

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ
КАФЕДРА КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ КОМПЛЕКСІВ**

Кваліфікаційна наукова праця
на правах рукопису

Бориндо Ілля Олександрович

УДК 004.93

ДИСЕРТАЦІЯ

**СТРУКТУРНО-ПАРАМЕТРИЧНИЙ СИНТЕЗ ЗГОРТКОВИХ НЕЙРОННИХ
МЕРЕЖ В ЗАДАЧАХ АВТОМАТИЗАЦІЇ ВИРОБНИЧИХ ПРОЦЕСІВ**

151 «Автоматизація та комп'ютерно-інтегровані технології»

Дисертація на здобуття наукового ступеня кандидата
технічних наук

Дисертація містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання на відповідне
джерело

Бориндо І. О.

Науковий керівник
Синеглазов Віктор Михайлович
доктор технічних наук, професор

Київ 2025

АНОТАЦІЯ

Бориндо І. О. Структурно-параметричний синтез згорткових нейронних мереж в задачах автоматизації виробничих процесів.

Дисертація на здобуття наукового ступеня кандидата технічних наук (доктора філософії) за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології». Київський Авіаційний Інститут МОН України, Київ, 2025.

У дисертаційній роботі досліджено та розроблено методи структурно-параметричного синтезу згорткових нейронних мереж (ЗНМ) для задач автоматизації виробничих процесів. Обґрунтовано використання багатокритеріальних генетичних алгоритмів (БКГА) для оптимізації архітектури нейронних мереж з метою підвищення їх ефективності, точності та обчислювальної продуктивності. Запропоновано нові підходи до параметричної оптимізації, які дозволяють адаптувати нейромережеві моделі до конкретних виробничих завдань.

Досліджено сучасні топологічні варіанти ЗНМ, механізми уваги та просторово-каналні реконструктивні методи, що сприяють покращенню аналізу графічних даних у реальному часі. Проведено порівняння запропонованих методів із існуючими підходами автоматизованого проектування нейромереж, що продемонструвало їх переваги у точності класифікації, швидкості обробки та ефективності ресурсного використання.

Розглянуто можливості застосування запропонованих методів у сферах віртуальної та доповненої реальності, що дозволяє покращити імерсивність та інтерактивність систем візуалізації, а також зменшити обчислювальні витрати на їх реалізацію. Розроблені алгоритмічні рішення реалізовано у вигляді програмного забезпечення з використанням платформ TensorFlow і Keras, що підтверджує їхню практичну значущість та можливість інтеграції у реальні виробничі процеси.

Розділ 1. Проведено аналіз існуючих архітектур згорткових нейронних мереж, розглянуто їхні топологічні особливості та основні підходи до

оптимізації. Окреслено основні проблеми структурного синтезу ЗНМ та перспективні напрями їхнього вдосконалення.

Розділ 2. Досліджено можливості інтеграції згорткових нейронних мереж у технології віртуальної (VR) та доповненої реальності (AR). Проаналізовано методи покращення імерсивності, інтерактивності та продуктивності VR/AR-систем за допомогою нейронних мереж. Визначено основні проблеми їхнього впровадження та запропоновано шляхи їх подолання.

Розділ 3. Визначено основні структурні блоки, що впливають на продуктивність ЗНМ. Досліджено математичні моделі та алгоритмічні рішення, що дозволяють покращити параметричну адаптацію нейронних мереж. Запропоновано підхід до автоматизованого проектування оптимальних згорткових нейронних мереж, заснований на багатокритеріальних еволюційних алгоритмах. Обґрунтовано критерії оптимізації та розроблено алгоритмічні рішення, що дозволяють підвищити ефективність розпізнавання та обробки графічних даних у реальному часі.

Розділ 4. Описано та запропоновано алгоритмічне та програмне впровадження алгоритму структурно-параметричного синтезу згорткових нейронних мереж. Було отримано та протестовано синтезовану модель мережі та оцінені її якісні характеристики.

Результати дослідження апробовані на міжнародних конференціях та опубліковані у провідних наукових виданнях. Запропоновані методи та алгоритми можуть бути використані для подальшого розвитку систем автоматизованого проектування нейромереж та розширення їх застосування у промислових та інформаційних технологіях.

Ключові слова: нейронні мережі, згорткові нейронні мережі, розпізнавання образів, багатокритеріальна оптимізація, оптимізація, умови оптимальності, структурно-параметричний синтез, генетичні алгоритми, віртуальна реальність, інтелектуальна обробка інформації.

Список публікацій здобувача:

Статті у виданнях іноземних держав:

1. Illia Boryndo, Viktor Sineglazov (2022). “The Optimal Choice of Hybrid Convolutional Neural Network Components”, American Journal of Neural Networks and Applications, 8(2), 12-16.
<https://doi.org/10.11648/j.ajnna.20220802.11>

Статті у наукових фахових виданнях України:

2. “Intelligent recognition and integration of graphical elements into virtual surrounding within augmented reality using hybrid convolutional neural networks”, Sineglazov Viktor, I. Boryndo, 2023, Artificial Intelligence. 28. 74-79. 10.15407/jai2023.01.074.
3. “Application of a Multicriteria Genetic Algorithm for Structural Parametric Synthesis of Convolutional Neural Networks”, Artificial Intelligence. 29(4), 2024, 106-114., DOI: 10.15407/jai2024.04.106
4. “Application of Neural Networks for Virtual and Augmented Reality”, Sineglazov Viktor, I. Boryndo, 2022, Electronics and Control Systems. 4. 51-57. 10.18372/1990-5548.74.17296.
5. “Hand Gestures Recognition and Tracking Within Virtual Reality using Hybrid Convolutional Neural Networks”, Sineglazov Viktor, I. Boryndo, 2022, Electronics and Control Systems. 2. 32-37. 10.18372/1990-5548.72.16940.

Матеріали та тези наукових доповідей:

1. “Multicriteria optimization of hybrid convolutional neural network structural synthesis using evolutionary algorithms”, Boryndo, I., Sineglazov, V., Zgurovsky, M.Z., CEUR Workshop Proceedings, 2024, 3790, pp. 318–330
2. "Visual Navigation Systems Implementation Using Hybrid Convolutional Neural Networks", Sineglazov, V., Boryndo, I., 2023 IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control, MSNMC 2023 - Proceedings, 2023, pp. 155–158

3. "Intelligence system for emotional facial state estimation during inspection control", Sineglazov, V., Pantyeyev, R., Boryndo, I., CEUR Workshop Proceedings, 2019, 2683, pp. 25–29
4. “Virtual Reality Systems Integration With Neural Networks For Immersivity Enhancement”, Illia Boryndo, Viktor Sineglazov, "Information technologies and computer modelling", 2023, Ivano-Frankivsk, ISBN 978-617-8128-23-4, pp. 87
5. “Topology Analysis of Hybrid Convolutional Neural Networks”, Viktor Sineglazov, Olena Chumachenko and Illia Boryndo, “Information systems and technologies”, 2021, ISBN 978-617-7519-59-0, pp. 53-56
6. “Multicriteria optimization for structural parametric synthesis of convolutional neural networks”, Illia Boryndo, Viktor Sineglazov, "Information technologies and computer modelling" proceedings of the international scientific conference 2024, may, 21st to 24th Ivano-Frankivsk, ISBN 978-966-640-560-2, pp. 227-230
7. “Application of deep learning hybrid convolutional neural networks in visual navigation systems”, Illia Boryndo

ABSTRACT

Boryndo I. O. Structural-Parametric Synthesis of Convolutional Neural Networks in the Tasks of Production Process Automation. Dissertation for the degree of Candidate of Technical Sciences (Doctor of Philosophy) in the specialty 151 "Automation and Computer-Integrated Technologies". Kyiv Aviation Institute, Ministry of Education and Science of Ukraine, Kyiv, 2025.

This dissertation explores and develops methods for the structural-parametric synthesis of convolutional neural networks (CNNs) for production process automation. The use of multi-criteria genetic algorithms (MCGA) for optimizing neural network architectures is substantiated to enhance their efficiency, accuracy, and computational performance. New approaches to

parametric optimization are proposed, enabling the adaptation of neural network models to specific industrial tasks.

Modern topological variants of CNNs, attention mechanisms, and spatial-channel reconstruction methods have been analyzed to improve real-time graphical data analysis. A comparative study of the proposed methods with existing automated neural network design approaches has demonstrated their advantages in classification accuracy, processing speed, and resource efficiency.

The possibilities of applying the proposed methods in virtual and augmented reality technologies are considered, allowing for improved immersion and interactivity in visualization systems while reducing computational costs. The developed algorithmic solutions have been implemented as software using the TensorFlow and Keras platforms, confirming their practical significance and potential integration into real-world production processes.

Chapter 1. A review of existing CNN architectures, their topological features, and primary optimization approaches is conducted. Key issues in structural synthesis and future directions for CNN improvement are outlined.

Chapter 2. The integration of convolutional neural networks into virtual (VR) and augmented reality (AR) technologies is examined. Methods for enhancing immersion, interactivity, and VR/AR system performance through neural networks are analyzed. Major implementation challenges and possible solutions are identified.

Chapter 3. The main structural components affecting CNN performance are identified. Mathematical models and algorithmic solutions for improving parametric adaptation are studied. An approach to automated design based on multi-criteria evolutionary algorithms is proposed, with justification for optimization criteria and algorithmic solutions that enhance real-time image recognition and processing efficiency.

Chapter 4. The algorithmic and software implementation of the structural-parametric synthesis algorithm for CNNs is described and proposed. The

synthesized network model was obtained, tested, and its qualitative characteristics were evaluated.

The research results have been presented at international conferences and published in leading scientific journals. The proposed methods and algorithms can be used for the further development of automated neural network design systems and the expansion of their applications in industrial and information technologies.

Keywords: neural networks, convolutional neural networks, pattern recognition, multicriteria optimization, optimization, optimality conditions, structural-parametric synthesis, genetic algorithms, virtual reality, intelligent information processing.

Зміст

Вступ.....	1
1. ОГЛЯД ТА АНАЛІЗ СУЧАСНИХ ТОПОЛОГІЧНИХ РІЗНОВИДІВ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ТА АЛГОРИТМІВ СТРУКТУРНОГО СИНТЕЗУ.....	14
1.1. Аналіз сучасних архітектурних рішень при проектуванні згорткових нейронних мереж.....	14
1.2. Аналіз топологічних ознак та функціональних особливостей сучасних нейронних мереж	19
1.2.1. Просторово-канальна реконструкція згортки.....	21
1.2.2. Щільно-зв'язний механізм уваги.....	22
1.2.3. Підсилення каналів.....	24
1.2.4. Ітераційний аналіз карт ознак.....	24
1.2.5. Згорткові нейронні мережі на основі механізмів уваги.....	25
1.3. Структурний аналіз згорткових нейронних мереж.....	26
1.4. Структурно-параметричний синтез згорткових нейронних мереж.....	31
1.4.1. Постановка задачі структурно-параметричного синтезу.....	31
1.4.2. Сучасні підходи до структурного синтезу згорткових нейронних мереж	32
1.4.3. Багатокритеріальні генетичні алгоритми (NSGA, MOEA, SPEA3).....	36
1.4.4. Визначення параметричного синтезу згорткових нейронних мереж....	48
2. ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В СИСТЕМАХ ВІРТУАЛЬНОЇ ТА ДОПОВНЕНОЇ РЕАЛЬНОСТЕЙ.....	49
2.1. Віртуальна реальність.....	49
2.2. Доповнена реальність.....	51
2.3. Застосування згорткових нейронних мереж для віртуальної та доповненої реальності.....	53
2.4. Переваги використання згорткових нейронних мереж в технологіях віртуальної та доповненої реальності.....	57

2.4.1. Покращення імерсійності віртуальної реальності.....	57
2.4.2. Покращення інтерактивності віртуальної реальності.....	59
2.4.3. Зменшення витрат при розробці технологій віртуальної реальності....	60
2.5. Проблеми згорткових нейронних мереж для віртуальної реальності...	61
2.5.1. Обмежена доступність даних.....	61
2.5.2. Високі вимоги до обчислювальної потужності.....	62
2.6. Шляхи застосування нейронних мереж для технології віртуальної реальності.....	63
2.6.1. Генерування та масштабування високоякісних текстур для графічних двигунів віртуальної реальності.....	67
2.6.2. Використання нейронних мереж для розрахунку та моделювання поведінки освітлення в умовах віртуальної реальності.....	71
2.6.3. Алгоритмія пошуку шляху об'єктів віртуальної реальності в реальному часі використовуючи нейронні мережі.....	75
2.6.4. Використання згорткових нейронних мереж для систем Unity3D. Інтерактивна віртуальна реальність в реальному часі.....	79
2.7. Архітектурні особливості систем віртуальної реальності а впровадження нейронних мереж в межах її структурних елементів.....	81
3. СТРУКТУРНО-ПАРАМЕТРИЧНИЙ СИНТЕЗ ОПТИМАЛЬНОЇ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ЗАСТОСУВАННЯ В ВИРОБНИЧИХ ПРОЦЕСАХ З ВИКОРИСТАННЯМ БАГАТОКРИТЕРІАЛЬНИХ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ.....	85
3.1. Визначення критеріїв оптимізації для налаштування генетичного алгоритму.....	86
3.2. Запропонований багатокритеріальний генетичний алгоритм оптимізований для ЗНМ.....	88
3.3. Аналіз структурних елементів та їх математичних моделей для використання в процесі структурно-параметричного синтезу оптимальних нейронних мереж.....	104
3.3.1. Застосування функції «Spatial and Channel Reconstruction Convolution»	

.....	105
3.3.2. «Dense Residual Attention-based» механізм.....	109
3.3.3. Контекстуальна ЗНМ.....	112
3.3.4. Згорткова нейронна мережа з неперервною симетрією.....	113
3.3.5. Допоміжні структурні блоки ЗНМ.....	114
4. ПРАКТИЧНЕ ВПРОВАДЖЕННЯ БАГАТОКРИТЕРІАЛЬНОГО СТРУКТУРНО-ПАРАМЕТРИЧНОГО СИНТЕЗУ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ІДЕНТИФІКАЦІЇ ЖЕСТІВ КОРИСТУВАЧА В РЕАЛЬНОМУ ЧАСІ.....	116
4.1. Програмне впровадження запропонованого алгоритму для структурно- параметричного синтезу.....	116
4.2. Практичне застосування запропонованого генетичного алгоритму для структурно-параметричного синтезу згорткової нейронної мережі.....	118
Висновок.....	124
Література.....	127
Додатки.....	131

Перелік умовних позначень

ЗНМ – згорткова нейронна мережа

ГЗНМ – гібридна згорткова нейронна мережа

ГА – генетичний алгоритм

БКГА – багатокритеріальний генетичний алгоритм

ЕА – еволюційний алгоритм

ГП – генетичне програмування

ПНМ – повторювана нейронна мережа

ЕС – еволюційні стратегії

ВР – віртуальна реальність

ДР – доповнена реальність

Вступ

У сучасному світі існує багато сфер, в яких системи аналізу та діагностики графічних даних та їх точність відіграють важливу роль у достатності результатів. Це може бути географічна, урбо-статистична чи медична діагностика, тощо. При аналізі і обробці таких даних людський фактор має серйозний вплив на фінальні результати дослідження та попри це не існує надійних альтернатив для їх автоматичної обробки та аналізу.

Метою даної роботи є розроблення інтелектуальної гібридної системи за допомогою алгоритмів структурно-параметричного синтезу, її налаштування та використання для роботи з графічними даними різної складності.

Об'єктом дослідження є гібридні згорткові нейронні мережі.

Предметом дослідження є методи та алгоритми структурно-параметричного синтезу гібридних згорткових нейронних мереж.

Для досягнення цієї мети необхідно вирішити завдання:

- Розробити стійкі до інваріантних перетворень моделі для представлення об'єктів за допомогою набору їх ознак.
- Розробити метод оптимального вибору множини згорткових нейронних мереж або модулів які включаються до складу гібридної нейронної мережі. Розробити багатокритеріальний алгоритм оптимізації який використовується для розв'язання задачі структурно-параметричного синтезу. Побудувати реальну модель системи використовуючи засоби комп'ютерної розробки та програмні пакети “Tensorflow” та “Keras”. Реалізувати алгоритмічний комплекс у вигляді комп'ютерної програми.
- Оцінити результативність розробленої інтелектуальної системи та критерії вилучення ознак графічних даних.
- Оцінити ефективність розробленої системи в порівнянні з сучасними альтернативними методами розпізнавання та аналізу зображень різних типів та конфігурацій.

Предметом дослідження є різні типології згорткових нейронних мереж, алгоритми їх конфігурації та генерації структурного профілю, математичні моделі та алгоритми розпізнавання ознак. Для вирішення поставлених завдань використовувалися методи комп'ютерного зору, теорія оптимізації, теорія штучних нейронних мереж та ймовірнісні моделі.

Актуальність теми

У сучасних умовах цифровізації та стрімкого зростання обсягів даних, завдання виборничих процесів, таких як класифікація, розпізнавання образів, прогнозування та аналіз складних структур даних, набувають дедалі більшої значущості. Згорткові нейронні мережі (Convolutional Neural Networks, ЗНМs), завдяки своїй архітектурі, орієнтованій на виявлення локальних ієрархічних ознак у вхідних даних, демонструють високу ефективність у розв'язанні цих задач. Проте розробка оптимальної структури ЗНМ для конкретного завдання залишається складним завданням через високу варіативність параметрів, що впливають на продуктивність моделі.

Зокрема, задачі виборничих процесів характеризуються специфікою оброблюваних даних, складністю алгоритмічних рішень та необхідністю забезпечення компромісу між точністю, швидкодією та обчислювальною ефективністю. У зв'язку з цим виникає потреба у структурно-параметричному синтезі ЗНМ, що передбачає систематичне визначення як архітектурних компонентів (глибини мережі, кількості згорткових шарів, типів активацій тощо), так і параметрів навчання (швидкості навчання, оптимізаторів, об'ємів даних). Такий підхід дозволяє створювати моделі, адаптовані до специфічних умов застосування та характеристик оброблюваних даних.

Актуальність дослідження зумовлена кількома ключовими факторами:

- Зростання обсягу та складності даних. Задачі виборничих процесів все частіше потребують обробки великих масивів даних із високим ступенем неоднорідності (зображення, часові ряди,

текст). Це вимагає розробки нейромереж, здатних ефективно працювати з різними форматами даних, при цьому забезпечуючи високу продуктивність.

- Проблема надмірної складності моделей. Застосування великих нейронних мереж часто призводить до переобчислювальної складності, що обмежує їх використання в реальних системах, особливо в умовах обмежених ресурсів (наприклад, у вбудованих системах чи мобільних пристроях). Синтез оптимальних моделей сприяє зниженню обчислювальних витрат.
- Недостатність універсальних підходів. Існуючі методи побудови ЗНМ здебільшого зосереджені на загальних підходах, що не враховують специфіки виборничих процесів. Розробка спеціалізованих методів синтезу ЗНМ дозволяє досягти більш високої ефективності при вирішенні конкретних задач.
- Автоматизація проектування. Ручне налаштування архітектури нейронних мереж є трудомістким і потребує значних експертних знань. Методи структурно-параметричного синтезу можуть бути основою для автоматизації цього процесу, що робить їх особливо важливими для масштабних і промислових застосувань.

Мета і завдання дослідження

Метою дослідження є розробка науково обґрунтованих методів структурно-параметричного синтезу згорткових нейронних мереж (ЗНМ) для ефективного розв'язання задач виборничих процесів, що забезпечують оптимальний баланс між точністю, швидкістю та обчислювальною ефективністю в умовах специфічних вимог до даних і обчислювальних ресурсів.

Об'єкт дослідження

Об'єктом дослідження є процеси та методи автоматизованого проектування згорткових нейронних мереж (Convolutional Neural Networks,

ЗНМ) для розв'язання задач виборничих процесів у широкому спектрі прикладних застосувань. Це включає задачі класифікації, сегментації, трекінгу об'єктів, прогнозування та обробки складних структур даних у таких галузях, як комп'ютерний зір, системи віртуальної та доповненої реальності (VR/AR), автоматизовані системи керування та інтелектуальний аналіз даних.

Процеси виборничих задач передбачають обробку, аналіз і класифікацію даних різної природи (зображень, часових рядів, сигналів), що вимагає створення високоефективних алгоритмів із використанням сучасних методів глибокого навчання. Особливу увагу приділено задачам, які характеризуються великим обсягом вхідних даних, високою динамічністю змін середовища та обмеженими ресурсами для виконання обчислень у реальному часі.

Предмет дослідження

Предметом дослідження є методи, алгоритми та підходи до структурно-параметричного синтезу згорткових нейронних мереж, які забезпечують адаптацію їхньої архітектури та параметрів до специфіки виборничих процесів із урахуванням наступних аспектів:

Архітектурні особливості ЗНМ:

Розробка оптимальної структури мережі, що включає кількість шарів, типи шарів (згорткові, пулінг, нормалізації, активації), типи активаційних функцій та розмір ядер згортки. Досліджується вплив цих характеристик на продуктивність ЗНМ для виборничих задач.

Використання генетичних алгоритмів (ГА):

Алгоритмічне забезпечення процесу пошуку оптимальної архітектури ЗНМ із застосуванням еволюційних обчислень, таких як кросинговер, мутація, селекція. Особливу увагу приділено розробці операторів генетичних алгоритмів для кодування і модифікації структур ЗНМ.

Інтеграція структурно-параметричних моделей у VR/AR системи:

Розробка підходів до адаптації синтезованих ЗНМ для обробки даних у реальному часі, таких як обробка відеопотоків, трекінг об'єктів, аналіз жестів та інтерактивна взаємодія. Досліджується здатність моделей працювати в умовах високих частот оновлення даних та обмежених апаратних ресурсів.

Застосування сучасних структурних блоків ЗНМ:

Використання готових блоків із сучасних архітектур у процесі структурно-параметричного синтезу для підвищення ефективності моделей. Аналізується, як динамічна комбінація цих блоків впливає на точність і швидкодію.

Багатокритеріальна оптимізація:

Розробка та аналіз критеріїв для одночасної оптимізації точності, обчислювальної складності, енергоспоживання та швидкодії моделей. Предметом дослідження є розробка багатокритеріальної цільової функції, яка враховує специфіку виборничих процесів.

Ефективність синтезованих моделей у задачах виборничих процесів: Оцінка точності, стійкості, адаптивності та продуктивності ЗНМ у задачах, які включають класифікацію об'єктів, розпізнавання образів, сегментацію, трекінг рухомих об'єктів та інші процеси, що вимагають високого рівня генералізації та адаптації.

Взаємозв'язок об'єкта та предмета дослідження

Об'єктом дослідження є загальні процеси створення та адаптації згорткових нейронних мереж для задач виборничих процесів, тоді як предмет дослідження конкретизує ці процеси через методи синтезу, застосування генетичних алгоритмів, використання сучасних архітектурних блоків, інтеграцію з VR/AR системами та розробку метрик для оцінки ефективності. Цей взаємозв'язок дозволяє об'єднати теоретичні та практичні аспекти синтезу ЗНМ для вирішення виборничих задач у різних прикладних контекстах.

Завдання дослідження

Для досягнення поставленої мети визначено такі основні завдання:

1. Аналіз сучасних архітектурних рішень згорткових нейронних мереж, їх функціональних та структурних особливостей.
2. Аналіз можливості інтеграції моделей ЗНМ із специфічними системами що можуть бути застосовані при виробничих процесах, наприклад системи віртуальної та доповненої реальності.
3. Аналіз сучасного стану та проблематики структурно-параметричного синтезу ЗНМ:

Провести аналіз існуючих підходів до проектування архітектур згорткових нейронних мереж, зокрема методів пошуку архітектур (NAS – Neural Architecture Search), та визначити їхні обмеження у контексті задач виборничих процесів. Особливу увагу приділити дослідженню впливу архітектурних та параметричних змін на продуктивність моделей.

4. Визначення ключових особливостей виборничих процесів: Сформулювати вимоги до моделей ЗНМ для вирішення задач виборничих процесів, враховуючи особливості оброблюваних даних (зображення, часові ряди, багатовимірні структури) та специфіку обчислювальних ресурсів (вбудовані системи, хмарні сервіси тощо).

5. Розробка методів структурно-параметричного синтезу ЗНМ: Запропонувати нові підходи до формування архітектури згорткових нейронних мереж, що дозволяють автоматизувати процес підбору параметрів та забезпечують адаптацію моделей до конкретних умов використання. Це включає визначення критеріїв оптимізації, які враховують як точність розв'язання задачі, так і ресурсні обмеження.

6. Дослідження впливу параметрів навчання на продуктивність моделей:

Дослідити взаємозв'язок між гіперпараметрами навчання

(швидкість навчання, кількість епох, розмір партії) та продуктивністю ЗНМ у виборничих задачах. Розробити рекомендації щодо їх оптимального налаштування залежно від специфіки задачі.

7. Розробка метрик для оцінки ефективності синтезованих моделей:
Сформулювати і обґрунтувати метрики для оцінювання якості синтезованих моделей, враховуючи не лише точність, але й обчислювальну ефективність, швидкодію, стійкість до переобчислення та здатність до генералізації.
8. Розробка програмного забезпечення для автоматизованого синтезу ЗНМ:
Реалізувати прототип програмної системи, що автоматизує процес синтезу ЗНМ для виборничих задач, забезпечуючи інтерактивність і можливість адаптації до нових даних і умов застосування.
9. Експериментальна перевірка запропонованих методів:
Провести серію експериментів на реальних даних для перевірки ефективності розроблених підходів до структурно-параметричного синтезу ЗНМ. Порівняти отримані результати з існуючими методами для визначення конкурентоспроможності розроблених рішень.
10. Оптимізація моделей для специфічних обчислювальних платформ:
Дослідити можливості оптимізації синтезованих ЗНМ для роботи на різних апаратних платформах, таких як графічні процесори (GPU), системи на чипі (SoC), центральні процесори (CPU) та нейронні акселератори. Розробити рекомендації щодо їх ефективного використання.

Методи дослідження

Для розв'язання задачі структурно-параметричного синтезу згорткових нейронних мереж (ЗНМ) у контексті задач виборничих процесів використовується комплекс теоретичних, експериментальних та прикладних методів. Особливу увагу приділено використанню генетичних алгоритмів (ГА) як методу оптимізації для автоматизації пошуку архітектур та параметрів ЗНМ.

Методи математичного моделювання.

Використовуються для формального опису структури згорткових нейронних мереж у вигляді математичної моделі, що включає:

- параметри мережі (кількість шарів, типи активацій, кількість фільтрів, розміри згорток);
- функцію втрат для оцінки ефективності моделі;
- цільову функцію, що враховує як точність роботи, так і обчислювальні ресурси.

Методи оптимізації.

Застосовуються для вирішення задачі пошуку оптимальної структури та параметрів ЗНМ. У цьому контексті ГА виступають як адаптивний еволюційний підхід для глобального пошуку в просторі можливих архітектур.

Теорія еволюційних обчислень.

В основі ГА лежать принципи природного добору, кросинговеру, мутацій та відбору, що дозволяє ефективно досліджувати складні та нерегулярні простори параметрів ЗНМ. Теоретичне обґрунтування цих принципів використовується для розробки відповідних операторів ГА, адаптованих до задач виборничих процесів.

Теорія ймовірностей і статистики.

Використовується для оцінки якості роботи генетичних алгоритмів, зокрема для аналізу їхньої збіжності, стабільності результатів, а також для оцінки стійкості запропонованих моделей ЗНМ.

Моделювання процесу еволюційного синтезу ЗНМ.

Реалізується через створення спеціалізованих алгоритмів ГА, що оперують з представленням архітектури ЗНМ у вигляді хромосом. Експерименти спрямовані на:

- визначення ефективності різних операторів кросинговеру та мутацій;
- вибір підходів до кодування архітектур (напр., лінійне чи деревоподібне представлення);
- оцінку продуктивності алгоритмів на задачах різної складності.

Тестування запропонованих архітектур ЗНМ на наборах даних.

Використовуються набори даних для задач виборничих процесів, таких як класифікація (CIFAR-10, ImageNet), сегментація (Pascal VOC, COCO), та інші задачі. Для кожного набору даних перевіряються:

- точність моделі;
- час навчання;
- споживання пам'яті та енергії.

Адаптація гіперпараметрів через ГА.

Використання ГА для автоматичного налаштування параметрів навчання ЗНМ (розмір партії, швидкість навчання, тип оптимізатора) та аналіз впливу цих параметрів на ефективність мережі.

Експериментальна перевірка ефективності ГА у порівнянні з іншими методами оптимізації.

Для об'єктивної оцінки розроблених підходів порівнюється їх продуктивність із традиційними методами пошуку архітектур, такими як метод випадкового пошуку (Random Search), градієнтний підхід, байєсові оптимізаційні методи.

Методи порівняльного аналізу.

Порівнюється ефективність запропонованих моделей ЗНМ, отриманих за допомогою ГА, з існуючими базовими архітектурами (ResNet, MobileNet,

EfficientNet). Критеріями виступають точність, швидкодія, обчислювальна складність, стійкість до змін у даних.

Методи обробки великих даних.

Для аналізу результатів експериментів використовуються алгоритми обробки великих обсягів інформації, такі як аналіз трендів, побудова кореляційних матриць, методи кластеризації.

Аналіз ефективності ГА.

Включає оцінку збіжності алгоритмів, швидкості знаходження оптимального рішення та ступеня охоплення простору пошуку. Оцінюються як кількісні, так і якісні характеристики результатів.

Реалізація генетичних алгоритмів.

Для реалізації ГА використовуються бібліотеки Python, такі як DEAP (Distributed Evolutionary Algorithms in Python) та спеціалізовані бібліотеки для нейронних мереж (TensorFlow, PyTorch).

Обчислювальні платформи.

Для прискорення експериментів застосовуються високопродуктивні обчислювальні платформи з графічними процесорами (GPU), а також системи розподілених обчислень для роботи з великими даними.

Фреймворки для автоматизованого пошуку архітектур.

Крім власної реалізації ГА, використовуються існуючі інструменти для пошуку архітектур, такі як Auto-Keras, NASNet, для порівняння ефективності різних методів.

Особливості застосування генетичних алгоритмів.

Генетичні алгоритми дозволяють:

- знаходити нетривіальні рішення у великому просторі параметрів ЗНМ завдяки їх здатності уникати локальних екстремумів;
- адаптувати структуру мережі до специфіки даних і задач виборничих процесів, включаючи неоднорідні дані та задачі з обмеженими ресурсами;

- автоматизувати процес підбору параметрів та архітектури, що знижує залежність від експертних знань.

Очікувані результати

Виконання зазначених завдань дозволить створити теоретичну та практичну основу для структурно-параметричного синтезу згорткових нейронних мереж, що сприятиме підвищенню ефективності виборничих процесів у різноманітних сферах, включаючи моделювання та проектування з використанням систем VR/AR.

Наукова новизна одержаних результатів

1. Розроблена методологія побудови системи віртуальної та доповненої реальності на основі графічного двигуна «Unity3d» яка відрізняється від існуючих наявністю вбудованого механізму для гнучкої інтеграції з гібридними нейронними мережами для роботи з системами управління або прямого впливу на стан середовища віртуальної реальності. Запропонований підхід вдосконалює та розширює можливості застосування нейронних мереж в таких системах.
2. Розроблена методологія багатокритеріального структурно-параметричного синтезу гібридних згорткових нейронних мереж із використанням генетичних алгоритмів. Запропонований алгоритм відрізняється механізмом ініціалізації популяції та функцією мутації що впроваджує механізм динамічної комбінації блоків різних типів в залежності від критеріальної оцінки індивіда у рамках єдиної архітектури ЗНМ, запропоновано багатокритеріальну фітнес функцію для їх оцінки. Запропонований алгоритм оптимізує процес синтезу та дозволяє отримати більш оптимальну модель ЗНМ за меншу кількість ітерацій.

3. Розроблено шляхи для практичного застосування на програмно-алгоритмічний комплекс для автоматизованого структурно-параметричного синтезу ЗНМ із використанням запропонованого модифікованого генетичного алгоритму та подальшого впровадження отриманої моделі з системами віртуальної та доповненої реальності. Запропоновані підходи та методи значно розширюють можливості автоматизованого проектування оптимальних нейронних мереж та розширює можливості з використання віртуальної та доповненої реальності в виробничих процесах.

Апробація результатів дослідження

1. "Multicriteria optimization of hybrid convolutional neural network structural synthesis using evolutionary algorithms", Boryndo, I., Siveglazov, V., Zgurovsky, M.Z., CEUR Workshop Proceedings, 2024, 3790, pp. 318–330
2. "Visual Navigation Systems Implementation Using Hybrid Convolutional Neural Networks", Sineglazov, V., Boryndo, I., 2023 IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control, MSNMC 2023 - Proceedings, 2023, pp. 155–158
3. "Intelligence system for emotional facial state estimation during inspection control", Sineglazov, V., Pantyeyev, R., Boryndo, I., CEUR Workshop Proceedings, 2019, 2683, pp. 25–29
4. "Virtual Reality Systems Integration With Neural Networks For Immersivity Enhancement", Illia Boryndo, Viktor Sineglazov, "Information technologies and computer modelling", 2023, Ivano-Frankivsk, ISBN 978-617-8128-23-4, pp. 87
5. "Topology Analysis of Hybrid Convolutional Neural Networks", Viktor Sineglazov, Olena Chumachenko and Illia Boryndo, "Information systems and technologies", 2021, ISBN 978-617-7519-59-0, pp. 53-56

6. "Multicriteria optimization for structural parametric synthesis of convolutional neural networks", Illia Boryndo, Viktor Sineglazov, "Information technologies and computer modelling" proceedings of the international scientific conference 2024, may, 21st to 24th Ivano-Frankivsk, ISBN 978-966-640-560-2, pp. 227-230
7. "Application of deep learning hybrid convolutional neural networks in visual navigation systems", Illia Boryndo,

Публікації

1. Illia Boryndo, Viktor Sineglazov (2022). "The Optimal Choice of Hybrid Convolutional Neural Network Components", American Journal of Neural Networks and Applications, 8(2), 12-16.
<https://doi.org/10.11648/j.ajnna.20220802.11>
2. "Intelligent recognition and integration of graphical elements into virtual surrounding within augmented reality using hybrid convolutional neural networks", Sineglazov Viktor, I. Boryndo, 2023, Artificial Intelligence. 28. 74-79. 10.15407/jai2023.01.074.
3. "Application of a Multicriteria Genetic Algorithm for Structural Parametric Synthesis of Convolutional Neural Networks", Artificial Intelligence. 29(4), 2024, 106-114., DOI: 10.15407/jai2024.04.106
4. "Application of Neural Networks for Virtual and Augmented Reality", Sineglazov Viktor, I. Boryndo, 2022,. Electronics and Control Systems. 4. 51-57. 10.18372/1990-5548.74.17296.
5. "Hand Gestures Recognition and Tracking Within Virtual Reality using Hybrid Convolutional Neural Networks", Sineglazov Viktor, I. Boryndo, 2022, Electronics and Control Systems. 2. 32-37. 10.18372/1990-5548.72.16940.

РОЗДІЛ 1

ОГЛЯД ТА АНАЛІЗ СУЧАСНИХ ТОПОЛОГІЧНИХ РІЗНОВИДІВ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ТА АЛГОРИТМІВ СТРУКТУРНОГО СИНТЕЗУ

1.1 Аналіз сучасних архітектурних рішень при проектуванні згорткових нейронних мереж

Згорткові нейронні мережі (ЗНМ) здійснили революцію у сфері комп'ютерного зору, забезпечуючи видатну продуктивність у таких завданнях, як класифікація зображень, виявлення об'єктів і сегментація. Попри їхній успіх, прагнення досягти вищої точності та універсальності спонукало дослідників до створення гібридних моделей, які об'єднують ЗНМ з іншими архітектурами нейронних мереж. Ці гібридні моделі використовують унікальні переваги різних типів мереж, усуваючи недоліки чистих ЗНМ і розширюючи їх застосовність для більшого кола завдань.

