Visual Object Recognition

Neurobiology 230 – Harvard / GSAS 78454

Gabriel Kreiman

**Email:** gabriel.kreiman@tch.harvard.edu

**Phone:** 617-919-2530

**Web site:** [http://tinyurl.com/vision-class](http://tinyurl.com/vision-class)

**Dates:** Mondays

**Time:** 3:30 – 5:30 PM

**Location:** Biolabs 1075
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?

- Vertical Component of Velocity

- Undergoes accelerated motion

- Accelerated by gravity (9.8 m/s² down)

\[ V_y = V_{o,y} - gt \]

\[ y = y_o + V_{o,y} t - 1/2gt^2 \]

\[ V_y^2 = V_{o,y}^2 - 2g(y - y_o) \]
A model for orientation tuning in simple cells

A feed-forward model for orientation selectivity in V1
(by no means the only model)

Hubel and Wiesel. J. Physiology (1962)
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 \( V(t) > V_{\text{thr}} \) (and \( V(t) \) is reset)
- A refractory period \( t_{\text{ref}} \) is imposed after a spike.
- Simple and fast.
- Does not consider spike-rate adaptation, multiple compartments, sub-ms biophysics, neuronal geometry

Line = I&F model
Circles = cortex

First 2 spikes adapted
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 \( V(t) > V_{\text{th}} \) (and \( V(t) \) is reset)
- A refractory period \( t_{\text{ref}} \) is imposed after a spike
- Simple and fast
- Does not consider:
  - spike-rate adaptation
  - multiple compartments
  - sub-ms biophysics
  - neuronal geometry

\[
\frac{dV}{dt} = -\frac{V}{R} + \frac{I}{C}
\]

% 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
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;
       spk(t)=1;
    % And reset the voltage
    end
end

% All of these lines are comments

This is the key line integrating the differential equation

\[
C \frac{dV(t)}{dt} = -\frac{V(t)}{R} + I(t)
\]
The Hodgkin-Huxley Model

\[ I(t) = C \frac{dV}{dt} + g_L(V - E_L) + g_K n^4(V - E_K) + g_{Na} m^3 h(V - E_{Na}) \]

where:
- \( i_m = \text{membrane current} \)
- \( V = \text{voltage} \)
- \( L = \text{leak channel} \)
- \( K = \text{potassium channel} \)
- \( Na = \text{sodium channel} \)

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

The Hodgkin-Huxley Model

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 $> \sim 1$ ms
- Analytic calculations in some cases
- Fewer free parameters than spiking models
- Easier/faster to simulate

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

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

\[ v = F(I_s) \]

\[ I_s = \text{total synaptic current} \]
\[ N = \text{total number of inputs} \]
\[ w_b = \text{synaptic weights} \]
\[ K_s(t) = \text{synaptic kernel} \]
\[ u_b = \text{input firing rates} \]

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

\[ F \text{ can be a sigmoid function} \]
\[ F(I_s) = [I_s - \gamma]_+ \]
Imagine that we want to classify the inputs $\mathbf{u}$ 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: $\{\mathbf{u}_m, v_m\}$

$$\mathbf{w} \rightarrow \mathbf{w} + \frac{\epsilon}{2} \left( v_m - v(\mathbf{u}_m) \right) \mathbf{u}_m$$  \hspace{1cm} \text{Perceptron learning rule}$$

Linear separability: can attain zero error  
Cross-validation: use separate training and test data  
There are several more sophisticated learning algorithms
Now imagine that \( v \) is a real value (as opposed to binary):

\[
\begin{align*}
  u &= f(s) \\
  v(s) &= w \cdot u
\end{align*}
\]

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

\[
E = \frac{1}{2} \sum_{m=1}^{N_s} (h(s^m) - v(s^m))^2
\]

\[
\nabla_w E = \left[ \frac{\partial E}{\partial w_b} \right]
\]

\[
w \rightarrow w + \varepsilon \nabla_w E
\]
Example: digit recognition in a feed-forward network trained by gradient descent

Example of handwritten digits (MINT database)

Classification error rates

Misclassified examples

The “blue brain” modeling project

- [http://bluebrain.epfl.ch](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


Primer on Hopfield networks
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)


- 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


