

17. Avilov V.I., Zhavoronkov L.G., Polupanov N.V., Khakhulin D.A., Smirnov V.A. Sinapticheskie ustroystva dlya neyromorfnykh sistem robototekhnicheskikh kompleksov [Synaptic devices for neuromorphic systems of robotic complexes], *Perspektivnye sistemy i zadachi upravleniya: Mater. XVIII Vserossiyskoy nauchno-prakticheskoy konferentsii i XIV molodezhnoy shkoly-seminara* [Prospective systems and control problems: Proceedings of the XVIII All-Russian scientific and practical conference and XIV youth school-seminar], 2023, pp. 169-173.
18. Tominov R.V., Avilov V.I., Chernenko N.E., Smirnov V.A. Issledovanie memristornogo efekta tonkoy plenki oksida titana dlya iskusstvennykh neyropodobnykh sistem [Study of the memristor effect of a thin titanium oxide film for artificial neuron-like systems], *Sb. trudov XIII Vserossiyskoy konferentsii molodykh uchenykh «Nanoelektronika, nanofotonika i nelineynaya fizika* [Collection of works of the XIII All-Russian conference of young scientists "Nanoelectronics, nanophotonics and nonlinear physics"], 2018, pp. 318-319.
19. Karen'kih O.G., Avilov V.I., Smirnov V.A., Fedotov A.A., Sharapov N.A. and Polupanov N.A. Modeling of local anodic oxidation of titanium oxide nanostructures formation process. *IOP Conf. Series: Materials Science and Engineering*, 2018, Vol. 443, 012013.
20. Avilov Vadim I., Tominov Roman V., Vakulov Zakhar E., Zhavoronkov Lev G., and Smirnov Vladimir A. Titanium oxide artificial synaptic device: Nanostructure modeling and synthesis, memristive crossbar fabrication, and resistive switching investigation, *Nano Research*, 2023, Vol. 16, pp. 10222-10233.
21. Avilov Vadim I., Tominov Roman V., Vakulov Zakhar E., Rodriguez Daniel J., Polupanov Nikita V., Smirnov Vladimir A. Nanoscale Titanium Oxide Memristive Structures for Neuromorphic Applications: Atomic Force Anodization Techniques, Modeling, Chemical Composition, and Resistive Switching Properties, *Nanomaterials*, 2025, Vol. 15 (1), 75.
22. Karen'kih O.G., Avilov V.I., Smirnov V.A., Sharapov N.A., Polupanov N.A. Modeling of titanium oxide nanostructures formation process by local anodic oxidation, *Abstract Book of International Conference "Scanning Probe Microscopy"*, 2018, 97.

Авилов Вадим Игоревич – Южный федеральный университет; e-mail: avilovvi@sfedu.ru; г. Таганрог, Россия; к.т.н.; доцент.

Душина Людмила Андреевна – Южный федеральный университет; e-mail: ldushina@sfedu.ru; г. Таганрог, Россия; студент.

Полупанов Никита Валерьевич – Южный федеральный университет; e-mail: npolupanov@sfedu.ru; г. Таганрог, Россия; студент.

Смирнов Владимир Александрович – Южный федеральный университет; e-mail: vasmirnov@sfedu.ru; г. Таганрог, Россия; к.т.н.; зав. кафедрой РТЭН.

Avilov Vadim Igorevich – Southern Federal University; e-mail: avilovvi@sfedu.ru; Taganrog, Russia; cand. of eng. sc.; associate professor.

Dushina Ludmila Andreevna – Southern Federal University; e-mail: ldushina@sfedu.ru; Taganrog, Russia; student.

Polupanov Nikita Valerievich – Southern Federal University; e-mail: npolupanov@sfedu.ru; Taganrog, Russia; student.

Smirnov Vladimir Aleksandrovich – Southern Federal University; e-mail: vasmirnov@sfedu.ru; Taganrog, Russia; cand. of eng. sc.; head of the department.

УДК 004.3, 004.9

DOI 10.18522/2311-3103-2025-5-214-229

Э.В. Мельник, Д.Е. Блох, А.И. Безмельцев, В.С. Панищев, С.Н. Полторацкий
ПРОЕКТИРОВАНИЕ МОДУЛЕЙ НЕЙРОСЕТЕЙ MLP И CNN НА ПЛИС
ДЛЯ ЗАДАЧ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ

Актуальность. Развитие методов машинного обучения и архитектур нейронных сетей, а также их распространение в различные сферы промышленности обуславливают актуальность решения задач по их аппаратной реализации. Использование программируемых логических интегральных схем в этой области позволит повысить скорость обработки данных и адаптивность реализуемых алгоритмов. Однако проектирование нейросетевых архитектур на программируе-

мых логических интегральных схемах сопряжено с рядом методологических и технических сложностей, включая оптимизацию параллельных вычислений, управление аппаратными ресурсами и обеспечение работы в условиях ограниченных вычислительных ресурсов. **Цель работы** – анализ и сравнение двух архитектур нейронных сетей, многослойного перцептрона (MLP) и сверточной нейронной сети (CNN), в контексте их аппаратной реализации на программируемых логических интегральных схемах (ПЛИС). Особое внимание уделяется компромиссу между точностью классификации и эффективностью использования ограниченных аппаратных ресурсов ПЛИС. **Методы исследования.** Для достижения цели была проведена разработка и симуляция двух модулей на ПЛИС Virtex 7, перцептронного и сверточного. Использовался набор данных MNIST, уменьшенный до 20×20 пикселей. Реализация включала этапы квантования параметров до фиксированного формата 16:16, оптимизацию гиперпараметров, применение табличных вычислений для нелинейных функций и оценку использования ресурсов ПЛИС. **Результаты и обсуждения.** MLP достиг точности 93% при использовании 11% логических элементов, в то время как CNN обеспечила точность 98%, но потребовала существенно больше ресурсов. Использование внутренних буферов для хранения промежуточных данных в CNN привело к превышению допустимых ресурсов. Вынужденный переход к внешней памяти увеличил задержки и объем портов ввода-вывода. **Выводы.** Исследование показало, что выбор архитектуры зависит от приоритетов: CNN обеспечивает лучшую точность, но менее эффективна в ресурсах. Для embedded-систем с ограничениями по памяти и потреблению энергии предпочтительна упрощенная MLP-реализация. Основными проблемами остаются нехватка внутренней памяти и высокая ресурсоемкость операций, что требует дальнейших исследований в области аппаратной оптимизации и адаптивного управления вычислениями.

ПЛИС; нейронные сети; сверточные сети; многослойный перцептрон; квантование; аппаратная реализация; embedded-системы.

E.V. Melnik, D.E. Blokh, A.I. Bezmeltsev, V.S. Panishchev, S.N. Poltoratsky

DESIGNING MLP AND CNN NEURAL NETWORK MODULES ON FPGA FOR IMAGE CLASSIFICATION TASKS

Relevance. The development of machine learning methods and neural network architectures, as well as their spread into various industrial sectors, determine the relevance of solving problems related to their hardware implementation. The use of programmable logic integrated circuits in this area will increase data processing speed and the adaptability of the implemented algorithms. However, designing neural network architectures on programmable logic integrated circuits is associated with a number of methodological and technical difficulties, including the optimization of parallel computing, hardware resource management, and ensuring operation under conditions of limited computing resources. **The purpose of this work** is to analyze and compare two neural network architectures, the multilayer perceptron (MLP) and the convolutional neural network (CNN), in the context of their hardware implementation on programmable logic integrated circuits (PLICs). Particular attention is paid to the trade-off between classification accuracy and the efficient use of limited FPGA hardware resources. **Research methods.** To achieve the goal, two modules were developed and simulated on a Virtex 7 FPGA, a perceptron and a convolutional module. The MNIST dataset, reduced to 20×20 pixels, was used. The implementation included quantizing parameters to a fixed 16:16 format, optimizing hyperparameters, using tabular computations for nonlinear functions, and evaluating FPGA resource usage. **Results and discussions.** MLP achieved 93% accuracy using 11% of logic elements, while CNN achieved 98% accuracy but required significantly more resources. The use of internal buffers to store intermediate data in CNN resulted in exceeding the allowable resources. The forced transition to external memory increased delays and the number of I/O ports. **Conclusions.** The study showed that the choice of architecture depends on priorities: CNN provides better accuracy but is less resource-efficient. For embedded systems with memory and power consumption constraints, a simplified MLP implementation is preferable. The main problems remain the lack of internal memory and the high resource intensity of operations, which requires further research in the field of hardware optimization and adaptive computation control.

