Visual Object Recognition
Computational Models and Neurophysiological Mechanisms
Neurobiology 130/230. Harvard College/GSAS 78454

Web site: http://tinyurl.com/visionclass (Class notes, readings, etc)

Location: Biolabs 1075

Time: Mondays 03:30 – 05:30

Dates: Friday 09/04*, Mondays 09/14, 09/21, 09/28, 10/05, 10/19, 10/26, 11/02, 11/09, **11/16**, 11/23, 11/30, 12/07*

Lectures:

Faculty: Gabriel Kreiman and invited guests

Contact information:

Gabriel Kreiman
gabriel.kreiman@tch.harvard.edu
617-919-2530
Office Hours: After Class. Mon 05:30-06:30

Visual Object Recognition
Computational Models and Neurophysiological Mechanisms
Neurobiology 230. Harvard College/GSAS 78454

Class 1. Sep-04 Introduction to pattern recognition. Why is vision difficult?
Class 2. Sep-14 Visual input. Natural image statistics. The retina.
Class 3. Sep-21 Psychophysics of visual object recognition [Ken Nakayama]
Class 4. Sep-28 Lesion studies in animal models. Neurological studies of cortical visual deficits in humans.
Class 5. Oct-05 Introduction to the thalamus and primary visual cortex [Camille Gomez-Laberge]
Oct-12 *Columbus Day. No class.*
Class 6. Oct-19 Adventures into *terra incognita*. Neurophysiology beyond V1 [Hanlin Tang]
Class 7. Oct-26 First steps into inferior temporal cortex [Carlos Ponce]
Class 8. Nov-02 From the highest echelons of visual processing to cognition [Leyla Isik]
Class 9. Nov-09 Correlation and causality. Electrical stimulation in visual cortex.
Class 10. Nov-16 Theoretical neuroscience. Computational models of neurons and neural networks.
Class 11. Nov-23 Computer vision. Towards artificial intelligence systems for cognition [Bill Lotter]
Class 12. Dec-03 Computational models of visual object recognition.

OUTLINE

1. **Why build computational models?**
2. Single neuron models
3. Network models
4. Algorithms and methods for data analysis

Why bother with computational models?

-Quantitative models force us to think about and formalize hypotheses and assumptions

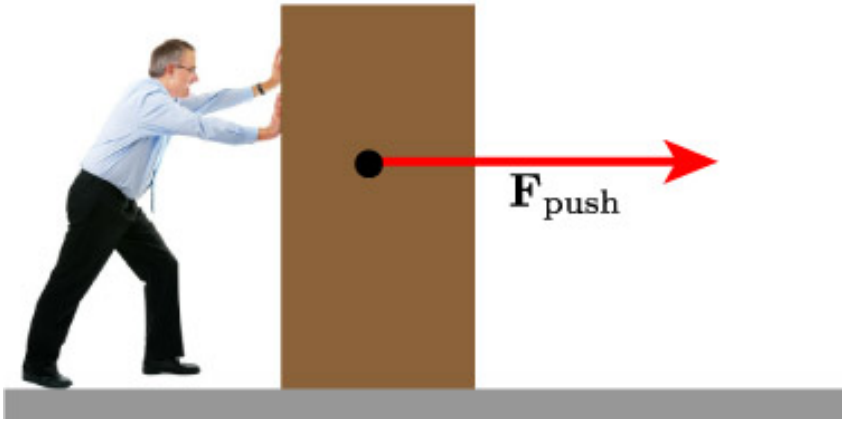
-Models can integrate and summarize observations across experiments, resolutions and laboratories

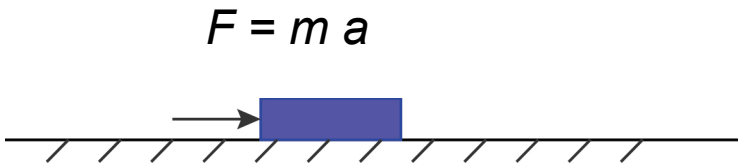
-A good model can lead to (non-intuitive) experimental predictions

-A quantitative model, implemented through simulations, can be useful from an engineering viewpoint (e.g. face recognition)

-A model can point to important missing data, critical information and decisive experiments

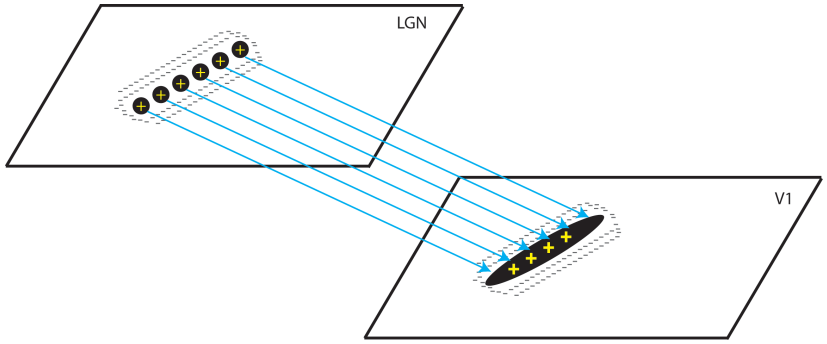
What is a model, anyway?





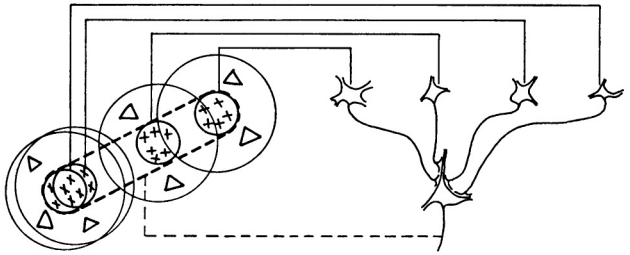
Which hand was the person using?
What is the shape/color/material of the object?
What day of the week is it?
What type of surface is it?
What is the temperature/humidity?
What is the force exerted by the person?
What is the weight of the object?
What is the force of gravity on this object?
Where is the force exerted?
What is the person wearing?
How much contact is there between the object and the surface?

