Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

Текстова частина до курсової роботи

за спеціальністю «Комп’ютерні науки» – 122

Керівник курсової роботи
Швай Надія Олександрівна

_____
(підпис)
“____” _____ 2023 р.

Виконала студентка
Лахтюк А.В.
“____” _____ 2023 р.

Київ 2023

Зміст

Перелік термінів, використаних у роботі

CNN/Згорткова нейронна мережа - Convolutional neural network

Операції об’єднання/pooling operations/операції агрегування – Pooling operations in CNNs

Датасет – dataset

Кернел – kernel

Анотація

Робота стосується розробки нейронних мереж, а саме Convolutional neural networks (CNNs), що використовуються переважно у програмах комп’ютерного зору та дозволяють проводити якісну класифікацію зображень. На її точність впливають різні параметри у тому числі pooling operations, що існують різних типів. Моделі, побудовані та описані тут, створені за використання датасетів CIFAR-10 та CIFAR-100. Код імплементовано на мові програмування Python, найважливішими використаними бібліотеками є keras та tensorflow.

Вступ

Актуальність, наукове та практичне значення обраної теми

Однією з найбільш багатообіцяючих тем в сучасному світі технологій є нейронні мережі. Вони широко використовуються у різних сферах для великої кількості різноманітних завдань, що не можуть бути вирішені за допомогою інших відомих чітких алгоритмів. Без глибинного навчання та нейронних мереж неможливо уявити програм з computer vision, natural language processing, speech recognition. Та водночас із часом впровадження та використання нейронних мереж поширюється на інші сфери тісно пов’язані або зовсім не пов’язані з інформаційними технологіями.

Аналогічно до сфери використання нейронних мереж уцілому, використання Convolutional neural networks збільшується. Водночас саме ці мережі стали хорошим інструментом у завданнях класифікації зображень та не лише. У різноманітних задачах комп'ютерного зору штучні нейронні мережі CNN піднялися на вершину.

Операції об’єднання (pooling operations) є невід’ємної частиною CNN. Існують різні типи цих операцій та водночас досі немає чітких правил їх використання. Дуже складно встановити який тип операцій буде найкращим для конкретного завдання та конкретного набору даних. У той час як хтось стверджує, що максимальне об’єднання може більше підійти для захоплення просторової інформації, інші вважають, що об’єднання по середньому значенню може краще сприяти завданню сегментації.

Обрана тема є актуальною так як розглядає один із фундаментальних аспектів роботи CNN.

Робота складається з 4 розділів. У першому я наводжу загальний опис нейронних мереж, CNN, досліджую практичність та доцільність обраної теми. У наступному розділі я додала опис обраних датасетів та бібліотек використаних для виконання практичної частини курсової роботи. Третій розділ включає в себе опис типів pooling operations, їх практичного використання. У четвертому розділі наведена інформація про базову структуру нейронної мережі, описані досліджені моделі та частково наведені результати дослідження. Вкінці додані висновки.