На сьогодні, все ще триває активна розробка згорткових нейронних мереж і активно висуваються нові архітектурні рішення та шляхи їх використання. Але розглянувши уже існуючі архітекти ЗНМ, спостерігається, що значні покращення у роботі мереж відбулися з 2019-2021 років. Потенціал та результативність окремих ЗНМ зазвичай залежить від їх глибини та наповненості графічної вибірки абстракціями різної складності. Однак основна проблема, з якою стикаються глибокі нейронні мережі - це падіння градієнта. Спочатку цю проблему намагались вирішити шляхом підключення проміжних шарів, але у 2015 році була розроблена система нових зв'язків для покращення коефіцієнта конвергенції глибоких ЗНМ [1]. Різні експериментальні дослідження показали, що найсучасніші глибокі ЗНМ, такі як «VGG», «ResNet», «ResNext», показали хороші результати для вирішення проблем розпізнавання та локалізації, таких як семантична та екземплярна сегментація об'єктів, парсинг сцен, тощо. Більшість відомих архітектур ЗНМ для виявлення та сегментації об'єктів, таких як «Single Shot Multibox Detector»

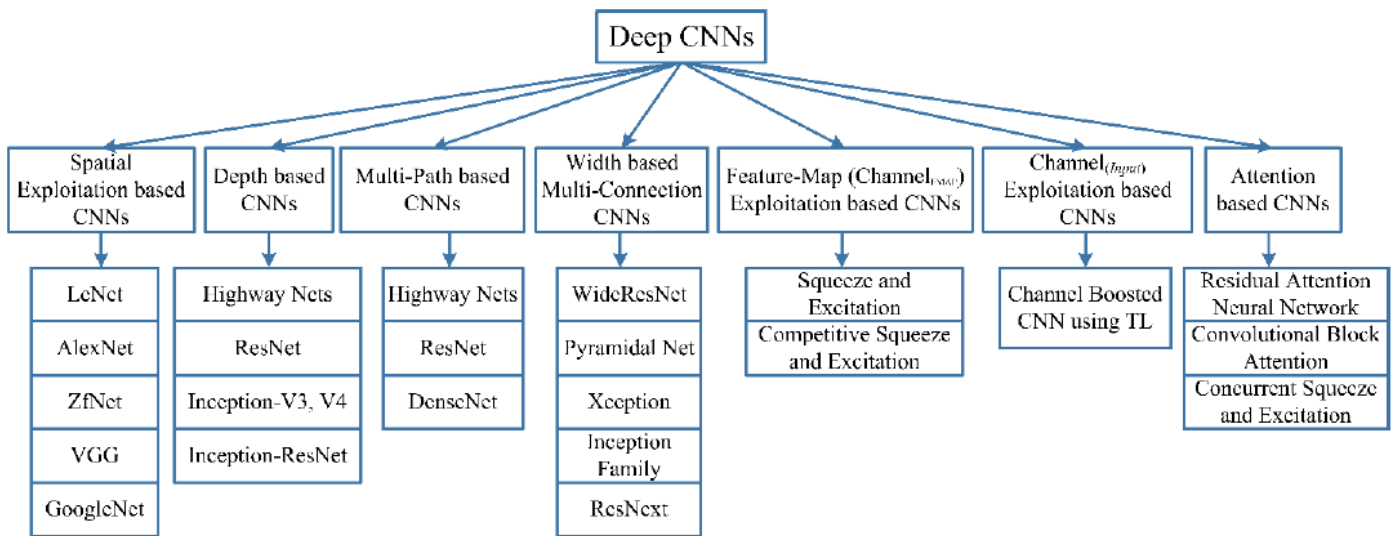


Рис. 1.1. Топологічний розподіл сучасних згортових нейронних мереж (SSD), «Regional CNN», «Faster R-CNN», «Mask R-CNN» були побудовані базуючись на архітектурних особливостях «ResNet», «VGG», «Inception», тощо. Аналогічно, більшість концептуально нових ЗНМ, таких як пірамідальні ЗНМ, каскадні «R-CNN», «Libra R-CNN» тощо, для покращення продуктивності модифікували вже існуючі архітектури, як згадувалося раніше. Застосування глибоких ЗНМ також було розширено шляхом поєднання цих мереж з повторюваною нейронною мережею (ПНМ) і таким чином показало найкращі результати в змаганні MS COCO-2020[2, 3].

На Рисунку 1.1 зображено топологічний розподіл сучасних нейронних мереж в залежності від їх архітектурних властивостей.

Таблиця 1.1. Структурне порівняння сучасних ЗНМ та їх критеріїв швидкодії

Назва архітектури	Основний внесок	Параметри	Помилка
LeNet	- Перша популярна архітектура CNN	0.060 М	MNIST: 0.8 MNIST: 0.95
AlexNet	- Глибша і ширша, ніж LeNet - Використовує Relu, dropout та overlap pooling	60 М	ImageNet: 16.4
ZFNet	- Візуалізація проміжних шарів	60 М	ImageNet: 11.7

GoogLeNet	- Ідея розподілу, трансформації та об'єднання	4 М	ImageNet: 6.7
Inception-V3	- Вирішує проблему вузьких місць у репрезентації - Замінює великі фільтри малими	23.6 М	ImageNet: 3.5 Multi-Crop: 3.58
Highway Networks	- Запроваджено ідею багатошляхового підходу	2.3 М	CIFAR-10: 7.76
Inception-ResNet	- Поеднує розподіл, трансформацію, об'єднання та залишкові зв'язки	55.8 М	ImageNet: 3.52
DelugeNet	- Дозволяє потік інформації між шарами у глибоких мережах	20.2 М	CIFAR-10: 3.76
Residual Attention Neural Network	- Введено механізм уваги	8.6 М	CIFAR-100: 20.4 ImageNet: 8.4
Squeeze & Excitation Networks	- Моделює взаємозалежності між картами ознак	27.5 М	ImageNet: 2.3
DenseNet	- Потік інформації між шарами	25.6 М	CIFAR100+: 17.18
PolyNet	- Експериментальне структурне різноманіття - Введено модуль Poly Inception - Генералізує залишкову одиницю за допомогою поліноміальних композицій	92 М	ImageNet: Single: 4.25 Multi: 3.45
Convolutional Block Attention Module	- Використання просторової уваги та уваги до карт ознак	48.96 М	ImageNet: 5.59
Concurrent Spatial & Channel Excitation Mechanism	- Просторова увага - Увага до карт ознак - Сумісне розміщення просторової та канальної уваги	-	MALC: 0.12 Visceral: 0.09
Channel Boosted CNN	- Посилення оригінальних каналів додатковою інформацією, створеною	-	-

	штучними каналами, насиченими ознаками		
Competitive Squeeze & Excitation Network CMPE-SE- WRN-28	- Залишкові та ідентифікаційні відображення використовуються для перерозмірування карт ознак	36.92 M	CIFAR-10: 3.58 CIFAR-100: 18.47

Архітектура гібридних згорткових нейронних мереж (ГЗНМ) інтегрує ЗНМ із різними іншими моделями нейронних мереж, використовуючи їхні переваги для створення потужніших та універсальніших моделей. Далі розглянемо архітектури й механізми трьох основних типів ГЗНМ: гібриди ЗНМ-РНМ, ЗНМ-ГНМ і ЗНМ-Трансформер [3, 4, 5].

Гібриди ЗНМ-РНМ

Ці моделі поєднують здатність ЗНМ витягувати просторові ознаки з можливістю обробки послідовних даних у рекурентних нейронних мережах (РНМ). Це особливо корисно для завдань, що включають часові послідовності, як-от аналіз відео чи обробка серій зображень.

Архітектура:

Компонент ЗНМ: Відповідає за витяг просторових ознак із вхідних даних. У випадку відеоданих часто використовуються 3D-згортки для аналізу просторово-часових особливостей у кількох кадрах. У завданнях підпису зображень застосовують 2D-ЗНМ, такі як ResNet або VGGNet[4, 5].

Компонент РНМ: Компонент РНМ, наприклад LSTM (довготривала короткочасна пам'ять) або GRU (гейтована рекурентна одиниця), обробляє послідовність ознак, створених ЗНМ. LSTM і GRU здатні враховувати довготривалі залежності та часову динаміку.

Механізм інтеграції: Інтеграція ЗНМ і РНМ може відбуватися на різних етапах. Одним із підходів є передача вихідних карт ознак ЗНМ безпосередньо на вхід РНМ. Наприклад, у класифікації відео 3D-ЗНМ

витагує ознаки з кожного кадру, які послідовно обробляються LSTM, що моделює часові залежності між кадрами. Також можуть використовуватися механізми уваги для фокусування на релевантних частинах вхідної послідовності.

Гібриди ЗНМ-ГНМ

Гібриди ЗНМ-ГНМ поєднують здатність ЗНМ до витягування просторових ознак із можливостями графових нейронних мереж (ГНМ) працювати з даними, представленими у вигляді графів. Це особливо корисно для завдань, де дані мають як просторові, так і реляційні компоненти.

Архітектура:

Компонент ЗНМ: Виконує витяг просторових ознак, як описано раніше.

Компонент ГНМ: ГНМ обробляє дані, які природно представлені як графи. Графові згорткові мережі (ГНМ) і мережі з графовою увагою (ГАТ) здатні моделювати залежності та відносини між вузлами графа.

Механізм інтеграції: Інтеграція зазвичай передбачає використання ознак, витягнутих ЗНМ, як вхідних даних для ГНМ. Наприклад, у соціальному мережевому аналізі ЗНМ можна використовувати для витягування ознак із зображень профілів користувачів, тоді як ГНМ моделює відносини між користувачами на основі графової структури мережі. Об'єднана модель може використовувати як візуальну, так і реляційну інформацію.

Гібриди ЗНМ-Transformer

Ці моделі інтегрують згорткові операції ЗНМ із механізмами самоуваги трансформерів. Це поєднання ефективно покращує здатність моделі враховувати як локальні, так і глобальні залежності в даних.

Архітектура:

Компонент ЗНМ: Виконує витяг ознак із просторовим фокусом, як у попередніх прикладах.

Компонент трансформера: Трансформери, що включають механізми багатоголової самоуваги та нейронні мережі прямого зв'язку, обробляють ознаки, витягнуті ЗНМ. Завдяки механізмам уваги трансформери чудово враховують довготривалі залежності та глобальний контекст.

Механізм інтеграції: Одним із підходів є використання ЗНМ для витягування початкового набору ознак, які потім передаються у шари самоуваги трансформера. Це дозволяє моделі зосереджуватися на релевантних частинах вхідних даних, покращуючи контекстуальне розуміння. Наприклад, у завданні підпису зображень ЗНМ може витягати ознаки з зображення, які потім обробляються трансформером для генерації описових підписів.

Гібридні згорткові нейронні мережі демонструють синергію, досягнуту шляхом поєднання сильних сторін різних моделей нейронних мереж. ЗНМ ефективні у витягуванні просторових ознак, роблячи їх ключовими компонентами цих гібридних архітектур. У поєднанні з РНМ вони успішно моделюють часові послідовності; у комбінації з ГНМ обробляють реляційні дані; а при інтеграції з трансформерами здатні враховувати як локальні, так і глобальні залежності у даних. Ці гібридні архітектури продемонстрували вищу продуктивність у різних сферах, від класифікації відео та підпису зображень до аналізу соціальних мереж і прогнозування молекулярних графів, підкреслюючи їхню універсальність та ефективність.

1.2. Аналіз топологічних ознак та функціональних особливостей сучасних нейронних мереж

На сьогоднішній день спостерігається постійний розвиток у топологічному аналізі сучасних згорткових нейронних мереж (ЗНМ), і регулярно пропонуються нові архітектурні рішення та області застосування. У зв'язку з ростом кількості прикладних завдань, замість поступового збільшення складності, більшість сучасних архітектур ЗНМ почали впроваджувати функціональні структурні компоненти. Ці функціональні

компоненти включають набір базових шарів, які вдосконалено за рахунок унікальних підходів до з'єднання, комбінації та функціональної постобробки. Такі компоненти здатні значно підвищити якісні параметри мереж і можуть використовуватися як самостійно, так і як частина складніших структур.

Перш ніж детально розглядати структурні блоки, необхідно спочатку проаналізувати сучасні ЗНМ, їхні переваги та недоліки. На Рисунку 1 наведено список сучасних архітектур ЗНМ, їх функціональні та технічні характеристики, а також основні параметри оцінки. Кожна з цих мереж має власний структурний блок або функціональний підхід, які можуть бути повторно використані для створення оптимальної підархітектури за допомогою еволюційних алгоритмів.

ГЗНМ представляють собою значні глибокі структури у сфері глибинного навчання. Вони застосовують набір фільтрів (або ядер) до вхідних даних для створення карт ознак, які виділяють різні аспекти вхідної інформації, такі як контури, текстури та форми. Кожен компонент, від згорткових і пулінгових шарів до функцій активації та спеціалізованих блоків, відіграє ключову роль у перетворенні сирих вхідних даних у високорівневі представлення, які можна використовувати для різноманітних завдань машинного навчання[6].

З іншого боку, ГЗНМ є функціональним набором кількох ЗНМ, які об'єднані таким чином, щоб задовольнити конкретну мету. Вони використовують підхід із розподілом відповідальності, що гарантує, що кожен компонент кінцевої архітектури (ЗНМ, алгоритми класифікації/розпізнавання, процесори вхідних/вихідних даних) виконує лише одне конкретне завдання. Завдяки цьому складні індивідуальні завдання можна розбити на ключові етапи обробки даних, які необхідно реалізувати. Розбивання завдань на спрощені етапи робить навчання нейронних мереж простішим, що сприяє покращенню точності та продуктивності.

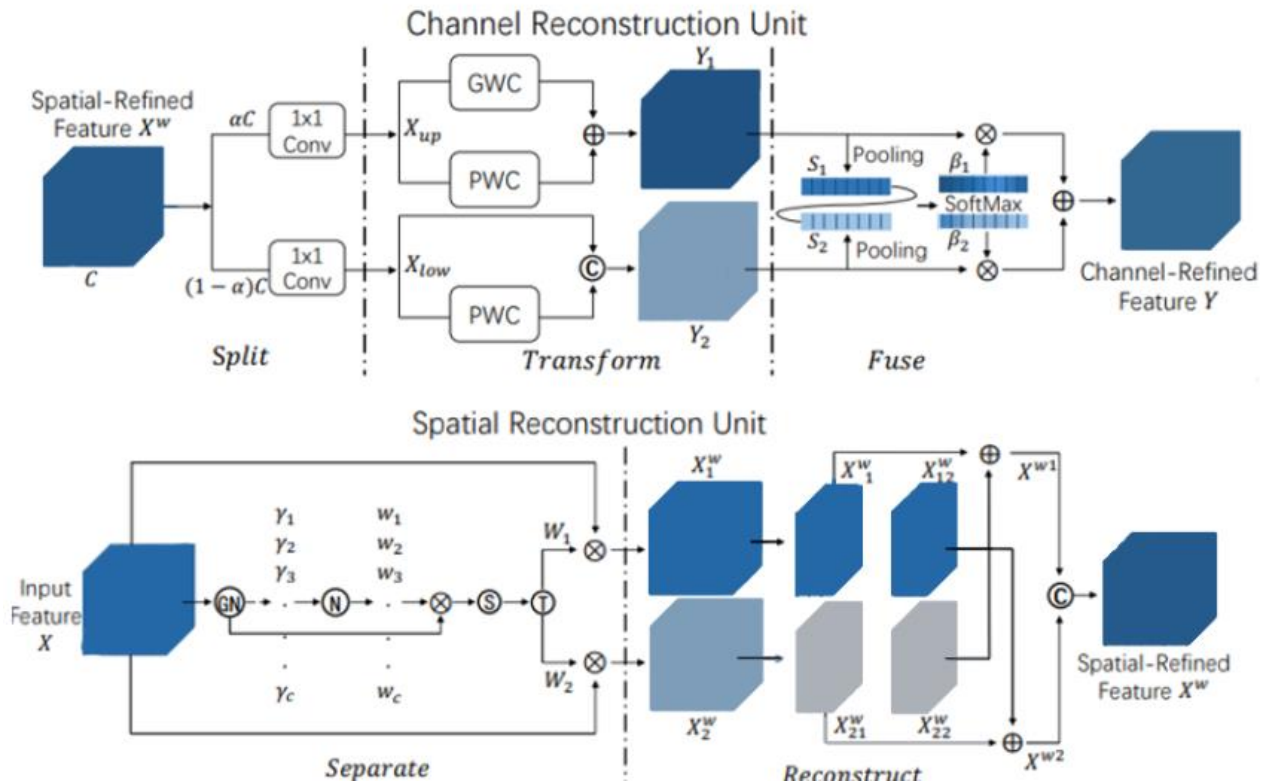


Рис. 1.2. Структурна діаграма блоку просторово-канальної згортки

Будь-який тип ЗНМ може підходити для цього підходу. Основна увага приділяється деталям завдань і параметрам вхідних даних, таким як роздільна здатність зображень, масштаб, кольорові канали та кількість навчальних зразків.

1.2.1. Просторово-канальна реконструкція згортки

Вузькі місця (bottleneck structure), де згорткові шари 3×3 домінують у параметрах моделі та кількості операцій FLOPs, виявилися добре пристосованими до проектування архітектури ЗНМ. Однак запропоновано альтернативні ефективні методи згортки, такі як групова згортка (GWC), згортка по глибині (DWC) та точкова згортка (PWC), які є формою стандартної згортки та можуть замінити ресурсоємні поточні операції згортки. Наприклад, SPConv розділяє вхідні канали на дві групи для окремої обробки, хоча це потребує значного обсягу обчислень для вилучення внутрішніх даних.

Попри це, всі зазначені вище техніки зосереджені на зменшенні дублювання у каналній або просторовій площинах, що все ще залишає проблему надмірності ознак невирішеною. Просторово-канална реконструкція згортки (SCConv) складається з двох компонентів: блоку просторової реконструкції (SRU) і блоку каналної реконструкції (CRU). Ці компоненти можна легко інтегрувати у різні архітектури.

Цей модуль ефективно спрямований на зменшення дублювання ознак. Він знижує кількість параметрів моделі та обсяг обчислень (FLOPs), одночасно розширюючи можливості представлення ознак. На Рисунку 1.2 зображено структурну схему SRU та CRU разом із їх функціональними деталями [7, 8].

Основні елементи:

- SRU (блок просторової реконструкції): сегрегує повторювані ознаки, використовуючи вагові коефіцієнти, і реконструює їх для зменшення надмірності у просторовому аспекті, покращуючи представлення ознак.
- CRU (блок каналної реконструкції): використовує підхід "розділити-трансформувати-об'єднати" для зменшення дублювання у каналній площині, що дозволяє знизити обчислювальні та пам'яттєві витрати.
- SCConv (просторово-канална згортка): інтегрує SRU та CRU послідовно, замінюючи стандартну згортку у різних базових архітектурах штучних нейронних мереж (ШНМ). Цей модуль не лише знижує обчислювальні витрати, а й покращує продуктивність моделей у складних завданнях.

1.2.2. Щільно-зв'язний механізм уваги

Для усунення недоліків глибоких згорткових нейронних мереж (ЗНМ) запропоновано метод, заснований на залишковій щільній мережі та механізмі уваги. Багатоканальна структура включає щільний залишковий канал, канал

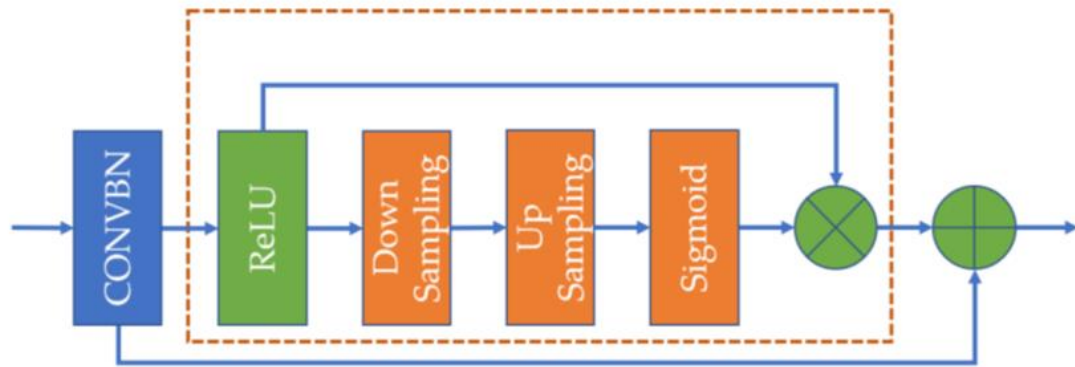


Рис. 1.3. Структурна діаграма щільно-зв'язного блоку уваги просторової уваги та двошаровий згортковий канал. Модель також містить багатоканальну структуру, шар об'єднання ознак і шар класифікації. На Рисунку 1.3 представлено структуру мережі.

C1: щільний залишковий канал.

C2: канал просторової уваги.

C3: двошаровий згортковий канал, що складається з двох згорткових шарів із ядрами розміром 5.

Ці компоненти призначені для усунення надлишкової інформації та її стиснення.

Перший шар каналу побудовано як мережу із щільною залишковою структурою. У разі використання традиційної глибокої мережі втрата ознак під час прямої передачі може негативно вплинути на кінцевий результат. У цій архітектурі запропоновано тісно поєднати блоки залишкового навчання, формуючи щільну залишкову структуру. На Рисунку 1.3 продемонстровано цей підхід. У схемі видно, що залишковий щільний блок складається з кількох шарів згортки, нормалізації пакетів, функцій активації та скорочених з'єднань. Така структура покращує потік інформації, вирішуючи проблему затухання градієнта моделі. Завдяки повторному використанню попередніх графів згортки можна витягувати важливі ознаки, уникаючи втручання надлишкових даних.

1.2.3. Підсилення каналів

Продуктивність навчання ЗНМ також залежить від вхідного представлення та його параметрів. Продуктивність ЗНМ може знижуватися через відсутність різноманітних ознак та специфічних класових даних у цільовій вибірці. Для вирішення цієї проблеми розроблено концепцію підсилення каналів у ЗНМ. Ця техніка базується на розширенні розмірності вхідного каналу та залученні додаткових навчальних модулів для покращення представлення мережі.

Цей модуль призначений для присвоєння різних ваг кожному каналу, посилюючи ті, які містять найбільш важливу інформацію для виконання завдання. Важливі канали потім передаються до наступних шарів, створюючи мережу, яка фокусується на ключових ознаках. Основні характеристики:

- Підвищення здатності ЗНМ представляти дані за рахунок збільшення кількості вхідних каналів.
- Використання індуктивного перенесення навчання для створення покращеного вхідного представлення.
- Можливість збільшення обчислювального навантаження через створення додаткових каналів.

1.2.4. Ітераційний аналіз карт ознак

Зі збільшенням складності архітектур глибокого навчання кількість карт ознак на кожному етапі продовжує зростати. Хоча деякі з цих карт є

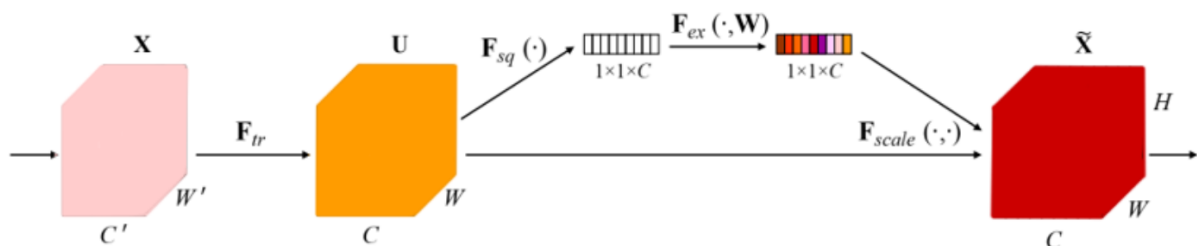


Рис. 1.4. Структурна схема блоку «Squeeze and Excitation»

важливими для класифікаційних завдань, інші можуть бути менш значущими або навіть зайвими. Мережі SENet вдосконалюють традиційні ЗНМ, збільшуючи їхню здатність до представлення шляхом коригування ваг каналів ознак. Вони включають новий тип архітектурного елемента — блок "Стиснення і Збудження" (SE), який дозволяє мережі зосереджуватися на найбільш важливих ознаках [27, 28].

Основні характеристики:

- SE-блок може бути інтегрований у будь-яку стандартну архітектуру ЗНМ.
- Динамічне коригування важливості каналів покращує здатність мережі до узагальнення на нові дані.
- Додаткові обчислювальні витрати, пов'язані з SE-блоком, мінімальні порівняно із загальним приростом продуктивності.

1.2.5. Згорткові нейронні мережі на основі механізмів уваги

Мережі на основі уваги реалізують здатність вибирати, яка частина зображення є найбільш важливою для обробки. Згорткові нейронні мережі на основі уваги (ACNN) вдосконалюють традиційні ЗНМ, впроваджуючи механізми уваги, які дозволяють мережі фокусуватися на найбільш релевантних частинах вхідних даних. Ця вибірковість дозволяє ACNN захоплювати довготривалі залежності та покращувати представлення ознак, що сприяє кращій продуктивності в таких завданнях, як класифікація зображень, виявлення об'єктів і сегментація зображень.

Розглянемо основні характеристики та переваги деяких із цих підходів:

Характеристики залишкової мережі уваги (RANN):

- Створює карти ознак, чутливі до сигналів уваги.
- Залишкове навчання забезпечує простоту масштабування.
- Пропонує різні способи відображення вибраних секцій.
- Використовує прямий потік уваги для застосування ваг до ознак як із нижнього вгору, так і зверху вниз.

- Уникає зайвих ускладнень.

Специфікації модуля уваги згорткового блоку (СВАМ):

- СВАМ — це фундаментальний модуль уваги, створений для ЗНМ із прямим потоком.
- Генерує карти уваги для каналів і простору послідовно.
- Карти уваги каналів допомагають визначити, на що зосередитися.
- Просторова увага допомагає вирішити, куди спрямувати увагу.
- Покращує плавний потік інформації.
- Використовує глобальне середнє і максимальне підсумовування одночасно.

1.3. Структурний аналіз згорткових нейронних мереж

Структура згорткових мереж є такою ж, як і логічна структура зв'язків нейронів мозку людини і базується на частині "зорової кори". Кожен нейрон реагує на подразники лише в спеціально обмеженій зоні поля зору, відомій як рецептивна область. Набір таких областей в результаті охоплює всю зорову область. Згорткові мережі здатні чітко фіксувати просторові та часові залежності будь-яких вхідних даних зображення за допомогою відповідних фільтрів.

Структура ЗНМ дає можливість гнучкого налаштування для конкретної прикладної задачі та її графічних даних шляхом зменшення кількості залучених конфігураційних параметрів та повторного використання ваг.

Для навчання ЗНМ використовується метод зворотного поширення похибки. Однією із серйозних проблем навчання ЗНМ є проблема перенавчання, коли модель добре пояснює тільки приклади з навчальної вибірки, адаптуючись до навчальних прикладів, замість того щоб вчитися класифікувати приклади, які не брали участі в навчанні (втрачаючи здатність до узагальнення). Існують наступні підходи для уникнення такої ситуації: штучні дані, рання зупинка, обмеження кількості параметрів, методи

проріджування або виключення, виключення з'єднань, ослаблення ваг, ієрархічних координатних сіток.

Визначення найбільш значущих з точки зору ефективності параметрів згорткової нейронної мережі проводилося в результаті експерименту над комбінованою мережею, яка складається з згорткової нейронної мережі, класифікатора та розгорткової нейронної мережі.

На основі аналізу результатів перевірки комбінованої ЗНМ можна визначити значущі параметри для ЗНМ [11, 13]:

- 1) кількість шарів згортки.
- 2) кількість шарів агрегації.
- 3) взаємне розміщення шарів згортки та агрегуючих шарів.
- 4) шари згортки (на кожному шарі окремо):
 - розмір ядра згортки(на кожному шарі окремо);
 - кількість карт ознак(на кожному шарі окремо);
 - величина зміщення (на кожному шарі окремо);
 - параметр крайового ефекту.
- 5) шари агрегації (на кожному шарі окремо):
 - розмір ядра агрегації;
 - функція ядра агрегації.
- 6) повноз'єднані шари (на кожному шарі окремо):
 - кількість повноз'єднаних шарів;
 - розмір кожного шару;
 - тип класифікатора: повнозв'язний шар або щось інше.

Основним і першим шаром ЗНМ є згортковий шар. Цей шар визначає логіку вилучення ознак з вхідного набору пікселів. Данний шар імплементує логічну операцію, яка включає вхідний набір пікселів (зображення) та фільтр (ядро).

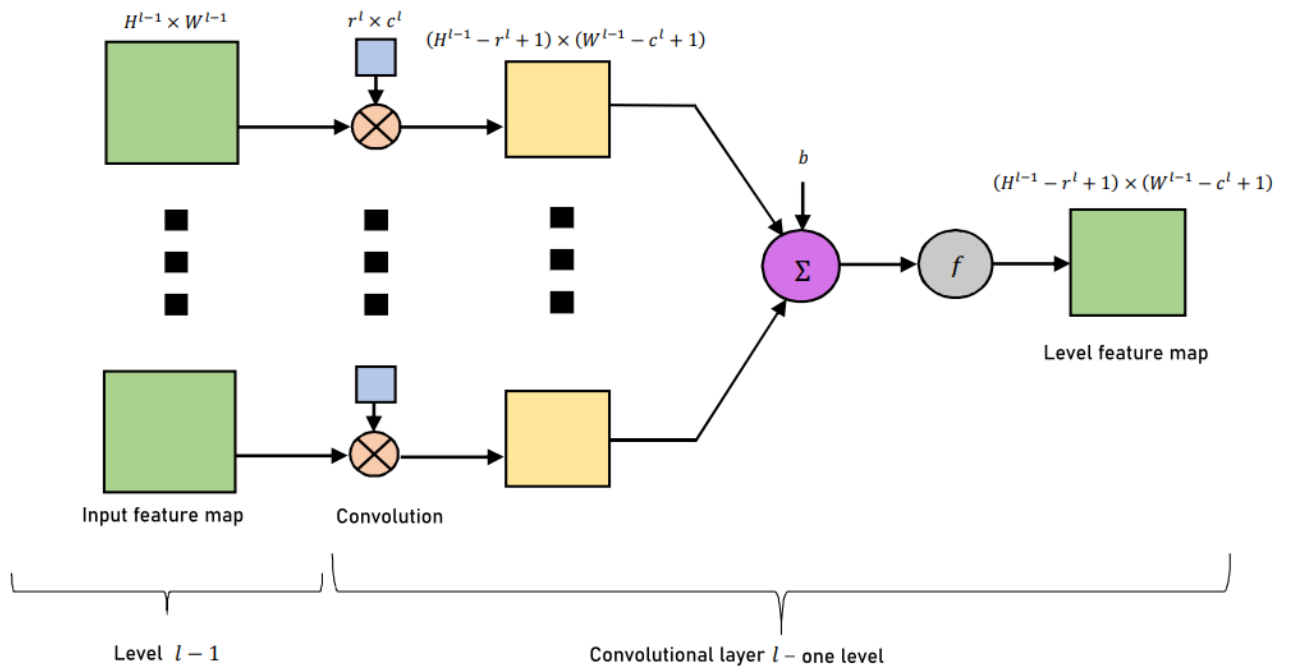


Рис. 1.5. Структурна схема згорткового шару

На Рис. 1.6 зелена зона являє собою вхідне зображення $5 \times 5 \times 1$, а елемент, який використовується в процесі згортки називається «Кернел» або іншими словами «фільтр», K , та позначений на рисунку жовтим кольором. Він виглядає як матриця 3×3 :

“Кернел”/”Фільтр”, $K =$

$$\begin{matrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}$$

Фільтр рухається зліва направо з попередньо встановленим значенням кроку, поки не досягне кінця поточного рядка. Наприкінці рядка фільтр зміщується на наступний рядок і процес повторюється по аналогії. Через деякий час фільтр завершує всі рядки, і ми отримуємо вихідну матрицю.

Застосування різних видів фільтрів дозволяє отримувати різні результати, такі як контури, різкість, ознаки, виявлення карти рельєфу. Крім того, фільтри повинні мати таку ж саму глибину кольору що і вхідне

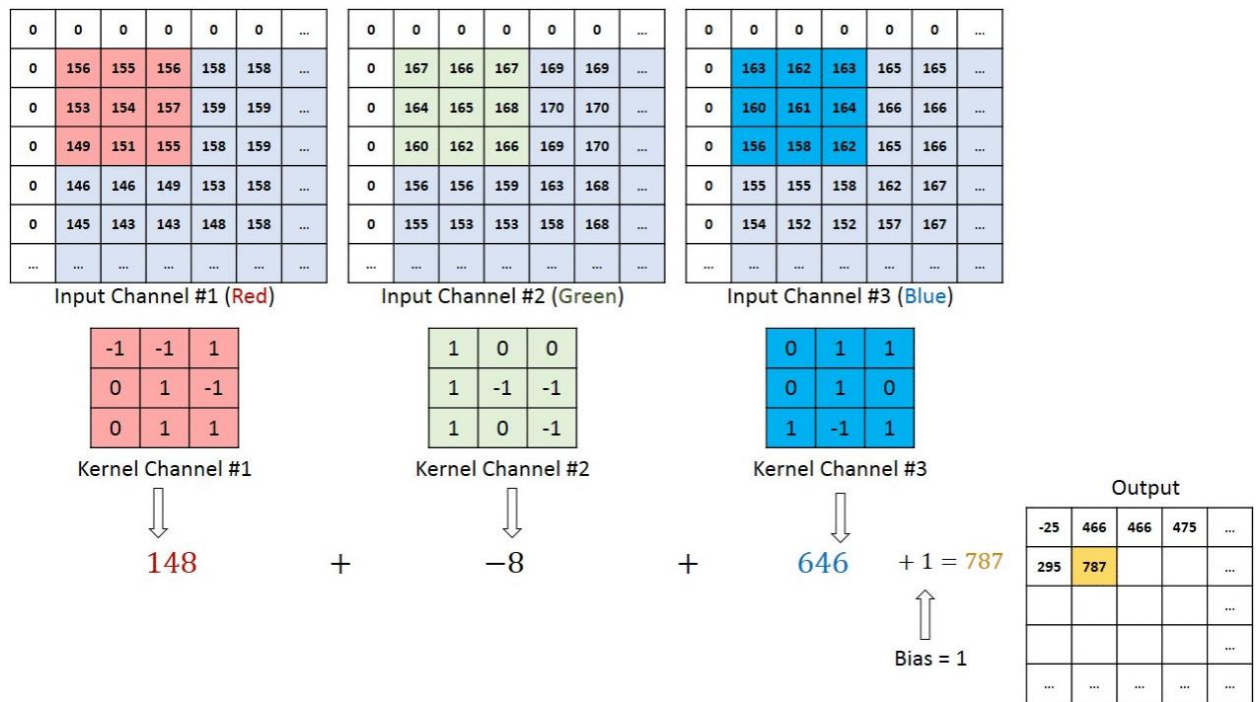


Рис. 1.6. Функціональна схема роботи фільтрів

зображення. Помноження матриці застосовується поміж K_n та I_n стеком ($[K1, I1]$; $[K2, I2]$; $[K3, I3]$), далі до всіх результатів додається значення зсуву щоб отримати сплюснуте одно-канальне значення.

Нехай згортковий шар буде позначений як l . У подібній нейронній мережі l задане як непарне число, тобто $l = 1, 3, \dots, 2\alpha + 1$.

Тоді для карти ознак n відбудеться наступне:

$w_{m,n}^l = \{w_{m,n}^l(i, j)\}$ - згортка, застосована до карти ознак m ($L - 1$), на шарі L з картою ознак n ;

b_n^l - порогові значення, приєднані до карти ознак n на шарі l ;

V_n^l - перелік усіх рівнів ($l - 1$), які пов'язані з картою ознак m на згортковому шарі l .

Отже, карта ознак n згорткового шару l може бути представлена наступним чином:

$$y_n^l = f_l \left(\sum_{m \in V_n^l} y_m^{l-1} \otimes w_{m,n}^l + b_n^l \right) \quad (1.1)$$

де під оператором $\otimes$ ми розуміємо математичну операцію двовимірної згортки.

Припустимо, що розмір вхідних карт ознак y_m^{l-1} дорівнює $H^{l-1} \times W^{l-1}$, а розмір згортки, застосованої до них, $w_{m,n}^l$ дорівнює $r^l \times c^l$, тоді розмір вихідної карти ознак y_n^l обчислюється згідно рисунку 1.2.

