2.3.1 *СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ, СТАТИСТИКА (ТЕХНИЧЕСКИЕ, ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ)*

SYSTEM ANALYSIS, INFORMATION MANAGEMENT AND PROCESSING, STATISTICS

1.2.4 *КИБЕРБЕЗОПАСНОСТЬ (ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ)*

CYBERSECURITY

DOI: 10.33693/2313-223X-2025-12-2-28-36

УДК: 004.032.26

ГРНТИ: 28.23.37

EDN: QGDHYM

Анализ эффективности и робастности нейронных сетей с ранними выходами в задачах компьютерного зрения

А.Н. Чесалин[a] ©, А.В. Ставцев[b] ©, Н.Н. Ушкова[c] ©, В.В. Чаругин[d] ©, В.В. Чаругин[e] ©

МИРЭА – Российский технологический университет, г. Москва, Российская Федерация

[a] **E-mail: chesalin_an@mail.ru**

[b] **E-mail: stavcev@mirea.ru**

[c] **E-mail: ushkova@mirea.ru**

[d] **E-mail: charugin_v@mirea.ru**

[e] **E-mail: charugin@mirea.ru**

Аннотация. Во многих встраиваемых системах и устройствах интернета вещей (IoT) применяются нейросетевые алгоритмы для различных задач обработки информации. При этом разработчики сталкиваются с проблемой недостаточности вычислительных ресурсов для эффективного функционирования, особенно в задачах реального (псевдо) времени. В связи с этим актуальной является задача нахождение баланса между качеством результатов и вычислительной сложностью. Одним из способов повышения вычислительной эффективности нейронных сетей, является применение архитектур нейронных сетей с ранними выходами (например, BranchyNet), позволяющие принимать решения до прохождения всех слоев нейронной сети, в зависимости от исходных данных при заданной достоверности результатов. *Цель исследования*: провести анализ применимости, эффективности и робастности нейронных сетей с ранними выходами (BranchyResNet18) в задачах компьютерного зрения. Анализ проводится на основе набора данных дорожных знаков GTSRB. *Методология исследования* представляет собой экспериментальный анализ эффективности на основе расчета количества операций с плавающей запятой (FLOP) для получения результатов с заданной точностью, и экспериментальный анализ робастности на основе

генерации различных шумовых воздействий и состязательных атак. *Результаты исследования*: получены оценки эффективности нейронных сетей с ранним выходом и их робастность к непреднамеренным и преднамеренным возмущениям.

Ключевые слова: нейронные сети с ранним выходом, повышение эффективности нейронных сетей, робастность, состязательные атаки

ОБРАЗЕЦ ЦИТИРОВАНИЯ: Чесалин А.Н., Ставцев А.В., Ушкова Н.Н., Чаругин В.В., Чаругин В.В. Анализ эффективности и робастности нейронных сетей с ранними выходами в задачах компьютерного зрения // Computational Nanotechnology. 2025. Т. 12. № 2. С. 28–36. DOI: 10.33693/2313-223X-2025-12-2-28-36. EDN: QGDHYM

DOI: 10.33693/2313-223X-2025-12-2-28-36

Analysis of the Effectiveness and Robustness of Neural Networks with Early Exits in Computer Vision Tasks

A.N. Chesalin[a] ©, A.V. Stavtsev[b] ©, N.N. Ushkova[c] ©, V.V. Charugin[d] ©, V.V. Charugin[e] ©

MIREA – Russian Technological University, Moscow, Russian Federation

[a] **E-mail: chesalin_an@mail.ru**

[b] **E-mail: stavcev@mirea.ru**

[c] **E-mail: ushkova@mirea.ru**

[d] **E-mail: charugin_v@mirea.ru**

[e] **E-mail: charugin@mirea.ru**

Abstract. Many embedded systems and Internet of Things (IoT) devices use neural network algorithms for various information processing tasks. At the same time, developers face the problem of insufficient computing resources for effective functioning, especially in real-time (pseudo-) tasks. In this regard, the urgent task is to find a balance between the quality of the results and computational complexity. One of the ways to increase the computational efficiency of neural networks is to use neural network architectures with early exits (for example, BranchyNet), which allow making decisions before passing through all layers of the neural network, depending on the source data for a given reliability of the results. *The purpose of the study*: to analyze the applicability, effectiveness and robustness of neural networks with early exits (BranchyResNet18) in computer vision tasks. The analysis is based on the GTSRB road sign dataset. *The research methodology* is an experimental efficiency analysis based on the calculation of the number of floating-point operations (FLOP) to obtain results with a given accuracy, and an experimental robustness analysis based on the generation of various noise effects and adversarial attacks. *Research results*: estimates of the effectiveness of neural networks with early exit and their robustness to unintended and intentional disturbances have been obtained.

Key words: neural networks with early exit, improving the efficiency of neural networks, robustness, adversarial attacks

FOR CITATION: Chesalin A.N., Stavtsev A.V., Ushkova N.N., Charugin V.V., Charugin V.V. Analysis of the Effectiveness and Robustness of Neural Networks with Early Exits in Computer Vision Tasks. *Computational Nanotechnology*. 2025. Vol. 12. No. 2. Pp. 28–36. (In Rus.). DOI: 10.33693/2313-223X-2025-12-2-28-36. EDN: QGDHYM

ВВЕДЕНИЕ. ПОСТАНОВКА ПРОБЛЕМЫ

Эффективное применение нейронных сетей стало возможны благодаря стремительному развитию вычислительных ресурсов и появлению графических процессоров (GPU). При этом не все системы, в которых применяются нейронные сети обладают достаточными мощностями для их применения. К таким системам относятся встраиваемые системы, устройства IoT, робототехнические системы и пр., в них как правило имеются сильные вычислительные ограничения и важна энергоэффективность.

