A Neural Network Approach to Understanding Implied Volatility Movements

Jay Cao, Jacky Chen, and John Hull*

Joseph L. Rotman School of Management
University of Toronto

October 2019

Abstract

We employ neural networks to understand volatility surface movements. We first use daily data on options on the S&P 500 index to derive a relationship between the expected change in implied volatility and three variables: the return on the index, the moneyness of the option, and the remaining life of the option. This model provides an improvement of 10.72% compared with a simpler analytic model. We then enhance the model with an additional feature: the level of the VIX index prior to the change being observed. This produces a further improvement of 62.12% and shows that the expected response of the volatility surface to movements in the index is quite different in high and low volatility environments.

JEL Classification: G13, G21

Key words: options, implied volatility movements, neural networks, deep learning

*We would like to thank Peter Christoffersen who, before his untimely death in mid-2018, provided the inspiration for this paper. We also thank the Rotman Financial Innovation Hub (FinHub) and the Global Risk Institute in Financial Services for support.

A Neural Network Approach to Understanding Implied Volatility Movements

1. Introduction

It is well established that there is a negative relationship between an equity's volatility and its price. Black (1976), Christie (1982), Cheung and Ng (1992) and Duffee (1995) demonstrate this using linear regressions of return on subsequent changes in volatility for individual stocks and stock portfolios. Other authors have documented that the negative relationship extends to implied volatilities as well as physical volatilities. Cont and da Fonseca (2002), for example, who carried out a principal components analysis of volatility surface movements, find that shifts in the level of implied volatilities are negatively correlated with the return on the underlying asset. Poulsen et al (2009) find that for both U.S. and European markets the correlation between returns and at-the-money implied volatilities is highly negative, about −0.85.

The reason for the negative relationship has been the subject of much research. Black (1976) suggested a leverage argument. As the equity price moves up (down), leverage decreases (increases) and as a result volatility decreases (increases). In the alternative volatility feedback effect hypothesis, the causality is the other way round. When there is an increase (decrease) in volatility, the required rate of return increases (decreases) causing the stock price to decline (increase). The two competing explanations have been explored by a number of authors including French *et al* (1987), Campbell and Hentschel (1992), Bekaert and Wu (2000), Bollerslev *et al* (2006), Hens and Steude (2009), and Hasanhodzic and Lo (2013). On balance, the empirical evidence appears to favor the volatility feedback effect. For example, the negative relationship seems to hold even when the equity is issued by a company that has very little debt in its capital structure.

Changes in equity prices do not lead to all implied volatilities changing by the same amount. In this paper we use machine learning to produce a model of the dependence of the volatility surface on return for an equity index. We first use a three-feature neural network model to

explore the relationship between the change in the implied volatility of an option on the S&P 500 index and:

a) the daily return of the index;
b) the moneyness of the option; and
c) the option's time to maturity

We will refer to this as the 'three-feature model.' We then add a market sentiment indicator, the VIX index, to create a more elaborate model, which we will refer to as the 'four-feature model.' Our results are based on about two million daily observations on call options on the S&P 500 between 2010 and 2017 from OptionMetrics.

Our measure of moneyness is the delta calculated from the practitioner Black-Scholes model. The practitioner Black-Scholes model is a model where the volatility parameter in the Black-Scholes formula is replaced by the implied volatility. The practitioner delta for a European option on an index is therefore

$$\delta_{\mathrm{BS}} = e^{-qT} N\left(\frac{\ln(S/K) + (r - q + \sigma_{\mathrm{imp}}^2/2)T}{\sigma_{\mathrm{imp}}\sqrt{T}}\right)$$

where N is the cumulative standard normal distribution function, S is the index level, K is the strike price, T is the time to maturity, r is the risk-free rate, q is the dividend yield and σ_{imp} is the implied volatility of the option.[1]

The practitioner Black-Scholes delta is a measure of moneyness widely used by practitioners. Indeed, practitioners often define at-the-money call options as options where $\delta_{\mathrm{BS}} = 0.5$ and at-the-money put options as options where $\delta_{\mathrm{BS}} = -0.5$. For call options, δ_{BS} is close to zero for deep out-of-the-money options and close to 1.0 for deep-in-the-money options. For put options, δ_{BS} is close to zero for deep out-of-the-money options and close to −1.0 for deep-in-the-money options.

[1] The practitioner gamma and vega are defined similarly by setting the volatility parameter equal to the implied volatility of the option under consideration.

Developing an empirical model for the relationship between volatility surface movements and equity returns is important for a number of reasons. It can be used to test the extent to which a particular stochastic volatility model is consistent with market data. This can be done by determining numerically the relationship between volatility surface movements and the features listed above for the stochastic volatility model under consideration and then comparing it with the empirically determined relationship. An empirical model has the potential to provide useful information for a trader who has to quote implied volatilities in a market where equity prices are moving fast. It can also be used to estimate a minimum variance delta for hedging. This is a hedge ratio that takes account of expected volatility changes as well as the change in the underlying asset price. The minimum variance delta is

$$\delta_{\mathrm{BS}} + v_{\mathrm{BS}} \frac{\partial \sigma_{\mathrm{imp}}}{\partial S}$$

where v_{BS} is the practitioner vega and $\partial \sigma_{\mathrm{imp}} / \partial S$ is estimated from empirical results.

