Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розпізнавання рукописного тексту засобами гібридних нейронних мереж
(тема)

Виконав:
здобувач четвертого року навчання,
групи ІТШІ-21-1
Юлія Приймаченко
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Штучний інтелект
(повна назва освітньої програми)
Керівник ас. Максим Поліїт
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Приймаченко Юлії Володимирівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розпізнавання рукописного тексту засобами гібридних нейронних мереж

затверджена наказом університету від 19 травня 20 25 р. № 378Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 червня 20 25 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані відомих наукових проектів щодо структури та розробки гібридних нейронних мереж для класифікації зображень, дані статей, результати експериментальних досліджень

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі та постановка задачі
2) Розробка моделі ГШНМ в задачах розпізнавання тексту
3) Програмна реалізація

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	19.05.2025	виконано
2	Аналіз предметної галузі і постановка завдання	20.05.2025 – 23.05.2025	виконано
3	Дослідження моделей ШНМ, що застосовуються	24.05.2025 – 28.05.2025	виконано
	для розпізнавання рукописного тексту		
4	Огляд існуючих інструментів для розпізнавання	29.05.2025 – 31.05.2025	виконано
	рукописного тексту		
5	Розробка моделі гібридної ШНМ для задачі	01.06.2025 – 13.06.2025	виконано
	розпізнавання тексту		
6	Тестування та експериментальне дослідження	14.06.2025 – 18.06.2025	виконано
	реалізованих ШНМ		
7	Оформлення пояснювальної записки	19.06.2025 – 20.06.2025	виконано
8	Захист	24.06.2025	виконано

Дата видачі завдання 19 травня 20 25 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Максим Поліїт
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 95 с., 15 рис., 2 дод., 20 джерел.

ГІБРИДНІ НЕЙРОННІ МЕРЕЖІ, КОЛІЗІЯ, РОЗПІЗНАВАННЯ ОБРАЗІВ, РУКОПИСНИЙ ТЕКСТ, ШТУЧНИЙ ІНТЕЛЕКТ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ, CNN.

Об'єкт дослідження – гібридна штучна нейронна мережа для виконання задачі розпізнавання рукописного тексту.

Предмет дослідження – аналіз та обробка методів застосування ГШНМ для розпізнавання рукописного тексту.

Мета роботи – дослідження системи на основі гібридної нейронної мережі, для вирішення задач з розпізнавання рукописного тексту та його джіджиталізації.

Методи дослідження – розробка та навчання системи на основі гібридної нейронної мережі, для вирішення задач з розпізнавання рукописного тексту та його джіджиталізації.

ABSTRACT

Bachelor’s thesis contains: 95 pp., 15 fig., 2 ann., 20 references.

ARTIFICIAL INTELLIGENCE, ARTIFICIAL NEURAL NETWORKS, CNN, COLLISION, HANDWRITING, HYBRID NEURAL NETWORKS, PATTERN RECOGNITION.

The object of research is a hybrid artificial neural network for handwriting recognition.

The subject of the study is the analysis and processing of methods for using a hybrid artificial neural network for handwriting recognition.

Purpose – to study a system based on a hybrid neural network for solving problems of handwriting recognition and its digitalisation.

Research methods – development and training of a system based on a hybrid neural network for solving problems of handwriting recognition and its digitalisation.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ГНМ – гібридна нейронна мережа;

ЗНМ – згорткова нейронна мережа;

ШНМ – штучна нейрона мережа;

MLP – Multi-Layer Perceptron – багатошаровий персептрон;

OCR – Optical Character Recognition – оптичне розпізнавання символів.

ВСТУП

В сучасному світі штучний інтелект є невід’ємною частиною нашого життя. Його існування допомагає нам полегшити життя у багатьох її сферах починаючи від простого пошуку у Google і закінчуючи тим, який фільм обрати щоб подивитися у вихідні. Так було не завжди, тому досі існує багато чого фізично, будь то папери, книги, документи тощо, але вони не існують на просторах інтернету. Папір має властивість зношуватися, втрачати цілісність, якість змісту, а отже це призводить до втрати певної інформації, зачасту дуже важливої інформації. Наразі ми все ще живемо у час переходу від паперів до електронного формату, адже із цим також є певні складнощі.

Для розпізнавання рукописного тексту використовуються технології штучного інтелекту, як розпізнавання образів. Це надалі допоможе відцифрувати паперові документи або навіть створити аудіо на основі цього документу. Перехід до електронного ведення документів призведе до полегшення ведення документообігу.

І досі є люди, які частково або повністю не володіють навичками користування діджітал технологіями такими як комп’ютер, смартфон, планшет тощо. Тому для таких людей і досі легше написання документу від руки, аніж друк на клавіатурі, адже це також потребує певних навичок. Дехто із письменників досі використовують написання на папері для легкості перевтілення думок у текст під час натхнення та створення тексту для книги, адже ми коли пишемо, то менше замислюємося, рука начебто сама пише те, що думаєш.

Принаймні технології не стоять на місці і тому діджиталізація документів стала більш простою та зручною для користувача. Загалом розпізнавання рукописного тексту це досить складний процес, адже він потребує враховувати зміну написання літер у поєднанні з іншими різними літерами, також треба завважити, що одна і та сама людина має різний почерк у різних ситуаціях, емоційних станах, при використанні різних ручок

тощо. Не рідкість ситуації, коли людина не може розібрати власну записку написану від руки, особливо якщо щось було написано наспіх.

На тему розпізнавання рукописного тексту вже існує досить багато різних досліджень та статей та для різних мов, але це все ще залишається відкритим питанням, яке має певні проблеми для ідеальної реалізації. А отже необхідно проаналізувати існуючі методи та виявити недоліки із подальшим аналізом методів їх уникнення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

Реалізація максимально ефективного алгоритму розпізнавання рукописного тексту засобами штучного інтелекту надасть ще більше можливостей та швидкості переходу до діджиталізації документообігу.

Слід зазначити багато переваг такої діджиталізації у багатьох сферах, адже документи більше не можна буде втратити, спалити, загубити тощо. Все буде зберігатися в загальній базі даних, до якої будуть мати доступ всі причетні до неї, що допоможе уникнути вказаних вище неприємних ситуацій із документами.

1.1 Аналіз предметної галузі

Загальною складністю в розпізнаванні рукописного тексту є різне написання літер у поєднанні з кожною іншою літерою, а також різноманіття письма однієї людини в різних умовах. Є потреба у відслідковуванні характерних штрихів при написанні, які будуть проявлятися при будь-яких умовах.

Також необхідний паралельний аналіз сенсу самого речення, щоб підібрати правильне слово, якщо є сумніви та декілька варіантів щодо написання слова в документі.

Вже існуючі системи розпізнавання рукописного тексту вже досягли значних успіхів у якості та швидкості самого процесу, але і досі бувають помилки, неточності, затримки або навіть неможливість розпізнати текст взагалі. Не для усіх проблем можна знайти рішення, адже сильно впливає також якість самого документу (чи не вицвів текст, цілісність самого паперу, і все таки бувають почерки, які не піддаються розумінню та розпізнаванню написаного тексту), якість та стан паперу.

1.2 Постановка задачі

Необхідно дослідити реалізовані засоби розпізнавання рукописного тексту для виявлення існуючих недоліків задля подальшого аналізу та розробки методів розпізнавання рукописного тексту із урахуванням та уникненням вказаних проблем.

1.3 Handwritten Text Recognition

Для розпізнавання рукописного тексту (Handwritten Text Recognition, HTR) найчастіше використовуються різні типи нейронних мереж, які комбінуються для досягнення високої точності.

До найпоширеніших архітектур для виконання поставленої задачі можна віднести:

а) Convolutional Neural Networks (CNN). Призначення: виділення ознак (features) з вхідного зображення. Як працює: CNN ідеально підходять для обробки зображень завдяки здатності виявляти локальні ознаки, як-от лінії, криві чи точки злиття;

б) Recurrent Neural Networks (RNN). Призначення: обробка послідовної інформації, зокрема аналіз текстових рядків. Типи:

– LSTM (Long Short-Term Memory): керує довготривалими залежностями;

– GRU (Gated Recurrent Unit): більш оптимізована версія для швидшого навчання;

в) Connectionist Temporal Classification (CTC). Призначення: перетворення послідовності ознак у текст без необхідності точної розмітки символів. Особливість: дозволяє працювати зі змінною довжиною вхідних та вихідних даних, що є важливим для розпізнавання рукописного тексту;

г) Transformers. Призначення: обробка послідовностей без необхідності рекурсії. Переваги: здатність паралельно обробляти великі обсяги тексту, що покращує швидкість та точність;

д) Hybrid Architectures. Комбінація CNN для екстракції ознак, RNN або Transformers для обробки послідовностей і CTC для перетворення послідовностей в текст.

1.3.1 Convolutional Neural Networks (CNN)

CNN – це клас нейронних мереж, що спеціалізуються на обробці даних із сітковою структурою, таких як зображення. Вони широко застосовуються для завдань комп'ютерного зору, зокрема для розпізнавання рукописного тексту, класифікації зображень, виявлення об'єктів та сегментації.

Згорткова нейронна мережа складається з 6-ти шарів, а саме:

- вхідний шар;
- згортковий шар;
- функція активації;
- шар підвибірки;
- шар нормалізації;
- повнозв’язний шар.

Вхідний шар містить в собі сітку пікселів зображення. Основний шар є згортковий шар, який застосовує фільтри для виявлення локальних ознактаких як краї, кути чи текстури. Найпопулярніша функція активації – ReLU (Rectified Linear Unit), яка замінює всі негативні значення нулями:

$$f(x) = \max(0, x). \tag{1.1}$$

В свою чергу шар підвибірки зменшує розмірність даних для зниження обчислювальної складності. Найчастіше використовуються:

– Max Pooling: вибирає максимальне значення в області;

– Average Pooling: обчислює середнє значення блоку.

Шар нормалізації прискорює навчання і стабілізує роботу мережі шляхом нормалізації вхідних даних кожного шару. Повнозв’язний шар з'єднує всі нейрони попереднього шару з кожним нейроном цього шару, що дозволяє здійснювати остаточну класифікацію чи регресію.

Попри переваги CNN є досить вагомі недоліки, такі як необхідність потужних графічних процесорів через великі обчислювальні ресурси та потреба у великій кількості даних для навчання задля забезпечення ефективності та високої точності роботи.

1.3.2 Recurrent Neural Networks (RNN)

RNN – це тип нейронних мереж, які добре підходять для обробки послідовних даних, таких як тексти, звуки або часові ряди. Їх ключова особливість – здатність зберігати та використовувати інформацію з попередніх станів, що дозволяє враховувати контекст.

На відміну від звичайних нейронних мереж, RNN мають зворотні зв’язки, що дозволяє передавати інформацію від одного кроку обчислення до наступного.

Спочатку мережа розгортається в часі, тобто вона бере елементи послідновності x_t (наприклад, символи чи слова), обчислює прихований стан h_tі передає його на наступний крок:

$$h_t = f(W_h \cdot h_{t-1} + W_x \cdot x_t + b), \tag{1.2}$$

де W_h – ваги для попереднього стану;

W_x – ваги для поточного входу;

b – зміщення;

f – функція активації (зазвичай tanh або ReLU).

Далі на кожному кроці мережа генерує вихідний сигнал y_t:

$$y_t = g(W_y \cdot h_t + b_y), \quad (1.3)$$

де g – функція активації, наприклад, Softmax для класифікації.

Всього є декілька типів рекурентних нейронних мереж:

а) Basic RNN. Найпростіша архітектура. Має обмеження: труднощі із запам’ятовуванням довгострокових залежностей через проблему зникнення градієнта;

б) LSTM (Long Short-Term Memory). Розширення RNN, що долає проблему зникнення градієнта. Використовує осередки пам’яті та механізми керування (гейти):

– вхідний гейт: контролює, яку інформацію з поточного вводу записати;

– гейт забування: визначає, яку інформацію забути;

– вихідний гейт: контролює, яку інформацію передати далі;

в) GRU (Gated Recurrent Unit). Спрощена версія LSTM, має менше параметрів, що робить її швидшою. Також ефективно працює з послідовностями;

г) Bidirectional RNN. Два RNN, які аналізують дані в обох напрямках: від початку до кінця і навпаки. Це особливо корисно для тексту, де контекст важливий в обох напрямах.

Головним недоліком рекурентної нейронної мережі є проблема зникаючого або вибухаючого градієнту. Коли градієнт стає занадто малим або великим, навчання стає складним.

LSTM і GRU вирішують цю проблему. Також RNN обробляють дані послідовно, що уповільнює навчання порівняно з архітектурами, як-от Transformers.

1.3.3 Connectionist Temporal Classification (CTC)

CTC – це спеціальна функція втрат, яка використовується для навчання моделей розпізнавання послідовностей. Її головна перевага – можливість працювати зі змінною довжиною вхідних та вихідних послідовностей, без необхідності точного вирівнювання (alignment) між ними.

