EE496 Spring 2016-2017
Homework 1

Training Multilayer Perceptron

April 2017

Middle East Technical University

Electrical and Electronics Engineering Department

JeanPiere Demir

Contents

1 Introduction

2 Dataset Description

Table 1: The description of the dataset

Class No	Motion	Training set	Validation set	Test set
1	**open**	30	13	8
2	**close**	28	15	16
3	**flexion**	24	16	13
4	**extension**	34	11	13
5	**ulnar d.**	24	17	18
6	**radial d.**	30	12	10
7	**pronation**	26	16	14
8	**supination**	28	12	19
TOTAL		**224**	**112**	**111**

In order to control a robotic prosthetic hand, Prosthetic Hand Controllers (PHCs) gets the signals generated by straining muscles in the forearm. We can, applying supervised learning, get a PHC to learn which hand movement the user wants to perform. There are given a data set consisting of three days of electromyographic (EMG) signals from a robotic prosthetic hand controller with the object to learn eight hand motions.

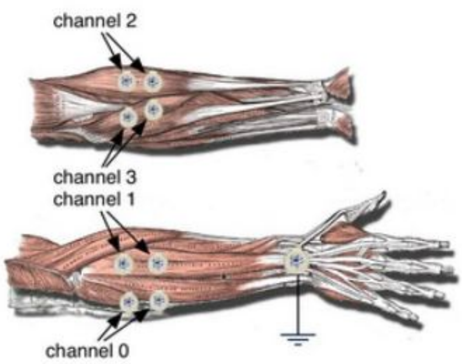

Figure 1: Sensor placement

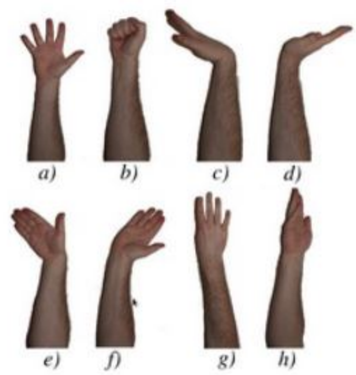

Figure 2: Motions : a) open, b) close, c) flexion, d) extension, e) ulnar deviation, f) radial deviation, g) pronation and h) supination

Figure 1: Places of the sensors

3 Results

3.1 Effects of Learning Rate and Hidden Layer

Table 2: Training Time

Learning Rate (1x10 $^{(-k)}$)	Number of Hidden Units				
	2	**4**	**8**	**16**	**32**
k=-4	0.250726	0.261960	0.274289	0.334180	0.338203
k=-3	0.268083	0.259496	0.278491	0.337856	0.303577
k=-2	0.282417	0.268865	0.279492	0.301628	0.333928
k=-1	0.287047	0.278920	0.271259	0.357787	0.331774
k=0	0.285164	0.275723	0.289950	0.348683	0.349340

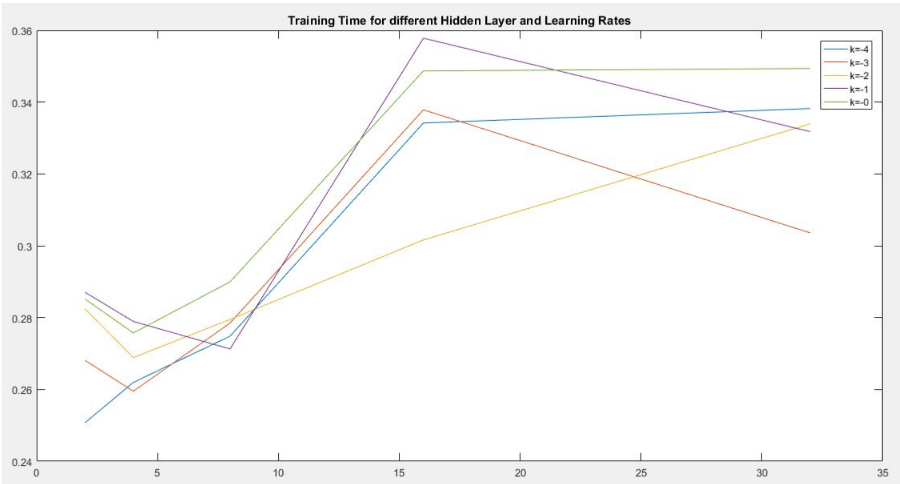

Figure 2: Training Time for different Hidden Layer and Learning Rate values

As we see from the figure , when our hidden layer numbers increase our training time will increase due to increase on computational process. However , when we increase learning rate , our system will jump more than before so computational process will decrease , but that depends on system structure.

