http://dx.doi.org/10.35596/1729-7648-XXXX-XX-X-XX-XX

Оригинальная статья
Original paper

УДК 004.021

АППАРАТНАЯ РЕАЛИЗАЦИЯ НЕЙРОННОЙ СЕТИ ПРЯМОГО РАСПРОСТРАНЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ РУКОПИСНЫХ ЦИФР НА БАЗЕ FPGA

Е.А. КРИВАЛЬЦЕВИЧ, М.И. ВАШКЕВИЧ

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

Поступила в редакцию (дату отмечает редакция)

Аннотация. Целью работы является разработка аппаратной реализации на базе ПЛИС типа FPGA однослойной нейронной сети прямого распространения для распознавания рукописных цифр, а также исследование влияния разрядности коэффициентов сети на точность распознавания и на аппаратные затраты ПЛИС. Обучение нейронной сети выполнялось с использованием 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. В результате проведения экспериментов были получены графики точности распознавания и количества аппаратных ресурсов ПЛИС в зависимости от разрядности представления коэффициентов нейронной сети. Также 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, обеспечивают экономное расходование ресурсов ПЛИС и позволяют получить высокую точность распознавания (92,4%).

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

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Кривальцевич Е.А., Вашкевич М.И. Аппаратная реализация нейронной сети прямого распространения для распознавания рукописных цифр на базе FPGA. *Цифровая трансформация*. 20**; **(*): ***-***.

HARDWARE IMPLEMENTATION OF A DIRECT PROPAGATION 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)

Submitted

Abstract. The aim of the work is to develop a hardware implementation based on FPGA of a single-layer direct propagation neural network for handwritten digit recognition, as well as to study the effect of the bit depth of network coefficients on the recognition accuracy and FPGA hardware costs. 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. As a result of the experiments, graphs of recognition accuracy and the amount of FPGA hardware resources were obtained depending on the bit depth of neural network coefficient representation. An analysis of the coefficients obtained as a result of neural network training was also performed using decomposition into bit planes. It is shown that 5 bits are sufficient to represent neural network coefficients, since they contain the main information learned by the network, provide economical use of FPGA resources and allow for high recognition accuracy (92.4%).

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

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

For citation. Krivalcevich E.A., Vashkevich M.I. Hardware implementation of a direct propagation neural network for recognizing handwritten digits based on FPGA. *Digital transformation.* 20**; **(*): ***-***.

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 FPGA относится возможность использовать пользовательские типы данных, позволяющие контролировать точность представления параметров нейросетевой модели [1-3]. Причем выбор точности представления напрямую будет влиять на аппаратные затраты ПЛИС, необходимые для обеспечения выполнения операций над данными.

Целью данной работы является разработка аппаратной реализации однослойной НС прямого распространения для распознавания рукописных цифр, а также исследование влияния разрядности коэффициентов НС на точность распознавания и на аппаратные затраты ПЛИС.

Процесс разработки и исследование аппаратной реализации НС был разбит на несколько этапов. На первом этапе выполнялась разработка и обучение модели с использованием языка Python и библиотеки PyTorch. На втором этапе выполнялась разработка архитектуры и описание IP-блока НС с использованием языка SystemVerilog. На третьем этапе проводилось прототипирование НС на отладочной плате Zybo Z7. На заключительном этапе выполнялся эксперимент и анализ полученных результатов.

Разработка программной модели НС

В работе рассматривается задача распознавания рукописных цифр по изображениям из набора данных MNIST. Используемый набор данных MNIST содержит 70 тыс. полутоновых изображений размера 28×28 пикселей рукописных цифр от 0 до 9 [4]. Набор разбит на две части: тренировочная выборка – 60 тыс. изображений и тестовая выборка – 10 тыс. изображений.

В работе используется однослойная НС прямого распространения, состоящая из полносвязного слоя с выходной функцией активации softmax [5]. Структура НС представлена на рис.1. 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:

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

где w_{ij} – весовой коэффициент, x_j – j - й пиксель изображения, b_i – смещение, i = 0, ... 9, y_i – вероятность того, что поданное на вход изображение относится к классу i.

