SAND2011-1228P

Neurons to Algorithms

Sandia team:
- Fred Rothganger
- Christy Warrender
- Ann Speed
- Wendy Shaneyfelt
- Steve Verzi
- John Battista

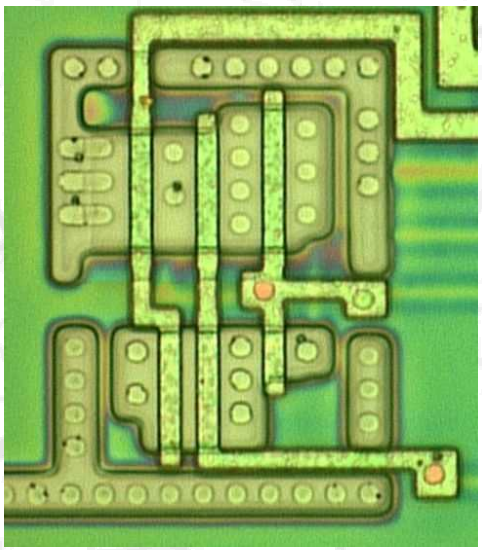
Outline:
- Problem & Motivation
- Approach
- Tasks

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

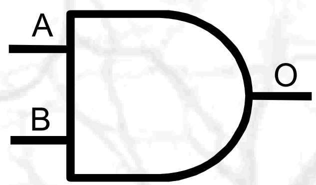
What problem are we trying to solve?

Describe the ***function*** of a select brain circuit.

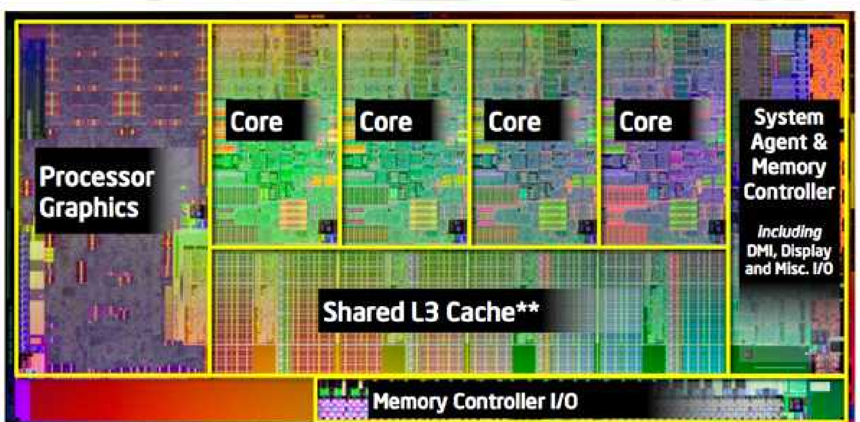
Abstract away physical details to explain what something does or how it interacts with other things.



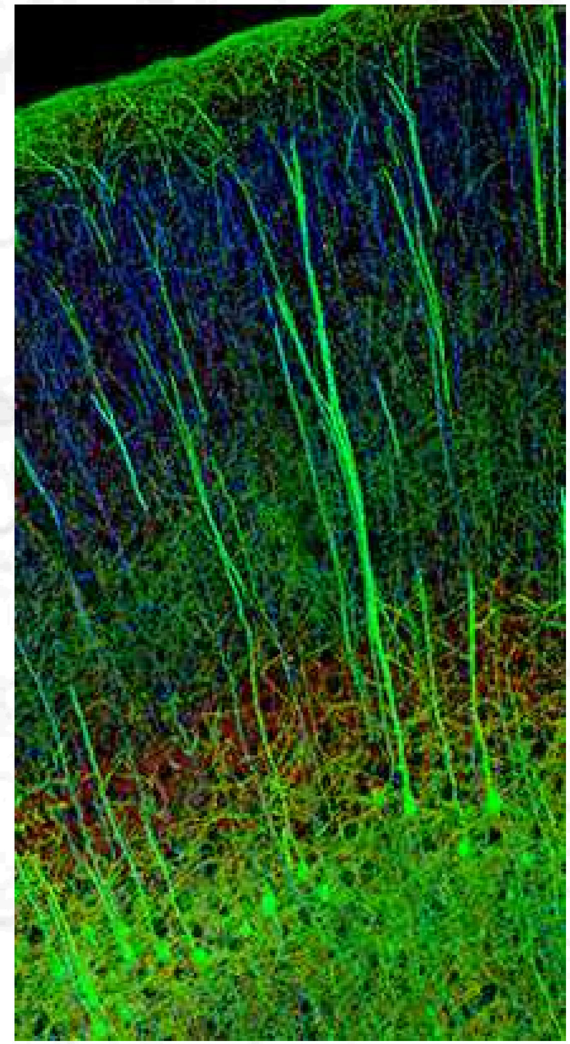
AND gate [Kömmerling & Kuhn 99]



A	B	O
F	F	F
F	T	F
T	F	F
T	T	T



Sandy Bridge die [Intel Corp.]

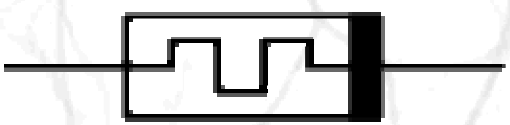


Neocortical (brain) slice
[Stephen Smith Lab at Stanford]

Why is this important?

