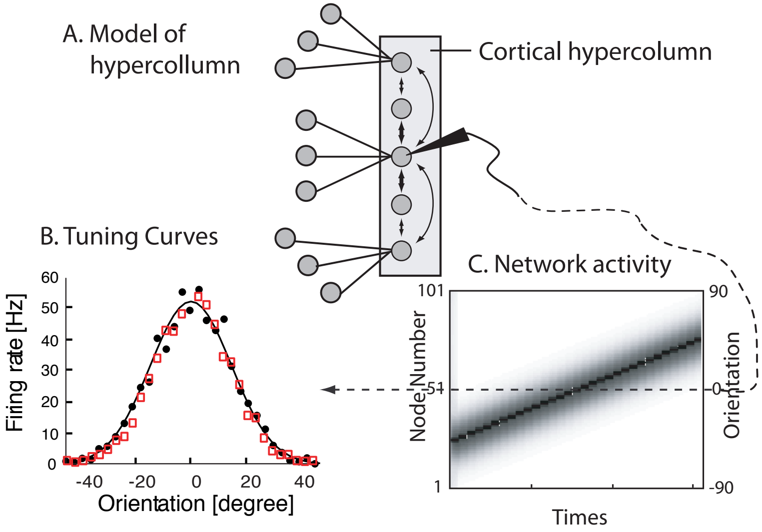
Fundamentals of Computational Neuroscience 2e

Thomas Trappenberg

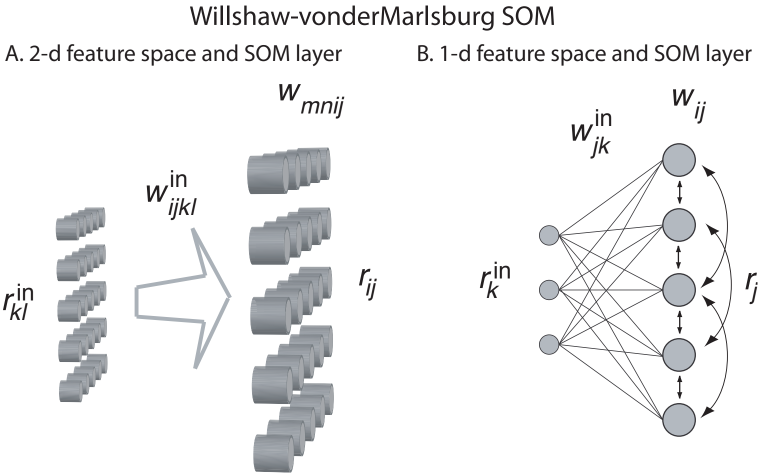
March 2, 2009

Chapter 7: Cortical maps and competitive population coding

Tuning Curves



Self-organizing maps



Update rule of (recurrent) cortical network:

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

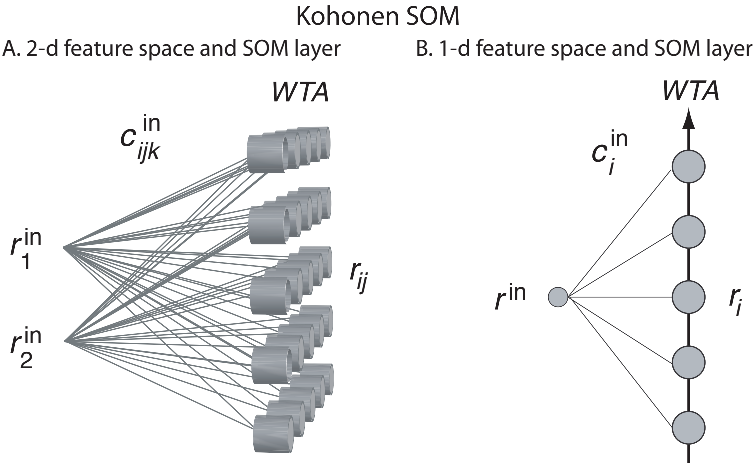
Activation function: $r_j(t) = \frac{1}{1+e^{\beta(u_j(t)-\alpha)}}$.