CTC дозволяє моделі навчитися зіставляти вхідну послідовність (наприклад, зображення рукописного тексту чи аудіо сигнал) із вихідною послідовністю (текст або мова) без попереднього маркування кожного фрагмента вхідних даних:

– вхід: послідовність ознак $X = [x_1, x_2, \ \dots \ , x_T]$, отриманих із CNN або RNN;

– вихід: послідовність символів $Y = [y_1, y_2, \ \dots \ , y_U]$, $U \leq T$.

CTC вирішує проблему через додавання спеціального символу blank є, який означає «нічого» (пустий символ). Це дозволяє розв’язати проблему непостійної довжини.

CTC найчастіше використовується в задачах, де немає прямого вирівнювання між входом і виходом:

– розпізнавання рукописного тексту: наприклад, зіставлення рядків тексту із зображеннями, навіть якщо символи на зображенні розташовані нерівномірно;

– автоматичне розпізнавання мови (ASR): перетворення аудіо-сигналу в текст;

– підписання жестів: перетворення відео із жестами в текстові підписи.

Для навчання CTC використовується алгоритм з трьох єтапів, а саме:

– використовуються CNN або RNN для отримання послідовності ознак із вхідних даних;

– CTC автоматично знаходить найкраще вирівнювання шляхів через

символ є;

– обчислення втрат. Обчислюється негативний логарифм ймовірності правильного результату.

Головною перевагою є відсутність необхідності попереднього вирівнювання між входом та виходом. Також CTC підтримує змінну довжину послідовностей.

Але не дивлячись на всі переваги, CTC має і певні недоліки, такі як обмежена точність для довгих вихідних послідовностей. А також досить складний процес декодування.

1.3.4 Transformers

Transformers – це архітектура нейронної мережі, яка революціонізувала обробку природної мови (NLP), комп'ютерний зір та багато інших областей. Вперше її представили у статті «Attention is All You Need» (2017) від Google. Вона вирішує проблему послідовної обробки вхідних даних, характерну для RNN, і дозволяє обробляти дані паралельно.

Transformers складається з Self-Attention (Механізм уваги), Multi-Head Self-Attention, Позиційне кодування (Positional Encoding).

Self-Attention (Механізм уваги) це ядро архітектури Transformer, яке дозволяє моделі фокусуватися на різних частинах вхідної послідовності при обробці кожного елемента.

Для кожного елемента послідовності обчислюються:

– Query (запит) – «Що я хочу знайти?»;

– Key (ключ) – «Що я можу запропонувати?»;

– Value (значення) – «Яка інформація зі мною пов'язана?».

Обчислюється оцінка подібності між Query і Key для кожного елемента, щоб визначити, які частини послідовності є найбільш релевантними для кожного елемента.

$$Attention\ (Q,K,V) = softmax\ \left(\frac{QK^T}{\sqrt{d_k}}\right)V\ , \quad (1.4)$$

де Q, K, V – матриці запитів, ключів і значень;

d_k – розмірність ключів, що використовується для масштабування.

Multi-Head Self-Attention замість однієї голови уваги використовує кілька (multi-head). Це дозволяє мережі фокусуватися на різних аспектах послідовності одночасно.

$$MultiHead(Q,K,V) = Concat(head_1, \dots, head_h)W^O, \quad (1.5)$$

де h – кількість «голів».

Позиційне кодування (Positional Encoding) оскільки модель обробляє послідовність як набір елементів без урахування порядку, додається інформація про позиції. Для цього використовується синусоїдальна функція.

$$PE_{(pos,2i)} = \sin(\frac{pos}{10000^{\frac{2i}{d}}}), \quad (1.6)$$

$$PE_{(pos,2i)} = \cos(\frac{pos}{10000^{\frac{2i}{d}}}). \quad (1.7)$$

Ці значення додаються до входу, щоб модель могла враховувати порядок.

Transformer складається з двох основних блоків:

а) енкодер (Encoder):

– складається з кількох ідентичних шарів;

– кожен шар має: механізм Multi-Head Self-Attention, Feed-forward нейронну мережу;

– використовується для перетворення вхідної послідовності у

приховане представлення;

б) декодер (Decoder):

– також складається з кількох ідентичних шарів;

– має: Masked Multi-Head Self-Attention (щоб модель не бачила майбутні символи під час генерації), механізм уваги до виходу енкодера;

– Feed-forward нейронну мережу;

– генерує вихідну послідовність.

Transformer використовується для обробки природної мови (NLP), комп’ютерного зору, часових рядів, мультимодальних даних.

До переваг Transformer можна віднести паралельність, гнучкість, універсальність.

1.3.5 Hybrid Architectures

Гібридні архітектури в машинному навчанні поєднують різні типи нейронних мереж або методів, щоб використовувати переваги кожного з них. Це дозволяє створювати більш гнучкі й ефективні системи для складних завдань.

Гібридні підходи можуть об'єднувати, наприклад, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Transformers, і навіть традиційні алгоритми машинного навчання (наприклад, методи опорних векторів чи дерев рішень).

Головними перевагами можна назвати гнучкість та ефективність. Як недоліки є складність реалізації, вимогливість до обчислювальних ресурсів та чутливість до узгодження компонентів.

Гібридні архітектури використовуються для обробки відео, генерації тексту за зображеннями, аналіз даних електрокардіограм.

1.4 Моделі розпізнавання Handwritten Text Recognition

Розпізнавання Handwritten Text Recognition відноситься до класу задач розпізнавання образів.

Розпізнавання образів можливе за допомогою порівняння із шаблонами, статистична класифікація, ШНМ та синтаксична і структурна відповідність.

Штучні нейронні мережі – це система простих процесорів, які між собою поєднані та мають взаємодію (рисунок 1.1 [4]). Взаємодія процесорів складається у періодичній передачі сигналів один одному. Головною перевагою штучних нейронних мереж є можливість навчання, на відміну від звичайних алгоритмів розпізнавання образів, адже це означає можливість вдосконалювати результати виконання задач [4].

Задля прискорення виконання процесу обробки вхідної інформації, штучні нейронні мережі використовують паралельну обробку інформації. Розглянемо просту структуру штучних нейронних мереж на рисунку 1.1.

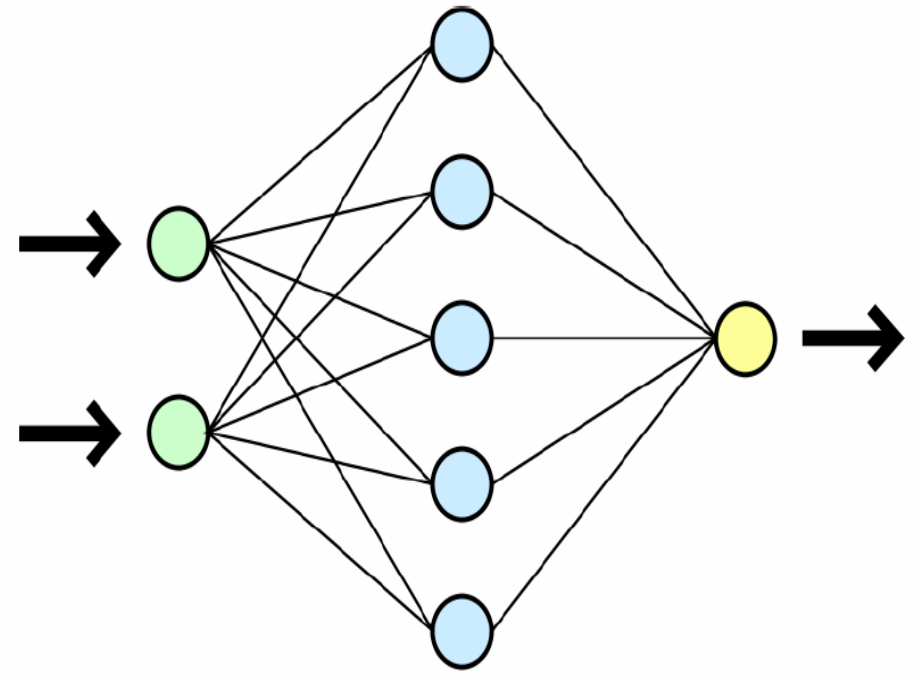

Рисунок 1.1 – Схема простої ШНМ

Зеленим кольором позначені вхідні нейрони, блакитним – приховані нейрони, жовтим – вихідний шар.

1.4.1 Perceptron

Персептрон – це історично важливий алгоритм, який заклав основу для сучасних досліджень у галузі штучного інтелекту та глибокого навчання. Хоча його можливості обмежені, він є ключовим кроком у розвитку нейронних мереж.

Персептрон отримує кілька входів $x_1, x_2, \ldots, x_n$, які проходять через вагові коефіцієнти $w_1, w_2, \ldots, w_n$ обчислює їх зважену суму, додає зміщення $(bias)b$, а потім застосовує активаційну функцію для прийняття рішення.

$$y = f(\sum_{i-1}^{n} w_i x_i + b), \tag{1.8}$$

де x_i – вхідні дані;

w_i – ваги, що відповідають важливості кожного входу;

b – зміщення, яке дозволяє зміщувати поріг активації;

f – активаційна функція.

В оригінальному персептроні використовується порогова функція.

$$f(x) = \begin{cases} 1, \text{якщо } x \geq 0 \\ 0, \text{якщо } x < 0 \end{cases}. \tag{1.9}$$

Ця функція визначає, чи буде нейрон активований (видасть 1), чи не буде (видасть 0).

Персептрон навчається за допомогою правила оновлення ваг, яке коригує ваги залежно від помилки у передбаченні. Процес навчання включає кілька етапів:

– ініціалізація. Всі ваги w_i і зміщення b ініціалізуються (зазвичай випадковими значеннями або нулями);

– прогноз. Обчислюється вихідний сигнал y за вхідними даними;

– оновлення ваг. Якщо передбачення неправильне, ваги коригуються за формулою:

$$w_i \leftarrow w_i + \Delta w_i \tag{1.10}$$

$$\Delta w_i = \eta \cdot (y_{true} - y_{pred}) \cdot x_i \tag{1.11}$$

де η – коефіцієнт навчання (learning rate);

y_{true} – справжній клас;

y_{pred} – передбачений клас.

Повторення. Процес повторюється для всіх прикладів у навчальному наборі до досягнення заданої точності або вичерпання кількості ітерацій.

Персептрон це свого роду схема, яка допомагає в моделюванні складного процесу мислення та сприйняття людиною навколишньої інформації.

Також було проведено експерименти задля отримання психологічно значимої інформації, а саме під час надання моделі двох різноманітних стимулів, вона повинна була відреагувати на них відмінно одна від одної. Такий експеримент надає розуміння щодо можливостей моделей до спонтанного розрізнення уникаючи втручання учителя або ж навпаки під примусом учителя задля проведення конкретної потрібної класифікації.

Сам процес навчання персептрону полягає в отриманні певної послідовності деяких образі, які є безпосередньо представниками різних розрізнених класів.

Під час виникнення правильної реакції відбувається її посилення, щоб після отримання контрольного стимулу можливо було визначити вирогідність отримання правильної реакції на стимули певного класу.

Надалі перевіряється чи збіглися між собою контрольний стимул та образ із навчальної послідовності. Після перевірки маємо два варіанти подальших дій, а саме:

– при виявленні розбіжності контрольного стимулу із образами з навчальної послідовності, тоді вважається, що в експерименті присутні окрім чистого розрізнення ще й певні елементи узагальнення;

– при наявності реакції певних наборів сенсорних елементів на контрольний стимул, набори є розразненними від тих, які було активовано раніше під впливом інших стимулів одного класу, тоді вважається, що експеримент був чистого узагальнення.

Необхідно також враховувати, що самі по собі персептрони не мають здатність та схильність до чистого узагальнення, проте це не має нікого впливу на можливість вирішення задач з розрізнення, тим паче, коли контрольний стимул максимально наближений до одного з образів, на кому персептрон вже набув досвіду [5].

Багатошаровий персептрон (MLP, Multi-Layer Perceptron) – це тип штучної нейронної мережі, який складається з кількох шарів нейронів, організованих у три основні типи шарів:

– вхідний шар: приймає вхідні дані;

– приховані шари: виконують основну обробку даних і витягнення ознак;

– вихідний шар: генерує результат моделі (класифікація, регресія тощо).

На відміну від простого персептрона, MLP здатний вирішувати нелінійні задачі завдяки використанню прихованих шарів і нелінійних активаційних функцій.

Розглянемо більш детально архітектуру багатошарового персептрона:

а) вхідний шар:

– кількість нейронів у цьому шарі дорівнює кількості ознак у даних;

– кожен нейрон приймає одну ознаку як вхід;

б) приховані шари:

– складаються з одного або кількох шарів нейронів;

– кількість нейронів у кожному шарі може бути довільною;

– кожен нейрон прихованого шару з’єднаний з усіма нейронами попереднього і наступного шарів;

в) вихідний шар. Кількість нейронів залежить від типу задачі:

– для класифікації: кількість класів;

– для регресії: 1 нейрон;

г) активаційні функції:

– для прихованих шарів: зазвичай використовуються нелінійні функції, такі як:

$$ReLU(f(x) = \max(0, x)), \tag{1.12}$$

$$Sigmoid\ (f(x) = \frac{1}{1+e^{-x}}, \tag{1.13}$$

$$Tanh\ (f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}; \tag{1.14}$$

д) для вихідного шару:

– Softmax (для багатокласової класифікації);

– лінійна функція (для регресії).