Для повышения эффективности нейронных сетей существует большое количество подходов, это: квантование моделей (Quantization) [1; 2], прореживание (pruning) [3], низкоранговое разложение (low-rank factorization of weight matrices) [4], дистилляция (distillation) [5; 6], кластеризация весов [7], применение ранних выходов [8]. Для получения эффективной нейронной сети представленные методы необходимо применять совместно, пример такой реализации представлен в работе [9; 10]. В данной работе рассматривается проблема построения нейронных сетей с ранним выходом для задач компьютерного зрения, и на различных наборах данных анализируется их эффективность и робастность к непреднамеренным (шуму) и преднамеренным помехам (состязательным атакам), что особенно актуально для данных систем.

Существуют разнообразные подходы к построению нейронных сетей с ранними выходами, это Conditional Deep Learning (CDL) [11], BranchyNet [12], Shallow-Deep Networks (SDN) [13], Synergistic Progressive Inference Neural Network (SPINN) [14], MSDNet (Multi-Scale Dense Networks) [15] и пр. Каждый из подходов имеет свои преимущества и недостатки, обзор данных подходов приводится в работах [16; 17]. Наиболее простой в реализации и удобной для применения (не требуется вносить изменения в исходную архитектуру нейронной сети, добавляются только вспомогательные выходы), при этом достаточно эффективной, является нейронная сеть BranchyNet. Данная нейронная сеть позволяет уменьшить количество вычислительных операций при незначительном снижении точности результатов. Нейронные сети с ранним выходом представляют собой логическое развитие теории каскадных классификаторов [18; 19], эффективно применяющихся в системах с жесткими ограничениями, но не достигают той точности которую можно получить применяя сверточные нейронные сети.

Далее в работе проводится анализ эффективности и робастности нейронной сети BranchyNet в задаче компьютерного зрения, а именно в распознавании дорожных знаков.

НЕЙРОСЕТЕВАЯ МОДЕЛЬ BranchyResNet18

Рассмотрим архитектуру BranchyNet. В качестве базовой (backbone) нейронной сети в данной работе выбрана известная архитектура ResNet18 [20; 21] (рис. 1). Используемая архитектура нейронной сети с ранними выходами BranchyResNet18 представлена на рис. 2 (рассматривается модифицированный вариант BranchyNet с легкими ранними выходами без сверточных слоев). BranchyResNet18 позволяет выполнять меньше операций для классификации изображений используя ранние выходы при достижении заданной уверенности в результате (используя настраиваемые пороги срабатывания ранних выходов). В качестве набора данных выбран популярный набор GTSRB – German Traffic Sign Recognition Benchmark, состоящий из более 50 000 изображений 43 дорожных знаков [22]. Данный набор часто используется для анализа качества и безопасности систем компьютерного зрения.

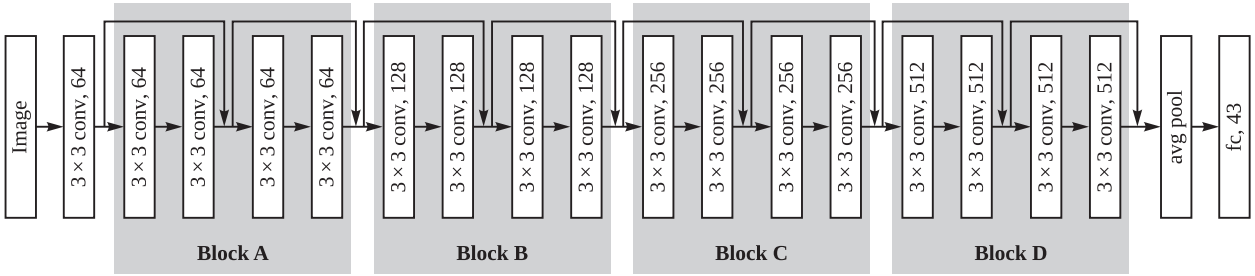

Рис. 1. Архитектура ResNet18

Fig. 1. Architecture ResNet18

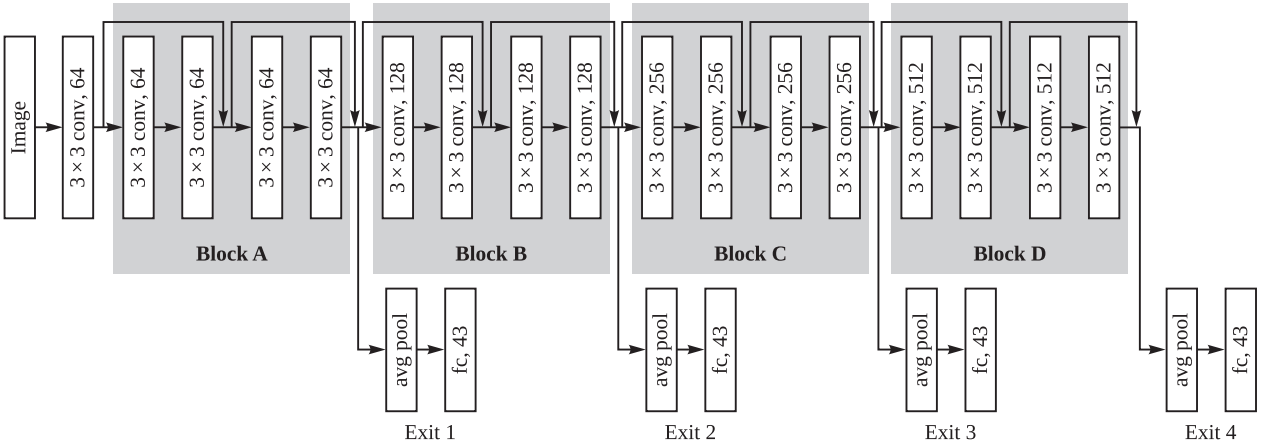

Рис. 2. Архитектура BranchyResNet18

Fig. 2. Architecture BranchyResNet18

АНАЛИЗ ЭФФЕКТИВНОСТИ BranchyResNet18

Анализ зависимости точности классификации от вычислительной сложности модели нейронной сети. В качестве меры вычислительной сложности модели используется количество операций с плавающей точкой (floating-point operations, FLOPs). FLOPs определяет общее количество вычислений с плавающей точкой (таких как сложение, вычитание, умножение, деление), необходимых для одного прохода модели вперед, обычно во время вывода. Расчет MFLOPs (megaFLOPs) с использованием библиотеки ptflops [23] для исследуемой модели BranchyResNet18, показал следующее количество операций для выхода из каждого блока (рис. 3):

