THESIS

STOCK MARKET PREDICTION USING MACHINE LEARNING

Submitted by

Efe Köseoğlu

Department of Computer Engineering

Çukurova University

(koseogluefe37@gmail.com)

Advised by

Prof. Dr. Selma Ayşe Özel

Spring 2023

TABLE OF CONTENTS

[ABSTRACT 4](#_Toc138293691)

[1. Introduction 5](#_Toc138293692)

[2. Literature Review 6](#_Toc138293693)

[2.1 Technical Analysis 6](#_Toc138293694)

[2.2 Fundamental Analysis 7](#_Toc138293695)

[3. Machine Learning 8](#_Toc138293696)

[3.1 Supervised Learning 9](#_Toc138293697)

[3.1.1 Regression 9](#_Toc138293698)

[3.1.2 Classification 10](#_Toc138293699)

[3.2 Unsupervised Learning 10](#_Toc138293700)

[3.2.1 Clustering 10](#_Toc138293701)

[3.3 Semi Supervised Learning 11](#_Toc138293702)

[3.4 Neural Networks 11](#_Toc138293703)

[3.4.1 Neurons (Nodes) 12](#_Toc138293704)

[3.4.2 Activation Function 13](#_Toc138293705)

[3.4.3 Layers 14](#_Toc138293706)

[3.4.4 Gradient Descent and Backpropagation 15](#_Toc138293707)

[3.4.5 Overfitting and Underfitting 15](#_Toc138293708)

[3.5 Deep Neural Network 16](#_Toc138293709)

[3.6 Recurrent Neural Networks (RNN) 17](#_Toc138293710)

[3.7 Long Short-Term Memory Networks 18](#_Toc138293711)

[4. Methodology 20](#_Toc138293712)

[4.1 Test Environment and Preparing Data 20](#_Toc138293713)

[4.2 Long Short-Term Memory Network 22](#_Toc138293714)

[4.3 Accuracy Testings 22](#_Toc138293715)

[4.4 Predicting Next Day 24](#_Toc138293716)

[5. Results 24](#_Toc138293717)

[5.1 Different Optimizer Results 26](#_Toc138293718)

[5.2 Different Loss Function Results 31](#_Toc138293719)

[5.3 Conclusion 33](#_Toc138293720)

[References 34](#_Toc138293721)

LIST OF FIGURES

[Figure 3. 1: Feed Forward Neural Network 10](#_Toc138283444)

[Figure 3. 2: Neuron 11](#_Toc138283445)

[Figure 3. 3: Activation Functions 12](#_Toc138283446)

[Figure 3. 4: Recurrent Neural Network 16](#_Toc138283447)

[Figure 3.5: LSTM network architecture 17](#_Toc138283448)

Figure 5. 1: META Stock Price History 25

Figure 5. 2: Amazon Stock Price History 25

Figure 5. 3: META Share Price with Adam Optimizer 26

Figure 5. 4: META Prediction Value with Adam Optimizer 26

Figure 5. 5: Second Time Iteration META Prediction Value 27

Figure 5. 6: META Share Price with SGD Optimizer 28

Figure 5. 7: META Prediction Value with SGD Optimizer 28

Figure 5. 8: Amazon Share Price with Adam Optimizer 28

Figure 5. 9: Amazon Prediction Value with Adam Optimizer 29

Figure 5. 10: Amazon Share Price with SGD Optimizer 29

Figure 5. 11: Amazon Prediction Value with Adam Optimizer 30

Figure 5. 12: META Share Price with Adam Optimizer, MAE 31

Figure 5. 13: META Prediction Value with Adam Optimizer, MAE 32

Figure 5. 14: Amazon Prediction Value with Adam Optimizer, MAE 32

Figure 5. 15: Amazon Prediction Value with Adam Optimizer, MAE 32

ABSTRACT

In this thesis, stock market prediction based on historical data of companies has been done by one of the machine learning methods called LSTM. Why has LSTM been chosen for this stock market prediction ? This question is briefly explained with a summary of machine learning methods through LSTM. One of the most crucial parts of a machine learning algorithm is updating weights and biases between nodes in a neural network. This operation is done by some calculations on loss functions. Loss functions are basically used for estimating error between observed value and desired value in dataset. To make updates on weights and biases optimizers have been developed such as gradient descent, adam, adagrad. What is the most efficient optimizer and loss function to use while predicting market value ? This question has been investigated. Also how the experts in the economy field evaluated market values ? This question is briefly explained with Technical analysis and Fundamental Analysis.

1. Introduction

According to the investopedia [1] the first company who holds the privilege of being the first company to offer equity shares of its business to the public was The Dutch East India Co. They were successful in managing the world’s first initial public offerings. Since the existence of stocks humankind has seen this as an opportunity to invest and make profitable incomes. But like every topic which has involved money, it has to proceed with caution in order to make a move in that field.

Turing machines, first described by Alan Turing in Turing 1936–7, are simple abstract computational devices intended to help investigate the extent and limitations of what can be computed[2]. Potential of computability provided by the Turing machine has evolved into a new era of information technologies which we are experiencing today. Machine learning is one of the sub branches of Artificial Intelligence focused on training the computers with various data sets then trying to get most efficient solutions/results depending on outputs of designed architecture in the algorithm.

Stocks are very unstable due to so many environmental variables such as natural disasters, political situations of a specific country, trending technologies of the era, CEO or other management staff changes etc. affects the stock prices in the long term or short term. In this high-risk and high-return market, the stock market has always been closely watched by investors (Daubechies, I. 1992), and stock forecasting has always been an attraction for researchers' attention. Can stock prices be predicted and could be a beneficial investment ? The Efficient Market Hypothesis (EMH) defends that stocks always trade at their fair value on exchanges, making it impossible for investors to purchase undervalued stocks or sell stocks for inflated prices [3]. And besides this, EMH considers all information has been reflected into stock prices. If these conditions are not fulfilled then that market wouldn’t be an efficient market anymore. Therefore it should be impossible to outperform the overall market through expert supervision on stocks or market timing1.

1Market timing: Market timing is the act of moving investment money in or out of a financial market—or switching funds between asset classes (grouping of investments) based on predictive methods. If investors can predict when the market will go up and down, they can make trades to turn that market move into a profit[4].

Only way to make a profit in these circumstances investors should purchase riskier investments. Followers of EMH believe that stock prices can not be predicted since it follows Random Walk (Shonkwiler, 2013; Hull, 2009; Malkiel, 2003).Furthermore some of the researches has already declared this theory become a doctrine (Manahov and Hudson, 2014).

If EMH’s arguments are right about the stock market then there would be no reason to invest, so there wouldn’t be a stock market after all until today.One of the opposite arguments of EMH is Adaptive Market Hypothesis (AMH) has been developed (Andrew Lo, 2004). AMH combines EMH’s philosophy with behavioral finance which means psychological influences and biases affect the financial behaviors of investors and financial practitioners[5]. According to Lo, the theory’s founder, believes that people are mainly rational, but sometimes can quickly become irrational in response to heightened market volatility. Irrational behaviors like loss aversion, overconfidence, overreaction to some event happened on stock market, in company or in a country builds the human evolutionary model. And this model leads to competition between investors, adaptation to unexpected events, overall natural selection[6].

2. Literature Review

Besides hypotheses about the relationship between stock market and the investors, Technical and Fundamental analysis are the two main approaches used by Finance practitioners in forecasting stock prices and making trading decisions[7]. By using these two analyses or combined inputs from them, impact on machine learning-based stock price forecasting has been investigated [8]. In [9] Long- Short Term Memory network used for the potential movement of stocks based on collected news articles from the previous day. This output is put into another layer of neural network as an input with correlated stock values to stock prediction.

2.1 Technical Analysis

Technical analysis forecasting based on price movement of virtually any tradable instrument which is under the effects of supply and demand logic, including stocks, bonds, futures and currency pairs. This explanation gives us the definition of “security” in finance which is interchangeable individual goods or assets of the same type. In other words technical analysis is the study of supply and demand forces as reflected in the market price movements of a security[10]. Through the years, researchers have developed different patterns and signals to support technical analysis.There are some commonly used indicators among the technical analysts which are price trends, chart patterns, volume and momentum indicators.

There is three general assumptions for this disciple:

1. The market discounts everything: Technical analysis also assumes that everything from a company's current state to various stock factors (value, size, momentum, quality) has been reflected into the stock. This perspective is compatible with EMH.

2. Price moves in trends:Stock prices are more likely to continue a past trend rather than move randomly.

3. History tends to repeat itself: Like AMH suggested, market actions are affected by human psychology. Based upon this behavior chart patterns have been developed to analyze certain cases. Consider that if there is fear about some company going to change it’s CEO there is a chart pattern about it.We can use this pattern to predict stock movement. Since this patterns have worked well in the pas it is assumed that they will continue to work well in the future[11].

Technical analysis also has limitations. For example sometimes investors put pre-defined rules to prevent loss or lock their profit in that position. If large numbers of investors have done this, when the stock price activates some predefined rule, there will be a huge number of sell orders which will push the stock down.Then, other investors will see the price decrease and also sell their positions. These actions will be a trend for a short term but over the long run, these exceptions won’t decide the price.

2.2 Fundamental Analysis

Stock's current price often does not fully reflect the value of the company when compared to publicly available financial data[12]. Main idea of the fundamental analysis is to determine the expected value of a company’s stock. If investors take a deep searching in stocks based on, in order:

* Current state of the economy in that country
* Strength of the specific industry
* Financial performance of the company providing the stock

Considering all of these circumstances, they will create a momentum which can be beneficial or damaging to investors. This is named as “intrinsic” value of the stock.

In Table 1 the relationship between intrinsic value and current market price is given. Once the intrinsic value of the stock is calculated, trading decisions are made based upon Table 1.

|  |  |
| --- | --- |
| Trading Signal | Intrinsic value vs. Current Market Price |
| BUY | Intrinsic value > Current Market Price |
| SELL | Intrinsic value < Current Market Price |

Table 1: Trading Decision for Fundamental Analysis

There could be a different results from various analyses so the mean value of these calculations will be considered as the intrinsic value.Fundamental analysis assumes that in the long run, the stock market will reflect its results.The problem is no one would estimate how long the run really is.It could be days or years.

In summary, while fundamental analysis focuses on evaluating the intrinsic value of a stock based on a company's financials and industry factors, technical analysis concentrates on analyzing historical price and volume data to predict short-term price movements.

3. Machine Learning

The term of machine learning popularized with the program developed to achieve victory in computer checkers in 1959 .This game is relatively simple yet it can be deeper with every move made by players. Arthur Samuel has developed a specific algorithm to calculate the optimal move in a search tree. This was one of the pioneer algorithms in the AI field which is now called alpha-beta pruning. In one of his conferences Samuel suggested “How can computers learn to solve problems without being explicitly programmed?"[13].

Machine learning can be considered as an advanced tool that is applicable in various fields such as medicine by data clustering, computer vision, speech recognition such as the fairly popular app ChatGPT. One of the crucial steps in machine learning to collect large amount of data. Of course there could be redundant data in the data set so quality of data will affect the results. Data set will be separated into test and train groups, on train data, generated algorithms will be applied such as classification, clustering and regression. Results from training operation will be compared with test data then a relationship will be established between these two data groups. To make more precise accuracy, train and test data sets could be swapped.

Machine learning methods are examined in three different branches which are supervised learning, unsupervised learning and semi supervised learning.

3.1 Supervised Learning

In a training set of N example input-output pairs (x1,y1), (x2,y2),…(xn,yn) where each yj was generated by an unknown function y = f(x), generate a function h that approximates the true function f [14]. Here x and y can be any value, doesn’t have to be a number, the function h is a hypothesis. One of the distinct factors of supervised learning is using labeled data while classifying or predicting outcome values. Hypothesis function generated through the training process so we can use it to mapping when new variables are given to the data set. The training data provided to the machines work as the supervisor that leads the machines through the learning process. Also the algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized. For finances, input data is usually the stock values and the output is usually a value of a particular stock on designated time. Supervised learning derived from two types of problems which are regression and classification.

3.1.1 Regression

While modeling statistical data, regression analysis aims to establish a relationship between a dependent variable and one or more independent variable.It predicts continuous values such as temperature,age,salary, stock price etc. For example in [9] regression analysis used while predicting stock values, independent variables were market data, news articles, correlated stocks, and its stock data are used as input. Considering all of these variables while the prediction process makes it more accurate overall.

Regression analysis has various types of subdivisions.More common ones are linear regression, logistic regression, polynomial regression, and stepwise regression.Linear regression model assumes that relationships between dependent and independent variables are linear. Logistic regression used for binary dependent variables.

3.1.2 Classification

Classification on the other hand makes discrete values while predicting. One of the basic examples of classification methods is k-Nearest Neighbors. For example in a multidimensional space that each class has a label, in here k is defined by the user, when the new sample arrives to this space, based on number k, nearest neighbors will be detected then this new sample will be labeled based on the majority of neighbors. This method rely on the number k which is defined by experts in their fields.

Another classification method Support Vector Machines aims to find optimal hyperplanes to separate data points. Assume that there is a 3 dimensional space then hyperpşanes are 2 dimensional. Optimal hyperplane would have the largest separation between data points, in other words margin.SVMs also can efficiently perform a non-linear classification using what is called the kernel trick1.

3.2 Unsupervised Learning

Unsupervised learning takes unlabeled data as an input, aims to discover any hidden groups in data without any output information.Since the data is unlabeled this method is slower than supervised learning. Its ability to discover similarities and differences in data groups make it the ideal solution for exploratory data analysis, image recognition and customer profiling. Unsupervised learning models are explained in the three main tasks: Clustering, association, and dimensionality reduction.

3.2.1 Clustering

One of the common clustering methods is k means clustering.

1Kernel trick: Kernel methods represent the techniques that are used to deal with linearly inseparable data or non-linear data set. The idea is to create nonlinear combinations of the original features to project them onto a higher-dimensional space via a mapping function, where the data becomes linearly separable. This operation is often computationally cheaper than the explicit computation of the coordinates.[15]

* First pick a number k for cluster centers, number k usually preferred to taken from outside resource
* Define k random points as cluster centers.
* Assign every data point to its closest cluster center.
* Take the mean of every separate cluster then assign the nearest point as a cluster center. Repeat

Iteration number determined by cost function.Cost function should be the minimum value when all cluster centers are at the optimum position. For the disadvantages of this model, because it starts with random points, each result obtained is not always optimum. Number k depends on outsource. When clusters look like concave shapes, it is not successful. Model is sensitive to outliers.

3.3 Semi Supervised Learning

Also called weak supervision, combines a small amount of labeled data with a large amount of unlabeled data during training. In real world problems most of the data are unlabeled. If researchers want to get a labeled data set, this will cost highly. Main idea is to train an initial model with the relatively smaller labeled data and then iteratively apply it to the greater number of unlabeled data. The heuristic approach of self-training (also known as self-learning or self-labeling) is historically the oldest approach to semi-supervised learning. Self-training can take any supervised method for classification or regression and modify it to work in a semi supervised way, taking advantage of labeled and unlabeled data [16].

3.4 Neural Networks

Neural networks inspired from connection between biological neurons.These networks usually contain multiple layers such as input layer, one or more hidden layers (middle layers), output layer. Each neuron in the network receives input from the previous layer, applies a mathematical operation to the input, and produces an output signal that is transmitted to the next layer. The strength of connections between neurons, called weights, are learned from training data. During the training process, the network adjusts the weights based on the input data and the desired output.

Neuronal excitability is controlled by electric current that makes depolarizes or hyperpolarizes the excitable cell membrane[17]. In order to make changes on cell membrane electric current needs to reach some threshold. Like biological neurons,nodes in the neural network has a threshold value that determines either that node is going to activate and send data to the next layer or no activation going to happen.

