УДК 004.8

DOI: 10.20535/2617-0965.2020.3.1.198586

Огляд методів реалізації нейронних обчислень на вбудованій системі

Скірко[f] П. О., ORCID 0000-0002-6709-1053
Редько[s] І. В., д.ф.-м.н. проф., ORCID 0000-0002-3121-1412
Кафедра Конструювання Електронно Обчислювальних Приладів
Факультет Електроніки
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Київ, Україна

***Анотація*—У статті виконаний огляд актуальних обмежень у впровадженні нейронних обчислень на вбудованих системах та шляхи їх подолання. Виконано порівняння класичної архітектури інтернету речей з обчисленнями у хмарі та більш сучасної – з частково перенесеною логікою на крайовий пристрій. Розглянуто які технології можуть бути застосовані для запровадження подібної системи та описано методику, що дозволяє досягти поставленої мети – тобто виконати корисні обчислення на крайовому пристрої у реальному часі.**

Ключові слова — вбудовані системи; крайовий пристрій; нейромережа; квантизація нейромережі; згорткова нейромережа; інтернет речей.

I. Вступ

За останнє десятиліття розвиток галузі штучного інтелекту у напрямку нейромереж, особливо згорткових глибинних нейромереж, призвів до розширення сфери їх застосування. Стали можливі наступні застосування цієї технології у користувацьких пристроях:

- розпізнавання обличчя (розблокування за лицем) [1];
- розпізнавання ситуації (падіння, бійка тощо);
- розпізнавання голосу [2] (голосові асистенти) тощо.

Традиційний підхід [3] до реалізації цього функціоналу полягає у надсиланні необроблених даних з пристрою користувача на сервер, обробка їх та повернення результату. Причиною, через яку більшість пристроїв побудована таким чином, є велика обчислювальна складність нейромереж та зручність їх оновлення у випадку, коли вони розташовані на сервері.

Проте, така архітектура стає незастосовною у таких випадках:

- необроблені дані з мікрофону, камери або інших датчиків можуть містити конфіденційну інформацію, яку не бажано пересилати на сервер
- очікувана кількість пристроїв створить навантаження на мережу і датацентри, яке буде занадто дорого обробляти
- затримка у передачі на сервер і назад є неприпустимою

У таких випадках є доцільним виконання нейромережних обчислень на стороні користувача.

II. Архітектура системи інтернету речей

Узагальнена архітектура системи інтернету речей може бути представлена наступним чином (Рис. 1):

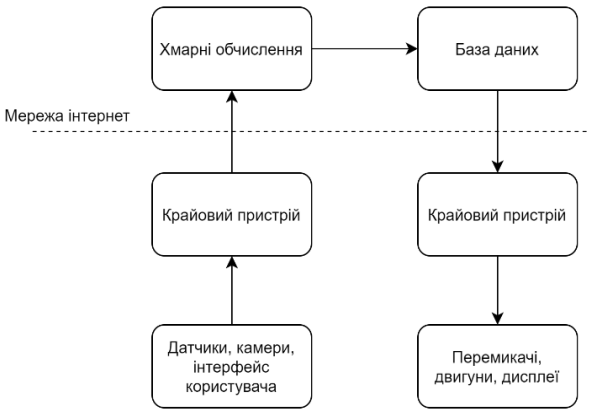

Рис. 1 Узагальнена схема системи інтернету речей

1) Інформація з навколишнього світу накопичується за допомогою датчиків, камер або вводиться користувачем.
2) Отримані дані агрегуються та піддаються первинній обробці крайовим пристроєм. Крайовий пристрій зазвичай характеризується малими габаритами, низькою вартістю (менше $100), низькими енерговитратами (мобільні пристрої з живленням від батареї).

DOI: 10.20535/2617-0965.2020.3.1.198586

3) Оброблені та стиснуті дані передаються на віддалені сервери через мережу інтернет, де відбувається подальша їх обробка з використанням більш ресурсомістких та складних алгоритмів.

4) Результатом хмарних обчислень стає оновлення інформації у базі даних.

5) Нова, більш актуальна інформація спускається на крайовий пристрій, де використовується щоб змінити стан певних індикаторів, двигунів, включити або виключити світло тощо.