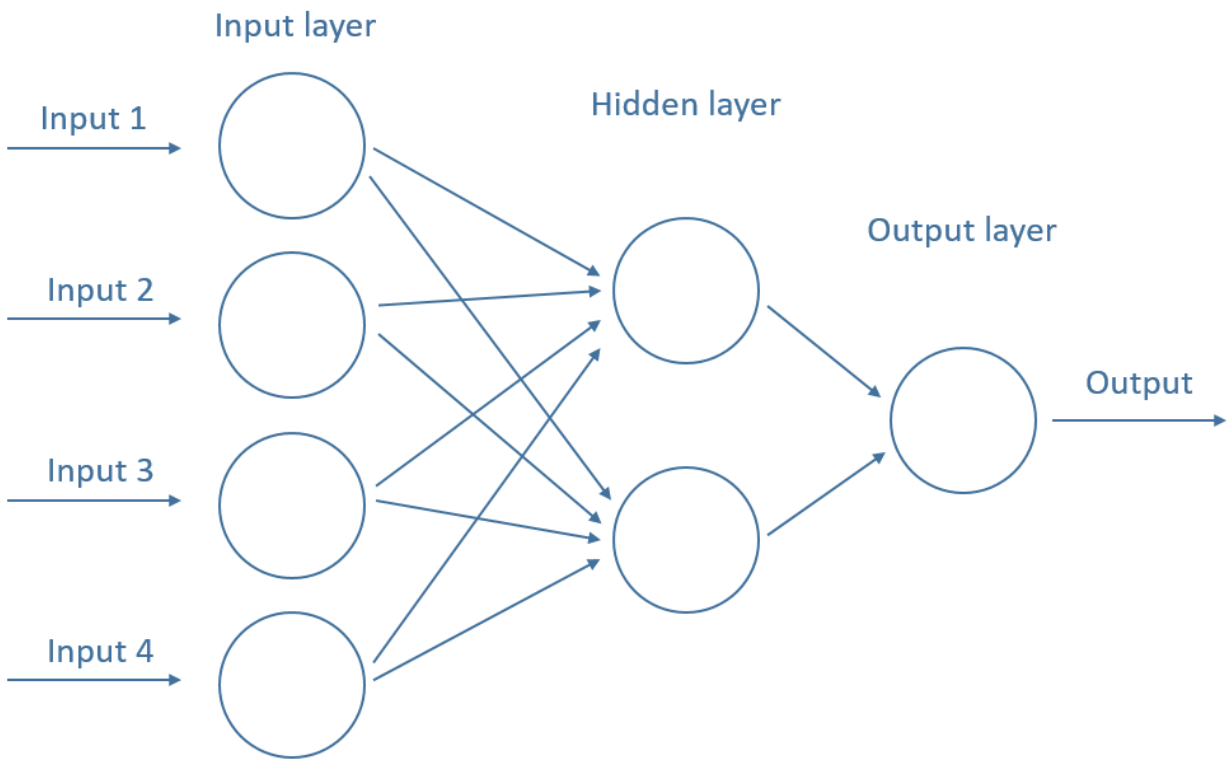
Метою роботи є дослідити та порівняти різні типи pooling operations у згорткових нейронних мережах.

Розділ 1

1.1 Нейронні мережі, загальний опис

Нейронні мережі являють собою обчислювальні системи, які створені за прикладом роботи людського мозку. Вони в основному складаються з великої кількості пов’язаних між собою обчислювальних вузлів (нейронів), що «навчаються» в процесі роботи мережі та оптимізовують отриманий результат. Нейронні мережі складаються з взаємопов'язаних вузлів обробки, або нейронів, які організовані в шари. Вихід з одного шару нейронів подається як вхід до наступного шару, що дозволяє мережі вивчати все більш складні функції з вхідних даних.[3]

Базова структура нейронної мережі може бути зображена наступним чином:



1.2 Convolutional neural networks

1.2.1 Загальний опис

Convolutional neural networks - це тип багатошарових моделей глибокого навчання, нейронна мережа, що працює за принципом згорткових [convolutional] шарів. Широко використовується в програмах комп'ютерного зору, таких як розпізнавання зображень, виявлення об'єктів, аналіз відео, класифікація і сегментація. Завдання CNN полягає в стисненні зображень у формат, який легше обробляти, зберігаючи елементи, важливі для отримання гідного передбачення. Це має вирішальне значення для проектування архітектури, яка здатна вивчати функції, а також масштабована для великих наборів даних. [10]

1.2.2 Архітектура

CNN складається з кількох типів різних шарів:

- Вхідний шар
- Згорткові шари
- Шари об'єднання (pooling)
- Повноз'єднувальні шари

Convolutional (згорткових) шари є основою цього типу нейронних мереж, вони призначені для виявлення локальних особливостей у вхідному зображенні. У цих шарах до вхідних даних застосовується набір фільтрів/кернелів, які можна вивчати, створюючи набір сукупностей функцій, які фіксують різні аспекти вхідних даних. Поєднуючи кілька згорткових шарів разом, CNN можуть вивчати все більш складні та абстрактні представлення вхідних даних. [3]

Окрім згортки, що виконується за рахунок фільтрів/кернелів важливою частиною CNN є нелінійна функція активації. Виходи лінійних операцій на кшталт згортки пропускаються через нелінійну функцію активації. Хоча певні нелінійні функції, такі як sigmoid або tanh функція використовувалися раніше

тому що вони є математичними представленнями біологічних нейронних дій. ReLU зараз є найбільш часто використовуваною нелінійною функцією активації.

