Artificial Neural Networks and Deep Architectures-DD2437

Learning and generalization in feed-forward networks - from perceptron learning to backprop

Rithika Harish Kumar, Suhas Sheshadri

Aim

The aim is to learn the behavior of learning algorithm for single layer and multi-layer neural network. In the first part perceptron, delta and generalized delta rules are used to classify linearly separable and non-separable data. In the second part multi-layer perceptron's are used to address the problem of chaotic time series prediction. Generalization and function approximation is also performed.

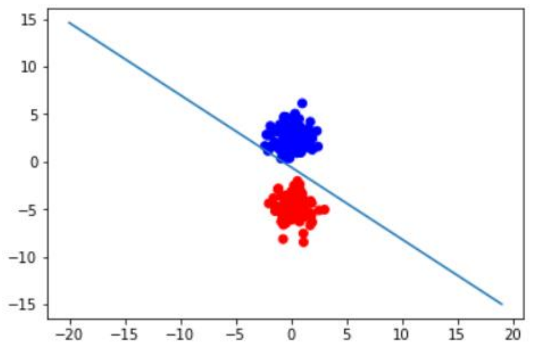
Tools used

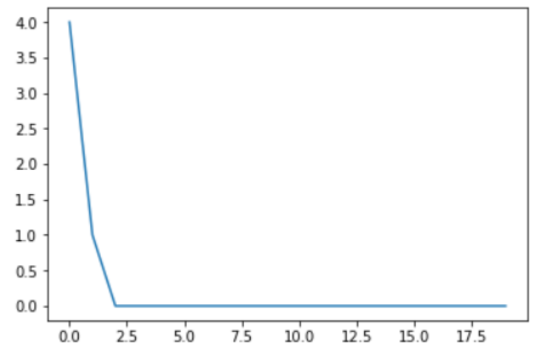
Jupyter Notebook is used to carry out the simulations and learning.
Keras framework is used for the Mackey glass time series findings.

Results

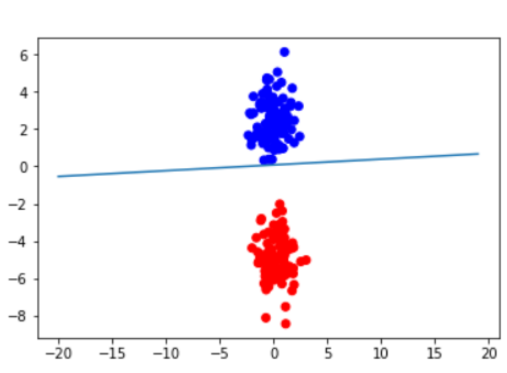
PART 1:

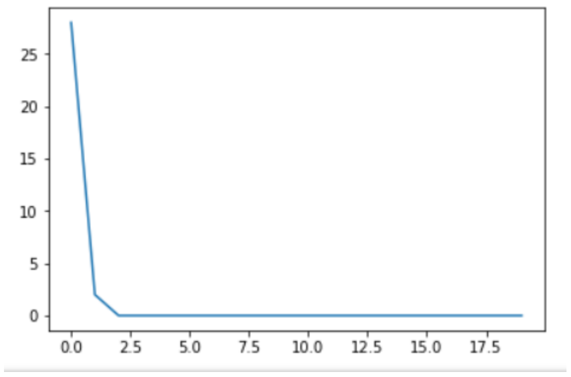
Single layer: Classification with perceptron rule





