

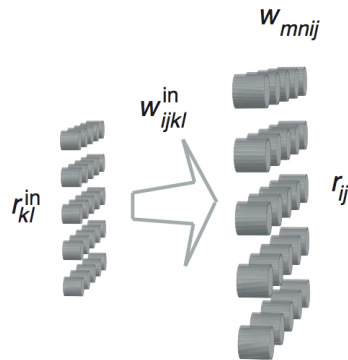
2

Self-organizing maps (SOMs)

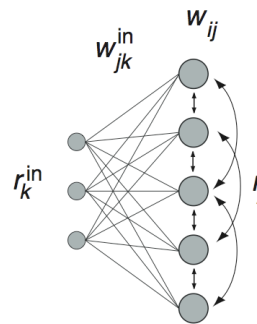


Willshaw-von der Malsburg model

A. 2D feature space and SOM layer



B. 1D feature space and SOM layer



3

Network equations

**Update rule of (recurrent) cortical network:**

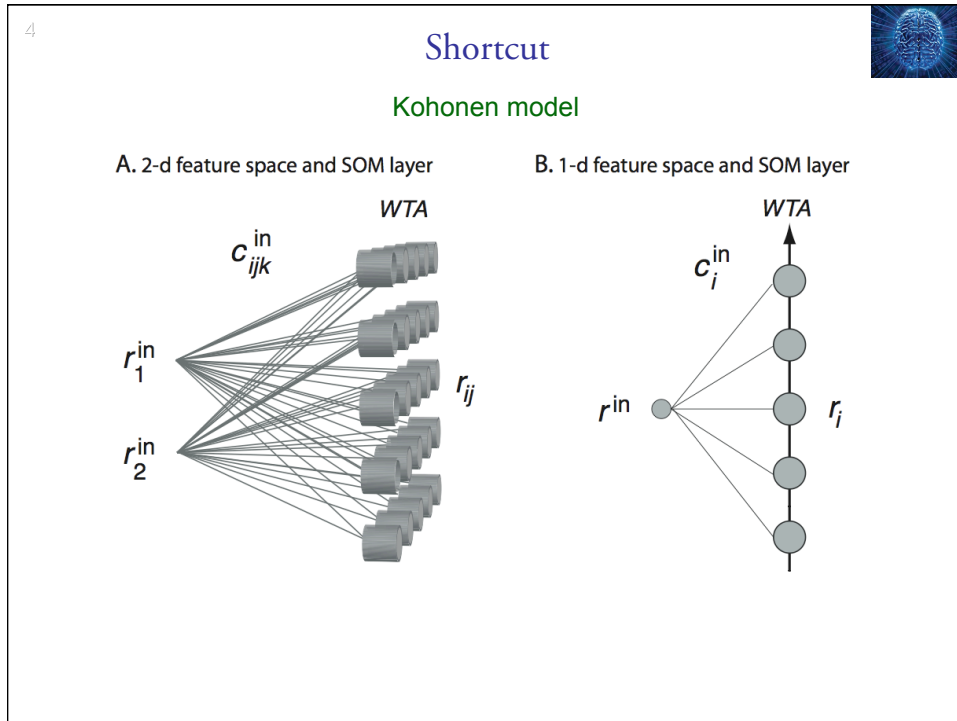
$$\tau \frac{du_i(t)}{dt} = -u_i(t) + \frac{1}{N} \sum_j w_{ij} r_j(t) + \frac{1}{M} \sum_k w_{ik}^{in} r_k^{in}(t)$$