FPGA; neural networks; convolutional networks; multilayer perceptron; quantization; hardware implementation; embedded systems.

Введение. Активное развитие методов машинного обучения и их интеграция в аппаратные платформы обусловили повышенный интерес к реализации нейронных сетей на программируемых логических интегральных схемах (ПЛИС) [1, 2]. Данное направление исследований продиктовано требованиями к повышению энергоэффективности, снижению задержек при обработке данных и адаптивности алгоритмов в условиях ограниченных вычислительных ресурсов [3, 4]. Однако проектирование нейросетевых архитектур на ПЛИС сопряжено с рядом методологических и технических сложностей, включая оптимизацию параллельных вычислений, управление аппаратными ресурсами [5, 6] и обеспечение соответствия между абстрактными математическими моделями и их физической реализацией.

В рамках настоящего исследования рассматривается задача разработки модуля распознавания цифр, являющейся типичным примером классификации изображений, для которой проведено сравнительное моделирование двух архитектур: многослойного перцептрона и сверточной нейронной сети. Выбор указанных моделей обусловлен необходимостью анализа принципиально различных подходов к обработке данных: перцептрон, основанный на полносвязных слоях, демонстрирует зависимость от глобальных признаков, тогда как сверточная сеть акцентирует внимание на локальных пространственных закономерностях. Сопоставление данных архитектур в контексте их аппаратной реализации позволяет выявить специфические ограничения, связанные с распределением логических элементов, использованием памяти и синхронизацией потоков вычислений. Последующее изложение сфокусировано на методологии преобразования программных моделей нейросетей в аппаратные описания на языке HDL, оценке эффективности использования ресурсов ПЛИС и анализе компромиссов между точностью классификации и быстродействием системы.

Постановка задачи. Эффективная реализация нейросетевых моделей на программируемых логических интегральных схемах определяется не только алгоритмической сложностью архитектур, но и ограничениями аппаратных ресурсов. В условиях embedded-систем требуется достижение компромисса между точностью классификации и эффективностью использования логики, памяти и вычислительных блоков.

Основная задача настоящей работы заключается в проектировании и сравнительном анализе модулей MLP и CNN для задачи классификации изображений, исследовании влияния архитектурных решений на показатели точности и ресурсоемкости, а также в разработке схемотехнических приёмов, позволяющих минимизировать использование памяти и сократить задержки обработки при сохранении приемлемого уровня классификации.

Методы решения. В качестве базовой модели для распознавания цифр был выбран многослойный перцептрон, обрабатывающий входные данные в виде одномерного вектора, полученного из изображения размером 20×20 пикселей в градациях серого. Преобразование двумерного изображения в векторную форму, несмотря на потерю пространственной информации, позволяет упростить реализацию полносвязных слоев, что критически важно для последующего переноса алгоритма на ПЛИС [7]. Архитектура сети включала входной слой (400 нейронов), скрытый слой переменного размера и выходной слой (10 нейронов), соответствующий количеству классов.

Экспериментальная часть исследования была направлена на определение оптимальных гиперпараметров, включая функцию активации скрытого и выходного слоев, размер скрытого слоя и количество эпох обучения [8]. Первоначально анализировалось влияние количества эпох на сходимость модели. Как показано на рисунке 1, зависимость точности и потерь от числа эпох демонстрирует, что без использования смещения (bias) в скрытом слое и функции активации ReLU модель достигает плато точности уже после пяти итераций, в то время как наличие смещения приводит к замедленной сходимости и колебаниям метрик. Данный результат подтвердил целесообразность ограничения числа эпох для минимизации вычислительных затрат без существенного ухудшения качества классификации.

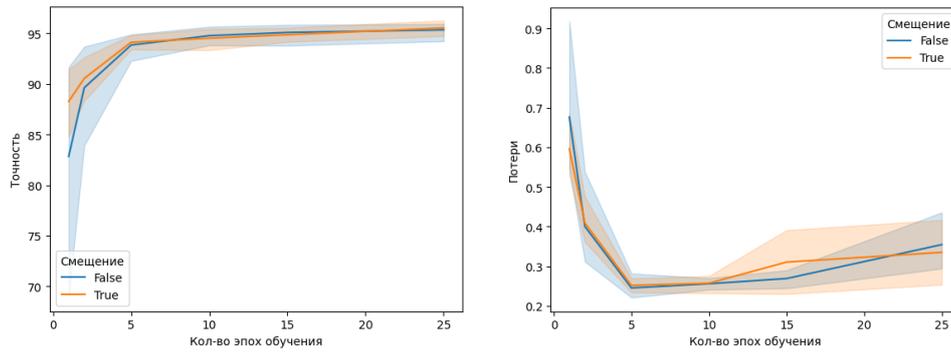


Рис. 1. Зависимость точности и потери от количества эпох обучения для сетей со смещением и без. Функция активации скрытого ReLU. Функция активации выходного Softmax

Далее исследовалось влияние размера скрытого слоя на обобщающую способность модели. Согласно данным, представленным на рис. 2, увеличение числа нейронов свыше 96 единиц не приводит к статистически значимому росту точности, однако пропорционально повышает требования к объему памяти и количеству логических элементов [9, 10] на ПЛИС. При сокращении скрытого слоя ниже 64 нейронов наблюдается резкое снижение точности, обусловленной недостаточной емкостью модели для выделения признаков. Таким образом, выбор скрытого слоя из 96 нейронов представляет собой компромисс между производительностью и аппаратной сложностью.

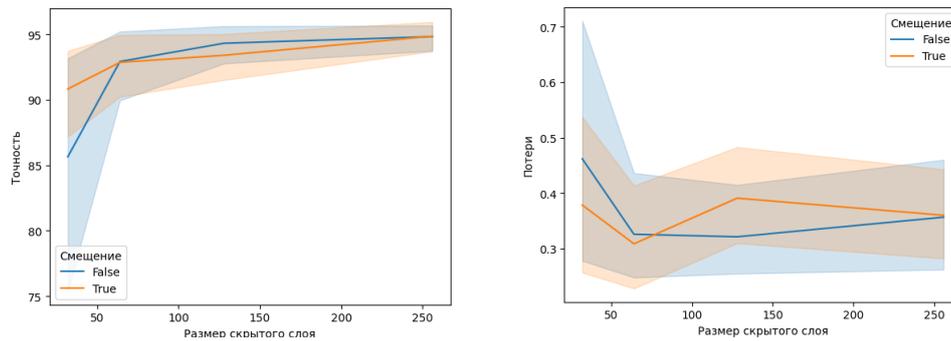


Рис. 2. Зависимость точности и потери от размера скрытого слоя для сетей со смещением (bias) и без. Функция активации скрытого ReLU. Функция активации выходного Softmax

Отдельное внимание уделялось анализу функций активации. Сравнение ReLU, сигмоиды и гиперболического тангенса в скрытом слое (рис. 3) выявило преимущество ReLU в контексте скорости обучения и устойчивости к проблеме затухающих градиентов [11, 12]. Для выходного слоя, как продемонстрировано на рис. 3 и 4, функция Softmax обеспечила более стабильную сходимость по сравнению с линейной активацией, благодаря нормализации выходных значений в вероятностное распределение.

Итоговая архитектура, сформированная на основе проведенных экспериментов, включает скрытый слой из 96 нейронов с функцией активации ReLU и выходной слой с Softmax, обученная за 5 эпох с достижением точности 93% на тестовой выборке. Ключевым выводом является отсутствие существенного прироста точности при усложнении модели, что аргументирует выбор минимально достаточной конфигурации, адаптированной под ограничения ПЛИС. Сокращение числа параметров и использование аппаратно-эффективных функций активации позволяют снизить задержки и энергопотребление, сохраняя приемлемый уровень классификации, что соответствует требованиям embedded-систем.

