МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Тюменский индустриальный университет»

Институт сервиса и отраслевого управления

Кафедра «Бизнес-информатики и математики»

**КУРСОВОЙ ПРОЕКТ**

**Дисциплина:** «Объектно-ориентированный анализ и программирование»

**Тема:** «Разработка нейросети на основе пакетов Python»

**Выполнили:**студент 1 курса направление 38.03.05 «Бизнес – информатика» группа БИб-19-1

Лисовский А.И.

студент 1 курса направление 38.03.05 «Бизнес – информатика» группа БИб-19-1

Михеенков А.С.

**Проверил:**

?

Тюмень – 2020

Оглавление

[1. Теоретические основы объектно-ориентированного программирования и нейронных сетей. 4](#_Toc60053464)

[1.1. История развития и основные понятия нейронных сетей и ООП. 4](#_Toc60053465)

[1.2. ООП. 4](#_Toc60053466)

[1.3. Нейронные сети. 5](#_Toc60053467)

[1.4. Язык программирования python. 5](#_Toc60053468)

[2. Практическая часть 7](#_Toc60053469)

# Введение

В наше время повсеместно начинают использоваться нейронные сети. А язык программирования Python является одним из самых распространённых. Неудивительно что существуют пакеты для работы с нейронными сетями на этом языке. Существует огромное количество способов их применения, в этой работе мы попробуем сделать модель для предсказания курса акций.

# Теоретические основы объектно-ориентированного программирования и нейронных сетей

## История развития и основные понятия нейронных сетей и ООП

Идея объектно-ориентированного программирования сформировалась 70-80 годы XX века. Основными понятиями ООП являются класс и объект (экземпляр класса).

Впервые понятие Нейронной сети было сформулировано У. Маккалоком и У. Питтсом. Их основной сутью является имитация нейронных связей мозга животных. Основными понятиями являются нейрон, связь и функция активации. Причём в этой области ООП может раскрыться во всей красе. Можно создать класс нейронов и нейронной сети. Однако современные пакеты облегчают жизнь разработчикам и скрывают многие тонкости, давая программистам удобный интерфейс создания и контроля нейронной сети.

## Основы объектно-ориентированного програмирования

Существует четыре основных принципа ООП:

* Абстракция, выделение в объекте изучения основных признаков и свойств
* Инкапсуляция, разделение кода на части и ограничение доступа к свойствам и полям
* Наследование, создание дочерних классов, наследующих характеристики класса родителя
* Полиморфизм, создание абстрактных классов, методы которых реализуются в дочерних классах

Достоинства ООП:

* Создание понятий близких к предметной области, по сути, можно расширить язык программирования своими понятиями
* Возможность создания сложных и комплексных программ при помощи инкапсуляции
* Возможность использовать один код несколько раз, при помощи наследования
* Относительная лёгкость в модификации программы
* Возможность создания и использования своих библиотек классов

Недостатки ООП:

* Небольшое снижение быстродействия программы
* Порог вхождения увеличивается, ООП сложнее в понимании чем процедурное программирование
* Неквалифицированное применение ООП может привести к ухудшению программы

## Нейронные сети

Нейронные сети – это часть более общего направления, машинного обучения, задачей которого является создание моделей прогнозирования. Нейронные сети – это один из мощнейших инструментов данной области

Также более объёмные и сложные нейронные сети входят в понятие глубокого обучения, одно из самых востребованных направлений в наши дни. Нейронные сети состоят из двух элементов: нейроны и связь между ними. У нейрона есть своего рода «порты» входа и выхода информации. Нейрон получает по связям информацию от других нейронов, преобразует её с помощью функции активации и отправляет следующим нейронам. В нейронной сети нейроны расположены по слоям, где каждый нейрон из одного слоя связан с каждым нейроном следующего слоя. Также имеются входной и выходной слои служащие для взаимодействия с нейронной сетью. Остальные слои называются скрытыми и зачастую на диаграммах их заменяют одним «слоем». Это базовая конструкция нейронной сети. Существуют другие более сложные конструкции как, например свёрточные или рекурсивные сети. При проектировании нейронных сетей обычно сталкиваются с двумя проблемами: переобучение сети и недообучение сети. В результате переобучения модель слишком подстраивается под обучающую выборку, и в результате не способна эффективно работать с генеральным множеством. В случае с недообучением модель не способна выделить существенные признаки в обучающей выборке, это означает то, что она либо слишком проста, либо обучающей выборки недостаточно и её надо расширять или изменять.

## Язык программирования python

Язык программирования python – это объектно-ориентированный язык, в котором почти каждый элемент является своего рода частью этой концепции. Основным преимуществом данного языка является его читабельность и низкий порог вхождения, что позволяет выбирать его в качестве первого языка. Однако не существует идеального инструмента, за высокую скорость написания кода приходится платить производительностью. Python – интерпретируемый язык, а значит по своей скорости он в значительной степени уступает таким компилируемым языкам как C++ или C#. Однако Python можно расширить при помощи пакетов, написанных другими программистами, и на данный момент уже существует огромное количество пакетов для разных целей. Например, для написания 2D игр существует пакет pygame, для создания сайтов – Flask, Django, и для нейронных сетей – Keras, построенный как интерфейс-окружение для Tensorflow.

# Практическая часть

## Выбор пакета для работы с нейронными сетями

Существует много пакетов для работы с нейронными сетями на языке Python: Theano, TensorFlow, Keras, Lasange, Neon и другие. Для данного проекта мы выбрали пакет Keras так, как он представляет своего рода надстройку для TensorFlow и Theano. Таким образом он включает в себя возможности этих двух пакетов и удобный интерфейс для создания и тестирования нейронных сетей.

## Выбор темы и поиск информации

Для данного проекта мы хотим попытаться сделать модель для предсказания курса акций на некоторый промежуток времени. Мы навряд ли сможем найти готовые наборы информации, поэтому мы обратились к сайту Московской биржи для поиска информации. На данном сайте можно узнать результаты торгов для многих компаний. Поэтому была написана программа для сбора данной информации. В первую очередь мы получили коды фирм, чьи акции представлены на данном сайте. Затем мы создали программу, которая каждые полсекунды отправляет запрос на сайт Московской биржи и получает результаты торгов на сто дней от указанной даты. В конечном итоге один элемент набора данных представляет собой два массива в каждом из которых находится нормализованная разница в ценах на момент закрытия торгов. Общая длинна этих двух массивов равна 100. Первый массив будет подаваться нейронной сети на вход, а второй будет выступать в качестве результата, который мы хотим получить. В итоге был получен массив длинной в 2539 элементов, которые состоят из двух массивов, описанных выше. Мы прекрасно понимаем то, что курс акций зависит от многих факторов, которые мы берём в расчёт, и то, что данного набора информации недостаточно для задачи такой сложности. Однако, было бы интересно попробовать сделать модель высокой точности при таких ограничениях.

## Создание и обучение модели.

В результате многократных экспериментов, в процессе которых мы изменяли характеристики сети с целью найти наиболее оптимальный вариант, была получена нейронная сеть со следующей структурой: первый входной слой с 93 слоями, затем два скрытых слоя с 10 нейронами и гиперболическим тангенсом в качестве функции активации и выходной слой с 7 нейронами и логистической функцией в качестве функции активации. Изначально модель создаётся со случайными значениями весов, поэтому значение среднеквадратической ошибки сильно колеблется от 3 сотых до 5 сотых, что не является таким ужасным результатом, но и хорошим назвать его сложно. Но с помощью такой модели уже можно делать относительные предсказания. Значения среднеквадратичной ошибки во время тестирования и обучения не сильно отличается. Из этого мы можем сделать вывод, что основной проблемой является нехватка набора данных, их просто не хватает для полноценного обучения модели, и при этом задача довольно сложная, потому что даже при многократном обучении модель справляется сравнительно также при тестировании. Таким образом происходит недообучение.

# Заключение

В процессе выполнения данного проекта был изучен пакет Keras, используемый для создания нейронных сетей. А также была спроектирована модель для прогнозирования курса акций Московской биржи. Хотя данная модель не идеальна с её помощью можно делать относительные прогнозы акций. Было очень интересно искать информацию и обрабатывать её. Ведь сейчас самым сложным в машинном обучении является поиск и обработка информации для обучения модели.