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

студента 5 курса 531 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Яшина Максима Алексеевича

Старший преподаватель
д. ф.-м. н., доцент _____ И. И. Слеповичев
подпись, дата

Саратов 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Возможности современных компьютеров позволяют производить различные вычисления со скоростью на десятки порядков превышающей возможности человеческого мозга. Однако ряд даже тривиальных для человека задач, не связанных с вычислениями, остается весьма сложным для вычислительной техники. Способность человека к ассоциативному хранению информации, обучению, обобщению и обработки информации с учетом контекста остается непревзойденной даже для современных суперкомпьютеров. Целью проектирования искусственных нейронных сетей (ИНС) является построение вычислительной структуры или алгоритма, работающего по принципам естественного интеллекта. К таким можно отнести следующие свойства нейронных сетей [1].

1. Нейронные сети, по аналогии с мозгом человека и животных, строятся из множества простых элементов, выполняющих элементарные действия и соединенных между собой различными связями.
2. Нейронные сети способны совершенствовать свою работу (обучаться или адаптироваться), используя примеры.
3. Нейросетевое решение задачи не требует от разработчика формулировки алгоритма решения поставленной задачи и его программирования. Нейронные сети, как правило, используют примеры «правильной» работы для построения своего метода решения задачи. При этом существует возможность выявления сетью скрытых закономерностей в задаче, неизвестных разработчику.

1 Основные принципы работы нейронных сетей

1.1 Что такое нейронная сеть: биологические и математические аналоги

Нейронные сети являются попыткой смоделировать работу биологических нервных систем. Основная идея состоит в использовании искусственных «нейронов» как элементарных вычислительных узлов. Биологический нейрон получает сигналы от других нейронов, суммирует их, и, если сумма превышает определённый порог, генерирует выходной сигнал. Так модель искусственного нейрона, предложенная Уорреном МакКаллоком и Уолтером Питтсом в 1943 году, была основана на принципе работы биологического нейрона. Искусственный нейрон МакКаллока-Питтса имеет N входных бинарных величин $x_1, \ldots, x_n$, которые трактуются как импульсы, поступающие на вход нейрону (Рисунок 1). В нейроне импульсы складываются с весами $\omega_1, \ldots, \omega_n$.

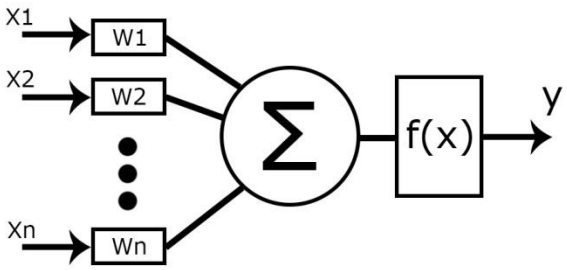

Рисунок 1 – Модель искусственного нейрона МакКаллока-Питтса

Выходной сигнал нейрона определяется по формуле:

$$y = f(\sum_{i=1}^{N} \omega_i x_i)$$

где f — нелинейная функция (функция активации) преобразует суммарный импульс в выходное значение нейрона. В модели МакКаллока-Питтса для этой цели использовалась функция Хевисайда. В дальнейшем было предложено использовать другие типы функций активации: логистическую сигмоидальную

($f(x) = \frac{1}{1+e^{(-x)}}$), гиперболической тангенс ($\tanh(x) = \frac{2}{1+e^{(-2x)}} - 1$) и радиально-базисную функцию. Такие функции активации обеспечивают более плавное изменение выходного сигнала нейрона [2].

1.2 Архитектура нейронных сетей: входной, скрытые и выходной слои

МакКаллок и Питтс предложили также метод объединения отдельных нейронов в искусственные нейронные сети. Для этого выходные сигналы нейрона передаются на вход следующему нейрону (Рисунок 2). Нейронная сеть состоит из нескольких слоев, на каждом из которых может находиться несколько нейронов. Из этих слоев и образуется архитектура нейронных сетей [2]. Рассмотрим каждый из них:

- Входной слой принимает данные в их исходной форме (например, пиксели изображения или числовые признаки).
- Скрытые слои выполняют преобразования данных, применяя линейные операции и функции активации. Их основная цель — выделить скрытые зависимости и признаки.
- Выходной слой производит результат, который может быть числовым значением (в задачах регрессии) или вероятностями классов (в задачах классификации).

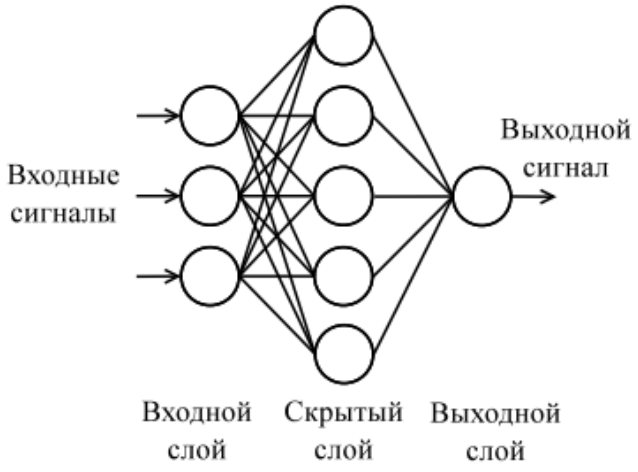

Рисунок 2 – Искусственная нейронная сеть

1.3 Обучение нейронных сетей

Обучение нейронной сети – это процесс определения весов соединений между нейронами таким образом, чтобы сеть приближала необходимую функцию с заданной точностью. Существует три подхода к обучению нейронных сетей: обучение с учителем (supervised learning), обучение без учителя (unsupervised learning) и обучение с подкреплением (reinforcement learning). При обучении с учителем на вход сети подаются наборы входных сигналов (объектов), для которых заранее известен правильный ответ (обучающее множество). Веса меняются по определенным правилам в зависимости от того, правильный ли выходной сигнал выдала сеть. При обучении без учителя на вход сети подаются объекты, для которых правильный выходной сигнал заранее не известен. Обучение с подкреплением предполагает наличие внешней среды, с которой взаимодействует сеть. Обучение происходит на основании сигналов, полученных от этой среды [2].

2 Классификация современных нейронных сетей

2.1 Полносвязные нейронные сети (Fully Connected Neural Networks, FNN)

2.1.1 Архитектура и особенности

Нейронная сеть, показанная на рисунке 2, называется полносвязной (Fully Connected Neural Networks, FNN) или многослойной перцептроном (Multilayer Perceptron, MLP). В такой сети каждый нейрон следующего слоя связан со всеми нейронами предыдущего слоя. Архитектура данной сети состоит из слоев, которые были рассмотрены выше, а именно: входного, скрытых и выходного слоя. Каждый слой можно описать уравнением:

$$h^{(l)} = f(W^{(l)}h^{(l-1)} + b^{(l)})$$

где $h^{(l)}$– выход на слое l, $W^{(l)}$ – вес, f – функция активации, а $b^{(l)}$ – смещения. FNN обучается с помощью алгоритма обратного распространения ошибки (backpropagation) и оптимизации весов методом градиентного спуска.

Ключевые особенности полносвязных сетей:

1. Универсальность: FNN могут аппроксимировать любые функции, если у них достаточно нейронов и скрытых слоёв.
2. Нелинейность: Использование функций активации, таких как ReLU или сигмоида, позволяет моделировать сложные зависимости.
3. Чувствительность к данным: FNN требуют числовых данных и часто плохо работают на задачах с временными рядами или структурированными данными без предварительной обработки [2].

2.1.2 Преимущества и недостатки

Преимущества:

1. Простота реализации: FNN легко реализовать и обучать, благодаря стандартным библиотекам (Keras, TensorFlow, PyTorch).
2. Гибкость: Они могут применяться к различным типам задач, включая классификацию, регрессию и прогнозирование.

