PERCEPTRON LOSS FUNCTION

Perceptron loss function is used to measure the error or mismatch between the predicted output of the perceptron and the actual output.

The most commonly used loss function in perceptron is mean squared error function.

During each iteration, the perceptron computes the predicted output using the current weights & updates the weights based on the difference between the predicted output and the target output.

$$\underset{(W_1, W_2 \dots W_n, b)}{L} = \frac{1}{n} \sum_{i=1}^{n} L(y^i, f(x_i) + \alpha R(W_1, W_2, \dots W_n)$$

$$L(y_i, f(x_i) = \max(0, -y_i f(x_i))$$

[Hinge Loss] where $f(x_i) = W_1 x_1 + W_2 X_2 \dots\dots W_n X_n + b$

$$\boxed{L = \frac{1}{n} \sum_{i=1}^{n} \max(0, -y_i f(x_i))}$$

EXPLAINATION OF LOSS FUNCTION :-

$$L = \frac{1}{n} \sum_{i=1}^{n} \max(0, -y_i f(x_i))$$

where, $f(x_i) = W_1 X_1 + W_2 X_2 + \dots W_n X_n + b$

$\max(0, -y_i f(x_i))$ means that

$\boxed{\begin{array}{l} \text{if, } -y_i f(x_i) \geq 0 \\ \quad \text{then value will be } -y_i f(x_i)) \\ \\ \text{if, } -y_i f(x_i)) < 0 \\ \quad \text{then value will be } 0 \end{array}}$

$$L = \frac{1}{n}\left[\max(0, -y_1 f(x_1) + \max(0, -y_2 f(x_2) + \ldots \ldots \max(0, -y_n f(X_n)\right]$$

$\boxed{\begin{array}{l} \text{if, } y_i f(x_i) \text{ is } -\text{ve} \\ \quad \text{then } -y_i f(x_i) \text{ becomes } +\text{ve and the} \\ \quad \text{value of loss function will be selected.} \\ \text{if, } y_i f(x_i) \text{ is } +\text{ve} \\ \quad \text{then } -y_i f(x_i) \text{ becomes } -\text{ve and the value} \\ \quad \text{of loss function will be } 0 \end{array}}$

Now, we will see how to calculate the value of weights and bias.

The main objective is to select the values for weights that will minimize the loss function.

Using gradient descent ÷

for i in epochs :

$$w_1 = w_1 + \eta \frac{dL}{dw_1}$$

$$\vdots$$

$$w_n$$

$$b = b + \eta \frac{\partial L}{\partial b}$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial f(x_i)} \times \frac{\partial f(x_i)}{\partial W_1}$$

$$\frac{\partial L}{\partial f(x_i)} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i & \text{if } y_i f(x_i) < 0 \end{cases} \qquad \frac{\partial f(x_i)}{\partial W_1} = x_{i1}$$

$$\boxed{\begin{aligned} \frac{\partial L}{\partial W_1} &= \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i x_{i1} & \text{if } y_i f(x_i) < 0 \end{cases} \\ \frac{\partial L}{\partial W_n} &= \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i x_{in} & \text{if } y_i f(x_i) < 0 \end{cases} \\ \frac{\partial L}{\partial b} &= \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i & \text{if } y_i f(x_i) < 0 \end{cases} \end{aligned}}$$

We can use different loss functions according to our needs

Example → When the activation function is sigmoid function then the loss function that is used with it is binary cross entropy.

$$\text{Sigmoid} \Rightarrow \sigma(z) = \frac{1}{1+e^{-z}}$$

Binary cross entropy → $-y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i)$

Perceptron == Logistic Regression (for sigmoid func.)