Chapter 10: The Rosenblatt Program

10.1 INTRODUCTION

Rosenblatt is a program written in Visual Basic 6.0 for the demonstration and exploration of perceptrons. It is designed for use on a computer based upon a Microsoft Windows operating system. A second program, RosenblattLite, is identical to Rosenblatt with the exception that it does not include the capability to save network results in Microsoft Excel workbooks. A third program, RosenblattJava, is identical to RosenblattLite, but is written in Java so that it can be used on computers that do not use the Windows operating system. In this chapter, Rosenblatt will be the only program referred to, as the user interface for it is identical to the interface for the other two programs. All programs are distributed as freeware from the following website:

http://www.bcp.psych.ualberta.ca/~mike/Book3/

The current program explores pattern classification with three different versions of the perceptron. In the first, the Rosenblatt training rule – which is equivalent to the Delta rule that was introduced with the James program in Chapter 3 – is used to train a perceptron with a threshold activation function. In the second, gradient descent is used to train a perceptron that uses a logistic activation function as a continuous approximation of the threshold activation function. In the third, a variation of the gradient descent rule is used to train a perceptron that uses a Gaussian activation function in its output units. This last function means that the output units in essence have two different thresholds, instead of one. These variations of the perceptron are described in more detail in Chapter 10 of (Dawson, 2003).

10.2 INSTALLING THE PROGRAM

Rosenblatt is distributed from the above website as a .zip file. The following steps will result in the program being installed on your computer:

1. Download the file Rosenblatt.zip to your computer by going to the website, click on the program icon, and save the file in any desired location on your computer.
2. Go to the saved Rosenblatt.zip file on your computer, and unzip it with a program like WinZip. The result will be three different objects: setup.exe, setup.lst and Rosenblatt.cab.
3. Run the setup.exe program. This will call an Install program that will complete the installation of the program on your computer, which will include the installation of an Examples folder with a few sample training files.

10.3 TRAINING A PERCEPTRON

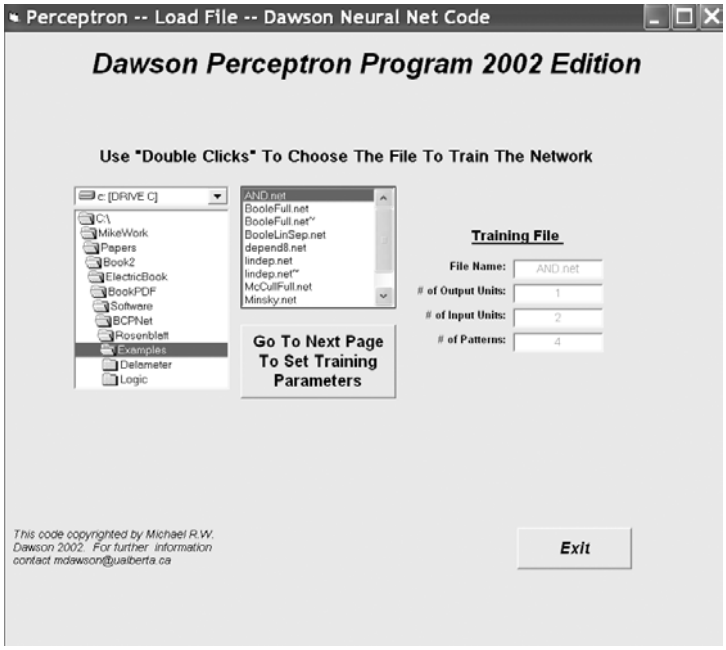
10.3.1 Starting The Program

The program can be started in two different ways. First, one can go into the directory in which the program was installed and double-click on the file “Rosenblatt.exe”. Second, one can go to the start button on the computer, choose programs, scroll to the program group BCPNet, and select the program Rosenblatt.exe.

10.3.2 Loading A File To Train A Network

After the program is started, the first form that appears is used to select a file for training the distributed memory. This form is illustrated below. By using the left mouse button and the drive selection tool located in the upper left of the form, one can choose a computer drive on which directories and files are located. The available directories on the selected drive are listed in the directory selection tool that is immediately below the drive selection tool. One opens a di-

rectory by double-clicking it with the left mouse button. If the directory contains any files that end with the extension .net, then these files will be displayed in the file selection box located in the upper middle of the form. The properties of .net files are described later in this book. These files have a particular format that the Rosenblatt program is designed to read, and only files that end in this extension can be used to train the network.



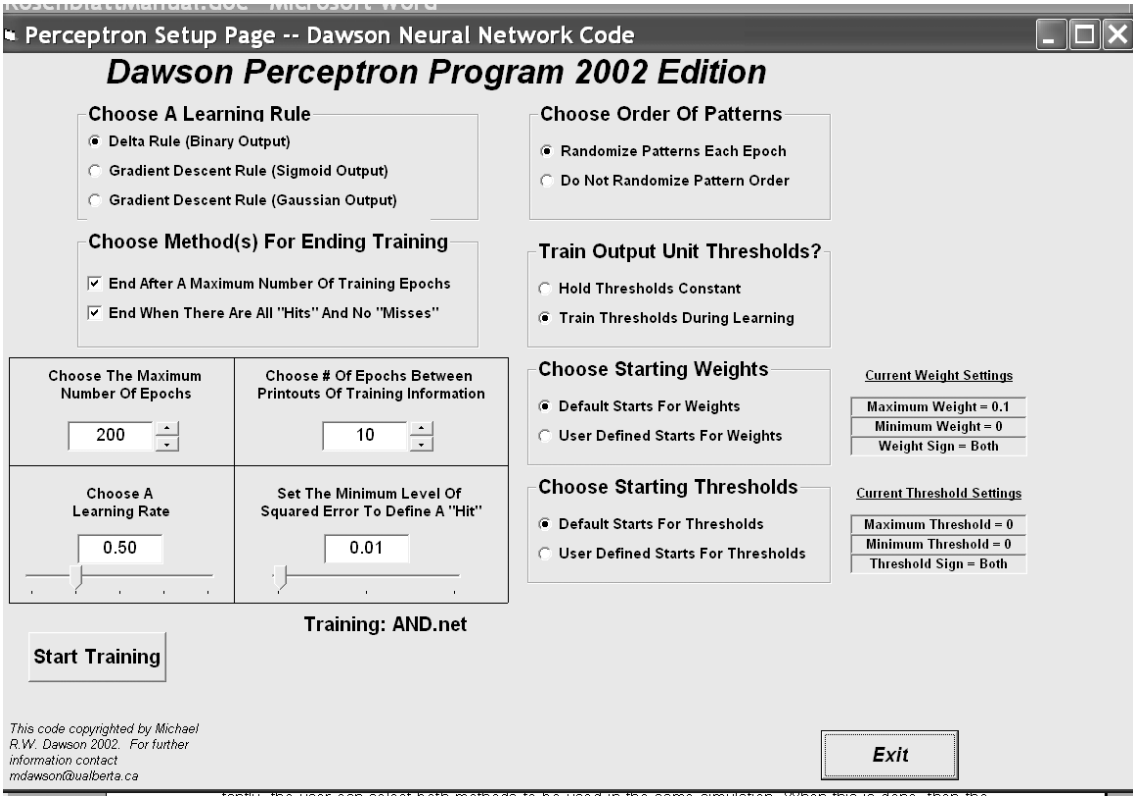
One chooses a .net file by double-clicking one of the file names that is displayed in the file selection box. When this is done, the program reads the desired file, some of the file's properties are displayed, and another button appears on the form. In the figure on the right, the file "AND.net" has been selected (and read). On the right of the form its general properties are displayed, and the button permitting the user to proceed to the next part of the program is displayed under the file selection box.

In this example, if "AND.net" has been selected, but is not really the file that is desired, one can simply go back to the file selection tools and choose another file. When its file name is double-clicked, the new file will be read in, and will replace the properties of the previous (undesired) file.

Once the desired file has been selected, all that is required is to press the "Go To Next Page To Set Training Parameters" button with a left-click of the mouse. If instead one desires to close the program, then one can left-click the "Exit" button displayed on the bottom right of the form.

10.3.3 Setting The Training Parameters And Training The Network

When the program reads in the .net file, this only determines how many processing units are connected in the network, and defines the input and desired output patterns that are used in training. It is up to the user to define what learning rule to use, and to specify the value of the parameters to control (and stop) learning. The second form displayed by the program, an example of which is shown below, allows the user to choose these parameters. This section describes how this is done. If the reader wishes to learn more about what exactly is accomplished by setting these values on this form, then he or she should look through Chapter 10 of (Dawson, 2003).



The second form consists of a number of different tools that can be used to quickly control the kind of learning that will be carried out by the distributed associative memory. The first tool is used to choose which of three learning rules is to be used to train the perceptron. This choice also determines what activation function is being used in the perceptron's output units. The default rule is the Delta rule. When this rule is selected, the activation function is a threshold function. An output unit will generate a response of 1 when its net input is greater than a threshold, and a response of 0 otherwise. The second rule is a gradient descent rule for training output units with a continuous activation function, which in this case is the logistic equation. The derivative of the logistic equation is used to speed learning up by scaling the error term. The third rule is a gradient descent rule for training output units that employ a Gaussian activation function (i.e., value units). This rule is based on (Dawson & Schopflocher, 1992b) modification of gradient descent training, which uses an elaborated error term. It too scales error with the derivative of the activation function to speed learning up. A left-click of the mouse on this tool is all that is required to select which of the three learning rules will be used.

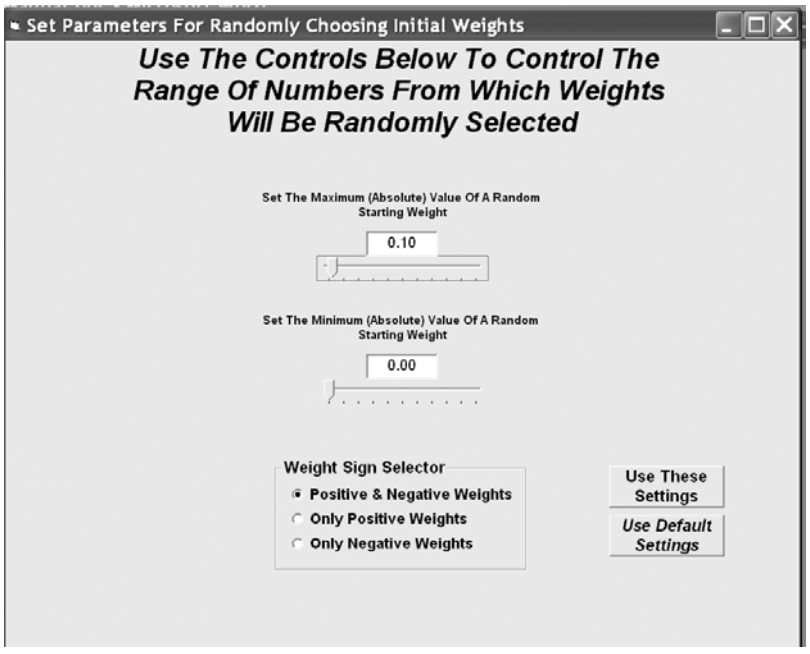
A second tool is used to choose a method for stopping training. In the first method, training stops after a maximum number of epochs (this value is set by the user). In the second method, training stops when there is a "hit" for every pattern and every output unit. This means that when each output is generating an acceptably accurate response for each pattern, training will stop. A left-click of the mouse is used to select either of these methods; when a method has been selected, a check mark appears in the tool. Importantly, the user can select both methods to be used in the same simulation. When this is done, then the simulation will stop as soon as one of the two conditions is met. This is the default situation, and it is recommended

A third tool determines the order in which patterns will be trained. The program is epoch-based, which means that each epoch or "sweep" of training involves presenting every pattern once to the perceptron. When a pattern is presented, output unit error is used to modify the weight values. One can have the program present patterns in a random order each epoch, which is the recommended practice. However, if pattern order is being manipulated, you can turn this

option off with a left-click of the mouse. When this is done, the patterns will always be presented in the order in which they are listed in the .net file that has been input.

A fourth tool determines whether the thresholds of the units (i.e., the threshold for the binary activation function, the bias for the logistic function, or the value of mu for the Gaussian function) should be trained. The default is to train the thresholds, because this permits the output units to "translate" their "cuts" through pattern space. However, in some situations it may be required to hold this value constant, which can be done with a left-click of the mouse button.

A fifth tool is used to determine the starting values of the connection weights, which are randomly selected from a distribution. In the default situation, the maximum value of a weight is 0.1, the minimum value is 0, and the sign option is "both", which means that negative and positive weights are possible. These defaults are displayed to the right of the weight-start tool. With these default values, weights will be randomly selected from a rectangular distribution that ranges from –0.1 to +0.1. However, in some cases it may be desirable to explore different starting states. This can be accomplished by left-clicking the "User defined starts for weights" option. When this option is selected, a new form appears, as is shown on the right. This form is used to set the minimum (absolute) value for a weight, the maximum (absolute) value for a weight, and the desired sign for the weight (positive, negative, or either). When the desired settings have been selected, the "Use These Settings" button will select them, and close the form. If it is decided that the default settings are desired, then this can be accomplished by using the "Use Default Settings" button. Whatever settings have been selected will be updated on the right of the settings form.



A sixth tool is used to determine the starting values of the randomly selected thresholds for the output units. The default is to assign every output unit a threshold of 0, regardless of which activation function has been selected. If different randomly selected starts are desired, then a left-click of the "User defined starts for thresholds" option will reveal a form similar to the form described above for manipulating the starting parameters for the weights.

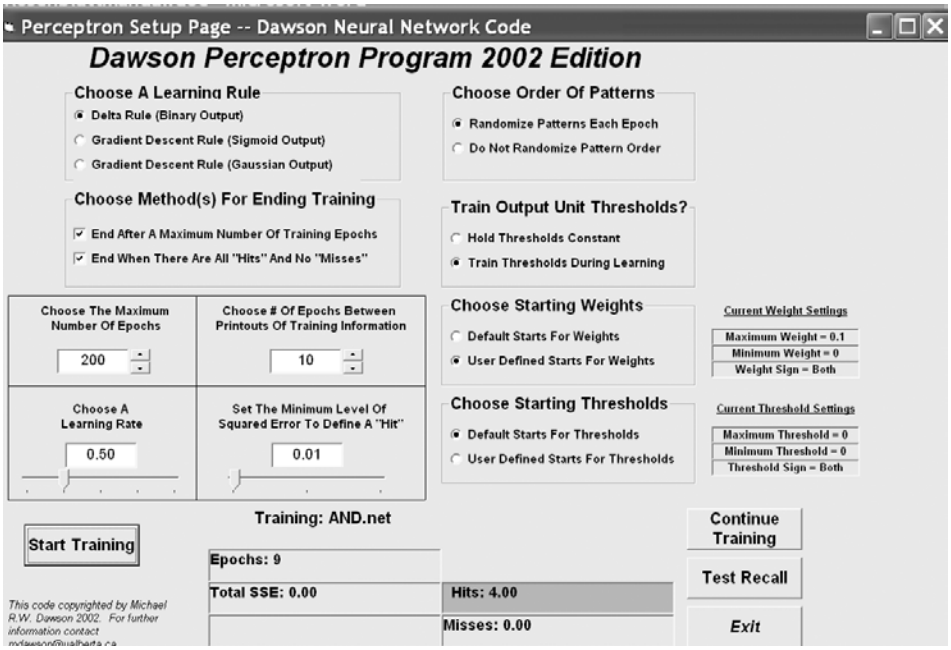
The four remaining tools on the form are used to set numerical values that control training. The first is a tool for specifying the maximum number of training epochs by left-clicking either arrow beside the value's box. This will either increase or decrease the value of this parameter, depending upon which arrow is selected. The maximum number of training epochs can also be

set directly by left-clicking the value's box with the mouse, and typing in the desired value. Note that it if the user chooses a value for this variable, then the "End After A Maximum Number Of Training Epochs" selection should also be selected. If this latter option does not have a check mark beside it, then the program will ignore this number when it is run! The default value (shown above) is 200.

The second is a tool for specifying the number of training epochs between printouts of training information. During training, the program will periodically print out information to tell the user how things are progressing. This includes information about what epoch has been reached, what the network SSE is, and the degree to which network SSE has changed since the last printout. The frequency of these printouts is controlled by the number displayed in this tool, which can be set in a fashion similar to that described for the previous tool. The default value (displayed in the figure) is 10. If this value is selected, then every 100 epochs the user will receive updates about network learning. The value selected for this parameter also defines the spacing of the x-axis of the "SSE by Epochs" plot that can be created from a form described later in this document.

The third is a tool for specifying the learning rate used by either learning rule. More details on the role of learning rate in the equations can be found in Chapter 10 of *Connectionism And Psychological Modeling*. The learning rate is used for all three learning rules. In setting the learning rule, two rules of thumb should be followed. First, if the learning rate is 0, then no learning will be accomplished. Second, it would not be typical to set learning rates greater than 1, although the user is free to explore the behavior of the network when this is done. The learning rate can be set in two different ways. One is to left-click on the arrow of the slider tool that is beside the value, hold the mouse button down, and use the mouse to slide the value of the learning rate up or down. The other is to select the box in which the learning rate is displayed, and to type in the desired learning rate. The default learning rate is 0.5. For some problems, when the Gaussian activation function is used, it may be desirable to speed learning up by *decreasing* this value to 0.1 or even to 0.01. For the other activation functions, the speed of learning can usually be increased by increasing the learning rate, provided that the learning rate is kept smaller than 1.0

The fourth is a tool for specifying the minimum level of error (that is, SSE) to define a "hit". The default value for this setting is 0.01. With this setting, this means that if the desired value of an output unit is 1.00, then if the unit generates activity of 0.9 or higher, a "hit" will have occurred. This is because 1.00 – 0.9 = 0.1, and the square of 0.1 is 0.01. Similarly, if the unit generates activity of 0.1 or smaller for a desired output of 0.00, then a "hit" will have occurred. If a more conservative definition of "hit" is desired, then this tool should be used to make the minimum SSE value smaller. If a more liberal definition is required, then this value should be made larger. The smaller the value, the longer it will take learning to occur. However, if this value is too large, learning will end quickly, but the perceptron's responses to stimuli will not be very accurate.

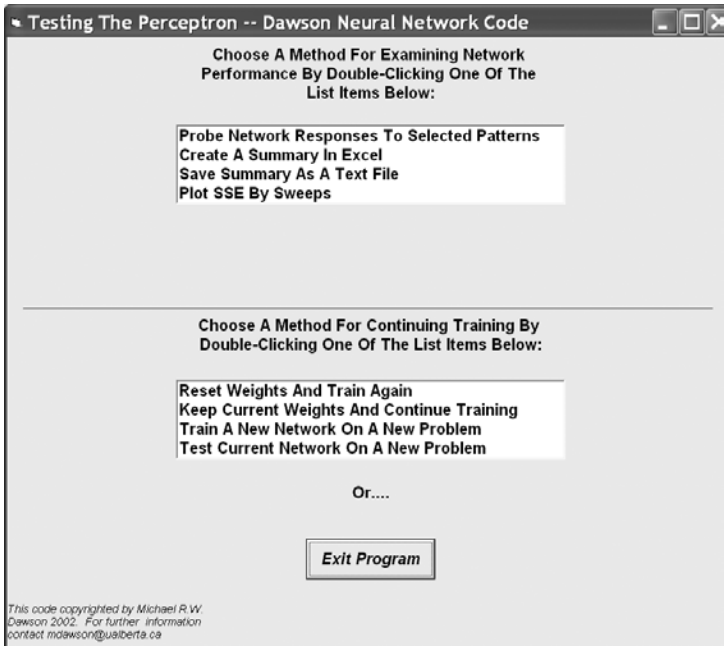


Once these tools have been used to select the desired training parameters, associations (memories) can be stored in the network by pressing the "Start Training" button with a left-click of the mouse. When this is done, new boxes appear on the form to show the user how training is proceeding (see the figure above). When training stops, two new buttons appear on the form. By pressing the "Continue Training" button, more training occurs using the settings that have already been selected on this form. By pressing the "Test Recall" button, the user moves to a new form that can be used to explore the performance of the trained network. The details of this form are described below. Of course, pressing the "Exit" button terminates the program. Note that as training proceeds, information about the number of sweeps, the total network SSE, and the number of hits and misses is displayed. In the preceding figure, training stopped after 9 epochs because SSE had dropped to zero, and there were 4 hits and 0 misses on the training patterns.

10.4 TESTING WHAT THE MEMORY HAS LEARNED

Once training has been completed, the perceptron has learned to classify a set of input patterns. With the press of the "Test Recall" button of the form that has just been described, the program presents a number of options for examining the ability of the network to retrieve the information that it has stored. Some of these options involve the online examination of network responses, as well as the plotting of learning dynamics. Other options permit the user to save properties of the network in files that can be examined later. One of these file options enables the user to easily manipulate network data, or to easily move the data into another program (such as a statistical analysis tool) for more detailed analysis (e.g., factor analytic analysis of final connection weights).