A model for orientation tuning in simple cells

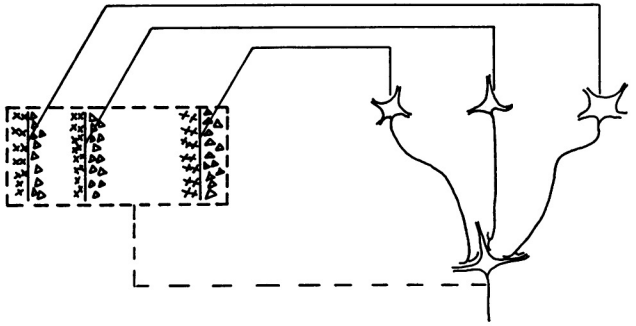


A feed-forward model for orientation selectivity in V1

(by no means the only model)



Text-fig. 19. Possible scheme for explaining the organization of simple receptive fields. A large number of lateral geniculate cells, of which four are illustrated in the upper right in the figure, have receptive fields with 'on' centres arranged along a straight line on the retina. All of these project upon a single cortical cell, and the



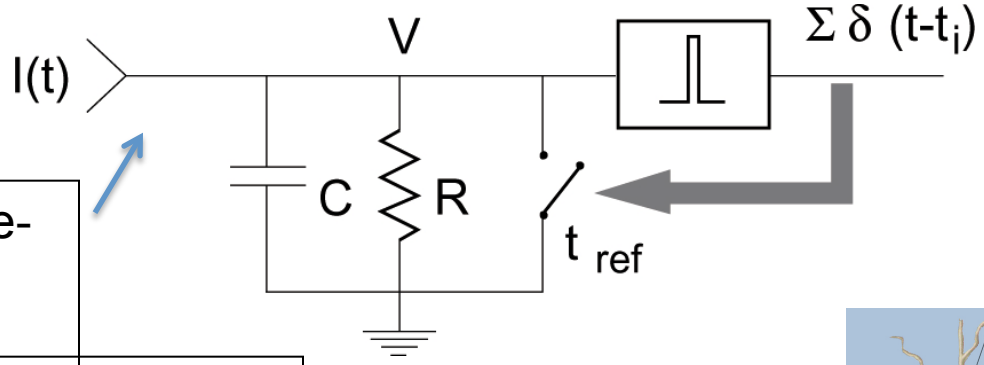
Text-fig. 20. Possible scheme for explaining the organization of complex receptive fields. A number of cells with simple fields, of which three are shown schematically, are imagined to project to a single cortical cell of higher order. Each projecting neurone has a receptive field arranged as shown to the left: an excitatory region to the left and an inhibitory region to the right of a vertical straight-line boundary. The boundaries of the fields are staggered within an area outlined by the interrupted lines. Any vertical-edge stimulus falling across this rectangle, regardless of its position, will excite some simple-field cells, leading to excitation of the higher-order cell.

Hubel and Wiesel. J. Physiology (1962)

OUTLINE

1. Why build computational models?
2. **Single neuron models**
3. Network models
4. Algorithms and methods for data analysis

