DOI: 10.15514/ISPRAS-2022-34(3)-5

Метод аппаратной реализации сверточной нейронной сети на основе системы остаточных классов

[1] М.В. Валуева, ORCID: 0000-0002-4732-3216 <mriya.valueva@mail.ru>
[1] Г.В. Валуев, ORCID: 0000-0003-2049-7213 <mail@gvvaluev.ru>
[2,3] М.Г. Бабенко, ORCID: 0000-0001-7066-0061 <mgbabenko@ncfu.ru>
[3,4,5] А. Черных, ORCID: 0000-0001-5029-5212 <chernykh@cicese.mx>
[5] Х.М. Кортес Мендоса, ORCID: 0000-0001-7209-8324 <kortesmendosak@susu.ru>
[1] Северо-Кавказский центр математических исследований СКФУ
355017, Россия, г. Ставрополь, ул. Пушкина, 1
[2] Северо-Кавказский федеральный университет,
355017, Россия, г. Ставрополь, ул. Пушкина, 1
[3] Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25
[4] Центр научных исследований и высшего образования,
Мексика, 22860, Нижняя Калифорния, Энсенада, ш. Тихуана-Энсенада, 3918
[5] Южно-Уральский государственный университет,
454080, Россия, Челябинск, проспект Ленина, 76

Аннотация. Сверточные нейронные сети (СНС) показывают высокую точность при решении задачи распознавания образов, но обладают высокой вычислительной сложностью, что приводит к медленной обработке данных. Для увеличения быстродействия СНС в данной работе предлагается метод аппаратной реализации СНС с вычислениями в системе остаточных классов с модулями специального вида 2^{α} и $2^{\alpha}-1$. В статье представлено аппаратное моделирование предлагаемого метода на FPGA на примере СНС LeNet-5, обученной на базах изображений MNIST, FMNIST и CIFAR-10. Моделирование показало, что применение предлагаемого подхода позволяет увеличить тактовую частоту и производительность устройства примерно на 11% –12%, по сравнению с традиционным подходом на основе позиционной системы счисления. Тем не менее, увеличение скорости работы устройства достигнуто за счет увеличения аппаратных затрат. Предлагаемый в статье метод может быть применен системах распознавания образов, когда необходимо обеспечить высокую скорость обработки данных.

Ключевые слова: сверточная нейронная сеть; система остаточных классов; распознавание образов; field-programmable gate array (FPGA)

Для цитирования: Валуева М.В., Валуев Г.В., Бабенко М.Г., Черных А., Кортес Мендоса Х.М. Метод аппаратной реализации сверточной нейронной сети на основе системы остаточных классов. Труды ИСП РАН, том 34, вып. 3, 2022 г., стр. 61-74. DOI: 10.15514/ISPRAS–2022–34(3)–5.

Благодарности: Работа выполнена при поддержке Российского научного фонда, проект №19-71-10033.

Method for Convolutional Neural Network Hardware Implementation Based on a Residue Number System

[1] M.V. Valueva, ORCID: 0000-0002-4732-3216 <mriya.valueva@mail.ru>
[1] G.V. Valuev, ORCID: 0000-0003-2049-7213 <mail@gvvaluev.ru>
[2,3] M.G. Babenko, ORCID: 0000-0001-7066-0061 <mgbabenko@ncfu.ru>
[3,4,5] A. Tchernykh, ORCID: 0000-0001-5029-5212 <chernykh@cicese.mx>
[5] J. M. Cortes-Mendoza, ORCID: 0000-0001-7209-8324 <kortesmendosak@susu.ru>
[1] North-Caucasus Center for Mathematical Research NCFU
1, Pushkin St., Stavropol, 355017, Russia
[2] North-Caucasus Federal University,
1, Pushkin st., Stavropol, 355017, Russia
[3] Ivannikov Institute for System Programming of the Russian Academy of Sciences
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia
[4] CICESE Research Center,
3918, Ensenada-Tijuana Highway, Ensenada, 22860, Mexico
[5] South Ural State University
76, Lenin prospekt, Chelyabinsk, 454080, Russia

Abstract. Convolutional Neural Networks (CNN) show high accuracy in pattern recognition solving problem but have high computational complexity, which leads to slow data processing. To increase the speed of CNN, we propose a hardware implementation method with calculations in the residue number system with moduli of a special type 2^{α} and $2^{\alpha}-1$. A hardware simulation of the proposed method on Field-Programmable Gate Array for LeNet-5 CNN is trained with the MNIST, FMNIST, and CIFAR-10 image databases. It has shown that the proposed approach can increase the clock frequency and performance of the device by 11%-12%, compared with the traditional approach based on the positional number system.

Keywords: convolutional neural network, residue number system, pattern recognition, field-programmable gate array (FPGA)

For citation: Valueva M.V., Valuev G.V., Babenko M.G., Tchernykh A., Cortes-Mendoza J. M. Method for Convolutional Neural Network Hardware Implementation Based on a Residue Number System. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 3, 2022, pp. 61-74 (in Russian). DOI: 10.15514/ISPRAS–2022–34(3)–5

Acknowledgements. This work was supported in part by the Russian Science Foundation, project number 19-71-10033.

