TBMI26 – Computer Assignment Report  
Supervised Learning

Deadline – March 15 2020

Fredrik Johansson

In order to pass the assignment you will need to answer the following questions and upload the document to LISAM. Please upload the document in PDF format. **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 four datasets from a machine learning perspective. Consider if you need linear or non-linear classifiers etc.**

The first two datasets consist of data 2D data with two classes. The data in the first dataset form two convex clusters which means that we can use a linear classifier. In the second dataset, the data form one convex and one non-convex cluster. Because we have one non-convex cluster, we must use a non-linear classifier.

The third dataset consist of 2D data with three classes and two of them are non-convex clusters. Therefore, we must use a non-linear classifier. The last dataset consists of 10 classes with 64 features each. With that many features a non-linear classifier is preferable.

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)

When writing numbers on a paper, the edges between the numbers and the paper will consist of a grayscale. By down sampling the grayscale can be removed so the data is reduced to binary, either paper(white) or number(black). Since all the numbers have this grayscale it is not unique for each digit. Therefore, by removing it we achieve a more robust feature representation.

1. **Give a short summary of how you implemented the kNN algorithm.**

I iterate through all the datapoints and for each point I calculate the distance to all other points. Then I store the k nearest datapoint indices, and with the indices I check which class I should predict based on the training data. Lastly, the final prediction is based on the class with the most amount of predictions.

1. **Explain how you handle draws in kNN, e.g. with two classes (k = 2)?**

I use the matlab function mode, if it is a tie it returns the class with the smallest number. It may not be the best approach, but it is okay for the performance needed. A better approach would for too example be to return the class with the smallest mean distance.

1. **Explain how you selected the best k for each dataset using cross validation. Include the accuracy and images of your results for each dataset.**
2. **Give a short summary of your backprop implementations (single + multi). You do not need to derive the update rules.**
3. **Present the results from the neural network 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.**
4. **Present the results, including images, of your example of a non-generalizable backprop solution. Explain why this example is non-generalizable.**
5. **Give a final discussion and conclusion where you explain the differences between the performances of the different classifiers. Pros and cons etc.**
6. **Do you think there is something that can improve the results? Pre-processing, algorithm-wise etc.**