Table 3: Accuracy for Training Set

Learning Rate (1x10 $^(-k)$)	**Number of Hidden Units**				
	2	**4**	**8**	**16**	**32**
k=-4	77.6786	76.4509	78.9063	76.3393	78.2366
k=-3	78.2366	77.0089	78.2366	77.5670	78.3482
k=-2	78.2366	77.6786	80.6920	81.4732	78.6830
k=-1	81.0268	82.8125	84.3750	95.2009	95.6473
k=0	84.0402	93.1920	97.6563	98.7723	97.8795

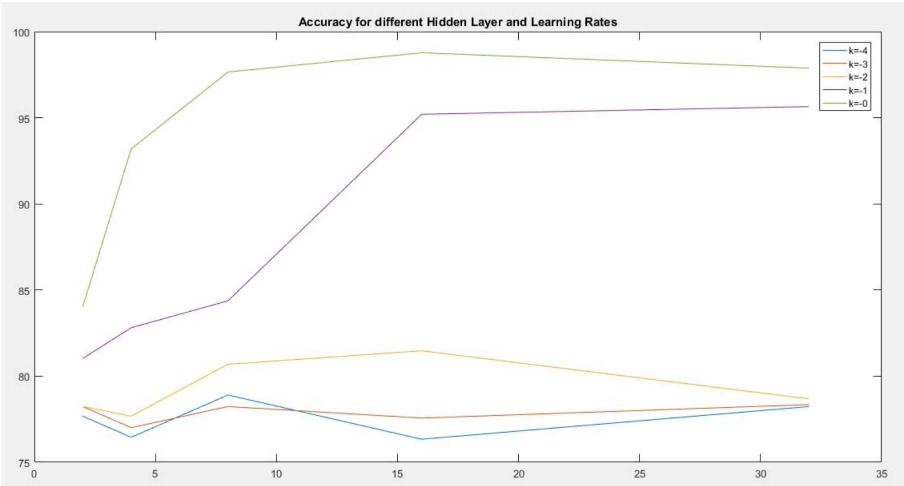

Figure 3: Accuracy on Training Set for different Hidden Layer and Learning Rate values

When our learning rate increase , our accuracy will increase due to change on the weights have become much more faster . So our accuracy improved. Also , when we increase the hidden layer there is not any linear improvement on accuracy due to we are working in limited epochs number.

Table 4: Accuracy for Validation Set

Learning Rate (1x10 $^{(} - k)$)	**Number of Hidden Units**				
	2	**4**	**8**	**16**	**32**
k=-4	77.6786	78.3482	80.5804	75	77.4554
k=-3	77.063	79.6875	79.2411	79.2411	78.5714
k=-2	77.9018	77.6786	77.2321	76.3393	76.7857
k=-1	78.5714	82.3661	86.3839	92.4107	95.0893
k=0	82.3661	91.9643	95.5357	97.0982	94.1071

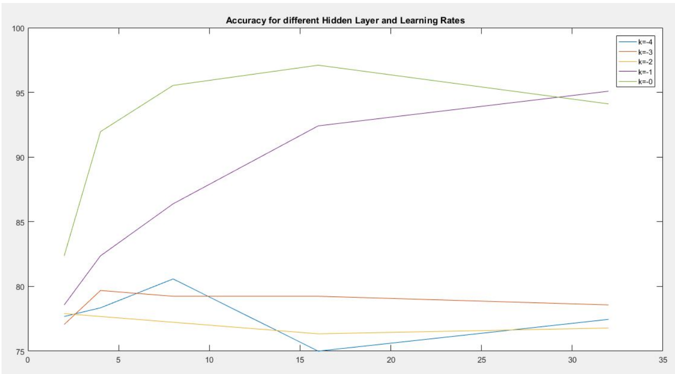

Figure 4: Accuracy on Validation Set for different Hidden Layer and Learning Rate values

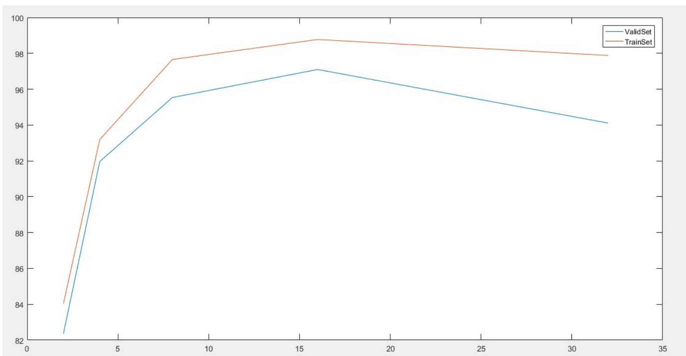

Figure 5: Accuracy comparison Validation Set vs Training Set

The accuracy improvement is same as training set accuracy. However ,the values of accuracies are not same due to the system trained with Training set .That show us however much we design and create a neural network the best accuracy can be obtained with Training Set.

