



ANALYSIS OF L2 REGULARIZATION HYPERPARAMETER FOR STOCK PRICE PREDICTION

Arjun Singh Saud^{1*}, Subarna Shakya²

¹Central Department of Computer Science & Information Technology, Tribhuvan University, Kathmandu, Nepal

²Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Lalitpur, Nepal

*Corresponding author: arjunsaud@cdcsit.edu.np

(Received: March 01, 2021; Revised: May 17, 2021; Accepted: June 04, 2021)

ABSTRACT

Nowadays stock price prediction is an active area of research among machine learning researchers. One of the main problems with machine learning models is overfitting. Regularization techniques are widely used approaches to avoid over-fitted models. L2 regularization is one of the most popular and widely used regularization techniques. Regularization hyperparameter (λ) is one key parameter to be optimized for a well-generalized machine learning model. Hyperparameters can't be learned by machine learning models during the learning process. We need to find their optimal value through experiments. This research work analyzed the L2 regularization hyperparameter used with a gated recurrent unit (GRU) network for stock price prediction. We experimented with five stocks from the Nepal Stock Exchange (NEPSE) and observed that stock price can be predicted with lower mean squared errors (MSEs) when the value of λ was around 0.0005. Therefore, this research paper recommended using $\lambda=0.0005$ with L2 regularization for stock price prediction.

Keywords: Gated recurrent unit, L2-regularization, Neural network, Regularization hyperparameter, Stock price.

INTRODUCTION

A large number of investors and traders are involved in the stock market daily and they use their knowledge and experience to estimate the future value of the stock price and make stock buy/sell decisions on the same basis. However, it is very difficult to predict stock price because it depends upon many known and unknown factors like current and projected economic status of the country, current and projected financial performance of the company, status of the company in the industry, brand, and goodwill of the company, political changes, rumors, etc. (Vijh *et al.*, 2020).

Technical analysis and fundamental analysis are the two widely used approaches for stock price forecasting. Fundamental analysis is popular among medium and long-term value investors. Fundamental analysts analyze company data to determine its growth potential (Vanstone & Finnie, 2009), whereas, technical analysis is a popular tool among short-term traders. Technical analysts believe that all the relevant information about a stock is reflected in stock price and hence fundamental analysis can be ignored (Murphy, 1999). They rely heavily upon technical indicators and candlestick patterns to forecast stock price movements (Li & Bastos, 2020). However, it is a difficult and time-consuming task for technical analysts to analyze a large volume of trading data and identify trends from it. Machine learning algorithms are response to this difficulty and are suitable for automated, fast, and more accurate forecasting of stock prices (Nabipour *et al.*, 2020; Parmar *et al.*, 2018; Sarode *et al.*, 2019; Shen & Shafiq, 2020).

Machine learning approaches like support vector machine, feed-forward neural networks, etc. were used by many researchers for predicting stock prices (Henrique *et al.*, 2018; Hu *et al.*, 2013; Jabin, 2014; Kumar *et al.*, 2011). However, due to their inability to remembering context, recurrent neural networks became popular for predicting stock price (Chen *et al.*, 2018; Li *et al.*, 2019). Again, due to the vanishing/exploding gradient problem associated with recurrent neural networks, long short-term memory (LSTM) and gated recurrent unit (GRU) networks became a state-of-art model for predicting stock prices (Lu *et al.*, 2020; Moghar & Hamiche, 2020; Shen *et al.*, 2018; Zhao *et al.*, 2021). Therefore, this research work used the GRU network as a machine learning model for stock price forecasting.

One of the main problems associated with the machine learning model is overfitting (Ying, 2019). Overfitting happens when a machine learning model learns minute details of the training dataset but unable to generalize on the unseen test data. Such models result in low bias and high variance. But, we need a machine learning model that works with low bias and low variance. Regularization is one solution to the overfitting problem associated with machine learning models (Nusrat & Jang, 2018). The main concept behind regularization is to reduce the complexity of a machine learning model by penalizing the larger weights. L2 regularization is one of the widely and regularization techniques.

This research work analyzed the regularization hyperparameter (λ) associated with the L2 regularization

and suggested the best value of this hyperparameter for stock price prediction.

Hyperparameters are the parameters that can't be learned by machine learning algorithms during the learning process (Wu *et al.*, 2019). Their optimized value needs to find through experiments. Tuning a machine learning algorithm means tuning the hyperparameters associated with the machine learning model. The optimal value of the regularization hyperparameter may differ within the problems. Thus, finding the optimal value of the regularization hyperparameter about stock price prediction is a topic to be researched.

