Actor-Critic Deep Reinforcement Learning: Inverted Pendulum on a Cart

Source Filename: /main.py

Rico A. R. Picone

Problem Statement

Write a deep reinforcement learning program to learn a policy network $\pi_\theta(s, a)$ to regulate the angular position of an inverted pendulum on a cart to be vertical. Use two networks, an actor (policy π_θ) and a critic (value $V_\phi(s)$). Use a temporal difference TD(0) method to update the networks. Use the Gymnasium cart-pole environment. Note that this has a discrete action space, able to push the cart with a force a unit leftward (0) or rightward (1). Train for several episodes and display the results.

Load Packages and Set Script Parameters

Begin by importing the necessary packages as follows:

```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import gymnasium as gym
import tensorflow as tf
import time
import animation  # Import the animation module (animation.py)
```

Set script parameters as follows:

```python
retrain = True  # Retrain the policy and value networks
save = True  # Save the rewards and policy and value networks
visualize = False  # TODO fix True Visualize the cart/pole environment
file_rewards = 'rewards.npy'  # File to save the rewards
file_policy = 'policy_network.keras'  # File to save the policy network
file_value = 'value_network.keras'  # File to save the value network
n_episodes = 300  # Number of episodes to train from
```

Set Up the Cart-Pole Environment

Set up the cart-pole environment as follows:

```
if visualize:
    env = gym.make('CartPole-v1', render_mode='human')  # Create environment
else:
    env = gym.make('CartPole-v1', render_mode=None)  # Create environment
env.reset()  # Reset the environment to the initial state
```

```
(array([-0.01711824, -0.01674245,  0.00248054, -0.00782781], dtype=float32),
 {})
```

Print out the action space and observation space:

```
print("Action space:", env.action_space)
print("Observation space:", env.observation_space)
```

```
Action space: Discrete(2)
Observation space: Box([-4.8000002e+00 -3.4028235e+38 -4.1887903e-01 -3.4028235e+38], [4.80
```

Test the environment by taking random actions for an episode and visualizing the results:

```
for i in range(20):
    if visualize:
        time.sleep(0.03)  # Pause loop for rendering
    s_new, r_new, term, trunc, info = env.step(
        env.action_space.sample()
    )  # Take a random action
    print("Step", i, "State", s_new, "Reward", r_new)
    if term:
        break
```

```
Step 0 State [-0.01745309  0.17834383  0.00232398 -0.29972708] Reward 1.0
Step 1 State [-0.01388621 -0.01681116 -0.00367056 -0.00631211] Reward 1.0
Step 2 State [-0.01422243 -0.21188028 -0.0037968   0.28521046] Reward 1.0
Step 3 State [-0.01846004 -0.40694788  0.00190741  0.5766935 ] Reward 1.0
Step 4 State [-0.026599   -0.21185271  0.01344128  0.28461206] Reward 1.0
Step 5 State [-0.03083605 -0.01692502  0.01913352 -0.00380146] Reward 1.0
Step 6 State [-0.03117455  0.17791738  0.01905749 -0.29038665] Reward 1.0
Step 7 State [-0.0276162   0.37276247  0.01324976 -0.57699865] Reward 1.0
Step 8 State [-0.02016095  0.5676962   0.00170978 -0.8654783 ] Reward 1.0
Step 9 State [-0.00880703  0.37255105 -0.01559978 -0.5722583 ] Reward 1.0
Step 10 State [-0.00135601  0.17765127 -0.02704495 -0.28453034] Reward 1.0
Step 11 State [ 0.00219702 -0.01707473 -0.03273555 -0.00049841] Reward 1.0
Step 12 State [ 0.00185552  0.17850104 -0.03274552 -0.3033274 ] Reward 1.0
Step 13 State [ 0.00542554 -0.01613931 -0.03881207 -0.02114888] Reward 1.0
Step 14 State [ 0.00510276  0.17951714 -0.03923505 -0.32582042] Reward 1.0
Step 15 State [ 0.0086931  -0.01502487 -0.04575146 -0.04576413] Reward 1.0
Step 16 State [ 0.0083926   0.18072225 -0.04666674 -0.3525238 ] Reward 1.0
Step 17 State [ 0.01200705  0.3764757  -0.05371721 -0.6595493 ] Reward 1.0
Step 18 State [ 0.01953656  0.57230246 -0.0669082  -0.968651  ] Reward 1.0
Step 19 State [ 0.03098261  0.3781395  -0.08628122 -0.69771457] Reward 1.0
```

Define the Policy and Value Networks and their Optimizers

Define a function that defines the architecture and compiles the policy network as follows:

```python
def policy_network_com(optimizer):
    """Define and compile the policy network.

    Input layer: 32 units, ReLU activation, input shape of observation space
    Output layer: 2 units, softmax activation, output shape of action space

    Returns:
    Compiled policy network
    """
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Input(env.observation_space.shape))
    model.add(tf.keras.layers.Dense(32, activation='relu'))
    model.add(tf.keras.layers.Dense(env.action_space.n, activation='softmax'))
    model.compile(optimizer=optimizer, loss='categorical_crossentropy')
    return model
```

Define a function that defines the architecture and compiles the value network as follows:

```python
def value_network_com(optimizer):
    """Define and compile the value network.

    Input layer: 32 units, ReLU activation, input shape of observation space
    Output layer: 1 unit, linear activation, output shape of 1 (value)
    """
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Input(env.observation_space.shape))
    model.add(tf.keras.layers.Dense(32, activation='relu'))
    model.add(tf.keras.layers.Dense(1, activation='linear'))
    model.compile(optimizer=optimizer, loss='mean_squared_error')
    return model
```

Define the optimizers for the policy and value networks:

```python
optimizer_policy = tf.keras.optimizers.Adam(learning_rate=0.001)
optimizer_value = tf.keras.optimizers.Adam(learning_rate=0.001)
```

Define the policy and value networks:

```
policy_network = policy_network_com(optimizer_policy)
value_network = value_network_com(optimizer_value)
```

Train the Policy and Value Networks

Define a function to update the policy and value networks using a temporal difference TD(0) method as follows:

```
def updater(policy_network, value_network, state, action, state_next, reward, gamma):
    """Update the policy and value networks using a temporal difference TD(0) method.

    Args:
    policy_network (keras.models.Sequential): Policy network
    value_network (keras.models.Sequential): Value network
    state (numpy.ndarray): State
    action (int): Action
    state_next (numpy.ndarray): Next state
    reward (float): Reward at old state
    gamma (float): Discount factor

    Returns:
    policy_network: Updated policy network
    value_network: Updated value network
    """
    with tf.GradientTape(persistent=True) as tape:
        tf.keras.utils.disable_interactive_logging()  # Suppress bars
        state = state.reshape(1, -1)  # 2D array
        state_next = state_next.reshape(1, -1)  # 2D array

        # TD(0) error
        value = value_network(state)  # Value of the current state
        value_next = value_network(state_next)  # Value of the next state
        td_target = reward + gamma * value_next  # TD(0) target
        td_error = td_target - value  # TD(0) error

        # Losses
        action_prob = policy_network(state)[0, action]  # Prob of action taken
        log_prob = tf.math.log(action_prob)  # Log prob of action taken (negative)
        policy_loss = -log_prob * td_error  # Policy loss
        value_loss = tf.square(td_error)  # Value loss (MSE)

        # Gradients
        policy_grads = tape.gradient(
            policy_loss, policy_network.trainable_variables
        )  # Gradient of the loss function wrt policy network parameters
        value_grads = tape.gradient(
            value_loss, value_network.trainable_variables
        )  # Gradient of the loss function wrt value network parameters

        # Clip gradients
        policy_grads = tf.clip_by_global_norm(policy_grads, 10.0)[0]
        value_grads = tf.clip_by_global_norm(value_grads, 10.0)[0]

        # Apply gradients
        optimizer_policy.apply_gradients(
            zip(policy_grads, policy_network.trainable_variables)
        )  # Update the policy network parameters
        optimizer_value.apply_gradients(
            zip(value_grads, value_network.trainable_variables)
        )  # Update the value network parameters

    return policy_network, value_network
```

In `animation.py`, we define two animation classes to monitor the training. The first, `Angle_policy_animation`, plots a sample of the policy network for the pole (pendulum) angle with all other states set to 0. Our intuition is that the policy network should output a higher probability of pushing the cart in the direction that will bring the pole closer to the vertical position (i.e., leftward if the pole is leaning left and rightward if the pole is leaning right). The second, `Angle_value_animation`, plots a sample of the value network for the pole (pendulum) angle with all other states set to 0. Here our intuition is that the value network should output a higher value for the pole angle closer to the vertical position.