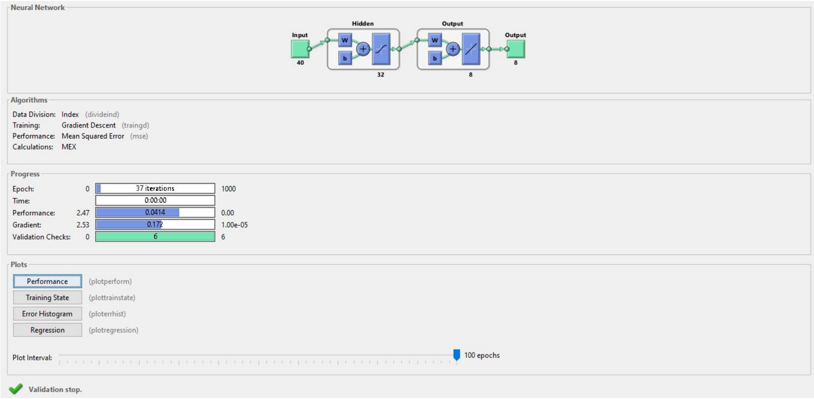

Figure 6: NNtool toolbox to show Properties

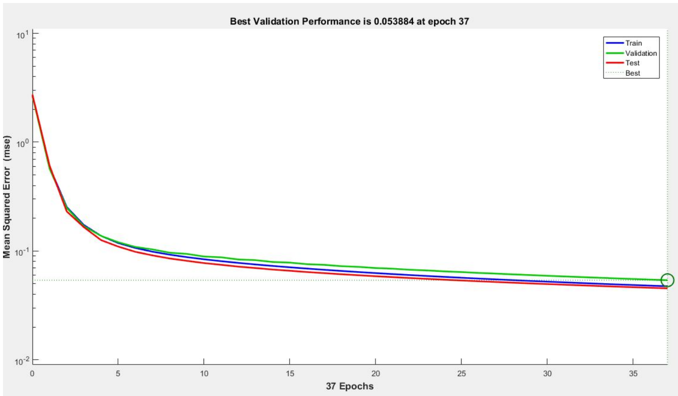

Figure 7: Epoch = 37 , Where error tendsto increase

As we see from the figures , after a while the epoch number is not reached to its maximum value. Also , Mean Square Error (MSE) value tends to increase. These phenomenon called Overfitting. To clarify, in Overfitting mode the system start to memorize the Train set and for a neural network that is very bad thing. To prevent that phenomenon we give Test Set to the system. The system test the system with Test Set while it enters to next epochs. In our system , that epoch value was 37 where error tends to increase .Plus our accuracy just before that epoch is 97.911.

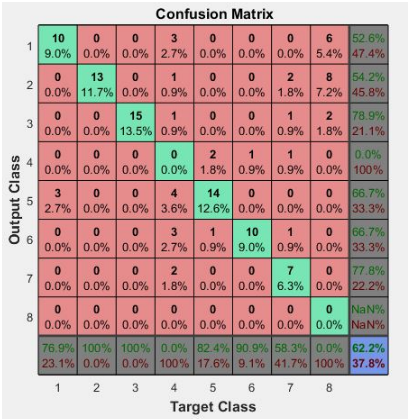

Figure 8: Confusion matrix for Test Set without Momentum Term HL = 32 , Learning rate = 1

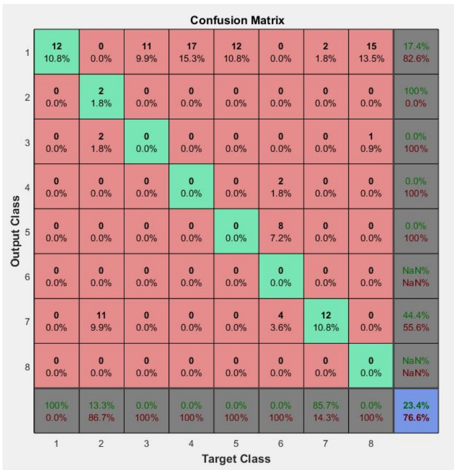

Figure 9: Confusion matrix for Test Set with Momentum Term HL = 32 , Learning rate = 1

As we see from the figures, when we add momentum term to the system our confusion value is decreasing because we add the effect of past values to the system. To be more spesific about monmentum term, momentum term is the attenuation of oscillations in the gradient descent. The geometric idea behind this idea can probably best be understood in terms of an eigenspace analysis in the linear case. If the ratio between lowest and largest eigenvalue is large then performing a gradient descent is slow even if the learning rate large due to the conditioning of the matrix. The momentum introduces some balancing in the update between the eigenvectors associated to lower and larger eigenvalues.

4 Conclusion

In that project we try to train a Prosthetic Hand Controllers (PHCs) with Multi Layer Perceptron (MLP) . MLP is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training the network .

In our dataset there was some correlations between training set and validation set. So we do not see too much overfitting problem . We see that while hidden layer number increased due to increase on process amount our training time decreased. Also , the best accuracy can be obtained by training and testing system with Train set. We try to decrease that effect so we use different topologies to overcome that problem. One of the solution is momentum term . Momentum Term, carry the information of past training data to future so our system can take lessons from its past data .

To sum up , in that project we learned how to use and construct MLP and apply to a problem.