CS 101 Lab 10

Artificial Neural Networks *Paul Cao*

In this lab, we will gain experience working with a neural network for character recognition. The lab report is due before class on Monday 4/18/2011. You need to download the data.zip from the lab 10 folder on angel. Unzip this file to your desktop to be used in the lab.

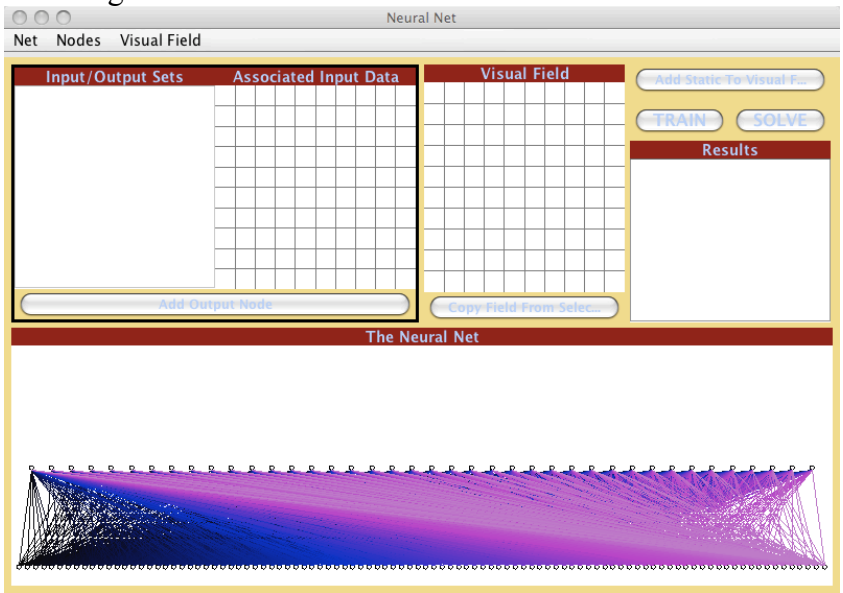
Background

A neural network consists of a set of input nodes, a set of output nodes, and one or more hidden layers of nodes that connect the input nodes to the output nodes. Input and output nodes can represent patterns, such as characters or images of faces, to be recognized by the network. For example, in the case of characters, pieces of a character in a visual field can map to individual input nodes in the network. To acquire knowledge of this pattern, the user trains the net by placing this pattern on its input nodes and running perhaps several iterations until the connection from input nodes to the output nodes reach a certain value. After the net has been trained to recognize a pattern, the user can present it with this pattern or a similar one as a trial input and ask the net to solve for the input pattern. If the input pattern is similar enough to one of the patterns represented in the network's set of output nodes, the net informs the user that it recognizes the input as the pattern represented by that node; otherwise, the net does not recognize the input pattern.

The neural net simulator

This simulator features a built-in network that can draw new output nodes (i.e. patterns to be recognized). We can train the net to recognize patterns and test new patterns on the net.

To open the neural network simulator, click **Neural Network** from the main menu. You should see the following window



The neural network window contains the following panes:

1. **Input/Output Sets** (upper-left pane). Here you can edit patterns to be added as output nodes to the net. You can transfer these patterns to and from files or from the **Visual Field (see below)**
2. **Visual Field** (upper-middle pane) . Here you can edit patterns to be presented as trial inputs to the net. You can transfer these patterns to and from files or from the **Input/Output Sets**
3. **Results** (upper-right pane). The results of training or solving are displayed here
4. **The Neural Net** (lower pane). Here is a visual representation of the neural net. The input nodes are at the bottom and the output nodes are at the top.

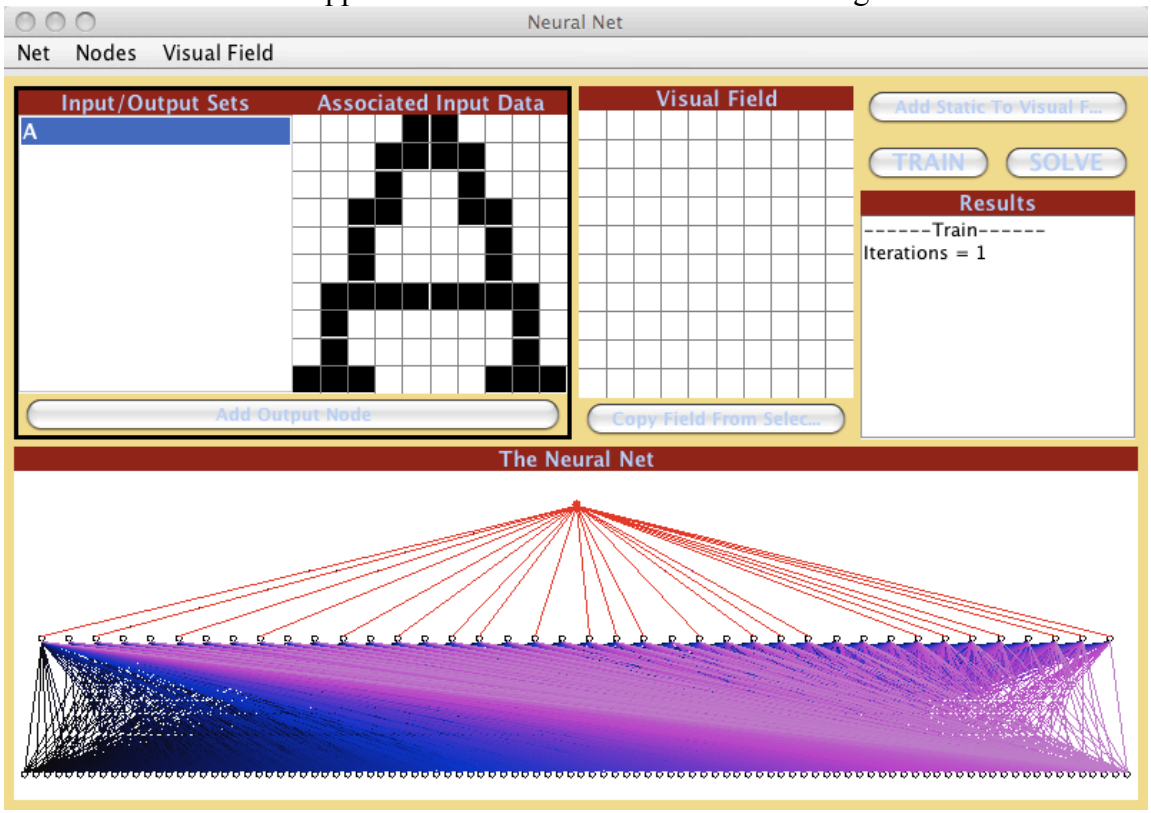
The three menus allow you to manage the net, the output nodes, and the visual field.

Exercise 1: Train the net to recognize letter A

A training session consists of entering the character to be learned and training the net. To get started, perform the following steps:

1. Click the **Add Output Node** button from the **nodes** menu.
2. Enter the letter A in the prompter, click the radio button **Visual Field from File**, then click **OK**
3. Select the file **Character 'A'.vfl** and click **OK**

Then click **Train**. Your application window should look like the figure below.



Note that the net now has a single output node. Each connection from this node to the net represents a black square in the pattern for the letter A.

Q1:/ How many iterations does it take for your neural net to learn this letter?

Exercise 2: Solve for a test character
There are three ways to create a sample pattern for input

1. Click the button **Copy Field fro Selected I/O Set**. Or
2. Open a visual field pattern from a file (**Visual Field** menu). Or
3. Draw the pattern by hand (click the squares in the visual field pane)

In here, we will use method 1. Copy the letter A from the Input Data to the Visual Field then click the **Solve** button.

Q2:/ What is the result?

Exercise 3: Training more than one character
The real power of a neural network lies in its ability to distinguish several different objects. To see how this is so, let's train a net to recognize the letters A, B, O, and Q. To clear the current network, select **New** from the **Net** menu. You will see a dialog box that allows you to specify the resolution of the visual field, the number of hidden nodes, and the percent of connectivity. Leave the defaults in place and click **Create the Net**. Then click **Add Output Node** four times to enter the four letters from their respective files. Click **Train** each time you added an output node.
Q3:/ How many output nodes are there in the net?

Q4:/ Is there any pattern that you observed on the number of iterations for each added output character? Explain briefly.

Try solving for each input letter after placing each letter in the visual field pane. Now click the **Add Static to Visual Field** button to add some static to the visual field with A in it. If you click solve,

Q4:/ What is the result (the output and the binary sequence)? Explain briefly.

In the end, load the bulls-eye into the visual field from the file and solve it.
Q5:/ What is the result? What kind of conclusions can you draw on the pattern recognition abilities of neural nets?

When you finished the lab, please log-off the computer.