

Использование бинарных нейронных сетей для быстрой классификации изображений в условиях ограниченных ресурсов

О. Г. Малеев

Санкт-Петербургский политехнический университет Петра Великого

maleev_og@spbstu.ru

Аннотация. Сверточные нейронные сети (CNN) демонстрируют высокую эффективность в задачах компьютерного зрения, однако их развертывание на ресурсно-ограниченных платформах (мобильные устройства, встраиваемые системы, датчики IoT) затруднено из-за высоких требований к вычислительным ресурсам и памяти. Бинарные нейронные сети (BNN), в которых веса и активации представлены значениями +1 и -1, позволяют заменить ресурсоемкие операции умножения на побитовые операции XNOR и popcount, сокращая размер модели до 32 раз и ускоряя инференс на порядок. В настоящей работе предложен метод адаптивной бинаризации с автоматической доработкой архитектуры сети, объединяющий подход AdaBinNet (адаптивный подбор бинарных множеств для весов и активаций) с ViReal-соединениями, обеспечивающими сохранение информационного потока через сеть. Эксперименты проведены на архитектурах ResNet-18 и VGG-11 с использованием датасета из 15 классов транспортных средств. Показано, что предложенный метод обеспечивает точность классификации, сопоставимую с полноточными моделями. Для предложенной архитектуры на базе Binary VGG-11 с автоматическим добавлением ViReal-соединений top-1 accuracy составила 74.22% (снижение на 1.32% относительно полноточной модели). Экспериментальные результаты подтверждают перспективность предложенного метода для создания энергоэффективных систем компьютерного зрения на мобильных и встраиваемых устройствах.

Ключевые слова: бинарные нейронные сети; сверточные нейронные сети; AdaBinNet; ResNet; VGG; классификация изображений; квантование

I. ВВЕДЕНИЕ

Компьютерное зрение является одним из наиболее динамично развивающихся направлений искусственного интеллекта. Задача классификации изображений находит применение в автономных транспортных средствах, медицинской диагностике, системах безопасности, промышленной инспекции и многих других областях. Основным инструментом решения этой задачи остаются сверточные нейронные сети (CNN) [1].

Развитие CNN идет по пути увеличения глубины и сложности архитектур. Даже относительно простая ResNet-34 [2] содержит около 21.8 млн параметров и требует 3.68×10^9 операций с плавающей запятой (FLOPs), а современные модели могут насчитывать сотни миллионов параметров. Это создает серьезные препятствия для развертывания CNN на устройствах с ограниченными ресурсами — встраиваемых устройствах, датчиках промышленного интернета вещей (IIoT) и портативных приборах.

Бинарные нейронные сети (BNN) представляют собой экстремальную форму квантования, при которой веса и активации кодируются всего одним битом (значения +1 и -1). Это позволяет заменить дорогостоящие операции умножения на побитовые XNOR и подсчет битов (popcount), которые могут быть эффективно реализованы на аппаратном уровне практически любого современного процессора, включая ARM-совместимые устройства и микроконтроллеры. Теоретически BNN обеспечивают сокращение требований к памяти в 32 раза и ускорение инференса до 58–64 раз.

Прямое квантование параметров, работающее неплохо до 4 бит, при дальнейшем снижении разрядности ведет к существенной потере точности или даже рассыпанию модели. Очевидно, сложности возникают из-за трех основных факторов:

- (1) дискретизация непрерывного распределения весов приводит к ошибкам квантования;
- (2) недифференцируемость функции sign затрудняет обучение градиентными методами;
- (3) ограниченная репрезентативная способность бинарных признаков снижает информативность модели.

II. ОБЗОР ЛИТЕРАТУРЫ

Первые работы по методологии бинаризации нейронных сетей были представлены в 2016 году [3]. Ключевым вкладом стала разработка метода обучения BNN с использованием straight-through estimator (STE), позволяющего обойти проблему недифференцируемости функции sign при обратном распространении ошибки. В работе было показано, что BNN могут достигать точности, сопоставимой с полноточными сетями на простом датасете (CIFAR-10), однако на более сложных задачах разрыв оставался существенным из-за быстрого затухания градиента.

В 2018 году предложена архитектура Bi-Real Net [4], в которой в архитектуру нейронной сети вводятся дополнительные ViReal-соединения (аналог skip-connections в архитектуре ResNet), которые пропускают вещественные активации через бинарные сверточные блоки, сохраняя информационный поток. Кроме того, авторы предложили улучшенную аппроксимацию производной для функции sign , повышающую стабильность обучения.

