ОПТИМІЗАЦІЯ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ЧЕРЕЗ РЕАЛІЗАЦІЮ ІДЕЇ РАННЬОГО ВИХОДУ

Кияк Орест-Теодор

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

orest-teodor.kyiak@lnu.edu.ua

Вступ. Більшість згорткових нейронних мереж (Convolutional Neural Networks, CNN) побудовані на основі принципу послідовного опрацювання усіх внутрішніх шарів нейронної мережі як при тренуванні, так і при тестуванні моделі. Однак у деяких випадках використання всіх шарів моделі є надлишковим і може негативно вплинути на результати її роботи. Це зокрема стосується задач класифікації зображень у сфері комп'ютерного зору, коли серед зображень зустрічаються такі, що не входять до досліджуваних класів, тобто є надлишковими даними. Тому розробка ефективних моделей, позбавлених зазначених недоліків, є актуальною проблемою. У цьому відношенні значної популярності набуває ідея впровадження у модель нейронної мережі так званого блоку "раннього виходу" (Early-Exit, EE) [1], [2], [3], який відповідатиме за відсіювання надлишкових даних. Ефективна реалізація таких моделей на практиці дасть змогу в наперед визначених місцях обчислювального графа динамічно обирати шлях під час отримання результату.

Метою даної роботи є модернізація CNN з VGG-19 архітектурою [4] шляхом впровадження в неї EE-блоку та дослідження характеристик побудованої нейронної мережі на задачах класифікації зображень.

Розроблена модель. Для проведення досліджень за основу було використано архітектуру згорткової нейронної мережі VGG-19 [4]. У модель, у якій реалізовано таку архітектуру, після першого блоку в згортковій основі було додано користувацький EE-блок (див. рис. 1). Його головна мета полягає у відсіюванні надлишкових зображень на ранніх

етапах обчислень, щоб зекономити ресурси моделі при класифікації необхідних зображень.

ЕЕ-блок створений на основі шарів максимізаційного агрегування (MaxPooling) та повністю зв'язних шарів (Fully-Connected, Dense), структура яких аналогічна до шарів у VGG-19 архітектурі. Також цей блок складається з шару, який вирівнює форму результату попереднього шару (Flatten). Це зроблено з метою застосування на останньому етапі повністю зв'язного шару, який на виході буде давати результат розподілу ймовірностей бінарної класифікації за допомогою функції активації *softmax*.

