Муниципальное общеобразовательное учреждение "Лицей №4 г.Дмитрова"

“Использование нейросетей для предсказания курса акций”

Автор:

Шевчук Кирилл Александрович

[kirill1shevchuk@gmail.com](mailto:kirill1shevchuk@gmail.com)

+79774705507

**Содержание**

[**Введение**](#_4bsp946gnadu) **3**

[Цель:](#_d06g2jtib1ww) 3

[Задачи:](#_55dnhj9efy45) 3

[**Работа над проектом**](#_sguqldtq864i) **4**

[Изучение рынка](#_mvjpkd26665g) 4

[Поиск базы данных](#_t10l5wgxh4ad) 4

[Работа с данными и их преобразование к нужному виду.](#_c0bm29lqzqhx) 5

[Создание модели нейросети](#_lulz1xiokah4) 6

[Немного о нейросетях](#_mdh06wlp00aq) 6

[Анализ результатов](#_tvlgxbafudlt) 9

[**Вывод**](#_b5o2adj6c9bh) **10**

[Возможные улучшения](#_kqfxbst626q1) 12

# Введение

Изначально я заинтересовался акциями, ценными бумагами и вообще инвестициями, а после поехал в лагерь и начал заниматься нейросетями. И появилась идея того, чтобы использовать недавно приобретенные навыки написания нейросетей для предсказания курса акций.

## Цель:

Написать нейросетевой алгоритм, который сможет предсказывать курс акции с точностью от 75% и вероятностью от 90%.

## Задачи:

1. Изучить конкурентов и прочие сервисы, которые предлагают предсказание курса чего-либо.
2. Найти базы данных с нужными мне данными.
3. Правильно изменить и подогнать данные, чтобы их поняла нейросеть.
4. Пишем модель обучения нейросети и смотрим то, как она обучалась.
5. Делаем красивые графики, чтобы показать предсказание.
6. Возможное развитие.
7. Результаты и вывод.

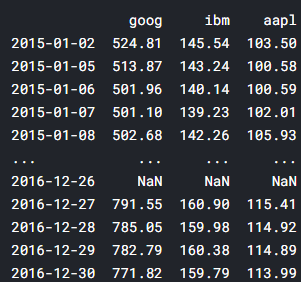
# Работа над проектом

## Изучение рынка

Начал я с того, что нашел несколько подобных проектов, а именно - [stocksneural.net](https://stocksneural.net/) и [iknowfirst.com](https://iknowfirst.com/). Они выполняют именно ту функцию о которой я и думал, а именно, предсказывают курс акций с помощью ML[[1]](#footnote-0). Думаю у них можно было бы позаимствовать идеи для возможно дальнейшего развития проекта

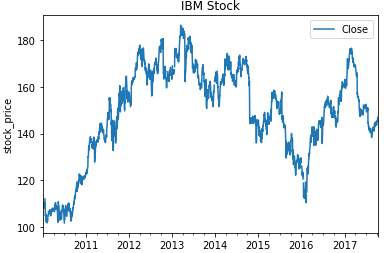
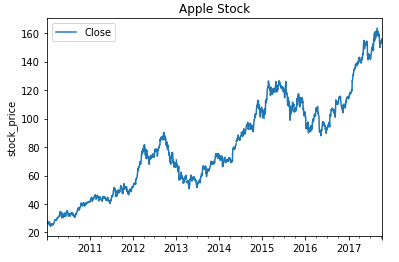
## Поиск базы данных

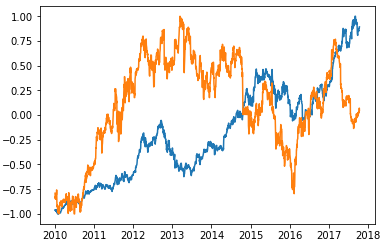
Одна из ключевых задач в написании любой нейронной сети для желаемой тебе задачи - найти большой объем данных нужных тебе и желательно, хоть немного структурированный, к счастью, брокеры и биржи собирают такие данные, а именно цену акции в начале, конце торгов, максимальную цену и дату. Мне повезло найти огромную базу данных по акциям компании Apple и IBM на временной линии с 2010 г по 2018 г. Я получил данные на площадке [Kaggle.com](https://www.kaggle.com/). Это площадка для соревнований среди людей, занимающихся ML, анализом данных, нейросетями и тд. Дальше я просто сделал таблицы чтобы посмотреть данные. <https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs>



