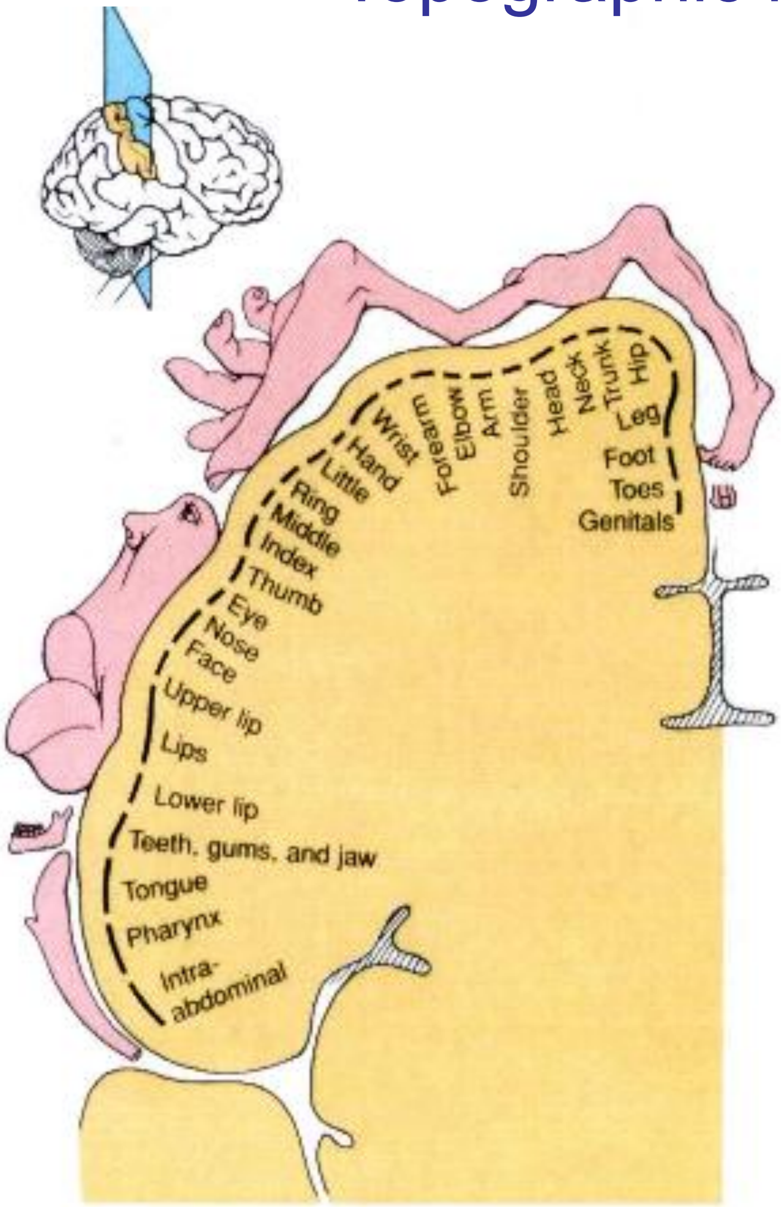
Neuroinformatics

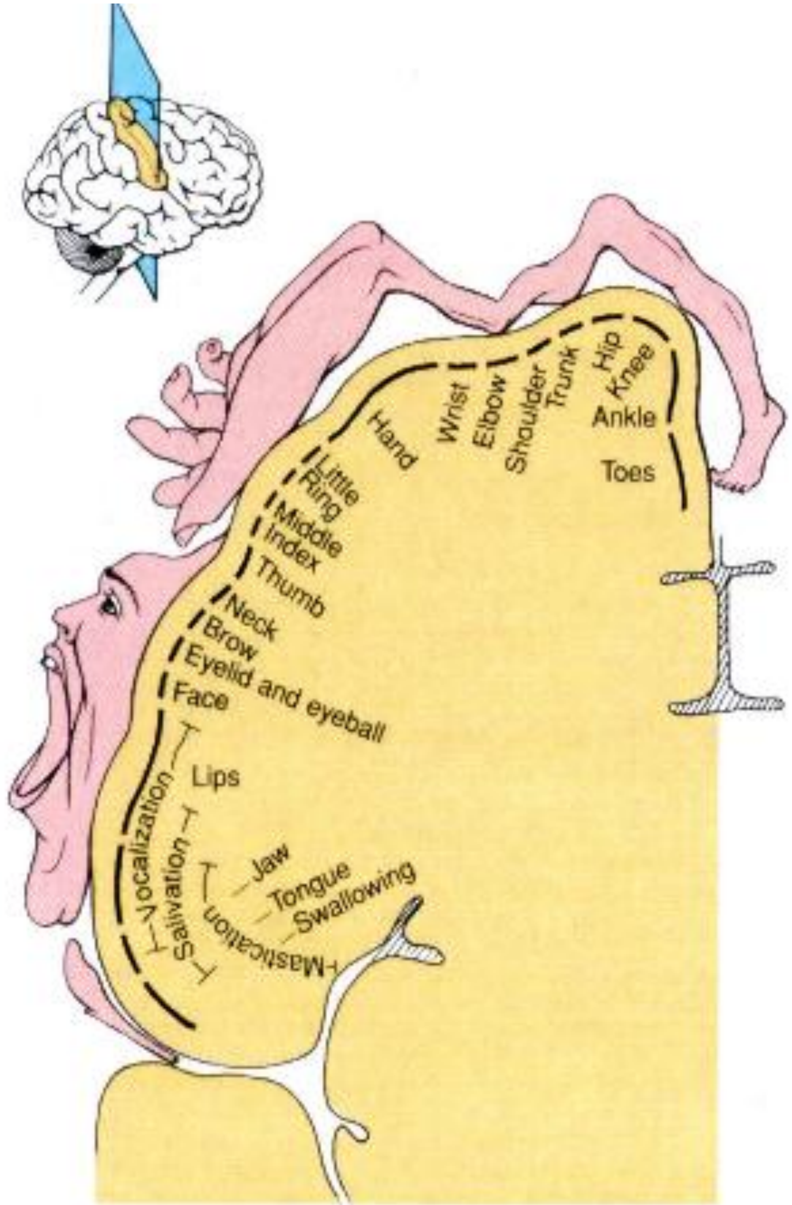
April 16, 2024

Dynamical neural fields (DNF)