Classification with delta rule





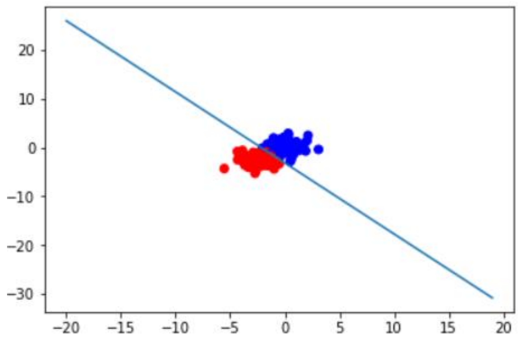
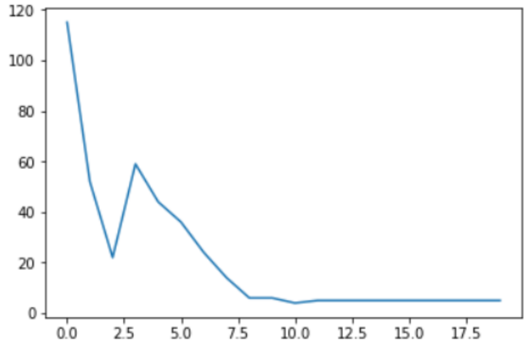
Artificial Neural Networks and Deep Architectures-DD2437

Learning and generalization in feed-forward networks - from perceptron learning to backprop

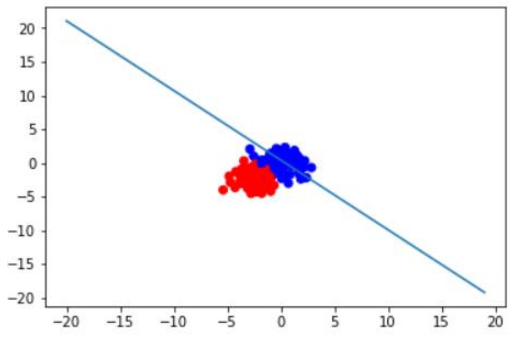
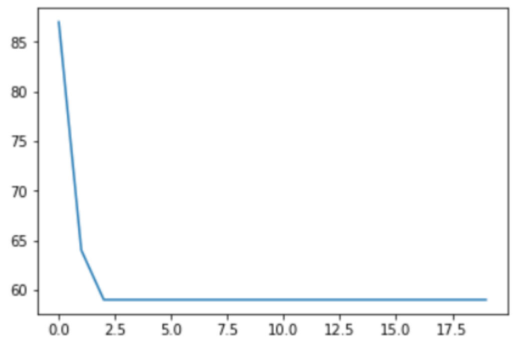
Rithika Harish Kumar, Suhas Sheshadri

Linearly un-separable data set:

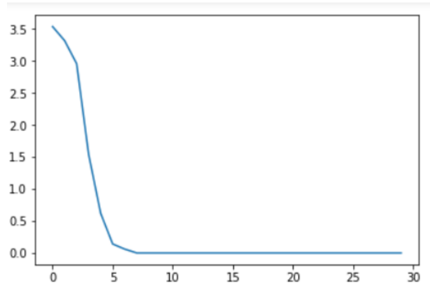
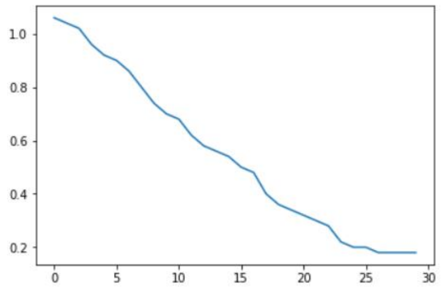
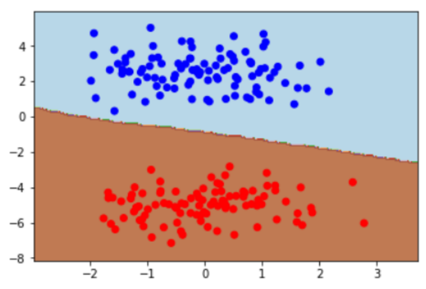
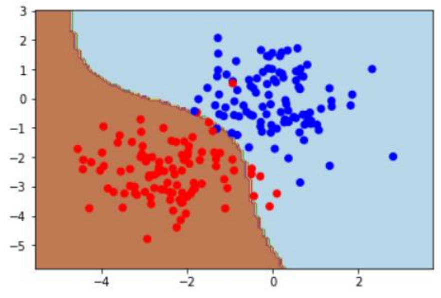
Classification with perceptron rule

