МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теоретических основ компьютерной безопасности и криптографии

**Обзор современных классов нейронных сетей**

РЕФЕРАТ

студентки 5 курса 531 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Богатовой Екатерины Дмитриевны

|  |  |  |
| --- | --- | --- |
| Старший преподаватель, доцент | \_\_\_\_\_\_\_\_\_\_ | И.И. Слеповичев |
|  | подпись, дата |  |

Саратов 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc104920839)

1. [Теоретический анализ нейронных сетей. Нейроны и искусственные нейронные сети 4](#_Toc104920841)

1.1 Нейроны [4](#_Toc104920843)

1.2 Функции активации  [6](#_Toc104920845)

1.3 Структура нейронной сети  [8](#_Toc104920846)

1. Современные классы нейронных сетей  [11](#_Toc104920840)

2.1 Классификация по топологии [11](#_Toc104920843)

2.2 Классификация по характеру обучения НС [14](#_Toc104920845)

2.3 Классификация по типу входной информации  [17](#_Toc104920843)

2.4 Классификация по типу настройки весов  [17](#_Toc104920845)

2.5 Классификация по характеру связей и архитектуре  [17](#_Toc104920843)

[ЗАКЛЮЧЕНИЕ 43](#_Toc104920839)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 44](#_Toc104920847)

**ВВЕДЕНИЕ**

Современные нейронные сети стали неотъемлемой частью искусственного интеллекта, обеспечивая значительные прорывы в различных областях науки и технологий. Они являются математическими моделями, созданными для имитации работы мозга человека, и успешно применяются. От компьютерного зрения и обработки естественного языка до генерации контента и управления сложными системами – нейронные сети демонстрируют исключительные возможности в решении задач, которые ранее считались недостижимыми.

Разнообразие классов нейронных сетей позволяет адаптировать их под уникальные требования задач. Сетевые архитектуры, такие как сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN), трансформеры и множество других отличаются по своей структуре, принципам работы и областям применения. Это разнообразие стало возможным благодаря интенсивным исследованиям, направленным на улучшение точности, производительности и эффективности моделей.

Цель данного реферата – предоставить обзор современных классов нейронных сетей, проанализировать их ключевые особенности, обсудить их преимущества и ограничения, а также рассмотреть примеры реальных применений. Данный анализ поможет лучше понять, как различные архитектуры нейронных сетей способствуют решению сложных задач в науке, бизнесе и повседневной жизни.

1. **Теоретический анализ нейронных сетей. Нейроны и искусственные нейронные сети**
   1. **Нейроны**

Под нейронными сетями подразумеваются вычислительные структуры, которые моделируют простые биологические процессы, обычно ассоциируемые с процессами в человеческом мозге. Они представляют собой распределённые и параллельные системы, способные к адаптивному обучению путём анализа положительных и отрицательных воздействий. Элементарным преобразователем в этих сетях является искусственный нейрон или просто нейрон, названный так по аналогии с биологическим прототипом.

То есть искусственные нейронные сети – это машинная интерпретация мозга человека, в котором находятся нейроны, передающие информацию в виде электрических импульсов.

Нейрон представляет собой единицу обработки информации в нейронной сети. Основная последовательность действий одного нейрона:

* Прием сигналов от предыдущих элементов сети
* Комбинирование входных сигналов
* Вычисление выходного сигнала
* Передача выходного сигнала следующим элементам нейронной сети

Рассмотрим поподробнее этапы работы сети.

Поскольку к каждому нейрону могут приходить несколько входных сигналов, то при моделировании нейронной сети необходимо задать определенное правило комбинирования их всех. Часто используется правило суммирования взвешенных значений связей. На Рисунке 1 показана модель нейрона, лежащего в основе искусственных нейронных сетей.

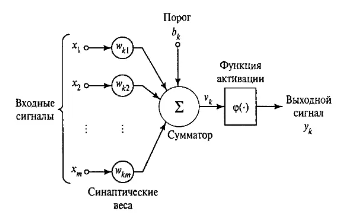


Рисунок 1 – Структура искусственного нейрона

В этой модели можно выделить три основных элемента:

1. Набор *синапсов* (synapse) или *связей*, каждый из которых характеризуется своим *весом* или *силой*. В частности, сигнал на входе синапса , связанного с нейроном , умножается на вес . Важно обратить внимание на то, в каком порядке указаны индексы синаптического веса . Первый индекс относится к рассматриваемому нейрону, а второй – к входному окончанию синапса, с которым связан данный вес. В отличие от синапсов мозга синаптический вес искусственного нейрона может иметь как положительные, так и отрицательные значения.
2. *Сумматор* (adder) складывает входные сигналы, взвешенные относительно соответствующих синапсов нейрона. Эту операцию можно описать как линейную комбинацию.
3. *Функция активации* (activation function) ограничивает амплитуду выходного сигнала нейрона. Эта функция также называется функцией *сжатия*. Обычно нормализованный диапазон амплитуд выхода нейрона лежит в интервале [0, 1] или [-1,1].

В рассматриваемую модель нейрона также включен *пороговый элемент* (bias), который обозначен символом . Эта величина отражает увеличение или уменьшение входного сигнала, подаваемого на функцию активации. Использование порога обеспечивает эффект *афинного преобразования* выхода линейного сумматора.

В математическом представлении функционирование нейрона можно описать следующей парой уравнений:

где – входные сигналы; – синаптические веса нейрона ; – линейная комбинация входных воздействий; – порог; – функция активации; – выходной сигнал нейрона.

Также в модели присутствует обозначение – это постсинаптический потенциал или потенциал активации, который вычисляется как:

* 1. **Функции активации**

Функции активации, представленные в формулах как определяют выходной сигнал нейрона в зависимости от потенциала активности . Существует три основных типа функции активации:

1. Функция *единичного скачка,* в технической литературе ее также называют *функцией Хэвисайда:*

Эту модель называют моделью МакКаллока-Питтса (именно они в 1943 году ввели понятие «нейронные сети» в фундаментальной статье о логическом исчислении идей и нервной активности), где выходной сигнал нейрона принимает значение 1, если неотрицательно, и 0 – в противном случае.

1. *Кусочно-линейная функция:*
2. *Сигмоидальная функция,* является самой распространенной функцией, используемой для создания искусственных нейронных сетей:

где – параметр наклона сигмоидной функции

Разновидностью является *гиперболический тангенс,*

Последние две функции активации обеспечивают более плавное изменение выходного сигнала нейрона. [1]

* 1. **Структура нейронной сети**

Нейроны собираются в сеть по слоям. Каждый слой состоит из узлов (нейронов), которые связаны друг с другом. Слои работают вместе для обработки данных с помощью серии преобразований. Проектирование связей между нейронами эквивалентно программированию системы для обработки входа и создания желаемого выхода.

Проектирование нейронной сети состоит из следующих этапов:

1) Упорядочение нейронов по слоям;

2) Определение связей между нейронами различных слоев, так же как и между нейронами внутри слоя;

3) Решение о том, каким образом нейрон получает вход и как создает выход;

4) определение силы связей внутри сети, чтобы узнать соответствующие значение весов путем использования контрольного набора данных.

Существует много способов объединения искусственных нейронов в сеть, для чего используются различные типы связей:

* *Полная связь* – каждый нейрон первого слоя связан с каждым нейроном второго слоя;
* *Частичная связь* – каждый нейрон первого слоя не обязательно должен быть связан со всеми нейронами второго слоя;
* *Однонаправленная связь* – нейроны первого слоя посылают выходы нейронам второго слоя, но не наоборот (может быть полной и частичной);
* *Д*вунаправленная связь – нейроны первого слоя связаны с нейронами второго слоя, которые, в свою очередь, имеют связь с нейронами первого слоя (может быть полной и частичной);
* Иерархическая связь – нейроны одного слоя связаны только с нейронами следующего слоя. Если связь не иерархическая, то нейроны одного слоя могут посылать свои выходы не только нейронам следующего слоя, но и другим слоям. Внутри одного слоя группируются нейроны одного типа, которые могут иметь или не иметь связи между собой. Обычно во многих вариантах топологий нейроны одного слоя не связаны между собой [4].

В зависимости от того, какой вид ИНС, определяется количество слоев. На Рисунке 2 приведена однослойная нейронная сеть, а на Рисунке 3 приведена многослойная нейронная сеть.

