**Федеральное государственное бюджетное образовательное учреждение высшего образования «Ярославский государственный университет им. П.Г. Демидова»**

Факультет: «Информатики и вычислительной техники»

Кафедра: «Компьютерных сетей»

КУРСОВАЯ РАБОТА

Тема: «Разработка классифицирующей нейронной сети для выявления пациентов, больных COVID-19»

Студент: Цыкарев Григорий Алексеевич

Группа: ИВТ-31БО

Научный руководитель:  
Алёшин Сергей Владимирович

**Ярославль – 2023г.**

Оглавление

[ВВЕДЕНИЕ 3](#_Toc148358944)

[ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc148358945)

[Историческая справка 5](#_Toc148358946)

[Области применения глубоких нейронных сетей 7](#_Toc148358947)

[Машинного обучение и глубокого обучение 8](#_Toc148358948)

[Задача T 9](#_Toc148358949)

[Мера качества P 10](#_Toc148358950)

[Опыт E 11](#_Toc148358951)

[Свёрточные нейронные сети 11](#_Toc148358952)

[Свёрточный слой 12](#_Toc148358953)

[Подвыборочный слой 12](#_Toc148358954)

[Полносвязный слой 13](#_Toc148358955)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 14](#_Toc148358956)

[Программные средства 14](#_Toc148358957)

[Реализация 14](#_Toc148358958)

[ЗАКЛЮЧЕНИЕ 17](#_Toc148358959)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 18](#_Toc148358960)

[ПРИЛОЖЕНИЯ 19](#_Toc148358961)

# ВВЕДЕНИЕ

С развитием компьютерной техники и интернет-технологий человека стало окружать огромное количество информации, к которой, помимо текста, также стали относиться изображения, видео, музыка. Бурное развитие технологий глубоко внедрилось и в рабочий процесс, сложно представить

современное рабочее место без компьютера или телефона с доступом в интернет. Данная тенденция в том числе относится и к работникам здравоохранения, которым приходится работать с большим объёмом личной информации пациентов. В связи с событиями последних лет стала актуальна проблема выявления пациентов, болеющих COVID-19, а также выявление степени тяжести заболевания. Помимо КТ для решения этих проблем применяется рентгенография, которая, хоть и не такая точная, но зато более доступная и мобильная. При большом количестве пациентов может помочь нейронная сеть, решающая задачу классификации рентгеновских снимков. Пускай она и не может заменить живого специалиста, но может значительно облегчить его работу.

Задача классификации - задача, в которой имеется некоторое множество объектов, разделённых по определённым признакам на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется выборкой. Классовая принадлежность остальных объектов неизвестна.

Классифицировать объект – значит, указать номер (или наименование) класса, к которому относится данный объект.

Классификация объекта – номер или наименование класса, выдаваемый алгоритмом классификации в результате его применения к данному конкретному объекту.

Актуальность работы заключается в том, что создание глубоких нейронных сетей является одной из приоритетных отраслей современных информационных технологий. Особенно это заметно в последнее время с появлением таких нейронных сетей, как ChatGPT, MidJourney, Stable Diffusion, Perplexity AI и так далее. Классифицирующие нейронные сети активно используются во множестве направлений, таких как компьютерное зрение, диагностирование заболеваний на основе базы медицинских изображений, фильтрация социальных сетей, анализ поведения пользователей для целевого маркетинга и так далее.

Целью выполнения курсовой работы является изучение и создание свёрточной нейронной сети, классифицирующей рентгеновские снимки грудных клеток для выявления COVID-19.

Цель обусловила постановку и последовательное решение в работе следующих задач:

1. Изучение принципов работы машинного обучения и глубоких нейросетей
2. Выбор датасета для решения поставленной задачи
3. Выбор инструментов для реализации нейросети
4. Реализация нейросети
5. Обучение нейросети

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## Историческая справка

Искусственный интеллект - это совокупность различных технологических и научных решений и методов, позволяющих разрабатывать программы, действующие подобно человеческому интеллекту.

Машинное обучение - это набор методов в области искусственного интеллекта, предоставляющих алгоритмы для создания машин, обучающихся на собственном опыте.

