Prediction of Financial Stock Price using ensemble deep learning models

A Project Report Submitted by

Mayank Gupta

in partial fulfillment of the requirements for the award of the degree of

Master of Technology(DCS)

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

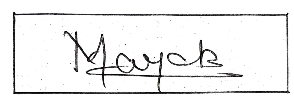
Indian Institute of Technology Jodhpur

Department of Mathematics

May, 2022

Declaration

I hereby declare that the work presented in this Project Report titled *Prediction of Financial Stock price using ensemble deep learning models* submitted to the Indian Institute of Technology Jodhpur in partial fulfilment of the requirements for the award of the degree of Master of Technology, is a bonafide record of the research work carried out under the supervision of Dr. V.V.M.S. Chandramouli. The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad for the award of any degree or diploma.



Signature

Mayank Gupta

M20MA005

Certificate

This is to certify that the Project Report titled *Prediction of Financial Stock price using ensemble deep learning models*, submitted by Mayank Gupta(M20MA005) to the Indian Institute of Technology Jodhpur for the award of the degree of Master of Technology, is a bonafide record of the research work done by him under my supervision. To the best of my knowledge, the contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Signature

Dr. V.V.M.S. Chandramouli

Acknowledgement

First of all, I convey my sincere gratefulness to the Department of Mathematics of Indian Institute of Technology Jodhpur, Rajasthan for giving me a chance to work out on my final year project work. I would also like to acknowledge with thanks the kind of patronage, loving, Inspiration, responsibility and timely guidance to my guide, Dr. V.V.M.S. Chandramouli, Assistant professor, IIT Jodhpur to work on this project. With immense pleasure, I express my heartfelt thanks to my parents and my friends who in one way or another shared their support and encourage me every time. Above all I thank God, the author of knowledge and wisdom for his countless love, giving me strength and always there for me.…

Abstract

Stock Price is highly volatile value and predicting it for future is difficult as it depends on several factors like social media, market crash, country's economy etc. It continues to remain as a challenging problem in machine learning. The noise and volatility of stock prices are not constant. Stock prices suddenly rise and fall but if rises, returns are very high. This has created a need to predict the stock price which gives an idea when to invest the money so as to get good returns. Stock prices are time series data and time series models like: ARIMA should work well but here our concern is about ensemble deep learning models.

In this project, We propose an ensemble model of CNN-BiLSTM and a modified Transformer (incorporating temporal effect of stocks) which will be predicting stock prices of two IT companies: Amazon and Facebook. Performance metric taken are Root Mean Square Error(RMSE) and Mean Absolute Error (MAE). We have done analysis on ensemble CNN-BiLSTM and on Transformer model. The performance has been analysed by finding the RMSE and MAE scores for each of the above models and concluded that minimum score in both the metric has been obtained in Transformer model.

Contents

List of Figures

List of Symbols

This list describes several symbols that will be later used within the body of the document

AI Artificial Intelligence

$CNBC$ Consumer News and Business Channel

CNN Convolutional Neural Network

$DP-LSTM$ Differential Privacy Long Short Term Memory

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

$LSTM$ Long Short Term Memory

$MAPE$ Mean Absolute Percentage Error

MCC Matthews Correlation Coefficient

ML Machine Learning

NLP Natural Language Processing

NSE National Stock Exchange

$NYSE$ New York Stock Exchange

$RMSE$ Root Mean Square Error

RNN Recurrent Neural Network

$SOFNN$ Self Organised Fuzzy Neural Network

WSJ Wall Street Journal

Prediction of Financial Stock Price using ensemble deep learning models

1 Introduction

A number of stocks are sold by company and among them stock price is the price of a single stock. Usually most of the people buy stocks of the companies so as to get some profit. Studying Stock price is a challenging task as the price depends on demand and supply and finding what factors are influencing demand and supply is difficult. Predicting the trend of stock (going up/down) is relatively easy and could be easily done by technical analysis and indicators. These factors mostly revolves around common issues in our society including market behaviour, company's positive and negative news etc. Hence forecasting stock price is renowned problem statement which every investor wants to solve accurately.

As we know a lot of trading happens everyday and every one wants to get profit on his/her investment and prior prediction of stock price of that stock would be useful in deciding buying or selling which in turn would help in profit. Many ML and AI techniques have been implemented to predict stock price but ensemble models have shown better performance as compared to individual ML models. Deep learning ensemble models are hybrid models by mixing more than one model with each other so as to have better performance. These models usually consists of series of neural networks which consists of neurons with activation function. Apart from existing deep learning models, we have used the most latest deep learning model i.e. Transformer. Transformers have shown good results in NLP related problems but they can also be used in stock price prediction as they can incorporate the temporal effect.

2 Literature survey

A lot of relevant research has been done on predicting stock prices and surveying them gave an idea of what current research is going on and what additional needs to added. Following are the papers that have been surveyed:

1. NSE Stock Market Prediction Using Deep-Learning Models *(Hiransha M , Gopalakrishnan E.A , Vijay Krishna Menon, Soman K.P)*

This paper mostly deals with various deep learning models and their results on NSE and NYSE stocks. Authors have used Multilayered perceptron, RNN, CNN, LSTM models for prediction. Training data is of 19 years with NSE data of Tata motors with daily closing price. Testing data was taken for 5 years for 2 different stocks of NYSE. Finally the model was able to predict the NYSE stock even though it was trained for NSE. The paper showed the ability of deep learning models to retain the property even though the dataset changes.

2. Carbon futures price forecasting based with ARIMA-CNNLSTM model *(Lei Ji, Yingchao Zou , Kaijian He, Bangzhu Zhu)*

This paper shows the application of ensemble ARIMA-CNN-LSTM deep learning model in carbon price prediction. Authors have used weekly future prices of EU emission trading system of 11 years with 70% train and rest 30% as test. The final ensemble model performs better than the individual model with RMSE and MAPE as performance metrics.

3. A novel ensemble deep learningmodel for stock prediction based on stock prices and news *(Yang Li, YI Pan)*

The authors have proposed a novel ensemble model of LSTM and GRU and news score for stock price prediction. The hybrid model has been trained with SP 500 data and some popular news sources like: CNBC, Reuters, WSJ, etc. and it gives better performance in terms of MSE as compared to existing ones(LSTM, GRU, DP-LSTM). The model has also been used for classifying the price trends(increase or decrease).

4. A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing *(Sidra Mehtab, Jaydip Sen)*

The authors have proposed a hybrid approach involving NLP, machine learning and deep learning. The authors have predicted the future stock price for NIFTY50 dataset for period of three years and the predictive model along with sentiment analysis of twitter data is fed to SOFNN to predict the future prices. Price movement and prediction has been done using various classification and regression models. Performance metrics used are: classification accuracy for classification and MAPE(Mean Absolute Percentage Error) for Regression. For regression LSTM and Random forest for classification performs the best.