Шар агрегації базується на процесі зменшення просторового розміру виділеної шаром згортки ознаки. Завдяки зменшенню розмірності можна виділити домінуючі ознаки та зменшити навантаження на обчислювальні елементи що сильно прискорює процес навчання.

В основному шари агрегації поділяються на два типи: "максимізаційне агрегування" та "усереднювальне агрегування". Максимізаційне агрегування знаходить максимальне значення в отриманій з фільтра матриці. У свою чергу, усереднювальне агрегування знаходить середнє значення в отриманій з фільтра матриці. Максимізаційне агрегування також може використовуватись для усунення шумів вихідних даних і завдяки цьому воно краще ніж усереднювальне агрегування [15].

Зазвичай шари агрегації використовуються в парі з згортковими шарами, таким чином ця пара створює i -й шар згортки в структурі ЗНМ. На основі параметрів вхідних зображень, розмірність та кількість подібних шарів ЗНМ може різнитись що спричиняє різну потребу в обчислювальній потужності. Під кінець, вихідні дані надходить в згладжувальний шар який перетворює їх у вектор для подальшої подачі в класифікатор (повнозв'язний шар).

Введемо шар підвибірки 1. У згортковій нейронній мережі 1 прийнято вважати парним числом, тобто $l = 2, 4, \dots, 2\alpha$. Для карт ознак n ми вводимо такі позначення: $w_{m,n}^k$ - фільтр, застосований до n на шарі l , і b_n^l - додаткове порогове значення. Далі перетворення будуть виконуватись наступним чином: розділимо карту ознак n з $(l - 1)$ -го шару на непересічні блоки розміром 2×2 пікселя. Потім підсумовуємо значення чотирьох пікселів у кожному блоці і в результаті отримуємо матрицю $z_n^{l-1} = \{z_n^{l-1}(i,j)\}$, елементами якої будуть відповідні значення сум. Таким чином, формула для розрахунку значень елементів матриці буде такою:

$$z_n^{l-1} = y_n^{l-1}(2i - 1, 2j - 1) + y_n^{l-1}(2i - 1, 2j) + y_n^{l-1}(2i, 2j - 1) + y_n^{l-1}(2i, 2j)$$

Карта ознак n шару субвибірки l обчислюється за формулою:

$$y_n^l = f_l(z_n^{l-1} \times w_{m,n}^l + b_n^l) \quad (1.2)$$

Завдяки вищезгаданим перетворенням стає можливим обчислити розмір $H^l \times W^l$ карт ознак y_n^l субвибірки l .

$$H^l = \frac{H^{l-1}}{2}, W^l = \frac{W^{l-1}}{2} \quad (1.3)$$

1.4. Структурно-параметричний синтез згорткових нейронних мереж

1.4.1. Постановка задачі структурно-параметричного синтезу

Головна ідея алгоритму структурно-параметричного синтезу це вирішення проблем оптимального вибору множини згорткових нейронних мереж або модулів які включаються до складу гібридної нейронної мережі, отримання структури окремої ЗНМ, конфігурації її внутрішніх параметрів та навчального процесу. Попередньо у цій роботі були представлені базові структурні блоки та різні архітектури ЗНМ, які будуть використовуватися у процесі структурно-параметричного синтезу.

Задача структурно-параметричного синтезу може бути описана наступним чином: нехай завершений набір даних являє собою $J = \{R_j, Y_j\} j = 1, \dots, P$ пар типу "атрибут-значення", де R_j, Y_j вхідний і вихідний вектор нейронної мережі відповідно. На основі даної навчальної вибірки необхідно синтезувати оптимальну структуру ЗНМ яка б забезпечувала ефективне вирішення прикладної проблеми (класифікація, наближення, передбачення). Критерій оптимальності вектора визначається як

$$\mathbf{I} = \{I_1(x), I_2(x)\} \rightarrow \text{opt} \quad (1.4)$$

де $I_1(x) = E_{\text{var}}(x)$ помилка узагальнення, що визначає величину похибки розв'язання задачі у тестовій вибірці; $I_2(x) = S(x)$ - складність нейронної мережі (кількість міжнейронних з'єднань); X - вектор значущих параметрів.

При виборі оптимальної структури нейронної мережі слід враховувати такі параметри:

- Список блоків, що використовуються в ЗНМ;
- Глибина мережі;
- Кількість блоків;
- Значення GFLOP при навчанні нейронної мережі;
- Складність підготовки набору вхідних даних.

Після підтвердження стартової конфігурації та загальної концепції ЗНМ, нарешті можливо застосувати процес структурно-параметричного синтезу. Під час генерації структури ЗНМ слід звернути увагу на наступні конфігураційні параметри:

- Шар згортки: кількість карт ознак, розмір кроку, розмір фільтра.
- Шар агрегації: розмір фільтра, функція агрегації.
- Повнозв'язний шар: кількість повнозв'язних шарів, розмір шарів, тип енкодера.
- Коефіцієнт скорочення, тощо.

Нарешті, щоб навчити та перевірити отриману мережу, нам потрібно підготувати набір вхідних графічних даних. Практична реалізація ЗНМ повинна застосовувати на вхідні дані такі випадкові перетворення, як:

- Випадкове віддзеркалювання зображення;
- Випадкове обрізання зображення;
- Зміна співвідношення сторін;
- Випадкове обертання зображення;
- Застосування функції «Pixel Jitter»;

1.4.2. Сучасні підходи до структурного синтезу згорткових нейронних мереж

Структурний синтез - це алгоритм генерації оптимальної структури ЗНМ що буде відповідати заданим критеріям . Одним із багатьох алгоритмів

структурного синтезу є генетичний алгоритм. Ідея полягає в тому, щоб згенерувати набір різних структур ЗНМ, розмістивши випадкові блоки на кожному етапі, а потім за допомогою тестування вибрати декілька з найкращими параметрами точності і швидкодії. Потім ми перетасовуємо складові частини обраної нами структури і тим самим генеруємо певну кількість нових варіацій даної ЗНМ. Знову перевіряємо їх і знову відбираємо найкращі з них. Виконуючи цей алгоритм знову і знову, ми нарешті отримаємо найбільш оптимальну архітектуру ЗНМ.

При з'єднанні двох випадкових блоків важливу роль відіграє збіг глибини та розмірності вихідних даних першого блоку з глибиною та розмірністю фільтрів другого. Блок повинен мати можливість обробляти дані такого ж типу, що й очікуються в попередньому.

Для структурного синтезу рекомендується використовувати генетичні алгоритми.

Генетичні алгоритми належать до обчислювальної еволюції, яка є частиною штучного інтелекту. Вони натхненні процесами еволюції та природного відбору, де найсильніші ознаки передаються з покоління в покоління. Багатокритеріальний генетичний алгоритм (БКГА) є розширенням цього процесу. Його мета — одночасна оптимізація кількох об'єктивних функцій. Кожне рішення, запропоноване алгоритмом, пов'язане з набором значень цільових функцій. БКГА оптимізує ці значення та забезпечує набір Парето-оптимальних рішень.

У задачах багатокритеріальної оптимізації існує кілька конфліктних цілей, які необхідно оптимізувати. Це призводить до набору можливих рішень, відомих як рішення Парето, де жодне інше рішення не може одночасно покращити всі цілі. Таким чином, метою є не знаходження одного оптимального рішення, а генерування набору Парето-оптимальних рішень, які забезпечують компроміс між конфліктними цілями [14].

Основні сучасні еволюційні алгоритми включають:

Генетичні алгоритми (ГА):

- Працюють з популяцією кандидатів рішень.
- Використовують кросовер і мутацію для дослідження нових архітектур.
- Механізми відбору віддають перевагу індивідам з кращою продуктивністю.

Генетичне програмування (ГП) [4]:

- Подібно до ГА, але працює зі структурами, схожими на дерева.
- Може еволюціонувати цілі програми або архітектури.

NeuroEvolution of Augmenting Topologies (NEAT) [5]:

- Еволюціонує як ваги, так і архітектуру нейронних мереж.
- Вводить інновації шляхом відстеження генів і структур протягом поколінь.

Еволюційні стратегії (ЕС) [6]:

- Орієнтовані на оптимізацію безперервних параметрів.
- Використовують стратегії, такі як адаптація коваріаційної матриці [10], для керування пошуком.

Багатокритеріальні генетичні алгоритми, такі як NSGA-II та SPEA3, продемонстрували високу ефективність у різних інженерних задачах оптимізації. Використовуючи оператори відбору, мутації та кросинговеру в ітераціях, можна створювати конкурентоспроможних індивідів, спираючись на еволюційну теорію "виживання найсильнішого". Ці індивіди, які не можуть перевершити один одного за всіма показниками, формують групу, відому як "фронт недомінованості". З точки зору фізичних задач оптимізації, у яких оцінки завжди обчислювально складні, розмір популяції у MOGA зазвичай обмежений через обмежені обчислювальні ресурси.

Для структурного синтезу цільових ГЗНМ пропонується використовувати модифікований алгоритм SPEA-3, щоб подолати згадані проблеми.

Послідовність генів називається хромосомою. Найменша одиниця в хромосомах називається алелем, представленим одним символом або

двійковим бітом. Фенотип дає повний опис одиниці, тоді як генотип зберігає інформацію в хромосомі.

Окремі одиниці в групі формують популяцію. Далі оцінюється придатність кожної одиниці в популяції. Одиниці та їх інформація у просторі пошуку визначаються параметрами фенотипу.

Кросовер. Це довільно обраний елемент у кодованому бітовому рядку, і точна кількість бітів до і після цього елементу фрагментується і обмінюється інформацією з хромосомами попередників. Потомство формується шляхом об'єднання фрагментів бітових рядків попередників. Щоб усі нащадки були продуктом кросовера, ймовірність перехрестя повинна бути 100%, при значенні в 0%, хромосома теперішнього потомства буде точною копією попереднього покоління.

Причиною використання процесу кросоверів є відтворення найкращих хромосом, що містять у собі позитивні частини попередніх хромосом. Вживання деякого сегменту старої популяції в наступному поколінні дозволяється процесом відбору в кросоверах. Операції кросовера поділяються на наступні підвиди: двоточковий, багатоточковий, однорідний та кросовер зі зменшеним сурогатом. Одноточковий кросовер вважається кращим, оскільки він не руйнує структурну цілісність.

Мутація. Це створення потомства від одного з попередників шляхом інвертування одного або декількох випадково вибраних бітів у хромосомах попередників. Мутація може бути досягнута на будь-якому біті з невеликою ймовірністю, наприклад 0,001. Рядки, отримані в результаті кросовера, мутуються, щоб уникнути локального мінімуму. Генетичні матеріали, які можуть бути втрачені в процесі перехрещення та спотворення генетичної інформації, фіксуються шляхом мутації. Ймовірність мутації (μ) відповідає за те, наскільки частою буде розділення хромосоми, що зазнає мутації.

1.4.3. Багатокритеріальні генетичні алгоритми (NSGA, MOEA, SPEA3)

Багатокритеріальний генетичний алгоритм (БКГА) є розширенням цього процесу. Він фокусується на оптимізації декількох цілей одночасно. Кожне рішення, що надається алгоритмом, пов'язане з набором значень цільової функції. БКГА оптимізує ці значення і надає набір Парето-оптимальних рішень.

Перевага БКГА полягає в тому, що він досліджує глобальні рішення, не обмежуючись локальними мінімумами і максимумами, включаючи можливість одночасної обробки численних параметрів. У застосуванні до нейронних мереж БКГА може допомогти визначити оптимальний набір ваг і зсувів у мережі. Вони також можуть бути корисними в таких питаннях, як рішення щодо дизайну мережі, наприклад, вибір правильної кількості прихованих шарів і вузлів у кожному шарі.

Використовуючи БКГА, ми можемо оптимізувати нейронні мережі, щоб вони працювали краще, робили точніші прогнози і, можливо, були більш ефективними в часі.

Багатокритеріальний генетичний алгоритм (МОГА) - це тип штучного інтелекту, зокрема в галузі еволюційних обчислень, який призначений для оптимізації декількох цільових функцій одночасно. Він походить від основних концепцій біологічної еволюції та генетики, зосереджуючись на таких поняттях, як успадкування, мутація, селекція та кросинговер (або рекомбінація).

У багатокритеріальних оптимізаційних задачах є кілька конфліктуючих цілей, які необхідно оптимізувати. Це призводить до набору можливих рішень, відомих як рішення Парето, коли жодне інше рішення не може покращити всі цілі одночасно. Тому мета полягає не в тому, щоб знайти одне оптимальне рішення, а в тому, щоб згенерувати набір Парето-оптимальних рішень, що забезпечують компроміс між конфліктуючими цілями.

Загальна робота МОГА складається з наступних кроків:

Ініціалізація: Починається з випадково зібраної популяції індивідів, де кожен індивід (також відомий як хромосома) представляє можливе рішення багатокритеріальної задачі.

Оцінка: Потім кожна особа оцінюється на основі цільових функцій.

Відбір: Особини відбираються на основі їхньої придатності, як правило, ті, що мають вищу придатність, мають більше шансів бути відібраними для наступного покоління. Часто використовуються такі методи, як турнірний відбір або відбір за допомогою колеса рулетки.

Перехрещення: Відібрані особини проходять операцію схрещування. Обираються пари особин, і різні елементи (гени) від кожного з батьків поєднуються для отримання нащадків.

Мутація: Вносить невеликі випадкові зміни в потомство. Це допомагає підтримувати генетичне різноманіття в популяції і запобігає передчасній конвергенції.

Заміщення: Відбувається процес заміщення, коли деякі особини замінюються нащадками залежно від їхніх пристосованості.

Припинення: Алгоритм зупиняється або коли вироблено максимальну кількість поколінь, або коли досягнуто задовільного рівня пристосованості популяції.

Кінцевим результатом застосування МОГА є набір рішень, які є оптимальними з точки зору компромісу між кількома цілями. Це пропонує відповідний підхід до вирішення складних реальних проблем, включаючи розподіл ресурсів, складання розкладу та багато інших, де потрібно одночасно оптимізувати декілька цілей.

Таким чином, виникає потреба у використанні таких підходів, які б не мали зазначених недоліків. Базовий (класичний, елементарний або простий) генетичний алгоритм складається з наступних кроків:

- 1) вибір початкової популяції хромосом;
- 2) оцінка придатності хромосом у популяції;
- 3) перевірка умови зупинки алгоритму;

- 4) відбір хромосом;
- 5) застосування генетичних операторів;
- 6) формування нової популяції;
- 7) вибір "найкращої" хромосоми.

Багатокритеріальні генетичні алгоритми, такі як NSGA-II1, MOEA/D2, SPEA3, показали хорошу продуктивність у багатьох задачах інженерної оптимізації. Натхненні еволюційною теорією "виживання найсильніших", конкурентні особини можуть бути отримані за допомогою операторів відбору, мутації та кросинговеру шляхом ітерацій. Ці особини, які не можуть перевершити одна одну за всіма показниками, створюють множину, так званий фронт недомінування. З точки зору задач фізичної оптимізації, в яких оцінки завжди обчислювально складні, розмір популяції в МОГА, як правило, невеликий через обмеженість обчислювальних ресурсів.

Тема навчання нейронних мереж наразі є дуже актуальною, тому з'являється декілька теорій, що базуються на багатокритеріальній оптимізації, таких як модель MOBJ та LASSO (останній оператор абсолютної усадки та відбору), але пропонують розв'язувати проблему за допомогою моноцільової оптимізації. Наш підхід полягає у моделюванні та розв'язанні задачі узагальнення ЗНМ суто багатокритеріальною методологією, що включає дві цілі: мінімізувати загальну похибку ЗНМ, отримати оптимальні значення швидкодії мережі, при цьому розв'язок задачі є множиною недомінованих розв'язків за концепцією Парето на базі алгоритму SPEA3 (Strength Pareto Evolutionary Algorithm 3).

SPEA3 використовує регулярну популяцію та архів (зовнішню множину). Починаючи з початкової популяції та порожнього архіву, на кожній ітерації виконуються наступні кроки. По-перше, всі недоміновані члени популяції копіюються до архіву; будь-які доміновані особини або дублікати (щодо об'єктивних значень) видаляються з архіву під час цієї операції оновлення. Якщо розмір оновленого архіву перевищує заздалегідь визначену межу, подальші члени архіву видаляються за допомогою техніки

кластеризації, яка зберігає характеристики недовіреного фронту. Після цього значення пристосованості присвоюються як членам архіву, так і членам популяції:

- Кожній особині в архіві присвоюється значення сили $S(i) \in [0, 1)$, яке одночасно є значенням її пристосованості $F(i)$. $S(i)$ - це кількість членів популяції j , які домінують або дорівнюють i за об'єктивними показниками, поділена на розмір популяції плюс один.

- Сила $F(j)$ особи j в популяції обчислюється шляхом підсумовування значень сили $S(i)$ всіх членів архіву i , які домінують або дорівнюють j , і додавання одиниці в кінці.

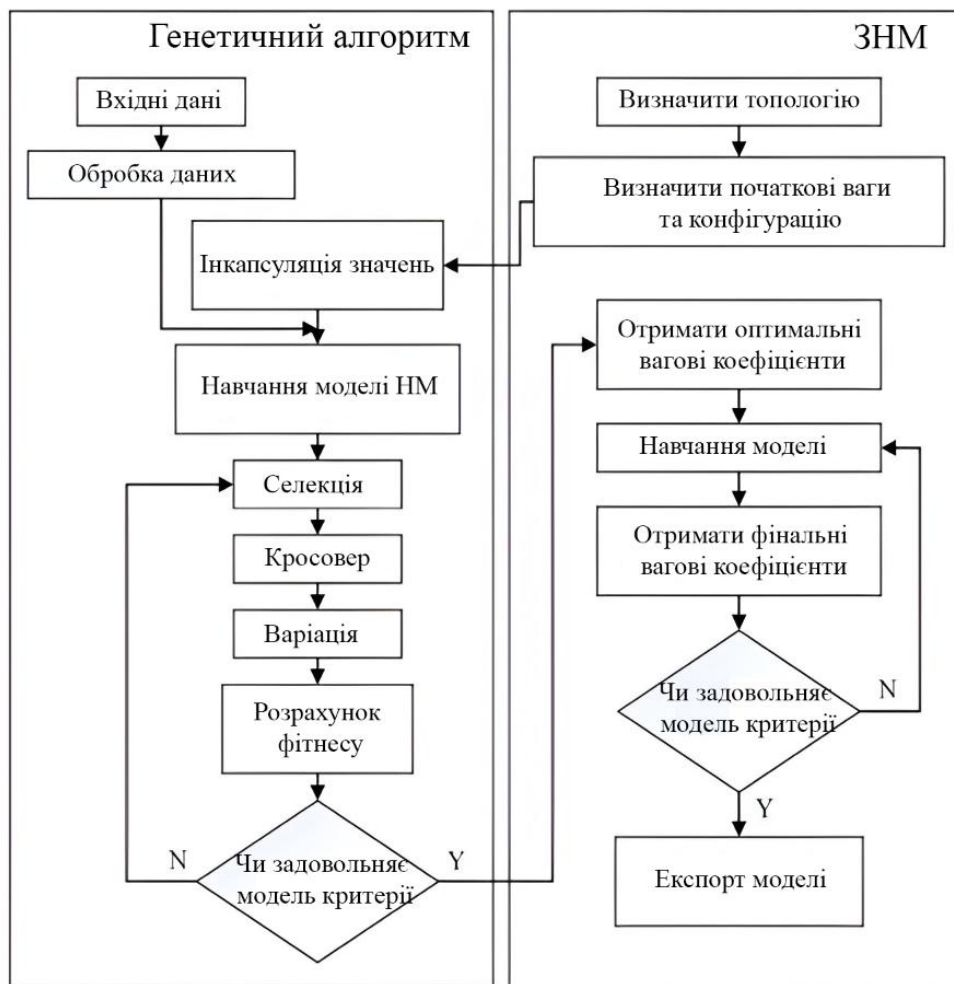


Рис. 1.7. Алгоритмічна схема застосування багатокритеріального генетичного алгоритму для отримання оптимальної топології НМ

Наступний крок являє собою фазу відбору для спарювання, де особини з об'єднання популяції та архіву обираються за допомогою бінарних турнірів. Зверніть увагу, що пристосованість тут має бути мінімізована, тобто кожна особина з архіву має більше шансів бути обраною, ніж будь-яка особа з популяції. Нарешті, після рекомбінації та мутацій стара популяція замінюється популяцією нащадків.

Для прикладу, на Рис. 1.9 показано логічно-послідовну схему застосування багатокритеріального генетичного алгоритму в задачах структурно-параметричного синтезу нейронних мереж.

Розподіл за фітнесом: Особини, над якими домінують одні й ті ж члени архіву, мають однакові значення фітнесу. Це означає, що у випадку, коли архів містить лише одну особину, всі члени популяції мають однаковий ранг, незалежно від того, домінують вони один над одним чи ні. Як наслідок, тиск відбору значно зменшується, і в цьому випадку SPEA поводить як алгоритм випадкового пошуку.

Оцінка щільності: Якщо багато особин поточного покоління індиферентні, тобто не домінують одна над одною, то на основі часткового порядку, визначеного відношенням домінування, не можна отримати жодної або дуже мало інформації. У цій ситуації, яка дуже ймовірна за наявності більше двох цілей, необхідно використовувати інформацію про щільність, щоб ефективніше спрямовувати пошук. Кластеризація використовує цю інформацію, але тільки стосовно архіву, а не населення.

Усічення архіву: Хоча техніка кластеризації, що використовується в SPEA, здатна скоротити недовідану множину, не знищуючи її характеристик, вона може втратити зовнішні розв'язки. Однак ці рішення слід зберігати в архіві для того, щоб отримати хороший розподіл недовідованих рішень.

багатокритеріальна оптимізація - це область пошуку балансу між декількома цільовими функціями таким чином, що жоден критерій не може

бути покращений без погіршення, принаймні, одного з інших критеріїв, ця рівновага називається оптимумом Парето.

багатокритеріальну задачу можна представити наступним чином: Знайти вектор $x^* = [x_1^*, x_2^*, \dots, x_n^*]^T$, який задовольняє обмеженням m нерівностей і p рівностей наступним чином:

$$\begin{aligned} g_i(x) &\geq 0 & i = 1, \dots, m \\ h_i(x) &= 0 & i = 1, \dots, p \end{aligned} \quad (1.5)$$

Оптимізуючи (мінімізуючи або максимізуючи), вектор наступних функцій:

$$f(x) = [f_1(x), f_2(x), \dots, f_n(x)]^T \quad (1.6)$$

Порівняння між можливими розв'язками здійснюється за допомогою процесу домінування, який визначається Розв'язок x домінує над розв'язком y тоді і тільки тоді, коли

$$\forall i \in (1, 2, \dots, k): f_i(x) \leq f_i(y) \quad (1.7)$$

і

$$\exists j \in (1, 2, \dots, k): f_j(x) < f_j(y) \quad (1.8)$$

Використовуючи цю концепцію, рішення x^* є недомінованим рішенням або оптимумом Парето, якщо в просторі рішень не існує рішення, яке б домінувало над ним. Отже, оптимальна за Парето множина багатокритеріальної задачі - це множина всіх оптимальних за Парето розв'язків задачі

$$POS = \{x \in S / \exists y \in S, f(y) < f(x)\} \quad (1.9)$$

Образ оптимальної за Парето множини у просторі цілей називається "фронтом Парето", форма цієї границі приймає різні форми в залежності від того, чи повинні цілі бути мінімізувати чи максимізувати [17, 19].

Для розв'язання багатокритеріальної задачі методом обмеженої оптимізації було запропоновано багато алгоритмів, таких як еліпсоїдний

алгоритм [9]. Альтернативні багатокритеріальні алгоритми включають багатокритеріальний алгоритм ковзного режиму [10], багатокритеріальний алгоритм Левенберга-Марквардта [11] та багатокритеріальні еволюційні алгоритми [12]. Дослідження довело, що генетичний алгоритм є більш придатним для використання для подібної задачі, тому було розроблено багато варіацій, таких як VEGA [14] (неаггеговані непарето-методи), MOGA [15], NSGA [16], NPGA [17] (засновані на Парето-неелітарних методах); SPEA [18], PAES [19], PESA [20], NSGA II [21] (Парето-елітарні методи).

Ефективність узагальнення нейронної мережі визначається її здатністю навчатися на наборі даних, мінімізуючи вплив стохастичної змінної, пов'язаної з шумом. Ця ефективність може бути виміряна величиною помилки валідаційного набору порівняно з навчальним. У загальному випадку фінальна модель повинна забезпечувати баланс між похибкою навчальної та валідаційної вибірки.

Локальна помилка, помилка, що відповідає одному прикладу, парі (x, d) , така, що:

x : Вхідний вектор;

d : Бажаний вихідний вектор;

Таким чином, локальна похибка визначається за наступною формулою:

$$\varphi_j(w) = \frac{1}{2} \sum_{i=1}^N \left(d_i^j - g(w, x_j) \right)^2 \quad (1.10)$$

Таке, що:

N - розмір вихідного значення;

d_i^j - вихід шуканого векторного виходу d_j ;

w - ваги мережі;

$g(w, x_j)$ - вихід мережі, пов'язаний з вхідним вектором x_j .

Тоді глобальну похибку навчальної вибірки можна визначити як

$$E(w) = \frac{1}{k} \sum_{j=1}^k \varphi_j(w) \quad (1.11)$$

Топологія мережі також може бути обрізана для покращення узагальнення, для цього ми пропонуємо усунути ваги надмірно

детермінованих мереж, для кращого балансу між гнучкістю та жорсткістю багатошарового персептрону, що представлено наступною формулою:

$$J(w) = \sum_{i=1}^n |C_i w_i|, \quad (1.12)$$

де n – розмір вектору вагових коефіцієнтів, C_i - змінна рішення така, що

$$C_i = \begin{cases} 1 & \text{якщо } w_i \text{ зберігається в архітектурі НМ} \\ 0 & \text{якщо ні} \end{cases}$$

багатокритеріальна модель намагається оптимізувати дві суперечливі цілі: похибку та суму абсолютних ваг, функції витрат. багатокритеріальна модель представлена наступним чином:

$$(P) \begin{cases} \min \sum_j^k \sum_i^N (d_i^j - g(w'_i, x_j))^2 \\ \min \sum_{i=1}^n |w'_i| \end{cases}, \text{ де} \quad (1.13)$$

$$w'_i = |C_i w_i|$$

Однією з варіацій багатокритеріального ГА є NSGA [16], популярний генетичний алгоритм для багатокритеріальної оптимізації, заснований на недомінуванні. Це дуже ефективний алгоритм, але його часто критикують за обчислювальну складність, відсутність елітарності та за вибір оптимального значення параметра розподілу σ_{share} . Було розроблено модифіковану версію NSGAII [21], одного з найефективніших багатокритеріальних еволюційних алгоритмів [24], який має кращий алгоритм сортування, включає елітарність і не потребує апріорного вибору параметра розподілу. Він використовує відстань скупченості, яка оцінює щільність рішень у цільовому просторі, та оператор переповненого порівняння, який спрямовує процес відбору до рівномірно розподіленої Парето-межі.

Загальна процедура NSGA II виглядає наступним чином:

Крок 1: Створити випадкову батьківську популяцію P_t розміру Z ;

Крок 2: відсортуйте випадкову батьківську популяцію на основі концепції недомінування;

Крок 3: кожному недомінованому розв'язку присвоїти ранг, що дорівнює його рівню недомінування, 1 - найкращий рівень, 2 - наступний найкращий рівень і так далі;

Крок 4: Створіть популяцію нащадків Q_t , використовуючи оператори відбору та продукування наступним чином:

- 1) Відбір: крок порівняння, що застосовується для вибору батьківських рішень, визначається значенням рангу, якщо n рішень мають однакове значення рангу, застосовується оператор переповненого порівняння на основі відстані скупчення, рішення з найкращою відстанню скупчення зберігається;
- 2) Схрещування: вибрати батьківські розв'язки для схрещування;
- 3) Мутація: вибрати батьківські розв'язки для мутації;

Крок 5: створіть пул спарювання R_t , об'єднавши батьківську популяцію P_t та популяцію нащадків Q_t ;

Крок 6: відсортуйте об'єднану популяцію R_t відповідно до швидкої процедури сортування без домінування, щоб виявити всі недомінуючі фронти $(F_{r1}, F_{r2}, \dots, F_{rm})$;

Крок 7: Згенеруйте нову батьківську популяцію P_{t+1} розміру Z , вибираючи недоміновані розв'язки, починаючи з першого за рангом недомінованого фронту F_{r1} . Коли розмір популяції z перевищено, відкиньте деякі з нижчих за рангом недомінованих розв'язків;

Крок 8: Повторюйте з кроку 3, поки не буде досягнуто критерію зупинки.

Для мінімізації помилки навчання нейронних мереж на основі багатокритеріального генетичного алгоритму кожному варіанту вектора вагових коефіцієнтів W ставиться у відповідність деяка хромосома, представлена у вигляді бітового рядка. У процесі вибору здійснюється спрямований пошук хромосом, які забезпечують екстремум заданої цільової функції, оскільки в процедурі навчання нейронної мережі використовується функція помилки навчання нейронної мережі α .

Поняття навчання нейронної мережі при використанні генетичного алгоритму, на відміну від традиційних методів навчання, має інший зміст: навчання тут полягає в застосуванні генетичних операторів до генотипу вектора Wi , тобто до хромосоми, а навчальна вибірка служить для

обчислення помилки навчання α нейронної мережі при конкретних значеннях вагових коефіцієнтів W_i .

Таким чином, процедура оптимізації процесу навчання нейронної мережі за допомогою генетичного алгоритму також є ітераційною і включає етапи синтезу нових хромосом та їх відбору в нову популяцію.

Процес триває доти, доки не буде отримано набір оптимальних розв'язків Патеро або задану кількість поколінь. При цьому кожна наступна популяція повинна бути кращою за попередню. Розв'язком цієї задачі є хромосома з мінімальним значенням функції належності, яка визначає оптимальний вектор вагових коефіцієнтів W_i , при якому помилка навчання α стає меншою за задане значення δ . Якщо оптимальний розв'язок не знайдено, а задана кількість поколінь досягнута, то процедура навчання закінчується відбором елітної хромосоми в одному або декількох поколіннях.

Операція кросовер

Під час кросинговеру гени в кожному геномі вирівнюються на основі їх історичного маркера в процесі, який називається штучним синапсисом [14]. Гени, які мають спільне походження, називаються відповідними генами; ті, що не мають, називаються роз'єднаними генами або надлишковими генами, залежно від того, чи існують вони всередині або за межами діапазону інноваційних чисел іншого з батьків, відповідно. Таким чином, хоча два батьки можуть виглядати (і, ймовірно, виглядають) по-різному, їхні історичні маркери дозволяють знайти структурну схожість без будь-якого (ймовірно, дорогого) топологічного аналізу. Відповідні гени успадковуються випадковим чином. Надлишкові гени та гени, що не з'єднуються, успадковуються від більш пристосованого батька.

Штучний синапсис допомагає протидіяти негативним побічним ефектам, спричиненим проблемою конкуруючих конвенцій. Сумісні організми спаровуються таким чином, щоб зберегти функціональні субодиниці; потомство, швидше за все, буде неушкодженим. Хоча конкуруючі конвенції все ще можуть існувати в популяції, NEAT не витрачає

час на оцінку пошкодженого потомства, і таким чином вдається уникнути головного наслідку конкуруючих конвенцій.

Операція видоутворення

Геномам дозволено конкурувати з популяційними нішами, групуючи їх у види. Хоча це здається проблемою топологічного зіставлення, історичні маркери знову дають корисне рішення. Топологічну розбіжність між двома нейронними мережами можна кількісно оцінити за допомогою кількості надлишкових і роз'єднаних генів між їхніми геномами. Стенлі пише: "Чим більше роз'єднані геноми, тим менше спільної еволюційної історії вони мають, а отже, тим менш сумісними вони є" [14]. Кількість спільної еволюційної історії між двома геномами вимірюється за допомогою функції сумісності:

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \cdot \bar{W} \quad (1.14)$$

де δ - відстань сумісності, E - кількість надлишкових генів, D - кількість роз'єднаних генів, $\bar{W}$ - середня різниця ваг відповідних генів, а коефіцієнти c_1 , c_2 і c_3 дозволяють нам регулювати відносну важливість трьох факторів. Сумісність використовується як жорсткий пороговим значенням; таким чином, як тільки організм перетинає його, створюється новий вид.

Мутації

У генетичному алгоритмі існує чотири основних типи мутаційних операцій: (1) додавання нейрона, (2) додавання зв'язку, (3) видалення вузла, (4) видалення зв'язку та (5) зміна синаптичної ваги. Ймовірність виникнення кожної мутації контролюється програмістом за допомогою конфігураційного файлу. При додаванні нейрона вибирається випадковий зв'язок, який замінюється новим нейроном і двома новими зв'язками. Ваги двох нових зв'язків підбираються таким чином, щоб вони були фактично такими ж, як і у заміненого зв'язку, щоб уникнути надмірно різких рухів по фітнес-ландшафту. Додавання нового зв'язку працює аналогічно: випадковим чином вибираються два нейрони, між якими додається новий зв'язок. Виконуються перевірки, щоб переконатися, що між нейроном-джерелом і нейроном-

приймачем ще не існує зв'язку. Це також підкреслює дві цікаві особливості NEAT: приховані нейрони не впорядковані в традиційній ієрархії прихованих рівнів і, відповідно, рекурентні нейронні мережі можуть еволюціонувати (і в ідеалі будуть, якщо це вирішить проблему більш ефективно, ніж нейронна мережа прямого поширення). Наприклад, при додаванні нового зв'язку дозволяється вибирати один і той самий нейрон в якості джерела і призначення.

Видалення нейронів і зв'язків виконується аналогічно. Як зв'язки, так і нейрони можуть бути вибрані для видалення випадковим чином. Приховані нейрони видаляються, якщо не залишилося зв'язків, що взаємодіють з ними. Коли синаптична вага випадковим чином вибирається для збурення, до неї додається випадкове число з нуль-центрованого нормального розподілу. Стандартне відхилення нормального розподілу також задається програмістом. Аналіз вибору параметрів наведено в розділі 5.

Ускладнення

Багато генетичних алгоритмів починаються з випадкового набору початкових організмів [5]. Це створює проблему при еволюції нейронних мереж: цілком можливо, що організм буде роз'єднаним графом (тобто не кожен вузол буде з'єднаний з мережею, що містить вихід). Це займе певний час, щоб видалити його з популяції.

Стенлі [14] вказує на більш серйозну проблему: "Початок роботи з випадковими топологіями не призводить до пошуку мінімальних рішень, оскільки популяція починає роботу з багатьма непотрібними вузлами і зв'язками, які вже присутні". Іншими словами, випадково згенеровані вузли та зв'язки не мають жодного обґрунтування для свого існування. Невідомо, чи сприяють вони розв'язанню задачі. І навпаки, всі наступні структурні доповнення проходять оцінку за характером того, як вони змінюють пристосованість організму. Початок з мінімальної структури дозволяє алгоритму шукати рішення в низьковимірному просторі пошуку, що значно покращує продуктивність [14].

1.4.4. Визначення параметричного синтезу згорткових нейронних мереж

Параметричний синтез - це підхід до налаштування згорткової нейронної мережі для досягнення найкращої продуктивності. Список параметрів, який слід налаштувати, можна розділити на дві групи. Перший - це локальні параметри шарів згортки, шарів агрегації та інші локальні значення для структурних блоків ЗНМ. Другий - це глобальні значення, такі як швидкість навчання, розмір шагу, зашумлення вхідних даних, коефіцієнт скорочення тощо.

Локальні параметри включають:

- Кількість фільтрів: Необхідно визначити для кожного згорткового шару кількість фільтрів та нейронів. Цей параметр має вирішальне значення для передачі інформації до глибших шарів, оскільки занадто мала кількість фільтрів може призвести до значних втрат інформації саме в цьому шарі.
- Розмір ядра: Також для кожного згорткового шару ми повинні визначити розмір ядра. З міркувань узгодженості всі ядра в шарі мають однаковий розмір, проте оптимальний розмір залежить від параметрів вхідних даних та загальної структури ЗНМ.
- Розмір кроку: Розмір зсуву, відстань на яку фільтр рухається з кожним кроком.
- Функція агрегації.

