Long Short-Term Memory Networks for Stock Forecasting

Christal O'Connell, Eliot Liucci, Eric Folsom, Nick Clausen

Montana State University - CSCI 550

December 2024

Purpose

- Stock forecasting is infamous as being very difficult
- Market variability can make stock behavior unpredictable
- Day traders use pattern recognition to determine where a stock is likely to go
- Is it possible to use deep learning to mimic this strategy?

Figure: Apple stock over the last year.

Long Short-Term Memory Networks

- Originally proposed in 1997 by Hochreiter [2]
- Special type of Recurrent Neural Network
- Designed to be able to deal with long-term dependencies within sequential data

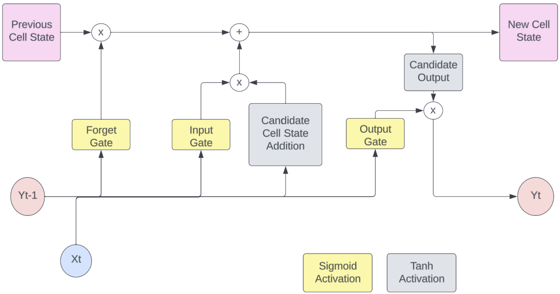
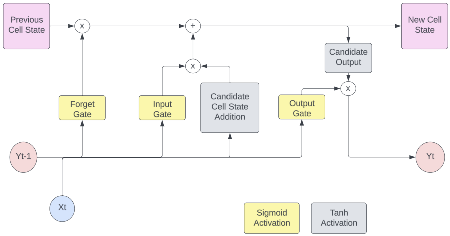


Figure: Inside of a LSTM Memory Cell. Yellow boxes use a Sigmoid activation, grey boxes use a Tanh activation.

Long Short-Term Memory Networks



Name	Equation
Forget Gate	$f_t = \sigma[W_f(y_{t-1}, x_t) + b_f]$
Input Gate	$i_t = \sigma[W_i(y_{t-1}, x_t) + b_i]$
New Candidate Value	$\tilde{C}_t = \tanh[W_c(y_{t-1}, x_t) + b_c]$
Updated Cell State	$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$
Output Gate	$o_t = \sigma[W_o(y_{t-1}, x_t) + b_o]$
Output Value	$y_t = o_t \cdot \tanh[C_t]$

Table: A table of equations corresponding to each gate and node.

Long Short-Term Memory Networks

- Memory cells can be used in place of typical neurons within a neural network
- Number of memory cells and number of layers can be changed

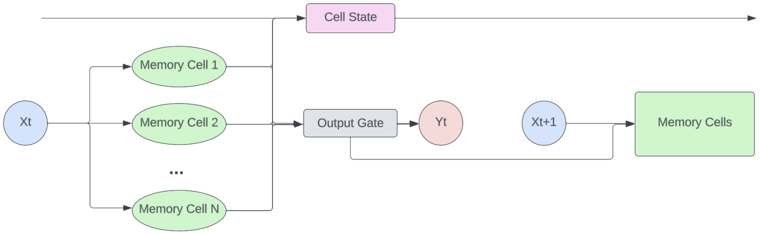


Figure: A figure showing how the memory cells are used when building LSTM networks.

Determining Model Architecture

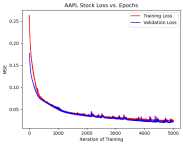
- Each model is a 3-layer LSTM built in `tensorflow` [1]
- Hyper-parameter were tuned using Bayesian Optimization [5]
  - More efficient than randomly searching
  - Similar efficiency to using Latin Square
- Models for all stocks were tuned to have 8-8-64 nodes
- Dropout rates different

	Stock	AAPL	AMZN	CAT	NVDA
1^{st} Layer	Nodes	8	8	8	8
	Dropout	0.01	0.1	0.15	0.15
2^{nd} Layer	Nodes	8	8	8	8
	Dropout	0.05	0.01	0.15	0.15
3^{rd} Layer	Nodes	64	8	64	8
	Dropout	0.15	0.05	0.15	0.15

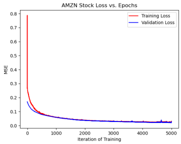
Table: A table of the architectures determined via Bayesian Optimization for each stock.

