УДК 004.032.26

ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ НИЗКОРАЗРЯДНЫХ ПРЕДСТАВЛЕНИЙ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ ДЛЯ ЭФФЕКТИВНЫХ ВЫЧИСЛЕНИЙ В НЕЙРОННЫХ СЕТЯХ

© **А. Ю. Кондратьев**[2], **А. И. Гончаренко**[1, 2]

[1] *Новосибирский государственный университет,*
630090, г. Новосибирск, ул. Пирогова, 2
[2] *ООО «Экспасофт»,*
630090, г. Новосибирск, ул. Николаева, 11
E-mail: a.kondratev@expasoft.ru
a.goncharenko@expasoft.ru

Изучена возможность работы нейронных сетей на типах данных низкой разрядности с плавающей запятой (minifloat). Выполнены вычисления с использованием аккумулятора float16 для промежуточных вычислений. Осуществлена проверка качества работы на свёрточных нейронных сетях GoogleNet, ResNet-50, MobileNet-v2, а также на рекуррентной сети DeepSpeech-v01. Эксперименты показали, что качество работы указанных нейронных сетей с 11-разрядными minifloat не уступает качеству работы сетей со стандартным типом float32 без проведения дополнительного обучения. Результаты свидетельствуют о том, что числа с плавающей запятой низкой разрядности можно использовать для создания эффективных вычислителей, специализирующихся на работе нейронных сетей.

Ключевые слова: нейронные сети, глубокое обучение, типы данных, minifloat, специализированные вычислители.

DOI: 10.15372/AUT20200110

Введение. За последнее десятилетие глубокое обучение стало доминирующим методом машинного обучения, показывающим значительные успехи в широком спектре задач, включая обработку изображений, машинный перевод, распознавание речи и многое другое. В каждой из этих областей глубокие нейронные сети достигают удовлетворительной точности благодаря использованию очень больших и глубоких моделей, требующих до 100 эпох вычислений во время обучения и миллиарды мультипликативно-аддитивных операций для выполнения однократного запуска.

Обучение и запуск нейронных сетей, как правило, проводятся на процессорах и графических ускорителях стандартной архитектуры, включающей в себя 32-разрядный или 16-разрядный формат данных с плавающей запятой, определённый в стандарте IEEE-754. Применение более сжатых форматов может дать существенные улучшения: более эффективное использование площади процессора и более низкое энергопотребление, поскольку энергоэффективность оборудования зависит квадратично от количества бит основного типа данных. Однако возможность применения более сжатых типов, чем 16-разрядный, требует тщательного исследования изменения показателей метрики качества нейронных сетей.

Известно, что нейронные сети способны без потери точности работать на современных видеокартах с использованием типа данных float16. Такой тип чисел с плавающей запятой половинной точности зафиксирован в стандарте IEEE-754 и имеет широкую аппаратную и программную поддержку. Кроме этого, в тензорных процессорах компании Google активно реализуется нестандартный тип данных bfloat16, который характеризуется более широким диапазоном значений экспоненты числа.

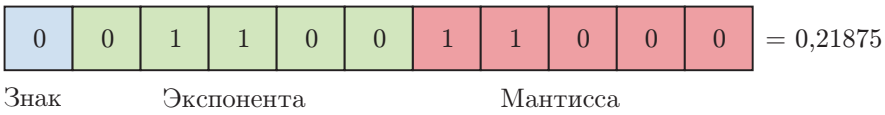

Рис. 1. Пример представления minifloat $\langle 5, 5\rangle$, имеющего 5 бит, отведённых под хранение показателя экспоненты, и 5 бит — под мантиссу числа

В [1] показано, что нейронные сети GoogleNet и DeepSpeech-v1 сохраняют качество своей работы при 12-разрядном minifloat, в котором 5 бит отведено под представление экспоненты числа и 7 бит — под представление мантиссы.

В [2] проведены эксперименты со свёрточными нейронными сетями AlexNet, VGG-16, GoogleNet, которые показали, что перечисленные нейронные сети способны работать практически без потери точности при использовании 10-разрядных fixed-point активаций и 8-разрядных minifloat.

Аналитики данных компании IBM провели работу по исследованию возможности обучения весов нейронной сети [3]. Удалось успешно тренировать нейронную сеть, используя 8-разрядное представление чисел с плавающей запятой для весов нейронных сетей AlexNet, ResNet-18, ResNet-50. Обученные для minifloat сети работают без потери точности при вычислениях с аккумулятором float16 промежуточных операций.

