CS F464 | Machine Learning
Aditya Agarwal (2017B1A71075H)
Manisha Kataria (2017B3A70354H)
Jalaj Bansal (2017B3A71610H)
Assignment 1

C. Perceptron Algorithm

1. A very brief description of your model and its implementation.

- Our prediction model is based on the perceptron algorithm which belongs to the discriminative class of algorithms. It builds a linear classifier using a linear predictor function that maps our input features to an output value. If it is a binary classifier like in our case, the function classifies as 0 or 1 or positive or negative class. The input features are assigned weights and these weights are learned by stochastic gradient descent algorithm to find the decision boundary.
- We have tried to keep our implementation as simple and readable as possible, using only numpy, pandas and math (optional) libraries.
- We read the dataset and assign arbitrary names to features and introduce x0 with all 1s as rows as a dummy feature which will help to have w0 as bias.
- The user can choose to normalize the dataset.
- Learning rate, number of epochs and threshold for stopping (optional) can be specified by the user.
- After the 70:30 split, we initialize random weights and feed them along with training data to our SGD algorithm to find the optimal weights.
- We calculate gradient and cost in each iteration and stop when cost is zero or epochs are finished.
- Grad = expected output - predicted output * x
- Cost = % misclassification = 100 * # Correctly classified samples / Total samples
- Weights are updated if there is misclassification depending on learning rate.
- w = w - (learnRate * grad)
- We then calculate test accuracy with the final weights to check convergence.
- We have also defined a predict function to predict class for new sample.

2. Accuracy of model

- Dataset 1 - 98.78% Test Accuracy ; 99.2% Highest training accuracy.
- Dataset 2 - 99.67-100 % Test Accuracy ; 100% Training accuracy.

3. Dataset which was more linearly separable

- Quite clearly, dataset 2 is more linearly separable as the algorithm quickly converges when training error for misclassification is zero.
- Even after 10^6 iterations, the algorithm did not converge for dataset 1.

4. Major limitations of the Perceptron classifier

- Linear models built by algorithms like the Perceptron classifier can only learn to approximate the functions for linearly separable datasets. A multi-layer perceptron or a neural network can help solve this problem.
- The algorithm is used only for Binary Classification problems. However, we can extend the algorithm to solve a multiclass classification problem by introducing one perceptron per class. i.e., each perceptron results in a 0 or 1 signifying whether or not the sample belongs to that class.