

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Інститут(факультет)	Комп'ютерних наук і програмної інженерії
Кафедра	Комп'ютерної математики і аналізу даних
Спеціальність	113 Прикладна математика
Освітня програма	Інтелектуальний аналіз даних
Спеціалізація	113.01 Інтелектуальний аналіз даних

До захисту допускаю

Завідувач кафедри

Л.М. Любчик

(ініціали та прізвище)

(підпис, дата)

ДИПЛОМНА РОБОТА

другого (магістерського) рівня вищої освіти

Тема роботи Автоматизована побудова архітектури глибоких нейронних мереж для вирішення задачі класифікації зображень

Шифр роботи КН-53АМН.02

(група, номер теми за наказом )

Виконавець Кузьмін Кирило Юрійович

(прізвище, ім'я, по-батькові)

Керівник асист., Колбасін Владислав Олександрович

(посада, прізвище, ім'я, по-батькові)

Харків 2019

[illegible]

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Інститут (факультет)	Комп'ютерних наук і програмної інженерії
Кафедра	Комп'ютерної математики і аналізу даних
Рівень вищої освіти	Другий (магістерський)
Спеціальність	113 Прикладна математика
Освітня програма	Інтелектуальний аналіз даних
Спеціалізація	113.01 Інтелектуальний аналіз даних (шифр і назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри Любчик Л.М.

« \_\_\_\_ » \_\_\_\_\_ 2019 року

**ЗАВДАННЯ  
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Кузьміну Кирилу Юрійовичу

1 Тема роботи Автоматизована побудова архітектури глибоких  
нейронних мереж для вирішення задачі класифікації зображень

керівник роботи Колбасін Владислав Олександрович, асист.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 12.02.2019 року № 250 СТ

2 Строк подання студентом роботи 15.05.2019 року

3 Вихідні дані до роботи Масивна база даних анатованих зображень

ImageNet

4 Перелік питань, які потрібно розробити у пояснювальній записці  
Огляд методів та моделей для вирішення задачі класифікації зображення; метод DARTS; методи автоматизованого побудування архітектури згорткових мереж; економічне обґрунтування; громадянська оборона; охорона праці; висновки;

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Презентація - аркушів А4

#### 6 Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічне обґрунтування	Александрова Т.Є., професор		
Охорона праці та навколишнього середовища	Кузьменко О.О., доцент		
Цивільна оборона	Кузьменко О.О., доцент		

7 Дата видачі завдання 04.02.2019

#### КАЛЕНДАРНИЙ ПЛАН

Номер етапу	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітки
1	Вибір і обґрунтування теми, постановка проблем і	04.02.2019	
2	Аналітичний огляд джерел, вибір методики досліджень	15.02.2019	
3	Підготовка і виконання пояснювальної записки	01.05.2019	
4	Економічне обґрунтування та підготовка висновків	01.05.2019	
5	Складання відомості документів, оформлення ПЗ	10.05.2019	
6	Виконання презентації, доповіді	15.05.2019	
7	Подання ДР на відгук та зовнішню рецензію	15.05.2019	
8	Подання ДР на допуск до захисту	15.05.2019	
9	Захист ДР	22.05.2019	

Студент

(підпис)

Кузьмін К. Ю.

(прізвище та ініціали)

Керівник роботи

(підпис)

Колбасін В. О.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Інститут	Комп'ютерних наук і програмної інженерії
Кафедра	Комп'ютерної математики і аналізу даних
Спеціальність	113 Прикладна математика
Освітня програма	Інтелектуальний аналіз даних
Спеціалізація	113.01 Інтелектуальний аналіз даних

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

другого (магістерського) рівня вищої освіти

на тему Автоматизована побудова архітектури глибоких нейронних  
мереж для вирішення задачі класифікації зображень

Виконав студент 6 курсу, групи КН-53АМН  
Кузьмін К.Ю.

(підпис, прізвище та ініціали)

Керівник Колбасін В.О.

(підпис, прізвище та ініціали)

Рецензент Воронін А.В.

(підпис, прізвище та ініціали)

Нормоконтроль Сазонова С.О.

(підпис, прізвище та ініціали)

Харків 2019

## РЕФЕРАТ

Пояснювальна записка до ДП: 89 с., 25 рис., 8 табл., 20 джерел  
Інформації

Ключові слова: *АЛГОРИТМ, DARTS, ENAS, RNN, CNN, PYTHON, TENSORFLOW, KERAS, NUMPY, MATPLOTLIB*

Даний дипломний проект присвячений алгоритмам автоматичної побудови архітектур згорткових нейронних мереж.

Метою роботи є ознайомлення з основними методами і способами класифікації зображення за допомогою згортальних нейронних мереж, побудова та тренування нейронної мережі.

Завдання було виконано за допомогою стандартних математичних методів і засобів високорівневого мови програмування Python, і за допомогою загальнодоступних бібліотек Keras, Numpy, TensorFlow.

## РЕФЕРАТ

Пояснительная записка к ДП: 89 с., 25 рис., 8 табл., 20 источников информации

*Ключевые слова: АЛГОРИТМ, DARTS, ENAS, RNN, CNN, PYTHON, TENSORFLOW, KERAS, NUMPY, MATPLOTLIB*

Данный дипломный проект посвящен алгоритмам автоматического построения архитектур сверточных нейронных сетей.

Целью работы является ознакомление с основными методами и способами классификации изображения с помощью сверточных нейронных сетей, построение и тренировка нейронной сети.

Задача была выполнена с помощью стандартных математических методов и средств высокоуровневого языка программирования Python, и с помощью общедоступных библиотек Keras, Numpy, TensorFlow.

## ABSTRACT

Explanatory note to the PD: 89 p., 25 fig., 8 tab., 20 sources

Information

Keywords: *ALGORITHM, DARTS, ENAS, RNN, CNN, PYTHON, TENSORFLOW, KERAS, NUMPY, MATPLOTLIB*

This project is dedicated to algorithms for the automatic construction of architectures of convolutional neural networks.

The point of the work is to familiarize with the basic methods and methods of image classification using convolutional neural networks, the construction and training of a neural network.

The task was performed using standard mathematical methods and means of a high-level programming language Python, and using publicly available libraries Keras, Numpy, TensorFlow.



## ЗМІСТ

Вступ.....	5
1 Рішення задачі класифікації за допомогою згорткових нейронних мереж .....	6
1.1 Задача класифікації зображень.....	6
1.2 Згорткові нейронні мережі .....	6
1.2.1 Основні шари в згорткових мережах .....	8
1.3 Згорткові нейронні мережі для класифікації зображень створені людьми .....	11
1.3.1 AlexNet.....	11
1.3.2 ZF Net.....	13
1.3.3 VGG Net.....	14
1.3.4 Inception .....	15
1.3.5 ResNet .....	16
1.3.6 Inception-v4 та Inception-ResNet .....	17
1.4 Постановка задачі .....	18
1.5 Висновки до розділу .....	19
2 Пошук архітектур нейронних мереж.....	20
2.1 Пошук нейронної архітектури за допомогою навчання з підкріпленням .....	20
2.1.1 Навчання з підкріпленням .....	22
2.1.2 Навчання згорткової мережі на CIFAR-10 .....	24
2.2 Регуляризована еволюція для пошуку архітектур класифікатора зображень .....	27
2.2.1 Пошуковий простір.....	27
2.2.2 Еволюційний алгоритм .....	29
2.2.3 Результати .....	33

2.3 Висновки до розділу .....	35
3 Рішення задачі пошуку архітектури.....	36
3.1 DARTS .....	36
3.1.1 Пошуковий простір.....	37
3.1.2 Безперервна релаксація та оптимізація.....	38
3.1.3 Наближення .....	40
3.1.4 Виведення дискретних архітектур .....	42
3.1.5 Пошук архітектури згорткової мережі на CIFAR-10 .....	43
3.1.6 Результати навчання .....	44
3.2 Висновки до розділу .....	49
4 Економічне обґрунтування.....	50
4.1 Розрахунок кошторису витрат на проведення й впровадження результатів науково-дослідної роботи.....	50
4.1.1 Розрахунок фонду заробітної плати виконавців .....	50
4.1.2 Відрахування на соціальне страхування.....	53
4.1.3 Розрахунок технологічної електроенергії.....	53
4.1.4 Розрахунок електроенергії, що витрачає на освітлення .....	54
4.1.5 Амортизаційні відрахування на устаткування .....	54
4.1.6 Вартість оренди приміщення для проведення НДР .....	55
4.1.7 Інші витрати .....	55
4.1.8 Вартість впровадження й освоєння результатів НДР .....	55
4.1.9 Витрати на проведення НДР .....	56
4.1.10 Планові накопичення .....	56
4.1.11 Кошторис витрат на проведення НДР .....	57
4.2 Класифікація й кодування запропонованої інновації .....	57
4.3 Розрахунок економічного ефекту від впровадження результатів НДР .....	59

4.4 Укрупнена оцінка прибутковості запропонованого інноваційного проекту .....	60
4.5 Висновки за розділом.....	63
5 Охорона праці і навколишнього середовища .....	65
5.1 Аналіз умов праці на робочому місці .....	65
5.2 Захист від шкідливого впливу факторів виробничого середовища ...	67
5.3 Електробезпека.....	71
5.4 Пожежна безпека .....	78
5.5 Охорона навколишнього середовища .....	80
5.6 Висновки до розділу .....	81
6 Цивільний захист.....	82
6.1 Система державного управління у сфері Цивільного захисту .....	82
6.2 Висновки до розділу .....	85
Висновки.....	86
Список джерел інформації.....	87

## ВСТУП

Відкриття сучасних архітектур нейронних мереж вимагає значних зусиль фахівців. Останнім часом спостерігається зростаючий інтерес до розробки алгоритмічних рішень для автоматизації процесу проектування архітектури. Архітектури, шукані програмами, досягли високої конкурентоспроможності в таких завданнях, як класифікація зображень і виявлення об'єктів.

Кращі алгоритми пошуку існуючої архітектури вимагають дуже багато обчислень, незважаючи на їх чудову продуктивність. Наприклад, для отримання сучасної архітектури для CIFAR-10 і ImageNet було потрібно 1800 днів GPU навчання з підкріпленням [1] або 3150 GPU днів еволюції [2]. Запропоновано кілька підходів для прискорення, таких як запровадження конкретної структури простору пошуку, вагові коефіцієнти або прогноз продуктивності для кожної окремої архітектури і розподіл ваги між декількома архітектурами, але основна проблема залишається. Невід'ємною причиною неефективності для домінуючих підходів, наприклад на основі навчання з підкріпленням, еволюції, пошуку по дереву Монте-Карло [3], оптимізації на основі послідовної моделі [4], або байєсівської оптимізації, є той факт, що пошук архітектури розглядається як чорна коробка. Також є проблема над дискретним простором пошуку, що призводить до великої кількості необхідних архітектурних оцінок.

# **1 РІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ**

## **1.1 Задача класифікації зображень**

Однією з базових задач в машинному зорі є класифікація зображень - визначення категорій об'єктів, який знаходиться на зображенні. Залежно від конкретного завдання, на зображенні може бути присутні як один об'єкт, так і декілька. Для оцінки алгоритмів машинного навчання зазвичай використовуються анотовані бази зображень, наприклад, CIFAR-10 та ImageNet,. Через те, що на зображеннях в базі ImageNet може бути присутнім декілька об'єктів, і лише один з них анотований, в ImageNet основною оцінкою помилки є top-5 помилка. При її використанні вважається, що алгоритм не помилився, якщо правильна категорія об'єкта знаходиться серед п'яти категорій, виданих алгоритмом як найбільш ймовірні. Внаслідок цього нейронні мережі для задачі класифікації оцінюються саме за допомогою top-5 помилки.

## **1.2 Згорткові нейронні мережі**

Нейронна мережа – це обчислювальні системи, створені вирішувати певні задачі, наприклад, класифікація чи сегментація. Нейрони збираються по шарам. Існує вхідний шар, на який подається вхідний сигнал (вхідне зображення), також є вихідний шар, з якого знімається результат роботи нейронної мережі, а також між ними знаходяться приховані шари.

Розглянемо одну з коротких нейронних мереж, CNN (Convolutional Neural Network) - це нейронна мережа, яку використовують як різновид багат шарових перцептронів, розроблений так, щоби вимагати використання мінімального обсягу попередньої обробки. Це досягається, замість використання повністю з'єднаних шарів, за допомогою згорткових

шарів, в яких до зображення застосовується операція згортки із серією фільтрів. Ці згорткові шари можна вивчати і з ними можна працювати.

Для згорткових шарів всі нейрони змушені використовувати ті ж ваги і підключатися лише до невеликої частини зображення (наприклад, 5x5 пікселів). Крім того, вони організовані в просторовій сітці по всьому зображенню. Через це розташування, вихід цієї сітки нейронів буде згорткою вибраних ваг з вхідним зображенням. Отже, ці вибрані ваги можна розглядати як фільтр, який згортається з зображенням, генеруючи вихід.

На кожному згортковому шарі, як правило, декілька фільтрів, що вивчаються, мають однаковий просторовий розмір, і згортання зображення з ними створює декілька вихідних зображень, де кожна з них позначається як “карта властивостей” згідно з рис. 1.2.

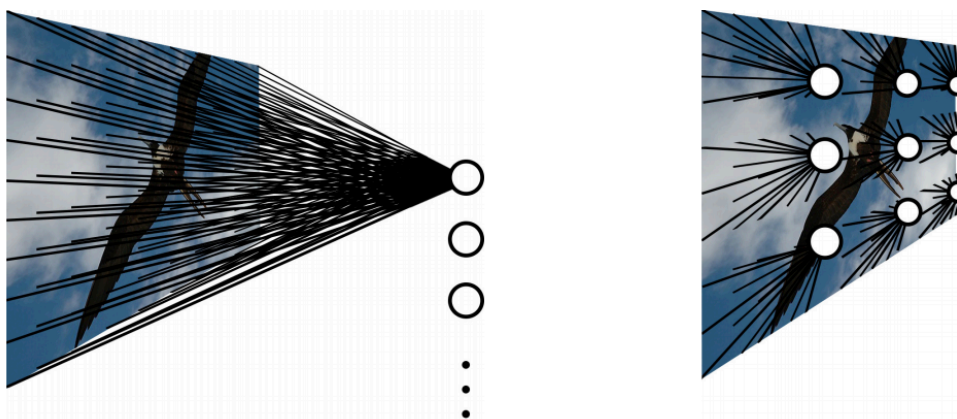


Рисунок 1.2 – Різниця між звичайними та згортковими шарами

Ці кілька функціональних карт потім складаються в 3D об'єкт даних з розмірами  $W \times H \times N$ , де  $W$  та  $H$  позначають просторові розміри даних, а  $N$  - кількість фільтрів, що використовуються в шарі. Цей об'єкт 3D-даних потім може згортатися з новим набором фільтрів, створюючи нові виходи.

Щоб швидко вказати розмір згорткового шару, використовується позначення  $W \times H \times D \times N$ , де  $W$  та  $H$  задають просторовий розмір використовуваного фільтра (тобто просторова протяжність в пікселях області, до якої кожен нейрон згорткового шару буде з'єднаний),  $D$  вказує, до скількох карт ознак буде з'єднуватися нейрон, а  $N$  позначає кількість навчальних фільтрів, які захоплюються цим конкретним згортковим шаром.

Крім згорткових шарів, згорткові нейронні мережі також зазвичай використовують два інших типи шарів: шари ReLU і Max-Pool шари. Шари ReLU (Ректифіковані лінійні одиничні шари) обчислюють вихід для кожного пікселя, як:

$$ReLU(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}, \quad (1.1)$$

де  $x$  – значення на вході шару.

Один із способів інтерпретації каскаду згорткових шарів - це функція екстрактора, яка після деяких шарів виробляє набагато менший об'єкт даних, ніж вхідний образ, але ми сподіваємося, що він все ще має такі ж значущі характеристики для вирішення поставленої задачі. Потім ця мережа тренується з використанням алгоритму зворотної помилки, який використовується для тренувань звичайних нейронних мереж.

Звичайно неможливо зберігати всі тренувальні приклади в пам'яті машини, тому для оптимізації ваг часто вибирається стохастичний градієнтний спуск, що обчислювально ефективніше простого градієнтного спуску, а також дозволяє швидше отримати правильне рішення.

### 1.2.1 Основні шари в згорткових мережах

Згортковий шар дозволяє об'єднувати значення розташованих поруч пікселів і виділяти більше узагальнені ознаки зображення. Для цього по

картинці послідовно ковзають квадратним вікном невеликого розміру (3x3, 5x5, 7x7 пікселів і т.д.) що називають ядром згортки (див. рис. 1.3). Кожен елемент ядра має свій ваговий коефіцієнт, який множиться на значення пікселя зображення, на який в даний момент накладено елемент ядра. Потім отримані для всього вікна числа складаються, і ця зважена сума дає значення чергового ознаки.

Для отримання матриці (“карти”) ознак всього зображення, ядро послідовно зсувається по горизонталі і вертикалі. У наступних шарах операція згортки застосовується вже до карт ознак, отриманим з попередніх шарів. Графічно процес проілюстровано на рисунку 1.3.

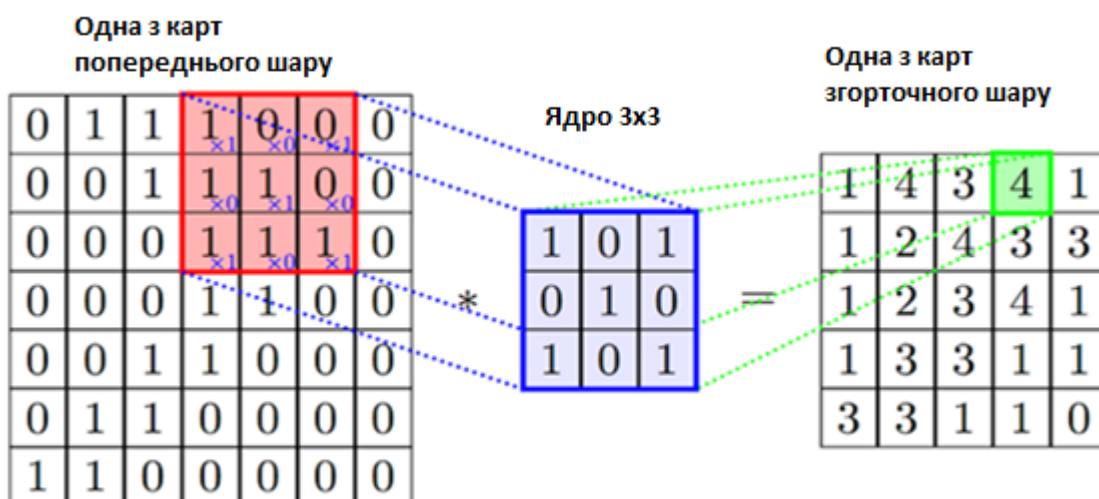


Рисунок 1.3 – Отримання матриці ознак

Зображення або карти ознак в рамках одного шару можуть скануватися не одним, а кількома незалежними фільтрами, даючи таким чином на вихід не одну карту, а кілька (їх ще називають “каналами”). Налаштування ваг кожного фільтра відбувається за допомогою методу зворотного поширення помилки.



