Final Project Proposal

Project Information:

Project Title: Deep Reinforcement Learning Trading Agent
Team Members: Diego Sanchez-Carapia
Selected Project: Deep Reinforcement Learning Trading Agent (8)
Date Submitted: November 10, 2025
Course: Computational Investing [CSCI 4170]

‘

Project Overview:

This project addresses the challenge of creating an intelligent trading agent that can learn how to make the best trading decisions on its own by interacting with financial markets. The biggest and main challenge is designing an agent that can handle the complex and high dimensional data that we get from the markets, since basic reinforcement learning methods struggle with scaling. This work matters because algorithmic trading is a multi-billion-dollar industry where even the smallest improvements can lead to massive profit gains as shown in high-frequency trading where microseconds count, where server locations matter, where you need custom networking to compete.

I plan to build a Deep Q-Network (DQN) trading agent using Pytorch. The agent will be able to learn to buy, sell, or hold stocks based on historical market data and technical indicators. Also, instead of using Q-tables. I'll implement a neural network to predict the best actions. In order to make training more stable, I'll use experience replay and a separate target network. Also the trading system will run in a custom Open AI Gym trading environment. It will have realistic constraints such as transaction costs and position limits. The input data will combine prices and technical indicators such as RSI, MACD, moving averages and Bollinger Bands. The reward function will be designed around risk-adjusted performance to maximize Sharpe ratio.

In the end I expect to have a working DQN trading agent that can learn profitable strategies from historical data (maybe outperform the market). I'll compare the results to traditional Q-learning models and create visualizations showing how the agent performed and behaved with different marker conditions. This project aligns directly with the course objectives by building on Project 3's Q'Learning and expanding them into deep reinforcement learning. It will demonstrate and show the advantages of using deep learning for realistic trading scenarios.

Milestones and Schedule:

Week	Dates	Milestone
Week 1	Nov 11- Nov 17	Complete HW 9 and 10 (Project 3)
Week 2	Nov 18 - Nov 24	Adapt Project 3 Q-learner for trading, implement DQN with experience replay
Week 3	Nov 25 - Dec 1	Train agents on the data, create visualizations, optimize learning rate.
Week 4	Dec 2 - Dec 4	Test agent on data, create visualizations, write report and record presentation video
Week 5	Dec 5	Final Presentation

Proposed Evaluation Rubric:

Criteria	Points	Description
DQN Implementation & Code quality	30	Neural Network, experience play buffer, clean PyTorch Code
Trading Environment Design	5	Realistic Constraints such as Transaction costs, position sizing
Learning	10	Evidence of Learning
Visualizations, Insights and Comparative Analysis	25	Trading behavior visualization, action distributions, performance across market conditions, clear plots
Total	70	

Effort Justification:

- What makes this project more substantial than a typical assignment?
  - This project takes about 1.5 more effort than a normal project because it combines several complex systems rather than focusing on only one. Such as Q-learning trader from project 3, A Deep Q-Network with neural networks and, a Testing framework to compare results.
  - The DQN on its own involves experience replay, target networks, epsilon-greedy exploration, and training loops which are all difficult on its own. I'm also creating a custom Open AI Gym environment with realistic trading rules.
- What new skills or techniques will you need to learn?
  - I'll need to learn PyTorch for neural networks, learn experience play and target networks, learn to design Gym environments, and calculate technical indicators like RSI and MACD. I will also evaluate results with financial metrics.
- What are the most challenging aspects?
  - Hyperparameter tuning, preventing overfitting
  - implementing reinforcement learning as it's very new to me
  - Implementing stable training with neural networks in reinforcement learning

Resources and Data:

- What datasets will you use? (Include sources or links)
  - yfinance: SPY and other individual stocks between 2015-2024
  - Possibly given CSV files
  - Source: yf.download('SPY', start='2015-01-01', end='2024-12-31')
- What libraries/tools are required?:
  - PyTorch, Open AI Gym, stable-baseline3, pandas, numpy, scikit-learn, matplotlib, seaborn,
- Are there any papers or references you plan to follow?
  - Possibly
    - van Hasselt et al. (2016) "Deep Reinforcement Learning with Double Q-learning" - AAAI