1. Введение

Одним из основных методов распознавания визуальных образов являются сверточные нейронные сети (СНС), поскольку они показывают высокую точность. Они широко используются в области медицины [1-3], в системах видеонаблюдения [4, 5], в робототехнике [6] и других областях. Но высокая точность распознавания достигается за счет увеличения количества слоев сети, а, следовательно, и ее вычислительной сложности [7]. Это приводит к необходимости разработки специализированных аппаратных ускорителей нейросетевых вычислений.

Одним из эффективных путей улучшения технических характеристик цифровых устройств является оптимизация вычислений на арифметическом уровне. Например, авторы работ [8-10] предлагают выполнять вычисления в системе остаточных классов (СОК) для построения эффективной аппаратной реализации глубоких нейронных сетей.

В данной работе предлагается метод аппаратной реализации СНС с вычислениями в СОК с модулями вида 2^α и 2^α-1. Так же в работе представлена реализация предлагаемого метода на программируемой пользователем вентильной матрице (field-programmable gate array,

FPGA) и произведено сравнение с реализаций СНС в традиционной позиционной системе счисления (ПСС).

Оставшаяся часть статьи организована следующим образом. В разд. 2 описан предлагаемый метод применения СОК для аппаратной реализации СНС. В разд. 3 представлены результаты аппаратного моделирования предлагаемого метода на примере СНС LeNet-5. Был произведен анализ результатов моделирования, которые представлены в разд. 4. Выводы по проведенному исследования находятся в разд. 5.

2. Предлагаемый метод

Любое целое число $0 \le A < M$ может быть однозначно представлено в СОК в виде остатков от деления на попарно взаимно простые модули $\{m_1, m_2, \dots, m_n\}$ как $A = \{a_1, a_2, \dots, a_n\}$, так что $a_i = |A|_{m_i}$ [11], при этом $M = \prod_{i=1}^{n} m_i$ называется динамическим диапазоном СОК. Для представления отрицательных чисел в СОК, динамический диапазон системы делится на две примерно равные части, при этом должно выполняться одно из следующих условий:

$$\begin{aligned} -\tfrac{M-1}{2} \le A \le \tfrac{M-1}{2} \text{ для нечетных } M, \\ -\tfrac{M}{2} \le A \le \tfrac{M}{2} - 1 \text{ для четных } M, \end{aligned} \quad (1)$$

Арифметические операции над числами, представленными в СОК, производятся параллельно по каждому модулю:

$$A * B = (|a_1 * b_1|_{p_i}, |a_2 * b_2|_{p_2}, \dots, |a_n * b_n|_{p_n}), \quad (2)$$

где $*$ означает операцию сложения, вычитания или умножения.

Число $A = \{a_1, a_2, \dots, a_n\}$ может быть преобразовано из СОК в позиционную систему счисления (ПСС) с использованием Китайской теоремы об остатках [11]

$$A = \left| \sum_{i=1}^{j} \left(\left| |M_i^{-1}|_{m_i} a_i \right| M_i \right) \right|_M, \quad (3)$$

где $M_i = \frac{M}{m_i}$, а $|M_i^{-1}|_{m_i}$ – мультипликативный обратный элемент для M_i.

Первым блоком, при выполнении вычислений в СОК, является перевод чисел из ПСС в СОК. Для преобразования числа необходимо вычислить остатки от деления по каждому модулю. Данная операция является ресурсозатратной. Использование модулей вида $\{2^{\alpha_1}, 2^{\alpha_2} - 1, \dots, 2^{\alpha_n} - 1\}$ позволяет избежать данной операции. Операция вычисления остатка от деления по модулю вида 2^α осуществляется простым оставлением α младших бит исходного числа, с отбрасыванием оставшихся старших значащих бит. Введем для устройства, выполняющего вычисление остатка от деления по модулю 2^α обозначение MOD_{2^α}. Для модулей вида $2^\alpha - 1$ вычисление остатка от деления α-битных чисел по модулю $2^\alpha - 1$ [12], обозначим данное устройство как $MOD_{2^\alpha - 1}$. При сложении по модулю $2^\alpha - 1$ больше двух слагаемых используют устройство для сложения нескольких чисел по модулю $MOMA_{2^\alpha - 1}$, которое состоит из дерева сумматоров с сохранением переноса и циклическим переносом старшего бита (EAC-CSA) [13, 14], а результат передается на сумматор Когге-Стоуна с циклическим переносом старшего бита (EAC-KSA) [14, 15]. Таким образом, устройство $PNS \to RNS$ для перевода числа DR-битного числа A в СОК с модулями вида $\{2^{\alpha_1}, 2^{\alpha_2} - 1, \dots, 2^{\alpha_n} - 1\}$ состоит из одного устройства MOD_{2^α} и $n - 1$ устройств $MOD_{2^\alpha - 1}$. На выходе устройства формируется число $\{a_1, a_2, \dots, a_n\}$, представленное в СОК и имеющее разрядности $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ соответственно. Далее выполняются вычисления в СОК параллельно по каждому вычислительному каналу, соответствующему модулю системы.

Все вычисления в предлагаемой архитектуре СНС производятся параллельно с использованием одного вычислительного канала по модулю 2^α и $n - 1$ вычислительных каналов по модулю вида $2^\alpha - 1$.

