Cortical Maps

Neurons in cortex are often organized in ***cortical maps***:

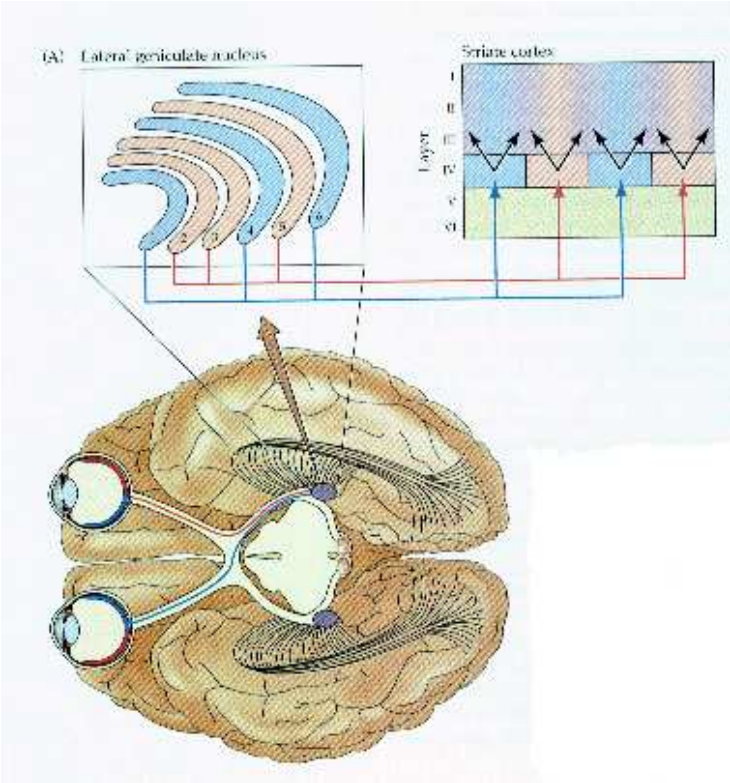
- Neurons with similar responses to some stimulus variable tend to be located at nearby cortical locations.

Examples of cortical maps can be found for example in primary visual cortex (V1).

There, neurons are mapped for example with respect to

- spatial location (topographic map)
- sensitivity to the left or right eye (occular dominance map)
- orientation (orientation map)
- color sensitivity (so-called blobs)

LGN and V1



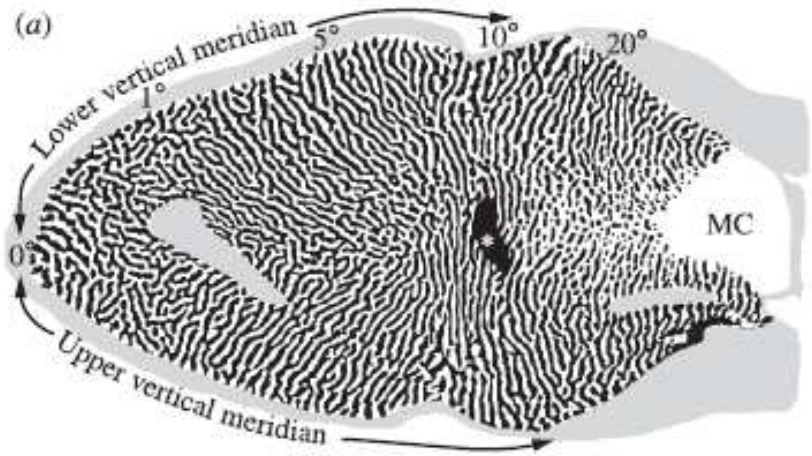
Ganglion cells of both retinas project to the *lateral geniculate nucleus (LGN)* of the thalamus.

The LGN of each hemisphere obtains projections from both eyes. However, neurons in LGN are only sensitive to one eye (monocular).

Neurons in LGN are separated based on their occularity.