Недостатки полносвязных сетей — большое количество параметров и подверженность переобучению. Для борьбы с этим используют регуляризацию, нормализацию, дропаут.

Полносвязные нейронные сети лежат в основе многих современных архитектур глубокого обучения. Их обычно комбинируют со сверточными и рекуррентными сетями в зависимости от типа данных и задачи [3].

2.1.3 Основные области применения

Полносвязные сети широко используются для классификации, где цель состоит в отнесении входных данных к одному из предопределённых классов. Примеры:

- Прогнозирование риска дефолта по финансовым данным.

- Классификация изображений, если данные предварительно приведены к векторной форме.

Также данный тип нейронной сети применяются для задач регрессии, таких как прогнозирование продаж, температуры или других временных рядов. Однако для более сложных временных зависимостей лучше подходят рекуррентные сети или трансформеры.

2.2 Сверточные нейронные сети (Convolutional Neural Networks, CNN)

2.2.1 Обучение, основы архитектуры: свертки, пуллинг

Сверточная нейронная сеть – это специальный вид нейронной сети для обработки данных с сеточной топологией. Примерами могут служить временные ряды, которые можно рассматривать как одномерную сетку примеров, выбираемых через регулярные промежутки времени, а также изображения, рассматриваемые как двумерная сетка пикселей. Сверточные сети добились колоссального успеха в практических приложениях. Своим названием они обязаны использованию математической операции свертки. Свертка – это особый вид линейной операции. Сверточные сети – это просто нейронные сети, в которых вместо общей операции умножения на матрицу, по крайней мере в одном слое, используется свертка.

В самом общем виде свертка – это операция над двумя функциями вещественного аргумента. Чтобы обосновать определение свертки, начнем с примеров возможных функций.

Допустим, что мы следим за положением космического корабля с помощью лазерного датчика. Наш датчик выдает единственное значение $x(t)$, положение корабля в момент t. Переменные x и t принимают вещественные значения, т.е. показания датчиков в любые два момента времени могут различаться.

Теперь предположим, что датчик подвержен помехам. Чтобы получить менее зашумленную оценку положения корабля, необходимо усерднить несколько результатов измерений. Разумеется, недавние измерения более важны, поэтому мы хотим вычислять взвешенное среднее, придавая недавним измерениям больший вес. Для этого можно воспользоваться весовой функцией $\omega(a)$, где a – давность измерения. Применив такую операцию усреднения в каждый момент времени, мы получим новую функцию, которая дает сглаженную оценку положения космического корабля:

$$s(t) = \int x(a)\omega(t-a)da$$

В терминологии сверточных сетей первый аргумент (в нашем примере функция x) называется входом, а второй (функция ω) – ядром. Выход иногда называют картой признаков.

Типичный слой сверточной сети состоит из трех стадий (Рисунок 3). На первой стадии слой параллельно выполняет несколько сверток и порождает множество линейных активаций. На второй стадии каждая линейная активация пропускается через нелинейную функцию активации, например функцию линейной ректификации. Эту стадию часто называют детекторной. На третьей стадии используется функция пулинга для дальнейшей модификации выхода слоя.

Функция пулинга заменяет выход сети в некоторой точке сводной статистикой близлежащих выходов. Например, операция max-пулинга возвращает максимальный выход в прямоугольной окрестности. Из других употребительных функций пулинга отметим усреднение по прямоугольной окрестности, L^2-норму в прямоугольной окрестности и взвешенное среднее с весами, зависящими от расстояния до центрального пикселя.

Рисунок 3 – Компоненты типичного слоя сверточной нейронной сети

Обучение сверточной сети является обучение признаков. Выходной слой, как правило, обходится относительно недорого, потому что ему на вход подается небольшое число признаков, прошедших несколько слоев пулинга. Если производится обучение с учителем методом градиетного спуска, то на каждом шаге необходимо выполнить полный цикл прямого и обратного распространений по всей сети. Один из способов уменьшить стоимость обучения сверточной сети – использовать признаки, для которых не применялось обучение с учителем.

Есть три основные стратегии получения сверточных ядер без обучения с учителем. Первый – просто инициализировать случайным образом. Второй – спроектировать вручную, настроив каждое ядро на обнаружение границ определенной ориентации или в определенном масштабе. Обучение признаков без учителя позволяет определять их отдельно от слоя классификации, занимающего верхнее место в архитектуре. Следовательно, можно выделить признаки для всего обучающего набора только один раз, по существу построив

новый обучающий набор для последнего слоя. Тогда обучение последнего слоя часто оказывается задачей выпуклой оптимизации в предположении, что этот слой реализует логистическую регрессию, методом опорных векторов или что-то подобное [4].

2.2.2 Преимущества CNN в обработке изображений

Основные особенности CNN в обработке изображений:

- Наличие сверточных слоев, выполняющих операцию свертки с изображением с помощью множества фильтров. Позволяет эффективно извлекать признаки.
- Наличие слоев подвыборки (pooling), уменьшающих размерность данных путем усреднения/максимилизации. Повышает устойчивость к искажениям.
- Разреженные (sparse) связи между нейронами. Каждый нейрон соединен только с небольшим локальным участком предыдущего слоя.
- Иерархическая организация: изображение обрабатывается последовательно от простых признаков к сложным.

Благодаря этому CNN хорошо масштабируются для изображений большого размера и устойчивы к сдвигам, поворотам и другим искажениям.

CNN широко применяются в задачах компьютерного зрения: классификации изображений, обнаружении объектов, семантической сегментации, распознавании лиц и др. Первые успехи CNN для изображений наблюдались в 2012 году, и с тех пор они стали одним из основных подходов глубокого обучения для компьютера.

2.2.3 Популярные модели: LeNet, AlexNet, VGG, ResNet

В 1980 году Кунихико Фукушима предложил архитектуру нейронной сети, которая называется неокогнитрон. Архитектура использовала аналогию со сложными и простыми клетками в зрительной коре кошки. Простые клетки

срабатывают в ответ на простые визуальные сигналы, такие как ориентация границ. Сложные клетки менее зависимы от пространственного расположения сигналов и ориентируются на более общие признаки. С этого и началась история создания и обучения нейронных сетей. С годами сверточную нейронную сеть развивали и изобретали новые архитектуры. Так в 1990 году Ян Лекун разработал архитектуру LeNet, которая стала первой практически применимой CNN. Она использовалась для распознавания рукописных цифр в банковских чеках [2].

В 2012 году Алекс Крижевски представил AlexNet, сеть, которая призвела революцию в области глубокого обучения, выиграв соревнование ImageNet с большим отрывом. Уникальность AlexNet заключалась в ее глубокой архитектуре с восемью слоями и применении GPU для ускорения вычислений. Она также внедрила Dropuot для борьбы с переобучением и ReLU в качестве функции активации, что значительно ускорило обучение.

В 2014-2015 годах были предложены глубокие архитектуры, такие как VGG (Visual Geometry Group) и ResNet (Residual Networks), которые улучшили производительность CNN за счет увеличения функции активации, что значительно ускорило обучение.

В настоящее время современные CNN продолжают развиваться, интегрируя идеи из других подходов, таких как Vision Transformers и self-supervised learning. Эти сети стремятся повысить эффективность и производительность, объединяя сильные стороны CNN (например, локальные свёртки) с глобальными механизмами внимания из трансформеров. Например, такие гибридные модели, как Swin Transformer, обеспечивают улучшенные результаты в задачах сегментации и классификации изображений. Self-supervised learning позволяет обучать модели без необходимости ручной разметки, что особенно важно для работы с большими объемами данных.