We are now ready to define a function to train the policy and value networks using a temporal difference TD(0) method (i.e., the `updater()` function we defined) as follows:

```
def train_policy_value_networks(
        env, policy_network, value_network, updater,
        n_episodes=1000, gamma=0.99, max_episode_steps=100,
        plot=False, print_=False
):
    """Train policy and value networks using a temporal difference TD(0) method

    Update the policy and value networks using the updater() function.

    Args:
    policy_network (keras.models.Sequential): Policy network
    value_network (keras.models.Sequential): Value network
    n_episodes (int): Number of episodes to train from
    gamma (float): Discount factor
    max_episode_steps (int): Maximum number of steps per episode
    Returns:
    policy_network: Trained policy network
    value_network: Trained value network
    rewards: Numpy array of episode rewards
    """
    rewards = []
    if plot:
        animation_policy = animation.Angle_policy_animation(policy_network).animate()
        animation_value = animation.Angle_value_animation(value_network).animate()
    for episode in range(n_episodes):
        if plot:
            animation_policy.policy_network = policy_network
            plt.figure(animation_policy._fig.number)  # Activate figure
            plt.pause(0.01)  # Pause for rendering
            animation_value.value_network = value_network
            plt.figure(animation_value._fig.number)  # Activate figure
            plt.pause(0.01)  # Pause for rendering
        state, _ = env.reset()  # Reset env (random initial state)
        episode_rewards = []
        for _ in range(max_episode_steps):
            prob = policy_network(state.reshape((1, -1))).numpy()[0]
                # Get action probabilities from policy network
            action = np.random.choice(env.action_space.n, p=prob)
                # Sample action from policy network
            state_new, reward, term, trunc, _ = env.step(action)  # Enact
            episode_rewards.append(reward)
            policy_network, value_network = updater(
                policy_network, value_network,
                state, action, state_new,
                reward, gamma
            )  # Update the policy and value networks
            if term or trunc:
                break
            state = state_new
        if print_:
            print(f"Episode {episode + 1}/{n_episodes}. "
                  f"Reward sum: {sum(episode_rewards)}"
                  f"\t{int(sum(episode_rewards)/500*100)*'='}"
            )
        rewards.append(sum(episode_rewards))
    return policy_network, value_network, rewards
```

Train the policy and value networks

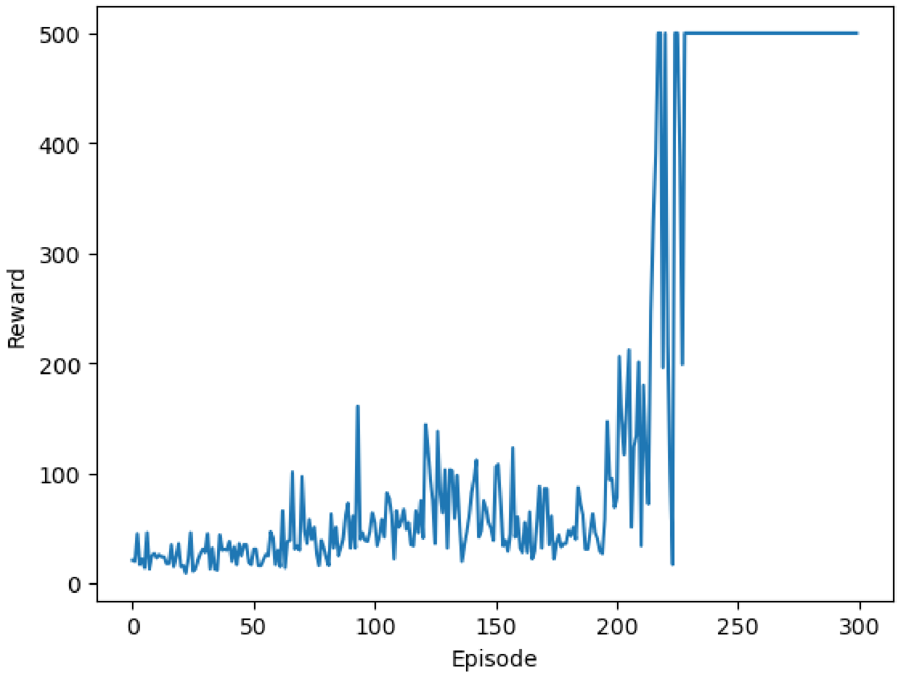
```
if retrain:
    policy_network, value_network, rewards = train_policy_value_networks(
        env, policy_network, value_network, updater,
        n_episodes=n_episodes, gamma=0.99, max_episode_steps=500,
        plot=False, print_=True
    )
    if save:
        policy_network.save(file_policy)  # Save the policy network
        value_network.save(file_value)  # Save the value network
        np.save(file_rewards, rewards)  # Save the rewards
else:
    policy_network = tf.keras.models.load_model(file_policy)
    value_network = tf.keras.models.load_model(file_value)
    rewards = np.load(file_rewards)
```

```
WARNING:tensorflow:Calling GradientTape.gradient on a persistent tape inside its context is

WARNING:tensorflow:Calling GradientTape.gradient on a persistent tape inside its context is

Episode 1/300. Reward sum: 21.0 ====

Episode 2/300. Reward sum: 20.0 ====

Episode 3/300. Reward sum: 45.0 =========

Episode 4/300. Reward sum: 17.0 ===

Episode 5/300. Reward sum: 22.0 ====

Episode 6/300. Reward sum: 14.0 ==

Episode 7/300. Reward sum: 46.0 =========

Episode 8/300. Reward sum: 13.0 ==

Episode 9/300. Reward sum: 25.0 =====

Episode 10/300. Reward sum: 27.0 =====

Episode 11/300. Reward sum: 23.0 ====

Episode 12/300. Reward sum: 26.0 =====

Episode 13/300. Reward sum: 24.0 ====

Episode 14/300. Reward sum: 24.0 ====

Episode 15/300. Reward sum: 18.0 ===

Episode 16/300. Reward sum: 18.0 ===

Episode 17/300. Reward sum: 35.0 =======

Episode 18/300. Reward sum: 15.0 ===

Episode 19/300. Reward sum: 24.0 ====

Episode 20/300. Reward sum: 36.0 =======
Episode 21/300. Reward sum: 15.0 ===

Episode 22/300. Reward sum: 16.0 ===
Episode 23/300. Reward sum: 9.0 =

Episode 24/300. Reward sum: 22.0 ====

Episode 25/300. Reward sum: 46.0 =========
Episode 26/300. Reward sum: 11.0 ==

Episode 27/300. Reward sum: 13.0 ==

Episode 28/300. Reward sum: 21.0 ====

Episode 29/300. Reward sum: 27.0 =====

Episode 30/300. Reward sum: 31.0 ======

Episode 31/300. Reward sum: 28.0 =====

Episode 32/300. Reward sum: 45.0 =========

Episode 33/300. Reward sum: 13.0 ==

Episode 34/300. Reward sum: 32.0 ======

Episode 35/300. Reward sum: 13.0 ==

Episode 36/300. Reward sum: 12.0 ==

Episode 37/300. Reward sum: 44.0 ========

Episode 38/300. Reward sum: 30.0 ======

Episode 39/300. Reward sum: 31.0 ======

Episode 40/300. Reward sum: 30.0 ======

Episode 41/300. Reward sum: 38.0 =======

Episode 42/300. Reward sum: 20.0 ====

Episode 43/300. Reward sum: 33.0 ======

Episode 44/300. Reward sum: 17.0 ===

Episode 45/300. Reward sum: 36.0 =======

Episode 46/300. Reward sum: 25.0 =====

Episode 47/300. Reward sum: 35.0 =======

Episode 48/300. Reward sum: 35.0 =======

Episode 49/300. Reward sum: 19.0 ===

Episode 50/300. Reward sum: 17.0 ===

Episode 51/300. Reward sum: 31.0 ======
Episode 52/300. Reward sum: 31.0 ======
Episode 53/300. Reward sum: 16.0 ===
Episode 54/300. Reward sum: 16.0 ===
Episode 55/300. Reward sum: 21.0 ====
Episode 56/300. Reward sum: 26.0 =====
Episode 57/300. Reward sum: 25.0 =====
Episode 58/300. Reward sum: 47.0 =========
Episode 59/300. Reward sum: 41.0 ========
Episode 60/300. Reward sum: 17.0 ===
Episode 61/300. Reward sum: 30.0 ======
Episode 62/300. Reward sum: 15.0 ===
Episode 63/300. Reward sum: 66.0 =============
Episode 64/300. Reward sum: 14.0 ==
Episode 65/300. Reward sum: 38.0 =======
Episode 66/300. Reward sum: 39.0 =======
Episode 67/300. Reward sum: 101.0 ====================
Episode 68/300. Reward sum: 31.0 ======
Episode 69/300. Reward sum: 34.0 ======
Episode 70/300. Reward sum: 30.0 ======
Episode 71/300. Reward sum: 97.0 ===================
Episode 72/300. Reward sum: 49.0 =========
Episode 73/300. Reward sum: 36.0 =======
Episode 74/300. Reward sum: 58.0 ===========
Episode 75/300. Reward sum: 40.0 ========
Episode 76/300. Reward sum: 51.0 ==========
Episode 77/300. Reward sum: 27.0 =====
Episode 78/300. Reward sum: 16.0 ===
Episode 79/300. Reward sum: 39.0 =======
Episode 80/300. Reward sum: 32.0 ======
Episode 81/300. Reward sum: 23.0 ====
Episode 82/300. Reward sum: 16.0 ===
Episode 83/300. Reward sum: 63.0 ============
Episode 84/300. Reward sum: 32.0 ======
Episode 85/300. Reward sum: 51.0 ==========
Episode 86/300. Reward sum: 25.0 =====
Episode 87/300. Reward sum: 32.0 ======
Episode 88/300. Reward sum: 41.0 ========
Episode 89/300. Reward sum: 62.0 ============
Episode 90/300. Reward sum: 73.0 ==============
Episode 91/300. Reward sum: 32.0 ======
