

Малобитные квантованные нейронные сети в системе остаточных классов с классическим или усредняющим накоплением

М. В. Зингеренко^{*,**},

^{*} ООО «Смарт Энджинс Сервис», г. Москва, Россия

^{**} Московский физико-технический институт (национальный исследовательский университет),
г. Долгопрудный, Россия

Поступила в редколлегию 07.10.2025 г. Принята 12.12.2025 г.

Аннотация—Для решения задач компьютерного зрения на устройствах с ограниченными ресурсами применяются малобитные равномерно квантованные нейросети, сохраняющие высокую точность при простой аппаратной реализации на центральных процессорах. В работе предложена модель вычисления равномерно квантованных нейронных сетей в системе остаточных классов, обеспечивающей параллельную аппаратно-эффективную арифметику без переносов и предназначенной для программируемых логических интегральных схем и специализированных устройств. Представлены модели всех основных слоев: линейных, сверточных, кусочно-линейных активаций и реквантование. Выполнена оценка разрядности аккумуляторов и выбор наборов модулей, гарантирующих отсутствие переполнения. Для повышения эффективности аппаратной реализации линейных слоев предложено использовать усредняющее накопление, которое позволяет уменьшить требуемую разрядность аккумулятора.

КЛЮЧЕВЫЕ СЛОВА: равномерное малобитное квантование, сверточные нейронные сети, система остаточных классов, энергоэффективность.

DOI: 10.53921/18195822_2025_25_4_896

1. ВВЕДЕНИЕ

Нейронные сети все чаще становятся вычислительным ядром современных интеллектуальных систем, обеспечивая работу систем в компьютерном зрении, обработке естественного языка, робототехнике и автономной навигации. Сверточные нейронные сети (СНС), широко применяются в областях медицинской визуализации [1], беспилотного транспорта [2], детекции объектов в реальном времени [3, 4] благодаря их способности обучаться иерархическим представлениям признаков с высокой точностью.

Чтобы сделать нейронные сети более эффективными на этапе исполнения, широко используется квантование. Оно уменьшает разрядность весов и активаций (например, до 8-битных или 4-битных целых чисел), резко снижая требуемый объем памяти и время вычислений [5]. На сегодняшний день известно множество подходов, позволяющих добиться того, что квантованные модели работают не только быстро, но и точно за счет калибровки и обучения с учетом квантования [6].

Одной из основных моделей квантования нейросетей является равномерное (аффинное) квантование. Оно спроектировано для быстрой работы на центральных процессорах (ЦП) общего назначения за счет использования компактных целочисленных данных и векторных вычислений.

Однако существует класс еще более ресурсно-ограниченных платформ, таких как программируемые логические интегральные схемы (ПЛИС) и интегральные схемы специального назначения, где критически важны энергоэффективность, скорость работы и отказоустойчивость. На таких платформах использование арифметики с плавающей точкой или даже универсальных целочисленных вычислительных блоков может быть неэффективным или невозможным. Для подобных условий одним из решения являются вычисления в системе остаточных классов (СОК). Представляя числа в виде векторов остатков, СОК обеспечивает вычисления без переносов и допускает эффективную параллельную реализацию, что естественным образом отображается на специализированную цифровую логику и аппаратные ускорители с ограничением по мощности [7].

Хотя СОК уже рассматривалась в контексте нейросетевых вычислений, большинство предыдущих работ сосредоточено на развертывании небольших, оптимизированных вручную сетей или на моделях, использующих тип данных с фиксированной точкой. В то же время применение равномерно квантованных моделей в области СОК остается практически не исследованным, несмотря на их возрастающую значимость и вычислительную эффективность на аппаратуре общего назначения.

Данная работа посвящена исследованию перспектив реализации равномерно квантованных нейронных сетей в системе остаточных классов. Предложен метод преобразования таких моделей (например, с 8-битным или 4-битным квантованием) в форму, позволяющую выполнять их вычисления полностью в СОК. Приведены оценки требуемой разрядности аккумуляторов для подбора подходящих модулей СОК и проведен анализ аппаратных характеристик различных конфигураций квантования. Экспериментально показано, что вычисления в СОК могут быть применены к существующим предобученным квантованным моделям после короткого дообучения.

Таким образом, показано, что равномерные модели квантования и методы обучения могут использоваться на вычислителях на базе СОК.

2. ОБЗОР ЛИТЕРАТУРЫ

2.1. Система остаточных чисел для вычислений нейронных сетей

Использование системы остаточных классов для ускорения нейронных сетей имеет давнюю историю исследований. Ранние работы [8] продемонстрировали реализуемость простых перцептронов с использованием остаточной арифметики. Позднее эту идею развили, предложив систолическую СОК-архитектуру [9] для многослойных перцептронов, использующую присущую модулярным операциям параллельность.

В дальнейшем в [10] представили вложенную СОК-структуру, ориентированную на исполнение СНС на ПЛИС, и показали, как 48-битные операции умножения-накопления могут быть разложены на меньшие модулярные блоки для получения низкой задержки вычислений и ресурсно-эффективной свертки.

Прорывом стала RNSNet [11] – полноценная архитектура ускорителя «in-memory», способная выполнять СНС полностью в области СОК. В этой конструкции умножители были заменены таблицами поиска, а для накопления использовались операции модулярного сложения, что обеспечило экономию энергии в 145.5 раз и ускорение в 35.4 раз по сравнению с традиционными реализациями на графическом процессоре. Однако данная работа не рассматривает малобитные представления и демонстрирует ограниченный набор архитектур нейронных сетей.

