**Perceptron Learning Lab** Name and period: Ayushmaan Ganotra and Bamit Joshi Period 4

**For boolean functions on n variables, a perceptron can learn:**

1. N=2:
   1. Learned: 14/16 = 87.5%
   2. LEARNABLE - 0000: [ 1, 0, -1] learnable, 0001: [ 2, 1, -2], learnable
   3. UNLEARNABLE - [-1, 0, 1] 0110
2. N=3:
   1. Learned: 104/256= 40.625%
   2. LEARNABLE – 00000100 - [ 2, -3, 1, -2] and 00000011 [ 2, 1, 0, -2]
   3. UNLEARNABLE -- 11111001 - [-1, 2, 0, 1] and] 11110110 - [-2, -1, 0, 3]
3. N=4:
   1. Learned: 1882/65536= 2.8717%
   2. LEARNABLE – 0000010101011101 - [ 3, 4, -2, 6, -6] and 0000000011111111 [1, 0, 0, 0, 0]
   3. UNLEARNABLE -- 0000001100100001 -[ 0, 3, 2, 0, -3] and 0000001011101110 - [ 2, 1, -1, -2, 0]

Create a training set and a testing set over 10 boolean inputs (x) where the function f(x) = majority. Use a training size of about 100 vectors. Plot the accuracy of a perceptron and of a decision tree, each on the testing set, for the target concept. The x-axis should be “training set size” and the y-axis “accuracy on test set”. Plot both functions on the same set of axes.

For the perceptron: use as many epochs as you deem necessary. For both: do NOT test on the training data!

