TBMI26 – Computer Assignment Report  
Supervised Learning

Deadline – March 15 2019

Author/-s:

In order to pass the assignment you will need to answer the following questions and upload the document to LISAM. **You will also need to upload all code in .m-file format**. We will correct the reports continuously so feel free to send them as soon as possible. If you meet the deadline you will have the lab part of the course reported in LADOK together with the exam. If not, you’ll get the lab part reported during the re-exam period.

1. **Give an overview of the data from a machine learning perspective. Consider if you need linear or non-linear classifiers etc.**

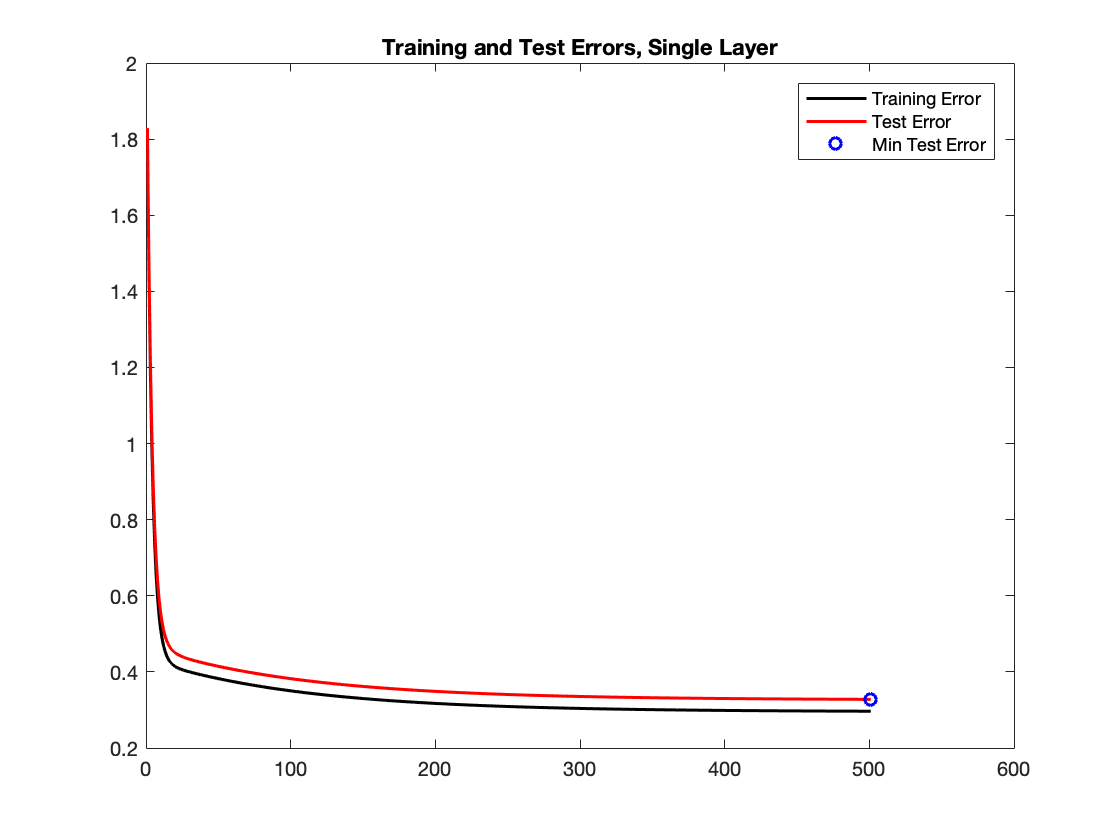
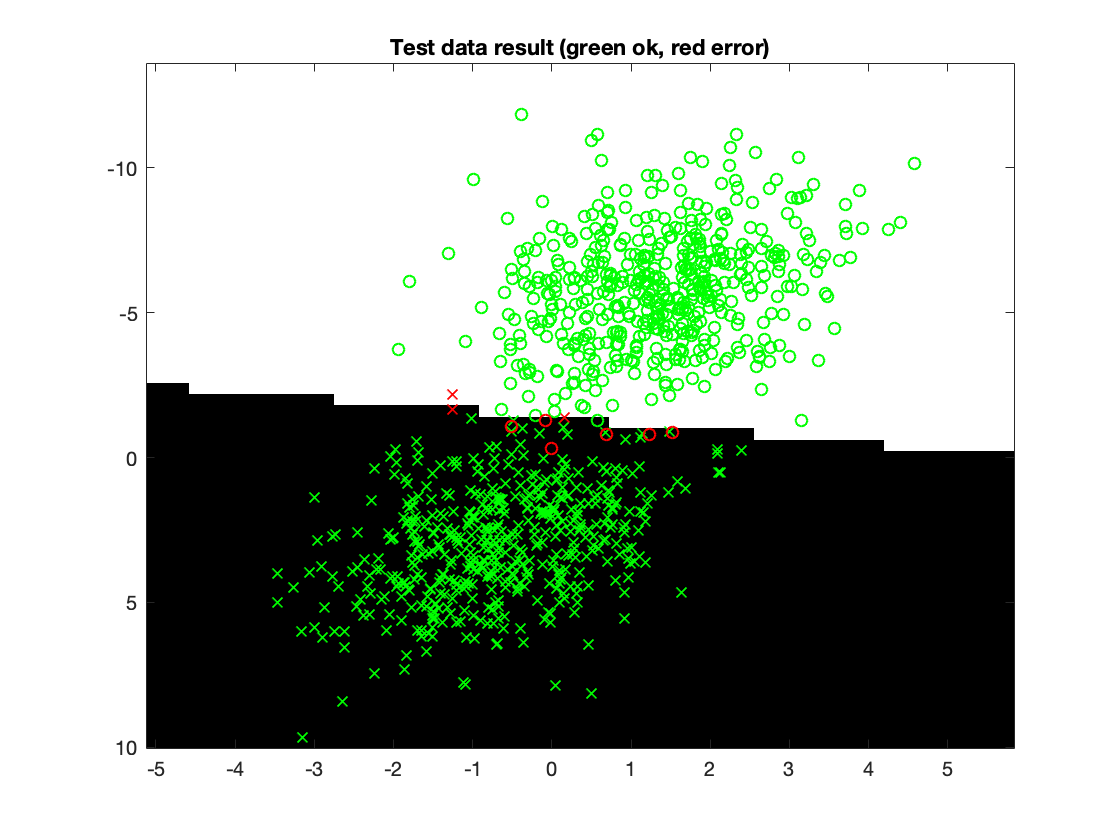
Dataset 1: Two clearly separated gaussian distributions, with little overlap. Can easily be separated by a line, meaning only a linear classifier is necessary.

