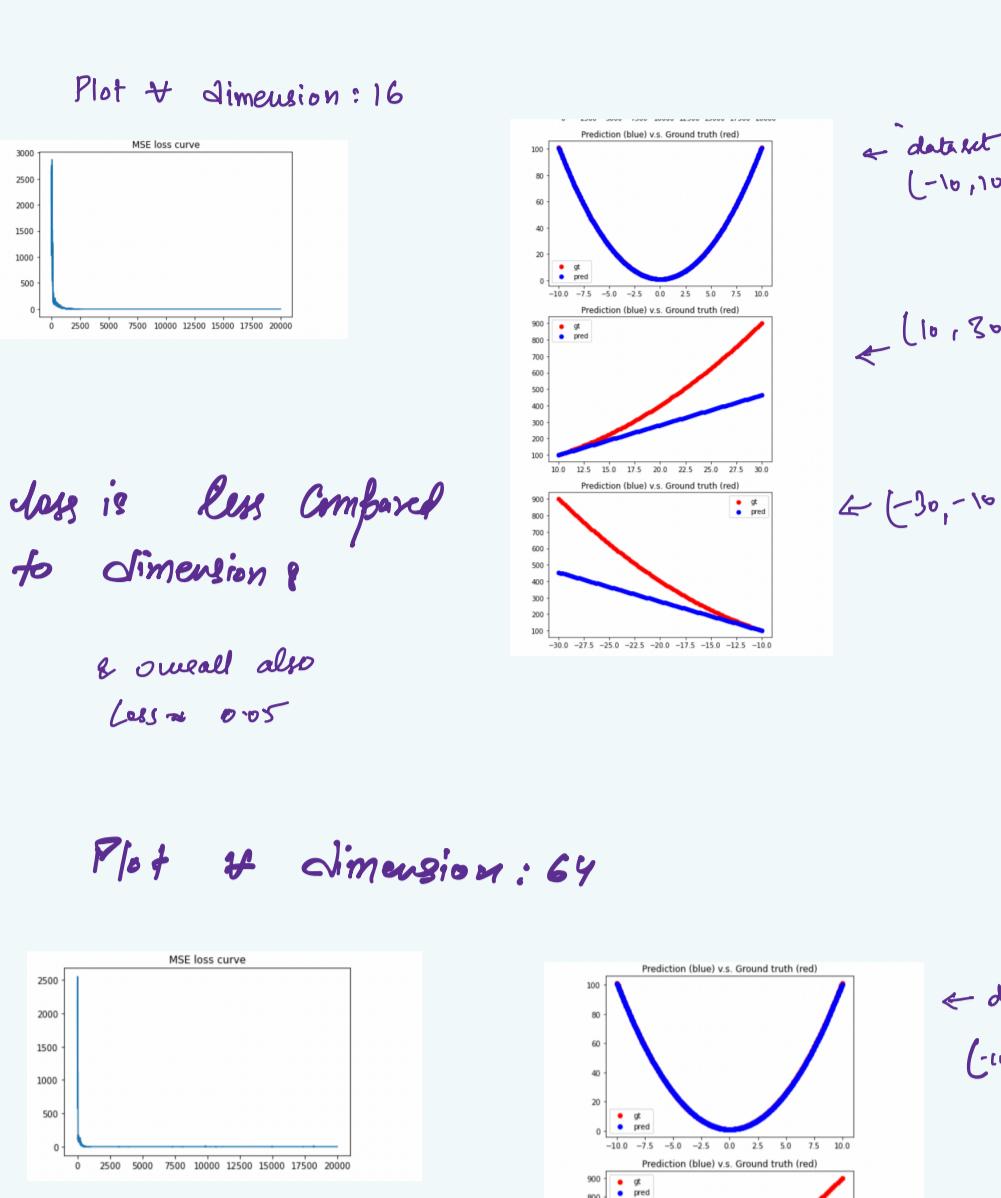


## Q.1.

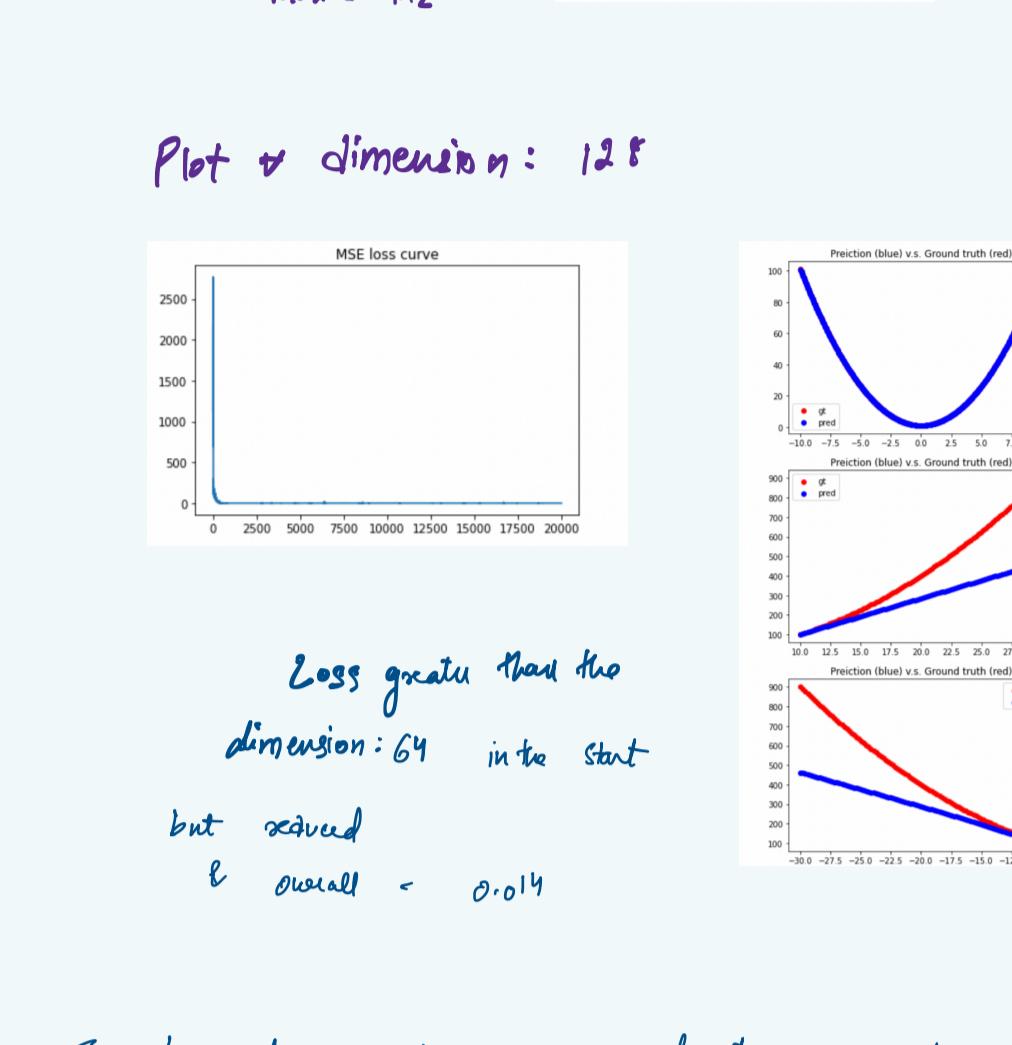
1.3 points. Nonlinear Function Approximation with Neural Networks

Universal approximation theorem tells us Neural Networks have a good deal of approximation capabilities. In this question, we will be approximating a low-dimensional non-linear function  $f(x) = 1 + x^2$  using Neural Network through stochastic gradient descent. Specifically, this is a simple regression problem in machine learning. The goal is to learn a parameterized model to predict the value of  $f(x)$  given  $x$ . To build the neural networks, we will use the PyTorch library (<https://pytorch.org>), a popular library to build deep neural networks. We recommend you to read the tutorials on the official website if you want to learn more about Pytorch (<https://pytorch.org/tutorials/>).

