http://dx.doi.org/10.35596/1729-7648-2025-23-2-101-108

УДК 004.021

ИССЛЕДОВАНИЕ АППАРАТНОЙ РЕАЛИЗАЦИИ НЕЙРОННОЙ СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ РУКОПИСНЫХ ЦИФР НА БАЗЕ FPGA

Е. А. КРИВАЛЬЦЕВИЧ, М. И. ВАШКЕВИЧ

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

Аннотация. Разработана аппаратная реализация на базе программируемых логических интегральных схем (ПЛИС) типа Field Programmable Gate Array однослойной нейронной сети прямого распространения для распознавания рукописных цифр. Исследовано влияние разрядности коэффициентов сети на точность распознавания и на аппаратные затраты ПЛИС. Обучение нейронной сети выполнялось с помощью базы рукописных цифр MNIST. Прототип нейронной сети был реализован в виде IP-ядра на отладочной плате ZYBO Z7. Разработанный прототип использовался для выполнения экспериментов с различной разрядностью представления коэффициентов нейронной сети. Построены графики точности распознавания и количества аппаратных ресурсов ПЛИС в зависимости от разрядности представления коэффициентов нейронной сети. Выполнен анализ полученных в результате обучения нейронной сети коэффициентов с использованием разложения на битовые плоскости. Показано, что для представления коэффициентов нейронной сети достаточно 5 разрядов, поскольку они содержат основную, усвоенную сетью, информацию, обеспечивая экономное расходование ресурсов ПЛИС и высокую точность распознавания (92,4 %).

Ключевые слова: нейронная сеть, распознание рукописных цифр, полносвязный слой, MNIST, FPGA, битовые плоскости.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. 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.

INVESTIGATION OF HARDWARE IMPLEMENTATION OF A FEEDFORWARD NEURAL NETWORK FOR HANDWRITTEN DIGIT RECOGNITION BASED ON FPGA

EGOR A. KRIVALCEVICH, MAXIM I. VASHKEVICH

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

Abstract. A hardware implementation based on Field Programmable Gate Array (FPGA) of a single-layer feedforward neural network for handwritten digit recognition has been developed. The effect of the network coefficient bit depth on the recognition accuracy and FPGA hardware costs has been studied. The neural network was trained using the MNIST handwritten digit database. The neural network prototype was implemented as an IP core on the ZYBO Z7 debug board. The developed prototype was used to perform experiments with different bit depths of neural network coefficient representation. Graphs of recognition accuracy and the amount of FPGA hardware resources depending on the bit depth of neural network coefficient representation have been constructed. The coefficients obtained as a result of neural network training have been analyzed using decomposition into bit planes. It has been shown that 5 bits are sufficient to represent neural network coefficients, since they contain the main information learned by the network, ensuring economical use of FPGA resources and high recognition accuracy (92.4 %).

Keywords: neural network, handwritten digit recognition, fully connected layer, MNIST, FPGA, bit planes.

Conflict of interests. The authors declare no conflict of interests.

For citation. Krivalcevich E. A., Vashkevich M. I. (2025) Investigation of Hardware Implementation of a Feed-forward 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).

Введение

Нейронные сети (НС) играют ключевую роль в развитии информационных технологий, особенно в таких областях, как компьютерное зрение и искусственный интеллект [1]. Широкое распространение НС приводит к тому, что появляется необходимость создания специальных аппаратных акселераторов, позволяющих повысить производительность приложений, основанных на нейросетевых технологиях. Программируемые логические интегральные схемы (ПЛИС) типа FPGA (Field Programmable Gate Array) представляют собой реконфигурируемые вычислительные платформы, имеющие невысокое энергопотребление. По этой причине ПЛИС часто выбирают в качестве вычислительной среды для реализации НС, особенно в тех случаях, когда производительности процессоров общего назначения недостаточно, а высокое энергопотребление графических процессоров неприемлемо. Это особенно актуально в контексте разработки встраиваемых систем и роботизированных платформ [1–3].

К преимуществам аппаратной реализации НС на базе ПЛИС относится возможность применять пользовательские типы данных, позволяющие контролировать точность представления параметров нейросетевой модели [2]. Причем выбор точности представления напрямую будет влиять на аппаратные затраты ПЛИС, необходимые для обеспечения выполнения операций над данными.

