PREDICTING VIX FUTURES

Major Qualifying Project

Advisors:

RANDY PAFFENROTH
STEPHAN STURM

External Advisor:

JÖRG OSTERRIEDER, ZHAW

Written By:

SCOTT GUBRUD

A Major Qualifying Project
WORCESTER POLYTECHNIC INSTITUTE

Submitted to the Faculty of the Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science.

AUGUST 31, 2020 - NOVEMBER 3, 2020

Abstract

This project used machine learning techniques to try and approximate the values of the VIX and VIX futures based on S&P 500 options. A number of feedforward neural networks were trained using various network architectures and feature representations. LASSO regression was used to select a subset of the available features that appear to be more important for predictions. This subset was then used as a feature set for several neural networks. All neural networks were then compared on basis of accuracy to set what effects changes in the number of features had on the accuracy of the resulting model.

Table of Contents

List of Tables

Table **Page**

LIST OF FIGURES

FIGURE **Page**

Chapter 1

Introduction

Financial markets are important for many reasons including appropriate allocations of capital, individual investments, and for society as a whole. One of the important aspects of financial markets are their *volatility* or how much they vary over time around some mean trend. Interestingly, tracking the volatility of financial markets is so important that there are specific financial tools that focus on estimating the real-time volatility of financial markets and the most important one of these tools is called the VIX. The VIX is actually calculated using financial instruments called *options*. An option is a contract which allows a particular stock (or other financial instrument) to be purchased at a particular price at some future time. Options and volatility are actually closely connected. In particular, if a market is changing dramatically over time (i.e., has high volatility) then the probability that some stock will either have a large, or small price, in the future is increased. Accordingly, as options are predictions of future prices, they can be used to estimate how volatile the purchasers of the options think the market will be in the future. The S&P 500 is an index that essentially tries to track the overall trend of the market, see Section 2.2.6 for further information. The precise calculation of the VIX, which will be discussed in detail in Section 2.2.7, involves taking a weighted sum of the prices of a large number of options on the S&P 500. The exact number of options used in its calculation depends on which options are being traded at that time. The VIX itself is not an asset that is traded on the market. However, enough individuals wanted to be able to trade in the VIX that VIX futures were created. A future is essentially just a contract that states that the holder will purchase an asset at a set price on a set date. In the case of the VIX, which cannot be traded, a futures contract simply states that the holder will be paid the value of the VIX on a set date. One of the major uses of VIX futures is as a hedging tool. Hedging is when an investor purchases some asset that would be expected to have its value increase in cases when other assets owned by that investor would have their value decrease in order to try and minimize potential losses. VIX futures work well for this purpose as the volatility of the market tends to increase during periods of significant market downturn. Machine learning techniques are ways of having a computer learn some relationship between some set of variables. Machine learning techniques tend to be very good at making predictions when they are supplied with a large number of previous examples of predictions. This can make them very useful in making predictions in financial markets where large amounts of historical data tend to be available. In this work I will attempt to use machine learning techniques to approximate both the value of the VIX and the price of VIX futures from the prices of options on the S&P500 and common financial variables derived from those prices. In Chapter 2 I will discuss some previous work on similar topics and some of the technical background from both machine learning and finance necessary to understand my work. In Chapter 3 I will discuss how I obtained and preprocessed my data set and I will discuss the methods used to run my tests. In Chapter 4 I will discuss the results of my tests. In Chapter 5 I will discuss the implications and future areas of research for my work.

Chapter 2

Background

In this Chapter we will introduce some of the necessary background for understanding our work. Section 2.1 discusses the statistical and machine learning behind our work. Section 2.2 discusses the basis for the financial instruments that our being studied. Section 2.3 gives an overview of existing literature for similar and related topics.

2.1 Machine Learning

For this project, we have two main goals. The first is trying to predict the value of the VIX and the price of VIX futures using the prices of S&P 500 options. The second is to discover which values used in that prediction have the most significant impact and try and to predict the same values using a subset of the original inputs. Machine learning has been shown to be highly useful in understanding financial markets[1][2] and provides highly useful tools for meeting both of these goals.

2.1.1 Notation

Here we will introduce common notation that will be used throughout the rest of the report. As will be discussed shortly all regression and machine learning will be concerned with a set of observations consisting of a vector of input variables and a single corresponding output variable. We will use $X \in \mathbb{R}^{N \times M}$ to represent the entire set of input vectors and $y \in \mathbb{R}^{N \times 1}$ to represent the entire set of output variables, where N is the number of observations and M is the number of features. $X^i \in \mathbb{R}^{1 \times M}$ and $y^i \in \mathbb{R}$ will be used to represent the input vector and the output variable of the ith observation. $X^i_j \in \mathbb{R}$ will represent the jth variable from the vector X^i. $\hat{y} \in \mathbb{R}^{N \times 1}$ will be used to represent the set of predicted values of y and $\hat{y}^i \in \mathbb{R}$ will be used to represent the predicted y for the ith observation.

2.1.2 Regression

In general regression consists of attempting to find a relationship between a set of input variables, known as features, and an output variable. One begins with a set of observations, $[(X^0, y^0)$, ..., $(X^N, y^N)]$. There exists some function that maps from X to y, $f_{true}(X^i) = y^i$. The goal of regression is generally to find some function, $f_{approximate}(X^i)$, that is as close as possible to $f_{true}(X^i)$. In order to measure how close $f_{approximate}(X^i)$ is to $f_{true}(X^i)$, one selects an error function that compares the predictions made by $f_{approximate}(X^i)$ to the actual value and returns a value quantifying the error. Regression techniques work to find $f_{approximate}(X^i)$ in order to minimize the error function over the available data points. A simplified description of the goal of regression is to find a function that captures as much of the relationship as possible between a set of input and output variables.

2.1.3 Error

When doing any sort of predicting, one important question to be able to ask is how good are your predictions. This usually takes the form of an error function that takes in a predicted output variable and the actual output variable and returns some measure of how different those two are. One commonly used function is squared error which is simply the square of the difference between the two, i.e. $(y^i - \hat{y}^i)^2$[3]. The difference is squared to make all errors positive and more heavily penalize larger errors. When looking for the error over a set of predictions, squared error becomes mean squared error(MSE) which is the average of the square of the differences between the predicted and the actual data, i.e. $MSE(y, \hat{y}) = \frac{\sum(y^i - \hat{y}^i)^2}{N}$[3]. Another method for measuring error is percent error. Percent error is the difference between the prediction and the actual value expressed as a percentage of the actual value, i.e. $percent\_error(y^i, \hat{y}^i) = 100 * \frac{(y^i - \hat{y}^i)}{y^i}$. This can be extended over a set of predictions into mean absolute percent error by taking the average of the absolute values of all the errors. In the context of machine learning, the particular error function that a model is working to minimize is called a loss function and the value of that loss function for some particular set of predictions and actual outputs is called the loss of that set.

2.1.4 Training

In machine learning, training is the stage when your model actually does its learning. A machine learning model will have the ability to learn any function within a set of functions[3]. Calling this whole set of functions $f^{\theta}_{approximate}(X^i)$ where each value for θ corresponds to a different function within the set, we can view training as the process by which the model uses the data to find the optimal value for θ according to model specific criteria. This selection is done in different ways for different models. One example of this process, known as back propagation is discussed in Section 2.1.10.

2.1.5 Test Train Split

In machine learning one usually has some data set and a model that requires data to be trained. One generally also wants to be able to evaluate their model to get a sense of how well it can be expected to perform. Because a model is fitting itself to the data it is provided, it will naturally tend to make more accurate predictions on data points it was trained on than on new unseen data points. For this reason, if one wants an accurate evaluation of how the model will perform on unseen data then one must test it on data that it has not been trained on. In order to accomplish this the data set that is being worked with is generally split up into a training set and a testing set[4]. The training set is used to train the model and the testing set is used to evaluate the model. When referring to the test set, we will call its feature set X_{test} and its output variables y_{test}. Similarly when referring to the training set, we will call its feature set X_{train} and its output variable y_{train}.

2.1.6 Cross Validation

When training a machine learning model, one wants to have as large a training set as possible. However, to provide an accurate report on how the model will perform on out of sample data some portion of the available data must not be trained on in order to evaluate the model. One way to deal with this issue is k-fold cross validation. This process consists of breaking up the available data into k sections for some integer k[5]. For example if one has 100 samples in their data set, and chooses to do cross validation with k=4, the broken up data would consist of elements 1 through 25, 26 through 50, 51 through 75, and 76 through 100. Then k different models are trained with each model being trained and evaluated on a different subset of data. The first model is trained on the data consisting of all the elements of the first k-1 folds and then evaluated on the kth fold. The second model is trained on the data consisting of all the elements of the first k-2 folds and the kth fold and then evaluated on the k-1 fold and so on for all k models. Continuing the previous example the first model would be trained on samples 1 through 75 and evaluated on samples 76 through 100. The second model would be trained on samples 1 through 50 and 76 through 100 and evaluate on samples 51 through 75, and so on. This process allows for the use of the entire available data set to be used for both training and testing without ever testing a model on data that it has been trained on. Figure 2.1 shows the breakup of training and testing data over the different folds of cross validation. The choice of k represents a trade off between the amount of data that any one model will be trained on, and the amount of data that will be available for evaluation along with the number of models one has to train. However, this process has its drawbacks. Obviously it is more computationally expensive to train several rather than one model. Additionally once you have several models you must have a way to select a single one

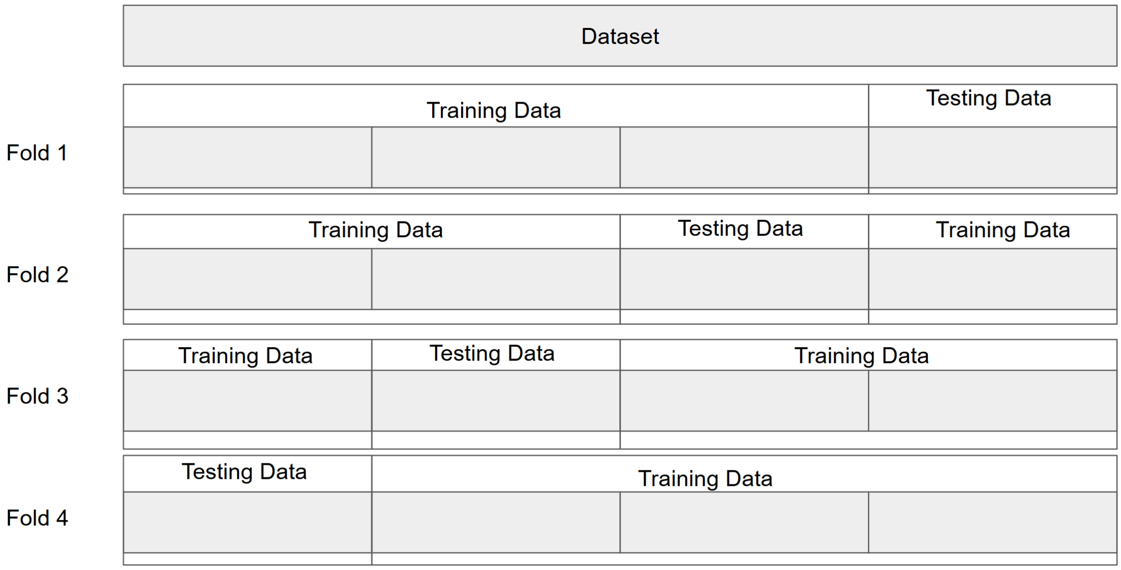


Figure 2.1: *A visualization of the test and train data under the different folds of cross validation. As can be seen the entire data set can be used for both testing and training without ever testing on the same data your training on.*

or aggregate the results of all the models if you want to make a single prediction. This can add complexity to the problem beyond the single model solution, but in the context of research where the goal is simply to compare how various models perform, having several models can also be an advantage.

2.1.7 Overfitting

Overfitting is when a model learns to predict the data set it was trained on very precisely at the expense of accuracy on new observations[4]. For example consider Figure 2.2. The data points shown as blue dots were generated by taking the identity function and randomly increasing or decreasing the output by a small amount. The red line shows a linear model fit to the data and the blue line shows an eight degree polynomial fit to the same data. The blue line is an example of an overfit model. It is perfectly accurate on all the data points that were provided to it, but if I continued to generate data using the same method this model would perform very poorly due to all the extreme shifts up and down to perfectly hit all the data it was provided. The linear model on the other hand despite having clearly worse performance on the data it was trained on would perform quite well on newly generated data.