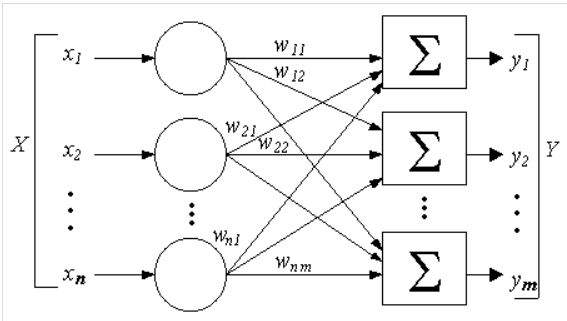


Рисунок 2 - Структура однослойной нейронной сети

1. *Однослойные нейронные сети*

* *Входной слой* – отвечает за получение необработанных входных данных. Выполняет только одну задачу — распределение входных сигналов остальным нейронам. Не выполняет каких-либо вычислений, и поэтому не будут считаться слоем.
* *Выходной слой –* сигналы с входного слоя сразу подаются на выходной. Он преобразует сигнал и сразу выдает результат.

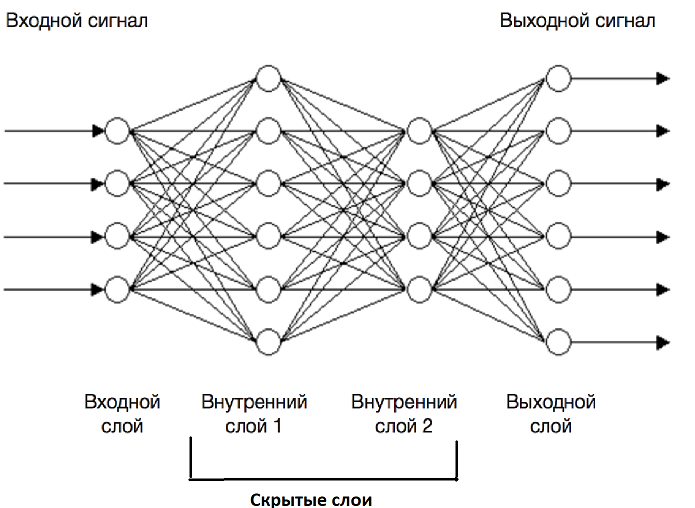


Рисунок 3 – Структура многослойной нейронной сети

1. *Многослойные нейронные сети*

* *Входной слой* – принимает данные в их исходной форме. Нейроны этого уровня соответствуют характеристикам входных данных. Например, при обработке изображений входной слой будет иметь нейроны для каждого значения пикселя.
* *Скрытый слой* – являются промежуточным между входным и выходным уровнями. Количество и размер скрытых слоев могут варьироваться в зависимости от сложности задачи. Они выполняют большую часть вычислений, необходимых сети. Нейроны получают входные данные либо от нейронов входного слоя, либо от нейронов ранее скрытого слоя, применяет функцию активации и после этого отправляет выходные данные нейронам следующего уровня.
* *Выходной слой* – получает данные от последнего скрытого слоя и производит выходной сигнал [2].

1. **Современные классы нейронных сетей**

Существуют различные классификации нейронных сетей. Рассмотрим основные из них.

* 1. **Классификация по топологии**

С точки зрения топологии можно выделить следующие основные типы нейронных сетей.

* + 1. **Полносвязные НС**

В полносвязных нейронных сетях каждый нейрон передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные сигналы подаются всем нейронам. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Примеры полносвязных НС приведены на Рисунке 4.

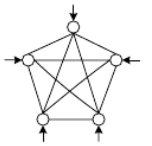
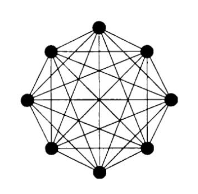


Рисунок 4 – Пример полносвязной нейронной сети

Полносвязные нейронные сети широко используются в задачах классификации и прогнозирования, например, в распознавании рукописного текста или в прогнозировании финансовых рынков

* + 1. **Неполносвязные НС**

Такие сети описываются неполносвязным ориентированным графом и обычно называются перцептронами. Подразделяются на однослойные (простейшие перцептроны) и многослойные, с прямыми, перекрестными и обратными связями. В нейронных сетях с прямыми связями нейроны j-ого слоя по входам могут соединяться только с нейронами нижележащих слоев. В нейронных сетях с перекрестными связями допускаются связи внутри одного слоя.

* + - 1. **Монотонные многослойные НС**

Это частный случай слоистых сетей с дополнительными условиями на связи и нейроны. Каждый слой кроме последнего (выходного) разбит на два блока: возбуждающий и тормозящий. Связи между блоками тоже разделяются на тормозящие и возбуждающие. Если от нейронов блока А к нейронам блока В ведут только возбуждающие связи, то это означает, что любой выходной сигнал блока является монотонной неубывающей функцией любого выходного сигнала блока. Если же эти связи только тормозящие, то любой выходной сигнал блока В является невозрастающей функцией любого выходного сигнала блока А. Для нейронов монотонных сетей необходима монотонная зависимость выходного сигнала нейрона от параметров входных сигналов.

* + - 1. **НС без обратных связей**

В таких сетях нейроны входного слоя получают входные сигналы, преобразуют их и передают нейронам первого скрытого слоя, и так далее вплоть до выходного, который выдает сигналы для интерпретатора и пользователя. Если не оговорено противное, то каждый выходной сигнал -го слоя подается на вход всех нейронов го слоя. Однако возможен вариант соединения q-го слоя с произвольным слоем. Среди многослойных сетей без обратных связей различают полносвязные (выход каждого нейрона –го слоя связан с входом каждого нейрона –го слоя) и частично полносвязные. Классическим вариантом слоистых сетей являются полносвязные сети прямого распростанения.

* + - 1. **Сети с обратными связями**

В сетях с обратными связями информация с последующих слоев передается на предыдущие. Различают следующие типы нейронных сетей с обратными связями:

* *Слоисто-циклические* – отличаются тем, что слои замкнуты в кольцо: последний слой передает свои выходные сигналы первому. Все слои равноправны и могут как получать входные сигналы, так и выдавать выходные;
* *Слоисто-полносвязные* – состоят из слоев, каждый из которых представляет собой полносвязную сеть, а сигналы передаются как от слоя к слою, так и внутри слоя. В каждом слое цикл работы распадается на три части: прием сигналов с предыдущего слоя, обмен сигналами внутри слоя, выработка выходного сигнала и передача к следующему слою;
* *Полносвязно-слоистые* – по своей структуре аналогичные слоисто-полно-связным, но функционирующим по-другому: в них не разделяются фазы обмена внутри слоя и передачи следующему, на каждом такте нейроны всех слоев принимают сигналы от нейронов как своего слоя, так и последующих.
  + 1. **Слабосвязные НС**

В слабосвязных нейронных сетях нейроны располагаются в узлах прямоугольной или гексагональной решетки. Каждый нейрон связан с четырьмя (окрестность фон Неймана), шестью (окрестность Голея) или восемью (окрестность Мура) своими ближайшими соседями. На Рисунке 5 приведены примеры таких НС.

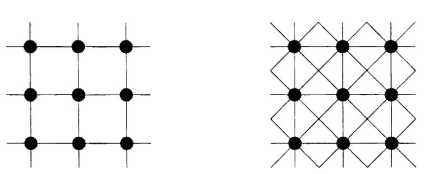


Рисунок 5 – Примеры слабосвязных нейронных сетей

Известные нейронные сети можно разделить по типам структур нейронов на *гомогенные* (однородные) и *гетерогенные* (неоднородные*)*. Гомогенные сети состоят из нейронов одного типа с единой функцией активации, а в гетерогенную сеть входят нейроны с различными функциями активации [5].

**2.2 Классификация по характеру обучения НС**

Рассмотрим парадигмы обучения нейронных сетей.

* + 1. **Обучение с учителем**

Начнем с парадигмы *обучения с учителем* (supervised learning). На рисунке 6 показана блочная диаграмма, иллюстрирующая эту форму обучения.

