

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
МАРІУПОЛЬСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ЕКОНОМІКО-ПРАВОВИЙ ФАКУЛЬТЕТ  
КАФЕДРА СИСТЕМНОГО АНАЛІЗУ ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ**

**До захисту допустити:  
В.о. зав. кафедри**

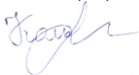


**Ганна МАРТИНЮК**

квітня 2024 р.

**СТВОРЕННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ  
ЗОБРАЖЕНЬ»**

Кваліфікаційна робота  
здобувача вищої освіти першого  
(бакалаврського) рівня  
освітньо-професійної програми  
«Комп'ютерні науки»  
Шихова Єгора Дмитровича  
Науковий керівник:  
Мнацаканян Марія Сергіївна,  
кандидат технічних наук, доцент,  
доцент кафедри системного аналізу та  
інформаційних технологій  
Рецензент:  
Лукашенко Вікторія Вікторівна,  
кандидат технічних наук, доцент,  
заступник декана факультету  
комп'ютерних наук та технологій  
Національного авіаційного університету

Кваліфікаційна робота захищена  
з оцінкою добре 82 (В)  
Секретар ЕК 

» червня 2024 р.

Київ - 2024

## АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи на тему «Створення нейронної мережі для розпізнавання зображень» складається з чотирьох розділів: теоретичного, аналітичного, дослідницького та практичного.

У першому розділі аналізуються основні теоретичні аспекти нейронних мереж, зокрема їх історія розвитку, різновиди та архітектури. Особлива увага приділяється згортковим нейронним мережам (CNN), які є найефективнішими для розпізнавання зображень. Розглядаються основні принципи роботи нейронних мереж, їхні ключові елементи та механізми навчання, включаючи алгоритм зворотного поширення помилки.

У другому розділі проводиться аналіз існуючих рішень у галузі розпізнавання зображень за допомогою нейронних мереж. Здійснюється порівняльний аналіз моделей AlexNet, VGG та ResNet, визначаються їхні переваги та недоліки. Окремо розглядаються практичні аспекти реалізації нейронних мереж, використання мови програмування Python та бібліотек TensorFlow і Keras для їх розробки.

У третьому розділі виконується дослідницька робота з розробки та навчання нейронної мережі для розпізнавання зображень. Описується процес збирання та підготовки даних для тренування моделей, а також детально розглядаються архітектури моделей, що використовуються. Наводяться результати експериментів з тренування та тестування моделей на різних наборах даних, таких як MNIST та ImageNet.

У четвертому розділі представлено практичне застосування розробленої нейронної мережі для розпізнавання зображень у різних сферах. Розглядаються можливості її використання в системах безпеки, виробництві, медицині та транспорті. Аналізується вплив впровадження нейронних мереж на ефективність та точність процесів розпізнавання, а також пропонуються рекомендації щодо подальшого розвитку та вдосконалення моделей.

## SUMMARY

The explanatory note for the qualification work on the topic "Creating a Neural Network for Image Recognition" consists of four chapters: theoretical, analytical, research, and practical.

In the first chapter, the main theoretical aspects of neural networks are analyzed, including their history of development, varieties, and architectures. Particular attention is paid to convolutional neural networks (CNNs), which are the most effective for image recognition. The basic principles of neural network operation, their key elements, and training mechanisms, including the backpropagation algorithm, are discussed.

The second chapter analyzes existing solutions in the field of image recognition using neural networks. A comparative analysis of the AlexNet, VGG, and ResNet models is conducted, highlighting their advantages and disadvantages. The practical aspects of implementing neural networks, including the use of the Python programming language and the TensorFlow and Keras libraries for their development, are also considered.

In the third chapter, research work on the development and training of a neural network for image recognition is performed. The process of collecting and preparing data for model training is described, and the architectures of the models used are examined in detail. The results of experiments on training and testing the models on various datasets, such as MNIST and ImageNet, are presented.

The fourth chapter presents the practical application of the developed neural network for image recognition in various fields. The potential uses of the network in security systems, manufacturing, healthcare, and transportation are discussed. The impact of implementing neural networks on the efficiency and accuracy of recognition processes is analyzed, and recommendations for further development and improvement of the models are provided.

## ЗМІСТ

ВСТУП .....	1
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ .....	3
1.1 Основи нейронних мереж .....	3
1.1.1 Визначення та історія .....	3
1.1.2 Типи нейронних мереж .....	4
1.1.3 Перцептрони та багатошарові перцептрони .....	7
1.2 Згорткові нейронні мережі (CNN) .....	9
1.2.1 Архітектура та компоненти .....	9
1.2.2 Принципи роботи та навчання .....	11
1.2.3 Застосування у розпізнаванні зображень .....	11
Висновки до розділу 1 .....	12
РОЗДІЛ 2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ .....	15
2.1 Відомі моделі .....	15
2.1.1 AlexNet: архітектура та вплив .....	15
2.1.2 VGG: архітектура та варіанти .....	15
2.1.3 ResNet: залишкові блоки та переваги .....	18
2.2 Порівняння моделей .....	19
Висновки до розділу 2 .....	20
РОЗДІЛ 3 ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ .....	22
3.1 Програмне забезпечення та бібліотеки .....	22
3.1.1 Python для нейронних мереж .....	22
3.1.2 TensorFlow: особливості та переваги .....	23
3.1.3 Keras: спрощення реалізації .....	25
3.2 Набори даних .....	26
3.2.1 MNIST: структура та використання .....	26
3.2.2 ImageNet: огляд та вплив .....	27
3.2.3 Інші набори даних та їхні особливості .....	28
Висновки до розділу 3 .....	31
РОЗДІЛ 4 РЕАЛІЗАЦІЯ НЕЙРОННОЇ МЕРЕЖІ .....	34
4.1 Підготовка середовища .....	34
4.2 Завантаження та підготовка даних .....	35
4.3 Підготовка середовища .....	36
4.4 Візуалізація даних .....	37
4.5 Побудова нейронної мережі .....	39
4.6 Тренування та оцінка моделі .....	41
4.7 Аналіз помилок та інші метрики .....	43
Висновки до розділу 4 .....	46
ВИСНОВКИ .....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	51

## ПЕРЕЛІК СКОРОЧЕНЬ

CNN	Згорткова нейронна мережа
RNN	Рекурентна нейронна мережа
MLP	Багатошаровий перцептрон
ReLU	Виправлений лінійний блок (функція активації)
SGD	Стохастичний градієнтний спуск
Adam	Адаптивне оцінювання моментів
RMSprop	Пропагація середньоквадратичного кореня
VGG	Мережа Візуальної Геометрії Групи
ResNet	Залишкова мережа
ILSVRC	Великий виклик візуального розпізнавання ImageNet
F1-Score	Гармонійне середнє значення точності та повноти

## ВСТУП

Технологічний прогрес невпинно рухається вперед, і ми все більше покладаємося на інформаційні системи у повсякденному житті, роботі, комунікаціях та виробництві. Інформаційні технології стали фундаментом сучасного суспільства, де автоматизація є невід'ємною складовою. Програми, веб-сайти, соціальні мережі та інші інструменти значно спрощують наші щоденні завдання та допомагають у вирішенні різних проблем.

Однією з основних переваг інформаційних технологій є можливість оптимізації та делегування процесів за допомогою зручних інтерфейсів та алгоритмів автоматизації. Сучасні програми не лише оптимізують процеси доставки та вибору товарів, але й допомагають приймати рішення на основі попередніх покупок. Камери спостереження здатні ідентифікувати осіб, а програми для обробки тексту виправляють помилки і навіть генерують унікальні тексти на задану тему. Ці можливості реалізуються завдяки технологіям машинного навчання та нейронних мереж.

У сучасному світі швидкого розвитку технологій нейронні мережі стали ключовим інструментом для вирішення різноманітних завдань. Вони широко використовуються в таких сферах, як економіка, бізнес, медицина, авіоніка, автоматизація виробництва, системи безпеки та охорони. Нейронні мережі можуть виконувати завдання з розпізнавання, обробки, трансформації даних та прогнозування змін на основі наявних даних.

Одне з найпоширеніших застосувань нейронних мереж — розпізнавання зображень та машинний зір. Нейромережі можуть швидко обробляти великі обсяги даних і виявляти задані об'єкти. Це відкриває широкий спектр можливостей, від відстеження порушень за камерами спостереження в реальному часі до аналізу якості виробництва. Важливою особливістю таких систем є їхня здатність до самоорганізації та навчання, що забезпечує високу точність розпізнавання.

Метою цього дослідження є аналіз і розробка систем, здатних швидко та точно визначати об'єкти на зображеннях. Тема є актуальною через великий попит на автоматизоване розпізнавання об'єктів, яке є трудомістким для людини, але може бути ефективно виконане автоматизованою системою.

Актуальність дослідження обумовлена постійною потребою в модернізації методів швидкого розпізнавання образів для автоматизації процесів верифікації.

#### Об'єкт та Предмет Дослідження

- Об'єкт дослідження: нейронні мережі як технологія реалізації систем для розпізнавання образів.
- Предмет дослідження: технології реалізації нейронних мереж.

Для досягнення поставленої мети визначено такі завдання:

- Провести аналіз теоретичних основ використання нейронних мереж у різних сферах.
- Оцінити існуючі типи нейронних мереж та вибрати найбільш відповідний.
- Розглянути архітектури нейронних мереж та вибрати найрелевантнішу для реалізації мети.
- Вивчити інструменти та мови програмування для розробки нейронних мереж.
- Оцінити базову модель нейронної мережі та її точність на різних наборах даних.
- Підсумувати результати роботи.

У дослідженні використовувались загальнонаукові методи, такі як спостереження, порівняння, рахунок, вимірювання, експеримент, узагальнення, абстрагування, формалізація.

Практичне значення отриманих результатів полягає в можливості застосування розробленої системи для автоматизації процесів розпізнавання зображень у таких сферах, як безпека, виробництво та інші галузі, де необхідна швидка та точна ідентифікація об'єктів.

# РОЗДІЛ 1

## ТЕОРЕТИЧНІ ОСНОВИ

### 1.1 Основи нейронних мереж

#### 1.1.1 Визначення та історія

Нейронні мережі є частиною обчислювальної техніки, яка досліджує системи, що імітують принципи роботи біологічних нейронних мереж у людському мозку. Головна мета нейронних мереж полягає у вирішенні складних завдань шляхом навчання та адаптації до умов довкілля. Вони здатні виконувати такі завдання, як розпізнавання образів, обробка мовлення та різні види класифікації, які традиційно вважалися складними для звичайного програмування.

Розвиток нейронних мереж почався у середині 20-го століття. У 1943 році психолог Воррен Маккалох та математик Вальтер Пітс створили математичну модель біологічного нейрону, що була опублікована у їхній праці "A Logical Calculus of the Ideas Immanent in Nervous Activity". Ця модель стала основою для подальших досліджень у галузі штучних нейронних мереж.

У 1950-х роках, завдяки розвитку технологій та збільшенню обчислювальних потужностей, дослідник Френк Розенблатт розробив перцептрон — одну з перших нейронних мереж, здатних навчатися розпізнавати патерни за допомогою алгоритмів навчання. Перцептрон став важливою віхою, оскільки продемонстрував можливість застосування машинного навчання для практичних завдань.

Однак ентузіазм щодо перцептрона та інших ранніх моделей нейронних мереж зменшився після публікації книги 1969 року "Перцептрони" Марвіна Мінського та Сеймура Пейперта. Вони критикували основні обмеження перцептронів, зокрема їхню нездатність розв'язувати нелінійно сепарабельні задачі, такі як проблема XOR. Це призвело до скорочення фінансування досліджень у цій галузі, що стало відомо як "перша зима штучного інтелекту".



