INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

CSE643 – Artificial Intelligence

Monsoon 2020 session

Quiz-3

Max marks: 10 (will be scaled down to 5 marks) **19-Nov-2020**

Submission deadline: 19-Nov-20 at 19:00 hrs

INSTRUCTIONS:

You will have to create a PDF file with your answers, name the file as AI-Q3-<Name>-<RollNo> and upload it on the classroom page by 7:00 pm.

In the answer sheet write your name and roll number.

In case you choose to have hand-written answers then those pages can be scanned and uploaded (make sure that it is clearly readable).

Q1: (2 marks)

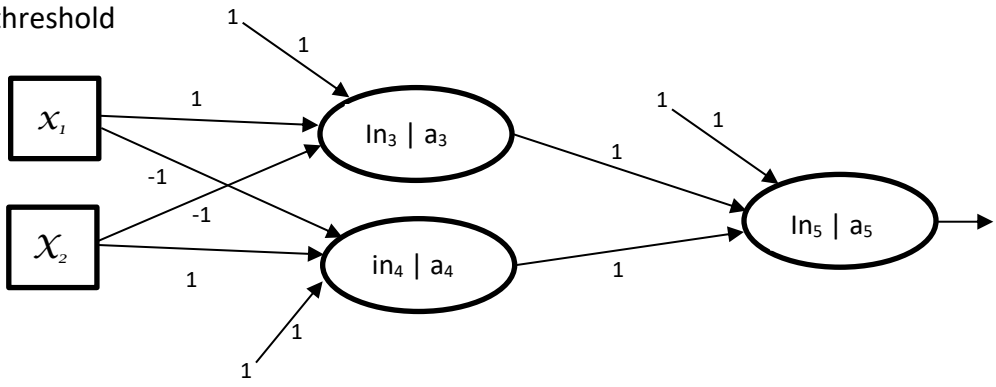
Create / Draw a three-unit neural network that has binary inputs x_1 and x_2 and a bias of 1 and output y, where y should implement the XOR function: $y = (x_1 \text{ XOR } x_2)$.

a) Assume that each unit uses a hard threshold activation function. Show all the weights.

b) Assume that each unit uses a summation activation function. Show all the weights.

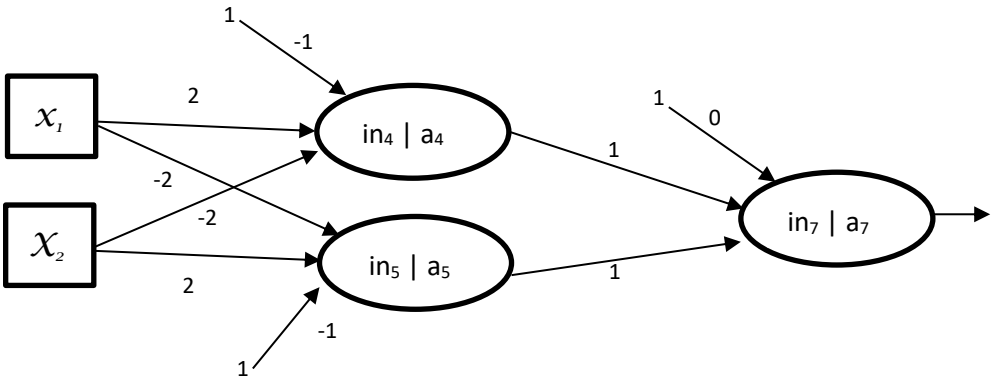
Answers

a) Hard threshold



$$y = h(x) = \begin{cases} 1 & \text{if } w^T x \geq 2 \\ 0 & \text{if } w^T x < 2 \end{cases}$$

b) Summation activation function



$$y = h(x) = \begin{cases} w^T x & \text{if } w^T x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Q2: (5 marks)

a) Describe in detail the Back-Propagation Learning Algorithm for Multi-Layer Perceptron (MLP) networks, and how it is related to gradient descent learning.

b) Suppose you are using an MLP for classification, and have two applications: one with two classes, and one with three classes. For each application, describe and justify particular choices for the error function and output activation function.

Answers

a) When training a neural network, the network learns and adjusts the weights that it gives to the input features through backpropagating the error in the output layer to the previous layers. To enable this process, we define a loss function that captures the L2 loss of the true output and the predicted output. The method calculates the gradient of the error function with respect to the neural network's weights and simulates the error propagation from succeeding layer to its preceding layer thereby optimizing weights. The algorithm works by computing the gradient of the loss function with respect to each weight by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer. Training a model is just minimizing the loss function, and to minimize we want to move in the negative direction of the derivative. Each step changes the weights to minimize the error.

