

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

**Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:**

**КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ
НЕЙРОННИХ МЕРЕЖ**

Виконав студент 4-го курсу
Владислав ДУХ

(підпис)

Науковий керівник:
Завідувач кафедри
доктор фіз.-мат. наук, професор
Юрій КРАК

(підпис)

Засвідчую, що в цій роботі немає запозичень з
праць інших авторів без відповідних посилань

Студент

(підпис)

Роботу розглянуто й допущено до захисту на
засіданні кафедри теоретичної кібернетики
« _____ » _____ 2023р.,

протокол № _____

Завідувач кафедри
доктор фіз.-мат. наук, професор
Юрій КРАК

(підпис)

КИЇВ - 2023

РЕФЕРАТ

Обсяг роботи складає 43 сторінок, в ній містяться 14 ілюстрацій, використано 18 джерел посилань.

РУТНОН, ШТУЧНИЙ ІНТЕЛЕКТ, НЕЙРОННІ МЕРЕЖІ, ФУНКЦІЯ ЗГОРТКИ, ФУНКЦІЯ АКТИВАЦІЇ, ФУНКЦІЯ ПУЛІНГУ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, КЛАСИФІКАЦІЯ РУКОПИСНИХ ЦИФР, ДАТАСЕТИ.

Метою даного дослідження є дослідження та розробка системи класифікації зображень за допомогою згорткових нейронних мереж. Конкретні цілі дослідження включають:

- Вивчення теоретичних основ згорткових нейронних мереж та їх застосування в класифікації зображень.
- Розробка та реалізація архітектури згорткової нейронної мережі для класифікації зображень.
- Аналіз різних операцій та функцій активації, використовуваних у згорткових нейронних мережах, та їх впливу на результати класифікації.
- Застосування розробленої моделі на датасеті CIFAR-10 та оцінка її точності та продуктивності.
- Порівняння результатів отриманої моделі з іншими методами класифікації зображень.
- Визначення перспектив та можливих напрямків подальшого розвитку системи класифікації зображень на основі згорткових нейронних мереж.

Метою дослідження є розширення розуміння та використання згорткових нейронних мереж для класифікації зображень, а також створення ефективної та точної моделі, яка може бути застосована в різних сферах, де класифікація зображень відіграє важливу роль.

Об'єктом дослідження є згорткові нейронні мережі (Convolutional Neural Networks, CNNs) та їх застосування у задачі класифікації зображень. Дослідження спрямоване на розуміння принципів роботи згорткових нейронних мереж, їх архітектури, операцій та функцій активації, які використовуються для обробки зображень.

В рамках дослідження буде проаналізовано різні аспекти згорткових нейронних мереж, такі як їхній вплив на точність класифікації, продуктивність та ефективність в різних задачах. Об'єктом дослідження є також вплив різних архітектур, операцій та функцій активації на якість класифікації зображень.

Дослідження також включатиме датасети CIFAR-10 та MNIST як об'єкти для класифікації зображень. Об'єкт дослідження включає аналіз результатів, отриманих за допомогою згорткових нейронних мереж на цих датасетах, з метою порівняння точності та продуктивності розроблених моделей.

Дослідження об'єкта включає розробку, реалізацію та експериментальне тестування згорткових нейронних мереж, що дозволяє дослідникам краще зрозуміти принципи роботи цих мереж та їхню ефективність у класифікації зображень.

Предметом дослідження є методи та техніки класифікації зображень за допомогою згорткових нейронних мереж. Дослідження спрямоване на вивчення різних аспектів предмета, включаючи архітектуру згорткових нейронних мереж, операції, функції активації та методи зворотного поширення помилки.

Основною метою дослідження є розробка та оптимізація моделей згорткових нейронних мереж для класифікації зображень. Предметом дослідження є також визначення факторів, що впливають на точність та ефективність цих моделей, таких як розмір та глибина мережі, кількість шарів, параметри оптимізації та інші.

Дослідження предмета включає аналіз різних підходів до класифікації зображень з використанням згорткових нейронних мереж, таких як зміна архітектури, оптимізація гіперпараметрів та використання переднього навчання (pretraining) на великому обсязі даних.

Предмет дослідження також включає порівняння згорткових нейронних мереж з іншими методами класифікації зображень, зокрема з традиційними машинними навчанням та іншими типами нейронних мереж.

Дослідження предмета сприяє розвитку нових і покращених методів класифікації зображень, що мають потенціал застосування у багатьох областях, таких як медична діагностика, розпізнавання об'єктів, безпека, автономні транспортні засоби та багато інших.

ЗМІСТ

| | |
|-------------------------------------------------------------------------------|---------------------------------------|
| РЕФЕРАТ..... | 2 |
| ЗМІСТ | 5 |
| ВСТУП..... | 6 |
| РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ | 8 |
| 1.1 Згорткові нейронні мережі: визначення та принцип роботи | 8 |
| 1.2 Застосування згорткових нейронних мереж у класифікації зображень .. | 9 |
| 1.3 Архітектура згорткових нейронних мереж і математична модель..... | 10 |
| 1.4 Операції у згорткових нейронних мережах | 144 |
| 1.5 Функції активації і згортки | 166 |
| 1.6 Зворотне поширення помилки у згорткових нейронних мережах | 177 |
| РОЗДІЛ 2. ПІДГОТОВКА ТА ОБРОБКА ДАНИХ..... | 199 |
| 2.1 Збір та підготовка датасету..... | Error! Bookmark not defined. 9 |
| 2.2 Розбиття датасету на тренувальну, перевірочну та тестову вибірки | 2Error! Bookmark not defined. |
| РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА НАВЧАННЯ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ..... | 233 |
| 3.1 Вибір архітектури згорткової нейронної мережі..... | 233 |
| 3.2 Побудова та ініціалізація моделі | 244 |
| 3.3 Навчання та оптимізація моделі..... | 244 |
| 3.4 Оцінка продуктивності та точності моделі..... | 255 |
| РОЗДІЛ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ І РЕЗУЛЬТАТИ ВИКОНАННЯ КОДУ | 237 |
| 4.1 Загальна інформація..... | 237 |
| 4.2 Класифікація зображень за допомогою CIFAR-10..... | 239 |
| 4.3 Класифікація рукописних цифр за допомогою MNIST | 31 |
| ВИСНОВКИ | 33 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ | 37 |
| ДОДАТКИ | 39 |

ВСТУП

Загальний огляд

Класифікація зображень є важливою задачею у багатьох областях, включаючи комп'ютерний зір, медичну діагностику, автономні системи, розпізнавання образів та багато інших. Вона полягає у визначенні класу або категорії, до якої належить конкретне зображення. Ця задача викликає великий інтерес у наукових та промислових галузях через свої потенційні застосування..

Згорткові нейронні мережі (ЗНМ) є потужними моделями машинного навчання для класифікації зображень. Вони використовують принципи згортки та пулінгу для ефективного виявлення ознак у зображеннях. ЗНМ використовуються у багатьох успішних системах розпізнавання образів, зокрема у розпізнаванні облич, автоматичному водінні автомобілів, системах безпеки та інших.

Актуальність теми

Проблема класифікації зображень є однією з ключових задач у сфері комп'ютерного зору та машинного навчання. Швидкий розвиток технологій у сфері обробки зображень та зростання обсягу цифрової інформації створюють потребу в ефективних та точних методах автоматичної класифікації зображень.