Очевидно, якщо ядро фільтра при скануванні не виходить за межі зображення, розмірність карти ознак буде менше, ніж у вихідної картинки. Якщо потрібно зберегти той же розмір, застосовують так звані padding – значення (див. рис. 1.4), якими доповнюється зображення по краях і які потім захоплюються фільтром разом з реальними пікселями картинки.

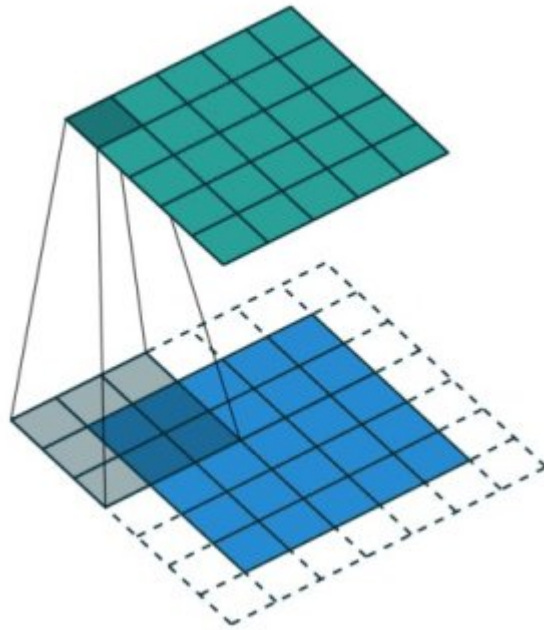


Рисунок 1.4 – Приклад покривання картинки ядром фільтра з доповненням paddings значеннями

Крім paddings на зміну розмірності так само впливають strides - значення кроку, з яким вікно переміщається по зображенню чи карті.

Згортка не є єдиним способом отримання узагальненої характеристики групи пікселів. Найпростіший спосіб для цього – вибрати один піксель по заданому правилу, наприклад – максимальний. Саме це і робить шар MaxPooling (див. рис. 1.5) .

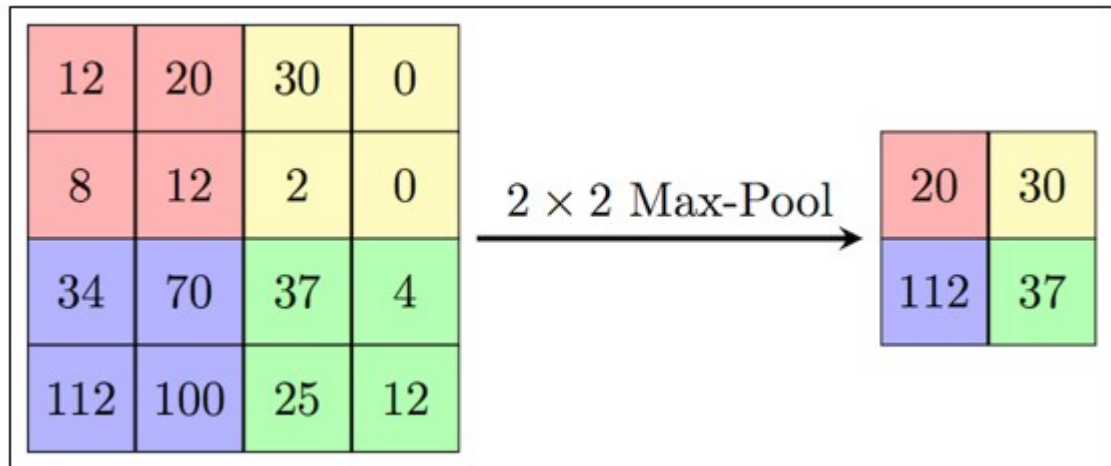


Рисунок 1.5 – Приклад роботи шару MaxPooling

На відміну від convolution, maxpooling зазвичай застосовується до непересічних груп пікселів.

### 1.3 Згорткові нейронні мережі для класифікації зображень створені людьми

В даному розділі розглянуто задачу класифікації зображень і дано короткий опис структури згорткових нейронних мереж. Проводиться огляд згорткових нейронних мереж створених людьми для виконання класифікацій зображень і зроблено порівняння їх чіткості на прикладі анотованих баз зображень ImageNet.

#### 1.3.1 AlexNet

У 2012 році на конкурсі ILSVRC за класифікацією зображень вперше перемогла нейронна мережа - AlexNet [5], досягнувши top-5 помилки 15,31%. Її архітектура зображена на рисунку 1.6.

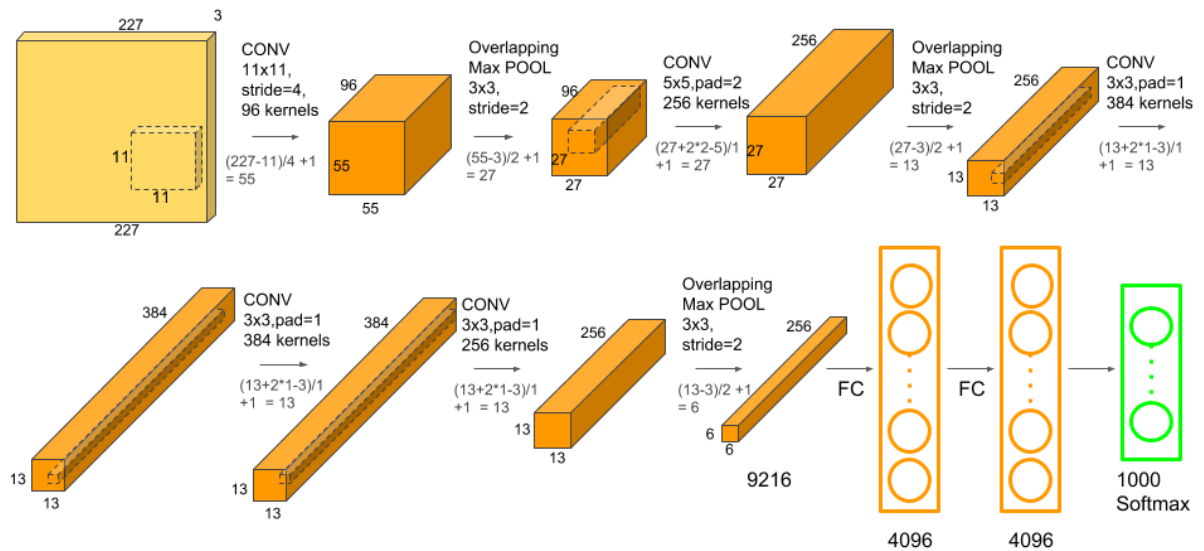


Рисунок 1.6 – Архітектура AlexNet

Для порівняння, метод, який не використовує згорткові нейронні мережі, отримав помилку 26,1%. У AlexNet були зібрані новітні на той момент техніки для поліпшення роботи мережі. Навчання AlexNet через кількість параметрів мережі відбувалося на двох GPU, що дозволило скоротити час навчання в порівнянні з навчанням на CPU. Також виявилось, що використання функції активації ReLU замість більш традиційних функцій сигмоид і гіперболічного тангенса дозволило знизити кількість епох навчання в шість разів. Формула ReLU дозволяє побороти проблему загасання градієнтів, властиву інших функцій активації. Крім іншого, в AlexNet була застосована техніка відсіву Dropout [6]. Вона полягає у випадковому відключенні кожного нейрона на заданому шарі з ймовірністю  $p$  на кожній епосі. Після навчання мережі, на стадії розпізнавання, ваги шарів, до яких був застосований dropout, повинні бути помножені на  $1 / p$ . Dropout виступає в ролі регуляризатора, не дозволяючи мережі перенавчатися. Для пояснення ефективності даної техніки існує кілька інтерпретацій. Перша полягає в тому, що dropout змушує нейрони

не покладатися на сусідні нейрони, а навчатися розпізнавати більш стійкі ознаки. Друга, більш пізня, полягає в тому, що, навчання мережі з dropout є апроксимацією навчання ансамблю мереж, кожна з яких представляє мережу без деяких нейронів. Таким чином, остаточне рішення приймає не одна мережа, а ансамбль, кожна мережа якого навчена по-різному, тим самим знижується ймовірність помилки.

### 1.3.2 ZF Net

ZF Net - переможець ILSVRC 2013 з top-5 помилкою 11,2%. Основним досягненням даної архітектури є створення техніки візуалізації фільтрів - мережі розгортки (deconvolutional network), що складається з операцій, в якомусь сенсі зворотних операцій мережі, як показано на рисунку 1.7.

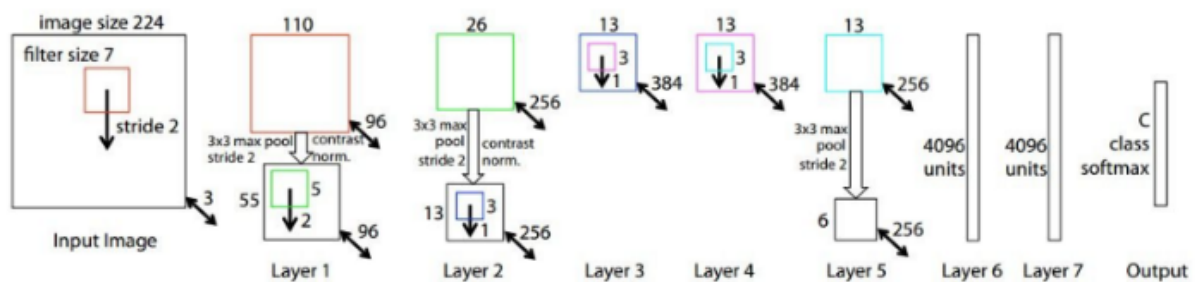


Рисунок 1.7 – ZF Net архітектура

В результаті мережа розгортки відображає прихований шар мережі на оригінальне зображення. Щоб вивчити поведінку фільтра на певному зображенні за допомогою навченої нейронної мережі, необхідно спочатку здійснити висновок мережею, після чого в шарі досліджуваного фільтра обнулити всі ваги, крім терезів самого фільтра, і потім подати отриману активацію на шар мережі розгортки. У мережі розгортки послідовно застосовуються операції Unpooling, ReLU і фільтрації. Unpooling частково відновлює вхід відповідного шару субдискретизація, запам'ятовуючи

координати, які вибрав шар субдискретизації. ReLU - звичайний шар, що застосовує функцію ReLU. Шар фільтрації виконує операцію згортки з вагами відповідного шару згортки, але ваги кожного фільтра «перевернуті» вертикально і горизонтально. Таким чином, вихідна активація фільтра рухається в зворотному напрямку, поки не буде відображена в оригінальному просторі зображення.

### 1.3.3 VGG Net

VGG Net - модель згорткової нейронної мережі, запропонована в роботі Simonyan та Zisserman [7], наведена на рис. 1.8.

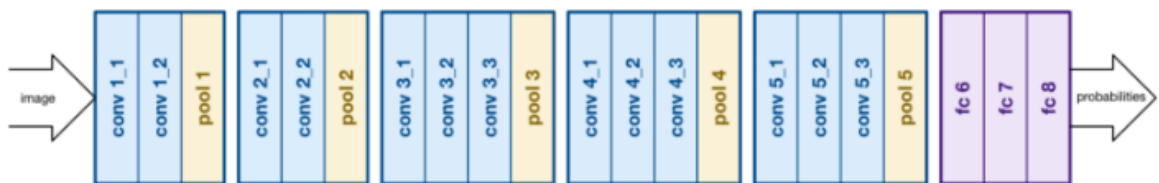


Рисунок 1.8 – VGG Net архітектура

В даній мережі відмовилися від використання фільтрів розміром більше, ніж 3x3. Автори показали, що шар з фільтром 7x7 еквівалентний трьом шарам з фільтрами 3x3, причому в останньому випадку використовується на 55% менше параметрів. Аналогічно шар з фільтром 5x5 еквівалентний двом шарам з фільтром 3x3, які економлять 22% параметрів мережі. Візуальне подання такої декомпозиції можна побачити на рисунку 1.9

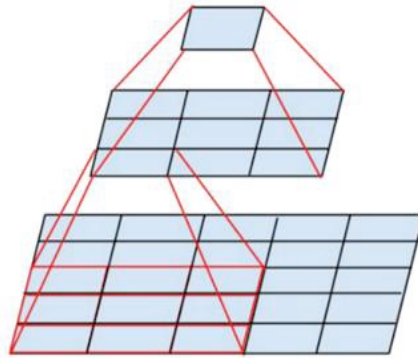


Рисунок 1.9 – декомпозиція фільтра 5x5

Декомпозиція фільтра 5x5 на змаганні ILSVRC 2014 ансамбль з двох VGG Net отримав top-5 помилку 7,3%. Хоча дана модель і не перемогла в змаганні, через її простоти вона використовується в більш складних мережах, призначених для детектування предметів, семантичної сегментації або маскування об'єктів.

### 1.3.4 Inception

Нейронна мережа Inception-v1 - переможець ILSVRC 2014 за top-5 помилкою 6,7%, також відомий як GoogLeNet [8], наведена на рис 1.10.

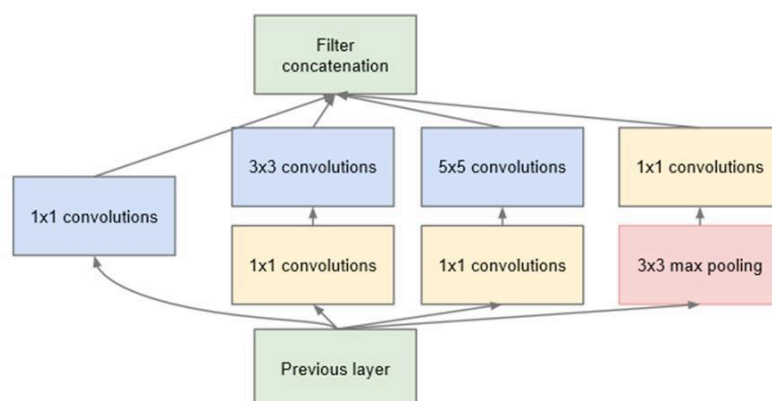


Рисунок 1.10 – Модуль Inception

Творці цієї мережі виходили з факту, що після кожного шару мережі необхідно зробити вибір - чи буде наступний шар сверткой з фільтром  $3 \times 3$ ,  $5 \times 5$ ,  $1 \times 1$  або ж шаром субдискретизації. Кожен з таких шарів корисний - фільтр  $1 \times 1$  виявляє кореляцію між каналами, в той час як фільтри більшого розміру реагують на більш глобальні ознаки, а шар субдискретизації дозволяє зменшити розмірність без великих втрат інформації. Замість того щоб вибирати, який саме шар повинен бути наступним, пропонується використовувати всі шари відразу, паралельно один одному, а потім об'єднати отримані результати в один. Щоб уникнути зростання числа параметрів, перед кожним шаром згортки використовується згортка  $1 \times 1$ , яка зменшує число карт ознак. Такий блок шарів назвали модулем Inception (Рисунок 1.10). Також в GoogLeNet відмовилися від використання повнозв'язну шару в кінці мережі, використовуючи замість нього шар Average Pooling (усереднення), завдяки чому різко зменшилася кількість параметрів в мережі. Таким чином, GoogLeNet, що складається з більш ніж ста базових шарів, має майже в 12 разів менше параметрів, ніж AlexNet (близько семи мільйонів параметрів проти 138 мільйонів).

### 1.3.5 ResNet

Переможцем ILSVRC 2015 з top-5 помилкою в 3,57% став ансамбль з шести мереж типу ResNet (Residual Network), розроблений в Microsoft Research. Автори ResNet помітили, що з підвищенням числа шарів згорткова нейронна мережа може почати деградувати - у неї знижується точність на валідаційних даних. Так як падає точність і на тренувальних даних, можна зробити висновок, що проблема полягає не в перенавчанні мережі. Було зроблено припущення, що якщо згорткова нейронна мережа досягла своєї межі точності на деякому шарі, то всі такі шари повинні будуть виродитися в тотожне перетворення, але через складність навчання глибоких мереж цього не відбувається. Для того щоб «допомогти» мережі,

було запропоновано ввести пропускі з'єднання (Shortcut Connections), зображені на рис. 1.11.

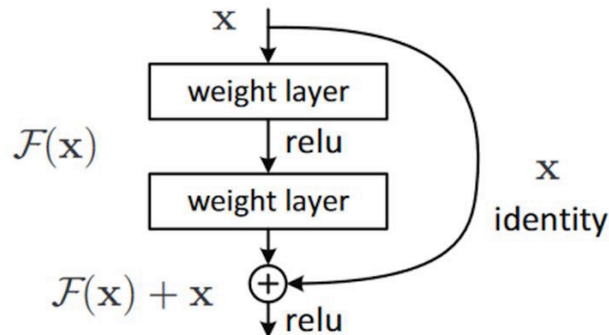


Рисунок 1.11 – Пропускне з'єднання

Нехай оригінальна мережа повинна обчислювати функцію  $H(X)$ . Визначимо її залишкову функцію як  $F(X) = H(x) - x$ , яка, в теорії, повинна простіше навчатися мережею. Додавши пропускі з'єднання, як показано на рисунку 1.11, мережа вчиться залишкової функції, яка потім складається з тотожним перетворенням.

Аналіз показав, що глибокі залишкові нейронні мережі можна вважати ансамблем, що складається з більш дрібних залишкових нейронних мереж, чия ефективна глибина збільшується в процесі навчання.

### 1.3.6 Inception-v4 та Inception-ResNet

Після успіху ResNet, в роботі Szegedy та Ioffe [16] були представлені наступні версії мережі Inception: Inception-v4 і Inception - ResNet. В обох варіантах модуль Inception був розбитий на модулі A, B і C для входів розмірністю  $35 \times 35$ ,  $17 \times 17$  і  $8 \times 8$  відповідно. Також були виділені блоки редукції, в яких відбувається зниження розмірності і збільшення глибини даних усередині мережі. У Inception-v4 головними нововведеннями є заміна Max Pooling на Average Pooling в самих модулях Inception. Для



Inception-ResNet в модулі Inception були додані пропускні з'єднання. Були сконструйовані дві версії мережі - Inception-ResNet-v1, для якої потрібно менше обчислень, і Inception-ResNet-v2. Як приклад на рисунку 1.12 представлений модуль Inception-ResNet-C для Inception-ResNet-v1.