2.3 Рекуррентные нейронные сети (Recurrent Neural Networks, RNN)

2.3.1 Идея рекуррентности и развертка графа вычислений

Рекуррентные нейронные сети, или RNN – это семейство нейронных сетей для обработки последовательных данных. Если сверточная сеть предназначена для обработки сетки значений **X** типа изображения, то рекуррентная нейронная сеть предназначена для обработки последовательности значений $x^{(1)}, \dots, x^{(\tau)}$. Если сверточная сеть легко масштабируется на изображения большой ширины и высоты, а некоторые сети даже могут обрабатывать изображения переменного размера, то рекуррентная сеть масштабируется на гораздо более длинные последовательности, чем было бы практически возможно для неспециализированных нейронных сетей. Большинство рекуррентных сетей способно также обрабатывать последовательность переменной длины.

Рекуррентные нейронные сети можно строить разными способами. Как почти любую функцию можно рассматривать как нейронную сеть прямого распространения, так и практически любую рекуррентную функцию можно рассматривать как рекуррентную нейронную сеть. Во многих RNN используется уравнение $h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta)$ или аналогичное для задания значений скрытых блоков.

Такая сеть показана на рисунке 4, в типичных RNN есть дополнительные архитектурные особенности, например выходные слои, которые читают информацию из состояния h, чтобы сделать предсказание.

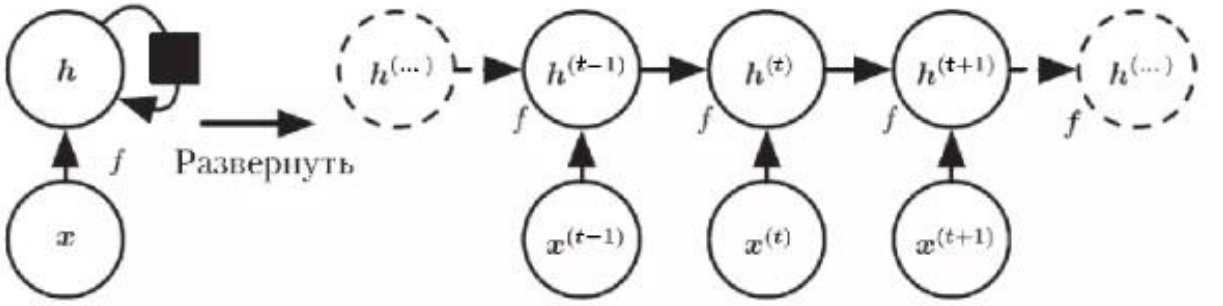

Рисунок 4 – Рекуррентная сеть без выходов

Когда рекуррентную сеть обучают решать задачу, в которой требуется предсказывать будущее по прошлому, сеть обычно обучается использовать $h^{(t)}$, как сводку, относящихся к задаче аспектов последовательности входных данных, предшествующей моменту t. В общем случае в сводке по необходимости утрачивается часть информации, потому что она отображает последовательность произвольной длины $(x^{(t)}, x^{(t-1)}, x^{(t-2)}, \dots, x^{(2)}, x^{(1)})$ на вектор фиксированной длины $h^{(t)}$. В зависимости от критерия обучения некоторые аспекты прошлой последовательности могут запоминаться в сводке с большой точностью, чем остальные. Например, если RNN используется для статистического моделирования языка, как правило, для предсказания следующего слова по известным предыдущим, то достаточно сохранить только информацию, необходимую для предсказания остатка предложения. Самая трудная ситуация складывается, когда мы хотим, чтобы вектор $h^{(t)}$ был достаточно полным для приближенного восстановления входной последовательности.

Граф вычислений – это формальный способ описать структуру множества вычислений, например необходимых для отображения входов и параметров на выходы и потерю. А развертка (unfolding) такого графа приводит к разделению параметров между структурными элементами глубокой сети [4].

Вооружившись механизмами развертки графов и разделения параметров, можно выделить несколько важных паттернов проектирования разнообразных рекуррентных сетей:

- Рекуррентные сети, порождающие выход на каждом временном шаге и имеющие рекуррентные связи между скрытыми блоками (Рисунок 5).

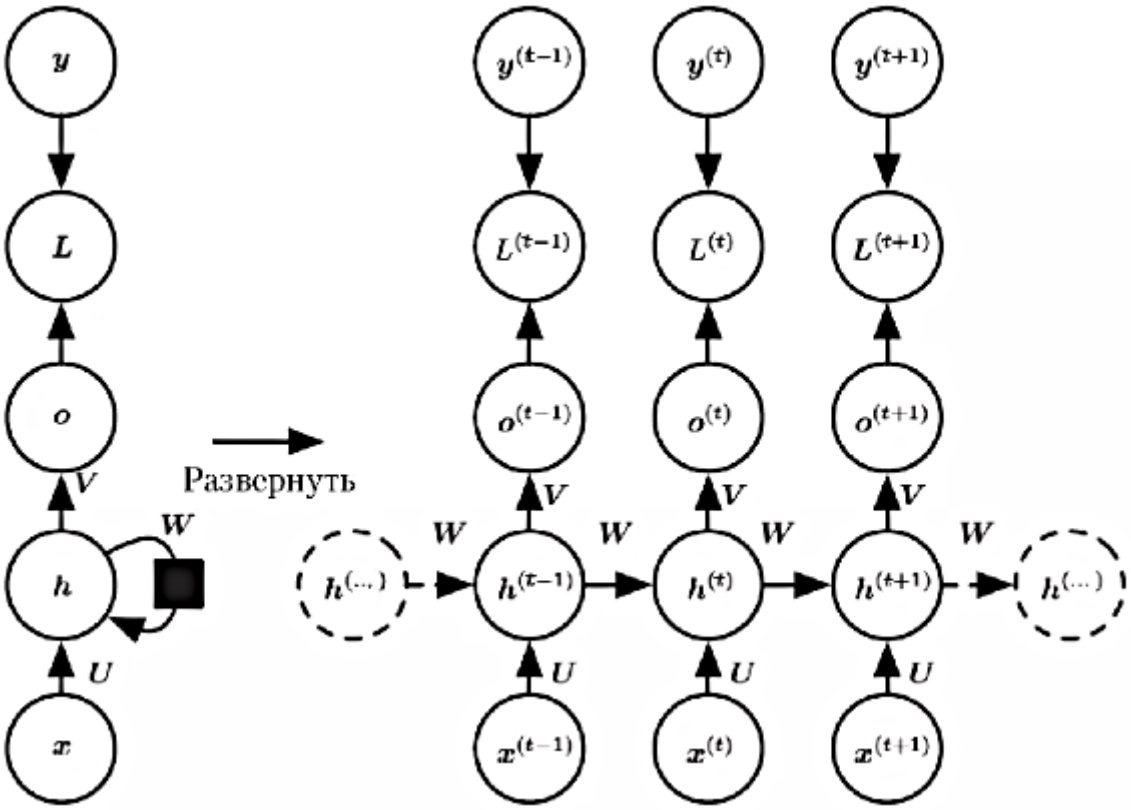

Рисунок 5 – Граф вычислений потерь при обучении рекуррентной сети

- Рекуррентные сети, порождающие выход на каждом временном шаге и имеющие рекуррентные связи только между выходами на одном временном шаге и скрытыми блоками на следующем (Рисунок 6).

