LSTM Stock Predictor

Use deep learning recurrent neural networks to model bitcoin closing prices. One model uses the Crypto Fear and Greed Index (FNG) to predict the closing price while the other model uses a window of closing prices to predict the nth closing price.

Please reference the "lstm_stock_predictor_fng_1_10.ipynb" and "lstm_stock_predictor_closing_2_2.ipynb" notebooks.

Preparing the data for training and testing

We:

1. Used the starter code as a guide to create a Jupyter Notebook for each RNN.
2. For the FNG model, we used the FNG values to try and predict the closing price using a function provided in the notebook to help with this.
3. For the closing prcie model, we used previous closing prcies to try and predict the next closing price using the provided function.
4. Used 70% of the data for training and 30% of the data for testing.
5. Applied a MinMaxScaler to the X and y values to scale the data for the model.
6. Reshaped the X_train and X_test values to fit the model's requirement of samples, time steps, and features.

Building and training custom LSTM RNNs

We:

1. Created the same custom LSTM RNN architecture in each Jupyter Notebook, using the FNG values to fit the data in one notebook whileusing the closing prices to fit the data in the other notebook.
2. Used the same parameters and training steps for each model to accurately compare the models.
3. Performed a series of iterations with different window and batch sizes to determine the optimal values for each model, both visually and numerically to minimize the loss amount.

Evaluating the performance of each model

- Which model has a lower loss?

> The Closing model had the lower loss for all window and batch sizes analyzed.

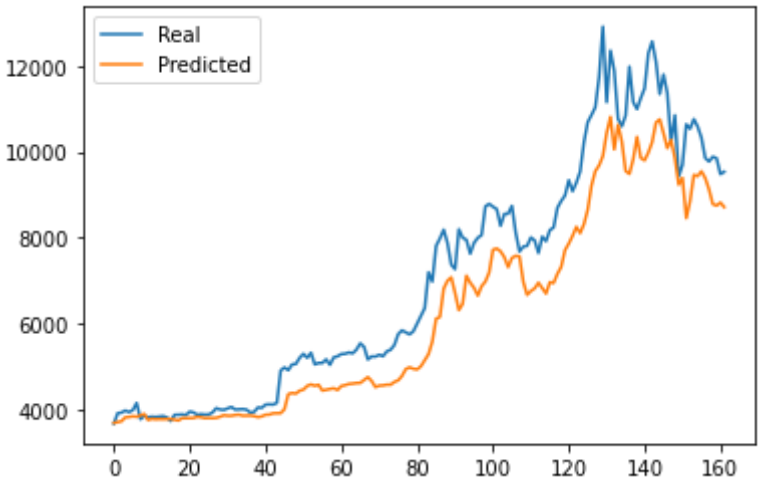
- Which model tracks the actual values better over time?

> The Closing model tracked the actual values better over time (based on visual inspection of the plots as well as the amount of loss).

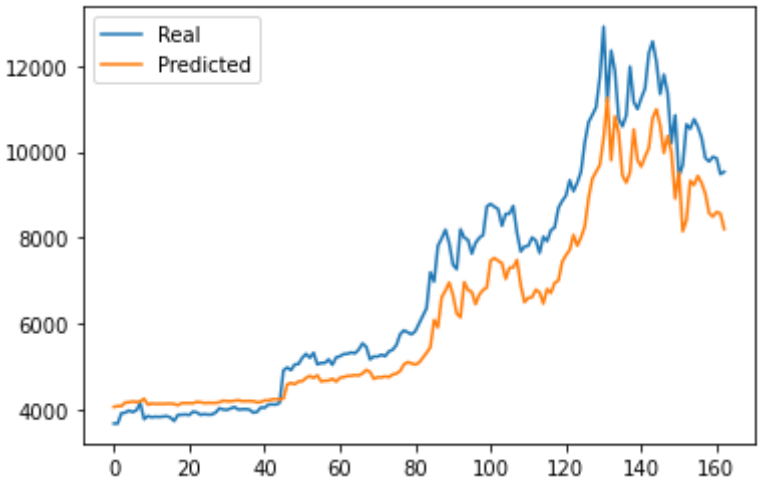
- Which window size works best for the model?

> For the Closing model, we found the optimal parameters were a window of 2 and a batch size of 2. With these settings, loss was minimized at a value of 0.0111. It should be noted that VISUALLY, a window of 1 and batch size of 2 seems to provide a closer fit with less smoothing, however, the loss with these parameters is slightly higher at 0.0115.
>
> For the inferior FNG model, we found the optimal parameters were a window of 1 and a batch size of 10. With these settings, loss was minimized at a value of 0.0823, however, visually the Predicted values approach a flat line as the batch size is increased.

