

ACKNOWLEDGEMENT

First and foremost, we would like to sincerely thank our advisor and mentor Prof. H.C. Taneja for the continuous support during our research and study, for his enthusiasm, motivation and for the freedom we were granted throughout the preparation of the major project. He offered us useful advice whenever we asked him and also guided us towards our goal whenever we needed help. His wisdom and experience helped us tremendously towards achieving our goals.

Our sincere thanks also goes to the Department of Applied Mathematics for their support, guidance, and encouragement from the research point of view as well as giving us a platform so that we could perform to the best of our abilities and learn new, advanced techniques.

Deepest gratitude are also due to all members of the panel for their interest in this work and for taking the time to evaluate this project report.

We would also like to thank our families for the constant motivation and encouragement throughout this project timeline. Without their unwavering encouragement, our path towards the completion of the project would have been significantly more difficult.

ADITYA PAREEK

2K15/MC/006

SHIKHAR BHARDWAJ

2K15/MC/078

TUSHAR PRASAD

2K15/MC/093

CONTENTS

Acknowledgement	iii
Abstract	v
Contents	vii
List of Figures	x
CHAPTER 1 OUTLINE OF THE REPORT	1
CHAPTER 2 INTRODUCTION	3
2.1 Motivation and Scope	5
2.2 Financial Markets	7
2.2.1 Price Chart	8
2.2.2 Trade History	8
2.2.3 Order Book	8
CHAPTER 3 BASICS ON REINFORCEMENT LEARNING	9
3.1 Introduction	9
3.1.1 Learning to Optimize Rewards	9
3.2 Policy Search	10
3.2.1 Policy Gradients	12
3.3 Markov Decision Processes	13
3.3.1 Markov Chains	13
3.3.2 Markov Decision Process	15

3.4	Temporal Difference Learning	17
3.4.1	Q-Learning Algorithm	18
3.4.2	State-Action-Reward-State-Action Algorithm	19
3.4.3	Exploration Policies	20
CHAPTER 4	DEEP REINFORCEMENT LEARNING	20
4.1	Introduction	20
4.2	Deep Reinforcement Learning	22
4.2.1	Problems	24
4.2.2	Solutions	26
4.2.3	Implementation Details	27
4.3	Double Deep Q-Networks	28
4.4	Deep Recurrent Q-Networks	31
4.4.1	Recurrent Neural Networks	32
4.4.2	Long Short Term Memory	34
4.4.3	DRQN	36
CHAPTER 5	EXPERIMENTAL IMPLEMENTATION	38
5.1	Data	38
5.1.1	Source	38
5.1.2	Pre-processing	40
5.2	Environment	41
5.2.1	Action Space	41
5.2.2	State Space	41

5.2.3	Reward	43
5.3	Agent	44
5.4	Model Architecture and Training	45
5.4.1	Weight/Parameter Initialisation	45
5.4.2	Training Scheme	45
5.4.3	Complete Learning Algorithm	46
5.4.4	Hyper-parameters	47
5.5	Model Evaluation and Testing	48
APPENDIX-A		50

LIST OF FIGURES

USD vs BTC	5
Price Chart	5
Order Book	6
Reinforcement learning system	8
RL components	9
Policy Search	10
Markov Chains	11
MDP example	12
Policy search illustration	13
Discretization of space	21
Inside a neuron	22
Deep neural network architecture	22
Deep Q-Network	23
Huber loss	28
DRQN	31
RNN vs ANN	32
RNN rolled	33
RNN unrolled	33
LSTM – small time gap	34
LSTM – big time gap	35
Inside RNN	36
Inside LSTM	36
Comparison of DQN and DRQN	37
Forex Data	39
Resampled Forex Data	40
Model Architecture	45
Code output	48
Code output curve	49
OpenAI gym	51
Backtrader	51
BT gym	52

ABSTRACT

An automated program that can provide high profit from the financial market is attractive to every market practitioner. The task of financial trading can be described as an agent that interacts with the market and try to achieve some inherent intrinsic goal. It is not required for the agent needs to be human as in modern financial markets, algorithmic trading accounts for high volume of trading activities.

Conventionally, success in financial markets is defined as the degree of closeness of the agent to its intrinsic goal. One of the most fundamental hypotheses of reinforcement learning is that goals of an agent can be expressed through maximizing long-term future rewards. Reward is a single scalar feedback signal that reflects the goodness of an agent's action in some state. This is called the reward hypothesis which states that *"All goals can be described by maximization of expected future rewards"*.

We try to model such a trading agent leveraging the recent advances in deep reinforcement learning. A Partially Observable Markov Decision Process is proven to be suitable for financial trading in general and is modelled and solved with the help of state-of-the-art Deep Recurrent Q-Learning (DRQN) Algorithm. The work is inspired from the paper: "Financial Trading as a Game: A Deep Reinforcement Learning Approach" by Chien-Yi Huang [2018]. Several modifications to the existing learning algorithm are implemented, that make it more suited for financial trading task –

1. A significantly small replay memory is employed compared to the ones used in modern Deep Reinforcement learning algorithms is implemented.
2. An action augmentation technique is implemented that mitigates the need for random exploration by giving extra reward feedback signals for all possible actions in a

particular state to the agent. This enables the model to use greedy exploration policy instead of the commonly used ϵ -greedy approach. However, this technique can only be used under certain market assumptions.

3. A longer sequence is sampled for the training of recurrent neural network. As a result, we now can train the agent every T steps, which greatly reduces the time required to train the model as the overall computation required is brought down by a factor of T .

All the mentioned points are combined into an online learning algorithm and is evaluated on the spot, on the forex market.

1. We use a Partially Observable Markov Decision Process (PO-MDP) model for the above mentioned task that is solvable by state-of-the-art deep reinforcement learning algorithm with the use of freely available data.
2. We modify the existing learning algorithm and then implement it, that make it more suited for financial trading task. This involves using a significantly small replay memory and sampling a longer sequence for training. We also employ a novel action augmentation technique to mitigate the need for random exploration in the financial trading environment.