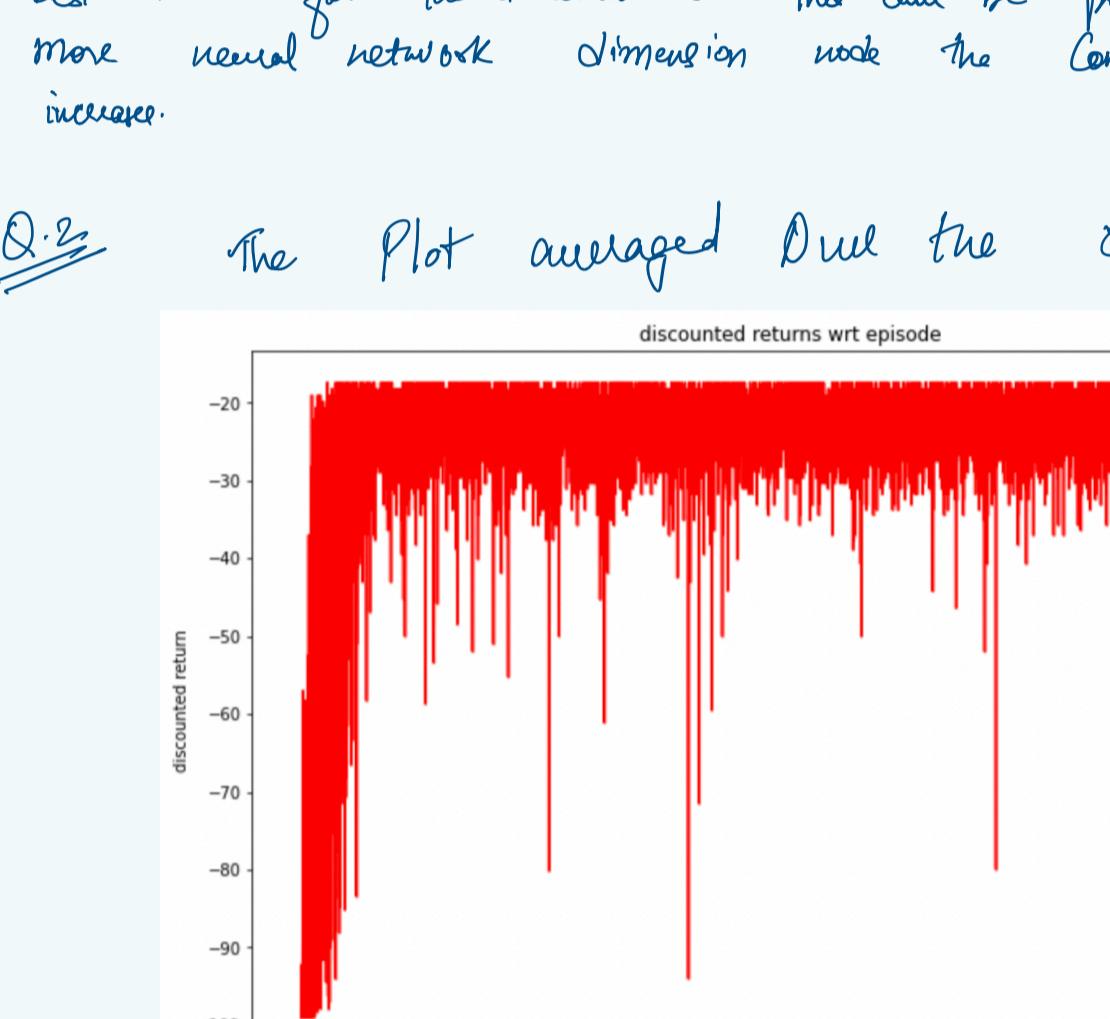
Plot &amp; dimension : 8