## Работа с данными и их преобразование к нужному виду.

Обычно, когда мы пытаемся передать какие-то данные нейросети, их лучше всего преобразовать в более понятный вид, к примеру, есть такая задача для начинающих в ML, как задача с титаником, а именно, нужно предсказать, выживет ли человек на титанике, по нужным имеющимся данных в виде имени, пола, номера билета, палубе, где была каюта и тд. Не все данные там в виде цифр, а нейросеть может принимать только их, поэтому нужно сделать из непонятных букв, дат и тд более понятные цифры. Именно в этой задаче мне нужно было изменить дату и сделать из нее дни от 1 до n и цену я изменил в значения от -1.00 до 1.00. А после этого, преобразовать все данные в так называемые тензоры[[2]](#footnote-1). Грубо говоря, это как матрица из линейной алгебры. Для этого я использую библиотеку Numpy.





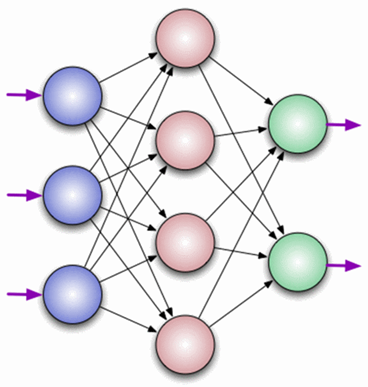
## Создание модели нейросети

### Немного о нейросетях

Для начала хочу немного рассказать о нейросетях и что это такое, чтобы немного ввести в курс дела. Что такое нейросеть и из чего она состоит?

Думаю из названия можно будет понять, что состоит она из нейронов, объединенных в какую-то сеть, но что такое нейрон и зачем нам это все?

**Нейрон** - это маленькая вычислительная единица или узел, она может принимать в себя какие-то значения, преобразовывать их путем каких-то математических вычислений и после отдавать данные другим нейронам или на выход. Рассмотрим на примере данной картинки:



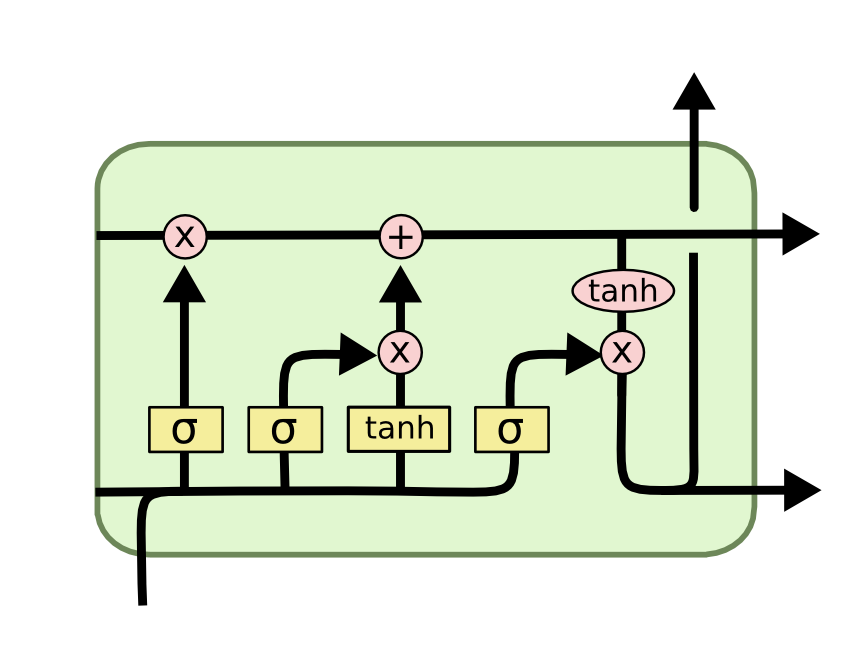
**Тут мы видим 9 нейронов**. **Синие** - те, куда подается информация(входные нейроны), **Красные** - вычислительные(скрытые нейроны), ну и **Зеленые** - выходные нейроны, которые выводят нам информацию. Все эти нейроны соединены **стрелками**, а именно синапсами.

Давайте рассмотрим на простом примере:

Возьмем точку в пространстве с 2-мя осями, X и Y. Эта точка двигается и каждую сотую секунды ее перемещают в какое-то место. Предположим, что это делает человек, а т.к. люди не способны на рандомные действия, их можно попробовать предугадать. Тогда мы заносим в нейросеть данные о ее местоположении в данный момент(1 синий - положение по X, 2 - положение по Y, 3 - номер измерения (или время)). После чего данные идут в скрытый слой(красные) и там с ними происходят какие-то вычисления, а именно домножение на веса синапсов(стрелочки) и умножение на какую либо функцию, после чего данные идут в выходной слой(зеленые) и мы получаем предсказанное значение, где будет находиться точка в следующий момент времени.

