Machine Learning Engineer Nanodegree Capstone Report

Daniel Chao Zhou

31 December 2019

1 Definition

1.1 Project Overview

The stock market prediction has been identified as a significant practical problem in the economic field. Trading algorithms rather than humans performed over 80% of trading in the stock market and the FOREX market. In the crypto-currency market, algorithmic trading is also a hot topic among investors. However, timely and accurate prediction of the market is generally regarded as one of the most challenging problems, since the environment is profoundly affected by volatile political-economic factors, such as legislative acts, political unrest, and mass irrational panic.

There are many studies regarding algorithmic trading in financial markets based on machine learning, where recurrent neural network (RNN) and reinforcement learning (RL) are being popular in recent years. In this study, a Bitcoin price predictor based on long short-term memory (LSTM, a variant of RNN) is presented.

1.2 Problem Statement

Given the trading data of a Bitcoin futures contract with each time step indicating one minute, the goal is to build a predictor for the volume-weighted average price (VWAP) of the next minute.

Table 1: column header semantics

`high,low`	highest/lowest price (two columns)
`close`	price of the last trade
`open`	`close` of the last time step
`vwap`	volume-weighted average price (VWAP)
`foreignNotional`	traded amount in units of US dollar
`homeNotional`	traded amount in units of Bitcoin
`trades`	number of trades
`volume`	alias of `foreignNotional`

Two predictors, an LSTM model as solution and a linear model as benchmark will be built and compared with the metrics as discussed as follows.

1.3 Metrics

The mean squared error (MSE) between labels y and predictions $\hat{y}$ will be used to evaluate the performance of both of the benchmark model and the solution model. For a given integer N and a time-series dataset, all consecutive sub-sequence of the time-series with length N will be used and equally contribute to the final MSE.

2 Analysis

2.1 Data Exploration

In this study, trading data of BitMEX's XBTZ19 contract, which is a Bitcoin futures contract expiring in December 2019, will be used to train and test the predictor. The dataset could be fetched from BitMEX's official API without charge of fee. The API concerned with the desired data is documented at `https://www.bitmex.com/api/explorer/#!/Trade/Trade_getBucketed`.

The dataset is a table that each row indicates one minute and each column indicates a specific data described as Table 1. Note that `open` is not defined as the price of the first trade in the specific time step, which in an unconventional definition and does not apply to other data sources.

Table 2: head and tail part of the dataset

	open	high	low	close	vwap	foreignNotional	homeNotional	trades
2019-06-14 08:31	NaN	NaN	NaN	NaN	NaN	0	0.000000	0
2019-06-14 08:32	NaN	NaN	NaN	NaN	NaN	0	0.000000	0
2019-06-14 08:33	8500.00	8500.00	8260.00	8260.00	8262.4143	201	0.024328	3
2019-06-14 08:34	8260.00	8390.00	8308.00	8308.00	8325.0083	13110	1.574852	7
2019-06-14 08:35	8308.00	8390.00	8319.50	8336.50	8322.9297	13200	1.585986	5
2019-06-14 08:36	8336.50	8317.50	8317.50	8317.50	8317.5000	10200	1.226346	3
2019-06-14 08:37	8317.50	8327.50	8327.50	8327.50	8328.0000	500	0.060040	1
2019-06-14 08:38	8327.50	8366.50	8362.50	8362.50	8365.4007	4001	0.478290	5
...	...	...	...	...	...	...	...	...
2019-12-27 11:54	7151.00	7155.50	7135.00	7137.50	7149.4960	632224	88.434680	62
2019-12-27 11:55	7137.50	7149.50	7137.50	7149.00	7141.3269	60153	8.423772	44
2019-12-27 11:56	7149.00	7142.00	7140.50	7141.50	7140.8169	407108	57.015170	34
2019-12-27 11:57	7141.50	7149.50	7141.00	7141.50	7141.3269	325738	45.615289	22
2019-12-27 11:58	7141.50	7150.00	7141.50	7150.00	7149.4960	390967	54.686307	31
2019-12-27 11:59	7150.00	7150.00	7141.00	7148.50	7142.8571	540701	75.700246	30
2019-12-27 12:00	7148.50	7148.50	7138.24	7138.24	7138.7778	41934166	5874.536073	59
2019-12-27 12:01	7138.24	7138.24	7138.24	7138.24	NaN	0	0.000000	0