Simple curiosity: we wish to know ourselves

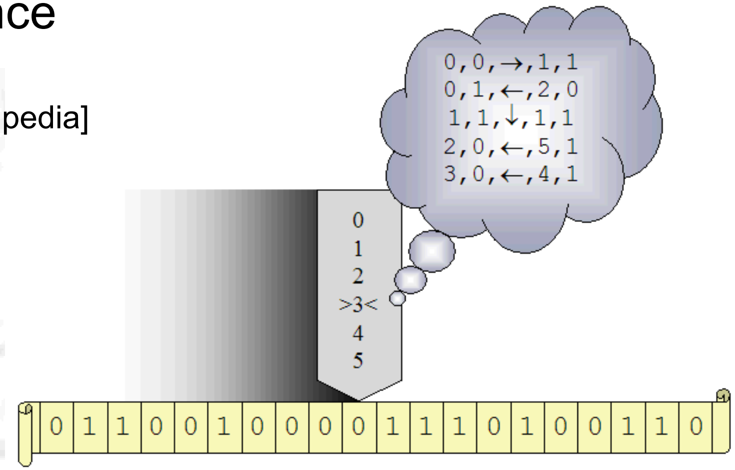
Applications: we wish to build artifacts with human levels of intelligence.



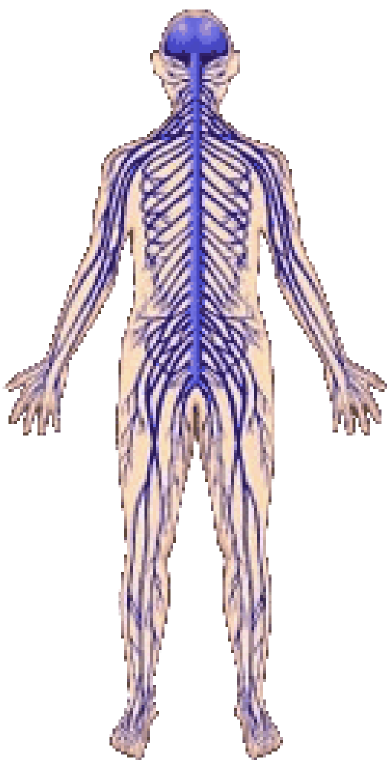
Exascale / Neuromorphic hardware: requires an understanding of the software.

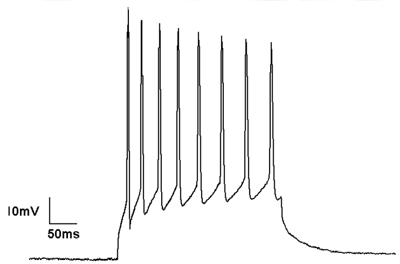
Algorithms as description

Algorithm = A set of rules that induce a sequence of operations on an array of natural numbers.
[modified from Wikipedia]



An algorithm can approximate real numbers.



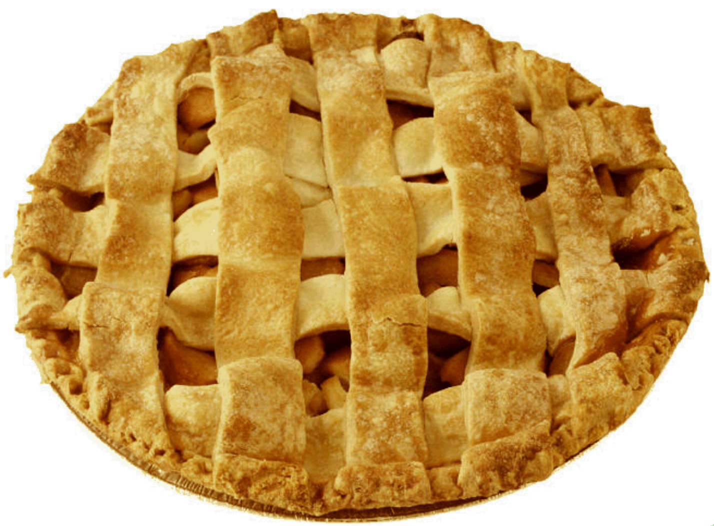


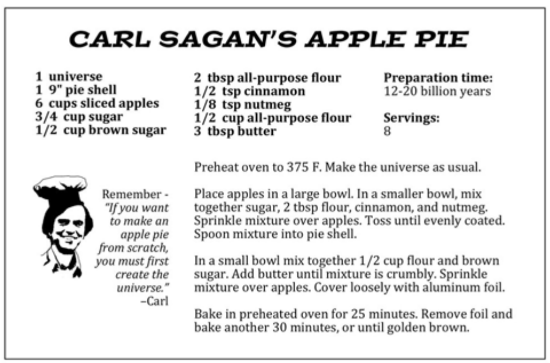
A sequence of operations on numerical values is an appropriate “language” in which to describe what the brain does.

- Discrete set of nerves entering and exiting the brain.
- Nerve signals can be expressed as a real-valued functions of time.
- Signal processing can be approximated.
- ***More abstract descriptions can take the form of operations on data structures.***

Algorithms as description

It's like … apple pie.





