BVA 529E FINAL

ANIL FERDİ KAYA 090180128

STOCK TRADER

1. STOCK DATA ENVIRONMENT

2. SGD STOCK TRADER

3. DQN STOCK

4. COMPARISION AND CONCULUSION

STOCK DATA ENVIRONMENT

THE STOCK DATA

. We have 3 different stocks from APPLE, MSI and STBUX
. We provide the agent trade the stocks and maximize the rewards

	AAPL	MSI	SBUX
0	67.8542	60.30	28.185
1	68.5614	60.90	28.070
2	66.8428	60.83	28.130
3	66.7156	60.81	27.915
4	66.6556	61.12	27.775
...	...	...	...

```
env = MultiStockEnv(train_data,initial_investment=20000)
state=env.reset() #initializes the state 0 from each stock and $20000 at hand
while not done:
    action=get_action(state) #This will be got from agent
    #perform the trade and go to next state, info will give the current value,
    #reward is incremental value gain
    next_state,reward,done,info=env.step(action)
    state=next_state
```

- We have initial invesment, time series of stocks.
- 3 different action for each stock 0,1,2 [sell,hold,buy]. (possible 27 action)
- S, state = [share1,share2,share3,price 1,price2,price3,portfolio] (2n+1 element)

MultiStockEnv the environment of our problem. You can see the .ipynb notebooks on class or my .ipynb notebooks.

SGD STOCK TRADER

SGD WITH MOMENTUM FUNC

$$V_t = \beta V_{t-1} + \alpha \nabla L(W, X, y)$$
$$W = W - Vt$$

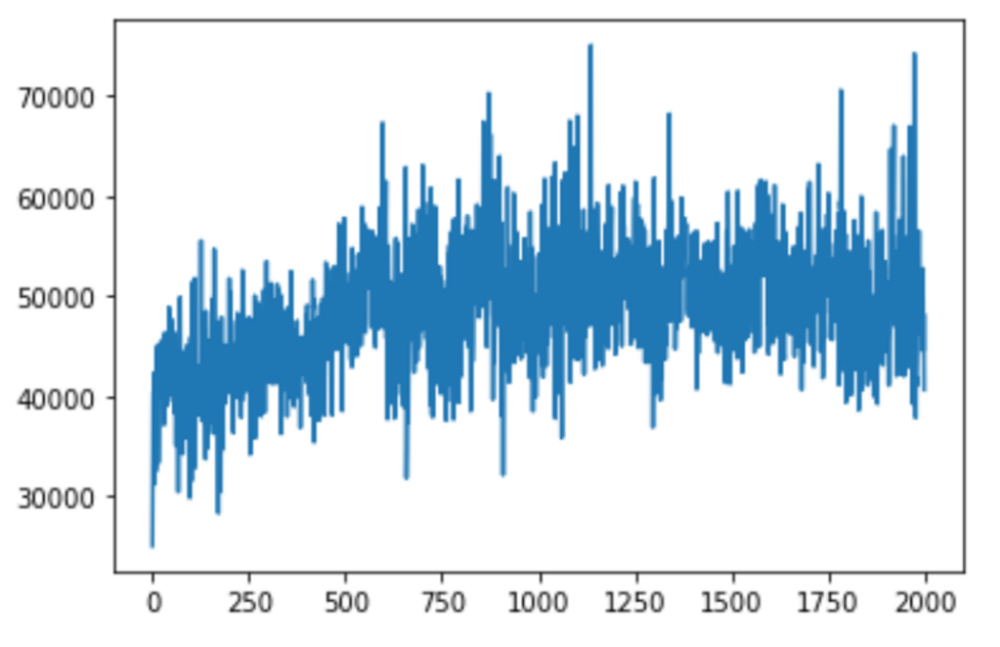
•Where β is the momentum term, α is the learning rate, and W, is the weight vector (or matrix), and