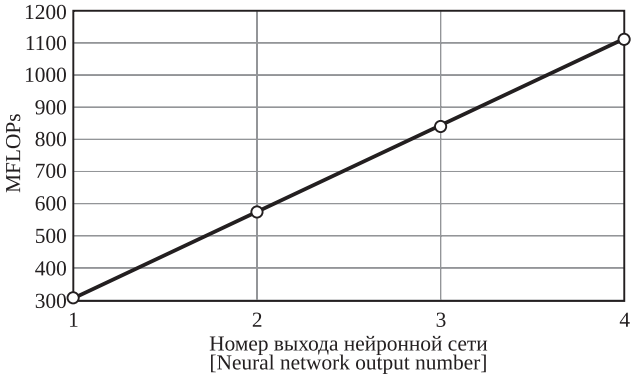

Рис. 3. Зависимость количества операций (FLOPs) от выхода нейронной сети

Fig. 3. The dependence of the number of operations (FLOPs) on the output of the neural network

Зависимость точности классификации от количества операций FLOPs показана на рис. 4.

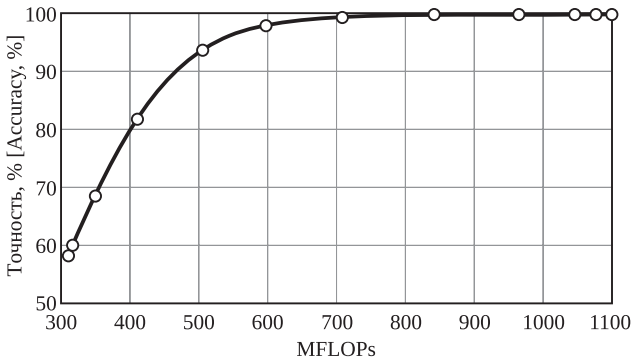

Рис. 4. Зависимость точности классификации от используемого количества операций FLOPs

Fig. 4. Dependence of classification accuracy on the number of FLOPs operations used

Из представленной зависимости можно отметить, что при значительном уменьшении количества FLOPs (с 1100 MFLOPs до 700MFLOPs) точность снижается незначительно (менее, чем на 0,5%). Таким образом, BranchyResNet18 позволяет значительно повысить эффективность базовой модели ResNet18 при заданном уровне точности: до 30% без изменения уровня точности, до 50% при уменьшении уровня точности менее чем на 0,5%, в два раза при уменьшении уровня точности на 5%.

Далее в исследовании робастности к непреднамеренным и преднамеренным воздействиям используется обученная модель BranchyResNet18, в которой уровень срабатывания ранних выходов настроен таким образом, чтобы обеспечить вычислительную эффективность в два раза больше, чем исходной модели ResNet18, при этом точность классификации не менее 95%, в этом случае большая часть объектов (более 70%) классифицируется первым и вторым ранним выходом.

АНАЛИЗ РОБАСТНОСТИ К НЕПРЕДНАМЕРЕННЫМ ВОЗМУЩЕНИЯМ (ШУМУ) BranchyResNet18

Проводится анализ зависимости точности от зашумленности изображения (разный шум). Рассматриваются разные зашумления изображений [24].

- Гауссовский шум (Gaussian noise) – для имитации ошибок датчиков, условий функционирования и природных вариаций:

$$x_{\text{noisy}} = x + N(\mu, \sigma),$$

где x – исходное изображение;
x_{noisy} – зашумленное изображение;
$N(\mu, \sigma^2)$ – нормально распределенный шум с математическим ожиданием μ и дисперсией σ^2.

- Равномерный шум (Uniform noise) – для имитации плохо откалиброванного сенсора или атмосферных помех:

$$x_{\text{noisy}} = x + U(-a, a),$$

где $U(-a, a)$ – равномерное распределение от $-a$ до $+a$.

- Импульсный шум (шум «соль и перец» Salt-and-Pepper noise) – некоторые пиксели случайным образом становятся полностью белыми (1,0) или черными (0,0). Для имитации ошибок передачи данных, такие как разрывы связи или повреждения матриц камер:

$$x_{\text{noizy}} = \begin{cases} 1, & \text{с вероятностью } p/2 \text{ (соль);} \\ 0, & \text{с вероятностью } p/2 \text{ (перец);} \\ x, & \text{с вероятностью } 1 - p. \end{cases}$$

- Blur (Размытие) – для имитации смазывания изображения от движения камеры, плохой фокусировки:

$$x_{\text{noisy}} = x * K,$$

где * – операция свертки;
K – ядро размытия (к примеру, гауссово).

На рис. 5 показаны зависимости точности классификации от уровня зашумления.