CARL SAGAN'S APPLE PIE

1 universe
1 9" pie shell
6 cups sliced apples
3/4 cup sugar
1/2 cup brown sugar

2 tbsp all-purpose flour
1/2 tsp cinnamon
1/8 tsp nutmeg
1/2 cup all-purpose flour
3 tbsp butter

Preparation time:
12-20 billion years

Servings:
8

Remember - *"If you want to make an apple pie from scratch, you must first create the universe."* –Carl

Preheat oven to 375 F. Make the universe as usual.

Place apples in a large bowl. In a smaller bowl, mix together sugar, 2 tbsp flour, cinnamon, and nutmeg. Sprinkle mixture over apples. Toss until evenly coated. Spoon mixture into pie shell.

In a small bowl mix together 1/2 cup flour and brown sugar. Add butter until mixture is crumbly. Sprinkle mixture over apples. Cover loosely with aluminum foil.

Bake in preheated oven for 25 minutes. Remove foil and bake another 30 minutes, or until golden brown.

An algorithm is a mathematically precise way to describe ***how to do*** the same thing that a neural circuit is doing.

Algorithms as differential equations

- Any problem in the complexity class *nondeterministic polynomial time* (NP) can be reduced to a Boolean expression [Cook-Levin theorem].
- Any Boolean expression can be reduced to differential equations:

AND: $$\frac{dC}{dt} = f(A \cdot B \cdot \ldots) - C$$

OR: $$\frac{dC}{dt} = f(A + B + \ldots) - C$$

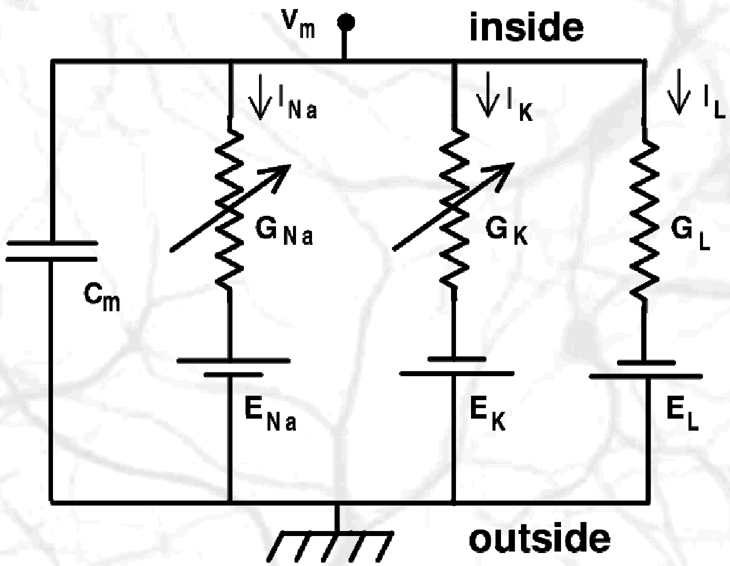
NOT: $$\frac{dC}{dt} = f(1 - A) - C$$

where $$f(x) = \begin{cases} 1 & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

and all initial inputs are either 1 or 0.

- All the problems the brain directly solves are likely to be in NP
  - Fixed amount of hardware.
  - Very limited time, so very limited iteration.
  - Approximate solutions to problems from higher classes may exist.
  - Unlikely that brain possesses an “oracle” proposing the correct solution.

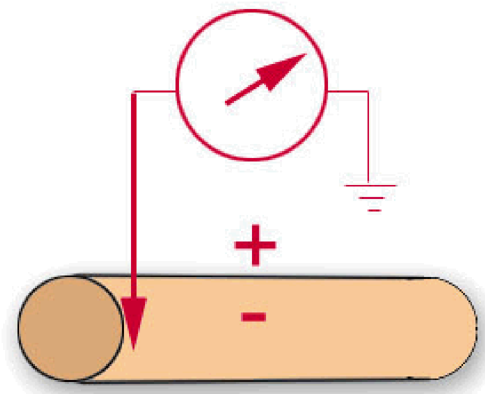
Neurons as differential equations

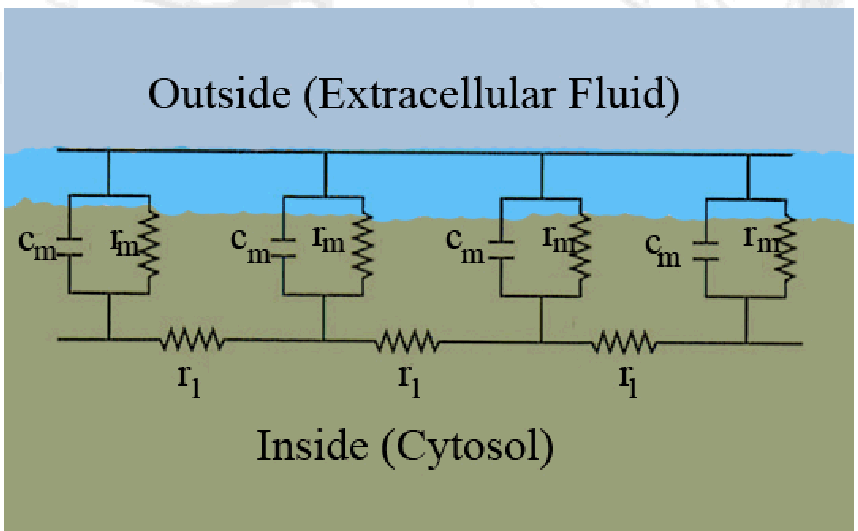


Hodgkin-Huxley:

$$C\frac{dV}{dt} = G_{Na}(V - E_{Na}) + G_K(V - E_K) + G_L(V - E_L)$$

where each $G_{\square}$ is a function of V and time. Specifically, each $G_{\square}$ decays exponentially towards a value characteristic of the given voltage V.





