# Generating a trading strategy using Reinforcement Learning

Yogesh Konadasula and Samarth Marudheri
*International School of Engineering*, sriyogesh94@gmail.com, samarth.marudheri@insofe.edu.in

***Abstract*** **- In the quest for a robust trading strategy capable of navigating the dynamics of a complex environment, Reinforcement Learning algorithms offer significant advantages over traditional Machine Learning techniques. In this paper, we propose multiple deep learning models capable of predicting signals that capture sentiment and major events that affect the stock prices of companies. Signals from the models along with a point estimate from a Time Series models are used as to build a trading agent using a state of the art Actor Critic Reinforcement Learning model. The approach yields a return of 14% average profit on the training data.**

## 1 Introduction

Approximating the underlying distribution of data sources that have a low signal to noise ratio is a hard problem. Stock data is one such source of data that has an extremely low signal to noise ratio; Using only stock prices to generate trading signals is error prone. Automated Trading Strategies are increasing their dominance in the volume of stocks traded every year. As of 2014 more than 75 percent of the stock traded at NYSE were automated trades. Current trading signals are based on very basic Statistical Arbitrage techniques and can perform well only in High Frequency Trading scenarios. Stock prices heavily depend on the trading behaviour in the markets. Such behaviour is heavily influenced by the news, market sentiment, company disclosures of new developments such as mergers, quarterly financial reports and estimates.

Additionally, using Reinforcement learning, a trading strategy can be built to not just maximize investment over a fixed number of time steps, but a system that has the freedom to explore multiple solutions over a differing number of time steps to maximize the reward signal.

This paper deals with developing a trading strategy using deep learning to understand unstructured sources along with stock prices to generate alphas. We use 8-K SEC filings of companies to capture the effect of major events and forecast stock prices using a Time Series model employing an LSTM network. An additional signal will be added to capture the sentiment of experts and the crowd from the microblogging platform, Stocktwits. Finally, we use the Actor-Critic Deep Reinforcement Learning model to generate an optimal trading policy based on the signals for two companies - General Electric and Apple Inc.

### 1.1 Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning fundamentally differing in its approach toward decision making from Supervised and Unsupervised Learning. Supervised Learning involves training from a set of labeled examples to generalize on unseen data whilst Unsupervised Learning aims to uncover patterns on unlabeled data. RL is a framework to through which an agent learns to make decisions through trial and error from its interactions with the environment. The goal of the agent is to maximize a scalar reward signal by performing actions defined by a policy. An optimal policy is essentially a mapping from states to actions that defines the behaviour of the agent to maximize the reward signal. The three main approaches to RL are described below :

Value Based Models (Critic Only): The goal of value based methods is to estimate the maximum expected future reward or value function for every action given a state. The value function allows the agent to choose between actions for all states thus defining optimal behaviour without learning an explicit policy. The estimate for value function is improved iteratively through the Bellman equation and for multiple trials. Q-learning is the most popular value-based RL algorithm and Deep Q-Learning employs a neural network for the value function approximation to output a vector of Q-values for each state-action pair. The drawbacks of these methods are its ineffectiveness in high dimensional states due to computation explosion and convergence problems owing to random changes in action for small changes in values.

Policy Based Models (Actor Only): Policy-based methods seek to directly learning the mapping between states and action without explicitly learning a value function. The advantage is to iteratively improve the policy through gradient ascent leading to stable policies that can converge faster with fewer parameters to be learnt. The approach is particularly useful for continuous action spaces.

Actor-Critic Models: Actor-Critic methods combine advantages of both methods by employing two agents as the name suggests - the actor which defines the actions of the model (policy-based) and the critic which measures how good the action taken by the actor is (value-based). Both neural networks run in parallel to learn two sets of weights to iteratively improve policy and value functions.

### 1.2 Reinforcement Learning for Trading

A literature survey suggests that most approaches of deep learning toward automated trading have been in the area of Supervised Learning. There are numerous drawbacks to the approach as pointed by Thomas Fischer[1]. The decision of a trader to buy or sell stocks is typically influenced by multiple factors and simply minimizing error on the forecast of stock prices would not align with the goal of an investor. Further, constraints owing to complex dynamics of the environment such as changing liquidity and transaction costs in real time must be taken in to account for an effective prediction. Reinforcement Learning provides the framework \ to map states with uncertainty to actions leading to an effective decision making system that can overcome these limitations. Most existing work to use RL models for trading use policy gradients methods (Actor Only) and Actor-Critic methods have not been explored exhaustively in this domain.

## 2 Architecture

This section outlines the architecture used to build the base models and the Actor-Critic RL model.

### 2.1 Text Analysis of 8-K Filings

One of the signals used to build the state of the RL model are 8-K filings of companies. 8-K filings are often used by investors where a company is required to disclose major events to shareholders such as mergers, acquisitions, change in leadership, bankruptcy, etc. As a result, 8-K filings contain pertinent information that often influences the share price of a company. Information is mostly in the form of text but may include financial statements and data tables. In this analysis, we retain only the textual data for analysis.