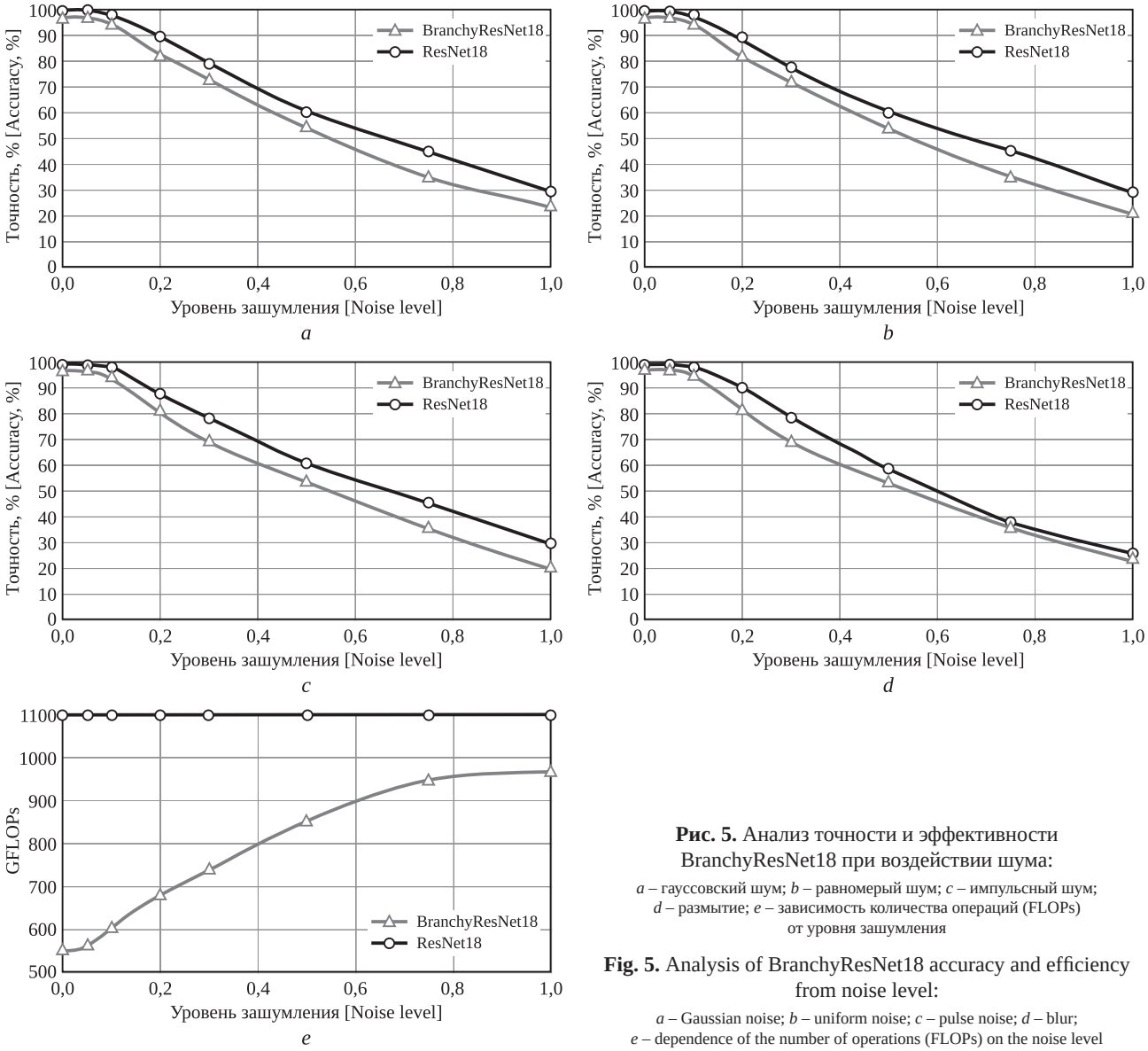

Рис. 5. Анализ точности и эффективности BranchyResNet18 при воздействии шума:

a – гауссовский шум; *b* – равномерый шум; *c* – импульсный шум; *d* – размытие; *e* – зависимость количества операций (FLOPs) от уровня зашумления

Fig. 5. Analysis of BranchyResNet18 accuracy and efficiency from noise level:

a – Gaussian noise; *b* – uniform noise; *c* – pulse noise; *d* – blur; *e* – dependence of the number of operations (FLOPs) on the noise level

Результаты, представленные на графиках на рис. 5, показывают, что:

- падение точности классификации BranchyResNet18 не сильно отличается от ResNet18, но в среднем на 5% выше;
- шум на уровне интенсивности 0,2 снижает качество классификации в среднем на 10%;
- эффективность применение BranchyResNet18 снижается при увеличении шума. Это связано с тем, что при повышении интенсивности атак ранние выходы не способны принять решение с заданной достоверностью и решение принимается последним выходом.

РОБАСТНОСТЬ К ПРЕДНАМЕРЕННЫМ ПОМЕХАМ (СОСТЯЗАТЕЛЬНЫМ АТАКАМ)

BranchyResNet18

Проводится анализ точности классификации при реализации атак на рассматриваемые модели. В исследовании используются известные атаки FGSM, BIM, PGD, DeepFool [25]:

- Атака Fast Gradient Sign Method (FGSM). Простейшая состязательная атака: создание возмущения, максимизирующего потери $L(x + \eta, y)$:

$$x_{\text{adv}} = x + \epsilon \operatorname{sign}(\nabla_x L(f(x), y)),$$

где y – истинное значение класса;
L – функция потерь модели (кросс-энтропия);
ϵ – коэффициент возмущения;
x_{adv} – атакованное изображение.

- Атака PGD (Projected Gradient Descent). Проецируется возмущение состязательности на ϵ-шар вокруг примера. Кроме того, перед вычислением градиента к исходному примеру добавляется равномерно рандомизированный шум.

$$x'_0 = x + U(-\epsilon, \epsilon),$$

$$x'_{t+1} = \prod_{B(x,\,\epsilon)} \left\{ x'_t + \alpha \operatorname{sign}\left(\nabla_{x'_t} L\left(f\left(x'_t\right), y\right)\right)\right\},$$

где $\prod_{B(x,\,\epsilon)}$ означает проекцию на $B(x, \epsilon)$;
U – равномерное распределение;
x'_t – обозначает пример состязательности после t шагов;
α – размер шага.