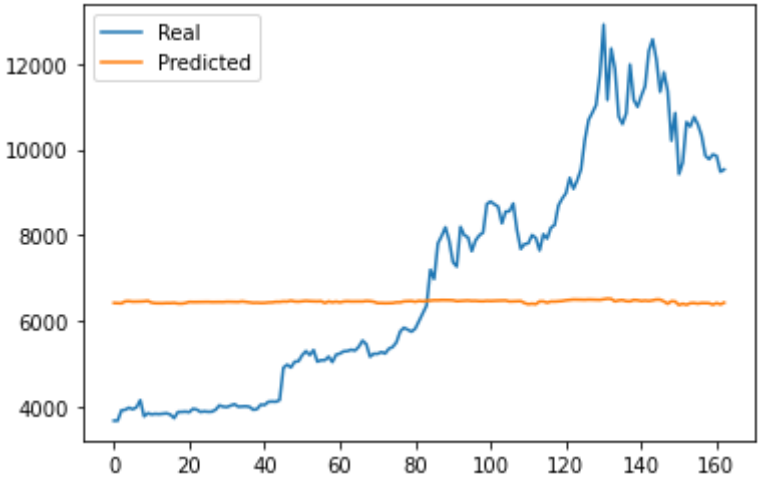
Optimal Closing Model Output (Window=2, Batch=2, Loss=0.0111):



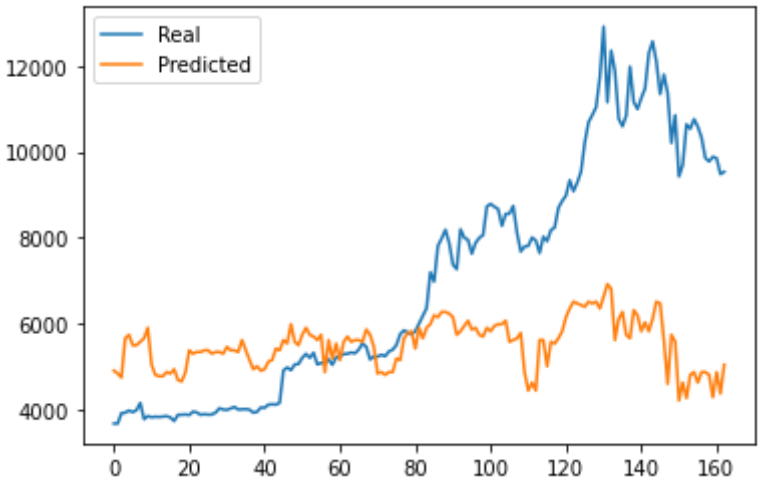
Slightly Less Optimal Closing Model Output (Window=1, Batch=2, Loss=0.0115):



Optimal FNG Model Output (Window=1, Batch=10, Loss=0.0823):



Visually Improved FNG Model Output (Window = 1, Batch = 1, Loss=0.0936):



While the optimal plots are displayed in this document, please feel free to view the other iterations with various parameter values in the "Iterations" directory.

Loss Values for a Variety of Window and Batch Sizes:

Batch=1	FNG	Closing
Window=1	0.0936	0.0163
Window=2	0.0984	0.0237
Window=3	0.0983	0.0237
Window=4	0.0983	0.0279
Window=5	0.0997	0.0279
Window=6	0.0991	0.0328
Window=7	0.1008	0.0367
Window=8	0.1040	0.0370
Window=9	0.1061	0.0432
Window=10	0.1055	0.0428
min	0.0936	0.0163
max	0.1061	0.0432

Window=1	FNG	Closing
Batch=1	0.0936	0.0163
Batch=2	0.0901	0.0115
Batch=3	0.0876	0.0285
Batch=4	0.0853	0.0441
Batch=5	0.0841	0.0570
Batch=6	0.0829	
Batch=7	0.0824	
Batch=8	0.0831	
Batch=9	0.0826	
Batch=10	0.0823	
Batch=11	0.0833	
Batch=15	0.0845	
Batch=20	0.0872	
Batch=50	0.1447	
Batch=100	0.1840	
min	0.0823	0.0115
max	0.0936	0.0570

Batch=2	FNG	Closing
Window=1	0.0901	0.0115
Window=2		0.0111
Window=3		0.0126
Window=4		
Window=5		
Window=6		
Window=7		
Window=8		
Window=9		
Window=10		
min	0.0901	0.0111
max	0.0901	0.0126