ЦИФРОВОЕ ПРАВО

Дата поступления рукописи в редакцию: 02.07.2026 г.
Дата принятия рукописи в печать: 03.08.2026 г.

DOI: 10.24412/2076-1503-2026-7-775-784

СЕМЕНОВ Александр Александрович,

научный сотрудник ФКУ НПО «СТиС» МВД России,

e-mail: mail@law-books.ru

ГОНЧАР Владимир Владимирович,

доцент кафедры информационных технологий и организации расследования киберпреступлений Московская академия Следственного комитета Российской Федерации имени А.Я. Сухарева,

e-mail: vg0778@bk.ru

РАЗВИТИЕ И АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ РЕКУРРЕНТНЫХ И СПАЙКОВЫХ НЕЙРОСЕТЕЙ

Аннотация. *В статье рассматриваются альтернативные архитектуры нейросетей, включая спайковые и рекуррентные модели. Особое внимание уделяется механизмам памяти, обработке последовательных данных и проблемам обучения таких сетей. Описаны принципы работы LSTM и GRU, а также практические аспекты применения рекуррентных нейросетей в задачах анализа последовательностей.*

Ключевые слова: *нейросети, рекуррентные нейросети, RNN, LSTM, GRU, спайковые нейросети, нейроморфные вычисления, последовательности данных, память нейросети, backpropagation through time.*

SEMENOV Alexander Alexandrovich,

Chief Researcher, Federal State Budgetary Institution “STIS” of the Ministry of Internal Affairs of Russia

GONCHAR Vladimir Vladimirovich,

Associate Professor, Department of Information Technology and Cybercrime Investigation Moscow Academy of the Investigative Committee of the Russian Federation named after A. Ya. Sukharev

DEVELOPMENT AND ARCHITECTURAL FEATURES OF RECURRENT AND SPIKING NEURAL NETWORKS

Annotation. *This article examines alternative neural network architectures, including spiking and recurrent models. Particular attention is paid to memory mechanisms, processing of sequential data, and training issues of such networks. The operating principles of LSTM and GRU are described, as well as practical aspects of applying recurrent neural networks to sequence analysis problems.*

Key words: *neural networks, recurrent neural networks, RNN, LSTM, GRU, spiking neural networks, neuromorphic computing, data sequences, neural network memory, backpropagation over time.*

Приступим к изучению разновидностей нейросетей. В свое время, когда еще преобладало мнение, что действительно полезная нейросеть должна быть подобна мышлению человека, полносвязные нейросети с этой точки зрения выглядели недостаточно убедительно, поскольку их аналогия с устройством биологических нейронов была «притянута за уши». В попытке придумать модель нейросети, более близкую к реальности, обратились к знаниям в области биофизики, которые давали следующее представление о том, как работают некоторые нейроны: нейрон принимает от предыдущих нейронов сигналы, складывающиеся в нем, как бы увеличивая его заряд; с течением времени заряд иссякает, но если он достигает определенного уровня, то инициирует передачу этим нейроном сигнала

другим нейронам. Нейросеть, состоящую из таких нейронов, назвали спайковой.

Спайковая нейросеть/импульсная нейросеть – нейросеть, состо. элементарных элементов, моделирующих накопление и ис некоторого заряда с течением времени.

Из описанного механизма вытекает, что в спайковой нейросети посылка одним элементом сигнала следующему элементу зависит не только от свойств элемента, но также и от моментов времени, в которых он в свою очередь получил сигнал. Одно и то же число сигналов может либо привести к испусканию элементом сигнала (именно из этой аналогии на самом деле и проистекает название «функция активации», если входные сигналы получены за короткий промежуток времени, либо не привести, если они распределены во времени. При этом числовыми параметрами одного элемента, определяющими его поведение, являются значение заряда, при котором происходит передача сигнала, объем накапливаемого заряда с получением сигнала от другого элемента и скорость иссякания заряда.

Нейроморфные вычисления – область исследований, в которой пытаются найти совершенные математические выражения для описания устройства человеческого мозга и воплотить их в вычислительной технике для достижения значимых практических результатов.

