

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра комп'ютерних інтелектуальних технологій та систем

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Нейромережеве розпізнавання тексту на зображеннях

Виконав:
студент

II курсу, групи КІТм-19-1
Ілляшенко С.А.

Спеціальність

123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми
Освітня програма

Освітньо-професійна

Комп'ютерні інтелектуальні технології
(повна назва освітньої програми)

Керівник:

проф. Безсонов О.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТС

Руденко О.Г.
(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
Кафедра _____ електронних обчислювальних машин
Рівень вищої освіти _____ другий (магістерський)
Спеціальність (напрямок) _____ 123 – Комп'ютерна інженерія
(код і назва)
Освітня програма _____ Комп'ютерні інтелектуальні технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Ілляшенку Святославу Анатолійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Нейромережеве розпізнавання тексту на зображеннях

затверджена наказом по університету від “ 11 ” листопада 2020 р. № 1582 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 10 грудня 2020 р.

3. Вхідні дані до роботи _____

Розпізнавання тексту на зображеннях

Мова програмування Delphi

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз проблемної області і постановка задачі

Проектування та розробка алгоритму розпізнавання зображень

Програмна реалізація модуля для розпізнавання тексту на зображеннях

Тестування

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайдів 10

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд стану проблеми та постановка задачі	11.11 – 13.11	
2	Аналіз літератури за напрямком	13.11 – 20.11	
3	Існуючі способи вирішення проблеми	21.11 – 23.11	
4	Розробка нейромережевого алгоритму	24.11 – 26.11	
5	Програмна реалізація	26.11 – 30.12	
6	Оформлення пояснювальної записки	01.12 – 07.12	
7	Оформлення графічної частини	08.12 – 10.12	

Дата видачі завдання 11 листопада 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф.Бессонов О.О.

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: LXX с., XXVII рис., III табл., XX дод., IX джерел.

РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, АНАЛІЗ ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ,

Метою атестаційної роботи є нейромережеве розпізнавання тексту на зображеннях.

У ході виконання атестаційної роботи запропоновано програмний комплекс з використанням нейронних мереж, що значно покращує якість та швидкість функції розпізнавання тексту з відсканованих документів.

Розроблено програмний продукт з використанням мови програмування Delphi версії 7 фірми Enterprise (Borland), яка надає найбільш широкі можливості для програмування додатків ОС Windows.

ABSTRACT

Explanatory note of attestation work: XX pages, XXVII figures, III tables, XX appendices, IX sources.

IMAGE RECOGNITION, IMAGE PROCESSING, MACHINE LEARNING

The major goal of this thesis is recognition of text on images using neural networks.

In order to do this, was proposed software package using neural networks, which significantly improves the quality and speed of the text recognition function from scanned documents.

A software product has been released from the list of programs Delphi version 7 from Enterprise (Borland), as we have the most wide range of possibilities for software additions to Windows.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	12
ВСТУП	13
1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ СИСТЕМИ РОЗПІЗНАВАННЯ ТЕКСТУ	16
1.1 Суть технічної проблеми	16
1.2 Існуючі способи вирішення проблеми	18
1.2.1 Розпізнавання по шаблонах	18
1.2.2 Структурний підхід	18
1.2.3 Контекстне розпізнавання	19
1.2.4 Нейронні мережі в системі розпізнавання	19
1.2.5 Біологічний прототип	20
1.2.6 Штучний нейрон	21
1.2.7 Одношарові штучні нейронні мережі	26
1.2.8 Багатошарові штучні нейронні мережі	27
1.2.9 Нейронні мережі Хопфілда і Хеммінга	29
2 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РОСПІЗНАВАННЯ ЗОБРАЖЕНЬ ІЗ ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ	35
2.1 Проектування програмного модуля розпізнавання зображень із застосуванням нейронних мереж	35
2.1.1 Розробка алгоритму функціонування мережі Хеммінга	36
2.1.2 Розробка алгоритму навчання штучної нейроної мережі	38
2.1.3 Розробка алгоритму порівняння зображень з використанням відстані Хеммінга	40
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	44
3.1 Програмна реалізація модуля для розпізнавання зображень	44
3.1.1 Обґрунтування вибору мови програмування	44

3.1.2 Розробка програмного продукту.....	45
4 ТЕСТУВАННЯ	49
4.1 Тестування приклад роботи програми та аналіз результатів	49
ВИСНОВКИ	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	62
ДОДАТОК А Графічний матеріал атестаційної роботи	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

ПК – персональний комп'ютер

НМ – нейронна мережа

ОС – операційна система

ПЗ – програмне забезпечення

ЗНМ – згорткова нейронна мережа

БНМ – багатошарова нейронна мережа

CNN – Convolutional Neural Network – згорткова нейронна мережа

ВСТУП

Обчислювальні системи існують вже не один десяток років. Метою їх створення була можливість замінити людину, зробити за нього трудомістку роботу, що вимагає складних обчислень.

Одне з таких завдань полягає в тому, як навчити комп'ютер розпізнавати образи (зокрема, який алгоритм необхідно створити, щоб комп'ютер міг сприймати зображення текстів).

Розпізнавання текстів - дуже складне завдання з теоретичної і практичної точок зору. Людина, наприклад, залучає для цього весь комплекс знань і досвіду. Він визначає текст з сукупності сигналів органів почуттів, виділяє кожен символ, виділяє характерні ознаки символів і на підставі свого досвіду приходить до висновку про значення символу і всього тексту в цілому.

Комп'ютер помиляється в процесі розпізнавання набагато частіше людини. Сьогодні не існує абсолютно точного методу визначення тексту і символу по їх зображенню. Чимало розроблених комерційних проектів використовують свої запатентовані методи і не можуть похвалитися ідеальним рішенням завдання.

У багатьох випадках хороше рішення дає комплексний підхід до задачі. Сама задача розпізнавання тексту поділяється на підзавдання: фільтрація зображення від шуму, виділення зображень символів з зображення тексту, виділення ознак символів і порівняння цих ознак із збереженими зразками. Кожна задача містить багато варіантів рішень, з яких лише деякі є більш-менш оптимальними.

Можливість розпізнавання спирається на схожість однотипних об'єктів. Незважаючи на те, що всі предмети і ситуації унікальні в строгому сенсі, між деякими з них завжди можна знайти подібності з того чи іншою ознакою. Звідси виникає поняття класифікації - розбиття всієї множини об'єктів на

непересічні підмножини - класи, елементи яких мають деякі схожі властивості, що відрізняють їх від елементів інших класів. І, таким чином, завданням розпізнавання є віднесення розглянутих об'єктів або явищ за їх опису до потрібних класам.

Прикладами сфер застосування розпізнавання можуть служити системи розпізнавання тексту, машинний зір, розпізнавання мови, відбитків пальців та інше. Незважаючи на те, що деякі з цих завдань вирішуються людиною на підсвідомому рівні з великою швидкістю, до теперішнього часу ще не створено комп'ютерних програм, вирішальних їх у настільки ж загальному вигляді.

Суть НМ полягає в тому, що мережа складається з елементів, котрі називаються формальними нейронами (formal neuron). Кожен нейрон приймає набір сигналів, що надходять на його входи від одної групи таких же нейронів, обробляє сигнали з врахуванням попередніх сигналів і адаптації до них на основі процедур навчання і передає результати обробки другій групі нейронів. Зв'язки між нейронами кодуються вагами, що відображають важливість їх інформації для визначення загального результату. Основний принцип настроювання нейронної мережі полягає в застосуванні процедур оптимізації та адаптації на основі певних критеріїв, здатності до перенавчання. Однією з переваг НМ є те, що всі елементи можуть функціонувати паралельно, тим самим істотно підвищуючи ефективність розв'язання задач, особливо при обробці зображень в реальному часі. Системи розпізнавання об'єктів зображення, що засновані на нейронних мережах, використовують ієрархічну архітектуру.

Мета дослідження – розробка програмного комплексу, основною функціональністю якого є розпізнавання тексту з відсканованих документів та застосування нейронних мережі Хемінга, що значно покращує якість та швидкість функції розпізнавання.

Об'єкт дослідження – процеси розпізнавання тексту з відсканованих документів.

Предмет дослідження – засоби та методи розпізнавання зображень.

Задачею дослідження є розробка інтелектуальної система із застосуванням нейронних мереж для розпізнавання зображень з текстом.

1. ТЕХНІКО – ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ СИСТЕМИ РОЗПІЗНАВАННЯ ТЕКСТУ

1.1 Суть технічної проблеми

Розпізнавання тексту є одним з напрямків розпізнавання зображень. Розпізнавання зображень являє собою дуже складну задачу в теоретичному і практичному сенсах, незважаючи на те, що з нею досить легко справляються чимало живих організмів і людина. Вкрай складно створити штучну систему і її технічно реалізувати для того, щоб ефективно виконувати цей процес. У даному випадку, під розпізнаванням розуміється співвіднесення зображення об'єкта, його образу, набору ознак самому об'єкту.

Прикладами і додатками систем розпізнавання образів можуть бути як розпізнавання тексту загалом, так і окремих його символів, розпізнавання мови, людських осіб, біометричних даних людини, штрих-кодів продуктів, номерів машин і т.д.

Прикладами розпізнавання тексту є: оцифровка зображень тексту (скановані книги, статті, журнали) для наступної роботи з його цифровим аналогом, обробка анкетних бланків, розпізнавання номерів машин і написів на об'єктах і т.д. Завдання розпізнавання тексту залишається актуальною на сьогоднішній день, так як не існує стовідсоткової універсальної системи з розпізнавання тексту. Система розпізнавання тексту передбачає наявність на вході зображення з текстом (у форматі даних графічного файлу). На виході системи повинен сформуватися текст, виділений з цього зображення.

Розпізнавання тексту включає в себе наступні підзадачі і підпроцеси:

- що поступає на вхід системи зображення повинно бути очищено від шуму і приведено до виду, що дозволяє ефективно виділяти символи і розпізнавати їх;

- система повинна розбити зображення на блоки тексту, ґрунтуючись на особливостях його вирівнювання та розподілу по декількох колонках;
- зображення з текстом має бути розділене на зображення рядків, а потім на зображення символів для того, щоб надалі обробити окремо кожен символ. Після даного кроку різні системи розпізнавання працюють за своїми специфічними алгоритмами;
- зображення символу може оброблятися цілком, для цього воно порівнюється з наявними шаблонами. Іншим варіантом є виділення характеристик зображуваного символу: відбір характерних ознак, і класифікація даних ознак за наявними в системі критеріям;
- на виході четвертого кроку з'являється можливий варіант літери. Проте зазвичай системи на цьому не зупиняються і продовжують роботу на основі інших методів, уточнюючи отриманий результат;
- результат розпізнавання може бути не задовільним. Для отримання більш хороших результатів у системі може бути вбудований блок навчання. За допомогою цього блоку можна задати системі приклади накреслення різних букв в даному шрифті. Після процесу навчання передбачається краща якість розпізнавання тексту;

Система розпізнавання тексту не завжди повинна слідувати всім описаним крокам, але основні дії процесу розпізнавання є загальними для будь-якого алгоритму.

Існує кілька основних систем розпізнавання тексту. Всі вони є комерційними продуктами, і багато внутрішні алгоритми їх роботи приховані від загального доступу. Принцип роботи подібних систем базується на кількох стратегіях, але, часто, алгоритм розпізнавання в найзагальнішому вигляді полягає у послідовному висуванні і перевірці гіпотез. При цьому порядок їх висунення управляється закладеними в програму знаннями про досліджуваний предмет і результатами перевірки попередніх гіпотез.

1.2 Існуючі способи вирішення проблеми.

1.2.1 Розпізнавання по шаблонах

Програмне забезпечення OCR (Optical Character Recognition оптичне розпізнавання символів) зазвичай працює з великим растровим зображенням сторінки з сканера. При цьому більшість систем має шаблони, створені для різних накреслень. Після кількох розпізнаних слів, програмне забезпечення визначає використовуваний шрифт і шукає відповідні пари тільки для цього шрифту. У деяких випадках програмне забезпечення використовує чисельні значення частин символу (пропорцій), щоб визначити новий шрифт. Це може поліпшити ефективність розпізнавання [4].

Прикладом, програма розпізнавання TypeReader використовує машинно- залежні алгоритми на основі шаблонного підходу.

Даний підхід вимагає створення шаблону для кожного шрифту. Програма TypeReader використовує 2100 різних варіантів накреслень символів.

1.2.2 Структурний підхід

Дуже великий попит у всьому світі має система OCR-Caere Omni Page Professional використовує алгоритм, заснований на знаходженні загальних специфічних особливостей символів.

Ця система містить 100 різних алгоритмів для ідентифікації 100 різних символів: верхнього та нижнього регістру від «А» до «Z», запису чисел і символів пунктуації. Кожен з цих алгоритмів шукає «особливості» накреслень типу «острівів», «півострівів», точок, прямих відбитків і дуг. Експертні системи також розглядають горизонтальні і вертикальні проекції відбитків букви і звертають увагу на основні особливості у створених кривих, підсумовуючи в них число темних пікселів. На жаль, нечіткий текст може

стати специфічною проблемою для цих структурних алгоритмів, так як відсутній піксель може розбивати довгий штрих або криву, а додаткове пляма бруду може закривати петлю.

