Stock Prediction using Reinforcement Learning Algorithms

Project Update Report

A picture containing window, drawing, light

Description automatically generated

By

Gulnara Timokhina

Mirsaeid Abolghasemi

Poornapragna Vadiraj

Varun Bhaseen

Date: November 10, 2020

Table of Contents

[1 Introduction 3](#_Toc56111966)

[2 Project Team 3](#_Toc56111967)

[3 Project Plan Snapshot from Trello 3](#_Toc56111968)

[4 Project Plan and Progress Status 4](#_Toc56111969)

[5 Effort Distribution Across Resources 6](#_Toc56111970)

[6 Project Timeline 7](#_Toc56111971)

# Introduction

The Primary purpose of the project is to create a Stock prediction model using multiple Reinforcement Learning Algorithms. We will be using DQN (Deep Q Network) as a baseline model for stock prediction which will be followed up by Actor-Critic, TRPO (Trust Region Policy Optimization), and PPO (Proximal Policy Optimization).

# Project Team

Entire project is managed through Trello Board ([Access Link to Trello Board](https://trello.com/invite/b/8nGWdcKI/bfdb6be2b48485cf57b5df6297e83859/reinforcement-learning-project)). Following are the team members for the project:

1. Gulnara Timokhina (Lead)
2. Mirsaeid Abolghasemi
3. Poornapragna Vadiraj
4. Varun Bhaseen

# Project Plan Snapshot from Trello

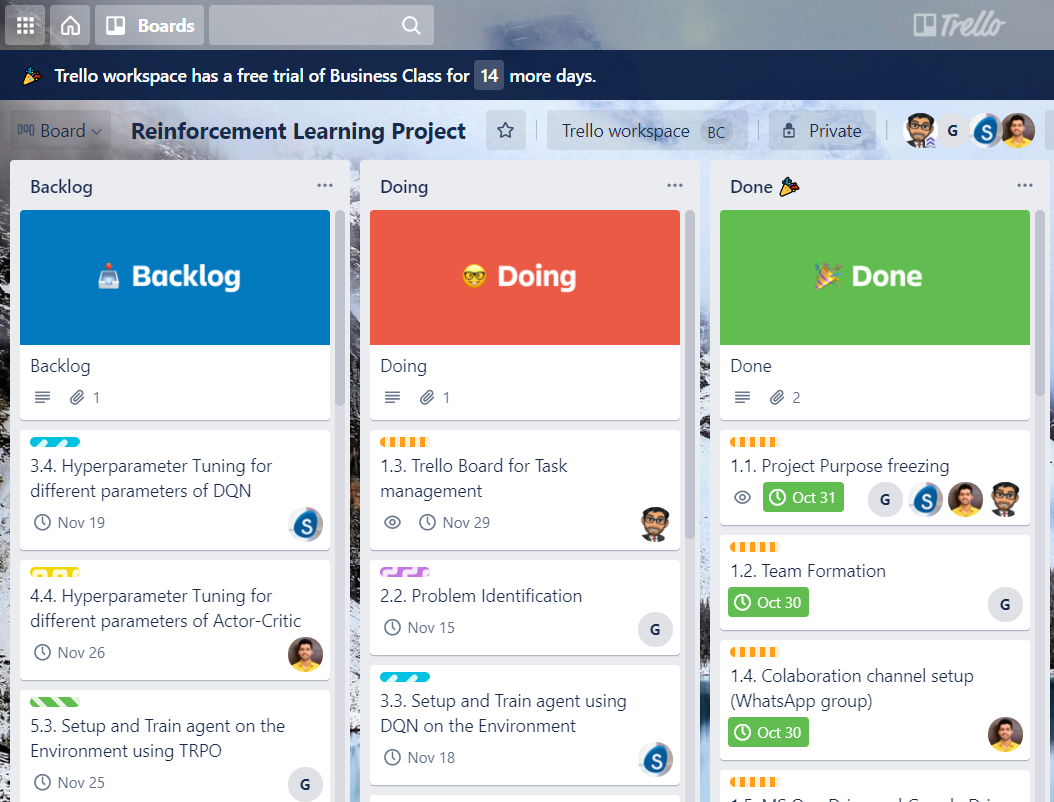


Figure : Trello Board List View (Source: Trello Board for Reinforcement Learning Project)