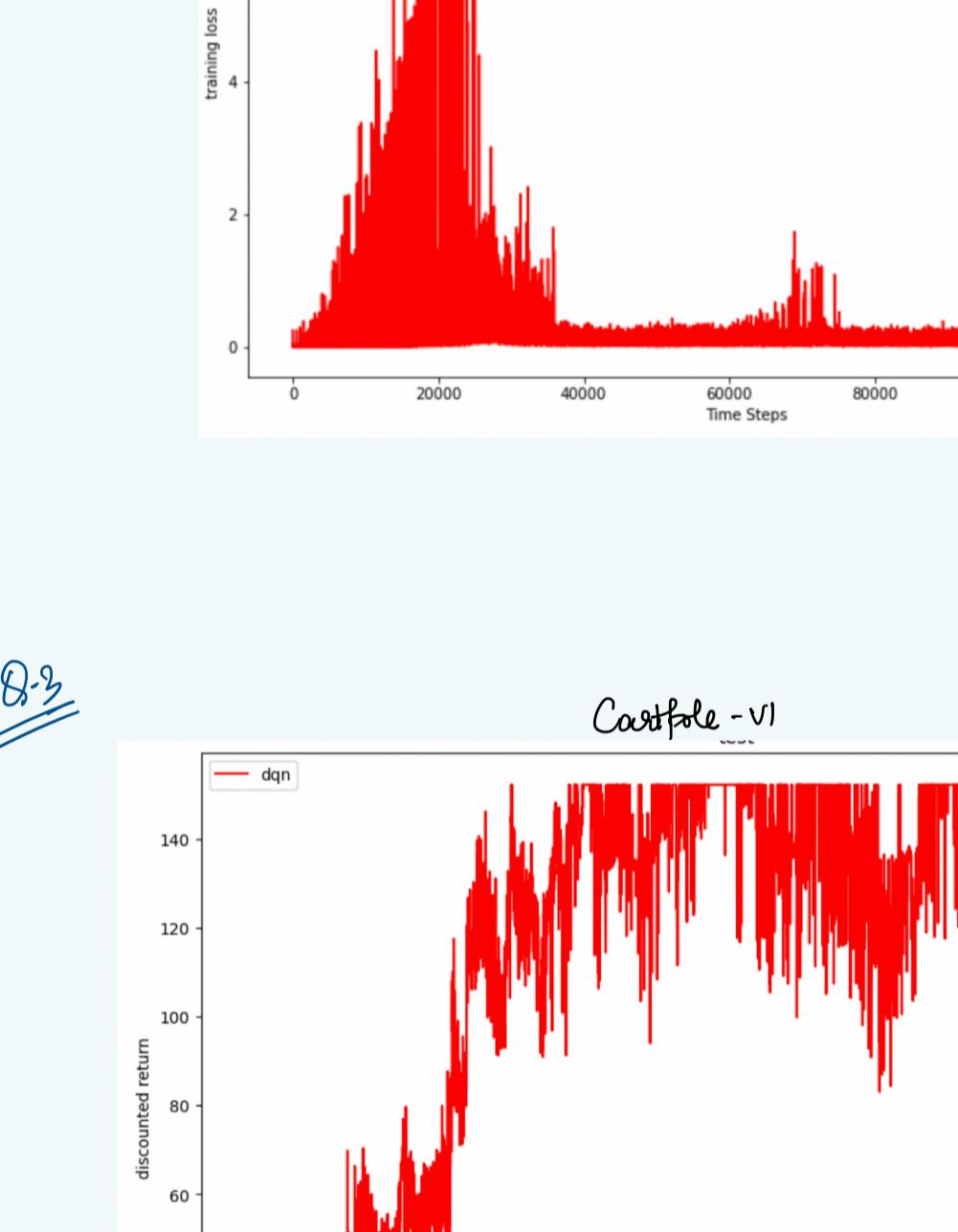
Plot &amp; dimension : 16