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

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

$$= A_w \left(e^{-((i-j)*\Delta x)^2 = 2l^2} - C \right)$$

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



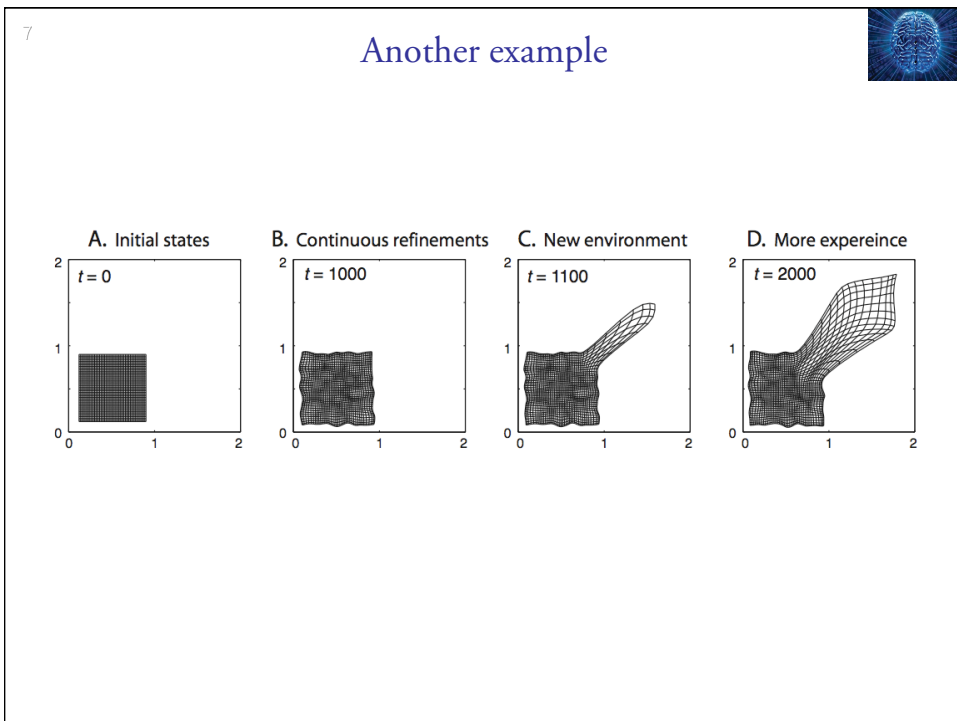
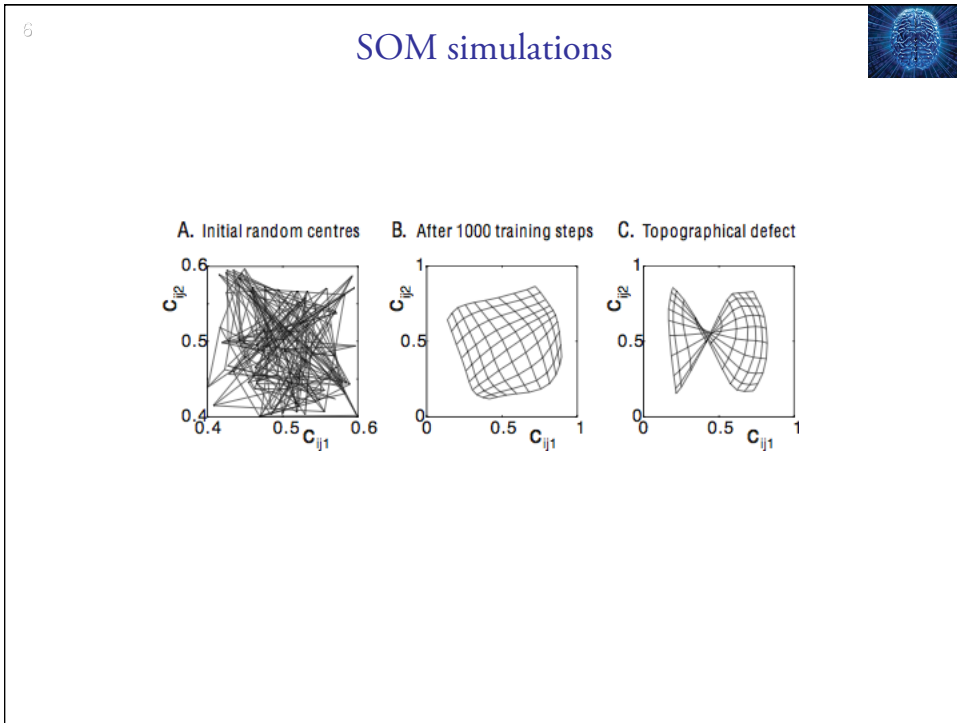
5

