Целочисленное квантование для вывода при глубоком обучении

Владимирова Э.В., студент кафедры информатики СПбГУ, техник-программист
ООО «Системы компьютерного зрения», Санкт-Петербург
st069281@student.spbu.ru

Аннотация

В статье рассмотрена новая реализация метода Quant-Noise для нейросетей ResNet-20 и EfficientNet-B3 ввиду отсутствия оригинальной и проведено сравнение эффективности обучения нейросети для подавления шума CBDNet с использованием методов LSQ, LSQ+ и QSin.

Введение

Практическое использование нейросетей в особенности на мобильных устройствах приводит к необходимости сжатия моделей по памяти и вычислительной сложности. Одним из зарекомендовавших себя методов является квантование весов сети и вычислений.

В рамках проекта «Discrete Optimization for Accurate Low-bit Quantization» лаборатории им. П. Л. Чебышёва по разработке нового метода квантования нейросетей для обработки изображений в форматы int8, int4 и int2 одной из промежуточных задач была подготовка данных для сравнения с результатами существующих методов на общем наборе архитектур нейросетей и данных.

В статье представленны новые результаты обучения нейросети CBDNet [3] на наборе изображений SIDD с использованием существующих методов квантования LSQ [7], LSQ+ [8] и QSin [10] и их сравнение. Также представлены результаты собственной реализации метода Quant-Noise [9] на нейросетях ResNet-20 [1] и EfficientNet-B3 [6] с наборами изображений CIFAR-10 и ImageNet в конфигурации w4a4 (квантование весов и активаций нейросети до типа int4).

Обучение с учётом квантования

Операция квантования вещественного числа представима функцией [5]:

$$x_q = \text{quantize}(x, b, s, z) = \text{clip}(\text{round}(s \cdot x + z), -2^{b-1}, 2^{b-1} - 1),$$

где x — исходное вещественное значение, b — количество бит в целочисленном типе, $z = -\text{round}(\beta \cdot s) - 2^{b-1}$ — нулевая точка квантования, масштаб (шаг) квантования —

$$s = \frac{2^b - 1}{\alpha - \beta},$$

функция обрезания —

$$\text{clip}(x, l, u) = \begin{cases} l, & x < l \\ x, & x \in [l; u]. \\ u, & x > u \end{cases}$$

Для весов рассматривается симметричное квантование, $z = 0$. Обучение квантованной сети методом градиентного спуска без дополнительных модификаций вызывает затрудения из-за кусочной постоянности функции квантования. Эта проблема решается либо гладкой регуляризацией функции квантования (например, QSin) [10], либо прямолинейным оцениванием (Straight-through Estimator, STE) [2, 7, 8], игнорирующим квантование на обратном проходе по сети и доопределяющим производную:

$$\widetilde{x} = \text{dequantize}(\text{quantize}(x, b, s), b, s).$$

Эти подходы обучения с учётом квантования (Quantization Aware Training, QAT), позволяют обучать квантованную сеть обычными методами.

Метрики

Для задач классификации (ResNet-20 и EfficientNet-B3) используется метрика точность (accuracy). Для задачи подавления шума используются метрики сходства изображений: пиковое отношение сигнала к шуму (PSNR) и индекс структурного сходства (SSIM).

Метод рандомизированного добавления шума квантования

Описание задачи

Метод рандомизированного добавления шума квантования (Quantization Noise, Quant-Noise) [9] является модификацией QAT и заключается в применении симуляции эффекта квантования к рандомизированной выборке определённого процента весов из всего набора. Доля квантуемых весов — гипер-

параметр метода. Выборка генерируется индивидуально для каждого прямого прохода в ходе обучения нейросети. Активации в дополненном процессе обучения не затрагиваются. Авторами приводятся численные доказательства большей эффективности описанного подхода в сравнении с оригинальным QAT для нейросетей из двух различных областей глубокого машинного обучения: обработки естественного языка и классификации изображений, из которых в рамках проекта актуальна вторая. Вместе с тем имеют место следующие проблемы:

- официальная реализация алгоритма не завершена для нейросетей, решающих задачи компьютерного зрения. Репозиторий [4], указанный в статье как источник кода, содержит не адаптированную к анализу цветных изображений и не обеспечивающую сходимость обучения квантованной нейросети версию алгоритма;

- подходящие техническому заданию проекта результаты представлены только для нейросети EfficientNet-B3 и датасета ImageNet, что усложняет сравнение эффективности Quant-Noise с остальными методами.