В качестве меры расстояния используется L_2.

- Атака DeepFool. Итеративно создается возмущение, минимальное по норме (обычно L_2), необходимое для изменения предсказанного класса. Классификатор аппроксимируется линейно, и на каждой итерации добавляется вектор, ведущий к ближайшей границе принятия решения [26].

$$w_k = \nabla f_k(x_t) - \nabla f_{\hat{k}}(x_t);$$

$$\delta_k = \frac{|\delta_k|}{\|w_k\|_2}, \quad \delta_k = f_k(x_t) - f_{\hat{k}}(x_t);$$

$$k^* = \arg\min_{k \neq \hat{k}} \delta_k, \quad x_{t+1} = x_t + \frac{\delta_{k^*}}{\|w_k\|_2^2} w_{k^*}.$$

- Basic IterativeMethod (BIM). Используется несколько градиентов для создания примера состязательности. Чтобы найти наилучшее возмущение, определяется размер шага α, меньший, чем θ. Формула выглядит следующим образом:

$$x'_{t+1} = \mathrm{clip}_{(x,\,\epsilon)}\left\{x'_t + \alpha\,\mathrm{sign}\left(\nabla_{x'_t} L\left(f\left(x'_t\right), y\right)\right)\right\},$$

где $\mathrm{clip}_{(x,\,\epsilon)}\{x'\}$ обозначает $\min(\max(x', x - \epsilon), x + \epsilon)$;
x'_t – состязательный пример после t шагов.

В качестве меры расстояния используется L_∞.

Для реализации атак используется бибиотека FoolBox [27].

На рис. 6 показаны зависимости робастности и эффективности при воздействии преднамеренными помехами.

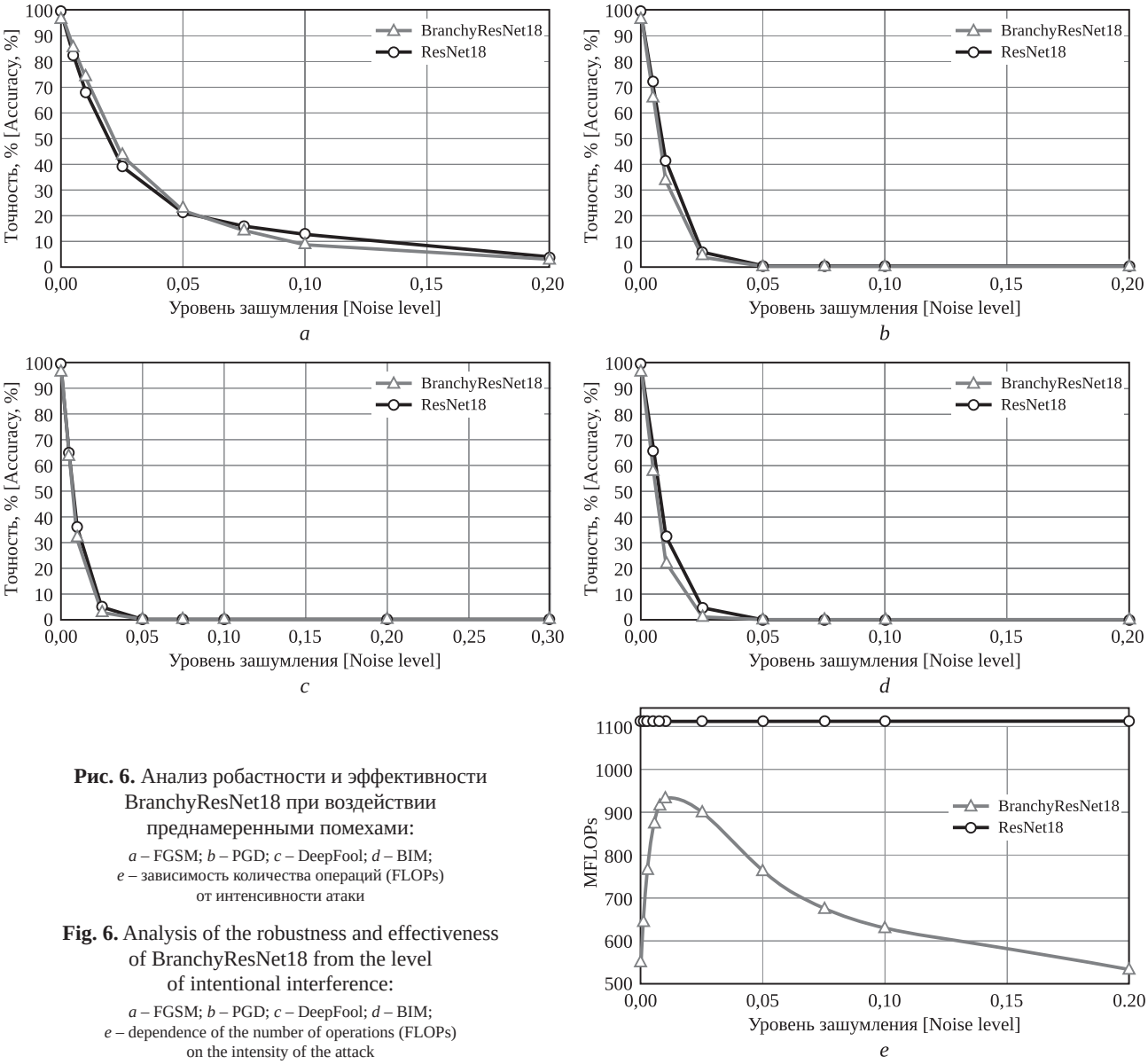

Рис. 6. Анализ робастности и эффективности BranchyResNet18 при воздействии преднамеренными помехами:
a – FGSM; *b* – PGD; *c* – DeepFool; *d* – BIM;
e – зависимость количества операций (FLOPs) от интенсивности атаки

Fig. 6. Analysis of the robustness and effectiveness of BranchyResNet18 from the level of intentional interference:
a – FGSM; *b* – PGD; *c* – DeepFool; *d* – BIM;
e – dependence of the number of operations (FLOPs) on the intensity of the attack

Результаты, представленные на графиках на рис. 6, показывают, что:

- устойчивость к атакам у ResNet и BranchyResNet18 находится на сопоставимом уровне, и уже при небольших состязательных возмущениях у представленных нейронных сетей значительно снижается точность. В связи с этим, при их обучении рекомендуется применять состязательное обучение и другие подходы к повышению устойчивости к состязательным атакам [28];
- при увеличении интенсивности атак количество операций BranchyResNet18 для принятия решения растет, но при большой интенсивности опять уменьшается. Это связано с тем, что при повышении интенсивности атак ранние выходы не способны принять решение с заданной достоверностью и решение принимается последним выходом, но при высокой интенсивности атаки искажают вход настолько сильно, что модель становится (ложно) уверенной уже на ранних уровнях и принимает неверное решение на раннем этапе.

ЗАКЛЮЧЕНИЕ

В работе проведено исследование проблемы повышения эффективности нейронных сетей для применения в системах, в которых существуют ограничения в вычислительных ресурсах. На наборе данных дорожных знаков проведен анализ эффективности и робастности архитектуры нейронной сети BranchyResNet18, показавший, что:

- BranchyResNet18 позволяет значительно повысить эффективность базовой модели ResNet18, при незначительном снижении точности классификации;
- робастность модели BranchyResNet18 незначительно ниже ResNet18, обе модели уязвимы к зашумлению данных, при обучении требуется применять методы робастного обучения;
- устойчивость к состязательным атака модели BranchyResNet18 и ResNet18 сопоставима, обе модели уязвимы к состязательным атакам и при обучении требуется применять методы состязательного обучения.

Применение нейронных сетей с ранним выходами является перспективным для применения во встраиваемых системах, системах реального времени и других системах с ограниченными вычислительными ресурсами. Архитектура BranchyResNet18 может быть использована для повышения эффективности базовой модели ResNet18 (при минимальном уменьшении точности возможно получить значительное уменьшение (в 1,5–2 раза) количества арифметических операций), но требуется повышение ее робастности используя робастные методы обучения нейронных сетей.

Литература

1. *Jacob B., Kligys S., Chen B.* et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018. Pp. 2704–2713. URL: https://arxiv.org/abs/1712.05877
2. *Торкунова Ю.В., Милованов Д.В.* Оптимизация нейронных сетей: методы и их сравнение на примере интеллектуального анализа текста // International Journal of Advanced Studies. 2023. Т. 13. № 4. С. 142–158. DOI: 10.12731/2227-930X2023-13-4-142-158.
3. *Han S., Pool J., Tran J., Dally W.J.* Learning both weights and connections for efficient neural networks // Advances in Neural Information Processing Systems (NeurIPS). 2015. URL: https://arxiv.org/abs/1506.02626
4. *Denton E.L., Zaremba W., Bruna J. et* al. Exploiting linear structure within convolutional networks for efficient evaluation // Advances in Neural Information Processing Systems (NeurIPS). 2014. URL: https://arxiv.org/abs/1404.0736
5. *Hinton G., Vinyals O., Dean J.* Distilling the knowledge in a neural network // arXiv Preprint arXiv:1503.02531. 2015. URL: https://arxiv.org/abs/1503.02531
6. *Иванов Е.А., Мамонова Т.Е.* Сравнение методов сжатия модели нейронной сети при использовании на микроконтроллере // Молодежь и современные информационные технологии: сборник трудов XXI Междунар. науч.-практ. конф. студентов, аспирантов и молодых ученых (Томск, 15–18 апреля 2024 г.). Томск, 2024. С. 176–180.
7. *Ullrich K., Meeds E., Welling M.* Soft weight-sharing for neural network compression // International Conference on Learning Representations (ICLR). 2017. URL: https://arxiv.org/abs/1702.04008

References

1. *Jacob B., Kligys S., Chen B.* et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018. Pp. 2704–2713. URL: https://arxiv.org/abs/1712.05877
2. *Torkunova Yu.V., Milovanov D.V.* Optimization of neural networks: Methods and their comparison using the example of text mining. *International Journal of Advanced Studies*. 2023. Vol. 13. No. 4. Pp. 142–158. (In Rus.). DOI: 10.12731/2227-930X2023-13-4-142-158.
3. *Han S., Pool J., Tran J., Dally W.J.* Learning both weights and connections for efficient neural networks. In: Advances in Neural Information Processing Systems (NeurIPS). 2015. URL: https://arxiv.org/abs/1506.02626
4. *Denton E.L., Zaremba W., Bruna J. et* al. Exploiting linear structure within convolutional networks for efficient evaluation. In: Advances in Neural Information Processing Systems (NeurIPS). 2014. URL: https://arxiv.org/abs/1404.0736
5. *Hinton G., Vinyals O., Dean J.* Distilling the knowledge in a neural network. *arXiv Preprint arXiv:1503.02531*. 2015. URL: https://arxiv.org/abs/1503.02531
6. *Ivanov E.A., Mamonova T.E.* Comparison of neural network model compression methods when used on a microcontroller. In: Youth and modern information technologies. Proceedings of the XXI International Scientific and Practical Conference of Students, Postgraduates and Young Scientists (Tomsk, April 15–18, 2024). Tomsk, 2024. Pp. 176–180.
7. *Ullrich K., Meeds E., Welling M.* Soft weight-sharing for neural network compression. In: International Conference on Learning Representations (ICLR). 2017. URL: https://arxiv.org/abs/1702.04008

