**EE 496**
**INTRODUCTION TO COMPUTATIONAL INTELLIGENCE**

**HOMEWORK**
**#1**

# Training Multilayer Perceptron


**Ayberk Aydın**

**1875699**

# I. INTRODUCTION

In this homework, we are going to train a multilayer perceptron with a dataset consisting of three days of electromyography (EMG) signals from a robotic prosthetic hand controller with the object to learn eight hand motions and observe the effects of different parameters such as hidden layer size and learning rate.

# II. DATASET DESCRIPTION

The main dataset is divided to 3 subsets, which are training set (50%), validation set (25%) and test set (%25). The number of samples in each class in each subset is shown in Table 1.

*Table 1: Number of samples in each class (Input vector size: 447x40, Output vector size: 447x8)*

| Class No | Motion | Training Set | Validation Set | Test Set |
|---|---|---|---|---|
| 1 | Open | 24 | 17 | 10 |
| 2 | Close | 32 | 15 | 12 |
| 3 | Flexion | 32 | 11 | 10 |
| 4 | Extension | 33 | 16 | 9 |
| 5 | Ulnar d. | 28 | 12 | 19 |
| 6 | Radial d. | 25 | 13 | 14 |
| 7 | Pronation | 20 | 15 | 21 |
| 8 | Supination | 31 | 12 | 16 |
| TOTAL | **447** | **223** | **112** | **112** |

# III. EXPERIMENTAL RESULTS

The experiments are done with the obtained and classified data with different parameters. The results are show in the figures and tables below.

*Table 2: Training time (sec) for 100 epochs on a computer with processor Intel Core i5-5200U @2.20GHz*

| Learning Rate(10^-k) | Number of hidden units | | | | |
|---|---|---|---|---|---|
| | 2 | 4 | 8 | 16 | 32 |
| k=-4 | 0.1900 | 0.2200 | 0.1900 | 0.1900 | 0.2700 |
| k=-3 | 0.1700 | 0.1700 | 0.1800 | 0.2000 | 0.2300 |
| k=-2 | 0.1800 | 0.1900 | 0.1700 | 0.1900 | 0.2400 |
| k=-1 | 0.1700 | 0.1800 | 0.1800 | 0.1800 | 0.2300 |
| k=0 | 0.1700 | 0.1700 | 0.1800 | 0.1900 | 0.2400 |