В 2022 году предложена архитектура AdaBinNet [5], где вместо фиксированного бинарного множества

$\{-1,+1\}$ подбираются оптимальные значения b_1 и b_2 на основе статистического анализа распределения весов и активаций для каждого слоя. Как отмечают авторы, разнообразие распределений весов и активаций в современных глубоких нейронных сетях очень высоко, поэтому стандартная sign-функция не может эффективно использоваться для бинаризации полноточных значений. Несмотря на большое количество исследований в этой области [6–9], проблема повышения точности BNN остается актуальной. Известно применение бинарных нейронных сетей для детектирования объекта на изображении [10]. Цель настоящей работы — разработка и экспериментальная оценка метода бинаризации сверточных нейронных сетей, позволяющего сохранить высокую точность классификации изображений при значительном снижении вычислительных затрат.

III. МЕТОДЫ

A. Обучение бинарных нейронных сетей

В бинарных нейронных сетях (BNN) веса и/или активации принимают только два значения: например, $+1$ и -1 . Для превращения вещественных чисел в бинарные часто используют функцию sign:

$$\text{sign}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

При прямом проходе (forward pass) такое преобразование не вызывает проблем, просто заменяем реальный вес w на $w_b = \text{sign}(w)$. Но при обратном распространении нужна производная $\frac{\partial w_b}{\partial w}$. Производная sign везде (кроме нуля) равна 0:

$$\frac{d}{dw} \text{sign}(w) = 0, w \neq 0.$$

Это значит, что градиент L по бинарному весу w_b не передается дальше на реальный вес w :

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial w_b} \cdot \frac{\partial w_b}{\partial w} = \frac{\partial L}{\partial w_b} \cdot 0 = 0.$$

Исчезновение градиента можно обойти, используя специальный сквозной оценщик градиента (Straight-Through Estimator),

На прямом проходе оставляем функцию sign:

$$w_b = \text{sign}(w).$$

На обратном проходе при оценке градиента используем:

$$\frac{\partial L}{\partial w} \approx \frac{\partial L}{\partial w_b}.$$

Это равносильно замене $\frac{\partial w_b}{\partial w}$ на 1:

$$\frac{\partial w_b}{\partial w} \approx 1.$$

Из сказанного выше следует, что обучение бинарных нейронных сетей является нетривиальной задачей и требует использования дополнительных алгоритмов для обучения и трансформации архитектуры.

Рассмотрим предлагаемый в данном исследовании подход в три этапа: бинаризация весов, бинаризация активаций и автоматическое изменение архитектуры нейронной сети.

B. Бинаризация весов

Стандартная бинаризация использует фиксированное множество $\{-1,+1\}$. Мы использовали предложенный в AdaBinNet подход, который предлагает для каждого сверточного слоя нейронной сети находить оптимальное бинарное множество $\{b_1, b_2\}$, где $b_1, b_2 \in \mathbb{R}$, $b_1 < b_2$, и порог β_w , разделяющий веса на два кластера.

Рассмотрим свертки с ядром $k \times k$, входными каналами C_{in} и выходными C_{out} , при размере карты признаков $H \times W$,

Пусть $w \in \mathbb{R}^{C_{out} \times C_{in} \times k_h \times k_w}$ — весовой тензор слоя.

Введем параметр центра β (сдвиг) и параметр масштаба α , тогда бинарное множество $\{b_1, b_2\}$ можно определить как:

$$b_1 = \beta - \alpha, b_2 = \beta + \alpha.$$

Бинаризованные веса вычисляются по правилу:

$$w_b = \begin{cases} b_1, & \text{если } w < \beta_w, \\ b_2, & \text{если } w \geq \beta_w, \end{cases}$$

где β_w — адаптивный порог (в работе используется $\beta_w = \text{mean}(w)$).

Параметры β и α определяются на этапе обучения. Далее используется straight-through estimator (STE). Градиенты вычисляются как:

$$\frac{\partial L}{\partial \beta} = \sum_i \frac{\partial L}{\partial w_{b,i}} \cdot \frac{\partial w_{b,i}}{\partial \beta}, \frac{\partial L}{\partial \alpha} = \sum_i \frac{\partial L}{\partial w_{b,i}} \cdot \frac{\partial w_{b,i}}{\partial \alpha},$$

причем производные $\partial w_{b,i} / \partial \beta$ и $\partial w_{b,i} / \partial \alpha$ равны $+1$ или -1 в зависимости от знака $(w_i - \beta_w)$.

C. Адаптивная бинаризация активаций

Вместо стандартной функции sign для активаций вводится параметрическая функция **Maxout**[5] с обучаемыми коэффициентами. Для вещественной активации a :

$$a_b = \text{Maxout}(a) = \max(p \cdot a + q, r \cdot a + s),$$

где p, q, r, s — обучаемые параметры. Мы использовали упрощенный вариант, рассчитываем a_b по формуле:

$$a_b = \begin{cases} \beta_{\text{pos}} \cdot a, & \text{если } a \geq 0, \\ \beta_{\text{neg}} \cdot a, & \text{если } a < 0, \end{cases}$$

после чего применяется функция sign: $\tilde{a}_b = \text{sign}(a_b)$.