Подальшим розвитком наведеної архітектури з урахуванням наведених у вступі обмежень та останніх технологічних розробок є перенесення з віддаленого сервера на крайовий пристрій частини обчислень [4], які раніше не були їм під силу. В наслідок цього, на крайовому пристрої з'являється певний об'єм обробленої інформації, яка може бути передана одразу на інший крайовий пристрій з метою швидкого реагування на зміни навколишнього середовища.

В результаті, архітектура набуває наступного вигляду (Рис. 2):

Рис. 2 Узагальнена схема інтернету речей з перенесеною частиною бізнес логіки на крайовий пристрій

У даній роботі розглядається перенесення на крайовий пристрій нейронних обчислень з метою зменшення навантаження на мережу, зменшення затримки при швидкому реагуванні, зниженні навантаження на центральний сервер та унеможливлення передачі певних конфіденційних даних стороннім особам.

III. Технологічні рішення, що дозволяють запровадити розглянуту архітектуру інтернету речей

На початковому етапі розвитку глибинних згорткових нейромереж, вони досягли успіху шляхом поступового збільшення кількості шарів та вагових коефіцієнтів та ускладнення обчислень на кожному нейроні [5]. Були розроблені LeNet [6] з 60K коефіцієнтів, що досягла точності вище 99% у розпізнаванні рукописних цифр, AlexNet [7] з 60M коефіцієнтів з точністю класифікації кольорових зображень 62.5% та SENet [8] з 25M коефіцієнтів з топ-5 похибкою 2.25% на тій же вибірці кольорових зображень (топ-5 похибка – це відсоток зображень, у яких правильна відповідь не потрапила у топ 5 припущень).

Ця обставина створила перепони на шляху використання їх на крайових пристроях. Шляхи їх подолання можна поділити на такі основні класи:

1) Розробка нових моделей нейромереж здатних розв'язувати подібні задачі використовуючи меншу кількість коефіцієнтів та простіші функції активації але з не надто високим падінням точності.

2) Застосування спеціалізованого апаратного забезпечення, яке здатне виконувати обчислення більш ефективно. Під ефективністю мається на увазі відношення кількості операцій на одиницю часу до споживаної електричної потужності.

3) Оптимізація існуючих моделей нейромереж шляхом зниження точності представлення коефіцієнтів, ігнорування коефіцієнтів близьких до нуля тощо.

До першого класу відносяться розробки Google MobileNet V1 [9] та V2 [10]. Вони базуються на попередніх моделях але мають меншу кількість шарів та нейронів у них що призвело до зменшення загальної кількості параметрів до 0.5-4.2 мільйонів. Застосування цих моделей на практиці для розв'язання задачі детектування та розпізнавання образів на кольоровому зображенні є ефективним: топ-5 похибка становить 10 і 9% відповідно.

Другий клас включає в себе ASIC (Application-specific integrated circuit — інтегральна схема для специфічного застосування), FPGA (Field-Programmable Gate Array — Програмована користувачем вентильна матриця) та GPU (Graphic Processing Unit — Графічний процесор). ASIC розробляються спеціально під конкретну модель нейромережі і мають максимально досяжну на сьогодні швидкодію та енергоефективність. Прикладом є Google TPU (Tensor Processing Unit — Тензорний блок обробки), Intel Movidius VPU (Vision Processing Unit). FPGA та системи на кристалах, що містять у собі FPGA як компонент, дозволяють реалізовувати на апаратному рівні будь-які нейромережі, у межах ресурсів конкретного кристалу. З FPGA стає можливою глибока оптимізація потоків даних, структури пам'яті та запровадження нових експериментальних типів даних. Ці особливості визначають їх роль як співставної альтернативи до ASIC з точки зору швидкодії та енергоефективності [11].

До третього класу належать алгоритми квантизації, та глибокого аналізу структури роботи нейромережі з метою виявлення які її частини вносять найбільший вклад у правильний результат. Квантизацією нейромережі називають переведення коефіцієнтів типу числа з плаваючою комою (зазвичай 32біт) у більш компактний формат (зазвичай ціле 8 бітне

DOI: 10.20535/2617-0965.2020.3.1.198586

число). Подібне перетворення, хоч і призводить до втрати інформації, дозволяє забезпечити точність, близьку до початкової [12]. Оптимізації також можливі з використанням технологій TensorFusion [13] та nVidia TensorRT.