Other research which uses machine learning for modeling volatility changes is Nian et al (2018). This focuses on minimum variance delta estimates and shows that machine learning can lead to hedging improvements. Our research is more general. We are concerned with understanding movements in the whole volatility surface.

Our objective is to use machine learning tools to estimate the function F in the relationship

$$E(\Delta\sigma_{\mathrm{imp}}) = F\left(\frac{\Delta S}{S}, \delta_{\mathrm{BS}}, T, V\right)$$

where E denotes expected value, σ_{imp} is an option's implied volatility, S is the S&P 500 index, δ_{BS} is the option's moneyness measure just mentioned, T is the option's time to maturity, and V is the level of the VIX index (observed immediately prior to the changes in the implied volatility and the index). Our research provides an application of multi-layer neural networks in finance.[2] As explained below, a multi-layer neural network is a useful tool for estimating complex non-linear functions when a large amount of data is available.

[2] Tests of the use of artificial neural networks for option pricing are provided by Hutchison et al (1994) and Cucklin and Das (2017).

Hull and White (2017) in considering minimum variance delta estimates propose the following analytic model:

$$E(\Delta\sigma_{imp}) = \frac{\Delta S}{S}\frac{a + b\delta_{BS} + c\delta_{BS}^2}{\sqrt{T}}$$

where a, b, and c are parameters. In this model the expected change in the implied volatility is linearly dependent on the return on the index, inversely proportional to the square root of the time to maturity and quadratic in the practitioner Black-Scholes delta. The parameters a, b, and c are estimated from data. This model was found to produce results that compared favorably with more elaborate stochastic volatility models, and we use it as a benchmark. We find that a three-feature neural network model produces a 10.72% improvement over this model. Adding the VIX index as a fourth feature produces a further improvement of 62.12%.

The organization of the paper is as follows. Section 2 describes the nature of neural networks. Section 3 explains the data and how it was used. Section 4 explains the way algorithms were implemented. Section 5 presents the results for the three-feature model. Section 6 examines the extra explanatory power of the VIX index and conclusions are in Section 7.

2. Neural Networks

Artificial neural networks (ANNs) are at the very core of deep learning. They were first introduced by McCulloch and Pitts (1943) who presented a simplified model of how the neurons in a human brain can perform computations. In recent years, improvements in computer processing speed and the large volumes of data that are being generated in many spheres have led to renewed interest in ANNs.

Traditionally, finance and economics have used linear models or models involving simple transformations of linear functions. ANNs enable non-linear functions involving many parameters to be estimated from large data sets. The structure of an ANN is shown in Figure 1. There are a number of inputs, referred to as features and one or more outputs, referred to as targets. In our first application, there are three features: index return, moneyness (as measured by the Black-Scholes delta), and time to maturity. There is one target, the change in the

implied volatility. We then add an additional market sentiment indicator, the VIX index, as a feature.

The inputs form the input layer and the outputs form the output layer. The calculations necessary to determine the output layer from the input layer involve one or more hidden layers. Each hidden layer has a number of nodes at which values are calculated. In Figure 1 there are n hidden layers, an input layer, and an output layer. The left-most layer is the input layer and contains the values of the input variables (or features). The right-most layer is the output layer and contains the output variables (or targets).

Instead of developing a model to estimate the outputs directly from the inputs we specify functions relating the values at the nodes comprising one layer to values at the nodes comprising the previous layer. The first function is used to transform the values of the features in the input layer to the values at the nodes in the first layer. Further functions are used to transform the values at the nodes in layer i to values at the nodes in layer i+1 ($1 \leq i \leq n-1$). A final function is used to transform the values at the nodes in layer n to the target values in the output layer. These functions are referred to as activation functions. ANNs that have multiple hidden layers are referred to as deep neural networks. See Hull (2019) for more details.

The Universal Approximation Theorem, derived by Hornik (1991), states that an ANN with a single hidden layer can approximate any function arbitrarily closely. However, a very large number of nodes may be required and in some situations it may be more practical to use multiple layers so that there are fewer nodes overall.[3]

Suppose that there are m_0 features, m_i nodes in hidden layer i ($1 \leq i \leq n$), and m_{n+1} targets. We will refer to the input layer as layer 0 and the output layer as layer n+1. Define $v_{i,j}$ as the value at the jth node of layer i ($0 \leq i \leq n+1$, $1 \leq j \leq m_i$). The variable $v_{0,j} = x_j$ is the value of the jth feature and $v_{n+1,j}$ is the estimate of the jth target given by the model.

