відповідь обґрунтуйте теоретично та доведіть експериментально.
8. Які функції активації нейронів найчастіше використовують?
9. Чи впливає кількість використаних ознак на швидкість навчання перцептрона? Відповідь обґрунтуйте теоретично та доведіть експериментально.
10. Багат шаровий перцептрон: модель і принципи побудови архітектури.
11. Які задачі можна вирішувати на основі багат шарових перцептронів, а які не можна? Обґрунтуйте і доведіть відповідь. Приведіть приклади.
12. Які методи навчання багат шарових нейромереж Вам відомі?

4 Самостійна робота № 1

ПРОГРАМНІ ЗАСОБИ ДЛЯ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

Мета роботи: освоїти засоби середовища MATLAB для створення на його основі прикладних інтелектуальних систем.

Завдання до роботи

1. Використовуючи рекомендовану літературу вивчити основні принципи побудови програм для пакету MATLAB.
2. Розробити на мові для пакету MATLAB функції, що демонструють роботу з матрицями (із використанням матричних та масивних операцій), будують графіки, зберігають змінні на диску. Розроблені функції інтегрувати до графічної форми, яку створити за допомогою GUI.
3. Порівняти можливості пакету MATLAB із відомими системами програмування.

Зміст звіту

1. Мета роботи.
2. Короткі теоретичні відомості.
3. Тексти програм та інтерфейсні форми, розроблені студентом.
4. Вхідні дані та результати роботи програми.
5. Висновки, що містять відповіді на контрольні питання, а також відображують результати виконання роботи та їх критичний аналіз.

Контрольні питання

1. Архітектура та характеристики пакету MATLAB.
2. Інтелектуальний аналіз даних у пакеті MATLAB.
3. Поняття та властивості інтелектуальної системи.
4. Агентний підхід до інтелектуальних систем.
5. Архітектура інтелектуальної системи.

5 Самостійна робота № 2

САМООРГАНІЗАЦІЯ ТА НАВЧАННЯ БЕЗ УЧИТЕЛЯ. КЛАСТЕР-АНАЛІЗ

Мета роботи: вивчити та засвоїти на практиці методи кластер-аналізу та його використання для навчання з учителем та без учителя.

Завдання до роботи

1. Ознайомитися з конспектом лекцій та рекомендованою літературою. На алгоритмічній мові програмування пакету MATLAB написати програму, що використовує функції кластер-аналізу (див. додаток Б).
2. Згідно з номером студента за журналом для відповідного номера варіанта V сформувані навчаючу вибірку за допомогою формул лабораторної роботи № 1.
3. На основі вибірки виділити центри кластерів шляхом чіткої та нечіткої кластеризації.
4. Виконати нормування навчаючої вибірки даних.
5. На основі пронормованої вибірки виділити центри кластерів шляхом чіткої та нечіткої кластеризації.
6. Порівняти результати для чіткої та нечіткої кластеризації для нормованої та ненормованої вибірок.

Зміст звіту

1. Мета роботи.
2. Короткі теоретичні відомості до роботи.
3. Текст розробленої програми.
4. Сформована навчаюча вибірка.
5. Пронормована вибірка.
6. Координати центрів кластерів для ненормованої та нормованої вибірок після чіткої та нечіткої кластеризації.
7. Висновки. У висновках треба проаналізувати результати роботи, а також лаконічно відповісти на контрольні питання.

Контрольні питання

1. Задача кластер-аналізу.
2. Чіткий кластер-аналіз.
3. Нечіткий кластер-аналіз.
4. Використання кластер-аналізу при побудові систем розпізнавання образів.
5. Навчання з учителем.
6. Навчання без учителя.
7. Подібність кластер-аналізу і метричної класифікації.
8. Лінійна роздільність і лінійна нерозділеність класів.
9. Чи впливає кількість використаних ознак на швидкість кластер-аналізу?
10. Що таке генеральна сукупність, вибірка, екземпляр, ознака?
11. Чи впливає обсяг навчаючої вибірки на швидкість кластер-аналізу?
12. Функції кластер-аналізу у пакеті MATLAB.
13. Функція `fcm`.
14. Функція `subclust`.
15. Застосування кластер-аналізу для категоризації даних.

6 Самостійна робота № 3

ГІБРИДНІ НЕЙРО-НЕЧІТКІ СИСТЕМИ

Мета роботи: Освоїти основні принципи побудови гібридних нейро-нечітких мереж у пакеті MATLAB.

Завдання до роботи

1. Згідно з номером індивідуального варіанта студента за журналом V згенерувати навчаючу вибірку даних на основі формул лабораторної роботи № 1.

2. Для згенерованої вибірки за допомогою редактору anfisedit побудувати нейро-нечітку модель. Спробувати використати різні алгоритми кластеризації, різні кількості функцій приналежності для входів, різні кількості циклів навчання та різні алгоритми навчання.

3. Протестувати побудовану модель.

4. Проаналізувати отримані результати та відповіді на питання: який алгоритм кластер-аналізу призводить до отримання мережі меншої складності; як впливає задана кількість циклів навчання на точність навчання; як впливає задана точність навчання на тривалість навчання; які вимоги мають висуватися до навчаючої вибірки та як це вплине на процес навчання.