Table 3: basic statistics of the dataset

	open	high	low	close	vwap	foreignNotional	homeNotional	trades
count	282449.00	282449.00	282449.00	282449.00	243964.00	2.82e+05	282451.00	282451.00
mean	9499.88	9502.96	9496.68	9499.87	9558.33	3.69e+04	3.94	23.43
std	1621.16	1622.74	1619.51	1621.17	1632.99	1.37e+05	16.63	46.27
min	6438.00	6443.50	6431.00	6438.00	6432.52	0.00e+00	0.00	0.00
25%	8158.00	8159.50	8157.00	8158.00	8180.62	6.05e+02	0.06	2.00
50%	9542.50	9546.00	9540.00	9542.50	9589.56	7.33e+03	0.77	10.00
75%	10616.50	10619.50	10614.00	10616.50	10656.43	3.20e+04	3.36	26.00
max	14600.00	14600.00	14539.00	14600.00	14547.57	4.19e+07	5874.53	1529.00

The dataset is formulated as $\{x_t | t = 1, 2, \ldots, T\}$, where x_t is a vector of the data in minute t, such that $x_t = (\texttt{open}_t, \texttt{high}_t, \texttt{low}_t, \texttt{close}_t, \texttt{vwap}_t, \ldots)$. A small sample and basic statistics are given in Table 2 and Table 3.

2.2 Exploratory Visualization

The VWAP and volume of the dataset is shown in Figure 1.

2.3 Algorithms and Techniques

The solution model consists of one LSTM layer and one linear layer. The LSTM layer is formally formulated as follows:

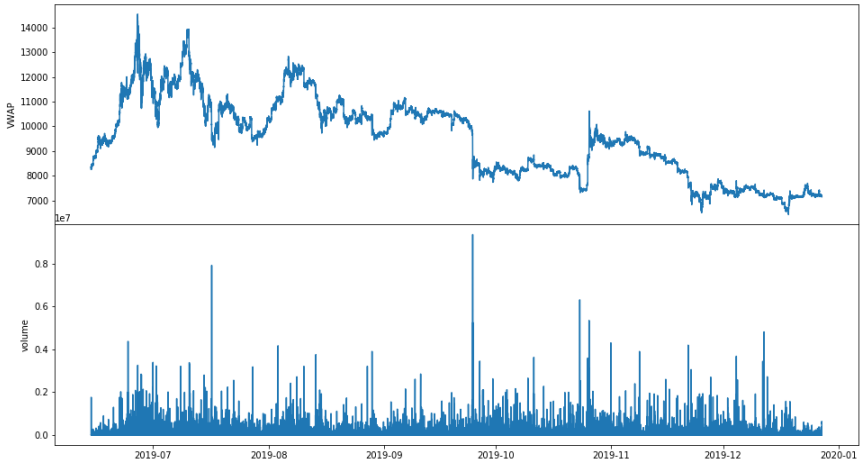

Figure 1: VWAP and volume of the dataset

$$
\begin{aligned}
i_t &= \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi}\right), \\
f_t &= \sigma\left(W_{if}x_t + b_{if} + W_{hf}h_{(t-1)} + b_{hf}\right), \\
g_t &= \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{(t-1)} + b_{hg}\right), \\
o_t &= \sigma\left(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho}\right), \\
c_t &= f_t * c_{(t-1)} + i_t * g_t, \\
h_t &= o_t * \tanh(c_t).
\end{aligned}
$$

where h_t is the hidden state at time t, c_t is the cell state at time t, x_t is the input at time t, and i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively. σ is the sigmoid function, and $*$ is the Hadamard product.

2.4 Benchmark

The benchmark model consists of two linear regression layers:

$$h = \mathrm{ReLU}\left(xA_1 + b_1\right),$$
$$y = hA_2 + b_2.$$

3 Methodology

3.1 Data Preprocessing

Data preprocessing steps are listed as follows.

Crawling Since new data are generating every minute, new rows could be fetched from the data source. The crawler should be able to handle locally cached data and progressively persisting new data.

Column dropping Columns `open` and `volume` do not provide new information and are dropped.

Null filling For time steps that do not contain any trades, the corresponding `vwap` columns are null. These items will be propagated with last valid observation with `pandas.DataFrame.ffill()`.

Feature engineering Moving average convergence/divergence (MACD) with short period of 12 and long period of 26, and relative strength index (RSI) with dataframe of 14, are calculated and appended as extra features for the following training and evaluating.

Normalisation All features will be normalised.

Labelling Each row will be labelled a learning target, with the VWAP of the next row.

Splitting The entire dataset will be split without shuffling into three consecutive parts for training, validation, and testing, while the lengths proportionate to 6:2:2.

3.2 Implementation

MSE and the LSTM model could be easily implemented with PyTorch's builtin `torch.nn.MSELoss` class and `torch.nn.LSTM` class.

3.3 Refinement

The learning rate is automatically adjusted with the scheduler `torch.optim.lr_scheduler.StepLR`. Moreover, the number of dimensions of the hidden layer are tuned with AWS SageMaker's hyperparameter tuning, which suggests that