Можно понять, что в нейросети с таким устройством ключевую роль будет играть именно ее архитектура, поскольку от нее будет зависеть, как минимум, число входов у каждого элемента и, следовательно, вероятность того, что элемент накопит достаточный заряд. Однако практических задач, на которых бы этот класс нейросетей явно продемонстрировал эффективность, пока не нашли, в связи с чем и не определились со способом их обучения – градиентным спуском или чем-то еще. С другой стороны, спайковые нейросети подражают биофизическим процессам лучше других придуманных нейросетей, в чем сторонники аналогии нейросетей с устройством мозга видят надежду на то, что спайковым нейросетям со временем будет найдено существенное применение. Грубо говоря, они предполагают, что спайковые нейросети хороши, просто люди пока не научились их учить.

Приметили также, что устройство спайковой нейросети достаточно хорошо может быть смоделировано аналоговым (т. е. таким, работа которого не разбита строго на такты, как у обычного процессора) физическим устройством, отсюда и появился интерес к разработке нейроморфных процессоров. С другой стороны, любые эксперименты со спайковыми нейросетями, разумеется, могут лежать только в математической плоскости, т. е. требуют конкретных математических моделей и формул, а их реализация (тем более обучение нейросети заранее не известным способом) на аппаратуре требует программируемости и, следовательно, создания цифрового вычислительного устройства. Этим объясняется беспорядок и путаница в области нейроморфных процессоров. Теоретически они должны быть аналоговыми, но делают цифровыми. От них ожидают решения перспективных задач, а каких конкретно и как обучить для решения этих задач нейросети – никто толком не знает.

Если отбросить биофизические аналогии, то от спайковых нейросетей остается следующая идея: элементам нейросети можно придавать некоторые переменные, значения которых могут модифицироваться и изменяться от одного использования нейросети к другому. Такую конструкцию можно интерпретировать как наличие памяти у нейросети, а можно – как генерацию в одном запуске нейросети части входных данных для следующего запуска, т. е. использование нейросетью себя же, в связи с чем за такими нейросетями закрепился термин «рекуррентные».

Рекуррентная нейросеть/RNN – нейросеть с памятью, т. е. такая нейросеть, генерирующая в результате запуска не только выходные значения, но и значения, которые будут использованы в следующем запуске ею же. Рекуррентной может называться не только вся нейросеть, но и ее отдельные слои, блоки или элементы, реализующие описанное свойство.

Эта идея хорошо подходит для задач, в которых данные сгруппированы в последовательности из однотипных переменных, особенно если длины этих последовательностей могут быть разными, например: перевод текстов с одного языка на другой, анализ звуковых данных, предсказание изменений экономических показателей по имеющимся измерениям и т. д.

Попробуем представить себе строение такой нейросети. Рассмотрим слой, у которого есть входные данные X и веса W, после выполнения вычислений некоторой функции f над которыми получаются выходные данные Y (рисунок 1, слева). В общем случае под этой схемой может скрываться любой обучаемый слой с одним входом и одним выходом. Сделаем так, чтобы этот слой умел запоминать последние выходные данные и использовать их при следующих вычислениях. Условимся, что он помнит последние вычисленные выходные данные Y под именем YПР. Функцию f и веса W придется модифицировать так, чтобы в вычислениях функции f совместно использовались X, YПР и W, в результате чего будут получаться выходные значения Y, которые будут передаваться на выход слоя, а также запоминаться для следующего запуска (рисунок 1, справа).

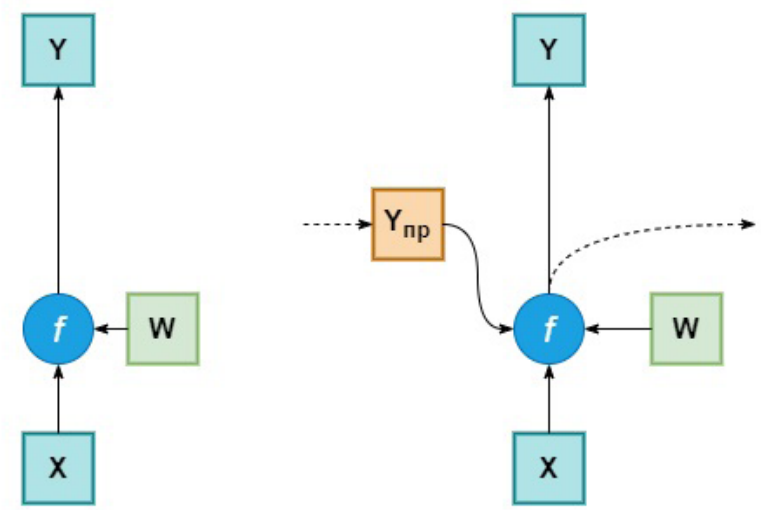


Рисунок 1.Функция и вес.