5. Виконати пункти 1-4 у режимі командного вікна без застосування редактору anfisedit, використовуючи функції модуля Fuzzy Logic Toolbox.

Зміст звіту

1. Мета роботи.
2. Короткі теоретичні відомості.
3. Номер варіанту та завдання до роботи.
4. Опис процесу та результати виконання завдань. Тексти програм. Згенеровані вибірки. Зображення структури отриманої мережі.
5. Висновки, що містять відповіді на контрольні запитання, а також відображують результати виконання роботи та їх критичний аналіз.

Контрольні питання

1. Структура та елементи нейро-нечітких мереж.
2. Застосування нейро-нечітких мереж для видобування знань з даних.
3. Нечітка кластеризація як підхід до представлення знань.
4. Який алгоритм кластер-аналізу призводить до отримання мережі меншої складності?
5. Як впливає задана кількість циклів навчання на точність навчання?
6. Як впливає задана точність навчання на тривалість навчання?
7. Які вимоги мають пред'являтися до навчальної вибірки та як це вплине на процес навчання?
8. Функції пакету MATLAB для створення нейро-нечітких мереж.
9. Редактор anfisedit.
10. Нейро-нечіткі мережі в задачах розпізнавання і прийняття рішень.

ЛІТЕРАТУРА

1. Джонс М.Т. Программирование искусственного интеллекта в приложениях / М.Т. Джонс. Пер. с англ. Осипов А.И.–М.: ДМК Пресс, 2004.– 312 с.
2. Дубровин В.И. Интеллектуальные средства диагностики и прогнозирования надежности авиадвигателей / В.И. Дубровин, С.А. Субботин, А.В. Богуслаев, В.К. Яценко. Монография.– Запорожье: ОАО "Мотор–Сич", 2003.– 279 с.
3. Зайченко Ю.П. Основы проектирования интеллектуальных систем. / Ю.П. Зайченко Навчальний посібник.– К.: Слово, 2004.– 352 с.
4. Кричевский М.Л. Интеллектуальные методы в менеджменте. / М.Л. Кричевский – СПб.: Питер, 2005.– 304 с.
5. Леоненков А.В. Нечеткое моделирование в среде MATLAB. / А.В. Леоненков. – СПб.: БХВ–Петербург, 2003. – 736 с.
6. Люгер Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж.Ф. Люгер Пер. с англ.– М.: Вильямс, 2005.– 864 с.
7. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2–е изд.: / С.Рассел Пер с англ.–М.: Вильямс, 2006.– 1408 с.
8. Рідкокаша А.А., Основы систем штучного інтелекту. / А.А. Рідкокаша, К.К. Голдер. Навчальний посібник.– Черкаси, "Відлуння–Плюс", 2002.–240 с.
9. Руденко О.Г. Основы теории искусственных нейронных сетей / О.Г. Руденко, Е.В. Бодянский. – Харьков Телетех, 2002. – 317с.
10. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : Навчальний посібник / С. О. Субботін. – Запоріжжя: ЗНТУ, 2008. –341 с.
11. Субботін С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей : Монографія / С. О. Субботін, А. О. Олійник, О. О. Олійник ; під заг. ред. С. О. Субботіна. – Запоріжжя: ЗНТУ, 2009. – 375 с.
12. Прогрессивные технологии моделирования, оптимизации и интеллектуальной автоматизации этапов жизненного цикла авиационных двигателей: Монография / А. В. Богуслаев, Ал. А. Олейник, Ан. А. Олейник, Д. В. Павленко, С. А. Субботин; Под ред. Д. В. Павленко, С. А. Субботина. – Запорожье: ОАО "Мотор Сич", 2009. – 468 с.
13. Хайкин С. Нейронные сети: полный курс, 2–е изд. / С. Хайкин: Пер. с англ.– М.: Вильямс, 2006.– 1104 с.

Додаток А

МОДЕЛЮВАННЯ НЕЙРОННИХ МЕРЕЖ У СЕРЕДОВИЩІ MATLAB

MATLAB являє собою математичний пакет, призначений для вирішення задач обчислювальної математики та побудови чисельних моделей складних об'єктів і процесів.

Пакет MATLAB складається з інтерпретатора – модельного середовища, що має термінальний інтерфейс, ядра (набору найпростіших стандартних операцій, функцій і процедур для обчислень), а також бібліотек функцій (Toolbox).

Перевагами пакета є: багаті графічні можливості, великий набір математичних функцій, простота вбудованої мови MATLAB, можливість автоматичного перетворення текстів програм мовою MATLAB у тексти програм на мові Сі, а також те, що тексти бібліотечних функцій постачаються у вихідному виді. До недоліків пакета варто віднести відсутність дружнього інтерфейсу користувача і низьку швидкість роботи.

Для моделювання НМ за допомогою пакета MATLAB необхідно встановити і використовувати бібліотеку Neural Network Toolbox. Розглянемо приклад моделювання і навчання нейромережі прямого поширення мовою пакета MATLAB (табл. А.1).

Окрім розглянутих засобів моделювання НМ пакет MATLAB, починаючи з версії 6.0, містить візуальний інтерфейсний модуль nntool, який входить до бібліотеки Neural Network Toolbox.

