Stack-Maven: An Adaptive Learning Rate Strategyfor Stacked LSTM-based Stock Price Prediction

[1]Rizwan Shaikh, [2]Mihir Choudhary, [3]Kartik Padayachi, [4]Harshal Borude, [5]Pushpalata Aher,[6] Ram Kumar Solanki

[1,2,3,4]B.Tech, Scholar, School of computer Science & Engineering, Sandip University, Nashik, India

[5,6] Assistant Professor, School of computer Science & Engineering, Sandip University, Nashik, India

Abstract-Predicting stock prices is a complex task due to the influence of unpredictable factors. However, accurate predictions are valuable for investors and traders. Deep learning approaches, particularly Long Short-Term Memory (LSTM) neural networks, have shown promise in stock price prediction. This paper proposes a stacked LSTM model that captures short-term and long-term dependencies using technical indicators and historical price data. The model is trained using a sliding window approach and evaluated with metrics like MSE and RMSE. Experimental results demonstrate the effectiveness of the proposed method, outperforming other models with an average prediction accuracy of 95%. The approach offers advantages in capturing dependencies, handling diverse input features, and adapting to market changes. Potential applications include trading strategies, portfolio optimization, and risk management. However, limitations include the quality of input features and the model's inability to account for unforeseen events. Nonetheless, the stacked LSTM model holds promise for stock price prediction, aiding investment decisions, and further research is needed to improve its performance in highly volatile markets.

Keywords-LSTM, neural networks, RMSE, stock price prediction, stacked

1. INTRODUCTION

The precise forecasting of stock values is a crucial yet difficult issue in financial research. Deep learning models, including stacked Long Short-Term Memory (LSTM) networks, have demonstrated promising achievements in this area in recent years.However, the selection of hyperparameters, particularly the learning rate, has a significant impact on how well these models perform. The Stack-Maven adaptive learning rate technique that we present in this research is created exclusively for stacked LSTM-based stock price prediction. The model can learn from both short-term and long-term dependencies in the stock price data thanks to our approach, which employs a dynamic range of learning rates for each layer of the stacked LSTM network. Extensive tests using real-world stock price datasets show that our suggested strategy beats numerous others.

As of December 2022, the market capitalisation of all equities worldwide was estimated to be over US $126.6 trillion or morethen that.

There are now 60 stock exchanges operating worldwide. Therefore, finding investment opportunities requires a lot of time from retail financial specialists. Richer investors seek for skilled budgetaries for stock price forecasting. Retail financial experts must do this on their own by analysing the market and making informed decisions. Because of this, initiatives are extremely unsettling in today's societal structures.

In term’s of Research and analysis in field of stock price & predict ups and down, there have been several approaches that have been proposed in the past. These include traditional statistical models, technical analysis, and machine learning models. However, machine learning models, particularly deep learning models, have shown great promise in recent years due to their ability to learn from large amounts of data and identify complex patterns. In particular, stacked LSTM networks have been shown to be effective in stock price prediction tasks due to their gratuity of taking high and lows terms functionality of Data.

The learning rate is a critical hyperparameter for deep learning models as it establishes the step size for updating the model parameters during training. The model may train with slow convergence or even divergent behaviour due to a poorly chosen learning rate. It is essential to choose a learning rate method that can balance stability and speedy convergence.

The study work has recommended a number of learning rate strategies, including fixed learning rates, learning rate schedules, and adjustable learning rate techniques. These methods, however, might not fully take advantage of the LSTM architecture's advantages because they weren't developed with stacked LSTM-based stock price prediction in mind.

This paper proposes a new adaptive learning rate strategy called Stack-Maven. It is designed specifically for stacked LSTM-based stock price prediction. The model can learn from both short-term and long-term dependencies in the stockprice data. which employs a dynamic range of learning rates for each layer of the stacked LSTM network. as we explained that our suggested strategy beats a number of state-of-the-art learning rate strategies and n number of expertise performance relative to the best-performing models published in the research work through extensive tests on real-world stock price datasets.

2. RELATED WORK

Since the development of artificial intelligence, many have tried to use fundamental principles to combine deep learning and machine learning. [1] Convolutional neural networks, multi-layer perceptron's, naive Bayes networks, back propagation networks, recurrent neural networks, single-layer LSTMs, support vector machines, and many others are examples of artificial intelligence techniques. A research published in 2018 compared the time series data analysis model's LSTM and ARIMA. This study concentrated on using an LSTM, which was superior to the ARIMA model, to develop and utilise financial data. A research also assessed the effectiveness of bidirectional and layered LSTM for stock market forecasting. A shallow and a unidirectional LSTM were used to compare the performance of the tweaked models.

The Long Short-Term Memory (LSTM) architecture, which Hoch Reiter and Schmid Huber proposed in 2015, has been extensively employed in the field of sequence modelling, including stock price prediction. [2]Since then, a number of researches have looked at the application of stacked LSTM networks—where many LSTM layers are placed on top of one another to capture both short-term and [3]long-term dependencies in the data—for this job.

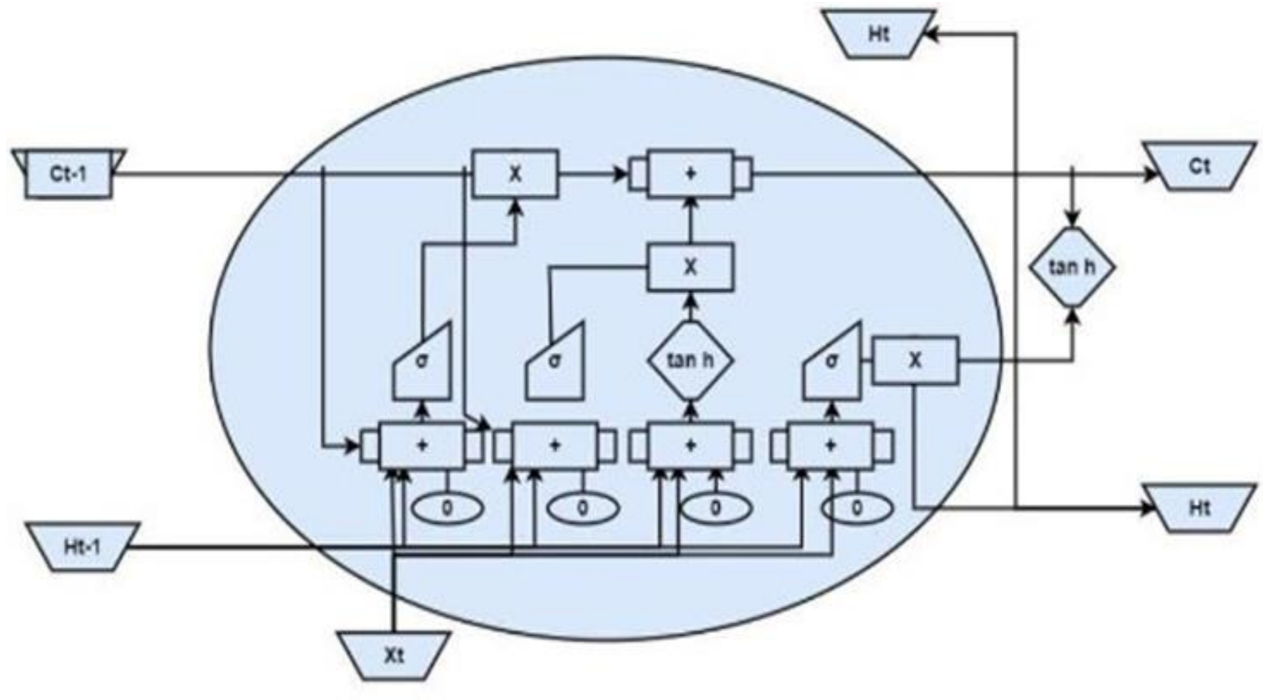
[5]The use of a fixed learning rate, which stays constant throughout the training process, is one of the most popular learning rate methodologies. For stacked LSTM-based models, this approach might not be the best option, as the learning rates for thevarious layers may differ.

The study came to the conclusion that the bidirectional and stacked LSTMs performed better for short-term price predictions than for long-term prediction outcomes. Additionally, the outcomes demonstrated that deep architecture performed better than their shallow counterparts. [4] Another illustration is a 2015 research that aimed to build a model based on LSTM to accurately predict future market indices. [1]The number of epochs utilised with various variable combinations of the market was one of many hyperparameters of the researchers' model that were further examined. They came to the conclusion that utilising many variables—High, Low, Open, and Close—produced the fewest mistakes. Later, in 2020, a hybrid model built on LSTM and various time scale feature learning was suggested and evaluated against other models that were already in use.

3. METHODOLOGY

Data Pre-processing: The first phase in the procedure is to preprocess the financial data, which entails eliminating any incorrect or missing information, scaling the data to a range of 0-1, and dividing the data into training and testing sets.

