Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)AY: 2023 - 24

Bhuvi Ghosh
60009210191

Subject: Reinforcement Learning
Experiment 2
The Upper Confidence Bound (UCB) Bandit Algorithm

AIM:
To solve the Bandit problem using the Upper Confidence Bound (UCB1) algorithm

THEORY:

A highly effective multi-armed bandit strategy is the Upper Confidence Bounds (UCB1) strategy. Rather than performing exploration by simply selecting an arbitrary action, chosen with a probability that remains constant, the UCB algorithm changes its exploration-exploitation balance as it gathers more knowledge of the environment.

Using the UCB1 strategy, we select the next action using the following:

$$A_t = \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\log t}{N_t(a)}} \right]$$

Where;

Q (a) is the estimated value of action 'a' at time step 't'.

N (a) is the number of times that action 'a' has been selected, prior to time 't'.'c' is a

confidence value that controls the level of exploration.

Exploitation:

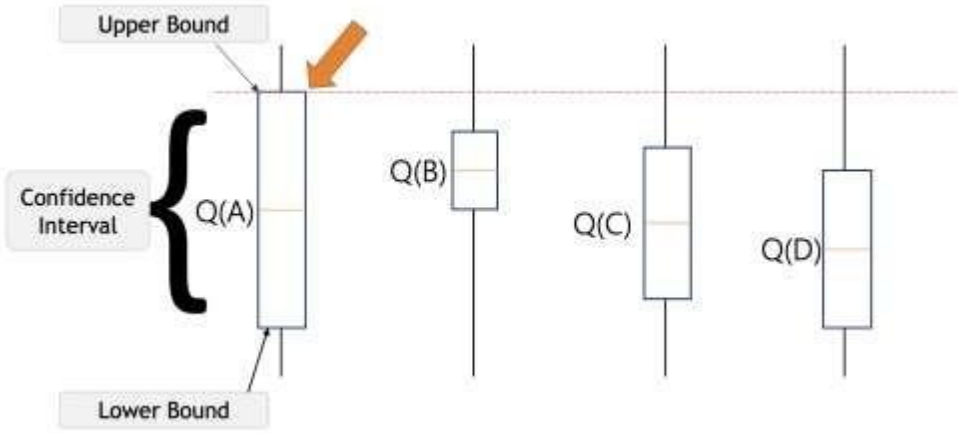
- Q (a) represents the exploitation part of the equation. UCB is based on the principle of "optimism in the fact of uncertainty", which basically means if you don't know which action is best then choose the one that currently looks to be the best. Taking this half of the equation by itself will do exactly that: the action that currently has the highest estimated reward will be the chosen action.

Exploration:

- The second half of the equation adds exploration, with the degree of exploration being controlled by the hyper-parameter 'c'. Effectively this part of the equation provides a measure of the uncertainty for the action's reward estimate.

- If an action has not been tried very often, or not at all, then N (a) will be small. Consequently, the uncertainty term will be large, making this action more likely to be selected. Every time an action is taken, we become more confident about its estimate. In this case N (a) increments, and so the uncertainty term decreases, making it less likely that this action will be selected as a result of exploration (although it may still be selectedas the action with the highest value, due to the exploitation term).

For example, let us say we have these four actions with associated uncertainties in the picture below, our agent has no idea which is the best action. So according to the UCB algorithm, it will optimistically pick the action that has the highest upper bound i.e., A. By doing this either it will have the highest value and get the highest reward, or by taking that we will get to learn about anaction we know least about.



ALGORITHM:

Input: N arms, number of rounds T ≥ N

1. For t = 1 . . . N, play arm t.
2. For t = N + 1 . . . T, play arm

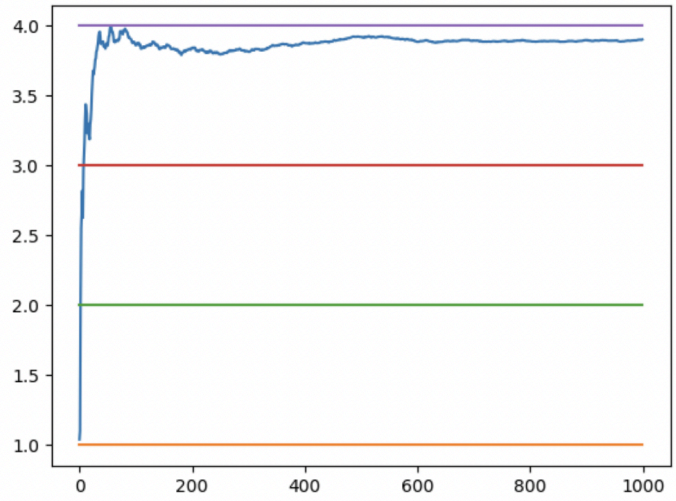
$$I_t = \underset{i \in \{1 \ldots N\}}{\arg\max} \, \mathrm{UCB}_{i,t-1}.$$

LAB ASSIGNMENT TO DO:

1. For ‘n’ arms, implement the UCB1 algorithm and calculate the overall reward.
2. Plot a graph of the UCB values for ‘n’ arms across k time steps and record the observations.

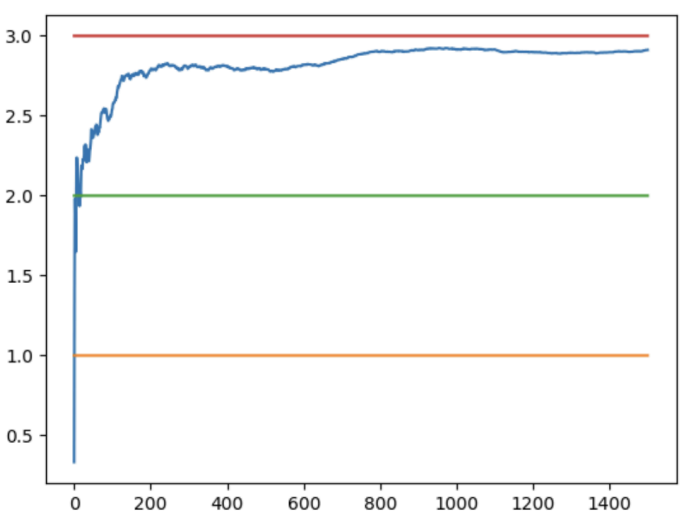
```python
import numpy as np
import matplotlib.pyplot as plt

class Action:
    def __init__(self, m):
        self.m = m
        self.mean = 0
        self.N = 0

    def select(self):
        return np.random.normal(0, 1) + self.m

    def update(self, x):
        self.N += 1
        self.mean = self.mean + (1 / self.N) * (x - self.mean)

def n_arm_bandit_ucb(ms, c, N):
    n = len(ms)
    actions = [Action(m) for m in ms]
    data = np.empty(N)

    for i in range(N):
        j = np.argmax([a.mean + c * np.sqrt(np.log(i) / a.N) for a in actions])
        x = actions[j].select()
        actions[j].update(x)
        data[i] = x

    cumulative_average = np.cumsum(data) / (np.arange(N) + 1)
    plt.plot(cumulative_average)
    for m in ms:
        plt.plot(np.ones(N) * m)
    plt.show()

    for i, action in enumerate(actions):
        print(f"Number of times arm {i + 1} selected: {action.N}")
        print(f"Mean of rewards from arm {i + 1}: {action.mean}")


means = [1, 2, 3, 4]
exploration_parameter = 2
```

```
iterations = 1000
n_arm_bandit_ucb(means, exploration_parameter, iterations)
```



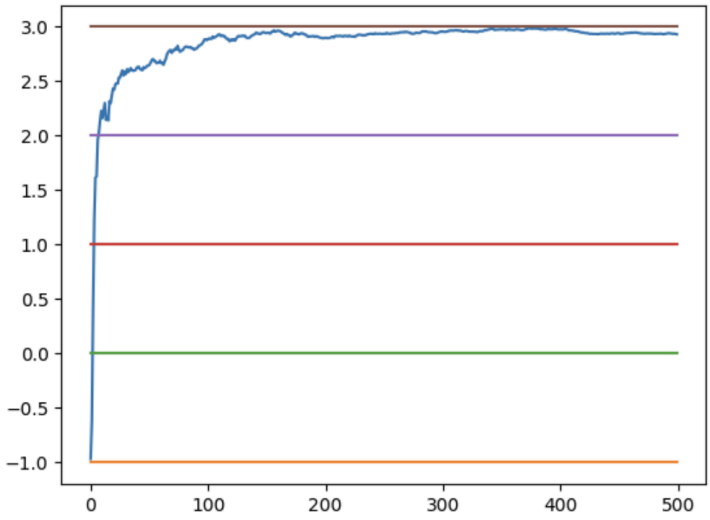

```
Number of times arm 1 selected: 6
Mean of rewards from arm 1: 1.7291379756217724
Number of times arm 2 selected: 4
Mean of rewards from arm 2: 1.409901971254012
Number of times arm 3 selected: 22
Mean of rewards from arm 3: 2.9769007104045926
Number of times arm 4 selected: 968
Mean of rewards from arm 4: 3.9479716624433543
```

```
means_example1 = [1, 2, 3]
exploration_parameter_example1 = 3
iterations_example1 = 1500
n_arm_bandit_ucb(means_example1, exploration_parameter_example1, iterations_example1)
```



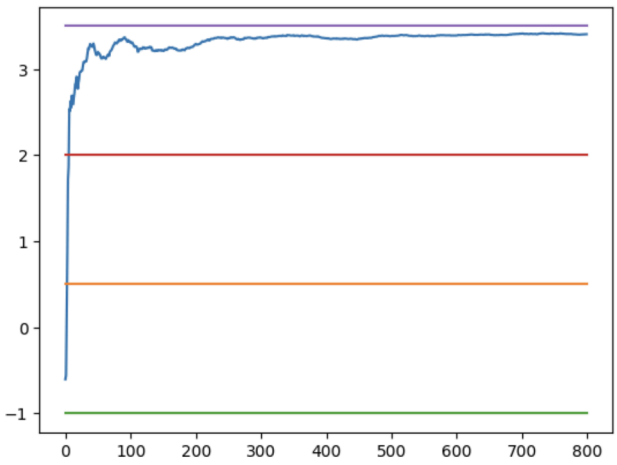

```
Number of times arm 1 selected: 13
Mean of rewards from arm 1: 0.8566195256461153
Number of times arm 2 selected: 62
Mean of rewards from arm 2: 2.141644993090686
Number of times arm 3 selected: 1425
Mean of rewards from arm 3: 2.9643480878716755
```

```
means_example2 = [-1, 0, 1, 2, 3]
exploration_parameter_example2 = 0.5
iterations_example2 = 500
n_arm_bandit_ucb(means_example2, exploration_parameter_example2, iterations_example2)
```




```
Number of times arm 1 selected: 1
Mean of rewards from arm 1: -0.9653203966256666
Number of times arm 2 selected: 1
Mean of rewards from arm 2: -0.25826890899755367
Number of times arm 3 selected: 2
Mean of rewards from arm 3: 2.005170080798045
Number of times arm 4 selected: 5
Mean of rewards from arm 4: 2.3339756420850244
Number of times arm 5 selected: 491
Mean of rewards from arm 5: 2.9562340693490494
```

```
means_example4 = [0.5, -1, 2, 3.5]
exploration_parameter_example4 = 2
iterations_example4 = 800
n_arm_bandit_ucb(means_example4, exploration_parameter_example4, iterations_example4)
```




```
Number of times arm 1 selected: 2
Mean of rewards from arm 1: -0.9924469352788367
Number of times arm 2 selected: 2
Mean of rewards from arm 2: -0.9891768676566017
Number of times arm 3 selected: 9
Mean of rewards from arm 3: 1.8417903800032491
Number of times arm 4 selected: 787
Mean of rewards from arm 4: 3.446687747924048
```