Homework 1: Multi-Armed Bandit

劉安得

106061151

1. Implementation

In ε-Greedy, if the action values are equal, select randomly from among all the actions with equal probability.

- give an action, receive a reward

  the reward should be given by a normal distribution with mean $q*(n)$ and variance 1.

```
reward = rd.normal(self.mean[action], self.variance)
```

- chooseAction

  It is better to always choose the best option, but to keep the exploration going, we sometimes choose random action with a probability of ϵ.

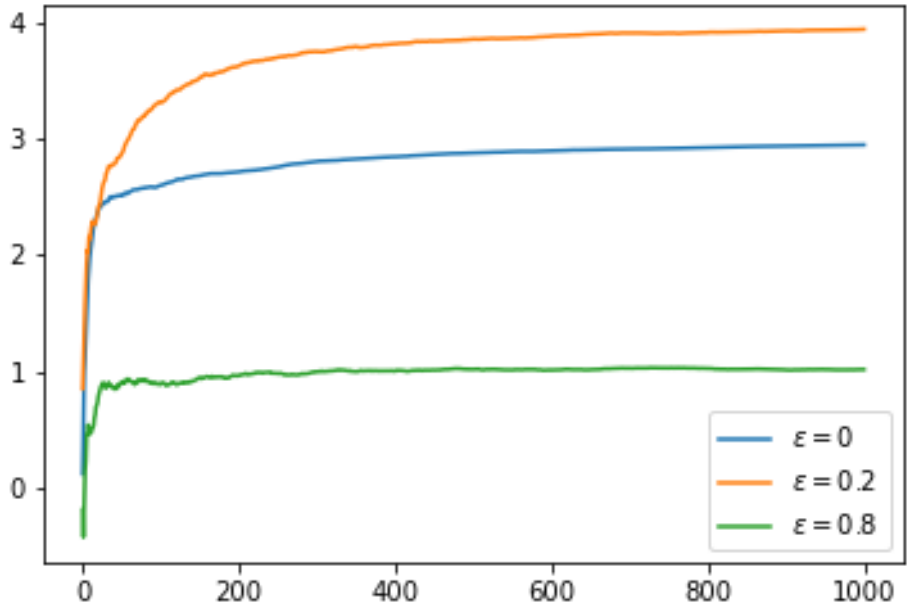
```
prob = rd.random()

if prob < epsilon:
    action = rd.randint(self.env.banditNums)
else:
    action = rd.choice(np.flatnonzero(self.actionValues ==
np.amax(self.actionValues)))
```

- updateValue

  After receiving the reward, we will update the action value. In this homework we simply use the average reward as the action value.

```
self.actionValues[action] += (reward-
self.actionValues[action])/self.chosenTimes[action]
```

2. Experiments and Analysis



When ε=0.2, ε-Greedy has the best performance, its average rewards can reach about 4 at the end of steps. However, when ε=0, because it lacks exploration, the algorithm sometimes misestimates the action values; when ε=0.8, although it has great exploration ability, the algorithm selects randomly too often.

If the algorithm randomly samples several time before estimating action values, it could always get the best result when $\varepsilon = 0$.