Цель исследований авторов – разработка аппаратной реализации однослойной НС прямого распространения для распознавания рукописных цифр, а также изучение влияния разрядности коэффициентов НС на точность распознавания и на аппаратные затраты ПЛИС. Для обучения НС использовалась база изображений рукописных цифр MNIST, которая является наиболее доступной и удобной для исследовательской задачи.

Процесс разработки и исследование аппаратной реализации НС разбивался на несколько этапов. На первом выполнялись разработка и обучение модели с использованием языка Python и библиотеки PyTorch. На втором этапе разрабатывались архитектура и описание IP-блока НС с применением языка SystemVerilog. На третьем проводилось прототипирование НС на отладочной плате ZYBO Z7. На заключительном этапе выполнялись эксперимент и анализ полученных результатов.

Разработка программной модели нейронной сети

Рассматривалась задача распознавания рукописных цифр по изображениям из набора данных MNIST, который содержал 70 тыс. полутоновых изображений размерами 28×28 пикселей рукописных цифр – от 0 до 9 [4]. Набор разбит на две части выборки: тренировочная и тестовая – 60 тыс. и 10 тыс. изображений соответственно. Использовалась однослойная НС прямого распространения, состоящая из полносвязного слоя с выходной функцией активации softmax [5]. Структура НС представлена на рис. 1.

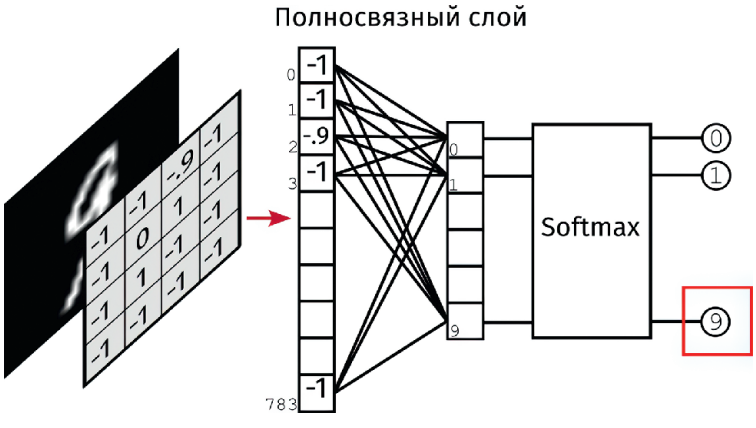

Рис. 1. Структура нейронной сети
Fig. 1. Structure of a neural network

На входе имеется 784 = 28×28 нейрона, каждый подключен к одному из пикселей изображения. На выходе получается слой с десятью нейронами по одному на каждую цифру. Каждый из 10 выходов формируется как линейная комбинация 784 входов:

$$y_i = \text{softmax}\left(\sum_{j=0}^{783} w_{ij}x_j + b_i\right), \quad (1)$$

где y_i – вероятность того, что поданное на вход изображение относится к классу i; w_{ij} – весовой коэффициент, определяющий влияние j-го пикселя входного изображения на вероятность отнесения входного изображения к i-му классу; x_j – j-й пиксель изображения; b_i – смещение (свободный член), $i = 0, \ldots, 9$.

Матрицу размерами 10×784, составленную из весовых коэффициентов w_{ij}, будем называть матрицей весов и обозначать через W. В процессе обучения НС использовался метод стохастического градиентного спуска [5, 6], который имеет два настроечных параметра: скорость обучения α и параметр инерции γ:

$$v_t = \gamma v_{t-1} + \alpha \nabla L; \quad (2)$$

$$W_t = W_{t-1} - v_t, \quad (3)$$

где v_t – скорректированный градиент с учетом параметра инерции; ∇L – градиент функции потерь; W_t – матрица весов НС на текущем шаге t.

В качестве функции потерь использовалась перекрестная энтропия

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=0}^{9} t_i^{(n)} \log(y_i^{(n)}), \quad (4)$$

где $t_i^{(n)}$ – i-я компонента метки n-го изображения, представленной в унитарном коде; N – количество изображений в базе.

