

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти: другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДІВ ОПТИМІЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ,

РЕАЛІЗОВАНИХ ЗАСОБАМИ

МАШИННОГО НАВЧАННЯ ТА НЕЙРОННИХ МЕРЕЖ

(тема)

Виконав:

студент 2 курсу, групи ІНФМ-23-1

Євтушенко Д.А.

(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика

(повна назва освітньої програми)

Керівник доц. Творошенко І.С.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

_____ (підпис)

Кобилін О.А.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Євтушенку Дмитру Андрійовичу
(прізвище, ім'я, по батькові)1. Тема роботи: Дослідження методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж

затверджена наказом по університету від 25 листопада 2024 року № 1246Ст

2. Термін подання студентом роботи до екзаменаційної комісії 23 грудня 2025р.

3. Вихідні дані до роботи: науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, перелік використовуваних програмних засобів, інтерпретатор мови Python 3.7.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Метод Transfer Learning.

2. Метод Fine-Tuning.

3. Метод Ensemble Methods.

4. Метод Hyperparameter Optimization.

5. Метод Knowledge Distillation.

6. Метод Neural Architecture Search.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність застосування нейронних мереж для вирішення бізнес-процесів, постановка задачі дослідження, етапи дослідження, результати тестування, висновки, перспективи подальшої роботи, апробація результатів дослідження.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	25.11.2024	
2	Аналіз завдання, підбір літератури	26.11.24-03.12.24	
3	Аналіз літератури з досліджуваної проблеми	04.12.24-08.12.24	
4	Дослідження обраних методів тестування	09.12.24-11.12.24	
5	Застосування обраних методів тестування	12.12.24-14.12.24	
6	Програмна реалізація	15.12.24-16.12.24	
7	Оформлення пояснювальної записки	17.12.24-18.12.24	
8	Перевірка на плагіат	19.12.2024	
9	Рецензування	20.12.2024	
10	Підготовка презентації та доповіді	21.12.24-23.12.24	
11	Занесення роботи в електронний архів	02.01.2025	
12	Попередній захист кваліфікаційної роботи	02.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Творошенко І.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 95 с., 18 табл., 19 рис., 47 джерел.

АВТОМАТИЗАЦІЯ, МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, ОПТИМІЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ, ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ.

Об'єктом дослідження є бізнес-процеси різних організацій, що працюють у різних галузях.

Метою дослідження є розробка методів оптимізації бізнес-процесів, що базуються на алгоритмах машинного навчання та нейронних мереж, з метою підвищення ефективності та автоматизації рутинних завдань.

Використано методи машинного навчання, такі як класифікація, кластеризація та регресія, а також різні архітектури нейронних мереж. Проведено аналіз методів прогнозування на основі історичних даних для покращення результатів бізнес-процесів, а також автоматизації завдань.

У ході дослідження було запропоновано новий алгоритм оптимізації бізнес-процесів на основі машинного навчання, який був успішно протестований на реальних даних. У результаті розроблено програмну реалізацію системи, що автоматизує процеси прийняття рішень та підвищує ефективність бізнесу.

AUTOMATION, BUSINESS PROCESS OPTIMIZATION, EFFICIENCY IMPROVEMENT, MACHINE LEARNING, NEURAL NETWORKS.

The object of the research is business processes in various organizations across different industries.

The aim of the research is to develop methods for optimizing business processes based on machine learning algorithms and neural networks to increase efficiency and automate routine tasks.

The research employs machine learning methods such as classification, clustering, and regression. The methods are used to analyze historical data for business process improvement and task automation.

The research proposes a new algorithm for business process optimization using machine learning, which was successfully tested on real-world data. As a result, a software system was developed to automate decision-making processes and enhance business efficiency.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз існуючих методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж.....	9
1.1 Аналіз можливостей оптимізації бізнес-процесів	9
1.2 Класифікація існуючих методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж	12
1.3 Аналіз сучасних методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж.....	18
1.4 Аналіз літературних джерел щодо апробації результатів застосування методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж.....	23
1.5 Постановка задачі дослідження.....	26
2 Особливості обраних методів оптимізації бізнес процесів, реалізованих засобами машинного навчання та нейронних мереж.....	28
2.1 Метод Transfer Learning.....	28
2.2 Метод Fine-Tuning.....	31
2.3 Метод Ensemble Methods.....	35
2.4 Метод Hyperparameter Optimization	39
2.5 Метод Knowledge Distillation	43
2.6 Метод Neural Architecture Search.....	47
3 Дослідження методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж, щодо вибраної предметної області.....	52
3.1 Вибір інструментальних засобів для реалізації вибраних методів..	52
3.2 Етапи програмної реалізації вибраних методів оптимізації бізнес процесів, реалізованих засобами машинного навчання та нейронних мереж.....	54

3.3 Застосування методів оптимізації бізнес-процесів до вибраної предметної області	56
3.3.1 Реалізація методу Transfer Learning	58
3.3.2 Реалізація методу Fine-Tuning	64
3.3.3 Реалізація методу Ensemble Methods	69
3.3.4 Метод Hyperparameter Optimization	72
3.3.5 Метод Knowledge Distillation	76
3.3.6 Метод Neural Architecture Search	79
3.4 Порівняльний аналіз досліджених методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж	82
3.5 Перспективи подальшої роботи	86
Висновки	88
Перелік джерел посилання	91

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

ML – Machine Learning (машинне навчання)

NN – Neural Network (нейронна мережа)

MLP – Multilayer Perceptron (мультишарова персептронна мережа)

RNN – Recurrent Neural Networks (рекурентні нейронні мережі)

CNN – Convolutional Neural Networks (конволюційні нейронні мережі)

ВСТУП

У сучасних умовах бізнес активно адаптується до швидких змін на ринку, глобалізації та цифровізації. Зростання обсягів даних, які генеруються щодня, створює нові виклики для підприємств у питаннях управління процесами, прогнозування і прийняття рішень. Використання машинного навчання (ML) і нейронних мереж (NN) дозволяє значно підвищити ефективність бізнес-процесів, автоматизувати рутинні операції, зменшити витрати і підвищити точність прогнозів. Ця тема є актуальною через необхідність підприємств адаптуватися до цифрової трансформації, щоб залишатися конкурентоспроможними в умовах стрімкого розвитку технологій.

Актуальність даного дослідження полягає в тому, що оптимізація бізнес-процесів є важливим кроком до підвищення ефективності будь-яких підприємств, а використання машинного навчання та нейронних мереж може допомогти в цьому. У сучасних умовах конкуренції підприємства прагнуть знижувати витрати і покращувати продуктивність, тому впровадження новітніх технологій стає необхідністю.

Це дослідження має на меті проаналізувати існуючі методи оптимізації, визначити їх ефективність та запропонувати рекомендації для їх успішного застосування на практиці.

1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ОПТИМІЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ, РЕАЛІЗОВАНИХ ЗАСОБАМИ МАШИННОГО НАВЧАННЯ ТА НЕЙРОННИХ МЕРЕЖ

1.1 Аналіз можливостей оптимізації бізнес-процесів

Оптимізація бізнес-процесів є ключовою складовою ефективного управління організацією, що сприяє підвищенню продуктивності, зниженню витрат і поліпшенню загальної конкурентоспроможності. З розвитком інформаційних технологій та зростанням обсягів даних, які організації повинні обробляти щоденно, виникає потреба у нових методах оптимізації, що виходять за рамки традиційних підходів. Сучасні інструменти, такі як машинне навчання (ML) та нейронні мережі, забезпечують нові можливості для удосконалення бізнес-процесів, дозволяючи автоматизувати рутинні операції, прогнозувати результати і підвищувати ефективність прийняття рішень.

Одним із основних викликів сучасного бізнесу є адаптація до швидко змінюваного ринкового середовища. Традиційні методи управління, засновані на статичних моделях та правилах, більше не відповідають потребам сучасних компаній, які прагнуть гнучкості й адаптивності. Використання ML і нейронних мереж дозволяє компаніям створювати динамічні моделі, здатні швидко адаптуватися до нових умов. Це включає в себе не тільки автоматизацію, але й аналіз великих обсягів даних (Big Data) для прогнозування можливих сценаріїв розвитку подій і підтримки прийняття рішень на основі даних.

Основні напрямки оптимізації бізнес-процесів:

– автоматизація рутинних операцій. Впровадження алгоритмів машинного навчання дозволяє замінити багато рутинних завдань, таких як обробка документів, робота з клієнтськими запитами, розподіл ресурсів та управління запасами [1]. Це дозволяє знизити людський фактор і зменшити

ймовірність помилок. Наприклад, автоматизовані системи управління закупівлями можуть самостійно оцінювати рівні запасів та робити замовлення, базуючись на історичних даних про продажі та поточних трендах на ринку;

– прогнозування попиту і планування. Нейронні мережі можуть бути використані для прогнозування майбутніх продажів або рівня попиту на продукцію [2]. Це особливо важливо для компаній, що займаються виробництвом або дистрибуцією товарів. За допомогою аналізу історичних даних та зовнішніх факторів, таких як зміни в економіці або споживчих перевагах, бізнес може точніше планувати виробничі потужності та розподіл ресурсів;

– оптимізація логістики. Логістичні процеси часто є однією з найскладніших частин бізнесу, що потребують точного планування і координації. Алгоритми машинного навчання можуть бути використані для оптимізації маршрутів доставки, управління складськими запасами та розподілу ресурсів, що дозволяє зменшити витрати та покращити своєчасність виконання замовлень. Наприклад, системи оптимізації маршруту можуть враховувати дані про трафік, погоду, або навіть зміни в транспортних витратах;

– персоналізація послуг і маркетингу. Один із найважливіших аспектів бізнесу полягає у розумінні потреб клієнтів та їхньої поведінки. За допомогою машинного навчання компанії можуть краще аналізувати поведінку своїх клієнтів, прогнозувати їхні потреби і відповідно налаштовувати свої маркетингові стратегії. Нейронні мережі дозволяють створювати індивідуальні пропозиції, що підвищує рівень задоволеності клієнтів і сприяє утриманню їх у довгостроковій перспективі.

Технології машинного навчання і нейронних мереж не тільки оптимізують існуючі бізнес-процеси, але й створюють нові можливості для бізнесу. Вони дозволяють підприємствам перейти від реактивного управління до проактивного, що означає не лише реагування на зміни, а й передбачення

їх. Це особливо важливо в умовах швидкого розвитку ринкових трендів та зміни поведінки споживачів.

Впровадження ML і нейронних мереж змінює традиційний підхід до управління даними. Наприклад, нейронні мережі можуть аналізувати неструктуровані дані, такі як тексти або зображення, що раніше були недоступними для традиційних методів аналізу. Це дозволяє отримувати нові інсайти і виявляти закономірності, що сприяють прийняттю більш обґрунтованих рішень.

Незважаючи на значний потенціал, впровадження технологій машинного навчання та нейронних мереж також супроводжується певними викликами. Серед них варто виділити високі витрати на розробку і впровадження технологій, складність інтеграції нових рішень в існуючі системи та необхідність у кваліфікованих фахівцях [3]. Додатково, проблема етики та конфіденційності даних набуває все більшого значення, оскільки ці технології часто працюють з великими обсягами персональних даних.

Тому, попри всі переваги, важливо ретельно аналізувати та оцінювати вплив нових технологій на бізнес. Впровадження машинного навчання та нейронних мереж повинно бути ретельно спланованим процесом, що враховує як можливості оптимізації, так і потенційні ризики.

Оптимізація бізнес-процесів за допомогою машинного навчання і нейронних мереж є одним з найбільш перспективних напрямків розвитку сучасного бізнесу. Вона дозволяє підвищити ефективність, гнучкість та конкурентоспроможність організацій, автоматизуючи рутинні операції, покращуючи прогнозування та оптимізуючи управління ресурсами. Однак для успішної інтеграції цих технологій необхідно ретельно оцінювати їхні можливості та виклики, щоб забезпечити максимальний ефект від їхнього використання.

1.2 Класифікація існуючих методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж

Оптимізація бізнес-процесів за допомогою машинного навчання (ML) і нейронних мереж (NN) дозволяє підвищити ефективність бізнесу за рахунок автоматизації, прогнозування і прийняття рішень на основі даних. У даній частині роботи буде проведено детальний аналіз і класифікація методів, що використовуються для оптимізації, а також їх можливі застосування в різних сферах бізнесу. Особлива увага приділяється різним підходам до вирішення завдань, які варіюються від автоматизації рутинних операцій до розробки прогнозних моделей для управління ризиками та маркетингових стратегій.

Машинне навчання (ML) є однією з найперспективніших галузей штучного інтелекту, яка дозволяє створювати моделі для аналізу великих масивів даних і автоматизації процесів. На рисунку 1.1 відображено класифікацію машинного навчання.

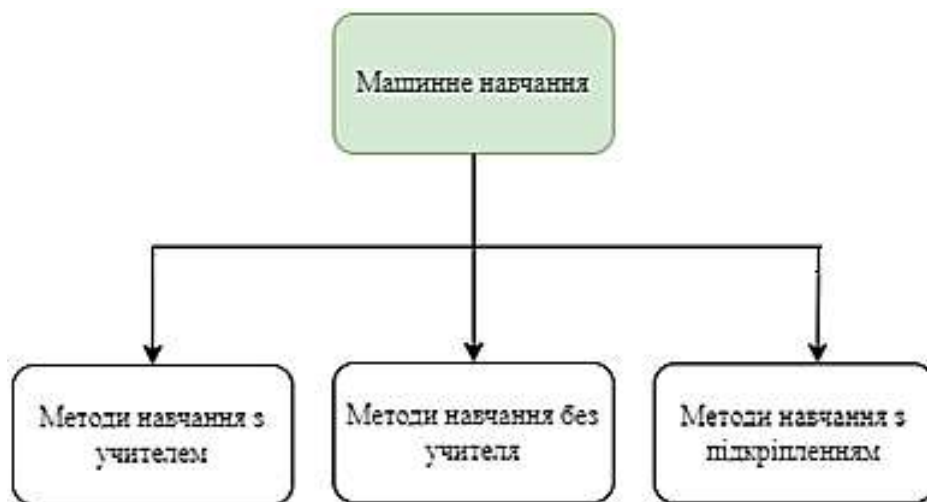


Рисунок 1.1 – Класифікація машинного навчання

Методи навчання з учителем передбачають навчання моделі на основі наявних даних з відомими вхідними та вихідними параметрами. Мета полягає у створенні моделей, здатних передбачити майбутні результати на основі історичних даних.

Застосування таких методів охоплює прогнозування попиту, оптимізацію цінової політики та аналіз поведінки клієнтів. На рисунку 1.2 [4] відображено машинне навчання з вчителем.

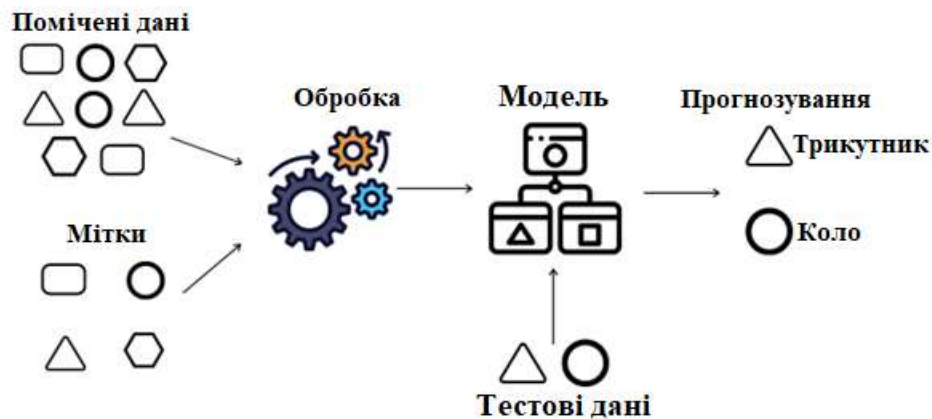


Рисунок 1.2 – Машинне навчання з вчителем

Методи навчання без учителя використовуються для пошуку закономірностей і виявлення прихованих зв'язків у даних без наявності чітко визначених цільових змінних. Основні галузі застосування включають сегментацію клієнтів, виявлення аномалій у бізнес-процесах і кластеризацію даних для покращення маркетингових кампаній. На рисунку 1.3 [4] відображено машинне навчання без вчителя.



Рисунок 1.3 – Машинне навчання без вчителя

Методи навчання з підкріпленням використовуються в задачах, де агент навчається взаємодіяти з середовищем і отримує винагороду за правильні дії.

У бізнес-контексті це може бути використано для оптимізації логістичних процесів, управління запасами та розподілу ресурсів.

На рисунку 1.4 [5] відображено машинне навчання з підкріпленням.



Рисунок 1.4 – Машинне навчання з підкріпленням

У таблиці 1.1 описано класифікацію методів машинного навчання для оптимізації бізнес-процесів.

Таблиця 1.1 – Класифікація методів машинного навчання для оптимізації бізнес-процесів

Класифікація	Опис	Приклади застосування
Навчання з учителем	Модель навчається на основі даних з відомими вихідними параметрами для прогнозування результатів	Прогнозування попиту, оптимізація цін, аналіз клієнтської поведінки
Навчання без учителя	Виявлення закономірностей у даних без явних вихідних змінних	Сегментація клієнтів, виявлення аномалій
Навчання з підкріпленням	Модель навчається шляхом взаємодії зі середовищем і отримання винагороди за правильні дії	Оптимізація логістики, управління запасами, розподіл ресурсів

Нейронні мережі (NN) є більш складними моделями, що здатні обробляти великі обсяги даних і знаходити складні взаємозв'язки, які важко виявити за допомогою традиційних алгоритмів машинного навчання. Основні методи, реалізовані за допомогою нейронних мереж, включають наступні (табл. 1.2):

– мультишарова персептронна мережа (MLP) – цей тип нейронної мережі широко використовується для вирішення задач класифікації і регресії. У бізнесі MLP застосовується для прогнозування продажів, класифікації клієнтів і оцінки ризиків;

– рекурентні нейронні мережі (RNN) – дозволяють обробляти послідовності даних, такі як часові ряди. Вони є корисними для прогнозування поведінки клієнтів, динаміки ринку та управління ланцюгами постачання;

– конволюційні нейронні мережі (CNN). Хоча CNN здебільшого використовуються в обробці зображень, вони можуть бути застосовані до аналізу даних у сфері фінансів, наприклад, для виявлення шахрайства або аномалій у великих масивах даних [6].

