УДК 004.85
DOI: 10.17586/0021-3454-2024-67-4-330-337

МЕТОДЫ ОПТИМИЗАЦИИ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ

Н. С. Мокрецов*, Е. Д. Архипцев

Санкт-Петербургский государственный электротехнический университет „ЛЭТИ“ им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия
**nikitamokrecov6374@gmail.com*

Аннотация. Рассмотрены методы построения ускорителей глубокого обучения. Показано, что традиционные подходы к обеспечению отказоустойчивости ускорителей глубокого обучения основаны на избыточных вычислениях, что приводит к значительным накладным расходам, включая время обучения, энергопотребление и размеры интегральных схем. Рассмотрен метод, основанный на учете различий в уязвимости отдельных нейронов и битов каждого нейрона, частично решающий проблему избыточности вычислений. Метод позволяет избирательно защищать компоненты модели на уровне архитектуры и схемы, что снижает накладные расходы без ущерба для надежности модели. Показано, что квантование модели ускорителя глубокого обучения позволяет представлять данные меньшим числом битов, что снижает требования к аппаратным ресурсам.

Ключевые слова: *глубокое обучение, ускоритель глубокого обучения, отказоустойчивость, межуровневая оптимизация, квантование модели обучения*

Ссылка для цитирования: *Мокрецов Н. С., Архипцев Е. Д.* Методы оптимизации моделей нейронных сетей // Изв. вузов. Приборостроение. 2024. Т. 67, № 4. С. 330—337. DOI: 10.17586/0021-3454-2024-67-4-330-337.

METHODS FOR OPTIMIZING NEURAL NETWORK MODELS

N. S. Mokretsov*, E. D. Arkhiptsev

St. Petersburg Electrotechnical University, St. Petersburg, Russia
**nikitamokrecov6374@gmail.com*

Abstract. Methods for building optimized deep learning accelerators are discussed. Traditional approaches to fault-tolerant deep learning accelerators are shown to rely on redundant computation, which results in significant overheads including training time, power consumption, and integrated circuit size. A method is proposed that considers differences in the vulnerability of individual neurons and the bits of each neuron, which partially solves the problem of computational redundancy. The method allows you to selectively protect model components at the architectural and circuit levels, which reduces overhead without compromising the reliability of the model. It is shown that quantization of the deep learning accelerator model allows data to be represented in fewer bits, which reduces hardware resource requirements.

Keywords: deep learning, deep learning accelerator, fault tolerance, cross-layer optimization, learning model quantization

For citation: Mokretsov N. S., Arkhiptsev E. D. Methods for optimizing neural network models. Journal of Instrument Engineering. 2024. Vol. 67, N 4. P. 330—337 (in Russian). DOI: 10.17586/0021-3454-2024-67-4-330-337.

Введение. Применение глубокого обучения выходит за рамки традиционных областей, таких как компьютерное зрение и обработка естественного языка. Современные приложения глубокого обучения применяются в сферах, в которых безопасность является критически важным параметром, например, автономное вождение, аэрокосмическая промышленность и робототехника [1]. Очевидно, что к таким приложениям предъявляются высокие требования по надежности в дополнение к требованиям по точности результата и скорости его получения.

Поскольку модели глубокого обучения включают в себя множество нелинейных функций, которые смягчают влияние ошибок на результаты, точность в получении результата ложится на саму модель глубокого обучения.

Требуемая скорость обучения достигается использованием механизма ускорения (ускорителя) глубокого обучения. Ускоритель реализуется как процессор в виде электронной схемы, специализированный под алгоритмы глубокого обучения, который может встраиваться от мобильных устройств до серверов облачных вычислений.

Модели глубокого обучения включают в себя множество нелинейных функций, что приводит к отсеиванию большего количества ошибок в промежуточных вычислениях и смягчению влияния случайных сбоев на результат вывода модели. Эти особенности глубокого обучения показывают, что такой подход более отказоустойчив, по сравнению с вычислениями общего назначения. Многие работы по созданию надежных ускорителей глубокого обучения используют это естественное свойство отказоустойчивости, добиваясь совместной оптимизации точности, надежности и производительности.