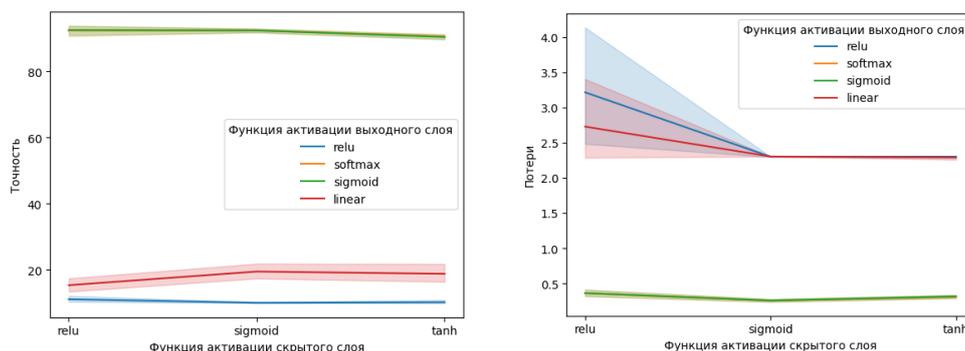


Рис. 3. Зависимость точности и потери от выбора функции активации скрытого слоя

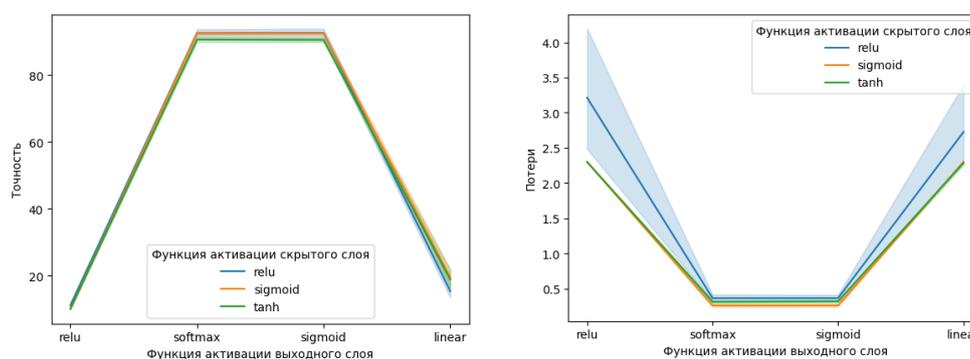


Рис. 4. Зависимость точности и потери от выбора функции активации выходного слоя

Сверточные нейронные сети (СНС) представляют собой альтернативный подход к задачам распознавания изображений, основанный на автоматическом выделении локальных пространственных признаков. В отличие от многослойного перцептрона, преобразующего входные данные в векторную форму, СНС сохраняет топологию изображения, что позволяет эффективно идентифицировать структурные паттерны [13, 14]. Исследуемая архитектура включает два сверточных слоя с ядром 3×3 , два подвыборочных слоя с фактором 2, а также полносвязные слои с 576 и 96 нейронами, что обеспечило точность 98% при обучении в течение 5 эпох.

Эксперименты по оптимизации гиперпараметров проводились на эталонном наборе данных MNIST, масштабированном до разрешения 20×20 пикселей. Первоначальный анализ влияния количества эпох на сходимость модели (рис. 5) выявил, что без использования смещения в сверточных слоях в сочетании с функцией активации ReLU позволяет достичь насыщения точности уже к пятой эпохе, тогда как наличие смещения приводит к нестабильному градиентному спуску. Данный результат согласуется с наблюдениями, полученными для перцептрона, подтверждая универсальность ограничения числа эпох для сокращения вычислительных затрат.

Важным аспектом проектирования стала оптимизация размера полносвязного слоя, следующего за сверточными блоками. Как показано на рис. 6, уменьшение числа нейронов ниже 96 единиц приводит к потере способности модели обобщать сложные признаки, извлеченные на предыдущих этапах. Увеличение же размера слоя свыше 128 нейронов не обеспечивает значимого прироста точности, но пропорционально увеличивает требования к памяти ПЛИС [15, 16]. Таким образом, выбор промежуточного значения в 96 нейронов обоснован балансом между емкостью модели и аппаратной реализуемостью.

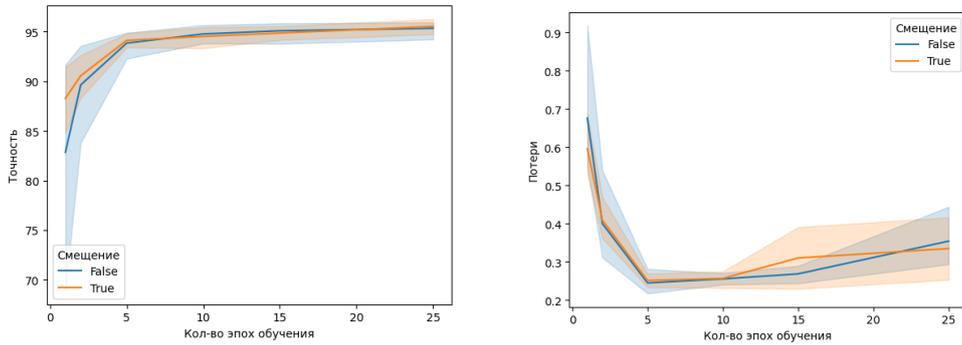


Рис. 5. Зависимость точности и потери от количества эпох обучения для сетей со смещением и без. Функция активации скрытого ReLU. Функция активации выходного Softmax

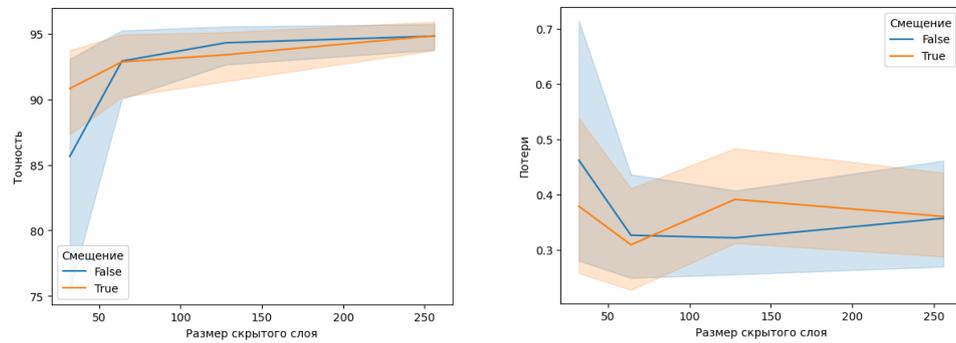


Рис. 6. Зависимость точности и потери от размера скрытого слоя для сетей со смещением (bias) и без. Функция активации скрытого ReLU. Функция активации выходного Softmax

Сравнение функций активации для скрытого слоя (рис. 7) подтвердило преимущество ReLU перед сигмоидой и гиперболическим тангенсом в контексте скорости обучения и устойчивости к затуханию градиентов. Для выходного слоя, как продемонстрировано на рис. 8, функция Softmax обеспечила стабильную сходимость за счет нормализации выходных значений, тогда как линейная активация приводила к неконтролируемому росту абсолютных величин, затрудняющему аппаратную реализацию.