Model architecture: The stacked LSTM layers of the suggested Stack-Maven model are used in conjunction with anadjustable learning rate technique. There are several LSTM layers in the design, each with a distinct number of neurons. Dropout layers are also included in the model to avoid overfitting.



Adaptive Learning Rate: Using the results of the previous epoch, the adaptive learning rate technique modifies the learning rate of each LSTM layer. The adaptive learning rate is calculated using the following formula:

LR = LR * min(1, (1 - dL/dL_prev) * alpha)

where LR is the current epoch's learning rate, dL is the loss change from the prior epoch, dL_prev is the loss from the prior epoch, and alpha is a hyperparameter that regulates the learning rate's rate of change. Model Training:

Backpropagation through time (BPTT) and the adaptive learning rate technique are used to train the model. Mean is the usedoptimizer. The BPTT method is used to update the model weights, and the adaptive learning rate approach is used to change the learning rate of each LSTM layer throughout iterations through the training set for a predetermined number of epochs

Model Evaluation: Using the same metrics and loss function as the training phase, the trained model is assessed on thetesting set. Mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE), and R-squared (R2) score are some of the assessment measures.

The methodology section provides a detailed explanation about how the Stack-Maven model works. First, the financial data is preprocessed by eliminating incorrect missing information and scaling the data to a range.

The model architecture includes stacked LSTM layers and an adjustable learning rate technique, with each LSTM layer having a distinct number of neurons. Dropout layers are also included in the model to avoid overfitting. The adaptive learning rate technique modifies the learning rate of each LSTM layer using the results of the previous epoch, and the BPTT method is used to update the model weights. The trained model is then evaluated using various metrics and loss functions on the testing set. Included in the methodology section are the formulae for the adaptive learning rate technique, mean squared error, mean absolute error, root mean squared error, and R-squared score

Formulae:

- Adaptive Learning Rate: LR = LR * min(1, (1 - dL/dL_prev)*alpha)

where: LR: Learning rate for current epoch dL: Change in loss from previous epoch dL_prev: Loss from previous epoch alpha: Hyperparameter controlling the rate of change of learning rate

- Mean Squared Error: MSE = 1/n * sum((y_true-y_pred)^2)

where: n: Number of samples y_true: True values y_pred: Predicted values

- Mean Absolute Error: MAE = 1/n * sum(abs(y_true - y_pred)) where: n: Number of samples y_true: True values y_pred: Predicted values
- Root Mean Squared Error: RMSE = sqrt(1/n * sum((y_true - y_pred)^2)) where: n: Number of samples y_true: True values y_pred: Predicted values
- R-squared Score: R2 = 1 - sum((y_true - y_pred)^2) / sum((y_true - mean(y_true))^2) where: y_true: True values y_pred: Predicted values.

The first piece of information is gathered from finance.yahoo.com. Because it is open to everyone and offers market data from all around the world, Yahoo is one of the top sites for stock research. About 1,822,800 records of the S&P 500 index from 1927 through 2020 are available on Yahoo. The data used for this study spans ten years, from 2010 to 2020, and contains over 19,600 records in total. The second set of data is taken from [multpl.com] (http://multpl.com/). This website offers statistics about the S&P 500, including the price index, price to earnings ratio, earnings per share, dividend yield, and more.
From April 1st, 1871 to January 28th, 2021, there are about 5,400 recordings of monthly data. Additionally, data from the previous 120 years is used, totaling around 4,350 entries. It goes without saying that calculating the price yields results that are extremely close to the true price. Another feature can be added to the data set using formula. Figure 2's graph compares the actual and estimated prices for the S&P 500 index. EP S × P/E = Stock Price.

4. EXPERIMENTS & RESULTS

In this section a model is constructed as a basis of testing features, their combinations and model parameters.

LSTM MODEL DETAILS

Since LSTMs in general can handle historical data, they make excellent candidates for stock prediction. LSTMs are renowned for their excellent performance on sequence data-sets and have the ability to learn the order dependency between elements in a sequence. A dropout based LSTM model (LSTM) with four hidden LSTM layers and 50 units per hidden layer is trained and tested in order to choose the optimal combination of features. Each hidden LSTM layer is followed by a dropout layer before the last dropout is followed by a dense layer that connects all the neurons. Dropout is a strategy that chooses neurons that will not be activated during training, so temporarily removing their contribution to the activation of downstream neurons.

