Osman Batuhan Şahin
osmanbatuhansahin@mu.edu.tr

Sunday 29th November, 2020

1 Introduction

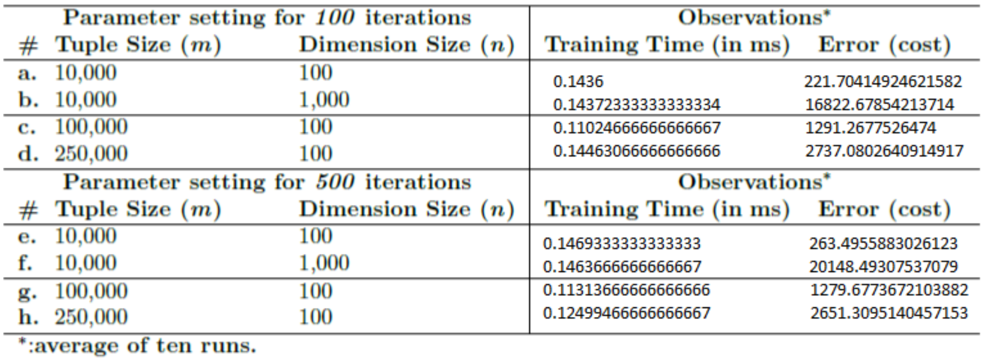
In this assignment, i made practice on applications of neural networks with using classification. Also i analyzed the effects of hyperparameters given to construct a network structure. I used sklearn numpy and matplotlib libraries in python. This assignment was important because I saw proof of some theoric information that I learned in lessons.

2 Single-layer Perceptron

2.1 Classification task

In this section I defined a function called task2-1(). It takes parameters tuple and dimension and this parameters become n-samples and n-features of make-classification dataset from sklearn. I selected this dataset because task was binary-class classification. I split the data for train and test. This function returns me time and error of applying single-layer perceptron with Stochastic Gradient Descent (SGD) solver. I got result 10 times and take average of it for consistency.

Table 1: The effectiveness and efficacy of single-layer perceptron with respect to the varying parameter settings.



Parameter setting for *100* iterations			Observations*	
#	Tuple Size (m)	Dimension Size (n)	Training Time (in ms)	Error (cost)
a.	10,000	100	0.1436	221.70414924621582
b.	10,000	1,000	0.14372333333333334	16822.67854213714
c.	100,000	100	0.11024666666666667	1291.2677526474
d.	250,000	100	0.14463066666666666	2737.0802640914917
Parameter setting for *500* iterations			Observations*	
#	Tuple Size (m)	Dimension Size (n)	Training Time (in ms)	Error (cost)
e.	10,000	100	0.1469333333333333	263.4955883026123
f.	10,000	1,000	0.1463666666666667	20148.49307537079
g.	100,000	100	0.11313666666666666	1279.6773672103882
h.	250,000	100	0.12499466666666667	2651.3095140457153

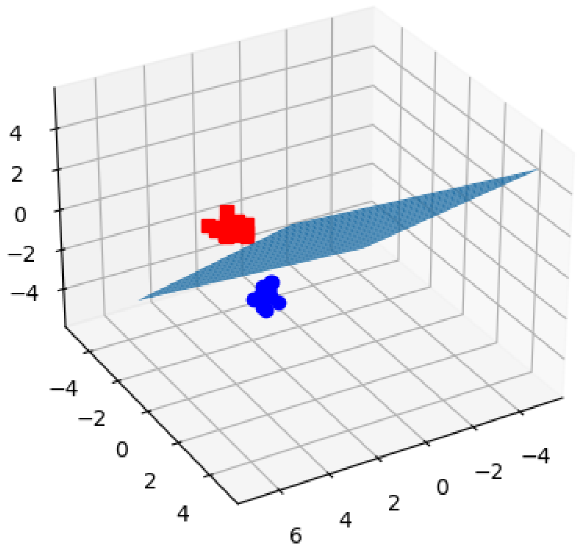
*:average of ten runs.

Result

I ignored warnings because pycharm gives convergence warning. Output is above. If we compare dimension sizes 1000 dimension size gives more error than 100.There is a big difference. But raining time did not effected with dimension size change. If we compare iteration change there is no change in training times but error value is better in low iterations generally. When tuple size increases, error value become bigger we can see. Training time is best with 100,000 tuples, it is interesting. I expected less tuple means less time. Finally, more dimension size and iterations increases error.