8. *Scardapane S., Scarpiniti M., Baccarelli E., Uncini A.* Why should we add early exits to neural networks? // arXiv Preprint arXiv:2004.12814. 2020. URL: https://arxiv.org/abs/2004.12814
9. *Cheng Y., Wang D., Zhou P., Zhang T.* A Survey of model compression and acceleration for deep neural networks // arXiv Preprint arXiv:1710.09282. 2017. URL: https://arxiv.org/abs/1710.09282
10. *Bajpai D.J., Hanawal M.K.* A survey of early exit deep neural networks in NLP // arXiv Preprint arXiv:2501.07670. 2025. URL: https://arxiv.org/abs/2501.07670
11. *Panda P., Sengupta A., Roy K.* Conditional deep learning for energy-efficient and enhanced pattern recognition // Design, Automation & Test in Europe Conference (DATE). 2016. Pp. 475–480.
12. *Teerapittayanon S., McDanel B., Kung H.T.* BranchyNet: Fast inference via early exiting from deep neural networks // arXiv Preprint arXiv:1709.01686. 2017. URL: https://arxiv.org/abs/1709.01686
13. *Kaya Y., Hong S., Dumitras T.* Shallow-deep networks: Understanding and mitigating network overthinking // Proceedings of the 36th International Conference on Machine Learning (ICML). 2019. Pp. 3301–3310.
14. *Laskaridis S., Venieris S.I., Almeida M. et* al. SPINN: Synergistic progressive inference of neural networks over device and cloud // Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom'20). 2020. Pp. 1–15.
15. *Huang G., Chen D., Li T.* et al. Multi-scale dense networks for resource efficient image classification // International Conference on Learning Representations (ICLR). 2018. 14 p.
16. *Kaya Y., Dumitras T.* When does dynamic computation help early-exiting neural networks? // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2021. Pp. 2709–2718.
17. *Venieris S.I., Laskaridis S., Lane N.D.* Dynamic neural networks: A survey // IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). 2023. Vol. 45. No. 2. Pp. 2076–2098.
18. *Viola P., Jones M.* Rapid object detection using a boosted cascade of simple features // CVPR. 2001. Vol. 1.
19. *Чесалин А.Н.* Применение каскадных алгоритмов классификации для совершенствования систем обнаружения вторжений // Нелинейный мир. 2022. Т. 20. № 1. С. 24–41.
20. *He K., Zhang X., Ren S., Sun J.* Deep residual learning for image recognition // arXiv:1512.03385v1 [cs.CV]. 2015.
21. *Sai Abhishek, Allena Venkata.* Resnet18 model with sequential layer for computing accuracy on image classification dataset // International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 2022. Vol. 10. Pp. 2320–2882.
22. *Stallkamp J., Schlipsing M., Salmen J., Igel C.* The German Traffic Sign Recognition Benchmark: A multi-class classification competition // IEEE International Joint Conference on Neural Networks (IJCNN). 2011.
23. *Sovrasov V.* Ptflops: A flops counting tool for neural networks in Pytorch framework. 2024. URL: https://github.com/sovrasov/flops-counter.pytorch
24. *Гонсалес Р., Вудс Р.* Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с.
25. *Kim H.* Torchattacks: A PyTorch repository for adversarial attacks // arXiv:2010.01950. 2020.
26. *Moosavi-Dezfooli S.-M., Fawzi A., Frossard P.* DeepFool: A simple and accurate method to fool deep neural networks // arXiv:1511.04599. 2015.

8. *Scardapane S., Scarpiniti M., Baccarelli E., Uncini A.* Why should we add early exits to neural networks? *arXiv Preprint arXiv:2004.12814*. 2020. URL: https://arxiv.org/abs/2004.12814
9. *Cheng Y., Wang D., Zhou P., Zhang T.* A Survey of model compression and acceleration for deep neural networks. *arXiv Preprint arXiv:1710.09282*. 2017. URL: https://arxiv.org/abs/1710.09282
10. *Bajpai D.J., Hanawal M.K.* A survey of early exit deep neural networks in NLP. *arXiv Preprint arXiv:2501.07670*. 2025. URL: https://arxiv.org/abs/2501.07670
11. *Panda P., Sengupta A., Roy K.* Conditional deep learning for energy-efficient and enhanced pattern recognition. In: Design, Automation & Test in Europe Conference (DATE). 2016. Pp. 475–480.
12. *Teerapittayanon S., McDanel B., Kung H.T.* BranchyNet: Fast inference via early exiting from deep neural networks. *arXiv Preprint arXiv:1709.01686*. 2017. URL: https://arxiv.org/abs/1709.01686
13. *Kaya Y., Hong S., Dumitras T.* Shallow-deep networks: Understanding and mitigating network overthinking. In: Proceedings of the 36th International Conference on Machine Learning (ICML). 2019. Pp. 3301–3310.
14. *Laskaridis S., Venieris S.I., Almeida M.* et al. SPINN: Synergistic progressive inference of neural networks over device and cloud. In: Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom'20). 2020. Pp. 1–15.
15. *Huang G., Chen D., Li T.* et al. Multi-scale dense networks for resource efficient image classification. In: International Conference on Learning Representations (ICLR). 2018. 14 p.
16. *Kaya Y., Dumitras T.* When does dynamic computation help early-exiting neural networks? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2021. Pp. 2709–2718.
17. *Venieris S.I., Laskaridis S., Lane N.D.* Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*). 2023. Vol. 45. No. 2. Pp. 2076–2098.
18. *Viola P., Jones M.* Rapid object detection using a boosted cascade of simple features. *CVPR*. 2001. Vol. 1.
19. *Chesalin A.N.* Application of cascade classification algorithms for improving intrusion detection systems. *Nonlinear World*. 2022. Vol. 20. No. 1. Pp. 24–41. (In Rus.)
20. *He K., Zhang X., Ren S., Sun J.* Deep residual learning for image recognition. *arXiv:1512.03385v1* [cs.CV]. 2015.
21. *Sai Abhishek, Allena Venkata.* Resnet18 model with sequential layer for computing accuracy on image classification dataset. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 2022. Vol. 10. Pp. 2320–2882.
22. *Stallkamp J., Schlipsing M., Salmen J., Igel C.* The German traffic sign recognition benchmark: A multi-class classification competition. In: IEEE International Joint Conference on Neural Networks (IJCNN). 2011.
23. *Sovrasov V.* Ptflops: A flops counting tool for neural networks in Pytorch framework. 2024. URL: https://github.com/sovrasov/flops-counter.pytorch
24. *Gonzalez R., Woods R.* Digital image processing. Moscow: Technosphere, 2005, 1072 p.
25. *Kim H.* Torchattacks: A PyTorch repository for adversarial attacks. *arXiv:2010.01950*. 2020.
26. *Moosavi-Dezfooli S.-M., Fawzi A., Frossard P.* DeepFool: A simple and accurate method to fool deep neural networks. *arXiv:1511.04599*. 2015.

