Brain-Inspired Learning Machines

Emre Neftci

Department of Cognitive Sciences, UC Irvine,

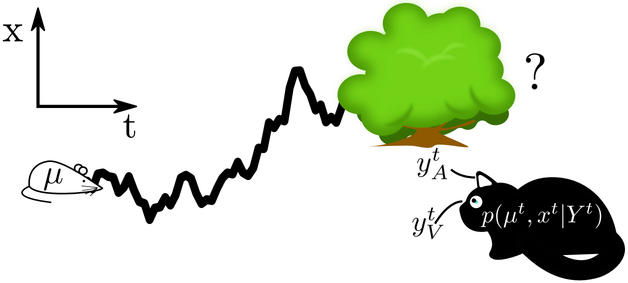
September 23, 2016

Emre Neftci
eneftci@uci.edu
SBSG 2308

Fri 1:00-3:50, Room SBSG 2200

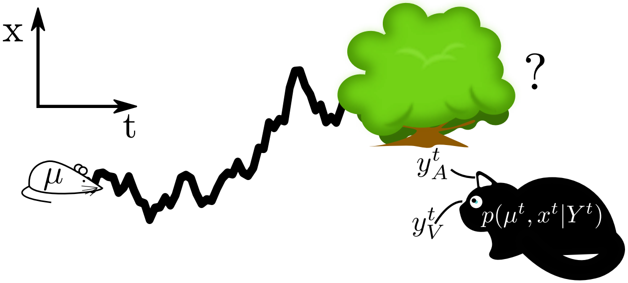
https://canvas.eee.uci.edu/courses/3021

The goal of a cognitive agent is to maximize the expected outcome of its actions



The goal of a cognitive agent is to maximize the expected outcome of its actions

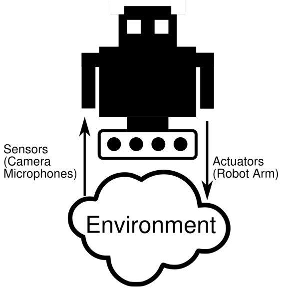
The environment, the sensory information is highly unreliable and ambiguous.



The goal of a cognitive agent is to maximize the expected outcome of its actions

The environment, the sensory information is highly unreliable and ambiguous.

The brain's "building blocks" are noisy!

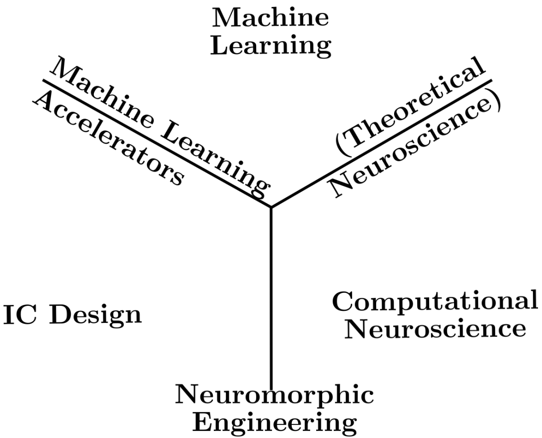


To sucessfully interact with the environment, the brains of cognitive agents need to perform:

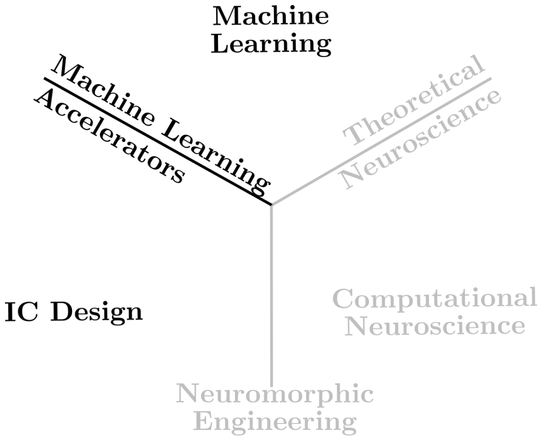
- Pattern Recognition
- Prediction
- Action Selection

How can the brain's "building blocks" be organized to perform *and learn to perform* these operations?

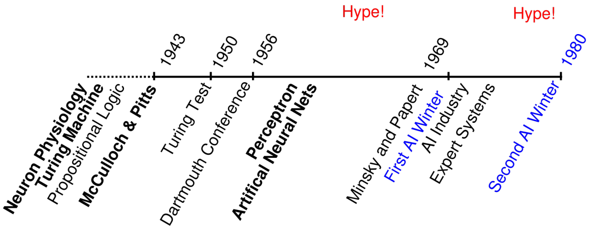
This course: a cross-disciplinary study of the building blocks of cognition

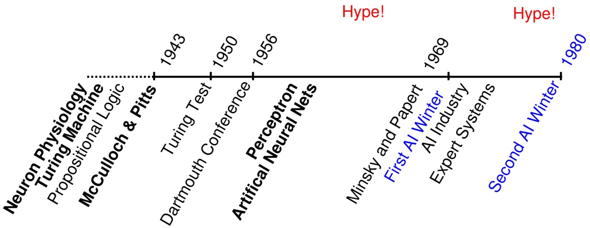


Brain-Inspired Learning Machines: where machine learning, neuroscience and IC design meet.


