

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

«Програмна реалізація інформаційної системи класифікації об'єктів по їх
зображенням»

Завідувач випускаючої кафедри

Довбиш А. С.

Керівник роботи

Петров С. О.

Студентки групи ІН – 62

Потапенко Д. О.

СУМИ 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2020 р.

ЗАВДАННЯ

до випускної роботи

Студента п'ятого курсу, групи ІН-62 спеціальності “Інформатика” денної форми навчання Потапенко Дарини Олексіївни.

Тема: «Програмна реалізація інформаційної системи класифікації об'єктів по їх зображенням»

Затверджена наказом по СумДУ № _____ от _____ 2020 р.

Зміст пояснювальної записки: 1) аналітичний огляд методів реалізації нейромережових систем класифікації об'єктів по їх зображенням; 2) постановка завдання й формування завдань дослідження; 3) опис основних положень, математичних моделей і критеріїв, що використовуються нейромережевою системою; 5) реалізація архітектури нейромережевої системи; 6) аналіз результатів моделювання.

Дата видачі завдання “ _____ ” _____ 2020 р.

Керівник випускної роботи _____ Петров С. О.

Завдання прийняв до виконання _____ Потапенко Д. О.

РЕФЕРАТ

Записка: 46 стор., 18 рис., 1 табл., 1 додаток, 10 джерел.

Об'єктом дослідження — є методи і алгоритми реалізації архітектур згорткових нейронних мереж.

Мета роботи — полягає в застосуванні концепції алгоритму глибокого навчання, а саме- згорткової нейронної мережі (CNN) в задачах класифікації зображень.

Методи дослідження — методи вирішення задач розпізнавання об'єктів з використанням штучних нейронних мереж.

Результати — вирішена задача розпізнавання об'єктів з використанням штучних нейронних мереж. В результаті роботи була реалізована нейромережева структура згорткового типу. Нейронна мережа складається з 12 шарів. Кількість нейронів зовнішнього шару мережі становить - 512 нейронів. Кількість нейронів внутрішнього шару - 10.

Ключові слова: ШТУЧНІ НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, ЗАДАЧІ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ.

ЗМІСТ

ВСТУП.....	1
1. ПОСТАНОВКА ТА АНАЛІЗ ЗАДАЧІ РОЗПІЗНАВАННЯ ТА КЛАСИФІКАЦІЇ ОБРАЗІВ.....	4
1.1. Огляд та порівняльна характеристика існуючих підходів до розпізнавання образів.....	4
1.2. Постановка задачі розпізнавання та класифікації образів методами нейронних мереж.....	7
1.3. Вибір моделі нейронної мережі для вирішення задачі розпізнавання образів.....	10
1.4. Постановка задачі.....	14
Висновки.....	14
2. СПЕЦИФІКАЦІЯ МОДЕЛІ НЕЙРОННОЇ МЕРЕЖІ.....	16
2.1. Модель нейронної мережі типу convolutional neural network.....	16
2.2. Специфікація характеристик нейронної мережі.....	20
2.3. Формулювання елементів вибірки.....	23
2.4. Математичне моделювання та параметри результуючої нейронної мережі.....	24
Висновки.....	26
3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ РЕЗУЛЬТУЮЧОЇ МОДЕЛІ.....	27

3.1. Побудова та ініціалізація нейромережі.....	27
3.2. Результати чисельного експерименту для задачі класифікації...	32
ВИСНОВКИ.....	35
СПИСОК ЛІТЕРАТУРИ.....	37
ДОДАТОК	39

ВСТУП

В сучасному світі застосування задач розпізнавання та класифікації об'єктів набуває все більшого значення. Особливо активно вони використовуються в системах управління та обробки інформації, системах прийняття рішень, автоматизованих системах та ін.

Системи розпізнавання графічної інформації відіграють важливу роль та є основними компонентами обчислювальних пристроїв, застосовуваних у різних сферах людської діяльності. Такі системи дозволяють значно спростити вирішення повсякденних задач, пов'язаних з обробкою графічної інформації, що й виділяє їм особливе місце у галузях

Технології розпізнавання та класифікації інформації можуть бути використані в найрізноманітніших сферах побутової та наукової діяльності людини. Послідовне перетворення та розподіл графічних об'єктів використовуються в сукупностях елементів забезпечення безпеки, перевірки якості доступу та керування, VR та системах пошуку інформації.

На разі, в існуючу епоху технологій, досить актуальними у виробництві стали нейромережеві системи розпізнавання тексту, відбитки пальців або обличчя людини та інших технологіях, що дозволяють підвищити рівень безпеки автентифікації, та забезпечити користувачів достатнім рівнем безпеки їх інформації. Вони використовуються в інтерфейсах програмного забезпечення, пов'язаних із безпекою та автентифікацією, а також використовуються для інших програм.

Таким чином, удосконалення реалізації розпізнавання комп'ютерними системами образів є актуальним. Застосування штучних нейромереж та

нейрокомп'ютерів є найбільш прогресивним напрямом вирішення задач розпізнавання образів.

Наразі існує значна кількість нейромережевих систем здатних до використання у вирішенні задач розпізнавання об'єктів.

Не зважаючи на значну кількість наукових досліджень що виконуються в цій області, наразі не існує найкращої в усіх аспектах моделі нейронної мережі.

Однією із найперспективніших архітектур на даний момент вважають згорткові нейронні мережі.

Перевагою згорткового типу мереж можна зазначити використання спільних вагів, що передбачає використання одного фільтру до кожного шару згортки що й підвищує точність роботи даної мережі.

Мета роботи — полягає в застосуванні концепції алгоритму глибокого навчання, а саме- згорткової нейронної мережі (CNN) в задачах класифікації зображень.

Для досягнення зазначеної мети **вирішено наступні завдання:**

1. Розглянути та проаналізувати існуючі методи вирішення задач розпізнавання;
2. Провести порівняльну характеристику штучних нейронних мереж для вирішення задач розпізнавання;
3. Обрати та детально описати використану в роботі архітектуру штучної нейронної мережі;
4. Створити математичну модель результуючої нейронної мережі;
5. Побудувати нейронну мережу;

6. Проаналізувати роботу та описати результати дослідження.

Об'єктом дослідження — є методи і алгоритми реалізації архітектур згорткових нейронних мереж.

Предметом дослідження — є реалізація та апробація результуючої моделі обраної нейронної мережі.

Методом вирішення поставленої задачі є вирішення задач розпізнавання об'єктів з використанням штучних нейронних мереж.

Результати дослідження — вирішена задача розпізнавання об'єктів з використанням штучних нейронних мереж. В результаті роботи була реалізована нейромережева структура згорткового типу. Нейронна мережа складається з 12 шарів. Кількість нейронів зовнішнього шару мережі становить - 512 нейронів. Кількість нейронів внутрішнього шару - 10.

1. ПОСТАНОВКА ТА АНАЛІЗ ЗАДАЧІ РОЗПІЗНАВАННЯ ТА КЛАСИФІКАЦІЇ ОБРАЗІВ

Людина постійно стикається з задачами розпізнавання, використовуючи органи чуття для збору інформації про об'єкти і мозок у якості класифікатора. З розвитком інформаційних та обчислювальних технологій виникла необхідність автоматизувати ці процеси. Саме тому розробка принципів і методів розпізнавання об'єктів стала одним із найактуальніших напрямків в галузях аналізу даних.

Розпізнавання образів - один з найсучасніших напрямків, пов'язаних з розробкою та побудовою принципів систем визначення приналежності даного об'єкту до одного із виділених класів. Існують різні способи реалізації таких систем на практиці. У даному розділі розглядається та проводиться порівняльний аналіз кожного з них.

1.1 Огляд та порівняльна характеристика існуючих підходів до розпізнавання образів.

В цілому виділяють три методи розпізнавання образів, кожен з яких має свої переваги та недоліки [1].

Метод перебору:

Суть цього методу досить проста та полягає в існуванні певної бази даних, для кожного об'єкту розпізнавання якої представлені всі можливі варіанти його відображення. Наприклад, для розпізнавання футбольного м'яча на фотографії за допомогою методу перебору можна розглядати базу даних, де міститься велика кількість зображень м'ячів під різними кутами, з різним масштабом, спотворенням і деформацією.

Перевагою даного методу можна зазначити його простоту, а саме - алгоритм порівнює зображення отриманого об'єкта з усіма, що знаходяться в базі даних і, на основі цих даних, видає результат.

Очевидним недоліком виступає неможливість розпізнавання об'єктів не зазначених в базі даних, оскільки дана система може розпізнати лише ті об'єкти, що містяться в ній і точність алгоритму напряму залежить від її наповненості.

Другий метод розпізнавання образів:

Другий метод оцінює характеристики отриманого образу за допомогою глибокого аналізу. Наприклад, для футбольного м'яча такими характеристиками можуть служити різні геометричні складові: форма, округлість, наявність кутів і т.д. Даний метод може використовуватися у вузьких напрямках, де необхідно розрізняти об'єкти по їх характеристикам, як графічним, так і звуковим (частотний, амплітудний аналіз звуку і т.д.), але для більш складних завдань використовуються більш точні методи аналізу.

Штучні нейронні мережі:

Застосування штучних нейронних мереж для розпізнавання образів є одним з найбільш популярних методів в силу своєї ефективності і продуктивності [2]. Він заснований на навчанні - на вході в нейронну мережу подається приклад для розпізнавання, який перетворюється в сигнал, що проходить по мережі, і в результаті на виході отримують відповідь. Якщо дану відповідь містить велику частку помилки, то відбувається коригування вагових коефіцієнтів всередині мережі, і процес починається спочатку і йде до тих пір, поки помилка мережі не буде мінімальною. Складність використання мереж полягає лише в правильній побудові її внутрішньої структури, яка буде

ефективна для вирішення даного завдання. Перевагою цього методу, як уже було згадано раніше, є його продуктивність і ефективність на тлі інших

Штучна нейронна мережа (ШНМ) - обчислювальна нелінійна модель, основою якої є нейронна структура мозку, здатна навчатися виконанню завдань: класифікації, передбачення, прийняття рішень, візуалізації та ін. завдяки розгляду та аналізу вже існуючих моделей-прикладів.

