CS230

Historical Backfilling of VIX data
Category: Finance

Matt Johnson
SUNet ID: mattj949
mattj949@stanford.edu

Alex Moraru
SUNet ID: amoraru
amoraru@stanford.edu

Abstract

In this paper we model VIX levels prior to the index's creation in 1986, using time series price data on the S&P 500 index, gold, and treasury yields. Working backwards in time, we train a 2-layer bidirectional LSTM network on data between 2020 and 1991, and after hyperparameter tuning on data between 1991 and 1988, we show that our model does a surprisingly good job at fitting low and medium frequency movements in, as well as the absolute levels of, the VIX index between 1988 and 1986. Using our trained model, we predict VIX levels going back to 1968, and illustrate a use case for this data by performing a backtest of a simple options strategy.

1 Introduction and Motivation

In 1973, Fischer Black, Myron Scholes and Robert Merton came up with an equation to price financial options, and this equation has gone on to revolutionize the world of finance. For any underlying (where an underlying could be a stock like Apple or Microsoft), a trader can input the current price, the strike price for the option, the option duration, and the estimated future standard deviation of the price series of the underlying. With this, the Black-Scholes (BS) model outputs a theoretical price for the option.

The only input to the BS model that is not deterministic, and hence must be estimated, is the future standard deviation ("volatility") of the price series of the underlying. Different market participants will have different estimates for this number, and one way to obtain an estimate is to use the BS model, combined with the current market price of an option, to back-out the market's estimate for future volatility. This number is known as the "implied volatility" of the option.

Implied volatility provides an insight into the level of uncertainty traders have towards the future price of an underlying. The goal of this project is try and model implied volatility, specifically the implied volatility on the S&P 500 index (the "VIX"), since it has been calculated on a daily basis since 1986, and hence gives us one of the longest "ground truths" on which to train our model. We utilize a bi-directional LSTM network, and for inputs we choose financial price data available prior to the VIX's creation in 1986 (Dow Jones index, S&P 500 index, gold, and treasury yields), as this allows us to interpolate VIX levels prior to 1986.

2 Related work

We were unable to find any literature directly concerned with the backfilling of VIX data, or of implied volatility data more broadly. Most literature concerning VIX modeling has to do with forecasting. Examples include using the VIX to predict future volatility (Martens and Zein) and

CS230: Deep Learning, Fall 2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

(Padhi and Shaikh), or using various contemporaneous and historical data series to predict future VIX values (Fernandes et al.). The literature has shown some success in VIX forecasting, both using parametric models like GARCH (Majmudar and Banerjee) and non-parametric models like LSTM networks (Hosker et al.). We overlap with the literature in our usage of LSTM neural networks, but our motivation to backfill VIX levels, as opposed to forecasting future VIX levels, is otherwise orthogonal to the current research.

3 Dataset and Features

3.1 Dataset

The dataset for this project is comprised of time series price data from Global Financial Data (GFD), a data provider. Our ground truth are daily VIX prices going back to 1986, while our input data are daily prices for the Dow Jones index (since 1885), S&P 500 index (GFD extended to 1928), Gold (since 1968), and Global Financial Data's US treasury bill index (since 1791). Our model requires price data on a daily frequency, and as a result we are limited by our data on gold, as a daily frequency only starts in 1969. Since we need ground truth VIX data to train our model and measure it's performance, we are restricting our training, development, and test sets to the time period in which we have VIX data (1986-today). The data was not shuffled prior to splitting to possibly give us a more accurate measurement of how the model performs when backfilling the VIX values (for justification see the appendix). The dataset is split as follows:

Training:	01/01/1991 - 12/31/2020	(20 years of daily data)
Development:	07/01/1988 - 12/31/1990	(2.5 years of daily data)
Test:	01/01/1986 - 06/30/1988	(2.5 years of daily data)
Backfill:	01/01/1969 - 12/31/1985	(16 years of daily data)

All of the inputs and output of the model were normalized so that all values are within the range of 0 and 1. This normalization prevents large inputs from slowing down learning and convergence of the network. We chose to normalize the data instead of standardizing the data with 0 mean and standard deviation of 1 since the distribution of the input and output data was not normal.

3.2 Feature Engineering

One of the biggest benefits of neural networks is their ability to perform their own feature engineering. The idea is that the neural network will figure out, for itself, what is important and what is not important. With this in mind, our first approach was to provide the raw time-series price data to a neural network. We quickly realized that the network performs quite poorly in this scenario. Financial theory helps explain why this might be the case: the VIX is theoretically independent of the absolute price level of the S&P 500 on which it is derived. Rather, it depends on changes in the price level (ie: volatility). With this in mind, we constructed the following features on the S&P 500:

- One day percentage change in prices
- Trailing 10-day volatility of price changes
- Trailing 30-day volatility of price changes

The idea behind the these features is based in theory: price changes over one day might help with high-frequency aspects of VIX prediction, while 10-day and 30-day trailing volatility calculations might help with the medium and low frequency aspects of VIX prediction.