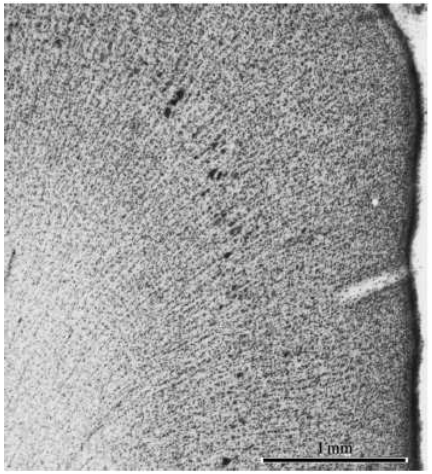
The occular dominance is preserved in layer IV of V1: ***occular dominance patterns***.

V1: Occular Dominance

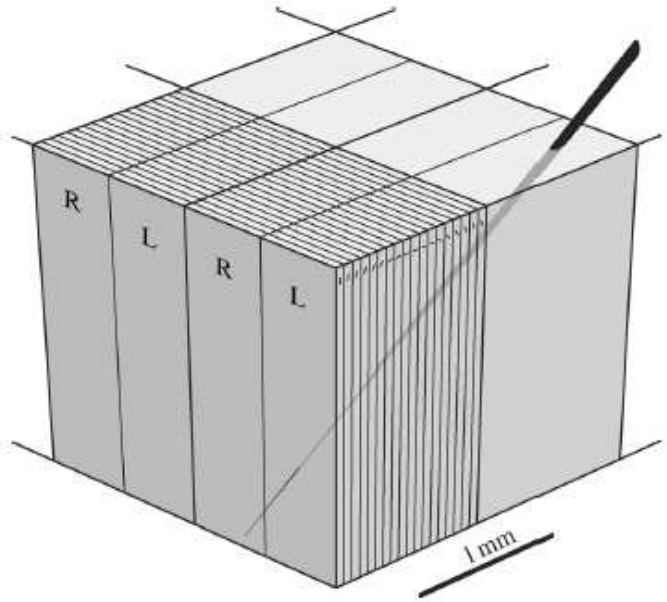


The occular dominance is organized in stripes (*striate cortex*).

Recall that neurons in cortex seem to be orgainzed in *minicolumns*.



V1: Orientation Map



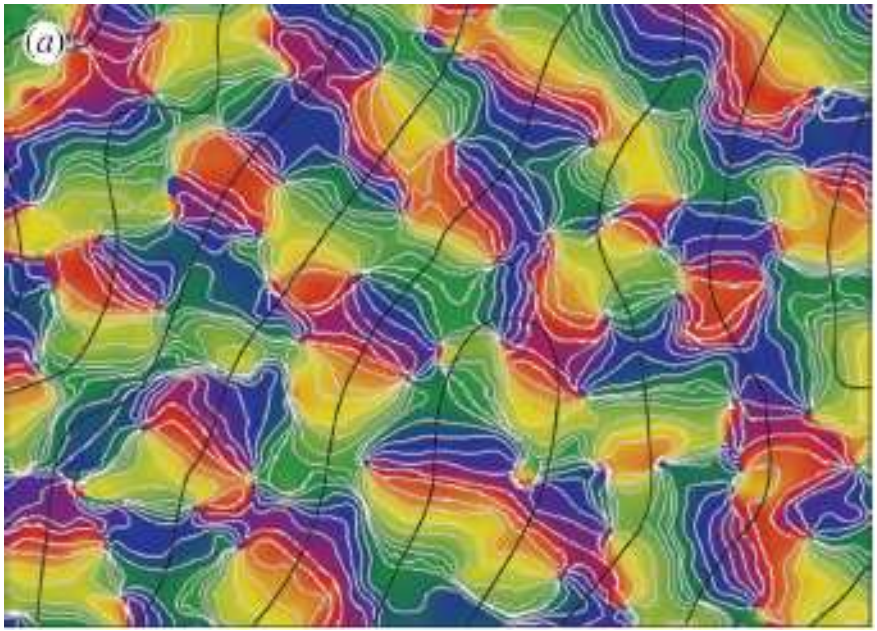
Neurons in V1 are sensitive to the orientation of bars and edges.

Hubel and Wiesel discovered that all neurons in a minicolumn are sensitive to the same orientation. Furthermore, the orientation changes smoothly as one traverses V1 horizontally.

This led to the so-called *ice-cube model* of V1: Minicolumns are bundled together such that one cube represents the whole set of visual features at a particular location.

Such a bundle of minicolumns is also called a ***hypercolumn***.