Глубокое обучение - это направление вобласти искусственного интеллекта (и, в частности, машинного обучения), которое основано на применении нейронных сетей, а также включает в себя исследование и разработку.

Основные этапы исторического развития машинного обучения:

История машинного обучения началась в 40-50-х годах 20 века. Первая модель компьютера с искусственным интеллектом была создана в рамках сверхсекретного американского проекта ЭНИАК в 1946 году.

В 1950 году Алан Тьюринг разработал собственную методику тестирования искусственного интеллекта. Тест оценивал интеллект компьютера и определял способность машины мыслить подобно человеку.

Нейросеть впервые была воссоздана в проекте «Персептрон» в 1958 году. Ее автором является американский ученый

Фрэнк Розенблатт, реализовавший свою идею в виде нейрокомпьютера «Марк-1».

В тот же год Марвин Минский создал первую вычислительную машину SNARC на основе нейросети, которая была создана годом раньше Фрэнком Розенблаттом.

Артур Самуэль в 1959 году впервые ввёл термин "машинное обучение", создав на примере игры в шашки программу, способную к самообучению.

В 1960-х года проводились исследования, связанные с применением компьютерного зрения и использованием простых шаблонов распознавания.

Далее машинное обучение продолжало своё развитие и в 1997 году был создан компьютер Deep Blue, который сумел обыграть в шахматы чемпиона мира Гарри Каспарова.

В 21 веке появилось такое понятие, как глубокое обучение, а также было разработано много алгоритмов для распознавания содержимого на фотографиях и видеороликах.

В 2016 году появилась программа AlphaGo, разработанная компанией DeepMind от Google, которая смогла победить чемпиона мира по игре в го Ли Седоля.

С помощью машинного обучения искусственный интеллект может выполнять такие задачи, как:

1. анализ данных
2. запоминание информации
3. составление различных прогнозов
4. воспроизведение готовых моделей
5. выбор наиболее подходящих вариантов из предложенных

В настоящее время машинное обучение постоянно развивается, позволяя создавать компьютеры, которые могут обучаться. Это поможет эффективно использовать компьютеры при решении рутинных и затратных операций.

## Области применения глубоких нейронных сетей

Глубокие нейронные сети нашли широкое применение в различных областях, таких как:

Компьютерное зрение: глубокие нейронные сети могут использоваться для распознавания объектов на изображениях, классификации изображений, обнаружения объектов и других задач.

В качестве распознаваемых образов на изображениях могут выступать рукописный или печатный текст, банковские карты, подписи на официальных документах, люди, животные, автомобили и так далее.

Классифицирующие нейронные сети используются для сортировки, например, рентгеновских снимком. Нейронная сеть обучается отличать снимки больных людей от снимков здоровых людей.

Это особенно актуально в последнее время в связи с пандемией COVID-19, из-за чего врачам приходится просматривать множество снимков, определяя больных пациентов.

Распознавание речи: глубокие нейронные сети могут использоваться для распознавания речи, синтеза речи, автоперевода.

В качестве примера можно привести приложение skype translator, которое переводит речь собеседников в реальном времени.

Оно использует глубокие нейронные сети для распознования речи и технологию статического машинного перевода Microsoft Translator.

На данный момент skype translator поддерживает 10 языков: английский, французский, немецкий, китайский (мандаринский диалект),

итальянский, испанский, португальский, арабский, японский и русский.

Обработка естественного языка: глубокие нейронные сети могут использоваться для анализа текста, машинного перевода, генерации текста и других задач, связанных с обработкой языка.

В качестве примера можно привести Google Neural Machine Translation (GNMT), созданную для повышения точности и скорости перевода.

В основе GNMT лежит нейросеть Artificial Neural Network (ANN), которая обучется на миллионах примеров переводов. ANN позволяет выполнять zero-shot перевод,

то есть переводить с одного языка на другой, не имея явные примеры для этой пары конкретных языков в обучающей выборке.

Биоинформатика: глубокие нейронные сети могут использоваться для анализа геномных данных, прогнозирования структуры белков и других задач, связанных с биологическими данными

Один из самых распростронённых примеров применения глубоких нейронных сетей в биоинформатике - это анализ последовательностей ДНК и РНК, с помощью которого можно предсказывать

функциональные свойства генов, идентифицировать гены, связанные с определёнными болезнями и выявлять мутации в геноме.