Основной целью использования приложения глубокого обучения является обеспечение надежности. Например, бортовой модуль глубокого обучения транспортного средства должен соответствовать стандарту надежности самого средства [2, 3]. Таким образом, отказоустойчивая архитектура ускорителя является основой для обеспечения надежности и самой модели глубокого обучения, что обусловливает актуальность темы исследования.

Обзор методов проектирования отказоустойчивого ускорителя глубокого обучения. Для повышения надежности ускорителей глубокого обучения предлагаются различные методы отказоустойчивого проектирования на разных уровнях абстракции, таких как схема, архитектура и алгоритм.

На уровне схемы процессора ускорителя могут использоваться традиционные методы отказоустойчивого кодирования, применяющие, в частности, код коррекции ошибок для защиты встроенных кэшей. С одной стороны, это решает проблему потери точности, вызванной сбоями, с другой — большой объем избыточных вычислений негативно влияет на скорость обработки, размеры микросхемы ускорителя и энергопотребление.

Для снижения стоимости избыточных вычислений в работе [4] предложена идея предоставления приоритетной защиты битам старшего порядка вычислительных блоков процессора ускорителя, при этом младшие биты игнорируются. В статье [5] предлагается использовать логику стохастических вычислений (работающую с вероятностными сигналами) взамен традиционной двоичной вычислительной логики с целью повышения энергоэффективности вычислений. В работе [6] предложено применять схему Razor — интеллектуального обработчика программного кода динамических веб-страниц — для обнаружения временных сбоев, вызванных перепадами напряжения. В статье [7] добавление избыточных соединений к ускорителю нейронных сетей Хопфилда и использование логики голосования для исправления ошибок помогает оптимизировать модель обучения.

Для оптимизации модели глубокого обучения на уровне ее архитектуры также предложено несколько решений. Например, в [8] представлена гетерогенная вычислительная архитектура для преодоления проблемы произвольных сбоев в вычислительном массиве ускорителя за счет использования вычислительного массива скалярного произведения, отличного от двумерного пульсирующего массива, для достижения повторного вычисления задач на любом вычислительном блоке. В работе [9] предлагается выделять в ускорителе глубокого обучения высоконадежные и обычные вычислительные области, которые используются для обработки чувствительных и не чувствительных к сбоям вычислительных задач соответственно. Распределение чувствительных и не чувствительных к сбоям вычислительных задач может меняться в зависимости от модели или входных данных. В статье [10] описываются типы архитектуры, в которых для оптимизации добавляется интегрированный обучающий модуль поверх ускорителя, который позволяет проводить параллельные вычисления нескольких меньших моделей глубокого обучения, чтобы исключить аппаратные сбои и повысить надежность вывода. В работе [11] описывается решение по оптимизации модели нейронной сети с помощью до-

бавления в вычислительный массив глубокого обучения блоков проверки четности для исправления ошибок в режиме реального времени.

На уровне алгоритма устойчивость моделей глубокого обучения к аппаратным сбоям повышается за счет избыточности учитываемых параметров, а точность — за счет изменения параметров модели или ее архитектуры [12, 13]. Аппаратная схема ускорителя при таком подходе остается неизмененной. Некоторые алгоритмы повышают отказоустойчивость модели глубокого обучения за счет применения эквивалентных вычислительных методов, вводя новые функции активации или числовые ограничения либо используя механизмы контрольной суммы для исправления ошибок [14, 15].

Как показывает анализ источников, большинство отказоустойчивых методов глубокого обучения используются в основном на одном уровне схемы, архитектуры или алгоритма. Различные методы имеют преимущества и недостатки, также существуют различные ограничения в сценариях их использования. Несмотря на то что возможно использовать некоторые комбинации различных уровней архитектуры для повышения точности результатов или снижения затрат на отказоустойчивость, по-прежнему не хватает методов межуровневого проектирования отказоустойчивых ускорителей глубокого обучения.