A nested family of single neuron models



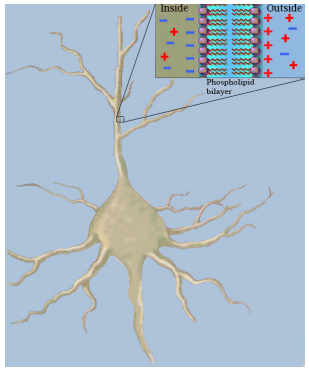
Filter operations

Integrate-and-fire circuit

Hodgkin-Huxley units

Multi-compartmental models

Spines, channels

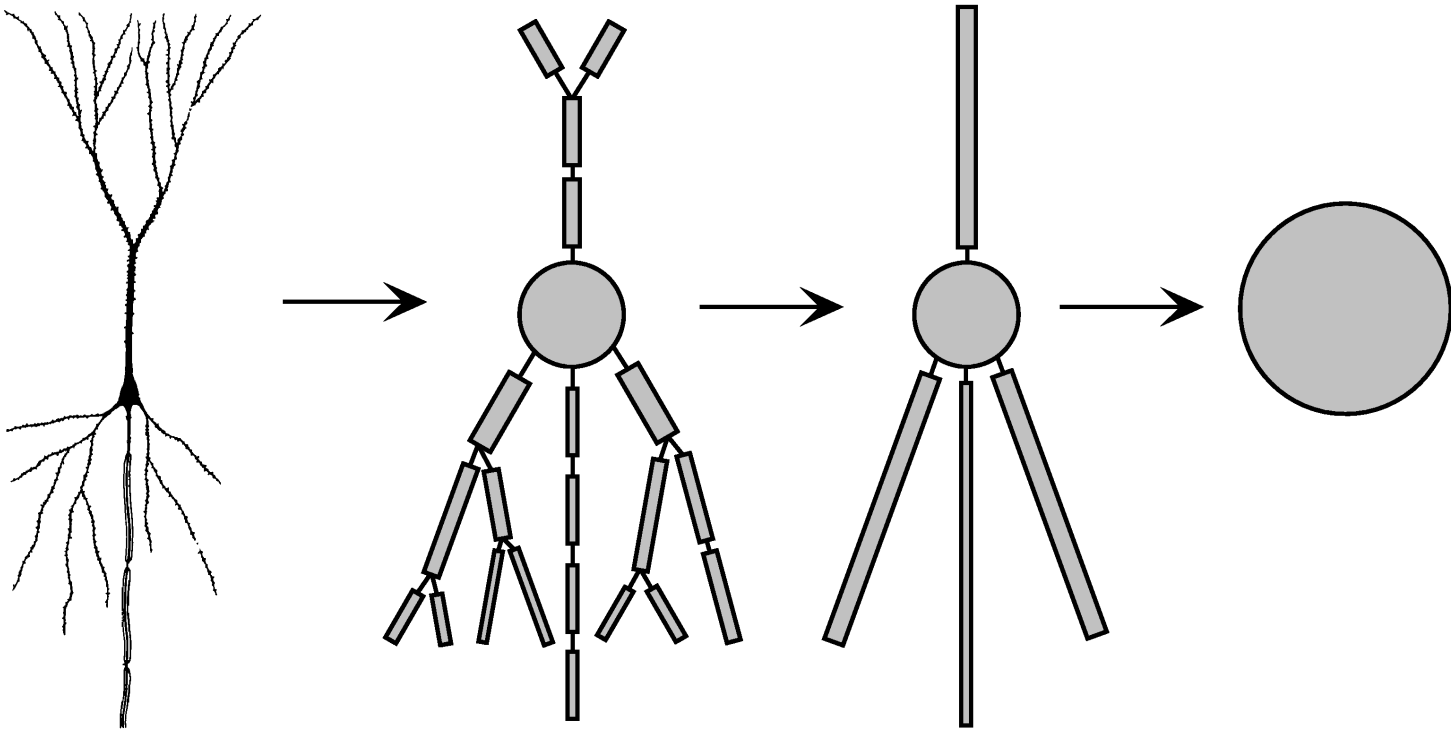


Biological accuracy

Lack of analytical solutions

Computational complexity

Geometrically accurate models vs. spherical cows with point masses



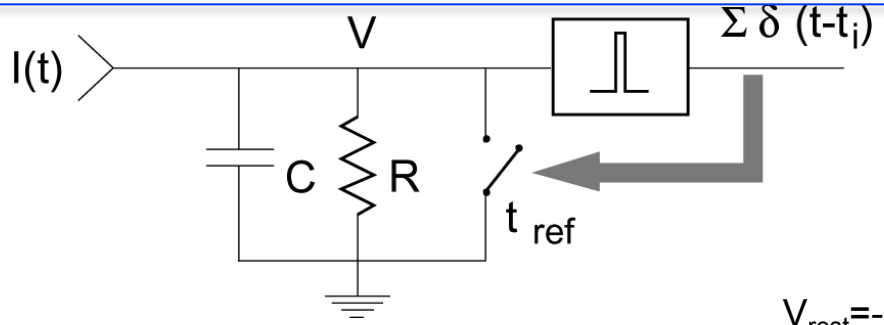
A central question in Theoretical Neuroscience:
What is the “right” level of abstraction?

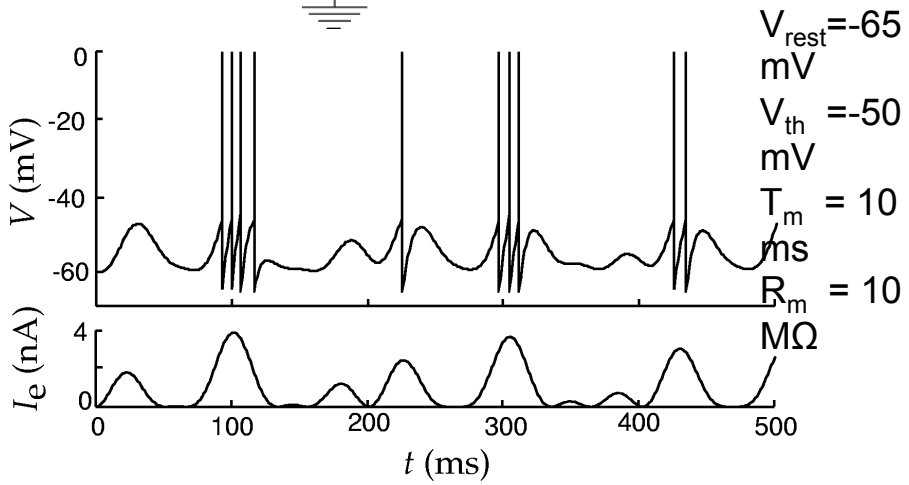
The leaky integrate-and-fire model

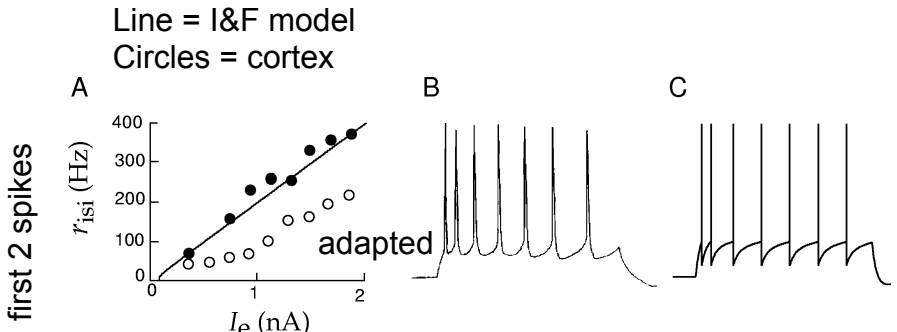
- Lapicque 1907
- Below threshold, the voltage is governed by:

$$C\frac{dV(t)}{dt} = -\frac{V(t)}{R} + I(t)$$

- A spike is fired when $V(t) > V_{thr}$ (and $V(t)$ is reset)
- A refractory period t_{ref} is imposed after a spike.
- Simple and fast.
- Does not consider spike-rate adaptation, multiple compartments, sub-ms biophysics, neuronal geometry





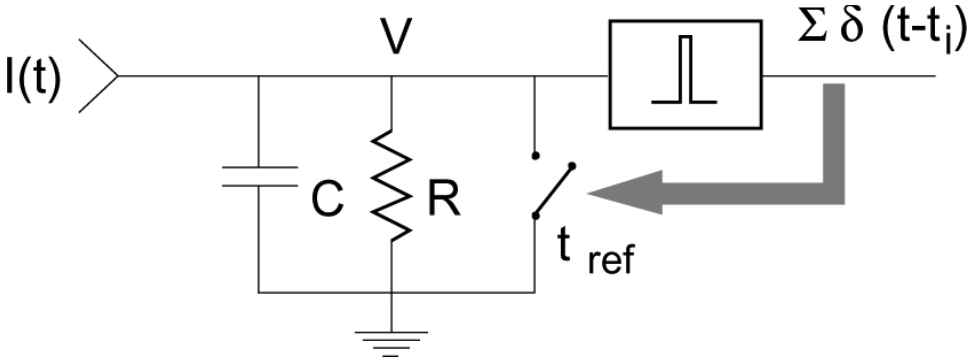


The leaky integrate-and-fire model

- Lapicque 1907
- Below threshold, the voltage is governed by:

$$C\frac{dV(t)}{dt} = -\frac{V(t)}{R} + I(t)$$

- A spike is fired when $V(t) > V_{thr}$ (and $V(t)$ is reset)
- A refractory period t_{ref} is imposed after a spike
- Simple and fast
- Does not consider:
  - spike-rate adaptation
  - multiple compartments
  - sub-ms biophysics
  - neuronal geometry




```
function
[V,spk]=simpleiandf(E_L,V_res,V_th,tau_m,R_m,I_e,dt
,n)

% ultra-simple implementation of integrate-and-fire
model
% inputs:
% E_L    = leak potential           [e.g. -65 mV]
% V_res  = reset potential          [e.g. E_L]
% V_th   = threshold potential      [e.g. -50 mV]
% tau_m  = membrane time constant   [e.g. 10 ms]
% R_m    = membrane resistance      [e.g. 10 MOhm]
% I_e    = external input           [e.g. white
noise]
% dt     = time step                [e.g. 0.1 ms]
% n      = number of time points    [e.g. 1000]
%
% returns
% V      = intracellular voltage    [n x 1]
% spk    = 0 or 1 indicating spikes [n x 1]

V(1)=V_res;      % initial voltage
spk=zeros(n,1);
for t=2:n
    V(t)=V(t-1)+(dt/tau_m) * (E_L - V(t-1) + R_m *
I_e(t));     % Key line computing the change in
voltage at time t
    if (V(t)>V_th)
% Emit a spike if V is above threshold
        V(t)=V_res;
% And reset the voltage
        spk(t)=1;
    end
end
```

Interlude: MATLAB is easy

$$C\frac{dV(t)}{dt} = -\frac{V(t)}{R} + I(t)$$

```
function [V,spk]=simpleiandf(E_L,V_res,V_th,tau_m,R_m,I_e,dt,n)

% ultra-simple implementation of integrate-and-fire model
% inputs:
% E_L    = leak potential           [e.g. -65 mV]
% V_res  = reset potential          [e.g. E_L]
% V_th   = threshold potential      [e.g. -50 mV]
% tau_m  = membrane time constant   [e.g. 10 ms]
% R_m    = membrane resistance      [e.g. 10 MOhm]
% I_e    = external input           [e.g. white noise]
% dt     = time step                [e.g. 0.1 ms]
% n      = number of time points    [e.g. 1000]
%
% outputs:
% V      = intracellular voltage    [n x 1]
% spk    = 0 or 1 indicating spikes [n x 1]

V(1)=V_res;       % initial voltage
spk=zeros(n,1);
for t=2:n
    V(t)=V(t-1)+(dt/tau_m) * (E_L - V(t-1) + R_m * I_e(t));      % Change in voltage at time t
    if (V(t)>V_th)                          % Emit a spike if V is above threshold
        V(t)=V_res;                             % And reset the voltage
        spk(t)=1;
    end
end
```

All of these lines are comments

This is the key line integrating the differential equation

The Hodgkin-Huxley Model

$$I(t) = C\frac{dV}{dt} + \bar{g}_L(V - E_L) + \bar{g}_K n^4(V - E_K) + \bar{g}_{Na} m^3 h(V - E_{Na})$$

where:
i_m = membrane current
V = voltage

L = leak channel
K = potassium channel
Na = sodium channel

g = conductances (e.g. g_{Na}=120 mS/cm^2; g_K=36 mS/cm^2; g_L=0.3 mS/cm^2)
E = reversal potentials (e.g. E_{Na}=115mV, E_K=-12 mV, E_L = 10.6 mV)
n, m, h = "gating variables", *n=n(t), m=m(t), h=h(t)*

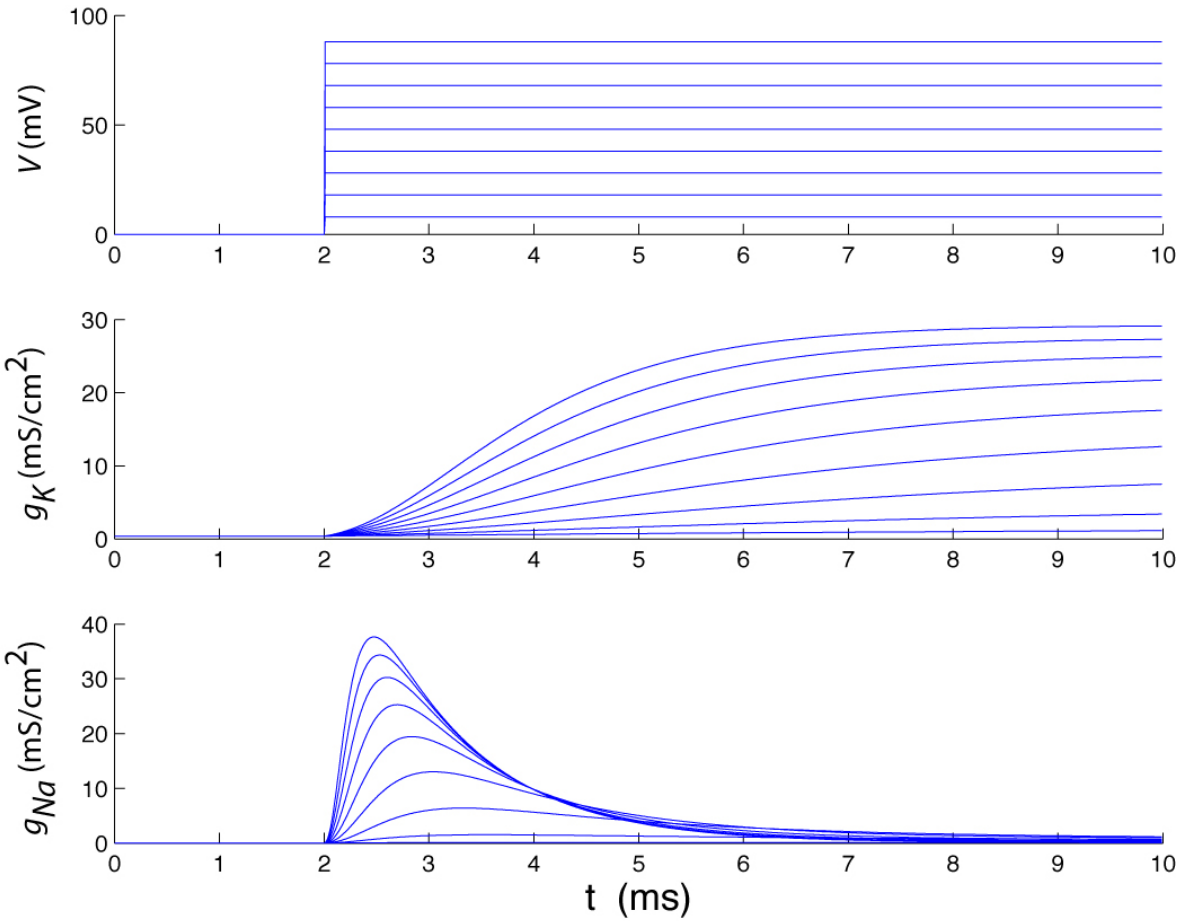
Hodgkin, A. L., and Huxley, A. F. (1952).
A quantitative description of membrane current and its application to conduction and excitation in nerve.
Journal of Physiology *117*, 500-544.

The Hodgkin-Huxley Model

```
% Gabbiani & Cox, Mathematics for Neuroscientists
% clamp.m
% Simulate a voltage clamp experiment
% usage:  clamp(dt,Tfin)
% e.g. clamp(.01,15)

function clamp(dt,Tfin)
vK = -6;     % mV
GK =  36;    % mS/(cm)^2
vNa = 127;   % mV
GNa = 120;   % mS/(cm)^2
for vc = 8:10:90,
    j = 2;t(1) = 0;v(1) = 0;
    n(1) = an(0)/(an(0)+bn(0));  % 0.3177;
    m(1) = am(0)/(am(0)+bm(0));  % 0.0529;
    h(1) = ah(0)/(ah(0)+bh(0));  % 0.5961;
    gK(1) = GK*n(1)^4;
    gNa(1) = GNa*m(1)^3*h(1);
    while j*dt < Tfin,
        t(j) = j*dt;
        v(j) = vc*(t(j)>2)*(t(j)<Tfin);
        n(j) = ( n(j-1) + dt*an(v(j)) )/(1 + dt*(an(v(
+bn(v(j))) );
        m(j) = ( m(j-1) + dt*am(v(j)) )/(1 + dt*(am(v(
+bm(v(j))) );
        h(j) = ( h(j-1) + dt*ah(v(j)) )/(1 + dt*(ah(v(
+bh(v(j))) );
        gK(j) = GK*n(j)^4;
        gNa(j) = GNa*m(j)^3*h(j);
        j = j + 1;
    end
    subplot(3,1,1); plot(t,v); hold on
    subplot(3,1,2); plot(t,gK); hold on
    subplot(3,1,3); plot(t,gNa); hold on
end
 subplot(3,1,1);ylabel('v','fontsize',16);hold off
subplot(3,1,2);ylabel('g_K','fontsize',16);hold off
subplot(3,1,3);xlabel('t  (ms)','fontsize',
16);ylabel('g_{Na}','fontsize',16);hold off

function val = an(v)
val = .01*(10-v)./(exp(1-v/10)-1);
function val = bn(v)
```



Simulated voltage-clamp experiments of Hodgkin and Huxley (1952). From Gabbiani and Cox 2010.

OUTLINE

1. Why build computational models?
2. Single neuron models
3. **Network models**
4. Algorithms and methods for data analysis

