УДК 004.77.032.26(043.2)

ОПТИМІЗАЦІЯ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ ДЛЯ РЕАЛЬНОГО ЧАСУ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ПРИСКОРЕННЯ НА GPU

Денис Рибак

Державний університет «Київський авіаційний інститут» , Київ

Науковий керівник – Тетяна Холявкіна, к.т.н., доцент.

Ключові слова: оптимізація нейромереж, GPU-прискорення, глибоке навчання, TensorRT, CUDA.

Вступ. Використання нейромережевих моделей у задачах реального часу вимагає оптимізації їх продуктивності для забезпечення швидкого інференсу. Один із підходів до підвищення швидкодії – використання технологій прискорення на графічних процесорах. Завдяки ефективному паралельному обчисленню сучасні GPU дозволяють значно зменшити час виконання глибоких нейромережевих моделей без втрати точності.

Матеріали та методи

У дослідженні використано фреймворки TensorFlow і PyTorch для навчання моделей, а також технології прискорення NVIDIA CUDA та TensorRT для оптимізації інференсу. Було протестовано кілька популярних архітектур, зокрема ResNet-50, MobileNetV2 та EfficientNet, із застосуванням методів квантизації, усічення та компіляції для GPU. Експерименти проводилися на апаратному забезпеченні з графічним процесором NVIDIA RTX 3090.

Результати. Оптимізація нейромереж із використанням TensorRT дозволила скоротити час інференсу в середньому на 45-70% залежно від архітектури. Для моделі ResNet-50 швидкість зросла на 48%, для MobileNetV2 – на 67%, а для EfficientNet – на 72%. Аналіз показав, що найкращі результати досягаються при зменшенні точності обчислень до FP16 та застосуванні специфічних профілів оптимізації TensorRT [1, 2, 3].

При аналізі продуктивності було помітно, що моделі з меншою кількістю параметрів, такі як MobileNetV2, демонструють значне прискорення завдяки квантизації, тоді як важчі моделі, такі як ResNet-50, отримують найбільшу вигоду від оптимізації пам’яті та паралельного обчислення. Оптимізація з використанням TensorRT дозволила зменшити затримку інференсу з 22 мс до 7 мс для MobileNetV2, що дозволяє використовувати її в реальному часі без помітної затримки для користувача. EfficientNet, завдяки своєму оптимізованому дизайну, після оптимізації TensorRT забезпечила найбільшу продуктивність у співвідношенні точності та

швидкості, зменшивши час інференсу на 72% при збереженні високої точності класифікації [4, 5].

Додатково проведене тестування показало, що використання GPU-прискорення на мобільних пристроях зменшило енергоспоживання на 25-30%, що є критичним фактором для інтеграції нейромереж у вбудовані системи та пристрої Інтернету речей. Аналіз споживання ресурсів показав, що використання FP16 і усічення менш значущих шарів зменшують не лише час виконання, але й використання відеопам'яті в середньому на 40%, що дозволяє запускати потужніші моделі на менш продуктивному обладнанні [6].

Висновок. Оптимізація нейромережевих моделей із використанням GPU-прискорення значно покращує їхню продуктивність у задачах реального часу. Найефективнішими підходами виявилися квантизація, усічення та компіляція з TensorRT. Отримані результати демонструють, що оптимізація дає змогу запускати глибокі моделі на пристроях із обмеженими ресурсами без значної втрати точності. Подальші дослідження можуть бути зосереджені на автоматизованій оптимізації моделей, інтеграції технологій розподіленого обчислення та зменшенні енергоспоживання при виконанні нейромереж.

Список використаних джерел:

1. Optimizing Deep Learning Models with TensorRT. URL: https://astconsulting.in/artificial-intelligence/ml-machine-learning/optimizing-deep-learning-models-with-tensorrt (date of access: 14.03.2025).

2. Accelerating Inference for Deep Learning Models. URL: https://github.com/triton-inference-server/tutorials/blob/main/Conceptual_Guide/Part_4-inference_acceleration/README.md (date of access: 14.03.2025).

3. Power of GPU Acceleration in Deep Learning: Elevating Model Training Performance. URL: https://www.linkedin.com/pulse/power-gpu-acceleration-deep-learning-elevating-model-training-sachin-1ivtc/ (date of access: 14.03.2025).

4. ResNet-50. URL: https://infohub.delltechnologies.com/en-us/l/power-ai-with-dell-and-nvidia/resnet-50/ (date of access: 14.03.2025).

5. How we made EfficientNet more efficient. URL: https://towardsdatascience.com/how-we-made-efficientnet-more-efficient-61e1bf3f84b3/ (date of access: 14.03.2025).

6. Real-Time Neural Network Optimization. URL: https://www.restack.io/p/model-optimization-answer-real-time-neural-network-optimization-cat-ai (date of access: 14.03.2025).