Assignment 1- (Logistic Regression & Neural Networks)

Ramesh Pavan Pothamsetty

10th October 2021

1 Assignment Overview

The goal of the assignment is to work with Pima Indians Diabetes dataset. In the first part of the assignment, I have implemented logistic regression using the using the sigmoidal basis function and log loss function. In the second part of the assignment, I have used a neural network with L1 regularization. In third part of the assignment, I have used L2 and Dropout regularizations and compared the accuracies between them.

2 Dataset

To implement linear regression models with Pima Indians Diabetes dataset with 768 samples. I have split data samples manually as training, validation and testing data, each constituting 60%, 20% and 20% respectively of overall data. Training dataset has 460 samples whereas validation and testing dataset has 154 samples each.

3 Python Editor

I have used Jupiter Notebook IDE for implementation.

4 Part-1

Feature extraction and data splitting

In the part one of the assignment, which is classification using logistic regression, I have used pandas data frame to store the data given diabetes.csv file. Then I have extracted features which are total in 8 to variable called “features” and outcome into variable called “diabetes_prediction”. I have not used any data normalization techniques. In the next stage I have done splitting of the data into test and temporary variables first as 60% and 40% respectively and the further divided the

temporary variable content into validation and testing with 20% of the original given dataset.

Cost estimation computation

In the data given we have eight independent variables and one dependent variable. Our goal using log loss function is to minimize loss and calculate weights Once we have predicted output and actual values, we calculate error/loss using the below function

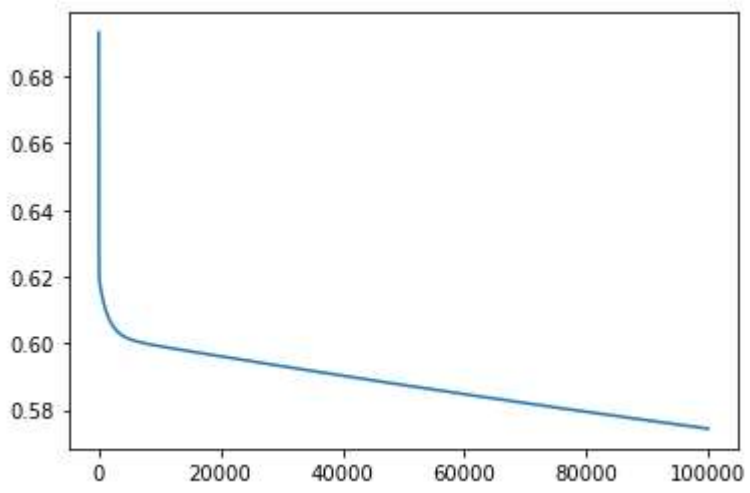
$$\text{Cost} = -1/m \sum_1^m (\text{ylog(p)} + (1 - \text{y})\text{log}(1 - \text{p}))$$

Logistic Regression Implementation

Cost estimation computation

In the implementation part I have used sigmoid function and log loss function above to calculate the cost. In further steps I have used gradient descent to calculate the weights for the model function which were initialized to zeros initially, for specific number of iterations and a constant learning rate which are hyper parameters here.

All the resulting cost values are stored in a list and the graph is plotted for the same.



Results

After finalizing weights which are obtained by a pair of iterations and learning rate, accuracies over test, validation and training sets are calculated which are presented below.

Accuracy for training data: 71.95%

Accuracy for validation data: 64.93%

Accuracy for testing data: 70.12%

5 Part-2

In the part two of the assignment, I have implemented neural network using L1 regularization for the classification which is imported from keras library. I have used two hidden layers using built-in Dense function with five neurons in the first hidden layer and eight neurons in the second. The parameters that I used to the Dense function are ReLU for activation function in hidden layer, sigmoid in output layer and L1 for kernel regularizer Since the classification is binary, I have used Binary Cross Entropy for loss. Also, I have chosen SGD (Stochastic Gradient Descent) for optimizer with default learning rate which are passed to model.compile() function.