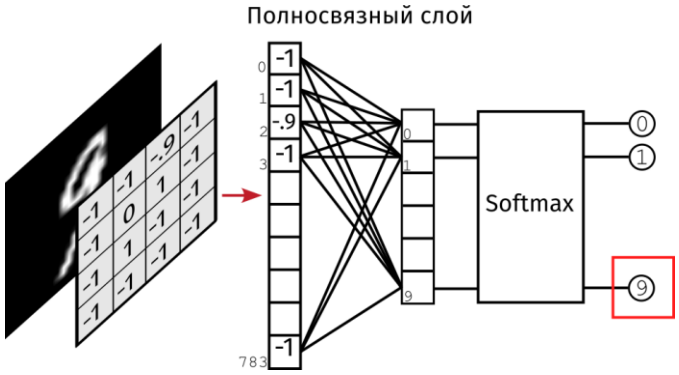


Рис. 1. Структура НС
Fig. 1. Structure of NN

Обучение НС выполнялось с использованием языка Python и библиотеки PyTorch. В процессе обучения НС использовался метод стохастического градиентного спуска [5], который имеет два настроечных параметра: скорость обучения α, а также параметр инерции γ:

$$\upsilon_i = \gamma \upsilon_{t-1} + \alpha \nabla L \quad (3)$$

$$w_t = w_{t-1} - \upsilon_t, \quad (4)$$

где υ_t – скорректированный градиент с учетом параметра инерции, ∇L – градиент функции потерь, w_{t-1} – веса НС на предыдущем шаге, w_t – веса НС на текущем шаге.

В качестве функции потерь использовалась – перекрестная энтропия:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[t_i \log(y_i) + (1-t_i)\log(1-y_i)], \quad (5)$$

где t_i – метка i-го изображения в унарном коде, N – количество изображений в базе.

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

Рис. 2. Результаты обучения НС
Fig. 2. NN training results

Аппаратная реализация НС на FPGA

На начальном этапе была разработана структура IP-блока НС, показанная на рис. 3. Вычислительной основой разработанного устройства являются десять MAC (*Multiply-ACcumulate*) ядер, выполняющих умножение вектора-изображения на матрицу весов НС. Для реализации умножителя MAC-ядра была выбрана матричная структура.

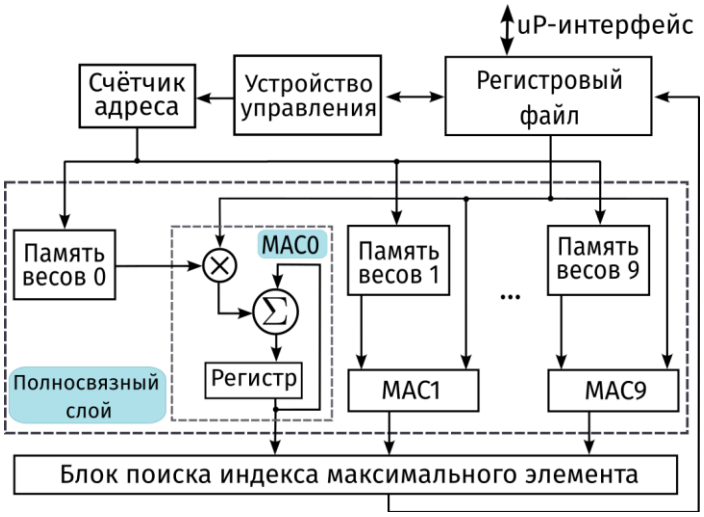


Рис. 3. Структура IP-блока НС
Fig. 3. The structure of a NN IP-block

IP-блок предполагается использовать как компонент системы на кристалле (СнК). Прием/передача данных и управление устройством осуществляется посредством регистрового файла, который имеет uP-интерфейс. По этому интерфейсу от процессорной системы (ПС) в IP-блок поступают последовательно 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. В данном блоке происходит сравнение всех входных значений и осуществляется выбор наибольшего элемента массива, индекс которого передаётся на выход в качестве результата распознавания. Найденное число передаётся обратно в ПС используя uP-интерфейс.

Общая структура разработанной системы для распознавания рукописных цифр на базе отладочной платы ZYBO-Z7 представлена на рис. 4.