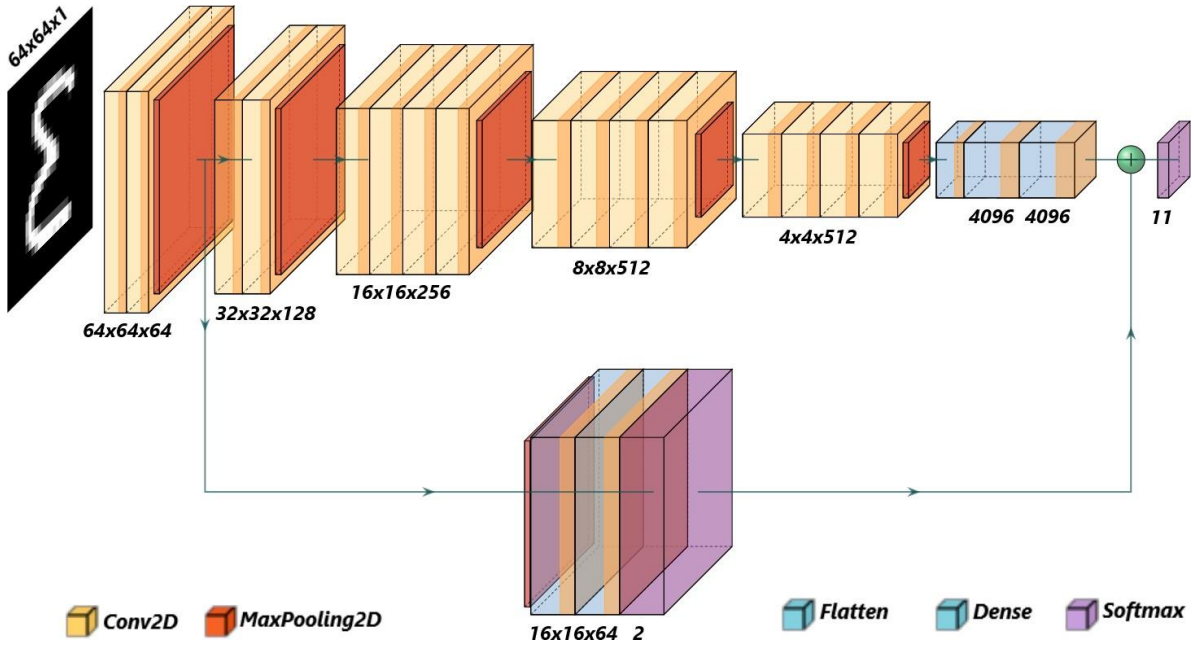

Рис. 1: Модернізована VGG-19 архітектура з реалізованим ЕЕ-блоком.

Функція втрати при виконанні ЕЕ-блоку під час тренування основана на обчисленні бінарної взаємної ентропії (Binary Crossentropy). Ця функція втрат визначає похибку при класифікації зображень між двома класами залежно від їхнього справжнього набору даних. Класифікація відбувається на ті, що належать до головного датасету і потребують додаткового опрацювання, або на ті, що належать до надлишкового набору даних і мають бути відсіяні ЕЕ-блоком. Формулу бінарної взаємної ентропії можна подати у вигляді:

$$L_1 = -\log\left(\frac{e^{k_p}}{e^{k_1} + e^{k_2}}\right), \quad (1)$$

де k_p – це елемент логіт-вектору, який відповідає індексу розміщення одиниці у One-Hot Encoded векторі представлення правильної категорії для відповідного зображення. У цьому випадку логіт-вектор – це вектор передбачених значень класифікації вхідних даних, елементи якого не є нормалізовані.

Формула для обчислення функції втрати при головній класифікації моделі відносно $(C-1)$ категорії має вигляд:

$$L_2 = -\log\left(\frac{e^{k_p}}{\sum_c^{C-1} e^{k_c}}\right), \quad (2)$$

При тренуванні кінцеве значення функції втрати відносно C класів, з яких $(C-1)$ категорій є головними і 1 надлишкова, обчислюють за формулою:

$$L = L_1 + \delta L_2\,, \quad (3)$$

де L_1 — бінарна взаємна ентропія відносно двох класів залежно від початкового датасету, L_2 — взаємна ентропія категорій для головної класифікації моделі відносно $(C-1)$ класу, δ — параметр, який набуває значення 0, якщо розглядається множина зображень з надлишкового класу, і 1, якщо зображення є з головного набору даних і потребують використання всіх шарів моделі.

Набори даних. При проведенні досліджень було розглянуто 3 об'єднання 6 наборів даних: MNIST, Omniglot, CIFAR-10, CIFAR-100, SVHN, GTSRB, які містять зображення $C = 11$ класів. У ролі головних були застосовані MNIST, CIFAR-10, SVHN, оскільки вони містять зображення, поділені між 10 різнними класами. Набори даних Omniglot, CIFAR-100, GTSRB були використані як надлишкові, тобто зображення в кожному з них були зведені до одного надлишкового класу, який необхідно відсіяти на етапі опрацювання EE-блоку.

Опис програмної реалізації. Було розроблено клас моделі на основі архітектури VGG-19, клас загального блоку в моделі, яка реалізує VGG-19 архітекутру та клас EE-блоку, який буде додаватись у згадану модель. У кожному класі також визначено метод *call()*, який описує послідовність виконання шарів цього класу під час прямого проходження. Також у класі моделі було перевизначено методи кроку тренування та тестування, які описують поведінку моделі при опрацюванні нею відповідних даних. Для реалізації програми було використано мову Python, зокрема бібліотеку Tensorflow 2 для побудови нейронних мереж і бібліотеку NumPy для роботи з векторними даними. Програму написано на платформі Kaggle з використанням GPU NVIDIA T4(x2).