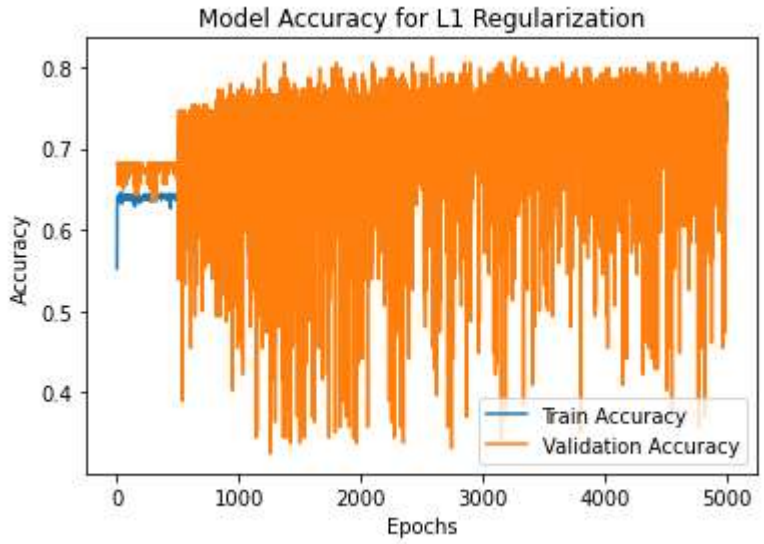
For training and validation of the model I have used model.fit() function to which training and validation datasets are passed. I have used batch gradient descent to which I have passed a batch size of 64.

Results

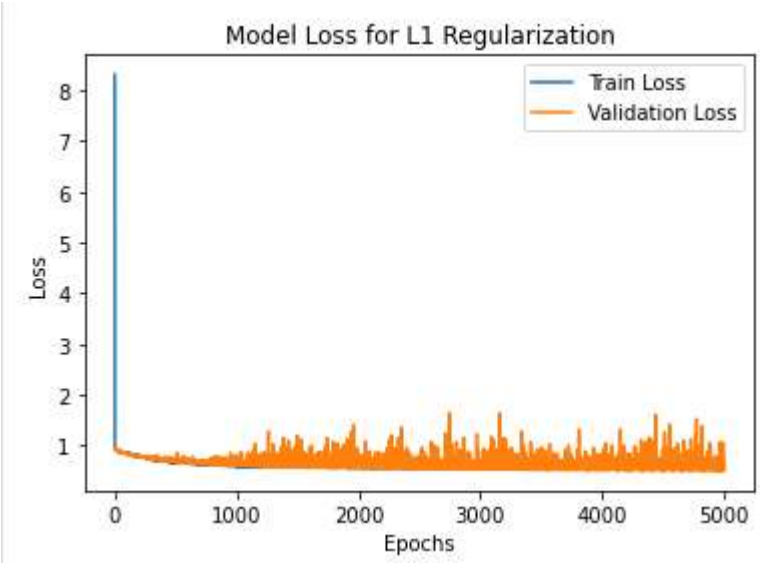
Below are the results for classification with neural network using L1 regularization.

Accuracy for testing data: 74.03%

Plot for training vs validation accuracy:



Plot for training vs validation loss:



5 Part-3

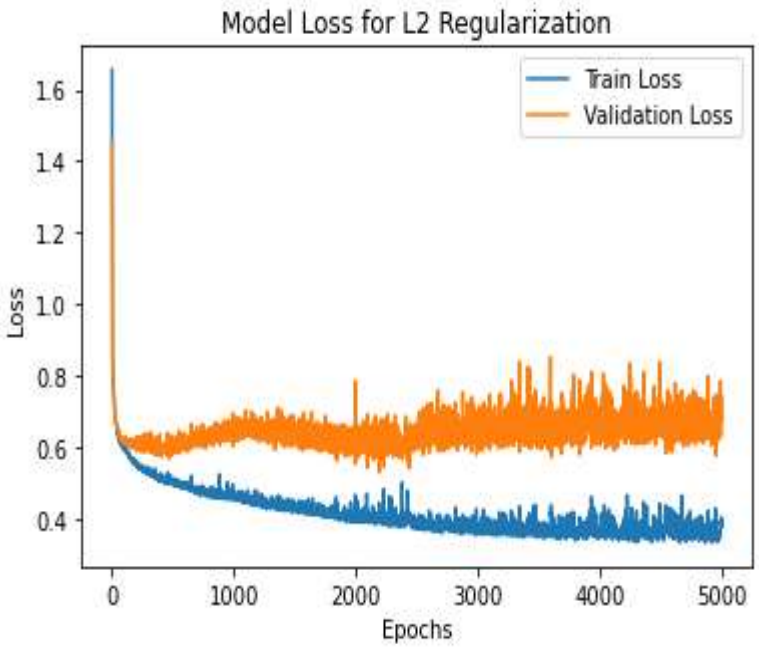
In part three of the assignment, I have implemented L2 and Dropout regularizations to draw out a comparison between the outputs. I have used a neural network with three hidden layers for both L2 and dropout regularization with eight neurons in each layer. When SGD optimizer is used for both the regularizations, it has been observed the test accuracy was about the same. In the next attempt I have tried Adam optimizer for L2 regularization and SGD for dropout. Using Adam optimizer improved the accuracy of the model.