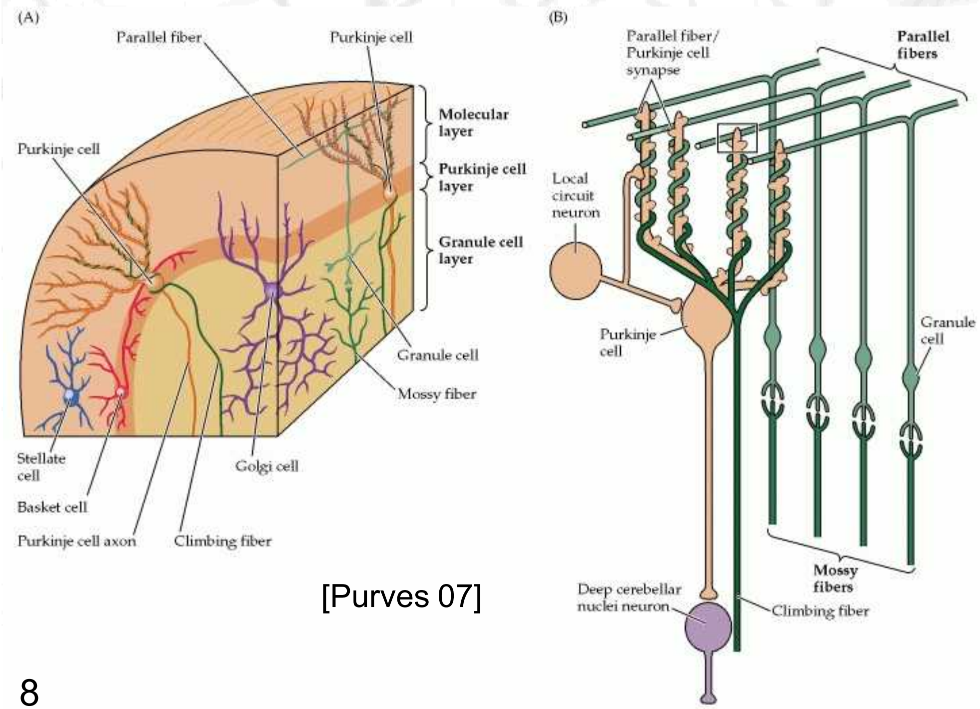
Cable equation:

$$C\frac{dV_i}{dt} = G(V_i - E) + \frac{1}{R}(V_{i+1} - 2V_i + V_{i-1})$$

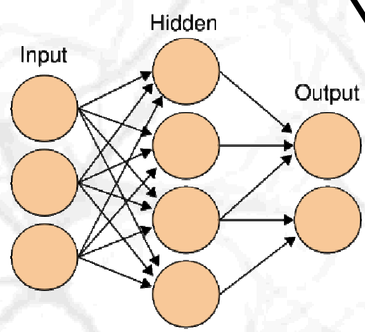
Connecting Neurons to Algorithms

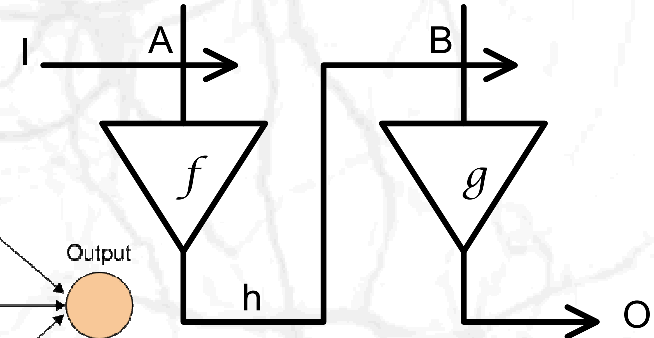
Neural networks – A useful level of abstraction where the differential equations of neurons and algorithms take on comparable forms

- State of each neuron represented by only one or very few variables (“point neurons”).
- State of algorithm represented as vectors of real values.
- Most operations expressed as a multivariate linear function composed with a non-linear “squashing” function: **O**=*f*(**Ax**+**b**)



[Purves 07]