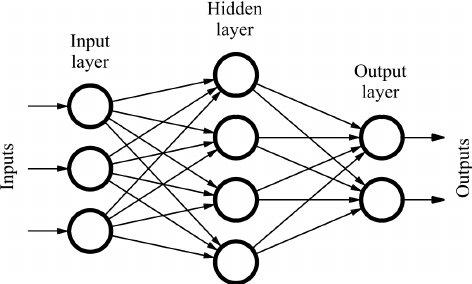


Figure 3. 1: Feed Forward Neural Network [18]

Figure 3.1 is an example of a feed-forward neural network. In this version of the network, information only moves in one direction from input through output. There are no cycles or loops in the network[19].Figure 3.1 is an example for a linear neural network, which contains only a single layer of output nodes.Each individual input node and its weights will be calculated. The error between these calculated outputs and final values will be optimized by adjusting the weights.

3.4.1 Neurons (Nodes)

Neurons or nodes are the building blocks of a neural network.For example hidden layer neurons will receive different input and weight calculations and generate new input for the next layer neuron.

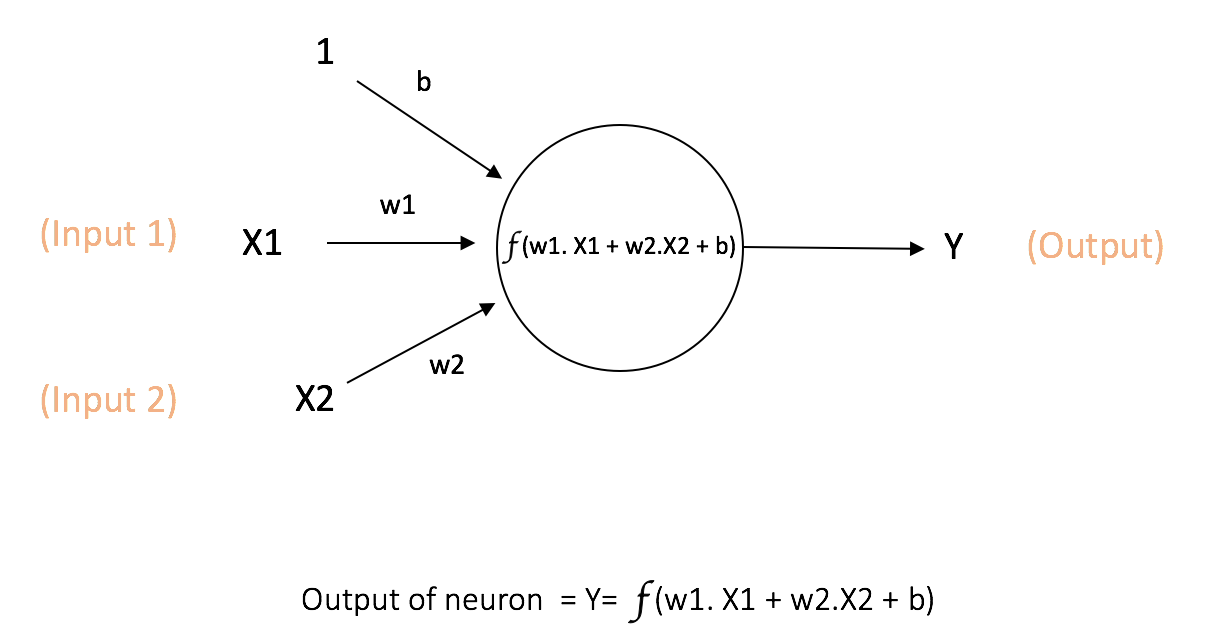


Figure 3. 2: Neuron[20]

In Figure 3.2, x1 and x2 are the inputs, w1 and w2 are the weights, b is the bias, y is the output and f is the activation function which determines the threshold. The first artificial neuron was the Threshold Logic Unit (TLU), or Linear Threshold Unit, [21] first proposed by Warren McCulloch and Walter Pitts in 1943.

3.4.2 Activation Function

To determine which neuron will be activated in the network, an activation function is required. By adding an activation function to each hidden layer neuron, the network will separate useful information from irrelevant information. Assume that there is a network without the activation functions.Then every neuron in the network will calculate linear transformation based on inputs, weights and biases. Hidden layers in the network will lose their meaning because regardless how many hidden layers we add, all layers will act in the same way because the combination of two linear functions is a linear function itself. This feature makes simplified networks but against complex tasks it will behave like a linear regression model.

There are different types of activation functions. More common ones are:

**Sigmoid**: It gives an output in the range between 0 and 1.

σ(x) = 1/(1 + e−x) (3.1)

**Tanh**: It gives an output in the range between -1 and 1.

tanh(x) = 2σ(2x) – 1 (3.2)

**ReLU**: In this type of function the activation is thresholded at zero.

f(x) = max(0, x) (3.3)

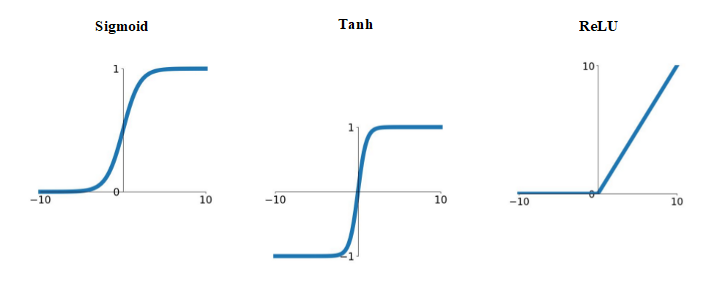


Figure 3. 3: Activation Functions

3.4.3 Layers

**Input Layer**: Initial values from the source will be taken in this layer. There won’t be any calculations.Information in this layer will be passed on to the hidden layer.

**Hidden Layer**: Information from input layer will be combined with weights and biases then this result will be used as an input for hidden layer.Activation functions also take part in this layer to decide which information will be passed on to the output layer.

**Output Layer**: Final layer of the network that delivers final result. This layer also uses activation functions however output layers usually use different activation functions than hidden layers due to the goal of the network.

3.4.4 Gradient Descent and Backpropagation

Assume that we have a linear regression model. To measure of how well a model is performing on a given task, Loss functions will be used. One of the examples for Loss functions is sum of the squared residuals. Residual is the difference between observed value and predicted value. By calculating squared residuals for all data points and adding them gives use sum of the squared residuals.We can plot different sum of the squared residual values by calculating via different intercept values also known as bias term of equation. So the question is what is the best value for intercept in order to determine minimal loss function value. A slow and painful method for finding the minimal sum of the squared residuals is to try a bunch more values for the intercept.Gradient Descent is more efficient in this situation.

Gradient Descent only does a few calculations far from the optimal solution and increases the number of calculations closer to the optimal value: in other words Gradient Descent identifies the optimal value by taking big steps when it is far away and smaller steps when it is close.

Backpropagation commonly used in neural networks for adjusting the weights of the network’s connections.Backpropagation moves in opposite direction from output to input. More specifically, backpropagation computes the gradient of a loss function with respect to the individual weights of the network, computing the gradient one layer at a time[22]. For each iteration, adjusting the network's weights in order to minimize the difference between the observed value and the predicted value. Let assume we have a linear regression model. Main idea for backpropagation is when a parameter is unknown, we use the chain rule to calculate the derivative of the sum of squared residuals with respect to the unknown parameter. Initialize the unknown parameter with a number for example 0, and use gradient descent to optimize the unknown parameter.

3.4.5 Overfitting and Underfitting

If a model trains for too long on the same data it starts to memorize noise or irrelevant information in the dataset instead of learning relationships in the data. The overfit model performs exceptionally well on the training data but performs poorly on new, unseen data.There is high probability that overfitting occurs when the model is too complex. One of the best methods to detect overfitting is testing the model on various data samples. High error value means overfitting. For example in k fold cross validation, the data set will divide into k equally sized subsets called folds. One subset will be chosen as a validation data , then training of the model carried out on k-1 subsets. After the observation and the scoring of this step, iterations repeat until every sample has been tested.

To avoid overfitting there are multiple ways. For example, early stopping monitors the model's performance on a validation set during training and stops the training process early when the performance doesn’t develop anymore. Another example is future selection or pruning.Logic of the pruning is selecting the most relevant features for your model can prevent it from fitting noise or irrelevant patterns in the data.

