EBU4203 Introduction to AI – Week 2 Tutorial 2023

Q1: STATE the elements of Neural networks and EXPLAIN the functionalities of them.

Answers:

Neurons: the fundamental units of a neural network

Layers (input, hidden and output layers): Layers facilitate the structured arrangement and processing of data through the network.

Weights: parameters within the network that transform input data within each node.

Biases: additional parameters that allow for greater flexibility in the network's computations.

Activation functions: introduce non-linearities into the network.

Q2: STATE and DISCUSS the four types of activation functions commonly used in the neural networks.

Answers:

Linear functions: that the output signal is proportional to the input signal to the neuron, used in regression problems.

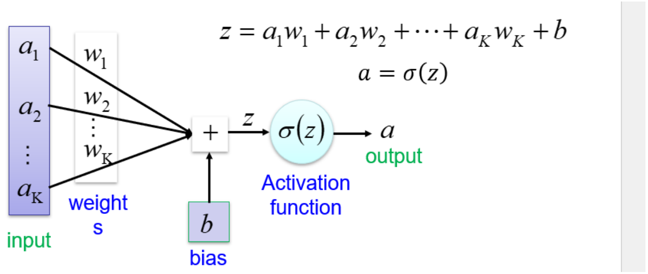
Sigmoid Function: takes a real-valued number and “squashes” it into the range between 0 and 1

Tanh function: takes a real-valued number and “squashes” it into range between -1 and 1

ReLU (Rectified Linear Unit): takes a real-valued number and thresholds it at zero

Q3: Given the input vector [a1,…,ak], weights [w1,…,wk] and bias b, assuming the activation function as σ, please explain what a single neuron in the neural networks does with formulas and diagrams.

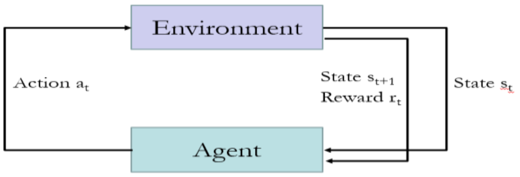
Answers:



Q4: How does reinforcement learning work? Please IDENTIFY the key elements and EXPLAIN the learning process.

Answers:

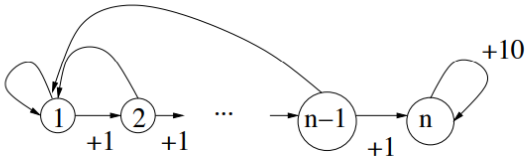
An agent (an AI) will learn from the environment by interacting with it (through trial and error) and receiving rewards (negative or positive) as feedback for performing actions. Actions, environment, agent, states, reward.



Q5: Consider the n-state Markov decision process MDP in the figure below. In state n there is just one action that collects a reward of +10, and terminates the episode. In all the other states there are two actions: float, which moves deterministically one step to the right, and reset, which deterministically goes back to state 1. There is a reward of +1 for a float and 0 for reset. The discount factor is γ =1/2.

i) Compute the optimal value function, V*(k) for all k=1,….,n-1.

ii) Assuming V*(n)=20, V*(1)=1, instead of taking deterministic strategy, the agent now has 0.8 probability to move to right, and 0.2 probability to reset, calculate the V*(n-1).



Answers:

i)

$$V^*(n-1) = 1+\frac{1}{2}V^*(n)$$

$$V^*(n-2) = 1+\frac{1}{2}V^*(n-1) = 1*\left(1+\frac{1}{2}\right)+\left(\frac{1}{2}\right)^2 V^*(n) = \frac{1-(1/2)^2}{1-1/2}+\left(\frac{1}{2}\right)^2 V^*(n)$$

$$\dots$$

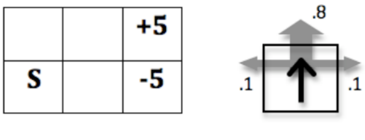
$$V^*(n-k) = 1+\frac{1}{2}V^*(n-k+1) = 1+\frac{1}{2}+\dots\left(\frac{1}{2}\right)^{k-1}+\left(\frac{1}{2}\right)^k V^*(n) = \frac{1-(1/2)^k}{1-1/2}+\left(\frac{1}{2}\right)^k V^*(n)$$

ii) Since V*(n)=20, to calculate V*(n-1), based on bellman equation,

V*(n-1)=0.8 x (1+$\frac{1}{2}$x20) + 0.2 x (0 + $\frac{1}{2}$ x 1) = 8.8 + 0.1 =8.9

Q6: This Gridworld problem is shown in Fig.1. The states are grid squares, dentified by their row and column number (row first). The agent always starts in state (1,1), marked with the letter S. There are two terminal goal states, (2,3) with reward +5 and (1,3) with reward -5.