Для обучения НС входные данные нормировались таким образом, чтобы среднеквадратическое отклонение равнялось 0,5. Масштабирование данных улучшает производительность и ускоряет процесс обучения НС. Обучение выполнялось на 10 тыс. эпохах, параметр скорости обучения α устанавливался равным 0,003, а инерции – γ = 0,9, что позволило ускорить сходимость процесса и избежать застревания в локальных минимумах функции потерь. На рис. 2 изображен график функции потерь, который показывает, что процесс оптимизации параметров НС сошелся и дальнейших итераций обучения не требуется.

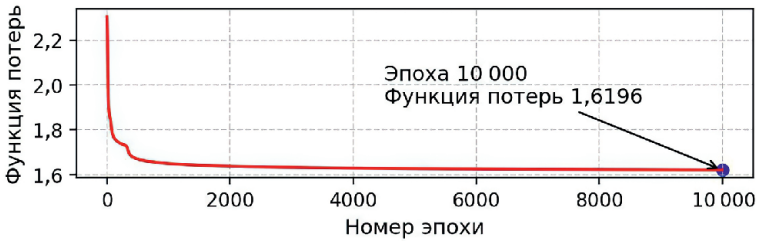

Рис. 2. Результаты обучения нейронной сети
Fig. 2. Neural network training results

Аппаратная реализация нейронной сети на FPGA

На начальном этапе была разработана структура IP-блока НС, показанная на рис. 3. Вычислительной основой разработанного устройства являлись десять MAC (Multiply-Accumulate Operation) ядер, выполняющих умножение вектора изображения на матрицу весов НС и объединенных в общий блок полносвязного слоя. Выбор именно десяти MAC-ядер определяется числом распознаваемых классов изображения, в общем случае расчет выходных значений полносвязного слоя может быть выполнен с использованием произвольного числа MAC-ядер. Однако выбор десяти MAC-ядер позволяет существенно упростить устройство управления. Для реализации умножителя MAC-ядра была выбрана матричная структура. IP-блок использовался как компонент системы на кристалле. Прием/передача данных и управление устройством осуществлялись посредством регистрового файла, который имеет uP-интерфейс.

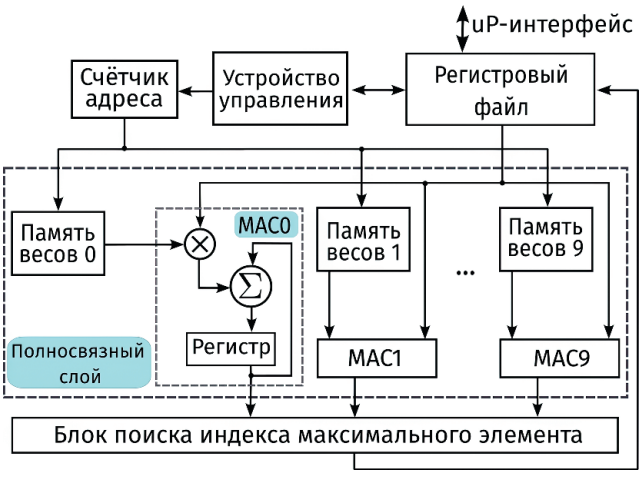

Рис. 3. Структура IP-блока нейронной сети
Fig. 3. Structure of the IP block of the neural network

По uP-интерфейсу от процессорной системы (ПС) в IP-блок последовательно поступают пиксели изображения. Значение очередного пикселя изображения подается на входы всех MAC-ядер, одновременно с этим устройство управления увеличивает значение счетчика, который указывает адрес текущего коэффициента НС, хранящегося в памяти. Каждое MAC-ядро производит 784 операции умножения значения пикселя на соответствующий весовой коэффициент НС. В результате расчета формируется массив из десяти элементов, представляющий выходные данные слоя. Далее полученный массив поступает на вход блока поиска индекса максимального элемента (на рис. 4 обозначен как «Индекс макс.»). В данном блоке происходит сравнение всех входных значений и осуществляется выбор наибольшего элемента массива, индекс которого передается на выход в качестве результата распознавания. Найденное число передается обратно в ПС, используя uP-интерфейс. Общая структура разработанной системы для распознавания рукописных цифр на базе отладочной платы Zybo Z7 представлена на рис. 4.