2.1.8 Linear Regression

One of the simpler approaches to solving a regression problem problem is to approximate $f_{approximate}(X^i)$ as a linear combination of the input features, $f_{approximate}(X^i) = C_0 + C_1X^i_1 + ... +$

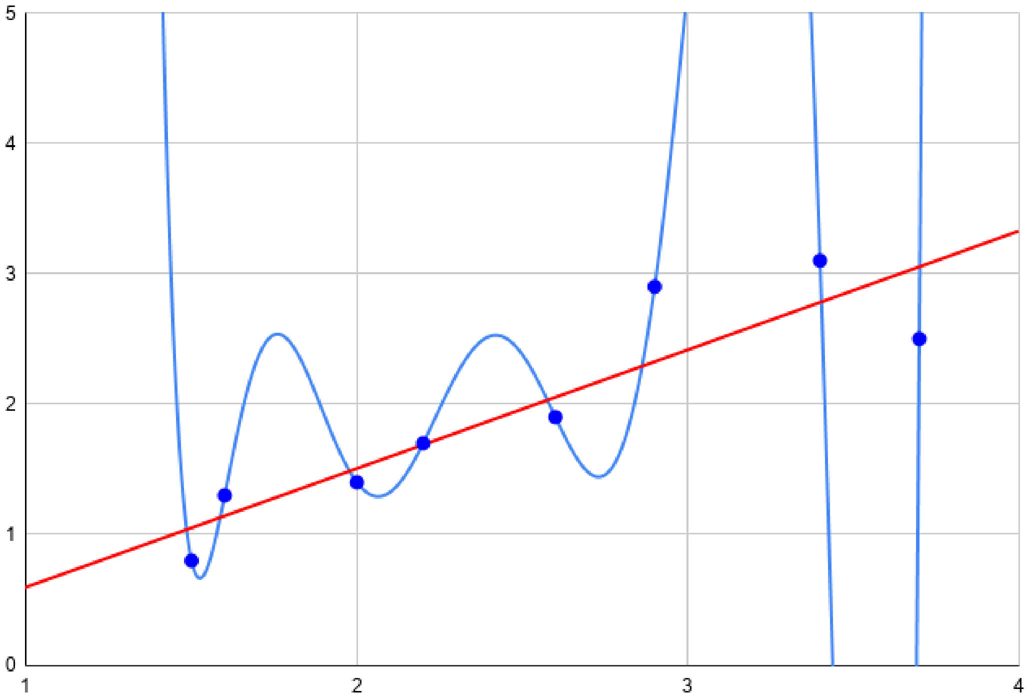


Figure 2.2: *An example of overfitting. The blue dots are a set of data points generated by adding some random noise to the identity function. The red line is a linear model fit to the data and the blue line is an eighth degree polynomial fit to the data*

$C_M X_M^i$. Basic linear regression picks $(C_0,...,C_M)$ to minimize

$$\sum_{i=0}^{N}(y^i - f_{approximate}(X^i))^2 \tag{2.1}$$

that is to minimize the sum of the squared error over your entire data set[3]. This method has the flaw that when used on a data set with a large number of features, many of which are unlikely to be significant, it will generally make use of all the features in its prediction which can make it less accurate on new observations as it will not try and discard irrelevant variables.

2.1.9 LASSO Regression

Least absolute shrinkage and selection operator (LASSO)[6] is a method that operates very similarly to traditional linear regression but adds an additional constraint in an attempt to address some of the issues with traditional linear regression. LASSO picks $(C_0,...,C_M)$ to minimize

$$\sum_{i=0}^{N}\left(y^i - f(X^i)\right)^2 \quad \text{given} \quad \sum_{i=0}^{j}|C_i| \leq t \tag{2.2}$$

This can be equivalently expressed as picking $(C_0,...,C_M)$ to minimize

$$\sum_{i=0}^{N}\left(y^i - f(X^i)\right)^2 + \alpha\sum_{i=0}^{j}|C_i| \tag{2.3}$$

where α implies a t based on the data being used. In this form α serves as a sort of penalty weighting for the magnitude of the coefficients. This means that as you increase alpha you would expect to see the magnitude of the coefficients get closer to and eventually be forced to zero. This is known as the Lagrangian form[7] and can be a computationally easier problem to solve because it relies simply on minimizing a function without any constraints. By constraining the sum of the coefficients, LASSO is forced to select only those features that are the most useful predictors and effectively discard the rest by setting their coefficients to zero. This can result in models that only use a few of the available input features while still providing a somewhat accurate prediction, which can be useful when looking to determine which features are actually important in predicting the output variable[8][9]. However, since LASSO is restricting the size of the coefficients it is typical to shift and scale the inputs and outputs to LASSO so that they are centered around zero and have a variance of one so that LASSO does not have to use up the entirety of its constraint simply to get an output that is in the range of some data that is centered around some very large value.

2.1.10 Neural Networks

Neural Networks are a powerful tool for learning complex non-linear relationships between data which can be useful, especially within the realm of finance[10][11][12].A neural network is a form of machine learning model that is composed of a network of neurons. A neuron is a logical unit that takes in a set of inputs, computes a linear combination of them using the corresponding edge weights, adds the bias and then applies its activation function and returns the output[13]. The activation function of a neuron allows you to control the range of the output of a neuron and can change how easy it is for a network to learn different relationships[4]. Generally an activation function is nonlinear as this allows the network to potentially learn more complicated non-linear relationships. One commonly used activation function is the rectified linear unit (ReLU), g(x) = max(0,x)[4]. As can be seen in Figure 2.3, ReLU is 0 when the input is less than 0 and acts as the identity function otherwise. In practice this often works well as an activation function, but the exact reasons why are not known. The defining characteristics of a neuron are the number of inputs, n, the edge weights associated with each of those inputs, $W \in \mathbb{R}^{n\times 1}$, the bias, $b \in \mathbb{R}$, and an activation function, g. A neuron is essentially just a unit that computes [4]

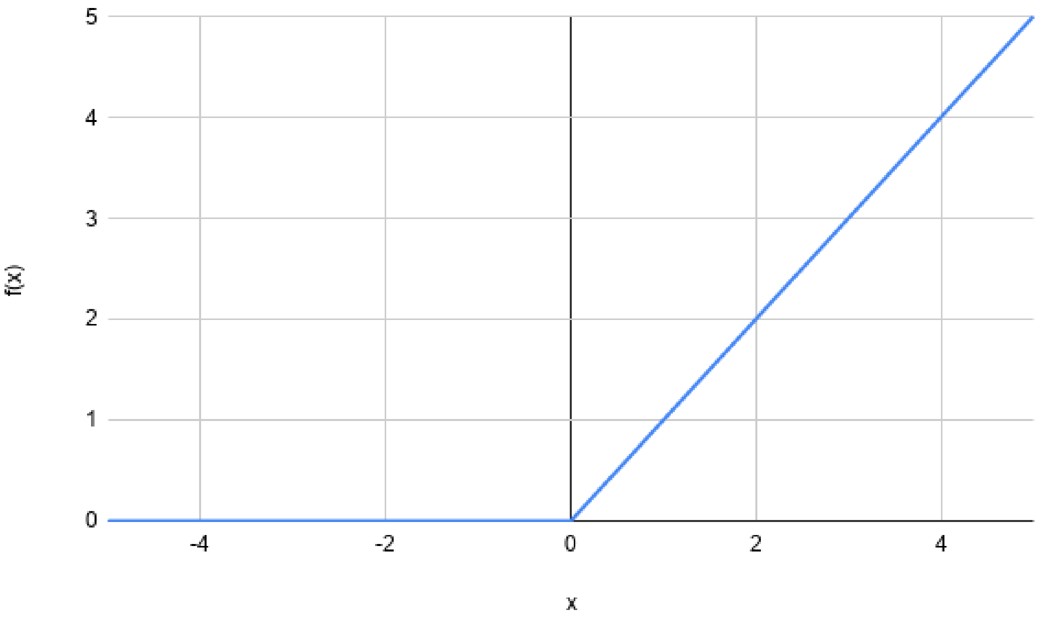


Figure 2.3: *A graph showing the ReLU activation function. The ReLU function acts as the identity function for $x \geq 0$ and is simply 0 for all $x < 0$*

$$\hat{y} = g\left(\sum_{i=0}^{n} w_i x_i + b\right) \tag{2.4}$$

A simple visualization of a neuron can be seen in Figure 2.4. The inputs are multiplied by the edge weights, summed, the bias is added, and the result is fed through the activation function then output. Many of these neurons can then be joined together using the output of some neurons as the input to others. This is typically done by organizing neurons into layers and then connecting different layers to one another[4]. More complicated network will allow outputs from some layers to be used as inputs to previous layers, but in the simplest form, known as a feed-forward neural network, the layers are organized sequentially so that each layer's outputs are used as inputs only for the next layer[4]. The network architecture is then defined by the size and activation function of each layer with the input size of each neuron in a layer being defined by the number of neurons in the previous layer. In order to do the calculation for the outputs of the net, let j^i be the number of inputs for layer i, k^i be the number of neurons in the layer for layer i, g^i be the activation function for layer i, b^i be the bias for layer i , and M^i be a j^i by k^i matrix composed of stacking vectors contain the edge weights for each of the neurons in the layer. Then for some input matrix X where each row is a set of inputs, we can generalize Equation 2.4 to find the corresponding set of outputs for a layer is

$$l^i = g^i\left(XM^i + b^i\right) \tag{2.5}$$

We can then find the output of the network by repeatedly applying this for each layer. For a network with d layers

$$output = g^d\left(g^{d-1}\left(...g^1\left(g^0(XM^0+b^0)M^1+b^1\right)...\right)M^{d-1}+b^{d-1}\right) \tag{2.6}$$

An example of such a network is shown in Figure 2.5. This example shows a network with four inputs than each feed into the first hidden layer, the results of the first hidden layer feed into the second hidden layer, and the results of the second hidden layer are used to generate the output. A hidden layer is a term used to refer to every layer except the inputs and the output. This term comes from the fact that the results from these layers are not directly seen at any point and are only used for calculating the results of the next layer. In order for a neural net to be a useful predictor good edge weights must be learned through training[4]. The most common form of training is known as back propagation[4]. Back propagation requires as inputs a neural network, a loss function, and a learning rate or optimizer. The loss function is just the error that the algorithm will actually be working to minimize. For regression typically it is mean squared error. The learning rate is a measure of how large the updates made to the edge weights will be during training. Larger learning rates mean larger changes which will make training the network faster but may lead to a higher error in the resulting model. However this trade off can somewhat be overcome as the learning rate need not be held constant throughout the entire process so rather than supplying a fixed learning rate one can supply an optimizer that dynamically determines the learning rate based on a number of factors observed during the training process[4]. This will generally allow the network to learn quickly, through a high learning rate, when the edge weights are far from optimal, and learn more slowly and precisely when it is approaching its desired end state. To begin the process edge weights are initialized randomly. The process then proceeds in a number of training steps, known as epochs. Each epoch consists of feeding forward, backpropagating, and updating. During feeding forward the inputs of the training data are fed into the network and propagated through all the layers to get the networks output. This output is then compared to the desired output to compute the loss. Then during the back propagation step that loss is propagated backwards through the network in order to calculate the derivative of the loss function with respect to the edge weight for each edge in the network. Then each edge weight is adjusted proportionally to its derivative with respect to loss and the learning rate. This process is then repeated until either a set number of epochs has passed or the improvements in the loss between epochs has gone below a certain threshold.