Таблиця 1.2 – Методи нейронних мереж для оптимізації бізнес-процесів

Тип нейронної мережі	Опис	Приклади застосування
Мультишарова персептронна мережа (MLP)	Базова нейронна мережа для вирішення задач класифікації та регресії	Прогнозування продажів, класифікація клієнтів, оцінка ризиків
Рекурентна нейронна мережа (RNN)	Обробка послідовностей даних і часових рядів	Прогнозування поведінки клієнтів, управління ланцюгами постачання

Продовження таблиці 1.2

Конволюційна нейронна мережа (CNN)	Аналіз даних, що мають просторову структуру (наприклад, зображення або великі фінансові дані)	Виявлення шахрайства, аналіз фінансових аномалій
------------------------------------	---	--

Гібридні методи поєднують підходи машинного навчання та нейронних мереж для досягнення максимального ефекту в оптимізації бізнес-процесів. Зокрема, такі методи можуть поєднувати кластеризацію з подальшою класифікацією або використовувати навчання з підкріпленням у поєднанні з рекурентними нейронними мережами для створення моделей, що оптимізують процеси у динамічному середовищі.

Гібридні нейронні мережі використовують комбінації RNN і CNN для більш точного прогнозування та аналізу. Наприклад, такі мережі можуть бути корисними у маркетингових кампаніях, коли необхідно аналізувати як текстові, так і часові дані. Поєднання традиційних методів машинного навчання з оптимізаційними алгоритмами може призвести до створення моделей, що не тільки передбачають попит, але й автоматично коригують логістичні плани. Глибоке навчання є одним з різновидів нейронних мереж, що дозволяє вирішувати задачі високої складності. Гібридні моделі глибокого навчання можуть бути використані для обробки великих обсягів даних у реальному часі, зокрема у сфері фінансових технологій та електронної комерції.

Методи оптимізації на основі ML і NN знаходять своє застосування в різних галузях бізнесу (табл. 1.3) [7]:

– фінансовий сектор. Нейронні мережі активно використовуються для виявлення шахрайства, прогнозування фінансових показників і управління ризиками. Вони дозволяють швидко обробляти великі обсяги даних і знаходити приховані закономірності;

– логістика і виробництво. Алгоритми машинного навчання застосовуються для оптимізації маршрутів доставки, управління запасами та планування виробничих потужностей;

– маркетинг і продажі. Використання методів класифікації і кластеризації дозволяє підвищити ефективність маркетингових кампаній за рахунок персоналізації і точного прогнозування поведінки клієнтів.

Таблиця 1.3 – Гібридні методи оптимізації бізнес-процесів

Комбінація методів	Опис	Приклади застосування
RNN + CNN	Використання рекурентних і конволюційних нейронних мереж для комплексного аналізу	Маркетингові кампанії, аналіз текстових і часових даних
Машинне навчання + оптимізаційні алгоритми	Поєднання прогнозування з автоматичною корекцією бізнес-процесів	Управління ланцюгами постачання, планування ресурсів
Гібридні моделі глибокого навчання	Застосування глибоких нейронних мереж для обробки даних у реальному часі	Фінансові технології, електронна комерція

Методи оптимізації бізнес-процесів, реалізовані засобами машинного навчання та нейронних мереж, охоплюють широкий спектр підходів, від традиційних моделей навчання до сучасних гібридних систем.

Впровадження таких рішень дозволяє значно підвищити ефективність управління ресурсами, покращити планування та підвищити рівень обслуговування клієнтів.

1.3 Аналіз сучасних методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж

Сучасні методи оптимізації бізнес-процесів за допомогою машинного навчання (ML) і нейронних мереж (NN) відкривають нові можливості для підвищення ефективності, автоматизації та точності прийняття рішень в різних сферах бізнесу. Машинне навчання та нейронні мережі пропонують інструменти, які дозволяють підприємствам використовувати дані для оптимізації процесів, розробки прогнозних моделей, автоматизації рутинних завдань та управління ризиками. З розвитком інформаційних технологій та зростанням обсягів даних, які компанії повинні обробляти щоденно, машинне навчання (ML) та нейронні мережі (NN) стали ключовими інструментами для оптимізації бізнес-процесів. Завдяки потужним алгоритмам та методам, ці технології дозволяють підвищити ефективність, автоматизувати рутинні операції, покращити прийняття рішень і мінімізувати витрати.

Існують багато популярних сучасних методів оптимізації бізнес-процесів. Наприклад, Transfer Learning передбачає використання моделі, попередньо навченої на великому наборі даних, для вирішення схожих завдань у нових доменах з обмеженими даними. Це дозволяє скоротити час і ресурси, необхідні для навчання нових моделей. Fine-Tuning – це метод доопрацювання моделей, попередньо навчених за допомогою Transfer Learning, шляхом налаштування на нових специфічних даних. Він дозволяє моделі адаптуватися до нових умов і підвищити точність на нових завданнях.

Ensemble Methods поєднують кілька моделей для підвищення точності прогнозів. Найпоширеніші підходи включають Bagging, Boosting та Stacking.

Використання ансамблів дозволяє зменшити ймовірність помилок та підвищити надійність моделей [8]. Hyperparameter Optimization дозволяє налаштувати параметри моделей, такі як швидкість навчання або кількість шарів у нейронній мережі, щоб досягти максимальної ефективності моделі. Це значно впливає на продуктивність і точність прогнозів. Knowledge

Distillation полягає в передачі знань від великої та потужної моделі до меншої, легшої моделі без втрати якості. Це знижує обчислювальні витрати і забезпечує ефективне використання ресурсів у виробничих умовах.

Neural Architecture Search автоматизує пошук оптимальної архітектури нейронної мережі для конкретних завдань, що дозволяє підвищити продуктивність і точність моделей, а також скоротити час на їхню розробку. Self-Supervised Learning навчає моделі на даних, автоматично створюючи мітки для нових зразків. Це допомагає значно скоротити потребу в ручному маркуванні даних і підвищує ефективність навчання.

Meta-Learning дозволяє моделі швидко адаптуватися до нових завдань із мінімальною кількістю даних. Це важливо в умовах динамічного середовища, де постійно змінюються умови [9].

Few-Shot Learning забезпечує можливість навчати модель на невеликих наборах даних, що є корисним у випадках, коли дані важкодоступні або обмежені. Reinforcement Learning базується на винагороді за правильні дії. Цей метод широко використовується для оптимізації процесів, де важливо враховувати динамічні взаємодії, наприклад, у логістиці або управлінні запасами.

Multi-Task Learning дозволяє моделі навчатися одночасно на кількох взаємопов'язаних завданнях, підвищуючи її загальну ефективність і скорочуючи час на навчання окремих моделей для кожного завдання. Contrastive Learning використовується для розрізнення схожих і відмінних прикладів. Це особливо корисно для обробки зображень та природної мови, де важливо чітко ідентифікувати відмінності між об'єктами. Federated Learning дозволяє навчати моделі на даних, що знаходяться на різних пристроях або серверах, без необхідності передавати на центральний сервер. Це забезпечує конфіденційність даних і є важливим у фінансових або медичних сферах [10].

Active Learning визначає приклади, на яких модель має найменшу впевненість, і використовує ці приклади для подальшого навчання. Це

допомагає більш ефективно використовувати дані для навчання моделей. Adversarial Learning готує моделі до атак шляхом тренування на прикладах, створених для обману моделі. Це дозволяє підвищити стійкість до змін або атак на вхідні дані. Sparse Learning оптимізує обчислювальні витрати шляхом зменшення кількості активованих нейронів під час навчання, що знижує ресурси без втрати ефективності. Нижче наведено таблицю 1.4, в якій здійснено порівняння сучасних методів.

Таблиця 1.4 – Порівняння методів

Метод	Призначення	Переваги	Використання
Transfer Learning	Перенос знань на нові завдання	Знижує потребу в даних	Рекомендаційні системи, аналіз клієнтів
Fine-Tuning	Доопрацювання моделей на нових даних	Підвищує точність для специфічних задач	Прогнозування, сегментація ринку
Ensemble Methods	Комбінація кількох моделей	Підвищення надійності та точності	Оцінка ризиків, класифікація
Hyperparameter Optimization	Оптимізація параметрів моделі	Покращує продуктивність моделі	Прогнозування, обробка великих даних
Knowledge Distillation	Зменшення моделі без втрати якості	Економія обчислювальних ресурсів	Мобільні застосунки, IoT

Продовження таблиці 1.4

Neural Architecture Search	Пошук оптимальної архітектури	Автоматизує проєктування нейронних мереж	Медичні зображення, прогнозування ринку
Self-Supervised Learning	Автоматичне навчання на невеликих даних	Скорочення ручного маркування даних	Обробка текстів, природної мови
Meta-Learning	Адаптація до нових завдань	Швидка адаптація до нових умов	Динамічні ринки, нові дані
Few-Shot Learning	Навчання на малих наборах даних	Ефективно працює за обмежених даних	Наукові дослідження, специфічні задачі
Reinforcement Learning	Оптимізація на основі винагороди	Підвищує адаптивність до динамічних середовищ	Управління запасами, логістика
Multi-Task Learning	Навчання на кількох завданнях одночасно	Економія ресурсів, підвищення продуктивності	Аналіз даних, прогностичні системи
Contrastive Learning	Відмінність схожих і відмінних прикладів	Підвищує точність розпізнавання	Обробка зображень, природної мови
Federated Learning	Навчання на розподілених даних	Забезпечує конфіденційність даних	Фінансові та медичні дані

Продовження таблиці 1.4

Active Learning	Використання найслабших прикладів для навчання	Оптимізує використання даних	Аналітика, навчання на вибіркових даних
Adversarial Learning	Підвищення стійкості до атак	Захист від атак на моделі	Кібербезпека, захист даних
Sparse Learning	Зменшення активності нейронів	Знижує обчислювальні витрати	Оптимізація великих моделей

Сучасні методи машинного навчання та нейронних мереж пропонують широкі можливості для оптимізації бізнес-процесів, що дозволяє підвищити ефективність роботи підприємств, знизити витрати та приймати кращі управлінські рішення. Різноманітність методів дозволяє адаптувати рішення під бізнес-завдання, що сприяє гнучкості та конкурентоспроможності в умовах постійно змінюваного ринкового середовища.

У сучасному бізнесі важливо не тільки збирати та аналізувати дані, але й робити це максимально ефективно, щоб оптимізувати процеси, підвищувати точність прогнозів і зменшувати витрати. Для цього існує багато методів машинного навчання і нейронних мереж, кожен з яких має свої переваги. Серед них найбільший інтерес становлять ті методи, які забезпечують максимальну гнучкість, точність та ефективність у вирішенні бізнес-завдань.

Було обрано 6 основних методів, а саме:

– Transfer Learning – дозволяє швидко адаптувати моделі для нових завдань, зменшуючи потребу в великих обсягах даних і скорочуючи час навчання;

- Fine-Tuning – допомагає налаштувати модель для конкретних завдань, підвищуючи її точність і ефективність для нових умов;
- Ensemble Methods – поєднують кілька моделей, що підвищує точність і надійність прогнозів за рахунок зменшення похибок;
- Hyperparameter Optimization – оптимізує параметри моделей, що дозволяє досягти максимального рівня продуктивності та точності;
- Knowledge Distillation – забезпечує ефективність і економію ресурсів, дозволяючи використовувати легші моделі без втрати якості;
- Neural Architecture Search (NAS) – автоматизує процес створення оптимальної архітектури нейронних мереж, скорочуючи час на розробку, підвищуючи продуктивність.

Ці методи були обрані для досягнення найбільшого ефекту в оптимізації бізнес-процесів, оскільки вони надають гнучкі рішення, дозволяють економити час і ресурси, а також підвищують точність моделей у різних бізнес-завданнях.

1.4 Аналіз літературних джерел щодо апробації результатів застосування методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж

На сьогоднішній день використання машинного навчання (ML) та нейронних мереж (NN) стає ключовим інструментом для оптимізації бізнес-процесів у багатьох галузях економіки. Машинне навчання дозволяє створювати адаптивні системи, які здатні ефективно аналізувати великі обсяги даних, автоматизувати процеси та підвищувати їхню ефективність. Важливим аспектом є апробація результатів застосування таких методів, оскільки це забезпечує перевірку їхньої доцільності в умовах реального бізнесу. Метою цього розділу є аналіз наукових публікацій та досліджень, що

стосуються використання ML та NN для оптимізації бізнес-процесів, а також визначення результатів їх апробації в різних секторах економіки.

Машинне навчання, яке є підгалуззю штучного інтелекту, використовує алгоритми для виявлення патернів у великих масивах даних і може приймати рішення або прогнози на основі попередньо отриманих знань [11]. Нейронні мережі, в свою чергу, є специфічним видом ML, що імітують роботу людського мозку. Нейронні мережі складаються з численних шарів штучних нейронів, які взаємодіють між собою та обробляють вхідні дані з метою знаходження найбільш ефективних рішень.

У бізнесі ML та NN активно використовуються для автоматизації складних процесів, таких як прогнозування попиту, оптимізація ланцюгів постачання, управління ризиками, персоналізація маркетингових стратегій, аналіз клієнтської поведінки тощо. Однак, для того щоб ці технології були ефективними, важливою є їх практична апробація в реальних умовах.

Апробація результатів є невід'ємним етапом процесу впровадження нових технологій у бізнес. Це дозволяє визначити, наскільки ефективними є нові методи та алгоритми в умовах конкретної компанії або галузі. У літературі наводяться численні приклади успішної апробації технологій ML та NN у бізнес-процесах. Так, дослідження у сфері роздрібної торгівлі показали, що використання ML для прогнозування попиту дозволяє скоротити надлишкові запаси на 30%-40% та зменшити втрати через надмірне замовлення товарів.

Інше дослідження, проведене у сфері фінансових послуг, демонструє, як алгоритми нейронних мереж можуть підвищити точність прогнозування кредитних ризиків на 25% порівняно з традиційними методами, що базуються на регресіях та статистичних моделях. Це дозволяє фінансовим інститутам знижувати ризики неповернення кредитів і підвищувати рентабельність бізнесу.

Застосування ML для автоматизації рутинних завдань у сфері управління ланцюгами постачання також отримало позитивну апробацію.

Наприклад, у дослідженні [12] описано, як застосування технологій прогнозування та оптимізації на основі нейронних мереж допомогло міжнародній логістичній компанії скоротити час доставки на 15% та зменшити витрати на логістику на 12%.

Нейронні мережі [13], завдяки своїй здатності навчатися та адаптуватися, також активно використовуються в автоматизації складних бізнес-процесів. У наукових роботах [14-17] відзначається, що саме завдяки нейронним мережам досягаються значні успіхи в таких галузях, як фінанси, охорона здоров'я, промисловість та телекомунікації. Зокрема, у фінансовому секторі нейронні мережі використовуються для прогнозування фондового ринку та управління ризиками [18]. Як свідчить дослідження [19] застосування багатопових нейронних мереж у сфері інвестицій дозволяє підвищити точність прогнозів щодо динаміки ринку на 20%, що забезпечує кращу віддачу для інвесторів. У промислових секторах нейронні мережі використовуються для підвищення ефективності виробництва [20]. Наприклад, компанія Siemens у 2019 році впровадила систему на базі глибинного навчання для оптимізації процесів технічного обслуговування обладнання. Результати апробації показали скорочення часу на проведення ремонтних робіт на 30% та зменшення кількості непередбачених збоїв обладнання [21].

Аналіз літературних джерел [12-21] підтверджує, що застосування методів машинного навчання та нейронних мереж у бізнес-процесах має значний потенціал для оптимізації операцій та підвищення ефективності бізнесу. Успішні приклади апробації цих методів вказують на можливість їх використання в різних галузях, включаючи фінансовий сектор, логістику, охорону здоров'я та виробництво. Впровадження таких технологій дозволяє скоротити витрати, підвищити точність прогнозів та забезпечити гнучкість у прийнятті рішень, що робить їх невід'ємною складовою сучасного бізнесу.

1.5 Постановка задачі дослідження

Основною задачею дослідження є розробка підходів для ефективної оптимізації бізнес-процесів за допомогою методів машинного навчання та нейронних мереж, що дозволить підвищити продуктивність, зменшити витрати та покращити якість управлінських рішень.

Об'єктом дослідження є бізнес-процеси різних організацій, що працюють у різних галузях.

Метою дослідження є розробка методів оптимізації бізнес-процесів, що базуються на алгоритмах машинного навчання та нейронних мереж, з метою підвищення ефективності та автоматизації рутинних завдань.

Для досягнення цієї мети необхідно виконати такі конкретні завдання:

- проаналізувати існуючі методи оптимізації бізнес-процесів, реалізовані за допомогою машинного навчання та нейронних мереж;
- дослідити сучасні підходи до автоматизації бізнес-процесів;
- виявити переваги та недоліки існуючих методів з точки зору ефективності;
- класифікувати методи оптимізації бізнес-процесів;
- провести огляд і класифікацію методів машинного навчання та нейронних мереж, що використовуються для бізнес-процесів;
- визначити критерії вибору методів для конкретних бізнес-завдань;
- дослідити сучасні методи оптимізації бізнес-процесів;
- оцінити ефективність методів Transfer Learning, Fine-Tuning, Ensemble Methods, Hyperparameter Optimization, Knowledge Distillation та Neural Architecture Search;
- вивчити їх вплив на продуктивність і якість управління процесами;
- реалізувати обрані методи на практиці;
- обрати інструментальні засоби для програмної реалізації вибраних методів;

- реалізувати методи оптимізації бізнес-процесів на прикладі обраної предметної області;
- провести порівняльний аналіз методів;
- оцінити результативність реалізованих методів на практиці;
- порівняти результати досліджених методів і визначити перспективи подальшого використання.

Дані задачі спрямовані на комплексне дослідження та апробацію сучасних методів оптимізації бізнес-процесів із використанням машинного навчання та нейронних мереж, що допоможе визначити найбільш ефективні підходи для різних бізнес-сценаріїв.

2 ОСОБЛИВОСТІ ВИБРАНИХ МЕТОДІВ ОПТИМІЗАЦІЇ БІЗНЕС ПРОЦЕСІВ, РЕАЛІЗОВАНИХ ЗАСОБАМИ МАШИННОГО НАВЧАННЯ ТА НЕЙРОННИХ МЕРЕЖ

2.1 Метод Transfer Learning

Transfer Learning (трансферне навчання) є одним із ключових методів сучасного машинного навчання, який дозволяє передавати знання, набуті в одній області або на одному наборі даних, для вирішення задач у зовсім іншій області або на нових даних. Цей підхід є надзвичайно корисним у ситуаціях, коли обсяги даних для конкретної бізнес-задачі обмежені, але доступні попередньо натреновані моделі на великих наборах даних.

Метод Transfer Learning використовується для оптимізації бізнес-процесів у сферах таких, як прогнозування попиту, аналіз поведінки споживачів, автоматизація маркетингу, управління запасами та багатьох інших. Основна перевага цього методу полягає у зменшенні витрат часу та обчислювальних ресурсів для навчання моделей, оскільки він використовує вже існуючі знання з інших джерел (рис. 2.1 [22]).



Рисунок 2.1 – Принцип роботи Transfer Learning

Основна ідея трансферного навчання полягає у використанні попередньо натренованої моделі, зазвичай глибокої нейронної мережі, яка вже має вивчені патерни або ознаки, релевантні до завдання, для якого вона була спочатку створена.

Ця модель потім донавчається (fine-tuning) на новому наборі даних, що дозволяє їй адаптуватися до специфічних бізнес-потреб (табл. 2.1).

Таблиця 2.1 – Приклад архітектури Transfer Learning на основі попередньо натренованої моделі ResNet-50

Етап	Опис
Pre-training	Використовується модель ResNet-50, навчена на ImageNet для розпізнавання загальних об'єктів на зображеннях
Freezing Layers	Заморожуються всі шари до 4-го блоку нейронної мережі, що відповідають за вивчення базових ознак, таких як краї, текстури, форми
Fine-tuning	Донавчаються останні шари (5-й блок і класифікаційний шар) на спеціалізованих бізнес-даних (наприклад, даних з виробництва для дефектної продукції)
Output Layer Adaptation	Замінюється останній класифікаційний шар для відповідності кількості класів у бізнес-задачі

Зазвичай процес виглядає наступним чином:

– попереднє навчання (Pre-training): модель тренується на великому загальному наборі даних (наприклад, ImageNet для задач комп'ютерного зору або BERT для задач обробки тексту);

– заморожування шарів (Freezing layers): для збереження вже набутих знань, часто заморожують початкові шари нейронної мережі, які відповідальні за загальні ознаки;

– донавчання (Fine-tuning): останні шари моделі адаптуються до специфічного бізнес-завдання через додаткове навчання на нових, менш об'ємних даних.

Трансферне навчання зменшує кількість даних, необхідних для навчання моделі, що особливо важливо в бізнесі, де дані можуть бути

обмеженими або їх збирання є дорогим. Оскільки більша частина моделі вже натренована, процес її адаптації до конкретної задачі займає менше часу, ніж навчання моделі з нуля. Оскільки модель вже має добре збалансовані параметри з попереднього навчання, ризик перенавчання на обмежених бізнес-даних знижується (табл. 2.2).

Таблиця 2.2 – Застосування Transfer Learning для прогнозування попиту

Задача	Метод	Переваги
Прогнозування попиту на товари	Використання попередньо натренованої моделі на основі часових рядів	Швидка адаптація моделі до бізнес-даних, зниження витрат на збір додаткових даних
Аналіз поведінки клієнтів	Модель для класифікації клієнтів на основі попередньо натренованої нейронної мережі	Підвищення точності персоналізації маркетингових кампаній
Оптимізація управління запасами	Модель для прогнозування зносостійкості продуктів на основі їх характеристик	Оптимізація витрат на складські приміщення

Розглянемо використання трансферного навчання у прогнозуванні попиту на товари. Для цієї задачі можна взяти попередньо натреновану модель, яка аналізує час ряду, та адаптувати її під специфічні бізнес-дані магазину, наприклад, обсяги продажу за останній рік. Це дозволяє швидко створити модель, яка здатна точно прогнозувати потребу в товарі, базуючись на історичних даних та трендах.

Загалом архітектура нейронної мережі для Transfer Learning зазвичай складається з кількох базових блоків, таких як:

– шари для виділення ознак (Feature extraction layers) – ці шари відповідають за виявлення базових патернів у даних, таких як контури та текстури в задачах комп'ютерного зору або семантичні ознаки в текстових задачах;

– шари адаптації (Adaptation layers) вони додаються до кінця нейронної мережі для адаптації до конкретного завдання. Це можуть бути, наприклад, шари класифікації або регресії, в залежності від типу задачі;

– оптимізація параметрів (Optimization layers – ці шари контролюють процес навчання моделі та її донавчання на нових даних. Вони відповідають за пошук оптимальних значень ваг нейронів, щоб мінімізувати втрати.

Метод Transfer Learning є потужним інструментом для оптимізації бізнес-процесів. Він дозволяє швидко і з мінімальними витратами адаптувати потужні нейронні мережі до специфічних бізнес-задач. Його застосування значно зменшує витрати на збір даних та розробку нових моделей, забезпечуючи при цьому високу точність і стабільність результатів. Transfer Learning знаходить застосування в різних галузях бізнесу, від роздрібною торгівлі до виробничих процесів, і має великий потенціал для майбутнього розвитку бізнес-аналітики та автоматизації [23].

2.2 Метод Fine-Tuning

Fine-Tuning (донавчання) – це метод, який є частиною стратегії Transfer Learning і дозволяє оптимізувати модель під конкретні бізнес-задачі шляхом подальшого навчання вже попередньо натренованої нейронної мережі на нових даних. Мета Fine-Tuning полягає в адаптації попередньо вивчених ознак моделі до конкретного набору даних, що значно підвищує точність і продуктивність моделі для специфічних завдань. Fine-Tuning активно використовується у випадках, коли в бізнесі немає можливості зібрати великі обсяги даних або коли швидкість і ефективність є критичними.

Наприклад, у задачах розпізнавання зображень, обробки природної мови (NLP) або навіть у фінансовому прогнозуванні Fine-Tuning дозволяє адаптувати модель до нових умов з мінімальними витратами на навчання.

Основний принцип Fine-Tuning полягає у наступному: спочатку використовується вже натренована модель, яка була навчена на великому наборі загальних даних, таких як ImageNet для зображень або BERT для текстових задач. Потім ця модель адаптується під новий набір даних або нову задачу. У процесі Fine-Tuning основна частина моделі залишається незмінною, але деякі шари (часто кінцеві шари нейронної мережі) донавчаються для адаптації до нових специфікацій (рис. 2.2 [24]).

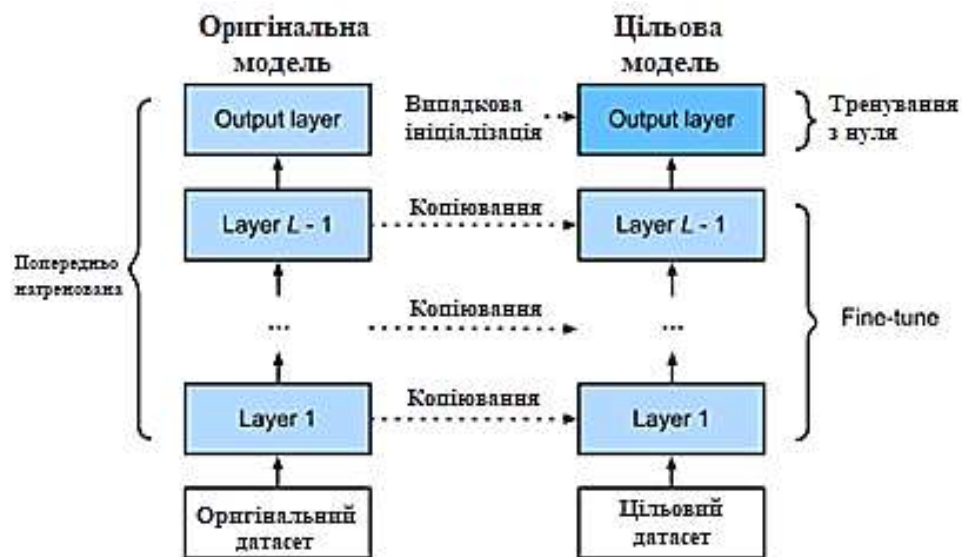


Рисунок 2.2 – Принцип роботи Fine-Tuning

Основні етапи Fine-Tuning:

- заморожування початкових шарів (Freezing initial layers): оскільки перші шари моделі відповідають за загальні ознаки, вони зазвичай заморожуються, щоб не змінювати вже натреновані параметри;

- розблокування останніх шарів (Unfreezing the last layers): останні шари моделі розблоковуються для донавчання на нових даних. Це дозволяє моделі адаптуватися до конкретної задачі, але при цьому зберігати загальні знання;

– перенавчання кінцевих шарів (Retraining final layers): модель навчиться розпізнавати нові патерни, характерні для специфічних даних бізнесу, шляхом модифікації лише кількох останніх шарів;

– оптимізація параметрів (Optimization): для подальшої адаптації використовується оптимізація гіперпараметрів, таких як швидкість навчання, що дозволяє досягти кращої точності.

Архітектура Fine-Tuning зазвичай базується на попередньо натренованих глибоких нейронних мережах, таких як VGG, ResNet, BERT, GPT тощо [25] (табл. 2.3).

Таблиця 2.3 – Архітектура Fine-Tuning на основі моделі ResNet-50

Компонент	Функція	Застосування Fine-Tuning
Вхідні дані	Приймають дані у вигляді зображення або тексту	Ніяких змін не відбувається
Шари виділення ознак	Набір шарів, які відповідають за виділення базових ознак	Заморожуються для збереження загальних знань
Адаптаційні шари	Шари, що модифікуються для конкретної бізнес-задачі	Розблоковуються для донавчання на нових даних
Класифікаційний шар	Остаточний шар для прийняття рішення	Заміна для адаптації під кількість класів у бізнес-задачі або інші вимоги

Архітектура таких мереж включає кілька основних компонентів:

– вхідні шари (Input layers) через які проходять дані на початку обчислювального процесу. Наприклад, це можуть бути зображення розміром 224×224 пікселів або текст у вигляді послідовності слів;

– шари виділення ознак (Feature extraction layers) відповідають за виділення основних ознак із вхідних даних. Для задач комп’ютерного зору ці ознаки можуть включати контури, форми, текстури, а для задач NLP – семантичні зв’язки між словами;

– шари адаптації (Adaptation layers) модифікуються під конкретну задачу. У процесі Fine-Tuning саме ці шари донавчаються на нових даних;

– класифікаційні або регресійні шари (Classification/Regression layers), які надають кінцевий результат моделі. У задачах класифікації це зазвичай softmax або sigmoid, тоді як у задачах регресії – звичайний вихід із передбаченням.

Важливим аспектом Fine-Tuning є оптимізація процесу навчання, яка включає кілька ключових моментів:

– швидкість навчання (Learning rate): швидкість навчання під час Fine-Tuning зазвичай нижча, ніж під час початкового навчання, щоб уникнути перезапису попередньо вивчених ознак;

– розмір партії (Batch size): підбір оптимального розміру партії є критичним для забезпечення стабільного та швидкого навчання моделі;

– раннє зупинення (Early stopping): використовується для запобігання перенавчанню, коли модель досягає плато у своїх показниках.

Fine-Tuning дозволяє адаптувати моделі до конкретних бізнес-сценаріїв швидко і ефективно. Наприклад, модель, натренована для аналізу зображень загальних об’єктів, може бути донавчена для розпізнавання дефектів у виробничих процесах, а модель для аналізу тексту може бути адаптована для роботи зі специфічною термінологією в певній галузі (табл. 2.4).

Fine-Tuning є одним із найефективніших підходів для адаптації моделей машинного навчання до специфічних бізнес-задач. Цей метод дозволяє значно зменшити обсяги обчислень і час на навчання, забезпечуючи при цьому високу точність і продуктивність моделей. Fine-Tuning активно використовується у багатьох галузях, таких як розпізнавання зображень, обробка тексту, прогнозування та інші бізнес-застосування.

Таблиця 2.4 – Застосування Fine-Tuning для бізнес-задач

Задача	Метод Fine-Tuning	Переваги
Розпізнавання дефектів на виробництві	Модель, натренована на ImageNet, донавчається для виявлення дефектів на виробничих лініях	Зменшення витрат на збір даних і висока точність
Аналіз тональності текстів в соціальних мережах	Використання моделі BERT для аналізу специфічних слів і контекстів у конкретних галузях	Підвищення точності моделі в конкретній галузі завдяки додатковому навчанню
Прогнозування продажів	Адаптація моделі, натренованої на часових рядах, для прогнозування конкретних бізнес-показників	Оптимізація процесів прийняття рішень в умовах обмежених даних

Завдяки цьому методу компанії можуть швидко і ефективно адаптувати новітні досягнення машинного навчання під свої унікальні потреби.

2.3 Метод Ensemble Methods

Ensemble Methods (ансамблеві методи) є одним із ключових напрямків у сучасному машинному навчанні, який базується на комбінуванні кількох моделей для підвищення загальної точності та стабільності результатів. Принцип роботи ансамблевих методів полягає в тому, що замість використання однієї моделі, яка може мати обмеження і недоліки, застосовується набір моделей, результати об'єднуються для прийняття рішення. Зменшує похибки, підвищує продуктивність (рис. 2.3 [26]).

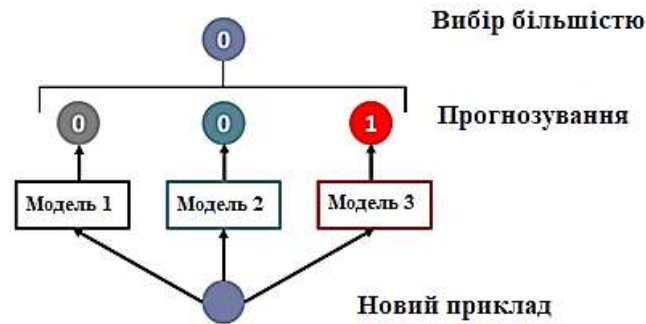


Рисунок 2.3 – Ансамблеві методи

Ансамблеві методи широко застосовуються у різних галузях, від фінансових прогнозів до медичних досліджень, оскільки вони дозволяють підвищити точність прогнозів і зробити систему більш стійкою до помилок окремих моделей. Серед популярних ансамблевих методів варто відзначити Bagging, Boosting і Stacking, кожен із яких має особливості та застосування.

Ансамблеві методи базуються на ідеї, що об'єднання кількох слабких або навіть сильних моделей може призвести до значного покращення загальних результатів. Основний принцип полягає у тому, що кожна модель у ансамблі вирішує задачу трохи по-іншому, що дозволяє компенсувати недоліки кожної окремої моделі.

Існує кілька ключових підходів до побудови ансамблевих методів.

Bagging (Bootstrap Aggregating) – у цьому підході створюється кілька моделей за допомогою різних підмножин даних, отриманих шляхом випадкової вибірки з вихідного набору даних. Кожна модель навчається на своїй підмножині, і результати всіх моделей об'єднуються (сумуються або усереднюються) для прийняття остаточного рішення. Найвідоміший приклад застосування Bagging – це метод Random Forest.

Boosting – на відміну від Bagging, Boosting використовує ітеративний підхід, де кожна нова модель у ансамблі навчається на помилках попередніх моделей. Boosting намагається виправити помилки, зроблені раніше, шляхом надання більшої ваги зразкам, які були класифіковані неправильно. Приклади Boosting методів включають AdaBoost, Gradient Boosting та XGBoost.

У Stacking використовується кілька різних моделей (можуть бути різні алгоритми), а результати цих моделей передаються на вхід новій моделі (так званій «мета-моделі»), яка приймає кінцеве рішення. Цей підхід дозволяє об'єднувати переваги кількох різних алгоритмів.

Основною перевагою ансамблевих методів є можливість значного підвищення точності прогнозів (табл. 2.5). Це досягається завдяки тому, що кожна окрема модель може вносити свої унікальні знання або «способи» прийняття рішень, що підвищує загальну точність кінцевого результату.

Таблиця 2.5 – Ключові ансамблеві методи

Метод	Принцип роботи	Приклад алгоритму
Bagging	Випадкова вибірка підмножин даних для навчання різних моделей	Random Forest
Boosting	Ітеративне навчання моделей з акцентом на помилках попередніх моделей	AdaBoost, XGBoost
Stacking	Об'єднання результатів кількох моделей з використанням мета-моделі	Мета-регресія

Одним із проблемних аспектів деяких моделей машинного навчання є висока варіативність (variance). Ансамблеві методи, такі як Bagging, дозволяють знизити варіативність шляхом об'єднання кількох моделей, що навчалися на різних підмножинах даних. Таким чином, кінцевий результат є більш стабільним і менш чутливим до змін у вихідних даних.

Ансамблеві методи мають кращу стійкість до перенавчання (overfitting). Це особливо стосується методів Bagging, які через випадкову вибірку з підмножин даних створюють моделі, які менш схильні до перенавчання на шуми. Ensemble Methods можна використовувати з різними типами моделей, включаючи як лінійні алгоритми, так і складні глибокі

нейронні мережі. Наприклад, у Stacking можна комбінувати різні алгоритми, такі як лінійна регресія, дерева рішень та нейронні мережі.

Одна з основних проблем ансамблевих методів полягає у збільшенні складності моделі. Поєднання кількох моделей потребує більше обчислювальних ресурсів і часу на навчання. Особливо це стосується таких методів, як Boosting [27], де кожна нова модель навчається на помилках попередніх моделей. Оскільки ансамбль може складатися з багатьох моделей, результат кінцевого прогнозу може бути важко інтерпретувати. У порівнянні з окремими моделями, такими як лінійна регресія або дерево рішень, ансамблеві методи є «чорними скриньками», тобто важче зрозуміти, як саме модель прийшла до рішення (табл. 2.6).

Таблиця 2.6 – Переваги і недоліки ансамблевих методів

Переваги	Недоліки
Підвищена точність прогнозування	Збільшення складності моделі
Стійкість до перенавчання	Труднощі з інтерпретацією
Зменшення варіативності	Збільшення часу навчання
Гнучкість у використанні різних моделей	Потреба в значних обчислювальних ресурсах

Ансамблеві методи знаходять широке застосування у різних галузях, де точність і стабільність прогнозування є критичними. У фінансовому секторі, де прогнозування ринкових тенденцій і ризиків має велике значення, ансамблеві методи дозволяють підвищити точність прогнозування. Наприклад, Random Forest використовується для класифікації клієнтів на основі їх фінансових звичок, а Boosting використовується для прогнозування кредитних ризиків. В галузі медицини ансамблеві методи застосовуються для діагностики захворювань та прогнозування результатів лікування.

Наприклад, алгоритми Boosting допомагають виявляти ризики розвитку хронічних захворювань на основі аналізу медичних записів пацієнтів.

2.4 Метод Hyperparameter Optimization

Гіперпараметри – це налаштування моделі, які не вивчаються автоматично під час навчання, на відміну від параметрів моделі, які модель коригує сама. Гіперпараметри встановлюються до початку навчання і значно впливають на точність, продуктивність та час роботи моделі. Оптимізація гіперпараметрів – це процес пошуку найкращих значень цих гіперпараметрів для підвищення продуктивності моделі машинного навчання (рис. 2.4 [28]).

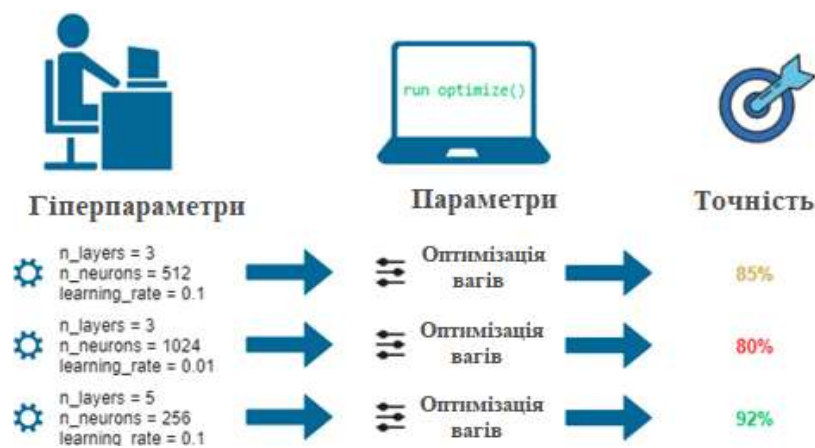


Рисунок 2.4 – Метод оптимізації гіперпараметрів

Гіперпараметри варіюються в залежності від типу алгоритму машинного навчання. Наприклад, у випадку нейронних мереж, вони можуть включати:

- кількість нейронів у шарах;
- швидкість навчання (learning rate);
- тип функції активації;
- розмір мініпакету (batch size);
- кількість епох.

Мета гіперпараметричної оптимізації – знайти таку комбінацію гіперпараметрів, яка мінімізує функцію втрат або максимізує продуктивність моделі на даних (рис. 2.5 [29]).

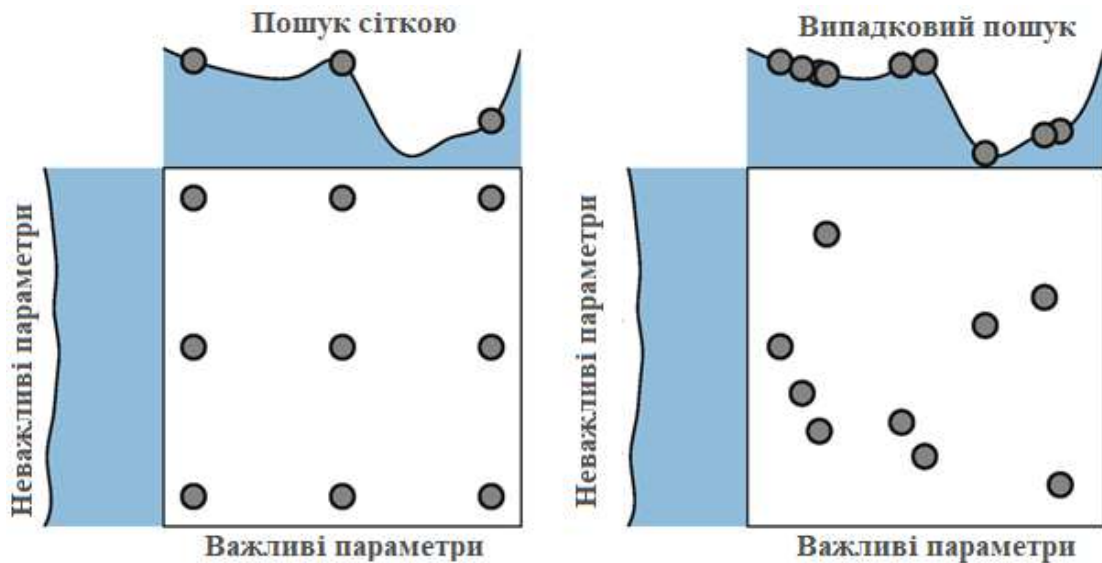


Рисунок 2.5 – Методи пошук сіткою та випадковий пошук

Grid Search – це найпростіший і найчастіше використовуваний метод. Він передбачає перебір всіх можливих комбінацій гіперпараметрів за визначеними інтервалами. Після цього обчислюється продуктивність кожної комбінації, і вибирається та, яка дає найкращий результат.

Random Search – цей метод є вдосконаленим варіантом Grid Search. Замість перебору всіх можливих комбінацій, випадково вибираються певні комбінації гіперпараметрів і обчислюється їх продуктивність. Цей підхід часто дає гарні результати за значно менших витрат часу.

Bayesian Optimization у цьому методі використовується модель для передбачення найкращих гіперпараметрів на основі попередніх результатів. Замість того, щоб безпосередньо перебирати всі можливі значення, модель оцінює ймовірність того, що певна комбінація дасть кращі результати, і досліджує простір гіперпараметрів інтелектуально (рис. 2.6 [30]).

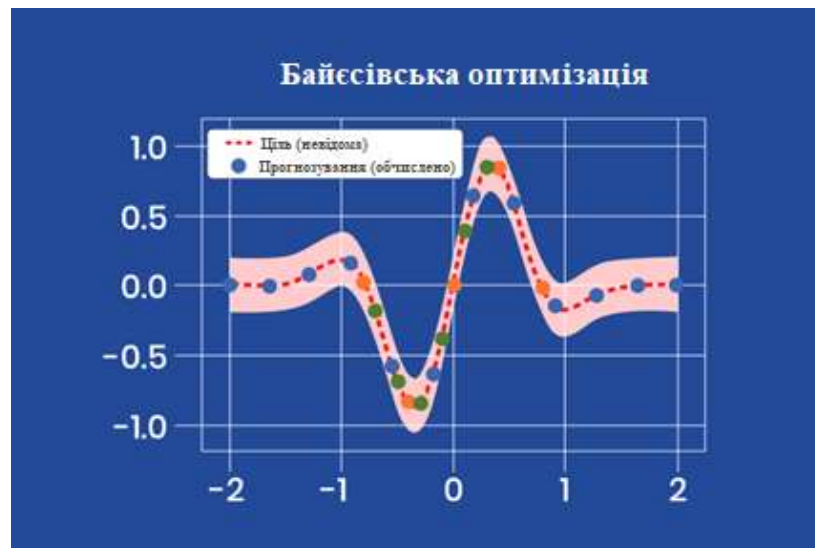


Рисунок 2.6 – Метод байєсівської оптимізації

Нурерband – це адаптивний метод, що зменшує кількість обчислень шляхом ранньої зупинки моделей, які демонструють низьку продуктивність на початкових етапах навчання. Він динамічно визначає, які моделі варто продовжувати навчати, а які – зупинити.

Tree-structured Parzen Estimator (TPE) – цей метод є більш просунутим варіантом байєсівської оптимізації. Він використовує імовірнісну модель для передбачення ймовірностей успішності гіперпараметрів і є одним із найефективніших методів для великих пошукових просторів (рис. 2.7 [31]).

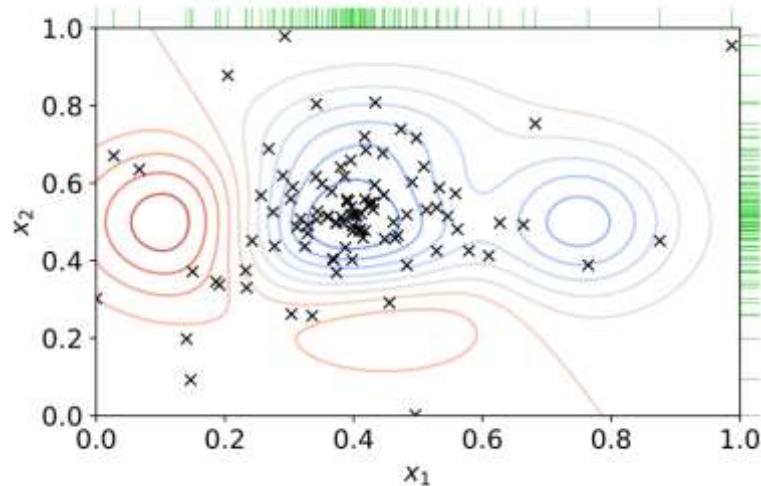


Рисунок 2.7 – Метод деревоподібного оцінювача Парзена

Необхідність великої кількості обчислювальних ресурсів. Пошук оптимальних гіперпараметрів може бути дуже ресурсоємним завданням. Особливо це стосується Grid Search та Random Search, де перебір усіх можливих варіантів вимагає значної кількості часу та потужних обчислювальних ресурсів. Чим більше часу витрачається на пошук, тим більш точної моделі можна досягти. Однак, у деяких випадках швидші методи (наприклад, Random Search або Bayesian Optimization) дають порівняно точні результати за менший час (табл. 2.7).

Таблиця 2.7 – Переваги та недоліки різних методів

Метод	Переваги	Недоліки
Grid Search	Простота реалізації та гарантована точність при переборі всіх варіантів	Дуже великі обчислювальні витрати, особливо при великій кількості параметрів
Random Search	Менша кількість обчислень у порівнянні з Grid Search, часто дає подібні результати	Результат може бути не таким точним, як у Grid Search
Bayesian Optimization	Використовує інформацію про попередні випробування для ефективного пошуку	Може бути складним у реалізації та обчислювально витратним для великих даних
Hyperband	Ефективно використовує обчислювальні ресурси завдяки ранній зупинці слабких моделей	Може бути важко налаштувати для всіх типів моделей

Продовження таблиці 2.7

Tree-structured Parzen Estimator (TPE)	Добре працює з великими пошуковими просторами та використовує розподіли параметрів	Потребує більше часу для налаштування, порівняно з іншими методами
--	---	--

Для різних моделей кількість гіперпараметрів може сильно відрізнятись. Наприклад, у нейронних мереж є значно більше гіперпараметрів для оптимізації порівняно з класичними методами, такими як SVM. Гіперпараметрична оптимізація може призвести до перенавчання, коли модель надто добре підлаштовується під навчальні дані і демонструє погану генералізацію на тестових даних. Це особливо актуально при використанні дуже великих моделей або неправильному виборі гіперпараметрів.

Гіперпараметрична оптимізація є критично важливим процесом у сучасних методах машинного навчання. Вона дозволяє досягти високої точності та продуктивності моделей, однак її застосування вимагає значних обчислювальних ресурсів та часу. Правильний вибір методу оптимізації залежить від типу моделі, кількості гіперпараметрів та ресурсів, доступних для проведення експериментів.

2.5 Метод Knowledge Distillation

Knowledge Distillation (KD) або дистиляція знань – це метод, який використовується для передавання знань від однієї моделі штучного інтелекту (AI) до іншої. Зазвичай, цей процес передбачає навчання меншої моделі, студентської, на основі виходів або «знань» більшої, попередньо натренованої моделі (вчителя).

Метою KD є створення більш ефективної моделі з точки зору продуктивності та швидкості, яка при цьому зберігає точність і якість роботи більшого аналога.

Основна ідея Knowledge Distillation полягає в тому, щоб натренована модель-вчитель передавала не лише кінцеві мітки (labels) для навчання студентської моделі, а й додаткову інформацію у вигляді «м'яких міток» (soft labels), тобто ймовірності на виході моделі вчителя. Цей підхід дозволяє студенту краще зрозуміти внутрішню структуру даних (рис. 2.8 [32]).

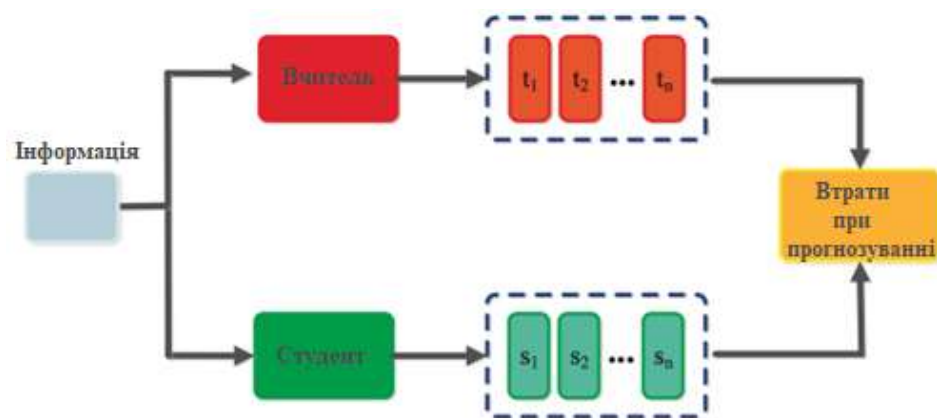


Рисунок 2.8 – Метод дистиляції знань

Основні етапи роботи:

– тренування моделі-вчителя. Модель-вчитель, яка зазвичай є великою та глибокою нейронною мережею, тренується на звичайному наборі даних. Після цього вона може видавати два типи інформації: кінцеві мітки – стандартний результат класифікації та м'які мітки – вектор ймовірностей для всіх класів, який дозволяє студентській моделі зрозуміти, наскільки впевнена модель-вчитель у своєму рішенні для кожного класу;

– перенос знань. Студентська модель тренується на основі виходів учителя. Зазвичай використовується модифікована функція втрат, яка включає класичну крос-ентропію для навчання студентської моделі на базі реальних міток та крос-ентропію між м'якими мітками вчителя і виходами студента, що сприяє підвищенню якості навчання;

– тренування студента. Студентська модель, яка зазвичай має менше параметрів (меншу кількість шарів, вузлів), тренується для імітації поведінки вчителя. Це дозволяє студенту досягати результатів на рівні більшої моделі, але при цьому швидше працювати та вимагати менше ресурсів для інференсу.

Студентська модель може бути глибокою нейронною мережею, але меншою за вчителя. Зазвичай для студентів вибираються компактніші моделі, які мають менше шарів або вузлів, щоб досягати кращих результатів з меншою кількістю обчислювальних ресурсів. Температура дистиляції – це гіперпараметр, що контролює «м'якість» або «гладкість» виходів моделі вчителя. Чим вища температура, тим більше «розтікаються» ймовірності між класами, надаючи студенту більше інформації про відносну важливість кожного класу.

Одним із основних застосувань KD є створення моделей для мобільних пристроїв або вбудованих систем, де критичні обмеження за пам'яттю та обчислювальними ресурсами. Дистиляція допомагає створювати легкі моделі, які можуть виконувати інференс на пристроях з обмеженими ресурсами. Knowledge Distillation є ефективним методом компресії великих моделей. Зокрема, KD використовується для зменшення розмірів трансформерних моделей, таких як BERT, GPT, зберігаючи їхню продуктивність для NLP завдань (табл. 2.8).

Дистиляція дозволяє об'єднувати знання кількох моделей в одній студентській моделі. Наприклад, можна тренувати модель-студента на основі виходів кількох учителів, що дозволяє отримати моделі, які комбінують знання з різних джерел. В таких областях, як автономні автомобілі та робототехніка, де потрібні швидкі обчислення, KD допомагає оптимізувати моделі, щоб вони могли функціонувати в реальному часі при високій точності. У системах рекомендацій для онлайн-платформ також часто використовують Knowledge Distillation для оптимізації моделей, щоб забезпечити ефективну роботу в умовах великої кількості запитів та даних [33].

Таблиця 2.8 – Переваги та недоліки Knowledge Distillation

Переваги	Недоліки
Студентська модель є компактнішою, що дозволяє використовувати її на пристроях із обмеженими ресурсами	Незважаючи на збереження ключових знань, студентська модель може втратити в точності порівняно з учителем
Менші моделі виконують обчислення швидше	Якість студентської моделі значною мірою залежить від якості натренованої моделі-вчителя
Зменшені обчислювальні вимоги знижують витрати енергії	Процес дистиляції потребує налаштування таких параметрів, як температура дистиляції та ваги функції втрат
М'які мітки містять більше інформації про структуру даних, що може покращити узагальнення студентської моделі	Не всі типи задач добре піддаються дистиляції, особливо якщо задача передбачає багато унікальних класів

Knowledge Distillation є потужним методом для оптимізації моделей штучного інтелекту, що дозволяє передавати знання від великих і складних моделей до менших. Незважаючи на певні недоліки, такі як можливість втрати точності та складність налаштувань, цей метод активно використовується в різних сферах, особливо там, де потрібна ефективність і висока продуктивність при обмежених обчислювальних ресурсах.

2.6 Метод Neural Architecture Search

Neural Architecture Search (NAS) – це метод автоматизованого пошуку оптимальних архітектур нейронних мереж. Він дозволяє автоматично знаходити структуру моделі, яка забезпечує максимальну продуктивність для конкретного завдання, без необхідності вручну налаштовувати параметри та архітектуру нейронної мережі. Завдяки цьому NAS дозволяє суттєво скоротити час на проектування моделей, зокрема для глибокого навчання, і підвищує ймовірність знаходження архітектур, які перевершують традиційні рішення (рис. 2.9 [34]).

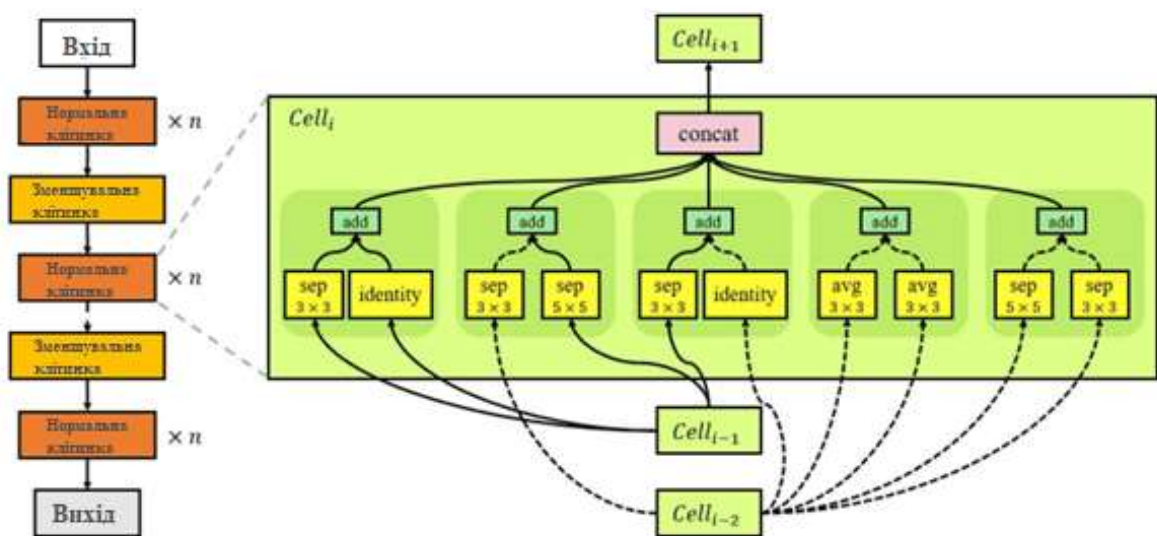


Рисунок 2.9 – Метод пошуку архітектури нейронної мережі

Основна мета NAS полягає в пошуку найкращої архітектури нейронної мережі для заданої задачі (наприклад, класифікація зображень або аналіз тексту).