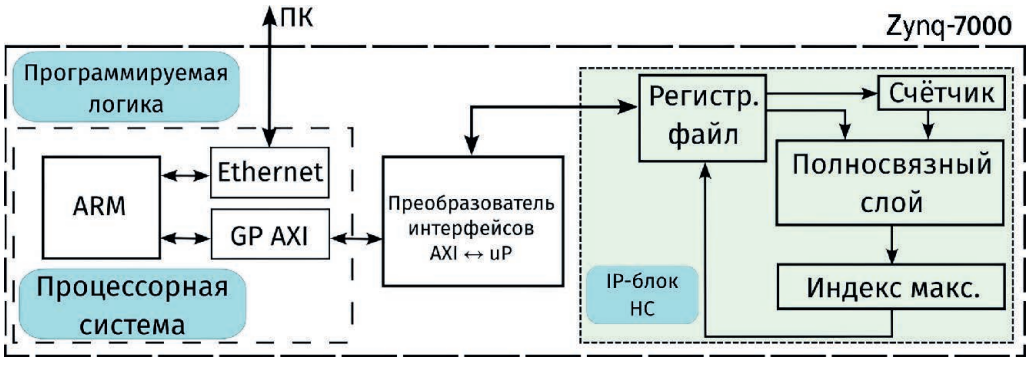

Рис. 4. Структура системы для распознавания рукописных цифр на базе отладочной платы Zybo Z7
Fig. 4. Structure of the system for handwritten digit recognition based on the Zybo Z7 debug board

Процессорная система состоит из процессора ARM Cortex-A9, контролера Ethernet для связи с персональным компьютером (ПК) и блока GP AXI для соединения по AXI-интерфейсам с другими блоками, расположенными в области программируемой логики кристалла Zynq-7000. Для соединения ПС с разработанным IP-блоком использовался преобразователь интерфейса uP в AXI4-Lite. ПС работает под управлением операционной системы Linux (PYNQ), на которой запущено ядро Jupyter Notebook. Python-библиотека PYNQ позволяет получить доступ к адресному пространству процессорной системы, на которое отображены регистры разработанного IP-блока. Таким образом, в представленном прототипе есть возможность подавать тестовые изображения непосредственно на аппаратный блок из блокнота Jupyter, что дает большую гибкость при отладке и тестировании проекта.

Экспериментальные исследования и их результаты

На этапе тестирования исследовалось влияние разрядности весовых коэффициентов на точность распознавания цифр и на аппаратные затраты FPGA. Разрядность коэффициентов НС изменялась от 2 до 16 бит. Для каждой разрядности производилась подача на НС всех 10 тыс. тестовых изображений базы MNIST. Для анализа полученных результатов выполнялось построение матрицы ошибок, которая показывает точность определения цифр в процентном соотношении. На рис. 5 представлен пример матрицы ошибок.

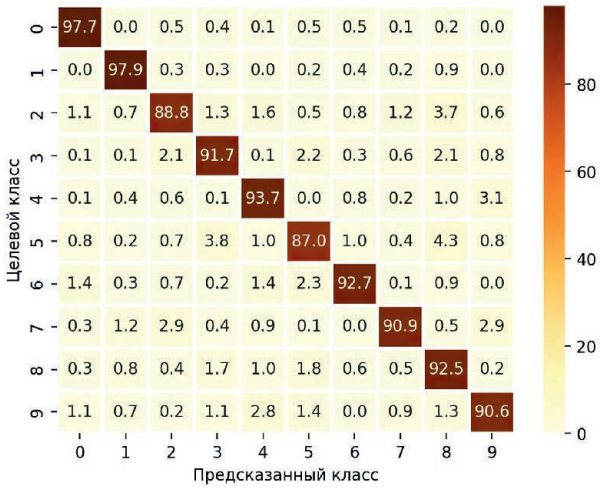

Рис. 5. Матрица ошибок для 5-разрядного представления весов нейронной сети
Fig. 5. Confusion matrix for 5-bit representation of neural network weights

На основании полученных результатов можно сделать вывод, что цифра 1 распознается НС лучше всего (точность – 97,9 %), хуже всего происходит распознавание цифры 5 (точность – 87,0 %), чаще всего НС путает цифру пять с восьмеркой (4,3 %) и тройкой (3,8 %). Общая точность распознавания – 92,4 %, что на 2,0 % больше, чем в [3], где рассматривалась FPGA-реализация сверточной НС.