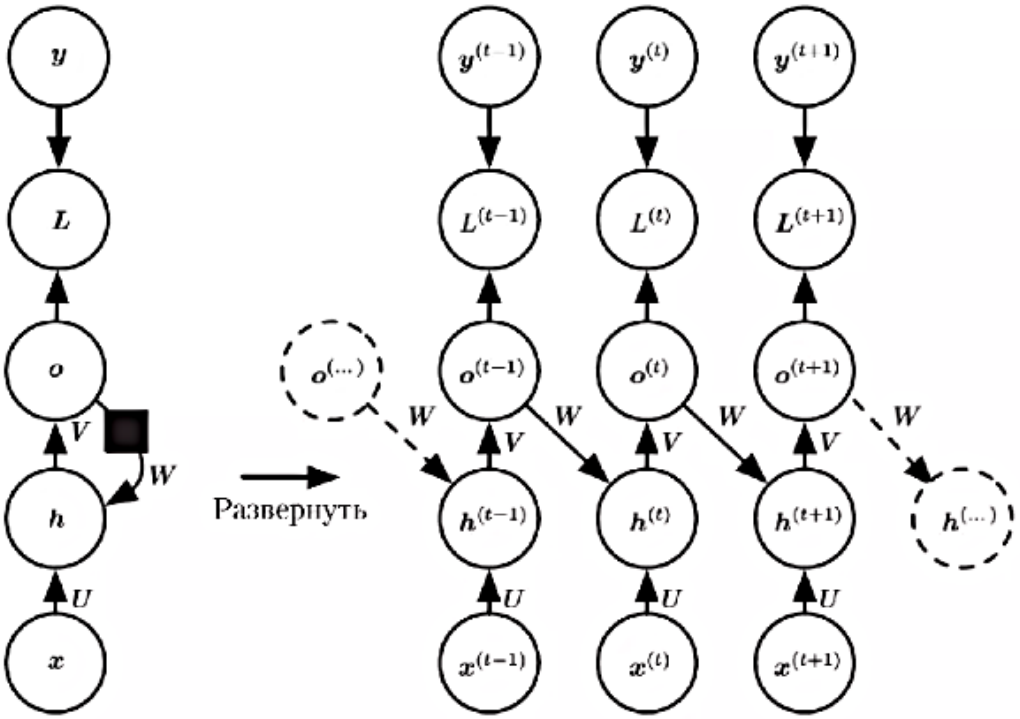

Рисунок 6 – RNN, в которой единственным видом рекурсии является обратная связь между выходным и скрытым слоями

- Рекуррентные сети с рекуррентными связями между скрытыми блоками, которые читают последовательность целиком, а затем порождают единственный выход (Рисунок 7).

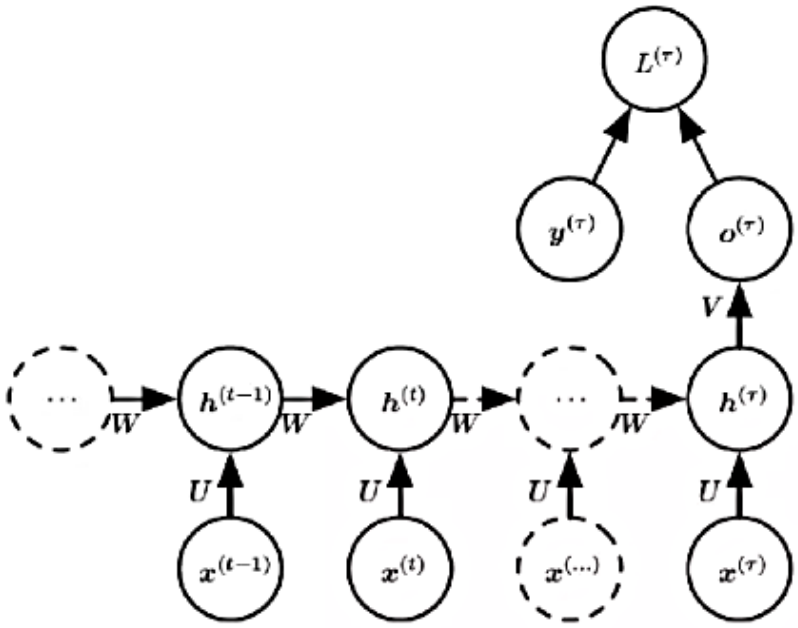

Рисунок 7 – Развернутая во времени рекуррентная нейронная сеть с единственным выходом в конце последовательности

2.3.2 Варианты архитектур: LSTM, GRU

Сети долго-краткосрочной памяти (Long Short Term Memory) - обычно просто называют “LSTM” - особый вид РНС, способных к обучению долгосрочным зависимостям. Они работают невероятно хорошо на большом разнообразии проблем и в данный момент широко применяются. LSTM специально спроектированы таким образом, чтобы избежать проблемы долгосрочных зависимостей. Запоминать информацию на длительный период времени - это практически их поведение по-умолчанию, а не что-то такое, что они только пытаются сделать.

В LSTM сетях удалось обойти проблему исчезновения или зашкаливания градиентов в процессе обучения методом обратного распространения ошибки. Сеть LSTM обычно управляется с помощью рекуррентных вентилей, которые называются вентили (gates) «забывания». Ошибки распространяются назад по времени через потенциально неограниченное количество виртуальных слоёв. Таким образом происходит обучение в LSTM, при этом сохраняя память о тысячах и даже миллионах временных интервалов в прошлом. Топологии сетей типа LSTM могут разрабатываться в соответствии со спецификой задачи. В сети LSTM даже большие задержки между значимыми событиями могут 79 учитываться, и тем самым высокочастотные и низкочастотные компоненты могут смешиваться.

Все рекуррентные нейронные сети имеют форму цепи повторяющих модулей (repeating module) нейронной сети. В стандартной РНС эти повторяющие модули будут иметь очень простую структуру (Рисунок 8), например, всего один слой гиперболического тангенса (tanhtanh).

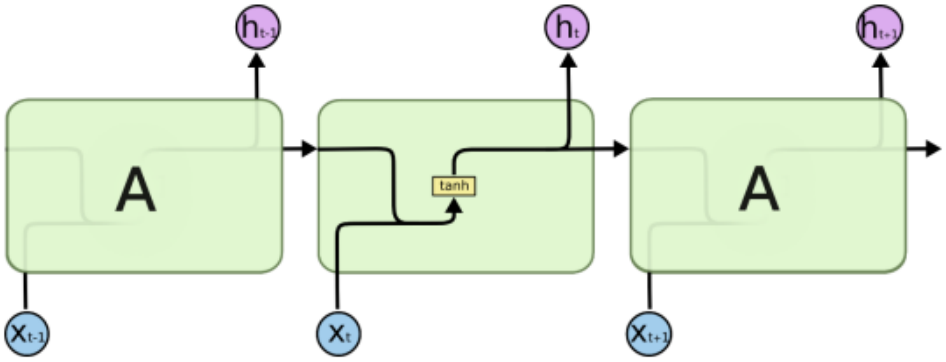

Рисунок 8 – Стандартный РНС с повторяющимися модулями

LSTM тоже имеют такую цепную структуру, но повторяющий модуль имеет другое строение. Вместо одного нейронного слоя их четыре, причем они взаимодействуют особым образом (Рисунок 9).

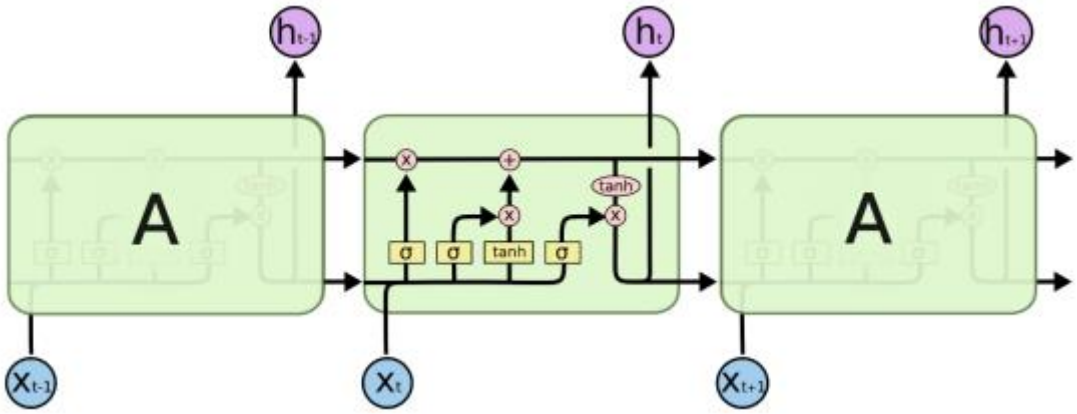

Рисунок 9 – Структура LSTM с повторяющимися модулями другого строения

На рисунке 9 каждая линия передает целый вектор от выхода одного узла к входам других. Розовые круги представляют поточечные операторы, такие как сложение векторов, в то время, как желтые 80 прямоугольники - это обученные слои нейронной сети. Сливающиеся линии обозначают конкатенацию, в то время как ветвящиеся линии обозначают, что их содержимое копируется, и копии отправляются в разные места.