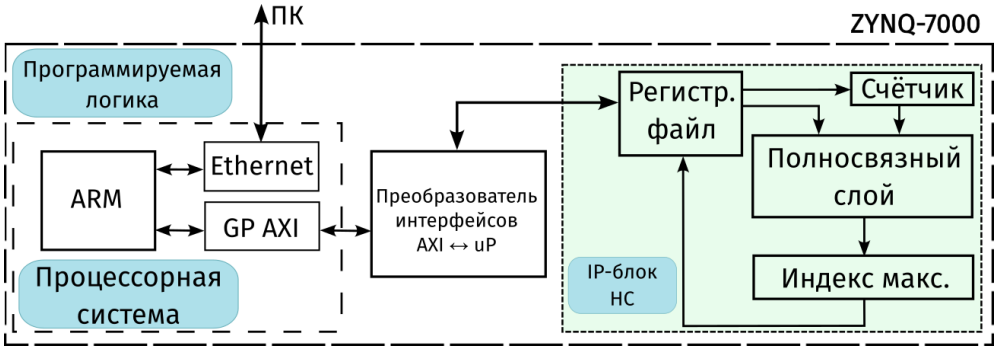


Рис. 4. Структурная реализация проекта
Fig. 4. Structural implementation of the project

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. Для соединения ПС с разработанным IP блоком используется преобразователь интерфейса uP в

AXI4-Lite. ПС работает под управлением ОС Linux (PYNQ), на котором 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 разработанном 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 выводы

На этапе тестирования исследовалось влияние разрядности весовых коэффициентов на точность распознавания цифр, а также 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 спутывания, которая показывает точность определения цифр в процентном соотношении. На рис. 5 представлен пример матрицы спутывания.

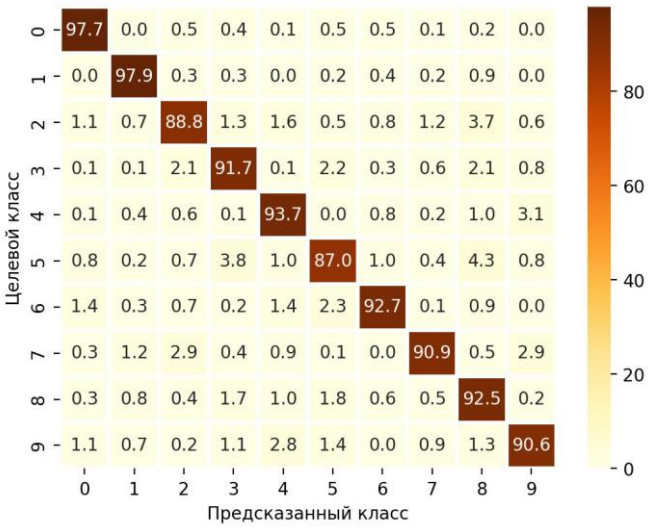


Рис. 5. Матрица спутывания для 5-разрядного представления весов НС
Fig. 5. Confusion matrix for 5-bit representation of NN weights

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 с тройкой (3,8%). Общая точность распознавания равна 92,4%, что на 2% выше, чем в работе [3], где рассматривалась FPGA-реализация сверточной НС.

Исследование аппаратных затрат осуществлялось на основе отчётов о размещение дизайна на FPGA, полученные 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 на одном графике совмещены точность распознавания и количество использованных элементов LUT и FF в зависимости от разрядности коэффициентов НС.

На графике 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 от разрядности имеет вид практически горизонтальной прямой, что свидетельствует о незначительном влиянии разрядности на их количество. График зависимости количество блоков LUT постоянно растёт с увеличением разрядности, что связано с увеличением размера умножителя и сумматора, которые реализуют блок MAC.

