Applying Deep Reinforcement Learning To Algorithmic Trading

1st Nikitin Petr
Department of Data Analysis and Machine Learning
Financial University under the Government of the Russian Federation
Moscow, Russia
PVNikitin@fa.ru and 0000-0002-2808-6526

2nd Korchagin Sergey
Department of Data Analysis and Machine Learning
Financial University under the Government of the Russian Federation
Moscow, Russia
SAKorchagin@fa.ru and 0000-0001-8042-4089

3rd Rimma Gorokhova
Department of Data Analysis and Machine Learning
Financial University under the Government of the Russian Federation
Moscow, Russia
RIGorokhova@fa.ru and 0000-0001-7818-8013

***Abstract*— The article presents an algorithm for trading long contracts with one asset in the financial market in the Python programming language using the LSTM neural network using the Keras library. The formalized LSTM model solves the vanishing gradient problem, which can hold the gradient of the objective function relative to the state signal. As applied to our problem, such an improvement in the model allows us to collect data on certain patterns of price changes. Sharpe Ratio is used to determine the optimal strategy and decision making at each time of application. The optimal minimum time period for the model operation has been determined; the signal transmission delay from the moment the market situation changes until the signal is received by the model, which will be infinitely small, and the computing power will be considered infinitely large. These assumptions give the right to say: when the market situation changes, the model is instantly ready to react and make a decision to sell, buy or hold an asset.**

Keywords — algorithmic trading, deep learning, reinforcement learning, recurrent neural networks, LSTM model.

I. Introduction

Trading on the stock exchange is currently gaining momentum and becoming more and more relevant. These issues were considered by both domestic and foreign researchers [1-3]. The introduction of computing systems and the increase in computing power makes it necessary to use deep learning, and it is reflected in a number of studies [4-6]. At the same time, there is a complication of the mathematical base of the work performed, which entails an increase in computing power, due to which machines per unit of time can process more metrics from year to year and give more accurate solutions [7, 8]. The use of deep learning has led to a quantum leap in algorithmic trading [9, 10].

To a certain extent, algorithmic trading can be applied to one degree or another in every financial market where data is not sparse [11, 12]. This criterion is met by the foreign exchange, index markets, blue chips, and the Treasury bond market. As part of this research, we will focus on the blue-chip and indices market.

When making a decision to include an asset or its derivatives in the portfolio, the following are investigated: volatility (risk) and the expected return (mathematical expectation) of the asset [5, 7, 10]. When trading one asset, the trainee agent has 3 options: buy, sell, hold. Generally speaking, there can be much more actions: take a short position, get cash, take an option, and so on, but as part of the work, we will consider a specific task. When solving the problem, we will rely on the supply-demand for the asset. If the difference between supply and demand is significant, that is, demand is less than supply in terms of price, then the market is in equilibrium, the agent expects a sideways trend. If the market is sparse in terms of demand, then this situation is a driver for the sale, since minor fluctuations in the market can trigger the triggering of stop-losses and a sharp decrease in the value of the asset.

II. Methodology

In the application of reinforcement learning to trading in the financial market, the observation will be the price of the asset, the volume and price of bid and ask, the base rate of the currency in which the asset is traded.
The action is buying / selling / holding an asset. The reward is the change in the Sharpe indicator. The policy of the agent being trained is to maximize the Sharpe exponent. In mathematical terms:

$$\pi^* = \underset{\pi}{argmax}\, E[Sh|\pi] \tag{1}$$

$$Sh = \sum_{t=0}^{\infty} \gamma^t sh_t \tag{2}$$

$$sh_t = \frac{E[R_t - R_0]}{\sqrt{var[R_t - R_0]}} \tag{3}$$

where R_t – is the return on the asset per month t,
R_0 – is the return on the risk-free asset for the month t.
γ – discount rate, $(\gamma \epsilon [0, 1])$.
The parameter determines the importance of future rewards and motivates the agent to take action. The higher this indicator, the more the model is focused on long-term results.

The formalized LSTM model solves the vanishing gradient problem, which can hold the gradient of the objective function relative to the state signal. As applied to our problem, such an improvement in the model allows us to collect data on certain patterns of price changes, that is, when predicting the price of the next step, we rely not only on the data of the previous step, but also on earlier data, when there was a similar state of the environment.

Since we apply the supervised learning method, the input time series are transformed into a sample with one label, the length of the time series is *T*, the window size is *W*, the input of the sample is: $s(x_t, x_{t+1}, \dots, x_t + W)$. How we will use the activation function ReLU.

We use MSE as the loss function. We use Adaptive Moment Estimation as the parameter optimization function. To determine the parameter adjustments, we will determine the moving average of the gradients.

The formula for changing the parameters of the model is written as follows:

$$w_p = w_{p-1} - \frac{\eta \cdot \widehat{m_p}}{\sqrt{\widehat{v_p}} + \epsilon} \quad (4)$$

Using the coefficients η and ε, we can adjust the rate of selection of the values of the model parameters.

III. Results and implications

To test the model, we used the AAPL ticker data from 03/14/2000 to 03/13/2019, as the risk-free rate was used by the US Federal Reserve System for the same period.

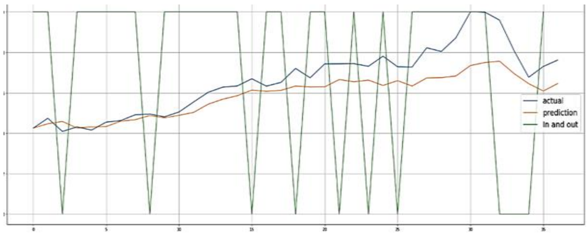

Fig. 1. Model graph for 36 months.