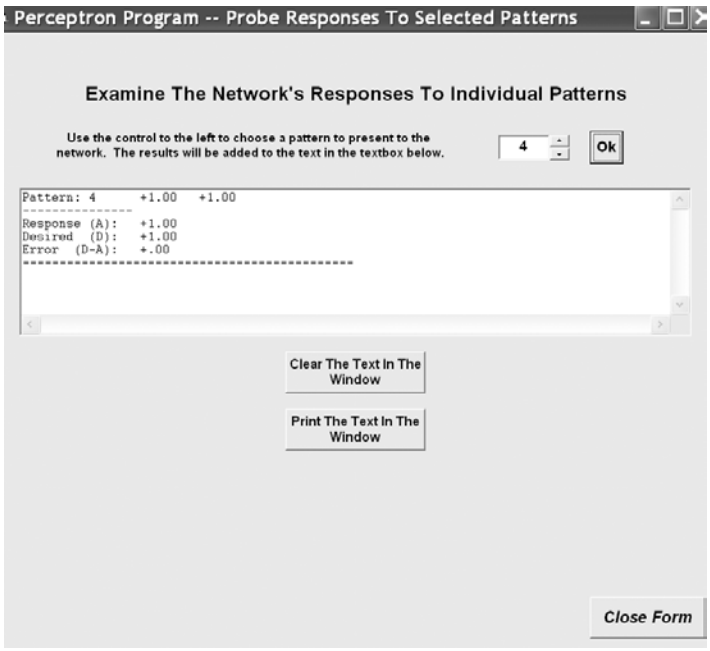
The user selects the specific activity to perform from either list that is illustrated in the figure below. Double-clicking the list item with the left mouse button results in the activity being carried out. The sections that follow first describe the different activities that are possible by selecting any one of the four actions laid out in the control box on the upper part of the form. Later sections describe the result of double-clicking any one of the three actions made available in the control box on the lower part of the form. Again, an "Exit Program" is also provided to allow the user to exit the program from this form.



10.4.1 TESTING RESPONSES TO INDIVIDUAL PATTERNS

After the network has learned some classifications, it may be of interest to the user to examine the particular responses of the network to individual cue patterns in the training set. For instance, in cases where the network is not performing perfectly, it could be that it is responding correctly to some cues, but not to others, rather than responding inaccurately to all of the stimuli. By double-clicking on the list item “Probe Network Responses To Selected Patterns”, the user causes the program to provide a form that allows the network to be tested one cue pattern at a time.

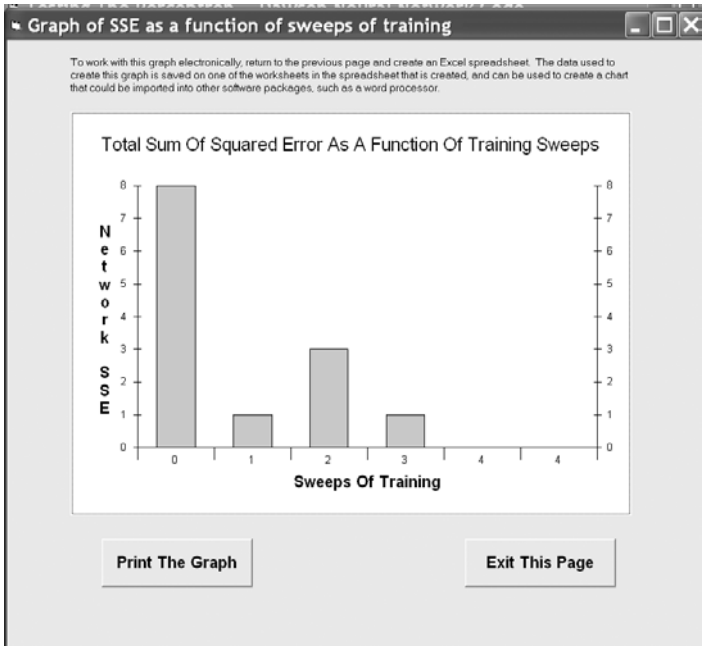
The form that permits this is depicted below. It provides a window in which network behavior is printed. When the form is initially presented, this window is blank. Left-button mouse clicks on the arrow controls at the top of the form are used to select the number of the pattern to be presented to the network. When the desired pattern number has been selected, the “Ok” button is pressed. The cue pattern is then presented to the network, and the network’s response is displayed. The display provides details about the cue pattern, the actual network response, the desired network response, and the error of the network. For instance, in the illustration, Pattern 4 has just been presented to the network.



More than one pattern can be tested in this way. The new pattern information is always displayed on top of previous pattern information. One can use the two scroll bars on the window to examine all of the information that has been requested. At any point in time, one can send this information to the system's default printer by pressing the button for printing. Also, one can erase the window by pressing the button for clearing the display. When the "Close Form" button is pressed, this form closes, and the user is back to the "Test Recall" list options.

10.4.2 Plotting Learning Dynamics

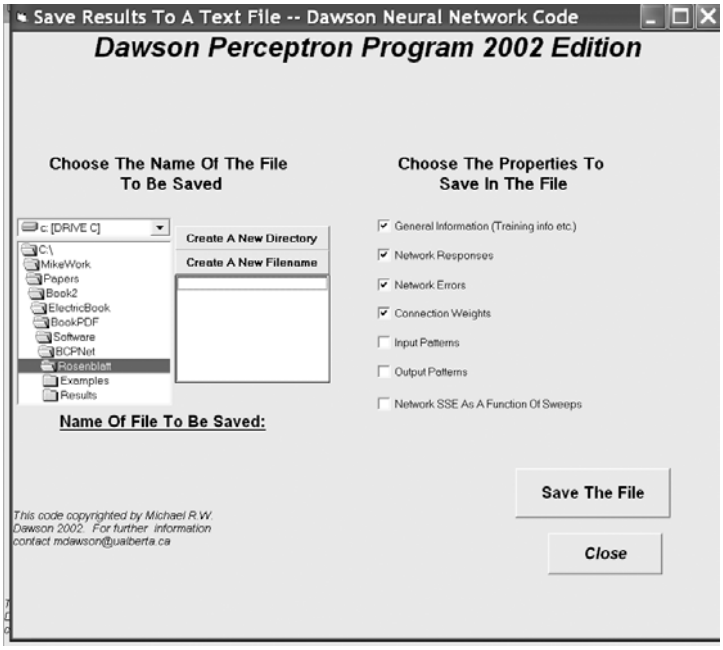
A comparison of the three learning rules for the perceptron might require examining how network error changes as a function of epochs of training. If the user chooses the "Plot SSE By Sweeps" option from the list in the network testing form, then the program automatically plots this information using a bar chart. An example chart is provided below. One can import this chart directly into a word processing document by simultaneously pressing the "Alt" and "Print Screen" keys on the keyboard (which copies the active window into the clipboard), going to the document, and pasting the clipboard into the document. One can print this chart on the default printer by left-clicking the mouse over the "Print The Graph" button. A left-click of the "Exit This Page" button closes the graph, and returns the user to the page that provides the options for testing network performance.



With respect to the graph produced in this form, the SSE axis is computed automatically, and the sampling of the bars across the Sweeps axis is determined by the choice of epochs between printouts made by the user on the program's second form. If the graph doesn't look quite right, then the user might consider re-running the simulation with a different choice for epochs between printouts. If a different kind of graph is desired, then the user might wish to save the network data to file. The data used to create this graph can be saved when this is done, and imported into a different software package that can be used to create graphs of different appearance.

10.4.3 Saving Results In A Text File

One of the options for storing information about network performance is to save network results as a text file. The form that permits this to be done, illustrated below, is accessed by choosing the list item "Save Summary As A Text File" from the "Test Network" page.



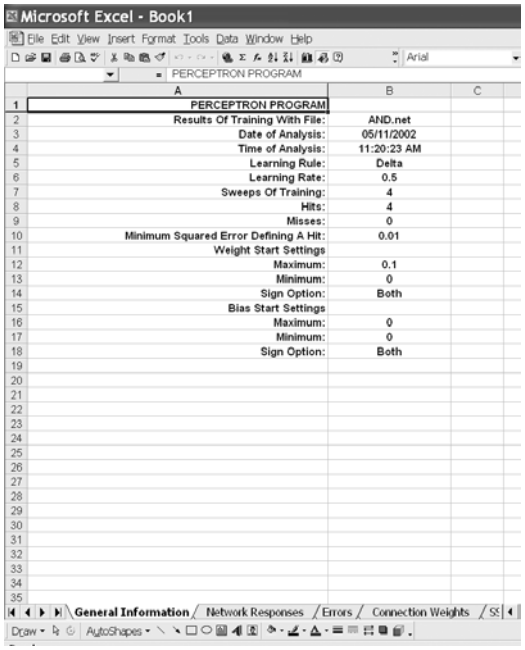
There are two sets of controls on this form. The first is a group of drive, directory, and file control boxes that are very similar to those found on the very first form seen when the program starts to run. One uses the drive and directory controls to navigate to a folder in which network data is to be saved. If it is necessary to create a new folder, a left-click of the mouse on the "Create A New Directory" button creates a dialog that permits the new directory to be named and created. Once the desired directory has been opened, the existing text files (.txt) in it are displayed. This is because the network data will be saved in such a file. One can overwrite an existing file by double-clicking it with the left mouse button. If a new file needs to be created, the dialog for doing so is accessed by a left-click of the mouse on the "Create A New Filename" button.

After choosing the location in which information is to be saved, the check boxes on the right of the form are set to determine what kinds of information will be saved. If a check box is not selected, then the corresponding information is simply not written to the file. To save the file, after the desired check boxes have been selected, the user left-clicks the "Save The File" button with the mouse. The form remains open after this is done, because in some instances the user might wish to save different versions of the network information in different locations. This form is closed by a left-mouse click on the "Close Button", which returns the user to the "Test Network" form.

10.4.4 Saving Results In An Excel Workbook

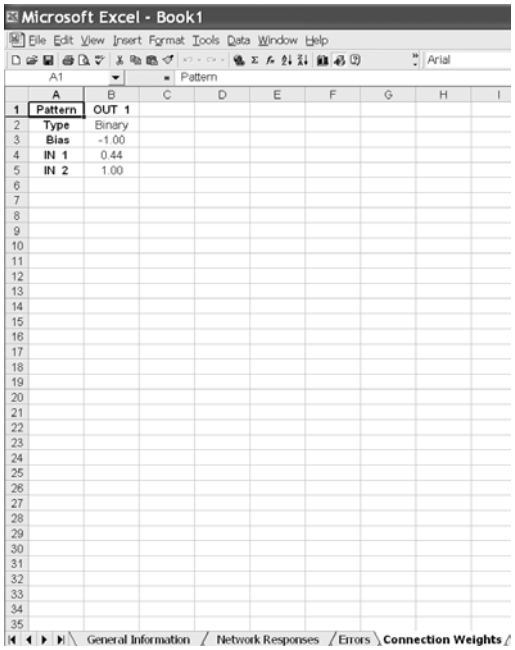
A second method for saving network performance is to save it in a structured Microsoft Excel workbook. This option is only available in the Rosenblatt program, and is not available in RosenblattLite or in RosenblattJava. It should obviously only be selected by users who also have Microsoft Excel installed on their computer. It is selected by a double-click of the "Create A Summary In Excel" list item that is offered in the "Test Network" form.

When this item is selected, a patience-requesting message is displayed on the "Test Network" form, and a number of different programming steps are taken to build an Excel Worksheet. When this is completed, the Worksheet is displayed as its own window, which will open on the user's computer in front of any of the Rosenblatt program's windows. If the worksheet has been created successfully, then the user should see something similar to the screen shot that is presented below.



All of the possible information that could be saved in the text version of a saved network is saved on this spreadsheet. Each different class of information is saved on its own worksheet in this Excel workbook. One can view different elements of this information by using the mouse to select the desired worksheet's tab on the bottom of the worksheet. The worksheet opens (as illustrated above) with the "General Information" tab selected.

When this workbook is open, it is running in Excel as a standalone program that is separate from the Rosenblatt software. One can select different tabs in the worksheet to examine network properties. For example, in the illustration belwo, the "Connection Weights" tab has been selected. After examining the worksheet, the user might wish to save it to disk. This is done by using the Save File utility provided by Excel.



One problem with having this information being displayed with a completely separate program is that it begins to use up memory resources on the computer that cannot be directly controlled by either program. For instance, it is possible to leave this workbook open, and to return to the Rosenblatt program. This practice is not recommended. Instead, potential system crashes are likely to be avoided by closing the Excel workbook before returning to Rosenblatt. When Rosenblatt is returned to, the "Test Network" form will still be displayed.

If saving Excel files from Rosenblatt causes system crashes, it is likely because of memory resource conflicts. The Excel options were built into Rosenblatt because they provide a convenient format for working with network data after training has been accomplished. The Excel data can also be easily copied and pasted into statistical packages like Systat. However, the Excel capability is not required for the distributed associative memory software to be used productively. If Excel problems are encountered frequently on your computer, our recommendation is to use RosenblattLite instead, and save network performance as text files only.

10.4.5 Leaving The "Test Network" Form

Once the user has finished examining the performance of a trained network, the list at the bottom of the "Test Network" form provides different options for network training. If the "Reset Weights And Train Again" option is selected, then all of the connection weights are randomized, the network is readied for training on the same problem that it has just learned, and the user is returned to the form that permits training parameters to be selected. If the "Keep Current Weights And Train Again" option is selected, the network is trained on the same problem, but the weights created from the learning that was just completed are not erased. The user is returned to the form that permits training parameters to be selected. They must be set again if settings other than the default settings are desired. If the "Train A New Network On A New Problem" option is selected, then the user is returned to the program's first form to be able to read in a new problem for training. If the "Train The Current Network On A New Problem" is selected, then the user can read in a new problem, but it will be presented to the network with the weights preserved from the previous training. This option can be used to study the effect of pretraining on learning a new problem. If none of these options are desired, then the program can be closed by pressing the "Exit Program" button with a left-mouse click.