Topographic map: Homunculus



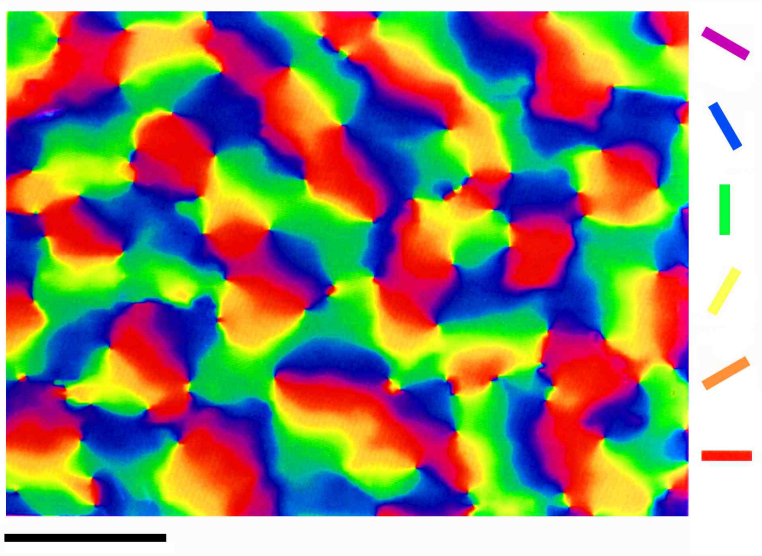
(a) Somatosensory cortex in right cerebral hemisphere



(b) Motor cortex in right cerebral hemisphere

Topographic map: other examples

Orientation map

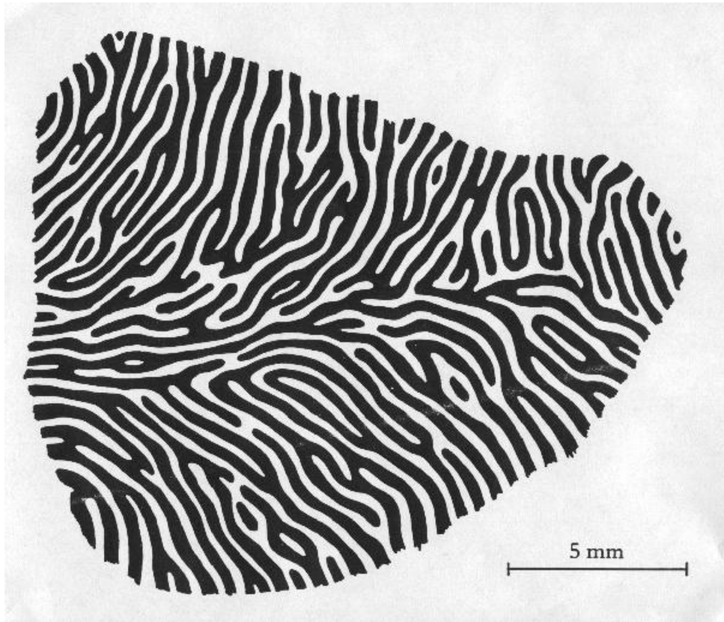


(http://www.scholarpedia.org/article/Visual_map)

•Hubel&Wiesel (1962, 1974): orientation selectivity and its locally continuity characteristic

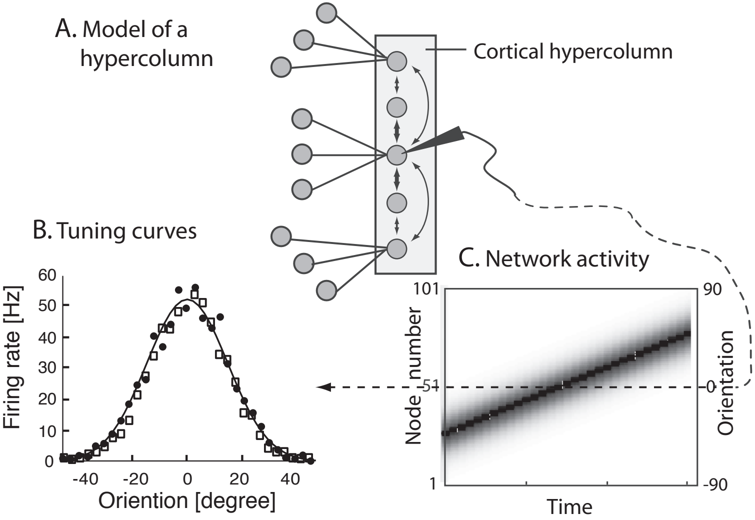
•Swindale (1982),Blasdel&Salama(1986), Swindale et al.(1987): 2D map

Ocular Dominance Columns



Reconstruction of the ocular dominance columns in area 17 of the right Hemisphere of a monkey (tangential section)

