

# Лабораторная работа 6. Выполнил Зоров Владислав Витальевич ИУ5-22м

На основе рассмотренных на лекции примеров реализуйте алгоритм DQN. В качестве среды можно использовать классические среды (в этом случае используется полносвязная архитектура нейронной сети). В качестве среды можно использовать игры Atari (в этом случае используется сверточная архитектура нейронной сети).

In [9]:

```
%matplotlib inline

import sys
import logging
import itertools
import copy

import numpy as np
np.random.seed(0)
import pandas as pd
import gym
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import torch.optim as optim
torch.manual_seed(0)
```

Out[9]:

```
<torch._C.Generator at 0x7a61d2d0cd10>
```

In [10]:

```
env = gym.make('MountainCar-v0')

# Вывод информации о переменных среды
env_vars = vars(env)
for key in env_vars:
    print(f"{key}: {env_vars[key]}")

# Вывод информации о переменных спецификации среды
spec_vars = vars(env.spec)
for key in spec_vars:
    print(f"{key}: {spec_vars[key]}")
```

```
env: <OrderEnforcing<PassiveEnvChecker<MountainCarEnv<MountainCar-v0>>>>
  _action_space: None
  _observation_space: None
  _reward_range: None
  _metadata: None
  _max_episode_steps: 200
  _elapsed_steps: None
  id: MountainCar-v0
  entry_point: gym.envs.classic_control.mountain_car:MountainCarEnv
  reward_threshold: -110.0
  nondeterministic: False
  max_episode_steps: 200
  order_enforce: True
  autoreset: False
  disable_env_checker: False
  apply_api_compatibility: False
  kwargs: {}
  namespace: None
  name: MountainCar
  version: 0
```

In [11]:

```
class DQNReplayer:
    def __init__(self, capacity):
        self.memory = pd.DataFrame(index=range(capacity),
                                   columns=['state', 'action', 'reward', 'next_state', 'terminated'])
        self.i = 0
        self.count = 0
        self.capacity = capacity

    def store(self, *args):
        self.memory.loc[self.i] = np.asarray(args, dtype=object)
        self.i = (self.i + 1) % self.capacity
        self.count = min(self.count + 1, self.capacity)

    def sample(self, size):
        indices = np.random.choice(self.count, size=size)
        return (np.stack(self.memory.loc[indices, field]) for field in
                self.memory.columns)

class DQNAgent:
    def __init__(self, env):
        self.action_n = env.action_space.n
        self.gamma = 0.99

        self.replayer = DQNReplayer(10000)

        self.device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

        self.evaluate_net = self.build_net(
            input_size=env.observation_space.shape[0],
            hidden_sizes=[64, 64], output_size=self.action_n
        ).to(self.device)
        self.optimizer = optim.Adam(self.evaluate_net.parameters(), lr=0.001)
        self.loss = nn.MSELoss()

    def build_net(self, input_size, hidden_sizes, output_size):
        layers = []
        for input_size, output_size in zip(
            [input_size,] + hidden_sizes, hidden_sizes + [output_size,]):
            layers.append(nn.Linear(input_size, output_size))
            layers.append(nn.ReLU())
        layers = layers[:-1]
        model = nn.Sequential(*layers).to(self.device)
        return model

    def reset(self, mode=None):
        self.mode = mode
        if self.mode == 'train':
            self.trajectory = []
            self.target_net = copy.deepcopy(self.evaluate_net)

    def step(self, observation, reward, terminated):
        if self.mode == 'train' and np.random.rand() < 0.001:
            # epsilon-greedy policy in train mode
            action = np.random.randint(self.action_n)
        else:
            state_tensor = torch.as_tensor(observation, dtype=torch.float).squeeze(0).to(self.device)
            q_tensor = self.evaluate_net(state_tensor)
            action_tensor = torch.argmax(q_tensor)
            action = action_tensor.item()
        if self.mode == 'train':
            self.trajectory += [observation, reward, terminated, action]
            if len(self.trajectory) >= 8:
                state, _, act, next_state, reward, terminated, _ = \
                    self.trajectory[-8:]
```

```

        self.replayer.store(state, act, reward, next_state, terminated)
        if self.replayer.count >= self.replayer.capacity * 0.95:
            # skip first few episodes for speed
            self.learn()
    return action

def close(self):
    pass

def learn(self):
    # replay
    states, actions, rewards, next_states, terminateds = \
        self.replayer.sample(1024)
    state_tensor = torch.as_tensor(states, dtype=torch.float).to(self.device)
    action_tensor = torch.as_tensor(actions, dtype=torch.long).to(self.device)
    reward_tensor = torch.as_tensor(rewards, dtype=torch.float).to(self.device)
    next_state_tensor = torch.as_tensor(next_states, dtype=torch.float).to(self.device)
    terminated_tensor = torch.as_tensor(terminateds, dtype=torch.float).to(self.device)

    # update value net
    next_q_tensor = self.target_net(next_state_tensor)
    next_max_q_tensor, _ = next_q_tensor.max(axis=-1)
    target_tensor = reward_tensor + self.gamma * \
        (1. - terminated_tensor) * next_max_q_tensor
    pred_tensor = self.evaluate_net(state_tensor)
    q_tensor = pred_tensor.gather(1, action_tensor.unsqueeze(1)).squeeze(1)
    loss_tensor = self.loss(target_tensor, q_tensor)
    self.optimizer.zero_grad()
    loss_tensor.backward()
    self.optimizer.step()

agent = DQNAgent(env)

def play_episode(env, agent, seed=None, mode=None, render=False):
    observation, _ = env.reset(seed=seed)
    reward, terminated, truncated = 0., False, False
    agent.reset(mode=mode)
    episode_reward, elapsed_steps = 0., 0
    while True:
        action = agent.step(observation, reward, terminated)
        if render:
            env.render()
        if terminated or truncated:
            break
        observation, reward, terminated, truncated, _ = env.step(action)
        episode_reward += reward
        elapsed_steps += 1
    agent.close()
    return episode_reward, elapsed_steps

print('==== train ====')
episode_rewards = []
for episode in itertools.count():
    episode_reward, elapsed_steps = play_episode(env, agent, seed=episode, mode='train')
    episode_rewards.append(episode_reward)
    print(f'train episode {episode}: reward = {episode_reward:.2f}, steps = {elapsed_steps}')
    if np.mean(episode_rewards[-10:]) > -110:
        break
plt.plot(episode_rewards)

print('==== test ====')
episode_rewards = []
for episode in range(100):
    episode_reward, elapsed_steps = play_episode(env, agent)

```



[illegible]

[illegible]

train episode 192: reward = -200.00, steps = 200  
train episode 193: reward = -200.00, steps = 200  
train episode 194: reward = -200.00, steps = 200  
train episode 195: reward = -195.00, steps = 195  
train episode 196: reward = -200.00, steps = 200  
train episode 197: reward = -200.00, steps = 200  
train episode 198: reward = -200.00, steps = 200  
train episode 199: reward = -200.00, steps = 200  
train episode 200: reward = -200.00, steps = 200  
train episode 201: reward = -200.00, steps = 200  
train episode 202: reward = -200.00, steps = 200  
train episode 203: reward = -186.00, steps = 186  
train episode 204: reward = -200.00, steps = 200  
train episode 205: reward = -177.00, steps = 177  
train episode 206: reward = -154.00, steps = 154  
train episode 207: reward = -177.00, steps = 177  
train episode 208: reward = -99.00, steps = 99  
train episode 209: reward = -200.00, steps = 200  
train episode 210: reward = -113.00, steps = 113  
train episode 211: reward = -172.00, steps = 172  
train episode 212: reward = -164.00, steps = 164  
train episode 213: reward = -114.00, steps = 114  
train episode 214: reward = -155.00, steps = 155  
train episode 215: reward = -87.00, steps = 87  
train episode 216: reward = -90.00, steps = 90  
train episode 217: reward = -92.00, steps = 92  
train episode 218: reward = -85.00, steps = 85  
train episode 219: reward = -156.00, steps = 156  
train episode 220: reward = -90.00, steps = 90  
train episode 221: reward = -153.00, steps = 153  
train episode 222: reward = -153.00, steps = 153  
train episode 223: reward = -94.00, steps = 94  
train episode 224: reward = -153.00, steps = 153  
train episode 225: reward = -157.00, steps = 157  
train episode 226: reward = -163.00, steps = 163  
train episode 227: reward = -153.00, steps = 153  
train episode 228: reward = -162.00, steps = 162  
train episode 229: reward = -94.00, steps = 94  
train episode 230: reward = -150.00, steps = 150  
train episode 231: reward = -148.00, steps = 148  
train episode 232: reward = -151.00, steps = 151  
train episode 233: reward = -200.00, steps = 200  
train episode 234: reward = -118.00, steps = 118  
train episode 235: reward = -200.00, steps = 200  
train episode 236: reward = -89.00, steps = 89  
train episode 237: reward = -200.00, steps = 200  
train episode 238: reward = -200.00, steps = 200  
train episode 239: reward = -200.00, steps = 200  
train episode 240: reward = -142.00, steps = 142  
train episode 241: reward = -200.00, steps = 200  
train episode 242: reward = -200.00, steps = 200  
train episode 243: reward = -200.00, steps = 200  
train episode 244: reward = -200.00, steps = 200  
train episode 245: reward = -154.00, steps = 154  
train episode 246: reward = -158.00, steps = 158  
train episode 247: reward = -91.00, steps = 91  
train episode 248: reward = -99.00, steps = 99  
train episode 249: reward = -85.00, steps = 85  
train episode 250: reward = -117.00, steps = 117  
train episode 251: reward = -159.00, steps = 159  
train episode 252: reward = -160.00, steps = 160  
train episode 253: reward = -117.00, steps = 117  
train episode 254: reward = -156.00, steps = 156  
train episode 255: reward = -88.00, steps = 88  
train episode 256: reward = -154.00, steps = 154  
train episode 257: reward = -152.00, steps = 152

```
train episode 258: reward = -173.00, steps = 173
train episode 259: reward = -200.00, steps = 200
train episode 260: reward = -200.00, steps = 200
train episode 261: reward = -186.00, steps = 186
train episode 262: reward = -170.00, steps = 170
train episode 263: reward = -200.00, steps = 200
train episode 264: reward = -200.00, steps = 200
train episode 265: reward = -164.00, steps = 164
train episode 266: reward = -159.00, steps = 159
train episode 267: reward = -200.00, steps = 200
train episode 268: reward = -200.00, steps = 200
train episode 269: reward = -145.00, steps = 145
train episode 270: reward = -162.00, steps = 162
train episode 271: reward = -200.00, steps = 200
train episode 272: reward = -140.00, steps = 140
train episode 273: reward = -130.00, steps = 130
train episode 274: reward = -171.00, steps = 171
train episode 275: reward = -200.00, steps = 200
train episode 276: reward = -200.00, steps = 200
train episode 277: reward = -200.00, steps = 200
train episode 278: reward = -200.00, steps = 200
train episode 279: reward = -119.00, steps = 119
train episode 280: reward = -200.00, steps = 200
train episode 281: reward = -127.00, steps = 127
train episode 282: reward = -200.00, steps = 200
train episode 283: reward = -200.00, steps = 200
train episode 284: reward = -200.00, steps = 200
train episode 285: reward = -122.00, steps = 122
train episode 286: reward = -125.00, steps = 125
train episode 287: reward = -132.00, steps = 132
train episode 288: reward = -124.00, steps = 124
train episode 289: reward = -119.00, steps = 119
train episode 290: reward = -119.00, steps = 119
train episode 291: reward = -120.00, steps = 120
train episode 292: reward = -115.00, steps = 115
train episode 293: reward = -115.00, steps = 115
train episode 294: reward = -114.00, steps = 114
train episode 295: reward = -116.00, steps = 116
train episode 296: reward = -113.00, steps = 113
train episode 297: reward = -112.00, steps = 112
train episode 298: reward = -111.00, steps = 111
train episode 299: reward = -111.00, steps = 111
train episode 300: reward = -111.00, steps = 111
train episode 301: reward = -112.00, steps = 112
train episode 302: reward = -90.00, steps = 90
train episode 303: reward = -114.00, steps = 114
train episode 304: reward = -108.00, steps = 108
==== test ====
```

```
test episode 0: reward = -85.00, steps = 85
test episode 1: reward = -111.00, steps = 111
test episode 2: reward = -108.00, steps = 108
test episode 3: reward = -114.00, steps = 114
test episode 4: reward = -90.00, steps = 90
test episode 5: reward = -115.00, steps = 115
test episode 6: reward = -113.00, steps = 113
test episode 7: reward = -112.00, steps = 112
test episode 8: reward = -114.00, steps = 114
test episode 9: reward = -113.00, steps = 113
test episode 10: reward = -113.00, steps = 113
test episode 11: reward = -112.00, steps = 112
test episode 12: reward = -113.00, steps = 113
test episode 13: reward = -110.00, steps = 110
test episode 14: reward = -111.00, steps = 111
test episode 15: reward = -114.00, steps = 114
test episode 16: reward = -107.00, steps = 107
test episode 17: reward = -114.00, steps = 114
```



