 **Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет**

**имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Защита информации» (ИУ10)

**ОТЧЁТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

Тип практики: Эксплуатационная 1

Название предприятия: Федеральное государственное бюджетное учреждение «27 Центральный научно-исследовательский институт» Министерства обороны Российской Федерации

Сроки практики: с 01.07.2024 г. по 28.07.2024 г.

Специальность: 10.05.07 «Противодействие техническим разведкам»

|  |  |
| --- | --- |
| Практикант:  студент группы ИУ10-85 (4 курс)  Ершов Максим Алексеевич | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись, дата) |
| Руководитель от предприятия: |  |
| Дротенко Алексей Александрович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись, дата) |
| Руководитель от кафедры:  Вайц Екатерина Викторовна | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись, дата) |
| Оценка «отлично» |  |

*Москва*

*2024 г.*

Оглавление

[Индивидуальное задание 3](#_Toc172667023)

[Теоретическая часть 4](#_Toc172667024)

[Введение в нейронные сети 4](#_Toc172667025)

[Виды нейронных сетей 7](#_Toc172667026)

[Методы обучения нейронной сети 7](#_Toc172667027)

[Виды ошибок при применении обученной нейронной сети 9](#_Toc172667028)

[Зачем глубокое обучение в современных исследованиях и приложениях? 10](#_Toc172667029)

[Роль глубокого обучения в искусственном интеллекте 10](#_Toc172667030)

[Понимание различных форм данных 12](#_Toc172667031)

[Свойства и зависимости DL 13](#_Toc172667032)

[Методы и приложения глубокого обучения 17](#_Toc172667033)

[Глубокие сети для контролируемого или дискриминирующего обучения 18](#_Toc172667034)

[Сверточная нейронная сеть (CNN или ConvNet) 19](#_Toc172667035)

[Рекуррентная нейронная сеть (RNN) и ее варианты 20](#_Toc172667036)

[Глубокие сети для генеративного обучения или обучения без учителя 23](#_Toc172667037)

[Генеративная состязательная сеть (GAN) 24](#_Toc172667038)

[Автоматическое кодирование (AE) и его варианты 26](#_Toc172667039)

[Разреженный автоэнкодер (SAE) 26](#_Toc172667040)

[Шумоподавляющий автоэнкодер (DAE) 27](#_Toc172667041)

[Сокращенный автоэнкодер(CAE) 28](#_Toc172667042)

[Вариационный автоэнкодер (VAE) 28](#_Toc172667043)

[Карта Кохонена или самоорганизующаяся карта (SOM) 29](#_Toc172667044)

[Ограниченная машина Больцмана (RBM) 30](#_Toc172667045)

[Сеть глубокого убеждения (DBN) 30](#_Toc172667046)

[Глубокие сети для гибридного обучения и другие подходы 31](#_Toc172667047)

[Гибридные глубокие нейронные сети 32](#_Toc172667048)

[Глубокое трансферное обучение (DTL) 33](#_Toc172667049)

[Глубокое обучение с подкреплением (DRL) 35](#_Toc172667050)

[Применение нейронных сетей в военной промышленности 38](#_Toc172667051)

[Фреймворки для машинного обучения 41](#_Toc172667052)

[Практическая часть 45](#_Toc172667053)

[Цель работы 45](#_Toc172667054)

[Ход работы 45](#_Toc172667055)

[Вывод 53](#_Toc172667056)

[Список литературы 54](#_Toc172667057)

# Индивидуальное задание

Исследование нейронных сетей и принципов их построения. Создание нейронной сети для определения побочных электромагнитных излучений в сигнале

# Теоретическая **часть**

## Введение в нейронные сети

Глубокое обучение (DL), отрасль машинного обучения (ML) и искусственного интеллекта (AI), в настоящее время рассматривается как основная технология сегодняшней революции. Благодаря своим возможностям обучения на основе данных технология DL, основанная на искусственных нейронных сетях (ANN), стала актуальной темой в контексте вычислительной техники и широко применяется в различных прикладных областях, таких как здравоохранение, визуальное распознавание, текстовый анализ, кибербезопасность и многие другие. Однако построение соответствующей модели DL является сложной задачей из-за динамического характера и вариативности реальных проблем и данных.

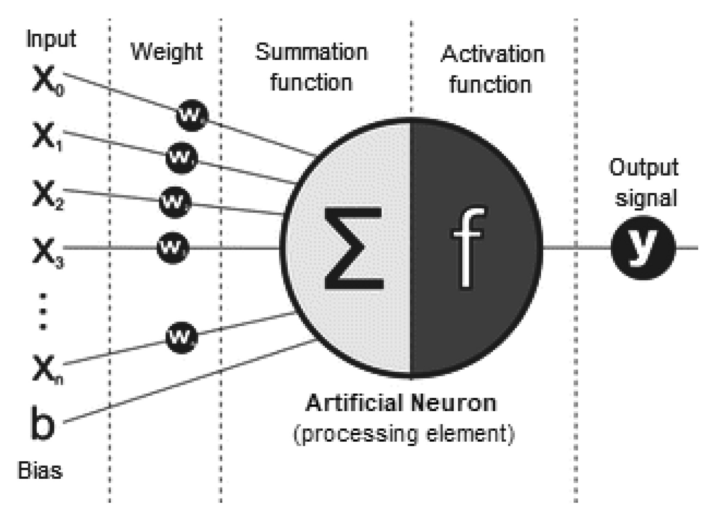
В конце 1980-х нейронные сети стали распространенной темой в области машинного обучения (ML), а также искусственного интеллекта (AI), благодаря изобретению различных эффективных методов обучения и сетевых структур. Такими инновационными методами были многослойные сети персептронов, обученные алгоритмами типа “Обратного распространения”, самоорганизующимися картами и сетями радиальных базисных функций. Хотя нейронные сети успешно используются во многих приложениях, позже интерес к исследованию этой темы снизился. После этого, в 2006 году, Хинтон и др. представили “Глубокое обучение” (DL), которое было основано на концепции искусственной нейронной сети (ANN). После этого глубокое обучение стало важной темой, что привело к возрождению исследований нейронных сетей, поэтому его иногда называют “нейронными сетями нового поколения”. Это связано с тем, что глубокие сети при надлежащем обучении добились значительных успехов в решении различных задач классификации и регрессии.

В настоящее время технология DL считается одной из актуальных тем в области машинного обучения, искусственного интеллекта, а также науки о данных и аналитики благодаря своим возможностям обучения на основе предоставленных данных. Многие корпорации, включая Google, Microsoft, Nokia и др., активно изучают это, поскольку оно может обеспечить значительные результаты в различных задачах классификации и регрессии и наборах данных. С точки зрения рабочей области, DL рассматривается как подмножество ML и искусственного интеллекта, и, таким образом, DL можно рассматривать как функцию искусственного интеллекта, имитирующую обработку данных человеческим мозгом. Всемирная популярность “глубокого обучения” растет день ото дня в день.

Глубокое обучение отличается от стандартного машинного обучения с точки зрения эффективности по мере увеличения объема данных. Технология DL использует несколько уровней для представления абстракций данных для построения вычислительных моделей. В то время как глубокое обучение требует много времени для обучения модели из-за большого количества параметров, его запуск во время тестирования занимает меньше времени по сравнению с другими алгоритмами машинного обучения.

В то время как сегодняшняя революция обычно фокусируется на “автоматизации, интеллектуальных системах”, основанных на технологиях, технология DL, созданная ANN, стала одной из ключевых технологий для достижения цели. Типичная нейронная сеть в основном состоит из множества простых, связанных обрабатывающих элементов или процессоров, называемых нейронами, каждый из которых генерирует серию действительных активаций для достижения целевого результата.

На рисунке 1 показано схематическое представление математической модели искусственного нейрона, т.е. обрабатывающего элемента, выделяющего входные данные (), вес (*w*), предвзятость (*b*), функция суммирования (∑), функция активации (*f*) и соответствующий выходной сигнал (*y*).



*Рис.1 - схематическое представление математической модели искусственного нейрона*

Технология DL на основе нейронных сетей в настоящее время широко применяется во многих областях и направлениях исследований, таких как здравоохранение, анализ настроений, обработка естественного языка, визуальное распознавание, бизнес-аналитика, кибербезопасность и многие другие.

## Виды нейронных сетей

Нейрон — это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. Связь между двумя нейронами называется синапс. У него есть один параметр - вес. Благодаря ему, входная информация изменяется, когда передается от одного нейрона к другому.

Нейронная сеть – одно из направлений искусственного интеллекта, цель которого смоделировать аналитические механизмы, осуществляемые человеческим мозгом. Нейронные сети способны самостоятельно обучаться и развиваться, строя свой опыт на совершенных ошибках.

Нейронные сети делятся на несколько видов:

* многослойные нейронные сети (перцептроны) – обрабатывают числовые данные;
* свёрточные нейронные сети - работают с изображениями;
* рекуррентные нейронные сети – собирают и обрабатывают информацию, которая меняется с течением времени;
* генеративные нейронные сети – создают контент (тексты, изображения).

Принцип работы нейронной сети можно разделить на шесть этапов:

1. Постановка задачи
2. Сбор исходных данных
3. Анализ данных
4. Обучение нейронной сети
5. Мониторинг нейронной сети
6. Дообучение нейронной сети

## Методы обучения нейронной сети

Существует два типа обучения нейронных сетей: с учителем и без него.

1. Процесс обучения с учителем

При таком процессе данные подают на вход сети, ожидая получить правильный ответ, которая даст нейронная сеть после обработки внутри своей структуры. Если результат, полученный на выходе, не совпадает с эталонным, то необходимо откорректировать весовые коэффициенты связи и запустить процесс заново, добиваясь снижения количества неправильных ответов.

Обучение с учителем подходит для решения вопросов, в которых известен требуемый результат. Например, для классификации изображений, распознавания звуков или голоса.