MODELS AND DATA

This section presents tools and techniques, stock data, data preprocessing, data preparation, the configuration of the GRU network, and the result processing and presentation strategy used for carrying out the research work.

Gated recurrent unit (GRU) network

One of the main problems associated with the recurrent neural network (RNN) is the vanishing/exploding gradient descent problem (Goodfellow *et al.*, 2016). The vanishing gradient problem is experienced due to small weights and the exploding gradient descent problem is experienced due to large weights. Small weights result in small gradients and large weights result in large gradients. If the gradient is small, it is harder for the neural network to update the weights and the longer it takes to get to the final result.

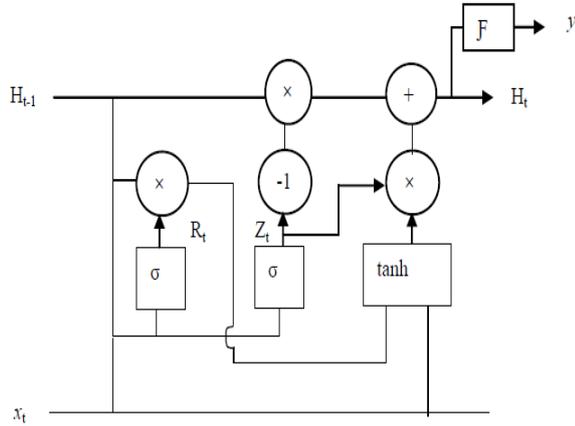


Fig. 1. The architecture of GRU Network

$$\begin{aligned}
 Z_t &= \sigma(W_z x_t + U_z H_{t-1}) \\
 R_t &= \sigma(W_r x_t + U_r H_{t-1}) \\
 H'_t &= \tanh(W_h x_t + (R_t \times H_{t-1}) U_h) \\
 H_t &= (Z_t \times H'_t) + ((1 - Z_t) \times H_{t-1})
 \end{aligned} \tag{1}$$

where, Z and R are update & reset gates and H'_t & H_t are candidate hidden state and hidden state

Exploding gradients are a problem where large error gradients accumulate and result in very large updates to neural network model weights during training. This has the effect of the model is unstable and unable to learn from the training data. LSTM (Hochreiter & Schmidhuber, 1997) and GRU (Cho *et al.*, 2014) networks are variants of RNNs that are proposed to tackle the vanishing/exploding gradient descent problem associated with them. Although both LSTM and GRU networks are similar, the structure of GRU networks is simpler than the structure of LSTM networks and it uses fewer gates than LSTM. GRU networks can be trained and generalized faster than LSTM networks. However, LSTM networks have higher expressive power than GRU networks (Weiss *et al.*, 2018). The architecture and mathematical expression of the GRU network are given below in Fig. 1 and equation (1), respectively.

L2 regularization

The generalization curve of Fig. 2 shows training loss and validation loss. The figure shows that training loss is continuously going down whereas validation loss is going up gradually after reaching some minimal point. This is a clear indication of model overfitting (Ying, 2019). This means the model becomes specialized in the training data set and is unable to generalize on the validation data set. Regularization is one of the techniques used for preventing overfitting. It solves the overfitting problem by discouraging model complexity. It does so by penalizing loss function using weights of the model (Goodfellow *et al.*, 2016).

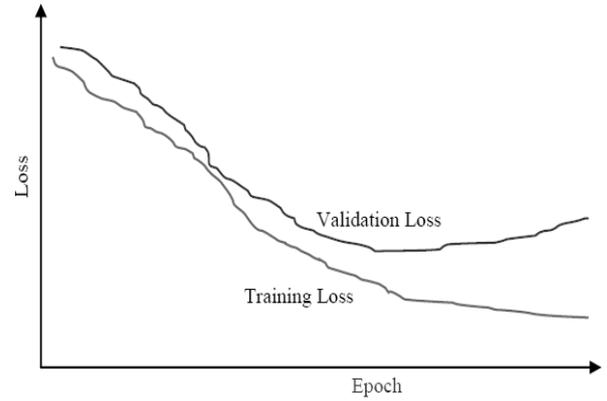


Fig. 2. Training and validation loss

Normally, the mean squared sum of errors is used as a loss function as given in equation (2). As the number of input features to the model increase, the model becomes complex and tries to fit every data point. If we penalize weights and keep them small some of the weights tend to zero. It makes those terms negligible and hence simplifies the model. L2 regularization adds the L2-norm of the weight vector to the loss function as shown in equation (3) (Goodfellow *et al.*, 2016). This additional term added to

the loss function is called regularization term which keeps weight small and thus keeps the model simple and avoids overfitting.