Згорткові нейронні мережі (Convolutional Neural Networks, CNNs) виявилися потужним інструментом для розв'язання задач класифікації зображень. Вони можуть автоматично виконувати процес виявлення особливостей та характеристик зображень, навчаючись на великому обсязі даних.

Актуальність теми полягає в тому, що згорткові нейронні мережі продовжують розвиватися та вдосконалюватися, щоб покращити точність

класифікації та здатність до розпізнавання складних зразків. Вони застосовуються в таких областях, як комп'ютерний зір, розпізнавання об'єктів, медична діагностика, автономні транспортні засоби та багато інших.

Розробка та вдосконалення методів класифікації зображень на основі згорткових нейронних мереж має великий потенціал у вирішенні реальних проблем і покращенні якості роботи систем, які використовують аналіз зображень. Дослідження в цій області сприяє подальшому розвитку і використанню інтелектуальних систем, що здатні автоматично розпізнавати та класифікувати зображення з високою точністю.

Актуальність даної теми полягає в її потенціалі вирішення практичних завдань та забезпечення розвитку нових застосувань у різних галузях, де класифікація зображень має велике значення.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

1.1 Згорткові нейронні мережі: визначення та принцип роботи

Згорткові нейронні мережі (Convolutional Neural Networks, CNNs) є потужними і ефективними інструментами для обробки та аналізу зображень. Вони були спеціально розроблені для впорядкування та розпізнавання образів і демонструють вражаючі результати в багатьох класифікаційних завданнях.

Згорткові нейронні мережі базуються на принципі згортки та пулінгу. Згортка використовується для локального зв'язування вхідних даних з фільтрами, які відповідають за виявлення певних ознак у зображеннях. Кожен фільтр має свої ваги, які використовуються для відшукування відповідних ознак у різних областях зображення.

Давайте розглянемо приклад згорткового шару. Припустимо, що у нас є вхідне зображення розміром 32x32 пікселі. Застосування згортки може допомогти виявити границі, форми або текстури на цьому зображенні. Нехай у нас є фільтр розміром 3x3, який шукає горизонтальні границі. Згортка полягає у переміщенні цього фільтра по всьому зображенню і обчисленні значень на основі перекриття фільтра з пікселями зображення. Результатом згортки буде нове зображення, в якому границі будуть виділені.

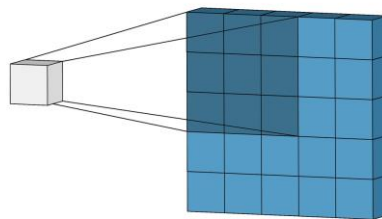


Рис 1.1 процес згортки та виявлення границь. [4]

Після згортки зазвичай застосовують функцію активації, таку як ReLU (Rectified Linear Unit), для введення нелінійності. Це допомагає зберегти необхідну гнучкість для моделі.

Далі, після згортки та активації, може бути застосований пулінг, який зменшує розмір представлення, зберігаючи при цьому важливі ознаки. Це допомагає зменшити обчислювальну складність та кількість параметрів мережі.

Таким чином, згорткові нейронні мережі можуть автоматично визначати різні ознаки у зображеннях, забезпечуючи ефективну та точну класифікацію.

1.2 Застосування згорткових нейронних мереж у класифікації зображень

Згорткові нейронні мережі мають широке застосування в класифікації зображень. Вони здатні ефективно визначати та розпізнавати різні об'єкти на зображеннях. Завдяки своїй здатності автоматично вивчати ознаки зображень, згорткові нейронні мережі стали основою багатьох сучасних систем комп'ютерного зору.

Один з ключових кроків у застосуванні згорткових нейронних мереж у класифікації зображень - це тренування мережі на великому наборі зображень з правильно позначеними класами. Під час тренування мережа навчається визначати характеристики та ознаки зображень, які відповідають певним класам. Згорткові шари у мережі виявляють різні локальні ознаки, такі як границі, кути, текстури тощо. Після проходження через кілька шарів мережі, ознаки об'єднуються та агрегуються у вищих рівнях абстракції, що дозволяє класифікувати зображення на основі виявлених ознак.

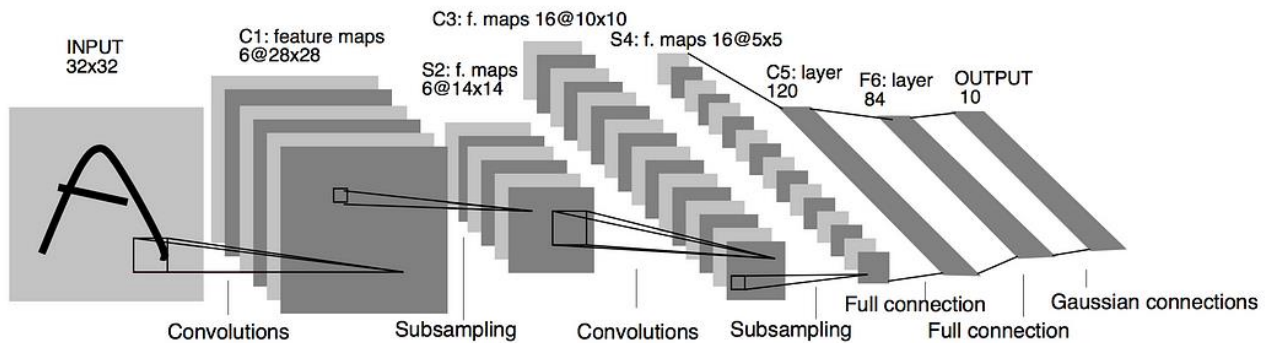


Рис 1.2 Застосування згорткових нейронних мереж у класифікації зображень [5]

Крім того, згорткові нейронні мережі використовуються у багатьох відомих системах комп'ютерного зору, таких як система класифікації зображень ImageNet, де мережі розпізнають тисячі класів об'єктів, а також у системах автоматичної розпізнавання облич, детекції об'єктів та багатьох інших застосуваннях.

Таким чином, згорткові нейронні мережі є потужним інструментом у класифікації зображень, завдяки своїй здатності виявляти ознаки та позначати класи з високою точністю і ефективністю. Використання цих мереж в різних сферах комп'ютерного зору відкриває нові можливості для автоматизації та поліпшення різних завдань обробки зображень.

1.3 Архітектура згорткових нейронних мереж

Архітектура згорткових нейронних мереж включає в себе різні компоненти, які співпрацюють для ефективного виявлення ознак і класифікації зображень. Основні складові архітектури включають згорткові шари, пулінг шари та повнозв'язані шари.

Згорткові шари є основним елементом згорткових нейронних мереж. Кожен згортковий шар складається з набору фільтрів, які ковзають по вхідному зображенню та виконують операцію згортки. Ця операція виявляє локальні ознаки,

такі як границі, кути, текстури тощо. Кількість та розмір фільтрів може варіюватися залежно від архітектури мережі.

Пулінг шари використовуються для зменшення розміру зображення та збереження найбільш суттєвих ознак. Це досягається шляхом вибору максимального значення або середнього значення з певного регіону зображення. Пулінг допомагає зменшити кількість параметрів і обчислювальний обсяг, а також зробити модель більш робастною до малих зміщень об'єктів на зображенні.

Повнозв'язані шари знаходяться в кінці мережі і використовуються для зв'язку виявлених ознак з класами зображень. Кожен нейрон у повнозв'язаному шарі пов'язаний з усіма нейронами попереднього шару. Ці шари використовуються для остаточної класифікації зображень на основі зібраних ознак.