Таким образом, работа с алгоритмом Quant-Noise потребовала независимой реализации.

Реализация

В официальном репозитории [4] симуляция квантования организована заменой свёрточных и линейных слоёв нейросети на аналогичные исходным слои с дополнительными параметрами. Применением описанных классовых преобразований к ResNet-20 и восстановлением работоспособности алгоритма была экспериментально выявлена непригодность оригинальной реализации: продемонстрированные при обучениях с набором конфигураций результаты значительно уступают заявленным в статье и полученным для реализации алгоритма на основе метода QAT.

В качестве базы для собственной реализации алгоритма Quant-Noise использована реализация метода QAT, созданная в рамках проекта с использованием PyTorch. Выборка весов, имеющих вид многомерного тензора, производится генерацией случайной бинарной маски по распределению Бернулли. Ввиду сложности архитектуры и длительности обучения целевых нейросетей для разработки были взяты следующие комбинации наборов данных и нейросетей на каждом этапе:

1. **запуск алгоритма**: датасет — MNIST, нейросеть — встроенный шестислойный классификатор;

2. **отладка и тестирование алгоритма**: датасет — CIFAR-10, нейросеть — ResNet-20;

3. **сравнение результатов со статьёй**: датасет — ImageNet, нейросеть — EfficientNet-B3.

Созданная в ходе проекта стабильная реализация Quant-Noise подтвердила ожидаемое улучшение точности по сравнению с QAT на нейросети ResNet-20 и датасете CIFAR-10 (Таблица 1). Тем не менее, обучение квантованной с Quant-Noise нейросети длилось в 1.5 раза дольше обучения квантованной с QAT и в 9 раз дольше обучения исходной вещественнозначной. Внесением оптимизаций в код и повышением размера батча разрыв между временем работы реализаций двух методов был сокращён.

Конфигурация	**QAT**	**Quant-Noise (prob=0.5)**	**Quant-Noise (prob=0.125)**
w8a8	91.76	**91.88**	91.87
w4a4	89.27	**90.54**	89.8

Таблица 1: Точность (accuracy) ResNet-20 на CIFAR-10 при обучении с методами QAT и Quant-Noise w4a4

На целевых нейросети EfficientNet-B3 и датасете ImageNet было выявлено существенное замедление обучения внедрённым Quant-Noise, в итоге частично нейтрализованное изменением способа генерации и кэшированием бинарных масок. При проверке гипотез о причинах и возможных решениях проблемы реализация была разбита на две в зависимости от наличия опции хранения масок. Профилирование реализаций QAT и Quant-Noise показало сопоставимость требуемых вычислительных и временных ресурсов для обучения нейросети EfficientNet-B3 с двумя методами (Таблица 2).

	floating point	**QAT**	**Quant-Noise (prob=0.5)**
Время на 1 эпоху (ч:мин)	1:55	6:05	6:28
Утилизация GPU (%)	91—93	57—67	57—68
Память GPU (%)	63.1	76.4	93.6

Таблица 2: Время 1 эпохи и загрузка GPU при обучении EfficientNet-B3 на ImageNet методами QAT и Quant-Noise w4a4

Также был проведён подбор наиболее оптимальных по времени и точности работы квантованной нейросети гиперпараметров обучения, с которыми

были достигнуты показатели точности в Таблице 3.

Модель	**Accuracy**
Данные авторов floating-point модели [6]	82.008
Валидация floating-point модели	82.032
QAT	65.324
Quant-Noise	77.154

Таблица 3: Точность (accuracy) EfficientNet-B3 на ImageNet при обучении методами QAT и Quant-Noise w4a4

Оценка показателей обучения целевых методов

Оценка эффективности разрабатываемого в рамках проекта алгоритма целочисленнго квантования строится на сравнении следующих показателей:

1. время обучения (время прохождения 1 эпохи);
2. скорость обучения (значения целевых метрик после 80 эпох обучения);
3. требования обучения к памяти графического процессора (максимальный доступный размер батча);

для нейросетей, квантованных с помощью лучшего по большинству указанных характеристик из выбранных ориентировочных методов: LSQ [7], LSQ+ [8] и QSin [10] — для конфигурации w4a4. В рамках поставленной задачи в качестве тестовой комбинации взяты нейросеть CBDNet [3] и 2 поднабора набора данных SIDD: Small и Medium. Эксперименты проведены на графическом профессоре NVIDIA Tesla A10.

Показатели в Таблицах 4, 5 получены при размере батча — 30 изображений. Максимальный размер батча для рассматриваемых методов — 120 изображений, из чего допустимо предположить примерное равенство требуемых для обучения ресурсов памяти GPU.

Датасет	**LSQ**	**LSQ+**	**QSin**
SIDD Small	**2:57**	3:31	4:08
SIDD Medium	**5:54**	7:02	8:16

Таблица 4: Время 1 эпохи обучения CBDNet методами LSQ, LSQ+, QSin w4a4 (мин:с)

Датасет	LSQ		LSQ+		QSin	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
SIDD Small	35.135	0.815	**35.221**	**0.816**	25.024	0.371
SIDD Medium	**35.531**	**0.828**	35.488	0.828	25.099	0.372

Таблица 5: Метрики CBDNet при обучении методами LSQ, LSQ+, QSin w4a4

Поскольку результаты LSQ+ и LSQ близки и LSQ лучше на большом наборе данных, то для визуального сравнения был выбран LSQ. На рисунке 1 видно, что, несмотря на относительно высокие значения метрик, квантование приводит к заметным визуальным артефактам.

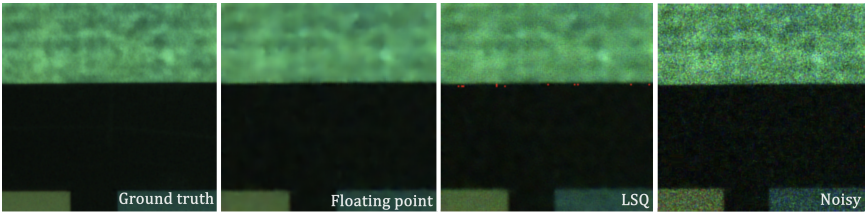

Рис. 1: Визуальное сравнение CBDNet при обучении методом LSQ w4a4

Заключение

В работе приведены итоги реализации метода квантования Quant-Noise для нейросети ResNet-20 на датасете CIFAR-10 и EfficientNet-B3 на датасете ImageNet, предложены способы повышения эффективности по использованию ресурсов памяти GPU и времени реализации. Реализация Quant-Noise в рамках проекта для квантования нейросети в формат int4 улучшает accuracy на 1.27% для ResNet-20 на CIFAR-10 и 11.83% EfficientNet-B3 на ImageNet по сравнению с QAT, чем подтверждает результаты оригинальной статьи.

Также проведено сравнение времени, скорости и требований к памяти для обучения нейросети CBDNet на поднаборах Small и Medium набора данных SIDD для методов квантования LSQ, LSQ+ и QSin. В качестве главного конкурирующего метода для дальнейшего сравнения берётся LSQ.

Список литературы

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. — arXiv, 2015

[2] Benoit Jacob, Skirmantas Kligys, Bo Chen. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. — arXiv, 2017

[3] Shi Guo, Zifei Yan, Kai Zhang et al. Toward Convolutional Blind Denoising of Real Photographs. — arXiv, 2019

[4] Fairseq. Training with Quantization Noise for Extreme Model Compression, 2020. — URL: `https://github.com/facebookresearch/fairseq/tree/main/examples/quant_noise`. (accessed: 2023-05-29)

[5] Hao Wu, Patrick Judd, Xiaojie Zhang, et al. Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation. — arXiv, 2020

[6] Mingxing Tan, Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. — arXiv, 2020

[7] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani et al. Learned Step Size Quantization. — arXiv, 2020

[8] Yash Bhalgat, Jinwon Lee, Markus Nagel et al. LSQ+: Improving low-bit quantization through learnable offsets and better initialization. — arXiv, 2020

[9] Angela Fan, Pierre Stock, Benjamin Graham et al. Training with Quantization Noise for Extreme Model Compression. — arXiv, 2021

[10] Kirill Solodskikh, Vladimir Chikin, Ruslan Aydarkhanov et al. Towards Accurate Network Quantization with Equivalent Smooth Regularizer. — European Conference on Computer Vision (ECCV), 2022