Optimal stock trading with TinkoffAPI

Ivan Lukyanenko

Optimization Class Project. MIPT. 2022

Introduction

As soon as trading became possible for private traders, but not only for companies the actuality of stock predicting researches was increased. There is no complete open-source examples how to build trading-bot using Tinkoff Open API. Here is the solution. The goal of the project is to understand how to predict stock prices in nearest future and build system of predicting and automated trading. In this article we providing manual of how to use Python Open API for trading and our GRU-network as an base of our trading strategy.

Tinkoff Open API

There is several unofficial Python SDK for Tinkoff API.

We are using openapi made by Awethon

openapi provides sufficient api's to work with your Tinkoff Investments.

The most inssteresting ones:

- OrdersApi() - post orders for buying/selling and cancelling active orders
- PorfolioApi() - info about your actual portfolio
- MarketApi() - info about the stock market

MyClient

In our project we encapsulate working with openapi-client in our MyClient class.

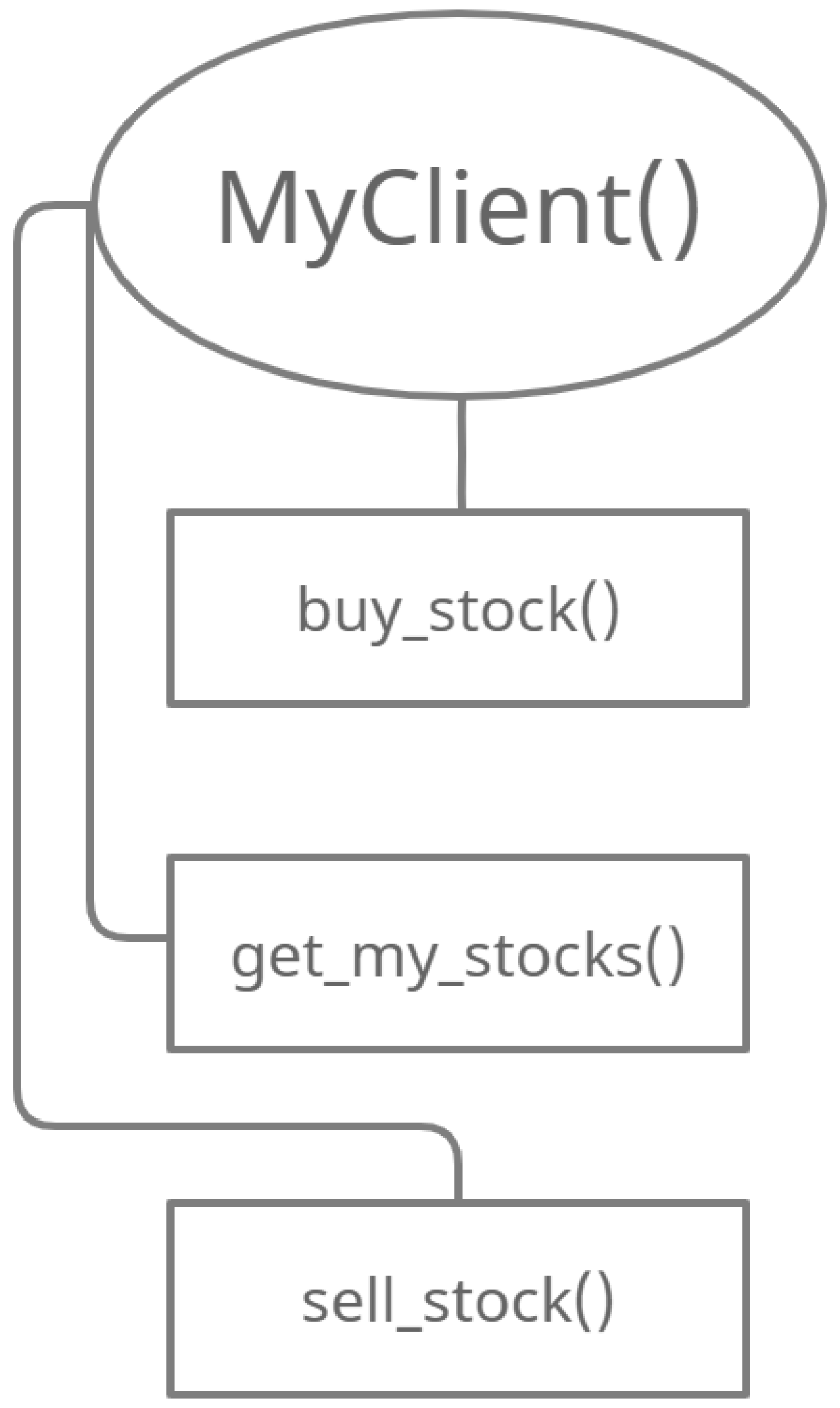


Figure 1: MyClient scheme

GRU

We use GRU for predicting stock price for the next day to make a deccision of buying/selling stock.