Предположим, что на вход сверточного слоя поступает набор из D карт признаков I_{in} состоящих из R строк, C столбцов. Это означает, что вход сверточного слоя можно описать как трехмерную функцию $I(x, y, z)$, где $0 \le x < R$, $0 \le y < C$ и $0 \le z < D$ пространственные координаты, а амплитуда I в любой точке с координатами (x, y, z) это интенсивность пикселей в этой точке. Процедура получения одной карты признаков в сверточном слое по модулю m_l может быть представлена в виде:

$$\left| I_f(x,y) \right|_{m_l} = \left| |b|_{m_l} + \left| \sum_{i=-t}^{t} \sum_{j=-t}^{t} \sum_{z=0}^{D-1} \left| W_{i,j,z} \right|_{m_l} \cdot |I(x+i, y+j, z)|_{m_l} \right|_{m_l} \right|_{m_l}, \quad (4)$$

где I_f – карта признаков после свертки, $W_{i,j,z}$ – это коэффициенты 3D-фильтра размерности $d \times d$ для обработки D двухмерных массивов, $t = \left\lfloor \frac{d}{2} \right\rfloor$ и b – смещение [16].

Пусть $F = \{F_0, F_1 \dots, F_{D-1}\}$ набор из D векторов размерности d^2, которые соответствуют фрагментам размерности $d \times d$ карт признаков I, поступающих на вход сверточного слоя. Аналогично, представим маску фильтра как набор из D векторов размерности d^2 и обозначим как $W = \{W_0, W_1 \dots, W_{D-1}\}$. Тогда операция свертки по модулю m_l для вычисления одного значения R карты признаков I_f может быть представлена в виде суммы одномерных сверток и прибавлении смещения

$$|R|_{m_l} = \left| |b|_{m_l} + \left| \sum_{i=0}^{D-1} \sum_{j=0}^{d^2-1} \left| W_{i,j} \right|_{m_l} \cdot \left| F_{i,j} \right|_{m_l} \right|_{m_l} \right|_{m_l}, \quad (5)$$

Одномерная свертка может быть реализована с помощью фильтра с конечной импульсной характеристикой (FIR) [17], который состоит из блоков умножения с накоплением (MAC). Блоки MAC состоят из генератора частичных произведений PPG [13], содержащего вентили AND, и дерева сумматоров. Результаты одномерной свертки суммируются с помощью сумматора со множественным входом. Если расчеты производятся по модулю 2^α, то биты старше α не участвуют в вычислениях. Для выполнения вычислений по модулю $2^\alpha - 1$ используется техника циклического переноса старших бит EAC. На рис. 1 представлена схема устройства $CONV_{2^\alpha}$ для свертки фрагмента изображения по модулю 2^α. Устройство свертки $CONV_{2^\alpha - 1}$ по модулю $2^\alpha - 1$ имеет аналогичную структуру.

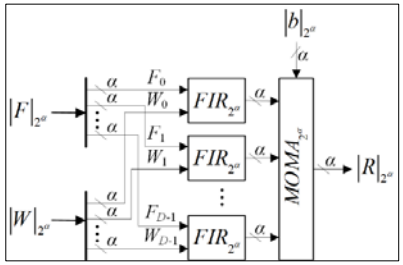

Рис. 1. Схема устройства $CONV_{2^\alpha}$ для свертки по модулю 2^α фрагмента карты признаков
Fig. 1. The $CONV_{2^\alpha}$ device circuit for modulo 2^α convolution of feature map fragment

На выходе сверточного слоя применяется функция активации. Наиболее распространенной функцией является линейный выпрямитель (ReLU) [18], которая имеет вид $max(0, R)$ и

сводится к определению знака числа. Количество карт признаков на выходе сверточного слоя соответствует количеству масок фильтров.

Операция определения знака числа является ресурсозатратной в СОК, так как требует вычисления позиционной характеристики числа [19]. Вычисление позиционной характеристики числа по формуле (3) требует вычисления остатка от деления на число M с разрядностью полного диапазона системы DR. На практике, одним из самых эффективных подходов является модификация КТО, называемая КТО с дробными величинами (КТОд) [20].

Согласно КТОд позиционную характеристику числа A' можно вычислить по формуле:

$$A' = \left| \sum_{i=1}^{n} a_i \tilde{k}_i \right|_{2^N}, \quad (6)$$

где $\tilde{k}_i = \left\lfloor 2^N \frac{\left|P_i^{-1}\right|_{p_i} P_i}{P} \right\rfloor$. Для гарантированного точного перевода чисел из СОК в ПСС достаточно выбрать N равным:

$$N = \lceil \log_2(P\mu) \rceil - 1, \quad (7)$$

где $\mu = -n + \sum_{i=1}^{n} p_i$.

Для определения знака числа в СОК по его позиционной характеристике A' необходимо выполнить проверку следующих условий:

- если $0 \le A' < 2^{N-1}$, тогда число A положительное;
- если $2^{N-1} < A' < 2^N$, тогда число A отрицательное.