Вхідні дані проходять через кілька шарів нейронів, причому кожен шар обчислює:

$$a^{(l)} = f(W^{(l)} \cdot a^{(l-1)} + b^{(l)}, \tag{1.15}$$

де $a^{(l)}$ – вихід l-го шару (активації);

$W^{(l)}$ – матриця ваг для l-го шару;

$b^{(l)}$ – вектор зміщень для l-го шару;

f – активаційна функція.

Multi-Level Perceptron (MLP) навчається на основі зворотного поширення помилки, тому така модель і є найбільш застосованою для розпізнавання рукописного тексту.

Процес навчання такої моделі виглядає наступним чином:

а) прямий прохід (Forward Propagation):

– вхідні дані проходять через всі шари мережі;

– обчислюється прогноз моделі;

б) обчислення функції втрат (Loss Function):

– порівнює прогноз моделі з істинними значеннями;

– приклади функцій втрат: для класифікації – крос-ентропія, для регресії – середньоквадратична помилка (MSE);

в) зворотне поширення помилки (Backpropagation):

– градієнти обчислюються за допомогою алгоритму зворотного поширення, щоб оновити ваги;

– використовується метод градієнтного спуску (або його варіанти, наприклад, Adam, RMSprop);

г) оновлення ваг:

– ваги кожного шару оновлюються за формулою:

$$W^{(l)} \leftarrow W^{(l)} - \eta \cdot \frac{\partial \text{Loss}}{\partial W^{(l)}}, \tag{1.16}$$

де η – швидкість навчання (learning rate);

д) повторення: процес повторюється для всіх епох або до досягнення прийнятної точності.

До переваг MLP можна віднести гнучкість, можливість роботи з нелінійними задачами та здатність до паралельної обробки.

До недоліків відноситься складність навчання, чутливість до даних та потреба в обчислювальних ресурсах.

1.4.2 Convolutional neural network (CNN)

Convolutional Neural Networks (CNN) – це спеціалізований тип штучних нейронних мереж, який ефективно працює з даними, що мають просторову або часову структуру, такими як зображення, відео, аудіо або текст. CNN особливо популярні у задачах обробки зображень завдяки їх здатності автоматично витягувати просторові ознаки з вхідних даних.

CNN складається з семи шарів, а саме: вхідний шар, шар згортки (Convolutional Layers), шар активаційних функцій, шар підвибірки (Pooling Layers), шар нормалізації, повнозв'язні шари (Fully Connected Layers), вихідний шар.

Сам процес навчання Convolutional Neural Networks (CNN) складається з наступних етапів:

а) прямий прохід (Forward Propagation):

– вхідні дані проходять через всі шари мережі, обчислюючи проміжні ознаки та остаточний результат;

б) функція втрат:

– оцінює, наскільки прогноз відрізняється від істинного значення.

– приклади: крос-ентропія (Cross-Entropy Loss) для класифікації, MSE (Mean Squared Error) для регресії;

в) зворотне поширення помилки (Backpropagation):

– обчислює градієнти функції втрат щодо ваг кожного шару, використовуючи правило ланцюга;

г) оновлення ваг:

– ваги оновлюються за допомогою оптимізаторів, таких як SGD, Adam, RMSprop тощо.

Convolutional Neural Networks (CNN) застосовуються для виконання наступних задач:

а) обробка зображень:

- розпізнавання об'єктів;
- сегментація зображень;
- детектування об'єктів;

б) відеоаналіз:

- аналіз відео потоків;
- розпізнавання жестів;

в) обробка тексту:

- аналіз тональності;
- класифікація текстів (у поєднанні з 1D-CNN);

г) медична діагностика:

- аналіз рентгенівських знімків;
- виявлення аномалій на МРТ або КТ;

д) аудіоаналіз:

- розпізнавання мовлення;
- класифікація звуків.

До головних переваг CNN можна віднести, що CNN автоматично виявляє важливі ознаки у даних, усуваючи необхідність ручного проектування ознак. Також завдяки пулингу та згортковим шарам CNN добре працюють із даними, які можуть змінюватися просторово. Глибокі CNN можуть вивчати складні залежності у великих наборах даних.

При тому у CNN є і недоліки, а саме те, що для якісного навчання потрібні великі набори даних. CNN потребують значних апаратних ресурсів, особливо для навчання. А також вибір розміру ядер, кількості шарів та інших параметрів впливає на продуктивність.

Завдяки згортковим штучним нейронним мережам був значний стрибок у розвитку задач із комп’ютерного зору та розпізнаванн образів. Також згорткові ШНМ знаходять використання у задачах розпізнавання мови, обробки часових рядів для аналізу змісту текстів та розпізнавання різноманітних образів.

У згорткових нейронних мережах використовується пряме

поширення, тобто послідовна активація шарів. Також треба зазначити залежність активації наступного шару від активації попереднього шару. Необхідно враховувати можливість паралельного обчислення активацій в межах одного шару.

Для навчання нейронної мережі необхідне використання досить великої кількості даних у навчальних вибірках. Особливістю згорткової ШНМ є організація нейронів на початкових рівнях у специфічну структуру: на перших шарах нейрони групуються у зображення певного розміру, які часто називають картами ознак. Різні карти в межах одного шару відповідають різним типам нейронів, що реагують на різні характеристики зображень [7].

Є декілька видів функцій активації, розглянемо їх:

– ReLU і Leaky ReLU – найпоширеніші функції для прихованих шарів завдяки простоті та ефективності;

– Softmax – стандарт для вихідного шару у багатокласовій класифікації;

– Sigmoid – використовується для бінарної класифікації або в специфічних задачах;

– Swish і ELU – перспективні функції, які можуть покращити продуктивність для складних задач.

Кінцевий вибір функції активації залежить від задачі, архітектури моделі та обчислювальних ресурсів.

1.4.3 Кохонена

Нейронна мережа Кохонена, або карта самоорганізації (Self-Organizing Map, SOM), – це один із типів штучних нейронних мереж, який належить до класу неперебіркових (unsupervised) алгоритмів навчання. SOM використовується для кластеризації, візуалізації багатовимірних даних та зменшення розмірності. Її основна мета – перетворити вхідні

багатовимірні дані у двовимірну карту, зберігаючи при цьому топологічні зв'язки між вхідними даними.

Карта Кохонена складається з двовимірної решітки нейронів. Кожен нейрон має ваговий вектор тієї ж розмірності, що й вхідні дані. Нейрони на карті зберігають просторові зв'язки, тобто схожі вхідні дані активують нейрони, розташовані поруч. У процесі навчання мережа змінює ваги нейронів так, щоб вони відображали структуру вхідних даних. Після навчання кожен нейрон відповідає за певний кластер вхідних даних.

Архітектура SOM складається з трьох складових: вхідний шар, вихідний шар та ваги.

Вхідний шар містить вектори даних $X = [x_1, x_2, \dots, x_d]$, де d – кількість ознак.

Вихідний шар (решітка нейронів) представлен у вигляді двовимірної сітки (наприклад, прямокутної або шестикутної).

Кожен нейрон має ваговий вектор $W = [w_1, w_2, \dots, w_d]$, який ініціалізується випадково.

Процес навчання SOM складається з кількох етапів:

а) ініціалізація:

– задається топологія карти (наприклад, розмір сітки);

– ініціалізуються ваги нейронів випадковими значеннями;

б) вибір вхідного вектора:

– випадковий вектор із набору даних обирається для навчання;

в) пошук нейрона, що переміг (Winner Node):

– обчислюється відстань (наприклад, за Евклідовою метрикою) між вхідним вектором і вагами кожного нейрона:

$$Distance = \|X - W\|; \tag{1.17}$$

– нейрон із найменшою відстанню стає нейроном-переможцем (Best Matching Unit, BMU).

г) оновлення ваг:

– ваги BMU та його сусідів оновлюються за правилом:

$$W_i(t+1) = W_i(t) + \eta_{(t)} \cdot h_{(i,BMU,t)} \cdot (X - W_i(t)), \quad (1.18)$$

де $\eta_{(t)}$ – швидкість навчання (learning rate), яка зменшується з часом; $h_{(i,BMU,t)}$ – функція сусідства, яка визначає вплив BMU на сусідні нейрони;

д) повторення:

– процес повторюється для всіх вхідних векторів протягом кількох епох.

Мережа Кохонена також застосовується у задачах розпізнавання зображень. Досить часто її роботу демонструють на прикладі розпізнавання рукописних цифр.

1.4.4 Restricted Boltzmann Machin

Обмежена машина Больцмана (RBM) – це стохастична нейронна мережа, яка використовується для моделювання розподілів і знаходження прихованих закономірностей у даних. Вона є двошаровою моделлю, що складається з видимих і прихованих нейронів, без з'єднань між нейронами одного шару. RBM широко застосовується для задач, пов'язаних із зменшенням розмірності, кластеризацією, генерацією даних і попереднім навчанням глибоких нейронних мереж.

Обмежена машина Больцмана (RBM) складається з двох шарів та ваг: видимий шар (Visible layer) та прихований шар (Hidden layer).

RBM визначає спільний розподіл між видимими v і прихованими h нейронами через енергію системи:

$$E(v,h) = -\sum_i b_i v_i - \sum_j c_j h_j - \sum_{ij} v_i W_{ij} h_j, \quad (1.19)$$

де v_i – стан і-го нейрона видимого шару;

h_j – стан j-го нейрона прихованого шару;

W_{ij} – вага між нейронами v_i та h_j;

b_i, c_j – зміщення (bias) для v_i та h_j відповідно.

Ціль навчання – знайти ваги W, що мінімізують розбіжність між реальним розподілом даних і розподілом, який моделює RBM. Для цього використовується метод максимізації правдоподібності [8].

Отже, RBM є потужною моделлю для навчання прихованих представлень даних.

Вона має широку область застосувань, включаючи зменшення розмірності, кластеризацію та рекомендаційні системи.

Хоча RBM поступилася місцем сучасним моделям (наприклад, автоенкодерам і трансформерам), вона залишається важливим інструментом у глибокому навчанні.

1.4.5 Hybrid Neural Networks

Гібридні нейронні мережі (ГШНМ) – це архітектури, які об'єднують кілька типів нейронних мереж або інтегрують інші підходи машинного навчання, статистики чи моделювання для розв'язання складних задач. Такий підхід дозволяє використовувати сильні сторони різних моделей і компенсувати їхні слабкості.

Мета гібридних нейронних мереж – поєднати різні алгоритми та методи для:

– підвищення точності: комбінування моделей дозволяє краще узагальнювати дані;

– адаптації до складних задач: використання різних компонентів для різних підзадач;

– гнучкості: інтеграція спеціалізованих моделей у загальну систему.

Головною перевагою ГШНМ є можливість обрати структуру, яка безпосередньо буде впливати на швидкість програмної реалізації. Така варіативність дає можливість використовувати ГШНМ для вирішення задач в реальному часі.

ГШНМ можна використовувати для вирішення таких задач як аналіз тексту, відео, медичні діагностики, системи рекомендацій, обробка природньої мови та аналіз фінансових даних.

Недоліки гібридних нейронних мереж:

– складність: проектування, реалізація та навчання гібридних моделей можуть бути складними;

– вимоги до ресурсів: такі моделі зазвичай вимагають більше обчислювальної потужності;

– чутливість до параметрів: потребують ретельного налаштування гіперпараметрів;

– ризик перенавчання: у разі недостатньої кількості даних або невідповідного налаштування моделі.

Можна використати певні алгоритми для визначення оптимальної кількості поєднаних нейронних мереж задля правильної організації вирішення задач із максимальною точністю.

Зазвичай такі алгоритми використовуються для класифікації зображень, як от зображення із різноманітними захворюваннями шкіри [9].

1.5 Handwritten Text Recognition

Розпізнавання рукописного тексту (Handwritten Text Recognition, HTR) є складною задачею, яка потребує спеціалізованих інструментів та алгоритмів.

З розвитком штучного інтелекту (ШІ) та машинного навчання, з'явилося багато програмних рішень і бібліотек для вирішення цієї задачі. OCR-системи (Optical Character Recognition), призначені для автоматичного

введення документів в комп'ютер. Огляд основних інструментів для HTR наведено нижче.

1.5.1 Tesseract OCR

Tesseract OCR – це один із найпопулярніших інструментів для оптичного розпізнавання тексту (Optical Character Recognition, OCR), який використовується для перетворення зображень тексту в цифровий формат. Цей інструмент відкритого коду спочатку був розроблений Hewlett-Packard, а згодом підтримка перейшла до Google.