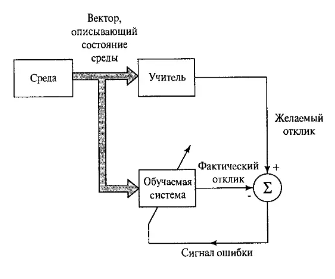


Рисунок 6 – Блочная диаграмма обучения с учителем

Концептуально участие учителя можно рассматривать как наличие знаний об окружающей среде, представленных в виде пар вход-выход. При этом сама среда неизвестна обучаемой нейронной сети. Теперь предположим, что учителю и обучаемой сети подается обучающий вектор из окружающей среды. На основе встроенных знаний учитель может сформировать и передать обучаемой нейронной сети желаемый отклик, соответствующий данному входному вектору. Этот желаемый результат представляет собой оптимальные действия, которые должна выполнить нейронная сеть. Параметры сети корректируются с учетом обучающего вектора и сигнала ошибки. *Сигнал ошибки* – это разность между желаемым сигналом и текущим откликом нейронной сети. Корректировка параметров выполняется пошагово с целью имитации нейронной сетью поведения учителя.

Эта эмуляция в некотором статистическом смысле должна быть оптимальной. Таким образом, в процессе обучения знания учителя передаются в сеть в максимально полном объеме. После окончания обучения учителя можно отключить и позволить нейронной сети работать со средой самостоятельно.

Описанная форма обучения с учителем является ничем иным, как обучением на основе коррекции ошибок. Это замкнутая система с обратной связью, которая не включает в себя окружающую среду. Производительность такой системы можно оценивать в терминах среднеквадратической ошибки или суммы квадратов ошибок на обучающей выборке, представленной в виде функции от свободных параметров системы. Для повышения про-извинительности системы во времени значение ошибки должно смещаться в сторону.

* + 1. **Обучение без учителя**

Альтернативная парадигма обучения без учителя (learning without a teacher) самим названием подчеркивает отсутствие руководителя, контролирующего процесс настройки весовых коэффициентов. При использовании такого подхода не существует маркированных примеров, по которым проводится обучение сети.

На рисунке 7 показана блочная диаграмма, иллюстрирующая эту форму обучения.



Рисунок 7 – Блочная диаграмма обучения без учителя

В этом случае существует лишь независимая от задачи мера качества представления, которому должна научиться нейронная сеть, и свободные параметры сети оптимизируются по отношению к этой мере. После обучения сети на статистические закономерности входного сигнала она способна формировать внутреннее представление кодируемых признаков входных данных и, таким образом, автоматически создавать новые классы [1]. То есть обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, а именно чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы.

* + 1. **Обучение с подкреплением**

В обучении с подкреплением (reinforcement learning, RL) формирование отображения входных сигналов в выходные выполняется в процессе взаимодействия с внешней средой с целью минимизации скалярного индекса производительности. На рисунке 8 показана блочная диаграмма одной из форм системы обучения с подкреплением.

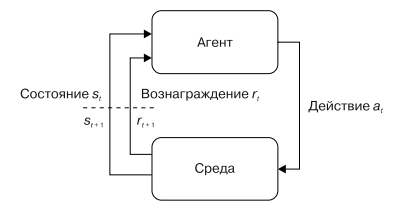


Рисунок 8 – Блочная диаграмма обучения с подкреплением

Задачи RL могут быть представлены как система, состоящая из агента и среды. Среда предоставляет информацию, описывающую состояние системы. Агент взаимодействует со средой, наблюдая состояние и используя данную информацию при выборе действия. Среда принимает действие и переходит в следующее состояние, а затем возвращает агенту следующее состояние и вознаграждение. Когда цикл «состояние → действие → вознаграждение» завершен, предполагается, что сделан один шаг. Цикл повторяется, пока среда не завершится, например, когда задача решена.

Основываясь на таком взаимодействии с окружающей средой, агент должен выработать стратегию, позволяя максимизировать величину вознаграждения [3].

* 1. **Классификация по типу входной информации**

Существуют также *бинарные, аналоговые* и *цифровые сети*. Первые из них – оперируют только двоичными сигналами, и вход каждого нейрона может принимать значение либо логического нуля (заторможенное состояние), либо логической единицы (возбужденное состояние). Вторые – входная информация представлена в форме действительных чисел (использование сигмоидальных функций позволило перейти от бинарных выходов нейрона к аналоговым). Третьи – оперируют информацией, представленной в цифровом виде [5].

* 1. **Классификация по типу настройки весов**
* *Сети с фиксированными связями* – весовые коэффициенты нейронной сети выбираются сразу, исходя из условий задачи. При этом

Где – характеризует весовые коэффициенты сети

* *Сети с динамическими связями* – для них в процессе обучения происходит настройка синаптических весов, т.е
  1. **Классификация по характеру связей и архитектуре**
     1. **С прямыми связями**

В них происходит распространение информации только в одном направлении от слоя к слою, от входных узлов, через скрытые узлы (если они есть), к выходным узлам, как показано на Рисунке 9. – характеризует синаптические связи, а – оператор нелинейного преобразования. То есть соединения между узлами не образуют цикл. Такая сеть была первым и самым простым типом искусственной нейронной сети.

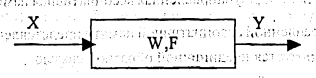


Рисунок 9 – НС с прямыми связями

Они составляют основу многих важных нейронных сетей, используемых в последнее время, таких как [сверточные нейронные сети](https://ru.wikipedia.org/wiki/Special:Search/%D1%81%D0%B2%D0%B5%D1%80%D1%82%D0%BE%D1%87%D0%BD%D1%8B%D0%B5%20%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5%20%D1%81%D0%B5%D1%82%D0%B8) (широко используемые в приложениях компьютерного зрения), [рекуррентные нейронные сети](https://ru.wikipedia.org/wiki/Special:Search/%D1%80%D0%B5%D0%BA%D1%83%D1%80%D1%80%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B5%20%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5%20%D1%81%D0%B5%D1%82%D0%B8) (широко используемые для понимания естественного языка и последовательного обучения) и так далее.

* + 1. **С обратным распространением информации**

Характеризуются и прямым, и обратным распространением информации между слоями сети.

* + - 1. **Релаксационные**

В них циркуляция информации происходит до того момента, пока не перестанут изменяться входные значения, то есть пока не наступит состояние равновесия. В основе функционирования, таких сетей лежит итеративный принцип работы. Он заключается в том, что на каждой итерации процесса происходит обработка данных, полученных на предыдущем шаге. Такая циркуляция информации продолжается до тех пор, пока не установится состояние равновесия. При этом состояния нейронных элементов перестаю т изменяться и характеризуются стационарными значениями. Для анализа устойчивости релаксационных нейронных сетей используются функции Ляпунова. Такие сети применяются в качестве ассоциативной памяти и для решения комбинаторных задач оптимизации. К релаксационным сетям относится нейронная сеть Хопфилда, Хэмминга, двунаправленная ассоциативная память и машина Больцмана. На рисунках 10 и 11 приведены схемы сетей, относящихся к этому типу.

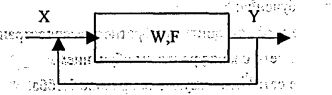


Рисунок 10 – Структура сети Хопфилда (характеризуется единичной обратной связью)

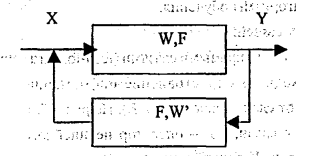


Рисунок 11 – Структура двунаправленной ассоциативной памяти (характеризуется неединичной обратной связью)

Такие НС применяются в качестве ассоциативной памяти и для решения комбинаторных задач оптимизации.

* + - 1. **Без релаксации**

Архитектура базируется на многослойном персептроне, а в основе обучения лежит алгоритм обратного распространения ошибки.

* + - * 1. **Рекуррентные сети**

В таком типе НС присутствует обратная связь между входом и выходом, как показано на Рисунке 12. Выходное значение определяется в зависимости как от входного, так и от предшествующих выходных значений. То есть в таких сетях выходы нейронных элементов последующих слоев имеют синаптические соединения с нейронами предшествующих слоев. Это приводит к возможности учета результатов преобразования нейронной сетью информации на предыдущем этапе для обработки входного вектора на следующем этапе функционирования сети. Рекуррентные сети (RNN) могут использоваться для решения задач прогнозирования и управления.

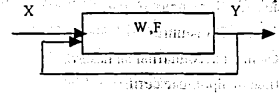


Рисунок 12 – Рекуррентная сеть

Существуют различные варианты архитектур рекуррентных нейронных сетей. В 1986 году Джордан предложил рекуррентную сеть, в которой выходы нейронных элементов последнего слоя соединены посредством специальных входных нейронов с нейронами промежуточного слоя. Такие входные нейронные элементы называются контекстными нейронами. Они распределяют выходные данные нейронной сети на нейронные элементы промежуточного слоя, как показано на Рисунке 13. Количество контекстных нейронов равняется числу выходных нейронных элементов рекуррентной сети. В качестве выходного слоя таких сетей используются нейронные элементы с линейной функцией активации.

