Corporate R&D

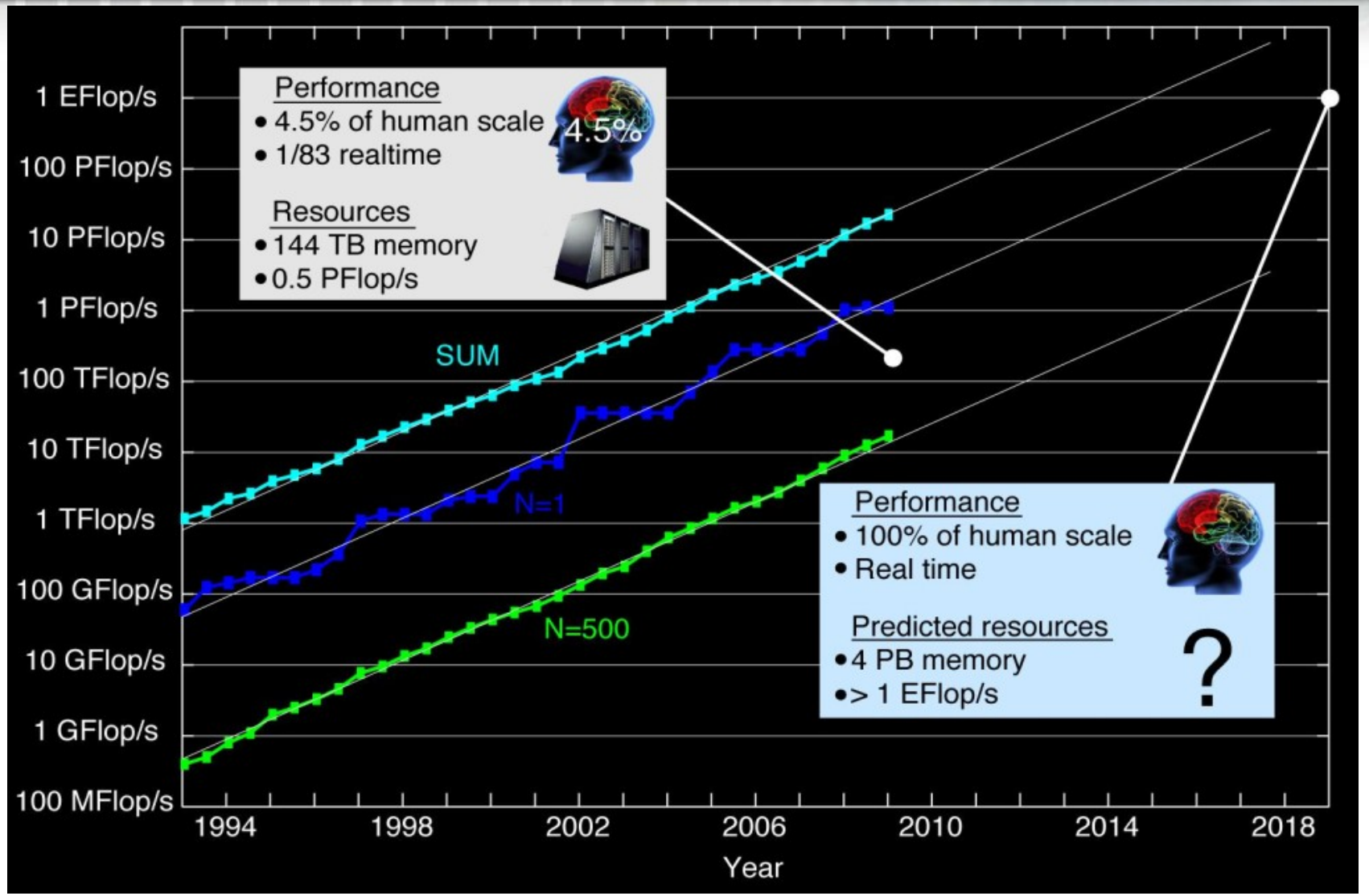
Biologically Inspired Computing

System, Hardware and Network Challenges

Venkat Rangan

Outline

- Motivation
- Biology/Neuroscience
- Computational Abstractions
- Tradeoffs and Challenges



Green: 500th fastest supercomputer, dark blue: fastest supercomputer, light blue: sum of top 500 supercomputers: D.S.Modha's Cognitive Computing blog

Object Recognition Video

- Watch this movie (Courtesy Irving Biederman)
  - 0-15s: When you see a knife, please yell “knife”.
  - 16-33s: When you see a gorilla, please yell “gorilla”.

How do you do it?

- Individual neurons are slow compared to a CPU
  - Max output frequency < 1000 Hz; avg. frequency ~ 1 Hz.
- Yet object recognition is fast and efficient
  - < 84ms (70ms object + 14ms blank) object exposure is sufficient for recognition.
  - < 5 processing stages are sufficient for basic object recognition.
  - Limited integration time is necessary for each stage of processing.
  - Object database is large

Why Imitate Biology?