В общем случае X, W и Y могут быть многомерными массивами, а функция f сколько угодно сложной, т. е. этим способом можно сделать более сложную рекуррентную модификацию из обучаемого слоя любой природы (полносвязный, сверточный и т. д.).

Рассмотрим теперь решение задачи с помощью рекуррентной нейросети. Как было отмечено ранее, такие нейросети хорошо подходят для данных, составляющих некоторую последовательность. Общая схема применения рекуррентной нейросети приведена на рисунке 2. Нейросеть принимает на вход последовательность из некоторого числа (как минимум, одного) значений Xi, но не сразу вся, а по одному элементу за запуск. При этом несколько запусков нейросети нарисованы рядом, а пунктирными стрелками между ними проиллюстрирована передача запомненных значений. Начиная с определенного запуска (хоть с первого, хоть с последнего, хоть с любого другого), ее выходные значения интерпретируются как последовательность из нескольких (как минимум, одного) полезных данных Yj, которая вся попадает в лосс L для оценки качества нейросети (при обучении). В общем случае между рекуррентными слоями могут быть также и обычные слои.

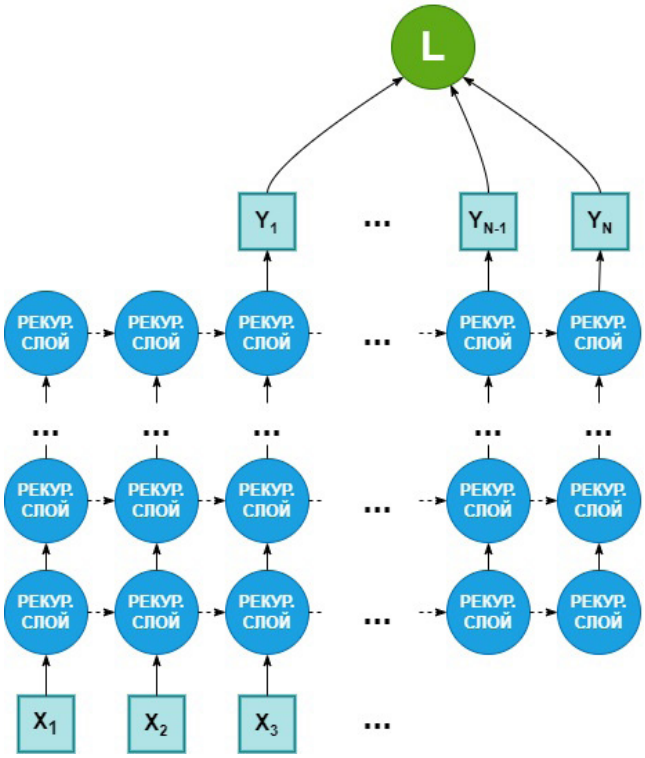


Рисунок 2. Общая схема применения рекуррентной нейросети.

Обратим внимание, что реализовать обучение такой нейросети методом обратного распространения ошибки не составит труда: оно будет выглядеть так же, как обучение нейросети, состоящей из нескольких параллельных ветвей, сходящихся к концу. Однако есть одно существенное отличие: на рисунке 2 одни и те же слои и, следовательно, их веса, отображены несколько раз, т. е. при вычислении градиента по весам каждое значение градиента будет состоять из нескольких слагаемых, по одному на каждый запуск нейросети. Эта техника применения обратного распространения ошибки называется «обратным распространением ошибки сквозь время» – backpropagation through time. Чтобы данный процесс не был бесконечным, его замыкают, для чего во время обучения ограничиваются последовательностями фиксированной длины. При этом начальные значения памяти каждого слоя заполняют чем-то, чаще всего – нулями, также поступают с входными данными запусков после того, как в нейросеть поступило последнее осмысленное значение Xi, а выходные данные нейросети до Y1 попросту игнорируют.

Процесс обучения рекуррентной нейросети мы описали, алгоритмически он все же получается более запутанным, чем для обычных, поэтому применяемые на практике слои несколько сложнее, чем на рисунке 1, рассмотрим их устройство.