8-K filings are scraped from the U.S. Securities and Exchange Commision website for Apple and General Electric using the BeautifulSoup library in Python. The Central Index Key (CIK) for a company is stored from the Wikipedia page for 'List of S&P 500 Companies'. A total of 350 documents was collected for the years from 2002 to 2019 for both companies. The text was pre-processed using the NLTK library to remove stopwords and punctuation as well as lemmatize the words. Each document was padded to a standard length and pre-trained word embeddings are downloaded from the Stanford NLP Glove corpus as a preprocessing step.

The financial data was appended based on the company and timestamp of the 8-K filing as demonstrated in the paper [2] by Stanford NLP. The open and adjusted close values for each company, VIX and S&P 500 index are downloaded through the AlphaVantage API. The moving average for stock price for the year, quarter and month is calculated before and after the report is released normalized by the stock index GSPC. If the normalized change is greater than 1%, the signal is marked as 'up' or 'down', else marked as 'stay'. Additional attributes including the category of the 8-K filing, VIX and GICS Sector are used for the Machine Learning model as suggested by this post[3].

Four Machine Learning models were built - MLP, RNN, CNN and a combination of RNN and CNN layers. Due to imbalance in the target class, the data was oversampled. The numeric data was standardized and categorical data was dummified. The output of the softmax layer of the best model, that is the predicted probabilities for each of the classes for both companies were recorded. The final format of the output probabilities is described in the results section.

### 2.2 Time Series Forecasting

The stock price for each company is predicted using an LSTM model that predicts the adjusted close values taking in to the historical trend of the actual stock prices. Data is used from 1998 to 2019 for the predictions. Each data point is comprised of five previous values (called look back) that predicts the sixth value with a moving window. The model consists of two LSTM and two dropout layers. The output of the model is a day-wise point estimate for the stock prices of General Electric and Apple Inc.

### 2.3 Sentiment Analysis of Stocktwits Data

Market sentiment is an important aspect that has a significant impact on stock prices. To capture sentiment toward a company, we proposed using data from the Stocktwits platform, a micro blogging platform similar to Twitter. The focus on finance on the platform with labeled sentiments by experts in the field would serve as a strong indicator with little noise. We week to incorporate this signal in the next stage of the project as we could collate data for only a few months.

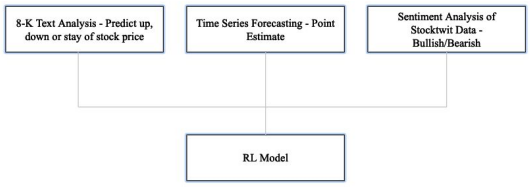


Figure 1: Alphas generated as input to the RL model

## 2.4 Actor Critic RL Model

Model States:The states in a RL model are based on the Markov assumption where each state ideally summarizes the past such that all relevant information is retained[Sutton]. The design of the state is non-trivial which will directly influence the performance of the algorithm. We propose a comprehensive state for the RL model effectively summarizing major events, sentiment and forecasts that directly influence the stock price. The state design for the model comprises of 10 entries as shown below:

[Apple Holdings, GE Holdings, Current Portfolio Value, Cash, Apple's Current Timestep Opening Price, GE's Current Timestep Opening, Apple Five Day Average Stock Price, GE Five Day Average Stock Price, Apple Time Series Forecast, GE Time Series Forecast]

Action Space:Further, the choice of the discrete or continuous action space determines the ease of convergence of the algorithm. To simplify the model and for better convergence, we choose a discrete action space of buy, sell or do nothing. The policy is framed using the official PyTorch code for Actor Critic models.

Episodes:Each episode has a terminal state in the model. An episode ends if the model runs out of money or sold more shares than it currently holds.

Reward Signal: The design of the reward signal is challenging with empirical experiments needed to understand which rewards work best. The reward signal in we use here is the difference between portfolio value before and after the episode.

Starting Cash: For the starting cash for each episode, we sample from a normal distribution with a mean of $1000 and standard deviation of $100.

RL Model: We employ the Actor-Critic (A-C) approach to RL given the hybrid advantages of the model.

# 3 Results

This section describes the results of each base model and the trading strategy output of the RL model.