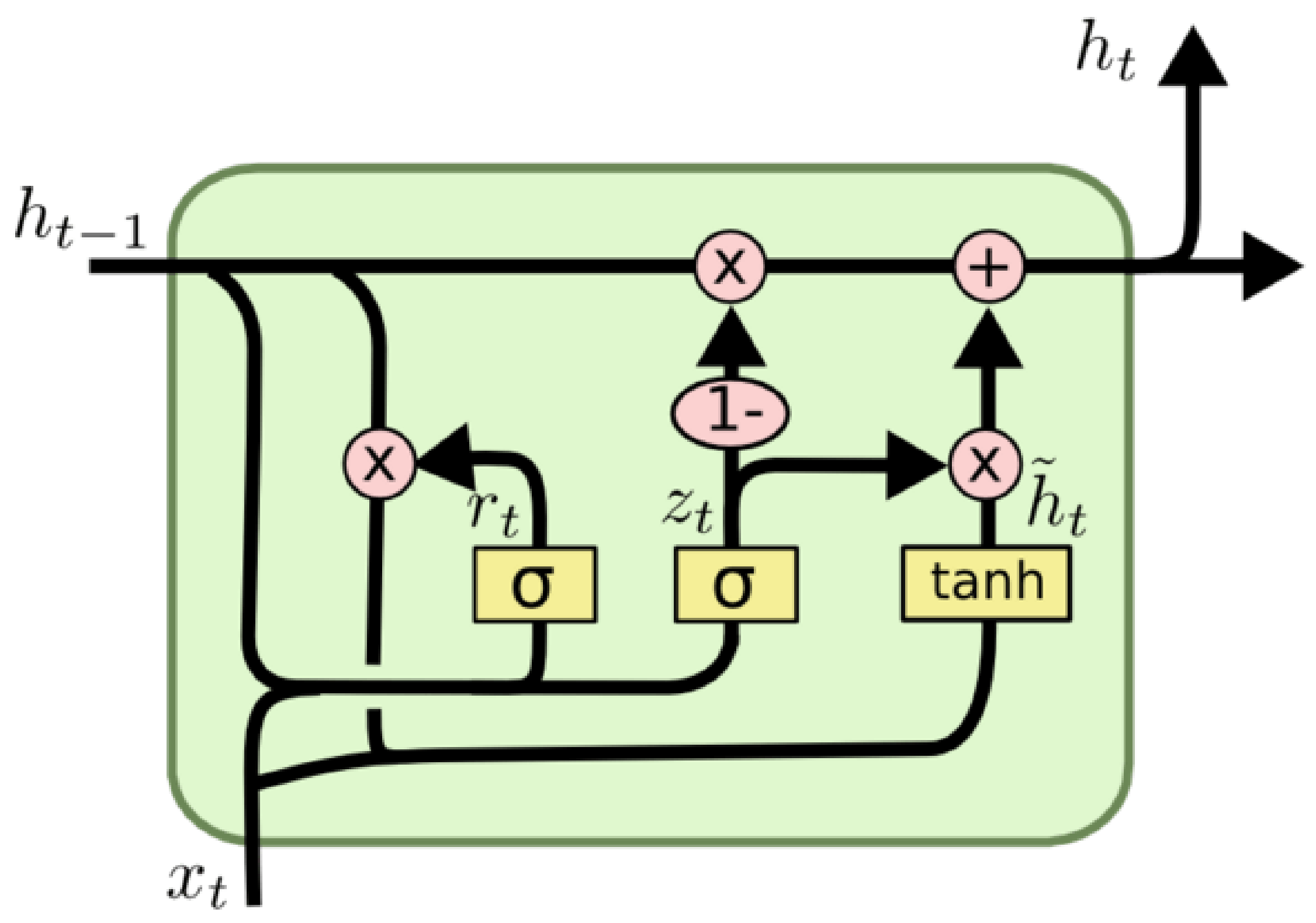


Figure 2: Gated Reccurent Unit

GRU - network is memory based network for temporal data. Using its hidden cells it take in account previous timestamp in a sequence. That's why we choose it for modeling sequences of stock prices.

Training

We consider our problem as regression task. That's why we are going to use MSELoss for training model.

