**Module 5**

1.What activation function should I use in the case of ?

1. Classifiers
2. Vanishing gradient problem
3. Generalised
4. Only in hidden layers
5. Only in Output layer

For classifiers: Use sigmoid or softmax.

Vanishing gradient problem: ReLU or its variants.

Generalized: ReLU.

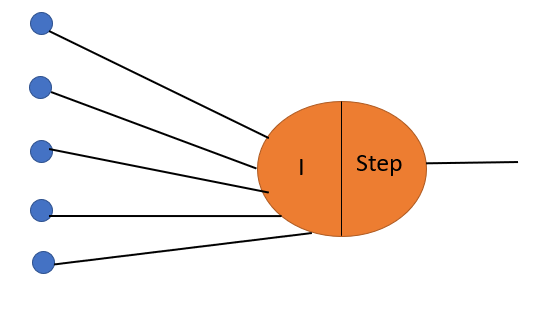
Only in hidden layers: ReLU/LeakyReLU/Maxout.

Only in Output layer: Linear/Identity for regression, sigmoid/tanh for binary classification, softmax for multi-class classification.

2. Calculate the error of a following perceptron

Whose values are given as below

|  |  |  |  |
| --- | --- | --- | --- |
| Sno | Inputs(x) | Weights | Output |
| 1 | 0.5 | 0.6 | 1.62 |
| 2 | 0.006 | 0.001 |  |
| 3 | 0.1 | 0.005 |  |
| 4 | 0.4 | 0.67 |  |
| 5 | 0.8 | 0.45 |  |



Let's say our activation function is a step function, where if the sum of inputs times weights is greater than a certain threshold, the output is 1, otherwise, it's 0.

Given:

Inputs (x)

Weights

Outputs (before applying activation function)

We can calculate the error using a simple formula. For each row: Error = (Output - Predicted\_Output)^2

Where Predicted\_Output is the output after applying the activation function.

Let's calculate the error for each row:

For the first row:

Input: 0.5

Weight: 0.6

Output (before activation): 1.62

Predicted Output (after activation): Assuming a threshold of 1, it would be 1

Error = (1.62 - 1)^2 = 0.3844

For the second row:

Input: 0.006

Weight: 0.001

Output (before activation): Not given

Predicted Output (after activation): Assuming a threshold of 1, it would be 0

Error = (0 - 0)^2 = 0

Similarly errors for third, fourth and fifth row are calculated, to get the total error, we sum up all these individual errors:

Total Error = 0.3844 + 0 + 0 + 1 + 1 = 2.3844

So, the total error of the perceptron is approximately 2.3844.

3. Write down the steps involved in Back Propagation and the disadvantages of using Back Propagation.

Forward Pass: Input through network, predict output.

Calculate Loss: Find the difference between predicted and actual output.

Backward Pass: Calculate gradients of loss w.r.t. weights and biases.

Update Weights: Adjust weights to minimize loss.

Disadvantages:

Computationally Intensive

Local Minima

Sensitivity to Initialization

Vanishing and Exploding Gradients

Overfitting