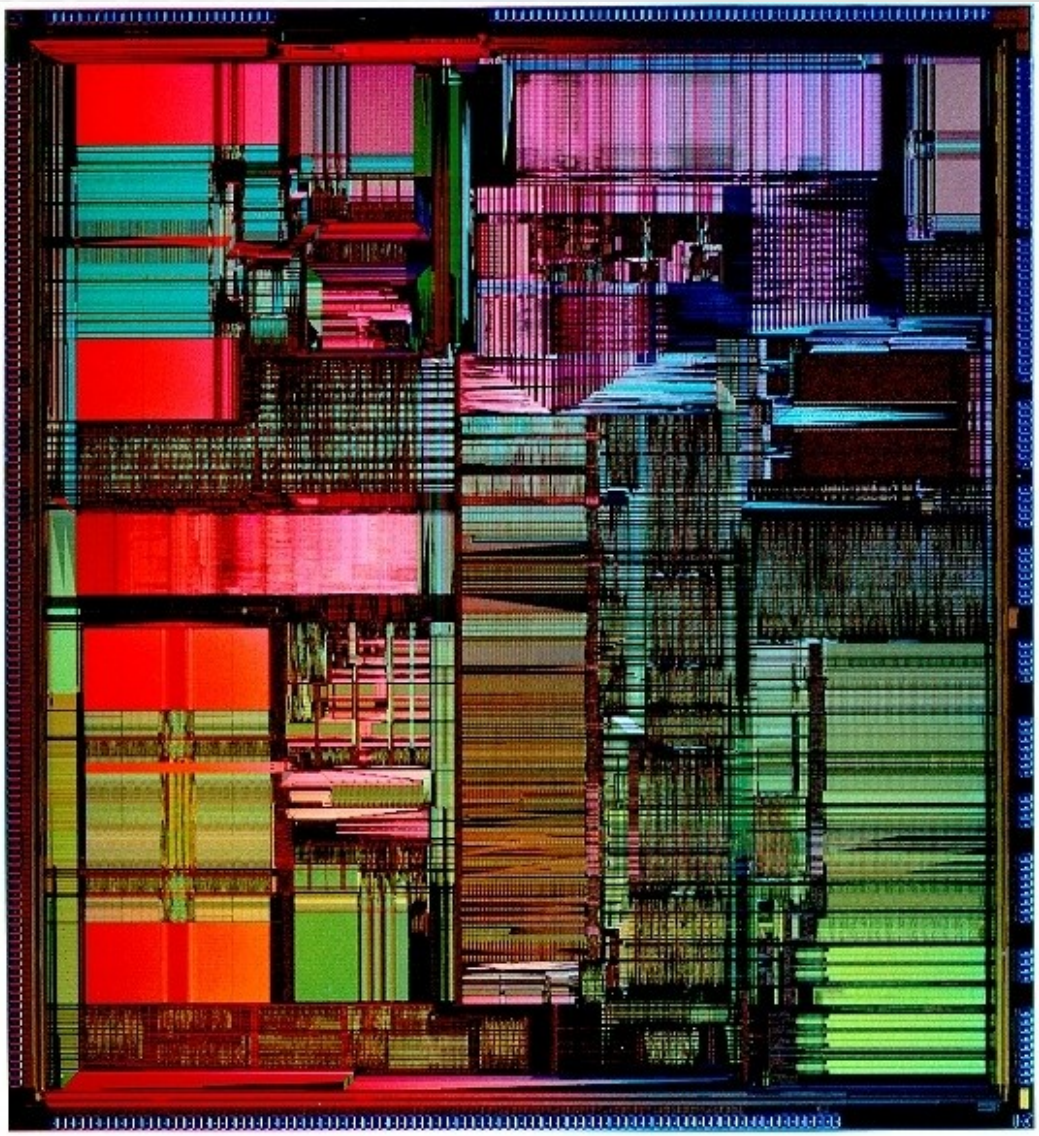
- Biological brains operate on real life problems
- Semiconductor issues & Evolution
  - Unreliable operation of circuits on very large chips with decreasing feature sizes
    - Brain architecture is extremely fault and damage tolerant
  - Algorithm development and software are always “long poles”
    - Brains are adaptive and self-learning. Non-learning systems are brittle
    - Intent is to have the machine learn it’s own solutions, we don’t always care about knowing the exact algorithm or the most optimal solution.
    - The machine will adapt as tasks or environment change
  - Fundamentally different approach of enabling the network to figure out patterns and make predictions rather than doing it manually with software

Motivation

- Brains are better at most real life tasks than current computers
- Silicon density on an exponential curve
- Advances in Computational and Experimental Neuroscience over the past decade
- Qualcomm has a strong background in semiconductor technology. Combined with a strong neuroscience team (Brain Corporation), we hope to answer some of the issues outlined in this talk

Neuroscience Today

- A typical lab: video
- An analogy:
  - How many probes do you need to see what this processor is doing?



© Intel Pentium pro processor core: 3.3 Million transistors

Similar Projects

FACETS
Fast Analog Computing with Emergent Transient States

BRAINS IN SILICON
STANFORD UNIVERSITY

Blue Brain Project

SpiNNaker
a universal Spiking Neural Network architecture

LABS hp

BRAIN CORPORATION

Harnessing the Power of Neuroscience Research

IBM

QUALCOMM

HRL LABORATORIES

Numenta

UCSD

EM Evolved Machines

All symbols are trademarks of respective companies/organizations

Outline

- Motivation
- Biology/Neuroscience
- Computational Abstractions
- Tradeoffs and Challenges

