REPORT

Problem Statement :To create a 2 layer Neural Network for a given Dataset

Given :Program that implements a Single Layer Neural Network

DATASET DESCRIPTION (The Glass dataset) :

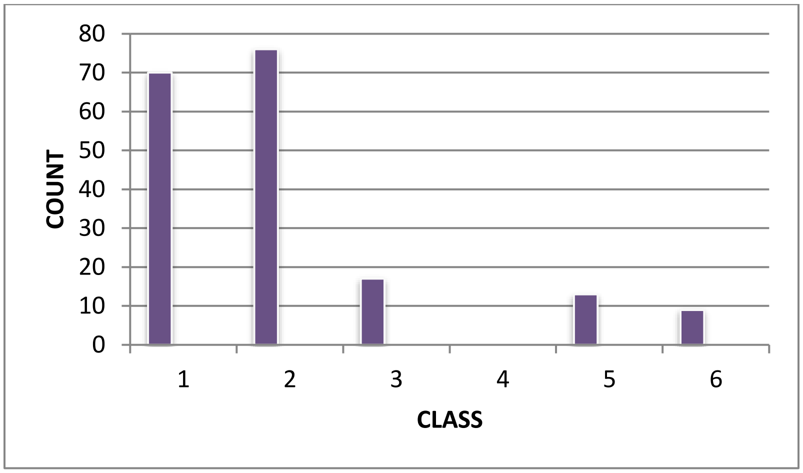
The data aims at classifying the type of glass based on its composition. The dataset comprises of 214 instances each having 11 features.

The dataset has the following attributes:

1. Id number: 1 to 214
2. RI: refractive index
3. Na: Sodium
4. Mg: Magnesium
5. Al: Aluminum
6. Si: Silicon
7. K: Potassium
8. Ca: Calcium
9. Ba: Barium
10. Fe: Iron
11. Type of glass: (class attribute)
    -- 1 building_windows_float_processed
    -- 2 building_windows_non_float_processed
    -- 3 vehicle_windows_float_processed
    -- 4 vehicle_windows_non_float_processed (none in this database)
    -- 5 containers
    -- 6 tableware
    -- 7 headlamps

We use attributes 2-10 for creating a model to predict values for 'Type'. The class distribution of the instances is as follows:

Class	1	2	3	4	5	6	7
Count	70	76	17	0	13	9	29



CLASS DISTRIBUTION

The Glass dataset is a multiclass dataset (7 classes). However, we were supposed to work with a binary class problem. Thus, we reorganized the dataset into two classes – 0 and 1, such that :

Class 2, 3, 4, 5 == Class 0 and Class 1, 6, 7 == Class 1

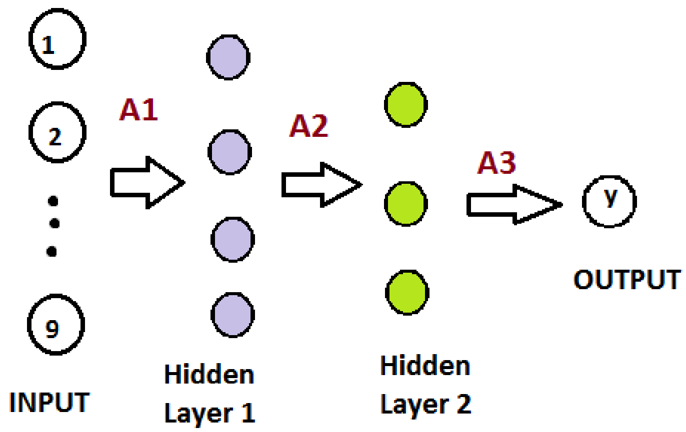
Class 0 now has 76 + 17 + 0 + 13 = 106 instances.

Class 1 now has 70 + 9 + 29 = 108 instances.

The class distribution is now balanced with a ratio of 106:108 = 0.98.

THE TWO LAYER NEURAL NETWORK

Initial model of neural network is :



Input Layer: It has 9 units each corresponding to each of the 9 features we chose.

Hidden Layers: As of now, we consider 4 neurons/units in first hidden layer and 3 in the second one.

Output Layer: There is a single output unit as there is just one feature that we are predicting.

At each layer, we calculate the activation value for each instance. A1, A2, A3 correspond to the activation function for each layer.