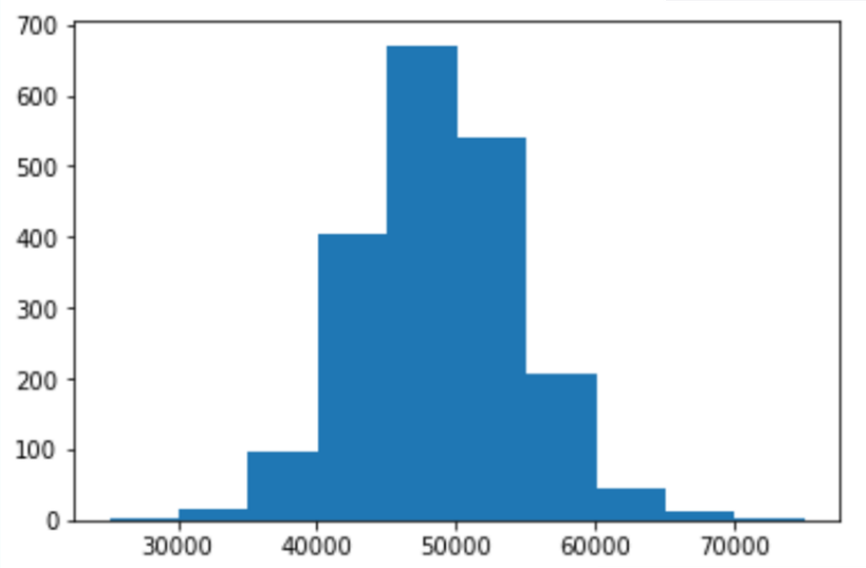
$$\nabla L(W, X, y) = \frac{1}{NK} 2X^T(\hat{y} - y)$$

is the gradient of the loss function (e.g., MSE)
K is the number of actions. Why do we divide to K as well?

SGD Training Results

. Dividing first %50 term of training
. Other %50 term of test
. Train Results;
. Our # of episodes: 2000