Classification with delta rule

Multi-layer: Linear Data and Non-Linear data: with 15 Hidden Nodes

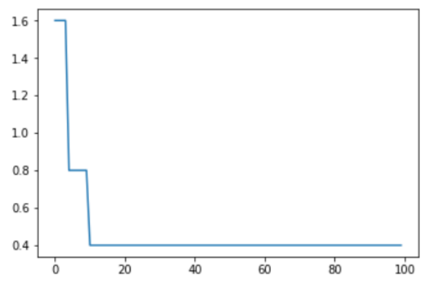
Artificial Neural Networks and Deep Architectures-DD2437

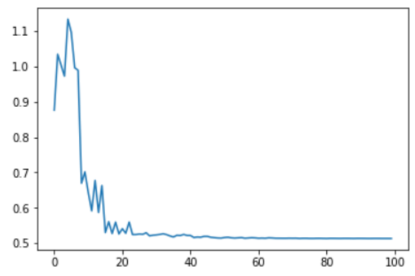
Learning and generalization in feed-forward networks - from perceptron learning to backprop

Rithika Harish Kumar, Suhas Sheshadri

Test Data set:

Hidden Nodes: 15 and 20





How do the training and test learning curves compare?

The test data has a higher MSE than that of the training data. Sometimes, the test data gives a lower MSE than the training data. This may be a case when there are outliers in the training data.

How do the training and test classification results depend on the size of the hidden layer?

The classification is better with an increase in the number of nodes in the hidden layer, up to a certain number of hidden nodes. The test data performance normalized after four hidden nodes, i.e. the MSE remains largely unchanged thereafter.

How many epoch iterations do you need for convergence?

With less than 10 iterations, convergence was achieved. The MSE decreases until the very last iteration, but not significantly.

Is there any difference between batch and sequential learning approaches?

The Sequential learner produces lower MSE but takes a few more epochs when compared to Batch learner.

Artificial Neural Networks and Deep Architectures-DD2437

Learning and generalization in feed-forward networks - from perceptron learning to backprop

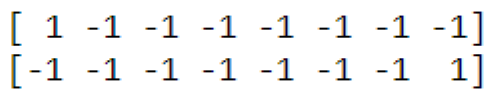
Rithika Harish Kumar, Suhas Sheshadri

Encoder:

With a proper selection of hidden nodes, the network always converges.

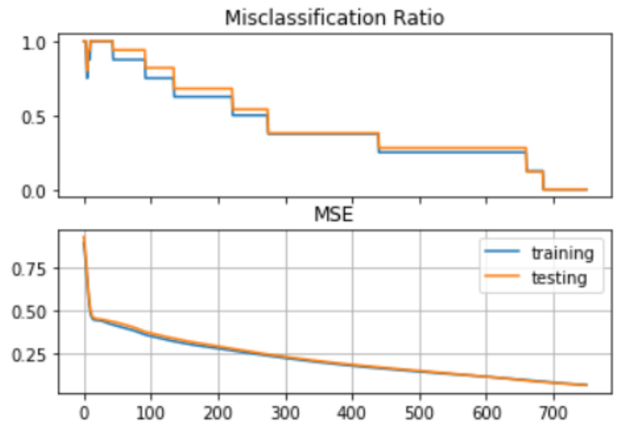
The hidden layer should be able to represent all the input sequences. A noisier model is obtained when the number of training samples exceeds the number of weights, since the weights cannot represent all the unique input combinations.

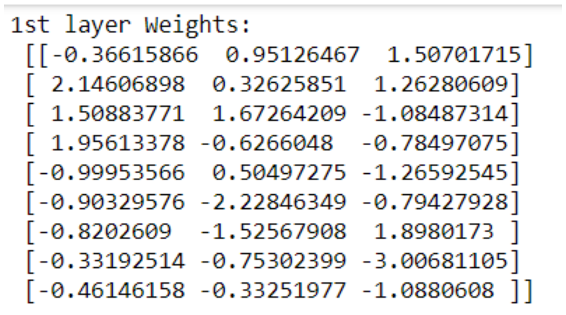
All the possible binary permutations of the three nodes handling the eight unique training samples are represented by the weight matrix.



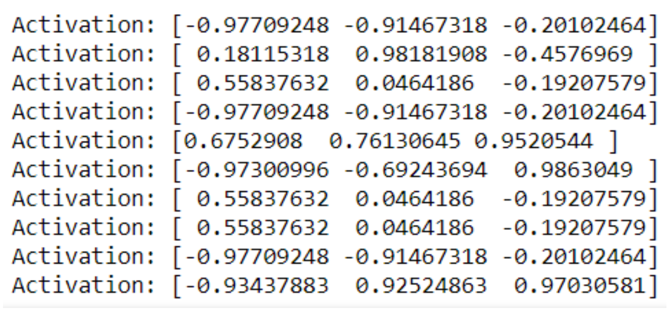
```
[ 1 -1 -1 -1 -1 -1 -1 -1]
[-1 -1 -1 -1 -1 -1 -1  1]
```

```
Misclassifications 0.0
MSE: 0.06002391621482802
```





```
1st layer Weights:
 [[-0.36615866  0.95126467  1.50701715]
 [ 2.14606898  0.32625851  1.26280609]
 [ 1.50883771  1.67264209 -1.08487314]
 [ 1.95613378 -0.6266048  -0.78497075]
 [-0.99953566  0.50497275 -1.26592545]
 [-0.90329576 -2.22846349 -0.79427928]
 [-0.8202609  -1.52567908  1.8980173 ]
 [-0.33192514 -0.75302399 -3.00681105]
 [-0.46146158 -0.33251977 -1.0880608 ]]
```



```
Activation: [-0.97709248 -0.91467318 -0.20102464]
Activation: [ 0.18115318  0.98181908 -0.4576969 ]
Activation: [ 0.55837632  0.0464186  -0.19207579]
Activation: [-0.97709248 -0.91467318 -0.20102464]
Activation: [0.6752908  0.76130645 0.9520544 ]
Activation: [-0.97300996 -0.69243694  0.9863049 ]
Activation: [ 0.55837632  0.0464186  -0.19207579]
Activation: [ 0.55837632  0.0464186  -0.19207579]
Activation: [-0.97709248 -0.91467318 -0.20102464]
Activation: [-0.93437883  0.92524863  0.97030581]
```

Artificial Neural Networks and Deep Architectures-DD2437

Learning and generalization in feed-forward networks - from perceptron learning to backprop

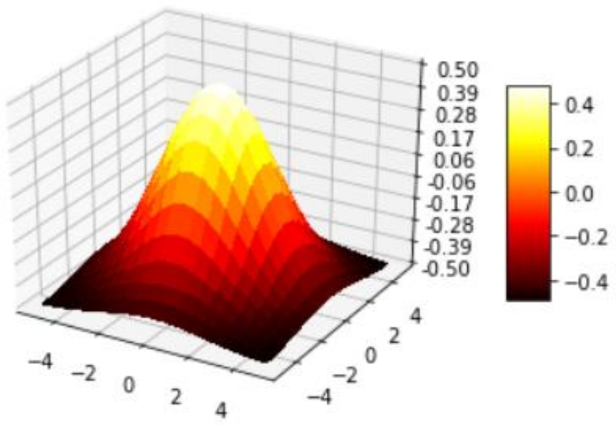
Rithika Harish Kumar, Suhas Sheshadri

Function approximation

Network with a few hidden nodes performed the best (i.e. 2, 3 and 4). A network with a large number of nodes (example: 25, 50 or above) resulted in overfitting.