Последующие исследования обобщили СОК-ориентированную аппаратную реализацию на более глубокие нейросетевые архитектуры [12]. В [13] предложили полноценный ускоритель

глубоких нейронных сетей на основе СОК-арифметики, показав существенную экономию площади и энергии. В еще одной работе [14] дополнительно упростили архитектуру вычислителя, используя разреженность весов и полностью устранив умножители за счет распределенной арифметики.

2.2. Методы квантования

Квантование – устоявшийся подход к снижению вычислительной сложности нейронных сетей, особенно для исполнения на периферийных устройствах. Якоб и др. [15] показали, что при использовании обучения с учетом квантования и калибровки равномерно квантованные 8-битные (INT8) модели сохраняют ту же точность, что и вещественные модели полной разрядности.

При этом все больше внимания уделяется форматам сверхнизкой точности, таким как 4-битные (INT4) и менее модели. В [16] предложили метод равномерного INT4 квантования весов, в котором для их подбора оптимизируется минимальная среднеквадратичная ошибка, что позволило минимизировать падение точности без дообучения. Параметрическая схема 4.6-битного квантования, предложенная в [17], обеспечивает ускорение в 1.5–1.6 раз по сравнению с INT8 при сохранении качества распознавания в задачах компьютерного зрения.

Лиу и др. [18] недавно представили всеобъемлющий обзор методов квантования с разрядностью 4–6 бит, выделив ключевые компромиссы между аппаратной эффективностью, точностью и простотой внедрения.

Эти исследования подтверждают, что равномерное квантование хорошо подходит для вычисления нейронных сетей как на центральных процессорах, так и на специализированной аппаратуре.

2.3. Квантованные нейросети и СОК

Хотя и квантование, и СОК подробно изучались по отдельности, их совместное применение остается относительно малоисследованным. Одной из немногих работ, связывающих эти направления, является [19], где СОК-арифметика применялась к реализации свертки алгоритмом Винограда, использующей 8-битные коэффициенты. Предложенный метод продемонстрировал ускорение в 2.3–4.7 раз.

3. РАВНОМЕРНОЕ КВАНТОВАНИЕ

Равномерное квантование является базовой техникой, применяемой для ускорения работы современных нейронных сетей. Принцип его работы заключается в преобразовании тензоров в формате с плавающей точкой (весов и активаций нейросети) в низкоразрядные целочисленные представления, которые обеспечивают вычислительно-эффективную аппаратную реализацию. Этот подход в первую очередь рассчитан на центральные процессоры и особенно полезен в практических приложениях, таких как вычисления в автономных системах и на конечных устройствах [20], например, мобильных телефонах [21].

3.1. Квантование и деквантование

Пусть дан вещественный вход $r \in [r_{\min}, r_{\max}]$. Равномерное квантование отображает его в целое число $q(r) \in [q_{\min}, q_{\max}]$ с помощью кусочно-линейного преобразования:

$$q(r) = \min \left(\max \left(\left\lfloor \frac{r}{s} + z \right\rfloor, q_{\min} \right), q_{\max} \right), \quad (1)$$

где s — положительный коэффициент масштабирования, а z — нулевая точка, обеспечивающая точное представление значения $r = 0$ (что важно, например, для операций дополнения нулями в слоях нейронной сети). Эта формулировка поддерживает как симметричные ($z = 0$), так и асимметричные ($z \neq 0$) схемы квантования.

Параметры s и z , как правило, определяются либо с помощью динамической калибровки (например, по диапазону минимум–максимум, с отсечением по процентилям [22]), либо в ходе обучения с учетом квантования, когда сеть дообучается так, чтобы сохранять качество несмотря на дискретизацию параметров [23].

Обратный процесс — деквантование — отображает целое значение q в вещественное \hat{r} :

$$\hat{r} = s \cdot (q - z)$$

Равномерное квантование обычно применяется к тензорам весов и активаций нейронных сетей с отдельными параметрами для каждого слоя или даже для каждого канала тензора. В данной работе рассматривается квантование с отдельными параметрами для каждого тензора.

3.2. Разрядности квантования

- **8 бит (INT8)** — такое квантование широко поддерживается на различных аппаратных платформах и обеспечивает хорошее соотношение точности и вычислительной эффективности: ускорение исполнения примерно в 2 раза по сравнению с вещественными моделями при падении качества менее чем на 1% [24].
- **4 бита (INT4)** — дает дополнительное сжатие и вычислительные преимущества (скорость растет в 3–4 раза по сравнению с вещественными сетями). Хотя возможна небольшая потеря точности, при использовании обучения с учетом квантования такой формат демонстрирует достаточно высокое качество и все чаще применяется в сильно ограниченных по ресурсам условиях [25].
- **4.6 бита** — квантование, предложенное в [26], обеспечивает примерно 23 уровня квантования, демонстрируя заметно более высокую точность при скорости, сопоставимой с 4-битными моделями.
- **1–2 бита** — такое квантование радикально уменьшает как размер модели, так и стоимость вычислений, обеспечивая сжатие до 16 раз по сравнению с вещественными нейросетями. Однако такие модели заметно снижают точность распознавания в прикладных задачах, поэтому применяются не столь широко.

3.3. Полный цикл квантования нейронной сети

Во время исполнения все основные слои квантованной нейронной сети работают с целочисленными весами и активациями. Рассмотрим соответствующие вычисления.

Квантованные линейные и сверточные слои. Пусть x — входной тензор, w — вес, b — смещение. Их квантованные формы согласно (1) обозначим как q_x с параметрами s_x, z_x и q_w с параметрами s_w, z_w . Вещественное скалярное произведение y , выполняемое линейным слоем:

$$y = \sum_i x_i \cdot w_i \approx s_x \cdot s_w \cdot \sum_i (q_{x,i} - z_x)(q_{w,i} - z_w)$$

Таким образом, квантованная часть вычисляется как:

$$q_{\text{acc}} = s_x s_w \left(\sum_i q_{x,i} q_{w,i} - z_w \sum_i q_{x,i} - z_x \sum_i q_{w,i} + \sum_i z_x z_w \right) \quad (2)$$

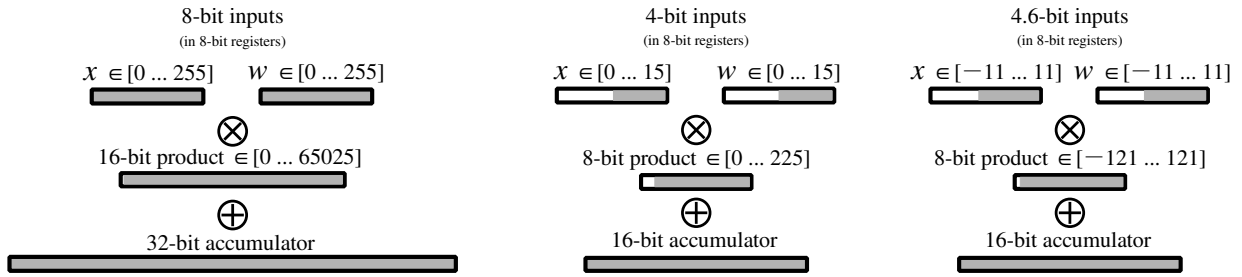


Рис. 1. Квантованное целочисленное умножение на ЦП для различных схем квантования.

Для хранения этого значения без переполнения используются аккумуляторы расширенной разрядности. Пример такой операции на ЦП показан на рис. 1. Заметим, что 4- и 4.6-битные входные регистры используются не полностью.

Линейный слой с древовидным усреднением

В ранее рассмотренном сверточном слое сумма произведений растёт пропорционально числу членов, поэтому для предотвращения переполнения требуется аккумулятор расширенной разрядности (например, 32 бита при 16-битных промежуточных значениях). Для вычислений в системе остаточных классов это означает необходимость увеличивать параметр t набора модулей и усложнять реализацию арифметики. Схема древовидного усреднения, позволяет снизить требуемую разрядность аккумулятора [27].

Ее идея показана на рис. 2 заключается в следующем: вместо последовательного накопления суммы в расширенном аккумуляторе выполняется попарное объединение слагаемых с помощью операции усреднения. На каждом уровне дерева формируются усреднённые суммы пар чисел, затем пар усреднённых сумм и т.д., пока не останется одно значение. В простейшем случае, когда число слагаемых является степенью двойки, процесс сводится к регулярному двоичному дереву усреднений.

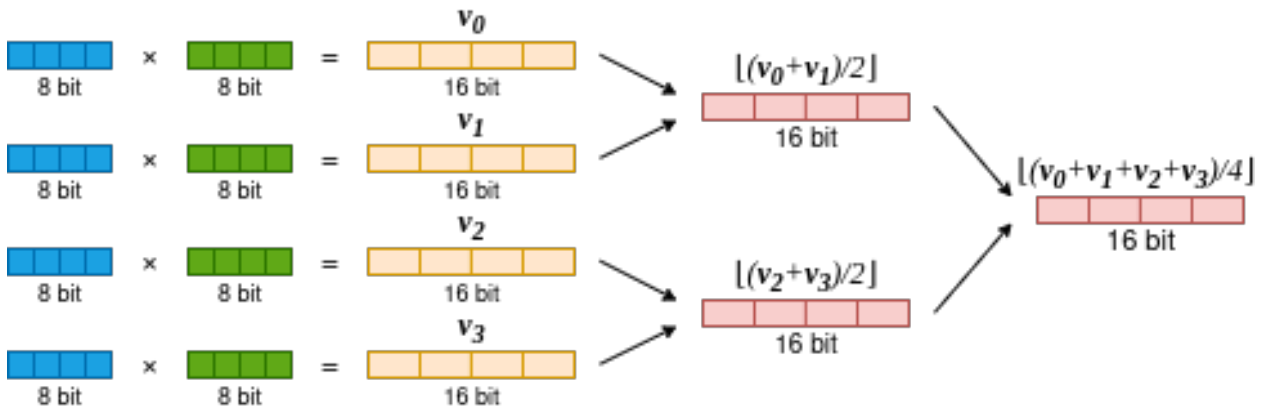


Рис. 2. Визуализация процесса древовидного усреднения при накоплении.

Главное свойство такого процесса состоит в том, что промежуточные результаты остаются в той же разрядности, что и входные значения. Например, при накоплении 16-битных чисел (типично возникающих как произведения 8-битных величин) усреднение позволяет удерживать значения в пределах 16 бит на всех уровнях дерева, избегая переполнения. Аналогично,

для 4-битных входов произведения имеют малую разрядность, и накопление можно выполнять в 8 битах, что значительно меньше, чем требуется при линейном суммировании.

После вычисления целочисленного среднего по множеству членов, итоговое значение необходимо масштабировать на «глубину» скалярного произведения (эффективное число слагаемых). Это масштабирование выполняется в конце и может требовать расширенной разрядности, однако оно выполняется один раз, тогда как основная часть накопления остаётся в фиксированном низкоразрядном формате.

Реквантование. После накопления результат может выходить за диапазон входных значений следующего слоя. Реквантование масштабирует его к целевой разрядности:

$$q_{\text{out}} = \left\lfloor \frac{M_{\text{req}} \cdot q_{\text{acc}}}{2^r} \right\rfloor + z_{\text{out}},$$

где q_{acc} – накопленный целочисленный результат, M_{req} – заранее вычисленный целочисленный множитель, соответствующий отношению масштабов, r – величина сдвига вправо, а z_{out} – новая нулевая точка.

Квантованные кусочно-линейные функции активации. Поскольку нулевая точка z_x соответствует вещественному нулю, кусочно-линейные функции активации, такие как ReLU, реализуются непосредственно над квантованными целыми числами:

$$q_y = \max(q_x, z_x).$$

Для «ограниченных» вариантов, например, ReLU6, квантованные выходы ограничиваются между целочисленными представлениями 0 и 6:

$$q_y = \min(\max(q_x, z_x), \left\lfloor \frac{6}{s_x} \right\rfloor + z_x).$$

Пакетная нормализация. На этапе исполнения слои пакетной нормализации (batch normalization) перед квантованием интегрируются в предыдущий сверточный или линейный слой.

Пулинг Слои пулинга или субдискретизации работают непосредственно с квантованными активациями без деквантования и вычислительно эффективным образом увеличивают рецептивное поле нейронной сети.

4. СИСТЕМА ОСТАТОЧНЫХ КЛАССОВ

Система остаточных классов – это представление чисел, в котором целые числа выражаются как векторы остатков по набору попарно взаимно простых модулей. Первоначально разработанная для быстрого параллельного вычисления арифметических операций, СОК привлекает все больше внимания в задачах аппаратно-эффективных вычислений, особенно в контексте нейросетевых ускорителей.

4.1. Основы СОК

В СОК целое число x представляется набором остатков:

$$[x]_R = (x \bmod m_1, x \bmod m_2, \dots, x \bmod m_k),$$

где $\{m_1, m_2, \dots, m_k\}$ – попарно взаимно простые целые числа. Динамический диапазон представления равен $M = \prod_{i=1}^k m_i$, то есть СОК может однозначно представлять все целые числа из диапазона $[0, M - 1]$.

Операции сложения, вычитания и умножения выполняются независимо для каждого числа набора:

$$(x \oplus y) \bmod m_i = ((x \bmod m_i) \oplus (y \bmod m_i)) \bmod m_i,$$

где \oplus обозначает операцию, например сложение или умножение. Таким образом, СОК устраняет переносы между «разрядами», обеспечивая параллельную арифметику с малой задержкой, особенно подходящую для систолических и вычислений «в памяти» (in-memory) [28].

Для обратного преобразования значения из СОК в позиционную систему обычно используют два метода:

- **Китайская теорема об остатках.** Для вектора остатков (x_1, x_2, \dots, x_k) исходное число $x \in [0, M)$ восстанавливается как:

$$x = \left(\sum_{i=1}^k x_i \cdot M_i \cdot M_i^{-1} \right) \bmod M,$$

где $M = \prod_{i=1}^k m_i$, $M_i = M/m_i$, а M_i^{-1} — мультипликативная обратная величина $M_i \bmod m_i$. Этот метод хорошо распараллеливается, но требует нескольких длинных умножений и модульных редукций.

- **Преобразование в смешанную систему оснований.** Остатки преобразуются в смешанное представление (r_1, r_2, \dots, r_k) , где:

$$r_i = \left(x_i - \left(\sum_{j=1}^{i-1} r_j \cdot \prod_{\ell=1}^{j-1} m_\ell \right) \bmod m_i \right) \cdot \left(\prod_{j=1}^{i-1} m_j \right)^{-1} \bmod m_i.$$

Затем вычисляется

$$x = r_1 + r_2 m_1 + r_3 m_1 m_2 + \dots + r_k m_1 m_2 \dots m_{k-1}.$$

Хотя этот метод избегает длинных умножений, он работает последовательно и, как правило, оказывается медленнее и сложнее для аппаратной реализации.

4.2. Квантованные числа в системе остаточных классов

Чтобы обеспечить аппаратно-эффективную модульную арифметику, используем набор из трех модулей, который наиболее часто применяется в СОК-системах [29]:

$$\text{ModuliSet} = \{2^t - 1, 2^t, 2^t + 1\},$$

где $t = \lceil \frac{n}{3} \rceil$, а n — требуемая разрядность аккумулятора.

Этот набор позволяет получать оптимизированные аппаратные реализации: операции по модулю $2^t \pm 1$ могут быть сведены к простым операциям «сдвиг-и-сложение», избегая деления или больших предподсчитанных таблиц. Как обсуждается в [30], такая структура также обеспечивает эффективную разводку соединений на ПЛИС-устройствах и хорошо подходит для вычислений в памяти благодаря отсутствию переносов и независимой обработке модулей.

Каждое значение X кодируется в СОК следующим образом:

$$X \rightarrow \left\{ R_1 = X \bmod (2^t - 1), R_2 = X \bmod 2^t, R_3 = X \bmod (2^t + 1) \right\}.$$

Сложение в СОК выполняется напрямую и хорошо распараллеливается: каждую операцию можно реализовать с помощью простых модульных сумматоров, не обрабатывающих переполнение.

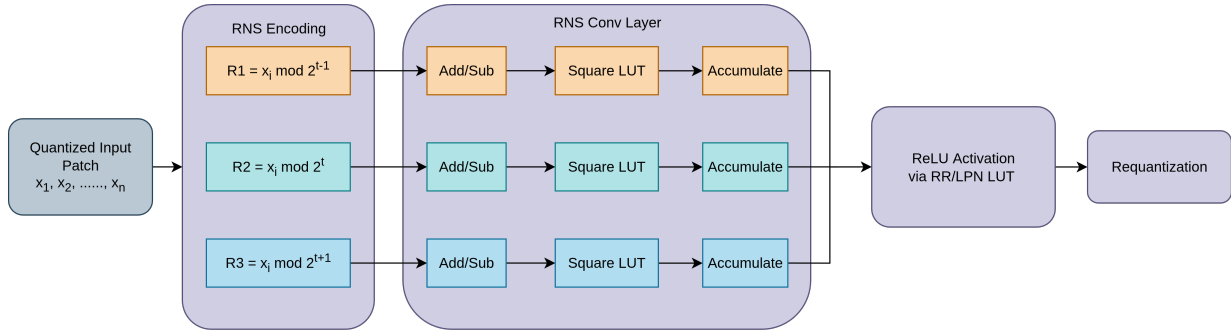


Рис. 3. Схема работы сверточного слоя в СОК.

Умножение можно реализовать, используя тождество, сводящее произведение к разности квадратов:

$$a \cdot b = \frac{(a + b)^2}{4} - \frac{(a - b)^2}{4}. \tag{3}$$

Это тождество использует только сложение, вычитание и операцию возведения в квадрат. Все они вычислительно эффективно реализуются в модульной форме. Значения квадратов можно хранить в предподсчитанной таблице, исключая вычисления «на лету» и дополнительно оптимизируя энергопотребление и площадь.

5. ВЫЧИСЛЕНИЯ НЕЙРОННОЙ СЕТИ В СОК

В этом разделе рассмотрены базовые операции для реализации квантованной нейронной сети в СОК, как показано на рис. 3). Все веса хранятся в формате СОК, а вход сети заранее преобразуется в этот формат. В конце полученный вектор преобразуется обратно в позиционную систему счисления.

5.1. Скалярное произведение

В данной работе скалярное произведение реализуется непосредственно в области СОК с использованием выражения (3). Тогда (4.1) принимает вид:

$$\sum x \cdot w \approx \sum \frac{(q_x + q_w)^2}{4} - \sum \frac{(q_x - q_w)^2}{4} - z_x \cdot \sum q_w - z_w \cdot \sum q_x + \text{const} \tag{4}$$

где z_x и z_w – нулевые точки квантования для активаций q_x и весов q_w . Все операции выполняются параллельно по набору модулей, то есть каждый член выражения обрабатывается независимо в своем канале остатков.

5.2. Слой активации

Кусочно-линейные функции активации особенно удобны в области СОК: если есть примитив для сравнения модулярного значения с нулем или с небольшим набором порогов, то такие функции, как ReLU, ReLU6 или HardTanh, сводятся к нескольким сравнениям и условным присваиваниям и могут быть реализованы без выхода из области СОК.

Однако сравнения в СОК не являются тривиальными из-за отсутствия естественного порядка среди остатков. Чтобы реализовать операции, основанные на сравнении, не восстанавливая значения в позиционной системе, используем метод на основе опорного значения [11].

Каждое число в СОК $X = (R_1, R_2, R_3)$ соответствует единственному целому числу в знаковом диапазоне $[-M/2, M/2)$, где $M = (2^t - 1)2^t(2^t + 1)$. Для любой фиксированной пары (R_1, R_3) существует единственное минимально возможное число (МВЧ) — наименьшее неотрицательное целое число, остатки которого по модулям $2^t - 1$ и $2^t + 1$ равны R_1 и R_3 соответственно. Для этой пары определим опорный остаток (ОО) как остаток МВЧ по модулю 2^t , то есть его компоненту R_2 . Значения ОО предварительно вычисляются и хранятся в небольшой таблице поиска, индексируемой по (R_1, R_3) .

Во время выполнения для заданного $X = (R_1, R_2, R_3)$ определим его знак, сравнивая R_2 с соответствующим $RR(R_1, R_3)$:

$$R_2 < RR(R_1, R_3) \Rightarrow X < 0,$$

$$R_2 \geq RR(R_1, R_3) \Rightarrow X \geq 0.$$

Такое сравнение работает потому, что при увеличении исходного целого числа на единицу пара (R_1, R_3) циклически повторяется с периодом $(2^t - 1)(2^t + 1)$, тогда как соответствующее R_2 на том же периоде монотонно убывает на единицу. Значения RR тем самым кодируют точку, в которой знаковое представление пересекает ноль.

Используя этот примитив, активация ReLU реализуется полностью в области СОК как

$$f(X) = \begin{cases} 0, & \text{если } R_2 < RR(R_1, R_3), \\ X, & \text{иначе.} \end{cases} \quad (5)$$

В более общем случае кусочно-линейные активации, такие как ReLU6 или HardTanh, требуют лишь «обрезки» входа по небольшому числу постоянных порогов. Эти пороги также представлены в СОК, а их соответствующие RR -значения предварительно вычислены, поэтому границы сегментов обрабатываются тем же типом RR -сравнения, что и выше.

5.3. Слой реквантования

Этот модуль критически важен для того, чтобы выход каждого слоя оставался совместимым с входным типом данных следующего слоя. В данной работе рассматриваются квантованные модели, в которых масштабы и смещения определены во время обучения. Как правило, они имеют максимально возможную точность и обеспечивают высокую вычислительную эффективность во время исполнения. Реквантование может выполняться следующим образом:

- **Подход на основе предподсчитанных таблиц.** Отображение значений аккумулятора в СОК в целевое квантованное значение (например, 4-битное или 4.6-битное) можно реализовать с помощью предподсчитанной таблицы. Такая таблица создается заранее и затем используется во время исполнения. Однако важно аккуратно оценить ее размер, чтобы гарантировать, что она укладывается в аппаратные ограничения вычислителя.
- **Вычислительный подход.** Можно применить алгоритм деления [31], чтобы выполнять реквантование напрямую. Более того, так как масштаб известен заранее, можно вычислить его обратную величину, что упростит исполнение.

5.4. Численный эффект реквантования в области СОК

В рассматриваемой реализации квантованной сети в СОК единственным источником неточности является реквантование в области СОК. Рассмотрим реквантование, реализованное с

Таблица 1. Разрядности аккумулятора (Acc) и t для 8-битного квантования.

Сеть	Датасет	Вход	Теор. Acc/ t	Эксп. Acc/ t
ResNet-18	CIFAR-10	$1 \times 10 \times 10$	29/10	20/7
ResNet-18	ImageNet	$3 \times 224 \times 224$	29/10	20/7
UNet	TCSIA	$3 \times 256 \times 256$	29/10	21/7

Таблица 2. Разрядности аккумулятора (Acc) и t для 4-битного квантования.

Сеть	Датасет	Вход	Теор. Acc/ t	Эксп. Acc/ t
ResNet-18	CIFAR-10	$1 \times 10 \times 10$	21/7	14/5
ResNet-18	ImageNet	$3 \times 224 \times 224$	21/7	14/5
UNet	TCSIA	$3 \times 256 \times 256$	21/7	14/5

помощью деления в пространстве остатков. Чтобы количественно оценить его численный эффект, был проведен эксперимент на сверточной сети LeNet-подобного типа [32], обученной на выборке CIFAR-10 [33]. Базовая вещественная модель достигает топ-1 точности 63.53%. Равномерное 8-битное квантование после обучения с использованием максимального и минимального значений диапазона для весов и активаций до дает точность 63.52%, а вычисления в области СОК приводят к точности 63.42%. Таким образом, для 8-битного квантования сама по себе СОК-арифметика вносит лишь пренебрежимо малую деградацию.

Для 4-битного квантования базовая LeNet-подобная модель с квантованием после обучения достигает точности 51.46% на CIFAR-10. Когда же все операции выполняются в рамках СОК-конвейера, точность падает до 30.36%. При этом одиночная операция деления в СОК дает лишь небольшое расхождение на целых значениях (поэлементная ошибка ограничена 1, а средняя абсолютная ошибка составляет около 0.2), однако такие возмущения накапливаются на протяжении нескольких слоев, приводя к существенной потере качества. Для устранения этого эффекта было применено обучение с учетом квантования. После краткого дообучения 4-битная СОК-модель восстанавливает точность примерно до 42%, что показывает: значительная часть ошибки, вносимой реквантованием в области СОК, может быть компенсирована обучением.

6. ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА РАЗРЯДНОСТИ АККУМУЛЯТОРА И ВЫБОР НАБОРА МОДУЛЕЙ

Для экспериментов были выбраны две репрезентативные архитектуры нейронных сетей: ResNet-18 и UNet. ResNet-18 – широко используемая сверточная архитектура для задач классификации изображений, а также извлечения признаков в задачах компьютерного зрения в целом. UNet, в свою очередь, является популярной архитектурой для семантической сегментации и биомедицинской визуализации. Кроме того, после квантования эти сети становятся достаточно компактными для развертывания на ЦП, при этом сохраняя достаточно высокое качество решения реальных задач.

Эти модели были обучены на стандартных наборах данных: CIFAR-10 [33] и ImageNet [34] для классификации изображений, а также на данных магнитно-резонансной томографии головного мозга из выборки TCSIA [35] для сегментации изображений. Была выполнена оценка требуемой разрядности аккумулятора, обеспечивающей вычисления без переполнения, как теоретически (на основе размеров слоев и максимальных/минимальных квантованных значений), так и экспериментально. Результаты представлены в табл. 1 и 2.

Экспериментальные результаты показывают, что фактическое использование аккумулятора ниже теоретических максимумов из-за разреженности активаций, что позволяет дополни-

Таблица 3. Наборы модулей СОК для различных разрядностей аккумулятора.

Размер аккумулятора	t	Набор модулей
14	5	{31, 32, 33}
20/21	7	{127, 128, 129}
29	10	{1023, 1024, 1025}

тельно оптимизировать аппаратные ресурсы. Получающиеся наборы модулей, требуемые для каждого значения t , приведены в табл. 3. Видно, что аппаратная сложность стандартной реализации с точки зрения разрядности аккумулятора немного ниже, чем у СОК-модели, где суммарная разрядность для всего набора модулей составляет примерно $3t$.