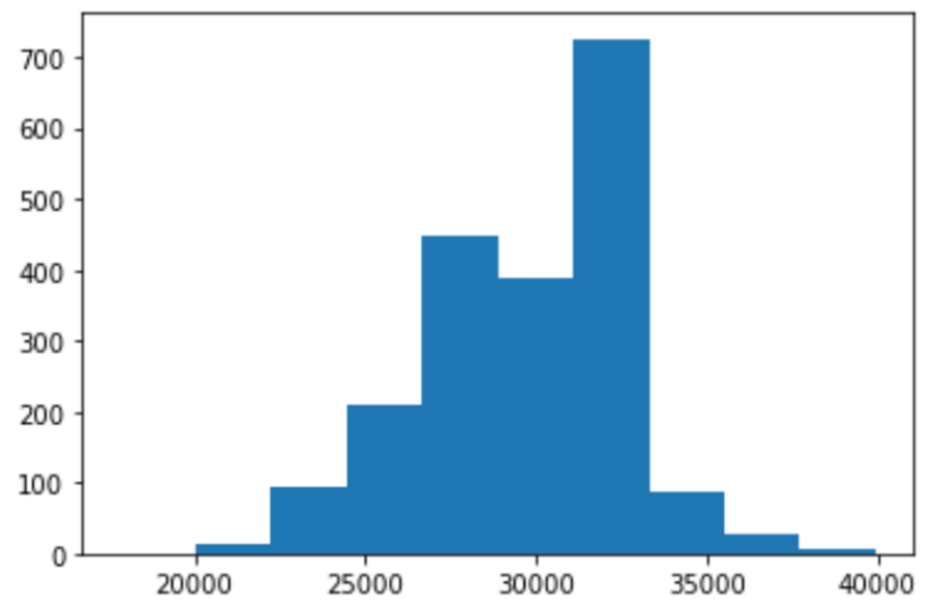




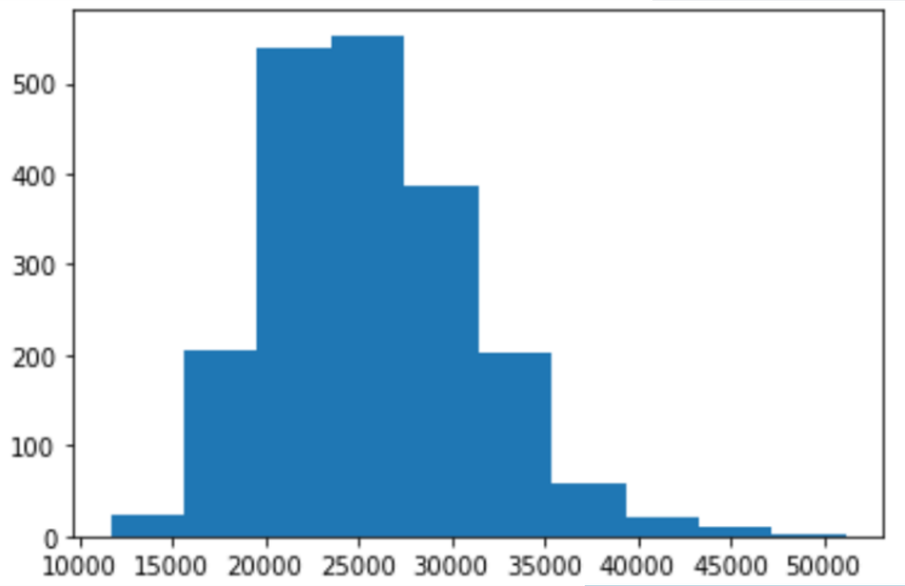
SGD Test and Random Results

. Test and Random Policy Results;

. Our # of episodes: 2000



.Test Policy



.Random Policy

DQN STOCK TRADER

- Used same environment
- NN Model Using For Function of Algorithm
- Single DQN Approach
- Code in .ipynb notebook