Представление minifloat. В предлагаемой работе исследуется возможность применения нестандартного типа данных minifloat для всех видов вычислений в работе нейронной сети. Формат чисел minifloat являет собой множество типов данных, аналогичных стандартным типам float32. Подразумевается, что их размер не превосходит 16 бит и соответственно выходит за рамки общепринятого стандарта IEEE-754. Представление числа в памяти (рис. 1) включает в себя последовательно один бит, определяющий знак числа, и определённое число бит, содержащих смещённый показатель экспоненты числа, а оставшиеся биты формируют значение мантиссы числа. Обозначение minifloat $\langle e, m\rangle$ показывает, что под представление экспоненты выделено e бит, а под мантиссу — m бит. В числе одинарной точности отводятся 8 бит под представление экспоненты и 23 бита под представление мантиссы. В стандартном числе половинной точности используется 5 и 10 бит для экспоненты и мантиссы соответственно. Мы будем исследовать minifloat, в которых количество бит под представление экспоненты не превосходит 5, а под представление мантиссы — 10 бит.

Соответствие стандарту при этом заключается в сохранении правил получения десятичного числа из бинарного представления (1), а также в наличии специальных состояний, обозначающих ± 0, $\pm\infty$ и $\pm$NaN (not a number):

$$F = (-1)^s 2^{(E-2^{e-1}+1)}(1 + M/2^m). \tag{1}$$

Стандарт также предусматривает два состояния числа с плавающей запятой: нормализованное и денормализованное. Особенность денормализованных чисел — это иная логика интерпретации мантиссы. При значении показателя экспоненты, равном нулю, к мантиссе не прибавляется неявная единица:

$$F = (-1)^s 2^{(E-2^{e-1}+2)} M/2^m. \tag{2}$$

Это позволяет получить дополнительную точность представления чисел в окрестности нуля, однако ведёт к усложнению аппаратной реализации и уменьшению быстродействия. В данной работе рассматриваются только нормализованные числа. Кроме того, зафиксирован алгоритм округления чисел к ближайшему при выполнении всех операций.

Алгоритм конвертации из числа одинарной точности в minifloat $\langle e, m\rangle$ заключается в отбрасывании избыточных бит мантиссы с округлением к ближайшему. При этом выполняется проверка переполнения порядка числа. Обратный алгоритм получения числа одинарной точности из minifloat тривиален, поскольку число одинарной точности обладает большей ёмкостью показателей экспоненты и мантиссы. Соответственно достаточно извлечь из minifloat данные показатели и сохранить в показателях числа float32, добавив противоположные сдвиги для показателя экспоненты.

Вычисления с аккумулятором. Функции свёртки и матричного перемножения, широко использующиеся в нейронных сетях, состоят из скалярных произведений, которые накапливают большое количество поэлементных произведений чисел с плавающей запятой. Поскольку сложение чисел с плавающей запятой включает в себя сдвиг вправо меньшего из двух операндов (на разницу в показателях степени), то может оказаться, что это меньшее число станет нулевым после сложения из-за недостаточного количества бит мантиссы. В контексте глубоких нейронных сетей показано [1], что данная проблема является особенно серьёзной, когда количество используемых бит аккумулятора резко снижается.

Аккумулятор большей разрядности позволяет бороться с этой проблемой и, как следствие, повысить точность вычислений. В представленной работе в качестве типа данных аккумулятора используется float16.

Эксперименты. Для проведения экспериментов по измерению качества работы нейронных сетей была выполнена модификация открытой библиотеки машинного обучения Tensorflow.

В исходный код C++ добавлен пользовательский тип данных minifloat $\langle e, m\rangle$. Реализованы эффективные функции преобразования из числа одинарной точности в minifloat и обратно. Переопределены все базовые операции: сложение, умножение, вычитание, деление, а также операции сравнения. Тип данных был зарегистрирован во внутренних программных системах Tensorflow. Вследствие этого стало возможным использовать тип данных minifloat для операций двумерной свёртки, матричного умножения, поэлементного сложения и многих других.

Реализовано высокоуровневое API (Application Programming Interface) для языка Python как часть Tensorflow, что позволяет для большинства аналитиков данных на привычном языке и фреймворке внедрять в вычислительные графы нейронных сетей тип minifloat как основной и запускать работу нейронных сетей.

Для проведения численных экспериментов выбраны свёрточные нейронные сети для классификации изображений: GoogleNet [4], ResNet-50 [5], MobileNet-v2 [6], а также рекуррентная нейронная сеть для распознавания речи DeepSpeech-v1 [7]. Обученные на Tensorflow модели нейронных сетей взяты из [8, 10].

Классификация изображений выполнена на тестовом наборе изображений ILSVRC-2012 [9]. Тестирование качества распознавания речи проведено на размеченных аудиозаписях LibriSpeech [11].

Так как обученные модели нейронных сетей предназначены для работы с типом данных float32, то для использования minifloat с аккумулятором в модель были добавлены две операции преобразования типа входных тензоров — весов и активаций (рис. 2). Вначале исходные тензоры приводятся к типу данных исследуемого minifloat $\langle e, m\rangle$, затем выполняется их преобразование в float16. Выходной тензор текущего свёрточного слоя является входным тензором активаций следующего слоя, в котором также будет производиться преобразование типа. Таким образом, основным типом данных для активаций и весов нейронной сети является minifloat $\langle e, m\rangle$.

Объектами исследования являются подготовленные типы minifloat с общей разрядностью от 6 до 16 бит включительно. При этом количество бит под представление экспоненты

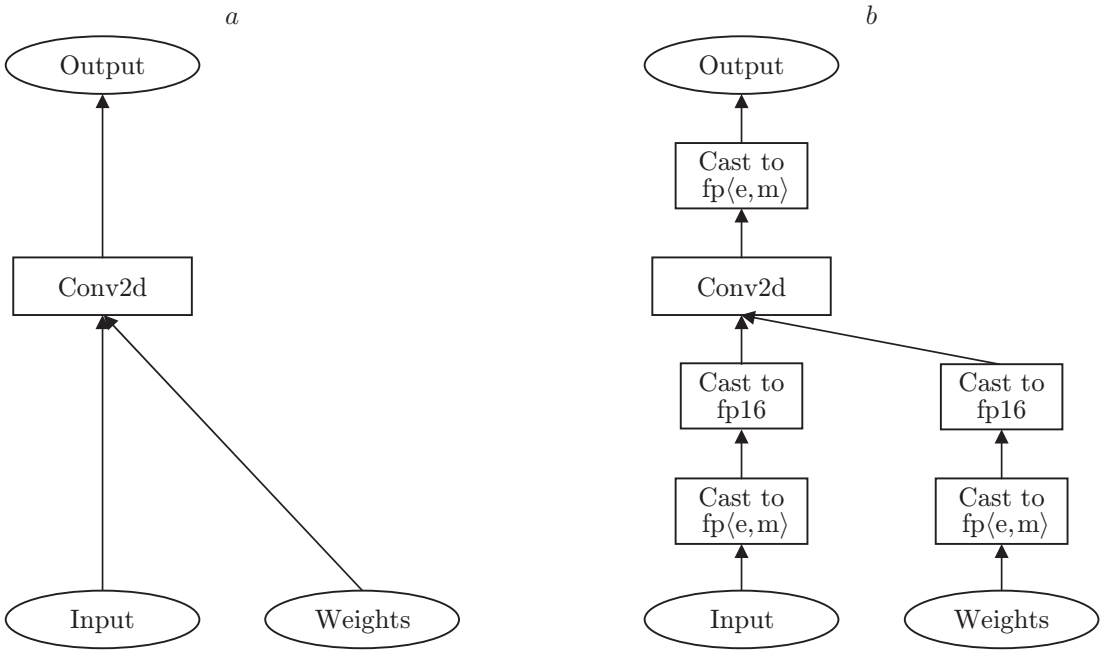

Рис. 2. Процедура конвертации свёрточного слоя нейронной сети: пример исходного (a) и модифицированного (b) свёрточных слоёв. Две последовательные операции преобразования типа позволяют получить данные в низкоразрядном представлении minifloat и проводить дальнейшие вычисления свёртки, сложения и активации с типом данных float16

Таблица 1

Точность работы GoogleNet (исходная точность равна 0,7099)

Разрядность экспоненты	Разрядность мантиссы								
	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$	$m = 7$	$m = 8$	$m = 9$	$m = 10$
$e = 3$	0,0014	0,0014	0,0014	0,0014	0,0014	0,0014	0,0014	0,0014	0,0014
$e = 4$	0,5085	0,6745	0,6943	0,6971	0,7009	0,7002	0,7008	0,7008	0,7009
$e = 5$	0,5255	0,6779	0,7008	0,7066	0,7070	0,7094	0,7099	0,7093	0,7097

меняется в диапазоне от 3 до 5, а размер мантиссы варьируется от 2 до 9 бит. Сравнение осуществлено с исходным качеством работы нейронных сетей на типе данных float32.

Для нейронных сетей GoogleNet, ResNet-50, MobileNet-v2 основной метрикой качества является точность, определяемая как отношение правильно классифицированных изображений к общему числу рассмотренных случаев.

По результатам, приведённым в табл. 1–3, видно, что наименьший minifloat, при котором сохраняется приемлемое качество данных нейронных сетей, — это 11-разрядный minifloat $\langle 5, 5\rangle$. Это значит, что количество бит представления мантиссы можно с минимальными потерями точности сократить до пяти относительно представления float16. При разрядности мантиссы, равной четырём, наблюдается заметный спад в 2 % точности на MobileNet-v2, что для ряда задач может быть уже неприемлемой потерей качества. В то же время необходимо отметить, что нейронные сети GoogleNet и ResNet-50 являются более устойчивыми к низкому количеству бит мантиссы и качественно работают с minifloat $\langle 5, 4\rangle$.

Что касается разрядности экспоненты, то с количеством бит менее 5 хорошо работает

Таблица 2

Точность работы ResNet-50 (исходная точность равна 0,7404)

Разрядность экспоненты	Разрядность мантиссы								
	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$
$e=3$	0,0011	0,0011	0,0011	0,0011	0,0011	0,0011	0,0011	0,0011	0,0011
$e=4$	0,0008	0,0010	0,0011	0,0011	0,0011	0,0011	0,0011	0,0011	0,0011
$e=5$	0,6610	0,7121	0,7377	0,7390	0,7386	0,7398	0,7408	0,7398	0,7408

Таблица 3

Точность работы MobileNet-v2 (исходная точность равна 0,7283)

Разрядность экспоненты	Разрядность мантиссы								
	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$
$e=3$	0,0010	0,0010	0,0010	0,0010	0,0010	0,0010	0,0010	0,0010	0,0010
$e=4$	0,1263	0,5669	0,6197	0,6643	0,6918	0,6939	0,6948	0,6943	0,6945
$e=5$	0,1785	0,5964	0,6993	0,7182	0,7263	0,7272	0,7274	0,7288	0,7280

только GoogleNet. Одновременно с этим сеть ResNet-50 не способна работать с 4 битами экспоненты, поскольку в процессе вычислений происходит переполнение чисел.

Метрикой качества для модели DeepSpeech-v1 является так называемый WER (Word Error Rate — уровень ошибочных слов). Метрика WER определяется как минимальное количество односимвольных изменений (вставок, удалений или замен), необходимых для получения истинного слова из предсказанного. Результаты для DeepSpeech представлены в табл. 4.

Полученные значения качества показывают, что для нейронной сети DeepSpeech использование 9-разрядного minifloat $\langle 4,4\rangle$ ведёт к потере 0,0047 единиц WER, что в большинстве реальных приложений является допустимым результатом. Использование 8-разрядного minifloat ведёт к заметному падению качества распознавания речи. Однако полученная метрика для minifloat $\langle 5,2\rangle$ и minifloat $\langle 4,3\rangle$ даёт возможность улучшить качество работы нейронной сети посредством дообучения весов и позволит использовать эти типы данных.

Анализ результатов. Исходя из полученных результатов экспериментов можно судить, что для нейронных сетей наибольшую важность имеет диапазон, в котором производится вычисление, а не точность самих вычислений. Это подтверждается фактом существования значения размера экспоненты, после которого качество работы нейронной сети значительно ухудшается.

При изменении размера мантиссы точность исследованных нейронных сетей либо уменьшается постепенно, либо не уменьшается вообще. При этом разрядность мантиссы определяется вычислительной сложностью каждой конкретной архитектуры. Например, самая сложная с вычислительной точки зрения нейронная сеть DeepSpeech-v1 допускает изменение мантиссы вплоть до 4 бит (как и экспоненты), в то время как мобильная архитектура MobileNet-v2 не позволяет уменьшить количество бит (менее 5) в представлении как мантиссы, так и экспоненты без значительной потери точности.

Полученные результаты свидетельствуют о том, что подход вычислений с низкой разрядностью оправдывает себя в случае избыточных нейронных сетей и требует аккуратного применения в случае мобильных видов архитектур нейронных сетей.