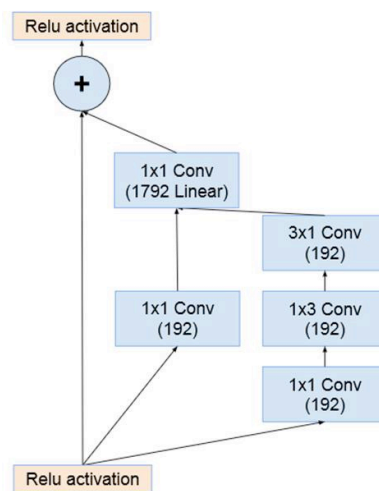


Рисунок 1.12 – Модуль Inception-ResNet-C для Inception-ResNet-v1

#### 1.4 Постановка задачі

Метою даної роботи є огляд та реалізація методів автоматичного побудування згорткових нейронних мереж для вирішення задачі класифікації.

Робота включає наступні етапи:

- 1) огляд та аналіз вже існуючих сучасних моделей згорткових нейронних мереж, створених людьми;
- 2) огляд методів для автоматичного побудування згорткових мереж;
- 3) огляд методу DARTS
- 4) побудова моделі на даних CIFAR-10;
- 5) аналіз отриманого результату.

### **1.5 Висновки до розділу**

У даному розділі дипломної роботи було оглянуто загальний принцип роботи згорткових мереж та розібрано деякі нюанси реалізації. Також були розглянуті архітектури сучасних моделей згорткових мереж створених людьми та проведено їх аналіз.

## **2 ПОШУК АРХІТЕКТУР НЕЙРОНИХ МЕРЕЖ**

Пошук архітектур нейронних мереж це спроба об'єднати всі блоки нейронних мереж найкращим чином, щоб отримати найвищу продуктивність в одній задачі, скажімо у категоризації зображень ImageNet. Всі архітектури були створені людьми за допомогою інтуїції та логіки. Але, як ви можете собі уявити, це не завжди може привести до найкращих рішень. Це не шлях до прогресу, оскільки ми будемо обмежені тим людськими когнітивними здібностями і ніколи не будемо шукати серед повного простору всіх можливих нейронних архітектур. Нейромережеві алгоритми будуть вигравати у людини в області пошуку нейронної архітектури, так само, як вони виграють у них і в шахи! Отже, шлях до прогресу - це використання нейронних мереж для пошуку кращих нейронних архітектур. Фактично ми будемо використовувати ті ж самі методи градієнтного спуску, щоб керувати величезним пошуком архітектури нейронної мережі. Але, як ви можете бачити, існує багато можливих нейронних будівельних блоків, а простір пошуку величезний. Уявіть собі, що ми спробуємо всі можливі згорткові шари: різне число входів виходів площин, розширення, глибинність, об'єднання, нелінійність і т.д. Це неймовірно величезний простір, на обробку якого не вистачить часу. Але ж існують підходи, наприклад жадібні алгоритми, які дозволяють досягати непоганих результатів.

### **2.1 Пошук нейронної архітектура за допомогою навчання з підкріпленням**

У цьому розділі представлено метод пошуку нейронних архітектур, що оснований на роботі Barret Zoph [13], у якому пошук хороших архітектур здійснюється на основі градієнту (Рисунок 2.1).

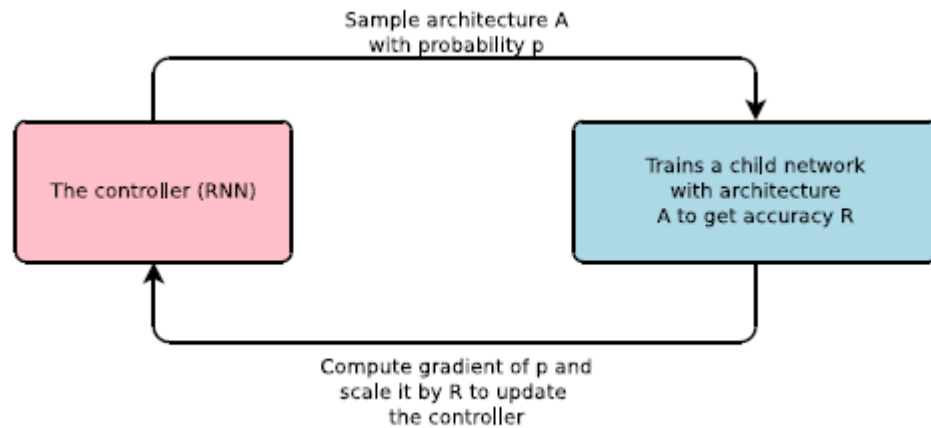


Рисунок 2.1 – Огляд пошуку нейронної архітектури

Ця робота ґрунтується на спостереженні, що структуру та зв'язок нейронної мережі зазвичай можна визначити за допомогою рядка змінної довжини. Тому для створення такого рядка можна використовувати рекурентну мережу – контролер. Навчання мережі, визначеної рядком - "дитяча мережа" – на реальних даних призведе до високої точності на набору перевірки. Використовуючи цю точність як сигнал винагороди, ми можемо обчислити градієнт політики для оновлення контролера. В результаті, в наступній ітерації, контролер надасть більш великі ймовірності архітектурам, які отримують високу точність. Іншими словами, контролер навчиться вдосконалювати свій пошук з часом. Експерименти показують, що пошук нейронної архітектури може створювати хороші моделі з нуля, досягнення, яке вважається неможливим з іншими методами. При розпізнаванні зображень за допомогою CIFAR-10 пошук нейронної архітектури може знайти нову модель ConvNet, яка є кращою, ніж більшість архітектур, винайдених людиною. Поточна модель на даних CIFAR-10 досягає помилки на тестовому наборі даних на рівні 3,65, що на 1.05x краще, ніж поточна найкраща модель.

### 2.1.1 Навчання з підкріпленням

Список параметрів (наприклад, ширини, висоти згортки, шагу і т.д.), які передбачає контролер, може розглядатися як список дій  $a_{1:T}$  для розробки архітектури дочірньої мережі. При збіжності ця дочірня мережа досягатиме точність  $R$  на тренувальному наборі даних. Ми можемо використовувати цю точність  $R$  як сигнал винагороди і використовувати навчання з підкріпленням для підготовки контролера. На рис 2.2 зображено приклад роботи контролера.

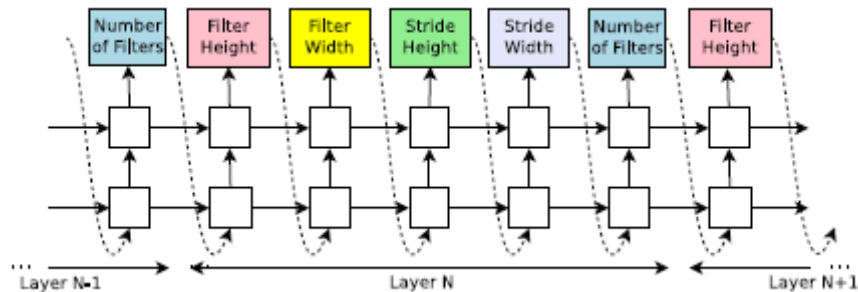


Рисунок 2.2 — Як контролер рекурентних нейронних мереж будує просту згорткову мережу.

Більш конкретно, щоб знайти оптимальну архітектуру, ми просимо нашого контролера максимізувати свою очікувану винагороду, представлену  $J(\theta_c)$ :

$$J(\theta_c) = E_{P_{a(1:T; \theta_c)}}[R], \quad (2.1)$$

де  $\theta_c$  – вектор параметрів передбачених контролером,

$p$  – ймовірність архітектури  $a$ .

Оскільки сигнал винагороди  $R$  не є диференційованим, нам потрібно використовувати метод градієнта політики для ітеративного оновлення  $\theta_c$ .

У роботі Barret Zoph [13], використовується правило REINFORCE від Williams (1992) [10]:

$$\nabla_{\theta_c} J(\theta_c) = \sum_{t=1}^T E_{p(a_{1:T}; \theta_c)} [\nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c) R], \quad (2.2)$$

Емпіричне наближення вищевказаної кількості (2.2):

$$\frac{1}{m} \sum_{k=1}^m \sum_{t=1}^T \log P(a_t | a_{(t-1):1}; \theta_c) R_k, \quad (2.3)$$

де  $m$  - це кількість різних архітектур, які контролер відбирає в одній партії,  $T$  - кількість гіперпараметрів, що наш контролер повинен передбачити, щоб створити архітектуру нейронної мережі.

Точність перевірки, що досягається  $k$ -ою архітектурою нейронної мережі після навчання на навчальному наборі даних, є  $R_k$ . Вищезгадане оновлення (2.3) є об'єктивною оцінкою для нашого градієнта, але має дуже високу дисперсію. Для зменшення дисперсії цієї оцінки ми використовуємо функцію базової лінії (2.4):

$$\frac{1}{m} \sum_{k=1}^m \sum_{t=1}^T \nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c) (R_k - b), \quad (2.4)$$

До тих пір, поки функція  $b$  базової лінії (2.4) не залежить від поточної дії, то це все ще є неупередженою оцінкою градієнта. У цій роботі наша базова лінія  $b$  експоненціально двигляє середню точність попередньої архітектури.

Прискорення навчання з паралелізмом та асинхронними оновленнями: у нейромережевому пошуку кожне оновлення градієнта до параметрів контролера  $c$  відповідає підготовці однієї дочірньої мережі до збіжності. Оскільки навчання дитячої мережі може тривати кілька годин,

ми використовуємо розподілене навчання та асинхронні оновлення параметрів для того, щоб прискорити процес навчання контролера. Ми використовуємо схему параметр-сервер, де є сервер параметрів  $S$ , який зберігає спільні параметри для реплік  $K$ -контролерів. Кожен контролер має репліки зразків  $m$  різних дочірніх архітектур, які навчаються паралельно. Потім контролер збирає градієнти за результатами конкретної мініатюри  $m$  архітектур при збіжності і відправляє їх на сервер параметрів для того, щоб оновити ваги всіх реплік контролера. У реалізації, збіжність кожної дитячої мережі досягається тоді, коли її підготовка перевищує певну кількість епох. Ця схема паралелізму узагальнена на рисунку 2.3:

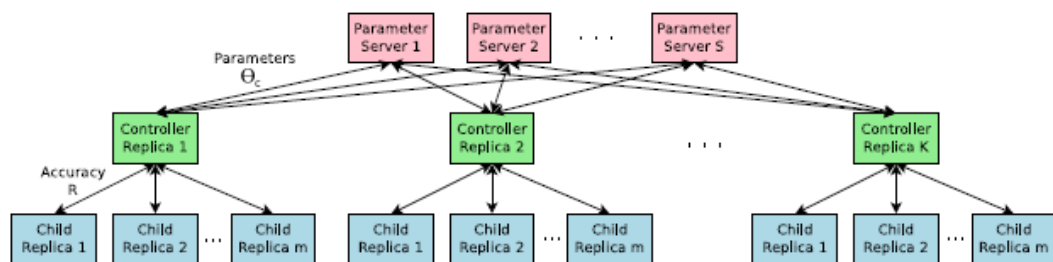


Рисунок 2.3 — Розподілене навчання для пошуку нейронної архітектури

### 2.1.2 Навчання згорткової мережи на CIFAR-10

Після того, як контролер тренує 12 800 архітектур, вибирається архітектура, яка досягає найкращої точності. Найкраща модель з цього пошуку тренується до збіжності, і тоді ми обчислюємо точність такої моделі і підсумовуємо результати.

По-перше, якщо попросити контролера не прогнозувати кроки або об'єднання, це може створити 15-шарову архітектуру, яка досягає 5,50% помилок на тестовому наборі. Ця архітектура має гарний баланс між точністю і глибиною. Насправді, це найдрібніша і, можливо, найбільш недорога архітектура серед найбільш ефективних мереж. Ця архітектура

показана на рисунку 2.4, де  $FH$  - висота фільтра,  $FW$  - ширина фільтра, а  $N$  - кількість фільтрів. Зверніть увагу, що пропущені з'єднання не є залишковими з'єднаннями. Якщо один шар має багато вхідних шарів, то всі вхідні шари об'єднані в одну розмірність глибини.





Рисунок 2.4 - Згорткова архітектура, виявлена даним методом, коли простір пошуку не має кроків або шарів об'єднання.

Примітною особливістю цієї архітектури є те, що вона має багато прямокутних фільтрів і має більші фільтри у верхніх шарах ніж у середині. Подібно до залишкових мереж, архітектура також має багато одностадійних пропусків. Ця архітектура є локальним оптимумом в тому сенсі, що якщо ми її змінимо, її продуктивність стане гірше. Наприклад, якщо ми щільно з'єднаємо всі шари з пропусками підключень, її продуктивність стає 5,56%. Якщо видалити всі з'єднання, які пропускають обчислення, її продуктивність падає до 7,97%. Додатково ця модель з 40 доданими фільтрами дорівнює 1.05x від точності моделі DenseNet, яка досягає 3.74%, маючи кращу продуктивність. Модель DenseNet, яка досягає частоти помилок 3,46%, використовує згортання 1x1 для зменшення загальної кількості параметрів, що у цьому алгоритмі не робиться, тому це не є точним порівнянням.

## **2.2 Регуляризована еволюція для пошуку архітектур класифікатора зображень**

### **2.2.1 Пошуковий простір**

Всі експерименти використовують пошуковий простір NASNet [11]. Це простір класифікаторів зображень, всі з яких мають фіксовану зовнішню структуру, позначену на рисунку 2.5 (ліворуч): передавальний стек Inception-подібних модулів, що називаються клітинами. Кожна комірка отримує прямий вхід від попередньої комірки (як зображено) і пропускає вхід з осередку перед ним (мал. 2.5, середина). Клітини в стеку бувають двох типів: нормальна клітина і редукційна клітина. Всі нормальні клітини обмежені тим, що мають ту ж саму архітектуру, що і редукційних клітини. Єдина відмінність між ними полягає в тому, що за кожним застосуванням редукційної клітини слідує крок, який зменшує розмір зображення, тоді як нормальні клітини зберігають розмір зображення. Як видно на рисунку, нормальні клітини розташовуються в трьох стеках N

клітин. Мета архітектурно-пошукового процесу полягає у виявленні архітектур нормальних і редукційних клітин.

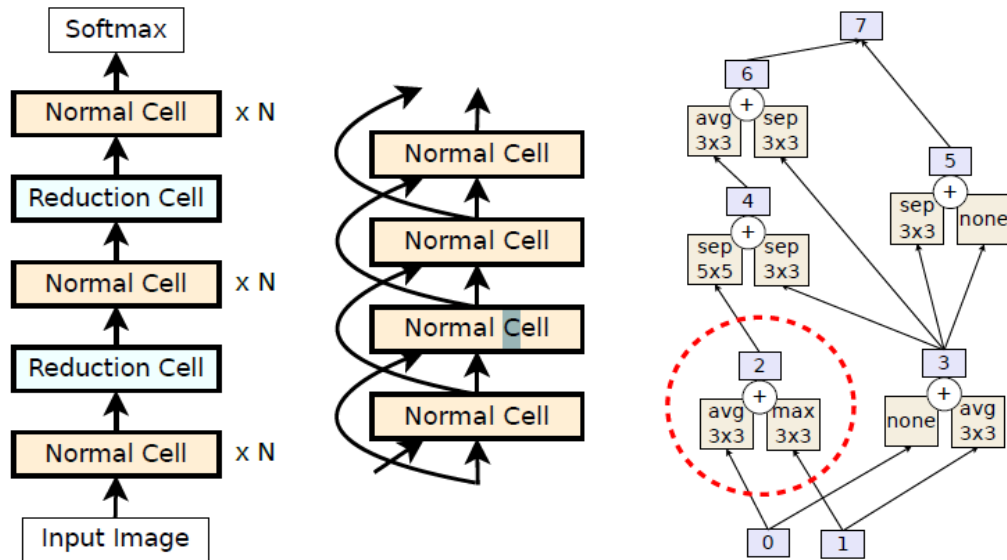


Рисунок 2.5 – NasNet пошуковий простір [11]

Як зображено на рисунку 2.5 (середній і правий), кожна клітина має два вхідні тензори активації і один вихід. Сама перша комірка приймає дві копії вхідного зображення. Після цього входи є виходами двох попередніх клітин. І нормальні, і редукційні елементи повинні відповідати наступній конструкції. Два тензори входу вважаються прихованими станами «0» і «1». Інці приховані стани потім будуються через попарні комбінації. Попарна комбінація зображена на рисунку 2.5 (праворуч, усередині пунктирного кола). Вона полягає в застосуванні операції до існуючого прихованого стану, застосуванню іншого оператора до іншого існуючого прихованого стану і додавання результатів для створення нового прихованого стану. Операції належать до фіксованого набору загальних операцій згорткових мереж, таких як згортки і шари об'єднання. Дозволено повторювати приховані стани або операції в комбінації. У прикладі на рисунку 2.5 (праворуч) перша попарна комбінація застосовує 3x3 average

pool до прихованого стану 0 і  $3 \times 3$  max pool до прихованого стану 1, щоб створити прихований стан 2. Наступна попарна комбінація може тепер вибирати з прихованих станів 0, 1 і 2 для створення прихованого стану 3 (вибрали 0 і 1 на рисунку 2.7) і так далі. Після рівно п'яти попарних комбінацій будь-які приховані стани, які залишаються невикористаними (приховані стани 5 і 6 на рисунку 2.5), об'єднуються для формування виходу комірки (прихованого стану 7). Дана архітектура повністю визначена п'ятьма попарними комбінаціями, які складають нормальну клітинку, і п'ять, які складають редукційну клітину. Як тільки архітектура вказана, модель все ще має два вільні параметри, які можуть бути використані для зміни її розміру і її точності: кількість нормальних клітин на стек (N) і кількість вихідних фільтрів операцій згортки (F). N і F визначаються вручну.

### **2.2.2 Еволюційний алгоритм**

У даному розділі описується еволюційний алгоритм Esteban Real [2]. Він зберігає популяцію P підготовлених моделей протягом усього експерименту. Популяція ініціалізується моделями з випадковими архітектурами. Всі архітектури, які відповідають описуваному простору пошуку, є можливими і однаково ймовірними.

---

**Algorithm 1** Aging Evolution
 

---

```

population  $\leftarrow$  empty queue ▷ The population.
history  $\leftarrow \emptyset$  ▷ Will contain all models.
while  $|population| < P$  do ▷ Initialize population.
    model.arch  $\leftarrow$  RANDOMARCHITECTURE()
    model.accuracy  $\leftarrow$  TRAINANDEVAL(model.arch)
    add model to right of population
    add model to history
end while
while  $|history| < C$  do ▷ Evolve for  $C$  cycles.
    sample  $\leftarrow \emptyset$  ▷ Parent candidates.
    while  $|sample| < S$  do
        candidate  $\leftarrow$  random element from population
        ▷ The element stays in the population.
        add candidate to sample
    end while
    parent  $\leftarrow$  highest-accuracy model in sample
    child.arch  $\leftarrow$  MUTATE(parent.arch)
    child.accuracy  $\leftarrow$  TRAINANDEVAL(child.arch)
    add child to right of population
    add child to history
    remove dead from left of population ▷ Oldest.
    discard dead
end while
return highest-accuracy model in history
  
```

---

Рисунок 2.6 – Еволюційний алгоритм

Після цього еволюція покращує початкову популяцію в циклах. На кожному циклі зразки  $S$  випадкових моделей з популяції, кожна з яких витягнута рівномірно випадковим чином з заміною. Модель з найбільшою точністю в межах цього зразка вибирається як батьківська. Нова архітектура, звана дитиною, будується з батьків шляхом застосування трансформації, званої мутацією. Мутація викликає просту і випадкову модифікацію архітектури і докладно описано нижче. Після побудови дитячої архітектури вона потім навчається, оцінюється і додається до населення. Цей процес називається турнірним відбором. У виборі турніру часто зустрічається фіксація розміру популяції на початковому значенні  $P$ .