5. Transformer-Based Capsule Network For Stock Movements Prediction *(Jintao Liu, Xikai Liu, Hongfei Lin, Bo Xu, Yuqi Ren, Yufeng Diao,Liang)*

Th authors have proposed a novel capsule network for stock price movement. The authors have used CapTE model which captures the semantic information from social media using Encoder. Performance metrics used is Matthews Correlation Coefficient (MCC) and the dataset is taken from Standard Poor's 500 list. The CapTE model gave 0.3481 MCC.

6. Time2Vec: Learning a Vector Representation of Time *(Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota,Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, Marcus Brubaker)*

The authors have proposed a novel method for learning the time embeddings in transformer model. This time2vector concent can be leveraged in various applications like stock price prediction. The time2vec approach divides the time into 2 components: periodic and non periodic. The same concept has been used in our work.

3 Problem Statement

The work presented in this report is based on Prediction of Stock price using ensemble CNN-BiLSTM and a modified Transformer model. The standard models (CNN and BiLSTM) have been included so as to compare the performance. Ensemble models have been used by researchers and still work is in progress. These ensemble models are used to obtain performance of the model but the performance increases upto certain extend and then stagnates. This limitation we are trying to address using latest Transformer model which has shown its spectacular performance in building tough NLP related use cases.

The standard Transfomer has been modified so that it can work for this regression task. Normally the Trasnformer consists of Encoder and Decoder but we have removed the decoder part and getting predictions directly from Encoder part. We have discussed about these models in detail in the next section.

4 Implementation

4.1 Facebook/Amazon loading dataset and Pre-processing

The initial analysis started with the facebook/Amazon dataset. The data has been downloaded using yfinance library. This library directly downloads data from yahoo finance. The data is taken from August 2014 to December 2021 on daily basis. For both the proposed models, train set taken is from August 2014 to February 2020 while the test set is the remaining year. There are total 1858 data points.

Since market remains close on Saturday and Sunday so their are missing values on these days. For pre-processing we have simply ignored the two days value and focused on Monday to Friday. The historical data consists of High, Low, Volume, Date, Open, Adjusted Close. The different stock prices here refers to Adjusted close price. We selected Adjusted close price because this gives the most accurate price after considering any corporate actions. The following are the snapshot of the dataset downloaded along with the Facebook and Amazon stock price variation.

Date	Open	High	Low	Close	Adj Close	Volume
2014-07-29	74.720001	74.919998	73.419998	73.709999	73.709999	41324000
2014-07-30	74.209999	75.190002	74.129997	74.680000	74.680000	36853000
2014-07-31	74.000000	74.169998	72.440002	72.650002	72.650002	43992000
2014-08-01	72.220001	73.220001	71.550003	72.360001	72.360001	43535000
2014-08-04	72.360001	73.879997	72.360001	73.510002	73.510002	30777000

Figure 4.1: Sample of facebook dataset

	Open	High	Low	Close	Adj Close	Volume
Date						
2014-07-29	321.980011	322.899994	319.500000	320.000000	320.000000	2883800
2014-07-30	321.450012	322.730011	318.500000	322.510010	322.510010	3969000
2014-07-31	320.010010	320.679993	311.859985	312.989990	312.989990	5192000
2014-08-01	313.690002	315.829987	304.589996	307.059998	307.059998	7441500
2014-08-04	308.839996	316.179993	308.500000	313.649994	313.649994	4200900

Figure 4.2: Sample of Amazon dataset

Since our work is based on ensemble models, the implementation started with ensemble model of CNN-BiLSTM with stock prices of facebook and Amazon. The input is taken by CNN and further Bi-LSTM layers are added to give the final output and this ensemble model improves the performance.

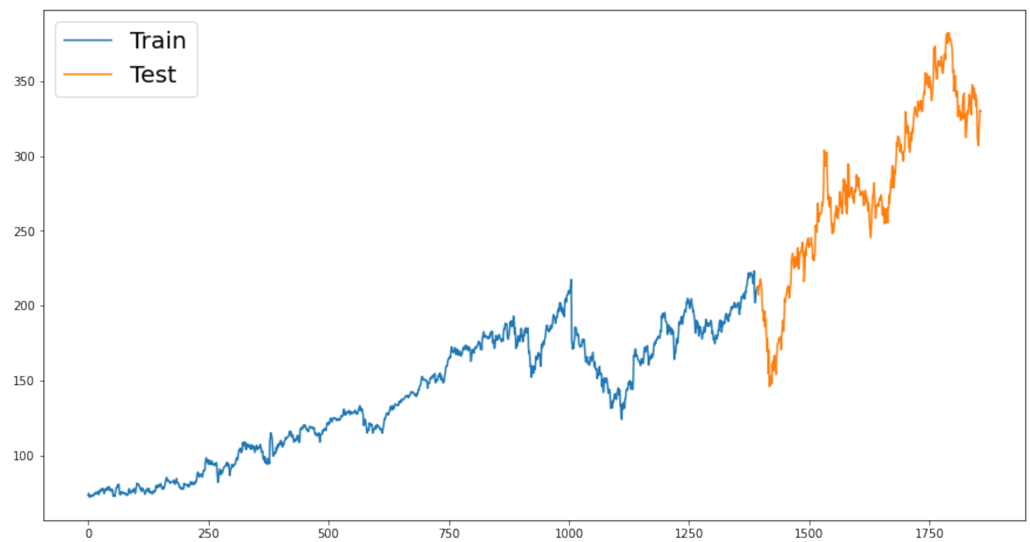


Figure 4.3: Facebook Stock Price variation

In time series data it is necessary to split the data in accordance to their respective date. So we have splitted the data (as mentioned above) in ***train (first 75% and test remaining 25%)***. The data has not been standardized between 0 & 1, we have directly taken the prices for the training. The following is the process being followed for price prediction (Fig 4.5).

