Дата поступления рукописи в редакцию: 25.04.2026 г.
Дата принятия рукописи в печать: 14.05.2026 г.
DOI: 10.24412/2224-9133-2026-4-52-58

СЕМЕНОВ Александр Александрович,
научный сотрудник
ФКУ НПО «СТиС» МВД России,
e-mail: mail@law-books.ru

АВЕТИСЯН Карэн Рафаелович,
старший преподаватель кафедры информационных технологий
и организации расследования киберпреступлений,
Московская академия Следственного комитета
Российской Федерации имени А.Я. Сухарева,
e-mail: Karen-Avetisyan-1989@bk.ru

ПРЕОБРАЗОВАНИЙ К ГЛУБОКОМУ ОБУЧЕНИЮ

Аннотация. *В работе рассматриваются базовые принципы построения нейросетей на примере полносвязной архитектуры. Показано, что использование только линейных преобразований недостаточно для моделирования сложных зависимостей, что обосновывает необходимость введения нелинейных функций активации. Описаны ключевые компоненты нейросети: полносвязные слои, функции активации, перцептрон и многослойные структуры. Рассмотрены особенности обучения с использованием градиентного спуска, включая функции потерь (кросс-энтропию) и нормализацию выходов с помощью softmax. Проанализированы ограничения полносвязных сетей и области их практического применения.*

Ключевые слова: *нейросеть, полносвязный слой, функция активации, сигмоида, softmax, перцептрон, многослойный перцептрон, глубокое обучение, градиентный спуск, обратное распространение ошибки, кросс-энтропия, классификация, конфиденс, линейные преобразования.*

SEMENOV Aleksandr Aleksandrovich,
Chief Researcher, Federal State Budgetary Institution "STIS"
of the Ministry of Internal Affairs of the Russian Federation

AVETISYAN Karen Rafaelovich,
Senior Lecturer, Department of Information Technology
and Cybercrime Investigation Organization
Moscow Academy of the Investigative Committee
of the Russian Federation named after A. Ya. Sukharev

FUNDAMENTALS OF CONSTRUCTING AND TRAINING FULLY CONNECTED NEURAL NETWORKS: FROM LINEAR TRANSFORMATIONS TO DEEP LEARNING

Annotation. *This paper examines the basic principles of constructing neural networks using a fully connected architecture as an example. It is shown that using linear transformations alone is insufficient for modeling complex dependencies, justifying the need to introduce nonlinear activation functions. Key components of a neural network are described: fully connected layers, activation functions, perceptrons, and multilayer structures. The specifics of training using gradient descent are discussed, including loss functions (cross-entropy) and output normalization using softmax. The limitations of fully connected networks and their practical applications are analyzed.*

Key words: *neural network, fully connected layer, activation function, sigmoid, softmax, perceptron, multilayer perceptron, deep learning, gradient descent, backpropagation, cross-entropy, classification, confidence, linear transformations.*

Попробуем составить простейшую нейросеть. Положим, у нас есть вектор *x* числовых значений, например, характеристик человека (старательность, внимательность, сила, ум, хитрость и т. д.), и мы хотим создать нейросеть, которая по ним будет определять, для выполнения какого рода задач (офисная работа, творческая, научная деятельность, инженерная и т. д.) этот человек годится лучше всего. Представим для начала, что нейросеть – это просто умножение некоторой матрицы весов *A* на вектор входных значений *x*, соответствующие соотношения для элементов векторов *x* и *y* и матрицы *A* выглядят так, как представлено на *рисунке 1* (здесь и далее по тексту в случаях, когда одновременно используются переменные одним буквенным обозначением, но одна из них дополнительно обладает индексом, переменная с индексом является частью переменной без индекса (значение вектора, столбец или строка матрицы и т. д.).

$$y = Ax \;\rightarrow\; y_i = \sum a_{ij} \cdot x_j$$

Рисунок 1. Соотношения для элементов векторов x и y и матрицы A.

Теоретически может случиться так, что при наличии достаточного датасета в результате машинного обучения получится подобрать весовую матрицу *A*, которая будет решать поставленную задачу удовлетворительно. В противовес этой гипотезе можно высказать следующее сомнение: как правило, сложные выводы не делаются в один шаг, они состоят из некоторой цепочки умозаключений, в которой окончательным выводам предшествуют какие-то промежуточные тезисы. По этой причине, если и не в приведенном примере, то по крайней мере в других более сложных задачах будет недостаточно одного матричного умножения. Можно попробовать реализовать нейросеть в виде нескольких матричных умножений, однако в силу свойств матриц несколько матричных умножений сведутся снова к одному (*рисунок 2*).

$$y = C\big(B(Ax)\big) = (CBA)x = Dx$$

Рисунок 2. Матричное умножение.

