EFME 2013 LU Exercise 3

Exercise due: 19th of January 2014 at 23:55

Abstract

The goal of this exercise is to give you an introduction to the Perceptron, and applying pattern recognition techniques on a real world problem. The extra exercise gives you an introduction into Multilayer Neural Networks.

1 How to submit your report

PLEASE NOTE: you are going to deliver TWO reports; the first one reporting on the results of the perceptron and the second one is a scientific report (read below).Read this document carefully till the end, before starting. For all the questions you should contact the teaching tutors in the Lab. You should make the report of the exercise available as a PDF document. The hand-in for this and later laboratory exercises is done by using TUWEL before the deadline (see above) as following (please follow this standard):

- The name of the file: 'EFME-', your group and exercise number, e.g. 'EFME-Gr0-Ex0' for the zero group and zero exercise.
- It should be a compressed file e.g. .7z or .zip, containing the PDF document with the ALL the MATLAB code necessary to RUN your solution (including your chosen images, and an instruction file readme.txt to run your code or better include a MATLAB main file that runs the whole excercise.

You may write the PDF document in English or in German. Include in the document your results and most importantly, a discussion of the results. DO NOT forget to attach the MATLAB code in the same zip-file. Be careful: If the attached MATLAB does NOT run, we will reject your exercise completely. It is NOT necessary to include a copy of all the code in the PDF document, although key parts necessary to explain a point can be included. It is necessary to comment the code in details. More details are available on the TUWEL web page.

The first report.

2 Peceptron (5 points)

Write a MATLAB function `[w] = perco(X,t,maxEpoches)`, which calculates the perceptron weight vector `w` for the training set `(X; t)` using the training method described in the lectures. The training set should be in the format described in the lectures (the ith column `X(:,i)` contains the ith training example and `t(i)` contains the ith target (class label), the latter using `+1/1` coding). In `maxEpoches`, one gives the maximal permissible number of training epochs.

2.1 Discussion

Apply the perceptron to the following problems: OR, AND and XOR. Do not forget to convert your training vectors into homogeneous coordinates. Discuss the results and the convergence of the algorithm.

2.2 Application

Train your perceptron on the input vectors in the data file `perceptrondata` with the targets (class labels) `perceptrontarget1` and `perceptrontarget2`. Plot the class membership of the training points for both sets of class labels.

Hints:

- The data can be read in using the MATLAB function load, e.g. `inputs = load('perceptrondata');`. Note that the training examples and the targets are saved as rows.
- The targets are encoded as `0/1`. This is as required by the MATLAB neural network toolbox. However, to use your perceptron code, you should convert the targets to `-1/+1` encoding.
- To separate the training set into positive and negative points, it is useful to use logical indices. For example, the row vector `t` holds the class label of the ith pattern `X(:,i)`. By using `logind = (t == 1)`, one obtains a logical index which chooses all vectors in `X` having a target of 1, e.g. `patterns1 = X(:,logind)`.

Second report

the part below should be discussed in the scientific report

3 Practical Application (20 points)

The aim of this exercise is to gain some experience in using pattern recognition algorithms on a real problem of classification of strokes in paintings.

So far, you have programmed the following pattern recognition algorithms:

- Perceptron.
- Mahalanobis Distance classifier.
- k-NN classifier.

Choose the datasets described in Section 3.1.2, train these three classifiers on it. You are going to compare their performance. Remember to divide the dataset into a training set and a test set, and use the same training and test sets for all the classifiers, such that the comparison is fair. Also consider using fewer features. If you like for an extra **5 bonus points**, you can also try out a neural network (see Section 4 for details). Have a look it is not so hard, and you get some extra points.

3.1 Stroke Classification

3.1.1 Introduction

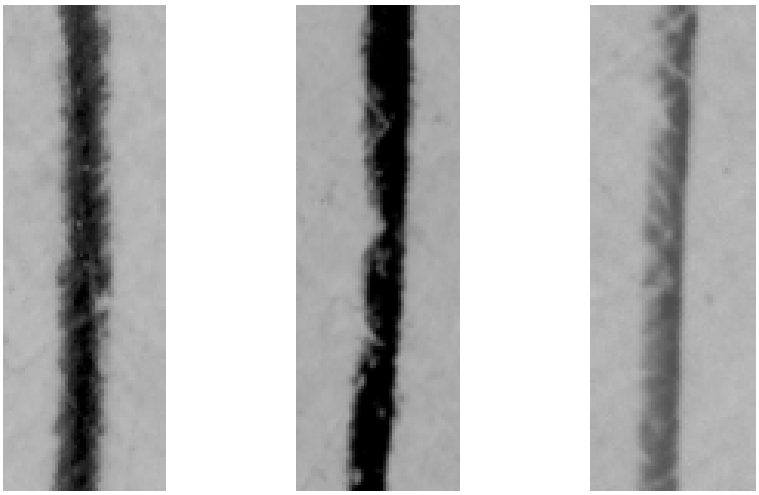
The main objective of this exercise is to classify the type of strokes used during creation of ancient paintings. For determining the stroke type, the paintings are photographed using infrared light and a special camera is used to capture the infrared reflectograms at different wavelengths. The acquired infrared reflectograms are preprocessed and clustered to a feature vector using wavelets, as proposed by Porter et al. in [PC96]. Figure 1 shows six types of strokes, where the first three of them (Figure 1a - 1c) can be summarized as *"dry strokes"*, while the remaining three (Figure 1d - 1f) are referred to as *"wet strokes"*. For further information related to stroke classification see [LKS04].

3.1.2 Data set

Use the data set given in the file `strokefeatures.mat` that contains the strokes to be classified. To load them into your Matlab workspace, use

```
load strokefeatures.mat;
```

either at the command window or within your code. `strokefeatures.mat` is a matrix consisting of 21 columns and 155 rows. Columns 1 - 10 contain the mean of the reflected infrared light at the corresponding wavelength. Columns 11 - 20 contain the standard deviation of columns 1 - 10 respectively. Column 21 contains the class of the strokes, where



(a) Black lead (b) Black chalk (c) Silver point

(d) Paint brush (e) Reed pen (f) Goose quill

Figure 1: Infrared reflectograms of six different types of strokes. The upper three of them are categorized as *"dry strokes"*, while the lower three are *"wet strokes"*.

- Black lead = 1,
- Black chalk = 2,
- Paint brush = 3,
- Reed pen = 4,
- Goose quill = 5 and
- Silver point = 6.

Use the classifier that you programmed and

1. (first classification problem) distinguish between two classes: *"dry strokes"* and *"wet strokes"*,
2. (second classification problem) now try to classify each class of stroke separately,

When you report on k-NN please provide the classification errors for $k = 1$ (Nearest Neighbour classifier) as well as for at least one other values of k. When you report on Mahalanobis classifier you could (but not necessary) choose one of the simplified models. Please report clearly what you have used.

3.2 Structure of the scientific report

Prepare a scientific report comparing the performance of the three classification algorithms on your chosen dataset on the two problems described above. Include graphs showing the different classification rates. Discuss and interpret the results. Can you think of good reasons for the different results produced by the different classifiers? Which combination of the 10 reflected infrared light wavelengths (column 1-10 at file `strokefeatures.mat`) leads to the best results? Does using fewer features improve the performance? What is a good way to chose the best features to keep? Is you classifier good enough to be used in real life? Your discussion should be precise and scientific. The report is the most important part of this exercise, as the MATLAB code for the algorithms used has already been marked in previous exercises. However, you must include all MATLAB script files necessary to produce the results given in the report. The scientific report should be around 10 pages long (longer is also okay) including images, tables and graphs.

4 Neural Network optional exercise (5 bonus points)

If you have time and energy and want some bonus points, learn how to use the MATLAB Neural Network Toolbox, implement a Back-propagation Neural Network and apply it to the same database as the other classifiers. Compare and interpret the results. Links to documentation on the Neural Network toolbox are given on the TUWEL web-page.

5 What have you learned

In the scientific report you should be able to discuss intelligently on the following:

- choosing a classifier for a problem.
- usage and testing of classifiers.
- feature selection, and
- discuss the results.

References

[LKS04] Martin Lettner, Paul Kammerer, and Robert Sablatnig. Texture Analysis of Painted Strokes. Technical Report PRIP-TR-089, Pattern Recognition and Image Processing Group (PRIP), Vienna University of Technology, June 2004.

[PC96] Robert Porter and Nishan Canagarajah. A Robust Automatic Clustering Scheme for Image Segmentation Using Wavelets. *IEEE Transactions on Image Processing*, 5(4):662–665, April 1996.