Episode 92/300. Reward sum: 61.0 ============
Episode 93/300. Reward sum: 32.0 ======
Episode 94/300. Reward sum: 161.0 ================================
Episode 95/300. Reward sum: 40.0 ========
Episode 96/300. Reward sum: 46.0 =========
Episode 97/300. Reward sum: 39.0 =======
Episode 98/300. Reward sum: 38.0 =======
Episode 99/300. Reward sum: 47.0 =========
Episode 100/300. Reward sum: 64.0 ============
Episode 101/300. Reward sum: 55.0 ===========
Episode 102/300. Reward sum: 34.0 ======
Episode 103/300. Reward sum: 42.0 ========
Episode 104/300. Reward sum: 58.0 ===========
Episode 105/300. Reward sum: 42.0 ========
Episode 106/300. Reward sum: 82.0 ================
Episode 107/300. Reward sum: 77.0 ===============
Episode 108/300. Reward sum: 62.0 ============
Episode 109/300. Reward sum: 22.0 ====
Episode 110/300. Reward sum: 66.0 =============
Episode 111/300. Reward sum: 51.0 ==========
Episode 112/300. Reward sum: 56.0 ===========
Episode 113/300. Reward sum: 67.0 =============
Episode 114/300. Reward sum: 49.0 =========
Episode 115/300. Reward sum: 55.0 ===========
Episode 116/300. Reward sum: 35.0 =======
Episode 117/300. Reward sum: 34.0 ======
Episode 118/300. Reward sum: 66.0 =============
Episode 119/300. Reward sum: 46.0 =========
Episode 120/300. Reward sum: 75.0 ===============
Episode 121/300. Reward sum: 41.0 ========
Episode 122/300. Reward sum: 144.0 ============================
Episode 123/300. Reward sum: 122.0 ========================
Episode 124/300. Reward sum: 95.0 ===================
Episode 125/300. Reward sum: 75.0 ===============
Episode 126/300. Reward sum: 36.0 =======
Episode 127/300. Reward sum: 138.0 ===========================
Episode 128/300. Reward sum: 82.0 ================
Episode 129/300. Reward sum: 64.0 ============
Episode 130/300. Reward sum: 103.0 ====================
Episode 131/300. Reward sum: 32.0 ======
Episode 132/300. Reward sum: 103.0 ====================
Episode 133/300. Reward sum: 102.0 ====================
Episode 134/300. Reward sum: 59.0 ===========
Episode 135/300. Reward sum: 98.0 ===================
Episode 136/300. Reward sum: 55.0 ===========
Episode 137/300. Reward sum: 20.0 ====
Episode 138/300. Reward sum: 35.0 =======
Episode 139/300. Reward sum: 47.0 =========
Episode 140/300. Reward sum: 61.0 ============
Episode 141/300. Reward sum: 82.0 ================
Episode 142/300. Reward sum: 95.0 ===================
Episode 143/300. Reward sum: 112.0 ======================
Episode 144/300. Reward sum: 42.0 ========
Episode 145/300. Reward sum: 48.0 =========
Episode 146/300. Reward sum: 75.0 ===============
Episode 147/300. Reward sum: 68.0 =============
Episode 148/300. Reward sum: 55.0 ===========
Episode 149/300. Reward sum: 50.0 ==========
Episode 150/300. Reward sum: 39.0 =======
Episode 151/300. Reward sum: 106.0 =====================
Episode 152/300. Reward sum: 108.0 =====================
Episode 153/300. Reward sum: 74.0 ==============
Episode 154/300. Reward sum: 34.0 ======
Episode 155/300. Reward sum: 39.0 =======
Episode 156/300. Reward sum: 29.0 =====
Episode 157/300. Reward sum: 45.0 =========
Episode 158/300. Reward sum: 123.0 ========================
Episode 159/300. Reward sum: 42.0 ========
Episode 160/300. Reward sum: 60.0 ============
Episode 161/300. Reward sum: 32.0 ======
Episode 162/300. Reward sum: 28.0 =====
Episode 163/300. Reward sum: 55.0 ===========
Episode 164/300. Reward sum: 28.0 =====
Episode 165/300. Reward sum: 65.0 =============
Episode 166/300. Reward sum: 22.0 ====
Episode 167/300. Reward sum: 29.0 =====
Episode 168/300. Reward sum: 56.0 ===========
Episode 169/300. Reward sum: 88.0 =================
Episode 170/300. Reward sum: 32.0 ======
Episode 171/300. Reward sum: 86.0 =================
Episode 172/300. Reward sum: 86.0 =================
Episode 173/300. Reward sum: 35.0 =======
Episode 174/300. Reward sum: 61.0 ============
Episode 175/300. Reward sum: 22.0 ====
Episode 176/300. Reward sum: 36.0 =======
Episode 177/300. Reward sum: 44.0 ========
Episode 178/300. Reward sum: 33.0 ======
Episode 179/300. Reward sum: 36.0 =======
Episode 180/300. Reward sum: 36.0 =======
Episode 181/300. Reward sum: 48.0 =========
Episode 182/300. Reward sum: 43.0 ========
Episode 183/300. Reward sum: 51.0 ==========
Episode 184/300. Reward sum: 40.0 ========
Episode 185/300. Reward sum: 87.0 =================
Episode 186/300. Reward sum: 72.0 ==============
Episode 187/300. Reward sum: 62.0 ============
Episode 188/300. Reward sum: 31.0 ======
Episode 189/300. Reward sum: 31.0 ======
Episode 190/300. Reward sum: 47.0 =========
Episode 191/300. Reward sum: 63.0 ============
Episode 192/300. Reward sum: 46.0 =========
Episode 193/300. Reward sum: 40.0 ========
Episode 194/300. Reward sum: 29.0 =====
Episode 195/300. Reward sum: 27.0 =====
Episode 196/300. Reward sum: 58.0 ===========
Episode 197/300. Reward sum: 147.0 =============================
Episode 198/300. Reward sum: 94.0 ==================
Episode 199/300. Reward sum: 95.0 ===================
Episode 200/300. Reward sum: 69.0 =============
Episode 201/300. Reward sum: 78.0 ===============
Episode 202/300. Reward sum: 206.0 =========================================
Episode 203/300. Reward sum: 149.0 =============================
Episode 204/300. Reward sum: 117.0 =======================
Episode 205/300. Reward sum: 165.0 =================================
Episode 206/300. Reward sum: 212.0 ==========================================
Episode 207/300. Reward sum: 51.0 ==========
Episode 208/300. Reward sum: 124.0 ========================
Episode 209/300. Reward sum: 134.0 ==========================
Episode 210/300. Reward sum: 201.0 ========================================
Episode 211/300. Reward sum: 34.0 ======
Episode 212/300. Reward sum: 180.0 ====================================
Episode 213/300. Reward sum: 120.0 ========================
Episode 214/300. Reward sum: 72.0 ==============
Episode 215/300. Reward sum: 246.0 =================================================
Episode 216/300. Reward sum: 328.0 ========================================================
Episode 217/300. Reward sum: 391.0 ========================================================
Episode 218/300. Reward sum: 500.0 ========================================================
Episode 219/300. Reward sum: 500.0 ========================================================
Episode 220/300. Reward sum: 196.0 =======================================
Episode 221/300. Reward sum: 500.0 ========================================================
Episode 222/300. Reward sum: 217.0 ===========================================
Episode 223/300. Reward sum: 100.0 ====================
Episode 224/300. Reward sum: 17.0 ===
Episode 225/300. Reward sum: 500.0 ========================================================
Episode 226/300. Reward sum: 500.0 ========================================================
Episode 227/300. Reward sum: 388.0 ========================================================
Episode 228/300. Reward sum: 199.0 =======================================
Episode 229/300. Reward sum: 500.0 ========================================================
Episode 230/300. Reward sum: 500.0 ========================================================
Episode 231/300. Reward sum: 500.0 ========================================================
Episode 232/300. Reward sum: 500.0 ========================================================
Episode 233/300. Reward sum: 500.0 ========================================================
Episode 234/300. Reward sum: 500.0 ========================================================
Episode 235/300. Reward sum: 500.0 ========================================================
Episode 236/300. Reward sum: 500.0 ========================================================
Episode 237/300. Reward sum: 500.0 ========================================================
Episode 238/300. Reward sum: 500.0 ========================================================
Episode 239/300. Reward sum: 500.0 ========================================================
Episode 240/300. Reward sum: 500.0 ========================================================
Episode 241/300. Reward sum: 500.0 ========================================================
Episode 242/300. Reward sum: 500.0 ========================================================
Episode 243/300. Reward sum: 500.0 ========================================================
Episode 244/300. Reward sum: 500.0 ========================================================
Episode 245/300. Reward sum: 500.0 ========================================================
Episode 246/300. Reward sum: 500.0 ========================================================
Episode 247/300. Reward sum: 500.0 ========================================================
Episode 248/300. Reward sum: 500.0 ========================================================
Episode 249/300. Reward sum: 500.0 ========================================================
Episode 250/300. Reward sum: 500.0 ========================================================
Episode 251/300. Reward sum: 500.0 ========================================================
Episode 252/300. Reward sum: 500.0 ========================================================
Episode 253/300. Reward sum: 500.0 ========================================================
Episode 254/300. Reward sum: 500.0 ========================================================
Episode 255/300. Reward sum: 500.0 ========================================================
Episode 256/300. Reward sum: 500.0 ========================================================
Episode 257/300. Reward sum: 500.0 ========================================================
Episode 258/300. Reward sum: 500.0 ========================================================
Episode 259/300. Reward sum: 500.0 ========================================================
Episode 260/300. Reward sum: 500.0 ========================================================
Episode 261/300. Reward sum: 500.0 ========================================================
Episode 262/300. Reward sum: 500.0 ========================================================
Episode 263/300. Reward sum: 500.0 ========================================================
Episode 264/300. Reward sum: 500.0 ========================================================
Episode 265/300. Reward sum: 500.0 ========================================================
Episode 266/300. Reward sum: 500.0 ========================================================
Episode 267/300. Reward sum: 500.0 ========================================================
Episode 268/300. Reward sum: 500.0 ========================================================
Episode 269/300. Reward sum: 500.0 ========================================================
Episode 270/300. Reward sum: 500.0 ========================================================
Episode 271/300. Reward sum: 500.0 ========================================================
Episode 272/300. Reward sum: 500.0 ========================================================
Episode 273/300. Reward sum: 500.0 ========================================================
Episode 274/300. Reward sum: 500.0 ========================================================
Episode 275/300. Reward sum: 500.0 ========================================================
Episode 276/300. Reward sum: 500.0 ========================================================
Episode 277/300. Reward sum: 500.0 ========================================================
Episode 278/300. Reward sum: 500.0 ========================================================
Episode 279/300. Reward sum: 500.0 ========================================================
Episode 280/300. Reward sum: 500.0 ========================================================
Episode 281/300. Reward sum: 500.0 ========================================================
Episode 282/300. Reward sum: 500.0 ========================================================
Episode 283/300. Reward sum: 500.0 ========================================================
Episode 284/300. Reward sum: 500.0 ========================================================
Episode 285/300. Reward sum: 500.0 ========================================================
Episode 286/300. Reward sum: 500.0 ========================================================
Episode 287/300. Reward sum: 500.0 ========================================================
Episode 288/300. Reward sum: 500.0 ========================================================
Episode 289/300. Reward sum: 500.0 ========================================================
Episode 290/300. Reward sum: 500.0 ========================================================
Episode 291/300. Reward sum: 500.0 ========================================================
Episode 292/300. Reward sum: 500.0 ========================================================
Episode 293/300. Reward sum: 500.0 ========================================================
Episode 294/300. Reward sum: 500.0 ========================================================
Episode 295/300. Reward sum: 500.0 ========================================================
Episode 296/300. Reward sum: 500.0 ========================================================
Episode 297/300. Reward sum: 500.0 ========================================================
Episode 298/300. Reward sum: 500.0 ========================================================
Episode 299/300. Reward sum: 500.0 ========================================================
Episode 300/300. Reward sum: 500.0 ==========================================================
```