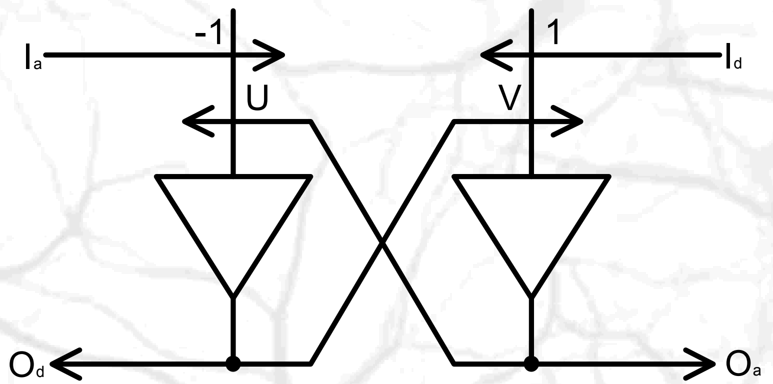


$$\tau \frac{d\mathbf{h}}{dt} = f(\mathbf{AI}) - \mathbf{h}$$

$$\tau \frac{d\mathbf{O}}{dt} = g(\mathbf{Bh}) - \mathbf{O}$$

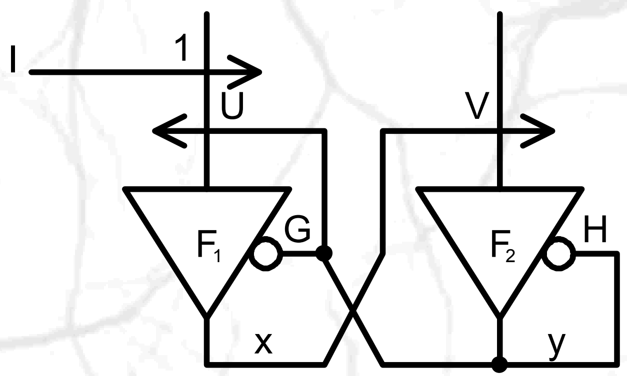
Algorithm Examples

Kalman Filter [Rao & Ballard 1999]



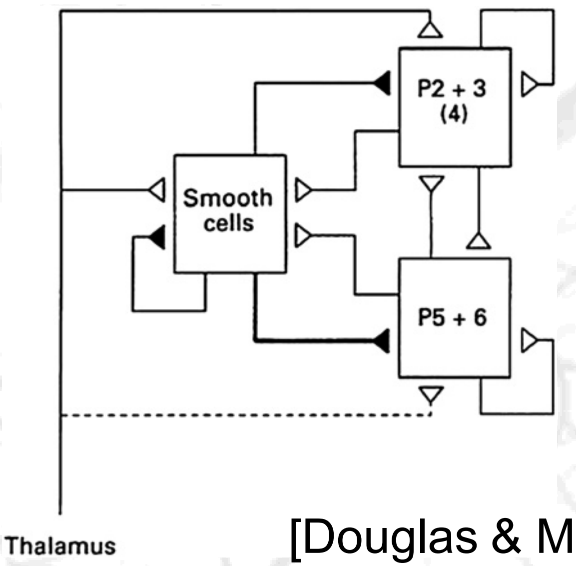
$$\mathbf{O}_d \leftarrow \mathbf{U}\mathbf{O}_a - \mathbf{I}_a$$
$$\mathbf{O}_a \leftarrow \mathbf{V}\mathbf{O}_d + \mathbf{I}_d$$
$$\mathbf{U} \leftarrow (1-L)\mathbf{U} + L\mathbf{O}_d\mathbf{O}_a^T$$
$$\mathbf{V} \leftarrow (1-L)\mathbf{V} + L\mathbf{O}_a\mathbf{O}_d^T$$

Online Clustering (ART) [Carpenter and Grossberg 1987]

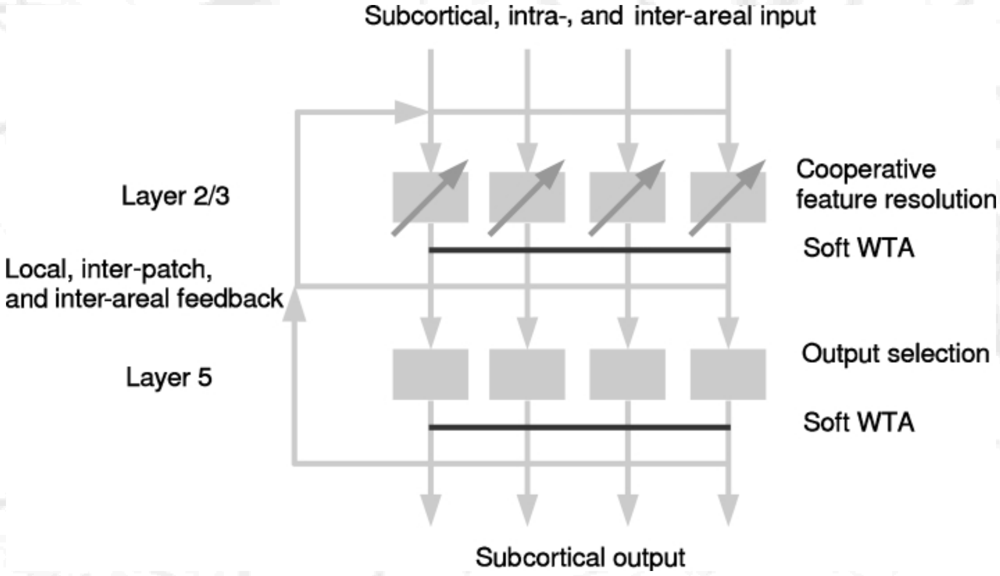


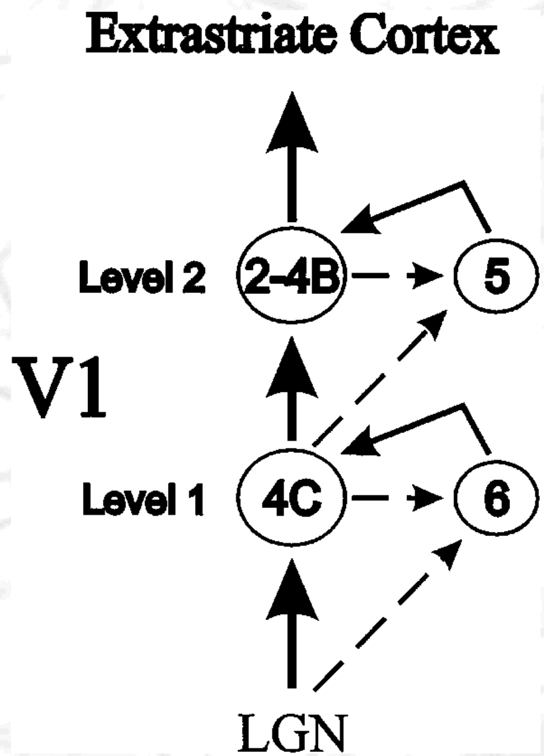
$$\mathbf{x} \leftarrow (1-L)\mathbf{x} + L[\mathbf{I} + \mathbf{U}\mathbf{y} - \mathbf{G}\mathbf{y}]$$
$$\mathbf{y} \leftarrow \mathbf{y} + L[\mathbf{V}\mathbf{x} - \mathbf{H}\mathbf{y}]$$
$$\mathbf{V} \leftarrow (1-\mathbf{y}) \circ \mathbf{V} + \mathbf{y}\mathbf{x}^T$$
$$\mathbf{U} \leftarrow \mathbf{U} \circ (1-\mathbf{y}) + \mathbf{x}\mathbf{y}^T$$

Bottom-Up Models by Physiologists



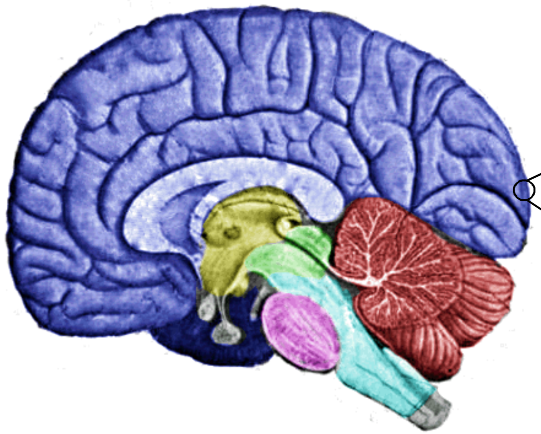
[Douglas & Martin]