Results

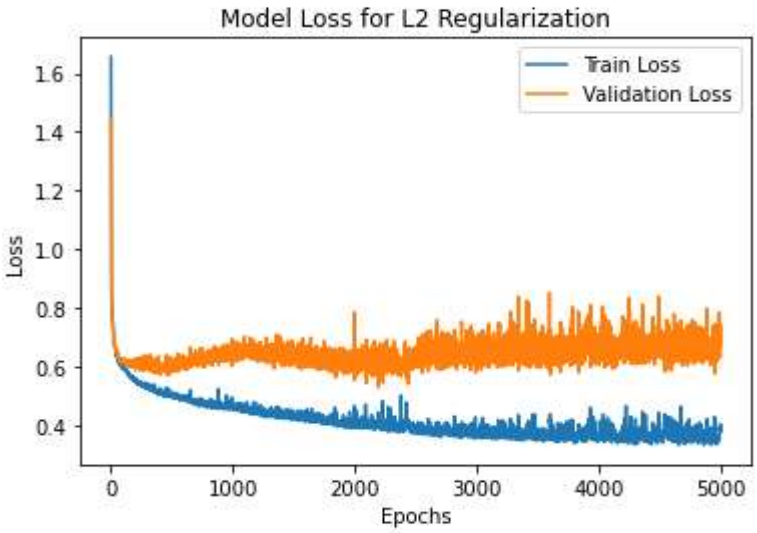
Below are the results for classification with neural network using L2 regularization.

Accuracy for testing data: 70.13%

Plot for training vs validation accuracy:



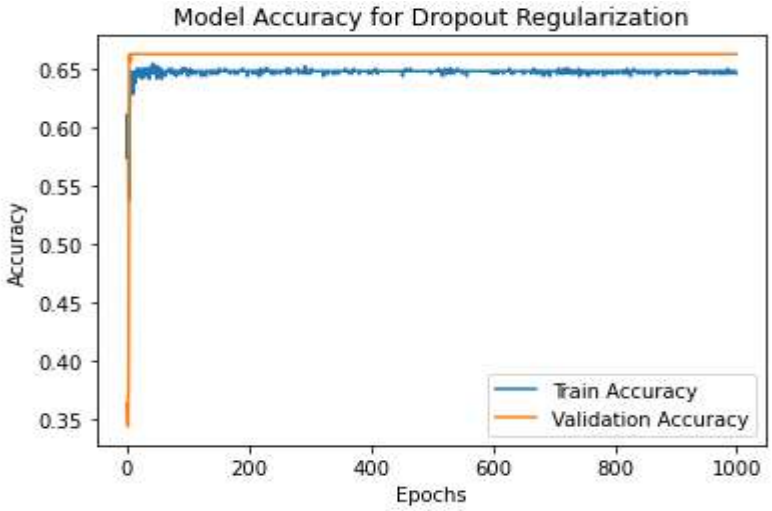
Plot for training vs validation loss:



Below are the results for classification with neural network using Dropout regularization.

Accuracy for testing data: 64.94%

Plot for training vs validation accuracy:



Plot for training vs validation loss: