EEEM005 ASSIGNMENT INFORMATION

SUBMIT YOUR REPORT TO SURREYLEARN AS SINGLE PDF FILE

LABEL REPORT NNSIMULATION, PROLOG OR HANDWRITING

TYPE YOUR ANSWERS IN A WORD DOCUMENT AND SAVE AS PDF. CHECK CAREFULLY THE PDF BEFORE YOU SUBMIT.

NO HAND-WRITTEN TEXT IS ALLOWED.

ANY FIGURES OR DIAGRAMS SHOULD BE PRODUCED ELECTRONICALLY EITHER BY WORD, MATLAB OR OTHER SUITABLE PROGRAM AND INSERTED INTO THE WORD DOCUMENT. IF YOU PREFER HAND-DRAWN FIGURES OR DIAGRAMS IT IS ALLOWED BUT THEY MUST BE CLEAR, CAREFULLY DRAWN AND INSERTED INTO THE WORD DOCUMENT.

ALL SCRIPTS WILL BE SUBMITTED TO TURNITIN.

SEE SURREYLEARN UNDER ASSIGNMENT FOLDER FOR DEADLINE.

LATE PENALTY APPLIES.

Please note that you can choose one of the following three assignments. The first is a matlab simulation using the Neural Network Toolbox. The second is a programming assignment using Prolog. The third is a report on Neural Networks based on your research and requires no simulation or programming. The marking scheme follows the assignment description.

For all the reports, please note the following guidelines:

Your report should contain an Executive Summary (approx. 500-700 words) Introduction and Conclusion, as well as Table of Contents(including section numbers)

> ***Executive summary: summarizes the report in such a way that readers can rapidly become acquainted with the whole report, including major findings and conclusion.***
>
> ***Introduction: identifies to the reader the background, aim and purpose of the report, with a short guide to the sections in the report.***
>
> ***Conclusions: summarises findings and any overall conclusions, also possibly discusses future work that you might later perform.***

SOFTWARE:

Both Matlab and SWI Prolog are available for use on your own laptop. If you do not have a laptop, you can use any of the FEPS Labs.

SWI Prolog can be found at http://www.swi-prolog.org/ and the FAQ is a good place to start. See also tutorials/beginner/getting_started tutorials/beginner/learn_prolog and tutorials/beginner/debugger.

Assignment 1) NN matlab simulation for ‘Understanding how to solve pattern recognition problems using backpropagation’

You may find the following three exercises useful as an introduction to the neural network toolbox in matlab. Initial Exercises E1 E2 E3 use matlab help documentation. Note that E3 is intended to demonstrate the difference between classification and regression, but is not needed for the assignment.

E1) Read ' Multilayer Shallow Neural Networks and Backpropagation Training'. In particular you should understand *feedforwardnet*.m and *train*.m. Note that specialized versions of the *feedforwardnet*.m include pattern recognition (*patternnet*.m) and fitting (*fitnet*.m), to be used in E2) and E3) respectively.

E2) Perform exercise in 'Classify Patterns with a shallow Neural Network' - cancer_dataset with *trainscg* and default parameters. Save the ‘advanced script’ at the end of the exercise and use this code as a basis for carrying out the assignment. *In particular, you need classification error rather than cross-entropy for Exp 1).*

E3) Perform exercise in ‘Fit Data with a shallow Neural Network’ - engine_dataset or building_dataset with *trainlm* and default parameters.

If you are having difficulty with the initial exercises E1,E2 particularly understanding the advanced script generated from E2 please contact me. For Exp 1) 2) 3) 4) I am happy to clarify what is required but cannot help you to carry out the implementation of assignment.

ASSIGNMENT - EXPERIMENTS

For help see eeem005_NNassignment info.pptx

Exp 1) Using cancer dataset (as in E2) and ‘*trainscg’ or an optimiser of your choice*, vary nodes and epochs (that is using early stopping for epochs) over suitable range, to find optimal choice in terms of classification test error rate of node/epochs for 50/50% random train/test split (no validation set). It is suggested that you initially try epochs = [1 2 4 8 16 32 64], nodes = [2 8 32], so there would be 21 node/epoch combinations. *(Hint 1: from the ‘advanced script’ in E2, nodes can be changed to xx, with `hiddenLayerSize = xx`; and epochs changed to xx by adding `net.trainParam.epochs = xx;` placed after `net = patternnet(hiddenLayerSize, trainFcn);` -- see ‘trainscg’ help documentation for changing epochs).* Repeat each of the 21 node/epoch combinations at least thirty times, with different 50/50 split and take average and report classification error rate and standard deviation(std). Graph classification train and test error rate and std as node-epoch changes, that is plot error rate vs epochs for different number of nodes. Report the optimal value for test error rate and associated node/epoch values. *(Hint 2: as epochs increases you can expect the test error rate to reach a minimum and then start increasing, you may need to set the stopping criteria to achieve the desired number of epochs – Hint 3: to find classification error rates for train and test set, you need to check the code from E2.* ***Note: classification error rate = number of mis-classifications divided by total number of patterns, fraction or percentage e.g. predict the class labels from trained network and compute number of differences with target labels. Note that train and test patterns/targets may be obtained using tr.trainInd and***

tr.testInd (see documentation for train.m) and simulate with y = net(x) and convert to labels as shown in the advanced script from E2. As an alternative, it is also possible to obtain classification error rate from the confusion matrix*).*

Exp 2) For cancer dataset, choose an appropriate value of node and epochs, based on Exp 1) and use ensemble of individual (base) classifiers with random starting weights and Majority Vote to see if performance improves - repeat the majority vote ensemble at least thirty times with different 50/50 split and average and graph (Each classifier in the ensemble sees the same training patterns). Repeat for a different odd number (prevents tied vote) of individual classifiers between 3 and 25, and comment on the result of individual classifier accuracy vs ensemble accuracy as number of base classifiers varies. Consider changing the number of nodes/epochs (both less complex and more complex) to see if you obtain better performance, and comment on the result with respect to why the optimal node/epoch combination may be different for an ensemble compared with the base classifier, as in Exp 1). *(Hint 4: to implement majority vote you need to determine the predicted class labels – probably easier to implement yourself rather than use the ensemble matlab functions. For the graph,* use *average value over thirty runs for y-axis. The x-axis should be Number of classifiers. Also for complexity, another graph with x-axis as Number of epochs)*

Exp 3) Repeat Exp 2) for cancer dataset with two different optimisers of your choice e.g. '*trainlm'* and '*trainrp'*. Comment and discuss the result and decide which is more appropriate training algorithm for the problem. In your discussion, include in your description a detailed account of how the training algorithms (optimisations) work.

Exp 4) ***Please note that you should only attempt this final section if you have time and after you have completed all other sections – only counts ten percent***. The requirement is to distinguish between two equi-probable classes of 'overlapping' two-dimensional Gaussians', where class 1 has mean $\mu_1 = [0,0]$ variance $\sigma_1^2 = 1$ and class 2 has $\mu_2 = [2,0]$ variance $\sigma_2^2 = 4$. It can be proved that the Bayes (optimum) boundary for this problem is a circle whose centre is located at $[-{}^{2}/_{3}\,, 0]$ and radius 2.34. Find the optimal choice of node/epochs in terms of test error rate, using 300/3000 pattern train/test split, possibly using an ensemble, and plot the decision boundary on the same plot as the Bayes boundary, and discuss your result. *(Hint 5:You may find the matlab function mvnrnd.m useful for generating random numbers*)

It is expected that the final report will be 9-12 pages, but can be longer. Put your matlab code in an appendix.

Assignment 1) Marking Scheme matlab simulation

Executive Summary [5]
Introduction [5]
Main body/Content
Exp 1) [20]
Exp 2) [20]
Exp 3) [30]
Exp 4) [10]
Use of clear/concise English, Report Presentation/ layout [5]
Conclusions [5]

Assignment 2) Prolog implementation of 'London Underground'

You may find the following exercise useful before starting the assignment. Implement tutorial sheet PROLOG 1, and note that Prolog for these problems can mostly be found in the notes.

ASSIGNMENT – PROLOG IMPLEMENTATION

The topic is Prolog implementation of 'London Underground' planner - choose a subset (10-20 stations) of the Underground, large enough for interesting queries (e.g. getting station A to B, with minimum time, fewest changes - the times can be estimated, or use the TFL journey planner for accurate times – for example https://tfl.gov.uk/maps/track/tube). In order to attain the highest mark, you should also attempt to implement AI search schemes for the London Underground, other than the default Prolog search. (see Bratko 4th edition chapters 11,12,13). You will also get credit if you introduce any novel additions to the journey planner.

I am happy to clarify what is required but cannot help you to carry out the implementation of the assignment. If you don't get a full working version, document what you have with any problems and you will get credit for that.

It is expected that the final report will be 9-12 pages, but can be longer.

Assignment 2) Marking Scheme Prolog

Executive Summary [5]

Introduction/Background [5]

Main body/Content

Prolog code, description of representation,

answers to Prolog queries, Prolog documentation [45]

AI Search schemes [25]

Conclusions [5]

Use of clear/concise English [10]

Report Presentation/ layout [5]

Assignment 3) 'Neural Networks for Handwriting Recognition' – report

As an alternative assignment, which requires no programming or simulation, you can write a report on 'Neural Networks for handwriting recognition'. It should include details of the NNs used, and the features that are input to the network. Also, please include both historically and current, the use of different neural networks for the application and the respective advantages/disadvantages. You should include historical survey, state of the art, case studies (two or three), and what can be expected in the future. The history should also cover technical/theoretical developments from its early beginnings up to the present day. For example, what is the difference between off-line and on-line recognition, and what recognition rates are currently attainable. The examples or case studies could be from industry, and should include both neural network technique and practical application, describing how the problem has been solved. **Guidelength is 4500 words not including executive summary (500-700 words approximately).** For the appropriate level of writing, imagine you are consultant for a company, presenting a report to an audience that is technical but not necessarily expert on all aspects of neural networks. If you want to include a lot of detail, use appendices – which is not included in the word count. **At the beginning of your report please include 1) the word count of the total document excluding Exec Summary, References and Appendices and 2) word count of Exec Summary.**

Assignment 3) Marking Scheme NN Report

Executive Summary [10]

Introduction/Background [10]

Main body/Content

historical survey, state of the art, future prospects [30]

case studies [20]

References [5]

Conclusions [5]

Use of clear/concise English [10]

Report Presentation/ layout [10]