Neuroscience 101

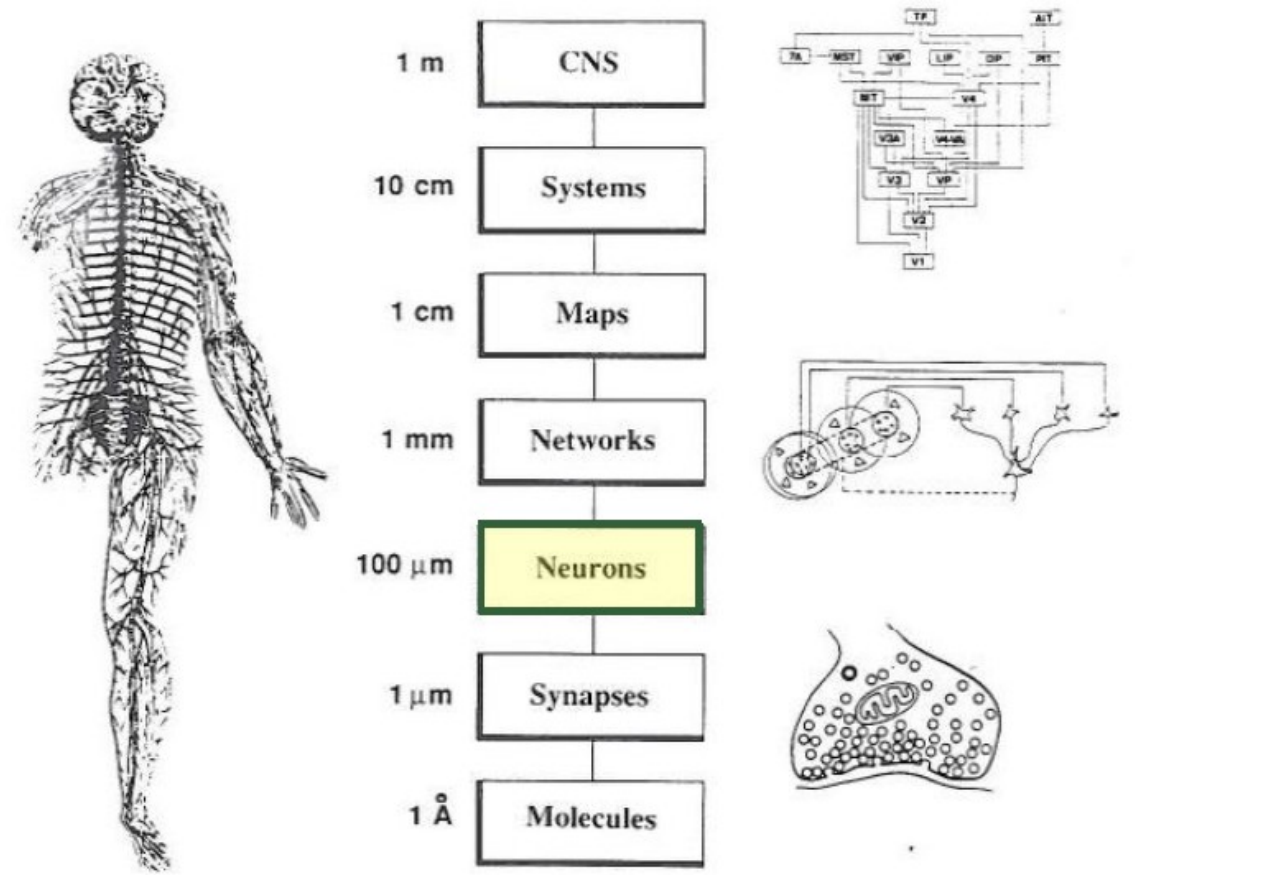
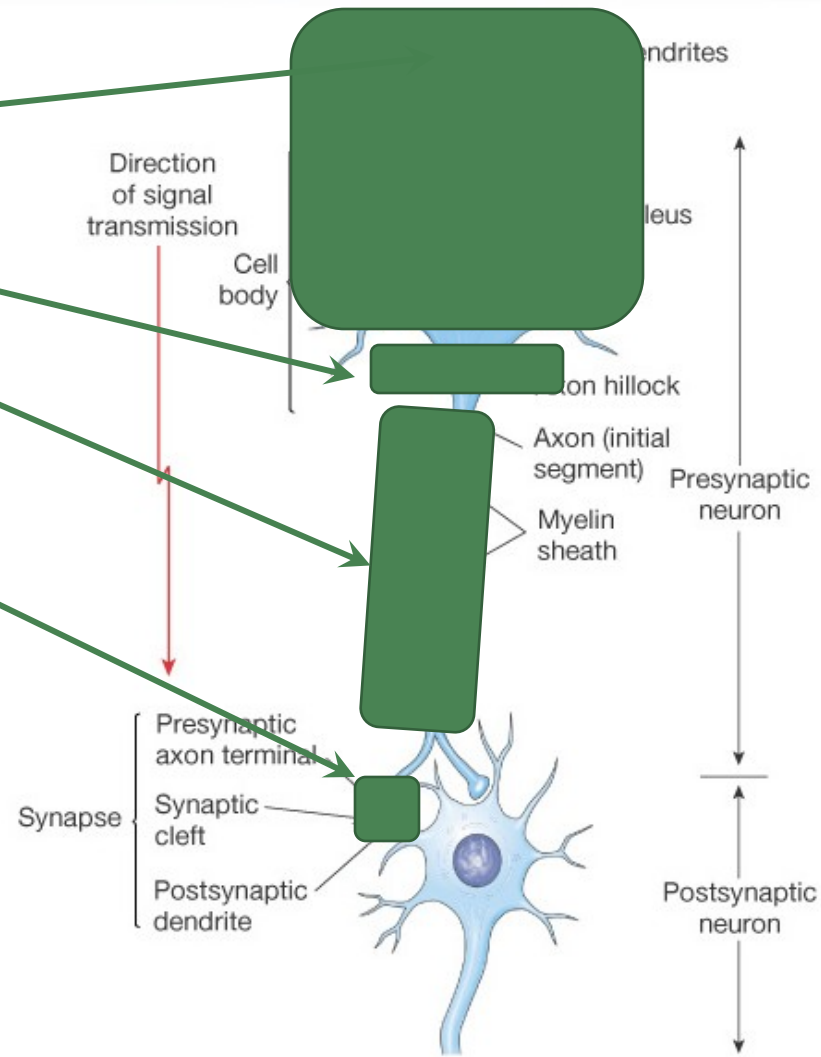
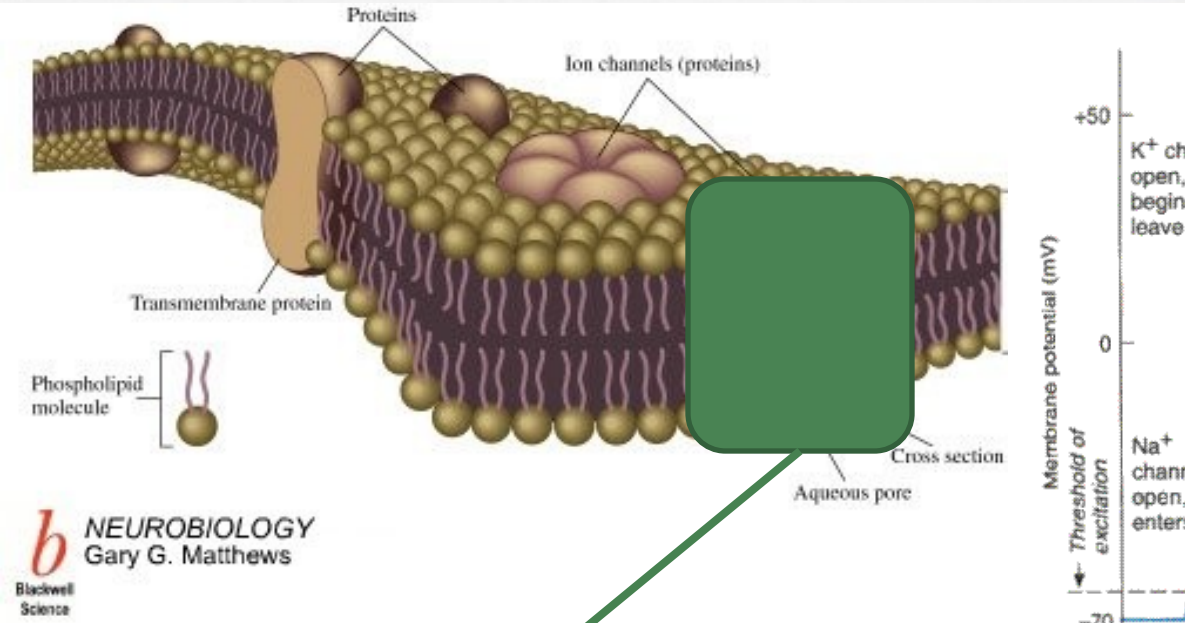