Це часто досягається додатковим кроком у кожному циклі: відкидання (або вбивство) найгіршої моделі у випадковій  $S$ -вибірці. На відміну від цього, у цій роботі віддається перевага новому підходу: вбиваючи найдавнішу модель в популяції, тобто вилучаючи з популяції модель, яку тренували найдавніше. Це сприяє новим моделям населення. Цей підхід будемо називати еволюцією старіння. У контексті пошуку архітектури еволюція старіння дозволяє нам більше досліджувати пошуковий простір, а не надто рано збільшувати належні моделі. Інтуїтивно, мутації можна розглядати як забезпечення дослідження, а вибір батьків забезпечує експлуатацію. Параметр  $S$  контролює агресивність експлуатації:  $S = 1$  зводиться до типу випадкового пошуку і  $2 < S < P$  призводить до еволюції різної жадібності. Нові моделі будуються шляхом застосування мутації до існуючих моделей, перетворюючи їхні архітектури випадковими способами. Для переміщення по простору пошуку описаному вище алгоритму навчання з підкріпленням, ми використовуємо дві основні мутації, які ми називаємо мутацією прихованого стану і мутацією операцій. Третя мутація, ідентичність, також можлива. Тільки одна з цих мутацій застосовується в кожному циклі, вибираючи між ними навмання.

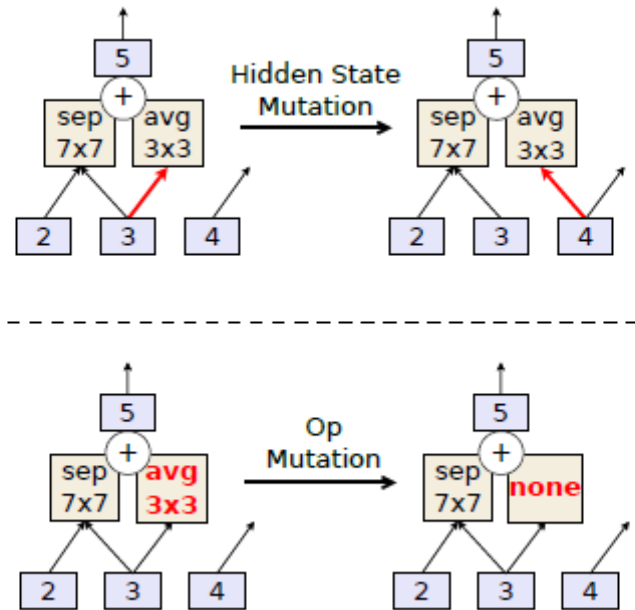


Рисунок 2.7 – Зображення двох видів мутації

Прихована мутація стану полягає в тому, що спочатку робиться випадковий вибір того, чи потрібно змінювати нормальну клітинку або відновну клітинку. Як тільки клітина обрана, мутація вибирає одну з п'яти попарних комбінацій рівномірно навмання. Як тільки попарна комбінація вибрана, один з двох елементів пари обирається рівномірно випадковим чином. Обраний елемент має один прихований стан. Цей прихований стан тепер замінений на інший прихований стан всередині клітини, за умови, що не утворюються петлі (щоб зберегти передавальний характер вузла). На рисунку 2.7 (зверху) показаний приклад. Мутація Op веде себе як прихована мутація стану до вибору однієї з двох осередків, однієї з п'яти попарних комбінацій, і одного з двох елементів пари. Тоді він відрізняється тим, що він змінює Op, а не прихований стан. Він виконує це, замінюючи існуючий оператор з випадковим вибором з фіксованого списку операцій. На рисунку 2.7 (внизу) показаний приклад.

Популярний підхід до еволюції пройшов через генераційні алгоритми. Всі моделі в популяції повинні завершити підготовку до того,

як буде розраховано наступне покоління. Еволюція поколінь стає неефективною в розподіленому середовищі, де для тренування кожної моделі використовується інша машина: машини, які тренують швидші моделі, закінчують раніше і повинні чекати простою, поки всі машини не будуть готові. Алгоритми реального часу вирішують цю проблему. Однак, на відміну від генераційних алгоритмів, ці моделі відкидають моделі відповідно до їхньої продуктивності або взагалі не відкидають їх, внаслідок чого моделі залишаються живими в популяції протягом тривалого часу - навіть для всього експерименту.

Наведемо докази того, що кінцеві терміни еволюції старіння можуть дати кращі результати, ніж прямий вибір турніру, зберігаючи при цьому свою ефективність. У цьому методі вік призначається генам, щоб розділити популяцію постійного розміру на групи, що називаються віковими шарами. Кожен шар містить осіб з генами подібного віку. Тільки після того, як гени пережили певний віковий розрив, вони можуть перейти до наступного шару. Мета полягає в тому, щоб обмежити конкуренцію (нові введені гени не можуть бути негайно поза конкуренцією з боку високо-відібраних старших). Їх алгоритм вимагає запровадження двох додаткових мета-параметрів (розмір вікового розриву і кількість вершин). На відміну від цього, в алгоритмі, вік призначається індивідам (а не генам) і використовується тільки для відстеження найстарішої особини в популяції. Це дозволяє видаляти таких найстаріших осіб на кожному циклі (зберігаючи постійний розмір населення). Зокрема, цей метод залишається схожим на природу (де молоді люди рідше помирають, ніж самі старі) і не вимагає додаткових мета-параметрів.

### **2.2.3 Результати**

Надані результати узяті із дослідження Esteban Real [2]. В даний час навчання з підкріпленням (RL) є переважним методом пошуку



архітектури. Тут буде порівняно еволюційний підхід з алгоритмом RL. Проведено широкомасштабні експерименти з пошуку архітектури на CIFAR-10. Рисунок 2.8 показує точність моделі, як експерименти прогресу, підкреслюючи, що еволюція дала більш точні моделі на ранніх стадіях, які могли б стати важливими в режимі, обмеженому ресурсами, де експерименти повинні бути зупинені рано (наприклад, коли 450 GPU для 7 днів занадто багато).

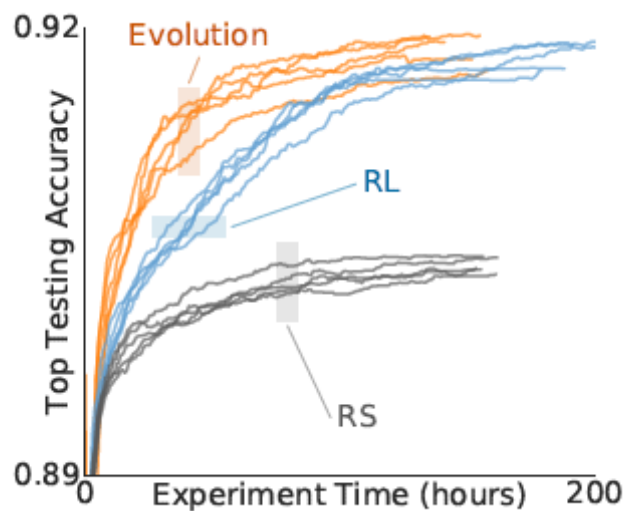


Рисунок 2.8 — Точність моделі відносно RL [1] to RS

На пізніх етапах, якщо дозволити працювати на повних моделях 20k (як у дослідженні базової лінії), еволюція виробила моделі з однаковою точністю. Важливо відзначити, що вертикальна вісь на рисунок 2.8 не представляє обчислювальної вартості моделей, тільки їх точність. Таким чином, еволюція займала ідеальну відносну позицію на цьому графіку в рамках нашого прикладу. До цих пір ми порівнювали еволюцію з нашим відтворенням експериментів у базовому дослідженні, але також є інформативним, щоб порівняти безпосередньо з результатами, про які повідомлялося в базовому дослідженні. Ми вибираємо нашу розвинуту

архітектуру з найвищою точністю перевірки і називаємо її AmoebaNet-A, яка зображена на рисунку 2.9.

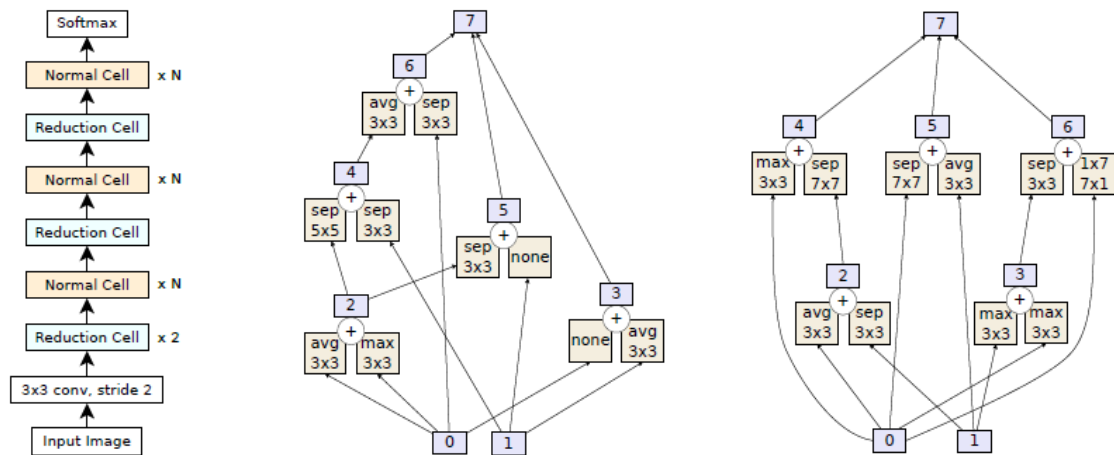


Рисунок 2.9 — Фінальна архітектура

Таблиця 2.1 — Порівняння точності тестування з вершиною моделі базового дослідження NASNet-A.

Модель	Параметри (М)	Помилка (%)
NASNet-A (baseline)	3.3	3.41
AmoebaNet-A (N=6, F=32)	2.6	3.40
AmoebaNet-A (N=6, F=36)	3.2	3.34

### 2.3 Висновки до розділу

У даному розділі дипломної роботи було зроблено опис моделей автоматичного побудування архітектури згорткових нейронних мереж.

Було оглянуто модель основану на навчанні з підкріпленням, яка використовує додатково контролер, побудований на рекурентній нейронній мережі. Також було оглянуто модель автоматичної побудови згорткової нейронної мережі на основі алгоритму еволюції, надані результати та їх аналіз.

## 3 РІШЕННЯ ЗАДАЧІ ПОШУКУ АРХІТЕКТУРИ

### 3.1 DARTS

У цій роботі ми підходимо до проблеми з іншого боку і пропонуємо метод ефективного пошуку архітектури під назвою DARTS (диференційований пошук архітектури). Замість того, щоб шукати дискретний набір архітектур-кандидатів, ми робимо простір пошуку безперервним, так що архітектура може бути оптимізована по відношенню до продуктивності набору валідації за допомогою градієнтного спуска. Ефективність даних на основі градієнта, на відміну від неефективного пошуку в чорній коробці, дозволяє DARTS досягати конкурентних показників за допомогою сучасних технологій, використовуючи на порядки менше ресурсів обчислення. Це також перевершує інший ефективний метод пошуку архітектури ENAS [12]. Примітно, що DARTS є простішим, ніж багато існуючих підходів, оскільки він не включає жодних контролерів (як у главі 2.1), гіпермережі або прогностичні показники, але воно є достатньо загальним для пошуку як згорткових, так і рекурентних архітектур.

Ідея пошуку архітектур в рамках безперервного домену не є новою [2], але є кілька основних відмінностей. Хоча попередні роботи прагнуть тонко налаштувати певний аспект архітектури, наприклад, форми фільтрів або структури розгалужень у згортковій мережі, DARTS здатний виявляти високопродуктивні архітектури зі складною топологією графів у просторі пошуку. Більш того, DARTS не обмежується жодною конкретною архітектурною сім'єю і здатний виявляти як згорткові, так і рекурентні мережі.

### 3.1.1 Пошуковий простір

Також як Zoph et al. [13], шукаємо обчислювальну клітину як будівельний блок кінцевої архітектури. Вивчену клітинку можна або скласти, щоб сформувати згорткову мережу, або рекурсивно з'єднати, щоб сформувати рекурентну мережу.

Кожен вузол  $x(i)$  є латентним представленням (наприклад, картою ознак у згорткових мережах), і кожен спрямований край  $(i, j)$  пов'язаний з деякою операцією  $o(i, j)$ , яка перетворює  $x(i)$ . Будемо вважати, що комірка має два вхідних вузла і один вихідний вузол. Для згорткових осередків вхідні вузли визначаються як вихідні клітинки у попередніх двох шарах. Для повторюваних комірок вони визначаються як вхідні дані на поточному кроці, та стан, що переноситься з попереднього кроку. Вихід комірки отримують, застосовуючи операцію відновлення (наприклад, конкатенацію) до всіх проміжних вузлів.

Кожен проміжний вузол обчислюється на основі всіх його попередників:

$$x^{(i)} = \sum_{i < j} o^{(i,j)}(x^{(j)}), \quad (3.1)$$

Спеціальна нульова операція також включена, щоб вказати на відсутність зв'язку між двома вузлами. Таким чином, завдання вивчення комірки зводиться до вивчення операцій по її краях.

На рисунку 3.1 можна побачити, що спочатку операції на краях невідомі (a), також безперервну релаксацію простору шляхом розміщення на кожному краю суміші операцій-кандидатів (b), спільну оптимізацію ймовірностей змішування і вагових коефіцієнтів мережі шляхом вирішення двовимірної задачі оптимізації (c), індукування остаточної архітектури з вивчених ймовірностей змішування (d).

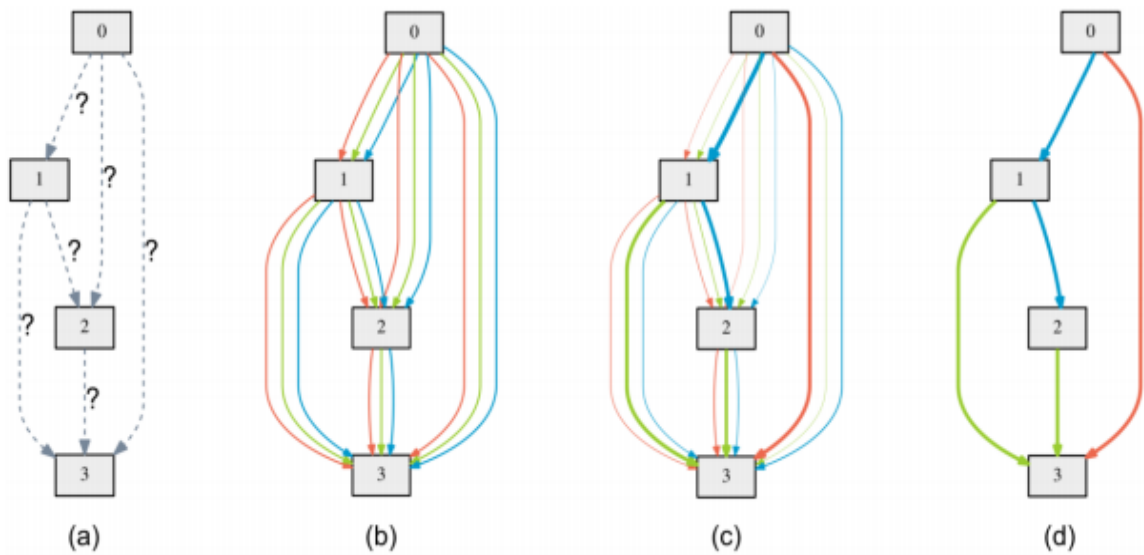


Рисунок 3.1 — Огляд DARTS

### 3.1.2 Безперервна релаксація та оптимізація

Нехай  $O$  — набір операцій-кандидатів (наприклад, згортка, максимальне об'єднання, нуль), де кожна операція являє собою деяку функцію  $o(\cdot)$ , яка буде застосована до  $x(i)$ . Для того, щоб простір пошуку був безперервним, ми розслабляємо категоричний вибір конкретної операції, як softmax над усіма можливими операціями:

$$\bar{o}^{(i,j)}(x) = \sum_{o \in O} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in O} \exp(\alpha_{o'}^{(i,j)})} o(x), \quad (3.2)$$

де операції змішування ваг для пари вузлів  $(i, j)$  параметризовані за допомогою вектора  $\alpha_o^{(i,j)}$  розмірності  $|O|$ .

Після релаксації завдання пошуку архітектури зводиться до вивчення безлічі безперервних змінних  $\alpha = \operatorname{argmax}_{o \in O} \{ \alpha_o^{(i,j)} \}$ , як показано на рисунку 3.1. Після закінчення пошуку, дискретна архітектура отримується шляхом заміни кожної змішаної операції  $\bar{o}^{(i,j)}$  з найбільш ймовірною

операцією, тобто  $o^{(i,j)} = \operatorname{argmax}_{o \in O} \alpha_o^{(i,j)}$ . У подальшому ми посилаємося на  $\alpha$  як на архітектуру (кодування).

Після релаксації наша мета полягає в спільному вивченні архітектури  $\alpha$  і ваг  $w$  у всіх змішаних операціях (наприклад, ваги фільтрів згортки). Аналогічний архітектурному пошуку за допомогою навчання з підкріпленням (дивитись главу 2.1) або еволюції (дивитись главу 2.2), де ефективність набору перевірок трактується як винагорода або придатність, DARTS прагне оптимізувати втрату на валідаційних даних, але з використанням градієнтного спуску.

Позначимо через  $L_{train}$  і  $L_{val}$  втрату на тренувальних даних і втрату на валідаційних даних відповідно. Обидві втрати визначаються не тільки архітектурою  $\alpha$ , але і вагою  $w$  у мережі. Мета архітектурного пошуку полягає в тому, щоб знайти  $\alpha^*$ , що мінімізує втрати валідації  $L_{val}(w^*, \alpha^*)$ , де ваги  $w^*$ , пов'язані з архітектурою, отримуються шляхом мінімізації втрат на навчання  $w^* = \operatorname{argmin}_w L_{train}(w, \alpha^*)$ . Мається на увазі рішення задачі оптимізації на двох рівнях [14] з  $\alpha$  як змінну верхнього рівня і  $w$  як змінну нижнього рівня:

$$\min_{\alpha} L_{val}(w^*(\alpha), \alpha), \quad (3.3)$$

$$s. t. w^*(\alpha) = \operatorname{argmin}_w L_{train}(w, \alpha), \quad (3.4)$$

Викладене формулювання також виникає в оптимізації гіперпараметрів на основі градієнта. Це пов'язано з тим, що безперервну архітектуру  $\alpha$  можна розглядати як особливий тип гіперпараметра, хоча його розмірність істотно вище, ніж скалярні гіперпараметри (наприклад, швидкість навчання), і її важче оптимізувати.