1. Процесс обучение без учителя

Процесс обучения без учителя предполагает наличие только входных данных. Алгоритмы обучения без учителя корректируют весовые коэффициенты таким образом, чтобы нейронная сеть могла из схожих по какому-то принципу данных на входе выдать результат, обнаруживающий другие взаимосвязи и закономерности между этими данными. В процессе обучения происходит выделение параметров, характерных для моделей обучающего материала, и дальнейшее объединение этих моделей в группировки по схожим признакам. Данные, которые поступают на вход, после обработки нейронной сетью сложатся в тот или иной ответ. Однако до обучения нельзя предугадать, в какой форме этот ответ поступит.

Обучение без учителя применяют для кластеризации, языковых моделей, обнаружения аномалий, статистических моделей.

Нейронные сети используются для решения сложных задач, которые требуют аналитических вычислений подобных тем, что делает человеческий мозг. Самыми распространенными применениями нейронных сетей является:

* Классификация — распределение данных по параметрам.
* Предсказание — возможность предсказывать следующий шаг
* Распознавание — получение выходной информации о принадлежности каждого исследуемого элемента к определенному классу.

## Виды ошибок при применении обученной нейронной сети

При применении обученной нейронной сети могут быть допущены ошибки двух родов:

* Ошибка первого рода - будет отвергнута правильная гипотеза. Её часто называют ложной тревогой, ложным срабатыванием.
* Ошибка второго рода - будет принята неправильная гипотеза. Её иногда называют пропуском события.

Ошибки первого и второго рода являются взаимно-симметричными, то есть если поменять местами гипотезы, то ошибки первого рода превратятся в ошибки второго рода и наоборот.

Вероятность ошибки первого рода при проверке статистических гипотез называют уровнем значимости и обычно обозначают греческой буквой 𝛼 (отсюда название - ошибка).

Вероятность ошибки второго рода не имеет какого-то особого общепринятого названия, она обозначается греческой буквой β (отсюда название β-ошибка). Однако с этой величиной тесно связана другая, имеющая большое статистическое значение — мощность критерия. Она вычисляется по формуле (1-β). Таким образом, чем выше мощность критерия, тем меньше вероятность совершить ошибку второго рода.

Обе эти характеристики обычно вычисляются с помощью так называемой функции мощности критерия. В частности, вероятность ошибки первого рода есть функция мощности, вычисленная при нулевой гипотезе. Для критериев, основанных на выборке фиксированного объёма, вероятность ошибки второго рода есть единица минус функция мощности, вычисленная в предположении, что распределение наблюдений соответствует альтернативной гипотезе.

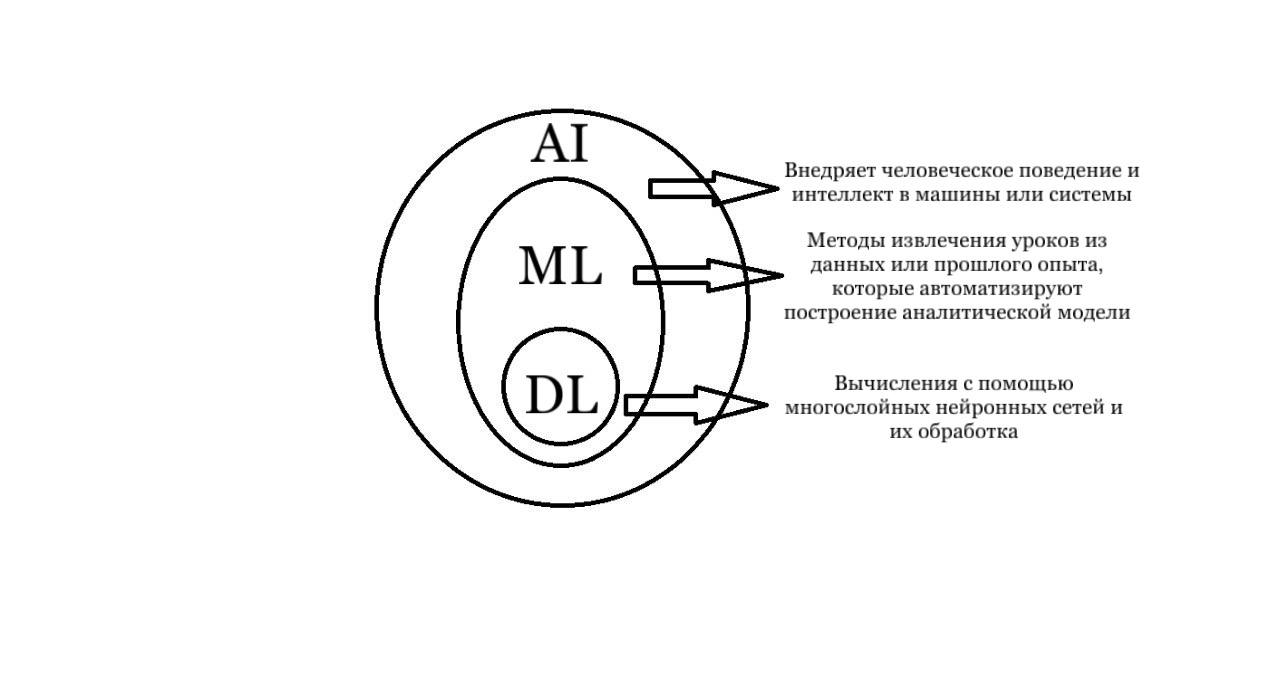
В статистических тестах обычно приходится идти на компромисс между приемлемым уровнем ошибок первого и второго рода. Зачастую для принятия решения используется пороговое значение, которое может варьироваться с целью сделать тест более строгим или, наоборот, более мягким. Этим пороговым значением является уровень значимости, которым задаются при проверке статистических гипотез.

## Зачем глубокое обучение в современных исследованиях и приложениях?

В центре внимания сегодня находится технологическая автоматизация, интеллектуальные системы в различных областях применения, включая интеллектуальное здравоохранение, бизнес-аналитику, умные города, кибербезопасность и многие другие. Подходы к глубокому обучению значительно выросли с точки зрения производительности в широком спектре приложений, особенно с учетом технологий безопасности, которые являются отличным решением для раскрытия сложной архитектуры в многомерных данных. Таким образом, методы DL могут сыграть ключевую роль в построении интеллектуальных систем, управляемых данными, в соответствии с современными потребностями, благодаря их превосходным возможностям обучения на основе исторических данных. Следовательно, DL может изменить мир, а также повседневную жизнь людей благодаря своей автоматизации и обучению на собственном опыте. Таким образом, технология DL имеет отношение к искусственному интеллекту, машинному обучению и науке о данных с расширенной аналитикой, которые являются хорошо известными областями информатики, в частности, к современным интеллектуальным вычислениям.

## Роль глубокого обучения в искусственном интеллекте

В настоящее время искусственный интеллект (AI), машинное обучение (ML) и глубокое обучение (DL) являются тремя популярными терминами, которые иногда используются как взаимозаменяемые для описания систем или программного обеспечения, которые ведут себя разумно. На рис. 2 изображено положение глубокого обучения в сравнении с машинным обучением и искусственным интеллектом. Согласно рис. 2, DL является частью ML, а также частью широкого спектра искусственного интеллекта. В целом, ИИ включает поведение человека и интеллект в машины или системы, в то время как ML - это метод обучения на основе данных или опыта, который автоматизирует построение аналитических моделей. DL также представляет методы обучения на основе данных, где вычисления выполняются с помощью многослойных нейронных сетей и их обработки. Термин “Глубокое” в методологии глубокого обучения относится к концепции множества уровней или стадий, на которых обрабатываются данные для построения модели, управляемой данными.



*Иллюстрация позиции глубокого обучения (DL) в сравнении с машинным обучением (ML) и искусственным интеллектом (AI)*

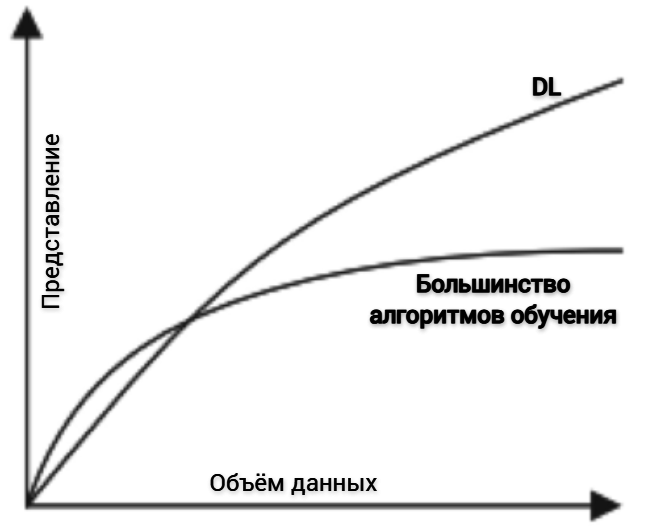
Таким образом, DL можно рассматривать как одну из ключевых технологий искусственного интеллекта, рубеж в развитии искусственного интеллекта, который может быть использован для построения интеллектуальных систем и автоматизации. Что еще более важно, оно выводит ИИ на новый уровень, называемый “Более интеллектуальным ИИ”. Поскольку DL способны обучаться на основе данных, существует тесная связь глубокого обучения и “Науки о данных”. Как правило, наука о данных представляет собой весь процесс поиска смысла или инсайтов в данных в конкретной проблемной области, где методы DL могут играть ключевую роль для продвинутой аналитики и интеллектуального принятия решений. В целом, мы можем сделать вывод, что технология DL способна изменить текущий мир, особенно с точки зрения мощного вычислительного механизма, и внести вклад в автоматизацию.