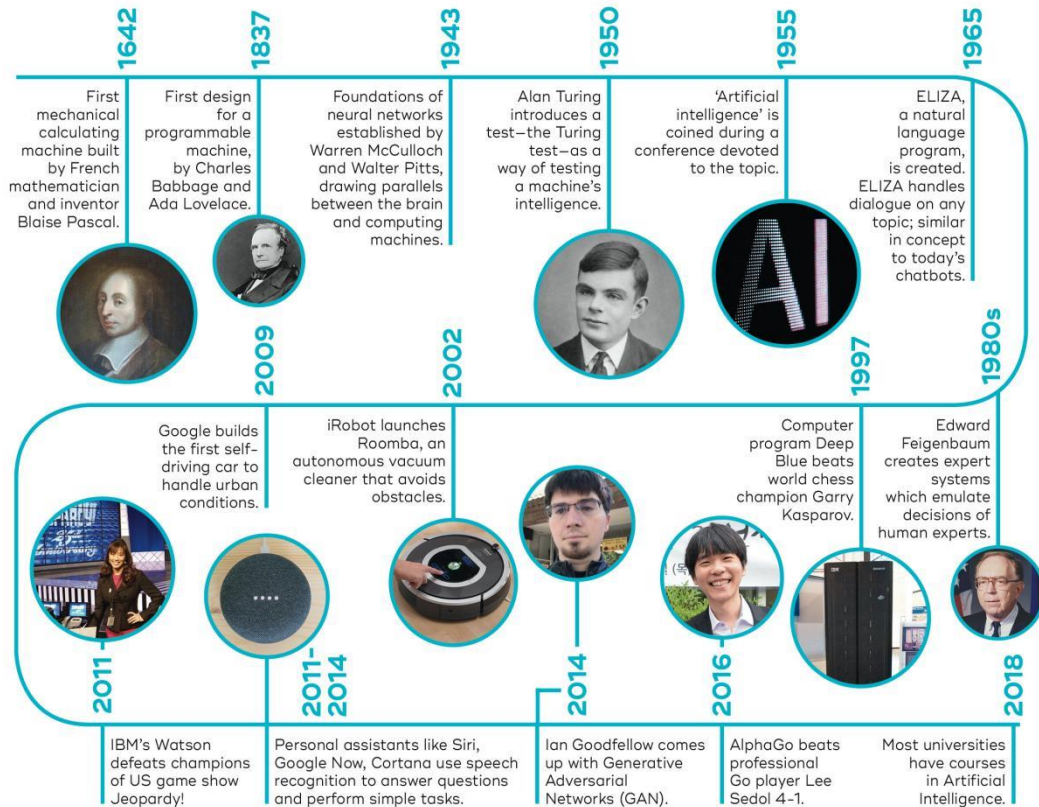


Рис. 1.1. – Історія штучного інтелекту

Інтерес до нейронних мереж повернувся у 1980-х роках з появою нових алгоритмів навчання, зокрема алгоритму зворотного поширення помилки, який був представлений у 1986 році Девідом Румельхартом, Джефрі Хінтоном та Рональдом Вільямсом. Цей метод дозволив тренувати багат шарові нейронні мережі, значно підвищивши їхню ефективність і здатність вирішувати нелінійні задачі. Відтоді галузь нейронних мереж продовжує розвиватися, особливо з появою згорткових нейронних мереж у 1990-х роках, які стали основою сучасних систем обробки зображень та відеоаналізу. Розвиток глибоких нейронних мереж, великий обсяг доступних даних та значне збільшення обчислювальних потужностей сприяли швидкому прогресу в машинному навчанні та штучному інтелекті, відкриваючи нові можливості в різних галузях та індустріях.

### 1.1.2 Типи нейронних мереж

Нейронні мережі мають різноманітні структури і призначені для вирішення широкого спектра завдань, починаючи від простих

класифікаційних задач і закінчуючи складними проблемами, такими як автоматичне генерування тексту. Серед основних типів нейронних мереж виділяються повністю з'єднані (feedforward) нейронні мережі, згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та мережі глибокого навчання з підсиленням.

Повністю з'єднані (feedforward) нейронні мережі є найбільш простим типом, в яких зв'язки між нейронами не утворюють циклів. Це базова модель нейронної мережі, де інформація передається в одному напрямку — від вхідних до вихідних вузлів. Повністю з'єднані мережі широко використовуються для задач класифікації та регресії, а також слугують основою для більш складних архітектур.

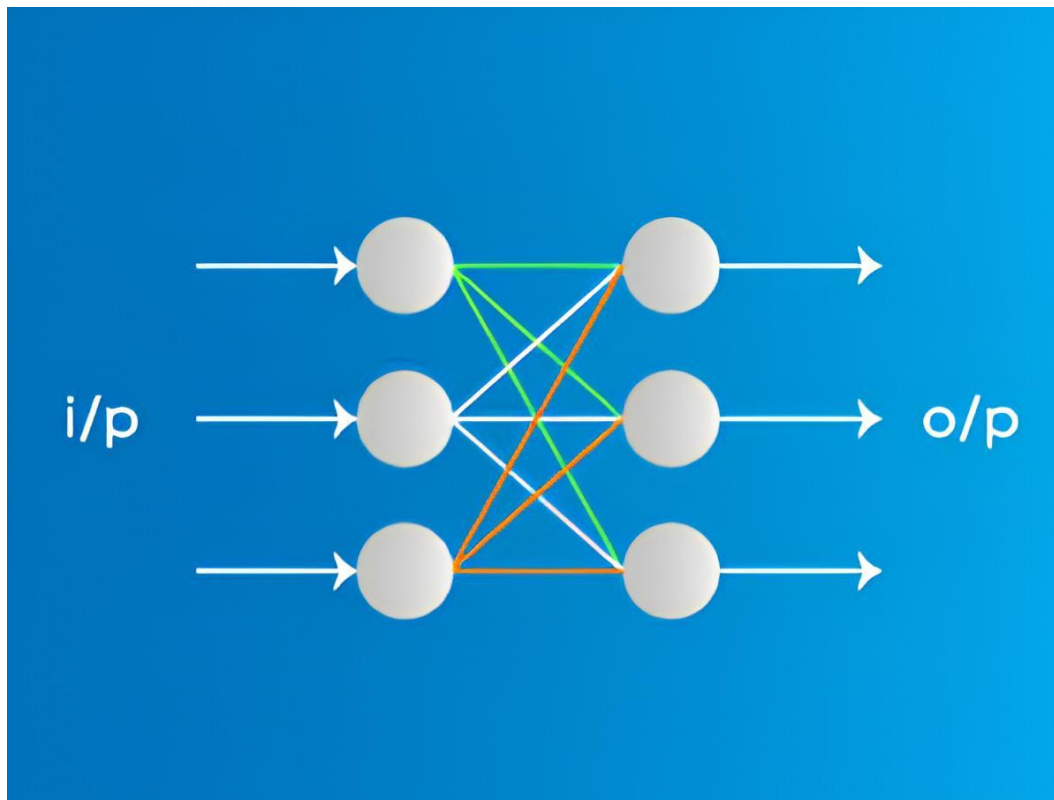


Рис. 1.2. – Нейронні мережі прямого зв'язку

Згорткові нейронні мережі (CNN) в основному використовуються для аналізу візуальних образів, оскільки вони здатні вловлювати просторові ієрархії у даних. CNN впроваджують шари згортки, які фільтрують вхідні дані, створюючи карти характеристик, що відображають наявність таких особливостей, як краї, кути тощо. Використання таких мереж включає

розпізнавання облич, автоматичне тегування фотографій та системи керування автомобілем.

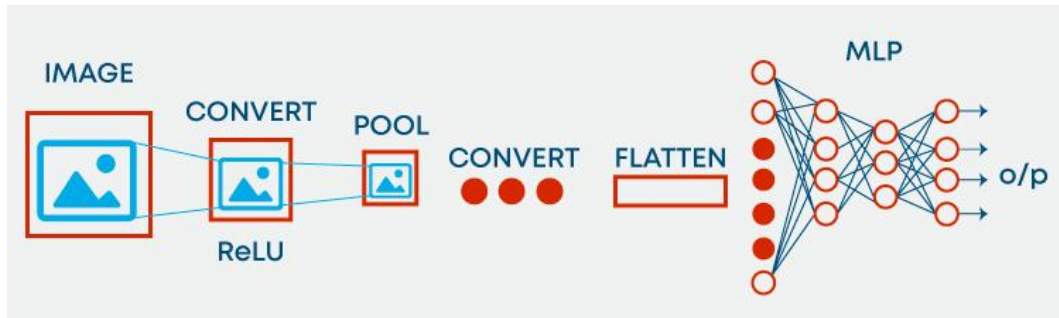


Рис. 1.3. – Згорткова нейронна мережа

Рекурентні нейронні мережі (RNN) оптимізовані для роботи з послідовними даними, такими як текст або часові ряди. Вони використовують свою внутрішню пам'ять для обробки вхідних послідовностей змінної довжини, що робить їх особливо підходящими для завдань, таких як прогнозування наступного слова в тексті або аналіз фінансових часових рядів.

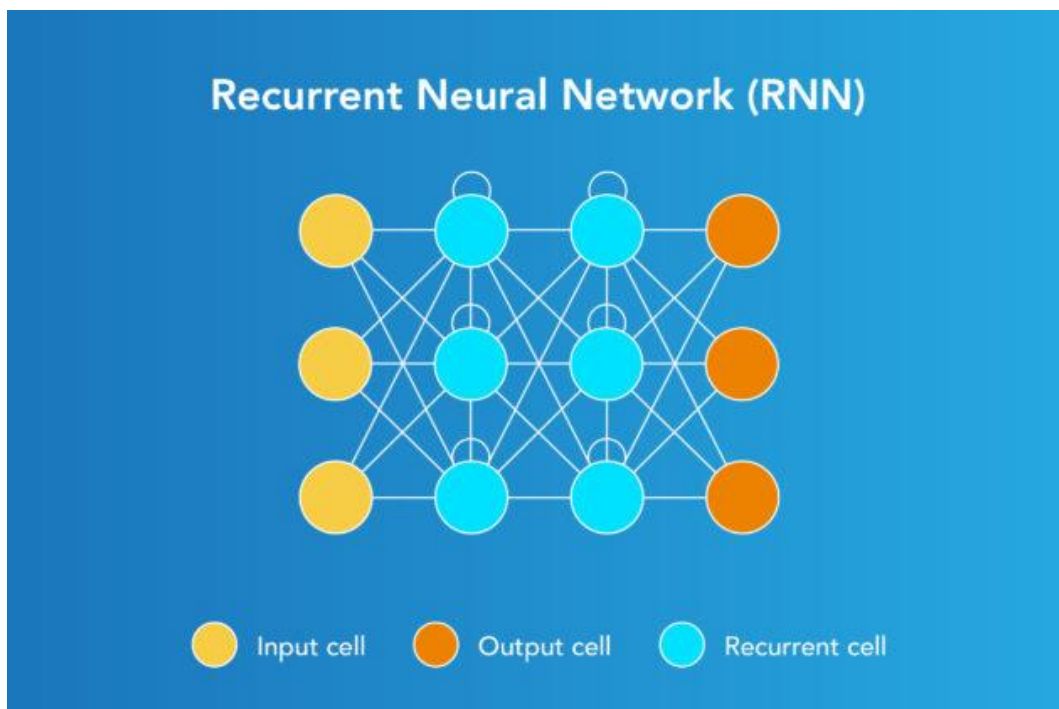


Рис. 1.3. – Рекурентні нейронні мережі

Мережі глибокого навчання з підсиленням використовують методи підсилення для тренування агентів, які намагаються максимізувати отриману нагороду за певний період, часто через спроби та помилки. Ці мережі набули

значної популярності після того, як DeepMind використала їх для тренування системи AlphaGo, яка зуміла перемогти світового чемпіона з гри Го.

Існують також спеціалізовані мережі, як-от генеративні змагальні мережі (GAN). Вони складаються з двох мереж: одна генерує дані (генератор), а інша оцінює ці дані (дискримінатор). GAN ефективно використовуються для створення реалістичних зображень, відео та аудіо з нуля.

### 1.1.3 Перцептрони та багатошарові перцептрони

Перцептрон є одним із найдавніших і найпростіших типів штучних нейронних мереж, запропонований Френком Розенблаттом у 1957 році. Він був створений як алгоритмічна імітація біологічного процесу в мозку, що відповідає за сприйняття та обробку інформації. Перцептрон складається з одного або кількох вхідних нейронів, які передають сигнали до одного вихідного нейрону, здатного виконувати прості класифікаційні завдання.