## Машинного обучение и глубокого обучение

**Глубокое обучение** – частный случай машинного обучения.

**Алгоритм машинного обучения** – это алгоритм, способный обучаться на данных. Говорят, что компьютерная программа обучается на опыте E относительно некоторого класса задач T и меры качества P, если качество на задачах из T, измеренное с помощью P, возрастает с ростом опыта E

### Задача T

Задача машинного обучения – это то, как система обрабатывает какой-либо пример. Примером, как правило, является вектор признаков, полученных в результате измерения какого-либо объекта или события, которые система должна научиться обрабатывать.

С помощью машинного обучения можно решить следующие задачи:

**Классификация.**

В данной задаче нейронная сеть должна определить, к какому категории относится тот или иной объект. Как правило, для решения задачи от алгоритма требуется породить функцию для распределения примеров по категориям:

F : ℝ^n ⟶ {1, …, k}

Если y = f(x), то модель относит входной пример, заданный вектором x, к категории с числовым кодом y.

**Регрессия.**

В данной задаче нейросеть должна на основе входных данных предсказать числовое значение. Для её решения требуется породить функцию

F : ℝ^n⟶ ℝ

От классификации задача регрессии отличается форматом вывода.

**Транскрипция.**

В данной задаче программе машинного обучению нужно на основе не структурированно представленных данных создать их дискретное текстовое представление. Пример – система распознавания речи, когда на вход поступает аудио-сигнал, а на выходе получается текст с распознанной речью.

**Машинный перевод.**

Преобразование последовательности символов одного языка в последовательность на другом языке. Языки не обязательно могут быть естественными, однако для них задача решается чаще всего.

**Обнаружение аномалий.**

В данной задаче нейронной сети на вход поступают примеры, некоторые из которых помечены как нетипичные. В дальнейшем такие программы могут применяются для выявления злоумышленников, пытающихся украсть деньги с кредитных карт.

**Синтез и выборка.**

В данной задаче программа машинного обучения должна на основе предоставленного датасета создать похожие на примеры из выборки объекты. Данная задача возникает, когда нужно создать большое количество однотипной информации, на создание которой либо не хватает ресурсов, либо времени.

### Мера качества P

Чтобы оценить, насколько хорошо программа решает поставленную задачу, нужно иметь количественное представление её успешности. Обычно данная мера P специфична для каждой задачи T.

Для задач классификации и транскрипции мерой успеха является количество правильно определённых примеров. Эквивалентный результат можно получить, если измерять частоту ошибок.

Иногда для некоторых задач бывает сложно подобрать меру качества. Особенно это относится к нейросетям, создающим изображения или музыку. В таких случаях у каждого пользователя своё представление о «качественном» результате, из-за чего достаточно сложно подобрать адекватную меру.

### Опыт E

Алгоритмы машинного обучения можно разделить на два больших класса: обучение с учителем и без. Первый класс подразумевает, что каждый пример датасета снабжён меткой, относящий пример к той или иной категории. Алгоритм должен проанализировать данные и научиться определять класс примера с помощью данных меток. Такие датасеты активно используются для задач классификации.

При обучении без учителя алгоритм должен на основе того, правильно или неправильно он решил поставленную задачу, сам научиться её решать. Такие наборы используются для задачи кластеризации, когда алгоритм распределяет данные по отдельным признакам по кластерам.

Также существуют и другие виды обучения, такие как обучение с подкреплением, когда нейронная сеть обучается в процессе взаимодействия с окружающей средой.

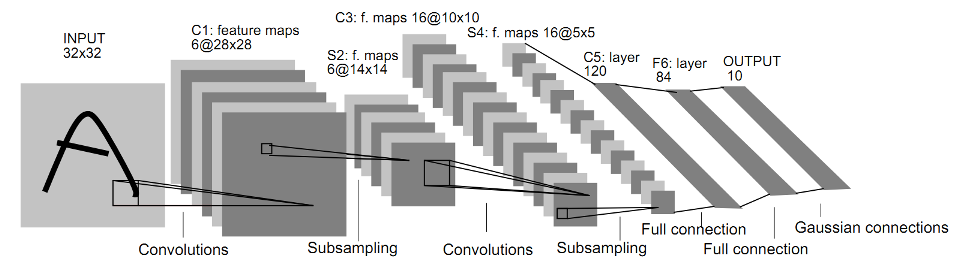
## Свёрточные нейронные сети

**Свёрточная 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.

Свое название свёрточная сеть получила по названию операции – свёртка.