## Понимание различных форм данных

Поскольку модели DL обучаются на основе данных, глубокое понимание и представление данных важны для построения интеллектуальной системы, управляемой данными, в конкретной прикладной области. В реальном мире данные могут иметь различные формы, которые обычно могут быть представлены следующим образом для моделирования глубокого обучения:

* Последовательные данные - это любые данные, в которых важен порядок, то есть набор последовательностей. При построении модели необходимо явно учитывать последовательный характер входных данных. Текстовые потоки, аудиофрагменты, видеоклипы, данные временных рядов - вот некоторые примеры последовательных данных.
* Изображение или 2D-данные. Цифровое изображение состоит из матрицы, которая представляет собой прямоугольный массив чисел, символов или выражений, расположенных по строкам и столбцам в 2D-массиве чисел. Матрица, пиксели, воксели и разрядность - это четыре основные характеристики или параметра цифрового изображения.
* Табличные данные. Табличный набор данных состоит в основном из строк и столбцов. Таким образом, табличные наборы данных содержат данные в столбчатом формате, как в таблице базы данных. Каждый столбец (поле) должен иметь название, и каждый столбец может содержать данные только определенного типа. В целом, это логическое и систематическое расположение данных в виде строк и столбцов, которые основаны на свойствах или функциях данных. Модели глубокого обучения позволяют эффективно обучаться на табличных данных и создавать интеллектуальные системы, управляемые данными.

Рассмотренные выше формы данных распространены в реальных прикладных областях глубокого обучения. Различные категории методов DL работают по-разному в зависимости от характера и характеристик данных, кратко обсуждаемых в разделе “Методы глубокого обучения и приложения” с презентацией таксономии. Однако во многих областях применения в реальном мире стандартные методы машинного обучения, в частности, методы, основанные на логических правилах или дереве, значительно эффективны в зависимости от характера приложения. На рисунке 3 также показано сравнение производительности DL и ML моделирования с учетом объема данных.



*Рис.3 - Иллюстрация сравнения производительности между глубоким обучением (DL) и другими алгоритмами машинного обучения (ML), где моделирование DL на основе больших объемов данных может повысить производительность*

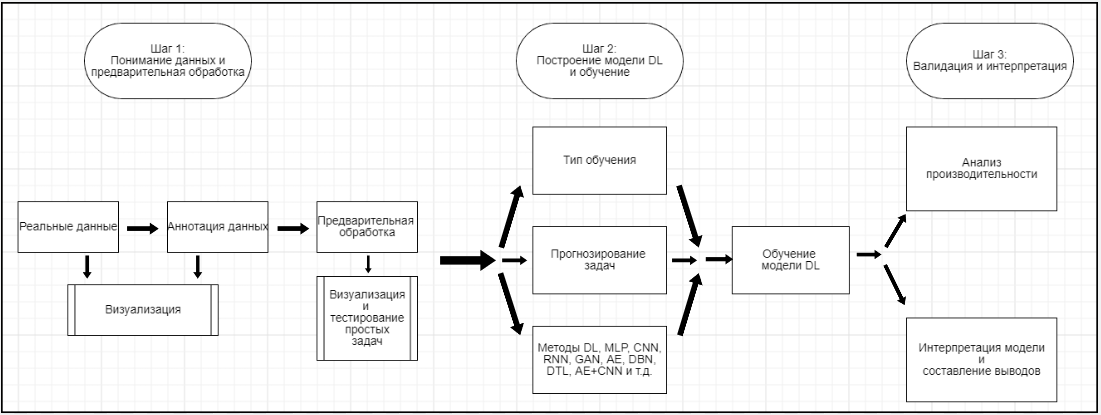
## Свойства и зависимости DL

Модель DL обычно проходит те же этапы обработки, что и моделирование с помощью машинного обучения. На рис. 4 показан рабочий процесс глубокого обучения для решения реальных задач, который состоит из трех этапов обработки, таких как понимание и предварительная обработка данных, построение модели DL и обучение, а также валидация и интерпретация. Однако, в отличие от моделирования ML, извлечение признаков в модели DL происходит автоматически, а не вручную. K-ближайший сосед, машины опорных векторов, дерево решений, случайный лес, наивный байесовский алгоритм, линейная регрессия, ассоциативные правила, кластеризация k-средних - вот некоторые примеры методов машинного обучения, которые обычно используются в различных областях применения. С другой стороны, модель DL включает в себя сверточную нейронную сеть, рекуррентную нейронную сеть, автоэнкодер, сеть глубокого убеждения и многие другие.

Ключевые свойства и зависимости методов DL, которые необходимо учитывать, прежде чем приступать к работе над моделированием DL для реальных приложений:

* Зависимости от данных. Глубокое обучение обычно зависит от большого объема данных для построения модели, управляемой данными, для конкретной проблемной области. Причина в том, что при небольшом объеме данных алгоритмы глубокого обучения часто работают плохо. Однако в таких обстоятельствах производительность стандартных алгоритмов машинного обучения будет улучшена, если будут использоваться указанные правила.
* Аппаратные зависимости. Алгоритмы DL требуют больших вычислительных операций при обучении модели с большими наборами данных. Поскольку чем больше объем вычислений, тем больше преимущество графического процессора над центральным, графический процессор в основном используется для эффективной оптимизации операций. Таким образом, для правильной работы с обучением глубокому обучению необходимо оборудование на графическом процессоре. Таким образом, DL больше полагается на высокопроизводительные машины с графическими процессорами, чем на стандартные методы машинного обучения.
* Процесс разработки объектов. Разработка объектов - это процесс извлечения объектов (характеристик, свойств и атрибутов) из необработанных данных с использованием знаний предметной области. Фундаментальным отличием DL от других методов машинного обучения является попытка извлечь высокоуровневые характеристики непосредственно из данных. Таким образом, DL сокращает время и усилия, необходимые для создания средства извлечения функций для каждой задачи.
* Обучение модели и время выполнения. В целом, обучение алгоритма глубокого обучения занимает много времени из-за большого количества параметров в алгоритме DL. Таким образом, процесс обучения модели занимает больше времени. Например, для завершения сеанса обучения моделям DL может потребоваться более недели, в то время как обучение алгоритмам ML занимает относительно немного времени, от нескольких секунд до нескольких часов. Во время тестирования запуск алгоритмов глубокого обучения занимает крайне мало времени по сравнению с некоторыми методами машинного обучения.
* Восприятие и интерпретируемость в стиле "черного ящика". Интерпретируемость является важным фактором при сравнении DL с ML. Трудно объяснить, как был получен результат глубокого обучения, то есть ”черный ящик". С другой стороны, алгоритмы машинного обучения, в частности, методы машинного обучения на основе правил предоставляют четкие логические правила (IF-THEN) для принятия решений, которые легко интерпретируются человеком.

Наиболее существенное различие между глубоким обучением и обычным машинным обучением заключается в том, насколько хорошо оно работает при экспоненциальном росте объема данных. Иллюстрация сравнения производительности между алгоритмами DL и стандартными алгоритмами ML была показана на рис. 3, где моделирование DL может повысить производительность за счет увеличения объема данных. Таким образом, DL-моделирование чрезвычайно полезно при работе с большим объемом данных из-за его способности обрабатывать огромное количество функций для построения эффективной модели, управляемой данными. С точки зрения разработки и обучения моделей DL, оно опирается на распараллеленные матричные и тензорные операции, а также вычисление градиентов и оптимизацию. Несколько библиотек и ресурсов DL, таких как PyTorch (с высокоуровневым API под названием Lightning) и TensorFlow (который также предлагает Keras в качестве высокоуровневого API), предлагают эти основные утилиты, включая множество предварительно обученных моделей, а также множество других необходимых функций для реализации и построения DL-моделей.

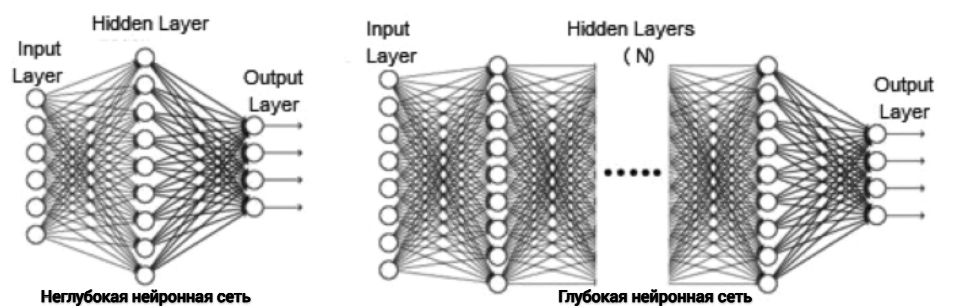


*Рис.4 - Типичный рабочий процесс DL для решения реальных задач, который состоит из трех последовательных этапов (i) понимания и предварительной обработки данных (ii) построения и обучения модели DL (iii) проверки и интерпретации*

## Методы и приложения глубокого обучения

Рассмотрим различные типы методов глубоких нейронных сетей, которые обычно учитывают несколько уровней этапов обработки информации в иерархических структурах для обучения.