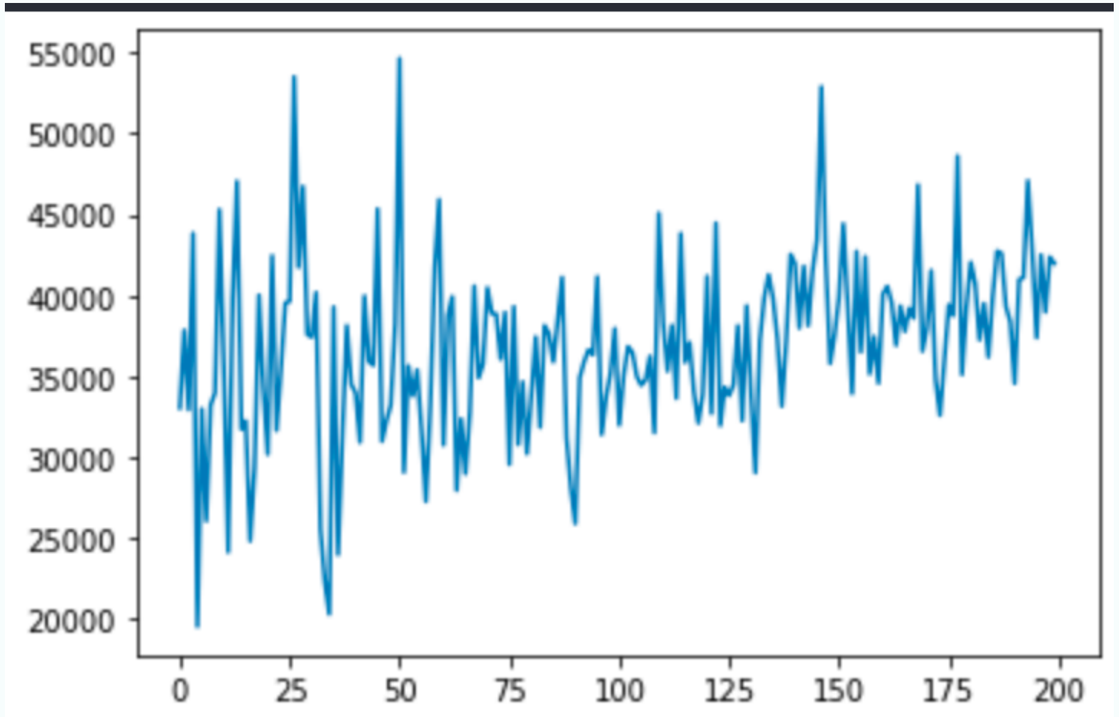
DQN Training Results

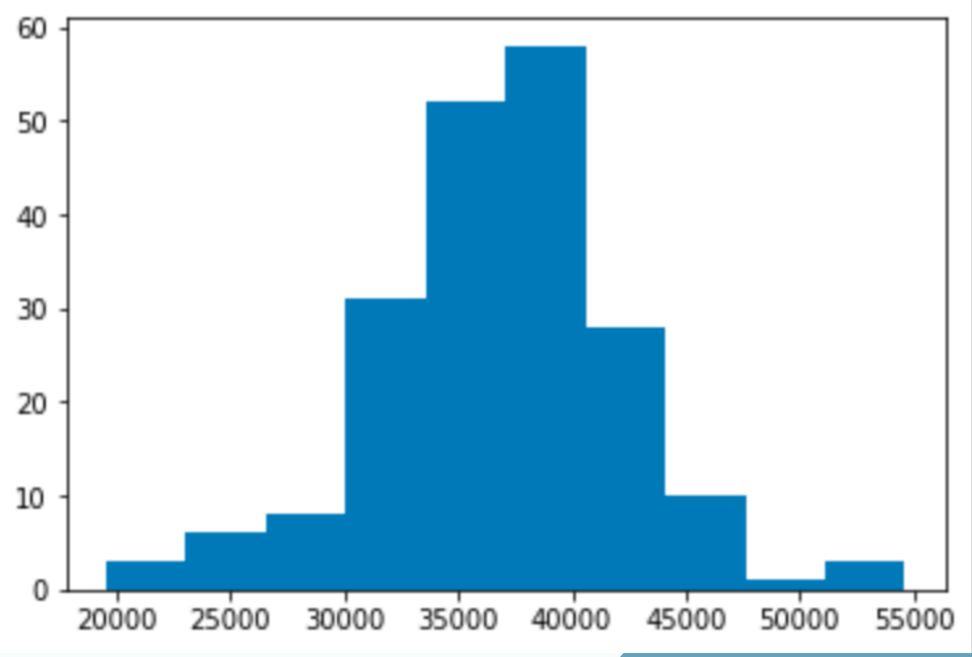
. Used same training and test data

Of samples 200

.. Training sample linearly increase

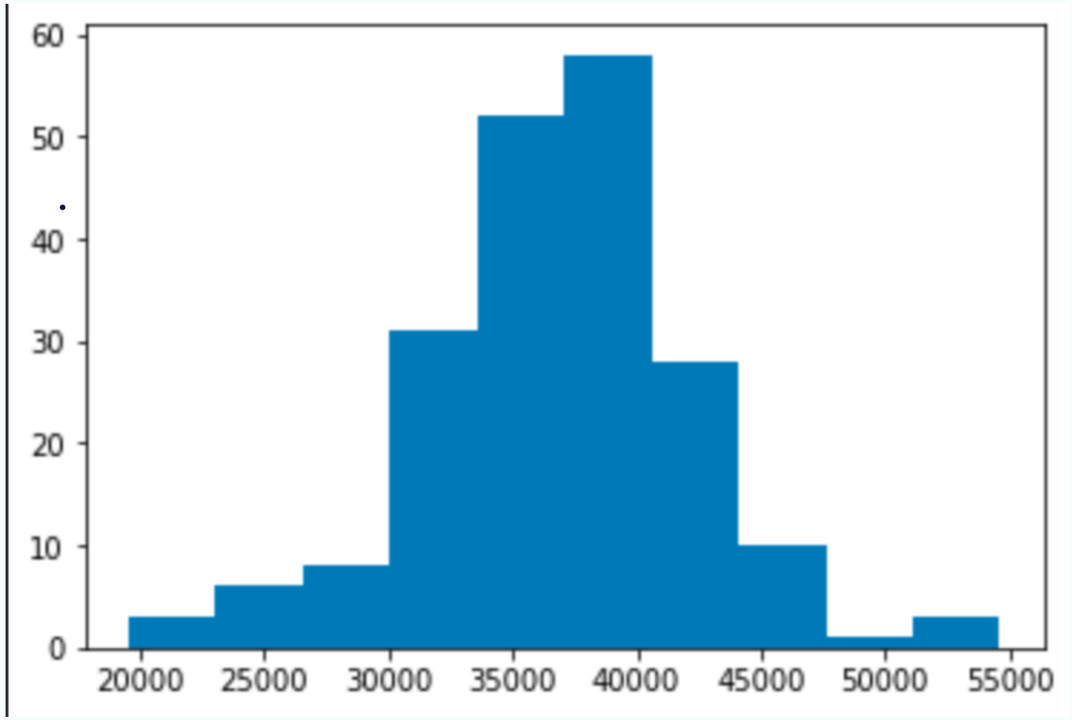
. Mean 36781.78243800005

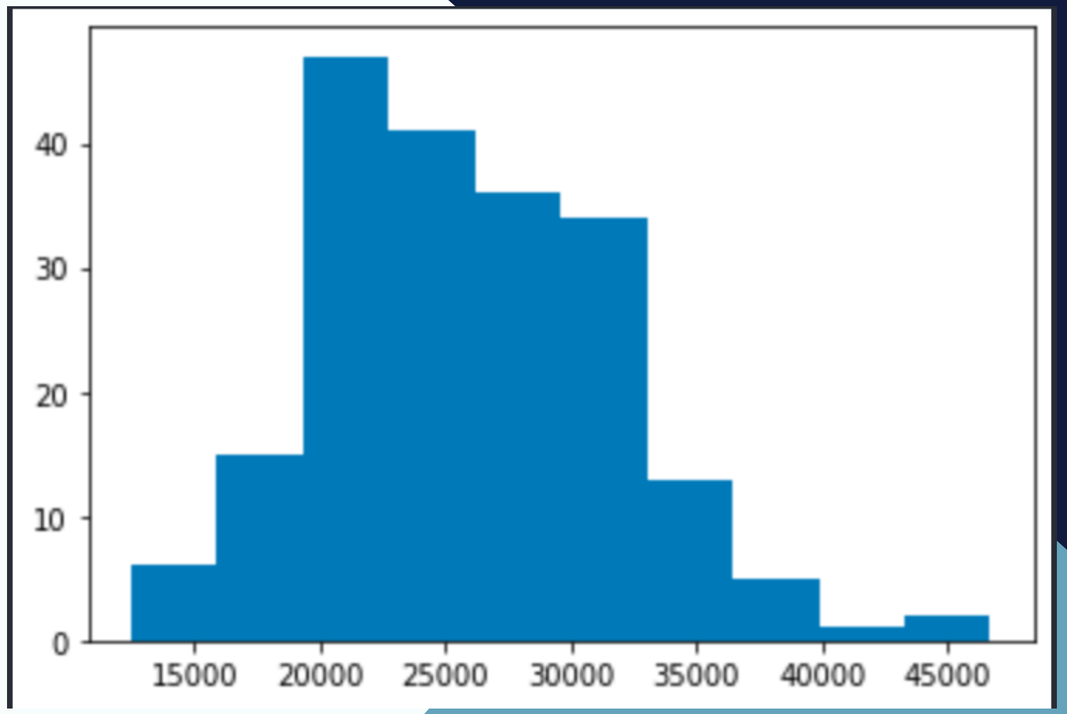




DQN Test Results

. Used same training and test data.
. Test portfolio on left
. Random portfolio on right.





Test portfolio mean: 25045.30584999997

Random portfolio mean: 26051.488099999682

Conculusion

- **SGD function with # of 2000 better portfolio results than DQN function with # 200 samples.**

- **DQN function with # of 200 random results mean better than random portfolio but the histogram shows the median of test set better than random.**

But we can see the test results little bit learn environment

- **DQN function with # of 200 Results can be perform well when increase the # of training samples.**

Training time is too long.