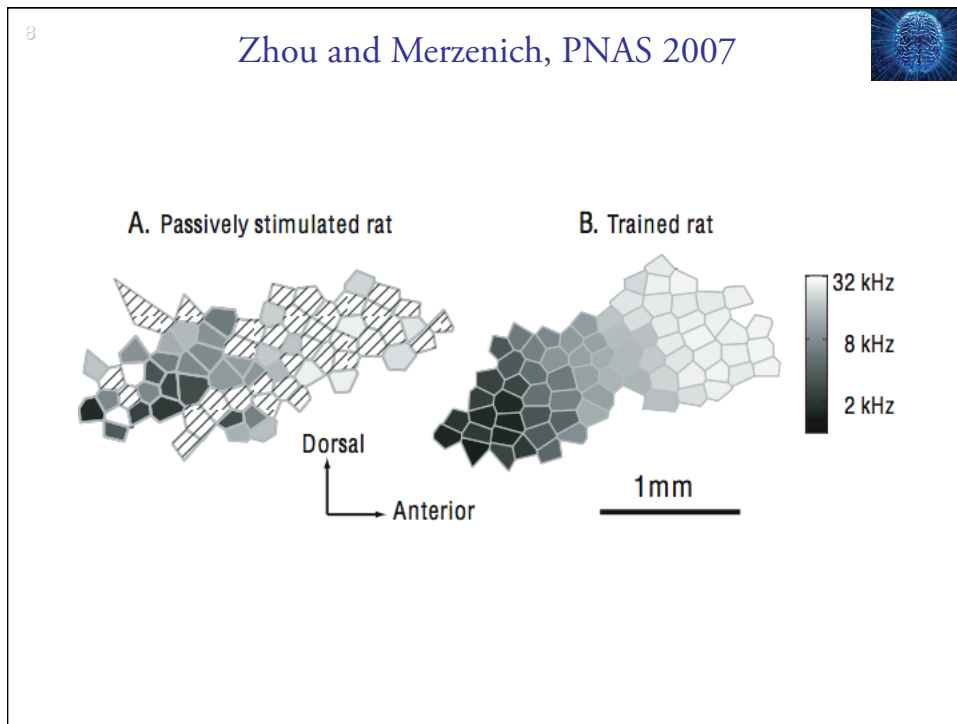
som.m

```

1  %% Two dimensional self-organizing feature map ala Kohonen
2  clear; nn=10; lambda=0.2; sig=2; sig2=1/(2*sig^2);
3  [X,Y]=meshgrid(1:nn,1:nn); ntrial=0;
4
5  % Initial centres of preferred features:
6  c1=0.5-.1*(2*rand(nn)-1);
7  c2=0.5-.1*(2*rand(nn)-1);
8
9  %% training session
10 while(true)
11     if(mod(ntrial,100)==0) % Plot grid of feature centres
12         clf; hold on; axis square; axis([0 1 0 1]);
13         plot(c1,c2,'k'); plot(c1',c2', 'k');
14         tstring=[int2str(ntrial) ' examples']; title(tstring);
15         waitforbuttonpress;
16     end
17     r_in=[rand;rand];
18     r=exp(-(c1-r_in(1)).^2-(c2-r_in(2)).^2);
19     [rmax,x_winner]=max(max(r)); [rmax,y_winner]=max(max(r'));
20     r=exp(-(X-x_winner).^2+(Y-y_winner).^2)*sig2);
21     c1=c1+lambda*r.*(r_in(1)-c1);
22     c2=c2+lambda*r.*(r_in(2)-c2);
23     ntrial=ntrial+1;
24 end