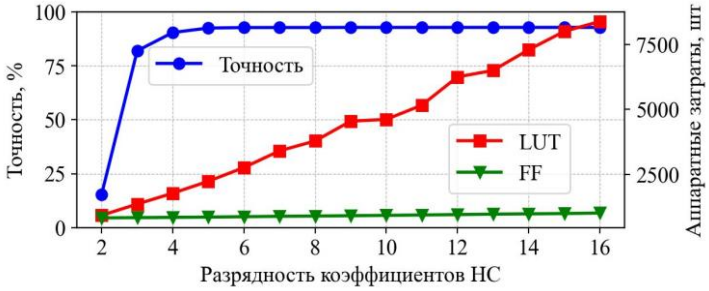


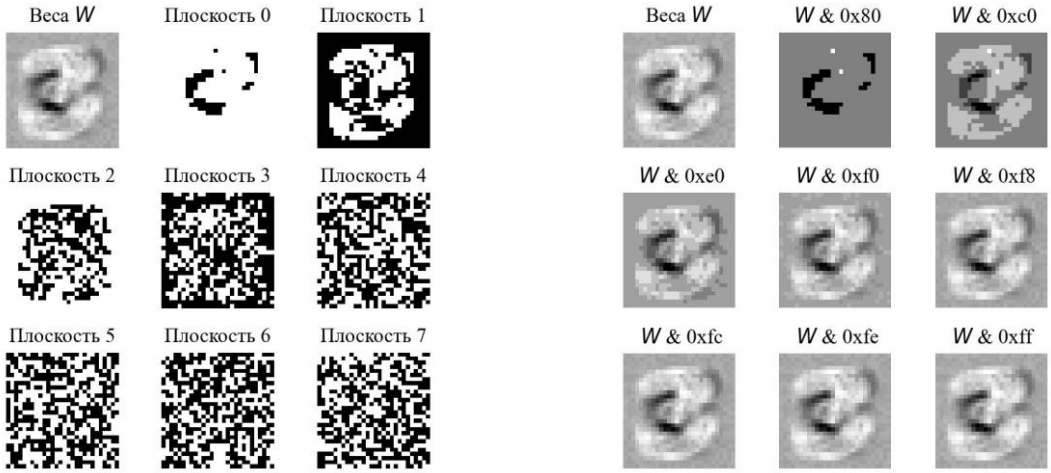
Рис. 6. Точность и аппаратные затраты на реализацию НС
Fig. 6. Accuracy and hardware cost for implementing NN

Общие затраты для платы ZYBO Z7 из семейства FPGA Xilinx Zynq-7000 и пятиразрядного представления весов НС представлены в таблице 1:

Таблица 1. Аппаратные затраты на реализацию НС на FPGA ZYBO Z7
Table 1. Hardware costs for implementation a NN on FPGA ZYBO Z7

Вариант блока Block variant	Количество Number of blocks	Доступно Available	В процентах Utility, %
LUT как логика / LUT as logic	2180	17600	12,39
LUT как память / LUT as memory	60	6000	1
Триггеры / Flip Flop	862	35200	2,45
Блочная память / BRAM18	10	120	8,33

Для того, чтобы объяснить феномен сохранения точности распознавания при уменьшении разрядности коэффициентов (рис. 6), был выполнен 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 28×28, которое затем раскладывалось на битовые плоскости. На рис. 7,а показан пример такого разложения для четвертой строки матрицы весов (т.е. для цифры три). На рис. 7,б показан вид выученного НС шаблона, если в нем оставить незануленными 1, 2 и т.д. разрядов. В результате разложения видно, что основная информация об изображении находится в пяти первых битовых плоскостях, что говорит нам о том, что более высокая точность весовых коэффициентов будет являться избыточной, т. к. не несет дополнительной информации.



а) *б)*

Рис. 7. Разложение весового ряда для цифры 3:
а) на битовые плоскости б) 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

Таким образом можно сделать вывод, что наиболее оптимальной разрядностью, которая позволяет с высокой вероятностью правильно распознать цифры на изображении и не использовать избыточные аппаратные ресурсы FPGA, будет пять бит.