На данный момент свёрточная нейронная сеть и ее модификации считаются лучшими по точности и скорости алгоритмами нахождения объектов на сцене. Начиная с 2012 года, нейросети занимают первые места на известном международном конкурсе по распознаванию образов ImageNet.

Свёрточная нейронная сеть состоит из нескольких видов слоёв: свёрточные слои, субдискретизирующие слои и слои персептрона.

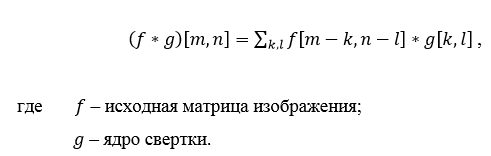


Первые два вида слоёв формируют входной вектор признаков для многослойного персептрона.

### Свёрточный слой

Свёрточный слой представляет из себя набор карт (карты признаков), у каждой карты есть синаптическое ядро. Количество карт зависит от задачи, если взять большое количество, то увеличится качество распознавания, но увеличится вычислительная сложность и наоборот. Ядро – это фильтр, который проходит по всей карте и находит определённые признаки. Размер ядра подбирается таким образом, чтобы размер карт свёрточного слоя был чётным.

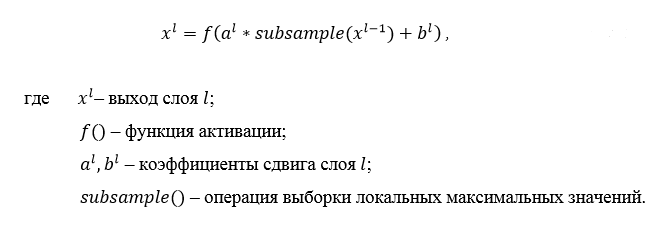
Изначально значения каждой карты свёрточного слоя равны 0. Значения весов ядер задаются случайным образом в области от -0.5 до 0.5. Ядро скользит по предыдущей карте и производит операцию свертка, которая часто используется для обработки изображений. Формула:



### Подвыборочный слой

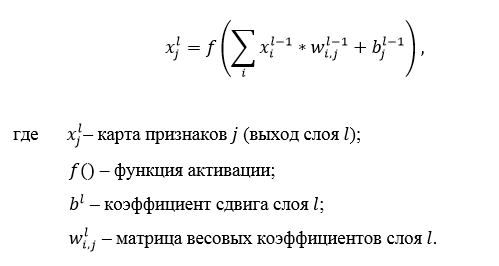
Подвыборочный слой также имеет карты, но их количество совпадает с предыдущим (свёрточным) слоем. Цель слоя – уменьшение размерности карт предыдущего слоя. Если на предыдущей операции свертки уже были выявлены некоторые признаки, то для дальнейшей обработки настолько подробное изображение уже не нужно, и оно уплотняется до менее подробного.

Формально слой может быть описан формулой:



### Полносвязный слой

Последний слой – это слой персептрона. Цель слоя – классификация. Нейроны каждой карты предыдущего подвыборочного слоя связаны с одним нейроном скрытого слоя. Таким образом число нейронов скрытого слоя равно числу карт подвыборочного слоя. Вычисление значений нейрона можно описать формулой:



# ПРАКТИЧЕСКАЯ ЧАСТЬ

## Программные средства

В качестве языка реализации был выбран python, так как для него написаны самые популярные библиотеки для создания нейронных сетей и работы с большими данными. Для решения данной задачи были выбраны библиотеки Keras и Pandas.