### 3.1.3 Наближення

Вирішення двовимірної оптимізації є непосильним, оскільки вимагатиме перерахунку  $w^*(\alpha)$  шляхом вирішення внутрішньої задачі оптимізації (3.4), коли відбувається будь-яка зміна в  $\alpha$ . Таким чином, ми пропонуємо приблизну процедуру оптимізації, де  $w$  і  $\alpha$  оптимізуються шляхом чергування кроків градієнтного спуску у вагових і архітектурних просторах відповідно, як показано на рисунку 3.2.

---

**Algorithm 1:** DARTS – Differentiable Architecture Search
 

---

Create a mixed operation  $\bar{o}^{(i,j)}$  parametrized by  $\alpha^{(i,j)}$  for each edge  $(i,j)$   
**while not converged do**  
     1. Update weights  $w$  by descending  $\nabla_w \mathcal{L}_{train}(w, \alpha)$   
     2. Update architecture  $\alpha$  by descending  $\nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$   
 Replace  $\bar{o}^{(i,j)}$  with  $o^{(i,j)} = \operatorname{argmax}_{o \in \mathcal{O}} \alpha_o^{(i,j)}$  for each edge  $(i,j)$

---

Рисунок 3.2 – Алгоритм наближення

На етапі  $k$  з урахуванням поточної архітектури  $\alpha_{k-1}$  отримуємо  $w_k$ , переміщуючи  $w_{k-1}$  у напрямку мінімізації тренувальної витрати  $L_{train}(w_{k-1}, \alpha_{k-1})$ . Потім, зберігаючи фіксовані ваги  $w_k$ , ми оновлюємо архітектуру таким чином, щоб зменшити втрату на валідаційному наборі даних після **одного** етапу сходження градієнта:

$$L_{val}(w_k - \xi * \nabla L_{train}(w_k, \alpha_{k-1}), \alpha_{k-1}), \quad (3.5)$$

де  $\xi$  – це швидкість навчання для цього віртуального кроку градієнта. Мотивація (3.5) полягає в тому, що ми хотіли б знайти архітектуру, яка має низьку втрату на валідаційних даних, коли її ваги оптимізовані за допомогою спуску градієнтів, де однокрокові розгорнуті ваги служать сурогатом для  $w^*(\alpha)$ . Для мета-навчання та трансферу моделей був використаний подібний підхід [15]. Примітно, що динаміка цього

ітеративного алгоритму визначає гру Stackelberg між оптимізатором  $\alpha$  (лідером) і оптимізатором  $w$  (послідовник), який зазвичай вимагає від лідера передбачити наступний крок наступного кроку для досягнення рівноваги. Хоча в даний час невідомо про гарантії збіжності цього алгоритма оптимізації, але на практиці він здатен сходиться з прийнятним вибором  $\xi$ . Також можна зауважити, що коли імпульс для оптимізації ваги включений, єдина навчальна мета (3.5) змінюється відповідно, і весь аналіз все ще застосовується.

Архітектурний градієнт дається шляхом диференціації (3.5) по  $w$  (пропускаємо індекс кроку  $k$  для стислості):

$$\nabla_{\alpha} L_{val}(w', a) - \xi \nabla_{a,w}^2 L_{train}(w, a) \nabla_{w'} L_{val}(w', a), \quad (3.6)$$

де  $w' = w - \xi \nabla_w L_{train}(w, a)$  позначає вагові коефіцієнти для однокрокової моделі вперед. Градієнт (3.6) містить матрично-векторний перемноження у своєму другому члені, який є дорогим для обчислення. На щастя, складність може бути значно зменшена за допомогою апроксимації кінцевої різниці:

$$\begin{aligned} \xi \nabla_{a,w}^2 L_{train}(w, a) \nabla_{w'} L_{val}(w', a) \approx \\ \frac{(\nabla_a L_{train}(w^+, a) - \nabla_a L_{train}(w^-, a))}{2e}, \end{aligned} \quad (3.7)$$

Оцінювання кінцевої різниці вимагає лише двох прямих пропусків для ваг і двох зворотних проходів для  $\alpha$ , а складність зменшується від  $O(|\alpha| |w|)$  до  $O(|\alpha| + |w|)$ .

Якщо  $\xi = 0$ , то похідна другого порядку в (3.6) зникне. У цьому випадку градієнт архітектури задається  $\nabla_{\alpha} L_{val}(w, a)$ , що відповідає простій евристичі оптимізації втрати на валідаційних даних за умови, що  $\alpha$



і  $w$  незалежні один від одного. Це призводить до деякого прискорення, але емпірично гірших показників, згідно з експериментальними результатами в табл. 3.1 та табл. 3.2.

### 3.1.4 Виведення дискретних архітектур

Після отримання безперервної архітектури, що кодує  $\alpha$ , дискретну архітектуру можна отримати:

- 1) Зберігаючи  $k$  найсильніших попередників для кожного проміжного вузла, де сила ребра визначається як  $\max_{o \in O, o \neq \text{zero}} \frac{\exp(a_o^{i,j})}{\sum_{o' \in O} \exp(a_{o'}^{i,j})}$ . Щоб зробити нашу похідну архітектуру порівнянною з такою в існуючих роботах, ми використовуємо  $k = 2$  для згорткових клітин.
- 2) Замінюючи кожну змішану операцію на як найбільш ймовірну операцію.

На рисунку 3.3 можна побачити динаміку навчання ітераційного алгоритму при  $L_{val}(w, \alpha) = \alpha w - 2\alpha + 1$  і  $L_{train}(w, \alpha) = w^2 - 2\alpha w + \alpha^2$ , починаючи з  $(\alpha(0), w(0)) = (2, -2)$ . Аналітичним рішенням для відповідної задачі двох-рівневої оптимізації є  $(\alpha^*, w^*) = (1, 1)$ , що висвічується в червоному колі. Пунктирна червона лінія вказує на можливий набір, де обмеження (3.4) виконується точно (а саме ваги в  $w$  є оптимальними для даної архітектури  $\alpha$ ). Приклад показує, що відповідний вибір  $\xi$  допомагає сходитися до кращого локального оптимуму.

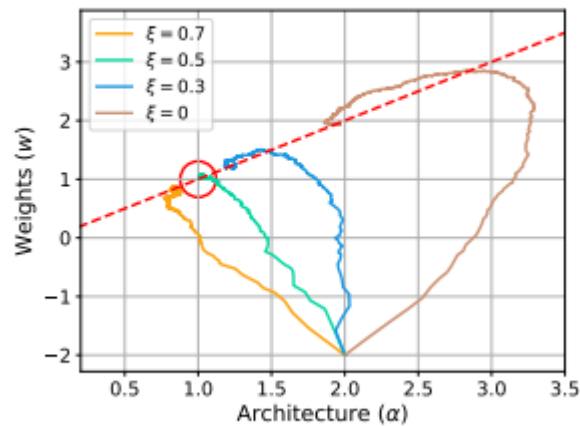


Рисунок 3.3 — Динаміка навчання алгоритму.

### 3.1.5 Пошук архітектури згорткової мережі на CIFAR-10

Ми включаємо наступні операції в  $O: 3 \times 3$  і  $5 \times 5$  роздільних згорток,  $3 \times 3$  і  $5 \times 5$  розріджених відокремлюваних згорток,  $3 \times 3$  максимального об'єднання,  $3 \times 3$  середнього об'єднання, ідентичності і нуля. Усі операції мають один крок (якщо застосовується) і згорнуті карти властивостей доповнюються для збереження їх просторового дозволу. Ми використовуємо порядок ReLU-Conv-BN для згорткових операцій, і кожна зрізана згортка завжди застосовується двічі [13].

Наша згорткова клітина складається з  $N = 7$  вузлів, серед яких вихідний вузол визначається як поглиблена конкатенація всіх проміжних вузлів (вхідні вузли виключені). Іншу частину налаштувань формується шляхом складання декількох клітин разом. Перший і другий вузли клітини  $k$  задаються рівними виходам клітини  $k - 2$  і клітини  $k - 1$  відповідно, а в разі необхідності вставляються згортки  $1 \times 1$ . Клітини, розташовані на  $1/3$  і  $2/3$  від загальної глибини мережі, є редукційними клітинами, в яких всі операції, розташовані поруч з вхідними вузлами, мають два кроки. Архітектура кодування це  $(\alpha_{normal}, \alpha_{reduce})$ , де  $\alpha_{normal}$  поділяється всіма

нормальними клітинами, а  $\alpha_{reduce}$  розділяється всіма редукційними клітинами.

Оскільки архітектура буде змінюватися протягом усього процесу пошуку, ми завжди використовуємо специфіку пакета для нормалізації. Досліджувані афінні параметри в усіх нормалізаціях пакетів вимикаються під час процесу пошуку, щоб уникнути масштабування виходів операцій-кандидатів.

Щоб виконати пошук архітектури, ми зберігаємо половину даних тренувань CIFAR-10 як набори перевірок. Невелика мережа, що складається з 8 клітин, навчається з використанням DARTS на 50 епох, з розміром партії 64 (як для навчального, так і для валідаційного наборів) і початковим числом каналів 16. Цифри, що були обрані для того, щоб мережа могла вписатися в один GPU. Ми використовуємо імпульс SGD для оптимізації ваг  $w$ , причому початкова швидкість навчання  $\eta w = 0,025$  (віджигається до нуля за косинусним графіком), імпульс 0,9 і розпад ваги  $3 \times 10^{-3}$ . Ми використовуємо Adam як оптимізатор для архітектурних змінних ( $\alpha$  в обох нормальних і редукційних клітинах), з початковою швидкістю навчання  $\eta \alpha = 3 \times 10^{-4}$ , імпульсом  $\beta = (0.5, 0.999)$  і ваговим спадом  $10^{-3}$ . Пошук займає один день з одним тільки GPU.

### 3.1.6 Результати навчання

На рисунку 3.4 ми відстежуємо останні архітектури з часом. Кожен знімок архітектури повторно тренується з нуля за допомогою навчального набору (для 100 епох на CIFAR-10) і потім оцінюється на наборі перевірок. Для кожного завдання ми повторюємо експерименти 4 рази з різними випадковими параметрами, і повідомляємо про медіану і найкращу (за прогон) перевірку продуктивності архітектур з плином часу. В якості посилок ми також повідомляємо про результати (при тій же самій системі

оцінювання; зі порівнянною кількістю параметрів) кращих існуючих клітин, виявлених за допомогою навчання з підкріпленням або еволюції, включаючи NASNet-A [1] (1800 GPU днів), АмобеаNet-A (3150 GPU днів) [2] і ENAS [3].

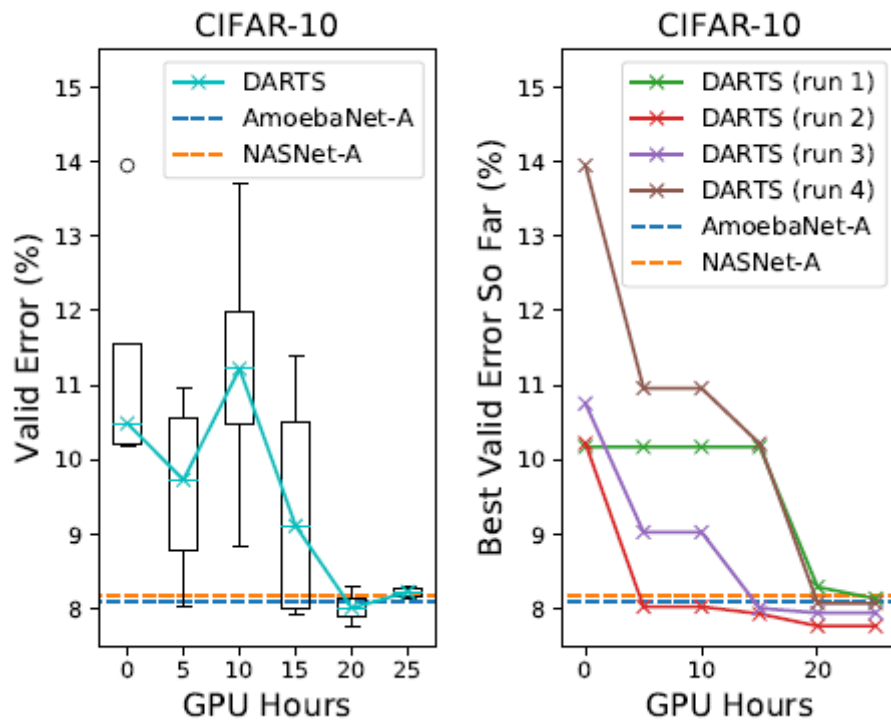


Рисунок 3.5 — Пошук прогресу DARTS для згорткових клітин на CIFAR-10

Щоб вибрати архітектуру для оцінки, ми запускаємо DARTS чотири рази з різними випадковими параметрами і підбираємо найкращу клітинку на основі результатів перевірки. Щоб оцінити вибрану архітектуру, ми випадковим чином ініціалізуємо її ваги (ваги, витягнуті під час процесу пошуку, відкидаються), тренуємо їх з нуля і повідомляємо про його роботу на тестовому наборі. Відзначимо, що тестовий набір ніколи не використовується для пошуку архітектури або вибору архітектури.

Велика мережа з 20 клітин навчається на 600 епох з розміром партії 96. Інші гіперпараметри залишаються такими ж, як ті, що використовуються для пошуку архітектури. Навчання займає 1,5 дні на одному GPU в PyTorch. Оскільки результати CIFAR піддаються високій дисперсії навіть з точно однаковою установкою, ми повідомляємо про середнє і стандартне відхилення 4 незалежних прогонів для нашої повної моделі. Щоб уникнути невідповідності між різними реалізаціями або налаштуваннями навчання (наприклад, розмірами партії), ми включили клітинку NASNet-A [1] і клітинку AmoebaNet-A [2] у нашу навчальну структуру і повідомили про свої результати під тими ж налаштуваннями, що й наші клітини.

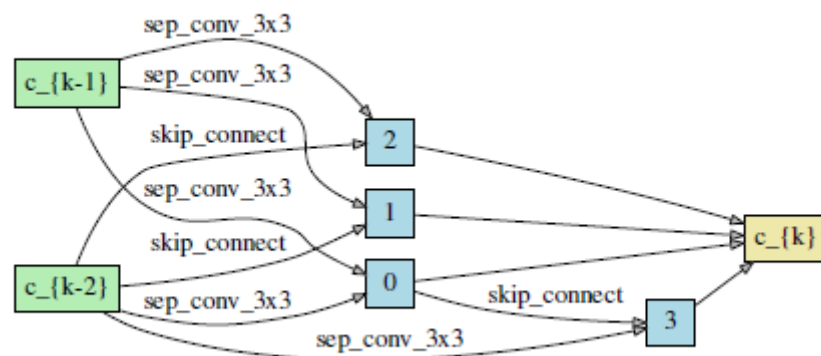
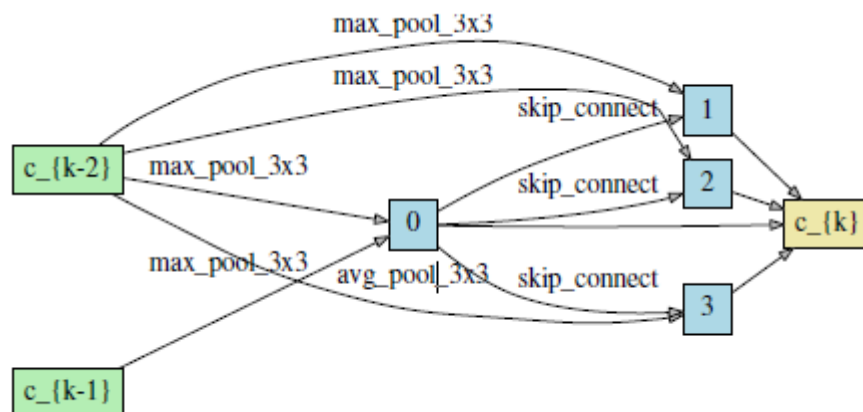


Рисунок 3.5 – Нормальна клітина навчена на CIFAR-10



### Рисунок 3.6 – Редукційна клітина навчена на CIFAR-10

Результати CIFAR-10 для згорткових архітектур представлені в таблиці 3.1. Зокрема, DARTS досягла порівняльних результатів з сучасним рівнем техніки [1], використовуючи на три порядки менші обчислювальні ресурси, тобто 1,5 або 4 дні GPU проти 1800 днів GPU для NASNet і 3150 днів GPU для AmoebaNet. Більш того, з трохи довшим часом пошуку, DARTS перевершив ENAS [3], виявивши клітини з порівнянними показниками помилок, але менше параметрів. Більш тривалий час пошуку пояснюється тим, що ми повторювали процес пошуку чотири рази для вибору стільника. Проте ця практика є менш важливою для згорткових клітин, тому що продуктивність відкритих архітектур не сильно залежить від ініціалізації.

Таблиця 3.1 — Зрівняння архітектур на CIFAR-10

Архітектура	Помилка (%)	Параметри (М)	Час навчання (GPU дні)	Метод пошуку
DenseNet-BC	3.46	25.6	-	вручну
NASNet-A + cutout [1]	2.65	3.3	1800	нав. з підкр.
AmoebaNet-A + cutout [2]	3.12	3.1	3150	еволюція
Hierarchical Evo	3.75	15.7	300	еволюція
ENAS + cutout [3]	2.89	4.6	0.5	нав. з підкр.
Random + cutout	3.49	3.1	-	-
DARTS + cutout	2.94	2.9	1.5	градієнтний
DARTS + cutout	2.83	3.4	4	градієнтний

На даних ImageNet навчається мережа з 14 клітин на 250 епохах з розміром партії 128, ваговим розпадом  $3 * 10^{-3}$  і початковим рівнем навчання  $SGD = 0,1$  (розкладений на 0,97 після кожної епохи). Інші гіперпараметри слідують такі, як і на CIFAR-10. Навчання триває 12 днів на одному GPU. Результати в Таблиці 3 показують, що клітина, вивчена на CIFAR-10, дійсно може бути передана до ImageNet. Варто зауважити, що DARTS досягає конкурентних показників за допомогою сучасного методу навчання з підкріпленням [1], використовуючи на три порядки менше обчислювальних ресурсів.

Таблиця 3.2 — Зрівняння архітектур на ImageNet

Архітектура	Помилка (top-5)	Параметри (М)	Час навчання (GPU дні)	Метод пошуку
Inception-v1	10.1	6.6	-	вручну
MobileNet	10.5	4.2	-	вручну
NASNet-A [1]	8.4	5.3	1800	нав. з підкр.
AmoebaNet-A [2]	8.0	5.1	3150	еволюція
DARTS (навчений на CIFAR-10)	9.0	4.9	4	градієнтний

### 3.2 Висновки до розділу

У даному розділі дипломної роботи було розглянуто метод автоматичного побудування архітектури згорткової нейронної мережі DARTS. Наведено алгоритм працювання моделі, надані результати навчання, аналіз результатів, та порівняння результатів моделі з іншими підходами.