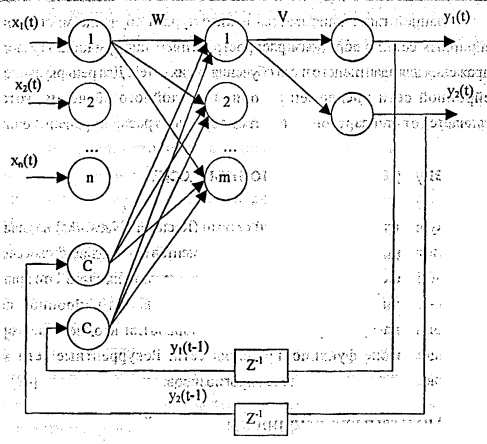


Рисунок 13 – Архитектура рекуррентной нейронной сети с обратными связями от нейронов выходного слоя, где z – элемент задержки, а С – контекстный нейрон

Выходное значение нейронного элемента последнего слоя определяется

где – весовой коэффициент между нейроном промежуточного и нейроном выходного слоя, пороговое значение нейрона выходного слоя, – выходное значение нейрона промежуточного слоя.

Взвешенная сумма нейронного элемента промежуточного слоя определяется следующим образом:

где - весовой коэффициент между нейроном входного и нейроном промежуточного слоя, - пороговое значение нейрона промежуточного слоя,; - размерность входного вектора, - количество нейронов выходного слоя; - весовой коэффициент между контекстным нейроном и нейроном промежуточного слоя.

Тогда выходное значение нейрона скрытого слоя

В качестве функции нелинейного преобразования обычно используется гаперболотеский тангенс или сигмоидная функция.

Другой вариант рекуррентной нейронной сети предложил Элман в 1990 году. В такой сети выходы нейронных элементов промежуточного слоя соединяются с контекстными нейронами входного слоя, как показано на Рисунке 14.

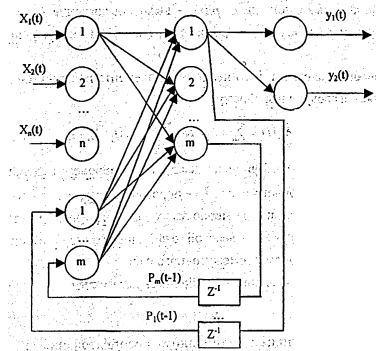


Рисунок 14 – Архитектура рекуррентной сети с обратными связями от нейронов скрытого слоя

Взвешенная сумма нейронного элемента промежуточного слоя определяется следующим образом:

где - количество нейронов промежуточного слоя, - выходное значение нейрона промежуточного слоя.

Тогда выходное значение нейрона скрытого слоя

Для построения рекуррентных нейронных сетей можно использовать также оба рассмотренных ранее подхода одновременно. В этом случае существуют обратные связи к контекстным нейронам как от нейронных элементов выходного, так и промежуточного слоя, как показано на рисунке 15.

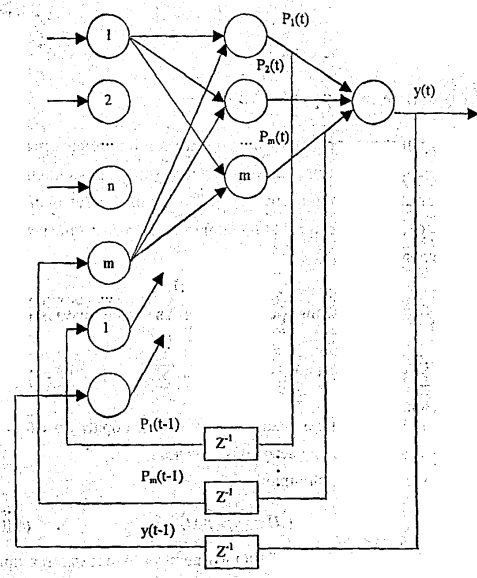


Рисунок 15 – Архитектура рекуррентной сети с обратными связями от нейронов скрытого и выходного слоев

Количество контекстных нейронов входного слоя равняется общему числу нейронов промежуточного и выходного слоев. Тогда

Алгоритм обучения, рекуррентной НС в общем случае состоит из следующих шагов:

1. В начальный момент времени все контекстные нейроны устанавливаются в нулевое состояние, т.е. их выходные значения равняются нулю.
2. Входной образ подается на сеть и происходит прямое распространение его в нейронной сети.
3. В соответствии с алгоритмом обратного распространения ошибки производится модификация весовых коэффициентов и пороговых значений нейронных элементов.
4. Устанавливается и осуществляется переход к пункту 2.

Обучение рекуррентной сети производится до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной. Процесс обучения также можно ускорить, если использовать определенные выражения для выходного слоя, а также используя различные функции активации.

Одна из привлекательных идей RNN состоит в том, что они потенциально умеют связывать предыдущую информацию с текущей задачей, так, например, знания о предыдущем кадре видео могут помочь в понимании текущего кадра. Если бы RNN обладали такой способностью, они были бы чрезвычайно полезны. Иногда для выполнения текущей задачи необходима только недавняя информация.

Рассмотрим, например, языковую модель, пытающуюся предсказать следующее слово на основании предыдущих. Если мы хотим предсказать последнее слово в предложении «облака плывут по *небу*», нам не нужен более широкий контекст. В этом случае довольно очевидно, что последним словом будет «небу». В этом случае, когда дистанция между актуальной информацией и местом, где она понадобилась, невелика, RNN могут обучиться использованию информации из прошлого. Но бывают случаи, когда нам необходимо больше контекста. Допустим, мы хотим предсказать последнее слово в тексте «Я вырос во Франции… Я бегло говорю *по-французски*». Ближайший контекст предполагает, что последним словом будет называние языка, но, чтобы установить, какого именно языка, нам нужен контекст Франции из более отдаленного прошлого.

То есть проблема с обычными RNN ячейками, описанными выше, заключается в том, что они не могут «удержать в памяти» слишком длинные последовательности. Связано это с тем, что когда мы прокидываем градиенты через, достаточно, длинную последовательность, то мы сталкиваемся с одной из двух проблем: либо градиенты уменьшаются настолько, что ошибки на конце последовательности перестают влиять на ее начало, либо градиенты увеличиваются, и процесс расходится. Такая же проблема существует и в обычных сетях: добавляя слои мы начинаем испытывать затруднения с тренировкой. Чтобы победить эту проблему было предложено заменить обычные RNN ячейки на более продвинутый вариант LSTM.

Долгая краткосрочная память (Long short-term memory; LSTM) – особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям. Они были представлены Зеппом Хохрайтер и Юргеном Шмидхубером в 1997 году, а затем усовершенствованы и популярно изложены в работах многих других исследователей. Они прекрасно решают целый ряд разнообразных задач и в настоящее время широко используются.

Любая рекуррентная нейронная сеть имеет форму цепочки повторяющихся модулей нейронной сети. В обычной RNN структура одного такого модуля очень проста, например, он может представлять собой один слой с функцией активации- гиперболический тангенс. Структура LSTM также напоминает цепочку, но модули выглядят иначе. Вместо одного слоя нейронной сети они содержат целых четыре, и эти слои взаимодействуют особенным образом. LSTM сети были призваны решить проблему выбора того, какую информацию пропускать на следующий слой, а какую оставить. Схематично структура LSTM приведена на Рисунке 16.