$$L = \frac{1}{n} \sum_{i=1}^n (t_i - y_i)^2 \quad (2)$$

Where t is the target value and y is the predicted value.

$$L = \frac{1}{n} \sum_{i=1}^n (t_i - y_i)^2 + \lambda \sum_{i=1}^n w_i^2 \quad (3)$$

Where λ is the regularization hyperparameter

Regularization hyperparameter (λ) is the penalty term that determines how to penalize the weights of the model. Its value lies between zero and one. If $\lambda=0$, it means no regularization at all. However, if λ is large, it penalizes the weights of the model heavily and most of the weights tend to zero. This may result in a very simple and under-fitted model with high bias. Thus, we need to choose the value of λ carefully to avoid overfitting.

Stock data

Historical trading data from the Nepal Stock Exchange (NEPSE) was used in this research work. Data of five stocks were collected from NEPSE. These five stocks, representing different sectors, were selected randomly. The five stocks selected for this research study were Machhapuchchhre Bank Limited (MBL), Kumari Bank Limited (KBL), Muktinath Bikas Bank Limited (MNBBL), Asian Life Insurance (ALICL), and Butwal Power Company Limited (BPCL). The data was daily trading data from March 2010 to February 2021. The eight features in the collected data were Serial Number, Date, Total Transactions, Total Traded shares, Total Amount, Maximum Price, Minimum Price, and Close price.

Data preprocessing

First, the unnecessary features like Serial Number, Total Transactions, and Total Amount were removed from the dataset. Then, the dataset was arranged in the order of oldest to newest date and the features Total Traded shares, Maximum Price, Minimum Price, and Close Price were renamed as Volume, High, Low, and Close respectively. Finally, the dataset was stored in comma-separated values (CSV) format.

Data preparation

First, the target feature "Next Day's Close" was generated, which was simply close price shifted back by one position. This feature was added because we aimed to predict the next day's close price using past trading data. Then, the features Volume, High, Low, Close, & Next Close was loaded and the dataset was divided to train/validate/test sets in an 8:1:1 ratio. Finally, the dataset was normalized using the standard scalar and the sequence data was created using window size five as

suggested by Saud and Shakya (2020). Thus, input to the GRU network were tuples d_{t-4} , d_{t-3} , d_{t-2} , d_{t-1} , & d_t and the target feature was c_{t+1} . Here t is the current trading day and c_{t+1} is the close price for the day $t+1$. Each d_i is a tuple (v_i, h_i, l_i, c_i) , where v_i is volume, h_i is the high price, l_i is the low price, and c_i is the close price for i^{th} trading day.

Configuration of GRU network

The configuration of the GRU network used in this research work was $4 \times 128 \times 128 \times 128 \times 128 \times 128 \times 1$. Each hidden layer of the network was equipped with an L2 regularizer. Hyperbolic tangent (Tanh) was used as activation function, sigmoid activation function was used as recurrent activation function, and linear activation function was used as activation in the output layer of the GRU network. Besides, Adam optimizer was used with the network so that the learning rate can adapt itself. It is the best optimization strategy for stock price prediction (Saud & Shakya, 2019). The GRU network was trained using a batch gradient descent approach with a batch size of 32.

Inverse scaling of predicted price

After training the GRU network it was used for predicting stock prices for test data. Since the input data was normalized, the predicted price was also in normalized form, whose range was different from actual stock prices and hence it is not understandable to naïve users. Therefore, the predicted stock price was inversely scaled. Then the mean squared error (MSE) between predicted stock price and actual stock prices was calculated and the result was captured.

RESULTS AND DISCUSSION

In this research work, the value of the regularization hyperparameter (λ) varied from 0 to 0.000000005 and the mean squared error (MSE) of the predicted stock price was captured in each case. For each value of λ , the stock price was predicted five times, and the average value of MSE was calculated. The obtained result is presented in Table 1 and the corresponding graphs are presented the Fig. 3. If we look at the graphs of Figs. 3(b) to 3(e), we can observe that the lowest value of MSE is achieved at $\lambda=0.0005$ for the stocks KBL, MNBBL, ALICL, and BPCL. But, for the stock MBL, as shown in Fig. 3(a), the lowest MSE value is achieved at $\lambda=0.000005$. However, this value of MSE is near to the value of MSE achieved at $\lambda=0.0005$.