Розглянемо популярні архітектури нейронних мереж, які добре показали себе в задачах NLP і рекомендуються до використання.

Багатошаровий перцептрон:

Багатошаровий перцептрон (MLP) - нейронна мережа прямого поширення сигналу (без зворотних зв'язків), в якій вхідний сигнал перетворюється в вихідний, проходячи послідовно через кілька шарів. [3]

Багатошаровий перцептрон складається з 3 або більше шарів. Він використовує нелінійну функцію активації, часто тангенціальну або логістичну, яка дозволяє класифікувати лінійно нероздільні дані. Кожен вузол в шарі з'єднаний з кожен вузлом в наступному шарі, що робить мережу повністю пов'язаною. Така архітектура знаходить застосування в задачах розпізнавання мови і машинному перекладі.

Згорткова нейронна мережа:

Згорткова нейронна мережа є розвитком багатошарової повної нейронної мережі і націлена на ефективне розпізнавання зображень.

Згорткова нейронна мережа (CNN) містить один або більше об'єднаних шарів згортки. CNN використовує варіацію багатошарового перцептрона, розглянутого вище. Згорткові шари використовують операцію згортки для

вхідних даних і передають результат до наступного шару. Ця операція дозволяє мережі бути глибшою з меншою кількістю параметрів [4].

Ключовою особливістю і відмінністю згорткової нейромережі від персептрона є те, що нейрони шару не мають індивідуальних вагових коефіцієнтів, а використовують розділені ваги: матриці вагів невеликого розміру, які ще називають ядрами згортки. Таким чином, згорткова мережа має набагато меншу кількість параметрів налаштування у порівнянні з повнозв'язною мережею, з чого слідує більш висока продуктивність і економічність використовуваної пам'яті. Згорткові нейронні мережі є потужним інструментом в задачах машинного зору. Семантичний розбір, пошук парафраз, розпізнавання мови - теж додатки CNN.

В роботі розглядатиметься підхід до розпізнавання і класифікації об'єктів на основі згорткових нейронних мереж (ЗНМ, convolutional neural network).

1.2 Постановка задачі розпізнавання та класифікації образів методами нейронних мереж.

Задачі розпізнавання образів є, по суті, дискретним аналогом задачі пошуку оптимальних рішень (дискретного програмування). До них відноситься широкий клас задач, в яких за деякою, зазвичай неповною, нечіткою інформацією потрібно встановити, чи володіють досліджувані об'єкти (ситуації, явища) фіксованим кінцевим набором властивостей, що дозволяє віднести їх до певного класу. Іншою важливою областю застосування теорії розпізнавання образів є рішення задач прогнозування поведінки об'єктів або розвитку ситуації.

До задач прогнозування відносяться задачі технічної та медичної діагностики, геологічного прогнозування, прогнозування властивостей хімічних

сполук, сплавів та нових матеріалів, прогнозування врожаю і ходу будівництва великих об'єктів, виявлення лісових пожеж, управління виробничими процесами і т.д.

У загальному випадку, будь-яку задачу можна класифікувати як розпізнавання образів якщо при наявності відомих вихідних даних, за рахунок використання певного алгоритму (методики) потрібно отримати рішення (образ)

Наведемо деякі типові приклади задач розпізнавання:

Таблиця 1.1 Типові приклади задач розпізнавання

Тип задачі	Властивості
Класифікація	Віднесення пред'явленого об'єкта (ситуації) по його формалізованому опису до одного із заданих класів. [5]
Реалізація асоціативної (контекстно-адресуємої) пам'яті	Порівняння спотвореного уявлення образу з оригіналом (відновлення вихідного образу).
Кластеризація	Розбиття множини об'єктів (ситуацій) по їх формалізованим описам на систему непересічних підмножин (класів). [5]
Абстрагування і	Визначення набору характеристик для побудови

формалізований опис	формалізованого опису об'єктів розпізнавання; оцінка інформативності окремих характеристик і їх поєднань.
Приведення вихідних даних до виду, зручного для розпізнавання	Побудова формалізованого опису об'єкта розпізнавання
Прогнозування	Передбачення значень, характеристик або поведінки системи в майбутньому на підставі попередніх і поточних спостережень
Апроксимація функцій	Пошук функції, найбільш відповідної до набору експериментальних даних
Оптимізація	Пошук ефективного або оптимального рішення цільової функції в умовах дії обмежень
Управління	Розрахунок вхідного впливу, при якому система буде слідувати бажаної траєкторії або відповідати заданим параметрам

Одним з найбільш ефективних і поширених способів подання і вирішення перерахованих вище завдань є штучні нейронні мережі (ШНМ). Штучна нейронна мережа - математична модель, а також її програмне втілення,

побудоване за принципом організації та функціонування біологічних нейронних мереж. [6]

Використання ШНМ для вирішення задачі класифікації обумовлено такими факторами як:

1. Можливість вирішення задач в умовах невизначеності.

Завдяки здатності до навчання нейронна мережа здатна вирішувати завдання з невідомими закономірностями і залежностями між вхідними та вихідними даними, що дозволяє їй працювати з неповними даними.

2. Стійкість до шумів у вхідних даних.

Нейронна мережа здатна самостійно виявляти неінформативні для аналізу параметри і відсіювати їх, у зв'язку з чим втрачається необхідність у попередньому аналізі вхідних даних.

3. Гнучкість структури нейронних мереж.

Компоненти нейрокомп'ютерів - нейрони і зв'язки між ними - можна поєднувати різними способами. За рахунок цього один нейрокомп'ютер можливо застосовувати для вирішення різних задач, часто не пов'язаних між собою.

4. Висока швидкодія.

Вхідні дані обробляються багатьма нейронами одночасно, завдяки чому нейронні мережі вирішують завдання швидше.

1.3 Вибір моделі нейронної мережі для вирішення задачі розпізнавання образів.

Існують різні парадигми штучних нейронних мереж що відрізняються математичною моделлю штучного нейрона, способом організації нейронів, методами навчання та ін. Найбільше практичне використання знаходять штучні нейронні мережі, розроблені на основі багатошарового персептрона і згорткових нейронних мереж.

Багатошаривий персептрон:

Наразі під багатошаровим персептроном розуміють мережу прямого поширення сигналів без зворотного зв'язку, що складається з декількох шарів нейронів з сигмоїдальною функцією активації. Кожен нейрон має зв'язки з усіма нейронами попереднього шару і з усіма нейронами наступного шару. Нейрони що належать одному шару не мають зв'язків між собою. Усі шари багатошарового персептрона крім останнього називаються прихованими. Таким чином, багатошаровий персептрон має шарувату структуру (Рис.3).

Кожен шар складається з нейронів, які отримують вхідні сигнали від шару нейронів, розташованого нижче, і посиляють вихідні значення на шар, який відповідно знаходиться вище. Сполучні зв'язки в межах одного шару відсутні. Вхідні сигнали подаються одразу на перший шар з прихованими нейронами, оскільки вхідний шар служить лише для ініціалізації вхідного вектора і ніякої обробки сигналу в цьому шарі не відбувається.

Виходи прихованих нейронів в свою чергу розподіляються по наступному шару прихованих нейронів, і так відбувається до тих пір, поки не досягається останній шар прихованих нейронів.

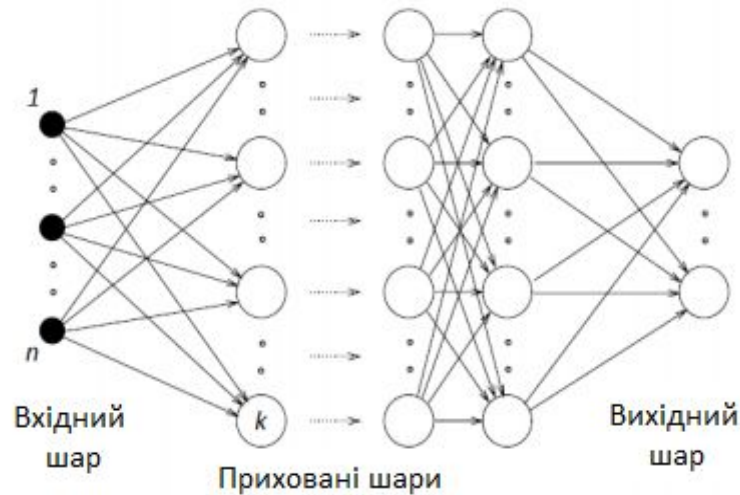


Рисунок 1:1 Структура багатошарового персептрона

Згорткова нейронна мережа:

Згорткова нейронна мережа (CNN) - це особлива архітектура штучної нейронної мережі, що повторює можливості ділянки кори головного мозку що відповідає за зір та має багатошарову структуру та здатна до більш ефективного розпізнавання складних структур.

Далі необхідно оглянути класичну структуру згорткової нейронної мережі детальніше. Мережа має багатошарову структуру. Після вхідного шару сигнал проходить декілька шарів згортки де на кожному шарі виконується послідовне чергування згортки і пулінгу. Чередування шарів дозволяє скласти «карти ознак» з яких, на кожному з наступних шарів карта зменшуватиметься в розмірах, але збільшуватиметься кількість каналів.

В практичному застосуванні це буде означати можливість розпізнавання складних підлеглих ознак. Після згорткових додатково додають персептрон, на вхід котрого будуть подаватися кінцеві карти ознак.

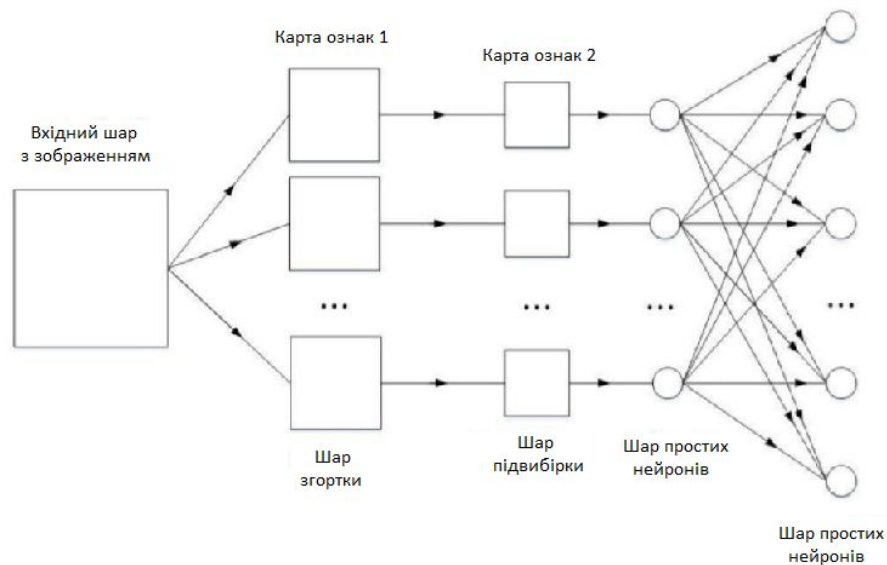


Рисунок 1.2 Структура згорткової нейронної мережі

Ключовою особливістю і відмінністю згорткової нейромережі від персептрона є те, що нейрони шару мають не індивідуальні вагові коефіцієнти, а використовують розділені ваги: матриці вагів невеликого розміру, які ще називають ядрами згортки. Таким чином, згорткова мережа має значно меншу кількість настроюваних параметрів у порівнянні з повнозв'язною мережею, з чого й слідує більш висока продуктивність і економічність використовуваної пам'яті.

В роботі розглядатиметься підхід до розпізнавання і класифікації об'єктів на основі згорткових нейронних мереж (ЗНМ, convolutional neural network).

Використання згорткового типу мереж обумовлено двома основними факторами:

- 1) зниження складності нейронної мережі та її навчання, в порівнянні з класичним багатозаровим персептроном, що є актуальним в області обробки та аналізу зображень;

2) підвищення стійкості розпізнавання до різних спотворень в порівнянні з класичними нейронними мережами та іншими методами класифікації зображень.

1.4 Постановка задачі.

Мета роботи полягає в застосуванні згорткової нейронної мережі в задачах класифікації зображень, а саме — реалізація нейромережевої структури для розпізнавання та класифікації об'єктів на зображеннях.

Для досягнення зазначеної мети **вирішено наступні завдання:**

1. Огляд та аналіз існуючих методів вирішення задач розпізнавання;
2. Створення математичної моделі результуючої нейронної мережі;
3. Побудова та ініціалізація нейронної мережі;
4. Аналіз та опис результатів дослідження.

1.5 Висновки.

В даному розділі нами було розглянуто та проведено порівняльний аналіз кожного з методів розпізнавання образів. Було проаналізовано та обрано підхід до вирішення задач розпізнавання об'єктів з використанням нейронних мереж.

Також з урахуванням всіх характеристик було обрано модель нейронної мережі для вирішення задачі розпізнавання образів - згорткову нейромережу. Перевагами даної мережі серед інших є застосування значно меншої кількості настроюваних параметрів, в зв'язку з чим їй притаманна більш висока продуктивність і економічність використовуваної пам'яті. Саме ці переваги стали вирішальними при обранні даної моделі.

В роботі розглядатиметься підхід до розпізнавання і класифікації об'єктів на основі згорткових нейронних мереж (ЗНМ, convolutional neural network).

2. СПЕЦИФІКАЦІЯ МОДЕЛІ НЕЙРОННОЇ МЕРЕЖІ

2.1 Модель нейронної мережі типу convolutional neural network.

«Згорткова нейронна мережа частіш за все представляє чередування згорткових, субдискретизуючих та повнозв'язних шарів вкінці, що можуть чередуватися довільним чином.

«У шарі згортки нейрони, які використовують одні і ті ж ваги, об'єднуються в карти ознак (feature maps), а кожен нейрон карти ознак пов'язаний з частиною нейронів попереднього шару. При обчисленні мережі виходить, що кожен нейрон виконує згортку деякої області попереднього шару (яка визначається безліччю нейронів, пов'язаних з даними нейроном)» [7].

Приклад архітектури згорткової нейронної мережі представлений на Рисунку 2.1.

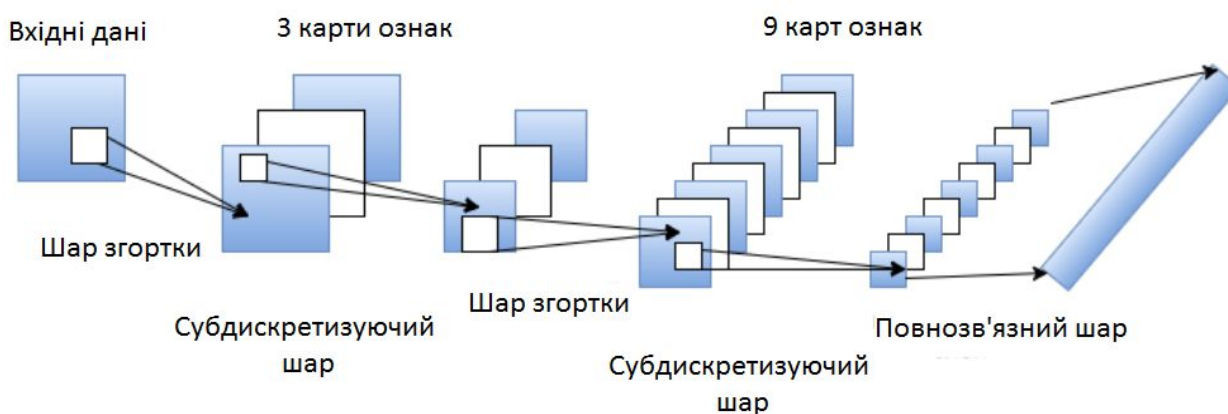


Рисунок 2.1 Архітектура згорткової нейронної мережі

Повнозв'язний шар:

Повнозв'язний шар є найпростішим і широко застосовуваним шаром нейронної мережі. Кожен нейрон в цьому шарі - перцептрон з нелінійною функцією активації. За функції активації прийнято використовувати або логістичну функцію виду: $f(x) = \frac{1}{1+e^{-x}}$, або гіперболічний тангенс $f(x) = A \cdot \tanh(Bx)$, де $A, B = \text{const}$. Кожний нейрон повнозв'язного шару з'єднується з виходами нейронів попереднього шару, підсумовує свої входи, перемножує їх на вагові коефіцієнти, додає поріг, а результат подає на вхід функції активації. Отримане значення і є виходом нейрону.

На рисунку 2.2 показано приклад повнозв'язного шару.

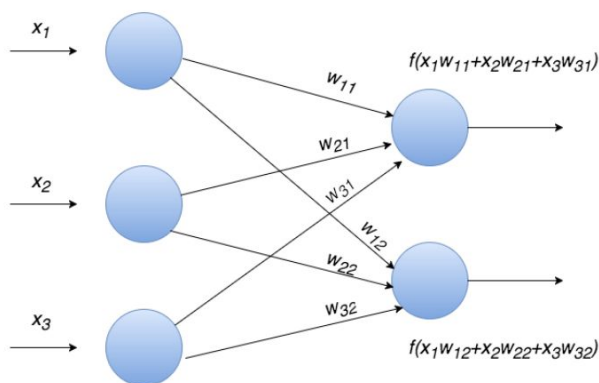


Рисунок 2.2 Повнозв'язний шар

Шар згортки:

На відміну від повнозв'язного, у шарі згортки нейрон поєднується з визначеною кількістю нейронів рівня що знаходився раніше, тобто згортковий шар аналогічний до застосування операції згортки, де використовується лише матриця вагів невеликого розміру (ядро згортки), яку «рухають» по всьому оброблюваному шару.

Ще одна особливість шару згортки заключається в тому, що він трохи зменшує зображення за рахунок крайових ефектів.

На Рисунку 2.3 зображено приклад шару згортки з ядром згортки розміру 3x3.

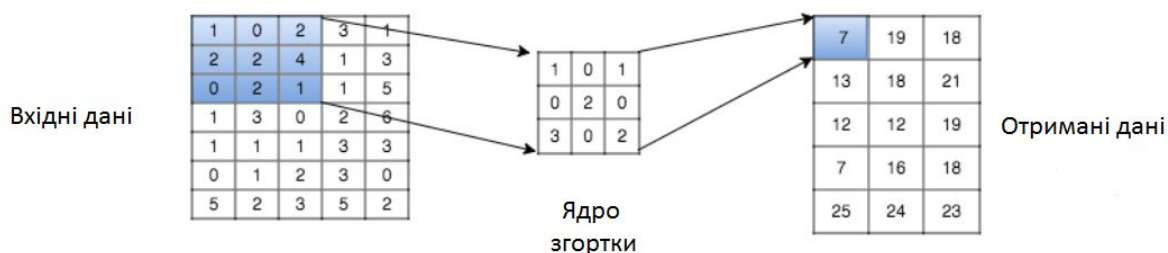


Рисунок 2.3 Шар згортки

Субдискретизуючий шар:

Субдискретизуючі шари зменшують розмірність зображення (зазвичай в декілька разів). Це виконується різними шляхами, але зазвичай використовується метод вибору максимального елемента (max-pooling) - карта ознак ділиться на ячейки, серед яких обираються максимальна за значенням.