The formula for calculating the $v_{i,j}$ ($1 \leq i \leq n+1$ and $1 \leq j \leq m_i$) can be written

[3] See Telgarsky (2016) for a discussion of this.

$$v_{i,j} = f_i\left(b_{i,j} + \sum_{k=1}^{m_{i-1}} w_{i-1,k,j} v_{i-1,k}\right) \tag{1}$$

In this formula f_i defines the activation function used to calculate values at the nodes of layer i ($1 \leq i \leq n + 1$). The $b_{i,j}$ and $w_{i,k,j}$ are parameters of the model. Specifically, $w_{i,k,j}$ is the weight assigned to the value at the kth node of layer i when the value at jth node of layer i+1 is being calculated and $b_{i,j}$ is a constant, known as the bias, which is added to the weighted value computed for the jth node of layer i.

The number of parameters in equation (1) can be quite large. For example if there are F features, H hidden layers, M nodes in each hidden layer, and T targets there are

$$(F+1)M + M(M+1)(H-1) + (M+1)T$$

parameters in total. We used three hidden layers and 80 nodes per hidden layer. The number of parameters in the three and four feature models were therefore 13,361 and 13,441, respectively. The huge number of parameters compared with traditional models naturally leads to overfitting concerns. As we discuss later these concerns are addressed by dividing the data into a training set, validation set, and test set and choosing an appropriate stopping rule for the algorithm.

The weights and biases are chosen to minimize an objective function that captures the difference between the estimated target values for the training set and the actual values. Our application involves only one target (the change in implied volatility) and our objective function is the mean squared error between the estimated target and the actual target across all the options used for training.

The minimization is accomplished using a steepest descent algorithm. Initial values are assigned to the weights and biases. An iterative procedure is then carried out to improve the objective function by changing these parameters. On each iteration a partial derivative of the objective function is calculated with respect to each of the parameters. Each parameter is then reduced by the product of its partial derivative and a constant, referred to as the learning rate. The iterations are referred to as epochs.

For large data sets and models involving many parameters this procedure is made computationally feasible by a technique known as backpropagation. This was proposed by Rummelhart et al (1986) and involves working back through the layers calculating the required partial derivatives using the chain rule.

The vanilla gradient descent algorithm described above can sometimes be slow. To speed up the learning process, several variations have been developed. For example,

- *Mini-batch stochastic gradient descent*. This algorithm randomly splits the training data into small mini-batches. Instead of using the whole training data to calculate gradient, it updates model parameters based on the gradient calculated from a single batch with each of the mini-batches being used in turn. Because the algorithm estimates the gradient using a small sample of the training data, it is less computationally expensive and often leads to much faster learning.
- *Gradient descent with momentum*. This algorithm calculates gradient as an exponentially decaying moving average of past gradients. This approach helps to build up parameter update 'velocity' in any direction that has a consistent gradient.
- *Gradient descent with adaptive learning rates*. A learning rate that is too small will result in many epochs being required to reach a reasonable result. A learning rate that is too high may lead to oscillations and a poor result. Different model parameters may benefit from different learning rates at different stages of training. Because choosing proper learning rates can be difficult, many algorithms try to automate the process. For example, RMSProp (Root Mean Squared Propagation) and Adam (Adaptive Moment Estimation) are both popular adaptive learning rate algorithms that adjust learning rate at each iteration for each model parameter.

In this paper, we use a mini-batch size of 512 and implement Adam methods with the parameters suggested in Kingma and Ba (2017).[4]

In practice, the algorithms we have described are not used to fully minimize the loss function. This would be computationally quite time consuming. Also the nature of the algorithms and the large number of parameters used are such that as training increases more of the

[4] Initial value of weights can also affect convergence speed. In our training, we apply the Glorot uniform initializer suggested by Glorot and Bengio (2010).

idiosyncrasies of the training data tend to be reflected in the model. A stopping rule is therefore specified both for computational efficiency and to avoid overfitting. We describe the stopping rule we used in Section 4.

3. Data

We used S&P 500 call options data from OptionMetrics between January 2010 and December 2017. The data for each option on each day includes the strike price, time to maturity, index level, and implied volatility, as well as hedge parameters such as delta, gamma, vega, and theta derived from the practitioner Black-Scholes model.

The data was filtered in a number of ways. We only retained options where the information provided was complete. Options with remaining lives less than 14 days were removed from the data set. Options for which the practitioner Black-Scholes delta was less than 0.05 or greater than 0.95 were removed from the data set. The data was then sorted to produce observations for the same option on two successive trading days. This resulted in about 2.07 million observations on daily volatility changes for 53,653 call options.

The three features we used in the first stage of this research are the S&P 500 daily change, time-to-maturity, and the practitioner Black-Scholes delta. There is one target, the implied volatility change. In the second stage we added the VIX index as a feature. A summary of statistical properties of the features and target variables is provided in Table 1.

To apply the neural network technique, we randomly divided the data into a training set, a validation set, and test set, with a 7:2:1 ratio. We used the training set and the validation set to train and fine-tune the neural network model, and then evaluated the model performance with the test set. All results presented are those for the test set.

