In [1]:
```
!pip install torch
```

```
Requirement already satisfied: torch in c:\users\sanah quazi\anaconda3\lib\site-pa
ckages (2.0.1)
Requirement already satisfied: typing-extensions in c:\users\sanah quazi\anaconda3
\lib\site-packages (from torch) (4.1.1)
Requirement already satisfied: jinja2 in c:\users\sanah quazi\anaconda3\lib\site-p
ackages (from torch) (2.11.3)
Requirement already satisfied: networkx in c:\users\sanah quazi\anaconda3\lib\site
-packages (from torch) (2.7.1)
Requirement already satisfied: sympy in c:\users\sanah quazi\anaconda3\lib\site-pa
ckages (from torch) (1.10.1)
Requirement already satisfied: filelock in c:\users\sanah quazi\anaconda3\lib\site
-packages (from torch) (3.6.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\sanah quazi\anaconda3
\lib\site-packages (from jinja2->torch) (2.0.1)
Requirement already satisfied: mpmath>=0.19 in c:\users\sanah quazi\anaconda3\lib
\site-packages (from sympy->torch) (1.2.1)
WARNING: Ignoring invalid distribution -oogle-auth (c:\users\sanah quazi\anaconda3
\lib\site-packages)
WARNING: Ignoring invalid distribution -oogle-auth (c:\users\sanah quazi\anaconda3
\lib\site-packages)
WARNING: Ignoring invalid distribution -oogle-auth (c:\users\sanah quazi\anaconda3
\lib\site-packages)
WARNING: Ignoring invalid distribution -oogle-auth (c:\users\sanah quazi\anaconda3
\lib\site-packages)
WARNING: Ignoring invalid distribution -oogle-auth (c:\users\sanah quazi\anaconda3
\lib\site-packages)
WARNING: Ignoring invalid distribution -oogle-auth (c:\users\sanah quazi\anaconda3
\lib\site-packages)
```

In [2]:
```
import gym
import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim
```

In [4]:
```
import random
# Environment
env = gym.make('MountainCar-v0')
state_dim = env.observation_space.shape[0]
action_dim = env.action_space.n

# Q-network
class QNetwork(nn.Module):
    def __init__(self):
        super(QNetwork, self).__init__()
        self.fc1 = nn.Linear(state_dim, 64)
        self.fc2 = nn.Linear(64, 64)
        self.fc3 = nn.Linear(64, action_dim)

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
        x = self.fc3(x)
        return x

# Initialize Q-network and target network
q_network = QNetwork()
target_network = QNetwork()
target_network.load_state_dict(q_network.state_dict())
target_network.eval()
```

```
# Hyperparameters
lr = 0.001
gamma = 0.99
epsilon_start = 1.0
epsilon_end = 0.01
epsilon_decay = 500
batch_size = 32
target_update_freq = 100

# Optimizer and loss function
optimizer = optim.Adam(q_network.parameters(), lr=lr)
loss_fn = nn.MSELoss()

# Replay Buffer
class ReplayBuffer:
    def __init__(self, capacity):
        self.capacity = capacity
        self.buffer = []
        self.position = 0

    def push(self, transition):
        if len(self.buffer) < self.capacity:
            self.buffer.append(None)
        self.buffer[self.position] = transition
        self.position = (self.position + 1) % self.capacity

    def sample(self, batch_size):
        batch = random.sample(self.buffer, batch_size)
        states, actions, rewards, next_states = zip(*batch)
        return np.stack(states), actions, rewards, np.stack(next_states)

    def __len__(self):
        return len(self.buffer)

# Exploration-exploitation policy
def select_action(state, epsilon):
    if np.random.uniform() < epsilon:
        return env.action_space.sample()
    else:
        q_values = q_network(torch.tensor(state, dtype=torch.float32))
        return q_values.argmax().item()

# Training the model
def train_model():
    if len(replay_buffer) < batch_size:
        return

    states, actions, rewards, next_states = replay_buffer.sample(batch_size)

    states = torch.tensor(states, dtype=torch.float32)
    actions = torch.tensor(actions, dtype=torch.long)
