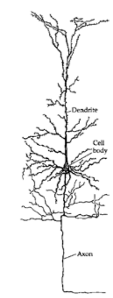
The Neuron

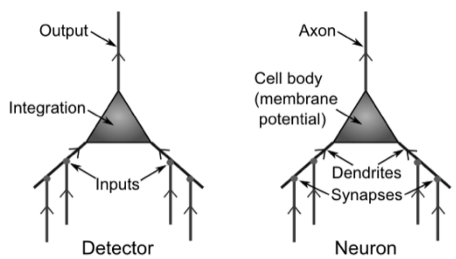
Computational Cognitive Neuroscience

Randall O'Reilly

The Basic Unit of Cognition!?



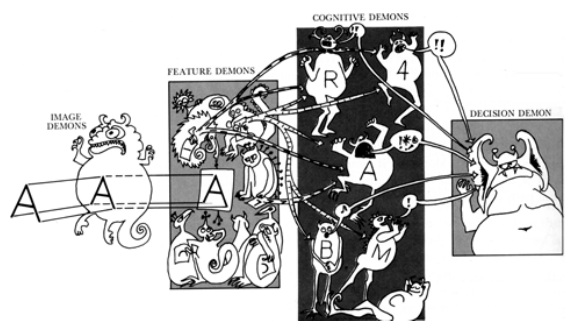
Detector Model



Is it really all just detection?

Pandemonium!

(Oliver Selfridge)



Feature Demons

1. Vertical Line: |
2. Horizontal Line: --
3. Up-Right Diagonal: /
4. Up-Left Diagonal: \

Cognitive Demons

5. T: 1,2
6. V: 3,4
7. A: 2,3,4
8. K: 1,3,4

Testing..

Testing..

Testing..

Testing..

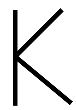
Testing..

T

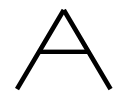
Testing..

A

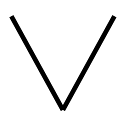
Testing..



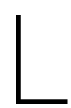
Testing..



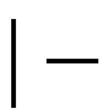
Testing..



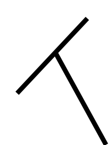
Ooops..



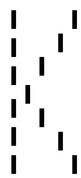
Ooops..



Ooops..



Ooops..



Pandemonium Summary

- Maybe you can see how collective action of many detectors, organized *hierarchically*, could achieve more complex cognition?
- But detection needs to be a lot more sophisticated..

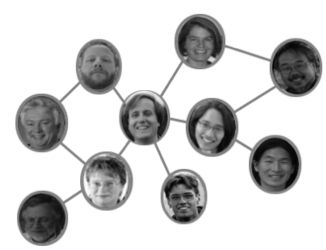
Neurons in the Dark

- Neurons live in the dark!
- “Hear” an incredible jumble of inputs.
- Have *absolutely no idea* what is going on in the real world outside their little area of the brain..

All of this is very counterintuitive given that we tend to think of neurons as communicating in full English sentences about the weather, etc..

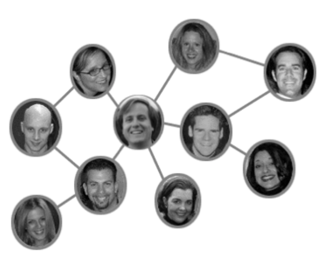
Neurons only get spikes, not words!

The Social Network



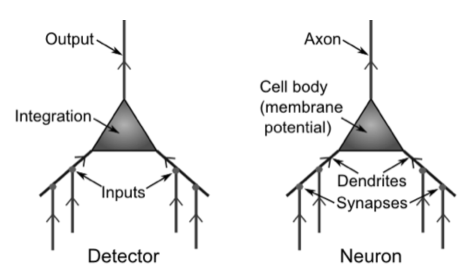
Neurons depend on network of “trust” built up over a long time period – only way they can overcome the jumble in the dark..

The Social Network



How do neurons ever know if senders change what they are encoding? How does the brain ever change?

Back to the Detector Model

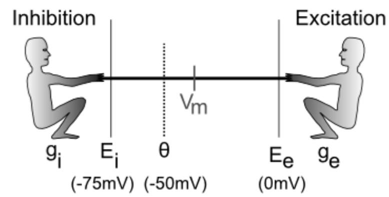


How do we simulate on a computer?