Таблица 4

Качество работы DeepSpeech-v1 (исходный WER = 0,1548)

Разрядность экспоненты	Разрядность мантиссы								
	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$	$m = 7$	$m = 8$	$m = 9$	$m = 10$
$e = 3$	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
$e = 4$	0,1881	0,1645	0,1595	0,1553	0,1576	0,1570	0,1560	0,1565	0,1566
$e = 5$	0,1891	0,1615	0,1546	0,1527	0,1558	0,1576	0,1565	0,1548	0,1567

Таблица 5

Оптимальные минифлоаты для используемых сетей

Архитектуры нейронных сетей	Оптимальный minifloat	Относительное падение качества
GoogleNet	$\langle 5, 4\rangle$	0,0091
ResNet-50	$\langle 5, 4\rangle$	0,0027
MobileNet-v2	$\langle 5, 5\rangle$	0,0101
DeepSpeech-v1	$\langle 4, 4\rangle$	0,0047

Заключение. Исследована применимость низкоразрядных представлений чисел с плавающей запятой (minifloat) для работы нейронных сетей. Эксперименты, выполненные с моделями GoogleNet, ResNet-50, MobileNet-v2, DeepSpeech-v1, продемонстрировали возможность использования универсального типа данных — 11-разрядного minifloat $\langle 5, 5\rangle$ в качестве основного типа данных для вычислений с аккумулятором float16. Как следует из табл. 5, для некоторых нейронных сетей возможно применение и более низкоразрядного типа данных.

Малая потеря качества работы нейронных сетей на 10-разрядных minifloat свидетельствует о возможности дообучения весов моделей и последующего использования без потери точности.

Полученные результаты позволяют судить об эффективности создания специализированных аппаратных вычислителей, например, на базе FPGA, предназначенных для работы нейронных сетей, поскольку сокращение разрядности основного типа ведёт к уменьшению объёма данных и потребляемой устройством электроэнергии.

СПИСОК ЛИТЕРАТУРЫ

1. **Кондратьев А. Ю., Гончаренко А. И., Зюбин В. Е.** Исследование применимости облегчённых типов данных с плавающей запятой для нейронных сетей // Сб. ст. II Всерос. науч.-практ. конф. с междунар. участием им. В. В. Губарева «Интеллектуальный анализ сигналов, данных и знаний: методы и средства». Новосибирск, 11–13 декабря, 2018. С. 514–522.
2. **Lai L., Suda N., Chandra V.** Deep convolutional neural network inference with floating-point weights and fixed-point activations // Machine Learning. Cornell Univers., 2017. URL: https://arxiv.org/abs/1703.03073 (дата обращения: 18.04.2019).
3. **Wang N., Choi J., Brand D. et al.** Training deep neural networks with 8-bit floating point numbers // Proc. of the 32nd Intern. Conf. on Neural Information Processing Systems. Montreal, Canada, 3–8 Dec., 2018. P. 7675–7684.
4. **Szegedy Ch., Liu W., Jia Y. et al.** Going deeper with convolutions // Proc. of the IEEE Conf. on Computer Vis. and Pattern Recognition (CVPR). Boston, USA, 7–12 June, 2015. P. 1–9.
5. **He K., Zhang X., Ren Sh., Sun J.** Deep residual learning for image recognition // Proc. of the IEEE Conf. on Computer Vis. and Pattern Recognition (CVPR). Las Vegas, USA, 27–30 June, 2016. P. 770–778.

6. **Sandler M., Howard A., Zhu M. et al.** Mobilenetv2: Inverted residuals and linear bottlenecks // Proc. of the IEEE Conf. on Computer Vis. and Pattern Recognition (CVPR). Salt Lake City, USA, 18–23 June, 2018. P. 4510–4520.
7. **Hannun A., Case C., Casper J. et al.** Deep speech: Scaling up end-to-end speech recognition // Computation and Language. Cornell Univers., 2014. URL: https://arxiv.org/abs/1412.5567 (дата обращения:27.05.2019).
8. **TensorFlow-Slim** image classification model library. URL: https://github.com/tensorflow/models/tree/master/research/slim (дата обращения: 18.04.2019).
9. **Russakovsky O., Deng J., Su H. et al.** Imagenet large scale visual recognition challenge // Intern. Journ. Comput. Vis. 2015. **115**, Iss. 3. P. 211–252.
10. **Project DeepSpeech:** An open source Speech-To-Text engine. URL: https://github.com/mozilla/DeepSpeech (дата обращения: 18.04.2019).
11. **Panayotov V., Chen G., Povey D., Khudanpur S.** Librispeech: An ASR corpus based on public domain audio books // Proc. of the IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP). Brisbane, Australia, 19–24 Apr., 2015. P. 5206–5210.

Поступила в редакцию 27.05.2019
После доработки 02.09.2019
Принята к публикации 03.09.2019