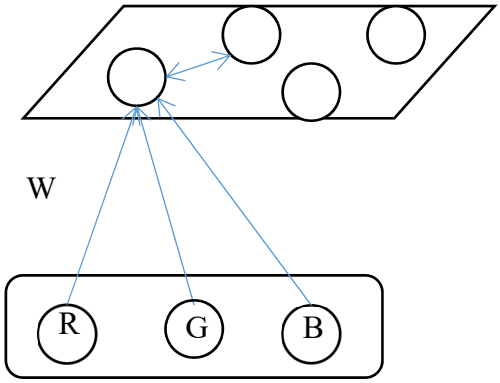
Q355
Week 4 assignment

Kohonen Self-organizing feature map (25 pts)

Consider a Kohonen self organizing feature map (SOFM), as described in Kohonen (1982). Consider a map with 2500 elements arranged in a 50x50 rectangular grid (matrix). Each input vector has 3 elements in the continuous range of [0,1] that correspond to a signal of one of the primary colors (red, green, blue). For example, (1, 0, 0) is an input pattern that corresponds to the color red.



Step 1. Implement a simple SOFM in python following the equations described in Kohonen (1982), but simplified as follows. Assume a weight matrix W of dimensions 50x50x3, that connects each input to each SOFM unit. The input to each SOFM unit is the external input plus the lateral input:

External input = W*inputs (this is simply an inner product of weights and inputs, e.g. the argument of equation 4 in Kohonen, 1982).

Lateral input from one SOFM unit A (with activity a) to another B:

a * 8, if the Euclidean distance between A and B is < 3

a * -1, if the Euclidean distance between A and B is between 3 and 8

0 otherwise

(note you should assume that the edges "wrap around")

The total activity "eta" (η) of each SOFM unit is a piece-wise sigmoid (Kohonen, 1982):

0, if the sum of the external and lateral input is <= 0

5, if the sum of the external and lateral input is >= 5

The sum of the external and lateral input, otherwise

Weights W from the external inputs to the SOFM units are adjusted by the following rule (cf. Kohonen, 1982, eqn 12):

W(t+1) = W(t) + learning*eta*inputs, divided (normalized) by the Euclidean length of the input weights to the SOFM unit. The learning rate (learning) is 0.01.

(9 points for writing code)

Step 2. Train the SOFM by presenting a "red" input 2/3 of the time and a "green" input 1/3 of the time, all intermixed randomly. Present 500 trials for training, and observe the weights by plotting them in RGB color after each trial. Notice how the representations change over time. Once the network has finished training, plot a screenshot of the color image. Describe the representations, and how they are distributed topographically. (5 pts)

Next, present a "red" input to the network, and calculate the total SOFM layer activation, summing over all 2500 cells. Then present a "green" input, and calculate the total SOFM layer activation for that. How do you account for the activations you found for each input? Why and how are they different? (3 points)

Step 3. Perform the training as in step 2 above, but this time present a "green" input 2/3 of the time and a "red" input 1/3 of the time. Plot a screenshot as before. How do you account for the difference between the screenshot of Step 3 and the screenshot of Step 2? (3 points)

Step 4. Compare the results of Steps 2 and 3 to the results of Jenkins & Merzenich (1990). How accurate (or inaccurate) of a model is the SOFM with regard to the phenomena described in Jenkins & Merzenich? (5 pts)