Несколько более существенно отличается от LSTM вентильная рекуррентная единица (Gated Recurrent Unit) или GRU. Она совмещает

забывающие и входные вентили в один «обновляющий вентиль» («update gate»). Она также сливает клеточное состояние со скрытым слоем и вносит некоторые другие изменения. Модель, получающаяся в результате, проще, чем обычная модель LSTM и она набирает популярность [5].

2.3.3 Проблема RNN долгосрочных зависимостей

Одна из идей, которая делает РНС столь притягательными, состоит в том, что они могли бы использовать полученную в прошлом информацию для текущих задач (Рисунок 10). Например, они могли бы использовать предыдущие кадры видео для понимания последующих. Иногда нам достаточно недавней информации, чтобы выполнять текущую задачу. Например, представим модель языка, которая пытается предсказать следующее слово, основываясь на предыдущих. Если мы пытаемся предсказать последнее слово в предложении “Тучи на небе”, нам не нужен больше никакой контекст - достаточно очевидно, что в конце предложения речь идёт о небе. В таких случаях, где невелик промежуток между необходимой информацией и местом, где она нужна, РНС могут научиться использовать информацию, полученную ранее.

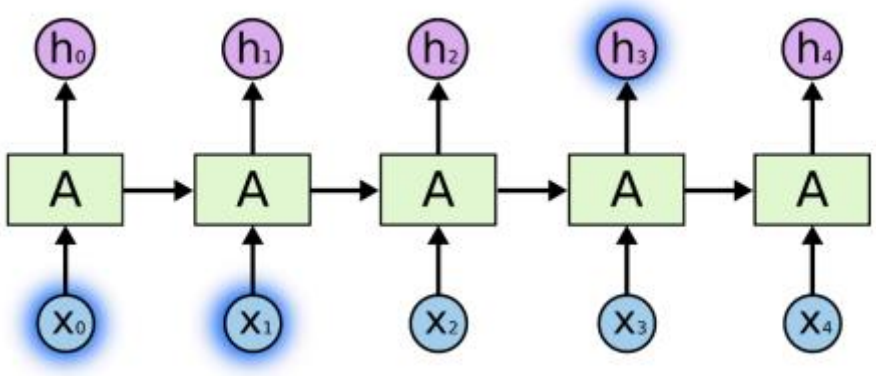

Рисунок 10 – Предсказание по контексту

Но также бывают случаи, когда нам нужен более широкий контекст. Предположим, нужно предсказать последнее слово в тексте «Я вырос во Франции… Я свободно говорю по-французски». Недавняя информация

подсказывает, что следующее слово, вероятно, название языка, но если мы хотим уточнить, какого именно, нам нужен предыдущий контекст вплоть до информации о Франции. Совсем не редко промежуток между необходимой информацией и местом, где она нужна, становится очень большим. К сожалению, по мере роста промежутка, РНС становятся неспособны научиться соединять информацию (Рисунок 11).

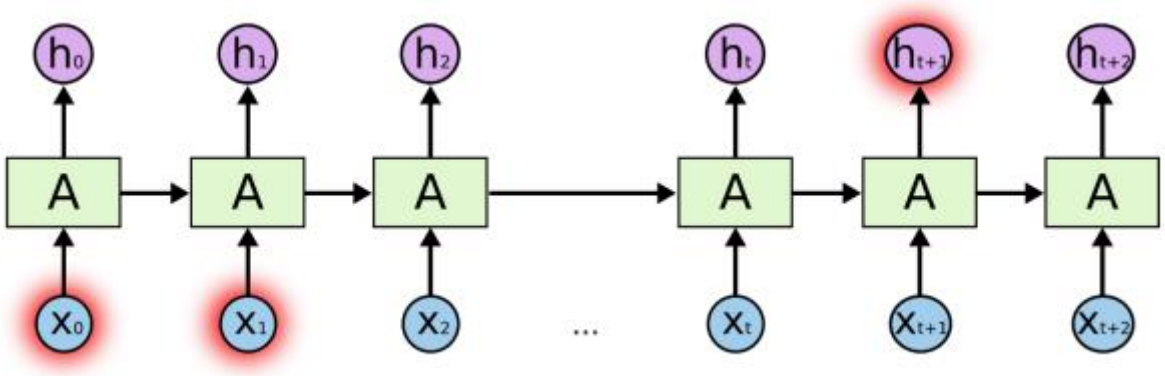

Рисунок 11 – Проблема долгосрочной зависимости

Теоретически, РНС способны обрабатывать такие долговременные зависимости. Человек может тщательно подобрать их параметры, чтобы решать игрушечные проблемы такой формы. Однако, на практике, РНС не способны выучить такое [5].

2.3.4 Области применения

Рекуррентные сети широко используются в задачах распознавания и синтеза речи, машинного перевода, генерации текста, прогнозирования временных рядов и других приложениях, связанных с обработкой последовательных данных.

2.3.4.1 Обработка естественного языка

RNN активно используются в задачах машинного перевода, где они помогают преобразовывать текст с одного языка на другой, обеспечивая сохранение смысла и контекста. В анализе тональности RNN анализируют последовательность слов, чтобы определить эмоциональную окраску текста,

такую как позитивный или негативный тон. В чат-ботах они используются для создания ответов, основываясь на предыдущих сообщениях, что делает взаимодействие более естественным. В системах распознавания речи RNN обрабатывают последовательность аудиосигналов, преобразовывая их в текст. Эти задачи требуют обработки последовательных данных, что делает RNN незаменимыми для таких приложений.

2.3.4.2 Прогнозирование временных рядов

Задачи прогнозирования временных рядов – сложный тип проблемы прогнозирующего моделирования. В отличие от регрессионного предсказательного моделирования временные ряды также добавляют сложность зависимости последовательности от входных переменных. Мощный тип нейронной сети, предназначенный для обработки последовательностей называется рекуррентными нейронными сетями. Сеть с 85 длинной короткой памятью или сеть LSTM - это тип рекуррентной нейронной сети, используемой в глубоком обучении, потому что можно успешно обучать очень большие архитектуры. В этом разделе мы разработаем ряд LSTM для стандартной задачи прогнозирования временных рядов. Эти примеры помогут вам разработать свои собственные структурированные LSTM-сети для задач прогнозирования временных рядов [5].

2.3.4.3 Генерация текстов

Сети могут создавать тексты, музыку, изображения и даже программный код, основываясь на обученных данных. Например, в области текстовой генерации рекуррентная нейронная сеть используется для написания статей, стихов или диалогов, а в музыкальной генерации — для создания мелодий в заданных стилях. Для программного кода RNN обучаются на существующих репозиториях и могут генерировать новые функции или скрипты, соответствующие заданным требованиям.

2.4 Генеративно-состязательные сети (Generative Adversarial Networks, GAN)

2.4.1 Архитектура: генератор и дискриминатор

Генерирующие состязательные сети (GAN) — это мощный класс нейронных сетей, которые используются для обучения без контроля. GAN состоят из двух нейронных сетей, дискриминатора и генератора. Они используют состязательное обучение для получения искусственных данных, идентичных реальным.

- Генератор пытается обмануть дискриминатор, который должен точно отличать полученные данные от подлинных, создавая случайные выборки шума.
- В результате этого конкурентного взаимодействия получаются реалистичные выборки высокого качества, которые способствуют продвижению обеих сетей. GAN зарекомендовали себя как универсальные инструменты искусственного интеллекта, о чем свидетельствует их широкое применение в синтезе изображений, передаче стиля и преобразовании текста в изображение.
- Они также произвели революцию в генеративном моделировании.

