Machine Learning
Assignment 1 Report
Abhishek Agarwal(2016126)

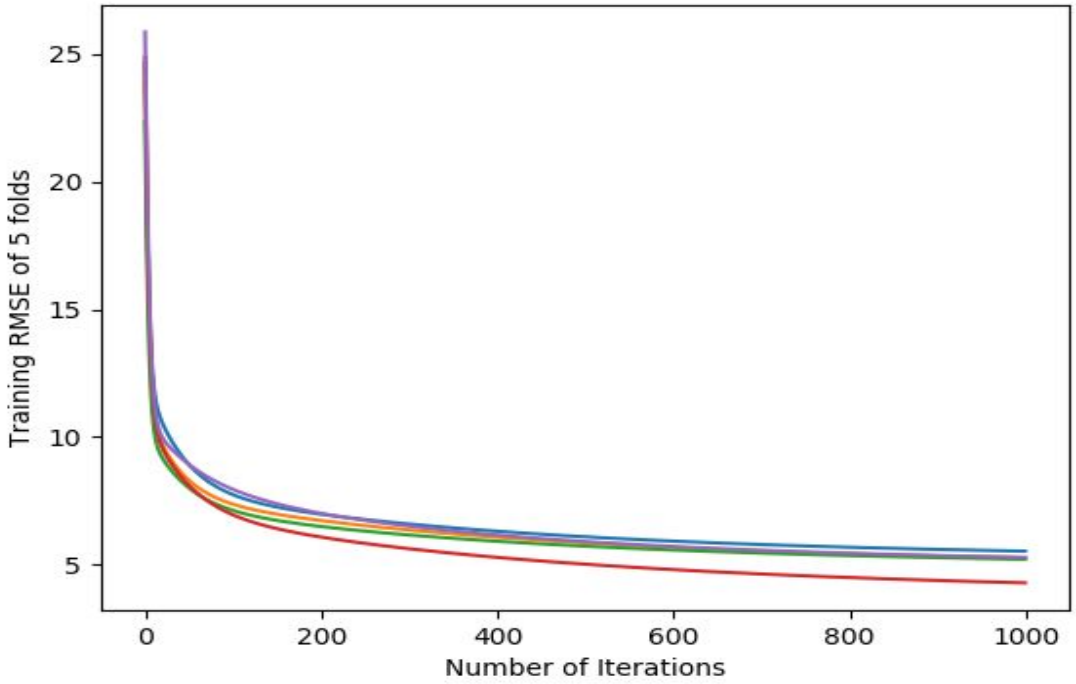
Ques 1:

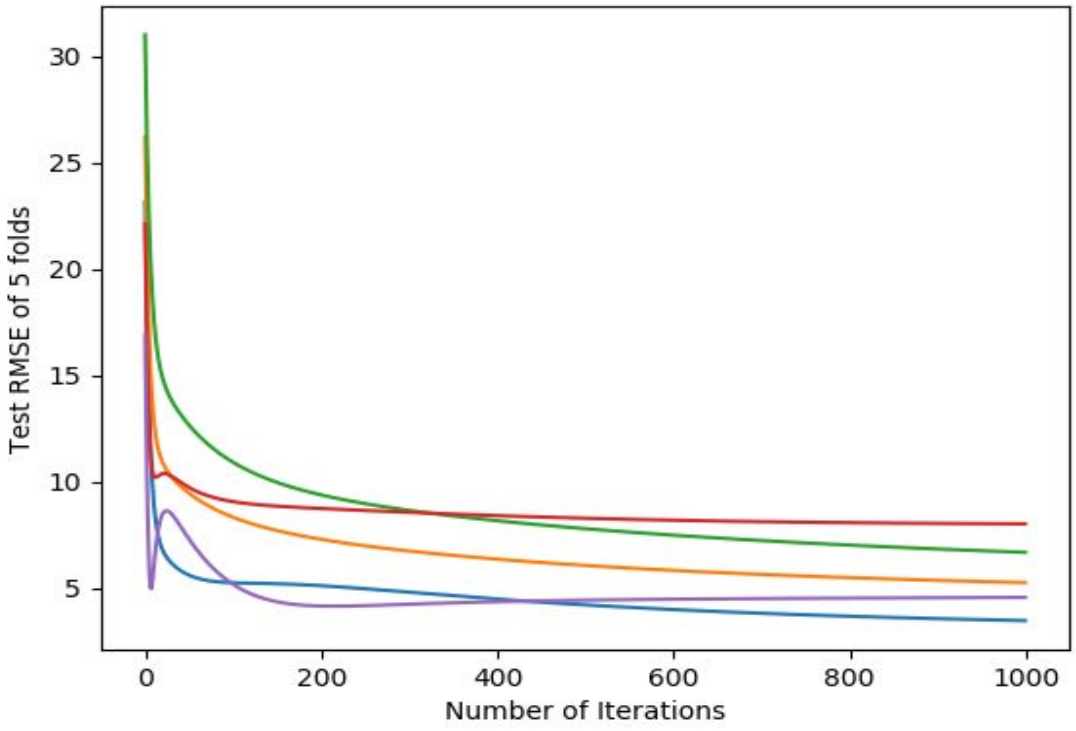
The first part has been done in the file with name q1a.py.
It consists of the following functions:

- *linear_regression(training_set_x, test_set_x, training_set_y, test_set_y, fold_num) :*

This function takes in 5 input parameters, which are explanatory from the name itself.
It calculates the RMSE at each step and updates the value of parameters after each iteration by calculating the gradient and the cost function.
It also plots the graph of RMSE vs number of iterations each time we call it for each fold.

The curve of RMSE of all these 5 folds (together) vs the number of iterations is plotted for both train and test set.





The learning rate is kept 0.04. The number of iterations are 1000.

As we can clearly see, the RMSE keeps on decreasing as the number of iterations increase.

The final train RMSE values for each fold are:

Fold 1: 5.731726835081182
Fold 2: 5.483756078000321
Fold 3: 5.401340799636045
Fold 4: 4.562085473309591
Fold 5: 5.5028186718661125

The final test RMSE values for each fold are:

Fold 1: 3.7247564594128666
Fold 2: 5.5464568496974795
Fold 3: 7.108837010316249
Fold 4: 8.082915050760176
Fold 5: 4.502318550168285

The Mean Train RMSE and Standard Deviation of all folds are as follows:

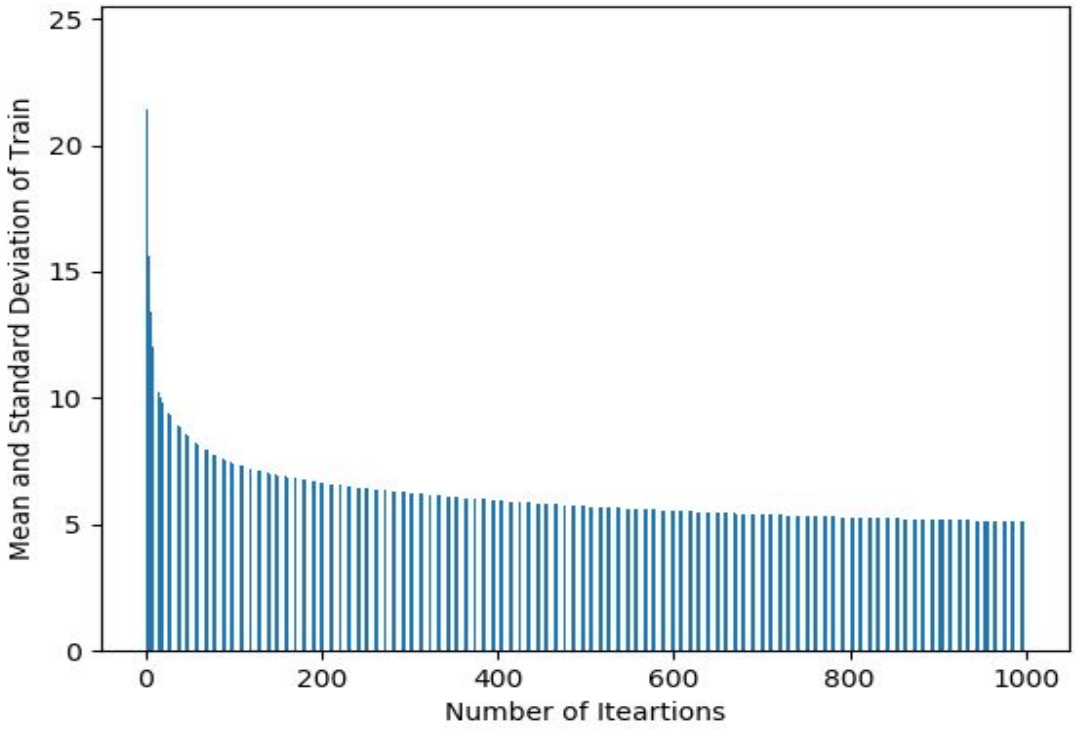
Training set:

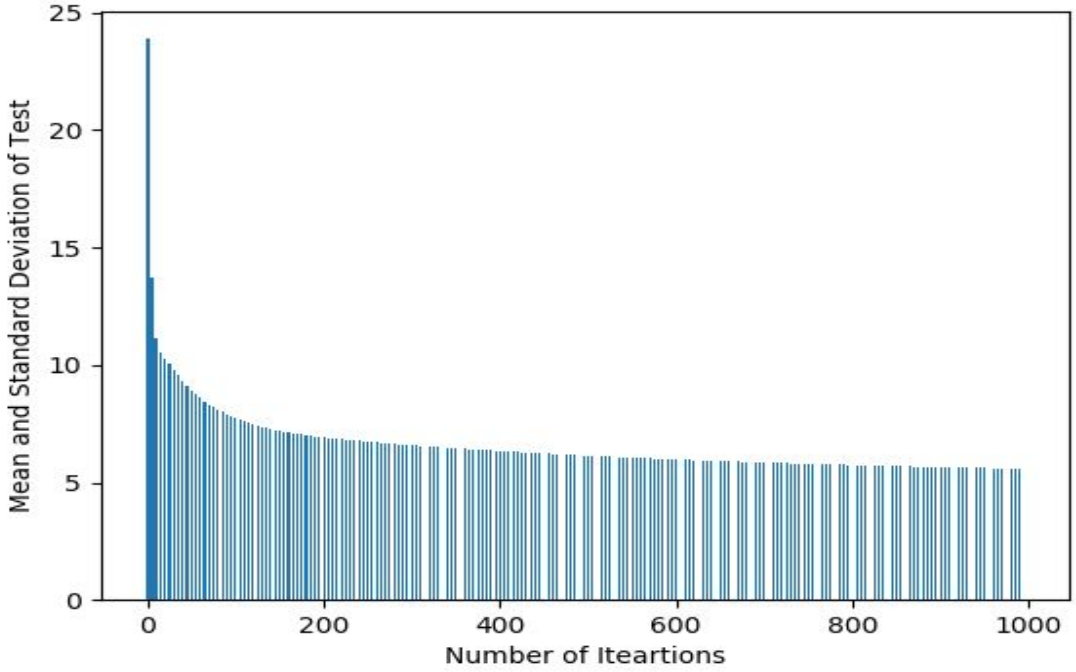
(5.336345571578651 +- 0.4498534687799787)

Test set:

(5.793056784071011 +- 1.8013475118904057)

- The graph of mean RMSE of all folds vs the number of iterations with standard deviation on each iteration is as follows:
  (Train and Test set respectively)


Mean and Standard Deviation of Train
Number of Iteartions
0
5
10
15
20
25
0
200
400
600
800
1000