Межуровневая оптимизация. Как показывает анализ источников, большинство отказоустойчивых методов глубокого обучения используются в основном на одном уровне схемы, архитектуры или алгоритма. Различные методы имеют преимущества и недостатки, также существуют различные ограничения в сценариях их использования. Несмотря на то что существуют некоторые комбинации различных уровней архитектуры для повышения точности результатов или снижения затрат на отказоустойчивость, по-прежнему не хватает методов межуровневого проектирования отказоустойчивых ускорителей глубокого обучения.

Общая архитектура многоуровневой оптимизации отказоустойчивого ускорителя глубокого обучения показана на рис. 1.

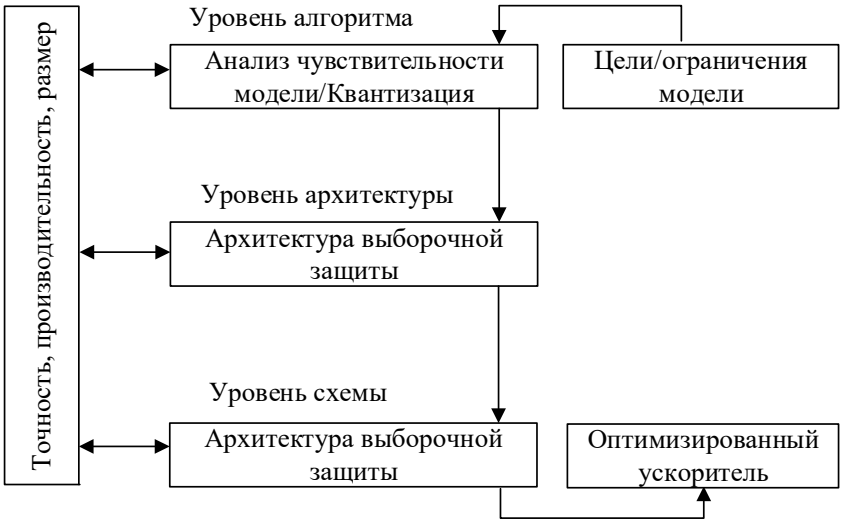

Рис. 1

В вычислительных задачах глубокого обучения целесообразно выделять важные и обычные вычисления, дополнительно разделять данные о нейронах на важные и обычные битовые данные. Такое разделение позволяет обеспечить избирательную защиту.

На уровне схемы предлагается резервирование с побитовой защитой — защищаются только важные логические разряды вычислительного блока. Для двух типов вычислительных массивов применяются различные методы защиты от избыточности, что еще больше снижает затраты на резервирование.

Таким образом, алгоритм построения межуровневой оптимизации имеет следующий вид.

1. Пользователь определяет цели модели обучения и ограничения, такие как производительность, надежность и доступные объемы вычислительных ресурсов. Пусть надежность обычно определяется точностью модели обучения, затраты на вычислительные ресурсы отражают объемы дополнительной вычислительной мощности (CPU, GPU, TPU и т.д.), необходимой для введения отказоустойчивости в модели, в сравнении со стандартной моделью нейронной сети.

2. Фреймворк анализирует чувствительность различных нейронов к сбоям на уровне алгоритма и использует это в качестве основы для разделения в глубоком обучении на важные вычисления и вычисления общего характера. Поскольку важные вычисления более чувствительны к сбоям и приводят к большей потере точности модели обучения, то для них требуются более надежные отказоустойчивые конструкции.

Для автоматического разделения на важные вычисления и вычисления общего характера в глубоком обучении можно воспользоваться расширением архитектуры HyCA [2] или настраиваемой архитектурой отказоустойчивого ускорителя глубокого обучения FlexHyCA, представленной в работе [16] и изображенной на рис. 2.

