Using machine learning and sentiment analysis to predict cryptocurrency price fluctuations

Sean Dubiel with mentorship from David Leland

University of Wisconsin Eau Claire

Introduction

Google's Machine Learning Library, Tensorflow, and the Python programming language were applied to predict cryptocurrency (e.g. Bitcoin) prices using technical indicators and quantified sentiment analysis. Sentiment analysis takes human readable language and can rate it on various qualities, in this case positive/negative sentiment and the objectivity/subjectivity of statements. Cryptocurrency prices are highly driven by speculation so combining sentiment analysis with traditional historical price analysis improves prediction. Data for sentiment analysis came from social media and website news on cryptocurrency. Using Recurrent Neural Networks trained with data from historical price analysis and quantified sentiment analysis from social media. Once trained on the given data, the program can detect patterns and precursors to market movements. Prediction accuracy is calculated and validation testing is conducted to improve performance. Future price points of various cryptocurrencies were predicted with better than 50% accuracy. Robustness of this program can be improved with more data and more validation testing, with the potential of automating this accuracy checking process. This work could also be applicable to other data prediction problems, for instance identifying precursors to artifacts (e.g. eye blinks) in electroencephalography (EEG) data to assist in removing or correcting noisy data segments.

Method

Setting Up Environments

- Tensorflow and supporting libraries are installed.
- I do most of my development on Linux and sometimes Windows.

Preparing Data

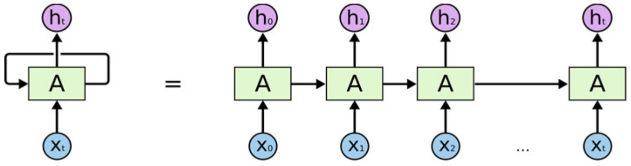
- I start with the date, price open, price low, price high, and percent change**.** For more accurate predictions I could use a smaller time sample than a full day. This would require more time and processing power.
- Features are pulled from the pricing data, these technical indicators are input as functions to be used on the data. Technical indicators can be changed or added to.
- Sentiment Analysis is used as an indicator, I use the volume of keywords found and their levels of positivity and negativity. There are many good sources to pull from, such as social media platforms and popular news sites. For the tests seen here, Twitter was used as the source.
- Once all technical indicators have been extracted they are placed into another file that will be used by the prediction software.

Prediction

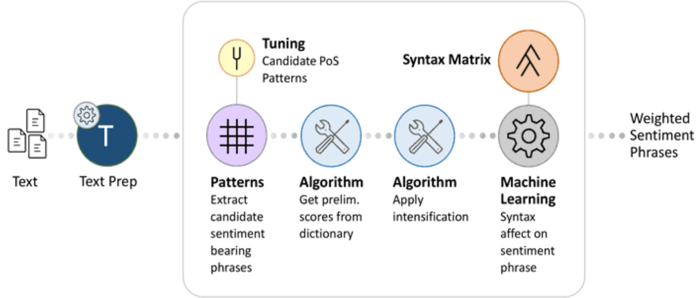
- Data prepared earlier is read in and normalized.
- Tensorflow hyperparmaters are set.
- Training is executed.
- For information on how Tensorflow is used refer to www.tensorflow.org for documentation.
- Results are outputted into a file and plotted for verification.

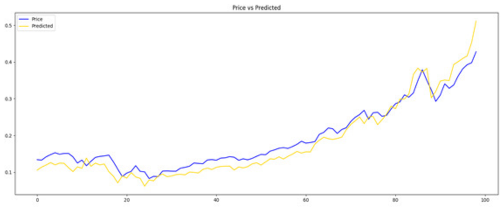
Analysis and Adjustments

- Future price points of various cryptocurrencies were consistently predicted with better than 50% accuracy. Often times predicted with much higher than 50% accuracy.
- Sentiment Analysis seems to work better with cryptocurrencies that have smaller volume. This could be because of their increased volatility and reactions to trends.
- Adjustments to Tensorflow can be made to improve accuracy, along with the introduction of more price points and technical indicators.
- Moving forward I would like to use more pricing and sentiment data, try different technical indicators, and use more computing power.

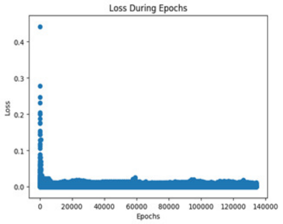


A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.


Tuning
Candidate PoS Patterns
Syntax Matrix
Text
T
Text Prep
Patterns
Extract candidate sentiment bearing phrases
Algorithm
Get prelim. scores from dictionary
Algorithm
Apply intensification
Machine Learning
Syntax affect on sentiment phrase
Weighted Sentiment Phrases



This is 100 days of bitcoin price trends next to the predictions. Actual prices are not included on the y-axis.



The lower the loss, the better the model. This shows the improvement over epochs (passes through the dataset) during the prediction phase.

Acknowledgements: Support provided by Student Blugold Commitment Differential Tuition, UW-Psychology Department, UW-Computer Science Department, and UW-Eau Claire Learning and Technology Services.

Discussion: Originally I set out to reduce artifacts in EEG data. In the future I would like to apply these concepts to that field. Using machine learning and Tensorflow specifically, we could reduce the time spent correcting or removing noisy data.

We thank the Office of Research and Sponsored Programs for supporting this research, and Learning & Technology Services for printing this poster.