4. Model Selection Criteria

Key elements of a neural network model are the activation function, the number of layers and the number of nodes per layer. The activation functions f_i in equation (1) for $i \leq n$ are designed

to distinguish between positive and negative signals. We considered four different activation functions that have been suggested in the literature: the sigmoid, the rectified linear unit (relu), the leaky relu, and the exponential linear unit. The functional forms are shown in Table 2. They all have attractive properties for backpropagation algorithms. For $i = n+1$ the activation function is $f(x) = x$ so that a linear function relates values at the nodes on the final hidden layer to the target. (This is usual practice when a continuous variable is being estimated.) We present results for a model with three hidden layers and 80 nodes per layer. We found models with sigmoid activation functions generally perform better (lower mean squared errors) and we will therefore only present results from using the sigmoid activation function.

To avoid overfitting, we experimented with a number of different early stopping rules. A common approach involves stopping when the mean square error for the validation set starts to trend up. For our data this happened only after a very large number of epochs if at all, a result which may be indicative of local overfitting.[5] In the end, we decided to use the smoothness of the predicted change in the volatility surface as our criterion. We manually inspected a three-dimensional plot of the volatility surface change as the number of epochs was increased and stopped when this was no longer smooth. This led to earlier stopping than that would be indicated by other rules. In both of the three-factor models and four-factor model we stopped after 4,000 epochs. The choice of the stopping rule did not affect the general shape of the volatility surface movements, but it did affect the smoothness of the results.

5. Results for Three-Feature Model

Hull and White (2017) propose a simple analytic model for determining volatility surface movements. Their model is

$$E(\Delta\sigma_{\text{imp}}) = \frac{\Delta S}{S}\frac{a + b\delta_{\text{BS}} + c\delta_{\text{BS}}^2}{\sqrt{T}} \tag{2}$$

[5] See for example Lawrence and Giles (2000).

This model involves three parameters, a, b, and c, which can be estimated using linear regression. The best fit parameters for our data are shown in Table 3. We use the model as a benchmark. Similar to Hull and White (2017), we define the Gain from using Model A rather than Model B as

$$\text{Gain} = 1 - \frac{SSE\left[\text{Model A}\right]}{SSE\left[\text{Model B}\right]} \tag{3}$$

where SSE denotes sum of squared errors.

The gain from using the three-feature machine learning model rather than the analytic model was 10.72%. The mean squared error for the test set was 0.0000984 (with implied volatilities measured as decimals). To investigate the sources of the gain we calculated the gain given by the three-feature model for a number of different subsets of the data. Our results are summarized in Table 4. This shows that the gain is greatest for (a) situations when the index return is higher than +1% or lower than −1% and (b) short maturity options.

Tables 5 and 6 show the volatility changes predicted by the analytic model in equation (2) and the three-feature model for index returns of −1.25% and +1.25%. Plots of the volatility surface changes are in Figure 2. As might be expected, the results from the two models are similar. The volatility surface moves up when the return is negative and moves down when the return is positive. The change decreases as the time to maturity increases and is greatest for low-delta and high-delta options.

The analytic model in equation (1) is linear in the return. The impact of a gain of X% on a particular option's implied volatility is equal and opposite to that of a gain of $-X$%. The same is not true for the three-feature neural network model. The reduction in implied volatilities arising from a daily return of 1.25% is on average about twice as great the increase in implied volatilities arising from a daily return of −1.25%. The change in the implied volatility predicted by the analytic model is too high for large negative returns and too low for large positive returns. This non-linearity suggests that the gamma, as well as the delta of a portfolio, may be affected by volatility uncertainty.

6. Results for Four-Feature Model

The four-feature model is designed to test whether the behavior of the volatility surface in high volatility environments is different from that in low volatility environments. The fourth feature is the value of the VIX index on the day before the index return and volatility change are observed. With a mean squared error of 0.0000372 for the test set, the four-feature model produces a gain of 62.12% over the three-feature model. Table 7 shows the gain as a function of the VIX index and the index return. It shows that the gain is greatest for high and low values of the index return and high and low values of the VIX.

Table 8 shows the expected changes in the volatility given by the four-feature model when the index return is +1.25% and −1.25%, and the VIX has values of 13% and 16%. Figure 3 shows corresponding charts. It is interesting to note that, when the VIX is low (13%) and there is a big increase in the index (+1.25%), all points on the volatility surface increase. This is quite different behavior from the average shown in Table 6. As with most of our other results, this one is most marked for high delta short maturity options. Table 6 shows that the expected change in the implied volatility of a three-month option with a delta of 0.9 is −62 basis points when the index return is +1.25%. Conditional on a low VIX index of 13% this change is 84 basis points, over 146 basis points greater. Presumably a high index return in a low volatility environment is seen as signal of high future volatilities.