D. Автоматическое изменение архитектуры нейронной сети

Проведенные эксперименты показали, что более сложные архитектуры сверточных нейронных сетей, в отличие от рассмотренных в работе[5], при преобразовании к бинарной форме могут плохо обучаться по алгоритму AdaBinNet. Например, для моделей типа VGG[11], где отсутствуют внутренние соединения между блоками (skip-connections), прямое применение бинарных сверток ведет к резкой потере информации. Для таких архитектур в данном исследовании предлагается автоматически добавлять соединение от входа к выходу бинарного сверточного блока после нормализации, подобно, как это предложено в Bi-real архитектуре:

$$\text{out} = \text{BatchNorm}(\text{BinaryConv}(x)) + x$$

В данной работе реализован алгоритм автоматического анализа преобразуемой к бинарному виду архитектуры и добавления дополнительных соединений после сверточного слоя, там, где это необходимо.

E. Общий алгоритм бинаризации нейронной сети/автоматическое изменение архитектуры нейронной сети

При выполнении данной работы разработан метод бинаризации, который включает автоматический анализ архитектуры нейронной сети, выбор и замену стандартных модулей сети на бинарные аналоги.

Процесс начинается с определения модулей для замены, при этом специальные функции помогают идентифицировать элементы, которые не следует включать в процесс бинаризации, основываясь на их конфигурации и названиях слоёв. Включенный словарь специальных значений облегчает поиск первого и последнего вхождения модуля, которые обычно не подвергаются бинаризации из-за их критического влияния на точность модели. Алгоритм рекурсивно обрабатывает все дочерние модули, выполняя бинаризацию.

Для стабилизации и ускорения обучения бинарных моделей также алгоритмом автоматически может быть добавлен слой нормализации по мини-батчам, что позволяет улучшить стабильность обучения бинарной нейронной сети.

Проведенные эксперименты на архитектурах ResNet и VGG продемонстрировали работоспособность предложенного автоматического алгоритма преобразования к бинарной сети.

IV. ЭКСПЕРИМЕНТЫ

Для проведения экспериментов сконструирован датасет из 15 классов транспортных средств на основе датасета ImageNet[12]. Выбранные классы включают различные типы транспортных средств: легковая машина, грузовая машина, скорая помощь, поезд, дом на колесах, пикап и др. Полученный датасет включает в себя 18000 тренировочных изображений, 750 валидационных и 1500 тестовых для проверки обученных моделей.

В экспериментах использовались метрики Top-1 и Top-5 — стандартные метрики качества классификации

для датасета ImageNet. Метрика Top-1 определяется как доля примеров, где самый вероятный предсказанный моделью класс полностью совпадает с истинной меткой. Метрика Top-5 определяется как доля примеров, где истинная метка входит в 5 наиболее вероятных ответов модели.

В качестве базовых архитектур для экспериментов выбраны ResNet-18 и VGG-11. ResNet-18 представляет современную архитектуру с механизмом skip-connections, эффективную для глубокого обучения. VGG-11 выбрана как классическая последовательная архитектура, не имеющая встроенных механизмов пропуска сигнала между блоками.

Кроме точности оценивались также приближенные значения количества операций FLOPs (число операций с плавающей точкой) и BOPs (число бинарных операций) для моделей и требования к памяти.

Для оценки BOPs операций использовалось допущение, что для свертки с ядром $k \times k$, входными каналами C_{in} и выходными C_{out} , при размере карты признаков $H \times W$ количество BOPs составляет:

$$\text{BOPs} = H \cdot W \cdot C_{in} \cdot C_{out} \cdot k^2.$$

Анализ результатов экспериментов демонстрирует, что разработанный подход обеспечивает точность классификации Top-1 и Top-5, сопоставимую с полноточными моделями. Например, бинаризация модели ResNet18 с использованием предложенного метода (см. таблица 1) позволила достичь top-1 accuracy на уровне 80.81%, что лишь на 0.32% уступает исходной полноточной модели, при этом размер модели уменьшился примерно в 4 раза, количество операций FLOPs сокращено на 94.4%.

Результаты экспериментов представлены в табл. 1.

