Lab 2
12 Feb 2009

Yesterday, we briefly discussed a type of connectionist (i.e. neural net) system known as “Kohonen nets” after their inventor. These maps have a few key features:

- The algorithms are unsupervised; that is, they are provided with unlabelled training data.
- There is an imposed topological structure to the neurons in the network, and a neuron knows which other neurons it’s “adjacent” to in the graph.

The basic mechanism is that once the initial neural net weights are (randomly) generated, for any given data point, one neuron will fire most strongly. That neuron is trained to move closer to the datum, and neurons in the neighbourhood of the firing neuron are also moved closer to the datum. Furthermore, the nodes one more step away from the firing neuron are inhibited, pushed slightly away from the datum (but don’t implement this part at first).

I’ve written a GUI to help you display this: it loads in and displays data and has stubs for the net’s methods. The rest is up to you.

Your job is to write \texttt{(update-net! net datum)}. The net is in the form I just described; the datum is a list of numbers (representing a vector in $d$-space). You need to modify the appropriate elements of the net, and return the result.

The starter code is \texttt{kohonen/} in the course directory.

The way this code is set up, it’ll be easy to try it out with other random or nonrandom data sets—I’ve given you a few already—and see how this affects the resulting net. Also, you can try adjusting your $\alpha$ value either a priori or on the fly once some initial learning has been done.