Використання nntool дозволяє більш зручними засобами, ніж написання програми вручну, будувати нейромережеві моделі об'єктів та процесів. Розглянемо деякі основні можливості та прийоми роботи із засобом nntool.

Після запуску Matlab.exe в командному вікні для початку роботи із nntool треба ввести: nntool. Після цього завантажиться засіб nntool (рис. А.1).

На панелі Network and Data ("Нейромережі та дані") користувач має натиснути кнопки для завдання вихідних даних для побудови нейромережевої моделі.

Кнопка New Data ("Нові дані") викликає редактор для створення нових даних (рис. А.2).

Поле Name ("Ім'я") задає ім'я нової змінної середовища MATLAB, до якої зберігається масив нових даних, що вводяться у полі Value ("Значення"). Панель Data Type ("Тип даних") визначає призначення введених даних: Inputs - входи мережі, Targets - цільові значення виходів мережі, Input Delay States та Layer Delay States - описи затримок на входах та у шарах мережі, Outputs - фактичні значення на виходах мережі, Errors - помилки мережі.

Таблиця А.1 – Моделювання і навчання перцептрона

$x = [0.1 \ 0.5 \ 0.2 \ 0.4 \ 0.3 \ 0.9;$ $0.9 \ 0.5 \ 0.8 \ 0.6 \ 0.7 \ 0.1;$ $0.3 \ 0.0 \ 0.6 \ 0.1 \ 0.2 \ 0.9];$	Задаємо значення ознак екземплярів навчальної вибірки: 6 екземплярів (стовпці), 3 ознаки (рядки).
$y = [1 \ 0 \ 1 \ 0 \ 1 \ 0];$	Задаємо номери класів для 6 екземплярів навчальної вибірки.
$net=newff(repmat([0 \ 1], 3, 1),$ $[2,1], \ {'logsig', \ 'logsig'},$ $'trainlm');$	Створюємо нейронну мережу <i>net</i> і визначаємо її топологію: діапазон зміни значень ознак [0 1], кількість ознак – 3, кількість вихідних змінних – 1, на першому шарі – 2 нейрони, на другому шарі 1 – нейрон, нейрони 1 і 2 шарів мають сигмоїдні функції активації (<i>logsig</i>), для навчання мережі використовується метод Левенберга-Марквардта (<i>trainlm</i>).
$net.trainparam.show=25;$	Задаємо період відображення інформації про процес навчання на екрані в циклах (епоках) навчання.
$net.trainparam.lr= 0.01;$	Задаємо крок навчання.
$net.trainparam.epochs=500;$	Задаємо максимально припустиму кількість циклів навчання (епох).
$net.trainparam.goal=0.01;$	Задаємо максимально припустиме значення критерію навчання (помилки навчання).
$ct=cputime;$	Визначаємо і запам'ятовуємо поточне значення лічильника часу в змінній <i>ct</i> .
$net=train(net, x, y);$	Навчаємо нейронну мережу <i>net</i> на основі навчальної вибірки, представленої набором значень ознак екземплярів <i>x</i> і набором значень відповідних їм номерів класів <i>y</i> .
$ct=cputime-ct$	Визначаємо поточне значення лічильника часу, віднімаємо від нього значення змінної <i>ct</i> – визначаємо час навчання НМ, що заносимо в змінну <i>ct</i> і видаємо на екран (ознака друку на екран – відсутність символу “;” наприкінці оператора).
$a=round(sim(net, x));$	Обчислюємо по навченій мережі <i>net</i> номери класів для екземплярів, що характеризуються набором значень ознак <i>x</i> .