# Project Plan and Progress Status

| Milestone Description | Assigned To | Start | Status |
| --- | --- | --- | --- |
| 1. Handshake and Collaboration Setup |  | 10/24/2020 |  |
| 1.1. Project Purpose freezing | Team | 10/30/2020 | Done |
| 1.2. Team Formation | Gulnara Timokhina | 10/24/2020 | Done |
| 1.3. Trello Board for Task management | Varun Bhaseen | 10/27/2020 | Doing |
| 1.4. Colaboration channel setup (WhatsApp group) | Poornapragna Vadiraj | 10/27/2020 | Done |
| 1.5. MS One Drive and Google Drive setup | Mirsaeid Abolghasemi | 11/2/2020 | Done |
| 2. Project Kick-off |  | 10/27/2020 |  |
| 2.1. Literature Research | Mirsaeid Abolghasemi | 11/5/2020 | Done |
| 2.2. Problem Identification | Gulnara Timokhina | 11/1/2020 | Doing |
| 2.3. Project Scoping | Poornapragna Vadiraj | 11/2/2020 | Done |
| 2.4. Abstract Draft | Varun Bhaseen | 10/27/2020 | Done |
| 2.5. Abstract Submission | Team | 10/30/2020 | Done |
| 3. DQN Baseline for Stock Prediction |  | 11/5/2020 |  |
| 3.1. Setup the Development and Testing Environment Locally for DQN | Mirsaeid Abolghasemi | 11/7/2020 | Done |
| 3.1.1. Setup Anaconda environment with all required libraries for Deep Learning (PyTorch, Tensorboard) | Mirsaeid Abolghasemi | 11/5/2020 | Done |
| 3.1.2. Install the Open AI Gym and all dependencies | Mirsaeid Abolghasemi | 11/7/2020 | Done |
| 3.2. Identify the Target Label in the Open AI Gym environment (StockTradingENV: GOOGL) for DQN | Mirsaeid Abolghasemi | 11/12/2020 | Done |
| 3.3. Setup and Train agent using DQN on the Environment | Mirsaeid Abolghasemi | 11/13/2020 | Doing |
| 3.4. Hyperparameter Tuning for different parameters of DQN | Mirsaeid Abolghasemi | 11/14/2020 | Backlog |
| 3.5. Record DQN results in Tensorboardx and publish link to teammates with logs | Mirsaeid Abolghasemi | 11/13/2020 | Doing |
| 4. Using Actor-Critic for Stock Prediction |  | 11/7/2020 |  |
| 4.1. Setup the Development and Testing Environment Locally fro Actor Critic | Poornapragna Vadiraj | 11/7/2020 | Done |
| 4.1.1. Setup Anaconda environment with all required libraries for Deep Learning (PyTorch, Tensorboard) | Poornapragna Vadiraj | 11/15/2020 | Done |
| 4.1.2. Install the Open AI Gym and all dependencies | Poornapragna Vadiraj | 11/13/2020 | Done |
| 4.2. Identify the Target Label in the environment (StockTradingENV: GOOGL) for Actor Critic | Poornapragna Vadiraj | 11/15/2020 | Done |
| 4.3. Setup and Train agent on the Environment using Actor-Critic | Poornapragna Vadiraj | 11/20/2020 | Doing |
| 4.4. Hyperparameter Tuning for different parameters of Actor Critic | Poornapragna Vadiraj | 11/21/2020 | Backlog |
| 4.5. Record results of Actor Critic in Tensorboardx and publish link to teammates with logs | Poornapragna Vadiraj | 11/22/2020 | Doing |
| 5. Using TRPO (Trust Region Policy Optimization) for Stock Prediction |  | 11/21/2020 |  |
| 5.1. Setup the Development and Testing Environment Locally for TRPO | Gulnara Timokhina | 11/7/2020 | Done |
| 5.1.1. Setup Anaconda environment with all required libraries for Deep Learning (PyTorch, Tensorboard) | Gulnara Timokhina | 11/15/2020 | Done |
| 5.1.2. Install the Open AI Gym and all dependencies | Gulnara Timokhina | 11/13/2020 | Done |
| 5.2. Identify the Target Label in the environment (StockTradingENV: GOOGL) for TRPO | Gulnara Timokhina | 11/15/2020 | Done |
| 5.3. Setup and Train agent on the Environment using TRPO | Gulnara Timokhina | 11/20/2020 | Backlog |
| 5.4. Hyperparameter Tuning for different parameters of TRPO | Gulnara Timokhina | 11/21/2020 | Backlog |
| 5.5. Record results of TRPO in Tensorboardx and publish link to teammates with logs | Gulnara Timokhina | 11/22/2020 | Backlog |
| 6. Using PPO (Proximal Policy Optimization) for Stock Prediction |  |  |  |
| 6.1. Setup the Development and Testing Environment Locally for PPO | Varun Bhaseen | 11/7/2020 | Done |
| 6.1.1. Setup Anaconda environment with all required libraries for Deep Learning (PyTorch, Tensorboard) | Varun Bhaseen | 11/15/2020 | Done |
| 6.1.2. Install the Open AI Gym and all dependencies | Varun Bhaseen | 11/13/2020 | Done |
| 6.2. Identify the Target Label in the environment (StockTradingENV: GOOGL) for PPO | Varun Bhaseen | 11/15/2020 | Doing |
| 6.3. Setup and Train agent on the Environment using PPO | Varun Bhaseen | 11/20/2020 | Backlog |
| 6.4. Hyperparameter Tuning for different parameters of PPO | Varun Bhaseen | 11/21/2020 | Backlog |
| 6.5. Record results of PPO in Tensorboardx and publish link to teammates with logs | Varun Bhaseen | 11/22/2020 | Backlog |
| 7. Performance Comparision of Agents |  |  |  |
| 7.1. Compare the accuracy of each technique for the trained agents | Team | 11/27/2020 | Backlog |
| 7.2. Time to convergence to most optimal prediction by each agent for each technique | Team | 12/2/2020 | Backlog |
| 7.3. Performance Evaluation for each Hypertuning instance for all agents | Team | 12/2/2020 | Backlog |
| 7.4. Develop and publish interactive visualization for each results and draw conculsions | Team | 12/7/2020 | Backlog |
| 8. Project presentation and closure | Team | 12/7/2020 | Backlog |