Training Process

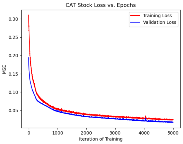
- LSTMs need a lot of sequential data for effective model convergence
- `yfinance` package in `Python` was used to read the past 10 years of daily closing prices in-line [4]
- Each time series was decomposed to remove any trends, resulting in a stationary time series
    - This allows for optimal LSTM performance when recognizing patterns



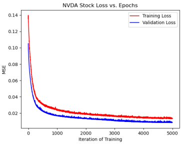
(a) AAPL



(b) AMZN



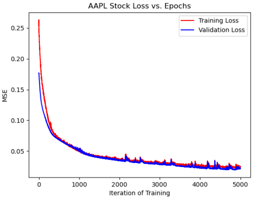
(c) CAT



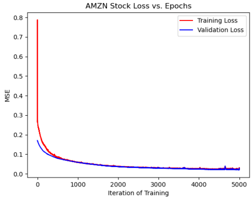
(d) NVDA

Figure: Loss against iterations trained for each stock.

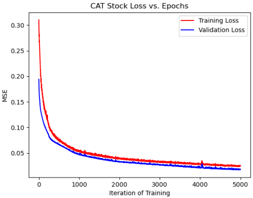
Training Process



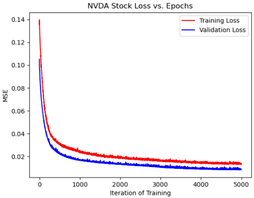
(a) AAPL



(b) AMZN



(c) CAT

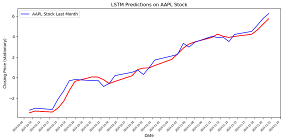


(d) NVDA

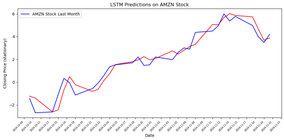
Figure: Loss against iterations trained for each stock.

Results

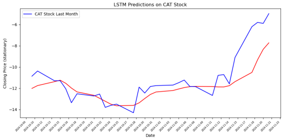
- The last month of closing prices was withheld from training and testing set
- LSTMs were used to forecast 31 days out
- Forecasts were then compared to actual stock performance



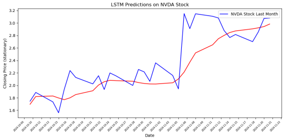
(a) AAPL



(b) AMZN



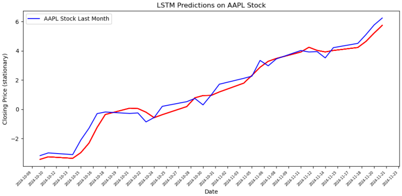
(c) CAT



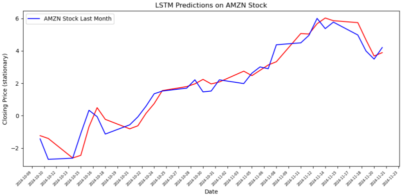
(d) NVDA

Figure: Blind test of 31 day forecasts for each model.

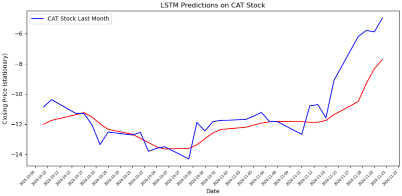
Results



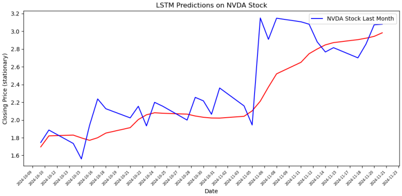
(a) AAPL



(b) AMZN



(c) CAT

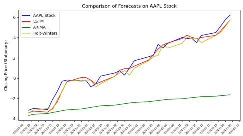


(d) NVDA

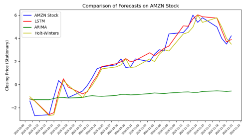
Figure: Blind test of 31 day forecasts for each model.