Перевагами можна вважати наявність відкритого коду, підтримує багато мов, підтримує LSTM-рекурентні нейронні мережі, основна орієнтація на друкований текст, але є підтримка рукописного тексту, Tesseract підтримує вхідні файли у багатьох різних форматах, може виводити результати у текстовому форматі або форматах HOCR/ALTO для збереження розмітки.

Робота Tesseract OCR полягає в наступних етапах: передобробка зображення, сегментація тексту, розпізнавання символів, постобробка.

Tesseract OCR зазвичай використовується для автоматизації документообігу, аналізу зображень, перетворення рукописного тексту та інтеграція у різні мови програмування.

Головним недоліком є обмежена точність при роботі з неякісними зображеннями або складним рукописним текстом.

1.5.2 Microsoft Azure Cognitive Services – Computer Vision API

Microsoft Azure Cognitive Services – Computer Vision API є хмарним сервісом, який дозволяє аналізувати зображення за допомогою інструментів штучного інтелекту. Він надає широкий набір функцій, включаючи розпізнавання рукописного тексту, друкованих документів, а також аналіз

візуального вмісту.

Microsoft Azure Cognitive Services – Computer Vision API використовується для таких задач як розпізнавання тексту (OCR), аналіз зображень, робота з документами, генерація описів, оптимізація для мобільних пристроїв, детекція об’єктів.

Розглянемо детальніше кожний етап:

а) розпізнавання тексту (OCR):

– підтримує друкований і рукописний текст;

– визначає позицію тексту на зображенні;

– може працювати з різними мовами;

б) аналіз зображень:

– розпізнає об'єкти, сцени та теги;

– визначає колірну палітру зображення;

– ідентифікує обличчя (включаючи демографічні дані, як-от вік і стать);

в) робота з документами:

– підтримує автоматичне зчитування форм і таблиць;

– працює з багатосторінковими документами (PDF, TIFF тощо);

г) генерація описів:

– автоматично генерує текстові описи для зображень.;

– підходить для інклюзивності, наприклад, для людей із порушеннями зору;

д) оптимізація для мобільних пристроїв:

– може обробляти фото, зроблені на смартфон, навіть якщо текст розташований під кутом;

е) детекція об’єктів:

– визначає конкретні об’єкти на зображенні (наприклад, автомобілі, меблі, їжу).

Робота Computer Vision API складається з таких етапів:

а) завантаження зображення:

– зображення може бути завантажене через локальний файл або URL;

б) вибір функції аналізу:

– розпізнавання тексту (OCR);

– витяг даних із форм;

– аналіз сцен чи об'єктів;

в) обробка даних у хмарі:

– зображення обробляється на серверах Microsoft Azure за допомогою попередньо навчених моделей ШІ;

г) повернення результатів:

– API повертає результати у форматі JSON, який містить розпізнаний текст, координати областей тексту або опис зображення.

Microsoft Azure Cognitive Services використовуються зазвичай для вирішення таких задач:

а) розпізнавання тексту з фотографій:

– API може зчитувати текст із документів, рекламних щитів або фотографій на телефоні, навіть якщо зображення нечітке;

б) автоматизація документообігу:

– зчитування тексту з рахунків, контрактів і форм;

– автоматична організація та пошук інформації;

в) аналіз візуального вмісту:

– ідентифікація об’єктів на зображеннях для інтернет-магазинів;

– генерація метаданих для пошуку у фотобібліотеках.

Розглянемо переваги Microsoft Azure Cognitive Services. До переваг можна віднести широкий функціонал, адже окрім OCR, сервіс підтримує аналіз об'єктів, генерацію описів і детекцію облич. Також сервіс підтримує багато мов, легко інтегрується з іншими продуктами Microsoft та завдяки використанню сучасних моделей нейронних мереж сервіс володіє високою точністю роботи.

Попри переваги є і недоліки використання та роботи сервісу, а саме такі як залежність від наявності підключення до інтернету, сервіс платний та він не працює офлайн.

Отже, Microsoft Azure Cognitive Services – Computer Vision API – це потужний і гнучкий інструмент для роботи з зображеннями та текстом. Він ідеально підходить для бізнесів, яким потрібні хмарні рішення для обробки тексту й аналізу візуального контенту. Завдяки інтеграції з іншими сервісами Azure, цей API забезпечує високий рівень продуктивності та зручності використання.

1.5.3 ABBYY FineReader

ABBYY FineReader – це один із провідних програмних продуктів для оптичного розпізнавання тексту (OCR), розроблений компанією ABBYY. Програма дозволяє перетворювати скановані документи, PDF-файли, фотографії та інші зображення тексту у редагований та пошуковий формат. FineReader відомий своєю високою точністю розпізнавання тексту та багатим функціоналом для обробки документів.

Программа ABBYY FineReader є досить багатофункціональною. До основних функцій можна віднести:

– OCR (оптичне розпізнавання тексту): розпізнає текст із друкованих та сканованих документів, зображень і PDF, підтримує понад 190 мов, включаючи українську;

– редагування PDF: дозволяє редагувати текст і зображення в PDF-документах без конвертації, додає примітки, позначки та підписи до PDF-файлів;

– конвертація документів: перетворює PDF і зображення у редаговані формати, такі як Word, Excel, PowerPoint або текст; підтримує збереження у форматах EPUB і HTML;

– робота з табличними даними: ефективно розпізнає таблиці та

зберігає їх структуру під час експорту в Excel;

– пошук тексту в документах: дозволяє створювати PDF-файли з функцією повнотекстового пошуку;

– порівняння документів: функція порівняння дозволяє визначати зміни між двома версіями документа, навіть якщо одна з них у сканованому вигляді;

– автоматизація обробки документів: використання сценаріїв для пакетної обробки документів (у корпоративній версії);

– розпізнавання рукописного тексту: може розпізнавати друкований рукописний текст, але точність залежить від чіткості письма.

Программа працює за наступним алгоритмом: відбувається сканування або завантаження документу, программа аналізує його структуру, розпізнає текст, результат розпізнавання можна експортувати у потрібний формат [10].

ABBYY FineReader має високу точність розпізнавання, підтримує багато мов. Також до переваг можна віднести інтуїтивно зрозумілий інтерфейс, що надає зручність у використанні. Можливість вибору форматів експорту, регіонів для розпізнавання та інших параметрів. Програма може одночасно працювати з великими обсягами документів, а також має можливість захисту даних за допомогою додавання паролів.

Головними недоліками ABBYY FineReader є висока вартість, для роботи з великими документами програма потребує потужного апаратного забезпечення. Хоча програма може працювати з друкованим рукописним текстом, розпізнавання ручного письма є складним завданням.

Розглянемо приклади використання программи:

– юридична сфера: конвертація сканованих договорів у редагований формат для подальшого використання;

– освіта: розпізнавання підручників, статей і рукописних заміток;

– медицина: автоматизація роботи з медичними картами та бланками;

– бізнес: автоматична обробка рахунків-фактур, контрактів і

фінансових документів.

Отже, ABBYY FineReader – це потужне рішення для обробки тексту, яке забезпечує високу точність OCR і зручність роботи. Він ідеально підходить для бізнесу, освіти та юриспруденції, де потрібна ефективна робота з документами. Попри вартість, програма виправдовує себе завдяки широкому функціоналу та багатозадачності.

1.5.4 FormXtra Capture

FormXtra Capture – це програмна платформа для обробки та витягу даних із документів, розроблена компанією Parascript. Вона поєднує передові технології оптичного розпізнавання тексту (OCR), інтелектуального розпізнавання символів (ICR), а також машинного навчання для автоматизації обробки паперових, сканованих і цифрових документів.

FormXtra Capture орієнтована на автоматизацію збору даних із форм, таблиць, рукописного тексту та складних документів, забезпечуючи високу точність і масштабованість у бізнес-середовищі.

FormXtra Capture є досить багатофункціональною. До основних можливостей можна віднести:

а) розпізнавання друкованого та рукописного тексту:

– підтримує OCR для друкованого тексту та ICR для рукописного тексту;

– визначає текст навіть на складних фонах або у зображеннях низької якості;

б) обробка форм:

– автоматично розпізнає поля, ключі та значення у структурованих формах;

– підтримує настроювані шаблони для обробки різних типів документів;

в) розпізнавання таблиць:

– екстракція даних із табличних документів із збереженням структури;

г) автоматизація обробки:

– вбудовані алгоритми машинного навчання дозволяють автоматично класифікувати документи за їх типами;

д) верифікація даних:

– вбудовані механізми перевірки забезпечують високу точність даних, наприклад, перевірку цифр у фінансових документах;

е) інтеграція:

– легко інтегрується з іншими системами, такими як ERP, CRM чи хмарні сервіси;

ж) підтримка різних форматів документів:

– працює з PDF, TIFF, JPEG, PNG та іншими популярними форматами.

Алгоритм роботи FormXtra Capture полягає в наступному [11]:

– сканування або імпорт документів: система отримує зображення документів через сканери, багатофункціональні пристрої або завантаження з локального диска чи хмарного сервісу;

– класифікація документів: алгоритми автоматично визначають тип документа (наприклад, рахунок-фактура, контракт чи форма);

– розпізнавання тексту та даних: OCR та ICR екстрагують текст із полів, таблиць і зон документа;

– верифікація: розпізнані дані проходять перевірку на коректність, використовуючи вбудовані алгоритми або бізнес-правила;

– експорт: дані експортуються у бажаному форматі, наприклад, Excel, CSV, XML чи інтегруються у базу даних або бізнес-додаток.

FormXtra Capture володіє досить високою точністю розпізнавання, постійно вдосконалюється через навчання на нових даних, що підвищує якість обробки з часом. Програма має можливість адаптуватися під різні

типи документів завдяки налаштовним шаблонам. Автоматизація обробки документів знижує потребу у ручній роботі. Підходить як для малого бізнесу, так і для великих корпорацій з великим обсягом документів.

В той же час FormXtra Capture має ряд недоліків, таких як досить висока ціна, інтеграція та початкова настройка можуть вимагати технічних знань і часу, для найкращих результатів потрібні документи високої якості; зображення низької роздільної здатності можуть впливати на точність розпізнавання.

1.5.5 PenReader

PenReader – це система розпізнавання рукописного тексту, розроблена компанією Paragon Software Group. Вона призначена для інтерпретації тексту, написаного вручну, на сенсорних екранах пристроїв, таких як смартфони, планшети, ноутбуки та інші цифрові платформи. PenReader підтримує багатомовність і орієнтована на використання в мобільних пристроях та інтерактивних системах введення.

Система PenReader володіє наступними характеристиками:

а) розпізнавання рукописного тексту:

– відтворює текст, написаний вручну стилусом, пальцем або іншим засобом введення;

– працює як із друкованим, так і з рукописним стилем написання;

б) мультиплатформність:

– сумісний із мобільними ОС, такими як Android, iOS, а також Windows;

в) багатомовність:

– підтримує понад 40 мов, включаючи англійську, німецьку, українську та багато інших;

г) інтуїтивне навчання:

– завдяки машинному навчанню може адаптуватися до стилю письма кожного користувача;

д) контекстне розпізнавання:

– використовує контекст для покращення точності розпізнавання слів та фраз;

е) робота без інтернету:

– може працювати в офлайн-режимі, що робить його зручним для використання в будь-яких умовах;

ж) жести та спеціальні символи:

– підтримує введення спеціальних символів і дій за допомогою жестів;

з) підтримка інтеграції:

– може інтегруватися в додатки як засіб введення тексту.

Програма підтримує 4 режими розпізнавання рукописного тексту [12]:

– злите розпізнавання, яке дозволяє користувачеві писати слова, не відриваючи рук від аркушу;

– побуквене розпізнавання, яке дозволяє розпізнати тільки один символ за певний проміжок часу;

– інтелектуальне розпізнавання, яке дозволяє коригувати результати розпізнавання безпосередньо у процесі писання (наприклад, якщо вводити по черзі символи «/», «\», «-», то програма спочатку розпізнає введення символу «слеш», потім виправить на букву Л, а потім на А);

– роздільне розпізнавання – схоже на побуквене, але єдиним символом вважається кожен написаний штрих.

PenReader працює за наступним алгоритмом: спочатку користувач вводить текст, пишучи його на екрані стилусом чи пальцем, програма аналізує написані символи, для кожного розпізнаного символу система враховує можливі варіанти та співставляє їх зі словником, для підвищення точності використовується контекст фрази або речення, результат вводиться в текстове поле, або передається іншому додатку.

Система PenReader досить проста у використанні, підьримує багато мов, програма «вчиться» на прикладах введення тексту, підлаштовуючись до стилю письма, немає потреби в інтернеті для базового функціоналу, швидке розпізнавання тексту навіть у режимі реального часу.

До недоліків можна віднести залежність від якості та зрозумілості почерку користувача. Деякі складні жести або спеціальні символи можуть бути інтерпретовані неправильно.