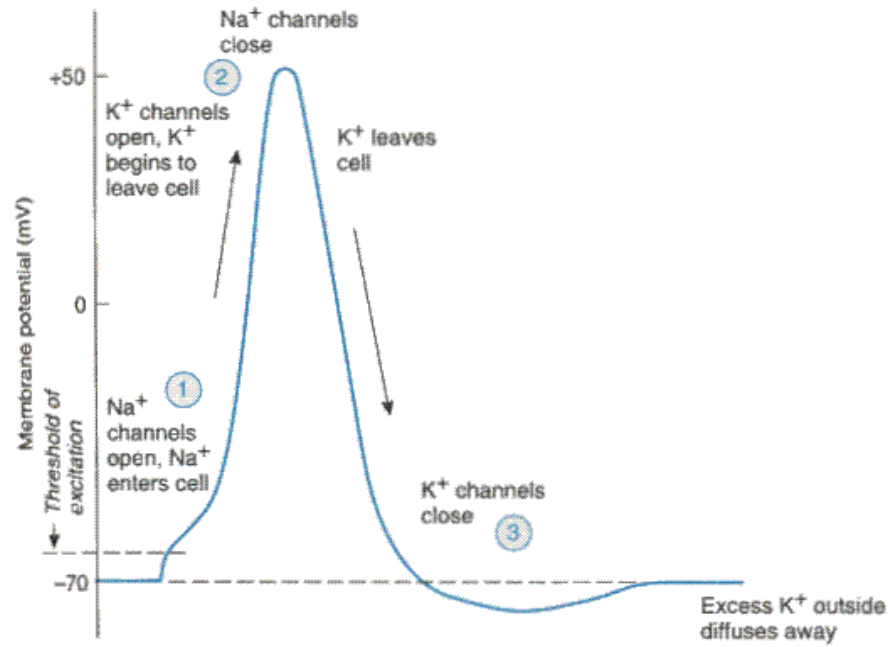
Figure 1.4 Schematic illustration of levels of organization in the nervous system. The spatial scales at which anatomical organizations can be identified varies over many orders of magnitude. Icons to the right represent structures at distinct levels: (top) a subset of visual areas in visual cortex (van Essen and Maunsell 1980); (middle) a network model of how ganglion cells could be connected to simple cells in visual cortex (Hubel and Wiesel, 1962), and (bottom) a chemical synapse (Kandel and Schwartz, 1985). (From Churchland and Sejnowski 1988.)

- Whats in a brain?
  - Neurons
    - Dendrites/Soma (Non-linear summation)
    - Axon hillock (ADC)
    - Axon (Connectivity/Delay)
  - Synapses
    - Memory
    - Plasticity
    - 1,000-10,000 synapses per neuron
  - Neurotransmitters/modulators
  - Vascular and other support circuits
- The Human brain has an estimated 10^{12} neurons and 10^{15} synapses.
  - A cockroach (1 million neurons) is capable of interesting behavior ☺



Membranes and Channels





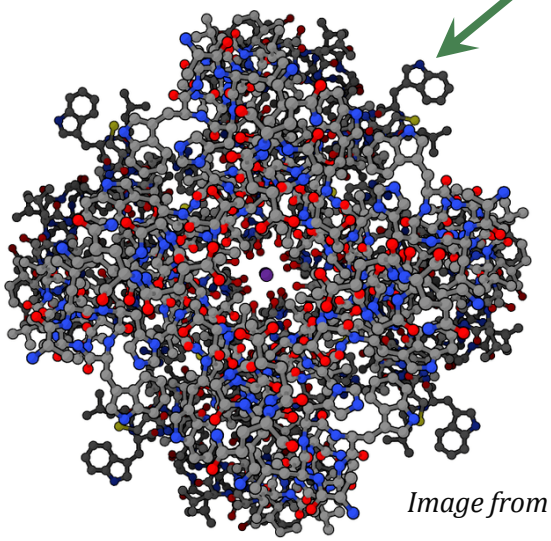


Image from Wikipedia

- Neuronal processes governed by channels (pores) in membranes
- Ion Pumps create an electrical potential across the membrane
- Pores can be opened/closed by various triggers: chemical, mechanical, heat, voltage

