Introduction to Neural Networks : Continuous Assessment Project

John A. Bullinaria - 2004

The continuous assessment component for the Introduction to Neural Networks module (which forms 30% of the total assessment for that module) will be based on your written report on the following mini-project.

The objective of this exercise is for you to gain practical experience in setting up, training and optimising a neural network designed to recover the underlying function from a set of noisy training data. All you have to do is work through the following simple steps:

(1) Download from the module web-site (http://www.cs.bham.ac.uk/~jxb/inn.html) the program 'datagen' that will generate your training, validation and testing data sets. Run that program on one of the School's Linux workstations, and enter your student ID number when requested. You will find that your data sets have appeared in the same directory. Do not just copy someone else's data sets – the program pseudo-randomly generates different data sets (from the same distributions) for each student.

(2) Set up a feed-forward neural network with one hidden layer of 20 sigmoidal units and an output layer of linear units (e.g. using a neural network simulator such as *javaNNS*).

(3) Train your neural network on your training data using back-propagation, and check how its performance on the validation data set changes. By systematically trying a range of values and plotting graphs, find values of the back-propagation learning rate η and momentum α that result in consistently fast and successful training.

(4) The test data set provided has no noise in it, so you can check how well your networks have recovered the underlying function. Using a small selection of learning parameters from (3), see how the performance on the test set varies during training.

(5) Investigate how varying the number of hidden units affects the results in (3) and (4). You may need to determine a new best learning rate and momentum for each case.

(6) Write a report on what you did and what you found. Include a discussion of its relevance to the optimization of generalization performance. A reasonable length for the report would be between 2000 and 3000 words, plus as many diagrams, tables and graphs as you think appropriate.

(7) If the above seems too easy, get some extra marks by running and describing further simulations. You could try to improve generalization by using weight decay, or explore the use of larger data sets or different initial weight distributions, or ...

There will be optional lab sessions to help you get started with *javaNNS* and to offer any other help and advice that may be needed with this assignment. These will take place on:

6pm Monday 8th / Tuesday 9th November – Ground floor lab – attend *only one* of these

6pm Monday 22nd / Tuesday 23rd November – Ground floor lab – attend *only one* of these

The report must be handed in to the School Office by 12noon on Wednesday 12 January 2005