2 РОЗРОБКА МОДЕЛІ ГШНМ В ЗАДАЧАХ РОЗПІЗНАВАННЯ ТЕКСТУ

2.1 Структура ШНМ та принцип роботи

Розглянувши вже існуючі програми розпізнавання рукописного тексту можна виділити головні проблеми у виконанні поставленої задачі, такі як залежність від якості вхідних даних, тобто наскільки чітко користувач написав літери, наскільки розбірливо, чи не зливаються літери, чи відслідковується загальний образ літер, за яким буде відбуватися розпізнавання та інше. Від цього також буде залежати і швидкість виконання поставленої задачі. Також є проблема вибору оптимальної архітектури штучної нейронної мережі і значні обчислювальні витрати під час навчання нейронної мережі, що дає зменшення кількості помилкових розпізнавань тексту. Отже, головним фактором та помічником вирішення проблем є вибір саме гібридних штучних нейронних мереж, що допоможе об’єднати всі позитивні якості декількох нейронних мереж і як результат підвищиться точність розпізнавання тексту.

В межах даної роботи розроблено додаток на основі гібридних штучних нейронних мереж для вирішення задач розпізнавання та класифікації рукописного тексту за зображеннями із використанням згорткових штучних нейронних мереж разом із багатошаровим персептроном.

Гібридні штучні нейронні мережі (ШНМ) об'єднують кілька підходів і архітектур для покращення ефективності розпізнавання зображень у складних умовах, зокрема за наявності шуму, часткових спотворень, низької якості зображення або інших перешкод.

Поєднання згорткової нейронної мережі (CNN) і багатошарового персептрона (MLP) – це популярний підхід у задачах, які потребують вилучення ознак із зображень та подальшої класифікації або регресії. Така

архітектура поєднує переваги CNN у роботі з просторовими даними та гнучкість MLP у побудові складних нелінійних залежностей. Також їхнє поєднання допомагає уникнути виникнення колізій як результат виконання задачі з розпізнавання.

Колізії – це ситуації, коли у двох і більше вихідних сигналів виникає однакове вихідне значення.

Існують конфлікти в оптимізації (gradient conflicts), тобто у багатовимірному просторі оптимізації різні параметри можуть мати суперечливі напрями градієнтів, що призводить до уповільнення або нестабільності процесу навчання.

Також є колізії функції активації, тобто насичення активацій. У деяких активаційних функціях (наприклад, сигмоїдальної чи гіперболічного тангенса), значення градієнтів можуть стати занадто малими, коли активація досягає насичення (plateau). Це заважає ефективному оновленню ваг і може спричинити «зникнення градієнтів».

Колізії в генеративних моделях, а саме взаємодія генератора і дискримінатора (GANs). У генеративно-змагальних мережах (GANs) можуть виникати ситуації, коли генератор і дискримінатор «конкурують», але один із них стає занадто сильним, спричиняючи колапс навчання.

Також є колізія даних, тобто перекриття кластерів у вхідних даних. Якщо різні класи в навчальному наборі даних мають схожі особливості, це може викликати труднощі в класифікації.

У великих мережах неправильна ініціалізація ваг може призвести до колізій між шарами, коли один шар стає домінуючим або створює вибухаючі/зникаючі градієнти.

У моделях із багатозадачним навчанням різні задачі можуть конкурувати за ресурси, що ускладнює досягнення високої точності для всіх задач одночасно.

У задачах комп'ютерного зору шуми чи артефакти можуть створювати «перешкоди» для моделі.

Використання методів програмного розпаралелювання алгоритму допоможе підвищити продуктивність роботи гібридної нейронної мережі. А отже, підмережі будуть виконувати одночасну обробку вхідних зображень тексту.

2.1.1 Гібридна штучна нейронна мережа

Гібридна штучна нейронна мережа – це модель, яка поєднує кілька типів архітектур або підходів у рамках однієї системи для досягнення кращої продуктивності. Гібридність дозволяє використовувати переваги різних методів, компенсуючи їхні недоліки, і забезпечує високу адаптивність до різноманітних задач.

Існує кінцевий набір навчальних пар $\{(X_i, Y_i)\}$, які задають відображення F множини X у множину Y: F: $(X \to Y)$, де матриця вхідних сигналів $X = [X_1, X_2, X_3, \dots X_k]$ – матриця вхідних сигналів, Y – вихід ГНМ.

Навчання гібридних штучних нейронних мереж (ГШНМ) є складним процесом, оскільки такі мережі поєднують різні типи архітектур і моделей для досягнення кращої продуктивності. Це вимагає правильного налаштування та оптимізації кожного компоненту мережі, а також налаштування взаємодії між різними підсистемами.

Головним принципом навчання є мінімізація функціоналу помилки, тобто підмережа, яка є складом гібридної нейронної мережі та виконує обчислення значення функції F(X_i) для вхідного вектора X_i, є експертом.

Сам процес навчання відбувається за принципом «навчання з учителем», а саме навчання на великій кількості прикладів, тобто на великій навчальній вибірці.

В межах даної роботи розглядається гібридна штучна нейронна мережа, в якій у якості експертів виступають згорткова штучна нейронна мережа та багатошаровий персептрон. Експерти попередньо навчені розпізнаванню кожної літери алфавіту.

Алгоритм роботи мережі полягає в послідовній обробці кожного символу із вхідного тексту. Кожен символ представлено у вигляді зображення розміром 32*32 пікселі. Зображення символу подається на вхід згорткової штучної нейронної мережі, а потім на вхід багатошарового персептрона. Тобто, це зображення надається експертам. Надалі на вихід надається масив, який містить тридцять три десяткових чисел. Числа масиву є ні що інше, як ймовірність відповідності вхідного символу кожній літері українського алфавіту. Мережа обирає елемент, який має найбільше значення ймовірності відповідності певній літері алфавіту та надає його на вихід. Кожній літері українського алфавіту заздалегідь присвоєно своє значення - нумерація, а отже номер вихідного елементу буде відповідати порядковому номеру відповідної літери.

Наступним етапом є порівняння результатів кожного з експертів. Якщо один з наданих елементів має більше значення, то він висувається як кінцевий результат символу, який було розпізнано.

Розглянемо схему глибинної нейронної мережі із двома експертами на рисунку 2.1.

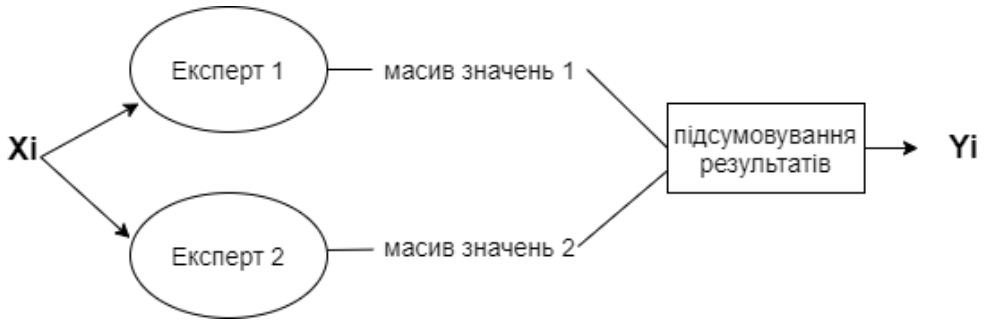

Рисунок 2.1 – Загальна схема глибинної нейронної мережі

На рисунку можемо бачити сам процес роботи такої мережі із двома експертами, на схемі X_i позначено вхідне значення у вигляді символу, Y_i позначає розпізнаний символ.

2.1.2 Експерти гібридної штучної нейронної мережі

Попереднє навчання експертів відбувається окремо один від одного та кожен за своїм алгоритмом навчання. Незабаром після навчання вже відбувається запуск безпосередньо гібридної штучної нейронної мережі.

2.1.2.1 Згорткова штучна нейронна мережа CNN

Згорткова штучна нейронна мережа – це тип нейронної мережі, що спеціально розроблений для ефективної обробки даних, які мають сітчасту структуру, наприклад, зображень, аудіо чи відео. CNN використовують згорткові шари для автоматичного вилучення просторових ознак, зберігаючи їх локальну структуру.

LeNet – одна з перших згорткових нейронних мереж (CNN), запропонована Яном ЛеКуном у 1989 році для задач розпізнавання рукописних символів і цифр, наприклад, у системах поштової автоматизації. Ця архітектура показала, як ефективно використовувати згорткові шари для вилучення ознак із вхідних зображень.

Архітектура складається з наступних складових (рисунок 2.2 [13]):

а) вхідний шар:

– приймає на вхід зображення розміром 32×3232 \times 3232×32 пікселів;

– якщо вхідне зображення має інший розмір, його необхідно масштабувати або змінювати розмір;

б) згорткові шари (Convolutional Layers):

– виконують згорткові операції для вилучення локальних ознак (контури, краї, текстури);

– у LeNet використовуються невеликі ядра згортки, зазвичай 5×55;

– активація здійснюється за допомогою функції tanh (у

сучасних моделях зазвичай використовується ReLU);

в) підвибірка (Pooling):

– застосовується субдискретизація (subsampling) або понижуюча вибірка (pooling), яка зменшує розмірність простору ознак;

– у LeNet використовується усереднююча підвибірка (Average Pooling), що на той час була популярною. Сьогодні частіше застосовується Max Pooling;

г) повнозв'язні шари (Fully Connected Layers):

– після кількох згорткових і пулінгових шарів вилучені ознаки передаються на вхід повнозв'язних шарів;

– ці шари виконують класифікацію на основі вилучених ознак;

д) вихідний шар:

– містить стільки нейронів, скільки є класів у задачі;

– застосовується функція активації softmax для отримання ймовірностей належності до кожного класу.

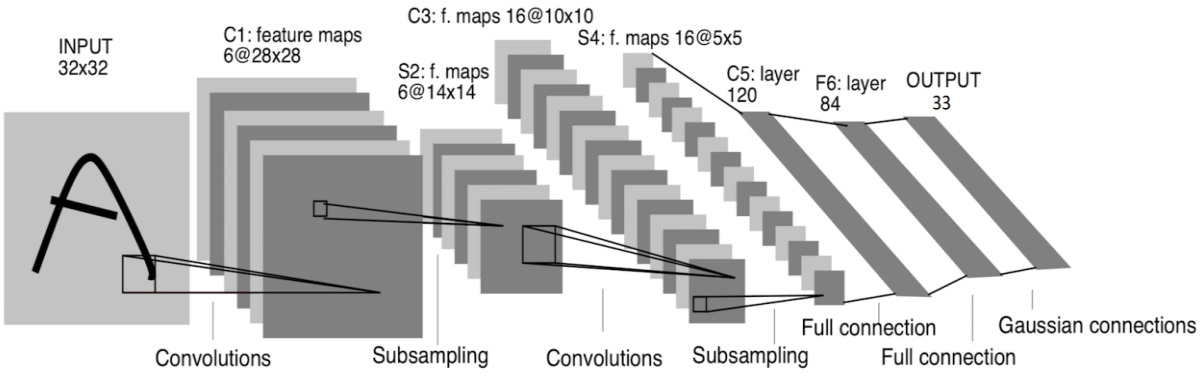

Рисунок 2.2 – Архітектура LeNet: повноз’єднана згорткова нейронна мережа

Головною перевагою такої архітектури є поєднання простоти з високою точністю розпізнавання образів. Завдяки використанню згорткових

шарів модель виявляє локальні ознаки, які є ключовими для розпізнавання об'єктів. LeNet стала фундаментом для багатьох сучасних згорткових мереж. Модель не є масштабованою для великих зображень чи складних задач, оскільки має порівняно малу кількість фільтрів і шарів.

Архітектура LeNet використовується для розпізнавання рукописного тексту, у банківських системах та системах автоматизації пошти.

Згортковий шар є ключовим компонентом згорткових нейронних мереж (Convolutional Neural Networks, CNN). Він використовується для автоматичного вилучення ознак із вхідних даних, зокрема зображень, відео або іншої багатовимірної інформації. Завдяки згортковим шарам, CNN можуть ефективно виявляти локальні патерни (наприклад, краї, текстури чи складні структури) у вхідних даних.

Згортковий шар працює за допомогою ядра згортки (фільтра), який проходить через вхідне зображення і виконує операцію згортки. Результат операції зберігається у вигляді нового масиву (тензора), що називається картою ознак (feature map).

Архітектура LeNet використовує шари згортки розміром 5*5. Фільтр рухається із зсувом в один піксель по всьому зображенню. Самі значення цього фільтра перемножуються із вихідними значенням кожних пікселів зображення, тобто виконується поелементне множення. Далі виконується підсумовування всіх попередніх результатів множення.

Унікальні позиції зображення, кожна з них, надає число, приклад розглянуто на рисунку 2.3 [13], тобто у разі використання шести фільтрів, то об'єм буде 28*28*6.

Пулінг – це шар у згорткових нейронних мережах (CNN), який зменшує розмірність вхідного тензора, зберігаючи важливу інформацію. Головна мета пулінгу – знизити обчислювальну складність мережі, зменшити кількість параметрів та зробити модель більш стійкою до шуму, спотворень і невеликих трансляцій вхідних даних [13].

