CSI 5325 Assignment 2

Sadia Nasrin Tisha

1st March 2022

1 Question 01:

- **Create Dataset:** I have generate 100 point dataset with +1 and -1 class here $C0$ is the centre -1 classes and $c1$ is the centre of +1 class. Here axes are normalized for viewing axis equal. Figure1 shows that the data are linearly separable.

- **Perceptron Learning Algorithm(PLA):** After run PLA starting from w = 0 until it converge, Figure2 shows the plot of data and hypothesis where PLA works well to separate the data linearly and achieve close solution. It denotes that there is no miss classified data between +1 and -1 classes.

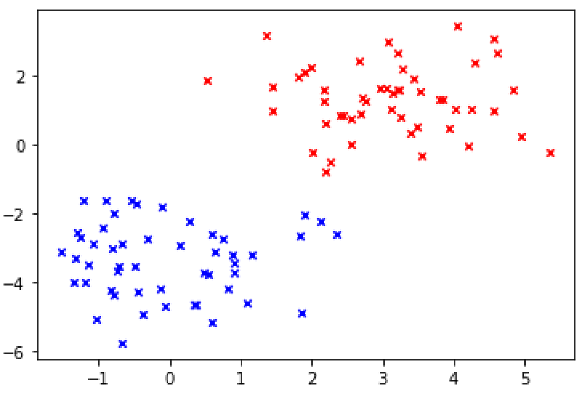

Figure 1: Linearly Separable Data

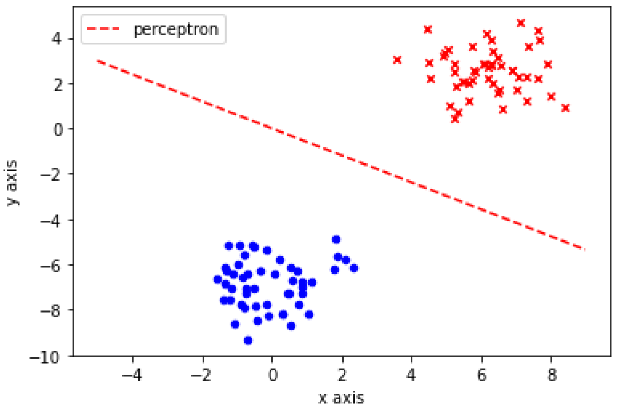

Figure 2: Applying PLA

After that I applied three apporaches and calculated the E_{in} and E_{out} and observe the results shows in given below:-

1. **The Pocket algorithm, starting from w = 0:** At first dataset is splited where there is 30% data in test set and 70% data in training set. Then run the pocket algorithm in training set with w=0 and calculate the E_{in}. After that based on the updated w in training set, E_{out} is calculated in test set. I observed the E_{in} E_{out} based on the iteration in pocket algorithm. For 50, 100, 500, 1000, 3000 iteration, in all cases the $E_{in} = 0$, that means there is no miss classified data in training sample. On the other hand, for lower iteration, the $E_{out} = 1$ for 50 iteration. But for 100, 500, 1000 and 3000 iteration the $E_{out} = 0$ shows in Figure3. So the result says that higher iteration helps to deal with miss-classification. After adding some significant outliers to the y = +1 class, there shows some missclassification in the algorithm and E_{out} and E_{in} error rate increase in these algorithm.

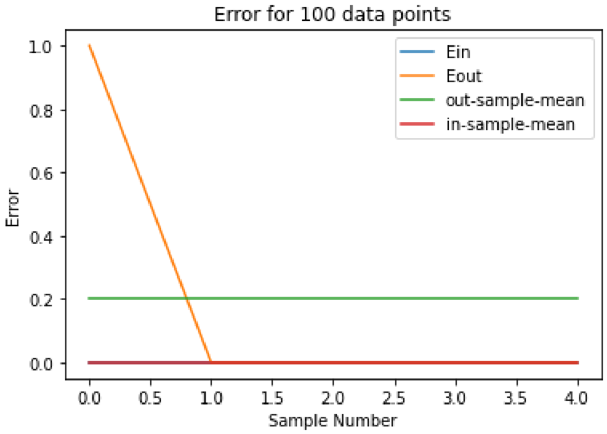

Figure 3: Error observation for 100 datapoints

2. **Linear regression:** In randomly generated 100 data points like previous, Linear Regression algorithm is applied in the where the Mean Absolute Error and Mean Squared Error is 0. That means that there is no miss classification after applying Linear Regression shows in figure4 . The updated w of model coefficient is used to calculate the E_{out} in test set where the value of $E_{out} = 4$ that means there is some missclassification in out sample. Here for 100, 500, 1000 iteration the in sample error shows 0 and E_{out} is in beetween 1-4. After adding some significant outliers to the y = +1 class, there is also showing some missclassification in the algorithm and E_{out} and E_{in} error rate increase in these algorithm.