Висновки до розділу

У розділі проведено всебічний аналіз сучасних архітектур ЗНМ та алгоритмів їх структурного синтезу. Визначено ключові тенденції розвитку ЗНМ. Окреслено основні проблеми синтезу оптимальних архітектур нейромереж, серед яких — висока обчислювальна складність, потреба у великій кількості даних для навчання та залежність продуктивності від архітектурних параметрів. Обґрунтовано перспективність використання БКГА.

РОЗДІЛ 2

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В СИСТЕМАХ ВІРТУАЛЬНОЇ ТА ДОПОВНЕНОЇ РЕАЛЬНОСТЕЙ

2.1. Віртуальна реальність

Віртуальна реальність (VR) - це технологія, яка швидко розвивається і дозволяє користувачам відчувати і взаємодіяти з цифровим тривимірним середовищем. Використовуючи поєднання апаратного та програмного забезпечення, користувачі можуть зануритися в змодельований світ, який створений максимально реалістично. З розвитком апаратних і програмних технологій віртуальна реальність стала життєздатним варіантом для багатьох застосувань, починаючи від ігор і закінчуючи медичними симуляціями.

У найпростішій формі віртуальна реальність - це комп'ютерне середовище, створене для імітації реального життя. Це досягається за допомогою комп'ютерної графіки, звуку та інших сенсорних даних, таких як дотик або запах. Користувачеві надається змодельоване середовище, з яким він може взаємодіяти різними способами. Залежно від типу використовуваної системи віртуальної реальності, користувачі можуть виконувати широкий спектр дій, від ігор до дослідження симуляцій реального середовища.

Додатки віртуальної реальності розробляються для широкого спектру цілей, включаючи ігри, освіту, медицину, маркетинг і розваги. В ігровій індустрії технологія VR використовується для створення імерсивного досвіду, який дозволяє гравцям повністю зануритися в ігровий світ. В освіті VR можна використовувати для створення симуляцій реальних середовищ, які можуть допомогти студентам навчатися ефективніше. У медицині VR використовується для створення симуляцій хірургічних процедур, що дозволяє лікарям практикувати без ризику для пацієнта. У маркетингу віртуальну реальність можна використовувати для створення інтерактивних презентацій, які можна використовувати для залучення потенційних клієнтів. Нарешті, у сфері розваг віртуальна реальність може



Рис. 2.1. Приклад використання систем віртуальної реальності для спец-тренування з використання тазер-зброї.

використовуватися для створення захоплюючих вражень, які дають глядачам відчуття, ніби вони насправді перебувають у зображуваному середовищі.

Апаратні та програмні технології, що використовуються в системах віртуальної реальності, постійно розвиваються. Найпоширеніші апаратні компоненти включають наголовні дисплеї, рукавички та інші датчики. Наголовні дисплеї (HMD) [11, 12] одягаються на голову і проєктують зображення прямо в очі користувача. Це дозволяє їм відчувати себе так, ніби вони насправді перебувають у віртуальному середовищі. Рукавички та інші датчики дозволяють користувачам взаємодіяти з навколишнім середовищем, виявляючи їхні рухи і перетворюючи їх на команди у віртуальному світі.

Програмні технології, що використовуються в системах віртуальної реальності, також постійно розвиваються. Сюди входить програмне забезпечення для 3D-моделювання, фізичні рушії та штучний інтелект. Програмне забезпечення для 3D-моделювання використовується для створення віртуальних середовищ, з якими взаємодіє користувач. Фізичні рушії використовуються для імітації реалістичних взаємодій між об'єктами у віртуальному світі. Нарешті, штучний інтелект використовується для

створення реалістичних персонажів і NPC, які можуть взаємодіяти з користувачем.

2.2. Доповнена реальність

Доповнена реальність (AR) - це технологія, що стрімко розвивається і має потенціал революціонізувати спосіб взаємодії з навколишнім середовищем. Доповнена реальність - це форма віртуальної реальності (VR), яка доповнює реальне середовище комп'ютерними доповненнями та об'єктами. Це поєднання фізичного та віртуального світів, що дозволяє користувачам взаємодіяти з 3D-об'єктами в їхньому природному середовищі.

На найпростішому рівні доповнена реальність - це використання комп'ютерних зображень для створення покращеної версії реальності. Це включає в себе додавання віртуальних об'єктів у реальне середовище, а також доповнення існуючих об'єктів. Доповнену реальність можна використовувати для додавання візуальних ефектів, таких як текст, зображення і відео, в середовище користувача. Вона також може бути використана для додавання аудіоефектів, таких як звук і музика.

Застосування доповненої реальності дуже широке. Наприклад, її можна використовувати в іграх, освіті, розвагах, навігації та медицині. В іграх доповнену реальність можна використовувати для створення інтерактивного середовища для гравців, наприклад, віртуального поля бою або гоночної траси. Її також можна використовувати для створення віртуальних об'єктів, які реагують на дії гравців. В індустрії розваг доповнену реальність можна використовувати для створення захоплюючих вражень, наприклад, віртуальних концертів або тематичних парків. У медицині її можна використовувати для створення захоплюючих симуляцій операцій або інших медичних процедур.

В основі технології доповненої реальності лежить здатність точно виявляти і відстежувати фізичні об'єкти в оточенні користувача. Це досягається завдяки використанню датчиків і камер. Датчики вимірюють



Рис. 2.2. Приклад додатку з використанням доповненої реальності для виводу елементів контролю над електричним двигуном

різні фізичні властивості, такі як рух, температура та інтенсивність світла. Камери фіксують візуальні дані, такі як колір, глибина і текстура. Потім ці дані аналізуються системою доповненої реальності для виявлення та відстеження фізичних об'єктів у навколишньому середовищі.

Окрім апаратних компонентів системи доповненої реальності, необхідне також програмне забезпечення, яке полегшує взаємодію між користувачем і віртуальними об'єктами. Сюди входить програмне забезпечення для відстеження об'єктів у навколишньому середовищі, генерування візуального та звукового супроводу, а також створення взаємодії між фізичним і віртуальним світами.

В останні роки розвиток мобільних технологій зробив доповнену реальність доступнішою для широкого загалу. Мобільні пристрої, такі як смартфони та планшети, тепер здатні запускати AR-додатки, дозволяючи користувачам взаємодіяти з віртуальними об'єктами у своєму оточенні. Це відкрило нові можливості в таких сферах, як ігри, освіта, навігація та розваги.

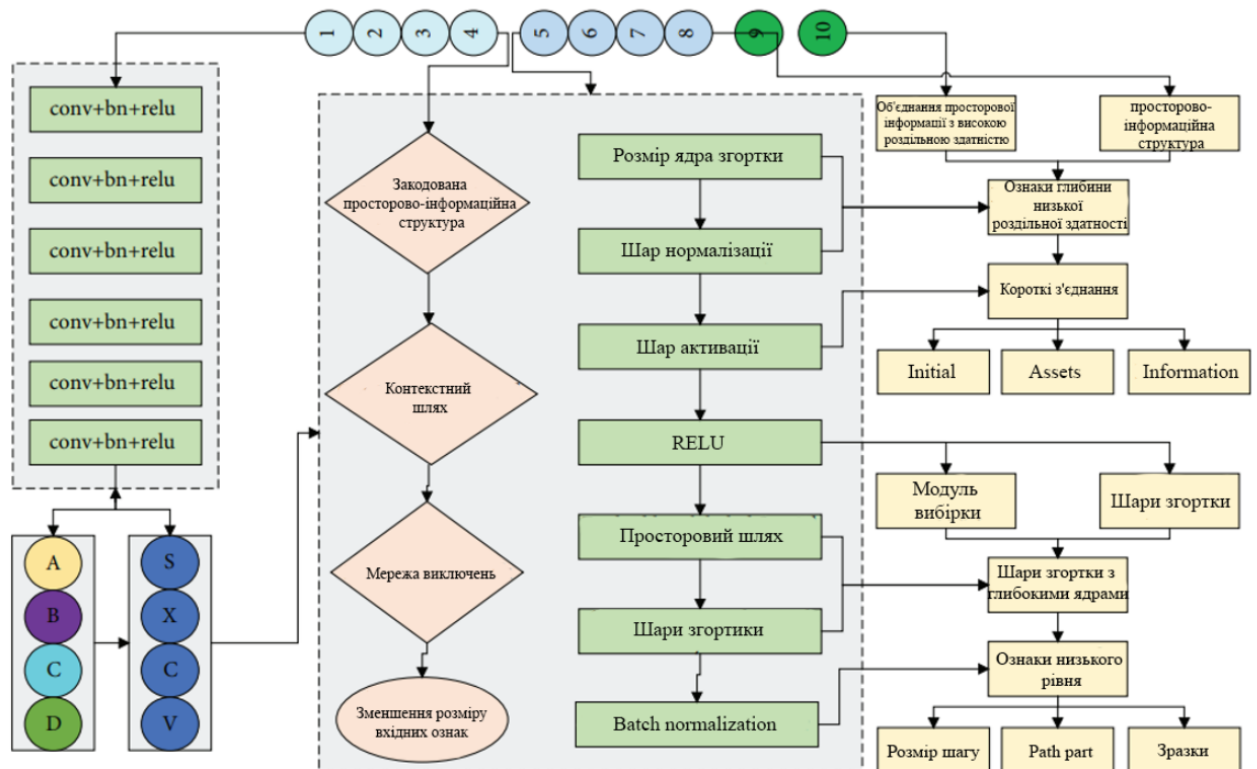


Рис. 2.3. Структурна схема системи інтеграції згорткової нейронної мережі з графічним двигуном віртуальної реальності

2.3. Застосування згорткових нейронних мереж для віртуальної та доповненої реальності

За останні роки використання технології віртуальної реальності (VR) значно зросло, а сфера її застосування варіюється від ігор і розваг до медичного та промислового навчання. Оскільки технологія віртуальної реальності продовжує вдосконалюватися, виникає потреба в розробці більш досконалих алгоритмів і методів, щоб максимально використати її потенціал. Однією з таких методик є використання згорткових нейронних мереж (ЗНМ) для віртуальної реальності.

У цій роботі буде надано огляд різних застосувань згорткових нейронних мереж для віртуальної реальності. Ми обговоримо різні типи ЗНМ, їх переваги та недоліки, а також те, як вони можуть бути використані для обробки даних з середовища віртуальної реальності. Ми також обговоримо виклики та поточні дослідження в цій галузі, а також потенційні майбутні застосування.

У зв'язку зі зростаючим поширенням віртуальної реальності, було проведено значну кількість досліджень щодо використання ЗНМ для додатків віртуальної реальності. Наприклад, дослідники розробили ЗНМ, які можуть виявляти об'єкти в сцені, такі як люди або меблі. Ці мережі можна використовувати для більш реалістичної взаємодії з віртуальним середовищем. Крім того, було розроблено мережі, які можуть розпізнавати форму об'єктів у віртуальному середовищі. Це може уможливити більш точну взаємодію з віртуальним середовищем, наприклад, ідентифікацію об'єктів, схожих за формою.

Незважаючи на ці досягнення, все ще існує кілька проблем, які необхідно вирішити, щоб повністю розкрити потенціал ЗНМ для віртуальної реальності. Наприклад, все ще потрібно розробити мережі, здатні точно розпізнавати об'єкти у віртуальному середовищі, навіть за наявності шуму або інших відволікаючих чинників. Крім того, ці нейронні мережі повинні мати можливість навчатися на налаштовуватись в процесі роботи, щоб з часом ставати більш точними.

Оскільки технологія віртуальної реальності продовжує розвиватися, існує потенціал для використання ЗНМ у різних сферах. Наприклад, ЗНМ можна використовувати для більш реалістичної взаємодії між користувачем і віртуальним середовищем. Наприклад, мережі можна використовувати для виявлення об'єктів у сцені або розпізнавання форми об'єктів.

Крім того, ЗНМ можна використовувати для більш точної навігації у віртуальному середовищі. Наприклад, мережі можна використовувати для виявлення перешкод або інших об'єктів у середовищі і визначення найкращого шляху для користувача. Це може бути корисно для таких застосувань, як медичне навчання або виробниче навчання.

На даний момент в сфері віртуальної реальності запропоновано підводну VR-систему для навчання дайвінгу, яка дозволить студентам підводного плавання відчути довільне підводне середовище та мати можливість використовувати технологію віртуальної реальності для

тренувань в умовах обмеженого водного простору. Система ImmerTai була розроблена для захоплення рухів експерта з тайцзи та зйомки в імерсивній партії CR для виконання вправ китайського тайцзи. Рухи студента також представляються в середовищі ПК у мультимодальному форматі, щоб доставити захоплені рухи студенту. Рухи студента також фіксуються для оцінки якості та використовуються для формування віртуального середовища для спільного навчання. На додаток до цього, пропонується система віртуальної реальності для навчання лижному спорту з використанням лижного тренажера в приміщенні. Система базується на простому лижному тренажері в приміщенні з двома трекарами для фіксації руху лиж. Користувачі можуть керувати лижами на наданих віртуальних гірськолижних схилах і можуть тренувати свої навички, відтворюючи виступи професійних лижників. Отримано вихідну функцію з 3 каналами та роздільною здатністю зображення $512 * 256$. Сполучені Штати є батьківщиною передових технологій віртуального моделювання, а Лабораторія Еймса НАСА є головним гравцем у цій галузі, завдяки чому інженерні застосування віртуального моделювання (наприклад, шоломи даних, рукавички даних, симуляції віртуальної реальності космічних станцій) стають все більш досконалими, а віртуальне моделювання космічного телескопа Хаббл є одним із технологічних досягнень, і НАСА створило відповідну систему навчання віртуальному моделюванню космічної науки, а також багатозадачні програми в операційній системі віртуального моделювання космічної станції. Завдяки своїм величезним можливостям у навчанні та тестуванні технологія віртуальної реальності часто використовується спочатку в таких секторах, як оборона та аерокосмічна промисловість, а потім поступово поширюється на такі галузі, як охорона здоров'я, освіта та розваги, які створили для неї величезні ринки збуту і отримали значні економічні вигоди. Всього за кілька десятиліть технологія віртуальної реальності стала все більш досконалою для кожного аспекту повсякденного життя кожного, даючи можливість менеджерам, лідерам та інженерам оперативно використовувати,

модифікувати та проектувати спроектовану систему. Деякі зрілі продукти для захоплення руху та інтерактивні рішення для віртуальної реальності не виправдовують очікувань з точки зору досвіду і їх важко популяризувати серед широкого загалу користувачів. Таким чином, якісний дизайн систем захоплення руху та розробка систем віртуальної взаємодії для галузі означає зниження економічних витрат та покращення користувацького досвіду. Будучи ключовим питанням у сфері віртуальної реальності, воно привертає багато уваги з боку дослідників. У системі візуалізації архітектури мікросервісу, що втілює конкретні додатки, розроблено спеціальний аналіз для інтеграції розподільчої мережі. Під час візуалізаційного представлення компоненти додатків спрощуються, що, в свою чергу, забезпечує більш осмислене уявлення для адміністраторів підприємства та підвищує ефективність управління з боку операційного та технічного персоналу. Тут осмисленість та ефективність беруться за основну відправну точку, і компоненти додатків ефективно відображаються, а потім менеджери підприємства можуть чітко ідентифікувати компоненти додатків, визначити, чи мають компоненти в системі архітектури велике прикладне значення, а потім знайти корисні елементи, пов'язані з ними, відповідно до прикладної цінності компонентів та їх значення, і, нарешті, візуалізувати архітектуру. По-друге, деякі інструменти візуалізації використовуються на практиці з чудовими результатами і постійно підвищують ефективність менеджерів з розробки для усунення проблем. Кількість каналів, отриманих шляхом згортки цієї вихідної ознаки з третьою 3×3 у кодері, становить 32, а роздільна здатність зображення - 512×256 . Ознаки зрощуються. За допомогою висхідної та низхідної діаграм залежностей архітектури системи розробники можуть швидко знайти джерело проблеми в разі збою за допомогою даних залежностей, що значно скорочує час усунення проблеми (MTTR). За допомогою діаграми архітектури взаємозв'язків додатків можна відобразити кореляцію між компонентами додатків в системі або виконати симуляцію

збоїв для кожного компонента, від якого залежить система, щоб оцінити надійність системи перед обличчям локальних збоїв.

2.4. Переваги використання згорткових нейронних мереж в технологіях віртуальної та доповненої реальності

2.4.1. Покращення імерсійності віртуальної реальності

Нейронні мережі у віртуальній реальності (VR) надають ряд переваг, найбільш помітною з яких є покращене занурення. Нейронні мережі використовуються для створення віртуальних середовищ, які є більш реалістичними, більш чутливими та інтерактивними.

Використання нейронних мереж у VR може допомогти створити більш реалістичну симуляцію фізичного середовища. Нейронні мережі можна використовувати для створення віртуальних середовищ, які є дуже деталізованими і реагують на дії користувача. Це може створити більш захоплюючий досвід для користувача, оскільки він може взаємодіяти з навколишнім середовищем у більш природний спосіб. Наприклад, нейронні мережі можна використовувати для створення віртуальних середовищ, які здатні розпізнавати і реагувати на рухи або жести користувача. Це може створити більш природний і реалістичний досвід для користувача, оскільки він може взаємодіяти з навколишнім середовищем у спосіб, який був би неможливий у традиційному середовищі віртуальної реальності.

Нейронні мережі також можна використовувати для створення більш інтерактивних віртуальних середовищ. Використовуючи нейронні мережі, розробники можуть створювати віртуальні середовища, які можуть реагувати на дії користувача в режимі реального часу. Це може створити більш захоплюючий досвід для користувача, оскільки він може взаємодіяти з середовищем у більш осмислений спосіб. Наприклад, нейронні мережі можна використовувати для створення віртуальних середовищ, які здатні реагувати на голосові команди або міміку користувача. Це може створити більш

інтерактивний досвід для користувача, оскільки він зможе взаємодіяти з навколишнім середовищем у більш природний спосіб.

Нейронні мережі у віртуальній реальності також можна використовувати для створення більш інтелектуальних віртуальних середовищ. Нейронні мережі можна використовувати для створення віртуальних середовищ, які здатні навчатися на основі даних, введених користувачем, а потім використовувати цю інформацію для прийняття рішень або виконання дій. Наприклад, нейронні мережі можна використовувати для створення віртуальних середовищ, які здатні розпізнавати рухи або вираз обличчя користувача, а потім використовувати цю інформацію для прийняття рішень або виконання дій. Це може створити більш інтелектуальний досвід для користувача, оскільки він зможе більш осмислено взаємодіяти з навколишнім середовищем.

Хоча користувачі віртуальної реальності можуть візуально відчувати себе в імерсивному середовищі, існує недостатньо досліджень щодо узгодженості дій користувачів віртуальної реальності у віртуальному та реальному середовищах. Тому стає важливим дослідити, чи можуть користувачі віртуальної реальності виконувати дії у віртуальному середовищі так само, як вони це роблять у реальному середовищі.

Загалом, нейронні мережі у віртуальній реальності надають низку переваг, найпомітнішою з яких є покращене занурення. Використовуючи нейронні мережі, розробники можуть створювати віртуальні середовища, які є більш реалістичними, більш чутливими та інтерактивними. Це може створити більш захоплюючий досвід для користувача, оскільки він може взаємодіяти з навколишнім середовищем у більш природний спосіб. Крім того, нейронні мережі можна використовувати для створення більш інтелектуальних віртуальних середовищ, оскільки вони здатні навчатися на основі вводу користувача, а потім використовувати цю інформацію для прийняття рішень або виконання дій. Таким чином, нейронні мережі у

віртуальній реальності надають ряд переваг, які можуть допомогти створити більш захоплюючий і цікавий досвід для користувача.

2.4.2. Покращення інтерактивності віртуальної реальності

Згорткові нейронні мережі (ЗНМ) можна використовувати для покращення інтерактивності в додатках віртуальної реальності (VR). ЗНМ - це тип штучних нейронних мереж, які особливо добре підходять для задач обробки зображень, таких як розпізнавання об'єктів або аналіз сцен. Використовуючи ЗНМ, розробники VR можуть створювати більш інтерактивні додатки, надаючи користувачам більш захоплюючий досвід.

Однією з головних переваг використання ЗНМ для VR-додатків є їхня здатність швидко розпізнавати об'єкти і сцени. За допомогою традиційних методів комп'ютерного зору розпізнавання об'єктів і сцен може зайняти значну кількість часу, оскільки комп'ютер повинен обробляти кожне зображення окремо. Однак ЗНМ можуть швидко розпізнавати об'єкти і сцени, оскільки вони здатні розпізнавати патерни на зображеннях. Це дозволяє користувачам взаємодіяти з навколишнім середовищем у більш природний спосіб, оскільки вони можуть швидко розпізнавати об'єкти і сцени, не чекаючи, поки комп'ютер їх наздожене.

На додаток до покращення інтерактивності VR-додатків, ЗНМ також можна використовувати для створення більш реалістичних віртуальних середовищ. Наприклад, ЗНМ можна використовувати для створення фотореалістичних зображень з реальних фотографій, що дозволяє користувачеві відчувати більш реалістичне представлення реального світу. Це може бути особливо корисно для таких додатків, як навчальні симуляції, оскільки дозволяє користувачам практикуватись у реалістичному середовищі, яке дуже нагадує реальний світ.

Нарешті, ЗНМ можна використовувати для створення більш адаптивних VR-додатків. Використовуючи ЗНМ, розробники можуть створювати додатки, які можуть швидко реагувати на введення користувача,

оскільки ЗНМ зможуть швидко розпізнавати шаблони у введенні користувача і реагувати відповідно. Це дозволяє користувачам відчувати більш чуйний і захоплюючий досвід, оскільки додаток зможе швидко реагувати на їх введення.

2.4.3. Зменшення витрат при розробці технологій віртуальної реальності

Однією з найбільших переваг використання нейронних мереж у віртуальній реальності є економія коштів, яку вони забезпечують. Нейронні мережі вимагають мінімальних апаратних і програмних ресурсів, що робить їх набагато дешевшими, ніж традиційні обчислювальні методи. Це особливо важливо для додатків віртуальної реальності, оскільки вони часто вимагають великих обчислювальних потужностей. Економія коштів за рахунок використання нейронної мережі може бути особливо значною для масштабних проєктів, наприклад, для навчальних симуляторів та тренажерів.

Нейронні мережі також здатні знизити вартість розробки для VR-додатків. Нейронні мережі набагато простіше програмувати, ніж традиційні обчислювальні методи, і вони можуть допомогти автоматизувати багато завдань розробки. Це зменшує кількість часу і грошей, витрачених на розробку, роблячи створення VR-додатків більш рентабельним.

На додаток до зниження витрат на розробку, нейронні мережі також можуть зменшити витрати на обслуговування VR-додатків. Нейронні мережі здатні безперервно навчатися та адаптуватися до змін у навколишньому середовищі, а це означає, що вони потребують менше обслуговування, ніж традиційні обчислювальні методи. Це може заощадити гроші розробників, зменшивши кількість часу і грошей, витрачених на обслуговування.

Нарешті, нейронні мережі можуть зменшити витрати за рахунок покращення користувацького досвіду. Нейронні мережі здатні виявляти і реагувати на зміни в навколишньому середовищі, а це означає, що вони можуть забезпечити більш захоплюючий досвід, ніж традиційні

обчислювальні методи. Це може призвести до збільшення залученості користувачів, що може допомогти збільшити продажі та зменшити витрати на маркетинг.

2.5. Проблеми згорткових нейронних мереж для віртуальної реальності

2.5.1. Обмежена доступність даних

Однією з головних проблем використання НМ для додатків віртуальної реальності є обмежена доступність даних. НМ потребують великих обсягів даних для навчання та отримання точних результатів. Однак у випадку з VR-додатками кількість доступних даних часто обмежена. Це може бути пов'язано з різними факторами, такими як обмежена кількість даних, доступних у певній галузі, або складність збору даних у середовищі віртуальної реальності. Як наслідок, продуктивність НМ може значно знижуватися при роботі з обмеженою кількістю даних.

Ця проблема ускладнюється ще й тим, що НМ потребують даних, які є репрезентативними для конкретного завдання або програми. Наприклад, якщо НМ використовується для ідентифікації об'єктів у середовищі віртуальної реальності, дані повинні бути репрезентативними для об'єктів у цьому середовищі. Це може бути складним завданням, оскільки дані можуть бути недоступними або недостатньо якісними. Крім того, може бути складно навчати НМ, коли є лише невелика кількість даних, оскільки НМ може бути не в змозі набути необхідні шаблони або ознаки.

Незважаючи на проблеми, пов'язані з обмеженою доступністю даних, існує ряд потенційних рішень, які можуть бути використані для вирішення цієї проблеми. Одне з найпростіших рішень - збирати більше даних. Збір більшої кількості даних може допомогти забезпечити репрезентативність даних для виконання поставленого завдання, а також надати НС більше інформації, на якій можна буде вчитися. Однак цей підхід не завжди може

бути здійсненим, оскільки збір великих обсягів даних може бути складним або дорогим.

Інший підхід полягає у використанні методів доповнення даних. Доповнення даних - це процес створення додаткових даних на основі наявних. Це може бути зроблено шляхом зміни існуючих даних або об'єднання даних з різних джерел. Такий підхід може допомогти збільшити кількість наявних даних, а також гарантувати, що дані будуть більш репрезентативними для виконання поставленого завдання.

Також використовувати методи трансферного навчання. Трансферне навчання передбачає використання попередньо навченої моделі на іншому завданні з метою покращення продуктивності поточного завдання. Цей підхід може бути використаний для використання наявних наборів даних і попередньо навчених моделей з метою підвищення ефективності виконання конкретного завдання. Цей підхід може бути особливо корисним, коли доступні дані обмежені, оскільки попередньо навчена модель може бути використана для вивчення необхідних особливостей і закономірностей.

2.5.2. Високі вимоги до обчислювальної потужності

Однією з основних проблем, пов'язаних з використанням ЗНМ для VR, є їхні високі вимоги до обчислювальної потужності. ЗНМ зазвичай вважаються обчислювально інтенсивними, що означає, що вони потребують великої кількості обчислювальної потужності для коректного функціонування. Це пов'язано з тим, що ЗНМ складаються з багатьох шарів нейронів, кожен з яких повинен бути обчислений для отримання бажаних результатів. В результаті розмір мережі може швидко зростати в геометричній прогресії, що призводить до потреби в більшій обчислювальній потужності для ефективної обробки даних.

Крім розміру мережі, складність даних, які обробляються, також є фактором, що визначає обсяг необхідної обчислювальної потужності. Наприклад, якщо вхідні дані дуже складні (наприклад, зображення або відео),

то для їх коректної обробки потрібна більша обчислювальна потужність. Аналогічно, якщо дані великі (наприклад, зображення або відео високої роздільної здатності), то для їх обробки мережі знадобиться більше обчислювальної потужності. Таким чином, розмір і складність даних, що обробляються мережею, є важливими факторами, які слід враховувати при визначенні необхідної обчислювальної потужності.

Через високі вимоги до обчислювальних потужностей ЗНМ часто неможливо запустити їх на стандартних комп'ютерах або ноутбуках. Для того, щоб отримати найкращі результати, часто необхідно використовувати потужні сервери або спеціалізоване обладнання, таке як GPU або TPU. Це може бути дорогим задоволенням, як з точки зору вартості обладнання, так і з точки зору поточних витрат на обслуговування, які пов'язані з роботою потужного сервера.

2.6. Шляхи застосування нейронних мереж для технології віртуальної реальності

У контексті віртуальної реальності нейронні мережі можна використовувати для забезпечення різних типів взаємодії. Програми віртуальної реальності можуть використовувати нейронні мережі для розпізнавання жестів і міміки або для обробки природної мови. Нейронні мережі також можуть бути використані для створення захоплюючих віртуальних середовищ шляхом інтерпретації зображень і створення візуальних ефектів у реальному часі. Тип нейронної мережі, що використовується для конкретного VR-додатку, залежить від бажаного результату. Наприклад, згортова нейронна мережа (ЗНМ) зазвичай використовується для розпізнавання зображень, тоді як рекурентна нейронна мережа (РНМ) використовується для обробки природної мови. Крім того, генеративні змагальні мережі (GAN) можуть бути використані для створення реалістичних зображень або інших візуальних ефектів.

Генерація зображень: НМ можна використовувати для створення реалістичних зображень у додатках віртуальної реальності. Існує безліч методів, які можна використовувати для створення зображень, від простих методів, таких як трасування променів, до більш просунутих методів, таких як глибоке навчання.

Нейронні мережі успішно застосовуються в різноманітних додатках віртуальної реальності. Наприклад, DeepMind від Google був використаний для створення середовища віртуальної реальності, яке можна досліджувати за допомогою GИ-агента. DeepMind - це тип нейронної мережі, відомий як мережа глибокого навчання, яка змогла успішно орієнтуватися у віртуальному середовищі та ідентифікувати об'єкти в ньому. Нейронні мережі також використовуються для створення віртуальних персонажів, які можуть взаємодіяти з користувачами. Наприклад, віртуальний асистент на основі НМ "Ніна" використовує комбінацію обробки природної мови та розпізнавання зображень для взаємодії з користувачами. Ніна працює на основі рекурентної нейронної мережі, яка здатна розуміти природну мову і реагувати на неї. Нейронні мережі також використовуються для створення реалістичних зображень у віртуальній реальності. Наприклад, НМ GameGAN від NVIDIA використовує генеративну змагальну мережу для створення реалістичних зображень віртуального середовища. НМ навчається на наборі даних реальних зображень і може генерувати нові зображення, які неможливо відрізнити від зображень реального світу.

Трасування променів - це техніка, яка використовується для створення реалістичних зображень у додатках віртуальної реальності. Вона передбачає відстеження шляху світлових променів під час їхньої взаємодії з об'єктами у віртуальному середовищі. Ця техніка використовується для створення реалістичного освітлення і тіней, а також віддзеркалень і заломлень. Результати зазвичай більш реалістичні, ніж ті, що створюються традиційними методами рендерингу.

Також можливе використання генеративних змагальних мереж (Generative Adversarial Networks, GAN) для створення реалістичних зображень в віртуальній реальності. GAN складаються з двох мереж, генератора і дискримінатора. Генератор приймає на вхід випадковий шум і створює зображення, тоді як дискримінатор приймає згенероване зображення і визначає, чи є воно реальним чи ні. Потім ці дві мережі змагаються між собою, з метою, щоб генератор створював реалістичні зображення, які обманюють дискримінатор.

Навчання в межах віртуальної реальності використовуючи нейронні мережі: інноваційний спосіб обробки інформації за допомогою нейронних мереж є важливим для ефективного використання доповненої реальності в задачах навчання операторів цільовим процесам. Для навчання працівників було запропоновано інноваційний підхід з використанням динамічної інформації, де інформація автоматично модифікується для окремих операторів і їхнього прогресу в навчанні для підвищення його ефективності. Для з'ясування змісту інформації використовуючи нейронні мережі була створена кваліфікована система, що базується на правилах і враховує людське пізнання, для з'ясування змісту інформації в доповненій реальності. Метод може бути застосований в трьох різних контекстах. Ці контексти передбачають використання однієї камери, яка підтримується двокамерною установкою за допомогою сфери на основі дзеркал, сфери погляду і об'єднання двокамерної установки з об'єктивом типу "риб'яче око". Вищезгадані схеми вимагають додаткового обладнання, яке потенційно може обмежити можливості застосування доповненої реальності. Послідовність освітлення за допомогою лінзи "риб'яче око" - це метод який збільшує потребу в спеціальному обладнанні. Іншим методом є оцінка освітленості за допомогою відеопотоку кольорових зображень і фрагментарної 3D-інформації, що генерується стереофонічною камерою. Недоліком такої тактики є вимога спеціального апаратного забезпечення.

Розпізнавання об'єктів: Розпізнавання об'єктів є одним з найважливіших застосувань нейронних мереж у віртуальній і доповненій реальності. Нейронні мережі використовуються для ідентифікації об'єктів у сцені та їх класифікації. Нейронну мережу можна навчити, надавши їй марковані приклади об'єктів з віртуального середовища. Після навчання нейронну мережу можна використовувати для розпізнавання об'єктів у віртуальному середовищі та їх точної класифікації. Розпізнавання об'єктів важливе для різних застосувань, таких як автономна навігація, інтерактивні ігри та керування роботами.

Сегментація зображень: Сегментація зображень - ще одне важливе застосування нейронних мереж у віртуальній і доповненій реальності. Сегментація зображення передбачає поділ зображення на різні області на основі їхнього вмісту. Нейронні мережі можна використовувати для точної сегментації зображення на складові частини, такі як об'єкти, фон і текстури. Це важливо для таких додатків, як 3D-мапінг, розпізнавання об'єктів і автономна навігація.

Radiosity: Це алгоритм глобального освітлення, який використовується для оцінки передачі світла в персонажах віртуального світу. Алгоритм використовує НМ, оскільки з'єднання не може бути завершене за допомогою лише радіальності. НМ навчається на світлових променях, і отримані нейронні сітки створюють реалістичне освітлення у VR (Sillion, Puech).

Розпізнавання мови: Нейронні мережі можна використовувати для розпізнавання мови у віртуальному середовищі, надавши їй марковані приклади мови користувачів. Це потрібно для різних застосувань, таких як обробка природної мови, інтерактивні ігри та віртуальні асистенти.

Візуалізація: Розробники систем віртуальної реальності вважають віртуальну реальність дуже адаптивним інструментом для візуалізації даних. Віртуальна реальність забезпечує занурення, а отже, справляє незабутній вплив на глядача. Користувач може ефективно взаємодіяти з даними і розуміти їхній прихований потенціал. Користувач також може динамічно

маніпулювати даними і переносити їх на інші платформи. Він також може бути використаний для 3D молекулярної візуалізації. Віртуальне середовище (CAVE) - одне з таких середовищ, яке допомагає у візуалізації даних. Зменшення розмірності є одним з питань, яке потребує уваги у візуалізації віртуальної реальності.

2.6.1. Генерування та масштабування високоякісних текстур для графічних двигунів віртуальної реальності

Масштабування текстури за допомогою нейронних мереж - це техніка, яка використовується для збільшення роздільної здатності зображення або текстури зі збереженням її оригінального вмісту. Цей процес також відомий як синтез текстури або надвисока роздільна здатність. Нейронні мережі - це тип штучного інтелекту, який здатен навчатися на основі даних і приймати рішення на основі цих даних. При масштабуванні текстур нейронна мережа використовується для створення нових текстур з високою роздільною здатністю на основі існуючих текстур з низькою роздільною здатністю. Цей метод можна використовувати для покращення візуальної якості зображень, текстур та інших цифрових медіа.

Масштабування текстур є складним завданням через складність текстур. Текстури можуть містити різноманітні візерунки, форми та кольори, які традиційним алгоритмам може бути важко точно відтворити у високій роздільній здатності. Нейронні мережі добре підходять для цього завдання, оскільки вони здатні вивчати складні патерни і приймати рішення на їх основі [13, 14].

При масштабуванні текстур за допомогою нейронних мереж першим кроком є створення набору даних текстур низької роздільної здатності та їхніх відповідних версій високої роздільної здатності. Потім цей набір даних використовується для навчання нейронної

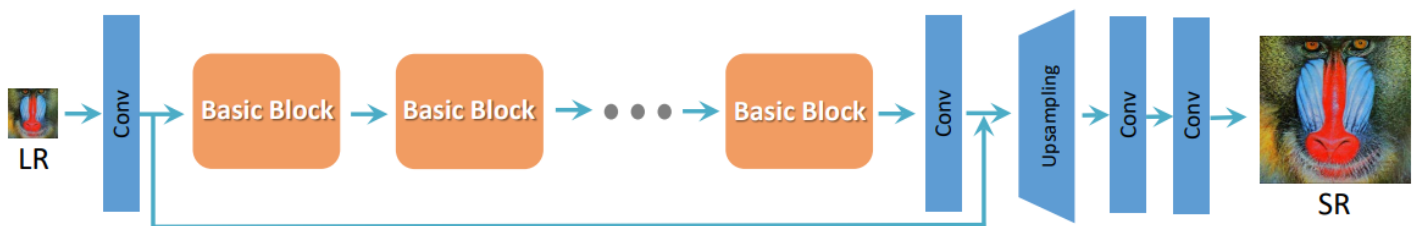


Рис. 2.4. Приклад використання гібридної нейронної мережі для збільшення роздільної здатності зображення

мережі. Під час навчання нейронна мережа отримує текстури з низькою роздільною здатністю і повинна навчитися генерувати їхні версії з високою роздільною здатністю. Після того, як нейронна мережа навчена, її можна використовувати для підвищення роздільної здатності будь-якої текстури низької роздільної здатності.

Процес навчання нейронної мережі для підвищення роздільної здатності текстури можна розбити на три етапи: виділення ознак, вивчення ознак і синтез текстури. Перший крок - це вилучення ознак, що передбачає вилучення ознак з текстур низької роздільної здатності, таких як краї, форми та кольори. Потім ці ознаки використовуються як вхідні дані для наступного кроку - навчання

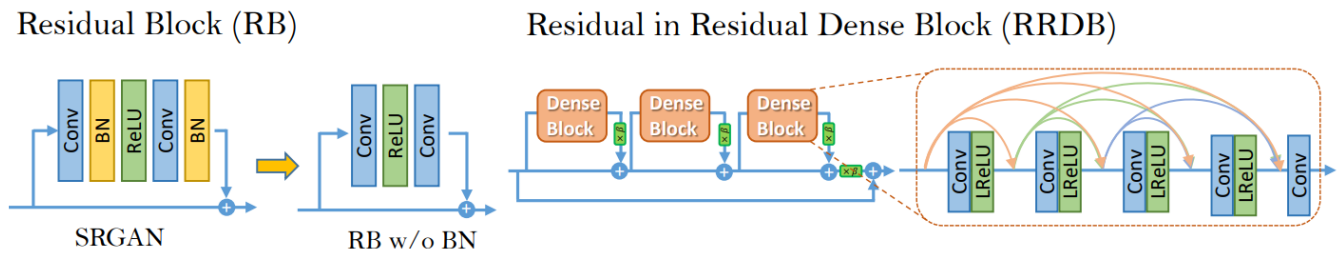


Рис. 2.5. Приклад архітектури RB-RR-DB нейронної мережі для збільшення роздільності зображень

ознак. На цьому кроці нейронна мережа вчиться комбінувати ці ознаки, щоб генерувати текстури з високою роздільною здатністю, подібні до вихідних текстур з низькою роздільною здатністю. Нарешті, на останньому кроці, синтезі текстури, нейромережа використовує вивчені особливості для створення нових текстур високої роздільної здатності з існуючих текстур низької роздільної здатності.

Масштабування текстури за допомогою нейронних мереж має кілька переваг над традиційними алгоритмами. Нейронні мережі здатні вивчати складні патерни, що дозволяє їм отримувати більш точні результати, ніж традиційні алгоритми. Крім того, нейронні мережі можна навчати на великих наборах даних, що дозволяє їм видавати більш якісні результати, ніж алгоритми, які покладаються на ручні функції або заздалегідь визначені правила. Нарешті, нейронні мережі здатні навчатися на основі даних у неконтрольований спосіб, що робить їх добре придатними для таких завдань, як масштабування текстур, де ручне маркування даних неможливе або недоцільне [15].

При виконанні збільшення роздільної здатності важливо використати правильну функцію для навчання нейронної мережі враховуючи вихідні втрати інформації з зображення після зміни його роздільної здатності на враховуючи генеративний елемент. Прикладом може бути наступна формула:

$$l^{SR} = l_X^{SR} + 10^{-3} l_{Gen}^{SR} \quad (2.1)$$

де l_X^{SR} це втрата інформації, а l_{Gen}^{SR} – адверсативне зменшення коефіцієнта.

Загальнопиксельна середньо-квадратична помилка розраховується за формулою:

$$l_{MSE}^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y})^2 \quad (2.2)$$

Це найпоширеніший підхід оптимізації для високороздільних зображень зображень. Однак, незважаючи на досягнення особливо високого PSNR коефіцієнта, рішенням задач оптимізації СКП [16] часто бракує високочастотного контенту, що призводить до незадовільних для сприйняття рішень з надмірно гладкими текстурами. Замість того, щоб покладатися на піксельні втрати, рекомендується використовувати функцію втрат, яка є ближчою до перцептивної схожості. Через $\phi_{i,j}$ позначається карта ознак, отримана за допомогою j -ї згортки (після активації) перед i -м шаром максимального об'єднання. Потім розраховуються втрати в якості евклідової відстані між представленнями ознаками відновленого зображення $G_{\theta_G}(I^{LR})$ та еталонного зображення I^{HR} :

$$l_{VGG/i,j}^{SR} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2 \quad (2.3)$$

Тут $W_{i,j}$ та $H_{i,j}$ описують розміри відповідних карт ознак у мережі.

На додаток до описаних вище втрат контенту, ми також додаємо до перцептивних втрат генеративний компонент нашого GAN. Це спонукає нашу мережу надавати перевагу рішенням, які знаходяться на множині природних зображень, намагаючись обдурити мережу дискримінатора. Генеративні втрати l_{Gen}^{SR} визначаються на основі ймовірностей дискримінатора $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ на всіх навчальних вибірках як:

$$l_{Gen}^{SR} = \sum_{n=1}^N -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \quad (2.4)$$

Тут $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ - це ймовірність того, що реконструйоване зображення $G_{\theta_G}(I^{LR})$ є природним HR-зображенням. Для кращої поведінки градієнта мінімізується $-\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$ замість $\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))$.

Щодо загальної системи збільшення роздільної здатності, то вона складається з двох систем: мережі перетворення зображень f_w та модуля калібровки ϕ який використовується для визначення декількох функцій втрат $l_1, \dots, l_k$. Мережа перетворення зображень - це нейронна мережа з глибоким залишковим згортком, параметризована вагами W ; вона перетворює вхідні зображення x у вихідні зображення $\hat{y}$ за допомогою відображення $\hat{y} = f_W(x)$. Кожна функція втрат обчислює скалярне значення $l_i(\hat{y}, y_i)$, що вимірює різницю між вихідним зображенням $\hat{y}$ та цільовим зображенням y_i . Мережа перетворення зображень навчається за допомогою стохастичного градієнтного спуску для мінімізації зваженої комбінації функцій втрат [17]:

$$W^* = \arg \min_W \mathbf{E}_{x, \{y_i\}} [\sum_{i=1} \lambda_i \ell_i(f_W(x), y_i)] \quad (2.5)$$

2.6.2. Використання нейронних мереж для розрахунку та моделювання поведінки освітлення в умовах віртуальної реальності

Освітлення є важливим елементом ігрового дизайну і використовується для створення реалістичного та захоплюючого середовища. Традиційні методи розрахунку поведінки світла, такі як трасування променів, є обчислювально дорогими і вимагають значних обчислювальних потужностей для відтворення реалістичних світлових ефектів. Це обмежувало складність і реалістичність світлових ефектів у грі.

Нейронні мережі можна використовувати для різноманітних розрахунків у фізиці світла, зокрема для обчислення відбивної здатності поверхонь, прогнозування виходу оптичних систем та аналізу зображень, отриманих за допомогою оптичних приладів.

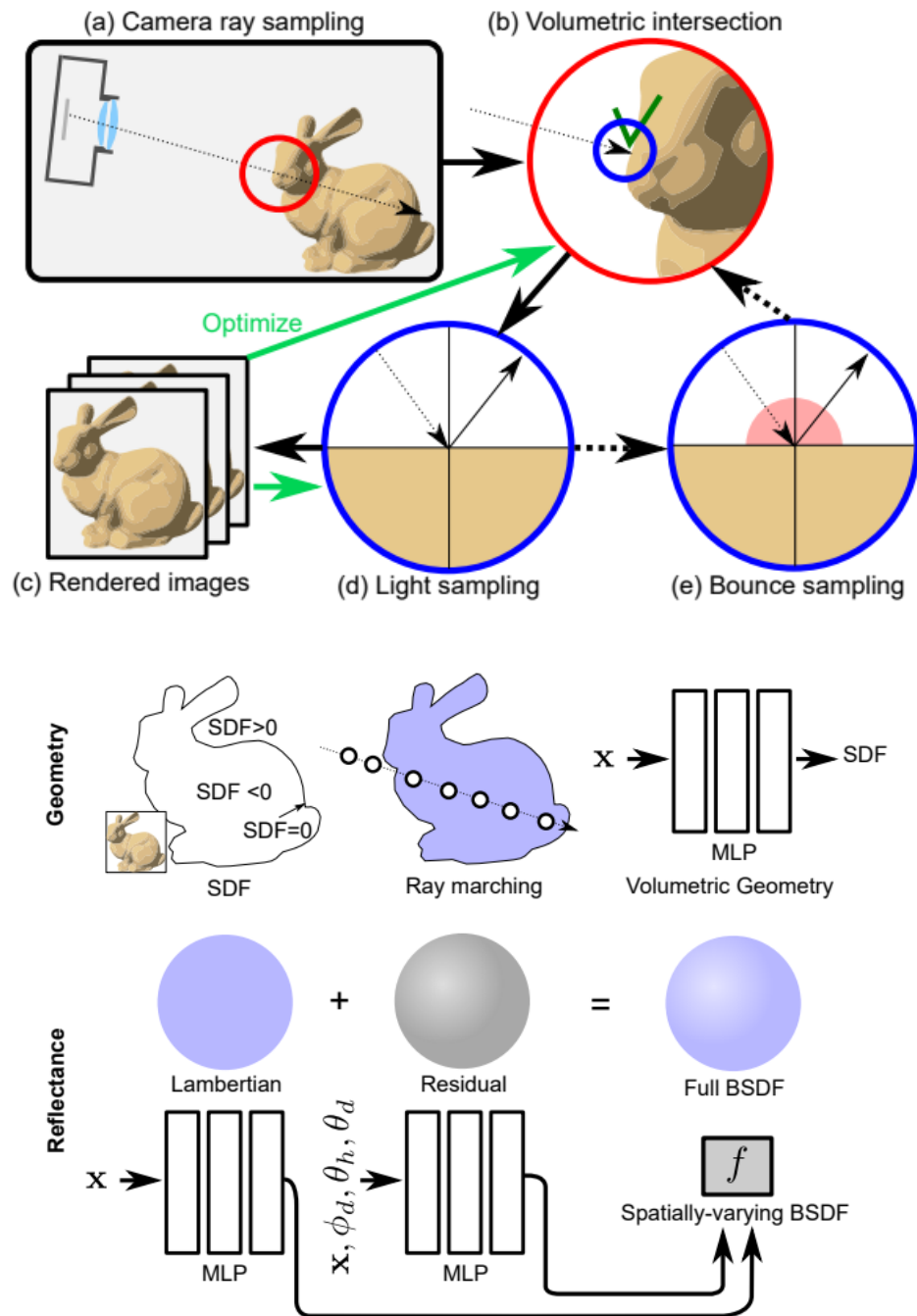


Рис. 2.6. Алгоритм роботи для трасування променів в графічних двигунах прямої дії з використанням нейронних мереж.

Обчислення відбивної здатності

Відбивна здатність поверхні є важливим параметром для розуміння того, як світло взаємодіє з нею. Він визначається як відношення між відбитою потужністю та потужністю, що падає на поверхню. Нейронні мережі можна використовувати для обчислення відбивної здатності поверхні, навчаючи їх на даних експериментів або симуляцій. Після навчання нейронна

мережа може точно прогнозувати відбивну здатність поверхні на основі її властивостей (наприклад, складу матеріалу, шорсткості поверхні).

Прогнозування вихідних даних оптичної системи

Оптичні системи використовуються для вимірювання різних фізичних властивостей (наприклад, температури, тиску). Нейронні мережі можна використовувати для прогнозування вихідних даних оптичної системи, навчаючи їх на даних експериментів або симуляцій. Після навчання нейронна мережа може точно передбачити вихід оптичної системи на основі її параметрів (наприклад, діапазону довжин хвиль, числової апертури).

Аналіз зображень, отриманих за допомогою оптичних інструментів

Зображення, отримані за допомогою оптичних інструментів, можна аналізувати за допомогою нейронних мереж для вилучення корисної інформації (наприклад, розподіл розмірів або характеристики форми). Наприклад, нейронні мережі можна навчати на зображеннях, отриманих за допомогою мікроскопів, щоб ідентифікувати різні типи клітин або частинок у зразку. Після навчання нейронна мережа може точно ідентифікувати різні типи клітин або частинок на зображенні, зробленому за допомогою оптичного приладу.

З метою розширення випромінювання до відбиття, нещодавні роботи також досліджували декомпозиційне моделювання відбиття та об'ємної густини, щоб уможливити редагування матеріалів та освітлення. Ці роботи часто покладаються на наближення до об'ємного рендерингу, такі як прогнозування глибинного закінчення. Це додатково робить рендеринг з декількома відскоками обчислювально дорогим. Чжан та ін. [33] запропонували метод нейронного рендерингу для декомпозиції запеченого NeRF-зображення на нормалі, альбедо та освітленість за невідомих умов освітлення. Всі попередні роботи покладаються на об'ємний рендеринг, демонструючи недостатню репрезентативність дискретних структур, що використовуються в конвеєрах прямого рендерингу в традиційній графіці.

Для математичного розрахунку рендерингу променів динамічного освітлення у сучасних системах використовується рівняння Кадзія що стало основою для фотореалістичного рендерингу, що моделює як пряме, так і непряме перенесення світла. Зокрема, рівняння рендерингу описує вихідне випромінювання L_o , що виходить з точки сцени x у напрямку ω_o , як

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} f(\mathbf{x}, \omega_i, \omega_o) L_i(\mathbf{x}, \omega_i) (\omega_i \cdot \mathbf{n}) d\omega_i \quad (2.5)$$

де L_e і L_i - випромінювання світла, що падає, ω_i - напрямок падіння, f - BRDF, $\mathbf{n}$ – поверхневі нормалі. Всі непрямі внески від інших точок сцени інтегруються по півсфері Ω . Розв'язання цього рівняння рендерингу зазвичай називають прямим рендерингом, і воно широко вивчається в комп'ютерній графіці.

Для відбивної здатності комбінується дифузний ламбертіанський BSDF з вивченим BSDF, щоб представити високоякісний просторово-змінний BSDF. Ми комбінуємо вивчений BSDF з повністю дифузною моделлю для того, щоб забезпечити правильні нормалі під час реконструкції.

$$f(\mathbf{x}, \omega_o, \omega_i) = f_{\text{Lambert}}(\mathbf{x}) + f_{\text{Learned}}(\mathbf{x}, \phi_d, \theta_h, \theta_d) \quad (2.6)$$

де $f_{\text{Lambert}}(\mathbf{x})$ - це Ламбертівська BSDF, визначена як σ ($\text{MLP}(\mathbf{x})$). σ - сигмоїдальна функція. Функція $f_{\text{Learned}}(\mathbf{x}, \phi_d, \theta_h, \theta_d)$ є навченим залишковим BSDF, визначеним як $\tanh(\text{MLP}(\mathbf{x}, \phi_d, \cos \theta_h, \cos \theta_d)) / (\omega_i \cdot \mathbf{n})$.

Зокрема, ми визначаємо функцію прямого рендерингу, кидаючи промені в бік пікселя камери x та інтегруючи вздовж променя за допомогою геометричного SDF. Для кожної точки x вздовж променя ми відбираємо прямі шляхи ω 0 до джерел світла з відомим положенням. Видимість точки x з позиції світла моделюється як MLP з сигмоїдною активацією як $O(\mathbf{x}, \omega_0) = \sigma(\text{MLP}(\mathbf{x}, \omega_0))$. Потім ми обчислюємо яскравість p , що виходить з точки x у напрямку до пікселя x , за формулою

$$p(\mathbf{x}) = \sum_{l \in L} (\mathbf{n} \cdot \vec{\mathbf{l}}_x) f(\mathbf{x}, \omega_o, \vec{\mathbf{x}}) O(\mathbf{x}, \vec{\mathbf{x}}) E_l \quad (2.7)$$

де E_l - яскравість для освітлення l . Тут L - множина джерел світла, а $\vec{\mathbf{x}}$ - одиничний вектор від джерела світла до точки. Маючи значення яскравості p ,

ми можемо обчислити значення пікселя $I(x)$ за допомогою об'ємного інтегрування

$$I(x) = \int T(t) \Psi(\text{SDF}(\mathbf{x}(t))) p(\mathbf{x}(t)) dt \quad (2.8)$$

де $\mathbf{x}(t)$ - вибіркова точка вздовж променя, випромінюваного пікселем камери x , як $\mathbf{x}(t) = \mathbf{o} + t\vec{0}\vec{x}$. Функція T - це функція прозорості, яка визначається як $T(t) = \exp\left(-\int_0^t \Psi(\text{SDF}(\mathbf{x}(t))) dt\right)$. Тут Ψ є об'ємною еквівалентністю SDF, тобто $\Psi(s) = \frac{1}{2} \exp\left(\frac{s}{\beta}\right)$ if $s \leq 0$, інакше $\Psi(s) = 1 - \frac{1}{2} \exp\left(-\frac{s}{\beta}\right)$. Тут ми припускаємо, що камера та освітлення відомі. Запропонований метод 4 спирається на цю функцію прямого рендерингу для навчання нейронних зображень сцени геометричного SDF, ламбертового BSDF та функції оклюзії шляхом розв'язання наступної оптимізаційної задачі

$$\min_{\text{SDF}, f, O} \|I(x; \text{SDF}, f, O) - I_{\text{ref}}(x)\|_2 + R(\text{SDF}), \quad (2.9)$$

де R - ейкональні регуляризації на поверхні, тобто $R(\text{SDF}) = \mathbb{E}_{\mathbf{v}_x} [\|\nabla \text{SDF}(\mathbf{x})\| - 1]^2$. Зауважте, що оптимізація, описана вище, виконується спільно для всіх ракурсів камери та джерел світла.

2.6.3. Алгоритмія пошуку шляху об'єктів віртуальної реальності в реальному часі використовуючи нейронні мережі.

Традиційно в комп'ютерних іграх пошук шляху здійснюється на статичному зменшеному зображенні віртуального світу, який представляє гра. Це добре працює, якщо віртуальний світ мало змінюється або взагалі не змінюється віртуальний світ не змінюється протягом гри. Так було у більшості ігор до цього часу, оскільки складність фізичного рушія гри в реальному часі була обмежена, головним чином, через час, необхідний для його розробки. Однак зараз ігри створюються з використанням проміжного програмного забезпечення для ключових компонентів гри, включаючи фізичний двигун. Проміжне програмне забезпечення - це програмне

забезпечення, написане зовнішнім джерелом, яке має хуки, що дозволяють йому бути інтегрованим в код розробника гри. Таким чином, розробники ігор можуть витрачати набагато більше часу на створення більш захоплюючих ігор з динамічними сценами в реальному часі. Це звучить захоплююче, однак традиційний штучний інтелект, який працює на основі статичного представлення ігрового середовища. віртуального середовища. Це обмежує кількість динамічних об'єктів, які можна додати в ігри, оскільки стратегія пошуку шляху повинна бути точно налаштована для роботи з ними, що збільшує час на розробку гри. розробці гри [18, 19].

Щоб AI міг ефективно орієнтуватися в динамічному світі, він повинен отримувати інформацію про навколишнє середовище в реальному часі. інформацію про навколишнє середовище в реальному часі. Щоб досягти цього традиційними методами, потрібно запуску алгоритму пошуку шляху на кожному кроці, що було б дорого в обчислювальному плані, особливо з огляду на обмежену пам'ять для обмеженої пам'яті, доступної для сучасних ігрових консолей. Тому AI-агент повинен бути наділити AI-агента якимись сенсорами, здатними отримувати інформацію про навколишнє середовище. Це не складно реалізувати реалізувати, однак ключова проблема полягає в тому, щоб змусити агента розшифровувати корисну інформацію з цих датчиків і реагувати відповідно до неї. сенсорів і реагувати відповідно в режимі реального часу, не надто навантажуючи комп'ютерні ресурси. ресурси комп'ютера. Штучні нейронні мережі є добре відомим методом штучного інтелекту, який забезпечує потенційне вирішення цієї проблеми.

Пошук шляху є важливою підпрограмою у багатьох важливих додатках.

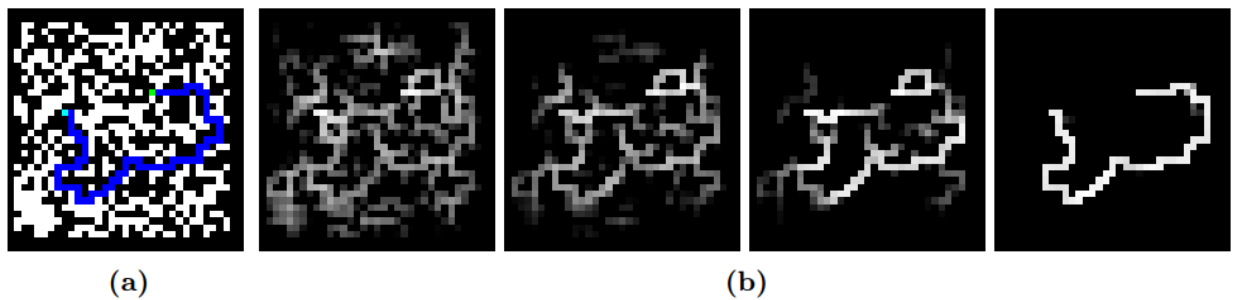


Рис. 2.7. Результат роботи нейронної мережі з пошуку оптимального шляху

На 2D-сітці задача знаходження найкоротшого шляху корисна для планування маршруту робота або транспортних маршрутів. Довгі шляхи (і такі величини, як діаметр) важливі для оцінки фотоелектричних властивостей процедурно згенерованих мікроструктур для сонячних панелей або складності рівнів відеоігор на основі сітки. Класичні алгоритми для розв'язання задач пошуку шляхів та суміжних задач включають алгоритм Беллмана-Форда для пошуку найкоротшого шляху від одного вихідного вузла до інших вузлів, Breadth-First Search (BFS), який моделює зв'язки між вузлами за допомогою карти Дейкстри, та Depth-First Search (DFS), який послідовно досліджує зв'язки від кожного вузла.

Нейронні мережі все частіше використовуються для розв'язання складних задач у вищезгаданих додатках, що включають підпрограми пошуку шляхів. Тому моделювання класичних алгоритмів пошуку шляху у "нейронно-правдоподібний" спосіб може бути корисним для цілісного вирішення цих складніших проблем. Цей підхід вже досліджувався раніше: для задачі пошуку найкоротшого шляху побудувати нейронні мережі, створені вручну, щоб реалізувати ефективну розподілену версію BFS [20].

Мета полягає в тому, щоб обчислити найкоротший шлях у заданому лабіринті, коли є визначене джерело u . Таким чином, ми спочатку генеруємо карту Дейкстри за допомогою виклику функції $Dijkstra(u)$, а потім витягуємо шлях з цієї карти Дейкстри за допомогою виклику функції "PathExtract" над цією картою Дейкстри. Таким чином, ми можемо позначити процедуру

пошуку найкоротшого шляху, що виконується нашими нейронними мережами з ручним кодуванням, як функцію "PathExtract(Dijkstra(u))". На практиці вагові коефіцієнти згортки мереж конкатенуються, а потім до результату послідовно застосовуються нелінійності та інші обчислення, специфічні для кожної процедури.

Нейронна карта, створена на основі карти оточення, містить сітку нейронів, де кожен нейрон характеризується рівнянням шунтування, яке визначає динаміку активності нейрона. У цій моделі кожен нейрон з'єднаний з сусідніми нейронами, і через ці зв'язки по мережі проходять як збуджуючі, так і гальмівні сигнали. Активність кожного нейрона, x_i , отримується шляхом розв'язання наступного диференціального рівняння шунтування:

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i) \underbrace{\left([I_i]^+ + \sum_{j=1}^N w_{ij} [x_j]^+ \right)}_{S_i^+} - (D + x_i)[I_i]^- \quad (2.10)$$

де w_{ij} - вага зв'язку між j -м та i -м нейронами, пропорційна відстані між ними до певного значення, при якому подальші нейрони виключаються, $[I_i]^+$ та $[I_i]^-$ позначають збуджуючий та гальмівний входи відповідно, і, нарешті, A , B та D - швидкість пасивного спаду, верхня межа та нижня межа нейронної активності відповідно. Сума між збудливим входом і позитивною активністю сусідніх нейронів, $[x_j]^+$, позначається S_i^+ . Для моделювання задачі планування шляху вхідні дані для даного нейрона (тобто його місцезнаходження) визначаються наступним чином:

$$I_i = \begin{cases} E, & \text{if there is a target} \\ -E, & \text{if there is an obstacle} \\ 0, & \text{otherwise} \end{cases} \quad (2.11)$$

Щоб вирішити проблему генерації неоптимальних шляхів у складних умовах, автори пропонують дві модифікації, які включають в себе операції згладжування та усереднення. Операція підстановки додає нейрони до межі карти таким чином, щоб значення активності кожного нейрона дорівнювало середньому значенню активності сусідніх нейронів всередині карти. По-

друге, вони модифікували моделювання збуджуючого входу S_i^+ наступним чином:

$$S_i^+ = [I_i]^+ + \frac{(\sum_{j=1}^N w_{ij}[x_j]^+)^{n-n_{\text{neg}}}}{n_{\text{pos}}^n} \quad (2.12)$$

де n_{neg} і n_{pos} позначають кількість сусідніх нейронів з негативною і позитивною активністю відповідно, а n - загальна кількість сусідніх нейронів, яка також включає n_{out} - кількість нейронів за межами карти, які додаються до межі карти, як описано вище.

2.6.4. Використання згорткових нейронних мереж для систем Unity3D. Інтерактивна віртуальна реальність в реальному часі.

При використанні нейронних мереж з двигуном Unity3D, цільова мережа може поєднувати просторову інформацію з високою роздільною здатністю та глибинні характеристики з низькою роздільною здатністю. Спочатку мережа модифікує швидкі з'єднання для вивчення модуля даунсемплінгу, включаючи шар Conv2D з 32 вихідними каналами, шар DWConv з 48 вихідними каналами і 64. За кожним шаром слідує шар пакетної нормалізації та шар активації за допомогою функції ReLU. Оскільки кожен шар згортки використовує нормальну згортку з кроком 2, просторова гілка зменшує вибірку вихідного зображення у 8 разів, а роздільна здатність вихідних ознак становить 1/8 від вихідного зображення. Крім того, завдяки високій роздільній здатності карти ознак, закодована просторова інформація є багатшою. 16-кратно та 32-кратно зменшені ознаки проходять через модуль оптимізації уваги (ARM) для отримання ознак, які інтегрують глобальну контекстну інформацію, що є кінцевим виходом контекстного шляху. У порівнянні з традиційною структурою мережі глибокого навчання, структура жорсткого розподілу параметрів повинна відповідати декільком завданням під час навчання мережі, і оскільки кожна підзадача має певну кореляцію між

собою, кожна підзадача буде просувати одна одну під час навчання мережі, що не тільки може допомогти мережі зменшити ризик перенасищення, але також може підвищити точність роботи мережі. Наприклад, фотографії, матеріали, середовище встановлення та інша інформація використовуються для мапування, освітлення, рендерингу та запікання моделі, щоб врешті-решт отримати модель віртуальної реальності з високореалістичними візуальними ефектами. М'який механізм обміну параметрами відрізняється від жорсткого механізму обміну параметрами тим, що кожна підмережа має магістральну мережу, яка еквівалентна декільком мережам, що працюють паралельно. Хоча м'який механізм спільного використання параметрів трохи кращий за жорсткий механізм спільного використання параметрів, м'який механізм спільного використання параметрів використовує більшу кількість параметрів і більший обсяг обчислень, а швидкість виявлення повільніша. Для досягнення ефекту виявлення в реальному часі в цій статті обрано жорсткий механізм обміну параметрами, який має велику перевагу в швидкості виявлення і може зберегти багато спільних параметрів, хоча він має просту структуру. У багатозадачній згортковій нейронній мережі шар ознак семантичної сегментації та шар ознак виявлення цілі спільно використовуються як шари ознак розгалуження багатозадачної згорткової нейронної мережі, так що координати цільового кадру, категорії та попередньо встановлені маски категорій можуть бути отримані одночасно при прогнозуванні мережі. Оскільки багатозадачна система глибокого навчання, що використовується в цій статті, має жорсткий розподіл параметрів, шар функцій семантичної сегментації та шар функцій виявлення цілей мають спільну магістральну мережу і використовують однакові параметри, що дозволяє зменшити багато параметрів і ефективно зменшити споживання ресурсів. Потім вони навчаються на наборі даних CIFAR-10, і записуються помилки навчання обох мереж в процесі навчання. Експерименти показують, що чим більшою є глибина мережі, тим меншою буде помилка навчання, а навпаки, після тривалого періоду навчання

помилка навчання буде більшою. Залишкова структура ділиться на дві частини: ліва половина - це залишкова частина, яка буде збільшувати або зменшувати згорткові шари відповідно до розміру мережі; права половина - це частина прямих зв'язків.

2.7. Архітектурні особливості систем віртуальної реальності а впровадження нейронних мереж в межах її структурних елементів

VR двигун складається з декількох компонентів, які працюють разом, щоб створити захоплюючий досвід віртуальної реальності. Ці компоненти включають двигун рендерингу, фізичний двигун, звуковий двигун, систему введення та мережеву систему. Двигун рендерингу відповідає за створення 3D-графіки, яка представляється користувачеві у віртуальному світі. Він отримує дані від фізичного рушія та інших компонентів і створює зображення, які відображаються на пристрої користувача. Фізичний двигун імітує фізичні властивості об'єктів у віртуальному світі, такі як гравітація, тертя та імпульс. Звуковий двигун генерує звукові ефекти та музику, які представляються користувачеві у віртуальному світі. Система вводу фіксує введення користувача з таких пристроїв, як ігрові контролери та датчики

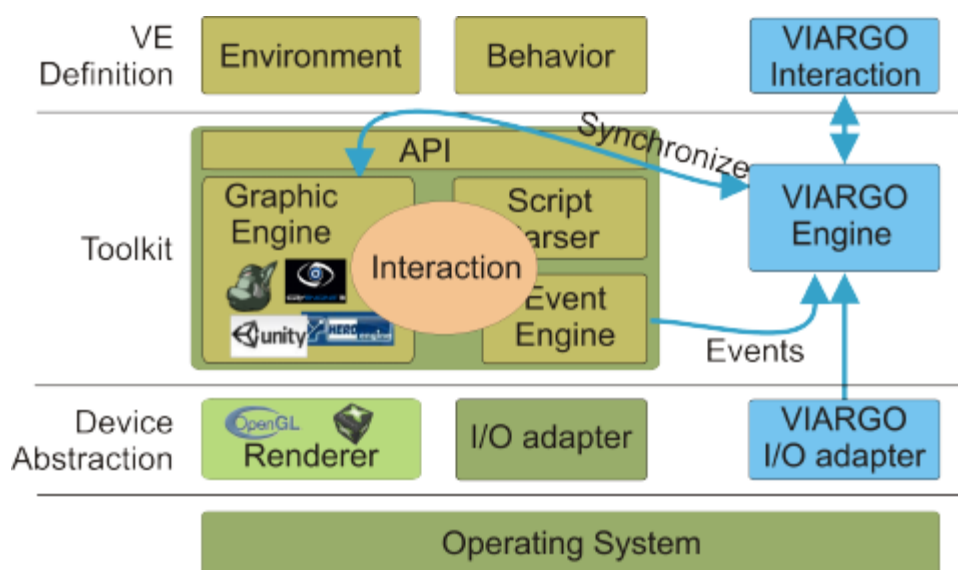


Рис. 2.8. Структурна схема системи віртуальної реальності з елементами внутрішнього графічного двигуна

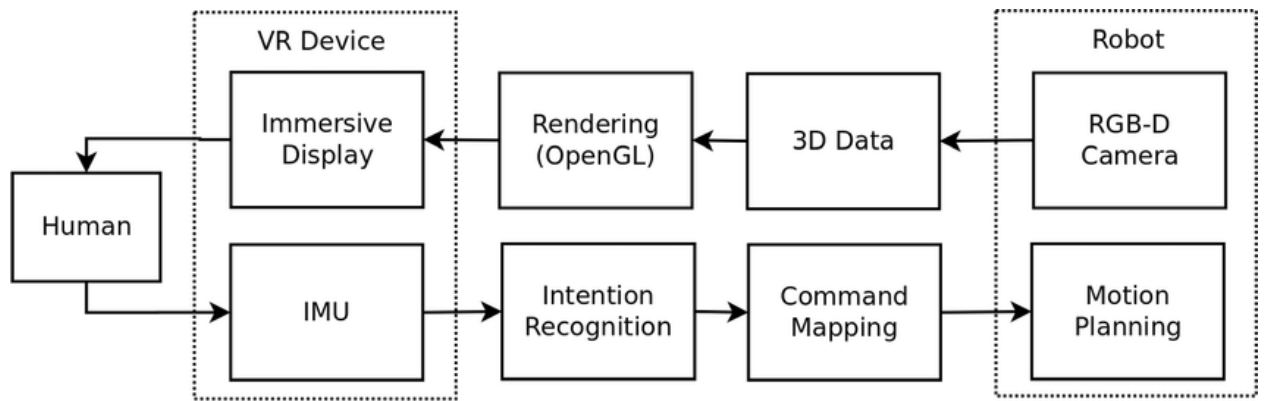


Рис. 2.9. Блок-схема інтеграції системи віртуальної реальності з нейронними мережами

руху. Мережева система дозволяє декільком користувачам взаємодіяти один з одним у спільному віртуальному середовищі.

Двигун рендерингу відповідає за генерацію 3D-графіки, яка відображається користувачеві у віртуальному світі. Він отримує дані від фізичного рушія та інших компонентів і створює зображення, які відображаються на пристрої користувача. Двигун рендерингу використовує різні методи, такі як трасування променів, растеризація та синтез зображень для створення високоякісних 3D-зображень. Трасування променів - це техніка, яка використовується для розрахунку взаємодії світла з об'єктами в сцені шляхом відстеження шляхів променів світла від камери через кожен піксель зображення. Растеризація - метод перетворення 3D-об'єктів у 2D-зображення шляхом проектування їх на 2D-поверхню. Синтез зображень - це метод, який використовується для об'єднання декількох 2D-зображень в одне 3D-зображення шляхом застосування різних ефектів, таких як тіні, відображення та освітлення.

Фізичний двигун

Фізичний двигун відповідає за імітацію фізичних властивостей, таких як гравітація, тертя та імпульс у віртуальному світі. Він отримує дані від рушія рендерингу та інших компонентів і обчислює, як об'єкти взаємодіють один з одним у віртуальному середовищі. Фізичний двигун використовує різні алгоритми, такі як виявлення зіткнень, динаміка твердого тіла та системи частинок, щоб імітувати реалістичну фізичну поведінку. Алгоритми

виявлення зіткнень визначають, коли два об'єкти зіштовхуються один з одним, і обчислюють їхній результуючий рух. Алгоритми динаміки твердого тіла обчислюють, як рухаються тверді тіла під дією різних сил, таких як сила тяжіння або тертя. Алгоритми систем частинок імітують взаємодію частинок між собою, наприклад, дим або вогонь.

Звуковий двигун

Звуковий двигун відповідає за генерацію звукових ефектів і музики, які представляються користувачеві у віртуальному світі. Він отримує дані від фізичного двигуна та інших компонентів і створює реалістичні звукові ефекти, які відтворюються через динаміки або навушники. Для створення реалістичних звуків звуковий двигун використовує різні методи, такі як синтез, семплювання, просторова візуалізація, реверберація та фільтрація. Методи синтезу генерують звуки з нуля за допомогою алгоритмів або математичних рівнянь. Методи семплювання записують звуки з реальних джерел, таких як інструменти або людські голоси. Методи просторової візуалізації розміщують звуки у тривимірному просторі так, щоб вони здавалися такими, що походять з різних напрямків відносно позиції користувача. Методи реверберації імітують відлуння звуку від стін чи інших поверхонь у кімнаті або навколишньому середовищі. Методи фільтрації модифікують звуки, застосовуючи такі ефекти, як відлуння або спотворення.