Motivation for SOM and DNF - Tuning Curves

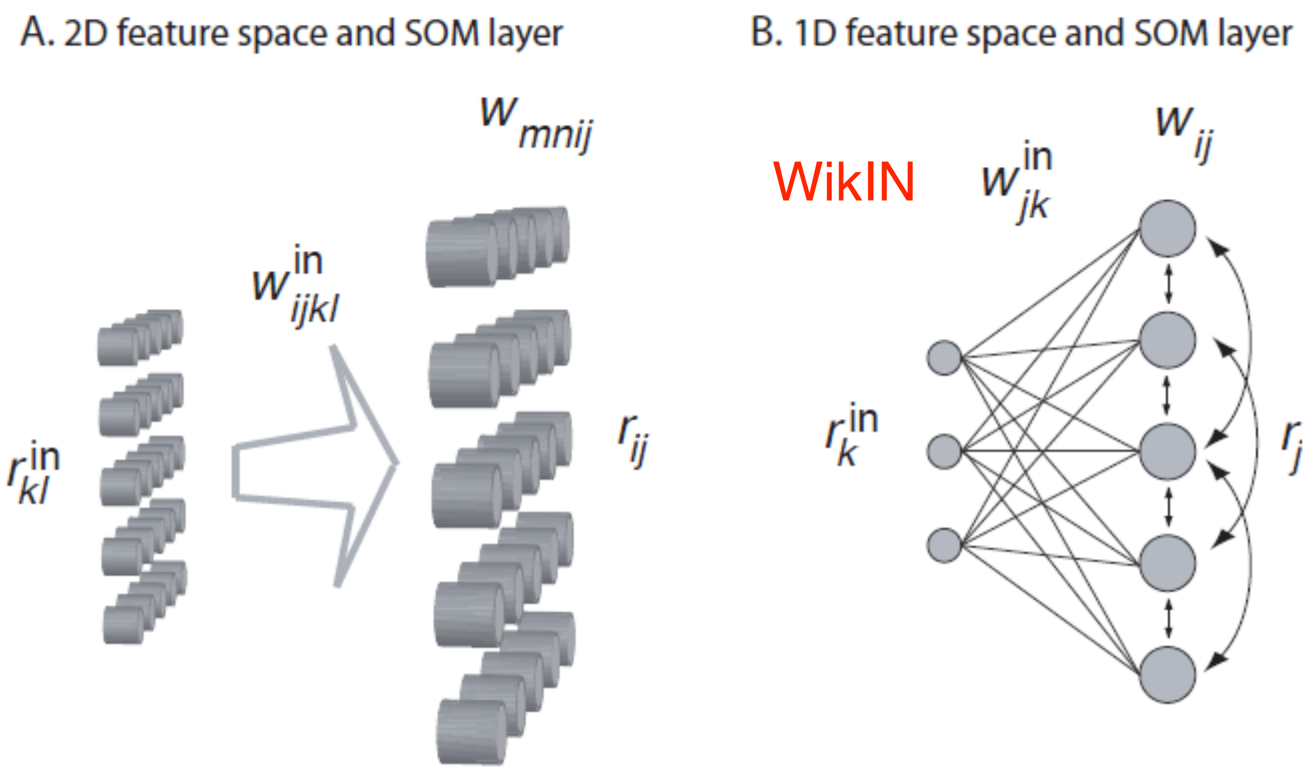


Self-organizing maps (SOMs)

David Willshaw
Edinburgh Univ., UK

Willshaw-von der Malsburg model

Christoph von der Malsburg
Bochum Univ. (now at FIAS, Frankfurt, Germany)



Network equations

Update rule of (recurrent) cortical network:

$$\tau \frac{\mathrm{d}u_i(t)}{\mathrm{d}t} = -u_i(t) + \frac{1}{N} \sum_j w_{ij} r_j(t) + \frac{1}{M} \sum_k w_{ik}^{\mathrm{in}} r_k^{\mathrm{in}}(t)$$

Activation function: $r_j(t) = \frac{1}{1+e^{\beta(u_j(t)-\alpha)}}$.

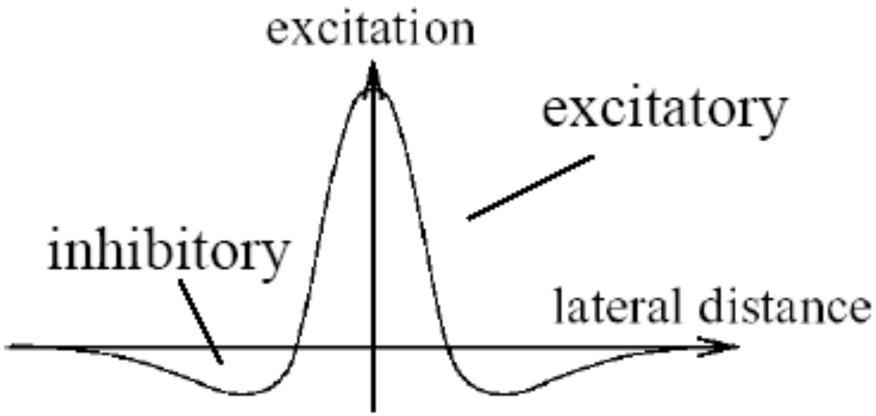
Lateral weight matrix: $w_{ij} \propto r_i r_j$