Заключение

В работе разработана структура IP-блока, реализующего НС прямого распространения для распознавания рукописных цифр. Исследовано влияние разрядности весовых коэффициентов НС на точность распознавания цифр с использованием разработанного прототипа НС на базе отладочной платы Zybo-Z7. Также использовано разложение весовых коэффициентов НС на битовые плоскости, чтобы наглядно продемонстрировать объем информации хранящейся в коэффициентах. Выполненный анализ позволяет сделать вывод о том, что шестая и последующие битовые плоскости не несут полезной информации и не влияют на точность распознавания. Получены зависимости аппаратных затрат FPGA от разрядности представления коэффициентов НС. Согласно результатам экспериментов предлагается использовать пять бит для представления весовых коэффициентов НС при реализации на базе FPGA, поскольку такая разрядность с одной стороны позволяет понизить аппаратные затраты, а с другой стороны обеспечить высокую точность распознавания.

Благодарности

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

Список литературы

1. Mittal S. A survey of FPGA-based accelerators for convolutional neural networks // Neural computing and applications. – 2020. – Т. 32. – No. 4. – P. 1109-1139
2. Ahmad A., Pasha M. A. FFConv: an FPGA-based accelerator for fast convolution layers in convolutional neural networks // ACM Transactions on Embedded Computing Systems (TECS). – 2020. – Т. 19. – No. 2. – P. 1-24.
3. Giardino D. et al. FPGA implementation of hand- written number recognition based on CNN // International Journal on Advanced Science, Engineering and Information Technology. – 2019. – Т. 9. – No. 1. – P. 167-171.
4. The MNIST database of handwritten digits [Электронный ресурс] – Электронные данные – Режим доступа: https://yann.lecun.com/exdb/mnist/
5. Николенко С.И. Глубокое обучение / С.И. Николенко, Кадурин А.А., Архангельская Е.В. – СПб.: Питер,2019. – 480 с.
6. Samaragh M. Customizing neural networks for efficient FPGA implementation //IEEE computer society // Annual International Symposium on Field-rogrammable Custom Computing Machines. – 2017.– P. 85-92.

References

1. S. A survey of FPGA-based accelerators for convolutional neural networks // Neural computing and applications. – 2020. – Т. 32. – No. 4. – P. 1109-1139
2. Ahmad A., Pasha M. A. FFConv: an FPGA-based accelerator for fast convolution layers in convolutional neural networks // ACM Transactions on Embedded Computing Systems (TECS). – 2020. – Т. 19. – No. 2. – P. 1-24.
3. Giardino D. et al. FPGA implementation of hand- written number recognition based on CNN // International Journal on Advanced Science, Engineering and Information Technology. – 2019. – Т. 9. – No. 1. – P. 167-171.

4. MNIST Handwritten Numbers database [Electronic resource] – Electronic data – Access mode: https://yann.lecun.com/exdb/mnist/Mittal
5. Nikolenko S.I. Deep learning / S.I. Nikolenko, Kadurin A.A., Arkhangelskaya E.V. – St. Petersburg: St. Petersburg, 2019. – 480 p.
6. Samaragh M. Customizing neural networks for efficient FPGA implementation //IEEE computer society // Annual International Symposium on Field-rogrammable Custom Computing Machines. – 2017.– P. 85-92.

Вклад авторов

Кривальцевич Е.А. реализовал и обучил НС, аппаратно реализовал структуру НС, а также провел экспериментальные исследования.

Вашкевич М.И. определил задачи, которые необходимо было решить в ходе проведения исследований, принимал участие в аппаратной реализации НС и тестировании на FPGA, участвовал в проведении экспериментальных исследований и интерпретации результатов эксперимента.

Authors contribution

Krivalcevich E.A. implemented and trained the NN, implemented the NN structure in hardware, and conducted experimental studies.

Vashkevich M.I. defined the tasks needed to be solved during the research, participated in the hardware implementation of the NN and testing on FPGA, participated in conducting experimental studies and interpreting the experimental results.

Сведения об авторах

Кривальцевич Е.А., студент 4 курса по специальности электронные вычислительные средства, кафедра электронных вычислительных средств, Белорусский государственный университет информатики и радиоэлектроники.

Вашкевич М.И. доктор техн. наук, профессор кафедры электронных вычислительных средств, Белорусский государственный университет информатики и радиоэлектроники.

Адрес для корреспонденции

220013, Республика Беларусь,
г. 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., 4th year student, specializing in electronic computing, Department of Electronic Computing, Belarusian State University of Informatics and Radioelectronics.

Vashkevich M.I. Dr. of 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