



СВЕТОЧ

Образовательный центр

ПРОГРАММА
ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

**“Анализ данных с использованием нейросетей в DATA
Science”**

Преподаватель курса
Ахмедов Марлен Игоревич

Выполнил
Комзин Виталий Борисович

№ потока

DS(144)_23-2.2

**СОДЕЙСТВИЕ
ЗАНЯТОСТИ**

Федеральный
проект

Москва **2023** г.



Проект по теме:

“ Предсказание цены на акции Сбербанка с помощью рекуррентной нейронной сети (RNN)”

содержание презентации:

Сбор данных

Подготовка данных

Выбор архитектуры модели

Обучение модели

Оценка модели

Визуализация результатов

Документация и отчет

Презентация проекта, его результаты

Дальнейшие улучшения



Сбор данных

Для своей задачи предсказания цены по акциям Сбербанка я взял данные с сайта компании Финам. База данных содержит порядка 4 тысяч записей за период с 2008 по 2023 год в виде дневных значений.

Данные загружаются в виде таблицы значений:

Дата, Время, Цена открытия, Максимум, Минимум, Цена закрытия, Объем торгов

	<DATE>	<TIME>	<OPEN>	<HIGH>	<LOW>	<CLOSE>	<VOL>
0	20080109	103000	10401.0	10401.0	10250.0	10398.0	318
1	20080109	104500	10374.0	10450.0	10374.0	10411.0	235
2	20080109	110000	10420.0	10465.0	10400.0	10400.0	712
3	20080109	111500	10410.0	10411.0	10380.0	10380.0	157
4	20080109	113000	10380.0	10410.0	10349.0	10355.0	158

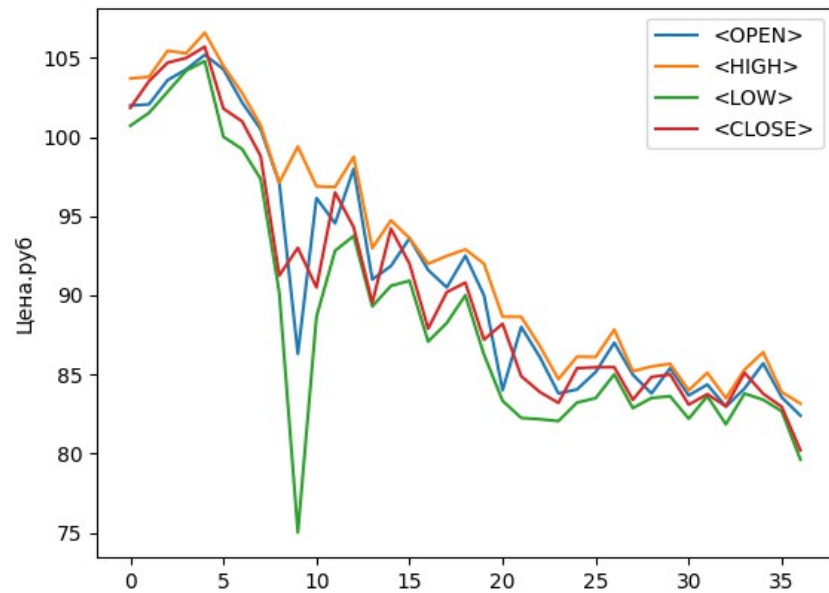


График движения цен акции Сбербанка за период

Подготовка данных

Данные взяты из подготовленной выборки, поэтому в самостоятельной обработке и чистке не было необходимости . После того как загрузил данные :

```
df = pd.read_csv('SBER_080109_231205')
```

Выбираю колонки для обработки нейросетью и разделяю на обучающую и тестовую выборки в соотношении 8:2

Split = 0.8

```
i_split = int(len(df) * split)
```

```
cols = ['<CLOSE>']
```

```
data_train = df.get(cols).values[:i_split]
```

```
data_test = df.get(cols).values[i_split:]
```

Подготовка данных

При обработке финансовых данных требуется нормализация, чтобы все данные имели одинаковую шкалу. Провожу нормализацию данных с помощью библиотеки sklearn

```
scaler = MinMaxScaler(feature_range=(-1, 1))  
data_train_scaled = scaler.fit_transform(data_train)  
data_test_scaled = scaler.fit_transform(data_test)
```

