Ryerson University

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

Course Title:	**Intelligent Systems**
Course Number:	**ELE888**
Semester/Year (e.g.F2018)	**F2018**

Instructor:	**Xiao-Ping Zhang**

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

Submission Date:	2019/03/25
Due Date:	2019/03/25

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Patel	Pranay	500702502	1	
Chong	Barry	500508396	1	
Kassam	Shahezad	500682174	1	

*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

ELE888 Intelligent Systems
Lab 3: Multi-Layer Neural Networks

Exercise Point 1:

Construct a 2-2-1 neural network using the batch backpropagation algorithm for solving the classical XOR problem:

This exercise point has been achieved in the code that is submitted with this exercise labelled as lab3-1.m

Exercise Point 2:

Verify that the computed final weight vectors satisfy the XOR operation. Plot the learning curve and note the number of epochs needed for convergence.

Below in **Figure 1** is the classification plot the neural network has output.

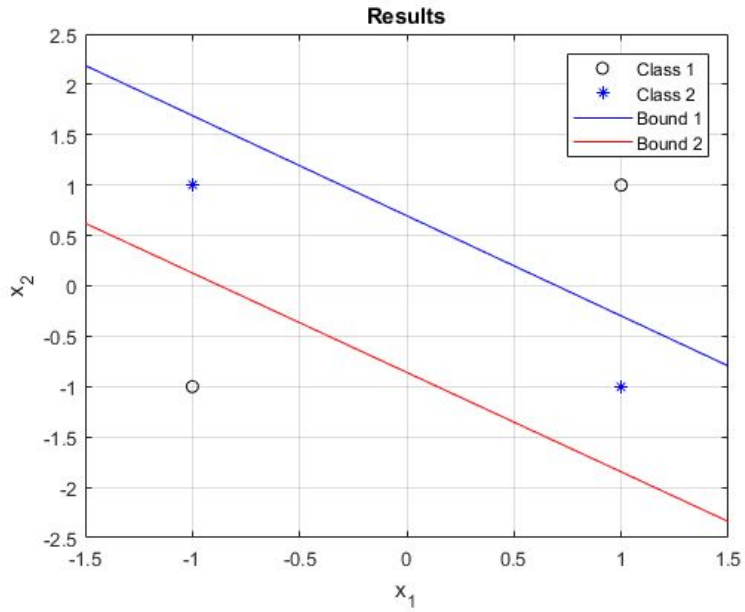

Figure 1: Decision Boundaries of the Multi-layer Neural Network

In the figure Class 1 represents the 0's output for the XOR logic gate and Class 2 represents the 1's output. The boundaries are the decision boundaries the neural network has developed from its weight backpropagation calculations and refinement to decide whether the result will be a 0 or a 1. The learning curve of the aforementioned backpropagation algorithm is shown below in **Figure 2**.

