# Introduction to Robotics

Coursework 3 (20%)

**Object Recognition**

Out: 28th November 2023 | In: 9th January 2024

**Task**

The coursework will use a subset of MNIST handwritten digit dataset (will be provided) that you have prepared in the lab sessions (see lab materials in Appendix), and you will perform the following tasks:

**Compulsory Section (You must complete this part)**

Design and implement a single layer perceptron to distinguish the digit 6 from the other 9 digits, i.e., when the input is 6, the output will be +1, and -1 otherwise. Do your experiment follow these guidelines:

- Split the data into a training set and a testing set.
- Plot at least 4 graphs (e.g., 2 different sets of initial weight, 2 different training rates) of error over time (%-incorrect classification on y axis, training epochs on x axis) for the training set.
- Plot the corresponding graphs for the testing set.

Explain the stopping criteria you used to terminate training.

**What to summit**

Submit an electronic copy (code and report). Your report which should include

- A brief explanation of your implementations
- Results and explanation of results
- Conclusions and reflections (or what you've learned about object recognition through doing this coursework)

**A demo**

You must demonstrate your implementation to the TAs.

**Optional Section (You don't have to do this part, but are strongly encouraged to have a go)**

Using as many Perceptrons as necessary, design a pattern recognition system that will put the digits into 4 groups: (0, 1) (2, 3,) (4, 5, 6) (7, 8, 9). Explain your implementation and results.

# Machine Learning Lab1

Throughout the lab sessions, we will continuously do experiments on the MNIST digits dataset. The ultimate goal is to build a digit recognition system using Matlab.

## Task 1: Getting started with Matlab

Matlab, which is short for **Mat**rix **Lab**oratory, is probably the most widely used scientific and engineering numerical software. To get started, you can either watch the video from:

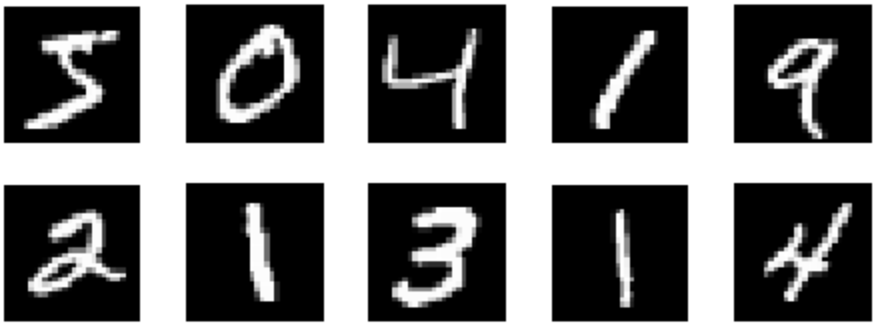
http://www.mathworks.co.uk/products/featured/videos/index.html

or read the book from:

http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf

## Task 2: Load and display the MNIST dataset

1. MNIST digits recognition dataset is one of the most widely used datasets in machine learning. It contains 60,000 training samples and 10,000 test samples. You can download it from http://yann.lecun.com/exdb/mnist/.
2. Please carefully read the instructions therein about the structure of how the data are stored. Try to load the data to Matlab and display them. The first 10 digits in the training set should be like this:



## Tips:

1. In loading the MNIST data, probably you will rely on the following functions in Matlab: fopen, fread and fclose.
2. In displaying the data, you will need to use the following functions: reshape, imagesc and imshow.
3. Just type ‘help XXX‘ in matlab to see how to use the above functions.
4. If you encounter the ‘out of memory’ problem in Matlab, which means the data is too large to fit in to the memory, and you have to delete unnecessary variables in the workspace or save them to the hard disk.

# Machine Learning Lab2

## Data preprocessing:

1. Firstly, please repeat lab1 to load the MNIST dataset to Matlab. As the original MNIST dataset contains 60000 digits for training, which is too large for our lab sessions. Therefore, we only use a subset of the first 1000 digits for training. Based on the lab1 sample code, you can achieve this by using the following code:

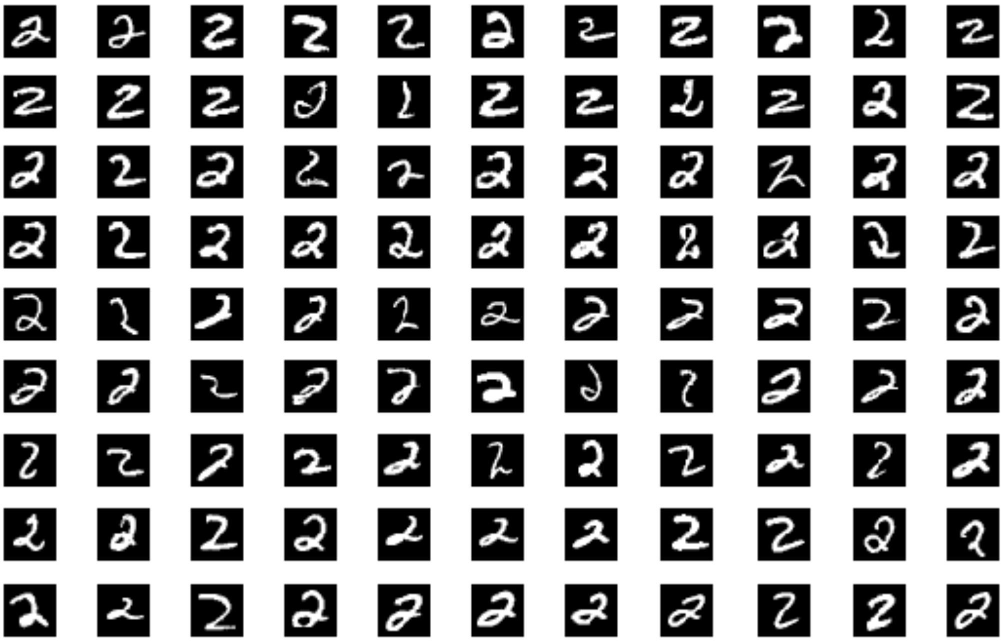
```
tr_label= tr_label(1:1000,1);
tr_feats= tr_feats(1:1000,:);
```

   To avoid repeating the above procedure in the future labs, you can directly download this small set from http://www.cs.nott.ac.uk/~qiu/Teaching/G53MLE/Labs/MNIST_subset.mat

2. To get a general picture of the data that we are going to process, you can count how many samples are contained in the training set for each digit. You can use the following code:

```
for i=0:9
   num_of_sample(i+1)=size(find(tr_label==i),1);
end
```

   If you look into the variable `num_of_sample,` you could see that the number of samples for each digit is roughly the same. You can also try to display all the samples for each digit contained in the training set.



3. It is sometimes necessary to normalize the data to make them more comparable. The aim of normalization is to make the Euclidean norm http://en.wikipedia.org/wiki/Euclidean_norm#Euclidean_norm of each digit equal to 1. You can judge by yourself in the future lab sessions that whether this normalization step is necessary or not.

# Machine Learning Lab3

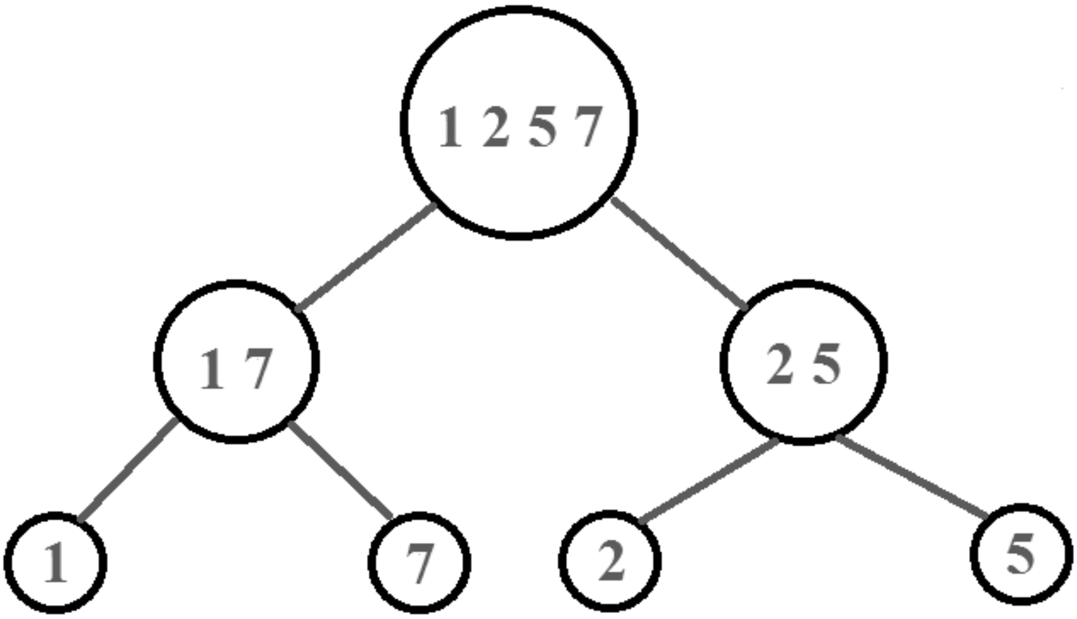
## Designing a Perceptron:

Then it is your job to implement the perceptron algorithm. Please try to encapsulate your algorithm as a function. A possible form of the function can take two input arguments, one is the feature vector (tr_feats) and the other is the label (tr_label), and it will output the learned weights:

```
function weight=Perceptron(feats,label)
```

After you have implemented the perceptron function, try to finish the following tasks:

1. Treat digit ‘2’ as positive, digit ‘5’ as negative, learn a percetron. Try to visualize the samples that are most likely to be ‘2’ or most likely to be‘5’. Also try to visualize the wrongly classified samples.
2. Now consider four digits, ‘2’, ‘5’ ‘1’, and ‘7’. If we consider ‘2’ and ‘5’ are more visually alike, and the same for ‘1’ and ‘7’, then we could design a ‘parent’ perceptron that consider ‘2’ and ‘5’ to be positive, ‘1’ and ‘7’ to be negative. We can also design two ‘child’ perceptrons, one for classifying ‘2’ against ‘5’, and the other distinguishes ‘1’ from ‘7’. Therefore, in total we need three perceptrons to classify these four digits. Do you have some other ideas? Try to implement them and compare the results.



3. Now consider all the 10 digits. Please think by yourself that how many perceptrons do you need to recognize them. Then implement it!