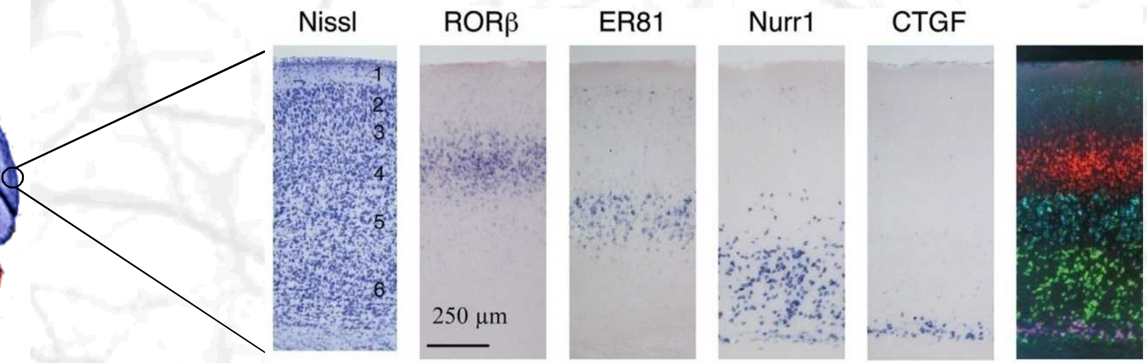




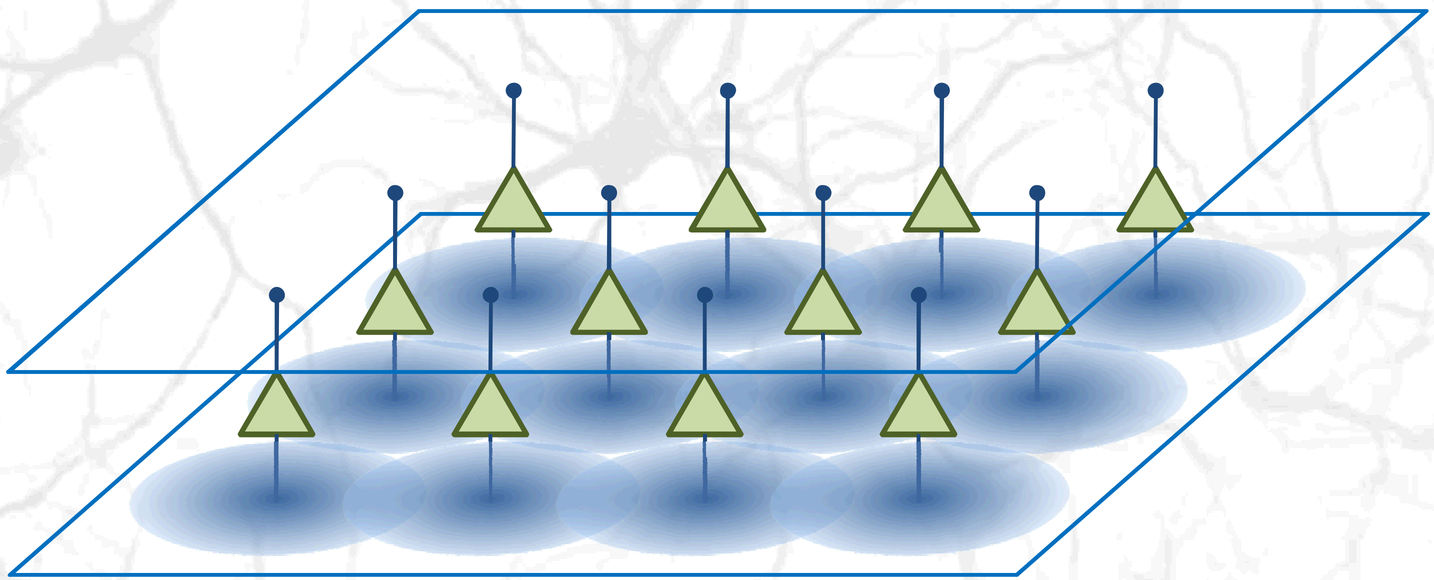
[Callaway]

Spatial structure of Neurons

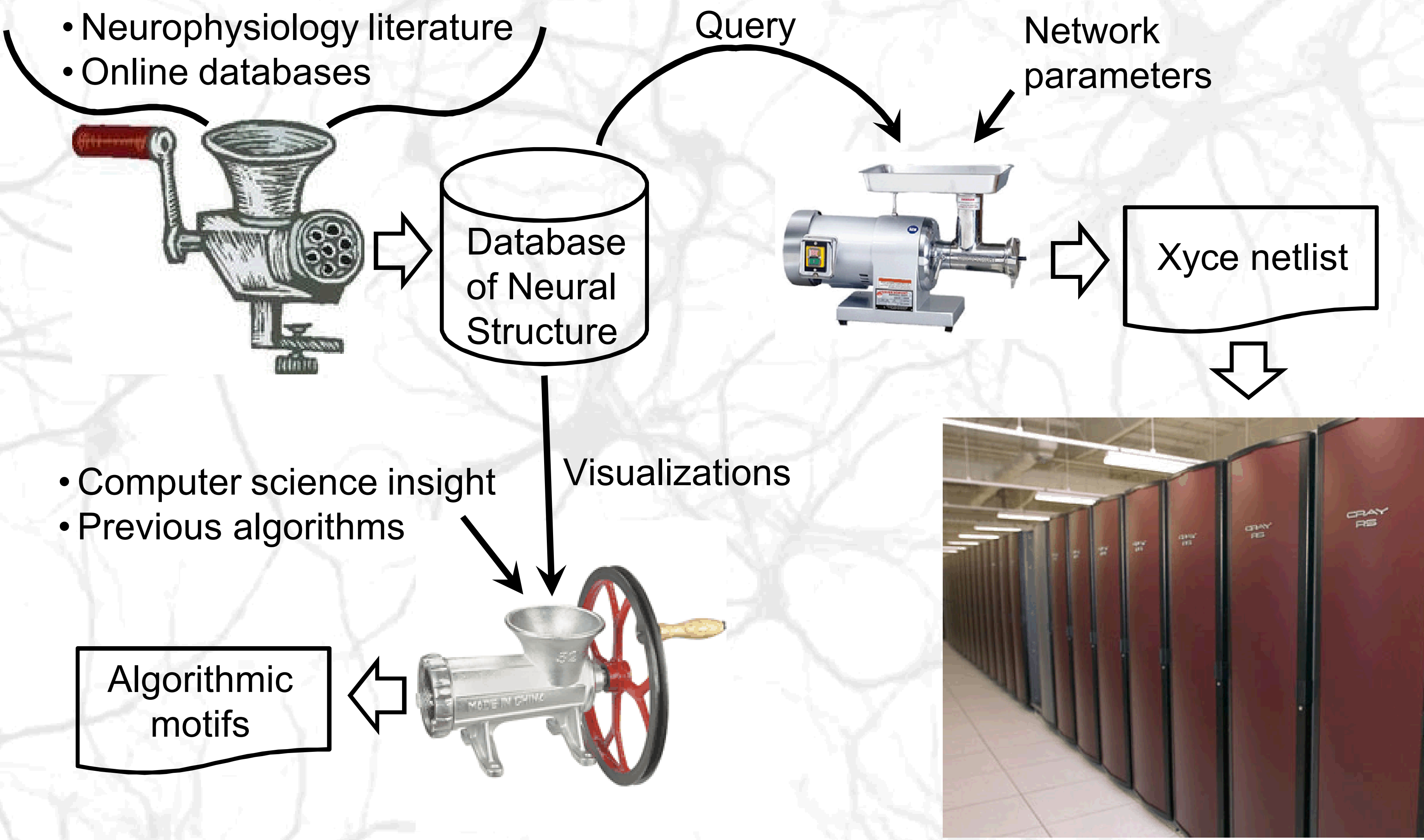




[Watakabe 09]

