CS295/CS395/CSYS395: Evolutionary Robotics

Programming Assignment 2 of 10

Assigned: Friday, September 9, 2011

Due: Friday, September 16, 2011 by midnight

Description: In this assignment you will be creating an artificial neural network (ANN). There are many kinds of ANNs, but they all share one thing in common: they are represented as a directed graph in which the nodes are models of biological neurons, and edges are models of biological synapses. Henceforth, the terms 'neurons' and 'synapses' will be used to describe the elements of an ANN. The behavior, in addition to the structure of an ANN is also similar to biological neural networks: (1) each neuron holds a value which indicates its level of activation; (2) each directed edge (synapse) is assigned a value, known as its 'strength', which indicates how much influence the source neuron has on the target neuron; and (3) the activation a of a neuron i at the next time step is usually expressed as

$$a_i = \sigma(\sum_{j=1}^{n} w_{ij} a_j) \tag{1}$$

where there are n neurons that connect to neuron i, a_j is the activation of the jth neuron that connects to neuron i, w_{ij} is the weight of the synapse connecting neuron j to neuron i, and $\sigma()$ is a function that keeps the activation of neuron i from growing too large.

In this project you will create an artificial neural network in Python, simulate its behavior over time, and visualize the resulting behavior.

Tasks:

1. **Back up your Python code from assignment 1.** Encapsulate your code from assignment 1 in a single file, Assignment_1.py, such that when you run it you can reproduce all of the visualizations. This will prove to you that that code is working fine, as you will use it in this and subsequent assignments.

2. Make a copy of Assignment_1.py and call it Assignment_2.py. You will use the matrix functions you developed in assignment 1.

3. First, we will create a neural network with 10 neurons. To do this, create a 50×10 matrix: element e_{ij} will store the value of neuron j at time step i. Name this matrix **`neuronValues`**.

4. Set each elements in the first row to a random value in $[0, 1]$: these values will represent the initial values of the neurons.

5. To visualize the network, we will place the 10 neurons in a circular pattern as shown in Fig. 1a. Create a new matrix **`neuronPositions=MatrixCreate(2,10)`** which will store the two-dimensional position of each neuron such that **`neuronPositions[0,i]`** will store the x-value for neuron i and **`neuronPositions[1,i]`** will store the y-value for neuron i.

6. Now, compute the positions of the neurons as follows:

```
angle = 0.0
angleUpdate = 2 * pi /numNeurons
for i in range(0,numNeurons)
    x = sin(angle)
    y = cos(angle)
    angle = angle + angleUpdate
```

7. Now, use this matrix and the plot() function to create the visualization shown in Fig. 1a. Hint: to create circles, you need to use `plot(...,'ko',markerfacecolor=[1,1,1], markersize=18)`. Copy and paste this visualization into your document.

8. To create the synapses, create a 10×10 matrix `synapses` and set each element to a value in $[-1, 1]$. A synapse with a negative weight *inhibits* its target neuron: the stronger the activation of the originating neuron, the lower the activation of the target neuron.

9. Create a plotting function that takes as input `neuronPositions` and draws a line between each pair of neurons. The resulting visualization should look like Fig. 1b. Store the resulting visualization in your document. **Note:** If you want to draw a line between the two points (x1,y1) and (x2,y2), you can use `plot([x1,x2],[y1,y2])`, and not `plot([x1,y1],[x2,y2])`.

10. Modify your plotting function such that it takes as input `neuronPositions` and `synapses`, and draws gray lines (plot(...,color[0.8,0.8,0.8])) for negatively-weighted synapses and black lines (plot(...,color[0,0,0])) for positively-weighted synapses. Store the resulting visualization, which should look like Fig. 1c, in your document.

11. Modify your plotting function again such that the width of each line indicates the magnitude of the corresponding synapse's weight. **Note:** The width of a line must be an integer value: e.g., plot(...,linewidth=2). To convert a synaptic weight to a number in 1,2,... use w = int(10*abs(synapses[i,j]))+1, and then plot(...,linewidth=w). Store the resulting visualization, which should look like Fig. 1d, in your document.

12. Now create a function that updates each neuron in the network: `neuronValues = Update (neuronValues,synapses,i)`. (You created `neuronValues` in step 3.) This function will iterate through each element in the ith row of `neuronValues`, and compute the sum

$$temp = \sum_{j=1}^{10} w_{ij} a_j \tag{2}$$

where a_j is the value of the jth neuron on the previous row, and w_{ij} is element e_{ij} in the matrix `synapses`. If this temporary sum is less than zero, round it to zero; if it is larger than one, round it to one. Store the result in `neuronValues`. **Note:** Before updating the neurons, make sure to set each neuron to a random value in [0,1]. **Thing to think about:** If the neurons are all set to zero initially, what do you think their values will be in subsequent time steps?

13. Now use the matrix imaging function you developed in the previous assignment to visualize how the neuron values change over time. This should produce an image similar to that shown in Fig. 1e. Copy and paste the image into your document. The image does not need to look exactly like that of Fig. 1e. In fact, re-run your program several times, and compare the images produced. Notice that the patterns of neuron activation vary greatly from one run to the next. We will discuss why this is the case in class.

14. **Things to think about:** Why do some of the neurons oscillate like this? What happens when all of the synaptic weights are set to -1, or to zero, or to 1? You do not need to include your answers in the document you submit for grading.

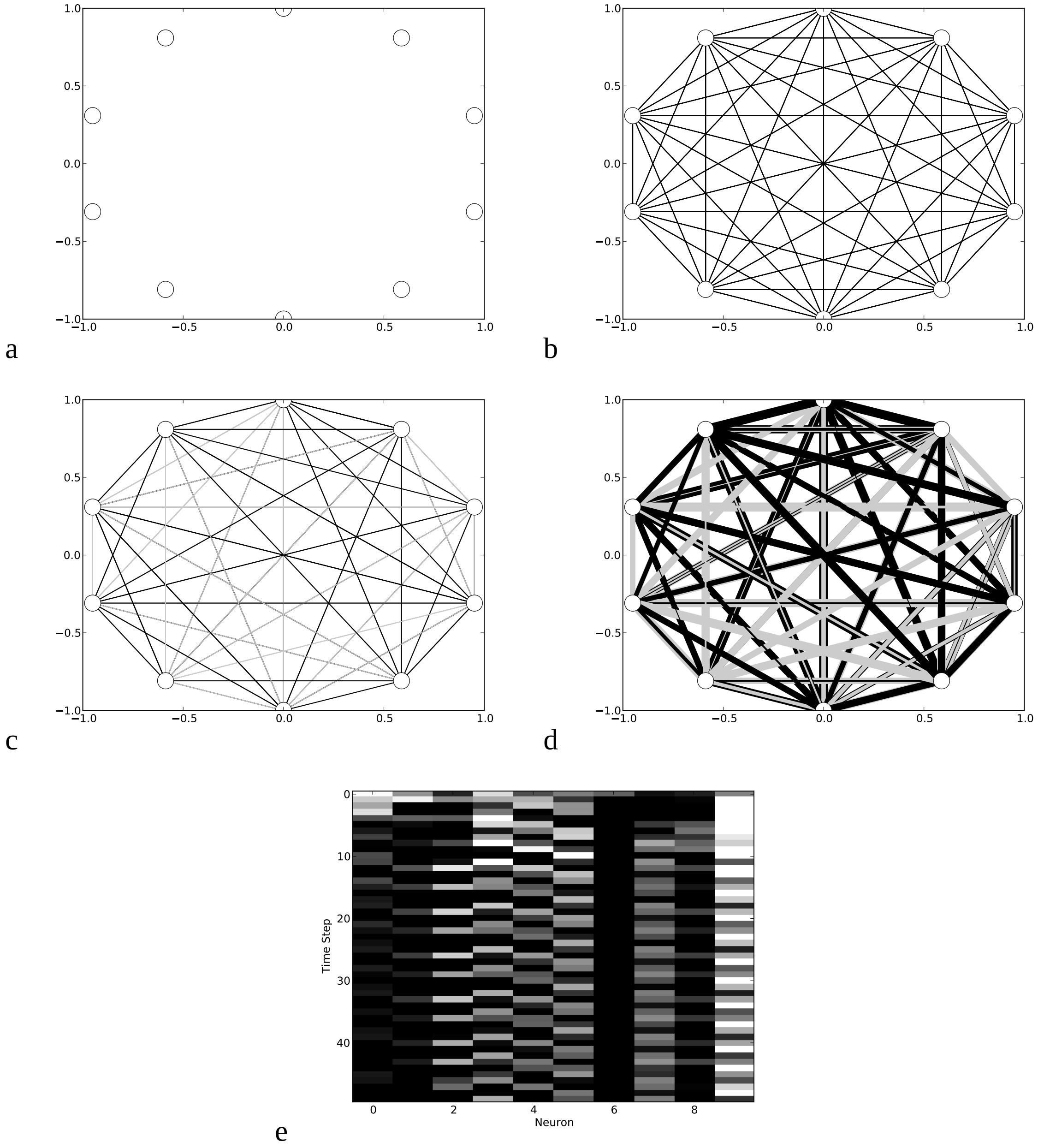

Figure 1: Visualizations demonstrating the successful creation of an artificial neural network. a: An ANN with 10 neurons and no synapses. b: An ANN with 10 neurons and $10 \times 10 = 100$ synapses. c: Gray and black lines indicate synapses with negative and positive weight, respectively. d: Line width indicates the magnitude of the synapse's weight. e: The pattern of neuron activity as time passes.