    rewards = torch.tensor(rewards, dtype=torch.float32)
    next_states = torch.tensor(next_states, dtype=torch.float32)

    q_values = q_network(states)
    q_values = q_values.gather(1, actions.unsqueeze(1)).squeeze(1)

    next_q_values = target_network(next_states).max(1)[0].detach()

    target_q_values = rewards + gamma * next_q_values

    loss = loss_fn(q_values, target_q_values)
```

```
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

    return loss.item()

# Training loop
replay_buffer = ReplayBuffer(capacity=10000)
epsilon = epsilon_start
total_episodes = 100

for episode in range(total_episodes):
    state = env.reset()
    episode_reward = 0

    while True:
        action = select_action(state, epsilon)

        next_state, reward, done, _ = env.step(action)

        replay_buffer.push((state, action, reward, next_state))

        state = next_state
        episode_reward += reward

        loss = train_model()

        if done:
            break

    # Update epsilon
    epsilon = epsilon_end + (epsilon_start - epsilon_end) * np.exp(-episode / epsi

    # Update the target network
    if episode % target_update_freq == 0:
        target_network.load_state_dict(q_network.state_dict())

    # Print episode information
    print(f"Episode: {episode+1}, Reward: {episode_reward}, Loss: {loss}")

# Test the trained model
test_episodes = 10
test_rewards = []

for episode in range(test_episodes):
    state = env.reset()
    episode_reward = 0

    while True:
        action = select_action(state, 0.0)
        next_state, reward, done, _ = env.step(action)

        state = next_state
        episode_reward += reward

        if done:
            break

    test_rewards.append(episode_reward)

average_reward = np.mean(test_rewards)
print(f"Average Test Reward: {average_reward}")
```

```
Episode: 1, Reward: -200.0, Loss: 0.00017455198394600302
Episode: 2, Reward: -200.0, Loss: 0.00019691722991410643
Episode: 3, Reward: -200.0, Loss: 2.434561429254245e-05
Episode: 4, Reward: -200.0, Loss: 4.571350928017637e-06
Episode: 5, Reward: -200.0, Loss: 3.079863063248922e-06
Episode: 6, Reward: -200.0, Loss: 3.9102942537283525e-06
Episode: 7, Reward: -200.0, Loss: 3.888374976668274e-06
Episode: 8, Reward: -200.0, Loss: 2.4112205210258253e-06
Episode: 9, Reward: -200.0, Loss: 2.756554067673278e-06
Episode: 10, Reward: -200.0, Loss: 1.91167828234029e-06
Episode: 11, Reward: -200.0, Loss: 1.9607491594797466e-06
Episode: 12, Reward: -200.0, Loss: 2.145804501196835e-06
Episode: 13, Reward: -200.0, Loss: 1.6730791685404256e-06
Episode: 14, Reward: -200.0, Loss: 2.518107066862285e-06
Episode: 15, Reward: -200.0, Loss: 2.7418293484515743e-06
Episode: 16, Reward: -200.0, Loss: 4.484117198444437e-06
Episode: 17, Reward: -200.0, Loss: 2.3820180103939492e-06
Episode: 18, Reward: -200.0, Loss: 1.907607156681479e-06
Episode: 19, Reward: -200.0, Loss: 2.7924363621423254e-06
Episode: 20, Reward: -200.0, Loss: 1.9944316136388807e-06
Episode: 21, Reward: -200.0, Loss: 2.989972699651844e-06
Episode: 22, Reward: -200.0, Loss: 3.445294169068802e-06
Episode: 23, Reward: -200.0, Loss: 2.2327647002384765e-06
Episode: 24, Reward: -200.0, Loss: 4.694551307693473e-07
Episode: 25, Reward: -200.0, Loss: 4.874002570431912e-07
Episode: 26, Reward: -200.0, Loss: 8.223879603974638e-07
Episode: 27, Reward: -200.0, Loss: 5.511922154255444e-07
Episode: 28, Reward: -200.0, Loss: 1.7028942238539457e-06
Episode: 29, Reward: -200.0, Loss: 5.238201765678241e-07
Episode: 30, Reward: -200.0, Loss: 9.858619023361825e-07
Episode: 31, Reward: -200.0, Loss: 4.1724584320945723e-07
Episode: 32, Reward: -200.0, Loss: 4.971466864844842e-07
Episode: 33, Reward: -200.0, Loss: 1.201263330585789e-06
Episode: 34, Reward: -200.0, Loss: 3.0870847922415123e-07
Episode: 35, Reward: -200.0, Loss: 5.963595981484104e-07
