Final project of Artificial Neural Networks (H02C4A)

Academic Year

2016-2017

1 General Guidelines

The final project consist of solving 2 problems:

1. nonlinear regression and classification with MLPs
2. character recognition with Hopfield network

Each problem uses an individual dataset, which depends on your name and your student number.

Each student has to write one report for each problem describing the used methodology, solution and results (describe and discuss the algorithms used and the different alternatives, the problems encountered, etc, add in an appendix the Matlab routines used). Each report should not exceed 4 pages (including text and figures, but not the appendix).

On each report, please state clearly:

- your name, student number and program (MAI, Bio-I, LI, Erasmus, etc.)

The reports of the final project and 4 other exercise sessions need to be delivered by Friday 2 June before 14:00 at the ESAT department reception. There will be a box where you can drop your report. In case you would be unable to deliver a hardcopy, please send the pdf version (together with a motivation) to Ida Tassens (ida.tassens@esat.kuleuven.be). Note that it is allowed to bring further updates on your report to the exam.

2 Problem 1: nonlinear regression and classification with MLPs

2.1 Regression

2.1.1 Problem Description and data preparation

In this problem, the objective is to approximate a nonlinear function using a feedforward artificial neural network. The nonlinear function is unknown, but you are given a set of 13 600 datapoints uniformly sampled from it.

You have to build your individual dataset from 5 existing nonlinear functions $f_1, f_2, \ldots, f_5$. In the given matlab datafile, you will find the following variables: $X1, X2, T1, T2, T3, T4$ and $T5$. The vectors $X1$ and $X2$ contain the input variables (in the domain $[0,1] \times [0,1]$). The vectors $T1$ to $T5$ are the 5 independent nonlinear functions evaluated at the corresponding point from $(X1, X2)$. In other words, $f_1(X1(i), X2(i)) = T1(i)$, $f_2(X1(i), X2(i)) = T2(i)$, …, $f_5(X1(i), X2(i)) = T5(i)$, for $i = 1, \ldots, N$, where N is the length of the vectors mentioned above. The datapoints are noise free (the evaluation is exact).

You have to build a new target $Tnew$, which represents an individual nonlinear function to be approximated by your neural network. For this, consider the largest 5 digits of your student number in descending order, represented by d_1, d_2, d_3, d_4, d_5 (with d_1 the largest digit). You have to build your individual target as follows:

$Tnew = (d_1T1 + d_2T2 + d_3T3 + d_4T4 + d_5T5)/(d_1 + d_2 + d_3 + d_4 + d_5)$.

For example, if your student number is m0224908, then your list of largest digits in descending order is 9, 8, 4, 2, 2 and therefore your target is:

$Tnew = (9T1 + 8T2 + 4T3 + 2T4 + 2T5)/(9 + 8 + 4 + 2 + 2)$.

2.1.2 Exercises

The problem can be split into three tasks:

- Define your datasets: your dataset consists now of $X1, X2$ and $Tnew$. Draw 3 (independent) samples of 1 000 points each. Use them as the training set, validation set, and test set, respectively. Motivate the choice of the datasets. Plot the surface of your training set.
- Build and train your feedforward Neural Network: use the training and validation sets. Build the ANN with 2 inputs and 1 output. Select a suitable model for the problem (number of hidden layers, number of neurons on each hidden layer). Select the learning algorithm and the transfer function that may work best for this problem. Motivate your decisions. When you try different networks, clearly explain at the end which one you would select as the best for this problem and why.
- Performance Assessment: evaluate the performance of your selected network on the test set. Plot the surface of the test set and the approximation given by the network. Plot the error level curves. Compute the Mean Squared Error on the test set. Comment on the results and compare with the training performance. What else could you do to improve the performance of your network?

2.2 Classification

In this problem, the goal is to build feedforward neural networks for classification. Moreover, it is also required to explore the benefits of using dimensionality reduction techniques.