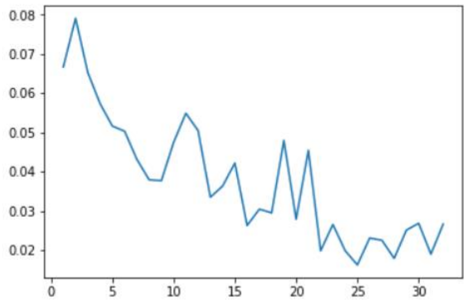
Networks with 2 to 7 nodes provides a good compromise between bias and variance.

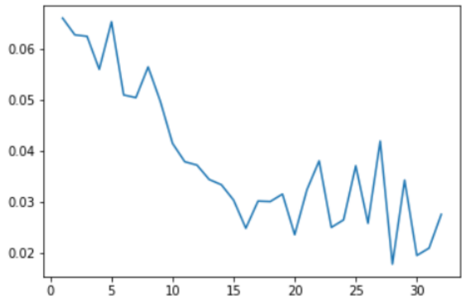
Learning after function approximation:



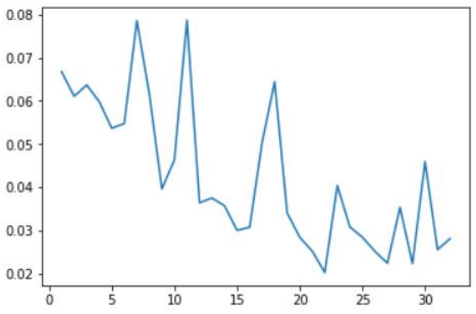
Different number of nodes in hidden layer:

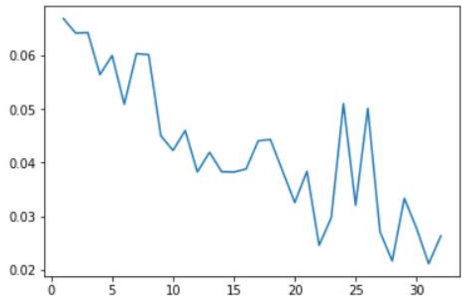
Nodes: 1 and 2





Nodes: 3 and 4



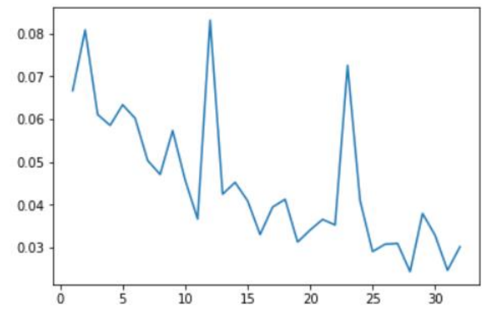


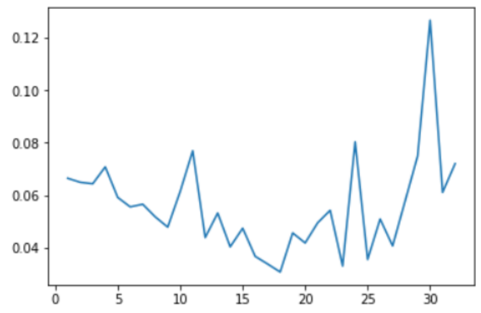
Artificial Neural Networks and Deep Architectures-DD2437

Learning and generalization in feed-forward networks - from perceptron learning to backprop