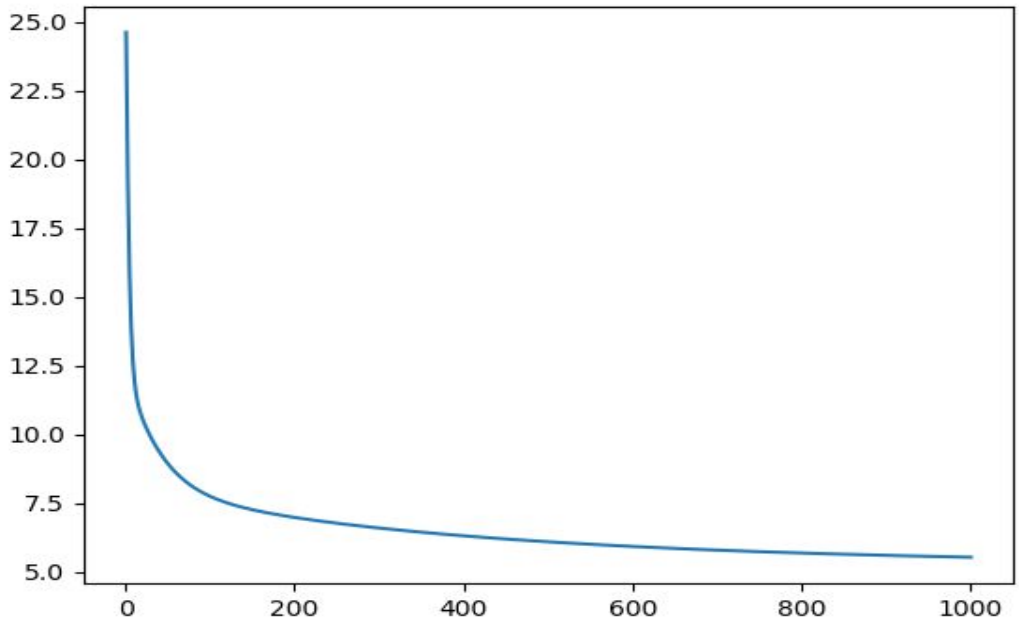

Mean and Standard Deviation of Test
Number of Iteartions
0
5
10
15
20
25
0
200
400
600
800
1000

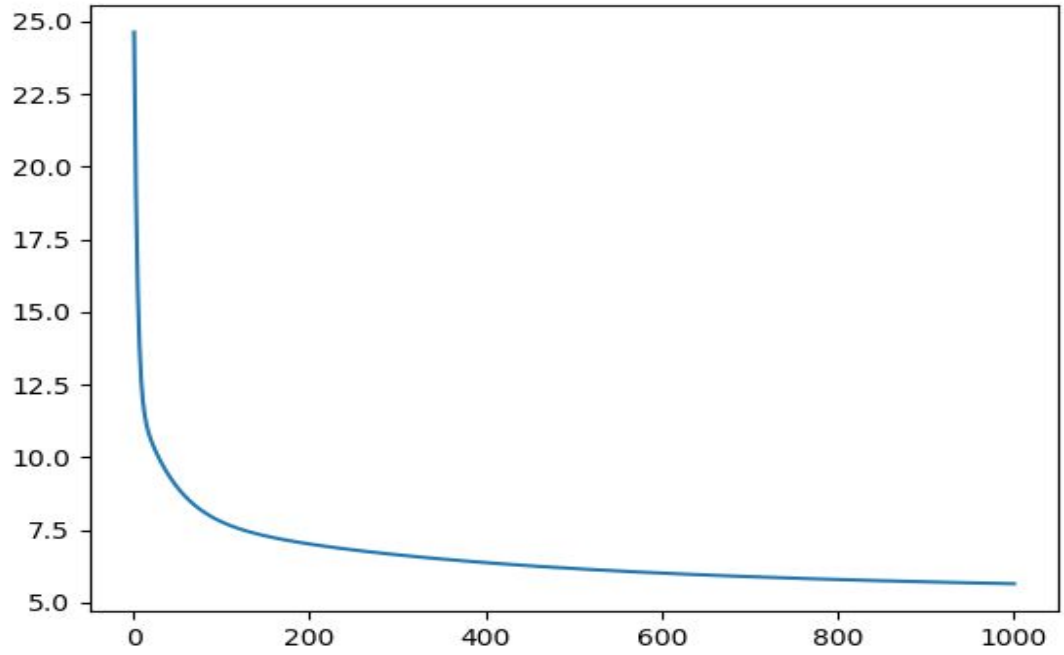
1(ii):

The code for this part is in 1(ii)
Fold 1 had the least test RMSE for part(i), therefore that is used here.
On performing L1 and L2 regularisation using Grid Search to find the hyperparameter, the following plots are obtained: (L1 and L2 respectively)

The Lambda chosen by Gridsearch for L2 is 3 and for L1 is 0.005.

Training RMSE after L2 regularization is: 5.6436657745252505
Test RMSE after L2 regularization is: 3.567376861368213
Running L1
Params = 0.005
Training RMSE after L1 regularization is: 5.5341532338879
Test RMSE after L1 regularization is: 3.43280589527201


25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
0
200
400
600
800
1000


25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
0
200
400
600
800
1000

1(iii):

The test RMSE for fold 1 originally was 3.7247564594128666.
After performing L1 and L2 regularization, the RMSE decreases to 3.567376861368213 and 3.43280589527201 respectively.
Thus, we observe that all the three models with and without regularization are a good fit to our data. This is because the difference in train and test error is not much in any of the folds.
In fold 4 without regularisation, we do obtain a slight overfitting of data because the difference in train and test error is more than 4.