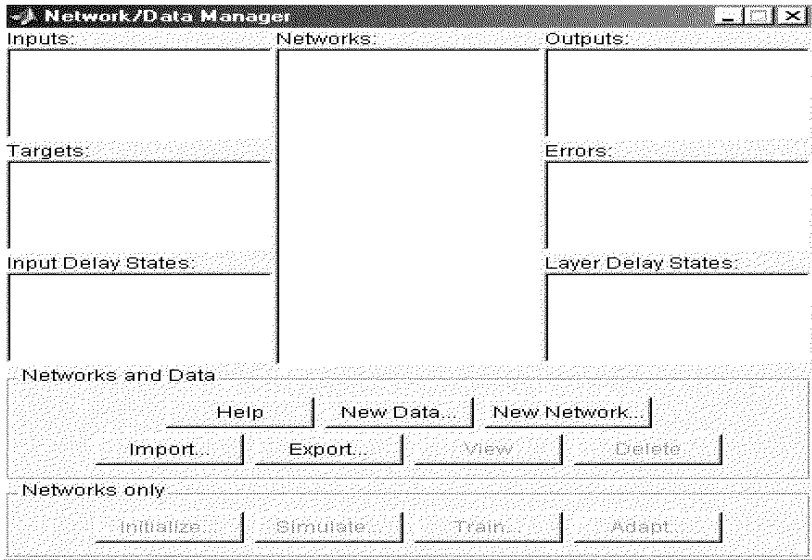


Рисунок А.1 – Головна діалогова форма засобу nntool

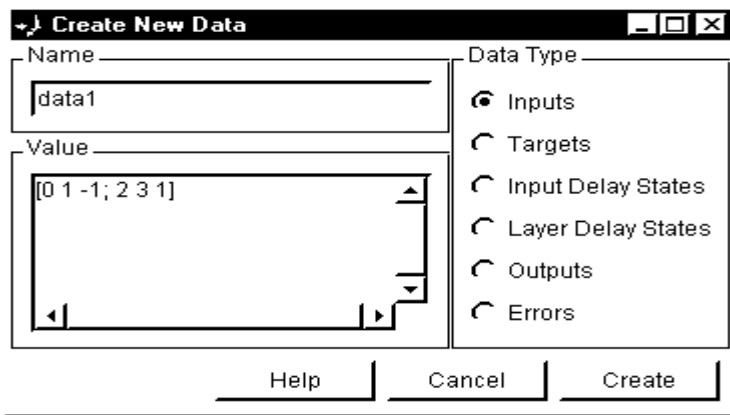


Рисунок А.2 - Редактор даних засобу nntool

Якщо дані уже існують у вигляді зовнішніх файлів або містяться у середовищі MATLAB у вигляді змінних, вони можуть бути імпортовані за допомогою кнопки Import ("Імпорт"). При цьому з'являється діалогова форма (рис. А.3).

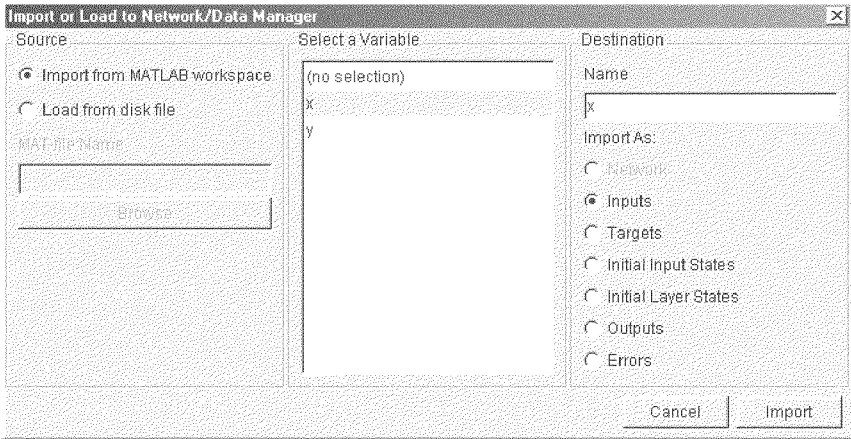


Рисунок А.3 - Діалогова форма імпорту даних

Поле Source ("Джерело") дозволяє обрати джерело введення даних: Import from Matlab workspace (імпорт даних із середовища MATLAB) або Load from disk file (завантаження даних із файлу на диску). Кнопка Browse дозволяє обрати необхідний файл.

Поле Select a Variable ("Вибір змінної") дозволяє вказати засобу nntool, яку змінну треба використовувати для імпорту даних.

Панель Destination ("Приймач") дозволяє задати змінну для прийому даних, що імпортуються. Її ім'я вказується у полі Name ("Ім'я"), а призначення (Import as) обирається із наведеного меню.

Кнопка Export ("Експорт") головної діалогової форми дозволяє зберегти дані із середовища nntool у файлі на диску, або передати їх до середовища MATLAB.

Кнопка "New Network" ("Нова мережа") викликає діалогову форму для конструювання нейромережі та визначення її параметрів (рис. А.4).

Поле Network Name ("Ім'я мережі") визначає ім'я змінної, де зберігається мережа. Список вибору Network Type ("Тип мережі") дозволяє обрати тип архітектури мережі (наприклад, Feed-forward backprop - багатoshарова нейромережа прямого поширення), поля Training Function, Adaptation Learning Function, Performance Function та Number of Layers визначають, відповідно, тип алгоритму навчання мережі, тип алгоритму адаптації ваг мережі, цільову функцію та кількість шарів мережі.

Панель Properties for Layer K дозволяє задати властивості для нейронів K-го шару мережі. У полі Number of Neurons вказують кількість нейронів для поточного шару, а у полі Transfer Function - тип функції активації нейронів поточного шару мережі.

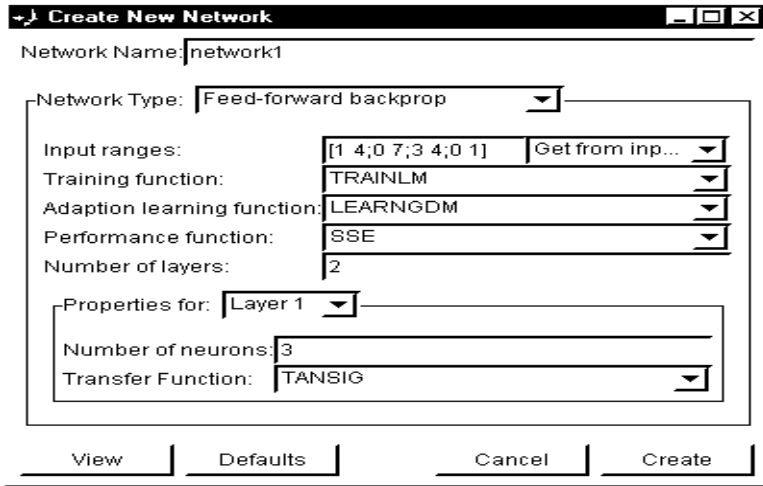


Рисунок А.4 - Форма конструювання нейромережі

Кнопка "View" дозволяє отримати графічне зображення схеми побудованої нейромережі (рис. А.5).