- ReLU (Rectified Linear Unit): ReLU є найбільш часто використовуваною функцією активації в CNN. Це проста нелінійна функція, яка повертає нуль для всіх від'ємних входів і вхідне значення для всіх позитивних входів. Функція активації ReLU є обчислювально ефективною і, як показано, добре виконує багато завдань класифікації зображень.
- Сигмоїд: Функція активації сигмоїда відображає будь-яке вхідне значення в значення від 0 до 1. Це гладка функція, яка широко використовується в задачах бінарної класифікації, таких як виявлення об'єктів або сегментація. Однак сигмоїдна функція рідше використовується в CNN через проблему зникаючого градієнта, що може ускладнити тренування глибоких мереж.
- Tanh (гіперболічний тангенс): гіперболічна дотична функція схожа на сигмоїдну функцію, але відображає будь-яке вхідне значення в значення між -1 і 1. Це гладка функція, яка може бути використана в задачах бінарної класифікації та задачах регресії. Однак, як і сигмоїдна функція, вона страждає від проблеми зникаючого градієнта.
- Maxout: Функція активації Maxout - функція, яку можна використовувати для апроксимації будь-якої опуклої функції. Це більш потужна функція активації, ніж ReLU, і може допомогти підвищити точність роботи мережі. Однак він є обчислювально дорогим і вимагає більше параметрів, ніж ReLU.
- Softmax: Функція softmax використовується у вихідному шарі CNN для отримання розподілу ймовірності над класами виводу. Він відображає будь-яке вхідне значення в значення між 0 і 1 і гарантує, що сума ймовірностей над усіма класами дорівнює одиниці. [9] [10]

Підсумовуючи, функції активації є важливими компонентами CNN, що дозволяє їм вивчати складні функції з вхідних даних. Вибір функції активації залежить від конкретного завдання і особливостей використовуваних даних. ReLU є найбільш часто використовуваною функцією активації завдяки своїй простоті та обчислювальній ефективності, але інші функції активації, такі як сигмоїд та tanh, також можуть бути використані в певних ситуаціях. Використання функцій Maxout і Softmax може допомогти підвищити точність роботи мережі в конкретних додатках.

Однією з проблем використання CNN для завдань пов’язаних з обробкою зображень є те, що просторовий розмір вхідного шару нейронів може стати дуже великим із збільшенням кількості шарів. Це може призвести до великої кількості параметрів і повільних обчислень, що ускладнює ефективне навчання мережі. Щоб вирішити цю проблему, CNN використовують операції об’єднання, які призначені для зменшення просторового розміру шару нейронів, зберігаючи їх важливі функції.

Шари об’єднання (у яких використовуються операції об’єднання) в CNN зазвичай застосовуються після згорткових шарів. Метою об’єднання є зменшення розмірності шару нейронів шляхом узагальнення інформації в кожному локальному регіоні. Найпоширенішими операціями об’єднання є максимальне об’єднання та об’єднання по середніх значеннях.

Наступний повноз’єднувальний шар з’єднує між собою всі вхідні нейрони з вихідними. Використовується після всіх попередньо згаданих шарів. На цьому етапі починається безпосередньо завдання класифікації. Остання функція активації у повноз’єднувальному шарі може відрізнятися від інших. Вибір цієї функції залежить від виконуваного завдання.

Важливо зазначити, що операції з об’єднання також можуть мати деякі недоліки. Наприклад, об’єднання може призвести до втрати просторової роздільної здатності, що може ускладнити точну локалізацію об’єктів у вхідному зображенні. Щоб вирішити цю проблему, деякі новітні архітектури CNN, такі як повністю згорткові мережі (FCN), замінили об’єднання поступовими згортками, які досягають подібного ефекту зменшення дискретизації, зберігаючи просторову роздільну здатність.

Загалом, операції об’єднання є важливим компонентом CNN, і вони відіграють важливу роль у зменшенні просторового розміру карт функцій і підвищенні обчислювальної ефективності мережі. Вибравши відповідний тип операції об’єднання та ретельно налаштувавши гіперпараметри мережі, можна досягти найсучаснішої продуктивності в широкому діапазоні завдань комп’ютерного зору.

Розділ 2

2.1 Вибір та опис датасетів для тренування згорткових нейронних мереж

Є певна кількість стандартних наборів даних, що використовуються для тренування нейронних мереж. У завданнях класифікації зображень часто використовуються MNIST, CIFAR-10, CIFAR-100. Обрані датасети дозволяють зосередитися на безпосередній роботі над створенням нейронних мереж та водночас дають можливість порівнювати отримані практичні результати на двох наборах різних даних.

Оцінка ефективності роботи кожного pooling методу проводилась на двох датасетах: CIFAR-10 та CIFAR-100.