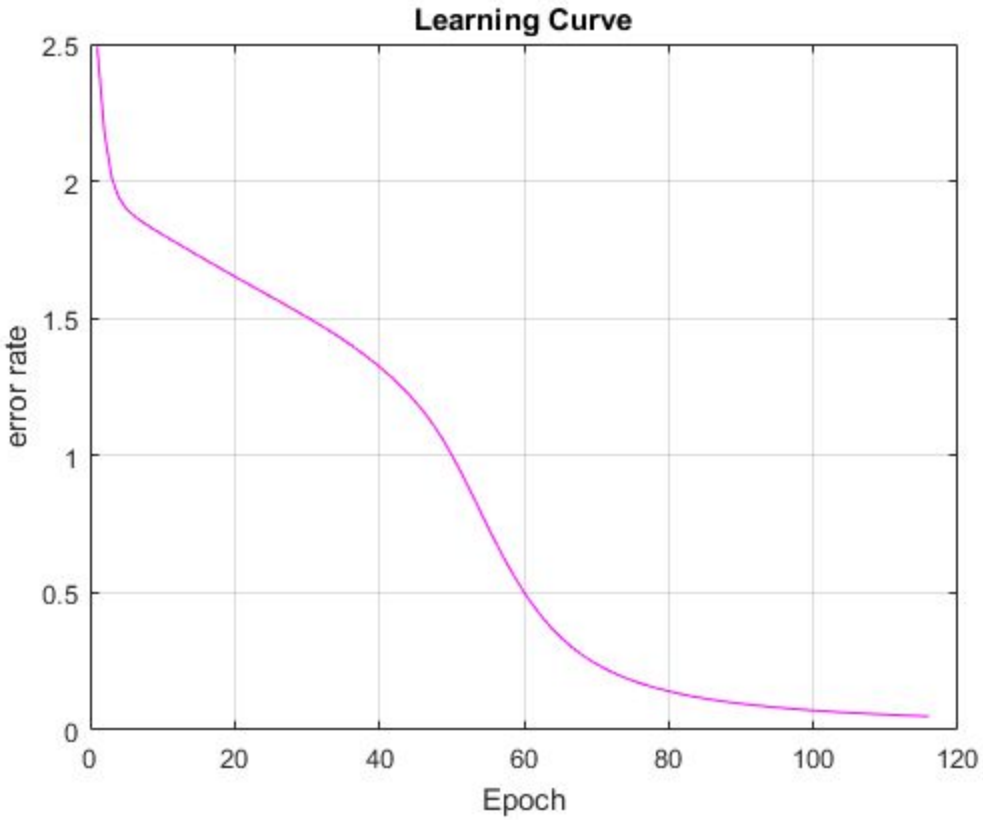

Figure 2: Learning Curve of the Neural Network Backpropagation Algorithm

From the learning curve we can see that the amount of Epochs required for convergence is 116. After running the neural network more than a few times it becomes clear that as long as there are more than 100 Epochs, the result will converge and have a accurate solution/output. However, when the epochs are below 100 for our neural network, the accuracy diminishes severely.

Exercise Point 3:

*Repeat Exercise 1 for the **"Wine"** data set from the UCI repository.*

Similar to exercise 1, the code for the neural network of classifying the wine is included in the submission of this report.

Exercise Point 4:

Compute the classification accuracy for the given data. Plot the learning curve and note the number of epochs needed for convergence.

Below in **Figure 3** is the classification boundary the neural network has created.

