**GOVERNING EQUATIONS**

**8.1) GRADIENT DESCENT**

Steps for Gradient Descent

1) Take random 𝚹

2) Update 𝚹 in a direction of decreasing gradient(slope)

3) Update gradients

𝚹 = 𝚹 - ղ\*ძf(𝚹) / ძ(𝚹)

where, ղ = learning rate; repeat step 2 until you reach the local minima

Backpropagation is the algorithm using which our model is based. The concept is similar to that of like when we teach a child when he makes mistakes. Our model also acts like a child that needs someone to teach it when it makes any mistakes .

Our model uses gradient descent and other functionality for its smooth working. By changing the weights it moves in the backward direction for re-training of networks and it continues to retrain until the model gives the most optimum results with least possible errors.David Rumelhart was the one to coin this algorithm in 1986 when he published a famous note on this algorithm, although it was introduced back in 1970s.

**8.2) SOFTMAX FUNCTION**

The softmax function is used to turns a vector of K real values into a vector of K real values that sum to 1. Values of input can be positive, negative, zero, or greater than one, but this function modify them into values between 0 and 1, so that we can use to interpret them as probabilities. If the inputs is small or negative, the softmax returns it into a small probability, and if an input is large, then it returns it into a large probability, but its value will always remain between 0 and 1.The softmax function is called as the softamax function, or multi-class logistic regression. This is because the softmax is a generalization of logistic regression that is used for multi-class classification, and its formula is very alike to the sigmoid function which is used for logistic regression. The softmax function is used in a classifier only when the classes are mutually exclusive.

13

Most of the multi-layer neural networks ends in a penultimate layer and outputs are real-valued scores that are not conveniently scaled and which is difficult to work with. This function is very useful because it converts it into the scores to a normalized probability distribution, which can be used to display to a user or can be used as input to other systems. For this reason it is usual to append a softmax function as the final layer of the neural networks.

Softmax Formula

The softmax formula is as follows:

