







# (Task 1) Implementation of Neural Network with Sigmoid and cross entropy loss function.

| Initially random dataset of 4000 data points having two classes 2000 data points each as shown in figure. |
| --- |
|  |

Data was shuffled and split into 3 parts. (Training 60%, Testing 20 % , Validation 20%)

| Simple neural network architecture was built with 2 input nodes and 1 output nodes. with weight vector of 2X1 |
| --- |
|  |

Following sigmoid functions are used in simple neural network.

| **Sigmoid** | **Sigmoid** **Derivative:** |
| --- | --- |
| S(x)=1/1+e^-x | S(x)=S(x)(1-S(x)) |

For cross entropy function

| **Cross** **Entropy** | |
| --- | --- |
|  | where 𝑦n is actual while𝑦̂n is predicted value.  Using learning rate and loss value backpropagation loss are calculated |



Parameters for training: (Number’s of Epochs=100,Learning rate: 0.001)

| Following graph shows Loss curve generated by loss function while training. | |
| --- | --- |
|  | Model save into pickle file and Accuracies are 100 percent In Training, Testing and Validation  Training Accuracy: 100.0 %  Testing Accuracy: 100.0 %  Validation Accuracy: 100.0 % |

# (Task 2) Implementation of Neural Network with a single hidden layer, multiple activation function and cross entropy loss function.

| Dataset of 1000 random points with two classes Red and Green | |
| --- | --- |
|  | Red label indicated by ‘1’ and Green label indicated with ‘0’ |

Data was shuffled and split into 3 parts. (Training 60%, Testing 20 % , Validation 20%). For training on data, a simple neural network was built with 1 hidden layer with 3 neurons. And weight vector of layer 1 are (input+1x3) and layer 2 (hidden layer+1x2), with +1 bais

Following activation function are used

| **Sigmoid** | **Tanh** | **ReLu** |
| --- | --- | --- |
| S(x)=1/1+e^-x | T(x)=e^x-e^-x/e^x+e^-x | F(𝑥) = max (0, 𝑥) |

For cross entropy function

| **Cross** **Entropy** | |
| --- | --- |
|  | where 𝑦n is actual while𝑦̂n is predicted value.  Using learning rate and loss value backpropagation loss are calculated |

Parameters for training: (Number’s of Epochs=1000,Learning rate: 0.009,Activation:Sigmoid)

| Following graph shows Loss curve generated by loss function while training. | |
| --- | --- |
| Loss Curve at 1000 epochs | Accuracy Graph at 1000 epochs |
| Model save into pickle file and Accuracies are 100 percent In Training, Testing and Validation  Training Accuracy: 100.0 %  Testing Accuracy: 100.0 %  Validation Accuracy: 100.0 % | |

# (Task 3) Implement a multi-layer Neural Network for multi-class classification.

MNIST Dataset 60000 samples for training and 10000 samples for testing with size of (28 X 28) . We load dataset with flatten and reshape of [60000, 28,28] to [60000, 784] to multi-class classification neural network.

| Mean of all images | Subtracted image from mean |
| --- | --- |
|  |  |