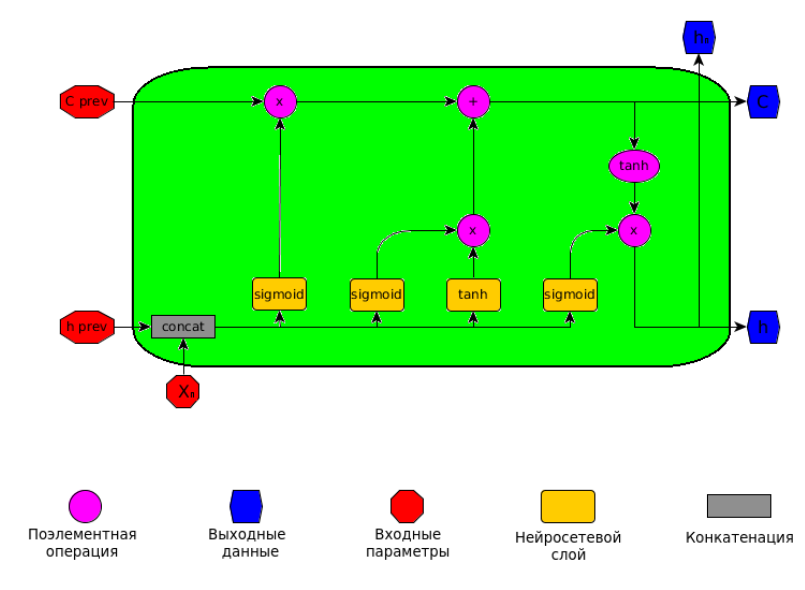


Рисунок 16 – Структуры LSTM

где каждый входной элемент — вектор 𝑥, содержащий признаковое описание объекта последовательности, ℎ𝑝𝑟𝑒𝑣 — результат, получившийся на выходе предыдущей ячейки, 𝐶𝑝𝑟𝑒𝑣 — это дополнительный выход предыдущей ячейки (он нужен, чтобы хранить информацию о предшествующих элементах последовательности. То есть его главное отличие от ℎ в том, что он несет информацию, главным образом, не о предыдущей ячейки, но о большом числе, идущих до нее), конкатенацию здесь следует понимать, как соединение двух векторов в один, размерность которого, соответственно, равна сумме размерностей исходных.

Одно из принципиальных отличий LSTM от остальных рекурентных сетей — это последовательность слоев, которая на диаграмме изображена прямой линией в верхней части, как показано на Рисунке 17. Через нее информация может спокойно проходить в следующую ячейку, при этом каждый новый элемент последовательности может воздействовать на нее по-разному, может поменять полностью, а может оставить без изменений.

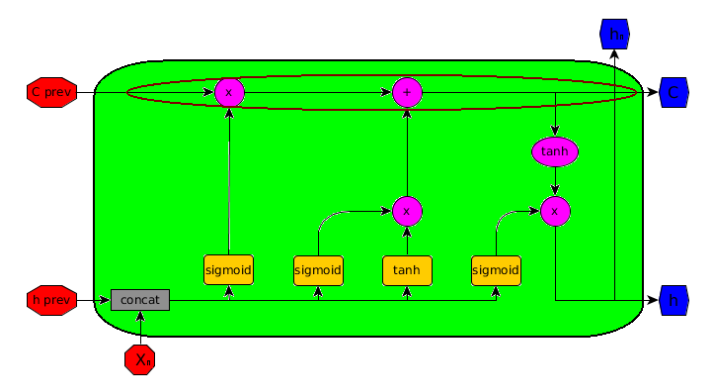


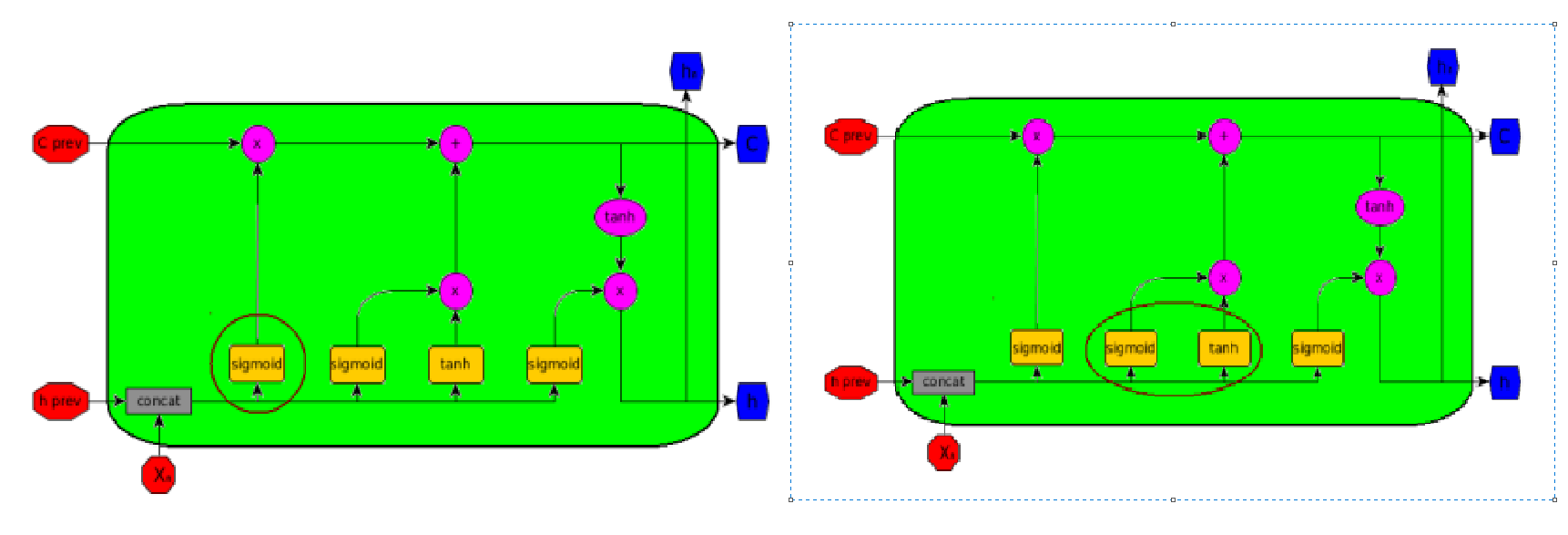
Рисунок 17 – Последовательность слоев LSTM

Обратимся теперь к самим нейросетевым слоям. Рассмотрим первую сигмоиду, выделенную в левом верхнем углу на Рисунке 18. Здесь сеть выбирает, какая информация о предшествующем элементе должна пройти через текущую ячейку. Для этого конкатенируются вектор выхода предыдущей ячейки ℎ𝑡−1 и новый элемент последовательности 𝑥𝑡 и в качестве одного вектора пропускаются через слой с сигмоидной функцией активации. Назовем вектор, полученный после конкатенации 𝑣𝑡.

Теперь рассмотрим следующие два слоя, выделенные в правом верхнем углу Рисунка 18. Они отвечают за то, какую новую информацию мы привнесем в модель. Сигмоиду в этой структуре обычно называют “input gates layer”. Она решает, какие значения вектора будут обновлены. Затем слой с гиперболическим тангенсом определяет новых “кандидатов” для 𝐶. Далее, эти вектора поэлементно перемножаются, чтобы потом прибавиться к новому значению 𝐶.

Далее происходит обновление значения новые вектора 𝐶, выделенного в левом нижем углу Рисунка 18. Ранее уже решено, какие значения будут записаны в 𝐶, остается лишь поэлементно домножить старое значение на то, что получено на предыдущем шаге и прибавить новые значения.

Осталось на основе информации, которая хранится в 𝐶, а также 𝑣𝑡 получить выходное значение ячейки, как показано в правом нижнем углу рисунка 18. Для этого пропустим 𝑣𝑡 через слой с сигмоидной функцией активации, чтобы решить, что из 𝐶 стоит выдавать в качестве выходного значения. Потом применим слой с тангенциальной функцией активации к 𝐶 и умножим на выход сигмоидного слоя, чтобы получить только нужные значения. Полученный результат и будет выходным значением.