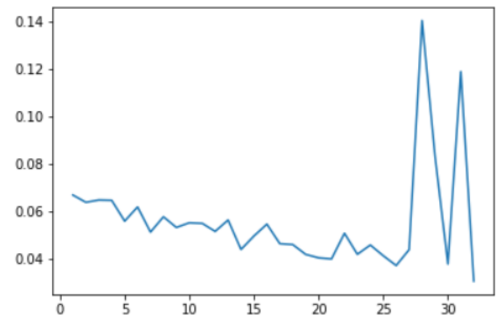
Rithika Harish Kumar, Suhas Sheshadri

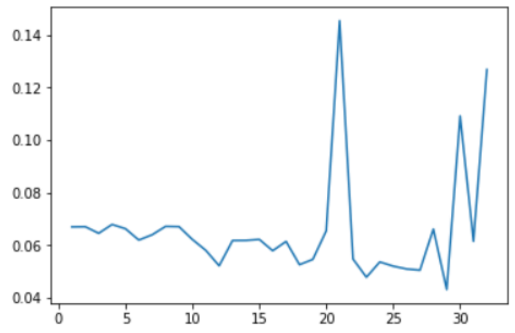
Nodes: 5 and 10





Nodes: 20 and 50

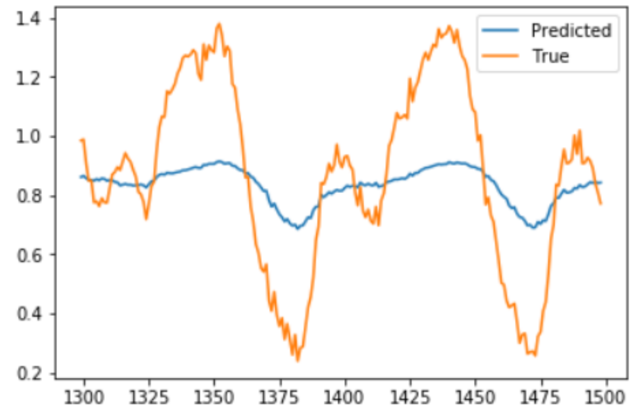




Mackey glass time series data

The lowest MSE on the test set was the model with zero regularization and 14 neurons in the hidden layer. Regularization is done to suppress the weights and to lower the variance of the model. We trained for 100 epochs and did not encounter an excessively complex model. The training and test sets have similar variance. Regularization was not required, may be due to the absence of outliers.

Early stopping resulted in a marginally lower computational time, but without any effect on the MSE.



```
STD:  0.03  MSE:  0.06719265350859974
Number of nodes in first:  8 Number of nodes in second:  4  Time:  5.008819737122394
```

Rithika Harish Kumar, Suhas Sheshadri

4.3.2

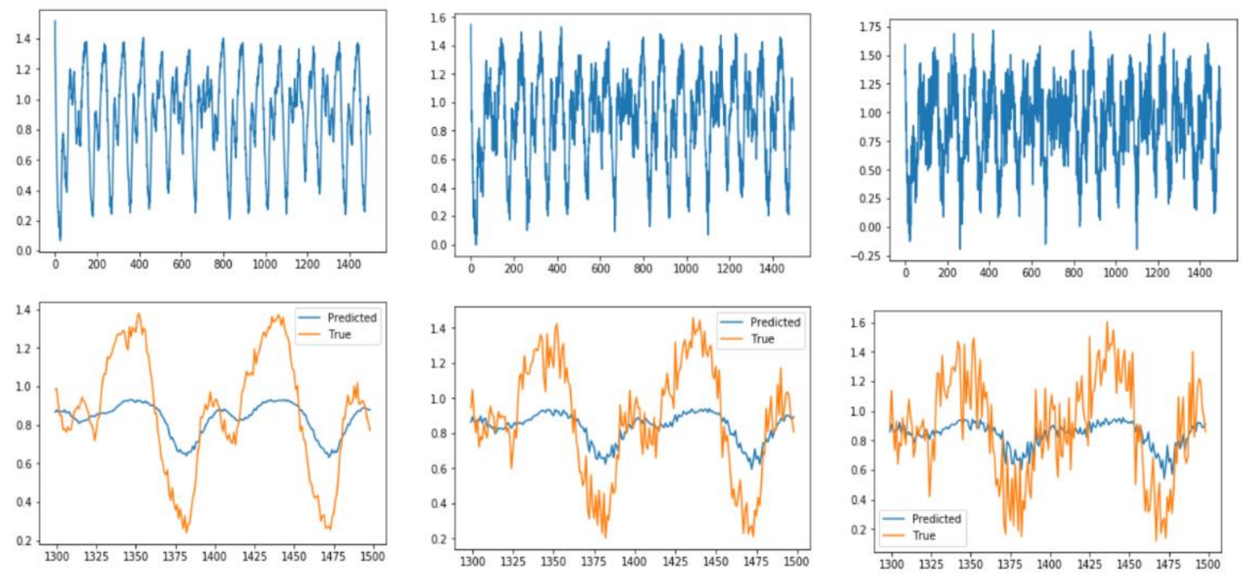
Increasing the noise resulted in difficulty to find a good fit. We noticed that the variance decreased when the first hidden layer had more nodes than the second layer. When both layers had same number of nodes or when the second layer had more nodes than the first layer, we observed no effect on the outcome.