ТАБЛИЦА I. Точность полноточной и бинарной модели ResNet-18

Модель	Top-1 точность	Top-5 точность	Размер, Мб	FLOPs / BOPs
ResNet-18 (FP32)	81.13%	96.92%	42.66	1.83×10^9
Бинарная ResNet-18 (AdaBinNet)	80.81%	96.83%	10.74	1.68×10^8 / 1.70×10^9

Для демонстрации возможностей предлагаемого метода бинаризации было также исследовано преобразование архитектуры VGG-11. Полноточная нейронная сеть преобразована двумя способами: прямое применение подхода AdaBinNet и применение предлагаемого метода автоматической доработки архитектуры с добавлением дополнительных соединений.

Результаты экспериментов представлены в табл. 2.

ТАБЛИЦА II. Точность полноточной и бинарной модели VGG-11

Модель	Top-1 точность	Top-5 точность	Размер, Мб	FLOPs/BOPs
VGG-11 (FP32)	75.54%	96.29%	491.43	7.63×10^9

Модель	Тор-1 точность	Тор-5 точность	Размер, Мб	FLOPs/ BOPs
Binary VGG-11 (только AdaBinNet)	≈ 62%	—	≈ 123	≈ 3.1×10 ⁸
Binary VGG-11 (AdaBinNet + дополнительные соединения по предложенному алгоритму)	74.22%	95.89%	122.90	3.12×10 ⁸ / 7.39×10 ⁹

Проведенные эксперименты наглядно демонстрируют эффективность предлагаемого метода автоматической доработки архитектуры. Если применение стандартного подхода бинаризации AdaBinNet дало неприемлемо низкую точность ≈ 62%, то использование предлагаемого метода внедрения дополнительных соединений повысило точность до 74.22%, снизив разницу в топ-1 с полноточной моделью до 1.32%. При этом число операций FLOPs выросло незначительно.

V. ЗАКЛЮЧЕНИЕ

В настоящей работе предложен метод адаптивной бинаризации сверточных нейронных сетей, объединяющий подход AdaBinNet с автоматическим преобразованием архитектуры нейронной сети.

Ключевые результаты:

- Предложен метод бинаризации нейронной сети с автоматической доработкой архитектуры, позволяющий существенно снизить требования к памяти и производительности вычислительного блока в задаче классификации изображений.
- Эксперименты с бинаризованной архитектурой ResNet-18 продемонстрировали точность метрики Тор-1 80.81% (снижение на 0.32% относительно полноточной модели), размер модели сокращен в 4 раза, число операций FLOPs — на 94.4%.
- Предложенный алгоритм автоматического преобразования нейронной сети при бинаризации позволил для архитектуры VGG-11 повысить точность с ≈ 62% до 74.22% (снижение на 1.32% относительно полноточной версии).

Полученные результаты подтверждают потенциал применения бинарных сверточных нейронных сетей в мобильных и маломощных устройствах. Разработанный метод может быть использован для создания энергоэффективных систем компьютерного зрения в робототехнике, системах безопасности, промышленном интернете вещей и портативных приборах.

СПИСОК ЛИТЕРАТУРЫ

- [1] LeCun Y. и др. Backpropagation Applied to Handwritten Zip Code Recognition // *Neural Comput.* 1989. Т. 1, № 4. С. 541–551.
- [2] He K. и др. Deep residual learning for image recognition // *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE, 2016. Т. 2016-Decem. С. 770–778.
- [3] Courbariaux M. и др. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1 // *ArXiv.* 2016. Т. abs/1602.0.
- [4] Liu Z. и др. Bi-Real Net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* 2018. Т. 11219 LNCS. С. 747–763.
- [5] Tu Z. и др. AdaBin: Improving Binary Neural Networks with Adaptive Binary Sets // *Lecture Notes in Computer Science.* 2022. Т. 13671 LNCS. С. 379–395.
- [6] Han K. и др. Training Binary Neural Networks through Learning with Noisy Supervision // *Proceedings of the 37th International Conference on Machine Learning / под ред. III H. D., Singh A. PMLR,* 2020. Т. 119. С. 4017–4026.
- [7] Bethge J. и др. Meliusnet: An improved network architecture for binary neural networks // *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2021. С. 1439–1448.
- [8] Rastegari M. и др. XNOR-net: Imagenet classification using binary convolutional neural networks // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* 2016. Т. 9908 LNCS. С. 525–542.
- [9] Lee C. и др. INSTA-BNN: Binary Neural Network with INSTance-aware Threshold // *Proceedings of the IEEE International Conference on Computer Vision.* 2023. С. 17279–17288.
- [10] Королев Д. О., Малеев О. Г. Исследование эффективности применения бинарных нейронных сетей при детектировании объекта на изображении // *Информатика и её применения.* 2023. Т. 17, № 3. С. 88-92. EDN: TOCVAL.
- [11] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition // *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* 2015.
- [12] Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks // *Communications of the ACM.* 2017. Т. 60, № 6. С. 84–90.