Deep Reinforcement Learning applied to trading

Thesis submitted at the Polytechnic University of Madrid
in partial fulfilment of the requirements for
the degree of Master in Signal Theory and Communications

by

Borja Gómez Solórzano
Quant at Darwinex

Advisor: Eduardo López

July 2018

Contents

Abstract

The goal of the paper is to present an authomatic trading strategy over a single asset based on a reinforcement learning approach. I follow Direct Reinforcement Learning to solve it, the difference between DRL and the commonly Q-Learning is the learning phase, the first tries to train the model optimizing the inmediate rewards and the second the long term rewards. In the case of price series DRL is more suitable, cause the high noise and the continuity in the signal. The target is to maximize an utility function, in this paper I will introduce some of them, like Sharpe Ratio. The trader's action is done by a decision function, this function is the output of a neural network with a deep architecture, the input of the network is the price series and some side information like other price series and indicators, this side information is needed for filtering the trading actions over the main asset traded. In this implementation I will have to face, in the training phase, with some problems of the backpropagation algorithm like named vanishing gradient, to solve this problem I implement a solution based on LSTM networks. Finally, I include graphics to show the execution of the algorithm over the time, the most important are: The equity curve, the trader's action and the evolution of the utility function. I will add also some trading based statistics to measure the performance of the strategy like drawdowns, p/l.

Introduction

Traditional trading algorithms

Traditionally, trading algortithms are build using technical analysis techniques based on asset price indicators like moving averages or oscillators like RSI and fixed parameters. This parameters are selected running different backtest over the history and are very sensitive to overfitting. The principal drawback of this approach is the difficult to adapt to different market regimes, Reinforcement Learning techniques have the capability to learn from interactions with the environment.

Deep Reinforcement Learning

Q-trader

Chapter 1

Basic concepts

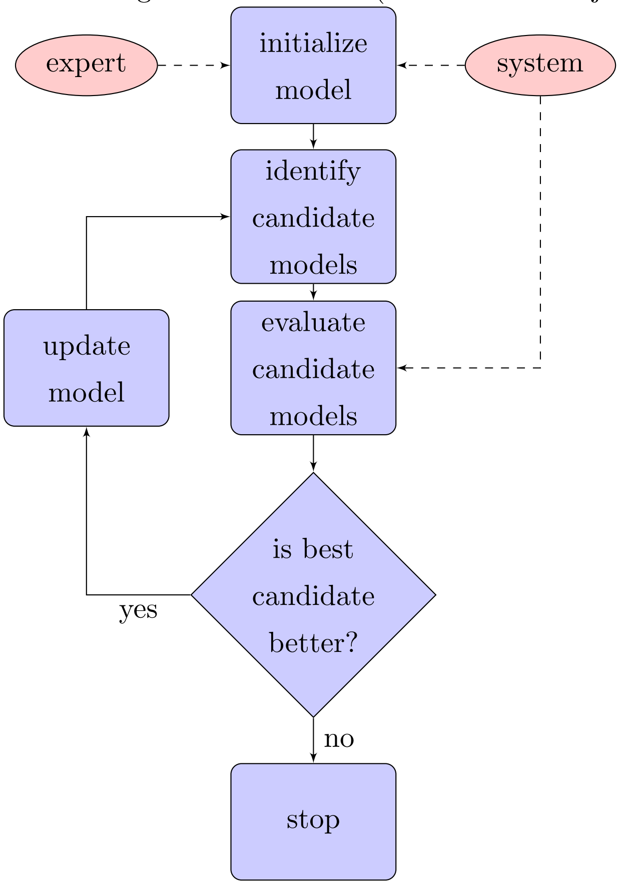
Chapter 2

Direct Reinforcement Learning

[1]

2.1 The model

Aquí va el diagrama del modelo (esto solo es un ejemplo, lo tengo que adaptar)


expert
initialize model
system
identify candidate models
update model
evaluate candidate models
is best candidate better?
yes
no
stop

Algorithm 2.1 Direct Reinforcement

```
1: **Input:** network architecture params
2: **Output:** $\pi_\theta$ policy, u=max utility
3: for (for each episode) do
4:     Initialize $\theta$ randomly
5:     for (for each step in the episode) do
6:         take action $a'$ $\pi_\theta(.|s)$ and observe state $s'$ and $r$
7:         $r = reward(s', a, a')$
8:         $a = a'$
9:     end for
10:    $u(s, a; \theta) = utility(rewards)$
11:    $\pi_\theta = \text{argmax}_{a' \in A}\, u(s, a; \theta)$
12: end for
13: return $u, \pi_\theta$
```

2.2 Neural network trader agent

I_t is all the information available at time t, including the prices and external information history. $F_t = F(\theta; F_{t-1}, I_t) \in [-1, 1]$ is the trader action (the percentage of asset bought), we supose in this first approach presented that the asset is infinitelly divisible, we use for this the tanh function. $U_t = U(R_1, R_2, ..., R_T)$ is the utility function that measures the goodness of the actions the trader takes and the one that we want to maximize. The learning algorithm is based on the gradient ascent with updating formula:

$$\frac{dU_t(\theta)}{d\theta} = \sum_{t=1}^{T} \frac{dU_t}{dR_t} \left\{ \frac{dR_t}{dF_t} \frac{dF_t}{d\theta} + \frac{dR_t}{d\theta} \frac{dF_t}{d\theta} \right\} \tag{2.1}$$

which is path dependent.

2.3 Utility function: Sharpe ratio

The rewards are computed as

$$R_t = F_{t-1} r_t - \delta |F_t - F_{t-1}| \tag{2.2}$$

where δ are the commissions per order and r_t the asset returns between times t and t-1. We define the following performance measure

$$S = \frac{\mu(R_t)}{\sigma(R_t)} \tag{2.3}$$

(where μ is the average and σ is the standard deviation) known as Sharpe ratio, which is a risk-adjusted measure of the strategy.

Chapter 3

LSTM network

[4]

3.1 Vanishing gradient

Chapter 4

Improvements

4.1 Side information

4.2 New utility functions: Downside deviation ratio

[2]

Chapter 5

Experiments

5.1 Performance measures

5.2 Results

Chapter 6

Conclusions

Appendix 1.

Bibliography

[1] John Moody and Matthew Saffell, Reinforcement Learning for Trading, *Advances in neural information processing systems · July 1999*

[2] John Moody, Lizhong Wu, Yuansong Liao & Matthew Saffell, Performance functions and Reinforcement Learning for trading systems and portfolios*, Journal of Forecasting, Volume 17, Pages 441-470, 1998.*

[3] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai, Deep Direct Reinforcement Learning for Financial Signal Representation and Trading, *IEEE Transactions on neural networks and learning systems*, *2016.*

[4] David W. Lu, Agent Inspired Trading Using Recurrent Reinforcement Learning and LSTM Neural Networks, *arXiv:1707.07338 [q-fin.CP]. July 2017.*