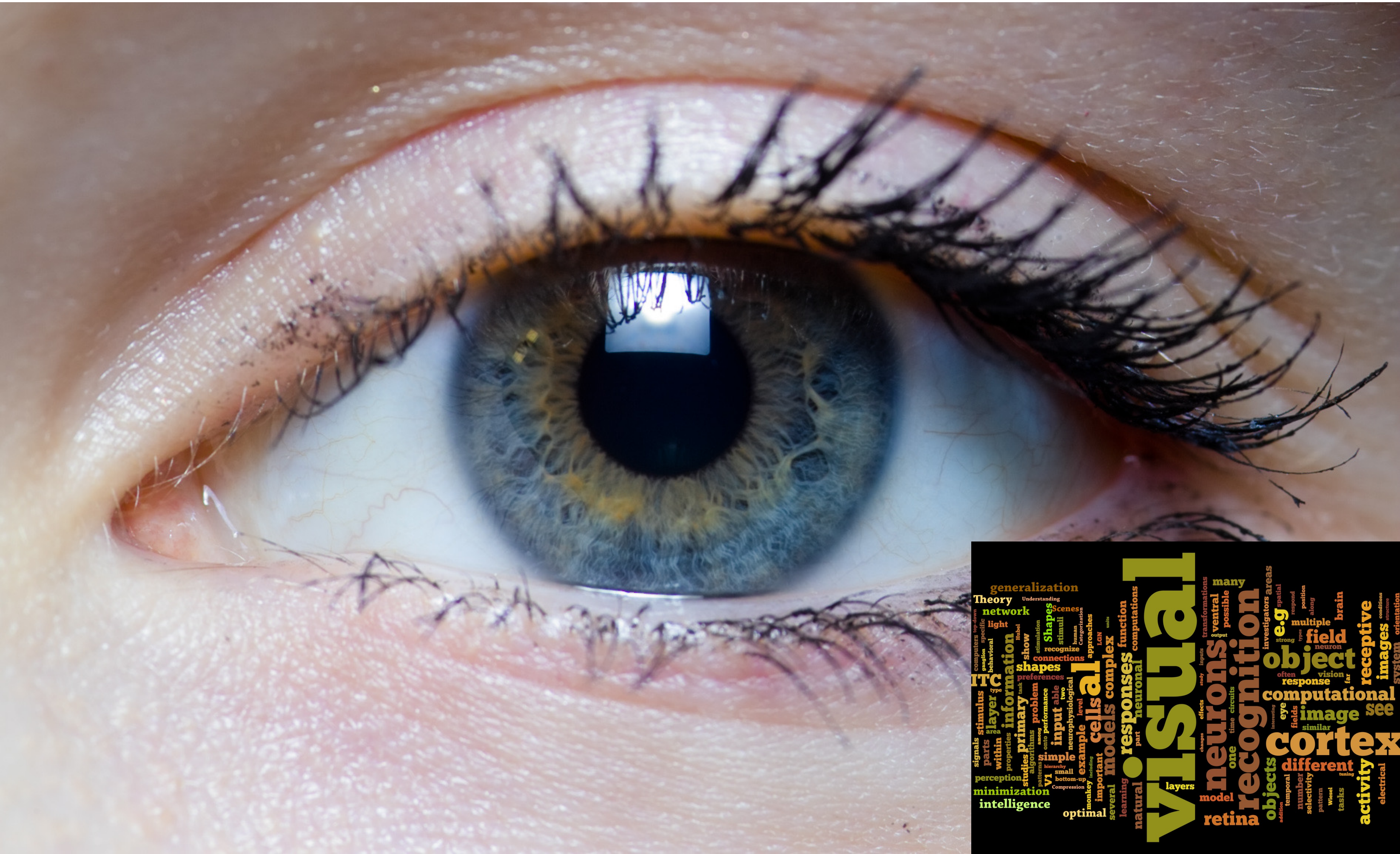
Visual Object Recognition

Computational Models and Neurophysiological Mechanisms

Neuro 130/230. Harvard College/GSAS 78454



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

Note: no class on 09/04/2023 (Labor Day)

Class 1 [09/11/2023]. Introduction to Vision

Class 2 [09/18/2023]. The Phenomenology of Vision

Class 3 [09/25/2023]. Natural image statistics and the retina

Class 4 [10/02/2023]. Learning from Lesions

Note: no class on 10/09/2023 (Indigenous Day)

Class 5 [10/16/2023]. Primary Visual Cortex

Class 6 [10/23/2023]. Adventures into *terra incognita*

Class 7 [10/30/2023]. From the Highest Echelons of Visual Processing to Cognition

Class 8 [11/06/2023]. First Steps into in silico vision

Class 9 [11/13/2023]. Teaching Computers how to see

Class 10 [11/20/2023]. Computer Vision

Class 11 [11/27/2023]. Connecting Vision to the rest of Cognition [Dr. Will Xiao]

Class 12 [12/06/2023]. Visual Consciousness

FINAL EXAM, PAPER DUE 12/11/2023. No extensions.

1. Why build models?
2. Single neuron models
3. Network models

Questions to keep in mind:

- What are models good for and (as important) not good for?
- What is the right level of abstraction?

What is a model?

- Model organisms
- Circuit model
- Ball-and-stick model
- Network model
- …

What is a model?

- Model organisms
- Circuit model
- Ball-and-stick model
- Network model
- …

A model is a *stand-in*

- Not the real thing
- Captures properties *of interest*
- More *useful* than the real thing in some way
  - Easier to manipulate
  - Cheaper to test, etc.

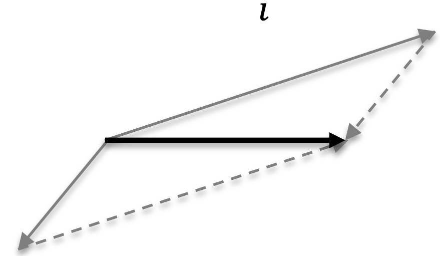
Why build models?

What's the alternative?

- Where on earth is this?
- How did this happen?
- What is the weather?
- How much money is being lost?
- How many containers are on the ship?
- *What is the mass of the ship?*
- *What is the net force on the ship?*
- Where is the force exerted?
- What did the captain eat for breakfast?
- How is this photo taken?
- …

$$\boldsymbol{F} = m\boldsymbol{a}$$

$$\boldsymbol{F} = \sum_i \boldsymbol{F}_i$$



Why build models?

(Good) Models:

- Represent understanding
  - What matters and what does not
  - What is cause and what is effect
- Are useful
- Are not the real thing!

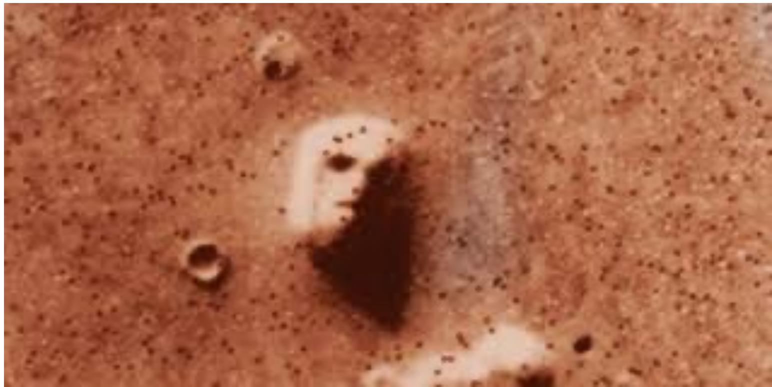
Why build *quantitative* models?

What's the alternative?

Verbal models:

> "We found an area in the fusiform gyrus […] that was significantly more active when the subjects viewed faces than when they viewed assorted common objects"
> (https://www.jneurosci.org/content/17/11/4302)

- What counts as "faces"?
- How much more active?
- Do results depend on details of the experiment? (Images used, presentation duration, what about during natural behavior, etc…)
- How would this area respond to, say, pareidolia?



Why build quantitative models?

Verbal models are:

- Vague, prone to subjective interpretation
- Unable to make quantitative predictions
- Not falsifiable

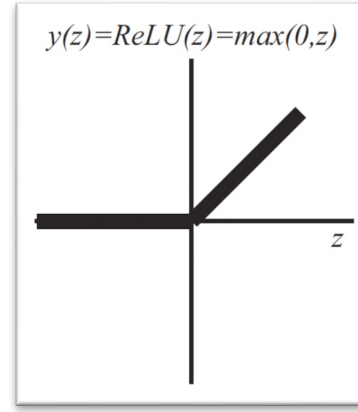
Quantitative models:

- Are formal, unambiguous, falsifiable
- *Can* capture diverse experiments, range of resolutions
- *Can* lead to (non-intuitive) predictions
- *Can* point to missing data, critical information, decisive experiments
- *Can* be useful as an engineering product (e.g., face recognition)

Models of the brain

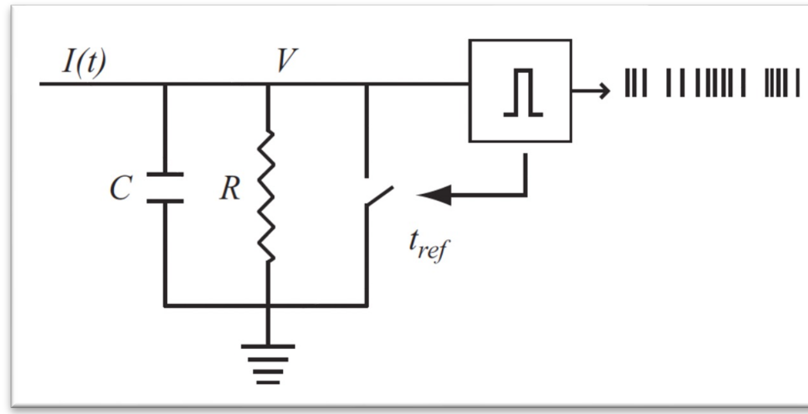
1. Why build models?
   - They represent understanding
   - They are useful (for testing, predicting, …)
2. **Single neuron models**
3. Network models

