http://dx.doi.org/10.35596/1729-7648-2026-24-2-69-78

УДК 004.021

АППАРАТНАЯ РЕАЛИЗАЦИЯ ДВУХСЛОЙНОЙ НЕЙРОННОЙ СЕТИ НА БАЗЕ FPGA: АНАЛИЗ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ ФУНКЦИЙ АКТИВАЦИИ ReLU И LeakyReLU

О. Р. СУББОТЕНКО, М. И. ВАШКЕВИЧ

Белорусский государственный университет информатики и радиоэлектроники (Минск, Республика Беларусь)

Аннотация. Исследованы методы эффективной аппаратной реализации нейронных сетей на программируемых логических интегральных схемах типа FPGA. В качестве ключевого аспекта рассматривается влияние выбора функций активации на характеристики разрабатываемого устройства. Предложен подход к использованию функций активации, допускающих эффективную аппаратную реализацию, в частности, обосновано применение LeakyReLU как компромисса между вычислительной простотой и точностью классификации. Для апробации подхода разработана архитектура двухслойной сети прямого распространения, выполнены оптимизация ее гиперпараметров и аппаратная реализация на плате PYNQ Z2. Проведен анализ влияния разрядности представления коэффициентов в формате с фиксированной запятой на точность распознавания базы данных MNIST и на аппаратные затраты. Экспериментально определена оптимальная разрядность дробной части (9 бит), обеспечивающая точность 95,27 % при экономном расходовании ресурсов программируемых логических интегральных схем. Дополнительно с использованием дивергенции Кульбака – Лейблера оценено искажение распределения весов при квантовании, на основе чего построена регрессионная модель для предсказания точности нейронной сети с квантованными коэффициентами.

Ключевые слова: нейронная сеть, LeakyReLU, распознавание рукописных цифр, полносвязный слой, программируемая логическая интегральная схема, PYNQ, дивергенция Кульбака – Лейблера.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Субботенко, О. Р. Аппаратная реализация двухслойной нейронной сети на базе FPGA: анализ эффективности применения функций активации ReLU и LeakyReLU / О. Р. Субботенко, М. И. Вашкевич // Доклады БГУИР. 2026. Т. 24, № 2. С. 69–78. http://dx.doi.org/10.35596/1729-7648-2026-24-2-69-78.

HARDWARE IMPLEMENTATION OF A TWO-LAYER NEURAL NETWORK BASED ON FPGA: ANALYSIS OF THE EFFICIENCY OF USING ReLU AND LeakyReLU ACTIVATION FUNCTIONS

OLGA SUBBOTENKO, MAXIM VASHKEVICH

Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)

Abstract. Methods for the efficient hardware implementation of neural networks on FPGA-type programmable logic integrated circuits are investigated. A key aspect is the influence of the choice of activation functions on the characteristics of the developed device. An approach to the use of activation functions that allow for efficient hardware implementation is proposed. In particular, the use of LeakyReLU as a compromise between computational simplicity and classification accuracy is justified. To test the approach, a two-layer feedforward network architecture was developed, its hyperparameters were optimized, and hardware implementation was carried out on a PYNQ Z2 board. An analysis of the impact of the bit depth of the coefficients in fixed-point format on the recognition accuracy of the MNIST database and on hardware costs is conducted. The optimal bit depth of the fractional part (9 bits) was experimentally determined, ensuring an accuracy of 95.27 % while economically using the resources of programmable logic integrated circuits. Additionally, using the Kullback – Leibler divergence, the distortion of the weight distribution during quantization was estimated, on the basis of which a regression model was constructed to predict the accuracy of a neural network with quantized coefficients.

Keywords: neural network, LeakyReLU, handwritten digit recognition, fully connected layer, programmable logic integrated circuit, PYNQ, Kullback – Leibler divergence.

Conflict of interests. The authors declare that there is no conflict of interests.

For citation. Subbotenko O., Vashkevich M. (2026) Hardware Implementation of a Two-Layer Neural Network Based on FPGA: Analysis of the Efficiency of Using ReLU and LeakyReLU Activation Functions. *Doklady BGUIR*. 24 (2), 69–78. http://dx.doi.org/10.35596/1729-7648-2026-24-2-69-78 (in Russian).

Введение

В настоящее время технологии искусственного интеллекта широко используются для таких задач, как распознавание речи, классификация и генерация изображений, диагностика заболеваний и проч. [1, 2]. В некоторых областях применения нейронных сетей (НС) их точность превосходит человеческую, однако достижение таких показателей требует построения сложных многослойных архитектур. Как следствие, вычислительная сложность современных нейросетевых моделей стремительно возрастает. Для ускорения НС широко применяются графические (GPU) и тензорные процессоры (TPU), специализирующиеся на параллельном выполнении матричных операций. Однако они обладают универсальной архитектурой и рассчитаны на использование форматов данных (например, FP32/FP16), которые не оптимальны для данной задачи. Следствием их применения является неоправданно высокое энергопотребление даже там, где можно было бы обойтись менее точными, но более производительными вычислениями.

Программируемые логические интегральные схемы (ПЛИС) типа FPGA (Field Programmable Gate Array) – реконфигурируемые вычислительные платформы, отличающиеся низким энергопотреблением и высокой производительностью [2–4]. Реализация НС на ПЛИС позволяет применять пользовательские типы данных, что напрямую влияет на аппаратные затраты и энергопотребление ПЛИС.

