Portfolio Management Using Deep Q Learning

Hamza El-Saawy (helsaawy), Olivier Jin (ojin)
CS229, Stanford University

Predicting

Building on previous work, we trained neural networks (NN) to manage a simple two-stock portfolio using Deep Q Learning (DQL) [1,2]. Each portfolio was made up of a high risk and a low risk stock, and the networks were trained on a variable number of closing price data points and reward functions.

Data

Closing stock prices were pulled from Google's API for the period of July 2001 to July 2016 [5]. 10 high risk and 10 low risk stocks were used, for a total of 100 different portfolios (See Figure 1).

Features

Our features given to neural network was the number of shares owned in each stock, the portfolio's current worth, the amount of cash leftover, and each stock's history for the current and prior days.

Models

Our DQL algorithm trained NNs using an experience replay (size 8) and a target network whose weights (w_T) were updated with: $w_T = w_T(1-\square)+\square w$, where w is the models network's weights ($\tau=10^{-3}$) [3,4]. Models were trained with different reward functions (Sharpe ratio, penalized return) [1,2] and number of days stock history (2 or 7). The Sharpe ratio is the average return divided by the return's standard deviation [1]. The penalized return was the previous day's return minus λ times the standard deviation of all the returns thus far, were λ was 0 or 0.5.

Results

Table 1 presents a summary of the different model's results vs two benchmarks: one where the portfolio was rebalanced every 30 days to maintain an even 50/50 split, and one where the portfolio was left as it.

Future

Given another 6 months, more focus would be given to tuning the model training environment, such as exploring the effect of more or less explorations or changing the neural network architecture to be be deeper or simpler. Moreover, generating better features to feed the neural network is another area that could yield significant gains.

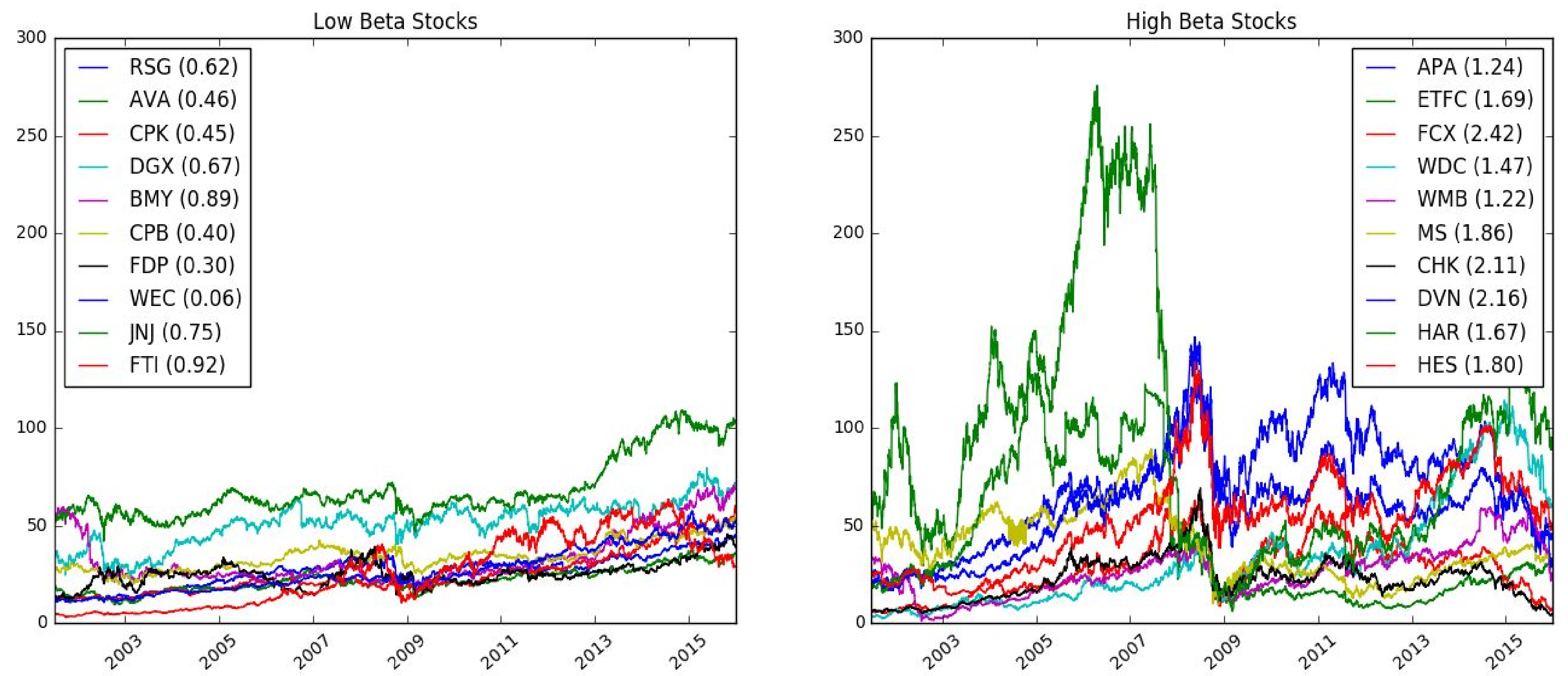

Figure 1: Stock prices for our chosen stocks. Each stock's beta value, an indicator of volatility, is in parentheses next to its symbol

	Sharpe Ratio	Avg. Return	Std. Dev. Returns
2 days, Penalized return (λ = 0)	-0.0068	-274.23	46229.25
2 days, Penalized return (λ = 0.5)	-0.0069	-274.23	45472.29
2 days, Sharpe ratio	-0.0069	-274.23	46229.25
7 days Penalized return (λ = 0)	-0.0051	-274.61	150175.60
7 days, Penalized return (λ = 0.5)	-0.0068	-274.61	46253.49
7 days, Sharpe ratio	-0.0051	-274.61	150367.10
Rebalance 30 Benchmark	0.0153	928.74	45132.28
Do Nothing Benchmark	0.0096	767.28	55842.02

Table 1: Performance of various models vs benchmarks

Discussion

Although the agent did quite well on some portfolios, (Figure 4), its average performance was quite lacking (Table 1). This ties back to both at the inability to beat the market and the simplicity of our model. Another interesting fact is the similarity of average standard deviation for the test set returns. It is likely that the models are converging on similar policies.

References

[1] John Moody and Mathew Saffell. Learning to trade via direct reinforcement. IEEE Transactions on Neural Networks, 12(4)

[2] Xiu Gao and Laiwan Chan. An Algorithm for Trading and Portfolio Management using Q-Learning and Sharpe Ratio Maximization. Proceedings of the International Conference on Neural Information Processing, 2000.

[3] Ben Lau. Using Keras and Deep Deterministic Policy Gradient to play TORCS, Oct 2106. URL yanpanlau:github:io/2016/07/10/FlappyBird-Keras:html. Accessed: 2016-11-18.

[4] Silver, David. Deep Reinforcement Learning. URL www:iclr:cc/lib/exe/fetch:php?media=iclr2015:silver-iclr2015:pdf. Accessed: 2016-11-18.

[5] Google Finance. URL www:google:com/finance. Accessed: 2016-11-18.

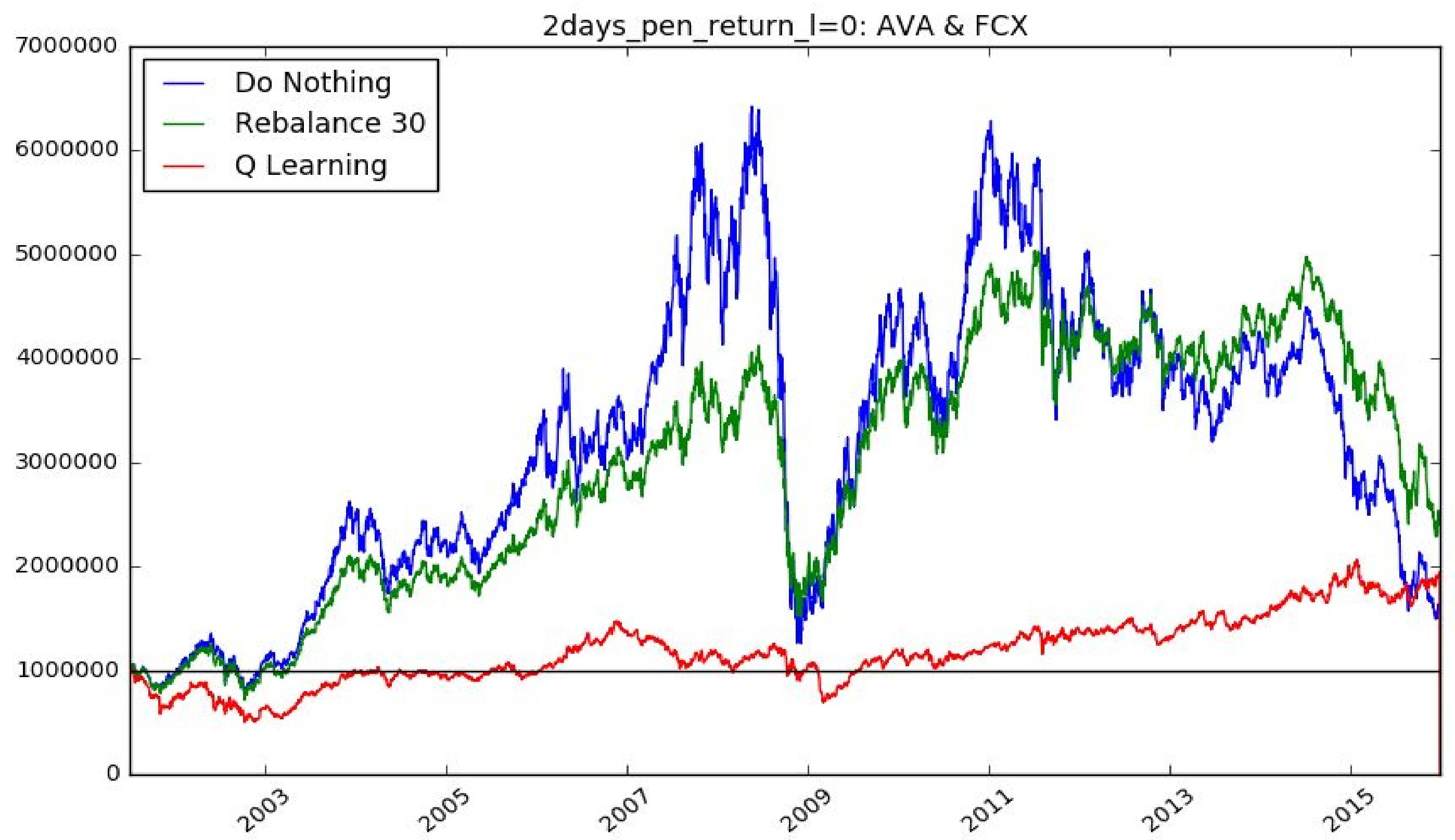

Figure 2: A portfolio for a network feed 2 days of data and using portfolio return as the reward

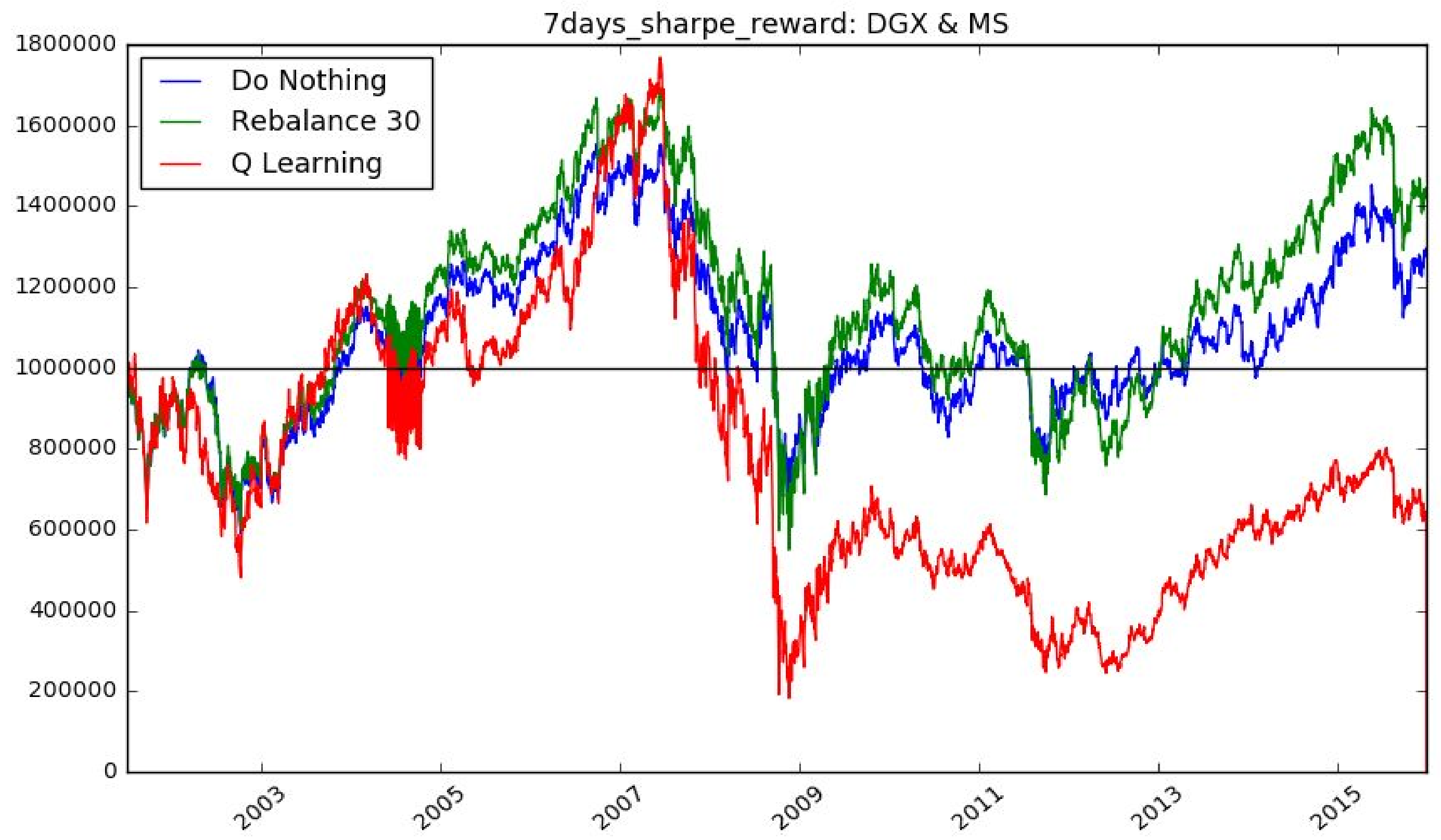

Figure 3: A portfolio for a network feed 7 days of data and using the Sharpe ratio reward

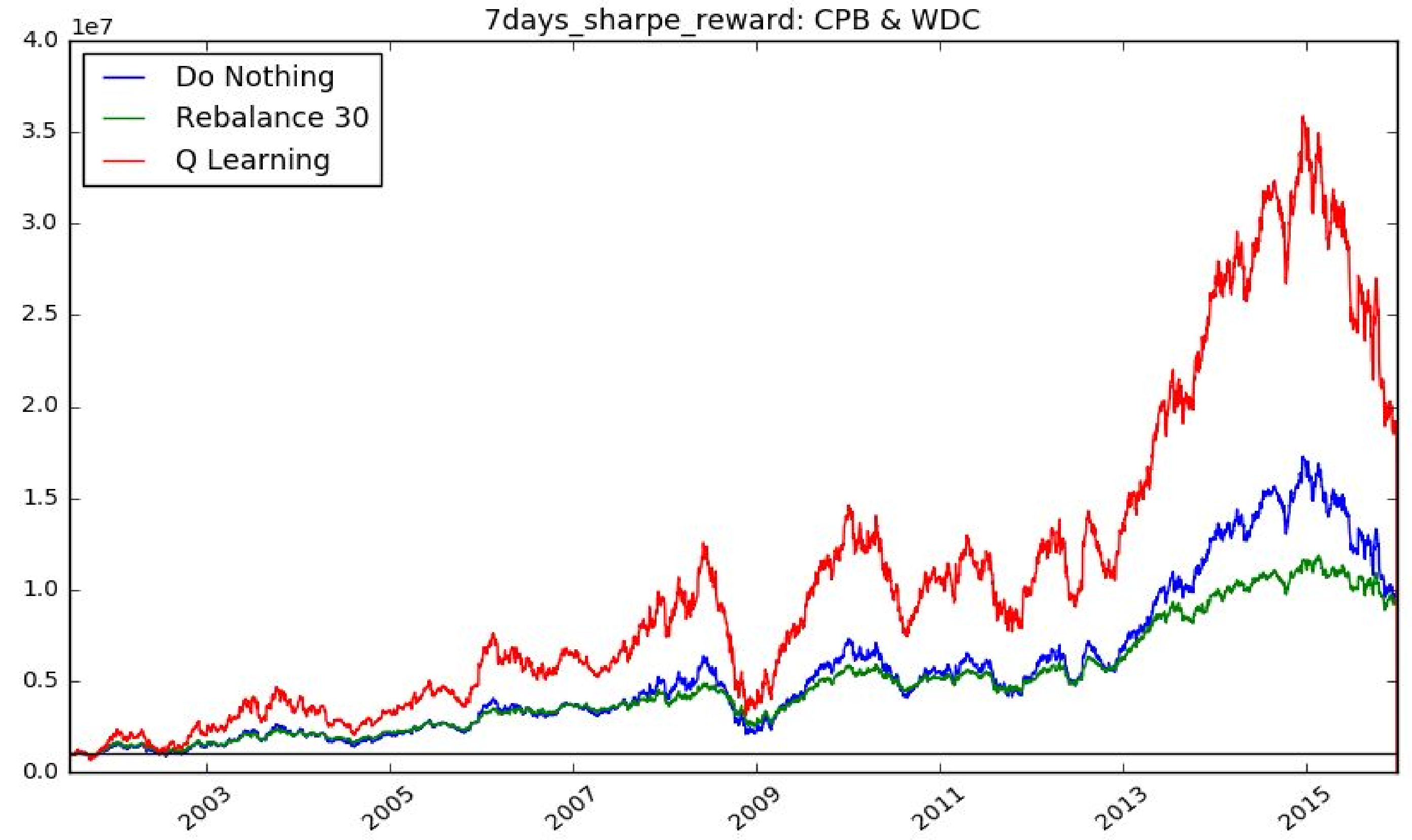

Figure 4: A portfolio for a network feed 7 days of data and using the Sharpe ratio reward