Even single neuron models have differing levels of abstraction



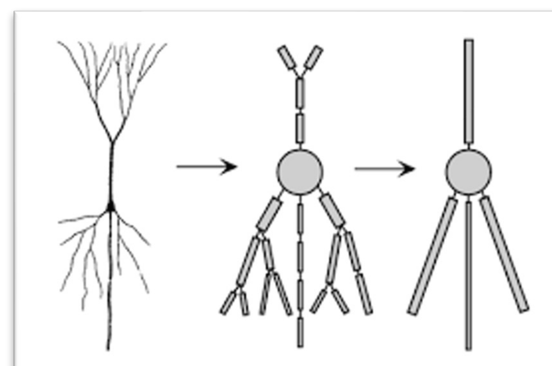
Thresholded weighted sum of inputs

Integrate-and-fire model



Hodgkin-Huxley model

Multi-compartment models



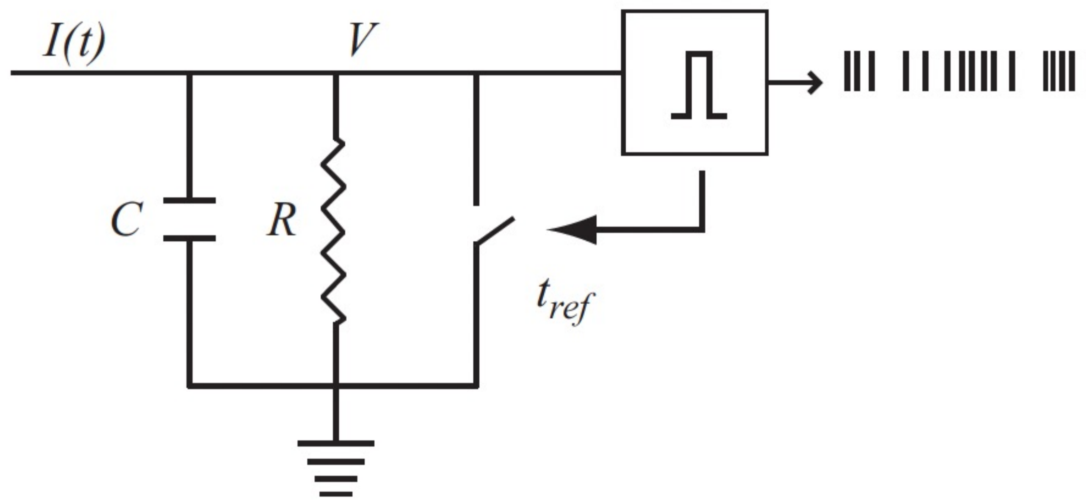
Spines and ion channels

Increasing:
- Biological realism
- Level of detail

Decreasing
- Analytical tractability
- Computational ease

Source: U Wash CSE 528

The leaky integrate-and-fire model (Lapicque 1907)



Below threshold, the voltage follows:
(Just physics, given the wiring diagram model above)

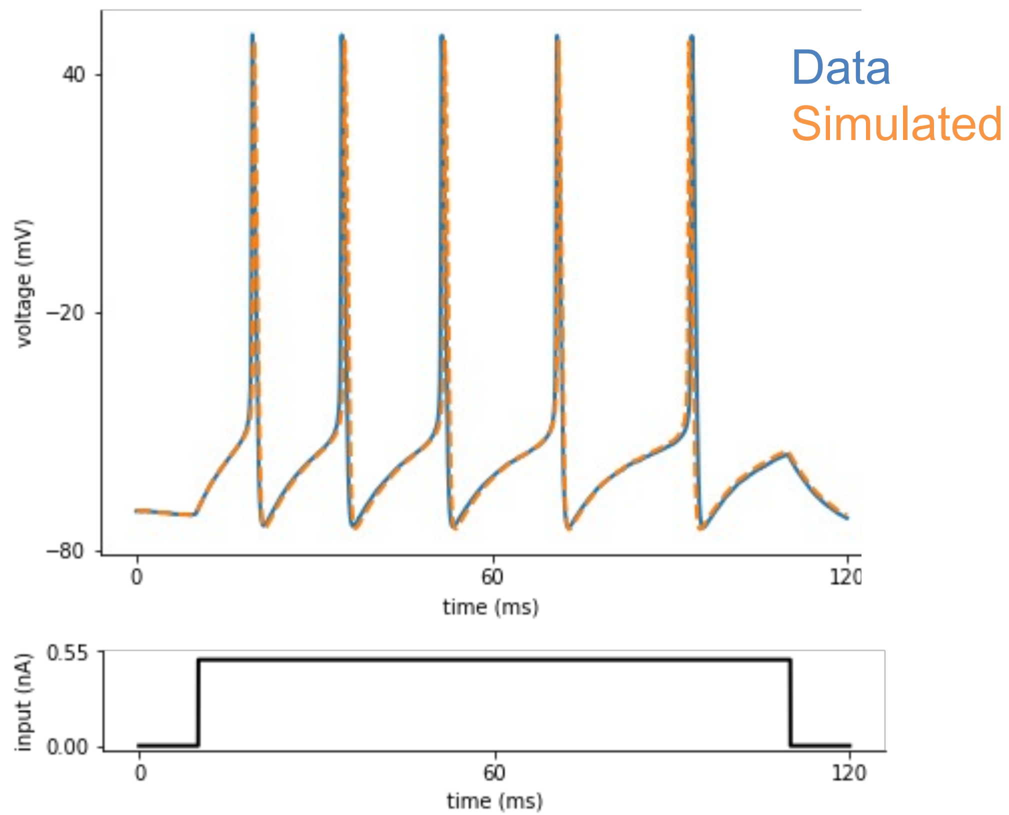
$$C\frac{dV(t)}{dt} = -\frac{V(t)}{R} + I(t)$$

1. A spike is fired when $V(t) > V_{thr}$; $V(t)$ is reset after each spike
2. After each spike, a refractory period t_{ref} is imposed

- Simple and fast
- Does not consider sub-ms dynamics (e.g., temporal shape of action potential), ion channel mechanics, spike-rate adaptation, neuronal geometry, etc

The Hodgkin-Huxley model

Gives us detailed (time-resolved) shape of action potential, as a function of input current



Source: mackelab/sbi
[python package tutorial]

The Hodgkin-Huxley model

Models: voltage and current across neuron membrane
(a “spike” is just change of voltage in time)

1. Neuron membrane ≈ capacitor
2. Current-voltage relationship for a capacitor:
4. Current due to ion flow (current is nothing but flow of electrical charge, e.g., ions):

$$I = C_m \frac{dV_m}{dt} \qquad + I_{ionic}$$

Rearrange:

$$\frac{dV_m}{dt} = \frac{1}{C_m}(I - I_{ionic})$$

$$\frac{dV_m}{dt} = \frac{1}{C_m}\left(I - \bar{g}_{\mathrm{K}} n^4 (V_m - V_{\mathrm{K}}) - \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{\mathrm{Na}}) - \bar{g}_l (V_m - V_l)\right)$$

The Hodgkin-Huxley model
(Slides by Ben de Bivort from LS50 2019)

$$\frac{dV_m}{dt} = \frac{1}{C_m}(I - \bar{g}_{\mathrm{K}}n^4(V_m - V_{\mathrm{K}}) - \bar{g}_{\mathrm{Na}}m^3h(V_m - V_{\mathrm{Na}}) - \bar{g}_l(V_m - V_l))$$

$$\frac{dn}{dt} = \alpha_n(V_m)(1-n) - \beta_n(V_m)n$$

$$\frac{dm}{dt} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m$$

$$\frac{dh}{dt} = \alpha_h(V_m)(1-h) - \beta_h(V_m)h$$

The Hodgkin-Huxley model
(Slides by Ben de Bivort from LS50 2019)

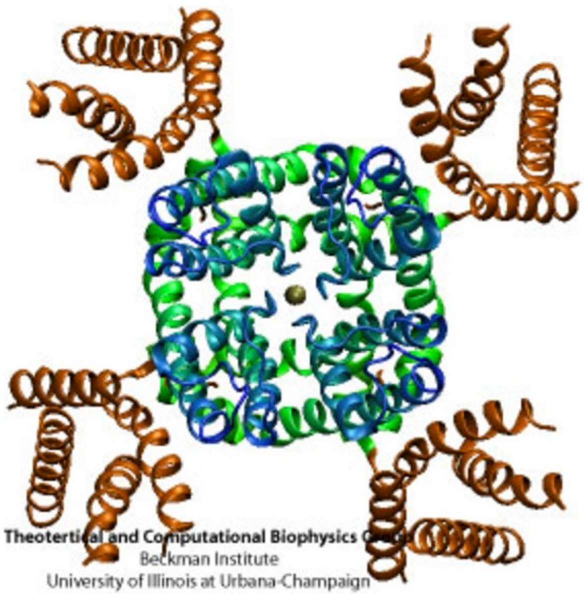
Potassium current, proportional to 1) a rate constant, 2) the 4th power of the fraction of occupied potassium channel sites, and 3) the membrane potential difference from the reversal potential of potassium

$$\frac{dV_m}{dt} = \frac{1}{C_m}(I - \bar{g}_{\mathrm{K}} n^4 (V_m - V_{\mathrm{K}}) - \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{\mathrm{Na}}) - \bar{g}_l (V_m - V_l))$$

$$\frac{dn}{dt} = \alpha_n(V_m)(1-n) - \beta_n(V_m)n$$

$$\frac{dm}{dt} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m$$

$$\frac{dh}{dt} = \alpha_h(V_m)(1-h) - \beta_h(V_m)h$$



The Hodgkin-Huxley model

(Slides by Ben de Bivort from LS50 2019)

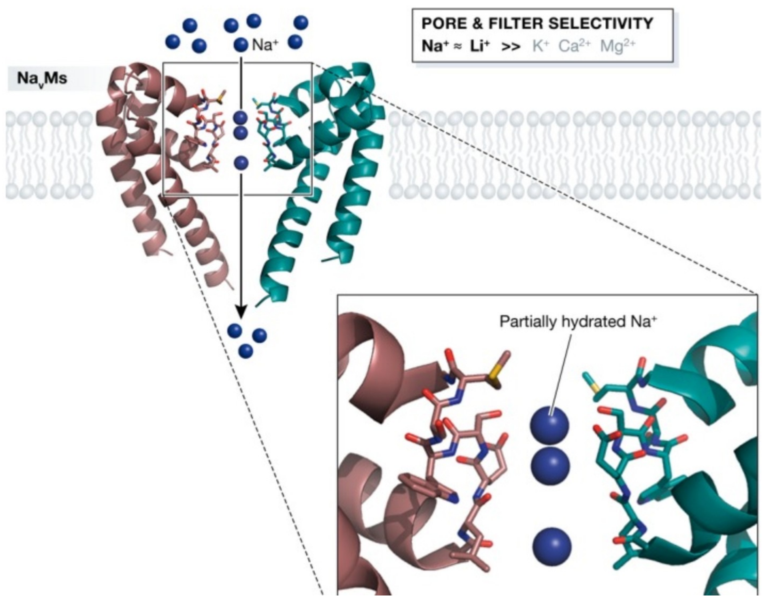
Sodium current, proportional to 1) a rate constant, 2) the 3rd power of the fraction of occupied sodium channel sites, 3) the portion of sodium channels in the activated state, and 4) the membrane potential difference from the reversal potential of sodium

$$\frac{dV_m}{dt} = \frac{1}{C_m}(I - \bar{g}_{\mathrm{K}} n^4 (V_m - V_{\mathrm{K}}) - \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{\mathrm{Na}}) - \bar{g}_l (V_m - V_l))$$

$$\frac{dn}{dt} = \alpha_n(V_m)(1-n) - \beta_n(V_m)n$$

$$\frac{dm}{dt} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m$$

$$\frac{dh}{dt} = \alpha_h(V_m)(1-h) - \beta_h(V_m)h$$



The Hodgkin-Huxley model
(Slides by Ben de Bivort from LS50 2019)