Figure 4.4: Amazon Stock Price variation

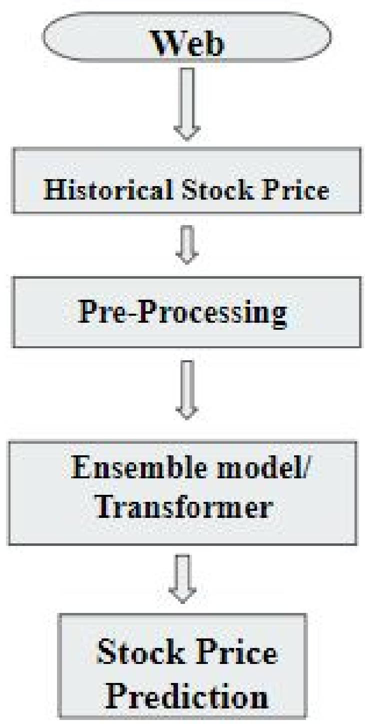


Figure 4.5: Process for Stock price prediction

4.2 Model learning

4.2.1 CNN

CNN is a deep neural network usually used for image classification. This model is also used for forecasting of time series. There are two components of CNN: convolution and pooling layer. In Convolution operation, we take a filter of some size and apply it to input data to obtain convolved feature maps and these features are propagated to the next layer. Pooling layer is used to reduce the size as the convolved features are of high dimension. Pooling reduces complexity and noise of network and trains the model quickly. Since we dont have images in our dataset so we are taking feature map as the output of each conv layer. The input feeds in the first layer of CNN which produces a feature map and with help of feature map the next predictions are made. The following is the architecture of CNN which consists of sequence of conv layers along with pooling layer with fully connected layer at last. This structure is not exactly used in our analysis. For converting CNN architecture for prediction, we have removed the last layer i.e. fully connected softmax layer. The output is simply propagated to the next layers of BiLSTM which gives final prediction. The following is the equation for the output of each neuron in a particular layer.

$$Output = act(w*x + b)$$

where output, act, w, x, b is the outcome after convolution operation, activation function, input vector, weight of the kernel, bias of the network respectively.

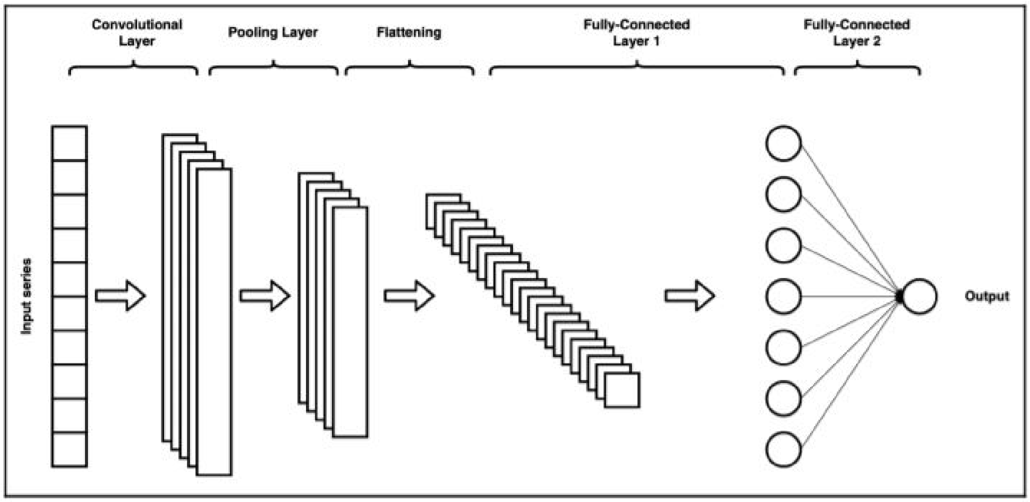


Figure 4.6: CNN architecture for stock prediction[7]

4.2.2 Bi-LSTM

LSTM stands for Long Short Term Memory and they are widely used to perform tasks which are highly dependent on retaining past information. LSTM have proved to be effective for stock market prediction as well. Basically it consists of three gates which are responsible for information flow: input gate, forget gate and output gate. Bidirectional LSTM differs only in one way. It stores the information from future to past as well and thus it handles the time information very well.

Figure 4.7 shows the architecture of LSTM which consists of sequence of LSTM cells. Only one cell is shown and different information gates are marked in the cell along with their equations. There are two different activation functios used in LSTM are: tanh and sigmoid. Sigmoid ensures whether the information is to be passed or not(like a boolean variable yes or no) whereas tanh function ensures that vanishing gradient issue should not arise.

Bi-LSTM is an enhanced version of LSTM. It is forward and backward LSTM which is capable of providing complete context information. Both forward and backward LSTM are connected to the same output as shown in Figure 4.8[6].

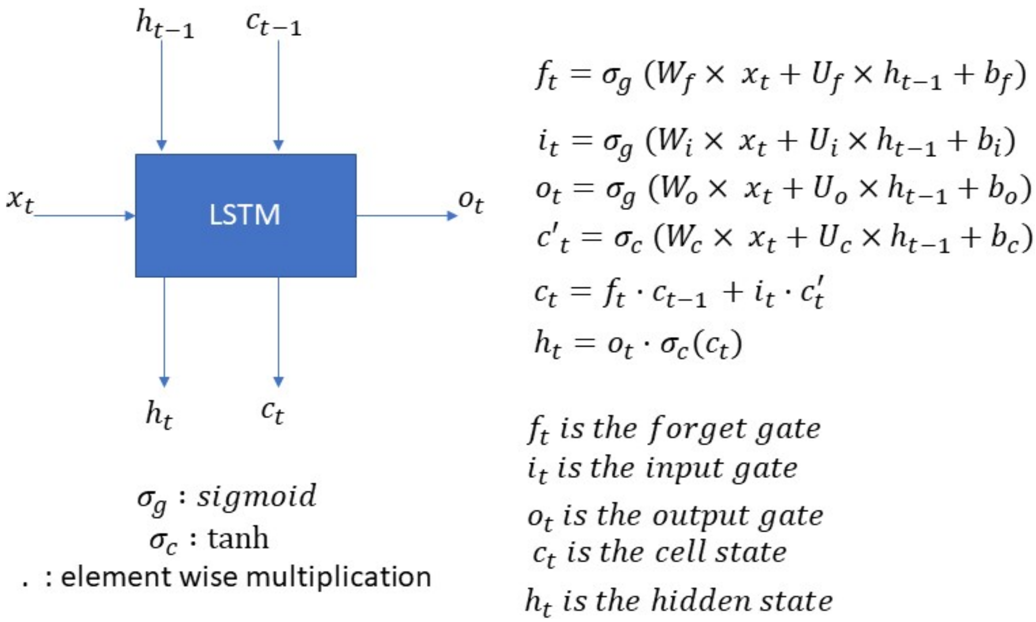


Figure 4.7: LSTM working calculations[7]