Q2)

The file is named q2.py.

Scikit learn library has been used to implement Logistic Regression for both L1 and L2.

The file has this funtion:

modify_y(current_class, y_values):

This takes input the current class number and the y values of our data. It modifies them by renaming the current class labels to 1 and the other classes to 0.

This modified y is then used to pass in the .fit function along with the x values.

The parameters obtained after training are stored in an array for both regularizations. Accuracy for each of them is then calculated by taking the ratio of correct predictions to the total number of predictions made for each class.

The accuracy obtained for L1 is as follows:

Train set:

Class 0: 97.9219038
Class 1: 82.7673518
Class 2: 83.68841545
Class 3: 89.43142717
Class 4: 81.4159292
Class 5: 37.79673591
Class 6: 95.4460174
Class 7: 79.50191571
Class 8: 97.74243202
Class 9: 89.60777741

Test Set:

Class 0: 96.9017094
Class 1: 82.71711092
Class 2: 83.09572301
Class 3: 89.30635838
Class 4: 79.32489451
Class 5: 36.80781759
Class 6: 96.15004936
Class 7: 78.71720117
Class 8: 96.11451943
Class 9: 85.98790323

For L2 the accuracy obtained is:

Train set:

Class 0: 97.80459192
Class 1: 98.10843015
Class 2: 91.04527297

Class 3: 90.36539407
Class 4: 93.82232811
Class 5: 87.05489614
Class 6: 96.19648644
Class 7: 93.80587484
Class 8: 88.30169318
Class 9: 89.92624874

Test set:

Class 0: 97.00854701
Class 1: 96.38865004
Class 2: 89.30753564
Class 3: 89.4026975
Class 4: 92.08860759
Class 5: 84.47339848
Class 6: 96.93978282
Class 7: 91.5451895
Class 8: 86.80981595
Class 9: 88.40725806

The train and test accuracy for L1 is: 83.90833% and 82.8% respectively.
The train and test accuracy for L2 is: 92.765% and 91.35% respectively.

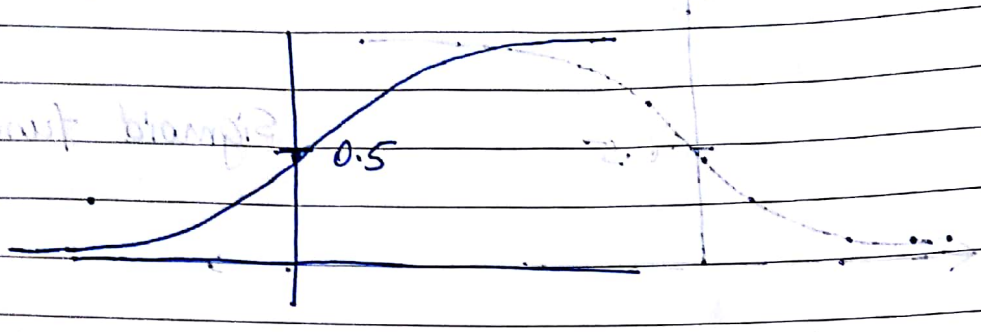
2(iii) : On observing the results and accuracies for L1 and L2 regression, we see that the models do not overfit or underfit the data. The difference in train and test accuracies is very very low for each class. Hence, the models are a good fit for the dataset.

ML ASSIGNMENT-1 (ABHISHEK AGARWAL 2016126)

DATE ___/___/___

Q.3 $\sigma(z) = \frac{1}{1+e^{-z}}$

Sigmoid is a non linear function



Sigmoid function is used in logistic reg. to model the probability of belonging to a class :

To find that, we pass $\theta^T x$ in place of z.

$$\sigma = \frac{1}{1+e^{-\theta^T x_i}}$$

Where θ is the vector of weights and x_i is the input vector of parameters.

Despite of the sigmoid function being non linear, binary classification is done by logistic regression uses linear seperation.

This is because, the decision boundary generated by the sigmoid function is linear.

That is,

The decision boundary is

$$w^T x = 0$$

which is the eqn of a hyperplane and is also linear in w's and x's.

DATE ____/____/____

All the probabilities get mapped to a value of $w^T x$.

For any p, we can find $w^T x$.

We classify as follows:

For $w^T x > p$; we have $p > 0.5$

for $w^T x < 0$, we have $p < 0.5$

Thus for $p \geq 0.5$ we can put in class 1

for $p < 0.5$ we can put in class 0.

$\therefore$ $w^T x = 0$ has created a linear boundary seperating the two classes.

DATE ___/___/___

Q.4 Logit transformation
Logistic Regression

Logit ⇒ log of odds
where odds are defined as

$$\text{Odds} = \frac{p}{1-p}$$

Since linear regression gives us output in the range $(-\infty, \infty)$, we need a function which can map it to $[0,1]$ so that we can use Logistic Regression and classify it into two distinct classes class 0 and class 1.

Let the probability of a random variable Y to be mapped to $\{0,1\}$ be defined as:

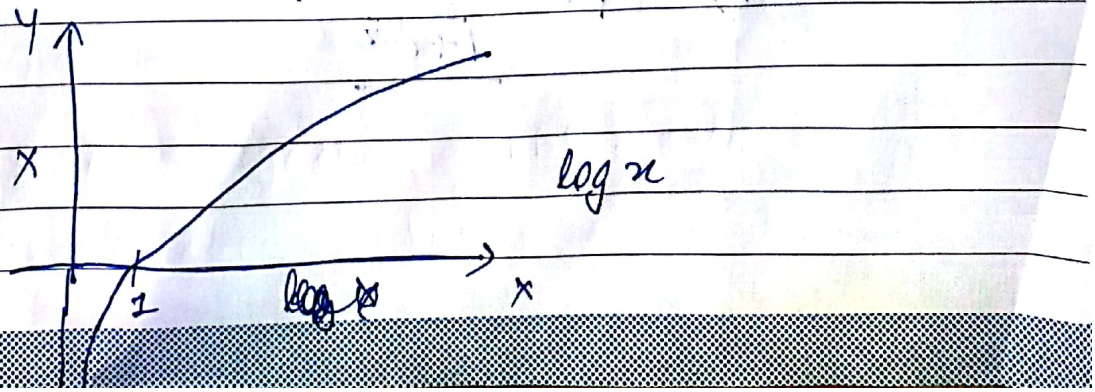
$$P(y) \begin{cases} p & y=1 \\ 1-p & y=0 \end{cases}$$

This can also be written as: $p^y(1-p)^{1-y}$

Linear regression cannot be used for binary classes
∴ to map the range of logistic regression to that of linear, we use logit transformation.

As $p \in [0,1]$

$$\text{odds} = \frac{p}{1-p} \in [0, \infty)$$



DATE ___/___/___

log function takes input from $(0, \infty)$

∴ We put the odds inside log function and obtain an output in the range $(-\infty, \infty)$

∴ $\log(\text{odds}) \in (-\infty, \infty)$

logit(p) is defined as log(odds)

Thus we can use linear regression to predict log(odds).

$\log(\text{odds}) = w_0 + w_1 x_1 + \dots + w_d x_d$

$= w^T x$

$\text{logit}(p) = w^T x$

$\log\left(\frac{p}{1-p}\right) = w^T x \quad \rightarrow (1)$

To get the p from $w^T x$ we use:

$(1) \Rightarrow \frac{p}{1-p} = e^{w^T x}$

$p = \frac{e^{w^T x}}{1 + e^{w^T x}} = \frac{1}{1 + e^{-w^T x}}$

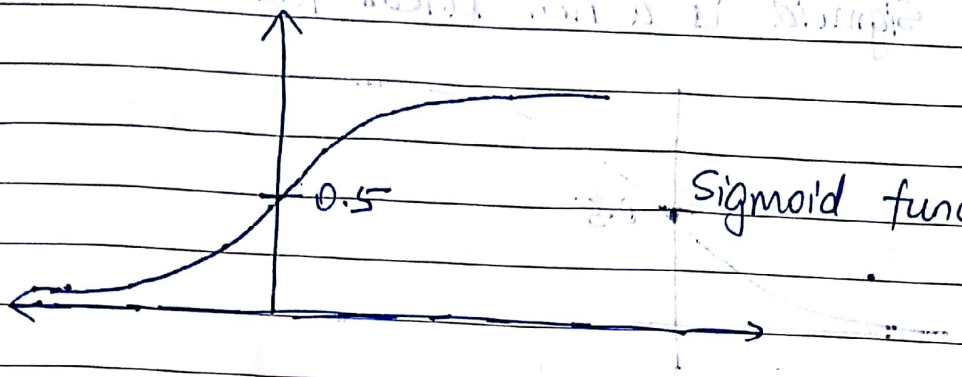
Let $w^T x = z$

$p = \frac{1}{1 + e^{-z}}$

Also, $1 - p = \frac{e^{-z}}{1 + e^{-z}}$

DATE ___/___/___

This value of p_i also known as the sigmoid function



When the value of $w^T x_i = 0$, the value of sigmoid is $1/2$.