Another interesting fact that can be observed from Figs. 3(a) to 3(e) is that at $\lambda=0$ higher values of MSE are achieved for all stocks. Then the MSE values seem to be in decreasing trend up to $\lambda=0.0005$. Once the value of λ goes below 0.0005, again higher values of MSE are observed. The reason behind this is that the trained model became overfitted at $\lambda=0$ and hence yielded higher values

of MSEs. A well-generalized model is achieved at $\lambda=0.0005$. Further, as the value of λ goes below 0.0005, model weights are penalized heavily. This results in under fitted model and hence again higher values of MSEs are obtained.

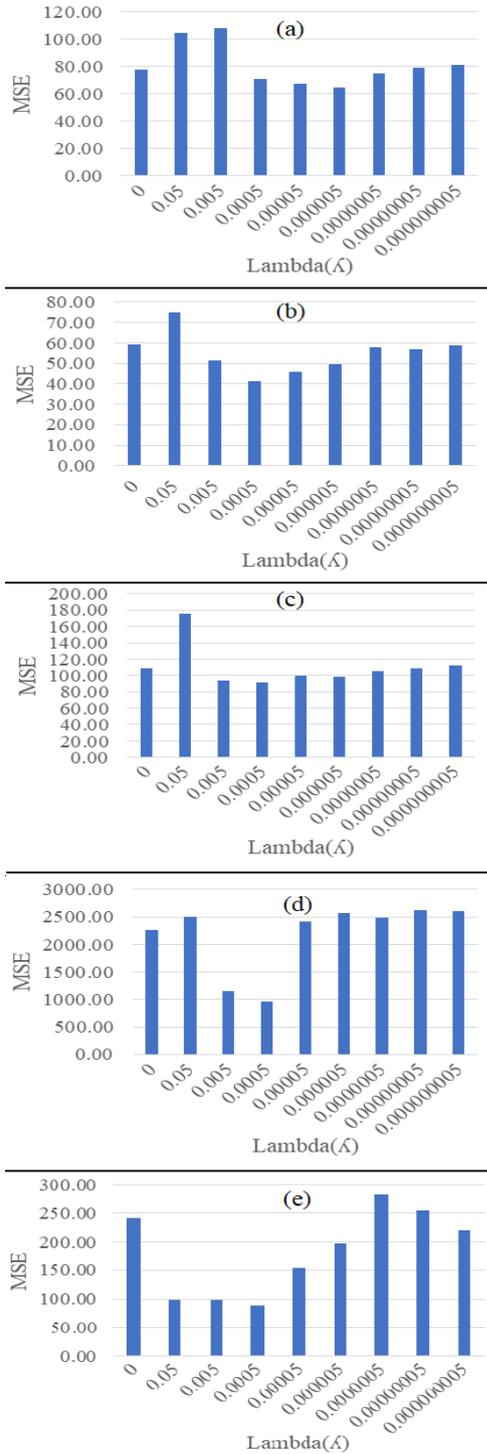


Fig. 3. MSE of (a) MBL (b) KBL (c) MNBBL (d) ALICL and (e) BPCL

Table 1. MSE for varying λ

| λ | Mean squared error (MSE) | | | | |
|-------------|--------------------------|--------------|--------------|---------------|--------------|
| | MBL | KBL | MNBBL | ALICL | BPCL |
| 0 | 77.71 | 59.42 | 108.73 | 2261.84 | 241.38 |
| 0.05 | 104.81 | 75.02 | 205.24 | 2509.72 | 98.41 |
| 0.005 | 108.22 | 51.26 | 94.58 | 1149.18 | 98.72 |
| 0.0005 | 70.62 | 41.28 | 92.09 | 969.85 | 89.44 |
| 0.00005 | 67.56 | 46.08 | 99.30 | 2409.54 | 153.72 |
| 0.000005 | 64.83 | 49.39 | 98.41 | 2570.80 | 197.04 |
| 0.0000005 | 74.81 | 58.09 | 105.46 | 2492.23 | 283.40 |
| 0.00000005 | 79.31 | 57.18 | 108.56 | 2627.43 | 254.95 |
| 0.000000005 | 81.13 | 58.59 | 112.28 | 2596.78 | 220.55 |

CONCLUSION

Stock price prediction is the field of interest for researchers from financial analysis, statistics, and machine learning domains. In this research work, the stock price of five stocks was predicted using the GRU network. L2 regularization was used with the GRU network and the effect of regularization hyperparameter (λ) was evaluated in stock price prediction. From the experimental results, we observed that the best fitted and well-generalized GRU network was achieved at $\lambda=0.0005$. Generally, higher values of MSEs were obtained for the value of λ above and below 0.0005. Thus, we concluded that, in reference to stock price prediction, 0.0005 is the best value for the L2 regularization hyperparameter.

