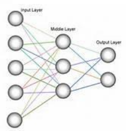
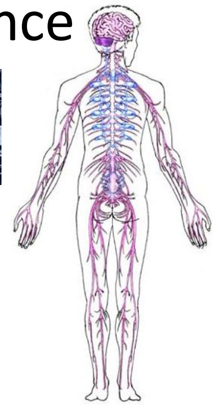
CISC 3250
Systems Neuroscience



Professor Daniel Leeds
dleeds@fordham.edu
JMH 332

Systems Neuroscience

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

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

2

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

3

Recommended student background

Prerequisite:

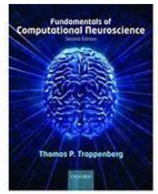
- Officially: CISC 2500 Information and Data Management
  or CISC 1800/1810 Intro to Programming

Math	Computer science
Some calculus	Some programming

4

Textbook(s)

Fundamentals of Computational Neuroscience, Second Edition, by Trappenberg



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

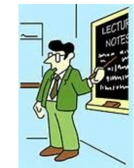
Computational Cognitive Neuroscience, by O'Reilly et al.

- **Optional**, alternate perspective

5

Website

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



Go online for

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

6

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 course topics with other students, but you must answer homeworks yourself (and exams!) yourself

7

Matlab

Popular tool in scientific computing for:

- Finding patterns in data
- Plotting results
- Running simulations

Student license for $50 on Mathworks site

Available in computers at JMH 330 and LL 612

8

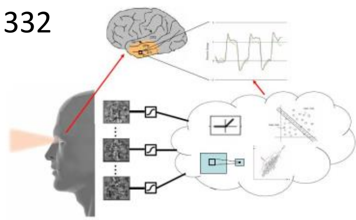
Your instructor

Prof. Daniel Leeds

E-mail: dleeds@fordham.edu

Office hours: **Mon 3-4, Thurs 12-1**

Office: JMH 332



computer science + psychology -> models of vision

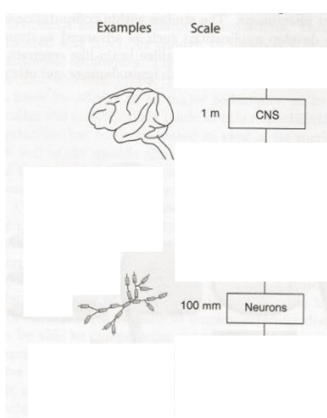
9

Introducing systems and computational neuroscience

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

11

Levels of organization



12

From a psychological perspective…

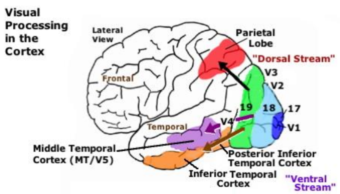
What are elements of cognition?

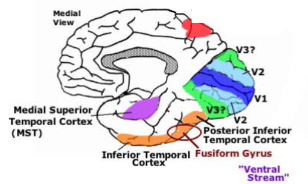
13

Systems neuroscience

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

- Visual object recognition



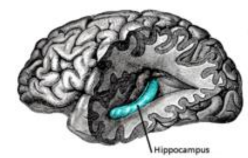


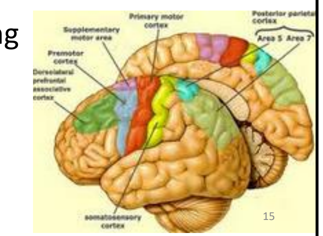
14

Systems neuroscience

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

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



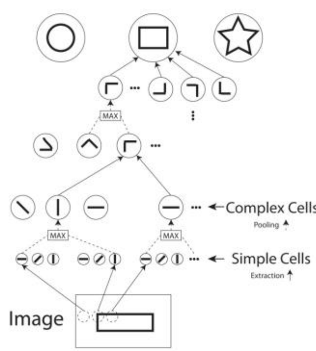


15

Computational neuroscience

Strategy used by the nervous system to solve problems

- Visual object perception through biological hierarchical model "HMAX"



16

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?

