**CM0669 Machine Learning and Computer Vision**

**Lab 3** Linear and Non-linear Neural Network classification/regression

**1. Classification with Linear Perceptron**

Download the matlab codes ‘Perceptron\_SL\_SW\_out\_SE\_VERS.m’, ‘Perceptron\_PL\_PW\_out\_SE\_VERS.m’, ‘Perceptron\_SL\_SW\_PL\_PW\_out\_SE\_VERS.m’, ‘Perceptron\_SL\_SW\_out\_VERS\_VIRG.m’, ‘Perceptron\_PL\_PW\_out\_VERS\_VIRG.m’, ‘Perceptron\_SL\_SW\_PL\_PW\_out\_VERS\_VIRG.m’ in a folder ‘Week3’.

Each Matlab code implements a linear Perceptron classifier for the training. However, as you have seen in Lab1 and 2, they differ in terms of the input and output used. For instance, ‘Perceptron\_SL\_SW\_out\_SE\_VERS.m’ is a linear perceptron classifier using as input **S**epal **L**ength and **S**epal **W**idth and gives as output the ‘**SE**tosa’ and ‘**VERS**icolor’ class labels (See figure below)



Similar to what you have seen in Lab1 and Lab2 for the evaluation of each classifier, 80 samples (specimens) have been used for the training while the testing is performed on 20 samples.

1. Open up Matlab and use “help newp” command in the command window. A helpful description of the built-in function ‘newp’ will be given with good examples. ‘hardlim’ represents binary step function used as the default transfer function for the perceptron.
2. Open up each Matlab code using the Matlab workspace and ensure you understand its content (note that each Matlab code is well commented.).
3. Run the codes and complete the results in the table given below.
4. Compare and analyse the results.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Linear Perceptron | | | | | | | | |
| Input | | | | Output | | | Evaluation | |
| Sepal  Length | Sepal  Width | Petal  Length | Petal  Width | Setosa | Versicolor | Virginica | Training Error (%) | Testing  Error (%) |
| Yes | Yes | No | No | Yes | Yes | No | 0 | 5 |
| No | No | Yes | Yes | Yes | Yes | No | 0 | 0 |
| Yes | Yes | Yes | Yes | Yes | Yes | No | 0 | 5 |
| Yes | Yes | No | No | No | Yes | Yes | 50 | 50 |
| No | No | Yes | Yes | No | Yes | Yes | 5 | 5 |
| Yes | Yes | Yes | Yes | No | Yes | Yes | 31.25 | 5 |

**2. Non-linear Neural network classification**

Neural networks are very good at pattern recognition problems. Compared to the perceptron, a neural network with enough elements (neurons) can classify any data with better accuracy. Neural networks are particularly well suited for complex decision boundary problems over many variables. In a feed forward neural network structure, layers which are not output layers are called hidden layers.

The Matlab codes whose name starting with ANN instead of Perceptron are classifiers using the Neural network architecture with 2 hidden layers. There are 10 neurons in the first hidden layer with ‘Tangent Sigmoid’ as transfer function. The second hidden layer has five neurons with ‘Tangent Sigmoid’ as transfer function. The last layer (output layer) is a single neuron with a linear transfer function but the output is rounded to be either (0 or 1) for appropriate classification (See lecture 3). The figure below illustrates the structure of the network in the case of two input features (Sepal length and Sepal Width)



1. Use “help newpr” in the command window. A helpful description of the built-in function ‘newpr’ will be given with good examples.
2. Open up each Matlab code (starting with ANN) using the Matlab workspace and ensure you understand its content (note that each Matlab code is well commented.).
3. Run the codes and complete the results in the table given below.
4. Interpret the results and compare them against the previous ones (linear perceptron).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Non-linear Neural network | | | | | | | | |
| Input | | | | Output | | | Evaluation | |
| Sepal  Length | Sepal  Width | Petal  Length | Petal  Width | Setosa | Versicolor | Virginica | Training Error (%) | Testing  Error (%) |
| Yes | Yes | No | No | Yes | Yes | No | 0 | 50 |
| No | No | Yes | Yes | Yes | Yes | No | 0 | 0 |
| Yes | Yes | Yes | Yes | Yes | Yes | No | 0 | 0 |
| Yes | Yes | No | No | No | Yes | Yes | 63.75 | 45 |
| No | No | Yes | Yes | No | Yes | Yes | 2.50 | 40 |
| Yes | Yes | Yes | Yes | No | Yes | Yes | 5 | 50 |

**3. Linear and Non-linear Neural network regression**

In ichthyology, the age of a species of fish is to be represented as a function of the length and water temperature. In order to design an automated system with a neural network that can predict the age of fish according to its length and the water temperature where it lives, the fish are kept in tanks at 25, 27, 29 and 31 degrees Celsius. After birth, a test specimen is chosen at random every 14 days and its length is measured. The measurements are recorded in a table which will then be used to train the system (see first 6 rows of the table). The training data set is saved to a file ‘fish\_training.mat’. It consists of 36 samples (i.e. 36 rows).

|  |  |  |
| --- | --- | --- |
| Temperature  (degrees Celcius) | Length (mm) | Age (days) |
| 31 | 326 | 139 |
| 27 | 122 | 28 |
| 25 | 260 | 55 |
| 25 | 212 | 41 |
| 31 | 214 | 55 |
| 29 | 452 | 111 |

To evaluate the performance of the system, a test data set consisting of 8 samples is used. This is saved in another file ‘fish\_testing.mat’. Two Matlab programmes are provided. ‘Linear\_regression\_neuron.m’ is an implementation of a regression system using a single linear neuron similar to what you have seen in lecture 1. ‘Neural\_network\_regression.m’ is the implementation of a neural network regression system. The structure of the network is the same as the one given in section 2 (previous section) but the transfer function of the neurons in the hidden layers is log-sigmoid instead.

1. Use “help newff” in the command window. A helpful description of the built-in function ‘newff’ will be given with good examples.
2. Open up each Matlab code and ensure you understand its content (note that each Matlab code is well commented.).
3. Run the codes and complete the results in the table given below.
4. Interpret the results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Temperature  (degrees Celsius) | Length (mm) | Actual age  (days) | Estimated age with linear regression | Estimated age with ANN |
| 25 | 394 | 97 | 102 | 95 |
| 25 | 447 | 111 | 119 | 128 |
| 27 | 326 | 69 | 89 | 64 |
| 27 | 402 | 83 | 113 | 65 |
| 29 | 214 | 41 | 61 | 51 |
| 29 | 289 | 55 | 85 | 76 |
| 31 | 271 | 69 | 87 | 76 |
| 31 | 302 | 83 | 97 | 105 |

Add a piece of Matlab code to calculate the testing regression error which is given by:

Testing\_error = ; for all test samples *i=1, 2, ..., N.* (*N*=8).