V1: Orientation Map



With optical imaging techniques, one can map out the orientation of neurons in V1.

One sees that orientations are changing smoothly, and that there are singularities, so-called ***pinwheels***.

The occular dominance boundaries tend to cross lines of iso orientation at right angles.

Note: Some animals do not have orientation maps, e.g. rats, some don't even have occular dominance columns.

Map Formation

Questions:

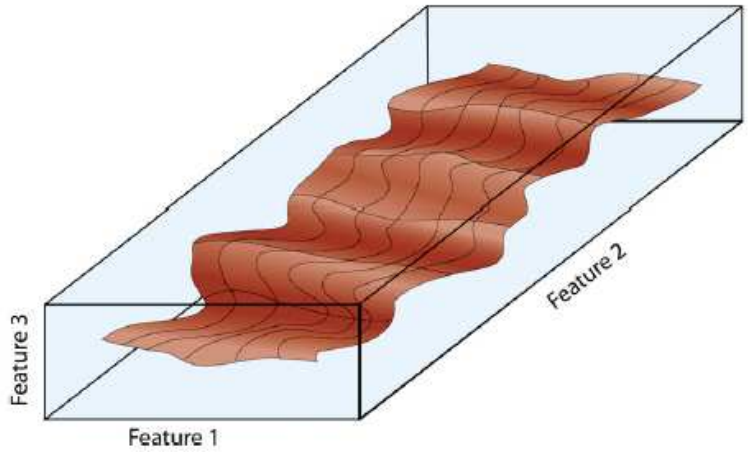
- ***How do maps emerge?*** Probably a mixture of the following:
  - genetically encoded.
  - activity-dependend synaptic modification.
- ***What is the functional role of cortical maps?*** Hypotheses:
  - Local computations make use of local and preferentially similar features. Grouping local features together saves wire length.
  - There is no real functional role of maps.

Elastic Net: Minimizing wire length

The elastic net model starts from the hypothesis that similar features have to be clustered together in order to minimize wire length of cortical processing.

It does not really model neuronal activity or plasticity. Instead it tries to solve a dimensionality reduction optimization prolem:

- Given a d dimensional feature space (e.g. four dimensional for location (x,y), occularity, and orientation).
- Find a mapping of the features onto the 2D cortical sheet such that all feature values are represented and similar feature values are nearby.



Elastic Net: Minimizing wire length

Consider a 40×40 array of cortical locations.

Each location j is sensitive to the parameters $\mathbf{w}_j = (x, y, r, \theta)^T$ in the following way:

The activity of location j is dependent on the distance of its prefered feature vector $\mathbf{w}_j$ from the features $\mathbf{v}$ of the actual input:

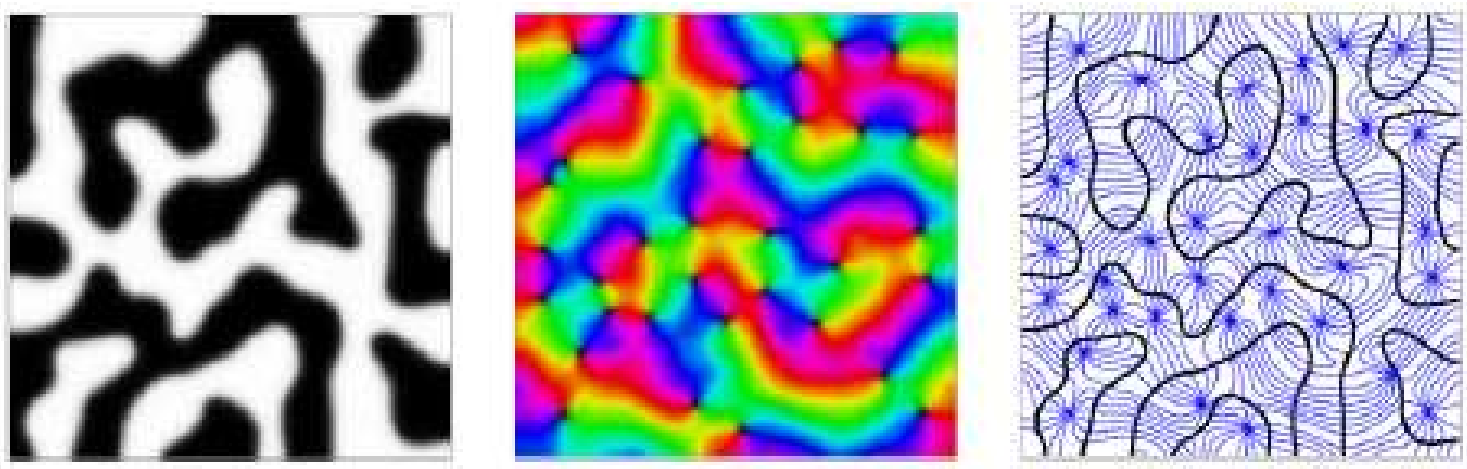
$$
\begin{aligned}
O_j &= \exp\left(-\frac{(\mathbf{v} - \mathbf{w}_j)^2}{2K^2}\right) \\
o_j &= \frac{O_j}{\sum_k o_k}
\end{aligned}
$$