An increased Regularization parameter resulted in higher MSE for a network with static noise. But, Regularization parameter can be increased with the increase in the noise, so that we can obtain a better fitting model.

The computational cost of Two-layer Perceptron is marginally lesser than that of a Three-layer one.

In the Three-layer perceptron, the MSE and the time reduced for an increase in the hidden nodes up to certain number of nodes. After that, there was an increase in the MSE and the time taken for fitting the model.

```
Single layer network:
```



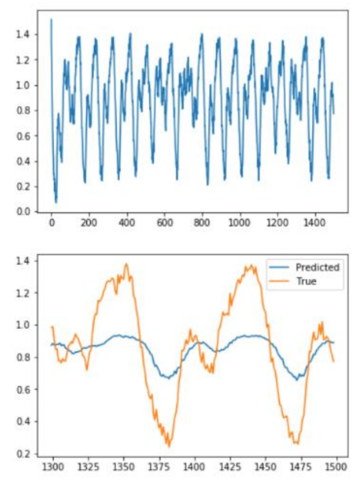
```
STD:  0.03  MSE:  0.05909857140255296
Number of nodes:  8  Time:  2.6797888240991767

STD:  0.09  MSE:  0.06285638673412373
Number of nodes:  8  Time:  3.3411247746794004

STD:  0.18  MSE:  0.07979870609119476
Number of nodes:  8  Time:  3.188040208096794
```

Artificial Neural Networks and Deep Architectures-DD2437

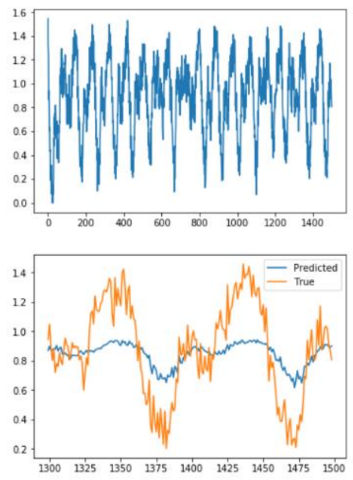
Learning and generalization in feed-forward networks - from perceptron learning to backprop

Rithika Harish Kumar, Suhas Sheshadri

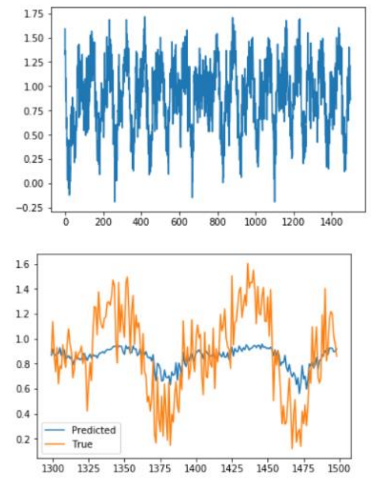
Double layer network:



```
STD:  0.03  MSE:  0.06062005486984746
Number of nodes in first:  8 Number of nodes in second:  1  Time:  2.897221099628581
```



```
STD:  0.09  MSE:  0.06434421931798334
Number of nodes in first:  8 Number of nodes in second:  1  Time:  3.7553804309476675
```



```
STD:  0.18  MSE:  0.08145377235117204
Number of nodes in first:  8 Number of nodes in second:  1  Time:  3.826011456869935
```

Conclusions:

Lab assignment successfully completed, aims achieved.