С другой стороны, можно представить себе массу зависимостей, которые если не точно, то хотя бы приблизительно, легче и точнее всего выразить

Нелинейность /активация – нелинейная функция с одним входным аргументом и одним выходным значением.

линейно, поэтому не хотелось бы отказываться от линейных операций в составе нейросетей. Подобно тому, как нельзя сыграть музыкальную симфонию на одном барабане или нельзя построить прочное здание одной отверткой для того, чтобы линейные конструкции в нейросетях не вырождались, перемножаясь друг с другом, и могли выражать какие-то сложные зависимости, к ним добавляют нелинейные математические выражения. Изначально такие выражения появились в попытке смоделировать активацию нейрона в мозгу, благодаря чему их иногда называют «активациями».

Чаще всего функции активации применяют ко всем выходным значениям слоя, реализующего то или иное линейное преобразование, (простейшим примером такого слоя является матричное умножение, как на рисунок 1). Применение активации к выходу одного слоя нейросети иногда удобно интерпретировать, как отдельный слой, при этом у этого слоя совершенно необязательно наличие собственных весов.

Слой активации – слой, состоящий из применения *функций активации к выходам предыдущего слоя. Чаще всего слой активации не содержит весов, т. е. необучаем.*

Полносвязный слой/full connected/dense/ linear layer – слой, представляющий собой матричное умножение вектора из всех входных значений на матрицу, при этом все элементы матрицы подлежат определению в процессе машинного обучения, т. е. являются весами. Следует обратить внимание на то, что англоязычный термин «linear» хоть и используется, но является крайне неудачным, так как называет именем общего понятия частный случай этого понятия.

Перцептрон – нейросеть, составленная из одного полносвязного слоя и следующего за ним слоя активации. Первым придуманным типом нейросетей был именно перцептрон.

Многослойный перцептрон/полносвязная нейросеть – нейросеть, составленная из нескольких чередующихся полносвязных слоев и слоев активации.

Глубокая нейросеть – термин, означающий нейросеть с более чем одним слоем, возникший вместе с появлением многослойного перцептрона. Фактически все используемые в современной практике нейросети глубокие.

Глубокое обучение – машинное обучение глубоких нейросетей. В настоящее время (здесь и далее под настоящим временем понимается начало 2025 г.) термин фактически употребляется синонимично термину «машинное обучение», однако формально им правильно называть именно обучение с применением метода обратного распространения ошибки, противопоставляя его некоторым принципиально другим, но менее распространенным, подходам к обучению.

Разумеется, не любая нелинейная функция одинаково хорошо подойдет для использования в нейросети. Следует иметь в виду, что при обучении нейросети градиентным спуском функции активации будут фигурировать в вычислении производных. Если функция активации будет обладать плохой аналитичностью, то зависимость ее выходного значения от входного будет плохо прослеживаться и, следовательно, градиентный спуск будет вести себя непредсказуемо, ломаться. Одной из наиболее распространенных и чуть ли не первой придуманной функцией активации является сигмоида (*рисунок 3*).

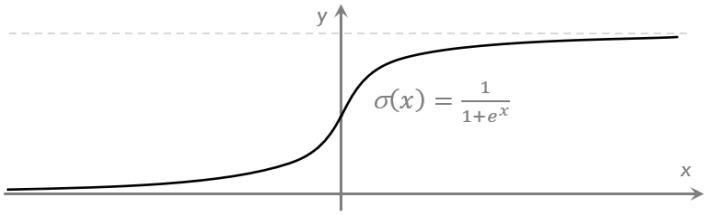


Рисунок 3. Сигмоида (формула № 4.3).

Сигмоида принимает значения от 0 до 1, ее первая производная везде больше нуля, а график функции тем более пологий, чем дальше значение аргумента от нуля. Если при обучении нейросети на некотором слое какое-то выходное значение окажется слишком большим по модулю, производная сигмоиды в соответствующей точке будет близка к нулю, что означает, что при вычислениях обратного хода соответствующая невязка будет близка к нулю. Следовательно, зависимые от него слагаемые при вычислении градиента также будут близки к нулю. Таким образом, использование сигмоиды тормозит изменение значений весов, которые порождают слишком большие выходные значения на слоях, стимулируя, наоборот, изменение значений остальных весов.

