Binary classification using logistic regression

Report by: 완(2019007901)

Running Environment

- OS: Windows 10 Home
- CPU: AMD Ryzen 7 5800H
- Language: Python 3.10

Introduction

In this program, we will train a model for binary classification using logistic regression where vector x = {x1,x2} will be inputted into the model. Before the training starts, we prepare 10000 training samples and 1000 test samples. The x1 and and x2 value will be randomized and this program will try to correctly classify whether the values of x1+x2 in the samples are positive or not. If positive, the output y will be labelled 1, else 0. Finally, we will find the accuracy of our model if the predicted results are correct or not.

The model is defined by y = sigma(Wx + b) where W and b are unknown parameters. Therefore, the goal of this training model is to accurately estimate the best value for W and b so that the output of the model gives the best prediction accuracy of y=1.

Estimated unknown function W & b

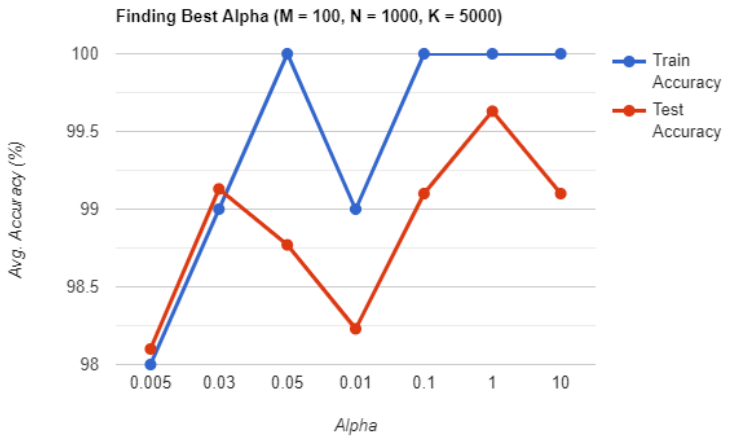
For our experiments, we will set the number of samples according to the assignment specifications (M=10000, N = 1000) and the number of iterations K will be set to 5000. After experimenting a few times with the model, we reached a conclusion where for every iteration, the values of W and b are increasing. The figure below shows the estimated function parameters W and b for this training.

```
[w1, w2, b] = [0.7465413269425484, 0.47627504611179516, 0.3657812594396684]
Cost: -0.000004
[w1, w2, b] = [3.1199126000075847, 3.1166957450165627, -0.03810104615177938]
Cost: -0.000000
[w1, w2, b] = [3.939773276927947, 3.936301642037849, -0.037567179431751536]
Cost: -0.000000
[w1, w2, b] = [4.516694421028623, 4.51329932009851, -0.03659352827102632]
Cost: 0.000000
[w1, w2, b] = [4.976146069877465, 4.972938627300812, -0.03582276990783209]
Cost: 0.000000
[w1, w2, b] = [5.363956119296287, 5.360983969946745, -0.03508614523443932]
Cost: 0.000000
[w1, w2, b] = [5.702690759970888, 5.699975977034431, -0.03432372842787988]
Cost: 0.000000
[w1, w2, b] = [6.005334286184348, 6.002886731100759, -0.03352081587964318]
Cost: 0.000000
[w1, w2, b] = [6.280125216242289, 6.277948293768151, -0.03267837460956596]
Cost: 0.000000
[w1, w2, b] = [6.532656911145071, 6.53075039039616, -0.03180232512319927]
Cost: 0.000000
Cost with n test samples =  -0.005252685070369487
Accuracy for 'm' train samples: 99.55000000000001%
Accuracy for 'n' test samples: 99.5%
```

Empirically determined (best) hyper parameter, α

The learning rate, α, is known as hyperparameter which affects greatly on the accuracy of our model. In this assignment, we also tasked to find the best α where the model could yield the highest

accuracy on average. For this, we will run the training on different α and observe which one yields the highest average accuracy on the training and test samples. The graph below shows the result.



From the graph, we can conclude that when α=1, it gives the best accuracy for both train and test samples. More efficient way that can determine the best α value is by looping the whole training, search the α value by binary search, and when the accuracy hits 100%, break the loop and the best α can be obtained.

Accuracy

	m=10, n=1000, K=5000	m=100, n=1000, K=5000	m=10000, n=1000, K=5000
Accuracy ('m' train samples)	100.0%	100.0%	99.55
Accuracy ('n' test samples)	98.5%	99.7%	99.5

	m=10000, n=1000, K=10	m=10000, n=1000, K=100	m=1000, n=1000, K=5000
Accuracy ('m' train set)	95.09%	99.02%	99.7%
Accuracy ('n' test samples)	94.0%	98.5%	99.8%

Discussion

From the accuracy table above, we can see that when the value of m changes and the value of K is fixed, the bigger the value of m, the difference between the accuracy of train samples and test samples become smaller. We can conclude from this that, the greater number of training samples given to the model, the more it will be fitted to the test sample, therefore increasing the probability of the accuracy being correct. We also can observe from the second table where the greater the value of K becomes, the accuracy of both training and test samples also increase. From this result, we can learn that the more the number of times our model learns (K), we can increase the accuracy of the prediction of the output. We also learn that by finding the best hyperparameter, α, which highly influences our model performance, we can further increase the model's accuracy. In a nutshell, by providing the best value of W, b, α and increasing the iteration number, the best model can be obtained.