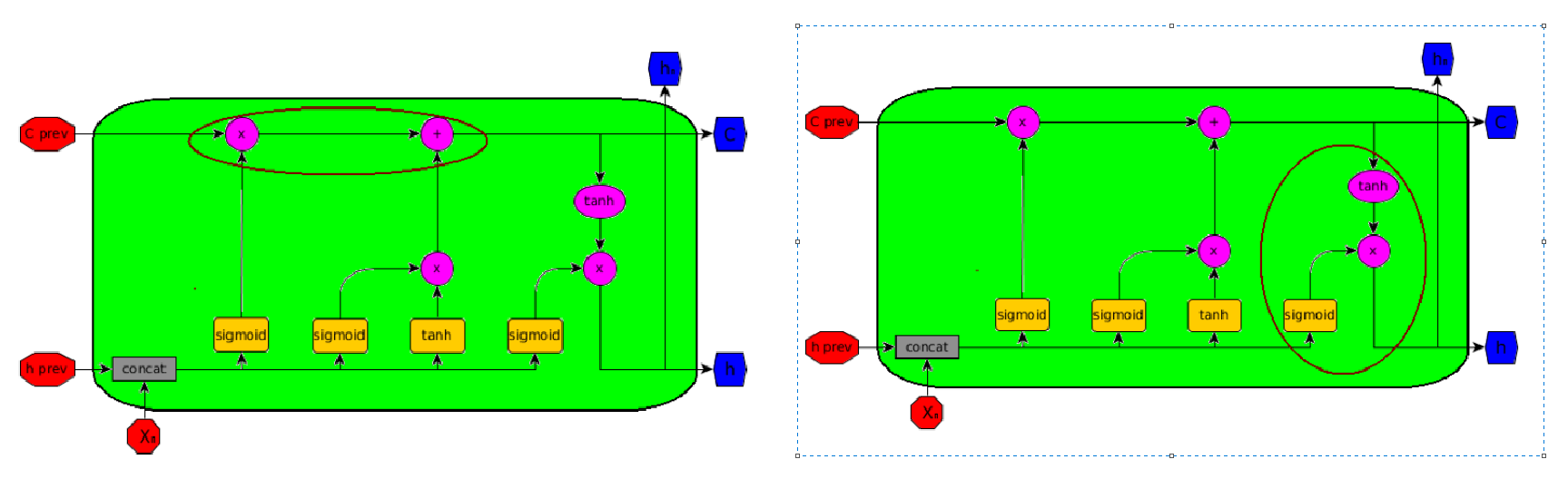


Рисунок 18 – Детальный разор схемы

То есть информация может быть отменена помощи так называемых фильтров. Они позволяют пропускать информацию на основании некоторых условий. Они состоят из слоя сигмоидальной нейронной сети и операции поточечного умножения. Сигмоидальный слой возвращает числа от нуля до единицы, которые обозначают, какую долю каждого блока информации следует пропустить дальше по сети. Ноль в данном случае означает «не пропускать ничего», единица – «пропустить все». В LSTM три таких фильтра, позволяющих защищать и контролировать состояние ячейки.

Существуют различные разновидности LSTM: LSTM with forget gate (которые позволяли дополнительно занулять некоторые компоненты внутреннего состояния прежде, чем передавать их дальше), LSTM with peephole conection (которое позволяет учесть текущее состояние в вычислении всевозможных ворот) [7].

**2.5.2.2.2 Рециркуляционные сети**

Характеризуются как прямым , так и обратным , преобразованием информации. Задачей такого преобразования является достижение наилучшего автопрогноза или самовоспроизводимости вектора . Рециркуляционные нейронные сети применяются для сжатия (прямое преобразование), определяется выражением: и восстановления исходной (обратное преобразование) информации, определяется выражением: . В качестве функции активации нейронных элементов может использоваться как линейная, так и нелинейная функции.

Такие сети являются самоорганизующимися в процессе работы. Обучение производится без учителя (являются самоорганизующимися в процессе работы). Схема рециркуляционных сетей приведена на рисунке 19.

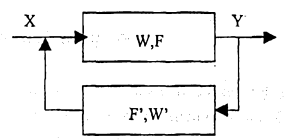
**

Рисунок 19 – Рециркуляционная сеть

Теоретической основой рециркуляционных нейронных сетей является анализ главных компонент.

Метод главных компонент применяется в статистике для сжатия информации без существенных потерь ее информативности. Он состоит в линейном ортогональном преобразовании входного вектора размерности n в выходной вектор размерности , где . При этом компоненты вектора являются некоррелированными и общая дисперсия после преобразования остается неизменной. Совокупность входных паттернов представим в виде матрицы:

где соответствует входному образу, .

Будем считать, что матрица является центрированной, то есть вектор математических ожиданий =0. Этого добиваются при помощи следующих преобразований:

Матрица ковариаций входных данных определяется как

где ковариация между и ой компонентой входных образов.

Элементы матрицы ковариаций можно вычислить следующим образом:

Метод главных компонент состоит в нахождении таких линейных комбинаций исходных переменных

что

Из последних выражений следует, что переменные некоррелированы, упорядочены по возрастанию дисперсии и сумма дисперсий входных образов остается без изменений. Тогда подмножество первых переменных характеризует большую часть общей дисперсии. В результате получается представление входной информации. Переменные , называются главными компонентами. В матричной форме преобразование главных компонент можно представить как

где строки матрицы должны удовлетворять условию ортогональности, то есть

при этом вектор

Также необходимо определить весовые коэффициенты .

При использовании линейной функции активации:

Yаилучший автопрогноз достигается тогда, когда матрица весовых коэффициентов сформирована в соответствии с методом главных компонент. При этом столбцы матрицы равняются собственным векторам ковариационной матрицы. Тогда:

Таким образом весовые коэффициенты линейной рециркуляционной нейронной сети можно определить при помощи метода главных компонент. В этом случае матрица является ортогональной и

Линейные рециркуляционные сети, в которых весовые коэффициенты определяются в соответствии с методом главных компонент называются РСА сетями.

Каждый из слоев нейронных элементов может использоваться в качестве входного или выходного. Если слой нейронных элементов служит в качестве входного, то он выполняет распределительные функции. В противном случае нейронные элементы слоя являются обрабатывающими. Весовые коэффициенты, соответствующие прямым и обратным связям, характеризуются матрицей весовых коэффициентов и . Архитектура рециркуляционной НС предоставлена на Рисунке 20.

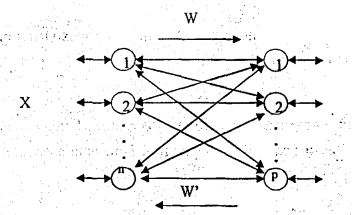


Рисунок 20 – Архитектура рециркуляционной НС

Для наглядности, рециркуляционную сеть можно представить в развернутом виде, как показано на Рисунке 21. Такое представление сети является эквивалентным и характеризует полный цикл преобразования информации. При этом промежуточный слой нейронных элементов производит кодирование (сжатие) входных данных , а последний слой осуществляет восстановление сжатой информации [6].

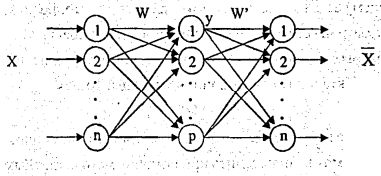


Рисунок 21 – Эквивалентное представление рециркуляционной сети

Недостатком таких сетей является сложность анализа временных последовательностей, например, звукового сигнала, изменение положения объекта во время движения, текстовая информация и многое другое.

* + 1. **Сверточные нейронные сети**

Сверточные нейронные сети (CNN) являются дальнейшим развитием многослойного персептрона и широко используются для обработки изображений. В отличие от многослойного персептрона, сверточные нейронные сети позволяют учитывать топологию изображений и инвариантны к сдвигам, масштабированию и другим искажениям входного образа.

Сверточная нейронная сеть объединяет три подхода при обработке изображений. Это использование локального рецептивного поля для каждого нейрона сверточного слоя, формирование сверточных слоев в виде набора карт, нейронные элементы которых имеют одинаковые синаптические связи, и наличие карт подвыборочного слоя, повышающего инвариантность сети к искажениям. Применение локального рецептивного поля позволяет нейронам одной карты признаков детектировать один и тот же стимул в разных фрагментах изображения. Использование идентичных нейронов в каждой карте дает возможность сократить количество настраиваемых синаптических связей сети. Подвыборочный слой осуществляет локальное усреднение или операцию максимума для каждого неперекрывающего фрагмента карты сверточного слоя, что позволяет уменьшить размерность соответствующих карт признаков. Общая архитектура сверточной сети показана на Рисунке 22.

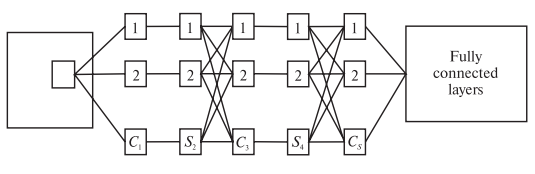


Рисунок 22 – Архитектура сверточной нейронной сети

где C1, C2, C3 – сверточные слои, и подвыборочные слои S2, S4.

Как уже отмечалось, сверточный слой состоит из множества карт признаков, где нейроны каждой карты содержат одни и те же наборы весов и порогов. В результате нейронные элементы каждой карты признаков выполняют одни и те же операции над различными частями изображения. Для сканирования изображения используется метод скользящего окна, которое называется локальным рецептивным полем для соответствующего ему нейрона карты признаков. Поэтому если размер скользящего окна, которое называется ядром, равняется (рецептивное поле), то каждый нейрон сверточного слоя связан с p2 элементами соответствующей области рецептивного поля изображения. Каждое рецептивное поле (окно) во входном пространстве образов отображается на специальный (отдельный) нейрон в каждой карте признаков. Если скользящее окно сканирует изображение с единичным шагом, то количество нейронов в каждой карте признаков определяется следующим образом:

где n\*n – размерность исходного изображения.