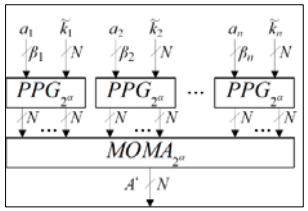

Рис. 2. Архитектура устройства NPC_{CRTf} для вычисления позиционной характеристики числа, представленного в СОК, с помощью КТОд
Fig. 2. Architecture of the NPC_{CRTf} device for calculating the positional characteristic of a number represented in RNS using CRTf

На рис. 2 представлена схема устройства, которое вычисляет позиционную характеристику числа по формуле (6) с помощью метода КТОд. Введем для него обозначение NPC_{CRTf}. На вход устройства поступает число $\{a_1, a_2, \dots, a_n\}$, представленное в СОК с модулями $\{m_1, m_2, \dots, m_n\}$ и разрядностями $\{\beta_1, \beta_2, \dots, \beta_n\}$ по каждому модулю соответственно. Также на вход подаются коэффициенты $\tilde{k}_i$, имеющие разрядность N бит, они являются константами и могут быть вычислены предварительно. Генерация частичных произведений осуществляется с помощью устройств PPG_{2^α}. Далее N-битные частичные произведения складываются с помощью устройства $MOMA_{2^\alpha}$. Устройства PPG_{2^α} и $MOMA_{2^\alpha}$ являются N-битными, то есть $\alpha = N$. Если старший значащий бит (Most Significant Bit, MSB) числа A' равен 0, то число A отрицательное, если равен 1, то положительное.

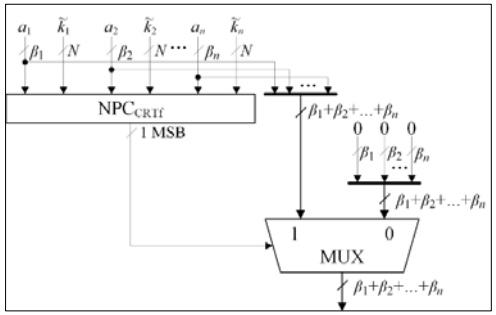

Рис. 3. Архитектура устройства ReLU для вычисления функции активации ReLU
Fig. 3. ReLU device architecture for computing the ReLU activation function

На рис. 3 представлена архитектура устройства ReLU для вычисления функции активации ReLU в СОК. Здесь знак числа определяется с помощью старшего значащего бита позиционной характеристики числа, который передается в качестве управляющего сигнала на вход мультиплексора MUX, принимающего решение какое значение подавать на выход устройства.

НС обычно использует большое количество фильтров в сверточном слое. Это приводит к резкому увеличению объема обрабатываемых данных в сети. Для их уменьшения используется слой выборки (pooling). Чаще всего используется выборка максимальных элементов в некоторой рассматриваемой окрестности (max pooling). На вход слоя поступает массив карт признаков, состоящий из D карт, содержащих R строк и C столбцов. Следовательно, вход данного слоя можно описать как трехмерную функцию $P_{in}(x_{in}, y_{in}, z)$, где $0 \le x_{in} < R$, $0 \le y_{in} < C$ и $0 \le z < D$ – пространственные координаты, а амплитуда P_{in} в любой точке с координатами (x_{in}, y_{in}, z) – интенсивность пикселей в данной точке. Процедуру максимального элемента из окрестности размером $s \times s$ с шагом s для z-ой карты признаков можно описать формулой:

$$P_{out}(x_{out}, y_{out}, z) = \max_{\substack{s \cdot x_{out} \le x_{in} \le s \cdot (x_{out}+1), \\ s \cdot y_{out} \le y_{in} \le s \cdot (y_{out}+1)}} \{P_{in}(x_{in}, y_{in}, z)\} \quad (8)$$

где $P_{out}(x_{out}, y_{out}, z)$ – набор из D карт признаков на выходе сверточного слоя, $0 \le x_{out} < \frac{R}{s} - 1$, $0 \le y_{out} < \frac{C}{s} - 1$.

Операция сравнения двух чисел в СОК сводится к сравнению их позиционных характеристик. Таким образом устройство для сравнения чисел в СОК по методу КТОд состоит из двух устройств NPC_{CRTf}, вычисляющих позиционную характеристику числа, и устройства COMP, выполняющего сравнение двух чисел в ПСС. На рис. 4 представлено устройство MAX, которое выполняет выбор большего числа из двух $A = \{a_1, a_2, \dots, a_n\}$ и $B = \{b_1, b_2, \dots, b_n\}$, представленных в СОК. Выход устройства COMP, выполняющего сравнение позиционных характеристик, подается в качестве управляющего сигнала на вход мультиплексора MUX, который принимает решение какое из двух чисел в СОК передать на выход устройства.

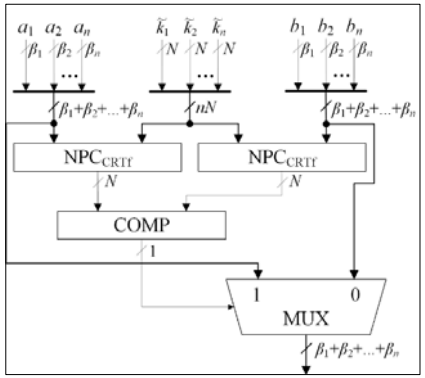

Рис. 4. Архитектура устройства MAX для выбора большего числа из двух в СОК
Fig. 4. MAX device architecture for choosing largest out of two numbers in RNS

Наиболее часто на слое max pooling используется $s = 2$. То есть рассматривается окрестность размерности 2×2, и вычисления производятся с шагом 2. Так как слой max pooling всегда идет после сверточного слоя, то позиционные характеристики чисел были рассчитаны при вычислении функции активации ReLU и поступают на вход устройства. Выбор наибольшего числа производится с помощью дерева устройств MAX.

