

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти)

на тему ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЛАСИФІКАЦІЇ
ТЕСТОВИХ ЗАВДАНЬ

Виконав: студент 2 курсу, групи КНТ-512м
спеціальності 123 Комп'ютерна інженерія

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні системи та мережі

СІРКО К. М.

(ПРИЗВИЩЕ та ініціали)

Керівник КАСЬЯН М.М.

(ПРИЗВИЩЕ та ініціали)

Рецензент МАЛИЙ О.Ю.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
 Кафедра «Комп'ютерні системи та мережі»
 Ступінь вищої освіти магістерський
 Спеціальність 123 Комп'ютерна інженерія
 (код і найменування)
 Освітня програма (спеціалізація) «Комп'ютерні системи та мережі»
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Зав. кафедри Кудерметов Р.К.

“ ” 2023 року

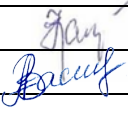
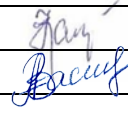
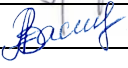
З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

СІРКО Катерина Михайлівна

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Інтелектуальна система класифікації тестових завдань
- керівник проєкту (роботи) к. т. н., доцент, КАСЬЯН Микола Миколайович
 (науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)
- затверджені наказом вищого навчального закладу від “24” жовтня 2023 року № 400
2. Строк подання студентом проєкту (роботи) 10 грудня 2023 року
3. Вихідні дані до проєкту (роботи) тестовий матеріал, основні форми та принципи композиції тестових завдань, основні показники якості тестових завдань, існуючі методи оцінки якості тестових завдань, моделі нейронних мереж для розв'язання завдань класифікації
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - 1) Аналіз методів оцінки якості тестових завдань;
 - 2) Розробка методу оцінки якості тестових завдань на базі нейронної мережі;
 - 3) Розробка системи класифікації тестових завдань.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проєкту (роботи)

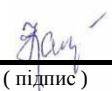
Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3	КАСЬЯН М. М., доцент		
нормоконтроль	ПОЛЬСЬКА О.В., ст. викл.		

7. Дата видачі завдання 01.10.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз технічного завдання	01.10.2023 р.	
2	Огляд основних форм і принципів композиції тестових завдань	05.10.2023 р.	
3	Аналіз основних показників якості тестових завдань та існуючих методів оцінки якості тестових завдань	10.10.2023 р.	
4	Розробка методу оцінки якості тестових завдань на базі нейронної мережі	15.10.2023 р.	
5	Вибір моделі та топології нейронних мереж	20.10.2023 р.	
6	Вибір алгоритму навчання нейронної мережі	25.10.2023 р.	
7	Навчання нейронних мереж	01.11.2023 р.	
8	Тестування нейронних мереж	10.11.2023 р.	
9	Розробка системи класифікації тестових завдань на базі нейронної мережі	20.11.2023 р.	
10	Обробка результатів системи класифікації тестових завдань	01.12.2023 р.	
11	Оформлення документації	10.12.2023 р.	

Студент  **Катерина СІРКО**
(підпис) (ім'я, ПРИЗВИЩЕ)

Керівник проєкту (роботи)  **Микола КАСЬЯН**
(підпис) (ім'я, ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 53 с., 6 таблиць, 14 рисунків, 14 джерел, 2 додатка.

ЕКСПЕРТНА ОЦІНКА, МЕТОДИ ОЦІНКИ ЯКОСТІ, НЕЙРОНА МЕРЕЖА, ПЕРСЕПТРОН, ПОКАЗНИКИ ЯКОСТІ, СТАТИСТИЧНА ОБРОБКА, ТЕСТОВЕ ЗАВДАННЯ

Мета магістерської роботи – розробити систему класифікації тестових завдань з оцінки якості на підставі результатів дослідження можливостей нейронних мереж.

У магістерській роботі проведено аналіз основних форм та принципів композиції тестових завдань, а також основних показників якості тестових завдань. Проведено дослідження існуючих методів оцінки якості тестових завдань.

Розроблено метод оцінки якості тестових завдань на базі нейронних мереж. У ході дослідження було спроектовано три нейронні мережі, проведено навчання та тестування. За результатами тестування було обрано одношаровий перцептрон, на базі якого розроблено та реалізовано на практиці систему класифікації тестових завдань з оцінки якості.

ABSTRACT

EN: 53 p., 6 tables, 14 pictures, 14 references, 2 appendices

EXPERT EVALUATION, QUALITY ASSESSMENT METHODS,
NEURON NETWORK, PERCEPTRON, QUALITY INDICATORS, STATISTICAL
PROCESSING, TEST TASK

The purpose of the master's thesis is to develop a system of classification of test tasks for quality assessment based on the results of research on the capabilities of neural networks.

The master's work analyzed the main forms and principles of the composition of test tasks, as well as the main indicators of the quality of test tasks. A study of existing methods of assessing the quality of test tasks was conducted.

A method for assessing the quality of test tasks based on neural networks has been developed. In the course of the study, three neural networks were designed, trained and tested. According to the test results, a single-layer perceptron was chosen, on the basis of which a system of classification of test tasks for quality assessment was developed and implemented in practice.

ЗМІСТ

Переліки скорочень та умовних позначок.....	7
Вступ.....	8
1 Аналіз методів оцінки якості тестових завдань	12
1.1 Огляд основних форм та принципів композиції тестових завдань	12
1.2 Аналіз основних показників якості тестових завдань	17
1.3 Методи оцінки якості тестових завдань.....	17
1.3.1 Статистична обробка результатів тестування.....	18
1.3.2 Експертна оцінка якості тестових завдань	19
2 Розробка методу оцінки якості тестових завдань на базі нейронної мережі....	20
2.1 Метод оцінки якості тестових завдань.....	20
2.2 Вибір нейронної мережі для системи класифікації тестових завдань	23
2.2.1 Вибір моделі та топології НМ.....	24
2.2.2 Навчання НМ.....	28
2.2.2 Тестування НМ.....	33
3 Розробка системи класифікації тестових завдань.....	35
3.1 Система класифікації тестових завдань.....	35
3.2 Обробка результатів розробленої системи	37
Висновки	42
Перелік джерел посилання	44
Додаток А Результати навчання НМ.....	46
Додаток Б Програмна реалізація розробленої системи.....	50

ПЕРЕЛІКИ СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

НМ - нейронна мережа;

СДО - системах дистанційного навчання;

MOODLE - Modular Object-Oriented Dynamic Learning Environment;

LMS - Learning Management System;

CMS - Content management system.

ВСТУП

Розробка та запровадження інноваційних інформаційних технологій в освіті є актуальним напрямом процесу інформатизації сучасного суспільства. Все більш популярними стають такі методи навчання, як онлайн навчання, дистанційне навчання, змішане навчання та ін. Такі технології навчання ґрунтуються на використанні засобів обчислювальної техніки та систем передачі даних для представлення та доставки знань, підтримки взаємодії студента та викладача, а також контролю знань.

В університетах все більше уваги приділяється існуванню таких засобів як e-learning та m-learning, які виступають у ролі автоматизованої системи управління навчальним процесом, запроваджуються віртуальні та дистанційні лабораторії. Основна мета цих засобів полягає в тому, щоб надати користувачам повнофункціональне, масштабоване та гнучке середовище для реалізації систем електронного навчання.

В даний час існує безліч систем електронної освіти, серед яких можна виділити Blackboard, WebCt, Moodle, IBM Learning Space та ін. У цій системі існує три типи формату онлайн-курсів: форум, структура (навчальні модулі без прив'язки до календаря), календар (навчальні модулі з прив'язкою до календаря). Курс може містити довільну кількість ресурсів та довільну кількість інтерактивних елементів курсу.

До ресурсів належать web-сторінки, книги, посилання на файли, каталоги. До інтерактивних елементів курсу відносяться Wiki, анкети, глосарій, завдання (завдання, яке вимагає від учнів підготувати відповідь в електронному вигляді та завантажити її на сервер), опитування, пояснення, тести, лекції.

З іншого боку стрімкий розвиток та експлуатація мобільних пристроїв (смартфони, планшети і т.д.), а також підвищення пропускної спроможності бездротового зв'язку сприяє розвитку нових способів навчання, зокрема, мобільної освіти (m-learning).

При цьому традиційні методи навчання та контролю (лекції, семінари, лабораторні заняття, іспити, заліки, колоквіуми тощо) не втрачають свого значення. Сучасні технології навчання їх виключають. У традиційному процесі навчання засоби e-learning та m-learning повинні використовуватися студентами для самостійної роботи, на яку відводиться більша частина курсу, а саме, 2/3 частини від загальних годин вивчення дисципліни, тобто змішане навчання (blended learning).

Змішане навчання – це використання сучасних дистанційних технологій у поєднанні із традиційним очним навчанням. При цьому основний акцент у навчанні переноситься на самостійну, індивідуальну роботу кожного студента. Завдання викладача в умовах сучасної технології навчання зводиться не лише до читання лекцій та проведення практичних та лабораторних занять, а також до організації робочих місць студентів, створення методичного забезпечення навчального процесу та систематичного контролю за самостійною роботою студентів та результатами своєї педагогічної роботи в цілому.

Такий підхід навчання має забезпечити кожного студента індивідуальною програмою з урахуванням персональних особливостей та уподобань, а також відповідною до його рівня знань. Тому саме розробка, впровадження засобів e-learning та m-learning та проектування онлайн курсів за допомогою цих інструментів набуває особливого значення, оскільки без них організувати індивідуальну роботу з кожним студентом викладачеві неможливо. Іншими словами, завдання сучасного педагога – створити всі умови для мотивованої самостійної роботи студентів та організувати зворотний зв'язок із студентською аудиторією.

