Centre for Mathematical Sciences, Mathematics, LTH, November 2020

# Medical Image Analysis, Assignment 1

## Introduction

In this assignment you are going to work with machine learning algorithms on both synthetic and real datasets. The datasets and some example scripts can be downloaded from the course home page. The purpose of this assignment is to develop your understanding on how to run some different types of machine learning algorithms. In particular we are going to look at

- Non-parametric estimation of probability density functions
- Machine Learning for Cell Type Classification using hand-coded features
- Machine Learning for Cell Type Classification using Deep Learning

Each assignment in the course is given a grade. Assignment 1 has three tasks (given in the gray frames below). If you complete the first task you'll pass and the assignment gets the grade 3. Minimum requirement for grade 4 is a successful completion of tasks one and two. Minimum requirement for grade 5 is a successful completion of tasks one, two and three. The teachers will correct and comment your solution and if you fail to pass the first time around we will ask you to improve and complement your report.

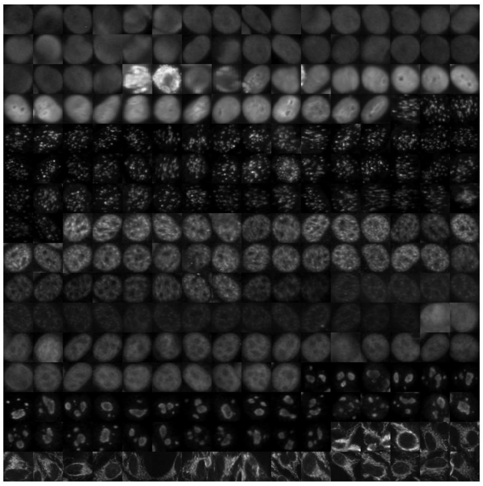
> The deep learning examples in matlab (for task 3) requires a reasonably recent matlab version. Download the latest version from program.ddg.lth.se 'Datordrift-gruppen'. The scripts for task 3 have been tested on matlab version R2018b, but might work on R2017b as well.

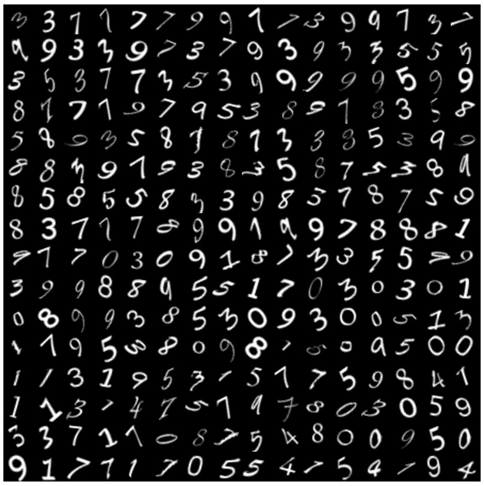
## The Rules

The assignment is **handed out** at the lecture **Tuesday 3 November**. The deadline is on **Sunday 15 November** (23:59 CET). Each student should hand in his or her own individual solution and should, upon request, be able to present the details in all the steps of the used algorithm. You are, however, allowed to discuss the assignment-problem with others. You may also ask your teachers and the course assistants for advice, if needed.

**The report.** Present your work in a report written in English or Swedish. The report should (at least) contain an introduction, a description of theory and the methods used, a section with the results and a conclusion where the results are discussed. Don't worry if your report becomes longer because of many figures. This is also fine.

**Submitting your work.** We are using Canvas for the assignments. Submit your solution in Canvas.





Figur 1: A montage of 256 of the 430 training images in the HEP2 database (left) and a montage of 256 of the 5000 training images in the digit database (right).

**Supervision sessions.** There will be four opportunities get help with the assignment; Tuesday 3 November 13-15, Thursday 5 November 13-15, Tuesday 10 November 13-15 and, finally, Thursday 12 November 13-15. All sessions take place on zoom `https://lu-se.zoom.us/j/63860476526`. The assistants are Gabrielle Flood and Anna Gummeson.

## Datasets

Several data sets are provided for the assignment. Download all datasets and scripts from
`http://www.ctr.maths.lu.se/matematiklth/personal/kalle/MBA_ML_2019.zip`

**Simulated data for Kernel Estimation:** For the first part (Kernal Estimation) you will produce the datasets yourself.

**HEP2:** The HEP2 dataset consists of gray scale images. These have already been cropped and resampled to the same size ($64 \times 64$). There area also segmentation masks of the same size. There are 433 training images and 430 test images. The images are taken from several different examinations. One can perhaps expect the cell images from the same examination to be more similar to each other as compared to images between exainations. The cell images are of six different types and the ground truth label for all training and test images are provided. Figure 1 (left) illustrate the appearance of 256 of these images.

**Digit images:** The digit dataset is provided with the MATLAB installation. The dataset consists of gray scale images of digits (0-9). These have already been cropped and resampled to the same size ($28 \times 28$). There are 5000 training images and 5000 test images. Figure 1 (right) illustrate the appearance of 256 of these images.

# 1 Kernel estimation

In this assignment we are going to model a two-class problem with only one feature. We are modelling the feature of each class as a Gaussian with equal standard deviation $s_1 = s_2 = s_{model}$. We are modelling the means as $m_1 = 0, m_2 = 1$. Thus for larger parameters $s_{model}$ we are getting increasingly difficult classification problems. In the assignment we are going to study two cases, i e $s_{model} = 0.4$ and 4. For each of these two models, perform the following

- generate training data (10 points for each class),
- generate testing data (1000 points for each class),
- use (non-parametric) parzen window estimation on training data using gaussian kernels, with two different widths, i e $s_{parzen} = 0.02$ and 1.
- For each case plot $p(x|y = 1)$, $p(x|y = 2)$, $p(x), p(y = 1|x), p(y = 2|x)$, both using the true model and from your estimates.
- Use the estimated probability density functions $p(y = 1|x)$ to classify the data. This is the so called plug-in idea. What is the error rate for the test data?
- If we use the true model, i.e. if we somehow knew that $p(x|y = 1)$ and $p(x|y = 2)$ are Gaussian with known mean and standard deviation, what would the optimal threshold be? For this threshold, calculate the error rate.

I would like comments on

- the comparisons between the model densities and the estimated densities of both $p(x|y = 1)$ and $p(x|y = 2)$. Do this for $s_{model} = 0.4$ and for the two different kernel widths $s_{parzen}$. Does the error in the estimate affect, or not, $p(y = 1|x)$.
- the comparison for $s_{model} = 4$ between nearest neighbour and plug-in classifier when $s_{parzen}$ is low. You don't actually need to run the nearest neighbour classifier. I just want you to reflect on the similarities between how the nearest neighbour classifier works and what happens when $s_{parzen}$ is low.

Also for the simulated datasets above, use cross validation to estimate kernel width. What is the estimated kernel width?

> For the report: Plot the estimated densities $p(x|y = 1)$, $p(x|y = 2)$, $p(x), p(y = 1|x), p(y = 2|x)$ for each combination of two model standard deviations $s_{model}$ and two kernel widths $s_{parzen}$. Comment on the quality of the estimated densities. What happens when the parameter $s_{parzen}$ is too low and what happens when it is too high? Calculate estimated error probabilities for the two models and the classifier when using the two different kernel widths as well as with the optimal theoretical threshold. Finally explain why using the method using very small kernel widths become more like the nearest neighbour method. Show how cross-validation is used to estimate kernel width. (Minimum requirement for the grade 3 (pass).)

**Coding tips:** There are many useful functions for handling statistics in matlab, see e.g. `help stats`. There are functions in matlab for generating random data, e.g. `rand`, `randn`. To know more about a function in matlab use `help xxx`, e.g. `help randn` to get help for the function `randn`. There are functions for standard probability density functions, e.g. `normpdf` and for cumulative distribution functions `normcdf`. Other useful functions You are going to use kernel density estimation several times in the code. It is probably a good idea to to write a function, e.g. `my_parzen` that can calculate the estimated density. `function f = my_parzen(x,T,h)` which takes training data $T$ as input as well as width $h$ and points $x$ at which you would like to evalute the function. This makes coding and debugging easier.

## 2 Machine Learning for Cell Type Classification using hand-coded features

Download all datasets and scripts from
`http://www.ctr.maths.lu.se/matematiklth/personal/kalle/MBA_ML_2019.zip`

Here we are going to study the HEP2 dataset, which contains cell images. There are 430 training images and 433 test images of 6 classes. Unpack the zip-file to obtain a folder `MBA_ML_2019` with a few matlab-scripts and a subfolder `databases` that contain a few datasets to work with.

We have prepared a script `example_hep_ml_v0.m` that loads the data (training images `X1`, training labels `Y1`, test images `X2`, test labels `Y2`). One common approach is to manually design features. An example script that illustrates this is `get_features.m`. It takes one HEP2 image as input and outputs a vector with a few features. As a start we have chosen the mean and standard deviation of the pixel values in the image. In the script `example_hep_ml_v0.m` we use the feature extraction to form matrices $X1f$ for the training data and $X2f$ for the testing data. **Note that we in this script use matlabs notation to have each exampel as a row in matrix $X1f$. In the image analysis course we often had feature vectors as columns.** These are then used for classification using a couple of different machine learning methodologies. The script also evaluates the trained classifiers on both the training data and the test data.

Study the script `example_hep_ml_v0`. In the script we use `gscatter`. Read more about gscatter, `help gscatter`. It is a useful script for visualizing features. By modifying the script `get_features` you can add your own features. Remember then to also name your features for future reference.

For the assignment. Invent your own feature set and try a few different techniques for classification on the data. A random classifier should give roughly 17% correct on the data. Try to get as good performance as possible.

For the report: Describe what features you have experimented with? Visualize how the examples in the training set occupy different regions in the feature space. What machine learning methods did you try? What combination of features and method would you recommend? How good do you think this combination would be on future data? Write out in the report a table of the accuracy on the training set and on the test set for your method (or methods if you have tried several ones). Comment on the difference between the accuracy on training data and test data. (Minimum requirement for the grade 4.)

# 3 Classification using Deep Learning (CNN)

Here we are going to study a dataset of handwritten numbers and the HEP2 dataset, which contains cell images. You have already unpacked the zip-file to obtain a folder `MBA_ML_2019` with a few matlab-scripts and a subfolder `databases` that contain a few datasets to work with.

We have prepared a script `example_digits_cnn_not_so_deep_0.m` that (i) loads the digit datasets, (ii) defines a deep learning architecture, (iii) trains the network and (iv) tests the result on both the training and test datasets. Run the dataset and make sure that you understand how the four different steps are coded. The deep learning architecture is in fact no so deep. It is essentially a linear logistic regression. Try to change the code to make the architecture deeper. See the lecture notes for inspiration. Does this improve the result?

Next try to make a classifier for the HEP2 dataset that you used in the previous example. Does the trained classifier work generalize well to the test set. Try to augment the data in the training set and see if you can make a system that works better. Also try experimenting with changing the learning rate and the maximum number of epochs.

For the report: **digit dataset:** Describe what architecture you have experimented with for the digit dataset and write out a table of training and testing accuracy for the not-so-deep architecture and for your deeper architecture. **HEP2 dataset:** Describe what architecture you have experimented with for the HEP2 dataset and write out a table of training and testing accuracy for a couple of different architecture. Comment on the difference between the accuracy on training data and test data. Describe what kind of augmentation you used on the training data. Reflect on how augmentation improves (or does not improve) on how the network generalized to the test data? Comment on your choice of learning rate and maximum number of epochs. (Minimum requirement for the grade 5.)