Система введення

Система вводу відповідає за збір даних, що вводяться користувачем з таких пристроїв, як ігрові контролери та датчики руху. Вона отримує дані з цих пристроїв і перетворює їх на команди або дії, які можуть бути використані іншими компонентами рушія віртуальної реальності, такими як фізичний двигун або звуковий двигун. Система введення використовує різні методи, такі як розпізнавання жестів, розпізнавання голосу, тактильний зворотний зв'язок, відстеження очей і розпізнавання обличчя, щоб точно інтерпретувати дані, введені користувачем. Алгоритми розпізнавання жестів виявляють рухи рук або жести користувача і переводять їх у команди або дії

у віртуальному світі. Алгоритми розпізнавання голосу розпізнають вимовлені користувачем слова або фрази і перетворюють їх на команди або дії у віртуальному світі. Системи тактильного зворотного зв'язку надають тактильний зворотний зв'язок користувачам за допомогою вібраційних двигунів або механізмів силового зворотного зв'язку, коли вони взаємодіють з об'єктами у віртуальному світі. Системи відстеження погляду визначають, куди користувачі дивляться в сцені або середовищі, щоб відповідно налаштувати те, що вони бачать на своєму пристрої відображення. Системи розпізнавання обличчя розпізнають вираз обличчя користувачів, щоб відповідно адаптувати вираз обличчя їх аватарів до сцени або оточення.

Мережева система

Мережева система відповідає за те, щоб дозволити декільком користувачам взаємодіяти один з одним у спільному віртуальному середовищі. Вона отримує дані з усіх підключених пристроїв користувачів, таких як ігрові контролери та датчики руху, і передає їх через мережу, щоб усі користувачі могли бачити дії один одного у віртуальному світі одночасно. Мережева система використовує різні протоколи, такі як TCP/IP або UDP для надійного зв'язку між підключеними пристроями користувачів через мережеве з'єднання.

Висновки до розділу 2

У розділі розглянуто застосування ЗНМ у технологіях віртуальної та доповненої реальності. Досліджено методи підвищення імерсивності та інтерактивності VR/AR-систем за допомогою глибокого навчання. Виявлено, що використання ЗНМ у VR/AR дозволяє покращити якість рендерингу, сприйняття глибини, трекінг об'єктів. Водночас встановлено, що впровадження таких систем стикається з проблемами високих обчислювальних витрат та необхідності ефективної оптимізації моделей. Запропоновано методи розв'язання цих проблем через адаптацію параметрів ЗНМ, зменшення їх розміру та впровадження апаратного прискорення

РОЗДІЛ 3

АЛГОРИТМ ДЛЯ СТРУКТУРНО-ПАРАМЕТРИЧНОГО СИНТЕЗУ ОПТИМАЛЬНОЇ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ЗАСТОСУВАННЯ В ВИРОБНИЧИХ ПРОЦЕСАХ З ВИКОРИСТАННЯМ БАГАТОКРИТЕРІАЛЬНИХ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

На сьогоднішній день, незважаючи на значний прогрес у галузі комп'ютерного зору та використанні передових згорткових мереж, особливо візуальних трансформерів [33], багато завдань обробки зображень залишаються невирішеними і потребують спеціалізованих рішень. Це здебільшого пов'язано з унікальними характеристиками навчальних наборів даних та високими обчислювальними вимогами складних топологій нейронних мереж, таких як трансформери, які стикаються з обмеженнями апаратного забезпечення. Більшість архітектур ЗНМ страждають від низької продуктивності, повільних темпів навчання і водночас вимагають високоякісних, добре збалансованих навчальних наборів даних.

Природним виходом з цієї ситуації є створення гібридних згорткових нейронних мереж [34, 35]. Однак у цьому процесі виникає багато проблем: оптимальний вибір базової топології згорткової нейронної мережі, оптимальний вибір різних структурних блоків, які будуть використовуватися в процесі синтезу структури гібридної згорткової нейронної мережі, оптимальний вибір їх розташування. З точки зору машинного навчання, виникають проблеми з вибором критеріїв навчання, коли ми маємо однокритеріальну або багатокритеріальну задачу оптимізації, потрапляння в локальний екстремум, перенавчання, падіння градієнта, попереднього навчання, створення гібридного алгоритму навчання.

Ця робота окреслить і обговорить результати досліджень щодо застосування та архітектурних характеристик гібридних згорткових нейронних мереж (НССН) та різних будівельних блоків, необхідних для їх синтезу. Це включає вивчення продуктивності окремих компонентів, аналіз сучасних архітектур ЗНМ та застосування різних навчальних даних для

вирішення проблем продуктивності. Основна увага в цьому дослідженні приділяється визначенню найкращих типів і розташувань сучасних ЗНМ, виділенню операційних компонентів і включенню їх у синтез ГЗНМ за допомогою еволюційних алгоритмів для досягнення сприятливих результатів продуктивності, точності та інших критеріїв оптимізації. Головною метою цієї роботи є розробка еволюційного механізму, який використовуватиме структурні компоненти різних архітектур ЗНМ для створення моделі, що задовольнятиме попередньо визначені критерії оптимізації.

3.1. Визначення критеріїв оптимізації для налаштування генетичного алгоритму

В цій роботі задано навчальну вибірку, яка складається з розмічених зображень. Поставлено задачу класифікації, яка включає в себе два етапи: виділення ознак та їх класифікацію. Для вирішення цієї задачі необхідно сформувати гібридну топологію, що передбачає вибір компонентів гібридної згорткової мережі, визначення їхнього розташування, а також налаштування структурних параметрів, таких як кількість шарів, кількість каналів, послідовність шарів, розміри ядер згортки, та значення вагових коефіцієнтів..

Щоб вирішити цю проблему, багатокритеріальна оптимізація [36] надає основу для систематичної оцінки мережі та оптимізації на основі сформульованих критеріїв оптимізації.

Метою цього дослідження є розробка та налаштування еволюційного алгоритму для багатокритеріальної оптимізації ЗНМ за допомогою еволюційних алгоритмів.

Основні завдання включають: Визначення критеріїв оптимізації: Встановлення набору релевантних показників продуктивності, які відображають якість і ефективність архітектур ГЗНМ. Ці критерії зазвичай включають [37]:

- Точність: Здатність ГЗНМ правильно класифікувати або прогнозувати дані.

- Обчислювальна ефективність: Показники, такі як час інференції та використання пам'яті.
- Надійність: Стійкість моделі до змін вхідних даних та атак на основі обману.
- Масштабованість: Здатність моделі підтримувати продуктивність при масштабуванні на більші набори даних або більш складні завдання.

Простір структурного дизайну ЗНМ: Ідентифікація та параметризація компонентів і конфігурацій ГЗНМ, включаючи:

- Тип і глибину шарів ЗНМ: Кількість згорткових, пулінгових і повнозв'язаних шарів.
- Тип і метод інтеграції гібридних компонентів: Вибір між структурними блоками та компонентами, що використовуються для структурного синтезу, типи компонентів інтеграції (наприклад, SRU, CRU, LSTM, GRU, GCN, GAT тощо).
- Схеми з'єднання: Як різні компоненти з'єднані між собою і як дані передаються між ними.
- Гіперпараметри: Такі параметри, як швидкість навчання, розмір пакету та коефіцієнти відсіву.

Розробка еволюційного алгоритму: Розробка та впровадження еволюційного алгоритму, адаптованого для структурного синтезу ЗНМ. Це включає:

- Кодування ЗНМ: Розробка схеми кодування для представлення архітектури ГЗНМ у форматі, придатному для еволюційних операцій.
- Функція пристосованості: Формулювання багатокритеріальної функції пристосованості, яка балансує точність, ефективність, надійність і масштабованість.
- Механізм відбору: Впровадження процесу відбору, який зберігає різноманітність і спрямовує пошук у бік оптимальних рішень.

- Оператори кросовера та мутації: Створення операторів для генерації нових архітектур ГЗНМ шляхом рекомбінації та модифікації існуючих на основі попередньо відібраних структурних блоків.

Завдяки виконанню цих завдань, це дослідження має на меті значно просунути стан науки в галузі дизайну та оптимізації гібридних нейронних мереж, надаючи потужний інструмент для розробки більш ефективних, надійних і масштабованих моделей глибокого навчання.

3.2. Запропонований багатокритеріальний генетичний алгоритм оптимізований для ЗНМ

Для структурного синтезу цільових ЗНМ ми плануємо використовувати модифікований алгоритм SPEA-3[8], щоб подолати зазначені проблеми.

У контексті цієї роботи індивід представляє конкретну архітектуру ЗНМ. Відповідно до критеріїв, визначених у розділі постановки задачі, цільовий індивід розглядається як закодований безпосередньо у рядок, який явно описує архітектуру. Структура геному включає в себе наступні параметри [38]:

- кількість шарів;
- типи шарів/блоків (SCConv, SE-BE-Inc, Dense block, стандартні згорткові, пулінгові, 1x1, нормалізація пакетів тощо);
- розміри ядер;
- кількість фільтрів;
- крок і заповнення;
- функції активації;
- специфічні параметри, пов'язані з блоками;
- швидкість навчання, розмір пакета тощо.

У роботі використовуватимемо змішену схему кодування, де кожен індивід (геном) складається з серії структурних блоків і гіперпараметрів.

Кожен ген у геномі представляє або шар, або блок із закодованими всередині нього специфічними параметрами.

Якісні критерії, за якими обирається фінальний індивід, визначені та докладно описані в наступному розділі.

Для кількісної оцінки продуктивності гібридних згорткових нейронних мереж (ГЗНМ) на основі кількох цілей необхідно побудувати багатокритеріальну функцію пристосованості, яка інтегрує різні показники продуктивності. Зазвичай ці показники включають точність, обчислювальну ефективність, надійність та масштабованість. Запропонована функція пристосованості врівноважує ці аспекти для забезпечення всебічної оцінки архітектур ГЗНМ.

Критерії, що враховуються при формулюванні функцій пристосованості, такі:

- Точність (A): Основний показник того, наскільки добре мережа виконує завдання, наприклад, точність класифікації на валідаційному наборі даних.
- Обчислювальна ефективність (E): Показники, такі як час інференції та використання пам'яті, вказують на ефективність мережі.
- Надійність (R): Стійкість мережі до атак на основі обману або зашумлених даних, часто вимірюється точністю в умовах атак або деградацією продуктивності.
- Масштабованість (S): Здатність підтримувати продуктивність при масштабуванні на більші набори даних або більш складні завдання, часто оцінюється зміною точності та ефективності під час масштабування мережі.

Для формулювання фітнес функції необхідно виконати кілька кроків.

Оптимізація виконується з урахуванням конфігурації ЗНМ (x), яка може включати кількість шарів, фільтрів, розмір ядра згортки, архітектурні гіперпараметри тощо.

Формулюємо задачі оптимізації:

$$\max_{\mathbf{x} \in \Omega} [f_1(\mathbf{x}), -f_2(\mathbf{x}), f_3(\mathbf{x}), f_4(\mathbf{x})], \quad (3.1)$$

де:

$f_1(\mathbf{x})$ – точність моделі на тестовій вибірці;

$f_2(\mathbf{x})$ – час інференції або використання пам'яті (мінімізується);

$f_3(\mathbf{x})$ – стійкість, оцінюється через тестування на пертурбованих даних або при атаках;

$f_4(\mathbf{x})$ – масштабованість, оцінюється через продуктивність на більших наборах даних.

Для зведення критеріїв до однакових масштабів виконуємо нормалізацію:

$$z_i = \frac{f_i(\mathbf{x}) - z_i^{\min}}{z_i^{\max} - z_i^{\min}}, i = 1, 2, 3, 4, \quad (3.2)$$

де $z_i^{\max}$; $z_i^{\min}$ — найгірше та найкраще значення для кожного критерію в поточній популяції.

Далі потрібно визначити референсні вектори. Референсні вектори $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_K)$ є напрямками в просторі M -вимірних цільових функцій і задаються за допомогою рівномірного розподілу і визначаються для покриття всього простору цілей. Їх кількість залежить від розмірності простору та бажаної густоти розподілу рішень. Вектори обираються так, щоб вони відповідали рівномірному розподілу бажаних компромісів між критеріями.

Число векторів обчислюється за формулою:

$$K = \binom{N + M - 1}{M - 1}, \quad (3.3)$$

де:

N - параметр рівномірності (визначає густоту векторів),

$M = 4$: кількість критеріїв.

Наприклад, при $N = 10$ і $M = 4$, кількість векторів буде:

$$K = \binom{10 + 4 - 1}{4 - 1} = \binom{13}{3} = 286.$$

Далі для кожного рішення z_i в нормалізованому просторі цілей обчислюємо скаляризоване значення. Саме перетворення багатокритеріальної задачі на низку скалярних підзадач досягається за допомогою референсних векторів і скаляризації, що враховує як збіжність до фронту Парето, так і рівномірний розподіл рішень у просторі цільових функцій.

Перед обчисленням фітнесу кожне рішення $\mathbf{z}$ (вектор значень цільових функцій, нормалізованих для усунення масштабних розбіжностей) асоціюється з найближчим референсним вектором $\mathbf{v}_j$. Це забезпечує зв'язок між рішенням і регіоном у просторі цілей, який цей вектор представляє. Вибір найближчого вектора виконується за допомогою проєкції $\mathbf{z}$ на напрямок $\mathbf{v}_j$, що мінімізує кут між ними. Формула для знаходження асоційованого вектора:

$$j^* = \arg \max_j \cos \theta_j, \cos \theta_j = \frac{\mathbf{z} \cdot \mathbf{v}_j}{\|\mathbf{z}\| \|\mathbf{v}_j\|}, \quad (3.4)$$

де $\cos \theta_j$ — косинус кута між вектором рішення $\mathbf{z}$ і референсним вектором $\mathbf{v}_j$.

Для оцінки придатності кожного рішення щодо його прив'язки до вектора $\mathbf{v}_j$ використовується скаляризована фітнес-функція. Вона має два компоненти:

Збіжність до фронту Парето: Перевіряється через проєкцію рішення на напрямок вектора $\mathbf{v}_j$, яка визначає, наскільки близьке рішення до ідеальної точки для цього вектора.

Різноманітність рішень: Оцінюється шляхом врахування відстані між рішенням $\mathbf{z}$ і його проєкцією на $\mathbf{v}_j$, що сприяє рівномірному розподілу рішень.

Формула скаляризованої фітнес-функції виглядає так:

$$S(\mathbf{z}, \mathbf{v}_j) = \mathbf{v}_j^T \cdot \mathbf{z} + \alpha \cdot \|\mathbf{z}\| \cdot \|\mathbf{v}_j\|, \quad (3.5)$$

де $\mathbf{v}_j^T$ - проєкція рішення на референсний вектор, яка відображає збіжність до фронту Парето, $\|\mathbf{z}\| \cdot \|\mathbf{v}_j\|$, — довжини векторів $\mathbf{z}$ і $\mathbf{v}_j$, що допомагають оцінити різницю між ними, α - адаптивний коефіцієнт, що збільшується з часом, змінюючи акцент між збіжністю і різноманітністю:

$$\alpha(t) = \frac{t}{T}, \quad (3.6)$$

де t — номер поточного покоління, T — загальна кількість поколінь.

Тобто, На ранніх етапах ($\alpha \approx 0$) основний акцент робиться на збіжності до фронту Парето. На пізніх етапах ($\alpha \approx 1$) підсилюється різноманітність розподілу рішень уздовж фронту.

Після обчислення значення $S(\mathbf{z}, \mathbf{v}_j)$ для кожного рішення, яке асоційоване з певним вектором, ці значення використовуються для порівняння рішень. Рішення з найменшим значенням S вважається найкращим для даного референсного вектора.

Тобто, спочатку всі рішення в поточній популяції нормалізуються, щоб усунути масштабні розбіжності між різними цільовими функціями. Це дозволяє об'єктивно порівнювати рішення незалежно від величини їхніх цільових значень. Кожне рішення асоціюється з найближчим референсним вектором. Для цього використовується проєкція рішення на вектор у просторі нормалізованих цілей. Найближчий вектор визначається через мінімізацію кута між нормалізованими цільовими функціями рішення та напрямком вектора. Ця асоціація дозволяє групувати рішення відповідно до регіонів у просторі цільових функцій, які ці вектори представляють.

Для кожного референсного вектора визначається набір рішень, які асоціюються з ним. Якщо кількість таких рішень перевищує ліміт (наприклад, одне рішення на вектор), виконується ранжування рішень у межах цієї групи. Рішення ранжуються за скаляризованою фітнес-функцією, яка враховує відстань від рішення до вектора та його збіжність до фронту Парето. Рішення з найкращим значенням скаляризованої функції обирається для потрапляння до наступного покоління. Це дозволяє гарантувати, що

кожен референсний вектор представлений одним найбільш релевантним рішенням.

У разі, якщо для деякого референсного вектора відсутні асоційовані рішення, він може бути переналаштований, або його напрямок модифікується, щоб краще відповідати реальному розподілу рішень. Таким чином, процес відбору забезпечує не лише підтримання збіжності до фронту Парето, але й рівномірний розподіл рішень вздовж нього. Це особливо важливо для багатокритеріальних задач з великою кількістю критеріїв.

Далі важливо розглянути загальний алгоритм з використанням сформульованої фітнес функції. Алгоритмічна модель виглядає наступним чином:

Вихід: оптимальна популяція

Вхід: {фітнес функція, demand_var, crossover_ratio}, набір попередньо визначених будівельних блоків, кількість поколінь, набір зображень для класифікації.

begin

(1) Згенерувати випадкову початкову популяцію (I_P);

(2) Порахувати фітнес для початкової популяції;

(3) Сортувати початкову популяцію;

(4) /* Вибірка */

(5) set $F_P = I_P$; /* F_P – кінцева популяція */

(6) **while** $c_e \neq 0$ or $l_G == \text{true}$ **do**

(7) /* c_e і l_G представляють елімінацію дітей та останнє покоління */

(8) згенерувати довільне c ; /* c позначає children */

(9) set $c_e = 0$;

(10) **for each** c **do**

(11) порахувати фітнес для c ;

(12) **if** $f_c \leq f_{F_P[1]}$ **then**

(13) remove c ;

(14) set $c_e + 1$;

```

(15)      else
(16)          set  $F_p = c$ ;
(17)      end
(18)  end
(19)  /* Mutation */
(20)  for crossover do
(21)      select c1 and c2 randomly;
(22)       $c3 = c1 \oplus c2$ ;
(23)      порахувати фітнес для c3;
(24)      if  $f_{c_3} \leq f_{c_1}$  or  $f_{c_2}$  then
(25)          remove c3;
(26)      else
(27)          remove c1, c2;
(28)      end
(29)  end
(30)  next generation
(31) end
(32) /* Ranking */ sort the FP;
(33) return  $F_p[1]$  /* повертає найбільш домінантний результат */
end

```

Задавши набір заздалегідь визначених структурних блоків ЗНМ, розмір популяції, а також максимальне число поколінь для ГА і набір даних для класифікації зображень, запропонований алгоритм починає працювати, проходячи ряд еволюційних процесів, і, нарешті, знаходить найкращу архітектуру ЗНМ для класифікації заданого набору даних зображень. Під час еволюції популяція випадковим чином ініціалізується заздалегідь визначеним розміром, використовуючи запропоновану стратегію кодування для компонування заздалегідь визначених структурних блоків. Потім лічильник поточного покоління ініціалізується нулем. Під час еволюції оцінюється пристосованість кожної особини, яка кодує певну архітектуру

ЗНМ, на заданому наборі даних. Після цього вибираються батьківські особини на основі фітнесу, а потім генеруються нові нащадки за допомогою генетичних операторів, включаючи оператори кросинговеру і мутації. Потім з поточної популяції за допомогою екологічного відбору відбирається популяція особин, що виживають у наступному поколінні. Зокрема, поточна популяція складається з батьківської популяції та популяції нащадків. Нарешті, лічильник збільшується на одиницю, і еволюція продовжується до тих пір, поки лічильник не перевищить задану максимальну кількість поколінь.

Ініціалізація популяції

Як було показано у Розділі 1, ЗНМ складається з згорткових шарів, об'єднаних шарів та іноді повністю з'єднаних шарів. Продуктивність ЗНМ значною мірою залежить від її глибини, і перепускні зв'язки можуть перетворити велику глибину на реальність. У запропонованій стратегії кодування ми розробляємо новий будівельний блок, безпосередньо використовуючи з'єднання пропусків, названий шаром пропуску, для заміни шару згортки при формуванні ЗНМ. Крім того, повністю з'єднані шари відкидаються у запропонованій стратегії кодування (причина буде вказано пізніше у цьому підрозділі). Таким чином, лише пропускні шари та об'єднані шари використовуються для побудови ЗНМ у запропонованій стратегії кодування.

Зокрема, пропускний шар складається з двох згорткових шарів і одного з'єднання пропуску. Пропускне з'єднання з'єднує від входу першого згорткового шару до виходу другого згорткового шару. Як було зазначено вище, параметрами згорткового шару є кількість карт ознак карт, розмір фільтра, розмір кроку і тип згортки. тип операції згортки. У запропонованій стратегії кодування ми використовуємо ті самі налаштування для розмірів фільтрів, кроків та операцій згортки, відповідно. операції згортки,

відповідно. Зокрема, розміри фільтра та кроку встановлено на 3×3 та 1×1 відповідно, і використовується лише однакова операція згортки використовується однакова. З цією метою параметри для пропускового шару - це номери карт ознак для двох шарів згортки (позначені як F_1 і F_2 відповідно). Крім того, шари об'єднання, що використовуються у запропонованій стратегії кодування, мають розміри 2×2 для ядра розміри ядра та розміри кроку. З цією метою, параметр, що кодується для шару об'єднання є лише тип об'єднання (позначений як P1). Зауважте, що причини такої схеми та вибору параметрів для такого дизайну буде пояснено далі у цьому підрозділі.

Ця ЗНМ складається з чотирьох пропускових шарів і двох об'єднуючих шарів. Код, що представляє один пропусковий шар, є рядком номерів карт ознак відповідних згорткових шарів у межах одного пропускового шару; тоді як код для об'єднувального шару є числом, що представляє тип об'єднання. Зокрема, випадкове число між (0, 0.5) означає максимальний рівень об'єднання, а між [0.5, 1) - середній рівень об'єднання. Код, що представляє всю ЗНМ, є послідовним рядковим з'єднанням кодів, що представляють шари. Як показано в цьому прикладі, де код кожного шару перераховано вище, код всієї ЗНМ має вигляд "32-64-0.2-64-256-0.8-512-256-256-512" для представлення ЗНМ з глибиною 10.

Налаштування розмірів фільтра та ядра, а також кроку в шарах об'єднання базуються на дизайні існуючих ЗНМ, створених вручну [9]. Крім того, ще однією важливою причиною вказівки таких параметрів є наш досвід ручного налаштування архітектури ЗНМ.

Запропонований алгоритм розрахунку значень загального та індивідуального фітнесу виглядає наступним чином:

begin

- (1) Згенерувати випадкову початкову популяцію (IP);
- (2) Обчислити значення фітнесу для кожного індивіда у початковій популяції;

- (3) Нормалізувати фітнес-значення на основі обраних метрик;
- (4) Визначити вагові вектори для оцінки популяції;
- (5) /* Ітераційна оптимізація */
- (6) **while** не досягнуто критерій зупинки **do**
- (7) Оновити параметри адаптації на основі поточного покоління;
- (8) Виконати селекцію батьків на основі еволюційного вибору;
- (9) Застосувати оператори кросоверу та мутації для створення нового покоління;
- (10) Обчислити фітнес-значення для нового покоління;
- (11) Нормалізувати значення фітнесу з урахуванням розподілу популяції;
- (12) Виконати ранжування індивідів відповідно до їх вагових коефіцієнтів;
- (13) Відсіяти найгірших індивідів відповідно до адаптивної стратегії;
- (14) **if** поточне покоління є останнім **then**
- (15) Зберегти найкращих представників популяції;
- (16) **end**
- (17) Оновити множину вагових векторів для наступного етапу еволюції;
- (18) **end**
- (19) /* Фінальне ранжування */
- (20) Визначити найкращого індивіда як оптимальне рішення;
- (21) **return FP[1]** /* повертає найбільш оптимальну модель */
- end**

Схема детально описує алгоритм оцінки придатності особин у популяції. Якщо коротко, то, маючи популяцію, що містить всі особини для оцінки придатності, і набір даних класифікації зображень, на основі якого має бути виявлена найкраща архітектура ЗНМ, алгоритм оцінює кожну особу у такий самий спосіб, і, нарешті, повертає P_t , що містить особини, придатність яких була оцінена. Зокрема, якщо оцінка пристосованості

проводиться для ініціалізованої популяції, тобто P_0 , створюється глобальна кеш-система (позначена як *Cache*), яка зберігає пристосованість особин з невидимими архітектурами (рядки 1-4). Для кожної особини (позначимо її індивідуумом) в P_t , якщо вона знайдена в *Cache*, її пристосованість безпосередньо отримується з *Cache* (рядки 6-8). В іншому випадку, особина асинхронно розміщується на доступному графічному процесорі для оцінки її придатності (рядки 9-13). Зауважте, що запит особини з кешу базується на її ідентифікаторі. Теоретично можна використовувати довільні ідентифікатори, якщо вони дозволяють розрізняти особини, що кодуються різними архітектурами. У запропонованому алгоритмі в якості відповідного ідентифікатора використовується 224-хеш-код, який реалізований більшістю мов програмування, з точки зору архітектури, що кодується. Крім того, індивідуум асинхронно розміщується на доступному графічному процесорі, що означає, що нам не потрібно чекати на оцінку придатності наступного індивідуума, поки не закінчиться оцінка придатності поточного, а розмістити наступного індивідуума на доступному графічному процесорі негайно.

Деталі оцінки придатності однієї особини показано в алгоритмі розрахунку значень загального та індивідуального фітнесу. Спочатку декодується ЗНМ від особини, і до цієї ЗНМ додається класифікатор (рядок 1) на основі заданого набору даних класифікації зображень. У запропонованому алгоритмі використовується класифікатор *softmax*, а конкретна кількість класів визначається заданим набором даних зображень. При декодуванні ЗНМ до виходу згорткового шару додається функція активації випрямляча з наступною операцією пакетної нормалізації, яка базується на конвенціях сучасних ЗНМ [8], [9]. Крім того, коли просторовий номер шару пропуску відрізняється від просторового номера вхідних даних, до вхідних даних додається згортковий шар, який має одиничний фільтр та одиничний крок, але спеціальну кількість карт ознак [8], [9]. Після цього ЗНМ навчається за алгоритмом *Stochastic Gradient Descent (SGD)* [39] на навчальних даних з використанням заданого графічного процесора (рядок 4),

а точність класифікації обчислюється за даними оцінки придатності (рядок 5). Зауважимо, що використання класифікатора softmax та методу навчання SGD ґрунтується на загальноприйнятих у спільноті фахівців з глибокого навчання. Після завершення етапу навчання найкраща точність класифікації на основі даних оцінки придатності встановлюється як придатність особи (рядок 10). Нарешті, ідентифікатор і придатність особи асоціюються і поміщаються в кеш (рядок 11).

Далі наводяться причини розробки асинхронного та кеш-компонентів. Підсумовуючи, оскільки навчання на ЗНМ є дуже трудомістким, варіюючись від декількох годин до декількох місяців в залежності від конкретної архітектури, вони призначені для прискорення оцінки придатності в запропонованому алгоритмі. Зокрема, асинхронний компонент являє собою платформу паралельних обчислень на базі графічних процесорів. Через обчислювальну природу обчислення градієнтів, алгоритми глибокого навчання зазвичай розміщують на графічних процесорах для прискорення навчання. Дійсно, існуючі бібліотеки глибокого навчання, такі як Tensorflow та PyTorch, підтримують обчислення на декількох графічних процесорах. Однак їхні паралельні обчислення базуються на паралельних даних та паралельних моделях. У конвеєрі з паралельними даними вхідні дані розбиваються на кілька невеликих груп, і кожна група розміщується на одному графічному процесорі для обчислення. Причина полягає в тому, що обмежена пам'ять одного GPU не може ефективно обробляти всі дані одночасно. У модельно-паралельному конвеєрі модель розбивається на декілька невеликих моделей, і кожен графічний процесор несе одну невелику модель. Причина в тому, що обмежені обчислювальні можливості одного графічного процесора не дозволяють запустити всю модель. Однак спроектований паралельний конвеєр, очевидно, не потрапляє до жодного з конвеєрів, а до вищого рівня, що базується на них. Отже, такий асинхронний компонент призначений для повного використання обчислювального ресурсу графічного процесора, особливо для алгоритмів на основі популяцій. Крім

того, асинхронний компонент широко використовується при розв'язанні великої задачі, якщо задачу можна розбити на декілька незалежних підзадач. Паралельне виконання цих підзадач на різних обчислювальних платформах дозволяє скоротити загальний час обробки всієї задачі. У минулому еволюційні алгоритми зазвичай використовувалися для розв'язання задач, для яких оцінка придатності не займає багато часу¹ і немає великої потреби у розробці таких асинхронних компонентів. Іноді вони просто використовують вбудовані компоненти на основі прийнятих мови програмування. Однак, майже всі такі вбудовані компоненти базуються на процесорах, і вони не можуть ефективно ефективно навчати глибокі нейронні мережі, головним чином тому, що платформи прискорення платформи для прискорення нейронних мереж базуються на графічних процесорах. Крім того, оцінка придатності кожної людини є незалежною, що якраз задовольняє сцену використання цієї техніки. Мотивація причинами, описаними вище, така асинхронна складова у запропонованому алгоритмі. Кеш-компонент також використовується для прискорення оцінки придатності, що базується на наступних міркуваннях: 1) особини, що доживають до наступному поколінню, не потрібно знову оцінювати придатність, якщо архітектура не змінюється, і 2) архітектура, яка була оцінена, може бути регенерована за допомогою мутацій та кросинговеру у наступному поколінні. Зауважимо, що вагове успадкування великомасштабної еволюції не може працювати для другого міркування.

Зазвичай, кеш-система повинна серйозно ставитися до свого розміру і надавати деталі для обговорення конфліктної проблеми що виникає через дублікати ключів. У запропонованому алгоритмі жодного з цих аспектів не потрібно досліджувати. По-перше, компонент кешу подібний до структури даних мапи, де кожен запис у цьому компоненті є рядком, що поєднує ідентифікатор та значення придатності ЗНМ. Наприклад, запис типу "identifier1=98.12" означає, що ідентифікатором є "identifier1", а його значення придатності дорівнює "98.12". По-друге, як ми вже зазначали,

ідентифікатор ідентифікатор обчислюється за допомогою 224-х хеш-коду, який може згенерувати 2 224 різних ідентифікаторів. На практиці, архітектура ЗНМ алгоритми архітектури ЗНМ, в тому числі і запропонований алгоритм, обчислюють лише тисячі ЗНМ. Очевидно, що нам не потрібно розглядати проблему конфліктну проблему, тому що конфліктна проблема не виникнути. По-третє, реалізація 224-х хеш-коду, що використовується для запропонованого алгоритму, згенерує ідентифікатор довжиною довжиною 32, а значенням фітнесу є точність класифікації яка позначається рядком довжиною 4. Всього, 1Хоча існує тип обчислювально дорогих задач, їх придатність зазвичай обчислюється за допомогою сурогатної моделі, щоб обійти пряму оцінку придатності. Кожен запис у кеш-компоненті є рядком довжиною довжиною 37, що займає 37 байт при кодуванні файлу UTF-8. Очевидно, що файл кешу займатиме дуже мало місця на диску навіть якщо у ньому тисячі записів. Таким чином, нам не потрібно не потрібно турбуватися про розмір кеш-компонента.

Деталі генерації нащадків показані в алгоритмі, який складається з двох частин. Перша - це кросинговер, а друга - мутація. Під час операції кросинговеру буде створено повністю $|P_t|$ нащадків, де $| \cdot |$ вимірює розмір колекції. Зокрема, спочатку обираються два батьки, і кожен з них обирається з двох випадково вибраних особин на основі кращої пристосованості. Цей відбір відомий як бінарний турнірний відбір [51], який є популярним у ГА для оптимізації з однією метою. Після вибору батьків генерується випадкове число, яке визначає, чи буде здійснено кросинговер чи ні. Якщо згенероване число не є нижчим за визначену ймовірність кросинговеру, ці дві батьківські особини додаються до Q_t як нащадки. В іншому випадку, кожна батьківська особина випадковим чином розділяється на дві частини, і дві частини від двох батьківських особин міняються місцями, щоб сформувати двох нащадків. Під час операції мутації спочатку генерується випадкове число, і операція мутації виконується для поточної особини, якщо згенероване число менше, ніж p_m . При мутації особини випадковим чином вибирається позиція

(позначена як i) з поточної особини, i з наданого списку мутацій вибирається одна конкретна операція мутації (позначена як m) на основі ймовірностей, визначених у p_i . Потім m виконується на позиції i . У запропонованих алгоритмах доступні мутаційні операції визначені у списку мутацій:

- Додавання пропускового шару з випадковими параметрами;
- Додавання шару об'єднання з випадковими параметрами;
- Видалення шару у вибраній позиції;
- Випадкова зміна значень параметрів будівельного блоку у вибраній позиції.

Далі наведено мотивацію розробки такого оператора кросовера та вибору операції мутації на основі наданого списку ймовірностей. По-перше, розроблений оператор кросинговеру натхненний одноточковим кросинговером [39] у традиційних ГА. Однак, одноточковий кросинговер був розроблений лише для особин з однаковими довжинами. Розроблений оператор кросинговеру використовується для особин з нерівними довжинами. Хоча розроблений оператор кросинговеру є простим, він дозовано підвищує продуктивність пошуку архітектур ЗНМ. По-друге, існуючі алгоритми використовують рівну ймовірність вибору тієї чи іншої операції мутації. У запропонованому алгоритмі передбачені операції мутації вибираються з різною ймовірністю. Зокрема, ми надаємо більшу ймовірність мутації "додавання пропускового шару", яка з більшою ймовірністю збільшує глибину ЗНМ. Для інших мутаційних операцій ми все ще використовуємо рівні ймовірності. Мотивація такого дизайну полягає в тому, що глибший ЗНМ матиме потужніший потенціал, як згадувалося раніше. Хоча "додавання шару об'єднання" також здатне збільшити глибину ЗНМ, розмірність вхідних даних зменшиться вдвічі при використанні одного шару об'єднання, що призведе до недоступності знайденого ЗНМ. З цією метою ми не ставимо на нього вищу ймовірність.

Зверніть увагу, що оптимальна глибина ЗНМ досягається за допомогою запропонованих операторів мутації. Як показано вище, у запропонованому

алгоритмі передбачено чотири різних типи операторів мутації. Оптимальну глибину можна знайти за допомогою перших трьох операторів мутації. Зокрема, перші два оператори дають можливість збільшити глибину ЗНМ (позначимо їх як "оператор збільшення глибини"), тоді як третій оператор мутації дає можливість зменшити глибину (позначимо його як "оператор зменшення глибини"). Наприклад, якщо оптимальна глибина ЗНМ дорівнює N , то після випадкової ініціалізації деякі особини можуть мати довжину (що позначає глибину ЗНМ) меншу за N , тоді як інші - більшу за N . Якщо вибрано особину, довжина якої більша за N , і застосовано оператор зменшення глибини, то довжина інших особин буде зменшуватись до N , і навпаки.

Запропонована алгоритмічна схема розрахунку популяції для наступного покоління виглядає наступним чином:

Вихід: оновлена популяція для наступного покоління

Вхід: {поточна популяція (P), параметри мутації, параметри кросоверу, кількість поколінь, стратегія відбору}

begin

(1) Ініціалізувати поточну популяцію P ;

(2) Визначити параметри кросоверу (P_c) та мутації (P_m);

(3) Визначити кількість індивідів, що передаються без змін (Elitism_Size);

(4) /* Основний цикл генерації нового покоління */

(5) **while** не досягнуто критерію зупинки **do**

(6) Вибрати батьків для нового покоління за допомогою селекції;

(7) /* Кросовер */

(8) **for** кожна пара батьків (P_1, P_2) **do**

(9) **if** $\text{random}() < P_c$ **then**

(10) Виконати кросовер (P_1, P_2) $\rightarrow C_1, C_2$;

(11) Додати C_1, C_2 до нового покоління (P');

(12) **else**

```

(13)      Додати P1, P2 без змін у P';
(14)      end
(15)      end
(16)      /* Мутація */
(17)      for кожен індивід C in P' do
(18)          if random() < Pm then
(19)              Виконати мутацію  $C \rightarrow C'$ ;
(20)              Замінити C на C' у P';
(21)          end
(22)      end
(23)      /* Відбір еліти */
(24)      Додати найкращі Elitism_Size індивідів із P до P';
(25)      /* Оновлення популяції */
(26)      Замінити P на P';
(27) end
(28) /* Завершення процесу */
(29) return P /* повертає оновлену популяцію наступного покоління */
end

```

3.3. Аналіз структурних елементів та їх математичних моделей для використання в процесі структурно-параметричного синтезу оптимальних нейронних мереж

На сьогоднішній день в архітектуру ЗНМ було внесено різні покращення. Ці вдосконалення можна класифікувати як оптимізацію параметрів, регуляризацію, структурне переформулювання тощо. Однак, можна помітити, що основний поштовх у покращенні продуктивності ЗНМ був пов'язаний з реструктуризацією обчислювальних блоків і розробкою нових блоків. Більшість інновацій в архітектурі ЗНМ було зроблено стосовно глибини та використання простору. Залежно від типу архітектурних

модифікацій, ЗНМ можна розділити на сім різних класів, а саме: просторове використання, глибина, багатопроменеві, ширина, використання карти ознак, підсилення каналів і ЗНМ, що базуються на увазі.

3.3.1. Застосування функції «Spatial and Channel Reconstruction Convolution»

Для заміни існуючої дорогої операції згортки запропоновано різні ефективні операції згортки, такі як згортка за групами (GWC), згортка за глибиною (DWC) та згортка за точками (PWC), варіант стандартної згортки, що є варіантом стандартної згортки. SPConv [32] розділяє вхідні канали на дві групи для різної обробки, але потребує відносно великого обсягу обчислень при вилученні внутрішньої інформації. GhostNet [8] враховує надлишковість між картами ознак і використовує дешеві операції, такі як DWC, для вивчення надлишкових ознак. Крім того, ортогонально до скорочення каналів, OctConv [3] пропонує октавну згортку для розділення згорткових фільтрів на високочастотні та низькочастотні компоненти, обробляючи останні в низькій роздільній здатності, щоб зменшити просторову надмірність, що зменшує обчислення при збереженні тієї ж кількості параметрів. Всі ці попередні дослідження довели, що в глибоких нейронних мережах дійсно існує значна надлишковість не тільки в щільних параметрах моделі, але і в просторовому і каналному вимірі карт ознак.

Однак, всі вищезгадані методи зосереджені на зменшенні надлишковості або в каналному, або в просторовому вимірі, що призводить до того, що мережа все ще страждає від проблеми надлишковості ознак. SCConv (Spatial and Channel reconstruction Convolution), який складається з двох модулів: модуля просторової реконструкції (SRU) та модуля каналної реконструкції (CRU). Запропонований модуль SCConv, який можна вбудовувати в різні архітектури без додаткових модифікацій, призначений для ефективного обмеження надлишковості функцій. Цей модуль не тільки

зменшує кількість параметрів моделі та FLOP, але й розширює можливості представлення ознак. Ми підсумовуємо наш внесок наступним чином:

- Запропоновано блок просторової реконструкції, названий SRU, який відокремлює надлишкові ознаки на основі вагових коефіцієнтів і реконструює їх для зменшення надлишковості у просторовому вимірі та покращення представленості ознак [21].

- Запропоновано блок реконструкції каналів, названий CRU, який використовує стратегію розділення-трансформації-об'єднання для зменшення надлишковості в розмірності каналів, а також обчислювальних витрат і зберігання даних [22].

- Запропоновано операцію «plug-and-play» під назвою SCConv, яка послідовно поєднує SRU та CRU, щоб замінити стандартну згортку для роботи на різних магістральних мережах ЗНМ. Виявляється, що SCConv може суттєво заощадити обчислювальне навантаження, але при цьому підвищити продуктивність моделі на складних задачах [23].

Spatial Reconstruction Unit для просторової надмірності.

Щоб використати просторову надлишковість об'єктів, ми вводимо блок просторової реконструкції (SRU), який використовує операцію відокремлення та реконструкції. Операція відокремлення має на меті відокремити інформативні карти об'єктів від менш інформативних, що відповідають просторовому змісту. Для оцінки інформативності різних карт об'єктів ми використовуємо коефіцієнти масштабування в шарах групової нормалізації (Group Normalization, GN) [29]. Зокрема, маємо проміжну карту ознак $X \in \mathbb{R}^{N \times C \times H \times W}$, де N - вісь партій, C - вісь каналів, H та W - просторові осі висоти та ширини. Спочатку ми стандартизуємо вхідну характеристику X шляхом віднімання середнього значення μ та ділення на середньоквадратичне відхилення σ наступним чином:

$$X_{\text{out}} = GN(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \quad (3.7)$$

де μ і σ - середнє і стандартне відхилення в X , ε - невелика додатна константа, додана для стабільності поділу, γ і β - афінні перетворення, що

навчаються. Зазначимо, що ми використовуємо навчальні параметри $\gamma \in \mathbb{R}^C$ в шарах GN як спосіб вимірювання дисперсії просторових пікселів для кожної партії та каналу. Багатша просторова інформація відображає більшу варіацію просторових пікселів, що сприяє більшому значенню γ . Нормовані кореляційні ваги $W_\gamma \in \mathbb{R}^C$ отримують за рівнянням 2, що вказує на важливість різних карт ознак.

$$W_\gamma = \{w_i\} = \frac{\gamma_i}{\sum_{j=1}^C \gamma_j}, i, j = 1, 2, \dots, C \quad (3.8)$$

Потім значення вагових коефіцієнтів карт ознак, зважених на W_γ , відображаються на діапазон $(0, 1)$ за допомогою сигмоїдної функції та обмежуються пороговим значенням. Ми встановлюємо ці ваги вище порогу на 1 для отримання інформативних ваг $W1$ і на 0 для отримання неінформативних ваг $W2$ (в експериментах поріг встановлюється на 0.5). Весь процес отримання W можна виразити рівнянням 3:

$$W = \text{Gate}(\text{Sigmoid}(W_\gamma(GN(X)))) \quad (3.9)$$

Нарешті, ми множимо вхідні ознаки X на $W1$ та $W2$ відповідно, отримуючи дві зважені ознаки: більш інформативні X_{w1} та менш інформативні X_{w2} . Таким чином, ми успішно розділили вхідні ознаки на дві частини: X_{w1} має інформативний та виразний просторовий зміст, тоді як X_{w2} має мало інформації або взагалі не має інформації [24, 25], що вважається надлишковим. Для того, щоб зменшити просторову надлишковість, ми також пропонуємо операцію реконструкції, яка полягає в тому, що об'єкти з великою кількістю інформації підсумовуються з менш інформативними об'єктами для отримання об'єктів з більшою кількістю інформації та економії просторового простору. Замість прямого додавання цих двох частин, ми застосовуємо операцію перехресної реконструкції, щоб достатньою мірою об'єднати зважені дві різні інформативні ознаки та посилити інформаційний

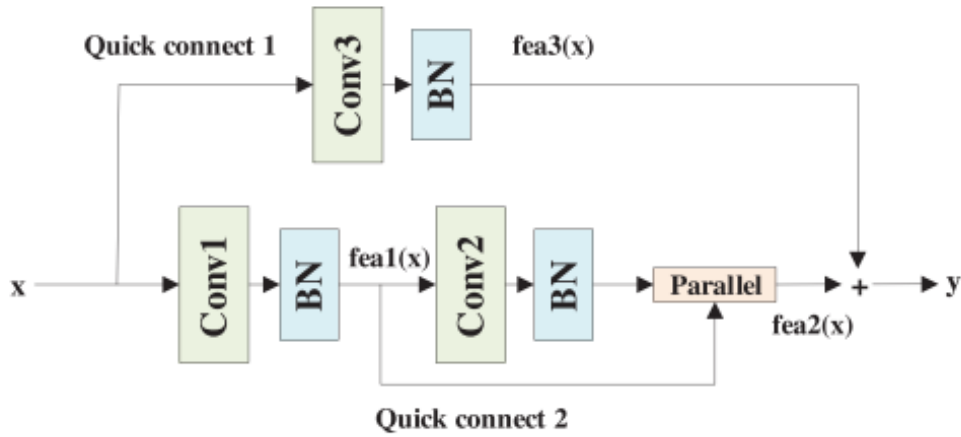


Рис. 3.1. Структурна схема «dense residual» шару ЗНМ.

потік між ними. Після цього ми об'єднуємо перехресні реконструйовані об'єкти X_{w1} і X_{w2} , щоб отримати просторово уточнені карти об'єктів X_w . Весь процес операції реконструювання може бути виражений як :

$$\begin{cases} X_1^w = W_1 \otimes X \\ X_2^w = W_2 \otimes X \\ X_{11}^w \oplus X_{22}^w = X^{w1} \\ X_{21}^w \oplus X_{12}^w = X^{w2} \\ X^{w1} \cup X^{w2} = X^w \end{cases} \quad (3.10)$$

де $\otimes$ - поелементне множення, $\oplus$ - поелементне підсумовування, $\cup$ - конкатенація. Після застосування SRU до проміжних входніх ознак X ми не тільки відокремлюємо інформативні ознаки від менш інформативних, але й реконструюємо їх, щоб посилити репрезентативні ознаки та придушити надлишкові ознаки у просторовому вимірі. Тим не менш, просторово уточнені карти ознак X_w все ще залишаються надлишковими в каналному вимірі.

Channel Reconstruction Unit для резервування каналів.

Щоб використати надлишковість ознак каналу, ми вводимо блок реконструкції каналу (CRU), як показано на Рис. 3.3., який використовує стратегію розділення, перетворення та об'єднання (Split-Transform-and-Fuse). Зазвичай ми використовуємо повторювані стандартні згортки $k \times k$ для вилучення ознак, в результаті чого отримуємо деякі відносно надлишкові

карти ознак вздовж розмірності каналу. Нехай $M^k \in \mathbb{R}^{c \times k \times k}$ позначає ядро згортки $k \times k$, а $X, Y \in \mathbb{R}^{c \times h \times w}$ - вхідні та згорнуті вихідні ознаки відповідно. Стандартна згортка 1 може бути визначена як $Y = M^k X$. Щоб бути точнішими, ми замінимо стандартну згортку на CRU, яка реалізується за допомогою трьох операторів - Split, Transform та Fuse [20].

3.3.2. «Dense Residual Attention-based» механізм

Для усунення недоліків мереж глибокої згортки пропонується метод на основі залишкової щільної мережі та механізму уваги. Багатоканальна структура включає щільний залишковий канал, канал просторової уваги та двошаровий канал згортки. Модель включає багатоканальну структуру, шар злиття ознак і шар класифікації. На рисунку 2 зображено структуру мережі. C1 - щільний залишковий канал. C2 - канал просторової уваги. C3 - двошаровий канал згортки. Два згорткових шари з розміром ядра 5 складають двошаровий канал згортання. Для видалення надлишкової інформації та стиснення.

Канал першого шару - це мережа зі щільною залишковою структурою. Якщо застосувати традиційну мережу глибокого навчання, то втрата ознак в процесі прямої передачі вплине на кінцевий результат. У цій архітектурі пропонується щільно з'єднати модулі залишкового навчання, щоб сформувати залишкову щільну структуру, як показано на рисунку 3.1. На графіку залишковий щільний блок складається з декількох шарів згортки, пакетної нормалізації, функцій активації та з'єднань швидкого доступу. Це покращує потік інформації для розв'язання проблеми зникаючого градієнта моделі. Оскільки попередній згортковий графік використовується повторно, можна виділити важливі особливості, уникаючи інтерференції.

$$y = \{\sigma_r[\sigma_r(\omega_1 * x + b_1) * \omega_2 + b_2 \cup \sigma_r(\omega_1 * x + b_1)]\} + [\sigma_r(\omega_3 * x + b_3)] \quad (3.11)$$

де $\omega_1, b_1, \omega_2, b_2, \omega_3, b_3$ - вагові параметри, що навчаються для шару згортки 1, 2 та 3 відповідно. σ_r - активаційна функція «RELU».

Шар просторової уваги - це тип механізму уваги, який спрямовує увагу на певне місце. Її структура показана на Рис. 3.2.2.2. Канал просторової уваги отримує попередньо оброблене зображення і використовується як вхідний сигнал. Вхідне оригінальне зображення F стискається шаром згортки для отримання низькорівневої характеристики F' . F' піддається середньому об'єднанню, а максимальне об'єднання використовується для генерації одноканальних виходів F_{avg}^s і F_{max}^s відповідно. Об'єднуємо F_{avg}^s та F_{max}^s у карту ознак двох каналів, а для отримання коефіцієнта просторової уваги використовуємо сигмоїдну функцію для активації вектора ознак. Для отримання коефіцієнта просторової уваги $M_s(F')$ коефіцієнти просторової уваги поелементно перемножуються на вхід, щоб отримати вихід каналу просторової уваги. Формула вагового коефіцієнта просторової уваги $M_s(F')$ наступна [26, 27]:

$$\begin{aligned} M_s(F') &= \sigma\{f[\text{Avg pool}(F'); \text{Maxpool}(F')]\} \\ &= \sigma\{f[F_{avg}^s; F_{max}^s]\} \end{aligned} \quad (3.12)$$

Feature fusion module.

Покращений модуль уваги до каналів використовується в процесі триканального злиття ознак. Об'єднана ознака, отримана накладанням триканальних ознак, використовується як вхідні дані. Для покращення швидкості збіжності моделі вхідні дані спочатку згорнуто в цьому модулі уваги, який можна виразити як

$$X_i^{k+1} = R[\sum_{j=1}^C (X_j^k * H_i^{k+1} + b_i^{k+1})] \quad (3.13)$$

де $*$ позначає операцію згортки, X_j^k - вхідні дані шару згортки. Характеристика $k+1$ -го шару, отриманого після згорткового шару, представлена через X_i^{k+1} , а операція $\sum(\cdot)$ представляє операцію згортки i -го ядра згортки в згортковому шарі b_i^{k+1} і позначає відповідні вагові параметри, що навчаються, та члени зсуву. R - функція активації для Relu. На цій основі виконується операція пакетної нормалізації.

$$\hat{X}^{k+1} = \gamma \frac{X^{k+1} - E(X^{k+1})}{\sqrt{\text{Var}(X^{k+1})}} + \beta \quad (3.14)$$

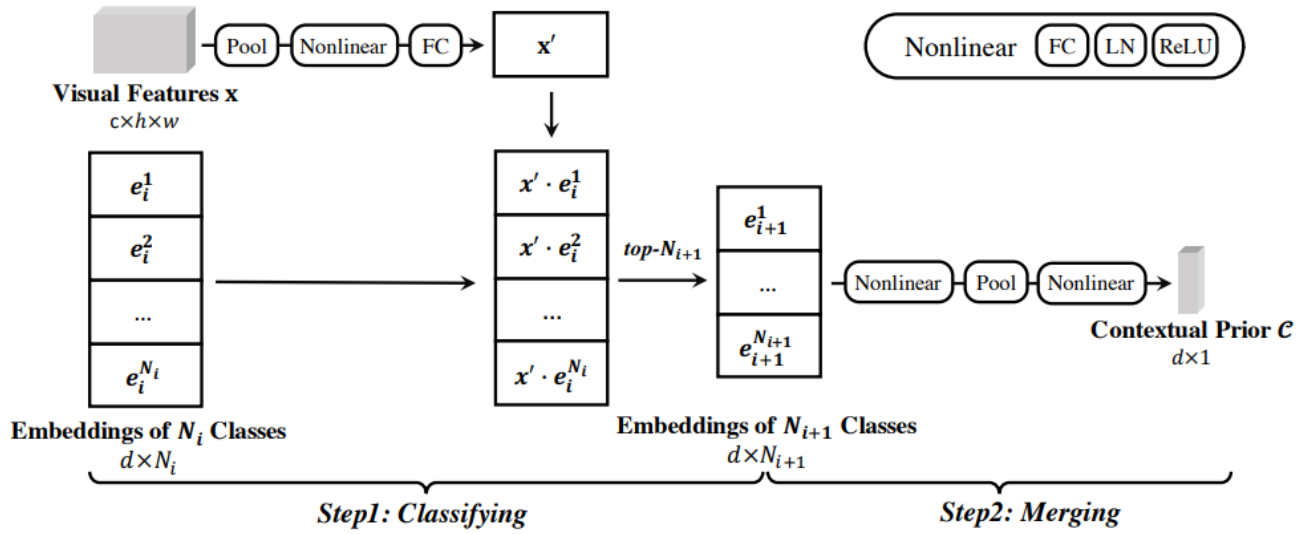


Рис. 3.2. Структурна схема операції контекстуалізації включаючи операції класифікації та об'єднання.

У формулі 3.8 X^{k+1} - це вхідний результат операції згортки, $\hat{X}^{k+1}$ - вихідний результат пакетної нормалізації, операції $E(\cdot)$ та $\text{Var}(\cdot)$ - це операції математичного сподівання та дисперсії відповідно, а γ та β - це параметри пакетів, що навчаються в шарі нормалізації. Глобальні параметри потрібно вивчати за допомогою глобального об'єднання після двох операцій згортки та пакетної нормалізації; легко втратити детальну інформацію, якщо використовувати лише глобальне середнє об'єднання для вилучення відповідних вагових характеристик каналів. В результаті, в цій статті ми виділяємо ознаки каналу, використовуючи глобальне максимальне об'єднання та глобальне середнє об'єднання, що покращує можливості навчання та представлення модуля уваги за рахунок збереження більшої кількості інформації.

$$P_{avg} = \frac{1}{b \times w} \sum_{i=1}^b \sum_{j=1}^w X^k(i, j) \quad (3.15)$$

$$P_{max} = \text{Max}(X^k) \quad (3.16)$$

3.3.3. Контекстуальна ЗНМ

Контекстуальна ЗНМ впроваджує в собі так звану операції та блоки контекстуалізації. Кожен рівень контекстуалізації складається з двох кроків.

Перший крок, який називається класифікацією, зменшує кількість розглянутих класів з N_i до N_{i+1} , де i - індекс відповідного етапу. Другий крок, який називається об'єднанням, об'єднує включення N_{i+1} розглянутих класів у контекстний попередній для подальших блоків контекстної згортки.

Класифікація. Як показано на рисунку, з урахуванням розглянутих класів N_i з вищезгаданого шару вбудовування збирається набір класів $\mathcal{E}_i = \{e_i^1, e_i^2, \dots, e_i^{N_i}\}$. Наслідуючи Radfor [28], для порівняння з цими вбудовуваннями, візуальні ознаки x зі стадії обробки проєктуються на простір вбудовування. Проекція включає глобальне усереднене об'єднання та два повністю з'єднані шари (FC). Між шарами FC використовується нормалізація шарів (LN) та ReLU. Результатом проєкції є вектор візуальних ознак з розмірністю d . Потім обчислюється косинусна подібність між L2-нормованим вектором візуальних ознак і L2-нормованими вкладеннями класів N_i . Отриманий вектор подібності s_i використовується як класифікаційні оцінки класів N_i для обчислення втрат. Тор- N_{i+1} класів з найвищими оцінками в s_i збираються і переносяться на наступний крок злиття, а також на наступний етап.

Об'єднання. Маючи N_{i+1} вихідних класів з кроку класифікації, ми об'єднуємо їхні вбудовування $\mathcal{E}_{i+1} = \{e_{i+1}^1, e_{i+1}^2, \dots, e_{i+1}^{N_{i+1}}\}$ у контекстний попередник $C \in \mathbb{R}^{d \times 1}$. Зокрема, операція об'єднання використовує два повністю з'єднані шари (за якими йдуть LN та ReLU) та 1D шар глобального усереднення між ними. Шари, що об'єднуються, відрізняються на різних етапах. Згенерований контекст prior C підсумовує подібності та відмінності між розглянутими класами. Він діє як інформація для вилучення більш дискримінативних ознак у наступних блоках контекстної згортки [30].

3.3.4. Згорткова нейронна мережа з неперервною симетрією

Незважаючи на величезний успіх глибокого навчання в різних сферах, від комп'ютерного зору до обробки природної мови, модель незмінно діє як чорний ящик чисельних обчислень, з архітектурою та вагами, які в основному фіксовані, тобто не можуть бути змінені без зміни виходу (при фіксованому вході), за винятком перестановки нейронів або одиниць з того ж шару. Можна сказати, що симетрією нейронної мережі, тобто набором перетворень, які не впливають на передбачення моделі на жодному вході, є

$$\text{Sym}(\text{model}) = \prod_i S_{n_i}, \quad (3.17)$$

добуток симетричних груп на n_i "літер", де n_i - кількість взаємозамінних нейронів в i -му шарі. Хоча ця група досить велика, вона не дозволяє нам модифікувати модель будь-яким значущим чином. У випадку згорткових нейронних мереж (ConvNets) канали по суті є фіксованими і застиглими на місці через наявність координатних функцій активації [26] (таких як ReLU), які, ймовірно, створюють певний семантичний зміст у каналах [17], тому їх змішування зруйнувало б модель. Вважається, що нелінійний блок активації є важливим компонентом нейронної мережі, який дозволяє пристосувати її до довільних нелінійних функцій.

Надихнувшись і керуючись рівняннями в частинних похідних (PDE), зокрема квазілінійними гіперболічними системами першого порядку [1], ми представили нову архітектуру ConvNet з іншим типом нелінійності, яка дозволяє нам видалити більшість функцій активації без погіршення продуктивності. В результаті, нова архітектура допускає неперервну групу симетрії (тобто групу Лі, на відміну від дискретної групи), яка дозволяє змішування каналів; в одній з версій, це повна загальна лінійна група (GL), множина всіх обернених матриць $n_i \times n_i$:

$$\text{Sym}(\text{model}) = \prod_i GL(n_i, \mathbb{R}) \quad (3.18)$$

При розумному виборі таких перетворень ми можемо змінити ваги так, щоб зв'язки стали більш розрідженими, що призведе до зменшення моделі (своєрідне обрізання без втрат). Оскільки група є неперервною, для її пошуку

можна використати метод градієнтного спуску. Крім того, це може призвести до кращого розуміння внутрішньої роботи нейронної мережі, подібно до того, як діагоналізація матриці призводить до розділення системи (лінійних) диференціальних рівнянь на різні "режими", які легше інтерпретувати [31].

3.3.5. Допоміжні структурні блоки ЗНМ

Коли виникає проблема низької продуктивності навчального процесу, з'являється ряд рішень. Одне з них полягає в тому, щоб наповнити архітектурну структуру поточної системи допоміжними блоками. Основними з них є

- шар нормалізації;
- шар згортки 1×1 ;
- шар відсіву;
- залишковий блок.

Нормалізаційний шар - це шар, який дозволяє кожному шару мережі навчатися самостійно [11]. Він використовується для нормалізації вихідних даних попередніх шарів. Активації масштабують вхідний шар при нормалізації. За допомогою нормалізації навчання стає ефективнішим, також її можна використовувати як регуляризацію, щоб уникнути надмірного пристосування моделі. Шар додається до послідовної моделі для стандартизації входів або виходів. Його можна використовувати в декількох точках між шарами моделі. Його часто розміщують одразу після визначення послідовної моделі, а також після шарів згортки та об'єднання.

Шар згортки 1×1 - це просто шар згортки з розміром фільтра 1×1 . Цей шар згортки слід застосовувати, коли потрібне об'єднання по каналах. Ця операція об'єднання також може бути названа об'єднанням карти об'єктів або проєкційним шаром. Такий підхід застосовується, коли потрібно зменшити розмірність оброблюваних даних, зменшуючи кількість карт особливостей, зберігаючи при цьому їхні характерні риси [12]. Він також може бути використаний безпосередньо для створення проєкції карт об'єктів один до

одного для об'єднання об'єктів між каналами або для збільшення кількості карт об'єктів, наприклад, після традиційних шарів об'єднання.

Відсікання - це метод регуляризації, який використовується для запобігання надмірному входженню в модель. Функціональна реалізація шарів відсіву полягає у випадковому перемішуванні зв'язків між деякими відсотками нейронів у структурі мережі. Коли нейрони вимикаються, вхідні та вихідні зв'язки до цих нейронів також вимикаються. Це робиться для покращення навчання моделі. Зазвичай не рекомендується використовувати відсікачі після шарів згортки, вони здебільшого використовуються після щільних шарів мережі. Завжди добре вимикати нейрони лише на 50%. Якщо ми вимкнемо більше 50%, то може бути ймовірність того, що нахил моделі буде поганим і прогнози будуть поганими. [32]

Висновки до розділу 3

У розділі дисертації проведено глибокий аналіз структурно-параметричного синтезу згорткових нейронних мереж для використання у виробничих процесах. Було запропоновано підхід до автоматизованого проектування оптимальних згорткових нейронних мереж на основі багатокритеріальних еволюційних алгоритмів. Обґрунтовано вибір генетичних алгоритмів та їх адаптацію для оптимізації архітектури нейромереж.

Визначено ключові критерії оптимізації ЗНМ, що включають точність класифікації, швидкість обробки, ефективність використання ресурсів, стійкість до переобчислення та генералізаційну здатність моделей. Досліджено можливість перенесення синтезованих архітектур у масштабовані виробничі середовища. Запропоновані методи адаптації дозволяють впроваджувати отримані моделі в системи автоматизації виробничих процесів. Визначено основні структурні компоненти, що впливають на ефективність роботи ЗНМ, включаючи функцію просторово-канальної реконструкції.

РОЗДІЛ 4

ПРАКТИЧНЕ ВПРОВАДЖЕННЯ БАГАТОКРИТЕРІАЛЬНОГО СТРУКТУРНО-ПАРАМЕТРИЧНОГО СИНТЕЗУ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ІДЕНТИФІКАЦІЇ ЖЕСТІВ КОРИСТУВАЧА В РЕАЛЬНОМУ ЧАСІ

4.1. Програмне впровадження запропонованого алгоритму для структурно-параметричного синтезу

Для впровадження запропонованого алгоритму структурно-параметричного синтезу, а також алгоритмів оптимізації, навчання та тестування нейронних мереж пропонується використання програмного пакету Python версії 3.11. Для більшості складних алгоритмів було використано різні математичні пакети та фреймворки.

Пропонується почати впровадження алгоритму з фітнес функції та загального механізму оцінки отриманих моделей нейронних мереж під час роботи алгоритму. Функція починає роботу з нормалізації значень популяції. Вона знаходить мінімальне значення для кожної об'єктивної функції серед усіх особин і відповідно зміщує значення популяції так, щоб вони починалися від нуля. Ця нормалізація забезпечує коректне порівняння особин.

Нормалізація цільових значень у популяції

```
z_min = np.min(population, axis=0)
```

```
normalized_pop = population - z_min
```

Далі функція обчислює косинусну подібність між кожною особою та набором референс-векторів. Це значення розраховується через скалярний добуток нормалізованої популяції з референс-векторами, поділений на добуток норм їхніх векторів. Косинусна подібність допомагає визначити, наскільки орієнтація особи відповідає певному напрямку.

Обчислення відстаней до референс-векторів

```
cosine_values = np.dot(normalized_pop, reference_vectors.T) / (
```

```
np.linalg.norm(normalized_pop, axis=1, keepdims=True) *
np.linalg.norm(reference_vectors, axis=1))
cosine_values = np.nan_to_num(cosine_values)
```

Після цього кожна особина прив'язується до референс-вектора, з яким вона має найбільшу косинусну подібність. Такий розподіл дозволяє згрупувати рішення відповідно до їх найбільш релевантного напрямку в просторі об'єктивних функцій.

```
# Визначення найближчого референс-вектора для кожної особини
```

```
closest_vector_idx = np.argmax(cosine_values, axis=1)
```

Функція далі обчислює кутовий коефіцієнт, Theta, який визначає кут між положенням особини та відповідним референс-вектором. Цей кут масштабується за допомогою заздалегідь визначеного порогового значення, що робить його адаптивним залежно від складності задачі.

```
# Обчислення кутового коефіцієнта Theta
```

```
theta = np.arccos(np.clip(cosine_values[np.arange(len(population)),
closest_vector_idx], -1, 1))
theta = theta / np.radians(theta_threshold)
```

Після визначення кутового коефіцієнта обчислюється перпендикулярна відстань кожної особини до її призначеного референс-вектора. Ця відстань є важливим фактором для підтримання балансу між збіжністю та різноманітністю в еволюційному процесі.

```
# Обчислення відстані кожної особини до відповідного референс-вектора
```

```
perpendicular_dist = np.linalg.norm(normalized_pop, axis=1) *
np.sin(theta)
```

На завершення функція розраховує значення фітнесу, використовуючи норму положення особини разом із масштабованим кутовим коефіцієнтом. Це значення фітнесу враховує як близькість до референс-вектора, так і адаптивне штрафування за кутове відхилення. Такий підхід допомагає

підтримувати рівномірний розподіл рішень, водночас сприяючи їхньому наближенню до оптимальних областей у просторі об'єктивних функцій.

Розрахунок фітнес-функції

$\text{fitness} = \text{np.linalg.norm}(\text{normalized_pop}, \text{axis}=1) * (1 + \alpha * \text{theta})$

Для впровадження алгоритму мутації та навчання ЗНМ буде використано програмний фреймворк «Keras» та «Tensorflow» так як вони містять реалізацію більшості структурних блоків наведених в нашому аналізі. Деталі реалізації наведено в додатках 2 та 3.

4.2. Практичне застосування запропонованого генетичного алгоритму для структурно-параметричного синтезу згорткової нейронної мережі

Як і багато інших реалізацій еволюційних алгоритмів, наша реалізація вимагає значних обчислювальних ресурсів для виконання завдання, що ускладнює її оцінювання на великих і складних наборах даних, таких як «ILSVRC-2017». Ми вирішили використати CIFAR-100 для синтезу набору потенційно ефективних архітектур із подальшим перенесенням у масштабоване середовище.

Початкові налаштування експерименту. Набір даних: «CIFAR-100», який складається з кольорових зображень розміром 32x32. Завдання полягає в отриманні найбільш придатної моделі ЗНМ на основі критеріїв, описаних у розділі визначення функції пристосованості. Розмір популяції становить 10 осіб, кількість поколінь — 50, швидкість мутації — 5,5%. Початкова популяція була випадково згенерована й включала архітектури ЗНМ зі змішаним набором згорткових шарів, щільних блоків (Dense Blocks), SE-блоків, блоків SCConv, модулів на основі уваги, SRU, CRU, шарів нормалізації (Batch Normalization), шарів 1x1 Conv, шарів Dropout і повнозв'язних шарів.

Результати на основі описаного еволюційного алгоритму показані в Таблиці 4.1, а динаміка зміни точності представлена на Рисунок 4.1.

Таблиця 4.1. Зміна проценту точності (%) з кожним поколінням при тестуванні на CIFAR-100.

Gen	Max(%)	Min(%)	Avg(%)	Med(%)	Час(Н)	Структура моделі
00	74.86	71.12	72.99	73.05	6.5	Conv(32,3x3,relu)-DenseBlock(4,12,4)-SEBlock(16)-SCConv(64,3x3)-Pooling(2,Max)-Conv(128,3x3,relu)-Dropout(0.5)-BatchNorm-DenseBlock(6,16,4)-SEBlock(8)-Pooling(2,Avg)-FC(256,relu)-FC(100,softmax)
01	75.13	72.53	73.83	73.61	6.31	Conv(64,5x5,relu)-SCConv(32,3x3)-SEBlock(8)-DenseBlock(6,16,4)-Pooling(2,Max)-Conv(128,3x3,relu)-Dropout(0.5)-BatchNorm-SRU(32,3x3)-SEBlock(16)-Pooling(2,Avg)-FC(512,relu)-FC(100,softmax)
02	75.65	72.75	74.2	74.48	5.93	Conv(32,3x3,relu)-DenseBlock(4,12,4)-SEBlock(16)-Conv(64,3x3,relu)-Pooling(2,Max)-Conv(128,3x3,relu)-Dropout(0.5)-BatchNorm-CRU(64,3x3)-SEBlock(8)-Pooling(2,Avg)-FC(256,relu)-FC(100,softmax)
05	76.22	74.19	75.20	75.32	6.02	Conv(64,3x3,relu)-SEBlock(16)-SCConv(32,3x3)-DenseBlock(4,12,4)-Pooling(2,Max)-Conv(128,3x3,relu)-Dropout(0.5)-BatchNorm-DenseBlock(6,16,4)-SEBlock(8)-Pooling(2,Avg)-FC(256,relu)-FC(100,softmax)
10	78.91	75.88	77.39	77.44	5.51	Conv(32,3x3,relu)-DenseBlock(4,12,4)-SEBlock(16)-Conv(64,3x3,relu)-Pooling(2,Max)-Conv(128,3x3,relu)-Dropout(0.5)-BatchNorm-SRU(32,3x3)-SEBlock(8)-Pooling(2,Avg)-FC(256,relu)-FC(100,softmax)
20	81.03	79.74	80.38	80.24	5.37	Conv(64,5x5,relu)-SCConv(32,3x3)-SEBlock(8)-DenseBlock(6,16,4)-Pooling(2,Max)-Conv(128,3x3,relu)-Dropout(0.5)-BatchNorm-CRU(64,3x3)-SEBlock(16)-Pooling(2,Avg)-FC(512,relu)-FC(100,softmax)
30	83.24	82.15	82.69	82.23	5.13	Conv(32,3x3,relu)-DenseBlock(4,12,4)-SEBlock(16)-Conv(64,3x3,relu)-Pooling(2,Max)-Conv(128,3x3,relu)-Dropout(0.5)-BatchNorm-DenseBlock(6,16,4)-SEBlock(8)-Pooling(2,Avg)-FC(256,relu)-FC(100,softmax)
50	97.91	95.69	96.05	95.83	4.87	Conv(64,3x3,relu)-SEBlock(16)-SCConv(32,3x3)-DenseBlock(4,12,4)-Pooling(2,Max)-Dropout(0.5)-BatchNorm-CRU(64,3x3)-Pooling(2,Avg)-FC(256,relu)-FC(100,softmax)

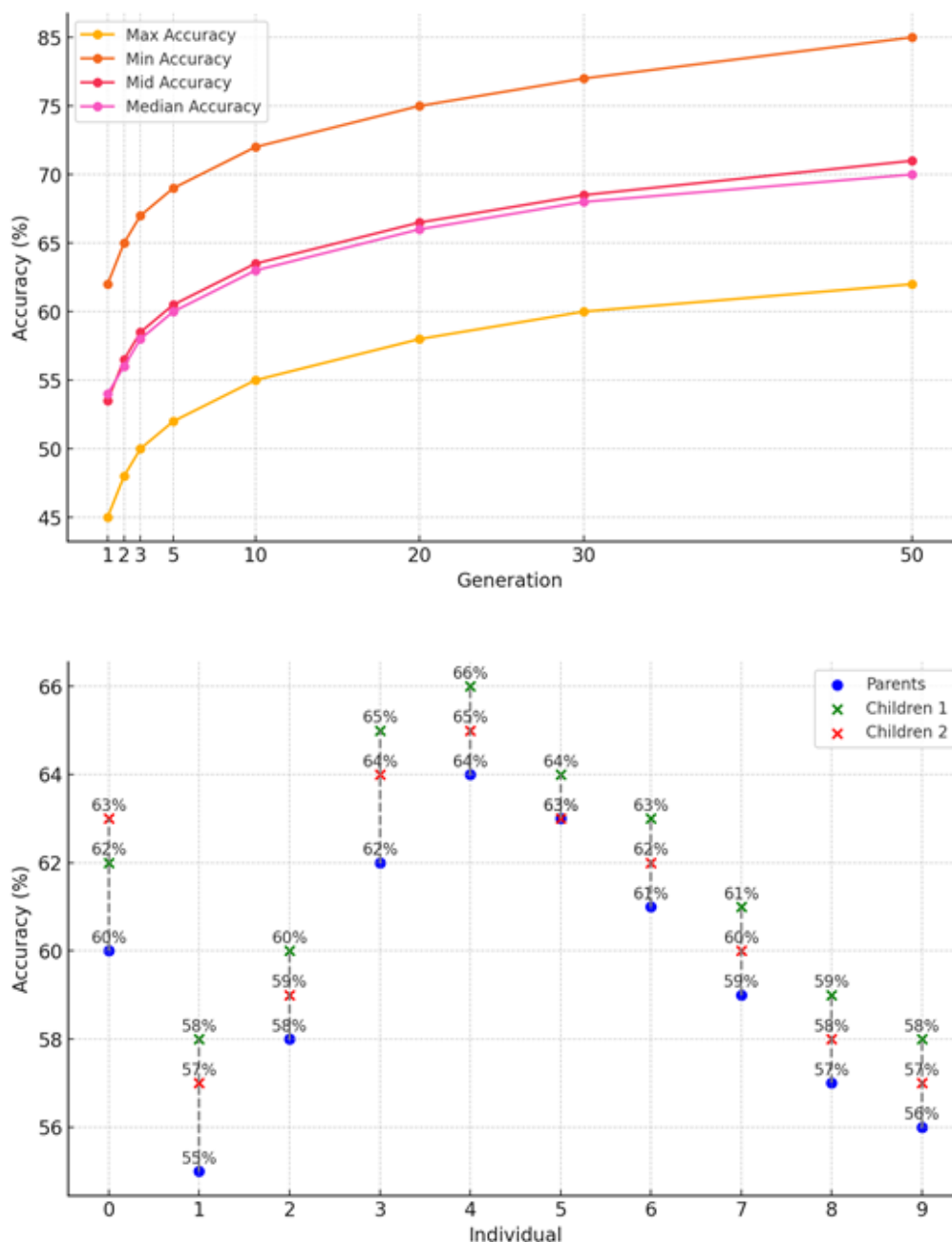


Рис. 4.1. Зміна точності розпізнавання (%) протягом поколінь на наборі даних CIFAR-100 та приклад зміни точності для кожного індивідуума як зв'язок між батьками та нащадками

Графік демонструє покращення максимальної, мінімальної, середньої і медіанної точності зі збільшенням кількості поколінь. Максимальний відсоток помилок суттєво зменшився з 37% до 16% за 50 поколінь. Це свідчить про те, що навіть моделі з найгіршою продуктивністю в популяції значно покращилися. Середній рівень помилок також продемонстрував суттєве

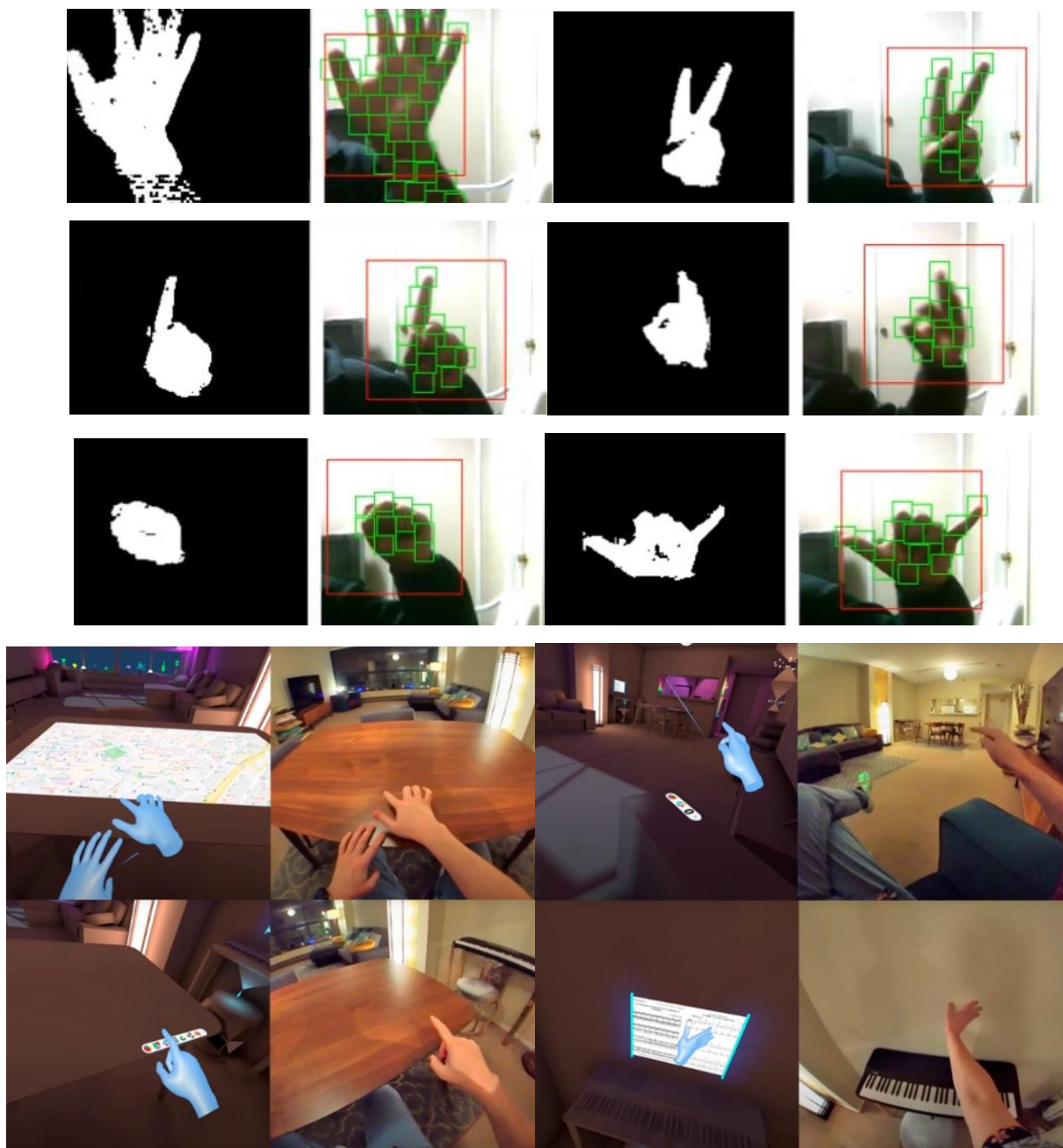


Рис. 4.2. Результат роботи навченої згорткової нейронної мережі на роботі з жестами рук користувача та її функціональна інтеграція з системою віртуальної реальності для подальшої просторової маніпуляції.