DECIDING THE ACTIVATION FUNCTIONS

We need to choose a combination of activation function for our network such that it gives the maximum accuracy. After various combinations and runs, the following results were obtained.

A1	A2	A3	H1	H2	Accuracy
Tanh	sigmoid	Sigmoid	4	3	97%
Tanh	**Tanh**	**sigmoid**	**4**	**3**	**100%**
Sigmoid	Tanh	Sigmoid	4	3	87%
Tanh	Tanh	tanh	4	3	--
sigmoid	sigmoid	Sigmoid	4	3	87%

H1 --- Number of hidden units in layer 1

H2 --- Number of hidden units in layer 2

We observe that **applying tanh function in layer 1 and layer 2**, and applying **sigmoid in layer 3** gives the maximum possible accuracy. Thus, we proceed with this combination of activation functions.

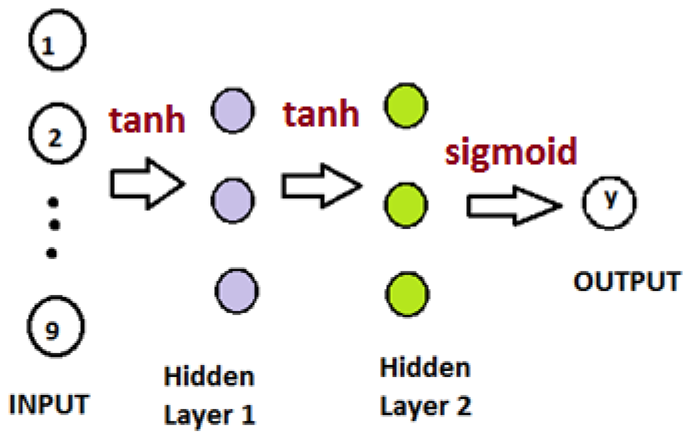
THE NUMBER OF NEURONS IN EACH HIDDEN LAYER

Now that we have fixed the activation functions to be used, we attempt to obtain the minimal possible neurons in each hidden layer. Certain combinations were tried to obtain the following results :

H1	H2	Accuracy
4	3	100%
3	**3**	**100%**
3	2	98%
3	1	98%
2	3	99%

We observe that even if we **reduce the number of neurons to 3 in the first layer and let it be 3 in the second layer**, we achieve the best possible accuracy, thus, we select this.

OUR FINAL NEURAL NETWORK LOOKS LIKE :



FLOW OF PROGRAM

1. IMPORTING THE DATASET :The pandas library was used to read the file glass.csv which had our dataset instances.
2. CONVERTING THE MULTICLASS PROBLEM TO A BINARY CLASSIFICATION PROBLEM:We handles the class imbalance problem by assigning the instances in such a way that the instances came out to be in the ratio 1:1 approximately.
3. CREATING A NEURAL NETWORK MODEL: A model with 9 input neurons,3 neurons in both the hidden layers and 1 output neuron.
4. INITIALISING THE PARAMETERS:The weight and Bias parameters were initialized for each layer.

   Shape of W1=(3,9)
   Shape of b1=(3,1)

   Shape of W2=(3,3)
   Shape of b2=(3,1)

Shape of W3=(1,3)
Shape of b3=(1,1)

5. FORWARD PROPAGATION :During forward propagation, the network is provided with input features in the input layer. The input is passed to the next hidden layers using the weights, biases and activation functions. Following this architecture, an output is obtained and compared to the original values.
6. COST COMPUTATION :Computed the cost using Cross Entropy Function.
7. BACKPROPAGATION :Computed the gradients by taking derivative of error with respect to weights to achieve a better result.
8. UPDATING PARAMETERS:Updated the parameters using Gradient Descent Update rule, learning rate and the values of gradients obtained by backpropagation.
9. PREDICTION:Performing Forward Propagation again with the finally updated parameters to predict the value of the target attribute.
10. ACCURACY : Accuracy is calculated as the ratio of the correctly predicted instances to the total number of instances.

COMPARISON WITH THE INBUILT LOGISTIC REGRESSION

Using the built in Logistic Regression Class and its methods on the same dataset with the same classification, the accuracy obtained was 95.79% whereas for our model the accuracy obtained was 100%.

SUBMITTED BY:
EKTA WAHI – 12
VAISHALI CHAWLA – 45