General leak current, proportional to a 1) rate constant, and 2) the membrane potential difference from the reversal potential of all other ion species.

$$\frac{dV_m}{dt} = \frac{1}{C_m}(I - \bar{g}_{\mathrm{K}} n^4 (V_m - V_{\mathrm{K}}) - \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{\mathrm{Na}}) - \bar{g}_l (V_m - V_l))$$

$$\frac{dn}{dt} = \alpha_n(V_m)(1 - n) - \beta_n(V_m)n$$

$$\frac{dm}{dt} = \alpha_m(V_m)(1 - m) - \beta_m(V_m)m$$

$$\frac{dh}{dt} = \alpha_h(V_m)(1 - h) - \beta_h(V_m)h$$

The Hodgkin-Huxley model
(Slides by Ben de Bivort from LS50 2019)

n = fraction of bound potassium channel sites

m = fraction of bound sodium channel sites

h = fraction of active-state sodium channel sites

$$\frac{dV_m}{dt} = \frac{1}{C_m}(I - \bar{g}_{\mathrm{K}} n^4 (V_m - V_{\mathrm{K}}) - \bar{g}_{\mathrm{Na}} m^3 h (V_m - V_{\mathrm{Na}}) - \bar{g}_l (V_m - V_l))$$

$$\frac{dn}{dt} = \alpha_n(V_m)(1 - n) - \beta_n(V_m) n$$

$$\frac{dm}{dt} = \alpha_m(V_m)(1 - m) - \beta_m(V_m) m$$

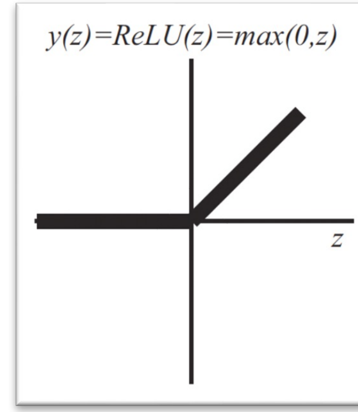
$$\frac{dh}{dt} = \alpha_h(V_m)(1 - h) - \beta_h(V_m) h$$

n, m and h are dimensionless and range from 0 to 1.

Could they go outside that range?

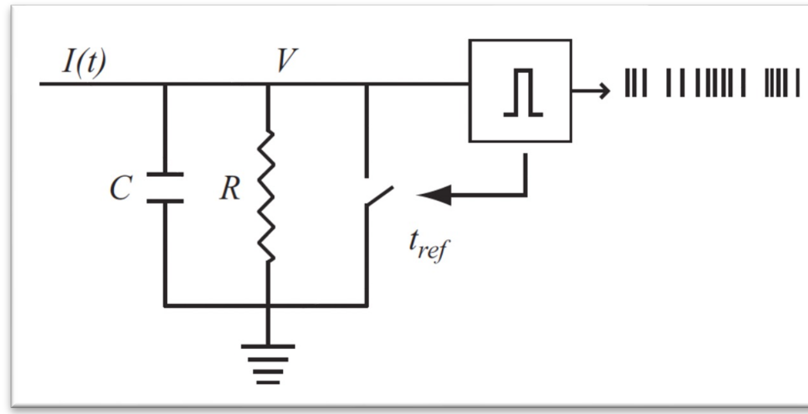
non-linear functions α_i and β_i describe how they grow and shrink

Single neuron models have differing levels of abstraction



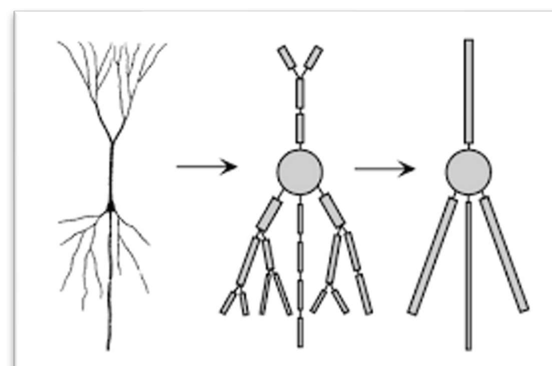
Weighted sum of inputs + nonlinearity

Integrate-and-fire circuit



Hodgkin-Huxley model

Multi-compartment models



Spines and ion channels

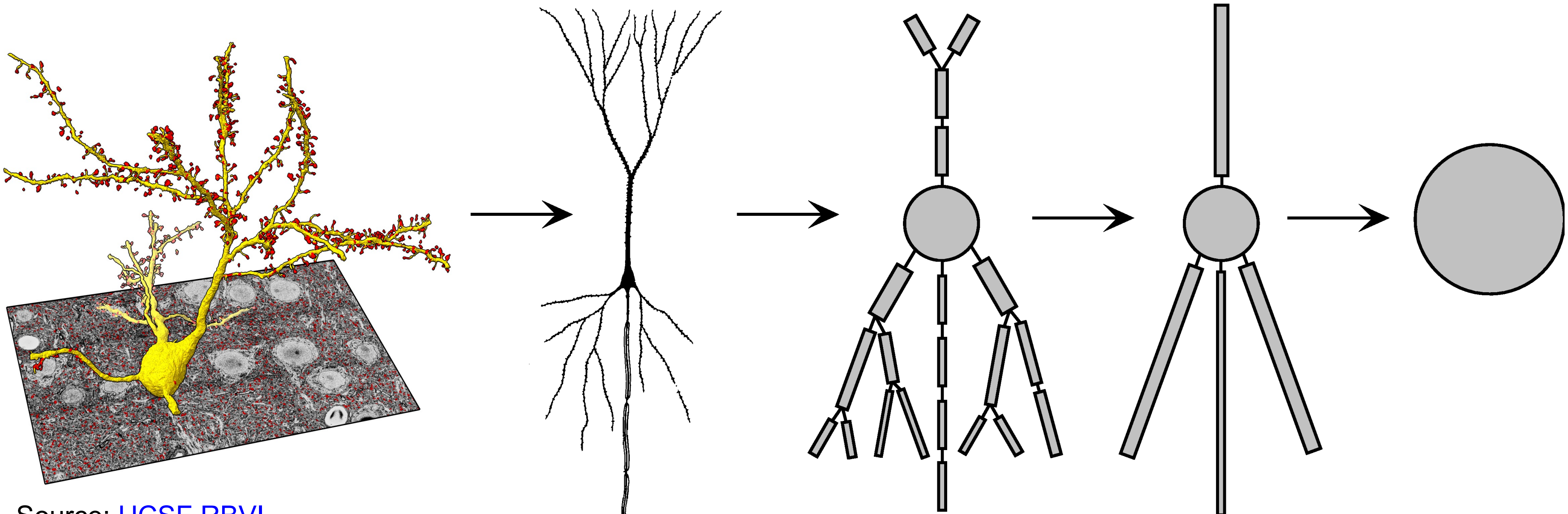
Increasing:

- Biological realism
- Level of detail

Decreasing

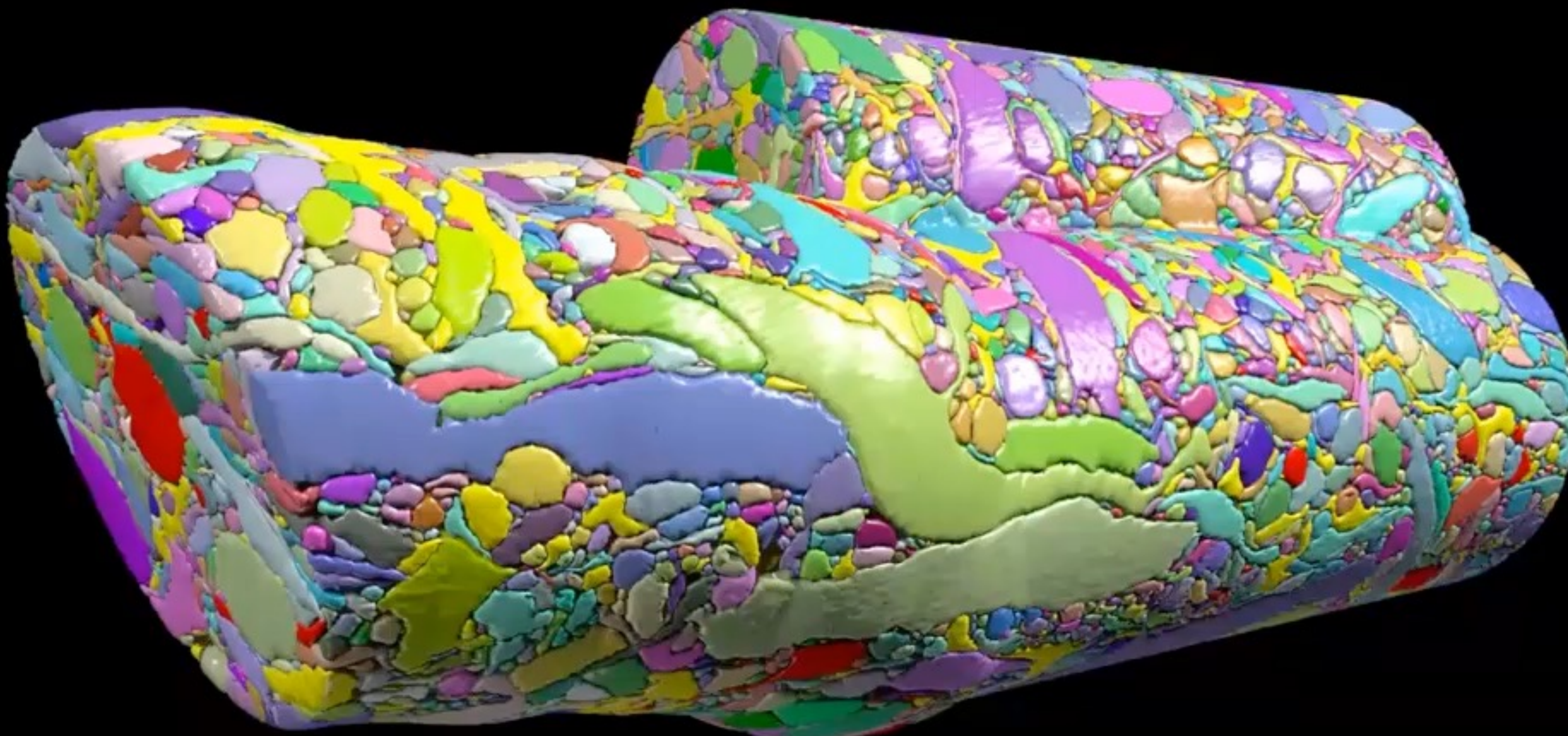
- Analytical tractability
- Computational ease