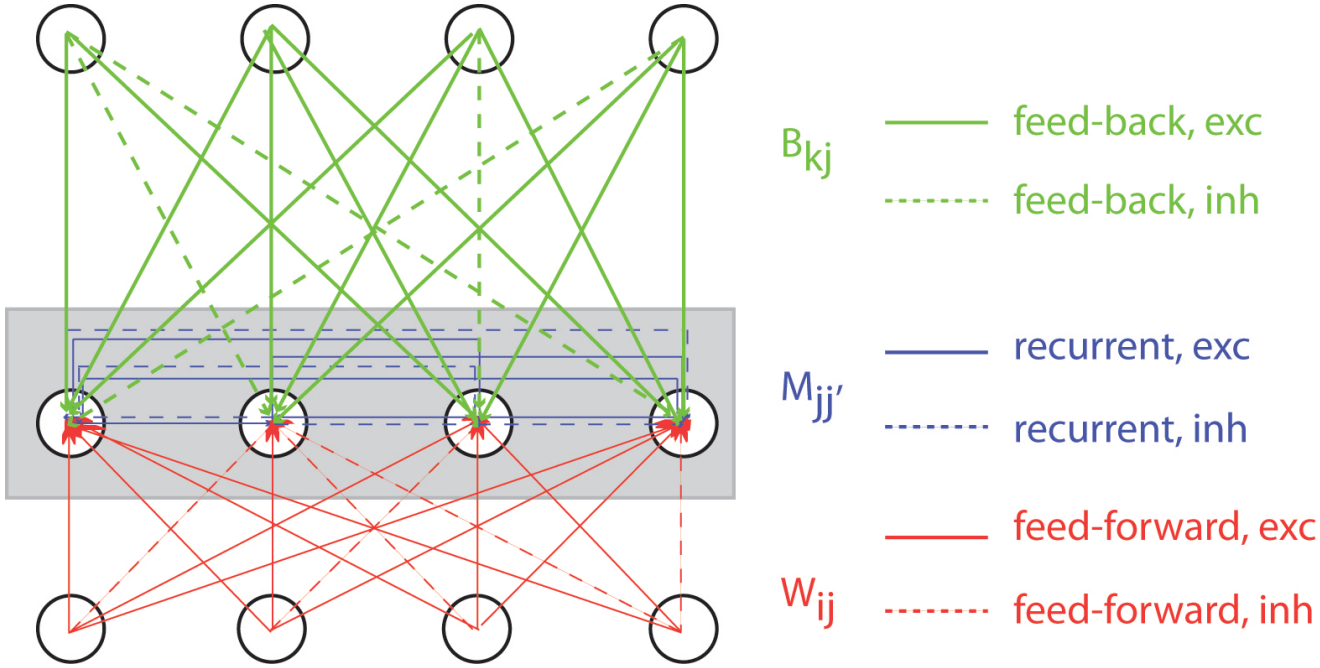
From neurons to circuits

- Single neurons can perform many interesting and important computations (e.g. Gabbiani et al (2002). Multiplicative computation in a visual neuron sensitive to looming. Nature 420, 320-324)

- Neurons are not isolated. They are part of circuits. A typical cortical neuron receives input from ~10^4 other neurons.

- It is not always trivial to predict circuit-level properties from single neuron properties. There could be interesting properties emerging at the network level.

Circuits – some basic definitions



Notes:

1. Connectivity does not need to be all-to-all
2. There are excitatory neurons and inhibitory neurons (and many types of inhibitory neurons)
3. Most models assume balance between excitation and inhibition
4. Most models do not include layers and the anatomical separation of forward and back pathways
5. There are many more recurrent+feedback connections than feed-forward connections (the opposite is true about models…)

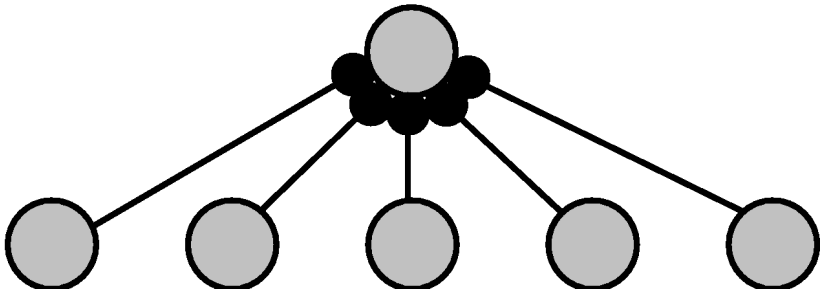
Firing rate network models – A simple feedforward circuit

- Time scales > ~ 1 ms
- Analytic calculations in some cases
- Fewer free parameters than spiking models
- Easier/faster to simulate

output v

weights $\mathbf{w}$

input $\mathbf{u}$



$$I_s = \sum_{b=1}^{N} w_b \int_{-\infty}^{t} d\tau K_s(t-\tau) u_b(\tau)$$

I_s = total synaptic current
N = total number of inputs
w_b = synaptic weights
$K_s(t)$ = synaptic kernel
u_b = input firing rates

$$\tau_s \frac{dI_s}{dt} = -I_s + \sum_{b=1}^{N} w_b u_b$$

if $K_s(t) = (1/\tau_s)\exp(-t/\tau_s)$

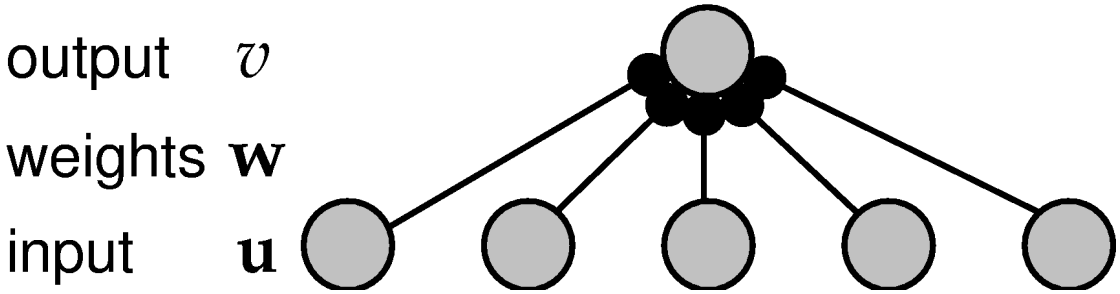
$$v = F(I_s)$$

F can be a sigmoid function
Or a threshold linear function:

$$F(I_s) = [I_s - \gamma]_+$$

Learning from examples – The perceptron

Imagine that we want to classify the inputs ***u*** into two groups "+1" and "-1"



$$v = \begin{cases} +1 & \text{if } \mathbf{w}.\mathbf{u} - \gamma \geq 0 \\ -1 & \text{if } \mathbf{w}.\mathbf{u} - \gamma < 0 \end{cases}$$

Training examples: $\{\mathbf{u}_m, v_m\}$

$$\mathbf{w} \rightarrow \mathbf{w} + \frac{\epsilon}{2}\left(v_m - v(\mathbf{u}_m)\right)\mathbf{u}_m$$

Perceptron learning rule

Linear separability: can attain zero error
Cross-validation: use separate training and test data
There are several more sophisticated learning algorithms

Learning from examples – Gradient descent

Now imagine that *v* is a real value (as opposed to binary)