Кнопка "Delete" головної форми дозволяє видалити непотрібний елемент даних (змінну або мережу).

Кнопка "Help" дозволяє викликати довідкову службу MATLAB з описом необхідних компонентів та поясненнями щодо їхнього використання.

Після побудови нейромережі у нижній частині головної діалогової форми nntool стає доступною панель Networks only, що призначена для роботи із побудованою мережею (рис. А.6).

При натисненні будь-якої з кнопок цієї панелі викликається діалогова форма Network ("Мережа"), яка містить набір закладок-панелей для роботи з мережею (рис. А.7-А.10).

Закладка Initialize ("Ініціалізація") дозволяє задати межі, в яких змінюються вхідні дані та розрахувати на їхній основі початкові значення ваг мережі.

Ініціалізована мережа може бути навчена за допомогою закладки Train ("Тренування, навчання"). Серед параметрів навчання, доступних на цій закладці обов'язково необхідно задати: goal - максимально припустиме значення цільової функції, epochs - максимальна припустима кількість циклів навчання мережі, show - шаг виводу на екран інформації про навчання мережі, задається в циклах навчання.

В процесі навчання середовище MATLAB буде графік зміни значення цільової функції по епохах - циклах навчання (рис. А.11).

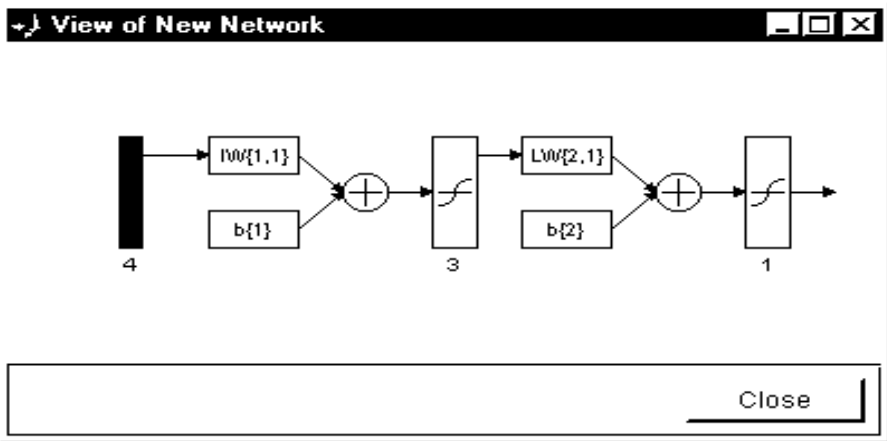


Рисунок А.5 - Приклад зображення форми нейромережі, побудованої за допомогою засобу nntool

