UNIVERSITY OF LIVERPOOL

Department of Electrical Engineering and Electronics

ELEC320

Assignment on Multi-Layer Perceptrons

Module	ELEC320
Coursework name	Assignment on Multi-Layer Perceptrons
Component weight	30%
Semester	2
HE Level	5
Work	Individual
Assessment method	Individual, formal word-processed report
Submission format	Online via VITAL
Submission deadline	Midnight on Friday 7th May, 2021
Late submission	Standard university penalty applies
Resit opportunity	August resit period
Marking policy	Marked and moderated independently
Anonymous marking	Yes
Feedback	via VITAL GradeMark® / Turnitin Feedback Studio
Subject of relevance	Neural Networks and modelling
Learning outcomes	Mathematical aspects of backpropagation Practical training of a neural network

Marking Criteria

Section	Marks available	Indicative characteristics	
		Adequate / pass (40%)	Very good / Excellent
Presentation and structure	20%	• Contains cover page information, table of contents, sections with appropriate headings. • Comprehensible language; punctuation, grammar and spelling accurate. • Equations legible, numbered and presented correctly. • Appropriately formatted reference list.	• Appropriate use of technical, mathematic and academic terminology and conventions. • Word processed with consistent formatting. • Pages numbered, figures and tables captioned. • All sections clearly signposted. • Correct cross-referencing (of figures, tables, equations) and citations. • Appendix containing source code and any additional evidence such as further results, graphs, photographs etc
Introduction, Method and Design	20%	• Steps presented clearly, numbered, and in passive tense. • Design of each task follows a logical sequence. • Code corresponds to design for each task.	• Calculations shown in full, justifying and explaining any decisions. • Code easy to follow, with clear comments and signposting. • Well-written code demonstrating programming proficiency.
Results	30%	• Results presented for each task, including screenshots of figures and numerical results. • Results for each task accompanied by a commentary.	• Correct results for each task. • Thorough and insightful commentary on all results explaining differences and demonstrating understanding of the underlying theory.
Discussion	30%	• Correct responses to discussion questions reflecting an understanding of the underlying concepts. • Conclusions correctly inferred from the results and linked back to experiment objectives. • Evidence of wider reading.	• Demonstrating an understanding of multi-layer perceptron. • Meaningful reflection on the outputs of the experiment.

Important: Marking of all coursework is anonymous. Do not include your name, student ID number, group number, email or any other personal information in your report submitted via VITAL. A penalty will be applied to submissions that do not meet this requirement.

Introduction

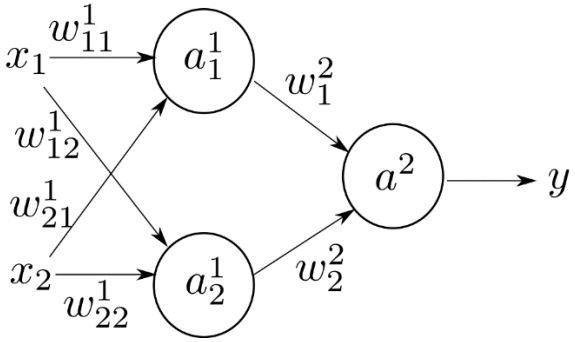
This assignment has two tasks with the objective of developing, exploring and testing the learning processes of artificial neural networks. The first task focuses on the underlying mathematical foundations. The second task focuses on the design of a neural network to solve a real-world problem, using Matlab neural network toolbox.

This assignment is assessed by means of a formal report that must be uploaded through Vital. The Matlab code to solve Task 2 must be submitted as an appendix to the report. While it is possible to solve Task 1 by writing Matlab code, the report should include detailed mathematical derivations with explanations, not Matlab code.

The report should contain detailed solutions and discussion of the results. The quality of the presentation and discussion is assessed.

Task 1 [45 Marks]

We consider a neural network with two layers and the following diagram



The equations that represent this neural network are

$$a_1^1 = \phi_1(w_{11}^1 x_1 + w_{21}^1 x_2)$$
$$a_2^1 = \phi_1(w_{12}^1 x_1 + w_{22}^1 x_2)$$
$$a^2 = \phi_2(w_1^2 a_1^1 + w_2^2 a_2^1)$$
$$y = a^2$$

The activation functions $\phi_1(\cdot)$ and $\phi_2(\cdot)$ are sigmoid and rectified linear units, with the equations:

$$\phi_1(x) = \frac{1}{1+e^{-x}}.$$
$$\phi_2(x) = \begin{cases} x, & x \geq 0 \\ 0, & otherwise \end{cases}$$

Note that this neural network does not include biases. The initial weights are $w_{11}^1 = 1,\ w_{12}^1 = 0.1$, $w_{21}^1 = 0.3, w_{22}^1 = 0.4, w_1^2 = 0.1, w_2^2 = 0.2.$ The following tasks must be completed.

1) For each neuron, calculate its induced local field and output for the following input value $x_1 = 1, x_2 = 2.$ [10 Marks]

2) If the desired output to the previous input is $d = 0.5$, use the backpropagation algorithm to compute the gradients of all weights. **[15 Marks]**
3) Calculate the updated weights for a learning rate 0.1. **[12 Marks]**
4) Calculate the output to the input value $x_1 = 1$, $x_2 = 2$ with the updated weights. Discuss the result. **[7 Marks]**

Task 2 [55 Marks]

In this task, we make use of the Matlab deep learning toolbox to design and evaluate a neural network in a real world data set and a synthetic data set. A preliminary step is to install Matlab, with the deep learning toolbox[1]. The student should first understand the basics through the Matlab tutorial for crab classification

https://uk.mathworks.com/help/deeplearning/ug/crab-classification.html

We will use the cancer dataset, in which we aim to classify tumours as either benign or malignant depending on the characteristics of sample biopsies. Run "open('cancer_dataset')" in Matlab to understand the dataset.

The Matlab script must start as

```
clear
clf
close all
setdemorandstream(1)
%Cancer dataset
[x,t] = cancer_dataset;
```

Above, we use setdemorandstream(1) to get reproducible results, this line must not be changed. Variable x represents the feature vectors and variable t the desired output for the feature vectors.

The first task is to understand and visualise the dataset. As the feature vectors have 9 dimensions, they are not straightforward to visualise, so we focus on the first two dimensions and the corresponding class. Draw a figure in which samples (data points) of the benign class are shown as a circle and samples of the malign class as a cross. In the figure, the x axis represents the first dimension and the y axis the second dimension of each feature. **[5 Marks]**

Hint: The command "plot" with arguments 'or' and 'xb' can be used to draw points represented as circles (with red colour) and crosses (with blue colour).

1) Based on the above figure, is it easy to perform the classification task based on the first two features? Justify your answer. **[5 Marks]**
2) How many data points are there in the dataset? How many data points corresponds to the benign class and the malign class? Explain the procedure to answer these questions. **[5 Marks]**

[1] https://www.liverpool.ac.uk/csd/software/software-downloads/

A fully connected neural network with one hidden layer with *N* neurons can be created with the command
net = patternnet(N);

This neural network is not complete as one needs to set the input and output neurons that match the data. The structure of the neural network is finalised by using the training command

[net,tr] = train(net,x,t);

This command trains the network with some standard parameters that are stored in the variable net.trainParam. In addition, the dataset is stored into three different subsets: training set, validation set and test set. The indices of x that correspond to each subset are stored in variables: tr.trainInd, tr.valInd and tr.testInd.

3) Use Matlab documentation and/or other resources (with citation) to explain the purpose of the training set, the validation set and the test set. **[5 Marks]**
4) Create and train a network with *N*=5. What is percentage of data points that belong to the training set, validation set and test set, respectively? Does this number change if we change *N*? **[5 Marks]**
5) Run "net.performFcn='mse';" to use the mean square error as the cost function to train the network. Set "net.trainParam.max_fail=10000;" so that the validation set is not used to stop the training. Run "setdemorandstream(1);" to set the random seed to be able to reproduce the results. Re-train the network and run the command "plotperform(tr)". This command shows a figure with the cost function against the number of epochs for the training, validation and test set. Include the figure in the report and discuss the results. **[5 Marks]**
6) Repeat step 6) but without calling the command "net.trainParam.max_fail=10000;". Now, we make use of the validation set to stop the training. Discuss the results. **[5 Marks]**
7) Plot the confusion matrices for the trained neural network in 7) clicking on "Confusion" in the dialog that arises after training. In the report, include the confusion matrices and explain them with your own words by looking at Matlab documentation and/or other resources (with citation). For the test set, what is the probability of correct classification in this example? **[5 Marks]**
8) We now proceed to evaluate a synthetic data set. For this task, you should create a new Matlab script (new m file). The m-file must start with "setdemorandstream(1)" to obtain reproducible results.
We consider points (x,y) that belong to the two dimensional plane and have two classes. Points in class one, represented as a vector [1,0] belong to the set

$$C_1 = \{(x,y): \sqrt{x^2+y^2} > \Gamma_1, \sqrt{x^2+y^2} < \Gamma_2\}$$

where $\Gamma_1 = 5$ and $\Gamma_2 = 20$. That is, the points that belong to class one meet $\sqrt{x^2+y^2} > \Gamma_1$ and $\sqrt{x^2+y^2} < \Gamma_2$. The label for class one is the column vector $[1,0]^T$. The points in class zero, with label $[0,1]^T$, correspond to the points in the plane that do not belong to class one. Sample 10000 points from a zero-mean Gaussian distribution with zero-mean and standard deviation 10 by writing

x=10*randn(2,N_data);

Obtain the class for each of these points and draw a figure all the points in the data set and their classes, such that the points in class one are shown with red colour, and the points in class zero with blue colour. [5 Marks]

9) Use patternnet to train a network with *N=2* and the default training parameters. Calculate the output of the neural network for the points in the dataset. The outputs are 2-D vectors, whose components sum to one. The most likely class for each data point is given by the component with the highest value. Draw a figure with all the points in the data set and their predicted classes by the neural network. Discuss the results. [5 Marks]

10) Re-run the code with *N=5*, *N*=10, *N*=20 showing the previous figure. Show the figures and discuss the results in the report. [5 Marks]