# Trading Stock with Deep Reinforcement Learning

Yixiao Song, Zhongjie Li, Stephen Guan Feng Yang
(syixiao/jay2015/sgyang)@stanford.edu

Stanford University

## Introduction

The stock market has largely been a mystery to many investors. Few have been able to consistently and reliably predict its movements. We hope to solve the challenge of stock trading by building an AI agent that can accurately predict stock trends.

## Problem Statement

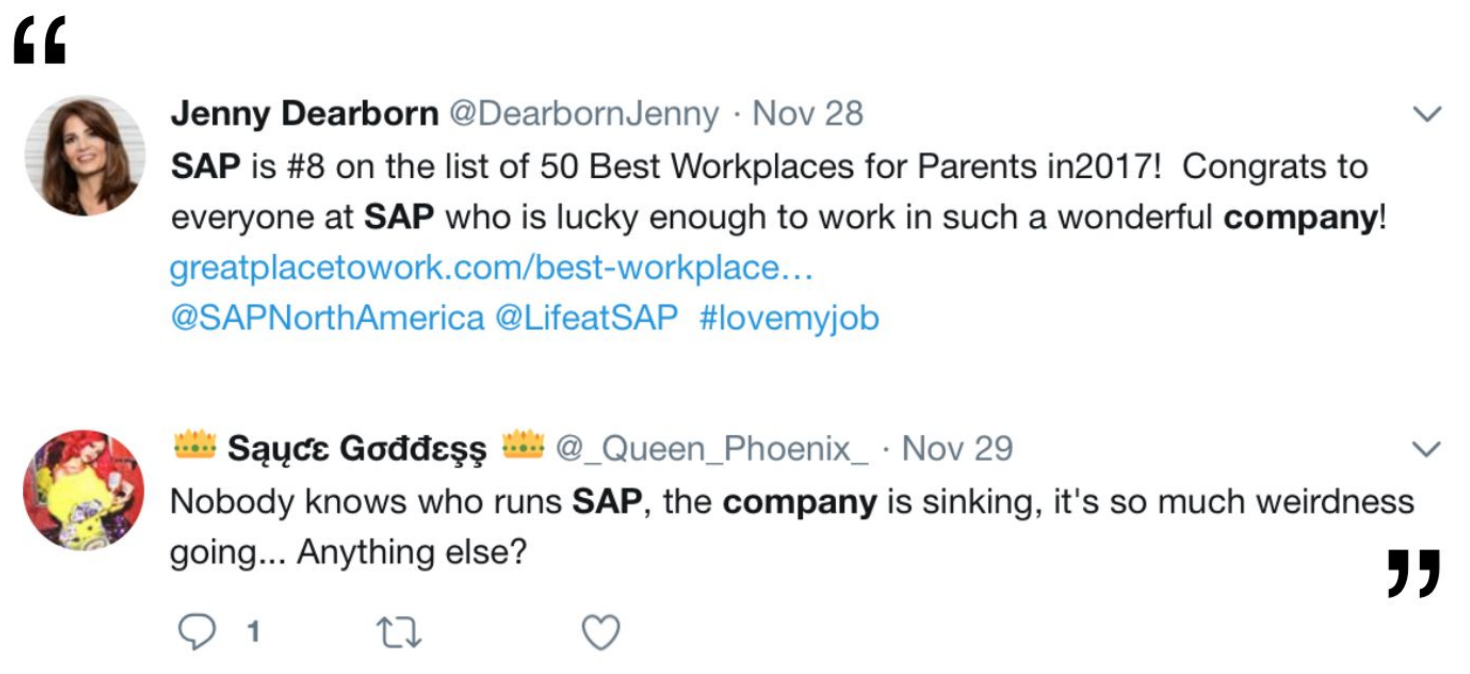
We hope to apply several variants of the technique of Reinforcement Learning, including deep Q learning, and double Q learning, to build a stock trading agent. We will compare the total rewards obtained by picking actions according to each of the generated policies.