Если скользящее окно сканирует изображение с шагом s, то количество нейронов в каждой карте признаков в общем случае вычисляется по формуле

Общее количество различных синаптических связей в сверточном слое равно

где M - общее количество карт признаков в сверточном слое.

Как следует из последнего выражения, применение сверточной сети позволяет сократить общее количество настраиваемых синаптических связей по сравнению с многослойным персептроном за счет использования идентичных нейронов в каждой карте признаков.

Как уже отмечалось, нейроны каждой карты признаков имеют одинаковый набор весов и порогов. В результате из одного и того же изображения можно извлечь множество различных признаков. Далее в целях уменьшения размерности карт признаков эти признаки обрабатываются следующим слоем S2, который называется объединяющим или подвыборочным. Он осуществляет сжатие карт признаков сверточного слоя при помощи операции максимизации или локального усреднения различных областей карт признаков. Для этого каждая карта признаков сверточного слоя разбивается на неперекрывающиеся области размером k\*k. Каждая область отображается в один нейрон соответствующей карты подвыборочного слоя. Каждый нейрон подвыборочного слоя вычисляет среднее или максимальное значение k2 нейронов в сверточном слое:

Количество нейронов в каждой карте подвыборочного слоя

Число карт признаков в слое пулинга будет таким же, как и в сверточном слое. Далее опять следуют сверточный (), пулинговый ( и сверточный (слои. При этом каждый нейронный элемент пулингового слоя в общем случае имеет синаптические связи со всеми нейронами сверточного слоя, как показано на Рисунке 22. Таким образом, сверточная нейронная сеть представляет собой сочетание сверточных и пулинговых слоев, которые выполняют нелинейное иерархическое преобразование входного пространства образов, как показано на Рисунке 23. Последний блок сверточной нейронной сети является многослойным персептроном, машиной опорных векторов или другим классификатором.

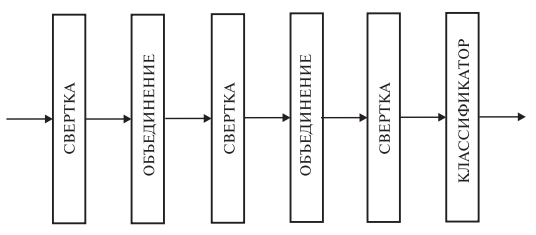


Рисунок 23 – Общая структура сверточной НС

Рассмотрим также традиционную сверточную нейронную сеть (LeNet-5) для классификации рукописных цифр, приведенную на Рисунке 24.

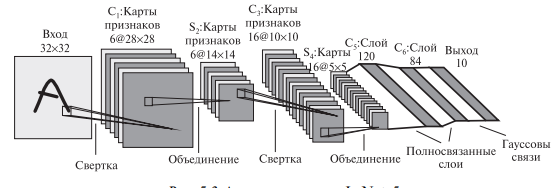


Рисунок 24 – Архитектура сети LeNet-5

Входное изображение имеет размер 32\*32 пикселя. Скользящее окно размерностью 5\*5 сканирует изображение с шагом 1, и каждый фрагмент изображения поступает на соответствующий нейронный элемент карты признаков сверточного слоя C1 нейронной сети. Слой C1 состоит из шести карт признаков, где каждая карта согласно выражению ), содержит 28\*28 нейронов. Слой S2 представляет собой подвыборочный слой с шестью картами признаков. Каждый нейрон подвыборочного слоя вычисляет среднее значение четырех нейронов в сверточном слое, т.е. . Тогда, согласно *,*  каждая карта подвыборочного слоя S2 содержит 14\*14 нейронных элементов. Слой C3 является сверточным слоем с 16 картами признаков и ядром сканирования 5\*5 для каждой карты подвыборочного слоя. Тогда размерность каждой карты признаков сверточного слоя C3 равняется 10\*10 нейронных элементов. В LeNet-5 слои S2 и C3 образуют не полностью связанную нейронную сеть, где нейроны этих слоев имеют синаптические связи между собой. Слой S4 является подвыборочным слоем с 16 картами признаков и ядром 2\*2 для каждой карты признаков сверточного слоя. Поэтому каждый нейрон подвыборочного слоя S4 вычисляет среднее значение четырех нейронов в сверточном слое C3, т.е. . Поэтому размерность карты признаков данного слоя составляет 5\*5 нейронных элементов. Следующий сверточный слой С5 сканирует подвыборочный слой окном размерностью 5\*5. При этом каждый нейрон подвыборочного слоя имеет синаптические связи со всеми нейронами сверточного слоя. Слой F6 содержит 84 нейронных элемента с функцией активации гиперболического тангенса и функционирует как классический персептронный слой. Выходной слой состоит из 10 нейронных элементов, каждый из которых формирует выходное значение в соответствии с радиально-базисной функцией активации.

Согласно выражению

чем больше отличается входной вектор от весового вектора соответствующего выходного нейронного элемента, тем меньше будет выходное значение этого нейрона.

Во многих практических приложениях важной проблемой являются ограниченные вычислительные ресурсы, которые не позволяют использовать сложные нейронные сети. Поэтому снижение сложности нейронных сетей при сопоставимой точности решения задачи – достаточно важная проблема [8]. Упрощенная архитектура по сравнению с LeNet-5 приведена на рисунке 25.

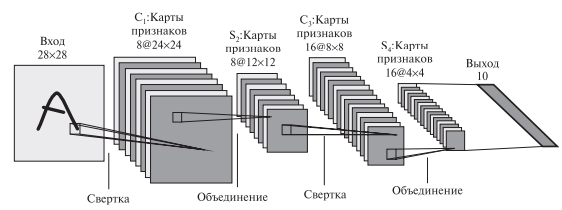
**

Рисунок 25 - Архитектура редуцированной сверточной нейронной сети

Использование ядер свертки дает меньшее количество настраиваемых весовых коэффициентов, что приводит к уменьшению времени и вычислительных ресурсов на обучение сети, а также помогает избежать попиксельного запоминания, подталкивая сеть к обобщению демонстрируемой информации. Локальное восприятие позволяет сохранить топологию изображения от слоя к слою при значительном сокращении вычислений. Благодаря сканированию целой области, а не отдельных точек, подход позволяет учесть свойства изображения, что увеличивает качество распознавания. Также НС такого типа имеет работоспособность к алгоритмам поворота и сдвига, инвариантность к масштабированию за счет субдискретизации. Обучается при помощи метода обратного распространения ошибки, при этом могут быть задействованы и другие методы обучения сети (в том числе и методы «без учителя», например, patch-based training, когда следующий слой свёртки обучается на кусочках уже обученного первого слоя).

Минус же в том, что такая НС имеет продолжительное время обучения для нейронной сети с числом слоёв свёртки свыше двух. Классификатор рукописных символов обучался более 64 часов. Необходимость в большом количестве примеров для обучения. Реализация классификатора рукописных символов использовала базу MNIST, которая содержит более 60 тысяч учебных образов с цифрами от 0 до 9 различного начертания. А также имеет слишком много варьируемых параметров сети: количество слоёв, размер ядра свёртки для каждого из слоёв, количество ядер для каждого из слоёв, шаг сдвига, необходимость использования слоёв подвыборки, степень уменьшения ими размерности, функция по уменьшению размерности и т.д.

* + 1. **Трансформеры НС**

Трансформер – архитектура глубоких нейронных сетей, основанная на [механизме внимания](https://neerc.ifmo.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC_%D0%B2%D0%BD%D0%B8%D0%BC%D0%B0%D0%BD%D0%B8%D1%8F) без использования [рекуррентных нейронных сетей](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%80%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B5_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8) . Впервые модель трансформера была предложена в статье Attention is All You Need от разработчиков Google в 2017 году. У RNN есть несколько известных минусов:

* Во-первых, RNN содержат всю информацию о последовательности в скрытом состоянии, которое обновляется с каждым шагом. Если модели необходимо «вспомнить» что-то, что было сотни шагов назад, то эту информацию необходимо хранить внутри скрытого состояния и не заменять чем-то новым. Следовательно, придется иметь либо очень большое скрытое состояние, либо мириться с потерей информации.
* Во-вторых, обучение рекуррентных сетей сложно распараллелить: чтобы получить скрытое состояние RNN-слоя для шага i + 1 необходимо вычислить состояние для шага *i*. В это занимает много времени и не очень эффективно работает на GPU, созданных для параллельных вычислений.