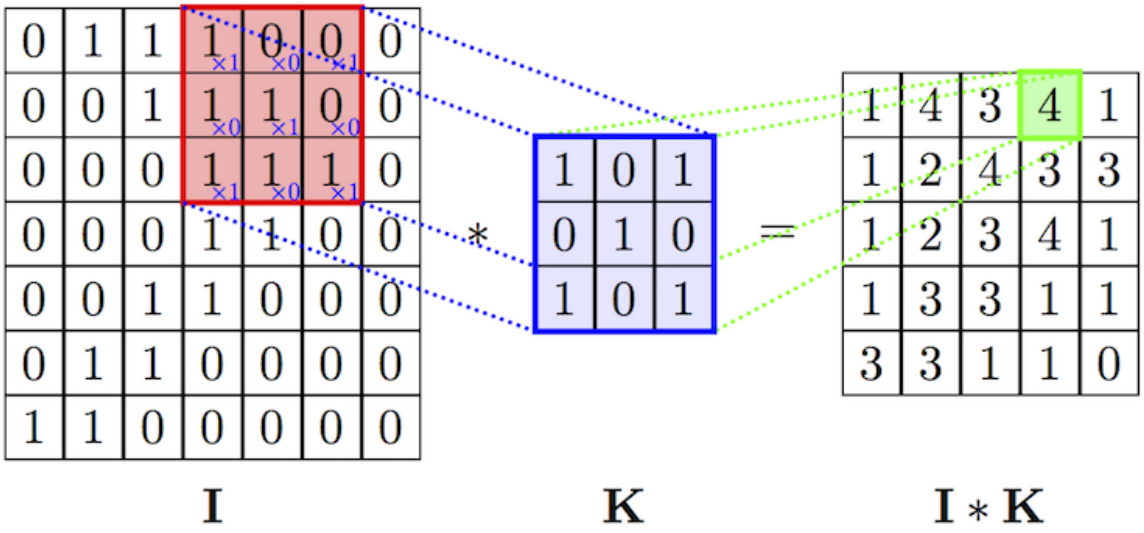

Рисунок 2.3 – Операція згортки та отримання значень

Пулінг працює шляхом виділення підобластей із вхідного тензора та обчислення одного значення для кожної з цих областей. Ця операція застосовується окремо до кожної карти ознак (feature map) [14]. На рисунку 2.4 можна побачити функцію максимуму Max Pooling.

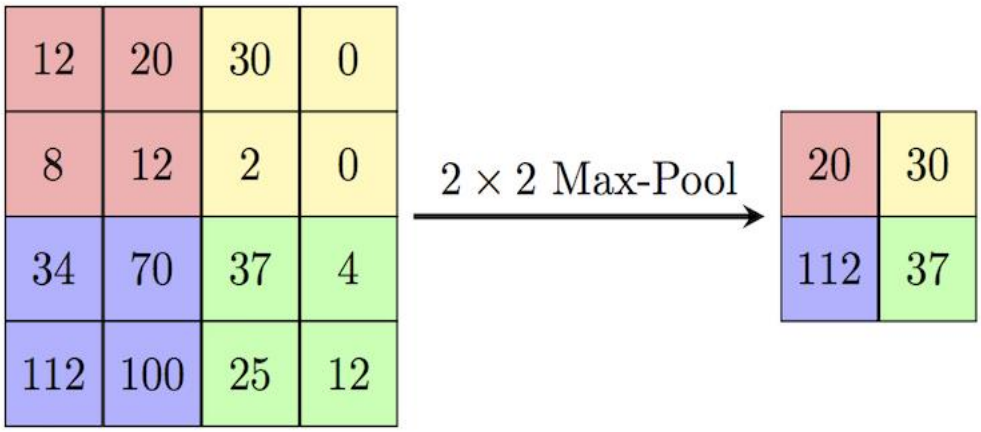

Рисунок 2.4 – Операція підвибірки (Max Pooling)

Перетворення охоплюють непересічні квадрати. З кожного такого квадрата вибирається максимальний елемент (нейрон). Потім цей нейрон приймається за елемент наступної, але вже зменшеною карти ознак.

З метою зменшення загального розміру зображення та збільшення його ступеню інваріантності фільтрів, які було застосовано по відношенню до нього, виконується саме ця дія.

Особливості пулінгу:

– зменшення розмірності: рменшує обчислювальні витрати та кількість параметрів;

– виділення ключових ознак: Max Pooling залишає найсильніші сигнали, ігноруючи слабкі або незначні;

– стійкість до шуму: допомагає моделі залишатися стійкою до невеликих змін у вхідних даних (трансляцій, спотворень);

– втрати інформації: зменшення розмірності призводить до часткової втрати даних. Для великих областей пулінгу це може стати проблемою.

Пулінг використовується для рівномірного зменшення розмірності, передавання важливих ознак, зменшує розмірність перед передаванням інформації до повнозв'язних шарів.

Пулінг працює за таким принципом, що при виявленні будь-яких ознак під час попередньої обробки, то надалі, при наступних обробках, вже немає необхідності в настільки детальному зображенні, тому виконується ущільнення зображення з детального до менш детального. Такі засоби фільтрації зайвих деталей допомагають уникнути перенавчання мережі в процесі навчання [14].

Персептрон – це нейронна мережа, яка є повноз’єднанною. Персептрон складається з декількох прихованих шарів. В таких прихованих шарах кожен їхній нейрон має зв’язок з кожним іншим нейроном (рисунок 2.5). Виконується переналаштування системи для більш абстрактних карток ознак внаслідок етапів згорток та ущільнення, використовуючи пулінг. Треба зауважити появу нових каналів на кожному наступному шарі та зменшення розмірності зображення в кожному каналі.

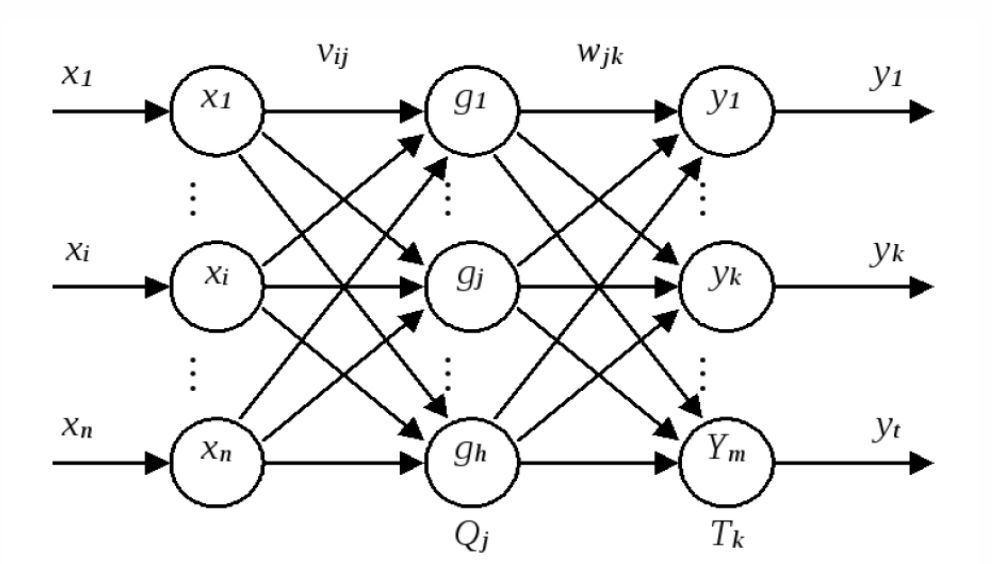

Рисунок 2.5 – Повноз'єднаний шар мережі

Як результат, ми маємо досить великий набір каналів, в яких зберігається достатня кількість даних невеликого об’єму щодо абстрактних понять, які виявлені із вихідного зображення. Дана інформація спочатку об’єднується, а потім передається багатошаровому персептрону, адже його вихід і є вихідним шаром нейронної мережі.

На вхід подається зображення у вигляді матриці розміром 32*32. Розглянемо, що впливає на розмір матриці:

– розмір карт ознак буде цілим числом в обох шарах;

– потрапляння нейрона шару згортки у центр поля.

Звідси ми отримуємо, що вхідний шар моделі LeNet містить 1024 нейронів. Надалі виконується операція згортки над вхідним шаром. Згортка складається з шести фільтрів, а отже як результат ми отримуємо шість карт ознак за розміром 28*28.

Наступним етапом виконується операція субдескритизації. В результаті виконання такої операції відбувається зменшення в два рази карти ознак.

Надалі виконується знову операція згортки, яка в цей раз вже складається з 16-ти фільтрів. На виході отримуємо результат із 16-ти карт

ознак, які мають розмір 10*10. Після виконання операції макспулінга відбувається зменшення розмірів карт в два рази, а отже їхній розмір становитиме 5*5. Після виконання всіх операцій, описаних вище, виконання нових операцій згортки та ущільнення більше не відбувається.

За допомогою збільшення кількості шарів, операція згортки виконує розширення рецептивних полів. Треба зауважити, що операція згортки не виконується глобально відносно всього зображення, а лише виконує операції над частиною зображення певного розміру.

Враховуючи властивості повноз’єднаних шарів, а саме, що всі вузли шару поєднані зі всіма вузлами наступного шару, то така властивість дозволяє виконати визначення глобального взаємозв'язку між характеристиками.

В межах даної роботи використано два повноз’єднаних шари, які до того ж нелінійні. Такий підхід допомагає у визначення ступеню взаємоджії та залежностей на рівні ознак.

Далі, наступним йде повноз’єднаний шар, який містить два приховані шари. Перший з цих шарів складається зі 120 нейронів та він на вході приймає 400 нейронів розмірами 16*5*5. Другий з цих шарів на вході приймає виходи від попереднього шару та він складається вже з 84 нейронів. Обидва ці шари використовують сигмоїдальну функцію активації. Ця функція обчислюється за наступною формулою 2.1.

$$OUT(NET) = \frac{1}{1+e^{-NET}}, \tag{2.1}$$

де OUT – вихід нейрона;

NET – вихід підсумовувального блока.

Головною перевагою сигмоїдальної функції є монотонність та диференційність протягом всього діапазону значень аргументу, наведено на рисунку 2.6. Така її властивість допомагає використовувати функцію в процесі створення нейронних мереж.

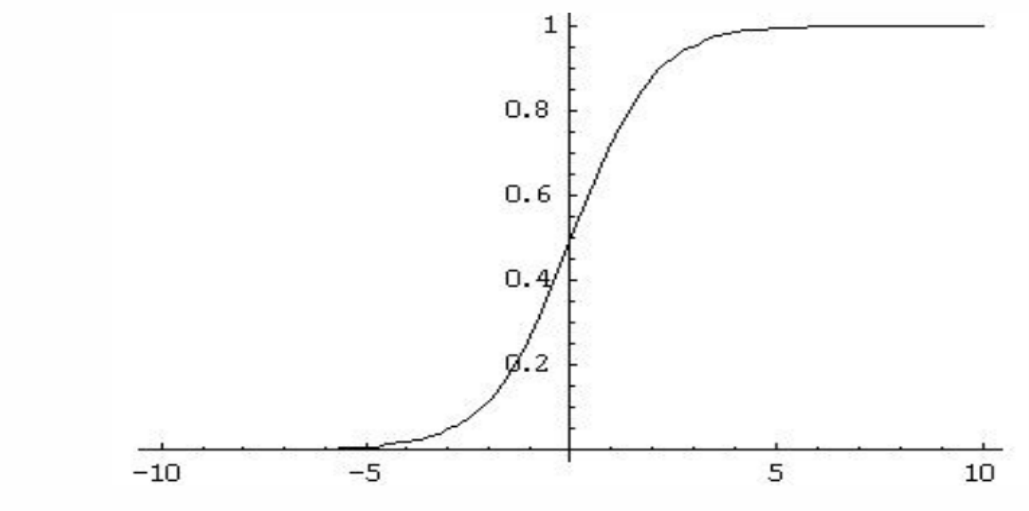

Рисунок 2.6 – Сигмоїдальна функція активації

В свою чергу логічна функція активації Softmax використовується у вихідному шарі. Робота такої функції полягає у перетворенні вектору в числа в діапазоні від 0 до 1, а отже їхня сума буде дорівнювати одиниці.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \tag{2.2}$$

де σ – вихідний вектор;

z – вхідний вектор;

i – номер нейрона, який обробляється;

k– кількість вихідних нейронів.

В CNN вихідний шар має всього 33 виходи, кожен з яких вказує на вирогідність співпадання із конкретною літерою з алфавіту на зображенні, яке подавалося на вхід нейронної мережі.

2.1.2.2 Багатошаровий Персептрон

Багатошаровий персептрон (MLP) – це вид штучної нейронної мережі, що складається з кількох шарів нейронів. На відміну від одношарового

персептрона, який може розв’язувати лише лінійно роздільні задачі, MLP здатний обробляти складні нелінійні залежності завдяки використанню прихованих шарів і нелінійних функцій активації (рисунок 2.7).

