

Лабораторная работа 3.

Вариант 14

Москалев Владислав Александрович.

Github репозиторий с лабораторными - https://github.com/worlddeleteRin/neural_sets

Акция – SIRI

Начальная дата – 2013.07.01

Конечная дата – 2016.02.01

Максимальный лаг – 4

Импортируем все необходимые компоненты и загружаем исходные данные. В моем случае в качестве исходных данных будут использоваться акции SIRI в промежутке с 2013.07.01. Строим график исходного временного ряда, смотрим на результаты:



Строим обучающую выборку, в моем случае интервал временного лага будет от 1 до 4. Разбиваем выборку на обучающую и тестовую, для обучающей берем 70%.

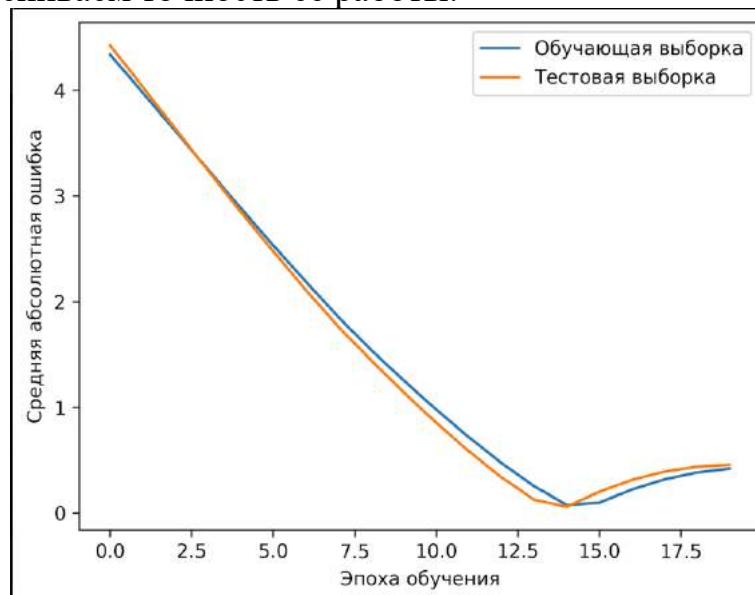
Создаем нейронную сеть, будет состоять из следующих слоев:

```
model = Sequential()
model.add(
    Dense(
        50,
        activation = 'relu',
        input_shape = (x_train.shape[-1],)
    )
)
model.add(
    Dense(
        50,
        activation = 'relu',
    )
)
model.add(
    Dense(1)
)
model.compile(
    optimizer = 'adam',
    loss = 'mse',
    metrics = ['mae']
)
```

Вывод model.summary() имеет следующий вид:

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 50)	200
dense_1 (Dense)	(None, 50)	2550
dense_2 (Dense)	(None, 1)	51
Total params: 2,801		
Trainable params: 2,801		

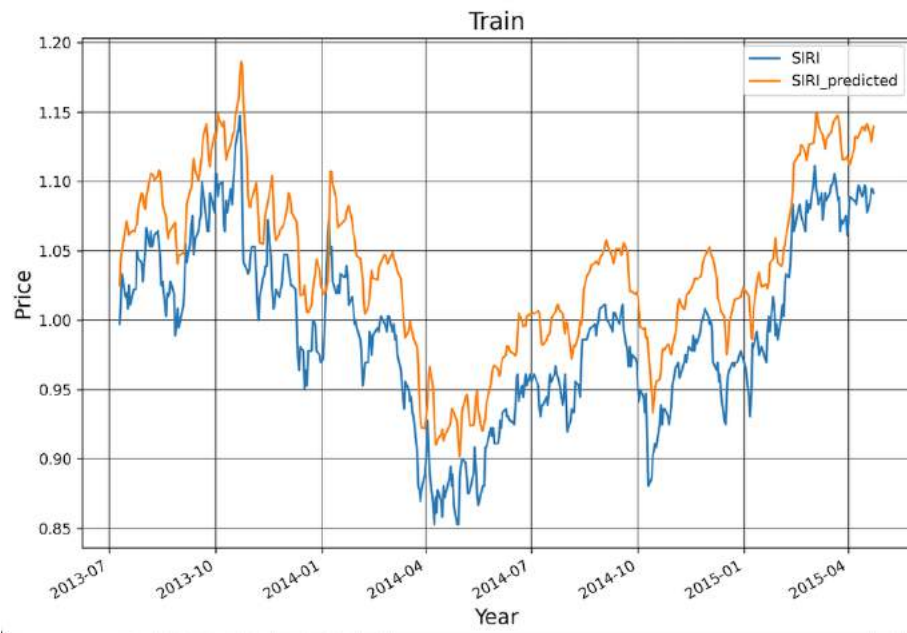
Обучаем сеть, оцениваем точность ее работы:



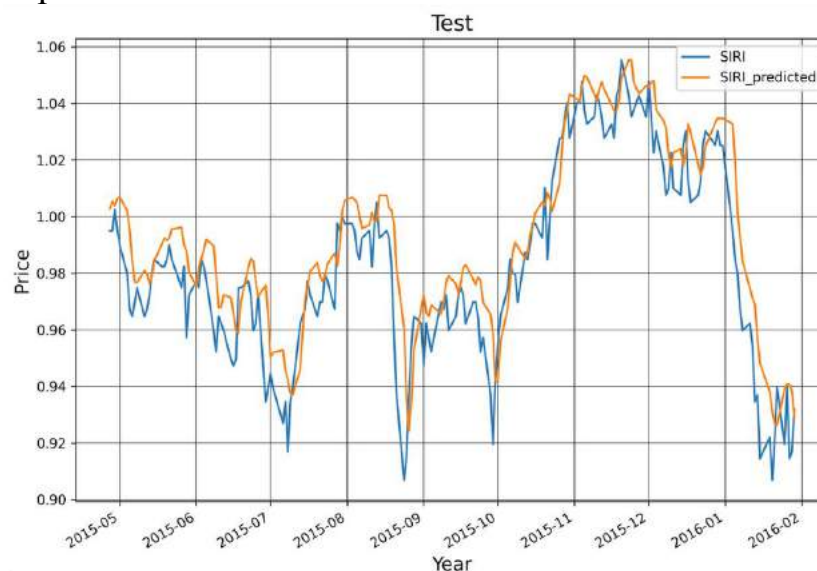
Как видно по графику, примерно с 14 эпохи обучения модели показатели средней абсолютной ошибки начинают увеличиваться, то есть, ухудшаться, что может свидетельствовать о переобучении модели.

```
3 Среднеквадратичное отклонение: 0.20980694890022278
4 Средняя абсолютная ошибка: 0.45531702041625977
```

Визуализация работы сети:
На тренировочной выборке:



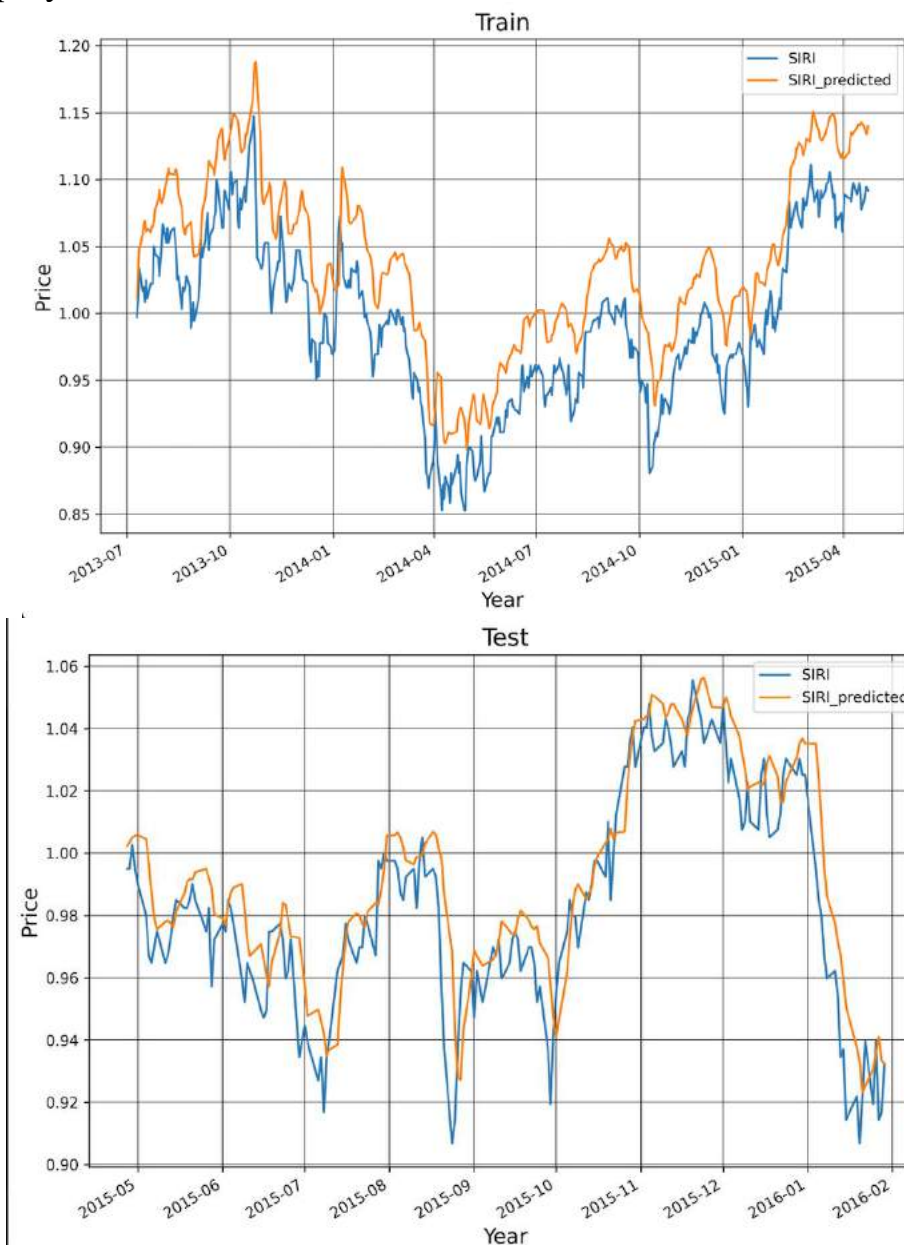
На тестовой выборке:



Оптимизируем гиперпараметры для нейронной сети и подбираем их. В качестве типа тюнера будет использовать RandomSearch. Статистика из `tuner.search_space_summary()`:

```
Default search space size: 4
activation (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'sigmoid', 'tanh', 'elu', 'selu'], 'ordered': False}
units_input (Int)
{'default': None, 'conditions': [], 'min_value': 4, 'max_value': 256, 'step': 4, 'sampling': None}
units_hidden (Int)
{'default': None, 'conditions': [], 'min_value': 4, 'max_value': 256, 'step': 4, 'sampling': None}
optimizer (Choice)
{'default': 'adam', 'conditions': [], 'values': ['adam', 'rmsprop', 'SGD'], 'ordered': False}
** Начинается большое обучение через 3 сек., готовимся ждать... **
```

Производим расчет обучений нашей модели с различной архитектурой 100 раз. Ждем... Выбираем наилучшую модель, обучаем нашу модель с ее помощью, смотрим на результаты!



[0.0028048616368323565, 0.04107581824064255]
Среднеквадратичное отклонение: 0.0028048616368323565
Средняя абсолютная ошибка: 0.04107581824064255

Судя по средней абсолютной ошибке модель должна предсказывать с точностью до 99,6%