Обе этих проблемы затрудняют применение RNN к по-настоящему длинным последовательностям: даже если дождаться конца обучения, модель по своей конструкции будет так или иначе терять информацию о том, что было в начале текста. Хочется иметь способ «читать» последовательность так, чтобы в каждый момент времени можно было обратиться к произвольному моменту из прошлого за константное время и без потерь информации. Таким способом и является лежащий в основе трансформеров механизм self-attention.

Устройство трансформера состоит из кодирующего и декодирующего компонентов. На вход принимается некая последовательность, создается ее векторное представление прибавляется вектор позиционного кодирования, после чего набор элементов без учета порядка в последовательности поступает в кодирующий компонент (параллельная обработка), а затем декодирующий компонент получает на вход часть этой последовательности и выход кодирующего. В результате получается новая выходная последовательность. Внутри кодирующего и декодирующего компонента нет рекуррентности. Кодирующий компонент состоит из кодировщиков, которые повторяются несколько раз, аналогично устроен декодирующий компонент. Трансформер — это поставленные друг за другом модели внимания, которые позволяют исходную последовательность векторов перевести в новую последовательность векторов, которые кодируют информацию о контексте каждого элемента. Трансформер-кодировщик переводит исходные векторы в скрытые, которые правильно сохраняют в себе информацию о контексте каждого элемента. Далее трансформер-декодировщик декодирует результат кодировщика в новую последовательность, которая состоит из элементов выходного языка. После по ним генерируются сами итоговые элементы с помощью вероятностной языковой модели. На Рисунке 26 приведена архитектура НС трансформера.

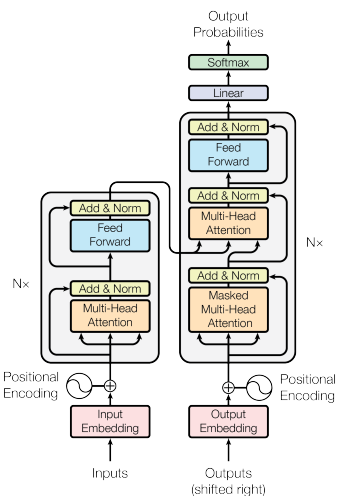


Рисунок 26 – Архитектура трансформера

Слева на схеме представлено устройство энкодера. Он по очереди применяет к исходной последовательности *N* блоков. Каждый блок выдаёт последовательность такой же длины. В нём есть два важных слоя, multi-head attention и feed-forward. После каждого из них к выходу прибавляется вход и затем активации проходят через «Add & Norm» (layer normalization).

У декодера схема похожая, но внутри каждого из блоков два слоя multi-head attention, в одном из которых используются выходы энкодера.

Первая часть transformer-блока — это слой self-attention – разновидность механизма внимания, задачей которой является выявление закономерности между входными данными. От обычного внимания его отличает то, что выходом являются новые представления для элементов той же последовательности, что мы подали на вход, причем каждый элемент этой последовательности напрямую взаимодействует с каждым.

Для каждого элемента xi получается обучаемым преобразованием три вектора:

* *qi*​ — запрос к базе данных;
* *ki*​ — ключи хранящихся в базе значений, по которым будет осуществляться поиск;
* *vi*​ — сами значения.

Первые два используются для того, чтобы посчитать важность элемента xj для элемента xi. Чтобы понять, насколько для пересчета вектора элемента xi. важен элемент xj берется вектор ключа второго элемента и умгжается на вектор запроса первого элемента. Так скалярно перемножаются вектор запроса на все векторы ключей, тем самым давая понять, насколько каждый входной элемент нужен, чтобы посчитать вектор первого элемента.

BERT (Bidirectional Encoder Representations from Transformers) и GPT (Generative Pre-trained Transformer) – это две разные модели для обработки естественного языка.

GPT эффективен в задачах, связанных с генерацией текста. Он генерирует в ответ на текстовый или устный запрос пользователя связные и информативные тексты, адаптируясь к стилю и формату. Эта технология используется в диалоговых приложениях: чат-ботах и виртуальных помощниках.  Модель GPT представляет собой обычную языковую модель, реализованную в виде последовательности слоев декодера трансформера. В качестве задачи при обучении выступает обычное предсказание следующего токена (то есть многоклассовая классификация по словарю).

BERT лучше подходит для задач, требующих понимания контекста и нюансов языка, например, при классификации предложений или поиске пар похожих документов. Он воспринимает предложение или текст целиком и лучше справляется с неоднозначностями в трактовке. BERT эффективен в анализе и классификации текста, а также в системах машинного перевода.  Отличается от GPT двунаправленностью внимания: это значит, что при обработке входной последовательности все токены могут использовать информацию друг о друге. Причем он не обучается для генерации с нуля, а использует предсказание случайно замаскированных слов по оставшимся и предсказание по паре текстовых фрагментов.

Выбор между BERT и GPT зависит от конкретных требований приложения.

Трансформеры – относительно новый тип нейросетей, направленный на решение последовательностей с легкой обработкой дальнодействующих зависимостей. На сегодня это самая продвинутая техника в области обработки естественной речи (NLP).

С их помощью можно переводить текст, писать стихи и статьи и даже генерировать компьютерный код. В отличие от рекуррентных нейронных сетей (RNN), трансформеры не обрабатывают последовательности по порядку.

**ЗАКЛЮЧЕНИЕ**

Современные классы нейронных сетей представляют собой мощный инструмент, который активно используется для решения широкого спектра задач в различных областях. Одни НС демонстрируют высокую эффективность в обработке изображений, другие – в анализе последовательных данных, а третьи – в задачах обработки естественного языка и генерации контента.

Несмотря на успехи, каждое из рассмотренных направлений имеет свои ограничения, которые стимулируют дальнейшие исследования. Например, необходимость значительных вычислительных ресурсов, сложность обучения глубоких архитектур и ограниченная интерпретируемость остаются актуальными вызовами для разработчиков и исследователей.

В технической литературе представлено множество различных классификаций нейронных сетей по определенным признакам. К сожалению, в рамках реферата проблематично описать максимально подробно каждый существующий класс, так как тема является достаточно объемной. Но мной были рассмотрены основные, наиболее часто встречающиеся и интересные, на мой взгляд, типы.

Проведенный обзор подчеркивает важность выбора правильной архитектуры нейронной сети для конкретной задачи, а также необходимость учета специфики данных и требований к модели. В будущем можно ожидать дальнейшего развития гибридных подходов, объединяющих сильные стороны различных классов сетей, а также появления более эффективных и интерпретируемых моделей, способных работать в условиях ограниченных ресурсов.

Таким образом, современные нейронные сети продолжают оставаться ключевым элементом в развитии искусственного интеллекта, открывая новые горизонты для научных исследований и практических приложений.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Хайкин, С. Нейронные сети: полный курс / С. Хайкин; пер. с англ. – М.: ООО «И.Д. Вильямс», 2016. – 1104 с.
2. Николаева, С.Г. Нейронные сети: учебное пособие / С.Г. Николаева. – Казань.: Казан. гос. энерг. ун-т, 2015. – 92 с.
3. Грессер, Л. Глубокое обучение с подкреплением: теория и практика на языке Python / Л. Грессер, В.Л. Кенг. – Спб.: Питер, 2022. – 416 с.
4. Бахтин, А.В. Элементы искусственного интеллекта в системах управления: учебное пособие / А.В. Бахтин, И.В. Ремизова. – СПб.:СПбГТУРП, 2014. – 55 с.
5. Круглов, В.В. Искуственные нейронные сети: теория и практика / В.В. Круглов, В.В. Борисов. – М.: «Горячая линия – Телеком», 2002. – 382 с.
6. Головко, В.А. Нейроинтелект: теория и примечание. Книга 1: организация и обучение НС с прямыми и обратными связями / В.А. Головко. – Брест.: БПИ, 1999. – 264 с.
7. Johnson, R. Supervised and Semi-Supervised Text Categorization using LSTM for Region Embeddings / R. Johnson, T. Zhang. - Journal of Machine Learning Research, 2016. – 18 c.
8. Головко, В.А. Нейросетевые технологии обработки данных / В.А. Головко, В.В. Краснопрошин. – Минск.: БГУ, 2017. – 263 с.