Лабораторна робота №1

**Тема**: Згорткові нейронні мережі на мові Python.

**Мета**: Одержати практичні навички для проектування та використання згорткових нейронних мереж на мові програмування Python з використанням бібліотеки Tensorflow.

1 Теоретичні відомості

Згорткові нейронні мережі (Convolutional neural networks - CNNs) подібні до звичайних штучних нейронних мереж. Вони також складаються з нейронів, які мають певні ваги і зміщення. Проте вхідними даними для них є зображення. Тобто вхідні дані можна подати у трьох вимірах – висота, ширина та кількість каналів. В той час як звичайні нейронні мережі приймають на вхід одномірний вектор. Саме через це CNN-мережі стали такими популярними. Оскільки такий підхід значно зменшує кількість параметрів в самій мережі і, відповідно, зменшує час навчання.

Типова згорткова нейронна мережа складається із таких шарів: згорткові, субдискретизуючі та повнозв’язні шари. Приклад такої мережі зображено на рисунку 1.1.

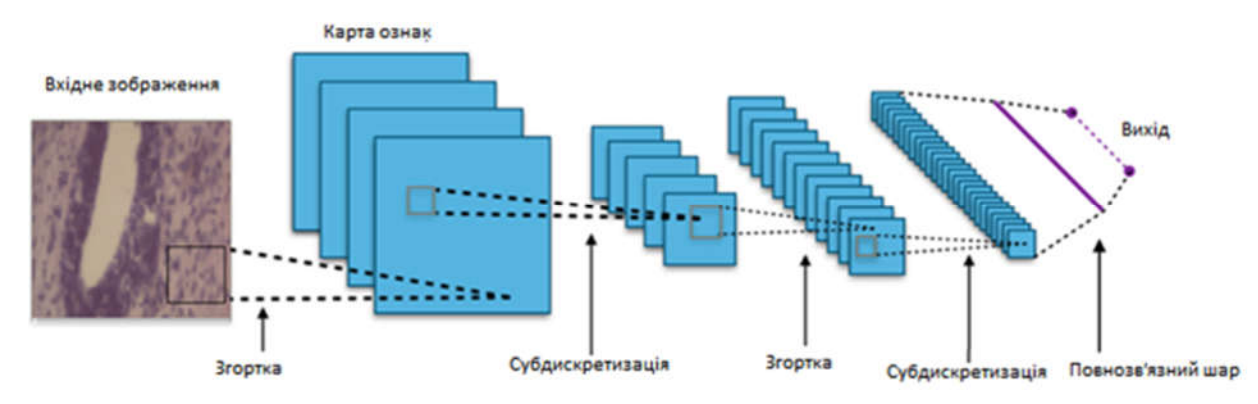


Рисунок 1.1 – Загальна структура згорткової нейронної мережі

*Згортковий шар.* Основними гіперпараметрами даного шару є розмір ядра (k - kernel size), кількість вхідних та вихідних фільтрів (f - filters), крок (s - stride) та доповнення нулями (p - padding). Принцип роботи полягає в наступному: пікселі ядра поелементно перемножуються із вхідними пікселями зображення та сумуються. Ядро переміщується із кроком, що дорівнює параметру stride, поки не «пройде» все вхідне зображення. Вихідний об’єм обчислюється за формулою O = (W-K+2P)/S+1.

Приклад. Нехай маємо вхідне зображення розміром 32x32x3 (W - ширина, H - висота, C - кількість каналів). Застосуємо згортковий шар k=3, s=1, p=0, f=96. Тоді отримаємо вихідне зображення розміром 30х30х96.

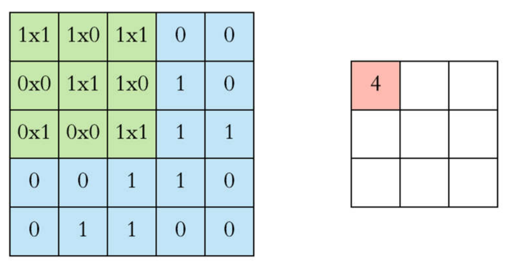


Рисунок 1.2 – Згортка зображення

*Субдискретизація.* Зазвичай цей шар використовується для зменшення розмірності зображення. В більшості випадків використовують максимальний пулінг, який вибирає максимальні значення із вхідного об’єму. Гіперпараметрами є розмір ядра та крок. Вихідний об’єм обчислюється за формулою O = (W-K)/S+1.

Приклад. Зображення 30х30х96 пікселів після застосування пулінгу буде мати об’єм 14х14х96 пікселів.

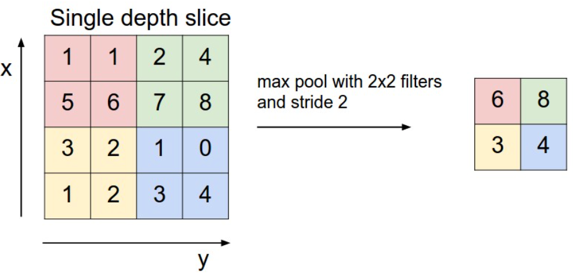


Рисунок 1.3 – Максимальний пулінг

*Повнозв’язний шар.* Цей шар застосовується в якості вихідного шару в задачах класифікації.

Детальнішу інформацію можна переглянути за посиланням: <https://cs231n.github.io/convolutional-networks>

2. Хід роботи.

* Скопіювати Colab Notebook за посиланням: <https://colab.research.google.com/drive/1cx95kKLN8-k60NYhPORBfvGopNYBQLxQ?usp=sharing>
* Експериментальним шляхом дослідити як змінюється точність класифікації при додаванні/видаленні шарів, коригуванні гіперпараметрів. Порівняти час навчання на CPU та GPU.

3. Структура звіту лабораторної роботи.

* Титульна сторінка.
* Тема та мета роботи.
* Код програми розв’язку індивідуального завдання.
* Копії екранів роботи програми.
* Висновки.

4. Контрольні запитання

1. Що таке CNN?

2. Які типи шарів використовуються в CNN та як вони працюють?

3 Відомі архітектури згорткових мереж.

4. Від чого залежить точність класифікації?

5. Що таке overfitting, underfitting та як з ними боротися?