Mathematically formilized our optimization problem is following:

$$\sum_{k=1}^{k=n} \left(f(w_k, x_k) - y_k\right)^2 \to \min_{w \in R^d} \tag{1}$$

where are

- d - dimension of parameters space
- n - number of train samples
- $f(w, x)$ - our GRU model
- y - real stock prices

We are solving this optimization problem with Adam optimizer with different learning rate decays.

metric/model	ADAM,lr=0.001	ADAM,cos-decay	ADAM,exp-decay
MSE	0.08	0.12	0.08
MAE	0.13	0.14	0.12

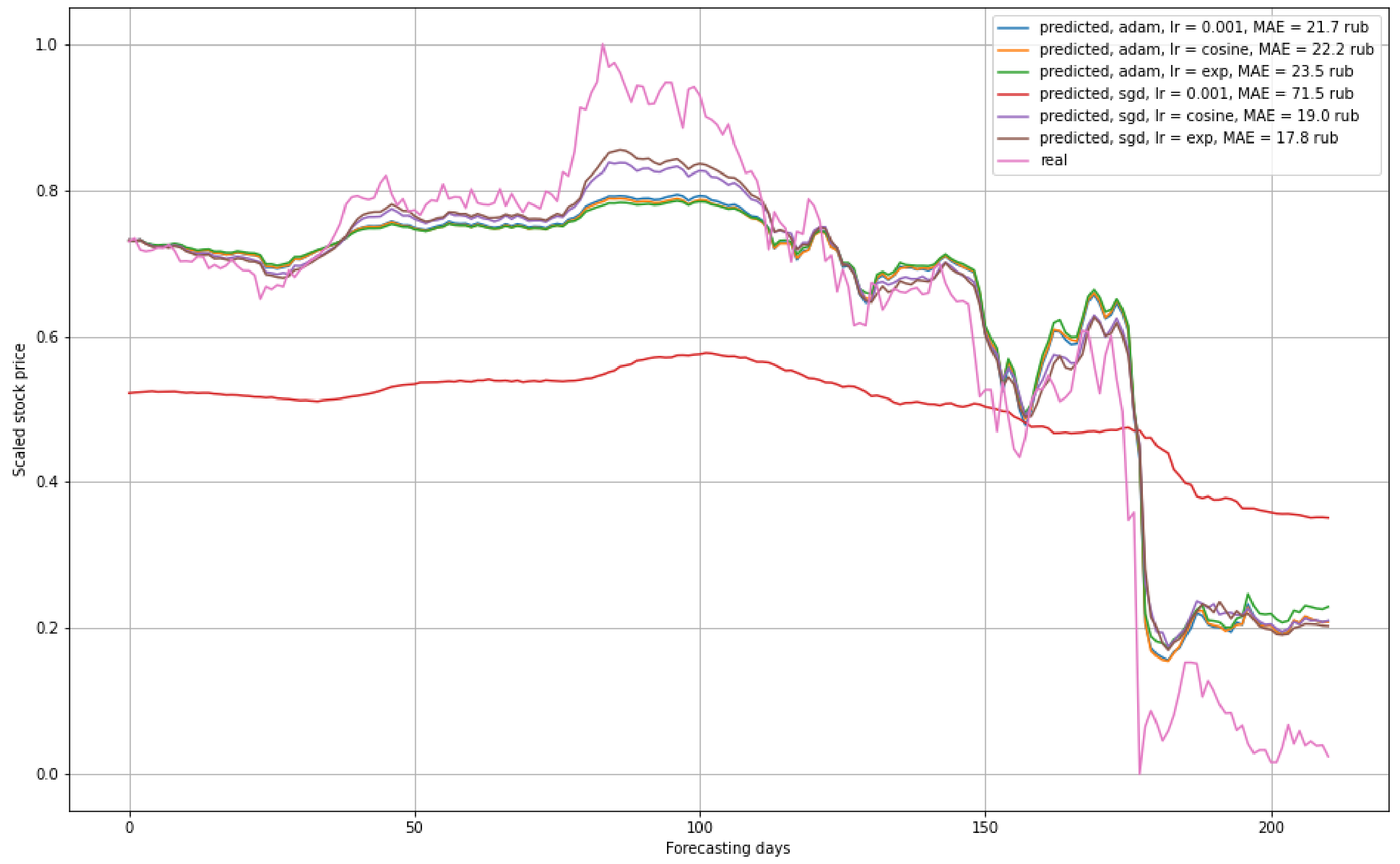
Table 1: model trained with ADAM.

