Logistic_Regresion

Logistic regression is a popular classification problem, in which sigmoid function is used to calculate the posterior probability of a class.

The underlying formula for classification of the perceptron algorithm is given by:

$\hat{y} = \sigma(w^T \phi(x))$ where sigmoid function is given by:

$$S(x) = \frac{1}{1 + e^{-x}}$$

We will minimize the function

(1-y_pred)^(1-y) * (y_pred)^y

The resultant gradient function is

(y_pred - y) * X

The w value is updated by :

$$w^{\tau + 1} = w^{\tau} + \eta.\text{Unknown character}\text{Unknown character}\text{Unknown character}(\sigma(w^T \phi(x))),$$

Gradient Descent Algorithm :

Gradient Descent is an optimization algorithm for finding a local minimum of a differentiable function. Gradient descent is simply used to find the values of a function's parameters (coefficients) that minimize a cost function as far as possible.

Stochastic Gradient Descent Algorithm :

In stochastic gradient descent, the true gradient of is approximated by a gradient at a single example: As the algorithm sweeps through the training set, it performs the above update for each training example. Several passes can be made over the training set until the algorithm converges.

Implementation:

- We imported math, pandas, numpy, matplotlib as dependencies

- The dataset is shuffled using sample function to give 10 independent shuffles and then divided them in the ratio 70:30
- For Gradient Descent Learning we have chosen learning rate as 0.01 and epochs as 500 to get the best accuracy.
- In gradient descent we have trained the model on each of the 10 splits and plotted the accuracies and losses for every 25 epochs
- Also the average test metrics are displayed.
- Later, the same learning rate and epoch as to train the model using Stochastic Gradient Descent.
- Later we took three different learning rates to train both stochastic and Gradient descent and plot the accuracy vs epoch

Results

Important Feature:

The important Feature of the dataset is **attr1** (Feature 1) as it has an absolute maximum in the coefficient of the model for both Gradient and StoChastic Gradient descent methods.
The values of w are as below for 1st iteration:

Gradient Descent:

[12.49703683, -15.23102513, -7.60119343, -9.96931819, -1.21222686

Stochastic Gradient Descent:

[1.44958082 , -1.62655586, -0.95894024, -1.00853822, -0.18833901]

Test And Train Metrics:

Train Data:

Train Data	GD	SGD
Accuracy	0.98875	0.9875
Precision	0.9886402	0.9836405
Recall	0.98636590	0.9882888

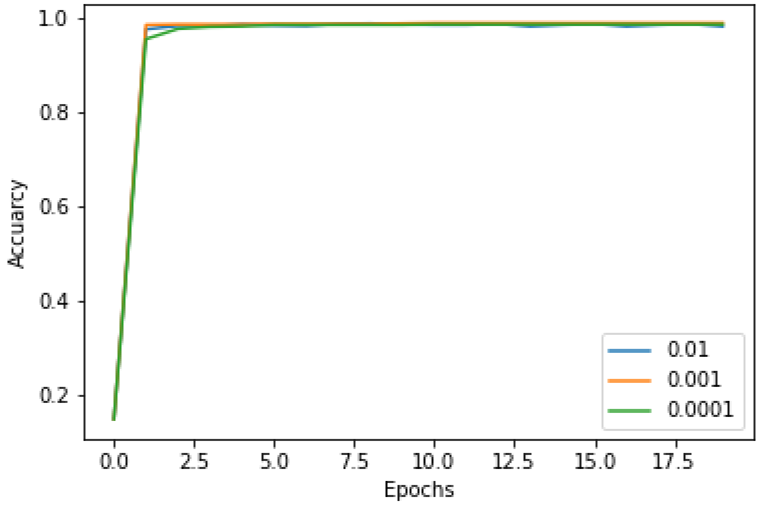
F1-score	0.98748749	0.9859497
loss	31.58988761	53.27844674

Test Data:

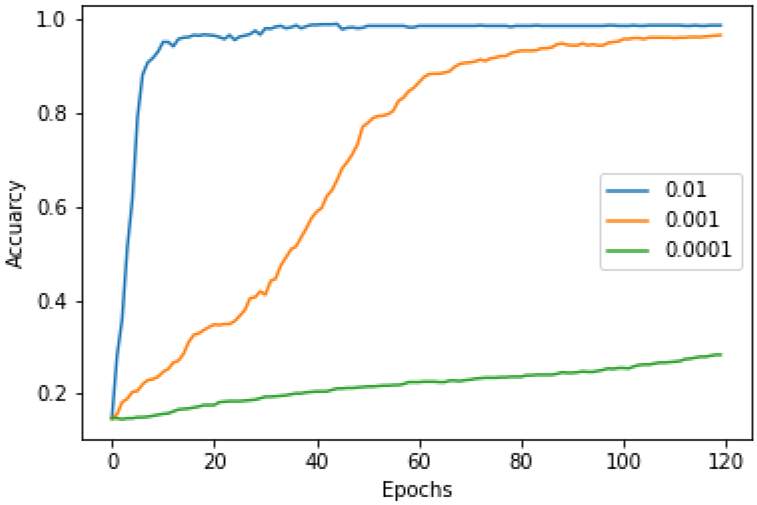
Test Data	GD	SGD
Accuracy	0.989320	0.9861650
Precision	0.989932	0.9816347
Recall	0.985684	0.9868953
F1-score	0.987742	0.9841887
loss	15.62878844	23.16564225

Plot for three different Learning Rates:

Gradient Descent:



Stochastic Gradient Descent:


1.0
0.8
0.6
0.4
0.2
Accuarcy
0
20
40
60
80
100
120
Epochs
0.01
0.001
0.0001