Заключительными слоями сети являются полносвязные слои нейронов, выполняющие функцию классификатора **Ошибка! Источник ссылки не найден.**. Пусть $X = \{x_i\}$, – вектор, подаваемый на вход $(p-1)$-го слоя, где $0 \le i < m$ и m– общее число входов. Каждый элемент вектора умножается на соответствующий весовой коэффициент $W_j = \{w_{ji}\}$, $0 \le i < m$, $0 \le j < n$, n– количество нейронов $(p-1)$-го слоя и результат суммируется

$$y_j = b_j + \sum_{i=0}^{m-1} w_{ji} x_i. \qquad (9)$$

Обозначим функцию активации для $(p-1)$-го слоя как $\phi(t)$, тогда результатом $(p-1)$ -го слоя является вектор $\{h_j\}$, $0 \le j < n$, элементы которого вычисляются как $h_j = \phi(y_j)$.

Результат вычислений $(p-1)$-го слоя подается на вход p-го слоя, производим процедуру умножения на соответствующие весовые коэффициенты, складываем и подаем на функцию активации, результатом вычислений является вектор $\{z_k\}$, $0 \le k < l$, l– количество нейронов p-го слоя. Результат последнего слоя нормализуется с помощью функции softmax, таким образом на выходе полносвязных слоев формируется вектор $\{g_k\}$, $0 \le k < l$, элементы которого вычисляются следующим образом $g_k = \frac{e^{z_k}}{\sum_{i=0}^{l-1} e^{z_i}}$. На выходе полносвязного слоя формируется вектор, количество элементов которого соответствует количеству классов и отображает вероятность принадлежности образа, подаваемого на вход сети, к каждому классу.

На рис. 5 представлена архитектура устройства FC_{2^α}, выполняющего вычисления по формуле (9) для j-го нейрона по модулю 2^α. Устройство $FC_{2^\alpha-1}$, выполняющее вычисления для j-го нейрона по модулю $2^\alpha - 1$ имеет аналогичную архитектуру, но использует технику EAC. Результаты работы устройств FC_{2^α} и $FC_{2^\alpha-1}$ подаются на вход устройства ReLU.

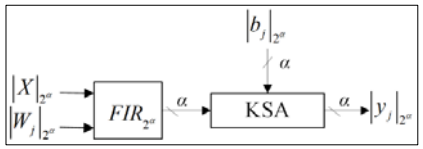

Рис. 5. Архитектура устройства FC_{2^α} полносвязного слоя с вычислениями по модулю 2^α
Fig. 5. Device architecture FC_{2^α} for calculating the output of one neuron of a fully connected layer with calculations modulo 2^α

Последним блоком системы, выполняющей вычисления в СОК, является преобразование результата обратно в ПСС. Для перевода числа из СОК в ПСС, согласно КТОд, необходимо умножить позиционную характеристику A' на модуль M. При этом результатом алгоритма являются старшие биты начиная с бита под номером $N+1$. Таким образом:

$$A = \frac{A'M}{2^N} \qquad (10)$$

В (10) операция деления при аппаратной реализации игнорируется, так как на выход подаются старшие значащие биты начиная с $(N+1)$-го. В программной реализации эта операция эквивалентна сдвигу на N разрядов вправо. Тогда в устройстве $RNS \to PNS$ для обратного преобразования чисел из СОК в ПСС позиционная характеристика числа вычисляется с помощью устройства NPC_{CRTf}, а умножение на динамический диапазон системы производится с помощью $(N+DR)$-разрядного генератора частичных произведений PPG_{2^α} и сумматора $MOMA_{2^\alpha}$ такой же разрядности.

На рис. 6 представлена архитектура СНС с вычислениями в СОК. На вход поступает изображение в виде последовательности пикселей. Сперва в блоке $PNS \to RNS$ производится преобразование данных в СОК. Затем они поступают в оперативное запоминающее устройство (ОЗУ) и передаются в другие блоки СНС (сверточные слои, слои max pooling и полносвязные слои). Весовые коэффициенты СНС хранятся в постоянном запоминающем устройстве. Перевод весовых коэффициентов в СОК производится с помощью блоков $PNS \to RNS$, затем данные в формате СОК поступают на сверточные и полносвязные слои. Результат работы СНС переводится в ПСС с помощью блока $RNS \to PNS$ и поступает на выход устройства.

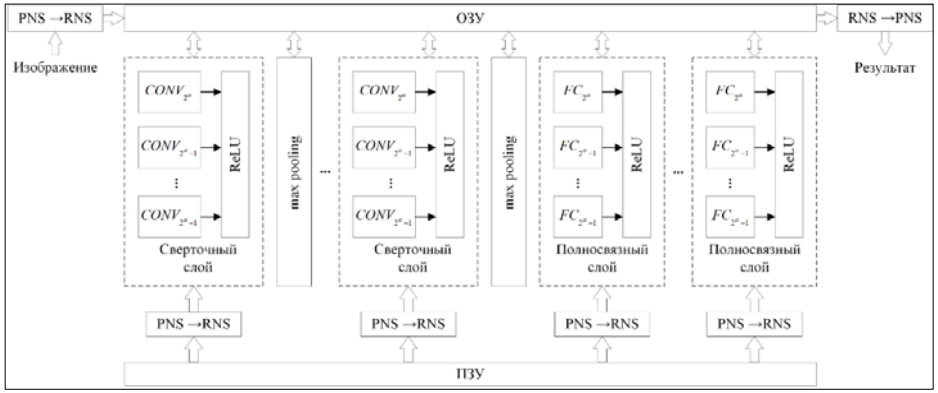

Рис. 6. Предлагаемая архитектура СНС с вычислениями в СОК
Fig. 6. Proposed CNN architecture with computations in RNS

Представленный метод для аппаратной реализации СНС с вычислениями в СОК позволяет выполнять вычисления в сверточных и полносвязных слоях параллельно с числами меньшей размерности, чем в ПСС, что увеличивает быстродействие системы. В следующем разделе представлен пример применения на практике предложенного метода.

3. Эксперимент и результаты

Рассмотрим применение предлагаемого метода аппаратной реализации СНС с использованием СОК на примере архитектуры LeNet-5 [22]. Обучение производилось с помощью библиотек машинного обучения TensorFlow [23] и Keras [24], использовался язык программирования Python. Функция активации гиперболический тангенс (tanh) заменена на ReLU. Параметры архитектуры LeNet-5 представлены в табл. 1.

Табл. 1. Параметры архитектур СНС LeNet-5
Table 1. Parameters of CNN LeNet-5 architecture

Слой	**Размер маски фильтра**	**Количество фильтров**	**Функция активации**
свертка	5×5×3	6	ReLU
max pooling	2×2	1	
свертка	5×5×6	16	ReLU
max pooling	2×2	1	
свертка	5×5×16	120	ReLU
полносвязный	84 нейрона		ReLU
полносвязный	10 нейронов		softmax

Для обучения СНС использовались базы изображений MNIST [22], FMNIST [25] и CIFAR-10 [26]. База MNIST содержит изображения рукописных цифр от 0 до 9 размера 28×28 в оттенках серого и состоит из 60000 изображений для обучения и 10000 изображений для тестирования. База FMNIST содержит 10 классов изображений одежды и обуви (футболка, брюки, свитер, платье, пальто, сандалии, рубашка, кроссовки, сумка, ботинки) размера 28×28 в оттенках серого, состоит из 60000 изображений для обучения и 10000 изображений для тестирования. База CIFAR-10 содержит 10 классов изображений (самолет, автомобиль, птица, кот, олень, собака, лягушка, лошадь, корабль, грузовик) размера 32×32 формата RGB, состоит из 50000 изображений для обучения и 10000 изображений для тестирования.

Весовые коэффициенты СНС являются вещественными числами. Для аппаратной реализации СНС требуется представить их в целочисленном виде. Для этого весовые коэффициенты необходимо умножить на 2^p и округлить результат к большему. После выполнения вычислений, полученный результат масштабируется на 2^{-p} и округляется к меньшему. Коэффициент масштабирования p показывает сколько бит необходимо для представления весовых коэффициентов в устройстве. От выбора p зависит точность работы СНС и объем памяти, необходимый для хранения весовых коэффициентов.

Был проведен эксперимент для выявления зависимости точности распознавания СНС от разрядности весовых коэффициентов, который показал, что разрядность весовых коэффициентов может быть уменьшена без потери точности распознавания. Для архитектуры СНС LeNet-5, обученной на базах MNIST и FMNIST, достаточно разрядности весовых коэффициентов 8 бит, а архитектуре, обученной на базе CIFAR-10, необходимо 12 бит. Таким образом, для архитектуры с 8-битным представлением весовых коэффициентов требуется в 4 раза меньший объем памяти, по сравнению с 32-битным представлением. А для архитектуры с 12-битным представлением весовых коэффициентов будет использоваться в 2,67 раз меньший, по сравнению с 32-битным представлением.

Табл. 2. Результаты аппаратного моделирования СНС LeNet-5
Table 2. Hardware simulation results of CNN LeNet-5

Параметр	**Набор данных**	**Система счисления**	
		ПСС	**СОК**
Тактовая частота, МГц	MNIST и FMNIST	50	56
	CIFAR-10	53	59
Количество LUT	MNIST и FMNIST	593291	647821
	CIFAR-10	612196	557297
Количество LUTRAM	MNIST и FMNIST	483	2212
	CIFAR-10	483	2212
Количество BRAM	MNIST и FMNIST	63,0	181,0
	CIFAR-10	69,5	200,5
Энергопотребление, Вт	MNIST и FMNIST	9,326	12,833
	CIFAR-10	11,718	13,518
Производительность, кадр/с	MNIST и FMNIST	272	305
	CIFAR-10	221	246

Было проведено сравнение предлагаемого метода аппаратной реализации СНС с вычислениями в СОК и традиционной архитектуры СНС в ПСС. Устройство с вычислениями в ПСС является 32-разрядным. Для организации вычислений в СОК был выбран набор модулей $\{2^{12}, 2^{11}-1, 2^{10}-1\}$. Аппаратное моделирование было проведено в среде Xilinx Vivado 2018.3 на целевой плате Virtex-7 xc7v2000tfhg1761-2L со стратегией оптимизации AreaOpimized_high. Результаты аппаратного моделирования представлены в табл. 2. Для оценки эффективности устройств были рассмотрены временные и аппаратные затраты устройств. К временным затратам относится тактовая частота устройства, измеряющаяся в МГц, и производительность, измеряющаяся как количество обработанных кадров в секунду (кадр/с). Для наборов данных MNIST и FMNIST размер кадра составляет 28×28 пикселей, а для набора данных CIFAR-10 размер кадра 32×32 пикселя. Под аппаратными затратами подразумевается количество занятых просмотровых таблиц (Look-Up-Table, LUT), памяти с произвольным доступом (Random Access Memory, RAM) LUTRAM и Block RAM (BRAM), а также энергопотребление устройства, которое измеряется в Вт.