Скорее всего, первое предсказание будет очень плохое, потому что у нейросети еще не выставлены правильные веса синапсов. Что такое веса? Веса - это как будто бы то, на сколько важно то или иное действие. Предположим, что нейросеть заметила, что изменив специальным образом какой то из весов, предсказание вышло гораздо точнее, а значит, надо запомнить, что повышения веса данного синапса - улучшило результат.

Так что грубо говоря, **работа нейросети это постоянно ошибаться** и выдавать ответ отличный от предсказания, но каждый раз когда она выдает ответ чуть ближе к нужному, мы его запоминаем. И еще работа нейросети это постоянные математические сложения, вычитания, умножения и использования различных функций, вроде сигмоиды или гиперболического тангенса.

**И вот, данные для нейросети готовы**, самое время заняться самым интересным, а именно написать модель нашей нейронки. Я буду использовать нейросеть типа LSTM. Именно этот тип имеет краткосрочную и долгосрочную память, это как раз то, что нам нужно для данных, которые имеют временные отрезки. Вот как это выглядит: 

Есть набор данных, который входит в слой(у меня их 34), далее, они идут в кратковременную память(на рисунке - вся нижняя полоска) и там преобразуются с помощью разных математических функций(тангенс, сигмоида и тд) и часть данных уходит в долговременную память(вся верхняя полоска), которая почти не изменяется и эта долговременная память проходит через все слои и почти никак не меняется.

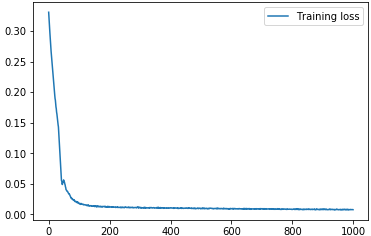
Через такие “ячейки” проходят тензоры, перемножаются, плюсуются и к ним применяются различные функции. Все это делается, для того чтобы потом понять, какие изменения для тензоры были наиболее эффективны и как это улучшило результат предсказания засчет показания весов[[3]](#footnote-2).

## Анализ результатов

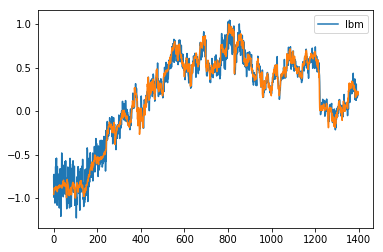
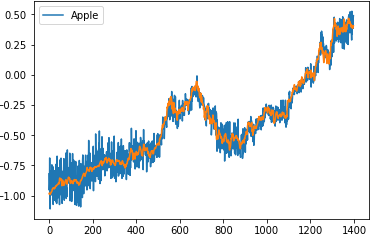
После запуска моей модели я решил поставить 100 эпох, каждая эпоха - запуск обучения нейросети и результат уже был довольно хорошим, а именно - примерно 2 процента ошибки.



После чего я решил добиться лучших результатов и просто увеличил количество эпох до 500, но ошибка уменьшилась всего до 1.6% процента ошибки. И я решил попробовать что-то новое, а именно - я добавил в свою модель функцию Dropout со значением 0,3. Данная функция делает так, что в рандомный момент, в моей нейросети отключится 30% нейронов. Зачем это надо? Затем, что есть такая проблема в обучении, как **переобучение**, а именно, проблема в том, что при большом количестве данных( как раз как у меня, учитывая около 2000 значений которые подаются в нейросеть) нейросеть просто начинает запоминать определенные участки и вместо предсказания - просто выдает забитый в памяти результат. Использование данной функции снизило ошибку до 1%, но я на этом не остановился и увеличил количество эпох до 1000 и это дало результат 0.7%.

.

Давайте теперь **рассмотрим график предсказания**.



У нас было 1400 дней из 2000 для того чтобы научиться предсказывать. В начале каждого графика видны явные проблемы, из-за не очень большой точности, но потом, ближе к середине, особенно на графике предсказания графика акций IBM точность явно возрастает, из-за все большего количества данных для обучения, к тому же, я использовал такую систему, что т.к. рынок сильно связан между собой, тем более в одной отрасли(IT-компании), прогноз курса IBM опирается на уже предсказанные данные для Apple и берет их в расчет тоже.