На етапі розвитку освітнього процесу серед першочергових стоять завдання різкого підвищення якості навчання. Важливою науковою проблемою є пошук ефективних шляхів розробки та впровадження у навчальний процес прогресивних інформаційних систем комп'ютерного тестування, призначених для інтенсивного інтерактивного управління навчальним процесом.

Тестова форма комп'ютерного контролю знань є одним із найактуальніших сучасних методологічних прийомів інтенсифікації процесу навчання та технологічних засобів ефективної організації зворотного зв'язку системи з об'єктом управління. Організація опитувань чи анкетування сприяє забезпеченню ефективного зворотного зв'язку із студентською аудиторією, дає можливість подальшого статистичного аналізу відповідей аудиторії, дозволяє здійснювати моніторинг навчального процесу очима студента для організації ефективного контролю якості контенту та викладання.

Актуальність тестового методу обумовлена його перевагами над іншими педагогічними методами контролю якості знань студентів [1]:

- наукова обґрунтованість тесту, що дає об'єктивну оцінку рівня знань, умінь та навичок студентів;
- технологічність тестових методів;
- точність визначень;
- наявність єдиних вимог для всіх піддослідних;
- економити навчальний час та підвищити ефективність праці викладачів, звільнити їх від рутинної роботи;
- автоматизувати процедури підготовки та проведення тестування;
- проводити масове тестування (як групових, і індивідуальних - у разі використання комп'ютерних класів) за відносно короткий час;
- сумісність тестових технологій з іншими сучасними освітніми технологіями;
- впровадити математико-статистичні методи обробки результатів та оцінки якості надійності та валідності тестів;
- оперативно опрацювати результати тестування та рекомендувати корекційні заходи удосконалення навчального процесу.

Таким чином обрана тема магістерської роботи є актуальною. Мета роботи - розробити систему класифікації тестових завдань з оцінки якості на підставі результатів дослідження можливостей нейронних мереж (НМ).

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз основних форм та принципів композиції тестових завдань, а також основних показників якості тестових завдань;
- провести дослідження існуючих методів оцінки якості тестових завдань;
- розробити метод оцінки якості тестових завдань на основі НМ;
- розробити систему класифікації тестових завдань;
- реалізувати розроблену систему класифікації тестових завдань.

Об'єктом дослідження є оцінка якості тестових завдань.

Предметом дослідження є методи визначення оцінки якості тестових завдань.

Наукова новизна магістерської роботи полягає у розробці методу оцінки якості тестових завдань на основі НМ.

Практична цінність магістерської роботи полягає у розробці системи класифікації тестових завдань на основі НМ.

1 АНАЛІЗ МЕТОДІВ ОЦІНКИ ЯКОСТІ ТЕСТОВИХ ЗАВДАНЬ

1.1 Огляд основних форм та принципів композиції тестових завдань

Алгоритм складання тестового матеріалу наступний [1,2]:

- структурування навчального матеріалу;
- становлення логічних зв'язків між елементами навчального контенту;
- складання тестових завдань;
- вибір оптимальної форми тестових завдань;
- вкладання плану тесту;
- перевірка тесту на великій вибірці піддослідних.

При структуруванні навчального матеріалу робиться приблизний розподіл відсоткового співвідношення змісту розділів навчального курсу і визначається необхідна кількість тестових завдань з кожного розділу. Підраховується загальна запланована кількість завдань в тесті, як правило, беруть 60-80 завдань, а максимальний час тестування обирають 90 хвилин. Розробляється специфікація тесту, яка ґрунтується на відповідності системи знань і вмінь певним відсотковим співвідношенням завдань з різних розділів. До переліку знань і вмінь загального характеру для технічних дисциплін включено [2]:

- відтворення фактів, понять, визначень і термінів;
- знання законів і формул;
- вміння застосовувати закони та формули для розв'язування задач;
- вміння інтерпретувати результати за допомогою графіків і схем;
- вміння робити оцінкові судження.

Існує кілька форм тестових завдань (рис. 1.1) [3], серед яких можна виділити чотири:

- а) з вибором правильної відповіді:
 - 1) з вибором однієї правильної відповіді;
 - 2) з вибором кількох правильних відповідей;

3) з вибором однієї, найбільш правильної відповіді з певної кількості відповідей, правильних різною мірою;

б) відкрита форма;

в) на встановлення відповідності:

1) на встановлення відповідності кожному елементу першого стовпця одного елемента другого;

2) на встановлення відповідності кожному елементу першого стовпця кількох елементів другого;

3) матричні завдання (число відповідностей розширюється на три та більше множин);

г) встановлення правильної послідовності.



Рисунок 1.1 – Класифікація тестових завдань

Інші форми тестових завдань є або варіантами однієї з перерахованих, або змішання двох і більше форм. Для побудови тестових завдань використовуються такі принципи композиції тестових завдань:

- принцип протиріччя (використовується заперечення у разі відповідей);
- принцип протилежності (одна змістовна відповідь замінюється іншою, антонімічною за змістом);
- принцип однорідності (підбираються відповіді, які відносяться до одного роду, виду тощо);
- принцип кумуляції (вміст другої відповіді вбирає вміст першого, вміст третього - другого і т.д.);
- принцип поєднання (поєднання слів (знаків) по два або по три, рідше чотири в кожній відповіді);
- принцип градування;
- принцип подвоєного протиставлення;
- принцип фасетності змісту завдання (фасет – це форма запису кілька варіантів однієї й тієї завдання);
- принцип імплікації, «якщо ..., то»;
- принцип формування завдань із відповідями, правильними різною мірою;
- поєднання принципів (використовуються кілька принципів композиції при створенні одного завдання).

Основні елементи тестових завдань:

- інструкції для піддослідних;
- зміст завдань;
- форма;
- зміст та кількість відповідей.

Для форми подання текстових тестових завдань наступні рекомендації [3]:

- тестові завдання однакової форми супроводжуються однією інструкцією для їх виконання;

- текст інструкції бажано відрізнятися від основного тексту шрифтом або кольором;
- завдання нумеруються арабськими цифрами, нумерація завдань різної форми наскрізна;
- завдання формулюється у стверджувальній формі стисло, чітко, без подвійного тлумачення;
- елементи відповіді частини тестового завдання мають окрему індексацію;
- відповіді розміщуються під запитальною частиною симетрично або поряд з нею;
- якщо відповідь передбачає певну процедуру обчислення, то остання має бути простою, без потреби застосування складних технічних засобів;
- форма подання тестових завдань не змінюється в межах блоку завдань, призначеного для тестування.

Найпростішими тестовими завданнями закритої форми за принципом побудови відповіді є альтернативні тестові завдання.

Альтернативні тестові завдання передбачають наявність двох варіантів відповіді, зокрема "правильно - неправильно", "так - ні". Їх використовують для прийняття рішення за змістом завдання без розкриття його суті або попередньої перевірки правильності вибору.

Тестові завдання з множинним вибором, у яких передбачається від трьох до п'яти можливих варіантів відповідей, доцільно використовувати при необхідності перевірити уміння правильно відтворювати отримані знання. Із запропонованих варіантів - правильним є лише один. Під час складання тестових завдання з множинним вибором треба звернути увагу на підбір варіантів відповідей, які мають бути досить схожими на правильні.

Відкритої форми тестові завдання передбачають вільні відповіді тих, хто тестується. За змістом - це твердження з невідомою змінною. Такі тестові завдання не містять запропонованих варіантів відповідей і використовуються

для того щоб виявити знання термінів, означень, понять поданих у навчальному матеріалі.

За місцем у навчальному процесі тестування можна поділяють на:

- вхідне;
- поточне;
- підсумкове.

Вхідне тестування використовується для з'ясування рівня володіння базовими знаннями, вміннями, навичками або готовності до сприймання нового матеріалу та застосування його при виконанні практичних або лабораторних робіт.

Поточне тестування використовується для з'ясування успішності засвоєння навчального матеріалу на певному етапі навчання (конкретної теми, розділу), виявлення прогалин у знаннях.

Підсумкове тестування використовується для виявлення рівня навчальних досягнень з курсу.

Досить активно тести використовуються в системах дистанційного навчання (СДО), зокрема безкоштовній системі Moodle (Modular Object-Oriented Dynamic Learning Environment) [4]. Системою дистанційного навчання Moodle користуються більш ніж у 100 країнах світу, зокрема Національний Університет «Запорізька політехніка» [5]. Moodle проектувалась у відповідності з сучасними педагогічними вимогами:

- з опорою на принципи і методи діяльнісного активного навчання;
- особисту спрямованість;
- спільну діяльність;
- критичну рефлексію.

В системі дистанційного навчання Moodle використовуються тестові завдання з різним набором запитань (множинний вибір), з вибором вірно/не вірно, на відповідність, передбачається коротка тестова відповідь, а також числова або обчислювана. Всі запитання зберігаються в базі даних і можуть використовуватись повторно в різних курсах.

1.2 Аналіз основних показників якості тестових завдань

Тестові завдання та варіанти відповідей мають бути змістовними та, по можливості, короткими. Короткість забезпечується ретельним підбором слів, символів, графіків, що дозволяють мінімум коштів добиватися ясності змісту завдання. Форма завдань правильна, якщо вона дозволяє точно виразити зміст, зрозуміла всім випробуваних, виключає можливість появи помилкових відповідей за формальними ознаками.

Основні показники якості тестових завдань:

- стислість (лаконічність) формулювання завдання;
- стислість (лаконічність) формулювання варіантів відповідей;
- точність (однозначність) формулювання завдання;
- точність (однозначність) формулювання варіантів відповідей;
- ясність інструкції до завдання;
- коректність форми завдання;
- достатність необхідної кількості варіантів відповідей;
- відповідність формулювання варіантів відповідей завдання;
- відповідність завдання навчальній дисципліні, розділу, темі;
- відповідність до рівня складності завдання.

1.3 Методи оцінки якості тестових завдань

Визначити якість тестових завдань можна автоматично в ході статистичної обробки результатів тестування [6-8] та за допомогою оцінки експертів.