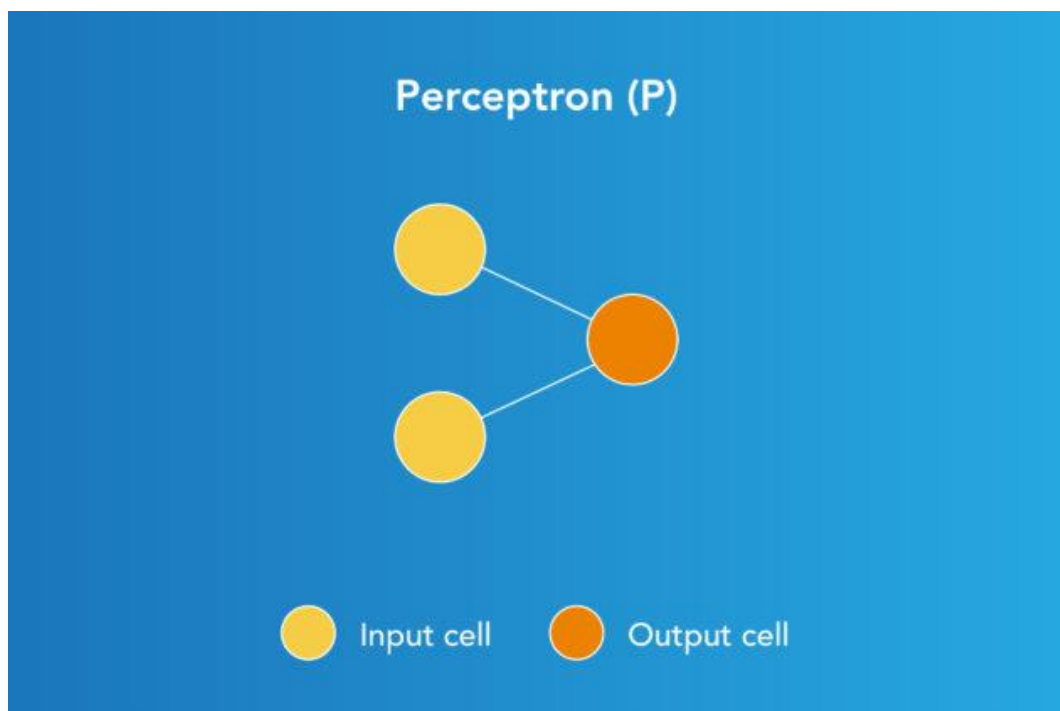


Рис. 1.4. – Пецептрон

#### Структура та принцип роботи перцептрона

Перцептрон приймає кілька вхідних бінарних сигналів і виробляє один вихідний сигнал. Вхідні сигнали множаться на ваги (числові параметри, які

налаштовуються під час тренування мережі), а результати сумуються. Ця сума потім перетворюється на вихідний сигнал за допомогою нелінійної функції, наприклад, порогової функції, яка визначає, чи активувати вихідний нейрон.

### Багатошарові перцептрони (MLP)

Незважаючи на початкові обмеження перцептронів, зокрема їхню неспроможність розв'язувати нелінійно сепарабельні задачі (як-от проблема XOR), вчені розробили більш складні архітектури, відомі як багатошарові перцептрони або MLP. MLP складається з трьох або більше шарів: вхідного шару, одного або декількох прихованих шарів та вихідного шару. Нейрони у прихованих шарах дозволяють MLP моделювати складніші функції порівняно з простим перцептроном.

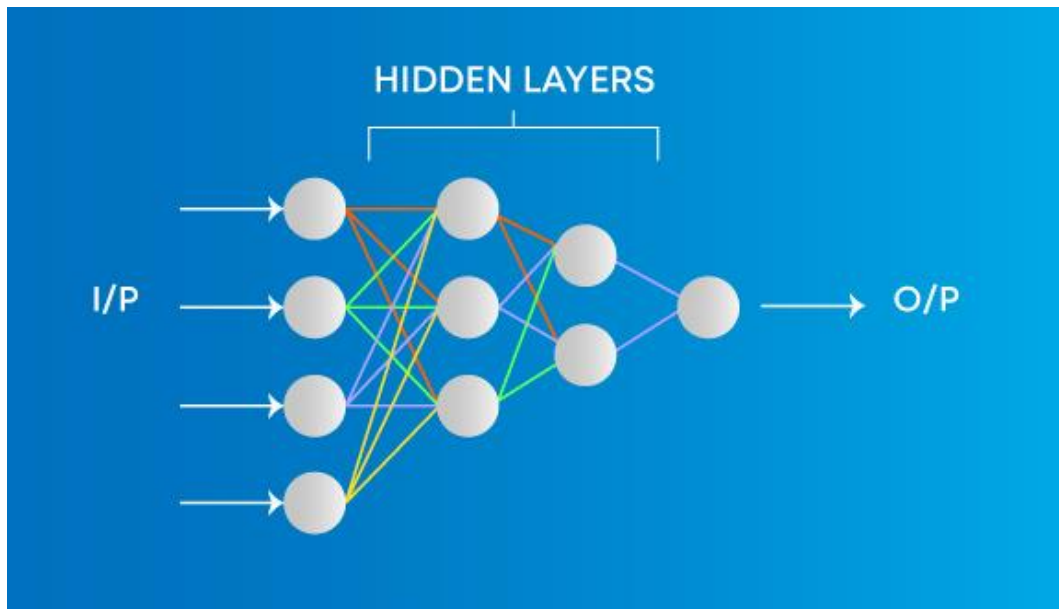


Рис. 1.5. – Багатошаровий перцептрон

### Функціонування MLP

У багатошарових перцептронах кожен нейрон у прихованому шарі з'єднаний з усіма нейронами попереднього шару, причому кожне з'єднання має свою вагу. Сигнали від нейронів попереднього шару множаться на відповідні ваги, сумуються, і результат передається через активаційну функцію, таку як сигмоїдна функція або ReLU. Після цього сигнал передається до наступного шару. Завдяки цьому процесу MLP може апроксимувати будь-яку неперервну

функцію, що підтверджується теоремою про універсальну апроксимацію, що робить MLP надзвичайно потужними у різноманітних застосуваннях.

### **Навчання MLP**

Навчання в MLP здійснюється методом зворотного поширення помилки. У цьому методі обчислюються градієнти функції втрат, а ваги оптимізуються за допомогою методів оптимізації, таких як стохастичний градієнтний спуск. Це дозволяє MLP адаптуватися до складних даних і навчатися, забезпечуючи високу точність у завданнях класифікації, регресії та інших.

### **Застосування MLP**

Завдяки своїй гнучкості та ефективності, MLP використовуються в різних сферах, таких як фінансові прогнози, розпізнавання образів, медичне діагностування та багато інших галузей. Їхня здатність до узагальнення та адаптації робить їх цінним інструментом у сучасному світі штучного інтелекту та машинного навчання.

## **1.2 Згорткові нейронні мережі (CNN)**

### **1.2.1 Архітектура та компоненти**

Згорткові нейронні мережі (CNN) відіграють ключову роль у сучасних системах обробки зображень і відео. Їх унікальна архітектура дозволяє ефективно виділяти важливі особливості з візуальних даних, забезпечуючи високу ефективність у таких завданнях, як розпізнавання образів, класифікація зображень та автоматичне бачення.

#### **Основні компоненти CNN:**

**Згортковий шар (Convolutional Layer):** Це основний будівельний блок CNN. Згортковий шар використовує набір вивчених фільтрів, кожен з яких активно сканує вхідне зображення на предмет специфічних особливостей, таких як краї, кути або текстури. Фільтри застосовуються до зображення за допомогою операції згортки, яка включає множення фільтра на вхідні дані і сумування результатів для формування карти активації (feature map). Карта активації відображає реакцію фільтра на кожну частину зображення.

**Шар пулінгу (Pooling Layer):** Пулінг зменшує розмір карт активацій, зберігаючи лише важливі особливості. Найпоширенішими типами пулінгу є максимальний (max pooling) і середній (average pooling). Максимальний пулінг передає найвище значення з кожної частини карти активації, а середній пулінг передає середнє значення, зменшуючи таким чином обчислювальне навантаження і покращуючи інваріантність до масштабу і зсуву.

**Повністю з'єднаний шар (Fully Connected Layer):** Після декількох згорткових та пулінгових шарів архітектура CNN зазвичай включає один або кілька повністю з'єднаних шарів. У цих шарах нейрони з'єднані з усіма активаціями попереднього шару. Їхня роль полягає у класифікації вхідних даних на основі виявлених особливостей. Повністю з'єднані шари перетворюють візуальні дані у форму, придатну для прийняття кінцевих рішень, таких як класифікація зображення.

**Функція активації (Activation Function):** Важливою складовою всіх шарів у CNN є функція активації, яка додає нелінійність до обробки мережі. Це дозволяє CNN навчатися складнішим і абстрактним шаблонам у даних. Найчастіше використовуються ReLU (Rectified Linear Unit) для згорткових шарів, тоді як сигмоїдна функція або softmax використовуються у вихідному шарі для задач класифікації.

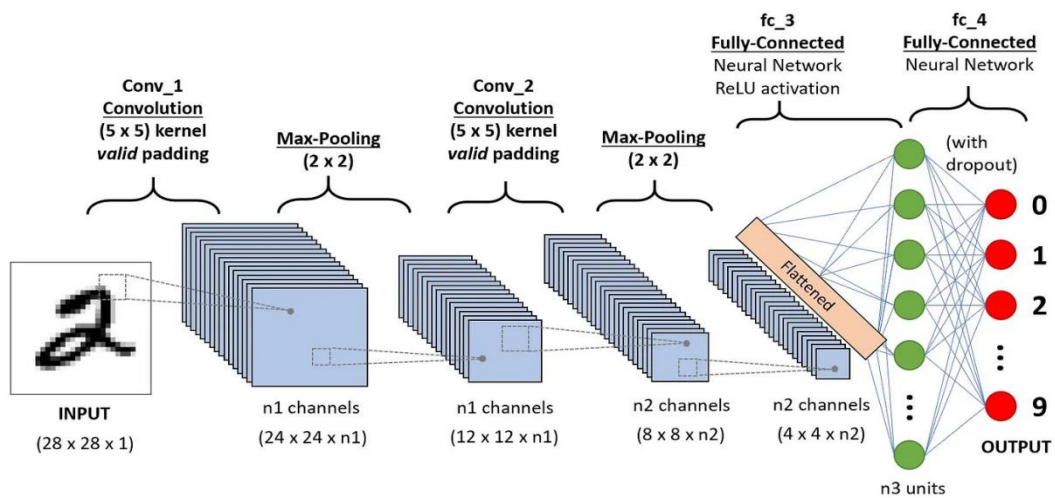


Рис. 1.6. – Архітектура CNN

CNN завдяки своїй структурі та функціонуванню ефективно вирішують завдання з аналізу візуальних даних, надаючи значні можливості в сучасних технологіях машинного навчання та штучного інтелекту.

### **1.2.2 Принципи роботи та навчання**

CNN відрізняються від традиційних нейронних мереж своєю унікальною архітектурою, спеціально розробленою для ефективної роботи з візуальними даними. Принципи роботи та навчання CNN базуються на кількох ключових механізмах, які дозволяють автоматично і ефективно виявляти складні шаблони в зображеннях та інших високорозмірних даних.

#### **Локальне сприйняття і згортка**

На відміну від повністю з'єднаних нейронних мереж, де кожен нейрон пов'язаний з усіма активаціями попереднього шару, у CNN кожен нейрон у згортковому шарі пов'язаний лише з невеликою областю вхідного об'єму, відомою як рецептивне поле. Нейрони в згортковому шарі застосовують фільтри до своїх рецептивних полів, що дозволяє їм виявляти локальні характеристики, такі як краї, кути або текстури, незалежно від їхнього розташування на зображенні.

#### **Просторова ієрархія ознак**

Ключова особливість CNN полягає в їх здатності виявляти складні ознаки на основі простіших. Наприклад, перші шари можуть виявляти прості краї, наступні шари — комбінації країв, що формують кути або складніші геометричні фігури, а глибші шари — розпізнавати об'єкти, складені з цих фігур.

#### **Зворотне поширення та навчання**

Навчання в CNN здійснюється за допомогою методу зворотного поширення помилки. Під час класифікації на виході мережі отримується вектор ймовірностей, який показує, до якого класу належить вхідне зображення. Під час навчання фактичні мітки зображення порівнюються з передбаченими, і розраховується втрата (loss), що показує, наскільки добре



мережа виконує своє завдання. Помилка розповсюджується назад через мережу, що дозволяє оптимізувати ваги фільтрів для мінімізації цієї втрати.

### **Функції активації**

Для додання нелінійності до процесу навчання в CNN використовуються активаційні функції, такі як ReLU (Rectified Linear Unit) або сигмоїд. ReLU є однією з найпопулярніших функцій активації в глибокому навчанні завдяки своїй ефективності та простоті, що допомагає уникнути затухання градієнтів під час навчання.

Завдяки цим принципам CNN демонструють вражаючу ефективність у вирішенні складних завдань візуального аналізу та інших доменів, де потрібно автоматично виявляти та інтерпретувати великі обсяги даних.

### **1.2.3 Застосування у розпізнаванні зображень**

CNN стали ключовим інструментом у розпізнаванні зображень завдяки їх здатності ефективно виявляти візуальні патерни та особливості. Процес розпізнавання зображень за допомогою CNN включає кілька основних етапів, які дозволяють мережі навчатися зі зразків та точно класифікувати нові зображення.

#### **Підготовка даних**

Перший крок у розпізнаванні зображень - це підготовка вхідних даних. Зображення зазвичай піддаються попередній обробці для покращення їх якості та забезпечення уніфікованості вхідних даних. Це може включати зміну розміру, нормалізацію інтенсивності пікселів та аугментацію даних, що збільшує різноманітність тренувального набору за рахунок застосування випадкових трансформацій, таких як повороти, зсуви та масштабування.