Machine Learning
Machine Learning Accelerators
Theoretical Neuroscience
IC Design
Computational Neuroscience
Neuromorphic Engineering

History of Artificial Intelligence and Neural Networks



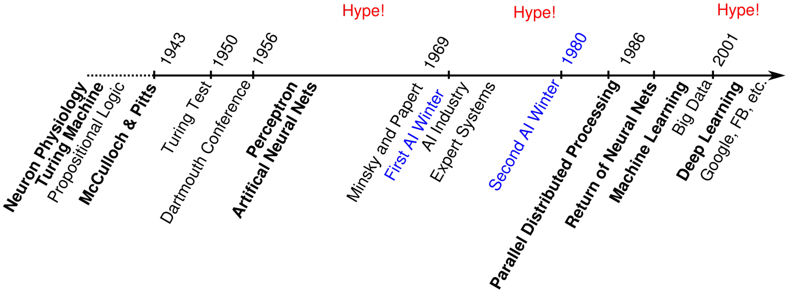


Early AI shortcomings:

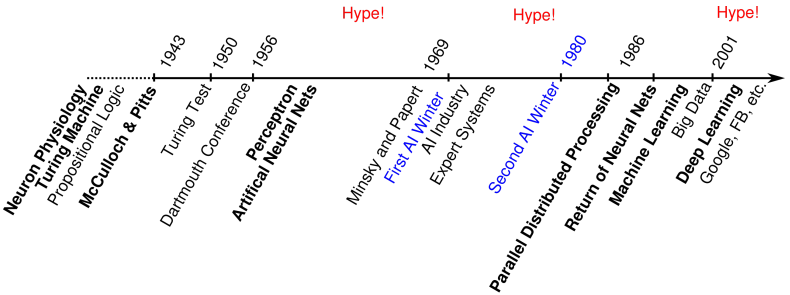
- Symbol based processing lacks domain-specific knowledge
- Combinatorial explosion: solutions to small problems did not scale to exponentially large problems.
- Solving a problem in principle is very different than solving it practically

There is a need to systematically learn domain-specific knowledge to solve practical AI tasks

Modern Artificial Intelligence and Machine Learning



Modern Artificial Intelligence and Machine Learning



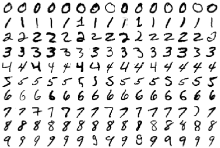
Good old online backpropagation for plain multilayer perceptrons yields a very low 0.35% error rate on the MNIST handwritten digits benchmark. All we need to achieve this best result so far are many hidden layers, many neurons per layer, numerous deformed training images to avoid overfitting, and graphics cards to greatly speed up learning.

Cireşan, Meier, Gambardella, and Schmidhuber, *Neural computation*, 2010

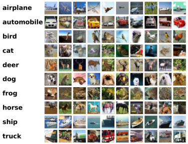
A lot of progress in machine learning can be attributed to better hardware and more data

A machine learning algorithm is a program that can learn to solve a given using **data**.

Example Datasets:



MNIST



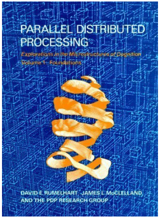
CIFAR-10

ImageNet

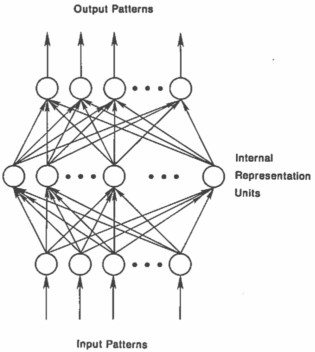
DARPA Neovision2 Tower benchmark

Provided enough data, machine learning usually can outperform experts' knowledge

“A set of approaches that models artificial intelligence using networks of simple (neuron-like) units.”



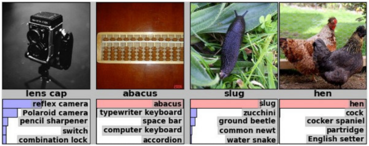
Rumelhart, McClelland, and Group,, 1988



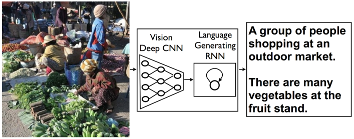
Neural networks can be viewed as machine learning algorithms that run on networks of neuron-like units

Applications of Artificial Neural Networks

- Image Recognition



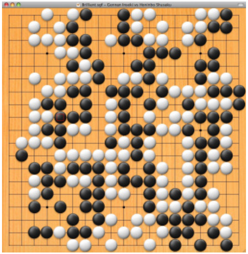
- Speech Recognition, *e.g.* Siri
- Natural Language Processing, *e.g.* Spam detection, building descriptions of images

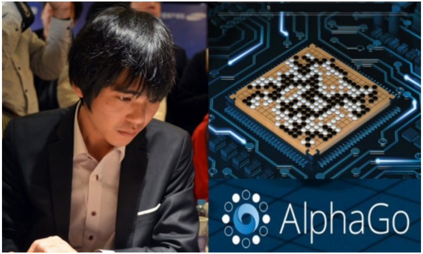


Google Research Blog, 2014

- Drug Discovery, Finance, Robots, Games, ...