2.1.1 CIFAR-10

Зручний у використанні, один із найбільш поширених наборів даних, що часто використовується у дослідженнях пов’язаних з комп’ютерним зором та машинним навчанням. Цей датасет містить 60 000 кольорових зображень 32 х 32 пікселі, що згруповані в 10 класів, відповідно по 6 000 зображень на клас. Цей набір розподілено таким чином: 50 000 навчальних розмічених зображень та 10 000 тестових для оцінки роботи моделі натренованої на навчальних даних. 10 класів, згаданих раніше, виглядають наступним чином:

1. літак
2. автомобіль
3. птах
4. кіт
5. олень
6. собака
7. жаба

8. кінь
9. корабель
10. вантажівка

Зображення в наборі даних CIFAR-10 мають низьку роздільну здатність і відносно простий фон, що робить набір даних хорошим еталоном для оцінки продуктивності моделей машинного навчання в задачах класифікації зображень. Набір даних був зібраний Канадським інститутом перспективних досліджень [Canadian Institute for Advanced Research] (CIFAR) у 2009 році, і з тих пір він став одним із найбільш широко використовуваних наборів даних у спільнотах комп’ютерного зору та машинного навчання.

Набір даних CIFAR-10 є складним, оскільки зображення малі та з низькою роздільною здатністю, а класи візуально подібні. Наприклад, клас "собака" містить зображення різних порід собак, які важко відрізнити один від одного. Подібним чином клас «кішка» містить зображення котів з різними кольорами та малюнками шерсті. Як наслідок, для досягнення високої точності набору даних CIFAR-10 потрібна модель машинного навчання, яка може ефективно вловлювати тонкі відмінності між зображеннями та добре узагальнювати нові, небачені зображення.

2.1.2 CIFAR-100

Набір даних CIFAR-100 схожий на набір даних CIFAR-10, але з більш детальними мітками. Він складається з 60 000 кольорових зображень 32 x 32 у 100 класах, по 600 зображень на клас. Набір даних також розділений на 50 000 навчальних зображень і 10 000 тестових зображень.

100 класів у наборі даних CIFAR-100 організовано в 20 суперкласів, кожен з яких містить 5 підкласів. Кожен підклас містить 100 зображень. 20 суперкласів

1. водні ссавці

2. риба
3. квіти
4. харчові контейнери
5. фрукти та овочі
6. побутові електроприлади
7. побутові меблі
8. комахи
9. великі хижаки
10. великі штучні речі на відкритому повітрі
11. великі природні сцени на відкритому повітрі
12. великі всеїдні та травоїдні тварини
13. ссавці середнього розміру
14. Безхребетні (не комахи)
15. Люди
16. рептилії
17. дрібні ссавці
18. дерева
19. Транспортні засоби 1
20. транспортні засоби 2

Зображення в наборі даних CIFAR-100 також мають низьку роздільну здатність і мають простий фон, але класи є більш дрібнозернистими та потребують моделі для захоплення більш детальних візуальних характеристик, щоб добре працювати. Набір даних CIFAR-100 часто використовується як еталонний набір даних для оцінки продуктивності моделей машинного навчання в задачах детального розпізнавання зображень.

Для досягнення високої точності набору даних CIFAR-100 потрібна модель, яка може фіксувати тонкі візуальні особливості та розрізняти візуально подібні класи. Набір даних є складним, оскільки існує багато класів зі схожими

візуальними характеристиками, такими як різні види квітів або домашніх меблів, а також тому, що зображення мають відносно низьку роздільну здатність.

2.2 Використані бібліотеки

При виконанні кодової частини своєї курсової роботи я використала наступні основні бібліотеки:

Numpy – бібліотека, що розширює спектр математичних функцій Python. Надає можливість використовувати багатовимірний масив та різні похідні об’єкти. [4]

Tensorflow – відкритий фреймвокр створений Google для реалізації різноманітних алгоритмів машинного навчання. Дозволяє зручно та швидко створювати складні нейронні мережі. [5]

Matplotlib – бібліотека для візуалізації різноманітних даних. Дозволяє створювати статичні, анімовані, інтерактивні візуалізації. [6]

Keras – відкрита бібліотека написана на Python для створення нейронних мереж, у томи числі CNNs. Є надбудовою над згаданої раніше бібліотекою Tensorflow, Theano, and CNTK. Дає користувачу можливість швидко створювати складні нейронні мережі. [7]

Розділ 3. Pooling operations

Є дві основні категорії Pooling operations: локальні та глобальні.