Исследование аппаратных затрат осуществлялось на основе отчетов о размещении проекта на FPGA, полученных в среде Xilinx Vivado. Анализ аппаратных затрат при различной разрядности весовых коэффициентов НС показал, что при уменьшении разрядности уменьшается число требуемых для реализации НС блоков LUT (Look-Up Tables) и FF (триггеров). Полученные результаты экспериментов представлены на рис. 6, где показаны точность распознавания, количество использованных элементов LUT и FF в зависимости от разрядности коэффициентов НС.

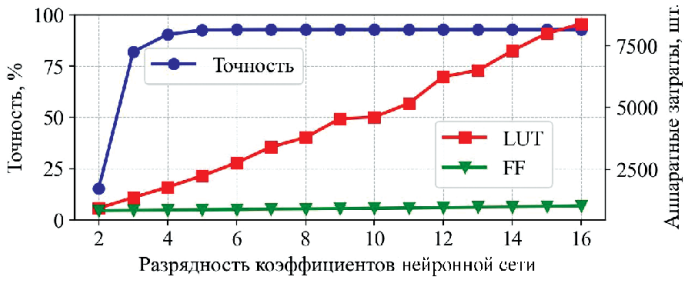

Рис. 6. Точность и аппаратные затраты на реализацию нейронной сети в зависимости от разрядности коэффициентов нейронной сети
Fig. 6. Accuracy and hardware costs for the implementation of a neural network depending on the bit depth of the neural network coefficients

Из рис. 6 видно, что с двухразрядного до пятиразрядного представления весовых коэффициентов наблюдается скачкообразный прирост в точности. Начиная с разрядности 5 и до разрядности 16 график точности принимает линейный вид, что свидетельствует об отсутствии значительных изменений в точности. График зависимости количества используемых триггеров FF от разрядности коэффициентов НС имеет вид практически горизонтальной прямой, что свидетельствует о незначительном влиянии разрядности на их количество. График зависимости ко-

личества блоков LUT постоянно растет с увеличением разрядности, что связано с увеличением размера умножителя и сумматора, которые используются в MAC-ядрах.

Затраты для платы Zybo Z7 из семейства FPGA Xilinx Zynq-7000 и пятиразрядного представления весовых коэффициентов НС приведены в табл. 1.

Таблица 1. Аппаратные затраты на реализацию нейронной сети на FPGA Zybo Z7
Table 1. Hardware costs for implementing a neural network on FPGA Zybo Z7

Вариант блока	Количество блоков, шт.		Использование, %
	использованное	доступное	
Общие затраты			
LUT как логика	2180	17 600	12,39
LUT как память	60	6000	1,00
Триггеры	862	35 200	2,45
Блочная память	10	120	8,33
Затраты на MAC-ядро			
LUT как логика	155	17 600	0,88
Триггеры	11	35 200	0,03
Индекс макс.			
LUT как логика	276	17 600	1,57
Полносвязный слой			
LUT как логика	1685	17 600	9,57
Триггеры	126	35 200	0,36
Блочная память	10	120	8,33

Для того чтобы объяснить феномен сохранения точности распознавания при уменьшении разрядности коэффициентов (рис. 6), выполняли анализ матрицы весов НС с использованием разложения шаблонов, полученных нейронной сетью в результате обучения, на битовые плоскости. Для этого каждая строка матрицы весов преобразовывалась в изображение размерами 28×28 и каждый пиксель переводился к типу uint8. Затем изображение раскладывалось на битовые плоскости. В данном случае битовая плоскость – это двоичное изображение, сформированное из набора битов, занимающих одинаковую позицию в двоичном представлении значений пикселей изображения.