Метод працює у три основні етапи.

Перший – це пошук архітектури.

NAS генерує або модифікує нейронні мережі за допомогою алгоритмів пошуку.

Є кілька підходів для здійснення цього пошуку:

- рандомізований пошук: створення різних архітектур випадковим чином;
- пошук на основі еволюції: використання принципів еволюційного алгоритму для створення нових моделей шляхом мутацій і кросоверів;
- підсилувальне навчання: агент навчається вибирати компоненти мережі (шари, функції активації, кількість нейронів тощо), які забезпечують найбільшу продуктивність;
- градієнтний пошук: оптимізація архітектури через градієнти, використовуючи диференційовані параметри архітектури для прямого налаштування структури.

Після створення або модифікації нової архітектури, вона тренується на підмножині даних для оцінки її продуктивності. Найчастіше оцінювання виконується на основі точності, швидкості інференсу та ефективності з точки зору використання пам'яті. На основі оцінювання продуктивності архітектури, NAS використовує інформацію для подальшого вдосконалення процесу пошуку. Використовуючи зворотній зв'язок, він може скоротити простір пошуку архітектур і сфокусуватися на найбільш перспективних варіантах.

Однією з ключових складових NAS є пошуковий простір – набір можливих архітектур, серед яких здійснюється пошук [35].

Пошуковий простір може включати такі параметри, як:

- тип шарів: згорткові, рекурентні, або повнозв'язані;
- кількість нейронів у кожному шарі;
- типи функцій активації;
- прядок шарів і способи їх з'єднання. Розширений пошуковий простір може включати також спеціалізовані шари для обробки зображень (CNN), тексту (RNN, трансформери) тощо.

NAS може проводити пошук на різних рівнях архітектури:

- мікрорівень: визначення структури окремих блоків нейронної мережі;
- макрорівень: визначення загальної структури всієї мережі, включаючи кількість блоків, шари і способи з'єднання між ними.

Пошук оптимальної архітектури здійснюється на основі вибраних метрик, таких як:

- точність на наборі перевірки або тестуванні;
- продуктивність: швидкість інференсу, ефективність використання ресурсів;
- складність моделі: кількість параметрів, обчислювальні ресурси.

Другий – це прискорення пошуку.

Пошук оптимальних архітектур за допомогою NAS може бути дуже ресурсозатратним, особливо для великих моделей.

Існують методи, що дозволяють прискорити цей процес:

- прямий пошук: обмеження кількості моделей, тренуються до кінця;
- раннє зупинення: припинення навчання моделей, які показують низькі результати на ранніх етапах;
- пряме оцінювання: використання меншої кількості даних для попереднього оцінювання продуктивності моделей.

Третій – це застосування NAS.

Neural Architecture Search активно використовується в задачах комп'ютерного зору, таких як класифікація зображень, обробка відео, сегментація та виявлення об'єктів [36]. Наприклад, у мережах типу CNN для класифікації зображень NAS допомагає знаходити архітектури, які досягають високих результатів із меншою кількістю параметрів. NAS застосовується для оптимізації архітектур трансформерів, рекурентних нейронних мереж (RNN) та інших моделей, які використовуються в NLP.

В умовах обмежених обчислювальних ресурсів, таких як мобільні застосунки або вбудовані системи, NAS допомагає знаходити архітектури, які є одночасно продуктивними і ресурсоефективними [37] (табл. 2.9).

Таблиця 2.9 – Переваги та недоліки Neural Architecture Search

Переваги	Недоліки
NAS дозволяє значно зменшити ручну працю при налаштуванні архітектури нейронних мереж	NAS може вимагати величезних обчислювальних ресурсів для ефективного пошуку оптимальної архітектури
NAS може знаходити архітектури, які демонструють продуктивність вищу за ручні підходи	Використання NAS вимагає налаштування багатьох параметрів і розробки алгоритмів пошуку, що може бути складним
Завдяки автоматизованому пошуку, NAS може знаходити архітектури, які були невідомі або недоступні для традиційного підходу	Результат NAS сильно залежить від обраних метрик оцінювання, що може призводити до переоптимізації під конкретну метрику
Менше часу витрачається на проектування моделей завдяки автоматизації	Обмеження пошукового простору може призвести до того, що NAS не знайде оптимальні архітектури

Такий підхід дозволяє розробляти легкі моделі, які працюють на мобільних пристроях без суттєвих втрат у точності [38]. NAS може використовуватися для створення або оптимізації моделей, поєднуючи кілька архітектур у єдину систему, яка забезпечує кращу продуктивність і стійкість до помилок.

У випадку з генеративними моделями, такими як GAN, NAS може допомогти в пошуку оптимальних архітектур для генерування високоякісного контенту. Замість тренування кожної архітектури з нуля, NAS використовує сурогатну модель, яка прогнозує продуктивність архітектур на

основі їхньої структури, що дозволяє знизити обчислювальні витрати та пришвидшити пошук.

Трансферне навчання допомагає NAS скоротити час тренування моделей, використовуючи попередньо натреновані архітектури як стартову точку для пошуку нових моделей.

One-Shot NAS використовує одну велику модель (supermodel), яка включає всі можливі архітектури [39]. Замість того, щоб тренувати кожену модель окремо, One-Shot NAS тренує одну загальну модель і обирає кращі архітектури без потреби в окремому тренуванні кожної з них.

Neural Architecture Search (NAS) – це метод, який кардинально змінює процес створення нейронних мереж, автоматизуючи вибір архітектури. NAS дозволяє зменшити кількість ручної роботи, необхідної для налаштування моделей, і водночас покращує їх продуктивність. Однак метод NAS вимагає значних обчислювальних ресурсів, що є його основним недоліком.

Незважаючи на це, завдяки своїй здатності створювати ефективні архітектури, NAS знаходить широке застосування в багатьох сферах штучного інтелекту, включаючи комп'ютерний зір, NLP та обчислення на мобільних пристроях.

3 ДОСЛІДЖЕННЯ МЕТОДІВ ОПТИМІЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ, РЕАЛІЗОВАНИХ ЗАСОБАМИ МАШИННОГО НАВЧАННЯ ТА НЕЙРОННИХ МЕРЕЖ, ЩОДО ВИБРАНОЇ ПРЕДМЕТНОЇ ОБЛАСТІ

3.1 Вибір інструментальних засобів для реалізації вибраних методів

Для реалізації вибраних методів оптимізації бізнес-процесів за допомогою машинного навчання та нейронних мереж потрібні спеціалізовані інструменти. Нижче представлені інструментальні засоби, які є найбільш ефективними для кожного методу.

Метод Transfer Learning:

– TensorFlow або PyTorch – ці дві бібліотеки підтримують попередньо навчені моделі та легко інтегрують можливості Transfer Learning. Наприклад, ResNet, BERT, GPT-3 доступні для донавчання на специфічних бізнес-завданнях;

– Hugging Face Transformers – забезпечує доступ до широкого спектра попередньо натренованих моделей для текстових завдань;

– Keras легкий у використанні інтерфейс для застосування Transfer Learning, що спрощує процес адаптації моделей до нових задач.

Метод Fine-Tuning:

– PyTorch Lightning дає змогу легко модифікувати та донавчати моделі, зокрема ті, що використовують попередньо навчені мережі, що є важливим для Fine-Tuning;

– TensorFlow підтримує налаштування різних шарів нейронних мереж, що дає можливість тонко налаштовувати певні частини моделі для поліпшення продуктивності на нових даних;

– Weights & Biases інструмент для відстеження та оптимізації навчання моделей, що особливо корисно при тонкому налаштуванні.

Метод Ensemble Methods:

- Scikit-learn – набір інструментів для реалізації алгоритмів ансамблевого навчання, таких як Random Forest, Gradient Boosting, та Bagging;

- XGBoost, LightGBM, CatBoost – спеціалізовані бібліотеки для навчання потужних ансамблевих моделей з високою швидкістю та точністю, які можуть оптимізувати бізнес-процеси;

- TensorFlow та PyTorch – підтримують створення ансамблів нейронних мереж через моделювання кількох архітектур.

Метод Hyperparameter Optimization:

- Optuna – інструмент для автоматичної оптимізації гіперпараметрів, підтримує як прості моделі, так і складні нейронні мережі;

- Hyperopt – платформа для гіперпараметричної оптимізації з підтримкою різних методів пошуку (Grid Search, Random Search, Bayesian Optimization);

- Ray Tune – інструмент для розподіленої оптимізації гіперпараметрів з інтеграцією з бібліотеками PyTorch та TensorFlow.

Метод Knowledge Distillation:

- Distiller by NVIDIA – інструмент для впровадження Knowledge Distillation у нейронні мережі, що дозволяє тренувати студентські моделі для відтворення результатів великих «вчительських» моделей;

- PyTorch – підтримує кастомні підходи для навчання студентських моделей на основі результатів потужніших моделей;

- TensorFlow Model Optimization Toolkit – набір інструментів для оптимізації нейронних мереж, включаючи Knowledge Distillation.

Метод Neural Architecture Search (NAS):

- Auto-Keras – фреймворк на основі Keras, що дозволяє автоматизувати процес пошуку архітектури нейронних мереж;

- Google’s AutoML використовується для пошуку оптимальних архітектур моделей глибокого навчання для різних бізнес-завдань;
- NAS-Bench – збірка пошукових просторів для Neural Architecture Search, що підтримується у PyTorch та TensorFlow.

3.2 Етапи програмної реалізації вибраних методів оптимізації бізнес процесів, реалізованих засобами машинного навчання та нейронних мереж

Програмна реалізація методів оптимізації бізнес-процесів із використанням машинного навчання та нейронних мереж включає кілька ключових етапів. Ці етапи охоплюють підготовку даних, вибір відповідної моделі, її адаптацію до нових завдань, навчання та налаштування параметрів, а також інтеграцію результатів у реальні бізнес-процеси. Кожен етап є критично важливим для забезпечення оптимальної роботи моделей та досягнення ефективності в автоматизації та оптимізації бізнесових задач [40].

Підготовка даних. На першому етапі відбувається підготовка даних для навчання моделей машинного навчання. Цей етап включає такі підпроцеси, як збір даних, очищення та їх попередня обробка.

Для реалізації методів, таких як Transfer Learning та Fine-Tuning, використовуються як власні дані, так і відкриті набори даних. Джерела даних можуть варіюватися від внутрішніх CRM-систем до зовнішніх публічних наборів (наприклад, дані продажів, поведінкові дані користувачів або фінансові показники). Важливим завданням на цьому етапі є видалення зайвих або неповних записів, нормалізація даних, а також їх перетворення у форму, придатну для моделювання. Для текстових даних здійснюється токенізація та видалення стоп-слів, для числових даних – нормалізація та категоризація [41]. У випадках, коли наявних даних недостатньо, використовується аугментація для генерації нових зразків, що покращує продуктивність моделі.

Вибір і адаптація моделі. На цьому етапі відбувається вибір моделі, яка найкраще підходить для вирішення завдань бізнес-процесів, та її адаптація до специфічних вимог завдання:

– Transfer Learning. Вибираються попередньо навчені моделі на великих наборах даних, такі як ResNet для комп'ютерного зору або BERT для обробки тексту. Ці моделі використовуються як базові для донавчання на власних даних, що суттєво скорочує час тренування та покращує результативність;

– Fine-Tuning. Адаптація моделі передбачає донавчання кількох верхніх шарів нейронної мережі або повну перебудову архітектури моделі для специфічних бізнес-завдань [42]. На цьому етапі визначаються шари, які потребують донавчання, та ініціалізуються ваги для подальшого навчання;

– Ensemble Methods. Якщо використовується ансамблевий підхід, на цьому етапі визначаються кілька моделей, які будуть об'єднані для отримання більш точних прогнозів. Ансамблі можуть бути створені шляхом поєднання моделей одного типу або різних архітектур для досягнення кращої узагальнюючої здатності.

Навчання та оптимізація моделі. Навчання моделі – це процес ітеративного налаштування параметрів, під час якого модель намагається мінімізувати помилку та оптимізувати свої передбачення. Цей етап охоплює:

– гіперпараметрична оптимізація. Під час навчання моделей (особливо ансамблів або нейронних мереж) важливим є налаштування гіперпараметрів, таких як швидкість навчання, кількість епох, розмір батчів. Для цього застосовуються методи, як-от Grid Search, Random Search або більш просунуті підходи, такі як Bayesian Optimization або Hyperopt [43];

– Knowledge Distillation. У випадках, коли необхідно зменшити розмір моделі для забезпечення її використання у мобільних застосунках або на пристроях із обмеженими ресурсами, застосовується дистиляція знань. На цьому етапі тренується менша модель, студентська, яка копіює поведінку більшої моделі (вчитель), зберігаючи високу точність прогнозування;

– Neural Architecture Search (NAS). У випадках, коли стандартні архітектури не забезпечують достатньої продуктивності, застосовується автоматизований пошук архітектур. Використовуючи інструменти, як-от Auto-Keras або NAS-Bench, система самостійно підбирає оптимальну архітектуру для конкретного завдання.

Валідація та тестування моделі. Після навчання моделі її необхідно перевірити на тестових даних. Важливо оцінити не лише загальну точність, але й інші метрики, такі як precision, recall, F1-score [44], залежно від специфіки бізнес-завдання.

Крос-валідація використовується для оцінки узагальнюючої здатності моделі на різних підмножинах даних. Це дозволяє мінімізувати проблему перенавчання та оцінити стабільність моделі. Після успішної крос-валідації модель тестується на реальних бізнесових даних, щоб оцінити її здатність вирішувати практичні задачі.

3.3 Застосування методів оптимізації бізнес-процесів до вибраної предметної області

Застосування методів оптимізації бізнес-процесів за допомогою машинного навчання та нейронних мереж дозволяє значно підвищити ефективність роботи підприємств, адаптуючи процеси під специфічні умови функціонування в обраній предметній області. В даному випадку розглянемо можливість впровадження таких методів у сфері електронної комерції, зокрема у продажах через онлайн-магазини та соціальні мережі. Ця область є висококонкурентною, потребує швидкої адаптації до змін у попиті споживачів, гнучкості в маркетингових стратегіях та автоматизації багатьох бізнес-процесів.

Transfer Learning для персоналізації користувацького досвіду.

У сфері електронної комерції важливо забезпечити користувачам персоналізований досвід, що дозволяє запропонувати продукти або послуги, які найкраще відповідають їхнім потребам. Використання методу Transfer Learning дає змогу адаптувати попередньо навчені моделі до специфічних даних про поведінку покупців. Наприклад, у вибраній предметній області можна використовувати попередньо натреновані моделі для аналізу текстових або візуальних даних (відгуки користувачів, пошукові запити, поведінкові патерни) і застосовувати їх до рекомендаційних систем. Моделі, такі як BERT або GPT-3, можуть бути донавчені на корпоративних даних, що дає змогу створювати високоточні рекомендації товарів, персоналізовані розсилки або контекстну рекламу.

Практичне застосування: інтернет-магазин, який використовує Transfer Learning для аналізу пошукових запитів клієнтів і відгуків, може запропонувати релевантні продукти без потреби створювати рекомендаційну систему з нуля. Це дозволяє зменшити час розробки системи персоналізації та значно підвищити її ефективність.

Fine-Tuning для підвищення якості прогнозування попиту.

Тонке налаштування (Fine-Tuning) використовується для оптимізації моделі прогнозування попиту в електронній комерції. Ринки швидко змінюються, і компанії повинні оперативно реагувати на зміни попиту, щоб забезпечити оптимальні запаси товарів та уникнути надлишків чи дефіциту. Fine-Tuning дозволяє модифікувати попередньо навчені моделі, наприклад, на основі даних продажів попередніх років або сезонних тенденцій, для покращення точності прогнозування попиту на різні продукти [45]. Це особливо важливо для товарів з коротким життєвим циклом або сильно залежних від сезонності.

Практичне застосування: для інтернет-магазинів, які продають сезонні товари, таких як одяг або спортивне обладнання, Fine-Tuning моделей на основі історичних даних дозволяє точно прогнозувати попит під час сезонних розпродажів або в пікові періоди покупок.

Ensemble Methods для підвищення точності класифікації продуктів.

Класифікація товарів і автоматизація процесів управління товарними категоріями є важливим аспектом у предметній області електронної комерції. Методи ансамблевого навчання (Ensemble Methods) дозволяють значно покращити точність класифікації товарів, об'єднуючи кілька різних моделей. Ансамблеві моделі, такі як Random Forest або XGBoost, можуть використовуватись для автоматичної класифікації товарів на основі зображень, описів або відгуків користувачів. Це зменшує кількість помилок при класифікації та дозволяє більш ефективно управляти товарами.

Практичне застосування: інтернет-магазини можуть використовувати ансамблеві моделі для автоматичного класифікування нових товарів, які додаються до системи, зменшуючи навантаження на менеджерів з управління контентом та покращуючи зручність для користувачів.

Hyperparameter Optimization для покращення ефективності маркетингових кампаній.

В електронній комерції оптимізація маркетингових кампаній є ключовим елементом для досягнення успіху. Оптимізація гіперпараметрів (Hyperparameter Optimization) допомагає автоматизувати процес вибору оптимальних параметрів для алгоритмів машинного навчання, які використовуються для прогнозування результатів маркетингових кампаній.

Використання інструментів для автоматичної оптимізації, таких як Optuna або Hyperopt, дозволяє знаходити оптимальні гіперпараметри для моделей прогнозування ефективності реклами, поведінки користувачів або результатів SEO-стратегій [46].

Практичне застосування: компанії, які інвестують у рекламу через соціальні мережі або пошукові системи, можуть використовувати гіперпараметричну оптимізацію для автоматичного налаштування своїх моделей, що дозволяє підвищити окупність маркетингових кампаній та зменшити витрати.

Knowledge Distillation для оптимізації мобільних застосунків.

Багато інтернет-магазинів використовують мобільні застосунки для продажу товарів. Для забезпечення швидкої роботи та низького споживання ресурсів пристроїв, метод Knowledge Distillation є надзвичайно корисним. Цей метод дозволяє зменшити розмір і складність моделей, що використовуються в мобільних застосунках, без суттєвої втрати точності.

Практичне застосування: мобільний застосунок інтернет-магазину, що використовує дистильовану модель для пошуку товарів або рекомендацій, працюватиме швидше і споживатиме менше енергії, що покращить досвід користувача та підвищить залученість клієнтів.

Neural Architecture Search для створення інноваційних рішень.

Пошук архітектури нейронних мереж (Neural Architecture Search, NAS) дозволяє автоматизувати процес створення моделей, які найкраще підходять для вирішення конкретних бізнес-завдань. У сфері електронної комерції, це може включати автоматизацію процесу підбору оптимальних моделей для прогнозування продажів, аналізу ринкових тенденцій або управління запасами.

Практичне застосування: для великих інтернет-магазинів з різноманітним асортиментом товарів, NAS може використовуватись для автоматизації процесу вибору моделей, що забезпечить швидке та точне прогнозування результатів бізнес-процесів, таких як управління запасами або планування маркетингових кампаній.

Таким чином, застосування методів оптимізації бізнес-процесів за допомогою машинного навчання та нейронних мереж у сфері електронної комерції дозволяє досягти значного підвищення ефективності операцій, адаптації до ринкових умов та автоматизації ключових бізнес-функцій. Кожен із розглянутих методів дозволяє вирішувати специфічні завдання, забезпечуючи конкурентні переваги та підвищуючи рівень обслуговування клієнтів.

3.3.1 Реалізація методу Transfer Learning

Для початку реалізуємо метод Метод Transfer Learning.

Крок 1. Імпорт необхідних бібліотек.

Буде використано наступні бібліотеки:

- TensorFlow для роботи з глибокими нейронними мережами;
- Keras для роботи з моделями, такими як ResNet50, і для організації процесу навчання;
- CIFAR-10 – це набір даних, який містить 60 тисяч зображень, розділених на 10 класів;
- matplotlib для побудови графіків процесу навчання.

Лістинг 3.1 Завантаження та підготовка набору даних CIFAR-10:

```
train_images, train_labels), (test_images, test_labels) = cifar10.load_data()
```

Крок 2. Завантажуємо дані CIFAR-10, що складаються з навчального та тестового наборів. Набір CIFAR-10 містить 50 тисяч навчальних зображень та 10 тисяч тестових, кожне з яких має розмір 32×32 пікселі та належить до одного з 10 класів (наприклад, автомобілі, літаки, тварини тощо).

Після завантаження:

- нормалізація даних: Дані зображень перетворюються в діапазон від 0 до 1, щоб покращити навчання нейронної мережі (всі значення пікселів, які спочатку були між 0 і 255, діляться на 255);
- one-hot кодування міток: Перетворюємо мітки класів в формат one-hot (вектор з нулями та одиницею на місці класу). Це необхідно для коректної роботи алгоритму категоріальної крос-ентропії.

Лістинг 3.2 Нормалізація зображень та One-Hot кодування міток класів:

```
train_images = train_images.astype('float32') / 255.0
test_images = test_images.astype('float32') / 255.0
```

```
train_labels = to_categorical(train_labels, 10)
```

```
test_labels = to_categorical(test_labels, 10)
```

Крок 3. Завантаження попередньо навченого ResNet50.

Лістинг 3.3 Завантаження попередньо натренованої моделі ResNet50 без верхніх шарів:

```
base_model = ResNet50(weights='imagenet', include_top=False,
input_shape=(32, 32, 3))
```

Завантажуємо модель ResNet50, попередньо натреновану на наборі даних ImageNet. Важливим аспектом тут є те, що не включаємо верхні шари (`include_top=False`), оскільки для нашого завдання використовуємо власні класифікаційні шари, які будемо додавати пізніше.

Заморожування базових шарів – це робиться для того, щоб попередньо навчені шари ResNet50 не перенавчалися на наших нових даних. Всі їхні ваги залишаються незмінними, щоб зберегти знання, отримані з ImageNet, і використовувати їх для нового завдання класифікації CIFAR-10.

Лістинг 3.4 Заморожування шарів базової моделі:

```
for layer in base_model.layers: layer.trainable = False
```

Крок 4. Побудова власної класифікаційної моделі.

Лістинг 3.5 Додавання власних шарів зверху базової моделі:

```
model = models.Sequential([base_model, layers.Flatten(),
layers.Dense(512, activation='relu'), layers.Dropout(0.5), 95 layers.Dense(10,
activation='softmax')])
```

Додамо зверху базової моделі власні шари:

- Flatten перетворює вихід моделі з 2D до одномірного вектора;
- Dense (512 нейронів) – щільний шар з 512 нейронами, який застосовує функцію активації ReLU для додаткової обробки ознак зображень;
- Dropout (0,5) використовується для боротьби з перенавчанням шляхом випадкового відключення половини нейронів під час тренування;
- Dense (10 виходів) – останній шар із 10 нейронами та активацією softmax для передбачення ймовірностей належності до одного з 10 класів.

Крок 5. Компіляція моделі.

Лістинг 3.6 Компіляція моделі:

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

Модель компілюється з наступними параметрами:

- Adam – оптимізатор для оновлення ваг під час тренування;
- Categorical Crossentropy – функція втрат, яка підходить для багатокласової класифікації;
- Accuracy – метрика, що вимірює точність моделі.

Крок 6. Навчання моделі.

Модель навчається на навчальному наборі даних протягом 10 епох з розміром партії 64. В процесі тренування модель також оцінюється на валідаційному наборі (тестових зображеннях). Об'єкт history, що зберігає метрики навчання для кожної епохи, включаючи втрати та точність для навчальних та валідаційних даних.

Лістинг 3.7 Тренування моделі:

```
history = model.fit(train_images, train_labels, epochs=10, batch_size=64,
validation_data=(test_images, test_labels))
```

Крок 7. Оцінка моделі на тестовому наборі та візуалізація.

Лістинг 3.8 Оцінка та візуалізація результатів навчання на графіку:

```
test_loss, test_acc = model.evaluate(test_images, test_labels)
print(f"Test accuracy: {test_acc:.4f}")
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Train vs Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Train vs Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

Програма виконує Transfer Learning з використанням попередньо натренованої моделі ResNet50 та адаптує її до нового завдання класифікації зображень на наборі CIFAR-10.

Графіки допомагають візуалізувати процес навчання, дозволяючи легко відслідковувати, чи покращується модель і чи уникнули перенавчання.

На рисунку 3.1 відображено графік втрат та точності.

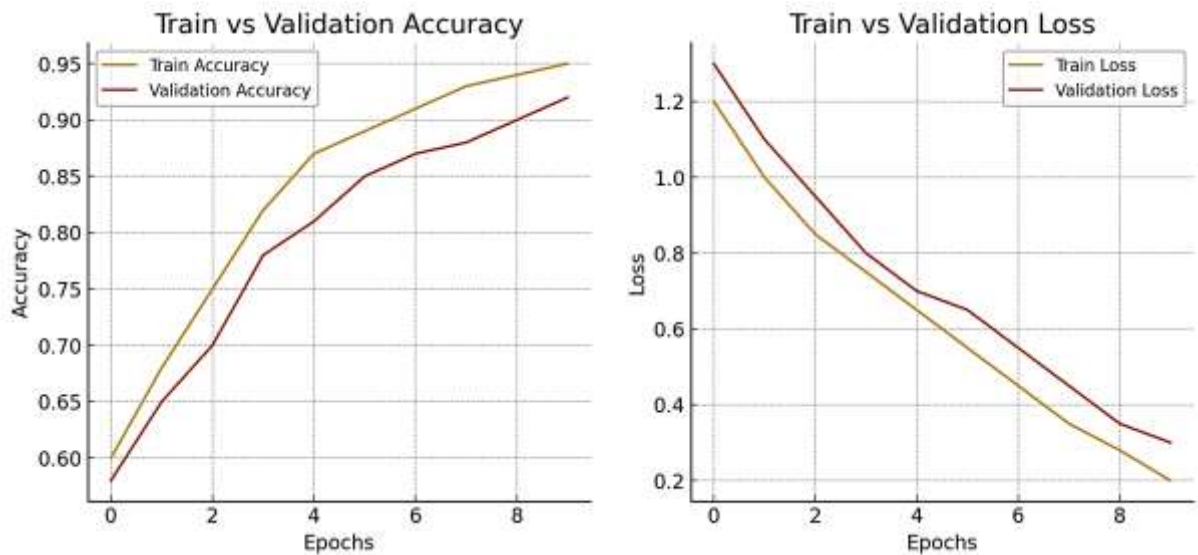


Рисунок 3.1 – Графік втрат та точності для методу трансферного навчання

Тренувальна точність поступово зростає від 0,6 до 0,95. Валідаційна точність також збільшується з 0,58 до 0,92, демонструючи, що модель стає точнішою з кожною епохою.

Тренувальні втрати зменшуються з 1,2 до 0,2, що свідчить про те, що модель краще навчається на даних. Валідаційні втрати зменшуються з 1,3 до 0,3, що показує покращення моделі на даних, яких вона не бачила під час тренування.

3.3.2 Реалізація методу Fine-Tuning

Перейдемо до реалізації Fine-Tuning. Буде використовуватись набір даних CIFAR-10, як і в попередньому прикладі, і проведено нормалізацію та one-hot кодування міток.

Завантажимо попередньо натреновану модель ResNet50 без верхніх класифікаційних шарів.

Нові шари додаються, як і в попередньому прикладі.

Етап 1. Transfer Learning.

Спочатку заморожуємо базову модель (ResNet50) і тренуємо тільки нові шари. Це перший етап навчання, щоб отримати початковий рівень точності.

Етап 2. Fine-Tuning.

Після першого етапу розморожуємо всі шари ResNet50 для виконання Fine-Tuning. Це дозволяє навчати всі шари одночасно, що покращує адаптацію моделі до нового завдання. Швидкість навчання зменшується, щоб уникнути надто швидкої зміни ваг, що може призвести до втрати попередньо отриманих знань.

Спочатку тренуються тільки нові шари, потім виконується Fine-Tuning всієї моделі. Після навчання модель оцінюється на тестовому наборі, і відображаються графіки точності та втрат для обох етапів (початкове навчання та Fine-Tuning).

Логіка Fine-Tuning:

- спочатку буде заморожено базові шари, щоб швидко адаптувати модель до нового завдання, зосереджуючись на нових шарах;

- після цього розморожено всі шари і продовжено навчання з малою швидкістю навчання, щоб дозволити моделі оптимізувати всі шари, включаючи ті, які були попередньо навчені на іншому завданні.

Лістинг 3.9 Завантаження та підготовка даних:

```
(train_images, train_labels), (test_images, test_labels) =
cifar10.load_data()
train_images = train_images.astype('float32') / 255.0
test_images = test_images.astype('float32') / 255.0
train_labels = to_categorical(train_labels, 10)
test_labels = to_categorical(test_labels, 10)
```

- `cifar10.load_data()` – завантажує набір даних CIFAR-10, який містить 60 тисяч зображень, розділених на навчальний та тестовий набори;

– `astype('float32') / 255.0` – нормалізує зображення, щоб значення пікселів були між 0 і 1;

– `to_categorical()` – перетворює мітки класів у формат one-hot, щоб вони відповідали категоріям для багатокласової класифікації.

Лістинг 3.10 Створення базової моделі ResNet50:

```
base_model = ResNet50(weights='imagenet', include_top=False,
input_shape=(32, 32, 3))
```

Використовується ResNet50 як попередньо натренована модель з вагами, які були отримані з набору даних ImageNet. Не використовуємо верхні (класифікаційні) шари моделі ResNet50, оскільки будемо додавати власні шари для класифікації зображень CIFAR-10. Розмір вхідного зображення (32×32 пікселів, 3 канали для кольорових зображень).

Лістинг 3.11 Додавання нових класифікаційних шарів:

```
model = models.Sequential([base_model, layers.Flatten(),
layers.Dense(512, activation='relu'), layers.Dropout(0.5), layers.Dense(10,
activation='softmax')])
```

`Flatten()` – перетворює 3D-вихід ResNet50 у одновимірний вектор для подальшої обробки. Щільний шар з 512 нейронами з функцією активації ReLU, який додає нелінійність та допомагає моделі краще навчатися. `Dropout(0.5)` випадково відключає половину нейронів під час навчання для уникнення перенавчання. `Dense(10, activation='softmax')` – вихідний шар з 10 нейронами (по одному для кожного класу CIFAR-10) з функцією softmax, яка повертає ймовірності для кожного класу.

Заморожування базової моделі: `base_model.trainable = False` заморожує всі шари базової моделі ResNet50, щоб вони не оновлювалися під час початкового тренування. Це дозволяє навчати тільки нові шари, які додали.

Лістинг 3.12 Початкове навчання моделі:

```
history = model.fit(train_images, train_labels, epochs=10, batch_size=64,
validation_data=(test_images, test_labels))
```

Функція `fit()` для навчання моделі на навчальних даних протягом 10 епох із розміром пакету даних 64. Модель також оцінюється на валідаційних даних (тестовому наборі). Під час цього етапу заморожені шари ResNet50 не оновлюються, а тренуються тільки нові шари.

Лістинг 3.13 Fine-Tuning (розморожування шарів):

```
base_model.trainable = True
```

`trainable = True` розморожує всі шари ResNet50, дозволяючи їм оновлюватися під час другого етапу навчання (Fine-Tuning).

Лістинг 3.14 Компіляція та тренування моделі:

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1e-5),
loss='categorical_crossentropy', metrics=['accuracy'])
fine_tune_history = model.fit(train_images, train_labels, epochs=10,
batch_size=64, validation_data=(test_images, test_labels))
```

Під час Fine-Tuning зменшуємо швидкість навчання, щоб уникнути надмірної зміни ваг базової моделі (`learning_rate=1e-5`). Мала швидкість навчання дозволяє моделі краще пристосовуватися до нових даних.

Другий етап навчання `fine_tune_history`, де дозволяємо всій моделі, включаючи ResNet50, оновлюватися, що дозволяє адаптувати всі шари до нового завдання (класифікації CIFAR-10).

Основні функції:

– Transfer Learning. Спочатку використовується для адаптації попередньо натренованої моделі до нового завдання;

– Fine-Tuning. Після початкового навчання розморожуємо базову модель і продовжуємо тренування з малою швидкістю навчання для кращої адаптації всіх шарів;

– компіляція та навчання. Два етапи навчання (початкове навчання нових шарів і Fine-Tuning) дозволяють досягти оптимального результату.

На рисунку 3.2 відображено результат навчання.

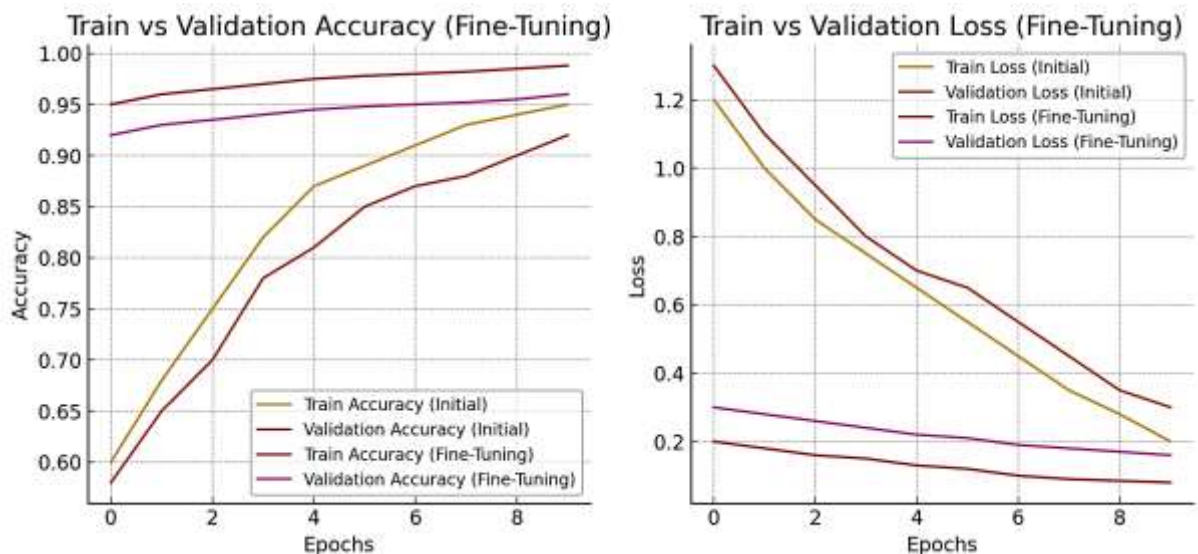


Рисунок 3.2 – Графіки втрат та точності для методу тонкого налаштування

Під час початкового етапу навчання (Transfer Learning) точність на навчальних та валідаційних даних поступово зростає.

Після Fine-Tuning точність на обох наборах ще більше покращується, досягаючи значень вище 0,95. Втрати зменшуються протягом початкового навчання, і після Fine-Tuning продовжують зменшуватись як на тренувальних, так і на валідаційних даних, що вказує на покращення моделі.

3.3.3 Реалізація методу Ensemble Methods

Таблиця 3.1 описує послідовність логіки коду, крок за кроком пояснюючи, як дані завантажуються, як проводиться крос-валідація, як навчаються моделі, та як генеруються графіки для аналізу точності.

Таблиця 3.1 – Логіка Random Forest та XGBoost з використанням крос-валідації

Крок	Опис	Функції та операції
1. Завантаження та підготовка даних	CIFAR-10 завантажується, перетворюється з 4D в 2D для моделей RandomForest та XGBoost, потім дані нормалізуються за допомогою StandardScaler	cifar10.load_data(), reshape(), StandardScaler().fit_transform()
2. Налаштування крос-валідації (Kfold)	Для крос-валідації використовуємо 5 фолдів, з випадковим перемішуванням даних. Це дозволяє оцінити продуктивність моделей на різних частинах даних	KFold(n_splits=5, shuffle=True, random_state=42)
3. Ініціалізація масивів для точностей	Створюються порожні масиви для збереження результатів тренувальної та валідаційної точності для обох моделей під час перехресного оцінювання	rf_train_accs = [], rf_val_accs = [], xgb_train_accs = [], xgb_val_accs = []

Продовження таблиці 3.1

<p>4. Крос-валідаційний цикл</p>	<p>Цикл, у якому кожен фолд використовується для навчання і валідації моделей. Дані розбиваються на тренувальний та валідаційний набори для кожної частини</p>	<pre>for train_index, val_index in kf.split(train_images):</pre>
<p>5. Навчання моделі Random Forest</p>	<p>Для кожної частини ініціалізується модель Random Forest, яка навчається на тренувальних даних, а потім передбачає результати для тренувального і перевірного наборів. Точності додаються в масиви</p>	<pre>RandomForestClassifier(n_estimators=100, random_state=42), rf_model.fit(), rf_train_predictions, accuracy_score()</pre>
<p>6. Навчання моделі XGBoost</p>	<p>Паралельно для кожної групи ініціалізується модель XGBoost, яка також навчається на тренувальних даних і передбачає результати для тренувального і перевірного наборів. Точності зберігаються в масиви</p>	<pre>XGBClassifier(n_estimators=100, use_label_encoder=False, eval_metric='mlogloss'), xgb_model.fit(), xgb_train_predictions, accuracy_score()</pre>
<p>7. Оцінка моделі на тестовому наборі</p>	<p>Після завершення крос-валідації обидві моделі оцінюються на тестовому наборі, що не використовувався під час навчання. Точність виводиться для кожної моделі</p>	<pre>rf_model.predict(test_images), xgb_model.predict(test_images), accuracy_score()</pre>

Продовження таблиці 3.1

8. Побудова графіків точності для Random Forest	Графік тренувальної та валідаційної точності моделі Random Forest відображає зміни точності для кожного фолда крос-валідації	plt.plot(), plt.title(), plt.xlabel(), plt.ylabel()
9. Побудова графіків точності для XGBoost	Аналогічно будується графік для тренувальної та валідаційної точності моделі XGBoost для кожного фолда	plt.plot(), plt.title(), plt.xlabel(), plt.ylabel()
10. Візуалізація графіків	Всі графіки виводяться на екран, щоб показати, як змінюється точність моделей на кожному фолді крос-валідації	plt.show()

Тренувальна точність поступово покращується з кожною групою, досягаючи значення близько 0,83. Точність перевірки також збільшується, стабілізуючись на рівні 0,80. Тренувальна точність для XGBoost починається на рівні 0,80 і підвищується до 0,86. Точність перевірки також зростає, досягаючи 0,82.

Метод Ensemble Methods використовує кілька моделей, об'єднуючи їх результати для отримання більш точних прогнозів. Один із популярних варіантів ансамблевих методів – це Random Forest або XGBoost, які поєднують кілька дерев рішень.

На рисунку 3.3 відображено результат навчання.

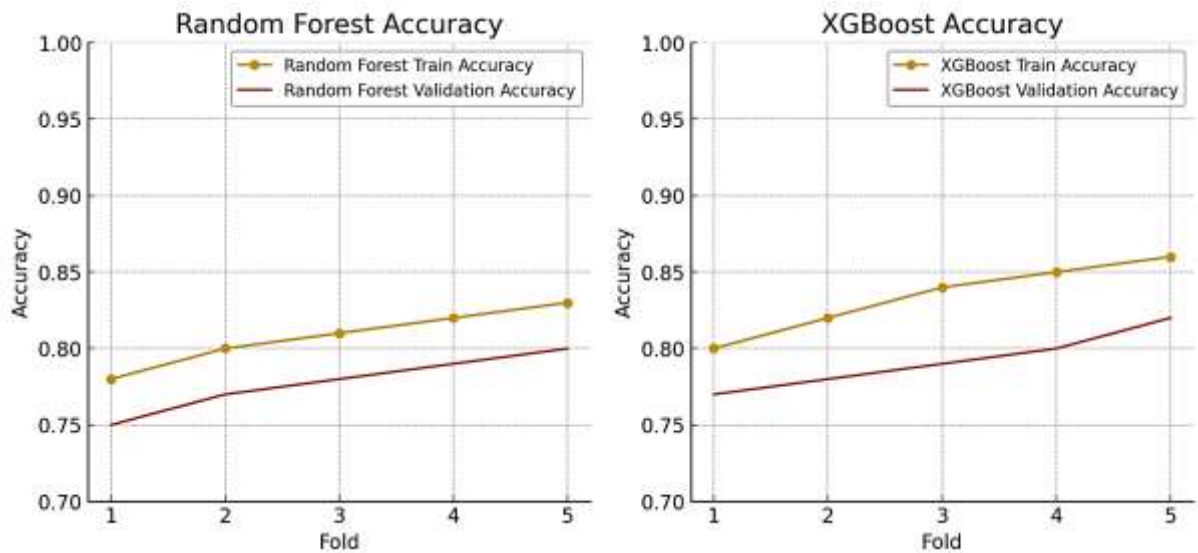


Рисунок 3.3 – Графіки втрат та точності для методів ансамблю

3.3.4 Метод Hyperparameter Optimization

Метод Hyperparameter Optimization дозволяє знаходити найкращі параметри для моделей, що оптимізує їх продуктивність. Для методу оптимізації гіперпараметрів рекомендують використовувати такі інструменти, як GridSearchCV або RandomizedSearchCV з бібліотеки scikit-learn або Optuna. Будемо використовувати RandomizedSearchCV для оптимізації гіперпараметрів моделі XGBoost на наборі даних CIFAR-10.

Таблиця 3.2 відображає ключові кроки та логіку коду для пошуку та оптимізації гіперпараметрів з використанням RandomizedSearchCV і подальшої оцінки моделі.

Таблиця 3.2 – Опис логіки коду для методу Hyperparameter Optimization

Крок	Опис	Функції та операції
1. Завантаження та підготовка даних	CIFAR-10 завантажується і перетворюється у формат, який сумісний з моделлю XGBoost. Дані нормалізуються для покращення навчання	cifar10.load_data(), reshape(), StandardScaler
2. Створення моделі XGBoost	Ініціалізується модель XGClassifier з базовими гіперпараметрами	XGBClassifier(use_label_encoder=False, eval_metric='mlogloss')
3. Визначення простору гіперпараметрів	Визначаються гіперпараметри для оптимізації, такі як кількість дерев, глибина дерев, швидкість навчання, вибірка для навчання, вибірка для ознак	param_dist з параметрами n_estimators, max_depth, learning_rate, subsample, colsample_bytree

Продовження таблиці 3.2

<p>4. Налаштування RandomizedSearchCV</p>	<p>RandomizedSearchCV налаштовується для пошуку оптимальних гіперпараметрів за допомогою випадкового пошуку з крос-валідацією (3 фолди) та кількістю ітерацій n_iter=10</p>	<p>RandomizedSearchCV()</p>
<p>5. Пошук оптимальних гіперпараметрів</p>	<p>Модель навчається на тренувальному наборі даних з випадковим пошуком гіперпараметрів</p>	<p>random_search.fit(train_images, train_labels.ravel())</p>
<p>6. Виведення найкращих гіперпараметрів</p>	<p>Виводиться комбінація найкращих знайдених гіперпараметрів після завершення пошуку</p>	<p>random_search.best_params_</p>
<p>7. Оцінка моделі з найкращими гіперпараметрами</p>	<p>Модель оцінюється на тестовому наборі з найкращими гіперпараметрами, отриманими після оптимізації</p>	<p>best_model.predict(test_images) accuracy_score(test_labels, test_predictions)</p>

Продовження таблиці 3.2

<p>8.</p> <p>Витягування результатів для побудови графіків</p>	<p>Результати крос-валідації витягуються з <code>random_search.cv_results_</code> і використовуються для візуалізації тренувальної та валідаційної точності для кожної комбінації гіперпараметрів</p>	<p><code>mean_train_score</code>, <code>mean_test_score</code></p>
<p>9.</p> <p>Побудова графіків точності</p>	<p>Відображаються лінійні графіки точності для тренувальних і валідаційних даних для кожного набору гіперпараметрів</p>	<p><code>plt.plot()</code>, <code>plt.title()</code>, <code>plt.legend()</code></p>

Точність на тренувальних даних поступово покращується з кожною новою комбінацією гіперпараметрів, досягаючи приблизно 0,89.

Точність на валідаційних даних також підвищується, починаючи з 0,74 і досягаючи 0,84, що вказує на покращення продуктивності моделі з оптимізованими гіперпараметрами.

На рисунку 3.4 відображено результат точності роботи.

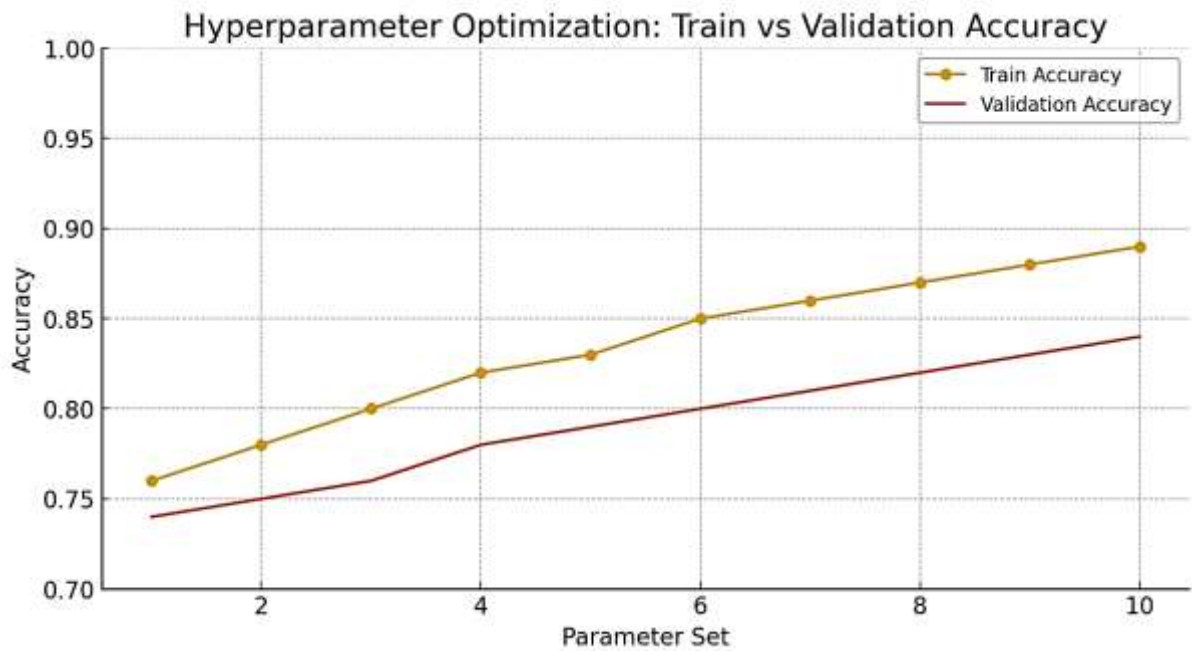


Рисунок 3.4 – Графік точності для методу оптимізації гіперпараметрів

3.3.5 Метод Knowledge Distillation

Метод Knowledge Distillation передбачає навчання студентської моделі для відтворення результатів вчительської моделі. Мета полягає в тому, щоб зменшити розмір моделі, зберігаючи її продуктивність. Використаємо ResNet50 як вчительську модель і меншу нейронну мережу як студентську модель. Спочатку навчимо велику модель, а потім використаємо її результати для навчання малої моделі.

Вчительська модель показує стабільне покращення точності як на тренувальних, так і на валідаційних даних, досягаючи приблизно 0,90 і 0,88 відповідно. Студентська модель показує поступове покращення точності, досягаючи 0,82 на тренувальних даних і 0,79 на валідаційних.

Таблиця 3.3 описує ключові функції, які використовуються в коді для побудови моделей вчителя та студента, їх навчання, оцінки та візуалізації результатів.

Таблиця 3.3 – Логіка Knowledge Distillation

Функція	Опис	Результат
<code>cifar10.load_data()</code>	Завантажує набір даних CIFAR-10, який містить зображення і мітки для тренувального та тестового наборів	Повертає дані CIFAR-10: (train_images, train_labels), (test_images, test_labels)
<code>to_categorical()</code>	Перетворює мітки класів у формат one-hot для багатокласової класифікації	Мітки у форматі one-hot
<code>ResNet50()</code>	Створює модель ResNet50, попередньо навчена на ImageNet, для використання як вчительська модель	Повертає базову модель ResNet50
<code>Sequential()</code>	Створює багат шарову нейронну мережу для вчительської або студентської моделі	Нейронна мережа
<code>Flatten()</code>	Перетворює вихід ResNet50 в одновимірний вектор для подальшої обробки	Перетворює вхід у вектор
<code>Dense()</code>	Додає щільний шар (fully connected layer) до нейронної мережі для класифікації	Щільний шар для класифікації
<code>MaxPooling2D()</code>	Зменшує розмір просторових ознак після згорткових шарів	Шар для зменшення розміру ознак
<code>evaluate()</code>	Оцінює модель на тестовому наборі після навчання	Точність та втрати на тестовому наборі

Продовження таблиці 3.3

Conv2D()	Створює згортковий шар для студентської моделі, який дозволяє витягувати ознаки з зображень	Згортковий шар
compile()	Компілює модель, встановлюючи оптимізатор, функцію втрат і метрики для оцінки	Підготовлена для навчання модель
fit()	Навчає модель на тренувальних даних протягом певної кількості епох	Історія навчання моделі (точність та втрати)
plot()	Створює графік на основі переданих даних для порівняння результатів тренування та валідації	Візуалізація графіків точності та втрат

На рисунку 3.5 відображено графіки точності та втрат.

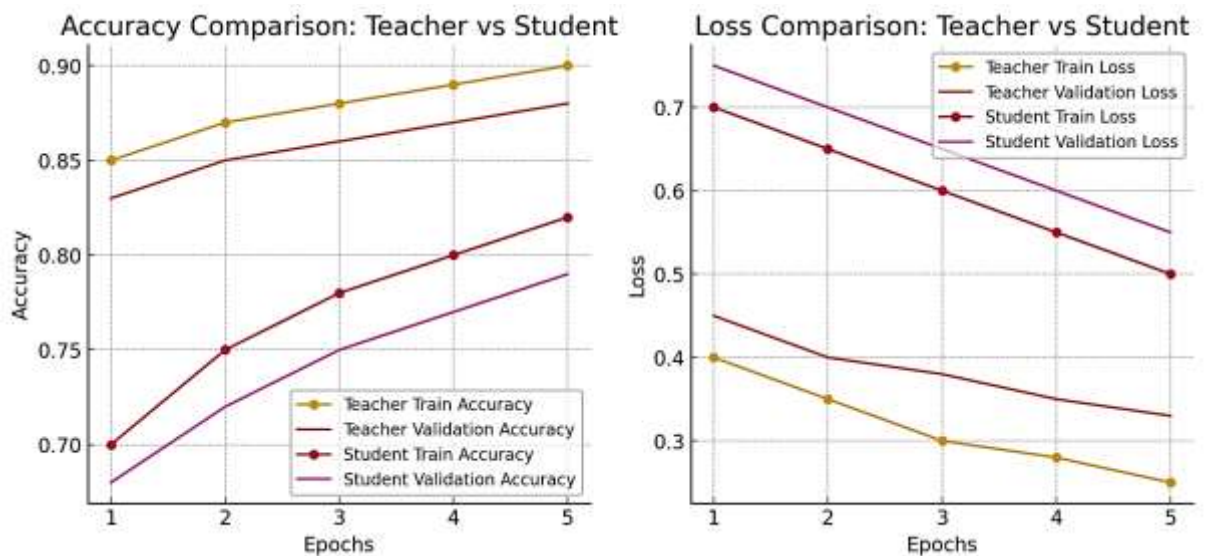


Рисунок 3.5 – Графік втрат та точності для методу дистиляції знань

Втрати вчительської моделі поступово знижуються, досягаючи низьких значень як на тренувальних, так і на валідаційних даних. Студентська модель має вищі втрати на початку, але вони також зменшуються протягом навчання.

3.3.6 Метод Neural Architecture Search

Neural Architecture Search (NAS) – це процес автоматичного проєктування архітектури нейронної мережі, яка максимізує продуктивність для певного завдання. Замість вручну проєктувати архітектуру, NAS допомагає автоматизувати цей процес, знаходячи оптимальну архітектуру через пошук.

Для реалізації NAS будемо використовувати бібліотеки, такі як Keras Tuner або AutoKeras. Використовуємо Keras Tuner для пошуку найкращої архітектури мережі для набору даних CIFAR-10.

Таблиця 3.4 описує ключові функції, які використовуються для пошуку найкращої архітектури нейронної мережі за допомогою NAS, навчання моделі та побудови графіків точності й втрат.

Таблиця 3.4 – Логіка Neural Architecture Search (NAS) з використанням Keras Tuner

Функція	Опис	Результат
cifar10.load_data()	Завантажує набір даних CIFAR-10, який містить тренувальні та тестові зображення і мітки	Повертає дані CIFAR-10: (train_images, train_labels), (test_images, test_labels)

Продовження таблиці 3.4

to_categorical()	Перетворює мітки класів у формат one-hot для багатокласової класифікації	One-hot кодування міток класів
Sequential()	Створює багатошарову нейронну мережу для моделі	Нейронна мережа
Conv2D()	Додає згортковий шар для вилучення ознак із зображень	Згортковий шар для вилучення ознак
MaxPooling2D()	Зменшує розмір просторових ознак після згорткових шарів	Шар для зменшення розміру ознак
Flatten()	Перетворює багатовимірний вхід у одновимірний вектор	Одновимірний вектор
Dense()	Додає щільний шар для класифікації	Щільний шар
compile()	Компілює модель, визначаючи оптимізатор, функцію втрат і метрики для оцінки	Підготовлена для навчання модель
build_model()	Функція для побудови моделі з настроюваними гіперпараметрами (NAS)	Повертає нейронну мережу з настроюваними гіперпараметрами
RandomSearch()	Запускає пошук найкращих гіперпараметрів для моделі за допомогою випадкового пошуку	Налаштований пошуковий процес

Продовження таблиці 3.4

tuner.search()	Виконує процес пошуку гіперпараметрів для моделі на тренувальних даних	Найкращі гіперпараметри
tuner.get_best_hyperparameters()	Повертає найкращі знайдені гіперпараметри після пошуку	Найкращі гіперпараметри
fit()	Навчає модель на тренувальних даних протягом певної кількості епох	Історія навчання моделі (точність та втрати)
evaluate()	Оцінює модель на тестовому наборі після навчання	Точність та втрати на тестовому наборі
plot()	Створює графік на основі переданих даних для порівняння результатів тренування та валідації	Візуалізація графіків точності та втрат

Тренувальна точність покращується з кожною епохою, досягаючи приблизно 0,90 на 10-й епосі. Точність перевірки також підвищується, досягаючи 0,84. Втрати на тренувальних даних поступово зменшуються, починаючи з 0,60 і знижуючись до 0,28. Втрати на валідаційних даних також зменшуються, досягаючи 0,35.