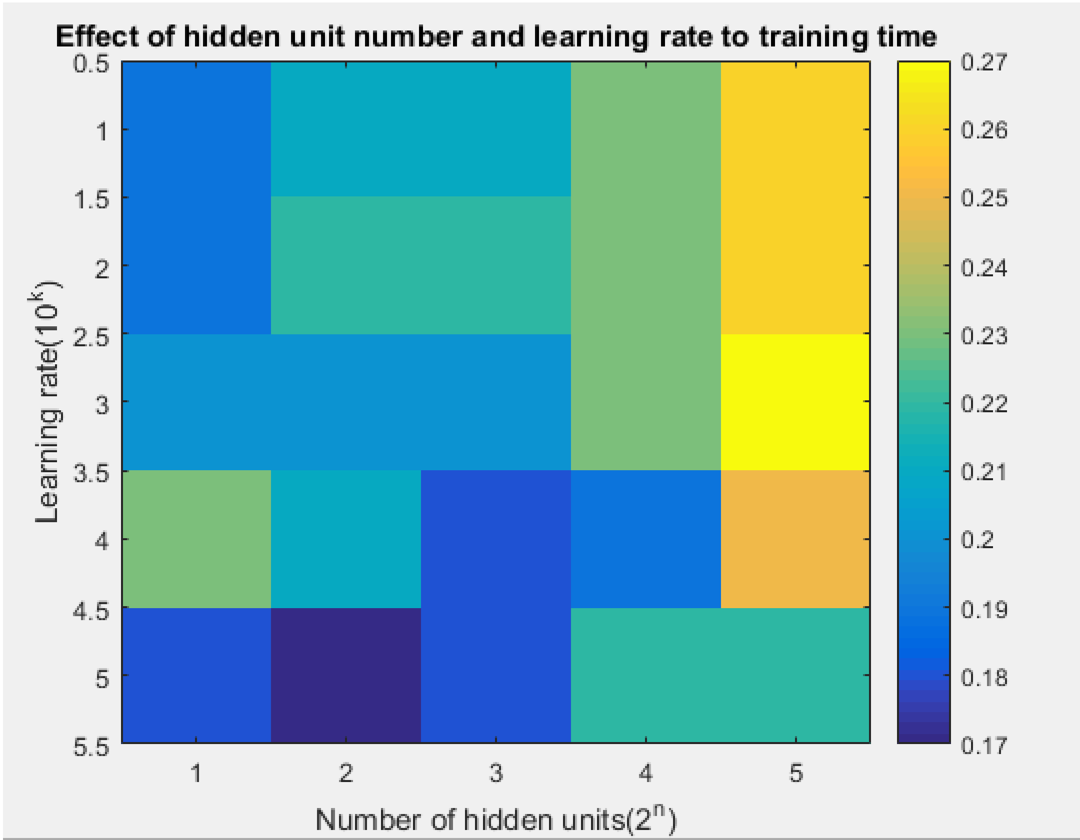


*Figure 1: Effect of number of hidden units and learning rate on training time (sec)*

The learning rate seems to have no effect on training time, however the number of hidden units increases the training time, which makes sense because the more neurons means more weights to change and also more derivatives to process by computer.

*Table 3: Accuracy (%) on the training set (100 epochs)*

| Learning Rate(10^-k) | Number of hidden units | | | | |
|---|---|---|---|---|---|
| | 2 | 4 | 8 | 16 | 32 |
| k=-4 | 14.80 | 19.28 | 17.94 | 4.930 | 15.70 |
| k=-3 | 11.21 | 21.97 | 7.170 | 18.83 | 8.070 |
| k=-3 | 14.35 | 11.66 | 27.35 | 14.80 | 17.94 |
| k=-1 | 13.45 | 37.22 | 68.61 | 77.13 | 86.55 |
| k=0 | 4.480 | 57.85 | 90.13 | 97.31 | 97.76 |