Обозначим через P_3 шаблон для цифры три, полученный из четвертой строки матрицы весов W. На рис. 7, *a* показан пример такого разложения шаблона P_3, на рис. 7, *b* – вид шаблона P_3, если в нем сохранить 1, 2 и т. д. значимых разрядов, используя операцию логического «И» с соответствующей битовой маской.

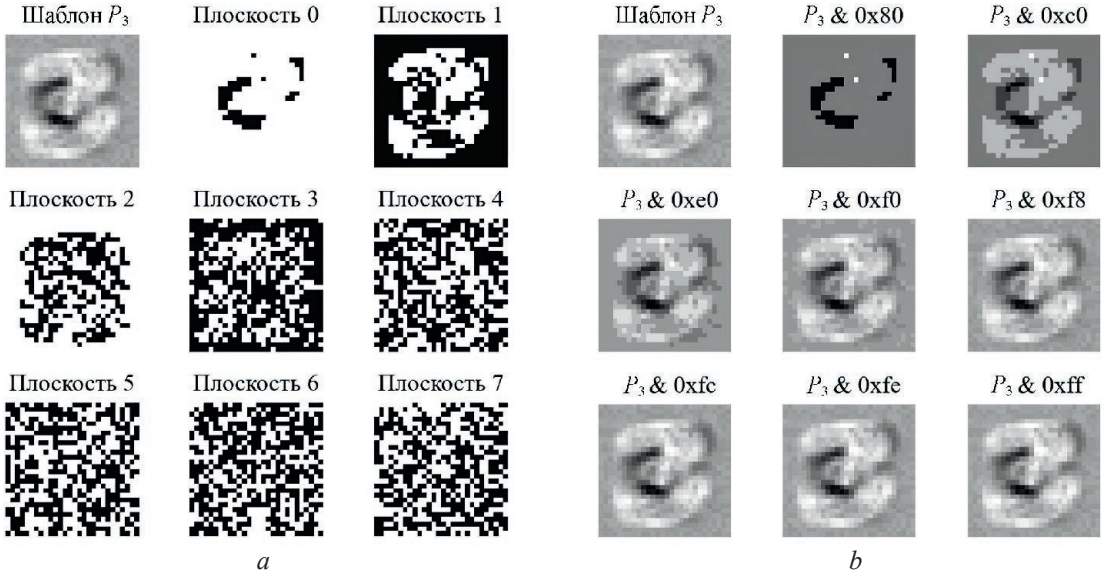

Рис. 7. Разложение весового ряда для цифры 3: *a* – на битовые плоскости; *b* – результат зануления части битовых плоскостей
Fig. 7. Decomposition of the weight series for the number 3: *a* – into bit planes; *b* – the result of zeroing out part of the bit planes

Из разложения, приведенного на рис. 7, видно, что основная информация об изображении находится в пяти первых битовых плоскостях, так как при добавлении остальных плоскостей существенные изменения на изображении не наблюдаются. Это указывает на то, что более высокая точность весовых коэффициентов будет избыточной, поскольку не несет дополнительной информации. Таким образом, наиболее оптимальной разрядностью, которая позволяет с высокой вероятностью правильно распознать цифры на изображении и не использовать избыточные аппаратные ресурсы FPGA, будет пять бит.

Заключение

1. Разработана структура IP-блока, реализующего нейронную сеть прямого распространения для распознавания рукописных цифр. Исследовано влияние разрядности весовых коэффициентов нейронной сети на точность распознавания цифр с использованием разработанного прототипа нейронной сети на базе отладочной платы Zybo Z7.

2. Для наглядной демонстрации объема информации, хранящейся в коэффициентах, использовали разложение весовых коэффициентов нейронной сети на битовые плоскости. Выполненный анализ позволяет сделать вывод о том, что шестая и последующие битовые плоскости не несут полезной информации и не влияют на точность распознавания.

3. Получены зависимости аппаратных затрат FPGA от разрядности представления коэффициентов нейронной сети. Предлагается использовать пять бит для представления весовых коэффициентов нейронной сети при реализации на базе FPGA, поскольку такая разрядность, с одной стороны, позволяет понизить аппаратные затраты, а с другой – обеспечить высокую точность распознавания.