В статье предложена и исследована архитектура НС для классификации изображений с помощью аппаратно-ориентированных активационных функций ReLU и LeakyReLU. В фокусе исследования находился вопрос оправданности применения LeakyReLU в архитектуре НС для ПЛИС, где традиционно используются активационные функции ReLU [2]. Проанализировано влияние изменения разрядности представления параметров НС на точность классификации и аппаратные затраты. Предложен подход к прогнозированию точности нейросетевой модели, достигаемой в результате квантования ее параметров.

Исследование проходило в несколько этапов. На первом разрабатывалась архитектура НС с активационной функцией ReLU/LeakyReLU и выполнялось ее обучение. Для этой задачи использовались изображения рукописных цифр из базы MNIST. На втором этапе создавалась структура и выполнялось описание IP-блока разработанной НС на языке SystemVerilog. На третьем – тестировалась работа IP-блока с параметрами различной разрядности, на четвертом этапе анализировались полученные данные и выполнялось их сравнение с другими реализациями.

Разработка программной модели нейронной сети

В процессе исследования рассматривалась задача распознавания рукописных цифр на изображениях из базы данных MNIST. В ней содержится 70 тыс. изображений размерами 28×28 пикселей рукописных цифр от 0 до 9 в оттенках серого. 60 тыс. изображений относятся к обучающей выборке, 10 тыс. – к тестовой. Все изображения помечены соответствующей цифрой.

Для решения данной задачи использовалась двухслойная НС прямого распространения. Структура предлагаемой НС, которая в дальнейшем будет обозначаться как MLP-2L, представлена на рис. 1.

На вход НС подавались изображения размерами 28×28 пикселей, которые предварительно преобразовывались в одномерный вектор. Таким образом, входной слой содержал 784 нейрона. Далее данные поступали в скрытый слой, где вычислялся результат по следующей формуле:

$$z_t^{[1]} = \sum_{s=0}^{783} \left(w_{t,s}^{[1]} x_s \right) + b_t^{[1]}, \tag{1}$$

где $z_t^{[1]}$ – значение преактивации скрытого слоя; $t \in [0;13]$; $w_{t,s}^{[1]}$ – вес первого слоя; x_s – входное значение (пиксель); $b_t^{[1]}$ – смещение t-го нейрона первого слоя.

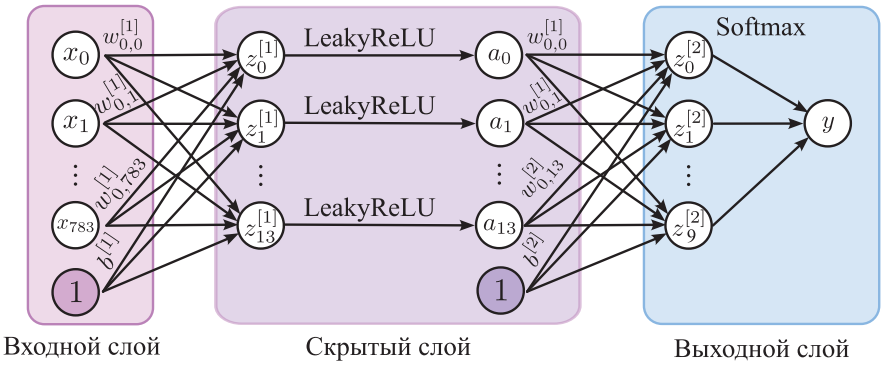


Рис. 1. Структура двухслойной нейронной сети MLP-2L
Fig. 1. Structure of the two-layer neural network MLP-2L

Рассчитанные значения поступали на вход активационной функции LeakyReLU, которую можно описать следующим выражением:

$$\text{LeakyReLU}(x) = \begin{cases} x, & x \geq 0; \\ neg\_slope \cdot x, & x < 0, \end{cases} \tag{2}$$

где x – нейроны, рассчитываемые во внутреннем слое; neg_slope – гиперпараметр НС, определяющий угол наклона для отрицательных входных значений.

Полученные значения подаются на вход выходного слоя, где преобразуются по формуле

$$z_t^{[2]} = \sum_{s=0}^{13} \left(w_{t,s}^{[2]} \cdot a_s \right) + b_t^{[2]}, \tag{3}$$

где $z_t^{[2]}$ – значение преактивации выходного слоя; $t \in [0;9]$; $w_{t,s}^{[2]}$ – вес второго слоя; a_s – выход скрытого слоя; $b_t^{[2]}$ – смещение t-го нейрона второго слоя.

Вычисленные 10 значений поступали на вход активационной функции softmax

$$y_t = \text{softmax}\left(z_t^{[2]} \right) = \frac{\exp\left(z_t^{[2]} \right)}{\sum_{j=0}^{9} \exp\left(z_j^{[2]} \right)}. \tag{4}$$

Значения y_t можно интерпретировать как вероятность отнесения входного изображения к классу t. В качестве окончательного решения выбирался класс, имеющий наибольшую вероятность.

Главной особенностью модели MLP-2L по сравнению с аналогичными двухслойными НС [2, 5] является использование на скрытом слое активационной функции LeakyReLU, которая впервые была предложена в [6]. На рис. 2 приведены графики активационных функций ReLU и LeakyReLU.