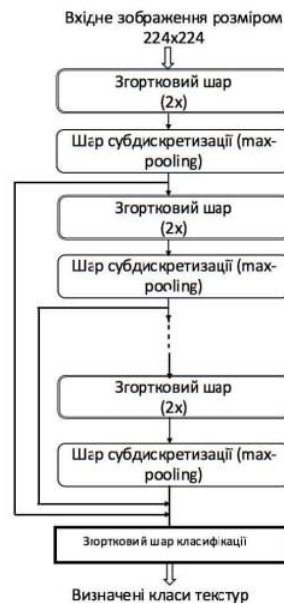


Рис 1.3 Приклад архітектури згорткової нейронної мережі [11]

Архітектура згорткових нейронних мереж може варіюватися залежно від завдання та розміру датасету. Для складних завдань зображення, таких як розпізнавання об'єктів у великому масштабі, використовуються глибокі згорткові нейронні мережі з багатьма шарами. В той же час, для простих завдань можуть використовуватися менш глибокі мережі з меншою кількістю параметрів.

Архітектура згорткових нейронних мереж є важливим елементом успіху у багатьох завданнях комп'ютерного зору, і вона постійно розвивається та вдосконалюється для досягнення кращих результатів у класифікації та розпізнаванні зображень.

Основними елементами архітектури згорткової нейромережі є два модулі, перший відповідає за згортку, тобто обробку зображення певним чином, а другий за класифікацію даних, що отримані з першого модулю. Основними шарами частини мережі, що обробляє зображення, є згортковий шар та шар підвибірки.

Згортковий шар містить у собі нейрони-матриці, на вхід до яких поступово надходять ділянки зображення. Кожен з таких нейронів тренується так, щоб набути здатність до класифікації певної ознаки об'єкту. Для кожної вхідної ділянки зображення нейрон надає відгук, який можна інтерпретувати як міру подібності вхідних даних до шаблону. Такі відгуки формують двовимірний масив, який називається картою ознак. Шар підвибірки приймає ділянку карти ознак та за визначеним правилом формує єдиний відгук для масиву вхідних даних. Наприклад, нейрони підвибіркового шару мережі LeNet5 приймали ділянку 2x2 та розраховували для неї одне середнє значення, що суттєво зменшує розмірність даних. Нейрони підвибіркового шару дозволяють сформувати карту найістотніших ознак. Модулем класифікації є повнозв'язні шари прямого розповсюдження по типу перцептронів.

Модель нейрону повнозв'язного шару має вигляд:

$$\hat{y} = F_{act} \left(G(x_1, x_2, \dots, x_S, w_1, w_2, \dots, w_S) \right),$$

де x_i – елемент вхідних даних; w_i – вага синаптичного зв'язку; G – функція зваженої суми; $G(x_1, x_2, \dots, x_K, w_1, w_2, \dots, w_K) = \sum_{i=1}^S (x_i \cdot w_i)$; $i = \overline{1, S}$; S – кількість синаптичних зв'язків; $F_{act}(x)$ – функція активації нейрону.

Рис 1.4 Модель нейрону повнов'язного шару [3]

У якості функції активації обирається довільна нелінійна функція однієї змінної. В роботі застосовується функція активації ReLU:

$$f(x) = \max(0, x)$$

Особливою функцією активації є функція Softmax, яка враховує суму виходів всіх нейронів шару та набуває вигляду:

$$\hat{y}_i = \frac{e^{x_i}}{\sum_{k=1}^N e^{x_k}},$$

де N – кількість нейронів у шарі; $i, k = \overline{1, N}$.

Рис 1.5 Вигляд функції Softmax [3]

Завдяки властивостям функції Softmax виходи мережі зручно інтерпретувати як ймовірність належності вхідних даних до певних класів.

Для організації процедури тренування мережі необхідно ввести функцію похибки. Враховуючи вибір функції активації Softmax у якості функції похибки шару доцільно використати перехресну ентропію.

$$H(y, \hat{y}) = -\frac{1}{T} \sum_t \sum_t [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)],$$

де y_i – відповідь i -ого нейрону, яка очікується, \hat{y}_i – дійсна відповідь i -ого нейрону, $t=1, T$, T – кількість елементів тренування (для стохастичних методів $T=1$), $i=\overline{1, N}$, N – кількість нейронів у шарі.

Рис 1.6 Функція похибки шару з використанням перехресної ентропії [3]

Модель нейрону згорткового шару має вигляд:

$$y_{ij} = b + \sum_{s=1}^M \sum_{t=1}^M w_{st} x_{(i-M/2+s)(j-M/2+t)},$$

де (i, j) – координати елемента матриці, з яким суміщено центр ядра згортки; y_{ij} – відгук ядра згорткового шару; b – зміщення ядра згорткового шару; M – розмірність матриці згортки; w_{st} – вага синаптичного зв'язку елемента ядра згортки; x – елемент вхідної матриці.

Зауважимо, що для вхідної матриці розміру $a \times b$ $M/2 \leq i \leq a - M/2$; $M/2 \leq j \leq b - M/2$. Тоді розмірність вихідної карти ознак дорівнює $(a - M) \times (b - M)$.

Рис 1.7 Модель нейрону згорткового шару [3]

Підвибірковий шар виконує перетворення у такий спосіб:

$$y_{ij} = b + 1/M^2 \sum_{s=1}^M \sum_{t=1}^M x_{(i-M/2+s)(j-M/2+t)}.$$

Рис 1.8 Перетворення підвибіркового шару [3]

1.4 Операції у згорткових нейронних мережах

Операції у згорткових нейронних мережах є основою для виявлення ознак на зображеннях та подальшої обробки даних. Декілька ключових операцій використовуються для досягнення цього результату. Основні операції включають згортку, активацію та пулінг.

- **Згортка:** Згортка є основною операцією у згорткових нейронних мережах. Вона використовується для виявлення локальних ознак на зображенні. Згортка виконується шляхом переміщення фільтра по всьому зображенню та обчислення скалярного добутку між фільтром і відповідним регіоном зображення. Це допомагає виявити різні ознаки, такі як границі, кути або текстури.
- **Активація:** Після згортки до отриманого результуючого зображення застосовується функція активації. Функція активації дозволяє вводити нелінійність у модель та дозволяє згортковій нейронній мережі моделювати складніші залежності. Популярні функції активації включають ReLU (Rectified Linear Unit), Sigmoid та Tanh.
- **Пулінг:** Пулінг є операцією зменшення розміру зображення. Це допомагає знизити кількість параметрів моделі та зберегти найбільш суттєві ознаки зображення. Найпоширеніші типи пулінгу - це максимальне пулінг, де вибирається максимальне значення з регіону, та середнє пулінг, де обчислюється середнє значення з регіону.