2.2 Finance

2.2.1 Terminology

Here I define some basic terminology that will be useful for discussing the topics in this section. Volatility is a way of measuring the spread of the returns for an asset or grouping of assets. High

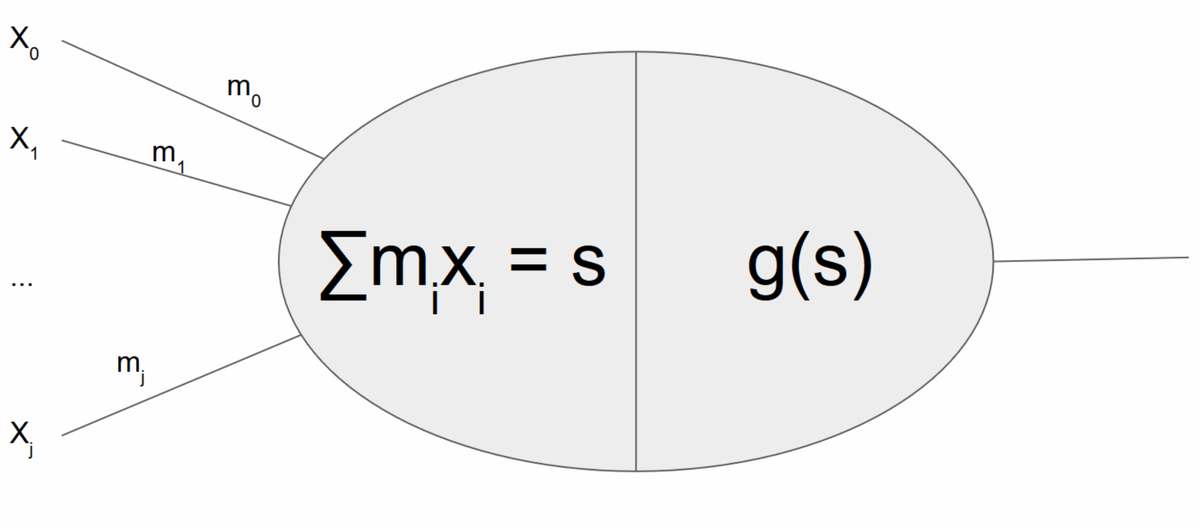


Figure 2.4: *Visualization of a single neuron of a neural net.*

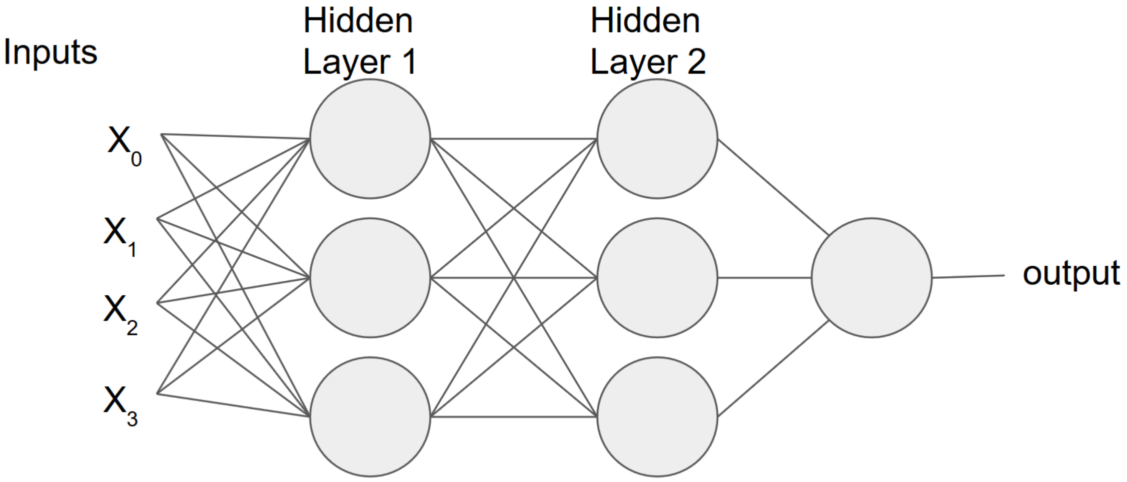


Figure 2.5: *An example of a neural net with four inputs, two hidden layers both of size three, and a single output. Edge weights and activation functions not shown to avoid clutter.*

volatility means that returns will have a larger magnitude, they may be positive or negative, but they will be large. Low volatility on the other hand means that returns will be comparatively small. Hedging is the act of protecting a position against extreme losses. Generally hedging includes activities such as holding some assets that would be expected to have high returns in the event of some disaster that will extremely devalue the other assets you are holding[14]. An index is a way of measuring some property of the market. The most common and well known indices track market returns on some group of assets. Arbitrage is a term that refers to an opportunity to make risk free profit in a market or across different markets[14]. The simplest example of arbitrage would be buying an asset in one market and simultaneously selling it in another market for a higher price. Most financial and pricing models will assume that there is no opportunity for arbitrage in the market.

2.2.2 Bid Ask Spread

When assets are traded on the open market there can be many individual buying and selling the asset. People looking to buy the asset put out bids which are open offers to buy the asset at a set price. People selling will put out asks which are simply open offers to sell the asset at a set price. The bid ask spread is the difference between the lowest ask and the highest bid or more simply the difference between the price that people are selling for and the price that people are buying for.

2.2.3 Derivatives

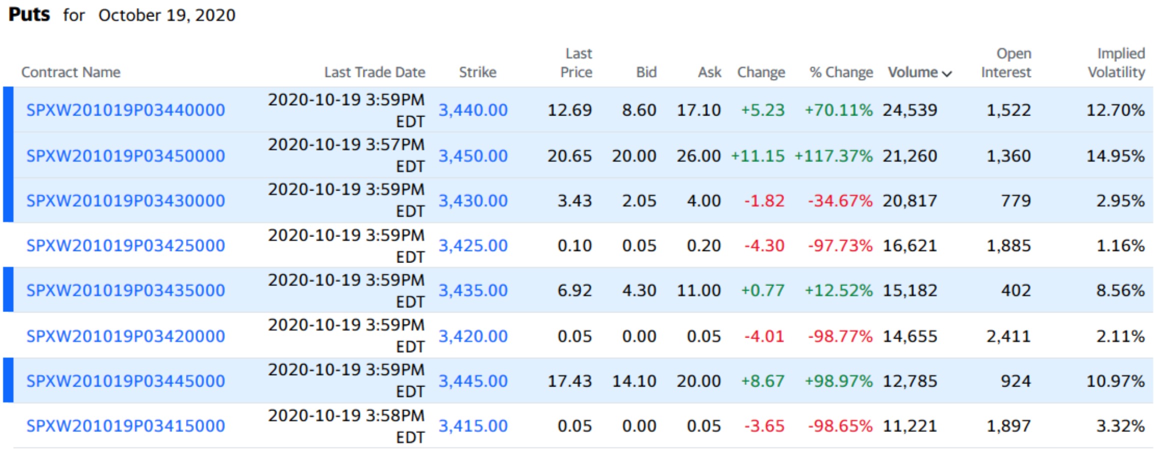
In finance, a derivative is an asset that derives its value from some other asset or index, known as the underlying[15]. Derivatives typically take the form of a potential or guaranteed exchange of the underlying at or before some fixed date, known as the expiration. Derivatives have a number of uses including hedging or allowing investors access to assets that would otherwise be hard to trade in. The most commonly used derivatives are futures and options discussed below.

2.2.4 Futures

Futures contracts are agreements to execute some sale of an asset at a fixed price, known as the forward price, on a fixed date, known as the expiration. One party agrees that they will purchase the underlying at the forward price on the expiration[15]. The price of a future for which the underlying is easily tradable and has no storage costs is quite simple to determine with some basic arguments. Consider an investor who buys one underlying for price U and sells a future with a forward price S on that asset for t time in the future. On the expiration data the investor will give up their asset and get the forward price in return, thus the value at that time will be S minus the value of the amount the investor paid for that asset in the first place accounting for the increase in value under the risk free interest rate, r i.e. $S - Ue^{rt}$. To get the present value of that future we can then simply discount this using the risk free interest rate to get the current value of the future i.e. $Se^{-rt} - U$. Simply put the value of a future is the difference between the present value of the strike price and the value of the underlying[16].

2.2.5 Options

Options are simply contracts that allow an individual to buy or sell some underlying at a set price, known as the strike price, by or on a given date, known as the expiration. Options that allow the holder to buy the underlying are known as call options and options that allow the holder to sell the underlying are known as put options. Options are, as their name implies, optional unlike futures. The holder has the right but not the obligation to make use of them. To make use of the right given to you by an option is known as exercising the option. When an option can be exercised depends on the specifics of the option, but there are two general categories. European options can be exercised on the expiration to buy or sell the underlying at the strike. American options can be exercised at any point before the expiration to buy or sell the underlying at the strike[15]. Options can generally be divided into three categories, in-the-money, at-the-money, and out-of-the-money depending on their strike relative to the underlying. Puts with a strike above the underlying and calls with a strike below the underlying are known as in-the-money options. These are options that it would make sense to exercise immediately if one could as they would allow one to sell above or buy below market price. Options with a strike equal to the current price of the underlying are known as at-the-money options. Puts with a strike below the price of the underlying or calls with a strike above the price of the underlying are known as out-of-the-money options[14]. These are options that it would not make sense to exercise immediately if one could, and they are generally used in order to hedge positions. The pricing of options is a complicated field and a number of models for finding fair prices of options exist. One popular model known as the Black-Scholes model is discussed in Section 2.2.10. However, with some very simple assumptions, one can determine the relationship between the price of a put and a call with the same strike and expiration date without specifying a concrete model. To do this consider a profile where one buys a call and sells a put with the same strike, s, and expiration date, t. At expiration if the value of the asset is above the strike price, then the call will be exercised and the asset purchased at the strike price. If the value of the asset is below the strike price, then the sold put will be exercised and the asset will be purchased at the strike price. Notice that this is equivalent to owning a future with a future price equal to the strike price and the same expiration. Therefore, at any time, the value of the difference between a call and put with the same price and expiration is the same as the value of a future with a future price equal to the strike and the same expiration. This property is known as the put-call parity[14].



Puts for October 19, 2020

Contract Name	Last Trade Date	Strike	Last Price	Bid	Ask	Change	% Change	Volume	Open Interest	Implied Volatility
SPXW201019P03440000	2020-10-19 3:59PM EDT	3,440.00	12.69	8.60	17.10	+5.23	+70.11%	24,539	1,522	12.70%
SPXW201019P03450000	2020-10-19 3:57PM EDT	3,450.00	20.65	20.00	26.00	+11.15	+117.37%	21,260	1,360	14.95%
SPXW201019P03430000	2020-10-19 3:59PM EDT	3,430.00	3.43	2.05	4.00	-1.82	-34.67%	20,817	779	2.95%
SPXW201019P03425000	2020-10-19 3:59PM EDT	3,425.00	0.10	0.05	0.20	-4.30	-97.73%	16,621	1,885	1.16%
SPXW201019P03435000	2020-10-19 3:59PM EDT	3,435.00	6.92	4.30	11.00	+0.77	+12.52%	15,182	402	8.56%
SPXW201019P03420000	2020-10-19 3:59PM EDT	3,420.00	0.05	0.00	0.05	-4.01	-98.77%	14,655	2,411	2.11%
SPXW201019P03445000	2020-10-19 3:59PM EDT	3,445.00	17.43	14.10	20.00	+8.67	+98.97%	12,785	924	10.97%
SPXW201019P03415000	2020-10-19 3:58PM EDT	3,415.00	0.05	0.00	0.05	-3.65	-98.65%	11,221	1,897	3.32%

Figure 2.6: *An example view of some S&P 500 put options taken from yahoo finance. Those highlighted in blue are in-the-money.*

2.2.6 The S&P 500

The S&P 500 (SPX) is an index made up of the stock prices of the 500 largest companies in US stock markets[17]. The effect of each of these stock prices is weighted based on the market capitalization of the company. The market capitalization of a company is the value of all its publicly traded stock. The S&P 500 is generally seen as a good indicator of how the overall market is doing.

2.2.7 The VIX

The Chicago Board Options Exchange Volatility Index, or VIX for short, is an index tracking the implied 30 day volatility of S&P 500 based on the prices of various SPX options[18]. Basically the VIX is a measure of how drastically people expect the market to change over the next 30 days. The VIX is calculated using out-of-the-money SPX calls and puts. To calculate the VIX one first needs the forward index level, F, which is derived from index option prices by finding the strike price with the least absolute difference between the call and put prices. The calculation of the VIX uses call options with strike prices greater than forward index level excluding those above two consecutive strikes with zero bid or ask and put options with strike prices less than forward index level excluding those below two consecutive strikes with zero bid or ask and excludes any options that are not being traded. Essentially this just means that to find all the puts used in calculating the VIX, you start at the strike price where the put and call prices are the closest and you work your way down the various strikes taking all those that have a bid and ask both above zero. If you come across a strike where the bid or the ask are zero you skip it unless the previous strike also had a bid or ask of zero in which case you stop and do not take any more strike prices into consideration. Repeat this process for the calls going up in strike price. These two sets of options are those that are used in the calculation of the VIX in the following formula.

$$\sigma^2 = \frac{2}{T}\sum_i \frac{\Delta K_i}{K_i^2} e^{RT} Q(K_i) - \frac{1}{T}\left[\frac{F}{K_0} - 1\right]^2 \tag{2.7}$$

$$\Delta K_i = \frac{K_{i+1} - K_{i-1}}{2} \tag{2.8}$$

The VIX value is $100 * \sigma$. T is time to expiration expressed in minutes. Using the strike $F = StrikePrice + e^{RT} * (CallPrice - PutPrice)$. K_0 is the first strike below F and K_i is the ith out of the money option. R is the risk free interest rate. $Q(K_i)$ is the mid point of the bid-ask spread for each option with strike K_i.[18]

2.2.8 Futures and VIX Futures

The VIX is not a directly tradable asset meaning that it is not something one could go and purchase or sell. However, there was a large enough demand for a way to directly trade in the VIX and allow investors to hold volatility in their account that VIX futures were introduced in 2004[18]. VIX futures contracts are a way of trading directly in the value of the VIX. A VIX future is simply a contract with an expiration date that says the holder will be paid money equal to the difference between value of the VIX at the expiration date and the value of the VIX at purchase, which differs from other futures contracts only because the VIX cannot be directly purchased so its value is exchanged instead. These contracts are useful because they allow traders to trade directly in market volatility which can be useful for hedging a position as poor market conditions tend to lead to high volatility.