## Datasets

The dataset we used for our results is the daily(Open and Close) stock price history of SAP from 1/1/2016 to 10/1/2017. We are training our data from 1/1/2016 to 1/1/2017, and then run prediction from 1/1/2017 to 10/1/2017.

We also have about 13 million tweets from twitter tracked by the company's name that we used to do sentiment analysis, which we hope to incorporate into features of some of the Q learning variants that we do.

For our current model, we introduced a 2$ cost with each action taken. We defined the total reward to be the total cash at the end of the test period, starting initially with $1000.



“

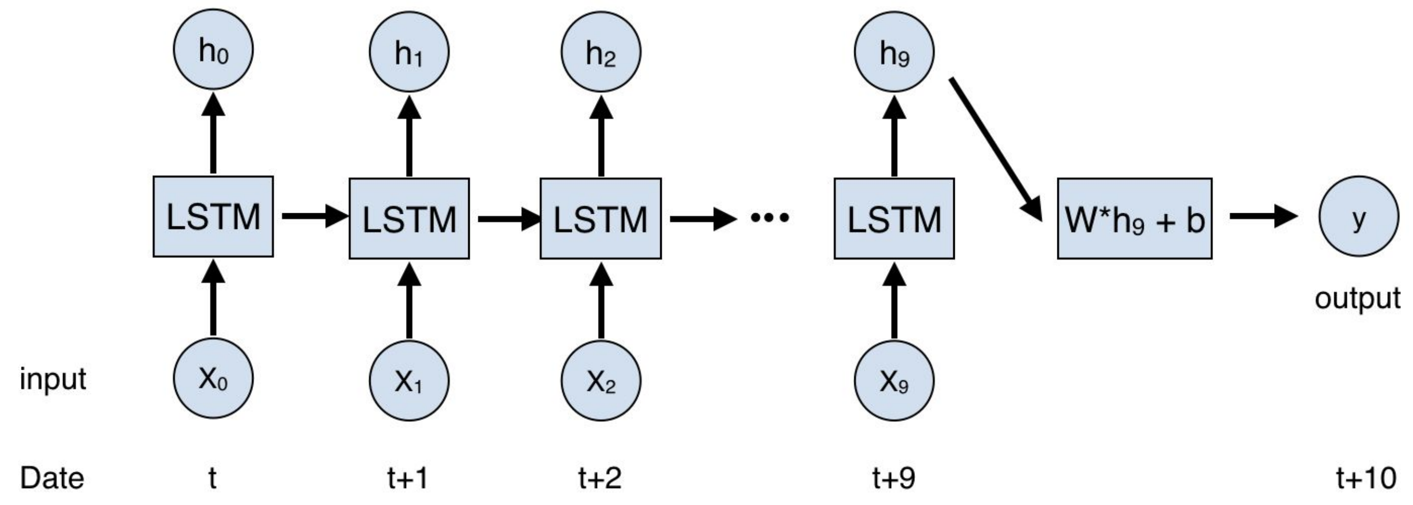
**Jenny Dearborn** @DearbornJenny · Nov 28
**SAP** is #8 on the list of 50 Best Workplaces for Parents in2017! Congrats to everyone at **SAP** who is lucky enough to work in such a wonderful **company**!
greatplacetowork.com/best-workplace…
@SAPNorthAmerica @LifeatSAP #lovemyjob