Plot &amp; dimension : 64



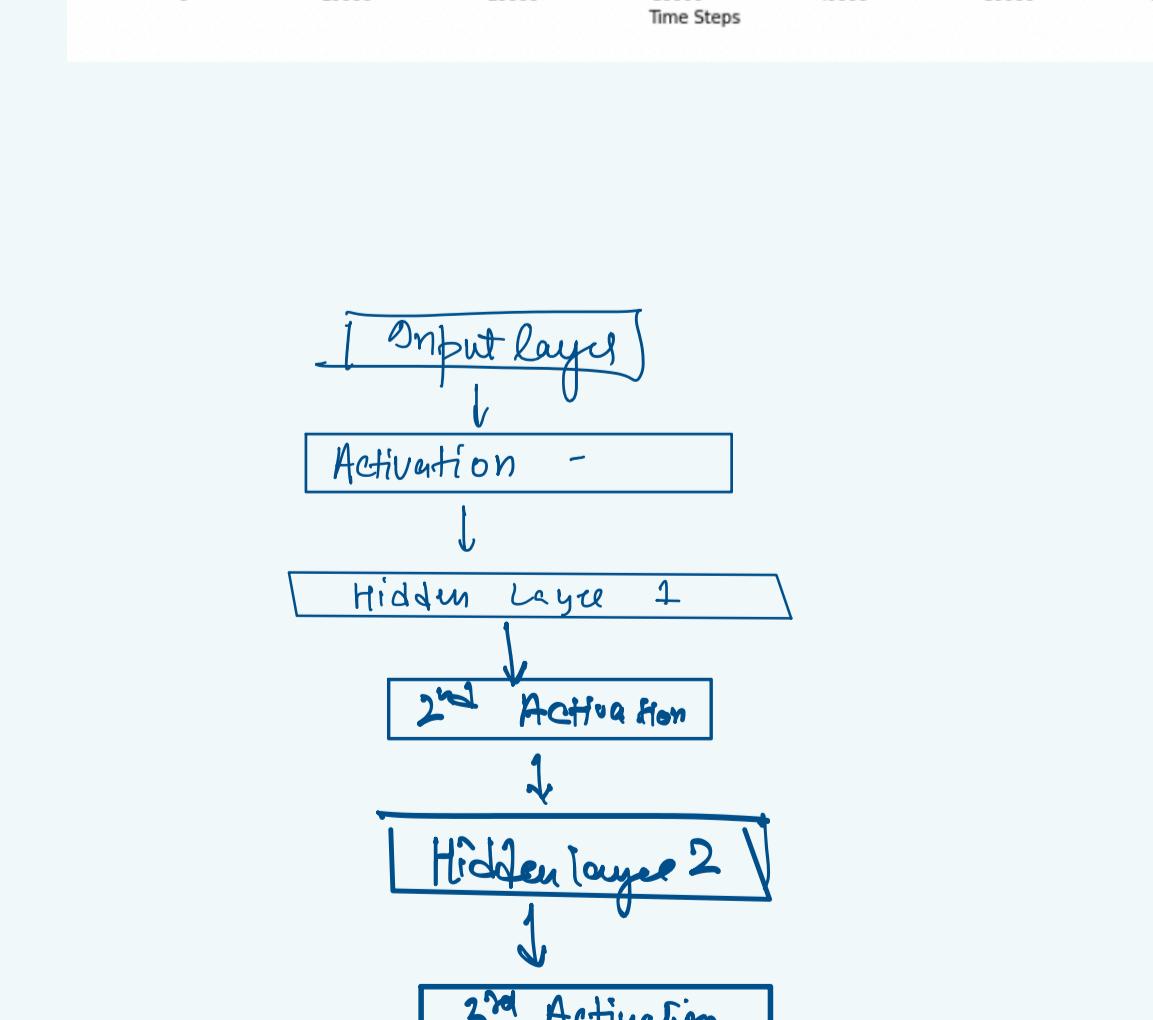
