Introduction

Algorithmic trading also known as automated trading is the process of following a defined set of instructions for placing a trade in order to generate profits at a speed and frequency that is impossible for a human trader using programmed computers.

An important factor affecting the trading decisions is the ability to predict the movement in stock market. The prediction of stock markets movement is considered to be a challenging task of financial time series prediction due to the complexity of the stock market with its noisy and volatile environment, considering the strong connection to numerous stochastic factors such as political events, newspapers as well as quarterly and annual reports.

In this project, we want to explore the possibility of building a machine learning agent that tries to learn an optimal trading policy/strategy using several machine learning techniques like q-learning, an incremental model based on reinforcement learning and combine it with Deep Learning.

Literature Review

Automated Trading isn’t a new concept. Over the years’ various documents have been published by several renowned authors, scientists and finance experts around the world.

From Supervised Machine Learning Models to Unsupervised one’s different people have tried their hand in this trade. The latest of the lot, including us, have been trying to use the concept of Reinforcement Learning a branch of Machine Learning (ML) which has recently been in the spotlight for being at the core of the system who beat the Go world champion in a 5-match series.

A lot of famous articles were written related to our problem statement.

In “Reinforcement Learning For Automated Trading” by Pierpaolo G. Necchi, he talks about how Machine Learning Models differ from Reinforcement Learning. He explains how the use Markov Decision Process affects the computer’s ability to take actions A at a given particular state S to maximize reward R. Taking such actions not only allows the computer to understand the stock market at a certain point of time and learn it, but go as further as to explore different options to maximize the reward.

In “The Predicting Power of Textual Information on Financial Markets” by Gabriel Pui Cheong Fung†, Jeffrey Xu Yu† , Hongjun Lu‡, they talk about how strong the relationship is between the time of news stories releasing and the time of the stock prices fluctuating in the actual market . In this paper, they proposed a systematic framework for predicting the tertiary movements of stock prices by analyzing the impacts of the news stories on the stocks by Mining Textual Documents and time series concurrently.

In “Agent Inspired Trading Using Recurrent Reinforcement Learning and LSTM Neural Networks” by David W. Lu, as the name suggests, he talks about the creation of “hedonistic” learning system. The idea of this learning system may be viewed as Adaptive Optimal Control, his version of reinforcement learning. In order to accomplish similar level of performance and generality, like a human, he felt that we needed to construct and learn the knowledge directly from raw inputs, such as vision, without any hand-engineered features, which can be achieved by deep learning of neural networks particularly an LSTM (Long Short Term Memory) by treating all the features as a function of time. Combining the two, he referred it to simply as deep reinforcement learning, which could create an artificial agent that is as close as we can to sanely call it true “artificial intelligence”.

In “The Structure of the stock trading system”, by Jae Won Lee, an article published more than a decade ago, in 2006, conveys how the dynamics have been shifting in terms of stock market analysis. He gives the insight on why automation in stock market analysis is important in the first place and he talks about using Deep Learning techniques and compares the use of a CNN as opposed to LSTM.

In this project we follow a mixed approach of combining both the implementations of a robust Q-Learning model and information gathered from various sources that may have big impact on the stock market and supplying this additional information to better learn an optimal policy/strategy. Thus the agent not only just learns an optimal trading strategy based on historical prices but also on additional information based on text analysis and trend of the market to make an informed decision.

Research Objective

After a thorough analysis of the stock market, these factors were brought into light that most affects the rise/fall of prices of an asset in the market.

## Industry Performance: More often than not, the stock price of the companies in the same industry will move in tandem with each other. But sometimes, the stock price of a company will benefit from a piece of bad news for its competitor if the companies are competing for the same market.

Investor sentiment: Investor or confidence can cause the market to go up or down, in one of the two ways:

* **bull market –** a strong stock market where stock prices are rising and investor confidence is growing. In short, an economic boom.
* **bear market –** a weak market where stock prices are falling and investor confidence is fading. In short, an economic recession.

Interest Rates: If a company borrows money to expand and improve its business, higher interest rates will affect the cost of its debt. This can reduce company profits and the dividends it pays shareholders. As a result, its share price may drop. And, in times of higher interest rates, investments that pay interest tend to be more attractive to investors than stocks.

Changes in economic policy: If a new government comes into power, or the board member of a company changes, it may decide to make new policies. Sometimes these changes can be seen as good for business, and sometimes not. They may lead to changes in inflation and interest rates, which in turn may affect stock prices.

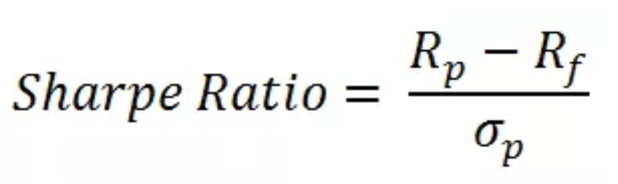
Value of the Country’s Currency: Many companies sell products to buyers in other countries. If their currency rises, their customers will have to spend more to buy their goods. This can drive down sales, which in turn can lead to lower stock prices. When the price of the currency falls, it makes it cheaper for others to buy our products. This can make stock prices rise.

Problem Definition

Initially when stock market automation was done using Machine Learning, while the system worked well for learning previous historical values in tandem and predicting whether the market will fall or rise based on previous values of a certain trade stock, it could never properly maximize the profit outcome. Using Q-Learning (or maybe TD Learning) we hope to successfully solve this problem by telling the user when to buy, hold or sell certain stocks.

We can also choose to compare different stock options, for a user to maximize his profit by making sure he doesn’t put all his eggs in one basket and instead divide this total capital in the most optimized way. This can be obtained by calculating the Sharpe Ratio.

Sharpe Ratio is defined as the ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk. The Sharpe ratio is calculated as follows:



Where,

Rp = Return of portfolio

Rf = Risk-Free rate

σp = Standard deviation of portfolio’s excess return

A Sharpe Ratio obtained between:

0-1 is a Safe Investment

1-2 is a Good Investment

2-3 is a Very Good Investment

3 and a above is an Excellent Investment

We hope to achieve a Sharpe Ratio between 2.2 and 2.6 in a our project as a development on the various pre existing one’s, all ranging between 1.4 and 2.1, by combining multiple features such as Q-Learning, LSTM Deep Learning Network and Text Mining and Analysis of factors that affect the stock market.

Existing System

In “Agent Inspired Trading Using Recurrent Reinforcement Learning and LSTM Neural Networks” by David W. Lu, he gives us valuable Insight on how and why he uses a Recurrent Reinforcement Learning Algorithm, instead of a normal Reinforcement Learning Algorithm.

He talks about combining the power of Recurrent Neural Networks with the power of reinforcement learning to make the series of States-Action-Rewards steps more optimized using an algorithm that performs stochastic optimization or effectively makes the algorithm a stochastic gradient ascent. It is then optimized and verified with a Weight Decay Variant as and we partially differentiate to go back to optimize the loss function generated.

In the next step we get an understanding of how he uses an LSTM to understand the dynamically changing market condition and how uses the informative features for his learning model. He gives us insight on the failings of Recurrent Neural Network, claiming that even though in theory it is able to connect previous information to the present task, something that both him and we are hoping to achieve, unfortunately in practice it is possible that the gap between relevant information and the point where that information is required because surmountably large. As the gap grows RNN cannot learn to connect the information properly.

Hence an LSTM was used because it solves the fundamental issue of the Gradient Vanishing Problem. However, to utilize the benefits of a recurrent neural networks, he incorporated LSTM and dropout in unison to tackle the challenges in a recurrent neural network, something we hope to incorporate in our project as well.

In the last section to fine-tune his model even more, he explores other methodologies that are closer to “Black-Box” Optimization. He explains “many real world optimization problems are too complex to model directly.” being his main motivation behind this idea. We also hope to follow a similar Heuristic Approach as and when required because even though it might not be clean and suave, it could solve our immediate problem.

Requirements Analysis:

Hardware Requirements:

Processor: Intel i5 6th Generation.

GPU: GTX 210/ Iris 620

RAM: 4Gb

Software Requirements:

#### Jupyter Notebook - The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

**Anaconda Navigator** - Anaconda Navigator is a desktop graphical user interface (GUI) included in **Anaconda**® distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands.