metric/model	SGD,lr=0.001	SGD,cos-decay	SGD,exp-decay
MSE	0.41	0.03	0.05
MAE	0.42	0.05	0.04

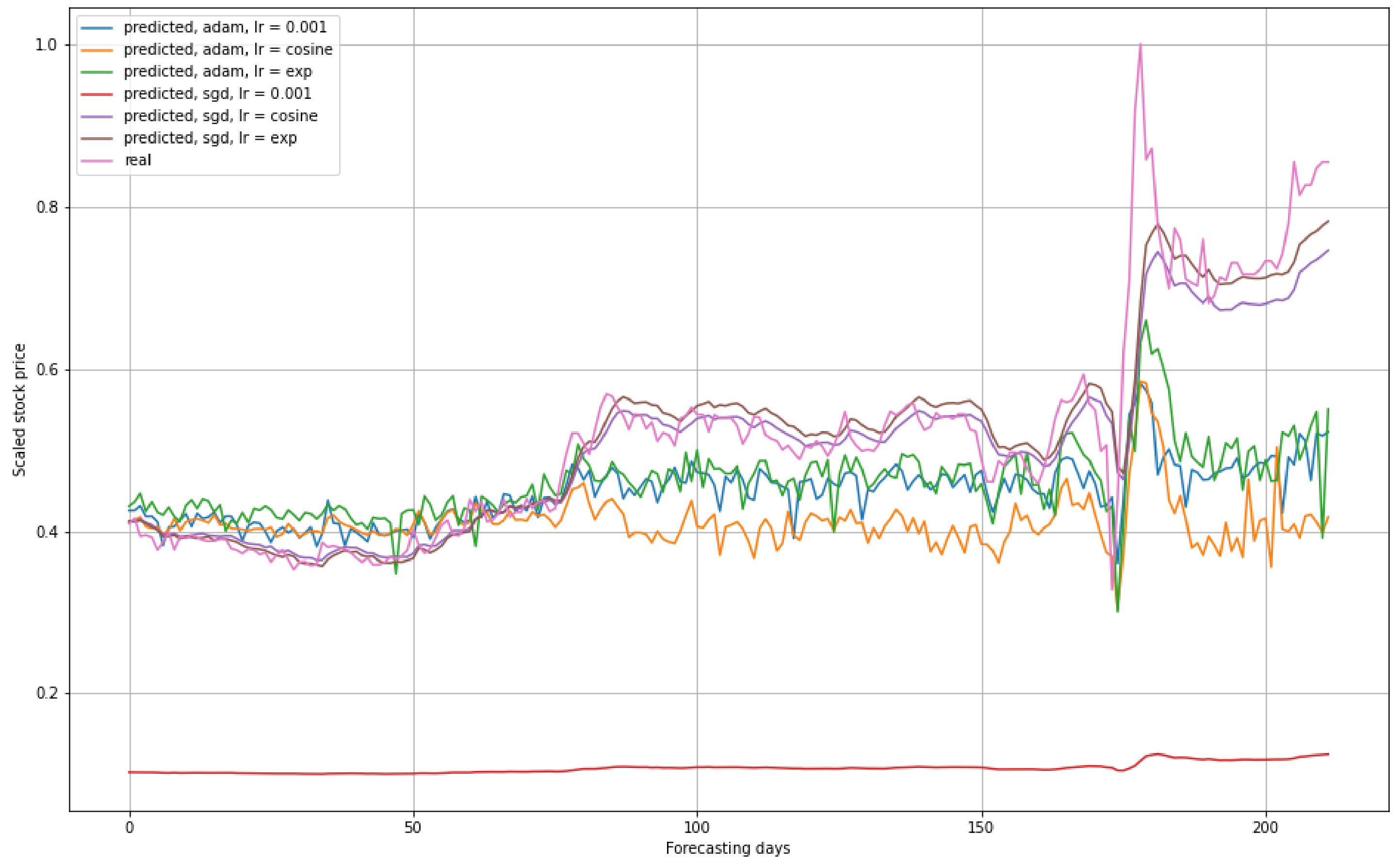
Table 2: model trained with SGD.

Results

Using our model on sber.me data with different training scenarios:



Plot 1: SBER.ME



Plot 2: PHOR.ME

Every training is done in 100 epochs in order to compare methods in the sample setting. As we can see model fitted with default SGD is underfitted. Other scenarios are pretty well and could use in real problem.

Conclusion

Despite unstable situation on Russian market we can see that our simple model could be used for forecasting stock prices. Using this model we automate decision making for buying/selling stocks.

References

- GitHub: git OST
- Colab: Colab OST
- TinkoffAPI: Awethon