

```
In [ ]: import gym

#定义环境
class MyWrapper(gym.Wrapper):
    def __init__(self):
        env = gym.make('Pendulum-v1', render_mode='rgb_array')
        super().__init__(env)
        self.env = env
        self.step_n = 0

    def reset(self):
        state, _ = self.env.reset()
        self.step_n = 0
        return state

    def step(self, action):
        state, reward, terminated, truncated, info = self.env.step(action)
        done = terminated or truncated
        self.step_n += 1
        if self.step_n >= 200:
            done = True
        return state, reward, done, info

env = MyWrapper()

env.reset()
```

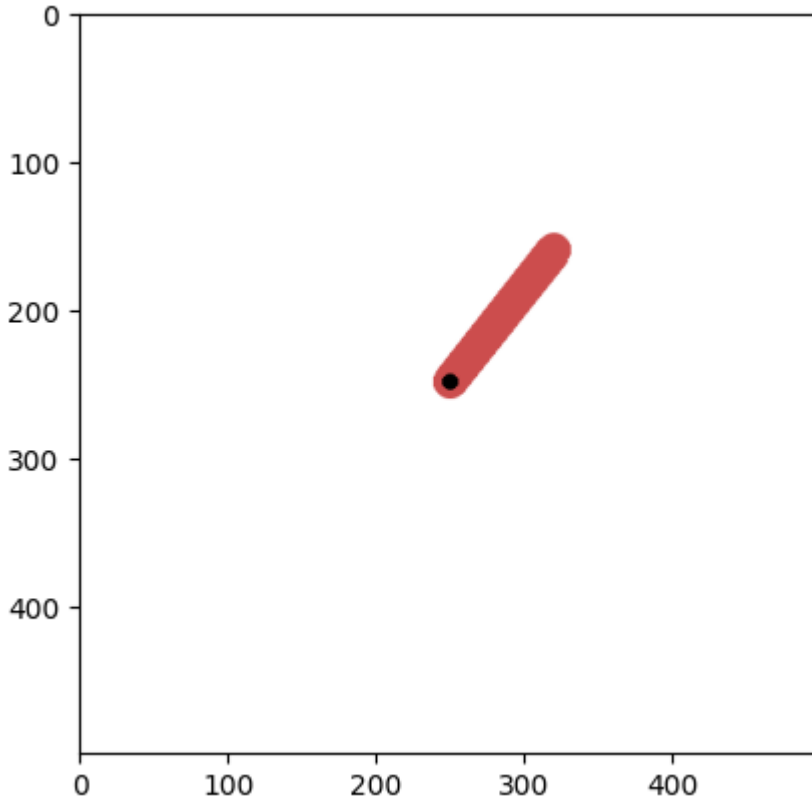
```
Out[ ]: array([ 0.7833687, -0.6215573,  0.8031729], dtype=float32)
```

```
In [ ]: from matplotlib import pyplot as plt

%matplotlib inline

#打印游戏
def show():
    plt.imshow(env.render())
    plt.show()

show()
```



```
In [ ]: for i in range(11):
        action_continuous = i
        action_continuous /=10
        action_continuous *=4
        action_continuous -= 2
        print(f'第 {i+1} 个动作: {action_continuous}')
```

第 1 个动作: -2.0
 第 2 个动作: -1.6
 第 3 个动作: -1.2
 第 4 个动作: -0.8
 第 5 个动作: -0.3999999999999999
 第 6 个动作: 0.0
 第 7 个动作: 0.3999999999999999
 第 8 个动作: 0.7999999999999998
 第 9 个动作: 1.2000000000000002
 第 10 个动作: 1.6
 第 11 个动作: 2.0

```
In [ ]: ## 创建两个网络, Actot、Critic
import torch
model_actor = torch.nn.Sequential(
    torch.nn.Linear(3,128),
    torch.nn.ReLU(),
    torch.nn.Linear(128,11),
    torch.nn.Softmax()
)
model_critic =torch.nn.Sequential(
    torch.nn.Linear(3,128),
    torch.nn.ReLU(),
    torch.nn.Linear(128,1)
)
model_actor(torch.randn(3,3)),model_critic(torch.randn(4,3))
```

```
d:\Anaconda install\envs\Gym\lib\site-packages\torch\nn\modules\container.py:217:
UserWarning: Implicit dimension choice for softmax has been deprecated. Change the
call to include dim=X as an argument.
```

```
input = module(input)
```

```
Out [ ]: (tensor([[0.0957, 0.0811, 0.0655, 0.0876, 0.1046, 0.1076, 0.1077, 0.0781, 0.095
3,
          0.0859, 0.0909],
          [0.0796, 0.0694, 0.0734, 0.1089, 0.0975, 0.0987, 0.1104, 0.0942, 0.106
8,
          0.0720, 0.0892],
          [0.1049, 0.1057, 0.0675, 0.0765, 0.0719, 0.1254, 0.0864, 0.0696, 0.134
8,
          0.0637, 0.0935]], grad_fn=<SoftmaxBackward0>),
tensor([[0.2486],
        [0.1126],
        [0.2664],
        [0.3611]], grad_fn=<AddmmBackward0>))
```

```
In [ ]: prob = model_actor(torch.randn(3,3))
prob
```

```
Out [ ]: tensor([[0.1249, 0.0718, 0.0422, 0.0722, 0.1112, 0.1278, 0.1251, 0.0736, 0.091
4,
          0.0564, 0.1032],
          [0.0694, 0.1218, 0.0659, 0.0935, 0.0574, 0.0981, 0.0838, 0.0723, 0.192
8,
          0.0431, 0.1020],
          [0.0908, 0.0741, 0.0663, 0.1003, 0.1034, 0.1029, 0.1156, 0.0844, 0.100
1,
          0.0714, 0.0906]], grad_fn=<SoftmaxBackward0>)
```

```
In [ ]: import random

def get_action(state):
    state = torch.FloatTensor(state).reshape(1, 3)

    prob = model_actor(state)

    action = random.choices(range(11), weights=prob[0].tolist(), k=1)[0]
    action_continuous = action
    action_continuous /= 10
    action_continuous *= 4
    action_continuous -= 2

    return action, action_continuous

get_action([0,1,2])
```

```
Out [ ]: (0, -2.0)
```

```
In [ ]: action_continuous
```

```
Out [ ]: 2.0
```

```
In [ ]: env.step([0.1])
```

```
Out[ ]: (array([ 0.79418635, -0.60767424,  0.35200495], dtype=float32),
        -0.514396180052578,
        False,
        {})
```

```
In [ ]: def get_data():
        states = []
        rewards = []
        actions = []
        next_states = []
        overs = []

        # 初始化游戏
        state = env.reset()
        over = False
        while not over:
            action, action_continuous = get_action(state)
            #print(f'执行动作:{action_continuous}')
            next_state, reward, over, _ = env.step([action_continuous])
            states.append(state)
            rewards.append(reward)
            actions.append(action)
            next_states.append(next_state)
            overs.append(over)

            state = next_state

        states = torch.FloatTensor(states).reshape(-1, 3)

        rewards = torch.FloatTensor(rewards).reshape(-1, 1)

        actions = torch.LongTensor(actions).reshape(-1, 1)

        next_states = torch.FloatTensor(next_states).reshape(-1, 3)

        overs = torch.LongTensor(overs).reshape(-1, 1)

        return states, rewards, actions, next_states, overs

get_data()
```

C:\Users\cgq10\AppData\Local\Temp\ipykernel_7988\303743510.py:23: UserWarning: Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at C:\actions-runner_work\pytorch\pytorch\buidler\windows\pytorch\torch\src\utils\tensor_new.cpp:248.)

```
states = torch.FloatTensor(states).reshape(-1, 3)
```

```
Out[ ]: (tensor([[ -2.2938e-01, -9.7334e-01,  8.3112e-02],
 [ -2.6363e-01, -9.6462e-01, -7.0689e-01],
 [ -3.3189e-01, -9.4332e-01, -1.4304e+00],
 [ -4.1977e-01, -9.0763e-01, -1.8978e+00],
 [ -5.2535e-01, -8.5089e-01, -2.3986e+00],
 [ -6.5710e-01, -7.5381e-01, -3.2767e+00],
 [ -7.8337e-01, -6.2156e-01, -3.6621e+00],
 [ -8.9812e-01, -4.3975e-01, -4.3083e+00],
 [ -9.7581e-01, -2.1863e-01, -4.6981e+00],
 [ -9.9991e-01,  1.3687e-02, -4.6820e+00],
 [ -9.6958e-01,  2.4476e-01, -4.6718e+00],
 [ -8.9351e-01,  4.4904e-01, -4.3682e+00],
 [ -7.7992e-01,  6.2588e-01, -4.2114e+00],
 [ -6.4989e-01,  7.6003e-01, -3.7420e+00],
 [ -5.2425e-01,  8.5156e-01, -3.1120e+00],
 [ -4.1793e-01,  9.0848e-01, -2.4133e+00],