# Effort Distribution Across Resources

| **Project Implementation Plan** | **Effort** |
| --- | --- |
| **Gulnara Timokhina** | **36** |
| 1.2. Team Formation | 6 |
| 2.2. Problem Identification | 4 |
| 5.1. Setup the Development and Testing Environment Locally for TRPO | 2 |
| 5.1.1. Setup Anaconda environment with all required libraries for Deep Learning (PyTorch, Tensorboard) | 3 |
| 5.1.2. Install the Open AI Gym and all dependencies | 2 |
| 5.2. Identify the Target Label in the environment (StockTradingENV: GOOGL) for TRPO | 2 |
| 5.3. Setup and Train agent on the Environment using TRPO | 5 |
| 5.4. Hyperparameter Tuning for different parameters of TRPO | 5 |
| 5.5. Record results of TRPO in Tensorboardx and publish link to teammates with logs | 7 |
| **Mirsaeid Abolghasemi** | **33** |
| 1.5. MS One Drive and Google Drive setup | 1 |
| 2.1. Literature Research | 6 |
| 3.1. Setup the Development and Testing Environment Locally for DQN | 2 |
| 3.1.1. Setup Anaconda environment with all required libraries for Deep Learning (PyTorch, Tensorboard) | 3 |
| 3.1.2. Install the Open AI Gym and all dependencies | 2 |
| 3.2. Identify the Target Label in the Open AI Gym environment (StockTradingENV: GOOGL) for DQN | 2 |
| 3.3. Setup and Train agent using DQN on the Environment | 5 |
| 3.4. Hyperparameter Tuning for different parameters of DQN | 5 |
| 3.5. Record DQN results in Tensorboardx and publish link to teammates with logs | 7 |
| **Poornapragna Vadiraj** | **32** |
| 1.4. Colaboration channel setup (WhatsApp group) | 3 |
| 2.3. Project Scoping | 3 |
| 4.1. Setup the Development and Testing Environment Locally fro Actor Critic | 2 |
| 4.1.1. Setup Anaconda environment with all required libraries for Deep Learning (PyTorch, Tensorboard) | 3 |
| 4.1.2. Install the Open AI Gym and all dependencies | 2 |
| 4.2. Identify the Target Label in the environment (StockTradingENV: GOOGL) for Actor Critic | 2 |
| 4.3. Setup and Train agent on the Environment using Actor-Critic | 5 |
| 4.4. Hyperparameter Tuning for different parameters of Actor Critic | 5 |
| 4.5. Record results of Actor Critic in Tensorboardx and publish link to teammates with logs | 7 |
| **Team** | **20** |
| 1.1. Project Purpose freezing | 2 |
| 2.5. Abstract Submission | 1 |
| 7.1. Compare the accuracy of each technique for the trained agents | 2 |
| 7.2. Time to convergence to most optimal prediction by each agent for each technique | 5 |
| 7.3. Performance Evaluation for each Hypertuning instance for all agents | 6 |
| 7.4. Develop and publish interactive visualization for each results and draw conculsions | 2 |
| 8. Project presentation and closure | 2 |
| **Varun Bhaseen** | **32** |
| 1.3. Trello Board for Task management | 2 |
| 2.4. Abstract Draft | 4 |
| 6.1. Setup the Development and Testing Environment Locally for PPO | 2 |
| 6.1.1. Setup Anaconda environment with all required libraries for Deep Learning (PyTorch, Tensorboard) | 3 |
| 6.1.2. Install the Open AI Gym and all dependencies | 2 |
| 6.2. Identify the Target Label in the environment (StockTradingENV: GOOGL) for PPO | 2 |
| 6.3. Setup and Train agent on the Environment using PPO | 5 |
| 6.4. Hyperparameter Tuning for different parameters of PPO | 5 |
| 6.5. Record results of PPO in Tensorboardx and publish link to teammates with logs | 7 |