2.2.9 Delta and Gamma

Delta is a financial variable used to measure the change in the price of a derivative given a change in the price of the underlying[14]. For example, a delta of 0.5 for an option on some stock means that if the price of that stock increased by $1, the price of the option would increase by $0.5. Gamma is simply the change in delta for a given change in the price of the underlying[14]. For example, if an option has a gamma of .2 for an increase in the price of the underlying by $1, we would see an increase in delta of .2.

2.2.10 Black-Scholes Model

In order to calculate delta and gamma one generally needs a financial model to derive such values. The Black-Scholes or Black-Scholes-Merton is a model for pricing European options based on something called the risk neutral argument which involves buying and selling the underlying and the option in a very specific way to eliminate risk[16]. The full theoretical basis of the model is beyond the scope of this paper, but the model gives us the following value of a call option at time t

$$C(S_t,t) = N(d_1)S_t - N(d_2)\left(Ke^{-r(T-t)}\right) \tag{2.9}$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right] \tag{2.10}$$

$$d_2 = d_1 - \sigma\sqrt{T-t} \tag{2.11}$$

where t is the time in years until expiry, T is the expiry, S_t is the price of the underlying at time t, $C(S_t,t)$ is the price of the option at time t, σ is the standard deviation of the stock's returns, K is the strike price, r is the risk free interest rate, and $N(.)$ is the standard normal cumulative distribution[19]. From the put-call parity, the value of a put can be calculated as

$$P(S_t,t) = N(-d_2)Ke^{-r(T-t)} - N(-d_1)S_t \tag{2.12}$$

Delta can then be calculated as the first derivative of value of the option with respect to the value of the underlying. For calls this gives us

$$N(d_1) \tag{2.13}$$

and for puts this gives us

$$N(d_1) - 1 \tag{2.14}$$

Gamma can then be calculated by taking the derivative of delta with respect to the price of the underlying. This gives us the same value for both puts and calls which is

$$\frac{N'(d_1)}{S\sigma\sqrt{T-t}} \tag{2.15}$$

2.3 Literature Overview

Hosker et al. compared a number of traditional models, such as linear regression, to a number of more advanced machine learning models, including various forms of neural networks, for forecasting VIX futures[20]. For inputs they used term structure spreads, which is way of measuring the difference in volatility over different time spans, high and low intraday prices for various other futures contracts, skew, which is a way of measuring the expectations of a downside event, some technical variables, which included things like a single for when the value crossed the prior 14 moving average, and the VVIX, which is an index calculated exactly like the VIX only using VIX options rather than S&P 500 options, to attempt to forecast the one month VIX futures contract that expires next both three and five days into the future. A number of models were built including linear regression, LASSO, random forest, and recurrent neural networks(RNN), which are neural networks that operate like feed forward neural networks but have some outputs from some layer fed backwards as inputs to previous layers. They found that more advanced machine learning techniques particularly RNN performed better than traditional methods like linear regression. Michael Yu used historical VIX and S&P500 prices as inputs to predict the direction of movement of the VIX, i.e. an increase or decrease, over the following 1 to 6 weekdays[21]. Yu used a gradient boosted decision tree as a model and was able to get accuracies between 55% and 65%. Decision trees are a machine learning method that builds a model that asks a series of questions to classify the inputs. The question at each stage is dependent on the answer to the previous questions, so evaluating the model can be viewed as descending down a tree where selecting the child to go to is answering the question. Gradient boosting is a method for training and combining a series of models into a single better models. It works by training the first model, finding what portion of the data that model performs the worst on, training another model to be better on that data, and repeating the process until the desired number of models is reached. Ballestra et al. used a feed forward neural net to predict open to close returns on VIX futures based on previous prices of VIX futures and several indices from Asian markets[22]. It was found that the neural net out performed linear regression in predicting the movement of VIX futures.

Chapter 3

Methodology

3.1 Data

Our data consists of the bid, ask, and trade prices for SPX options at all available strikes that expire between 23 and 37 days in the futures, the values of the SPX and VIX, and the bid, ask, and trade prices for monthly VIX futures all gathered from Bloomberg as bars with a length of one minute from the first week of August 2020. The close price from these bars was then taken to be the value of that asset at that time. . Delta and gamma for each option were then calculated and added to the data set using the Mibian implementation of the Black-Scholes model for option pricing. Due to lack of trading not all strikes had values at each time step. To fill in the gaps any price that had no value at a given time step was assumed to be priced the same as the previous time step. This left undefined only those values for which all previous values were also undefined. For some extremely infrequently traded options this was still a large part of the data set. To deal with this each strike was evaluated to see how often it appeared without a value in the data set. All those without values in at least 5% of the data set were dropped. From the resulting data set any time step for which we did not have values for each feature was dropped. This process reduced the size of our data set from 3128 entries to 2577 entries. These features were then changed to being defined relative to the value of the SPX rather than as raw values. So each option price was replaced with that option price divided by the value of the SPX at that time. These features were then aligned such that each feature rather than representing a particular strike instead represented a strike relative to the underlying, .i.e. a feature was the relative price of the first option above the strike rather than an option with a strike price of 3000. We then went through and used Black-Scholes to calculate the delta and gamma of each option at each time step. This left us with a data set of 1752 entries of 333 features.

3.2 Methodology

Our methodology consisted of evaluating two major model types using several different feature sets for multiple output variables. The major model types were LASSO regression and a feed forward neural network. As input features set we started using the prices of all available SPX options and the delta and gamma associated with each of those. We then also tested the top five features from that set as selected by LASSO and we tried adding the value of the VIX at each time step. For output variables we tried predicting both the values of VIX futures and the VIX itself at that same time step. All evaluations were done using cross fold validation with 5 folds. For the LASSO model, we used sklearn's implementation and trained the model using a variety of different values for the regularization parameter alpha varying for 0 to 2 by increments of .2. Before training the inputs and outputs were regularized using sklearn's preprocessing scale method to give them a mean of zero and a standard deviation of 1. In order to extract the features that the LASSO models used as predictors, we took each feature and found the average of the coefficients associated with that feature by each training of LASSO. For the neural network, all models were built using keras out of a number of dense layers. The optimizer was adam, the loss was MSE, and the activation function was ReLU for all tested models. Each model was trained for 50 epochs. We tested a number of different neural net configurations varying both the depth of the network and the size of the individual layers. For depth, we tried models with 1, 6, and 20 hidden layers. For layer size, we tried models with 5 and 136 units per layer. We train these models on five different input-output combinations. We trained two sets of models using a feature set consisting of the prices of each of the options and the delta and gamma associated with those prices. The first set of models was predicting the price of the VIX and the second was predicting the price of VIX futures. We then trained two sets of models using a feature set consisting of the prices of each of the options, the delta and gamma associated with those prices, and the VIX. The first set of these models was also predicting the VIX and the second set was predicting VIX futures. The final set of models was trained using a feature set consisting of the top five most important features as selected by LASSO to predict the prices of VIX futures. In order to produce a more intuitively meaningful evaluation of the results of training the models, for each model a graph of the percent error on the test and training set were generated. To generate these graphs, the trained models were taken and used to generate predictions on their testing and training sets which were then converted into percent errors. The percent errors for each of the testing sets were all combined into one large set and the percent errors for the training set were all combined into one large set. A frequency distribution of each of these sets was then generated.

Chapter 4

Results

4.1 LASSO

For the LASSO model, as we can see in Figure 4.1 increased alpha predictably leads to an increased error for alpha from zero up until around .8 when we see error stay more or less the same for the rest of the alphas. We also see quite a wide spread of errors, especially for higher values of alpha. For example at alpha of 1.0 we see an MSE between around .25 and almost 3.0. We can also see that over the different values of alpha our MSE's form three distinct lines and a set of two points that stay relatively close to one another. This likely indicates that the different folds used for cross validation have some differences in the data they contain. For example, the highest line of values likely is the result of the fold that that was being tested against has some more pronounced differences between itself and the original data set then other folds. As we increase alpha we also see a decrease in the number of features used in our prediction. Figure 4.2 shows the number of features required by the LASSO model to make its prediction at each value for alpha. As expected a higher alpha leads to a lower number of features being selected with an alpha of above 1.2 selecting no features to use implying that the model is just making a static prediction.

4.2 Neural Net

4.2.1 Using the Prices of Options, Delta and Gamma to Predict the VIX

The first set of models was trained to predict the VIX from the prices of options and the deltas and gammas associated with those prices. The graph of test loss over the training period for the model with six hidden layers of size five can be seen in Figure 4.3. This shows the general expected

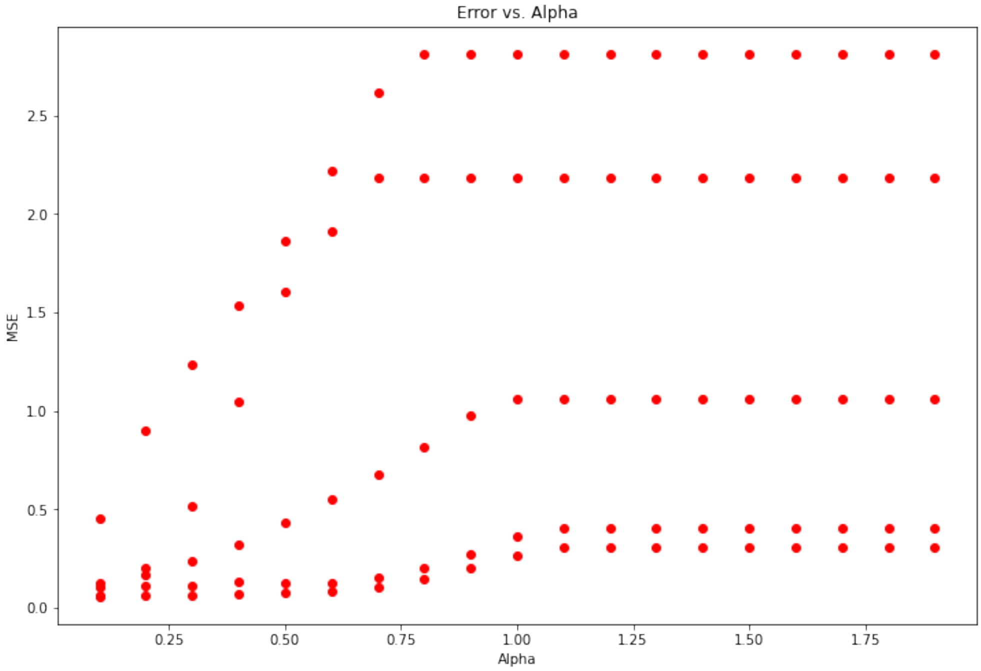


Figure 4.1: *MSE of LASSO model on test set as a function of alpha. There are five data points at each value of alpha representing the five different folds in cross validation.*

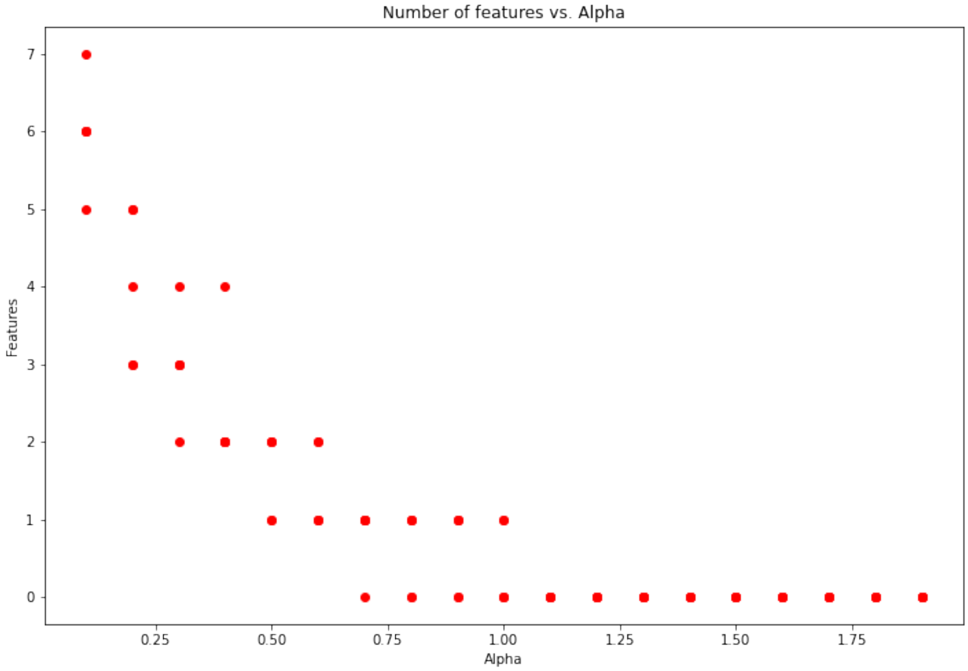


Figure 4.2: *Number of features selected by each LASSO model as a function of alpha. A feature was considered to have been selected by alpha if its coefficient was above .0001*