Multi-compartmental models



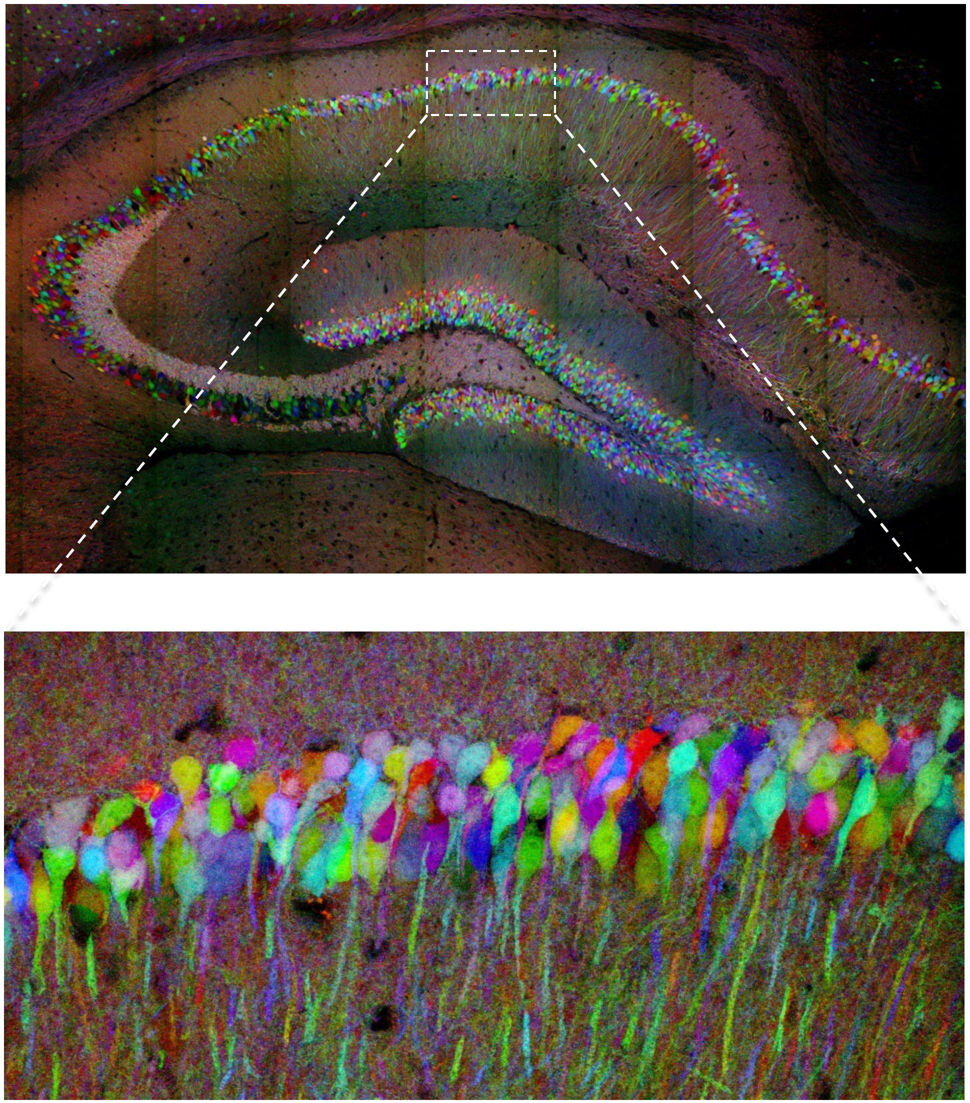
Source: UCSF RBVI

What is the “right” level of abstraction?
—a central question in neuroscience

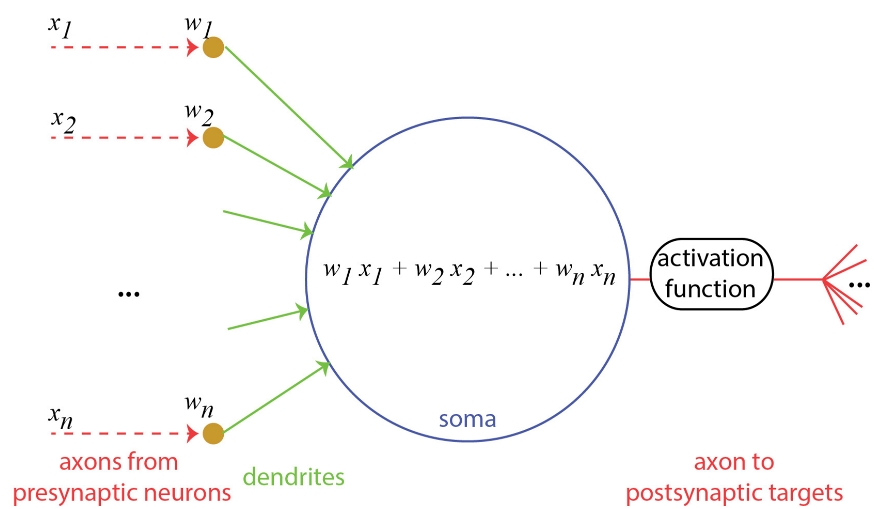


Kasthuri et al., 2015

Weighted sum + nonlinearity: typical "neuron" in network models

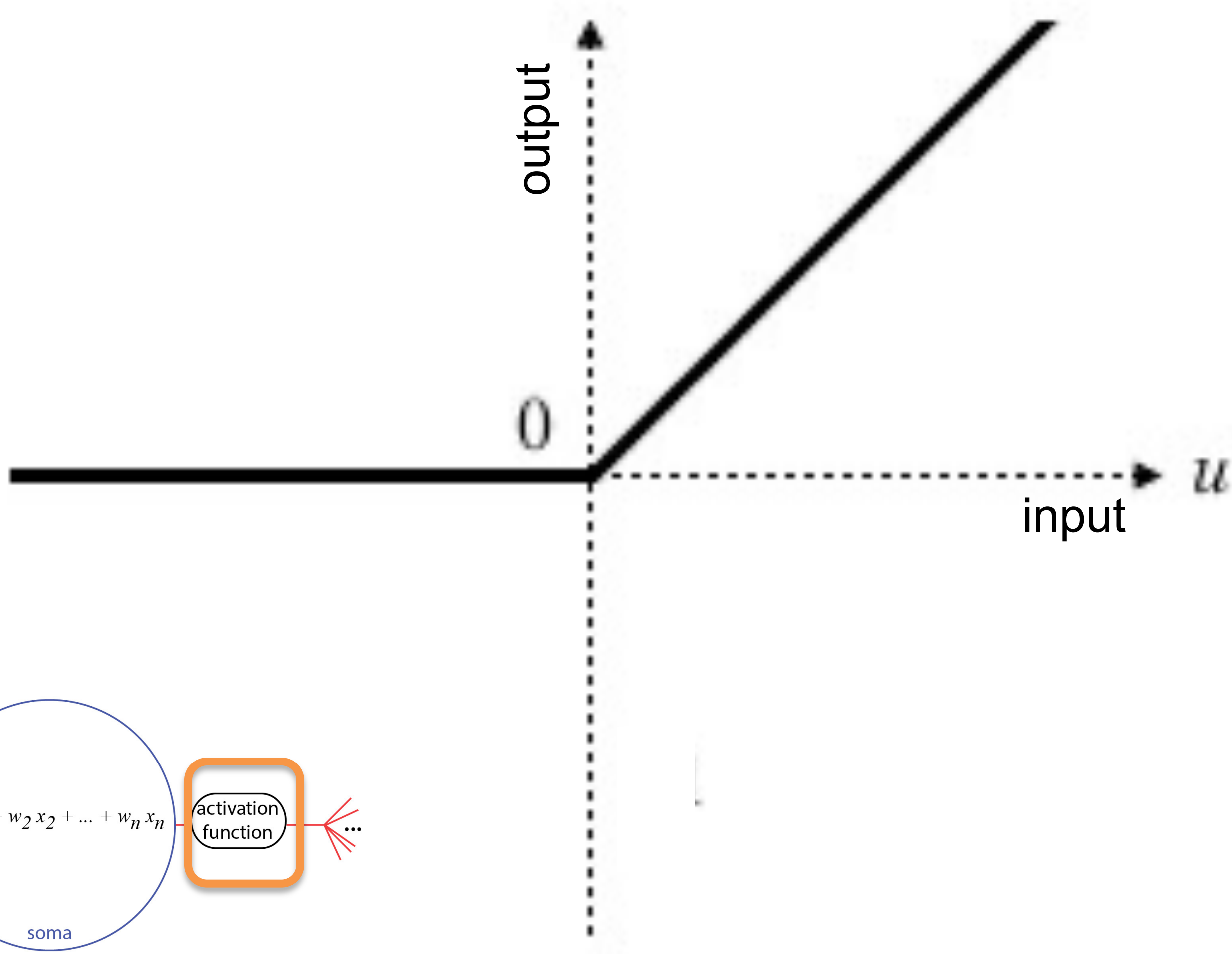


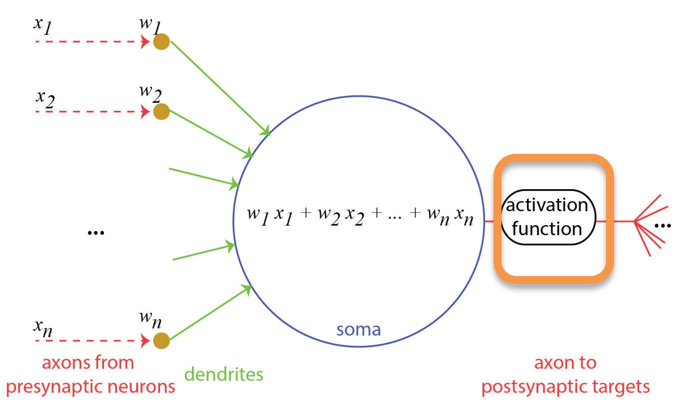
Source: Livet et al., 2007



Rectified linear unit (ReLu): The most common activation function

$$f(u) = \max(0, u)$$