From PMF of y: $p^y (1-p)^y$

Log likelihood function is:

$$l(w) = \sum_{i=1}^{N} y_i \log(p(x_i; w)) + (1-y_i) \log(1 - p(x_i; w))$$

$$l(w) = \sum_{i=1}^{N} y_i \{ \log p(x_i; w) - \log(1-p(x_i, w)) \} + \log(1-p(x_i; w))$$

$$= \sum_{i=1}^{N} y_i \log\left(\frac{p(x_i; w)}{1-p(x_i; w)}\right) + \log\left(\frac{1}{1+e^{w^T x_i}}\right)$$

$$l(w) = \sum_{i=1}^{N} y_i w^T x_i + \log(1 + e^{w^T x_i})$$

To find max likelihood, we differentiate

$$\frac{\partial}{\partial w^j} l(w) = \sum_{i=1}^{N} x_i^j (y_i - p(x_i; w))$$

∴ we obtain the gradient ascent rule:

$$\boxed{w^j(t+1) = w^j(t) + \eta \sum_{i=1}^{N} x_i^j (y_i - p(x_i; w))}$$

where $w^j(t+1)$ denotes the next w's on $(t+1)^{th}$ iteration

Q.5 T.P: Entropy of multivariate Gaussian variable $X \sim N(\mu, \Sigma)$ is:

$$H[x] = \frac{1}{2}\ln|\Sigma| + \frac{D}{2}(1 + \ln(2\pi))$$

Soln D: Length of x vector

Σ: covariance matrix

Since, $H(p) = -\int p(x) \ln(p(x))\, dx$

$\therefore \quad H(X) = -\int N(X|\mu, \Sigma) \ln N(X|\mu, \Sigma)\, dx$

Since The PDF of multivariate gaussian is given as:

$$f_X(x_1, x_2 \cdots, x_k | \mu, \Sigma) = \frac{\exp\left[\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]}{\sqrt{(2\pi)^k |\Sigma|}}$$

Substituting this value,

$$H(X) = -\int N(X|\mu, \Sigma) \ln\left[\frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)\right] dx$$

$$H(X) = -\int N(X|\mu, \Sigma) \ln\left(\frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}}\right) dx \quad \text{(I)}$$

$$-\int N(X|\mu, \Sigma) \ln\left(\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)\right) dx \quad \text{(II)}$$

Term I is integration of PDF multiplied by a constant since, Integration of PDF will be 1.

DATE ____/____/____

$$H(X) = -\ln\left((2\pi)^{-D/2}|\Sigma|^{-1/2}\right) - \int N(X|\mu,\Sigma)\left(\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) dx$$

This can also be written as:

$$= \frac{D}{2}\ln(2\pi) + \frac{1}{2}\ln|\Sigma| + \frac{1}{2}\int \text{tr}\left[\Sigma^{-1}(x-\mu)(x-\mu)^T\right] N(X|\mu,\Sigma)\, dx$$

$$= \frac{D}{2}\ln|2\pi| + \frac{1}{2}\ln|\Sigma| + \frac{1}{2}\text{tr}\left[\Sigma^{-1}\int (X-\mu)(X-\mu)^T N(X|\mu,\Sigma)\right]$$

$$= \frac{D}{2}\ln|2\pi| + \frac{1}{2}\ln|\Sigma| + \frac{1}{2}\text{tr}\left(\Sigma^{-1}\Sigma\right)$$

$$= \frac{D}{2}\ln|2\pi| + \frac{1}{2}\ln|\Sigma| + \frac{1}{2}\text{tr}(I)$$

$$= \frac{D}{2}(\ln 2\pi) + \frac{1}{2}\ln|\Sigma| + \frac{D}{2}$$

$$\boxed{H[X] = \frac{1}{2}\ln|\Sigma| + \frac{D}{2}\left[1 + \ln(2\pi)\right]}$$