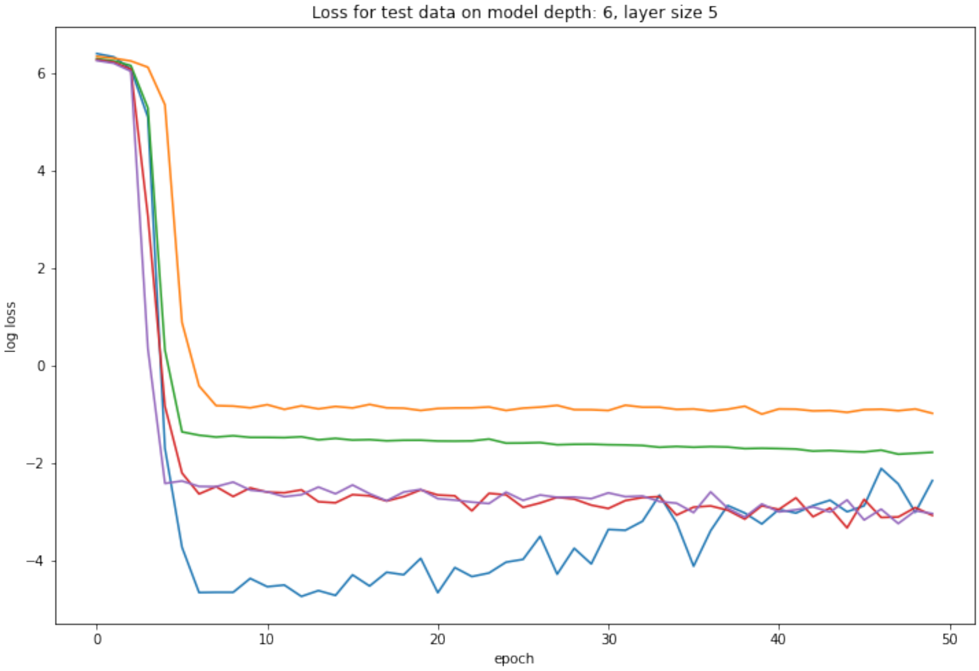


Figure 4.3: *The natural log of the test loss of the model with six hidden layers of size five trained to predict the VIX from the prices of the options and their associated deltas and gammas over the training period. The different colored lines represent the different folds for 5 fold cross validation.*

behavior over the training period as we see the loss start very high and drop very quickly for a short period then level off. We can also see that the fold represented by the blue line probably began to see some slight overfitting as the error is slightly increasing over the last 30 epochs. This effect is even more pronounced in Figure 4.4 where you can see the yellow fold hits a low loss before increasing rather significantly and holding steady. This behaviour may have been able to be prevented by reducing the training time but, this may not have given other models enough time to finish learning. This can be seen in Figure 4.4 as the yellow fold's jump in loss occurs while the blue fold's loss is still in the process of dropping quickly. The losses for most models follow the general form of Figure 4.3.

To make the error of the models a little easier to interpret Figures 4.6 and 4.5 show the distributions of the percent errors on predictions made by the model on the training and testing set respectively for the model with six hidden layers of size five. These graphs show most of our errors to be within the range of about 4% and show as we would hope that our errors on the test set look very similar to our errors on the training set. This indicates that the potential overfitting we spotted in the graphs of the training process has not had a huge impact on our results as overfitting would lead to errors on the training set that are much closer to zero than those in the testing set. Distributions of percent error for other models looked similar, all looking like a noisy normal distribution centered somewhere near 0. The average absolute percent error of each model is summarized in table 4.1.

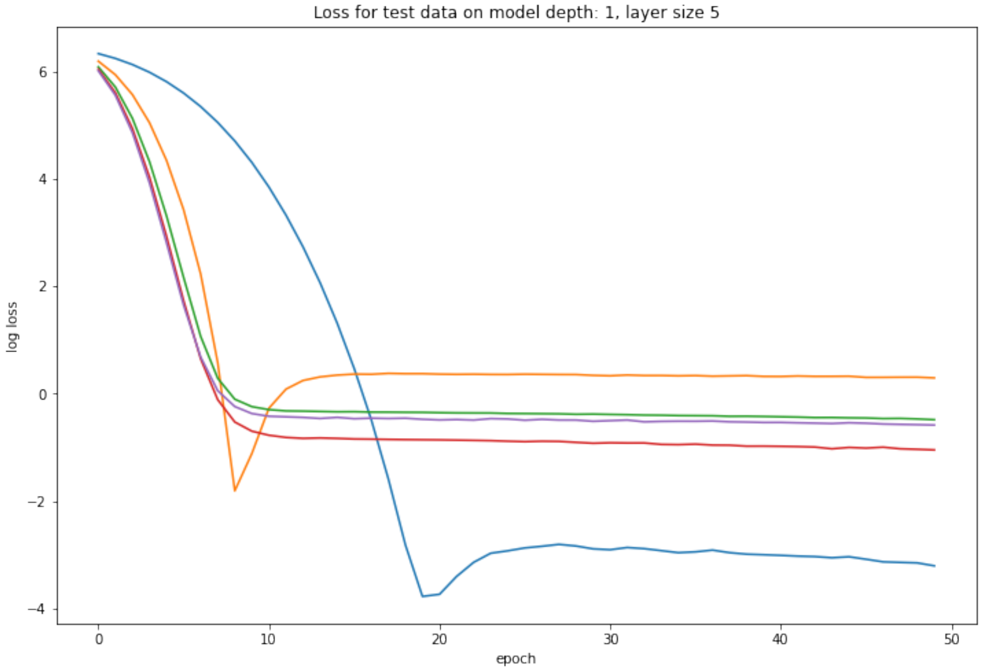


Figure 4.4: *The natural log of the test loss of the model with a single hidden layer of size five trained to predict the VIX from the prices of the options and their associated deltas and gammas over the training period. The different colored lines represent the different folds for 5 fold cross validation.*

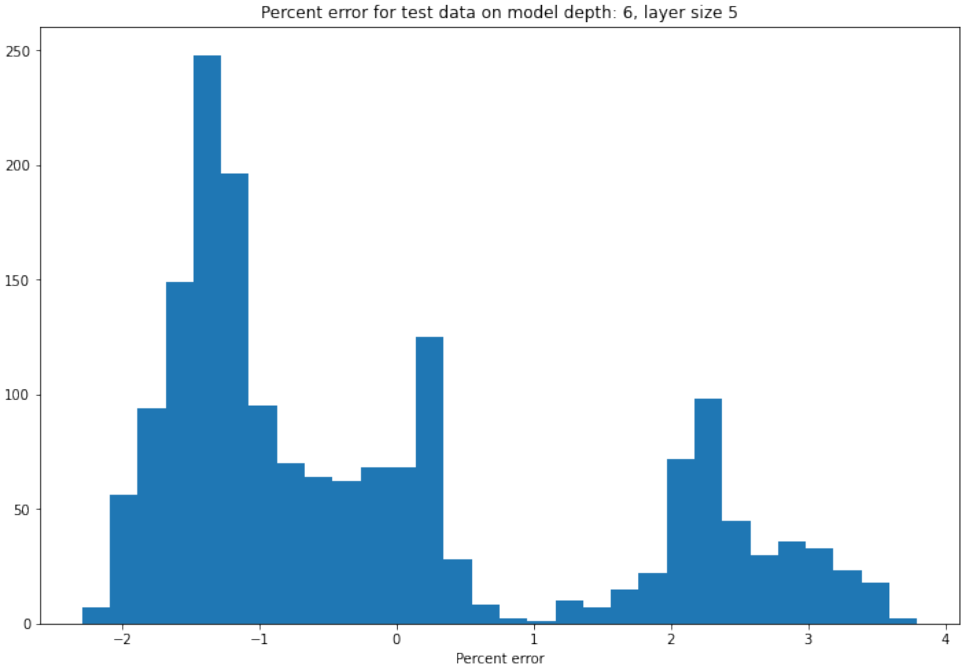


Figure 4.5: *The distribution of the percent errors on the test data of the model with six hidden layers of size five trained to predict the VIX from the prices of the options and their associated deltas and gammas over the training period.*

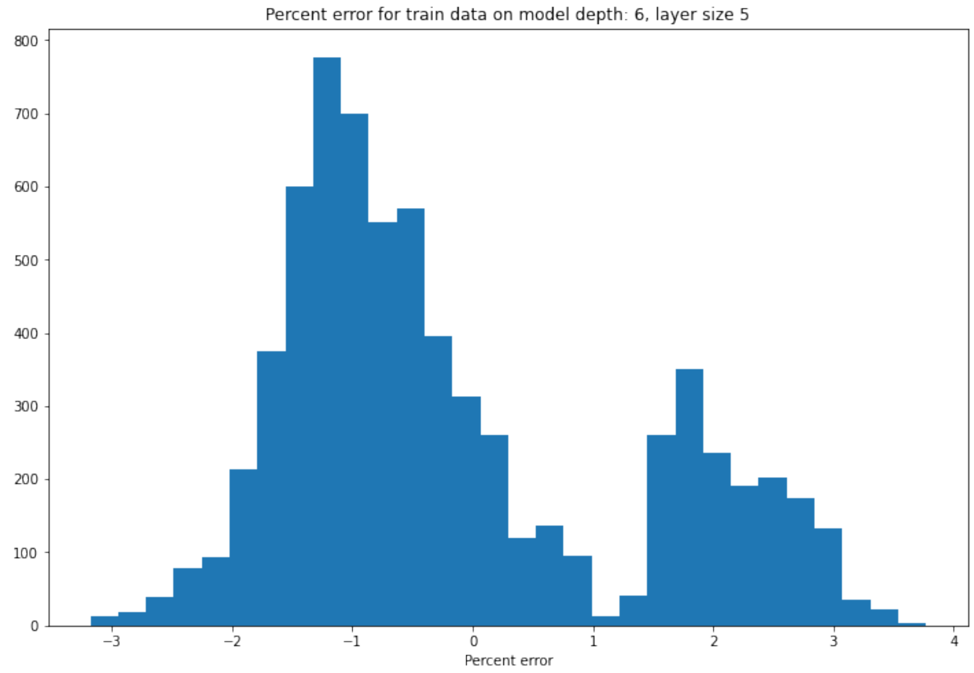


Figure 4.6: *The distribution of the percent errors on the train data of the model with six hidden layers of size five trained to predict the VIX from the prices of the options and their associated deltas and gammas over the training period.*

		Depth		
		1	6	20
Layer Size	5	2.8%	1.5%	3.1%
	136	1.8%	1.2%	1.2%

Table 4.1: *Average absolute percent error on test predictions of models trained on the features set consisting of the options prices and the deltas and gammas to predict the VIX.*

4.2.2 Using the Prices of Options, Delta and Gamma, and the VIX to Predict the VIX

The second set of models was trained to predict the VIX from the prices of options, the deltas and gammas associated with those prices, and the VIX. This set of models has the VIX both as a feature and as the output so we would expect them to be very accurate and they theoretically could be perfectly accurate. The graph of test loss and the over the training period for the model with six layers of size 5 can be seen in Figure 4.7. This shows the general behavior of all models in this set. Very similarly to the models in the previous set, we see a large drop off followed sometimes by a short increase then a leveling off. The percent error on testing and training sets for the model with six hidden layers of size five can be seen and Figures 4.8 and 4.9. Again these graphs are very similar across all models so just one has been shown as an example. Table 4.2 summarizes the average absolute percent error of all the models in this set. The most notable

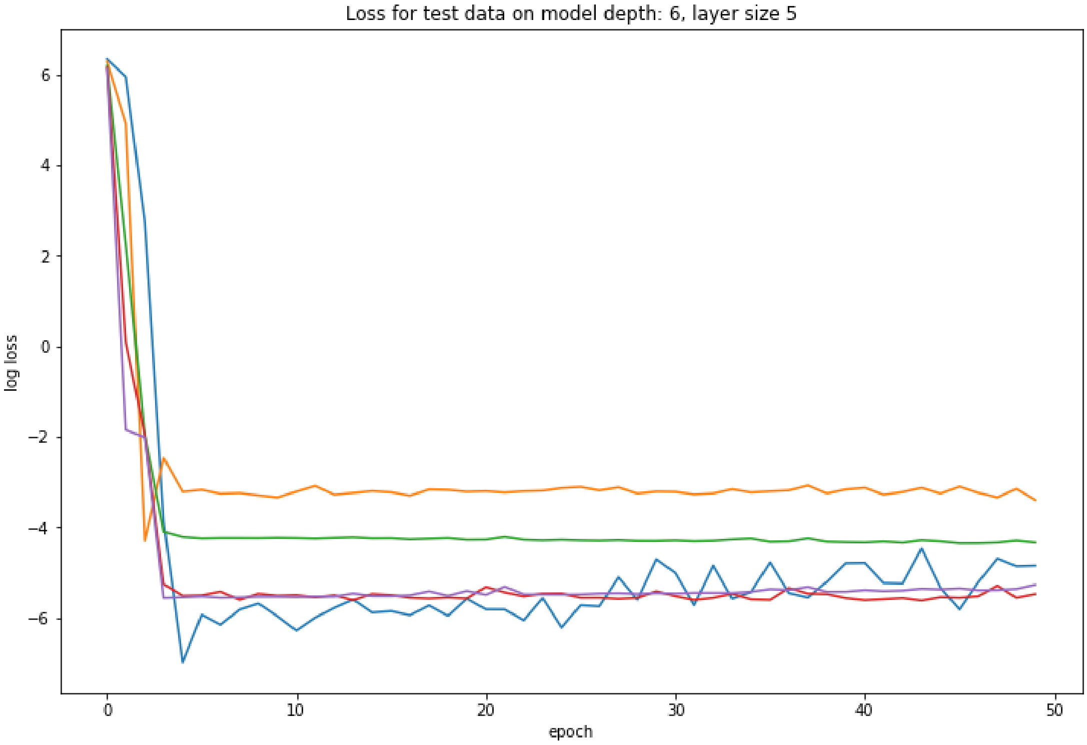


Figure 4.7: *The natural log of the test loss of the model with six hidden layers of size five trained to predict the VIX from the prices of the options, their associated deltas and gammas, and the VIX. The different colored lines represent the different folds for 5 fold cross validation.*

		Depth		
		1	6	20
Layer Size	5	0.7%	0.4%	2.4%
	136	0.4%	.14%	1.0%

Table 4.2: *Average absolute percent error on test predictions of models trained on the features set consisting of the options prices, the deltas and gammas, and the VIX to predict the VIX.*

information in this table is that the model with twenty hidden layers of size 5 seems to perform much worse than all the other models. Comparing these results to the results from the previous set of models, we see that unsurprisingly adding the VIX to the feature set improves our accuracy for predicting the VIX.

4.2.3 Using the Prices of Options, Delta and Gamma to Predict VIX Futures

The third set of models was trained to predict VIX futures from the prices of options and the deltas and gammas associated with those prices. The graph of test loss and the over the training period for the model with six layers of size 5 can be seen in Figure 4.10. Again we see the same behavior, a sharp drop off followed by a leveling out. All models in this set followed this general

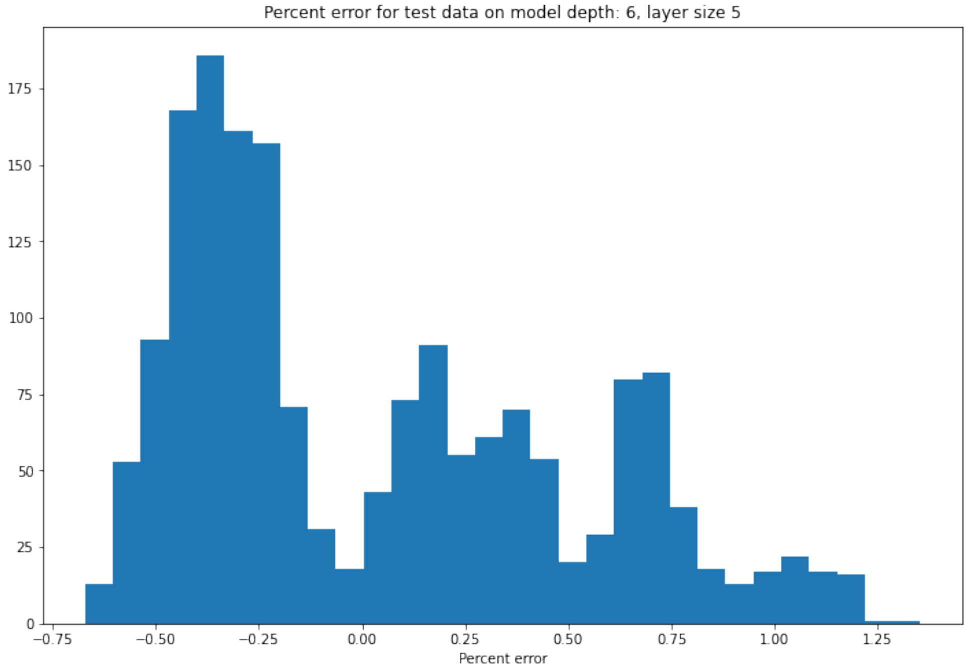


Figure 4.8: *The distribution of the percent errors on the test data of the model with six hidden layers of size five trained to predict the VIX from the prices of the options and their associated deltas and gammas, and the VIX.*

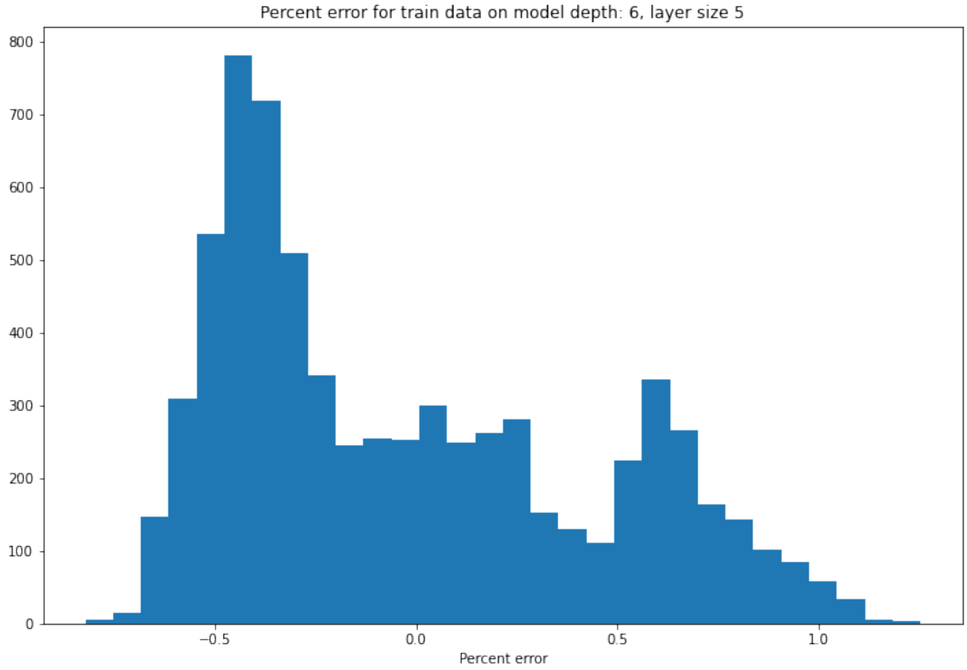


Figure 4.9: *The distribution of the percent errors on the train data of the model with six hidden layers of size five trained to predict the VIX from the prices of the options, their associated deltas and gammas, and the VIX.*

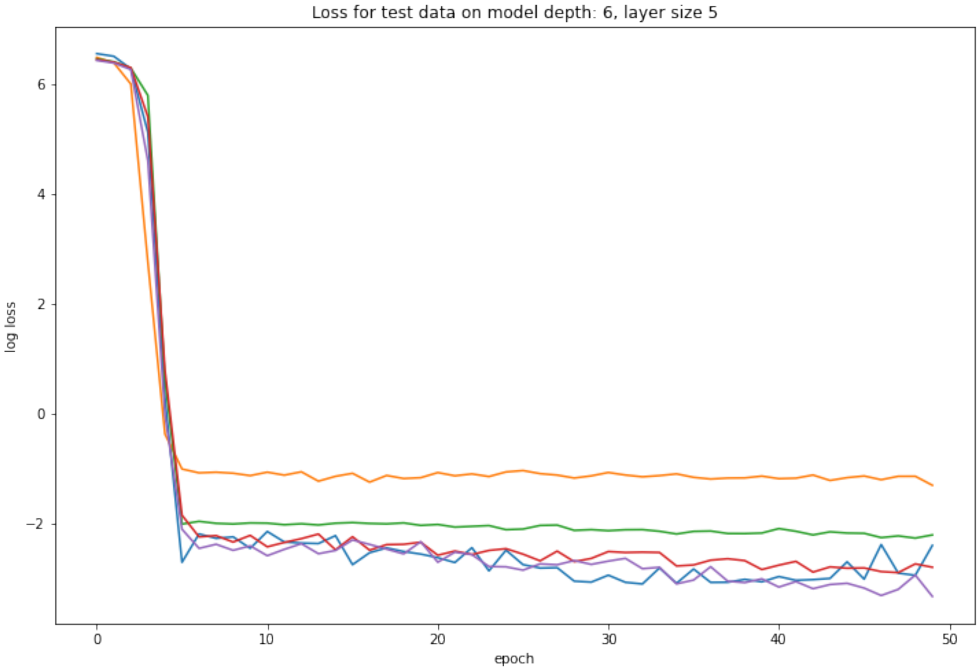


Figure 4.10: *The natural log of the test loss of the model with six hidden layers of size five trained to predict VIX futures from the prices of the options, their associated deltas and gammas, and the VIX. The different colored lines represent the different folds for 5 fold cross validation.*

		Depth		
		1	6	20
Layer Size	5	3.6%	1.1%	1.8%
	136	2.3%	1.2%	1.2%

Table 4.3: *Average absolute percent error on test predictions of models trained on the features set consisting of the options prices and the deltas and gammas to predict VIX futures.*

behavior. The percent error on testing and training sets for the model with six hidden layers of size five can be seen and Figures 4.11 and 4.12. Again these graphs are very similar across all models so just one has been shown as an example. Table 4.3 summarizes the average absolute percent error of all the models in this set. Comparing this set of models to the first set of models we see that the change from predicting the VIX to predicting VIX futures does not seem to have largely affected the error of the models. This is not terribly surprising as the prices of VIX futures tend to be pretty closely correlated with the VIX so we would not expect them to be significantly harder or easier to predict.

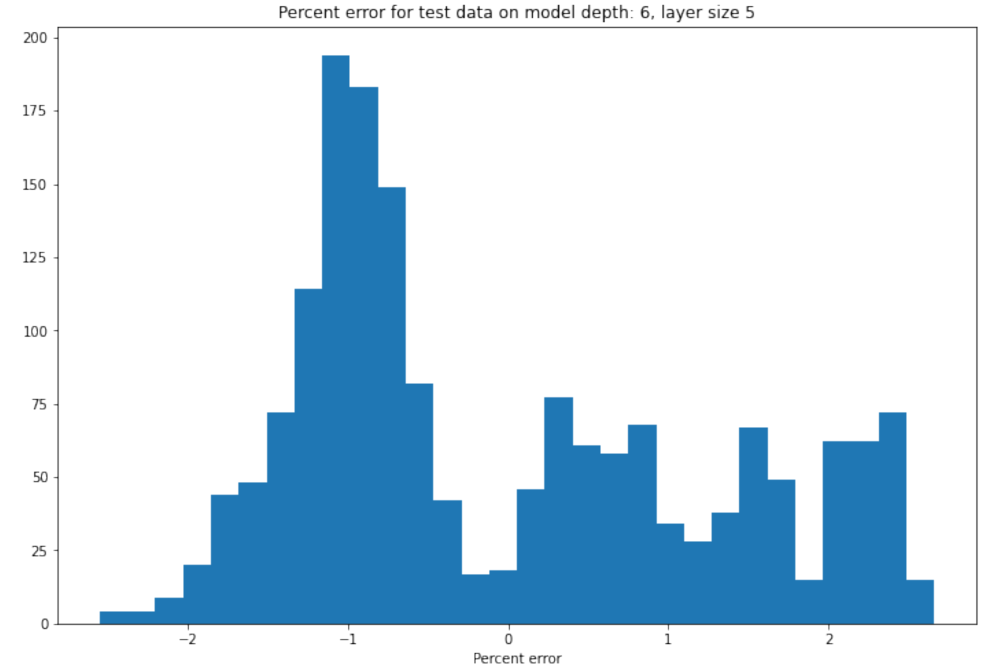


Figure 4.11: *The distribution of the percent errors on the test data of the model with six hidden layers of size five trained to predict VIX futures from the prices of the options and their associated deltas and gammas, and the VIX.*

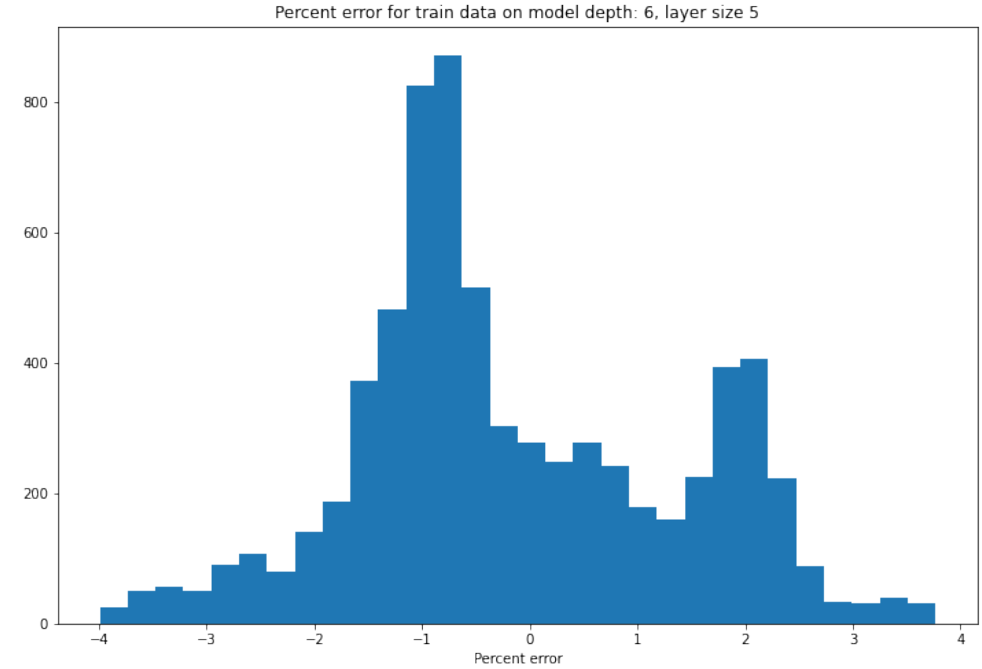


Figure 4.12: *The distribution of the percent errors on the train data of the model with six hidden layers of size five trained to predict VIX futures from the prices of the options, their associated deltas and gammas, and the VIX.*

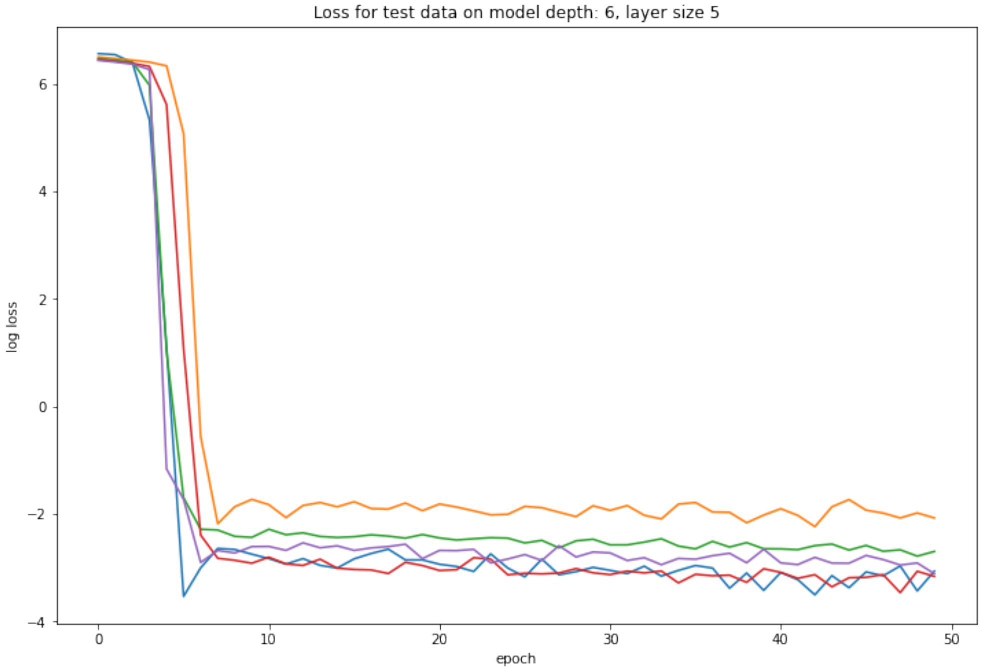


Figure 4.13: *The natural log of the test loss of the model with six hidden layers of size five trained to predict VIX futures from the prices of the options, their associated deltas and gammas, and the VIX. The different colored lines represent the different folds for 5 fold cross validation.*

4.2.4 Using the Prices of Options, Delta and Gamma, and the VIX to Predict VIX Futures

The fourth set of models was trained to predict VIX futures from the prices of options, the deltas and gammas associated with those prices, and the VIX. The graph of test loss and the over the training period for the model with six layers of size 5 can be seen in Figure 4.13. Again we see the same behavior, a sharp drop off followed by a leveling out. All models in this set followed this general behavior. The percent error on testing and training sets for the model with six hidden layers of size five can be seen and Figures 4.14 and 4.15. Again these graphs are very similar across all models so just one has been shown as an example. Table 4.4 summarizes the average absolute percent error of all the models in this set. Comparing these models to the previous set of models we see that adding the VIX to the feature set does appear to increase their predictive power. This is not unexpected as the strong correlation between the price of VIX futures and the VIX would make the VIX a useful starting point for making predictions.

4.2.5 Using a Subset of the Prices of Options and Delta and Gamma to Predict VIX Futures

The fifth set of models was trained to predict VIX futures from a subset of the prices of options, the deltas and gammas associated with those prices. This subset was chosen by taking the

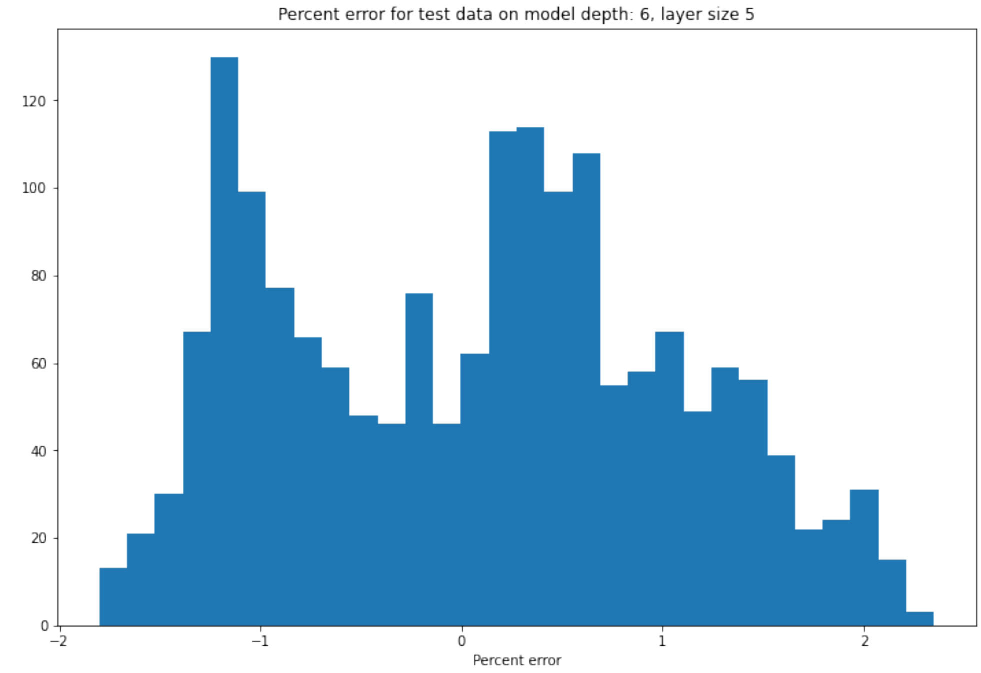


Figure 4.14: *The distribution of the percent errors on the test data of the model with six hidden layers of size five trained to predict VIX futures from the prices of the options and their associated deltas and gammas, and the VIX.*

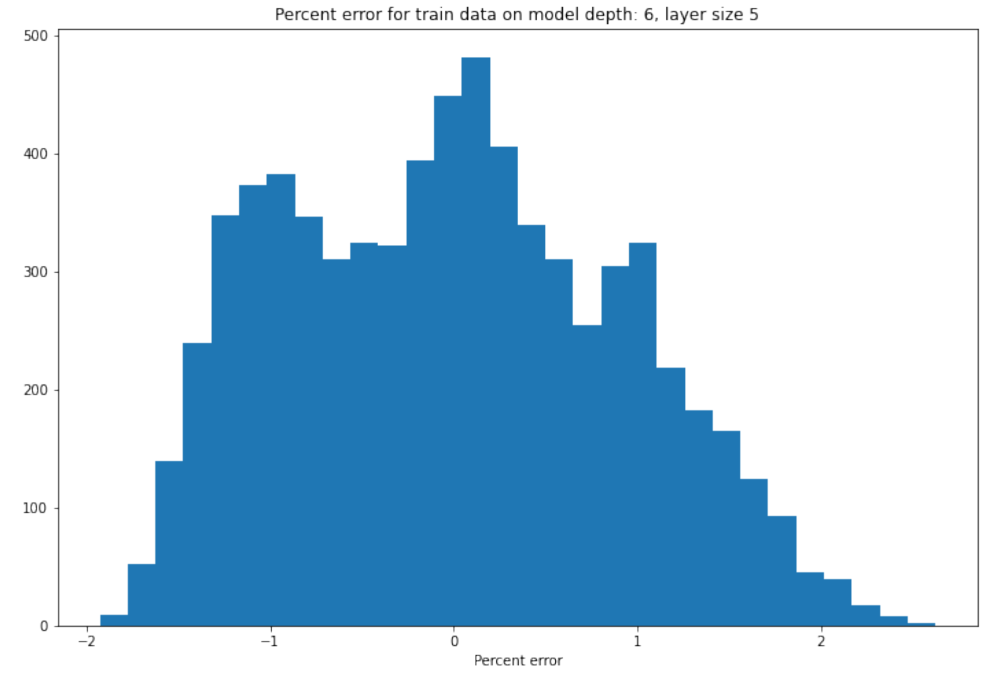


Figure 4.15: *The distribution of the percent errors on the train data of the model with six hidden layers of size five trained to predict VIX futures from the prices of the options, their associated deltas and gammas, and the VIX.*

		Depth		
		1	6	20
Layer Size	5	1.3%	0.8%	1.5%
	136	0.8%	.4%	1.0%

Table 4.4: *Average absolute percent error on test predictions of models trained on the features set consisting of the options prices, the deltas and gammas, and the VIX to predict VIX futures.*

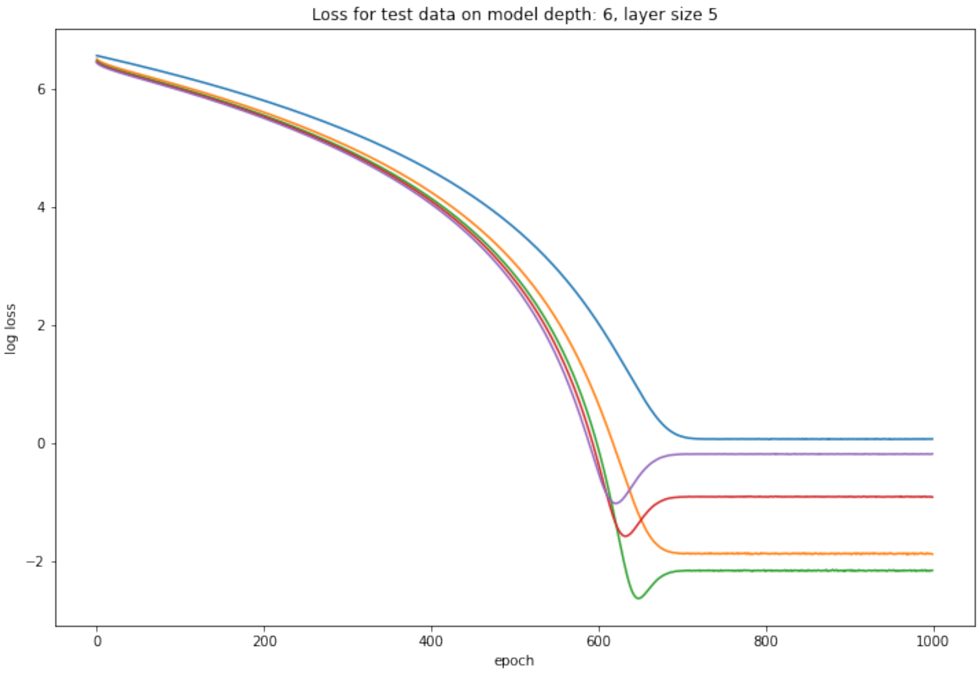


Figure 4.16: *The natural log of the test loss of the model with six hidden layers of size five trained to predict VIX futures from a subset of the prices of the options and their associated deltas and gammas. The different colored lines represent the different folds for 5 fold cross validation.*

five features with the highest average coefficients over all of our LASSO models. These models required a much longer training time in order to get to a point where the loss had leveled out. Each of these models was trained for 1000 epochs. The graph of test loss and the over the training period for the model with six layers of size five can be seen in Figure 4.16. Here we see a variation from what previous models training has looked like. The drop off in error is much more gradual to begin with then becomes more extreme until the loss hits its low point. The loss then in some cases raises slightly and levels off or simply levels off. This behavior was consistent across all the models trained in this set. The percent error on testing and training sets for the model with six hidden layers of size five can be seen and Figures 4.17 and 4.18. The error graphs for this set are all not just similar but nearly identical. Compare for example Figures 4.19 and 4.17. These Figures although they are showing the errors of two models with extremely different structures but these graphs are nearly identical. Table 4.5 summarizes the average absolute percent error