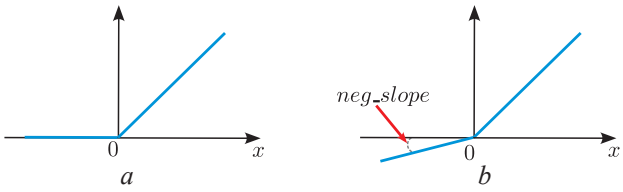


Рис. 2. Сравнение активационных функций ReLU (*a*) и LeakyReLU (*b*)
Fig. 2. Comparison of ReLU (*a*) and LeakyReLU (*b*) activation functions

Основное преимущество LeakyReLU перед ReLU – преодоление проблемы «умирающих» нейронов. В НС с функцией ReLU отрицательное значение преактивации обнуляет выход нейрона, что приводит в процессе обратного распространения ошибки к нулевому градиенту на этом нейроне и полной остановке его обучения. LeakyReLU, сохраняя малый, но ненулевой наклон для отрицательной области аргумента, гарантирует прохождение градиента всегда, тем самым повышая устойчивость процесса обучения.

С точки зрения схемотехники функция ReLU имеет более простую реализацию, поскольку не требует вычисления произведения $neg\_slope \cdot x$, как функция LeakyReLU. В данной статье для сокращения аппаратных затрат на LeakyReLU установлено значение $neg\_slope = 2^{-k}$, где параметр k представляет собой положительное целое число, определяемое в процессе оптимизации гиперпараметров. Вследствие накладываемого ограничения, умножение на параметр neg_slope можно реализовать в ПЛИС при помощи операции сдвига на k разрядов вправо, что с точки зрения аппаратных затрат значительно более эффективно, чем умножение на константу общего вида.

Аппаратная реализация нейронной сети на FPGA

Для аппаратной реализации НС был разработан IP-блок, структура которого приведена на рис. 3.

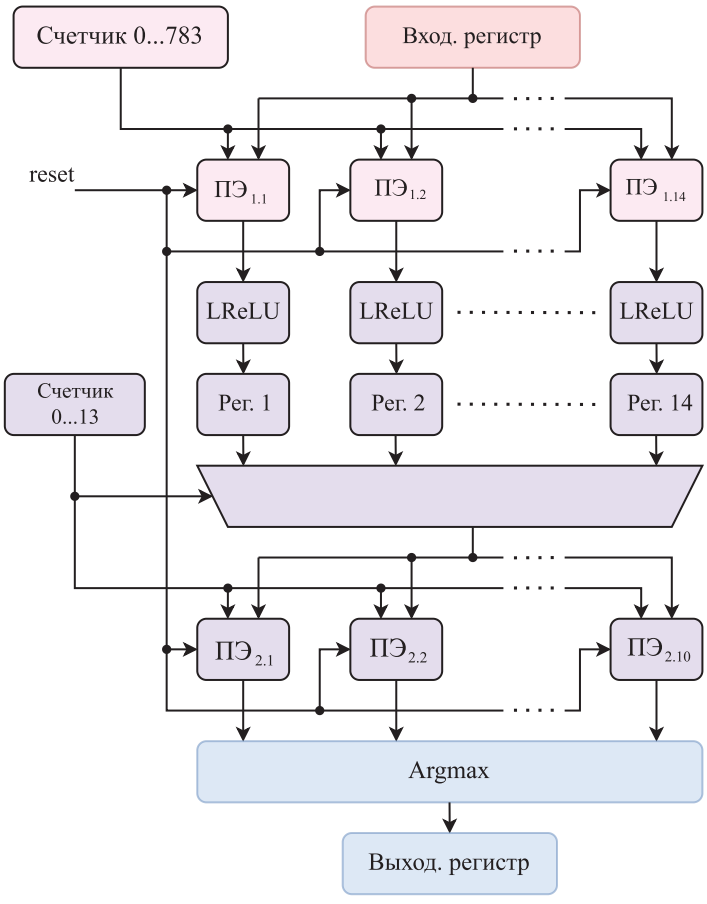
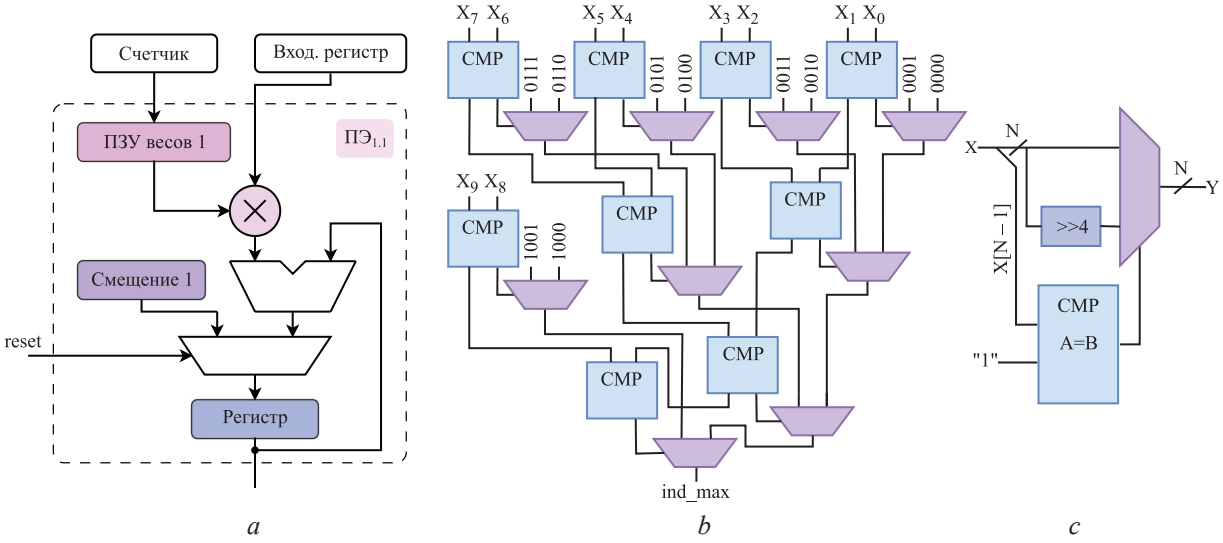


Рис. 3. Структура IP-блока двухслойной нейронной сети MLP-2L
Fig. 3. The structure of the IP block of the two-layer neural network MLP-2L

Полносвязные слои НС реализованы при помощи процессорных элементов (ПЭ), каждый из которых соответствует одному нейрону. Для данной реализации необходимо 24 ПЭ: 14 для первого слоя и 10 для второго. В каждом из них выполняется операция умножения с накоплением, для чего используется матричный умножитель, поскольку его структура позволяет вычислить произведение за один период тактового сигнала. На вход умножителей ПЭ первого слоя подаются пиксели обрабатываемого изображения и соответствующие весовые коэффициенты, полученные на этапе обучения модели. Для умножителей ПЭ второго слоя входными являются значения функции LeakyReLU, а также весовые коэффициенты. При сбросе каждый ПЭ инициализируется соответствующим значением смещения. Структура одного ПЭ приведена на рис. 4, *а*.

Функция активации softmax широко используется при проектировании и обучении НС, однако ее реализация на FPGA осложняется вычислением экспоненты. Более того, вычисление softmax является избыточным на этапе исполнения (inference) НС и может быть заменено вычислением

функции argmax [7]. В структуре IP-блока НС (рис. 3) для реализации этой функции используется модуль argmax, изображенный на рис. 4, *b*. Это устройство осуществляет поиск индекса максимального элемента на выходе последнего слоя НС. Выходной слой содержит десять нейронов, каждый из которых соответствует определенной цифре (от 0 до 9). Таким образом, найденный индекс однозначно определяет класс, к которому сеть отнесла входной образ. Функция активации LeakyReLU аппаратно реализуется при помощи компаратора, сдвигового регистра и мультиплексора, адресным сигналом для которого является знаковый бит поступившего на вход числа (рис. 4, *c*).



a *b* *c*

Рис. 4. Функциональные модули устройства: *a* – структура процессорного элемента; *b*, *c* – структура модулей argmax и LeakyReLU соответственно

Fig. 4. Functional modules of the device: *a* – structure of the processing element; *b*, *c* – the structure of the argmax and LeakyReLU modules, respectively

IP-блок двухслойной НС описан на языке SystemVerilog и используется как компонент системы на кристалле (СнК). В качестве аппаратной платформы для реализации проекта использовалась отладочная плата PYNQ Z2 (Python Productivity for Zynq) на базе ZYNQ-7000, которая представляет собой СнК, объединяющую процессор ARM Cortex-A9 и программируемую логику FPGA (рис. 5).