There are many ways in which this research work can be extended. One simple approach is to replicate the research work for predicting the price of stocks listed in other stock exchanges. This will increase the reliability of the research work. Another approach is to replicate the research work for different configurations of the GRU network, which will further increase the reliability of the research work. In addition, the research work can be replicated for other deep learning models besides the GRU network.

REFERENCES

Chen, W., Yeo, C. K., Lau, C. T., & Lee, B. S. (2018). Leveraging social media news to predict stock index movement using RNN-boost. *Data & Knowledge Engineering, 118*, 14-24.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078v3

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning (adaptive computation and machine*

- learning series*) (Illustrate). Massachusetts, USA: The MIT Press.
- Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of Finance and Data Science*, 4(3), 183-201.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- Hu, Z., Zhu, J., & Tse, K. (2013). Stocks market prediction using support vector machine. In *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering* (pp. 115-118). IEEE. <https://doi.org/10.1109/ICIII.2013.6703096>
- Jabin, S. (2014). Stock market prediction using feed-forward artificial neural network. *International Journal of Computer Applications*, 99(9), 4-8.
- Kumar, P. N., Seshadri, G. R., Hariharan, A., Mohandas, V. P., & Balasubramanian, P. (2011). Financial market prediction using feed-forward neural network (pp. 77-84). https://doi.org/10.1007/978-3-642-20209-4_11
- Li, A. W., & Bastos, G. S. (2020). Stock market forecasting using deep learning and technical analysis: a systematic review. *IEEE Access*, 8, 185232-185242.
- Li, C., Song, D., & Tao, D. (2019). Multi-task recurrent neural networks and higher-order Markov Random fields for stock price movement prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1141-1151). New York, USA: ACM. <https://doi.org/10.1145/3292500.3330983>
- Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*, 2020, 1-10.
- Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science*, 170, 1168-1173.
- Murphy, J. J. (1999). *Technical analysis of the financial markets: a comprehensive guide to trading methods and applications*. New York, USA: New York Institute of Finance.
- Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E., & Shahab, S. (2020). Deep learning for stock market prediction. *Entropy*, 22(8), 840. <https://doi.org/10.3390/e22080840>
- Nusrat, I., & Jang, S.-B. (2018). A comparison of regularization techniques in deep neural networks. *Symmetry*, 10(11), 648. <https://doi.org/10.3390/sym10110648>
- Parmar, I., Agarwal, N., Saxena, S., Arora, R., Gupta, S., Dhiman, H., & Chouhan, L. (2018). Stock market prediction using machine learning. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)* (pp. 574-576). IEEE. <https://doi.org/10.1109/ICSCCC.2018.8703332>
- Sarode, S., Tolani, H. G., Kak, P., & Lifna, C. S. (2019). Stock price prediction using machine learning techniques. In *2019 International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 177-181). IEEE. <https://doi.org/10.1109/ISS1.2019.8907958>
- Saud, A. S., & Shakya, S. (2019). Analysis of gradient descent optimization techniques with gated recurrent unit for stock price prediction: A case study on banking sector of Nepal stock exchange. *Journal of Institute of Science and Technology*, 24(2), 17-21.
- Saud, A. S., & Shakya, S. (2020). Analysis of lookback period for stock price prediction with RNN variants: A case study on banking sector of NEPSE. *Procedia Computer Science*, 167, 788-798.
- Shen, G., Tan, Q., Zhang, H., Zeng, P., & Xu, J. (2018). Deep Learning with gated recurrent unit networks for financial sequence predictions. *Procedia Computer Science*, 131, 895-903.
- Shen, J., & Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, 7(1), 66. <https://doi.org/10.1186/s40537-020-00333-6>
- Vanstone, B., & Finnie, G. (2009). An empirical methodology for developing stock market trading systems using artificial neural networks. *Expert Systems with Applications*, 36(3), 6668-6680.
- Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques. *Procedia Computer Science*, 167, 599-606.
- Weiss, G., Goldberg, Y., & Yahav, E. (2018). On the practical computational power of finite precision RNNs for language recognition. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* (Vol. 2, pp. 740-745). Melbourne. <https://doi.org/10.18653/v1/p18-2117>
- Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., & Deng, S.-H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and*

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168, 022022. <https://doi.org/10.1088/1742-6596/1168/2/022022>

Zhao, J., Zeng, D., Liang, S., Kang, H., & Liu, Q. (2021). Prediction model for stock price trend based on recurrent neural network. *Journal of Ambient Intelligence and Humanized Computing*, 12(1), 745-753.