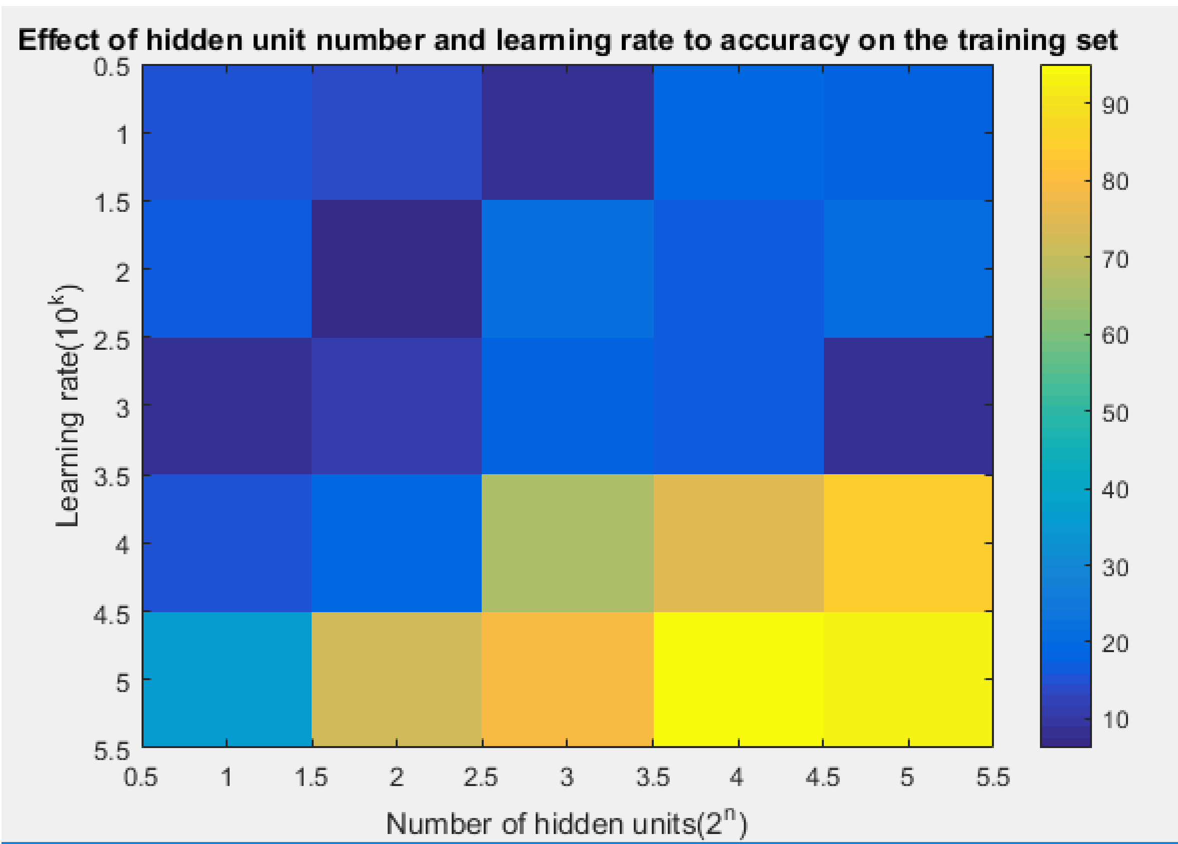


*Figure 2: Effect of number of hidden units and learning rate on training set accuracy (%)*

With increasing number of hidden units, accuracy on the training set increases. The increasing learning rate is also increases the accuracy. However, this does not mean it is always better to have more learning rate to obtain better results. Excessively high learning rate can make the training unstable, which we did not observe in our case. Also, making the hidden layer infinitely large does not also improve performance considerably but only increases the training time.

*Table 4: Accuracy (%) on validation set (100 epochs)*

| Learning Rate(10^-k) | Number of hidden units | | | | |
|---|---|---|---|---|---|
| | 2 | 4 | 8 | 16 | 32 |
| k=-4 | 10.71 | 27.68 | 20.54 | 8.930 | 15.18 |
| k=-3 | 13.39 | 20.54 | 5.360 | 17.86 | 18.75 |
| k=-3 | 12.50 | 16.96 | 25.89 | 13.39 | 17.86 |
| k=-1 | 6.250 | 36.61 | 60.71 | 80.36 | 79.46 |
| k=0 | 5.360 | 50.89 | 90.18 | 96.43 | 98.21 |

The best performance is obtained with the learning rate 1 and 32 hidden units.