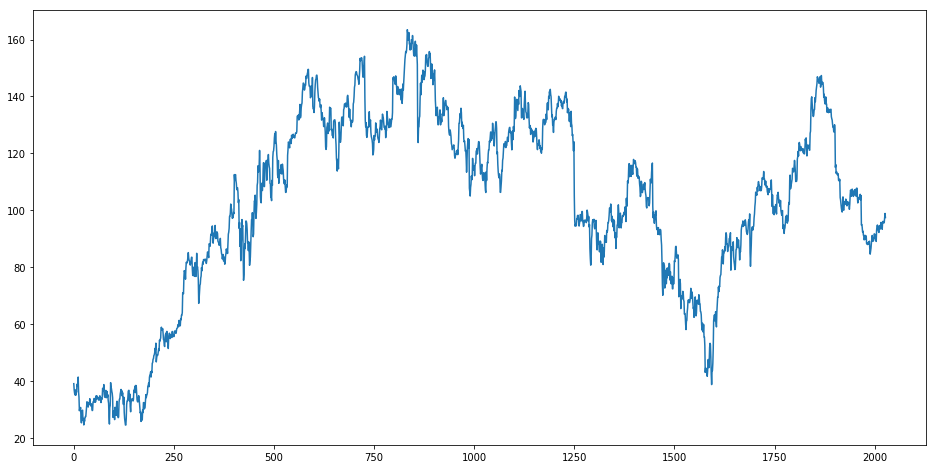
Теперь можно попробовать предсказать реальный результат, а не просто обучить нейросеть.

# Вывод

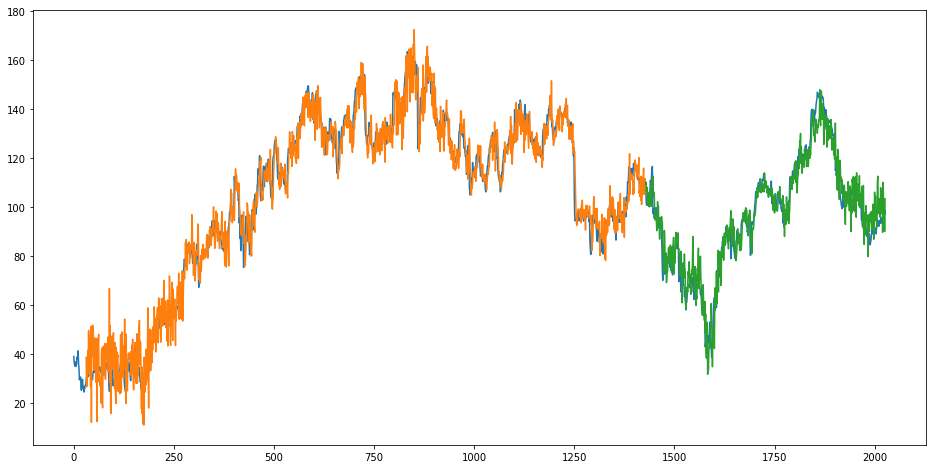
Потратив почти 40 часов чистой работы на написание нейросети и около 7 часов на написание документации, я закончил.

Вот итоговые результаты:

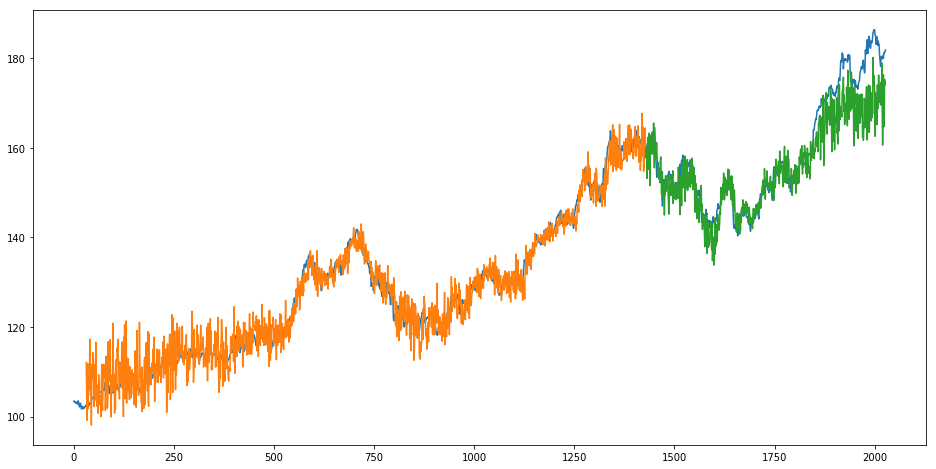
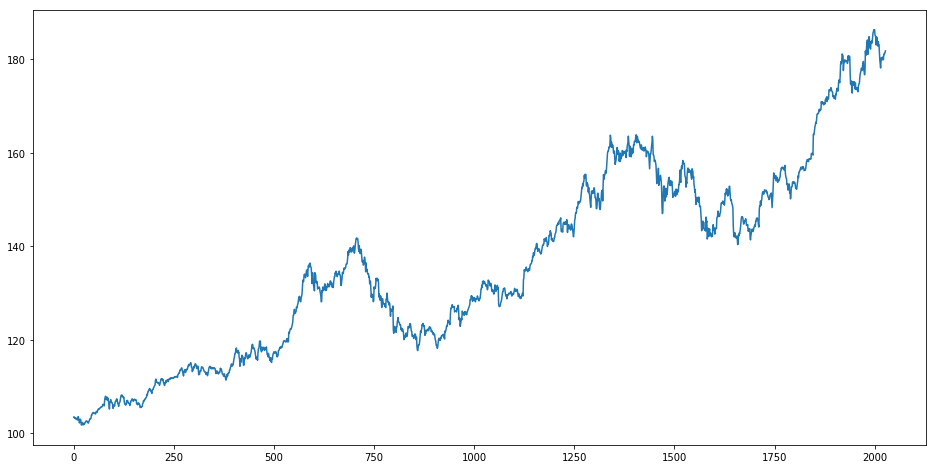
Первый график - курс акций IBM настоящий.



