ANN Project (Assignment 1)

Q1.

- Forward Propagation: - In this process training data is passed through (in the forward direction) through the hidden layers and the respective activation functions with the ai of training the neural network.
- Backward Propagation: - Given an ANN and an error function Backpropagation is the calculation of gradient proceeds backwards through the network update the model weights based on feedback of the previous runs.
-

Q2. Let W[4][5] be the Weight matrix and B[1][5] be the Bias. X[1][4] are the inputs and A[1][5] are the Hidden Layers

Forward Propagation

Then A[1][5] = X_i[1][j]*W[j][5] + B[1][5] which is matrix multiplication via looping where I (inputs) varies from 0 to 3 , j (weights) varies from 0 to 4.

Output = ∑ Activation(A[][]) (Final Step)

Backward Propagation

Calculating derivative by chain rule by using error/loss/cost function at the output stage and moving backwards through the layer using appropriate weights.

Q3.

- Sigmoid: - f(x) = 1/(1+exp(-x)), f'(x) = exp(-x)/(1+exp(-x))^2
- Relu: -f(x) = max (0, x), f'(x) = {0, if x < 0 (undefined at x = 1,)
  1, if x > 1}
- Leaky Relu: - f(x) = 0.01x for x < 0 and x for x >= 0. f'(x) = 0.01 for x < 0 and 1 for x > 0
- Tanh: - f(x) = $e^x – e^{-x}$ / $e^x + e^{-x}$ and f'(x) = 4 / $(e^x + e^{-x})^2$
- Softmax: - f(x)(i) = e^{x_i} /$\sum_k e^{x_k}$ for all k = 1 to N and f'(x) = f(x)(i)*[δ_{ij} – f(x)(j)]

Where δ represents the Kronecker delta function