Machine Learning Engineer Nanodegree

Capstone Proposal

Abdel Affo

November 16th, 2018

Creating an automated trading agent using deep reinforcement learning

Domain Background

Algorithmic trading has grown in popularity the past few years, with an estimated 70% of US trade volume being generated through this type of trading. I've been interested in stock trading and technical analysis for a long time. Technical analysis is the belief that future stock prices can be inferred from past trading history. With my newly acquired knowledge of reinforcement learning, I believe that the stock trading problem can be formulated as a Markov Decision Process problem, with the market being the environment, and the trader or trading system being the agent.

Problem Statement

The goal of this project is to investigate if a reinforcement-learning agent can be trained to trade a given stock successfully. The agent will be trained on historical prices, and then will decide what decision to make in order to maximize profit.

Datasets and Inputs

The dataset used in this project is a time series dataset provided by Quandl, free to download and use. The data set format is as follows:

Input Data fields

- Open – Opening price
- High – High price of the day
- Low – Low price of the day

- Close – Closing price
- Volume – Daily volume
- Ex-Dividend – Dollar amount of cash dividend
- Split Ratio – Ratio of the number of new shares to the number of old shares
- Adj. Open – Adjusted opening price
- Adj. High – Adjusted high price
- Adj. Low – Adjusted low price
- Adj. Close – Adjusted closing price
- Adj. Volume – Adjusted volume

Adjusted prices are back-adjusted assuming that all corporate actions are reinvested into the current stock.

The agent will be operating on a single stock, Altria Group (MO) and the dates will be from January 1st 2015 to February 28th 2017.

This data can be accessed at:

- https://www.quandl.com/databases/WIKIP/WIKI-PRICES/export

The data set will be handled as a time series set, and will be split into a training set and a test set. The training data interval will be from January 1st 2015 to December 31st 2016, and the testing interval will be from January 1st 2017 to February 28th 2017, to ensure that future data isn't used during training. There will be 504 data points in the training set, and 39 data points in the test set.

Solution Statement

I will be using a Deep Reinforcement model to approach this problem. Trading can be formulated as a Markov Decision Process problem by defining the right environment, states, actions and rewards. The model chosen is a Deep Q-Network (DQN) model, which can be used to approximate the optimal policy using neural networks.

Benchmark Model

Two benchmark models will be used in this project. Because it's a reinforcement-learning problem the agent will be compared to an agent taking random actions at each time step. Because our problem is also a trading problem, we'll also be comparing it to an agent applying a buy and hold strategy, which buys the stock at the initial time step, and hold the position the entire time.

Evaluation Metrics

The criterion use to evaluate the models' performance will be the profit made by trading the stock over the entire period of time.

Project Design

Pre-processing

The first step in the project will be to analyze the data provided in order to remove unneeded features, and create additional features that will help achieve more accurate predictions. The test set will be used to evaluate the performance of our trained agent. The next step is to create an environment that the agent can interact with, using the time series data to define the environment's states.

Environment

Each row of the time series dataset will be a considered a state of the environment. The agent can take a single action each day, with the daily profit or loss computed following the action. The possible actions that the agent can take are to buy a position using all the available funds, hold that position, or sell the entire position. The daily profit or loss will be used as the reward, with the goal of maximizing the total profit.

Model

The model created for this project will be using the Double DQN (Deep Q Network) approach. Using this approach, two DQN models are defined to decouple the action selection from the target Q-value generation. The first model is used to select what the best action to take for the next state is. The second model, called the target network, is used to compute the target Q value resulting from the action taken. The Double DQN approach helps reduce the overestimation of Q-values and achieve faster convergence. The two networks will have the same architectures, with their hyperparameters selected and tuned in order to achieve stable learning.

References:

Zhengyao Jiang, Dixing Xu, Jinjun Liang. A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. arXiv:1706.10059, (2017)

Saud Almahdi, Steve Yang. An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. Expert Systems with

Applications. 87. 10.1016/j.eswa.2017.06.023. (2017)

Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang (Bruce)Yang, Anwar Walid .
Practical Deep Reinforcement Learning Approach for Stock Trading. (2018)