Кінцевий результат у вигляді системи інтернету речей з перенесеними нейронними обрахунками на крайовий пристрій досягається поєднанням усіх вище розглянутих підходів.

Прикладом є реалізація розпізнавання обличчя у роботах [14], [15]. У якості апаратної бази авторами обрано Google Coral Board, яка на сьогодні є одним з найоптимальніших рішень на ринку з точки зору ціни та швидкодії. Coral Board базується на розглянутій вище архітектурі Google TPU. Ця апаратна платформа оптимізована під нейромережі Google MobileNet V1&V2, які в свою чергу розроблялись спеціально для мобільного застосування та характеризуються відносно малим розміром та можливістю їх гнучко підлаштовувати під конкретну задачу з метою зменшення часу обчислень. Після навчання, нейромережа підлягає операції freezing, тобто замороження коефіцієнтів, зберігання їх в один файл та видалення з неї усієї допоміжної інформації, яка потрібна тільки під час навчання. MobileNet є нейромережею, яка за дизайном добре піддається квантизації, а Google Coral працює тільки з 8-бітними числами, тому модель квантизується та компілюється під виконання на даній платі.

Після проходження цього процесу, ця демонстраційна установка дозволяє розпізнавати обличчя з частотою до 25 кадрів на секунду, споживаючи менше 0.5Вт електричної потужності, що значно менше у порівнянні з імплементаціями на CPU та GPU.

Висновки

Згорткові глибинні нейромережі досягли такої стадії розвитку, коли можуть бути застосовані для таких задач як розпізнавання пози, обличчя, ситуації, голосу але класичні найточніші моделі вимагають для виконання потужний сервер, що створює високі вимоги до надійності та пропускної здатності мережі, що звязує сервер на крайовий пристрій.

У той же час, існують такі задачі, в яких затримки мережі неприпустимі, її надійність не може гарантуватись, або характер даних, що необхідно обробити не дозволяє виводити їх за межі крайового пристрою. Для цих випадків були розроблені технологічні рішення, що дозволяють виконувати нейронні обчислення на крайовому пристрої у реальному часі, тобто розпізнавання візуальних образів 25 кадрів на секунду. Ці технологічні рішення включають в себе нові нейромережі зі зменшеною кількістю коефіцієнтів та спрощеною архітектурою, спеціалізовані апаратні рішення для обрахунку подібних нейромереж та застосування квантизації до вже навченої нейромережі. Поєднання всіх прийомів дозволяє досягти поставленої мети.

Перелік посилань

[1] S. Balaban, “Deep learning and face recognition: the state of the art,” 2015, p. 94570B, **DOI:** 10.1117/12.2181526.

[2] M.A.Anusuya and S.K.Katti, “Speech Recognition by Machine: A Review,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 6, no. 3, pp. 181–205, 2009, **URL:** https://arxiv.org/ftp/arxiv/papers/1001/1001.2267.pdf.

[3] A. El Hakim, “Internet of Things (IoT) System Architecture and Technologies, White Paper,” 2018,
DOI: 10.13140/RG.2.2.17046.19521.

[4] M. G. S. Murshed, D. Murphy, Christopher Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, “Machine Learning at the Network Edge: A Survey,” p. 33, 2020,
URL: https://arxiv.org/pdf/1908.00080.pdf.

[5] M. P. Véstias, “A Survey of Convolutional Neural Networks on Edge with Reconfigurable Computing,” *Algorithms*, vol. 12, no. 8, p. 154, Jul. 2019, **DOI:** 10.3390/a12080154.

[6] L. Jackel, J. Denker, A. Harris, and P. Y. Simard, “Learning Algorithms For Classification: A Comparison On Handwritten Digit Recognition,” 2000,
URL:
https://www.researchgate.net/publication/2376934_Learning_Algorithms_For_Classification_A_Comparison_On_Handwritten_Digit_Recognition.

[7] S. Hochreiter, “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions,” *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 06, no. 02, pp. 107–116, Apr. 1998, **DOI:** 10.1142/S0218488598000094.

