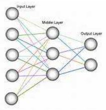
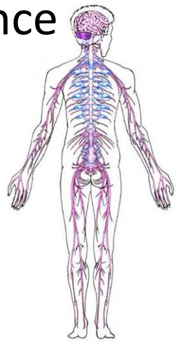
CISC 3250
Systems Neuroscience



Professor Daniel Leeds
dleeds@fordham.edu
JMH 328A

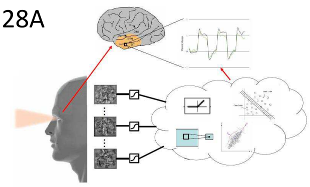
Systems Neuroscience

- How groups of neurons work together to achieve intelligence
- How the nervous system performs computations

- Requirement for the Integrative Neuroscience major
- Elective in Computer and Information Science

Your instructor

- Prof. Daniel Leeds
- E-mail: dleeds@fordham.edu
- Office hours: Tuesday and Thursday, 1-2pm
- Office: 328A



computer science + psychology -> brain models

Objectives

To understand information processing in biological neural systems from computational and anatomical perspectives

- Understand the function of key components of the nervous system
- Understand how neurons interact with one another
- Understand how to use computational tools to examine neural data

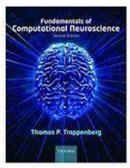
Recommended student background

- Prerequisite: CISC 2500 Data and Information Management – *not strict requirement this semester*

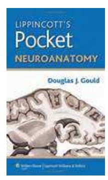
Math	Computer science
Some calculus	Some programming

Textbook(s)

Fundamentals of Computational Neuroscience, Second Edition, by Trappenberg



- **Required**
- We will focus on the ideas and study a relatively *small set* of equations



Lippincott's Pocket Neuroanatomy, by Gould

- **Optional**, better anatomy diagrams

Website

http://storm.cis.fordham.edu/leeds/cisc3250/

Go online for

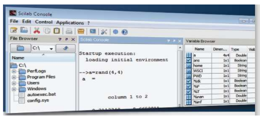
- Lecture slides
- Assignments
- Course materials/handouts
- Announcements

Requirements

- Attendance and participation
  - 1 unexcused absence allowed
  - Ask and answer questions in class
- Homework: Roughly 5 across the semester
- Exams
  - 2 midterms, in February and April
  - 1 final, in May
- Don't cheat
  - You may discuss homeworks with other students, but your submitted work must be your own

Software

We will use Scilab – an environment for numeric analyses and computational modeling.



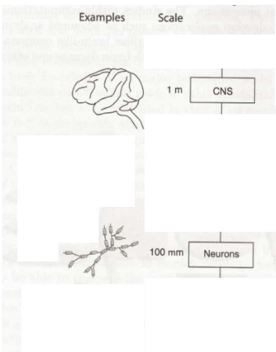
- Free
- Runs on all popular operating systems
- Similar to the very-popular Matlab®

http://www.scilab.org

Introducing systems and computational neuroscience

- How groups of neurons work together to achieve intelligence
- How the nervous system performs computations

Levels of organization



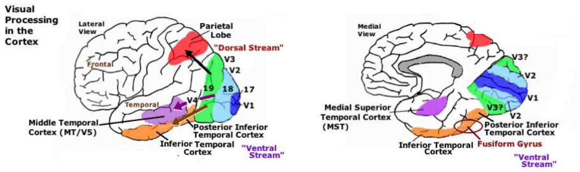
From a psychological perspective…

What are the elements of cognition?

Systems neuroscience

Regions of the central nervous system associated with particular elements of cognition

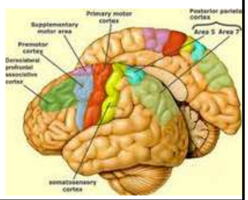
- Visual object recognition



Systems neuroscience

Regions of the central nervous system associated with particular elements of cognition

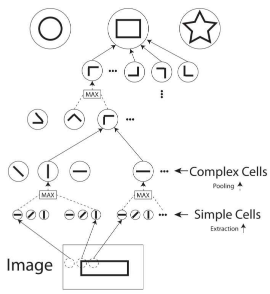
- Visual object recognition
- Motion planning and execution
- Learning and remembering



Computational neuroscience

Strategy used by the nervous system to solve problems

- Visual object perception through biological hierarchical model "HMAX"



Computational neuroscience as "theory of the brain"

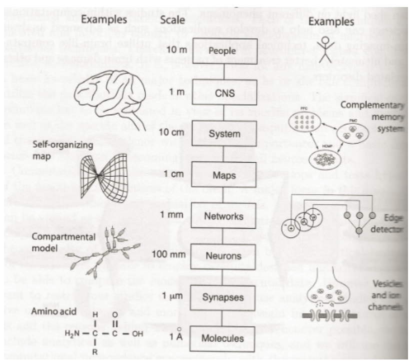
David Marr's three levels of analysis (1982):

- **Computational theory:** What is the computational goal and the strategy to achieve it?
- **Representation and algorithm:** What are the input and output for the computation, and how do you mathematically convert input to output?
- **Hardware implementation:** How do the physical components perform the computation?

Marr's three levels for "HMAX" vision

- **Computational theory:** Goal is to recognize objects
- **Representation and algorithm:**
  - **Input:** Pixels of light and color
  - **Output:** Label of object identity
  - **Conversion:** Through combining local visual properties
- **Hardware implementation:**
  - Visual properties "computed" by networks of firing neurons in object recognition pathway

Levels of organization

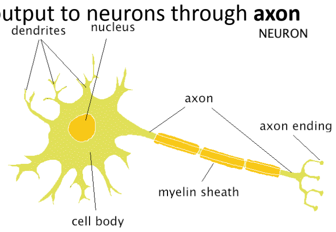


Course outline

- Philosophy of neural modeling
- The neuron – biology and input/output behavior
- Learning in the neuron
- Neural systems and neuroanatomy
- Information representation with features in computer science
- Representations in the brain
- Perception
- Memory/learning
- Motor control

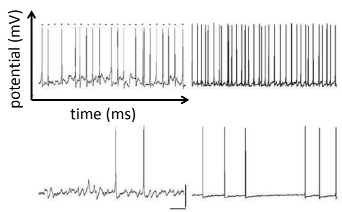
The neuron

- Building block of all the systems we will study
- Cell with special properties
  - **Soma** (cell body) can have 5-100 µm diameter, but **axon** can stretch over 10-1000 cm in length
  - Receives input from neurons through **dendrites**
  - Sends output to neurons through **axon**



Neuron membrane voltage

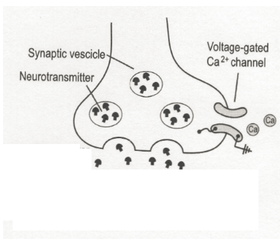
- Voltage difference across cell membrane
  - **Resting potential**: ~-65 mV
  - **Action potential**: quick positive **spike** in voltage



Example neural signals

More on the action potential

- Action potential begins at axon hillock and travels down axon
- At each axon terminal, spike results in release of **neurotransmitters**
- **Neurotransmitters (NTs)** attach to dendrite of another neuron, causing voltage change in this second neuron



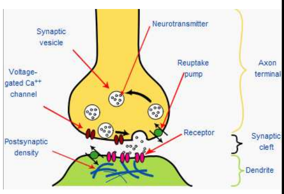
Inter-neuron communication

Neuron receives input from 1000s of other neurons

- **Excitatory** input can increase spiking
- **Inhibitory** input can decrease spiking

A **synapse** links neuron A with neuron B

- Neuron A is **pre-synaptic**: axon terminal outputs NTs
- Neuron B is **post-synaptic**: dendrite takes NTs as input



Modeling voltage over time

Equations focusing on **change** in voltage *v*

Components:

- Resting state potential (voltage) E_L
- Input voltages RI
- Time t

$$\tau \frac{dv(t)}{dt} = \underbrace{-(v(t) - E_L)}_{\text{change towards resting state}} + \underbrace{RI(t)}_{\text{incorporate new input information}}$$

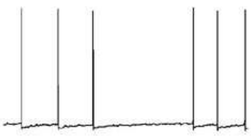
Voltage over time: reset

$$\tau \frac{dv(t)}{dt} = -(v(t) - E_L) + RI(t)$$

When voltage passes threshold v_{thresh}, voltage reset to v_{res}

$$v(t^f) = v_{thresh}$$
$$v(t^f + \delta) = v_{res}$$

δ is small positive number close to 0



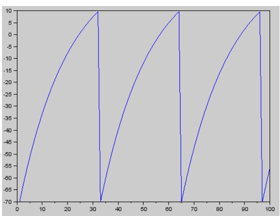
Voltage over time

Coding in scilab: $\tau \frac{dv(t)}{dt} = -(v(t) - E_L) + RI(t)$

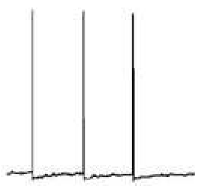
```
dt=0.001          // ms time increment
vCurr=-50         // vCurr is current voltage
vRest=-70         // vRest is resting voltage
vThresh=20        //vThresh is reset threshold
tau=20            // tau is scaling factor
for time = 1:100
  vCurr = vCurr+(input(time)-(vCurr-vRest))*dt/tau
```