Сверточные нейросети до определенного момента страдали от свойства градиентного спуска скорее изменять значения слоев в конце нейросети, нежели слоев в начале. В процессе обучения рекуррентных нейросетей эта проблема проявляется не только при большой длине нейросети, но также и при длинных последовательностях: градиентный спуск может изменять значения весов так, чтобы они реагировали в большей степени на последние попавшие в нейросеть данные, нежели на попавшие ранее. Как следствие, рекуррентные нейросети плохо работали с длинными последовательностями. Чтобы решить такую проблему, применили механизм: придумали от запуска к запуску запоминать в слое не только предыдущее выходное значение Y, но и некоторое значение S, которое бы в качестве слагаемого содержало, в том числе, аналогичное значение с предыдущего запуска SПР, схема подобного слоя приведена на рисунке 3, знаком «+» обозначено сложение двух массивов.

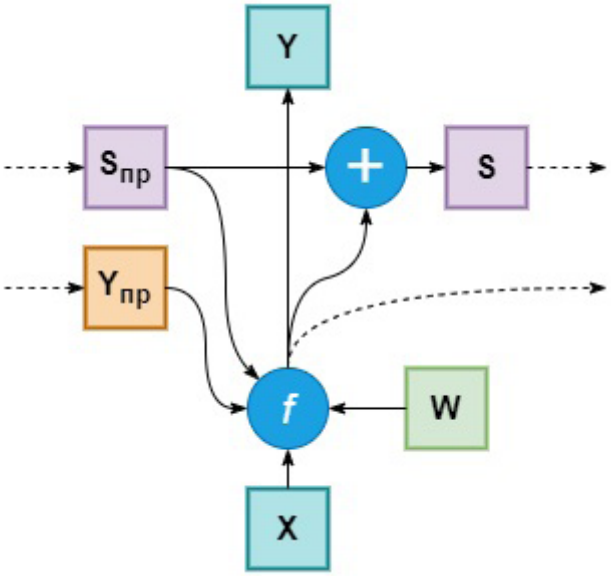


Рисунок 3. Схема слоя со значением S.

В разных практических задачах могут возникать разные последовательности с разным характером взаимосвязей между ними.

В некоторых последовательностях длина связи не влияет на ее значимость (например, связь слов в длинном сложноподчиненном предложении), в некоторых, напротив, может быть связана с ней на прямую (например, в последовательности видеокадров). Чтобы как-то навязать нейросети ту логику работы, которая (по предположению разработчика) лучше всего соответствует особенностям конкретной задачи, в рекуррентный слой добавляют дополнительные функции с определенными ролями. Пример такого построения с формулами приведен на рисунке 4, в нем:

буквой «σ» обозначена сигмоида, tanh означает функцию гиперболического тангенса;

символом «+» обозначено поэлементное сложение двух массивов одинаковой формы, символом «Ø» – поэлементное умножение двух массивов одинаковой формы;

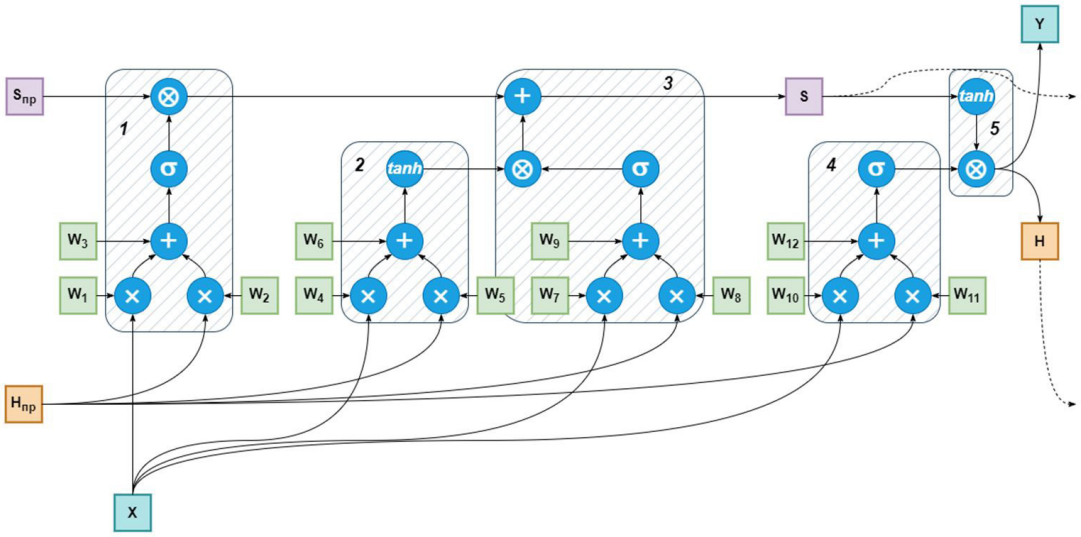
символом «×», в зависимости от типа данных, обозначено умножение двух чисел, умножение матрицы весов на вектор данных или выполнение операции свертки.

Как видно, один такой рекуррентный слой фактически состоит из нескольких других слоев. Когда его придумывали, рассуждали так:

Информация о нескольких предыдущих запусках, хранящаяся в S, может быть полезна или нет. Чтобы нейросеть могла отбросить эту информацию в необходимых обстоятельствах, нужно сделать так, чтобы эта информация умножалась на коэффициенты в диапазоне от 0 до 1, зависящие от X и НПР. Для этого (заштрихованная зона 1 на рисунке 4) выполняют перемножение этих данных с весами W1 и W2, добавляют к результату биас (биасы) W3, после чего к полученному массиву применяют сигмоиду, а полученные после нее значения домножают поэлементно на значения SПР.

Чтобы контролировать, насколько большую долю вновь вычисленные значения (заштрихованная зона 2 на рисунке 4) буду составлять в значении S, нужно эти значения домножить на коэффициенты в диапазоне от 0 до 1, зависящие от X и НПР, и только тогда прибавить к SПР (заштрихованная зона 3 на рисунке 4).

Только часть полученного значения S должна быть передана в качестве выходного значения, эту долю определят коэффициенты в диапазоне от 0 до 1, зависящие от X и НПР (заштрихованая зона 4), которые будут потом поэлементно умножены на значения S, предварительно подвергнутые преобразованию функцией гиперболического тангенса (заштрихованная зона 5 на рисунке 4).



$$G_1 = \sigma(W_1 \times X + W_2 \times H_{\text{ПР}} + W_3),$$
$$G_2 = \sigma(W_7 \times X + W_8 \times H_{\text{ПР}} + W_9),$$
$$G_3 = \sigma(W_{10} \times X + W_{11} \times H_{\text{ПР}} + W_{12}),$$
$$S = G_1 \otimes S_{\text{ПР}} + G_2 \otimes tanh(W_4 \times X + W_5 \times H_{\text{ПР}} + W_6), H = Y = G_3 \otimes tanh(S)$$

Рисунок 4. Пример построения с формулами.

Как видно, функциям G1, G2 и G3 в приведенном примере придается как бы роль светофоров. При этом нет никакой гарантии, что математический смысл операций в обученной нейросети будет соответствовать той логике, которая была придана изначально при составлении слоя. Такое распределение ролей существует только в воображении разработчика и нужно для того, чтобы помочь ему не запутаться и гипотетически обосновать, что сконструированный рекуррентный слой может полностью выполнить свою часть задачи. Функции указанного типа называются «воротами».

Ворота/вентили/глазки/gates – комбинации функций в рекуррентных слоях, которым при построении рекуррентного слоя мысленно придается какая-то регулирующая роль. Разные схемы построения рекуррентных слоев отличаются, в первую очередь, разным количеством и порядком использования ворот и, как следствие, немного разными свойствами.

Hidden state и memory cell – термин, которым в контексте рекуррентных слоев иногда называют сохраненное выходное значение H и передаваемое значение S, соответственно. Обозначаются чаще всего, соответственно, H и C (автор использовал в тексте букву «S» по той причине, что в русскоязычном тексте латинская буква «C» легко путается с кириллической).

LSTM/Long short-term memory – схема построения рекуррентного слоя, в которой, помимо hidden state, есть memory cell. Существуют разные вариации LSTM с разным набором ворот.

GRU/Gated recurrent unit – наиболее популярная схема построения рекуррентного слоя среди отличных от LSTM. Существуют разные вариации GRU с разным набором ворот.

В зависимости от того, какие данные обрабатываются рекуррентным слоем, выбирается соответствующая операцию «×». Например, если нейросети обрабатывает картинки, то X и Y будут трехмерными картами признаков, соответственно, также трехмерными картами признаков будут S и H. В таком случае один рекуррентный слой будет содержать в себе восемь сверточных слоев, пять слоев активации, один слой поэлементного сложения (eltwise) и три слоя поэлементного умножения.

