Balancing Cart-pole using Hill Climbing method

In Hill Climb we use random initialization of weights and at every episode add some noise to it and keep those weights if the agent improves.

Initialize:

1. Importing different libraries numpy, random, gym.

2. Initialize cart pole environment provided by OpenAI.

3. Initialize the number of state and actions for the environment.

4. Initialize total episodes to iterate for = 50 and target value = 200

Function Build Model :

1. Generate weight matrix(4x2) and noise scale with random small values.
2. Best Reward as the minimum value possible and Best weight to be current weight.

Function get Action :

1. Dot product of state matrix(1x4) and Weights(4x2) which provides us the action matrix(1x2).
2. Returns us which action to perform, that is left(0) or right(1).

Function update Model :

1. Update best reward with new reward if it is greater than best, Also save the weights and reduce noise(in our case we have reduced to half).
2. If new reward is not greater then best reward, we increase noise and therefore look for other options/regions.
3. Lastly, Update the weights with summation of best weights and scaled noise.

At end, Calculating the total reward for each episode and printing each episode and its respective reward earned for 50 episodes.

Additionally, displaying the episodes which reached the target value successfully.

Output : Hill Climb for 'CartPole-v0' environment(Total episodes=30)

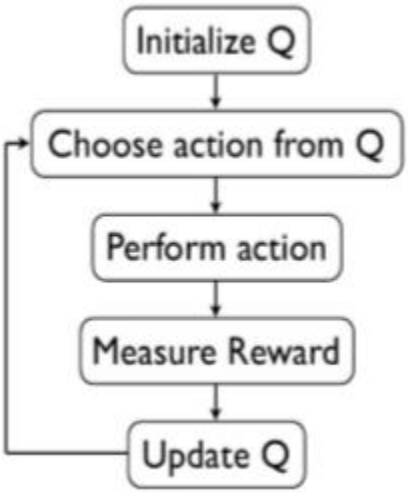
```
Observation space: Box(-3.4028234663852886e+38, 3.4028234663852886e+38, (4,), float32)
Action space: Discrete(2)
Episode Number : 1, Rewards(Total): 9.00
Episode Number : 2, Rewards(Total): 8.00
Episode Number : 3, Rewards(Total): 200.00
Episode Number : 4, Rewards(Total): 200.00
Episode Number : 5, Rewards(Total): 200.00
Episode Number : 6, Rewards(Total): 200.00
Episode Number : 7, Rewards(Total): 200.00
Episode Number : 8, Rewards(Total): 200.00
Episode Number : 9, Rewards(Total): 200.00
Episode Number : 10, Rewards(Total): 200.00
Episode Number : 11, Rewards(Total): 200.00
Episode Number : 12, Rewards(Total): 134.00
Episode Number : 13, Rewards(Total): 96.00
Episode Number : 14, Rewards(Total): 200.00
Episode Number : 15, Rewards(Total): 200.00
Episode Number : 16, Rewards(Total): 200.00
Episode Number : 17, Rewards(Total): 200.00
Episode Number : 18, Rewards(Total): 200.00
Episode Number : 19, Rewards(Total): 172.00
Episode Number : 20, Rewards(Total): 200.00
Episode Number : 21, Rewards(Total): 200.00
Episode Number : 22, Rewards(Total): 200.00
Episode Number : 23, Rewards(Total): 200.00
Episode Number : 24, Rewards(Total): 200.00
Episode Number : 25, Rewards(Total): 200.00
Episode Number : 26, Rewards(Total): 200.00
Episode Number : 27, Rewards(Total): 200.00
Episode Number : 28, Rewards(Total): 200.00
Episode Number : 29, Rewards(Total): 200.00
Episode Number : 30, Rewards(Total): 200.00
Episode(s) exceeding rewards: 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30
```

Output : Hill Climb 'CartPole-v1' environment(Total episodes=30)

```
Observation space: Box(-3.4028234663852886e+38, 3.4028234663852886e+38, (4,), float32)
Action space: Discrete(2)
Episode Detail:: 1, Rewards(Total): 8.00
Episode Detail:: 2, Rewards(Total): 9.00
Episode Detail:: 3, Rewards(Total): 10.00
Episode Detail:: 4, Rewards(Total): 10.00
Episode Detail:: 5, Rewards(Total): 10.00
Episode Detail:: 6, Rewards(Total): 10.00
Episode Detail:: 7, Rewards(Total): 10.00
Episode Detail:: 8, Rewards(Total): 9.00
Episode Detail:: 9, Rewards(Total): 9.00
Episode Detail:: 10, Rewards(Total): 9.00
Episode Detail:: 11, Rewards(Total): 9.00
Episode Detail:: 12, Rewards(Total): 128.00
Episode Detail:: 13, Rewards(Total): 86.00
Episode Detail:: 14, Rewards(Total): 35.00
Episode Detail:: 15, Rewards(Total): 10.00
Episode Detail:: 16, Rewards(Total): 500.00
Episode Detail:: 17, Rewards(Total): 12.00
Episode Detail:: 18, Rewards(Total): 50.00
Episode Detail:: 19, Rewards(Total): 500.00
Episode Detail:: 20, Rewards(Total): 500.00
Episode Detail:: 21, Rewards(Total): 500.00
Episode Detail:: 22, Rewards(Total): 500.00
Episode Detail:: 23, Rewards(Total): 500.00
Episode Detail:: 24, Rewards(Total): 500.00
Episode Detail:: 25, Rewards(Total): 500.00
Episode Detail:: 26, Rewards(Total): 500.00
Episode Detail:: 27, Rewards(Total): 500.00
Episode Detail:: 28, Rewards(Total): 500.00
Episode Detail:: 29, Rewards(Total): 500.00
Episode Detail:: 30, Rewards(Total): 500.00
Episode(s) exceeding rewards: 16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30
```

Balancing Cart-pole using simple Q-Learning Method

Q-Learning does not have any model/network, It generates a Q-table wherein Q stands for Quality and comprises of State-Action pairs.



$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)$$

Initialize:

1. Importing different libraries numpy, random, gym and math.
2. Initialize cart pole environment provided by OpenAI.
3. Initialize the number of state and actions for the environment.
4. Initialize total episodes to iterate for = 200 and target value = 200
5. Initialize Q-Table with 0 and Discount factor = 0.99(as remains in same world)
6. Applied Logarithmic decay to both learning and exploration rate.

Function Choose action :

1. If the random value is greater than exploration rate, it will return the best action from the Q-Table for the particular state.

Function State to Bucket :

1. Used to update the continuous states to discrete states.
2. If state value is less than lower range, It is updated to 0
3. If state value is greater than higher range, value -1

Function QAgent :

1. Get explore and learning rate as per time step.
2. Get action for random using choose action function.
3. Get best Q value and update it in table.
4. Set state to next state and repeat again.

At end, Displaying each episode with its respective time step for which pole balanced upright during 200 episodes.

Additionally, displaying the episodes which reached the target value successfully.

Output : Q-Learning 'CartPole-v1' environment(Total episodes=200)

```
Episode 0 finished after 11.000000 time steps
Episode 1 finished after 8.000000 time steps
Episode 2 finished after 33.000000 time steps
Episode 3 finished after 11.000000 time steps
Episode 4 finished after 40.000000 time steps
Episode 5 finished after 27.000000 time steps
Episode 6 finished after 31.000000 time steps
Episode 7 finished after 27.000000 time steps
Episode 8 finished after 12.000000 time steps
Episode 9 finished after 16.000000 time steps
Episode 10 finished after 11.000000 time steps
Episode 11 finished after 7.000000 time steps
Episode 12 finished after 46.000000 time steps
Episode 13 finished after 16.000000 time steps
Episode 14 finished after 60.000000 time steps
Episode 15 finished after 20.000000 time steps
…

..

..

Episode 190 finished after 249.000000 time steps
Episode 191 finished after 249.000000 time steps
Episode 192 finished after 249.000000 time steps
Episode 193 finished after 249.000000 time steps
```

```
Episode 194 finished after 249.000000 time steps
Episode 195 finished after 249.000000 time steps
Episode 196 finished after 249.000000 time steps
Episode 197 finished after 249.000000 time steps
Episode 198 finished after 249.000000 time steps
Episode 199 finished after 249.000000 time steps

------successful Episodes--------

Episode 89 finished after 207.000000 time steps
Episode 140 finished after 249.000000 time steps
Episode 141 finished after 220.000000 time steps
Episode 142 finished after 247.000000 time steps
Episode 143 finished after 249.000000 time steps
Episode 148 finished after 239.000000 time steps
Episode 149 finished after 249.000000 time steps
Episode 150 finished after 204.000000 time steps
Episode 151 finished after 249.000000 time steps
Episode 152 finished after 249.000000 time steps
Episode 153 finished after 249.000000 time steps
Episode 154 finished after 249.000000 time steps
Episode 155 finished after 249.000000 time steps
Episode 156 finished after 249.000000 time steps
Episode 157 finished after 209.000000 time steps
Episode 158 finished after 249.000000 time steps
Episode 160 finished after 249.000000 time steps
Episode 161 finished after 249.000000 time steps
Episode 162 finished after 249.000000 time steps
Episode 163 finished after 249.000000 time steps
Episode 164 finished after 249.000000 time steps
Episode 165 finished after 249.000000 time steps
Episode 166 finished after 249.000000 time steps
Episode 167 finished after 249.000000 time steps
Episode 168 finished after 249.000000 time steps
Episode 169 finished after 249.000000 time steps
Episode 170 finished after 249.000000 time steps
Episode 171 finished after 249.000000 time steps
Episode 172 finished after 249.000000 time steps
Episode 173 finished after 249.000000 time steps
Episode 174 finished after 249.000000 time steps
Episode 175 finished after 249.000000 time steps
Episode 176 finished after 249.000000 time steps
Episode 177 finished after 249.000000 time steps
Episode 178 finished after 249.000000 time steps
Episode 179 finished after 249.000000 time steps
Episode 180 finished after 249.000000 time steps
Episode 181 finished after 249.000000 time steps
Episode 182 finished after 249.000000 time steps
Episode 183 finished after 249.000000 time steps
Episode 184 finished after 249.000000 time steps
Episode 185 finished after 249.000000 time steps
Episode 186 finished after 249.000000 time steps
Episode 187 finished after 249.000000 time steps
Episode 188 finished after 249.000000 time steps
```

```
Episode 189 finished after 249.000000 time steps
Episode 190 finished after 249.000000 time steps
Episode 191 finished after 249.000000 time steps
Episode 192 finished after 249.000000 time steps
Episode 193 finished after 249.000000 time steps
Episode 194 finished after 249.000000 time steps
Episode 195 finished after 249.000000 time steps
Episode 196 finished after 249.000000 time steps
Episode 197 finished after 249.000000 time steps
Episode 198 finished after 249.000000 time steps
Episode 199 finished after 249.000000 time steps
```

Note :

We have limited the max time step for each iteration to 250 in order to save processing time when dealing with 'CartPole-v1' which default has 500 time steps.