Time Series Forecasting with Deep Learning Models

Eric Steen (esteen1@stanford.edu), Orion Darley (oriond@stanford.edu), Ryan Silva (rdsilva@stanford.edu)

Department of Computer Science, Stanford University

Predicting

We apply Deep Recurrent Neural Networks with Long Short Term Memory on time series forecasting, in particular to predict the very volatile Bitcoin (BTC) financial asset price over a number of input sequence lengths and forecast lengths. We reference prior work on ARIMA [1] to give insights on our baseline time series model.

Data

Our Data is the daily historical data of Bitcoin (BTC) sourced from a reputable exchange, along with other features including technical indicators. Our sample was drawn from daily prices since 01/01/2017. Our model performed best after regularization and normalization.

Features

For the Bitcoin index, three categories of daily variables are used as model predictors. The first category contains historical Bitcoin trading data, such as the open, close, high, and low bitcoin prices of the day or hour.

The second category consists of volume and percentage change in price recorded during trading. These variables are widely used in stock and commodities trading.

We also examine derived features as the third type of input. Derived features include the other attributes' variation, exponentially weighted smoothing averages, seasonal decomposition, and trend decomposition, as well as Technical Indicators: MACD, Average True Range, RSI, Bollinger Bands. Prior work on Quantitative trading strategies provides some guidance on this front [2], [3].

Models

In terms of model architecture, we explore a variety of LSTM models with up to six layers, with 3-4 LSTM layers and 1-2 Dense layers with a final Dense ouput layer were the main focus.

We implement dropout and batch normalization after every LSTM layer in the model. Each LSTM layer has the same number of hidden units, which we use as a hyper-parameter in our search.

We model the problem as a many-to-many sequence forecasting problem, and thus the final layer in our models have a number of output units equal to the number of time steps to be forecasted. Our models were trained with a mean squared error loss function applied for the back propagation phase of the learning algorithm. [3]

Results

We implemented a simple looping construct to search the optimal set of hyper-parameters similar to the following pseudo-code:
We plotted our top 25 models with forecasts and evaluated them on Mean absolute percentage error (MAPE), which is defined as:

$$\frac{1}{n}\sum\frac{|Actual - Forecasted|}{|Actual|}$$

We initially focus on quantity of models over quality, trying different architectures and approaches including binary classification to determine optimal window length and forecast length, finally settling on 5-6 layer LSTM architectures which output a continuous future expected price value. We then moved quality to the head of our decision making processes to tune the final models. Rapid exploration turned out to be advantageous in uncovering our most promising models early on. [2]
As you can see below, our predictions showed a solid amount of learning (low mean concise absolute error).

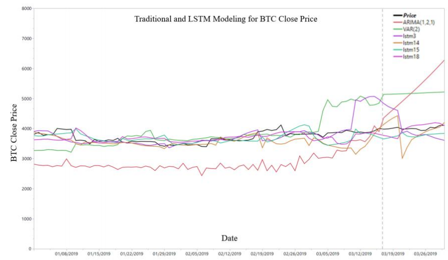

Fig. 1: Plot of our best performing models.

LSTM no. 14 outperformed the top 25 models with a MAPE of 0.022 on the training set and 0.005 on the test dataset. Fourteen other LSTMs outperformed the top performing VAR model and the four ARIMA models ranked at the bottom of the top 25 models list. The best VAR model ranked no. 16 with a MAPE of 0.26 on the training set and 0.267 on the test dataset. The best ARIMA model ranked no. 22 with a MAPE of 0.526 on the training set and 0.477 on the test dataset.

Model Name	Features	# Hidden Nodes/layer	Dropout	Sequence Length	Training Score	Test Score
lstm14	13	50	38%	100	0.022	0.005
lstm18	13	300	42%	130	0.003	0.042
lstm15	13	250	41%	120	0.067	0.051
lstm3	13	50	10%	100	0.122	0.083
VAR(2)	4				0.269	0.293
ARIMA(1,2,1)	1				0.527	0.477

Fig. 2: Best Performing Models

Discussion

After extensive exploratory Our findings were conclusive: We found that Deep Recurrent Neural Networks were superior to ARIMA and VAR models when tuned extensively. The MAPE was significantly better for well-tuned LSTM models. We noticed that normalization and regularization of our inputs, along with low-to-moderate levels of dropout and 5-40 hidden units improved our results.

Future

We are optimistic that further questioning and additional human insights along with better feature engineering, more high quality data, rigorous optimization, and additional techniques such as attention models and other emerging research findings will give significant lift to this research. Technically, we hope to expand upon and improve our analysis both from a high level architecture and hyper-parameter optimization (grid or bayesian search), and in terms of low level technical details such as automation, and agile development methodology.

In particular, the ability to automatically configure the number of layers will be useful, and the ability to run several different architectures and hyper-parameters over a given dataset will be preferred over a high amount of manual tuning. We also plan to add better error handling, unit and integration testing, and improve the overall quality of the code.

References

[1] Zhang Jiang. "Exploration of predicting power of arima." In: ().

[2] Kaur. "Quantitative trading strategies using deep learning, pairs trading." In: ().

[3] Slottje. Shaw Persson. "Hybrid autoregressive recurrent neural network architecture for algorithmic trading of cryptocurrencies." In: ().