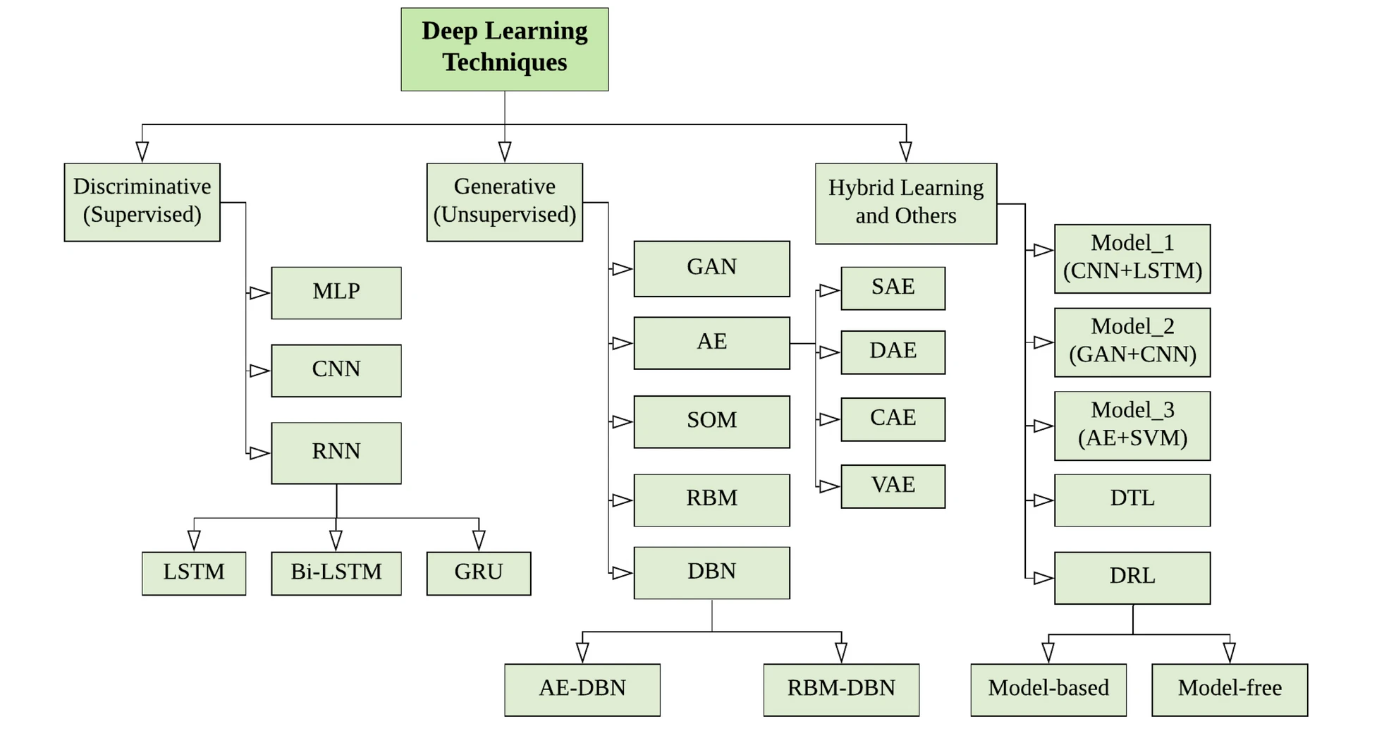
Типичная глубокая нейронная сеть содержит несколько скрытых уровней, включая входные и выходные уровни. На рисунке 5 показана общая структура глубокой нейронной сети и сравнение с неглубокой сетью.



*Рис.5 - Общая архитектура. Мелкая сеть с одним скрытым слоем и глубокая нейронная сеть с несколькими скрытыми слоями*

Чтобы представить нашу таксономию, мы разделяем методы DL в целом на три основные категории:

* глубокие сети для контролируемого или дискриминационного обучения
* (ii) глубокие сети для неконтролируемого или генеративного обучения
* (ii) глубокие сети для гибридного обучения, объединяющего оба вида обучения и соответствующие другие, как показано на рис. 6.



*Рис.6 - Таксономия методов DL*

Далее кратко рассмотрим каждый из этих методов, которые могут быть использованы для решения реальных задач в различных прикладных областях в соответствии с их возможностями обучения.

## Глубокие сети для контролируемого или дискриминирующего обучения

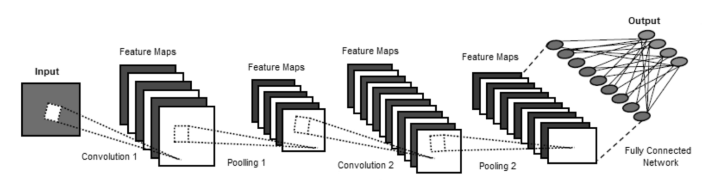
Эта категория методов DL используется для обеспечения дискриминационной функции в приложениях с контролем или классификацией. Дискриминирующие глубокие архитектуры обычно предназначены для предоставления дискриминирующих возможностей для классификации шаблонов путем описания апостериорных распределений классов, обусловленных видимыми данными. Дискриминационные архитектуры в основном включают многослойный персептрон (MLP), сверточные нейронные сети (CNN или ConvNet), рекуррентные нейронные сети (RNN) вместе с их вариантами.

###### Многослойный персептрон (MLP)

Многослойный персептрон (MLP), метод контролируемого обучения, который представляет собой тип искусственной нейронной сети с прямой связью (ANN). Он также известен как базовая архитектура глубоких нейронных сетей (DNN) или глубокое обучение. Типичный MLP - это полностью подключенная сеть, состоящая из входного уровня, который принимает входные данные, выходного уровня, который принимает решение или прогнозирует входной сигнал, и одного или нескольких скрытых уровней между этими двумя, которые рассматриваются как вычислительный механизм сети. Выходные данные MLP-сети определяются с использованием различных функций активации, также известных как передаточные функции, таких как ReLU (выпрямленная линейная единица), Tanh, Сигмоидальная и Softmax. Для обучения MLP используется наиболее широко используемый алгоритм “Обратного распространения”, метод контролируемого обучения, который также известен как самый базовый строительный блок нейронной сети. В процессе обучения применяются различные подходы к оптимизации, такие как стохастический градиентный спуск (SGD), BFGS с ограниченной памятью (L-BFGS) и адаптивная оценка момента (Adam). MLP требует настройки нескольких гиперпараметров, таких как количество скрытых слоев, нейронов и итераций, что может сделать вычислительно дорогостоящим решение сложной модели. Однако благодаря частичной подгонке MLP дает преимущество в изучении нелинейных моделей в режиме реального времени или онлайн.

### Сверточная нейронная сеть (CNN или ConvNet)

Сверточная нейронная сеть (CNN или ConvNet) - популярная дискриминирующая архитектура глубокого обучения, которая обучается непосредственно на основе входных данных без необходимости извлечения функций человеком. На рисунке 7 показан пример CNN, включающий несколько сверток и слоев объединения. В результате CNN улучшает дизайн традиционных ANN, подобных упорядоченным MLP-сетям. Каждый уровень в CNN учитывает оптимальные параметры для получения значимого результата, а также снижает сложность модели. CNN также использует "отсев", который может решить проблему перенастройки, которая может возникнуть в традиционной сети.



*Рис. 7 - Пример сверточной нейронной сети (CNN или ConvNet), включающей несколько уровней свертки и объединения*

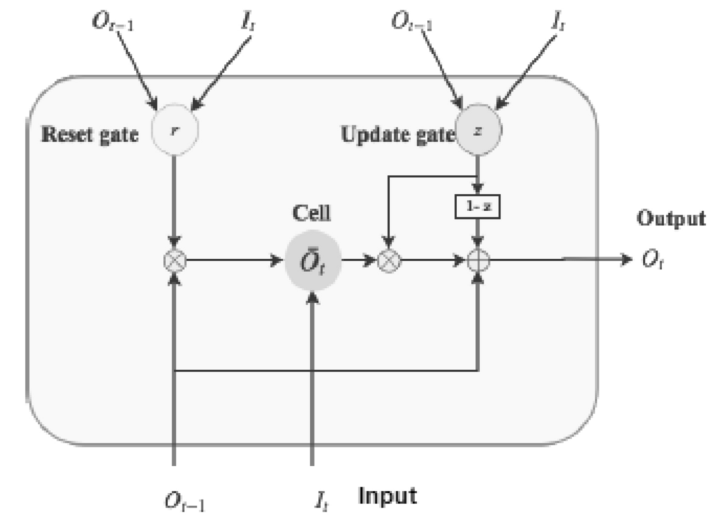
CNN специально предназначены для работы с различными 2D-формами и, таким образом, широко используются в визуальном распознавании, анализе медицинских изображений, сегментации изображений, обработке естественного языка и многом другом. Возможность автоматического обнаружения важных функций на основе входных данных без необходимости вмешательства человека делает его более мощным, чем традиционная сеть. В этой области существует несколько вариантов CNN, которые включают visual geometry group (VGG), AlexNet, Xception, Inception, ResNet и т.д., которые могут использоваться в различных прикладных областях в соответствии с их возможностями обучения.

### Рекуррентная нейронная сеть (RNN) и ее варианты

Рекуррентная нейронная сеть (RNN) - еще одна популярная нейронная сеть, которая использует последовательные данные или временные ряды и передает выходные данные предыдущего этапа в качестве входных данных для текущего этапа. Как и CNN, рекуррентные сети обучаются на основе входных данных обучения, однако отличаются своей “памятью”, которая позволяет им влиять на текущие входные данные и выходные данные за счет использования информации из предыдущих входных данных. В отличие от обычного DNN, который предполагает, что входные и выходные данные независимы друг от друга, выходные данные RNN зависят от предшествующих элементов в последовательности. Однако стандартные рекуррентные сети сталкиваются с проблемой исчезновения градиентов, что затрудняет изучение длинных последовательностей данных.

Рассмотрим несколько популярных вариантов рекуррентной сети, которые сводят к минимуму проблемы и хорошо работают во многих областях реальных приложений:

* Долговременная кратковременная память (LSTM). Это популярная форма архитектуры RNN, которая использует специальные модули для решения проблемы исчезающего градиента, которая была представлена Хохрайтером. Ячейка памяти в блоке LSTM может хранить данные в течение длительного времени, а потоком информации в ячейку и из нее управляют три шлюза. Например, "Элемент забывания" определяет, какая информация из ячейки предыдущего состояния будет сохранена в памяти, а какая информация, которая больше не является полезной, будет удалена, в то время как "Элемент ввода" определяет, какая информация должна поступать в состояние ячейки, а "Элемент вывода’ определяет выходные данные и управляет ими. Поскольку сеть LSTM решает проблемы обучения рекуррентной сети, она считается одной из наиболее успешных RNN.
* Двунаправленный RNN / LSTM. Двунаправленные RNN соединяют два скрытых слоя, которые работают в противоположных направлениях, с единым выводом, позволяя им принимать данные как из прошлого, так и из будущего. Двунаправленные RNN, в отличие от традиционных рекуррентных сетей, обучены предсказывать как положительное, так и отрицательное направление времени одновременно. Двунаправленный LSTM, часто известный как BiLSTM, является расширением стандартного LSTM, которое может повысить производительность модели в вопросах классификации последовательностей. Это модель обработки последовательностей, состоящая из двух LSTM: одна передает входные данные вперед, а другая - назад. Двунаправленный LSTM, в частности, является популярным выбором в задачах обработки естественного языка.
* Закрытые рекуррентные блоки (GRU). Закрытый рекуррентный блок (GRU) - еще один популярный вариант рекуррентной сети, использующий методы стробирования для контроля и управления информационным потоком между ячейками в нейронной сети. GRU похож на LSTM, однако имеет меньше параметров, поскольку у него есть элементы сброса и обновления, но отсутствует элемент вывода, как показано на рис. 8. Таким образом, ключевое различие между GRU и LSTM заключается в том, что GRU имеет два элемента (элементы сброса и обновления), тогда как LSTM имеет три элемента (а именно элементы ввода, вывода и забывания). Структура GRU позволяет адаптивно фиксировать зависимости от больших последовательностей данных, не отбрасывая информацию из более ранних частей последовательности. Таким образом, GRU - это немного более упрощенный вариант, который часто предлагает сопоставимую производительность и значительно быстрее в вычислениях. Хотя было показано, что GRU демонстрируют лучшую производительность на определенных небольших и менее частых наборах данных, оба варианта RNN доказали свою эффективность при получении результата.



*Рис. 8 - Базовая структура ячейки GRU, состоящая из элементов сброса и обновления*

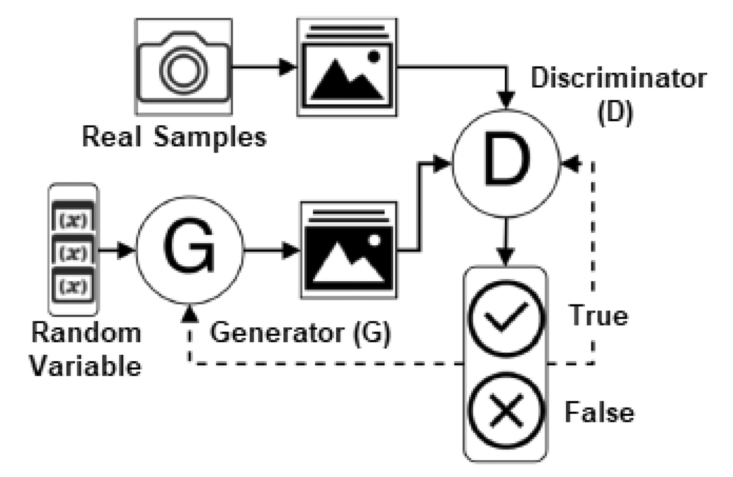
В целом, основным свойством рекуррентной сети является то, что она имеет по крайней мере одно соединение с обратной связью, которое позволяет выполнять активации по циклу. Это позволяет сетям выполнять временную обработку и изучение последовательностей, таких как распознавание или воспроизведение последовательностей, временная ассоциация или прогнозирование и т.д.

## Глубокие сети для генеративного обучения или обучения без учителя

Эта категория методов DL обычно используется для характеристики корреляционных свойств высокого порядка или признаков для анализа или синтеза шаблонов, а также совместных статистических распределений видимых данных и связанных с ними классов. Ключевая идея генеративных глубоких архитектур заключается в том, что в процессе обучения точная контрольная информация, такая как метки целевых классов, не вызывает беспокойства. В результате методы этой категории в основном применяются для обучения без учителя, поскольку методы обычно используются для изучения функций или генерации и представления данных. Таким образом, генеративное моделирование может использоваться и в качестве предварительной обработки для контролируемых учебных задач, что обеспечивает точность дискриминантной модели. Обычно используемыми методами глубоких нейронных сетей для неконтролируемого или генеративного обучения являются генеративная состязательная сеть (GAN), автоэнкодер (AE), ограниченная машина Больцмана (RBM), самоорганизующаяся карта (SOM) и сеть глубокого убеждения (DBN) вместе с их вариантами.

### Генеративная состязательная сеть (GAN)

Генеративная состязательная сеть (GAN), разработанная Яном Гудфеллоу, представляет собой тип архитектуры нейронной сети для генеративного моделирования для создания новых правдоподобных образцов по запросу. Оно включает автоматическое обнаружение и изучение закономерностей или шаблонов во входных данных, чтобы модель могла использоваться для генерации или вывода новых примеров из исходного набора данных. Как показано на рис. 9, GAN состоят из двух нейронных сетей: генератора G, который создает новые данные, обладающие свойствами, аналогичными исходным данным, и дискриминатора D, который предсказывает вероятность того, что последующая выборка будет взята на основе фактических данных, а не данных, предоставленных генератором. Таким образом, в GAN-моделировании и генератор, и дискриминатор обучены конкурировать друг с другом. В то время как генератор пытается обмануть дискриминатор, создавая более реалистичные данные, дискриминатор пытается отличить подлинные данные от поддельных, сгенерированных G.



*Рис.9 - Схематическая структура стандартной генеративной состязательной сети (GAN)*

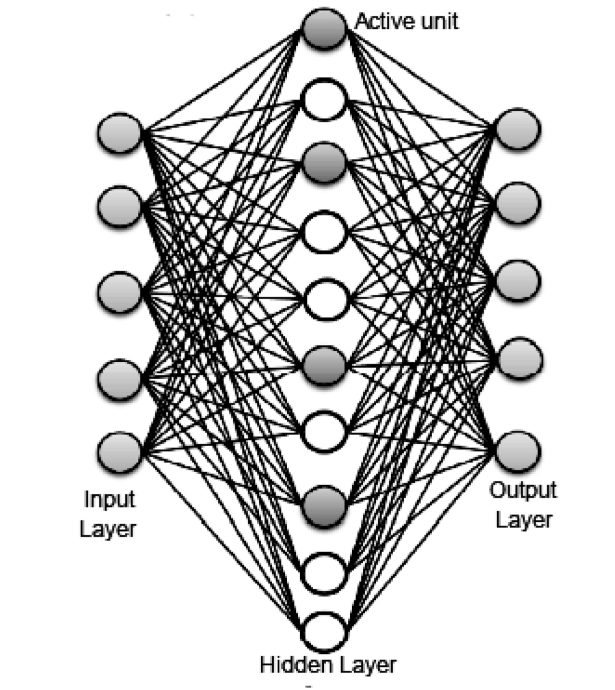
Как правило, развертывание сети GAN предназначено для задач обучения без учителя, но оно также зарекомендовало себя как лучшее решение для обучения без учителя и с подкреплением, а также в зависимости от задачи. GAN также используются в современных исследованиях трансфертного обучения для обеспечения выравнивания пространства скрытых признаков. Обратные модели, такие как двунаправленный GAN (BiGAN), также могут изучать отображение данных в скрытое пространство, аналогично тому, как стандартная модель GAN изучает отображение скрытого пространства в распределение данных. Потенциальными областями применения сетей GAN являются здравоохранение, анализ изображений, увеличение объема данных, генерация видео, голоса, пандемии, контроль трафика, кибербезопасность и многие другие, число которых быстро растет. В целом, GAN зарекомендовали себя как всеобъемлющая область независимого расширения данных и как решение проблем, требующих генеративного решения.

### Автоматическое кодирование (AE) и его варианты

Автоматическое кодирование (AE) - популярная технология обучения без учителя, в которой нейронные сети используются для изучения представлений. Обычно автокодеры используются для работы с многомерными данными, а уменьшение размерности объясняет, как представляется набор данных. Кодировщик, код и декодер - это три части автоэнкодера. Кодировщик сжимает входные данные и генерирует код, который декодер впоследствии использует для восстановления входных данных. AES недавно использовались для изучения генеративных моделей данных. Автокодер широко используется во многих обучающих задачах без учителя, например, при уменьшении размерности, извлечении признаков, эффективном кодировании, генеративном моделировании, шумоподавлении, обнаружении аномалий или выбросов и т.д.. Анализ главных компонент (PCA), который также используется для уменьшения размерности огромных наборов данных, по сути, аналогичен однослойному AE с линейной функцией активации. Регуляризованные автоэнкодеры, такие как шумоподавляющие и сокращающие, полезны для изучения представлений для последующих задач классификации, в то время как вариационные автоэнкодеры могут использоваться в качестве генеративных моделей.

### Разреженный автоэнкодер (SAE)

Разреженный автоэнкодер имеет ограничение по разреженности на уровне кодирования как часть требований к обучению. В SAE может быть больше скрытых модулей, чем входных данных, но только небольшому количеству скрытых модулей разрешено быть активными одновременно, что приводит к разреженной модели. На рисунке 10 показана схематическая структура разреженного автоэнкодера с несколькими активными блоками в скрытом слое. Таким образом, эта модель обязана реагировать на уникальные статистические характеристики обучающих данных в соответствии с ее ограничениями.



*Рис. 10 - Схематическая структура разреженного автоэнкодера (SAE) с несколькими активными блоками (закрашенный круг) в скрытом слое*

### Шумоподавляющий автоэнкодер (DAE)

Шумоподавляющий автоэнкодер - это вариант базового автоэнкодера, который пытается улучшить представление (извлечь полезные функции) путем изменения критерия восстановления и, таким образом, снижает риск изучения функции идентификации. Другими словами, он получает поврежденную точку данных в качестве входных данных и обучается восстанавливать исходные неискаженные входные данные в качестве выходных данных путем минимизации средней ошибки восстановления по обучающим данным, т.е. очистки поврежденных входных данных или шумоподавления. Таким образом, в контексте вычислительной техники DAE можно рассматривать как очень мощные фильтры, которые можно использовать для автоматической предварительной обработки. Например, автоэнкодер с шумоподавлением может использоваться для автоматической предварительной обработки изображения, тем самым повышая его качество для точности распознавания.

### Сокращенный автоэнкодер(CAE)

Идея сокращенного автоэнкодера, предложенная Рифаи заключается в том, чтобы сделать автоэнкодеры устойчивыми к небольшим изменениям в обучающем наборе данных. В своей целевой функции CAE включает явный регуляризатор, который заставляет модель изучать кодировку, устойчивую к небольшим изменениям входных значений. В результате чувствительность изученного представления к вводимым данным обучения снижается. В то время как DAE способствуют надежности реконструкции, как обсуждалось выше, CAE способствуют надежности представления.

### Вариационный автоэнкодер (VAE)

Вариационный автоэнкодер обладает принципиально уникальным свойством, которое отличает его от классического автоэнкодера, рассмотренного выше, что делает его таким эффективным для генеративного моделирования. VAEs, в отличие от традиционных автоэнкодеров, которые отображают входные данные в скрытый вектор, преобразуют входные данные в параметры распределения вероятностей, такие как среднее значение и дисперсия распределения Гаусса. VAE предполагает, что исходные данные имеют базовое распределение вероятностей, а затем пытается обнаружить параметры распределения. Хотя этот подход изначально был разработан для обучения без учителя, его использование было продемонстрировано в других областях, таких как обучение без учителя и обучение под наблюдением.

Хотя более ранняя концепция AE обычно предназначалась для уменьшения размерности или изучения функций, упомянутых выше, в последнее время AE были выдвинуты на передний план генеративного моделирования, даже генеративная состязательная сеть является одним из популярных методов в этой области. AES эффективно используются в различных областях, включая здравоохранение, компьютерное зрение, распознавание речи, кибербезопасность, обработку естественного языка и многие другие. В целом, мы можем сделать вывод, что автоматическое кодирование и его варианты могут играть значительную роль в обучении функциям без учителя с использованием архитектуры нейросети.

### Карта Кохонена или самоорганизующаяся карта (SOM)

Самоорганизующаяся карта (SOM) или карта Кохонена - это еще одна форма неконтролируемого метода обучения для создания низкоразмерного (обычно двумерного) представления многомерного набора данных при сохранении топологической структуры данных. SOM также известен как алгоритм уменьшения размерности на основе нейронной сети, который обычно используется для кластеризации. SOM адаптируется к топологической форме набора данных, многократно перемещая свои нейроны ближе к точкам данных, что позволяет нам визуализировать огромные наборы данных и находить вероятные кластеры. Первый уровень SOM - это входной уровень, а второй уровень - выходной уровень или карта объектов. В отличие от других нейронных сетей, которые используют обучение с исправлением ошибок, например, обратное распространение с градиентным спуском. В СОМ используется конкурентное обучение, в котором используется функция соседства для сохранения топологических характеристик входного пространства. SOM широко используется в различных приложениях, включая идентификацию шаблонов, диагностику состояния здоровья или медицинских заболеваний, обнаружение аномалий и обнаружение атак вирусов или червей. Основное преимущество использования SOM заключается в том, что это может упростить визуализацию и анализ многомерных данных для понимания закономерностей. Уменьшение размерности и кластеризация сетки позволяют легко наблюдать сходство в данных. В результате SOMs может сыграть жизненно важную роль в разработке эффективной модели, основанной на данных, для конкретной проблемной области, в зависимости от характеристик данных.

### Ограниченная машина Больцмана (RBM)

Ограниченная машина Больцмана (RBM) также представляет собой генеративную стохастическую нейронную сеть, способную изучать распределение вероятностей по входным данным. Машины Больцмана обычно состоят из видимых и скрытых узлов, и каждый узел связан со всеми остальными узлами, что помогает нам понять нарушения, изучая, как система работает в обычных условиях. RBM - это подмножество машин Больцмана, которые имеют ограничение на количество соединений между видимым и скрытым слоями. Это ограничение позволяет алгоритмам обучения, таким как алгоритм контрастного расхождения на основе градиента, быть более эффективными, чем алгоритмы для машин Больцмана в целом. RBM нашли применение в уменьшении размерности, классификации, регрессии, совместной фильтрации, изучении объектов, тематическом моделировании и многих других. В области моделирования глубокого обучения они могут обучаться как под наблюдением, так и без него, в зависимости от задачи. В целом, RBMS может автоматически распознавать закономерности в данных и разрабатывать вероятностные или стохастические модели, которые используются для выбора или извлечения признаков, а также для формирования глубокой сети доверия.

### Сеть глубокого убеждения (DBN)

Сеть глубоких убеждений (DBN) представляет собой многоуровневую генеративную графическую модель объединения нескольких отдельных неконтролируемых сетей, таких как AEs или RBMS, которые используют скрытый уровень каждой сети в качестве входных данных для следующего уровня, то есть подключаются последовательно. Таким образом, мы можем разделить DBN на (i) AE-DBN, известную как stacked AE, и (ii) RBM-DBN, известную как stacked RBM, где AE-DBN состоит из автокодеров, а RBM-DBN состоит из ограниченных машин Больцмана, рассмотренных ранее. Конечная цель - разработать методику обучения без учителя для каждой подсети, которая зависит от контрастного расхождения. DBN может фиксировать иерархическое представление входных данных на основе их глубокой структуры. Основная идея DBN заключается в обучении неконтролируемых нейронных сетей прямой связи с немаркированными данными перед точной настройкой сети с помощью помеченных входных данных. Одним из наиболее важных преимуществ DBN, в отличие от типичных сетей поверхностного обучения, является то, что он позволяет обнаруживать глубокие закономерности, что позволяет развивать способности к рассуждению и улавливать глубокую разницу между нормальными и ошибочными данными.

Непрерывный DBN - это просто расширение стандартного DBN, которое допускает непрерывный диапазон десятичных чисел вместо двоичных данных. В целом, модель DBN может сыграть ключевую роль в широком спектре приложений для обработки многомерных данных благодаря своим мощным возможностям извлечения признаков и классификации и стать одной из важных тем в области нейронных сетей.

Подводя итог, можно сказать, что методы генеративного обучения обычно позволяют нам генерировать новое представление данных посредством поискового анализа. В результате эти глубокие генеративные сети могут использоваться в качестве предварительной обработки для задач контролируемого или дискриминационного обучения, а также для обеспечения точности модели, где обучение представлению без учителя может позволить улучшить обобщение классификатора.

## Глубокие сети для гибридного обучения и другие подходы

В дополнение к рассмотренным выше категориям глубокого обучения популярны гибридные глубокие сети и несколько других подходов, таких как глубокое трансферное обучение (DTL) и глубокое обучение с подкреплением (DRL).

### Гибридные глубокие нейронные сети

Генеративные модели поддаются адаптации и позволяют учиться как на помеченных, так и на немаркированных данных. Дискриминационные модели, с другой стороны, не способны учиться на немаркированных данных, но при этом превосходят свои генеративные аналоги в контролируемых задачах. Платформа для одновременного обучения как глубоким генеративным, так и дискриминационным моделям может использовать преимущества обеих моделей, что стимулирует развитие гибридных сетей.

Гибридные модели глубокого обучения обычно состоят из нескольких (двух или более) моделей глубокого базового обучения, где базовой моделью является дискриминационная или генеративная модель глубокого обучения. Основанные на интеграции различных базовых генеративных или дискриминационных моделей, нижеприведенные три категории гибридных моделей глубокого обучения могут быть полезны для решения реальных проблем. К ним относятся следующие:

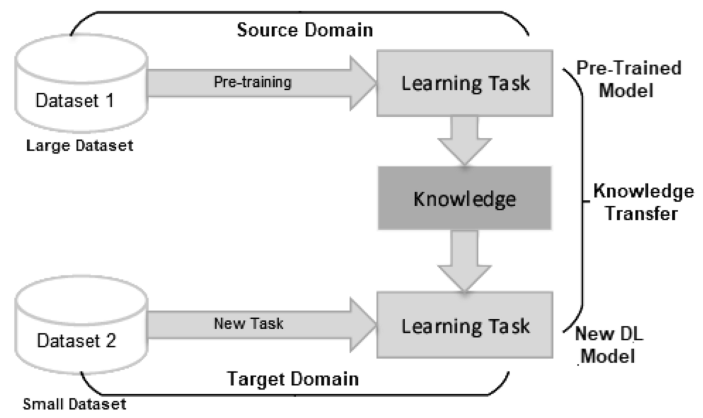
* Гибрид Model 1: Интеграция различных генеративных или дискриминационных моделей для извлечения более значимых и надежных функций. Примерами могут быть CNN + LSTM, AE + GAN и так далее.
* Гибрид Model 2: Интеграция генеративной модели с последующей дискриминационной моделью. Примерами могут быть DBN + MLP, GAN + CNN, AE + CNN и так далее.
* Гибрид Model 3: Интеграция генеративной или дискриминационной модели с последующим классификатором, не связанным с глубоким обучением. Примерами могут быть AE + SVM, CNN + SVM и так далее.

Таким образом, в широком смысле мы можем заключить, что гибридные модели могут быть либо классификационно-ориентированными, либо неклассификационными в зависимости от целевого использования. Тем не менее, большинство исследований, связанных с гибридным обучением в области глубокого обучения, ориентированы на классификацию или обучающие задачи под контролем.

Неконтролируемые генеративные модели со значимыми представлениями используются для улучшения дискриминационных моделей. Генеративные модели с полезным представлением могут предоставлять более информативные и маломерные характеристики для различения, а также они могут позволить улучшить качество и количество обучающих данных, предоставляя дополнительную информацию для классификации.

### Глубокое трансферное обучение (DTL)

Трансфертное обучение - это метод эффективного использования ранее полученных знаний о модели для решения новой задачи с минимальной подготовкой или точной настройкой. По сравнению с типичными методами машинного обучения, DL требует большого объема обучающих данных. В результате потребность в значительном объеме маркированных данных является серьезным препятствием для решения некоторых важных задач, специфичных для конкретной предметной области, особенно в медицинском секторе, где создание крупномасштабных высококачественных аннотированных медицинских наборов данных является сложным и дорогостоящим. Кроме того, стандартная модель DL требует больших вычислительных ресурсов, таких как сервер с поддержкой GPU, хотя исследователи упорно работают над ее улучшением. В результате для решения этой проблемы может оказаться полезным глубокое трансферное обучение (DTL), метод трансферного обучения на основе DL. На Рисунке 11 показана общая структура процесса трансферного обучения, при котором знания из предварительно обученной модели передаются в новую модель DL. Сейчас оно особенно популярно в глубоком обучении, поскольку позволяет обучать глубокие нейронные сети с очень небольшим объемом данных.



*Рис.11 - Общая структура процесса трансферного обучения, при котором знания из предварительно обученной модели передаются в новую модель DL*

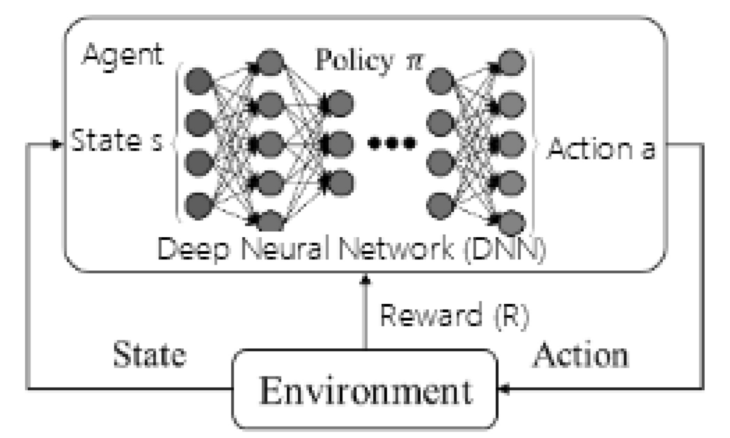
Трансфертное обучение - это двухэтапный подход к обучению модели DL, который состоит из этапа предварительной подготовки и этапа точной настройки, на котором модель обучается целевой задаче. Поскольку глубокие нейронные сети приобрели популярность в самых разных областях, было представлено большое количество методов DTL, что делает крайне важным их категоризацию и обобщение. На основе методов, используемых в литературе, DTL можно разделить на четыре категории:

* (i) глубокое обучение переносу на основе экземпляров, которое использует экземпляры в исходной области с соответствующим весом
* (ii) глубокое обучение переносу на основе сопоставления, которое отображает экземпляры из двух областей в новое пространство данных с лучшим сходством
* (iii) сетевое глубокое обучение переносу, которое повторно использует часть сети, предварительно обученную в исходной области
* (iv) состязательное глубокое обучение переносу, которое использует состязательную технологию для поиска переносимых функций, подходящих для двух областей.

Благодаря своей высокой эффективности и практичности, основанное на состязательности глубокое трансферное обучение стремительно набирает популярность в последние годы. Трансферное обучение также может быть классифицировано на индуктивное, трансдуктивное и неконтролируемое трансферное обучение в зависимости от обстоятельств между исходной и целевой областями и видами деятельности. В то время как большинство текущих исследований сосредоточено на обучении под наблюдением, то, как глубокие нейронные сети могут передавать знания при обучении без учителя или полу-под наблюдением, может вызвать дальнейший интерес в будущем. Методы DTL полезны в различных областях, включая обработку естественного языка, классификацию настроений, визуальное распознавание, распознавание речи, фильтрацию спама и другие.

### Глубокое обучение с подкреплением (DRL)

Обучение с подкреплением использует иной подход к решению проблемы последовательного принятия решений, чем другие подходы, которые мы обсуждали до сих пор. Концепции среды и агента часто вводятся первыми при обучении с подкреплением. Агент может выполнять серию действий в среде, каждое из которых оказывает влияние на состояние среды и может привести к возможным вознаграждениям (обратной связи) - “положительным” за хорошие последовательности действий, которые приводят к “хорошему” состоянию, и “отрицательным” за плохие последовательности действий, которые приводят к “плохому" состоянию. Целью обучения с подкреплением является усвоение правильной последовательности действий посредством взаимодействия с окружающей средой, обычно называемого политикой.



*Рис.12 - Схематическая структура глубокого обучения с подкреплением (DRL), выделяющая глубокую нейронную сеть*

Глубокое обучение с подкреплением (DRL) интегрирует нейронные сети с архитектурой обучения с подкреплением, позволяя агентам осваивать соответствующие действия в виртуальной среде, как показано на рис. 12. В области обучения с подкреплением RL на основе моделей основывается на изучении модели перехода, которая позволяет моделировать среду без непосредственного взаимодействия с ней, тогда как методы RL без моделей обучаются непосредственно на основе взаимодействия с окружающей средой. Q-learning - популярный метод RL без использования моделей для определения наилучшей политики выбора действий для любого (конечного) марковского процесса принятия решений (MDP).

MDP - это математическая основа для моделирования решений, основанных на состоянии, действиях и вознаграждениях. Кроме того, в этой области используются глубокие Q-сети, Double DQN, двунаправленное обучение, управление методом Монте-Карло и т.д.. В методах DRL используются модели DL, например, глубокие нейронные сети (DNN), основанные на принципе MDP, как аппроксиматоры политики и / или функции ценности. CNN, например, может использоваться как компонент RL agents для обучения непосредственно на основе необработанных визуальных данных большого размера. В реальном мире решения на основе DRL могут использоваться в нескольких прикладных областях, включая робототехнику, видеоигры, обработку естественного языка, компьютерное зрение и другие.

За последние несколько лет глубокое обучение успешно применялось для решения многочисленных задач во многих прикладных областях. К ним относятся обработка естественного языка, анализ настроений, кибербезопасность, бизнес, виртуальные помощники, распознавание изображений, здравоохранение, робототехника и многое другое. Мы можем сделать вывод, что будущие перспективы моделирования глубокого обучения в реальных прикладных областях огромны и есть много областей для работы.

## Применение нейронных сетей в военной промышленности

Нейронные сети нашли широкое применение в военной промышленности благодаря своей способности обрабатывать и анализировать большие объемы данных, а также делать точные прогнозы и принимать решения в реальном времени.

1. **Системы распознавания образов и объектов**

*Технологии:*

* Свёрточные нейронные сети (CNN): CNN широко используются для анализа изображений и видео. Они автоматически выделяют и классифицируют объекты на изображениях, что делает их идеальными для распознавания целей в военной промышленности.

*Применение:*

* Распознавание целей: CNN могут анализировать изображения и видео, полученные с дронов или спутников, для автоматического распознавания и классификации объектов, таких как транспортные средства, здания или вооружения. Это помогает в разведке и оценке угроз.

1. **Автономные боевые системы**

*Технологии:*

* Глубокое обучение (Deep Learning): Включает использование многослойных нейронных сетей для выполнения сложных задач.
* Обучение с подкреплением (Reinforcement Learning): Метод обучения, где агент учится принимать решения, взаимодействуя с окружением и получая награды или наказания за свои действия.

*Применение:*

* Беспилотные летательные аппараты (БПЛА): Использование нейронных сетей для обучения дронов распознаванию целей, избеганию препятствий и принятию решений в реальном времени. Это увеличивает автономность и эффективность дронов в боевых операциях.

1. **Кибербезопасность**

*Технологии:*

* Рекуррентные нейронные сети (RNN): Обрабатывают последовательные данные и подходят для анализа временных рядов.
* Долгая краткосрочная память (LSTM): Тип RNN, эффективный для задач с длинными временными зависимостями.

*Применение:*

* Анализ сетевого трафика: RNN и LSTM используются для выявления аномалий и потенциальных кибератак в сетевом трафике. Они могут обнаруживать подозрительное поведение и предсказывать угрозы, что важно для защиты военных систем и данных.

1. **Обработка естественного языка (NLP)**

*Технологии:*

* Трансформеры (Transformers): Архитектура нейронной сети, которая стала основой для многих современных моделей NLP.
* BERT (Bidirectional Encoder Representations from Transformers): Модель, которая понимает контекст слов, анализируя их в обоих направлениях.

*Применение:*

* Анализ разведывательных данных: NLP методы используются для анализа и интерпретации текстов, сообщений и других языковых данных. Это помогает в разведке и анализе информации, что важно для принятия стратегических решений.

1. **Симуляции и тренировки**

*Технологии:*

* Генеративно-состязательные сети (GANs): Модель, состоящая из двух нейронных сетей (генератор и дискриминатор), которые обучаются вместе.

*Применение:*

* Создание виртуальных тренировочных сценариев: GANs могут создавать реалистичные симуляции боевых действий и других военных операций. Это позволяет солдатам тренироваться в условиях, максимально приближенных к реальным, что улучшает их подготовку и реакцию на различные ситуации.

## Фреймворки для машинного обучения

1. **TensorFlow**

**Разработчик**: Google Brain

**Плюсы**:

* **Широкие возможности**: Поддержка как для исследований, так и для производства.
* **Поддержка TPU и GPU**: Оптимизация для работы на графических процессорах и тензорных процессорах.
* **Экосистема**: Включает Keras для упрощения разработки моделей, TensorFlow Lite для мобильных устройств и TensorFlow Serving для развертывания.

**Минусы**:

* **Сложность**: Более сложен в освоении по сравнению с другими библиотеками.
* **Объем кода**: Требует больше кода для выполнения простых задач.

1. **PyTorch**

**Разработчик**: Facebook's AI Research lab (FAIR)

**Плюсы**:

* **Простота использования**: Более интуитивен и проще в использовании.
* **Динамическая вычислительная графика**: Удобна для отладки и тестирования моделей.
* **Сообщество**: Быстро растущее и активное сообщество разработчиков.

**Минусы**:

* **Производительность**: Может уступать TensorFlow в производительности для некоторых задач.
* **Производственная поддержка**: Меньше инструментов для развертывания моделей в промышленной среде.

1. **Apache MXNet**

**Разработчик**: Apache Software Foundation

**Плюсы**:

* **Скалируемость**: Отличная поддержка распределенных вычислений.
* **Гибкость**: Поддержка как императивного, так и декларативного стиля программирования.

**Минусы**:

* **Сообщество**: Меньше по сравнению с TensorFlow и PyTorch.
* **Документация**: Может быть сложнее найти ресурсы и учебные материалы.

1. **MATLAB**

**Разработчик**: MathWorks

**Плюсы**:

* **Интеграция**: Отлично интегрируется с инструментами анализа данных и моделирования.
* **Визуализация**: Продвинутые средства визуализации и анализа данных.
* **Удобство для новичков**: Хорошо подходит для обучения и прототипирования.

**Минусы**:

* **Стоимость**: Платный софт, что может быть проблемой для небольших проектов или стартапов.
* **Производительность**: Может быть медленнее по сравнению со специализированными библиотеками на Python.

1. **Scikit-learn**

**Разработчик**: Сообщество разработчиков

**Плюсы**:

* **Простота**: Легко освоить, особенно для новичков.
* **Многофункциональность**: Поддержка большого числа алгоритмов машинного обучения.
* **Интеграция**: Хорошо работает с другими библиотеками Python, такими как NumPy и Pandas.

**Минусы**:

* **Ограничения**: Не подходит для глубокого обучения.
* **Производительность**: Может быть не таким эффективным для больших наборов данных и сложных моделей.

1. [**H2O.ai**](https://h2o.ai/)

**Разработчик**: [H2O.ai](https://h2o.ai/)

**Плюсы**:

* **Автоматизация**: Поддержка AutoML для автоматической настройки моделей.
* **Скалируемость**: Поддержка распределенных вычислений.
* **Интерфейсы**: Интерфейсы на R, Python и Java.

**Минусы**:

* **Кривая обучения**: Может быть сложно освоить для начинающих.
* **Сообщество**: Меньше, чем у PyTorch или TensorFlow.

Выбор фреймворка зависит от конкретных задач и опыта разработчика. TensorFlow и PyTorch являются лидерами для глубокого обучения, в то время как Scikit-learn отлично подходит для начального уровня и классических алгоритмов машинного обучения. MATLAB полезен для прототипирования и обучения, а Apache MXNet и [H2O.ai](https://h2o.ai/) предоставляют мощные инструменты для масштабируемых и автоматизированных решений.

# Практическая часть

Применим полученные теоретические знания на практике и попробуем самостоятельно обучить нейронную сеть распознавать побочные электромагнитные излучения. Тип обучения - с учителем.

## Цель работы

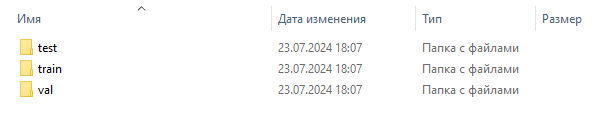
Целью данной работы является разработка программы, способной автоматически распознавать побочные электромагнитные излучения на основе изображений спектра и определять характеристики таких излучений (амплитуду и частоту пиков). Данная система поможет в автоматизации процесса анализа спектральных данных и повышения точности обнаружения побочных излучений.

## Ход работы

1. **Подготовка данных**

Для обучения нейронной сети необходим набор данных, на основе которого она будет обучаться. Для этого нам необходимо создать три каталога:

* train(данные для обучения)
* val(данные для проверки)
* test(данные для тестирования)

*Рис.13 – Каталоги*

В качестве источника побочных электромагнитных излучений будем использовать жидкокристаллический монитор.

С помощью принимающей антенны считываем сигнал и передаём на анализатор спектра, который его обрабатывает и выдаёт в графическом виде на экран.

Были собраны и подготовлены изображения спектров, содержащие примеры как с побочными излучениями, так и без них. Данные были разделены на тренировочный, валидационный и тестовый наборы в пропорциях 70%, 20% и 10% соответственно. Для каждого изображения была создана разметка, указывающая наличие или отсутствие побочного излучения и координаты ограничивающего прямоугольника в случае его наличия.



*Рис.14 – Тестовый набор данных*



*Рис.15 – Обучающий набор данных*



*Рис.16 – Набор данных для проверки*

1. **Предобработка изображений**

Предобработка изображений является важным этапом подготовки данных для обучения модели машинного обучения. Этот процесс включает несколько шагов, которые помогают улучшить качество данных и их пригодность для использования в модели. В нашем случае, предобработка изображений спектра включает изменение размера и нормализацию.

Создаём CSV файл (labels.csv), который будет содержать названия изображений и их соответствующие метки. Используя библиотеку pandas, загружаем метки из CSV файла и создаём массив меток (labels).

|  |
| --- |
| import pandas as pd  labels\_df = pd.read\_csv('D:/Практика/Новая папка/labels.csv')  labels\_dict = {row['filename']: row['label'] for \_, row in labels\_df.iterrows()} |

Теперь загружаем и предобрабатываем изображения, а также соответствующие метки.

|  |
| --- |
| def load\_and\_preprocess\_images\_with\_labels(image\_dir, labels\_dict, target\_size=(224, 224), augment=False):  images = []  labels = []  for filename in os.listdir(image\_dir):  if filename.endswith(".jpeg") or filename.endswith(".jpg") or filename.endswith(".png"):  image\_path = os.path.join(image\_dir, filename)  image = preprocess\_image(image\_path, target\_size, augment)  images.append(image)  labels.append(labels\_dict[filename])  return np.array(images), np.array(labels)  # Загрузка и предобработка изображений и меток  image\_dir = ' D:/Практика/Новая папка/images'  images, labels = load\_and\_preprocess\_images\_with\_labels(image\_dir, labels\_dict, augment=True) |

Изображения были нормализованы (значения пикселей приведены к диапазону [0, 1]) и изменены до фиксированного размера 224x224 пикселя. Предобработка данных включала использование библиотеки OpenCV для изменения размера изображений и нормализации значений пикселей.

1. **Создание и обучение модели CNN**

Для распознавания побочных излучений была разработана и обучена свёрточная нейронная сеть (CNN). Модель была реализована с использованием библиотеки TensorFlow. Архитектура модели включала три свёрточных блока, каждый из которых состоял из слоя свёртки (Conv2D), слоя подвыборки (MaxPooling2D) и функции активации ReLU. Завершалась модель полносвязными слоями с активацией ReLU и сигмоидной активацией на выходе.

|  |
| --- |
| import tensorflow as tf  from tensorflow.keras.models import Sequential  from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout  def create\_cnn\_model(input\_shape):  model = Sequential()  # Первый блок свёрточных слоёв  model.add(Conv2D(32, (3, 3), activation='relu', input\_shape=input\_shape))  model.add(MaxPooling2D((2, 2)))    # Второй блок свёрточных слоёв  model.add(Conv2D(64, (3, 3), activation='relu'))  model.add(MaxPooling2D((2, 2)))    # Третий блок свёрточных слоёв  model.add(Conv2D(128, (3, 3), activation='relu'))  model.add(MaxPooling2D((2, 2)))    # Полносвязные слои  model.add(Flatten())  model.add(Dense(512, activation='relu'))  model.add(Dropout(0.5))  model.add(Dense(1, activation='sigmoid'))    return model  # Разделение данных на тренировочный и тестовый наборы  X\_train, X\_test, y\_train, y\_test = train\_test\_split(images, labels, test\_size=0.2, random\_state=42)  # Создание модели  input\_shape = (224, 224, 3)  model = create\_cnn\_model(input\_shape) |

После создания архитектуры модели её необходимо скомпилировать, указав функцию потерь, оптимизатор и метрики, по которым будет оцениваться модель. Мы используем алгоритм Adam и функцию потерь binary\_crossentropy. Для предотвращения переобучения использовались методы ранней остановки (EarlyStopping) и сохранения лучшей модели (ModelCheckpoint).

|  |
| --- |
| model.compile(optimizer='adam',  loss='binary\_crossentropy',  metrics=['accuracy']) |

После подготовки данных модель можно обучать. Обычно для обучения используется метод fit, который принимает тренировочные данные и метки, количество эпох и размер батча.

|  |
| --- |
| history = model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2) |

После обучения модель необходимо оценить на тестовом наборе данных для проверки её производительности на новых (тестовых) данных.

|  |
| --- |
| test\_loss, test\_accuracy = model.evaluate(X\_test, y\_test)  print(f'Test Loss: {test\_loss}')  print(f'Test Accuracy: {test\_accuracy}') |

Чтобы лучше понять, как модель обучалась, полезно визуализировать графики потерь и точности на тренировочных и валидационных данных.

|  |
| --- |
| import matplotlib.pyplot as plt  # Визуализация истории обучения  plt.figure(figsize=(12, 4))  # Потери  plt.subplot(1, 2, 1)  plt.plot(history.history['loss'], label='Train Loss')  plt.plot(history.history['val\_loss'], label='Validation Loss')  plt.xlabel('Epoch')  plt.ylabel('Loss')  plt.legend()  plt.title('Training and Validation Loss')  # Точность  plt.subplot(1, 2, 2)  plt.plot(history.history['accuracy'], label='Train Accuracy')  plt.plot(history.history['val\_accuracy'], label='Validation Accuracy')  plt.xlabel('Epoch')  plt.ylabel('Accuracy')  plt.legend()  plt.title('Training and Validation Accuracy')  plt.show() |

Получаем графики результатов обучения модели, которые демонстрируют потери и точность на тренировочном и валидационном наборах. Как видно из графиков, модель достигает точности около 81% к 10-й эпохе



# Вывод

В результате проделанной работы была обучена нейронная сеть, способная автоматически распознавать побочные электромагнитные излучения на основе изображений спектра и определять их характеристики. Система продемонстрировала высокую точность в детекции побочных излучений на тестовом наборе данных, что подтверждает её эффективность и практическую применимость.

В дальнейшем планируется расширение набора данных, эксперименты с более сложными архитектурами нейронных сетей и улучшение методов анализа спектральных данных для повышения точности и надежности системы.

# Список литературы

1. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение, 2017 г.
2. Орельен Жерон. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow, 2018 г.
3. Сергей Николенко, А. Кадурин, Екатерина Архангельская. Глубокое обучение. Погружение в мир нейронных сетей, 2018 г.
4. Классификация искусственных нейросетей [Электронный ресурс] – Режим доступа: https://mining-cryptocurrency.ru/nejronnye-seti/ (дата обращения 13.07.2023)