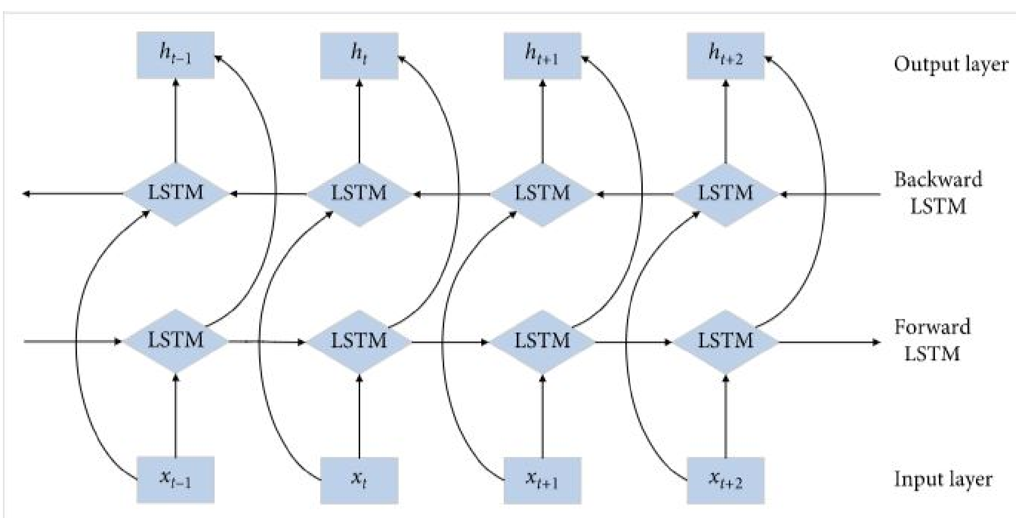


Figure 4.8: BI-LSTM architecture[7]

4.2.3 Transformer

Transformers are one of the most efficient neural network architecture. It has given promising results in NLP related tasks like: Neural Machine Translation, Text Summarization etc. A transformer architecture mainly consists of two parts: Encoder and Decoder. The encoder takes the input and finally the output comes from decoder part. Encoder consists of attention and multi-head attention networks with feed-forward neural network. In our case we are getting output from Encoder itself so we do not need decoder part. The following is the general high level look of Transformer in Machine Translation.

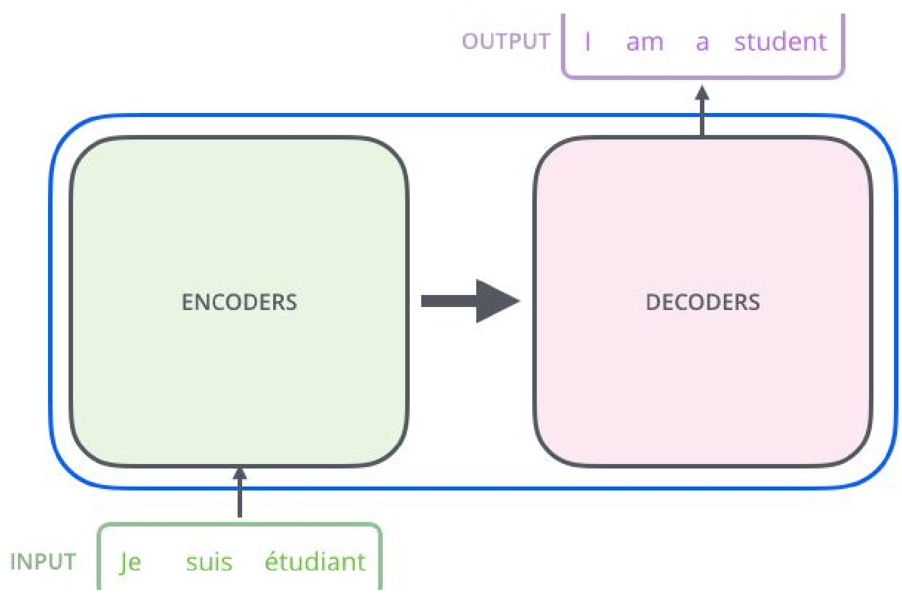


Figure 4.9: Transformer in Machine Translation[**9**]

Encoders: The encoder consists of self-attention layers, Feed forward network along with batch normalization layers. The self attention layer takes input in the form of positional embeddings for NLP related tasks. These embeddings get processed after each layer and finally the output is fed into the decoder part. The following image shows the encoder working.

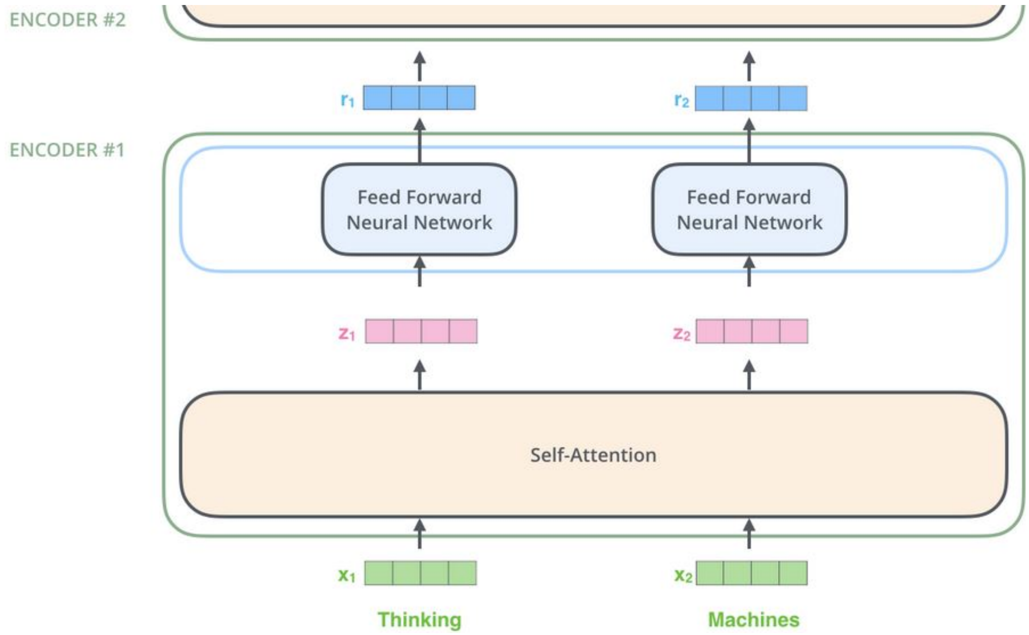


Figure 4.10: Encoder Working[9]

A stock price follows some periodic patterns so it is important to cover the temporal effect for prediction. Time2vector concept has been leveraged to provide this effect. The following equation shows the two components of time2vector.

$$t2v(\tau) = \omega\tau + \phi \text{ for non periodic component}$$

$$t2v(\tau) = f(\omega\tau + \phi) \text{ for periodic component}$$

The first term in the equation represents the non-periodic or linear component. It is simply the output of the neuron $W{*}x + bias$. This can be considered as equation of line $y{=}m{*}x{+}c$ where slope is m. The second component is the periodic or non linear part with activation function(sine). The periodic component can be a particular range of price which repeats over seasons. Non periodic component could be covid disease which occurs in old age people with high chances. These two components captures the temporal effect and helps in efficient stock prediction. Following are the visual representation of both these components.