Результати. Результати експериментів апробації моделі, реалізованої на основі VGG-19 архітектури з впровадженим EE-блоком, продемонстровано на рис. 2 і в таблицях 1-4. Як видно з графіків, модель з EE-блоком демонструє подібні результати до моделі без його використання відносно значення функції втрат і точності, якщо розглядати об'єднаний датесет MNIST і Omniglot. Однак, якщо проаналізувати таблиці результатів моделей над наборами даних SVHN і GTSRB, можна стверджувати, що у деяких випадках модель з EE-блоком класифікує вхідні дані з вищою точністю. Крім цього, подані таблиці показують, що час передбачення даних у моделі з EE-блоком зазвичай менший, ніж у моделі без його використання, хоча це залежить від кількості надлишкових даних.

Табл. 1: Результати моделі без EE-блоку над наборами даних MNIST, Omniglot

Надлишок, %	Втрата	Точність, %	Час передбачення, сек	Inference time, сек
57	0.023	99.3	21.463	0.000926

Табл. 2: Результати моделі з EE-блоком над наборами даних MNIST, Omniglot

Надлишок, %	Втрата	Точність, %	Точність EE, %	Час, сек	Inf. time, сек
57	0.034	99.4	99.7	17.503	0.000755

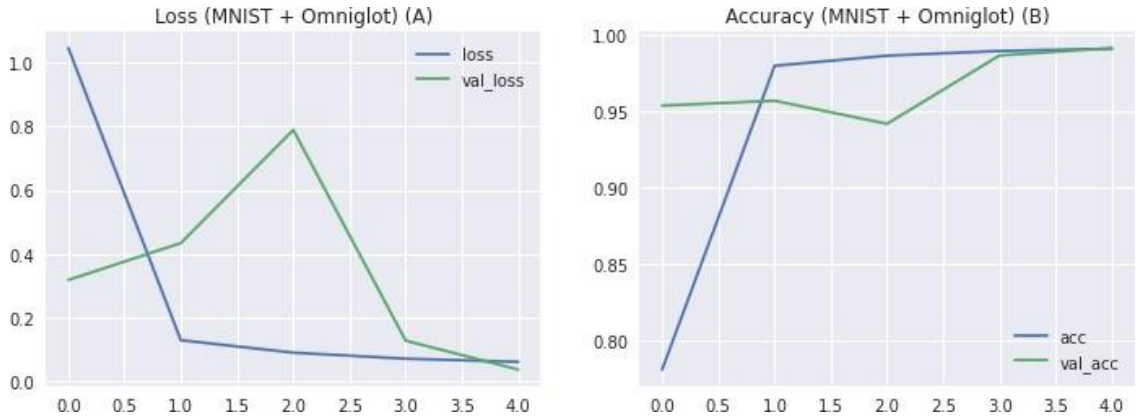

Рис. 2: Графіки функції втрати (A) та точності (B) моделі з EE-блоком на основі MNIST та Omniglot наборів даних

Табл. 3: Результати моделі без EE-блоку над наборами даних SVHN, GTSRB

Надлишок, %	Втрата	Точність, %	Час передбачення, сек	Inference time, сек
28.57	1.281	78.91	21.724	0.001034

Табл. 4: Результати моделі з EE-блоком над наборами даних SVHN, GTSRB

Надлишок, %	Втрата	Точність, %	Точність EE, %	Час, сек	Inf. time, сек
28.57	0.2836	91.29	99.9	18.66	0.000889

Висновки. Як свідчать результати апробації розробеної CNN з впровадженими блоками “раннього виходу”, описаний підхід дає змогу ефективно відсіювати надлишкові зображення на ранніх етапах опрацювання моделі. Зокрема досягнуто зменшення на 20% часу роботи моделі з використанням об’єднаного набору даних MNIST і Omniglot та на 15% стосовно датасетів SVHN і GTSRB.

Список джерел і літератури

[1] S. Scardapane, M. Scarpiniti, E. Baccarelli, A. Uncini *Why should we add early exits to neural networks?* CoRR – 2020. arxiv.org/abs/2004.12814.

[2] Y. Kaya, S. Hong, T. Dumitras *Shallow-Deep Networks: Understanding and Mitigating Network Overthinking* – 2018. doi: 10.48550/ARXIV.1810.07052.

[3] S. Teerapittayanon, B. McDanel, H. T. Kung *BranchyNet: Fast inference via early exiting from deep neural networks* ICPR – 2016.

[4] K. Simonyan, A. Zisserman *Very Deep Convolutional Networks for Large-Scale ImageRecognition* ICLR – 2015.