CS 584 - HM3

Alaa Ayach A20317680

Problem Statement

In this assignment I am trying to implement Discriminative learning for both Logistic and Neural networks algorithms as discriminant functions. I changed parameters such as data, number of layers, and number of classes.
I analyzed the differences comparing the different results based on confusion Matrix.

Proposed Solution

I implemented all functions from scratch using only matrix manipulation given by R. I used aggregate function to compute sums.
I used confusion Matrix to come up with the error.

Implementation Details

I encapsulate all the work in a modular way using files and parameters. Those files took the parameters that I talked about above. used some package like cvTools to help in cross validation.

Results and Discussion

1 Logistic regression two classes (Logistic Iris.R)

I loaded the Iris dataset using the first two classes making Iris-setosa class =0 and Iris-versicolor class = 1

confusion matrix on ***test data***

yhat	Iris-setosa	Iris-versicolor
Iris-setosa	0	10
Iris-versicolor	10	0

The results are perfect as the Iris dataset is well classified

2 Logistic regression combinations (Logistic Iris non-linear.R) & Evaluation

I used here combinations from the features I have,
for example in iris dataset I have 4 features let them be x1, x2 , x3, x4
I will add x5 = 8 * sigmoid (x1), x6 = 4 * sigmoid(x2)

I got this result from that:

yhat	Iris-setosa	Iris-versicolor
Iris-setosa	0	10
Iris-versicolor	10	0

For this dataset I got the same results

5 MLP

As we need to minimize the error function I will take the gradient for it to find the final parameters and then I will compare it with the solution we have from class.

function: $1/2\sum_{mi=1}(y_{(i)} - \hat{y}_{(i)})^2$

$\partial E/\partial v = \partial E/\partial yh * \partial yh/\partial v$
$= \Sigma (yh - y) * yh * (1 - yh) z$
$v = v - \mu \Sigma (yh - y) * yh * (1 - yh) z$

$\partial E/\partial w = \partial E/\partial yh * \partial yh/\partial z * \partial z/\partial w$
$= \Sigma (yh - y) * yh * (1 - yh) v * z (1-z) * x$
$w = w- \mu \Sigma (yh - y) * yh * (1 - yh) v * z (1-z) * x$

Conclusion: The parameters we have are pretty much closer to the ones we had in class except that the Derivative for yhat was different as we have sigmoid activation on the output

6 MLP 3-class (3-class NN.R)

yhat	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-versicolor	10	20	10

I got here a bad accuracy %50