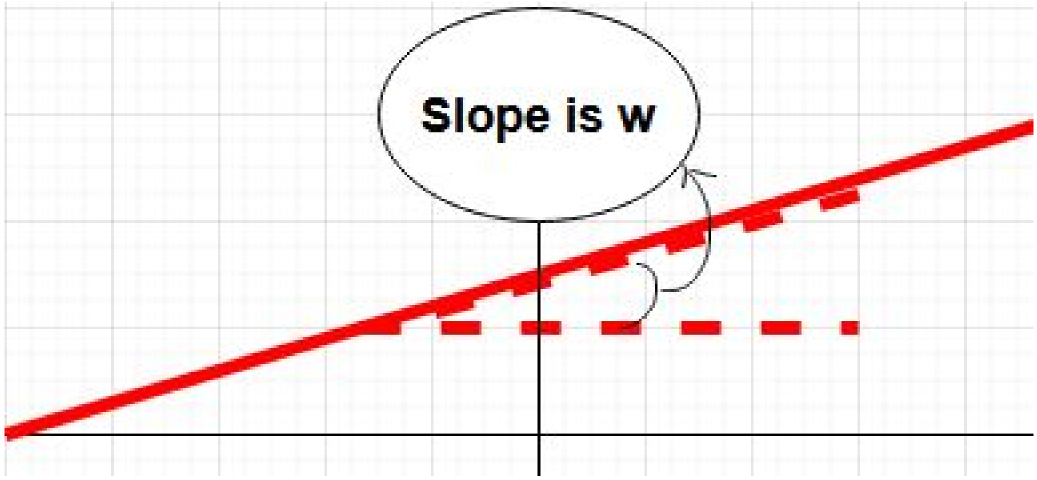


Figure 4.11: Linear component

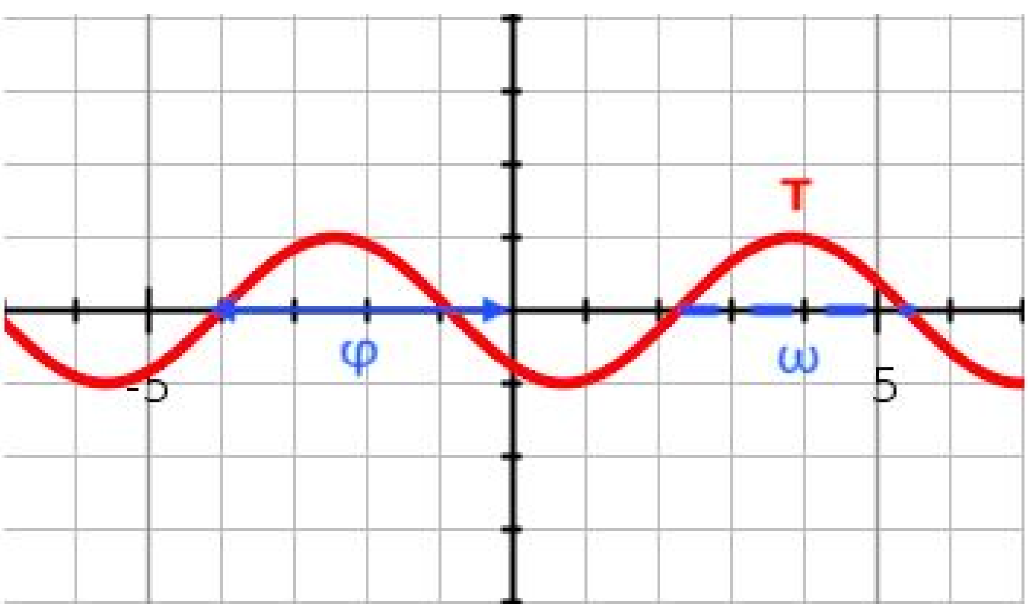


Figure 4.12: Non-Linear component

5 Training

5.0.1 Bi-LSTM

The ensemble model consists of conv layers and Bidirectional LSTM layers. We have kept our model to be small to avoid high complexities in the network. *Two* conv layers with *64 and 50 neurons* along with *relu and relu* activation function. Three Bidirectional LSTM layers each with *50 neurons along with Adam optimizer* have been used. The model has been trained with *15 epochs* for each model. The following image shows the process for Bi-LSTM predictions.

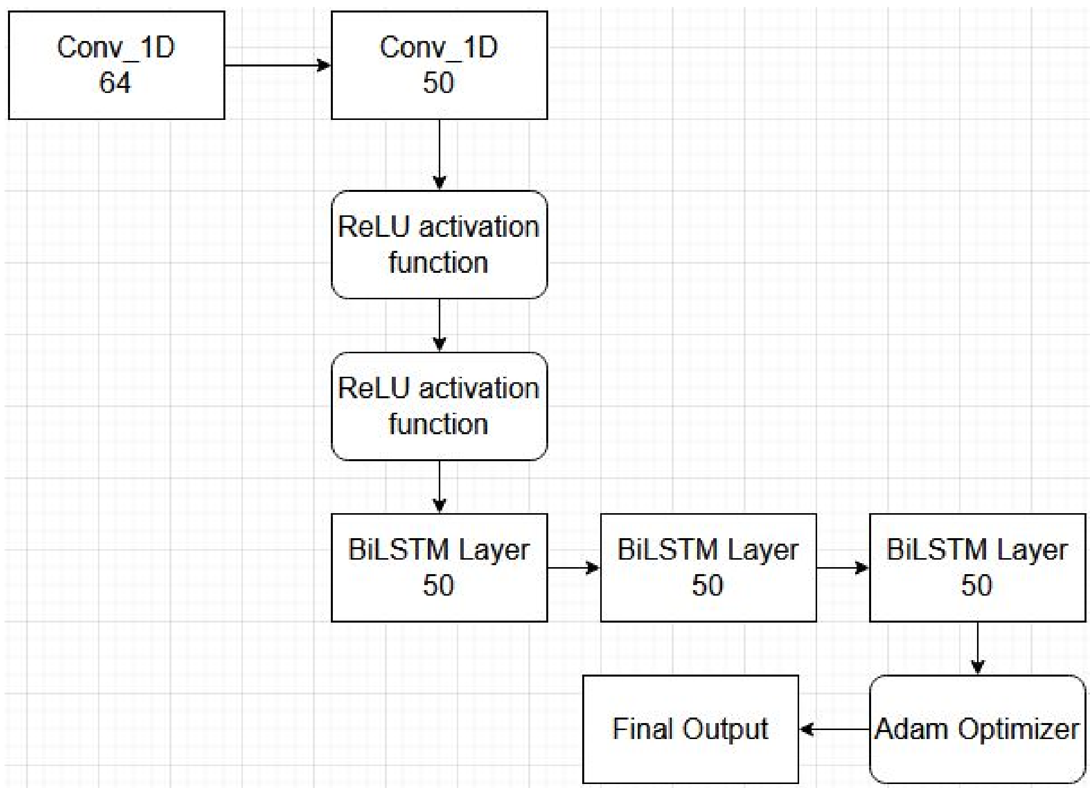


Figure 5.1: Bi-LSTM prediction flow

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 98, 64)            256

 conv1d_1 (Conv1D)           (None, 94, 50)            16050

 bidirectional (Bidirectiona  (None, 94, 100)          40400
 l)