test episode 18: reward = -107.00, steps = 107  
test episode 19: reward = -115.00, steps = 115  
test episode 20: reward = -113.00, steps = 113  
test episode 21: reward = -87.00, steps = 87  
test episode 22: reward = -107.00, steps = 107  
test episode 23: reward = -110.00, steps = 110  
test episode 24: reward = -108.00, steps = 108  
test episode 25: reward = -115.00, steps = 115  
test episode 26: reward = -112.00, steps = 112  
test episode 27: reward = -110.00, steps = 110  
test episode 28: reward = -114.00, steps = 114  
test episode 29: reward = -84.00, steps = 84  
test episode 30: reward = -86.00, steps = 86  
test episode 31: reward = -107.00, steps = 107  
test episode 32: reward = -115.00, steps = 115  
test episode 33: reward = -114.00, steps = 114  
test episode 34: reward = -115.00, steps = 115  
test episode 35: reward = -112.00, steps = 112  
test episode 36: reward = -98.00, steps = 98  
test episode 37: reward = -112.00, steps = 112  
test episode 38: reward = -111.00, steps = 111  
test episode 39: reward = -114.00, steps = 114  
test episode 40: reward = -115.00, steps = 115  
test episode 41: reward = -111.00, steps = 111  
test episode 42: reward = -113.00, steps = 113  
test episode 43: reward = -108.00, steps = 108  
test episode 44: reward = -83.00, steps = 83  
test episode 45: reward = -115.00, steps = 115  
test episode 46: reward = -113.00, steps = 113  
test episode 47: reward = -110.00, steps = 110  
test episode 48: reward = -97.00, steps = 97  
test episode 49: reward = -111.00, steps = 111  
test episode 50: reward = -88.00, steps = 88  
test episode 51: reward = -114.00, steps = 114  
test episode 52: reward = -108.00, steps = 108  
test episode 53: reward = -85.00, steps = 85  
test episode 54: reward = -108.00, steps = 108  
test episode 55: reward = -113.00, steps = 113  
test episode 56: reward = -108.00, steps = 108  
test episode 57: reward = -112.00, steps = 112  
test episode 58: reward = -114.00, steps = 114  
test episode 59: reward = -108.00, steps = 108  
test episode 60: reward = -112.00, steps = 112  
test episode 61: reward = -115.00, steps = 115  
test episode 62: reward = -113.00, steps = 113  
test episode 63: reward = -114.00, steps = 114  
test episode 64: reward = -108.00, steps = 108  
test episode 65: reward = -98.00, steps = 98  
test episode 66: reward = -113.00, steps = 113  
test episode 67: reward = -110.00, steps = 110  
test episode 68: reward = -114.00, steps = 114  
test episode 69: reward = -84.00, steps = 84  
test episode 70: reward = -115.00, steps = 115  
test episode 71: reward = -102.00, steps = 102  
test episode 72: reward = -113.00, steps = 113  
test episode 73: reward = -115.00, steps = 115  
test episode 74: reward = -108.00, steps = 108  
test episode 75: reward = -114.00, steps = 114  
test episode 76: reward = -98.00, steps = 98  
test episode 77: reward = -114.00, steps = 114  
test episode 78: reward = -114.00, steps = 114  
test episode 79: reward = -108.00, steps = 108  
test episode 80: reward = -84.00, steps = 84  
test episode 81: reward = -88.00, steps = 88  
test episode 82: reward = -113.00, steps = 113  
test episode 83: reward = -111.00, steps = 111

```
test episode 84: reward = -88.00, steps = 88
test episode 85: reward = -84.00, steps = 84
test episode 86: reward = -115.00, steps = 115
test episode 87: reward = -108.00, steps = 108
test episode 88: reward = -87.00, steps = 87
test episode 89: reward = -109.00, steps = 109
test episode 90: reward = -113.00, steps = 113
test episode 91: reward = -110.00, steps = 110
test episode 92: reward = -109.00, steps = 109
test episode 93: reward = -113.00, steps = 113
test episode 94: reward = -108.00, steps = 108
test episode 95: reward = -113.00, steps = 113
test episode 96: reward = -91.00, steps = 91
test episode 97: reward = -86.00, steps = 86
test episode 98: reward = -110.00, steps = 110
test episode 99: reward = -83.00, steps = 83
average episode reward = -106.74 ± 10.10
```