1.2.3 Контекстне розпізнавання

У програмне забезпечення системи OCR часто включається словники для допомоги алгоритмам розпізнавання. Словники надають довідки в багатьох випадках, але швидко відмовляють, коли, наприклад, мають справу з власними іменами, які не перебувають у словнику. Цей ефект особливо помітний у російській програмі FineReader, який частіше, ніж у середньому по всім символам, помиляється в словах, які відсутній в його словнику.

1.2.4 Нейронні мережі в системі розпізнавання

Нейронні мережі - це структура пов'язаних елементів, на яких задані функції перетворення сигналу, а також коефіцієнти, які можуть бути налаштовані на певний характер роботи. Частина елементів структури виділені як вхідні: на них надходять сигнали ззовні, частина-вихідні: вони формують результуючі сигнали. Сигнал, який проходить через нейронну мережу, перетворюється відповідно до формул на елементах мережі, і на виході формується відповідь.

Нейронна мережа може служити в системі розпізнавання тексту в якості класифікатора. Цей класифікатор можна навчати, налаштовуючи коефіцієнти на елементах мережі, і, таким чином, прагнути до ідеального результату розпізнавання.

Нейронні мережі з успіхом можуть застосовуватися в системах розпізнавання тексту, але існує велика кількість недоліків, які перешкоджають їх широкому застосуванню. Для побудови мережі, що забезпечує розпізнавання кожного символу тексту, необхідно побудувати

досить велику мережу елементів, що призводить до великих витрат пам'яті. Ще сильніше витрачаються ресурси системи в процесі розпізнавання, так як функції на елементах мережі працюють з числами з плаваючою крапкою. Крім цього нейронні мережі необхідно навчати на всі випадки, що, однак, не гарантує точного результату. І, нарешті, робота нейронної мережі з розпізнавання тексту багато в чому залежить від конфігурації мережі функцій, заданих в елементах, що вимагає великих зусиль для побудови ефективно працюючої мережі.

1.2.5 Біологічний прототип

Розвиток штучних нейронних мереж надихається біологією. Тобто, розглядаючи мережеві конфігурації і алгоритми, дослідники застосовують терміни, запозичені з принципів організації мозкової діяльності. Але на цьому аналогія закінчується. Наші знання про роботу мозку настільки обмежені, що мало б знайшлося точно доведених закономірностей для тих, хто побажав би керуватися ними. Тому розробникам мереж доводиться виходити за межі сучасних біологічних знань у пошуках структур, здатних виконувати корисні функції. У багатьох випадках це призводить до необхідності відмови від біологічної правдоподібності, мозок стає просто метафорою, і створюються мережі, неможливі в живій матерії або вимагають неправдоподібно великих допущень про анатомію і функціонування мозку.

Незважаючи на те, що зв'язок з біологією слабка і часто несуттєва, штучні нейронні мережі продовжують порівнювати з мозком. Їх функціонування часто має зовнішню схожість з людським пізнанням, тому важко уникнути цієї аналогії. На жаль, такі порівняння неплодотворні і створюють невиправдані очікування, що неминуче ведуть до розчарування.

Нервова система людини, побудована з елементів, званих нейронами, має приголомшливу складність. Близько 10^{11} нейронів беруть участь в приблизно 10^{15} передавальних зв'язках, що мають довжину метр і більше.

Кожен нейрон володіє багатьма властивостями, спільними з іншими органами тіла, але йому притаманні абсолютно унікальні здібності: приймати, обробляти і передавати електрохімічні сигнали по нервових шляхах, які утворюють комунікаційну систему мозку.

На рисунку 1.1 показана структура пари типових біологічних нейронів. Дендрити йдуть від тіла нервової клітини до інших нейронів, де вони приймають сигнали в точках з'єднання, званих синапсами. Прийняті синапсом входні сигнали передаються до тіла нейрона. Тут вони підсумовуються, причому одні входи прагнуть порушити нейрон, інші - перешкодити його збудженню.

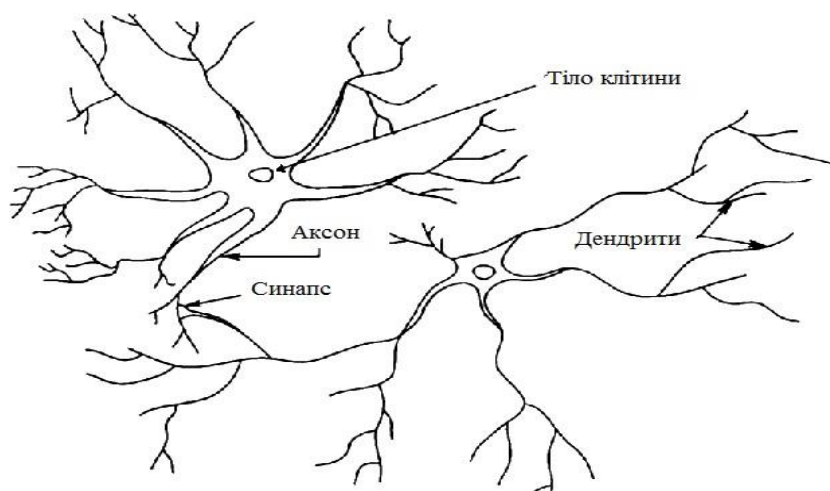


Рисунок 1.1 – Структура пари біологічних нейронів

Коли сумарне збудження в тілі нейрона перевищує деякий поріг, нейрон збуджується, посилаючи по аксону сигнал іншим нейронам. У цієї основної функціональної схеми багато ускладнень і винятків, тим не менше, більшість штучних нейронних мереж моделюють лише ці прості властивості.

1.2.6 Штучний нейрон

Штучний нейрон імітує в першому наближенні властивості біологічного нейрона. На вхід штучного нейрона надходить деяка безліч

сигналів, кожен з яких є виходом іншого нейрона. Кожен вхід множиться на відповідну вагу, аналогічний синаптичній силі, і всі твори сумуються, визначаючи рівень активації нейрона.

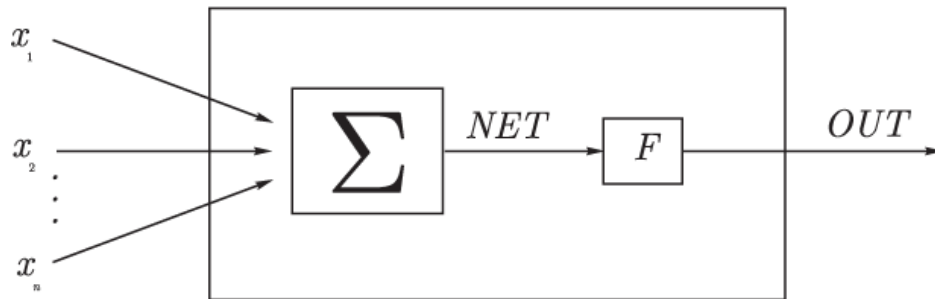


Рисунок 1.2 – Математична модель штучного нейрона

На рисунку 1.2 представлена модель, що реалізує цю ідею. Безліч вхідних сигналів. Ці вхідні сигнали, в сукупності позначаються вектором X , відповідають сигналам, що приходять в синапси біологічного нейрона. Кожен сигнал множиться на відповідну вагу, і надходить на сумуючий блок, позначений Σ . Кожна вага відповідає "силі" однієї біологічної синаптичної зв'язку. (Безліч ваг в сукупності позначається вектором W .) Сумуючий блок, що відповідає тілу біологічного елемента, складає зважені входи алгебраїчно, створюючи вихід, який ми будемо називати NET . У векторних позначеннях це може бути компактно записано таким чином:

$$NET = XW \quad (1.1)$$

Сигнал NET далі, як правило, перетворюється активаційною функцією F і дає вихідний нейронний сигнал OUT . Активаційна функція може бути звичайною лінійною функцією.

$$OUT = F(NET), \quad (1.2)$$

де T - деяка постійна порогова величина, або ж функція, більш точно моделює нелінійну передатну характеристику біологічного нейрона і надає нейронній мережі великі можливості.

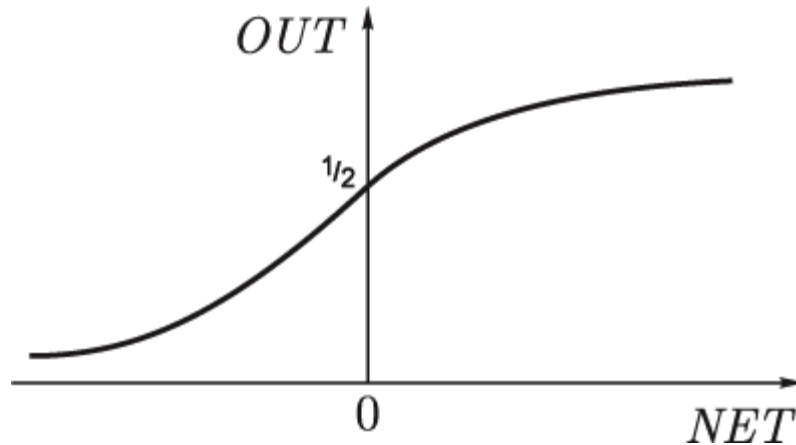


Рисунок 1.3 – Сигмоїдальна функція

На рисунку 1.2 блок, позначений F , приймає сигнал NET і видає сигнал OUT . Якщо блок F звужує діапазон зміни величини NET так, що при будь-яких значеннях NET значення OUT належать деякому кінцевому інтервалу, то F називається "стискаючою" функцією. У якості "стискаючої" функції часто використовується логістична або "сигмоїдальна" (S-подібна) функція, показана на рисунку 1.3. Ця функція математично виражається як:

$$F x = \frac{1}{1+e^{-x}} \quad (1.3)$$

Таким чином,

$$OUT = \frac{1}{1+e^{-NET}} \quad (1.5)$$

За аналогією з електронними системами активаційну функцію можна вважати нелінійно підсилювальною характеристикою штучного нейрона.

Коефіцієнт посилення обчислюється як відношення приросту величини OUT до викликавшого його невеликому збільшенню величини NET . Він виражається нахилом кривої при певному рівні збудження і змінюється від малих значень при великих негативних збудженнях (крива майже горизонтальна) до максимального

значення при нульовому збудженні і знову зменшується, коли збудження стає великим позитивним

Слабкі сигнали мають потребу у великому мережевому посиленні, щоб дати придатний до використання вихідний сигнал. Однак підсилювальні каскади з великими коефіцієнтами посилення можуть призвести до насичення виходу шумами підсилювачів (випадковими флуктуаціями), які присутні в будь-якій фізично реалізованій мережі. Сильні вхідні сигнали, у свою чергу, також будуть приводити до насичення підсилювальних каскадів, виключаючи можливість корисного використання виходу. Центральна область логістичної функції, має великий коефіцієнт посилення, вирішує проблему обробки слабких сигналів, в той час як області з падаючим посиленням на позитивному і негативному кінцях підходять для великих збуджень. Таким чином, нейрон функціонує з великим посиленням в широкому діапазоні рівня вхідного сигналу

(1.4)

$$OUT = \frac{1}{1+e^{-NET}} = F(NET)$$

Інший широко використовуваною активаційною функцією є гіперболічний тангенс. За формою вона схожа з логістичною функцією і часто використовується біологами в якості математичної моделі активації нервової клітини.

Подібно логістичної функції гіперболічний тангенс, який наведено на рисунку 1.4, є S-подібною функцією, але він симетричний відносно початку координат, і в точці $NET = 0$ значення вихідного сигналу OUT дорівнює

нулю. На відміну від логістичної функції, гіперболічний тангенс приймає значення різних знаків, і це його властивість застосовується для цілого ряду мереж.

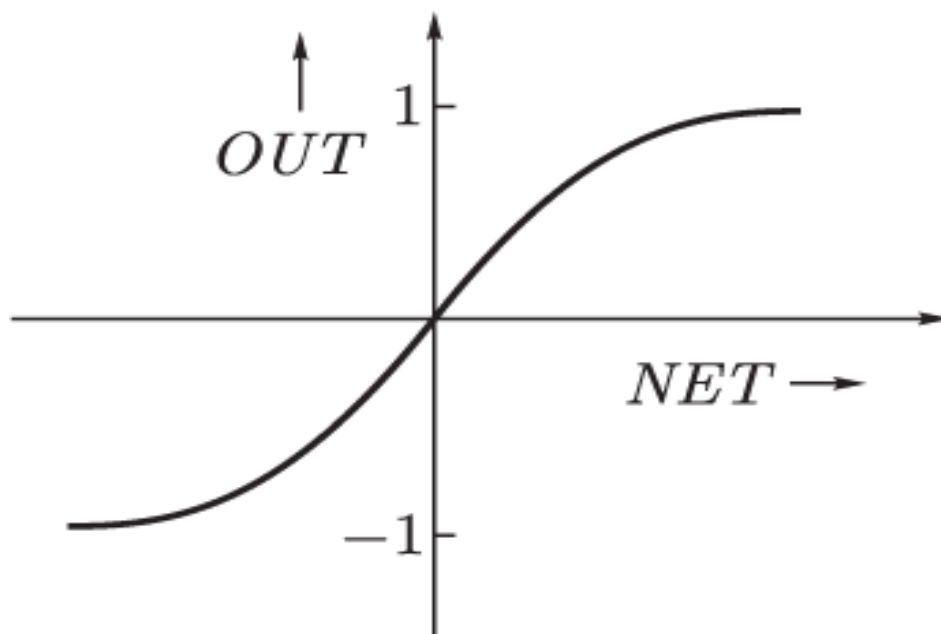


Рисунок 1.4 – Гіперболічний тангенс

Розглянута проста модель штучного нейрона ігнорує багато властивостей свого біологічного двійника. Наприклад, вона не бере до уваги затримки в часі, які впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал. І, що більш важливо, вона не враховує впливів функції частотної модуляції або синхронізуючої функції біологічного нейрона, які ряд дослідників вважають вирішальними у нервової діяльності природного мозку [2].

Незважаючи на ці обмеження, мережі, побудовані з таких нейронів, виявляють властивості, що сильно нагадують біологічну систему. Тільки час і дослідження зможуть відповісти на питання, чи є подібні збіги випадковими або ж вони є наслідок того, що в моделі вірно схоплені найважливіші риси біологічного нейрона.

1.2.7 Одношарові штучні нейронні мережі

Хоча один нейрон і здатний виконувати найпростіші процедури розпізнавання, але для серйозних нейронних обчислень необхідно з'єднувати нейрони в мережі. Найпростіша мережа складається з групи нейронів, що утворюють шар, як показано в правій частині рисунку 1.5.

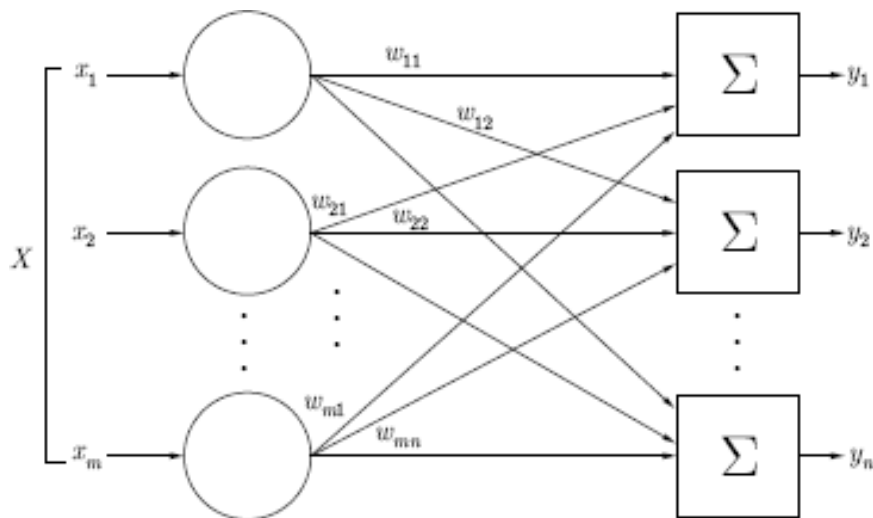


Рисунок 1.5 – Одношарова штучна нейронна мережа.

Зазначимо, що вершини-кола зліва служать лише для розподілу вхідних сигналів. Вони не виконують будь-яких обчислень і тому не будуть вважатися шаром. Для більшої наочності позначимо їх колами, щоб відрізнити їх від обчислюючих нейронів, позначених квадратами. Кожен елемент з безліччю входів X окремою вагою з'єднаний з кожним штучним нейроном. А кожен нейрон видає зважену суму входів в мережу. У штучних і біологічних мережах багато з'єднань можуть бути відсутні, але тут вони показані все для демонстрації загальної картини. Можуть існувати також з'єднання між виходами і входами елементів в шарі.

Зручно вважати ваги елементами матриці W . Матриця має m рядків і n стовпців, де m - число входів, а n - число нейронів.

Таким чином, обчислення вихідного вектора N , компонентами якого є виходи *OUT* нейронів, зводиться до матричного множення $N = XW$, де N і X - вектори-рядки.

1.2.8 Багат шарові штучні нейронні мережі

Більші і складні нейронні мережі володіють, як правило, і великими обчислювальними можливостями. Хоча створені мережі всіх конфігурацій, які тільки можна собі уявити, пошарова організація нейронів копіює шаруваті структури певних відділів мозку. Виявилося, що такі багат шарові мережі володіють великими можливостями, ніж одношарові, і в останні роки були розроблені алгоритми для їх навчання. Багат шарові мережі можуть будуватися з каскадів шарів. Вихід одного шару є входом для наступного шару. Подібна мережа показана на рисунку 1.6 і знову зображена з усіма сполуками. Багат шарові мережі не можуть привести до збільшення обчислювальної потужності в порівнянні з одношаровою мережею, якщо активаційна функція між шарами лінійна.

Це показує, що двошарова лінійна мережа еквівалентна одному шару з ваговою матрицею, що дорівнює добутку двох вагових матриць. Отже, будь-яка багат шарова лінійна мережа може бути замінена еквівалентною одношаровою мережею. Однак одношарові мережі досить обмежені за своїми обчислювальними можливостями. Таким чином, для розширення можливостей мереж в порівнянні з одношаровою мережею необхідна нелінійна активаційна функція.

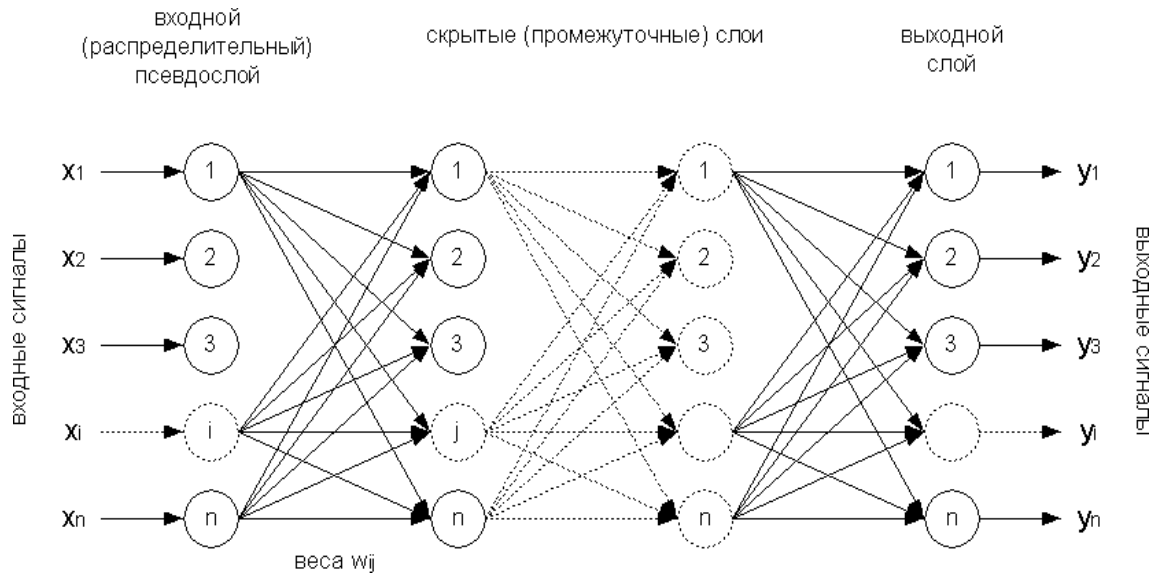


Рисунок 1.6 – Багатошарова штучна нейронна мережа.

У мереж, розглянутих досі, не було зворотних зв'язків, тобто з'єднань, що йдуть від виходів деякого шару до входів цього ж шару або попередніх шарів. Цей спеціальний клас мереж, званих мережами без зворотних зв'язків або мережами прямого поширення, представляє великий інтерес і широко використовується. Мережі більш загального вигляду, що мають з'єднання від виходів до входів, називаються мережами із зворотними зв'язками. У мереж без зворотних зв'язків немає пам'яті, їх вихід повністю визначається поточними входами і значеннями ваг. У деяких конфігураціях мереж із зворотними зв'язками попередні значення виходів повертаються на входи; вихід, отже, визначається як поточним входом, так і попередніми виходами. Тому мережі із зворотними зв'язками можуть мати властивості, подібними з короткочасною людською пам'яттю, де мережеві виходи теж частково залежать від попередніх входів.

На жаль, немає загальноприйнятого способу підрахунку числа шарів в мережі. Багатошарова мережа складається, як показано на рисунку 1.6, з чергуючих множин нейронів і ваг. Раніше, у зв'язку з рисунком 1.5, вже говорилося, що вхідний шар не виконує підсумовування. Ці нейрони служать

лише як розгалужень для першої множини ваг і не впливають на обчислювальні можливості мережі. З цієї причини перший шар не приймається до уваги при підрахунку шарів, і мережу, яка подібна до зображеної на рисунку 1.6, вважається двошаровою, оскільки тільки два шари виконують обчислення. Далі, ваги шару вважаються пов'язаними з наступними за ними нейронами. Отже, шар складається з безлічі ваг з наступними за ними нейронами, підсумовують зважені сигнали.

1.2.9 Нейронні мережі Хопфілда і Хеммінга

Серед різних конфігурацій штучних нейронних мереж (НМ) зустрічаються такі, при класифікації яких за принципом навчання, строго кажучи, не підходять ні навчання з учителем, ні навчання без вчителя. У таких мережах вагові коефіцієнти синапсів розраховуються тільки один раз перед початком функціонування мережі на основі інформації про оброблені дані, і все навчання мережі зводиться саме до цього розрахунку. З одного боку, пред'явлення апріорної інформації можна розцінювати, як допомога вчителя, але з іншого - мережа фактично просто запам'ятовує зразки до того, як на її вхід надходять реальні дані, і не може змінювати свою поведінку, тому говорити про ланку зворотного зв'язку з "світом "(вчителем) не доводиться. З мереж з подібною логікою роботи найбільш відомі мережа Хопфілда і мережа Хеммінга, які зазвичай використовуються для організації асоціативної пам'яті. Далі мова піде саме про них.

Структурна схема мережі Хопфілда наведена на рисунку 1.7. Вона складається з єдиного шару нейронів, число яких є одночасно числом входів і виходів мережі. Кожен нейрон пов'язаний синапсами з усіма іншими нейронами, а також має один вхідний синапс, через який здійснюється введення сигналу. Вихідні сигнали, як звичайно, утворюються на аксонах.

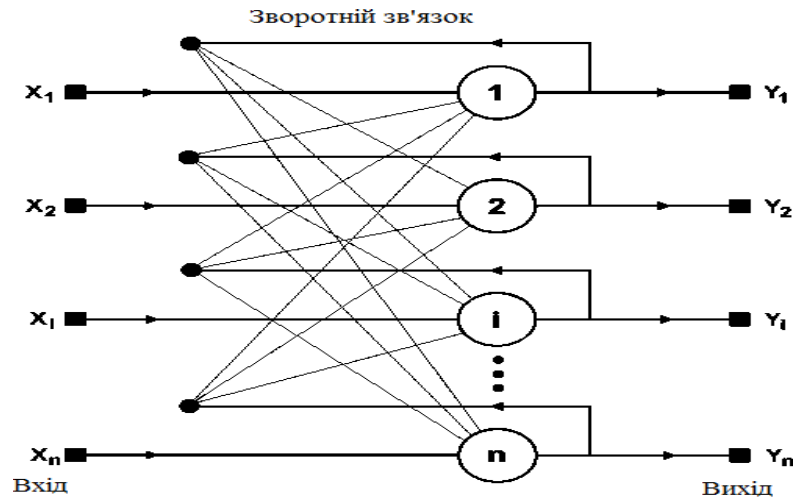


Рисунок 1.7 – Структурна схема мережі Хопфілда.

Завдання, яке вирішується даною мережею в якості асоціативної пам'яті, як правило, формулюється таким чином. Відомий деякий набір двійкових сигналів (зображень, звукових оцифровок, інших даних, що описують якісь об'єкти або характеристики процесів), які вважаються зразковими. Мережа повинна вміти з довільного неідеального сигналу, поданого на її вхід, виділити ("згадати" за частковою інформацією) відповідний зразок (якщо такий є) або "дати висновок" про те, що вхідні дані не відповідають жодному із зразків. У загальному випадку, будь-який сигнал може бути описаний вектором $X = \{x_i; i = 0 \dots n-1\}$, n - число нейронів в мережі і - розмірність вхідних та вихідних векторів. Кожен елемент x_i дорівнює або $(+1)$, або (-1) . Позначимо вектор, що описує k -ий зразок, через X^k , а його компоненти, відповідно, - x_i^k , $k = 0 \dots m-1$, m - число зразків. Коли мережа розпізнає (або "згадає") який-небудь зразок на основі пред'явлених їй даних, її виходи будуть містити саме його, тобто $Y = X^k$, де Y - вектор вихідних значень мережі: $Y = \{y_i; i = 0, \dots, n-1\}$. В іншому випадку, вихідний вектор не співпадає з одним зразковим [6].

Якщо, наприклад, сигнали являють собою якісь зображення, то, відобразивши в графічному вигляді дані з виходу мережі, можна буде

побачити картинку, повністю збігається з однієї зі зразкових (у випадку успіху) або ж "вільну імпровізацію" мережі (у випадку невдачі).

На стадії ініціалізації мережі вагові коефіцієнти синапсів встановлюються таким чином:

$$W_{ij} = \begin{cases} \frac{1}{m-1} \sum_{k=0}^{m-1} x_i^k x_j^k, & i \neq j, \\ 0, & i = j. \end{cases} \quad (1.5)$$

Тут ij - індекси, відповідно, предсинаптичного і постсинаптичного нейронів; x_i^k, x_j^k - i -тий та j -тий елементи вектора k -ого зразка.

Алгоритм функціонування мережі наступний (p - номер ітерації):

1. На входи мережі подається невідомий сигнал. Фактично його введення здійснюється безпосередньою установкою значень аксонів:

$$y_i(0) = x_i, \quad i = 0 \dots n-1, \quad (1.6)$$

тому позначення на схемі мережі вхідних синапсів в явному вигляді носить чисто умовний характер. Нуль в дужках праворуч від y_i означає нульову ітерацію в циклі роботи мережі.

2. Розраховується новий стан нейронів

$$S_j^{p+1} = \sum_{i=0}^{n-1} w_{ij} y_i^p, \quad j = 0 \dots n-1, \quad (1.7)$$

і нові значення аксонів

$$y_i^{p+1} = f_0(S_j^{p+1}), \quad (1.8)$$

де f - активаційна функція у вигляді стрибка, приведена на рисунку 1.8(а).

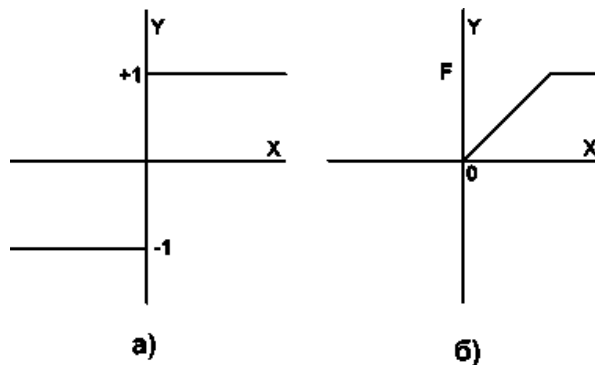


Рисунок 1.8 – Активаційні функції

3. Перевірка, чи змінилися вихідні значення аксонів за останню ітерацію. Якщо так - перехід до пункту 2, інакше (якщо виходи застabilізовані) - кінець. При цьому вихідний вектор являє собою зразок, найкращим чином поєднується з вхідними даними.

Як говорилося вище, іноді мережа не може провести розпізнавання і видає на виході неіснуючий образ. Це пов'язано з проблемою обмеженості можливостей мережі. Для мережі Хопфілда число образів, що запам'ятовуються m не повинно перевищувати величини, приблизно рівної $(0.15 \times n)$. Крім того, якщо два образи А і Б сильно схожі, вони, можливо, будуть викликати у мережі перехресні асоціації, тобто пред'явлення на входи мережі вектора А призведе до появи на її виходах вектора Б і навпаки.

Коли немає необхідності, щоб мережа в явному вигляді видавала зразок, тобто досить, скажімо, отримувати номер зразка, асоціативну пам'ять успішно реалізує мережа Хеммінга. Дана мережа характеризується, в порівнянні з мережею Хопфілда, меншими витратами на пам'ять і обсягом обчислень, що стає очевидним з її структури, яка наведена на рисунку 1.9.

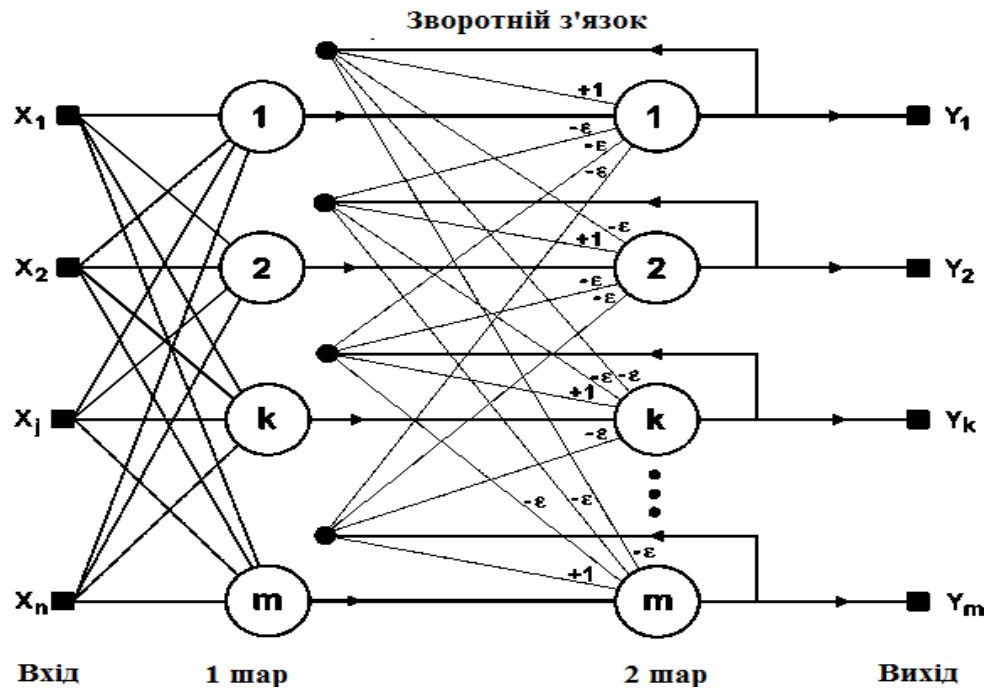


Рисунок 1.9 – Структурна схема мережі Хеммінга.

Мережа складається з двох шарів. Перший і другий шари мають по m нейронів, де m - число зразків. Нейрони першого шару мають по n синапсів, з'єднаних із входами мережі (утворюючими фіктивний нульовий шар). Нейрони другого шару пов'язані між собою інгібторними (негативними зворотними) синаптичними зв'язками. Єдиний синапс з позитивним зворотним зв'язком для кожного нейрона з'єднаний з його ж аксоном.

Ідея роботи мережі полягає в знаходженні відстані Хеммінга від тестованого образу до всіх зразків. Відстанню Хеммінга називається число відрізняючихся бітів у двох бінарних векторах. Мережа повинна вибрати зразок з мінімальною відстанню Хеммінга до невідомого вхідного сигналу, в результаті чого буде активізований тільки один вихід мережі, відповідний цим зразком.

На стадії ініціалізації ваговим коефіцієнтам першого шару і порогу активаційної функції привласнюються наступні значення:

$$w_{ik} = \frac{x_i^k}{2}, i = 0 \dots n - 1, k = 0 \dots m - 1, \quad (1.9)$$

$$T_k = \frac{n}{2}, k = 0 \dots m - 1,$$

де x_i^k - і-ий елемент k-ого зразка.

Вагові коефіцієнти гальмуючих синапсів у другому шарі будуть рівними деякій величині $0 < \epsilon < 1 / m$. Синапс нейрона, пов'язаний з його ж аксоном має вагу +1.

2. СПЕЦІАЛЬНА ЧАСТИНА. РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ІЗ ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ.

2.1 Проектування програмного модуля розпізнавання зображень із застосуванням нейронних мереж.

Основні завдання, які ставляться перед нейронними мережами, відносяться до завдань розпізнавання образів. Вони полягають в тому, щоб класифікувати вхідний образ, тобто віднести його до якого-небудь відомого мережі класу. Спочатку мережі даються еталонні образи - такі образи, приналежність яких до певного класу відома. Потім на вхід мережі подається деякий невідомий образ, і мережа намагається за певним алгоритмом співвіднести його з яким-небудь еталонний зразком. Можна сказати, що нейромережі проводять кластеризацію образів. Так як кластерний аналіз застосовується дослідниками ринку цінних паперів, то нейронні мережі можуть бути використані і для прогнозування вартості акцій, що є актуальним завданням, до того ж строго нерозв'язною на даний момент.

Мережа Хемінга має ряд переваг над мережею Хопфілда. Вона здатна знайти мінімальну похибка, якщо похибки вхідних бітів є випадковими і незалежними. Для функціонування мережі Хемінга потрібна менша кількість нейронів, оскільки середній шар вимагає лише один нейрон на клас, замість нейрона на кожен вхідний вузол. І, зрештою, мережа Хемінга не страждає від неправильних класифікацій, які можуть трапитись у мережі Хопфілда. У цілому, мережа Хемінга швидше і точніше, ніж мережа Хопфілда.

Коли немає необхідності, щоб мережа в явному вигляді видавала зразок, тобто досить, скажімо, отримувати номер зразка, асоціативну пам'ять успішно реалізує мережа Хеммінга. Дана мережа характеризується, в порівнянні з мережею Хопфілда, меншими витратами на пам'ять і обсягом

обчислень, що стає очевидним з її структури, яка представлена на рисунку 1.9, тому в реалізації програмного модуля буде обрана саме вона.

2.1.1 Розробка алгоритму функціонування мережі Хеммінга

Алгоритм функціонування мережі Хеммінга наступний:

1. На входи мережі подається невідомий вектор $X = \{x_i; i = 0 \dots n-1\}$, виходячи з якого розраховуються стани нейронів першого шару (верхній індекс у дужках вказує номер шару):

$$y_j^1 = s_j^1 = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, j = 0 \dots m-1 \quad (2.1)$$

Після цього отриманими значеннями ініціалізуються значення аксонів другого шару:

$$y_j^2 = y_j^1, j = 0 \dots m-1 \quad (2.2)$$

2. Обчислити нові стани нейронів другого шару:

$$s_j^{2p+1} = y_j^{2p} - \varepsilon \sum_{k=0}^{m-1} y_k^{2p}, k \neq j, j = 0 \dots m-1 \quad (2.3)$$

і значення їх аксонів:

$$y_j^{2p+1} = f(s_j^{2p+1}), j = 0 \dots m-1 \quad (2.4)$$

Активаційна функція f має вигляд порога (рис.1.2б), причому величина F повинна бути досить великою, щоб будь-які можливі значення аргументу не призводили до насичення.

3. Перевірити, чи змінилися виходи нейронів другого шару за останню ітерацію. Якщо так - перейти до кроку 2. Інакше - кінець. Алгоритм представлений у вигляді схеми на рисунку 2.1.

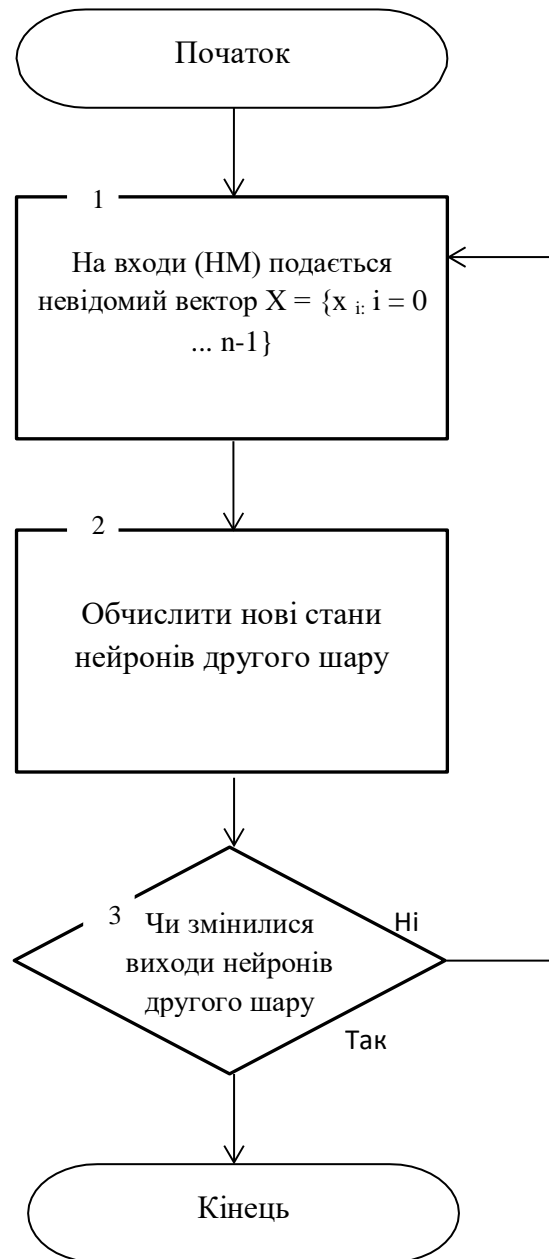


Рисунок 2.1 – Схема алгоритму функціонування мережі Хеммінга

З оцінки алгоритму видно, що роль першого шару досить умовна: скориставшись один раз на кроці 1 значеннями його вагових коефіцієнтів, мережа більше не звертається до нього, тому перший шар може бути взагалі виключений з мережі (замінений на матрицю вагових коефіцієнтів).

2.1.2 Розробка алгоритму навчання штучної нейронної мережі

Серед усіх цікавих властивостей штучних нейронних мереж жодне не захоплює так уяви, як їх здатність до навчання. Їх навчання до такої міри нагадує процес інтелектуального розвитку людської особистості, що може здатися, ніби нами досягнуто глибоке розуміння цього процесу. Але, проявляючи обережність,

слід стримувати ейфорію. Можливості навчання штучних нейронних мереж обмежені, і потрібно вирішити багато складних завдань, щоб визначити, чи перебуваємо ми на правильному шляху.