Благодаря состязательному обучению эти модели вступают в конкурентную борьбу до тех пор, пока генератор не научится создавать реалистичные образцы, обманывая дискриминатор примерно в половине случаев.

Генеративные состязательные сети (GAN) можно разделить на три части:

- Генеративная: для изучения генеративной модели, которая описывает, как генерируются данные в терминах вероятностной модели.
- Состязательная: Слово "состязательный" относится к противопоставлению одного объекта другому. Это означает, что в

контексте GANs результирующий результат сравнивается с фактическими изображениями в наборе данных. Механизм, известный как дискриминатор, используется для применения модели, которая пытается отличить настоящие изображения от поддельных.

- Сети: Используйте глубокие нейронные сети в качестве алгоритмов искусственного интеллекта (ИИ) для целей обучения.

Модель генератора

Ключевым элементом, ответственным за создание свежих и точных данных в генеративной состязательной сети (GAN), является модель генератора. Генератор принимает случайный шум в качестве входных данных и преобразует его в сложные выборки данных, такие как текст или изображения. Обычно это изображается как глубокая нейронная сеть.

Распределение исходных данных для обучения фиксируется слоями обучаемых параметров в процессе обучения. Генератор корректирует свои выходные данные, чтобы создавать образцы, максимально приближенные к реальным данным, по мере обучения с помощью обратного распространения ошибки для точной настройки параметров.

Успех генератора зависит от его способности генерировать высококачественные, разнообразные образцы, которые могут обмануть дискриминатор.

Модель дискриминатора

Искусственная нейронная сеть, называемая дискриминаторной моделью, используется в Генеративных состязательных сетях (GaN) для различения сгенерированных и фактических входных данных. Оценивая входные выборки и распределяя вероятность подлинности, дискриминатор функционирует как двоичный классификатор.

Со временем дискриминатор учится отличать реальные данные из набора данных от искусственных образцов, созданных генератором. Это

позволяет ему постепенно совершенствовать свои параметры и повышать уровень мастерства [6].

2.4.2 Преимущества и недостатки

Преимущества GAN заключаются в следующем:

1. Генерация синтетических данных: GAN могут генерировать новые синтетические данные, которые напоминают некоторые известные распределения данных, что может быть полезно для дополнения, обнаружения аномалий или творческих задач.
2. Высококачественные результаты: GAN могут создавать высококачественные фотореалистичные результаты при синтезе изображений, видео, музыки и другие.
3. Обучение без учителя: GAN могут обучаться без маркированных данных, что делает их пригодными для выполнения задач обучения без присмотра, где маркированные данные скудны или их трудно получить.
4. Универсальность: GAN могут применяться для широкого круга задач, включая синтез изображений, преобразование текста в изображение, перевод изображения в изображение, обнаружение аномалий, расширение данных и другие.

Недостатки GAN заключаются в следующем:

1. Нестабильность обучения: GAN может быть сложно обучать, что может привести к нестабильности, сбою режима или неспособности к сближению.
2. Вычислительные затраты: GAN могут требовать больших вычислительных ресурсов и могут быть медленными в обучении, особенно для изображений с высоким разрешением или больших наборов данных.
3. Переобучение: GAN могут переобучаться на обучающих данных, создавая синтетические данные, которые слишком похожи на обучающие данные и не отличаются разнообразием.

4. Предвзятость и несправедливость: GAN могут отражать предвзятость и несправедливость, присутствующие в обучающих данных, что приводит к дискриминационным или предвзятым синтетическим данным.
5. Интерпретируемость и подотчётность: GAN могут быть непрозрачными и сложными для интерпретации или объяснения, что затрудняет обеспечение подотчётности, прозрачности или справедливости в их применении [6].

2.4.3 Примеры применения

GAN, или генеративные состязательные сети, имеют множество применений во многих различных областях. Вот некоторые из широко признанных применений GAN.:

1. Синтез и генерация изображений: GAN-сети часто используются для задач синтеза и генерации изображений. Они могут создавать свежие, реалистичные изображения, имитирующие обучающие данные, путём изучения распределения, объясняющего набор данных. Эти типы генеративных сетей способствовали созданию реалистичных аватаров, фотографий с высоким разрешением и новых произведений искусства.
2. Перевод изображений в изображения: GAN-сети могут использоваться для решения задач, связанных с переводом изображений в изображения, где цель состоит в том, чтобы преобразовать входное изображение из одной области в другую, сохранив его ключевые особенности. GAN-сети могут использоваться, например, для преобразования дневных изображений в ночные, преобразования рисунков в реалистичные изображения или изменения творческого стиля изображения.
3. Синтез изображений из текста: GAN-модели используются для создания визуальных образов на основе текстовых описаний. GAN-модели могут создавать изображения, которые соответствуют текстовому описанию, например фразе или подписи. Это приложение может повлиять на

реалистичность визуального материала, создаваемого с помощью текстовых инструкций.

4. Расширение данных: GAN-модели могут расширять существующие данные и повышать устойчивость и обобщаемость моделей машинного обучения за счёт создания синтетических образцов данных.
5. Генерация данных для обучения: GAN могут повышать разрешение и качество изображений с низким разрешением. Обучаясь на парах изображений с низким и высоким разрешением, GAN могут генерировать изображения с высоким разрешением на основе изображений с низким разрешением, что позволяет улучшить качество изображений в различных областях, таких как медицинская визуализация, спутниковая визуализация и улучшение качества видео.

2.5 Трансформеры (Transformers) и их развитие

Трансформеры — это архитектура нейронных сетей, которая была предложена для обработки последовательностей данных, таких как текст. Впервые концепция трансформеров была представлена в 2017 году исследователями Google в статье «Attention is All You Need». Она оказала революционное влияние на области обработки естественного языка (NLP), благодаря внедрению механизмов внимания, которые позволяют моделям эффективно анализировать длинные последовательности данных.

Трансформеры устранили ограничения, связанные с рекуррентными и сверточными сетями, такие как сложности параллелизации и трудности обработки очень длинных последовательностей [7].

2.5.1 Архитектура трансформеров: механизмы внимания

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, о котором далее пойдет речь.

Ниже приведено устройство (Рисунок 12) архитектуры «трансформер» из оригинальной статьи:

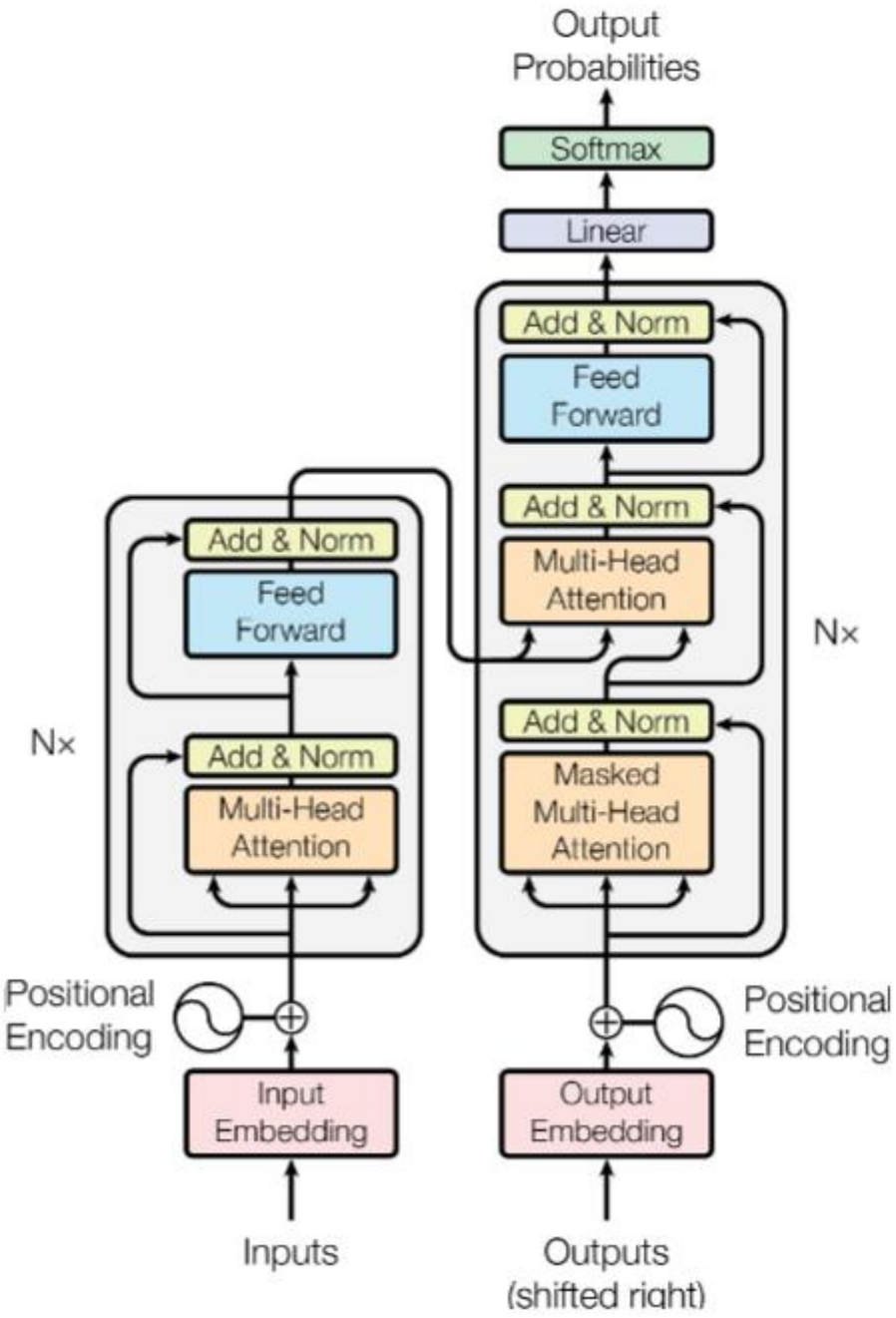

Рисунок 12 – Архитектура «Трансформер»

Слева на схеме представлено устройство энкодера. Он по очереди применяет к исходной последовательности N блоков (Рисунок 13):

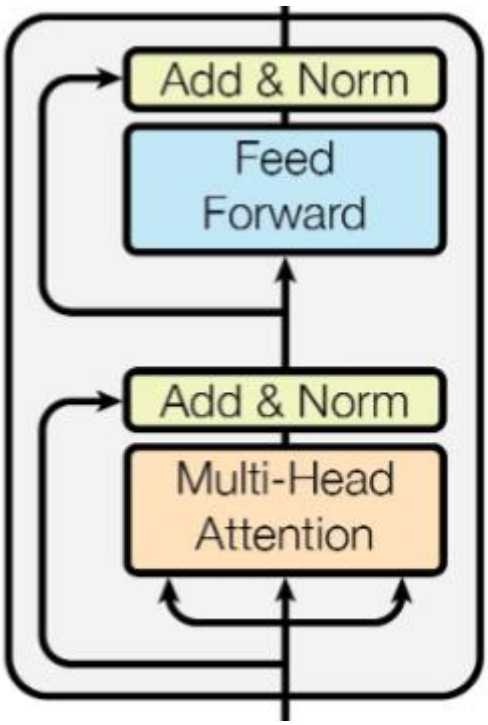

Рисунок 13 – Схема устройства энкодера

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 (это стандартный подход под названием residual connection) и затем активации проходят через слой layer normalization: на рисунке эта часть обозначена как «Add & Norm».

У декодера схема похожая, но внутри каждого из N блоков два слоя multi-head attention, в одном из которых используются выходы энкодера [8].

2.5.2 Популярные модели: BERT, GPT

Несомненно, трансформер-модели не были бы так интересны, если бы практически все задачи NLP сейчас не решались бы с помощью этой архитектуры. Главными факторами, повлиявшими на бурный рост популярности идеи self-attention, послужили два семейства хорошо всем известных архитектур — BERT и GPT, которые в некотором роде являются энкодером и декодером трансформера, которые зажили своей жизнью.

Модель GPT (Generative Pretrained Transformer) хронологически появилась раньше. Она 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). Важно, что в качестве маски внимания как раз выступает нижнетреугольная матрица: в противном случае возникла бы утечка в данных из-за того, что токены из «прошлого» будут видеть «будущее». Полученную модель можно использовать для генерации текстов и всех задач, которые на это опираются. Даже ChatGPT, обученная на специальных инструкциях, по своей сути незначительно отличается от базовой модели.

Как понятно из названия, модель BERT (Bidirectional Encoder Representations from Transformers) отличается от GPT двунаправленностью внимания: это значит, что при обработке входной последовательности все токены могут использовать информацию друг о друге.

Это делает такую архитектуру более удобной для задач, где нужно сделать предсказание относительно всего входа целиком без генерации, например, при классификации предложений или поиске пар похожих документов. Важно, что при этом BERT не учится генерировать тексты с нуля: одна из его задач при обучении — это masked language modeling (предсказание случайно замаскированных слов по оставшимся, изображено на рисунке ниже), а вторая — next sentence prediction (предсказание по паре текстовых фрагментов, следуют они друг за другом или нет).

Заметим, что самое ключевое отличие в моделях BERT и GPT (а не в задачах для обучения или применениях) можно свести к использованию разных видов внимания (Рисунок 14) [8].

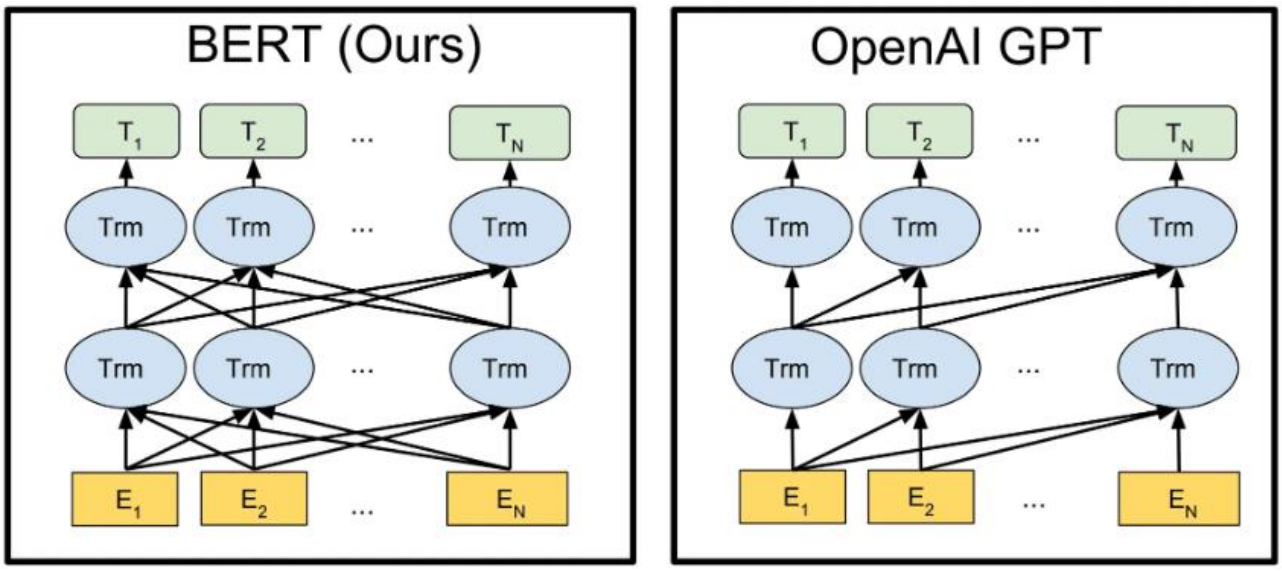

Рисунок 14 – Отличия между вниманием BERT и GPT

2.5.3 Применение в NLP

Трансформеры произвели революцию в обработке естественного языка (NLP), благодаря своей способности учитывать глобальный контекст текста и эффективно работать с большими объемами данных.