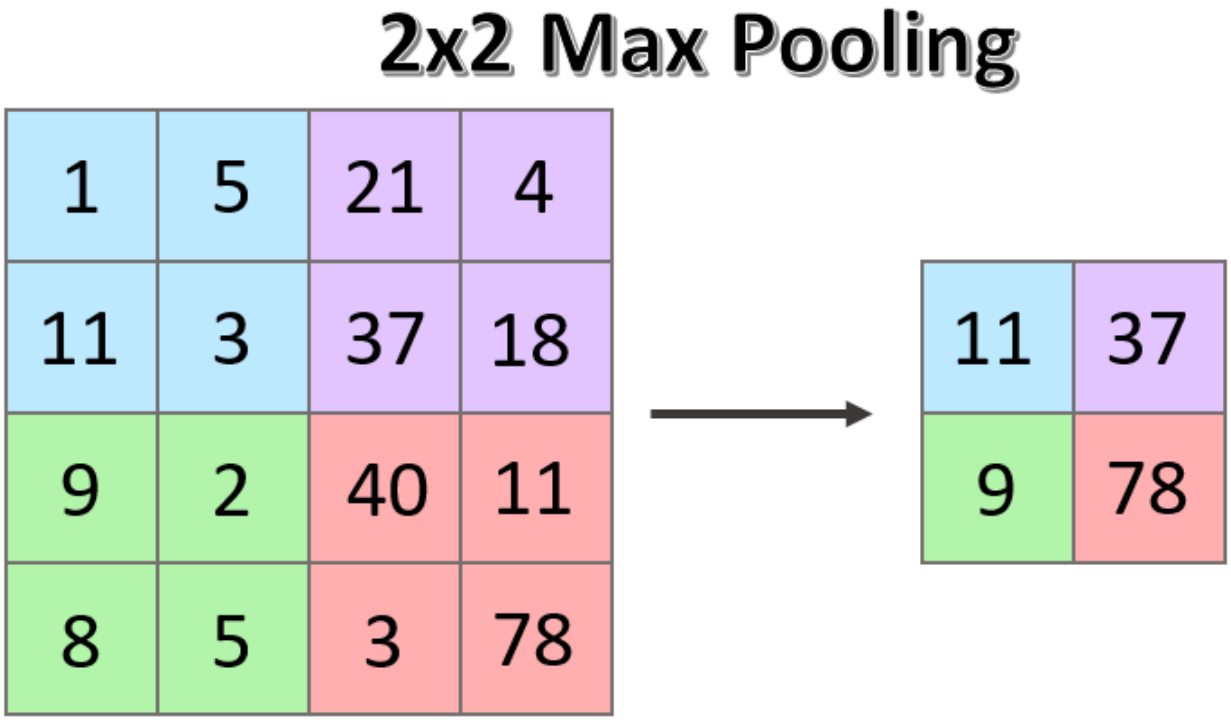
Найчастіше використовувані операції об’єднання [pooling]: максимальне об’єднання та середнє об’єднання:

3.1 Max pooling

Максимальне об’єднання поділяє вхідне зображення на прямокутні підобласті, що не перекриваються, і виводить максимальне значення в кожній підобласті.

Max pooling operation визначається розміром рамки підобласті, що використовуватиметься для зменшення початкової карти (зазвичай 2x2 або 3x3) і кроком, що визначає відстань між підобластями (зазвичай 2 або 3). Операція застосовується незалежно до кожного каналу карти вхідних функцій.

Для кожної підобласті обчислюється максимальне значення в цій підобласті та виводиться на наступний рівень. Інші значення в межах субрегіону відкидаються. Цей процес повторюється для кожного субрегіону на карті об’єктів, у результаті чого виходить менша вихідна карта об’єктів із зменшеними просторовими розмірами.



Максимальне об’єднання зменшує просторові розміри карти функцій, роблячи наступні шари обчислювально ефективнішими, вводить певний ступінь трансляційної інваріантності, оскільки максимальне значення в субрегіоні буде інваріантним для малих перекладів у вхідних даних, підвищує рівень абстракції на карті ознак, оскільки максимальне значення представляє найважливішу функцію в цьому субрегіоні. Ця операція корисна для виявлення наявності