Мережа навчається, щоб для деякого безлічі входів давати бажане (або, принаймні, сообразное з ним) безліч виходів. Кожне таке вхідний (або вихідна) множина розглядається як вектор. Навчання здійснюється шляхом послідовного пред'явлення вхідних векторів з одночасною підстроюванням ваг відповідно до певної процедури. У процесі навчання ваги мережі поступово стають такими, щоб кожен вхідний вектор виробляв вихідний вектор.

Навчання нейромережі з вчителем. Розрізняють алгоритми навчання з вчителем і без вчителя. Навчання з учителем припускає, що для кожного вхідного вектора існує цільовий вектор, що представляє собою необхідний вихід. Разом вони називаються навчальною парою. Зазвичай мережа навчається на деякому числі таких навчальних пар. Пред'являється вихідний вектор, обчислюється вихід мережі і порівнюється з відповідним цільовим вектором, різницю (помилка) за допомогою зворотного зв'язку подається в

мережу, і ваги змінюються відповідно до алгоритму, що прагнуть мінімізувати помилку.

Вектори навчальної множини пред'являються послідовно, помилки обчислюються і ваги підлаштовуються для кожного вектора доти, поки помилка по всьому навчальному масиву не досягне прийнятно низького рівня.

Навчання нейромережі без вчителя. Незважаючи на численні прикладні досягнення, навчання з учителем критикувалося за свою біологічну неправдоподібність. Важко уявити навчальний механізм в мозку, який би порівнював бажані і дійсні значення виходів, виконуючи корекцію за допомогою зворотного зв'язку. Навчання без вчителя є набагато більш правдоподібною моделлю навчання для біологічної системи. Розвинена Кохоненом і багатьма іншими, вона не потребує цільовому векторі для виходів і, отже, не вимагає порівняння з зумовленими ідеальними відповідями. Навчальне безліч складається лише з вхідних векторів.

Алгоритм навчання нейроної мережі:

У даній дипломній роботі в якості методу навчання використовується дельта- правило Відроу-Хоффа (навчання "з вчителем"). Розглянемо принцип роботи даного методу.

Стан нейрона і визначимо наступним чином:

$$NET_i = \sum_{j=0}^n x_j w_{i,j} \quad (2.5)$$

Як функції активації будемо використовувати сигмоїдальну функцію (1.6).

Ваги нейронів на кожному кроці навчання будемо визначати наступним чином:

$$w_{i,j} \text{ } t + 1 = w_{i,j} \text{ } t + a x_j e_i^k \quad (2.6)$$

2.1.3 Розробка алгоритму порівняння зображень з використанням відстані Хеммінга

Суть розпізнавання будь-якого тексту - порівняння з образом. У дитинстві, дивлячись на текст, ми бачимо всього лише картинку. Ми ще не наділені необхідними критеріями її аналізу, для того щоб зрозуміти, що на картинці не просто малюнок, а текст, і не просто текст, а якусь розповідь. Ми ростемо, і починаємо отримувати образи символів, дізнаємося що малюнок з буквою «А» позначає букву А, а не пароходик. Стаючи старше, ми набираємо базу образів (або базу критеріїв образів), за якою шляхом порівняння та аналізу ми можемо сказати, який символ перед нами.

Точно така ж ситуація і з комп'ютером. Спершу він просто отримує в пам'ять картинку, потім шукає на ній якісь області, і після цього порівнює ці області зі своєю базою і проводить аналіз, результатом якого буде визначення символу на зображенні. Передаємо в пам'ять картинку, шукаємо на картинці якісь області, порівнюємо з базою, але методи порівняння картинки досі незрозумілі.

Сучасні методи порівняння зображення діляться на дві категорії:
1.Порівняння з еталоном.

Порівняння за критеріямиМетод нейронних мереж може використовуватись в обох випадках.

Порівняння за критеріями найбільш трудомістким, тому що виконати алгоритмічне завдання з визначенням характерних рис символу складно, та й ефективність подібного алгоритму буде напевно слабка. Пов'язано це зі схожістю символів, а якщо ми будемо розпізнавати текст низького розширення, то витрат на відточування подібного алгоритму буде в рази більше [16].

Порівняння з еталоном - завдання більш просте і більш ефективне. Тут все максимально просто - створити еталонне зображення і порівняти їх.

Ми будемо використовувати метод порівняння - попіксельно. Суть - почергове порівняння двох відповідних пікселів двох чорно-білих картинок. Оскільки два пікселі повинні бути відповідними, то картинки повинні бути однакові за розміром.

Для простоти ми будемо розглядати тільки чорно-білі зображення. Наприклад малюнок складається всього з двох пікселів. Тоді множину всіх об'єктів, яку можна буде зобразити (універсальна множина), яка представлена на рисунку 2.3, складається з чотирьох об'єктів: (0,0), (0,1), (1,0), (1,1), де 1 - чорний піксель, 0 – білий.









X_1	X_2	X_1	X_2
		0	0
		1	0
		0	1
		1	1
А)		Б)	

Рисунок 2.3 – Універсальна множина

Всі об'єкти універсальної множини можна розмістити у вершинах одиничного квадрата, таким чином, множині фігур, зображених на двухпіксельному полі, може бути порівнювати безліч точок у двовимірному просторі. Ребру цього квадрата буде відповідати перехід від одного зображення до іншого. Для переходу від (1,1) до (0,0) потрібно буде пройти два ребра, для переходу від (0,1) до (0,0) - одне. Зазначимо, що число ребер у нашому переході - це кількість незбіжних пікселів двох зображень. Висновок: відстань від одного малюнка до іншого дорівнює числу незбіжних пікселів в них. Ця відстань називається відстанню Хеммінга, яка графічно зображена на рисунку 2.4.

Наступний етап роботи нашої програми - складання бази (масив записів символу і його графічного представлення).

Ми зробимо стандартну базу зображень українських символів і цифр без застосування графічних редакторів – у вихідному коді. Для цього буде потрібно задати зображення потрібного розміру, відкрити режим канви, і написати в зображенні необхідну літеру.

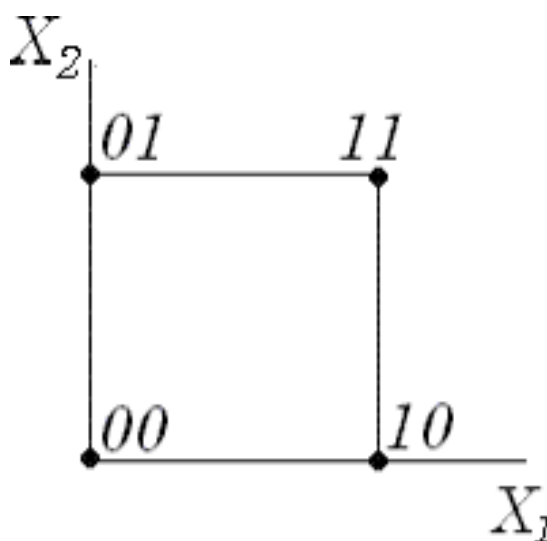


Рисунок 2.4 – Представлення відстані Хеммінга на системі координат

Тут же необхідно обробити отриману картинку так, як ми обробляємо вихідну, щоб еталон був максимально наближений до розпізнаваної картинці і навпаки.

Питання дещо ускладнюється, якщо потрібно порівняти зображення різних розмірів. Однак досить очевидним рішенням цієї проблеми є приведення всіх зображень до однакового розміру. Залишається вибрати цей розмір - якщо вибрати його занадто маленьким, то очевидно, що відмінності між зображеннями будуть нівелюватися, і у нас буде багато хибно позитивних спрацьовувань. При занадто великому розмірі невиправдано підвищується ресурсомісткість алгоритму порівняння.

Досвід показує, що найбільш оптимальним є розмір наведеного зображення від 16x16 до 64x64. Так само досвід показує, що колірні характеристики зображення роблять набагато менший вплив на візуальні відчуття схожості зображень в порівнянні з характеристиками яскравості, тому їх можна відкинути.

Отже, алгоритм порівняння схожих зображень буде включати в себе наступні кроки. Приведення всіх зображень до одного розміру. Відкидання колірної інформації (перетворення в сіре зображення). Знаходження середньоквадратичної різниці для кожної пари зменшених сірих зображень.

Порівняння отриманої середньоквадратичної різниці деяким порогом.

Тут поріг визначає міру схожості зображень. Так якщо вибрати порогом 0%, то алгоритм знайде тільки повністю ідентичні зображення. При 5% порозі алгоритм зможе знайти також візуально схожі зображення, які можуть змінюватися роздільною здатністю, якістю стиснення, наявністю дрібних написів і т.д. при незначній кількості помилкових спрацьовувань. При порозі вище 10%, як правило, кількість помилкових спрацьовувань може перевищувати число позитивних спрацьовувань. Даний алгоритм в тому чи іншому варіанті повинен бути застосований у будь-якій програмі, що здійснює пошук схожих зображень. Варіації зазвичай стосуються способу отримання зменшеного зображення, розмірів зменшеного зображення і способів його квантування по кольоровій палітрі, вибору метрики, визначає ступінь відмінності зображень.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Програмна реалізація модуля для розпізнавання зображень

3.1.1 Обґрунтування вибору мови програмування

Система програмування Delphi версії 7 фірми Enterprise (Borland) надає найбільш широкі можливості для програмування додатків ОС Windows.

Delphi - це продукт Borland International для швидкого створення додатків. Процес створення інтерфейсу майбутньої програми нагадує забаву з ігровим комп'ютерним конструктором. Тому RAD-середовища ще називають візуальними середовищами розробки: якими ми бачимо робітники і діалогові вікна програми при проектуванні, такими вони і будуть, коли програма запрацює.

Високопродуктивний інструмент візуального побудови додатків включає в себе справжній компілятор коду і надає засоби візуального програмування, кілька схожі на ті, що можна виявити в Microsoft Visual Basic (вона не є RAD-системою) або в інших інструментах візуального проектування. В основі Delphi лежить мова Object Pascal, який є розширенням об'єктно-орієнтованої мови Pascal.

Насамперед Delphi призначений для професійних розробників, бажаючих дуже швидко розробляти програми в архітектурі клієнт-сервер. Delphi виробляє невеликі за розмірами високоефективні виконувані модулі (.Exe і .Dll), тому в Delphi повинні бути, перш за все, зацікавлені ті, хто розробляє продукти на продаж. З іншого боку невеликі за розмірами і швидко виконувані модулі означають, що вимоги до клієнтських робочих місцях істотно знижуються - це має важливе значення і для кінцевих користувачів.

3.1.2 Розробка програмного продукту

Для реалізації дипломного проекту була обрана мова Object Pascal. Розробка програмного продукту велася в середовищі програмування Delphi 7.

Як й інші традиційні інженерні дисципліни, розробка програмного забезпечення має справу з проблемами якості, вартості та надійності. Деякі програми містять мільйони рядків вихідного коду, які, як очікується, повинні правильно виконуватися в умовах, що змінюються. Складність ПЗ порівнянна з складністю найбільш складних з сучасних машин.

На сьогоднішній день вважається кращою програма, яка потребує менших затрат на розробку та супроводження, а не найкоротша чи найшвидша. Найбільш перспективний напрям організаційного розвитку є напрямок розподілу праці, спрощення кожної функції або операції, нормування праці, поетапний контроль результатів, високий рівень стандартизації. Така організація праці у виробництві ПП становить технологію виробничого процесу. Немає універсальної технології розробки програм, але більшість запропонованих технологій мають спільні риси, адже об'єкт праці – програмний продукт відрізняється продуктовою однаковістю, одним рівнем розвитку обчислювальної техніки та самої галузі програмування. За рахунок модифікації й оновлення процедурні технології успішно конкурують із новітніми сучасними концепціями. Розробка програми на основі наперед підготовлених компонентів (технологія збірного програмування) є засобом переведення програмування на індустріальну основу, зменшення складності програмування реальних задач.

Розглянемо докладніше ієрархію форм дипломного проекту та їх призначення та зв'язку.

Перерахуємо форми використані в проекті і опишемо їх призначення:

- MainForm - головна форма проекту. Використовується для завантаження розпізнаваного зображення, його обробки, а також для управління іншими формами;

- LearnForm - форма, за допомогою якої буде проводитися створення, навчання, тестування і збереження нейромережі;
- UtilForm - форма, за допомогою якої буде формуватися база еталонних зображень;
- NoizeLevelForm - форма, основним завданням якої, буде визначення порогу відсікання, при очищенні зображення;
- RecognizeForm - форма, яка безпосередньо буде служити для виведення розпізнаного тексту;
- LogForm - форма, в яку будуть передаватися відомості про роботу програми, для їх відображення та збереження;

Розглянемо основні модулі програми:

- модуль main.pas-головний модуль програми, через який здійснюється управління рештою функцій і модулями програми;
- модуль add.pas містить в собі процедури і функції програми які не зв'язані з розпізнаванням або навчанням нейромережі. Тобто додаткові функції, наприклад пошук файлів, очищення пам'яті, порівняння двох зображень, вирівнювання компонента за стовпцями і т.д;
- модуль loadImageUnit.pas містить в собі функції, що управляють відображенням завантаженого зображення;
- модуль clearUtil.pas містить в собі функції для очищення зображення від шумів;
- модуль convertToGray.pas містить в собі функції для конвертації зображення в чорно-біле растрове зображення;
- модуль Learn.pas містить в собі функції для управління процесом навчання;
- модуль learnFunctions.pas містить в собі функції, які застосовуються при навчанні нейромережі;
- модуль Log.pas містить в собі функції, які застосовуються для формування, відображення, збереження логу;
- модуль NNClass.pas - відповідає за управління класом нейромережі.