Dataset 2: While one set is like before, the other reaches around it in a manner which makes linear separation impossible. This can be solved with a simple transform of the data, meaning a non-linear classifier is necessary, albeit not a complicated one.

Dataset 3: Much like in the last case, the different sets envelop each other. This time there is also the complication of having three different distributions. Same verdict as last dataset.

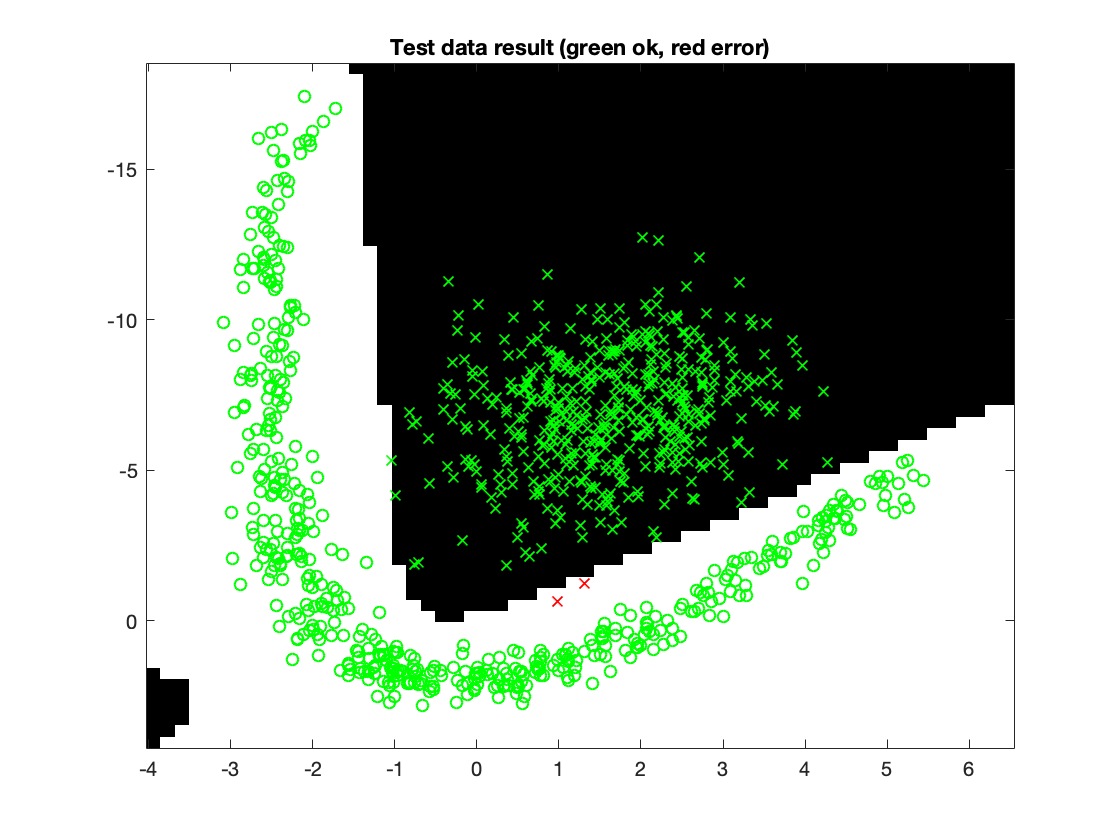
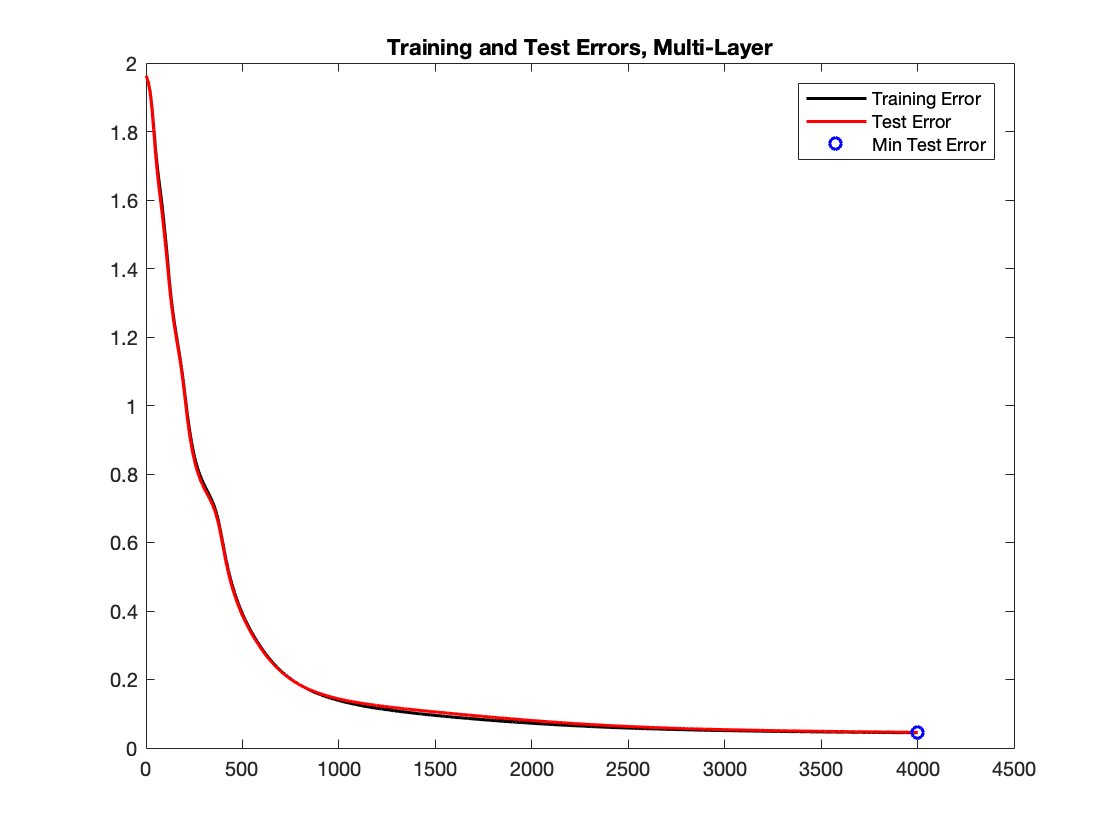
Dataset 4: Very complicated dataset with 65 features requiring complicated conclusions to be drawn as to how they interact with each other to form numbers. Absolutely necessary to use a non-linear classifier. Doubt it is even possible to train a linear one to any effect.

1. **Explain why the down sampling of the OCR data (done as pre-processing) result in a more robust feature representation. See** [**http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits**](http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits)
2. **Give a short summery of how you implemented the kNN algorithm.**
3. **Explain how you handle draws in kNN, e.g. with two classes (k = 2)?**
4. **Explain how you selected the best k for each dataset using cross validation. Include the accuracy and images of your results for each dataset.**
5. **Give a short summery of your backprop network implementations (single + multi). You do not need to derive the update rules.**
6. **Present the results from the backprop training and how you reached the accuracy criteria for each dataset. Motivate your choice of network for each dataset. Explain how you selected good values for the learning rate, iterations and number of hidden neurons. Include images of your best result for each dataset, including parameters etc.**

****

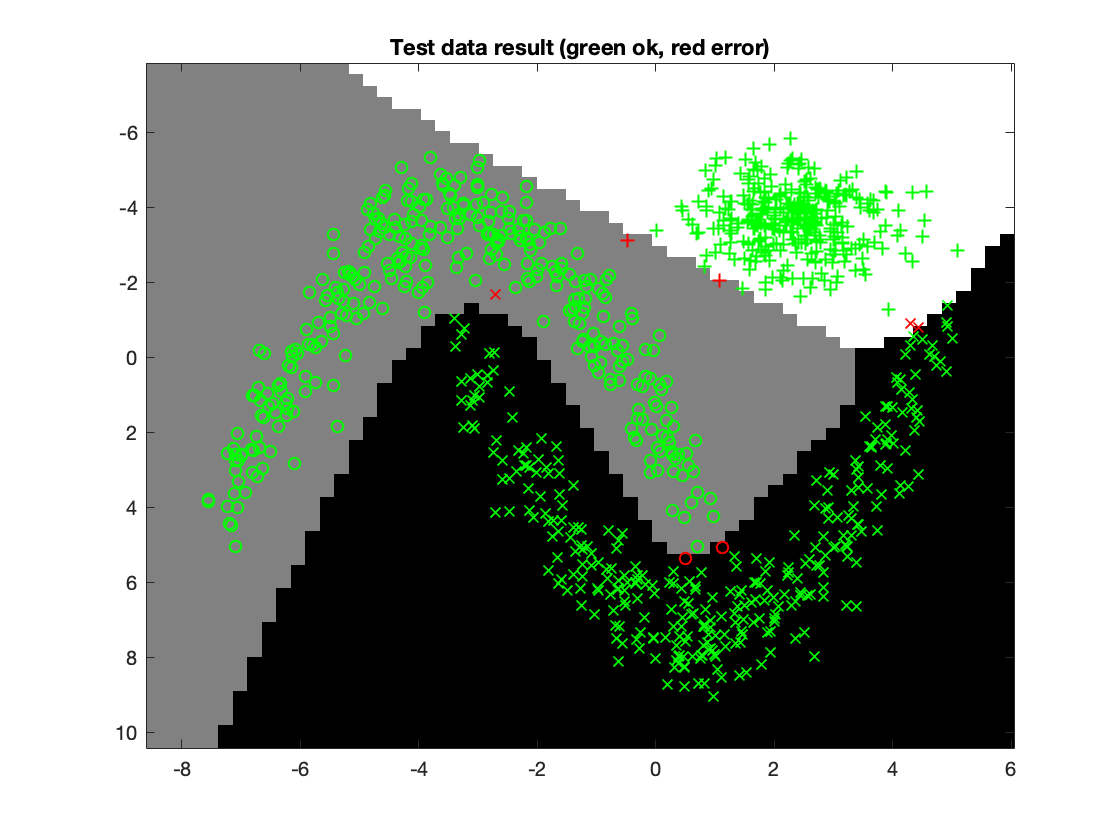
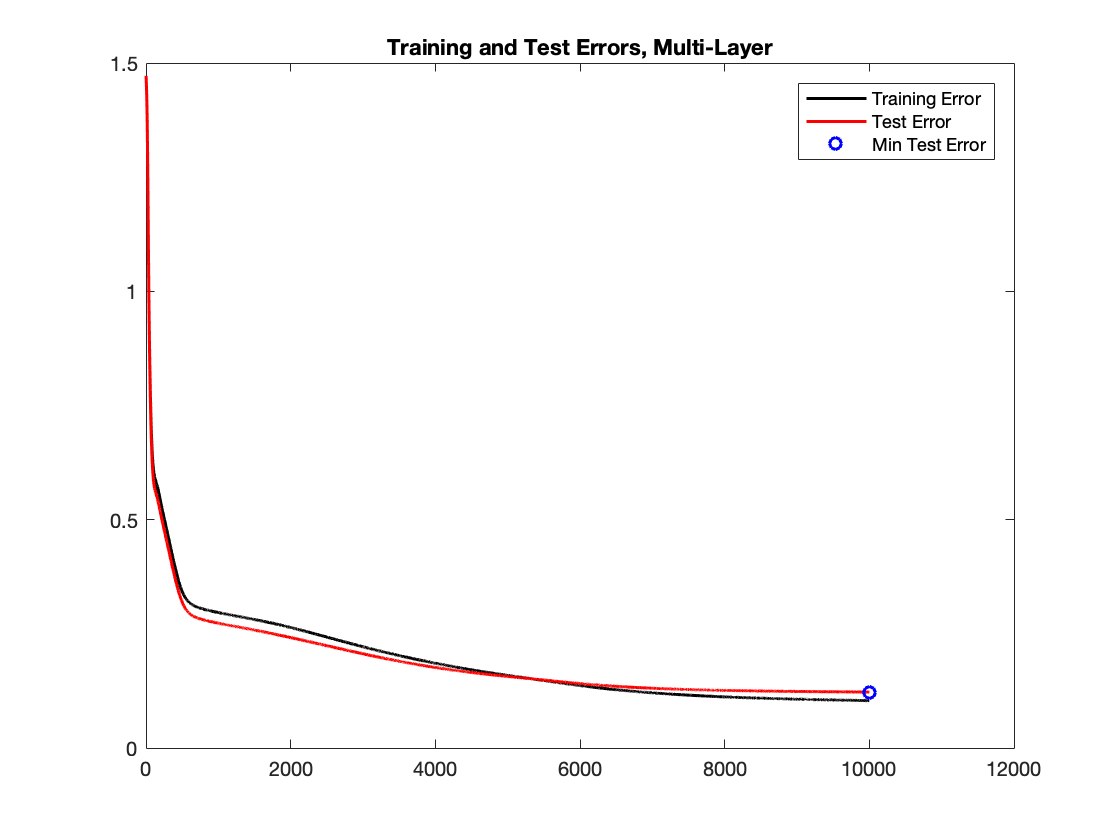
Dataset 1: --Single Layer Network--

* Accuracy: 99.1%
* Training Time: 0.0410 s
* Epochs: 500
* Learning Rate: 0.005
* Hidden Neurons: -
* Motivation: Linearly separable dataset, which means it should have no need for any hidden layers. Additionally, epochs can be set low and learning rate high as the network will converge early.



