Neural Networks, Winter, 2008
G.G.Lendaris

PRELIMINARY INFORMATION re. PATTERN RECOGNITION COURSE PROJECT and start-up assignment.

PART A of Course Project: February 18, 2008 DUE: Wed., February 27, 2008

This exercise is intended as start-up preparation for a larger-scale set of experiments to be carried out over the remaining weeks of the term after this preliminary set is due in.

The data is related to automatic land-use classification via photographic imagery of Phoenix, Arizona, taken by NASA from the satellite called SKYLAB II in 1974. There are two pairs of .nna files: each pair contains a data set for Training and a data set for Testing (Generalization), and each pair provides a different number of inputs for the neural net: 194 and 54. In the 194 input case, the data represent "raw measurements" from a preprocessor; in the 54 input case, the data represent "features" that were extracted from the 194 measurements. There are 292 total events in each set, approximately 150 are in the Training set, and the remainder are in the Test (Generalize) set.

A number of experiments have previously been carried out with this data with neural nets of the "backpropagation" variety. Some results will be described in class.

For your "course project" each of you is to carry out some experiments with this data on another of the neural network paradigms [see bottom of this sheet].

For this week, however, your task is to *familiarize yourself with the data set* and various of the NeuralWorks processes you will need to carry out your experiments. Data available on Web site.

Create a multi-layer, feedforward network for the **54 input data set**; the size of the input layer will be 54 elements [NOTE: ALLWAYS STUDY YOUR .nna FILES. IN THIS CASE, FIELDS 1, 56, 62 AND 63 CONTAIN IDENDIFICATION DATA; TELL NWORKS TO START LOOKING IN FIELD 2 FOR THE (54) INPUTS AND FIELD NUMBER 57 FOR THE (5) OUTPUTS]; a hidden layer of 8 elements, and an output layer of 5 elements. Such nets have been successfully trained (using the Backpropagation algorithm) to over 90% correct responses on both the train and generalize data sets, with the nets being trained from 6,000 to 30,000 pattern presentations (N = 6,000; N = 30,000). See READ.ME file that is with the data.

Take "checkpoints" of the runs at appropriate intervals (use NWorks *checkpoints* in the Run menu -- see "NeuralWare Check Point Bug Fix" appended here), and after the run is completed, go through and do a recall process on each of the snapshots. Use whatever tools are at your disposal (e.g., within NWorks or outside of NWorks: EXCEL, MATLAB, etc.) to determine the performance of the NN at various stages during the learning checkpoints process-- make sure you understand the naming convention). Create plots of the *learning dynamics* for each experiment. Show comparative results for different experimental conditions. NOTE: when you do the recalls, if you instruct NeuralWorks to APPEND the results in the .nnr file, all the results are sequentially put into the same results file. The alternative is an OVERWRITE option, and if you weren't aware of this, there would be some surprises.

Additional Assignment: familiarize yourself [via lecture material, Texts 1 & 2, and any other source you wish to consult] with the following paradigms sufficiently to allow yourself to bring a list of 3 choices with you to class on **Wednesday, Feb. 27th**, and I will make assignments accordingly: LVQ, SOM, RBF, PNN, GRNN, Counterprop, Fuzzy Artmap -- part of your PROJECT will be to learn enough about the selected paradigm to carry out meaningful experiments.

ADDENDUM: NeuralWare Check-Point Bug Fix.

Note from NeuralWare's support person:

Hi Dr Lendaris,

I hope all is going well. Investigating the CheckPoint-Save-Multiple-Networks problem has uncovered a work-around that is easier than the SaveAs/Close/Open used in the past.

If your students use a double-quoted name like this: "mynet00" then all works well. That is to say that just prior to training they should do a SaveAs command, and TYPE over any existing entry like untitled.nnd to be "mynet00". Dropping the .nnd portion and using double quotes is the trick. So rather than save to mynet00.nnd, save to "mynet00", quotes included. This is needed only when using {0, 00, or 000} at the end of a filename, along with Run/CheckPoints, to automatically save numerous models. In typical ProII/PLUS work such final zeros are not used, and double-quoting things is unnecessary.

On the subject of the older 'File/Change Directory' menu item, that was removed to substantially improve usage. The feature we refer to 'round here as "path retention". ProII/PLUS was enhanced to use standard Windows dialogs and browse buttons to locate things like *.nnd, *.nna or *.txt files. The best part is that once you navigate to a directory (using File/Open, File/SaveAs, or the InstaNet Browse buttons) then that directory is remembered from one session to the next. In the past you had to keep going back to a hard disk drive or directory using File/CurrentDirectory. Now you just go there once using the drop-down locator in File/Open, and ProII/PLUS saves that to into our *.ini file. The next time you run ProII/PLUS and click File/Open, the earlier drive and path is the default location. We expect that path retention makes the product easier to use than the earlier method.

There is an alternative to using Run/CheckPoints to auto-save models, to review the network's progress over time. The product can save progress by making a graph (a chart), and numerically write (append) all graph activity to a log file. Here's how...

You can create graphical instruments to monitor almost any type of activity in the Professional II/PLUS, looking at individual PEs, Layers, Connections, or groups of each. Our graphs can monitor and log the desired information to disk.

To log information, edit the instrument by double-clicking on it. In the lower-left of the Instrument Edit dialog select the Logging Active box and okay the dialog. Now when you perform a Learn, Recall, SaveBest or Test operation, the graphic data displayed in the instrument will also be written to a text file named "instrum.nnp" by default. Subsequent Learn, Recall, SaveBest or Test operations will, by default, append to this file. See the Reference Guide page RF-172 (PDF page 186 of 297) for more details.

Hope all is going well in your course, and Happy Thanksgiving!

PS: I heard from student Jeff Weintraub that your class is a tremendous help to him in understand n.nets. He said he's been around them for 15 years, and your class is great.

Best regards,
Bob Everly
Product Support, help@neuralware.com
10/2006