a i - j u n k i e

Kohonen's Self Organizing Feature Maps

"I cannot articulate enough to express my dislike to people who think that understanding spoils your experience... How would they know?"

Marvin Minsky

Introductory Note

This tutorial is the first of two related to self organising feature maps. Initially, this was just going to be one big comprehensive tutorial, but work demands and other time constraints have forced me to divide it into two. Nevertheless, part one should provide you with a pretty good introduction. Certainly more than enough to whet your appetite anyway!

I will appreciate any feedback you are willing to give - good or bad. My ears are always open for praise, constructive criticism or suggestions for future tutorials. Please drop by the forum after you've finished reading this tutorial and let me know what you think... reader feedback is one of the things that makes maintaining a site like this worthwhile. You can find the forum here.

Overview

Kohonen Self Organising Feature Maps, or SOMs as I shall be referring to them from now on, are fascinating beasts. They were invented by a man named Teuvo Kohonen, a professor of the Academy of Finland, and they provide a way of representing multidimensional data in much lower dimensional spaces - usually one or two dimensions. This process, of reducing the dimensionality of vectors, is essentially a data compression technique known as *vector quantisation*. In addition, the Kohonen technique creates a network that stores information in such a way that any topological relationships within the training set are maintained.

A common example used to help teach the principals behind SOMs is the mapping of colours from their three dimensional components - red, green and blue, into two dimensions. Figure 1 shows an example of a SOM trained to recognize the eight different colours shown on the right. The colours have been presented to the network as 3D vectors - one dimension for each of the colour components - and the network has learnt to represent them in the 2D space you can see. Notice that in addition to

clustering the colours into distinct regions, regions of similar properties are usually found adjacent to each other. This feature of Kohonen maps is often put to good use as you will discover later.

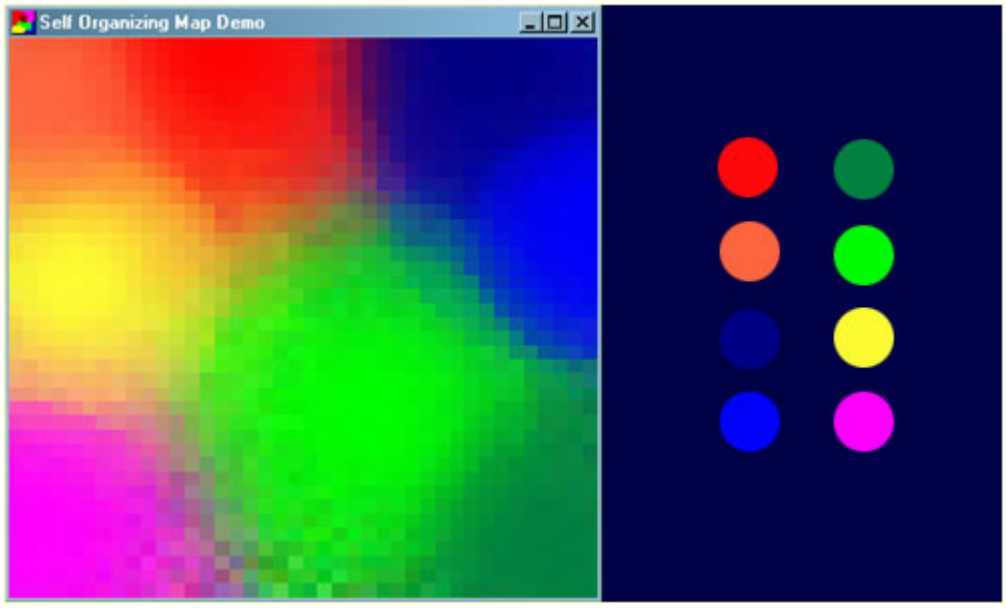

Figure 1
Screenshot of the demo program (left) and the colours it has classified (right).

One of the most interesting aspects of SOMs is that they learn to classify data *without supervision*. You may already be aware of supervised training techniques such as backpropagation where the training data consists of vector pairs - an input vector and a target vector. With this approach an input vector is presented to the network (typically a multilayer feedforward network) and the output is compared with the target vector. If they differ, the weights of the network are altered slightly to reduce the error in the output. This is repeated many times and with many sets of vector pairs until the network gives the desired output. Training a SOM however, requires no target vector. A SOM learns to classify the training data without any external supervision whatsoever. Neat huh?

Before I get on with the nitty gritty, it's best for you to forget everything you may already know about neural networks! If you try to think of SOMs in terms of neurons, activation functions and feedforward/recurrent connections you're likely to grow confused quickly. So dig out all that knowledge from your head and shove it to one side before you read any further. Done that? Great, let's get on with the tutorial then...

You can download the accompanying source code from here. For those of you without compilers the zip file also contains an executable.

(Update: A reader, *Kintar*, has also submitted a Java version. You can grab it here)

Network Architecture

For the purposes of this tutorial I'll be discussing a two dimensional SOM. The network is created from a 2D lattice of 'nodes', each of which is fully connected to the input layer. Figure 2 shows a very small

Kohonen network of 4 X 4 nodes connected to the input layer (shown in green) representing a two dimensional vector.

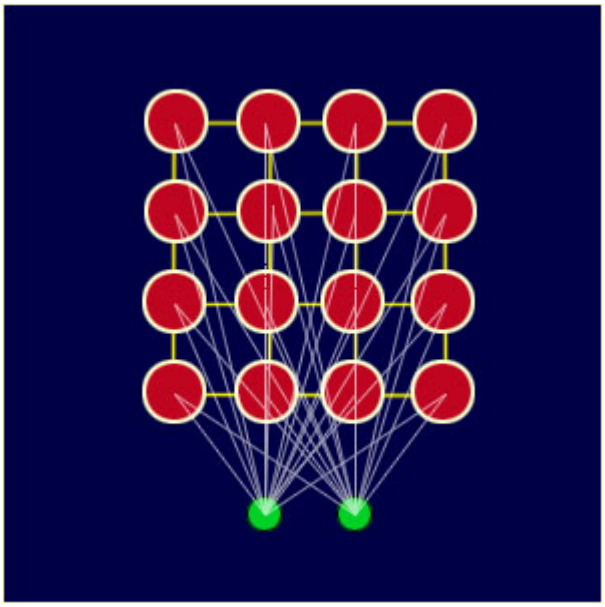

Figure 2
A simple Kohonen network.

Each node has a specific topological position (an x, y coordinate in the lattice) and contains a vector of weights of the same dimension as the input vectors. That is to say, if the training data consists of vectors, *V*, of *n* dimensions:

$$V_1, V_2, V_3...V_n$$

Then each node will contain a corresponding weight vector *W,* of *n* dimensions:

$$W_1, W_2, W_3...W_n$$

The lines connecting the nodes in Figure 2 are only there to represent adjacency and do not signify a connection as normally indicated when discussing a neural network. *There are no lateral connections between nodes within the lattice.*

The SOM shown in Figure 1 has a default lattice size of 40 X 40. Each node in the lattice has three weights, one for each element of the input vector: red, green and blue. Each node is represented by a rectangular cell when drawn to your display. Figure 3 shows the cells rendered with black outlines so you can clearly see each node.

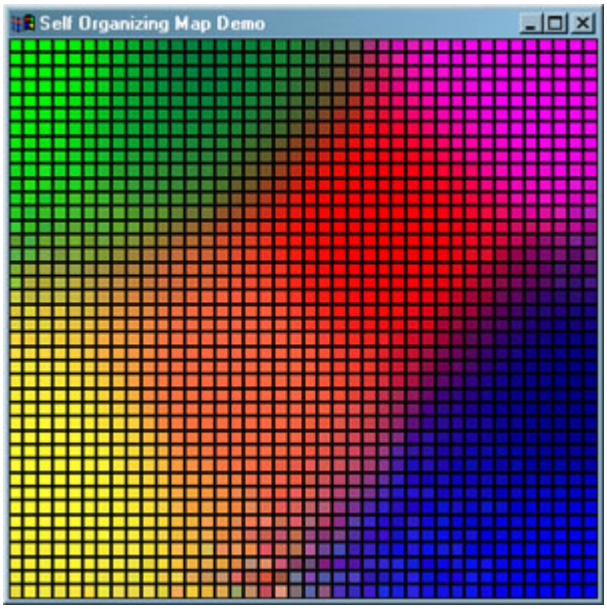

Figure 3
Each cell represents a node in the lattice.
