Assignments Neural Networks Course

VU University Amsterdam

2015-2016†

† Part of these assignments are based on my predecessors: Wojtek Kowalczyk and Zoltan Szlavik

Assignment 1: Perceptron

During the first lecture of the course, you have learned how a simple perceptron is specified, and also the perceptron learning rule has been treated. Your first task is to use Matlab to implement such a perceptron, including its learning algorithm. After this, you are going to apply this perceptron to two classification problems. One dataset which you define yourself, and a second dataset consisting of a dataset provided to you (called two_class_example_not_separable.dat). You should deliver all the scripts you used as well as a written report with the results. Make sure that you identify your group in each document.

(a)

Before you start to implement the perceptron learning rule in Matlab you start to define a dataset (which is also very convenient to test whether your algorithm is working properly). The dataset you should generate should contain pairs of (x,y)-coordinates (the inputs for the network) and a class to which they belong, which can either have the value "-1" or "1". A total of a thousand points should be present. A requirement for the dataset is that it should be *linearly separable*. In you Matlab specification, also include the ability to plot the (x,y) pairs and their class.

(b)

The next step is to implement a perceptron with signum function (in Matlab this function is called "sign"), hereby closely look at the algorithm presented in the book on page 84 (which can be seen on the slides as well). Make the parameters of the algorithm easily adjustable (i.e. learning rate and the number of iterations).

(c)

Test the algorithm on the dataset you have produced, is it able to classify all cases correctly? Use a standard error measurement method to determine the error produced by the network. Make a plot of the decision boundary that has been found by the network. Furthermore, also express in a graph what the influence of the learning rate upon the overall error produced by the network.

(d)

Repeat the experiments as expressed under (c) for the dataset provided called two_class_example_not_separable.dat. Be careful with the class values that have been assigned (do they match your algorithm?). Describe the same aspects of the behaviour of the algorithm. Does it show "bad" behaviour for this dataset?

Assignment 2: Statistical Learning

As discussed during the lecture, statistics based approach and neural network go very much hand in hand. The goal of this assignment is to use Matlab to perform some statistical operations and get more in depth knowledge of applying the statistical rules. For the second part of this assignment (application of the least mean squares approach) you will use the Netlab environment, which is a Matlab package dedicated to Neural Networks as well as statistical methods. Again, deliver all scripts you used as well as a document containing your answers and the accompanying discussion.

(a)

First, we will focus on the example of classification of a character based upon the height/width ratio again. Create a dataset that expresses the relationship between the height/weight feature and the class (i.e. express the table shown with the distinct columns for xl in the form of a matrix). You are asked to produce a script that: (1) calculates all relevant probabilities (priors, class-conditionals, posteriors); (2) generates the two histogram plots shown on the slide upon the data obtained under (1), and (3) calculate for all possible splits the number of misclassified cases (using decision rule: X < split >= C1 otherwise C2). What is the best decision boundary? Assume that it costs a lot more to misclassify a b as an a than it does to classify an a as a b (a cost of 2 vs. 1), what is now the optimal decision boundary?

(b)

We are now going to return to the two datasets you used during the previous assignment (i.e. the one you generated yourself as well as the dataset provided to you). Apply the least mean squares (LMS) approach (you can use the Netlab glm classes for this purpose). Use the same approach as you have previously used to calculate the error in order to make the results comparable. Show the decision boundary that has been found, and make a comparison between the results you have obtained using the LMS approach and the perceptron. Also discuss the influence of the parameter settings upon the behaviour of the algorithm.

Assignment 3: Multi-Layer Perceptrons

In this assignment you will work with a more complex form of a neural network, namely a multi-layer perceptron. In Netlab the associated functions are called mlp. Hereby, multiple learning algorithms, including the backpropagation algorithm that has been discussed during the lecture are available. In this assignment you are going to look at a regression task as well as a classification task. Again, deliver a report as well as all the scripts you have used to produce your results.

(a)

Use a multi-layer perceptron network to perform a regression task. 3 datasets are provided: line.mat, sinus.mat and irregular.mat. Each of these datasets consists of two matrices: an input matrix with input patterns (one pattern per row) and a value. Use Netlab to develop an MLP network for this regression task, hereby try various numbers of hidden nodes (in particular, try 2,3,4,5,7,10,15,25) and use a 5-fold cross validation to measure the accuracy of your network, whereby you use the Root Mean Squared Error as a measure of the network accuracy. Describe the results you obtain, and plot the decision boundaries found. Start with using the standard backpropagation algorithm, does this work well for all datasets? Furthermore, discuss how the number of hidden neurons influences the accuracy on the train as well as the test set. Now apply the Matlab standard polyfit algorithms (polyfit) which tries to fit polynomials to a dataset, hereby use the cross validation approach again. How does the accuracy compare with the MLP network? How does the degree influence the results?

(b)

Now we are going to use an MLP network for a binary classification task, namely to recognize whether people are wearing glasses or not. In the file pics.mat you can find a number of pictures of people that are either wearing glasses or not. The vector pics expressed the picture itself in the form of a vector of graysclae values between 0 and 256. The length of the vector is 2576 which represents the pixels of the 56x46 image. To display an image, you can use the command:

imagesc(reshape(pics(200,:),56,46))

Take the dataset and develop a MLP network with as an input the number of pixels of the image and as output a single node indicating whether the person is wearing glasses or not. Use 10-fold cross validation to measure the accuracy (can you just choose the folds randomly?). Should you just use the pixels of the input as is? If you want, you can also use a filter to detect features in the image, for example a Gabor filter. What is the accuracy you can achieve? What is the influence of increasing/decreasing the number of hidden neurons? Produce the confusion matrix for various settings and discuss the results.

Assignment 4: RBF networks and Support Vector Machines

In this assignment you are going to apply Radial Basis Function networks as well as Support Vector Machines (SVM's) to the problem of face recognition that we have already seen., namely to detect whether people are wearing glasses. Hereby, a functionality is present in Netlab for RBF networks (.m files starting with rbf). Unfortunately, this is not the case for SVM's. Therefore, you are asked to use another Matlab toolkit for that purpose, namely from Leuven: http://www.esat.kuleuven.ac.be/sista/lssvmlab/. Again, deliver a report in which you discuss the results are well as all the underlying scripts.

(a)

Instead of using a MLP network, we are now going to use an RBF network for classification of the images of people wearing glasses you have previously worked with. Hereby, apply the same filtering techniques (if any) as you have used before for the MLP network. Use different number of kernels (i.e. hidden neurons). Make a comparison between the performance using various numbers for k thereby utilizing 10-fold cross validation, use a confusion matrix. Discuss the differences between the various settings as well as the difference with the MLP approach.

(b)

In order to get accustomed with the SVM toolkit, apply SVM to the simple classification dataset you have composed yourself in assignment 1a. The toolkit offers the possibility of automated transformation of the input data, can you see any influence of using this transformation technique? And what can you say about the decision boundary which has been found?

(c)

Apply SVM in a similar manner as discussed under 4a. Include a discussion on the differences between the two approaches.

Assignment 5: Principal Component Analysis

Principal component analysis (PCA) is frequently used to compress images. In this assignment, this is precisely the purpose. Apply PCA to the image dataset you have already obtained. Netlab includes pca facilities. Submit a report with your findings as well as all scripts used.

(a)

Use the images that you used in assignment again (i.e. pics.mat). Use the PCA approach and use different numbers of principal components (varying between the minimum 1 and maximum 2576) to find a suitable encoder en decoder for the images. Define a distance measure between the original and image after being encoded and decoded again. Describe how the number of principal components influences this distance between the original and the image resulting after encoding and decoding. Furthermore, show some example original images versus the restored images after respectively encoding and decoding.

Assignment 6: Self-Organizing Maps

The final technique you are going to explore concerns self-organizing maps (again present in the Netlab package). You are going to investigate the organizing behaviour of these maps using a simple dataset that you generate yourself. Submit a report with your findings as well as all scripts used.

(a)

Generate a thousand (x,y) points in a non evenly distributed manner (!) on a grid. Apply a two-dimensional SOM using Netlab and plot the resulting lattice. Hereby, produce a plot similar to the one presented in the book on page 474 (also shown on the slides during the lecture). Try experimenting with different sizes for the lettice. What is the influence of thereof upon the outcome? And what is the influence of the learning rate upon the outcome?

Final assignment (7)

Panic on the 4th of May 2010

As you probably know, on the 4th of May 2010 during the national remembrance of the people that died during wars or peacekeeping missions a large panic occurred due to a person that started to shout during a two minute period of silence, see for instance http://www.youtube.com/watch?v=bATBjy85sHI . In order to be able to cope better with such occurrences, there is a great need for models that are able to predict the behaviour of people in such a crowd in case of a panic situation. Various models have been developed for this purpose that for instance rely on particle physics or theories from psychology. However, once data is present about the movements during an actual event (such a May 4th) developing a Neural Network to make such predictions also becomes feasible. This is precisely what you are going to do in this assignment. A dataset has been developed by researchers at the VU University (see JAAMAS paper) which represents the coordinates of 30 people in the crowd on May 4th over time. These people have been selected because they are representative for the movement of a whole group of people. You should produce a single neural network that can replicate the behaviour as described in the dataset. You can visualize their movements using the Matlab code expressed in start_final_assignment.m.

Step 1.

Determine the inputs and outputs of the neural network you are going to develop, and write the scripts needed to transform the original data. You can for instance think of using the distance to the source of the panic as input, or the angle, but feel free to choose any inputs you like. Of course, you can also apply the approaches you have learned during the lecture (e.g. subtract mean etcetera) to improve the performance of the network.

Step 2.

Enrich the dataset of the 30 people (as this is a relatively small dataset for machine learning) by generating more data based upon these representative movements, e.g. generate data for people that are standing close to the 30 people you have data of and assume that they show similar patterns of movement.

Step 3.

Define an error measure for the performance of the neural network. Use 10-fold cross validation.

Step 4.

Define the architecture of the network and the learning algorithm. You are free to choose any method you have seen during the course and find suitable. Experiment with different parameters and eventually discuss their influence (e.g. number of hidden neurons, learning rate, etcetera). Let the algorithms run and analyse the results.

Step 5.

Write a report discussing all you experiences during steps 1-3. Please clearly state the rationale behind important choices. Also deliver all the accompanying scripts that you have used to produce your results.