Rewards are 0 in non-terminal states. (The reward for a state is received as the agent moves into the state.) The transition function is such that the intended agent movement (North, South, West, or East) happens with probability 0.8. With probability 0.1 each, the agent ends up in one of the states perpendicular to the intended direction. If a collision with a wall happens, the agent stays in the same state. Please answer the following questions.

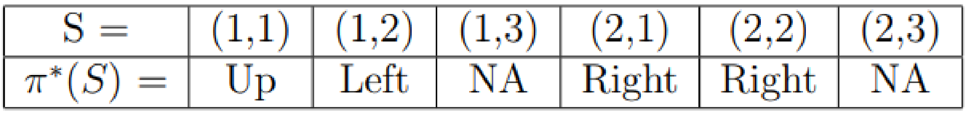


(a) (b)

Figure 1: (a) Gridworld MDP. (b) Transition function.

(i). Draw the optimal policy for this grid.

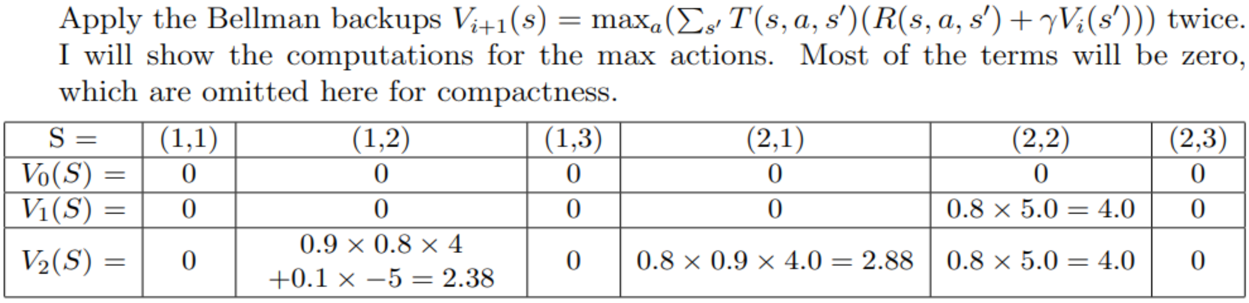
Answers:



S =	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
$\pi^*(S)$ =	Up	Left	NA	Right	Right	NA

(ii). Suppose the agent knows the transition probabilities. Give the first two rounds of value iteration updates for each state, with a discount factor γ= 0.9. (Assume V0 is 0 everywhere and compute Vi for times i = 1, 2).

Answers:



Apply the Bellman backups $V_{i+1}(s) = \max_a(\sum_{s'} T(s,a,s')(R(s,a,s') + \gamma V_i(s')))$ twice. I will show the computations for the max actions. Most of the terms will be zero, which are omitted here for compactness.

S =	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
$V_0(S) =$	0	0	0	0	0	0
$V_1(S) =$	0	0	0	0	$0.8 \times 5.0 = 4.0$	0
$V_2(S) =$	0	$0.9 \times 0.8 \times 4$ $+0.1 \times -5 = 2.38$	0	$0.8 \times 0.9 \times 4.0 = 2.88$	$0.8 \times 5.0 = 4.0$	0

(iii). Suppose the agent does not know the transition probabilities. What does it need to be able do (or have available) to learn the optimal policy?

Answers:

The agent must be able to explore the world by taking actions and observing the effects.

(iv). When using Q-learning to solve this GridWolrd problem, how do you formulate it as a Markov decision process (MDP)?

Answers:

States: $S_{1,1} \sim S_{2,3}$

Actions: North, South, West, or East

Transition probabilities: 0.8 with intended movement, and with probability 0.1 each, the agent ends up in one of the states perpendicular to the intended direction.

Reward: R=+5 when reaches the state $S_{2,3}$ and R=-5 for state $S_{1,3}$, all the other states, R=0

Discount factor: $\gamma = 0.9$

(v) Based on the formulated MDP above, please create the Q-table. When assuming the agent moves two steps towards right, calculate the Q-value and update the Q-table. (learning rate $\alpha = 0.1, \gamma = 0.9$)

Answers:

Q_values	Up	Down	Left	Right
$S_{1,1}$	0	0	0	0
$S_{1,2}$	0	0	0	0
$S_{1,3}$	0	0	0	0
$S_{2,1}$	0	0	0	0
$S_{2,2}$	0	0	0	0
$S_{2,3}$	0	0	0	0

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$$

First step: move right, Q ($S_{1,1}$, right) = 0+ 0.1 (0+0.9x0-0)=0, the Q-table keeps the same.

Second step: move right, Q ($S_{1,2}$, right) = 0 + 0.1 (-5+0.9 x 0 -0)= -0.05, the updated Q-table is

Q_values	Up	Down	Left	Right
$S_{1,1}$	0	0	0	0
$S_{1,2}$	0	0	0	-0.05
$S_{1,3}$	0	0	0	0
$S_{2,1}$	0	0	0	0
$S_{2,2}$	0	0	0	0
$S_{2,3}$	0	0	0	0