Episode: 36, Reward: -200.0, Loss: 2.5966101020458154e-06
Episode: 37, Reward: -200.0, Loss: 2.2410490601032507e-06
Episode: 38, Reward: -200.0, Loss: 1.6570721754760598e-06
Episode: 39, Reward: -200.0, Loss: 2.515473397579626e-06
Episode: 40, Reward: -200.0, Loss: 3.401553385629086e-06
Episode: 41, Reward: -200.0, Loss: 1.8779475112751243e-06
Episode: 42, Reward: -200.0, Loss: 2.1485898571427242e-07
Episode: 43, Reward: -200.0, Loss: 1.6281448552035727e-05
Episode: 44, Reward: -200.0, Loss: 3.627663431871042e-07
Episode: 45, Reward: -200.0, Loss: 4.490413232360879e-07
Episode: 46, Reward: -200.0, Loss: 4.887617706117453e-07
Episode: 47, Reward: -200.0, Loss: 1.718001527706292e-07
Episode: 48, Reward: -200.0, Loss: 1.1264583008596674e-06
Episode: 49, Reward: -200.0, Loss: 2.5261880409743753e-07
Episode: 50, Reward: -200.0, Loss: 3.8399758750529145e-07
Episode: 51, Reward: -200.0, Loss: 1.494622438258375e-06
Episode: 52, Reward: -200.0, Loss: 1.2628822787519312e-07
Episode: 53, Reward: -200.0, Loss: 2.5904987523972522e-06
Episode: 54, Reward: -200.0, Loss: 6.825362561357906e-06
Episode: 55, Reward: -200.0, Loss: 7.611296268805745e-07
Episode: 56, Reward: -200.0, Loss: 2.812394086504355e-07
Episode: 57, Reward: -200.0, Loss: 4.82019117953314e-07
Episode: 58, Reward: -200.0, Loss: 2.6867203359870473e-06
Episode: 59, Reward: -200.0, Loss: 3.67697225556185e-07
Episode: 60, Reward: -200.0, Loss: 5.735311106036534e-07
Episode: 61, Reward: -200.0, Loss: 5.505864351107448e-07
Episode: 62, Reward: -200.0, Loss: 2.7663551009027287e-05
Episode: 63, Reward: -200.0, Loss: 2.429474079690408e-07
Episode: 64, Reward: -200.0, Loss: 4.593672201735899e-06
```

```
Episode: 65, Reward: -200.0, Loss: 3.5598924341684324e-07
Episode: 66, Reward: -200.0, Loss: 5.819468924528337e-07
Episode: 67, Reward: -200.0, Loss: 7.539600801464985e-07
Episode: 68, Reward: -200.0, Loss: 6.794356295358739e-07
Episode: 69, Reward: -200.0, Loss: 9.720977800498076e-08
Episode: 70, Reward: -200.0, Loss: 0.0001580979151185602
Episode: 71, Reward: -200.0, Loss: 1.1603015082073398e-06
Episode: 72, Reward: -200.0, Loss: 4.005835307907546e-06
Episode: 73, Reward: -200.0, Loss: 3.192668316387426e-07
Episode: 74, Reward: -200.0, Loss: 8.280824204121018e-08
Episode: 75, Reward: -200.0, Loss: 2.8530030249385163e-07
Episode: 76, Reward: -200.0, Loss: 4.743658337247325e-07
Episode: 77, Reward: -200.0, Loss: 1.1220370765840926e-07
Episode: 78, Reward: -200.0, Loss: 9.118728485191241e-06
Episode: 79, Reward: -200.0, Loss: 5.984561084915185e-06
Episode: 80, Reward: -200.0, Loss: 2.0176701127638808e-06
Episode: 81, Reward: -200.0, Loss: 6.931623488526384e-07
Episode: 82, Reward: -200.0, Loss: 5.814256837766152e-06
Episode: 83, Reward: -200.0, Loss: 1.5338508774220827e-07
Episode: 84, Reward: -200.0, Loss: 2.4569697387732958e-08
Episode: 85, Reward: -200.0, Loss: 1.0346636969416068e-07
Episode: 86, Reward: -200.0, Loss: 4.910010034109291e-07
Episode: 87, Reward: -200.0, Loss: 1.4531149190588621e-06
Episode: 88, Reward: -200.0, Loss: 2.901888819906162e-06
Episode: 89, Reward: -200.0, Loss: 1.9281881122878985e-06
Episode: 90, Reward: -200.0, Loss: 8.860058642312652e-07
Episode: 91, Reward: -200.0, Loss: 1.619875291680728e-07
Episode: 92, Reward: -200.0, Loss: 2.102268297221599e-07
Episode: 93, Reward: -200.0, Loss: 2.3569687073177192e-06
Episode: 94, Reward: -200.0, Loss: 2.4496952732988575e-07
Episode: 95, Reward: -200.0, Loss: 1.4939577340555843e-05
Episode: 96, Reward: -200.0, Loss: 1.4105093214311637e-05
Episode: 97, Reward: -200.0, Loss: 1.8681680558074731e-06
Episode: 98, Reward: -200.0, Loss: 2.903818767663324e-07
Episode: 99, Reward: -200.0, Loss: 3.729534364538267e-05
Episode: 100, Reward: -200.0, Loss: 1.988005124076153e-06
Average Test Reward: -200.0
```

In [ ]: