Bio-Inspired Artificial Intelligence
Report - Assignment 3 - 03/31/17
Perceptron
Dr. Prof. J. Marshall
Alexander Jermann

Assignments

1. The previously defined set of weights (0.1, 0.2, -0.3) only yields a TSS error of 0.0 for AND targets. Using the same weight values on the other logical functions renders TSS error rates ranging from 2.0 to 4.0.

TSS error rates	
ANDtargets	0.0
ORtargets	2.0
NANDtargets	4.0
NORtargets	2.0
XORtargets	3.0

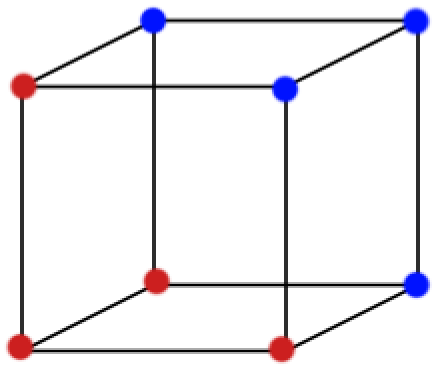
2. See the list below, where the TSS error values, produced by the weights in respect to the targets listed in the exercise, are listed.

weights		bias	TSS error
0.1	0.2	0.3	1.0
0.0	0.0	0.0	1.0
-0.1	-0.2	-0.3	3.0
1.2	-2.3	-0.2	2.0
-0.3	0.3	-0.3	4.0
0.2	-0.1	0.1	1.0

3. From 2 (see above), we can determine that the perceptron that produced the highest TSS error for the targets [1, 0, 1, 1] was the one with the weights (-0.3,0.3,-0.3).
Using a learning rate of 0.1 and the train function it took the perceptron 15 epochs to learn the following correct set of weights: +0.2 -0.1 +0.0 (meaning no bias). When verified with the test function these weights returned a TSS error of 0.0.

4. The new target values were defined as NTWOtargets. It took 7 epochs to retrain the perceptron to fit the new target values. The resulting weight and bias values were: -0.1000 +0.1000 -0.1000. Following the theorem of perceptrons, the perceptron can be retrained on other logical functions as well as long as they're linearly separable. My tests were consistent with the theorem. Training the perceptrons to fit the targets always yielded a set of corresponding weights. The only time the train function did not seem to be able to find weights that rendered a TSS error that was not higher than 0.0 was trying to train the perceptron to fit the XORtargets. This follows directly from the fact that the XOR logical function is not linearly separable.

5. Yes and No. I came to the conclusion that the Carry perceptron can be trained, while the Sum perceptron *cannot* be trained.
Carry Perceptron: In my experiments the Carry perceptron was fully trained after 8 epochs. The resulting weights are as follow: +0.1677 +0.1720 +0.0334 -0.1924.
Sum Perceptron: In my second experiment the Sum perceptron did not seem to find any weights and bias that renders a error rate of zero, meaning that the algorithm was running in an 'endless' loop. The answer to this seemingly peculiar behavior can be explained using a cube.



If we interpret the inputs as coordinates in a three dimensional space and connect the dots with lines, we obtain a cube. We label each vertex with the target values corresponding to the coordinates (input values). In the figure above the red dots represent a target value of zero and the blue dots represent a target value of one. Following the theorem of Perceptrons we recognize that there is no plane that divides the blue and red dots into separate sets. In other words it is not linearly independent.

6. See submitted code

7. The experiment was run 50 times, in order to get a good average value of epochs needed to train the perceptron and to obtain a error rate percentage that is meaningful. Using the first 277 entries in the dataset it took an average of 17 epochs to train the perceptron. Testing the perceptron on the last 277 data entries resulted in an average TSS error of 6.04 and a percentage error rate of wrong classifications of 2.18%. The lowest TSS error was 3.0 and the highest was 8.0. The shortest amount of epochs to train the perceptron was 4.0 and the longest 26.0.