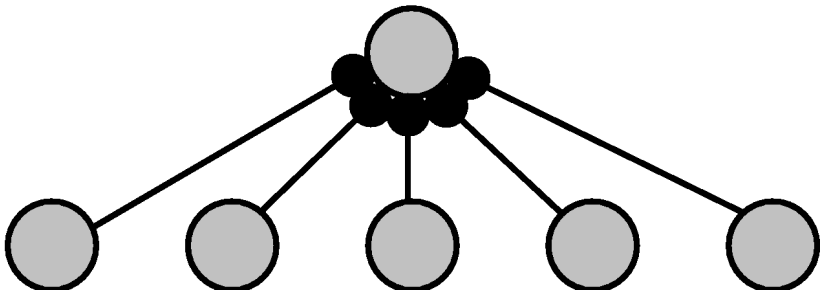
$$\mathbf{u} = \mathbf{f}(s)$$

$$v(s) = \mathbf{w}.\mathbf{u}$$

output v

weights $\mathbf{w}$

input $\mathbf{u}$



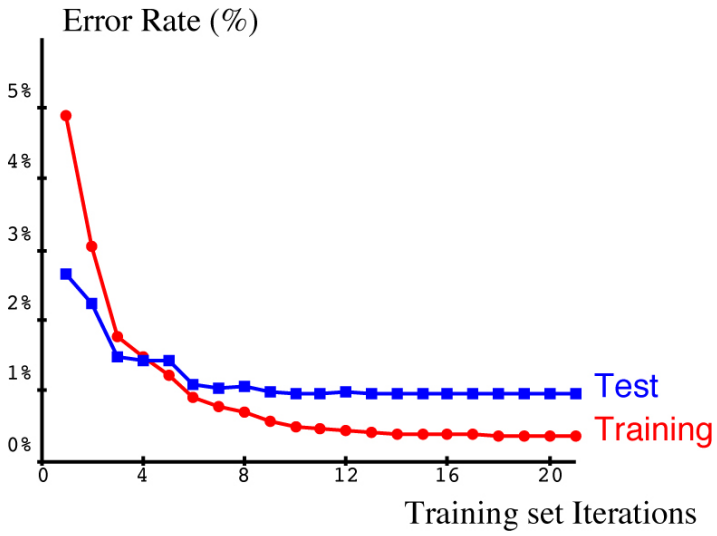
We want to choose the weights so that the output approximates some function *h(s)*

$$E = \frac{1}{2}\sum_{m=1}^{N_S}\left(h(s^m) - v(s^m)\right)^2$$

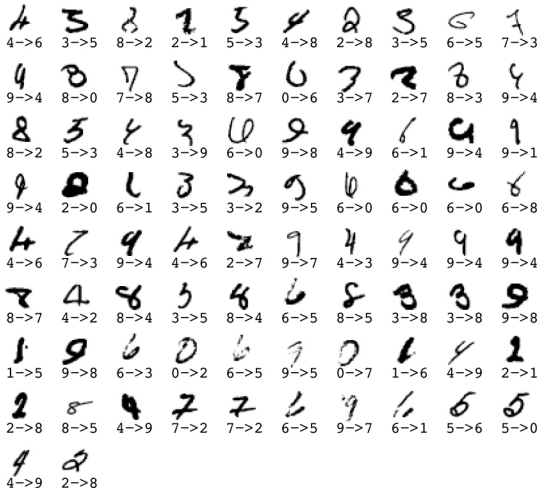
$$\mathbf{w} \rightarrow \mathbf{w} + \epsilon \nabla_{\mathbf{w}} E \qquad \nabla_{\mathbf{w}} E = \left[\frac{\partial E}{\partial w_b}\right]$$

Example: digit recognition in a feed-forward network trained by gradient descent

Example of hand-written digits (MINT database)



Classification error rates



Misclassified examples

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. Proc of the IEEE 86:2278-2324.

The “blue brain” modeling project

-http://bluebrain.epfl.ch

- IBM’s Blue gene supercomputer

- “Reverse engineer” the brain in a “biologically accurate” way

- November 2007 milestone: 30 million synapses in “precise” locations to model a neocortical column

- Compartmental simulations for neurons

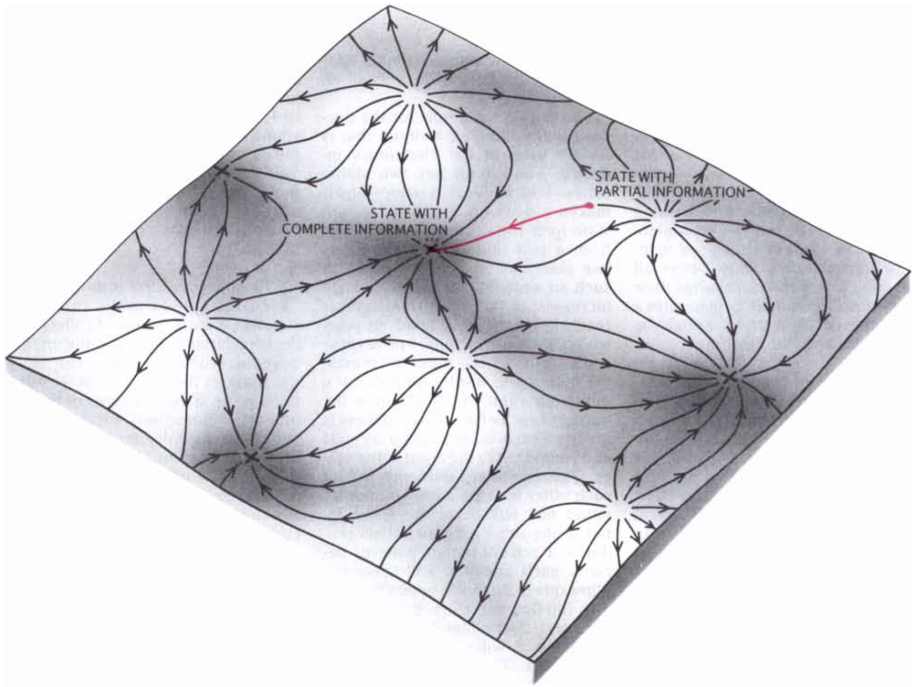
- Needs another supercomputer for visualization (10,000 neurons, high quality mesh, 1 billion triangles, 100 Gb)

QUESTION: What is the “right” level of abstraction needed to understand the function of cortical circuitry?