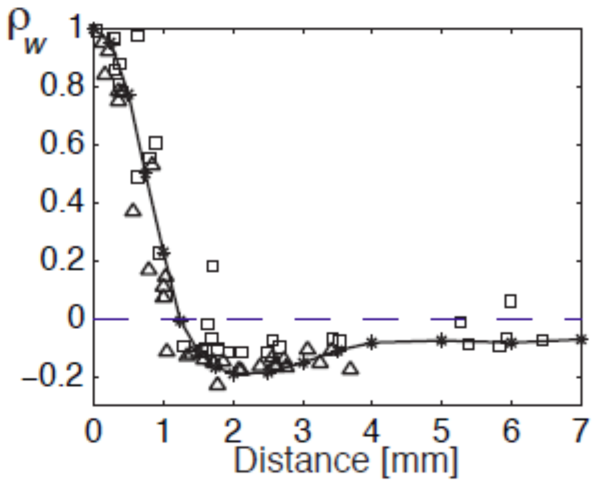
$$= A_{\mathrm{w}} \left(\mathrm{e}^{-((i-j)*\Delta x)^2/2\sigma^2} - C \right)$$

Input weight matrix: $w_{ij}^{\mathrm{in}} \propto r_i r_j^{\mathrm{in}}$

Stephen Grossberg
Boston Univ. USA

A principle of SOM: cooperation and competition





Cooperation: Short-range excitation
Competition: long-range inhibition
(note: local inhibition)

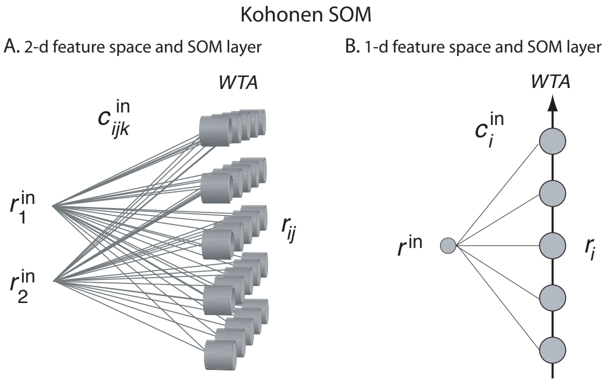
Interaction strength from cell recordings in superior colliculus (Trappenberg et al., 2001)

Self-organizing maps (SOMs)

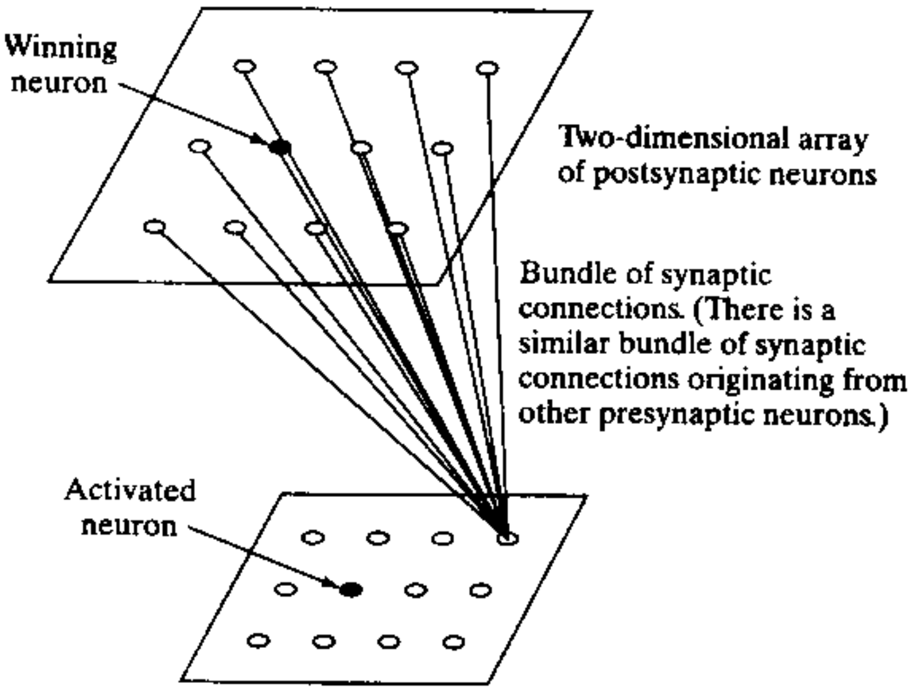
- The development of SOM as a neural model is motivated by the topographical nature of cortical maps.
- Visual, tactile, and acoustic inputs are mapped in a topographical manner.
- Visual: retinotopy (position in visual field), orientation, spatial frequency, direction, ocular dominance, etc.
- Tactile: somatotopy (position on skin,thumb and SMS)
- Acoustic: tonotopy (frequency)
- Self-organizing maps (SOM) is based on competitive learning, where output neurons compete with each other to be activated (Kohonen, 1982)
- The output neuron that activates is called the winner-takes-all neuron
- Lateral inhibition is one way to implement competition for map formation (von der Malsburg 1973)
- In SOM, neurons are placed on a lattice, on which a meaningful coordinate system for different features is created (feature map).
- The lattice thus forms a topographic map where the spatial location on the lattice is indicative of the input features.