Receptive fields of all units are updated according to

$$\Delta \boldsymbol{w}_j = \alpha o_j (\mathbf{v} - \mathbf{w}_j) + \beta \sum_{k \in N_j} (\mathbf{w}_k - \mathbf{w}_j).$$

Elastic Net: Minimizing wire length

As a result, the elastic net approach produces maps which have similar properties as cortical maps.



Note:

- The model can just give evidence for a principle *why* maps could be formed.
- It actually does not tell us how it is formed (e.g. which synaptic plasticity rule).

SOFM and Competitive Learning

Self-Organizing Feature Maps (SOFM) a.k.a. Kohonen Maps: One uses a self-organizing neural network to create a map in a low-dimensional space, using competitive learning.

Competitive Learning:

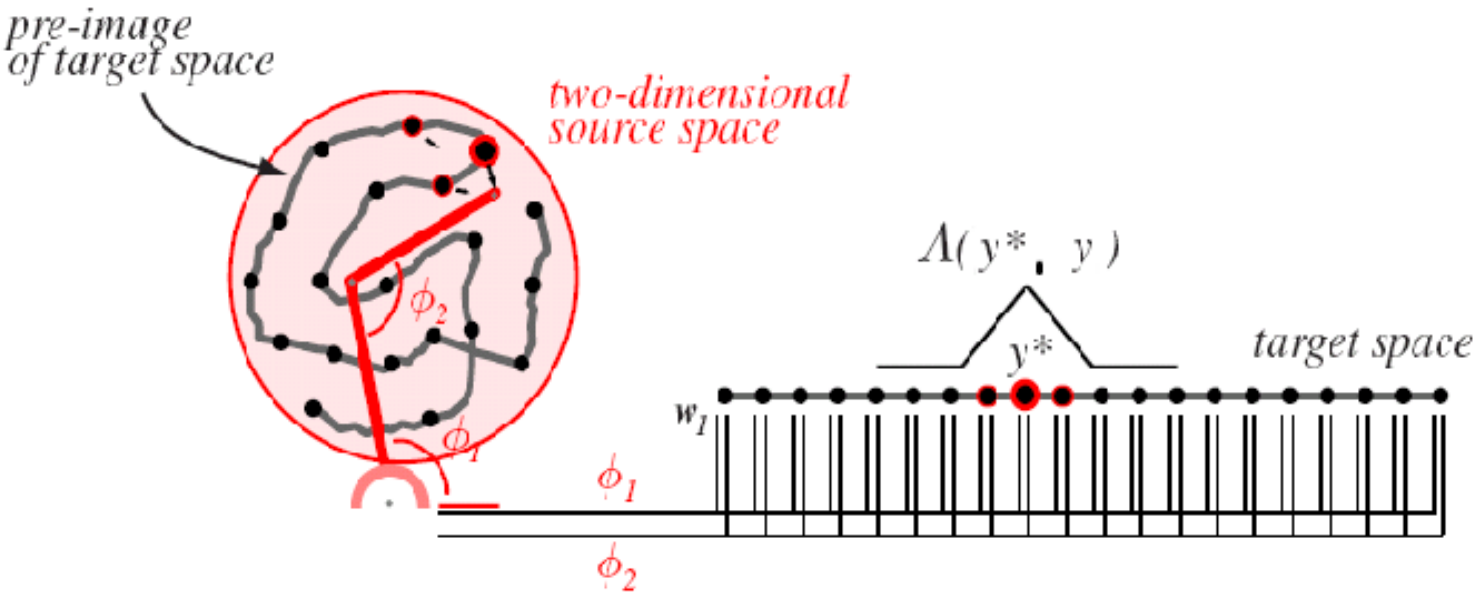
Consider a single layer neural network with inputs $\mathbf{x} \in \mathbb{R}^d$ fully connected to m output units.