Можно придумать много функций, которые будут обладать теми же свойствами, что и сигмоида, в том числе и похожим графиком, однако конкретно сигмоида удобна тем, что ее производная легко вычисляется на основе самого значения сигмоиды, что упрощает подсчет невязок во время обратного хода (*рисунок 4*).

$$\sigma'(x) = \left(\frac{1}{1+e^x}\right)' = \left(\frac{-e^x}{(1+e^x)\cdot(1+e^x)}\right) = \sigma(x)\cdot(1-\sigma(x))$$

Рисунок 4. Подсчет невязок во время обратного хода

Положим теперь, что мы определились с некоторым количеством полносвязных слоев и активациями между ними. Вернемся к задаче определения рода действительно по характеристикам человека. Положим, что выходной вектор последнего полносвязного слоя состоит из четырех значений, соответствующих четырем родам деятельности. В качестве обучающего датасета используем набор характеристик сотрудников, а в качестве разметки – векторы из четырех значений, описывающие числами от 0 до 1 способность каждого сотрудника заниматься каждым из четырех родов деятельности. Представим себе, что после обучения мы подставили в нейросеть данные двух сотрудников и получили следующие результаты (*рисунок 5*).

$$(0.1, 0.7, 0.2, 0.1), (0.9, 0.8, 0.9, 0.9)$$

Рисунок 5. Данные двух сотрудников в нейросети.

Второе значение первого вектора больше других значенийв векторе, из чего вытекает, что первый сотрудник лучше всего справится с задачами рода, соответствующего второй позиции в векторе. У второго сотрудника, наоборот, второе значение – наименьше в векторе. Однако любое значение второго вектора больше любого значения первого. Отсюда легко проследить, что второй сотрудник, с точки зрения нашей нейросети, во всем лучше первого, но может потеряться тот вывод, что тяга ко второму роду деятельности у первого сотрудника ярко выражена, в то время как второй сотрудник справляется со всеми примерно одинаково. Как видно, числовые значения векторов может быть сложно интерпретировать, не зная диапазон возможных значений или не имея возможности сравнить их с другими результатами.

Для ситуаций, когда от нейросети хотят, чтобы она выдавала некоторый всегда выровненный вывод в предсказуемом диапазоне, используют специальные выравнивающие функции, наиболее распространенной из которых является функция софтмакс (softmax) (*рисунок 6*).

$$s(x),\ s_i = \frac{e^{x_i}}{\sum e^{x_k}}$$

Рисунок 6. Функция софтмакс (softmax).

Функция софтмакс всегда выдает вектор значений от 0 до 1, сумма которых всегда равна единице, при чем если одно значение входного вектора было больше другого, то это отношение сохранится для соответствующих значений выходного вектора. В силу указанных свойств выходные значения софтмакса удобно интерпретировать, как распределение вероятностей, и использовать при решении задачи классификации, т. е. присвоения переданному в нейросеть объекту того или иного класса из нескольких. Значения, полученные таким образом на одном запуске нейросети, часто небрежно называют вероятностями, из-за чего не погруженные в детали люди начинают путать эту величину с точностью работы нейросети в целом.

Задача классификации – задача, при которой заранее известны несколько классов и от нейросети требуется уметь отнести переданную ей единицу входных данных к одному из этих классов.

Конфиденс/confidence (в контексте решения нейросетью задачи классификации или похожей) – относительная вероятность принадлежности к указанному классу по сравнению с вероятностями принадлежности к другим классам.

Понятие конфиденса не имеет прямого отношения к реальной вероятности принадлежности объекта к тому или иному классу или к точности нейросети в целом. Например, если у нейросети высокая точность, т. е. она почти всегда идеально срабатывает, то это означает, что максимальное значение конфиденса с вероятностью, близкой к 1, соответствует действительному классу объекта, даже если оно будет числом около 0.1. Напротив, у нейросети с низкой точностью даже высокое значение конфиденса около 0.99 может ошибочно указывать на класс, к которому объект в действительности не принадлежит.

Теперь необходимо определить лосс. Для задачи классификации такой раздел математики, как математическая статистика, рекомендует функцию, называемую кросс-энтропией. Для вектора результатов *y* и вектора идеальных желаемых значений Y, взятого из разметки, кросс-энтропия вычисляется по формуле (*рисунок 7*).

$$L(y, \hat{y}) = -\sum_i \hat{y}_i \cdot log\,(y_i)$$

Рисунок 7. Вычисление кросс-энтропии.