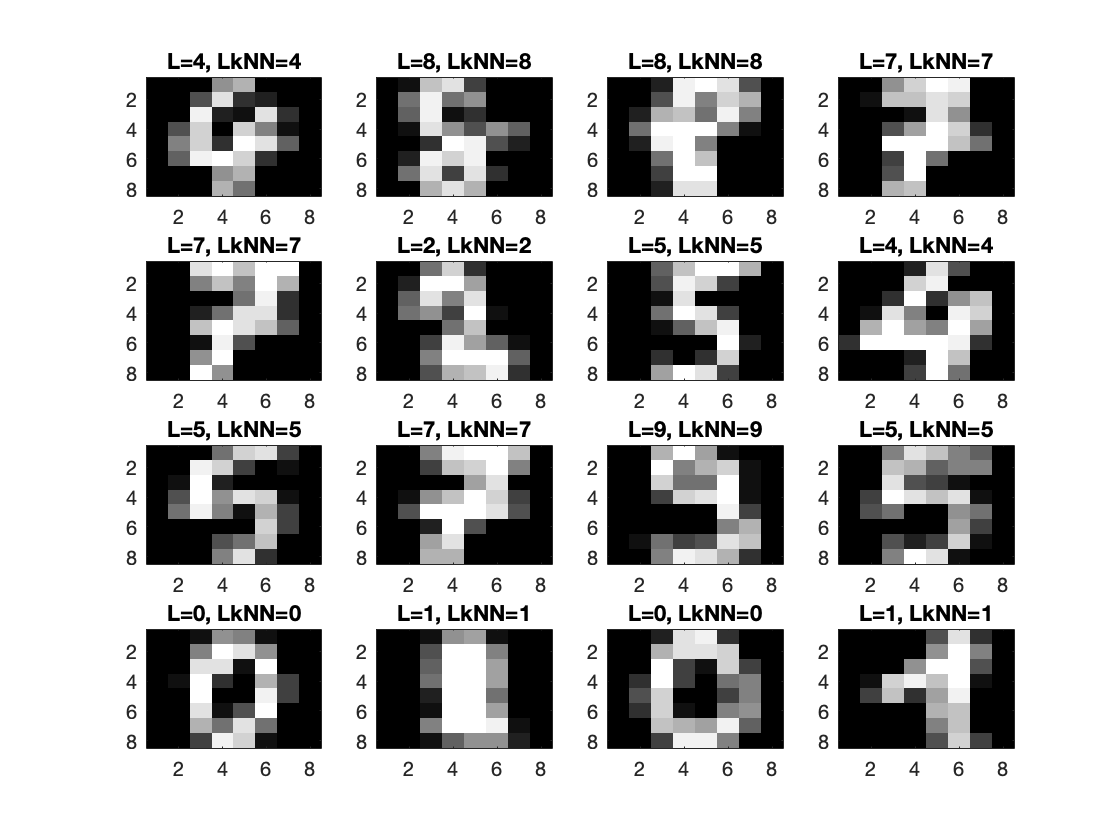
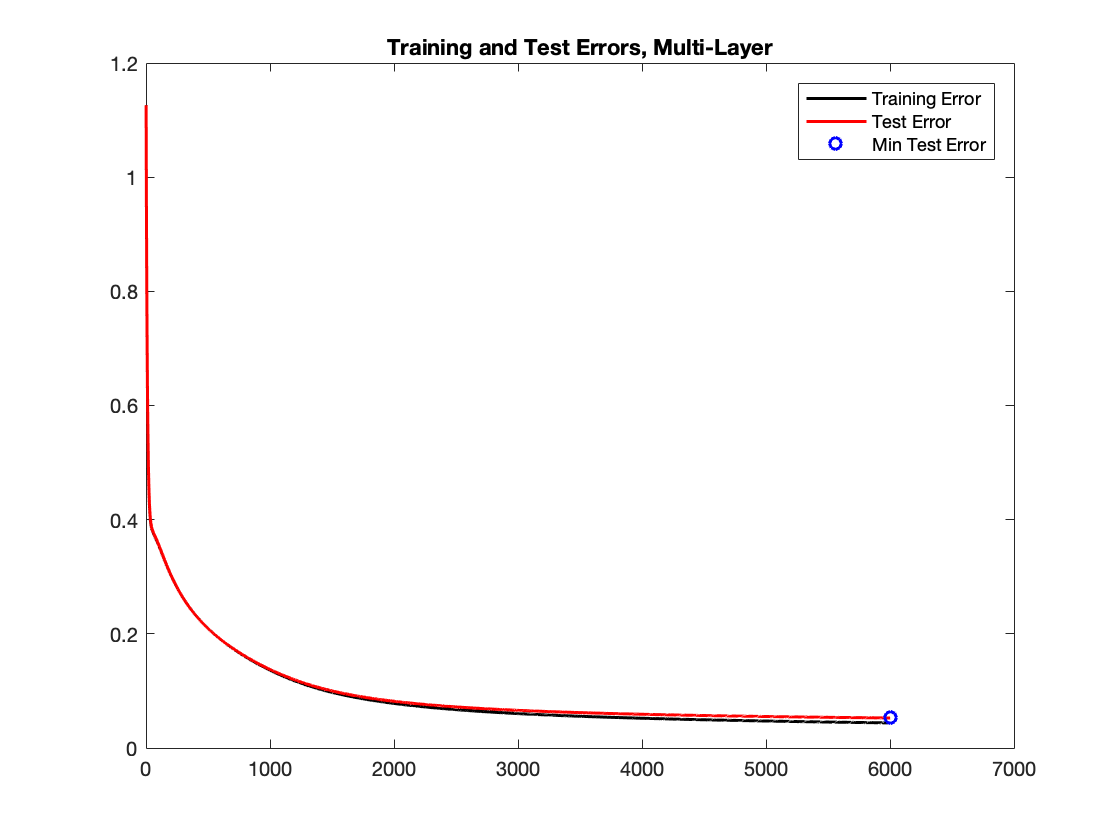
Dataset 2:

* Accuracy: 99.8%
* Training Time: 1.1839 s
* Epochs: 4000
* Learning Rate: 0.01 (Can be high, as there is ample space between classes and even sharp lines will separate the dataset nicely)
* Hidden Neurons: 3 (Minimum amount required to get the bent shape required to fit the data)
* Motivation: The distance between datapoints of different labels is large, so the network can afford to work fast and dirty. Additionally, the data only really needs one more dimension to be linearly separable, so only 3 hidden neurons are needed.



Dataset 3:

* Accuracy: 99.3%
* Training Time: 4.5765 s
* Epochs: 10000
* Learning Rate: 0.01
* Hidden Neurons: 5
* Motivation: Because of the more complicated nature of the separation of data, as well as the increase in number of neurons (which in turn also is because of the complexity), the network needed more epochs to train. The number of hidden neurons is the lowest number that can describe a separation like that of the image, keeping training time low by not adding unnecessary neurons to train. Clear separations between labels and few outliers means we can still have a fast learning rate.



Dataset 4:

* Accuracy: 96.5%
* Training Time: 62.6811 s
* Epochs: 6000
* Learning Rate: 0.001
* Hidden Neurons: 33
* Motivation: This dataset had a much higher number of relevant features, and 9 output labels. As such, many more hidden neurons were required. Additionally, the learning rate had to be decreased, probably because of the close similarity between many targets, leading to an unstable predictor if the learning rate is just a little too high.

1. **Present the results, including images, of your example of a non-generalizable backprop solution. Explain why this example is non-generalizable.**
2. **Give a final discussion and conclusion where you explain the differences between the performances of the different classifiers. Pros and cons etc.**
3. **Do you think there is something that can improve the results? Pre-processing, algorithm-wise etc.**