Our results show that the VIX index and the return on the index interact in a way that makes the basic Hull-White three-parameter model at best an incomplete description of volatility surface movements. We illustrate this in Table 9 which shows the Hull-White parameters for different ranges of the index return and the VIX index. The parameters a can be viewed as an indicator of the size of volatility surface movements for low delta options. A negative value of a indicates that positive returns lead to negative volatility surface movements and vice versa. It can be seen that for returns less than -1%, a is approximately the same regardless of the VIX index. For returns greater than -1%, the values of a indicate that the magnitude of the low-delta volatility movements increases as VIX increases. For values of the VIX less than 19, the magnitude of low-delta volatility surface movements decreases as the index return increases. When VIX is low and the return is highly positive, a is positive indicating the volatility surface moves in the opposite direction to that normally expected for a positive return. This is consistent with our ANN result mentioned above.

The parameter *b* can be interpreted as the slope of the implied volatility as a function of delta for low delta options. It can be seen that this slope is always positive and tends to increase as the VIX index increases and the index return increases. The parameter *c* measures the extent the curvature of the relationship between implied volatility and delta (see Figures 2 and 3). As in the case of the low-delta slope, this is greatest in situations where the VIX index is high and the index return is high.

As mentioned earlier, one application of this research is to minimum variance delta hedging. The Hull-White model sets the minimum variance delta as

$$\delta_{BS} + \frac{v_{BS}}{S\sqrt{T}}(a + b\delta_{BS} + c\delta_{BS}^2)$$

where δ_{BS} and v_{BS} are the delta and vega parameters calculated from the practitioner Black-Scholes model, S is the index level, and T is the time to maturity. It is clear from our research that the current level of the VIX, which is not included in the Hull-White model, has a bearing on volatility movements and therefore on the minimum variance delta. A potential simple improvement on the Hull-White model is to use the final row in Table 9 to adjust the *a*, *b*, and *c* parameters according to the level of the VIX index. For example, for an option with a delta of 0.5, the table indicates that the delta adjustment when the VIX is greater than 19 is 48% higher than when the VIX is in the 13 to 19 range, and this is 30% greater than when the VIX is less than 13.

7. Conclusions

Machine learning is usually used as a prediction tool. The values of features observed prior to time *t* are used to predict target values at or after time *t*. In this paper we show how machine learning can be used to explore a nonlinear relationship between variables. We consider the relationship between the change in the S&P 500 index and the contemporaneous change in the implied volatility of an option on the index as a function of option's maturity and its moneyness. Our results are generally supportive of the negative correlation between the implied volatilities and asset returns that has been documented in the literature and is

discussed in the introduction. However, we do find one notable exception. When volatilities are low and the index return is particularly high there is a tendency for volatilities to increase.

The use of a three feature neural network model refines the expected volatility change estimates produced by Hull and White (2017). The difference between the two models is most marked for high-delta short-maturity options and when extreme positive or negative returns are observed. The Hull-White model tends to understate the impact of large positive returns and overstate the impact of large negative returns.

Volatility surface movements depend on the initial level of volatility. We demonstrated this by including the level of the VIX index on Day $t-1$ as a feature to determine expected volatility surface changes between Day $t-1$ and Day t. It is normally the case that the whole volatility surface moves up when the index declines and moves down when the index increases. As mentioned, we have shown that this is not necessarily what happens when there is a large positive return in the index. When the large positive return occurs in an environment where volatilities were (prior to the return being observed) low, the movement in the volatility surface is the opposite of that normally associated with a positive return. For example, when the VIX is 13% and a +1.25% index return occurs, the expected changes in implied volatilities are mostly positive. When the VIX is a more normal 16% the expected changes are highly negative. The changes are most marked for high-delta short-maturity options.

This research assumed a stationary model. When we tested the model on the most recent 10% of the data it performed slightly less well than on a test set which was randomly chosen from our complete data set. A possible area for further research is to extend our model to one where the time sequence of the data is taken into account. This could be done using a recurrent neural network where equation (1) is modified so that $v_{i,j}$ depends on the previous day's estimates as well as on the current day's values at the immediately preceding nodes. The long short-term memory approach of Hochreiter and Schmidhuber (1997) could also be used.

Another possible extension of our research would be to train a model on the errors in an analytic model such as Hull and White (2017) rather than on the movements in the volatility surface itself. This is analogous to a widely used machine learning method known as gradient

boosting where there are a sequence of predictors each one trying to correct the errors of the previous one.

References

Bekaert, G. and Wu G., Asymmetric volatility and risk in equity markets. *Rev Financ Stud*, 2000, **13**, 1–42.

Black, F., "Studies of stock price volatility changes", 1976*, Proceedings of the Business and Economics Section of the American Statistical Association*, 177–181.

Bollerslev, T., Litvinova J., and Tauchen G., "Leverage and volatility feedback effects in high-frequency data," *J Financ Econom*, 2006, **4**, 353-384.

Campbell, J. Y., and Hentschel L., No news is good news: an asymmetric model of changing volatility in stock returns, *J Financ Econ*, 1992, **31**, 281–331.

Cheung, Y.-W. and Ng L., Stock price dynamics and firm size: an empirical investigation", *J Financ*, 1992, **47***,* 1985–1997.

Christie, A., The stochastic behavior of common stock variances: value, leverage, and interest rate effects, *J Financ Econ*, 1982, **10**, 407–432.