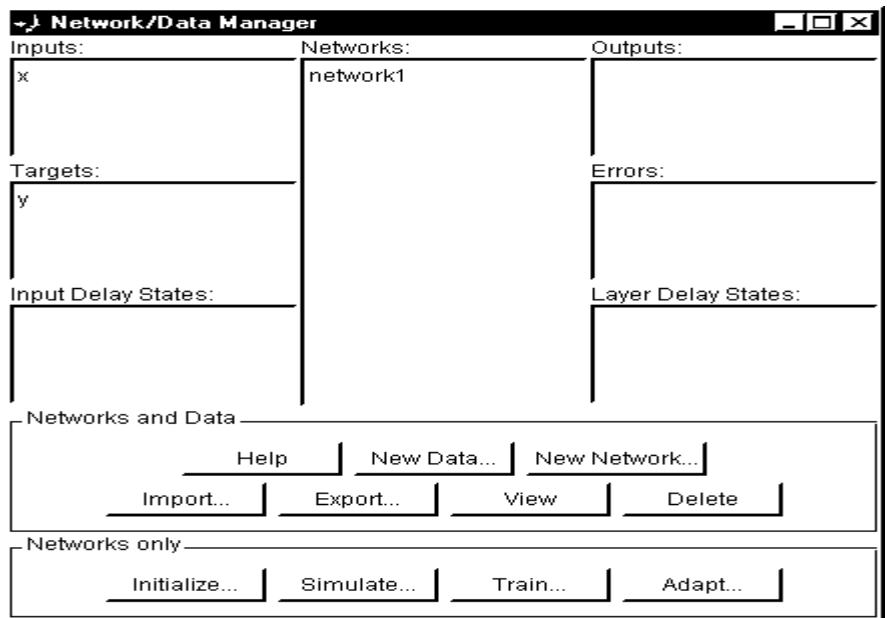


Рисунок А.6 - Головна діалогова форма nntool після побудови мережі

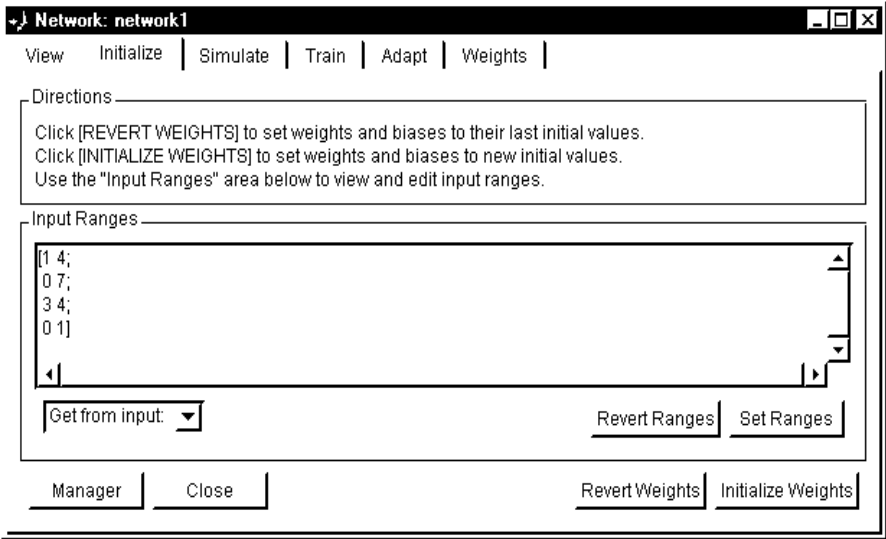


Рисунок А.7 - Форма Network: закладка Initialize

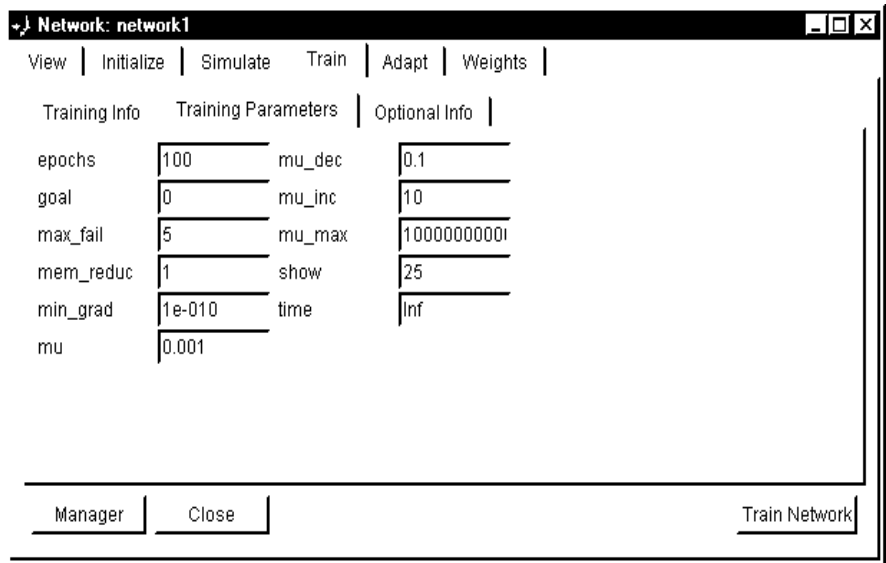


Рисунок А.8 - Форма Network: закладка Train

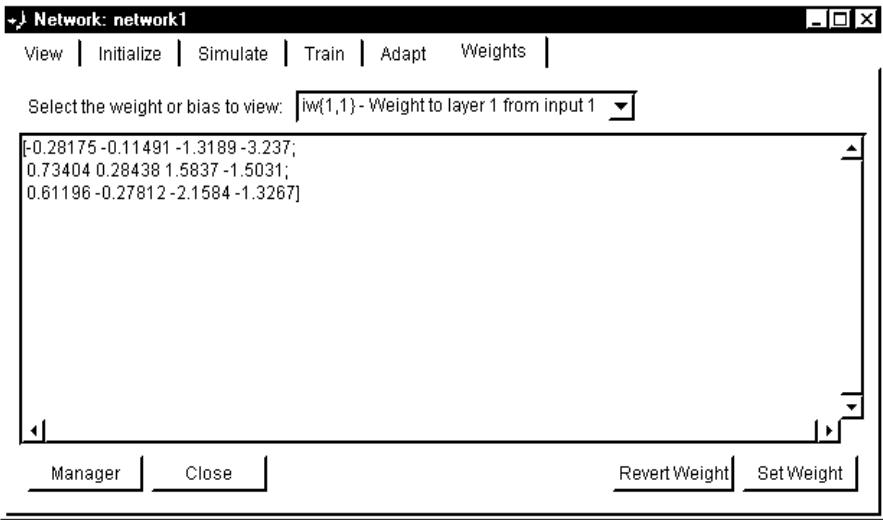


Рисунок А.9 - Форма Network: закладка Weights

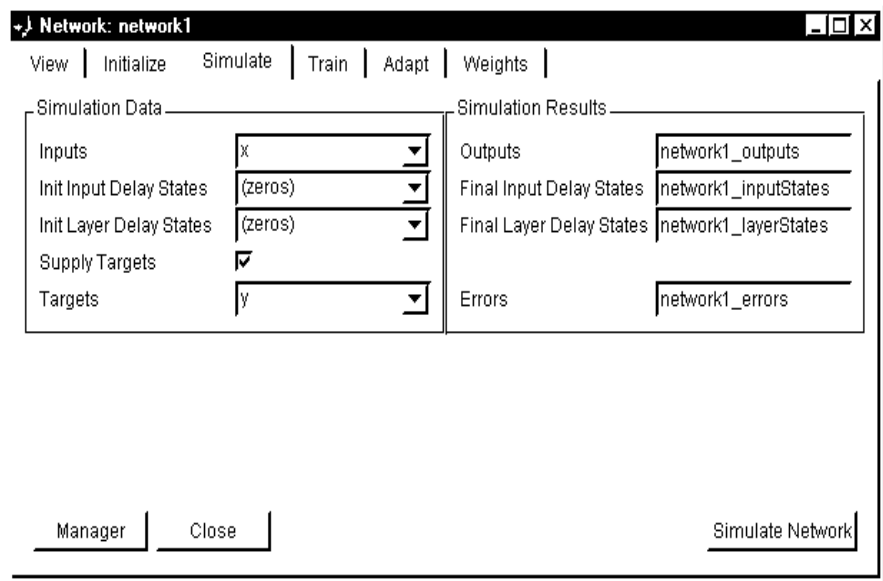


Рисунок А.10 - Форма Network: закладка Simulate

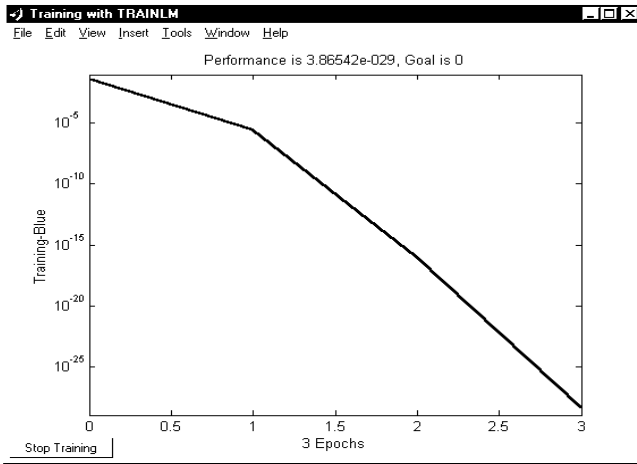


Рисунок А.11 - Графік зміни значення цільової функції в процесі навчання

Ваги мережі, що була навчена, можна переглянути, використовуючи закладку Weights ("Ваги").

Після того, як мережа навчилася, її можна використовувати для розпізнавання за допомогою закладки Simulate ("Моделювання").

Нейронні мережі, створені за допомогою функцій MATLAB, мають єдиний формат представлення, як це показано у табл. А.2.

Таблиця А.2 - Представлення структури нейромережі у MATLAB

Поле структури	Опис, значення поля
numInputs	кількість входів мережі
numLayers	кількість (схованих) шарів мережі без урахування вхідного шару
biasConnect, inputConnect, layerConnect, outputConnect, targetConnect	булеві масиви, що визначають зв'язки між структурними елементами мережі
numOutputs	кількість виходів мережі
numTargets	кількість цільових ознак
numInputDelays	кількість затримок вхідного шару
numLayerDelays	кількість затримок схованих шарів
inputs	входи мережі
layers	шари мережі
outputs	виходи мережі