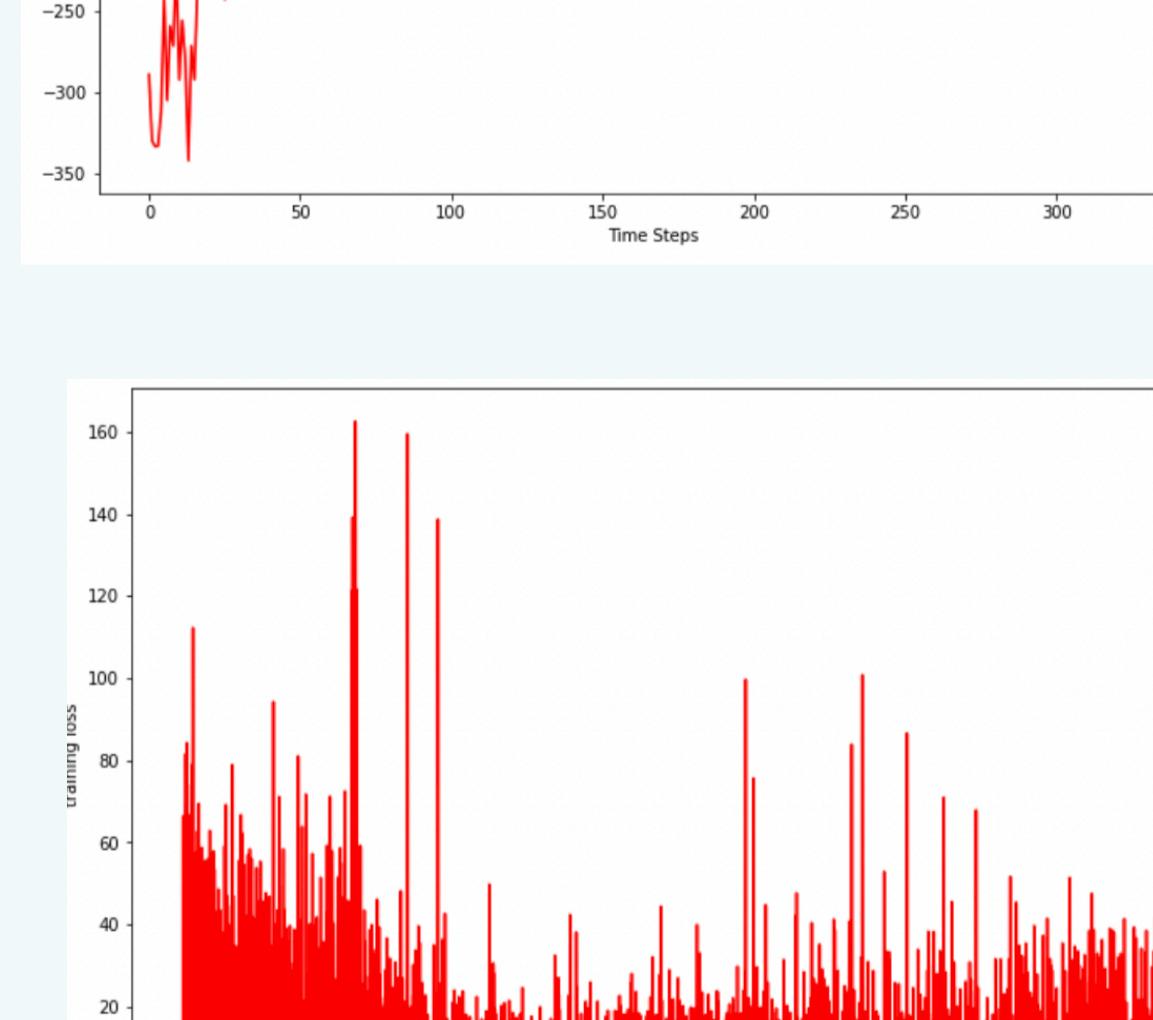
Plot &amp; dimension : 128



