Neural Network ReadMe

Saketh Annimalla-sxa230044
Revanth Chandra Gampa-RXG230014
Omar Mukhtar Shaik-oxs230000

No of free days used : 1

2 Programming Part (60 points)

Data set is obtained from UCI repo: https://archive.ics.uci.edu/dataset/850/raisin
Data set:
https://raw.githubusercontent.com/Saketh-09/ArtificialNeuralNetwork/master/Raisin_Dataset.csv

Images of the Kecimen and Besni raisin varieties were obtained with CVS. A total of 900 raisins were used, including 450 from both varieties, and 7 morphological features were extracted.

List of features

Area	Integer	Gives the number of pixels within the boundaries of the raisin.
MajorAxisLength	Continuous	It measures the environment by calculating the distance between the boundaries of the raisin and the pixels around it.
MinorAxisLength	Continuous	Gives the length of the main axis, which is the longest line that can be drawn on the raisin.
Eccentricity	Continuous	Gives the length of the small axis, which is the shortest line that can be drawn on the raisin.
ConvexArea	Integer	It gives a measure of the eccentricity of the ellipse, which has the same moments as raisins.
Extent	Continuous	Gives the number of pixels of the smallest convex shell of the region formed by the raisin.
Perimeter	Continuous	Gives the ratio of the region formed by the raisin to the total pixels in the bounding box.

Target

Target represents a categorical value indicating whether the data corresponds to **Kecimen** or **Besni** raisin.

Splitting pre-processed dataset into 80%(train) & 20%(test)

Methods in the neural network

Preprocess : performs data pre-processing including encoding categorical values to numericals
Split_the_data: splits the data into test & train
Methods for different activation functions: sigmoid, tanh, relu, softmax
Forward: calculates the outputs from forward pass
Train: iterates over epochs and updates the weights & biases based on activation function

Neural Network ReadMe

Saketh Annimalla-sxa230044
Revanth Chandra Gampa-RXG230014
Omar Mukhtar Shaik-oxs230000

activation_function	hidden_layer_size	learning_rate	epochs	batch_size	train_data_accuracy	test_data_accuracy
Sigmoid	2	0.005	100	30	86.37	88.0
Sigmoid	2	0.005	500	30	87.556	87.556
Sigmoid	2	0.005	1000	30	86.963	88.889
Sigmoid	2	0.005	5000	30	87.111	90.222
Sigmoid	2	0.01	100	30	85.778	88.0
Sigmoid	2	0.01	500	30	87.259	88.0
Sigmoid	2	0.01	1000	30	87.407	88.889
Sigmoid	2	0.01	5000	30	87.704	88.444
Sigmoid	3	0.005	100	30	49.778	50.667
Sigmoid	3	0.005	500	30	49.778	50.667
Sigmoid	3	0.005	1000	30	86.222	86.222
Sigmoid	3	0.005	5000	30	87.259	88.889
Sigmoid	3	0.01	100	30	49.778	50.667
Sigmoid	3	0.01	500	30	87.111	88.0
Sigmoid	3	0.01	1000	30	86.815	89.333
Sigmoid	3	0.01	5000	30	88.148	88.444
Sigmoid	4	0.005	100	30	84.296	84.889
Sigmoid	4	0.005	500	30	86.222	86.667
Sigmoid	4	0.005	1000	30	86.963	87.111
Sigmoid	4	0.005	5000	30	87.556	87.556
Sigmoid	4	0.01	100	30	85.778	86.222
Sigmoid	4	0.01	500	30	87.407	88.0
Sigmoid	4	0.01	1000	30	86.667	90.222
Sigmoid	4	0.01	5000	30	88.741	88.444
Sigmoid	5	0.005	100	30	85.333	88.0
Sigmoid	5	0.005	500	30	87.111	88.0
Sigmoid	5	0.005	1000	30	87.704	88.0
Sigmoid	5	0.005	5000	30	87.407	88.444
Sigmoid	5	0.01	100	30	86.667	87.556
Sigmoid	5	0.01	500	30	87.407	87.556
Sigmoid	5	0.01	1000	30	86.815	87.111
Sigmoid	5	0.01	5000	30	88.444	87.556
Tanh	2	0.005	100	30	86.222	87.111
Tanh	2	0.005	500	30	87.259	88.0
Tanh	2	0.005	1000	30	86.963	88.889