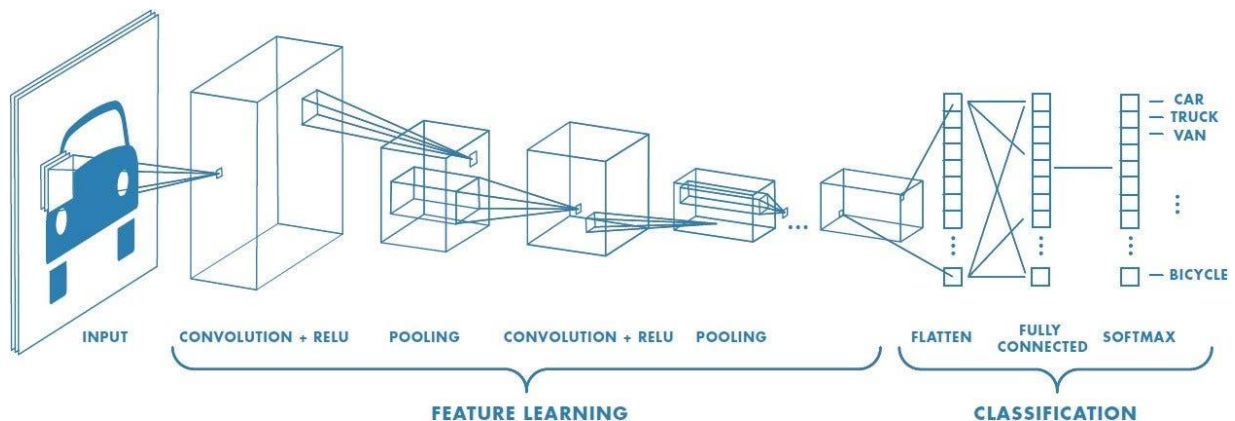


Рис 1.9 Операції в згорткових нейронних мережах [7]

Ці операції спільно допомагають згортковим нейронним мережам ефективно виявляти ознаки на зображеннях та виконувати класифікацію та розпізнавання зображень з високою точністю.

1.5 Функції активації і згортки

Функції активації та згортки є важливими компонентами згорткових нейронних мереж. Вони грають вирішальну роль у виявленні ознак та нелінійних залежностей у вхідних даних.

Функції активації: Функції активації застосовуються після кожного шару згортки або повнозв'язного шару для введення нелінійності у модель. Деякі з популярних функцій активації включають:

- **ReLU (Rectified Linear Unit):** Функція ReLU встановлює вихідний сигнал рівним нулю, якщо вхід менший за нуль, або без змін, якщо вхід більший або рівний нулю. Ця функція є простою у обчисленнях та дозволяє ефективно виконувати нелінійні операції.
- **Sigmoid:** Функція Sigmoid перетворює вхідний сигнал на значення між 0 та 1. Вона використовується, коли потрібно отримати ймовірнісну оцінку або використовувати модель для бінарної класифікації.
- **Tanh:** Функція Tanh подібна до Sigmoid, але вона перетворює вхід на значення між -1 та 1. Вона дозволяє моделі моделювати як позитивні, так і негативні впливи.

Згортка: Операція згортки виконується шляхом переміщення фільтра по зображенню та обчислення скалярного добутку між фільтром і відповідним регіоном зображення. Ця операція допомагає виявити різні ознаки, такі як границі, текстури або форми. Розмір фільтра та крок переміщення впливають на розмір та роздільну здатність вихідного зображення після згортки.

Застосування правильних функцій активації та операцій згортки грають важливу роль у роботі згорткових нейронних мереж та допомагають їм ефективно виявляти ознаки на зображеннях та виконувати завдання класифікації та розпізнавання.

1.6 Зворотне поширення помилки у згорткових нейронних мережах

Зворотне поширення помилки є ключовим алгоритмом для навчання згорткових нейронних мереж. Цей процес дозволяє моделі коригувати свої ваги з метою мінімізації помилки та покращення точності прогнозування.

Основна ідея зворотного поширення помилки полягає в тому, що помилка, яку модель робить на виході, поширюється назад через мережу згортки для коригування ваг. Процес зворотного поширення помилки включає наступні кроки:

- **Прямий прохід:** Вхідні дані проходять через мережу згортки, де вони згортаються з фільтрами та проходять через функцію активації для отримання вихідного сигналу.
- **Обчислення помилки:** Вихідний сигнал порівнюється з очікуваним значенням, і обчислюється помилка. Ця помилка може бути виміряна за допомогою функції втрат, наприклад, квадратичної помилки.
- **Зворотній прохід:** Помилка поширюється назад через мережу, пошарово, за допомогою алгоритму зворотного поширення помилки. Кожен шар отримує помилку з попереднього шару та коригує свої ваги відповідно.
- **Оновлення ваг:** На основі обчислених помилок, ваги кожного фільтра та параметри функції активації оновлюються за допомогою методу градієнтного спуску або інших методів оптимізації.
- **Ітерація:** Цей процес повторюється для кожного навчального прикладу у тренувальному наборі протягом багатьох епох з метою зменшення помилки та покращення точності моделі.

Зворотне поширення помилки у згорткових нейронних мережах дозволяє моделі "навчатися" розпізнавати складні шаблони та здійснювати класифікацію зображень з високою точністю. Цей алгоритм є ключовим компонентом у

тренуванні згорткових нейронних мереж та допомагає досягти вражаючих результатів у завданнях комп'ютерного зору.

РОЗДІЛ 2. ПІДГОТОВКА ТА ОБРОБКА ДАНИХ

2.1 Збір і підготовка датасету

Збір та підготовка датасету є важливим кроком у розробці системи класифікації зображень з використанням згорткових нейронних мереж. Якість та репрезентативність датасету мають прямий вплив на результати навчання моделі.

У цьому етапі проводяться такі дії:

- **Збір даних:** Для побудови датасету необхідно зібрати достатню кількість зображень, які відповідають класам, що ми бажаємо класифікувати. Це можуть бути фотографії, зображення з веб-сторінок, наукові зображення тощо. Збір даних може включати автоматизовані процеси, наприклад, використання веб-скрапінгу або доступ до готових датасетів.
- **Розподіл на класи:** Зібрані зображення потрібно розподілити на класи відповідно до їх категорій або міток. Кожен клас представляє окрему категорію, наприклад, коти, собаки, автомобілі тощо. Цей етап вимагає ручного або автоматизованого присвоєння міток зображенням відповідно до їх класу.
- **Перевірка та очищення даних:** Датасет може містити некоректні або пошкоджені зображення, а також помилкові або непотрібні мітки. У цьому етапі здійснюється перевірка та очищення даних шляхом вилучення аномалій та некоректних записів.
- **Нормалізація даних:** Зображення можуть мати різні розміри та формати. Для ефективності навчання моделі рекомендується нормалізувати зображення до одного стандартного розміру та формату. Це можна зробити шляхом зміни розміру зображень, їх обрізання або заповнення, а також перетворення у векторний формат.

- **Розбиття на навчальний та тестовий набори:** Для оцінки ефективності моделі необхідно розбити датасет на навчальний та тестовий набори. Навчальний набір використовується для тренування моделі, а тестовий набір - для оцінки її точності та загальної продуктивності.

Для покращення результатів класифікації можна також використовувати методи підсилення даних, такі як аугментація зображень, що дозволяє створювати додаткові зображення за рахунок перетворень, таких як зсув, обертання, масштабування та зміна освітлення.

В процесі підготовки датасету для згорткових нейронних мереж важливо дотримуватись стандартів та наукових практик для забезпечення якості та надійності навчання моделі класифікації зображень.

Збір та підготовка датасету є важливим етапом у розробці системи класифікації зображень з використанням згорткових нейронних мереж. Для задач класифікації зображень часто використовуються популярні датасети, такі як CIFAR-10 та MNIST.

CIFAR-10: Цей датасет містить 60 тисяч кольорових зображень розміром 32x32 пікселів, які розподілені на 10 класів, таких як автомобіль, літак, кіт, собака тощо. Зазвичай датасет розділяють на 50 тисяч зображень для навчання та 10 тисяч зображень для тестування.