Продовження табл. А.2

Поле структури	Опис, значення поля
targets	цільові ознаки
biases	масив порогів нейронів мережі
inputWeights	масив вагових коефіцієнтів вхідного шару мережі
layerWeights	масив вагових коефіцієнтів схованих шарів мережі
adaptFcn	ім'я функції адаптації нейронів
initFcn	ім'я функції ініціалізації мережі
performFcn	ім'я цільової функції навчання (помилки)
trainFcn	ім'я функції, що реалізує процес навчання
adaptParam	параметри адаптації мережі
initParam	параметри ініціалізації мережі
performParam	параметри цільової функції мережі
trainParam.epochs	максимально допустима кількість ітерацій навчання (epoch)
trainParam.goal	максимально допустиме значення цільової функції навчання
trainParam.max_fail	максимально допустима кількість відмов у процесі навчання мережі
trainParam.mem_reduc	коефіцієнт, що регулює (зменшує) використання пам'яті при навчанні нейромереж
trainParam.min_grad	мінімально допустиме значення градієнту цільової функції
trainParam.mu, trainParam.mu_dec, trainParam.mu_inc, trainParam.mu_max	параметри методу Левенберга-Марквардта
trainParam.show	кількість ітерацій, через яку будуть відображатися зміни стану процесу навчання мережі
trainParam.time	максимально допустимий час навчання мережі у секундах
IW	масив значень ваг вхідного (першого) шару
LW	масив значень ваг схованих шарів
b	масив значень порогів нейронів
userdata	дані користувача

Додаток Б

КЛАСТЕР-АНАЛІЗ У СЕРЕДОВИЩІ MATLAB

Модуль Fuzzy Logic Toolbox пакету MATLAB містить функції для виділення кластерів.