 [ -3.4062e-01,  9.4020e-01, -1.6720e+00],
 [ -2.9765e-01,  9.5467e-01, -9.0682e-01],
 [ -2.9140e-01,  9.5660e-01, -1.3081e-01],
 [ -3.2785e-01,  9.4473e-01,  7.6664e-01],
 [ -4.1030e-01,  9.1195e-01,  1.7752e+00],
 [ -5.0617e-01,  8.6243e-01,  2.1592e+00],
 [ -6.2649e-01,  7.7943e-01,  2.9260e+00],
 [ -7.4901e-01,  6.6256e-01,  3.3905e+00],
 [ -8.6590e-01,  5.0022e-01,  4.0075e+00],
 [ -9.5659e-01,  2.9143e-01,  4.5626e+00],
 [ -9.9886e-01,  4.7646e-02,  4.9612e+00],
 [ -9.7902e-01, -2.0375e-01,  5.0569e+00],
 [ -9.0671e-01, -4.2176e-01,  4.6041e+00],
 [ -7.9983e-01, -6.0022e-01,  4.1678e+00],
 [ -6.7733e-01, -7.3568e-01,  3.6576e+00],
 [ -5.6533e-01, -8.2486e-01,  2.8659e+00],
 [ -4.7193e-01, -8.8164e-01,  2.1872e+00],
 [ -3.9234e-01, -9.1982e-01,  1.7660e+00],
 [ -3.3383e-01, -9.4263e-01,  1.2561e+00],
 [ -3.2207e-01, -9.4672e-01,  2.4918e-01],
 [ -3.5504e-01, -9.3485e-01, -7.0086e-01],
 [ -4.2237e-01, -9.0643e-01, -1.4620e+00],
 [ -5.1426e-01, -8.5763e-01, -2.0818e+00],
 [ -6.1658e-01, -7.8729e-01, -2.4850e+00],
 [ -7.4007e-01, -6.7253e-01, -3.3755e+00],
 [ -8.6046e-01, -5.0952e-01, -4.0599e+00],
 [ -9.5248e-01, -3.0461e-01, -4.5021e+00],
 [ -9.9613e-01, -8.7887e-02, -4.4305e+00],
 [ -9.8850e-01,  1.5124e-01, -4.7964e+00],
 [ -9.3198e-01,  3.6252e-01, -4.3830e+00],
 [ -8.3836e-01,  5.4511e-01, -4.1111e+00],
 [ -7.2371e-01,  6.9011e-01, -3.7023e+00],
 [ -6.0273e-01,  7.9794e-01, -3.2447e+00],
 [ -5.0001e-01,  8.6602e-01, -2.4662e+00],
 [ -4.3026e-01,  9.0271e-01, -1.5767e+00],
 [ -3.7814e-01,  9.2575e-01, -1.1397e+00],
 [ -3.6303e-01,  9.3178e-01, -3.2538e-01],
 [ -3.6924e-01,  9.2933e-01,  1.3346e-01],
 [ -4.0750e-01,  9.1320e-01,  8.3046e-01],
 [ -4.7017e-01,  8.8257e-01,  1.3954e+00],
 [ -5.6080e-01,  8.2795e-01,  2.1173e+00],
 [ -6.5732e-01,  7.5361e-01,  2.4383e+00],
 [ -7.6267e-01,  6.4678e-01,  3.0035e+00],
 [ -8.6785e-01,  4.9682e-01,  3.6685e+00],
```

[-9.4896e-01, 3.1540e-01, 3.9812e+00],
[-9.9362e-01, 1.1275e-01, 4.1577e+00],
[-9.9621e-01, -8.7014e-02, 4.0023e+00],
[-9.5739e-01, -2.8880e-01, 4.1170e+00],
[-8.8045e-01, -4.7415e-01, 4.0204e+00],
[-7.8862e-01, -6.1488e-01, 3.3648e+00],
[-6.9569e-01, -7.1834e-01, 2.7836e+00],
[-6.1322e-01, -7.8991e-01, 2.1849e+00],
[-5.4090e-01, -8.4108e-01, 1.7724e+00],
[-4.8681e-01, -8.7351e-01, 1.2616e+00],
[-4.6010e-01, -8.8787e-01, 6.0650e-01],
[-4.6007e-01, -8.8788e-01, 6.0273e-04],
[-4.9978e-01, -8.6615e-01, -9.0531e-01],
[-5.7048e-01, -8.2131e-01, -1.6749e+00],
[-6.7180e-01, -7.4074e-01, -2.5909e+00],
[-7.7956e-01, -6.2633e-01, -3.1465e+00],
[-8.7948e-01, -4.7594e-01, -3.6162e+00],
[-9.5698e-01, -2.9014e-01, -4.0332e+00],
[-9.9665e-01, -8.1743e-02, -4.2508e+00],
[-9.9223e-01, 1.2445e-01, -4.1321e+00],
[-9.4418e-01, 3.2944e-01, -4.2187e+00],
[-8.6215e-01, 5.0665e-01, -3.9117e+00],
[-7.5779e-01, 6.5250e-01, -3.5917e+00],
[-6.5472e-01, 7.5588e-01, -2.9223e+00],
[-5.6385e-01, 8.2588e-01, -2.2954e+00],
[-5.0055e-01, 8.6571e-01, -1.4960e+00],
[-4.6080e-01, 8.8750e-01, -9.0671e-01],
[-4.3663e-01, 8.9964e-01, -5.4109e-01],
[-4.3994e-01, 8.9803e-01, 7.3646e-02],
[-4.6257e-01, 8.8658e-01, 5.0717e-01],
[-5.0854e-01, 8.6104e-01, 1.0521e+00],
[-5.7972e-01, 8.1481e-01, 1.6979e+00],
[-6.6749e-01, 7.4462e-01, 2.2490e+00],
[-7.6511e-01, 6.4390e-01, 2.8075e+00],
[-8.6176e-01, 5.0732e-01, 3.3504e+00],
[-9.4090e-01, 3.3869e-01, 3.7309e+00],
[-9.8750e-01, 1.5762e-01, 3.7449e+00],
[-9.9958e-01, -2.8871e-02, 3.7431e+00],
[-9.7825e-01, -2.0743e-01, 3.6014e+00],
[-9.2252e-01, -3.8595e-01, 3.7459e+00],
[-8.3916e-01, -5.4389e-01, 3.5764e+00],
[-7.3272e-01, -6.8053e-01, 3.4685e+00],
[-6.2677e-01, -7.7920e-01, 2.8981e+00],
[-5.3771e-01, -8.4313e-01, 2.1937e+00],
[-4.7561e-01, -8.7966e-01, 1.4413e+00],
[-4.5429e-01, -8.9086e-01, 4.8160e-01],
[-4.7054e-01, -8.8238e-01, -3.6654e-01],
[-5.2040e-01, -8.5392e-01, -1.1483e+00],
[-5.8975e-01, -8.0758e-01, -1.6688e+00],
[-6.7979e-01, -7.3341e-01, -2.3345e+00],
[-7.7436e-01, -6.3275e-01, -2.7645e+00],
[-8.7219e-01, -4.8917e-01, -3.4791e+00],
[-9.4960e-01, -3.1347e-01, -3.8459e+00],
[-9.9235e-01, -1.2348e-01, -3.9010e+00],
[-9.9756e-01, 6.9834e-02, -3.8737e+00],
[-9.6458e-01, 2.6380e-01, -3.9413e+00],
[-8.9600e-01, 4.4406e-01, -3.8634e+00],
[-8.1117e-01, 5.8481e-01, -3.2904e+00],
[-7.2605e-01, 6.8764e-01, -2.6718e+00],
[-6.5241e-01, 7.5787e-01, -2.0360e+00],

[-5.9990e-01, 8.0007e-01, -1.3476e+00],
[-5.8185e-01, 8.1330e-01, -4.4759e-01],
[-5.7624e-01, 8.1728e-01, -1.3762e-01],
[-6.0509e-01, 7.9615e-01, 7.1535e-01],
[-6.5373e-01, 7.5673e-01, 1.2525e+00],
[-7.0930e-01, 7.0491e-01, 1.5200e+00],
[-7.7766e-01, 6.2868e-01, 2.0487e+00],
[-8.4252e-01, 5.3866e-01, 2.2202e+00],
[-9.0320e-01, 4.2923e-01, 2.5042e+00],
[-9.5100e-01, 3.0918e-01, 2.5861e+00],
[-9.8286e-01, 1.8437e-01, 2.5780e+00],
[-9.9891e-01, 4.6600e-02, 2.7763e+00],
[-9.9530e-01, -9.6792e-02, 2.8712e+00],
[-9.7208e-01, -2.3467e-01, 2.7986e+00],
[-9.2977e-01, -3.6814e-01, 2.8026e+00],
[-8.7160e-01, -4.9021e-01, 2.7065e+00],
[-7.9953e-01, -6.0062e-01, 2.6389e+00],
[-7.3121e-01, -6.8215e-01, 2.1284e+00],
[-6.7374e-01, -7.3897e-01, 1.6168e+00],
[-6.3587e-01, -7.7180e-01, 1.0026e+00],
[-6.1465e-01, -7.8880e-01, 5.4372e-01],
[-6.1890e-01, -7.8547e-01, -1.0788e-01],
[-6.4360e-01, -7.6536e-01, -6.3698e-01],
[-6.8655e-01, -7.2708e-01, -1.1510e+00],
[-7.3965e-01, -6.7299e-01, -1.5163e+00],
[-8.0198e-01, -5.9735e-01, -1.9611e+00],
[-8.6193e-01, -5.0703e-01, -2.1691e+00],
[-9.2289e-01, -3.8505e-01, -2.7293e+00],
[-9.7029e-01, -2.4194e-01, -3.0181e+00],
[-9.9693e-01, -7.8302e-02, -3.3196e+00],
[-9.9715e-01, 7.5462e-02, -3.0783e+00],
[-9.7706e-01, 2.1298e-01, -2.7817e+00],
[-9.4483e-01, 3.2756e-01, -2.3820e+00],
[-8.9934e-01, 4.3725e-01, -2.3763e+00],
[-8.4675e-01, 5.3199e-01, -2.1684e+00],
[-7.9096e-01, 6.1187e-01, -1.9494e+00],
[-7.5118e-01, 6.6010e-01, -1.2505e+00],
[-7.2365e-01, 6.9016e-01, -8.1541e-01],
[-7.1958e-01, 6.9441e-01, -1.1779e-01],
[-7.2314e-01, 6.9070e-01, 1.0302e-01],
[-7.4624e-01, 6.6568e-01, 6.8105e-01],
[-7.8047e-01, 6.2520e-01, 1.0603e+00],
[-8.1739e-01, 5.7608e-01, 1.2292e+00],
[-8.5623e-01, 5.1659e-01, 1.4213e+00],
[-8.9273e-01, 4.5059e-01, 1.5087e+00],
[-9.2487e-01, 3.8028e-01, 1.5467e+00],
[-9.5489e-01, 2.9695e-01, 1.7719e+00],
[-9.7911e-01, 2.0334e-01, 1.9346e+00],
[-9.9335e-01, 1.1516e-01, 1.7871e+00],
[-9.9953e-01, 3.0741e-02, 1.6935e+00],
[-9.9848e-01, -5.5052e-02, 1.7165e+00],
[-9.9158e-01, -1.2948e-01, 1.4952e+00],
[-9.8298e-01, -1.8370e-01, 1.0981e+00],
[-9.6946e-01, -2.4524e-01, 1.2603e+00],
[-9.5750e-01, -2.8843e-01, 8.9642e-01],
[-9.5185e-01, -3.0657e-01, 3.8009e-01],
[-9.5230e-01, -3.0515e-01, -2.9834e-02],
[-9.5962e-01, -2.8131e-01, -4.9870e-01],
[-9.6749e-01, -2.5290e-01, -5.8968e-01],
[-9.7328e-01, -2.2964e-01, -4.7936e-01],

```
[-9.8254e-01, -1.8604e-01, -8.9159e-01],  
[-9.9237e-01, -1.2326e-01, -1.2711e+00],  
[-9.9863e-01, -5.2372e-02, -1.4236e+00],  
[-9.9984e-01, 1.7745e-02, -1.4028e+00],  
[-9.9566e-01, 9.3088e-02, -1.5095e+00],  
[-9.8638e-01, 1.6446e-01, -1.4397e+00],  
[-9.7611e-01, 2.1728e-01, -1.0764e+00],  
[-9.6595e-01, 2.5872e-01, -8.5342e-01],  
[-9.5947e-01, 2.8180e-01, -4.7938e-01],  
[-9.5822e-01, 2.8602e-01, -8.8027e-02],  
[-9.6411e-01, 2.6552e-01, 4.2649e-01],  
[-9.7194e-01, 2.3524e-01, 6.2563e-01],  
[-9.8059e-01, 1.9608e-01, 8.0206e-01],  
[-9.9051e-01, 1.3747e-01, 1.1891e+00],  
[-9.9610e-01, 8.8179e-02, 9.9222e-01],  
[-9.9937e-01, 3.5369e-02, 1.0584e+00],  
[-9.9995e-01, -9.8677e-03, 9.0488e-01],  
[-9.9850e-01, -5.4714e-02, 8.9748e-01],  
[-9.9581e-01, -9.1436e-02, 7.3644e-01],  
[-9.9144e-01, -1.3058e-01, 7.8786e-01]],  
tensor([[ -3.2489],  
[ -3.4267],  
[ -3.8518],  
[ -4.3776],  
[ -5.0889],  
[ -6.3090],  
[ -7.4477],  
[ -9.0723],  
[-10.7419],  
[-11.9759],  
[-10.5603],  
[ -9.0700],  
[ -7.8515],  
[ -6.5908],  
[ -5.4742],  
[ -4.5904],  
[ -3.9598],  
[ -3.5906],  
[ -3.4869],  
[ -3.6911],  
[ -4.2935],  
[ -4.8833],  
[ -5.9095],  
[ -6.9939],  
[ -8.4600],  
[-10.1822],  
[-12.0339],  
[-11.1838],  
[ -9.4440],  
[ -7.9763],  
[ -6.6993],  
[ -5.5375],  
[ -4.7339],  
[ -4.2099],  
[ -3.8143],  
[ -3.6139],  
[ -3.7887],  
[ -4.2414],  
[ -4.8920],  
[ -5.6176],
```


[-6.9199],
[-8.4447],
[-10.0515],
[-11.2914],
[-11.2433],
[-9.5974],
[-8.2697],
[-7.0350],
[-5.9725],
[-4.9973],
[-4.3137],
[-3.9666],
[-3.7857],
[-3.8003],
[-4.0318],
[-4.4396],
[-5.1445],
[-5.8297],
[-6.8485],
[-8.2191],
[-9.5416],
[-10.9037],
[-10.9330],
[-9.8102],
[-8.6301],
[-7.2800],
[-6.2514],
[-5.4559],
[-4.9043],
[-4.4824],
[-4.2349],
[-4.2004],
[-4.4680],
[-5.0277],
[-5.9955],
[-7.0650],
[-8.3069],
[-9.7333],
[-11.1705],
[-10.8101],
[-9.6529],
[-8.3440],
[-7.1998],
[-6.0736],
[-5.2365],
[-4.6131],
[-4.2875],
[-4.1205],
[-4.1091],
[-4.2358],
[-4.5387],
[-5.0810],
[-5.8033],
[-6.7517],
[-7.9321],
[-9.2125],
[-10.3032],
[-11.0907],
[-9.9014],
[-8.9408],

[-7.8702],
[-6.9302],
[-5.8949],
[-5.0551],
[-4.4820],
[-4.1959],
[-4.2605],
[-4.6189],
[-5.1255],
[-5.9200],
[-6.8012],
[-8.1297],
[-9.4485],
[-10.6296],
[-10.9365],
[-9.8175],
[-8.6854],
[-7.4191],
[-6.3949],
[-5.6207],
[-5.0882],
[-4.8280],
[-4.7783],
[-4.9827],
[-5.3743],
[-5.7974],
[-6.4839],
[-7.1126],
[-7.9086],
[-8.6648],
[-9.4037],
[-10.3498],
[-10.0943],
[-9.2221],
[-8.4298],
[-7.6495],
[-6.9331],
[-6.1694],
[-5.5979],
[-5.2084],
[-5.0149],
[-5.0106],
[-5.1936],
[-5.5513],
[-6.0062],
[-6.6442],
[-7.2833],
[-8.2872],
[-9.3055],
[-10.4892],
[-10.3509],
[-9.3435],
[-8.4541],
[-7.7964],
[-7.1313],
[-6.5487],
[-6.0161],
[-5.7317],
[-5.6412],
[-5.6615],

```
[ -5.8705],  
[ -6.1986],  
[ -6.5428],  
[ -6.9594],  
[ -7.3828],  
[ -7.8101],  
[ -8.3802],  
[ -9.0032],  
[ -9.4785],  
[ -9.9641],  
[ -9.8226],  
[ -9.2982],  
[ -8.8675],  
[ -8.5345],  
[ -8.2012],  
[ -8.0248],  
[ -8.0200],  
[ -8.1847],  
[ -8.3673],  
[ -8.4930],  
[ -8.8109],  
[ -9.2702],  
[ -9.7459],  
[ -9.9559],  
[ -9.5204],  
[ -9.0687],  
[ -8.6574],  
[ -8.3681],  
[ -8.1807],  
[ -8.1360],  
[ -8.2714],  
[ -8.4731],  
[ -8.7354],  
[ -9.1675],  
[ -9.4211],  
[ -9.7620],  
[ -9.8896],  
[ -9.6098],  
[ -9.3576],  
[ -9.1262]]),  
tensor([[ 4],  
[ 5],  
[ 9],  
[ 8],  
[ 1],  
[ 8],  
[ 2],  
[ 4],  
[ 8],  
[ 5],  
[ 7],  
[ 2],  
[ 5],  
[ 6],  
[ 6],  
[ 6],  
[ 6],  
[ 6],  
[ 6],  
[ 8],  
[10],
```

[0],
[7],
[3],
[7],
[8],
[8],
[6],
[0],
[3],
[4],
[1],
[4],
[9],
[8],
[0],
[1],
[4],
[6],
[9],
[0],
[2],
[4],
[10],
[0],
[10],
[5],
[5],
[4],
[8],
[9],
[1],
[7],
[1],
[5],
[3],
[6],
[0],
[5],
[8],
[4],
[4],
[1],
[8],
[7],
[0],
[3],
[4],
[8],
[7],
[5],
[6],
[1],
[3],
[0],
[5],
[5],
[4],
[5],
[8],
[2],

[6],
[4],
[8],
[6],
[8],
[4],
[0],
[4],
[1],
[3],
[5],
[4],
[5],
[6],
[5],
[1],
[3],
[3],
[10],
[7],
[10],
[4],
[3],
[3],
[0],
[2],
[3],
[7],
[4],
[7],
[1],
[5],
[8],
[7],
[3],
[3],
[9],
[8],
[7],
[7],
[10],
[0],
[9],
[4],
[0],
[5],
[0],
[3],
[1],
[1],
[6],
[6],