 bidirectional_1 (Bidirectio  (None, 94, 100)          60400
 nal)

 bidirectional_2 (Bidirectio  (None, 100)              60400
 nal)

 dense (Dense)               (None, 1)                 101

=================================================================
Total params: 177,607
Trainable params: 177,607
Non-trainable params: 0
```

Figure 5.2: CNN-BiLSTM Parameters

So we can see that the total parameters both for Amazon and Facebook dataset are same i.e. **177607** and these parameters are trainable parameters. There are zero Non-Trainable parameters. Though these parameters are quite large and it takes around **30 minutes** to train the model on 8GB RAM with intel 5 core processor with Google colab GPU enabled. So the training time taken depends on the system specifications.

5.0.2 Transformer

Input for Self-Attention layer The transformer time embeddings component (periodic and non-periodic) are first created with the help of the equations as described above. These components along with 4 existing features (open,high,low and adjusted close prices) will be taken as input for the single head attention layers to calculate the attention weight using query, key and value.

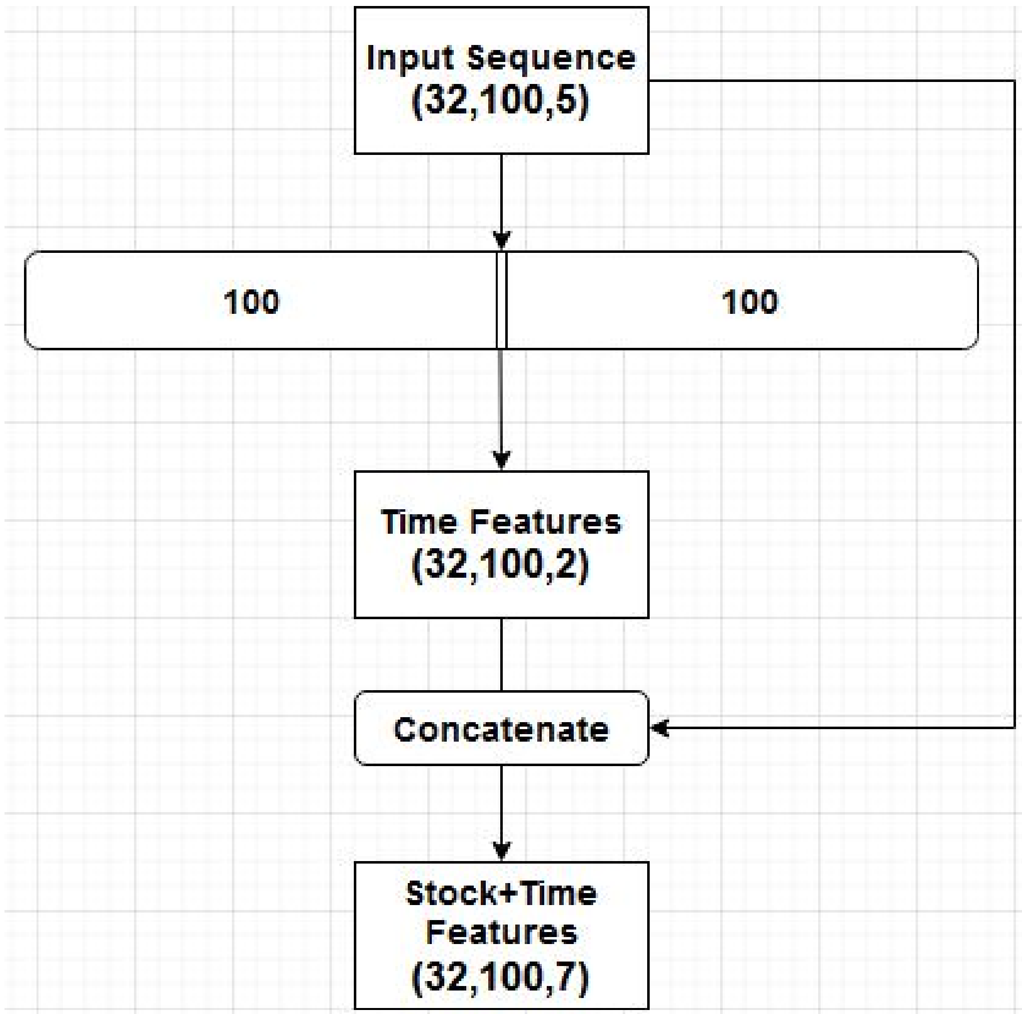


Figure 5.3: Concatenating time features with stock features

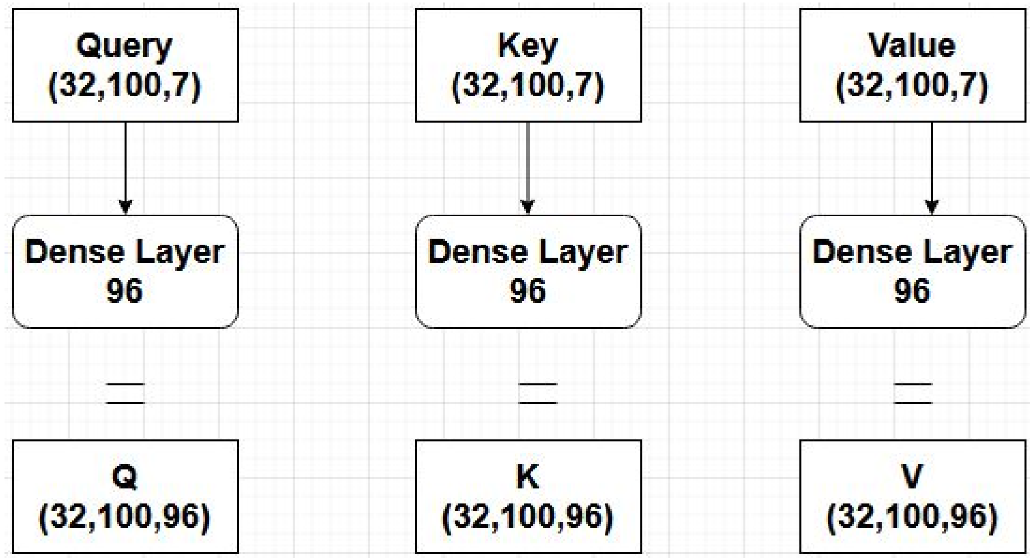


Figure 5.4: Linear Transformation using query, key and value

Single and Multi-head Attention The attention weights help in finding the importance of a specific time series. These values are calculated using dot product of query and key and the resulting answer is divided by dimension of dense layer and resulting value is passed to softmax to find the probabilities and these values are mutiplied with value(V) to find the attention scores After aggregating the attention weights of multiple single attention we have provided multi head attention as well to improve the performance of the model. More the number of attentions heads, more the model is able to incorporate the temporal effect. The single and multi head attentions together forms the basis for input for Encoder layer which consists of two dense layers (with 64 and 1 neuron