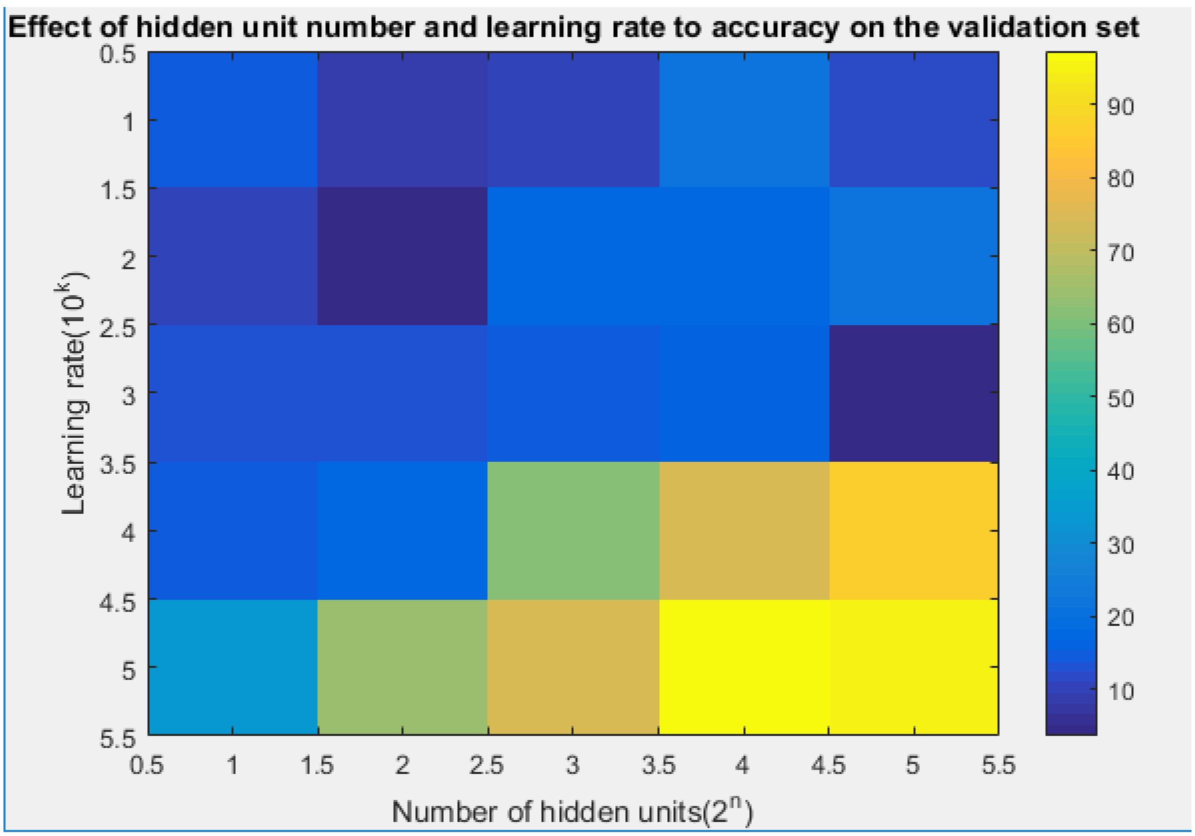


*Figure 3: Effect of number of hidden units and learning rate on validation set accuracy (%)*

Same trend with the training set also applies for the validation set. So, the same conclusions also hold for the validation set.

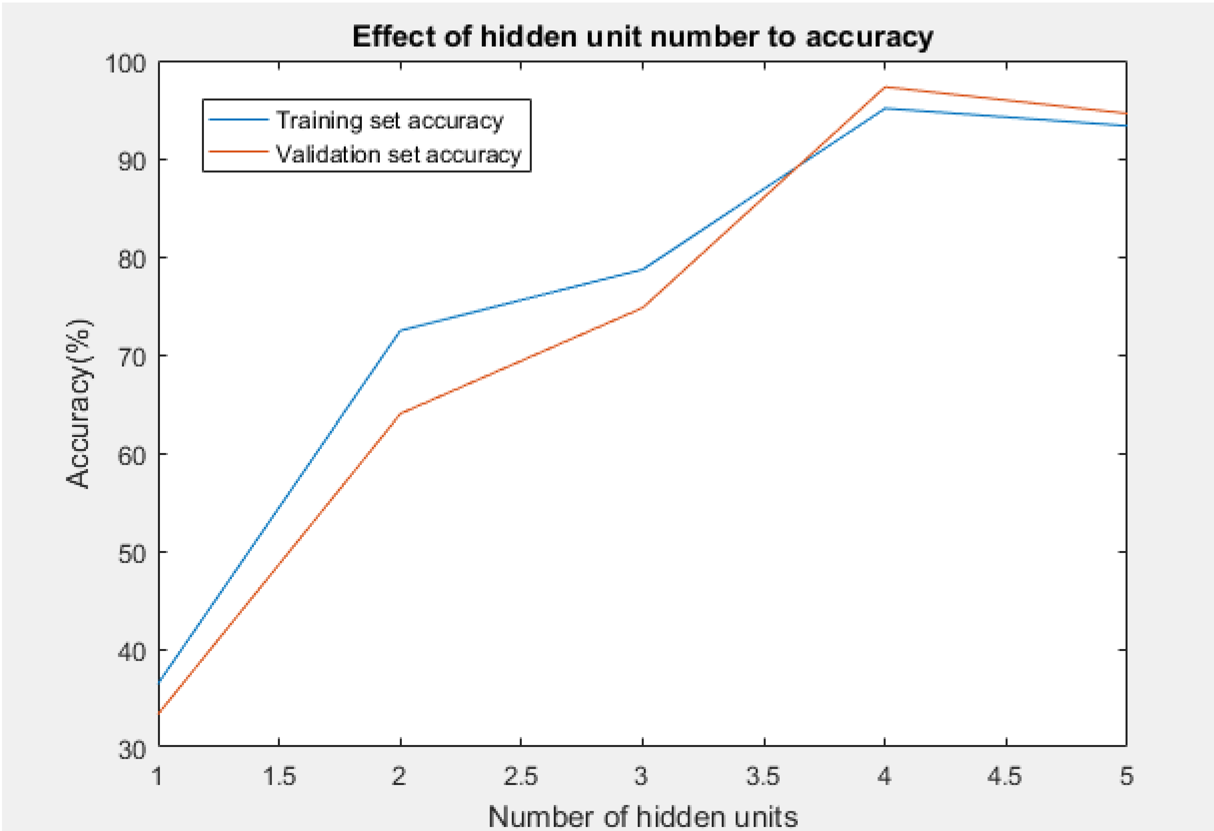


*Figure 4: Comparison of Accuracy (%) on train and validation sets (Learning rate=1, Number of epochs=100)*

