WEEK 3 ASSIGNMENT CHRONIS SAKOS

We adapted the Example_KaggleHiggs2.py in order to explore the Kaggle H ~>τ+τ- script. This script will solve a 2-output target classification problem based on input data from the Kaggle dataset. At first, we used the smaller training data set.

Initial Script

The given script had set parameters for learning rate 0.001, keep probability 0.9, 100 epochs and lambda parameter equal to 2. The value for the train and the test accuracy after running the initial script was 0.629 and 0.617 respectively.

Learning Rate

Then, we modified the learning rate and obtained the following results:

LEARNING RATE	TRAIN ACCURACY	TEST ACCURACY
0.001	0.629	0.617
0.01	0.650	0.644
0.1	0.601	0.619

We can see that the best value of accuracy is obtained when we set the learning rate parameter to 0.01 and from now on, we keep this value for the evaluation.

Dropout

Moreover, it is commonly known that the different value for learning rate can alter the influence of the training epochs. That is the reason why we explored the optimal dropout keep probability concerning the number of training epochs. So, we changed the keep probability percentage in order to evaluate the influence of dropout the accuracy of the samples. As we can see in the results below, the maximum value of accuracy is achieved for a keep probability of 70%, if we run 100 epochs.

KEEP PROBABILITY	TRAIN ACCURACY	TEST ACCURACY
90%	0.654	0.652
70%	0.698	0.692
50%	0.652	0.655
30%	0.638	0.634

However, for 300 training epochs, we can get higher accuracy for every dropout percentage. The ideal keep probability is 60%, with train and test accuracy equal to 0.739 and 0.732 respectively.

Weight Regularization

Then, we examined different values for the lambda parameter, namely the weight regularization parameter.

For a value of 0.2 we get a train accuracy of 0.809, which is higher than any other value of lambda, but the test accuracy value is not that big, which is a sign of overtraining.

The most ideal value for the parameter seems to be 20, as the train and test accuracy is high enough and the gap between the two is much smaller in comparison with the other values.

LAMBDA	TRAIN ACCURACY	TEST ACCURACY
0.2	0.809	0.718
2	0.727	0.724
20	0.749	0.725

Hidden Layers

We also evaluated how the addition of one and two more hidden layers with different number of nodes affected the accuracy. However, neither case was able to increase the accuracy of the network. Any effect had neither the increase nor the decrease of the last hidden layer's number of nodes to the performance, as well.

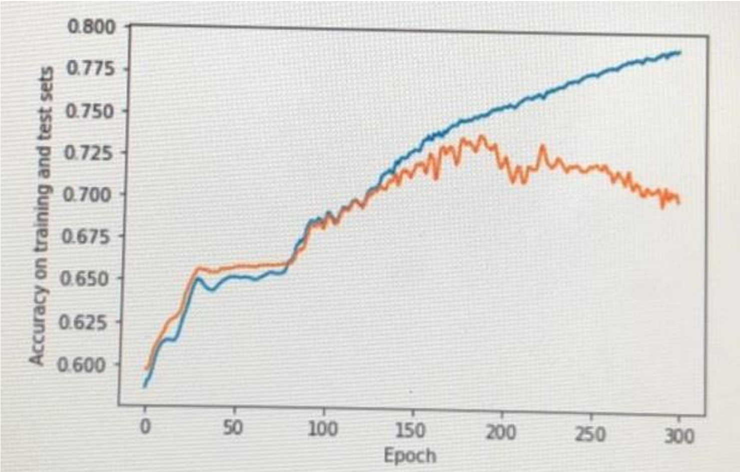
Complete Dataset

After examining the small dataset (train_sml_sig.csv and train_sml_bg.csv), we considered training the complete dataset (train_sig.csv and train_bg.csv), whilst we increased the training data to 20.000. We set the learning rate equal to 0.02, the training epochs equal to 300, the keep probability equal to 60% and the lambda parameter equal to 20. The numerical results are as following:

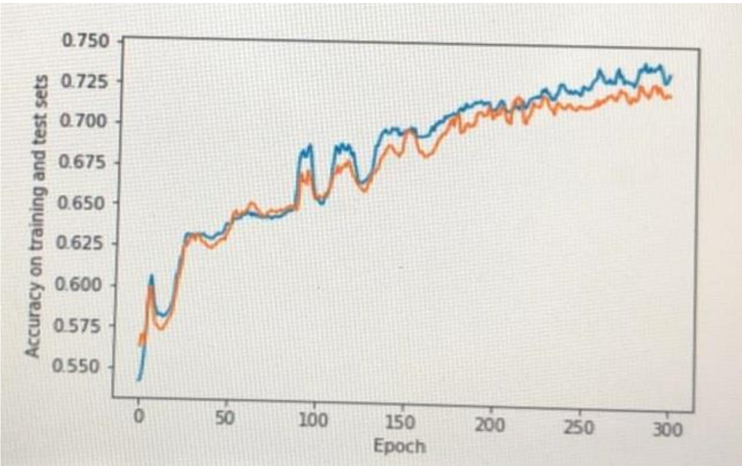
Train Accuracy = 0.760
Test Accuracy = 0.698

The network seems to be over trained, as the accuracy of the training set keeps growing while the accuracy of the test set stays almost stable.

After training both the small and the complete dataset, we conclude that the small dataset leads to higher accuracy and is not prone to overtraining as the complete dataset.



Complete dataset



Small dataset

To conclude, the ideal values for the parameters of this problem seem to be the following:

1.Use of the small dataset for training
2.Learning rate = 0.01
3.Training epochs = 300
4.Dropout keep probability = 60%
5.Lambda parameter = 20
6.Number of data = 2000