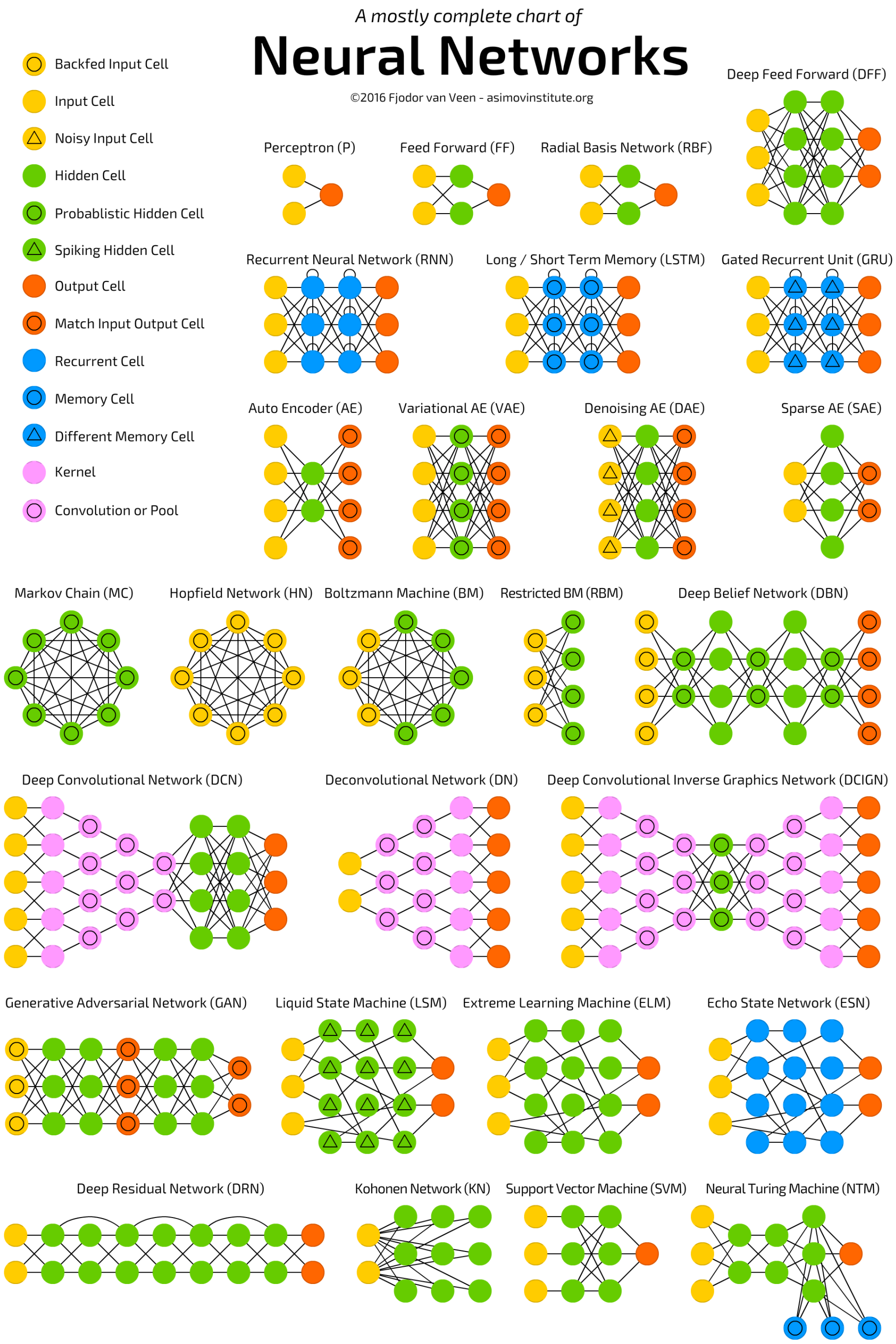
Models of the brain

1. Why build models?

2. Single neuron models
   - Capture varying levels of detail,
     from static to dynamic to multi-compartmental models

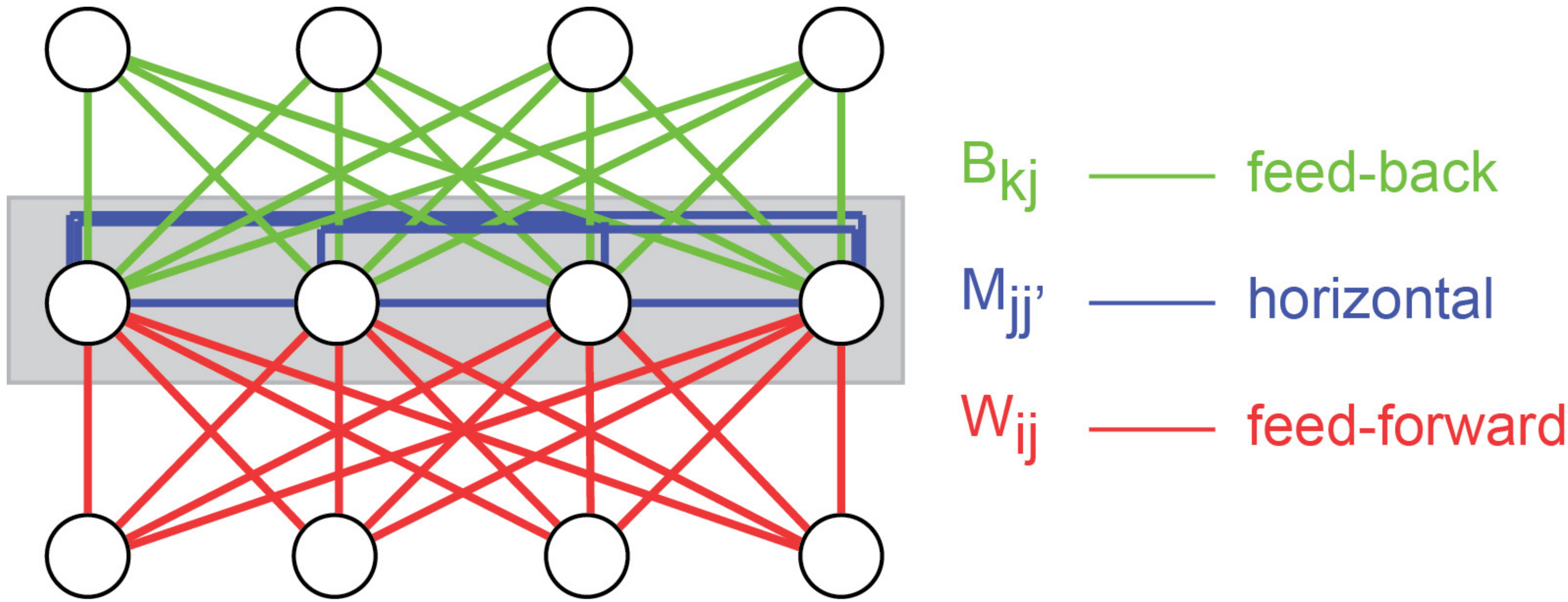
3. **Network models**
   - Supervised learning; perceptron and MNIST
   - (Backpropagation)
   - Convolution
   - Hopfield network

From a (few) simple neuron type(s), a wide variety of networks



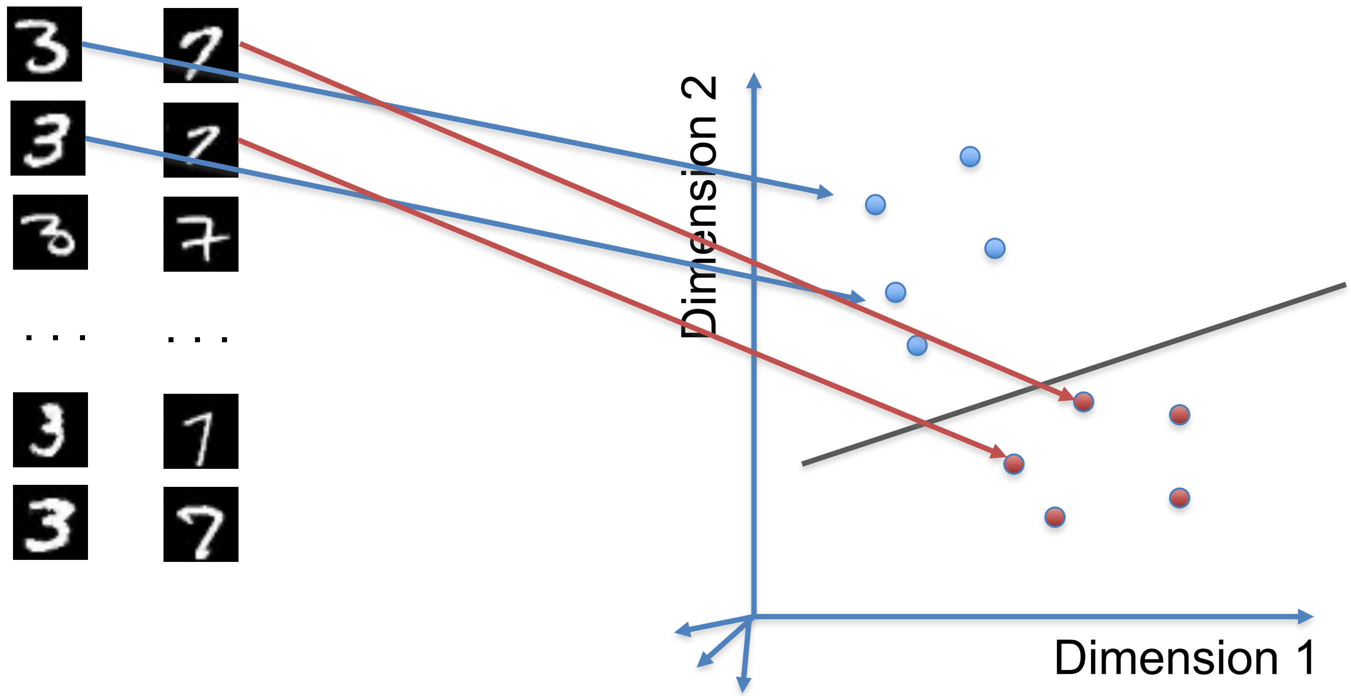
https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