The Learned model seems pretty accurate especially for the case of within the range  $[-10, 10]$ , but for the cases outside the range the model seems to diverge from the Ground truth as can be clearly observed from the graphs.

The difference between the model having different layer dimensions is that there is different space square loss & the best result for the dimension 64. This can be possible as the more neural network dimension make the complexity & computation increase.

Q.2 The Plot averaged Due the 5 trials



Loss plot for Cartpole-v1



hyperparameters for Cartpole

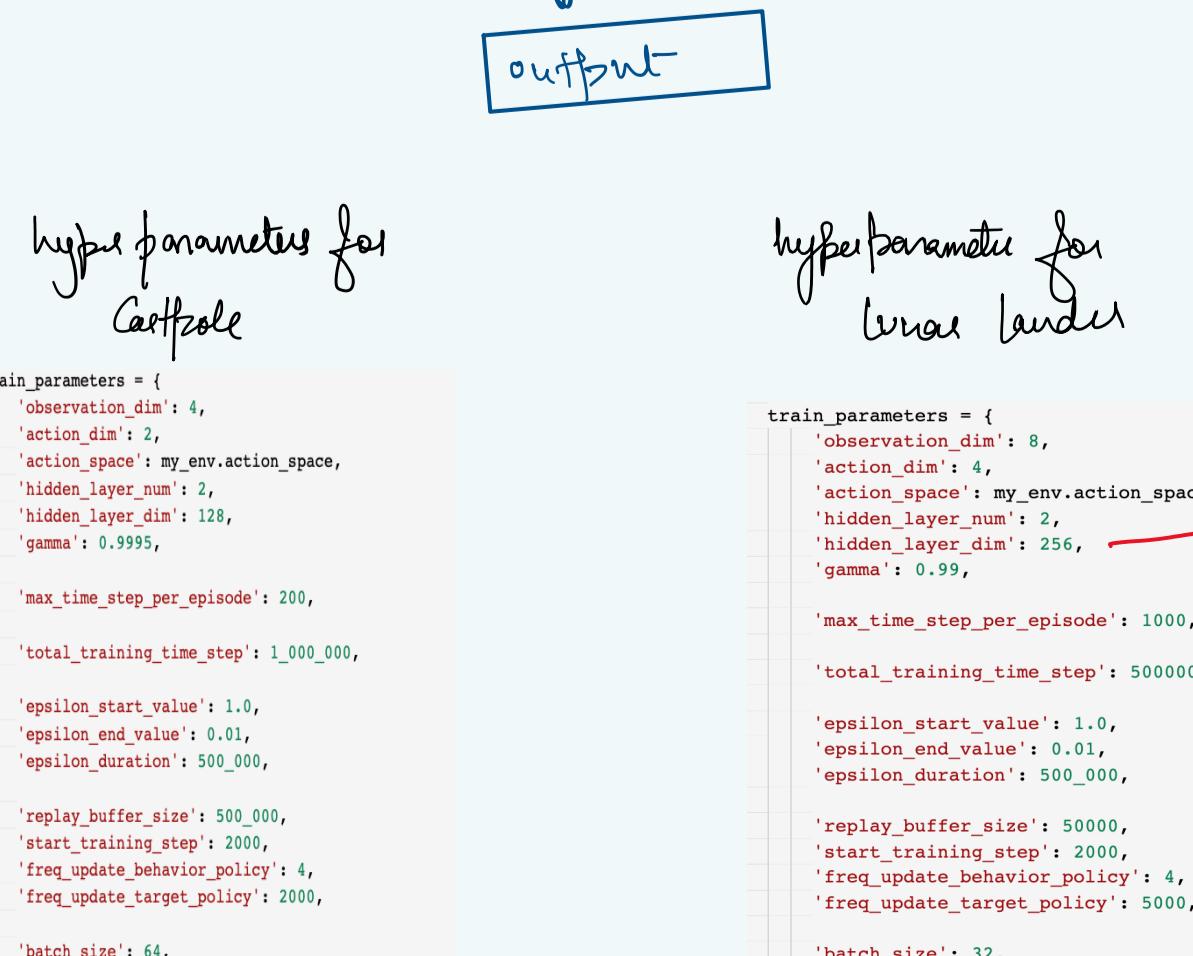
```
train_parameters = {
    'observation_dim': 4,
    'action_dim': 2,
    'action_space': 'my_env.action_space',
    'hidden_layer_num': 2,
    'hidden_layer_dim': 128,
    'gamma': 0.995,
    'max_time_step_per_episode': 200,
    'total_training_time_step': 1_000_000,
    'epsilon_start_value': 1.0,
    'epsilon_end_value': 0.01,
    'epsilon_duration': 500_000,
    'replay_buffer_size': 500_000,
    'start_training_step': 2000,
    'freq_update_behavior_policy': 4,
    'freq_update_target_policy': 2000,
    'batch_size': 64,
    'learning_rate': 1e-3,
    'model_name': 'cartpole_v1.pt'
}
```

hyperparameters for lunar lander

```
train_parameters = {
    'observation_dim': 8,
    'action_dim': 4,
    'action_space': 'my_env.action_space',
    'hidden_layer_num': 2,
    'hidden_layer_dim': 256,
    'gamma': 0.99,
    'max_time_step_per_episode': 1000,
    'total_training_time_step': 500000,
    'epsilon_start_value': 1.0,
    'epsilon_end_value': 0.01,
    'epsilon_duration': 500_000,
    'replay_buffer_size': 50000,
    'start_training_step': 2000,
    'freq_update_behavior_policy': 4,
    'freq_update_target_policy': 5000,
    'batch_size': 32,
    'learning_rate': 1e-4,
    'model_name': 'lunar_lander.pt'
}
```

Q.3

Cartpole - VI



layer dim  
target update frequency  
learning rate