COMPSYS726 Assignment 2 - Guidance on Experimental Method

A number of you have been asking me what it is I am looking for in your reports. A quick answer is, to demonstrate a good method to try and find the best performing network with the given blood transfusion dataset.

A good starting point is to use is to build a single hidden layer network and to use just one of the normalised blood transfusion datasets. A general rule of thumb is to use 2N+1 neurons in the hidden layer, where N is 4, the number of input dimensions. This is usually sufficient, but you should try several other possibilities. You will find that just adding, or subtracting one neuron has little effect. In the past, when I have been conducting this type of search I have tried doubling, or halving the number of neurons for each step in the investigation.

Initially, you should keep the learning rate fixed at, say, 0.1 and vary the hidden layer size. Try two hidden layers to see if that performs better. The key value that you should base your decisions about performance on is the number of correct test values as a percentage of all test values.

When you look at the graph, you should see an initial drop in the error as depicted by both the training curve and the testing curve. You should be able to estimate the point where there are enough epochs to be sure that there is unlikely to be any more error reduction. You should then stick to that number of epochs for most of your work, but keep an eye on this point in case the network behaviour changes once you begin to home in on a good network.

Each time you begin a simulation run the initial weights are randomized. This means that you will almost always get a slightly different result. When you are getting close to what you think might be a good network, run the simulation three times to get a feel for consistency of behaviour.

The next parameter you should then experiment with is the learning rate. Try a range of values between 0 and 1. The main purpose of the learning rate is to reduce the number of required epochs during the training session. Once the value gets too high the output plot is very spiking (unstable) and you are unlikely to get a good result. However, sometimes there is a sweet spot where a particular learning rate step size works well.

Given that you now have a network and learning rate that you are happy with, try some of the following experiments:

- For example, if the network is trained using too many hidden layer neurons, you should be able to observe a very low training error, but a very large test error, showing that the network is memorising the data rather than generalising.
- The best generalisation results are usually found when the number of hidden layer neurons is minimised while still converging to a lower test error. How low can you go?
- If you run the network for a very large number of epochs, you may be able to observe the effect of newer learning overwriting earlier learning, that is, the training and test errors may increase again.

- There was more than one normalisation approach. Does it make any difference using the other normalisation methods, e.g. applying those datasets to your best network?
- Does changing the hidden layer squashing function make any difference?
- Does the bias term make any difference at all

Some of you have tried to install PyBrain at home on your own Ubuntu system. If you are having trouble, try the following link (thank you to the student who found this):

http://ytbau.blogspot.co.nz/2010/11/pybrain-v03-installation-on-ubuntu.html

As mentioned earlier, the assignment length should be 6-10 pages. I am not looking for 10 pages of dense writing. Use tables and graph plots to demonstrate any points that you are making. The assignment dropbox is now open and called assignment 2 PyBrain. The course is COMPSYS726 and the due date is midnight on 25th May 2012.