Basic connection types in a circuit



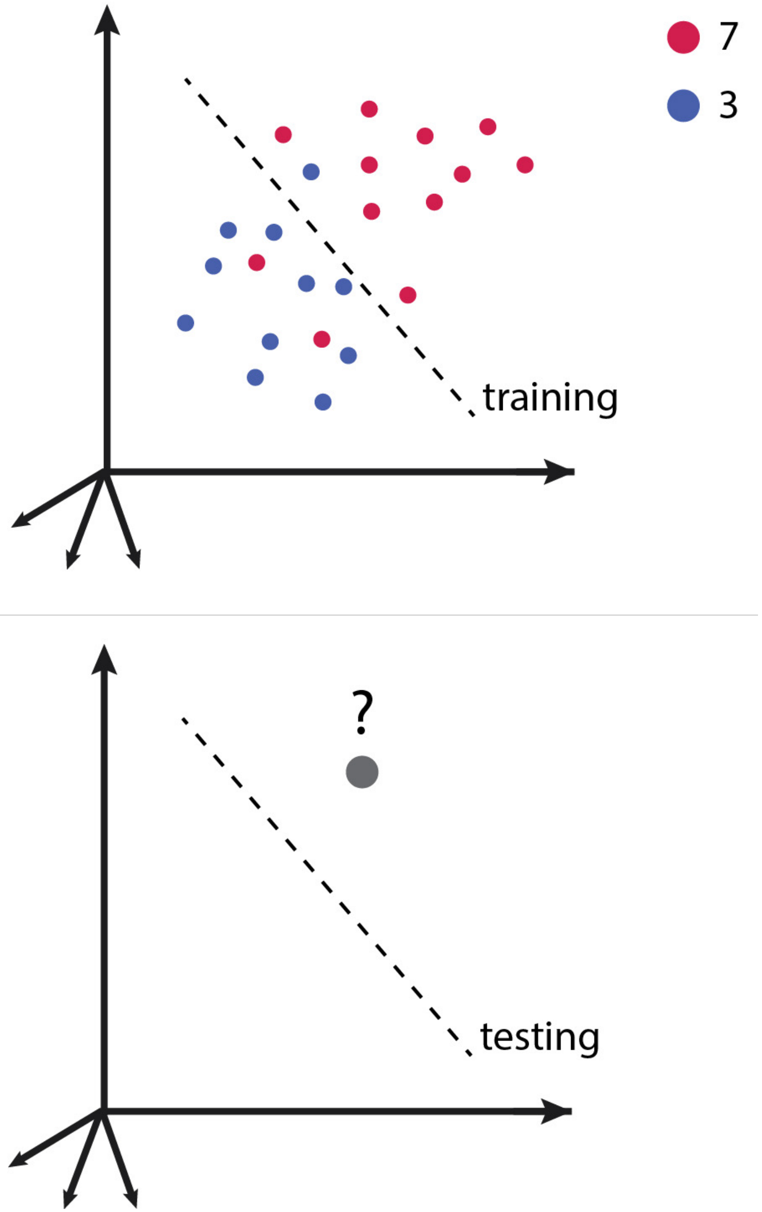
Supervised learning on MNIST (digit classification)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	115	121	162	253	253	213	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	63	107	170	251	252	252	252	252	250	214	0	0	0	0	0	0
0	0	0	0	0	0	0	0	25	192	226	226	241	252	253	202	252	252	252	252	252	225	0	0	0	0	0	0
0	0	0	0	0	0	0	68	223	252	252	252	252	252	39	19	39	65	224	252	252	183	0	0	0	0	0	0
0	0	0	0	0	0	0	186	252	252	252	245	108	53	0	0	0	150	252	252	220	20	0	0	0	0	0	0
0	0	0	0	0	0	70	242	252	252	222	59	0	0	0	0	0	178	252	252	141	0	0	0	0	0	0	0
0	0	0	0	0	0	185	252	252	194	67	0	0	0	0	17	90	240	252	194	67	0	0	0	0	0	0	0
0	0	0	0	0	0	83	205	190	24	0	0	0	0	0	121	252	252	209	24	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	77	247	252	248	106	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	253	252	252	102	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	134	255	253	253	39	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6	183	253	252	107	2	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	10	102	252	253	163	16	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	13	168	252	252	110	2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	41	252	252	217	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	40	155	252	214	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	165	252	252	106	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	43	179	252	150	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	137	252	221	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	67	252	79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

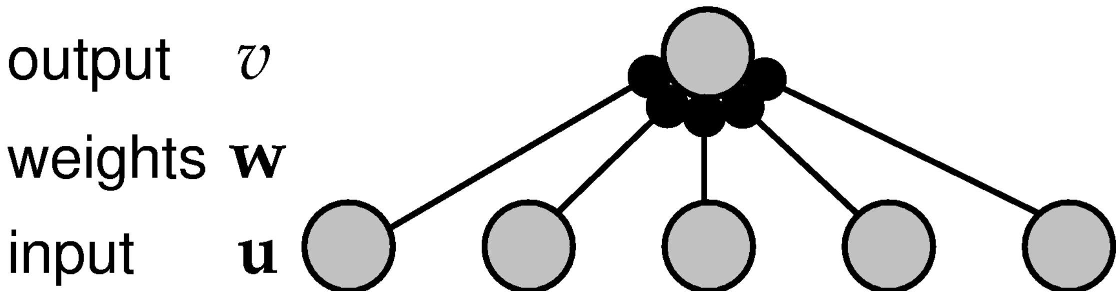


Dimension 2
Dimension 1

Cross validation



The perceptron (1-layer linear network)



Task: binary classification (e.g., 3 vs. 7)

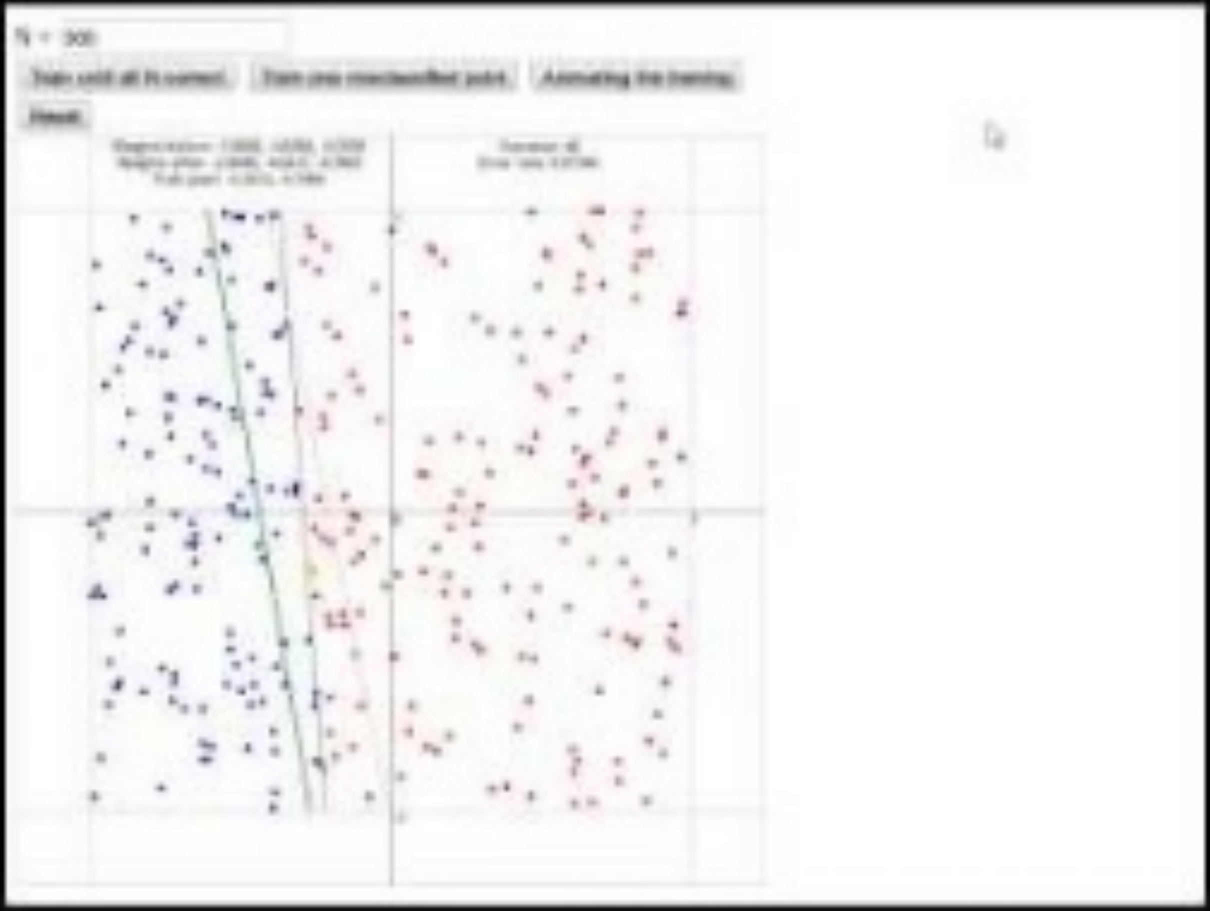
$$v = \begin{cases} +1 & \text{if } \mathbf{w}.\mathbf{u} - \gamma \geq 0 \\ -1 & \text{if } \mathbf{w}.\mathbf{u} - \gamma < 0 \end{cases}$$

Learning rule:

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

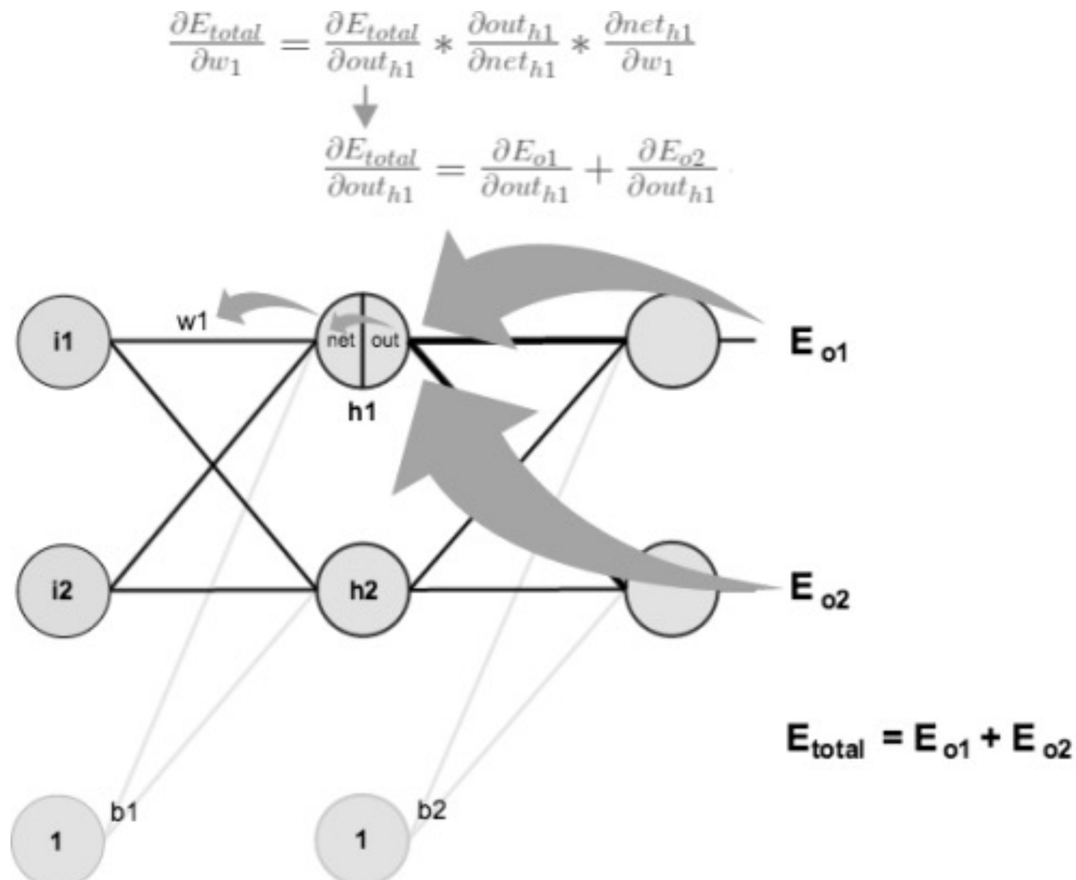
For training examples $\{\mathbf{u}_{\mathrm{m}}, v_m\}$, $\mathrm{m} = 1,\ldots,$dataset size

Training the perceptron



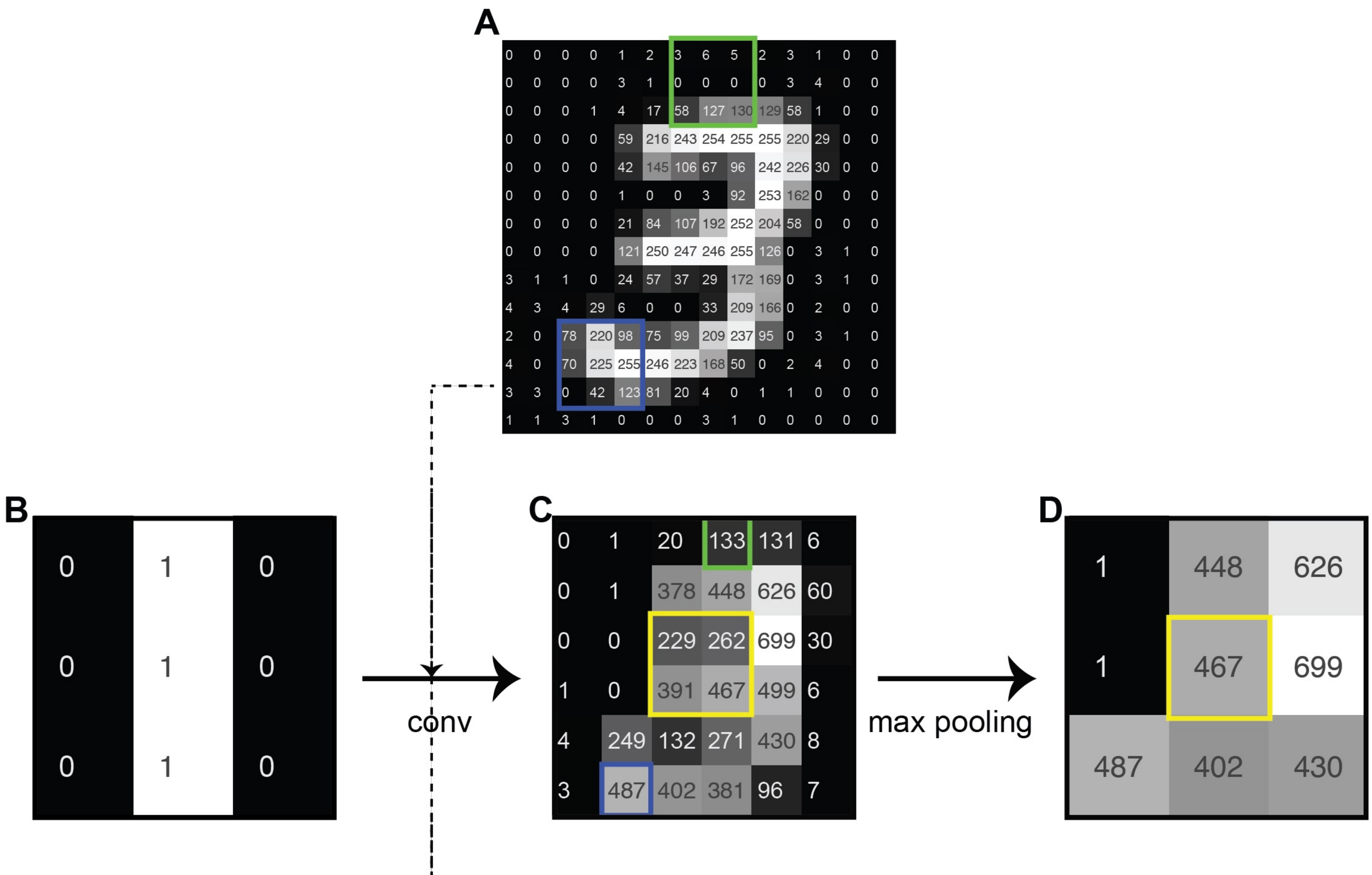
Source: Xelio Cheong (YouTube)

Multilayer networks and backpropagation

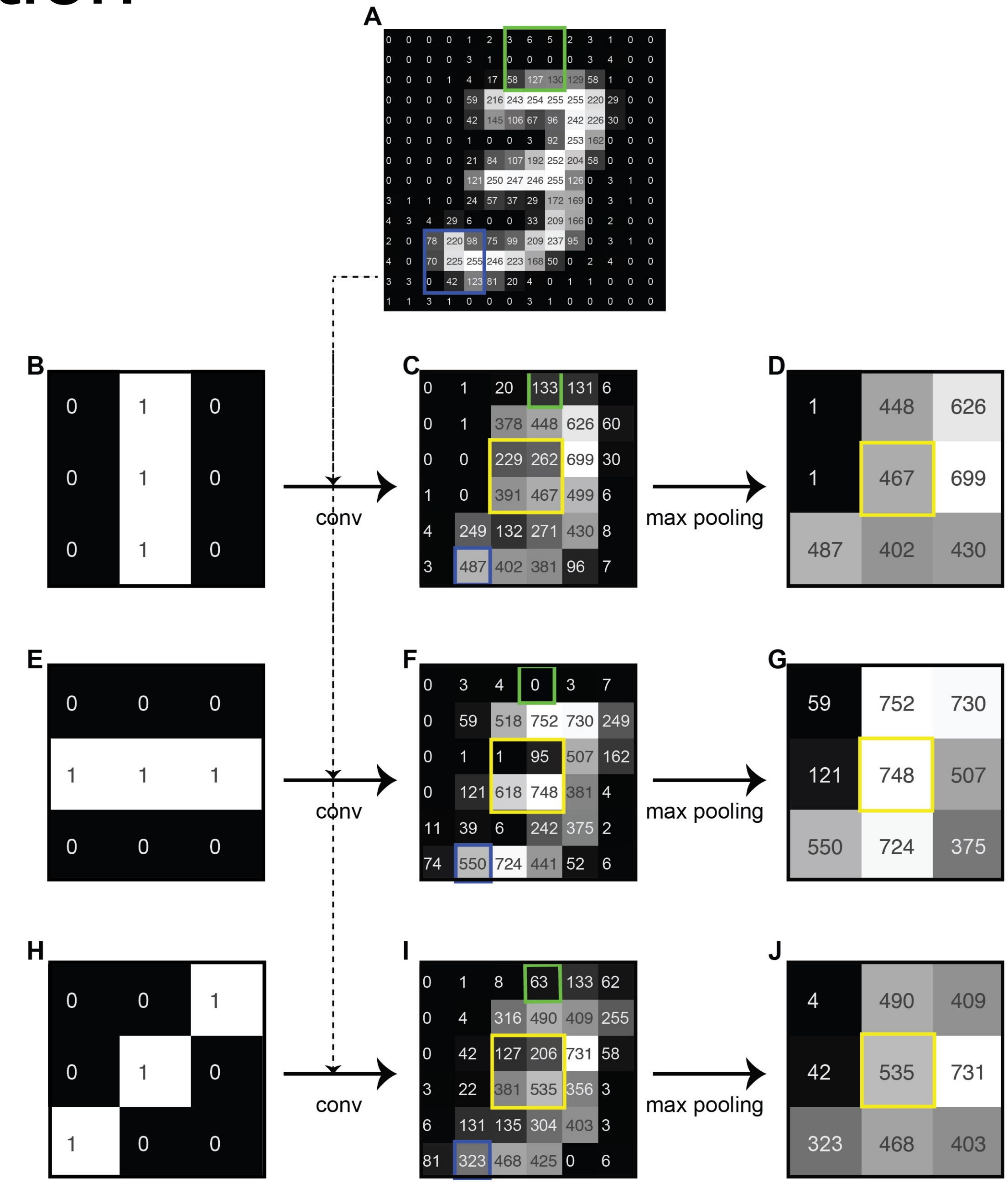


Source: [Matt Mazur]()