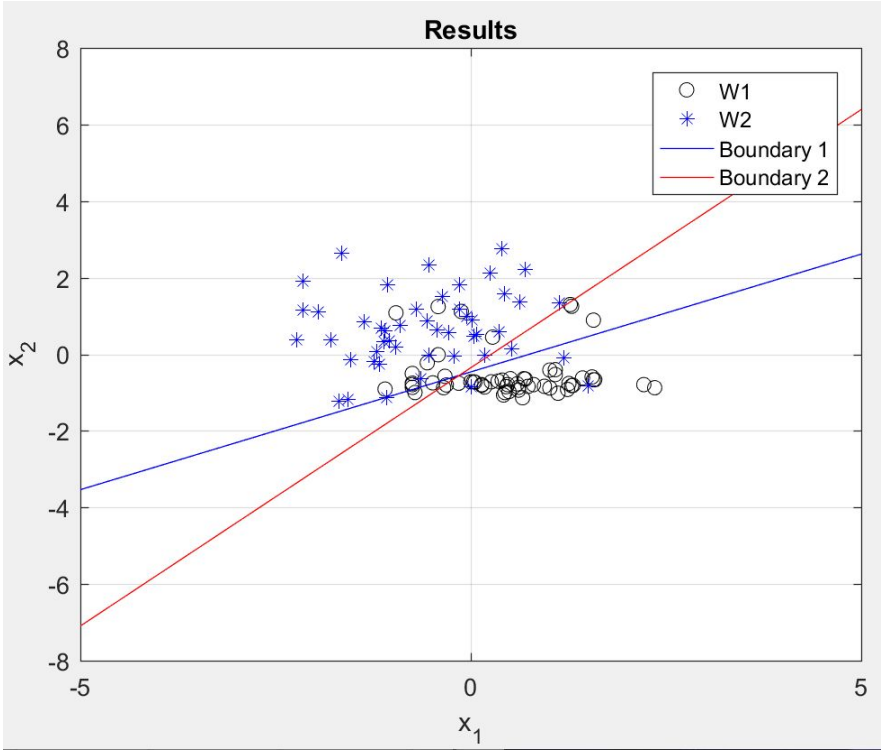

Figure 3: Decision boundaries of the Multi-layer Neural Network using wine data

As seen in the figure above, the neural network devised two clear boundaries to classify the given data. The learning curve of how the neural network devised these boundaries with respect the epoch and error rate is shown below in **Figure 4**.

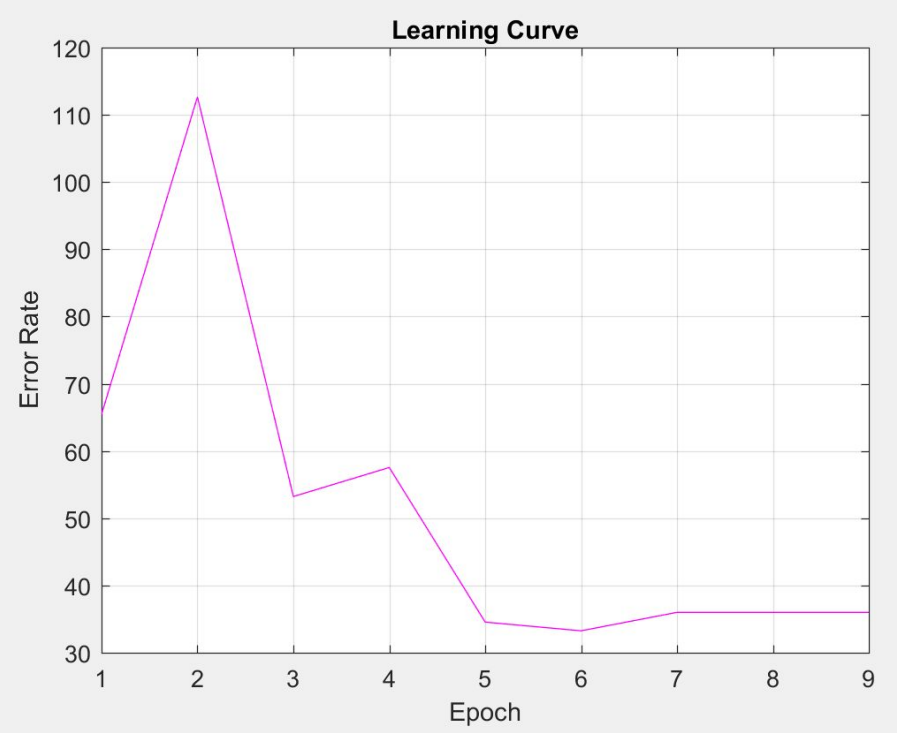

Figure 4: Learning Curve of the Neural Network Backpropagation Algorithm based on wine data

From the learning curve, **Figure 4**, we can see that the amount of Epochs required for convergence is 9, with a classification accuracy of 83.18% as seen in **Figure 3**, the data is well classified.

Conclusion

The experiment provided positive results, clearly showing the results of a multi-layer neural network. The first exercise with using backpropagation algorithm to show the classic XOR problem was a simple way of showing it classifying correctly. The second part using wine data, with a high accuracy of 83% shows the algorithm working properly and well. The multi-layer neural network allows the classification to be more accurate as shown in the graphs, figure 1 and figure 3. These results show a simple implementation of a multi-layer neural network, using a backpropagation algorithm.