- модуль `noizeLevelSetup.pas` містить функції що відповідають за установку порогового значення при обробці зображень;
- модуль `selectRecArea.pas` - містить в собі функції, які відповідають за сегментацію зображення і виділення областей розпізнавання;

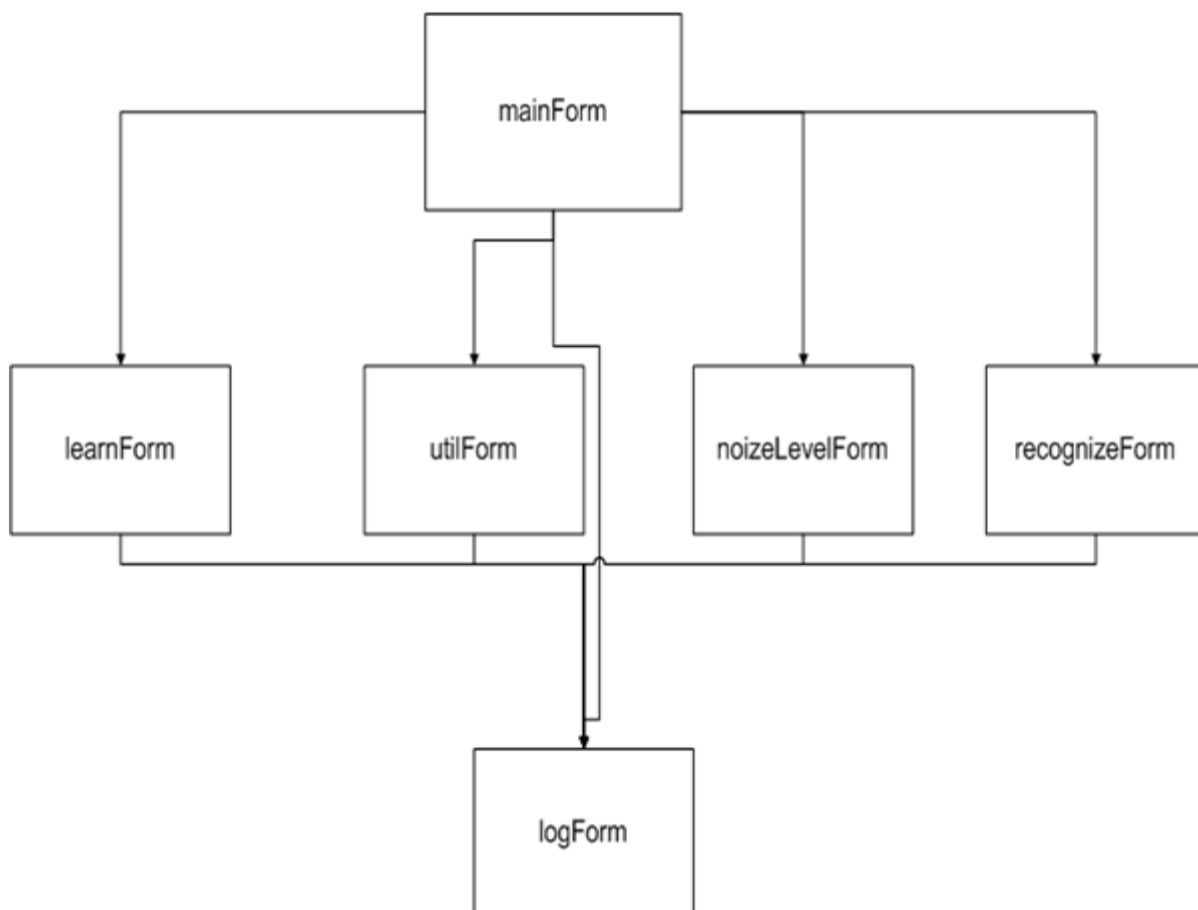


Рисунок 3.1 – Схема взаємодії форм

Розглянемо процес роботи програми в цілому. Розпізнаваний документ завантажується в програму. Якщо існує навчена нейронна мережа для такого типу документів, то користувач вибирає її, і система розпізнає документ. Якщо ж навченої мережі не існує, то конвертуємо зображення в градації сірого, після чого виділяються всі сегменти для розпізнавання. Після цього сформовані сегменти обробляються за допомогою утиліти для формування еталонів для навчання. Після формування бази, запускається утиліта для навчання нейромережі, виставляються первинні значення для нейромережі і

виробляється навчання. Після навчання, мережа тестується, і якщо результат незадовільний, то мережа перенавчається. Якщо результат тестування навченої нейромережі задовільний, то нейромережа з усіма параметрами зберігається у файл. Після чого проводиться розпізнавання і висновок розпізнаного документа.

4. ТЕСТУВАННЯ

4.1 Тестування приклад роботи програми та аналіз результатів

1. Запуск програми.

При запуску програми з'являється її головне вікно яке представлене на рисунку 4.1. За допомогою його ведеться управління роботою програми.

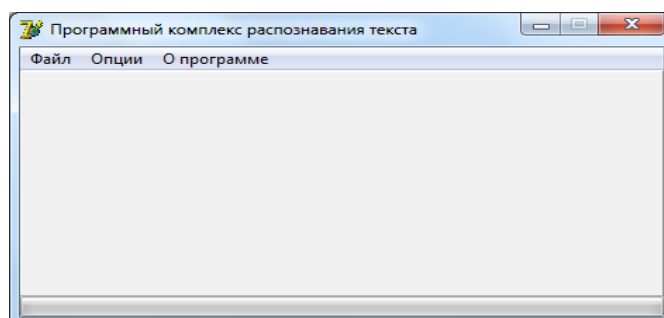


Рисунок 4.1 – Головне вікно програми

2. Вибір документа для розпізнавання і його завантаження.

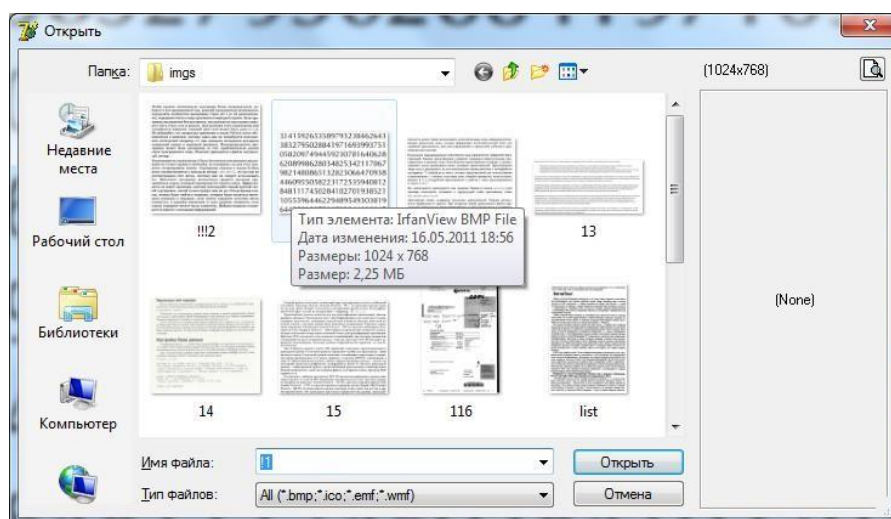


Рисунок 4.2 – Вибір документа для розпізнавання

Для вибору документа, який потрібно розпізнати, необхідно натиснути меню «Файл - Завантажити ...», після чого з'явиться вікно вибору файлу як представлено на рисунку 4.2.

3. Конвертація документа в градації сірого, очищення від шумів.

Після вибору документа, він завантажується в головну форму програми яку представлено на рисунку 4.3, після чого, можна приступати до процесу його обробки.

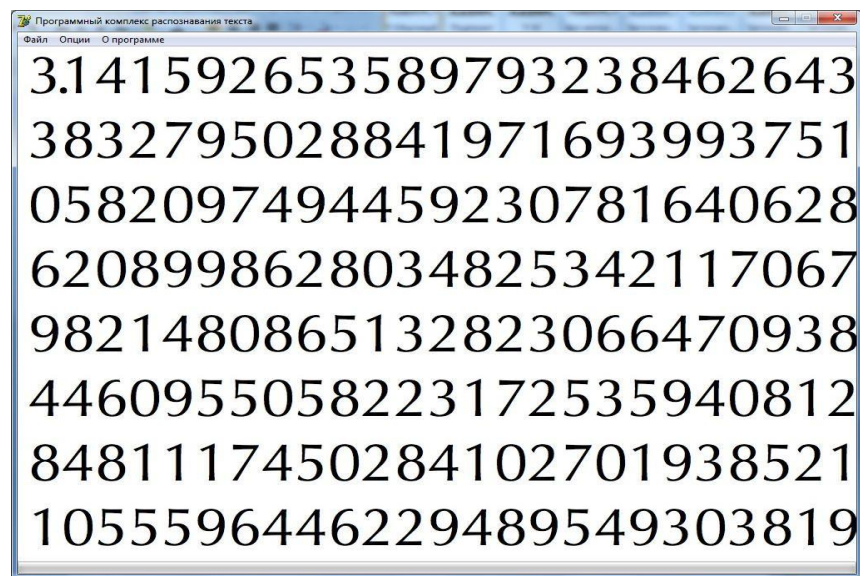


Рисунок 4.3 – Головне вікно з завантаженим документом

Для того, що б провести конвертацію документа в градації сірого кольору і очистити його від шумів, необхідно натиснути меню «Опції-Конвертувати в градації сірого», після чого з'явиться форма з назвою «Граничний рівень» яку представлено на рисунку 4.4. У цій формі, за допомогою повзунка, необхідно встановити необхідний рівень порогу. Після чого необхідно натиснути кнопку "Ок".

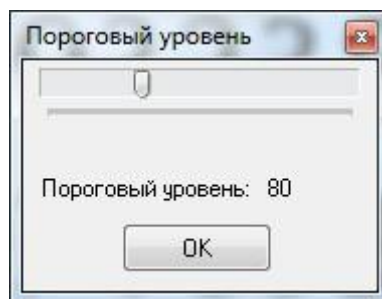


Рисунок 4.4 – Вікно вибору порогового рівня

Після вибору рівня порогу, почнеться процес очищення зображення від шумів і конвертація його в чорно-біле зображення, яке представлено на рисунку 2.11.

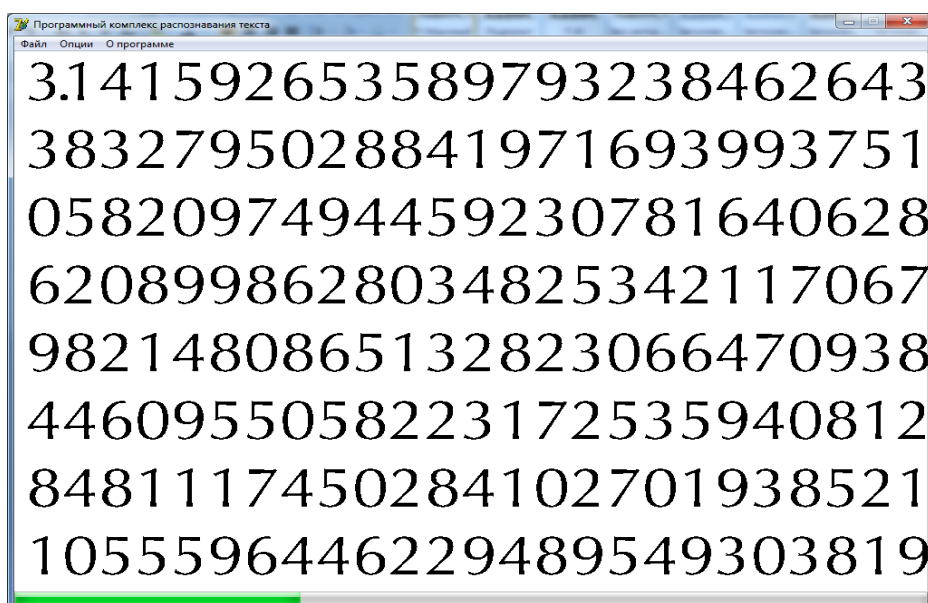


Рисунок 4.5 – Процес порогового відсікання зображення

4. Виділення зон розпізнавання (сегментація зображення).

Після закінчення конвертації зображення в чорно-білий колір і очищення його від шумів, необхідно натиснути меню «Опції - Виділити зони розпізнавання». Процес виділення зон розпізнавання, який наведений на

рисунку 4.6, займає досить довгий час, тому внизу вікна є індикатор процесу виконання процедури.