There is a strong positive correlation between training set accuracy and validation set accuracy after training. The reason is the samples are classified randomly and the neural network is also giving similar outputs for similar inputs.

The best results are obtained with learning rate 1 and 32 hidden neurons. With these parameters, the maximum epoch limit is increased to 1000. Then a new neural network is generated and trained with these parameters. It stopped by validation check to prevent overfitting. The network is also tested with the test set. Figure 5 and Table 5 shows the confusion matrix of test set after the training.

*Table 5: Confusion matrix for test set (32 hidden units, 174 epochs, learning rate=1, no momentum)*

| | | | Predicted Class | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | TOTAL |
| Actual Class | 1 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 |
| | 2 | 0 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| | 3 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 10 |
| | 4 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 9 |
| | 5 | 0 | 0 | 0 | 1 | 18 | 0 | 0 | 0 | 19 |
| | 6 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 14 |
| | 7 | 0 | 0 | 0 | 0 | 0 | 1 | 20 | 0 | 21 |
| | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 16 |
| | TOTAL | 10 | 12 | 10 | 10 | 18 | 15 | 20 | 16 | 112 |

The test results are accurate in general, except for the two false classification. A pronation (7) is identified by radial d. (6) and an ulnar d. (5) is identified by extension (4). The other classes are identified perfectly. Total accuracy is 0.982.

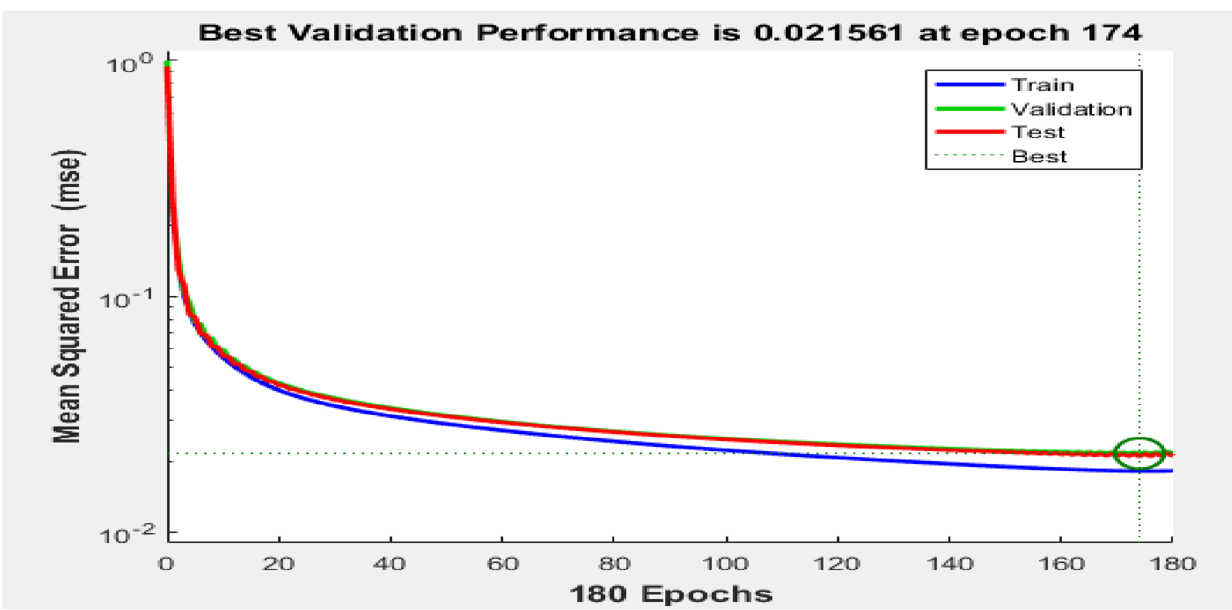


*Figure 5: Effect of Number of epochs on MSE (Learning rate=1, Number of hidden units=32, no moment)*

The law of diminishing returns is observed since with the decreasing cost, the magnitude of gradient also decreases. Also, the overfitting slowly starts to occur at the training, which can be observed from the small MSE oscillation at validation at the end. This is the reason why the training stopped before the max epoch number.