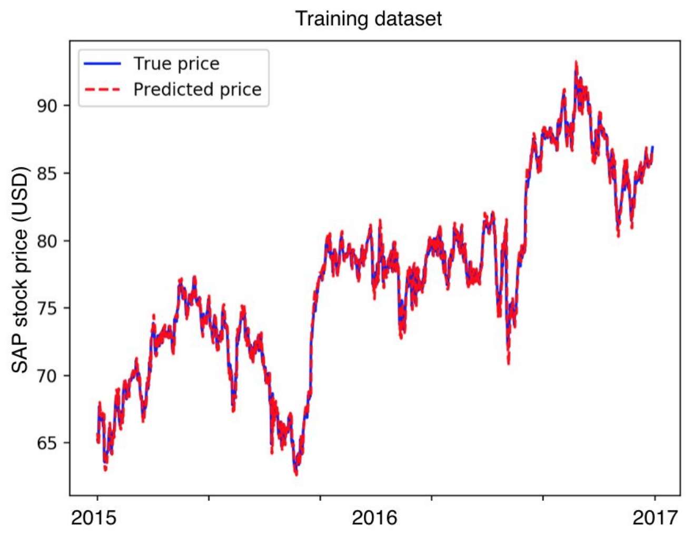
**Sąųcε Goďđεşş** @_Queen_Phoenix_ · Nov 29
Nobody knows who runs **SAP**, the **company** is sinking, it's so much weirdness going... Anything else?

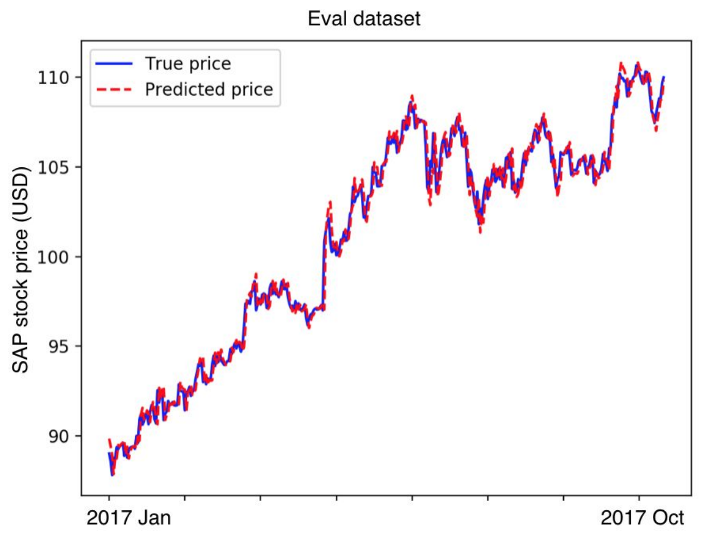
”