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

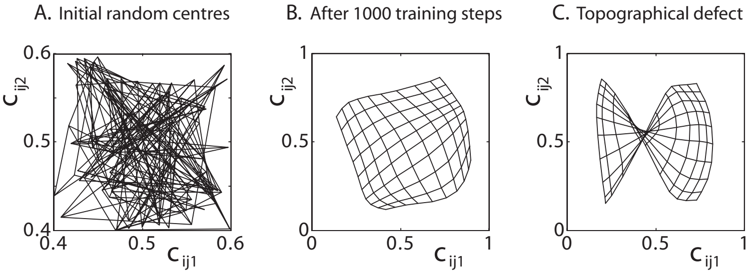
Shortcut


Kohonen SOM
A. 2-d feature space and SOM layer
B. 1-d feature space and SOM layer
WTA
c^{in}_{ijk}
r^{in}_1
r^{in}_2
r_{ij}
WTA
c^{in}_i
r^{in}
r_i

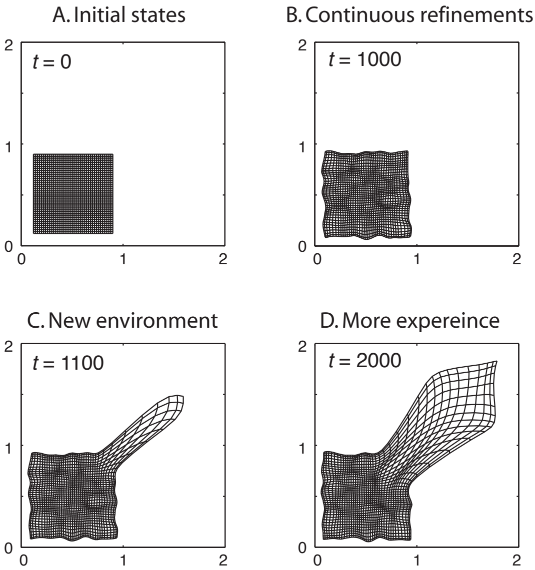
som.m

```
%% 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
```

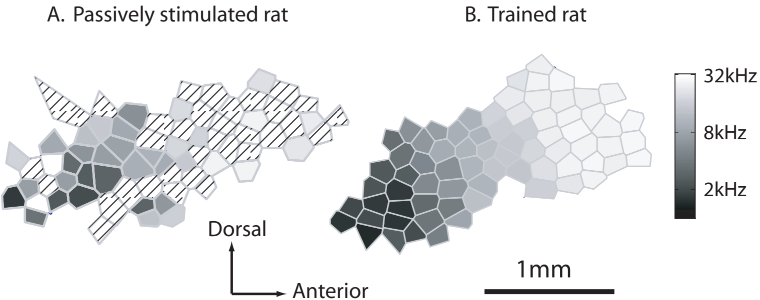
SOM simulation



Another example



Zhou and Merzenich, PNAS 2007



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:

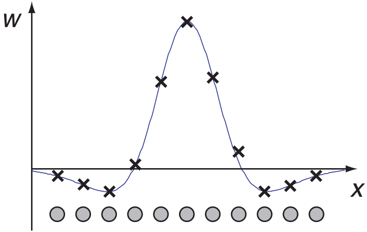
$$x \rightarrow i\Delta x \quad \text{and} \quad \int \mathrm{d}x \rightarrow \Delta x \sum$$

Lateral weight kernel

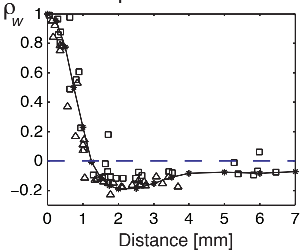
$$\mathbf{w}^{\mathrm{E}}(|x-y|) = A_{\mathrm{w}} \mathrm{e}^{-(x-y)^2/4\sigma_r{}^2}$$