При этом все восемь сверточных слоев выполняются параллельно, не друг за другом, т. е. они не позволяют выстроить несколько уровней признаков, как это позволили бы сделать восемь последовательных слоев. Таким образом, для получения признака определенного уровня понадобится столько же рекуррентных слоев, сколько понадобилось бы не рекуррентных.

Как видно, замена полносвязной или сверточной нейросети на рекуррентный аналог усложняет нейросеть на порядок. Однако на практике так делать и не нужно. Вместо этого пытаются по природе задачи интуитивно предположить, на каком уровне или нескольких уровнях признаков целесообразно сопоставление данных признаков между несколькими запусками нейросети, в соответствии с чем в нейросети предусматривают один или несколько рекуррентных слоев, а оставшиеся части нейросети составляют из обычных слоев. Чем больше таких уровней, тем больше рекуррентных слоев и меньше обычных. Соответственно, полностью рекуррентные нейросети целесообразнее применять в тех задачах, в которых вся содержательная информация раскрывается именно в последовательности входных данных, а не в отдельной единице (например, при классификации текста).

Примерами задач, решаемых как с помощью обычных нейросетей, так и с помощью рекуррентных, является обнаружение объектов на видео или классификация видео. С помощью обычной сверточной нейросети можно обработать каждый кадр видео, но часть информации может содержаться только непосредственно в последовательности кадров (например, направление движения предметов на видео). Поэтому полезно каким-то образом сопоставлять результаты обработки нескольких кадров. Если при этом особенности задачи не диктуют какой-нибудь очевидный алгоритм такого сопоставления, то поиск этого алгоритма целесообразно возложить на нейросеть.

Для задачи классификации видео (например, определения темы видеоролика) можно предложить архитектуру нейросети, которая будет выполнять сверточную обработку кадра и после сопоставлять выявленные признаки между несколькими кадрами, в результате чего давать признаки классов (рис. 9.5, сверху). В задаче детекции объектов объекты разного размера могут обнаруживаться на разных слоях нейросети, принимая это во внимание можно составить сверточную нейросеть, в которой после каждого пулинга будет следовать один рекуррентный сверточный слой, а после последнего будет обычная для задачи детекции постобработка. Например, первая схема успешно применена для выявления сцен насилия 13F [2], а вторая – для выявления объектов, масштаб которых может в разы меняться от кадра к кадру (рисунок 5) 14F [1].

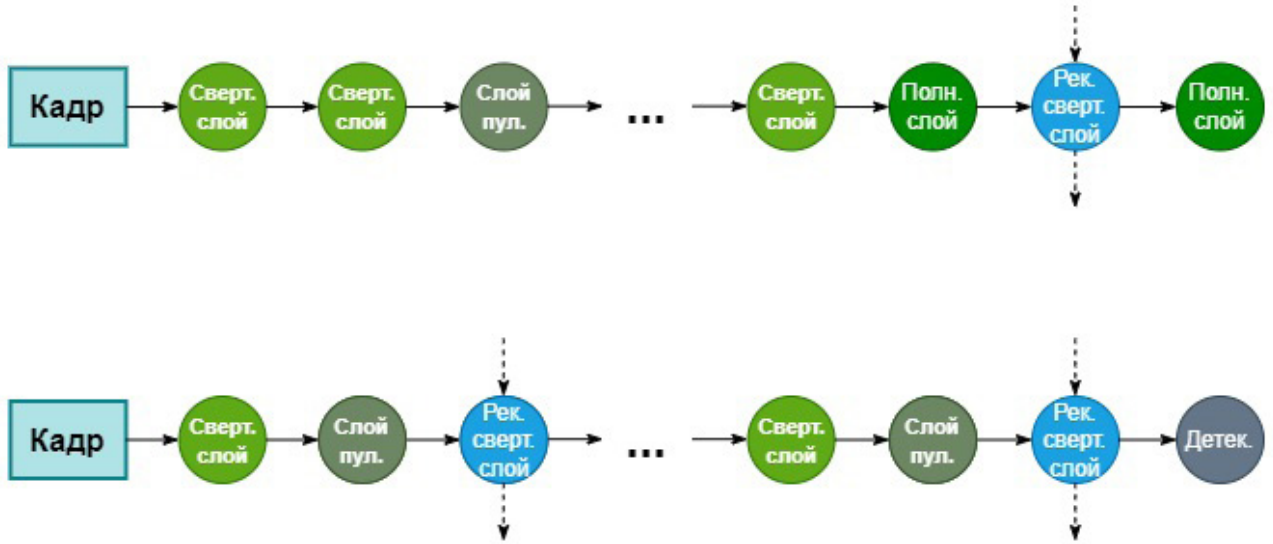


Рисунок 5. Схемы выявления нейросетями сцен.