- The unit which is closest to the input pattern $\mathbf{x}$ wins: $i(\mathbf{x}) = \arg\min_j ||\mathbf{x} - \mathbf{w}_j||$ (competition, Winner-Take-All networks).
- Learning is done with regard to the winner,e.g., only the winner adapts its weights: $\Delta \mathbf{w}_{i(\mathbf{x})} = \eta(\mathbf{x} - \mathbf{w}_{i(\mathbf{x})})$.
- The goal is to find *prototypes* for the input, i.e., for each input, only a single unit is activated. Hence, many inputs are mapped onto a prototype vector, defined by the responding unit. Therefore, the whole input space is sampled by m prototype vectors.

Self-Organizing Feature Maps

In a SOFM, the output units have some spatial topology. E.g., in a 1D map, units are lined up on a line, in 2D they may be located on a 2D grid. Let $\mathbf{y}_k$ be the location of output unit k.

Idea: Given inputs $\mathbf{x}(1), \ldots, \mathbf{x}(n)$, learn a mapping from the original space to the feature space given by the output units of the network.



Note: The space is sampled by m points (number of output units) $\longrightarrow$ vector quantization.

Self-Organizing Feature Maps

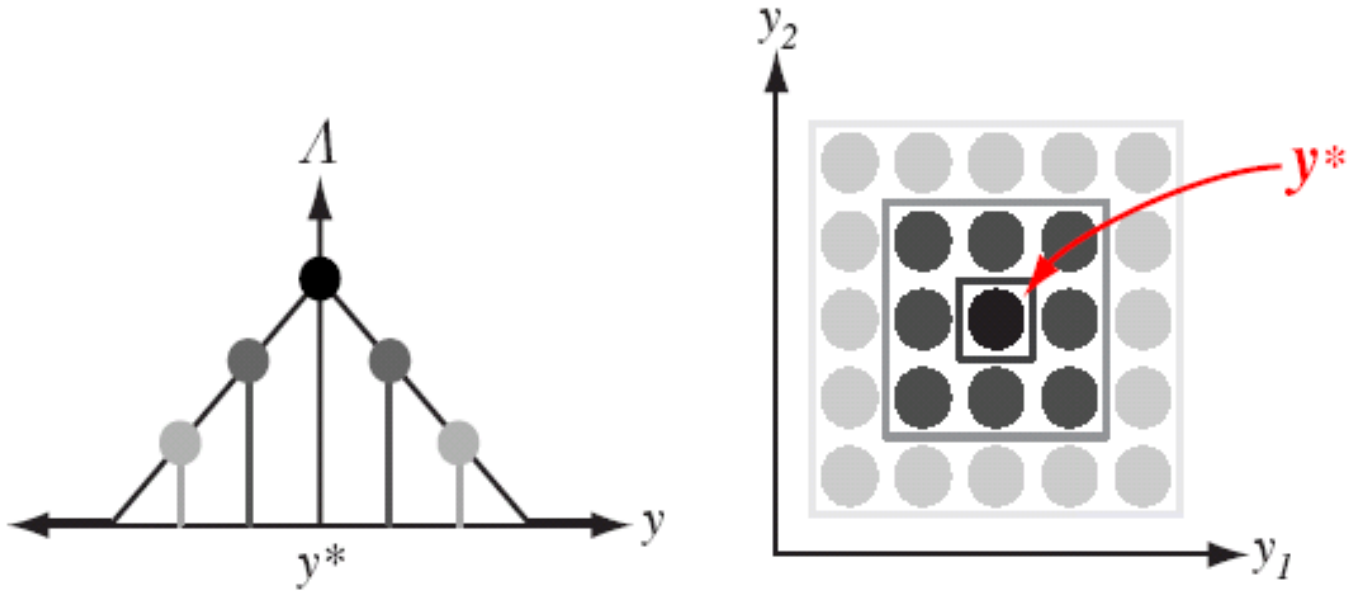
Let $\mathbf{y}^*$ be the (location of the) winning unit.