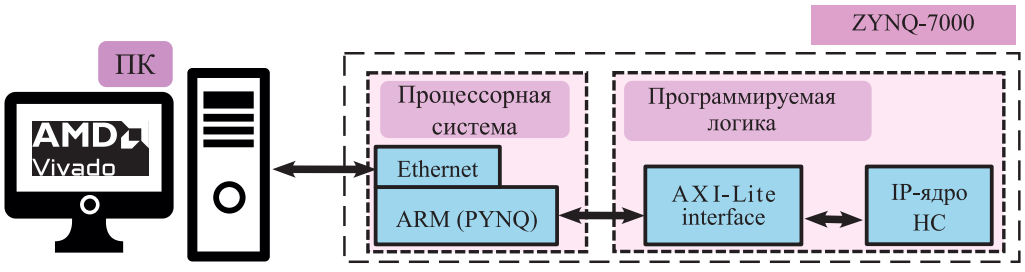


Рис. 5. Реализация нейронной сети на базе платформы PYNQ Z2

Fig. 5. Implementation of a neural network based on the PYNQ Z2 platform

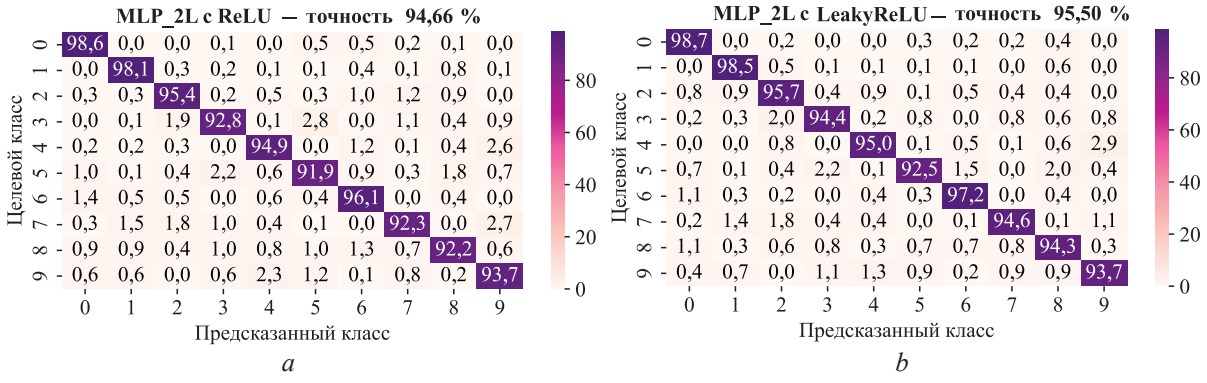
Для упрощения разработки и тестирования на данной платформе используется технология PYNQ, которая представляет собой специализированный дистрибутив операционной системы (ОС) Linux для гетерогенных вычислительных систем семейства Xilinx Zynq, дополненный набором библиотек Python, предоставляющих программный интерфейс (API) для управления IP-ядрами, реализованными в программируемой логике данной СнК. В ОС Linux запущено ядро Jupyter Notebook, где создан блокнот для управления работой IP-блока. Из процессорной системы посредством интерфейса AXI-Lite в программируемую логику передается по одному пикселю изображения. По окончании вычислений результат распознавания передается обратно в процессорную систему по тому же интерфейсу.

Экспериментальные исследования и их результаты

Обучение НС MLP-2L выполнялось с использованием языка Python и библиотеки PyTorch. В качестве функции потерь применялась перекрестная энтропия, оптимизация выполнялась методом Adam. Регулировка скорости обучения η в процессе обучения осуществлялась по методу косинусного отжига с перезапуском. В данном планировщике предполагается, что скорость обучения η изменяется от значения η_{max} по косинусному закону до η_{min} в течение периода T_0, после чего процесс повторяется. Общее число циклов повторения определяется как N_{epochs}/T_0, где N_{epochs} – общее число эпох обучения модели. В процессе обучения принимали $N_{epochs} = 200$ и $\eta_{min} = 10^{-6}$. Значения T_0 и η_{max} выбирались посредством оптимизации гиперпараметров.

Поиск гиперпараметров выполнялся при помощи байесовской оптимизации по методу TPE (tree-structured Parzen estimator), реализованой в библиотке Optuna. Данный подход позволяет эффективно исследовать пространство гиперпараметров, строя вероятностную модель целевой функции и последовательно выбирая наиболее перспективные комбинации. Далее приведены оптимизируемые гиперпараметры и диапазоны их значений. Скорость обучения η_{max} и коэффициент L2-регуляризации λ задавались в непрерывном диапазоне с логарифмическим масштабированием, что обусловлено необходимостью равномерного перебора значений в широком интервале: для η_{max} – от $1 \cdot 10^{-4}$ до $5 \cdot 10^{-2}$, для λ – от 10^{-7} до $3 \cdot 10^{-3}$; вероятность отключения нейронов (дропаут-регуляризация) выбиралась равномерно из отрезка [0,01, 0,5]; параметр neg_slope функции LeakyReLU оптимизировался как категориальная переменная, принимающая значения из набора [2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 0]. Такой дискретный набор охватывает характерные для данной функции значения, включая нулевое (соответствующее обычному ReLU), и позволяет выявить наиболее эффективный вариант. Параметр T_0 также рассматривался как категориальный с возможными значениями [20, 10, 4, 2]. Наконец, размер мини-батча выбирался из степеней двойки в диапазоне от 32 до 2048.

В ходе оптимизации гиперпараметров для модели MLP-2L максимальная точность с функцией ReLU составила 94,66 %, тогда как с LeakyReLU – 95,50 % ($neg\_slope = 2^{-4}$). Таким образом, применение LeakyReLU в аппаратной реализации является оправданным, поскольку обеспечивает прирост качества классификации. Для наглядного представления и анализа результатов полученных вариантов модели MLP-2L строились матрицы ошибок (рис. 6), анализ которых показал, что для функций ReLU и LeakyReLU наибольшая точность распознавания у цифры 0 – 98,6 и 98,7 % соответственно, наименьшая – у цифры 5 – 91,9 и 92,5 %.



a *b*

Рис. 6. Матрица ошибок для нейронной сети с активационной функцией: *a* – ReLU; *b* – LeakyReLU
Рис. 6. Error matrix for a neural network with activation function: *a* – ReLU; *b* – LeakyReLU

Следующим этапом было тестирование НС MLP-2L, на котором исследовалось влияние представления весов НС на точность распознавания и аппаратные затраты FPGA. Разрядность дробной части весов изменялась от 2 до 12. Также проводилось исследование аппаратных затрат для каждой FPGA-реализации НС на основе отчетов об использованных ресурсах ПЛИС, полученных в САПР Vivado 2024.1. С ростом разрядности коэффициентов НС увеличивается точность распознавания рукописных цифр, и в то же время увеличивается количество требуемых блоков LUT (Look-Up Tables) и триггеров FF (Flip Flop). Результаты экспериментов приведены на рис. 7 в виде графика, на котором отображена информация о точности распознавания требуемых блоков LUT и FF для каждой разрядности коэффициентов НС.

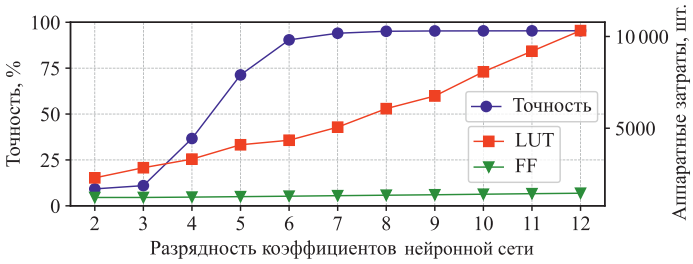


Рис. 7. Точность и аппаратные затраты на реализацию нейронной сети
Fig. 7. Accuracy and hardware costs of implementing a neural network

Анализ зависимости точности распознавания от разрядности дробной части весовых коэффициентов показывает, что при увеличении числа разрядов с 3 до 7 наблюдается скачкообразный рост точности. При дальнейшем наращивании разрядности рост точности замедляется и приобретает характер, близкий к линейному. Увеличение разрядности закономерно ведет к росту аппаратных затрат. Количество LUT возрастает с увеличением разрядности каждого MAC-ядра, что связанно с квадратичным ростом сложности матричного умножителя и увеличением разрядности сумматора. В отличие от LUT, число триггеров FF остается практически неизменным во всем диапазоне исследуемых разрядностей, что говорит о незначительном влиянии разрядности весовых коэффициентов на количество триггеров.

Аппаратные затраты отладочной платы PYNQ Z2 на базе ПЛИС ZYNQ-7000 для случая представления коэффициентов НС MLP-2L приведены в табл. 1. Данные соответствуют случаю представления коэффициентов с 9-ю разрядами в дробной части.

Таблица 1. Аппаратные затраты на реализацию нейронной сети MLP-2L на FPGA PYNQ Z2
Table 1. Hardware costs for implementing the MLP-2L neural network on the PYNQ Z2 FPGA

Вариант блока	Количество	Доступно	Использование, %
LUT	6757	53 200	12,70
FF	1374	106 400	1,29
Блочная память	12	140	8,57

Результаты сравнения модели MLP-2L с другими НС прямого распространения [1] приведены в табл. 2. Сравнение проводилось по следующим критериям: точность распознавания, тактовая частота, время обработки одного изображения, аппаратные затраты.

Таблица 2. Сравнение MLP-2L с другими нейронными сетями прямого распространения
Table 2. Comparison of MLP-2L with other feedforward neural networks

Показатель	Модель					
	MLP-2L	RVNN_64	CVNN_64	RVNN_128	CVNN_128	RVNN_Raw_MNIST
Точность распознавания, %	95,27	85,9	87,0	87,5	88,3	96
Период тактовых импульсов, нс	18,11	8	10,5	8	10	8
Время обработки, мс	0,16	0,64	0,84	1,28	1,6	3,92
LUT	6757	9123	17 723	13 122	24 164	20 993
FF	1374	3110	5936	5703	11 520	16 066
Блочная память	12	–	–	–	–	160
DSP	–	485	1333	469	1333	507

Предлагаемую модель MLP-2L по точности превосходит лишь RVNN_Raw_MNIST. MLP-2L требует наименьших ресурсов ПЛИС и быстрее всех перечисленных моделей обрабатывает одно изображение. Однако период тактовых импульсов по сравнению с остальными принимает наибольшее значение, что связано с реализацией матричного умножителя MAC-ядер на LUT-блоках.

С целью анализа влияния квантования коэффициентов НС на их численные значения была рассчитана дивергенция Кульбака – Лейблера (рис. 8). Она характеризует степень расхождения между исходными коэффициентами и их представлением в формате Q7.n, где n – количество битов в дробной части. На рис. 8 использованы следующие обозначения: KL – рассчитанное значение дивергенции Кульбака – Лейблера; w_n, w_n^q – распределение коэффициентов n-го слоя НС до и после квантования соответственно. На графиках точности распознавания в зависимости от количества бит, приходящегося на дробную часть ($Q = 4$ и $Q = 5$), виден скачкообразный прирост точности, что также повлияло на различие в гистограммах для весов и их квантованных версий для коэффициентов первого и второго слоев. Дивергенция для $Q = 4$ и $Q = 5$ битов в дробной части для первого слоя отличается в 2703 раза, для второго – в 14 раз.