Cont, R. and da Fonseca J., "Dynamics of implied volatility surfaces" *Quant Financ*, 2002, **2**, 45-60.

Culkin, R. and Das, S.R, Machine learning in finance: the case of deep learning for option pricing, *J Invest Mgmt*, 2017, **15**, 92-100.

Duffee, G., Stock returns and volatility: a firm level analysis, *J Financ Econ*, 1995*,* **37**, 399–420.

French, K. R., Schwert G. W., and Stambaugh R.F., Expected stock returns and volatility, *J Financ Econ,* 1987*,* **19**, 3–29.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feed forward neural networks, 2010, Available online at http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

(accessed January 8, 2019)

Hasanhodzic, J. and Lo A., "Black's leverage effect is not due to leverage," Working Paper, MIT, 2013.

Hens, T. and Steude, S.C., "The leverage effect without leverage", *Financ Res Lett*, 2009, **6**, 83–94.

Hochreiter S. and Schmidhuber J. Long short-term memory, Neural Computation, 1997, **9**, 1735-1780.

Hornik, K., Approximation capabilities of multilayer feedforward networks, *Neural Networks*, 1991, **4**, 251-257.

Hull, J., *Machine Learning in Business: An Introduction to the World of Data Science*, www-2.rotman.utoronto.ca/~hull/mlbook.

Hull, J and White A., Optimal delta hedging for options, *J Bank Financ,* 2017, **82**, 180-190.

Hutchison, J.M., Lo, A.W., and Poggio T., A nonparametric approach to pricing and hedging derivative securities via learning networks, *J Finance*, 1994, **49**, 851-889.

Kingma, D. P. and Ba. J. , Adam, A Method for Stochastic Optimization, 2017, Available online at https://arxiv.org/pdf/1412.6980.pdf. (accessed January 8, 2019)

Lawrence, S. and. Giles C.L., 2000, Overfitting and neural networks: conjugate gradient and backpropagation, *Proceedings of International Joint Conference on Neural Networks, Como, Italy, IEEE Computer Society*, Los Alamitos, CA, 2000, 114-119.

McCulloch, W. and Pitts W., A logical calculus of ideas in nervous activity, 1943, *B Math Biophys*, **5**, 115-133.

Nian, K., Coleman T.F., and Li Y., Learning minimum variance discrete hedging directly from the market, *Quant Financ*, 2018, **18**, 1115-1128.

Poulsen, R., Schenk-Hoppé K.R., and Ewald,C-O, Risk minimization in stochastic volatility models: model risk and empirical performance, *Quant Finance,* 2009*,* **9**, 693-704.

Rummelhart, G., Hinton G., and Williams R., Learning internal representations by error propagation, *Nature*, 1986, **323**, 533-536.

Telgarsky, M., Benefits of depth in neural networks, *JMLR: Workshop and Conference Proceedings,* 2016, **49**, 1-23.

Table 1: Summary Statistics of Features and Target

	S&P500 Daily Change	Time-to-Maturity	Delta	VIX	Implied Volatility Change
Mean	0.05%	0.81	0.63	15.89	-0.06%
Std	0.87%	0.97	0.29	5.41	1.12%
Min	-6.66%	0.06	0.05	9.14	-45.30%
Median	0.05%	0.34	0.72	14.42	-0.01%
Max	4.74%	4.38	0.95	48.00	36.85%

Table 2: Alternative Activation Functions. The value used for *a* in the leaky relu and exponential linear unit activation functions was 0.03.

Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$
Relu	$f(x) = \max(x, 0)$
Leaky relu	$f(x) = \begin{cases} x & x \geq 0 \\ ax & x < 0 \end{cases}$
Exponential linear unit	$f(x) = \begin{cases} x & x \geq 0 \\ a(e^x - 1) & x < 0 \end{cases}$

Table 3: Parameters estimated for the analytic model in equation (2) using training set and validation data sets: January 2010 to December 2017. Time is measured in years and the implied volatility change is measured in decimal form.

Parameter	Value	t-statistic
a	− 0.2329	−165.3
b	0.4176	66.5
c	− 0.4892	−84.5

Table 4: Percentage gain of three-feature model over analytic model in equation (2) for different index returns and different times to maturity

Time to Maturity	Index Return				
	<-1%	**−1% to 0**	**0 to 1%**	**>1%**	**All**
0-6m	28.39	3.65	1.07	26.13	11.48
6m to 1yr	20.56	-1.42	-2.29	14.13	5.14
1yr to 2yr	13.49	1.62	-1.69	15.94	4.52
>2yr	11.12	2.02	3.23	7.54	4.23
All	26.76	3.28	0.98	25.09	10.72

Table 5: Expected daily changes in volatility given by the analytic model in equation (2) for options with different moneyness and time to maturity. Volatility is measured in basis points per year. Moneyness is measured by the practitioner Black-Scholes delta. The table considers scenarios where the daily return on the index is (a) −1.25%, (b) +1.25%