Results

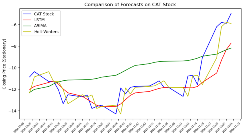
- Blind forecasts from LSTM (red) were compared to other popular time series models
  - Holt-Winters Exponential Smoothing (yellow)
  - Autoregressive Integrated Moving Average (green)



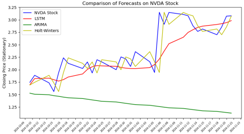
(a) AAPL



(b) AMZN



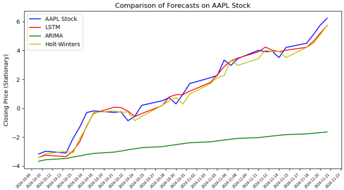
(c) CAT



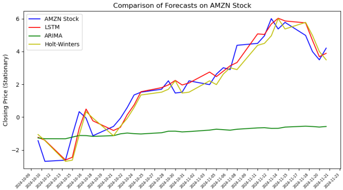
(d) NVDA

Figure: Comparing LSTM forecasts to other methods

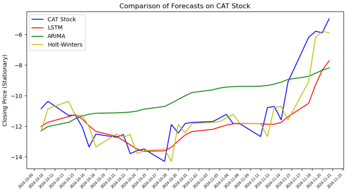
Results



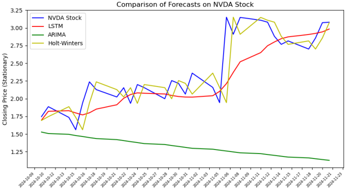
(a) AAPL



(b) AMZN



(c) CAT

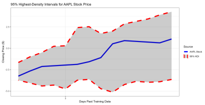


(d) NVDA

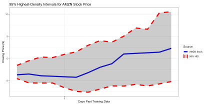
Figure: Comparing LSTM forecasts to other methods

Results

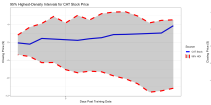
- To get measures of uncertainty for LSTM forecasts, we repeated the following 100 times for each stock;
  - Bootstrapped the training data using a Moving Block Bootstrap [3]
  - Trained tuned model architecture for 8,000 iterations
  - Forecasted 14 trading days out
- 95% highest density intervals were obtained from all forecasts



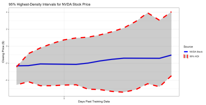
(a) AAPL



(b) AMZN



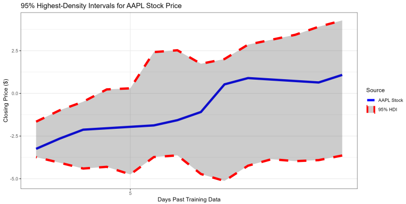
(c) CAT



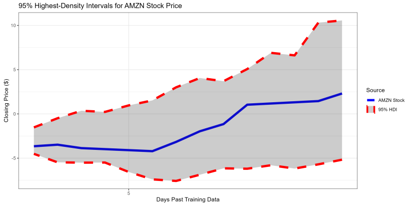
(d) NVDA

Figure: Prediction intervals for each stock.

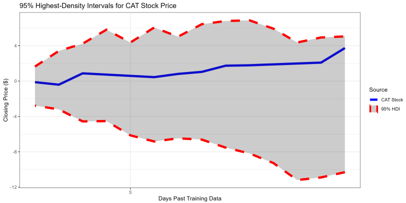
Results



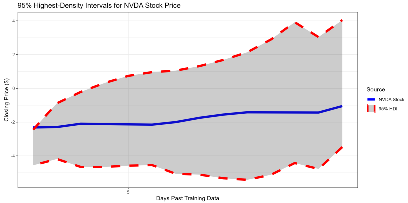
(a) AAPL



(b) AMZN



(c) CAT



(d) NVDA

Figure: Prediction intervals for each stock.

References

[1] M. Abadi and A. Agarwal. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[3] S. N. Lahiri. *Resampling methods for dependent data*. Springer Science & Business Media, 2013.

[4] T. Peters. The Zen of Python. PEP 20, 2004.

[5] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.