[5],
[8],
[8],
[10],
[4],
[5],
[4],
[7],

```
[ 4],  
[ 6],  
[ 6],  
[ 8],  
[ 6],  
[ 9],  
[ 2],  
[ 5],  
[ 3],  
[10],  
[ 9],  
[ 9],  
[ 1],  
[ 3],  
[ 2],  
[ 9],  
[ 4],  
[ 8],  
[ 0],  
[ 6],  
[ 3],  
[ 0],  
[ 1],  
[ 0],  
[ 0],  
[ 4],  
[ 4],  
[ 0],  
[ 2],  
[ 5],  
[ 2],  
[ 0],  
[10],  
[ 2],  
[ 0],  
[ 2],  
[ 1],  
[ 7],  
[10],  
[ 1],  
[ 1],  
[ 4],  
[ 6],  
[ 3],  
[ 5],  
[ 9],  
[ 6],  
[ 8],  
[ 8],  
[10],  
[ 5],  
[ 5],  
[ 9],  
[ 0],  
[ 5],  
[ 2],  
[ 5],  
[ 3],  
[ 7],  
[ 6]]),
```

```
tensor([[ -2.6363e-01, -9.6462e-01, -7.0689e-01],
        [ -3.3189e-01, -9.4332e-01, -1.4304e+00],
        [ -4.1977e-01, -9.0763e-01, -1.8978e+00],
        [ -5.2535e-01, -8.5089e-01, -2.3986e+00],
        [ -6.5710e-01, -7.5381e-01, -3.2767e+00],
        [ -7.8337e-01, -6.2156e-01, -3.6621e+00],
        [ -8.9812e-01, -4.3975e-01, -4.3083e+00],
        [ -9.7581e-01, -2.1863e-01, -4.6981e+00],
        [ -9.9991e-01,  1.3687e-02, -4.6820e+00],
        [ -9.6958e-01,  2.4476e-01, -4.6718e+00],
        [ -8.9351e-01,  4.4904e-01, -4.3682e+00],
        [ -7.7992e-01,  6.2588e-01, -4.2114e+00],
        [ -6.4989e-01,  7.6003e-01, -3.7420e+00],
        [ -5.2425e-01,  8.5156e-01, -3.1120e+00],
        [ -4.1793e-01,  9.0848e-01, -2.4133e+00],
        [ -3.4062e-01,  9.4020e-01, -1.6720e+00],
        [ -2.9765e-01,  9.5467e-01, -9.0682e-01],
        [ -2.9140e-01,  9.5660e-01, -1.3081e-01],
        [ -3.2785e-01,  9.4473e-01,  7.6664e-01],
        [ -4.1030e-01,  9.1195e-01,  1.7752e+00],
        [ -5.0617e-01,  8.6243e-01,  2.1592e+00],
        [ -6.2649e-01,  7.7943e-01,  2.9260e+00],
        [ -7.4901e-01,  6.6256e-01,  3.3905e+00],
        [ -8.6590e-01,  5.0022e-01,  4.0075e+00],
        [ -9.5659e-01,  2.9143e-01,  4.5626e+00],
        [ -9.9886e-01,  4.7646e-02,  4.9612e+00],
        [ -9.7902e-01, -2.0375e-01,  5.0569e+00],
        [ -9.0671e-01, -4.2176e-01,  4.6041e+00],
        [ -7.9983e-01, -6.0022e-01,  4.1678e+00],
        [ -6.7733e-01, -7.3568e-01,  3.6576e+00],
        [ -5.6533e-01, -8.2486e-01,  2.8659e+00],
        [ -4.7193e-01, -8.8164e-01,  2.1872e+00],
        [ -3.9234e-01, -9.1982e-01,  1.7660e+00],
        [ -3.3383e-01, -9.4263e-01,  1.2561e+00],
        [ -3.2207e-01, -9.4672e-01,  2.4918e-01],
        [ -3.5504e-01, -9.3485e-01, -7.0086e-01],
        [ -4.2237e-01, -9.0643e-01, -1.4620e+00],
        [ -5.1426e-01, -8.5763e-01, -2.0818e+00],
        [ -6.1658e-01, -7.8729e-01, -2.4850e+00],
        [ -7.4007e-01, -6.7253e-01, -3.3755e+00],
        [ -8.6046e-01, -5.0952e-01, -4.0599e+00],
        [ -9.5248e-01, -3.0461e-01, -4.5021e+00],
        [ -9.9613e-01, -8.7887e-02, -4.4305e+00],
        [ -9.8850e-01,  1.5124e-01, -4.7964e+00],
        [ -9.3198e-01,  3.6252e-01, -4.3830e+00],
        [ -8.3836e-01,  5.4511e-01, -4.1111e+00],
        [ -7.2371e-01,  6.9011e-01, -3.7023e+00],
        [ -6.0273e-01,  7.9794e-01, -3.2447e+00],
        [ -5.0001e-01,  8.6602e-01, -2.4662e+00],
        [ -4.3026e-01,  9.0271e-01, -1.5767e+00],
        [ -3.7814e-01,  9.2575e-01, -1.1397e+00],
        [ -3.6303e-01,  9.3178e-01, -3.2538e-01],
        [ -3.6924e-01,  9.2933e-01,  1.3346e-01],
        [ -4.0750e-01,  9.1320e-01,  8.3046e-01],
        [ -4.7017e-01,  8.8257e-01,  1.3954e+00],
        [ -5.6080e-01,  8.2795e-01,  2.1173e+00],
        [ -6.5732e-01,  7.5361e-01,  2.4383e+00],
        [ -7.6267e-01,  6.4678e-01,  3.0035e+00],
        [ -8.6785e-01,  4.9682e-01,  3.6685e+00],
        [ -9.4896e-01,  3.1540e-01,  3.9812e+00],
```

[-9.9362e-01, 1.1275e-01, 4.1577e+00],
[-9.9621e-01, -8.7014e-02, 4.0023e+00],
[-9.5739e-01, -2.8880e-01, 4.1170e+00],
[-8.8045e-01, -4.7415e-01, 4.0204e+00],
[-7.8862e-01, -6.1488e-01, 3.3648e+00],
[-6.9569e-01, -7.1834e-01, 2.7836e+00],
[-6.1322e-01, -7.8991e-01, 2.1849e+00],
[-5.4090e-01, -8.4108e-01, 1.7724e+00],
[-4.8681e-01, -8.7351e-01, 1.2616e+00],
[-4.6010e-01, -8.8787e-01, 6.0650e-01],
[-4.6007e-01, -8.8788e-01, 6.0273e-04],
[-4.9978e-01, -8.6615e-01, -9.0531e-01],
[-5.7048e-01, -8.2131e-01, -1.6749e+00],
[-6.7180e-01, -7.4074e-01, -2.5909e+00],
[-7.7956e-01, -6.2633e-01, -3.1465e+00],
[-8.7948e-01, -4.7594e-01, -3.6162e+00],
[-9.5698e-01, -2.9014e-01, -4.0332e+00],
[-9.9665e-01, -8.1743e-02, -4.2508e+00],
[-9.9223e-01, 1.2445e-01, -4.1321e+00],
[-9.4418e-01, 3.2944e-01, -4.2187e+00],
[-8.6215e-01, 5.0665e-01, -3.9117e+00],
[-7.5779e-01, 6.5250e-01, -3.5917e+00],
[-6.5472e-01, 7.5588e-01, -2.9223e+00],
[-5.6385e-01, 8.2588e-01, -2.2954e+00],
[-5.0055e-01, 8.6571e-01, -1.4960e+00],
[-4.6080e-01, 8.8750e-01, -9.0671e-01],
[-4.3663e-01, 8.9964e-01, -5.4109e-01],
[-4.3994e-01, 8.9803e-01, 7.3646e-02],
[-4.6257e-01, 8.8658e-01, 5.0717e-01],
[-5.0854e-01, 8.6104e-01, 1.0521e+00],
[-5.7972e-01, 8.1481e-01, 1.6979e+00],
[-6.6749e-01, 7.4462e-01, 2.2490e+00],
[-7.6511e-01, 6.4390e-01, 2.8075e+00],
[-8.6176e-01, 5.0732e-01, 3.3504e+00],
[-9.4090e-01, 3.3869e-01, 3.7309e+00],
[-9.8750e-01, 1.5762e-01, 3.7449e+00],
[-9.9958e-01, -2.8871e-02, 3.7431e+00],
[-9.7825e-01, -2.0743e-01, 3.6014e+00],
[-9.2252e-01, -3.8595e-01, 3.7459e+00],
[-8.3916e-01, -5.4389e-01, 3.5764e+00],
[-7.3272e-01, -6.8053e-01, 3.4685e+00],
[-6.2677e-01, -7.7920e-01, 2.8981e+00],
[-5.3771e-01, -8.4313e-01, 2.1937e+00],
[-4.7561e-01, -8.7966e-01, 1.4413e+00],
[-4.5429e-01, -8.9086e-01, 4.8160e-01],
[-4.7054e-01, -8.8238e-01, -3.6654e-01],
[-5.2040e-01, -8.5392e-01, -1.1483e+00],
[-5.8975e-01, -8.0758e-01, -1.6688e+00],
[-6.7979e-01, -7.3341e-01, -2.3345e+00],
[-7.7436e-01, -6.3275e-01, -2.7645e+00],
[-8.7219e-01, -4.8917e-01, -3.4791e+00],
[-9.4960e-01, -3.1347e-01, -3.8459e+00],
[-9.9235e-01, -1.2348e-01, -3.9010e+00],
[-9.9756e-01, 6.9834e-02, -3.8737e+00],
[-9.6458e-01, 2.6380e-01, -3.9413e+00],
[-8.9600e-01, 4.4406e-01, -3.8634e+00],
[-8.1117e-01, 5.8481e-01, -3.2904e+00],
[-7.2605e-01, 6.8764e-01, -2.6718e+00],
[-6.5241e-01, 7.5787e-01, -2.0360e+00],
[-5.9990e-01, 8.0007e-01, -1.3476e+00],

[-5.8185e-01, 8.1330e-01, -4.4759e-01],
[-5.7624e-01, 8.1728e-01, -1.3762e-01],
[-6.0509e-01, 7.9615e-01, 7.1535e-01],
[-6.5373e-01, 7.5673e-01, 1.2525e+00],
[-7.0930e-01, 7.0491e-01, 1.5200e+00],
[-7.7766e-01, 6.2868e-01, 2.0487e+00],
[-8.4252e-01, 5.3866e-01, 2.2202e+00],
[-9.0320e-01, 4.2923e-01, 2.5042e+00],
[-9.5100e-01, 3.0918e-01, 2.5861e+00],
[-9.8286e-01, 1.8437e-01, 2.5780e+00],
[-9.9891e-01, 4.6600e-02, 2.7763e+00],
[-9.9530e-01, -9.6792e-02, 2.8712e+00],
[-9.7208e-01, -2.3467e-01, 2.7986e+00],
[-9.2977e-01, -3.6814e-01, 2.8026e+00],
[-8.7160e-01, -4.9021e-01, 2.7065e+00],
[-7.9953e-01, -6.0062e-01, 2.6389e+00],
[-7.3121e-01, -6.8215e-01, 2.1284e+00],
[-6.7374e-01, -7.3897e-01, 1.6168e+00],
[-6.3587e-01, -7.7180e-01, 1.0026e+00],
[-6.1465e-01, -7.8880e-01, 5.4372e-01],
[-6.1890e-01, -7.8547e-01, -1.0788e-01],
[-6.4360e-01, -7.6536e-01, -6.3698e-01],
[-6.8655e-01, -7.2708e-01, -1.1510e+00],
[-7.3965e-01, -6.7299e-01, -1.5163e+00],
[-8.0198e-01, -5.9735e-01, -1.9611e+00],
[-8.6193e-01, -5.0703e-01, -2.1691e+00],
[-9.2289e-01, -3.8505e-01, -2.7293e+00],
[-9.7029e-01, -2.4194e-01, -3.0181e+00],
[-9.9693e-01, -7.8302e-02, -3.3196e+00],
[-9.9715e-01, 7.5462e-02, -3.0783e+00],
[-9.7706e-01, 2.1298e-01, -2.7817e+00],
