Investing in SPY ETF: Deep Learning on SPY Constituents' Momentum Data

Dale Angus
dangus@stanford.edu

Introduction

Could an ETF constituents' price and related information derived from a series of prices, e.g., momentum, be used to reliably predict the ETF's price outcome? Two deep learning models were used to answer this question.

- **Model A** predicts outcome as either up or down
- **Model B** predicts outcome as falling within a defined percentage range

Dataset

- The **SPY** ETF has **504** constituents.
- The dataset is composed of the **2-year historical data**, around **527** trading days.
- The dataset has around **265,600** rows.
- The dataset is **split 90-10** between the **Training set** (239,040 rows) and **Test set** (26,560 rows)
- The price data was gathered from Yahoo Finance. The other features are computed.

Features

The 16 features are derived from the following:

- Weighting and Sector
- Open*, High*, Low*, Close
- Simple Moving Averages, 10 and 20-day*
- Linear Regression Slopes, 10 and 20-day*
- Awesome Oscillator* and 34-day Momentum*
- A derivative of the 21-day 2nd degree Polynomial Regression*
- Price ratio information of **related ETFs**, i.e., **DIA, QQQ, IWM**
- Value of the VIX index

**normalized by dividing by the close price*

Models

Model A	
Type	**6-Layer Binary Classifier**
Layer Dimensions	16, 15, 10, 9, 5, 4, 1/ 13, 12, 8, 6, 4, 2, 1[1]
Output	**0 or 1**
Framework Used	n/a (**NumPy**)
Learning Rate	.0075[1]
Iterations	100,000 or until difference between costs is <= 7×10^{-6}

Model B	
Type	**3-Layer Softmax Classifier**
Layer Dimensions	16/13, 25, 12, 8[1]
Output	**0 to 7** which corresponds to the following ranges, $\begin{bmatrix} \Delta \leq -3\% \\ -3\% < \Delta \leq -2\% \\ -2\% < \Delta \leq -1\% \\ -1\% < \Delta \leq 0\% \\ 0\% < \Delta \leq 1\% \\ 1\% < \Delta \leq 2\% \\ 2\% < \Delta \leq 3\% \\ 3\% < \Delta \end{bmatrix}$
Framework Used	**Tensorflow**
Minibatch Size	32
Learning Rate	.0001
Optimizer	**Adam**
Epochs	1500

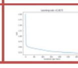
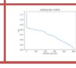
Cost function for both models

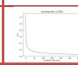
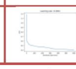
$$-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log\sigma\left(z^{|L|(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-\sigma\left(z^{|L|(i)}\right)\right)\right)$$

Both use the Xavier Initialization.

[1] *with/without related ETF information*

Results

Model A		
	w/ related ETF data	w/o related ETF data
Training Accuracy	91.07%	60%
Test Accuracy	90.89%	59.60%
Cost Plot		

Model B		
	w/ related ETF data	w/o related ETF data
Training Accuracy	98.3%	80.25%
Test Accuracy	98.1%	80.29%
Cost Plot		

Adding other related price ratios to the set of momentum features dramatically improved the accuracy of the two models. This could be due to the high correlations between ETFs.

ETF	SPY	DIA	QQQ	IWM
SPY	1.00 1.00	0.96 0.98	0.93 0.97	0.92 0.96
DIA	0.96 0.98	1.00 1.00	0.83 0.92	0.87 0.94
QQQ	0.93 0.97	0.83 0.92	1.00 1.00	0.82 0.91
IWM	0.92 0.96	0.87 0.94	0.82 0.91	1.00 1.00

ETF Correlation Matrix
http://www.quantf.com/ETF-correlations.php

Use Case and Future Work

In production, using real-time data as substitute for closing data, these can be used to make trading decisions before the market closes.

Future improvements could be done in finding better features; and to predict the next day's outcome, next 2 days and so on. This can be replicated for other ETFs.