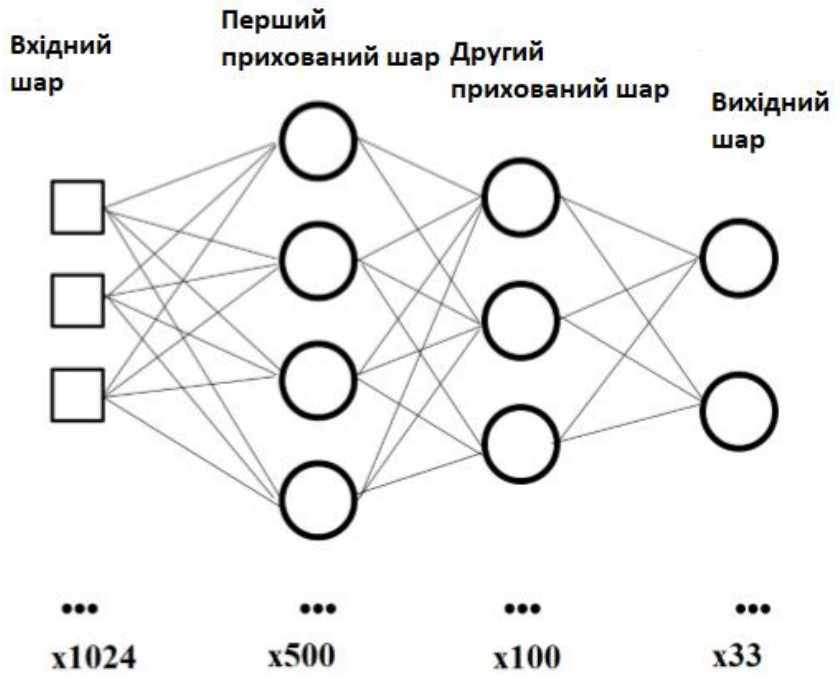

Рисунок 2.7 – Multi-Layer Perceptron

У спільній роботі із CNN як результат ми отримуємо більш точні результати, а також збільшення швидкості навчання на відміну від ситуацій, коли використано занадто велику кількість шарів.

Багатошаровий персептрон складається з таких складових:

– вхідний шар, який складається з 1024 нейронів;

– перший прихований шар, який складається з 500 нейронів;

– другий прихований шар, який складається з 100 нейронів;

– вихідний шар, який складається з 33 нейронів, що являють собою значення ймовірностей приналежності вхідного символу до кожної букви.

Розглянемо структуру нейрону у прихованному шарі (рисунок 2.8).

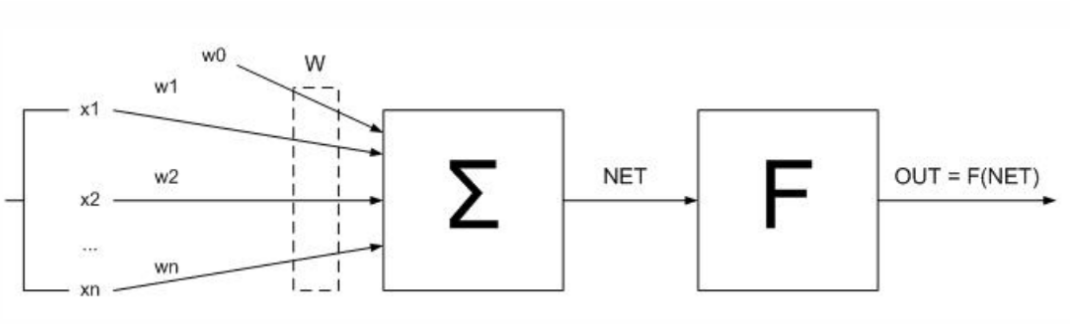

Рисунок 2.8 – Нейрон прихованого шару

Для нейронів прихованого шару зазвичай використовується функція активації Leaky ReLU. На рисунку 2.9 відображено графік функції активації Leaky ReLU.

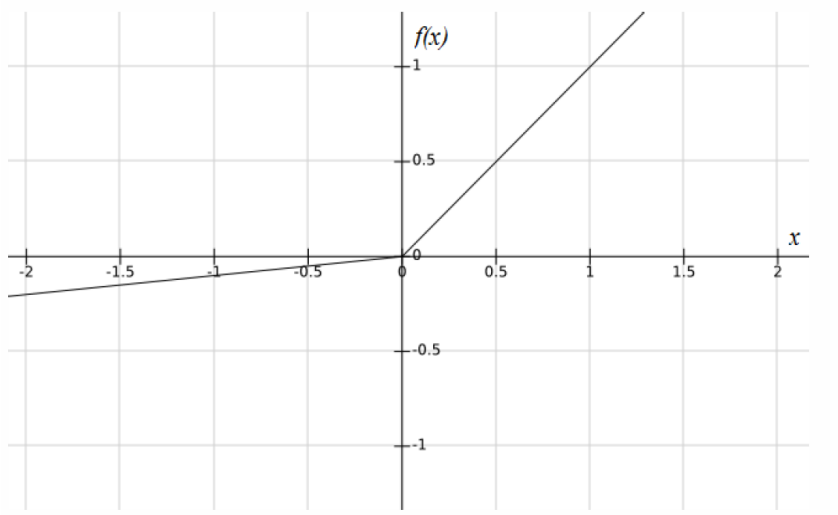

Рисунок 2.9 – Функція активації Leaky ReLU

Наступними розглянемо нейрони вихідного шару. Для них використовується функція активації Softmax. Щодо їхньої структури, то вона однакова, як і в нейронах прихованого шару.

Метод зворотного поширення помилки (Backpropagation) – це алгоритм навчання багатошарових нейронних мереж, який дозволяє оновлювати ваги нейронів на основі похибки, отриманої на виході.

Цей метод використовується разом із градієнтним спуском для мінімізації функції втрат, дозволяючи мережі поступово покращувати свої передбачення.

Метод зворотного поширення помилки (Backpropagation) використовується для навчання мережі. Головний принцип роботи цього методу полягає у поширенні сигналів помилки у зворотному напрямку, а саме від виходів мережі до входів мережі.

Метод зворотного поширення помилки працює за наступним алгоритмом дій:

а) форвардне поширення (Forward Propagation):

– вхідні дані проходять через мережу, кожен нейрон обчислює лінійну комбінацію своїх входів та застосовує функцію активації;

– у результаті отримується вихідний прогноз мережі;

б) обчислення похибки (Loss Calculation):

– обчислюється функція втрат L, яка показує, наскільки прогноз відрізняється від реального значення;

– наприклад, для задачі регресії можна використовувати середньоквадратичну помилку (MSE):

$$L = \frac{1}{2}(y_{\text{реальне}} - y_{\text{прогноз}})^2; \tag{2.3}$$

– для класифікації часто використовується крос-ентропія;

в) зворотне поширення градієнтів (Backpropagation of Gradients):

– обчислюється похідна функції втрат відносно кожного параметра мережі;

– використовується правило ланцюга (chain rule) для розрахунку внеску кожного шару у загальну похибку;

г) оновлення ваг (Weight Update) за допомогою градієнтного спуску:

– ваги оновлюються за правилом:

$$w := w - \eta \frac{\partial L}{\partial w}, \tag{2.4}$$

де η – швидкість навчання (learning rate);

– це дозволяє мережі поступово зменшувати помилку та покращувати свої передбачення.

Метод зворотного поширення помилки є ключовим алгоритмом навчання нейронних мереж, що дозволяє їм адаптуватися до вхідних даних. Завдяки цьому методу можна ефективно навчати навіть глибокі нейронні мережі, що складаються з багатьох шарів.

Процес навчання мережі полягає в обчисленні вектору помилки у вигляді різниці між отриманими даними та тими, які вже відомі, тобто дані з досвіду. Вектор помилки в свою чергу використовується як засіб модифікації вагових коефіцієнтів вихідного шару. Це зроблено з метою зменшення вектору помилки у разі повторного подання одного й того ж набору вхідних даних [15].

Зазделегідь до навчання необхідно обрати значення швидкості навчання, тобто параметр градієнтних алгоритмів навчання нейронних мереж, який допомагає керувати величиною корекції ваг на кожній ітерації.

Використовуючи формулу 2.5 ми можемо розрахувати модифікацію значення ваги між вихідним шаром та прихованим.

$$New\_weight = \alpha * old\_weight + \mu * der_k * err * v, \tag{2.5}$$

де α – момент інерції;

lasweight – вага у момент попереднього навчання;

μ – швидкість навчання;

der_k – похідна від значення нейрону вихідної мережі;

err – помилка;

v – значення нейрону прихованої мережі.

Наступним кроком виконується так само зміна вагових коефіцієнтів

прихованого шару. Та далі ми розраховуємо нове значення ваг між вхідним та прихованим шарами за формулою 2.6.

$$New\_weight = \alpha * old\_weight + \mu * der_j * err * x, \quad (2.6)$$

де α – момент інерції;

lasweight – вага у момент попереднього навчання;

μ – швидкість навчання;

der_j – похідна від значення нейрону прихованої мережі;

err – помилка;

x – значення нейрону вхідної мережі.

Епоха (Epoch) – це один повний прохід через весь навчальний набір даних під час тренування нейронної мережі. В даній роботі було виконано 1000 епох. Така кількість епох повністю відповідає кількості зображень кожної літери алфавіту.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Середовище розробки

Головною задачею даної роботи є розпізнавання рукописних літер українського алфавіту, тож доцльно буде використовувати Microsoft Visual Studio та .NET Framework 4.7.

Доцільність вибору саме таких засобів пояснюється тим, що Microsoft Visual Studio надає можливості створення різних проектів на різних мовах, таких як Visual C++ та Visual C#. А також є можливість застосовувати .NET для різних платформ [16].

В даній роботі було використано мову програмування C#. C# – це об'єктно-орієнтована, багатоцільова мова програмування, розроблена компанією Microsoft у 2000 році як частина платформи .NET. Вона поєднує в собі простоту та ефективність C++, розширені можливості Java та зручність Python.

C# підтримує класи, успадкування, поліморфізм, інкапсуляцію та інші концепції ООП. Мова має суворий контроль типів даних, що зменшує кількість помилок під час виконання. C# використовує збирання сміття (GC), що допомагає уникнути витоків пам’яті. C# широко використовується для розробки настільних програм (Windows, macOS, Linux), веб-додатків (ASP.NET), мобільних застосунків (Xamarin), ігор (Unity) та багато іншого. Завдяки .NET Core і .NET 5+, C# дозволяє створювати додатки для Windows, macOS і Linux [17].

C# використовується для виконання наступних задач:

- розробка Windows-додатків (WPF, WinForms, UWP);
- веб-розробка (ASP.NET, Blazor);
- мобільна розробка (Xamarin);
- ігрова індустрія (Unity);
- системне програмування;

– штучний інтелект і машинне навчання (ML.NET).

C# – це потужна, гнучка і сучасна мова програмування, яка широко застосовується в розробці програмного забезпечення. Завдяки підтримці платформи .NET, автоматичному керуванню пам'яттю та об'єктно-орієнтованому підходу, C# є чудовим вибором для багатьох проєктів. Також мова програмування C# надає можливість застосування різноманітних бібліотек, а також класів.

3.2 Робота гібридної нейронної мережі

В роботі гібридної нейронної мережі першим етапом є подача на вхід вхідних даних, а саме безпосередньо зображення рукописного тексту. Зображення може бути як фотографією, рисунком або ж сканом документа.

Головною проблему для розпізнавання наданого зображення є погана якість, маленький розмір, недоліки самого паперу, темний бекграунд або ж його нерівномірність, закреслений текст, тощо. Отже, надане зображення спершу повинно пройти його оброку.

Обробка зображення складається з наступних етапів:

– виконується нормалізація наданого зображення в розмір 32x32 пікселя;

– ліквідація перешкод і затемнень;

– перетворення наданого зображення в чорно-білий формат.

Сам процес обробки наданого зображення полягає спершу в процесі розділення зображення на певні ділянки, якими є пікселі. Наступний крок, виконується визначення кольору кожного пікселя зображення. При визначеному кольорі «чорний» буде присвоєно значення 1, при визначеному кольорі «білий» буде присвоєно значення 0. Очевидно, що білий колір то буде фон зображення, а чорний – сама літера. Як результат обробки, ми отримуємо зображення розміром 32x32 із відображеною літерою.

Структура самої програми складається з чотирьох частин. Кожна така частина називається модулем. Перший буде модуль, який безпосередньо відповідає за обробку вхідного зображення та називаєтсья такий модуль Image Processing. Другий модуль полягає в самому процесі реалізації CNN, тобто згорткової нейронної мережі. Третій модуль полягає в процессі створення Multilayer Perceptron, тобто багатошарового персептрону. І останній, четвертий, модуль виконує створення мережі саме гібридної (Hybridization), виконує введення даних та виведення оброблених нею результатів (рисунок 3.1).