#### **Згортка та пулінг**

Ключові компоненти CNN, згорткові шари, застосовують набір фільтрів до вхідних зображень для створення карт характеристик. Кожен фільтр виявляє специфічні особливості, такі як краї, кути чи текстури. Шари пулінгу використовуються для зменшення розміру карт характеристик, роблячи

представлення більш керованим і менш чутливим до точного розташування особливостей у зображенні.

### **Активація та класифікація**

Після кількох шарів згорток та пулінгу, активність мережі проходить через один або кілька повністю з'єднаних шарів, які синтезують інформацію з карт характеристик для класифікації зображення. Це часто включає застосування функції активації, такої як softmax, яка перетворює вихідні значення мережі на ймовірності приналежності до конкретних класів.

### **Зворотне поширення помилки**

Під час тренування CNN використовує метод зворотного поширення для оновлення ваг у мережі на основі помилки між прогнозованими та фактичними мітками. Цей процес включає обчислення градієнтів функції втрат відносно кожної ваги в мережі, що дозволяє оптимізувати параметри для покращення точності класифікації.

## **Висновки до розділу 1**

У першому розділі розглянуто теоретичні основи нейронних мереж, включаючи їхню історію, різні типи, а також детально описано згорткові нейронні мережі та їхнє застосування в розпізнаванні зображень. Вивчення цих аспектів дозволило глибше зрозуміти, як нейронні мережі здатні аналізувати та обробляти інформацію на рівні, подібному до людського сприйняття.

Перцептрони та багат шарові перцептрони визначені як фундаментальні елементи нейронних мереж, які заклали основу сучасних досліджень у галузі машинного навчання. Вони продемонстрували базові принципи навчання та класифікації, що є важливими для розуміння більш складних мереж.

Згорткові нейронні мережі, як основа сучасного комп'ютерного зору, показали свою ефективність у вирішенні складних завдань розпізнавання

образів. Принципи роботи та навчання CNN, включаючи згортку, пулінг, повністю з'єднані шари та активаційні функції, демонструють потужність цих мереж у виявленні ієрархічних ознак з великих даних. Просторова ієрархія ознак, яку забезпечують згорткові шари, дозволяє CNN ефективно адаптуватися до різних типів візуального контенту, поліпшуючи точність і швидкість обробки зображень.

У контексті розпізнавання зображень CNN здатні самостійно вчитися від основних до складних ознак без необхідності ручної інженерії ознак, що робить їх незамінними в багатьох застосуваннях. Їхня здатність обробляти природні образи на високому рівні забезпечила прогрес у таких областях, як медичне зображення, системи спостереження та автономне водіння.

Таким чином, розуміння теоретичних основ нейронних мереж та їхнє практичне застосування у розпізнаванні зображень не лише підвищує точність технологічних систем, але й відкриває нові можливості для інновацій у майбутньому.

## РОЗДІЛ 2

### АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

#### 2.1 Відомі моделі

##### 2.1.1 AlexNet: архітектура та вплив

AlexNet є однією з найвідоміших архітектур згорткових нейронних мереж, яка стала важливою віхою в історії глибокого навчання. Розроблена Олексієм Крижевським, Іллею Суцкевером та Джефрі Гінтоном, ця модель здобула перемогу на конкурсі ImageNet Large Scale Visual Recognition Challenge (ILSVRC) у 2012 році, значно випередивши своїх конкурентів. Її успіх не лише привернув увагу до потенціалу глибоких нейронних мереж у задачах комп'ютерного зору, але й стимулював активні дослідження в цій сфері.

##### Архітектура AlexNet

Архітектура AlexNet складається з восьми вагових шарів, з яких п'ять є згортковими, а три — повністю з'єднаними. Однією з інновацій AlexNet було використання ReLU (Rectified Linear Units) як активаційних функцій, що збільшило нелінійність без значного ускладнення обчислень. Це дозволило прискорити тренування мережі порівняно з традиційними сигмоїдними активаційними функціями.

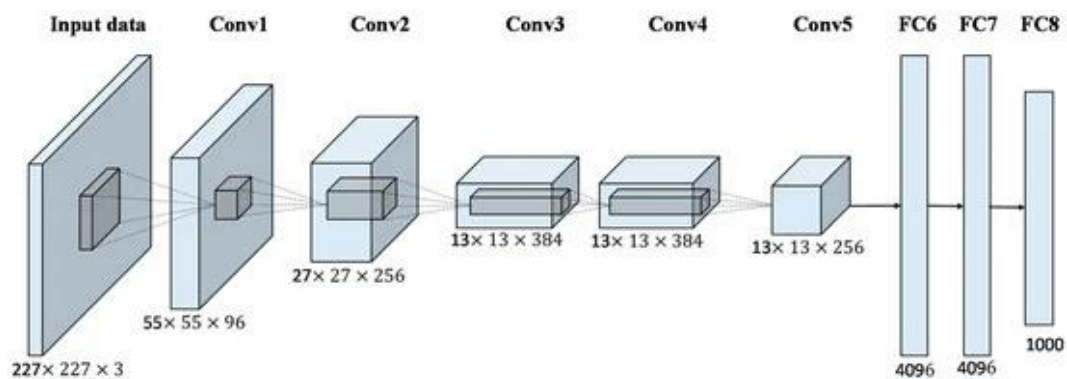


Рис. 2.1. – Архітектура AlexNet

Ще однією важливою особливістю AlexNet було використання техніки "Dropout" для запобігання перенавчанню у повністю з'єднаних шарах. Dropout підвищує здатність мережі до узагальнення, випадково "відключаючи"

частину нейронних з'єднань під час тренування, що допомагає уникнути залежності від конкретних ознак у тренувальному наборі даних.

Також AlexNet використовувала переконволюцію для зменшення розмірності вхідних зображень між згортковими шарами, зберігаючи важливі особливості. Це дозволило збільшити глибину та складність мережі без значної втрати інформації.

### **Вплив на розвиток галузі**

AlexNet значно підвищила інтерес до глибокого навчання і згорткових нейронних мереж у науковому співтоваристві та індустрії. Її успіх на ILSVRC-2012 наочно продемонстрував, що глибокі нейронні мережі можуть перевершувати традиційні методи в багатьох задачах обробки зображень. Після цього значно зросла кількість досліджень і розробок у галузі глибокого навчання, що призвело до швидкого розвитку більш складних та ефективних мережевих архітектур.

Таким чином, AlexNet не лише покращила можливості машинного зору, але й стала ключовим стимулом для інновацій у галузі штучного інтелекту, зокрема у розробці нових методів, технологій та комерційних застосувань, які ми спостерігаємо сьогодні.

#### **2.1.2 VGG: архітектура та варіанти**

VGG (Visual Geometry Group), розроблена групою дослідників з Оксфордського університету, є однією з найвпливовіших архітектур у сфері глибокого навчання та комп'ютерного зору. Ця модель була створена з метою збільшення глибини згорткових нейронних мереж для покращення точності розпізнавання. VGG була представлена у 2014 році і здобула друге місце на змаганні ImageNet того року, завоювавши широке визнання завдяки своїй простій, але ефективній структурі.

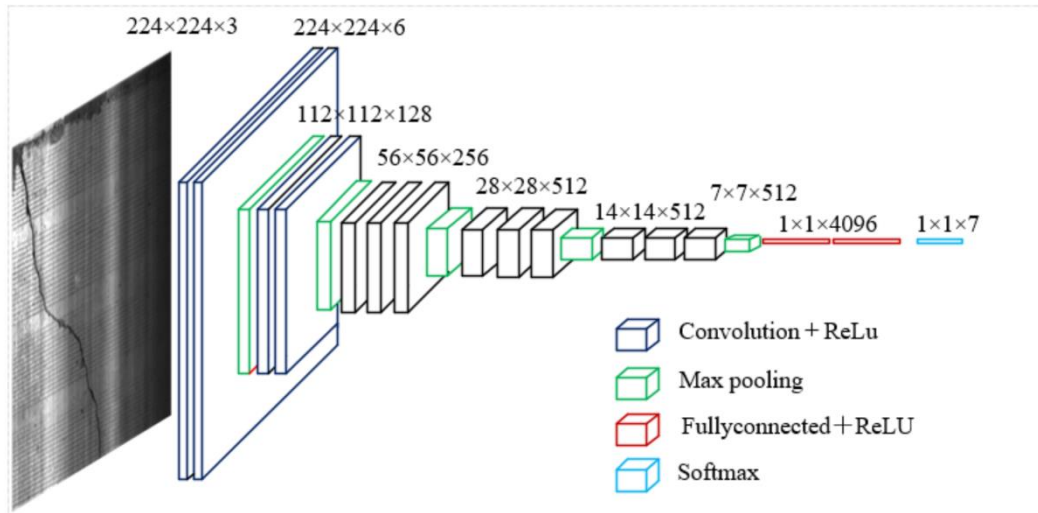


Рис. 2.2. – Архітектура VGG-16

### Відмінності від AlexNet

На відміну від AlexNet, яка використовувала великі згорткові фільтри на початкових шарах (11x11 і 5x5), VGG застосовує послідовність декількох менших фільтрів розміром 3x3 на всіх шарах. Це дозволяє збільшити глибину мережі, зберігаючи відносно малу кількість параметрів. Декілька послідовних згорткових шарів з фільтрами 3x3 можуть охоплювати більший рецептивний контекст порівняно з одним шаром із великим фільтром.

Крім того, VGG використовує більше шарів згортки та активації ReLU на кожному шарі, що сприяє більш глибокому та точному навчанню за рахунок збільшення нелінійності та глибини мережі.

### Варіанти моделі VGG

Основні версії VGG, які набули популярності, включають VGG-16 та VGG-19. Числа вказують на кількість шарів з вагами (згорткові та повністю з'єднані шари). VGG-16 має 13 згорткових шарів і 3 повністю з'єднані шари, тоді як VGG-19 містить 16 згорткових шарів і 3 повністю з'єднані шари. Більша кількість шарів у VGG-19 дозволяє виявляти більш складні особливості в зображеннях, але також збільшує обчислювальну складність і потребу в ресурсах для тренування та впровадження моделі.

Застосування VGG у різних задачах показало, що глибина мережі є критичним фактором у покращенні точності обробки зображень, хоча це

також збільшує складність виправлення помилок і тренування. Незважаючи на ці виклики, VGG залишається однією з найпопулярніших архітектур для багатьох досліджень і застосувань у галузі комп'ютерного зору.

### 2.1.3 ResNet: залишкові блоки та переваги

Residual Network (ResNet) є однією з найвпливовіших архітектур у сфері глибокого навчання, запропонованою Kaiming He та його колегами у 2015 році. ResNet революціонізувала можливості глибоких нейронних мереж завдяки впровадженню залишкових блоків, що дозволило тренувати архітектури, які значно перевищують попередні глибини, з тисячами шарів.

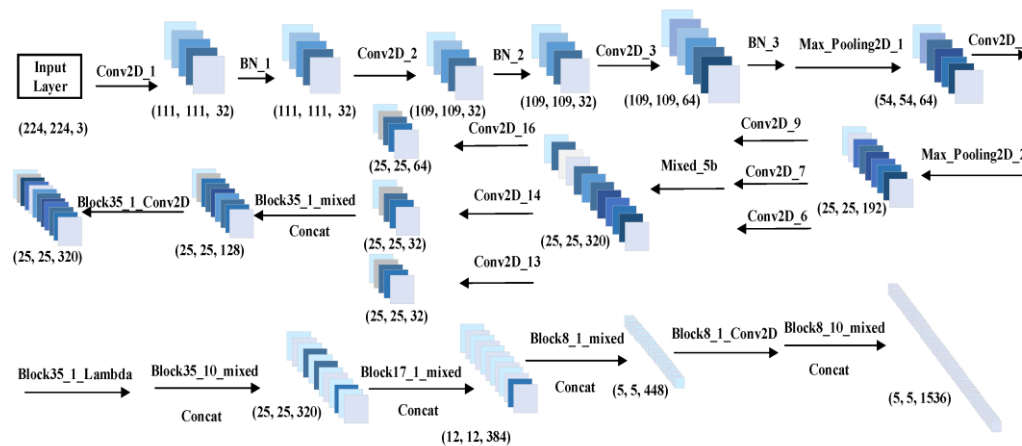


Рис. 2.3. – Архітектура мережі ResNetV2

### Залишкові блоки та їх вплив

Основною інновацією ResNet є використання залишкових блоків, які дозволяють сигналам обходити один або кілька шарів через скіп-з'єднання (skip connections). Ці з'єднання прямо передають вхідні дані на більш глибокі рівні мережі, що зменшує проблему зникнення або вибуху градієнтів під час навчання, забезпечуючи більш стабільне та ефективне навчання глибоких мереж.