Можно убедиться с помощью ручки и бумаги, что если совместить кросс-энтропию и софтмакс в единую функцию, то значения входных невязок r_i лосса будут вычисляться по элементарной формуле (*рисунок 8*).

$$L(y,\hat{y}) = -\sum_i \hat{y}_i \cdot log\left(\frac{e^{x_i}}{\sum e^{x_k}}\right), r_i = L'_{x_i} = \frac{e^{x_i}}{\sum e^{x_k}} - \hat{y}_i$$

Рисунок 8. Вычисление входных невязок ri лосса.

Определим теперь действия обратного хода для слоев нейросети. Для слоя активации с сигмоидой вычисление значений входных невязок r_i по значениям выходных невязок q_i и элементов x_i входной карты признаков выполняется в соответствии с формулой на *рисунке 9.*

$$r_i = q_i \cdot \sigma(x_i) \cdot (1 - \sigma(x_i))$$

Рисунок 9. Вычисление значений входных невязок ri
по значениям выходных невязок qi и элементов xi входной карты признаков

Чтобы разобраться с тем, как выглядят вычисления обратного хода для полносвязного слоя, представим себе какой-нибудь лосс L и подсчитаем его производную по одному из входных значений полносвязного слоя и по одному из значений весов a_{ik}. Для удобства снова обозначим входные и выходные невязки r_i и q_i соответственно (*рисунок 10*).

$$y_i = \sum_k a_{ik} \cdot x_k, L,$$
$$r_k = L'_{x_k} = \sum_i \left(L'_{y_i} \cdot \left(\sum_k a_{ik} \cdot x_k\right)'_{x_k}\right) =$$
$$\sum_i L'_{y_i} \cdot a_{ik} = \sum_i q_i \cdot a_{ik}, r = A^T q,$$
$$L'_{a_{ik}} = \sum_i \left(L'_{y_i} \cdot \left(\sum_k a_{ik} \cdot x_k\right)'_{a_{ik}}\right) = q_i \cdot x_k$$

Рисунок 10. Производная лосса L.

Формула, представленная на рисунке 9, демонстрирует удобное свойство слоев, выполняющих линейные операции. Как видно, число операций прямого хода для слоя равно числу операций обратного хода для вычисления невязок и числу операций обратного хода для вычисления градиентов. На самом деле в современных нейросетях чаще всего большая часть вычислений реализована в виде тех или иных комбинаций линейных слоев. Как следствие, количество операций во время обучения современных нейросетей, как правило, приблизительно в три раза больше числа операций во время инференса. Это следует помнить при оценке времени выполнения вычислений.

Слоям, осуществляющим линейные преобразования в прямом ходе, соответствуют также линейные преобразования в обратном ходе.

Таким образом, мы разобрали все действия прямого и обратного хода для полносвязного слоя, слоя активации с сигмоидой и лосса, составленного из софтмакса и кросс-энтропии, что позволяет запрограммировать шаг градиентного спуска для любой нейросети, состоящей из указанных слоев и применяющих указанный лосс.

Полносвязные нейросети относительно сложно учить, поскольку в них каждый вес используется для единственной операции, т. е. весов, попросту говоря, слишком много, в то время как значение каждого веса слишком незначительно. На практике полносвязные нейросети применяются в задачах, в которых входные данные обладают относительно небольшой размерностью, а природа задачи не подразумевает многоэтажных зависимостей, т. е. допускает составление нейросети из 2–4 слоев. Такие задачи, как правило, не сильно интеллектуальны. Примерами могут послужить некоторые простые задачи управления в робототехнике (или, что почти то же самое, управлении в компьютерных играх), разного рода примитивные сенсоры, а также классификация объектов на маленьких (около 32×32) фрагментах изображений. В последнем случае каждый пиксель фрагмента порождает как минимум одно входное значение нейросети, при этом обнаружение объекта требует, как минимум, сопоставления инфор-

мации на соседних пикселях. Отсюда интуитивно понятно, что число выходных значений первого слоя будет сравнимо с числом входных. Таким образом, зависимость требуемого числа весов от размера изображения для полносвязных нейросетей близка квадратичной, поэтому применение полносвязных слоев для анализа изображений нормальных размеров практически неосуществимо.

