**Assignment 4 write-up**

CISC352 Artificial Intelligence

April 4th 2017

Anna Ilina (10150979)

Jake Pittis (10148335)

**Problem 1. Artificial Neural Networks: ORPerceptron**

*Four-Input OR-Training Using a Single-Layer Perceptron Network.*

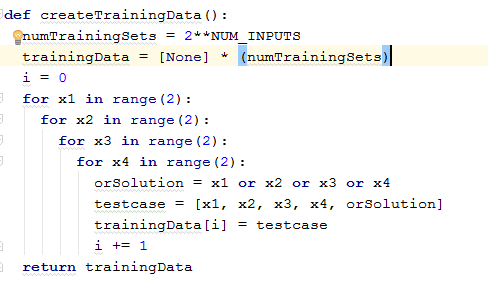
This problem was solved in Python 2.7, using the Random library.

The solution to this problem was pair-programmed by Jake and Anna.

**Creating the training dataset**

To create training data for the perceptron, I create an array of all the possible input sets to the 4-input OR function in the *createTrainingData()* method, and stored the OR target solution, in the format

[x1, x2, x3, x4, ORsolution]



**Structure of the perceptron training function**

Next, the perceptron was trained in the *trainPerceptron()* method. For training, 4 weights were used, corresponding to the 4 inputs. The weights were initialized as random values between -1 to 1 using the random.uniform() function. The program repeatedly loops through each of the 16 training sets, adjusting weights according to a step activation function when an input set is incorrectly classified, until it goes through a loop where all the training sets are classified correctly; at this point the perceptron is fully “trained” and the values of the weights are returned.

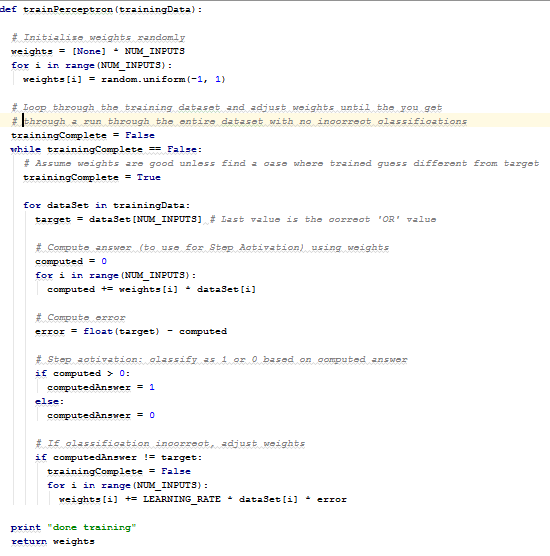
**Step activation function**

To train our perceptron, we used a step activation function with a threshold of t=0 and a learning rate of a=0.1. For inputs xi and weights wi, if the value Σ(xi \* wi) was above the threshold, the input set was classified as ‘1’. Otherwise, the input set was classified as ‘0’. If the input set was incorrectly classified, the weights were adjusted according to the following formula, where e is the error value (e = expected classification

𝑤i ← 𝑤i + (𝑎×𝑥i×𝑒), 𝑤𝑤

where e error from the expected classification (0 or 1), obtained by the forumula:

e = (expected classification) - Σ(xi \* wi)



**Using trained perceptron to classify data**

The weighs from the trained perceptron are used to classify input data using a step activation function in the *classifyData()* method.

