Forex Trading with Deep Reinforcement Learning

Vincent Wang

September 25, 2018

1 Analysis

The Problem Successful traders are often those that have lots of experience with markets. Over time they gain some intuition or "feel" for how the market will act. That being said however, markets move randomly. Trading, especially forex (foreign exchange) trading, has been likened to gambling because of this - it's risky and very difficult to reliably predict (source 1). Even when a correct prediction is made, margins in forex are very small so turning a profit is difficult, especially when taking into account the broker's fees to carry out the trade.

Solution To approach this problem, we will be using a technique in machine learning called reinforcement learning - letting an agent learn what actions are good/bad to take through trial and error in its environment. This has been used widely over the past few years to solve problems such as learning to play computer games as well as being used for applications such as market trading. For complex problems such as those stated above, reinforcement learning is more effective and easier to do than hard coding a strategy.

Requirements The completed solution should be a trained agent that is able to simulate trade between two or three currency pairs *specify*, being able to consistently turn a profit of at least *specify percentage* over a *given time frame*. It should be able to simulate trades in real time, using live forex market data. The program will start with *specify amount* of gold in each currency, and the program's portfolio will be expressed in gold at each point. This has been chosen as gold can act as a stable, cross-currency baseline, allowing us to evaluate and compare performance effectively.

2 Initial Thoughts on Design

- Use a DQN?
  - Tensorflow?
  - Pytorch? (possibly more intuitive, trains much faster?)
- Use Alpha Vantage for both training data for live data
  - ha ha ha

3 Short Term Objectives

Continue top-down research:

- Learn more about markets - think about how network ins and outs could be designed
- Learn more about DQNs

- Consider frameworks
  - Tensorflow vs Pytorch
  - Implement an OpenAI gym DQN in te chosen one for familiarity
- Another entry in the list