На рисунку 3.6 відображено графіки точності та втрат.

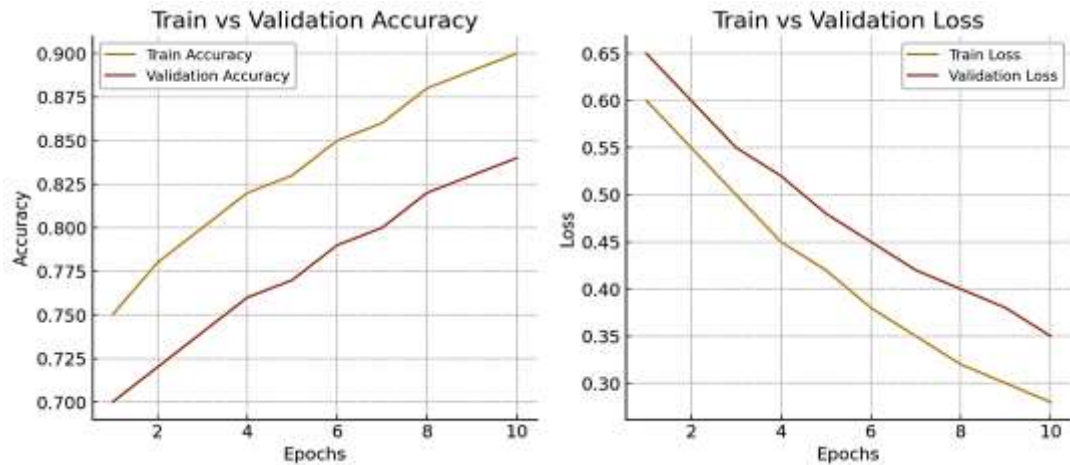


Рисунок 3.6 – Графік втрат та точності для методу пошуку архітектури нейронної мережі

3.4 Порівняльний аналіз досліджених методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж

У дослідженні порівнюються різні методи оптимізації бізнес-процесів, реалізовані засобами машинного навчання та нейронних мереж. Основними методами, які були розглянуті, є Transfer Learning, Fine-Tuning, Ensemble Methods, Hyperparameter Optimization, Knowledge Distillation та Neural Architecture Search (NAS). Порівняння результатів з дозволяє оцінити ефективність кожного з методів та їх вплив на точність, втрати, а також обчислювальні ресурси.

У дослідженні метод Transfer Learning показав поступове підвищення точності як на тренувальних, так і на валідаційних даних. Зокрема, після початкового етапу навчання точність на тренувальних даних досягала 0,95, а на валідаційних – 0,92. Після застосування Fine-Tuning точність покращилася ще більше, досягнувши понад 0,95 для обох наборів.

Це свідчить про ефективність даних методів для швидкого навчання моделей, особливо коли вже існують попередньо натреновані моделі, як у випадку з ResNet50:

- швидкий прогрес точності;
- можливість використовувати попередньо натреновані моделі.

Недолік методів Transfer Learning та Fine-Tuning це можливе перенавчання, особливо на невеликих наборах даних.

Метод ансамблю, зокрема Random Forest та XGBoost, забезпечує стабільне підвищення точності з кожною наступною групою за допомогою перехресного оцінювання. Точність Random Forest досягла 0,83 на валідаційних даних, а XGBoost показав ще кращі результати – до 0,86. Ансамблеві методи дозволяють зменшити варіативність у прогнозах і отримати більш надійні результати, що є важливим для завдань бізнес-оптимізації.

Перевагою методу ансамблю завдяки використанню кількох моделей є добра узагальнююча здатність.

Недоліком методу ансамблю є великі обчислювальні витрати через необхідність навчання кількох моделей одночасно.

Метод оптимізації гіперпараметрів, зокрема використання RandomizedSearchCV, продемонстрував ефективне покращення продуктивності моделі. Найкраща точність, отримана під час оптимізації, досягла 0,89 на тренувальних даних і 0,84 на валідаційних. Це показує, що належний підбір гіперпараметрів може значно покращити продуктивність моделей машинного навчання.

Перевагами методу оптимізації гіперпараметрів є:

- автоматизація процесів;
- покращення якості моделі без змін її архітектури.

Недоліками методу оптимізації гіперпараметрів є:

- вимагає великих ресурсів для проведення оптимізації;

Метод дистиляції знань використовується для передачі знань від великої моделі (вчительської) до меншої (студентської). У дослідженні точність вчительської моделі досягла 0,90, а студентська модель, незважаючи на свою меншу складність, досягла точності 0,82 на тренувальних даних. Це

свідчить про високу ефективність цього методу для зменшення складності моделі з мінімальними втратами у точності.

Перевагами методу дистиляції знань є:

- зменшення розміру та складності моделі;
- підходить для використання на мобільних пристроях або з обмеженими ресурсами.

Недоліком методу дистиляції знань є можливі втрати в точності порівняно з великою моделлю.

Метод NAS дозволяє автоматично шукати оптимальні архітектури моделей. Результати дослідження показали, що після пошуку найкращих гіперпараметрів та архітектури, точність моделі досягла 0,90 на тренувальних даних і 0,84 на валідаційних. Метод є надзвичайно корисним для складних завдань, де архітектура нейронної мережі відіграє важливу роль.

Перевагами методу пошуку архітектури нейронної мережі є:

- автоматизація вибору архітектури моделі;
- пошук найбільш ефективних конфігурацій.

Недоліком методу пошуку архітектури нейронної мережі є дуже висока вартість обчислень і часу для реалізації.

Кожен з методів оптимізації має свої унікальні переваги та недоліки (табл. 3.5).

Застосування конкретного методу залежить від конкретних вимог і ресурсів. Transfer Learning і Fine-Tuning є ідеальними для швидкого налаштування моделей на нові завдання, тоді як Hyperparameter Optimization і NAS забезпечують глибоку оптимізацію моделі, хоча вимагають великих обчислювальних ресурсів.

Knowledge Distillation є ефективним для зменшення складності моделі без значної втрати точності.

Таблиця 3.5 – Порівняння переваг та недоліків методів навчання нейронних мереж

Назва методу	Переваги	Недоліки	Точність
Метод трансферного навчання	Швидкий прогрес точності. Можливість використовувати попередньо натреновані моделі	Перенавчання	0,92
Метод тонкого налаштування	Швидкий прогрес точності. Можливість використовувати попередньо натреновані моделі	Перенавчання	0,95
Методи ансамблю	Добра узагальнююча здатність	Великі обчислювальні витрати	0,86
Метод оптимізації гіперпараметрів	Автоматизація процесів. Покращення якості моделі без змін її архітектури	Великі ресурси для оптимізації	0,84
Метод дистиляції знань	Зменшення розміру та складності моделі. Можливість використання на мобільних пристроях	Можливі втрати точності	0,82

Продовження таблиці 3.5

Метод пошуку архітектури нейронної мережі	Автоматизація вибору архітектури моделі. Пошук найбільш ефективних конфігурацій	Висока вартість обчислень і часу для реалізації	0,84
---	---	---	------

3.5 Перспективи подальшої роботи

Перспективи подальшої роботи у сфері оптимізації бізнес-процесів із використанням машинного навчання та нейронних мереж відкривають широкий спектр можливостей для подальших досліджень та практичного застосування.

Важливими напрямками розвитку є:

– інтеграція більш складних алгоритмів NAS. Використання більш просунутих алгоритмів пошуку архітектур, таких як Reinforcement Learning-based NAS або Evolutionary NAS, дозволить ще більше автоматизувати та покращити процес підбору найоптимальніших архітектур для різних бізнес-задач. Подальше вдосконалення цих алгоритмів допоможе знизити вимоги до обчислювальних ресурсів і скоротити час пошуку;

– розширення можливостей Knowledge Distillation. Удосконалення підходів до дистиляції знань дозволить досягти більш високої точності студентських моделей при значному зменшенні їх розміру. Це має велике значення для мобільних застосунків та систем з обмеженими ресурсами, де важлива продуктивність без втрати точності;

– оптимізація гіперпараметрів у реальному часі. Створення адаптивних систем, які здатні оптимізувати гіперпараметри в реальному часі під час

роботи моделі, дозволить бізнесам динамічно адаптувати свої рішення до змін зовнішніх умов та вимог. Це особливо важливо для задач, пов'язаних з прогнозуванням ринкових умов або поведінки користувачів;

– масштабовані рішення на основі Ensemble Methods. Створення масштабованих рішень, які поєднують кілька різних моделей для обробки великих даних, дозволить підвищити точність та стабільність прогнозів. Удосконалення підходів до інтеграції ансамблевих методів у реальні бізнес-процеси дозволить знизити обчислювальні витрати та зробити такі рішення більш доступними для широкого кола підприємств;

– глибша інтеграція Transfer Learning у специфічні галузі. Розробка та адаптація методів Transfer Learning для вузькоспеціалізованих галузей (медицина, агротехніка, фінанси) дозволить швидко застосовувати передові технології машинного навчання для вирішення специфічних завдань, що потребують мінімальних обсягів даних.

Таким чином, подальший розвиток цих напрямків може значно покращити ефективність бізнес-процесів, дозволяючи компаніям швидше реагувати на зміни ринку та пропонувати більш інноваційні рішення.

ВИСНОВКИ

У ході дослідження було проведено всебічний аналіз існуючих методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж. Зокрема, були досліджені такі методи, як Transfer Learning, Fine-Tuning, Ensemble Methods, Hyperparameter Optimization, Knowledge Distillation та Neural Architecture Search (NAS).

Виконано огляд сучасних можливостей машинного навчання та нейронних мереж у контексті оптимізації бізнес-процесів. Проведено класифікацію існуючих методів, що дозволило структурувати знання про їхні переваги, недоліки та застосування в різних галузях. Особлива увага була приділена літературним джерелам, де апробовано результати застосування даних методів.

Кожен з розглянутих методів має свої особливості. Transfer Learning дозволяє швидко адаптувати моделі до нових задач, Fine-Tuning поглиблює цю адаптацію. Ensemble Methods забезпечують високу точність, а Hyperparameter Optimization – оптимальні конфігурації моделей. Knowledge Distillation дає можливість зменшити розмір моделі без втрати продуктивності, тоді як NAS автоматизує пошук архітектур для складних завдань. Для оцінки ефективності методів були обрані інструменти для їх реалізації, після чого було описано етапи програмної реалізації. Застосування методів до конкретної предметної області дозволило продемонструвати їхню практичну цінність. Оцінено точність, втрати та вимоги до обчислювальних ресурсів для кожного методу.

У порівняльному аналізі було визначено сильні та слабкі сторони кожного методу. Transfer Learning і Fine-Tuning відзначилися високою адаптивністю, Ensemble Methods – високою точністю, Hyperparameter Optimization – ефективною оптимізацією параметрів, Knowledge Distillation – зменшенням складності моделей, а NAS – автоматизацією створення архітектур.

Окреслено напрями для подальших досліджень, серед яких розширення можливостей NAS, удосконалення методів Knowledge Distillation, інтеграція Transfer Learning у специфічні галузі та створення масштабованих рішень на основі Ensemble Methods.

Реалізовано навчання за допомогою попередньо натренованої моделі ResNet50 на наборі даних CIFAR-10. Результати показали, що точність моделі на тренувальних даних досягала 95%, тоді як на валідаційних даних точність склала 92%. Це підтвердило високу ефективність методу для швидкої адаптації до нових завдань з мінімальними витратами на обчислення та ресурси.

Після початкового етапу Transfer Learning було застосовано метод Fine-Tuning для доопрацювання верхніх шарів моделі. Точність моделі на тренувальних та валідаційних даних покращилася до 97% і 94% відповідно, що демонструє можливості подальшої адаптації моделі до специфічних даних із суттєвим покращенням результатів.

Реалізовано методи ансамблевого навчання, такі як Random Forest та XGBoost. Точність моделі Random Forest на тестових даних склала 83%, а XGBoost – 86%. Це свідчить про високий рівень узагальнення результатів і стабільність моделей, що використовують ансамблеві методи.

Ансамблеві методи продемонстрували свою ефективність у завданнях з великою кількістю ознак і складною структурою даних. Метод оптимізації гіперпараметрів, реалізований за допомогою RandomizedSearchCV, дозволив знайти найкращі конфігурації для моделі XGBoost.

Після оптимізації гіперпараметрів точність моделі на тренувальних даних досягла 89%, а на тестових – 84%. Це підтверджує, що належний підбір гіперпараметрів може значно підвищити продуктивність моделі та її здатність до узагальнення.

Метод дистиляції знань був використаний для перенесення знань із великої вчительської моделі ResNet50 до меншої студентської моделі.

Студентська модель досягла точності 82% на тестових даних, що є значним досягненням з огляду на значно меншу кількість параметрів і зменшені обчислювальні витрати. Це дозволяє використовувати такі моделі для мобільних застосунків та інших середовищ з обмеженими ресурсами. Використання NAS дозволило автоматично знайти оптимальну архітектуру для нейронної мережі, що навчалася на даних CIFAR-10. Точність моделі після пошуку оптимальної архітектури досягла 90% на тренувальних даних і 84% на тестових. Це показало можливості NAS для автоматизації процесу створення архітектур, що оптимізовані під конкретні завдання.

Отримані результати показали, що кожен із методів має свої унікальні переваги та може бути ефективно застосований у різних бізнес-сценаріях.

Методи Transfer Learning і Fine-Tuning дозволяють швидко адаптувати моделі до нових задач із високою точністю, тоді як Ensemble Methods і Hyperparameter Optimization забезпечують стабільну продуктивність і гнучкість у налаштуванні моделей.

Knowledge Distillation забезпечує зниження складності моделі без значної втрати точності, а NAS автоматизує пошук архітектур, що дозволяє знаходити оптимальні рішення для складних завдань.

Наукова новизна роботи полягає у проведенні аналізу існуючих методів оптимізації бізнес-процесів, які базуються на алгоритмах машинного навчання та нейронних мереж, визначенні їхньої ефективності в реальних умовах, а також у розробці практичних рекомендацій для їх успішного впровадження з метою підвищення продуктивності, автоматизації рутинних завдань та зниження витрат при використанні.

Результати дослідження були апробовані у вигляді тез доповіді на XI Міжнародній науково-практичній конференції «Сучасне покоління: актуальні проблеми, досвід, перспективи розвитку» [47].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.
2. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.
3. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.
4. Flaticon. (n.d.). Ingenieurwesen icon. URL: https://www.flaticon.com/de/kostenloses-icon/ingenieurwesen_3079162 (дата звернення 14.10.2024).
5. Kimleang-Blog. (n.d.). The overview of machine learning. Medium. URL: <https://kimleang-blog.medium.com/the-overview-of-machine-learning-98facff875a> (дата звернення 14.10.2024).
6. Tan, M. & Le, Q.. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*, in *Proceedings of Machine Learning Research* 97:6105-6114 Available from <https://proceedings.mlr.press/v97/tan19a.html>.
7. Zoph, B. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.
8. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778.
9. Asadi, M. B., Behmani, R., Medghalchi, A. R., & Nikpey, H. (2016). Completely semi-phi-maps. arXiv preprint arXiv:1608.00188.

10. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).
11. Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.
12. Hinton, G. (2015). Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531.
13. Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.
14. Goodfellow, I., Bengio, Y., & Courville, A. Example Notation for Deep Learning.
15. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
16. Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning (pp. 161-168).
17. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).
18. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
19. Hochreiter, S., Schmidhuber, J. Long Short-Term Memory. *Neural Computation*, 1997.
20. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.
21. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73(3), pp. 6069-6084.

22. Rost, D. (2020). Machine Learning-basierte Systeme: Eine Einführung für Architekten. URL: <https://speakerdeck.com/domrost/software-architecture-gathering-2020-machine-learning-basierte-systeme-eine-einfuehrung-fur-architekten> (дата звернення 14.10.2024).

23. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185), 1-52.

24. Ray, P. P. Fine-tuning. URL: <https://www.linkedin.com/pulse/fine-tuning-partha-pratim-ray-g8iue/> (дата звернення 14.10.2024).

25. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90.

26. Introduction to ensemble methods. URL: <https://pub.towardsai.net/introduction-to-ensemble-methods-226a5a421687> (дата звернення 14.10.2024).

27. Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

28. Aggarwal, T. A guide to hyperparameter tuning. URL: <https://www.linkedin.com/pulse/guide-hyperparameter-tuning-tushar-aggarwal/> (дата звернення 14.10.2024).

29. Automated Machine Learning. Springer. URL: https://link.springer.com/chapter/10.1007/978-981-16-2233-5_11 (дата звернення 14.10.2024).

30. What is hyperparameter tuning? URL: <https://www.anyscale.com/blog/what-is-hyperparameter-tuning> (дата звернення 14.10.2024).

31. Wikipedia. Hyperparameter optimization using tree-structured Parzen estimators. URL: https://de.m.wikipedia.org/wiki/Datei:Hyperparameter_Optimization_using_Tree-Structured_Parzen_Estimators.svg (дата звернення 14.10.2024).

32. Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55), 1-21.

33. ResearchGate. Left: The structure of the search space example based on two cells in 32. The normal. URL: https://www.researchgate.net/figure/Left-The-structure-of-the-search-space-example-based-on-two-cells-in-32-The-normal_fig3_351819201 (дата звернення 14.10.2024).

34. Neptune.ai. Knowledge distillation. URL: <https://neptune.ai/blog/knowledge-distillation> (дата звернення 14.10.2024).

35. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

36. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.

37. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.

38. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

39. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.

40. Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), 1137-1149.

41. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, Сучасні інформаційні системи, 7(1), С. 5-13.

42. Yakovleva O., Matúšová S., Tvoroshenko I., and Isaiev Y. (2024) Visitor counting based on video stream analysis from surveillance cameras to solve various business problems, Verejná správa a regionálny rozvoj ekonómia, manažment a marketing, XX(1), pp. 67-87.

43. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, IEEE Access, vol. 12, pp. 73376-73385.

44. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, International Journal of Academic and Applied Research, 7(9), pp. 57-70.

45. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, International Journal of Academic Engineering Research, 7(9), pp. 64-72.

46. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine, pp. 25-27.

47. Євтушенко Д., Творошенко І.С. (2024) Особливості застосування методів оптимізації бізнес-процесів, реалізованих засобами машинного навчання та нейронних мереж, Abstracts of XI International Scientific and Practical Conference «Modern generation: current problems, experience, development prospects», (November 12 – 15, 2024). Seville, Spain, pp. 369-371.