[8] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation Networks,” 2019,
URL: https://arxiv.org/pdf/1709.01507.pdf

[9] M. Howard, Andrew G. Zhu, B. Chen, and D. Kalenichenko, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017,
URL: https://arxiv.org/pdf/1704.04861.pdf

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520, **DOI:** 10.1109/CVPR.2018.00474.

[11] S. I. Venieris, A. Kouris, and C.-S. Bouganis, “Toolflows for Mapping Convolutional Neural Networks on FPGAs,” *ACM Comput. Surv.*, vol. 51, no. 3, pp. 1–39, Jul. 2018, **DOI:** 10.1145/3186332.

[12] P. Gysel and S. Motamedi, Mohammad Ghiasi, “Hardware-oriented Approximation of Convolutional Neural Networks,” 2016, **URL: https://arxiv.org/abs/1604.03168**

[13] A. Zadeh, M. Chen, and S. Poria, “Tensor Fusion Network for Multimodal Sentiment Analysis,” 2017,
URL: https://arxiv.org/pdf/1707.07250.pdf

[14] P. F. T. Madio, “A FaceNet-Style Approach to Facial Recognition on the Google Coral Development board,” 2019. [Online]. Available: https://towardsdatascience.com/a-facenet-style-approach-to-facial-recognition-dc0944efe8d1

[15] C. Knauf and P. Strobel, “Realtime face detection and filtering with the Coral USB accelerator,” 2019. [Online]. Available: https://blog.codecentric.de/en/2019/11/realtime-face-detection-and-filtering-with-the-coral-usb-accelerator/.

UDC 004.8

DOI: 10.20535/2617-0965.2020.3.1.198586

A Survey of Methods of Implementation Deep Convolutional Neural Network on Embedded System

P. O. Skirko, ORCID 0000-0002-6709-1053
I. V. Redko, Dr.Sc.(Phys.-Math.) Prof., ORCID 0000-0002-3121-1412
Faculty of electronics
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
Kyiv, Ukraine

***Abstract*—Convolutional neural networks have reached a stage of development where they can be applied to tasks such as posture, face, voice and situation recognition with high accuracy, but inference of classic high precision models require a powerful server that creates high requirements for the reliability and bandwidth of the network that connects the server to the Edge device.**

Traditional architecture of Internet of Things can be represented as follows: information from the outside world is accumulated through sensors, cameras or input by the user; the data obtained is aggregated and subjected to initial processing by the edge device (the edge unit is typically characterized by small dimensions, low cost, and low power consumption); the processed and compressed data is transmitted to the remote servers via the Internet, where it is further processed using more resource-intensive and complex algorithms; the cloud computes results and updates information in the database; the new, more up-to-date information goes down to the edge device, where it is used to change the status of certain indicators, engines, turn on or off the light, etc.

At the same time, there are tasks in which delays of the network are unacceptable, its reliability cannot be guaranteed, raw data from the microphone, camera, or other sensors may contain sensitive information that is not desirable to send to the server, the expected number of devices will create such load of network and datacenter that will be too expensive to process.

For these cases, technological solutions have been developed that allow performing neural computations on the edge device in real time, that is, recognizing visual images of 25 frames per second. These techniques includes: development of new neural network models capable of solving similar problems using fewer coefficients and simpler activation functions, but with not too high a drop in accuracy; applying specialized hardware, capable of performing calculations with higher efficiency (efficiency refers to the ratio of the number of operations per unit of time to the power consumed); optimization of existing neural network models by reducing the accuracy of coefficients representation, ignoring the coefficients close to zero, etc. The first class includes Google MobileNet V1 and V2 development. They are based on previous models but have fewer layers and neurons in them, which reduced the total number of parameters to 0.5-4.2 million. The second class includes ASIC (Application-specific integrated circuit), FPGA (Field-Programmable Gate Array) and GPU (Graphic Processing Unit). ASICs are specifically designed for a specific neural network model and have the highest performance and energy efficiency available today. The third class includes quantization algorithms and a thorough analysis of the structure of the neural network to identify which parts of it contribute most to the correct result. Such conversion, although loss of information, allows for accuracy close to the original.

The combination of all the techniques allows you to achieve the task.

Keywords — edge device; convolutional network; inference; network quantization; internet of things.