Can be learned from Gaussian response curves of individual nodes

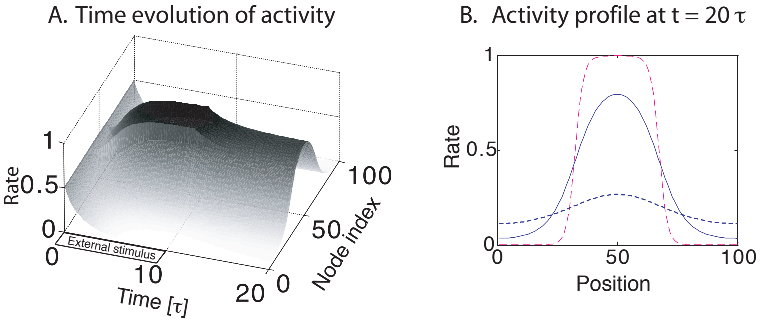
A. Mexican-hat weight kernel



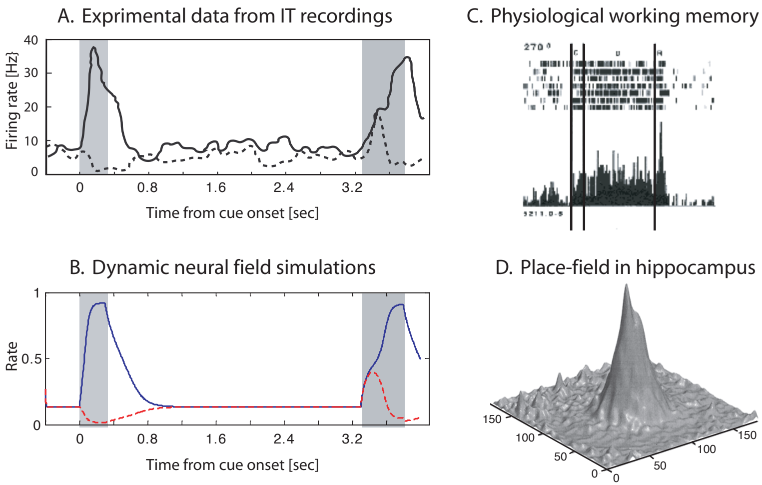
B. Effective interactions in the superior colliculus



Self-sustained activity packet



DNF example


A. Exprimental data from IT recordings
Firing rate [Hz}
40
30
20
10
0
0
0.8
1.6
2.4
3.2
Time from cue onset [sec]
C. Physiological working memory
B. Dynamic neural field simulations
Rate
1
0.5
0
0
0.8
1.6
2.4
3.2
Time from cue onset [sec]
D. Place-field in hippocampus
150
100
50
0
0
50
100
150

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

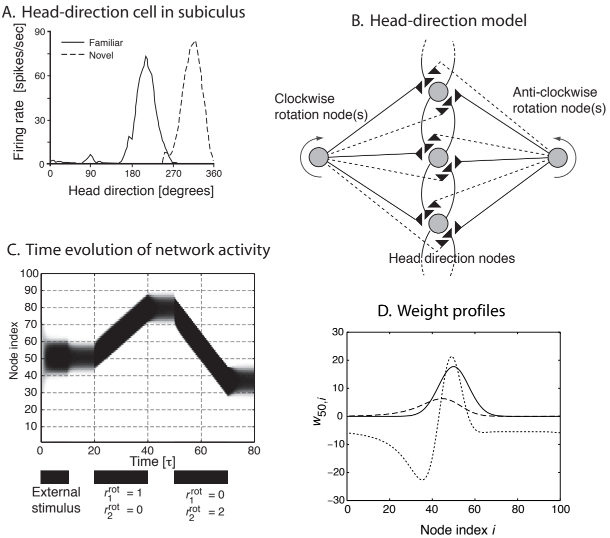
```
function udot=rnn_ode(t,u,flag,nn,dx,w,I_ext)
% odefile for recurrent network
 tau_inv = 1.;       % inverse of membrane time constant
 r=1./(1+exp(-u));
 sum=w*r*dx;
 udot=tau_inv*(-u+sum+I_ext);
return
```

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

Path integration



Population coding

Probability of neural response for a sensory input:
$P(\mathbf{r}|s) = P(r_1^s, r_2^s, r_3^s, ...|s)$

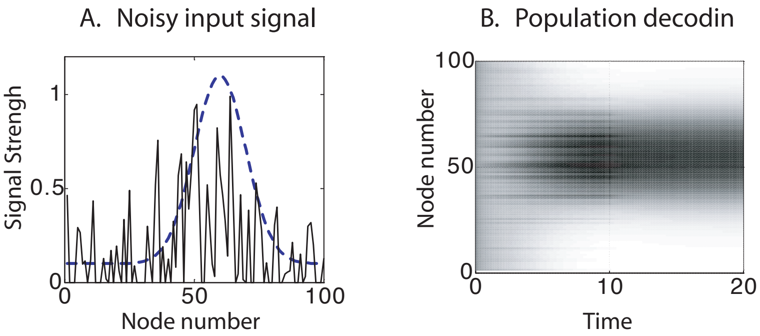
Decoding: $P(s|\mathbf{r}) = P(s|r_1^s, r_2^s, r_3^s, ...)$

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

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

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

Implementations 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.