

Thomas Han

## Deep Learning Assignment 4

Part I:

The implementation can be found in both `datasets.py` and `prediction_game.py`.

I implemented the function `generate_x_y_data_v5()` that basically has very similar implementation as `v4()`. Except it has a `load_purpose` input and when it's set to 1, it loads the latest 300 days of data if the data file is absent.

Also, I implemented function like `play_a_game()`, `buy_or_sell()`, `buy_order()` and `sell_order()`. The first function gets `x0`, `expected_output` and `x1 ~ x5`, and also handles making the decision and place transaction orders. `buy_or_sell()` handles making the decision based on `x0 ~ x5`, and the `buy_or_sell()` and `buy_order()` are used to make transactions.

Let's compare the results then:

Seq_length	Batch_size	Hidden_dims	Num_iters	Result	Comments
60	100	200	1000	\$110047.03	If you go to chart, 1000 iterations is not necessary because the loss goes almost flat around 300. But later I kept using 500 iterations to be safe.
60	200	200	500	\$110125.74	Trying to crank up the batch size and see if it works, and it did.
60	500	200	500	\$110237.58	Kept going at 500 batch_size, but the change is very minial.
260	100	200	500	\$100000.00	I picked 260 because there are 52 trade weeks a year, and there has to be strong pattern over 260 days, but it didn't work as I thought it would.

Conclusion:

I don't think it is even a realistic approach to predicting stock prices because you need more information and you have to understand the fundamentals of the industry and some accounting reports. However, I am rather surprised from the results I was getting. Nonetheless, given SPY was on fire last year (extremely low volatility, stable growth), and considering SPY had a soaring over 20% return last year, my result was good, but still lost (to SPY actual).

## Part II:

Part two was relatively easier than part one. All I did was when loading the stock data from file, it converts every five points to the average of those five points. As you can see from the command line output, the shape of data decreased from (1247, 60, 1) to (248, 12, 1). Notice that the seq\_length is changed from 60 to 12 (from days to weeks,  $60 / 5 = 12$ ) as well. Per requirement, the num\_predictions changed from 300 to 60 to reflect the changes from days to weeks.

Seq_length	Batch_size	Hidden_dims	Num_iters	Result	Comments
60	500	200	500	\$110237.58	The best result from part1.
12	500	200	500	\$108294.41	Reading data, one can find the trade frequency decreased, and the bot became less willing to sell stocks (actually didn't sell anything at the end, the result is converted from the 210 shares it owns, and \$50,720 cash it has).

## Conclusion:

This way is probably less profitable and trade frequency is lower, but it's very hard to tell anything more than that considering how limited the size of data is and how market being volatile.

### Part III:

We are to change from 5 days to 9 days; the changes are reflected in the code of `play_a_game()`: it was predicting for 5 days, now it does 9. Also I changed the `buy_or_sell()` function too, which I will explain next.

The original function:

Sell:  $x_0 > (x_1+x_2)/2 > (x_3+x_4)/2 > x_5$

Buy:  $x_0 > (x_1+x_2)/2 > (x_3+x_4)/2 > x_5$

Then I changed the equation for nine days to:

Sell:  $x_0 > (x_1+x_2)/2 > (x_3+x_4)/2 > (x_5 + x_6)/2 > (x_7+x_8)/2 > x_9$

Buy:  $x_0 < (x_1+x_2)/2 < (x_3+x_4)/2 < (x_5 + x_6)/2 < (x_7+x_8)/2 < x_9$

Result:

Seq_length	Batch_size	Hidden_dims	Num_iters	Result	Comments
60	500	200	500	\$110, 237.58	The best result from part1.
60	500	200	500	\$113, 107.50	Reading the data, one can say the result is better, and the bot is definitely more willing to trade at the early stage comparing to our part1 and part2 bots, but not as willing to sell as previous bot (Note that, same as part 2 bot, this bot held 18 shares when the game is finished).

### Conclusion:

As you can see from the previous table and combined from the console output, the bot becomes more willing to trade (however being limited by its capital). I think it is more willing to buy because the market's trend was very steady in 2017 and having a 9-day prediction makes it easy to recognize this pattern.