Kohonen - Shortcut

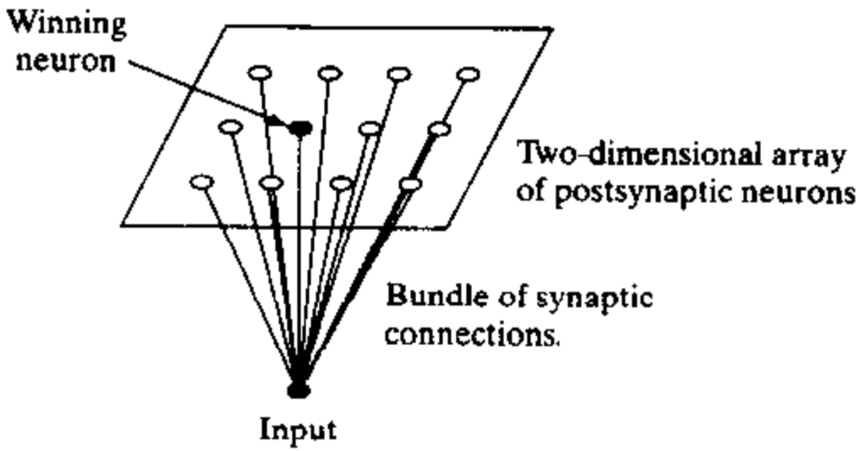
- Willshaw-von der Malsburg model: input neurons arranged in 2D lattice, output in 2D lattice. Lateral excitation/inhibition (Mexican hat) gives rise to soft competition. Normalized Hebbian learning. Biological motivation.
- Kohonen model: input of any dimension, output neurons in 1D, 2D, or 3D lattice. Relaxed winner-takes-all (neighborhood). Competetive learning rule. Computational motivation.



Two approaches for SOMs



(a) Willshaw-von der Malsburg's model



(b) Kohonen model

Developed for a retinotopic map
Input space is already topographic (retina)
Lateral connectivity captures C&C
The winning neuron occurs through neural dynamics
Can be both global and local competition

Input space is a continuous value
No lateral connectivity or neural dynamics
First find winning neuron (competition)
Then, learning of this neuron affects the neighbors (cooperation)
Global competition (no other possibility)

Kohonen model

- cortical sheet activation, σ_r^2 width of activated area, activation fce resembels tuning curves, radial-basis networks

$$r_{ij} = \exp(-\sum_k (c_{ijk} - r_k^{in})^2 / 2\sigma_r^2)$$

- strength connection around the winning node r_{ij}^*, WTA rule - winner takes all

$$\Delta c_{ijk} = \epsilon r_{ij}^* (r_{in} - c_{ijk})$$

- ML approach (Matlab implementation):
  $w^i(q) = w^i(q-1) + \alpha(p(q) - w^i(q))$, i are lying in neighborhood $N(i)_d = \{j, d_{ij} < d\}$

SOM Algorithm

1. Randomly initialize weight vectors w_i
2. Randomly sample input vector x
3. Find Best Matching Unit (BMU)

$$i(x) = \arg\min_j ||x - w_j||$$

4. Update weight vectors, where $h(j, i(x))$ is neighborhood function of BMU

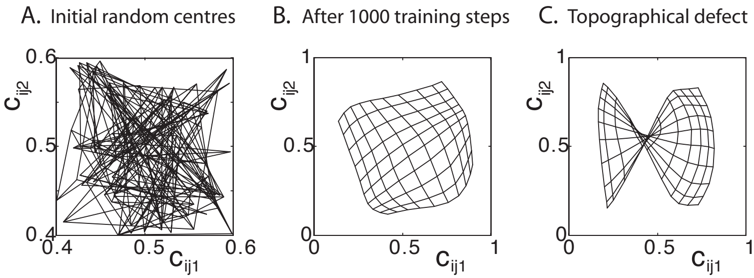
$$w_j = w_j + \epsilon h(j, i(x))(x - w_j)$$

5. Repeat steps 2-4

som.m

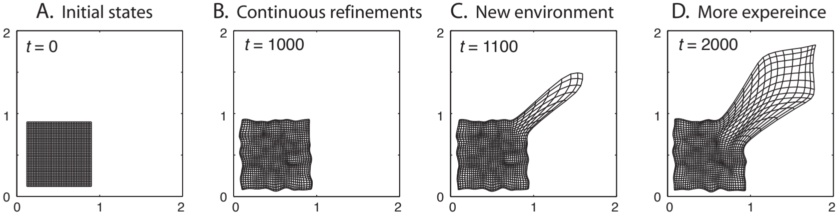
```
%% Two dimensional self-organizing feature map al la Kohonen
clear; nn=10; lambda=0.2; sig=2; sig2=1/(2*sig^2);
[X,Y]=meshgrid(1:nn,1:nn); ntrial=0;

% Initial centres of prefered features:
c1=0.5-.1*(2*rand(nn)-1);
c2=0.5-.1*(2*rand(nn)-1);

%% training session
while(true)
   if(mod(ntrial,100)==0) % Plot grid of feature centres
       clf; hold on; axis square; axis([0 1 0 1]);
       plot(c1,c2,’k’); plot(c1’,c2’,’k’);
       tstring=[int2str(ntrial) ’ examples’]; title(tstring);
       waitforbuttonpress;
   end
   r_in=[rand;rand];
   r=exp(-(c1-r_in(1)).^2-(c2-r_in(2)).^2);
   [rmax,x_winner]=max(max(r)); [rmax,y_winner]=max(max(r’));
   r=exp(-((X-x_winner).^2+(Y-y_winner).^2)*sig2);
   c1=c1+lambda*r.*(r_in(1)-c1);
   c2=c2+lambda*r.*(r_in(2)-c2);
   ntrial=ntrial+1;
end
```

SOM simulation



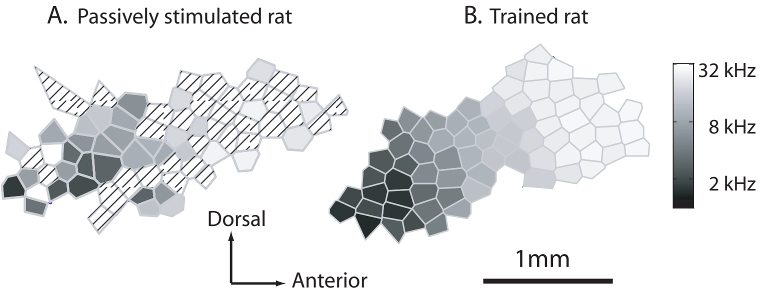
Another example

- Simulating development processes
- SOM can represent new domains, representation less fine-grained compared to initial domain
- Early in life exposed to broad feature space (learning languages)

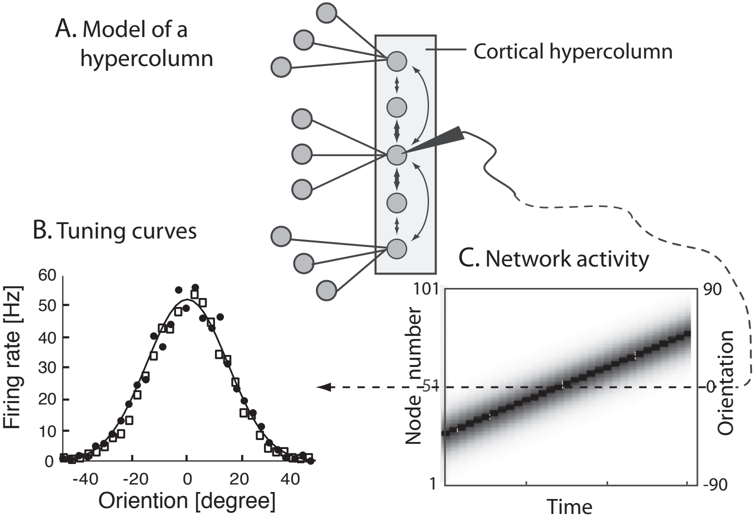


Representational plasticity - Zhou and Merzenich, PNAS 2007

- rat pups raised in noisy environment $\leftarrow$ severely impaired tonotopicity (tones representations) in primary auditory cortex - A1
- no recovery after stimulation with sounds of different frequencies
- stimulation by discrimination task with food reward $\leftarrow$ rats were able to recover tonotopic maps
- traditionally SOM maps are driven by data: bottom - up approach
- top-down processing explains those experimental results (reinforcement learning)



Tuning Curves



Dynamic Neural Field Theory

Field dynamics:

$$\tau \frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} = -\mathbf{u}(\mathbf{x}, t) + \int_{\mathbf{y}} \mathbf{w}(\mathbf{x}, \mathbf{y}) \mathbf{r}(\mathbf{y}, t) \mathrm{d}\mathbf{y} + I^{\mathrm{ext}}(\mathbf{x}, t)$$

$$\mathbf{r}(\mathbf{x}, t) = g(\mathbf{u}(\mathbf{x}, t)),$$

Continuous version of equations above with discretization:

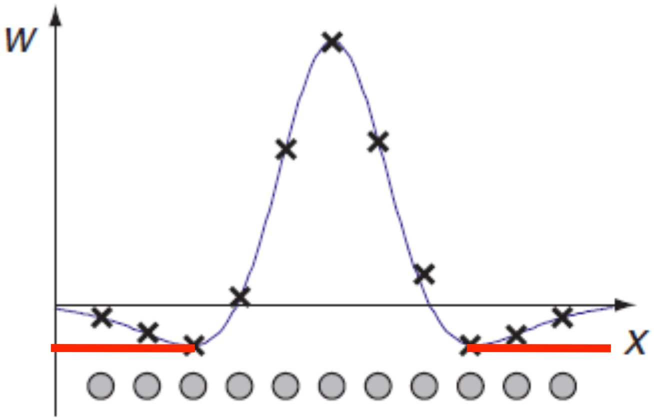
$$x \to i\Delta x \text{ and } \int \mathrm{d}x \to \Delta x \sum$$

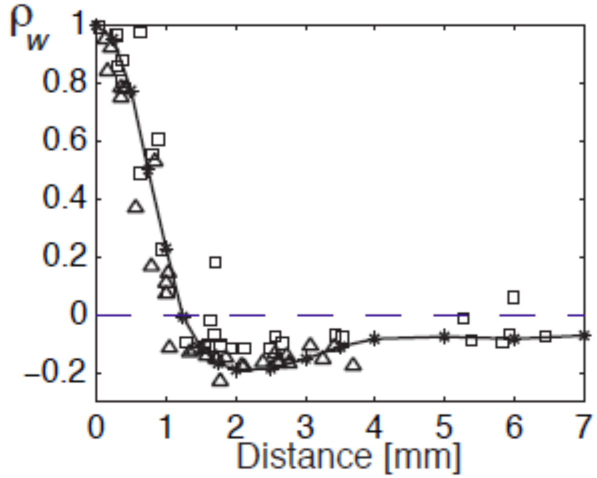
Main assumption: Short-distance excitation and long-distance inhabitation

The center-surround interaction (weight) kernel

$$\mathbf{w}^{\mathrm{E}}(|x-y|) = A_{\mathrm{w}} e^{-(x-y)^2/4\sigma_r{}^2} - A_{\mathrm{w}}\, C$$

Can be learned from Gaussian response curves of individual nodes





Black solid line: a Mexican hat activation pattern (in 3D, local competition)

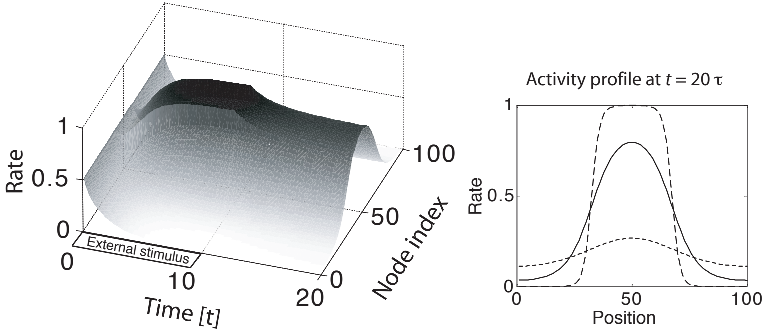
can be obtained with subtraction of two Gaussians.

matched with physiological data (right, Trappenberg et al., 2001)

Red Solid line: Gaussian with negative bias (global competition)

Self-sustained activity packet

- growing activity: $C << E$, whole map is active, undesirable
- decaying activity: $C >> E$, decaying after removal of external input
- memory activity: stability even when external input is removed !
- simulation: string external stimulus: nodes 40-50, excitatory weights to nearby nodes, active nodes: activity packets, buble or bump $\leftarrow$ continuos attractor neural networks $\leftarrow$ working memory, $A_w = 4, C = 0.5$



dnf.m

```
%% Dynamic Neural Field Model (1D)
 clear; clf; hold on;
 nn = 100; dx=2*pi/nn; sig = 2*pi/10; C=0.5;

%% Training weight matrix
 for loc=1:nn;
     i=(1:nn)'; dis= min(abs(i-loc),nn-abs(i-loc));
     pat(:,loc)=exp(-(dis*dx).^2/(2*sig^2));
 end
 w=pat*pat'; w=w/w(1,1); w=4*(w-C);
%% Update with localised input
 tall = []; rall = [];
 I_ext=zeros(nn,1); I_ext(nn/2-floor(nn/10):nn/2+floor(nn/10))=1;
 [t,u]=ode45('rnn_ode',[0 10],zeros(1,nn),[],nn,dx,w,I_ext);
 r=1./(1+exp(-u)); tall=[tall;t]; rall=[rall;r];
%% Update without input
 I_ext=zeros(nn,1);
 [t,u]=ode45('rnn_ode',[10 20],u(size(u,1),:),[],nn,dx,w,I_ext);
 r=1./(1+exp(-u)); tall=[tall;t]; rall=[rall;r];
%% Plotting results
 surf(tall',1:nn,rall','linestyle','none'); view(0,90);
```

rnn_ode.m

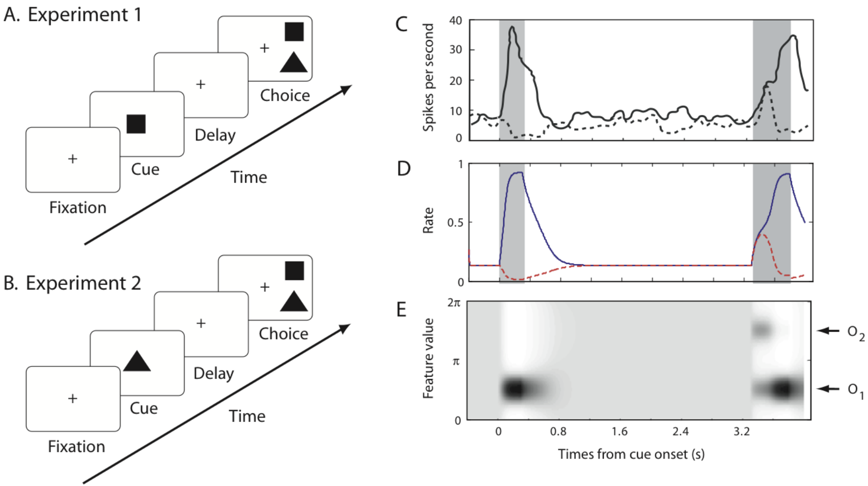
```
function udot=rnn_ode(t,u,flag,nn,dx,w,I_ext)
% odefile for recurrent network
 tau_inv = 1.;        % inverse of membrane time constant
 r=1./(1+exp(-u));
 sum=w*r*dx;
 udot=tau_inv*(-u+sum+I_ext);
return
```

Update rule of (recurrent) cortical network:

$$\tau \frac{\mathrm{d}u_i(t)}{\mathrm{d}t} = -u_i(t) + \frac{1}{N}\sum_j w_{ij} r_j(t) + \frac{1}{M}\sum_k w_{ik}^{\mathrm{in}} r_k^{\mathrm{in}}(t)$$

Activation function: $r_j(t) = \frac{1}{1+e^{\beta(u_j(t)-\alpha)}}$.

DNF example - Chelazzi, Nature, 1993

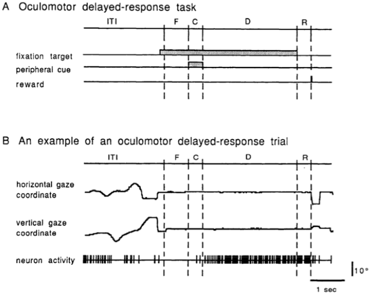


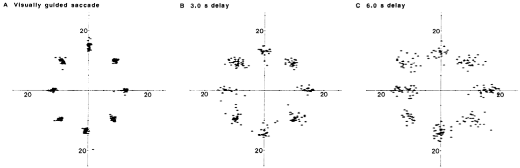
DNF example - Chelazzi, Nature, 1993, Matlab code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % 1-d Continuous Attractor Neural Network with Hebbian learning
 % two gaussian signal: decision network
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 clear;  close all;
 nn = 100; dx=2*pi/nn; % number of nodes and resolution in deg
 %weight matrices
    sig = 2*pi/20;
    w_sym=hebb(nn,sig,dx);
    w_inh=0.07;%use 0.04, 7,6,3; 3*(sqrt(2*pi)*sig)?2/nn;
    w=500*(w_sym-w_inh);
%inputs
    perc=0.01; Is=11;
    Ia=(1+0.5*perc)*Is;
    Ib=(1-0.5*perc)*Is;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%   Experiment    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    param=0;
%%%% no external input
     u0 = zeros(nn,1)-10;
     I_ext=zeros(nn,1);
     tspan=[0,40];
     [t,u]=ode45('rnn_ode_u',tspan,u0,[],nn,dx,w,I_ext);
     r=f1(u);
%%%% external cue
    u0 = u(size(t,1),:);
    I_ext=zeros(nn,1);
    loc1=pi/2;%+pi/16;
    loc2=3*pi/2;%-pi/16;
    I_ext=I_ext+in_signal_pbc(loc1,Is,sqrt(2)*sig,nn,dx);
    tspan=[40 70];
    [t2,u]=ode45('rnn_ode_u',tspan,u0,[],nn,dx,w,I_ext);
    r=[r;f1(u)];
t=[t;t2];
%%%% no external input
    u0 = u(size(t2,1),:);
    I_ext=zeros(nn,1);
    param=0;
    tspan=[70,370];
    [t2,u]=ode45('rnn_ode_u',tspan,u0,[],nn,dx,w,I_ext);
    r=[r;f1(u)];
t=[t;t2];
```

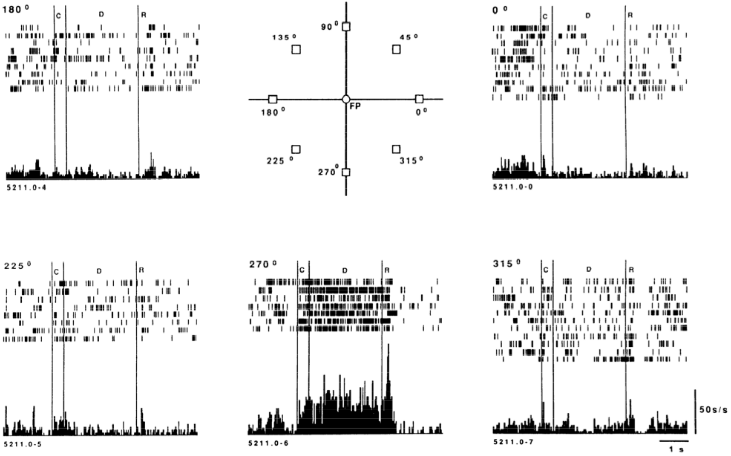
Working memory by ongoing firing - sustained DNF buble

- F- fixation period (0.75s), C-cue period (0.5s), D - delay period (3-6 s), R - response period (0.5s) $\rightarrow$ reward



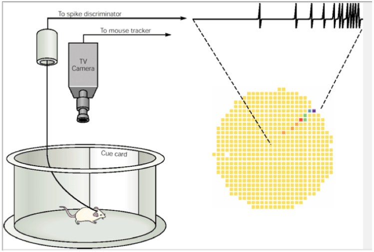


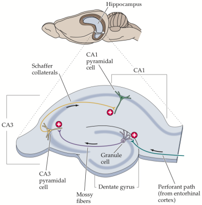
Directional delay period activity



Place cells

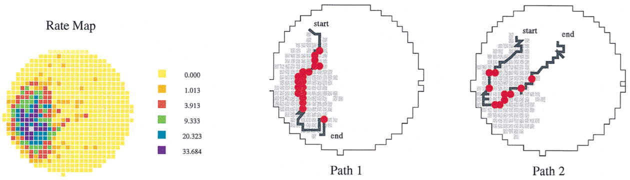
- Place cells are neurons in the hippocampus that exhibit a high rate of firing whenever an animal is in a specific location (pyramidal cells in CA1,CA4)
- On initial exposure to a new environment, place fields become established within minutes. The place fields of cells tend to be stable over repeated exposures to the same environment.
- Remapping - In a different environment, however, a cell may have a completely different place field or no place field at all





Place cells - 16 mins experiment

- colored circular region is an overhead view of a 76 cm diameter cylinder, each small square region (pixel) is about 2.5 cm squared, firing rate $\rightarrow$ total number of spikes fired in the pixel divided by the total time spent in the pixel.
- hungry rat ran around for 16 min chasing small food pellets, the black line indicates the rat's path and the red dots the locations at which action potentials were fired, action potentials were fired all along the second path even though the rat turned and ran out of the field in the direction opposite to its entry; this is an indication that the firing is not directionally selective.
- `http://www.youtube.com/watch?v=PGHRDcPKio8`



Further Readings

Teuvo Kohonen (1989), **Self-organization and associative memory**, Springer Verlag, 3rd edition.

David J. Willshaw and Christoph von der Malsburg (1976), **How patterned neural connexions can be set up by self-organisation**, in **Proc Roy Soc** B 194, 431–445.

Shun-ichi Amari (1977), **Dynamic pattern formation in lateral-inhibition type neural fields**, in **Biological Cybernetics** 27: 77–87.

Huge R. Wilson and Jack D. Cowan (1973), **A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue**, in **Kybernetik** 13:55-80.

Kechen Zhang (1996), **Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory**, in **Journal of Neuroscience** 16: 2112–2126.

Simon M. Stringer, Thomas P. Trappenberg, Edmund T. Rolls, and Ivan E.T. de Araujo (2002), **Self-organizing continuous attractor networks and path integration I: One-dimensional models of head direction cells**, in **Network: Computation in Neural Systems** 13:217–242.

Alexandre Pouget, Richard S. Zemel, and Peter Dayan (2000), **Information processing with population codes**, in **Nature Review Neuroscience** 1:125–132.

Miikkulainen R., **Computational Maps in the Visual Cortex**, Springer, 2005