TNI: Computational Neuroscience

Instructors: Peter Latham
Maneesh Sahani
Peter Dayan

TAs: Loic Matthey, loic.matthey@gatsby.ucl.ac.uk
Ritwik Niyogi, ritwik.niyogi@gatsby.ucl.ac.uk

Website: www.gatsby.ucl.ac.uk/~lmatthey/teaching/tn1/

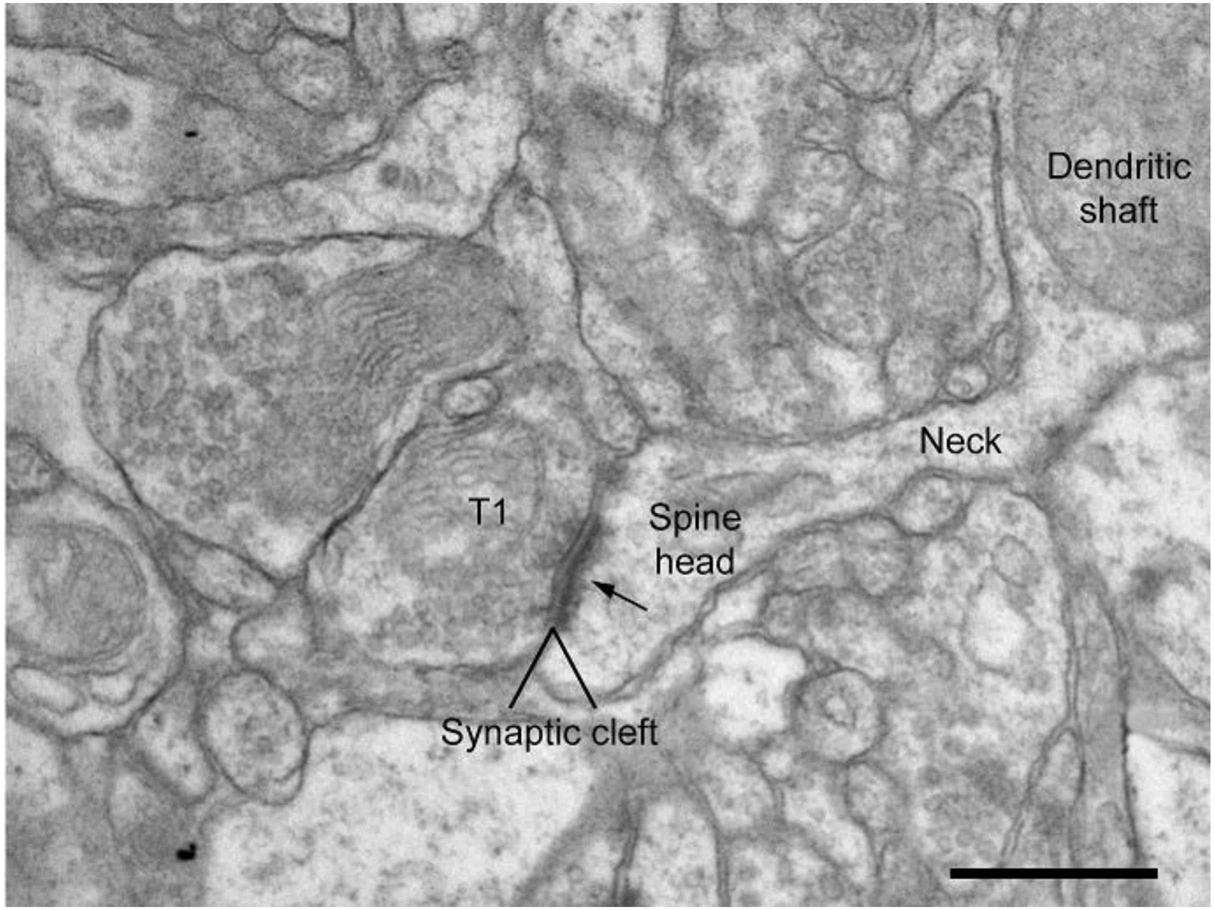
Lectures: Tuesday/Friday, 11:00-1:00.
Review: Tuesday, 4:30-6:30.

Homework: Assigned Friday, due Friday (1 week later).
first homework: assigned Oct. 8, due Oct. 15.

Theoretical Neuroscience

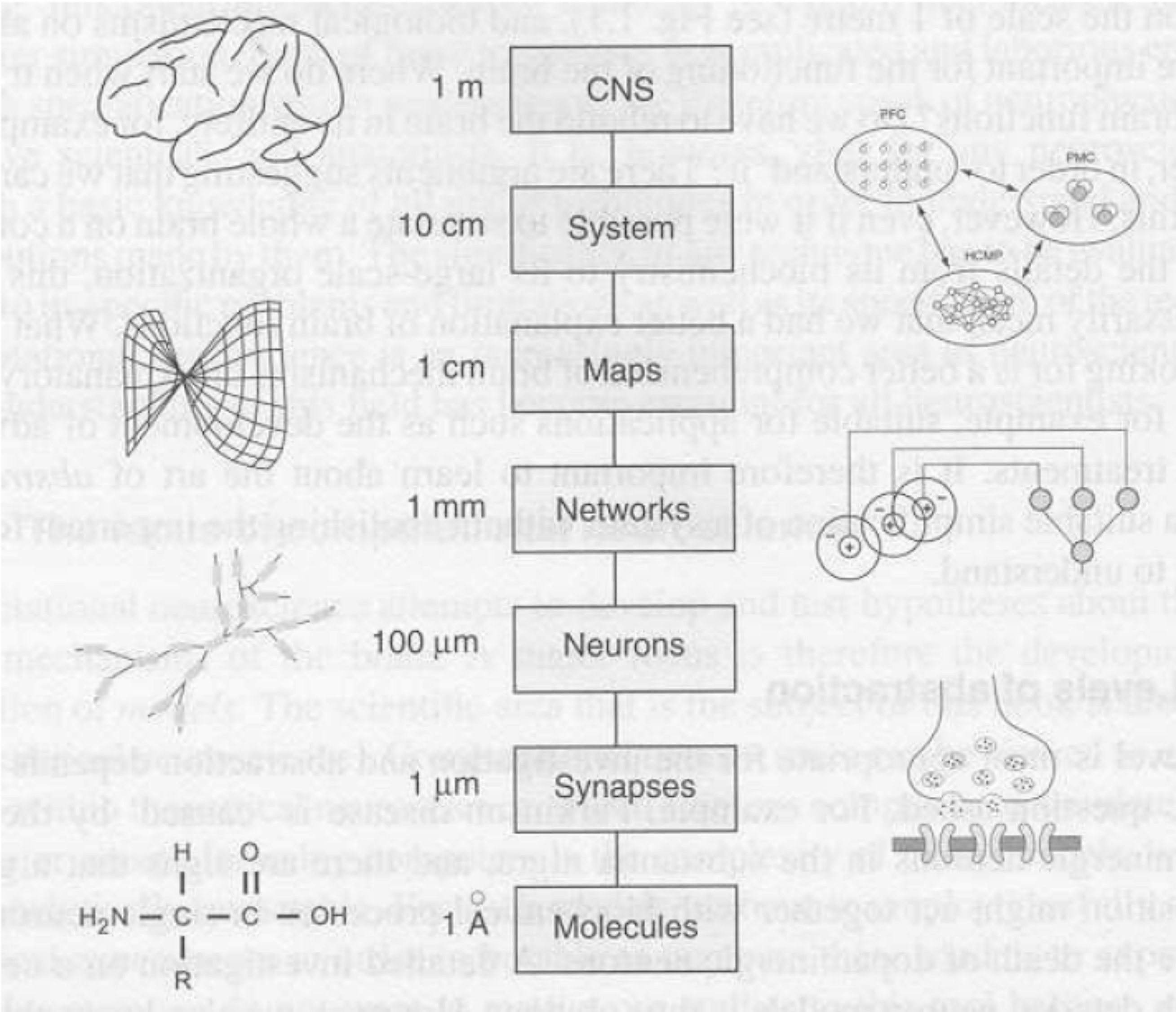
- neuroscience:
  - how does the brain work?
- theoretical neuroscience:
  - data analysis:
    - how can we extract; characterize spikes/anatomy?
  - mathematical neuroscience:
    - reductive modeling of a natural phenomenon
  - computational neuroscience:
    - the brain computes...

There are about 150 trillion cubes of this size in your brain!



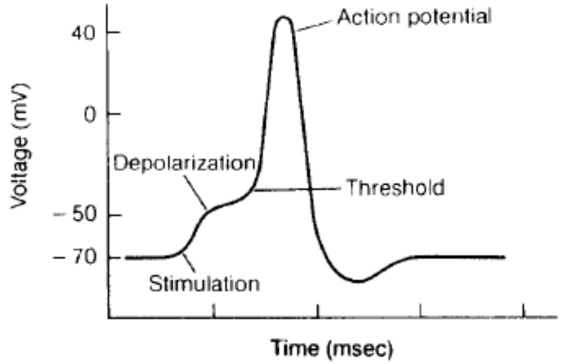
0.46μm

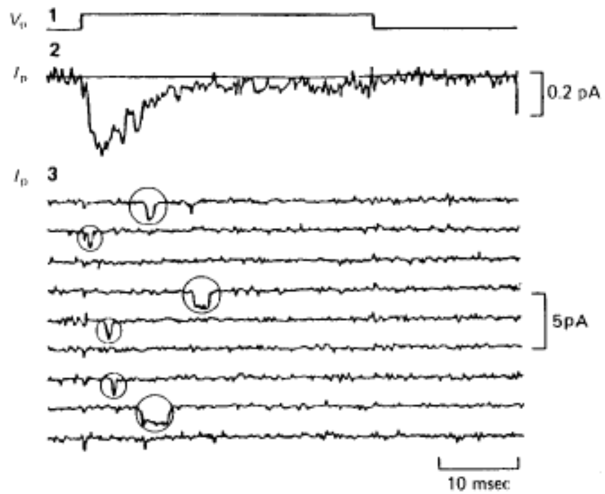
Levels of Reduction

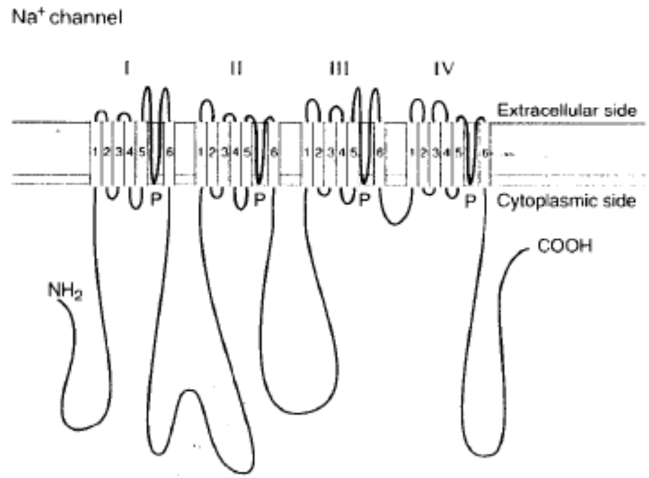


Reductive Models

- descriptive:
  - characterize as a cubic spline
- mechanistic:
  - characterise in terms of gating:
    - explanatory model of spike, from
    - descriptive model of the gate
- now: do a better job:







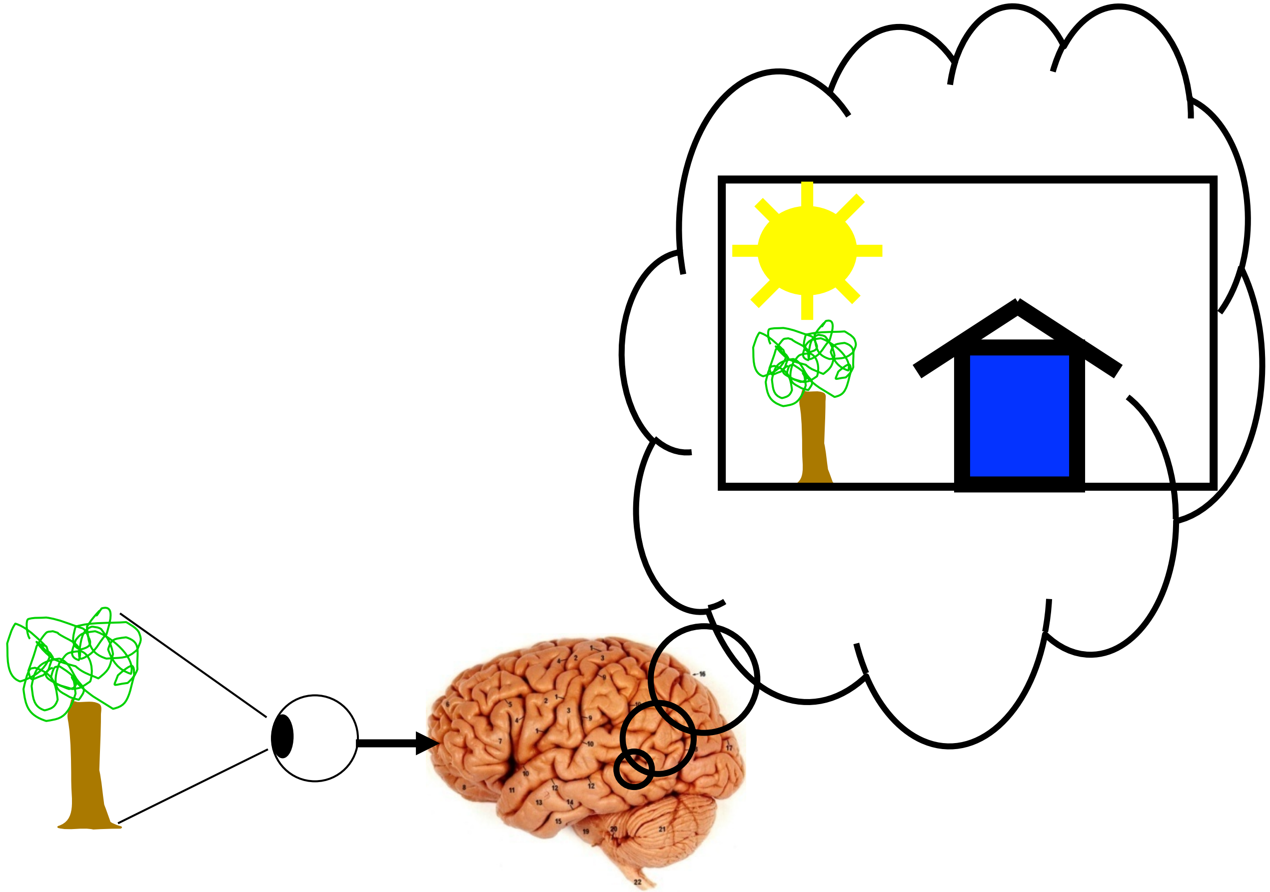
Marrian Analysis

- interpretive patina around reductive model
  - computation
    - goal; intent
    - logic of the strategy
  - algorithm
    - effective procedure for realizing computation
    - representations (coding)
  - implementation
    - neural substrate

Example #1: memory.

the problem:

recall events, typically based on partial information.



Example #1: memory.

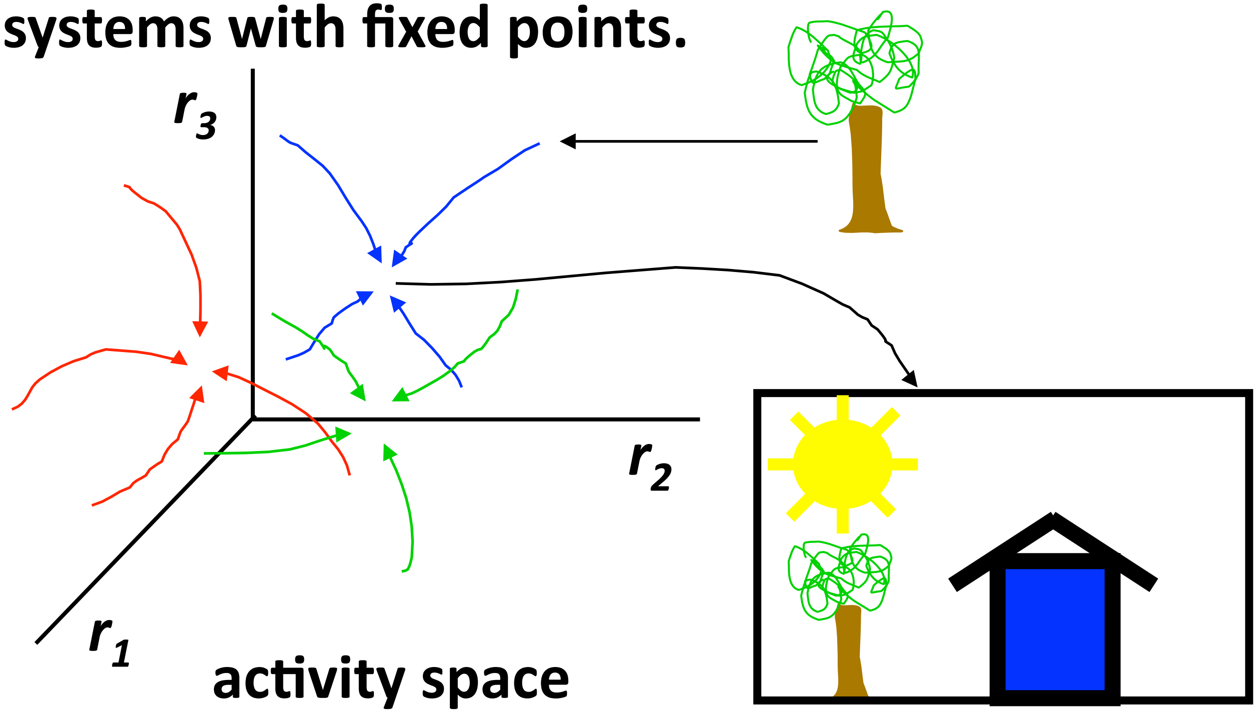
the problem:

recall events, typically based on partial information.

associative or content-addressable memory.

an algorithm:

dynamical systems with fixed points.



Example #1: memory.

the problem:

recall events, typically based on partial information.

associative or content-addressable memory.

BUT: which one to recall (depends on environment)

an algorithm:

dynamical systems with fixed points.

neural implementation:

Hopfield networks.

$$x_i = \mathrm{sign}\left(\sum_j J_{ij} x_j\right)$$

Example #2: vision.

the problem (Marr):

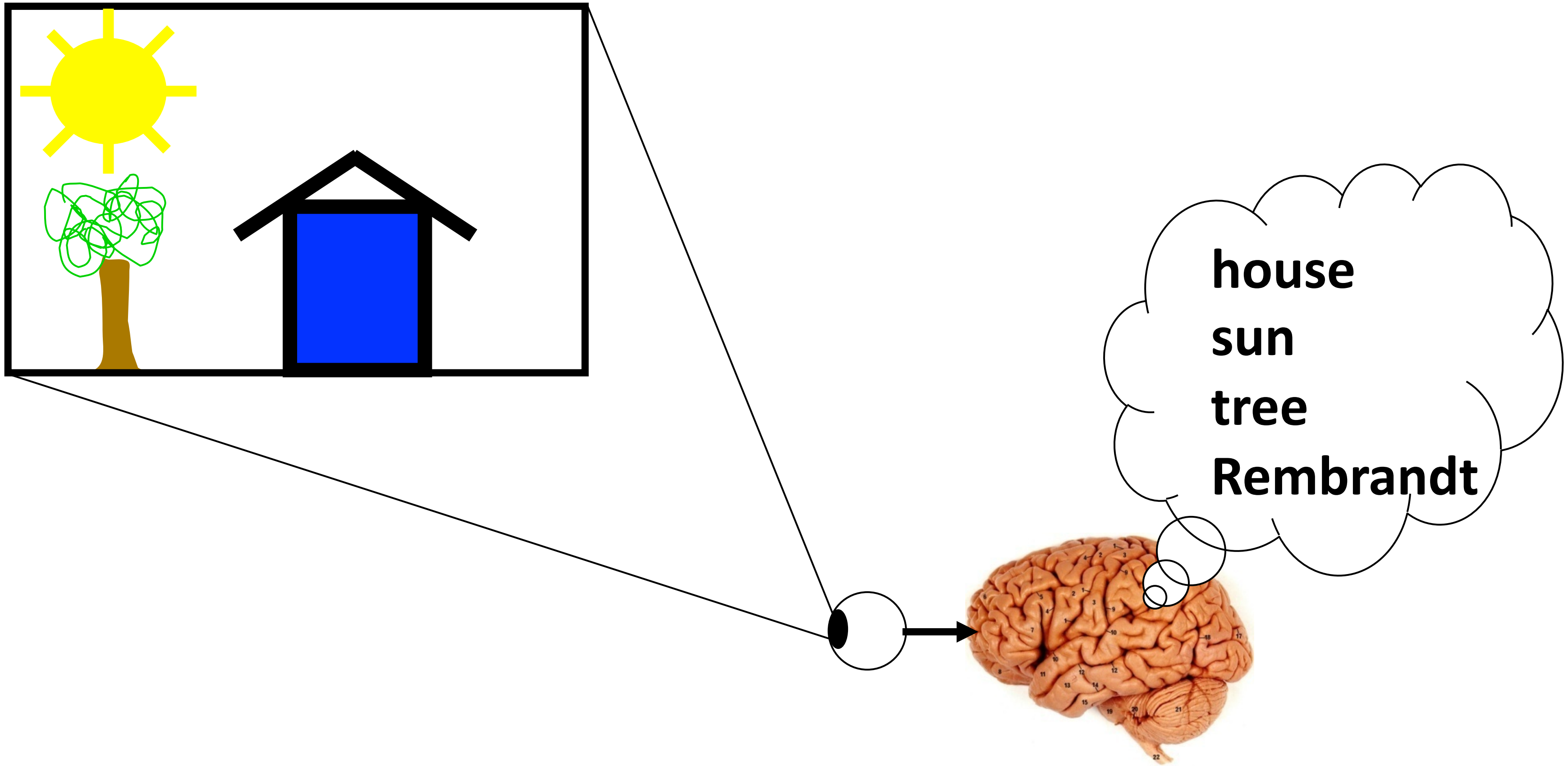
2-D image on retina →
3-D reconstruction of a visual scene.

Example #2: vision.

the problem (modern version):

2-D image on retina →

recover the latent variables.

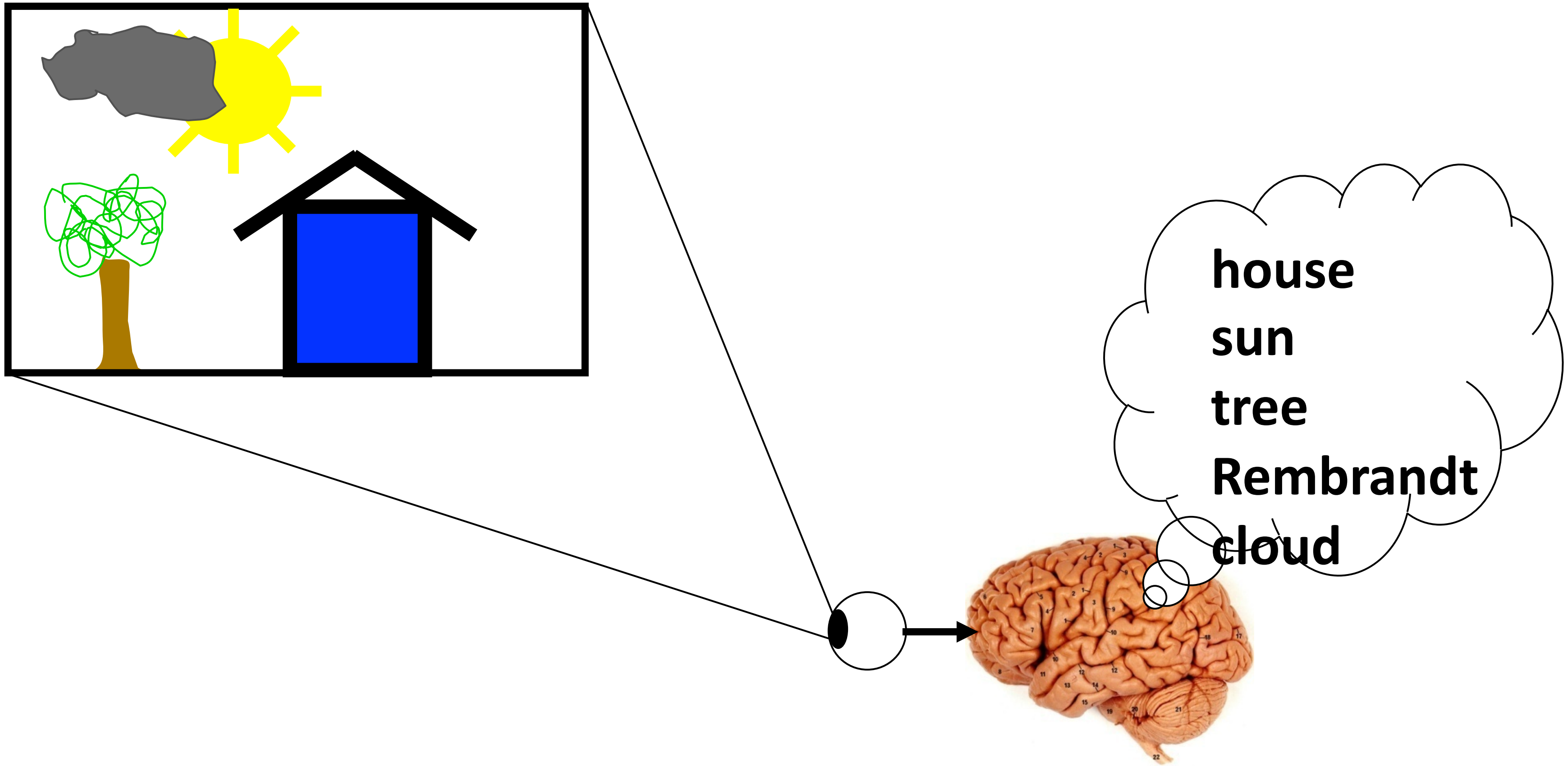


Example #2: vision.

the problem (modern version):

2-D image on retina →

recover the latent variables.



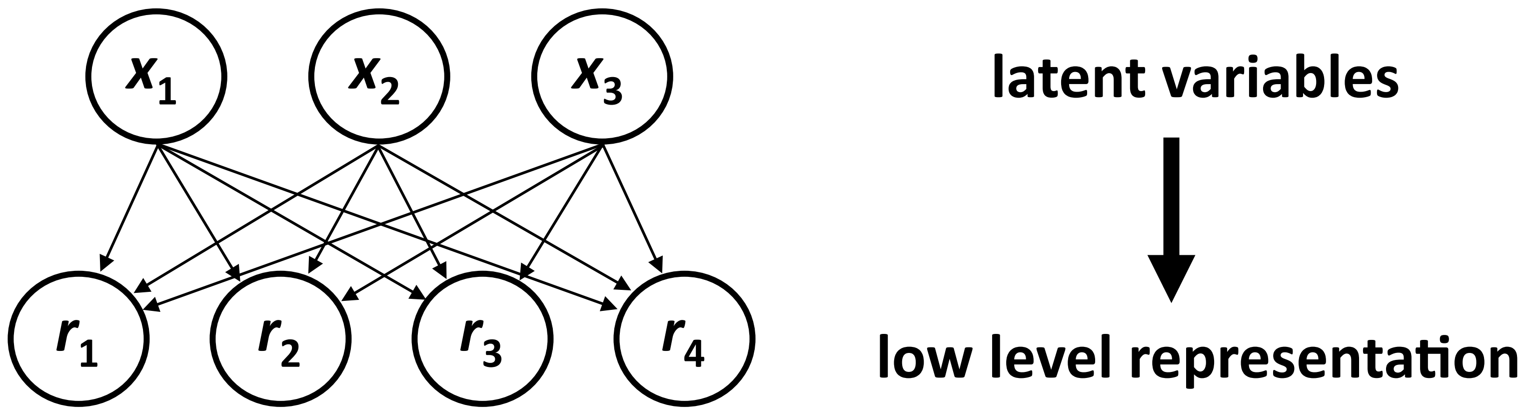
Example #2: vision.

the problem (modern version):

2-D image on retina →
reconstruction of latent variables.

an algorithm:

graphical models.



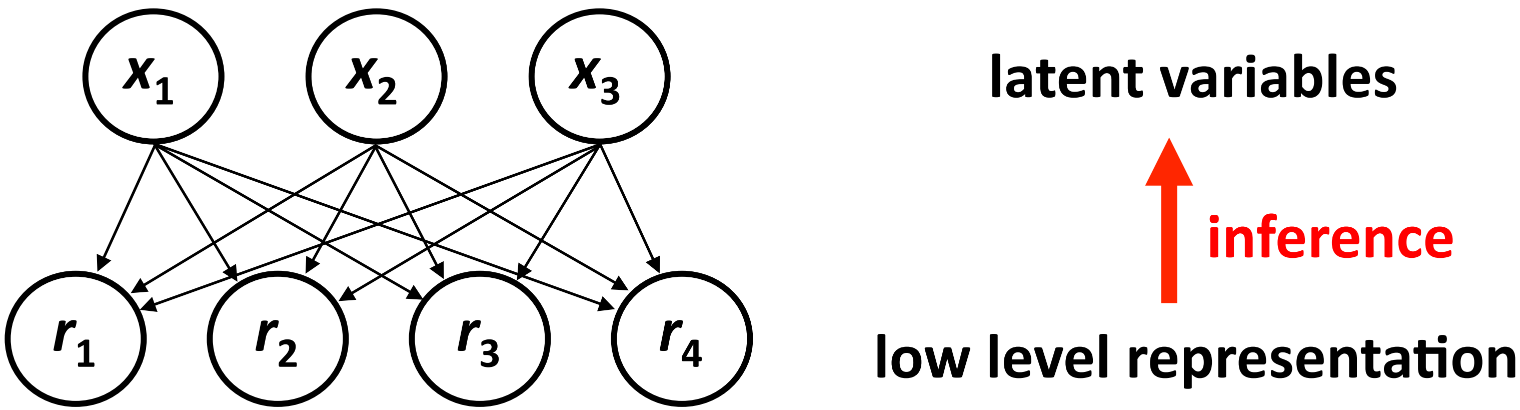
Example #2: vision.

the problem (modern version):

2-D image on retina →
reconstruction of latent variables.

an algorithm:

graphical models.



Example #2: vision.

the problem (modern version):

2-D image on retina →
reconstruction of latent variables.

an algorithm:

graphical models.

implementation in networks of neurons:

little clue.

Comment #1:

the problem:
the algorithm:
neural implementation:

Comment #1:

the problem:	**easier**
the algorithm:	**harder**
neural implementation:	**harder**

often ignored!!!

Comment #1:

the problem:	**easier**
the algorithm:	**harder**
neural implementation:	**harder**

A common approach:

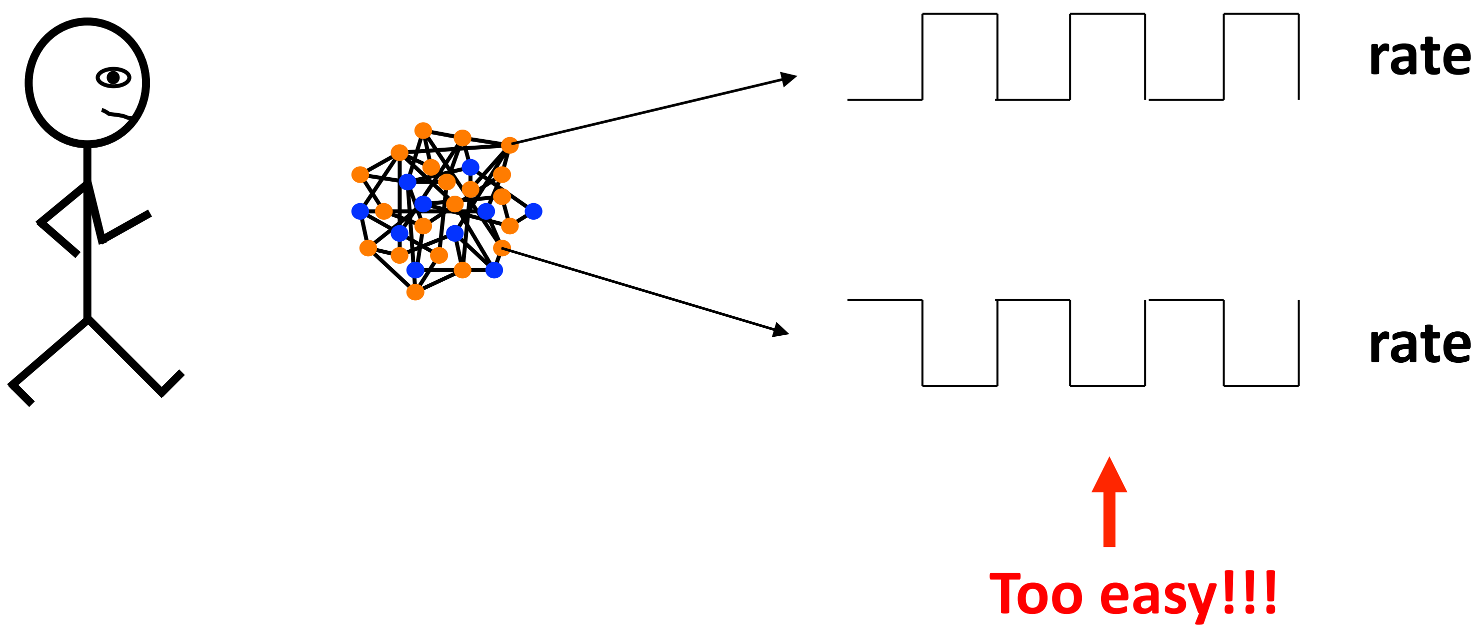
Experimental observation → model

Usually very underconstrained!!!!

Comment #1:

the problem:	easier
the algorithm:	harder
neural implementation:	harder

Example *i*: CPGs (central pattern generators)



Comment #1:

the problem:	**easier**
the algorithm:	**harder**
neural implementation:	**harder**

Example *ii*: single cell modeling

$$C\, dV/dt = -g_L(V - V_L) - n^4(V - V_K) \ldots$$

$$dn/dt = \ldots$$

…

lots and lots of parameters … which ones should you use?

Comment #1:

the problem:	**easier**
the algorithm:	**harder**
neural implementation:	**harder**

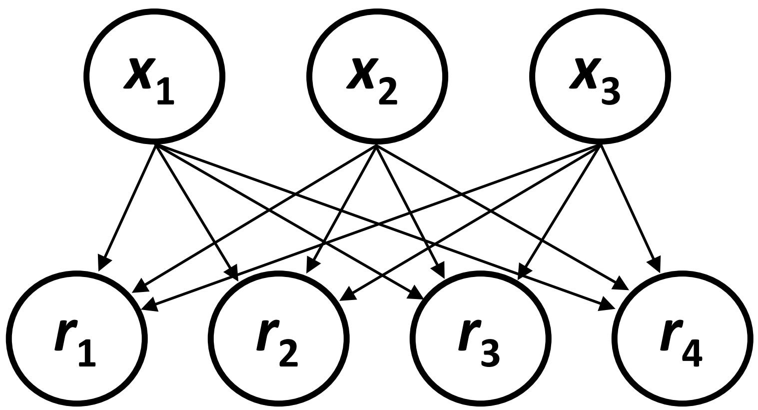
Example *iii*: network modeling

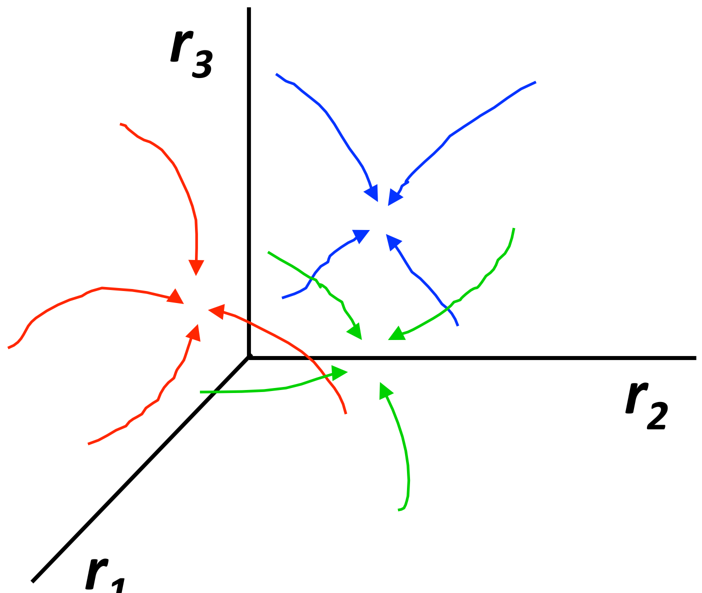
lots and lots of parameters × thousands

Comment #2:

the problem:	**easier**
the algorithm:	**harder**
neural implementation:	**harder**

You need to know a lot of maths





activity space

Marrian Conditioning

prediction: of important events
control: in the light of those predictions

- Ethology
  - optimality
  - appropriateness
- Psychology
  - classical/operant conditioning
- Computation
  - dynamic progr.
  - Kalman filtering
- Algorithm
  - TD/delta rules
  - simple weights
- Neurobiology

neuromodulators; midbrain; sub-cortical; cortical structures

Comment #3:

the problem:	easier
the algorithm:	harder
neural implementation:	harder

<u>This is a good goal, but it's hard to do in practice.</u>

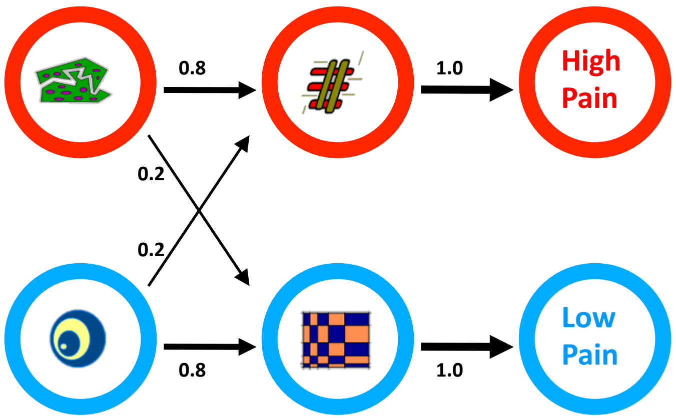
Our actual bread and butter:

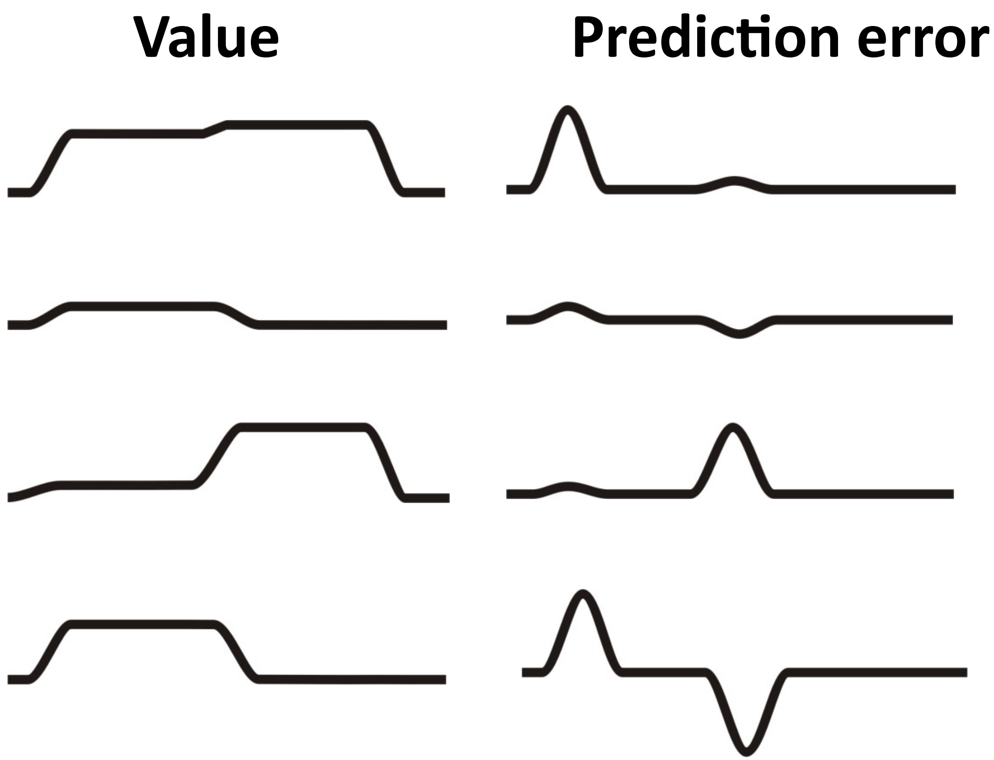
1. Explaining observations (mathematically)
2. Using sophisticated analysis to design simple experiments that test hypotheses.

two experiments: RL and visual salience

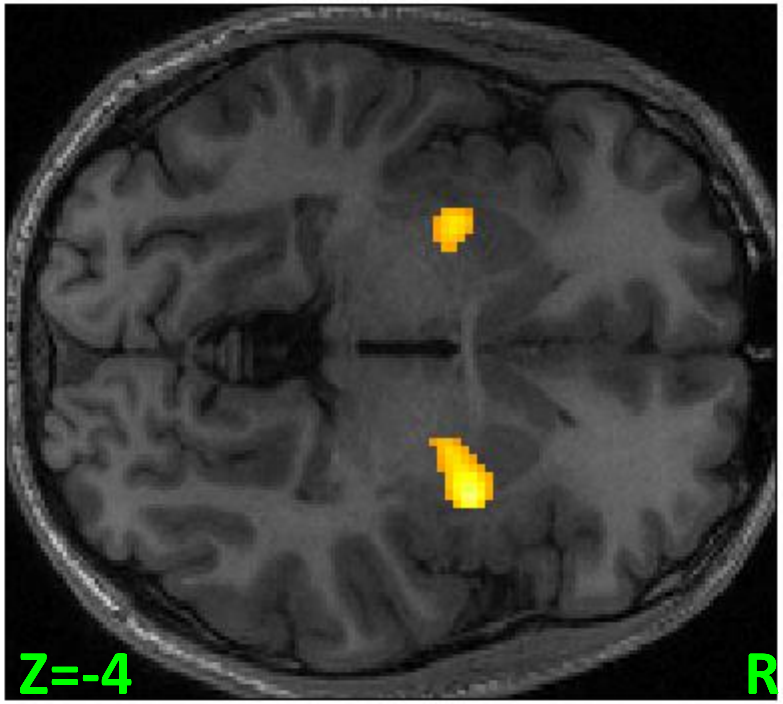
Temporal Difference Prediction Error

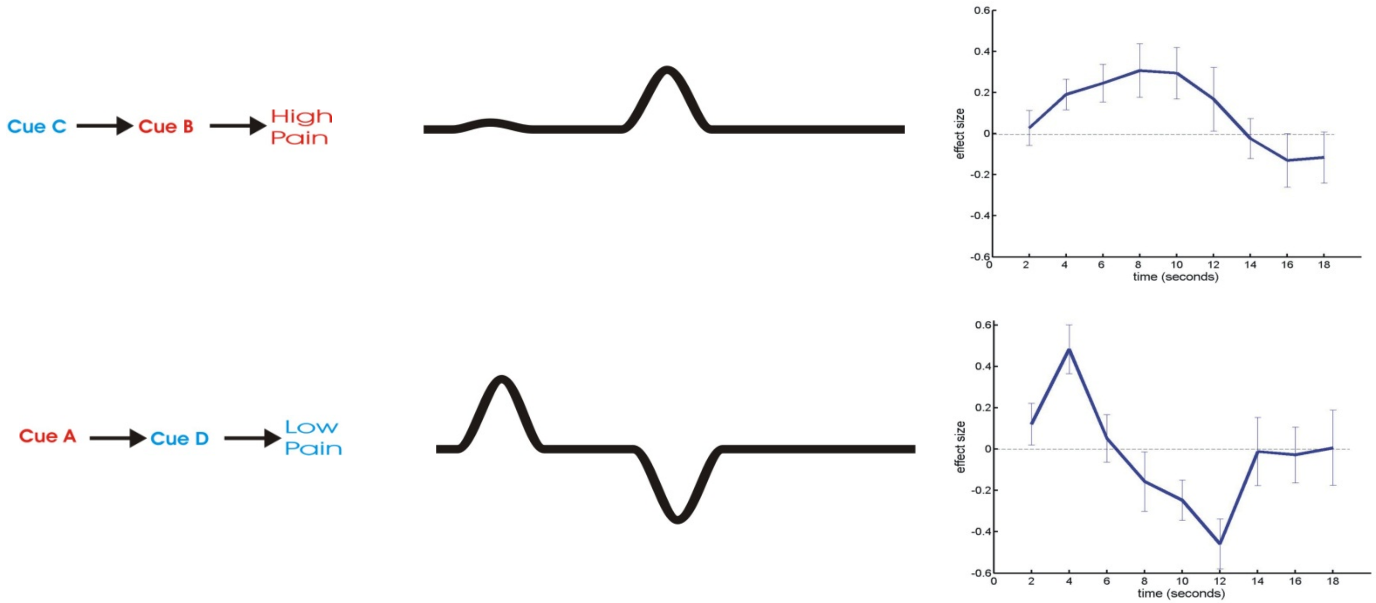
TD error $\delta(t) = r(t) + V(t+1) - V(t)$





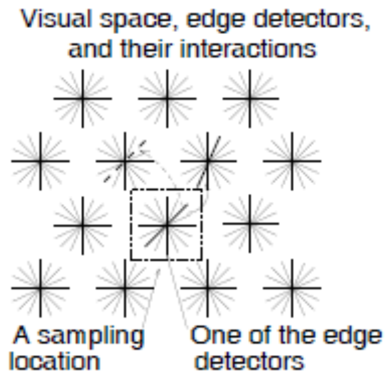
TD prediction error:
ventral striatum

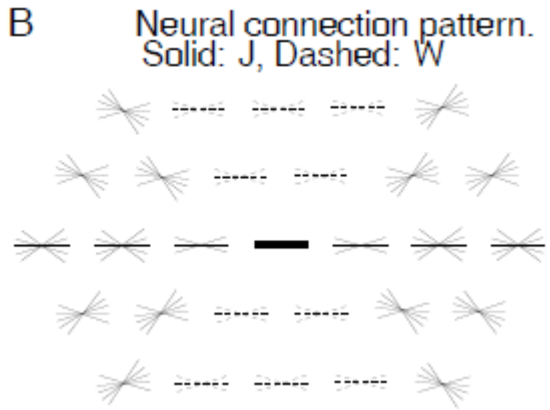




Visual Salience (Li/Zhaoping)

- problem:
  - segmentation without classification
- algorithm:
  - interacting neural elements with a connection field
- implementation:
  - horizontal connections in V1!

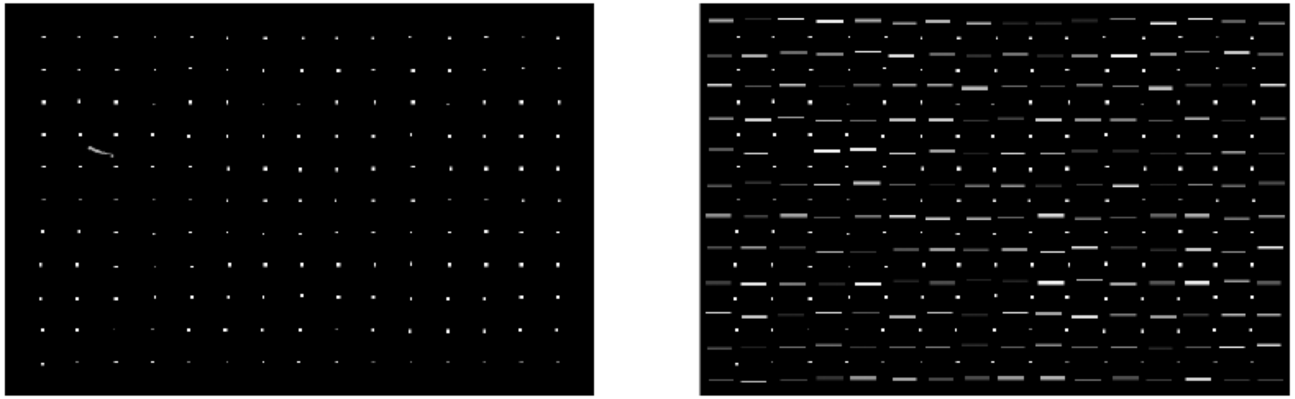




Monocular Popout

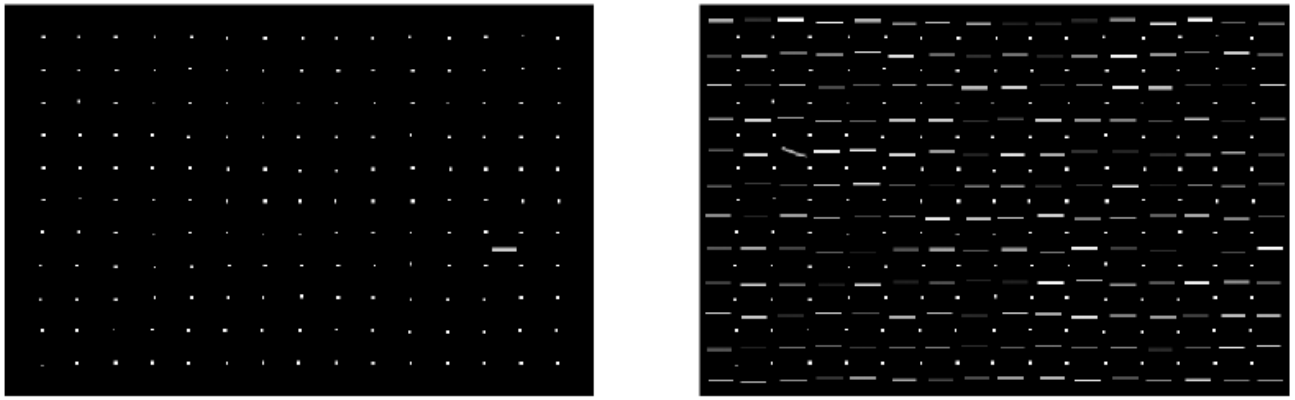
A A dichoptic congruent stimulus in Experiment 1

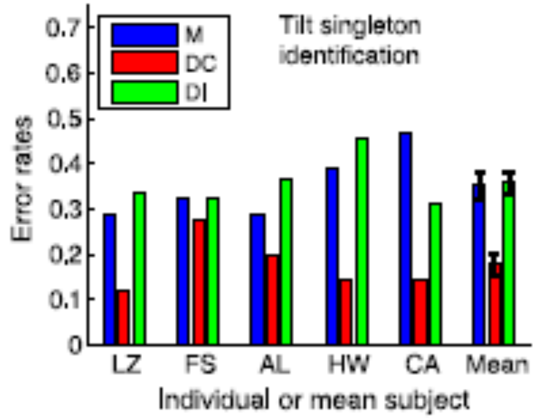
Left eye input Right eye input



B A dichoptic incongruent stimulus in Experiment 1

Left eye input Right eye input





Comment #4:

the problem:	**easier**
the algorithm:	**harder**
neural implementation:	**harder**

these are linked!!!

some algorithms are easy to implement on a computer but hard in a brain, and vice-versa.

Comment #4:

hard for a brain, easy for a computer:

A^{-1}

$z=x+y$

$\int dx \ldots$

optimal draughts

easy for a brain, hard for a computer:

speech recognition

go

inference from diverse, weak, hierarchical statistical constraints

Comment #4:

the problem:	easier
the algorithm:	harder
neural implementation:	harder

these are linked!!!

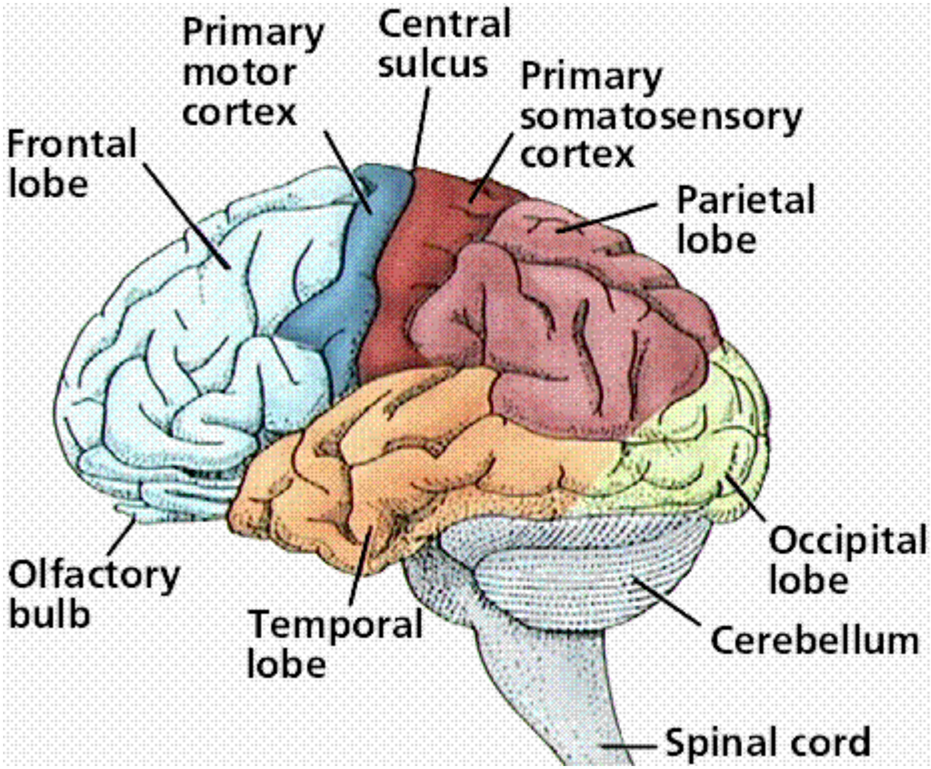
some algorithms are easy to implement on a computer but hard in a brain, and vice-versa.

we should be looking for the vice-versa ones.

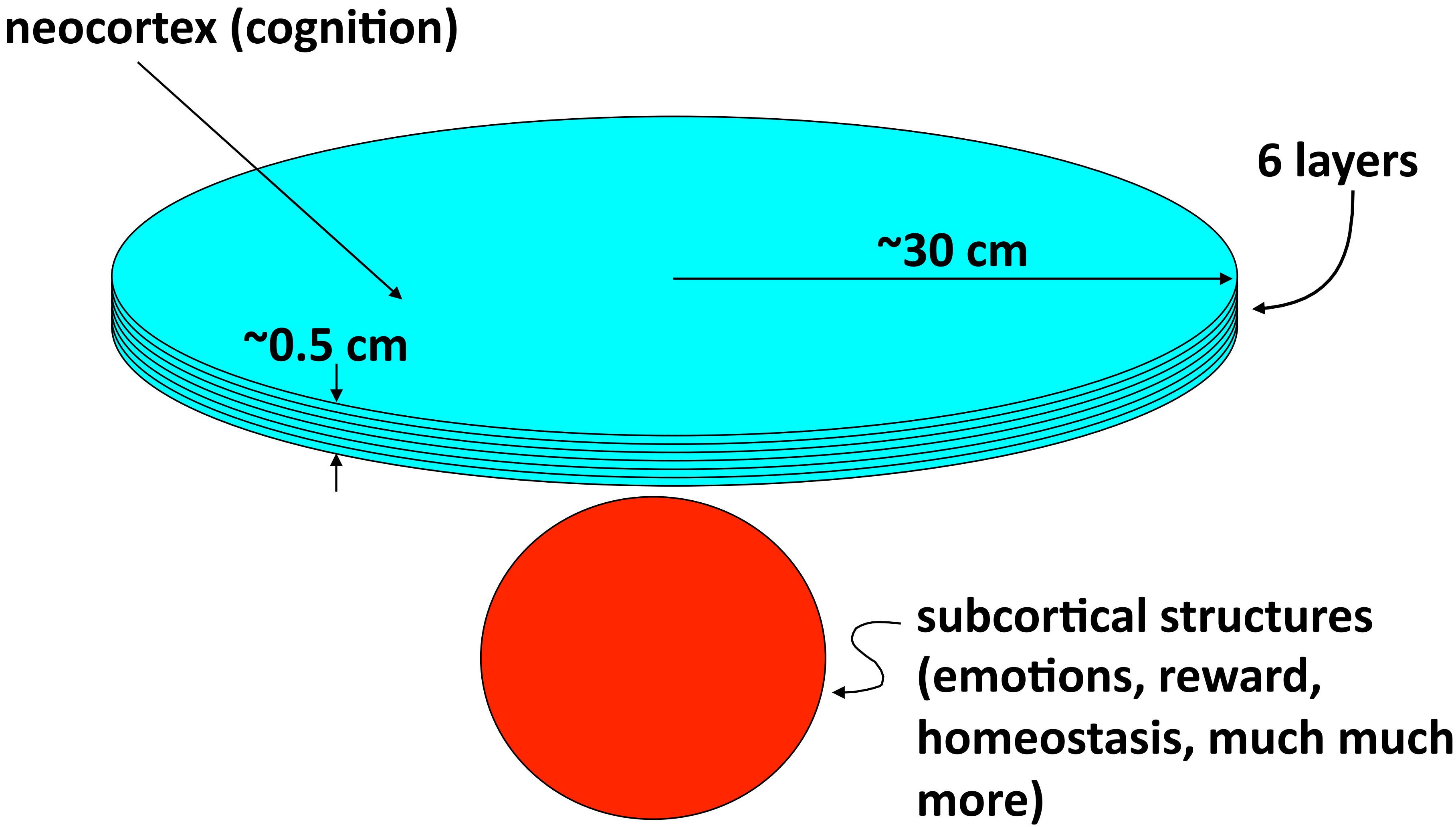
it can be hard to tell which is which.

Basic facts about the brain

Your brain

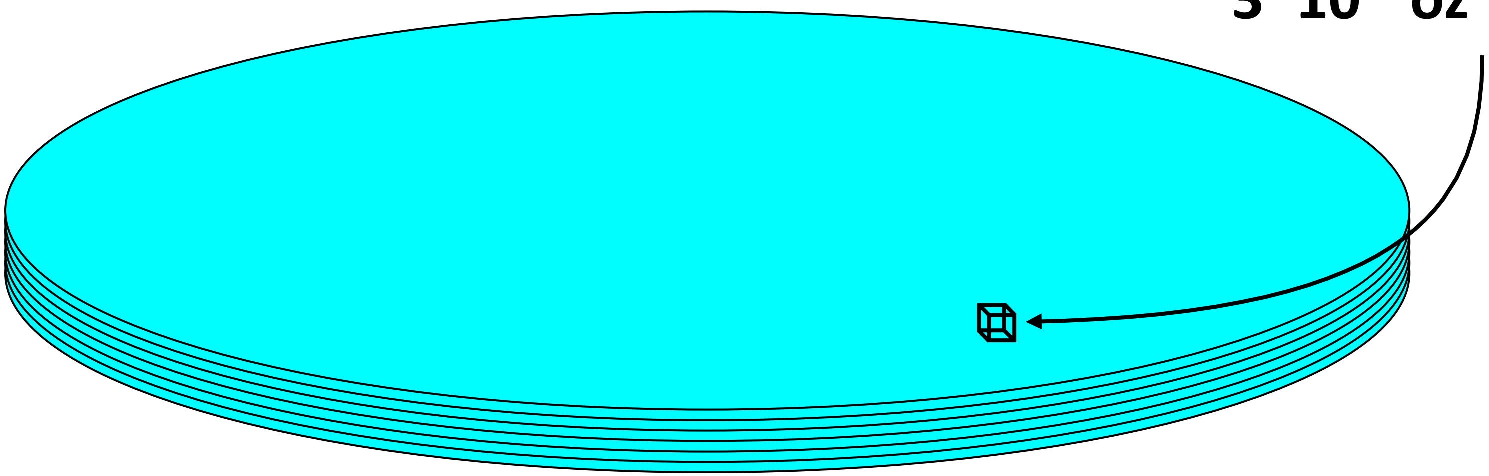

Primary motor cortex
Central sulcus
Primary somatosensory cortex
Frontal lobe
Parietal lobe
Occipital lobe
Olfactory bulb
Temporal lobe
Cerebellum
Spinal cord

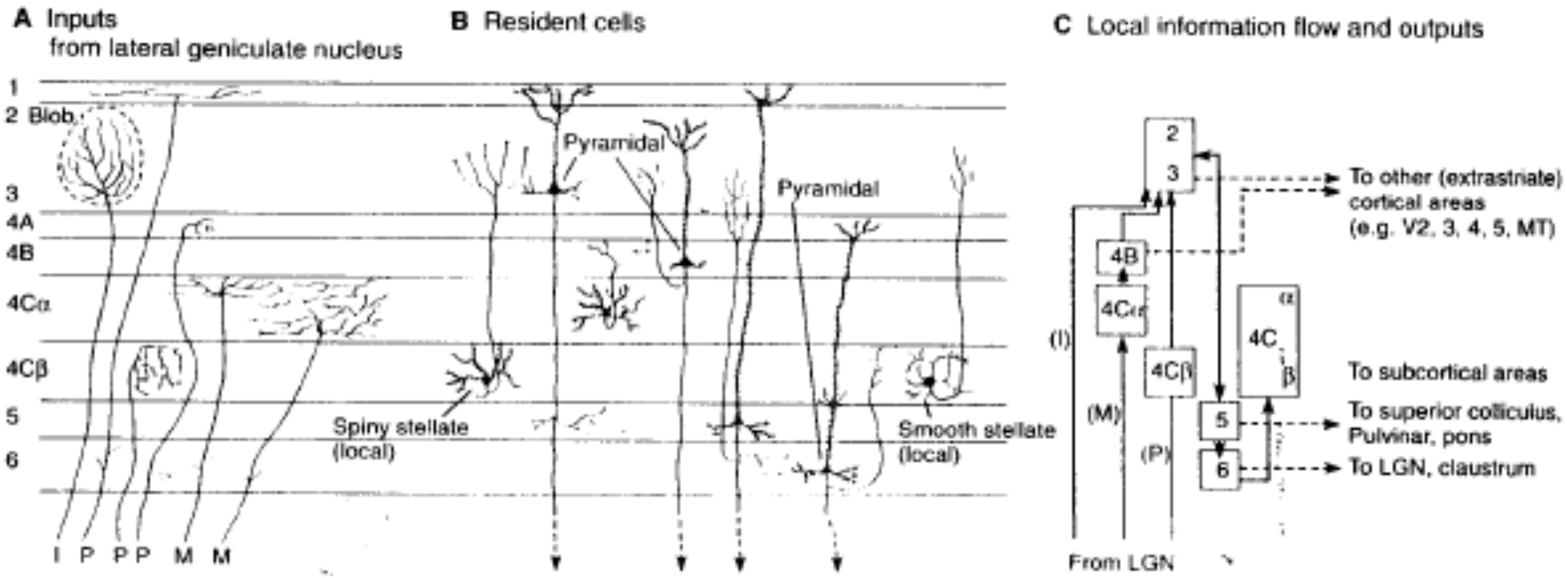
Your cortex unfolded



Your cortex unfolded

1 cubic millimeter,
~3*10^{-5} oz





1 mm^3 of cortex:

50,000 neurons
10000 connections/neuron
(=> 500 million connections)
4 km of axons

1 mm^3 of cortex:

50,000 neurons
10000 connections/neuron
(=> 500 million connections)
4 km of axons

1 mm^2 of a CPU:

1 million transistors
2 connections/transistor
(=> 2 million connections)
.002 km of wire

1 mm^3 of cortex:

50,000 neurons
10000 connections/neuron
(=> 500 million connections)
4 km of axons

whole brain (2 kg):

10^{11} neurons
10^{15} connections
8 million km of axons

1 mm^2 of a CPU:

1 million transistors
2 connections/transistor
(=> 2 million connections)
.002 km of wire

whole CPU:

10^9 transistors
$2*10^9$ connections
2 km of wire

1 mm^3 of cortex:

50,000 neurons
10000 connections/neuron
(=> 500 million connections)
4 km of axons

1 mm^2 of a CPU:

1 million transistors
2 connections/transistor
(=> 2 million connections)
.002 km of wire

whole brain (2 kg):

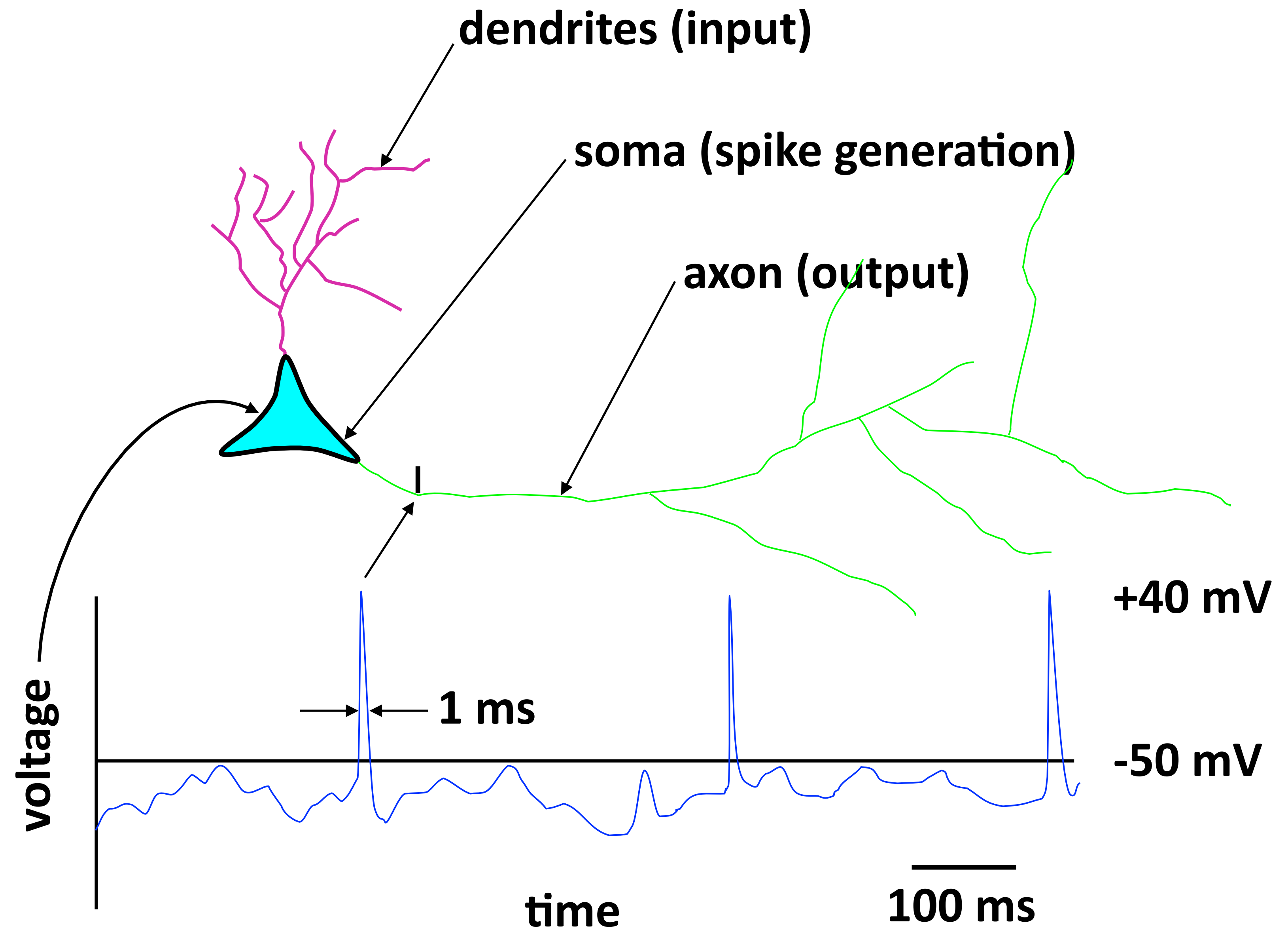
10^{11} neurons
10^{15} connections
8 million km of axons

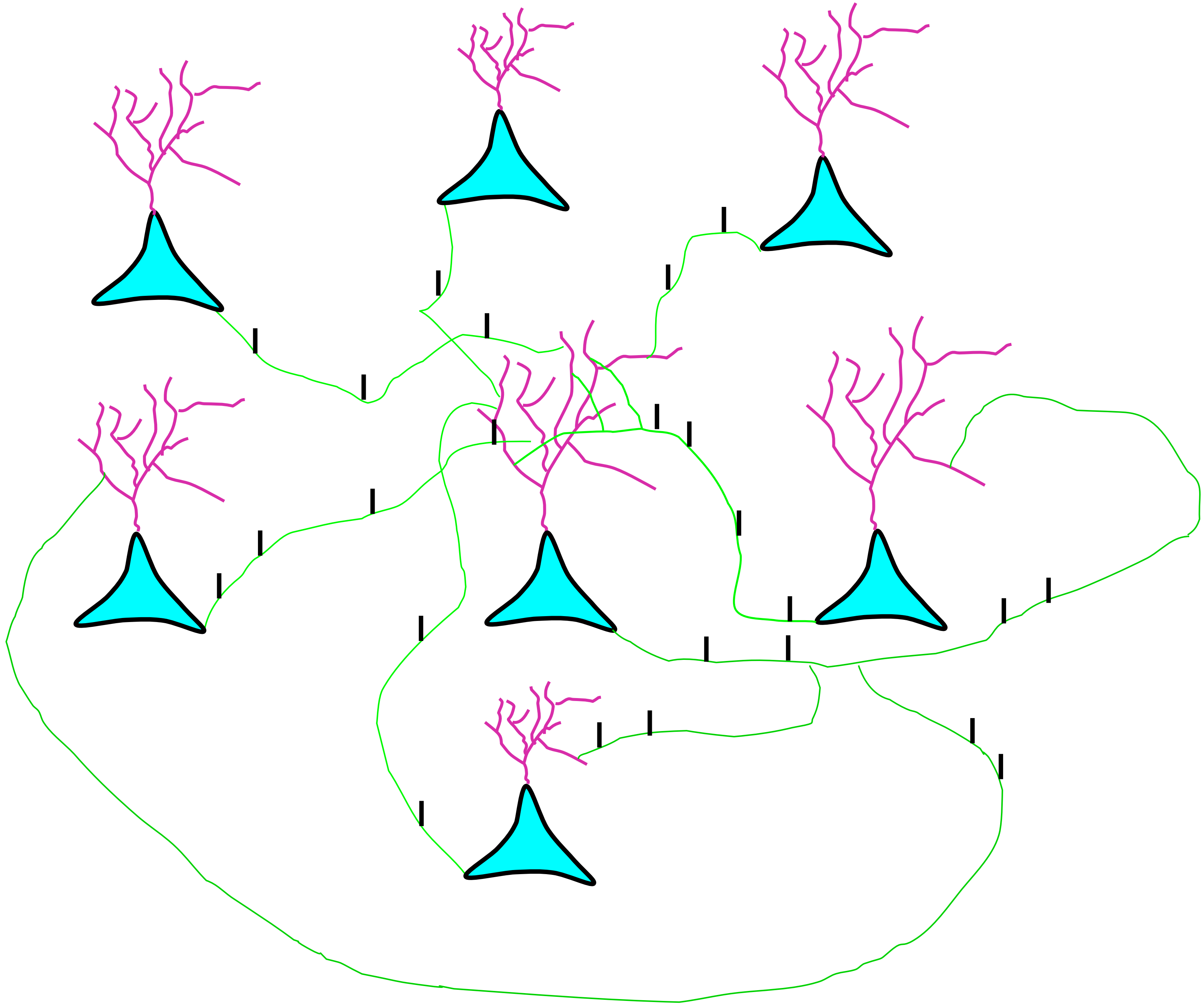
whole CPU:

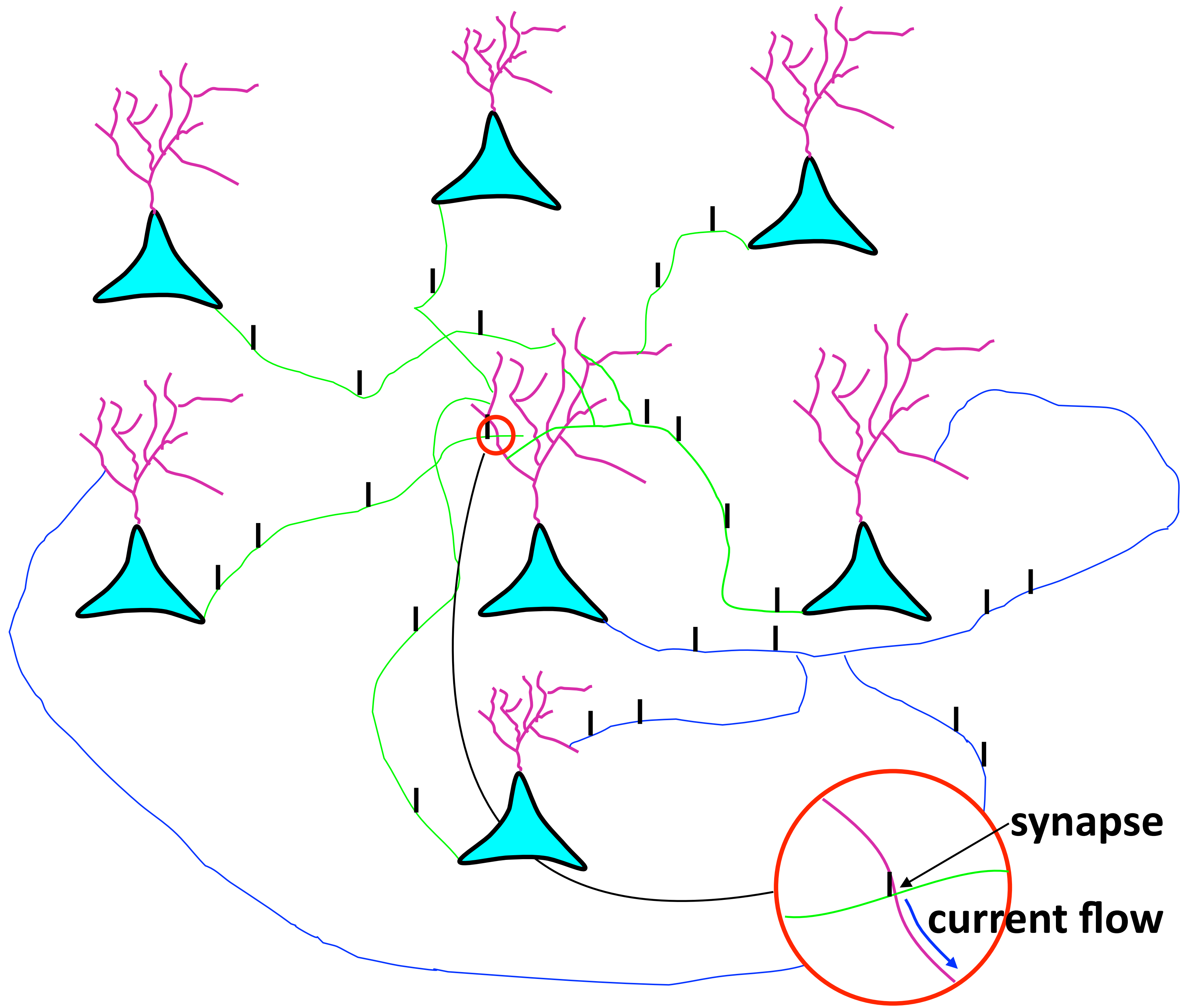
10^9 transistors
$2*10^9$ connections
2 km of wire

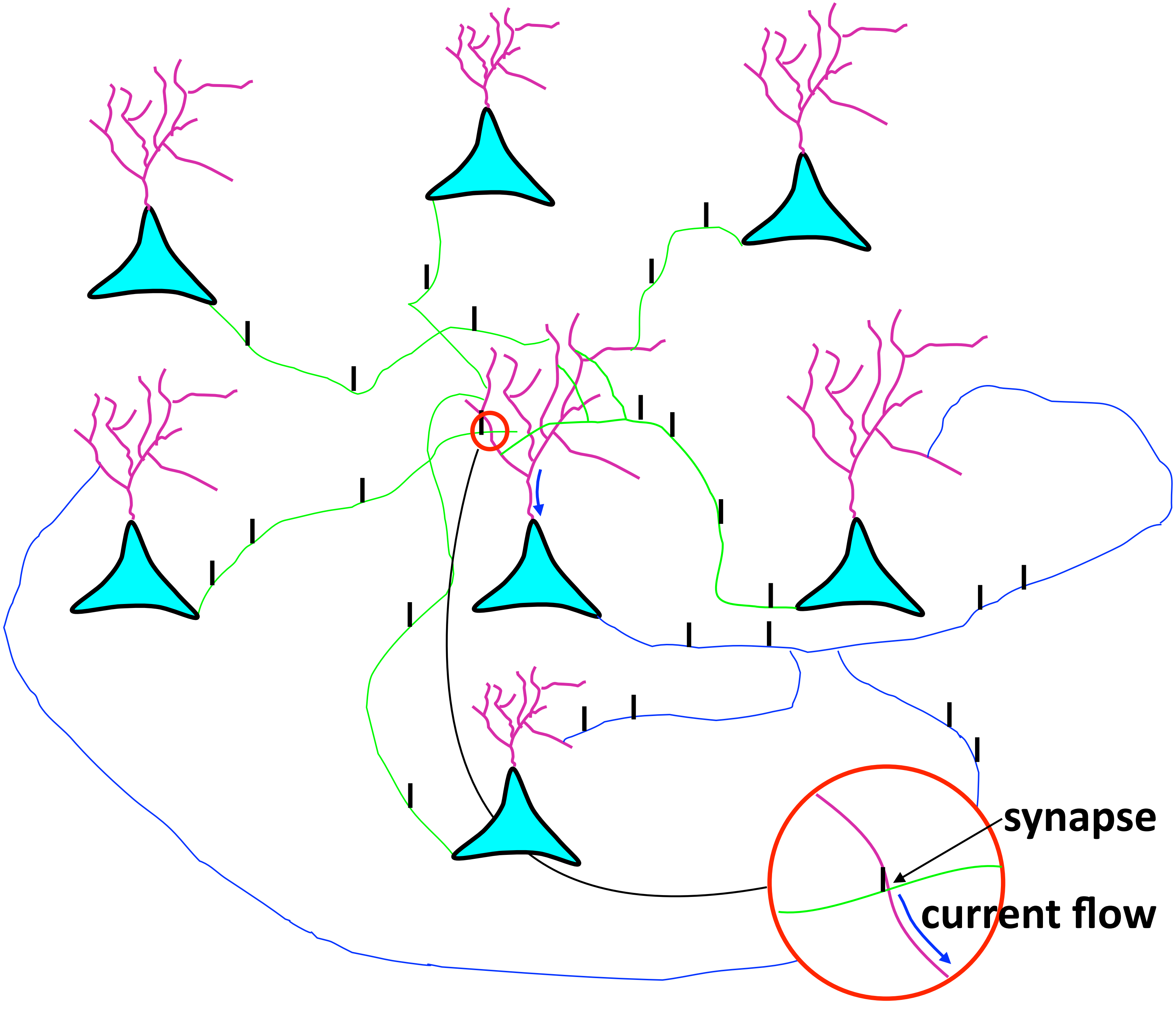
What do we know about the brain?

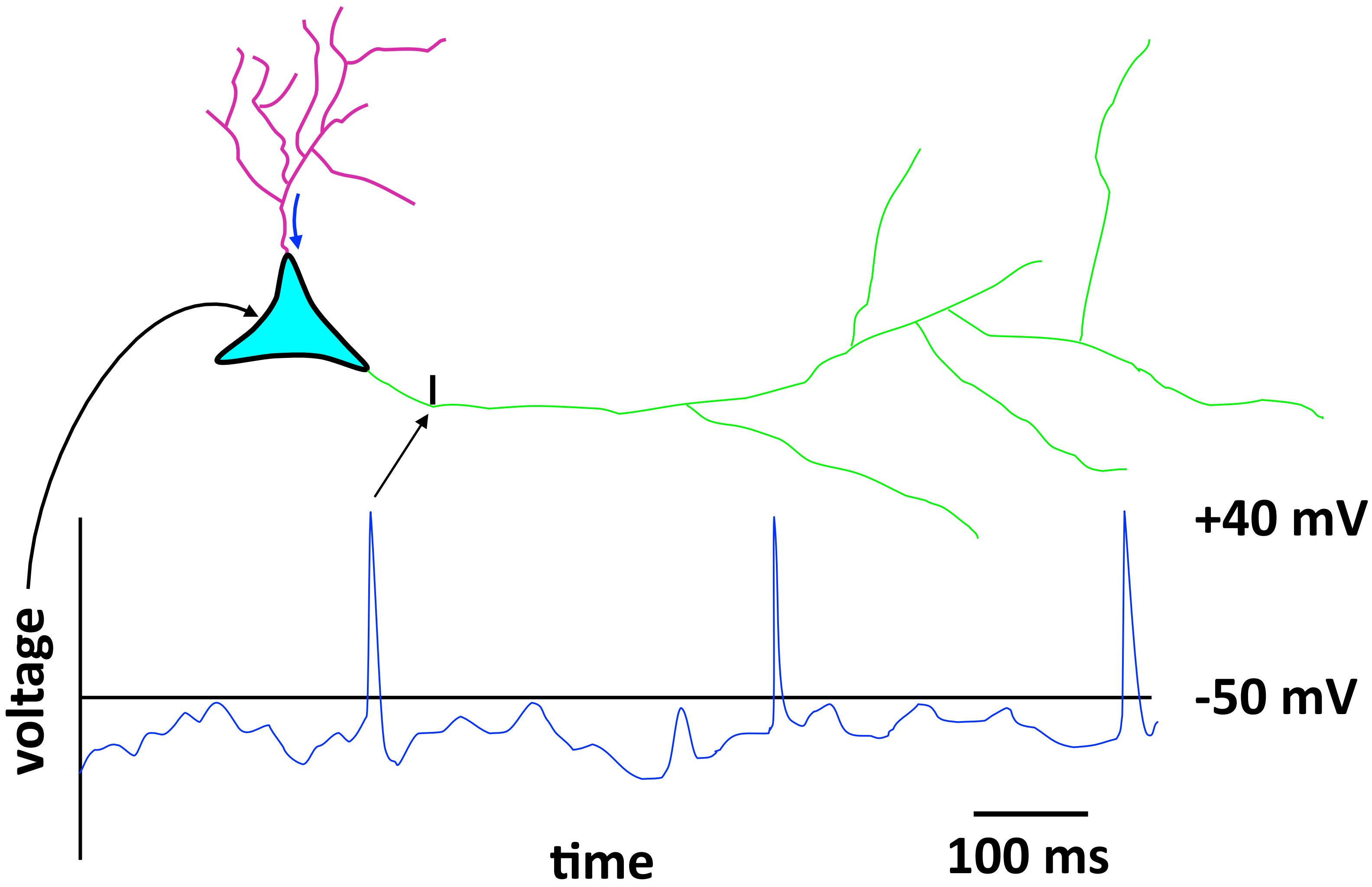
Highly biased


dendrites (input)
soma (spike generation)
axon (output)
+40 mV
1 ms
-50 mV
voltage
time
100 ms

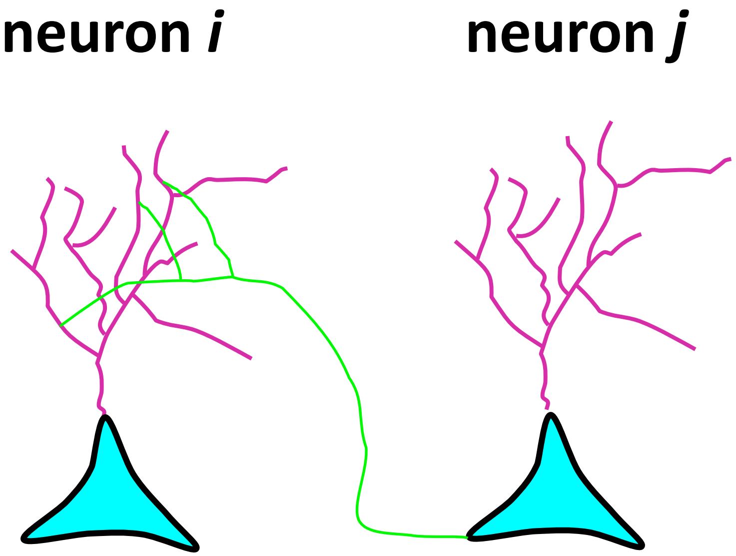



synapse
current flow

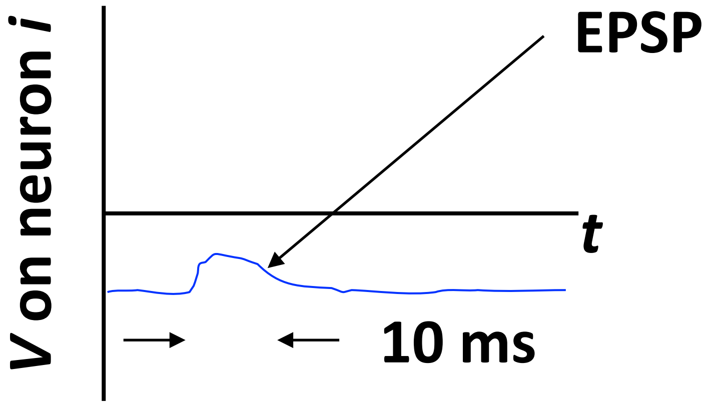

synapse
current flow

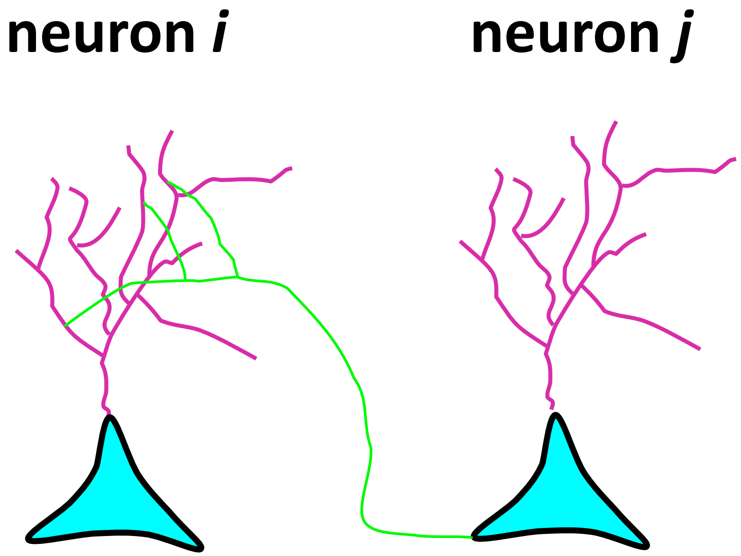


+40 mV
-50 mV
voltage
time
100 ms

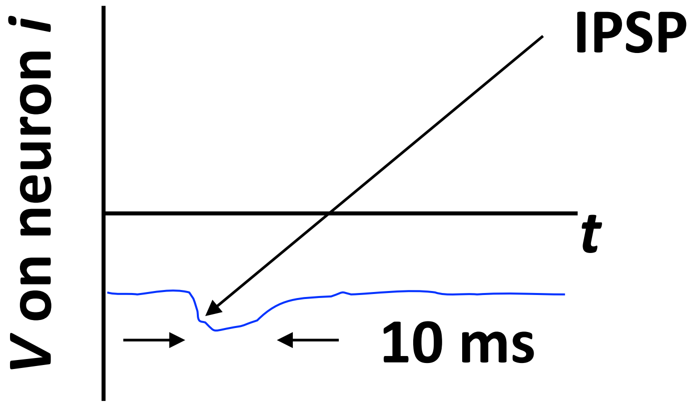


neuron j emits a spike:

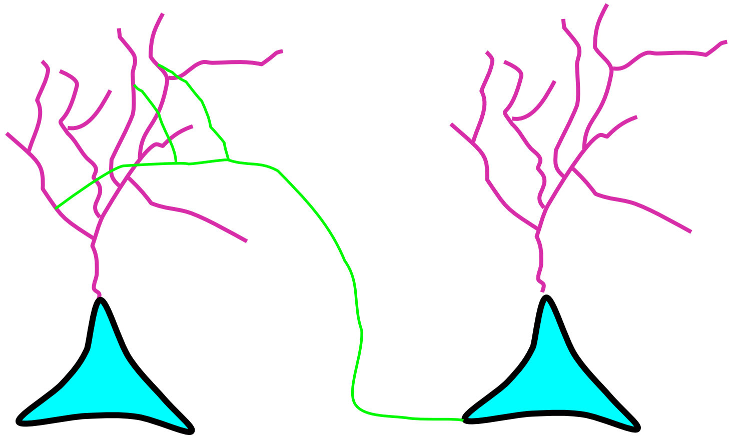




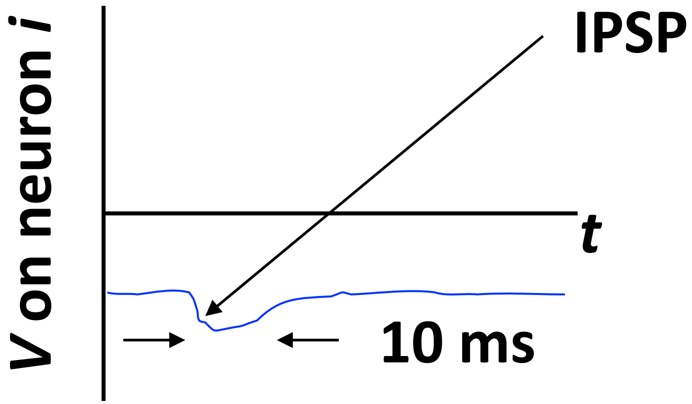
neuron *j* emits a spike:



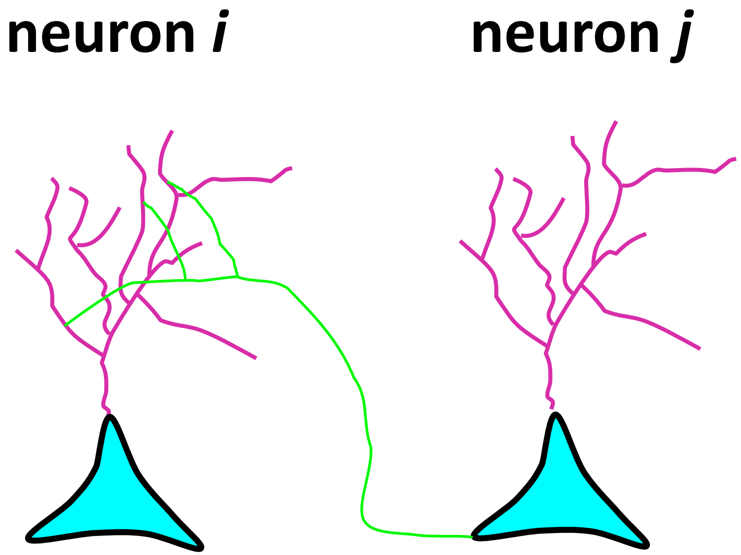
neuron i neuron j



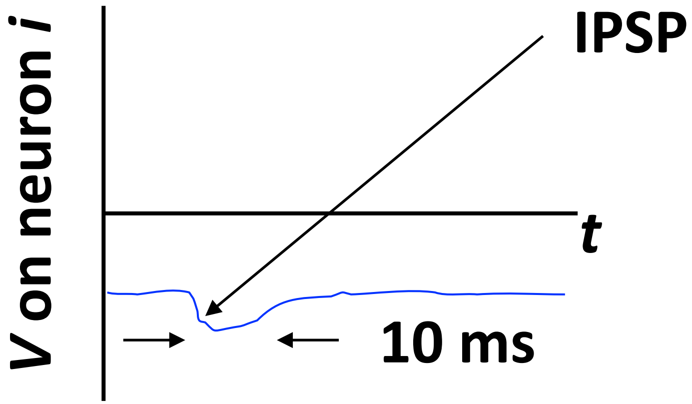
neuron j emits a spike:



amplitude = w_{ij}

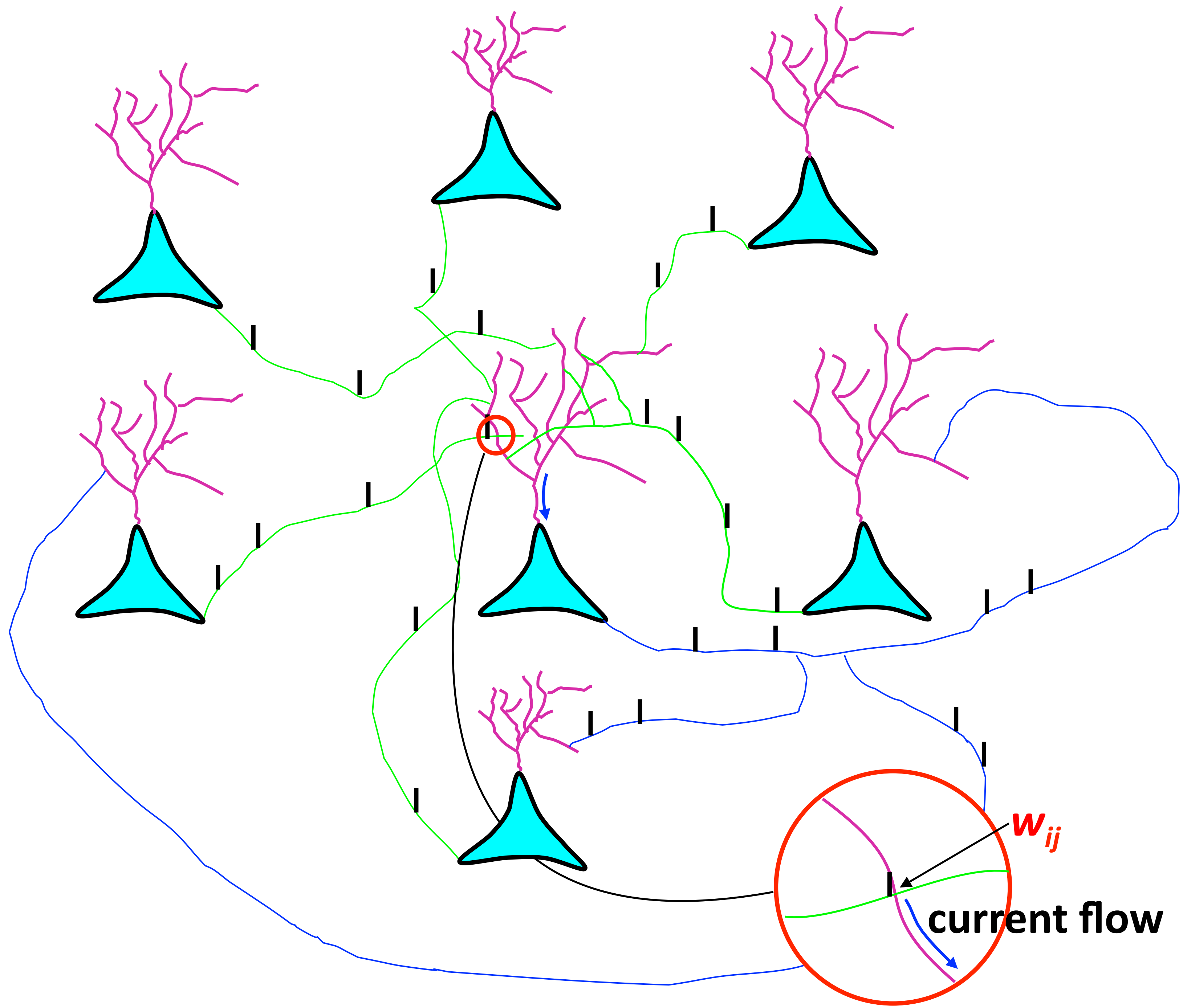


neuron *j* emits a spike:



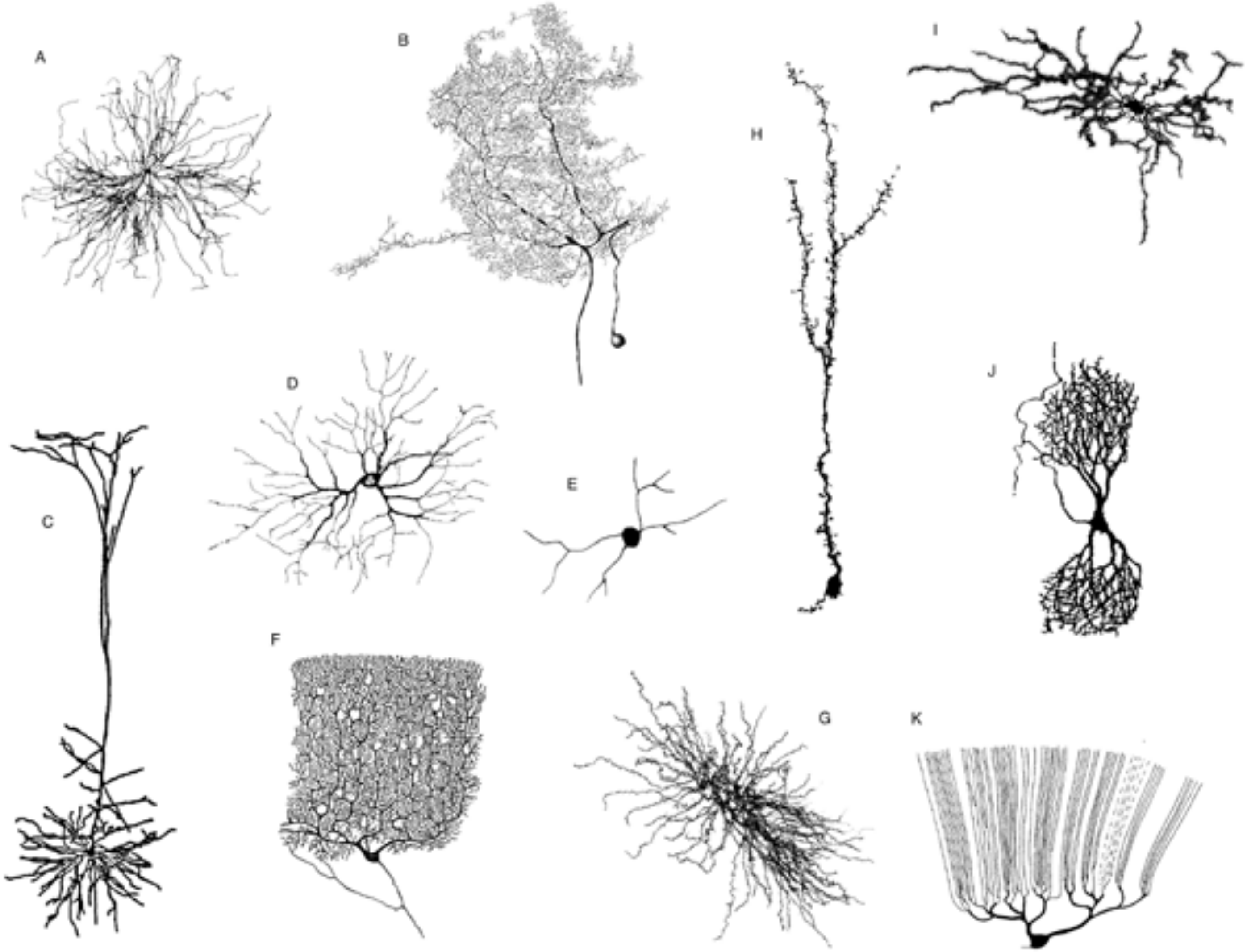
changes with learning

amplitude = w_{ij}

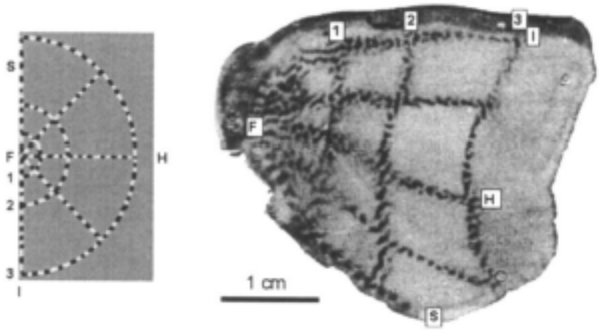


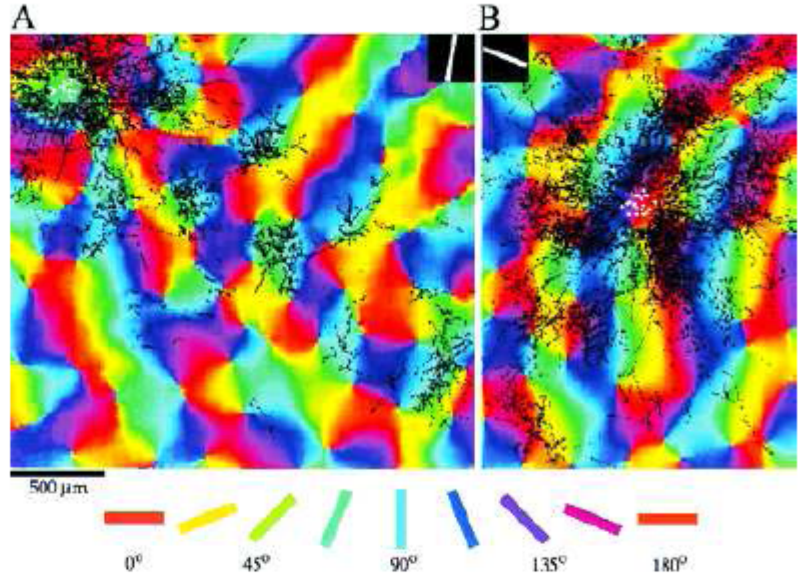
w_{ij}
current flow

Real Dendrites



a. <u>Anatomy.</u> We know a lot about what is where. But be careful about labels: neurons in motor cortex sometimes respond to color.

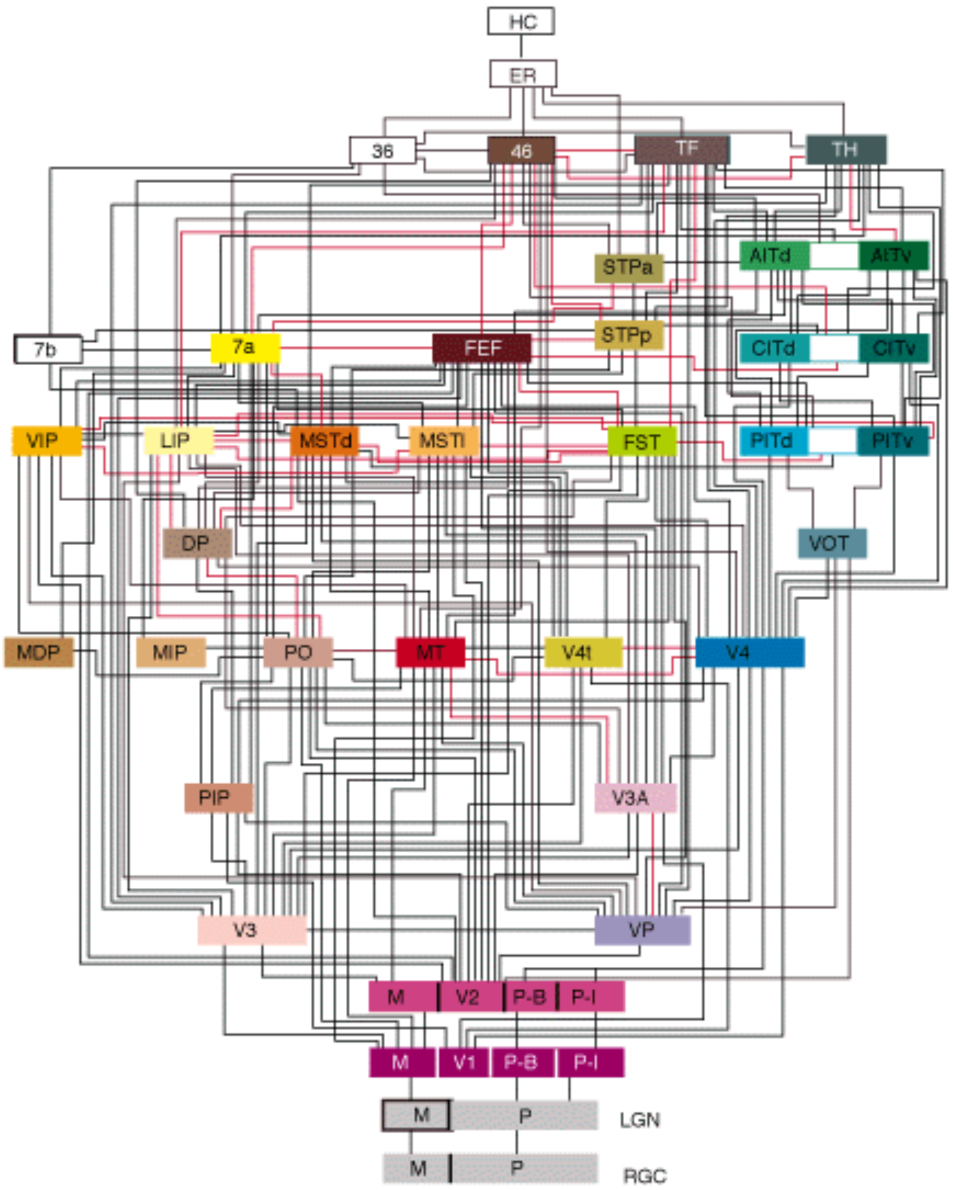


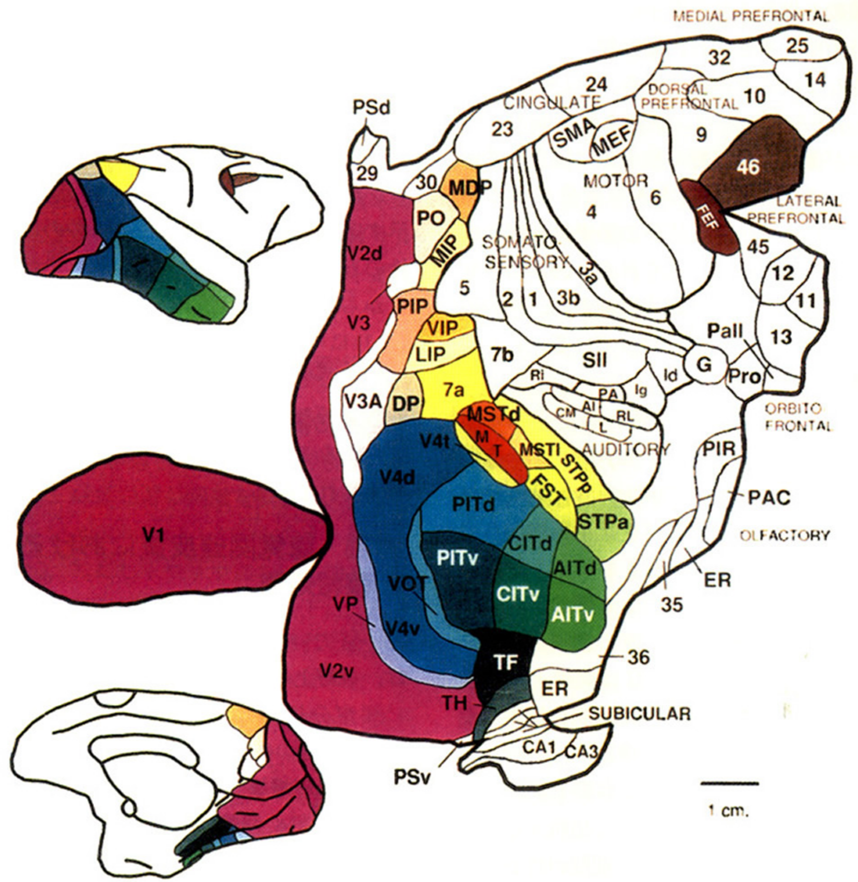


a. <u>Anatomy.</u> **We know a lot about what is where. But be careful about labels: neurons in motor cortex sometimes respond to color.**

<u>Connectivity.</u> **We know (more or less) which area is connected to which.**

The van Essen diagram



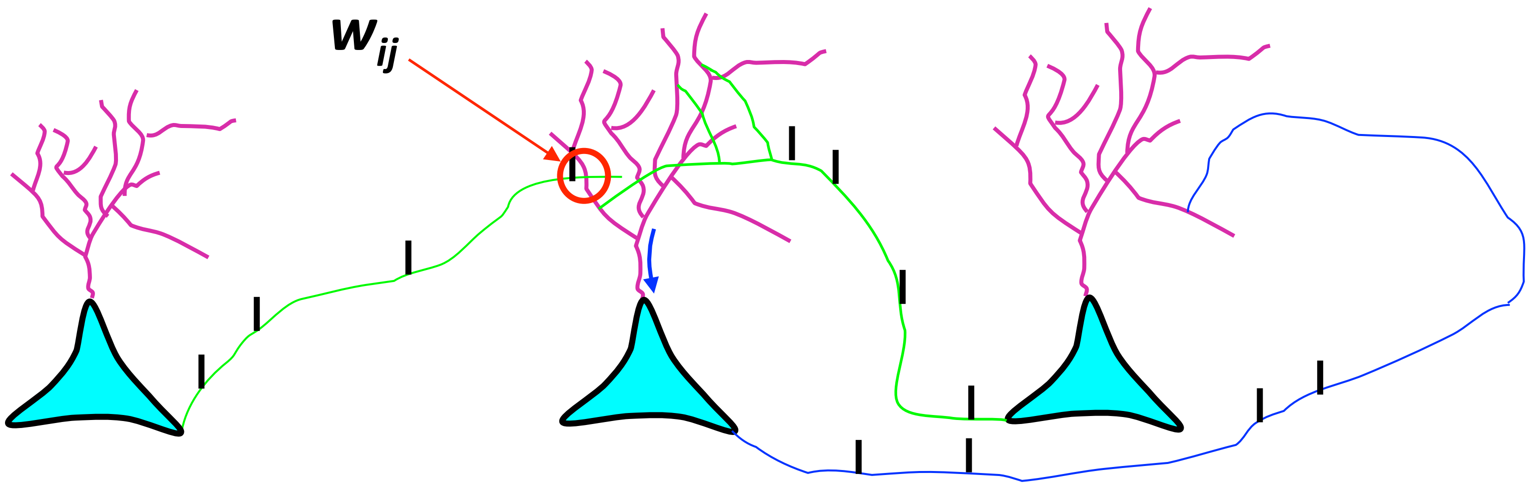


a. Anatomy. We know a lot about what is where. But be careful about labels: neurons in motor cortex sometimes respond to color.

Connectivity. We know (more or less) which area is connected to which.

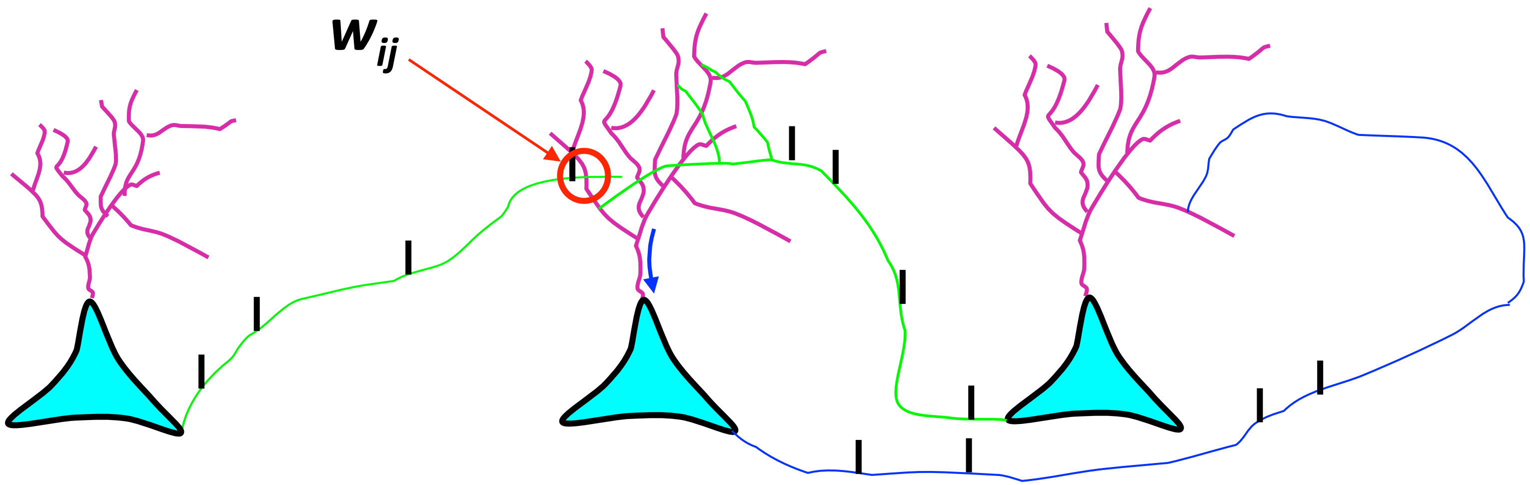
a. <u>Anatomy.</u> We know a lot about what is where. But be careful about labels: neurons in motor cortex sometimes respond to color.

<u>Connectivity.</u> We know (more or less) which area is connected to which. We don't know the wiring diagram at the microscopic level.

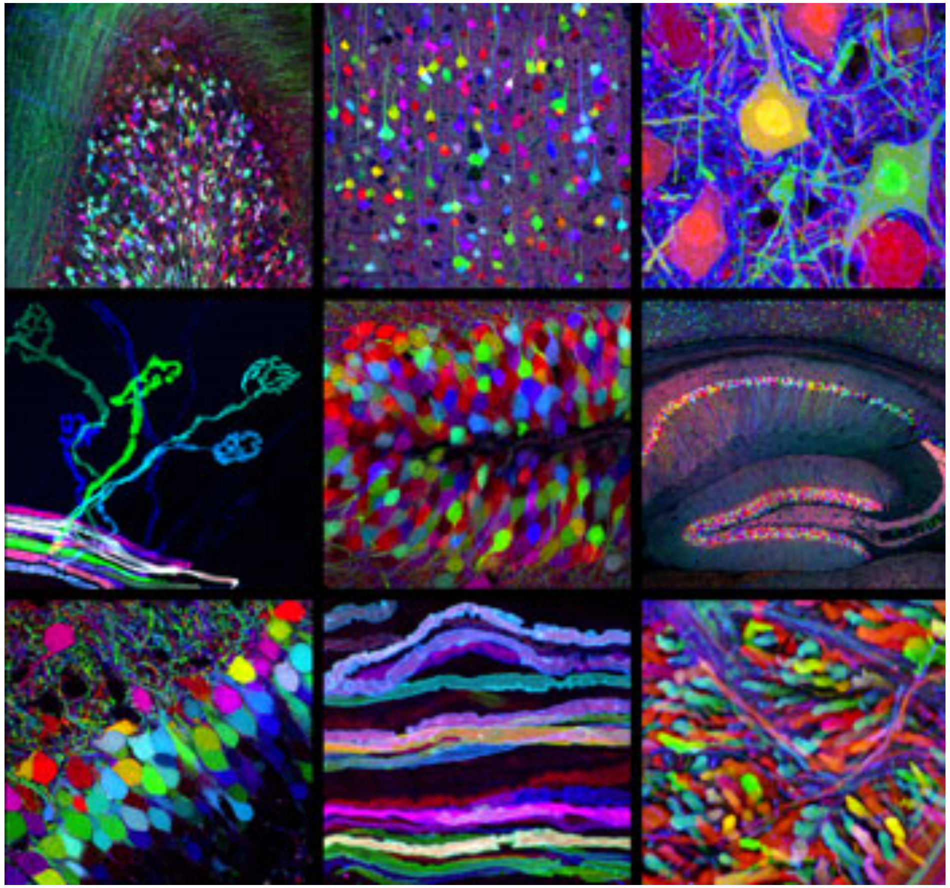


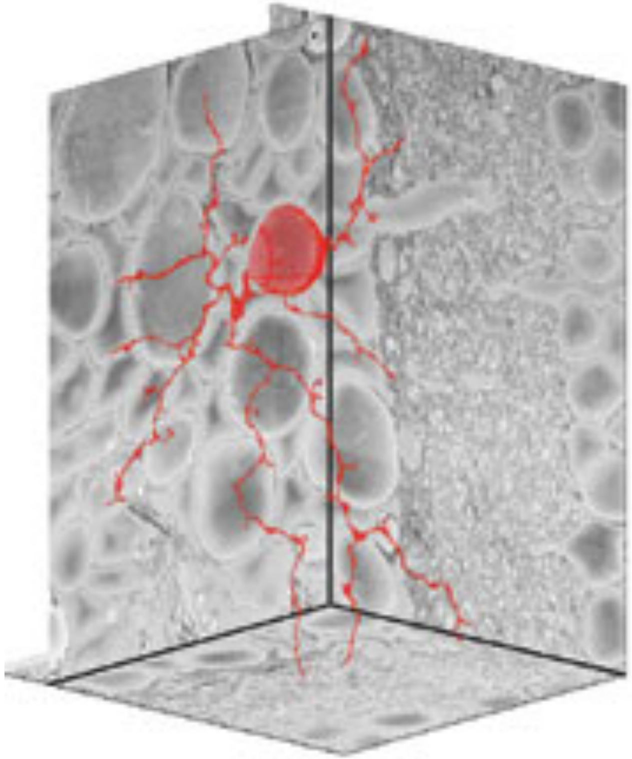
a. <u>Anatomy.</u> **We know a lot about what is where. But be careful about labels: neurons in motor cortex sometimes respond to color.**

<u>Connectivity.</u> **We know (more or less) which area is connected to which. We don't know the wiring diagram at the microscopic level. But we might in a few decades!**



Brainbow; Retina

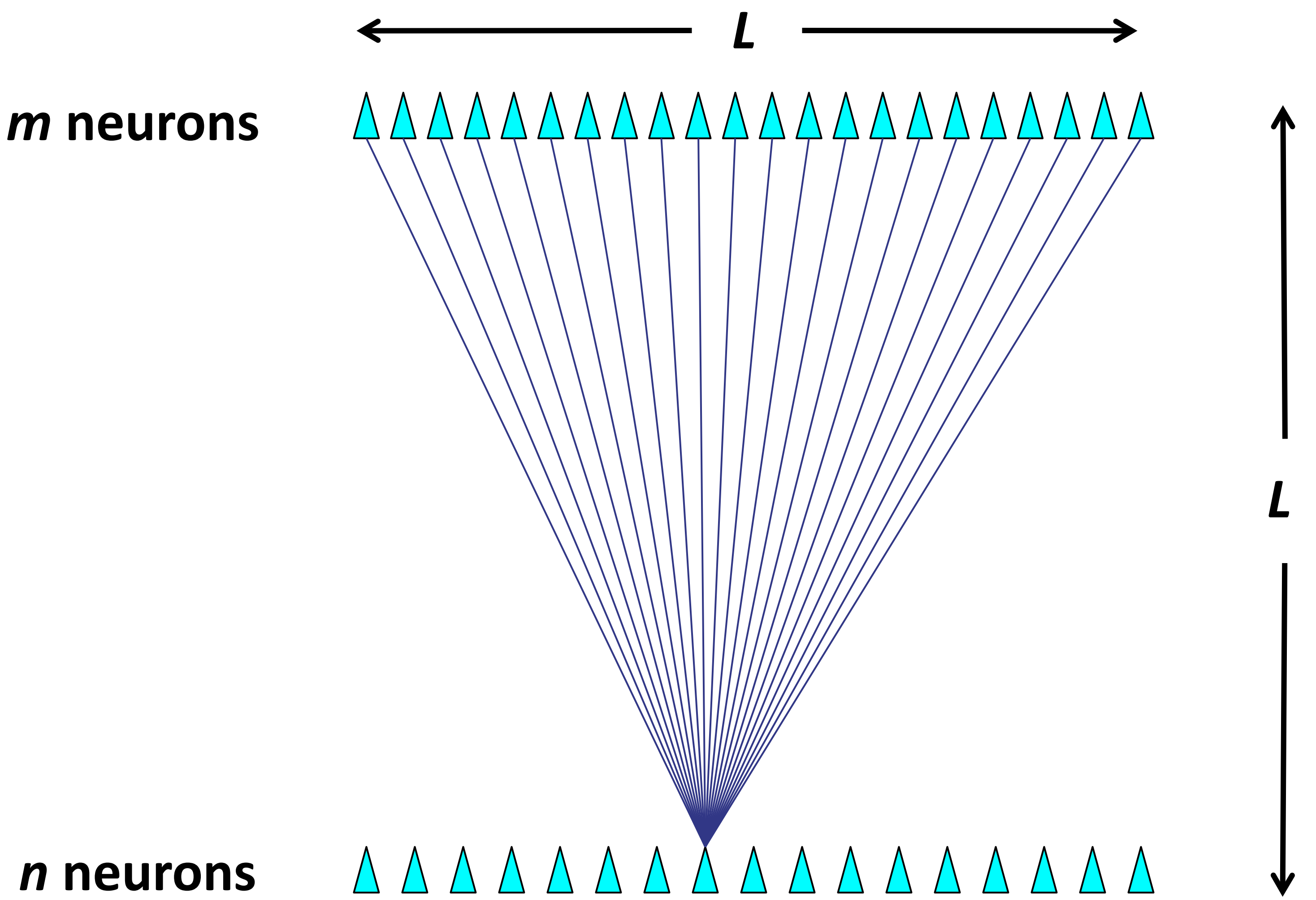




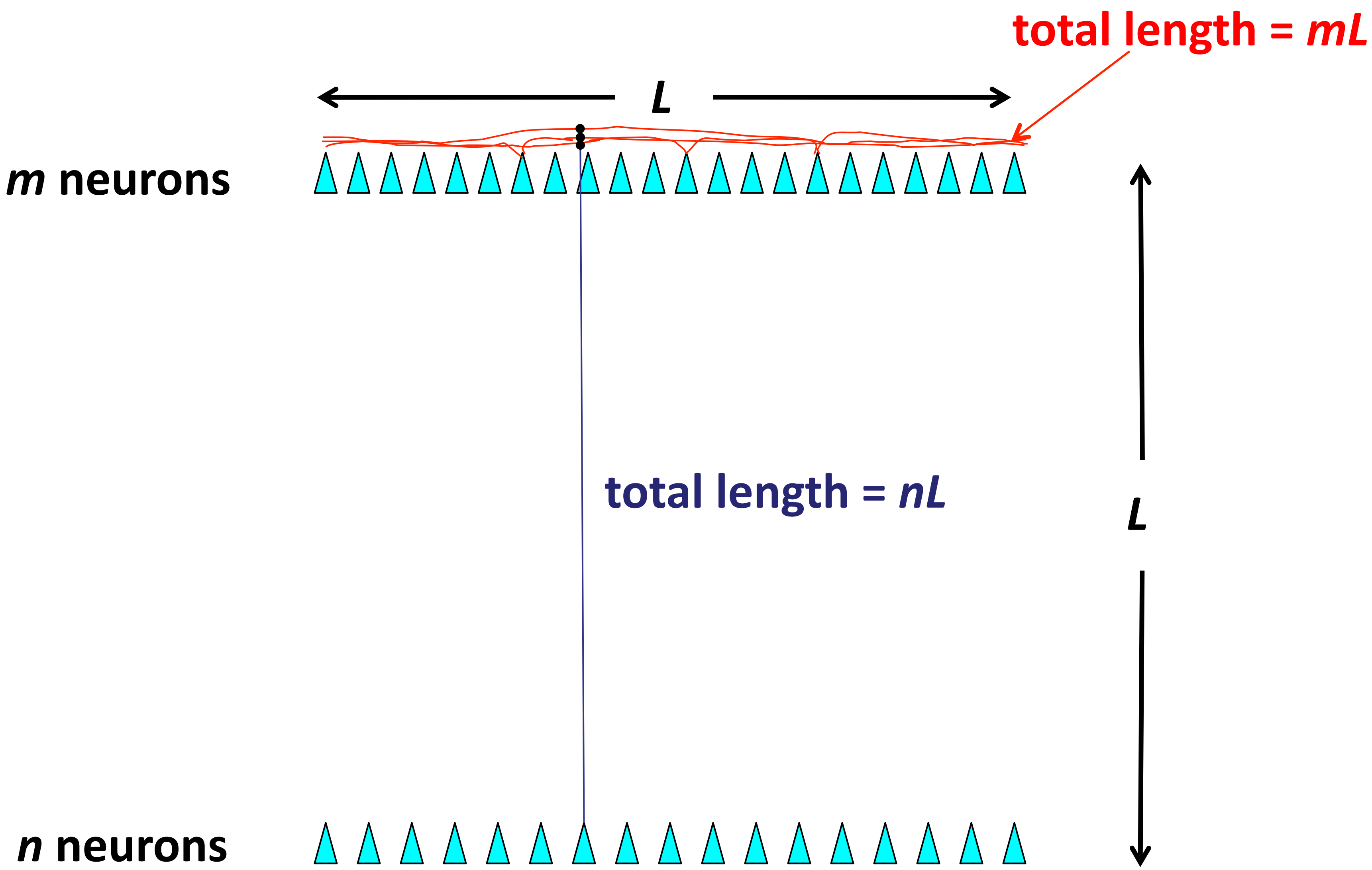
b. <u>Single neurons.</u> **We know very well how point neurons work (think Hodgkin Huxley).**

<u>Dendrites.</u> Lots of potential for incredibly complex processing.

My guess: all they do make neurons bigger and reduce wiring length (see the work of Mitya Chklovskii).



total wire length without dendrites: ~*nmL*



total wire length without dendrites: ~*nmL*

total wire length with dendrites: ~(*n*+*m*)*L*

b. **<u>Single neurons.</u> We know very well how point neurons work (think Hodgkin Huxley).**

<u>Dendrites.</u> Lots of potential for incredibly complex processing.

His guess: all they do is make neurons bigger and reduce wiring length (see the work of Mitya Chklovskii).

Requires: dendritic democracy...

How much PEL would bet that: 20 p.

c. <u>The neural code.</u>

His guess: once you get away from periphery, it's mainly firing rate: an inhomogeneous Poisson process with a refractory period is a good model of spike trains.

How much PEL would bet: £100.

<u>The role of correlations.</u> Still unknown.

His guess: don't have one.

<u>The roles of oscillations.</u> Much more complicated

d. <u>Networks of neurons.</u>

- **feedforward**
  - **many computations**
  - **kernel-universality**

- **recurrent:**
  - **`rate-based' neural dynamics**
    - **few key algorithms:**

      associative memory
      selective amplification
      resonance
  - **spike-based neural dynamics**
    - **balanced networks**

      associative memory

e. <u>Learning.</u> We know a lot of facts (LTP, LTD, STDP).

- **it's not clear which, if any, are relevant.**
- **the relationship between learning rules and computation is essentially unknown.**

• **supervised learning**	**(cerebellum)**
• **unsupervised learning**	**(neocortex)**
• **reinforcement learning**	**(basal ganglia)**

A word about learning (remember these numbers!!!):

You have about 10^{15} synapses.

If it takes 1 bit of information to set a synapse,
you need 10^{15} bits to set all of them.

30 years ≈ 10^{9} seconds.

To set 1/10 of your synapses in 30 years,

you must absorb 100,000 bits/second.

Learning in the brain is almost completely unsupervised!!!

f. **Where we know algorithms we know the neural implementation (sort of):**

sound localization, addition, reward learning

This is not a coincidence...

Remember David Marr:

1. **the problem (computational level)**
2. **the strategy (algorithmic level)**
3. **how it's actually done by networks of neurons (implementational level)**

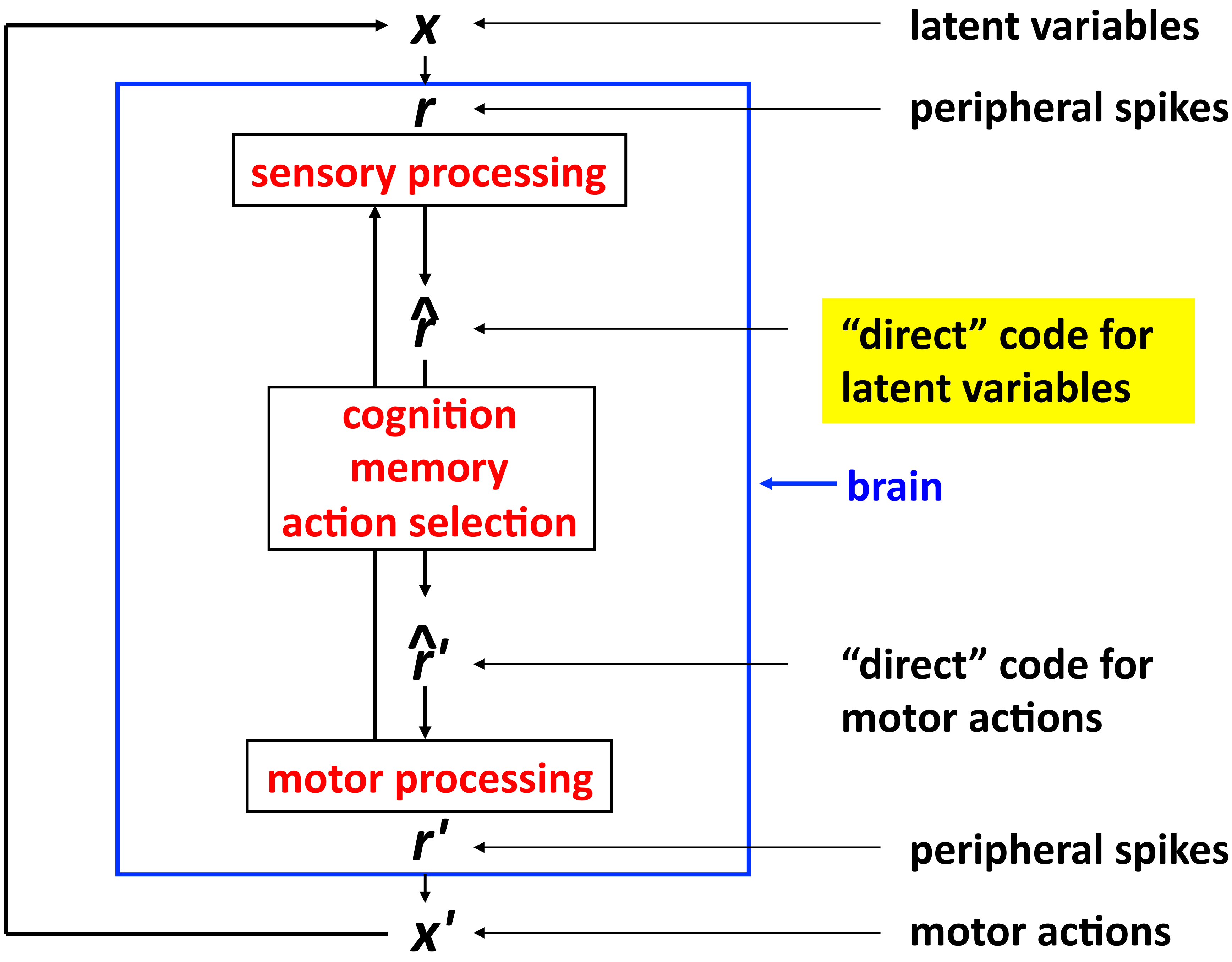
What we know: PEL's score (1-10).

a. Anatomy.	5
b. Single neurons.	6
c. The neural code.	6
d. Recurrent networks of neurons.	3
e. Learning.	2

The hard problems:

1. How does the brain extract latent variables?	1.001
2. How does it manipulate latent variables?	1.002
3. How does it learn to do both?	1.001

Perception Action Cycle



x
latent variables
r
peripheral spikes
sensory processing
$\hat{r}$
"direct" code for latent variables
cognition
memory
action selection
brain
$\hat{r}'$
"direct" code for motor actions
motor processing
r'
peripheral spikes
x'
motor actions

versus weakly tickled internal processing?
autopoiesis

Outline:

1. Systems neuroscience Dayan
2. Language of neurons: neural coding. Sahani
3. Basics: single neurons/axons/dendrites/synapses. Latham
4. Learning at the network and behavioral level. Dayan
5. What we know about networks (very little). Latham
6. Uncertainty Dayan