Outline

- Motivation
- Biology/Neuroscience
-
- Tradeoffs and Challenges

The importance of timing

- Time(Delay) is important
  - Leads to combinatorial explosion
  - Intractable for analytical solutions
- Brain possibly uses some combination of rate coding and spike timing

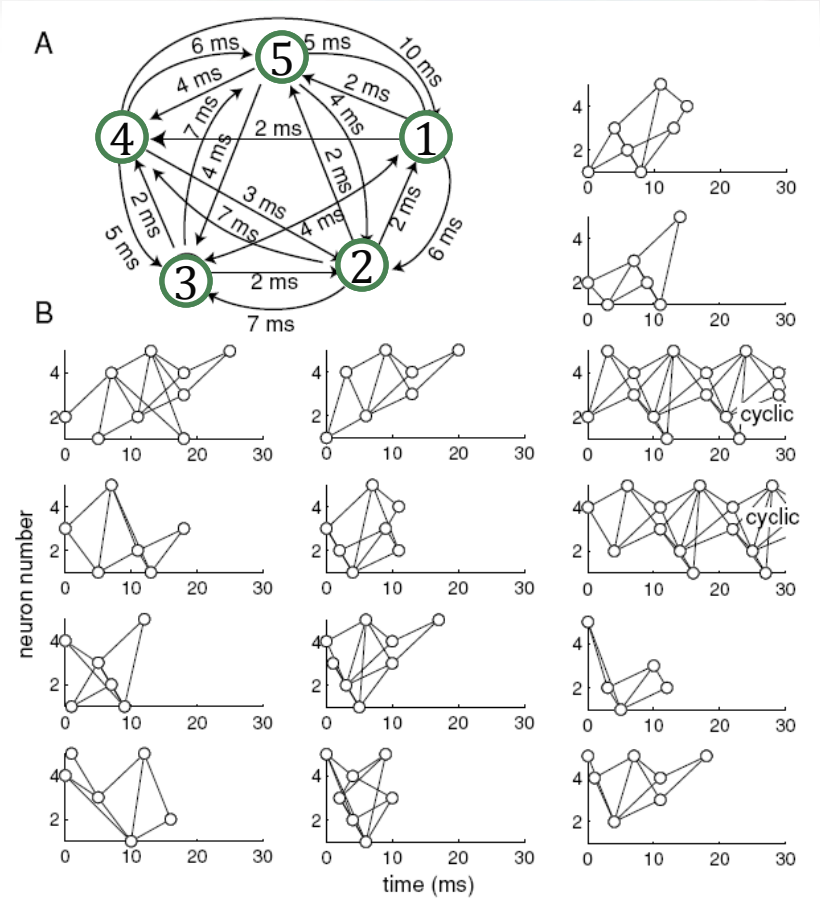


Figure 10: (A) A toy network of five neurons with axonal conduction delays and strong synapses (two simultaneous spikes is enough to fire any neuron). (B) The delays in the network are optimized so that it has 14 polychronous groups, including two cyclic groups that exhibit reverberating activity with a time (phase) shift.

Polychronization: Computation with Spikes, E.M. Izhikevich, Neural Computation 18, 245-282 (2006)

Synaptic weight change algorithms

- Long term, local Hebbian learning
  - "Neurons that wire together, fire together, neurons that fire out of sync, lose their link"
  - STDP is a modified version of Hebb's postulate
  - Biological weight change algorithm
    - Local computation in synapses
    - Helps to stabilize the network
- Short term plasticity: STD/STP
- Global knobs/Feedback
  - Reward/Attention system?
  - Dopamine for learning and ACH for attention
  - What is the biological mechanism of action of these chemicals?

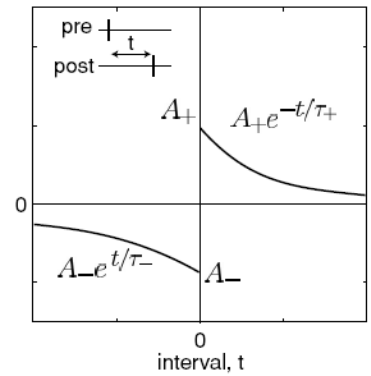
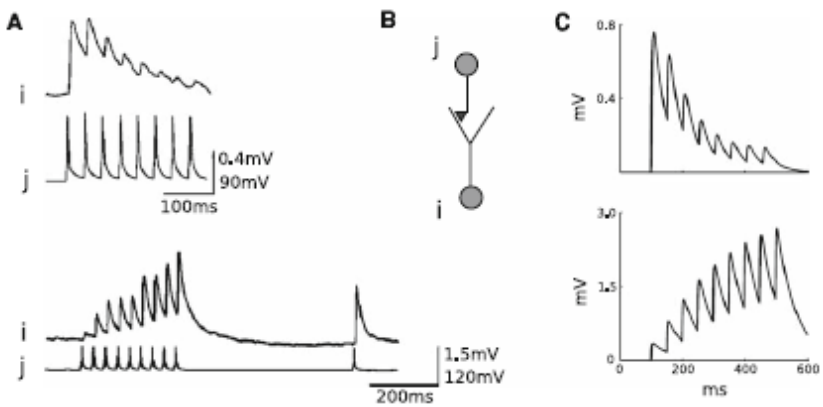