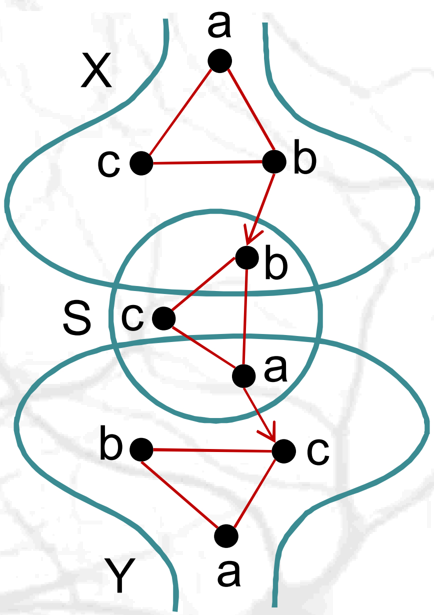


Finding the Algorithmic Needle in the Neural Haystack



Neural structure database

Some challenges…



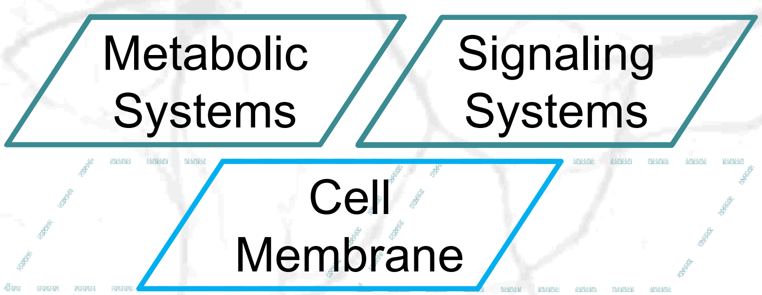
$S.b \leftarrow 0.97 * X.b$

$Y.c \leftarrow \sum_{S} S.a$

etc…

Computationally explicit

- Other available neural DBs rarely specify neural structure in terms of variables and equations.
- Varying specificity in the literature.
- Some sources may contradict each other.



Structures at different levels of abstraction

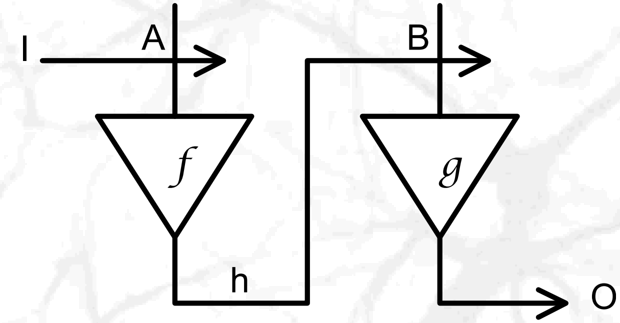
- No strict hierarchical decomposition
- Connectivity may cross scale levels

How to query?

- Need sufficient closure to make working model.

Questions or Comments?

Shorthand Notation



Differential

$$\tau \frac{d\mathbf{h}}{dt} = f(\mathbf{AI}) - \mathbf{h}$$

$$\tau \frac{d\mathbf{O}}{dt} = g(\mathbf{Bh}) - \mathbf{O}$$

Difference

$$\mathbf{h}_{t+1} = (1 - \frac{\Delta t}{\tau})\mathbf{h}_t + \frac{\Delta t}{\tau} f(\mathbf{AI}_t)$$

$$\mathbf{O}_{t+1} = (1 - \frac{\Delta t}{\tau})\mathbf{O}_t + \frac{\Delta t}{\tau} g(\mathbf{Bh}_t)$$

Update

$$\mathbf{h} \leftarrow (1 - \frac{\Delta t}{\tau})\mathbf{h} + \frac{\Delta t}{\tau} f(\mathbf{AI})$$

$$\mathbf{O} \leftarrow (1 - \frac{\Delta t}{\tau})\mathbf{O} + \frac{\Delta t}{\tau} g(\mathbf{Bh})$$

Implicit squashing functions

$$\mathbf{h} \leftarrow (1 - \frac{\Delta t}{\tau})\mathbf{h} + \frac{\Delta t}{\tau} \mathbf{AI}$$

$$\mathbf{O} \leftarrow (1 - \frac{\Delta t}{\tau})\mathbf{O} + \frac{\Delta t}{\tau} \mathbf{Bh}$$

Use “L” for “learning rate” or decay rate

$$\mathbf{h} \leftarrow (1 - L)\mathbf{h} + L\mathbf{AI}$$

$$\mathbf{O} \leftarrow (1 - L)\mathbf{O} + L\mathbf{Bh}$$

If τ=t

$$\mathbf{h} \leftarrow \mathbf{AI}$$

$$\mathbf{O} \leftarrow \mathbf{Bh}$$