Залишкові блоки дозволяють мережі вчитися на різниці між вхідними даними та виходом, замість того, щоб намагатися безпосередньо вивчити вихідні дані. Це спрощує процес навчання, оскільки мережі легше оптимізувати функції, які припускають невеликі коригування до вхідних даних, ніж вивчити весь шаблон від початку.

## **Переваги у навчанні глибоких мереж**

Однією з ключових переваг ResNet є її здатність навчатися ефективно незалежно від мережевої глибини. Традиційно, глибокі мережі стикалися з падінням продуктивності та збільшенням помилок під час навчання через зникнення градієнтів. ResNet вирішує цю проблему, дозволяючи градієнтам легко проходити через архітектуру за допомогою скіп-з'єднань, що забезпечує більшу точність навіть у дуже глибоких мережах.

ResNet демонструє виняткові результати у багатьох задачах комп'ютерного зору та інших областях, де потрібна обробка складних зразків. Вона показала, що збільшення глибини мережі може значно підвищити її продуктивність, якщо є заходи для ефективного управління градієнтами.

У результаті, архітектура ResNet справила значний вплив на розвиток глибокого навчання, стимулюючи розробку нових глибоких та ефективних мережевих структур у різних дослідницьких та промислових застосуваннях, збільшуючи потенціал штучного інтелекту в рішенні все більш складних завдань.

## **2.2 Порівняння моделей**

AlexNet, VGG, і ResNet є ключовими архітектурами в галузі глибокого навчання, кожна з яких має унікальні характеристики та застосування. Ці моделі продемонстрували свою ефективність на стандартних наборах даних, таких як ImageNet, що є важливим бенчмарком для оцінювання алгоритмів у галузі комп'ютерного зору.



Таблиця 1 – Ключові характеристики і застосування моделей AlexNet, VGG, і ResNet

Модель	AlexNet	VGG-16	ResNet-50
Архітектура	8 шарів	16 шарів	50 шарів з залишковими блоками
Параметри	~60 млн	~138 млн	~25 млн
Точність на ImageNet	Топ-5 помилок 15.3%	Топ-5 помилок 7.3%	Топ-5 помилок 3.57%
Характерні застосування	Прототипування, початкове навчання	Розпізнавання зображень, медична імагінг	Автоматичне водіння, системи відеоспостереження

### Вплив і продуктивність

AlexNet здійснила революцію, показавши значний стрибок у точності розпізнавання зображень. З часом моделі, такі як VGG і ResNet, показали ще більші покращення точності. VGG, зі своєю більш глибокою та однорідною структурою, досягла вищої точності, але з більшим обчислювальним навантаженням. ResNet змогла використовувати ще більш глибокі структури без втрати продуктивності завдяки своїм залишковим блокам, що значно підвищило точність без збільшення часу тренування.

### Специфічні сценарії використання

- AlexNet: Використовується в додатках, де переважає потреба в швидкості над високою точністю.
- VGG: Застосовується в задачах, що вимагають високої точності, особливо в областях, де важливі деталі та текстури.
- ResNet: Ідеально підходить для вимогливих застосувань з великою точністю, де критично важливо ефективно навчати дуже глибокі мережі.

### Конструктивні особливості

Конструктивні особливості кожної архітектури мають безпосередній вплив на їхню продуктивність:

- AlexNet: Забезпечує основу для досліджень у глибокому навчанні.
- VGG: Вдосконалює ці здібності за рахунок більшої глибини.

- ResNet: Вирішує проблеми, пов'язані з навчанням дуже глибоких мереж, завдяки своїм залишковим блокам.

## Висновки до розділу 2

У другому розділі ми провели порівняльний аналіз трьох фундаментальних архітектур згорткових нейронних мереж: AlexNet, VGG, і ResNet. Кожна з цих моделей внесла значний вклад у розвиток технологій глибокого навчання і має свої унікальні характеристики та області застосування.

AlexNet стала першопроходцем, продемонструвавши потенціал глибоких нейронних мереж у комп'ютерному зорі. Її успіх спонукав значне зростання інтересу та досліджень у цій області.

VGG покращила підходи до глибини мережі, забезпечуючи більшу точність завдяки своїм однорідним і глибшим структурам. Ця модель демонструє важливість глибини для покращення здатності мережі вчитися складнішим представленням даних.

ResNet, з інноваційним впровадженням залишкових блоків, розв'язала проблему зникнення градієнтів у дуже глибоких мережах, дозволяючи навчання мереж із значно більшою кількістю шарів. Це значно підвищило точність класифікації на складних наборах даних.

Порівняння цих моделей виявило, що глибина і спосіб організації шарів мають критичне значення для точності та ефективності мереж. Кожна модель має свої оптимальні сценарії застосування, що залежать від специфічних вимог до швидкості, точності та обчислювальних ресурсів. Таке розуміння дозволяє краще вибирати та адаптувати нейронні мережі до конкретних задач, забезпечуючи високу продуктивність системи.

## РОЗДІЛ 3

### ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ

#### 3.1 Програмне забезпечення та бібліотеки

У сучасній епосі глибокого навчання вибір правильних інструментів та бібліотек є ключовим для створення ефективних нейронних мереж. Існують різні види програмного забезпечення для розробки нейронних мереж: від низькорівневих бібліотек, які дозволяють детально контролювати кожен аспект навчання, до високорівневих інструментів, що надають можливість швидкого прототипування та впровадження моделей.

Одним з головних факторів, що впливає на ефективність розробки, є здатність інструменту оптимально використовувати обчислювальні ресурси, такі як центральні процесори (ЦПУ), графічні процесори (ГПУ) або навіть тензорні процесори (TPU). Наприклад, TensorFlow і PyTorch демонструють високу продуктивність на ГПУ, що особливо важливо для навчання великих моделей.

Ще один важливий аспект - це здатність до інтеграції з іншими бібліотеками та системами. Інтеграція з популярними бібліотеками для обробки даних, такими як Pandas і NumPy, а також для візуалізації даних, як-от Matplotlib і Seaborn, значно полегшує процес аналізу даних і інтерпретації результатів моделювання.

##### 3.1.1 Python для нейронних мереж

Python є однією з найпопулярніших мов програмування у сфері штучного інтелекту завдяки своїй гнучкості, простоті та широкому набору бібліотек. Ця мова підтримує багато бібліотек для глибокого навчання, таких як TensorFlow, Keras і PyTorch, які забезпечують зручні API для проектування, тренування та валідації нейронних мереж.

Однією з основних переваг Python є його читабельність і простота написання коду, що робить його доступним як для досвідчених програмістів, так і для новачків у цій галузі. До того ж, Python має велику спільноту

розробників та науковців, яка постійно працює над вдосконаленням мови через створення нових бібліотек та інструментів.

Бібліотека NumPy дозволяє ефективно працювати з багатовимірними масивами, що є основою для обробки та аналізу даних у нейронних мережах. Pandas надає високорівневі структури даних і функції для аналізу табличних даних, що є важливими для передобробки даних перед навчанням моделей.

Крім того, інтеграція Python з системами обробки даних і машинного навчання, такими як Apache Spark через бібліотеку PySpark, розширює можливості роботи з великими наборами даних у розподіленому середовищі, що є критичним для великих проектів глибокого навчання.



Рис. 3.1. – Інтеграція Python у глибоке навчання

Ці фактори роблять Python ідеальною мовою для початку роботи з нейронними мережами, а також для їх подальшого розвитку та досліджень у цій галузі.

### 3.1.2 TensorFlow: особливості та переваги

TensorFlow, розроблена компанією Google, є однією з провідних бібліотек для глибокого навчання. Вона забезпечує потужні та гнучкі інструменти для створення, навчання та впровадження моделей машинного навчання.

#### Особливості

- **Граф обчислень:** TensorFlow базується на концепції графа обчислень, де вузли представляють операції, а ребра — тензори, які передаються між вузлами. Ця структура дозволяє візуалізувати модель та оптимізувати її виконання через паралелізм та розподілені обчислення.
- **Автоматичне диференціювання:** TensorFlow автоматично обчислює градієнти для тензорів, що спрощує процес навчання нейронних мереж.

#### Переваги

- **Масштабованість:** TensorFlow підтримує розгортання на різних платформах, від мобільних пристроїв до серверних кластерів, що робить його придатним для комерційного використання та наукових досліджень.
- **Гнучкість:** TensorFlow дозволяє створювати не лише стандартні моделі, такі як CNN або RNN, але й експериментувати з новітніми архітектурами через власний API для визначення нових операцій.

#### Приклади застосування

- **Комп'ютерний зір:** TensorFlow широко використовується для розпізнавання зображень та відеоаналітики. Моделі CNN, побудовані на TensorFlow, ефективно ідентифікують та класифікують об'єкти на зображеннях.
- **Обробка природної мови (NLP):** За допомогою рекурентних мереж або новітніх трансформерів, таких як BERT, TensorFlow застосовується для аналізу текстів, автоматичного перекладу та генерації тексту.
- **Прогнозування часових рядів:** TensorFlow також використовується для аналізу фінансових даних, прогнозування руху на фондових ринках та оптимізації логістичних ланцюгів.

Завдяки активній спільноті та постійному розвитку, TensorFlow продовжує залишатися на передовій інновацій у сфері штучного інтелекту, надаючи дослідникам і розробникам потужні інструменти для реалізації складних проектів глибокого навчання.

### 3.1.3 Keras: спрощення реалізації

Keras — це високорівнева бібліотека для нейронних мереж, написана на Python, яка функціонує як оболонка для більш низькорівневих бібліотек, таких як TensorFlow, Theano або CNTK. Вона була створена для полегшення швидкого експериментування і забезпечення простого переходу від ідеї до результату, без необхідності глибоко занурюватися у складнощі низькорівневих обчислень.

#### Основні особливості Keras

- **Спрощений інтерфейс:** Keras надає прості та зрозумілі абстракції для створення нейронних мереж, що робить код легким для написання та розуміння. Модульна структура бібліотеки дозволяє легко комбінувати необхідні блоки для створення складних архітектур.
- **Гнучкість:** Завдяки підтримці різних бекендів, Keras дозволяє розробникам використовувати одну й ту ж модель на різних обчислювальних платформах, що є великою перевагою для виробничого розгортання.
- **Легка інтеграція:** Як оболонка для більш складних бібліотек, Keras може легко інтегруватися з TensorFlow та іншими аналітичними інструментами для проведення глибоких досліджень і оптимізації моделей.

#### Переваги використання Keras

- **Швидкість розробки:** Keras значно спрощує процес створення прототипів. Лише кілька рядків коду можуть визначити повністю функціональну та оптимізовану модель для складних задач, таких як класифікація зображень.
- **Спільнота та документація:** Keras має одну з найактивніших спільнот розробників і науковців у галузі штучного інтелекту, яка постійно оновлює та розширює інструкції та документацію.

## Приклади застосування Keras

- **Промислове виробництво:** Компанії використовують Keras для розробки інтелектуальних систем контролю якості продукції на конвеєрах, де швидкість реалізації та легкість використання є критичними для тестування та впровадження нових ідей.
- **Медицина:** Keras застосовується для розробки моделей діагностики за допомогою зображень МРТ, КТ та рентгенівських знімків, що дозволяє швидко адаптувати сучасні наукові досягнення до конкретних медичних задач.
- **Стартапи:** Багато стартапів у сфері технологій обирають Keras для розробки своїх перших продуктів, оскільки мінімальні витрати на впровадження та можливість швидкого масштабування є для них привабливими.

Keras є чудовим вибором для дослідників, розробників і стартапів, які бажають скористатися можливостями глибокого навчання без необхідності занурюватися у складні деталі реалізації моделей. Це робить Keras однією з найбільш доступних і популярних бібліотек у світі штучного інтелекту.

## 3.2 Набори даних

У сфері штучного інтелекту та машинного навчання набори даних є ключовими компонентами, оскільки їх якість і різноманітність значною мірою впливають на ефективність і точність моделей. Якісні тренувальні набори даних дозволяють моделям краще виявляти складні закономірності та здійснювати точні прогнози.

### 3.2.1 MNIST: структура та використання

Набір даних MNIST (Modified National Institute of Standards and Technology database) є класичним та одним із найпоширеніших у машинному навчанні та комп'ютерному зорі. Цей набір включає 70,000 зображень рукописних цифр від 0 до 9, які використовуються для тренування та тестування алгоритмів розпізнавання зображень.