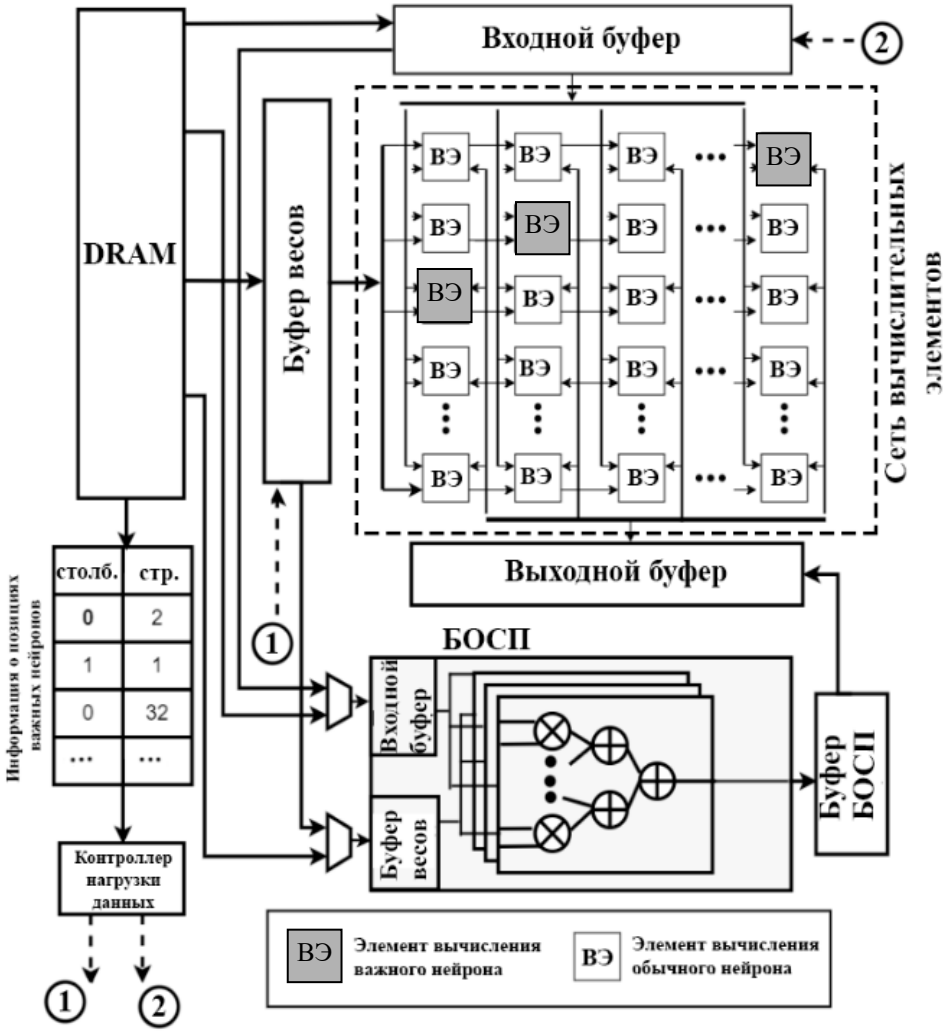

Рис. 2

FlexHyCA состоит из плоской сети вычислительных элементов для обработки обычных вычислений и блока обработки скалярного произведения (БОСП) небольшого количества важных вычислений, на которые распространяются требования высокой надежности. Информация о расположении важных нейронов попадает в сеть вычислительных элементов для сегментации элементов по важности вычисляемого в нем нейрона. За счет этого БОСП может

повторно использовать данные, загруженные в кэш сети вычислительных элементов, либо напрямую считывать необходимые данные из оперативной памяти (DRAM).

С учетом разницы в чувствительности к сбоям между важными и обычными битами нейронов в моделях глубокого обучения используются схемы селективной защиты битов на вычислительных блоках FlexHyCA. Только важные биты обычных нейронов и непосредственно связанные логические схемы важных битов важных нейронов избыточно защищаются, чтобы сократить дополнительные вычислительные ресурсы отказоустойчивого проектирования.