Underfitting is the opposite of overfitting and occurs when a machine learning model is unable to capture the pattern characteristics in model and complexities of the data during training. Because of the lack of training process model performs poorly not only on the training data but aslsı on unseen data too. Underfitting is detectable based on two factor, bias and variance. Bias refers to the assumptions made by the model which is actually error rate of the training data. Variance is difference between the error are of training data and testing data. If there is a high bias and low variance, this indicates that data is underfitting.

3.5 Deep Neural Network

The “deep” word refers to depth of layers in a neural network.Deep neural network consists of multiple layers between input layer and output layer. Increase in hidden layer numbers means that neural networks will be able to learn more complex data set patterns. Most neural networks are feed forward like in Figure 3.1 however deep neural networks use backpropagation to adjust weights and biases of its nodes.Simple neural network is more dependent on human intervention. Human experts determine the hierarchy of features to understand the differences between data inputs [23].However deep neural networks can automatically determine labels of the data set.

Deep neural networks can be examined on different branches specified on various fields.For example convolutional neural networks (CNNs) are based on convolution operations, CNNs used in image and video recognition, image classification,natural language processing. Another example is recurrent neural networks (RNNs) which have connections that allow information to flow in cycles used in handwriting recognition, speech recognition. Since this thesis uses one of the recurrent neural network branches, this part will explain in detail.

3.6 Recurrent Neural Networks (RNN)

Recurrent neural networks(RNNs) designed to process sequential data which are nodes in the dataset are dependent on the other nodes, or time series data. In traditional feedforward neural networks like in Figure 3.1 input processing is independent from each other, RNNs hidden layer allows them to gather information from previous inputs and use it to affect the processing of future inputs in the sequence. One of the distinguishing characteristic of RNNs is their ability to handle sequential data of changeable lengths.This makes them well suited for tasks such as natural language processing, speech recognition, machine translation.Another unique characteristic of RNNs is unlike feedforward neural networks which have different weights and biases related with nodes, RNNs share the same weight parameter across each layer of network[24].Yet still weights update has done by gradient descent and backpropagation.

RNNs use a slightly different backpropagation method called backpropagation through time (BPTT) algorithm to calculate gradients. The main difference is BPTT sums errors at each time step however feedforward networks do not need to sum errors.

In Figure 3.4 rolled and an unrolled (or unfolded) structure of RNN has been showed. Unfolded and folded representations are equivalent of each other. Instead of remembering which value is in the loop and which value is in the input, we can unroll the feedback loop by making a copy of the neural network for each input value. The terms in Figure 3.4 are:

xt: The input at time step t

ht:hidden state at time step t

yt: output at time step t

V: communication between steps

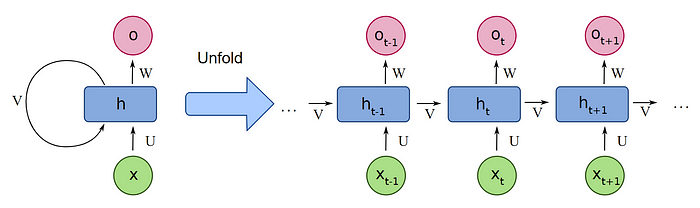


Figure 3. 4: Recurrent Neural Network [25]

RNNs may have encountered major problems while backpropagating. Problem is that the more we unroll a RNN, it will become harder to train.Each time we unroll the dataset this action depends on weights and biases. In order to find the parameter values that give us the lowest values for the Loss Function, we usually want to take relatively small steps. However, when the gradient contains a huge number, then we’ll end up taking relatively large steps. And instead of finding the optimal parameter, we’ll just bounce around the dataset till weights get too large and become NaN values. This is called exploding gradients. On the contrary if the gradient contains a too small number, it continues to become smaller which means an update on the weights will be insignificant. Learning is not possible in these conditions.This is called vanishing gradients.

3.7 Long Short-Term Memory Networks

If we want to use a neural network to predict stock prices then we need a neural network that works with different amounts of sequential data so we need the neural network to be flexible in terms of how much sequential data we use to make a prediction. Since RNNs use unfolded memory which is obtained by unfolding the network based on weights, RNNs are not suited for long term operations. Long Short-Term Memory (LSTM) which is a type of RNN that is designed to avoid the exploding/vanishing gradient problem.The main idea behind the how LSTM works is that instead using the same feedback loop connection for events that happened long ago and events that just happened yesterday to make a prediction about tomorrow, LSTM uses two separate paths to make predictions about tomorrow. One path is for Long Term Memories and one is for Short Term Memories. Lack of Weights allows the Long-Term Memories to flow through a series of unrolled units without causing the gradient to explode or vanish. Short Term Memories are directly connected to weights that can modify them. Figure 3.5 LSTM network architecture shown.

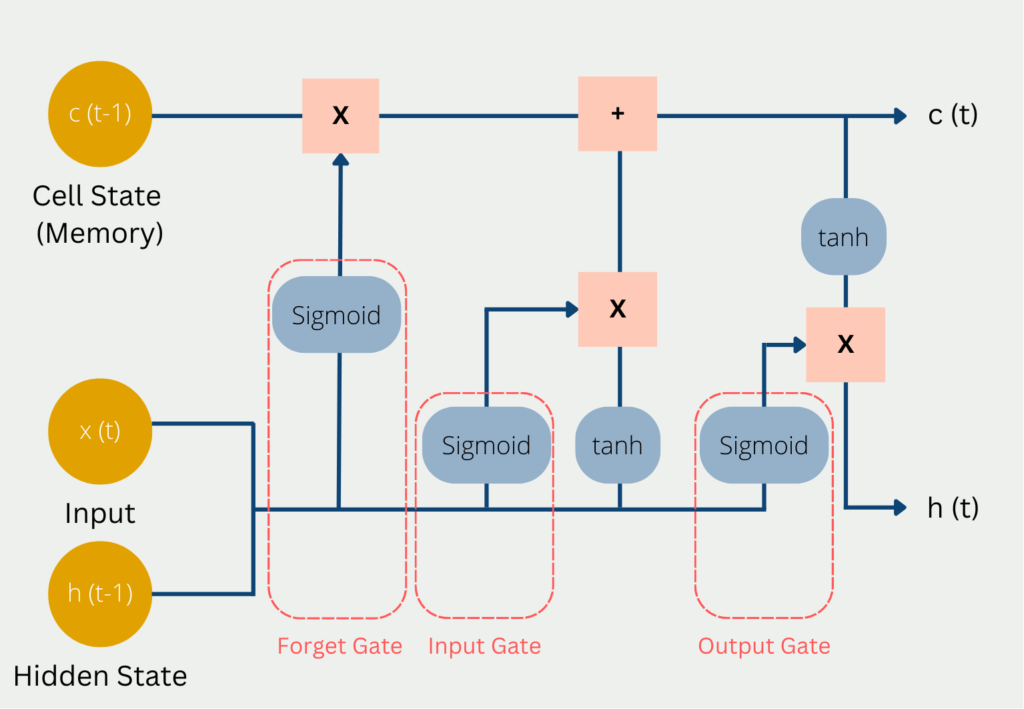


Figure 3.5: LSTM network architecture [26]

LSTM structure can be divided into 3 main sections:

1. Forget gate: First stage in a LSTM unit determines what percentage of the Long-Term Memory is remembered.

2. Input Gate: In the Middle stage the block which is using tanh activation function combines the short term memory with the input, to create a potential long term memory. Block which is using sigmoid activation function determines what percentage of that potential memory to add to the long term memory. Multiply short term memory and input by their respective weights then add a bias term to this result from this calculation we get input value for tanh function. Get y value via the input according to activation function.Now LSTM has to decide how much of this potential memory to save. Multiply short term memory and input by their respective weights then add values together and add a bias term to it then we get input value for sigmoid function. Get y value via the input according to activation function.Finally multiply these two results and add to previous long term memory.

