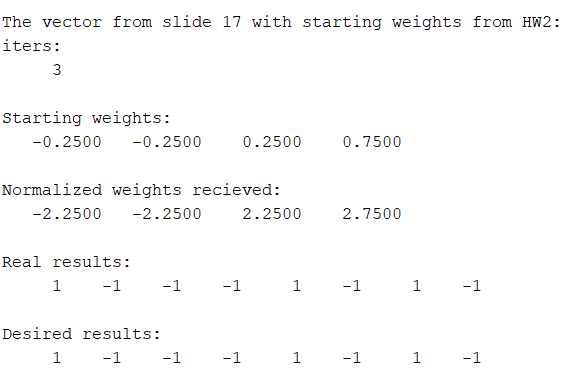
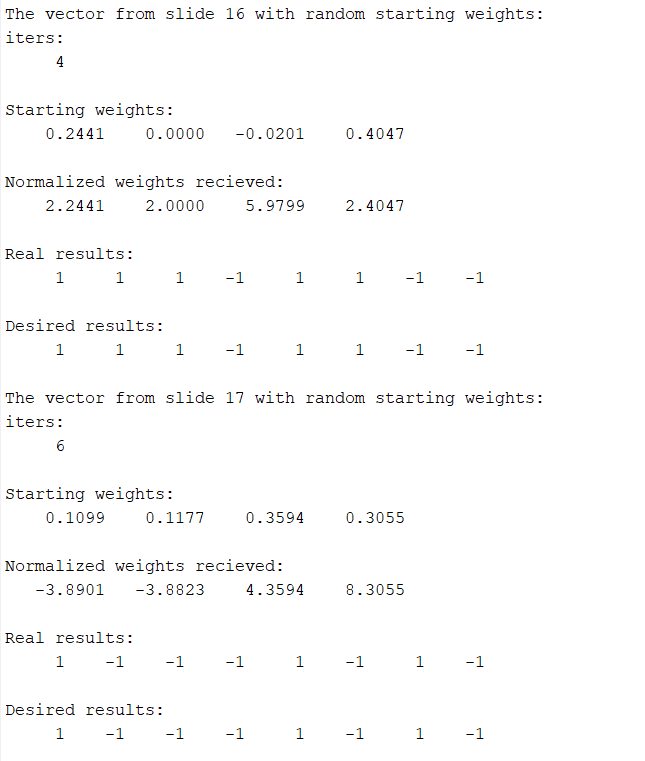
Alex Gibbons

CMPT 465

Project 3

Due October 5, 2023

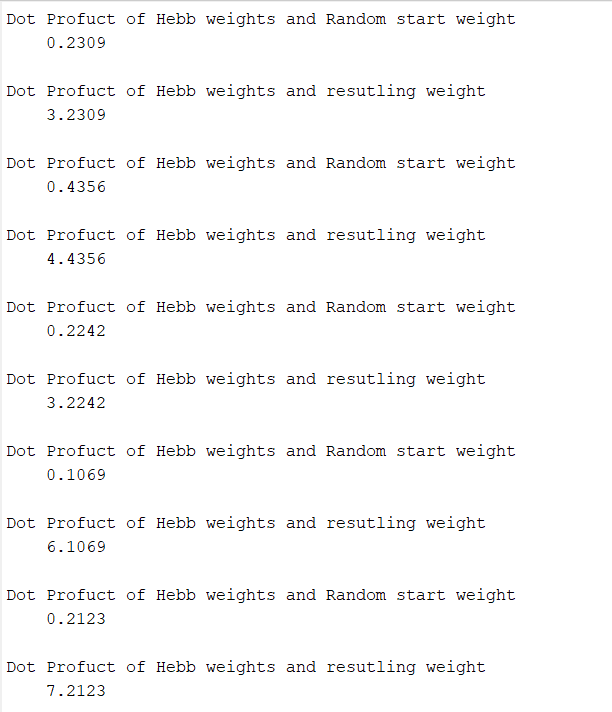
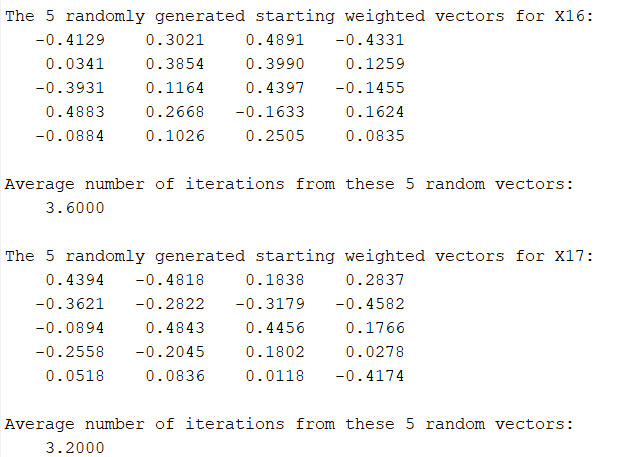
Question 2)

Both vectors from slides 16 and 17 were correctly implemented by the error-correction learning rule.