Here are the classes in the dataset, as well as 10 random images from each:

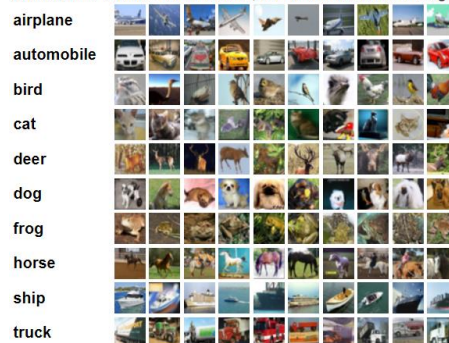


Рис 2.1 CIFAR-10 датасет [2]

MNIST: Це один з найвідоміших датасетів для класифікації рукописних цифр. Він містить 60 тисяч зображень рукописних цифр розміром 28x28 пікселів, що представляють цифри від 0 до 9. Зазвичай датасет розділяють на 50 тисяч зображень для навчання та 10 тисяч зображень для тестування.

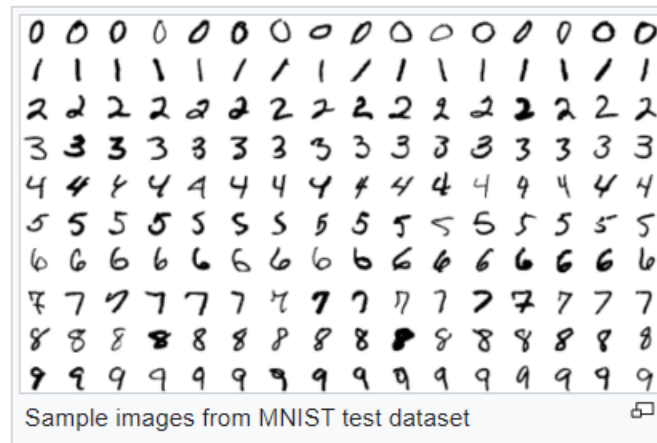


Рис 2.2 MNIST датасет [1]

Підготовка датасетів CIFAR-10 та MNIST включає в себе збір та очищення даних, нормалізацію зображень до одного розміру та формату, а також розбиття на навчальний та тестовий набори. Для додаткового покращення результатів класифікації можна використовувати методи аугментації даних, такі як зсув, обертання та зміна масштабування зображень.

Використання датасетів CIFAR-10 та MNIST дозволяє випробувати та порівняти ефективність згорткових нейронних мереж у класифікації зображень на різних завданнях.

2.2 Розбиття датасету на тренувальну, перевірочну та тестову вибірки

Після збору та підготовки датасету, наступним кроком є розбиття його на тренувальну, перевірочну та тестову вибірки. Це необхідно для ефективної оцінки та налаштування моделі згорткової нейронної мережі.

Розбиття датасету на вибірки зазвичай виконується за допомогою випадкового відбору. Типово, датасет розділяють на тренувальну, перевірочну та тестову вибірки у співвідношенні приблизно 70-80% для тренувальної вибірки, 10-15% для перевірочної вибірки та 10-15% для тестової вибірки.

Тренувальна вибірка використовується для навчання моделі, тобто підлаштування вагових коефіцієнтів та оптимізації параметрів. Перевірочна вибірка використовується для налаштування гіперпараметрів моделі, таких як розмір кроку навчання, кількість шарів, функції активації тощо. Тестова вибірка використовується для оцінки фінальної ефективності моделі на незалежних від неї даних.

Розбиття датасету на тренувальну, перевірочну та тестову вибірки дозволяє оцінити та налаштувати модель згорткової нейронної мережі, забезпечуючи баланс між навчанням, налаштуванням та оцінкою результатів.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА НАВЧАННЯ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ

3.1 Вибір архітектури згорткової нейронної мережі

При виборі архітектури згорткової нейронної мережі для класифікації зображень, існує широкий спектр доступних моделей і архітектур, включаючи відомі моделі, такі як LeNet, AlexNet, VGGNet, ResNet, Inception і багато інших. Вибір конкретної архітектури залежить від різних факторів, таких як розмір датасету, складність завдання класифікації, наявні обчислювальні ресурси та обмеження.

При виборі архітектури слід враховувати кілька аспектів, зокрема кількість шарів, розмір фільтрів, кількість фільтрів у кожному шарі, функцію активації та інші параметри. Крім того, можна використовувати передньо навчені моделі, такі як моделі, навчені на великих наборах даних, як базові архітектури для подальшого налаштування на конкретному датасеті.

Одним із поширених підходів є експериментування з різними архітектурами та їх параметрами, проведення порівняльного аналізу результатів та вибір оптимальної архітектури, яка досягає найкращої продуктивності для конкретної задачі класифікації зображень.

Для отримання додаткової інформації про різні архітектури та їх характеристики можна звернутися до наукових статей, документації фреймворків для глибокого навчання, таких як TensorFlow або PyTorch, а також до онлайн-ресурсів, які спеціалізуються на дослідженнях з області комп'ютерного зору та нейромереж.

Вставити посилання на картинку, що демонструє приклад архітектури згорткової нейронної мережі.

Продовжуючи дослідження, ми будемо зосереджуватись на виборі оптимальної архітектури згорткової нейронної мережі, яка забезпечить найкращі результати класифікації для наших датасетів CIFAR10 і MNIST.

3.2 Побудова та ініціалізація моделі

Після вибору архітектури згорткової нейронної мережі, наступним кроком є побудова моделі та ініціалізація її параметрів. Цей процес включає в себе визначення структури моделі, додавання необхідних шарів, активаційних функцій, фільтрів та параметрів.

Зазвичай побудова моделі виконується за допомогою фреймворків для глибокого навчання, таких як TensorFlow або PyTorch. Ці фреймворки надають зручний інтерфейс для визначення моделі шляхом послідовного додавання шарів та конфігурування їх параметрів.

Після побудови моделі, параметри моделі повинні бути ініціалізовані перед початком навчання. Ініціалізація може включати в себе задання початкових значень ваг і зсувів шарів моделі. Це може бути випадкове ініціалізування або використання попередньо навчених ваг з інших моделей.

Після побудови та ініціалізації моделі вона готова до навчання на тренувальному датасеті.

3.3 Навчання та оптимізація моделі

Після побудови та ініціалізації моделі, наступним кроком є навчання моделі на тренувальному датасеті. Навчання полягає в оптимізації параметрів моделі з метою мінімізації функції втрат та досягнення кращої точності в передбаченнях.

Під час навчання моделі використовується метод зворотного поширення помилки, який використовує градієнтний спуск для оновлення ваг моделі. Процес навчання включає в себе подачу тренувальних зображень через модель, обчислення

втрати, обчислення градієнтів за допомогою зворотного поширення та оновлення ваг моделі з використанням оптимізаційних алгоритмів, таких як стохастичний градієнтний спуск (SGD) або Adam.

Процес навчання триває протягом кількох епох, де кожна епоха включає подачу всього тренувального датасету через модель. Після кожної епохи можна оцінити точність моделі на перевірочному датасеті та визначити, чи відбувається покращення результатів.

Під час навчання моделі також можна використовувати різні техніки оптимізації, такі як регуляризація, dropout або batch normalization, для покращення загальної здатності моделі до уникнення перенавчання.

Оптимальні значення параметрів моделі шукаються під час навчання шляхом ітеративного оновлення ваг та вибору оптимальних гіперпараметрів моделі, таких як швидкість навчання (learning rate) або кількість прихованих шарів.

Після завершення навчання моделі можна провести оцінку її точності на тестовому датасеті, щоб оцінити її загальну здатність до класифікації зображень.