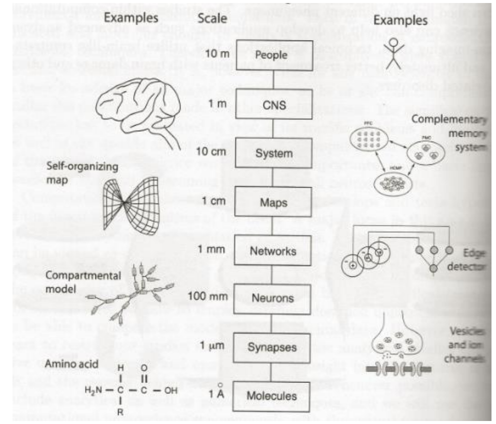
17

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

18

Levels of organization



19

Course outline

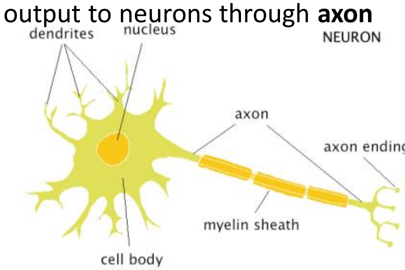
- Philosophy of neural modeling
- The neuron – biology and input/output behavior
- Learning in the neuron
- Neural systems and neuroanatomy
- Representations in the brain
- Perception
- Memory/learning
- Motor control

Plus: Matlab programming

20

The neuron

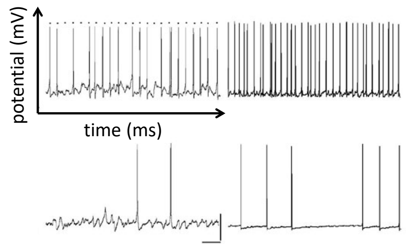
- Building block of all the systems we will study
- Cell with special properties
  - **Soma** (cell body) can have 5-100 μm diameter, but **axon** can stretch over 10-1000 cm in length
  - Receives input from neurons through **dendrites**
  - Sends output to neurons through **axon**



21

Neuron membrane voltage

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

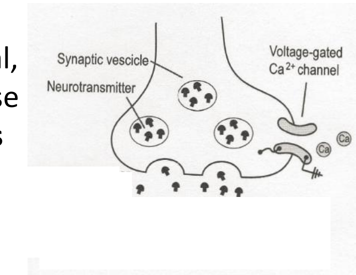


Example neural signals

22

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

23

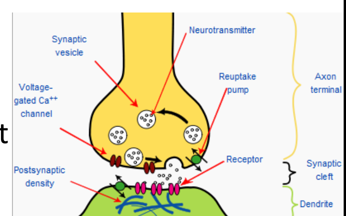
Inter-neuron communication

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



24

More on neuron membrane voltage

- Given no input, membrane stays at resting potential (~ -65 mV)

Inputs:

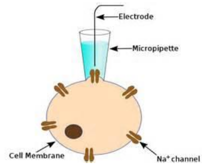
- Excitation temporarily increases potential
- Inhibition temporarily decreases potential

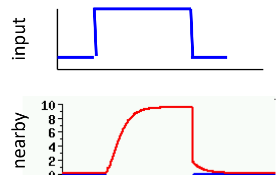
Continual drive to remain at rest

25

Patch clamp experiment

- Attach electrode to neuron
- Raise/drop voltage on electrode
- Measure nearby voltage (with another electrode)

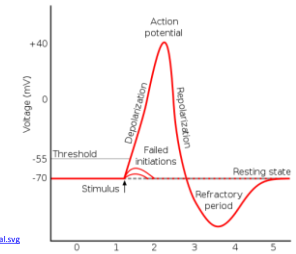




Simplification of neurophysiology experiment

More on the action potential

1. Accumulated excitation passes certain level
2. Non-linear increase in membrane voltage
3. Rapid reset



http://commons.wikimedia.org/wiki/File:Action_potential.svg
CC User: Chris 73

Modeling voltage over time

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

Components:

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

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

Simulation

- Initial voltage
- Time interval for update
- Input at each time
- Apply rule to compute new voltage at each time

Applying dv/dt step-by-step

E_L=-65mV v(0ms)=-65mV τ=1
RI(t)=20mV (from t=0ms to 1000ms)
time step: 10ms

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

- v(10ms) = v(0ms) + $\frac{dv(0\text{ms})}{dt}x\frac{10}{1000}$
  = -65 + **[-(-65- -65) + 20]** x $\frac{10}{1000}$
  = -65 + **20** x $\frac{10}{1000}$
  = **-64.8**
- v(20ms) = v(10ms) + $\frac{dv(10\text{ms})}{dt}x\frac{10}{1000}$
  = -64.8 + **[-(-64.8- -65) + 20]** x $\frac{10}{1000}$
  = -64.8 + **-0.2+20** x $\frac{10}{1000}$
  = -64.8 + **19.8** x $\frac{10}{1000}$
  = **-64.602**

Applying dv/dt step-by-step

E_L=-65mV v(0ms)=-65mV τ=1/10
RI(t)=20mV (from t=0ms to 1000ms)
time step: 10ms

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

- v(10ms) = v(0ms) + $\frac{dv(0\text{ms})}{dt}x\frac{10}{1000}$
  = -65 + **10x[-(-65- -65) + 20]** x $\frac{10}{1000}$
  = -65 + **200** x $\frac{10}{1000}$
  = **-63**
- v(20ms) = v(10ms) + $\frac{dv(10\text{ms})}{dt}x\frac{10}{1000}$
  = -63 + **10x[-(-63- -65) + 20]** x $\frac{10}{1000}$
  = -63 + **10x[-2+20]** x $\frac{10}{1000}$
  = -63 + **10x[180]** x $\frac{10}{1000}$
  = **-61.2**

Changing model terms

τ has inverse effect

- increase τ decreases update speed
- decrease τ increases update speed

RI(t) has linear effect

- increase RI(t) increases update speed
- decrease RI(t) decreases update speed

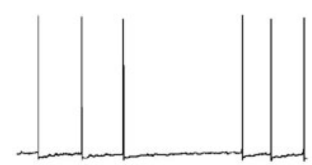
Voltage over time: reset

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

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

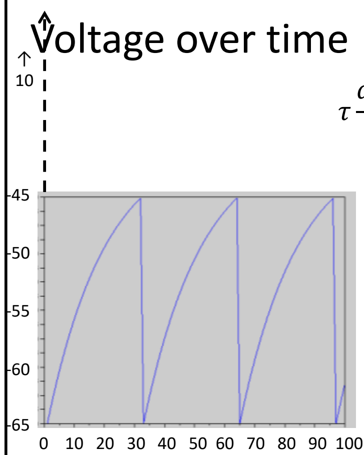
$$v(t^f)=v_{thresh}$$
$$v(t^f+\delta)=v_{res}$$

δ is small positive number close to 0

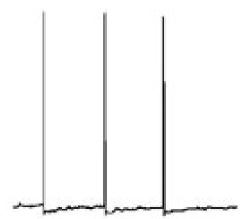


Voltage over time

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

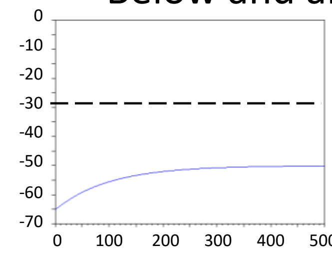


Simulated

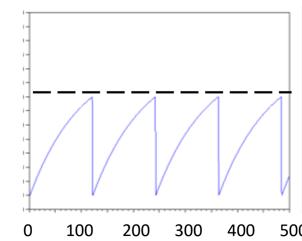

Biological

35

Below and above threshold



+15mv input



+50mv input

E_L=-65mV

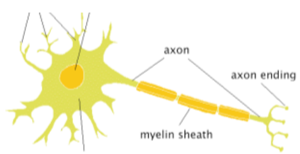
Newly added:

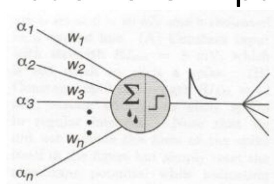
If input constant for long time RI(t)= k mV

Output v(t) will plateau to E_L+k if E_L+k<v_{thresh}

36

Accumulating information over inputs





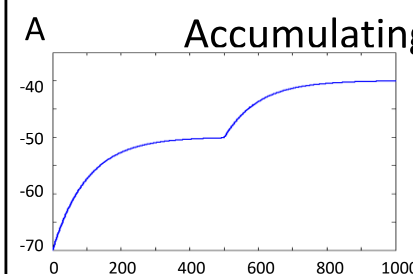
Positive and negative weighted inputs from dendrites *wα* added together:

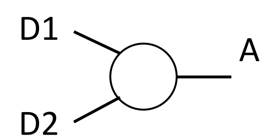
$$RI(t) = \sum_j w_j \alpha_j(t)$$

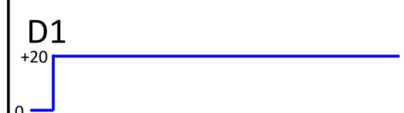
j is **index** over dendrites; first-pass model

38

Accumulating inputs

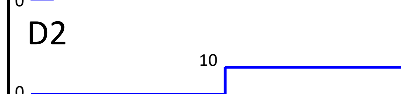






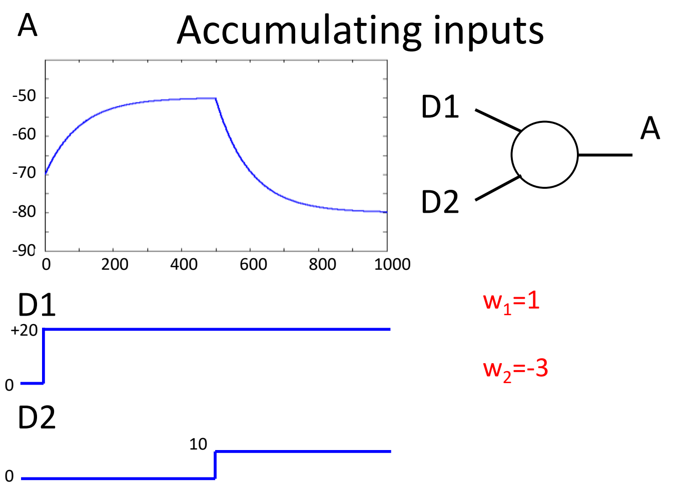
w_1=1

w_2=1



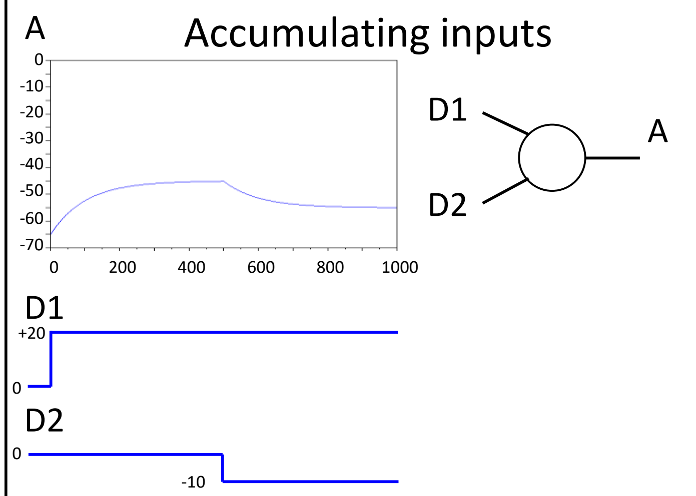
39

Accumulating inputs



41

Accumulating inputs



43

Chemical level: NT receptors

Pre-synaptic: α

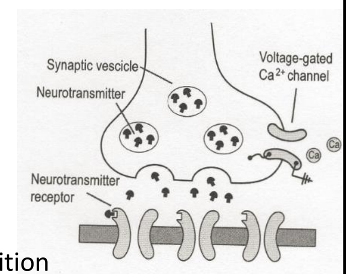
- Amount of NT released

Post-synaptic: w

- Number of receptors in dendrite membrane
- Efficiency of receptors

$+w$ or $-w$

- Reflect excitation or inhibition
- One NT type per synapse
- Fixed sign per NT



45

Form of dendrite input

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

Pre-synaptic neuron spikes

↓

Neurotransmitter (NT) released

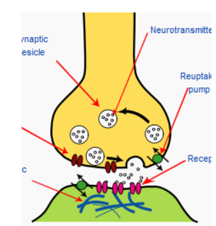
↓

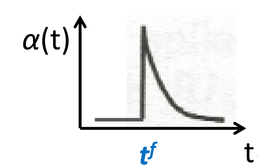
NT received by **post-synaptic** dendrite at time t^f

↓

Post-synaptic voltage rises and then fades, *α(t)*

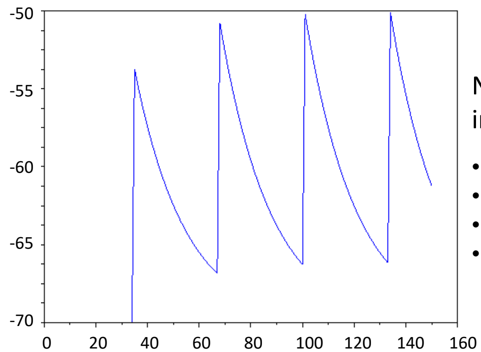
$$\boxed{RI(t) = \sum_j w_j \alpha_j\,(t)}$$





46

$$\boxed{RI(t) = \sum_j w_j \alpha_j (t)}$$



New pre-synaptic inputs at

- 34 ms
- 68 ms
- 100 ms
- 135 ms

47

"Leaky integrate-and-fire" neuron

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

48

Activation function

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

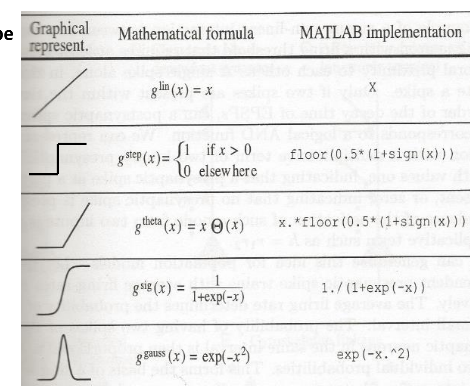
$RI(t) = \sum_j w_j \alpha_j(t)$ Sum inputs

$\downarrow$

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

49

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)`

50

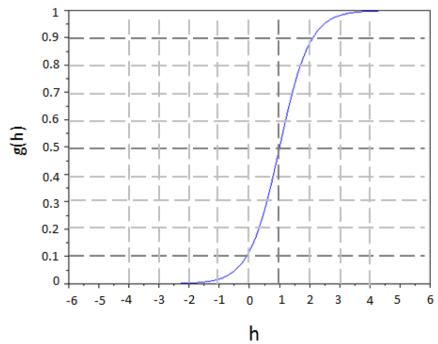
An example sigmoid

g(2)= **0.9**

g(1)= **0.5**

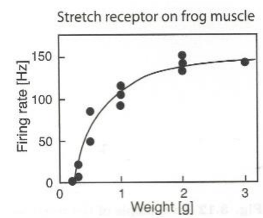
g(0)= **0.1**

g(-4)= **0**

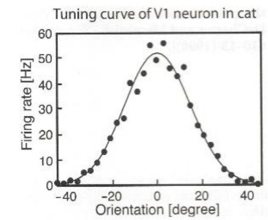


Tuning curves

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



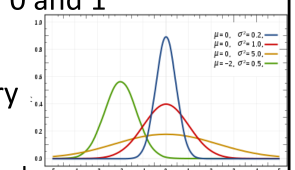
Adrian,
J Physiol 1926.



Henry et al.,
J Neurophys
1974.

Variations in activation functions

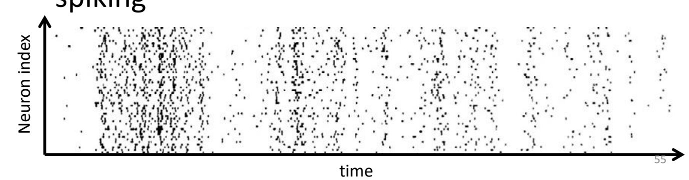
- Activation function has fixed shape
  - Sigmoid is S shape, Radial is Bell shape
- By default, transition between 0 and 1
- Some details of shape may vary
  - Smallest and lowest value
  - Location of transition between values



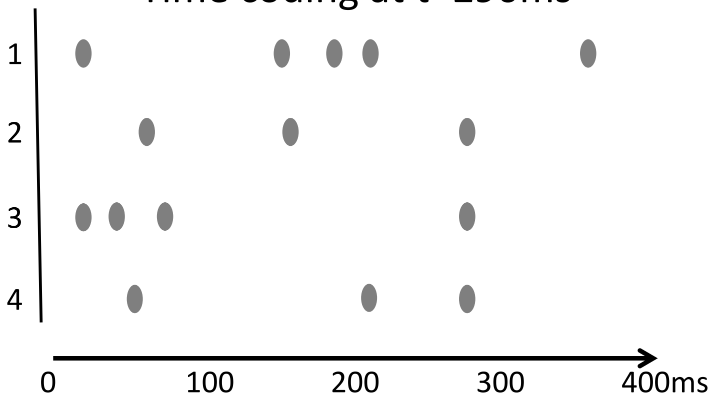
Neural coding

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

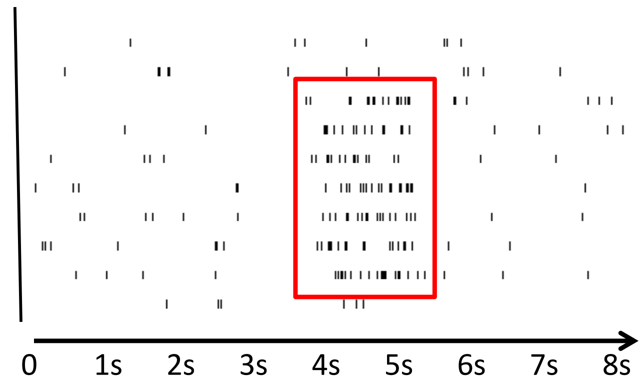


Time coding at t=290ms



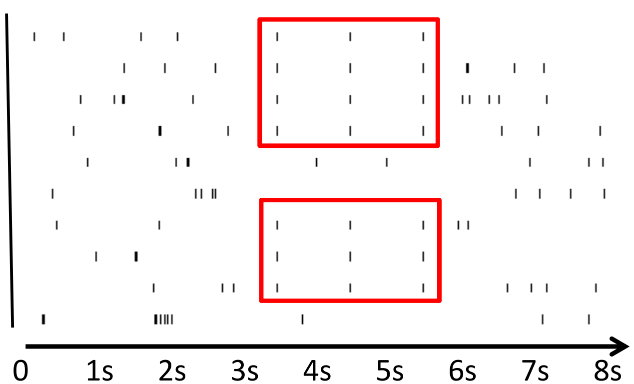
56

Rate coding: 3.5 – 5.5s



57

Spike time coding, 3-6s

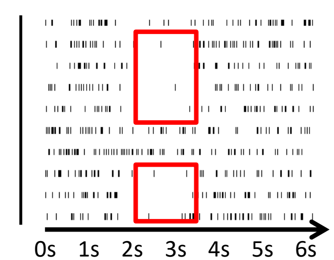


58

Inhibition can be informative

Inputs of interest can produce

- Below-normal spike rate
- Decreased synchrony among neurons



Coding through rate inhibition, roughly in 2-3s interval

Take note of baseline. Rate and time coding are **deviations** from baseline

59

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

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

60