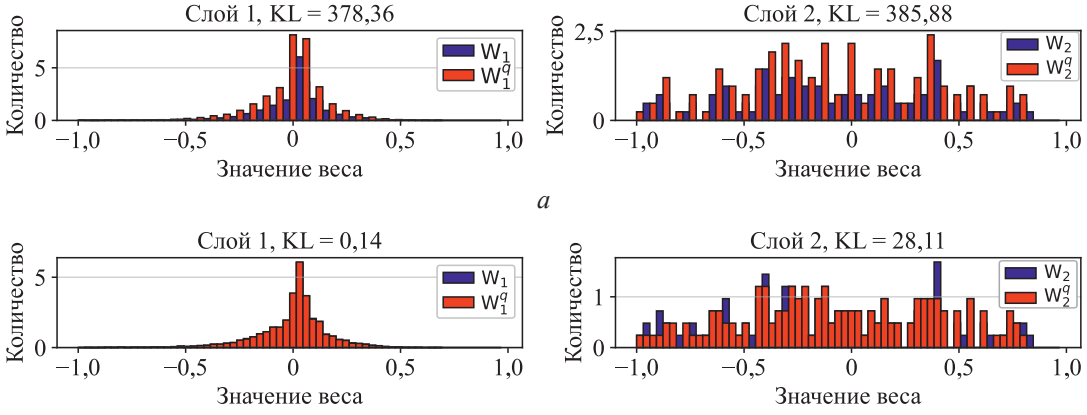


Рис. 8. Гистограммы коэффициентов нейронной сети до и после квантования при распределении весов и их квантованных версий для: *a* – $Q = 4$; *b* – $Q = 5$
Fig. 8. Histograms of neural network coefficients before and after quantization for the distribution of weights and their quantized versions for: *a* – $Q = 4$; *b* – $Q = 5$

Также исследовалась возможность предсказания точности модели НС с квантованными параметрами. Для этой цели строилась модель линейной регрессии (рис. 9), в которой предикторами выступали значения дивергенции Кульбака – Лейблера для каждой разрядности, а целевой переменной – точности распознавания.

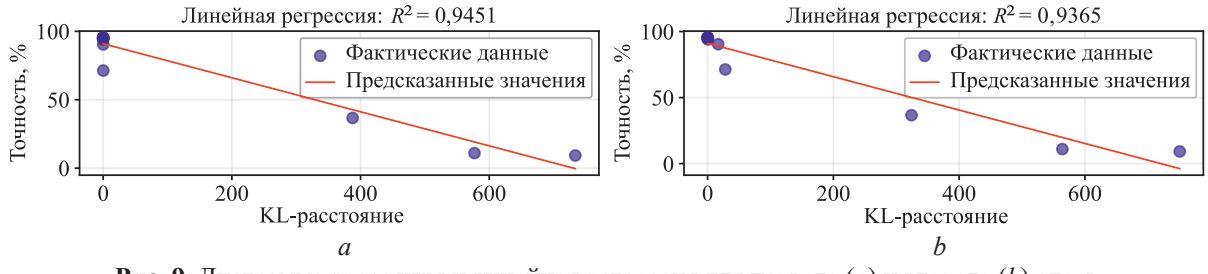


Рис. 9. Диаграмма рассеяния и линейная регрессия для первого (*a*) и второго (*b*) слоев
Fig. 9. Scatterplot and linear regression for the first (*a*) and second (*b*) layers

Коэффициент детерминации R^2 для обоих слоев принимает высокие значения (0,9451 и 0,9365), что свидетельствует о достоверности предсказания точности распознавания для каждой разрядности.

Заключение

1. Представлены результаты исследования подходов к эффективной аппаратной реализации нейронной сети на FPGA. На примере двухслойной сети прямого распространения, реализованной на платформе PYNQ Z2, продемонстрирована методика проектирования, включающая оптимизацию гиперпараметров, выбор функций активации и анализ влияния разрядности представления коэффициентов.

2. В ходе исследования подтверждена эффективность предложенного подхода к выбору функций активации, допускающих простую схемотехническую реализацию. Сравнительный анализ показал, что использование LeakyReLU обеспечивает прирост точности классификации на 0,84 % по сравнению с ReLU при сопоставимой сложности аппаратной реализации.

3. Исследовано влияние разрядности представления весовых коэффициентов нейронной сети на точность и аппаратные затраты. Установлено, что оптимальной для разработанной модели нейронной сети при ее реализации на FPGA является разрядность дробной части 9 бит, при которой достигается высокая точность распознавания (95,27 % для модели, использующей LeakyReLU). Выявленный характер зависимостей позволяет производить обоснованный выбор разрядности на ранних этапах проектирования за счет поиска компромисса между точностью вычислений и аппаратной сложностью.

4. Для количественной оценки искажений, вносимых квантованием, применена дивергенция Кульбака – Лейблера. На основе полученных значений дивергенции построена регрессионная модель, демонстрирующая высокую предсказательную способность (R^2 = 0,9451 для первого и R^2 = 0,9365 для второго слоев). Данный результат подтверждает возможность прогнозирования точности нейросетевой модели с квантованными коэффициентами без проведения полного цикла аппаратного тестирования.

5. Исследование выполнено в рамках работы над научным проектом в лаборатории БГУИР-YADRO в 2025/2026 учебном году.