Index return = -1.25%

B-S Delta	Time to Maturity			
	3m	6m	1yr	1.5yr
0.1	49	35	25	20
0.3	38	27	19	15
0.5	37	26	18	15
0.7	45	32	23	18
0.9	63	45	32	26

Index return = +1.25%

B-S Delta	Time to Maturity			
	3m	6m	1yr	1.5yr
0.1	-49	-35	-25	-20
0.3	-38	-27	-19	-15
0.5	-37	-26	-18	-15
0.7	-45	-32	-23	-18
0.9	-63	-45	-32	-26

Table 6: Expected daily volatility changes given by three-feature model for options with different moneyness and time to maturity. Volatility is measured in basis points per year. Moneyness is measured by the practitioner Black-Scholes delta. The table considers scenarios where the daily return on the index is (a) −1.25%, (b) +1.25%.

Index return = -1.25%

B-S Delta	Time to Maturity			
	3m	6m	1yr	1.5yr
0.1	33	23	16	6
0.3	18	14	10	8
0.5	17	14	8	6
0.7	20	16	9	9
0.9	29	21	9	8

Index return = +1.25%

B-S Delta	Time to Maturity			
	3m	6m	1yr	1.5yr
0.1	-54	-42	-36	-28
0.3	-41	-32	-25	-25
0.5	-39	-32	-24	-23
0.7	-43	-36	-25	-20
0.9	-62	-43	-26	-14

Table 7: Percentage gain of four-feature model over three-feature model for different index returns and different times to maturity

VIX Index	Index Return				
	<-1%	**-1% to 0%**	**0% to 1%**	**>1%**	**All**
<=13%	85.00	52.26	53.69	89.23	55.23
13% to 19%	67.73	51.34	46.64	70.17	53.41
>=19%	80.01	72.09	69.69	86.02	78.78
All	77.01	55.49	53.41	80.80	62.12

Table 8: Expected daily volatility changes given by four-feature model for options with different moneyness and time to maturity. Volatility is measured in basis points per year. Moneyness is measured by the practitioner Black-Scholes delta. The table considers scenarios where the daily return on the index is −1.25% and +1.25%, and the VIX index is 13%, 16%.

Index Return = -1.25%; VIX =13%

B-S Delta	Time to Maturity			
	3m	6m	1yr	1.5yr
0.1	41	25	12	5
0.3	21	14	9	5
0.5	16	8	3	2
0.7	15	6	1	1
0.9	13	-1	-10	-11

Index Return = +1.25%; VIX = 13%

B-S Delta	Time to Maturity			
	3m	6m	1yr	1.5yr
0.1	4	1	0	1
0.3	16	11	6	5
0.5	26	18	12	8
0.7	37	25	16	12
0.9	84	62	45	35

Index Return = -1.25%; VIX =16%

B-S Delta	Time to Maturity			
	3m	6m	1yr	1.5yr
0.1	41	30	20	14
0.3	36	28	20	15
0.5	34	26	20	16
0.7	34	25	21	18
0.9	46	32	28	29

Index Return = +1.25%; VIX = 16%

B-S Delta	Time to Maturity			
	3m	6m	1yr	1.5yr
0.1	-88	-58	-33	-24
0.3	-76	-49	-27	-19
0.5	-72	-47	-27	-19
0.7	-88	-59	-37	-27
0.9	-188	-114	-59	-38

Table 9: Regression parameters under different market scenarios. Time is measured in years and the implied volatility change is measured in decimal form.

Index	VIX≤13			13<VIX<19			VIX≥19		
Return	*a*	*b*	*c*	*a*	*b*	*c*	*a*	*b*	*c*
≤−1%	− 0.269	0.297	− 0.357	− 0.228	0.233	− 0.137	− 0.280	0.476	− 0.508
−1% to +1%	− 0.133	0.297	− 0.282	− 0.208	0.407	− 0.394	− 0.291	0.556	− 0.631
≥+1%	0.008	0.370	− 0.357	− 0.191	0.466	− 0.540	− 0.277	0.737	− 1.033
All	− 0.156	0.275	−0.278	− 0.208	0.358	− 0.343	− 0.274	0.572	− 0.730

Figure 1: The structure of an artificial neural network. The $v_{0,j}$ are the inputs and the $v_{n+1,j}$ are the outputs.

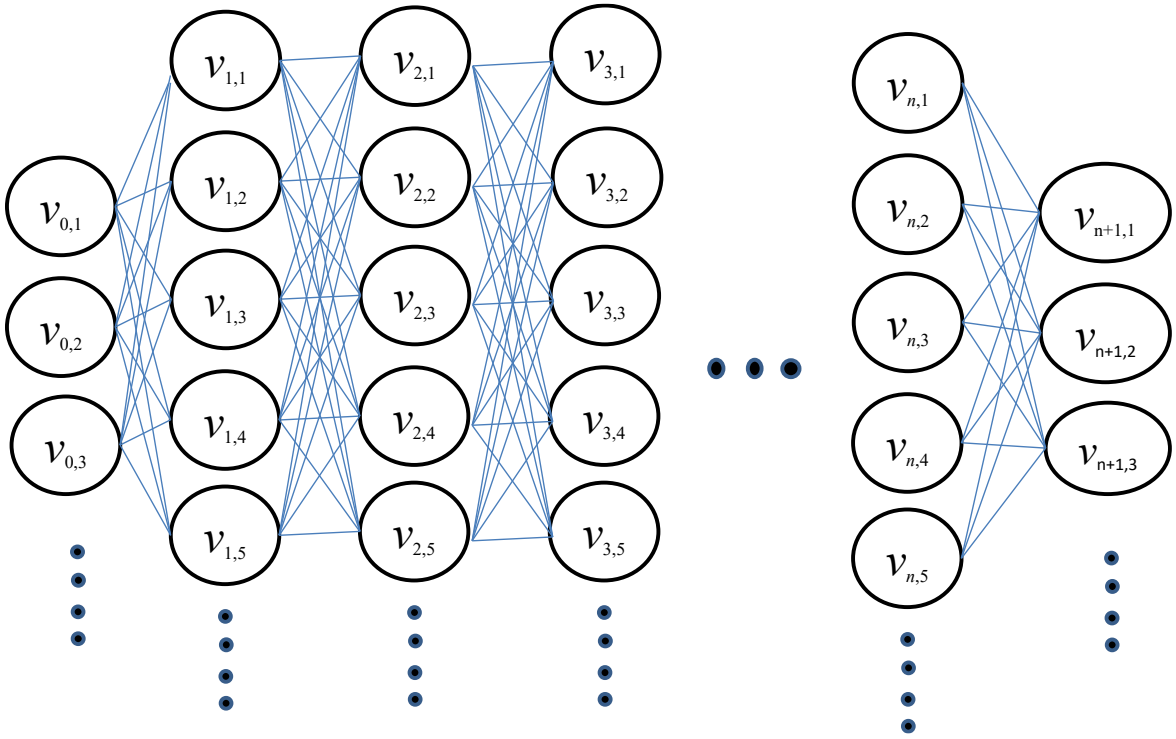

Figure 2: Expected change in implied volatility for analytical and machine learning 3-feature model

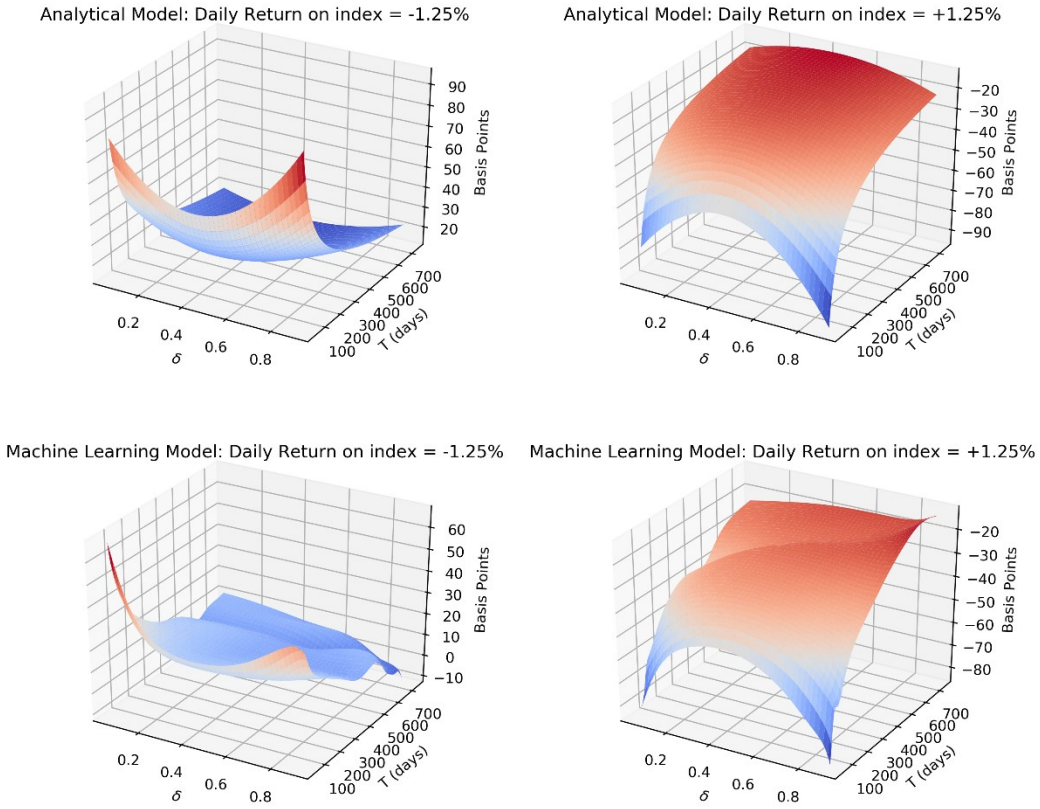

Figure 3: Expected change of implied volatility surface for 4-feature machine learning model