Tanh	2	0.005	5000	30	87.852	88.0
Tanh	2	0.01	100	30	85.63	87.556
Tanh	2	0.01	500	30	87.259	87.556
Tanh	2	0.01	1000	30	87.111	88.889
Tanh	2	0.01	5000	30	87.407	88.889
Tanh	3	0.005	100	30	87.556	87.556
Tanh	3	0.005	500	30	87.704	88.0
Tanh	3	0.005	1000	30	86.667	88.889
Tanh	3	0.005	5000	30	87.407	88.444
Tanh	3	0.01	100	30	85.481	84.0
Tanh	3	0.01	500	30	86.37	88.0
Tanh	3	0.01	1000	30	87.111	87.111
Tanh	3	0.01	5000	30	87.704	89.778
Tanh	4	0.005	100	30	86.222	86.667
Tanh	4	0.005	500	30	86.667	88.889
Tanh	4	0.005	1000	30	87.704	87.111
Tanh	4	0.005	5000	30	83.556	84.444
Tanh	4	0.01	100	30	87.407	87.556
Tanh	4	0.01	500	30	85.778	87.556
Tanh	4	0.01	1000	30	87.556	89.333
Tanh	4	0.01	5000	30	87.259	88.889
Tanh	5	0.005	100	30	87.556	87.111
Tanh	5	0.005	500	30	87.704	87.111
Tanh	5	0.005	1000	30	87.556	86.667
Tanh	5	0.005	5000	30	87.852	88.889
Tanh	5	0.01	100	30	87.259	88.0
Tanh	5	0.01	500	30	86.963	87.111
Tanh	5	0.01	1000	30	82.963	82.222
Tanh	5	0.01	5000	30	87.556	89.778
Relu	2	0.005	100	30	86.519	89.778
Relu	2	0.005	500	30	73.63	72.0
Relu	2	0.005	1000	30	49.926	50.667
Relu	2	0.005	5000	30	49.778	50.667
Relu	2	0.01	100	30	69.778	67.111
Relu	2	0.01	500	30	49.778	50.667
Relu	2	0.01	1000	30	49.778	50.667
Relu	2	0.01	5000	30	49.778	50.667
Relu	3	0.005	100	30	86.963	86.667
Relu	3	0.005	500	30	49.778	50.667

Neural Network ReadMe

Saketh Annimalla-sxa230044
Revanth Chandra Gampa-RXG230014
Omar Mukhtar Shaik-oxs230000

Relu	3	0.005	1000	30	49.778	50.667
Relu	3	0.005	5000	30	49.778	50.667
Relu	3	0.01	100	30	69.63	67.111
Relu	3	0.01	500	30	49.778	50.667
Relu	3	0.01	1000	30	49.778	50.667
Relu	3	0.01	5000	30	49.778	50.667
Relu	4	0.005	100	30	83.556	84.0
Relu	4	0.005	500	30	49.778	50.667
Relu	4	0.005	1000	30	49.778	50.667
Relu	4	0.005	5000	30	49.778	50.667
Relu	4	0.01	100	30	50.222	49.333
Relu	4	0.01	500	30	49.778	50.667
Relu	4	0.01	1000	30	49.778	50.667
Relu	4	0.01	5000	30	49.778	50.667
Relu	5	0.005	100	30	82.37	80.0
Relu	5	0.005	500	30	49.778	50.667
Relu	5	0.005	1000	30	49.778	50.667
Relu	5	0.005	5000	30	49.778	50.667
Relu	5	0.01	100	30	50.222	50.222
Relu	5	0.01	500	30	49.778	50.667
Relu	5	0.01	1000	30	49.778	50.667
Relu	5	0.01	5000	30	49.778	50.667

Observations

- For the considered dataset learning rate of 0.005 gave better accuracy than learning rate of 0.01
- Sigmoid activation function resulted in better accuracy when compared to tanh and relu activation functions
- When learning rate is 0.005 it is observed that accuracy is increased on increasing the epochs, whereas when learning rate is 0.01 an oscillation in accuracy is observed on increasing epochs.
- On increasing the number of units in the hidden layer we have observed an increase in training accuracy whereas there is a slight decrease in test accuracy. This might indicate the case of overfitting for our dataset.