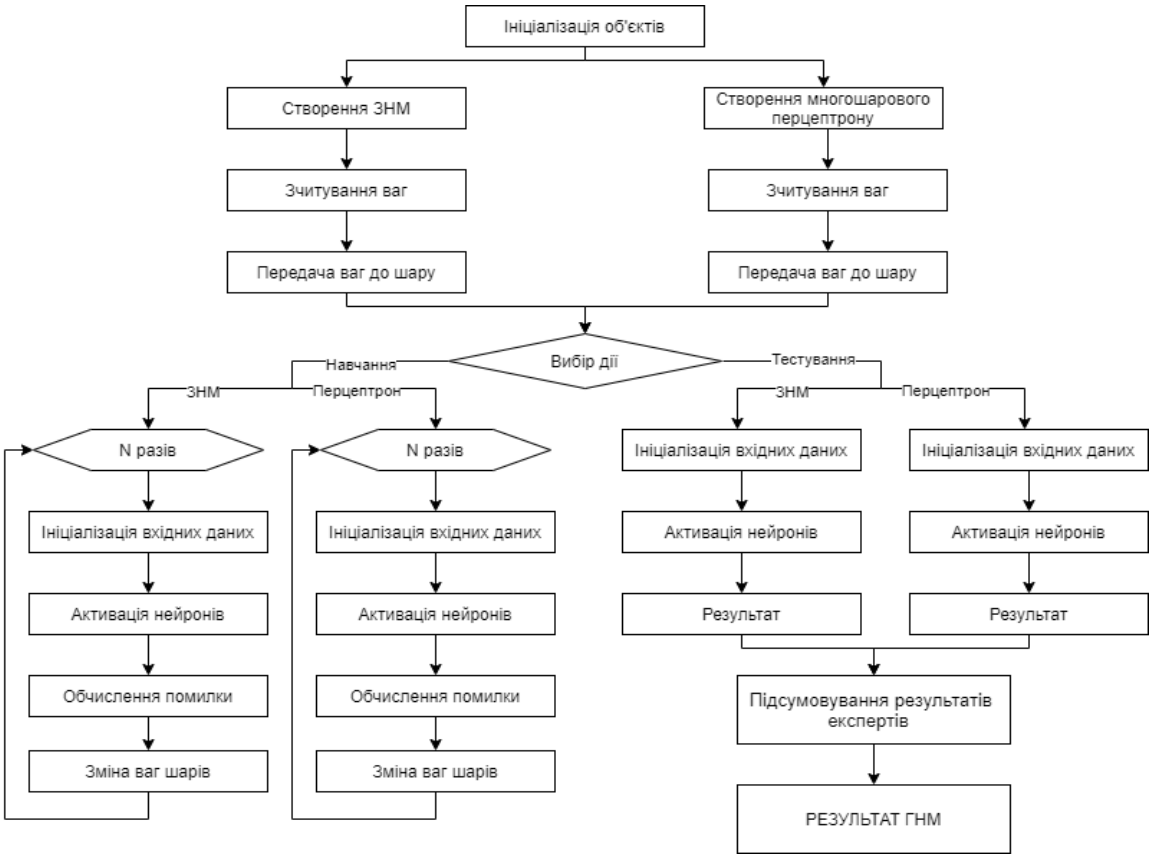

Рисунок 3.1 – Схема програми

Під час розробри програми було створено класи, тож розглянемо основні з них:

– class ConvolutionFirst – це клас, який відповідає за перший шар

згортки;

– class FirstPooling – це клас, який відповідає за виконання першої операції субдискретизації (пулінгу);

– class SecondConvolution – це клас, який відповідає за другий шар згортки;

– class SecondPooling – це клас, який відповідає за виконання другої операції субдискретизації (пулінгу);

– class InputLayer – це клас, який відповідає за вхідний шар;

– class HiddenLayer – це клас, який відповідає за прихований шар;

– class OutputLayer – це клас, який відповідає за вихідний шар;

– class NeuronActivation – це клас, який відповідає за активацію нейрону;

– class Neuron – це клас, який відповідає за ініціалізацію нейрону.

З метою тестування розробленої в межах даної роботи системи, було використано набір даних, який складається з варіацій зображень кожної літери алфавіту. На кожну літеру приходиться 1200 зображень з різними шрифтами написання даної літери. Для створення набору зображень для тестування системи було використано сервіс, який містить зображення літер українського алфавіту.

В свою чергу для навчання нейронної мережі було застосовано 1000 зображень, приклад яких можна побачити на рисунку 3.2, а також було використано 100 зображень для тестування мережі після навчання, приклад цих зображень можна побачити на рисунку 3.3. Зображення літер використовувалися друкованих, рукописних, рядкових та заглавних.

Рисунок 3.2 – Приклад зображень літери А, обраної для навчання мережі

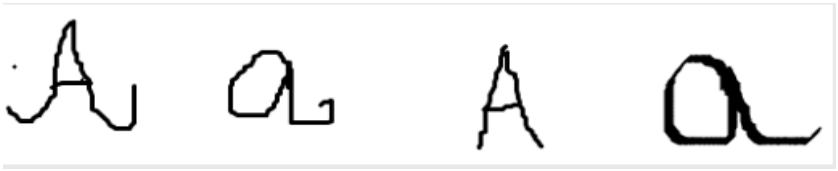

Рисунок 3.3 – Приклад зображень літери А, обраної для тестування мережі після навчання

Під час тестування вже навченої мережі, використовуються зображення рукописних літер, які були написані реальною людиною. Як результат тестування, ми отримаємо результати виконання розпізнавання експертом 1 та експертом 2, а також загальний отриманий результат гібридною нейронною мережею.

Під час виконання розпізнавання літери «а» системою було відзначено, що система має не значну різницю в коефіцієнтах збігу для літери «а», як і для літери «о», (рисунок 3.4).

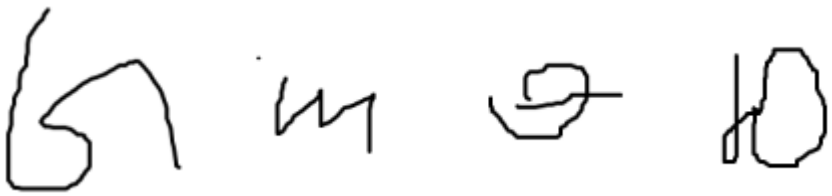

Рисунок. 3.4 – Літери, розпізнавання яких складає певні труднощі

Тобто, можна зробити висновок, що хоч система і правильно впоралася із поставленою задаченю, але для неї небагато різниці між цими двома літерами через їхню схожість, а отже при певних умовах, а саме при нерозбірковості в написанні між «а» та «о» може виникнути помилка в результаті, адже люди всі пишуть по різному і не всі пишуть розбірливо та чітко окреслюючи кожну літеру.

3.3 Порівняння роботи ГНМ із роботою окремих її складових

Розглянемо переваги використання гібридних нейронних мереж для виконання даної задачі. Якщо розглянути результати виконання однієї й тієї самої задачі (розпізнавання рукописних літер) ГНМ та окремих її складових – експертів 1 та 2, то можна побачити досить значний відрив у якості виконання задачі.

На рисунку 3.5 можна побачити безпосередньо саме порівняння результатів розпізнавання символів та якість проведеного розпізнавання, використовуючи однакові вхідні дані, а саме набор із символів самого алфавіту, 100 символів та 1000 символів.

	33	**100**	**1000**
Гібридна ШНМ	30	94	971
Згорткова ШНМ	24	88	896
Багатошаровий персептрон	29	84	899

Рисунок 3.5 – Результат виконання задачі різними мережами

Можна зазначити, що результативність розпізнавання гібридної нейронної мережі достатньо перевищує результативність виконання тієї ж задачі іншими мережами. У відсотковому співідношенні результативність розпізнавання зображень засобами ГНМ складають приблизно 75%.

ВИСНОВКИ

Ця робота була спрямована на перевірку гіпотези про те, що використання гібридної нейронної мережі (ГНМ) у задачі розпізнавання рукописного тексту дозволяє підвищити точність передбачень.

Для цього було виконано такі кроки:

– проаналізовано існуючі моделі штучних нейронних мереж та інструменти, що застосовуються для розпізнавання рукописного тексту;

– розроблено концепцію та архітектуру гібридної нейронної мережі;

– реалізовано цю модель у вигляді програмного рішення;

– проведено тестування та порівняння запропонованої ГНМ із традиційними моделями, щоб оцінити її ефективність;

– підтверджено, що гібридний підхід дозволяє знизити кількість помилкових розпізнавань, завдяки чому загальна точність класифікації зростає.

У розробленій моделі поєднано два типи нейронних мереж: згорткову нейронну мережу (CNN) та багатошаровий персептрон (MLP). Остаточний результат формується шляхом об'єднання вихідних даних від обох мереж. Тестування показало, що такий підхід дає кращі результати порівняно з використанням лише однієї згорткової мережі, оскільки він частково усуває проблему колізій під час розпізнавання.

Окрім цього, дослідження продемонструвало, що якість роботи моделі значною мірою залежить від кількості навчальних даних. У перспективі можна розширити дослідження на складніші завдання, наприклад, розпізнавання слів, що пишуться разом, або аналіз почерку в більш складних умовах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Приймаченко Ю.В., Поліїт М.Р. Латентно-семантичний аналіз (LSA): визначення схожості між документами на основі смислового контексту. *Science in the modern world: innovations and challenges*. 2025. Т. 7. С. 190–195.

2. Чому у розпізнаванні рукописного тексту важливий контекст. *IDR Intelligent Document Recognition*. URL: https://idr.com.ua/article/08_2013/3.html (дата звернення: 07.02.2025).

2. Scherer D., Müller A., Behnke S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. *Artificial Neural Networks – ICANN 2010*. Berlin, Heidelberg, 2010. P. 92–101. URL: https://doi.org/10.1007/978-3-642-15825-4_10 (date of access: 07.02.2025).

3. ImageNet Large Scale Visual Recognition Challenge / O. Russakovsky et al. *International Journal of Computer Vision*. 2015. Vol. 115, no. 3. P. 211–252. URL: https://doi.org/10.1007/s11263-015-0816-y (date of access: 07.02.2025).

4. Zeiler M. D., Fergus R. Visualizing and Understanding Convolutional Networks. *Computer Vision – ECCV 2014*. Cham, 2014. P. 818–833. URL: https://doi.org/10.1007/978-3-319-10590-1_53 (date of access: 07.02.2025).

5. Learning Convolutional Feature Hierarchies for Visual Recognition. *List of Proceedings*. URL: https://papers.nips.cc/paper_files/paper/2010/hash/a01610228fe998f515a72dd730294d87-Abstract.html (date of access: 07.02.2025).

6. Image processing of human corneal endothelium based on a learning network / W. Zhang et al. *Applied Optics*. 1991. Vol. 30, no. 29. P. 4211. URL: https://doi.org/10.1364/ao.30.004211 (date of access: 07.02.2025).

7. М. Тим Джонс. Програмування штучного інтелекту в додатках.

2-ге вид. Print2print, 2017. 312 с.

8. Субботін С. О. Нейронні мережі: теорія та практика : навч. посіб. Житомир, 2020. 184 с.

9. Новотарський, М. А., Нестеренко Б. Б. «Штучні нейронні мережі: обчислення.» *Праці Інституту математики НАН України. Київ: Ін-т математики НАН України* 2004. 408 с.

10. Кононюк А. Нейроні мережі і генетичні алгоритми. *Головна сторінка DSpace*. URL: http://ir.nmu.org.ua/handle/123456789/146635 (дата звернення: 07.02.2025).

11. Добровська Л. М., Добровська І. А. Теорія та практика нейронних мереж. Київ : НТУУ «КПІ», 2015. 396 с.

12. Bengio Y., Courville A., Goodfellow I. Deep Learning. MIT Press, 2016. 800 p.

13. Bishop C. M., Bishop H. Deep Learning. Cham : Springer International Publishing, 2024. URL: https://doi.org/10.1007/978-3-031-45468-4 (date of access: 07.02.2025).

14. Hastie T. Handwritten digit recognition via deformable prototypes. Toronto : University of Toronto, Dept. of Statistics, 1992. 13 p.

15. Jianguo Wang, Hong Yan. A hybrid method for unconstrained handwritten numeral recognition. URL: https://doi.org/10.1016/S0167-8655(00)00029-5 (date of access: 07.02.2025).

16. Haykin S. Neural Networks and Learning Machines. Pearson Education, Limited, 2009. 937 p.

17. Murphy K. P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012. 1104 p.

18. Braspenning P. J., Thuijsman F. Artificial Neural Networks: An Introduction to Ann Theory and Practice. Springer, 1995. 293 p.

19. Hybrid Intelligent Systems / ed. by A. Abraham et al. Cham : Springer International Publishing, 2018. URL: https://doi.org/10.1007/978-3-319-76351-4 (date of access: 07.02.2025).

20. Fukushima K. Neocognitron for handwritten digit recognition. *Neurocomputing*. 2003. Vol. 51. P. 161–180. URL: https://doi.org/10.1016/s0925-2312(02)00614-8 (date of access: 07.02.2025).