On the graph shown in Figure 1, we can observe that over 36 months the model advises to take 9 long positions of different duration, of which only one with 0 gross profitability. This experiment suggests that the algorithm has a high enough accuracy for planning cash management. We can understand how long over the course of several years the money will be in the asset, and how much it will be inactive.

Figure 2 shows a graph of the deviation of predicted prices from real ones for the next 3 years after the training sample. Predicted prices are plotted on the OX axis, and real prices are plotted on the OY axis.

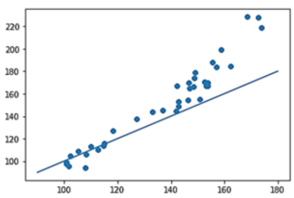

Fig. 2. Graph of the deviation of predicted prices.

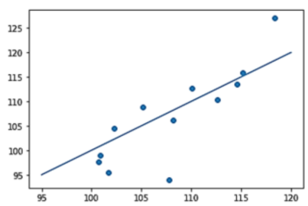

Fig. 3. Graph of the deviation of the predicted price from the real.

The average prediction error was 15.99%, while the asset price increased by 42%, hence the error was 38% of the asset price change. This means that using a model on a three-year horizon, we can predict the general trend of movement and the approximate growth rate of the asset.

Figure 3 shows the deviations of the predicted price from the real one for annual forecasting.

The average prediction error was 5% with an asset price increase of 29%, therefore, the error was 16% of the asset price change. This means that on the annual horizon, the model also predicts the trend and, more accurately, the rate of growth of the asset.

IV. Conclusion

The developed model allows you to conduct a monthly trend, based on technical analysis, predict the trend for 3 years, predict the asset growth rate, asset volatility.

Based on this data, we can plan a portfolio in the long term, how often we can transfer from one asset to another.

To improve the quality of prediction, it makes sense to include in the input data the annual and quarterly reports of the company that issued the securities. Using this data, we will be able to calculate such indicators as the level of debt in relation to revenue, the level of operating profit, free cash flow and the dynamics of these indicators over time. Include the most significant indicators in the agent's policy, thereby improving the model. More financial data about the company will provide the model with a larger observational space. Therefore, it is necessary to work deeper into the methodology of model rewards so that the goal of the algorithm is closer to the goal of the trader - to extract as much profit from the trade as possible. On the other hand, an interesting area of research is to consider the distribution according to which the price of an asset is determined, which will help manage uncertainty.

References

[1] R. A. Kasyanov, "MIFID II: регулирование алгоритмической торговли и высокочастотной алгоритмической торговли," Финансовое право, vol. 19(2), pp. 3–7, 2018.

[2] M. Haiyan et al, "Blockchain financial investment based on deep learning network algorithm," Journal of Computational and Applied Mathematics, vol. 372, 2020, 112723.

[3] T. Borges, R. Neves, "Ensemble of machine learning algorithms for cryptocurrency investment with different data resampling methods," Applied Soft Computing, vol. 90, pp. 106187, 2020.

[4] V. M. Sineglazov, R. S. Koniushenko, "Deep learning fuzzy classifier," Electronics and Control Systems, vol. 2(60), pp. 35-42, 2020.

[5] F. Liu, R. Tang, X. Li et al, "State representation modeling for deep reinforcement learning based recommendation," Knowledge-Based Systems, vol. 205, 2020, 106170 – URL: https://doi.org/10.1016/j.knosys.2020.106170.

[6] F. Liu, et al, "Top-aware reinforcement learning based recommendation," Neurocomputing, vol. 417, 2020, pp. 255-269 – URL: https://doi.org/10.1016/j.neucom.2020.07.057.

[7] H. Parka, et al, "An intelligent financial portfolio trading strategy using deep Q-learning," Expert Systems with Applications, vol. 158, 2020, 113573 – URL: https://doi.org/10.1016/j.eswa.2020.113573.

[8] C. Betancourt, W.-H. Chen, "Deep reinforcement learning for portfolio management of markets with a dynamic number of assets," Expert Systems with Applications, vol. 164, February 2021, 114002 - URL: .https://doi.org/10.1016/j.eswa.2020.114002.

[9] K. Lei, B. Zhang,Y. Li et al, "Time-driven feature-aware jointly deep reinforcement learning forfinancial signal representation and algorithmic trading," Expert Systems With Applications, vol. 140, 2020, 112872 – URL: https://doi.org/10.1016/j.eswa.2019.112872.

[10] T. T. Nguyen et al, "A multi-objective deep reinforcement learning framework," Engineering Applications of Artificial Intelligence, vol. 96, November 2020, 103915. - https://doi.org/10.1016/j.engappai.2020.103915

[11] M. Riesenera, "Applying Supervised and Reinforcement Learning to Design Product Portfolios in Accordance with Corporate Goals

Procedia CIRP," Procedia CIRP, vol. 91, 2020, pp. 127-133 – URL: https://doi.org/10.1016/j.procir.2020.02.157.

[12] G. Jeonga, Y. Kim, "Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning," Expert Systems with Applications, vol. 117, 2019, pp. 125-138 – URL: https://doi.org/10.1016/j.eswa.2018.09.036.