Keras — это API глубокого обучения, работающий на платформе машинного обучения TensorFlow. Keras спроектирован таким образом, чтобы быть компактным, модульным и расширяемым. При этом он достаточно прост в освоении, что позволяет больше сконцентрироваться на решении задачи, а не на программной реализации. Keras очень популярен, поэтому для него существует достаточно много как англоязычных, так и русскоязычных учебных материалов.

Pandas – это высокоуровневая основанная на NumPy библиотека с открытым исходным кодом для обработки, анализа и манипулирования большими данными. Данная библиотека очень удобна для работы со структурированной информацией, такой как датасеты для обучения нейронных сетей.

Также для реализации был использован свёрточный слой нейросети ResNetV2, основанной на семействе архитектур Inception. Данная нейросеть обучена на более чем миллионе изображений от базы данных ImageNet, поэтому она отлично подойдёт для решения поставленной задачи.

## Реализация

В начале импортируем все необходимые библиотеки. Затем создаём переменную train\_df типа DataFrame с помощью функции read\_csv, из колонок оставляем только путь до изображения и метку. Эта переменная предназначена для работы с изображениями для обучения. То же самое делаем с переменной test\_df, которая будет нужна для тестирования сети.

Создаём переменные train\_path и test\_path, хранящие путь до тренировочных и тестовых изображений.

После этого делаем выборку из пяти тысяч уникальных записей с положительными метками и такую же выборку с отрицательными метками. Делим тестовую выборку на валидационную и тестовую в соотношении один к девяти.

Затем создаём переменную target\_size для задания размера одного изображения и переменную batch\_size для задания размера пакета наблюдения.

С помощью функции ImageDataGenerator производим аугментацию данных для улучшения качества обучения модели. В качестве функции препроцессинга выберем функцию нейросети ResNetV2. То же самое делаем и для тестовой выборки.

Затем с помощью функции flow\_from\_dataframe считываем изображения и их метки для дальнейшего обучения, в качестве параметров задаём вышеобъявленные переменные target\_size и batch\_size. Делаем это для тренировочной, валидационной и тестовой выборок.

Теперь переходим к проектированию модели нейронной сети. Берём свёрточный слой нейросети ResNetV2 и записываем его в переменную base\_model. Затем создаём последовательную модель, состоящую из следующих слоёв:

1. свёрточный слой – base\_model
2. слой пуллинга
3. полносвязный слой, состоящий из ста двадцати восьми нейронов и функции активации Rectified Linear Unit (ReLU)
4. слой батч-нормализации для нормализации активаций
5. слой Dropout для уменьшения переобучения
6. выходной слой с функцией активации sigmoid

После проектировки компилируем получившуюся модель. В качестве функции потерь указываем Binary cross entropy, так как у нас всего два класса изображений, оптимизатор adam и следующие метрики: accuracy для определения верно классифицированных объектов по отношению ко всем объектам, precision для оценки правильности определения положительных примеров, recall для измерения доли верно классифицированных положительных объектов относительно всех положительных объектов, AUC для аппроксимации кривых precision recall, TruePositives для вычисления правильно определённых положительных результатов, TrueFalses для определения правильно определённых негативных результатов, FalsePositives и FalseNegatives для определения неправильно определённых положительных и отрицательных результатов соответственно.

Также для задания определённого поведения обучения будем использовать коллбэки: ModelCheckpoint для сохранения весов во время обучения, EarlyStopping для автоматической остановки обучения, если производительность модели на валидационных данных не улучшается с течением времени и ReduceLROnPlateau для уменьшения скорости обучения (learning rate) при отстутствии улучшения модели на валидационных данных.

Теперь обучаем модель с помощью функции fit, в качестве параметров указваем вышеописанные переменные для обучения и валидации, переменную с количеством эпох и список коллбэков.

Результат обучения:

# ЗАКЛЮЧЕНИЕ

В заключение подведём основной итог курсовой работы. Для создания классифицирующей нейронной сети были рассмотрены основы машинного обучения, основные задачи машинного обучения и принципы работы свёрточных нейронных сетей. Для программной реализации в качестве свёрточного слоя был взят слой хорошо обученной нейросети, а для решения конкретной задачи были добавлены собственные слои, также была произведена тонкая настройка коллбэков для оптимизации обучения. Поставленные цели достигнуты.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

# ПРИЛОЖЕНИЯ