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.**
2. **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)
3. **Give a short summery of how you implemented the kNN algorithm.**
4. **Explain how you handle draws in kNN, e.g. with two classes (k = 2)?**
5. **Explain how you selected the best k for each dataset using cross validation. Include the accuracy and images of your results for each dataset.**
6. **Give a short summery of your backprop network implementations (single + multi). You do not need to derive the update rules.**
7. **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.**
8. **Present the results, including images, of your example of a non-generalizable backprop solution. Explain why this example is non-generalizable.**
9. **Give a final discussion and conclusion where you explain the differences between the performances of the different classifiers. Pros and cons etc.**
10. **Do you think there is something that can improve the results? Pre-processing, algorithm-wise etc.**