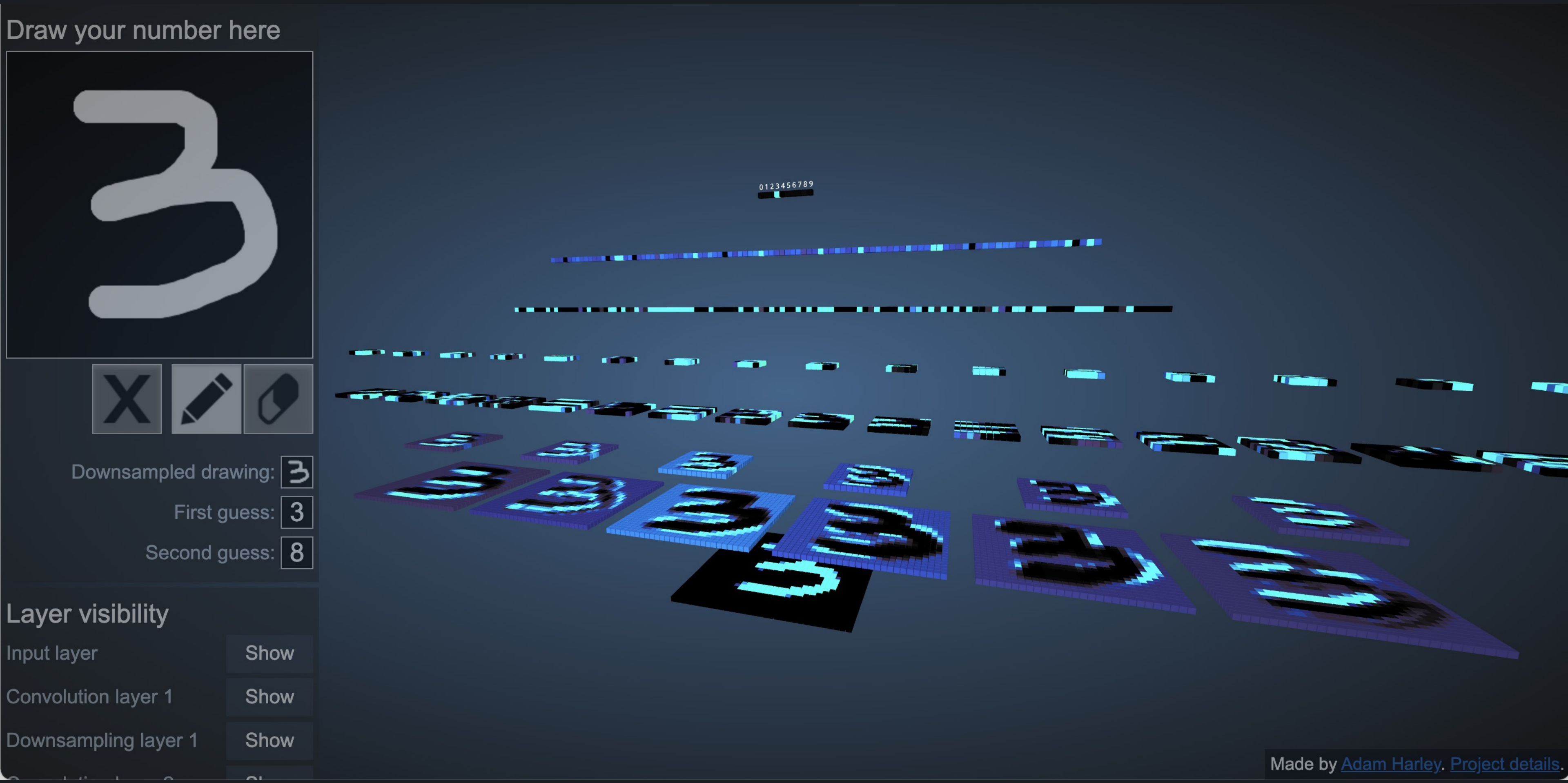
Convolution—
The workhorse of visual neural networks
(and a misnomer)



Convolution



Visualization of a conv net

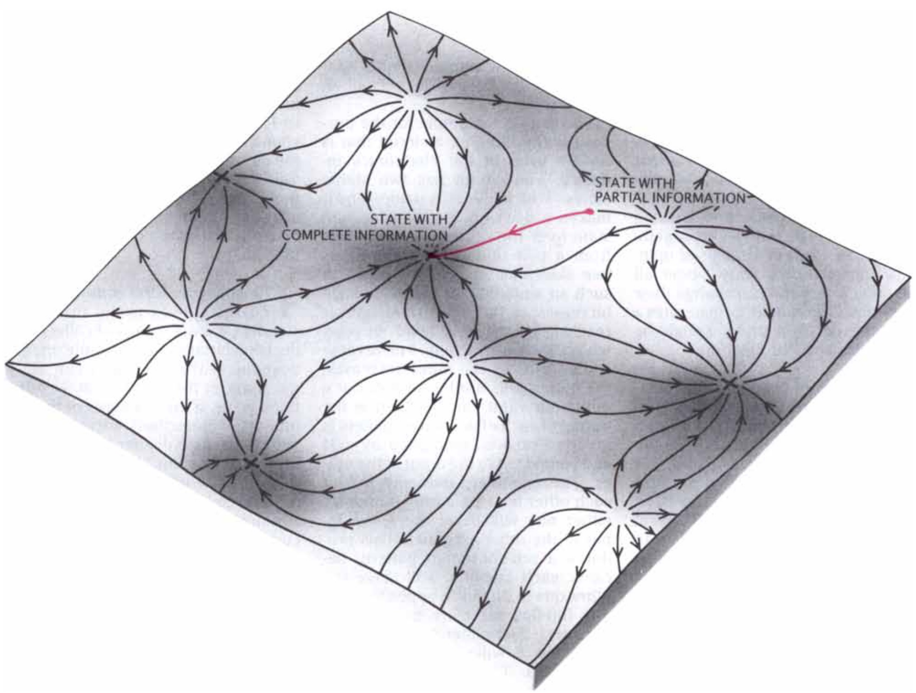


https://www.cs.ryerson.ca/~aharley/vis/conv/

Hopfield networks

$w_{ii} = 0$ No self connections

$w_{ij} = w_{ji}$ Symmetric connections



$\boldsymbol{s} = [s_1, \dots, s_N]$

State vector

$s_i[t+1] = sign(\sum_{j \neq i}^{N} w_{ij} s_j[t] - \theta)$

State update

$$E = -\frac{1}{2} \sum_{i,j} w_{ij} s_i s_j + \sum_i s_i \theta_i$$

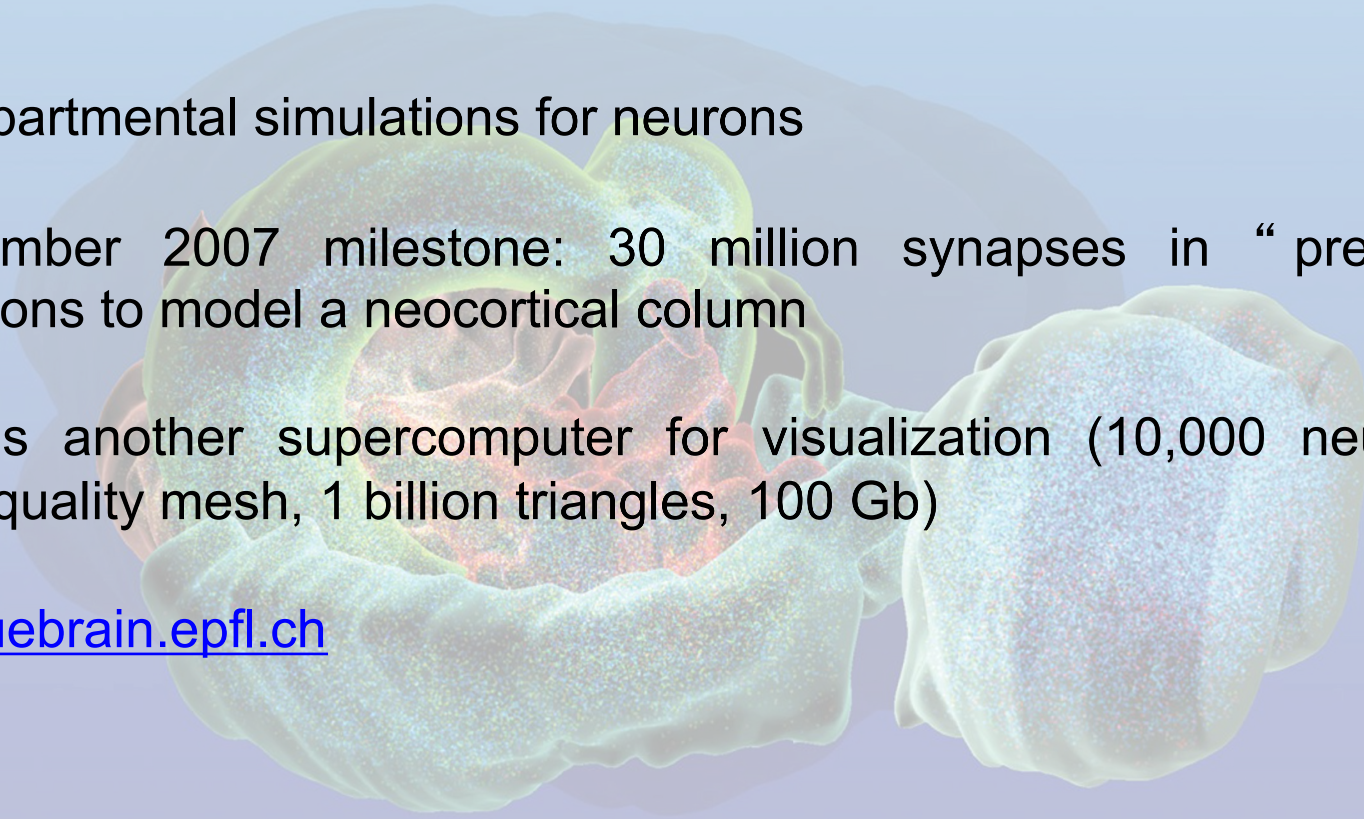
Energy function

$$w_{ij} = \frac{1}{n} \sum_{\mu=1}^{n} \epsilon_i^{\mu} \epsilon_j^{\mu}$$

Hebbian learning

Hopfield, 1982
Tank and Hopfield, 1987

The blue brain modeling project



- Compartmental simulations for neurons
- November 2007 milestone: 30 million synapses in “precise” locations to model a neocortical column
- Needs another supercomputer for visualization (10,000 neurons, high quality mesh, 1 billion triangles, 100 Gb)

http://bluebrain.epfl.ch

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

Summary

1. Why build models?
   - They represent understanding
   - They are useful (for testing, predicting, …)
2. Single neuron models
   - Capture varying levels of detail,
     from static to dynamic to multi-compartmental models
3. Network models
   - Supervised learning; perceptron and MNIST
   - (Backpropagation)
   - Convolution
   - Hopfield network