Overall Strategy

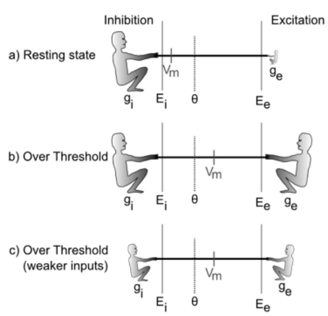
- Neurons are electrical systems, can be described using basic electrical equations.
- Use these equations to simulate on a computer.
- Need a fair bit of math to get a full working model (more here than most chapters), but you only really need to understand *conceptually*.

The Tug-of-War

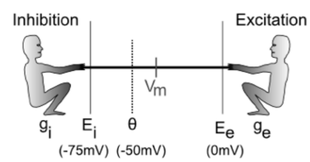


How strongly each guy pulls: I = g (E-Vm)
g = how many input channels are open
E = driving potential (pull down for inhibition, up for excitation)
Vm = the "flag" – reflects net balance between two sides

Relative Balance..



Equations..

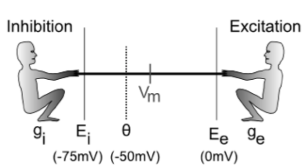


$$I_{net} = I_e + I_i + I_l = g_e\left(E_e - V_m\right) + g_i\left(E_i - V_m\right) + g_l\left(E_l - V_m\right)$$

$$V_m\left(t\right) = V_m\left(t-1\right) + dt_{vm} I_{net}$$

$$V_m(t) = V_m(t-1) + dt_{vm}\left[g_e(E_e - V_m) + g_i(E_i - V_m) + g_l(E_l - V_m)\right]$$

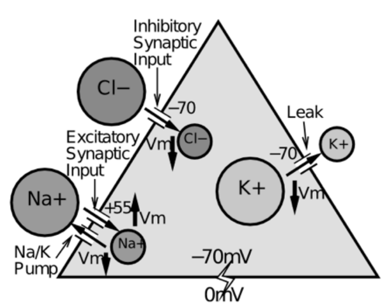
Equilibrium



$$V_m = \frac{g_e}{g_e + g_i + g_l} E_e + \frac{g_i}{g_e + g_i + g_l} E_i + \frac{g_l}{g_e + g_i + g_l} E_l$$

This is just the balance of forces..

The Full Story..



Input Conductances and Weights

- Just add 'em up (and take the average)

$$g_e(t) = \frac{1}{n}\sum_i x_i w_i$$

- Key concept is *weight:* how much unit listens to given input
- Weights determine what the neuron detects
- Everything you know is encoded in your weights..

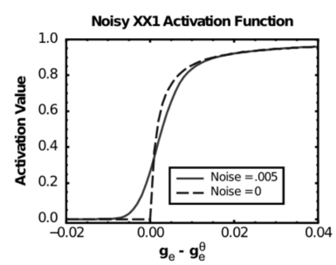
Generating Output

- If Vm gets over threshold, neuron fires a spike.
- Spike resets membrane potential back to rest.
- Has to climb back up to threshold to spike again

Rate Code Approximation

- Brain likes spikes, but rates are great!
  - Instantaneous and steady – smaller, faster models
  - But definitely lose several important things
  - Soln: do it both ways, and see what the diffs are..
- Goal: equation that makes good approx of actual spiking rate for same sets of inputs.

Sigmoidal Activation



- Threshold
- Saturating
- Smooth

Rate Code Equations

- A little bit tricky because Vm doesn't work.
- Need to use excitatory conductance – threshold
- XX1 equation: $y = \frac{1}{\left(1 + \frac{1}{\gamma[g_e - g_e^\Theta]_+}\right)}$
- ge-theta: $g_e^\Theta = \frac{g_i(E_i - \Theta) + g_l(E_l - \Theta)}{\Theta - E_e}$
- Tracking Vm timecourse: $g_e^* = \left(\frac{V_m - V_m^{rest}}{0.95(V_m^\Theta - V_m^{rest})}\right) g_e$