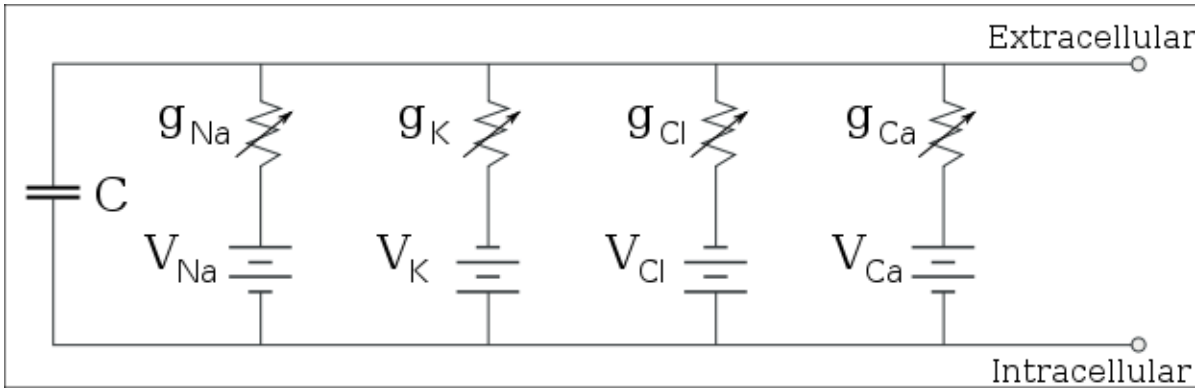
Figure 4: STDP rule (spike-timing-dependent plasticity, or Hebbian temporally asymmetric synaptic plasticity): The weight of synaptic connection from pre- to postsynaptic neuron is increased if the postneuron fired after the presynaptic spike, that is, the interspike interval $t > 0$. The magnitude of change decreases as A_+e^{-t/τ_+}. Reverse order results in a decrease of the synaptic weight with magnitude A_-e^{t/τ_-}. Parameters used: $\tau_+ = \tau_- = 20$ ms, $A_+ = 0.1$, and $A_- = 0.12$.



Polychronization: Computation with Spikes, E.M. Izhikevich, Neural Computation 18, 245-282 (2006)
Phenomenological models of synaptic plasticity based on spike timing, Morrison et al, Biol Cybern, DOI 10.1007/s00422-008-0233-1

Neuron Models

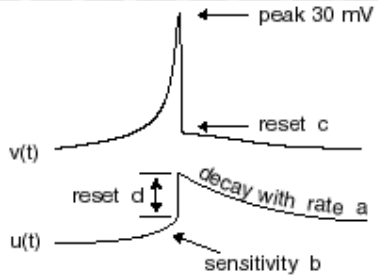
- Voltage sources and non-linear, voltage gated conductances
- Membrane acts as a capacitor
- Various computational models proposed, varying degrees of biological accuracy/relevance
  - Integrate and fire (Linear/Quadratic)
  - Simple Model
  - Hodgkins-Huxley
  - Many others

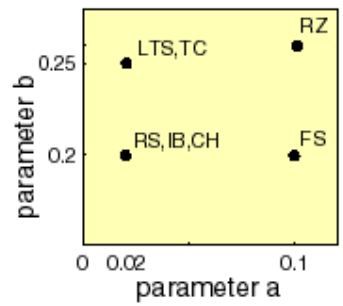


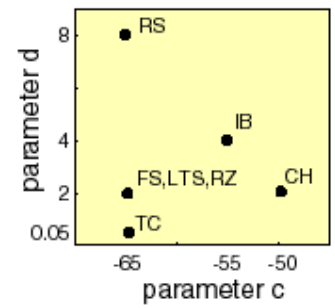
Simple Model

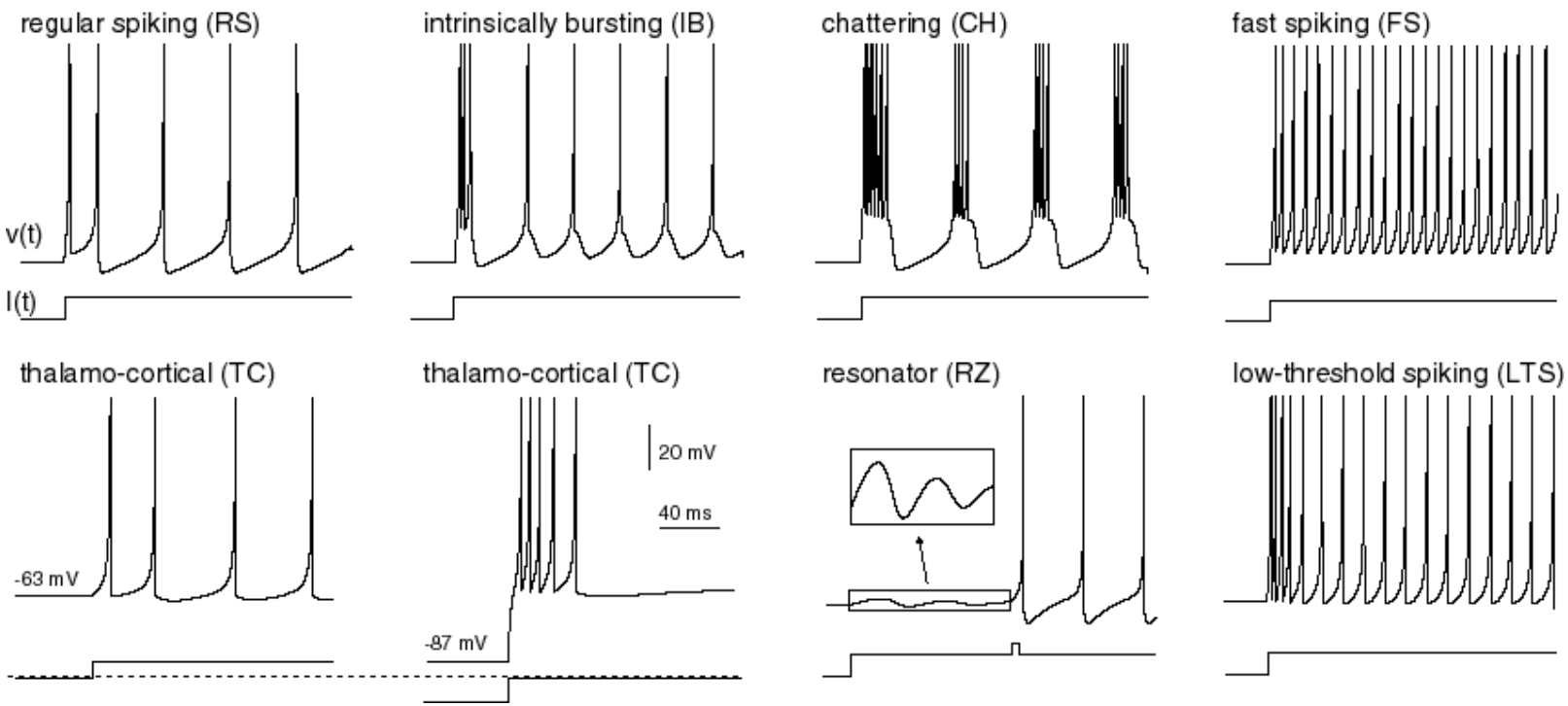
$v' = 0.04v^2 + 5v + 140 - u + I$
$u' = a(bv - u)$