2.2 Visualization of decision boundary

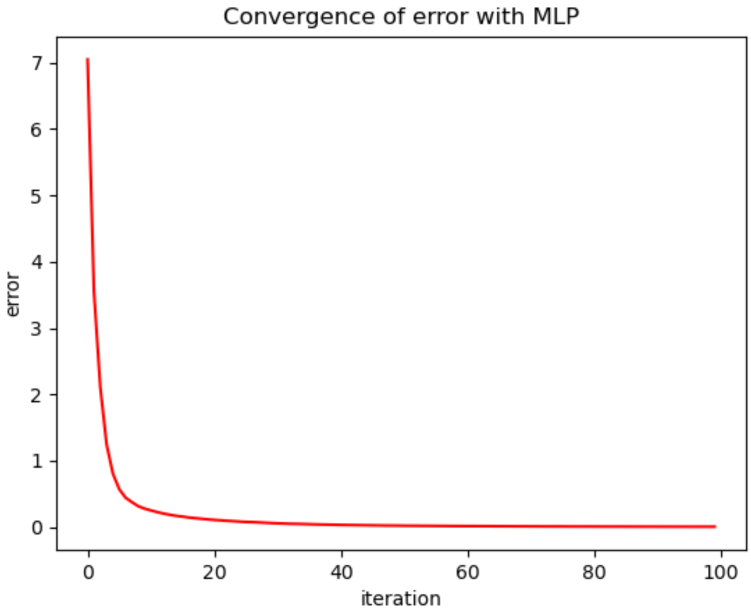
In this section I loaded iris dataset. I splitted as train and test and applied single-layer perceptron network to fit a model on iris dataset. Then I plotted test and prediction values to a 3D surface.



3 Multi-layer Perceptron

3.1 Error convergence with multi-layer perceptron

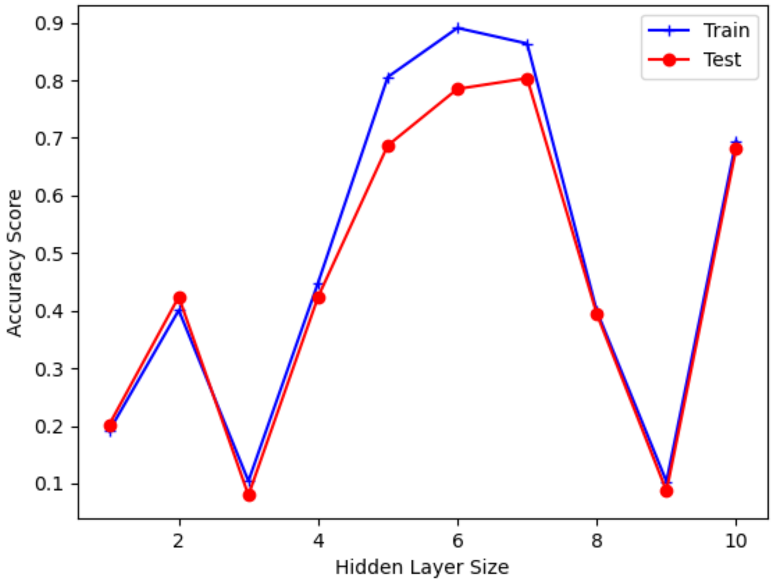
In this task I selected load-digits from sklearn dataset and I split the dataset, gave 0.7 for train. I run MLPClassifier on dataset with hidden-layer-sizes=(50,) and max-iter=100. I found losscurve for determine error it makes my work easier. It was linear regression on first task. Then i plotted error and iteration.



If we look at figure above, error and iterations inverse proportion.

3.2 Effects of multi-layet perceptron structure on train test scores.

In this task I split digits dataset into train and test. I applied multi-layer perceptron network with H hidden layer size to handle dataset. I take accuracy scores with score and in task3-2() function. I gave parameters, number of neurons in hidden layers to function as bases of two and stored returns in two arrays.



We can see where are best and worst accuracy scores in figure above.

4 Conclusion

In conclusion, I practised applications of neural networks visualization. I analyzed on different parameters, tuple, dimension iteration sizes. I worked with single and multi layer perceptrons. I practised my sklearn numpy matplotlib skills but I should learn them better.