## 3.1 8-K Filings Predictions

The MLP model yielded the best result of the four Machine Learning models trained. We believe the result is a consequence of relatively less number of data points. The train-test split of the data consisted of a split in the ratio of 70:30 with a total of 350 data points. After training for 10 epochs, the train data accuracy was 87% and 69.9% accuracy on the test data. The predicted probabilities from the softmax layer for the three classes and both companies were further preprocessed to obtain the output format as shown below.

| | Date | AppleDown | AppleStay | AppleUp | GeDown | GeStay | GeUp |
|---|---|---|---|---|---|---|---|
| 15 | 2014-01-16 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.00000 | 0.000000 |
| 16 | 2014-01-17 | 0.0 | 0.0 | 0.0 | 1.327911 | 1.12416 | 1.567204 |
| 17 | 2014-01-18 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.00000 | 0.000000 |
| 18 | 2014-01-19 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.00000 | 0.000000 |
| 19 | 2014-01-20 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.00000 | 0.000000 |
| 20 | 2014-01-21 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.00000 | 0.000000 |
| 21 | 2014-01-22 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.00000 | 0.000000 |
| 22 | 2014-01-23 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.00000 | 0.000000 |

Figure 2 : Sample output of 8-K filings prediction model

As seen from Figure 2, the final output was a sparse matrix with an entry for every day from 2014 to 2018. If an 8K filing was made by either company during that period, the predictions are appended. Additionally a value of 1 is added to all the probability values to differentiate probability from a null value indicated by a 0.

## 3.2 Time Series Forecast

The LSTM model was run for 500 epochs for both companies. The actual and predicted stock prices for GE on test data are shown below.

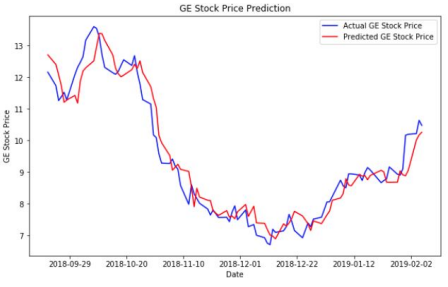


Figure 3 : GE stock price predictions on test data

The actual and predicted stock prices for Apple on test data are shown below.

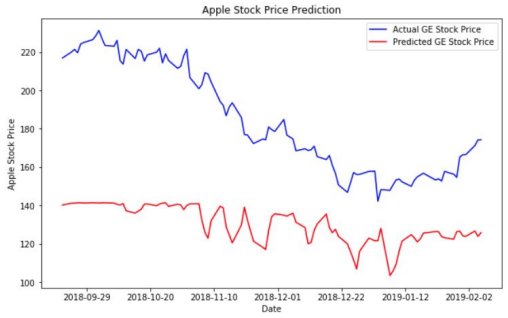


Figure 4: Apple stock predictions on test data

### 3.3 RL Trading Agent

The results of the trading strategy is presented in this section. The RL model was trained for 360 episodes. The model was then tested on the train data to evaluate the performance of the agent in real time. agent in real time. Out of the 50 episodes, the agent epochs. The model was then tested on the train data for 50 episodes to evaluate performance of the agent in real time. Out of the 50 episodes, the agent failed to finish a game for an average of 20 games. A failure is constituted by the agent going bankrupt, selling more shares than it owns or buying more shares than it can afford. Learning the rules of the trade is considered to be a part of training the RL algorithm. We believe training the agent further and tweaking learning parameters will result in lower failure of future episodes. Out of the succeeded episodes, the agent returned an average of 14%. Overall, the profit for all games was an average of 8.6%. The decision process of the agent for the 50 episodes for Apple shares can be visualized in Figure 5. The red dot represents sell, green buy and blue represent a decision to hold the shares.

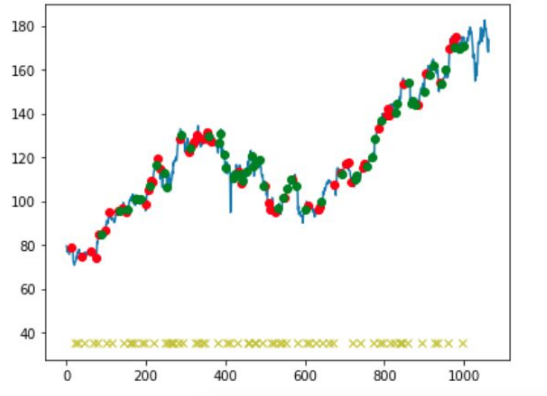


Figure 5: Agent behaviour for Apple shares

## 4 Conclusion

We believe the success of the project was to build a stable trading strategy with an agent capable of making intuitive decisions based on Reinforcement Learning by congregating multiple signals from base models. The model achieved a 14% return on train data. We conclude with the following suggestions for the next steps to improve on the baseline:

- Revisit each of the base models to fine tune parameters to improve predictions.
- Tune reward, states, and other parameters of the RL model for better performance.
- Understand back testing for trading for strong performance on unseen data.
- Experiment with multiple companies to generalize and improve solution.

## 5 References

[1] Fischer, Thomas G. in *Reinforcement learning in financial markets - a survey*

[2] Heeyoung Lee Mihai Surdeanu Bill MacCartney Dan Jurafsky In *On the Importance of Text Analysis for Stock Price Prediction*

[3] Yusuf Aktan in *Using NLP and Deep Learning to Predict Stock Price Movements*

[4] Tom Grek in *A blundering guide to making a deep actor-critic bot for stock trading*

[5] João Maria Branco Carapuço in *Reinforcement Learning Applied to Forex Trading*

[6] David Silver in *Many Faces of Reinforcement Learning*