На Рисунку 2.4 показаний приклад субдискретизуючого шару з методом вибору максимального елемента. Вхідний шар розбивається на клітинки заданого розміру і з кожної з них береться максимальне значення (іноді середнє значення по клітці).



Рисунок 2.4 Субдискретизуючий шар

Dropout шар:

Однією з найактуальніших та насущих проблем реалізації та експлуатації нейронних мереж – є проблема перенавчання.

«Перенавчання (overfitting) - одна з проблем глибоких нейронних мереж (Deep Neural Networks, DNN). Суть даної проблеми полягає в наступному: модель добре пояснює лише приклади з навчальної вибірки, адаптуючись до навчальних прикладів, замість того аби вчитися класифікувати приклади, які не брали участі у навчанні (втрачаючи здатність до узагальнення)» [8].

Останнім часом було запропоновано безліч варіантів вирішення даної проблеми, але саме метод Dropout зайняв лідуєче місце. Завдяки простоті використання та досягненням у практичних результатах метод Dropout набув значної популярності та визнаності в науковому світі.

Метод Dropout - один із найефективніших способів уникнути можливості перенавчання мережі.

«Dropout-регуляризація заключається зміні структури мережі: кожен нейрон викидається з певною ймовірністю p . За такої прорідженої мережі проводиться навчання, для решти вагів робиться градієнтний крок, після чого всі викинуті нейрони повертаються до нейромережі. Таким чином, на кожному

кроці стохастичного градієнту ми налаштуємо одну з можливих 2^N архітектур мережі, де під архітектурою ми розуміємо структуру зв'язків між нейронами, а через N визначаємо сумарну кількість нейронів» [9].

«При тестуванні нейромережі нейрони вже не викидаються, але вихід кожного нейрона множиться на $(1 - p)$ - завдяки цьому на виході нейрона ми будемо отримувати маточікування відповіді по всім 2^N архітектурам» [9].

На рисунку 2.5 зображено графічне представлення методу Dropout. Зліва (а) зображено мережу до використання методу Dropout, а праворуч (б) – та сама мережа після застосування техніки Dropout. Мережа (а), зображена ліворуч, використовується при тестуванні.

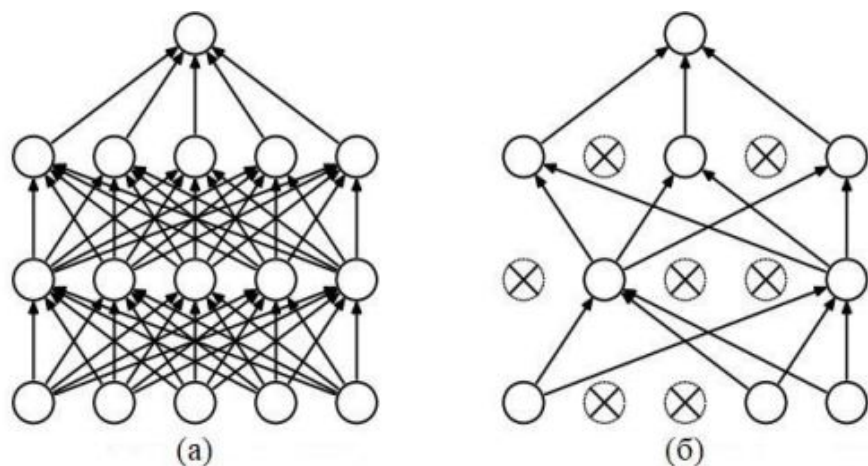


Рисунок 2.5 Техніка Dropout.

2.2 Специфікація характеристик нейронної мережі.

Convolution:

Згортковий (convolutional) шар є аналогом застосування декількох фільтрів до зображення, де під зображенням мається на увазі вихід попереднього шару.

Приклад програмної реалізації даного шару:

```
# Пеший шар згортки
model.add(Conv2D(32, (3, 3), padding='same',
input_shape=(32, 32, 3), activation='relu'))
```

Kernel (визначення, які діменшини використані в дипломній роботі)

Для шара згортки визначається його ядро (kernel). Ядро являє собою вікно або фільтр, який ковзає по всій області попередньої карти і знаходить певні ознаки об'єктів. Розмір ядра обирається таким чином, аби розмір карт шару згортки був парним, це дозволяє не втрачати інформацію при зменшенні розмірності в підвибірковому шарі. Зазвичай обирається в межах від 3x3 до 7x7 і обчислюється слідуючим чином:

$$M \times N \times C \times F$$

F - кількість каналів вихідного зображення, тобто кількість фільтрів.

$M \times N \times C$ - розмір кожного фільтра, де M і N - відповідно ширина і висота вікна фільтра, а C - кількість каналів вхідного зображення.

В даній роботі було використано ядро згортки розмірністю 3x3.

ReLU:

Останнім етапом згортки є застосування функції активації до всіх значень отриманої матриці. При роботі з зображеннями в якості функції активації практично завжди застосовується функція ReLU. ReLU має наступну формулу $f(x) = \max(0, x)$ і реалізує простий пороговий перехід в нулі. Основна причина вибору даної функції полягає в тому, що при навчанні глибоких мереж не виникає проблеми вкрай маленького градієнта при великих x. Також завдяки

обрізанню негативних значень ReLU збільшує розрідженість значень внутрішніх шарів мережі, що добре забезпечує нелінійність, а також є корисним з обчислювальної точки зору. Виникнення розрідженості відбувається при значеннях $x < 0$. Графік функції ReLU зображено на Рисунку 2.6.

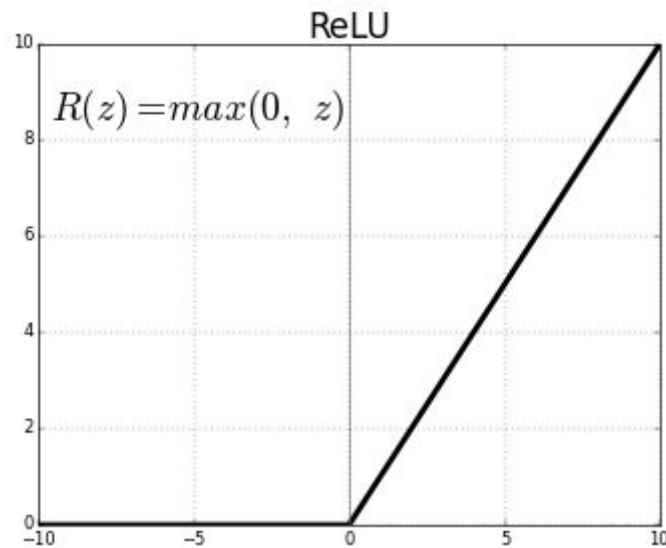


Рисунок 2.6 Графік функції ReLU

Pool:

Pooling служить для зменшення розміру матриці. Його функція полягає у покроковому зменшенні розміру зображення з метою зменшення кількості параметрів і обчислень в мережі, а також контролю перенавчання. Основним використовуваним методом pooling'a є max-pooling. Вхідний шар розбивається на клітинки заданого розміру і з кожної з них береться максимальне значення (іноді середнє значення по клітці). Таким чином, при розмірі вікна $M \times N$ зображення зменшується в $M \cdot N$ разів. В даній роботі використано шар пулінгу розмірністю 2×2 .

Приклад програмної реалізації даного шару:

```
model.add (MaxPooling2D (pool_size= (2, 2)))
```

Feed-forward neural network:

Мережі прямого поширення (Feed-forward neural network) (feed-forward мережі) - штучні нейронні мережі, в яких сигнал поширюється строго від вхідного шару до вихідного. У зворотному напрямку сигнал не поширюється. Такі мережі широко використовуються і цілком успішно вирішують певні класи завдань: прогнозування, кластеризація та розпізнавання.

2.3 Формування елементів вибірки.

Для навчання нейронної мережі використовувались зображення, представлені набором даних CIFAR10. Даний набір складається з 50000 навчальних образів та 1000 тестових зображень розміром 32x32. Зображення набору розділені на 10 класів, таких як: літаки, автомобілі, птахи, кішки, собаки, олені, жаби, коні, кораблі та вантажівки. Кожен клас містить 6000 кольорових зображень. Приклади зображень з навчальної вибірки представлені на Рисунку 2.7.

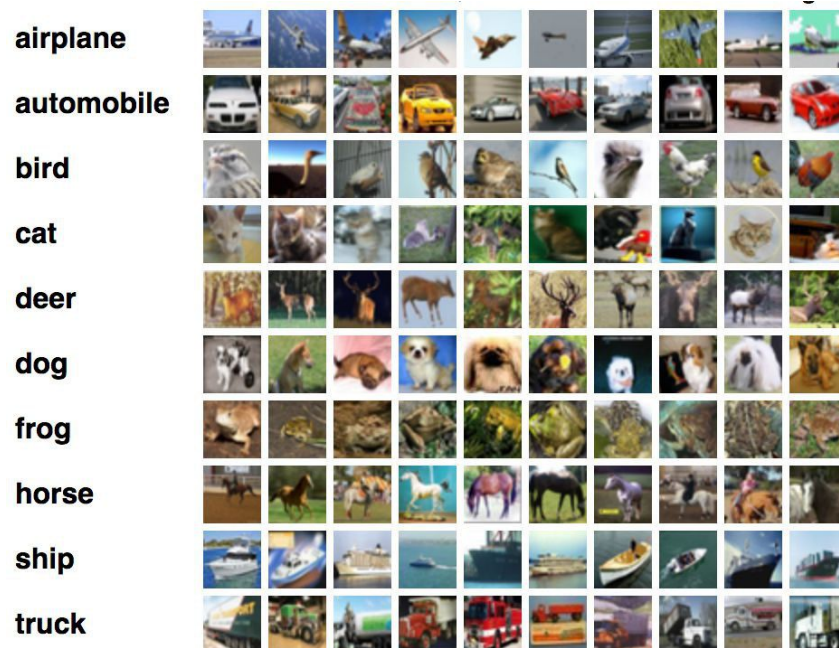


Рисунок 2.7 Приклади зображень навчальної вибірки.

2.4 Математичне моделювання та параметри результуючої нейронної мережі.

Для опису математичної моделі даної нейронної мережі будемо використовувати такі позначення.

Під $l \in [1 ; L]$ будемо розуміти розглядуваний наразі шар нейронної мережі, де $L = 2a+2$, $a \in \mathbb{Z}^+$ - кількість шарів в мережі. За N^l позначимо кількість карт ознак шару l , а за $f_l(\cdot)$ - функцію активації розглядуваного шару l . Також, під змінною y_n^l , будемо розуміти n -ну карту ознак шару l . [10]

Математична модель шару згортки:

Розглянемо шар згортки l . В архітектурі згорткової нейронної мережі l приймає непарні значення, тобто $l = 1, 3, \dots, 2a + 1$.

Тоді, для карти ознак n матиме місце наступне:

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

b_n^l - порогові ознаки, приєднувані до карти ознак n на шарі l .

V_n^l - список усіх рівнів шару $(l - 1)$, які з'єднуються з картою ознак n шару l

Таким чином карта ознак n шару згортки l буде обчислюватися за формулою:

$$y_n^l = f_l \left(\sum_{m \in V_n^l} y_m^{l-1} \otimes w_{m,n}^l + b_n^l \right) (2.1)$$

де оператор \otimes означає математичну операцію двовимірної згортки.

Припустимо що розмір вхідних карт ознак y_m^{l-1} дорівнює $H^{(l-1)} \times W^{(l-1)}$, а розмір застосовуваної до них згортки $w_{m,n}^l - r^l \times c^l$, тоді розмір вихідної карти ознак y_n^l обчислюється (рис.9):

$$(H^{(l-1)} - r^l + 1) \times (W^{(l-1)} - c^l + 1) \quad (2.2)$$

де $H^{(l-1)}$, $W^{(l-1)}$ ширина та висота вхідної карти ознак.

$w_{m,n}^l$ - згортка застосовувана до вхідної карти ознак m шару $(l - 1)$, $r^l \times c^l$ - розміри (ширина та висота) ядра матриці згортки.

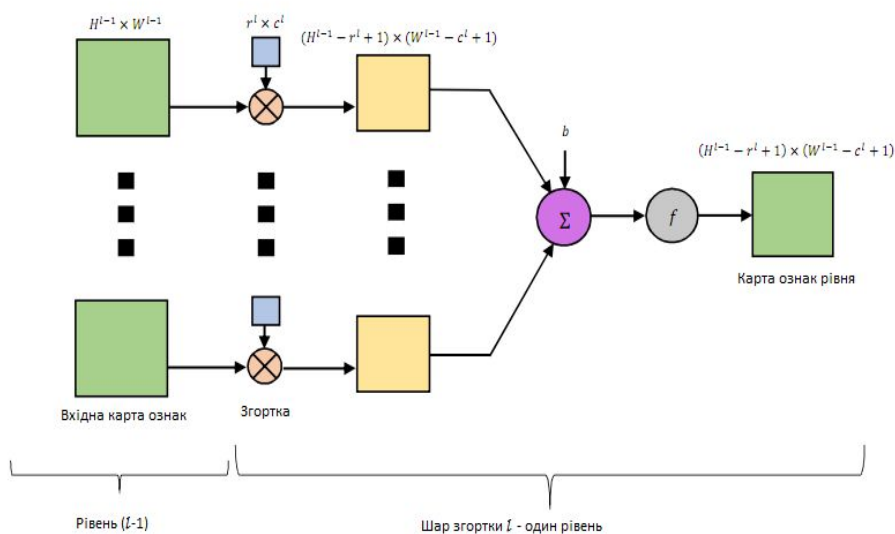


Рисунок 2.8 Схеми шару згортки l .

Математична модель субдискретизуючого шару:

Розглянемо субдискретизуючий шар l . В згортковій нейронній мережі l приймає непарні значення, тобто $l = 1, 3, \dots, 2a + 1$. Для карти ознак p введемо

слідуючі означення: $w_{m,n}^k$ - фільтр застосовуваний до p на шарі l , та b_n^l - додаткові порогові ознаки.

Розділимо карту ознак p ($l - 1$)-го шару на неперетинаючися блоки розміром 2×2 пікселі. Потім, підсумуючи значення чотирьох пікселів в кожному блоці, отримаємо матрицю $z_n^{l-1} = \{z_n^{l-1}(i, j)\}$, елементами якої будуть відповідні значення сум. Таким чином, формула для розрахунку значень елементів матриці виглядатиме:

$$z_n^{l-1} = y_n^{l-1}(2i - 1, 2j - 1) + y_n^{l-1}(2i - 1, 2j) + y_n^{l-1}(2i, 2j - 1) + y_n^{l-1}(2i, 2j) \quad (2.3)$$

Де y_n^{l-1} - карта ознак p шару згортки ($l - 1$), i, j - нейрони заданого шару.

Карта ознак p субдискретизуючого шару l обчислюється наступним чином:

$$y_n^l = f_l(z_n^{l-1} \times w_{m,n}^l b_n^l) \quad (2.4)$$

Де $f_l(.)$ - функція активації розглядуваного шару, z_n^{l-1} - матриця елементів субдискретизуючого шару, $w_{m,n}^l$ - фільтр застосовуваний до p на шарі l та b_n^l - додаткові порогові ознаки.

Аналізуючи все вищесказане стає можливим порахувати розмір $H^l \times W^l$ карти ознак y_n^l субдискретизуючого шару l (рис.2.9):

$$H^l = \frac{H^{l-1}}{2}, \quad W^l = \frac{W^{l-1}}{2} \quad (2.5)$$

Де H^l, W^l - ширина та висота карти ознак субдискретизуючого шару l .

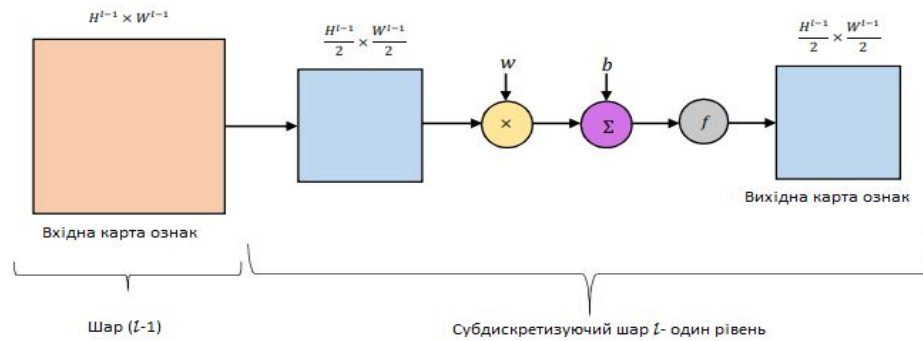


Рисунок 2.9 Схема субдискретизуючого шару.

Математична модель вихідного шару:

В даному підрозділі розглядатимемо вихідний шар L , який складається з одиничних нейронів. N^L - кількість нейронів на даному шарі. Як і в попередніх шарах $w_{m,n}^L$ - фільтр застосовуваний до карти ознак m останнього шару згортки для отримання переходу до нейрону n вихідного шару. Нехай b_n^L - порогове значення додане до нейрона n .

Користуючись наведеними значеннями, отримуємо формулу для розрахунку значень вихідного нейрона n :

$$y_n^L = f_L \left(\sum_{m=1}^{N^{L-1}} y_m^{L-1} w_{m,n}^L + b_n^L \right) \quad (2.6)$$

Де $f_L(\cdot)$ - функція активації розглядуваного шару L , N^{L-1} - кількість нейронів на даному шарі, y_m^{L-1} - карта ознак m шару згортки $(L - 1)$.

Таким чином, виходом згорткової нейронної мережі є вектор типу:

$$y=[y_1^L, y_2^L, \dots, y_{N^L}^L] \quad (2.7)$$

2.5. Висновки.

В даному розділі визначено порядок формування елементів вибірки, яка застосовується для побудови мережі, розроблено математичну модель та визначено параметри результуючої нейронної мережі.

Для навчання нейронної мережі використовувались зображення, представлені набором даних CIFAR10. Даний набір складається з 50000 навчальних образів та 1000 тестових зображень розміром 32x32. Зображення набору розділені на 10 класів, таких як: літаки, автомобілі, птахи, кішки, собаки, олені, жаби, коні, кораблі та вантажівки. Кожен клас містить 6000 кольорових зображень.

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ РЕЗУЛЬТУЮЧОЇ МОДЕЛІ.

3.1. Побудова та ініціалізація нейромережі.

В даній роботі була реалізована модель згорткової нейронної мережі. Мережа складається з 2 послідовно повторюючихся ділянок, а саме: 2 шари згортки, шар підвибірки, класифікатору, який містить повнозв'язний та вихідний шари.

Структура нейронної мережі представлений 12 шарами, а саме: 4 шари згортки, 2 шари підвибірки, 3 шари регуляризації, 2 повнозв'язних шари та шар Flatten- перетворення 2D зображень в плоскі. Також в мережі застосовано шар регуляризації Dropout з коефіцієнтами 0,5 та 0,25.

Перший та другий шари використаної нейронної мережі - згорткові. На вхід подається зображення розміром 32×32 пікселя, до якого застосовується згортка розміром 3×3 . Кількість використаних фільтрів - 32 (рис.3.1).

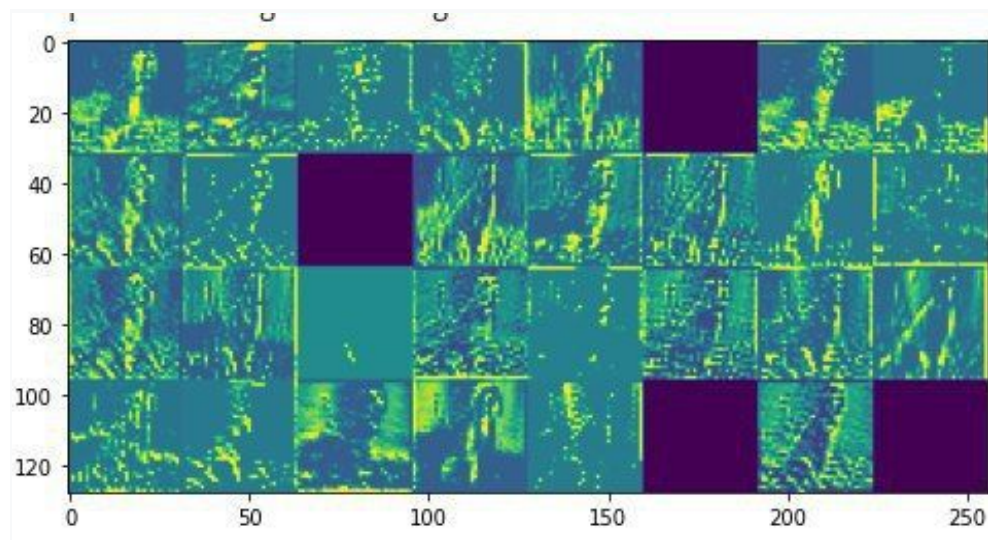


Рисунок 3.1 Фільтри першого та другого шарів мережі, налаштовані на розпізнавання найпростіших ознак.

Розмір вихідних карт ознак даних шарів згортки становить 28×28 пікселів.

Приклад програмної реалізації даного шару:

```
# Первый сверточный слой
model.add (Conv2D (32, (3, 3), padding='same',
                  input_shape= (32, 32, 3), activation='relu'))
# Второй сверточный слой
model.add (Conv2D (32, (3, 3), activation='relu', padding='same'))
```

Третій шар реалізованої мережі - шар Субдискретизації. Основною функцією даного шару було зменшення розмірності вхідного зображення і вибір пікселя з максимальним значенням серед кількох сусідніх. У даній мережі такий вибір здійснювався між 4 сусідніми пікселями (рис.3.2).

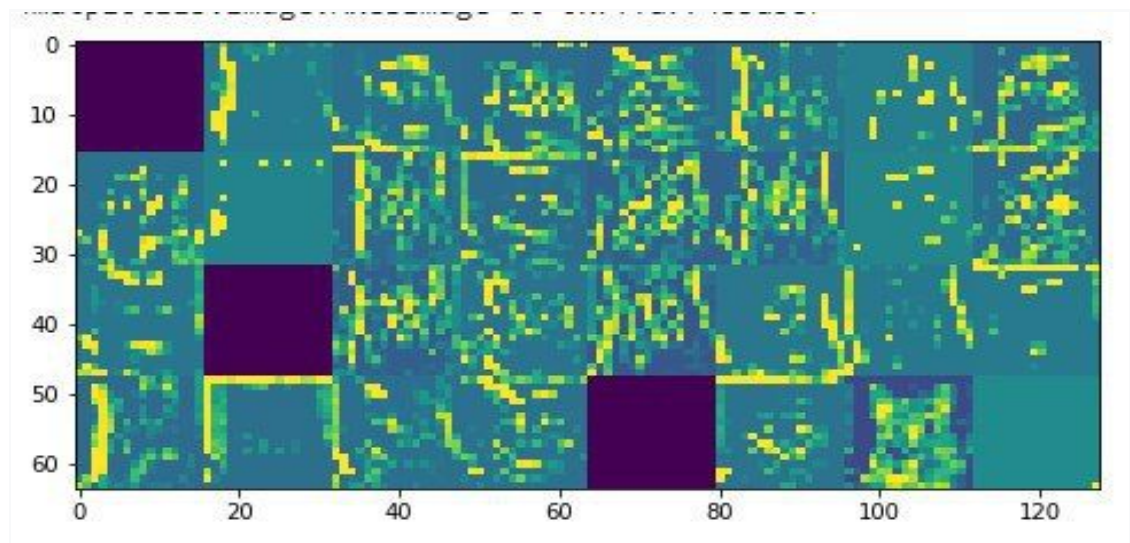


Рисунок 3.2 Застосування Субдискретизації до одного із зображень навчальної вибірки.

Приклад програмної реалізації даного шару:

```
# Первый слой подвыборки
model.add (MaxPooling2D (pool_size= (2, 2)))
```

В четвертому шарі було застосовано метод Dropout з коефіцієнтом 0,25 для уникнення перенавчання мережі.

Приклад програмної реалізації даного шару:

```
# Слой регуляризации Dropout
model.add (Dropout (0.25))
```

П'ятим і шостим шарами розглянутої нейронної мережі були повторно застосовані шари згортки, які брали на вхід 32 карти ознак попереднього шару. На даному етапі в якості фільтрів обиралися матриці розміру 3×3, формуючи на виході 64 карт ознак розміром 14×14 пікселів (рис.3.3).

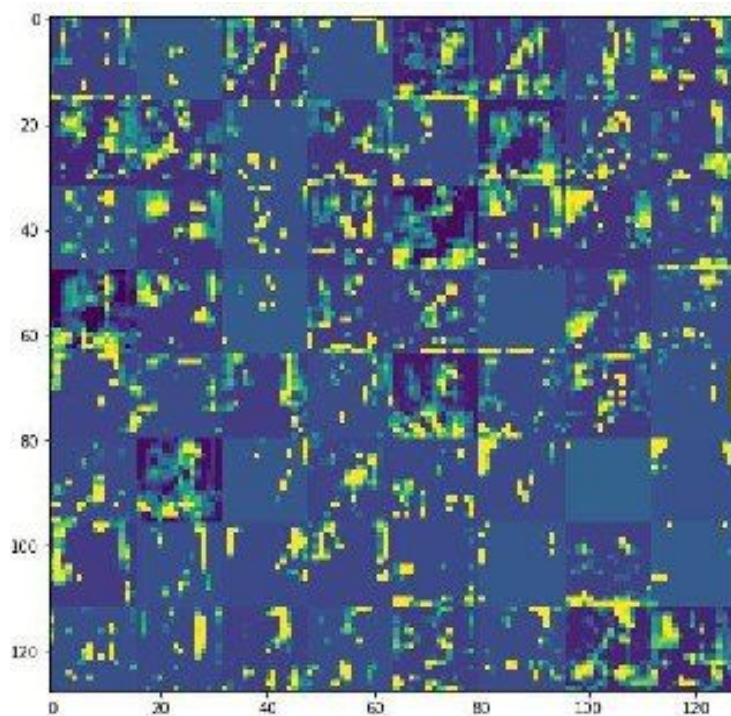


Рисунок 3.3 Карти ознак п'ятого і шостого шару мережі.

Приклад програмної реалізації даного шару:

```
# Третий сверточный слой
model.add (Conv2D (64, (3, 3), padding='same', activation='relu'))
# Четвертый сверточный слой
```

```
model.add (Conv2D (64, (3, 3), activation='relu'))
```

Сьомий, як і другий шар нейронної мережі, був шаром Субдискретизації. Виходом даного шару послужили 64 карт ознак - 7×7 пікселів (рис.3.4).

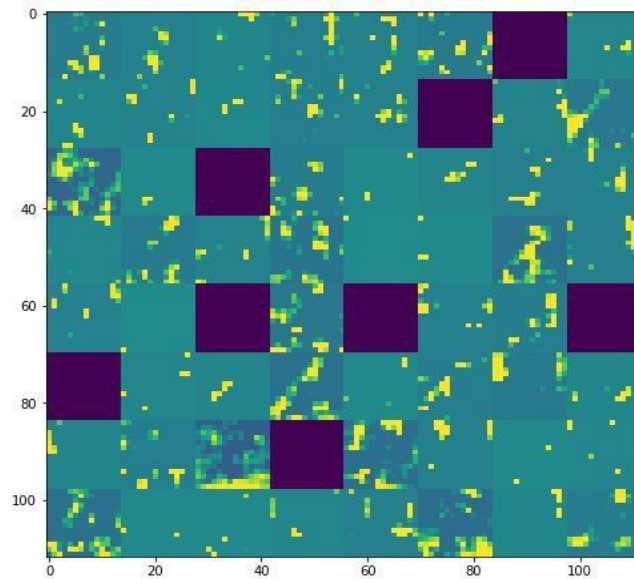


Рисунок 3.4 Карти ознак Субдискретизуючого шару.

Приклад програмної реалізації даного шару:

```
# Второй слой подвыборки
model.add (MaxPooling2D (pool_size= (2, 2)))
```

Восьмим (повторно) реалізовано Dropout шар з коефіцієнтом 0,25 для уникнення перенавчання мережі.

Приклад програмної реалізації даного шару:

```
# Слой регуляризации Dropout
model.add (Dropout (0.25))
```

Дев'ятим представлено шар, що перетворює дані двовимірної матриці в вектор, званий Flatten. Він дозволяє обробляти вихідні дані стандартними повнозв'язними шарами.

Приклад програмної реалізації даного шару:

```
# Слой преобразования данных из 2D представления в плоское
model.add (Flatten ())
```

Десятим шаром виступав повнозв'язний шар Dense, завданням цього шару було застосування нелінійної функції ReLU виду:

$$f(x) = \max(0; x) \quad (3.1)$$

Використання цієї функції є найзручнішим при навчанні згорткової нейронної мережі в силу простоти обчислення її градієнта

$$ReLU'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (3.2)$$

Це значно зменшує час навчання мережі, збільшуючи при цьому нелінійність виходу.

Приклад програмної реалізації даного шару:

```
# Полносвязный слой для классификации
model.add (Dense (512, activation='relu'))
```

Одинадцятий - Dropout шар з коефіцієнтом 0.5 (для уникнення перенавчання мережі).

Приклад програмної реалізації даного шару:

```
# Слой регуляризации Dropout
model.add (Dropout (0.5))
```

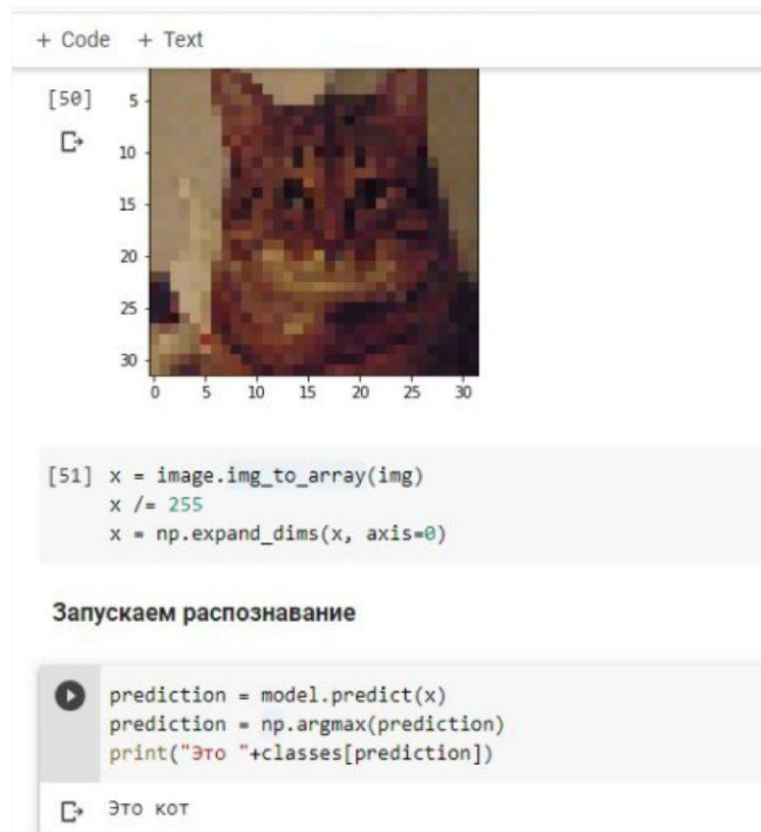
Дванадцятий - Вихідний повнозв'язний шар Dense, що містить 10 нейронів. Основним завданням даного шару була класифікація. За функцію активації даного шару обиралась softmax-функція виду:

$$f_i(\underline{x}) = \frac{e^{x_i}}{\sum_{j=1}^J e^{x_j}} \quad (3.3)$$

Приклад програмної реалізації даного шару:

```
# Вихідной пов'язаний шар
model.add(Dense(nb_classes, activation='softmax'))
```

Приклади роботи реалізованої згорткової нейронної мережі представлені на рисунках 3.5.



The screenshot shows a Jupyter Notebook interface. At the top, there are tabs for '+ Code' and '+ Text'. Below the tabs, a small image of a cat is displayed with a grid overlay indicating its dimensions (0 to 30 on both axes). Below the image, there is a code cell with the following code:

```
[50] 5
      10
      15
      20
      25
      30
      0 5 10 15 20 25 30
```

```
[51] x = image.img_to_array(img)
      x /= 255
      x = np.expand_dims(x, axis=0)
```

Below the code cell, there is a section titled 'Запускаем распознавание' (Starting recognition). Underneath, there is a code cell with the following code:

```
prediction = model.predict(x)
prediction = np.argmax(prediction)
print("Это "+classes[prediction])
```

At the bottom of the screenshot, the output of the code is displayed: 'Это кот' (This is a cat).

Рисунок 3.5 Приклад роботи реалізованої програми.

3.2. Результати чисельного експерименту для задачі класифікації.

Для навчання нейронної мережі використовувались зображення, представлені набором даних CIFAR10, а саме 50000 навчальних образів і 1000 тестових зображень розміром 32x32. Зображення набору розділені на 10 класів, таких як: літаки, автомобілі, птахи, кішки, собаки, олені, жаби, коні, кораблі та

вантажівки. Кожен клас містить 6000 кольорових зображень. Навчання мережі проводилося протягом 25 епох. Час навчання кожної епохи складає 10 секунд, а отже навчання нейронної мережі займатиме близько 5 хвилин.

```

- 10s - loss: 1.0547 - accuracy: 0.6517 - val_loss: 0.9107 - val_accuracy: 0.6784
[ ] Epoch 4/25
  - 10s - loss: 0.9195 - accuracy: 0.6750 - val_loss: 0.8044 - val_accuracy: 0.7162
[ ] Epoch 5/25
  - 10s - loss: 0.8329 - accuracy: 0.7046 - val_loss: 0.7405 - val_accuracy: 0.7414
Epoch 6/25
  - 10s - loss: 0.7702 - accuracy: 0.7286 - val_loss: 0.6809 - val_accuracy: 0.7582
Epoch 7/25
  - 10s - loss: 0.7265 - accuracy: 0.7450 - val_loss: 0.6598 - val_accuracy: 0.7690
Epoch 8/25
  - 10s - loss: 0.6716 - accuracy: 0.7628 - val_loss: 0.6377 - val_accuracy: 0.7790
Epoch 9/25
  - 10s - loss: 0.6333 - accuracy: 0.7773 - val_loss: 0.6676 - val_accuracy: 0.7706
Epoch 10/25
  - 10s - loss: 0.6004 - accuracy: 0.7869 - val_loss: 0.6300 - val_accuracy: 0.7830
Epoch 11/25
  - 10s - loss: 0.5661 - accuracy: 0.8008 - val_loss: 0.6161 - val_accuracy: 0.7912
Epoch 12/25
  - 10s - loss: 0.5353 - accuracy: 0.8104 - val_loss: 0.6167 - val_accuracy: 0.7888
Epoch 13/25
  - 10s - loss: 0.5146 - accuracy: 0.8167 - val_loss: 0.6086 - val_accuracy: 0.7936
Epoch 14/25
  - 10s - loss: 0.4863 - accuracy: 0.8286 - val_loss: 0.6169 - val_accuracy: 0.7954
Epoch 15/25
  - 10s - loss: 0.4739 - accuracy: 0.8310 - val_loss: 0.6045 - val_accuracy: 0.8000
Epoch 16/25
  - 10s - loss: 0.4493 - accuracy: 0.8406 - val_loss: 0.6640 - val_accuracy: 0.7844
Epoch 17/25
  - 10s - loss: 0.4277 - accuracy: 0.8480 - val_loss: 0.6058 - val_accuracy: 0.7966
Epoch 18/25
  - 10s - loss: 0.4127 - accuracy: 0.8514 - val_loss: 0.6237 - val_accuracy: 0.7974
Epoch 19/25
  - 10s - loss: 0.4031 - accuracy: 0.8566 - val_loss: 0.5987 - val_accuracy: 0.8052
Epoch 20/25
  - 10s - loss: 0.3868 - accuracy: 0.8618 - val_loss: 0.6458 - val_accuracy: 0.7964
Epoch 21/25
  - 10s - loss: 0.3659 - accuracy: 0.8683 - val_loss: 0.6068 - val_accuracy: 0.8062
Epoch 22/25
  - 10s - loss: 0.3624 - accuracy: 0.8703 - val_loss: 0.6355 - val_accuracy: 0.8018
Epoch 23/25
  - 10s - loss: 0.3445 - accuracy: 0.8774 - val_loss: 0.6285 - val_accuracy: 0.8078
Epoch 24/25
  - 10s - loss: 0.3333 - accuracy: 0.8811 - val_loss: 0.6384 - val_accuracy: 0.8040
Epoch 25/25
  - 10s - loss: 0.3333 - accuracy: 0.8811 - val_loss: 0.6384 - val_accuracy: 0.8040

```

Рисунок 3.6 Звіт з реалізації процесу навчання мережі

Тестування нейронної мережі відбувалося на 10 тестових прикладах. Алгоритм програми показав високу ефективність у 8 з 10 тестових випадків. Точність роботи на тестових даних складає 80.13%, це доводить те що даний метод класифікації виявився досить точним та підтверджує доцільність застосування використаної в даній роботі моделі нейронної мережі.