The weight update for input $\mathbf{x}$ is given by

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \eta(t) \cdot \Lambda(\mathbf{y}(k), \mathbf{y}^*) \cdot (\mathbf{x} - \mathbf{w}_k(t)),$$

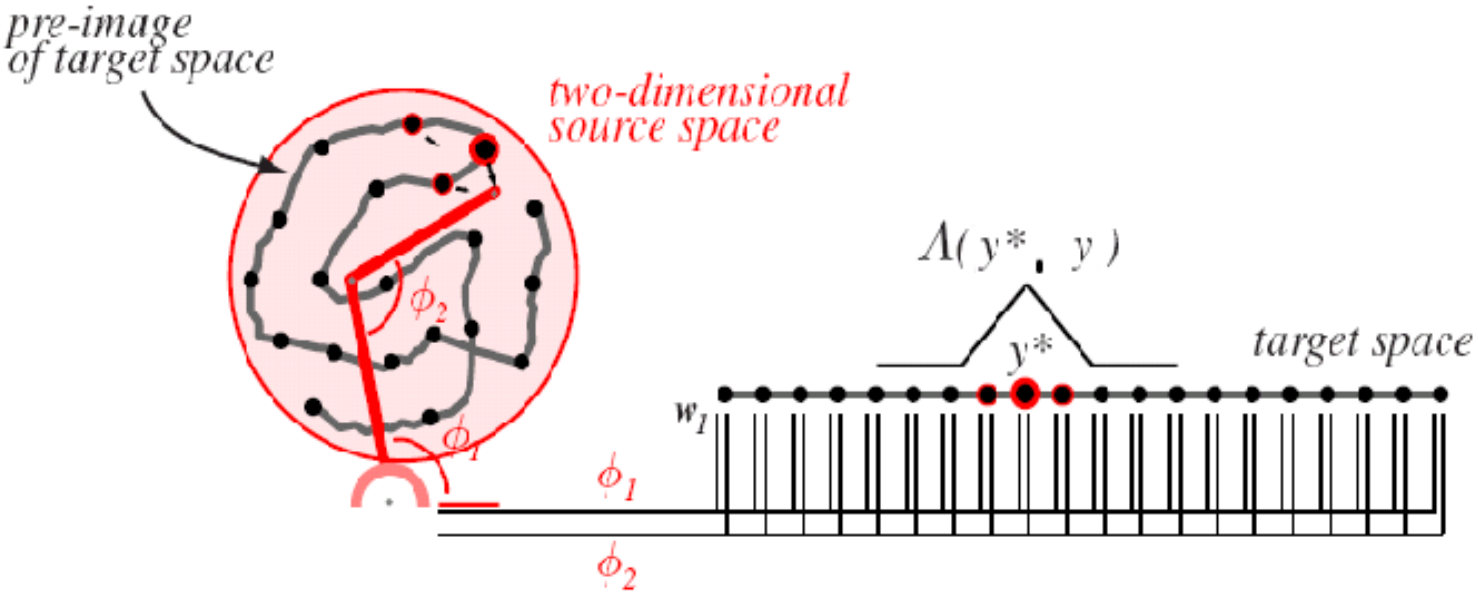
where $\Lambda(\mathbf{y}(k), \mathbf{y}^*)$ is called the *window function*. It has value 1 for $\mathbf{y}(k) = \mathbf{y}^*$, and small values for locations nearby.