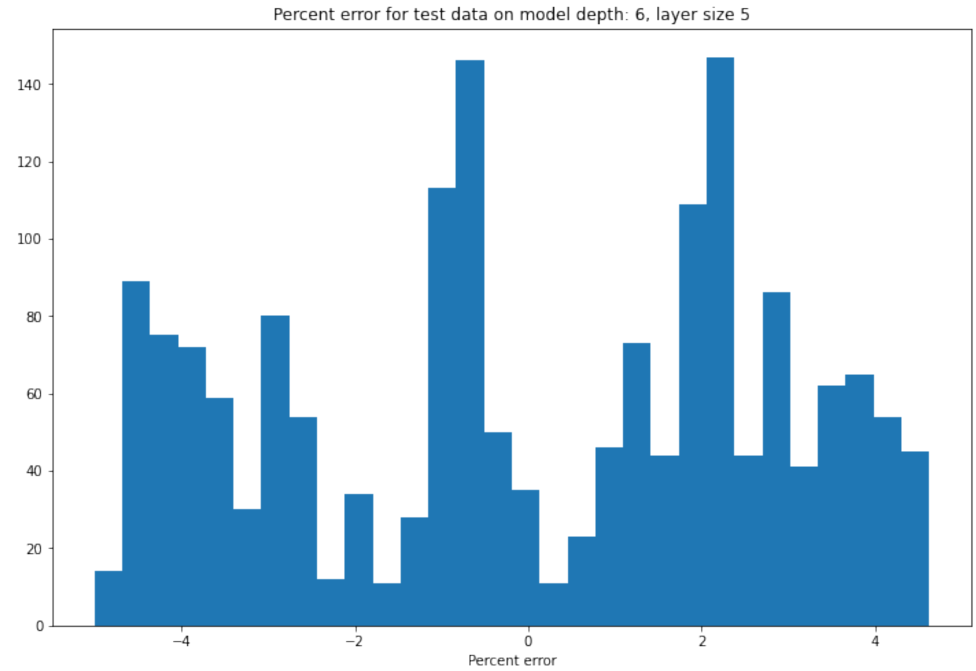


Figure 4.17: *The distribution of the percent errors on the test data of the model with six hidden layers of size five trained to predict VIX futures from a subset the prices of the options and their associated deltas and gammas.*

		Depth		
		1	6	20
Layer Size	5	2.4%	2.4%	2.4%
	136	2.4%	2.4%	2.4%

Table 4.5: *Average absolute percent error on test predictions of models trained on the features set consisting of a subset of the options prices and the deltas and gammas to predict VIX futures.*

of all the models in this set. The most striking thing about these results is how consistent they are. All of the models ended up with a test error of around 2.4%. When looking at the full precise results these errors only differ by a few hundredths of a percent. This likely means that over the long training time, all of these networks are learning very similar functions. This indicates that the more complex functions that the larger networks are capable of learning are no better than the simpler functions learned by the smaller network in this situation. When comparing these results to the set of models trained on the entire feature set rather than just the subset, we see that while drastically reducing the size of our feature set has made our predictions worse, it has only increased their errors by around 1 percentage point. This would indicate that the majority of the information needed to approximate the price of VIX futures is captured in a small number of features.

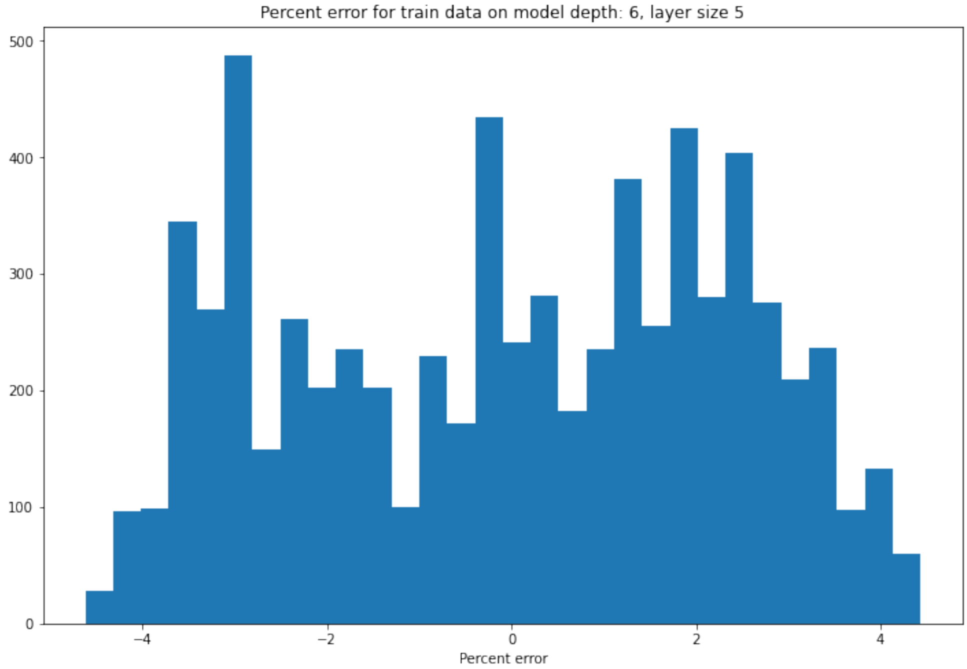


Figure 4.18: *The distribution of the percent errors on the train data of the model with six hidden layers of size five trained to predict VIX futures from a subset of the prices of the options and their associated deltas and gammas.*

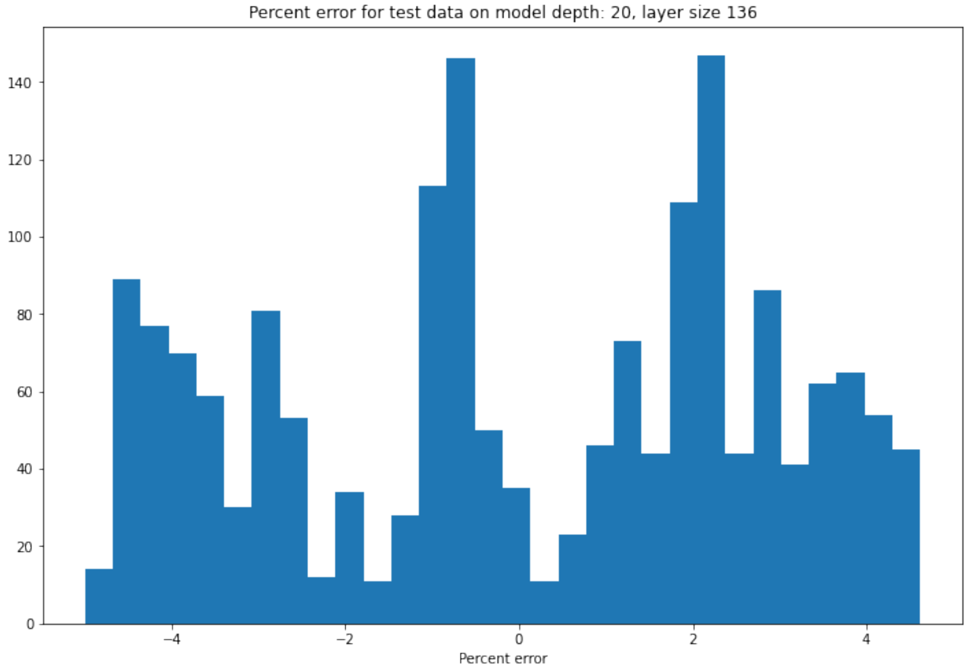


Figure 4.19: *The distribution of the percent errors on the test data of the model with twenty hidden layers of size 136 trained to predict VIX futures from a subset the prices of the options and their associated deltas and gammas.*

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

The results show that approximating both the VIX and VIX futures is possible with the prices of a set number SPX options and furthermore with a small subset of these options. It was found that one can reduce the feature set for prediction down to as few as five features with only a minor decrease in accuracy. Future work can be done to investigate how accurate this prediction remains during chaotic periods when the VIX and VIX futures generally have more erratic behaviors. Such work could also investigate how stable the selected subset of features is over time, i.e. if there is a single subset that works best for making predictions at all times or if different market conditions lead to better predictions with different predictors. Future work could also investigate if these features are useful for predicting the future values of the VIX or future prices of VIX futures.

BIBLIOGRAPHY

[1] S. P. Chatzis, V. Siakoulis, A. Petropoulos, E. Stavroulakis, and N. Vlachogiannakis, “Forecasting stock market crisis events using deep and statistical machine learning techniques,” *Expert systems with applications*, vol. 112, pp. 353–371, 2018.

[2] J. De Spiegeleer, D. B. Madan, S. Reyners, and W. Schoutens, “Machine learning for quantitative finance: fast derivative pricing, hedging and fitting,” *Quantitative Finance*, vol. 18, no. 10, pp. 1635–1643, 2018.

[3] G. James, D. Witten, T. Hatie, and R. Tibshirani, *An Introduction to Statistical Learning*. New York: Springer, 2017.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[5] T. Hastie, R. Tibshirani, and J. Friedmen, *"The Elements of Statistical Learning"*. New York: "Springer", 2017.

[6] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: http://www.jstor.org/stable/2346178

[7] B. Beavis and I. M. Dobbs, *Optimization and Stability Theory for Econmic Analysis*. New York: Cambridge University Press, 1990.

[8] T. Panagiotidis, T. Stengos, and O. Vravosinos, “On the determinants of bitcoin returns: A lasso approach,” *Finance Research Letters*, vol. 27, pp. 235–240, 2018.

[9] G. Sermpinis, S. Tsoukas, and P. Zhang, “Modelling market implied ratings using lasso variable selection techniques,” *Journal of Empirical Finance*, vol. 48, pp. 19–35, 2018.

[10] J. B. Heaton, N. G. Polson, and J. H. Witte, “Deep learning for finance: deep portfolios,” *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3–12, 2017.

[11] R. Culkin, “Machine learning in finance : The case of deep learning for option pricing,” 2017.

[12] W. Bao, J. Yue, and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory,” *PloS one*, vol. 12, no. 7, p. e0180944, 2017.

[13] B. C. Bangal, “Automatic generation control of interconnected power systems using artificial neural network techniques,” Master’s thesis, Bharath University, 2009.

[14] *"The Book of Jargon: Hedge Funds"*. "Latham & Watkins LLP", 2015, https://www.lw.com/admin/Upload/Documents/BoJ_Hedge_Funds-locked-March-2015.pdf.

[15] G. Kürthy, J. Varga, T. Pesuth, Ágnes Vidovics-Dancs, I. Gelányi, G. Sebestyén, E. Boros, G. Sztanó, and E. Varga, *"Basics of Finance"*. Budapest: "Corvinus University of Budapest Department of Finance", 2018, http://unipub.lib.uni-corvinus.hu/3842/1/pfi-briefings.pdf.

[16] R. G. Clarke, H. de Silve, and S. Thorley, *"Fundamentals of Futures and Options"*. "Research Foundation of CFA Institute", 2013, https://cfainstitute.org/-/media/documents/book/rf-publication/2013/rf-v2013-n3-1-pdf.ashx.

[17] S. Global, “S&P U.S. Indices Methodology,” https://www.spglobal.com/spdji/en/documents/methodologies/methodology-sp-us-indices.pdf, 2020, accessed: 2020-11-02.

[18] C. B. O. Exchange, “VIX White paper,” https://www.cboe.com/micro/vix/vixwhite.pdf, 2009, accessed: 2020-10-20.

[19] F. Black and M. Scholes, “The pricing of options and corporate liabilities,” *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973. [Online]. Available: https://doi.org/10.1086/260062

[20] J. Hosker, S. Djurdjevic, H. Nguyen, and R. Slater, “Improving vix futures forecasts using machine learning methods,” *SMU Data Science Review*, vol. 1, no. 4, 2018. [Online]. Available: https://scholar.smu.edu/datasciencereview/vol1/iss4/6

[21] M. Yu, “Predicting the volatility index returns using machine learning,” Master’s thesis, University of Toronto, 2017.

[22] L. V. Ballestra, A. Guizzardi, and F. Palladini, “Forecasting and trading on the vix futures market: A neural network approach based on open to close returns and coincident indicators,” *International Journal of Forecasting*, vol. 35, no. 4, pp. 1250 – 1262, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207019301372