## Stock Prediction(feature)

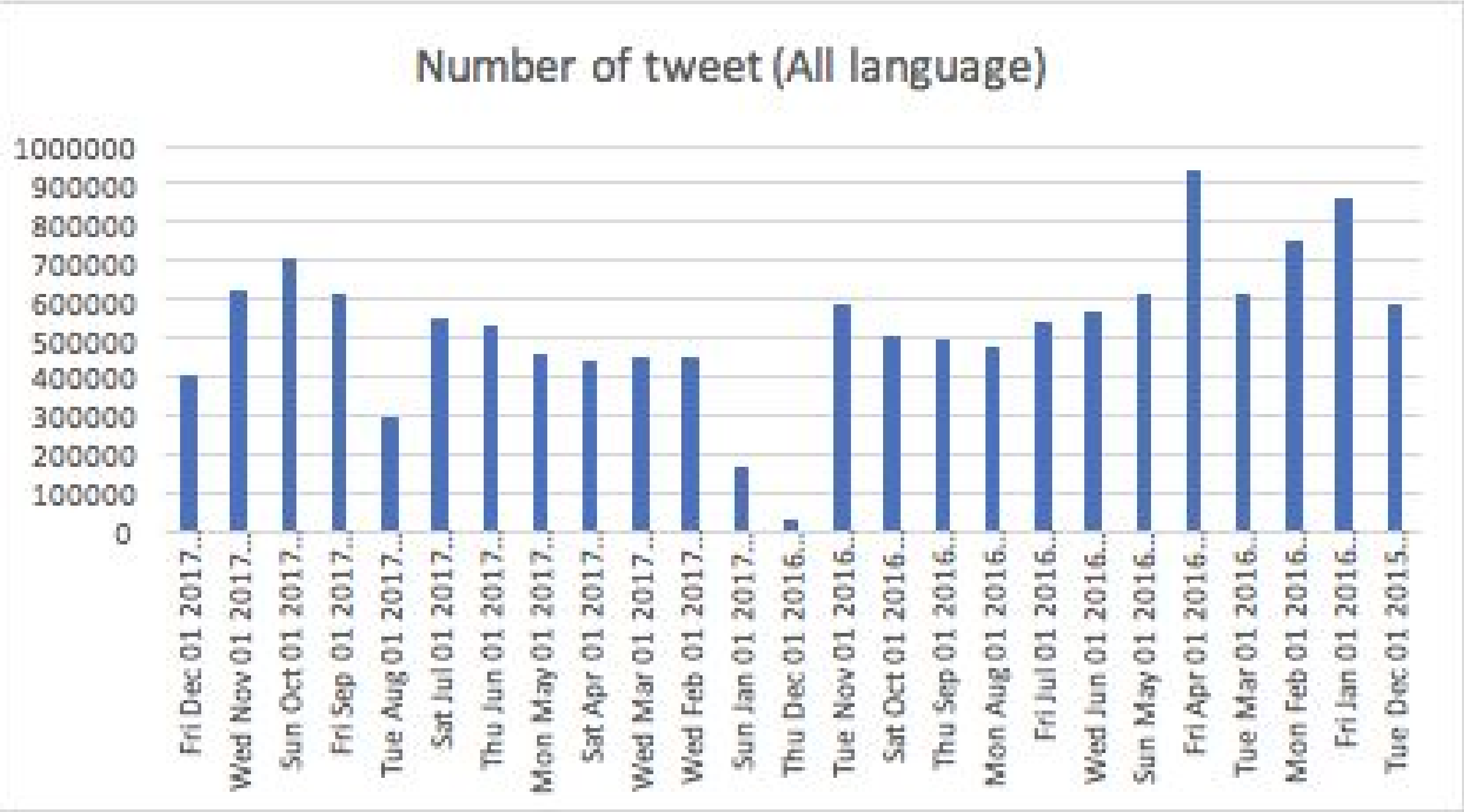
To predict stock price in next day, we implemented 3 different models, including linear model, fully connected network and LSTM model. Following shows the LSTM model and its results.

