

Emotion Detection

Part 1: Logistic Regression Module:

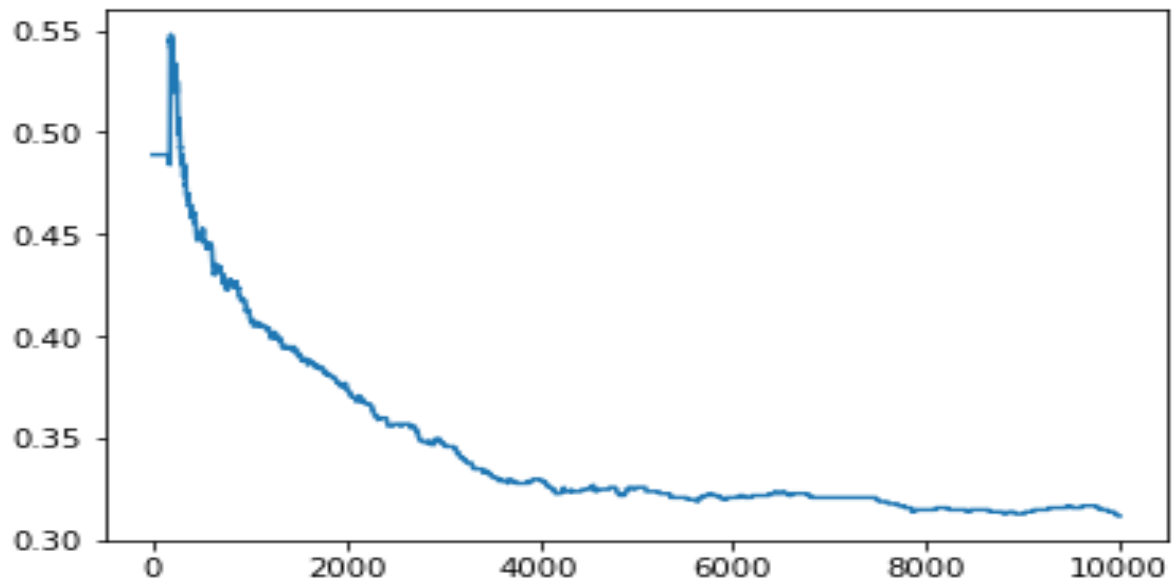
Description

In a way logistic regression is an extension of linear regression, the primary difference is the prediction values obtained from both where linear regression is based on a continuous function, logistic regression gives only specific predictions that is classification. It is a binary classifier, which takes one of the two values zero or one to determine whether a given value is part of class one or zero. Based on the features of the input value logistic regression classifier fits the data to a function and then pass the predicted value through a sigmoid function that normalizes the predicted values as zero or one which classifies them accordingly. Classification is the method to find the decision boundary between the given classes and this is achieved through logistic regression which is called a supervised learning algorithm. As the function may seem unstable, loss functions are used to learn based on previous inputs and regularizers are used to impose penalties on these loss functions thus achieving better success, gradient descent is used to converge on the minima to get the best features possible for prediction, these weights are then used for prediction of test input.

Logistic regression faces problems of underfitting and overfitting quite often and regularizers are used to prevent this specific problem, it is a very good method to handle high correlation between features, the idea behind regularization is to introduce additional bias to parameter weights. L2 regularization is the most common regularizer.

Requirements in Skeleton Code

1]Graph showing the change in error over time for the Validation set:



2] Best validation error obtained is 0.349

3] Accuracy obtained for the test set is %68.066

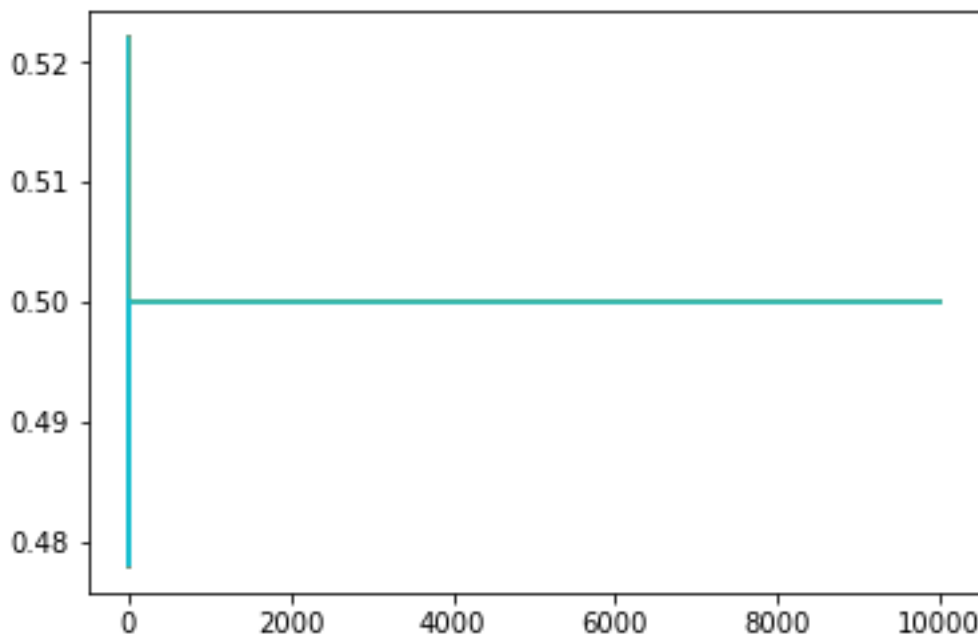
Part 2: Neural Network Module:

Description

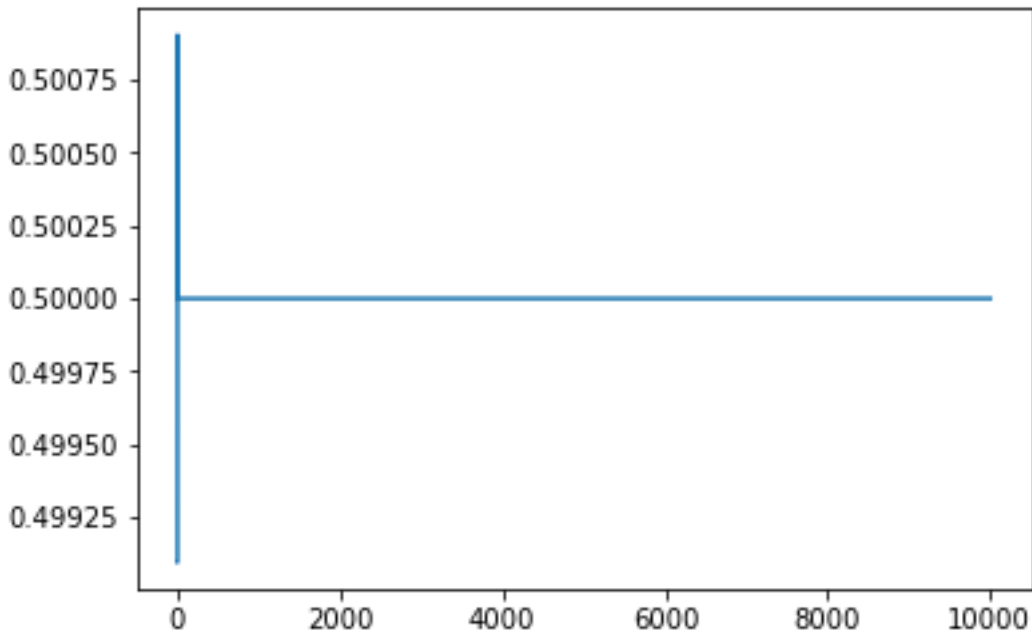
Neural Networks are used for classification using multilayer feed-forward networks and the primary algorithm used is back propagation which is a supervised learning algorithm. The networks are inspired by the neurons in the human brain which transfer their signals forwards. The idea is to modify the weights inside the network to get the desired output. There exists a hidden layer that changes the weight values based on the output received from layer one and these updated weights are used for further processing. The sum of the weighted values inside the neuron is called the activation values which is transferred forwards, now coming to back-propagation the values predicted from the forward network is compared to the expected output and error is calculated and is passed through the network again as the penalty and this is done until an acceptable output is achieved. While training the neuron we end up with the values for the weights for each layer which is then used to predict the test set by passing them through the network and use the trained weights. The network can be adjusted with a trail and error approach to get the best possible model, this achieved by changing the neurons used in the hidden layers, normally a neural network has one fully connected layer, a hidden layer and an output layer that predicts the results. The output layer has neurons equal to the number of classes for the problem.

Requirements in Skeleton Code

1] Graph showing the change in error over time for the Validation set:



2] Graph showing the change in error over time for the Train set:



3] Best validation error obtained is 0.469

4] Accuracy obtained for the test set is %58.666

Part 3: Support Vector Machines:

Description

SVM is support vector machine which is a supervised classifier that follows the concept of mapping points in a space that defines distance and each input is denoted by vectors and a decision boundary is defined to distinctly separate the vectors in space based on the predicted classes. SVM can also be used for non-linear classification by mapping non-linear data in a higher dimensional feature space.

The accuracy obtained from using the sci-kit-learn SVM model(SVC) on the training and testing files gives a testing accuracy of %70.3333.