# Project Timeline

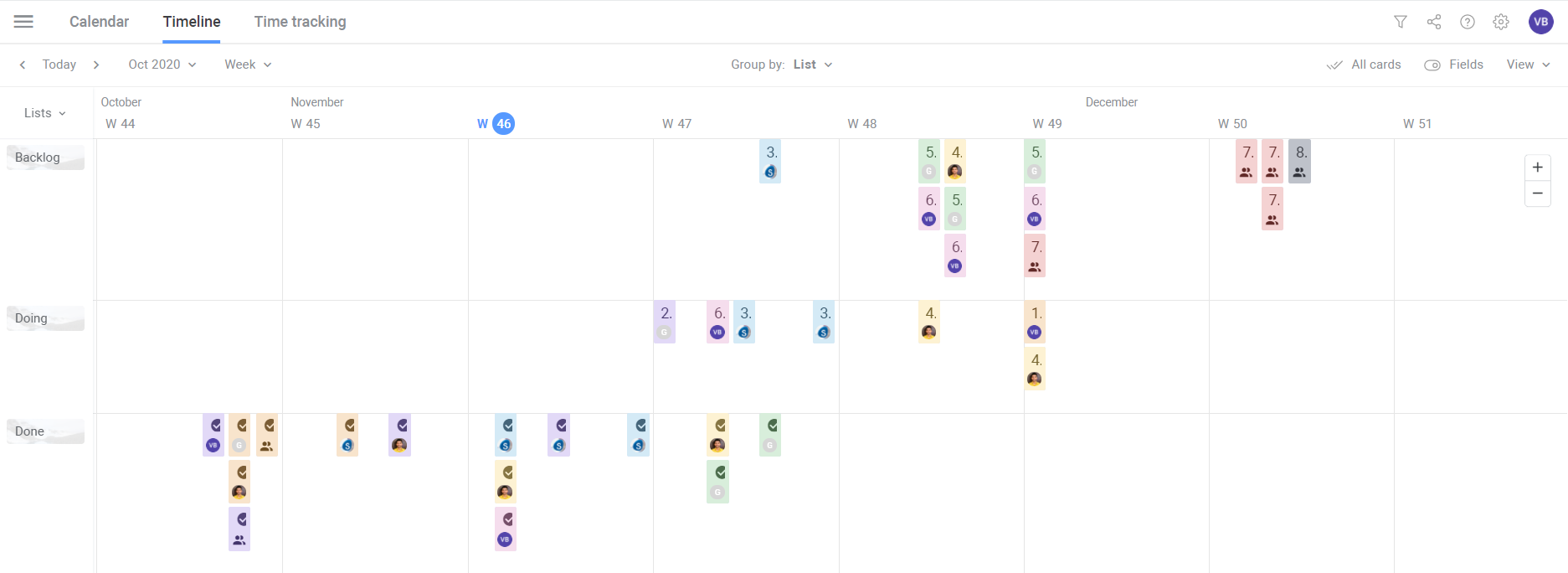


Figure : Trello Project Timeline Report (Source: Trello board for Reinforcement Learning Project)

\*\*\* End of Document\*\*\*