3. Output Gate: Last Stage updates the short term memory.We start with the New Long-Term memory and use it as input to the tanh activation function and after using the updated value of long term memory into the tanh function we get potential short term memory. Now LSTM has to decide how much of this potential short term memory to pass on. This part will use sigmoid function again. And multiply result of the both functions to get new short term memory. This result will be the output from this entire LSTM unit.

There are several advantages using LSTM rather than RNNs:

* LSTM reuses the exact same weights and biases so it can handle data sequences of different lengths.
* LSTM has to remember what happened on day 1 in order to correctly predict the different output values on the day it is going to predict.
* Using separate paths for long term memories and short term memories LSTM networks avoid the exploding vanishing gradient problem and that means we can unroll them more times to accommodate longer sequences of input data than a vanilla RNN.

4. Methodology

This section is about explaining the code and how it actually implemented on the training data.Test environment and datasets briefly explained. Detailed process of LSTM network has been evaluated.

4.1 Test Environment and Preparing Data

The main purpose of this thesis is changing the optimizer of the LSTM model and changing the loss function used in the prediction process, how these two factors affect the result of the LSTM model and which optimizer and loss function would give the better result. There are different types of optimization algorithms such as Gradient Descent, Stochastic Gradient Descent which is a variant of gradient descent with additional properties, Adam(Adaptive Moment Estimation) which uses a combination of gradient descent with momentum [27]. To make a proper comparison, preparing a dataset is the crucial part of the machine learning process .In order to do this Python language has been selected for this thesis. Python has efficient libraries for data processing like Numpy and Pandas which are making easier manipulation on datasets with less complicated syntax.

First dataset obtained from yfinance API. Since yfinance has already integrated into the python environment there was no additional process other than adding it as a library. From this point we have obtained our data via API so next step is preparing data.

scaler = MinMaxScaler(feature\_range=(0,1))  
scaled\_data = scaler.fit\_transform(data['Close'].values.reshape(-1,1))

prediction\_days = 60

Scale down all the values that we have so that they fit in between 0 and 1. Let's say if we have the lowest price of 10 dollars and the highest price of 500 dollars press, all those values so that they fit in between 0 and 1. This is from the sklearn preprocessing module. Now we’re not going to transform the whole data frame, we’re only going to be interested in the closing price because we’re not going to predict the opening price. We’re going to predict the price after the markets have closed. How many days do I want to base my prediction on, which means how many days do I want to look back to decide what the price is going to be for the next day. .

for x in range (prediction\_days, len(scaled\_data)):  
  
 x\_train.append(scaled\_data[x-prediction\_days:x,0])  
 y\_train.append(scaled\_data[x,0])  
  
x\_train,y\_train= np.array(x\_train),np.array(y\_train)  
x\_train =np.reshape(x\_train, (x\_train.shape[0],x\_train.shape[1],1))

For the range of the loop, start counting the 60th index to the last index. We’re going to append 60 values and then the next value as a training example because this is labeled data. We know the first 60 values and we also know the there is goint to be 61th value but we need to prepare the 61th value here in a data set, we need to prepare the whole training data in a way that we have 60 values and then the model can predict what the next value is going to be .We don’t have any negative values up until next. y\_train is going to be the 61st value. Convert them into numpy arrays. Reshape the x\_train in order to train with a neural network model.

4.2 Long Short-Term Memory Network

In this section building the LSTM model has been explained.

model=Sequential()  
  
model.add(LSTM(units=50,return\_sequences=True,input\_shape=(x\_train.shape[1],1)))  
model.add(Dropout(0.2))

We will be adding layers from now on.The structure of the models will be implemented like one layer of LSTM then one layer of dropout and at the end of this process there will be a dense layer that is going to be one unit and this one unit is going to be the stock price prediction. The Dropout layer randomly sets input units to 0 with a frequency of rate 0.2, at each step during training time, which helps prevent overfitting. There is possibility that we can achieve our goals in less layers or more layers or more units, by experimenting ideal number could be achieved. More units or layers we add, longer we are going to do the training. And again there is a possibility of our result will overfit if we use too many layers of sophistication.

Return\_sequences is true because LSTM is a recurrent cell so it’s going to feed back the information. It’s not just going to feed forward the information like an ordinary dense layer. We’re going to feed it into 25 epochs which means the model is going to see the same data 24 times. 32 batch size means that the model is going to see 32 units at once all the time.

4.3 Accuracy Testings

In this section we are going to figure out how well the model would perform on the past data. So we’re not going to directly predict the next future data we don’t know yet, we’re going to see how well this model performs based on the data that we already have. This data has to be data that the model has not seen before.

After getting the data:

* Get the prices.
* Scale the prices.
* Concatenate a full data set of the data we want to predict on.

total\_dataset=pd.concat((data['Close'],test\_data['Close']))

Then create a total\_dataset to combine the training data and test data. For this we’re going to concatenate methods in the pandas library. Concatenate the close values of data with close values of test\_data.

model\_inputs = total\_dataset[len(total\_dataset)-len(test\_data)-prediction\_days:].values  
model\_inputs = model\_inputs.reshape(-1,1)  
model\_inputs = scaler.transform(model\_inputs)

Our model is going to see model\_input as an input value while predicting the price. Subtracting the “prediction days'' means we want to start calculations as soon as possible and semicolon means taking all values up until the end.

x\_test= []  
  
for x in range(prediction\_days,len(model\_inputs)):  
 x\_test.append(model\_inputs[x-prediction\_days:x,0])

In this part we’re going to make predictions. With subtracting predictions\_days:x we guarantee that the value won’t be a negative number.

predicted\_prices= model.predict(x\_test)  
predicted\_prices= scaler.inverse\_transform(predicted\_prices)

Predicted prices are going to be scaled so we need to reverse scale them, inverse transform them. Using the same scaler with inverse transform method to get actual predicted prices.After these processes, matplotlib.pyplot library has been used to make visualizations.

4.4 Predicting Next Day

If accuracy testing has been completed without any unexpected errors then the next step is predicting actual prices. Since we’ve already prepared the desired day’s data and builded LSTM model, we can implement the same method in accuracy testing on the desired day.

real\_data = [model\_inputs[len(model\_inputs) +1 - prediction\_days:len(model\_inputs+1),0]]  
real\_data = np.array(real\_data)  
real\_data = np.reshape(real\_data,(real\_data.shape[0],real\_data.shape[1],1))

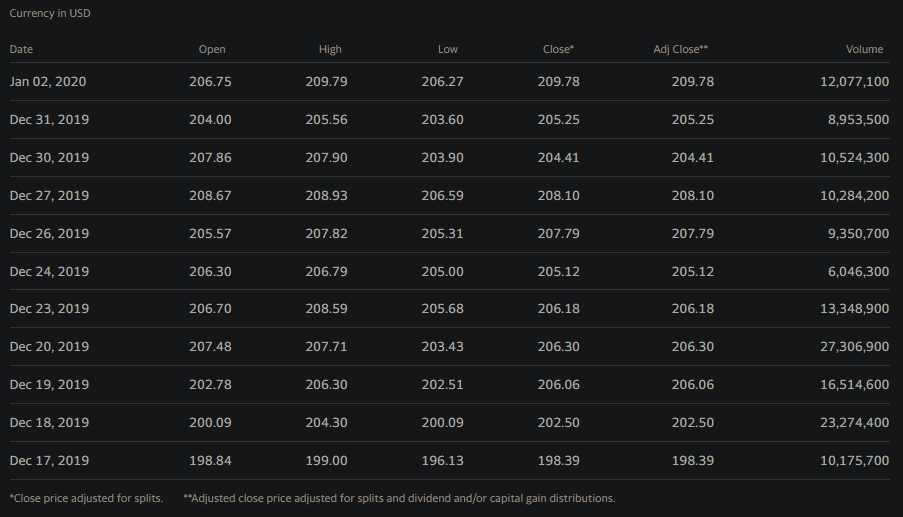
For the prediction of the next day first we create a list of names as real\_data. prediction\_days: len(model\_inputs+1) in here +1 indicates that the next day is going to be predicted.

prediction= model.predict(real\_data)

So we are going to use the real data as the input and then we’re going to predict an unknown day.

5. Results

In this chapter, LSTM results have been discussed based on different parameters such as optimizer and loss function changes. Before jumping into the conclusion on results, in figure 5.1 META (Facebook) stock price history and in Figure 5.2 Amazon stock price history has been shown. Prediction date (02.01.2020) and last 10 day’s data before prediction day included only but actual prediction based on last 60 days.

Figure 5. 1 META Stock Price History

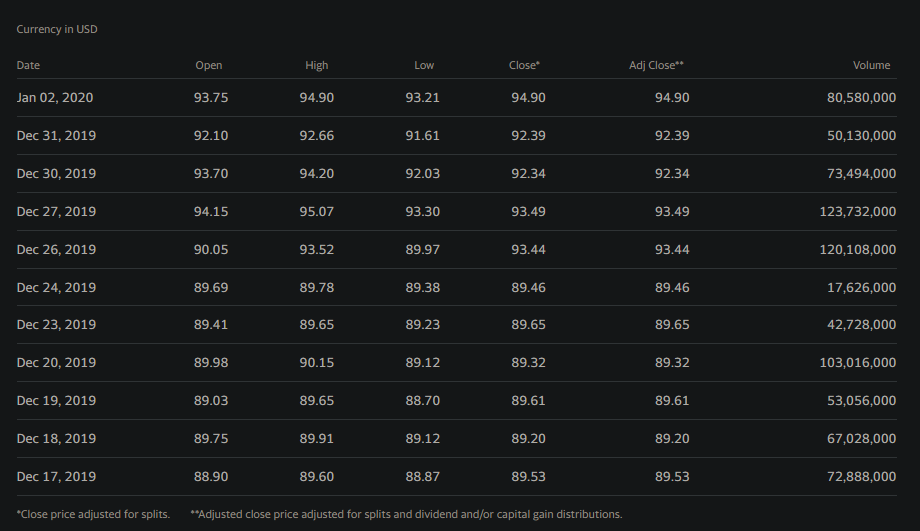


Figure 5. 2: Amazon Stock Price History

5.1 Different Optimizer Results

Optimizer: Adam

Loss Function: Mean Squared Error



Figure 5. 3: META Share Price with Adam Optimizer

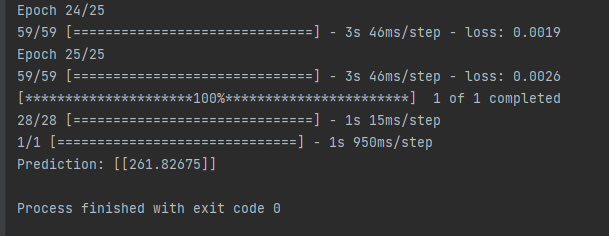


Figure 5. 4: META Prediction Value with Adam Optimizer

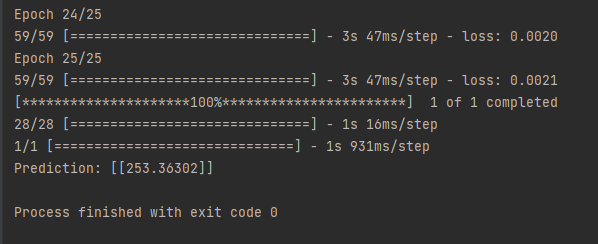


Figure 5. 5: Second Time Iteration META Prediction Value

For every iteration of the LSTM network could end up with different results. This is because LSTM is a non-deterministic algorithm. This means that every iteration result could be different due to random initialization of the weights.

Optimizer: Stochastic Gradient Descent (SGD)

Loss Function: Mean Squared Error



Figure 5. 6: META Share Price with SGD Optimizer

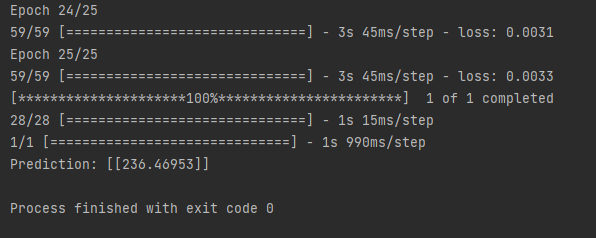


Figure 5. 7: META Prediction Value with SGD Optimizer

Optimizer: Adam

Loss Function: Mean Squared Error

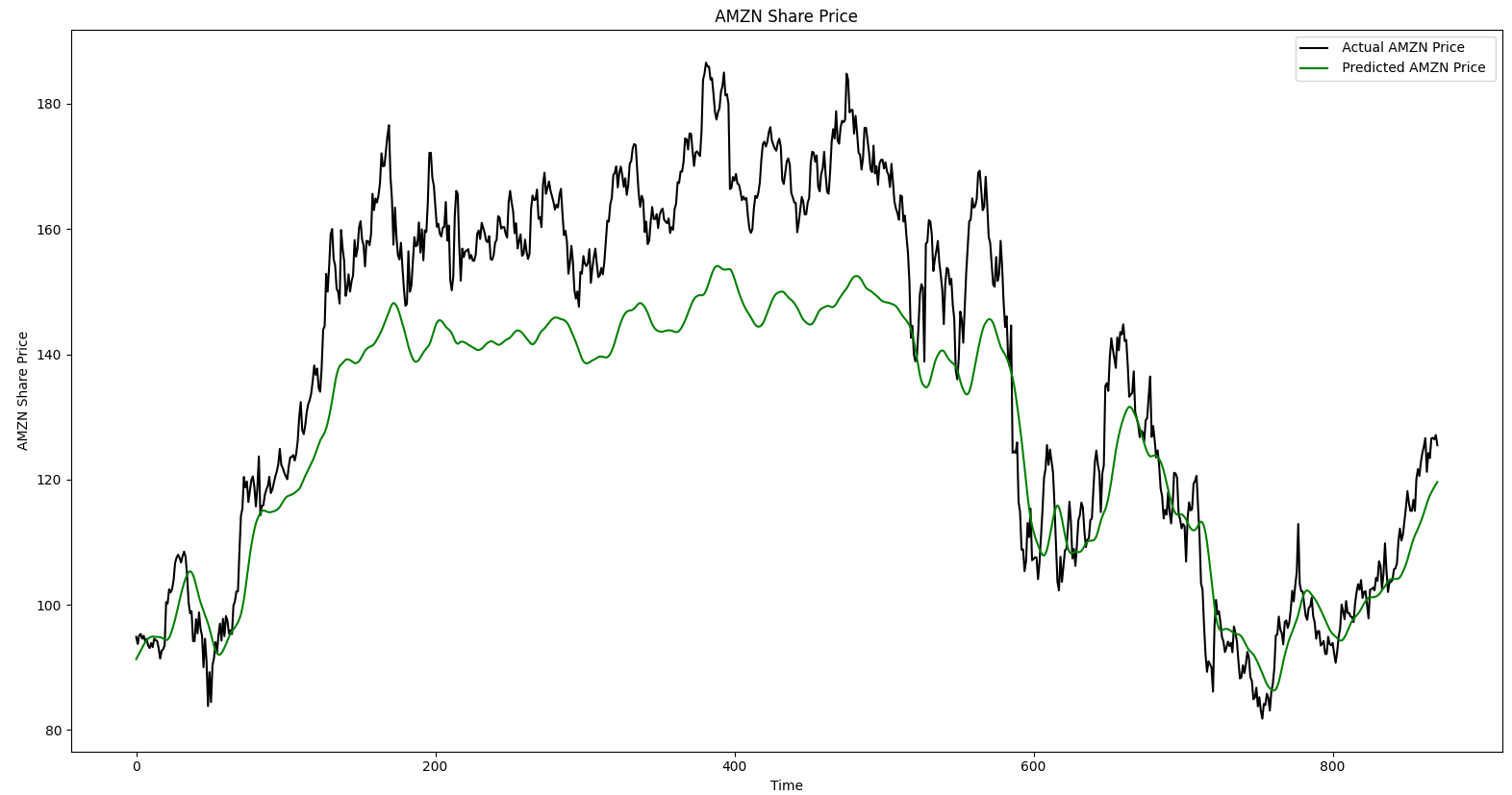


Figure 5. 8: Amazon Share Price with Adam Optimizer

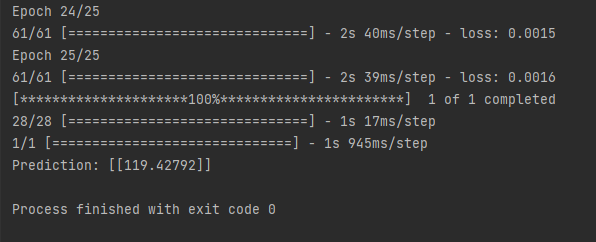


Figure 5. 9: Amazon Prediction Value with Adam Optimizer

Optimizer: SGD

Loss Function: Mean Squared Error

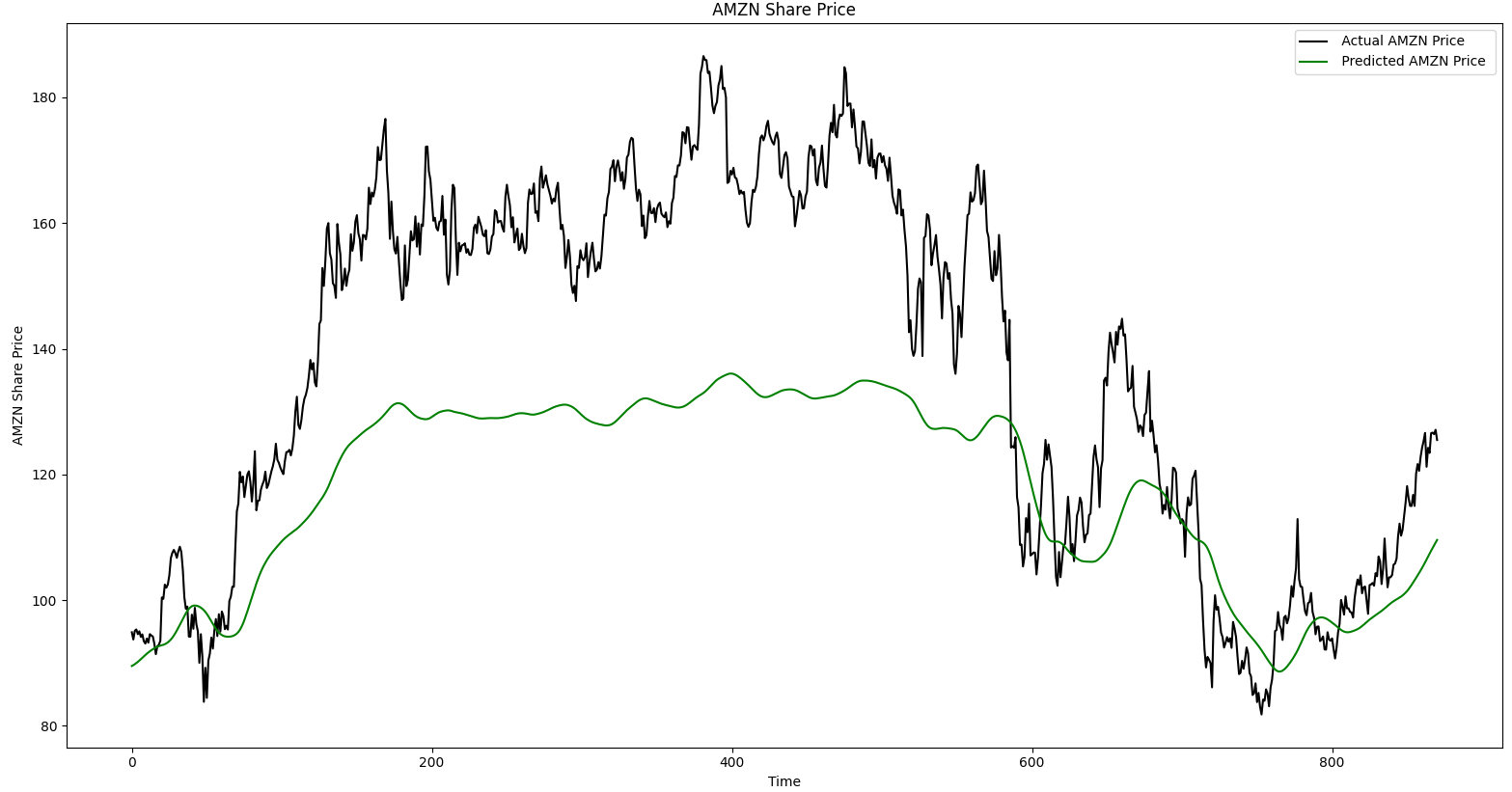


Figure 5. 10: Amazon Share Price with SGD Optimizer

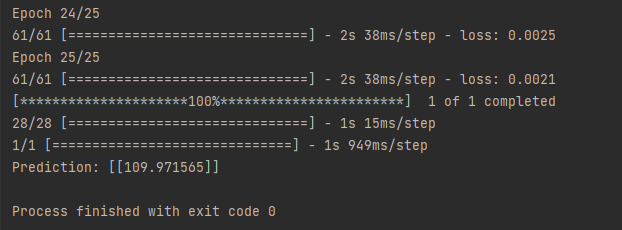


Figure 5. 11: Amazon Prediction Value with Adam Optimizer

Unlike gradient descent, stochastic gradient descent takes momentum as a parameter during the prediction process. Comparing Figures 5.3 with Figure 5.6, Adam optimizer performed more efficiently than SGD optimizer. This is because adam optimizer is a variation of SGD based on estimation of first and second moments of the gradient to adapt the learning rate for each weight of the neural network [28].Additionally comparing Figure 5.8 and Figure 5.10 reinforce this argument. But when the data set changed even Adam optimizer’s performance was not reliable as seen in Figure 5.8 yet still much closer to the actual results than SGD.