Since we are trying to learn/ adjust weights so that we can minimize the loss, we take gradients of the loss function. We know from maths that any function is minimal when its derivative is equal to zero. Thus, we calculate the gradient of the loss function w.r.t. weights and then update each weight in the output layer and preceding layers. So we update the weight at the k^{th} layer $w_{j,k} \leftarrow w_{j,k} + \alpha \times a_j \times \Delta_k$ where α is the learning rate, a_j is the activation received from the j^{th} neuron and Δ_k is the gradient of the loss, i.e. $\Delta_k = Err_k \times g'(in_k)$, where g is the activation over the linear combination of the inputs multiplied by the weights. Similarly, we adjust the weights for previous layers through backpropagation since some part of the error that occurs at the k^{th} layer is due to the error in the weights at the previous j^{th} layer, where $w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta_j$ and Δ_j is the gradient of the loss, i.e. $\Delta_j = \Sigma w_{j,k} \Delta_k \times g'(in_j)$. This is done for all neurons in all layers upto the input layer.

b) i) Two class classification

Error function: Cross entropy loss is used. It is defined as:

$$\text{Error} = -\frac{1}{N} \sum_{i=1}^{N} (y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i))$$

where N is the number of input samples and the probability of prediction $p(y_i)$ is calculated using sigmoid function

Activation function: Sigmoid activation function $g(\mathbf{w.x}) = \frac{1}{(1+e^{-(\mathbf{w.x})})}$

Justification: Cross entropy loss uses a combination of class prediction with the probability of that class prediction. Thus, when the probability of prediction of the correct class is high then the loss is lower and when the probability of the prediction of the correct class is low then the loss is higher. This aids in the training phase to adjust the weights appropriately.

Using sigmoid activation function gives us the probability of the prediction for a particular target class in the range between [0,1]. Thus, it can be used for 2-class classification.

ii) Three class classification

Error function: We use Categorical cross entropy loss defined as:

$$\text{Loss} = -\sum_{i=1}^{k} y_i \log(p(y_i))$$

Where *k* is the number of classes (3 in this case) and $p(y_i)$ is the probability of prediction for that class

Activation function: Softmax activation function $g((\mathbf{w.x})_i) = \frac{e^{(\mathbf{w.x})_i}}{\sum_j^k e^{(\mathbf{w.x})_j}}$

Justification: Using the categorical cross entropy loss allows the adjusting of the weights for the 3 classes. And using softmax activation function gives us the predictions for the 3 classes.

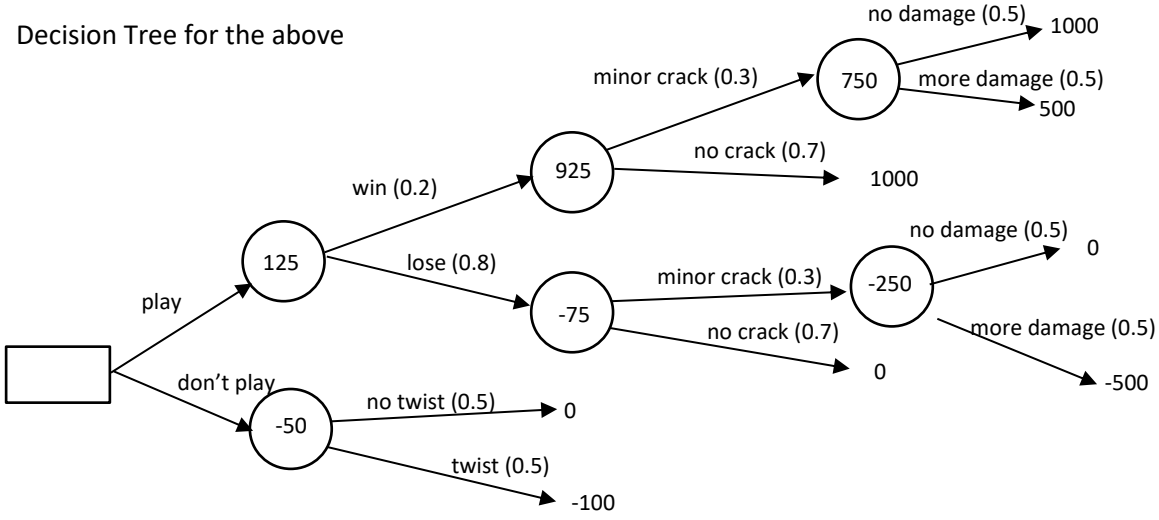
Q3: (3 marks)

You are a hockey player and represent your college team. During a practice match you sprained your ankle. Your doctor suspects that there could be a minor crack in your ankle and predicts it with a probability of 0.3 (i.e. the doctor is 30% sure that there is a crack) and advises you complete rest. However, there is an important game tomorrow and you feel that you should play for the team to win (with a win probability of 0.2). If actually you have a minor crack in the ankle and you play then you will risk further damage that can ground you for a long time. Hence, the utility values are as follows: If your team wins and you have not damaged your ankle further then you get Rs.1000/-. If your team wins but your ankle is more painful and damaged you gain a total of Rs.500/- (Rs 1000/- minus Rs 500/- you spend on the medical advice). If your team loses and your ankle is not damaged further then you gain Rs 0/-. But if your team loses and your ankle is damaged further then lose Rs 500/- for medical advice. If you do not play and your ankle is as before then you gain Rs 0/-. But if you don't play but twist your ankle at home then you lose Rs 100/- for pain-management ointment.

a. Draw the decision tree for the above situation.
b. Evaluate the tree and determine the best choice and its expected utility.

Answers

a. Decision Tree for the above



b. Utility of play = 125 and Utility of don't play = -50. So the best choice is that you play the game the next day.