Список литературы:

[1] Chen Zhang, Joohee Kim Video object detection with two-path convolutional LSTM pyramid. 2016. URL: https://www.researchgate.net/publication /343702579_Video_Object_Detection_With_ Two-Path_ Convolutional_LSTM_ Pyramid/fulltext/5f3b2658928 51cd302013482/Video-Object-Detection-With-Two-Path-Convolutional-LSTM-Pyramid.pdf/ (дата обращения: 16.01.2026).

[2] Hanson A., Koutilya PNVR, Sanjukta Krishnagopal, Larry Davis Bidirectional Convolutional LSTM for the Detection of Violence in Videos. 2018. URL: https://openaccess.thecvf.com/content_eccv_2018_workshops/w10 /html/Hanson_Bidirectional_Convolutional_LSTM_or_the_Detection_of_ Violence_in_ Videos_ ECCVW_2018_paper.html (дата обращения: 15.01.2026).

[3] Howard A.G., Menglong Zhu, Bo Chen, Kalenichenko D. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017. URL: https://arxiv.org/pdf/1704.04861/ (дата обращения: 11.01.2026).

[4] Iandola F.N., Han S., Moskewicz M.W., Ashraf K., Dally W.J., Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size Recognition. 2016. Published as a conference paper at ICLR. 2015. URL: _and_05MB_model_size (дата обращения: 10.01.2026).

[5] Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015. URL: https://arxiv.org/pdf/1502.03167/ (дата обращения: 13.01.2026).

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Deep Residual Learning for Image Recognition. 2015. URL: https://arxiv.org/pdf/1512.03385 (дата обращения: 11.01.2026).

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Spatial pyramid pooling in deep convolutional networks for visual recognition. 2015. URL: https://arxiv.org/pdf/1406.4729/ (дата обращения: 11.01.2026).

[8] Krizhevsky I., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e 924a68c45b-Paper.pdf (дата обращения: 12.01.2026).

[9] Lavin A., Scott Gray Fast Algorithms for Convolutional Neural Networks. 2015. URL: https://arxiv.org/abs/1509.09308 (дата обращения: 11.01.2026).

[10] Mikolov T., Kai Chen, Corrado G., Jeffrey Dean Efficient Estimation of Word Representations in Vector Space. 2013. URL: https://arxiv.org/abs/1301.3781 (дата обращения: 15.01.2026).

[11] Norman P. Jouppi, Cliff Young, Nishant Patil and others In-Datacenter Performance Analysis of a Tensor Processing Unit. 2017. URL: https://arxiv.org/pdf/1704.04760 (дата обращения: 11.01.2026).

[12] Ronneberger O., Fischer Ph., Brox Th. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. URL: https://arxiv.org/pdf/1505.04597/ (дата обращения: 13.01.2026).

[13] Szegedy C., Wei Liu, Chapel Hill, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich Going Deeper with Convolutions. 2014. URL: https://arxiv.org/pdf/1409.4842 (дата обращения: 12.01.2026)

[14] Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2015. Published as a conference paper at ICLR 2015. URL: https://arxiv.org/pdf/1409.1556 (дата обращения: 10.01.2026).

[15] Wei Liu, Anguelov D., Erhan D., Szegedy C., Reed S., Cheng-Yang Fu, Berg A.C. SSD: Single Shot MultiBox Detector. 2016. URL: https://arxiv.org/pdf/1512.02325 (дата обращения: 11.01.2026).

[16] Официальный сайт ONNX. URL: https://onnx.ai/. (дата обращения: 11.01.2026).

[17] Основы технологий искусственного интеллекта: учебное пособие/ А.А. Семенов, В.В. Гончар, А.С. Эрдниев. – Московский университет МВД России имени В.Я, Кикотя, 2025. – 190с.

[18] Финансовый мониторинг: учебное пособие / В.И. Глотов, А.У. Альбеков, О.Н. Тисен и др. – КноРус, 2022. – 198 с.

[19] Тисен О.Н. Противодействие современным приемам и способам вербовки в международные террористические организации // Российская юстиция. 2018, №9. С. 51-54.

[20] Гончар В.В. О важности формирования единообразного понятийного аппарата необходимого для расследования преступлений в сфере компьютерной информации// Вестник Экономической безопасности. 2018. №1. С. 225-230.

[21] Орлова А.А., Гончар В.В. Особенности расследования преступлений, совершаемых в сфере информационных технологий // Безопасность бизнеса. 2021. № 4. С. 25-29.

References:

[1] Chen Zhang, Joohee Kim Video object detection with two-path convolutional LSTM pyramid. 2016. URL: https://www.researchgate.net/pub-

lication/343702579_Video_Object_Detection_With_Two-Path_ Convolutional_LSTM_ Pyramid/fulltext/5f3b2658928 51cd302013482/Video-Object-Detection-With-Two-Path-Convolutional-LSTM-Pyramid.pdf/ (access date: 01/16/2026).

[2] Hanson A., Koutilya PNVR, Sanjukta Krishnagopal, Larry Davis Bidirectional Convolutional LSTM for the Detection of Violence in Videos. 2018. URL: https://openaccess.thecvf.com/content_eccv_2018_workshops/w10 /html/Hanson_Bidirectional_Convolutional_LSTM_or_the_ Detection_of_Violence_in_ Videos_ECCVW_2018_paper.html (accessed: 01/15/2026).

[3] Howard A.G., Menglong Zhu, Bo Chen, Kalenichenko D. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017. URL: https://arxiv.org/pdf/1704.04861/ (access date: 01/11/2026).

[4] Iandola F.N., Han S., Moskewicz M.W., Ashraf K., Dally W.J., Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size Recognition. 2016. Published as a conference paper at ICLR. 2015. URL: https://www. researchgate.net/publication/ 301878495_SqueezeNet_AlexNet-level_accuracy_ with_50x_fewer_ parameters_ and_05MB_model_size (access date: 01/10/2026).

[5] Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015. URL: https://arxiv.org/pdf/1502.03167/ (access date: 01/13/2026).

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Deep Residual Learning for Image Recognition. 2015. URL: https://arxiv.org/pdf/1512.03385 (access date: 01/11/2026).

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Spatial pyramid pooling in deep convolutional networks for visual recognition. 2015. URL: https://arxiv.org/pdf/1406.4729/ (access date: 01/11/2026).

[8] Krizhevsky I., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d-6b76c8436e 924a68c45b-Paper.pdf (access date: 01/12/2026).

[9] Lavin A., Scott Gray Fast Algorithms for Convolutional Neural Networks. 2015. URL: https://arxiv.org/abs/1509.09308 (access date: 01/11/2026).

[10] Mikolov T., Kai Chen, Corrado G., Jeffrey Dean Efficient Estimation of Word Representations in Vector Space. 2013. URL: https://arxiv.org/abs/1301.3781 (accessed: 15.01.2026).

[11] Norman P. Jouppi, Cliff Young, Nishant Patil and others. In-Datacenter Performance Analysis of a Tensor Processing Unit. 2017. URL: https://arxiv.org/pdf/1704.04760 (accessed: 11.01.2026).

[12] Ronneberger O., Fischer Ph., Brox Th. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. URL: https://arxiv.org/pdf/1505.04597/ (accessed: 13.01.2026).

[13] Szegedy C., Wei Liu, Chapel Hill, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich Going Deeper with Convolutions. 2014. URL: https://arxiv.org/pdf/1409.4842 (accessed: 12.01.2026)

[14] Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2015. Published as a conference paper at ICLR 2015. URL: https://arxiv.org/pdf/1409.1556 (accessed: 10.01.2026).

[15] Wei Liu, Anguelov D., Erhan D., Szegedy C., Reed S., Cheng-Yang Fu, Berg A.C. SSD: Single Shot MultiBox Detector. 2016. URL: https://arxiv.org/pdf/1512.02325 (accessed: 11.01.2026).

[16] Official website of ONNX. URL: https://onnx.ai/. (accessed: 11.01.2026).

[17] Fundamentals of Artificial Intelligence Technologies: a tutorial / A.A. Semenov, V.V. Gonchar, A.S. Erdniev. - Moscow University of the Ministry of Internal Affairs of Russia named after V. Ya, Kikot, 2025. - 190 p.

[18] Financial Monitoring: a tutorial / V.I. Glotov, A.U. Albekov, O.N. Tisen, et al. – KnoRus, 2022. – 198 p.

[19] O.N. Tisen. Counteracting Modern Techniques and Methods of Recruitment into International Terrorist Organizations // Russian Justice. 2018, No. 9. pp. 51-54.

[20] V.V. Gonchar. On the Importance of Forming a Uniform Conceptual Apparatus Necessary for Investigating Crimes in the Sphere of Computer Information // Bulletin of Economic Security. 2018. No. 1. pp. 225-230.

[21] A.A. Orlova, V.V. Gonchar. Features of Investigating Crimes Committed in the Sphere of Information Technology // Business Security. 2021. No. 4. pp. 25-29.