Plot the Results

Plot the rewards as follows:

```
episodes = np.arange(len(rewards))
plt.figure()
plt.plot(episodes, rewards)
plt.xlabel('Episode')
plt.ylabel('Reward')
plt.draw()
```

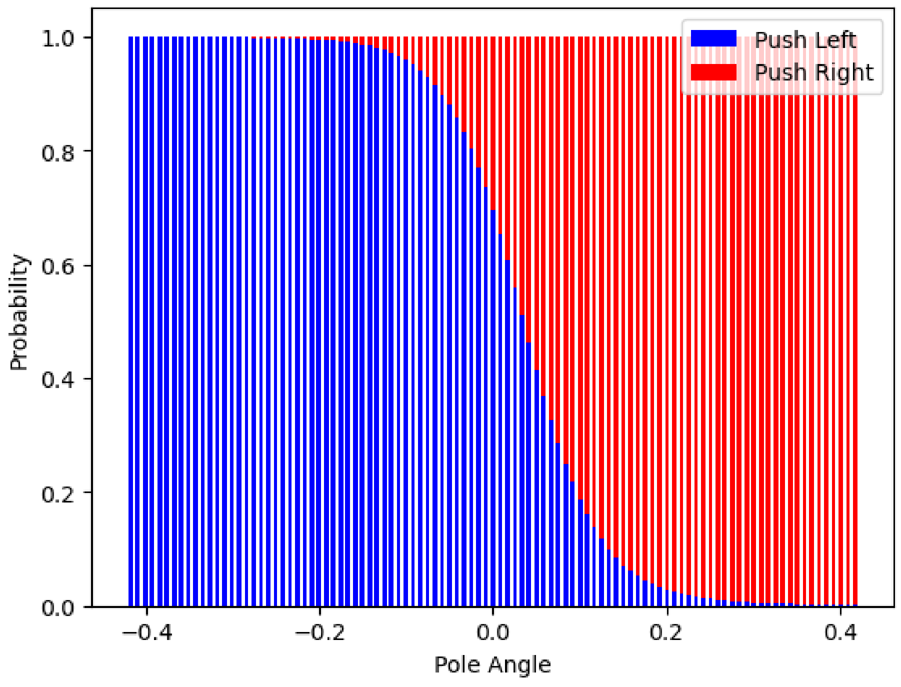


This plot shows the reward obtained in each episode. The reward is the sum of the rewards obtained in each step of the episode (i.e., how long the pole was kept upright). Because our method is on-policy, the reward is expected to increase over time as the policy network improves. We observe that on average the reward does in fact increase over time.

Plot a sample of the policy network for the pole (pendulum) angle as follows:

```
fig, ax = plt.subplots()
angles = np.linspace(-0.418, 0.418, 101)  # Pole angles
probs_left = np.zeros((101))  # Probability of pushing left
probs_right = np.zeros((101))  # Probability of pushing right
```

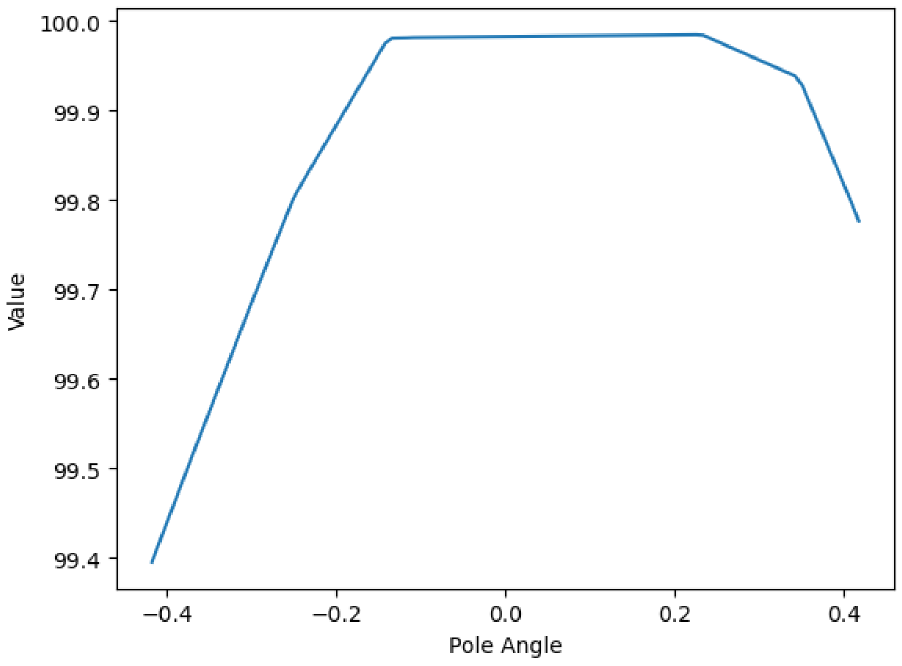
```
for i in range(len(angles)):
    state = np.array([0, 0, angles[i], 0])  # State
    prob = policy_network(state.reshape((1, -1))).numpy()[0]  # Probabilities
    probs_left[i] = prob[0]
    probs_right[i] = prob[1]
ax.bar(angles, probs_left, width=0.005, color='b')
ax.bar(angles, probs_right, width=0.005, bottom=probs_left, color='r')
ax.set_xlabel('Pole Angle')
ax.set_ylabel('Probability')
ax.legend(['Push Left', 'Push Right'])
plt.draw()
```



This plot shows the probability of pushing the cart left or right for different pole angles. We observe that the policy network outputs a higher probability of pushing the cart in the direction that will bring the pole closer to the vertical position (i.e., leftward if the pole is leaning left and rightward if the pole is leaning right). This is consistent with our intuition of an optimal policy.

Plot a sample of the value network for the pole (pendulum) angle as follows:

```
fig, ax = plt.subplots()
angles = np.linspace(-0.418, 0.418, 101)  # Pole angles
values = np.zeros((101))  # Value
for i in range(len(angles)):
    state = np.array([0, 0, angles[i], 0])  # State
    values[i] = value_network(state.reshape(1, -1)).numpy()[0][0]
ax.plot(angles, values)
ax.set_xlabel('Pole Angle')
ax.set_ylabel('Value')
plt.show()
```



This plot shows the value of the pole angle. We observe that the value network outputs a higher value for the pole angle closer to the vertical position. This is consistent with our intuition of an optimal value function.