2.2.1 Problem description and data preparation

Here you will experiment on a real-life dataset related to the classification of red and white wine quality (for more details see [1]). In the original problem there are 10 different classes. In this case, each student will be assigned an individual dataset, in order to distinguish between 2 classes denoted as C_+ and C_- in Table 1:

- download the data from Toledo
- you have to use an individual dataset, depending on the last digit of your student number. To know how to construct your dataset, see Table 1. For instance, if your student number is m022490**1**, you have to solve the binary classification problem of class 6 vs. class 7, white wine.

Digit	C_+	C_-	Color
0	5	6	
1	6	7	
2	5	7	White
3	6	7+8	
4	4	5+6	
5	5	6	
6	5	6+7	
7	4+5	6	Red
8	3+4+5	6	
9	5	6+7+8	

Table 1: **Individual dataset assignment.** Digit refers to the last digit of your student number, C_+ represents the first class of your binary classification problem and C_- the second class.

2.2.2 Exercises

The problem can be split into three tasks:

- Create and train a feedforward neural network. You will need to decide the network architecture, select the transfer functions, and the learning parameters. Justify your choices. Train your network with the training data set. Calculate the correct classification ratio (CCR) for the validation set. The CCR is defined as: CCR = Number of Correctly classified data x 100 / Total number of data.
- Compute the eigenvectors and eigenvalues of the covariance matrix of the training set. Plot the eigenvalue spectrum. Select a number of Principal Components to perform dimensionality reduction via PCA. Motivate your decision. Reconstruct all your training, validation and test data using the selected principal components. In this way you obtain a new dataset with a lower dimension than the original data.
- Using the reconstructed dataset design a new feedforward network to classify the data. Again, train this new network to obtain the highest CCR on the new validation data. Train different networks, and finally select the network that you think is the best for this problem. Compare both networks, the one trained using the original data and the one trained using the lower dimensional data.

3 Problem 2: character recognition with Hopfield networks

3.1 Problem description and data preparation

In this problem you have to investigate the retrieval-capabilities of the Hopfield network, applied to the letters of the alphabet. You can use the script `prprob` to help you to build the letters. This script provides the letters of the alphabet in capitals as 7x5 black-white pixel maps.

You have to use an individual collection of characters. Your collection consists of all the letter of the alphabet in CAPITAL characters, together with all the letters of your name, in LOWERCASE characters.

Create your individual collection of characters by pre-pending the lowercase characters of your first and last name to the set of all capital characters (so that the lowercase characters come first). For example, if your name is John Doe, your collection of characters is the sequence: j,o,h,n,d,e,A,B,C,D,E,F,G,...etc.

You have to transform the pixels from 0s and 1s into -1s and +1s in order to train a Hopfield network with them. Display at least the first 10 characters of your set.

3.2 Exercises

You have to solve three tasks:

- Build a Hopfield neural network that is capable of retrieving the first 5 of these characters. Distort each character by inverting three randomly chosen pixels (so you change 1 to -1 and viceversa) and check if the network is able to recall these distorted patterns. Discuss the existence of spurious patterns.
- In this part you must determine the critical loading capacity of the network. Choose a number P, and store P characters from your collection in a Hopfield neural network. Distort these characters by inverting three randomly chosen pixels and try to retrieve them. Do this by iterating the network for a chosen number of iterations and clipping (rounding) the pixels of the final state to +1 and -1. Calculate the error for several values of P and plot this error as a function of P. The error is the total number of wrong pixels over all the retrieved characters. Estimate the critical loading capacity. Compare your estimated capacity to the theoretical loading capacity of the Hopfield neural network that uses the Hebb-rule for uncorrelated patterns. Discuss the existence of spurious patterns.
- What options would you have when you need to store 25 characters with the matlab routines in such a way that characters with three random inversions can be retrieved perfectly? Demonstrate a method using the first 25 characters of your collection as a basis.

References

[1] http://archive.ics.uci.edu/ml/datasets/Wine+Quality