**Pandas** - In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

**Keras** - Keras is an open source neural network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

**TensorFlow** - TensorFlow is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks

**NumPy** - NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Proposed Methodologies

**Describing Issues In Existing Methodologies:**

In previous Proposed Methodologies, especially the one’s that have been closely followed, the Sharpe Ratio (the most important factor when it comes to assessing the Profits Generated against a risk free Investment), have all achieved a score ranging between 1.4 and 2.1.

One of the best papers upon which we base our project have 2 major points to be talked about.

One is the use of Recurrent Reinforcement Learning model which is more of a White Box Texting method, combining a Recurrent Neural Network with that of a Reinforcement Learning to generate mathematical formulas to solve the problem.

The other one, is lack of any sort of vindictive text analysis and emotions that might actually affect the stock market. We hope to capitalize on this.

**Describing Need For New Technology**:

A Sharpe Ratio, against an Investment, of 3.0 is considered magnanimous and makes the Investment an excellent choice. However in practice it is impossible for us to achieve such numerical values now.

Instead we will try and hope to achieve a Sharpe Ratio anywhere between 2.2-2.6 which will be an increment on the existing technology.

**Method of Implementation:**

As spoken about earlier, we intend to tackle or at least lessen the main problems in the existing methodologies by these two following ways.

We intend to use Q-Learning instead of a direct Recurrent Reinforcement Learning as it more Robust to adjusting itself for models when the data set is much cleaner (something we hope to get either directly or by pre-processing of data as per requirement), just like Deep Learning Neural Net as Opposed to a Machine Learning Network.

We hope to inculcate the factors which we have previously spoken about, that have a great say in affecting the stock market. We intend to extract information mainly from Text Articles either using Text Mining Techniques or Computer Vision to add meaning to our dataset.

**Describing Modules and It’s Functionalities:**

We use to use Q-Learning over the following features in a cleaned dataset (obtained after pre-processing of data.)

For our problem, we used the following feature vectors:

(a) Number of Stocks of each asset,

(b) Current Stock Price of each asset and,

(c) Cash in Hand.

We will get a certain Sharpe Ratio as an output, which we intend to compare with David W. Lu’s Recurrent Reinforcement Output.

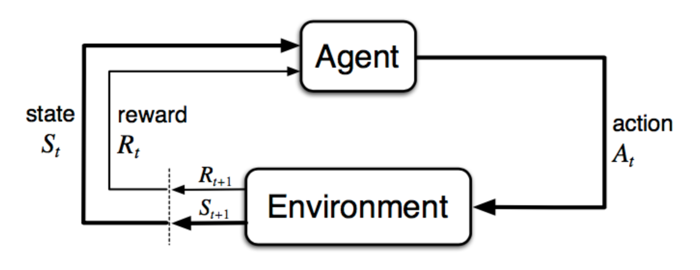
Then we will feed the network with Technical Mathematical Financial Models, such as Moving Averages, Larry William’s R Index, etc. (P.S. This part of our model is variable and undecided as of yet and requires more Financial Research and Understanding.)

Then we will analysis Text Articles to extract meaningful information, (information stated above), and perform Text Mining after which we will use a Word2Vec Model to feed information into a LSTM, combined the results obtained from the previous Network.

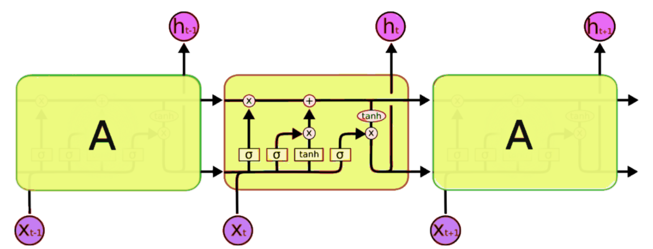
Combining all 3 Networks, we hope to get a final output of a greater Sharpe Ratio than previous papers.

**System Design**

Q-Learning MDP Architecture:



LSTM Architecture:



Our Project Pipeline:

