Neural Networks(CSCI-835)

Assignment I Report

By

Dawit Beshah (1144901

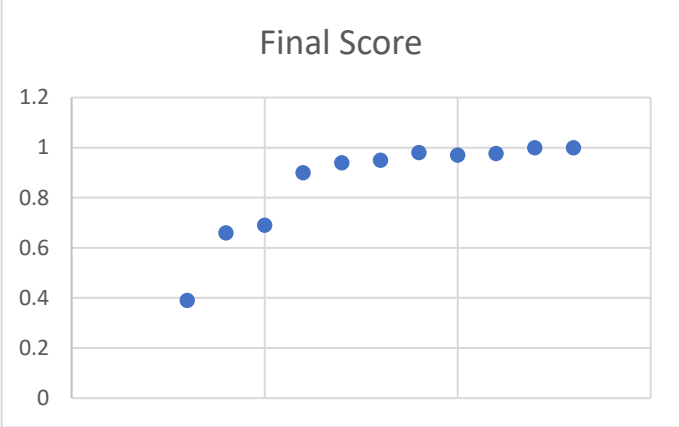
March 10, 2017

Experiment one: **Different Neural Network Architecture**

Approach: starting from low number of the hidden layer neurons that of [2, 4, 1] and keep increasing the number neurons proportionally and look up the effect on different metrics like accuracy (final score), Log loss, Execution time, number of misclassifications and cross validation.

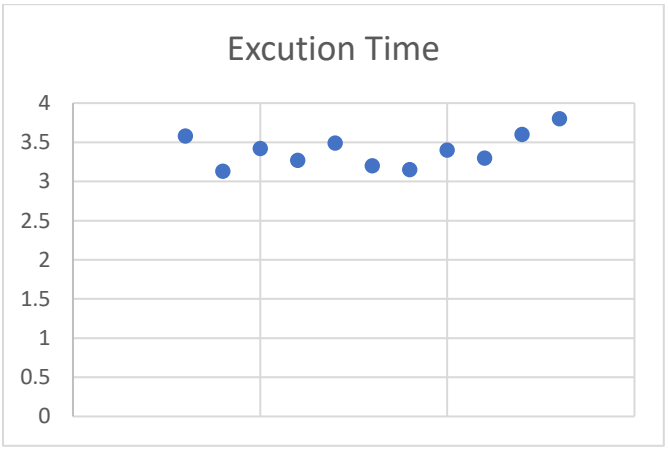
Observation:

# of neurons at the hidden layer Vs Accuracy



Hidden Layer Configuration	Final Score
[2, 4, 1]	0.39
[4, 8, 2]	0.66
[6, 12, 3]	0.69
[8, 16, 4]	0.9
[10, 20, 5]	0.94
[12, 24, 6]	0.95
[14, 28, 7]	0.98
[16, 32, 8]	0.97
[18, 36, 9]	0.977
[20, 40, 10]	1
[40, 80, 20]	1

# of neurons at the hidden layer Vs Execution time

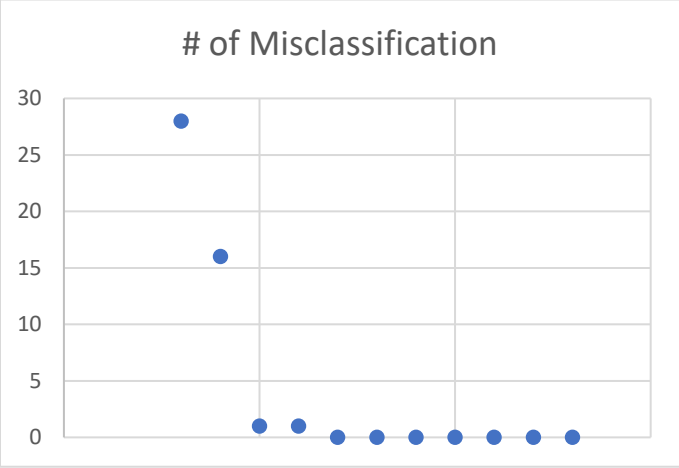


Hidden Layer Configuration	Execution Time
[2, 4, 1]	3.58
[4, 8, 2]	3.13
[6, 12, 3]	3.42
[8, 16, 4]	3.27
[10, 20, 5]	3.49
[12, 24, 6]	3.2
[14, 28, 7]	3.15
[16, 32, 8]	3.4
[18, 36, 9]	3.3
[20, 40, 10]	3.6
[40, 80, 20]	3.8

# of neurons at the hidden layer Vs Log Loss

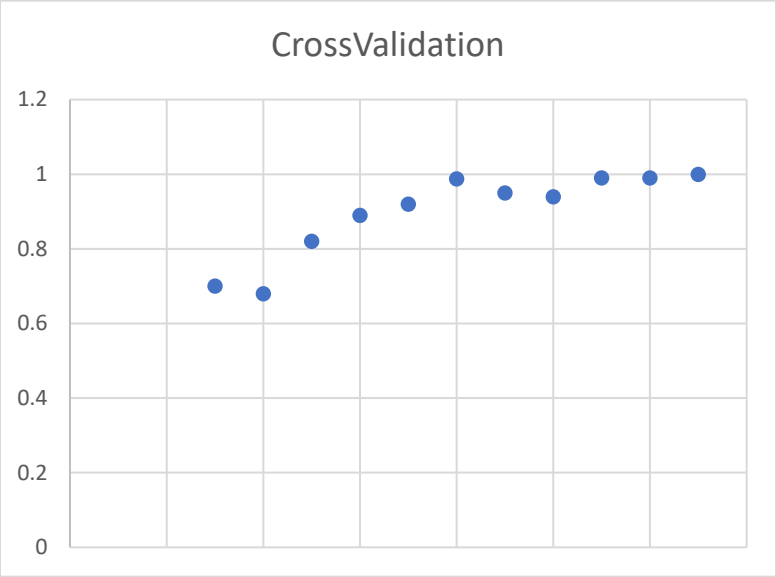
Hidden Layer Configuration	Log Loss
[2, 4, 1]	1.09
[4, 8, 2]	0.88
[6, 12, 3]	0.75
[8, 16, 4]	0.05
[10, 20, 5]	2.02x10^-6
[12, 24, 6]	1.1X10^-6
[14, 28, 7]	1.1X10^-6
[16, 32, 8]	1.9x10^-6
[18, 36, 9]	2.2X10^-6
[20, 40, 10]	1.5X10^-6
[40, 80, 20]	1.7X10^-6

# of neurons at the hidden layer Vs # of Misclassification



Hidden Layer Configuration	# of Misclassification
[2, 4, 1]	28
[4, 8, 2]	16
[6, 12, 3]	1
[8, 16, 4]	1
[10, 20, 5]	0
[12, 24, 6]	0
[14, 28, 7]	0
[16, 32, 8]	0
[18, 36, 9]	0
[20, 40, 10]	0
[40, 80, 20]	0

# of neurons at the hidden layer Vs cross validation



Hidden Layer Configuration	Cross Validation
[2, 4, 1]	0.7
[4, 8, 2]	0.68
[6, 12, 3]	0.82
[8, 16, 4]	0.89
[10, 20, 5]	0.92
[12, 24, 6]	0.988
[14, 28, 7]	0.95
[16, 32, 8]	0.94
[18, 36, 9]	0.99
[20, 40, 10]	0.99
[40, 80, 20]	1

Experiment Two: Optimizers

Approach: applying number of optimizer to check their effect on the classification result. By keep the neural network configuration constant (i.e with hidden layer [12, 24, 6] observe the effect of applying the optimizers. And by tuning the values of the parameters that used by the optimizer specially the learning rate overlook the significance.

Finding:

[the attached Excel file]

Outlier Confusion Matrixes

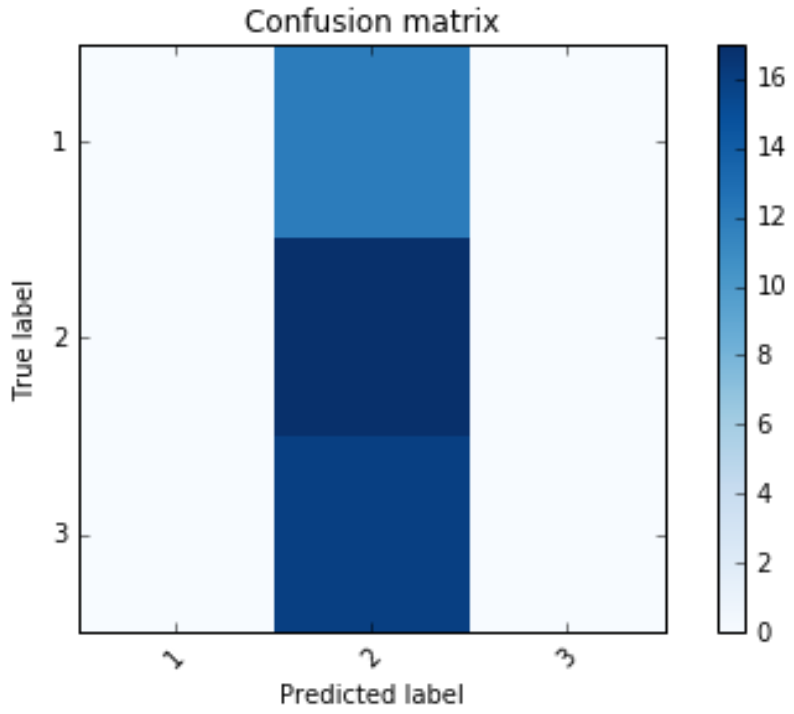


Figure 1: Low number of neuron hidden layer Network ([2, 4, 1])

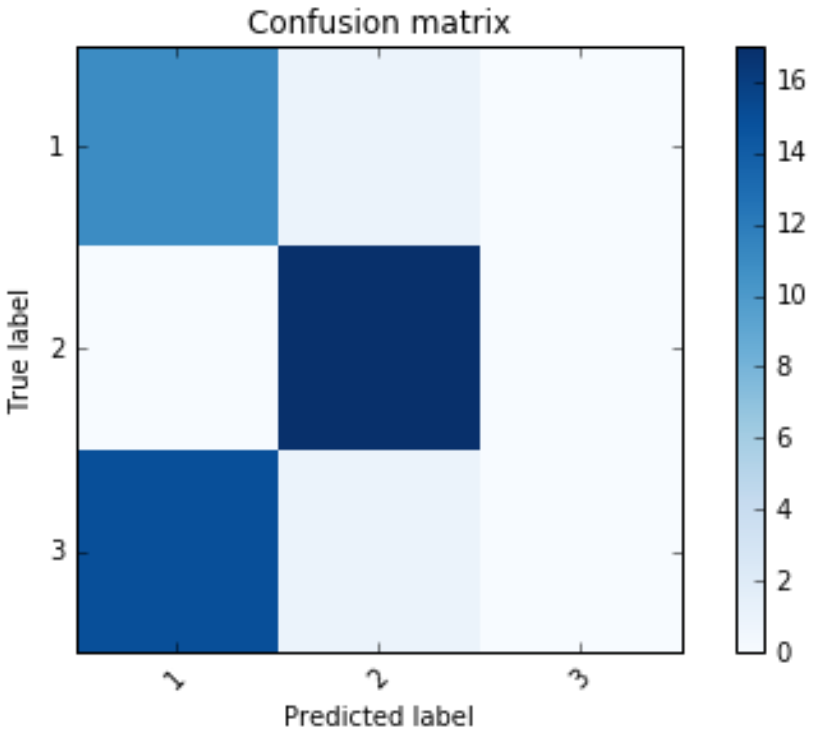


Figure 2: [12, 24, 6] hidden layer network with Ftrl Optimizer (learning rate=0.01)

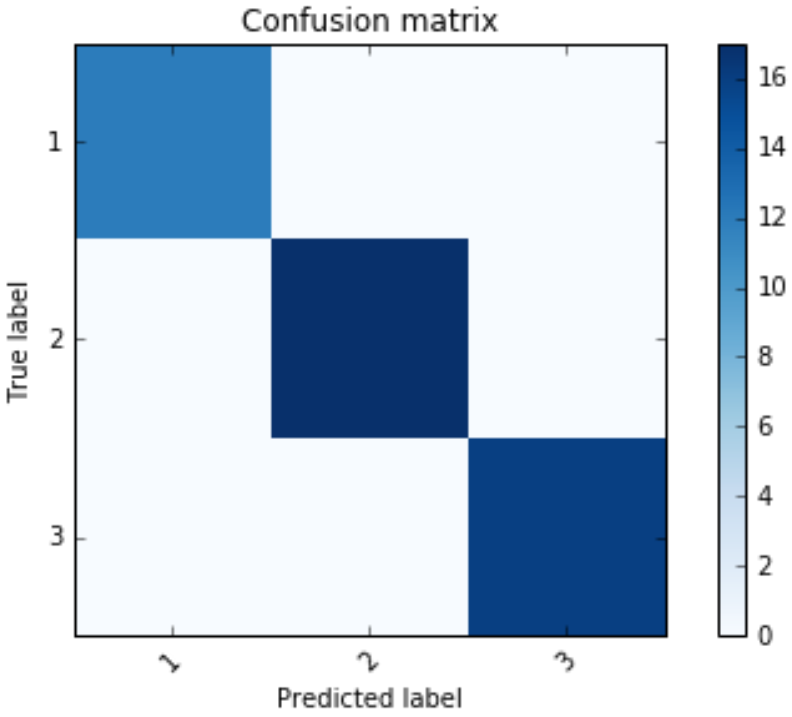


Figure 3: one of the perfect classification