Однако, использование усредняющего накопления позволяет дополнительно ограничить требуемый параметр t , поскольку промежуточные значения остаются в фиксированной разрядности и не требуют расширенного аккумулятора. Цена этого подхода – необходимость выполнять операцию усреднения, то есть вычислять $\lfloor (a + b)/2 \rfloor$, что для СОК нетривиально.

Тем не менее, при малых t эту операцию можно реализовать таблично. Например, при $t = 3$ динамический диапазон равен $M = 7 \cdot 8 \cdot 9 = 504$. Тогда таблица для операции усреднения по неупорядоченным парам $\{a, b\}$ имеет $\frac{M(M+1)}{2} = 127260$ записей. Несмотря на дополнительные затраты памяти, уменьшение t упрощает арифметико-логические блоки и сокращает критический путь.

Однако для точной оценки сложности необходимо смоделировать или реализовать предложенный СОК-конвейер квантованного исполнения на ПЛИС. Такая реализация позволит измерить энергопотребление в реальных условиях, площадь ПЛИС и определить практическую применимость ускорения на базе СОК для конечных и встраиваемых устройств.

7. ЗАКЛЮЧЕНИЕ

В данной работе представлен подход, позволяющий вычислять малобитные равномерно квантованные нейронные сети полностью в системе остаточных классов.

Для 4 и 8-битных квантованных нейронных сетей была построена модель исполнения в системе остаточных классов для всех основных слоев, включая свертку, активацию и реквантование. Было экспериментально показано, что вычисление нейронных сетей в СОК численно осуществимо, особенно при 8-битной точности, и что дообучение способно компенсировать значительную часть ошибки аппроксимации, вносимой реквантованием в СОК при меньших разрядностях.

Хотя в данной работе не приведена полноценная реализация предложенной СОК-модели на ПЛИС, существующие СОК-ускорители задают полезную нижнюю оценку ожидаемых аппаратных преимуществ. В работе [11] сообщается примерно о 36% экономии полезной площади для таблиц и снижении потребления энергии на 21–23%, а также о росте пропускной способности до 2.8 раз и повышении энергоэффективности до 2.7 раз в случае, если сэкономленная площадь реинвестируется в параллелизм. В другой работе показано, что СОК-умножитель для весов СНС может быть примерно на 64.7% быстрее и использовать на 43.5% меньше логических элементов, чем 8-битный бинарный умножитель [36]. В совокупности эти результаты указывают на то, что ПЛИС-реализация предложенной модели в СОК, вероятно, сможет обеспечить 30–40% сокращение площади и 20–30% снижение энергопотребления при сохранении производительности. Альтернативно, если высвобожденные ресурсы использовать для увеличения параллелизма, схема может продемонстрировать улучшение как пропускной способности, так и энергоэффективности в 2–3 раза.

Дополнительно отметим, что снижение требований к разрядности аккумулятора напрямую усиливает преимущества СОК: меньший динамический диапазон позволяет использовать бо-

лее компактные наборы модулей, уменьшая сложность модульных блоков и критический путь. В этом контексте перспективным направлением является усредняющее (древовидное) накопление в области СОК, позволяющее удерживать промежуточные значения в фиксированной разрядности.

В целом предложенный подход переносит стандартное равномерное квантование в модель вычислений, использующую систему остаточных классов и намечает путь к ускорителям нейронных сетей следующего поколения, ориентированным на компактность, скорость и энергоэффективность.

СПИСОК ЛИТЕРАТУРЫ

1. I. D. Mienye, T. G. Swart, G. Obaido, M. Jordan, and P. Ilono, “Deep convolutional neural networks in medical image analysis: A review,” *Information* **16**(3) (2025).
2. X. Zhu, L. Wang, C. Zhou, X. Cao, Y. Gong, and L. Chen, “A survey on deep learning approaches for data integration in autonomous driving system,” (2023).
3. E. Andreeva, V. Arlazarov, A. Gayer, E. Dorokhov, A. Sheshkus, and O. Slavin, “Document recognition method based on convolutional neural network invariant to 180 degree rotation angle,” *ITiVS* (4), 87–93 (2019).
4. A. V. Gayer, “Context-independent fast text detection method for recognizing phone numbers,” *Trudy ISA RAN (Proceedings of ISA RAS)* **74**(3), 39–47 (2024).
5. R. Krishnamoorthi, “Quantizing deep convolutional networks for efficient inference: A whitepaper,” (2018).
6. M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, “A white paper on neural network quantization,” (2021).
7. M. Valueva, N. Nagornov, P. Lyakhov, G. Valuev, and N. Chervyakov, “Application of the residue number system to reduce hardware costs of the convolutional neural network implementation,” *Mathematics and Computers in Simulation* **177**, 232–243 (2020).
8. G. Martinelli and R. Perfetti, “Rns neural networks,” 2955–2958 vol.4 (01 1990).
9. C. N. Zhang, M. Wang, and C. C. Tseng, “Residue systolic implementations for neural networks,” *Neural Computing & Applications* **3**, 149–156 (Sep 1995).
10. H. Nakahara and T. Sasao, “A deep convolutional neural network based on nested residue number system,” in 2015 25th International Conference on Field Programmable Logic and Applications (FPL), 1–6 (2015).
11. S. Salamat, M. Imani, S. Gupta, and T. Rosing, “Rnsnet: In-memory neural network acceleration using residue number system,” in 2018 IEEE International Conference on Rebooting Computing (ICRC), 1–12 (2018).
12. N. Samimi, M. Kamal, A. Afzali-Kusha, and M. Pedram, “Res-dnn: A residue number system-based dnn accelerator unit,” *IEEE Transactions on Circuits and Systems I: Regular Papers* **67**(2), 658–671 (2020).
13. A. Roohi, M. Taheri, S. Angizi, and D. Fan, “Rnsim: Efficient deep neural network accelerator using residue number systems,” in 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), 1–9 (2021).
14. V. Sakellariou, V. Paliouras, I. Kouretas, H. Saleh, and T. Stouraitis, “A multiplier-free rns-based cnn accelerator exploiting bit-level sparsity,” *IEEE Transactions on Emerging Topics in Computing* **12**(2), 667–683 (2024).
15. B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” (2017).