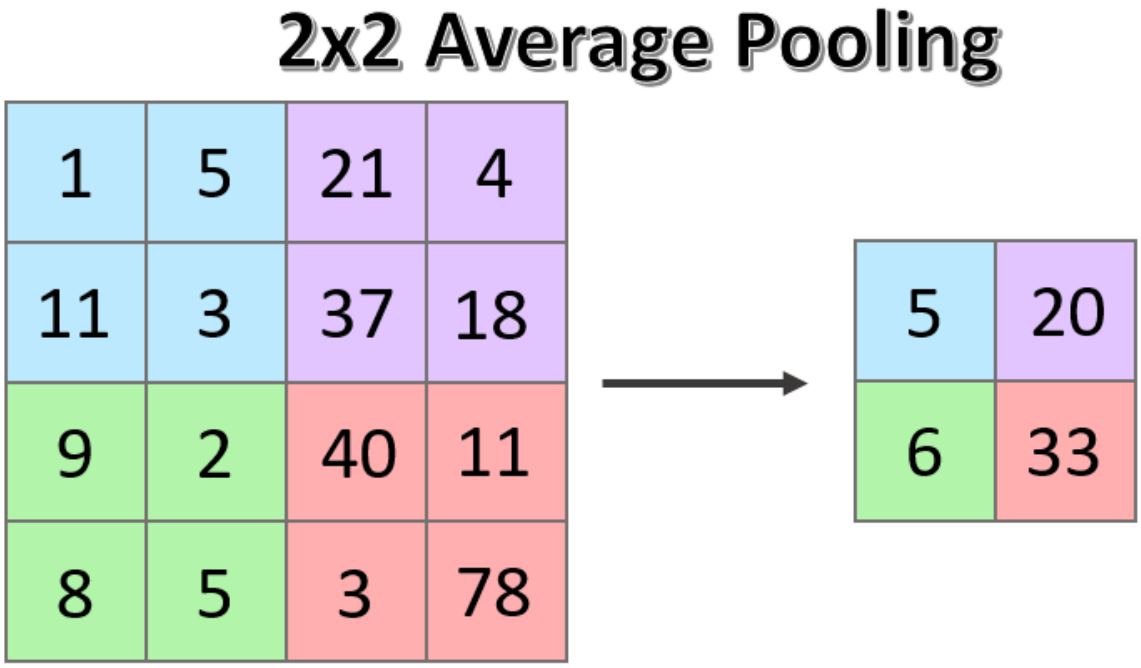
певних особливостей, наприклад країв або кутів, і допомагає зменшити вплив невеликих перекладів або спотворень у вхідному зображенні. Зараз максимальне агрегування переважно використовується, як базове та дефолтне у згорткових нейронних мережах.

Однак максимальне об'єднання також може призвести до втрати інформації, оскільки відкинуті значення в кожній підгрупі неможливо відновити на наступних рівнях. Це можна частково вирішити, використовуючи менший крок, що призводить до деякого ступеня перекриття між сусідніми підгрупами. [1]

$$f_{max}x = \max\{x_i\}_{i=1}^{N}$$

3.2 Average pooling

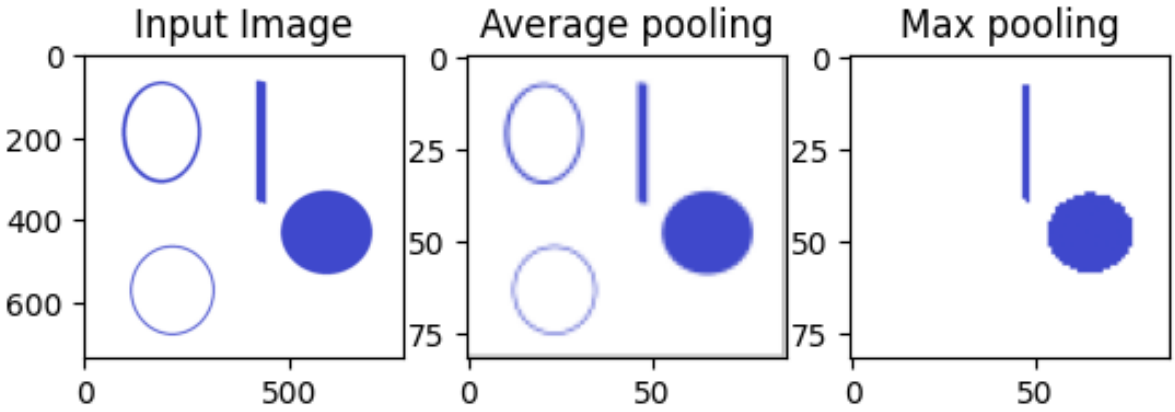
Ця операція працює аналогічно до попередньо описаної операції max pooling за відмінністю лише того, що замість вибору найбільшого значення в межах однієї підгрупи average pooling залишає середнє значення усіх компонентів підгрупи. Ця операція корисна для згладжування карт функцій і зменшення впливу шуму або невеликих варіацій у вхідному зображенні.



У порівнянні з максимальним об’єднанням, об’єднання по середньому значенню дає меншу схильність до перенавчання. Однак воно також може призводити до втрати інформації. [1]

$$f_{avg}x = \frac{1}{N}\sum_{i=1}^{N}|x_i|$$

Для візуального зображення відмінностей у роботі максимального та середнього об’єднань можна використати стандартні матриці та операції над ними:



3.3 Mixed max-average pooling

Лінійна комбінація максимального об’єднання з об’єднанням по середньому значенню з використанням вагів можу допомогти уникнути проблем пов’язаних з використанням цих об’єднань поодинці.

$$f_{mix} = af_{max}x + (1-a)f_{avg}$$

3.4 Global max pooling

Використовується вкінці моделі, щоб отримати зображення, яке далі передається до шару, що виконує лінійну класифікацію для отримання результатів класифікації. Працює схоже до того, як працює max pooling, але не використовує підобласті, а натомість формує вектор найбільших значень кожного вхідного елементу. [1] Глобальні операції об'єднання працюють

аналогічно до неглобальних відповідників, але розмір підобласті дорівнює розміру загальної області вхідного шару.

3.5 Global average pooling

Працює ідентично до того, як працює Global max pooling, але замість обирання найбільшого значення, обчислює середнє та додає його до результуючого вектору.

3.6 Stochastic pooling

Перенавчання - поширена проблема в машинному навчанні, в тому числі в згорткових нейронних мережах (CNN). Перенавчання відбувається, коли модель занадто складна і занадто добре тренується на тренувальних даних, до того моменту, коли вона починає запам'ятовувати навчальні дані, замість того, щоб вивчати основні закономірності та відносини, які узагальнюють до нових, невидимих даних. [1]

Для усунення недоліків максимального об’єднання та об’єднання по середньому значенню було введене стохастичне об’єднання. Воно використовує метод випадкового ймовірнісного вибору елементів під час тренування і завдяки цьому модель може вийти набором подібних між собою мереж з різними pooling конфігураціями, що визначають різних членів цього набору.

Розділ 4. Опис створених нейронних мереж

Базова архітектура використана мною для виконання порівняння різних типів pooling operations використовує два згорткові шари, після яких іде pooling операція, а вкінці переважно для отримання результатів використовується повноз’єднувальний шар. Функція активації, що була використана мною в нейронних мережах - ReLu

Loss функція, яку я переважно використовувала це categorical_crossentropy. Optimization функція – rmsprop.

Отримані результати

Тренування моделей CIFAR-10 не вимагає значних потужностей та зазвичай не є часозатратним для роботи самої моделі. Але тим не менш створити модель, що матиме високу точність вихідних даних не просто, потрібно експериментувати, підбирати кількість шарів, типи та послідовність операцій та параметрів.

Знати кількість вхідних і вихідних шарів і кількість їх нейронів - найпростіша частина. Кожна мережа має один вхідний шар і один вихідний шар. Кількість нейронів у вхідному шарі дорівнює кількості вхідних змінних у оброблюваних даних. Кількість нейронів у вихідному шарі дорівнює кількості виходів, пов'язаних з кожним входом. Але завдання полягає в тому, щоб знати кількість прихованих шарів і їх нейронів.

На жаль, не можна аналітично обчислити кількість шарів або кількість вузлів для використання у шарі у штучній нейронній мережі для вирішення конкретної задачі передбачувального моделювання в реальному світі. Кількість шарів і кількість вузлів в кожному шарі є модельними гіперпараметрами, які потрібно вказати і дізнатися тестовим шляхом, щоб побачити, що найкраще працює для конкретного набору даних.

Наприклад, розмір фільтра є одним з таких гіперпараметрів, які потрібно вказати перед навчанням вашої мережі.

Для проблеми розпізнавання зображень, якщо велика кількість пікселів необхідна для того, щоб мережа розпізнала об'єкт, варто використовувати великі фільтри (як 11x11 або 9x9). Якщо ж можна виконати завдання використавши при цьому меншу розмірність проміжних даних, то можна обмежитися використанням невеликих фільтрів (3x3 або 5x5).

Моделі розроблені мною під час виконання цієї роботи:

Model	Architecture

Type 1 Max pooling usage	Conv2D, Conv2D, MaxPooling2D Conv2D, Conv2D, MaxPooling2D Conv2D, Conv2D, MaxPooling2D Flatten() Dense(), Dense()
Type 2 Average pooling usage	Conv2D, Conv2D, AveragePooling2D Conv2D, Conv2D, AveragePooling2D Conv2D, Conv2D, AveragePooling2D Flatten() Dense(), Dense()
Type 3 Max + 2*Average Pooling	Conv2D, Conv2D, MaxPooling2D Conv2D, Conv2D, AveragePooling2D Conv2D, Conv2D, AveragePooling2D Flatten() Dense(), Dense()
Type 4 2* Max + Global Max Poling	Conv2D, Conv2D, MaxPooling2D Conv2D, Conv2D, MaxPooling2D Conv2D, Conv2D, GlobalMaxPooling2D Flatten() Dense(), Dense()
Type 5 2* Average + Global Average Poling	Conv2D, Conv2D, AveragePooling2D Conv2D, Conv2D, AveragePooling2D Conv2D, Conv2D, GlobalAveragePooling2D Flatten() Dense(), Dense()
Type 6 +Batch Normalization	Conv2D, BatchNormalization, Conv2D, BatchNormalization, MaxPooling2D, Dropout,

	Conv2D, BatchNormalization, Conv2D, BatchNormalization, MaxPooling2D, Dropout, Conv2D, BatchNormalization, Conv2D, BatchNormalization, MaxPooling2D, Dropout, Flatten, Dense, Dropout, Dense

Отримані точності для тренувального та тестового сетів для вище поданих типів моделей можна побачити у наступній таблиці:

Model	Train Accuracy (%)	Test Accuracy (%)
Type 1	71.8	71.46
Type 2	67.92	66.36
Type 3	69.87	68.88
Type 4	71.03	70.96
Type 5	72	72.03
Type 6	83.4	83.34

Загалом відмінність у точностях є порівняно незначною, але тим не менш можна помітити певні розбіжності у значеннях для використання різних типів pooling operations. Наприклад, очевидно, що об’єднання по середньому значенню дає меншу результуючу точність, ніж максимальне і водночас об’єднання по середньому разом з використанням глобального об’єднання по середньому дає краший результат, ніж просто використання середнього.

Використання наступного датасету переважно вимагає більших потужностей та ретельнішого пошуку підходів, так як отримати високу точність для класифікації у 100 класів з тією ж загальною кількістю зображень, що використовується для побудови класифікаторів у 10 класів, є непросто. Найкращим результатом досягнутим мною є: 63% точності. При цьому для побудови цієї моделі я використовувала max pooling та batch normalization.

Висновки

У цій курсовій роботі я дослідила тип нейронних мереж, що працює за рахунок згорток, навела їх архітектурний склад та провела аналіз різних типів pooling operations та того, як вони можуть впливати на точність роботи згорткових нейронних мереж. Для виконання різних завдань за допомогою CNNs можуть підійти різні pooling operations. Часто для найкращого результату варто поєднувати використання різних операцій у різних шарах. Для першого (CIFAR-10) датасету найкраща отримана точність 83%, для другого (CIFAR-100) 63%

Джерела:

1. «Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study» Rajendran Nirthika, Siyamalan Manivannan, Amirthalingam Ramanan, Ruixuan Wang [Електронний ресурс]: режим доступу: https://link.springer.com/article/10.1007/s00521-022-06953-8
2. “A Comparison of PoolingMethods for Convolutional Neural Networks” Afia Zafar, Muhammad Aamir, Nazri Mohd Nawi, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta and Sultan Almotairi [Електронний ресурс]: режим доступу: https://www.researchgate.net/publication/363090585_A_Comparison_of_Pooling_Methods_for_Convolutional_Neural_Networks
3. An Introduction to Convolutional Neural Networks [Електронний ресурс]: режим доступу: https://arxiv.org/abs/1511.08458
4. Numpy documentation [Електронний ресурс]: режим доступу: https://numpy.org/doc/
5. Tensorflow documentation [Електронний ресурс]: режим доступу: https://www.tensorflow.org/
6. Matplotlib documentation [Електронний ресурс]: режим доступу: https://matplotlib.org/
7. Keras documentation [Електронний ресурс]: режим доступу: https://keras.io/
8. The CIFAR-10 & CIFAR-100 datasets [Електронний ресурс]: режим доступу: https://www.cs.toronto.edu/~kriz/cifar.html
9. Activation Functions in Neural Networks [Електронний ресурс]: режим доступу: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6
10. “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions” Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie & Laith Farhan [Електронний

ресурс]: режим доступу: https://link.springer.com/article/10.1186/s40537-021-00444-8