Reinforcement Learning For Automated Trading

Reid Falconer, Sam MacIntyre, Maximilian Zebhauser and Hector Cano

Reinforcement Learning for Automated Trading

Motivation and Executive Summary

- **Trading stocks** is reward maximizing, sequential process of decision making with a high level of uncertainty, the idea would be to automatize this process
- To tackle this problem we extended the dynamic programming approach and used **reinforcement learning** (RL), which is the third machine learning paradigm next to supervised and unsupervised learning
- We trained the trading agent to decide whether **to sell, hold or buy a certain stock** on given data and evaluate its performance by comparing it to a random or a holding trading strategy
- Through our experiments with AAPL and WWL stocks, the trading agent **outperforms** two basic trading strategies (random action & buy and hold) in total portfolio value and is also able to effectively manage volatility
- Surprisingly, the initial budget appeared to be a strong determinant of the behavior of our agent
- There are many possible future developments that could be explored such as using full **portfolio management**, modifying the reward function, performing online-learning and doing further hyper-parameter tuning

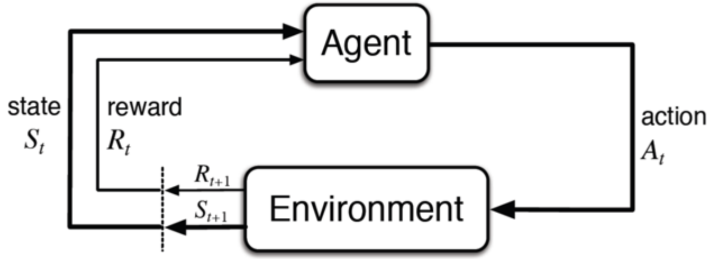
How to train a BOT to trade stocks

Underlying Theory and Methodology

- In RL the agent is trained to develop a **policy** which maximizes the discounted **reward** in a given environment based on its actions taken:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=o}^{\infty} \gamma^k R_{t+k+1}$$

- Generally, RL is modeled as Markov Decision Process:



- The corresponding Bellman equation:

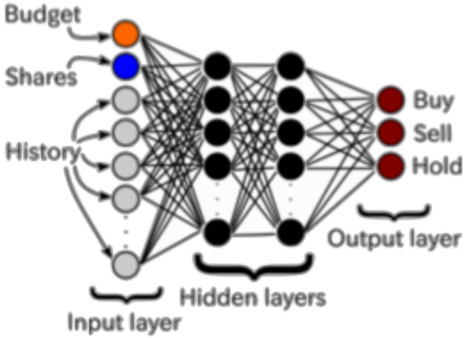
$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t | S_t = s] = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')]$$

- More specifically, we use the paradigm of **Q-learning** which iteratively learns the quality of state action combinations. In our case, we use a neural network to approximate the quality function:

$$\hat{Q}(s,a) = R(s,a) + \gamma \max_{a' \in A} Q(s,a)$$

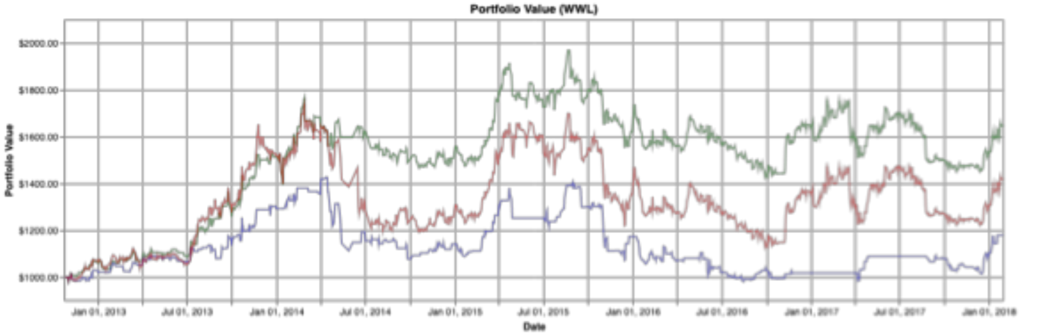
$$Loss = \left\| Q - \hat{Q} \right\|_2$$

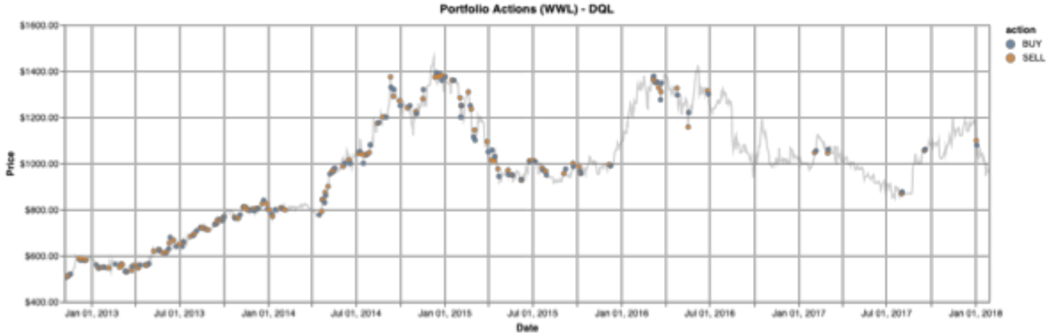
- This is called **deep reinforcement learning:**



Performance of the Trading BOT

Implementation and Evaluation





	AAPL			WLL		
	DQL	BH	RAND	DQL	BH	RAND
Accumulated Return(%)	1.67	1.33	0.59	0.64	0.41	0.18
Sharp Ratio	0.29	0.22	0.26	0.25	0.20	0.09