4. Исследование выполнено в рамках работы над научным проектом в лаборатории YADRO-БГУИР в 2024/2025 учебном году.

Список литературы

1. Mittal, S. A Survey of FPGA-Based Accelerators for Convolutional Neural Networks / S. Mittal // Neural Computing and Applications. 2020. Vol. 32, No 4. P. 1109–1139.
2. Ahmad, A. FFConv: An FPGA-Based Accelerator for Fast Convolution Layers in Convolutional Neural Networks / A. Ahmad, M. A. Pasha // ACM Transactions on Embedded Computing Systems (TECS). 2020. Vol. 19, Iss. 2. P. 1–24.
3. FPGA Implementation of Hand-Written Number Recognition Based on CNN / D. Giardino [et al.] // International Journal on Advanced Science, Engineering and Information Technology. 2019. Vol. 9, No 1. P. 167–171.
4. Common Visual Data Foundation [Electronic Resource]. Mode of access: https:// https://github.com/cvdfoundation/mnist.
5. Николенко, С. Глубокое обучение. Погружение в мир нейронных сетей / С. Николенко, А. Кадурин, Е. Архангельская. СПб.: Питер, 2019.
6. Samaragh, M. Customizing Neural Networks for Efficient FPGA Implementation / M. Samaragh, M. Ghasemzadeh, F. Koushanfar // IEEE Symposium on Field-Programmable Custom Computing Machines. USA: California, 2017. P. 85–92.

Поступила 23.11.2024 Принята в печать 09.01.2025

References

1. Mittal S. (2020) A Survey of FPGA-Based Accelerators for Convolutional Neural Networks. *Neural Computing and Applications*. 32 (4), 1109–1139.
2. Ahmad A., Pasha M. A. (2020) FFConv: An FPGA-Based Accelerator for Fast Convolution Layers in Convolutional Neural Networks. *ACM Transactions on Embedded Computing Systems (TECS)*. 19 (2), 1–24.
3. Giardino D., Matta M., Silvestri F., Spano S., Trobiani V. (2019) FPGA Implementation of Hand- Written Number Recognition Based on CNN. *International Journal on Advanced Science, Engineering and Information Technology*. 9 (1), 167–171.
4. Common Visual Data Foundation. Available: https:// https://github.com/cvdfoundation/mnist.
5. Nikolenko S. I., Kadurin A. A., Arkhangelskaya E. V. (2019) *Deep Learning*. *A Dive Into the World of Networks*. Saint Petersburg, Peter Publishing House (in Russian).

6. Samaragh M., Ghasemzadeh M., Koushanfar F. (2017) Customizing Neural Networks for Efficient FPGA Implementation. *IEEE Symposium on Field-Programmable Custom Computing Machines*. USA, California. 85–92.

Received: 23 November 2024 Accepted: 9 January 2025

Вклад авторов

Кривальцевич Е. А. реализовал и обучил нейронную сеть, выполнил аппаратную реализацию нейронной сети, провел экспериментальные исследования, подготовил рукопись статьи.

Вашкевич М. И. определил задачи, которые следовало решить в ходе проведения исследований, принимал участие в аппаратной реализации нейронной сети и тестировании на FPGA, участвовал в проведении экспериментальных исследований и интерпретации результатов эксперимента. Выполнил редактирование текста статьи.

Authors' contribution

Krivaltsevich E. A. implemented and trained the neural network, performed hardware implementation of the neural network, conducted experimental studies, prepared the manuscript of the article.

Vashkevich M. I. defined the tasks that had to be solved during the research, participated in the hardware implementation of the neural network and testing on FPGA, participated in conducting experimental studies and interpreting the experimental results. Edited the text of the article.

Сведения об авторах

Кривальцевич Е. А., студент, Белорусский государственный университет информатики и радиоэлектроники

Вашкевич М. И., д-р техн. наук, проф. каф. электронных вычислительных средств, 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

Krivalcevich E. A., Student, Belarusian State University of Informatics and Radioelectronics

Vashkevich M. I., Dr. Sci. (Tech.), Professor at the Electronic Computing Facilities Department, 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 Iosifovich