16. Y. Choukroun, E. Kravchik, F. Yang, and P. Kisilev, "Low-bit quantization of neural networks for efficient inference," in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 3009–3018 (2019).
17. A. Trusov, E. Limonova, D. Nikolaev, and V. V. Arlazarov, "4.6-bit quantization for fast and accurate neural network inference on cpus," *Mathematics* **12**(5) (2024).
18. K. Liu, Q. Zheng, K. Tao, Z. Li, H. Qin, W. Li, Y. Guo, X. Liu, L. Kong, G. Chen, Y. Zhang, and X. Yang, "Low-bit model quantization for deep neural networks: A survey," (2025).
19. Z.-G. Liu and M. Mattina, "Efficient residue number system based winograd convolution," (2020).
20. W. Chen, H. Qiu, J. Zhuang, C. Zhang, Y. Hu, Q. Lu, T. Wang, Y. Shi, M. Huang, and X. Xu, "Quantization of deep neural networks for accurate edge computing," (2021).
21. D. M. Ershova, A. V. Gayer, P. V. Bezmaternykh, and V. V. Arlazarov, "Yolo-barcode: towards universal real-time barcode detection on mobile devices," *Computer Optics* **48**(4), 592–600 (2024).
22. A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," (2021).
23. I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," (2016).
24. K. Zhao, S. Huang, P. Pan, Y. Li, Y. Zhang, Z. Gu, and Y. Xu, "Distribution adaptive int8 quantization for training cnns," (2021).
25. D. Bablani, J. L. McKinstry, S. K. Esser, R. Appuswamy, and D. S. Modha, "Efficient and effective methods for mixed precision neural network quantization for faster, energy-efficient inference," (2024).
26. A. V. Trusov, "Training 4.6-bit convolutional neural networks with a hardtanh activation function," *Pattern Recognition and Image Analysis* **35**, 44–64 (Mar 2025).
27. K. S. Kiyamova, E. E. Limonova, M. V. Zingerenko, D. P. Nikolaev, and V. V. Arlazarov, "Fast approximate matrix multiplication for 8-bit neural networks using tree averaging," in ICMV 2025 (IN PRINT), 1–8, Society of Photo-Optical Instrumentation Engineers (SPIE), Bellingham, Washington 98227-0010 USA (2025).
28. G. Alsuhli, V. Sakellariou, H. Saleh, M. Al-Qutayri, B. Mohammad, and T. Stouraitis, "Number systems for deep neural network architectures: A survey," (2023).
29. V. Kuchukov, D. Telpukhov, M. Babenko, I. Mkrchan, A. Stempkovsky, N. Kucherov, T. Ermakova, and M. Grigoryan, "Performance analysis of hardware implementations of reverse conversion from the residue number system," *Applied Sciences* **12**(23) (2022).
30. A. Omundi and B. Premkumar, *Residue Number System: Theory and Implementation* (01 2007).
31. M. Hitz and E. Kaltofen, "Integer division in residue number systems," *IEEE Transactions on Computers* **44**(8), 983–989 (1995).
32. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* **86**(11), 2278–2324 (1998).
33. A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto (05 2012).
34. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)* **115**(3), 211–252 (2015).
35. K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, "The cancer imaging archive (tcia): Maintaining and operating a public information repository," (7 2013).
36. S. Sivkov, "Residue number systems quantization for deep learning inference," *WSEAS TRANSACTIONS ON COMPUTERS* **22**, 296–301 (12 2023).

Low-bit Quantized Neural Networks in the Residue Number System with Classical or Tree-Averaging Accumulation

M. V. Zingerenko

Low-bit affine quantized neural networks are widely used to solve computer vision tasks on resource-constrained devices, preserving high accuracy with a simple CPU-friendly hardware realization. In this work, we propose a model for executing uniformly quantized neural networks in the residue number system (RNS), which enables parallel, hardware-efficient, carry-free arithmetic and is well suited for FPGA and ASIC implementations. We present RNS-domain models for the main network components, including linear and convolutional layers, piecewise-linear activation functions, and requantization. We also estimate the required accumulator bit-widths and select moduli sets that guarantee overflow-free computation. Finally, to further improve the hardware efficiency of linear layers, we propose tree-averaging accumulation, which reduces the required accumulator width.

KEYWORDS: affine lowbit quantization, convolutional neural networks, residue number system, energy efficiency.