## **4 ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ**

### **4.1 Розрахунок кошторису витрат на проведення й впровадження результатів науково-дослідної роботи**

Виконання наукових досліджень, а також впровадження результатів науково-дослідної роботи (НДР) вимагає певних витрат, які необхідно розглядати як додаткові капіталовкладення. Витрати на проведення й впровадження результатів НДР відносяться до виробничих витрат. Як правило, всі витрати документально оформляються у вигляді кошторису. Основними статтями кошторису витрат є заробітна плата, нарахування на заробітну плату, вартість електроенергії (технологічна й освітлювальної), вартість оренди приміщення, амортизаційні відрахування на обчислювальну техніку, вартість впровадження й освоєння результатів НДР і планові накопичення.

#### **4.1.1 Розрахунок фонду заробітної плати виконавців**

Розрахунок фонду заробітної плати виконавців проводиться виходячи зі штатного розкладу й зайнятості виконавців у даній НДР. Виконавцями даної НДР є керівник дипломної роботи, консультанти частини економічного обґрунтування, частини охорони праці й частини цивільного захисту, а також інженер-математик. Штатний розклад приведено в таблиці 4.1.

Таблиця 4.1 - Штатний розклад

Посада	Кількість виконавців	Час зайнятості, міс	Коефіцієнт трудової участі	Оклад на місяць, грн.	Заробітна плата, $Зп_{\text{оклад}}$ , грн.
Керівник роботи, доцент	1	4	0,075	8 000,00	2 400,00
Консультант частини економічного обґрунтування, професор	1	4	0,005	10 000,00	200,00
Консультант частини охорони праці, доцент	1	4	0,005	8 000,00	160,00
Консультант частини цивільного захисту, доцент	1	4	0,005	8 000,00	160,00
Виконавець, інженер-математик дослідник	1	4	1	10 000,00	40 000,00
Усього				44 000,00	42 920,00

Заробітна плата виконавців НДР складається з основної заробітної плати й різних доплат до неї:

$$Зп = Зп_{\text{осн}} + Зп_{\text{д}}, \quad (4.1)$$

де  $Зп_{\text{осн}}$  – основна заробітна плата;

$Зп_{\text{д}}$  – доплати до заробітної плати.

$$Зп_{\text{осн}} = Зп_{\text{оклад}} + Зп_{\text{прем}}, \quad (4.2)$$

де  $Зn_{оклад}$  – розмір заробітної плати за штатним розкладом;  
 $Зn_{прем}$  – розмір премій.

$$Зn_{\partial} = K_{\partial} \cdot Зn_{осн}, \quad (4.3)$$

де  $K_{\partial}$  – коефіцієнт доплат до заробітної плати,  $K_{\partial} = 0,09$

.

$$Зn_{прем} = K_{прем} \cdot Зn_{оклад}, \quad (4.4)$$

де  $K_{прем}$  – коефіцієнт преміювання,  $K_{прем} = 0.12$ ;

Розрахуємо заробітну плату для керівників дипломної роботи:

Керівник дипломної роботи:

$$Зn = 2400 \cdot (1,0 + 0,12) \cdot (1,0 + 0,09) = 2\,929,92 \text{ грн};$$

Консультант частини з економічного обґрунтування:

$$Зn = 200 \cdot (1,0 + 0,12) \cdot (1,0 + 0,09) = 244,16 \text{ грн};$$

Консультант частини охорони праці:

$$Зn = 160 \cdot (1,0 + 0,12) \cdot (1,0 + 0,09) = 195,32 \text{ грн};$$

Консультант частини цивільної оборони:

$$Зn = 160 \cdot (1,0 + 0,12) \cdot (1,0 + 0,09) = 195,32 \text{ грн};$$

Інженер-математик:

$$Зn = 40\,000 \cdot (1,0 + 0,12) \cdot (1,0 + 0,09) = 48\,832,00 \text{ грн};$$

Всього витрати на заробітну плату складають:

$$Зn_{заг} = 2\,929,92 + 244,16 + 195,32 + 195,32 + 48\,832,00 = 52\,396,72 \text{ грн.}$$

Всього витрати на заробітну плату склали 52 396,72 грн.

#### 4.1.2 Відрахування на соціальне страхування

Відрахування на соціальне страхування й інші відрахування розраховуються на підставі отриманого значення фонду заробітної плати:

$$B_{\text{вд}} = K_{\text{вд}} \cdot Z_n, \quad (4.5)$$

де  $K_{\text{вд}}$  – коефіцієнт нарахувань на фонд заробітної плати, приймається в розмірі 0,22.

$$B_{\text{вд}} = 0,22 \times 52\,396,72 = 1375,10 \text{ грн.}$$

#### 4.1.3 Розрахунок технологічної електроенергії

Розрахунок *технологічної електроенергії* проводиться виходячи із завантаження устаткування, що використовується під час проведення НДР (ЕОМ, принтер, сканер і ін.), по наступній формулі:

$$E_{\text{тех}} = P \sum_{i=1}^N P_i T_i, \quad (4.6)$$

де  $P$  – тариф на електроенергію,  $P = 1,7808$  грн/кВт;

$P_i$  – споживана потужність  $i$ -ої одиниці встаткування комп'ютера,

$$P_i = 0.024 \text{ кВт/год};$$

$T_i$  – час роботи  $i$ -ої одиниці встаткування комп'ютера,  $T_i = 336$  год.

$$E_{\text{тех}} = 1,7808 \times 0,024 \times 336 = 14,36 \text{ грн.}$$

#### 4.1.4 Розрахунок електроенергії, що витрачає на освітлення

Розрахунок електроенергії, що витрачає на освітлення, виконується виходячи з норм охорони праці по освітленню робочих місць

$$E_{осв} = P \cdot N_{л} \cdot \Pi_{л} \cdot T, \quad (4.7)$$

де  $P$  – тариф на електроенергію,  $P = 1.7808$  грн/кВт.

$N_{л}$  – кількість ламп,  $N_{л} = 2$  шт;

$\Pi_{л}$  – споживана потужність однієї лампи,  $\Pi_{л} = 0,04$  кВт/год;

$T$  – час роботи ламп на висвітлення,  $T = 169$  год;

$$E_{осв} = 1,7807 \times 2 \times 0,04 \times 169 = 24,07 \text{ грн.}$$

#### 4.1.5 Амортизаційні відрахування на устаткування

Амортизаційні відрахування на устаткування розраховуються виходячи з залишкової вартості устаткування та його часу використання за наступною формулою:

$$A = \frac{a_{EOM}}{12} \sum_{i=1}^N 3B_i \cdot T_i, \quad (4.8)$$

де  $a_{EOM}$  – річна норма амортизації, що приймається в розмірі 26 % залишкової вартості устаткування;

$3B_i$  – залишкова вартість  $i$ -ої одиниці устаткування комп'ютера,

$$3B_i = 4739,78 \text{ грн;}$$

$T_i$  – час використання  $i$ -ої одиниці устаткування:

– комп'ютера,  $T_1 = 3,5$  міс;

– принтера,  $T_2 = 3,5$  міс;

$$A = \frac{0,26}{12} \times 4739,78 \times 3,5 = 359,43 \text{ грн.}$$

#### 4.1.6 Вартість оренди приміщення для проведення НДР

Розрахунок вартості оренди приміщення для проведення НДР враховує його площу та термін оренди та проводиться за наступною формулою:

$$Д = K_a \cdot S \cdot P \cdot T_{OP}, \quad (4.9)$$

де  $K_a$  – коефіцієнт, що враховує податок на майно,  $K_a = 1,25$ ;

$S$  – площа приміщення, де проводилася НДР,  $S = 10 \text{ м}^2$ ;

$P$  – вартість оренди одного квадратного метра приміщення,

$P = 200 \text{ грн/міс}$ ;

$T_{OP}$  – строк оренди,  $T_{OP} = 3,5 \text{ міс}$ ;

Таким чином, отримаємо вартість оренди приміщення:

$$Д = 1,2 \times 3,5 \times 200 \times 10 = 8750,00 \text{ грн.}$$

#### 4.1.7 Інші витрати

Інші витрати (опалення, робота кондиціонера й ін.) приймаються в розмірі 6% від вартості оренди приміщення:

$$З_{ін} = 8750,00 \times 0,06 = 525,00 \text{ грн.}$$

#### 4.1.8 Вартість впровадження й освоєння результатів НДР

Вартість впровадження й освоєння результатів НДР визначається виходячи з особливостей виробничого середовища організації, що потребує результатів даної НДР і може містити в собі витрати на

придбання або переналагодження устаткування, залучення додаткової або більш кваліфікованої робочої сили та ін.

Так як в завершенні роботі ми отримуємо математичну модель, яка автоматизовано створює згорткові нейронні мережі для класифікації зображень, зроблену за допомогою середовища програмного забезпечення Jupyter notebook – 1 400,00 грн., його налаштування – 750 грн., навчання працівників – 2 100 грн. Отже за рік витрати організації складатимуть:

$$B_{\text{вп}} = 1\,400,00 + 750,00 + 2\,100,00 = 4\,250,00 \text{ грн.}$$

#### 4.1.9 Витрати на проведення НДР

Витрати на проведення НДР являють собою суму витрат по окремих статтях:

$$З = З_n + B_{i\partial} + E_{\text{tex}} + E_{\text{осв}} + A + Д + З_{\text{ін}} + B_{\text{вп}}, \quad (4.9)$$

де  $З_{\text{ін}}$  – інші витрати;

$B_{\text{вп}}$  – вартість впровадження й освоєння результатів НДР.

Таким чином, витрати на проведення НДР становлять:

$$\begin{aligned} З = & 52\,396,72 + 1375,10 + 14,36 + 24,07 + 359,43 + 8750,00 + \\ & + 525,00 + 4250,00 = 67\,694,68 \text{ грн.} \end{aligned}$$

#### 4.1.10 Планові накопичення

Планові накопичення обираються в розмірі 30 % від витрат на проведення НДР:

$$\text{ПН} = 67\,694,68 \times 0,3 = 20308,40 \text{ грн.}$$

#### 4.1.11 Кошторис витрат на проведення НДР

Кошторис витрат на проведення НДР є сумою витрат на проведення НДР і планових накопичень. Результати розрахунку кошторису витрат приведено в таблиці 4.2.

Таблиця 4.2 - Кошторис витрат на проведення НДР

Стаття витрат	Сума, грн
1 Заробітна плата	52396,72
2 Відрахування на соціальне страхування	1375,1
3 Технологічна електроенергія	14,36
4 Електроенергія на освітлення	24,07
5 Амортизаційні відрахування на устаткування	359,43
6 Вартість оренди приміщення	8750,00
7 Інші витрати	525,00
8 Вартість впровадження й освоєння результатів НДР	4250,00
9 Разом витрат	67 694,68
10 Планові накопичення	20 308,40
Усього кошторис витрат на проведення НДР	87 617,07

#### 4.2 Класифікація й кодування запропонованої інновації

Основними критеріями класифікації інновацій повинні бути:

- комплексність набору класифікаційних ознак, що враховують, для аналізу й кодування;
- можливість кількісного (якісного) визначення критерію;
- наукова новизна й практична цінність запропонованої ознаки класифікації.

З урахуванням наявного досвіду й наведених критеріїв пропонується наступна класифікація нововведень і інновацій (див. таблицю 4.3).



Таблиця 4.3 - Класифікація нововведень і інновацій

Ознака класифікації	Види інновацій
1 Рівень новизни інновації	1.1 Радикальні (відкриття, винаходи) 1.2 Ординарні (ноу-хау, раціоналізаторські пропозиції)
2 Стадія життєвого циклу товару, на якій впроваджується інновація	2.1 Інновації, впроваджувані на стадії стратегічного маркетингу 2.2 Інновації, впроваджувані на стадії НДОКР 2.3 Інновації, впроваджувані на стадії ОТПВ 2.4 Інновації, впроваджувані на стадії виробництва 2.5 Інновації, впроваджувані на стадії сервісного обслуговування
3 Масштаб новизни інновації	3.1 Інновації у світовому масштабі 3.2 Інновації в країні 3.3 Інновації в галузі 3.4 Інновації для підприємства
4 Галузь народного господарства, де впроваджується інновація	4.1 Наука 4.2 Освіта 4.3 Соціальна сфера 4.4 Матеріальне виробництво 4.5 Роботи й послуги
5 Сфера застосування інновації	5.1 Інновації для внутрішнього застосування 5.2 Інновації для нагромадження в організації 5.3 Нововведення для продажу
6 Частота застосування інновації	6.1 Разові 6.2 Повторювані
7 Форма нововведення – основа інновації	7.1 Відкриття, винаходи, патенти 7.2 Раціоналізаторські пропозиції 7.3 Ноу-хау 7.4 Товарні знаки, торговельні марки, емблеми 7.5 Нові документи, що описують технологічні, виробничі, управлінські процеси, конструкції, структури, методи

Кінець таблиці 4.3

Ознака класифікації	Види інновацій
8 Вид ефекту, отриманого в результаті впровадження інновації	8.1 Науково-технічний 8.2 Соціальний 8.3 Екологічний 8.4 Економічний (комерційний) 8.5 Інтегральний
9 Підсистема системи керування, у якій впроваджується інновація	9.1 Підсистема наукового супроводу 9.2 Цільова підсистема 9.3 Підсистема, що забезпечує 9.4 Керована підсистема 9.5 Керуюча підсистема

Наведена класифікація охоплює всі аспекти інноваційної діяльності. Для спрощення управління інноваційною діяльністю на основі цієї класифікації інновації можна кодувати. Кодування може бути спрощене (з одним знаком для ознаки) і детальне (із двома й більше знаками для ознаки). У цьому випадку використовується спрощене кодування, при якому код інновації буде мати 9 цифр. Номер цифри відповідає ознаці класифікації в запропонованому вище порядку, а значення цифри відповідає виду інновації.

Після аналізу проекту, було визначено наступний код класифікації нововведень і інновацій: 2.2.4.5.1.2.5.4.1.

### **4.3 Розрахунок економічного ефекту від впровадження результатів НДР**

Економічний ефект розраховується виходячи із суми, отриманих від впроваджених результатів НДР, доходів:

$$D_e = \sum_{i=1}^n D_i, \quad (4.10)$$

де  $D_i$  – величина додаткових доходів або економії коштів, отриманих у результаті впровадження НДР, по  $i$ -му фактору.

Програмне забезпечення може замінити фахівців у сфері машиного навчання. Як правило, заробітна плата спеціаліста у сфері машиного навчання становить близько 30000 грн. Зменшення загальної заробітної плати у два рази за рахунок скорочення кількості працівників призведе до економії коштів, що на рік складатиме:

$$D_1 = 30000 \times 12 = 360\,000,00 \text{ грн/рік.}$$

#### **4.4 Укрупнена оцінка прибутковості запропонованого інноваційного проекту**

Укрупнена оцінка прибутковості інноваційного проекту, запропонованого в дипломній роботі, припускає визначення наступних показників:

- 1) чистий дисконтований доход по роках реалізації проекту;
- 2) чиста поточна вартість проекту по роках реалізації проекту;
- 3) індекс прибутковості проекту;
- 4) внутрішня норма прибутковості;
- 5) строк окупності проекту.

Розрахунок цих показників проводиться виходячи з наступних даних:

- 1) одноразові витрати в розрахунковому році (кошторис витрат на проведення й впровадження результатів НДР);
- 2) щорічні очікувані доходи від проекту;
- 3) процентна ставка в розрахунковому році;
- 4) інфляція на розглянутому ринку;
- 5) рівень ризику проекту.

Для початку визначимо ставку дисконту проекту по формулі

$$d = k + i + r, \quad (4.11)$$

де  $k$  – ціна капіталу (процентна ставка), частки одиниці,  $k = 0,11$ ;

$i$  – інфляція на ринку, частки одиниці,  $i = 0,16$ ;

$r$  – рівень ризику проекту, частки одиниці,  $r = 0,07$ .

$$d = 0,11 + 0,16 + 0,07 = 0,34$$

Чистий дисконтований дохід розраховуємо по формулі

$$ЧДД_t = \frac{D_t - K_t}{(1 + d)^t}, \quad (4.12)$$

де  $D_t$  – доходи  $t$ -го року, грн.;

$K_t$  – капіталовкладення (витрати)  $t$ -го року, грн (у цьому випадку кошторис витрат на НДР).

Чисту поточну вартість для  $t$ -го року реалізації проекту визначаємо по формулі

$$ЧПВ_t = -\frac{K_{t-1}}{(1 + d)^{(t-1)}} + \frac{D_t}{(1 + d)^t}. \quad (4.13)$$

Розрахунок даного показника варто здійснювати до першого позитивного значення ЧПВ. Цей рік і завершить розрахунковий період для даного інноваційного проекту. Приклад розрахунку чистого дисконтного доходу і чистої поточної вартості приведено в табл. 4.4.

Таблиця 4.4 – Розрахунок економічних показників

$t$	D	K	$\frac{1}{(1+d)^t}$	$\frac{D}{(1+d)^t}$	$\frac{K}{(1+d)^t}$	$\text{ЧДД}_t$	$\text{ЧПВ}_t$
0	0	87 617,07	1	0	-87 617,07	-87 617,07	-87 617,07
1	360000,00	0	0,75	270 000,00	0	270 000,00	182 382,93
Всього	360000,00	87 617,07	–	270000,00	87 617,07	182 382,93	–

Індекс прибутковості визначимо по формулі

$$IP = \frac{\sum_{t=1}^T \frac{D_t}{(1+d)^t}}{\sum_{t=1}^T \frac{K_t}{(1+d)^t}} \quad (4.14)$$

де  $T$  - кількість років у розрахунковому періоді.

Підставляючи значення в формулу, маємо  $IP = 9,64$ .

Внутрішня норма прибутковості являє собою ставку дисконту, при якій величина дисконтованих доходів усього розрахункового періоду дорівнює дисконтованим капіталовкладенням. Цей показник допомагає ухвалювати рішення щодо доцільності розробки й впровадження інноваційного проекту в умовах мінливих процентних ставок, ризиках, інфляції.

Внутрішню норму прибутковості можна визначити з табл. 4.5.

Таблиця 4.5 – Розрахунок ВНП

$d$	0,34	0,44	0,54	0,64	0,706
$\text{ЧДД}$	182 382,93	150402,42	80220,42	10082,42	-37,58

Виходячи з табл. 4.5 ВВП складає приблизно 0,706.

Строк окупності розраховується починаючи з місяця запуску проекту до місяця в якому досягається наступна рівність

$$\sum_{t=1}^T \frac{D_t}{(1+d)^t} = \sum_{t=1}^T \frac{K_t}{(1+d)^t} \quad (4.15)$$

З табл. 4.4 можна зробити висновок, що термін окупності складає 2 місяця.