3. Квантование модели, которое позволяет использовать меньшую разрядность данных для вычислений и тем самым делать обработку данных в глубоком обучении более энергоэффективной. Как правило, процессор, на котором строится ускоритель глубокого обучения, поддерживает квантование с фиксированной точкой базовых вычислительных единиц, таких как блоки умножения-суммирования.

Для демонстрации результатов применения описанного выше подхода в работе [16] использовалось ПО, имитирующее случайные сбои (soft error), обозначенные как частота битовых ошибок (ЧБО). ЧБО устанавливает вероятность случайных сбоев в каждом блоке памяти, содержащем кэш. Для двух экспериментов (I — ЧБО = 0,0001; II — ЧБО = 0,0002) были установлены два различных ограничения по надежности/точности. По сравнению с обычной моделью глубокого обучения (с такой же функцией и ограничениями, но без требования оптимизации), потеря точности составляет менее 3 % в эксперименте I и менее 5 % — в II, при этом потеря производительности и пропускной способности составляет менее 10 %. Стоимость дополнительных вычислительных ресурсов сведена к минимуму при условии соблюдения ограничений по точности, производительности и пропускной способности.

В экспериментах использовались ImageNet в качестве датасета, а VGG 16 и ResNet50 — в качестве типичных моделей глубокого обучения для сравнения характеристик моделей. Обе модели были модифицированы с использованием 8-разрядного целочисленного квантования, в результате чего точность квантованной модели составила 72,95 и 75,96 % соответственно.

С целью проверки этого алгоритма межуровневого оптимизированного проектирования для ускорителя глубокого обучения с точки зрения аппаратной надежности (точности), накладных расходов на аппаратные ресурсы и производительности, сравниваются базовая конструкция ускорителя — Base, конструкция селективного тройного модульного резервирования (TMR — triple modular redundancy) на уровне схемы (CRT — circuit) с различной длиной битового слова: TMR-CRT1, TMR-CRT2, TMR-CRT3; схема селективного тройного модульного резервирования на уровне архитектуры (ARCH — architecture): TMR-ARCH; схема селективного тройного модульного резервирования на уровне алгоритма (ALG — algorithm): TMR-ALG; и схема межуровневого (CL — cross-layer) резервирования — TMR-CL.

Суть эксперимента заключалась в эмулировании случайных сбоев (fault injection) с упомянутыми выше частотами. Base — исходная модель, в которой не используется какая-либо защита от ошибок. Модели TMR соответствуют применению различных схем защиты с результирующим значением точности модели. На рис. 3 представлены изменения в точности моделей *T* вследствие применения межуровневой оптимизации (*а* — точность модели VGG 16 после оптимизации ускорителя глубокого обучения для эксперимента I; *б* — точность модели ResNet50 после оптимизации ускорителя глубокого обучения для эксперимента I; *в* — точность модели VGG 16 после оптимизации ускорителя глубокого обучения для эксперимента II; *г* — точность модели ResNet50 после оптимизации ускорителя глубокого обучения для эксперимента II).

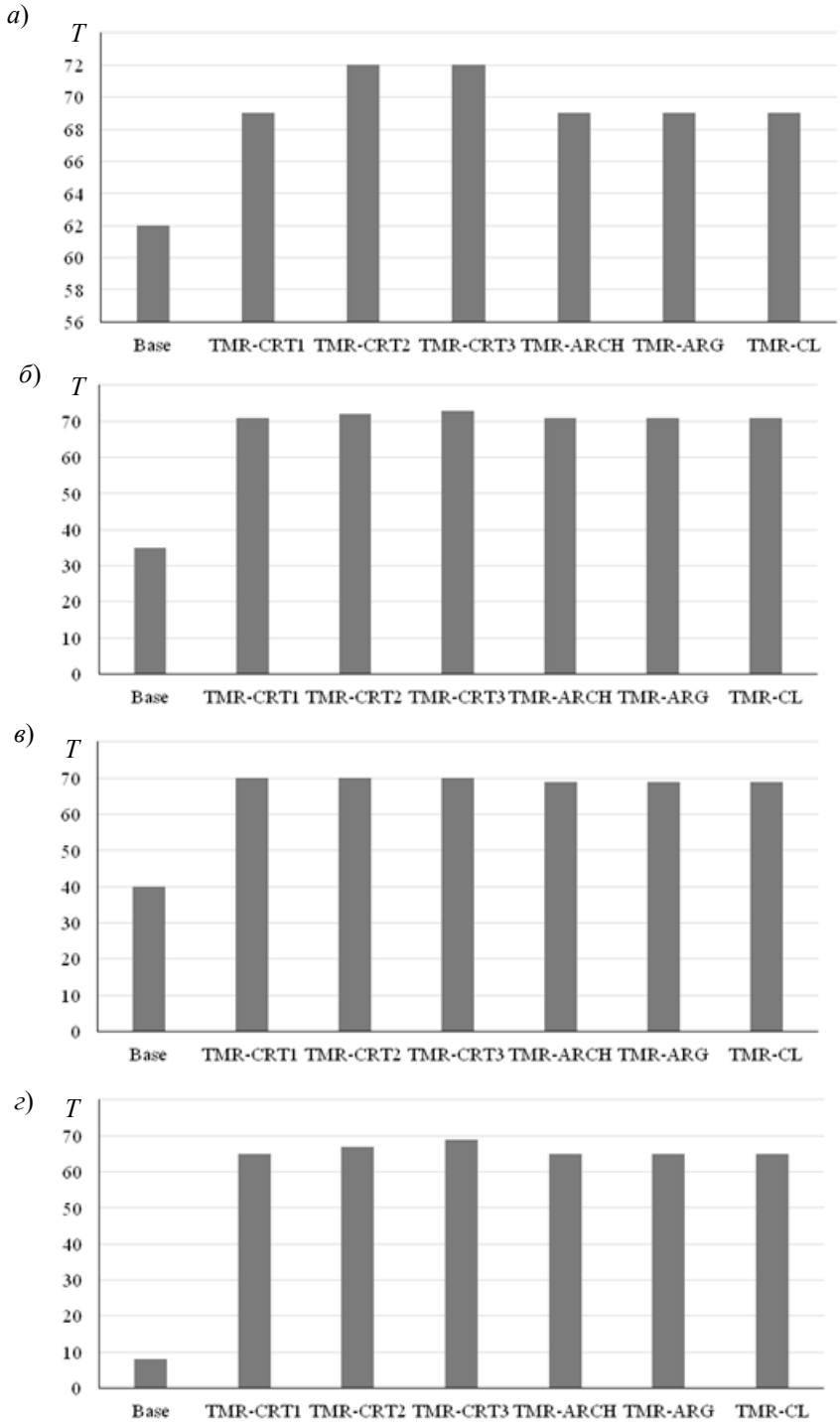

Рис. 3

В целом, различные отказоустойчивые стратегии могут соответствовать требованиям обеспечения точности. Однако относительная гибкость таких конструкций, как TMR-ARCH, TMR-ALG и TMR-CL, позволяет находить настройки, которые точно соответствуют требованиям пользователя. Напротив, степень детализации TMR-CRT относительно высока, а различия в точности значительны. TMR-CRT1 не соответствует требованиям по точности в соответствии со сценарием II, в то время как TMR-CRT2 и TMR-CRT3 превысили требования пользователя к точности.

Выводы. В обеспечении надежной работы критически важных приложений, основанных на моделях нейронных сетей, отказоустойчивое проектирование ускорителя глубокого обуче-

ния является определяющим фактором. Ускорители глубокого обучения способствуют повышению производительности и энергоэффективности обучения глубоких нейронных сетей.

Основные подходы к оптимизации ускорителя глубокого обучения специализируются только на одном из уровней абстракции моделей и зачастую основаны на избыточных вычислениях, что приводит к значительным затратам вычислительных ресурсов. Для обхода этих ограничений предлагается рассмотреть различия в уязвимости схемы к сбоям с точки зрения нейронных вычислений и разрядностей отдельного нейрона сети, и, основываясь на этих различиях, выбрать соответствующие методы выборочной защиты алгоритма, архитектуры и схемы модели глубокого обучения.