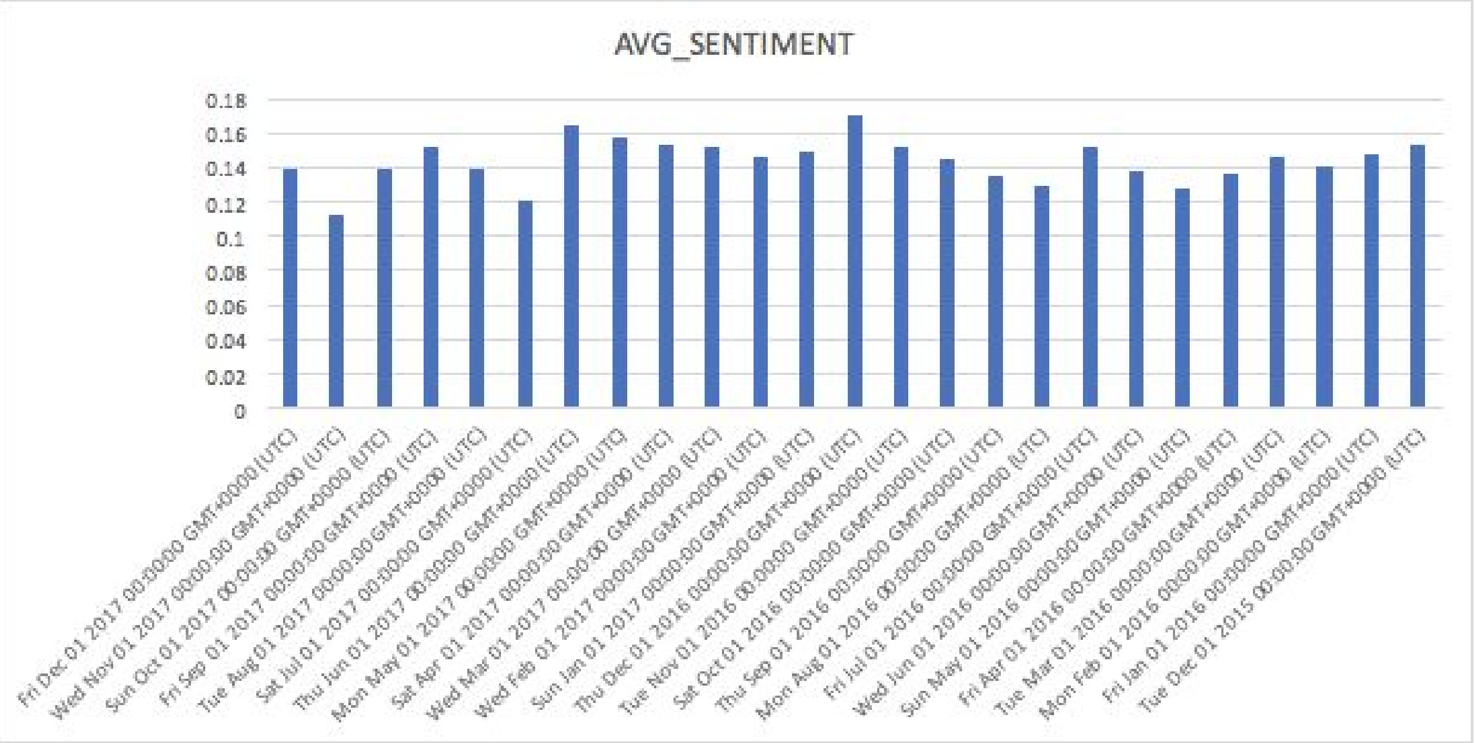






## Sentiment Analysis(feature)





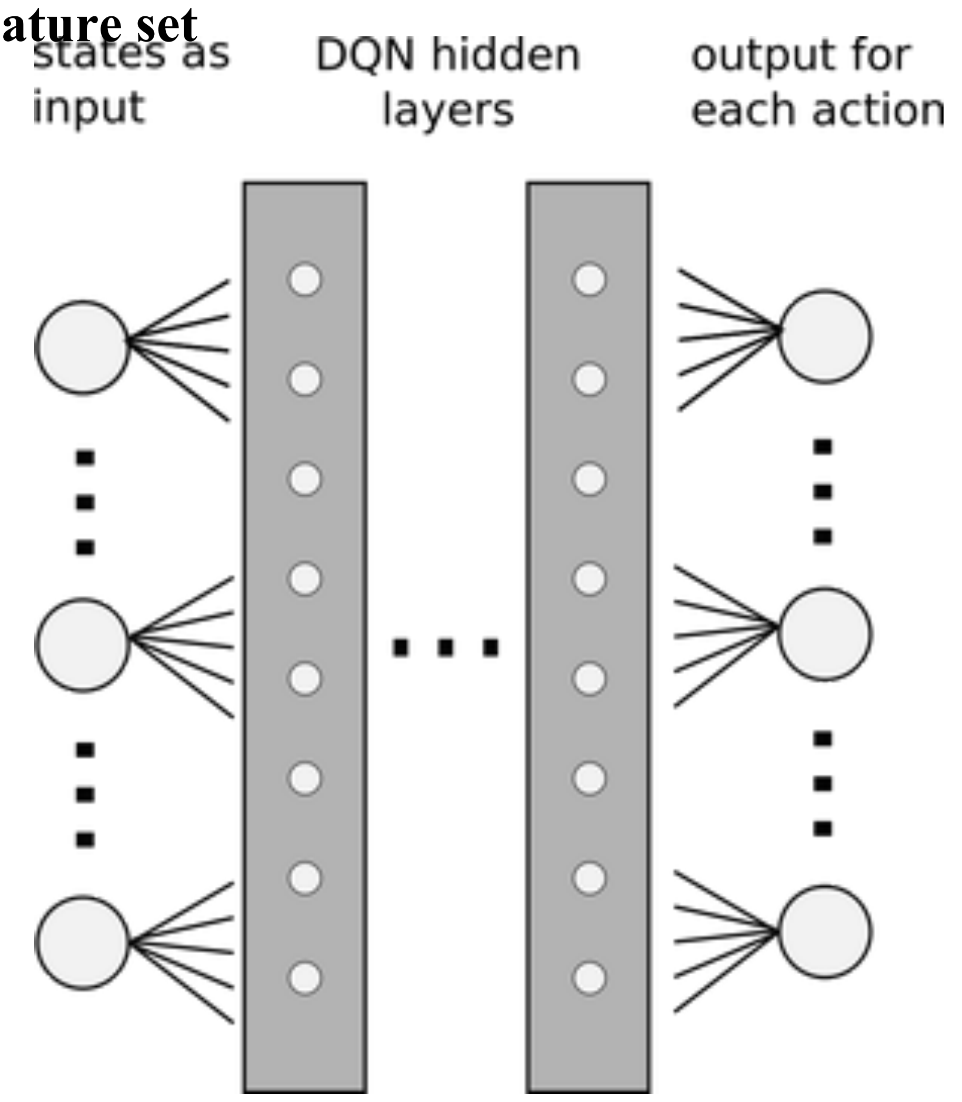
## Trading Policy

1. **DQN (1 hidden layer)**
2. **DQN (2 hidden layer)**
3. **DQN (2) +RAM(Randomized Agent Memory)**