#### **4.5 Висновки за розділом**

У даному розділі дипломної роботи було проведено економічне обґрунтування розробки продукту для дослідження та аналізу зображень клітин, зроблених за допомогою мікроскопа.

Розраховано кошторис витрат на НДР. Були розраховані витрати на заробітну платню виконавців дипломної роботи, витрати на електроенергію, амортизаційні відрахування, відрахування на соціальне страхування, оренду приміщення, витратні матеріали і планові накопичення. Було встановлено код інновації. Також був зроблений розрахунок економічного ефекту від впровадження результатів НДР, розрахована укрупнена оцінка прибутковості запропонованого проекту та визначено його строк окупності. Даний проект є прибутковим, оскільки індекс прибутковості більше одиниці. Цей показник допомагає ухвалювати рішення щодо доцільності розробки й впровадження інноваційного проекту в умовах мінливих процентних ставок, ризиках, інфляції. Отримані дані наведені у таблиці 4.6.

Таблиця 4.6 – Техніко-економічні показники

Найменування показника	Величина
1. Кошторис витрат на НДР, грн	87 617,07
2. Код інновації	2.2.4.5.1.2.5.4.1
3. Економічний ефект, грн	360000,00
4. Індекс прибутковості проекту	9,66
5. Внутрішня норма прибутковості	0,706
6. Строк окупності проекту, міс	2

## **5 ОХОРОНА ПРАЦІ І НАВКОЛИШНЬОГО СЕРЕДОВИЩА**

### **5.1 Аналіз умов праці на робочому місці**

Розділ виконано для етапу розробки на ЕОМ алгоритму автоматичного пошуку архітектур згорткових нейронних мереж.

Робота виконувалась на кафедрі «Комп'ютерної математики та аналізу даних» НТУ «ХПІ», яка розташована на другому поверсі семи поверхової будівлі.

Обладнання, приміщення і режим праці користувача повинні відповідати вимогам наступних нормативно-технічних документів:

— НПАОП 0.00-1.28-10. Правила охорони праці під час експлуатації електронно-обчислювальних машин.

— ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.

— ДСТУ Б.В.1.1-36:2016 Визначення категорій приміщень, будинків та установок за вибухопожежною та пожежною небезпекою.

— ДБН В.1.1-7-2016. Пожежна безпека об'єктів будівництва. Загальні вимоги.

— ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень.

— ДБН В.2.5-67:2013. Опалення, вентиляція та кондиціонування.

— ДБН В.2.5-28-2018. Природне та штучне освітлення.

— ДСН 3.3.6.037-99. Санітарні норми виробничого шуму, ультразвуку та інфразвуку.

— ГОСТ 12.1.029-80 ССБТ. Средства и методы защиты от шума. Классификация.

— ДСТУ ГОСТ 12.1.012:2008. Вібраційна безпека. Загальні вимоги.



— ДСН 3.3.6.039-99. Санітарні норми виробничої загальної та локальної вібрації.

— ДСТУ ГОСТ 2656885:2009. Вібрація. Методи і засоби захисту. Класифікація.

— ДСТУ ГОСТ 12.1.038:2008. Електробезпека. Гранично допустимі рівні напруг дотику і струмів.

— ПУЕ-2017. Правила улаштування електроустановок. – Чинний від 21.08.2017.

— НПАОП 40.1-1.32-01. Правила будови електроустановок. Електрообладнання спеціальних установок.

— ДСТУ ГОСТ 7237:2011. Електробезпека. Загальні вимоги та номенклатура видів захисту.

— ГОСТ 14254-96. Степени защиты, обеспечиваемые оболочками.

— НАПБ А.01.001-2014. Правила пожежної безпеки в Україні.

— ДСТУ БВ.2.5-38:2008. Інженерне обладнання будинків і споруд. Улаштування блискавка захисту будівель і споруд (ІЕС 62305:2006, NEQ).

— НАПБ Б.06.004-2005. Перелік однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежа гасіння та пожежної сигналізації.

— ГН 3.3.5-8.6.6.1-2002. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу.

Загальна характеристика виробничого приміщення, в якому виконувалась робота, приведена у таблиці 5.1.

В таблиці 5.2 надано перелік потенційних небезпечних та шкідливих факторів на робочому місці користувача ЕОМ з монітором на рідинних кристалах

## **5.2 Захист від шкідливого впливу факторів виробничого середовища**

Підтримка оптимальних параметрів мікроклімату в робочій зоні здійснюється відповідно вимог ДБН В.2.5-67:2013 за допомогою кондиціонеру, який регулює температуру повітря. Передбачена можливість природнього провітрювання приміщення. У холодний період року проводиться опалення від центральної тепломережі.

Таблиця 5.1 – Характеристика виробничого приміщення

№ п/п	Найменування показника	Характеристика показника	Обґрунтування вибору значення показника	Документ, що регламентує цій показник	Примітка
1	2	3	4	5	6
1	Розміри приміщення (м);	5x9,4x4,2	На одне р.м. з ЕОМ не менше 6.0 м <sup>2</sup> площі	ДСанПіН 3.3.2-007-98	Фактично 7,5 м <sup>2</sup> на одне р.м. з ЕОМ, що відповідає нормі
	кількість робочих місць (р.м.)	6			
2	Природне освітлення, вікна виходять на південь	Бокове, одностороннє; азимут 110°	Див. таблицю 5.2	ДБН В.2.5.-28-18	
			КПО не нижче 1.5 %	ДСанПіН 3.3.2-007-98	для р.м. з ЕОМ
3	Штучне освітлення, кількість світильників N; джерела світла	Загальне рівномірне; N=20; люмінесцентні лампи	Див. таблицю 5.2	ДБН В.2.5.-28-06	
			не нижче 300-500 лк	ДСанПіН 3.3.2-007-98	для робочих місць з ЕОМ
4	Характеристика трифазної електричної мережі	Чотири провідна з глухо заземленою нейтраллю напругою 380/220 В, частотою 50 Гц	Довгі кабельні мережі великої ємності	ПУЄ	
5	Клас приміщення за ступенем небезпеки ураження електрострумом	З підвищеною небезпекою	Є можливість одночасного дотику до металоконструкцій будівлі, що мають з'єднання з землею, та до металевих корпусів ЕОМ	ПУЕ	Необхідно передбачити заходи безпеки згідно вимог ПУЕ
6	Категорія приміщення з вибухо – пожежа небезпеки	В	Є тверді спаленні матеріали: папір, деревина тощо	ДСТУ Б.В.1.1-36:2016	
7	Ступінь вогнестійкості будівельних конструкцій	II	7-и поверхова будівля; категорія В	ДБН В.1.1.7–2016	
			Не нижче II ступеню	НПАОП 0.00-1.28-10	для будівель з ЕОМ

Кінець таблиці 5.2

№	Назва фактора	Джерела виникнення	Умови роботи	Нормативні параметри, їх значення	Документ, що регламентує показник
1	2	3	4	5	6
I. Небезпечні фактори					
1	Висока електрична напруга	Мережа живлення устаткування	Нормальний режим роботи	Струм $I_h = 0,3 \text{ мА}$ ; Напруга $U_{\text{дот}} < 2 \text{ В}$	ДСТУ ГОСТ 12.1.038:08
II. Шкідливі фізичні фактори					
2	Несприятливе освітлення	Стан систем природного та штучного освітлення	МРОР 0,3-0.5 мм; розряд III; підрозряд «в», фон середній, контраст середній	КПО $D_h^{\text{сум}} = 1,2 \%$ ; освітленість $E_{\text{мін}} = 300 \text{ лк}$	ДБН В.2.5.-28-18
3	Несприятливий мікроклімат: температура (t), відносна вологість (φ), швидкість руху (v)	Стан систем опалення та вентиляції	Категорія важкості робіт Ia; холодний період	Оптимальні: t – 22-24 °C; φ – 40-60 %; v – не більше 0,1 м/с	ДСН 3.3.6.042-99
4	Підвищений рівень шуму,	Кондиціонери кулери, системи освітлювання, перетворювачі напруги, принтери	Творча діяльність, програмування	Рівень звуку $L_A = 50 \text{ дБА}$	ДСН 3.3.6.037-99
5	Вібрація	»—«	Загальна технологічна, категорія 3, тип «в», умови комфорту	Рівень віброшвидкості $L_V = 75 \text{ дБ}$	ДСТУ ГОСТ 12.1.012:08 ДСН 3.3.6.039-99
6	Психо-фізіологічна перенапруга	Монотонність праці, розумова напруга, статичність і незручність пози		1 та 2 клас умов праці для напруженості і важкості трудового процесу	ГН 3.3.5-8.6.6.1-2002

Згідно ДСН 3.3.6.042-99, у приміщеннях із значними площами застелених поверхонь передбачаються заходи щодо захисту:

а) від перегрівання при попаданні прямих сонячних променів в теплий період року (орієнтація віконних прорізів схід - захід, улаштування лоджій, жалюзі, сонцезахисних плівок та інше);

б) від радіаційного охолодження – в зимовий (використання стін певної товщини, подвійних стекол).

Робочі місця повинні бути віддалені від стін на відстань не менше 1м.

Визначений в таблиці 5.2 коефіцієнт природного освітлення реалізується через вікна визначеної площини, яка розраховується при проектуванні будівлі, а нормований показник штучного освітлення ( $E_{\text{мін}}$ ) реалізується шляхом встановлення визначеної кількості світильників і вибором потужності ламп в них.

Згідно вимог ДСанПіН 3.3.2.007-98, в разі штучного освітлення як джерела світла мають застосовуватись переважно люмінісцентні лампи типу ЛБ і світильники серії ЛПОЗб із дзеркальними ґратами, укомплектовані високочастотними пускорегулювальними апаратами (ВЧ ПРА).

Система загального освітлення має становити суцільні або преривчасті лінії світильників, розташовані збоку від робочих місць (переважно ліворуч), паралельно лінії зору працюючих. Слід передбачити обмеження прямої блискості від джерел природного та штучного освітлення та обмежувати відбиту блискість на робочих поверхнях. Необхідно чистити вікна і світильники не менше двох разів на рік та вчасно замінювати перегорілі лампи.

Заходи захисту від шуму та вібрації повинні відповідати вимогам ГОСТ 12.1.029-80 і ДСТУ ГОСТ 2656885:2009. Устаткування, що є джерелом шуму, слід розташовувати поза приміщенням для роботи з ЕОМ.

Для забезпечення допустимих рівнів шуму у виробничих приміщеннях слід застосовувати засоби звукопоглинання, наприклад, перфоровані плити, панелі, підвісні стелі.

Як захист від шуму, який створюється вентиляторами системних блоків, використовується звукоізоляційний корпус. Вентилятор можна замінити на більш якісний або на мідні радіатори з водяним охолодженням. Крім того встановлюють перехідник з регулятором напруги і швидкості обертання процесорного кулера, а при монтажі кулерів металеві гвинти замінюють гумовими пробками, що дозволяють ізолювати вентилятор від корпусу. Якщо принтер розташований на твердій поверхні, то для зменшення вібрації потрібно підстелити під нього щільний прогумований килимок.

### **5.3 Електробезпека**

Персональна ЕОМ є однофазним споживачем електроенергії, який живиться від трифазної чотирьох провідної мережі перемінного струму напругою 380/220В частотою 50Гц з глухо заземленою нейтраллю.

У разі випадкового дотику до струмопровідних частин, що знаходяться під напругою, або появі напруги дотику на металевих кожухах електроустаткування, наприклад, при пошкодженні ізоляції можливі нещасні випадки в результаті дії електричного струму

Клас пожежа небезпечної зони приміщення, згідно ПУЕ та НПАОП 40.1-1.32-01, – П-Па, бо у приміщенні знаходяться тверді спаленні матеріали.

Для приміщень з підвищеною небезпекою поразки людини електричним струмом ПУЕ передбачені конструктивні, схемно-конструктивні й експлуатаційні міри електробезпеки (ДСТУ ГОСТ 7237:2011).

1) Експлуатаційні міри. Необхідно дотримуватися правил безпеки при роботі з високою напругою і використовувати наступні запобіжні

заходи, що передбачені НПАОП 0.00-1.28-10: не підключати і не відключати кабелю, якщо обладнання знаходиться під напругою; технічне обслуговування і ремонтні роботи виконувати тільки при вимкненому живленні в мережі; встановлювати у приміщенні загальний вимикач для відключення електроустаткування. Забороняється залишати працюючу апаратуру без нагляду.

2) Конструктивні заходи. ЕОМ відноситься до електроустановок до 1000 В закритого виконання, усі струмоведучі частини знаходяться в кожухах. Вибираємо ступінь захисту оболонки від зіткнення персоналу із струмоведучими частинами усередині захисного корпусу і від потрапляння води усередину корпусу IP-44, де перша "4" - захист від твердих тіл, розміром більш 1.0 мм, друга "4"-захист від бризків води (ГОСТ 14254-96).

3) Як схемно-конструктивна міра безпеки застосовується подвійна ізоляція (для монітору), малі напруги до 42 В, занулення (так як мережа живлення до 1000 В з глухо заземленою нейтраллю). Відповідно ДСТУ ГОСТ 7237:2011, занулення – це навмисне електричне з'єднання металевих неструмоведучих частин комп'ютера, які у випадку аварії можуть виявитися під напругою, з нульовим захисним провідником.

Занулення використовується в чотири провідних трифазових мережах із заземленою нейтраллю напругою до 1000 В.

Розрахунок занулення виконано згідно з вимогами методичних вказівок[18].

Мета розрахунку – визначення такого перерізу нульового захисного провідника, при якому струм короткого замикання ( $I_K$ ) у задане число разів ( $K$ ) перевищить номінальний струм апарату захисту ( $I_{\text{ном}}^{A3}$ ), що забезпечить селективне відключення споживача, тобто повинна виконуватися умова:

$$I_K \geq K \cdot I_{\text{ном}}^{A3} \quad (5.1)$$

### Вихідні дані для розрахунку

- 1)  $P_1$  - потужність однофазового споживача електроенергії, наприклад, електронно-обчислювальної машини (ЕОМ), 350Вт;
- 2)  $P_2$  - потужність усіх споживачів, які живляться від цього фазового провідника (кондиціонери, вентилятори, освітлювальні прилади, інші ЕОМ, принтери, тощо), 2кВт;
- 3)  $l_1$  - довжина ділянки 1, 20м (до 100 м);
- 4)  $l_2$  - довжина ділянки 2, 129м (до 500 м).
- 5)  $U_L$ - лінійна напруга;  $U_L=380$  В;  $U_\phi$ - фазова напруга;  $U_\phi=220$  В
- 6) Матеріал проводів (алюміній);
- 7) Спосіб прокладки проводів на ділянці 1-2. На ділянці 2 кабель пролягає у землі, на першій – в металевих трубах.

### Послідовність розрахунків

#### 1 Вибір запобіжника

1.1 Визначення струму  $I_1$ , що живить електроустановку (ЕУ) потужністю  $P_1$ , Вт:

$$I_1 = \frac{P_1}{U_\phi} = \frac{350}{220} = 1,59 \text{ А}, \quad (5.2)$$

1.2 Визначення пускового струму  $I_{\text{пуск}}$  ЕУ потужністю  $P_1$ , Вт:

$$I_{\text{пуск}} = \frac{K_n}{K_T} I_1 = \frac{3}{2,5} \cdot 1,59 = 1,9 \text{ А}, \quad (5.3)$$

де  $K_n$  - коефіцієнт кратності пускового струму;

$K_T$  - коефіцієнт важкості пуску, залежить від часу пуску;

$K_T = 1,6$ ; якщо час пуску понад 10 с - тяжкий пуск;

$K_T = 2$ ; якщо час пуску дорівнює 10 с - середній пуск;

$K_T = 2,5$ ; якщо час пуску дорівнює 5 с - легкий пуск.

Для ЕОМ :  $K_n = 3$ ;  $K_T = 2,5$ .



### 1.3 Вибір апарата захисту

Номінальний струм, при якому спрацьовує апарат захисту, повинен перевищувати  $I_{\text{пуск}}$ , інакше апарат захисту буде спрацьовувати при кожному вмиканні електроустановки.

В нашому випадку  $I_{\text{ном}}^{A3}$  дорівнює 4 А, тому обираємо запобіжник ВПШ 6-12.

## 2 Визначення струму короткого замикання фази на корпус ЕУ

2.1 Струм короткого замикання  $I_k$  визначаємо за формулою (5.4):

$$I_k = \frac{U_\phi}{\frac{Z_{\text{ТР}}}{3} + Z_{\text{пфн}}}, \quad (5.4)$$

де:  $Z_{\text{ТР}}$  - повний опір трансформатора, Ом;

$Z_{\text{пфн}}$  - повний опір петлі фаза-нуль, Ом.

### 2.2 Визначення повного опору трансформатора

Величина  $Z_{\text{ТР}}$  залежить від потужності трансформатора, конструктивного виконання, напруги і схеми з'єднання його обмоток (зіркою або трикутником).

Потужність трансформатора визначається за умовою:

$$N_{\text{ТР}} = 4 \cdot P_2 = 4 \cdot 2 = 8 \text{ кВт}, \quad (5.5)$$

Отже  $Z_{\text{ТР}}$  дорівнює 3,110 Ом, так як  $N_{\text{ТР}}=8$  кВт.

### 2.3 Визначення повного опору петлі фаза-нуль

Повний опір петлі фаза-нуль визначається по формулі:

$$Z_{\text{пфн}} = \sqrt{(R_\phi + R_{\text{нз}})^2 + X^2}, \text{ Ом}, \quad (5.6)$$

де:  $R_{\Phi}$ ,  $R_{H3}$  – активні опори фазового і нульового захисного провідників, відповідно, Ом;

$X$  – індуктивний опір петлі фаза-нуль, який визначається за формулою:

$$X = X_{\Phi} + X_{H3} + X_{B3}, \text{ Ом} \quad (5.7)$$

де:  $X_{\Phi}$ ,  $X_{H3}$  – внутрішні індуктивні опори фазового і нульового провідників, відповідно, Ом;

$X_{B3}$  – зовнішній індуктивний опір, який зумовлено взаємоіндукцією петлі фаза-нуль, Ом.

Для мідних та алюмінієвих провідників  $X_{\Phi}$  та  $X_{H3}$  порівняно малі (близько 0,0156 Ом/км), тому ними можна знехтувати.

Зовнішній індуктивний опір  $X_{B3}$  залежить від відстані між проводами  $D$  та їхнього діаметру  $d$ . Якщо нульові захисні проводи прокладають спільно з фазовими, значення  $D$  мале й порівняльне з діаметром  $d$ , тому опір  $X_{B3}$  незначний (не більш 0,1 Ом/км) і ним можна знехтувати. Тоді

$$Z_{\Pi\Phi H} = R_{\Phi} + R_{H3}, \quad (5.8)$$

Таким чином формула (5.4) приймає вид:

$$I_K = \frac{U_{\Phi}}{\frac{Z_{TP}}{3} + R_{\Phi} + R_{H3}}, \text{ А} \quad (5.9)$$

#### 2.4 Визначення активного опору фазового провідника:

$$R_{\Phi} = R_{\Phi 1} + R_{\Phi 2}, \quad (5.10)$$

де  $R_{\Phi 1}, R_{\Phi 2}$  – опір фазового провідника на ділянках 1 та 2, відповідно, Ом.

Для провідників з кольорових металів:

$$R_{\Phi 1} = \rho \cdot l_1 / S_{\Phi 1}, \text{Ом} \quad (5.11)$$

$$R_{\Phi 2} = \rho \cdot l_2 / S_{\Phi 2}, \text{Ом} \quad (5.12)$$

де  $\rho$  – питомий опір,  $\frac{\text{Ом} \cdot \text{мм}^2}{\text{м}}$ , який дорівнює для алюмінію 0,028;

$S_{\Phi 1}, S_{\Phi 2}$  – перерізи фазового провідника для ділянок 1 та 2, відповідно,  $\text{мм}^2$ .

Перерізи фазових проводів визначають при проектуванні електричної мережі струму, умов прокладання кабелю, матеріалу провідників і т.п. (таблиця 5.3).

Для ділянки 1 вибираємо переріз, який відповідає струму  $I_1$ , для ділянки 2 – струму  $I_2$ . В нашому випадку переріз для ділянки 1 є  $2 \text{ мм}^2$ , а для другої –  $4 \text{ мм}^2$ . Тому:

$$R_{\Phi 1} = \rho \cdot l_1 / S_{\Phi 1} = \frac{0,028 \cdot 20}{2,5} = 0,22 \text{ Ом}$$

$$R_{\Phi 2} = \rho \cdot l_2 / S_{\Phi 2} = \frac{0,028 \cdot 129}{2,5} = 1,44 \text{ Ом}$$

$$R_{\Phi} = R_{\Phi 1} + R_{\Phi 2} = 1,66 \text{ Ом}$$

Струм  $I_2$  визначаємо за формулою:

$$I_2 = \frac{P_2}{U_{\Phi}} = \frac{2000}{220} = 9,09 \text{ А}, \quad (5.13)$$

## 2.5 Визначення опору нульового захисного провідника

$$R_{H3} = R_{H31} + R_{H32}, \text{ Ом}, \quad (5.14)$$

де  $R_{H31}$ ,  $R_{H32}$  – опір нульового захисного провідника на ділянках 1 та 2, відповідно, Ом.

Згідно НПАОП 0.00-1.28-10, площа перерізу нульового робочого та нульового захисного провідників в груповій три провідній мережі повинна бути не менш площі фазового провідника, тобто:

$$S_{H31} = S_{\Phi1}, S_{H32} = S_{\Phi2}$$

Відповідно,  $R_{H3} = R_{\Phi}$ .

Отже:

$$I_K = \frac{U_{\Phi}}{\frac{Z_{TP}}{3} + R_{\Phi} + R_{H3}} = \frac{220}{\frac{3,11}{3} + 1,66 + 1,66} = 50,49 \text{ А}$$

3 Перевірка виконання умов надійності та ефективності роботи занулення

3.1 Повинно виконуватися співвідношення (5.1):

$$I_K \geq K \cdot I_{НОМ}^{A3}$$

$$50,49 \geq 3 \cdot 4 \text{ А}$$

де  $K$  – запас надійності;  $K = 3$  – для запобіжників.

3.2 Утрати напруги на ділянках 1 та 2 не повинні перебільшувати 22 В:

$$U_{П1} + U_{П2} \leq 22 \text{ В}, \quad (5.16)$$

$$U_{П1} = I_1 \cdot R_{\Phi1}, \quad (5.17)$$

$$U_{П2} = I_2 \cdot R_{\Phi2}, \quad (5.18)$$

В нашому випадку:

$$U_{П1} = I_1 \cdot R_{Ф1} = 0,34 \text{ В}$$

$$U_{П2} = I_2 \cdot R_{Ф2} = 13,08 \text{ В}$$

Отже обидві умови виконуються.

В результаті розрахунку у якості запобіжника було обрано запобіжник типу ВПШ 6-12, а перерізи фазового і нульового захисного провідників дорівнюють 2.5 мм на ділянках 1 та 2.

#### **5.4 Пожежна безпека**

У зв'язку з поширенням комп'ютерної техніки, що може привести до загоряння, треба передбачати можливі наслідки і розробляти заходи щодо їх попередження. Причинами загоряння стають: несправність електричного обладнання, пошкодження ізоляції, коротке замикання кола струму, перегрів проводів, поганий контакт в місцях з'єднання; розряди статичної електрики, які особливо небезпечні в вибухонебезпечних приміщеннях, блискавка.

Пожежна безпека забезпечується наступними мірами:

- 1) системою запобігання пожеж;
- 2) системою пожежного захисту;
- 3) організаційними заходами щодо пожежної безпеки;

Система запобігання пожеж передбачає запобігання утворенню пального середовища і запобігання утворенню в пальному середовищі джерел запалювання.

Для зменшення небезпеки утворення в пальному середовищі джерел запалювання передбачено:

- 1) використання електроустаткування, що відповідає класу пожежа небезпечної зони приміщення П-Па за ПУЕ та НПАОП 40.1-1.32-

01: ступінь захисту електроапаратури не менш IP-44, ступінь захисту світильників IP-2X (згідно НАПБ А.01.001-2014);

2) забезпечення захисту від короткого замикання (контроль і профілактика ізоляції, використання запобіжників);

3) вибір перетину провідників по максимально допустимому нагріванню.

4) будівлі, в яких встановлено обладнання інформаційних технологій чи будь-яке інше електронне обладнання, чутливе до атмосферних перешкод, незалежно від кількості уражень об'єктів за рік потребує I або II рівня блискавка захисту (ДСТУ БВ.2.5-38:2008).

Система протипожежного захисту призначена для локалізації та гасіння пожежі. При виборі засобів гасіння пожежі для забезпечення безпеки людини від можливості поразки електричним струмом у приміщенні відповідно вимог НАПБ А.01.001-2014 передбачено використання вуглекислотних вогнегасників ВВК-5. Вогнегасник знаходиться на видному і легко доступному місці. При виникненні пожежі передбачені можливості аварійного відключення апаратури і комунікацій та повідомлення в пожежну охорону по телефону. У якості сповіщувачів використовуються система автоматичної пожежної сигналізації відповідно вимог НАПБ Б.06.004-2005. Ступінь вогнестійкості будинку II, що відповідає вимогам НПАОП 0.00-1.28-2010, згідно яких комп'ютери повинно розташовувати в будівлях не нижче II ступеню (ДБН В.1.1-7-2016). У приміщенні є два незалежних виходи для евакуації людей під час пожежі.

Організаційними заходами протипожежної профілактики є вступний інструктаж при надходженні на роботу, навчання виробничого персоналу протипожежним правилам, видання необхідних інструкцій і плакатів, засобів наочної агітації, наявність плану евакуації.

### **5.5 Охорона навколишнього середовища**

Проблема охорони й оптимізації навколишнього природного середовища виникла як неминучий наслідок сучасної промислової революції.

Збільшення використання енергії призводить до порушення екологічної рівноваги природного середовища, яке складалася століттями.

Поряд з цим, підвищення технічної оснащеності підприємств, застосування нових матеріалів, конструкцій і процесів, збільшення швидкостей і потужностей виробничих машин впливають на навколишнє середовище.

Основними задачами Закону України "Про охорону навколишнього природного середовища", прийнятого 25 червня 1991 року, є регулювання відносин в області охорони природи, використання і відтворення природних ресурсів, забезпечення екологічної безпеки, попередження і ліквідація наслідків негативного впливу на навколишнє середовище господарської й іншої діяльності людини, збереження природних ресурсів, генетичного фонду, ландшафтів і інших природних об'єктів.

При масовому використанні моніторів та комп'ютерів не можна не враховувати їхній вплив на навколишнє середовище на всіх стадіях – при виготовленні, експлуатації та після закінчення терміну служби.

Міжнародні екологічні стандарти, що діють на сьогоднішній день в усьому світі, визначають набір обмежень до технологій виробництва та матеріалів, які можуть використовуватися в конструкціях пристроїв. Так, за стандартом ТСО-95, вони не повинні містити фреонів (турбота про озоновий шар), полівінілхлориді, бромідів (як засобів захисту від загоряння).

У стандарті ТСО-99 закладене обмеження за кадмієм у світлочутливому шарі екрана дисплея та ртуті в батареях; є чіткі вказівки відносно пластмас, лаків та покриттів, що використовуються.

Поверхня кнопок не повинна містити хром, нікель та інші матеріали, які визивають алергічну реакцію. ГДК пилу дорівнює  $0,15 \text{ мг/м}^3$ , рекомендовано  $0,075 \text{ мг/м}^3$ ; ГДК озону під час роботи лазерного принтеру –  $0,02 \text{ мг/м}^3$ . Особливо жорсткі вимоги до повторно використовуваних матеріалів.

Міжнародні стандарти, починаючи з ТСО-92, включають вимоги зниженого енергоспоживання та обмеження припустимих рівнів потужності, що споживаються у неактивних режимах.

Дотримання приведених нормативних параметрів небезпечних і шкідливих виробничих факторів дозволить забезпечити більш здорові і безпечні умови роботи користувача ЕОМ.

### **5.6 Висновки до розділу**

У даному розділі дипломної роботи було проведено аналіз умов праці на робочому місці, було оглянуто основні заходи безпеки на робочому місці, також було розраховано діаметр заземлення ЕОМ для безпечної праці.



## **6 ЦИВІЛЬНИЙ ЗАХИСТ**

### **6.1 Система державного управління у сфері Цивільного захисту**

Розділ виконано відповідно методичним вказівкам «Охорона праці та навколишнього середовища» [18].

Згідно кодексу цивільного захисту України [19], забезпечення реалізації державної політики у сфері цивільного захисту здійснюється єдиною державною системою цивільного захисту, яка складається з функціональних і територіальних підсистем та їх ланок.

Положення про єдину державну систему цивільного захисту, типові положення про функціональну і територіальну підсистеми затверджуються Кабінетом Міністрів України.

Основними завданнями єдиної державної системи цивільного захисту є забезпечення готовності міністерств та інших центральних та місцевих органів виконавчої влади, органів місцевого самоврядування, підпорядкованих їм сил і засобів до дій, спрямованих на запобігання і реагування на надзвичайні ситуації; забезпечення реалізації заходів щодо запобігання виникненню надзвичайних ситуацій; навчання населення щодо поведінки та дій у разі виникнення надзвичайної ситуації; виконання державних цільових програм, спрямованих на запобігання надзвичайним ситуаціям, забезпечення сталого функціонування підприємств, установ та організацій, зменшення можливих матеріальних втрат; опрацювання інформації про надзвичайні ситуації, видання інформаційних матеріалів з питань захисту населення і територій від наслідків надзвичайних ситуацій; прогнозування і оцінка соціально-економічних наслідків надзвичайних ситуацій, визначення на основі прогнозу потреби в силах, засобах, матеріальних та фінансових ресурсах; створення, раціональне збереження і використання резерву матеріальних та фінансових ресурсів, необхідних для запобігання і реагування на надзвичайні ситуації; оповіщення

населення про загрозу та виникнення надзвичайних ситуацій, своєчасне та достовірне інформування про фактичну обстановку і вжиті заходи; захист населення у разі виникнення надзвичайних ситуацій; проведення рятувальних та інших невідкладних робіт щодо ліквідації наслідків надзвичайних ситуацій, організація життєзабезпечення постраждалого населення; пом'якшення можливих наслідків надзвичайних ситуацій у разі їх виникнення; здійснення заходів щодо соціального захисту постраждалого населення; реалізація визначених законом прав у сфері захисту населення від наслідків надзвичайних ситуацій, в тому числі осіб (чи їх сімей), що брали безпосередню участь у ліквідації цих ситуацій; інші завдання, визначені законом.

Функціональні підсистеми єдиної державної системи цивільного захисту

Функціональні підсистеми єдиної державної системи цивільного захисту (далі - функціональні підсистеми) створюються центральними органами виконавчої влади у відповідній сфері суспільного життя.

Положення про функціональні підсистеми розробляються на підставі типового положення про таку підсистему і затверджуються центральними органами виконавчої влади, що їх створили, за погодженням із центральним органом виконавчої влади, який забезпечує формування та реалізує державну політику у сфері цивільного захисту.

У разі якщо діяльність центральних органів виконавчої влади спрямовується і координується Кабінетом Міністрів України через відповідного міністра, зазначені положення затверджуються такими міністрами за погодженням із центральним органом виконавчої влади, який забезпечує формування та реалізує державну політику у сфері цивільного захисту.

Перелік центральних органів виконавчої влади, що створюють функціональні підсистеми, визначається положенням про єдину державну систему цивільного захисту.

Безпосереднє керівництво функціональною підсистемою покладається на керівника органу, суб'єкта господарювання, що створив таку підсистему.

До складу функціональних підсистем входять органи управління та підпорядковані їм сили цивільного захисту, відповідні суб'єкти господарювання, які виконують завдання цивільного захисту.

Територіальні підсистеми єдиної державної системи цивільного захисту та їх ланки

Територіальні підсистеми єдиної державної системи цивільного захисту (далі - територіальні підсистеми) діють в областях, містах Києві та Севастополі.

Положення про територіальні підсистеми розробляються на підставі типового положення про таку підсистему і затверджуються відповідно Радою міністрів чи місцевими державними адміністраціями за погодженням із центральним органом виконавчої влади, який забезпечує формування та реалізує державну політику у сфері цивільного захисту.

Ланки територіальних підсистем створюються районними у містах Києві та Севастополі державними адміністраціями - у районах, районах у містах Києві та Севастополі; органами місцевого самоврядування - в обласних центрах, у містах обласного і районного значення. Положення про ланку територіальної підсистеми затверджується органом, що її створив.

Безпосереднє керівництво територіальною підсистемою, її ланкою покладається на посадову особу, яка очолює орган, що створив таку підсистему, ланку.

Безпосереднє керівництво територіальною підсистемою Автономної Республіки Крим покладається на Раду міністрів Автономної Республіки Крим.

До складу територіальних підсистем та їх ланок входять органи управління та підпорядковані їм сили цивільного захисту, відповідні суб'єкти господарювання.

## **6.2 Висновки до розділу**

Таким чином забезпечення реалізації державної політики у сфері цивільного захисту здійснюється єдиною державною системою цивільного захисту, яка складається з функціональних і територіальних підсистем та їх ланок. Положення про єдину державну систему цивільного захисту, типові положення про функціональну і територіальну підсистеми затверджуються Кабінетом Міністрів України. Єдина державна система цивільного захисту виконує дуже багато важливих для суспільства функцій.

## ВИСНОВКИ

В результаті виконання дипломної роботи були розглянуті методи автоматичного побудування згорткових нейронних мереж, для вирішення задачі класифікації зображень.

Було розроблено програмне забезпечення, метою якого є пошук оптимальної архітектури згорткових мереж та також її навчання. Для цього на мові програмування Python, за допомогою загальнодоступних бібліотек Keras, Numpy та TensorFlow був реалізований алгоритм побудови архітектури на основі градієнтного спуску на даних CIFAR-10. Результати були проаналізовані.

Поставлена задача була реалізована. Отриману програму можна використовувати для побудування інших згорткових мереж для класифікації будь-яких зображень.

## СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

- 1 Zoph Barret, Vasudevan Vijay, Shlens Jonathon, and Quoc V Le. Learning transferable architectures for scalable image recognition. // <https://arxiv.org/pdf/1707.07012.pdf>, 2017.
- 2 Esteban Real, Aggarwal Alok, Huang Yanping, and Quoc V Le. Regularized evolution for image classifier architecture search. // <https://arxiv.org/pdf/1802.01548.pdf>, 2018.
- 3 Negrinho Renato and Gordon Geoff. Deeparchitect: Automatically designing and training deep architectures. // <https://arxiv.org/pdf/1704.08792.pdf>, 2017.
- 4 Liu Chenxi, Zoph Barret, Shlens Jonathon, Wei Hua, Li-Jia Li, Li Fei-Fei, Yuille Alan, Huang Jonathan, and Murphy Kevin. Progressive neural architecture search. // <https://arxiv.org/pdf/1712.00559.pdf>, 2017a.
- 5 Krizhevsky Alex. Learning multiple layers of features from tiny images, 2009.
- 6 Srivastava Nitish, Hinton Geoffrey, Krizhevsky Alex, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15 (1), 211-252 c.
- 7 Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. // <https://arxiv.org/pdf/1409.1556.pdf>, 2015.
- 8 Sergey Ioffe, Christian Szegedy. Batch normalization: Accelerating deep network training by reducing the internal covariate shift. // <https://arxiv.org/pdf/1502.03167.pdf>, 2015.
- 9 Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778c, 2016.
- 10 Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In Machine Learning, 1992.

11 Zoph B., Vasudevan V., Shlens J., and Q. V. Le. Learning transferable architectures for scalable image recognition. In CVPR, 2018.

12 Pham Hieu, Melody Y Guan, Zoph Barret, Quoc V Le, and Jeff Dean. Authors' implementation of "Efficient Neural Architecture Search via Parameter Sharing. \\\ <https://github.com/melodyguan/enas/tree/2734eb2657847f090e1bc5c51c2b9cbf0be51887>, 2018a.

13 Zoph Barret and Quoc V Le. Neural architecture search with reinforcement learning. // <https://arxiv.org/pdf/1611.01578.pdf>, 2016.

14 G Anandalingam and TL Friesz. Hierarchical optimization: An introduction. Annals of Operations Research, 34(1):1–11, 1992.

15 Finn Chelsea, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. // <https://arxiv.org/pdf/1703.03400.pdf>, 2017.

16 Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. // <https://arxiv.org/pdf/1602.07261.pdf>, 2016.

17 Методичні вказівки для розробки розділу «Охорона праці та навколишнього середовища» у випускних дипломних роботах студентів інженерно – фізичного факультету та факультету «Інформатика і управління» очної та заочної форм навчання/Уклад. Березуцький В.В., Кузьменко О.О. , Латишева М.М. – Харків: НТУ «ХПІ», 2012. – 60 с. – Укр. мовою.

18 Методичні вказівки до виконання розділу «Цивільний захист» у дипломних проектах спеціаліста та магістра для студентів усіх спеціальностей /уклад.: Любченко І. М., Мягкий В. О., Твердохлєбова Н. Є., Толстоусова О. В. – Х. : НТУ «ХПІ», 2013. – 48 с. - Укр.

19 Кодекс Цивільного захисту України. ВРУ від 02.10.2012 р. №5403-VI. Редакція 2018 р.

20 Цивільний захист: навчальний посібник /Бахарева Г. Ю., Твердохлебова Н. Є., Любченко І. М., Гуренко І. В. та ін. – Харків: НТУ «ХП», 2015. – 115 с.