[-9.4483e-01, 3.2756e-01, -2.3820e+00],
[-8.9934e-01, 4.3725e-01, -2.3763e+00],
[-8.4675e-01, 5.3199e-01, -2.1684e+00],
[-7.9096e-01, 6.1187e-01, -1.9494e+00],
[-7.5118e-01, 6.6010e-01, -1.2505e+00],
[-7.2365e-01, 6.9016e-01, -8.1541e-01],
[-7.1958e-01, 6.9441e-01, -1.1779e-01],
[-7.2314e-01, 6.9070e-01, 1.0302e-01],
[-7.4624e-01, 6.6568e-01, 6.8105e-01],
[-7.8047e-01, 6.2520e-01, 1.0603e+00],
[-8.1739e-01, 5.7608e-01, 1.2292e+00],
[-8.5623e-01, 5.1659e-01, 1.4213e+00],
[-8.9273e-01, 4.5059e-01, 1.5087e+00],
[-9.2487e-01, 3.8028e-01, 1.5467e+00],
[-9.5489e-01, 2.9695e-01, 1.7719e+00],
[-9.7911e-01, 2.0334e-01, 1.9346e+00],
[-9.9335e-01, 1.1516e-01, 1.7871e+00],
[-9.9953e-01, 3.0741e-02, 1.6935e+00],
[-9.9848e-01, -5.5052e-02, 1.7165e+00],
[-9.9158e-01, -1.2948e-01, 1.4952e+00],
[-9.8298e-01, -1.8370e-01, 1.0981e+00],
[-9.6946e-01, -2.4524e-01, 1.2603e+00],
[-9.5750e-01, -2.8843e-01, 8.9642e-01],
[-9.5185e-01, -3.0657e-01, 3.8009e-01],
[-9.5230e-01, -3.0515e-01, -2.9834e-02],
[-9.5962e-01, -2.8131e-01, -4.9870e-01],
[-9.6749e-01, -2.5290e-01, -5.8968e-01],
[-9.7328e-01, -2.2964e-01, -4.7936e-01],
[-9.8254e-01, -1.8604e-01, -8.9159e-01],


```

state, reward, over, _ = env.step([action_continuous])
reward_sum += reward

#打印动画
if play:
    display.clear_output(wait=True)
    show()

return reward_sum

test(play=False)

```

Out[]: -1187.931817627959

```

In [ ]: def train():
    optimizer_actor = torch.optim.Adam(model_actor.parameters(), lr=2e-3)
    optimizer_critic = torch.optim.Adam(model_critic.parameters(), lr=1e-2)

    loss_fn = torch.nn.MSELoss()

    for i in range(2000):
        states, rewards, actions, next_states, overs = get_data()

        values = model_critic(states)

        targets = model_critic(next_states)

        targets *= 0.98

        targets *= 1 - overs

        targets += rewards

        delta = (values - targets).detach()

        probs = model_actor(states)

        probs = probs.gather(dim=1, index=actions)

        loss = (-probs.log() * delta).mean()

        loss_critic = loss_fn(values, targets.detach())

        optimizer_actor.zero_grad()
        loss.backward()
        optimizer_actor.step()

        optimizer_critic.zero_grad()
        loss_critic.backward()
        optimizer_critic.step()

    if i % 100 == 0:
        test_result = sum([test(play=False) for _ in range(10)]) / 10
        print(f"epoch:{i}, score:{test_result}")

train()

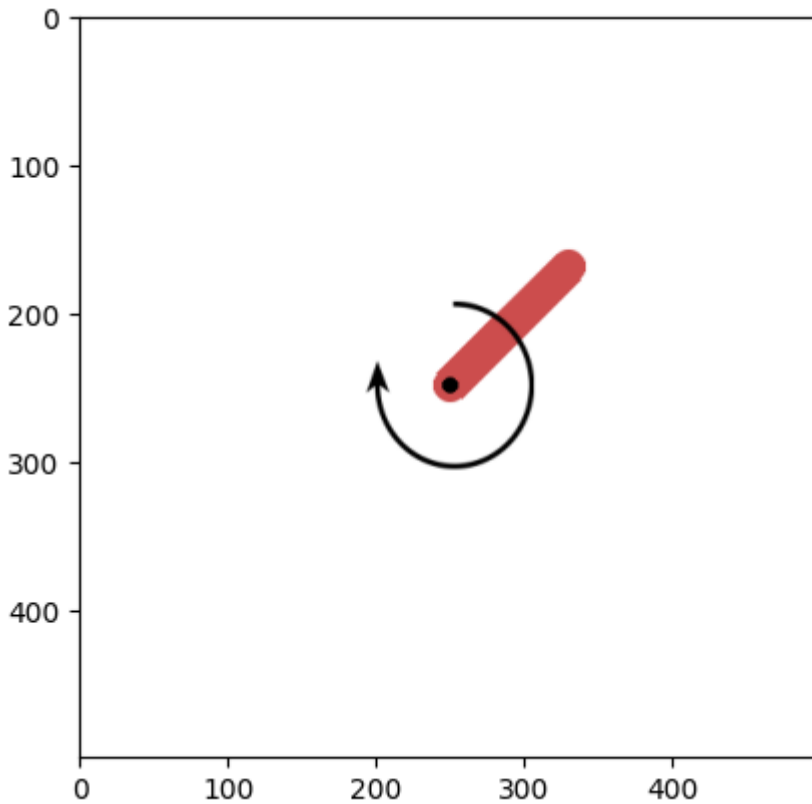
```

```
epoch:0, score: -1604.3220631118272  
epoch:100, score: -1610.8431765297357  
epoch:200, score: -1616.144260773333  
epoch:300, score: -1588.5297118263807  
epoch:400, score: -1590.3223259581275  
epoch:500, score: -1567.1966724326264  
epoch:600, score: -1604.0504779529751  
epoch:700, score: -1603.3357954987064  
epoch:800, score: -1607.7916256604726  
epoch:900, score: -1634.2924843349479  
epoch:1000, score: -1589.9032769812475  
epoch:1100, score: -1588.6364149038077  
epoch:1200, score: -1577.2657015663062  
epoch:1300, score: -1600.1749498167303  
epoch:1400, score: -1620.199190519759  
epoch:1500, score: -1606.7593020980962  
epoch:1600, score: -1642.8911228827506  
epoch:1700, score: -1609.6638874609766  
epoch:1800, score: -1598.0150007838213  
epoch:1900, score: -1596.7344934484488
```

```
In [ ]: model_actor(torch.tensor([env.reset()]))
```

```
Out[ ]: tensor([[0.0331, 0.0298, 0.0122, 0.0099, 0.0070, 0.0142, 0.0229, 0.0075, 0.0672,  
                0.0060, 0.7902]], grad_fn=<SoftmaxBackward0>)
```

```
In [ ]: test(play=True)
```



```
Out[ ]: -1598.2854908945142
```