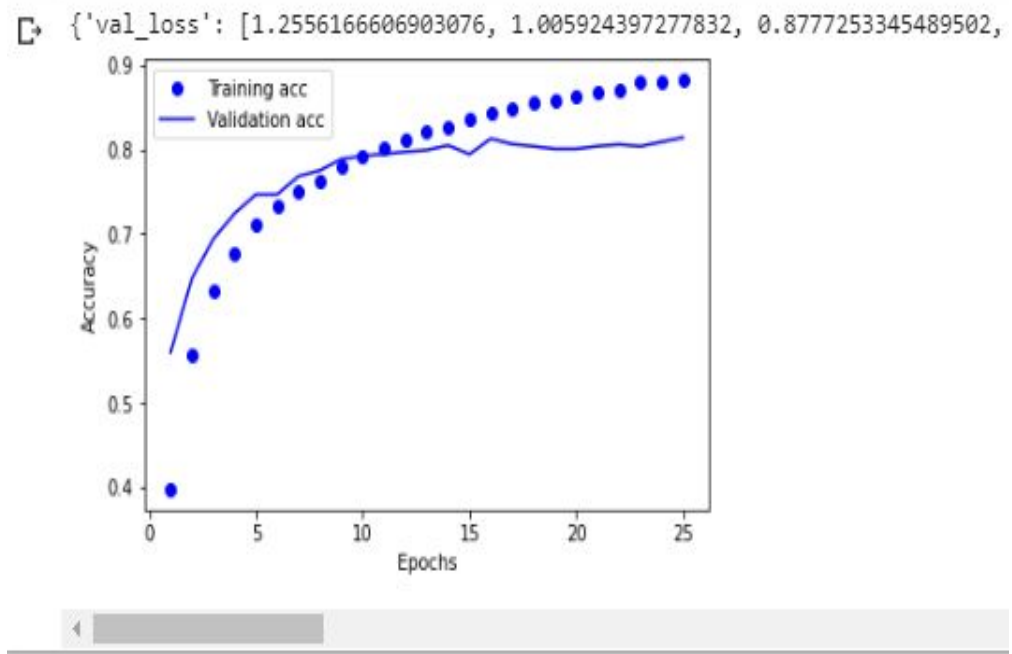


Рисунок 3.7 Графік точності навчання кожної епохи

ВИСНОВКИ

Ефективністю застосування нейронних мереж є їх спроможність до навчання на основі даних оточуючого середовища. В результаті навчання нейронні мережі здатні підвищити свою продуктивність. Процес підвищення їх продуктивності відбувається за певних умов з плином часу за допомогою інтерактивного процесу корекції синаптичних вагів нейронів мережі. За ідеальних умов нейронна мережа отримує знання із навколишнього середовища на кожній ітерації процесу свого навчання.

В роботі проаналізовано існуючі підходи до розпізнавання образів. За результатами проведеного аналізу обрано найоптимальнішу модель нейронної мережі для вирішення задачі розпізнавання - згорткову нейромережу. Перевагами даної мережі серед інших є застосування значно меншої кількості настроюваних параметрів, в зв'язку з чим їй притаманна більш висока продуктивність і економічність використовуваної пам'яті. Саме ці переваги стали вирішальними при обранні даної моделі.

В роботі розглянуто підхід до розпізнавання і класифікації об'єктів на основі згорткових нейронних мереж (ЗНМ, convolutional neural network).

Обрану базову модель нейронної мережі було навчено на десяти вхідних наборах даних. Для навчання нейронної мережі використовувались зображення, представлені набором даних CIFAR10. Даний набір складається з 50000 навчальних образів та 1000 тестових зображень розміром 32x32. Зображення набору розділені на 10 класів, таких як: літаки, автомобілі, птахи, кішки, собаки, олені, жаби, коні, кораблі та вантажівки. Кожен клас містить 6000 кольорових зображень.

Результатом роботи стала реалізована нейромережева структура згорткового трипу для вирішення задач розпізнавання об'єктів. Тестування

нейронної мережі відбувалося на 10 тестових прикладах. Алгоритм програми показав високу ефективність у 8 з 10 тестових випадків. Це підтверджує доцільність застосування використаної в даній роботі моделі нейронної мережі.

Таким чином, враховуючи вищевикладене, задача розпізнавання об'єктів, зазначена даною дипломною роботою розкрита в повній мірі.

СПИСОК ЛІТЕРАТУРИ

1. Чичварин, Н. В. Распознавание образов / Н. В. Чичварин [Электронный ресурс] / Национальная библиотека им. Н. Э. Баумана.
2. Сердюков, В. И. Использование элементов искусственного интеллекта для повышения надежности технических изделий / В. И. Сердюков, Н. А. Сердюкова, С. И. Шишкина // Вестник машиностроения. — 2017. — № 10. — С. 29-32
3. <http://ru.wikipedia.org/wiki/>
4. Воронцов, К. В. Курс лекций по машинному обучению / К. В. Воронцов. — 2015 Шеннона и Дж. Маккарти. — М. : Изд-во иностр. лит., 1956. — С. 363 – 384. (Перевод английской статьи 1943 г.).
5. Девід Е. Голберг «Дзен та мистецтво генетичного алгоритму». 3rd International Conference on Genetic Algorithms, pp. 80-85.
6. Мак-Каллок У.С. Логическое исчисление идей, относящихся к нервной активности / У.С. Мак-Каллок, В. Питтс ; // Автоматы / под ред. К. Э.
7. Yann LeCun Leon Bottou, Y. B. Gradient-based learning applied to document recognition / Yoshua Bengio Yann LeCun, Leon Bottou, Patrick Haffner // IEEE. — 1998
8. Кононюк Анатолий Ефимович Общая теория распознавания, стр. 349 <http://ecat.diit.edu.ua/ft/Recognition1.pdf>

9. Карпович А. В. Використання згорткових нейронних мереж для задачі класифікації текстів // Міжнародний науковий журнал "Інтернаука". — 2018. — №14. <https://doi.org/10.25313/2520-2057-2018-14-4105>
10. Son Lam Phung, Abdesselam Bouzerdoun. MATLAB Library for Convolutional Neural Networks. Technical report. <http://www.uow.edu.au/~phung/>

ДОДАТОК

Код програмної реалізації реалізованої моделі:

На даному етапі виконується імпорт необхідних модулів та окремих їх об'єктів для подальшого використання в роботі:

```
#Імпорт модуля для роботи з багатовимірними масивами, матрицями
import numpy as np
#Імпорт набору даних cifar10 для навчання нейромережі
from keras.datasets import cifar10
#Моделі нейронної мережі
from keras.models import Sequential
#Шари нейронної мережі
from keras.layers import Dense, Flatten
from keras.layers import Dropout
from keras.layers.convolutional import Conv2D, MaxPooling2D
#Додаткові утиліти, що використовуватимуться для подальшого перетворення даних
from keras.utils import np_utils
from keras.preprocessing import image
#Модулі для відображення графіків
import matplotlib.pyplot as plt
from scipy.misc import toimage
%matplotlib inline
```

Вводимо змінні з визначення основних характеристик мережі:

```
# Розмір міні-вибірки
batch_size = 128
# Кількість класів зображення
nb_classes = 10
# Кількість епох навчання
nb_epoch = 25
# Розмір зображення
img_rows, img_cols = 32, 32
# Кількість каналів зображення: RGB
img_channels = 3
# Назви класів з набору даних CIFAR-10
classes= ['літак', 'автомобіль', 'птаха', 'кіт', 'олень', 'собака',
'жаба', 'кінь', 'корабель', 'вантажівка']
```

Завантажуємо тестові дані Cifar10 для навчання мережі:

```
(X_train, y_train), (X_test, y_test) = cifar10.load_data ()
```

Огляд тестового прикладу даних:

```

n = 11
plt.imshow (toimage (X_train[n]))
plt.show ()
print ("Номер класса:", y_train[n])
print ("Тип об'єкта:", classes [y_train[n][0]])

```

Нормалізуємо дані:

```

# Надання змінним значень типу флот
X_train = X_train.astype ('float32')
X_test = X_test.astype ('float32')
# Значення змінних знаходяться в діапазоні від 0 до 255
X_train /= 255
X_test /= 255

```

Перетворюємо вірні відповіді в формат one hot encoding:

```

#Перетворення масиву в число від 0 до 10
Y_train = np_utils.to_categorical (y_train, nb_classes)
Y_test = np_utils.to_categorical (y_test, nb_classes)

```

Створюємо нейронну мережу:

```

# Створюємо послідовну модель
model = Sequential ()
# Перший шар згортки
model.add (Conv2D (32, (3, 3), padding='same',
                  input_shape= (32, 32, 3), activation='relu'))
# Другий шар згортки
model.add (Conv2D (32, (3, 3), activation='relu', padding='same'))
# Перший шар підвибірки
model.add (MaxPooling2D (pool_size= (2, 2)))
# Шар регуляризації Dropout
model.add (Dropout (0.25))
# Третій шар згортки
model.add (Conv2D (64, (3, 3), padding='same', activation='relu'))
# Четвертий шар згортки
model.add (Conv2D (64, (3, 3), activation='relu'))
# Другий шар підвибірки
model.add (MaxPooling2D (pool_size= (2, 2)))
# Шар регуляризації Dropout
model.add (Dropout (0.25))
# Шар перетворення даних з 2D уявлення в плоске
model.add (Flatten ())
# Повнозв'язний шар для класифікації
model.add (Dense (512, activation='relu'))
# Шар регуляризації Dropout
model.add (Dropout (0.5))
# Вихідний повнозв'язний шар
model.add (Dense (nb_classes, activation='softmax'))

```

Друкуємо інформацію про мережу :

```

print (model.summary ())

```


Компілюємо модель:

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

Навчаємо нейронну мережу:

```
history = model.fit (X_train, Y_train,
                    batch_size=batch_size,
                    epochs=nb_epoch,
                    validation_split=0.1,
                    shuffle=True,
                    verbose=2)
```

Оцінюємо якість навчання мережі:

```
# Оцінюємо якість навчання моделі на тестових даних
# Виводимо на екрані процент якості навчання
scores = model.evaluate (X_test, Y_test, verbose=0)
print ("Точность работы на тестовых данных: %.2f%%" % (scores [1]*100))

# Створюємо графік та виводимо процент якості на кожній епохі навчання
history_dict = history.history
print (history_dict)
acc_values = history_dict ['accuracy']
val_acc_values = history_dict ['val_accuracy']
epochs = range (1, len (acc_values) + 1)
plt.plot (epochs, acc_values, 'bo', label='Training acc')
plt.plot (epochs, val_acc_values, 'b', label='Validation acc')
plt.xlabel ('Epochs')
plt.ylabel ('Accuracy')
plt.legend()
plt.show()
```

Зберігаємо навчену нейронну мережу:

```
model_json = model.to_json ()
json_file = open ("cifar10_model_d.json", "w")
json_file.write (model_json)
json_file.close ()
model.save_weights ("cifar10_model_d.h5")
!ls
files.download ("cifar10_model_d.json")
files.download ("cifar10_model_d.h5")
```