if $v = 30$ mV,
then $v \leftarrow c$, $u \leftarrow u + d$





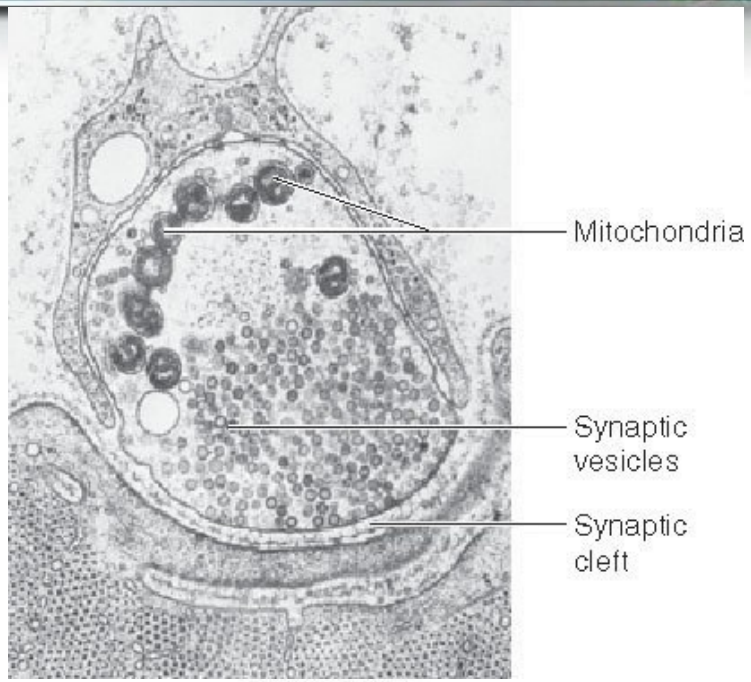


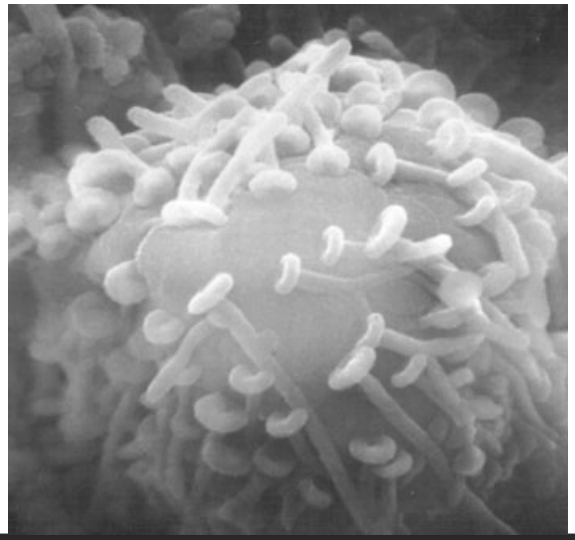


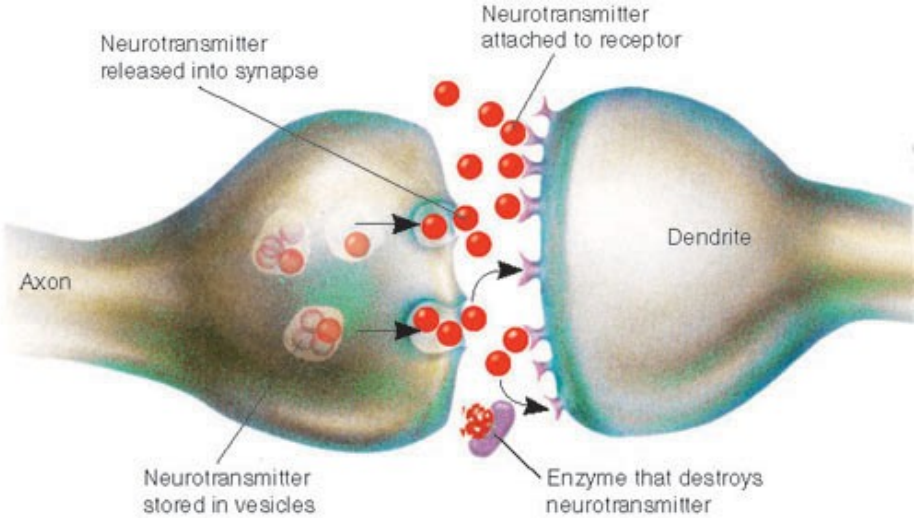
Electronic version of the figure and reproduction permissions are freely available at www.izhikevich.com

Synapses

- Synapses are dynamic weights/multipliers
  - Weight update using STDP/STP and ???
- Need very large numbers of these elements
  - Must be low power (in pico Joules per synapse operation)
  - Extremely small, sub micron dimensions for a synapse
- Enormous amounts of memory, computation and circuitry
  - Digital solution uses multiple bits of memory to store each synaptic weight and synaptic state
  - Analog memory: Floating gates, memristors (HP) or ???