4 Methods

4.1 Model Architecture

The goal of this project is to predict historical time series VIX values. Hence, a recurrent neural network (RNN) seemed like the most logical approach. A single-layer, many-to-one long short term memory (LSTM) model was used as a baseline model. A LSTM is a special type of RNN which uses gates to add or remove information from the cell state. The output of the model is a single VIX prediction for a given day, and the time series of the features described in section **3.2** are used as input data.

Since we are modeling VIX values in the past, we naturally turned to a bidirectional LSTM network, as we are not directionally constrained in a temporal sense. This architecture is able to learn relationships near and far both before and after the point of prediction. A bidirectional LSTM layer consists of two LSTMs: one taking the input in a forward direction, and the other in a backward direction. The addition of more bidirectional LSTM layers allows the model to learn more complicated relationships between the input and the output. The architecture of the 2-layer bidirectional LSTM model that we used is shown in Figure 1.

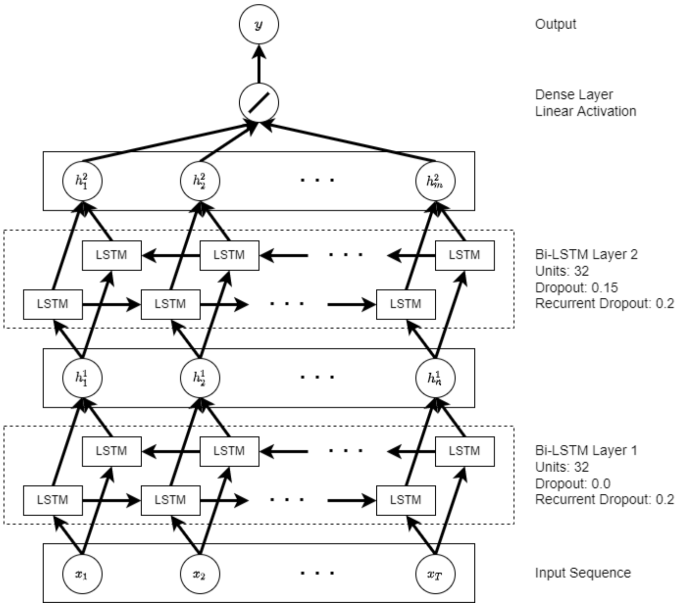

Figure 1: 2-layer, many-to-one bidirectional LSTM Architecture

5 Experiments

5.1 Hyperparameter Tuning

The hyperparameters of the model were tuned by conducting a random search over the configuration space. In the interest of time, we only tested 150 unique model configurations. As a result, given the large size of the configuration space, it is unlikely that we chose the globally optimal set of hyperparameters. Table 1 shows the configuration space and selected hyperparameters.

Table 1: Random search hyperparameter tuning results for deep bidirectional LSTM

Hyperparameter	Type of Sampling	Possible of values	Selection
Learning rate	logarithmic	0.0001 - 0.01	0.002
Number of Bi-LSTM Layers	Discrete Uniform	1,2, or 3	2
Layer 1 units	Discrete Uniform	16, 32, 64, or 128	32
Layer 1 dropout	Discrete Uniform	0 - 0.5 in increments of 0.05	0.0
Layer 1 recurrent dropout	Discrete Uniform	0 - 0.5 in increments of 0.05	0.2
Layer 2 units	Discrete Uniform	16, 32, 64, or 128	32
Layer 2 dropout	Discrete Uniform	0 - 0.5 in increments of 0.05	0.15
Layer 2 recurrent dropout	Discrete Uniform	0 - 0.5 in increments of 0.05	0.2
Layer 3 units	Discrete Uniform	16, 32, 64, or 128	N/A
Layer 3 dropout	Discrete Uniform	0 - 0.5 in increments of 0.05	N/A
Layer 3 recurrent dropout	Discrete Uniform	0 - 0.5 in increments of 0.05	N/A

5.2 Loss Functions

Predicting outlier price movements in the VIX is just as important as predicting the absolute value of the VIX. So, we wanted to see whether different loss functions could help our model better fit these outlier price movements. We trained and fit our model using mean absolute error (MAE), mean squared error (MSE), and a custom loss function that we call mean ˆ4 error (M4E):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad M4E = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^4$$

For this application, the advantages of MSE over MAE are that for small errors, MSE helps converge to the minima efficiently since the gradient reduces gradually and MSE is more sensitive to outliers. So, the idea behind our custom loss function, M4E, is to exaggerate these benefits by squaring the errors once more. Figure 2 shows a comparison of the predictions of the model on the training and development set for each loss function.