В первом приведенном примере авторы применили двустороннюю рекурсию. Этот прием целесообразен в случаях, когда можно себе представить анализ (или генерацию) последовательности как в прямом, так и в обратном порядке (например, если нейросеть используется для перевода текстов с языка на язык, а в языках используется разный порядок членов предложения). При использовании такого приема значения Y и H уже не совпадают, а вместо одного значения H имеются два значения H1 и H2, одно из которых передается в одном направлении, а другое – в противоположном. Значение Y вычисляется на основе обоих значений H1 и H2. Наиболее простая схема реализации этого принципа без деталей приведена на рисунке 6.

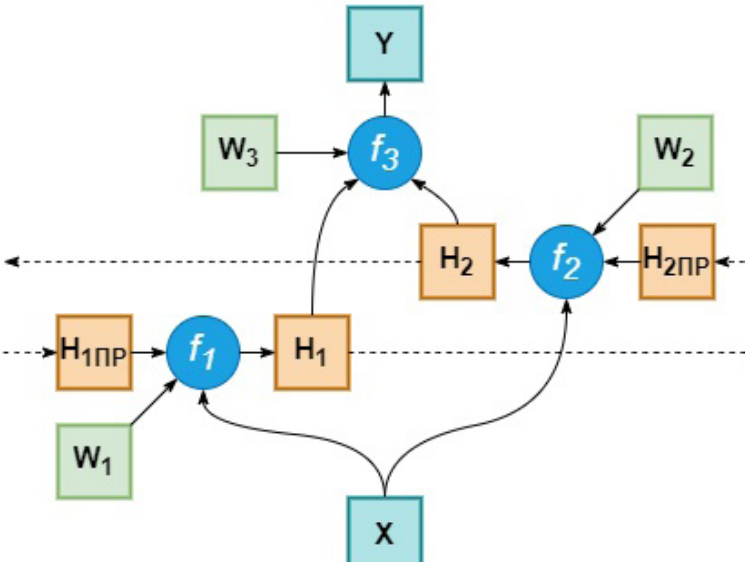


Рисунок 6. Простая схема реализации принципа без деталей.

Двусторонняя рекурсия не мешает применению метода обратного распространения ошибки, но несколько затрудняет процесс: просчет невязок так, как это показано на рисунке 2, может выполняться только строго от слоя к слою, при чем на каждом слое часть невязок, связанных со значением H1, вычисляются в одном направлении (от первого запуска к последнему), а часть, связанная с H2, наоборот, в обратном.

Список литературы:

[1] Chen Zhang, Joohee Kim Video object detection with two-path convolutional LSTM pyramid. 2016. // Research Gate. – URL: https://www.researchgate.net/publication /343702579_ Video_Object_Detection_With_ Two-Path_ Convolutional_LSTM_ Pyramid/fulltext/5f3b265892851cd302013482/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. // CVF Open Access. – 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. // arXiv.org. – 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. // Research Gate. – URL: https://www. researchgate.net/publication/ 301878495_SqueezeNet_AlexNet-level_accuracy_ with_50x_fewer_ parameters_ 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. // arXiv.org. – 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. // arXiv.org. – 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. // arXiv.org. – 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. // NeurIPS Proceedings. – 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. // arXiv.org. – 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. // arXiv.org. – 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. // arXiv.org. – 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. // arXiv.org. – 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. // arXiv.org. – 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. // arXiv.org. – 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. // arXiv.org. – 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] Овчинский А.С., Аветисян К.Р. Аудиальные воздействия на участников массовых мероприятий как объект административно-правового регулирования// Вестник Московского университета МВД. 2015. № 1-. С. 293-297.

[21] Зиборов О.В., Аветисян К.Р. Комплексный подход при противоборстве организации массовых беспорядков// Вестник Московского университета МВД. 2023. № 7. С. 100-104.