each) with ReLU and Linear activation function respectively. To avoid overfitting, Global Average pooling and dropout(with 0.1 probability value) has been used. The model has been trained with **15 epochs**. The training did not require high GPU enabled device and could be done easily with the help of free version of Google colab.

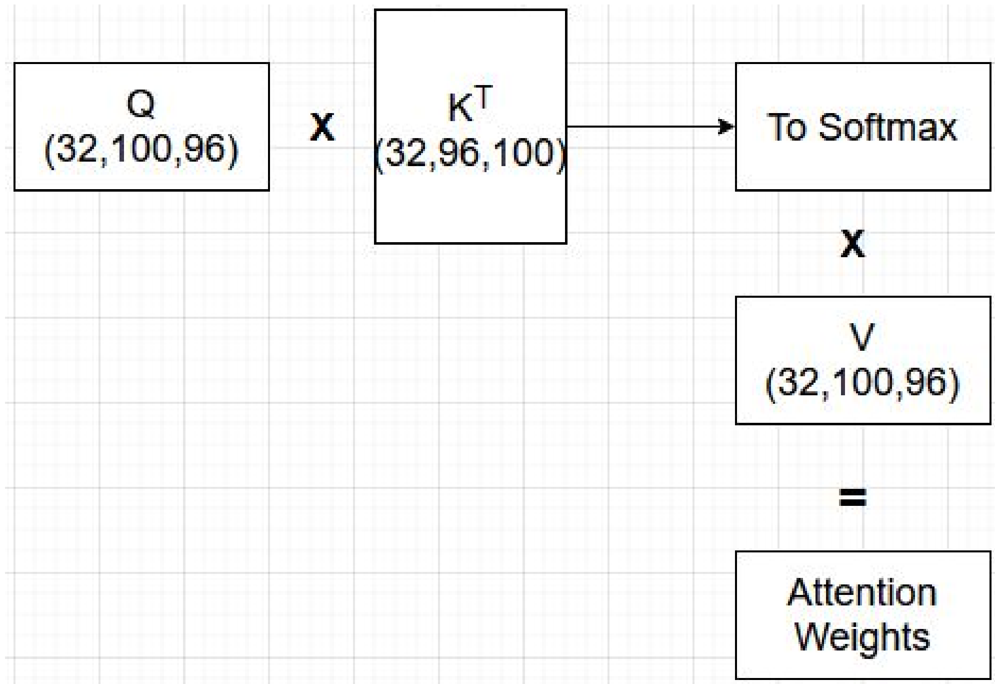


Figure 5.5: Calculations for Attention score

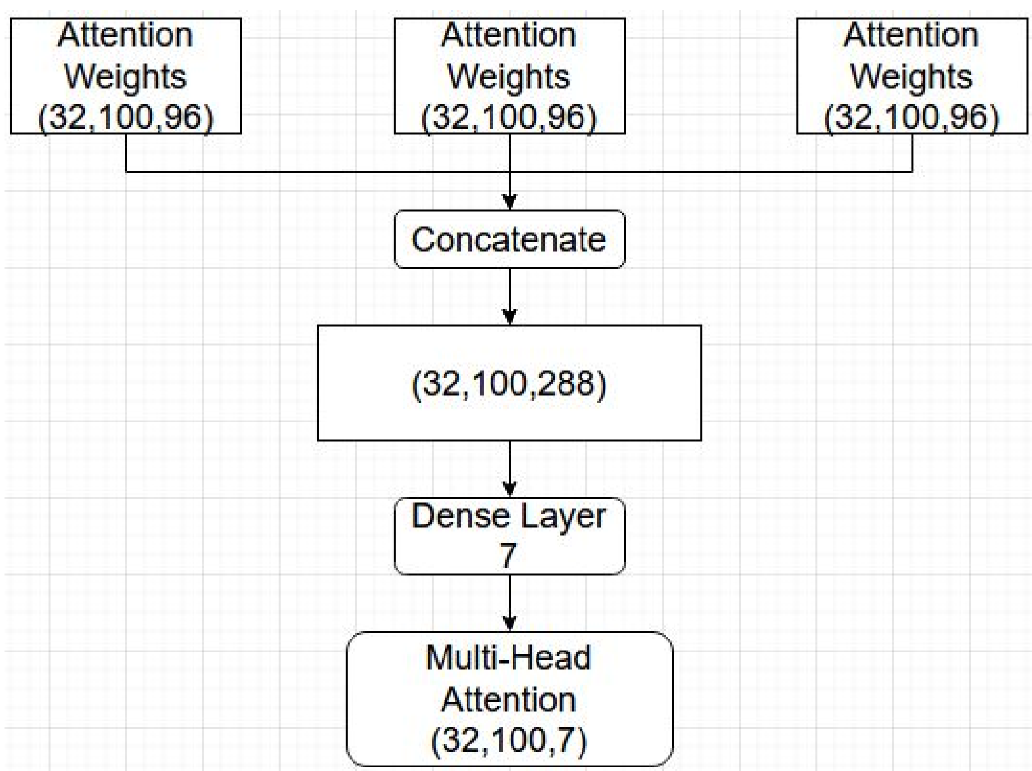


Figure 5.6: Multi-Head Attention Layer

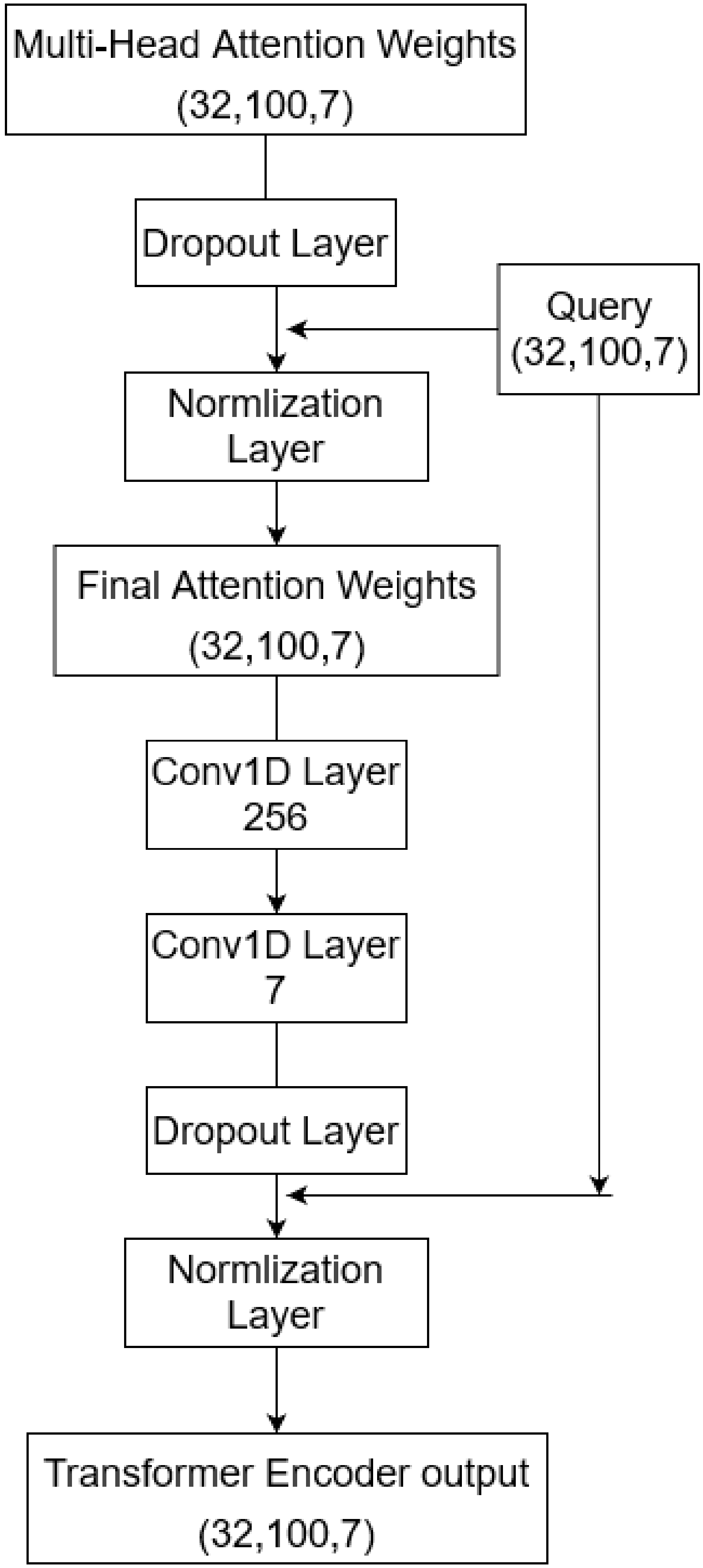


Figure 5.7: Transformer Encoder Layer

6 Performance metrics

Since our problem statement is based on prediction of stock prices so it is a regression problem and thus we have selected 2 metrics for this: Root Mean Square Error(RMSE) Mean Absolute Error(MAE). Since RMSE is sensitive to outliers so we have selected other metric as well. The formulas for both the metrics is given by:

$$\sum_{i=1}^{D} |(x_i - y_i)/n| \text{ —-> Mean Absolute Error}$$

$$\sqrt{(}\sum_{i=1}^{D}(x_i - y_i)^2/n) \text{ —-> Root Mean Square Error}$$

where n is number of data points, x_i is the i-th value and y_i is the corresponding prediction.

7 Experimental findings

As mentioned, analysis has been performed on both Amazon and Facebook dataset with CNN-Bi-LSTM ensemble model and with Transformer model. We found that Transformer model is giving best result on both the datasets as compared to the CNN-Bi-LSTM ensemble model. The Time2Vector concept outperforms the existing model that we took. The performance metric as already described are:Root Mean Square Error(RMSE) Mean Absolute Error(MAE).

The following is the Test set RMSE and MAE scores calculated for the different models:

	Mean Absolute Error(MAE)				Root Mean Square Error(RMSE)			
	CNN	BiLSTM	CNN-BiLSTM	Transformer	CNN	BiLSTM	CNN-BiLSTM	Transformer
Facebook	289.22	257.05	242.65	59.74	291.27	259.75	243.82	71.95
Amazon	3077.63	2934.16	2608.06	847.99	3080.62	2936.44	2609.79	1022.55

Figure 7.1: Test set RMSE and MAE scores

8 Conclusion and Future plan of work