After that trial, a momentum term of 0.9 is added and the training is repeated. Figure 6 and Table 6 shows the results.

*Table 6: Confusion matrix for test set (32 hidden units, 1000 epochs, learning rate=1, Momentum=0.9)*

| | | | Predicted Class | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | TOTAL |
| Actual Class | 1 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 |
| | 2 | 0 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| | 3 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 10 |
| | 4 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 9 |
| | 5 | 0 | 0 | 0 | 0 | 19 | 0 | 0 | 0 | 19 |
| | 6 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 14 |
| | 7 | 0 | 0 | 0 | 0 | 0 | 1 | 20 | 0 | 21 |
| | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 16 |
| | TOTAL | 10 | 12 | 10 | 9 | 19 | 15 | 20 | 16 | 112 |

The test performance is increased slightly with a momentum term of 1. The misidentified sample from ulnar d. is no longer identified as extension, however the other misidentified pronation sample is still identified as radial d.

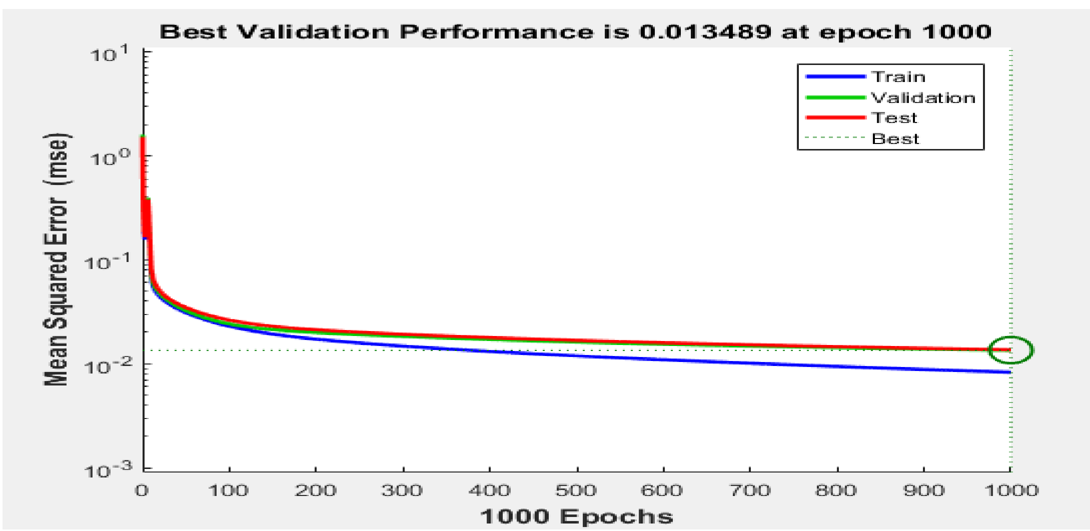


*Figure 6: Effect of Number of epochs on MSE (Learning rate=1, Number of hidden units=32, Moment=0.9)*

The mean square error performance is increased with the momentum term (from 0.02 to 0.01). Unlike the previous case with no momentum, training reached the maximum epoch in this case. However, this may be also caused from totally random starting point to gradient descent.

# IV. CONCLUSIONS

In this experiment, a multilayer feed forward neural network is trained with different parameters and the effect of these parameters to training process and training results are observed. For the training, a dataset consisting of electromyography (EMG) signals from robotic prosthetic hand controller is scaled between -1 and 1; and classified randomly as training, validation and test sets.

From these experiments, it is observed that the best learning rate is 1 amongst 0.0001, 0.001, 0.01, 0.1 and 1; and the best number of hidden layer units is 32 amongst 2, 4, 8, 16 and 32. Also, a momentum term is increased the performance further. Furthermore, effect of a validation set and overfitting as premature ending of a training is observed.

The results are impressive as a usage of neural networks. Even a small dataset is used, the neural network learned to classify the data with an accuracy of 98% in the matter of seconds.