Lecture 7.3 Exercises: Cholinergic Modulation

See the text file 'readme.txt' for useful installation and setup information.

Run the base model by typing:

>> AChAssociativeMemory

You should see a few windows appear. Windows on the left are the input patterns (top), output pattern to which they should be associated (left) and initial weight/connection matrix (center). These 3 windows are used to learn/train the network. Windows on the right are the test/recall windows: The cues (partial inputs) and on top, the output patterns are on the right, and the connection matrix is again plotted in the center window. In addition to these windows, you will see a central slider/window representing the level of Ach ([0 1]).
Note: In some of these windows, you will see a 'change' button. Clicking it will allow you to toggle a neuron on/off. The 'recompute' buttons re-learn (left) or re-test (right) the network.

1. Test recall. As is, the recall is perfect (the output patterns are exactly those associated with the partial input pattern presented). Toggle a few cue neurons, one at a time and retest.
    a. Can you generate a perfect output with as little as 1 neuron in each cue?
    (ans: yes. Pattern1=4, pattern2=2)

    b. Can you generate a perfect output with overlapping input cue patterns?
    (ans: yes. Pattern1:[0,0,1,1,0,0,0,0,0,1])

    c. Can you generate a partial output pattern for at least one of the two patterns?
    (ans: extremely difficult. If you do, let us know! This indicates that the 'attractors' are deep and that the patterns are well separated)

2. Test learning. Note that the input patterns overlap, but the associated patterns do not. Introduce overlaps in the associates.
    a. Start with 1 neuron overlap. What do you see in the weight matrix?
    (ans: the weights become heterogeneous. In general, the network should still do a good job recalling the correct patterns)
    b. Try 2,3,4 neuron overlaps. How does the performance change?
    (ans: again, even with large overlaps, the recall is virtually perfect.)

3. Close all the windows (>> close all), and restart the base simulation. Now reduce the cholinergic modulation by about ½, and re-learn.
    a. What do you see?
    (ans: the recall is strongly affected. One of the pattern may be correctly recalled, while the other is not at all)

    b. Can you 'rescue' the recall by introducing more cues?
    (ans: not easily)

4. Edit AChAssociativeMemory.m, and add a third pair of patterns (apre3, apost3, acue3) to the mix (for convenience, a set has been included and commented out. Just uncomment, before trying your own). Close all windows, and restart. You should see pattern 3 perfectly recalled even though the cue contains only 1 neuron and the output pattern is in complete overlap with associate-pattern 1 (recalled as pattern 2).

5. Progressively decrease the levels of Ach. Observe the recall performance.
(ans, about ¼ of the way down, recall pattern1 (but not 2 and 3) becomes noisy/incorrect, and worsens. The other patterns start to degrade at about 30% Ach when recall-pattern3 becomes identical to recall pattern2 (because of their overlap). Further decrease in Ach yields worse performance, although recall-pattern2 seems unaffected throughout and remains correct)