Typical window functions:



Self-Organizing Feature Maps

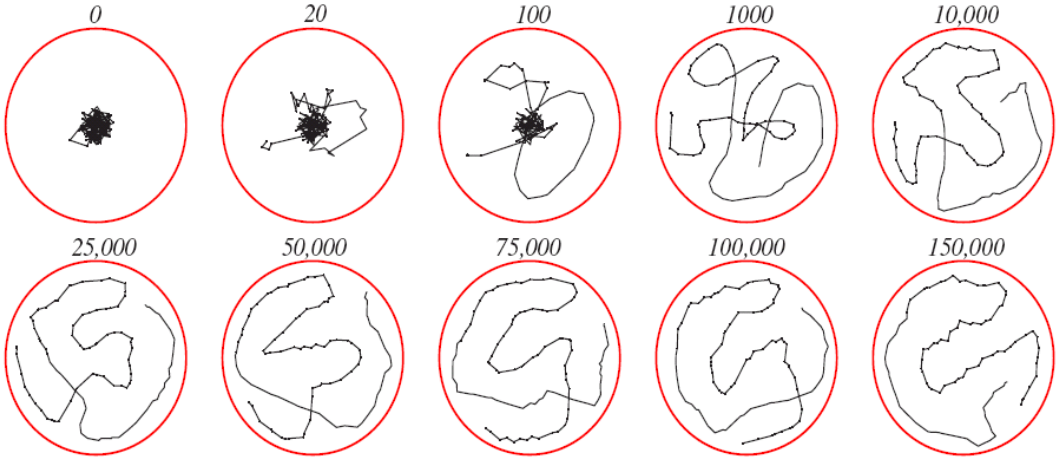
$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \eta(t) \cdot \Lambda(\mathbf{y}(k), \mathbf{y}^*) \cdot (\mathbf{x} - \mathbf{w}_k(t))$$



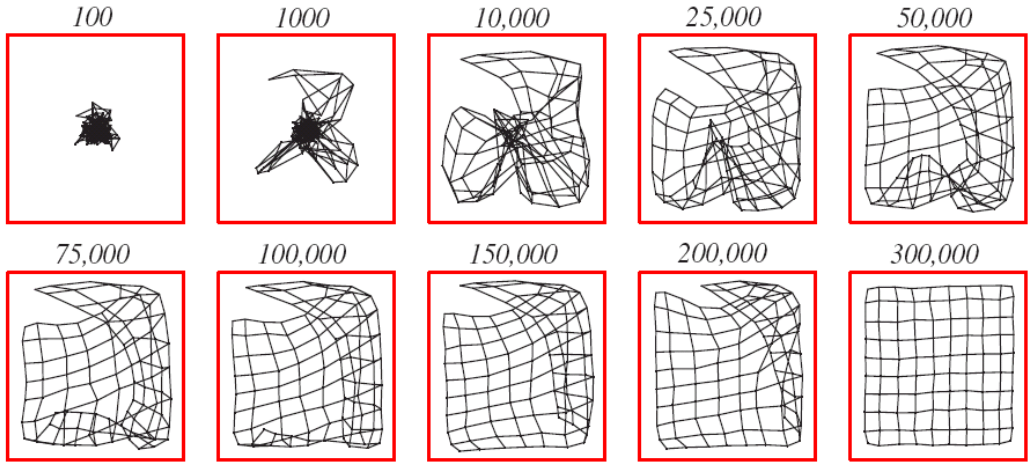
- The winning unit is pulled toward the input sample (it will therefore most strongly respond to the mean of the inputs where it wins).
- Units which are near the winning unit are also (but less strongly) pulled toward the input sample. Units nearby in feature space will tend to be activated for inputs nearby in source space.

Self-Organizing Feature Maps

Example: Unfolding of a 1D network for the example discussed above (150000 iterations).