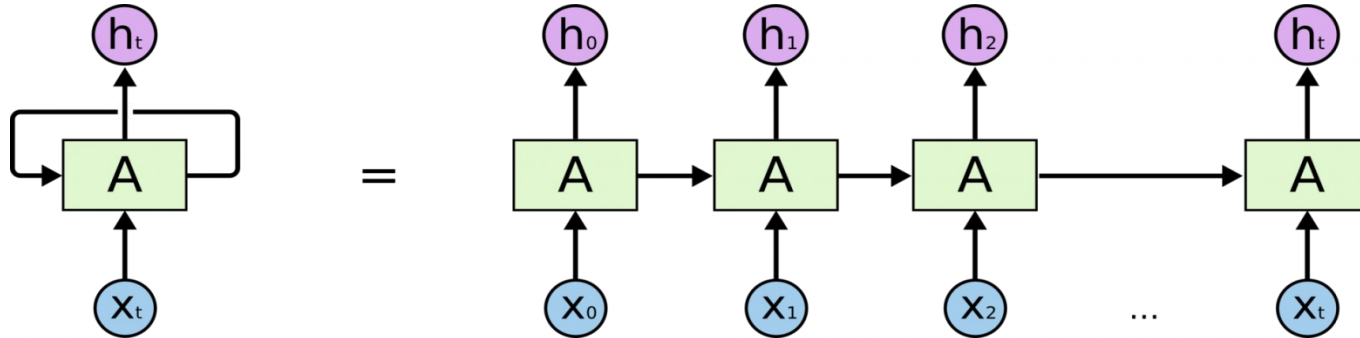
Определяю глубину предсказаний равную 10 дням и формирую соответствующие массивы

```
look_back = 10  
X, y = create_dataset(data_train_scaled, look_back)  
X_test, y_test = create_dataset(data_test_scaled, look_back)
```

Выбор архитектуры модели

Модель Sequential представляет собой линейный стек слоев.

Рекуррентные нейросети используются для обработки временных рядов и текстов. Они учитывают временную динамику данных и позволяет передавать информацию с одного шага сети на другой. Поэтому я остановил свой выбор на Рекуррентной нейросети с памятью.LSTM



Обучение модели

После выбора архитектуры нейросети ее необходимо обучить. Обучение нейросети заключается в поиске оптимальных параметров, которые наилучшим образом соответствуют данным. Для обучения нейросети используются алгоритмы оптимизации, такие как градиентный спуск или метод Ньютона. Обучение продолжается до тех пор, пока нейросеть не начнет давать хорошие результаты на тестовых данных. Дообучение модели не использовалось.

```
model = Sequential()  
model.add(LSTM(4, input_shape=(1, look_back)))  
model.add(Dense(1))
```


Оценка модели

После обучения нейросети необходимо оценить ее качество. Для этого используются различные метрики, такие как точность, полнота, F1-мера и другие. Выбор метрики зависит от конкретной задачи. Например, для задачи классификации обычно используется точность, а для задачи регрессии - среднеквадратичная ошибка. Для оценки качества модели выбрана среднеквадратичная ошибка. С каждой следующей эпохой обучения ошибка уменьшается и в итоге составляет:
0.0027

Epoch 5/5

3174/3174 - 7s - loss: 7.8524e-04 - 7s/epoch - 2ms/step

25/25 - 1s - loss: 0.0027 - 870ms/epoch - 35ms/step

Потери на тестовых данных 0.0027108564972877502

25/25 [=====] - 1s 3ms/step



Оценка модели

После обучения нейросети необходимо оценить ее качество. Для этого используются различные метрики, такие как точность, полнота, F1-мера и другие. Выбор метрики зависит от конкретной задачи. Для задачи классификации используется точность, а для задачи регрессии - среднеквадратичная ошибка. Для оценки качества модели мной выбрана среднеквадратичная ошибка. С каждой следующей эпохой ошибка уменьшается и в итоге составляет:
0.0027