Машинный перевод стал одной из первых областей, где трансформеры показали своё превосходство. Модели, такие как Google Translate, используют архитектуры трансформеров для достижения высокой точности перевода. Например, механизм self-attention позволяет учитывать как ближайший, так и отдалённый контекст, что особенно важно для сложных языковых конструкций. Это обеспечивает перевод, который ближе к человеческому уровню понимания текста.

Модели трансформеров, такие как GPT, используются для создания интерактивных чат-ботов, способных вести осмысленные и продолжительные диалоги. Эти чат-боты находят применение в коммерческих и образовательных системах, помогая автоматизировать поддержку клиентов и обучающие программы.

Основные версии GPT:

- GPT-3 – третья версия модели, выпущенная в 2020 году. Имеет 175 миллиардов параметров. Является одной из самых больших моделей в NLP. GPT-3 демонстрирует удивительную способность генерировать тексты высокого качества по заданной теме или начальной фразе.
- GPT-3.5 – улучшенная версия GPT-3, выпущенная OpenAI в 2022 году. Содержит 280 млрд параметров вместо 175 млрд у GPT-3. Показывает более качественную генерацию текста и лучше справляется с длинными контекстами.
- GPT-4 – это четвёртая версия семейства моделей GPT, выпущенная в 2023 году. Модель знаменита улучшенной архитектурой, обеспечивающей более глубокое понимание контекста, повышение точности логических выводов и значительное улучшение в области фактической корректности. В отличие от своих предшественников, GPT-4 поддерживает мультимодальный ввод, что позволяет ей обрабатывать как текстовые данные, так и изображения. Благодаря этим изменениям модель нашла применение в более широком спектре задач, включая анализ данных, генерацию текстов высокой точности и творческие проекты. Однако точные характеристики модели, такие как количество параметров, остаются закрытыми, подчёркивая её уникальность и значимость в современной науке об искусственном интеллекте.
- GPT-5 – гипотетическая следующая версия Generative Pre-trained Transformer, о которой на момент 2024 года официальной информации не было представлено. Однако ожидания от GPT-5 включают ещё более совершенное понимание контекста, расширение мультимодальных возможностей (например, полноценное взаимодействие с текстами, изображениями, видео и аудио) и значительные улучшения в области адаптации к узким профессиональным задачам. Предполагается, что модель сможет обрабатывать ещё больший объём данных одновременно,

поддерживать более сложные логические выводы и обеспечивать более естественное взаимодействие с пользователями. Однако до официального анонса эти характеристики остаются на уровне предположений.

Таким образом, компания OpenAI продолжает активную работу по улучшению возможностей GPT в генерации и понимании естественного языка. Каждая новая версия демонстрирует все более впечатляющие результаты благодаря масштабированию моделей и новым ПО [3].

3 Перспективы развития

- Обучение на небольших данных: Одним из вызовов для нейронных сетей является разработка методов, позволяющих эффективно обучать модели на небольших объемах данных, что особенно важно для медицинских и научных приложений.
- Объяснимый ИИ: Важным направлением развития является создание объяснимых моделей нейронных сетей, способных объяснять свои решения, что позволит повысить доверие к этой технологии и расширить ее применение в чувствительных областях.
- Мультимодальность: Развитие моделей, способных работать с данными различных модальностей (изображения, текст, звук и т. д.), открывает новые возможности для создания более гибких и универсальных систем искусственного интеллекта.
- Самообучение и саморазвивающиеся системы: Будущее нейронных сетей, вероятно, связано с созданием систем, способных не только учиться на основе имеющихся данных, но и самостоятельно исследовать окружающую среду и улучшать свои навыки и знания.

ЗАКЛЮЧЕНИЕ

Нейронные сети – мощный инструмент для работы с большими объемами данных, позволяющий решить множество нетрадиционных задач за короткое время. Простота использования таких сетей заключается в их обучаемости, нет необходимости изучать различные алгоритмы и нанимать высококвалифицированных специалистов, потому как обучение происходит на примерах.

Полносвязные, сверточные и рекуррентные сети стали основой для многих современных разработок, таких как обработка изображений, временных рядов и текста. Более сложные архитектуры, такие как трансформеры и генеративные состязательные сети, позволили выйти за рамки традиционных подходов, предоставив возможности для генерации данных, автоматического перевода, анализа текста и создания мультимодальных моделей.

Перспективы развития нейронных сетей тесно связаны с совершенствованием их архитектуры, снижением потребности в больших объемах данных, а также с расширением областей применения. Упор на объяснимость моделей и их энергоэффективность станет ключевым фактором для широкого внедрения нейронных сетей в социально значимые и чувствительные сферы, такие как медицина и право. Но потенциал пока не будет раскрыт полностью, так как существует ряд проблем, которые еще решаются в настоящее время. Одной из таких проблем является недостаточная скорость передачи сигнала внутри нейронной сети, поскольку аппаратная составляющая слаба. Все зависит от того смогут ли данные передаваться вычислительными машинами со скоростью близкой к скорости человеческой мысли.

Таким образом, развитие нейронных сетей остается активной и перспективной областью, которая будет определять дальнейший прогресс искусственного интеллекта, внося значительный вклад в улучшение качество жизни общества.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Аникеев, М. В. Обзор современных типов нейронных сетей / М. В. Аникеев, Л. К. Бабенко, О. Б. Макаревич // Радиоэлектроника, информатика, управление. – 2001. – № 1. – С.48-56.
2. Созыкин, А. В. Обзор методов обучения глубоких нейронных сетей / А. В. Созыкин // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. – 2017. – Т. 6, № 3. – С. 28-59.
3. Neurones [Электронный ресурс] : Нейронные сети / текст доступен по лицензии Google Sites. – URL: https://sites.google.com/view/neurones/ (дата обращения: 17.12.2024). – Загл. с экрана. – Последние изменение страницы: 21:08, 20 августа 2024 года. – Яз. рус.
4. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль ; пер. с анг. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652с.: цв. ил.
5. Гафаров, Ф. М. Искусственные нейронные сети и их приложения: учеб. Пособие / Ф. М. Гафаров, А. Ф. Галимянов. – Казань: Изд-во Казан. ун-та, 2018. – 121с.
6. GeeksforGeeks [Электронный ресурс] : генеративные состязательные сети (GAN) / текст доступен по лицензии Create Commons Attribution-ShareAlike; GeeksforGeeks. – URL: https://www.geeksforgeeks.org/generative-adversarial-network-gan/ (дата обращения: 17.12.2024). – Загл. с экрана. – Последнее изменение страницы: 9 августа 2024 года. – Яз. анг.
7. Pussadeniya, N. [Электронный ресурс] : Attention is the Key: Understanding the Transformer Architecture / текст доступен по лицензии Medium. – URL: https://medium.com/@Nirodya_Pussadeniya/attention-is-the-key-understanding-the-transformer-architecture-38f6acc2c313 (дата обращения: 17.12.2024). – Загл. с экрана. – Последнее изменение страницы: 14 февраля 2023 года. – Яз. анг.

8. Яндекс.Образование [Электронный ресурс] : Трансформеры в машинном обучении / текст доступен по лицензии Yandex LLC; Яндекс.Образование. – URL: https://education.yandex.ru/handbook/ml/article/transformery (дата обращения: 18.12.2024). – Загл. с экрана. – Яз. рус.