Outline

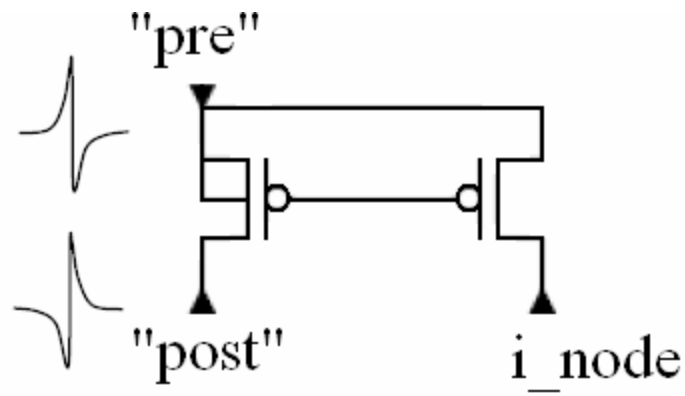
- Motivation
- Biology/Neuroscience
- Computational Abstractions
-

Technology Tradeoffs

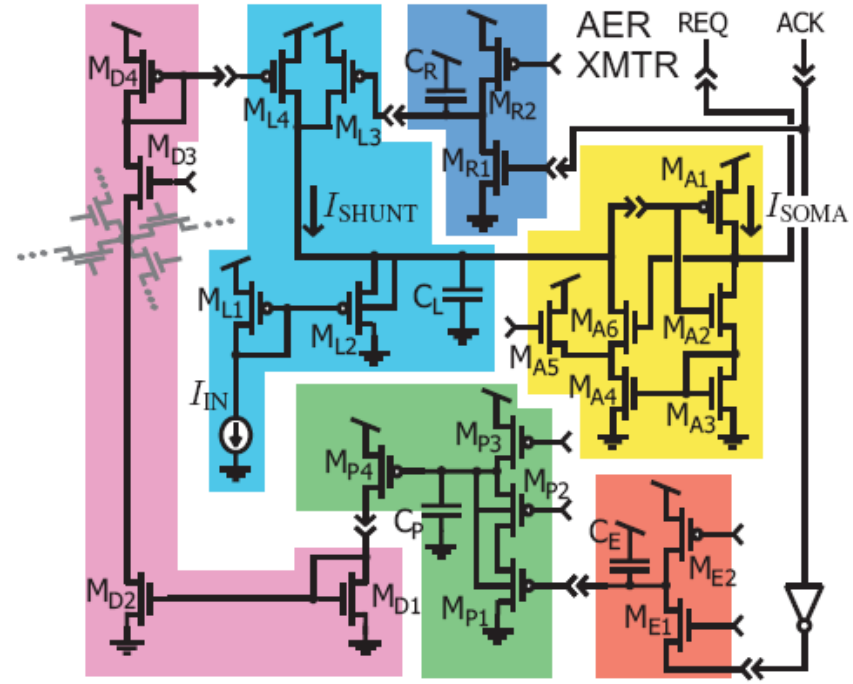
Technology	Relative Area	Design time	Real time operation	Power consumption	Clone-ability
Biological Brains	Very small	Millions of years	Yes	Very low	Low
PC	Huge	Short	for smaller networks	High	High
DSP/GPU	Huge	Short	for smaller networks	High	High
FPGA/dVLSI	Medium	Medium	for smaller networks	Medium-High	High
Above threshold aVLSI	Small	Long	Yes	Medium	Medium
Sub-threshold aVLSI	Small	Long	Yes	Low	Low

Neuromorphic Engineering

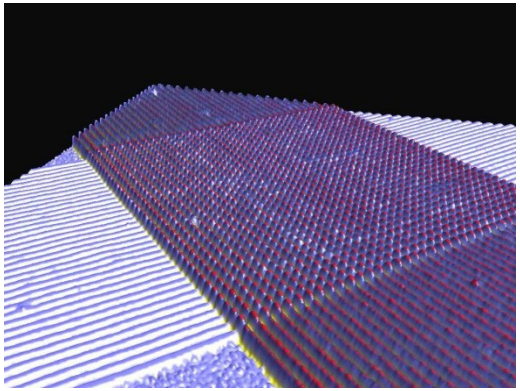
- Carver Mead pioneered utilizing sub-threshold MOS neuromorphic circuits
- Synapses using Floating gate/memristors
- Digital connectivity



Haas et al, Two Transistor Synapse with Spike Timing Dependent Plasticity



Arthur ISCAS 2006



Memristive synapses fabricated by HP

Engineering Challenges

- "Neural" hardware
  - Uploading pre-trained sequences into a new piece of hardware
  - System expandability: adding in old pieces of hardware to a new network
  - "Service packs" to fix bugs?
  - What happens when you lose power? Does it die? How much?
- Tools
  - Compilers, debuggers and high level programming tools for neural hardware
- So you have the biggest ever network of neurons/synapses, now what?
  - Need to develop a "lesson plan" to teach it: what is a good lesson?
  - How do you know it works: metrics to evaluate learning

Thank you!