1. Example 1:

Chess game

State: during a game, the location of the chess units

Action: make a move

Reward: win (+1) lose (-1)

The limitation of this example is that simply by assigning rewards to winning and losing, the agent is not learning strategies of winning the game (or by a very slow pace)

Example 2:

Robbery

State: your inventory (full or not)

Action: select an item to rob

Reward: the item itself (the value of the item)

In this example, the burglar (agent) tends to select relatively smaller or lighter items with high values for multiple runs (assuming the burglar is not arrested)

Example 3:

On your way home

State: at home or not

Action: at a crossroad, select a direction to go

Reward: for shorter time spent on going home, the higher the reward

In this example, once you find the shorter path which allows you going home quicker, you will be more likely pick this path for the future

1. Similar to playing a game, due to the complexity of a maze with lots of dead ends, simply by setting a reward at the end does not show any improvement on reaching a goal. The agent escaping the maze will still attempt to try the routes that do not lead to the exit, thus it inefficiently attempts to escape.

To effectively communicate to the agent, a better design of reward would be setting a reward every time the agent moves to a location that leads to the exit, then if the agent tries to escape the maze for the next run (another episode), it will learn to pick the path leading to the rewards.

1. Gt = Rt+1 + γRt+2 + γ2Rt+3 + γ3Rt+4 + γ4Rt+5 + …

= Rt+1 + γGt+1

R1 = -1 R2 = 2 R3 = 6 R4 = 3 R5 = 2 γ = 0.5

G5 = 0 G4 = 2 G3 = 3 + 0.5(2) = 4 G2 = 6 + 0.5(4) = 8 G1 = 2 + 0.5(8) = 6

G0 = -1 + 0.5(6) = 2

1. R1 = 2 R>1 = 7 γ = 0.9

G1 = 7 / (1 – 0.9) = 70 G0 = 2 + 0.9(70) = 65

1. vπ(s)

= (1/4)[0 + 0.9(0.7)] + (1/4)[0 + 0.9(-0.6)] + (1/4)[0 + 0.9(-1.2)] + (1/4)[0 + 0.9(-0.4)]

= (1/4)[0.9(-1.5)] = (1/4)(-1.35) ← round to nearest 10-1

= (1/4)(-1.4) = -0.35 ← round to nearest 10-1

= -0.4

1. qπ(s,a) = ∑a π(a|s) ∑s’,r p(s’,r|s,a) [r + γqπ(s’,a’)]
2. The signs are very important, since the reward value is the motivation of agent.

Gt = ∑k = 0∞ γkRt+k+1

= ∑k = 0∞ γk(Rt+k+1 + c)

= (∑k = 0∞ γkRt+k+1) + (∑k = 0∞γkc)

vc = ∑k = 0∞γkc

1. It will have effect, because if a negative reward value add a constant increased to a positive reward, the agent will incorrectly do learning.

Example:

If a negative reward route leading to a dead end in a maze changes to a positive value due to adding a constant, the agent will keep running into that dead end.

1. vπ(s) = ∑a **E**π[Rt | st = s, at = a] π(s,a)

vπ(s) = ∑a π(a|s) qπ(s,a)

1. Starting from A’ = 16

symbol

↓

vπ(s) = 16 / 0.94 = 24.387 to three decimal places

≈ 24.4