```





9

Dynamic Neural Field (DNF) Theory

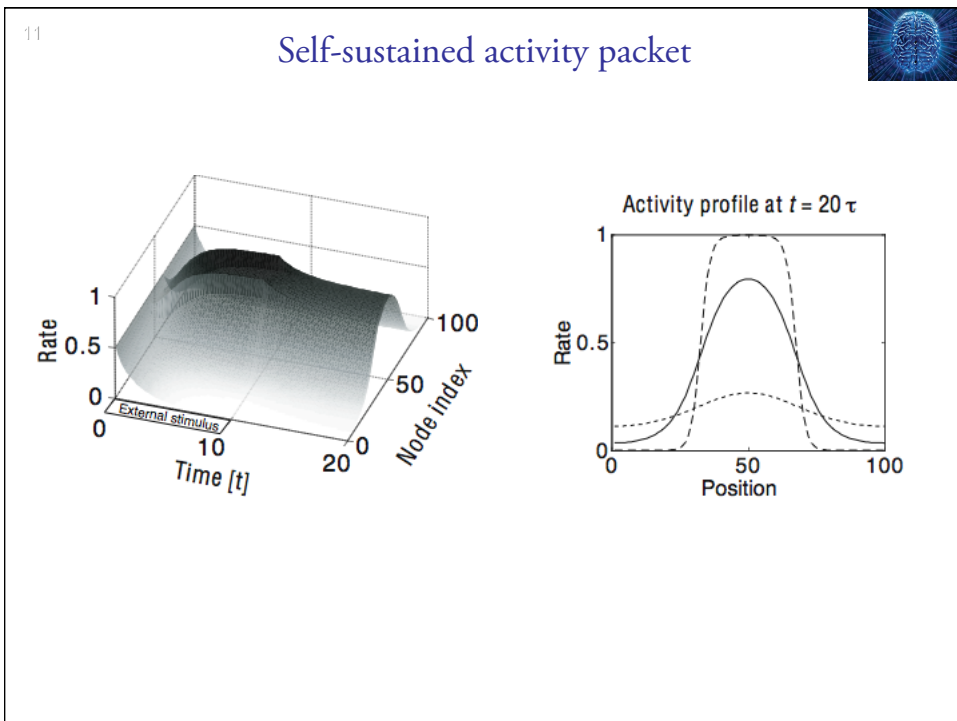
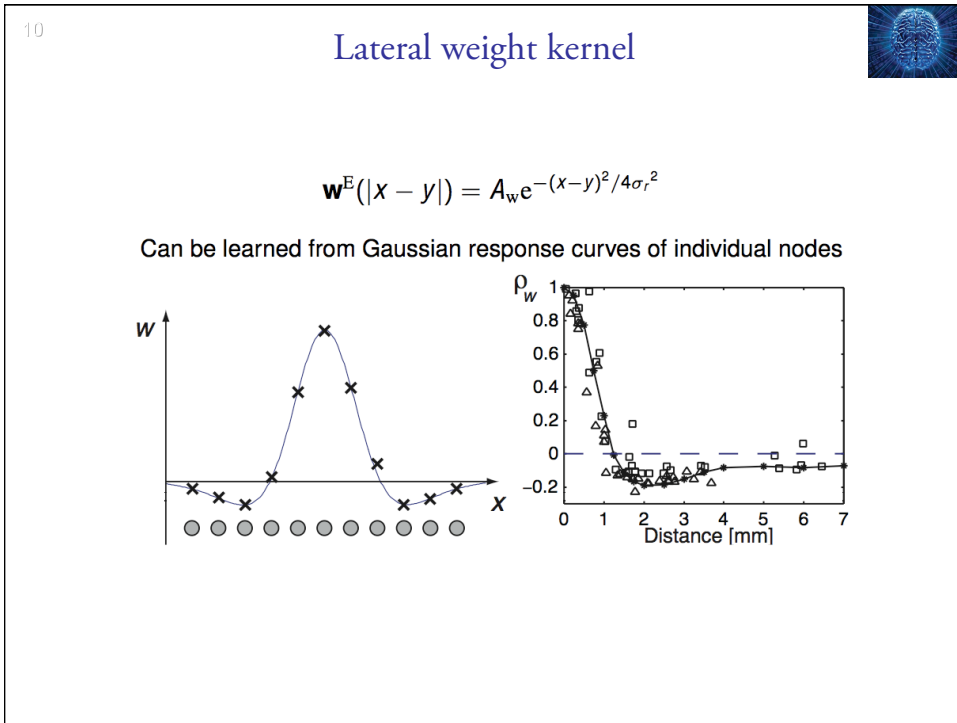
Field dynamics:

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

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

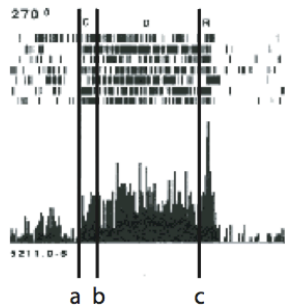
Continuous version of equations above with discretization:

$$x \rightarrow i\Delta x \text{ and } \int dx \rightarrow \Delta x \sum$$

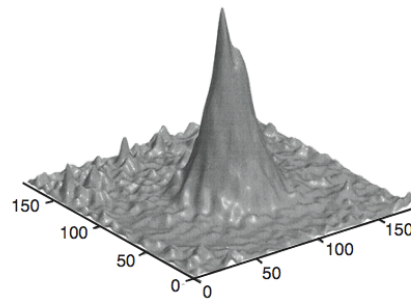


12

DNF example



PFC (Funahashi, Bruce & Goldman-Rakic, 1989)



Hippocampus (Samsonovich & McNaughton, 1997)