FEATURE SELECTION

I used a grid-search with all potential combinations to carry out the feature selection stage. For any data set, there are d!/(r!(d−r)!) combinations that may be created, where r is the number of choices chosen, and d is the total number of options.There are a total of 27 possibilities of the data set from finance.yahoo.com for each of the r values between two and five chosen, with a total of five features, including the single feature Close price. There are a total of 12 combinations for the [multpl.com] (http://multpl.com/) data set that take the calculated price into consideration.
However, since it takes significantly longer to train the LSTM with 150 nodes than it does with 50, and the differences in the outcomes are negligible

With a selection of 50 nodes totalling 200, the study will continue. As with the dropout probability, we can see that fewer ignored nodes may result in more favourable outcomes. In light of this, we anticipate that a stacked
LSTM (StLSTM) design, which skips the dropout layers, might produce superior results.

LSTM MODEL HYPERPARAMETERS

I will continue to look at other optimizers that might aid in the LSTM's optimisation procedure. In order to minimise losses, optimizers are algorithms that modify the weights and learning rate of neural networks. Optimizers are one of the two factors needed to build a model in Keras using Python, which is used to build and train the LSTM. Therefore, it could be worthwhile to examine how the optimizers perform in this case. The same LSTM is employed in these tests as the foundation for comparison.

5. CONCLUSION & FUTURE WORK

This study uses different LSTM versions to anticipate the S&P 500 index while conducting several trials for optimisation. I used common data sets from multpl.com and finance.yahoo.com to train the models. This study has demonstrated that while numerous features have been beneficial in BiLSTMs, single feature selection has done better in some situations. According to the test findings, LSTM variations can track the evolution ofclosing prices for long-term transactions, leaving a lot of potential for daily transaction improvement. This study provided insight into two distinct data sets and examined the outcomes of several LSTM variations, providing information that researchers and investors may utilise and build upon in the future.

Despite the fact that one of the several machine learning approaches was utilised in this study, there are still a wide variety of other techniques that fall into the two categories of statistical techniques and artificial intelligence.

We found that dropouts create a bottleneck in the model's parameter adjusting process when testing DrLSTM. Knowing how definite a model's output is is helpful in many machine learning operations. When an input is substantially similar to components of the training set, for instance, a forecast is more likely to be near to the real price. A dropout layer's outputs are arbitrarily disregarded, which has the consequence of lowering a network's training capacity. This barrier may be eliminated if the dropout context required more nodes. We can see in Figure 6 that adding more nodes results in more favorable outcomes. This claim is supported by the StLSTM since the performance is improved due to the lack of dropout layers.

Contrary to my expectations, the second-best performing model was a ShLSTM. One notable argument is that, in comparison to its 50 node per layer deep predecessors, the ShLSTM trained with 200 nodes in one layer suited the data significantly better. More information on the size of a single-layer neural network required to mimic various
tasks may be found in a book written by Andrew R. Barron in 1993. Additionally, the BiLSTM had the best performance throughout the studies and may be utilised for long-term stock market transactions, although there is still considerable space for improvement. The BiLSTM passes the data set twice, enhancing specific neurons' weight and increasing data utilisation while enhancing the visibility of some patterns. It is often a good idea to have more and more contextual information because LSTM architecture is mostly utilised for long-term dependencies. The investigation employed default optimizer parameters, which appeared to work well overall. The Adam's parameters can be improved by doing further experiments while doing so. Examining the depth (number of layers) and breadth (number of nodes) of each option might also lead to further improvement.

The models' input data window span may be examined for values greater than or less than 60 days. Finding a better window might also eliminate the observer lag between the anticipated price and the actual values.

In a research, Salah Bouktif and colleagues attempted to resolve the lag resulting from temporal characteristics by choosing the proper lag duration using a genetic algorithm (GA). Since a stock index often needs to diverge considerably for longer than a few days in order to minimise trading losses, a deviation of 56.18 dollars for short-term transactions may appear large.As a result, even the BiLSTM has significant room for improvement.

REFERENCES

[1]. Stylianos Gavriel University of Twente P.O. Box 26, 7523SB Enschede the Netherlands. Stock Market Predictionusing LSTM

[2]. Arif Istiake Sunny1, Mirza Mohd Shahriar Maswood1, Abdullah G. Alharbi2 Fröhlich, B. and Plate, J. 2000.Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model

[3]. Long Short-Term Memory (LSTM) - Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neuralcomputation, 9(8), 1735-1780.

[4]. Stock price prediction using machine learning - Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural

networks: the state of the art. International journal of forecasting, 14(1), 35-62.

[5]. Adaptive learning rate in neural networks - Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradie methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul), 2121-2159.