Voltage over time

$$\tau \frac{dv(t)}{dt} = -(v(t) - E_L) + RI(t)$$

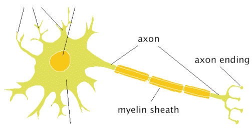


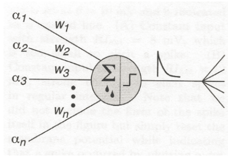
Simulated



Biological

Accumulating information over inputs





Positive and negative weighted inputs from dendrites α added together:

$$I(t) = \sum_j w_j \alpha_j(t)$$

j is **index** over dendrites; first-pass model

Form of dendrite input

$$\tau \frac{dv(t)}{dt} = -(v(t) - E_L) + RI(t)$$

Pre-synaptic neuron spikes

↓

Neurotransmitter (NT) released

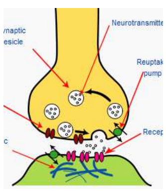
↓

NT received by **post-synaptic** dendrite at time t^f

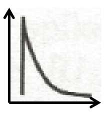
↓

Post-synaptic voltage spikes and fades, $\alpha(t-t^f)$

$$\boxed{I(t) = \sum_j \sum_{t_j^f} w_j \alpha(t - t_j^f)}$$



$\alpha(t-t^f)$



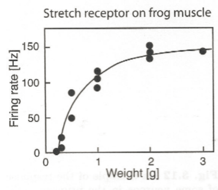
t

"Leaky integrate-and-fire" neuron

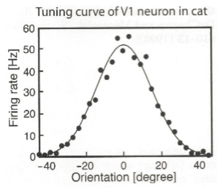
- Sum inputs from dendrites ("integral") $I(t) = \sum_j \sum_{t_j^f} w_j \alpha(t - t_j^f)$
- Decrease voltage towards resting state ("leak") $\tau \frac{dv(t)}{dt} = -(v(t) - E_L) + RI(t)$
- Reset after passing threshold ("fire") $v(t^f + \delta) = v_{res}$

Tuning curves

Some single neurons fire in response to "perceiving" a quality in the world



Adrian,
J Physiol 1926.

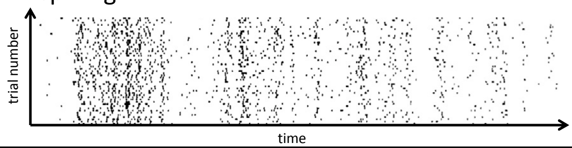


Henry et al.,
J Neurophys
1974.

Neural coding

Perception, action, and other cognitive states represented by firing of neurons

- Coding by rate: high rate of pre-synaptic spiking causes post-synaptic spiking
- Coding by spike timing: multiple pre-synaptic neurons spiking together causes post-synaptic spiking



Computing spike rate

- Add spikes over a period of time

$$v(t) = \frac{num\ spikes\ in\ \Delta T}{\Delta T}$$

- Average spikes over a set of neurons

$$A(t) = \lim_{\Delta T \to 0} \frac{1}{\Delta T} \frac{num\ spikes\ in\ N\ neurons}{N}$$

Activation function

Often non-linear relation between dendrite input and axon output $\tau \frac{dv(t)}{dt} = -(v(t) - E_L) + RI(t)$

$I(t) = \sum_j \sum_{t_j^f} w_j \alpha(t - t_j^f)$ Sum inputs

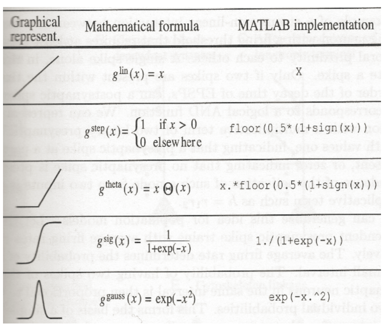
$\downarrow$

$RI(t)$ Multiply by *R*

$\downarrow$

$g(RI(t))$ Apply (non-linear?) transformation to input

Activation function



Function type	Graphical represent.	Mathematical formula	MATLAB implementation
Linear		$g^{\text{lin}}(x) = x$	`x`
Step		$g^{\text{step}}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{elsewhere} \end{cases}$	`floor(0.5*(1+sign(x)))`
Threshold-linear		$g^{\text{theta}}(x) = x\,\Theta(x)$	`x.*floor(0.5*(1+sign(x)))`
Sigmoid		$g^{\text{sig}}(x) = \frac{1}{1+\exp(-x)}$	`1./(1+exp(-x))`
Radial-basis		$g^{\text{gauss}}(x) = \exp(-x^2)$	`exp(-x.^2)`

Multiplying inputs

$$h_i = \sum_{jk} w_{ijk} r_j^{in} r_k^{in}$$