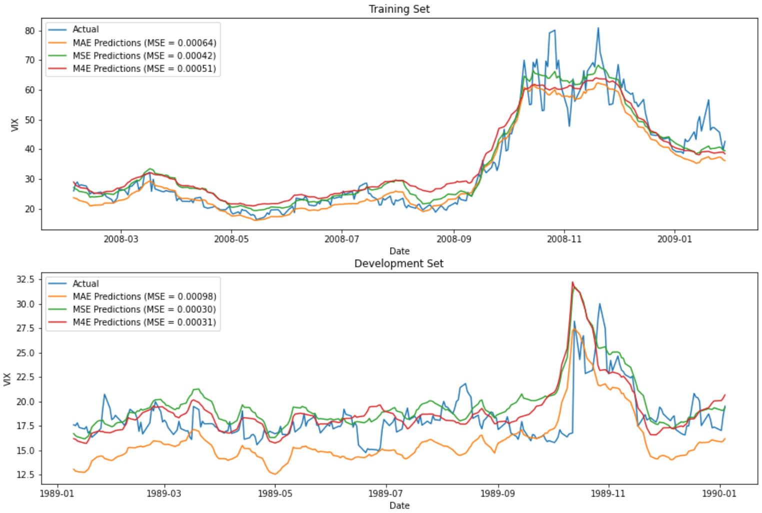

Figure 2: Resulting fit from different loss functions

From figure 2 it's apparent that the predictions using MSE and M4E as the loss functions qualitatively fit the training and development sets better than MAE. However, the predictions from MSE and M4E loss qualitatively seem to fit the data equally as well, the main difference being that the predictions from M4E seem smoother than the predictions from MSE. Since any gain in performance is negligible, any benefit of using M4E loss over MSE loss is outweighed by the popularity of using MSE loss in regression (easy comparison with other models that use MSE loss) and inconvenience of having to define a custom loss function. The quantitative results (the mean squared errors of the predictions using each loss function) also support this conclusion.

6 Results

Initially our model tended to overfit. We wanted to avoid data augmentation since financial data can be small, noisy, and non-stationary. So, instead we aimed to correct this by using hyperparameter tuning to find a reasonable model complexity, introducing dropout, and implementing early stopping.

Using the hyperparameters provided in section **5.1**, and setting Mean Squared Error as our loss function and evaluation metric, the model appears to fit our train, dev, and test set data surprisingly well. Figure 3 shows the output predictions with the mean squared error listed in the legends. Unwittingly, the test data incorporates one of the biggest financial crises in modern history: Black Monday, when 8 global markets, including the U.S. financial markets, declined more than 20%. We are pleased to observe that our model appears to capture much of the VIX's behavior during this time period. However, we note that while our model does a good job with the low and medium frequency fluctuations in the VIX, it fails to capture higher frequency/resolution behavior. Please see the appendix for a time series plot of an option strategy backtest using our model's output between 1969 and 1985.

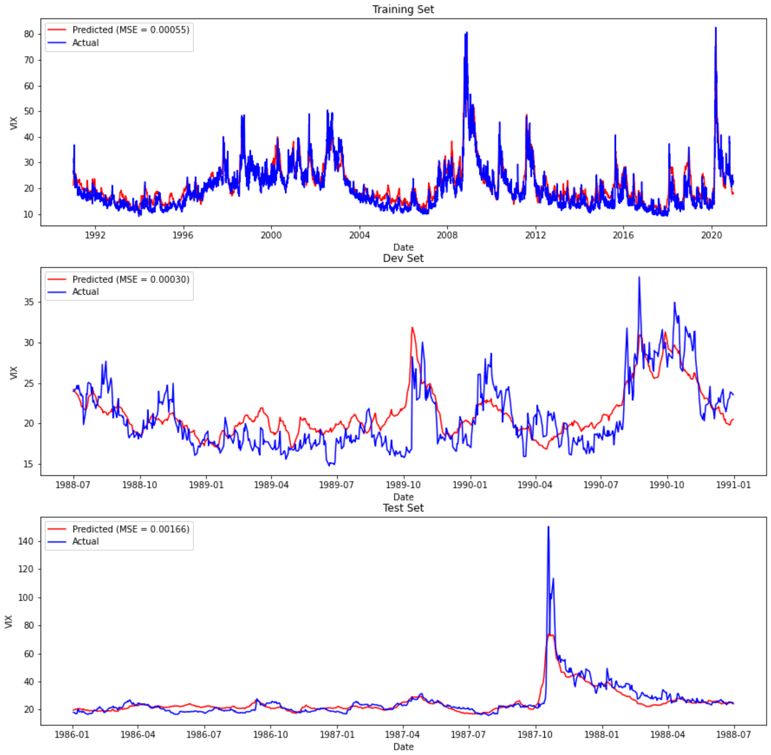

Figure 3: Training, Validation and Test Predictions - Note "Black Monday" crisis in October, 1987

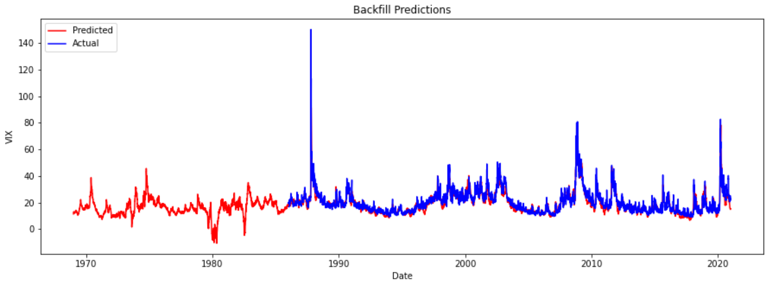

Figure 4: Backfill to 1968

7 Conclusion/Future Work

In this paper, we demonstrate the effectiveness of a 2-layer bi-directional LSTM neural network in backfilling historical VIX data. Our model successfully models the low and medium frequency relationship between the VIX and our input data, although it fails to capture higher frequency behavior in the VIX index. We believe this can be attributed to a divergence between the theoretical meaning of the VIX index (as 30-day implied volatility) and the reality of the market dynamics which influence the VIX level - notably market sentiment/emotions. If the data was available, we believe that greater accuracy could be achieved by using factors generated on intraday data: perhaps hourly price changes or hourly volatility figures. In addition, we expect that benefits could be derived from additional data collection, feature engineering, data augmentation, longer hyperparameter tuning, as well as more sophisticated methods to tackle overfitting.

Contributions

Contribution	**Matt Johnson**	**Alex Moraru**
Data Collection	✓	
Data Processing	✓	✓
Data Visualization		✓
Feature Engineering	✓	
Basic LSTM Model		✓
Bidirection LSTM Model	✓	✓
Hyperparameter Tuning	✓	✓
Evaluating different Loss Functions		✓
Options Strategy Backtester	✓	

References

Chollet, Francois, et al. “Keras”. GitHub, 2015. github.com/fchollet/keras.

Fernandes, M., et al. “Modeling and predicting the CBOE market volatility index”. *Journal of Banking & Finance*, 2014, pp. 1–10. doi:https://doi.org/10.1016/j.jbankfin.2013.11.004.

Global Financial Data. 2021, Accessed through Stanford’s Graduate School of Business. globalfinancialdata.com.

Harris, Charles R., et al. “Array programming with NumPy”. *Nature*, vol. 585, no. 7825, Sept. 2020, pp. 357–362. doi:10.1038/s41586-020-2649-2.

Hosker, J., et al. “Improving VIX Futures Forecasts using Machine Learning Methods”. *SMU Data Science Review*, 2018. doi:https://scholar.smu.edu/datasciencereview/vol1/iss4/6.

Majmudar, U. and A. Banerjee. “VIX Forecasting”. 2004. doi:https://doi.org/10.2139/ssrn.533583.

Martens, M. and J. Zein. “Predicting financial volatility: High-frequency time series forecasts vis-à-vis implied volatility”. *Journal of Futures Markets*, 2004, pp. 1005–1028. doi:https://doi.org/10.1002/fut.20126.

Martin Abadi, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015, Software available from tensorflow.org. www.tensorflow.org/.

Padhi, P. and I. Shaikh. “On the relationship of implied, realized and historical volatility: evidence from NSE equity index options”. *Journal of Business Economics and Management*, 2014, pp. 915–934. doi:https://doi.org/10.3846/16111699.2013.793605.

Pedregosa, F., et al. “Scikit-learn: Machine Learning in Python”. *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825–2830.

Appendix

7.1 Dataset Splitting

Since the purpose of this model is to predict VIX values before data for the index was available, the data was not shuffled before splitting it into training, development, and test sets to possibly give us a more accurate measurement of how the model performs. For example, in practice the model would train on all available VIX data back to 1986 and attempt to predict VIX values before 1986. Assume for a second that the VIX was created in 1930 instead of 1986. In that case, it's very likely that the distribution of VIX data from 1930-1986 is different from the distribution of VIX data from 1986 onward. So, a model to predict VIX values that is trained only on data after 1986 will likely perform more poorly than a model trained on the same number of randomly-sampled examples from the entire date range of available data (1930 onwards). This is exactly what we are trying mimic by only training on data from 1991 onward and validating the model's performance on data from 1986-1991.

7.2 Model Architecture Choice

One of the most common methods of predicting multiple steps in a time series with a RNN is building a model with multiple outputs (one output for each time step being predicted). Since the purpose of this project is to predict VIX values very far into the past, this method would not work well for a couple of reasons. First of all, training a model with a very large number of outputs would take much longer than training a model with fewer outputs. However, the bigger problem is that this method requires a large number of available output data for training. For example, a model built to predict 30 years of daily VIX data into the past would require 30 years of daily VIX data for a single training example. So, given our training set size of 30 years of VIX data, we would only be able to provide 1 training example for such a model.

In order to get around this issue, an alternate method is to at takes in a time series and outputs a single prediction. The model can then be run multiple times to make predictions for multiple time steps. For example, to predict the value of the VIX 3 times setps in the past, the model will be run 3 times using input data from $t^n, ..., t^0$ to predict the VIX t^{-1}, input data from $t^{n-1}, ..., t^{-1}$ to predict the VIX at t^{-2}, and input data from $t^{n-2}, ..., t^{-2}$ to predict the VIX at t^{-3}. Note that if you want to include VIX data as an input to the model and you only have VIX data from t^0 to t^n, you could feed the output prediction of the VIX for t^{-1} as an input to the model to predict the VIX at t^{-2}. However, because predictions are used in place of observations, this recursive strategy allows prediction errors to accumulate such that performance can quickly degrade as the prediction time horizon increases. Since the goal of this project is to predict very far into the past, we decided to not include the VIX itself as an input to the model.

7.3 Model Training and Validation

As discussed in the **Dataset** section, the training set includes VIX data from 01/01/1991 - 12/31/2020 and the validation set includes VIX data from 07/01/1988 - 12/31/1990 as the target output for the model. Since we are using data from the 30 days following the date being predicted as our input to the model, the full date range for the training input actually ranges from 01/02/1991 - 01/30/2021. The Adam optimizer was used during training with mean squared error as the loss function and a batch size of 128. To prevent overfitting we implemented early stopping with a patience of 5 epochs.

7.4 Option Strategy Backtest

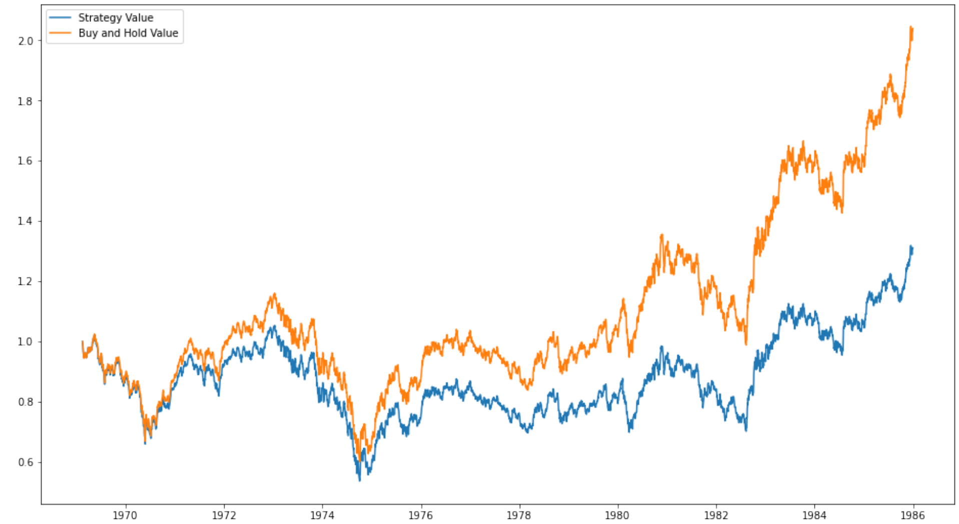

Figure 5: Backtest. The strategy consists of buying a 30-day at-the-money call option whenever the S&P 500 has increased by more than 1% in a month. We sell the option when there are fewer than 10 days until expiration. We allocate 0.5% of the total portfolio to the option, with the other 99.5% going into the S&P 500. The results agree with backtests performed using real VIX data, and illustrate the type of analysis that can be done with our modelled VIX data.