DS340W – Lab1
Daehoon Gwak

Verify effectiveness of incremental learning in logistic regression

1. Problem Statement

For this lab, we have six datasets and they were trained by six classifiers respectively. This occurred time inefficiency. So, our target would reduce computational process by using only one classifier to apply six datasets.

2. Method

We decided to use logistic regression with Stochastic Gradient Descent(SGD) to adjust the classifier. The main reason why we used SGD instead of Naïve Gradient Descent(GD) is because GD is iterative method updating parameters to optimize error function, but it requires every data point to calculate the gradient. On the other hand, SGD requires a single example to update the parameters to minimize error function, where it eventually improves the speed for the implementation. SGD also uses regularization term for its loss function, especially uses 'penalty' and 'alpha' parameter on its code. Penalty parameter contains 'l1' and 'elasticnet', where they might bring sparsity to the feature selection which is not achievable with 'l2'. 'l2' represents standard linear regularization parameter. 'alpha' is another regularization term which multiplies the regularization term. Mathematically, this regularization term would be:

$$J_{regularized} = \underbrace{-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log\left(a^{[L](i)}\right) + (1-y^{(i)})\log\left(1-a^{[L](i)}\right)\right)}_{\text{cross-entropy cost}} + \underbrace{\frac{1}{m}\frac{\lambda}{2}\sum_{l}\sum_{k}\sum_{j} W_{k,j}^{[l]2}}_{\text{L2 regularization cost}}$$

Now, we need to find the stopping criteria for this function such as *tol*, *max_iter*, and *early_stopping*. *tol* tells us when we have to stop since it stops when (loss > best_loss - tol) for consecutive epochs. *max_iter* shows the maximum number of epochs and it is useful to figure out the best accuracy of the model. *early_stopping* helps us to decide to terminate training when validation score is not improving. In the actual code for this Lab, we used *tol=None.*

3. Experiment

1) Data Preparation

We have 6 datasets and they represent six consecutive days(0 to 6). each data file contains two features variables and one class variable.

2) Find the best learning rate using day0 file

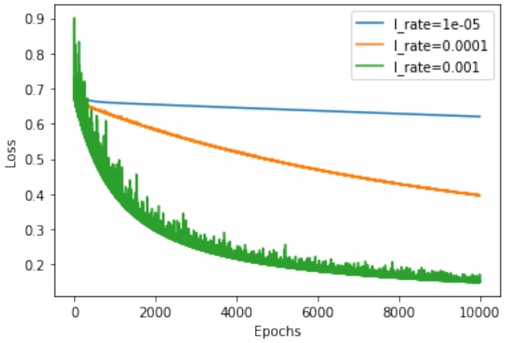

Figure 1: Learning rate plot

> As we can see in 'Figure 1', the loss function iterates for 10000 epochs. As the number of epochs increase, all the learning rate seem to be decrease. However, rate '0.001' shows the lowest loss as the epoch increases and it decreases steeply compared to other rates. As a result, we decided to choose the best learning rate is 0.001. This also satisfies our goal which is to minimize the loss value.

3) Parameter Tuning – find the best accuracy
> We have used the parameters on the SDGClassifier such as:

a. *loss = 'log'*: we are using logistic regression for loss function.
b. *penalty = 'l2'*: it is used for regularization purpose and fits in this case.
c. *alpha=0.0005*: 0.0005 value shows the highest accuracy in this case.
d. *max_iter=10000*: maximum number of iterations and shows the fair amount of accuracy.
e. *tol=None*: Set it to None because training won`t stop when (loss > best_loss - tol) for consecutive epochs.
f. *learning_rate='constant'*: literally meaning learning rate and in this case, we set it to constant because we use parameter 'eta0' later.
g. *eta0=1e-3*: it is an initial learning rate, and in this case, it is 0.001

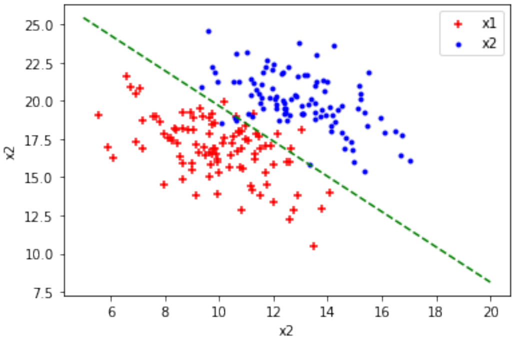

Figure 2: Plot with decision boundary

> The decision boundary looks fine and the accuracy for this model is 95.50%

4) Adjust to the other days

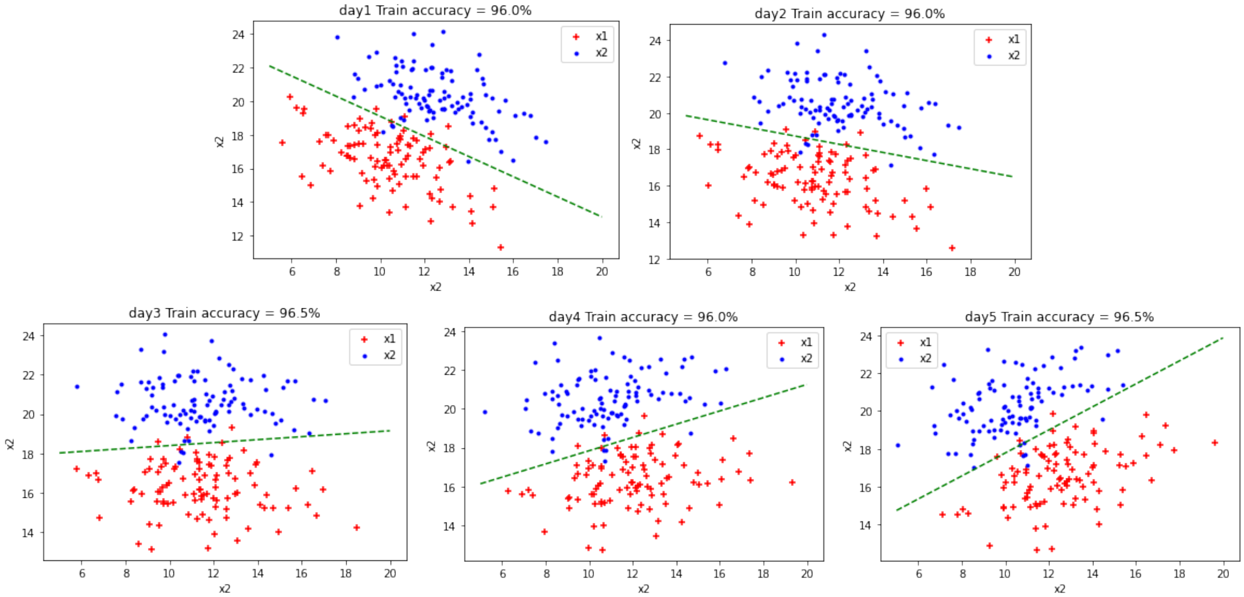

Figure 3: Plot with decision boundary for day1 to day5

> As we can see, the accuracies are about the same as we got in day0 data.

3. Final Thoughts

> As we implemented the incremental learning for the model, we iterated about 100 times to get about same accuracy for day0 data. It is very interesting to know that the number of epochs decreased from 10,000 to 100 times and still have about the same accuracy. Plus, it saves time since we only apply one classifier to other multiple datasets where we had a problem at the beginning of this Lab. Considering the result, we can finalize that it is more efficient to use incremental learning for logistic regression classifiers.