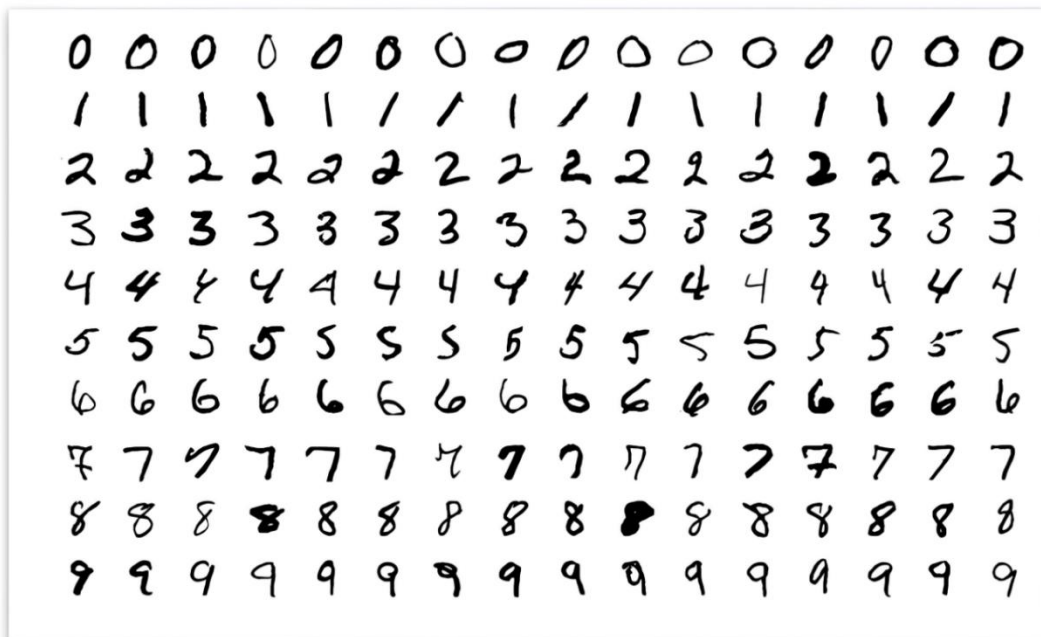


Рис. 3.1. – приклад MNIST

Кожне зображення в MNIST має розмір 28x28 пікселів. Значення пікселів варіюються від 0 до 255, що вказує на інтенсивність чорного кольору, де 0 означає прозорість, а 255 — повну непрозорість. Завдяки цій структурі, MNIST є ідеальним для вивчення обробки та аналізу зображень, а також для розробки та тестування алгоритмів машинного навчання.

Важливою особливістю MNIST є те, що всі зображення нормалізовані за розміром і центровані, що значно полегшує процес навчання нейронних мереж. Це означає, що всі цифри розташовані в центрі зображення, усуваючи потребу в додатковій передобробці для масштабування або вирівнювання.

У глибокому навчанні MNIST часто використовується як базовий датасет, аналогічний "hello world" у програмуванні. Його використання дозволяє новачкам швидко освоїти основи побудови та тренування нейронних мереж, таких як одношарові перцептрони, MLP та CNN. Завдяки простоті та великому обсягу даних, MNIST є ідеальним полігоном для тестування початкових гіпотез, експериментування з архітектурами моделей та налаштування параметрів перед їх застосуванням до більш складних і різноманітних наборів даних.



Важливість MNIST також полягає у його ролі як освітнього інструменту та бенчмарка для алгоритмів. Цей набір даних дозволяє студентам та дослідникам вивчати технічні аспекти роботи з нейронними мережами та глибше зрозуміти ключові концепти машинного навчання, такі як класифікація, втрата та оптимізація. MNIST залишається незамінним ресурсом для освіти та досліджень, навіть у світі, де з'являються все більш складні та різноманітні датасети.

### 3.2.2 ImageNet: огляд та вплив

ImageNet є одним із найбільших та найвпливовіших наборів даних у сфері комп'ютерного зору, розробленим для підтримки розвитку автоматичного розпізнавання образів та машинного навчання. Цей набір включає понад 14 мільйонів анотованих зображень, розподілених у понад 20,000 категорій, кожна з яких відповідає певному синонімічному сету у WordNet. Велика кількість та різноманітність зображень робить ImageNet ідеальним для тренування глибоких нейронних мереж, зокрема CNN.



Рис. 3.1. – ImageNet

Цей набір даних зіграв ключову роль у щорічному змаганні ImageNet Large Scale Visual Recognition Challenge, що значно сприяло прогресу в комп'ютерному зорі та глибокому навчанні. Завдяки ILSVRC, було

продемонстровано потужність різних архітектур нейронних мереж, включаючи такі значущі моделі, як AlexNet, VGG, і ResNet, які показали революційні результати.

Крім впливу на розробку архітектур CNN, ImageNet також вніс значний вклад у стандартизацію методів оцінки ефективності нейронних мереж. Його використання дозволяє порівнювати продуктивність різних моделей, забезпечуючи об'єктивну основу для оцінки та вдосконалення технік машинного навчання.

Завдяки своєму обсягу та складності, ImageNet стимулював розвиток нових технічних підходів у навчанні мереж, таких як методи оптимізації, регуляризації та аугментації даних. Ці інновації допомагають моделям краще узагальнювати знання, що переходять від тренувальних даних до реальних сценаріїв. Такі технологічні досягнення підвищують здатність систем комп'ютерного зору розпізнавати, аналізувати та інтерпретувати зображення в складних і непередбачуваних умовах.

Окрім технічних досягнень, ImageNet також відіграє важливу роль у навчанні. Набір даних є фундаментальним ресурсом для навчання студентів та дослідників основам комп'ютерного зору, машинного навчання та штучного інтелекту. Він допомагає новачкам зрозуміти складність роботи з реальними даними та виклики, що виникають при розробці й впровадженні новітніх технологій.

Завдяки своїй масштабності, складності та впливу на галузь, ImageNet залишається критично важливим інструментом для розвитку та вдосконалення технологій комп'ютерного зору та глибокого навчання.

### **3.2.3 Інші набори даних та їхні особливості**

У сфері машинного навчання існує багато наборів даних, кожен із яких має свої унікальні характеристики та застосування. Ось огляд трьох таких наборів даних, які активно використовуються у наукових дослідженнях та промислових розробках:

- CIFAR-10 і CIFAR-100: Ці набори даних містять кольорові зображення різних об'єктів та тварин у низькій роздільній здатності (32x32 пікселі). CIFAR-10 включає 60,000 зображень, розділених на 10 класів, тоді як CIFAR-100 містить 60,000 зображень, розподілених на 100 класів. Ці набори даних важливі для тестування алгоритмів класифікації зображень, забезпечуючи середню складність між MNIST і ImageNet, що робить їх оптимальними для проміжних етапів навчання та тестування моделей.

- COCO (Common Objects in Context): COCO є одним із найбільших та найрізноманітніших наборів даних для обробки зображень і відео, містить понад 330,000 зображень з анотаціями об'єктів, людей, тварин, транспортних засобів та повсякденних об'єктів. Використовується для задач детекції об'єктів, сегментації інстанцій та розпізнавання капцій. Цей набір даних є важливим завдяки своїй комплексності та різноманітності сценаріїв реального світу, ставши стандартом для тестування передових моделей комп'ютерного зору.

- LFW (Labeled Faces in the Wild): LFW складається з понад 13,000 зображень облич людей, зібраних з Інтернету, кожне з яких анотоване для ідентифікації особи. Цей набір даних переважно використовується для задач верифікації та ідентифікації осіб в неконтрольованих умовах, граючи важливу роль у розвитку алгоритмів біометричної безпеки та розпізнавання осіб, ключових для систем безпеки та мобільних додатків.

Таблиця 2 – Порівняльні характеристики наборів даних CIFAR-10, CIFAR-100, COCO та LFW

Характеристика	CIFAR-10	CIFAR-100	COCO	LFW
Кількість зображень	60,000	60,000	Більше 330,000	Понад 13,000
Роздільність	32x32 пікселі	32x32 пікселі	Різна, включаючи високу роздільність	Різна
Кількість класів	10	100	Велика кількість категорій об'єктів	Ідентифікація особи, не поділена на класи
Тип анотації	Класифікація зображень	Класифікація зображень	Детекція об'єктів, сегментація, капції	Ідентифікація особи
Складність	Середня (простіше ніж ImageNet)	Вища складність ніж CIFAR-10	Висока, комплексні сценарії в реальному світі	Висока, неконтрольовані умови
Застосування	Навчання та тестування моделей	Навчання та тестування моделей	Розширені дослідження в комп'ютерному зорі	Біометрична безпека, мобільні додатки

Кожен із цих наборів даних вносить значний вклад у розвиток та валідацію новітніх технологій машинного навчання, надаючи дослідникам та розробникам інструменти для створення все більш складних і ефективних систем штучного інтелекту. Вони демонструють важливість різноманітності в наборах даних і необхідність точного та ретельного аналізу для подальшого прогресу в галузі.

### Висновки до розділу 3

У цьому розділі ми розглянули різноманітність інструментів та технологій, що застосовуються для розробки та тренування нейронних мереж. Важливість вибору правильних інструментів не можна недооцінювати, оскільки вони значно впливають на ефективність розробки, швидкість тренування моделей та кінцеву точність.

**Програмні бібліотеки:** Ми детально розглянули Python та його бібліотеки, такі як TensorFlow та Keras. Ці інструменти спрощують процес моделювання, надають гнучкість у розробці та підтримують масштабованість рішень.

**Набори даних:** Вибір наборів даних, які використовуються для тренування, є критично важливим. Набори даних як MNIST, CIFAR, ImageNet та інші є не лише інструментами для валідації моделей, але й джерелами вивчення патернів та особливостей, які необхідні для вирішення реальних задач.

**Інновації та розвиток:** Використання передових технологій у тренуванні нейронних мереж дозволяє не тільки підвищити точність моделей, але й забезпечити нові можливості для інновацій у різних галузях, від автоматизації та безпеки до розробки нових лікувальних методик у медицині.

**Ефективність та доступність:** Сучасні інструменти роблять процес розробки нейронних мереж доступнішим, що стимулює дослідження та розвиток у цій області не тільки серед великих компаній, але й у академічних кіл та серед хобістів.

**Освітня роль:** Інструменти та набори даних, які ми розглянули, є важливими не тільки для практичного використання, але й для освітніх цілей. Вони допомагають новим дослідникам швидше входити в галузь та сприяють загальному розвитку наукової громади.

Ці висновки підкреслюють не тільки важливість технологій та інструментів у сучасному світі штучного інтелекту, але й відображають

динамічний характер цієї галузі, де постійний розвиток є ключовим до досягнення нових наукових та технологічних вершин.

## РОЗДІЛ 4

### РЕАЛІЗАЦІЯ НЕЙРОННОЇ МЕРЕЖІ

#### 4.1 Підготовка середовища

Реалізація та тренування моделі нейронної мережі було виконано за допомогою середовища Google Colab, що надає доступ до хмарної платформи з масштабованими обчислювальними ресурсами, включаючи високопродуктивні графічні процесори (GPU). Використання Google Colab дозволяє значно скоротити час тренування моделей глибокого навчання, що є критично важливим для ефективної обробки великих наборів даних. Крім того, середовище Colab пропонує зручні інструменти для спільної роботи та легке інтегрування з Google Drive, що спрощує управління даними та кодом.

Ініціалізація середовища програмування у Colab вимагає налаштування Python і завантаження необхідних бібліотек, які використовуються для створення та тренування нейронної мережі. Нижче наведено код для імпорту бібліотек, що залучені в процесі роботи:

```
# Імпортування необхідних бібліотек
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
```

#### Роз'яснення ролі кожної бібліотеки:

- **NumPy (np):** Ця бібліотека є фундаментом для наукових обчислень у Python. Вона надає потужні мультидименсіональні масиви та інструменти для роботи з цими масивами. Використання NumPy у цьому контексті дозволяє зручно виконувати операції з даними перед їх поданням у нейронну мережу.
- **Matplotlib (plt):** Використовується для створення графіків, які можуть відображати процес тренування або показувати зображення з датасету.

Візуалізація даних є ключовим аспектом аналізу під час розробки моделей машинного навчання.

- **TensorFlow і Keras:** TensorFlow - це відкрита бібліотека для машинного навчання, розроблена Google для роботи з великими обсягами даних. Keras, будучи частиною TensorFlow, надає простіші інструменти для створення та тренування нейронних мереж, що дозволяє швидко експериментувати з різними архітектурами.

## 4.2 Завантаження та підготовка даних

Один із ключових аспектів реалізації нейронної мережі полягає у виборі та підготовці відповідного набору даних. Для цієї роботи використовувався набір даних Fashion MNIST, який є стандартним набором у галузі машинного навчання для бенчмаркінгу алгоритмів класифікації зображень.

### Завантаження даних

Fashion MNIST складається з 70,000 зображень, розподілених на 60,000 для тренування та 10,000 для тестування. Кожне зображення представляє одну з 10 категорій одягу і має розмір 28x28 пікселів, представлений в градаціях сірого (чорно-біле). Цей набір даних вже стандартно розділений на тренувальний та тестовий набори, що полегшує процес підготовки та оцінки моделі. Код для завантаження даних за допомогою TensorFlow Keras API:

```
(train_images, train_labels), (test_images, test_labels)
= keras.datasets.fashion_mnist.load_data()
```

### Визначення класів

Оскільки назви категорій одягу відсутні безпосередньо у наборі даних, важливо визначити їх для подальшого використання в аналізі та візуалізації. Класи визначені як наступні:

```
class_names = ['T-shirt/top', 'Trouser', 'Pullover',
'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle
boot']
```



### Підготовка даних

Перед тренуванням моделі, зображення були нормалізовані для перетворення значень пікселів з діапазону 0-255 у діапазон 0-1. Ця нормалізація є важливим кроком, оскільки вона допомагає підвищити швидкість тренування мережі, забезпечуючи кращу конвергенцію. Ось як це виглядало в коді:

```
train_images = train_images / 255.0  
test_images = test_images / 255.0
```

### Переваги використання Fashion MNIST

Fashion MNIST вважається покращеною альтернативою класичному набору даних MNIST з рукописними цифрами. Його переваги полягають у тому, що він більш складний у розпізнаванні форм та моделей, але при цьому не настільки складний, що вимагає передових архітектур нейронних мереж. Така складність робить його ідеальним для базових навчальних проектів та експериментів з новими архітектурами мереж або оптимізаційними стратегіями.

## 4.3 Підготовка середовища

Реалізація та тренування моделі нейронної мережі було виконано за допомогою середовища Google Colab, що надає доступ до хмарної платформи з масштабованими обчислювальними ресурсами, включаючи високопродуктивні графічні процесори (GPU). Використання Google Colab дозволяє значно скоротити час тренування моделей глибокого навчання, що є критично важливим для ефективної обробки великих наборів даних. Крім того, середовище Colab пропонує зручні інструменти для спільної роботи та легке інтегрування з Google Drive, що спрощує управління даними та кодом.

Ініціалізація середовища програмування у Colab вимагає налаштування Python і завантаження необхідних бібліотек, які використовуються для

створення та тренування нейронної мережі. Нижче наведено код для імпорту бібліотек, що залучені в процесі роботи:

```
# Імпортування необхідних бібліотек
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
```

#### Роз'яснення ролі кожної бібліотеки:

- **NumPy (np):** Ця бібліотека є фундаментом для наукових обчислень у Python. Вона надає потужні мультидименсіональні масиви та інструменти для роботи з цими масивами. Використання NumPy у цьому контексті дозволяє зручно виконувати операції з даними перед їх поданням у нейронну мережу.
- **Matplotlib (plt):** Використовується для створення графіків, які можуть відображати процес тренування або показувати зображення з датасету. Візуалізація даних є ключовим аспектом аналізу під час розробки моделей машинного навчання.
- **TensorFlow і Keras:** TensorFlow - це відкрита бібліотека для машинного навчання, розроблена Google для роботи з великими обсягами даних. Keras, будучи частиною TensorFlow, надає простіші інструменти для створення та тренування нейронних мереж, що дозволяє швидко експериментувати з різними архітектурами.

#### 4.4 Візуалізація даних

Візуалізація даних є важливою частиною аналітичного процесу в машинному навчанні, особливо під час підготовки даних та первинного аналізу. Це дозволяє не тільки перевірити якість даних, але й зрозуміти особливості набору даних, з яким працює модель. У контексті Fashion MNIST, візуалізація зображень з набору даних може допомогти ідентифікувати, як зображення представлені та як вони відрізняються між різними класами.

### Код для візуалізації зразків даних

Для візуалізації зображень з набору даних Fashion MNIST використовується бібліотека Matplotlib. Наступний код створює сітку із 25 зображень (5x5), де кожне зображення анотоване міткою, що вказує на клас одягу:

```
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i], cmap=plt.cm.binary)
    plt.xlabel(class_names[train_labels[i]])
plt.show()
```

Цей блок коду виконує наступні дії:

- **Створення фігури:** Встановлює розмір вікна візуалізації для зручного перегляду.
- **Перебір зображень:** Цикл for пробігає через перші 25 зображень тренувального набору даних.
- **Підготовка субплотів:** Кожне зображення розміщується в окремій клітинці сітки 5x5.
- **Відображення зображень:** Зображення відображаються у відтінках сірого з використанням колірної карти binary.
- **Анотація зображень:** Під кожним зображенням вказується назва класу, до якого воно належить, за допомогою мітки xlabel.

### Значення візуалізації

Ця візуалізація допомагає не тільки зрозуміти, які типи зображень входять до набору даних, але й оцінити різноманітність зразків, що представлені в кожному класі. Візуальний огляд даних може також допомогти виявити можливі проблеми з якістю даних, такі як неоднорідність зображень або помилки в мітках.



Рис. 4.1. Візуалізація матриці даних

## 4.5 Побудова нейронної мережі

Після підготовки і візуалізації даних наступним кроком у реалізації нейронної мережі є її побудова. Використання Keras, високорівневого API TensorFlow, дозволяє ефективно створювати та тренувати моделі машинного навчання.

Код для створення моделі:

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```

### Опис компонентів моделі:

- **Flatten Layer:** Цей шар перетворює кожне вхідне зображення з двовимірної матриці (28x28 пікселів) в одновимірний масив (784 пікселі). Це необхідно, оскільки наступний щільний шар (Dense) вимагає вхідних даних у вигляді вектора.
- **Dense Layer:** Щільний шар або повністю з'єднаний шар є основним будівельним блоком нейронних мереж. Перший щільний шар містить 128 нейронів і використовує ReLU (Rectified Linear Unit) як функцію активації. ReLU допомагає моделі вчитися складніші патерни в даних, оскільки вона ефективно вирішує проблему зникаючих градієнтів, зустрічених в сигмоїдних та тангенсних функціях активації.
- **Output Layer:** Вихідний шар складається з 10 нейронів, кожен з яких відповідає за класифікацію однієї з 10 категорій одягу. Функція активації "softmax" використовується в цьому шарі, оскільки вона перетворює вихідні значення моделі у вірогідності, що допомагає в класифікації.

### Компіляція моделі:

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

- **Optimizer:** Adam (Adaptive Moment Estimation) є одним із найефективніших оптимізаторів, оскільки він коригує швидкість навчання на основі параметрів. Це допомагає покращити конвергенцію тренування.
- **Функція втрати:** Sparse Categorical Crossentropy використовується як функція втрат для багатокласової класифікації, що дозволяє моделі ефективно обробляти мітки класів у формі цілих чисел.
- **Metrics:** Точність ('accuracy') використовується для оцінки результатів тренування. Це показує частку правильно класифікованих зображень.

## 4.6 Тренування та оцінка моделі

Тренування моделі є критичним етапом в процесі розробки нейронних мереж, який визначає якість та ефективність кінцевої моделі. Оцінка моделі дозволяє визначити, наскільки добре вона справляється з новими, невідомими даними.

### Тренування моделі

Модель була тренувана на вибірці з 60,000 тренувальних зображень із набору даних Fashion MNIST. Тренування проводилося протягом 10 епох з використанням наступного коду:

```
history = model.fit(train_images, train_labels,  
epochs=10, validation_split=0.2)
```

#### Роз'яснення параметрів:

- **train\_images, train\_labels:** Вхідні дані та відповідні мітки, які будуть використані для тренування моделі.
- **epochs=10:** Кількість епох, протягом яких вся тренувальна вибірка буде пред'явлена моделі. Одна епоха означає один прохід через весь тренувальний набір даних.
- **validation\_split=0.2:** Частка тренувальних даних, яка буде відокремлена та використана як валідаційний набір. У цьому випадку 20% даних використовуються для валідації.

Цей код виконує тренування моделі, використовуючи 80% даних для тренування та 20% даних як валідаційну вибірку, що дозволяє моніторити перенавчання моделі під час тренування. Валідаційна вибірка використовується для оцінки моделі після кожної епохи, надаючи важливі дані про те, як модель справляється з невідомими даними.

### Оцінка моделі

Після завершення тренування модель була оцінена на 10,000 тестових зображеннях, щоб перевірити її загальну точність. Оцінка моделі виконувалася за допомогою наступного коду:

```
test_loss, test_acc = model.evaluate(test_images,  
test_labels, verbose=2)
```

```
print('\nTest accuracy:', test_acc)
```

Цей крок надає остаточну точність моделі, яка є показником того, наскільки добре модель здатна класифікувати нові зображення, які вона не бачила під час тренування.

### Візуалізація процесу тренування

Для кращого розуміння процесу тренування і ідентифікації потенційного перенавчання корисно візуалізувати зміни втрат і точності на тренувальних та валідаційних даних протягом епох:

```
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'],
label='validation accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0, 1])
plt.legend(loc='lower right')
plt.show()
```

Ця візуалізація допомагає виявити етапи, на яких модель може почати перенавчатися, тобто коли точність на тренувальних даних продовжує рости, тоді як на валідаційних даних вона стагнує або падає.

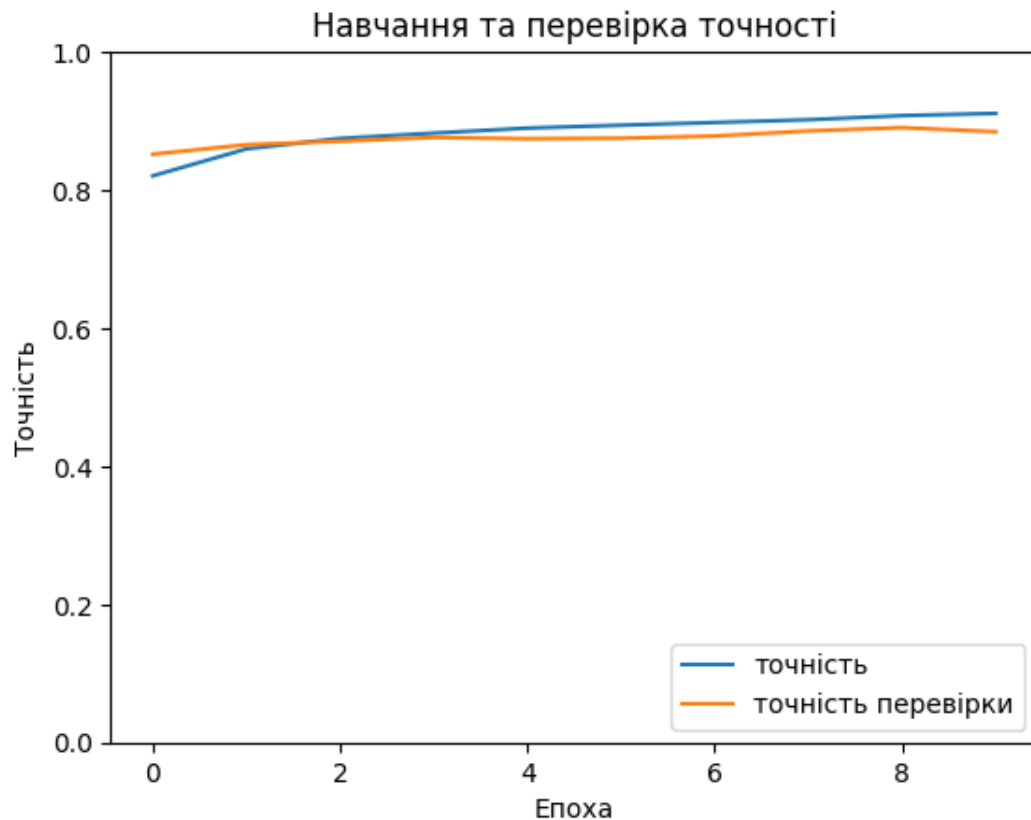


Рис. 4.2. Візуалізація процесу тренування

#### 4.7 Аналіз помилок та інші метрики

Детальний аналіз помилок та різноманітні метрики оцінки важливі для глибшого розуміння поведінки нейронної мережі. Ці інструменти допомагають ідентифікувати слабкі сторони моделі та напрямки для подальших удосконалень.

