HW1

一、环境配置

1.1 conda环境创建

`conda create -n gym python=3.8`

1.2 安装gym

直接`pip install -r requirements.txt`会遇到`extras_require`的报错，发现是pip setuptools版本不匹配的问题，将setuptools降级到65.5.0即可安装，或者使用conda也可以。安装后运行experiment1.py时会报错找不到pyglet库，直接安装会出现版本不匹配的问题，安装1.5.0版本即可解决。

二、CartPole-v0环境测试

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

测试如下：

```
Ep.: 100 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  175.240
Ep.: 101 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  175.320
Ep.: 102 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  176.780
Ep.: 103 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  177.960
Ep.: 104 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  179.850
Ep.: 105 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  181.770
Ep.: 106 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  183.450
Ep.: 107 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  185.350
Ep.: 108 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  187.260
Ep.: 109 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  189.160
Ep.: 110 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  190.820
Ep.: 111 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  192.740
Ep.: 112 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  194.630
Ep.: 113 , timesteps: 200, noise_scale (Gt >= best(Gt)): 0.0010, Gt 86.6020,    Avg.Score:  196.210
Environment solved in 113 episodes!     Average Score: 196.21
```

三、Acrobot-v1

```
### Rewards

    The goal is to have the free end reach a designated target height in as few
steps as possible,
    and as such all steps that do not reach the goal incur a reward of -1.
    Achieving the target height results in termination with a reward of 0. The
reward threshold is -100.
```

根据官方文档将阈值设置为-100，实际上默认的也是-100：

```
register(
    id="Acrobot-v1",
    entry_point="gym.envs.classic_control.acrobot:AcrobotEnv",
    reward_threshold=-100.0,
    max_episode_steps=500,
)
```

根据官方文档给出的观察空间和动作空间的维数，调整对应参数：

Action Space

The action is discrete, deterministic, and represents the torque applied on the actuated joint between the two links.

Num	Action	Unit
0	apply -1 torque to the actuated joint	torque (N m)
1	apply 0 torque to the actuated joint	torque (N m)
2	apply 1 torque to the actuated joint	torque (N m)

Observation Space

The observation is a `ndarray` with shape `(6,)` that provides information about the two rotational joint angles as well as their angular velocities:

Num	Observation	Min	Max
0	Cosine of `theta1`	-1	1
1	Sine of `theta1`	-1	1
2	Cosine of `theta2`	-1	1
3	Sine of `theta2`	-1	1
4	Angular velocity of `theta1`	~ -12.567 (-4 * pi)	~ 12.567 (4 * pi)
5	Angular velocity of `theta2`	~ -28.274 (-9 * pi)	~ 28.274 (9 * pi)

```
class Policy():
    def __init__(self, s_size=6, a_size=3):
        self.w = 1e-4*np.random.rand(s_size, a_size)  # weights for simple linear
policy: state_space x action_space

    def forward(self, state):
        x = np.dot(state, self.w)
        return np.exp(x)/sum(np.exp(x))

    def act(self, state):
        probs = self.forward(state)
        action = np.argmax(probs)
        return action
```

测试结果如下：

```
Ep.: 589 , timesteps:  99, noise_scale (Gt < best(Gt)): 2.0000, Gt -62.6536,    Avg.Score:  -102.360
Ep.: 590 , timesteps: 107, noise_scale (Gt < best(Gt)): 2.0000, Gt -65.5388,    Avg.Score:  -102.370
Ep.: 591 , timesteps:  66, noise_scale (Gt < best(Gt)): 2.0000, Gt -47.9659,    Avg.Score:  -102.190
Ep.: 592 , timesteps:  83, noise_scale (Gt < best(Gt)): 2.0000, Gt -56.1382,    Avg.Score:  -101.590
Ep.: 593 , timesteps:  92, noise_scale (Gt < best(Gt)): 2.0000, Gt -59.9315,    Avg.Score:  -101.660
Ep.: 594 , timesteps:  93, noise_scale (Gt < best(Gt)): 2.0000, Gt -60.3322,    Avg.Score:  -101.830
Ep.: 595 , timesteps:  99, noise_scale (Gt < best(Gt)): 2.0000, Gt -62.6536,    Avg.Score:  -101.960
Ep.: 596 , timesteps:  67, noise_scale (Gt < best(Gt)): 2.0000, Gt -48.4863,    Avg.Score:  -101.740
Ep.: 597 , timesteps: 102, noise_scale (Gt < best(Gt)): 2.0000, Gt -63.7628,    Avg.Score:  -101.470
Ep.: 598 , timesteps:  66, noise_scale (Gt < best(Gt)): 2.0000, Gt -47.9659,    Avg.Score:  -101.230
Ep.: 599 , timesteps:  66, noise_scale (Gt < best(Gt)): 2.0000, Gt -47.9659,    Avg.Score:  -100.980
Ep.: 600 , timesteps:  88, noise_scale (Gt < best(Gt)): 2.0000, Gt -58.2879,    Avg.Score:  -100.830
Ep.: 601 , timesteps:  66, noise_scale (Gt < best(Gt)): 2.0000, Gt -47.9659,    Avg.Score:  -100.560
Ep.: 602 , timesteps:  65, noise_scale (Gt >= best(Gt)): 1.0000, Gt -47.4404,   Avg.Score:  -96.200
Environment solved in 602 episodes!     Average Score: -96.20
```