Эксперименты показывают, что межуровневая оптимизация ускорителя глубокого обучения позволяет настроиться под требования пользователя по точности модели обучения с учетом ограничений на имеющиеся ресурсы по энергоэффективности и производительности.

СПИСОК ЛИТЕРАТУРЫ

1. *Chen Y., Luo T., Liu S., Zhang S., He L., Wang J., Li L., Chen T., Xu Z., Sun N.* Dadiannao: A machine-learning supercomputer // Annual IEEE/ACM Intern. Symp. on Microarchitecture. 2014. Vol. 47. P. 609—622.

2. *Liu C., Chu C., Xu D., Wang Y., Wang Q., Li H., Li X., Cheng K., Hyca T.* A hybrid computing architecture for fault-tolerant deep learning // IEEE Transact. on Computer-Aided Design of Integrated Circuits and Systems. 2021. Vol. 41, N 10. P. 3400—3413.

3. *Dixit A., Wood A.* The impact of new technology on soft error rates // 2011 Intern. Reliability Physics Symposium. IEEE. 2011. P. 5B—4.

4. *Hoang L. H., Hanif M. A., Shafique M.* Ft-clipact: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation // Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE. 2020. P. 1241—1246.

5. *Ardakani A., Gross W. J.* Fault-tolerance of binarized and stochastic computing-based neural networks // IEEE Workshop on Signal Processing Systems (SiPS). IEEE. 2021. P. 52—57.

6. *Mittal S.* A survey on modeling and improving reliability of dnn algorithms and accelerators // J. of Systems Architecture. 2020. Vol. 104. P. 101.

7. *Chen Z., Li G., Pattabiraman K.* A low-cost fault corrector for deep neural networks through range restriction // Annual IEEE/IFIP Intern. Conf. on Dependable Systems and Networks (DSN). IEEE. 2021. Vol. 51. P. 1—13.

8. *Chen Y. H., Emer J., Sze V.* Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks // ACM SIGARCH computer architecture news. 2016. Vol. 44, N 3. P. 367—379.

9. *Libano F., Wilson B., Anderson J., Wirthlin M. J., Cazzaniga C., Frost C., Rech P.* Selective hardening for neural networks in fpgas // IEEE Transact. on Nuclear Science. 2018. Vol. 66, N 1. P. 216—222.

10. *Mahdiani H. R., Fakhraie S. M., Lucas C.* Relaxed fault-tolerant hardware implementation of neural networks in the presence of multiple transient errors // IEEE Transact. on Neural Networks and Learning Systems. 2012. Vol. 23, N 8. P. 1215—1228.

11. *Schorn C., Guntoro A., Ascheid G.* Accurate neuron resilience prediction for a flexible reliability management in neural network accelerators // Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE. 2018. P. 979—984.

12. *Мокрецов Н. С., Татарникова Т. М.* Самоорганизующиеся нейронные клеточные автоматы для обучения с подкреплением и эволюционного развития // Изв. СПбГЭТУ ЛЭТИ. 2023. Т. 16, № 7. С. 68—75.

13. *Sovetov B. Y., Tatarnikova T. M., Cehanovsky V. V.* Detection system for threats of the presence of hazardous substance in the environment // Proc. of 22nd Intern. Conf. on Soft Computing and Measurements, SCM 2019. 2019. P. 121—124.

14. *Wang H., Feng R., Han Z. F., Leung C. S.*Admm-based algorithm for training fault tolerant rbf networks and selecting centers // IEEE Transact. on Neural Networks and Learning Systems. 2017. Vol. 29, N 8. P. 3870—3878.

15. *Bertoa T. G., Gambardella G., Fraser N. J., Blott M., McAllister J.* Fault tolerant neural network accelerators with selective tmr // IEEE Design & Test. 2022. https://doi.org/10.1109/MDAT.2022.3174181.

