­Neural Network Lab, May 2017

Name:

Period:

Date Submitted:

**Instructions: Complete the parts below. You may leave the instructions in tact or delete them but please leave the 4 parts clearly separate. Submit final version on dropbox, as indicated on BB.**

**Part 1:**

Using the 14 linearly separable Boolean functions on 2 variables, how many 2-2-1 networks can you construct to calculate XOR? List all the solutions you find. How many unique solutions exists if you consider symmetry (e.g. ABC is just a mirror image of BAC)? Describe how your algorithm works.

(Note to list the solutions you should provide a numbered list of the 14 functions and their weights, as you learned them in your previous lab. Then solutions should take the form (x,y,z) where (x,y,z) is a triple of numbers identifying the functions.)

**Sample (delete this):**

14 learnable functions:

0: [-0.5, -0.5, -1.5],

1: [ 0.5, 0.5, -0.5],

2: [ 1.5, -2.5, -0.5],

3: [ 1.5, 0.5, -0.5],

4: [-1.5, 1.5, -0.5],

5: [-0.5, 1.5, -0.5],

6: [ 1.5, 1.5, -0.5],

7: [-0.5, -0.5, 0.5],

8: [ 0.5, -1.5, 0.5],

9: [ 1.5, -0.5, 0.5],

10: [-2.5, 0.5, 1.5],

11: [-0.5, 1.5, 0.5],

12: [-2.5, -1.5, 3.5],

13: [ 0.5, 0.5, 0.5]]

Solutions

1: 1 6 4

2: 1 7 7

3: 2 4 6

4: 2 9 9

5: 4 2 6

6: 4 11 9

7: 6 1 2

8: 6 12 1

9: 7 1 7 (repeated)

10: 7 12 4

11: 9 2 11

12: 9 11 12

13: 11 4 11

14: 11 9 12 (repeated)

15: 12 6 1

16: 12 7 2

**Part 2:**

Construct a NN using any topology you like to recognize coordinates within the unit circle. Implement your best model in code and compute its accuracy over 10,000 randomly selected points within the square -3/2 <= x,y <= 3/2. Describe your best model and report its accuracy.

Describe the first NN you came up with and how your design evolved into its final form. If your best model uses sigmoid actuator functions, what is the optimal “k” and “c” values you found (exponential coefficient, and rounding threshold). If your best model uses step actuators, how do you design the nodes and find their weights? Discuss your implementation challenges and how you determined your best model.

**Part 3:**

Implement an ad-hoc learning algorithm to discover the weights in a 2-2-1 network topology for XOR, using sigmoid actuator functions. Show the final weights learned by your algorithm (rounded and formatted nicely). Write pseudocode that clearly describes how your search algorithm works. Also discuss how your ideas evolved and any failed attempts along the way.

**Analysis**: Explore the convergence rate of your algorithm as a function of the learning rate. Count the number of iterations required for your algorithm to converge, averaged over several samples (say 10-100 per trial), as you vary the learning rate, lambda, and produce a graph. The appropriate range of lambda will depend on your implementation, but generally a lambda of “1” means you add “1” to the weights at each step. Find the approximate lambda which seems to minimize convergence time.

Describe how your algorithm depends on lambda and show a graph of “lambda vs. iterations”.

**Part 4: (Bonus/Extension)**

Modify your learning algorithm (or create a new one) to find optimal weights for part 2. You may also want to include “k” and “c” in your search space. Describe your results.