##### Матриця помилок

Матриця помилок дозволяє візуалізувати, як модель класифікує зображення, вказуючи, які класи часто плутаються. Для створення та візуалізації матриці помилок використовуються наступні кроки:

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
# Передбачення класів на тестовому наборі
predictions = model.predict(test_images)
predicted_classes = np.argmax(predictions, axis=1)
# Створення матриці помилок
```



```

cm = confusion_matrix(test_labels, predicted_classes)
# Візуалізація матриці помилок
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt="d", cmap='Blues')
plt.title('Confusion Matrix')
plt.ylabel('Actual Labels')
plt.xlabel('Predicted Labels')
plt.show()

```

Ця візуалізація надає чітке уявлення про те, які класи модель класифікує точно, а які плутає. Це особливо корисно для ідентифікації шаблонів помилок, які можуть вказувати на необхідність змін в підготовці даних або архітектурі моделі.

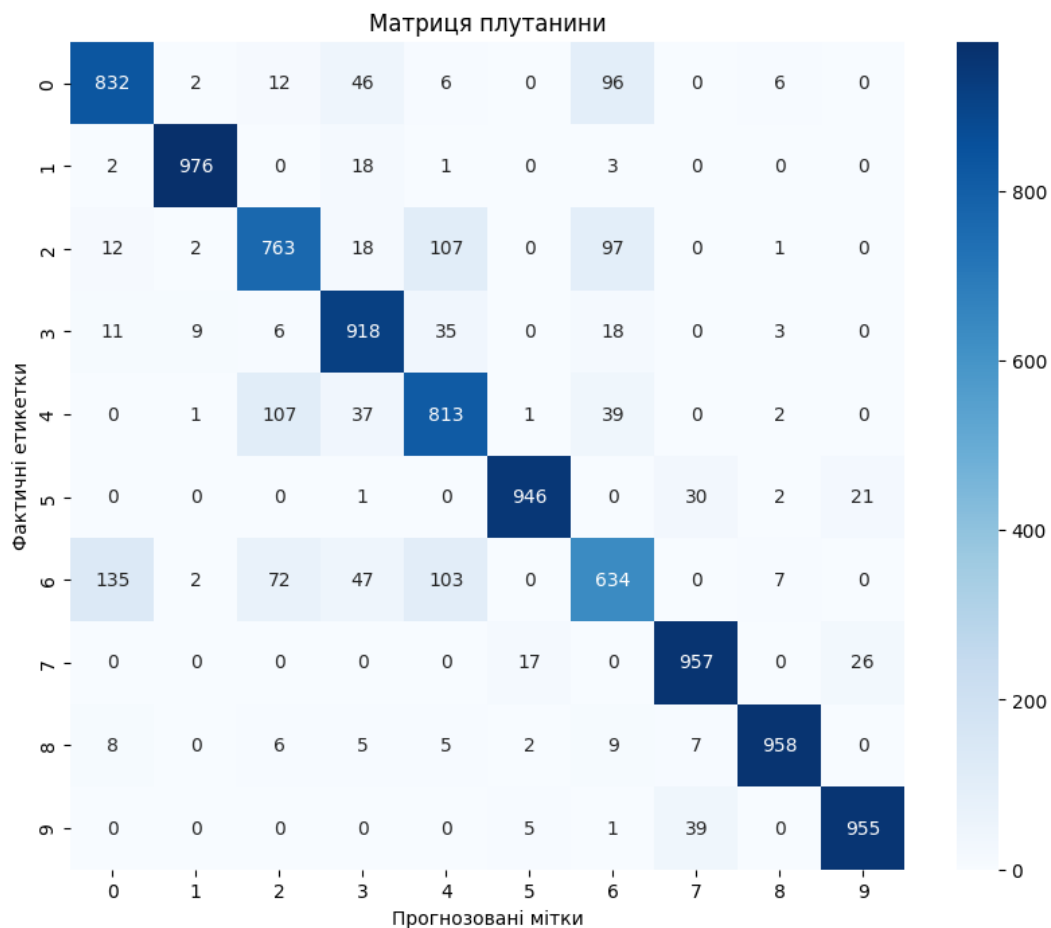


Рис. 4.3. Матриця помилок

### Звіт про класифікацію

Для оцінки точності, повноти та F1-середнього для кожного класу використовується звіт про класифікацію:

```

from sklearn.metrics import classification_report
# Генерація звіту про класифікацію
report = classification_report(test_labels,
predicted_classes)
print(report)

```

Цей звіт надає цінну інформацію про продуктивність моделі по кожному класу, що допомагає виявити класи, для яких модель виконує недостатньо добре. Це може вказувати на необхідність більш збалансованого набору тренувальних даних або на необхідність рефінансування гіперпараметрів для певних класів.

Таблиця 3 – Звіт результати класифікації моделі,

Клас	Точність	Відгук	F1-бал	Підтр.
0 T-shirt	0.83	0.83	0.83	1000
1 Trouser	0.98	0.98	0.98	1000
2 Pullover	0.79	0.76	0.78	1000
3 Dress	0.84	0.92	0.88	1000
4 Coat	0.76	0.81	0.79	1000
5 Sandal	0.97	0.95	0.96	1000
6 Shirt	0.71	0.63	0.67	1000
7 Sneaker	0.93	0.96	0.94	1000
8 Bag	0.98	0.96	0.97	1000
9 Boot	0.95	0.95	0.95	1000
Середнє	0.87	0.88	0.87	10000

## **Аналіз результатів**

Загальна точність моделі становить 87%, що є досить високим показником для багатьох застосувань, особливо в контексті бенчмаркінгу таких базових моделей.

### **Точність по класах:**

Клас 1 (Штани) має найвищу точність (0.98) і F1-бал (0.98), що свідчить про те, що модель дуже добре класифікує цю категорію.

Клас 6 (Сорочка) показує найнижчу точність (0.71) і F1-бал (0.67), що може вказувати на змішування цього класу з іншими категоріями, можливо, через схожість

### **Відгук (Recall):**

Високий відгук у класів 1, 5, 7, 8 і 9 (всі мають відгук 0.93 або вище) показує, що модель добре ідентифікує більшість випадків цих класів.

Низький відгук у класу 6 може свідчити про те, що багато справжніх сорочок було неправильно класифіковано як інші категорії.

F1-бал, який є гармонійною середньою між точністю і відгуком, показує збалансовану міру продуктивності кожного класу:

Високі F1-бали для класів, що стосуються взуття (класи 5, 7, 9), підкреслюють, що модель особливо добре справляється з їх розпізнаванням.

Нижчі F1-бали для класів одягу, особливо для сорочок і пуловерів (класи 2, 4, 6), можуть вказувати на потребу в подальших удосконаленнях у класифікації подібних типів одягу.

## **Висновки до розділу 4**

Четвертий розділ роботи детально охоплює процес розробки та оцінки нейронної мережі для класифікації зображень одягу, використовуючи набір даних Fashion MNIST. Використання хмарної платформи Google Colab сприяло ефективному тренуванню моделі за рахунок доступу до розширених обчислювальних ресурсів. Нормалізація та адекватна підготовка даних виявилися критично важливими для забезпечення правильного навчання

моделі, підкреслюючи значення якісної передобробки даних у глибокому навчанні.

Проста архітектура моделі, створена за допомогою Keras, показала хороші результати, що свідчить про можливості базових нейронних мереж у задачах класифікації зображень. Використання матриці помилок та звіту про класифікацію дозволило детально оцінити продуктивність моделі, ідентифікувати її сильні сторони та області, що вимагають покращення.

Аналіз перенавчання моделі через динаміку точності на тренувальних та валідаційних даних виявив ознаки перенавчання, що є ключовим для розробки ефективних моделей глибокого навчання. Загалом, результати дослідження демонструють потенціал застосування глибокого навчання для вирішення практичних задач у сфері комп'ютерного зору.

## ВИСНОВКИ

Проведене дослідження підтвердило важливість і ефективність використання нейронних мереж для розпізнавання зображень. У процесі роботи було вивчено теоретичні основи нейронних мереж, проведено аналіз існуючих рішень та досліджено технології й інструменти для їх розробки та навчання. Отримані результати показують, що нейронні мережі можуть значно підвищити ефективність і точність розпізнавання зображень, що має важливе практичне значення для багатьох галузей.

Було вивчено історію розвитку нейронних мереж, їх основні типи та архітектури. Зокрема, згорткові нейронні мережі (CNN) виявилися найбільш ефективними для розпізнавання зображень. Значну увагу приділено історичному розвитку нейронних мереж, від перших концепцій до сучасних моделей, що активно використовуються сьогодні. Проведено порівняльний аналіз моделей AlexNet, VGG та ResNet, які є фундаментальними архітектурами в галузі глибокого навчання. Виявлено, що кожна з цих моделей має свої унікальні переваги та обмеження, а їх застосування залежить від конкретних завдань і вимог. AlexNet стала піонером у сфері глибокого навчання, продемонструвавши високу ефективність і точність. Модель VGG вирізняється своєю глибиною і детальною структурою, що дозволяє досягти більшої точності у розпізнаванні. ResNet, зі своєю інноваційною архітектурою залишкових блоків, подолала проблему зникнення градієнтів, дозволяючи створювати ще глибші та ефективніші моделі.

Досліджено використання мови програмування Python і бібліотек TensorFlow та Keras для розробки нейронних мереж. Python є основною мовою для розробки нейронних мереж завдяки своїй простоті, гнучкості та наявності великої кількості спеціалізованих бібліотек. TensorFlow і Keras є провідними інструментами для створення, навчання і тестування нейронних мереж. Вони забезпечують високу продуктивність і зручність у роботі, дозволяючи розробникам швидко створювати і тестувати нові моделі. Досліджено

популярні набори даних, такі як MNIST і ImageNet, які використовуються для тренування і тестування нейронних мереж. MNIST є класичним набором даних для тестування алгоритмів розпізнавання зображень, а ImageNet — масштабним і різноманітним набором, який став стандартом для оцінки ефективності різних моделей глибокого навчання.

Практичне значення отриманих результатів полягає в можливості застосування розробленої системи для автоматизації процесів розпізнавання зображень у різних сферах. Зокрема, такі системи можуть бути використані в системах безпеки для ідентифікації осіб, виявлення підозрілих об'єктів і ситуацій. Використання нейронних мереж у системах безпеки значно підвищує рівень захисту, оскільки вони здатні швидко і точно розпізнавати обличчя, автомобільні номери та інші об'єкти, що дозволяє оперативно реагувати на потенційні загрози. У виробництві нейронні мережі можуть бути застосовані для контролю якості продукції та виявлення дефектів, що сприяє підвищенню ефективності виробничих процесів і зменшенню втрат. У медицині системи на основі нейронних мереж використовуються для аналізу медичних зображень, таких як рентгенівські знімки, МРТ та КТ. Це дозволяє лікарям швидше і точніше діагностувати захворювання, що значно підвищує якість медичної допомоги. У транспорті нейронні мережі можуть бути використані для автоматичного розпізнавання номерних знаків, контролю дорожнього руху та забезпечення безпеки на дорогах. Автоматизація цих процесів дозволяє зменшити кількість дорожньо-транспортних пригод і підвищити ефективність роботи транспортних систем.

У сучасному світі швидкого розвитку технологій нейронні мережі є потужним інструментом, який дозволяє значно підвищити ефективність і точність процесів розпізнавання зображень. Проведене дослідження підтвердило, що нейронні мережі здатні навчатися на великих обсягах даних і постійно вдосконалюватися, що робить їх незамінними у багатьох галузях. Подальші дослідження і розвиток у цій сфері відкривають нові можливості для автоматизації та оптимізації різноманітних процесів, забезпечуючи високий

рівень точності і надійності результатів. Постійне вдосконалення алгоритмів і архітектур нейронних мереж, а також розвиток обчислювальних потужностей сприяє розширенню їх застосувань і підвищенню ефективності в різних галузях. Інновації в цій сфері продовжують змінювати наше уявлення про можливості машинного навчання та штучного інтелекту, роблячи їх невід'ємною частиною сучасного життя і технологічного прогресу.

Таким чином, дослідження і розвиток нейронних мереж є критично важливими для подальшого прогресу в галузі штучного інтелекту. Їх застосування відкриває нові горизонти для вирішення складних завдань і створення інноваційних рішень, що сприяє покращенню якості життя і підвищенню ефективності різноманітних процесів у сучасному світі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. McCulloch, W. S., & Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*.
2. Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*.
3. Minsky, M., & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. The MIT Press.
4. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
6. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
7. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85-117.
8. Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.
9. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*.
10. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *NIPS*.
11. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *ECCV*.
12. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv*.
13. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *CVPR*.
14. Canziani, A., Paszke, A., & Culurciello, E. (2016). An Analysis of Deep Neural Network Models for Practical Applications. *arXiv*.



15. Lutz, M. (n.d.). Learning Python.
16. Chollet, F. (n.d.). Deep Learning with Python.
17. TensorFlow documentation. Available at: <https://www.tensorflow.org/>
18. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
19. Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv.
20. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780.
21. Abadi, M., et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv.
22. Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv.
23. Oliveira, L. S., & Sabourin, R. (2012). Machine Learning and Applications: Machine Learning and Pattern Recognition Methods. Springer.
24. Silver, D., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.
25. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*.
26. Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level Convolutional Networks for Text Classification. *Advances in Neural Information Processing Systems*.
27. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. arXiv.