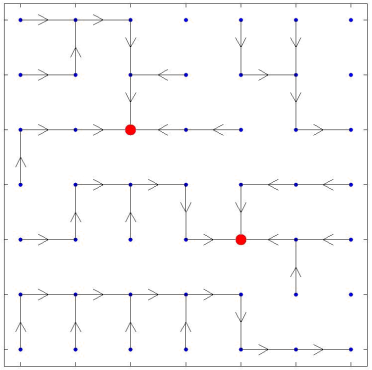
A case study in collective computation

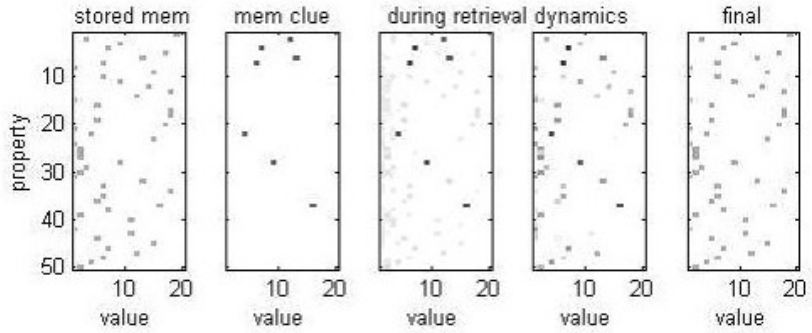
Hopfield, J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. **PNAS** 79:2554-2558.

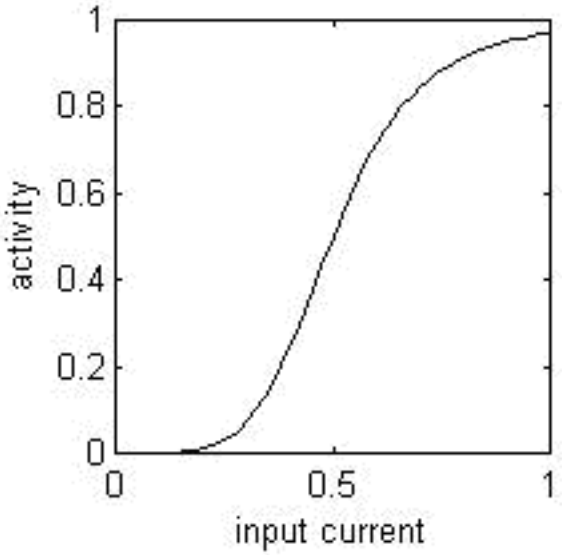
Tank, D., and J. Hopfield. 1987. Collective computation in neuron-like circuits. **Scientific American** 257:104-114

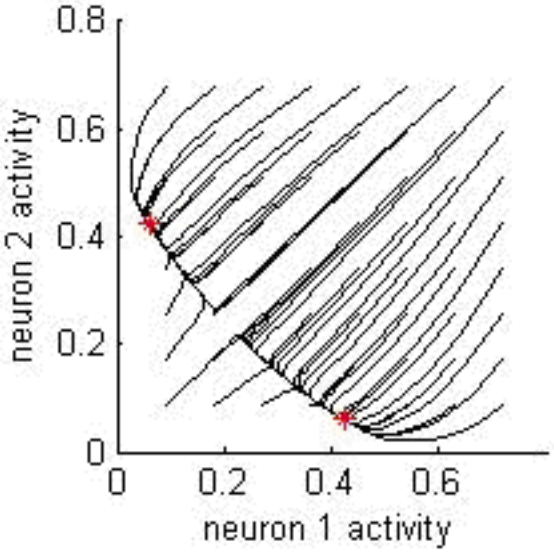


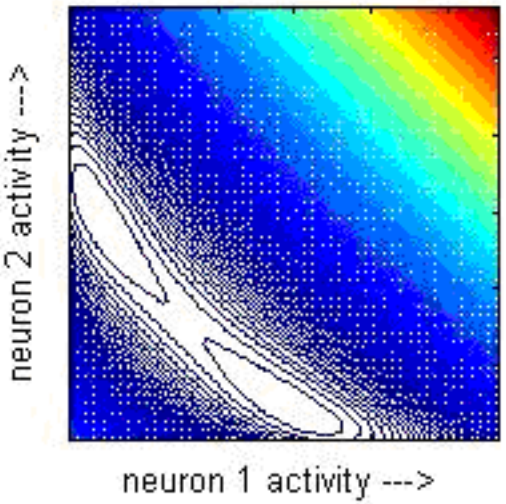
Primer on Hopfield networks











Hopfield. Hopfield Networks. Scholarpedia

OUTLINE

1. Why theoretical neuroscience?
2. Single neuron models
3. Network models
4. **Algorithms and methods for data analysis**

Some examples of computational algorithms and methods

- Different techniques for time-frequency analysis of neural signals (e.g. Pesaran et al 2002, Fries et al 2001)
- Spike sorting (e.g. Lewicki 1998)
- Machine learning approaches to decoding neuronal responses (e.g. Hung et al 2005, Wilson et al 1993, Musallam et al 2004)
- Information theory (e.g. Abbott et al 1996, Bialek et al 1991,)
- Neural coding (e.g. Gabbiani et al 1998, Bialek et al 1991)
- Definition of spatio-temporal receptive fields, phenomenological models, measures of neuronal synchrony, spike train statistics

Further reading

•Abbott and Dayan. Theoretical Neuroscience - Computational and Mathematical Modeling of Neural Systems [2001] (ISBN 0-262-04199-5). MIT Press.

•Koch. Biophysics of computation [1999] (ISBN 0-19-510491-9). Oxford University Press.

•Hertz, Krogh, and Palmer, *Introduction to the theory of neural computation*. [1991] (ISBN 0-20151560-1). Santa Fe Institute Studies in the Sciences of Complexity.

•Gabbiani and Cox. [2010]. Mathematics for Neuroscientists (London: Academic Press).