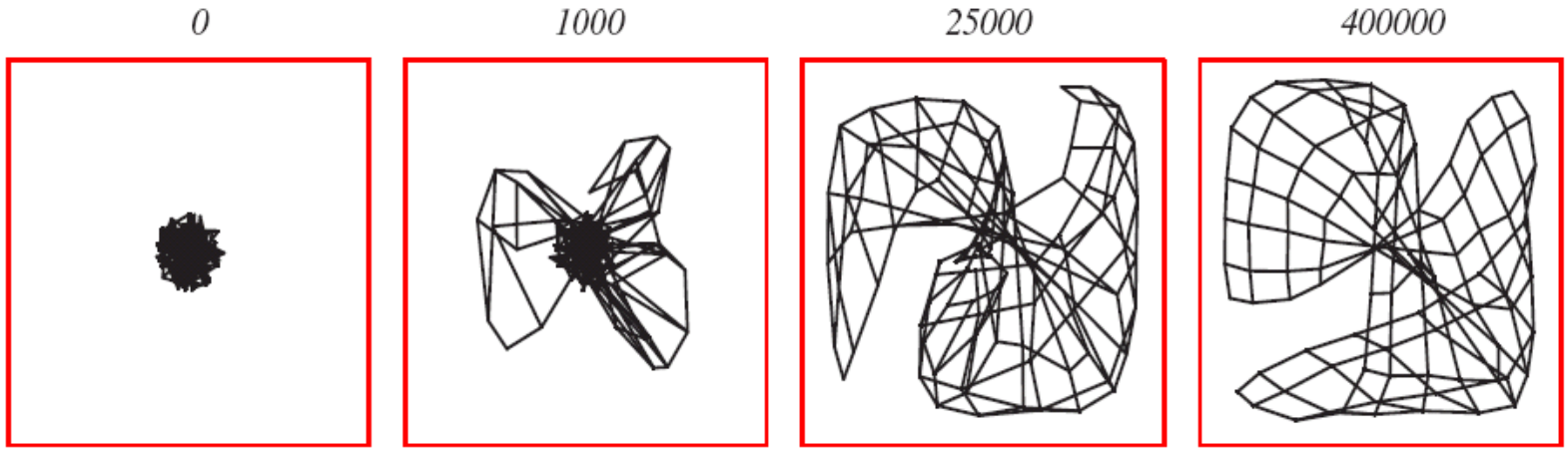


Example: Unfolding of a 2D network for a 2D source space.

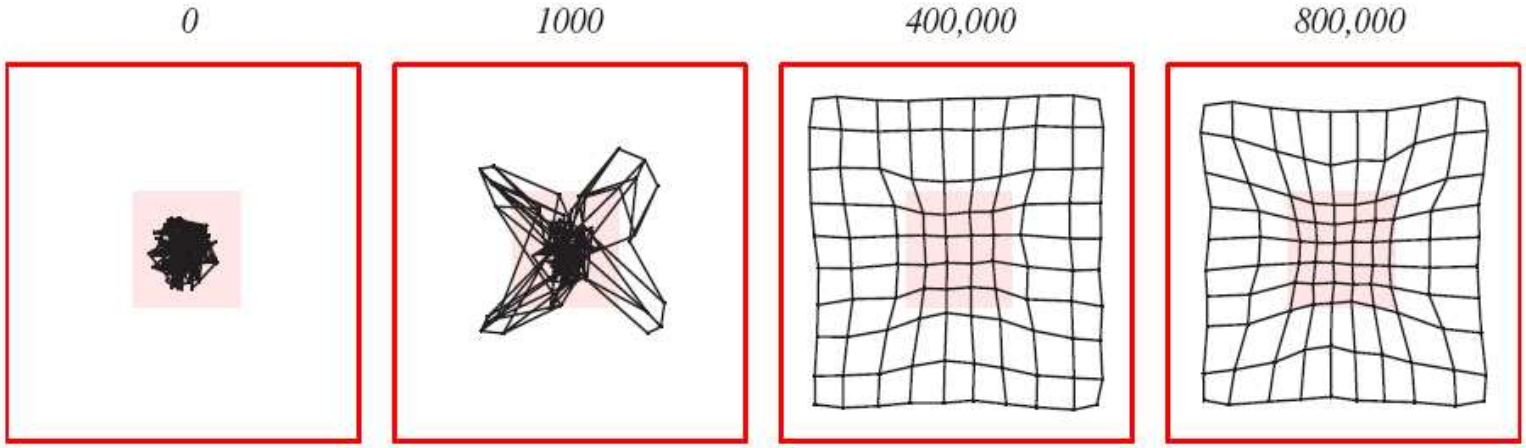


Self-Organizing Feature Maps

Problem: Upper left and lower right parts of the network have different orientation.



Interesting feature: The network accounts for sampling probabilies of points in source space



Application Areas

- Map formation in cortex
- Visualization of high-dimensional data
- Control of robot arms (mapping high-dimensional joint-space onto 3D)
- Preprocessing
- Image Compression
- Classification, in conjunction with supervised techniques, or as Learning Vector Quantization.

Robot Arm Application (Neural Gas)