Список литературы

1. Ahmad, M. FPGA Implementation of Complex-Valued Neural Network for Polar-Represented Image Classification / M. Ahmad, L. Zhang, M. E. H. Chowdhury // Sensors. 2024. Vol. 24, No 3.
2. Kwon, J. Design of a Low-Area Digit Recognition Accelerator Using MNIST Database / J. Kwon, S. Kim // JOIV: International Journal on Informatics Visualization. 2022. Vol. 6, No 1. P. 53–59.
3. FPGA Acceleration on a Multilayer Perceptron Neural Network for Digit Recognition / I. Westby [et al.] // The Journal of Supercomputing. 2021. Vol. 77, No 12. P. 14356–14373. https://doi.org/10.1007/s11227-021-03849-7.
4. Кривальцевич, Е. А. Исследование аппаратной реализации нейронной сети прямого распространения для распознавания рукописных цифр на базе FPGA / Е. А. Кривальцевич, М. И. Вашкевич // Доклады БГУИР. 2025. Т. 23, № 2. С. 101–108. http://dx.doi.org/10.35596/1729-7648-2025-23-2-101-108.
5. Субботенко, О. Р. FPGA реализация двухслойной нейронной сети прямого распространения для распознавания изображений / О. Р. Субботенко, М. И. Вашкевич // Информационные технологии и системы 2025 (ИТС 2025): материалы Междунар. науч. конф., Минск, 19 нояб. 2025. Минск: Белор. гос. ун-т информ. и радиоэлек., 2025. С. 153–154.
6. Maas, A. L. Rectifier Nonlinearities Improve Neural Network Acoustic Models / A. L. Maas, A. Y. Hannun, A. Y. Ng // Proceedings of the 30th International Conference on Machine Learning (ICML), 2013. P. 1–6.
7. Субботенко, О. Р. Разработка аппаратного модуля вычисления функции argmax на базе FPGA / О. Р. Субботенко // Компьютерные системы и сети: материалы 61-й науч. конф. аспир., магистрантов и студ., Минск, 22–26 апр. 2025 г. Минск: Белор. гос. ун-т информ. и радиоэлек., 2025. С. 584–585.

Поступила 06.03.2026 Принята в печать 03.04.2026

References

1. Ahmad M., Zhang L., Chowdhury M. E. H. (2024) FPGA implementation of Complex-Valued Neural Network for Polar-Represented Image Classification. *Sensors*. 24 (3).
2. Kwon J., Kim S. (2022) Design of a Low-Area Digit Recognition Accelerator Using MNIST Database. *JOIV: International Journal on Informatics Visualization*. 6 (1), 53–59.
3. Westby I., Yang X., Liu T., Xu H. (2021) FPGA Acceleration on a Multilayer Perceptron Neural Network for Digit Recognition. *The Journal of Supercomputing*. 77 (12), 14356–14373. https://doi.org/10.1007/s11227-021-03849-7.
4. Krivalcevich E. A., Vashkevich M. I. (2025) Investigation of Hardware Implementation of a Feedforward Neural Network for Handwritten Digit Recognition Based on FPGA. *Doklady BGUIR*. 23 (2), 101–108. http://dx.doi.org/10.35596/1729-7648-2025-23-2-101-108 (in Russian).
5. Subbotenko O. R., Vashkevich M. I. (2025) FPGA Implementation of a Two-Layer Direct Propagation Neural Network for Image Recognition. *Information Technologies and Systems 2025 (ITS 2025): Proceedings of the Int. Conf., Minsk, Nov. 19*. Minsk, Belarusian State University of Informatics and Radioelectronics. 153–154 (in Russian).

6. Maas A. L., Hannun A. Y., Ng A. Y. (2013) Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Proceedings of the 30th International Conference on Machine Learning (ICML)*. 1–6.
7. Subbotenko O. R. (2025) Development of a Hardware Module for Calculating the Argmax Function Based on FPGA. *Faculty of Computer Systems and Networks: Proceedings of the 61st Scientific Conference of Graduate Students, Undergraduates and Students, Minsk, Apr. 22–26*. Minsk, Belarusian State University of Informatics and Radioelectronics. 584–585.

Received: 6 March 2026 Accepted: 3 April 2026

Вклад авторов

Субботенко О. Р. реализовала и обучила нейронную сеть, аппаратно реализовала структуру нейронной сети, провела экспериментальные исследования, подготовила черновик статьи.

Вашкевич М. И. определил цель и задачи исследований, принимал участие в аппаратной реализации нейронной сети и тестировании на FPGA, участвовал в проведении экспериментальных исследований, интерпретации результатов эксперимента и работе над текстом статьи.

Authors' contribution

Subbotenko O. implemented and trained the neural network, implemented the neural network structure in hardware, conducted experimental studies, and prepared a draft of the article.

Vashkevich M. defined the purpose and objectives of the research, participated in the hardware implementation of the neural network and testing on FPGA, participated in conducting experimental studies, interpreting the experimental results and working on the text of the article.

Сведения об авторах

Субботенко О. Р., студентка, Белорусский государственный университет информатики и радиоэлектроники

Вашкевич М. И., д-р техн. наук, проф. каф. встраиваемых вычислительных систем, Белорусский государственный университет информатики и радиоэлектроники

Адрес для корреспонденции

220013, Республика Беларусь,
Минск, ул. П. Бровки, 6
Белорусский государственный университет
информатики и радиоэлектроники
Тел.: +375 17 293-84-20
E-mail: vashkevich@bsuir.by
Вашкевич Максим Иосифович

Information about the authors

Subbotenko O., Student, Belarusian State University of Informatics and Radioelectronics

Vashkevich M., Dr. Sci. (Tech.), Professor at the Department of Embedded Computing System, Belarusian State University of Informatics and Radioelectronics

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovki St., 6
Belarusian State University
of Informatics and Radioelectronics
Tel.: +375 17 293-84-20
E-mail: vashkevich@bsuir.by
Vashkevich Maxim