StarAi: Deep Reinforcement Learning

Lesson 1: Epsilon-Greedy

Lesson 1: Epsilon-Greedy

Part 1: The Multi Armed bandit problem

Lesson 1: Objectives

1. Toy problem: The multi-armed bandit
2. Explain exploration vs exploitation
3. Introduce how epsilon greedy solves EvE
4. Introduction to OpenAi gym & why it is important.
5. Solve the multi-armed bandit problem with OpenAi gym

Why Epsilon Greedy?

Epsilon Greedy will act as one of the “pillars” in implementing more complex RL algorithms.

What the shell is a bandit?

So what the shell is a “multi armed bandit”?

One definition of Reinforcement Learning

"Finding the optimal **strategy** to solving a problem in the face of massive uncertainty."

One definition of Reinforcement Learning

"Finding the optimal strategy to solving a problem in the face of **massive uncertainty**."

Actual photo of me driving to work

In reinforcement learning the name given to the ***strategy*** that we are following to solve a given problem is called a ***policy***.

Policy Example #1: Following the "policy" of not stopping at traffic lights

We would like to determine the best slot machine to use (the best policy), given our uncertainty in each slot machine's payout.

Lesson 1: Epsilon-Greedy

Part 2: Exploration vs. Exploitation

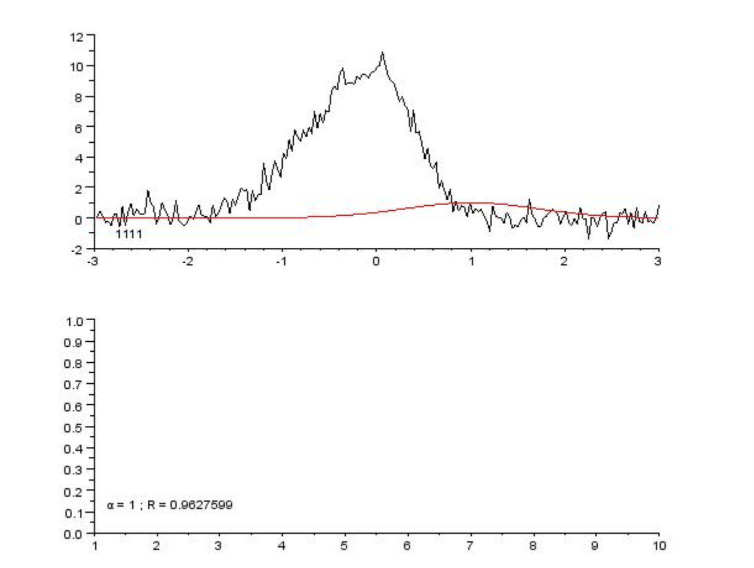
The meaning of life? - EvE

Exploration vs Exploitation: Example 2

VS

Being in a job vs searching for a new one

Exploration vs Exploitation: Example 3



VS

Using an existing trading algorithm vs searching for a new algorithm

Lesson 1: Epsilon-Greedy

Part 3: The epsilon greedy algorithm

Simply, the epsilon Greedy algorithm is this:

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Repeat forever:

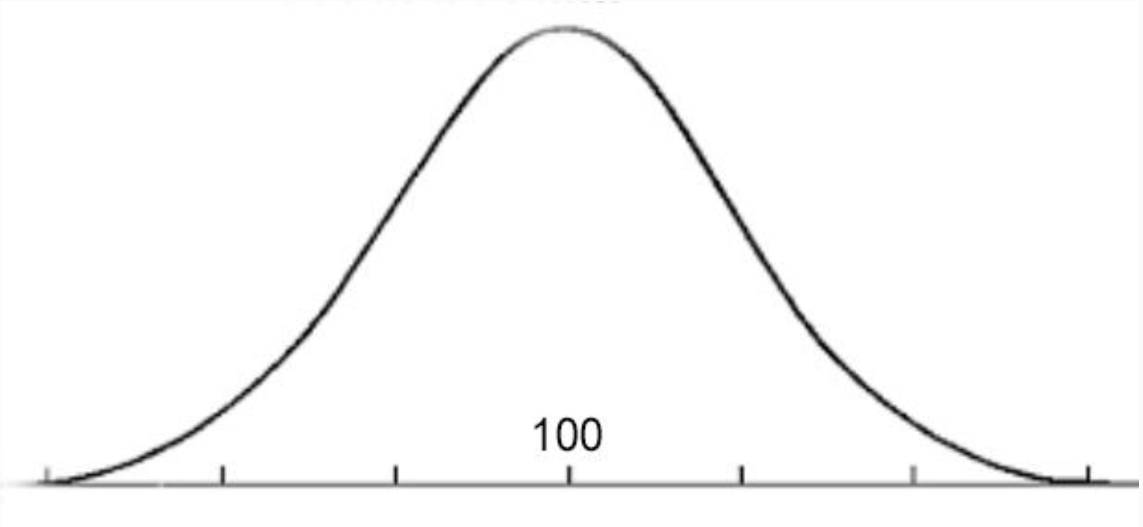
$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$

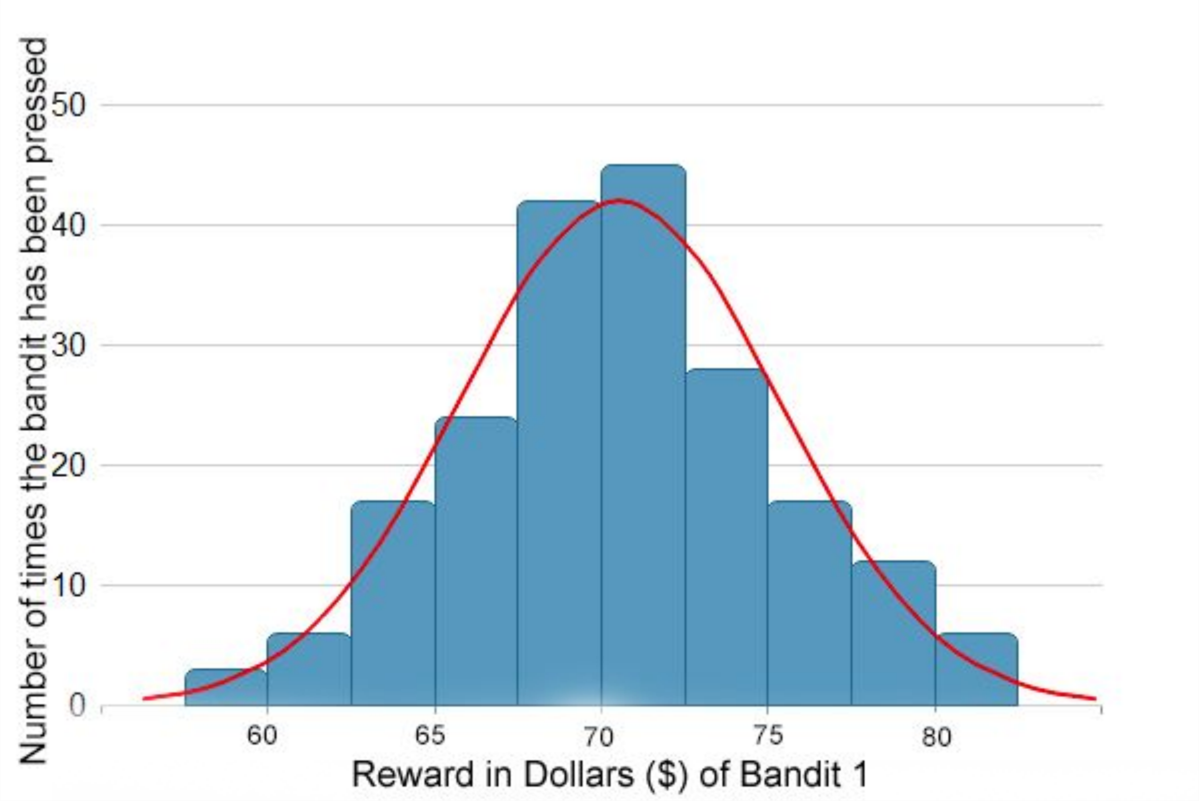
$N(A) \leftarrow N(A) + 1$

$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\left[R - Q(A)\right]$

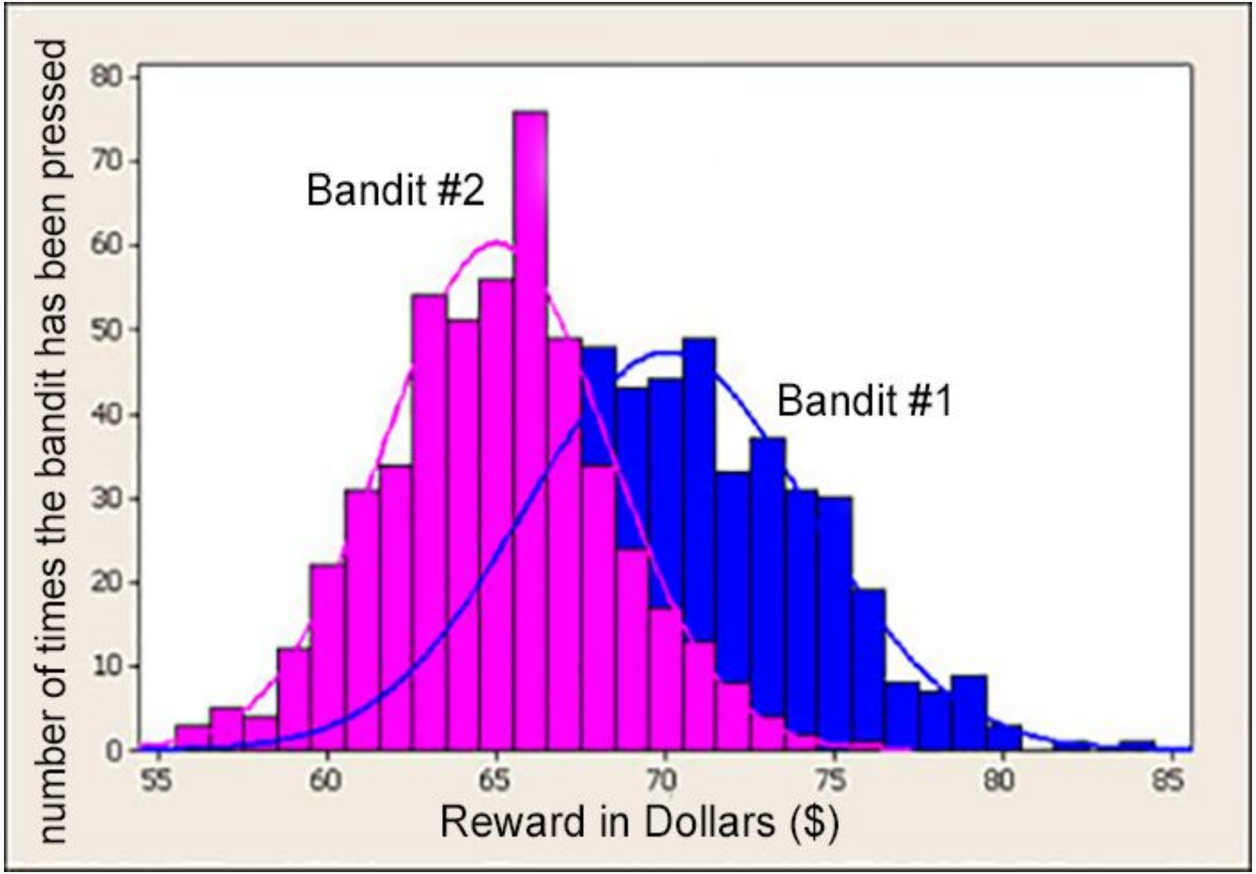
The Bell Curve , in machine learning we call it the Normal Distribution



We “pull” Bandit 1 many times and we get:



Adding Bandit #2, which has a different distribution.



Epsilon is a fancy name for this symbol

ε

Epsilon is the probability of exploration to exploitation

ε = prob. of

loitation = 1

Defining epsilon, continued

When $\varepsilon = 1$

Exploration is maximized

Choose actions at random.

When $\varepsilon = 0$

Exploitation is maximized

Choose the best action.

But how do we control Epsilon?

We can subtract *any* mathematical function from 1 to “scale” Epsilon, like so:

Where: $$\varepsilon = 1 - f(x)$$

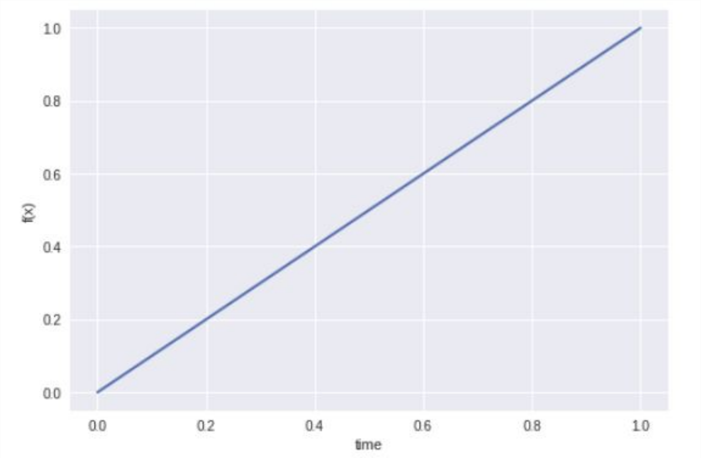
Remember:

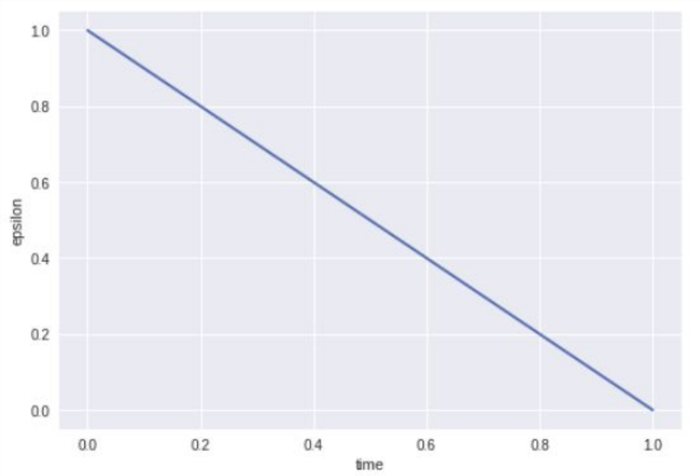
$\varepsilon = 1$ Exploration is maximized.

$\varepsilon = 0$ Exploitation is maximized

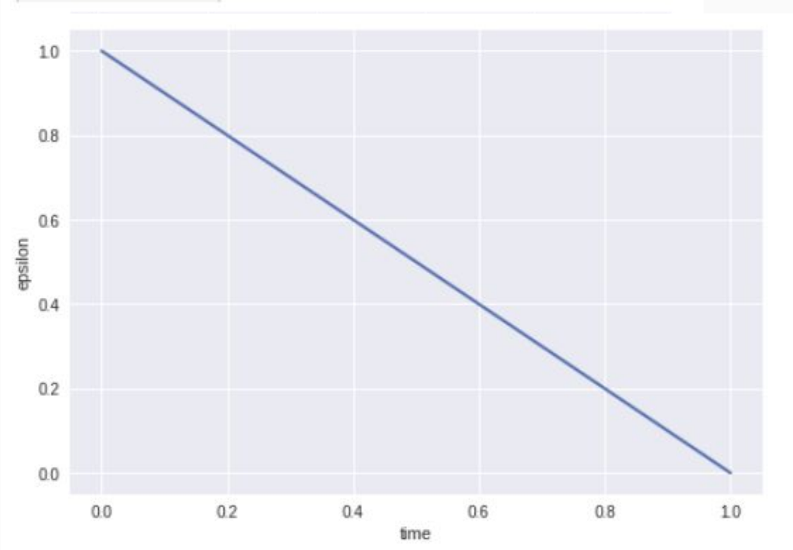
So what function can we use for f(x)? for this example let's use the most complex function imaginable: a straight line.

$$\varepsilon = 1 - f(x)$$

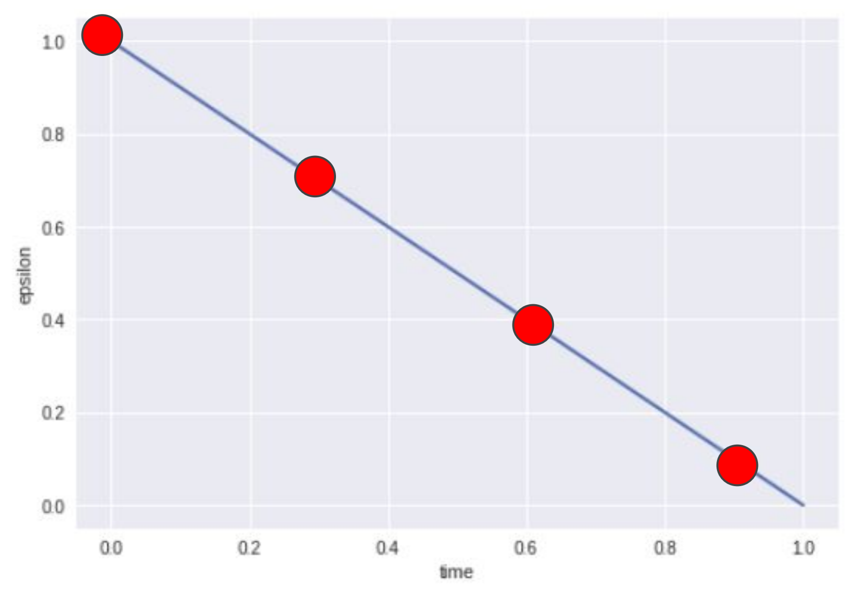




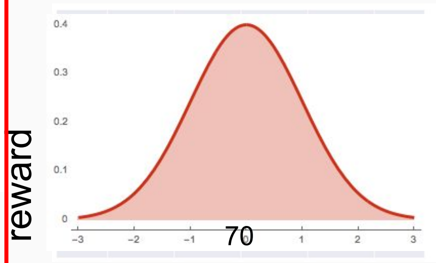
We can think of Epsilon as the volume knob controlling how much exploration we do.



Putting it all together: how epsilon works in the multi-armed bandit problem

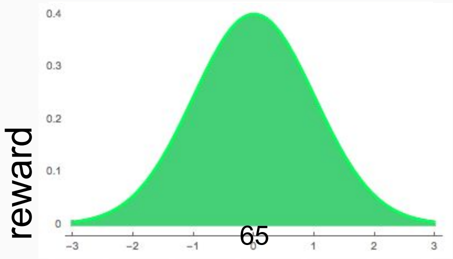


Bandit # 1



Avg.Payout $ 70

Bandit # 2



Avg.Payout $ 65

Lesson 1: Epsilon-Greedy

Part 4: Brief introduction to OpenAi Gym & why it important.

Enter OpenAi Gym

Lesson 1: Epsilon-Greedy

Part 5: Let’s implement your first precursor RL algorithm algorithm - Epsilon-Greedy - in OpenAi Gym