Epoch 5/5

3174/3174 - 7s - loss: 7.8524e-04 - 7s/epoch - 2ms/step

25/25 - 1s - loss: 0.0027 - 870ms/epoch - 35ms/step

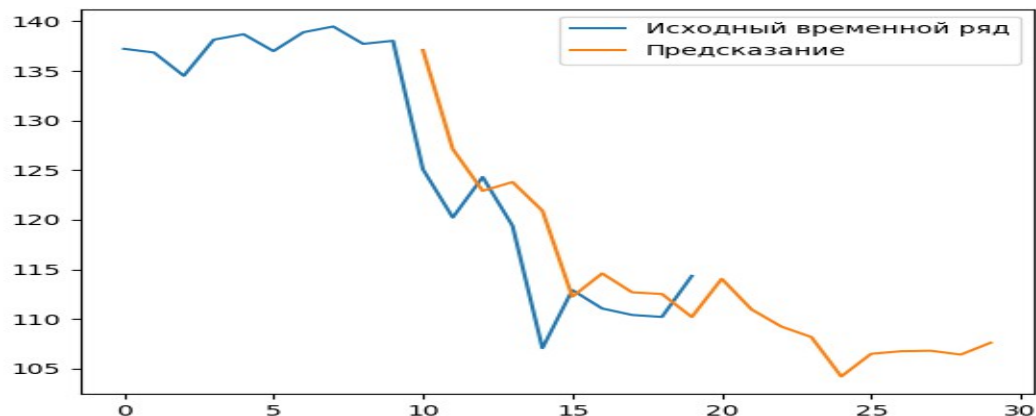
Потери на тестовых данных 0.0027108564972877502

25/25 [=====] - 1s 3ms/step



Визуализация результатов

Полученные результаты анализируются и визуализируются с помощью графика для лучшего понимания и интерпретации результатов. Это помогает определить, насколько эффективно модель справляется с поставленной задачей.



Документация и отчет

В процессе работы над проектом мною были рассмотрены три вида нейросетей:

1 нейронная сеть на языке Питон без применения библиотек Keras и Tensorflow, состоящая из:

- двух входов,
- двух нейронов в скрытом слое
- слой вывода с одним нейроном

2 сеть прямого распространения с одним скрытым слоем

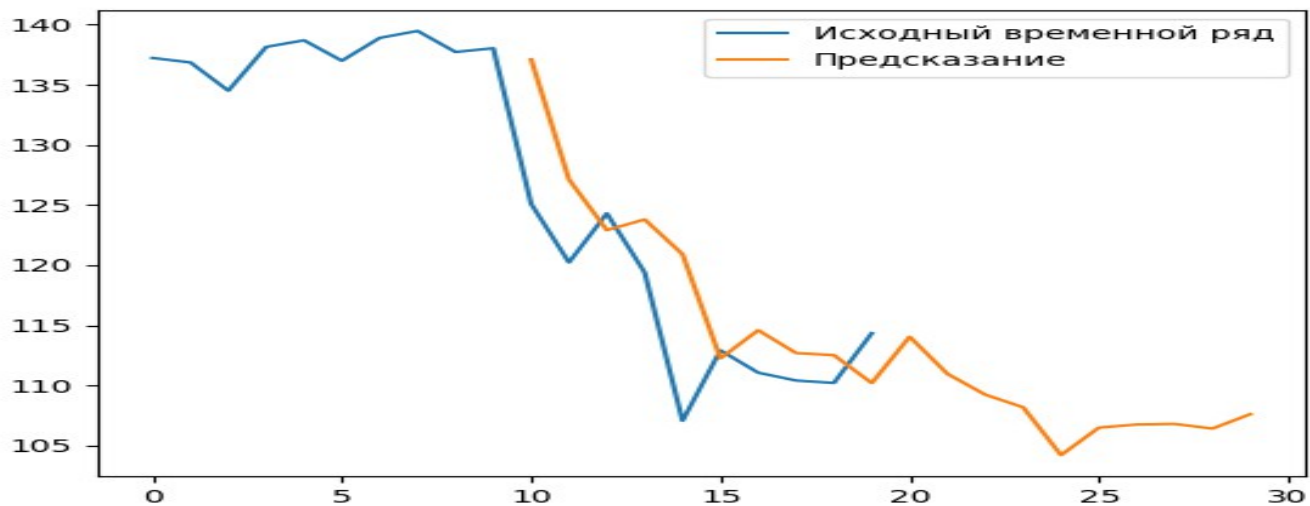
3 рекуррентная сеть (RNN) с памятью, как наиболее подходящая для обработки временных рядов

А также ознакомился и применил библиотеки для обработки данных Pandas и Sklearn, Библиотеку для визуализации данных Matplotlib.pyplot

Получил результаты обучения нейросетей и ознакомился с методами их оценки.

Презентация проекта, его результаты

Результатом проекта является предсказание цены акций Сбербанка в течении последующих 10 дней



Дальнейшие улучшения

Для улучшения проекта и расширения задачи можно применить изменение архитектуры модели или изменение функций потерь. Также можно рассмотреть возможность использования других типов моделей и добавление новых источников данных для улучшения результатов.



СВЕТОЧ

Образовательный центр

Благодарю за внимание!

**СОДЕЙСТВИЕ
ЗАНЯТОСТИ** | Федеральный
проект

info@eduom.ru

