INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI
भारतीय प्रौद्योगिकी संस्थान तिरुपति

Machine Learning for Image Processing

Programming Assignment-2

Objective: The objective of this programming assignment is to provide hands-on practice on the implementation of a two-class classification model using a Neural Network (NN) with one hidden layer. The code should be written from scratch in python without using any inbuilt packages for building “your” neural network! **Numpy** is sufficient for this implementation.

1) Perform a two-class classification of the data given in **data.csv** by building your own multi-layer neural network from scratch with one hidden layer containing multiple neurons, and an output layer with a single neuron.

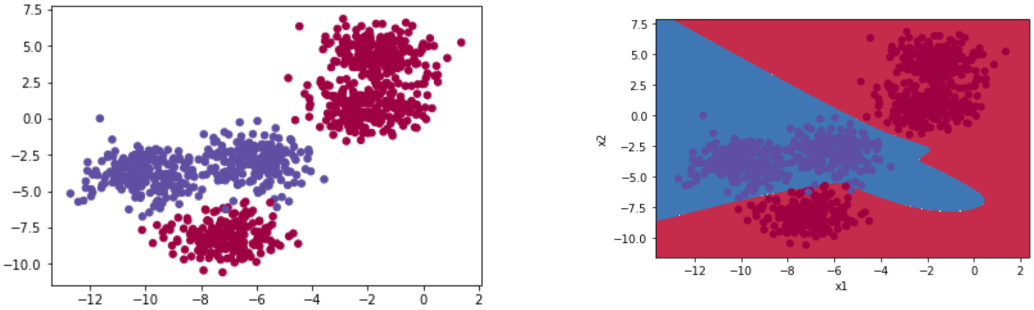
For this purpose, write your own function such that the number of neurons in the hidden layer can be varied by simply passing the value of the number of neurons as an argument to that function. Initialize the weight vectors randomly for each neuron by following the recommendations discussed in the 32nd lecture of the MLIP course! Use Sum of Squared Difference (SSD) as the loss function for each neuron, and sigmoid function as the activation function. Use gradient descent algorithm as the optimizer where all samples are used while updating the weights in each iteration.

- Experiment with your NN by starting with two neurons in the hidden layer, and keep increasing them until you get reasonable classification accuracies (> 95%). Report the final number of neurons you have used in the hidden layer, and display the decision lines you obtained for each of the neurons in the hidden layer by overlaying them on the training data. Also, calculate and specify the class-wise accuracies obtained for your method.
- Plot the decision boundary that divides the two classes (Hint: For all points in the plane, predict the class using the trained model and highlight with two colors. Check the reference figures)

2) Repeat problem-1, but now with cross-entropy as the loss function instead of SSD. You can use the same number of neurons to which you have already converged in problem-1. Calculate and specify the class-wise accuracies obtained for your method. Write your observations by comparing results obtained from the SSD loss function and cross-entropy loss function.

Sample Results

Scatter plot of the training data and the decision boundaries with 5 neurons in the hidden layer.


7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−12
−10
−8
−6
−4
−2
0
2
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
x2
−12
−10
−8
−6
−4
−2
0
2
x1