Ryerson University

Department of Electrical & Computer Engineering
Faculty of Engineering & Architectural Science

Course Title:	**Intelligent Systems**
Course Number:	**ELE 888**
Semester/Year (e.g.F2016)	**W2021**

Instructor:	**Dr. Xiao-Ping Zhang**

Assignment/Lab Number:	**Lab 3**
Assignment/Lab Title:	**Multilayer Neural Networks**

Submission Date:	**Mar-28-2021**
Due Date:	**Mar-28-2021**

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
An	**Nicholas**	**500750378**	**3**	**NA**
Parmar	**Gurvir**	**500765786**	**1**	**GP**
Shreekant	**Vatsal**	**500771363**	**1**	**VS**

**By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.pdf*

Objective:

To implement a multi-layer neural network using the backpropagation algorithm to classify linear non-separable data. Multi-layer neural network (MNN) implements the linear discriminant functions; however, the input patterns are mapped non-linearly. Utilizing simple equations where the form of nonlinearity can be learned through a training data indicates how powerful and easy neural networks are. A popular method for training the MNN is based on the gradient descent in error is known as backpropagation algorithm.

The sensitivity of unit *k* is given by

$$\delta_k = (t_k - z_k) * f^l(net_k)$$

And the sensitivity for a hidden unit is given by

$$\delta_j = f^l(net_j) \sum_{k=1}^{c} w_{kj}\delta_k$$

The objective of this laboratory estimate the weight vectors of the input-hidden and hidden-output layer for a given activation function. $f(net_k)$ and $f(net_j)$ are the non-linear functions.

Observation:

XOR:

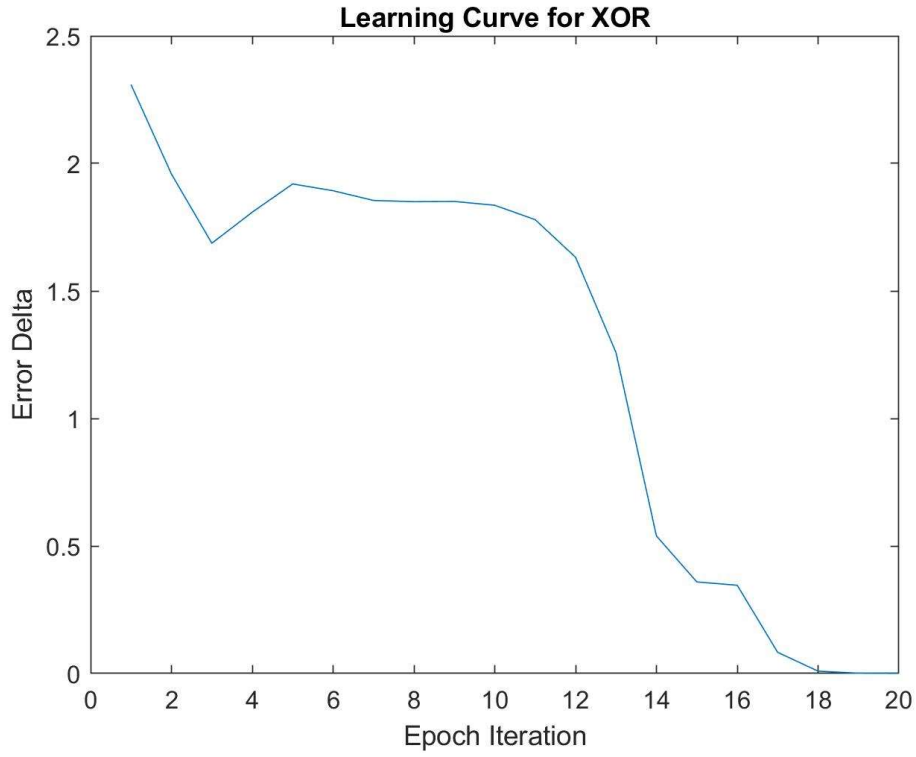

Fig 1: XOR Learning Curve

Table 1: XOR with 20 iterations & Wine Training Accuracy of %98.6442

Weight Vector (wkj)	Weight Vector (wj1)	Weight vector (wj2)
-3.394	1.9315	2.898
3.9951	1.717	-2.1641
2.6187	1.7861	-2.8405

Wine Dataset:

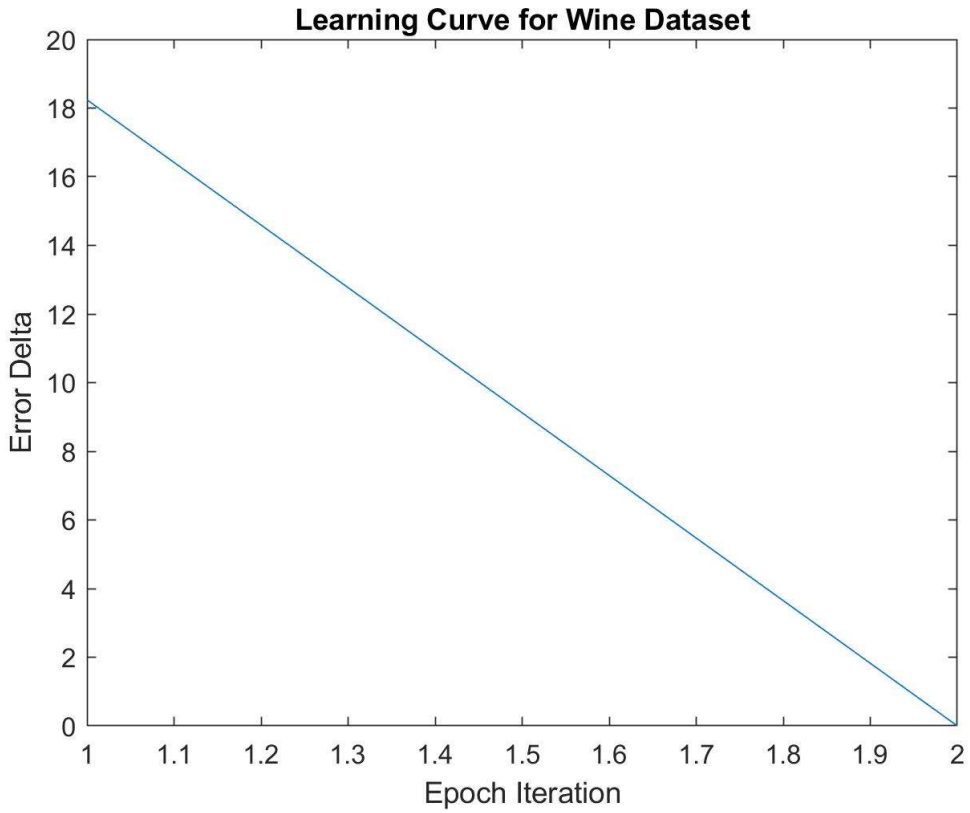

Fig 2.1: Wine Dataset Learning Curve with 2 Iterations

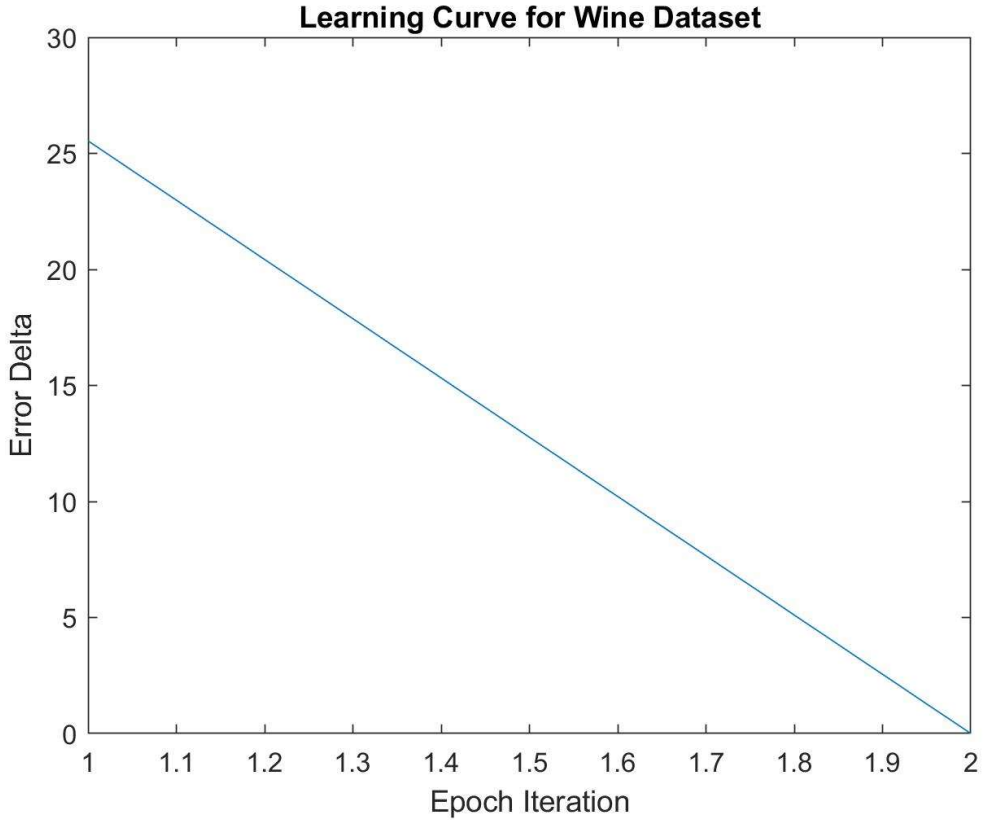

Fig 2.2: Wine Dataset Learning Curve with 2 Iterations

Table 2: Wine Dataset with 2 iterations & Wine Training Accuracy of %99.9892

Weight Vector (wkj)	Weight Vector (wj1)	Weight vector (wj2)
3.2224	1.9047e-06	0.65
6.0314	0.50007	-0.7
-5.6314	0.10001	-0.9

Table 3: Wine Dataset with 2 iterations & Wine Training Accuracy of %99.9995

Weight Vector (wkj)	Weight Vector (wj1)	Weight vector (wj2)
4.7114	5.5479e-06	0.65
8.324	0.50012	-0.7
-7.924	0.10002	-0.9

Conclusion

The implemented a multi-layer neural network that uses a back-propagation algorithm that classifies non-separable data. For the first part of the lab, creating a 2-2-1 neural network solved for the XOR problem. The operation learning threshold was 0.001 and the weight vector was 0.1. The two inputs were two 4x1 vectors, with an accuracy of 98.6% after 20 epochs. The second portion consisted of a same 2-2-1 neural network. However, it was applied using a wine data set to classify non-separable data. Using the classes featured, the accuracy computed was 100%.