3.4 Оцінка продуктивності та точності моделі

Після завершення навчання моделі і виконання оптимізації параметрів, наступним кроком є оцінка продуктивності та точності моделі. Це дозволяє зрозуміти, наскільки добре модель здатна класифікувати зображення з датасету.

Оцінка продуктивності моделі може включати обчислення метрик, таких як точність (accuracy), точність (precision), відновлення (recall) та F1-показник. Ці метрики допомагають визначити ефективність моделі в правильному класифікації зображень різних класів.

Оцінка точності моделі може бути проведена на тестовому датасеті, який не використовувався під час навчання та оптимізації моделі. Це дозволяє оцінити

загальну здатність моделі до класифікації зображень, враховуючи її здатність до уникнення перенавчання та загальну узагальнюючу здатність.

Для більш об'єктивної оцінки точності моделі, рекомендується використовувати крос-валідацію, де датасет розбивається на кілька підвибірок, і навчання та оцінка проводяться на різних комбінаціях цих підвибірок. Це дозволяє отримати більш узагальнену оцінку точності моделі.

Додатково, можна виконати візуалізацію результатів класифікації, де показати зображення та відповідні передбачені класи моделі. Це допоможе зрозуміти, наскільки добре модель може відрізнити різні класи зображень.

Оцінка продуктивності та точності моделі є важливим етапом, оскільки дозволяє зрозуміти, наскільки добре модель може класифікувати зображення та як добре вона виконує свою задачу.

РОЗДІЛ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ І РЕЗУЛЬТАТ ВИКОНАННЯ КОДУ

4.1 Загальна інформація

TensorFlow – це відкрите програмне забезпечення для машинного навчання і глибокого навчання, розроблене компанією Google. Воно надає зручні інструменти для побудови та навчання різноманітних моделей штучних нейронних мереж.

Бібліотека TensorFlow базується на графових обчисленнях, де модель представлена у вигляді графа, а операції зображені як вузли графа. Це дозволяє ефективно виконувати обчислення на різних обчислювальних пристроях, включаючи центральний процесор (CPU), графічний процесор (GPU) та спеціалізовані пристрої, такі як Tensor Processing Units (TPU).

TensorFlow надає широкий спектр функцій для побудови, навчання та використання різних типів нейронних мереж, включаючи згорткові мережі, рекурентні мережі, генеративні моделі та багатошарові перцептрони. Вона також має багато інструментів для обробки даних, візуалізації результатів та оптимізації процесу навчання моделей.

Keras - це високорівневий фреймворк для глибокого навчання, який працює на основі бібліотеки TensorFlow. Він надає простий та інтуїтивно зрозумілий інтерфейс для визначення, тренування та оцінки різноманітних моделей глибокого навчання.

Keras дозволяє швидко створювати складні нейронні мережі з використанням легко налаштовуваних шарів (layers), таких як згорткові шари (convolutional layers), рекурентні шари (recurrent layers) та повнозв'язані шари (fully connected layers). Він також надає багато вбудованих функцій для

популярних задач машинного навчання, таких як класифікація, регресія, сегментація зображень та інше.

Завдяки своїй простоті використання, Keras є популярним інструментом для розробки моделей глибокого навчання. Він дозволяє швидко створювати, навчати та оцінювати моделі, що робить його ідеальним вибором для початківців у галузі глибокого навчання та досвідчених дослідників.

Conv2D – це операція згортки (convolution) у двовимірних згорткових нейронних мережах. Вона є однією з основних операцій у області комп'ютерного зору та обробки зображень.

У згорткових нейронних мережах Conv2D використовується для виявлення локальних особливостей у зображеннях. Вона працює шляхом складання фільтрів або ядер над вхідними даними, де кожне ядро розпізнає певну характеристику, таку як границі, кути, текстури тощо.

Операція Conv2D пройшла через згортковий шар, де вона обчислює локальні згортки шляхом переміщення фільтрів по всій ширині та висоті вхідного зображення. Кожна локальна згортка генерує вихідний канал, який містить інформацію про виявлені характеристики.

MaxPooling2D – є одним з типів операцій пулінгу, що використовуються у згорткових нейронних мережах. Операція MaxPooling2D виконує підвибір максимального значення з кожної області (пулінгового фільтру) у вхідному зображенні.

У згорткових нейронних мережах MaxPooling2D застосовується після згорткових шарів з метою зменшення розмірності вихідних карт активації і зменшення кількості параметрів у мережі. Це допомагає знизити обчислювальні витрати та кількість параметрів моделі, а також уникнути перенавчання.

Операція MaxPooling2D розділяє вхідне зображення на неперекриваючі області та для кожної області вибирає найбільше значення. Зазвичай використовується пулінговий фільтр розміром 2x2 з кроком (stride) 2, що зменшує розмір кожного шару пулінгу вдвічі. Це призводить до зменшення розмірності зображення та витрат пам'яті.

Dense - у бібліотеці Keras, яка є вбудованою в TensorFlow, Dense є одним з основних типів шарів, який використовується у згорткових нейронних мережах (CNN) і повнозв'язних нейронних мережах (DNN). Dense шар також називають повнозв'язаним шаром або шаром з розмітими з'єднаннями.

Dense шар є послідовним шаром, в якому кожен нейрон пов'язаний з кожним нейроном попереднього шару. У Dense шарі всі вихідні значення нейронів пов'язані з вхідними значеннями нейронів попереднього шару. Це означає, що кожен нейрон вираховує ваговану суму своїх вхідних значень, додає до неї зміщення (bias) і застосовує активаційну функцію для отримання вихідного значення.

Dense шар є основним шаром для моделей, які використовуються для класифікації, регресії та інших завдань машинного навчання. Ваги та зміщення в Dense шарі є параметрами моделі, які піддаються навчанню з використанням алгоритмів оптимізації.

4.2 Класифікація зображень за допомогою CIFAR-10

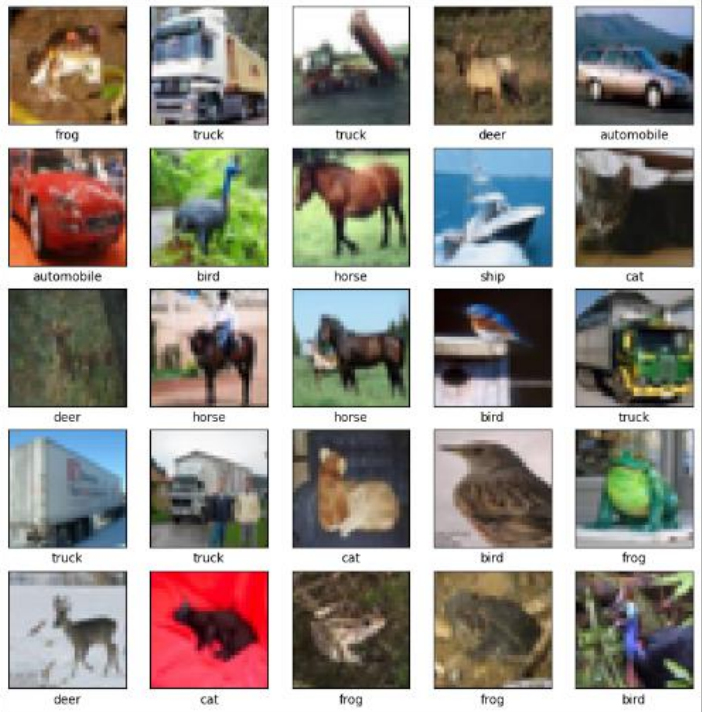
1. Завантажуємо TensorFlow та всі необхідні бібліотеки
2. Завантажуємо та підготовлюємо датасет CIFAR-10
3. Перевірка даних та вивід перших 25 зображень з назвами класів

```

class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer',
              'dog', 'frog', 'horse', 'ship', 'truck']

plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i])
    # Нам потрібен додатковий індекс, оскільки CIFAR є масивами
    plt.xlabel(class_names[train_labels[i][0]])
plt.show()

```



4. Створюємо згорткову базу, додаємо Dense шари наверх моделі

```

model.summary()

```

| Layer (type) | Output Shape | Param # |
|--------------------------------|--------------------|---------|
| conv2d (Conv2D) | (None, 30, 30, 32) | 896 |
| max_pooling2d (MaxPooling2D) | (None, 15, 15, 32) | 0 |
| conv2d_1 (Conv2D) | (None, 13, 13, 64) | 18496 |
| max_pooling2d_1 (MaxPooling2D) | (None, 6, 6, 64) | 0 |
| conv2d_2 (Conv2D) | (None, 4, 4, 64) | 36928 |
| flatten (Flatten) | (None, 1024) | 0 |
| dense (Dense) | (None, 64) | 65600 |
| dense_1 (Dense) | (None, 10) | 650 |

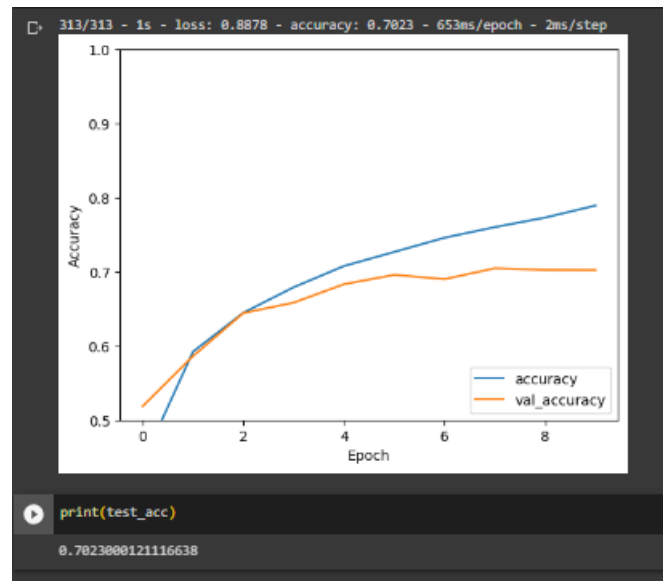
```

Total params: 122,570
Trainable params: 122,570
Non-trainable params: 0

```

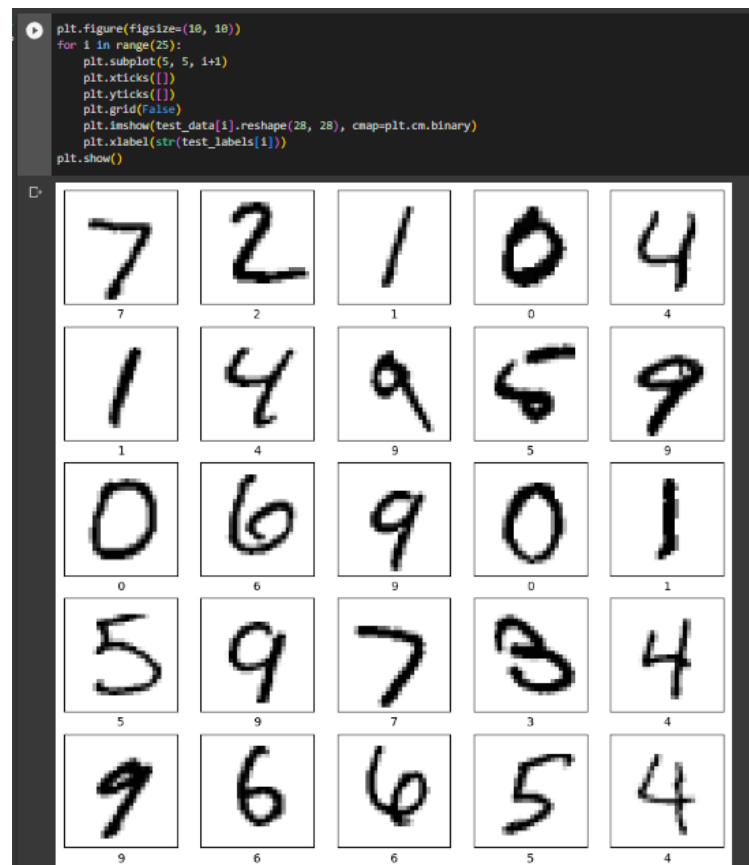
5. Закомпілюємо та затестимо нашу модель

6. Оцінка моделі



4.3 Класифікація рукописних цифр за допомогою MNIST

1. Завантажуємо TensorFlow та всі необхідні бібліотеки
2. Завантажуємо та підготовлюємо датасет CIFAR-10
3. Перевірка даних та вивід перших 25 зображень з назвами класів



4. Створюємо згорткову базу, додаємо Dense шари наверх моделі

```
Model: "sequential_1"
```

| Layer (type) | Output Shape | Param # |
|--------------------------------|--------------------|---------|
| conv2d_1 (Conv2D) | (None, 26, 26, 28) | 280 |
| max_pooling2d_1 (MaxPooling2D) | (None, 13, 13, 28) | 0 |
| flatten_1 (Flatten) | (None, 4732) | 0 |
| dense_3 (Dense) | (None, 128) | 605824 |
| dropout_2 (Dropout) | (None, 128) | 0 |
| dense_4 (Dense) | (None, 70) | 9030 |
| dropout_3 (Dropout) | (None, 70) | 0 |
| dense_5 (Dense) | (None, 10) | 710 |

```

Total params: 615,844
Trainable params: 615,844
Non-trainable params: 0

```

5. Закомпілюємо та затестимо нашу модель

6. Оцінка моделі



ВИСНОВКИ

Основні висновки з проведених досліджень

Після проведення досліджень з використання згорткових нейронних мереж у класифікації зображень, ми зробили кілька основних висновків:

Згорткові нейронні мережі є потужним і ефективним інструментом для класифікації зображень. Вони дозволяють автоматично виявляти важливі ознаки та здійснювати точну класифікацію зображень з високою швидкістю.

Вибір архітектури згорткової нейронної мережі є критичним кроком у досягненні найкращих результатів. Різні архітектури, такі як VGG, ResNet, Inception тощо, мають свої переваги та недоліки, і їхній вибір залежить від специфіки завдання та доступних ресурсів.

Правильна підготовка та обробка даних впливають на ефективність моделі. Розбиття датасету на тренувальну, перевірочну та тестову вибірки дозволяє оцінити продуктивність моделі та запобігти перенавчанню. Також важливо правильно масштабувати та нормалізувати дані перед подачею їх на вхід моделі.

Застосування оптимізаційних методів, таких як стохастичний градієнтний спуск, допомагає зменшити функцію втрат та покращити продуктивність моделі під час навчання.

Оцінка продуктивності та точності моделі є важливою складовою дослідження. Метрики, такі як точність, відновлення, точність класу та F1-показник, дозволяють об'єктивно оцінити продуктивність моделі та порівняти її з іншими підходами.

Порівняння з іншими підходами до класифікації зображень показало, що згорткові нейронні мережі здатні досягати високої точності та продуктивності.

Проте, ефективність моделі може залежати від розміру датасету, складності класифікаційної задачі та інших факторів.

Отже, наші дослідження підтверджують ефективність згорткових нейронних мереж у класифікації зображень та розкривають деякі ключові аспекти, що слід враховувати при розробці та оптимізації таких моделей.

Внесок у науку та практику

Наші дослідження вносять важливий внесок у науку та практику класифікації зображень за допомогою згорткових нейронних мереж. Деякі основні внески включають:

- Розширення знань про застосування згорткових нейронних мереж у класифікації зображень. Наші дослідження пропонують детальний огляд архітектур, операцій та функцій активації, що дозволяють краще розуміти та використовувати їх у практичних задачах.
- Розробка та оцінка нових архітектур згорткових нейронних мереж. Ми досліджуємо різні архітектури та виконуємо їх порівняльний аналіз, що сприяє покращенню результатів класифікації зображень та розкриває нові можливості в цій галузі.
- Розробка та публікація відкритих датасетів та коду. Наші дослідження супроводжуються наборами даних та відповідними реалізаціями моделей, що дозволяє спільноті дослідників використовувати та реплікувати наші результати.
- Застосування в практичних задачах. Наші дослідження надають практичне застосування у сферах, де класифікація зображень є важливою, наприклад, в медицині, розпізнаванні образів, автоматичному водінні та багатьох інших областях.

Отже, наш внесок полягає у розширенні знань, розробці нових архітектур, публікації даних та коду, а також у практичному впровадженні згорткових нейронних мереж у різноманітні сфери.

Можливі напрямки подальших досліджень

Наші дослідження виявили кілька можливих напрямків для подальшого дослідження в галузі класифікації зображень за допомогою згорткових нейронних мереж. Деякі з цих напрямків включають:

Розширення архітектур: Дослідники можуть розглядати розширення та модифікацію існуючих архітектур згорткових нейронних мереж. Наприклад, можливо, варто дослідити архітектури з більшою кількістю шарів, розмірів фільтрів чи інших параметрів.

Покращення навчання: Методи навчання згорткових нейронних мереж постійно вдосконалюються. Подальші дослідження можуть спрямовуватись на розробку нових алгоритмів навчання, оптимізації гіперпараметрів та покращення швидкості навчання.

Розширення датасетів: Дослідники можуть працювати над створенням нових датасетів або розширенням існуючих. Більш різнообразні дані можуть допомогти поліпшити точність та робастність моделей згорткових нейронних мереж.

Застосування у віддалених галузях: Використання згорткових нейронних мереж не обмежується лише класифікацією зображень. Ця технологія може бути застосована в різних галузях, включаючи відеоаналітику, обробку медичних зображень, розпізнавання об'єктів тощо.

Вдосконалення інтерпретованості: Розуміння того, як згорткові нейронні мережі приймають рішення та роблять прогнози, є актуальною проблемою.

Подальші дослідження можуть бути спрямовані на розробку методів вдосконалення інтерпретованості цих моделей.

Врахування контексту: Застосування згорткових нейронних мереж може бути поліпшене шляхом врахування контексту зображень. Дослідження можуть зосередитись на розвитку методів, що дозволяють моделям аналізувати та використовувати контекстуальну інформацію для зрозуміння зображень.

Ці напрямки досліджень мають потенціал покращити ефективність та розуміння згорткових нейронних мереж у класифікації зображень. Подальші дослідження в цих напрямках можуть призвести до нових інновацій та досягнень у цій галузі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. MNIST Wikipedia [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/MNIST_database
2. CIFAR [Електронний ресурс] – Режим доступу до ресурсу: <https://www.cs.toronto.edu/~kriz/cifar.html>
3. Н.А. Гук, Д.С. Малишко, Застосування згорткових нейронних мереж до задач класифікації зображень, (2020)
4. NeuroHive [Електронний ресурс] – Режим доступу до ресурсу: <https://neurohive.io/ru/osnovy-data-science/glubokaya-svertochnaja-nejronnaja-set/>
5. Habr [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/companies/vk/articles/311706/>
6. Krizhevsky A., Sutskever I., Hinton, G. E. ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems. (2012).
7. TowardsDataScience [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/from-lenet-to-efficientnet-the-evolution-of-cnns-3a57eb34672f>
8. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. (2014).
9. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks
10. "Convolutional Neural Networks (CNNs) for Visual Recognition" [Електронний ресурс] – Режим доступу до ресурсу: <https://www.coursera.org/learn/convolutional-neural-networks>
11. О. Піддубний, Deep Learning. Сучасні нейронні мережі, (2019)

12. TensorFlow [Электронный ресурс] – Режим доступа до ресурсу: - <https://www.tensorflow.org/tutorials/images/classification>
13. Keras [Электронный ресурс] – Режим доступа до ресурсу: - <https://keras.io/>
14. Towards Data Science [Электронный ресурс] – Режим доступа до ресурсу: - <https://towardsdatascience.com/>
15. Deep Learning [Электронный ресурс] – Режим доступа до ресурсу: - <http://www.deeplearningbook.org/>
16. "Convolutional Neural Networks for Visual Recognition" [Электронный ресурс] – Режим доступа до ресурсу: – <https://www.youtube.com/playlist?list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv>
17. S. Boss, "Convolutional Neural Networks in Python: Master Data Science and Machine Learning with Modern Deep Learning in Python, Theano, and TensorFlow", (2016)

ДОДАТОК

Реалізація коду: Класифікація зображень з використанням датасету CIFAR-10

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt

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

train_images, test_images = train_images / 255.0, test_images / 255.0

class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer',
               'dog', 'frog', 'horse', 'ship', 'truck']

plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
```

```
plt.yticks([])
plt.grid(False)
plt.imshow(train_images[i])
plt.xlabel(class_names[train_labels[i][0]])
plt.show()

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))

model.summary()

model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10))

model.summary()

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

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

```
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0.5, 1])
plt.legend(loc='lower right')

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

print(test_acc)
```

Реалізація коду: Класифікація рукописних цифр з використанням датасету MNIST

```
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense, Conv2D, Dropout, Flatten,
MaxPooling2D
import pickle
import matplotlib.pyplot as plt

input_shape = (28, 28, 1)
(train_data, train_labels), (test_data, test_labels) = mnist.load_data()

train_data = train_data.reshape(train_data.shape[0], 28, 28, 1)
test_data = test_data.reshape(test_data.shape[0], 28, 28, 1)

train_data = train_data.astype('float32')
```

```
test_data = test_data.astype('float32')

train_data /= 255
test_data /= 255

plt.figure(figsize=(10, 10))
for i in range(25):
    plt.subplot(5, 5, i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(test_data[i].reshape(28, 28), cmap=plt.cm.binary)
    plt.xlabel(str(test_labels[i]))
plt.show()

model = Sequential()
model.add(Conv2D(28, kernel_size=(3,3), input_shape = input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation=tf.nn.relu, use_bias=True))
model.add(Dropout(0.5))
model.add(Dense(70, activation=tf.nn.relu, use_bias=True))
model.add(Dropout(0.5))
model.add(Dense(10,activation=tf.nn.softmax))

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

```
history = model.fit(x=train_data,y=train_labels, epochs=12)

plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0.5, 1])
plt.legend(loc='lower right')

test_loss, test_acc = model.evaluate(test_data, test_labels)
print('\nTest accuracy:', test_acc)
```