А вот это самое интересное - оранжевое это данные, по которым училась моя нейросеть и после каждой точки на графике, она проверяла, на сколько верный результат. А вот зеленое - результаты, которые моя нейросеть вообще не видела, но предсказала. И заметно, что к концу, есть небольшие проблемы с точностью предсказанием из-за того, что каждый новый результат предсказания основывается на прошлых результатах, а значит предсказание на предсказании, из-за чего точность с каждым разом уменьшается.



Но даже это не отменяет того факта, что точность предсказания в удивительно высока.

Вот подобные же данные для Apple. Он немного хуже, т.к предсказывался первым. 

Честно говоря, я сам не мог ожидать такие хорошие результаты.

## Возможные улучшения

Если продолжать проект и улучшать, сделать интерфейс и внедрить все внутрь биржи, либо парсить[[4]](#footnote-3) данные с биржи на какой-либо свой сайт или в приложение, с учетом всего этого и каждый день обновляя данные для нейросети, думаю можно создать очень хороший продукт, который будет достаточно конкурентоспособен. Еще есть идеи того, чтобы кроме обычного математического предсказания, сделать предсказания еще и по новостям, но тут есть огромная проблема, чтобы парсить все данные по новостям в темах биржи, а еще научить распознавать уже другую нейросеть, какая новость будет вести курс акций вверх, а какая вниз. Идей для улучшения много, если говорить проще. Но **итог: у меня получилось** написать достаточно хорошую нейросеть с большой точностью предсказания.

# Источники

1. <https://github.com/Hydra1War/Rate-prediction> - ссылка на гитхаб репазиторий проекта.
2. <https://www.python.org/> - я использовал Python, как язык програмирования для написания проекта.
3. <https://pytorch.org/> - основная библиотека, с помощью которой работает нейросеть.
4. <https://numpy.org/> - библиотека для разных математический операций.
5. <https://pandas.pydata.org/> и <https://matplotlib.org/> - библиотеки для анализа результатов и создания графиков.
6. <https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs> - база данных для обучения нейросети.
7. <https://stackoverflow.com/> - форум для программистов на который я обращался если сам не мог решить ошибку в программе.
8. <https://docs.google.com/document/d/1wztnnR7A3F-W6PdVmvISPw3zavRzX5BceASMUR79ieQ/edit?usp=sharing> - реферат по проекту.

1. ML - Machine Learning(машинное обучение) [↑](#footnote-ref-0)
2. **Те́нзор** (от [лат.](https://ru.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *tensus*, «напряженный») — применяемый в [математике](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) и [физике](https://ru.wikipedia.org/wiki/%D0%A4%D0%B8%D0%B7%D0%B8%D0%BA%D0%B0) вид линейного многокомпонентного алгебраического объекта (объекта [линейной алгебры](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0)), заданного на [векторном пространстве](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE). [↑](#footnote-ref-1)
3. Веса - своеобразная ценность нейронов. Если мы знаем, что какое-то изменение в тензоре сильно повлияло на результат предсказания, то если повлияло в лучшую сторону - увеличиваем веса. Если повлияло в худшую сторону - уменьшаем. [↑](#footnote-ref-2)
4. Парсить - брать данные с какого либо сайта и переносить [↑](#footnote-ref-3)