16. *Rabe M., Milz S., Mader P*. Development methodologies for safety critical machine learning applications in the automotive domain: A survey // Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. 2021. P. 129—141.

Сведения об авторах

Никита Сергеевич Мокрецов — аспирант; Санкт-Петербургский государственный электротехнический университет „ЛЭТИ“ им. В.И. Ульянова (Ленина), кафедра информационных систем; E-mail: nikitamokrecov6374@gmail.com

Евгений Дмитриевич Архипцев — аспирант; Санкт-Петербургский государственный электротехнический университет „ЛЭТИ“ им. В.И. Ульянова (Ленина), кафедра информационных систем; E-mail: lokargenia@gmail.com

Поступила в редакцию 04.12.23; одобрена после рецензирования 08.12.23; принята к публикации 08.02.24.

REFERENCES

1. Chen Y., Luo T., Liu S., Zhang S., He L., Wang J., Li L., Chen T., Xu Z., Sun N. *Annual IEEE/ACM Intern. Symp. on Microarchitecture*, 2014, vol. 47, pp. 609–622.
2. Liu C., Chu C., Xu D., Wang Y., Wang Q., Li H., Li X., Cheng K., Hyca T. *IEEE Transact. on Computer-Aided Design of Integrated Circuits and Systems*, 2021, no. 10(41), pp. 3400–3413.
3. Dixit A., Wood A. *2011 Intern. Reliability Physics Symp*., IEEE, 2011, pp. 5B–4.
4. Hoang L.H., Hanif M.A., Shafique M. *Design, Automation & Test in Europe Conf. & Exhibition (DATE)*, IEEE, 2020, pp. 1241–1246.
5. Ardakani A., Gross W.J. *IEEE Workshop on Signal Processing Systems (SiPS)*, IEEE, 2021, pp. 52–57.
6. Mittal S. *Journal of Systems Architecture*, 2020, vol. 104, pp. 101.
7. Chen Z., Li G., Pattabiraman K. *Annual IEEE/IFIP Intern. Conf. on Dependable Systems and Networks (DSN)*, IEEE, 2021, vol. 51, pp. 1–13.
8. Chen Y. H., Emer J., Sze V. *ACM SIGARCH computer architecture news*, 2016, no. 3(44), pp. 367–379.
9. Libano F., Wilson B., Anderson J., Wirthlin M. J., Cazzaniga C., Frost C., Rech P. *IEEE Transact. on Nuclear Science*, 2018, no. 1(66), pp. 216–222.
10. Mahdiani H. R., Fakhraie S. M., Lucas C. *IEEE Transact. on Neural Networks and Learning Systems*, 2012, no. 8(23), pp. 1215–1228.
11. Schorn C., Guntoro A., Ascheid G*. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2018, pp. 979–984.
12. Mokretsov N.S., Tatarnikova T.M. *Proc. of Saint Petersburg Electrotechnical University*, 2023, no. 7(16), pp. 68–75. (in Russ.)
13. Sovetov B.Y., Tatarnikova T.M., Cehanovsky V.V. *Proc. of 22nd Intern. Conf. on Soft Computing and Measurements*, SCM 2019, 2019, pp. 121–124.
14. Wang H., Feng R., Han Z.F., Leung C.S. *IEEE Transact. on Neural Networks and Learning Systems*, 2017, no. 8(29), pp. 3870–3878.
15. Bertoa T.G., Gambardella G., Fraser N. J., Blott M., McAllister J. *IEEE Design & Test*., 2022, https://doi.org/10.1109/MDAT.2022.3174181.
16. Rabe M., Milz S., Mader P. *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2021, pp. 129–141.

Data on authors

Nikita S. Mokretsov — Post-Graduate Student; St. Petersburg Electrotechnical University, Department of Information Systems; E-mail: nikitamokrecov6374@gmail.com

Evgeny D. Arkhiptsev — Post-Graduate Student; St. Petersburg Electrotechnical University, Department of Information Systems; E-mail: lokargenia@gmail.com

Received 04.12.23; approved after reviewing 08.12.23; accepted for publication 08.02.24.