Функція *subclust* визначає координати центрів кластерів шляхом чіткої кластеризації зі зменшенням кількості кластерів.

Функція *subclust* знаходить оптимальну точку даних для визначення центра кластера ґрунтуючись на щільності оточення точок даних. Усі точки даних у межах відстані *RADII* до цієї точки видаляються, щоб визначити наступний кластер даних та його центр. Цей процес повторюється поки усі дані не знаходяться у межах відстані *RADII* до центра кластера.

[*C*] = SUBCLUST (*X*, *RADII*) кластеризує точки даних S^xN матриці *X*, де *S* - кількість точок даних, *N* - кількість координат точок даних, *RADII* - значення між 0 та 1, що визначає розмір кластера в кожному з вимірювань даних, приймаючи, що дані знаходяться у межах діапазону [0, 1] (Встановлення меншого радіуса кластера буде звичайно створювати більше менших за розміром кластерів. Коли *RADII* є скаляром, він застосовується до усіх вимірів даних. Коли *RADII* є вектором, він має окреме значення для кожного виміру даних), та повертає центри кластерів як рядки матриці *C*, що має розмір V^xN , де *V* - кількість кластерів.

[*C*] = SUBCLUST (... , *XBOUNDS*) також визначає матрицю *XBOUNDS*, розміром 2^xN , що використовується для нормалізації даних *X* у діапазон [0, 1]. Кожний стовпець *XBOUNDS* містить мінімальні та максимальні значення для відповідної розмірності даних. Якщо *XBOUNDS* - порожня матриця або не використовується, тоді за замовчуванням використовуються мінімальні та максимальні значення даних *X*.

[*C*] = SUBCLUST (... , *OPTIONS*) визначає вектор для зміни значень за замовчуванням параметрів алгоритму: *OPTIONS*(1) - коефіцієнт, що використовується для множення на значення *RADII* для визначення осередку центру кластера, у межах якого існування інших центрів кластерів заборонено; *OPTIONS*(2) - коефіцієнт прийняття, що встановлює потенціал як частку потенціалу центра першого кластера, вище якої інша точка даних буде прийнята як центр кластера; *OPTIONS*(3) - коефіцієнт відхилення, що встановлює потенціал як частку потенціалу центра першого кластера, нижче якої інша точка даних буде відхилена як центр кластера; *OPTIONS*(4) - ознака відображення поточної інформації, якщо не встановлена як 0.

Значеннями вектора *OPTIONS* за замовчуванням є [1.25 0.5 0.15 0].

Приклад використання subclust.

```
X1 = 10*rand(50,1);
X2 = 5*rand(50,1);
X3 = 20*rand(50,1)-10;
X = [X1 X2 X3]; % генеруємо вибірку даних
[C] = subclust(X,0.5); % знаходимо центри кластерів
```

Функція *fcm* здійснює нечітку кластеризацію на основі методу нечітких *s*-середніх та має формат виклику:

```
[CENTER, U, OBJ_FCN] = fcm(DATA, N_CLUSTER, OPTIONS)
```

де *N_CLUSTER* - кількість кластерів в наборі даних масиву *DATA*, який має розміри *S***N*, *S* - кількість точок даних, *N* - кількість координат точок, *CENTER* - матриця з координатами центрів кластерів (кластери містяться у рядках, ознаки - у стовпцях), *U* - матриця функції приналежності, що містить рівні приналежності кожної точки масиву *DATA* до кожного кластера, *OBJ_FCN* - значення цільової функції для центрів кластерів, *OPTIONS* - необов'язковий параметр, що задає вектор опцій для процесу кластеризації: *OPTIONS*(1) - експонента для матриці *U* (за замовчуванням: 2.0), *OPTIONS*(2) - максимальна кількість ітерацій (за замовчуванням: 100), *OPTIONS*(3) - мінімально прийнятне покращення цільової функції (за замовчуванням: 10^{-5}), *OPTIONS*(4): ознака відображення проміжних результатів (за замовчуванням: 1).

На кожній ітерації цільова функція мінімізується для знаходження кращого розташування кластерів. Процес кластеризації зупиняється, коли максимально прийнятна кількість ітерацій є досягнутою, або коли покращення цільової функції між двома послідовними ітераціями змінюється меншою ніж мінімально прийнятний приріст.

Приклад використання fcm.

```
data = rand(100,2); % генеруємо вибірку
[center,U,obj_fcn] = fcm(data,2); % виконуємо кластер-аналіз
plot(data(:,1), data(:,2),'o'); % зображуємо дані на графіку
hold on;
maxU = max(U);
% Знаходимо точки з найвищим рівнем приналежності до першого кластера
index1 = find(U(1,:) == maxU);
% Знаходимо точки з найвищим рівнем приналежності до другого кластера
index2 = find(U(2,:) == maxU);
line(data(index1,1),data(index1,2),'marker','*','color','g');
line(data(index2,1),data(index2,2),'marker','*','color','r');
% Зображуємо центри кластерів на графіку
plot([center([1 2],1)],[center([1 2],2)],'*','color','k');
```