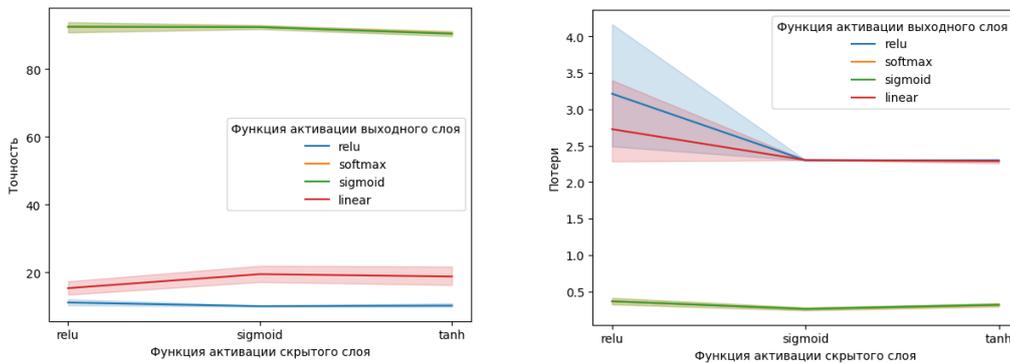


Рис. 7. Зависимость точности и потери от выбора функции активации скрытого слоя

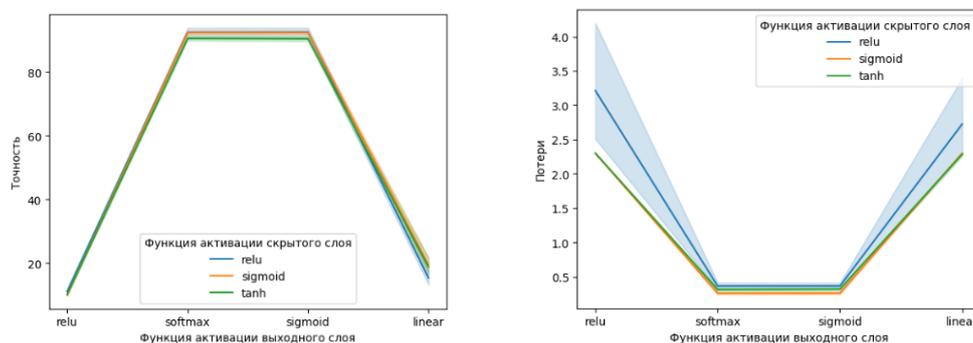


Рис. 8. Зависимость точности и потери от выбора функции активации выходного слоя

Итоговая архитектура включает два сверточных слоя с ядром 3×3 , 32 и 64 фильтра соответственно, каждый из которых сопровождается операцией макс-пулинга 2×2 , что сокращает пространственные размеры признаковых карт. Данная конфигурация обеспечивает сохранение ключевых признаков при минимизации объема данных, передаваемых на полносвязный слой из 576 нейронов, который, в свою очередь, проецируется на скрытый слой с 96 нейронами.

Общий вывод экспериментальной части заключается в том, что увеличение глубины сети, добавление дополнительных сверточных слоев или расширение ядер свертки свыше 3×3 не приводят к статистически значимому улучшению точности, но существенно усложняют аппаратную реализацию. Снижение разрешения входного изображения до 20×20 пикселей компенсируется способностью сверточных слоев к инвариантному выделению признаков, что делает предложенную архитектуру оптимальной для развертывания на ПЛИС.

Переход от программной модели к аппаратной реализации нейросети на программируемых логических интегральных схемах потребовал адаптации алгоритма под ограничения целевой платформы. Ключевым этапом стало квантование весовых коэффициентов и активаций обученной модели до формата с фиксированной точкой 16:16, что обеспечило баланс между точностью вычислений и эффективным использованием ресурсов ПЛИС. Данный подход минимизировал ошибки округления при сохранении приемлемой разрядности для представления динамического диапазона значений, характерного для функций активации ReLU и Softmax.

Структурная организация модуля, представленная на рис. 9, включает три основных компонента: входной слой, скрытый слой и вычисление softmax [17, 18]. Входными сигналами системы являются тактовый импульс и сигнал сброса, синхронизирующие этапы обработки и обеспечивающие корректную инициализацию регистров. Выходные сигналы содержат предсказанную цифру, закодированную четырьмя битами, значение максимальной активации выходного слоя в формате 0:16, а также двухбитный индикатор состояния, отражающий текущую фазу вычислений.

Алгоритм работы модуля реализует детерминированную последовательность состояний, управляемую конечным автоматом. На этапе инициализации активируется сброс внутренних буферов, а входное изображение преобразуется в одномерный вектор. Последующая обработка скрытого слоя предполагает параллельное вычисление взвешенных сумм для групп нейронов, что достигается за счет оптимизации распределения умножителей и сумматоров. Функция активации ReLU аппаратно реализована через условное обнуление отрицательных значений, что исключает необходимость использования сложных математических блоков.

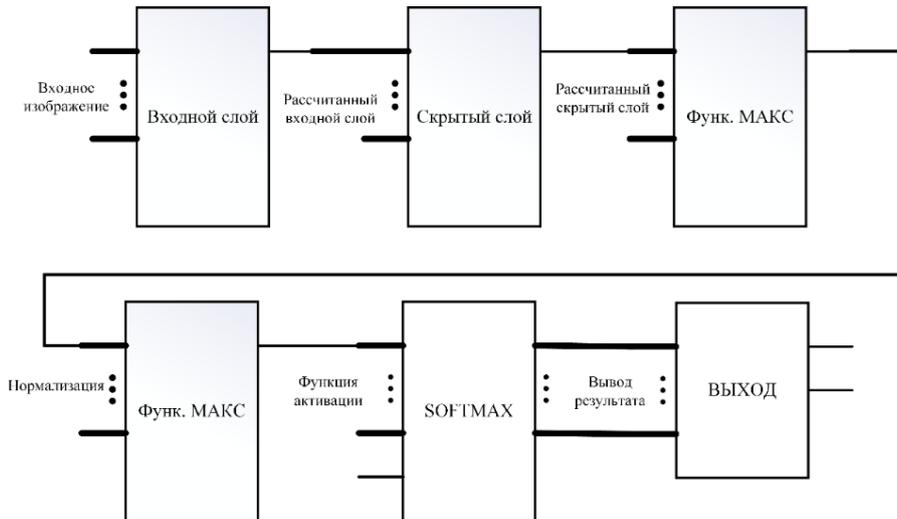


Рис. 9. Структурное представление разработанной архитектуры модуля

Переход к обработке выходного слоя инициирует вычисление активаций с последующей нормализацией через функцию Softmax. Для устранения риска числового переполнения при расчете экспоненциальных значений применяется предварительное вычитание максимальной активации из всех элементов выходного слоя. Экспоненцирование реализовано через поиск в предвычисленной LUT-таблице, охватывающей диапазон от -10 до 10 с шагом 0.1, что обеспечило погрешность менее 0,1% при сокращении аппаратных затрат по сравнению с последовательными алгоритмами приближения. Нормализованные вероятности получаются путем деления каждого экспоненциального значения на их сумму, рассчитанную с использованием конвейерного сумматора.

На завершающем этапе определяется индекс нейрона с максимальной вероятностью, который кодируется в четырехбитный выход. Значение сохраняется для последующего анализа достоверности предсказания, что актуально в системах с повышенными требованиями к надежности.

Предложенная архитектура демонстрирует эффективное использование ресурсов ПЛИС за счет конвейеризации операций, минимизации объема памяти для весовых коэффициентов и замены ресурсоемких операций (например, экспоненцирования) на табличные вычисления. Ключевым компромиссом стала незначительная деградация точности (с 98% до 91%), обусловленная квантованием, однако данное снижение компенсируется сокращением задержки обработки. Результаты подтверждают применимость подхода для embedded-систем, где критически важны энергоэффективность и детерминированное время отклика.

Реализация сверточной нейронной сети на программируемых логических интегральных схемах сопряжена с необходимостью адаптации пространственно-ориентированных операций, таких как свертка и пулинг, к последовательной потоковой обработке [19, 20]. Как и для перцептронной модели, было произведено квантование параметров модели до формата с фиксированной точкой 16:16 для минимизации использования ресурсов ПЛИС. Структурная схема модуля, представленная на рис. 10.

Входные сигналы модуля, включая тактовый импульс и сигнал сброса, синхронизируют работу конечного автомата, управляющего девятью состояниями обработки. Выходные сигналы сохраняют формат, аналогичный реализации перцептрона, что обеспечивает совместимость модулей в составе комплексных систем.

Алгоритм работы начинается с подачи входного изображения 20×20 во входную область памяти модуля. На этапе инициализации активируется сброс буферов. Обработка первого сверточного слоя реализована через скользящее окно 3×3 , вычисляющее взве-

шенную сумму для каждого положения ядра. Для ускорения вычислений коэффициенты свертки хранятся в блоке ROM, а частичные суммы аккумулируются с использованием конвейерных сумматоров. Результаты свертки передаются в буфер, где макс-пулинг с фактором 2 выполняется путем параллельного сравнения значений в окне 2×2, что сокращает пространственные размеры признаков карт в четыре раза.