4. **DDQN(2)(Double learning) +RAM**

$$Q_1(s,a) \rightarrow r + \gamma Q_2(s', argmax_a Q_1(s',a))$$
$$Q_2(s,a) \rightarrow r + \gamma Q_1(s', argmax_a Q_2(s',a))$$

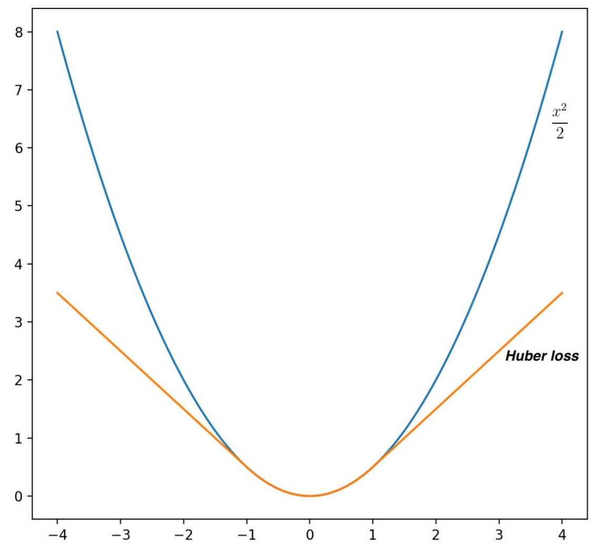
5. **DDQN(2)+RAM+PER(Prioritize experience replay)**