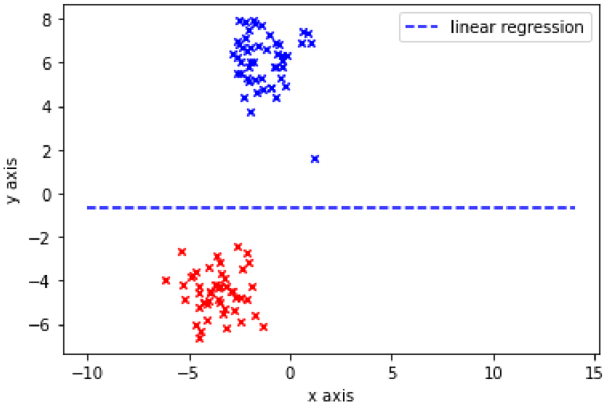

Figure 4: Linear Regression

3. **The Pocket algorithm, starting from the solution given by linear regression:** Again pocket algorithm is applied with updated w in linear regression where the calculated $E_{in} = 0$ in training set and also $E_{out} = 0$ for test set for 1000 iteration.

After applying two algorithms on a classification task, it shows that, both algorithm works well for separating the data linearly with minimum error. Pocket algorithm shows zero in sample error and also shows zero out sample error for greater iterations. On the other hand, linear regression have zero in sample error but there was some out sample error because of miss-classification. But after After adding some significant outliers to the y = +1 class, in both algorithm shows some missclassification and the error rate got increase. For different iteration pocket algorithm shows inconsistent E_{out} rate where in linear regression the E_{out} rate was consistent in beetween 0-4. It is because linear regression can handle outliers and missclassified data better than pocket algorithm and in my observation as the data was linearly separable that is why both algorithm works very well.

2 Question 2

1. Given that,
   $\sigma(\alpha) = \frac{1}{1+exp(-\alpha)}$

   Here,
   $\frac{\partial\sigma}{\partial\alpha} = -(\frac{1}{1+exp(-\alpha)})^2(-exp(-\alpha))$
   $\Rightarrow \frac{\partial\sigma}{\partial\alpha} = exp(-\alpha) \times \frac{1}{1+exp(-\alpha)} \times \frac{1}{1+exp(-\alpha)}$
   $\Rightarrow \frac{\partial\sigma}{\partial\alpha} = 1 - \frac{1}{1+exp(-\alpha)} \times \frac{1}{1+exp(-\alpha)}$
   $\Rightarrow \frac{\partial\sigma}{\partial\alpha} = (1-\sigma(\alpha))\sigma(\alpha)$

   Hence, $\frac{\partial\sigma}{\partial\alpha} = \sigma(\alpha)(1-\sigma(\alpha))$

2. Given that,
   $\ell(w) = log\prod_{n=1}^{N} P(y_n|x_n) = \sum_{n=1}^{N} logP(y_n|x_n)$

   $P(y|x) = \sigma(yw^Tx)$
   From (1) we got that, $\frac{\partial\sigma}{\partial\alpha} = \sigma(\alpha)(1-\sigma(\alpha))$
   Gradient of the log-likelihood,

   $$\nabla_w\ell(w) = \frac{\partial\ell(w)}{\partial w} = \sum_{n=1}^{N}\frac{\partial}{\partial w}logP(y_n|x_n)$$

   $$= \sum_{n=1}^{N}\frac{\partial}{\partial w}log(\sigma(y_nw^Tx_n))$$

   $$= \sum_{n=1}^{N}\frac{1}{\sigma(y_nw^Tx_n)}\frac{\partial}{\partial w}\sigma(y_nw^Tx_n)$$

   $$= \sum_{n=1}^{N}\frac{1}{\sigma(y_nw^Tx_n)}\sigma(y_nw^Tx_n)(1-\sigma(y_nw^Tx_n))y_nx_n$$

   $$= \sum_{n=1}^{N}(1-\sigma(y_nw^Tx_n))y_nx_n$$

   Hence. $\nabla_w\ell(w) = \sum_{n=1}^{N}(1-\sigma(y_nw^Tx_n))y_nx_n$

3. From(2) we got that, $\nabla_w\ell(w) = \sum_{n=1}^{N}(1-\sigma(y_nw^Tx_n))y_nx_n$

   Update step for gradient ascent of $\ell(w)$ using the gradient,

```
(a) Initialize the weights at time step $t = 0$ to $w(0)$ .
(b) for $t = 0, 1, 2, \dots$ do
(c)     Compute the gradient
            $$g_t = \nabla_w \ell(w) = \sum_{n=1}^{N} (1 - \sigma(y_n w^T x_n)) y_n x_n$$
(d)     Set the direction to move, $V_t = g_t$
(e)     Update the weights: $w(t + 1) = w(t) + \eta V_t$.
(f)     Iterate to the next step until it is time to stop.
(g) Return the final weights $w$
```

4. After implementing the gradient ascent for learning logistic regression, like Question 1, I have made 100 point random dataset and fit the model with different learning rate and observe the iteration for learning, the weight and E_{in} and E_{out}. I have used multiple learning rate where in all learning rate the value of E_{in} is almost similar which is between 28-36. But for learning rate 0.005 I found E_{out} is lower than previous and for each iteration the E_{out} converge almost 0 that means the converge to local minimum where the steps proportional to the positive of the gradient of the function at the current point .

   On the other hand, I have also run linear regression and pocket algorithm in the same datapoint to compare the result where linear regression gives lower E_{in} and E_{out} with lower iteration than logistic regression. On the other hand, pocket algorithm gives 0 E_{in} and E_{out}, which is surprising for me.

3 Programming Problems