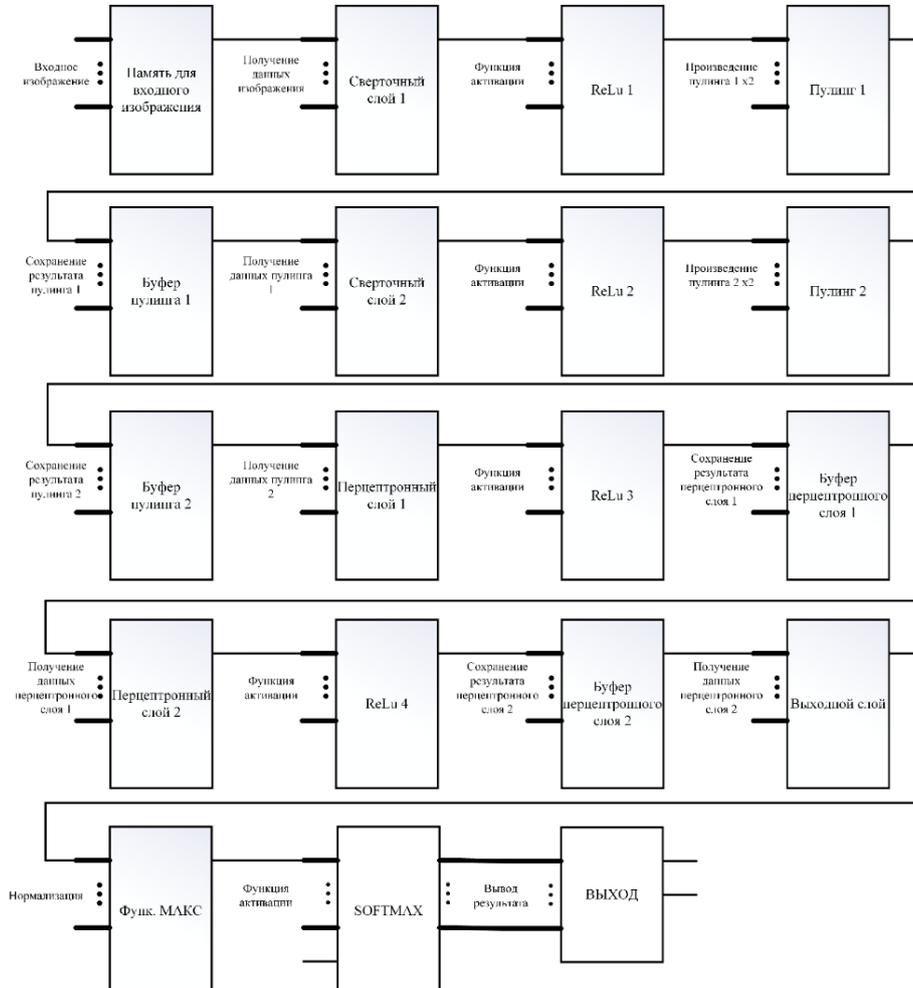


Рис. 10. Структурное представление разработанной архитектуры модуля сверточной нейронной сети

Последовательная обработка второго сверточного слоя и пулинга повторяет описанные этапы, но с увеличенным числом фильтров (64 ядра), что требует расширения буферов памяти для промежуточных данных. Переход к полносвязному слою сопровождается преобразованием трехмерных признаков карт в одномерный вектор, который обрабатывается через матрично-векторные умножения с использованием блочной обработки для экономии ресурсов. Аппаратная реализация ReLU в скрытом слое сводится к условному обнулению отрицательных значений, что исключает необходимость сложных вычислений.

Функция Softmax реализована по аналогии с перцептроном: предвычитание максимальной активации, табличное экспоненцирование и нормировка. Завершающий этап фиксирует индекс нейрона с максимальной вероятностью, формируя выходной сигнал.

Сравнение с реализацией перцептрона выявило, что сверточная сеть обеспечивает более высокую точность за счет автоматического выделения признаков, однако требует в несколько раз больше ресурсов, что аргументирует выбор архитектуры в зависимости от приоритетов задачи.

Результаты и обсуждение. В качестве платформы для реализации модуля была использована ПЛИС Virtex 7 XC7VX485T-FFG1157-1. Для вычисления производительности системы была произведена симуляция разработанного модуля при частоте тактирования в 100 МГц. В качестве входных данных используется изображение 20 на 20 пикселей оттенков серого в байтовом представлении. Они были взяты из набора для тестирования и изменены с помощью наложения шума и размытия (рис. 11).

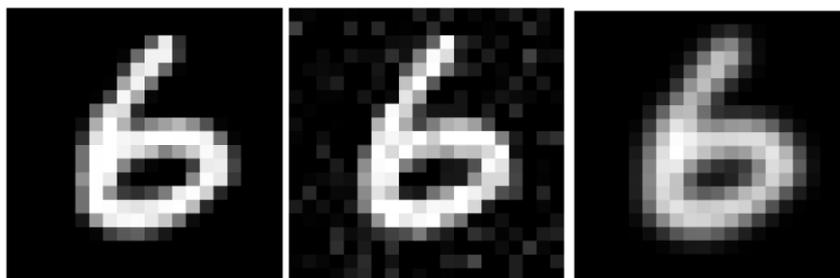


Рис. 11. Тестовые изображения для проверки разработанных модулей

На рис. 12 представлен результат симуляции перцептронного модуля.

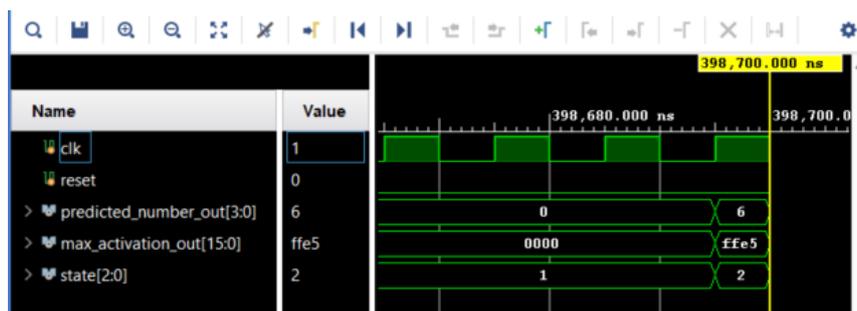


Рис. 12. Результат симуляции перцептронного модуля для тестовых изображений

После синтеза модуля на выбранной ПЛИС, было получено значение использования ресурсов, представленных на рис. 13 и табл. 1.

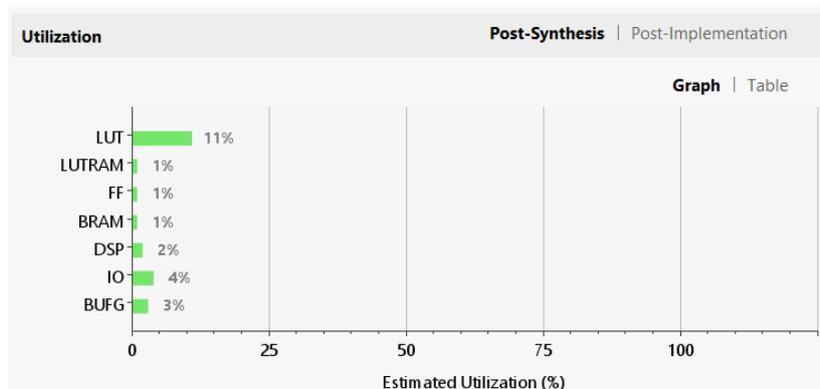


Рис. 13. Использование ресурсов ПЛИС Virtex 7 XC7VX485T-FFG1157-1 для перцептронного модуля

Таблица 1

**Использование ресурсов ПЛИС Virtex 7 XC7VX485T-FFG1157-1
для перцептронного модуля**

Ресурс	Оценка	Доступно	Использовано, %
LUT	33550	306900	11.05
LUTRAM	62	130800	0.05
FF	542	607200	0.09
BRAM	0.5	1030	0.05
DSP	46	2800	1.64
IO	25	600	4.17
BUFG	1	32	3.13

На рис. 14 представлен результат симуляции сверточного модуля.

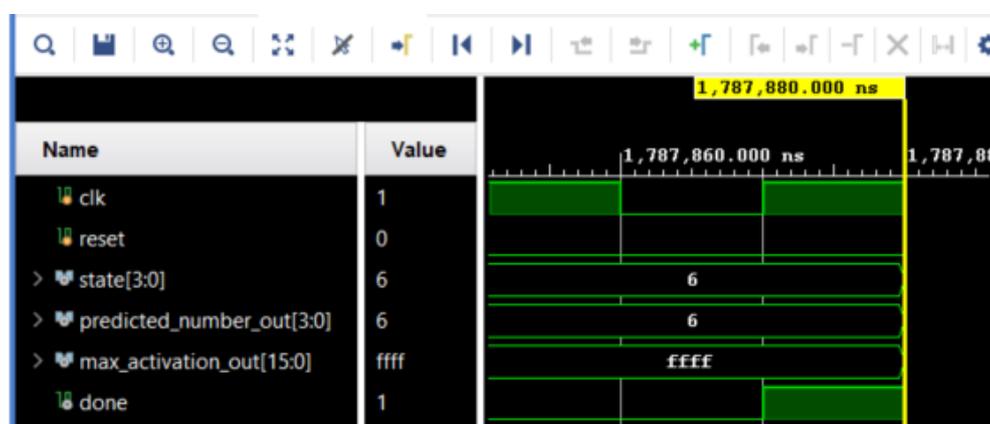


Рис. 14. Результат симуляции сверточного модуля для тестовых изображений

Синтез разработанного модуля на выбранной ПЛИС показал, что реализация была неэффективной, при использовании внутренних буферов для хранения промежуточных расчетов между слоями. Из-за нехватки памяти синтез задействовал LUT блоки для хранения информации. Значения использования ресурсов представлены на рис. 15 и табл. 2.

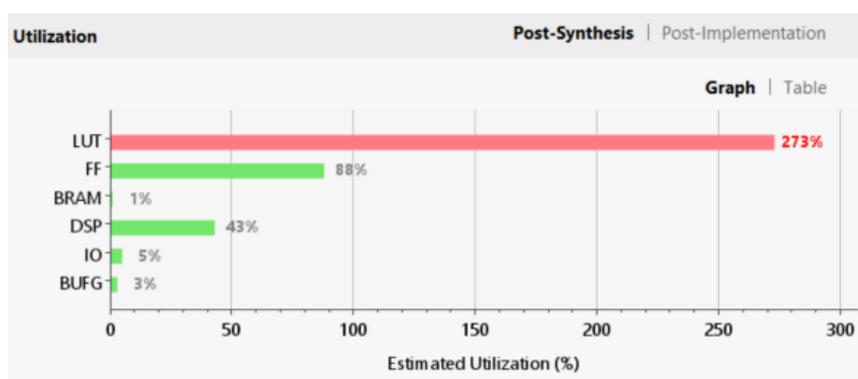


Рис. 15. Использование ресурсов ПЛИС Virtex 7 XC7VX485T-FFG1157-1 для сверточного модуля с внутренней реализацией буферов памяти

Таблица 2

**Использование ресурсов ПЛИС Virtex 7 XC7VX485T-FFG1157-1
для сверточного модуля с внутренней реализацией буферов памяти**

Ресурс	Оценка	Доступно	Использовано, %
LUT	827334	306900	272.51
FF	531382	607200	87.51
BRAM	5.5	1030	0.53
DSP	1200	2800	42.86
IO	27	600	4.5
BUFG	1	32	3.13

При переносе буферов для хранения промежуточно рассчитанных значений между слоями во внешние сточки, требования к портам и шине данных резко возрастают (рис. 16, табл. 3)

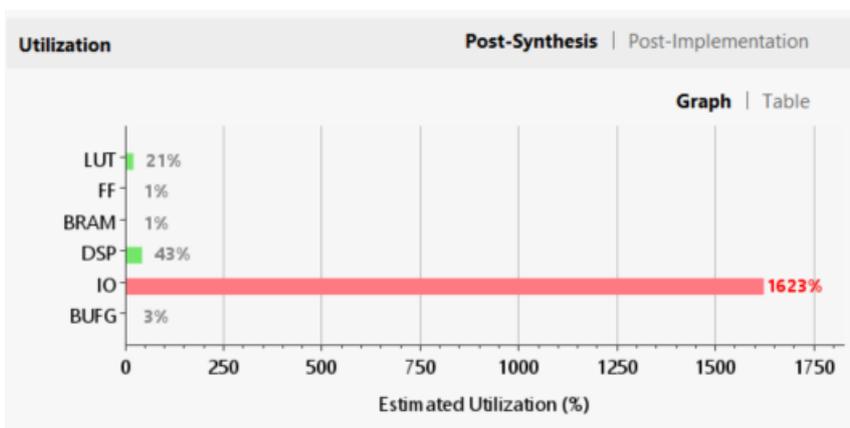


Рис. 16. Использование ресурсов ПЛИС Virtex 7 XC7VX485T-FFG1157-1
для сверточного модуля с внешней реализацией буферов памяти

Таблица 3

**Использование ресурсов ПЛИС Virtex 7 XC7VX485T-FFG1157-1
для сверточного модуля с внешней реализацией буферов памяти**

Ресурс	Оценка	Доступно	Использовано, %
LUT	65113	306900	21.45
FF	604	607200	0.1
BRAM	5.5	1030	0.53
DSP	1200	2800	42.86
IO	9736	600	1622.67
BUFG	1	32	3.13

Реализация нейронных сетей на программируемых логических интегральных схемах сталкивается с фундаментальными ограничениями, связанными с аппаратными ресурсами и архитектурными особенностями целевой платформы. Одной из ключевых проблем является ограниченный объем внутренней памяти, доступной для хранения промежуточных данных между слоями. Каждый этап обработки, включая свертку, пулинг и полносвязные преобразования, требует буферизации признаков карт, весовых коэффициентов и активаций, объем которых растет экспоненциально с увеличением глубины сети и разрешения входного изображения. Использование внутренних блоков памяти (BRAM) для этих целей быстро исчерпывает доступные ресурсы, что вынуждает прибегать к внешней памяти. Однако подключение внешних накопителей сопровождается

ся сужением шины данных и увеличением задержек доступа, что критически снижает пропускную способность конвейера и нарушает детерминированность временных характеристик системы.

Дополнительным вызовом становится высокая ресурсоемкость операций с данными повышенной разрядности. Квантование параметров до формата 16:16, несмотря на снижение ошибок округления, требует значительного количества логических элементов для реализации арифметических операций, включая умножение и накопление. В условиях дефицита ресурсов ПЛИС синтезатор вынужден заменять специализированные DSP-блоки на конфигурируемые LUT-таблицы, что приводит к неоптимальному распределению логики и ограничивает возможности параллельной обработки. Данная проблема усугубляется при реализации сложных функций активации или нормализации, где требуется использование дополнительных блоков для вычисления экспонент или сравнения значений.

Таким образом, проектирование нейросетевых модулей на ПЛИС требует поиска компромисса между точностью модели, объемом используемой памяти и степенью параллелизма вычислений. Необходимость минимизации обращений к внешней памяти и оптимизации распределения логических ресурсов остается критической задачей, особенно для глубоких сетей с многоуровневой иерархией признаков.

Заключение. Реализация нейронных сетей на программируемых логических интегральных схемах представляет собой сложную инженерную задачу, требующую баланса между вычислительной эффективностью, точностью модели и аппаратными ограничениями. Как демонстрируют проведенные исследования, ключевым фактором успешного внедрения является оптимизация ресурсоемких операций, таких как свертка, пулинг и матричные умножения, через использование конвейеризации, квантования данных и замены сложных математических функций на табличные вычисления. Однако даже при тщательном проектировании сохраняются фундаментальные ограничения, связанные с дефицитом внутренней памяти и необходимостью распределения логических элементов между конкурирующими задачами.

Применение архитектурных решений, таких как потоковая обработка данных и блочная арифметика, позволяет частично компенсировать эти ограничения, но не устраняет их полностью. Например, использование внешней памяти для хранения промежуточных признаков карт, хотя и расширяет доступный объем данных, вводит дополнительные задержки, снижающие детерминированность системы. Аналогично, замена DSP-блоков на LUT-таблицы для реализации операций умножения-накопления увеличивает гибкость проектирования, но сокращает ресурсы для параллельных вычислений.

Перспективным направлением для дальнейших исследований остается разработка методов адаптивного управления памятью и аппаратно-ориентированного сокращения моделей. Такие подходы могли бы минимизировать зависимость от внешних накопителей и повысить энергоэффективность без существенной потери точности. Кроме того, интеграция специализированных аппаратных ускорителей для операций свертки и Softmax способна снизить нагрузку на программируемую логику, что особенно актуально для систем реального времени.

Успешное развертывание нейросетевых алгоритмов на ПЛИС определяется не только выбором оптимальной архитектуры, но и глубоким пониманием взаимосвязи между программной моделью и аппаратной реализацией. Достижение приемлемого компромисса между этими аспектами открывает путь к созданию энергоэффективных embedded-решений, способных выполнять сложные задачи компьютерного зрения в условиях ограниченных вычислительных ресурсов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Vineetha K.V., Reddy M.M.S.K., Ramesh C., Kurup D.G. An efficient design methodology to speed up the FPGA implementation of artificial neural networks // Engineering Science and Technology, an International Journal. – 2023. – Vol. 47. – Art. 101542. – DOI: 10.1016/j.jestch.2023.101542.
2. Zhan J.Y., Yu A.T., Jiang W., Yang Y.J., Xie X.N., Chang Z.W., Yang J.H. FPGA-based acceleration for binary neural networks in edge computing // Journal of Electronic Science and Technology. – 2023. – Vol. 21, No. 2. – Art. 100204. – DOI: 10.1016/j.jnlest.2023.100204.

3. *Gyulai-Nagy Z.V.* Acceleration of Neural Network training algorithms via FPGA devices // *Procedia Computer Science*. – 2023. – Vol. 225. – P. 2674-2683. – DOI: 10.1016/j.procs.2023.10.259.
4. *Saady M.M., Essai M.H.* Hardware implementation of neural network-based engine model using FPGA // *Alexandria Engineering Journal*. – 2022. – Vol. 61, No. 12. – P. 12039-12050. – DOI: 10.1016/j.aej.2022.05.035.
5. *Крутиков А.К., Мельцов В.Ю.* Метод формирования многоярусной нейросетевой системы прогнозирования с возможностью реконфигурации // *Известия Юго-Западного государственного университета*. – 2024. – Т. 28, № 4. – С. 104-123. – DOI: 10.21869/2223-1560-2024-28-4-104-123. – EDN: IEBISN.
6. *Boudjadar J., Islam S.U., Buyya R.* Dynamic FPGA reconfiguration for scalable embedded artificial intelligence (AI): A co-design methodology for convolutional neural networks (CNN) acceleration // *Future Generation Computer Systems*. – 2025. – Vol. 169. – Art. 107777. – DOI: 10.1016/j.future.2025.107777.
7. *Mehrabi A., Bethi Y., van Schaik A., Afshar S.* An Optimized Multi-layer Spiking Neural Network implementation in FPGA Without Multipliers // *Procedia Computer Science*. – 2023. – Vol. 222. – P. 407-414. – DOI: 10.1016/j.procs.2023.08.179.
8. *Коновальчик А.П.* Архитектура высокопроизводительных вычислительных систем на основе ПЛИС // *Известия Юго-Западного государственного университета*. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. – 2011. – № 2. – С. 6-9. – EDN: PZRVTN.
9. *Лебедев М.С., Белецкий П.Н.* Реализация искусственных нейронных сетей на ПЛИС с помощью открытых инструментов // *Тр. ИСП РАН*. – 2021. – Т. 33, № 6. – С. 175-192. – DOI: 10.15514/ISPRAS.2021.33(6).12.
10. *Тарасов И.Е., Потехин Д.С., Платонова О.В.* Перспективы применения софт-процессоров в системах на кристалле на базе программируемых логических интегральных схем // *Russian Technological Journal*. – 2022. – Т. 10, № 3. – С. 24-33. – DOI: 10.32362/2500-316X-2022-10-3-24-33.
11. *Титенко Е.А., Титов В.С., Коновальчик А.П.* Высокопроизводительные вычислительные системы на основе ПЛИС // *Известия Юго-Западного государственного университета*. – 2012. – № 4-2(43). – С. 73а-77. – EDN: PGPOQD.
12. *Namboothiripad M.K., Vadhyan G.* Efficient implementation of artificial neural networks on FPGAs using high-level synthesis and parallelism // *International Journal of Advanced Technology and Engineering Exploration*. – 2024. – Vol. 11, No 119. – P. 1497-1511. – DOI: 10.19101/IJATEE.2023.10102538.
13. *Khalil K., Mohaidat T., Darwich M., Kumar A., Bayoumi M.* Efficient Hardware Implementation of Artificial Neural Networks on FPGA // *Proceedings of AICAS*. – 2024. – P. 233-237. – DOI: 10.1109/AICAS59952.2024.10595867.
14. *Tasci M., Istanbullu A., Tumen V., Kosunalp S.* FPGA-QNN: Quantized Neural Network Hardware Acceleration on FPGAs // *Applied Sciences*. – 2025. – Vol. 15, No. 2. – Art. 688. – DOI: 10.3390/app15020688.
15. *Gholami A., Kim S., Dong Z., Yao Z., Mahoney M.W., Keutzer K.* A Survey of Quantization Methods for Efficient Neural Network Inference // *ArXiv*. – 2021. – abs/2103.13630.
16. *Крышнев Ю.В., Соболев В.И.* Аппаратная реализация искусственной нейронной сети на FPGA для распознавания написанных от руки цифр // *Современные проблемы машиноведения*. – 2020. – С. 165-167.
17. *Блох Д.Е., Безмельцев А.И., Панищев В.С.* Нейросетевой модуль классификации рукописной цифры на ПЛИС // *Тринадцатый Национальный суперкомпьютерный форум*. – 2024.
18. *Блох Д.Е., Безмельцев А.И.* Распознавание рукописного ввода цифр на ПЛИС // *Интеллектуальные информационные системы: тенденции, проблемы, перспективы «ИИС – 2024»*. – Курск: Университетская книга, 2024. – С. 55-57. – EDN: SWQBIZ.
19. *Kadhim Z., Abdullah H., Ghathwan K.* Artificial Neural Network Hyperparameters Optimization: A Survey // *International Journal of Online and Biomedical Engineering (iJOE)*. – 2022. – Vol. 18. – P. 59-87. – DOI: 10.3991/ijoe.v18i15.34399.
20. *Воронцов К.В.* Реализация на ПЛИС нейросети для распознавания изображений // *Радиоэлектроника, электротехника и энергетика*. – 2025. – Вып. 25. – EDN: KQWENQ.

REFERENCES

1. *Vineetha K.V., Reddy M.M.S.K., Ramesh C., Kurup D.G.* An efficient design methodology to speed up the FPGA implementation of artificial neural networks, *Engineering Science and Technology, an International Journal*, 2023, Vol. 47, Art. 101542. DOI: 10.1016/j.jestch.2023.101542.
2. *Zhan J.Y., Yu A.T., Jiang W., Yang Y.J., Xie X.N., Chang Z.W., Yang J.H.* FPGA-based acceleration for binary neural networks in edge computing, *Journal of Electronic Science and Technology*, 2023, Vol. 21, No. 2, Art. 100204. DOI: 10.1016/j.jnlest.2023.100204.

3. Gyulai-Nagy Z.V. Acceleration of Neural Network training algorithms via FPGA devices, *Procedia Computer Science*, 2023, Vol. 225, pp. 2674-2683. DOI: 10.1016/j.procs.2023.10.259.
4. Saady M.M., Essai M.H. Hardware implementation of neural network-based engine model using FPGA, *Alexandria Engineering Journal*, 2022, Vol. 61, No. 12, pp. 12039-12050. DOI: 10.1016/j.aej.2022.05.035.
5. Krutikov A.K., Mel'tsov V.Yu. Metod formirovaniya mnogoyarusnoy neyrosetevoy sistemy prognozirovaniya s vozmozhnost'yu rekonfiguratsii [Method for forming a multi-layer neural network forecasting system with the possibility of reconfiguration], *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta* [News of Southwestern State University], 2024, Vol. 28, No. 4, pp. 104-123. DOI: 10.21869/2223-1560-2024-28-4-104-123. EDN: IEBISN.
6. Boudjadar J., Islam S.U., Buyya R. Dynamic FPGA reconfiguration for scalable embedded artificial intelligence (AI): A co-design methodology for convolutional neural networks (CNN) acceleration, *Future Generation Computer Systems*, 2025, Vol. 169, Art. 107777. DOI: 10.1016/j.future.2025.107777.
7. Mehrabi A., Bethi Y., van Schaik A., Afshar S. An Optimized Multi-layer Spiking Neural Network implementation in FPGA Without Multipliers, *Procedia Computer Science*, 2023, Vol. 222, pp. 407-414. DOI: 10.1016/j.procs.2023.08.179.
8. Konoval'chik A.P. Arkhitektura vysokoproizvoditel'nykh vychislitel'nykh sistem na osnove PLIS [Architecture of high-performance computing systems based on FPGA], *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika, informatika. Meditsinskoe priborostroenie* [News of Southwestern State University. Series: Management, computing, informatics. Medical instrument engineering], 2011, No. 2, pp. 6-9. EDN: PZRVTN.
9. Lebedev M.S., Beletskiy P.N. Realizatsiya iskusstvennykh neyronnykh setey na PLIS s pomoshch'yu otkrytykh instrumentov [Implementation of artificial neural networks on PLDs using open tools], *Tr. ISP RAN* [Proceedings of the Institute of Systems Engineering, Russian Academy of Sciences], 2021, Vol. 33, No. 6, pp. 175-192. DOI: 10.15514/ISPRAS.2021.33(6).12.
10. Tarasov I.E., Potekhin D.S., Platonova O.V. Perspektivy primeneniya soft-protessorov v sistemakh na kristalle na baze programmiruemykh logicheskikh integral'nykh skhem [Prospects for the use of soft processors in on-chip systems based on programmable logic integrated circuits], *Russian Technological Journal* [Russian Technological Journal], 2022, Vol. 10, No. 3, pp. 24-33. DOI: 10.32362/2500-316X-2022-10-3-24-33.
11. Titenko E.A., Titov V.S., Konoval'chik A.P. Vysokoproizvoditel'nye vychislitel'nye sistemy na osnove PLIS [High-performance computing systems based on FPGAs], *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta* [News of Southwestern State University], 2012, No. 4-2(43), pp. 73a-77. EDN: PGPOQD.
12. Namboothiripad M.K., Vadhyan G. Efficient implementation of artificial neural networks on FPGAs using high-level synthesis and parallelism, *International Journal of Advanced Technology and Engineering Exploration*, 2024, Vol. 11, No 119, pp. 1497-1511. DOI: 10.19101/IJATEE.2023.10102538.
13. Khalil K., Mohaidat T., Darwich M., Kumar A., Bayoumi M. Efficient Hardware Implementation of Artificial Neural Networks on FPGA, *Proceedings of AICAS*, 2024, pp. 233-237. DOI: 10.1109/AICAS59952.2024.10595867.
14. Tasci M., Istanbulu A., Tumen V., Kosunalp S. FPGA-QNN: Quantized Neural Network Hardware Acceleration on FPGAs, *Applied Sciences*, 2025, Vol. 15, No. 2, Art. 688. DOI: 10.3390/app15020688.
15. Gholami A., Kim S., Dong Z., Yao Z., Mahoney M.W., Keutzer K. A Survey of Quantization Methods for Efficient Neural Network Inference, *ArXiv*, 2021. abs/2103.13630.
16. Kryshnev Yu.V., Sobolev V.I. Appartnaya realizatsiya iskusstvennoy neyronnoy seti na FPGA dlya raspoznavaniya napisannykh ot ruki tsifr [Hardware implementation of an artificial neural network on FPGA for handwritten digit recognition], *Sovremennye problemy mashinovedeniya* [Modern Problems of Machine Learning], 2020, pp. 165-167.
17. Blokh D.E., Bezmel'tsev A.I., Panishchev V.S. Neyrosetevoy modul' klassifikatsii rukopisnoy tsifry na PLIS [Neural network module for classification of handwritten digits on FPGA], *Trinadtsaty Natsional'nyy superkomp'yuternyy forum* [Thirteenth National Supercomputer Forum], 2024.
18. Blokh D.E., Bezmel'tsev A.I. Raspoznavanie rukopisnogo vvoda tsifr na PLIS [Recognition of handwritten digit input on FPGA], *Intellektual'nye informatsionnye sistemy: tendentsii, problemy, perspektivy «IIS – 2024»* [Intelligent Information Systems: Trends, Problems, Prospects “IIS – 2024”]. Kursk: Universitetskaya kniga, 2024, pp. 55-57. EDN: SWQBIZ.
19. Kadhim Z., Abdullah H., Ghathwan K. Artificial Neural Network Hyperparameters Optimization: A Survey, *International Journal of Online and Biomedical Engineering (iJOE)*, 2022, Vol. 18, pp. 59-87. DOI: 10.3991/ijoe.v18i15.34399.
20. Vorontsov K.V. Realizatsiya na PLIS neyroseti dlya raspoznavaniya izobrazheniy [Implementation of a neural network for image recognition on a PLIS], *Radioelektronika, elektrotehnika i energetika* [Radioelectronics, Electrical Engineering, and Energy], 2025, Issue 25. EDN: KQWEHQ.

Мельник Эдуард Всеволодович – Южный федеральный университет; e-mail: evmelnik@sfedu.ru; г. Ростов-на-Дону, Россия; д.т.н.; профессор кафедры вычислительной техники.

Блох Денис Евгеньевич – Юго-Западный государственный университет; e-mail: den5553@yandex.ru; г. Курск, Россия; кафедра вычислительной техники; аспирант.

Безмельцев Александр Игоревич – Юго-Западный государственный университет; e-mail: a.i.bezmeltsev@yandex.ru; г. Курск Россия; кафедра вычислительной техники; аспирант.

Панищев Владимир Славиевич – Юго-Западный государственный университет; e-mail: gskunk@yandex.ru; г. Курск, Россия; к.т.н.; доцент; доцент кафедры вычислительной техники.

Полторацкий Сергей Николаевич – Юго-Западный государственный университет; e-mail: merlinserg@list.ru; г. Курск, Россия; к.т.н.; доцент кафедры вычислительной техники.

Melnik Eduard Vsevolodovich – Southern Federal University; e-mail: evmelnik@sfedu.ru; Rostov-on-Don, Russia; dr. of eng. sc.; professor, Department of Computer Engineering.

Blokh Denis Evgenievich – Southwestern State University; e-mail: den5553@yandex.ru; Kursk, Russia; the Department of Computer Engineering; postgraduate student.

Bezmeltsev Alexander Igorevich – Southwestern State University; e-mail: a.i.bezmeltsev@yandex.ru; Kursk, Russia; the Department of Computer Engineering; postgraduate student.

Panishchev Vladimir Slavievich – Southwestern State University; e-mail: gskunk@yandex.ru; Kursk, Russia; cand. of eng. sc.; associate professor; associate professor, Department of Computer Engineering.

Poltoratsky Sergey Nikolaevich – Southwestern State University; e-mail: merlinserg@list.ru; Kursk, Russia; cand. of eng. sc.; associate professor, Department of Computer Engineering.

УДК 004.89

DOI 10.18522/2311-3103-2025-5-229-243

В.А. Частикова, К.В. Козачёк, Е.С. Коробская, В.П. Кравцов

ОБНАРУЖЕНИЕ КИБЕРВТОРЖЕНИЙ НА ОСНОВЕ СЕТЕВОГО ТРАФИКА И ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЯ С ИСПОЛЬЗОВАНИЕМ ДАТАСЕТА UNSW-NB15

В статье основное внимание уделяется исследованию поведения пользователей и созданию поведенческих моделей. Это помогает улучшить точность определения аномалий и оперативно выявлять нестандартную активность в сети. Целью данного исследования является сравнительный анализ эффективности двух моделей машинного обучения – многослойного перцептрона (MLP) и алгоритма Random Forest – для обнаружения кибервторжений на основе анализа сетевого трафика и поведения пользователей. Поведенческие модели позволяют выявлять отклонения от нормальной активности пользователей и сетевых взаимодействий, что значительно повышает полноту обнаружения кибервторжений. При проведении исследования использовался набор данных UNSW-NB15, который включает актуальные типы атак и характеристики как сетевого трафика, так и пользовательской активности. Перед реализацией моделей была проведена предварительная обработка данных, выбор признаков, нормализация и кодирование категориальных признаков. Оценка моделей осуществлялась с использованием различных метрик, таких как точность (accuracy), полнота (recall), AUC-ROC, precision, F1-score и другие. Результаты исследования показали, что алгоритм Random Forest обеспечивает высокую точность классификации (95%), а многослойный перцептрон (MLP), в свою очередь, достиг выдающихся результатов по AUC (0.9830) и точности (precision, 0.9869). В работе представлен анализ и характеристика методов анализа поведения пользователей и классификации сетевого трафика, показано сравнение наборов данных для систем обнаружения вторжений (IDS), а также даны практические рекомендации по выбору моделей в зависимости от условий эксплуатации. Результаты исследования могут быть полезны при разработке адаптивных систем защиты, которые сочетают высокую точность и скорость работы.

Обнаружение кибервторжений; машинное обучение; UNSW-NB15; многослойный перцептрон (mlp); random forest; классификация сетевого трафика; AUC-ROC.