1.3.1 Статистична обробка результатів тестування

До основних показників статистичної обробки результатів тестування відносяться:

- частка правильних відповідей;
- частка неправильних відповідей;
- дисперсія j -ого тестового завдання (показник варіації);
- стандартне відхилення результатів випробуваних за j -им завданням;
- логіт проблеми j -ого тестового завдання.

Частка правильних відповідей обчислюється за формулою:

$$P_j = R_j / N \quad (1.1)$$

де R_j - число правильних відповідей на j -е завдання;
 N - кількість тестових завдань у тесті.

Частка неправильних відповідей обчислюється за формулою:

$$Q_j = W_j / N \quad (1.2)$$

де W_j - число неправильних відповідей на j -е завдання.

Дисперсія j -ого тестового завдання обчислюється за формулою:

$$P_j Q_j \quad (1.3)$$

Стандартне відхилення результатів випробуваних за j -им завданням обчислюється за формулою:

$$\sqrt{P_j Q_j} \quad (1.4)$$

Логіт проблеми j -ого тестового завдання обчислюється за формулою:

$$\ln P_j Q_j \quad (1.5)$$

Всі ці показники ґрунтуються на кількості правильних та неправильних відповідей.

Недоліком статистичної обробки результатів тестування і те, що вихідний тестовий матеріал необхідно випробувати досить великий вибірці учнів (щонайменше 150-200 людина). Зазвичай це накопичення даних протягом 3-5 років. За цей період можуть змінюватись умови та середовище навчання, що впливає на статистичні показники результатів тестування. Також зміни умов та середовища навчання впливає і на рівень складності тестового завдання, тому одне й те саме питання для різних студентів, які вивчали дисципліну у різних викладачів з різного методичного забезпечення, може мати для студентів різний рівень складності.

Також статистичні показники залежать і від мотивації тестованих. Зрозуміло, що мотивація студентів при підсумковому тестуванні є вищою, ніж при проміжному тестуванні, який проходить протягом навчання і його можна перездати до кінця навчання. Також підготовка до підсумкового тестування студента залежить від типу контролю дисципліни (залік, залік з оцінкою або іспит). Якщо за підсумком здачі дисципліни студент отримує залік, де будь-яка позитивна оцінка є заліком, і це не впливає на середній бал, то, як правило, і рівень підготовки нижче. Відповідно, показники статистичної обробки результатів тестування будуть у тестових завдань з високим рівнем складності низькі, що може призвести до виключення даних тестових завдань.

1.3.2 Експертна оцінка якості тестових завдань

Найбільш ефективними вважаються експертні методи [6,7], до яких належать індивідуальна експертна оцінка, морфологічний експертний метод, рейтинг, самооцінка обстежуваних, педагогічний консиліум, групова експертна оцінка та ін. При використанні експертних методів однією з основних проблем

є об'єктивність оцінки якості тестових завдань, компетентність експертів та наявність таких.

Виходячи з вище викладеного можна зробити висновок, що в даний час виникає необхідність у розробці більш зручного, не потребує великих тимчасових витрат та пошуку компетентних експертів, об'єктивного методу оцінки якості тестових завдань за певних умов та середовища навчання з можливістю подальшого доопрацювання як тестового, так і навчального матеріалу.

2 РОЗРОБКА МЕТОДУ ОЦІНКИ ЯКОСТІ ТЕСТОВИХ ЗАВДАНЬ НА БАЗІ НЕЙРОННОЇ МЕРЕЖІ

2.1 Метод оцінки якості тестових завдань

На рисунку 2.1 представлена структура потоків даних методу, що розробляється оцінки якості тестових завдань, в яку входять наступні компоненти:

- а) студенти;
- б) викладач;
- в) інформаційна система:
 - система тестування;
 - інструментарій для проведення анкетування;
 - база даних;
 - система класифікації тестових завдань на базі нейронної мережі.

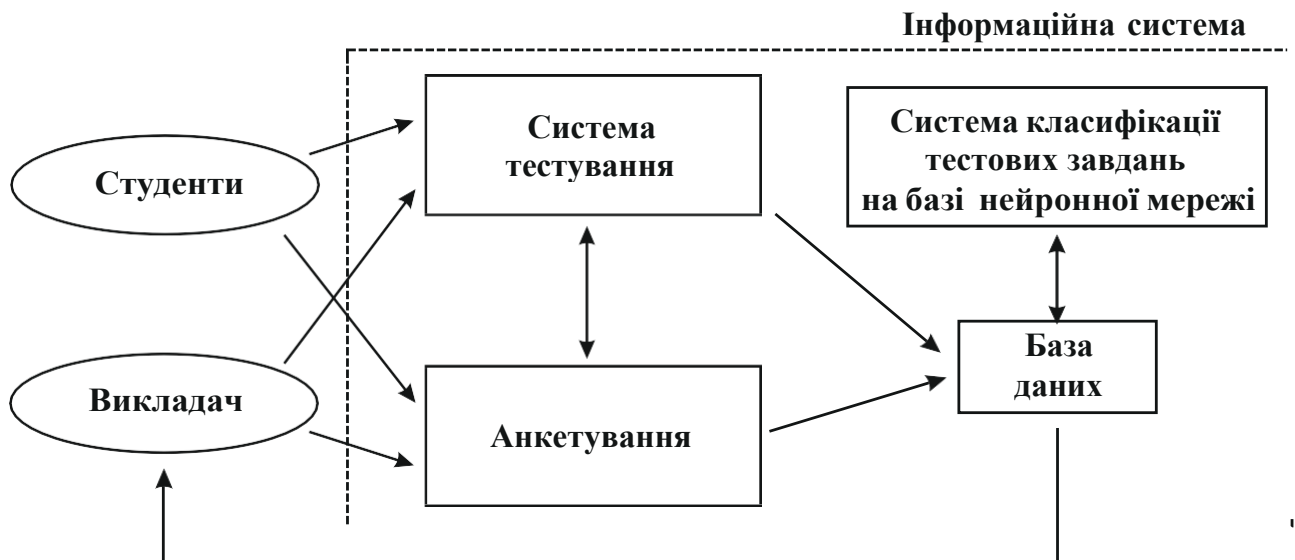


Рисунок 2.1 - Структура потоків даних методу оцінки якості тестових завдань

Основний компонент інформаційної системи є система тестування. На сьогоднішній день розроблено безліч різноманітних систем тестування, тому немає потреби розробляти нову. Серед існуючих систем тестування можна виділити комп'ютерну систему тестування знань OpenTEST 2.0 [9], розроблену в Харківському національному університеті радіоелектроніки, яка перебуває в експлуатації протягом кількох років у низці навчальних закладів України. Також дану компоненту може замінити система електронної освіти (LMS, CMS), до якої входить інструментарій комп'ютерного тестування. Зокрема, у Національний Університет «Запорізька політехніка» використовують Moodle – систему управління вмістом сайту, розроблену для створення викладачами якісних online-курсів навчальних дисциплін [5].

Після тестування студент, який отримав високий або достатній рівень, повинен відповісти на запитання анкети, пов'язані з показниками якості тестових завдань, на які дав правильні відповіді тестованій, для цього необхідний інструментарій для проведення анкетування (опитування). В даний час вже з'являються програми, що дозволяють створювати тести, опитування та анкети, наприклад, Mentimeter [10], mQlicker [11], Kahoot [12]. Для нашої роботи можна взяти вже розроблений додаток та синхронізувати із системою тестування або розробити свій програмний модуль для проведення опитування.

Розробником навчального та тестового матеріалу є викладач.

У ролі експертів, які оцінюватимуть показники якості тестових завдань, відбиратимуться студенти, які отримали високий або достатній рівень знань за результатами тестування та правильно відповіли на відповідне тестове завдання. Таким чином, як експерт на кожне тестове завдання будуть підбиратися різний склад студентів та різна кількість на кожне тестове завдання.

Також, до інформаційної системи має входити база даних, яка призначена для зберігання результатів тестування та оцінок показників якості тестових завдань та відбору даних для подальшої обробки (класифікації тестових завдань з оцінки якості).

Відібрані в базі даних результати опитування за кожним тестовим завданням подаються на нейронну мережу, на виході якої отримуємо підсумкову оцінку якості тестового завдання. Якщо оцінка якості тестового завдання низька чи середня, то викладач має можливість проаналізувати показники якості та доопрацювати це тестове завдання.

На рисунку 2.2 наведено алгоритм методу оцінки якості тестових завдань на базі нейронної мережі.

У рамках цієї магістерської роботи необхідно реалізувати систему класифікації тестових завдань з оцінки якості на підставі результатів дослідження можливостей нейронних мереж (НМ).



Рисунок 2.2 - Алгоритм методу оцінки якості тестових завдань

2.2 Вибір нейронної мережі для системи класифікації тестових завдань

В останні десятиліття у світі бурхливо розвивається нова прикладна область математики, що спеціалізується на НМ [13,14]. Різні сфери застосування НС - це автоматизація процесів розпізнавання образів, адаптивне управління, класифікація, апроксимація функціоналів, прогнозування, створення експертних систем, організація асоціативної пам'яті тощо.

Для розробки рішення на основі нейронної мережі потрібне таке [14]:

- вибрати відповідну модель мережі;
- визначити топологію мережі (тобто число елементів та їх зв'язку);
- вказати параметри навчання.

2.2.1 Вибір моделі та топології НМ

Перед вибором моделі та топології НМ треба задати вихідні дані.

Студент відповідає на всі питання анкети, які пов'язані з показниками якості тестового завдань. Запитань в анкеті десять. На вхід НМ подається відповіді у вигляді двійкового вектора з десяти елементів відповідно, де 1 означає, що студент оцінив показник якості високо, а 0 означає, що показник якості низький. На вході НМ повинні отримати якість тестового завдання, де 1 означає, що оцінка якості висока, а 0 означає, що оцінка якості низька.

Основу кожної НМ становлять відносно прості, здебільшого – однотипні, елементи (комірки), що імітують роботу нейронів мозку (штучний нейрон). Кожен нейрон характеризується своїм поточним станом за аналогією з нервовими клітинами головного мозку, які можуть бути збуджені або загальмовані. Він має групу синапсів - односпрямованих вхідних зв'язків, з'єднаних з виходами інших нейронів, а також має аксон - вихідний зв'язок даного нейрона, з яким сигнал (збудження або гальмування) надходить на синапс наступних нейронів. Загальний вид нейрона наведено рисунку 2.3.

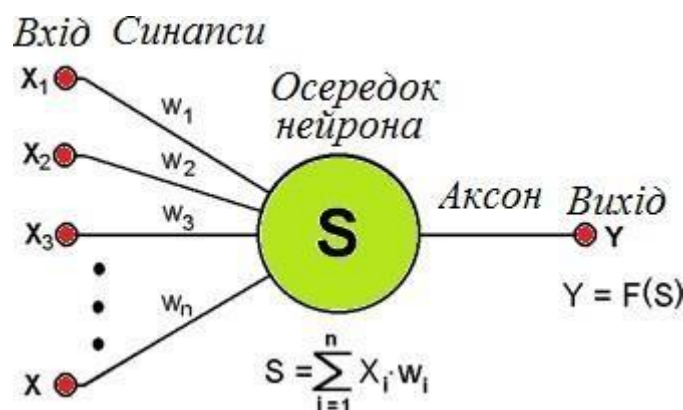


Рисунок 2.3 - Штучний нейрон

Кожен синапс характеризується величиною синаптичного зв'язку або його вагою w_i , який за фізичним змістом еквівалентний електричній провідності. Поточний стан нейрона визначається як зважена сума його входів:

$$s = \sum_{i=1}^n x_i \cdot w_i \quad (2.1)$$

де n - число входів нейрона;

x_i - значення i -го входу нейрона;

w_i - вага i -го синапсу.

Вихід нейрона є функцією його стану:

$$y = f(s) \quad (2.2)$$

Нелінійна функція f називається активаційною і може мати різний вигляд, як показано на рисунку 2.4.

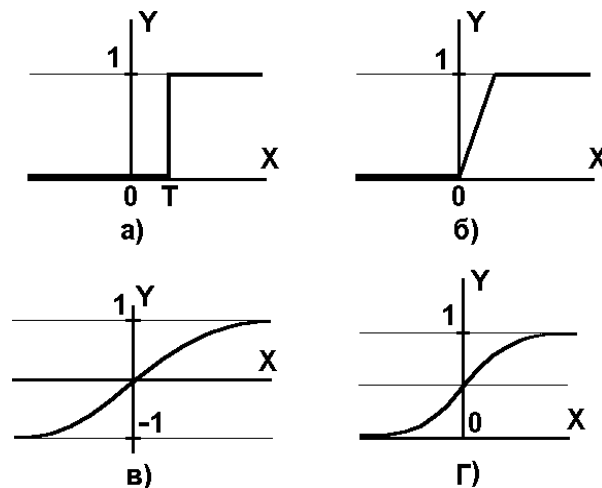


Рисунок 2.4 - Функції активації

а) функція одиничного стрибка; б) лінійний поріг (гістерезис);

в) сигмоїд – гіперболічний тангенс; г) сигмоїд (логістична)

Однією з найпоширеніших є нелінійна функція з насиченням, так звана логістична функція або сигмоїд:

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (2.3)$$

При зменшенні α сигмоїд стає більш пологим, у межі при $\alpha=0$ вироджуючись у горизонтальну лінію на рівні 0.5, при збільшенні α сигмоїд наближається на вигляд до функції одиничного стрибка з порогом T у точці $x=0$. З виразу для сигмоїда очевидно, що вихідне значення нейрона лежить у діапазоні $[0,1]$.

Всім НМ властивий принцип паралельної обробки сигналів, який досягається шляхом об'єднання великої кількості нейронів у так звані шари та сполуки певним чином нейронів різних шарів, а також, у деяких конфігураціях, і нейронів одного шару між собою, причому обробка взаємодії всіх нейронів ведеться пошарово. На рисунку 2.5 представлений двошаровий персептрон.

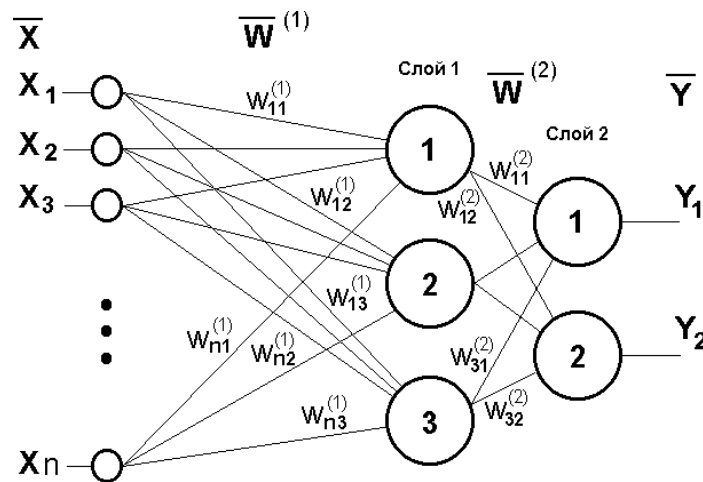


Рисунок 2.5 - Двошаровий персептрон

Більшість відомих на сьогоднішній день СР використовують для знаходження зваженої суми входів нейрона формулу (2.1).

Для проєктування НМ використовуємо пакет прикладних програм Neural Network Toolbox (Нейронні мережі) потужного середовища математичного моделювання MATLAB.

Для вирішення задачі беремо: одношаровий персептрон з десятима входами та одним виходом та з функцією активації одиничного стрибка (рис. 2.6); багатошаровий персептрон із десятима входами, одним виходом, трьома нейронами у прихованому шарі з логістичною функцією активації (рис. 2.7); багатошаровий персептрон із десятима входами, одним виходом, п'ятьма нейронами у прихованому шарі з логістичною функцією активації (рис. 2.8).

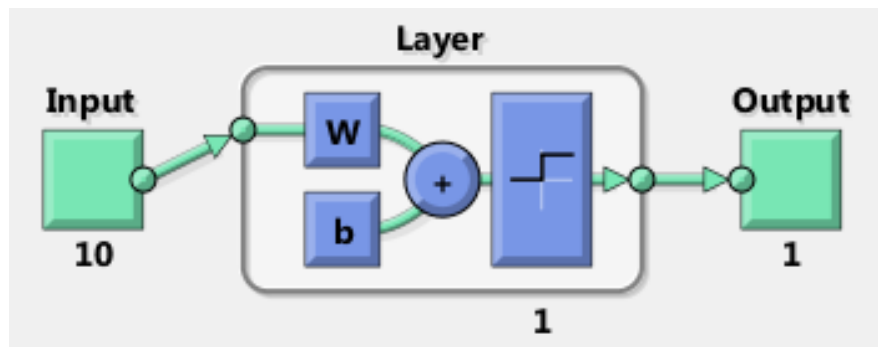


Рисунок 2.6 - Одношаровий персептрон з десятима входами та одним виходом

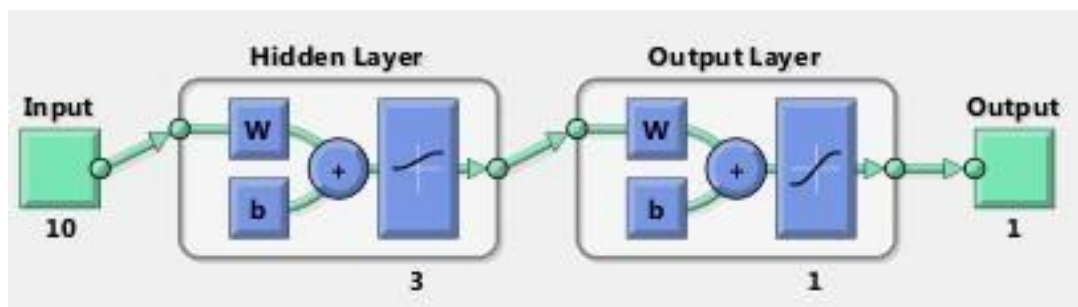


Рисунок 2.7 - Багатошаровий персептрон з десятима входами, одним виходом та трьома нейронами у прихованому шарі

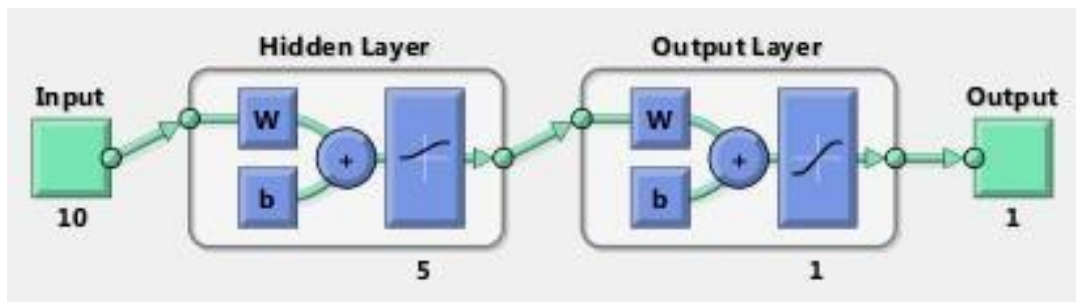


Рисунок 2.8 - Багатошаровий персептрон з десятима входами, одним виходом та п'ятьма нейронами у прихованому шарі

Важливо помітити, що для кожного шару (прихованого та вихідного) зображених на НМ є вектор зсувів. Для нейронів прихованого шару довжина вектора зсувів b_1 відповідає кількості нейронів прихованого шару, а нейрона вихідного шару b_2 складається з одного елементів. Усунення кожного нейрона підсумовується з виваженим входом і результат цієї суми є аргументом функції активації нейрона.

Далі сформуємо навчальний набір спостережень. Навчальна послідовність представлена 353-ма вхідними образами і таким же числом очікуваних оцінок $\{\text{inputs}; \text{targets}\}$, де inputs – вектори входів, targets – вектори цілей. Таблиця вхідних образів та відповідні їм очікувані оцінки наведено у додатку А.

2.2.2 Навчання НМ

Для визначення значень ваг та зсувів, які мінімізують помилку навчання, слід навчити НМ. Усі алгоритми навчання функціонують покроково, і ці кроки прийнято називати епохами (циклами). На кожному циклі на вхід мережі послідовно подаються всі елементи навчальної послідовності, потім обчислюється вихідні значення мережі, порівнюються з цільовими та обчислюється функціонал помилки. Значення функціоналу, а також його градієнта використовуються для коригування ваги та зсувів, після чого всі дії повторюються. Початкові значення ваги і зміщень вибираються випадковим чином, а процес навчання припиняється, коли виконано певну кількість циклів,

або коли помилка досягне деякого малого значення або перестане зменшуватися.

За такої формалізації завдання навчання передбачаються відомими бажані (цільові) реакції мережі на вихідні сигнали, що асоціюється з присутністю вчителя, а тому такий процес навчання називають навчанням з учителем. Саме така парадигма буде використана під час навчання у нашому випадку. Іноді алгоритми навчання з учителем називають також алгоритмами із заохоченням.

Для навчання мережі було обрано градієнтний алгоритм навчання під назвою Rprop (Resilientpropagation). Вибір даного алгоритму для навчання нейронної мережі, що вирішує задачу класифікації, обумовлений тим, що алгоритм Rprop є найшвидшим із усіх градієнтних алгоритмів навчання, що є в арсеналі MATLAB.

Процеси навчання для кожної з НМ у середовищі математичного моделювання MATLAB показані на рисунках 2.9, 2.10, 2.11.

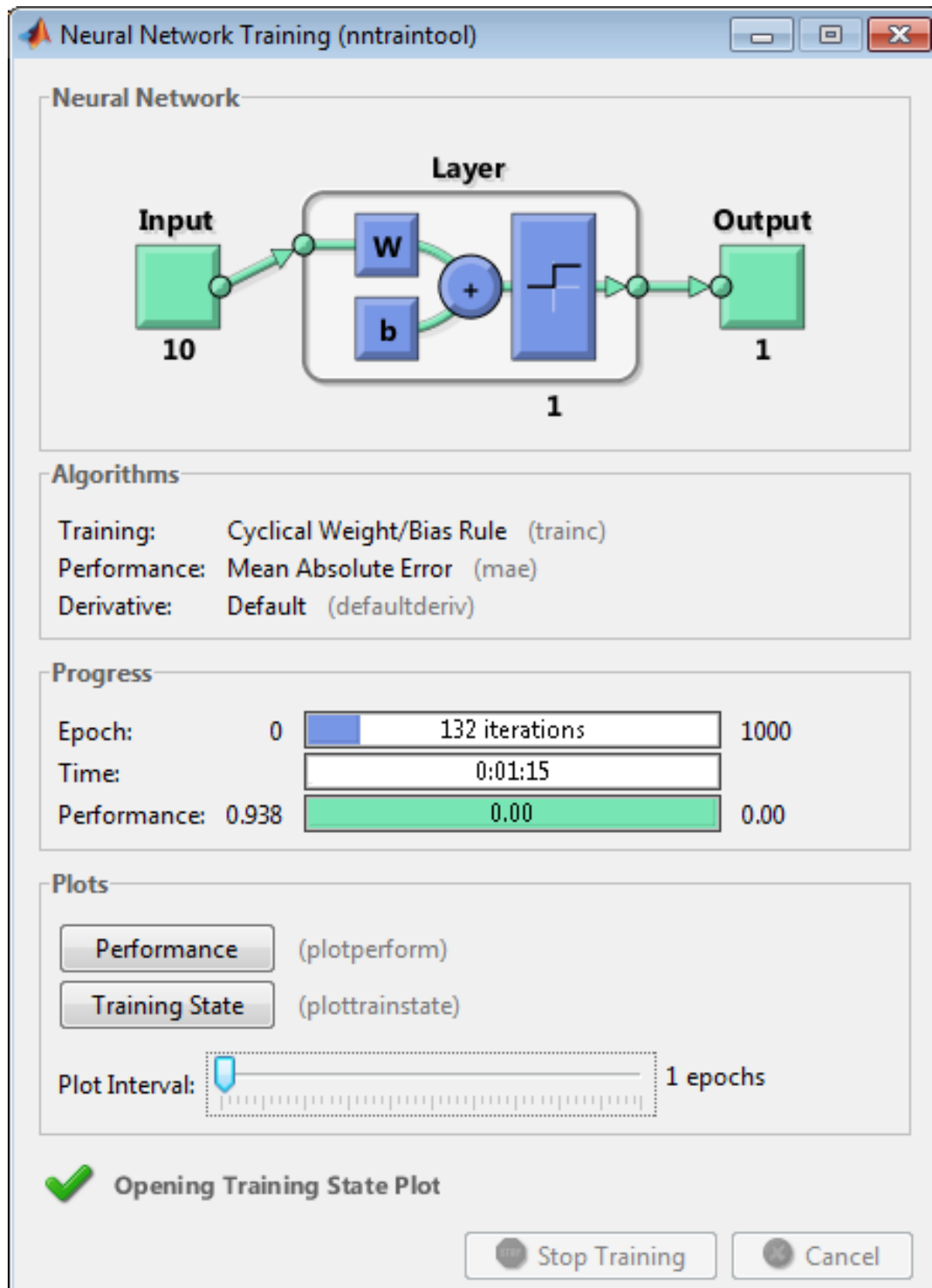


Рисунок 2.9 - Вікно процесу навчання для одношарового персептрону з десятима входами та одним виходом

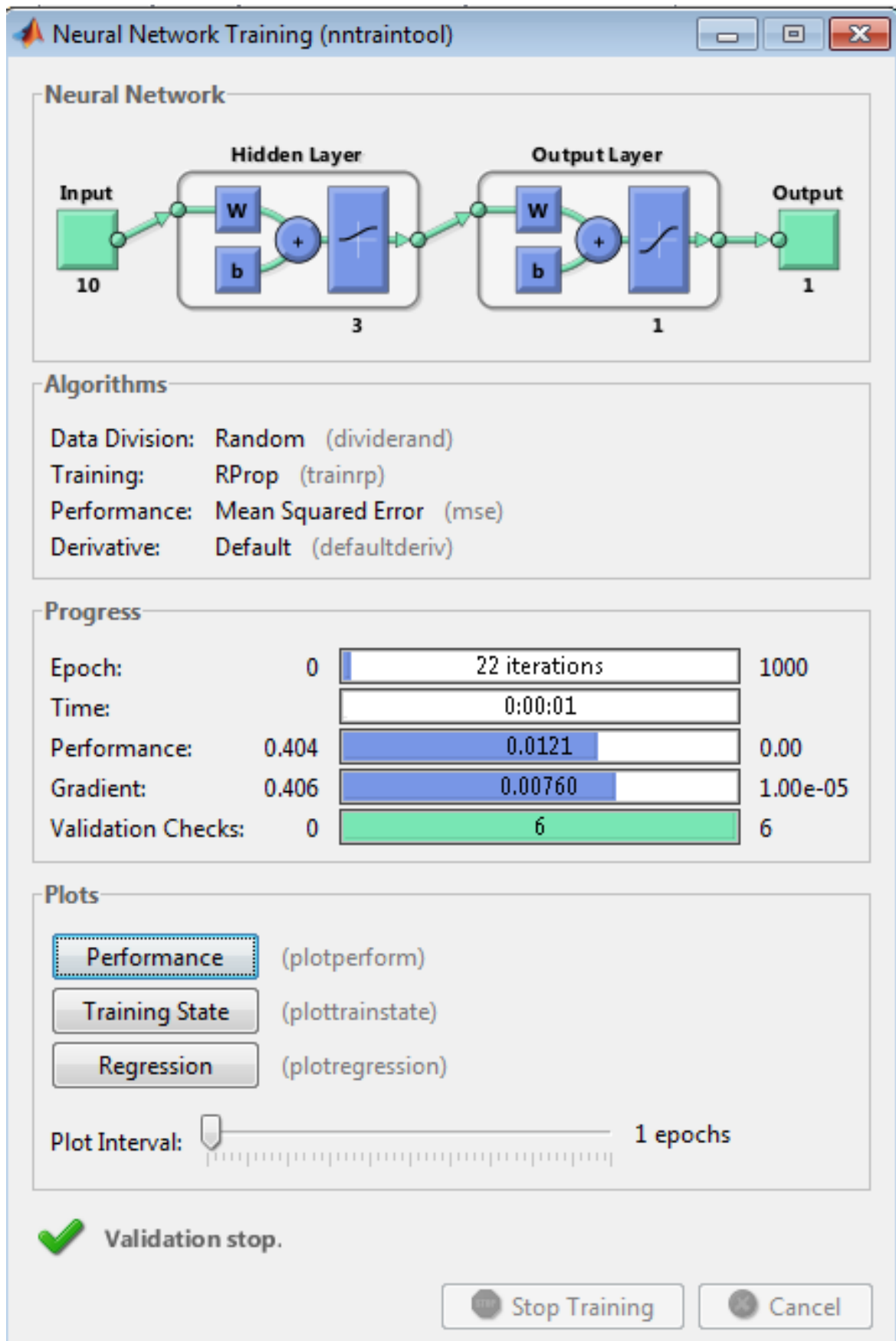


Рисунок 2.10 - Вікно процесу навчання для багатошарового персептрона з десятима входами, одним виходом та трьома нейронами у прихованому шарі

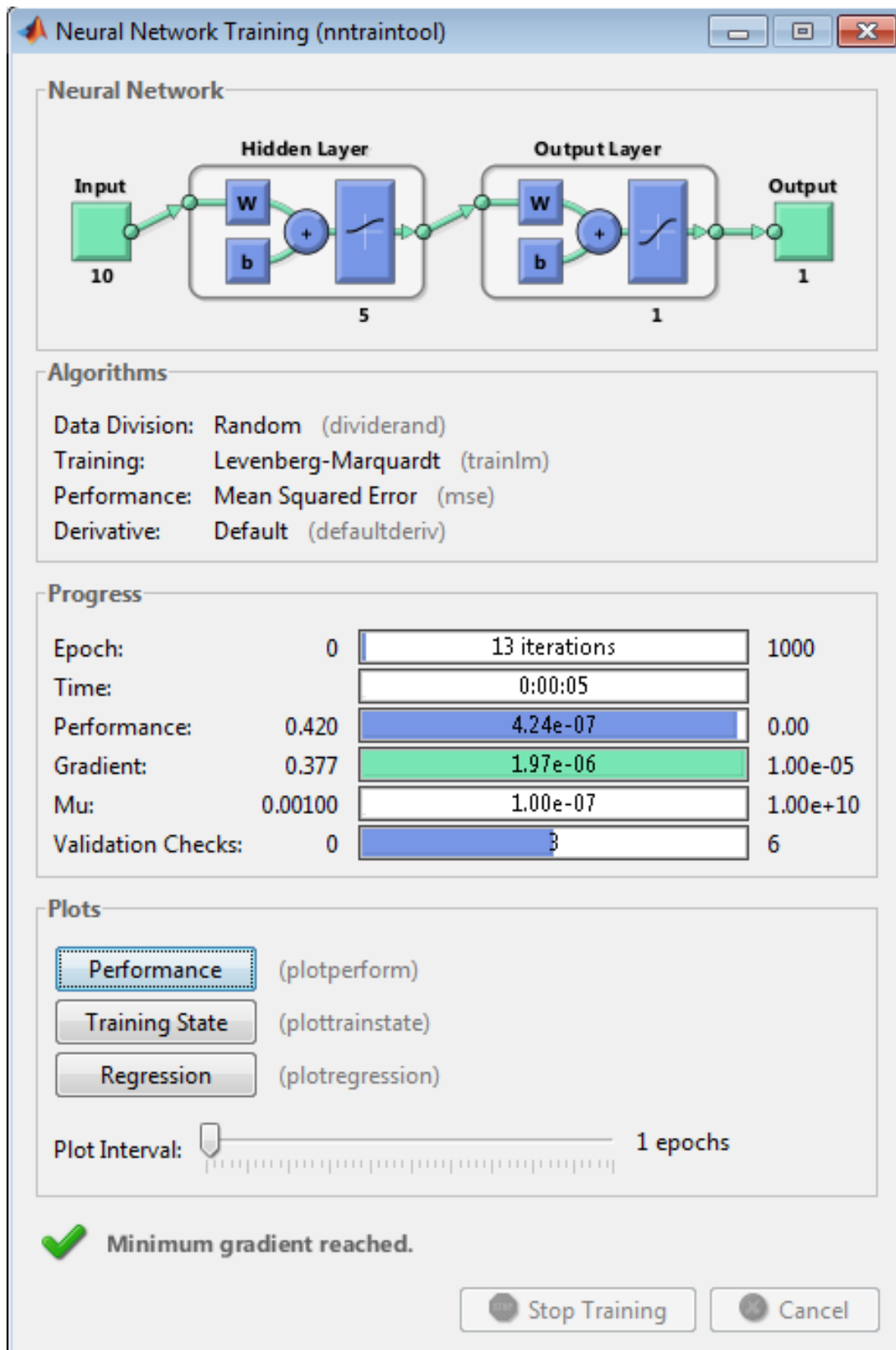


Рисунок 2.11 - Вікно процесу навчання для багатошарового персептрона з десятима входами, одним виходом та п'ятьма нейронами у прихованому шарі

Таблиця 2.3 – Результати тестування багатошарового персептрону з десятьма входами, одним виходом та п'ятьма нейронами у прихованому шарі

Вхідні сигнали, X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0			
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1		
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0		
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0		
	5	1	0	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0		
	6	1	1	0	1	1	1	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	
	7	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	1	1	1	0	0	0
	8	1	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0	1	1	0	0	1	0	0	0
	9	1	1	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0	1	0	1	0	0	0	0
	10	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0	1	0	0	0	0	0
Y		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
Y1		0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	

В результаті одношаровий персептрон не зробив жодної помилки, всі 25 поданих на вхід образу були вірно класифіковані. Багатошаровий персептрон з десятьма входами, одним виходом і трьома нейронами в прихованому шарі зробив 10 помилок, а багатошаровий персептрон з десятьма входами, одним виходом і п'ятьма нейронами в прихованому шарі зробив 1 помилку з 25 поданих на вхід образу.

Проведений аналіз показав, що завдання класифікації тестових завдань за безліччю вхідних параметрів може бути адекватно вирішена нейронною мережею прямого розповсюдження інформації (персептроном). Однак складності при використанні такого підходу можуть бути у виборі архітектури НМ та підготовці репрезентативної навчальної вибірки для налаштування НМ. Таким чином, для системи класифікації тестових завдань з оцінки якості вибираємо персептрон із десятьма входами, одним виходом з наступними ваговими коефіцієнтами $w = [11;9;8;5;1;1;1;1;1;1]$ та зміщенням $b = -37$, отриманими в результаті навчання НМ.

3 РОЗРОБКА СИСТЕМИ КЛАСИФІКАЦІЇ ТЕСТОВИХ ЗАДАнь

3.1 Система класифікації тестових завдань

В результаті дослідження можливостей НМ, описаних у другому розділі, система класифікації тестових завдань за оцінкою якості має наступний алгоритм роботи:

- опрацьовуємо результати опитування, які представлені у вигляді двовимірної матриці X , де кількість рядків відповідає показникам якості і дорівнює десяти, а кількість стовпців відповідає кількості студентів (i), які оцінили це тестове завдання;

- подаємо на вхід персептрона перший стовпець матриці X і на виході отримуємо першу оцінку якості тестового завдання q_1 :

$$q = X^{<i>T} * w^T + b \quad (3.1)$$

де w – вагові коефіцієнти персептрону, $w = [11; 9; 8; 5; 1; 1; 1; 1; 1; 1]$;

b – зсув персептрону, $b = -37$;

- подаємо наступний стовпець матриці X та на виході отримуємо наступну оцінку якості Q_i , в результаті отримуємо кількість оцінок якості тестового завдання, що відповідає кількості експертів;

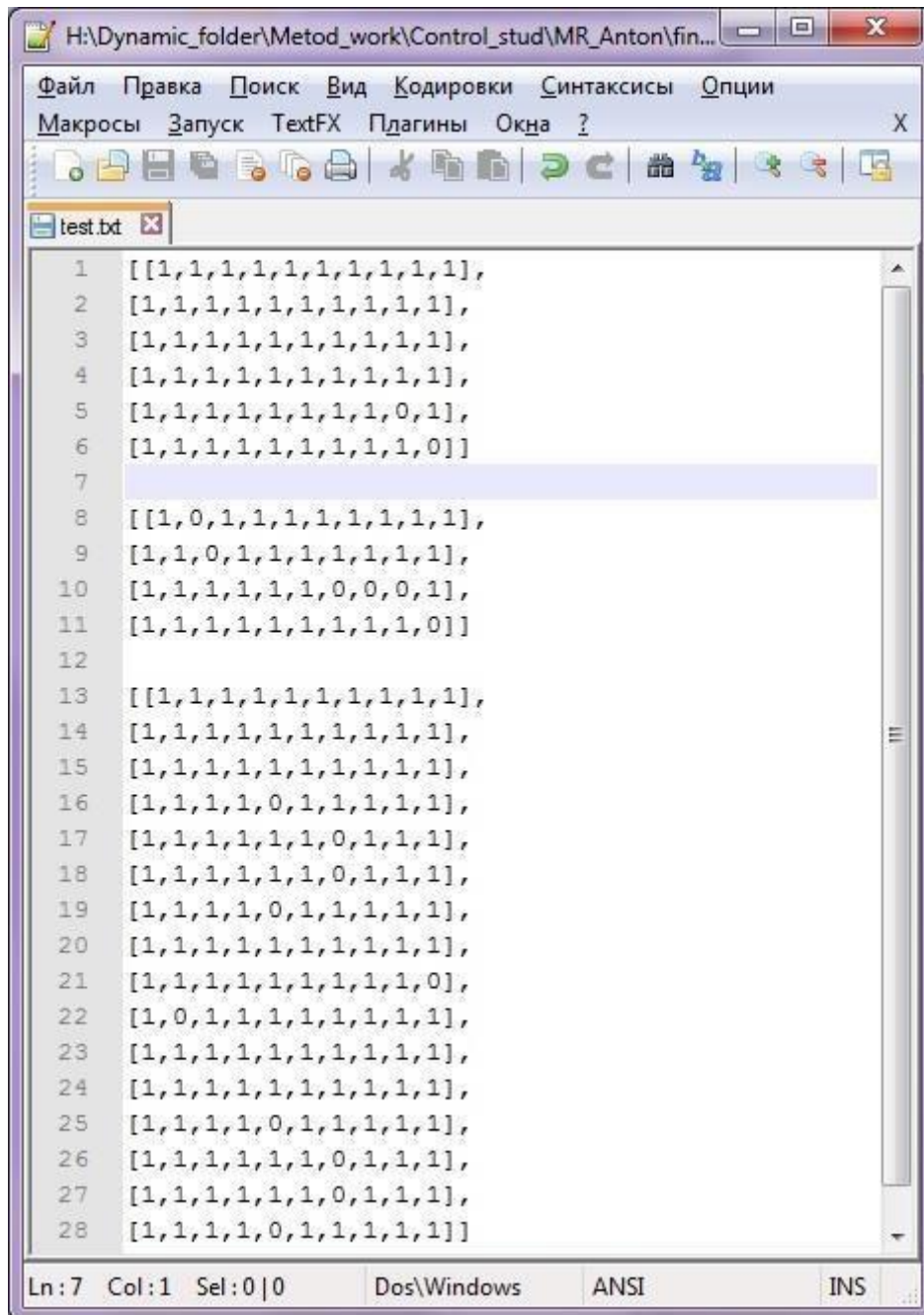
- обчислюємо підсумкову оцінку якості тестового завдання:

$$Q = \frac{q_i}{i} \quad (3.2)$$

Система класифікації тестових завдань з оцінки якості розроблена із застосуванням мови програмування C++ за допомогою інструментарію розробки Qt framework для графічного інтерфейсу, з використанням