13

dnf.m



```

1  %% Dynamic Neural Field Model (1D)
2  clear; clf; hold on;
3  nn = 100; dx=2*pi/nn; sig = 2*pi/10; C=0.5;
4
5  %% Training weight matrix
6  for loc=1:nn;
7      i=(1:nn)'; dis= min(abs(i-loc),nn-abs(i-loc));
8      pat(:,loc)=exp(-(dis*dx).^2/(2*sig^2));
9  end
10 w=pat*pat'; w=w/w(1,1); w=4*(w-C);
11 %% Update with localised input
12 tall = []; rall = [];
13 I_ext=zeros(nn,1); I_ext(nn/2-floor(nn/10):nn/2+floor(nn/10))=1;
14 [t,u]=ode45('rnn_ode',[0 10],zeros(1,nn),[],nn,dx,w,I_ext);
15 r=1./(1+exp(-u)); tall=[tall;t]; rall=[rall;r];
16 %% Update without input
17 I_ext=zeros(nn,1);
18 [t,u]=ode45('rnn_ode',[10 20],u(size(u,1),:),[],nn,dx,w,I_ext);
19 r=1./(1+exp(-u)); tall=[tall;t]; rall=[rall;r];
20 %% Plotting results
21 surf(tall',1:nn,rall','linestyle','none'); view(0,90);

```

14

rnn.ode



```

1 function udot=rnn_ode(t,u,flag,nn,dx,w,l_ext)
2 % odefile for recurrent network
3 tau_inv = 1.; % inverse of membrane time constant
4 r=1./(1+exp(-u));
5 sum=w*r*dx;
6 udot=tau_inv*(-u+sum+l_ext);
7 return

```

Update rule of (recurrent) cortical network:

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

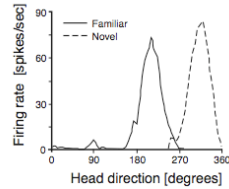
Activation function: $r_j(t) = \frac{1}{1 + e^{\beta(u_j(t) - \alpha)}}$

15

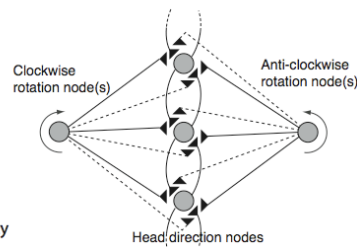
Path integration



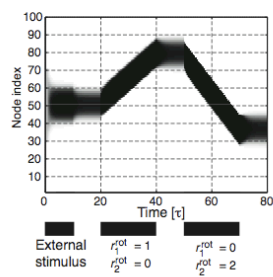
A. Head-direction cell in subiculum



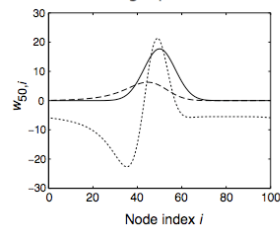
B. Head-direction model



C. Time evolution of network activity



D. Weight profiles



16

Population coding



Probability of neural response for a sensory input:

$$P(\mathbf{r}|\mathbf{s}) = P(r_1^s; r_2^s; r_3^s; \dots | \mathbf{s})$$

Decoding: $P(\mathbf{s}|\mathbf{r}) = P(\mathbf{s} | r_1^s; r_2^s; r_3^s; \dots)$

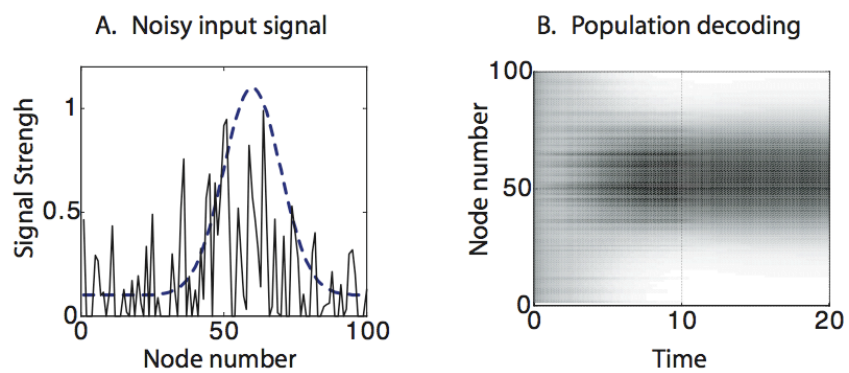
Stimulus estimate: $\hat{\mathbf{s}} = \arg \max_{\mathbf{s}} P(\mathbf{s}|\mathbf{r})$

Bayes's theorem: $P(\mathbf{s}|\mathbf{r}) = \frac{P(\mathbf{r}|\mathbf{s})P(\mathbf{s})}{P(\mathbf{r})}$

Maximum likelihood estimate: $\hat{\mathbf{s}} = \arg \min \sum_i \left(\frac{r_i - f_i(\mathbf{s})}{\sigma_i} \right)^2$

17

Implementation of decoding mechanisms with DNF



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.