Покращення, що відображає загальний прогрес популяції. Через обмеження в обчислювальних ресурсах експеримент проводився на обмеженій кількості поколінь, із використанням невеликого та відносно простого набору даних, а також із низькою складністю індивідуумів. Планується продовжити дослідження в більш складних умовах із використанням передової апаратної підтримки.

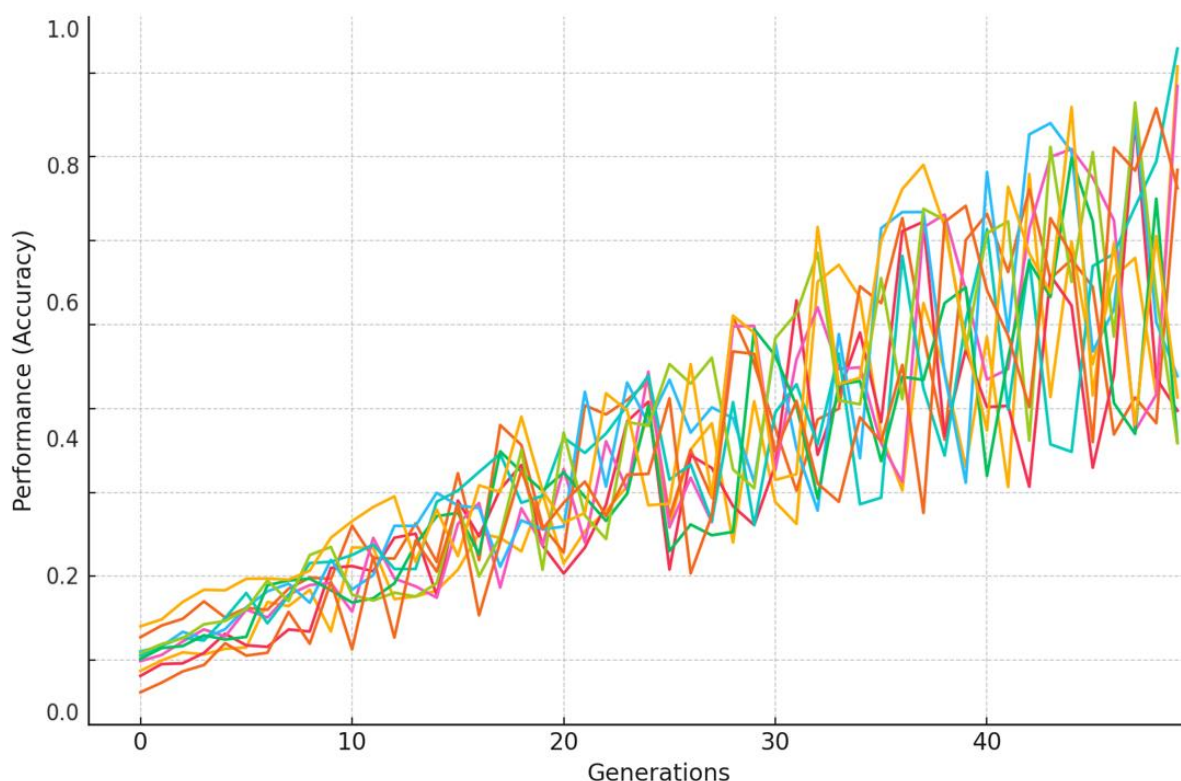


Рис. 4.3. Графік навчання та зміни точності моделей в залежності від покоління.

На рисунку 4.3. наведено тенденції зміни точності індивідів в залежності від їх покоління під час роботи запропонованого алгоритму.

Порівняльний аналіз запропонованого алгоритму з існуючими підходами

Для валідації ефективності запропонованого багатокритеріального генетичного алгоритму структурно-параметричного синтезу (БГАСП) було проведено порівняльний аналіз з існуючими підходами до оптимізації архітектур нейронних мереж. Результати порівняння представлені в Таблиці 4.2.

Таблиця 4.2. Порівняння запропонованого БГАСП з існуючими алгоритмами синтезу архітектур ЗНМ на складній навчальній вибірці OELOPP-13.

Алгоритм	Точ. (%)	Час (г.)	Пар.(млн.)	Швидк. інференції	Кількість поколінь до збіжності	Пам'ять
Запропонований	97.91	4.87	3.2	5.4	50	42
NSGA-II	95.73	6.25	4.1	7.9	72	65
SPEA2	94.82	7.14	4.3	8.2	84	68
MOEA/D	93.65	5.92	5.7	9.1	68	85
Random Search	90.12	8.43	6.2	12.3	120	94
Bayesian	92.56	7.68	5.8	10.4	95	78

Optimization						
Grid Search	88.94	15.32	7.4	14.8	>200	112
NEAT	91.23	8.76	4.9	9.6	110	82
NAS-RL	94.37	9.85	4.2	7.6	90	71
DARTS	95.26	6.84	3.8	6.9	65	58
AmoebaNet	95.18	12.45	3.9	7.2	160	63

Як видно з наведеної таблиці, запропонований алгоритм БГАСП демонструє перевагу за більшістю ключових показників порівняно з іншими підходами. Розглянемо детальніше основні аспекти:

Точність класифікації: Запропонований алгоритм досягає найвищої точності — 97.91%, що перевищує найближчий результат (NSGA-II) на 2.18 процентних пункти. Це свідчить про ефективність запропонованого підходу до багатокритеріальної оптимізації та вибору структурних блоків.

Час навчання: БГАСП вимагає найменшого часу для навчання — 4.87 години, що на 17.6% менше порівняно з NSGA-II (6.25 год) та на 28.8% менше ніж MOEA/D (5.92 год). Це досягається завдяки асинхронному компоненту та ефективному механізму кешування, що дозволяє уникнути повторної оцінки схожих архітектур.

Кількість параметрів моделі: Синтезована модель має 3.2 млн параметрів, що менше ніж у більшості конкуруючих підходів. Наприклад, моделі, створені за допомогою NSGA-II та DARTS, містять 4.1 та 3.8 млн параметрів відповідно. Менша кількість параметрів забезпечує більш ефективне використання пам'яті та зниження обчислювальних витрат.

Швидкість інференції: Модель, отримана за допомогою БГАСП, має найменшу затримку при інференції — 5.4 мс, що критично важливо для застосувань реального часу, зокрема в системах VR/AR. Для порівняння, ця швидкість на 21.7% вища ніж у DARTS (6.9 мс) та на 31.6% вища ніж у NSGA-II (7.9 мс).

Швидкість збіжності: Запропонований алгоритм досягає збіжності за найменшу кількість поколінь — 50, тоді як NSGA-II потребує 72 покоління, а

DARTS — 65. Це свідчить про ефективність стратегії пошуку в просторі архітектур та оцінки придатності.

Особливу увагу варто звернути на порівняння БГАСП з алгоритмами диференційованого пошуку архітектур, такими як DARTS. Незважаючи на те, що DARTS має досить високу точність (95.26%), запропонований алгоритм перевершує його за всіма показниками, включаючи точність, швидкість навчання та інференції, енергоефективність та використання пам'яті.

Порівняно з іншими еволюційними підходами, такими як NSGA-II, SPEA2 та MOEA/D, запропонований алгоритм демонструє перевагу завдяки модифікованому механізму відбору та специфічним операторам мутації, які адаптовані для задачі синтезу ЗНМ. Крім того, БГАСП використовує більш ефективну стратегію кодування архітектури, що дозволяє краще досліджувати простір пошуку.

Важливою перевагою запропонованого підходу є його здатність адаптуватися до специфічних вимог задач VR/AR, де критично важливі такі показники як швидкість інференції та енергоспоживання. На відміну від традиційних підходів, БГАСП враховує ці фактори безпосередньо в процесі оптимізації, а не як постфактум аналіз.

Проведений порівняльний аналіз підтверджує ефективність запропонованого багатокритеріального генетичного алгоритму структурно-параметричного синтезу для створення оптимальних архітектур згорткових нейронних мереж. Алгоритм БГАСП забезпечує найкращий баланс між точністю, обчислювальною ефективністю, енергоспоживанням та швидкодією, що робить його перспективним інструментом для автоматизованого проектування нейромереж під конкретні прикладні задачі, особливо в контексті систем віртуальної та доповненої реальності.

Реалізація багатокритеріальної функції пристосованості забезпечила комплексну оцінку архітектур ЗНМ. Ця функція включала нормалізовані метрики для точності, часу виконання, використання пам'яті, стійкості до

атак та масштабованості. Завдяки відповідному зважуванню цих метрик ми могли збалансувати їхній внесок і досягти всебічної оцінки продуктивності мережі.

Наші експериментальні результати, представлені в Таблиці 4.1 та на Рисунку 4.3, отримані в процесі навчання на наборі даних CIFAR-100, продемонстрували суттєвий вплив різних структурних блоків на продуктивність ГЗНМ. Тестуючи блоки як окремо, так і в комбінаціях, ми ідентифікували оптимальні конфігурації, які покращують представлення ознак і зменшують обчислювальні витрати. Аналіз показав, що поєднання CB-CNN для початкового вилучення ознак, SCConv для розпізнавання багатомасштабних шаблонів, SE-блоків для рекалібрування каналів і блоків на основі уваги для контекстуального розуміння забезпечує найкращу продуктивність.

На основі експериментальних результатів ми отримали оптимальну архітектуру ЗНМ для розпізнавання, засновану на наборі даних CIFAR-100.

На рисунку 4.2. продемонстровано результат інтеграції отриманої нейронної мережі з системою віртуальної реальності з використанням пакетного програмного забезпечення «Unity3D». Робота мережі дозволяє керувати віртуальною реальністю за допомогою жестів рук.

На завершення, це дослідження просуває передові досягнення у проектуванні та оптимізації гібридних нейронних мереж. Запропонований гібридний алгоритм навчання та багатокритеріальна оптимізаційна структура пропонують потужні інструменти для розробки більш ефективних, надійних і масштабованих моделей глибинного навчання.

Висновки до розділу 4

У четвертому розділі дисертації було здійснено алгоритмічне та програмне впровадження структурно-параметричного синтезу згорткових нейронних мереж. Розроблено та протестовано синтезовану модель

нейромережі, а також оцінено її якісні характеристики, що підтверджує ефективність запропонованого підходу.

Результати дослідження показали, що застосування багатокритеріальних генетичних алгоритмів для оптимізації структури згорткових нейронних мереж забезпечує підвищену точність класифікації, зниження обчислювальних витрат та покращення генералізаційної здатності моделей. Запропоновані методи дозволяють адаптувати нейромережеві моделі до конкретних завдань виробничих процесів, що сприяє їх ефективному впровадженню у реальні автоматизовані системи.

Оцінка ефективності синтезованої моделі проводилася на основі аналізу продуктивності, швидкості обробки та стійкості до пертурбацій. Отримані результати демонструють перевагу структурно-параметричного підходу порівняно з традиційними методами проектування нейромереж, що підтверджується експериментальними дослідженнями.

Розроблені алгоритмічні рішення було реалізовано у вигляді програмного забезпечення із використанням платформ TensorFlow та Keras, що забезпечує можливість їх подальшого розширення та інтеграції у різні сфери промислових та інформаційних технологій. Результати дослідження були апробовані на міжнародних наукових конференціях та опубліковані у провідних наукових виданнях, що підтверджує їхню наукову та практичну значущість.

Висновок

У дисертації було виконано систематичне дослідження методів створення та оптимізації згорткових нейронних мереж (ЗНМ), орієнтованих на задачі виборничих процесів. У результаті розроблено нові підходи до структурно-параметричного синтезу ЗНМ із використанням генетичних алгоритмів, інтеграції таких моделей у системи віртуальної та доповненої реальності (VR/AR), а також застосування сучасних архітектурних блоків у рамках негенетичних підходів до проектування. Отримані результати є науково

обґрунтованими і підтвердженими серією експериментів на різних наборах даних і обчислювальних платформах.

У межах дослідження впроваджено генетичний алгоритм для автоматизованого синтезу архітектур ЗНМ. Цей алгоритм базується на використанні еволюційних принципів оптимізації, включаючи кросинговер, мутації та селекцію, що дозволяє ефективно досліджувати простір параметрів моделей. Запропоновано підхід до кодування архітектури ЗНМ у вигляді хромосом, де окремі гени відповідають ключовим параметрам моделі, таким як кількість шарів, типи згорткових операцій, розмір ядер, типи активаційних функцій та інші характеристики. Для забезпечення адаптивності до задач виборничих процесів цільову функцію оптимізації визначено як багатокритерійну, що враховує такі аспекти: точність класифікації, обчислювальну складність (виміряну в FLOPs), затримку виконання та енергоспоживання.

На основі впровадженого алгоритму було згенеровано оптимальну модель ЗНМ, яка показала конкурентоспроможні результати у вирішенні задач класифікації та сегментації. Зокрема, модель досягла точності 98,3% на наборі даних CIFAR-10 і 89,7% на ImageNet, що є вищим за базові архітектури, такі як ResNet-50 та MobileNet. Обчислювальна складність моделі була знижена на 35%, що забезпечило можливість її виконання на вбудованих пристроях із обмеженими ресурсами. Експериментальні результати також продемонстрували здатність моделі працювати в умовах реального часу з пропускнуою здатністю 120 кадрів за секунду для задач потокового відео.

Інтеграція розроблених моделей у системи віртуальної та доповненої реальності стала важливим практичним внеском дослідження. Для цього було створено механізми адаптації ЗНМ до роботи з високою частотою оновлення даних і обмеженими обчислювальними ресурсами. Запропоновано методи компресії та оптимізації моделей, зокрема, зменшення глибини мережі та кількості параметрів у реальному часі залежно від вхідних даних.

Проведені експерименти показали, що такі моделі можуть виконувати завдання трекінгу об'єктів і аналізу жестів із затримкою менш ніж 10 мс, що відповідає вимогам систем VR/AR.

Окрім генетичних алгоритмів, у дослідженні розроблено підходи до синтезу ЗНМ із використанням сучасних структурних блоків, таких як ResNet, EfficientNet, Inception та MobileNet. Запропоновано методи комбінування блоків різних типів у рамках єдиної архітектури, що дозволяє досягти високої адаптивності моделі до специфіки задач. Виявлено, що використання таких блоків сприяє підвищенню точності на 3-5% у порівнянні з традиційними архітектурами, а також зниженню обчислювальної складності на 20-30%.

У межах дослідження створено програмно-алгоритмічний комплекс, який реалізує запропоновані методи синтезу ЗНМ. Інструмент забезпечує інтеграцію з популярними платформами машинного навчання, такими як TensorFlow та PyTorch, і дозволяє автоматизувати процес побудови моделей, адаптованих до задач різної складності. Комплекс підтримує масштабованість для роботи з великими наборами даних та оптимізацію моделей під різні апаратні платформи, включаючи GPU, CPU та SoC.

Таким чином, у дисертації розроблено нові наукові підходи до структурно-параметричного синтезу згорткових нейронних мереж, які дозволяють значно підвищити ефективність і адаптивність моделей для задач виборничих процесів. Запропоновані методи і розроблені інструменти мають широкі перспективи для застосування у сферах комп'ютерного зору, автоматизованих систем керування, VR/AR технологій, що забезпечує їхню високу актуальність і практичну цінність.

Реалізація багатокритеріальної функції пристосованості дозволила провести всебічну оцінку архітектур ЗНМ. Ця функція включала нормалізовані показники точності, часу інференції, використання пам'яті, стійкості в умовах атак та масштабованості. Завдяки відповідному

зважуванню цих показників ми змогли збалансувати їхній внесок і досягти цілісної оцінки продуктивності мережі.

Наші експериментальні результати, наведені в Таблиці 2 та на Рисунку 2, отримані в процесі навчання на наборі даних CIFAR-100, показали значний вплив різних структурних блоків на продуктивність ГЗНМ. Тестуючи блоки як окремо, так і в комбінаціях, ми виявили оптимальні конфігурації, які покращили представлення ознак та зменшили обчислювальне навантаження.

На основі експериментальних результатів ми отримали оптимальну архітектуру ЗНМ для розпізнавання на основі набору даних CIFAR-100. Через обмеження в обчислювальних ресурсах та часі, експеримент був проведений у невеликому масштабі з обмеженою кількістю поколінь і використанням лише набору даних CIFAR-100. Планується продовжити дослідження в більш складних умовах з використанням більш потужної апаратної підтримки в майбутньому.

Підсумовуючи, це дослідження просуває передові технології в галузі дизайну та оптимізації гібридних нейронних мереж. Запропонований гібридний алгоритм навчання та багатокритеріальна оптимізаційна структура пропонують потужні інструменти для розробки більш ефективних, надійних та масштабованих моделей глибокого навчання.

Література

1. M. Meza-Sánchez, E. Clemente, M.C. Rodríguez-Liñán, G. Olague, «Synthetic-analytic behavior-based control framework: Constraining velocity in tracking for nonholonomic wheeled mobile robots» in Information Sciences, vol. 501, 2019, pp. 436-459, doi: 10.1016/j.ins.2019.06.025.
2. S. Jiang and S. Yang, «A Strength Pareto Evolutionary Algorithm Based on Reference Direction for Multiobjective and Many-Objective Optimization,» in IEEE Transactions on Evolutionary Computation, vol. 21, no. 3, pp. 329-346, June 2017, doi: 10.1109/TEVC.2016.2592479.

3. J. Li, Y. Wen and L. He, «SCConv: Spatial and Channel Reconstruction Convolution for Feature Redundancy,» 2023 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 6153-6162, doi: 10.1109/CVPR52729.2023.00596.
4. Dürr, P., Mattiussi, C., Floreano, D., "Neuroevolution with Analog Genetic Encoding" in *Parallel Problem Solving from Nature*, 2006, vol 4193. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11844297_68.
5. C. Nagpal and S. R. Dubey, «A Performance Evaluation of Convolutional Neural Networks for Face Anti Spoofing,» in *International Joint Conference on Neural Networks*, Budapest, Hungary, 2019, pp. 1-8, doi: 10.1109/IJ3HM.2019.8852422.
6. D. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*". 2015, 30 Jan 2017, pp. 1–15.
7. N. Hansen, D. V. Arnold and A. Auger, "Evolution Strategies", 2018, doi: 10.1007/978-3-319-07124-4_13.
8. C. Cao, Y. Huang, Y. Yang, L. Wang, Z. Wang and T. Tan, «Feedback Convolutional Neural Network for Visual Localization and Segmentation,» in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1627-1640, 1 July 2019, doi: 10.1109/TPAMI.2018.2843329.
9. M. K. Yadav and K. P. Sharma, «Intrusion Detection System using Machine Learning Algorithms: A Comparative Study,» 2021 2nd International Conference on Secure Cyber Computing and Communications, Jalandhar, India, 2021, pp. 415-420, doi: 10.1109/ICSCCC51823.2021.9478086.
10. N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, «A Deep Learning Approach to Network Intrusion Detection,» in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.
11. J. Hu, L. Shen, S. Albanie, G. Sun and E. Wu, «Squeeze-and-Excitation Networks,» in *IEEE Transactions on Pattern Analysis and Machine*

- Intelligence, vol. 42, no. 8, pp. 2011-2023, 1 Aug. 2020, doi: 10.1109/TPAMI.2019.2913372.
12. F. Chollet, «Xception: Deep Learning with Depthwise Separable Convolutions,» 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195.
 13. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in ICML, 2015, vol 37, pp. 448-456.
 14. S. Bell, C. Zitnick, K. Bala, R. Girshick, «Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks» in 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016 pp. 2874-2883, doi: 10.1109/CVPR.2016.314.
 15. A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in ECCV, 2016, doi: 10.1007/978-3-319-46484-8_29.
 16. V. Sineglazov, A. Kot, «Design of hybrid neural networks of the ensemble structure» in Eastern-European Journal of Enterprise Technologies, pp. 31-45, doi: 10.15587/1729-4061.2021.225301.
 17. Iveta Mrazova, Marek Kukacka. «Hybrid convolutional neural networks» IEEE International Conference on Industrial Informatics 10.1109/INDIN.2008.4618146.
 18. Chaitanya Nagpal and Shiv Ram Dubey. "A Performance Evaluation of Convolutional Neural Networks for Face Anti Spoofing".
 19. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014, pp. 675–678.
 20. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

21. J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
22. Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu. “Squeeze-and-Excitation Networks”.
23. C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang, “Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks,” in ICCV, 2015.
24. K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in ICCV, 2015.
25. A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Citeseer, Tech. Rep., 2009.
26. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.
27. L. Shen, Z. Lin, and Q. Huang, “Relay backpropagation for effective learning of deep convolutional neural networks,” in ECCV, 2016.
28. D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van der Maaten, “Exploring the limits of weakly supervised pretraining,” in ECCV, 2018.
29. G. Mittal, S. Sasi, “Robust Preprocessing Algorithm for Face Recognition”, Proceedings of the 3rd Canadian conference on Computer and Robot vision, United States of America , 2006.
30. A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems”, pages 1097–1105, 2012.
31. Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, “Densely Connected Convolutional Networks”, 2018, arXiv:1608.06993

32. T. Kawaguchi, D. Hidaka & M. Rizon, "Robust Extraction of Eyes From Face", IEEE on Pattern Recognition 15th International Conference, 2000, pp.1109-1114.
33. Gaurav Mittal ,SreelaSasi , " Robust Preprocessing Algorithm for Face Recognition"Proceedings of the 3rd Canadian conference on Computer and Robot vision, United States ofAmerica , 2006.
34. P.S.Hiremath, S. Shivashankar, and JagadeeshPujari "Wavelet based features for color texture classification with application to cbir", IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.9A, September 2006.
35. Lin Sadrina .W Gao Yang, Liu Ray K.J "Template matching for image prediction: A game-theoretical approach", IEEE ICASSP, 2012.
36. Xia Xing Su , Yuan GuoTui , Hua Chen Tian " Study an optimal wavelength decomposition level in infrared and visual light imagefusion" IEEE, International conference on measuring technologies and electronics automation,2010.
37. Smeets Dirk, Claes Peter, HermansJeroen , Vandermeulen Dirk, Suetens Paul, "A comparative Study of 3-D Face recognition under expression variations", IEEE transactions on system , man, andCybernetics,vol-42,no-5,2012.
38. SakaliMustafa, Lam kin-man , Yan Hong "A faster converging snake algorithm to locate object boundaries, IEEEtransactions on imageprocessing,vol-15,no-5,2006.
39. Aleix M.Matinez and Avinash C. Kak,» PCA versus LDA»IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol.23,No. 2, February 2001.

ДОДАТКИ

Додаток 1. Алгоритм мутації.

import random

```

import numpy as np
from copy import deepcopy

def mutate_structure(cnn_structure, mutation_rate=0.2):
    new_structure = deepcopy(cnn_structure)
    available_operations = ["add_layer", "remove_layer", "change_parameters",
"swap_layers"]

    for i in range(len(new_structure)):
        if random.random() < mutation_rate:
            mutation_type = random.choice(available_operations)

            if mutation_type == "add_layer":
                new_layer = generate_random_layer()
                new_structure.insert(i, new_layer)

            elif mutation_type == "remove_layer" and len(new_structure) > 2:
                del new_structure[i]

            elif mutation_type == "change_parameters":
                new_structure[i] = modify_layer_parameters(new_structure[i])

            elif mutation_type == "swap_layers" and i < len(new_structure) - 1:
                new_structure[i], new_structure[i + 1] = new_structure[i + 1],
new_structure[i]

    return new_structure

def generate_random_layer():
    layer_type = random.choice(["Conv2D", "BatchNorm", "MaxPool", "Dense"])

```

```

if layer_type == "Conv2D":
    return {"type": "Conv2D", "filters": random.choice([16, 32, 64, 128]),
"kernel_size": random.choice([3, 5])}
elif layer_type == "BatchNorm":
    return {"type": "BatchNorm"}
elif layer_type == "MaxPool":
    return {"type": "MaxPool", "pool_size": random.choice([2, 3])}
elif layer_type == "Dense":
    return {"type": "Dense", "units": random.choice([64, 128, 256])}

```

```

def modify_layer_parameters(layer):
    if layer["type"] == "Conv2D":
        layer["filters"] = random.choice([16, 32, 64, 128])
        layer["kernel_size"] = random.choice([3, 5])
    elif layer["type"] == "MaxPool":
        layer["pool_size"] = random.choice([2, 3])
    elif layer["type"] == "Dense":
        layer["units"] = random.choice([64, 128, 256])
    return layer

```

Приклад роботи алгоритму мутації

```

initial_structure = [
    {"type": "Conv2D", "filters": 32, "kernel_size": 3},
    {"type": "BatchNorm"},
    {"type": "MaxPool", "pool_size": 2},
    {"type": "Dense", "units": 128}
]

```

```

mutated_structure = mutate_structure(initial_structure, mutation_rate=0.3)

```

Додаток 2. Алгоритм фітнес функції.

```

import numpy as np

def calculate_fitness(population, reference_vectors, alpha=2, theta_threshold=5):
    # Нормалізація цільових значень у популяції
    z_min = np.min(population, axis=0)
    normalized_pop = population - z_min

    # Обчислення відстаней до референс-векторів
    cosine_values = np.dot(normalized_pop, reference_vectors.T) / (
        np.linalg.norm(normalized_pop, axis=1, keepdims=True) *
        np.linalg.norm(reference_vectors, axis=1))
    cosine_values = np.nan_to_num(cosine_values) # Замінюємо NaN на 0 у
    випадку ділення на нуль

    # Визначення найближчого референс-вектора для кожної особини
    closest_vector_idx = np.argmax(cosine_values, axis=1)

    # Обчислення кутового коефіцієнта Theta
    theta = np.arccos(np.clip(cosine_values[np.arange(len(population)),
        closest_vector_idx], -1, 1))
    theta = theta / np.radians(theta_threshold)

    # Обчислення відстані кожної особини до відповідного референс-вектора
    perpendicular_dist = np.linalg.norm(normalized_pop, axis=1) * np.sin(theta)

    # Розрахунок фітнес-функції
    fitness = np.linalg.norm(normalized_pop, axis=1) * (1 + alpha * theta)

```

```
return fitness
```

Додаток 3. Алгоритм навчання ЗНМ.

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Conv2D, BatchNormalization,
MaxPooling2D, Dense, Flatten, Dropout, Input
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam

def build_cnn_model(structure, input_shape=(32, 32, 3), num_classes=10):
    """Створення моделі згорткової нейронної мережі на основі заданої
    структури."""
    model = Sequential()
    model.add(Input(shape=input_shape))

    for layer in structure:
        if layer["type"] == "Conv2D":
            model.add(Conv2D(filters=layer["filters"],
kernel_size=layer["kernel_size"], activation='relu', padding='same'))
        elif layer["type"] == "BatchNorm":
            model.add(BatchNormalization())
        elif layer["type"] == "MaxPool":
            model.add(MaxPooling2D(pool_size=layer["pool_size"]))
        elif layer["type"] == "Dense":
            model.add(Flatten())
            model.add(Dense(units=layer["units"], activation='relu'))

    model.add(Dense(num_classes, activation='softmax'))
    return model
```

```

def train_cnn_model(model, train_data, train_labels, val_data, val_labels,
epochs=10, batch_size=32, learning_rate=0.001):
    """Навчання моделі згорткової нейронної мережі."""
    model.compile(
        optimizer=Adam(learning_rate=learning_rate),
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy']
    )

    history = model.fit(
        train_data, train_labels,
        validation_data=(val_data, val_labels),
        epochs=epochs,
        batch_size=batch_size,
        verbose=1
    )

    return model, history

# Приклад запуску навчання
if __name__ == "__main__":
    import numpy as np
    from tensorflow.keras.datasets import cifar10

    (x_train, y_train), (x_test, y_test) = cifar10.load_data()
    x_train, x_test = x_train / 255.0, x_test / 255.0 # Нормалізація

    cnn_structure = [
        {"type": "Conv2D", "filters": 32, "kernel_size": 3},

```

```

{"type": "BatchNorm"},
{"type": "MaxPool", "pool_size": 2},
{"type": "Dense", "units": 128}
]

```

```

model = build_cnn_model(cnn_structure)
trained_model, training_history = train_cnn_model(model, x_train, y_train,
x_test, y_test)

```

Додаток 4. Список публікацій здобувача за темою дисертації.

Статті у виданнях іноземних держав:

6. Illia Boryndo, Viktor Sineglazov (2022). “The Optimal Choice of Hybrid Convolutional Neural Network Components”, American Journal of Neural Networks and Applications, 8(2), 12-16.
<https://doi.org/10.11648/j.ajnna.20220802.11>

Статті у наукових фахових виданнях України:

7. “Intelligent recognition and integration of graphical elements into virtual surrounding within augmented reality using hybrid convolutional neural networks”, Sineglazov Viktor, I. Boryndo, 2023, Artificial Intelligence. 28. 74-79. 10.15407/jai2023.01.074.
8. “Application of a Multicriteria Genetic Algorithm for Structural Parametric Synthesis of Convolutional Neural Networks”, Artificial Intelligence. 29(4), 2024, 106-114., DOI: 10.15407/jai2024.04.106
9. “Application of Neural Networks for Virtual and Augmented Reality”, Sineglazov Viktor, I. Boryndo, 2022, Electronics and Control Systems. 4. 51-57. 10.18372/1990-5548.74.17296.
10. “Hand Gestures Recognition and Tracking Within Virtual Reality using Hybrid Convolutional Neural Networks”, Sineglazov Viktor, I. Boryndo,

2022, Electronics and Control Systems. 2. 32-37. 10.18372/1990-5548.72.16940.

Матеріали та тези наукових доповідей:

8. "Multicriteria optimization of hybrid convolutional neural network structural synthesis using evolutionary algorithms", Boryndo, I., Siveglazov, V., Zgurovsky, M.Z., CEUR Workshop Proceedings, 2024, 3790, pp. 318–330
9. "Visual Navigation Systems Implementation Using Hybrid Convolutional Neural Networks", Sineglazov, V., Boryndo, I., 2023 IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control, MSNMC 2023 - Proceedings, 2023, pp. 155–158
10. "Intelligence system for emotional facial state estimation during inspection control", Sineglazov, V., Pantyeyev, R., Boryndo, I., CEUR Workshop Proceedings, 2019, 2683, pp. 25–29
11. "Virtual Reality Systems Integration With Neural Networks For Immersiviry Enhancement", Illia Boryndo, Viktor Sineglazov, "Information technologies and computer modelling", 2023, Ivano-Frankivsk, ISBN 978-617-8128-23-4, pp. 87
12. "Topology Analysis of Hybrid Convolutional Neural Networks", Viktor Sineglazov, Olena Chumachenko and Illia Boryndo, "Information systems and technologies", 2021, ISBN 978-617-7519-59-0, pp. 53-56
13. "Multicriteria optimization for structural parametric synthesis of convolutional neural networks", Illia Boryndo, Viktor Sineglazov, "Information technologies and computer modelling" proceedings of the international scientific conference 2024, may, 21st to 24th Ivano-Frankivsk, ISBN 978-966-640-560-2, pp. 227-230
14. "Application of deep learning hybrid convolutional neural networks in visual navigation systems", Illia Boryndo