**Project Title**

Re-implementation “*An application of deep reinforcement learning to algorithmic trading*”, Theate and Ernst (2021) for BLG 561E Deep Learning Lecture.

**Aim**

The aim of this project is to design a trading policy in the context of algorithmic trading using a feedforward deep neural network which can make trading decision on stock markets based on previous movements of stocks.

**Summary**

Algorithmic trading is a method of making trading decisions in various markets/mediums such as stock markets, foreign exchange, and energy exchange based on a set of mathematical rules realized by a computer. In general, a basic trading algorithm takes various information such as *Open*, *Close*, *Volume* values of a stock or a set of stocks as an argument, including past and real-time values, then the algorithm generates a signal indicating the operation to realize such as buying or selling. Designing an algorithmic trading strategy is a complex decision-making problem since the environment (e.g., stock markets) in which an algorithmic trading method operates have a stochastic and sequential nature. On the other hand, the success of *deep learning* applications in various fields such as speech recognition, image classification or natural language processing has made deep learning techniques a potential candidate in designing trading policies.

In this project, we are going to build (re-implement) a feedforward deep neural network which is going to serve as novel trading policy. The implemented deep neural network is going to make trading decision for a set of predetermined stocks individually. (e.g., the network will not manage a portfolio.) Then the results of the deep neural network are going to be compared with classical trading strategies such as buy and hold (B&H) and trend following moving averages (TF) etc.

**Design & Methods**

There are three main components of this projects: Data obtainment, Trading operations and Deep learning network.

1. **Data obtainment**: To be able to generate the same results obtained in the article, the same datasets will be used. The dataset consists of eight years of past data (from 2012 to 2017 for training and from 2018 to 2019 (including) for test) of various stocks such as APPLE, GOOGLE and TESLA etc.
2. **Trading operations:** In normal circumstances, to be able realize a trade operation there must be a match between bid and ask orders. In order to detect these matches, one must be able access the *order book* of the stock in question. However, for the sake of simplicity we will assume that the total number of buy and sell orders are much smaller than the volume of the stock in question. Therefore; we are going to implement only buy and sell operations with trading costs.
3. **Deep Learning Network**: The deep neural network architecture is a feedforward deep neural network, since the dataset we are going to use has a sequential nature (time-series). The lost function is chosen as Huber loss function.

**Milestones**

1. Obtainment, manipulation and augmentation of datasets.
2. Implementation of fundamental operations (Buy & Sell) including trading costs (spread, market Impact, timing etc.) and constraints (mechanisms to prevent unrealistic buying or selling scenarios).
3. Testing the implemented fundamental operations using a dummy algorithmic trading strategy.
4. Implementation of the deep neural network architecture in Pytorch framework.
5. Reproduction of the article results.

**References**:

1. Theate T. & Ernst D. (2021). *An application of deep reinforcement learning to algorithmic trading.* Expert Systems with Applications, 173.