Predicting Bitcoin Price Trends

Cindy Jiang
cindyj@stanford.edu
Department of Computer Science, Stanford University

Orien Zeng
ozeng@stanford.edu
Department of Computer Science, Stanford University

Lucas Lin
lucaslin@stanford.edu
Department of Computer Science, Stanford University

Overview

Goal: Predict whether bitcoin prices will go up or down based on a snippet of Bitcoin trading history.
Dataset: Time series data of transactions on a Bitcoin Coinbase exchange
Output: Binary classification to 1 (price increases) or 0 (price decreases).
Approach: 3-component model with a Fourier transform, stacked autoencoder, and LSTM.

1. Fourier transform: Extract frequency content and denoise input
2. Autoencoder: Compress Fourier output and extract high-level features
3. LSTM: Input the sequence of encoded outputs for classification

Results

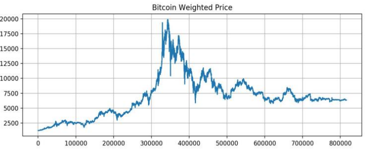
Model	Accuracy	Precision	Recall	F1
Baseline CNN	60.8%	66.7%	**72.82%**	64.78%
LSTM #1	69.4%	**74.24%**	63.64%	67.22%
LSTM #2	**69.8%**	70.88%	65.44%	**67.82%**

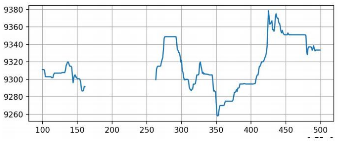
LSTM #1: 3.2e-5 learning rate, no dropout, 200 epochs and 1 layer
LSTM #2: 1.3e-4 learning rate, no dropout, 100 epochs and 2 layers

Dataset

Kaggle: Bitcoin Historical Data

- 4 years in 1-minute increments (around 2 million rows)
- 7 features
  - Weighted Price
  - Volume

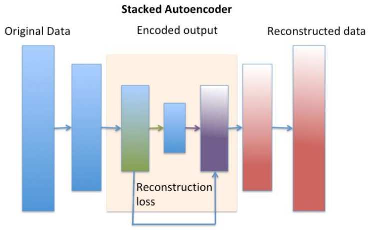




Preprocessing

Split dataset 70%/15%/15%

- *Problem*: NaN's in dataset
  - Consists of around 5.4% of the entire dataset
  - Discarded or interpolated NaN's based on density of "good" points around the NaNs



The layers in the box shown above
represent the layers currently being trained in the model

Discussion

Limitations

- Small dataset
- Unlabeled regions in dataset

Future work

- Profitability analysis
- Wavelet vs fourier
- Input gradient heatmap

Architecture

Fourier transform

- Time series to frequency
- Rescale channels to (-1, 1)
- Phase info as angle in the complex plane

3-layer stacked Autoencoder

- Layers trained in series
- Encoding loss is separate for each layer

LSTM

- Stack 10 timesteps
- Hyperparameters chosen via grid search
- Fully connected layer in output with sigmoid activation

Reference

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7), e0180944.