5.2 Different Loss Function Results

Optimizer: Adam

Loss Function: Mean Absolute Error



Figure 5. 12: META Share Price with Adam Optimizer, MAE

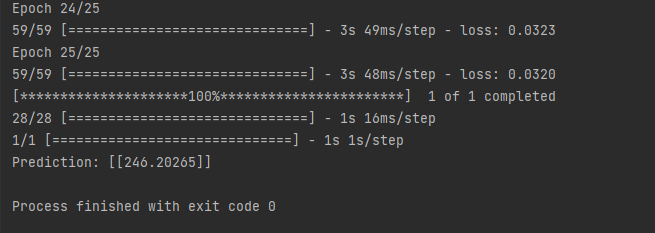


Figure 5. 13: META Prediction Value with Adam Optimizer, MAE

Optimizer: Adam

Loss Function: Mean Absolute Error

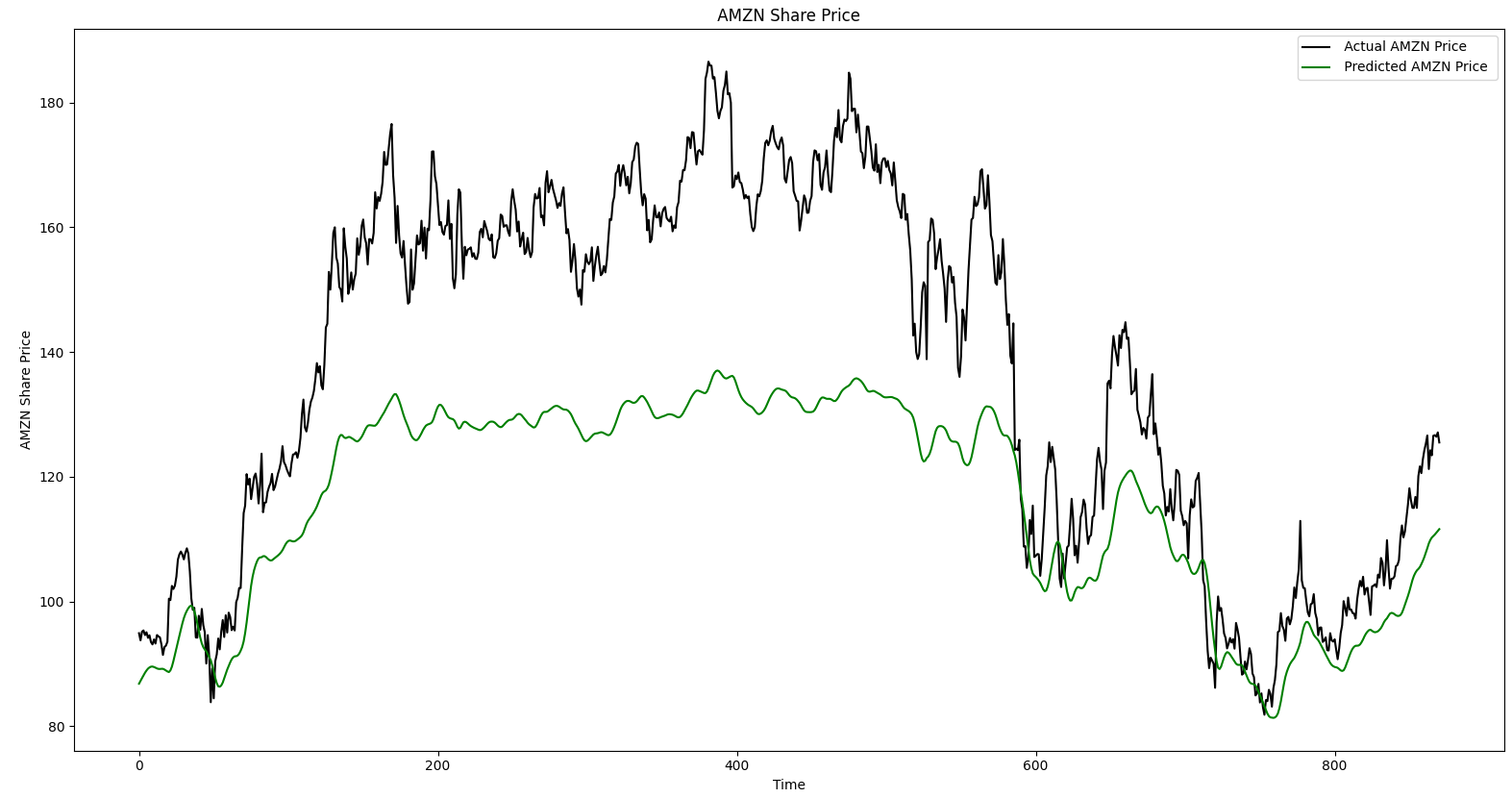


Figure 5. 14: Amazon Prediction Value with Adam Optimizer, MAE

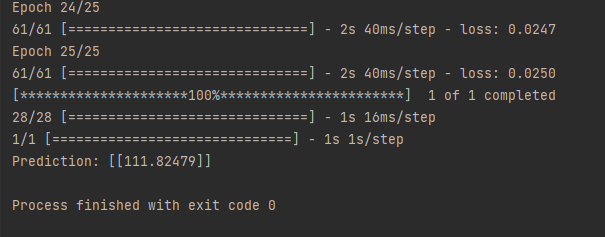


Figure 5. 15: Amazon Prediction Value with Adam Optimizer, MAE

As the name suggests, Mean Squared Error and Mean Absolute Error calculations explain itselves. Changing loss functions while predicting stock market value does not affect the result drastically as optimizer’s did. Yet still comparing Figure 5.3 with 5.12, there is an observable difference in final graphs.

5.3 Conclusion

Stock market prediction is a complex topic since various factors could change the final result. As technical analysis suggests, this thesis prepared LSTM networks based on historical data of companies. Some of the researchers take other parameters into account while predicting market value such as news articles but this kind of training process will be longer. The main idea of ​​this thesis was to determine which optimizer and loss function has a more impactful effect than the others while estimating the market value of a particular company. Adam optimizer introduced to the machine learning community is relatively new when compared to other optimizers. In the end Adam optimizer is more efficient than others since it takes innovative calculations into account. This indicates that when researchers try new solutions on stock market predicting, final results will be closer to the true values of tomorrow’s result. Changing loss functions also has an impact on prediction. For each individual situation, different loss functions can have better results than another one. First analyzing these situations then applying appropriate solutions would be wiser but this analysis process needs more data to classify each situation.

References

[1] <https://www.investopedia.com/ask/answers/08/first-company-issue-stock-dutch-east-india.asp>

[2] <https://plato.stanford.edu/entries/turing-machine/>

[3] The Library of Economics and Liberty. “Efficient Capital Markets”. by Steven L. Jones and Jeffry M. Netter. https://www.econlib.org/library/Enc/EfficientCapitalMarkets.html

[4] <https://www.investopedia.com/terms/m/markettiming.asp>

[5] <https://www.investopedia.com/terms/b/behavioralfinance.asp>

[6] Lo, A. "The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective, <https://www.researchgate.net/publication/228183756_The_Adaptive_Markets_Hypothesis_Market_Efficiency_from_an_Evolutionary_Perspective>

[7] M. C. Thomsett, Getting Started Getting Started in Stock Analysis, Illustrated Edition. Hoboken, Singapore: John Wiley Publishing, 2015.

[8] Erhan Beyaz, Effective Stock Prıce Forecasting Using Machine Learning Techniques Whilst Accounting For The State Of The Market.

[9] Hari Kiran Sai Surayagari, Stock Market Predictions Using Machine Learning

[10] Kirkpatrick, Charles D.; Dahlquist, Julie R. (2006). Technical Analysis: The Complete Resource for Financial Market Technicians. Financial Times Press. ISBN 978-0-13-153113-0.

[11] <https://cmtassociation.org/kb/technical-analysis-three-premises/>

[12] <https://www.investopedia.com/terms/i/intrinsicvalue.asp>

[13] Koza, John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A. (1996). "Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming". Artificial Intelligence in Design '96. Artificial Intelligence in Design '96. Springer, Dordrecht. pp. 151–170. doi:10.1007/978-94-009-0279-4\_9. ISBN 978-94-010-6610-5

[14] ] Stuart J. Russell and Peter Norvig. Artificial Intelligence: A Modern Approach.(3rd ed.) Prentice Hall, 2010. p. 695.

[15] Theodoridis, Sergios (2008). Pattern Recognition. Elsevier B.V. p. 203. ISBN 9780080949123.

[16] Chapelle, Olivier; Schölkopf, Bernhard; Zien, Alexander (2006). Semi-supervised learning. Cambridge, Mass.: MIT Press. ISBN 978-0-262-03358-9.

[17] Ye, H., Steiger, A. Neuron matters: electric activation of neuronal tissue is dependent on the interaction between the neuron and the electric field. J NeuroEngineering Rehabil 12, 65 (2015). <https://doi.org/10.1186/s12984-015-0061-1>

[18]<https://www.researchgate.net/publication/234055177_Computational_Methods_and_Optimization/figures?lo=1>

[19] Zell, Andreas (1994). Simulation Neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.). Addison-Wesley. p. 73. ISBN 3-89319-554-8.

[20] https://www.kdnuggets.com/2016/11/quick-introduction-neural-networks.html

[21] Martin Anthony (January 2001). Discrete Mathematics of Neural Networks: Selected Topics. SIAM. pp. 3–. ISBN 978-0-89871-480-7.

[22] Kelley, Henry J. (1960). "Gradient theory of optimal flight paths". ARS Journal. 30 (10): 947–954.

[23] <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>

[24] <https://www.ibm.com/topics/recurrent-neural-networks>

[25] https://medium.com/deeplearningbrasilia/deep-learning-recurrent-neural-networks-f9482a24d010

[26] <https://databasecamp.de/en/ml/lstms>

[27] <https://medium.com/analytics-vidhya/this-blog-post-aims-at-explaining-the-behavior-of-different-algorithms-for-optimizing-gradient-46159a97a8c1>

[28]https://optimization.cbe.cornell.edu/index.php?title=Adam#:~:text=Adam%20optimizer%20is%20the%20extended,was%20first%20introduced%20in%202014.