The starting weights and resulting weights were often similar in that they shared digits after the decimal point, however, this was not always the case.

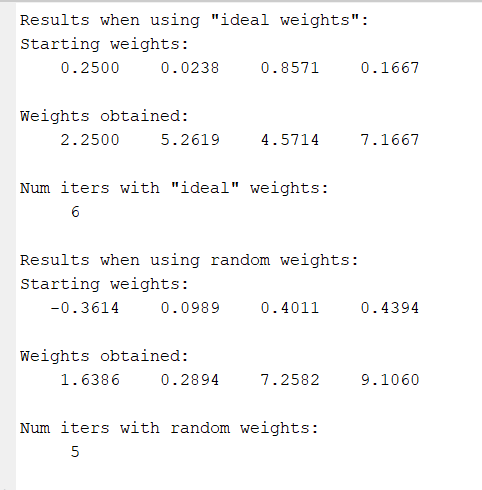
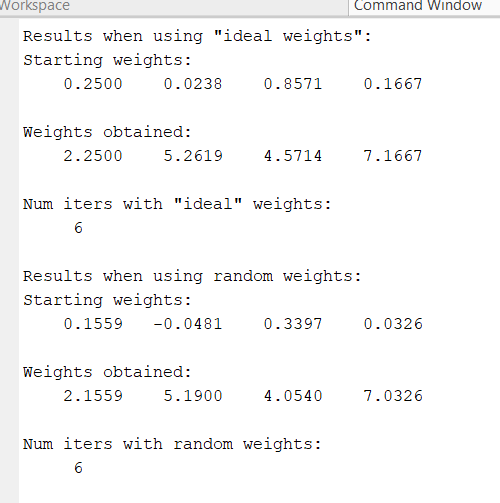
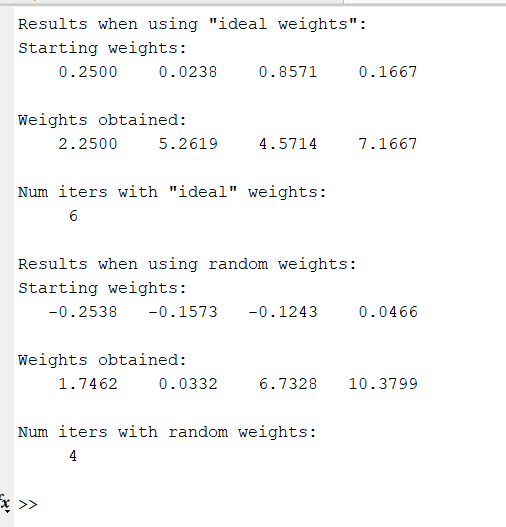
When using randomly generated starting weights, the function finished typically anywhere from 3 to 6 iterations, in this case it was 4 and 6 respectively.

When using the normalized Hebbian weights obtained in Homework 2, the function finished in only 3 iterations and the resulting weights do not resemble much likeness to the ones received when using randomly generated weights.

Question 3) 

When using randomly generated weights, the function finished on average 3.6 iterations for the vector from slide 16 and 3.2 iterations for the vector from slide 17.

When comparing the dot products of the Hebbian weighting vector and 5 starting random weighting vectors with the dot products of the same Hebbian weighting vector with the corresponding 5 resulting weighting vectors a similarity between the products can be seen. They both share similarities in the digits following the decimal point. However, I ran this test multiple times and concluded that although it is a common occurrence, it does not happen every time.

Question 4)

For this scenario, I used the results from 3 different randomly generated weighting vectors to have a better understanding of the results. When using the weights obtained from Homework 2, it finished in 6 iterations, however, on average the randomly generated weights finished in 5 iterations. This tells me that the Hebbian weights are not ideal starting weights. Another note I made is that the starting weighting vector and resulting weighting vector are much less likely to show similarities when the input type is real compared to binary.