27. *Rauber J., Brendel W., Bethge M.* Foolbox: A Python toolbox to benchmark the robustness of machine learning models // arXiv:1707.04131. 2017.
28. *Goodfellow I., Shlens J., Szegedy C.* Explaining and harnessing adversarial examples // arXiv:1412.6572. 2015.

27. *Rauber J., Brendel W., Bethge M.* Foolbox: A Python toolbox to benchmark the robustness of machine learning models. *arXiv:1707.04131*. 2017.
28. *Goodfellow I., Shlens J., Szegedy C.* Explaining and harnessing adversarial examples. *arXiv:1412.6572*. 2015.

Статья проверена программой Антиплагиат. Оригинальность – 97,42%

Рецензент: *Холопов В.А.*, кандидат технических наук, доцент; заведующий, кафедра промышленной информатики; МИРЭА – Российский технологический университет

Статья поступила в редакцию 20.05.2025, принята к публикации 16.06.2025
The article was received on 20.05.2025, accepted for publication 16.06.2025

СВЕДЕНИЯ ОБ АВТОРАХ

Чесалин Александр Николаевич, кандидат технических наук, доцент; заведующий, кафедра компьютерной и информационной безопасности, Институт искусственного интеллекта; МИРЭА – Российский технологический университет; г. Москва, Российская Федерация. ORCID: 0000-0002-1154-6151; РИНЦ Author ID: 765419; SPIN-код: 4334-5520; E-mail: chesalin_an@mirea.ru

Ставцев Алексей Вячеславович, кандидат физико-математических наук; доцент, кафедра компьютерной и информационной безопасности, Институт искусственного интеллекта; МИРЭА – Российский технологический университет; г. Москва, Российская Федерация. РИНЦ Author ID: 165541; SPIN-код: 4948-2180; E-mail: stavcev@mirea.ru

Ушкова Надежда Николаевна, старший преподаватель, кафедра компьютерной и информационной безопасности, Институт искусственного интеллекта; МИРЭА – Российский технологический университет; г. Москва, Российская Федерация. РИНЦ Author ID: 1220726; SPIN-код: 1935-5513; E-mail: ushkova@mirea.ru

Чаругин Валентин Валерьевич, преподаватель, кафедра компьютерной и информационной безопасности, Институт искусственного интеллекта; МИРЭА – Российский технологический университет; г. Москва, Российская Федерация. ORCID: 0009-0001-1450-0714; РИНЦ Author ID: 1294084; SPIN-код: 7264-9403; E-mail: charugin@mirea.ru

Чаругин Валерий Валерьевич, преподаватель, кафедра компьютерной и информационной безопасности, Институт искусственного интеллекта; МИРЭА – Российский технологический университет; г. Москва, Российская Федерация. ORCID: 0009-0003-4950-7726; РИНЦ Author ID: 1294077; SPIN-код: 4080-4997; E-mail: charugin_v@mirea.ru

ABOUT THE AUTHORS

Alexander N. Chesalin, Cand. Sci. (Eng.), Associate Professor; Head, Department of Computer and Information Security, Institute of Artificial Intelligence; MIREA – Russian Technological University; Moscow, Russian Federation. ORCID: 0000-0002-1154-6151; RSCI Author ID: 765419; SPIN-code: 4334-5520; E-mail: chesalin_an@mirea.ru

Alexey V. Stavtsev, Cand. Sci. (Phys.-Math.); associate professor, Department of Computer and Information Security, Institute of Artificial Intelligence; MIREA – Russian Technological University; Moscow, Russian Federation. RSCI Author ID: 165541; SPIN-code: 4948-2180; E-mail: stavcev@mirea.ru

Nadezhda N. Ushkova, senior lecturer, Department of Computer and Information Security, Institute of Artificial Intelligence; MIREA – Russian Technological University; Moscow, Russian Federation. RSCI Author ID: 1220726; SPIN-code: 1935-5513; E-mail: ushkova@mirea.ru

Valentin V. Charugin, lecturer, Department of Computer and Information Security, Institute of Artificial Intelligence; MIREA – Russian Technological University; Moscow, Russian Federation. ORCID: 0009-0001-1450-0714; RSCI Author ID: 1294084; SPIN-code: 7264-9403; E-mail: charugin@mirea.ru

Valery V. Charugin, lecturer, Department of Computer and Information Security, Institute of Artificial Intelligence; MIREA – Russian Technological University; Moscow, Russian Federation. ORCID: 0009-0003-4950-7726; RSCI Author ID: 1294077; SPIN-code: 4080-4997; E-mail: charugin_v@mirea.ru