**6 Add Stock prediction ,and Sentiment Analysis into DQN feature set**

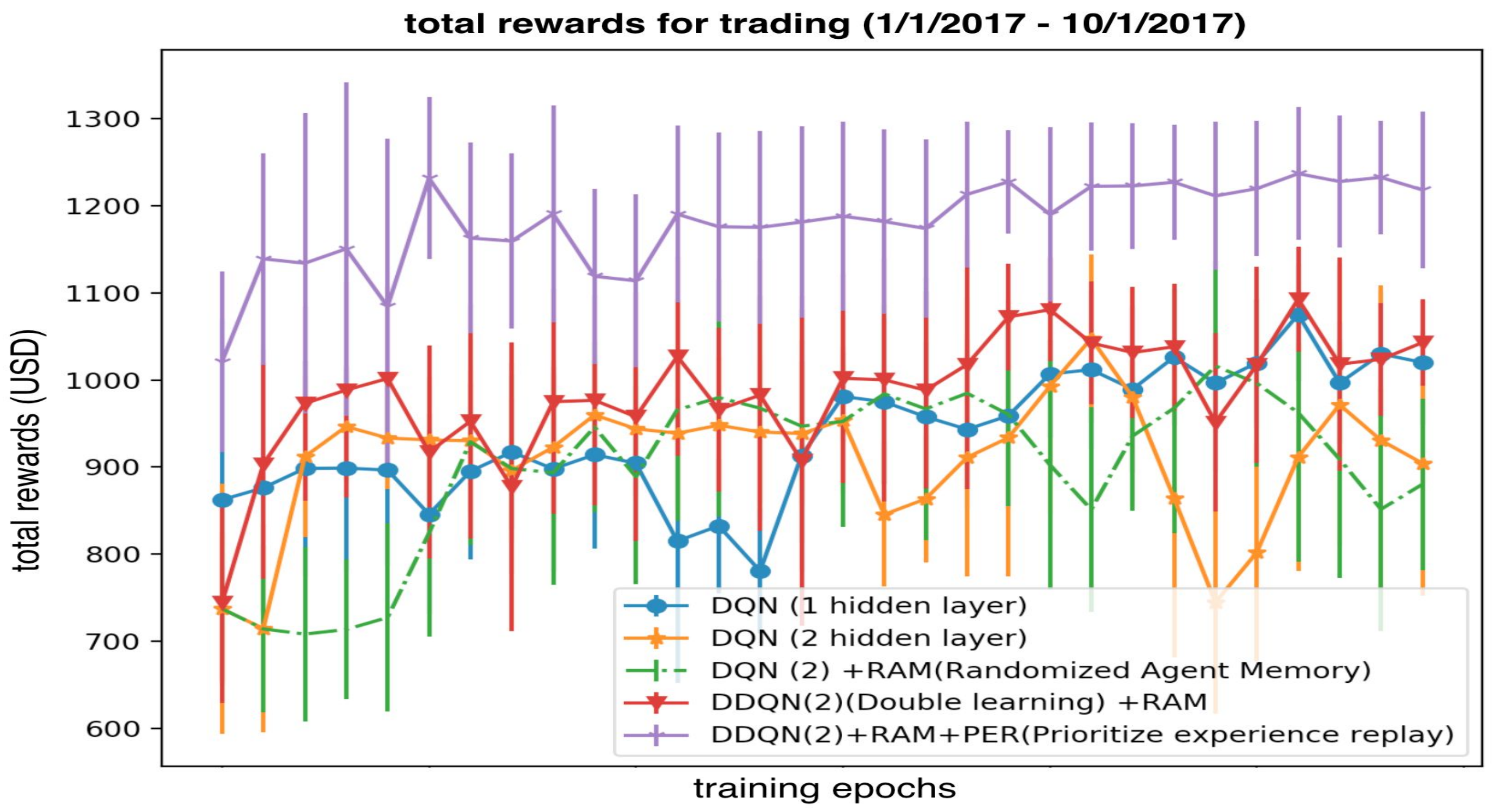


The DQN network size and layer may need additional adjustment, Too large can lead to overfitting and too small may fail to approximate Q.

Loss function we are using MSE, However other loss function could also have better estimation.eg. huber loss



Huber loss



total rewards for trading (1/1/2017 - 10/1/2017)
total rewards (USD)
training epochs
DQN (1 hidden layer)
DQN (2 hidden layer)
DQN (2) +RAM(Randomized Agent Memory)
DDQN(2)(Double learning) +RAM
DDQN(2)+RAM+PER(Prioritize experience replay)