Сравнение и обсуждение результатов аппаратного моделирования представлено в следующем разделе.

4. Обсуждение результатов

В табл. 2 представлены результаты аппаратного моделирования, которые показали, что использование предлагаемого метода на основе СОК позволяет увеличить тактовую частоту и производительность устройства примерно на 11% –12% по сравнению с ПСС. Тем не менее, предлагаемый метод требует больше аппаратных ресурсов. Энергопотребление устройства, разработанного по предлагаемому методу на 15% –38% выше, по сравнению с известным методом на основе ПСС. Кроме того, количество блоков памяти так же увеличилось примерно в 3–4 раза. Устройство на основе предлагаемого метода с 8-битным представлением весовых коэффициентов занимает на 9% больше LUT, но устройство с 12-битным представлением весовых коэффициентов использует на 9% меньше LUT по сравнению с методом на основе ПСС.

Основываясь на результатах аппаратного моделирования, можно сделать вывод, что предлагаемый метод аппаратной реализации СНС с вычислениями в СОК может быть

успешно применен в практических приложениях распознавания визуальных образов, где скорость обработки информации играет ключевую роль. Если требуется минимизировать аппаратные затраты, то традиционный метод на основе ПСС предпочтительнее.

Предложенный в работе метод, может быть адаптирован к другим архитектурам нейронных сетей, что расширяет область его практической значимости.

5. Заключение

В данной работе предложен метод аппаратной реализации СНС с вычислениями в СОК с модулями вида $\{2^{\alpha_1}, 2^{\alpha_2}-1, \ldots, 2^{\alpha_n}-1\}$. Проведено аппаратное моделирование на FPGA на примере СНС LeNet-5 и баз изображений MNIST, FMNIST и CIFAR-10. Результаты моделирования показали, что применение предлагаемого подхода к аппаратной реализации СНС позволяет увеличить тактовую частоту и производительность устройства примерно на 11% –12%, по сравнению с традиционным подходом на основе ПСС. Преимущество в скорости работы устройства достигнуто за счет увеличения аппаратных затрат. Предложенный метод может быть применен в таких практических приложениях как распознавание изображений, анализ речи и создание робототехнических систем.

Список литературы / References

[1] Ashiq F., Asif M. et al. CNN-Based Object Recognition and Tracking System to Assist Visually Impaired People. IEEE Access, vol. 10, 2022, pp. 14819-14834.

[2] Moon C.-I., Lee O. Skin Microstructure Segmentation and Aging Classification Using CNN-Based Models. IEEE Access, vol. 10, 2022, pp. 4948-4956.

[3] Mondal A.K., Bhattacharjee A. et al. xViTCOS: Explainable Vision Transformer Based COVID-19 Screening Using Radiography. IEEE Journal of Translational Engineering in Health and Medicine, vol. 10, 2022, article no. 1100110, pp. 1-10.

[4] Elharrouss O., Almaadeed N. et al. FSC-Set: Counting, Localization of Football Supporters Crowd in the Stadiums. IEEE Access, vol. 10, 2022, pp. 10445-10459.

[5] Vieira J.C., Sartori A. et al. Low-cost CNN for Automatic Violence Recognition on Embedded System. IEEE Access, vol. 10, 2022, pp. 25190-25202.

[6] Wong C.-C., Chien M.-Y. et al. Moving Object Prediction and Grasping System of Robot Manipulator. IEEE Access, vol. 10, 2022, pp. 20159-20172.

[7] Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. In Proc. of the 25th International Conference on Neural Information Processing Systems, vol. 1, 2012, pp, 1097–1105.

[8] Nakahara H., Sasao T. A deep convolutional neural network based on nested residue number system. In Proc. of the 25th International Conference on Field Programmable Logic and Applications (FPL), 2015, pp. 1-6.

[9] Nakahara H., Sasao T. A High-speed Low-power Deep Neural Network on an FPGA based on the Nested RNS: Applied to an Object Detector. In Proc. of IEEE International Symposium on Circuits and Systems (ISCAS), 2018, pp. 1-5.

[10] Salamat S., Imani M. et al. RNSnet: In-Memory Neural Network Acceleration Using Residue Number System. In Proc. of IEEE International Conference on Rebooting Computing (ICRC), 2018, pp. 1-12.

[11] Omondi A., Premkumar B. Residue Number Systems: Theory and Im-plementation. London, Imperial College Press, 2007. 296 p.

[12] Chervyakov N.I., Lyakhov P.A. et al. Residue Number System-Based Solution for Reducing the Hardware Cost of a Convolutional Neural Network. Neurocomputing, vol. 407, 2020, pp. 439-453.

[13] Parhami B. Computer arithmetic: algorithms and hard-ware designs. Oxford University Press, 2010. 492 p.

[14] Vergos H.T., Dimitrakopoulos G. On Modulo 2^n+1 Adder Design. IEEE Transactions on Computers, vol. 61, issue 2, 2012, pp. 173-186.

[15] Kogge P.M., Stone H.S. A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations. IEEE Transactions on Computers, vol. C-22, issue 8, 1973, pp. 786–793.

[16] Chervyakov N.I., Lyakhov P.A., Valueva M.V. Increasing of Convolutional Neural Network Performance Using Residue Number System. In Proc. of the International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), 2017, pp. 135–140.

[17] Tung C., Huang S.A High-Performance Multiply-Accumulate Unit by Integrating Additions and Accumulations into Partial Product Reduction Process. IEEE Access, vol. 8, 2020, pp. 87367-87377.

[18] Habibi Aghdam H., Jahani Heravi E. Guide to Convolutional Neural Networks. Springer, 2017, 305 p.

[19] Valueva M., Valuev G. et al. Construction of Residue Number System Using Hardware Efficient Diagonal Function. Electronics, vol. 8, issue 6, 2019, article no 694, pp. 1-16.

[20] Chervyakov N.I., Molahosseini A.S. et al. Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem. International Journal of Computer Mathematics, vol. 94, issue 9, 2017, pp. 1833–1849.

[21] Haykin S.S. Neural networks: a comprehensive foundation. Prentice Hall, 1999. 842 p.

[22] LeCun Y., Bottou L. et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, issue 11, 1998, pp.2278-2324.

[23] Abadi M., Agarwal A. et al. TensorFlow: Large-scale machine learning on heterogeneous systems. arXiv preprint arXiv:1603.04467, 1916, 19 p.

[24] Xiao H., Kashif R., Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017, 6 p.

[25] Krizhevsky A. Learning Multiple Layers of Features from Tiny Images. Technical Report TR-2009, University of Toronto, 2009, 60 p.

Информация об авторах / Information about authors

Мария Васильевна ВАЛУЕВА получила степени бакалавра и магистра прикладной математики и компьютерных наук в СКФУ в 2016 и 2018 годах соответственно, где в настоящее время работает над кандидатской диссертацией. Работает в Северо-Кавказском центре математических исследований младшим научным сотрудником. Ее исследовательские интересы включают высокопроизводительные вычисления, модулярную арифметику, искусственные нейронные сети и цифровую обработку изображений.

Maria Vasilyevna VALUEVA – received the bachelor's and master's degrees in applied mathematics and computer science from NCFU in 2016 and 2018, respectively, where she is currently pursuing the Ph.D. degree. She works with the North-Caucasus Center for Mathematical Research as a Junior Researcher. Her research interests include high performance computing, modular arithmetic, artificial neural networks, and digital image processing.

Георгий Вячеславович ВАЛУЕВ получил степень магистра прикладных компьютерных наук в СКФУ в 2020 году, где в настоящее время обучается в аспирантуре. Работает в Северо-Кавказском центре математических исследований младшим научным сотрудником. Его исследовательские интересы включают высокопроизводительные вычисления, цифровую обработку изображений искусственные нейронные сети.

Georgii Vyacheslavovich VALUEV received the master's degree in applied computer science from NCFU), in 2020, where he is currently pursuing the Ph.D. degree. He works at the North-Caucasus Center for Mathematical Research as a Junior Researcher. His research interests include high performance computing, digital image processing and artificial neural networks.

Михаил Григорьевич БАБЕНКО – кандидат физико-математических наук. Сфера научных интересов: облачные вычисления, высокопроизводительные вычисления, система остаточных классов, нейронные сети, криптография.

Mikhail BABENKO - Ph.D. in Physics and Mathematics. His research interests include cloud computing, high-performance computing, residue number systems, neural networks, cryptography.

Андрей Николаевич ЧЕРНЫХ получил степень доктора наук в Институте системного программирования им. В.П. Иванникова РАН. Он является профессором Центра научных исследований и высшего образования в Энсенаде, Нижняя Калифорния, Мексика. В научном плане его интересуют многоцелевая оптимизация распределения ресурсов в облачной среде, проблемы безопасности, планирования, эвристики и метаэвристики, интернет вещей и т.д.

Andrei TCHERNYKH received his doctor of science degree at Ivannikov Institute for System Programming of the Russian Academy of Sciences. He is holding a full professor position in computer science at CICESE Research Center, Ensenada, Baja California, Mexico. He is interested in grid and cloud research addressing multi-objective resource optimization, both, theoretical and experimental, security, uncertainty, scheduling, heuristics and meta-heuristics, adaptive resource allocation, and Internet of Things.

Хорхе Марио КОРТЕС МЕНДОСА получил степень бакалавра компьютерных наук в Автономном университете Пуэблы в 2008 г., степень магистра в 2011 г. и степень доктора философии в области компьютерных наук в 2018 году в Исследовательском центре CICESE. Он является членом Национальной системы исследователей Мексики (SNI) с 2020 года. Его основные интересы включают облачные вычисления, балансировку нагрузки, распределенные вычисления и планирование.

Jorge Mario CORTES-MENDOZA is received his Bachelor’s degree in Computer Science from the Autonomous University of Puebla (Benemérita Universidad Autónoma de Puebla, México) in 2008, the Master’s degree in 2011 and the Ph.D. degree in 2018 from CICESE Research Center in Computer Science. He is a member of the National System of Researchers of Mexico (SNI) since 2020. His main interests include cloud computing, load balancing, distributed computing, and scheduling.