Self-organised feature mapping.

The VQ, LVQ systems have no structure (topology) on their output units whatsoever. All the output units are equally close (or distant).

We would like to have a concept of nearness in the output units, so that *nearby patterns could have correspondingly nearby classifications*. Obviously, this is possibly only if *nearby* has some meaning on the output layer.

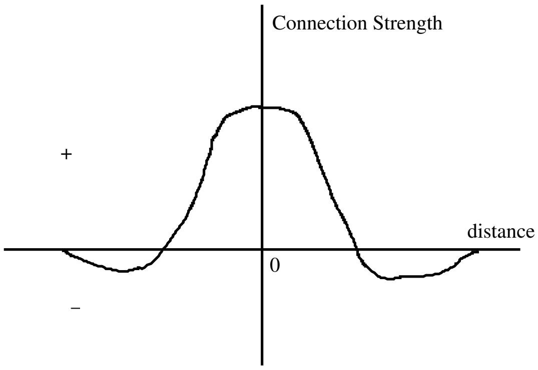
What we seek is thus a *topology-preserving map*.

These occur in biology: for example,

retinotopic maps in the visual cortex
tonotopic maps in the auditory cortex
"body" map in the somatosensory cortex.

What form should the output unit cluster take to achieve this?

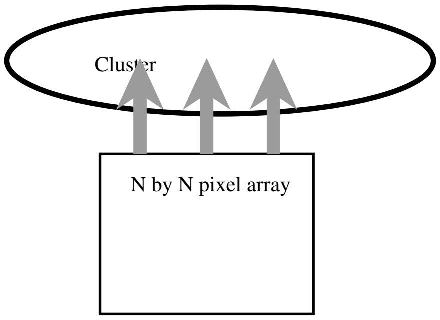
In 1973, von der Malsburg proposed a self-organised bar detector with just these properties. He proposed that the output units should have *mexican-hat* lateral connections:



Each unit excites itself and its near neighbours. It inhibits units which are further away, and (slightly) excites very distant neighbours.

As with the WTA network, these lateral connections are non-adaptive.

Bar-detector architecture.



Only the upward connections are adaptive, and the learning rule used is a Stent-Singer postsynaptic rule.

The stimuli used are bars, centred in the centre of the pixel array.

The effect is that

* bars of similar orientation receive similar codings
* coding is *continuous*: even unseen bars elicit correct responses.
* the system displays hyperacuity, as a result of the coarse coding it uses.

(slide of results)

ohonen self-organising maps.

These extend von der Malsburg's network concept: Kohonen realised that the topology of the output units depended on the exact form of the lateral interconnections.

These units can be connected
* in a line
* in a circle

or in a net of more than one dimension - e.g. in an open or closed regular mesh.

In fact, Kohonen's algorithm use *implicit* network interconnections (rather than the explicit organisation used by von der Malsburg). There are no actual lateral connections but there ***is*** a concept of a neighbour. The learning algorithm is

(i) Find the unit i such that d($\mathbf{W}_i$,$\mathbf{X}$) is minimal. (i.e. find the winner)
(ii) adjusts the weight vector for both i and its neighbours.

It is the definition of the *neighbours* that defines the topology.

Kohonen's clustering algorithm

Repeat
(1) Supply the input vector
(2) Find the winning unit
(3) Modify the weights:
$\Delta w_{kj} = \eta \Lambda(k,i)(X_j - w_{kj})$
until stability is achieved.

Notes:
If the **X** and the $\mathbf{W}_i$ are normalised, then the winner election is equivalent to choosing the unit for which the dot product $\mathbf{X}.\mathbf{W}_i$ is maximised. But this is often not the case.

$\Lambda(k,i)$ is the *neighbourhood function*. If $\Lambda(k,i) = 1$ iff k=i, and is 0 elsewhere, the algorithm reduces to LVQ

(information sheets on K-nets, and topologies)

This is a much used network. It is not without its difficulties:
Convergence can be slow: often two oppositely directed maps start to form, and the network can be difficult to untwist.
It is often difficult to decide what the best output unit topology should be, both in terms of size, and of what $\Lambda(k,i)$ should be.

Using the Kohonen clustering algorithm.

1. When to use it.

The Kohonen clustering algorithm is particularly useful when the input data has high dimensionality, but is believed to form into a number of clusters.

The network projects the inputs on to an output layer. This results in one unit winning. However although in training, one unit is normally chosen as the winner, both it, and its neighbours are updated to point towards the input vector.
On test, the network will give output over some **range** of output units. This can be used as the network's output.

2. How to use it.

Given a set of input data of some dimension, N, it is not clear how many units the output layer should have, nor is it clear what topology should be used.

In general, one should use a much larger number of output units than the number of clusters one is expecting to find. Each cluster will normally extend over a number of output units.

This still does not help in deciding the topology of the output layer. One needs to consider the form of the data in deciding whether
* the surface should be closed or open
* the dimensionality of the output layer.

3. Why is it so popular?

The Kohonen network is probably the most used neural network after the BP network.

This is because the projection it carries out maintains some of the topology of the input space. Thus, even although the number of output units is large (and generally larger than the number of input units), the actual output from the network for any one input pattern is normally spread across only a small number of units. It can thus be used to find clusters in the data.

It thus tackles *the curse of high dimensionality*.

In addition, it is self-organised, and is relatively easy to apply.