інтегрованого середовища розробки Qt Creator. На рисунку 3.1 представлені тестові вхідні дані для системи, рисунку 3.2 - робоче вікно системи класифікації тестових завдань з оцінки якості.

Програмна реалізація розробленої системи представлена у додатку Б.



```
1 [[1,1,1,1,1,1,1,1,1,1],
2 [1,1,1,1,1,1,1,1,1,1],
3 [1,1,1,1,1,1,1,1,1,1],
4 [1,1,1,1,1,1,1,1,1,1],
5 [1,1,1,1,1,1,1,1,0,1],
6 [1,1,1,1,1,1,1,1,1,0]]
7
8 [[1,0,1,1,1,1,1,1,1,1],
9 [1,1,0,1,1,1,1,1,1,1],
10 [1,1,1,1,1,1,0,0,0,1],
11 [1,1,1,1,1,1,1,1,1,0]]
12
13 [[1,1,1,1,1,1,1,1,1,1],
14 [1,1,1,1,1,1,1,1,1,1],
15 [1,1,1,1,1,1,1,1,1,1],
16 [1,1,1,1,0,1,1,1,1,1],
17 [1,1,1,1,1,1,0,1,1,1],
18 [1,1,1,1,1,1,0,1,1,1],
19 [1,1,1,1,0,1,1,1,1,1],
20 [1,1,1,1,1,1,1,1,1,1],
21 [1,1,1,1,1,1,1,1,1,0],
22 [1,0,1,1,1,1,1,1,1,1],
23 [1,1,1,1,1,1,1,1,1,1],
24 [1,1,1,1,1,1,1,1,1,1],
25 [1,1,1,1,0,1,1,1,1,1],
26 [1,1,1,1,1,1,0,1,1,1],
27 [1,1,1,1,1,1,0,1,1,1],
28 [1,1,1,1,0,1,1,1,1,1]]
```

Рисунок 3.1 – Тестові вхідні дані для системи класифікації тестових завдань з оцінки якості

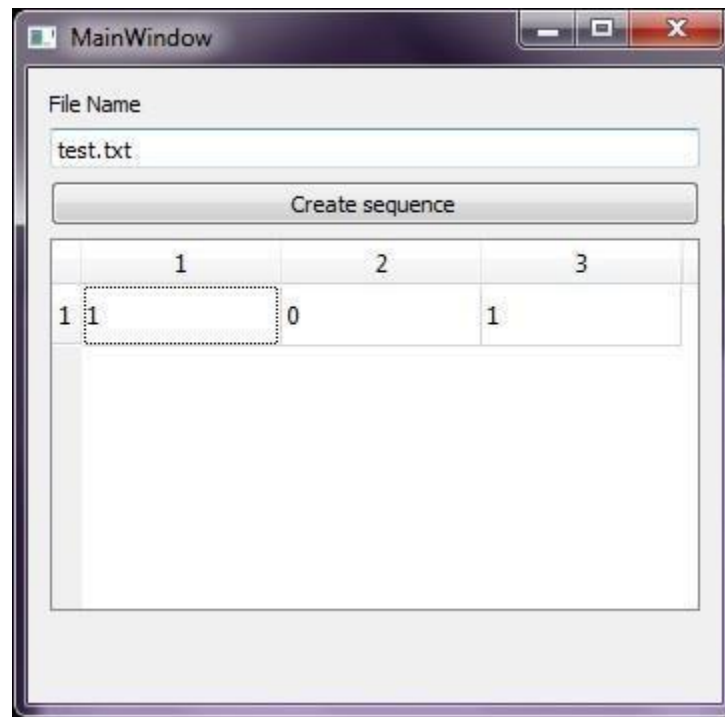


Рисунок 3.2 – Робоче вікно системи класифікації тестових завдань з оцінки якості

3.2 Обробка результатів розробленої системи

Для тестування системи класифікації тестових завдань з оцінки якості було проведено опитування студентів третього курсу, які пройшли тестування з дисципліни «Комп'ютерна логіка» та отримали високий чи достатній рівень знань, де студенти, які проходили тестування, виступали в ролі експертів та визначали десять показників якості за кожним тестовим завданням, перелік яких наведено у п. 1.2. У тестуванні брало участь 25 студентів, 16 із них отримали високий чи достатній рівень знань. Ці 16 студентів визначали показники якості тільки тестових завдань, на які дали правильні відповіді. Тестування складалося з десяти тестових завдань, і в результаті було сформовано десять вхідних матриць X за кожним тестовим завданням відповідно (табл. 3.1).

Таблиця 3.1 – Результати опитування студентів

[illegible]

Продовження таблиці 3.1

№ тест. завдан-ня	Показ-ники якості	Студенти															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-
	5	1	1	1	0	1	1	0	1	1	1	1	1	1	1	-	-
	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-
	7	0	1	1	1	0	1	1	0	1	1	1	1	1	1	-	-
	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-
	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-
	10	1	1	1	1	1	1	1	1	1	1	1	1	1	0	-	-
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
	4	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
	5	1	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1
	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	7	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1
	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	0	0	0	1	0	0	1	0	0	1	0	-	-	-	-	-
	2	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-
	3	1	1	1	1	1	1	0	1	1	1	1	-	-	-	-	-
	4	1	1	1	1	1	1	1	0	1	1	1	-	-	-	-	-
	5	1	0	1	1	0	1	1	1	1	1	0	-	-	-	-	-
	6	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-
	7	1	1	0	1	1	1	1	1	1	1	1	-	-	-	-	-
	8	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-
	9	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-
	10	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-
	5	1	1	1	0	1	1	1	1	1	0	1	0	1	-	-	-
	6	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-
	7	1	1	1	1	0	1	1	1	1	1	1	1	1	-	-	-
	8	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-
	9	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-
	10	1	1	1	1	1	1	1	1	1	1	1	1	0	-	-	-

Продовження таблиці 3.1

№ тест. завдан-ня	Показ-ники якості	Студенти															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	10	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-
	2	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-
	3	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-
	4	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-
	5	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-
	6	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-
	7	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-
	8	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-
	9	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-
	10	1	1	0	1	0	0	0	0	0	0	0	0	-	-	-	-

В результаті роботи системи отримали оцінки якості десяти тестових завдань з дисципліни "Комп'ютерна логіка", показані в таблиці 3.2.

Таблиця 3.2 - Результати роботи системи класифікації тестових завдань

	Номер тестового завдання									
	1	2	3	4	5	6	7	8	9	10
Q	1	1	0	1	1	1	0	1	1	1

За результатами роботи розробленої системи видно, що вісім тестових завдань із десяти класифіковані за високою оцінкою якості, а два (3 та 7) за низькою. Третє завдання не чітко сформульовано і мало однозначне формулювання питання, сьоме завдання містило у формулюванні питання багато зайвих слів, що й вплинуло на низьку оцінку якості тестового завдання. Відповідно низька якість тестового завдання відобразилась на оцінці рівня знань студентів. З 16 студентів, які здобули високий чи достатній рівень знань, дали правильну відповідь на третє тестове завдання 9 осіб, на сьоме – 11 осіб.

Дев'яте та десяте тестові завдання показали високу оцінку якості, але, як видно з таблиці 3.1, не відповідають рівню складності тестового завдання. Так як на дев'яте запитання всі студенти дали правильну відповідь, можна зробити висновок, що не складне тестове завдання потрапило до завдань високого рівня складності. На десяте завдання правильно відповіли 12 студентів із 16, і можна зробити висновок, що складне завдання потрапило до завдань низького рівня складності. Невідповідність тестових завдань рівням складності також призводять до не зовсім точних показників рівня знань студентів.

За підсумками тестування розробленої системи класифікації тестових завдань з оцінки якості можна зробити висновок, що це система працює правильно, але вимагає доопрацювання по класифікації тестових завдань за рівнями складності, так як це впливає на оцінку якості рівня знань студентів.

ВИСНОВКИ

При виконанні магістерської роботи проведено аналіз основних форм та принципів композиції тестових завдань. Також проведено аналіз основних показників якості тестових завдань, на підставі якого були сформовані характеристики, за якими НМ вирішуватиме завдання класифікації з оцінки якості тестових завдань:

- стислість (лаконічність) формулювання завдання;
- стислість (лаконічність) формулювання варіантів відповідей;
- точність (однозначність) формулювання завдання;
- точність (однозначність) формулювання варіантів відповідей;
- ясність інструкції до завдання;
- коректність форми завдання;
- достатність необхідної кількості варіантів відповідей;
- відповідність формулювання варіантів відповідей завдання;
- відповідність завдання навчальній дисципліні, розділу, темі;
- відповідність до рівня складності завдання.

З перерахованих показників якості перші чотири найбільш важливі і якщо хоча б одне з них не відповідає вимогам, тестове завдання вважається завданням низької якості.

Проведено аналіз існуючих методів оцінки якості тестових завдань. Визначити якість тестових завдань можна автоматично в ході статистичної обробки результатів тестування та за допомогою оцінки експертів.

Недоліком статистичної обробки результатів тестування і те, що вихідний тестовий матеріал необхідно випробувати досить великий вибірці учнів (щонайменше 150-200 людей). Зазвичай це накопичення даних протягом 3-5 років. За цей період можуть змінюватись умови та середовище навчання, що впливає на статистичні показники результатів тестування.

При використанні експертних методів однією з основних проблем є об'єктивність оцінки якості тестових завдань, компетентність експертів та наявність таких.

Розроблено метод оцінки якості тестових завдань на базі нейронних мереж. У ході дослідження було спроектовано три нейронні мережі, проведено навчання та тестування. За результатами тестування було обрано одношаровий персептрон, на базі якого розроблено та реалізовано на практиці систему класифікації тестових завдань з оцінки якості. Система була протестована і за підсумками тестування, розробленої системи класифікації тестових завдань з оцінки якості, можна дійти невтішного висновку, що це система працює правильно, але вимагає доопрацювання по класифікації тестових завдань за рівнями складності, т.к. це впливає оцінку якості рівня знань студентів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Білан Л. В. Сучасні методики контролю якості навчання студентів / Л. В. Білан, Ю. В. Гуцуляк // Науковий вісник нац. університету біоресурсів і природокористування України. – 2010. – N146. – С. 258-262.
2. Психологія діяльності та навчальний менеджмент: Навч. посіб. / За заг. ред. М.В. Артющиної. – К.: КНЕУ імені Вадима Гетьмана, 2007. – 366 с. URL: <https://studfile.net/preview/8839472/page:73>.
3. Форми тестових завдань. Форма подання тестового завдання. URL: https://pidru4niki.com/12590605/informatika/formi_testovih_zavdan_forma_podannya_testovogo_zavdannya
4. Moodle – безкоштовна система управління дистанційними курсами. URL: <http://moodle.org/>.
5. Система дистанційного навчання НУ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА» URL: <https://moodle.zp.edu.ua/>.
6. Олейник Н.М. Тест как инструмент измерения уровня знаний и трудности заданий в современной технологии обучения: Учеб. пособие по спец. Курсу / Н.М. Олейник. – Донецк: ДонГУ. – 1991. – 168 с.
7. Крислов В.А. Методика анализа педагогических тестов по результатам тестирования / В.А. Крислов, Т.В. Онищенко, Н.В. Русинова // Труды Одесского политехнического университета. – 2007. – № 2(28). – С. 198-203.
8. Крислов В.А. Технология оценки и повышения репрезентативности педагогического теста на базе статистических характеристик учебного материала / В.А. Крислов, Е.А. Городничая, Т.В. Онищенко, НгокХиуВу // Электротехнические и компьютерные системы. – 2014. – № 13(89). – С. 99-104.
9. Напрасник С.В., Цимбалюк Е.С., Шкиль А.С. Компьютерная система тестирования знаний OpenTEST 2.0 // Сборник научных трудов 10-й

международной конференции УАДО Образование и виртуальность – 2006. – Харьков-Ялта.– 2006.– Харьков: ХНУРЭ.– 2006.– С. 454–461.

10. Сервісів запитань Mentimeter. URL: <http://mentimeter.com/>.

11. Мобільні опитування mQlicker. URL: <https://mqlicker.com/>.

12. Сервіс Kahoot: мобільні опитування. URL: <https://getkahoot.com/>.

13. Robert Callan. The Essence of Neural Networks (The Essence of Computing Series). – 1998. – 248p. – ISBN 978-0139087325.

14. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применения / Е.В. Бодянский, О.Г. Руденко. – Х.: ТЕЛІТЕХ, – 2004. – 372с.

Таблиця А.1- Вхідні сигнали навчання НМ та відповідні їм очікувані результати

[illegible]

Продовження таблиця А.1

Вхідні сигнали, X	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
	4	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1
	5	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
	6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
	7	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0
	8	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0
	9	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
	10	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Вхідні сигнали, X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1
	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
	6	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0
	7	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1
	8	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	1	0	0	0	0
	9	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0
	10	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Вхідні сигнали, X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
	4	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
	5	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
	6	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1
	7	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0	0	1	0	0	0
	8	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0
	9	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0
	10	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	0	0	1	0
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Вхідні сигнали, X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	5	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
	6	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0
	7	1	0	0	0	1	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	1
	8	0	1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	1
	9	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0	0	1	0	0	1	0	1
	10	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	1	0
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Продовження таблиця А.1

Вхідні сигнали, X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	4	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	5	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
	6	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1
	7	1	0	0	0	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0	1	1
	8	0	1	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	1
	9	0	1	0	1	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	1	1	0	1	1	1
	10	1	0	1	1	0	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	0	1	1	1	1
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Вхідні сигнали, X	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
	3	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	
	4	1	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	
	5	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	
	6	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	
	7	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	
	8	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	
	9	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	
	10	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Вхідні сигнали, X	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	
	3	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	
	4	1	0	0	0	0	0	0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0	0	
	5	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	
	6	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	
	7	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	0	
	8	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	
	9	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	
	10	0	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Вхідні сигнали, X	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	
	4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	
	5	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	
	6	1	1	1	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	
	7	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	0	1	1	1	
	8	0	1	1	1	1	0	1	1	1	0	1	1	0	1	1	0	0	1	1	1	1	0	1	1	
	9	1	0	1	1	1	1	0	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	0	1	
	10	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0	1	0	0	1	1	1	1	1	0	
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Додаток Б

ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНОЇ СИСТЕМИ

Лістинг Б.1 - Файл "perceptron.h"

```

#ifndef PERCEPTRON_H
#define PERCEPTRON_H
#include <qdebug.h>
#include <QStringList>
#include <vector>

//data from trained NN
static const int vec_size = 10;
static const int weights[vec_size] = {11,9,8,5,1,1,1,1,1,1};
static const int offset = -37;

typedef std::vector<int> quality_t;
typedef std::vector<quality_t> tz_t;

class perceptron
{
public:
    void read_file(const QString& filename );
    void process_strings();
    std::vector<int>& process_data();
private:
    QStringList m_lst;
    std::vector<tz_t> m_data;
    std::vector<int> m_tz_results;

    int iteration(int* res);
};

#endif // PERCEPTRON_H

```

Файл "perceptron.cpp"

```

#include "perceptron.h"
#include <QFile>
#include <QTextStream>

int perceptron::iteration(int* input)
{
    int output = 0;
    for( int i = 0; i < vec_size; ++i)
    {
        output+= input[i]*weights[i];
    }
    output+=offset;
    return output < 0 ? 0 : 1;
}

```

```

}

void perceptron::read_file(const QString& filename )
{
    QFile file(filename);
    if (!file.open(QIODevice::ReadOnly | QIODevice::Text))
        throw std::exception("bad file");
    QTextStream stream(&file);
    QString line;
    if( !m_lst.isEmpty() )
        m_lst.clear();
    do {
        line = stream.readLine();
        if( !line.isNull() )
            m_lst.append(line);
    } while (!line.isNull());
}

void perceptron::process_strings()
{
    tz_t tz;
    bool need_push = false;
    for(QStringList::iterator it = m_lst.begin(); it !=
m_lst.end(); ++it)
    {
        if(*it == QString() )
            continue;
        if(it->startsWith("[("))
        {
            it->remove(0,2);
        }
        else
        {
            it->remove(0,1);
        }
        need_push = it->endsWith("]]");

        it->remove(it->indexOf("]"),2);

        QStringList x_input = it->split(",");
        Q_ASSERT(x_input.size() != vec_size);
        quality_t quality;
        for(QStringList::iterator x_it = x_input.begin(); x_it !=
x_input.end(); ++x_it)
        {
            quality.push_back(x_it->toInt());
        }
        tz.push_back(quality);

        if(need_push)
        {
            m_data.push_back(tz);
            tz.clear();
        }
    }
}

```

```

        need_push = false;
    }
}

std::vector<int> &perceptron::process_data()
{
    for(std::vector<tz_t>::iterator tz_it = m_data.begin(); tz_it
!= m_data.end(); ++tz_it )
    {
        int tz_result = 0;
        for(std::vector<quality_t>::iterator qual_it = tz_it-
>begin(); qual_it != tz_it->end(); ++qual_it )
        {
            tz_result += iteration(qual_it->data());
        }
        tz_result = tz_result - tz_it->size()/2;
        m_tz_results.push_back(tz_result<0? 0: 1);
    }
    return m_tz_results;
}

```

Лістинг Б.2 - Файл “mainwindow.h”

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

private:
    Ui::MainWindow *ui;

public slots:
    void OnPushButton();
};

#endif // MAINWINDOW_H

Файл “mainwindow.cpp”

#include "mainwindow.h"
#include "ui_mainwindow.h"

```

```

#include "perceptron.h"
#include <qdebug.h>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    connect(ui->pushButton, SIGNAL(clicked()), this,
    SLOT(OnPushButton())) );
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::OnPushButton()
{
    perceptron rd;
    rd.read_file(ui->fileName->text());
    rd.process_strings();
    std::vector<int>& results = rd.process_data();
    ui->tableWidget->setRowCount(1);
    ui->tableWidget->setColumnCount(results.size());
    for(int i = 0; i < results.size(); ++i)
    {
        QTableWidgetItem *newItem = new
QTableWidgetItem(tr("%1").arg(results[i]));
        ui->tableWidget->setItem(0, i, newItem);
    }
}
Файл "main.cpp"

#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}

```