We have implemented ensemble model of CNN-BiLSTM and Transformer with Time2Vector embeddings and Transformer model has performed well as compared to standard models that we considered (CNN and BiLSTM) on Amazon and facebook dataset with test set RMSE score of 1022.55 and 71.95 and MAE score of 847.99 and 59.74 . There are still more work which can be done to improve the performance of Transformer model. We can optimise the parameters using standard AI algorithms like: Genetic or Hill climbing or can tune the hyper-parameters in model training. This modification would make the Transformer more robust. We can also normalize the prices so as to reduce the error score relatively.

9 References

1. Hiransha M , Gopalakrishnan E.A , Vijay Krishna Menon, Soman K.P, “NSE Stock Market Prediction Using Deep-Learning Models,”International Conference on Computational Intelligence and Data Science

2. Lei Ji, Yingchao Zou , Kaijian He, Bangzhu Zhu(2019),”Carbon futures price forecasting based with ARIMA-CNNLSTM model”,”7th International Conference on Information Technology and Quantitative Management”

3. Yang Li, Yi Pan, ”A novel ensemble deep learningmodel for stock prediction based on stock prices and news” International Journal of Data Science and Analytics 2021.

4. Sidra Mehtab, Jaydip Sen. ”A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing” https://arxiv.org/abs/1912.07700 (2019)

5. Jintao Liu, Xikai Liu, Hongfei Lin, Bo Xu1, Yuqi Ren, Yufeng Diao1, Liang Yang. ”Transformer-Based Capsule Network For Stock Movements Prediction” https://aclanthology.org/W19-5511 (2019)

6. Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota,mSanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, Marcus Brubaker, Borealis AI ”Time2Vec: Learning a Vector Representation of Time” https://arxiv.org/pdf/1907.05321 (2019)

7. Hands-On Stock Price Time Series Forecasting using Deep Convolutional Networks: https://www.analyticsvidhya.com/blog/2021/08/hands-on-stock-price-time-series-forecasting-using-deep-convolutional-networks/

8. Tutorial on LSTMs: A Computational Perspective: https://towardsdatascience.com/tutorial-on-lstm-a-computational-perspective-f3417442c2cd

9. Tutorial on Transformers: The Illustrated Transformer: https://jalammar.github.io/illustrated-transformer/.