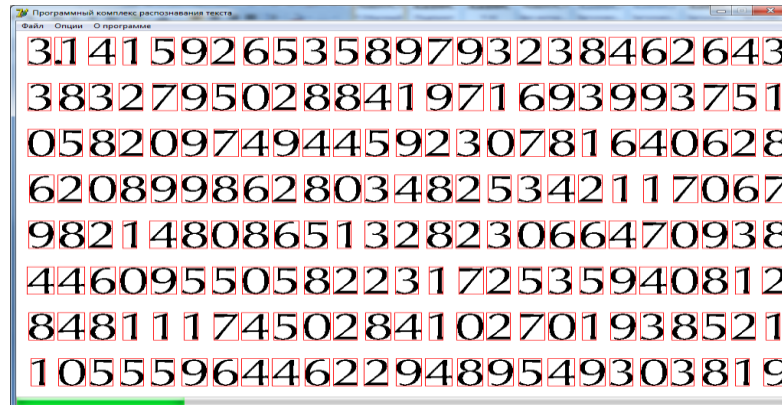


Рисунок 4.6 – Процес виділення сегментів для розпізнавання

5. Формування навчальної вибірки (еталонної бази).

Після виділення зон розпізнавання, файли з сегментами за замовчуванням зберігаються в домашній каталог програми, в папку «recSrc». Для того, щоб навчити нейромережу, необхідно сформувати еталонну базу для навчання, в якій елементи навчання (виділені сегменти), будуть відсортовані за своїми ознаками.

Для формування бази еталонів, необхідно натиснути меню «Опції - Сформувати базу для навчання», після чого з'явиться форма «Підготовка бази еталонів», яка представлена на рисунку 4.7.

Спочатку необхідно вибрати папку, в якій зберігаються необроблені сегменти. Для цього необхідно натиснути кнопку із заголовком «...», навпроти рядка «Виберіть адресу з необробленими картинками»

Після цього необхідно вибрати папку, в якій буде сформована база. Для цього необхідно натиснути кнопку із заголовком «...», навпроти рядка «Виберіть адресу для збереження бази картинок»

Щоб зменшити розмір навченою вибірки, необхідно очистити її від дублікатів, тобто від однакових файлів. Для цього необхідно натиснути кнопку із заголовком «Видалити дублікати». Після цього, почнеться

процедура очищення сегментів від дублікатів. Хід даної процедури, будуть відображати дві смуги прогресу. Нижня, відповідає за загальний хід операції, верхня за перевірку поточного сегмента, на наявність дублікатів.

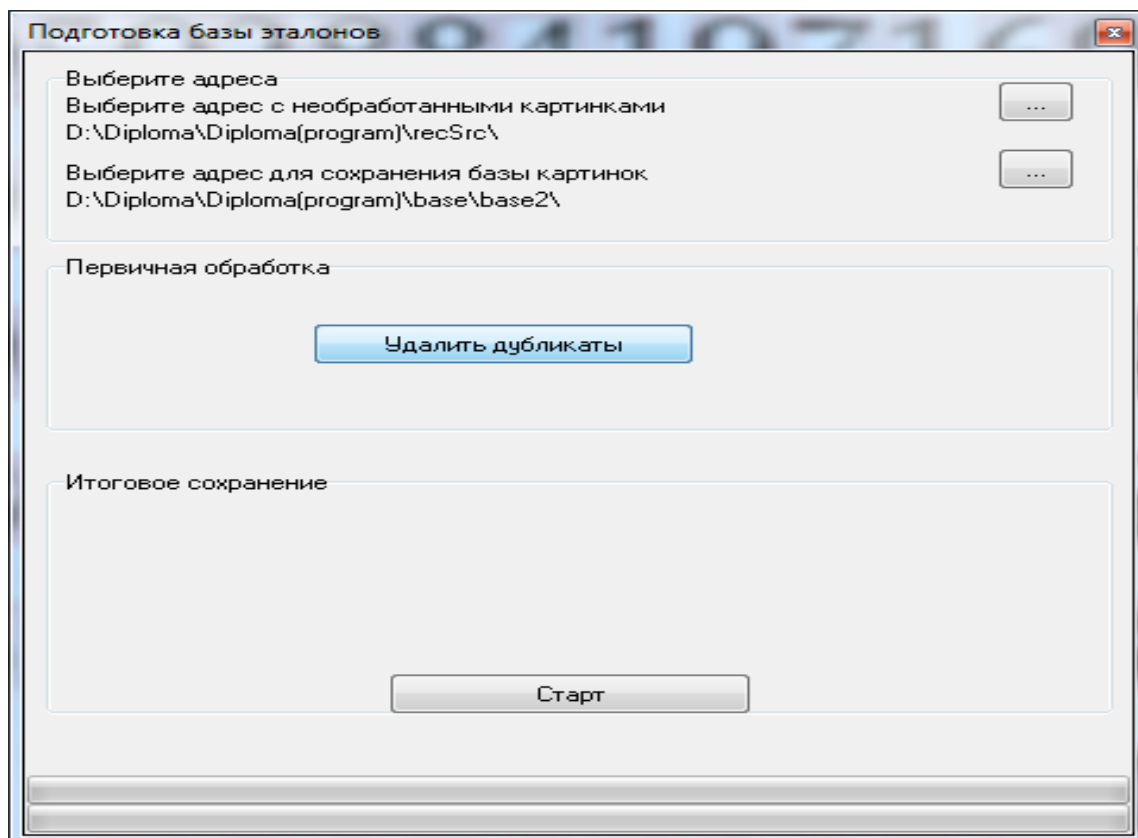


Рисунок 4.7 – Форма підготовки еталонних зображень для навчання нейромережі

Після видалення всіх дублікатів, можна приступати до формування еталонної бази (навчальній вибірці). Для цього необхідно натиснути кнопку «Старт» рисунок 4.8, після чого з'явиться вікно з заголовком «Введіть ім'я», а в нижній частині вікна в розділі «Підсумкове збереження», з'явиться зображення символу. У полі «Введіть ім'я символу», вікна «Введіть ім'я», необхідно ввести відображається символ.

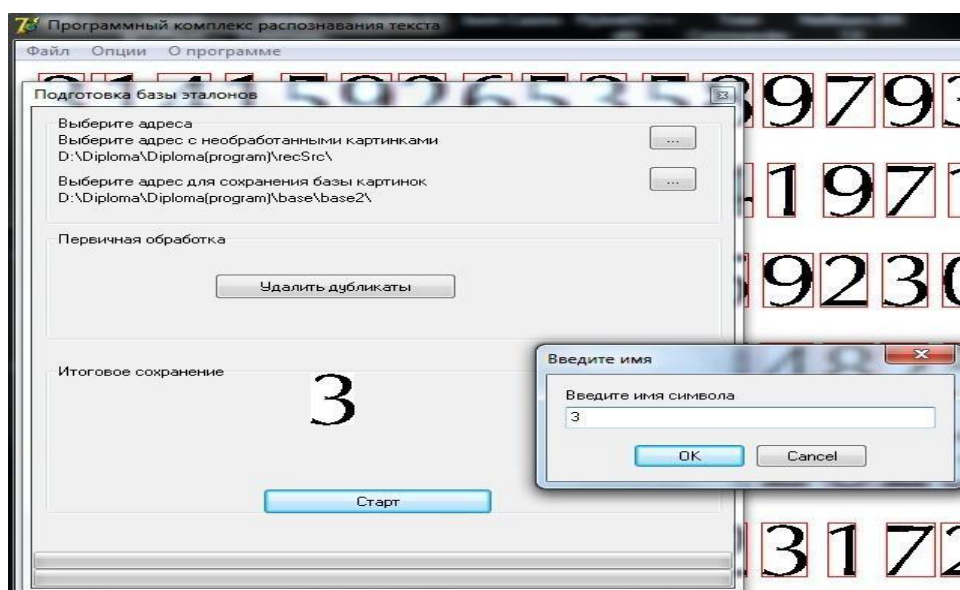


Рисунок 4.8 – Процес формування бази еталонів

6. Навчання нейромережі.

Після того, як буде навчена вся вибірка, можна приступати до навчання нейромережі. Для цього в головній формі необхідно натиснути меню «Опції - Навчання». Після чого з'явиться форма «Навчання», яка наведена на рисунку 4.9.

Детальніше розглянемо форму навчання нейромережі. У ній міститься п'ять розділів для налаштування, навчання, тестування і відображення інформації про нейромережі.

Розділ «Первинна настройка мережі та її нормалізація», необхідна для установки параметрів нейромережі. Розглянемо докладніше його пункти:

- пункт «Вибрати папку з еталонними символами», служить для того, що б вибрати папку з раніше сформованої базою символів;
- пункт «Проаналізувати еталони і сформувати НМ», служить для того, що б проаналізувати базу символів, і встановити параметри входів у нейромережу;
- пункт «Поставити кількість виходів і сформувати класифікатор», необхідний лише в тому випадку, якщо відомо кількість нейронів (виходів) нейронної мережі;

- пункт "Зберегти первинний файл даних мережі», служить для збереження файлу нейронної мережі;

- пункт «Завантажити існуючу мережу», необхідний лише в тому випадку;

- якщо потрібно довчити мережу або перевчити;

Розділ «Дані про мережі», служить для виведення загальної інформації про мережу, а також інформації про навчання. Розгляньмо докладніше його пункти:

- пункт «Папка з еталонними символами», служить для відображення адреси бази з еталонними символами;

- пункт «Кількість файлів на обробку», служить для відображення інформації про кількість файлів з бази еталонів, які належать даному класу;

- пункт «Розмірність мережі W », служить для відображення максимальної ширини картинки з навчальної множини;

- пункт «Розмірність мережі H », служить для відображення максимальної висоти картинки з навчальної множини;

- пункт «Кількість входів в мережу», служить для відображення інформації про кількостях входів у нейронну мережу;

- пункт «Кількість виходів», служить для відображення інформації про кількостях виходів з нейронної мережі;

- пункт "Файл нейромережі зберігається в», служить для відображення інформації про адресу зберігання файлу нейромережі.

- пункт «Ім'я учня символу», служить для відображення інформації про ім'я якого навчають символу, в процесі навчання;

- пункт «Коефіцієнт помилки», служить для відображення інформації, про помилку мережі в процесі навчання, для даного учня символу;

Розділ «Детальне налаштування мережі», містить у собі елементи, для налаштування параметрів мережі, таких як швидкість навчання і коефіцієнт допустимої помилки. Розгляньмо докладніше його пункти:

- пункт «а - коефіцієнт навчання», служить для налаштування параметра швидкості навчання мережі;
- пункт «R - припустима помилка мережі», служить для налаштування параметра коефіцієнта допустимої помилки мережі;
- розділ «Результати тестування», містить елементи управління і відображення процесу тестування мережі. Розгляньмо докладніше його пункт;
- кнопка «Тестувати мережу», служить для того, щоб вибрати з тестованої вибірки файл, і запустити процедуру тестування;

Розділ «Список файлів на навчання», містить елементи управління і відображення процесу навчання мережі. Розгляньмо докладніше його пункти:

- а) пункт «Групове навчання», встановлюється активним, лише в тому випадку, коли еталонна база була сформована вручну, а не за допомогою утиліти формування бази еталонів;
- б) кнопка «Старт», відповідає за запуск процедури навчання;
- в) елемент список, відображає список навченої вибірки;

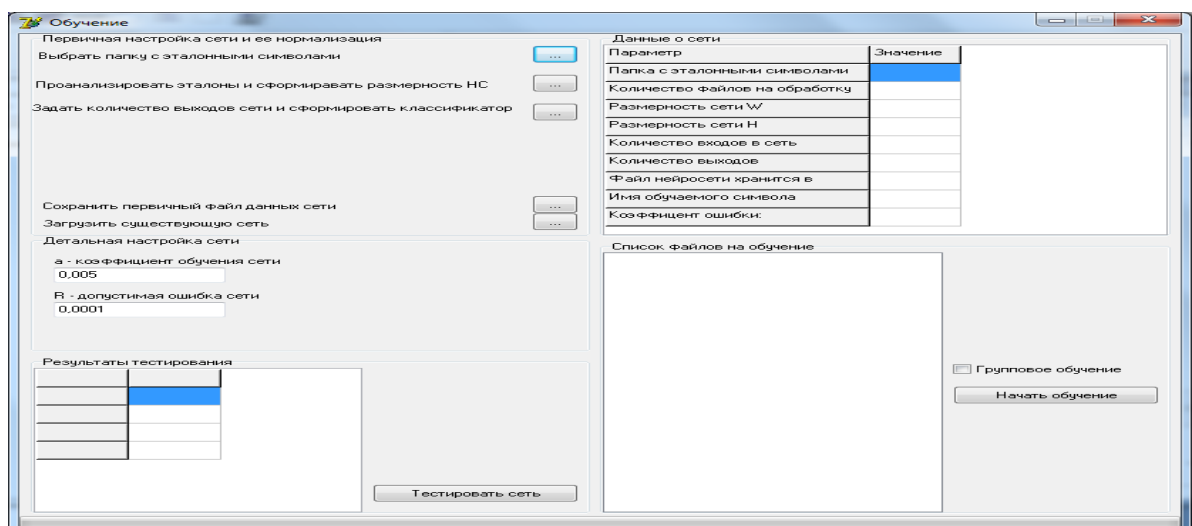


Рисунок 4.9 – Форма навчання нейромережі

Спочатку необхідно вибрати каталог, в якому містяться, еталонні зображення. Для цього необхідно натиснути на кнопку із заголовком «...», у

пункті «Вибрати папку з еталонними символами», розділу «Первинна настройка мережі та її нормалізація». Після вибору папки, в пункті «Папка з еталонними символами», розділу «Дані про мережі», повинен відобразитися адресу зберігання бази еталонів.

Після вибору папки з базою еталонів, необхідно проаналізувати базу, і встановити параметри мережі. Для цього необхідно натиснути кнопку із заголовком «...», у пункті «Проаналізувати еталони і сформувати розмірність

НМ», розділу «Первинна настройка мережі та її нормалізація». Після чого в пунктах «Розмірність мережі W», «Розмірність мережі H», «Кількість виходів в мережу», розділу «Дані про мережі», повинна відобразитися інформація про результати аналізу мережі.

Після аналізу первинних даних, тільки в тому випадку, якщо відома розмірність виходів нейронів і їх класи, необхідно натиснути кнопку із заголовком «..», у пункті «Поставити кількість виходів і сформувати класифікатор», розділу «Первинна настройка мережі та її нормалізація». Після чого з'явиться вікно з заголовком «Поставте кількість виходів в мережі», яке наведене на рисунку 4.10. У текстовому полі необхідно ввести кількість виходів мережі і натиснути кнопку «ОК». Якщо ввести в поле параметр «-» - та кількість виходів буде невідомим.

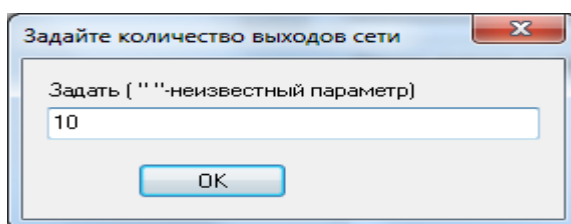


Рисунок 4.10 – Форма введення параметра кількості виходів мережі

Після введення кількості виходів мережі з'явиться форма з ім'ям «Ім'я нейрона», яка наведена на рисунку 4.11. У текстовому полі необхідно ввести ім'я класу нейрона і натиснути кнопку «ОК». Форма «Ім'я нейрона»,

з'явиться стільки разів, скільки було вказано значень розмірності виходів нейромережі. І кожного разу необхідно ввести ім'я відповідного класу нейрона.

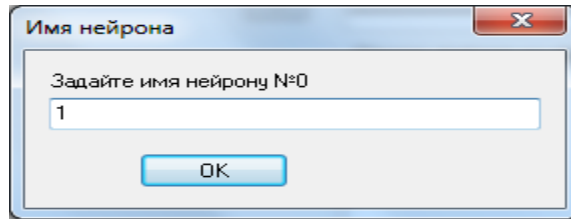


Рисунок 4.11 – Форма для введення імені класу нейрона

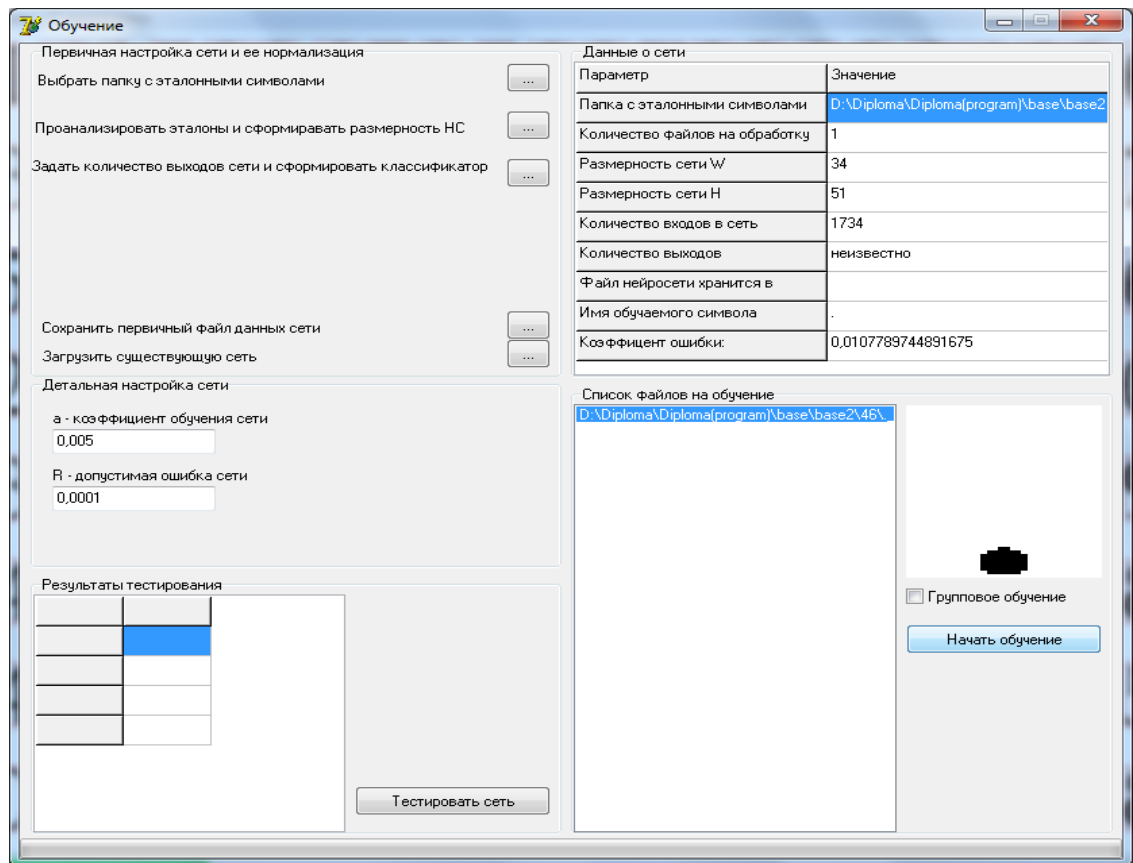


Рисунок 4.12 – Процесс навчання нейромережі

Для установки параметрів мережі α і R , швидкості навчання і допустимої помилки мережі відповідно, значень відмінних від тих, що встановлені за замовчуванням, необхідно встановити нові значення в розділі «Детальне налаштування мережі», у відповідних полях.

Після виконання всіх вищеописаних операцій, можна приступати до процесу навчання нейромережі. Для цього необхідно натиснути кнопку із заголовком

«Старт». Після чого почнеться процедура навчання мережі, яку можна побачити на рисунку 4.12.

7. Тестування нейромережі

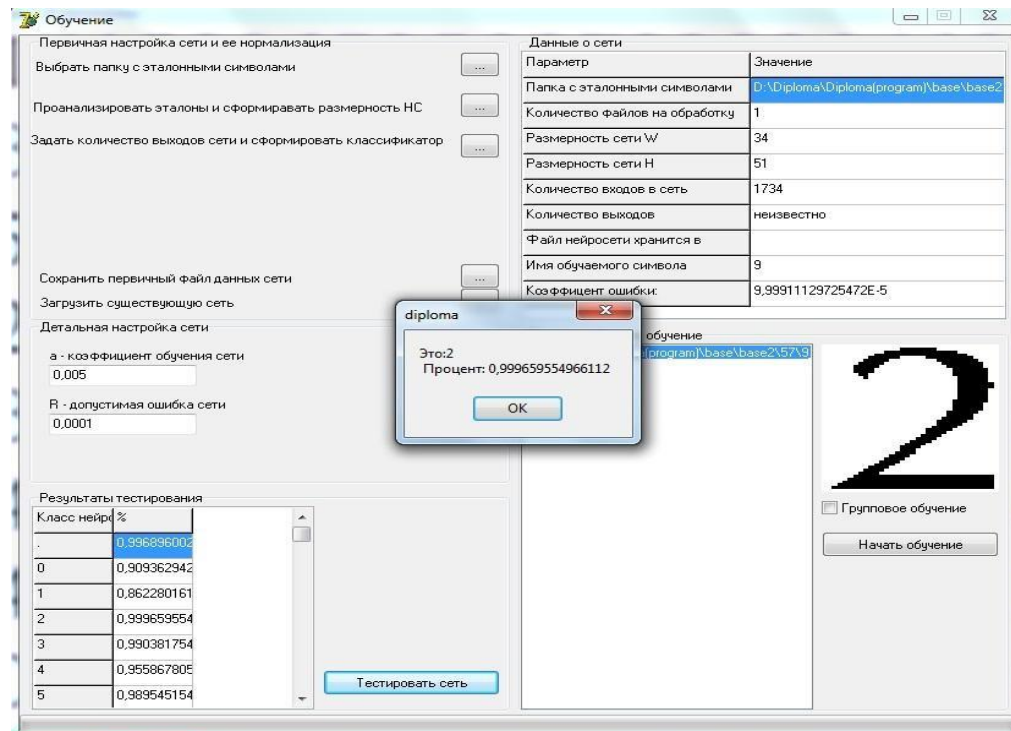


Рисунок 4.13 – процес тестування навченої нейромережі

Після завершення навчання, тобто після налаштування вагових коефіцієнтів, можна приступати до перевірки правильності навчання нейромережі. Для цього необхідно натиснути кнопку із заголовком «Тестувати мережа». У вікні, вибрати тестовий приклад, і натиснути кнопку «Відкрити». Після цього на екрані з'явиться вікно з відповіддю і процентним збігом, яке представлено на рисунку 4.12. А в розділі «Результати

тестування» відобразиться повна інформація про тестування мережі на даному екземплярі.

Процес тестування нейронної мережі необхідно провести на всіх тестових зразках. Якщо результат задовільний, то необхідно натиснути кнопку «Зберегти первинний файл даних мережі», у вікні вибрати адресу для збереження файлу, ввести ім'я файлу, натиснути кнопку «Зберегти». Якщо результат, незадовільний, то необхідно провести нове навчання нейромережі.

ВИСНОВКИ

В результаті виконання дипломної роботи було створено інтелектуальний систему, що дозволяє розпізнавати текст з відсканованих текстових зображень. Обрана реалізація програми вимагає мінімальні обчислювальні затрати, що дає можливість використовувати її на комп'ютерах з слабкими можливостями.

В процесі моделювання інтелектуального модуля було розроблено алгоритм функціонування мережі Хеммінга та алгоритм навчання штучної нейронної мережі з використанням дельта-правила Відроу-Хоффа що дозволило нейронній мережі служити в системі розпізнавання тексту в якості класифікатора. Цей класифікатор можна навчати, налаштовуючи коефіцієнти на елементах мережі і таким чином, прагнути до ідеального результату розпізнавання.

Розроблена програма розпізнавання текстових зображень використовує технологію OCR даний підхід вимагає створення шаблону для кожного шрифту. Розроблена програма розпізнає текст двох видів шрифтів Arial і Times New Roman. В процесі розпізнавання база еталонних зображень може поповнюватись новими видами шрифтів.

Для дослідження проблеми була введена база даних (БД) з еталонних образів, що складається з різновидів 10 цифр та 33 букв українського алфавіту, що в сумі склало 43 різних символічних знаків.

Застосована нейромережа імітує штучний інтелект, тим самим, дозволяючи навчати програму, для вірного розпізнавання зображень при наступному використанні програми. Програмна реалізація інтелектуальної системи здійснена в середовищі програмування Delphi 7.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. V. P. Kozemiako, O.K. Kolesnytskyj , T. S. Lischenko Optical Fibers and Their Applications 2012, 86980M (January 11, 2013); doi:10.1117/12.2019340 From Conference Volume 8698 Optical Fibers and Their Applications 2012 Krasnobrod, Poland | October 10, 2012.
2. W. Maass and C. M. Bishop Hemming Networks, editors. MIT Press : Cambridge. 2001. ISBN 0-262-13350-4.
3. В.Ф. Бардаченко, О.К. Колесницький, С.А. Василецький Перспективи застосування імпульсних нейронних мереж з таймерним представленням інформації для розпізнавання динамічних образів УСiМ. 2003, №6. С. 73 – 82.
4. Распознавание текста, используя расстояние Хемминга URL: <http://habrahabr.ru/post/90867/>
5. Зайченко Ю. П. Основи проектування інтелектуальних систем. Київ: Видавничий дім «Слово», 2004. – 352с.
6. Применение нейросетей в распознавании изображений URL: <http://habrahabr.ru/post/74326/>
7. R. Legenstein, W. Maass, S. Haykin What makes a dynamical system computationally powerful? New Directions in Statistical Signal Processing: From Systems to Brain. Cambridge, MIT Press, 2007. P. 127–154.
8. Tamagawa D., Taniguchi E., Yamada T. Evaluating city logistics measures using a multi-agent model // Procedia-Social and Behavioral Sciences. 2010. Т. 2. №. 3. С. 6002– 6012.
9. Горелик А. Л. Методы распознавания/Москва, 2004. 262 с.
10. Бонгард М. М. Проблема узнавания/Бонгард М. М – М.: Наука, 1967. 320 с.
11. Билибин К.И. Аппаратная реализация нейронных сетей/Билибин К.И., Власов А.И. – М.: Наука, 2000. 345 с.