References:

[1] Chen Zhang, Joohee Kim Video object detection with two-path convolutional LSTM pyramid. 2016. // Research Gate. – 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. // CVF Open Access. – 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. // arXiv.org. - URL: https://arxiv.org/pdf/1704.04861/ (access date: 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. // Research Gate. – URL: https://www. researchgate.net/publication/ 301878495_SqueezeNet_AlexNet-level_accuracy_ with_50x_fewer_ parameters_ 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. // arXiv.org. - URL: https://arxiv.org/pdf/1502.03167/ (access date: 13.01.2026).

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

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

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

[9] Lavin A., Scott Gray Fast Algorithms for

Convolutional Neural Networks. 2015. // arXiv.org. - URL: https://arxiv.org/abs/1509.09308 (access date: 11.01.2026).

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

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

[12] Ronneberger O., Fischer Ph., Brox Th. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. // arXiv.org. - URL: https://arxiv.org/pdf/1505.04597/ (access date: 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. // arXiv.org. - URL: https://arxiv.org/pdf/1409.4842 (accessed on: 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. // arXiv.org. - URL: https://arxiv.org/pdf/1409.1556 (access date: 10.01.2026).

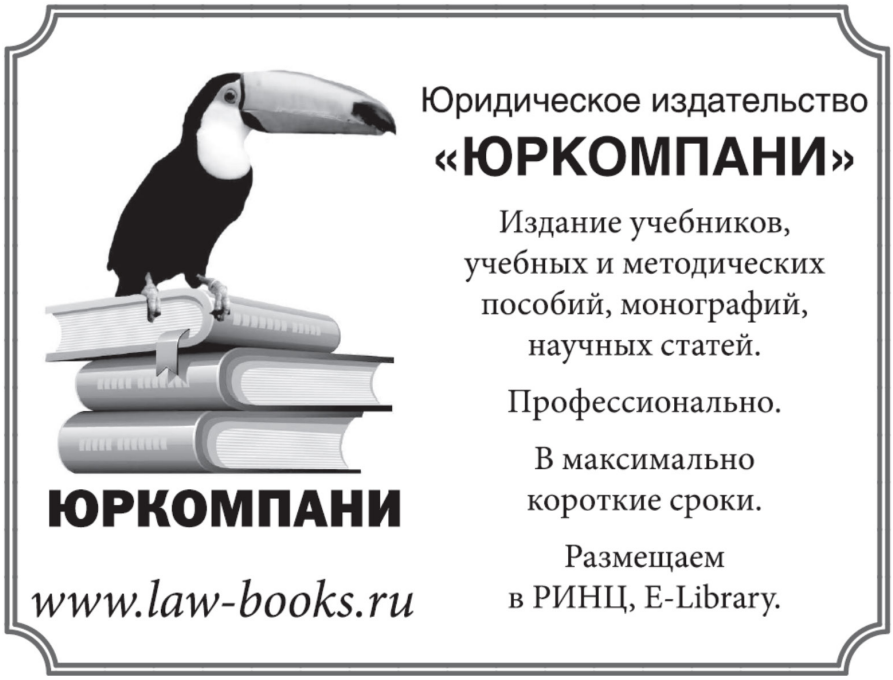
[15] Wei Liu, Anguelov D., Erhan D., Szegedy C., Reed S., Cheng-Yang Fu, Berg A.C. SSD: Single Shot MultiBox Detector. 2016. // arXiv.org. - URL: https://arxiv.org/pdf/1512.02325 (access date: 11.01.2026).

[16] ONNX official website. - URL: https://onnx.ai/ (access date: 11.01.2026).

[17] Fundamentals of artificial intelligence technologies: a textbook/A.A. Semenov, V.V. Gonchar, A.S. Erdniev. - Moscow University of the Ministry of Internal Affairs of Russia named after V.Ya, Kikotya, 2025. - 190s.

[18] Financial monitoring: training manual/V.I. Glotov, A.U. Albekov, O.N. Tisen et al. - KnoRus, 2022. - 198 s.

[19] Tisen O.N. Countering modern methods and methods of recruitment to international terrorist organizations//Russian Justice. 2018, №9. S. 51-54.

[20] Ovchinsky A.S., Avetisyan K.R. Audio effects on participants of mass events as an object of administrative and legal regulation//Bulletin of the Moscow University of the Ministry of Internal Affairs. 2015. № 1-. S. 293-297.

[21] Ziborov O.V., Avetisyan K.R. An integrated approach to confronting the organization of mass riots//Bulletin of the Moscow University of the Ministry of Internal Affairs. 2023. № 7. S. 100-104.