Deep artificial neural networks are state-of-the-art most of these applications





- The game Go: 19×19 grid, 10^{360} possible states. Number of atoms in the universe: 10^{82}
- Mar. 2016: Google Deepmind's AlphaGo beats World champion.

"Deep neural networks trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play."

Silver et al., *Nature*, 2016

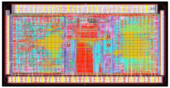
Machine Learning Accelerators

TPU (Google)

Movidius (Intel)

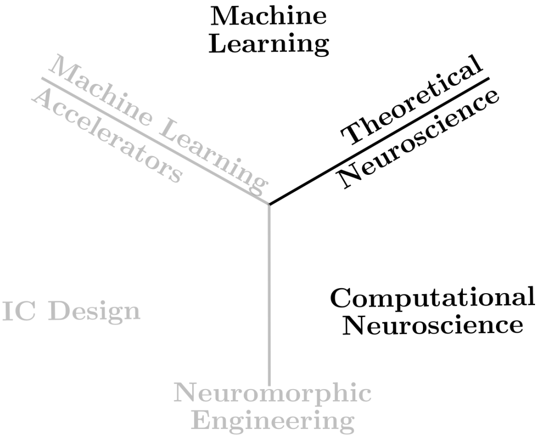
neon
framework by nervana

Nervana (Intel)

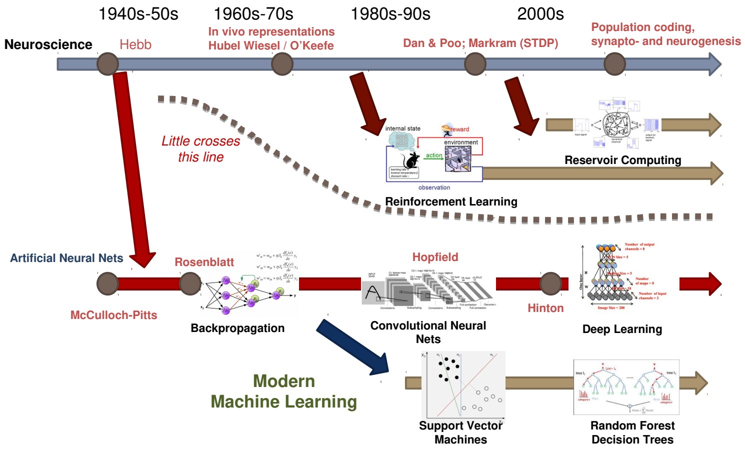


NN-x

TeraDeep


Machine Learning
Machine Learning Accelerators
Theoretical Neuroscience
IC Design
Computational Neuroscience
Neuromorphic Engineering

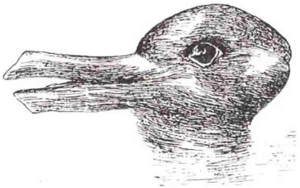
There is a wide gap between AI and Machine Learning



B. Aimone, NICE workshop 2015

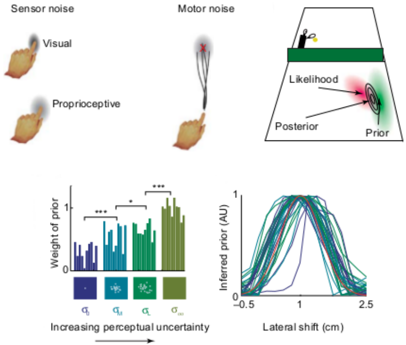
This course aims at bridging the gap between brains and machine learning

Neural circuits perform (approximate) Probabilistic inference



Perception as unconscious inference

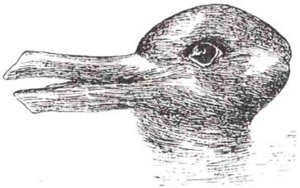
Helmholtz, 1867



Humans implicitly use Bayesian inference

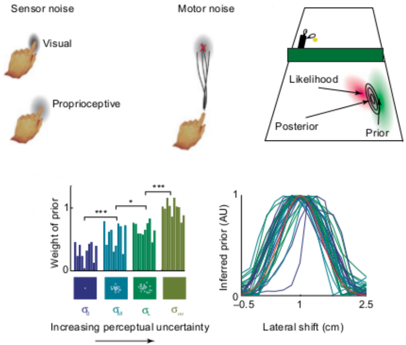
Körding and Wolpert, *Nature*, 2004

Neural circuits perform (approximate) Probabilistic inference



Perception as unconscious inference

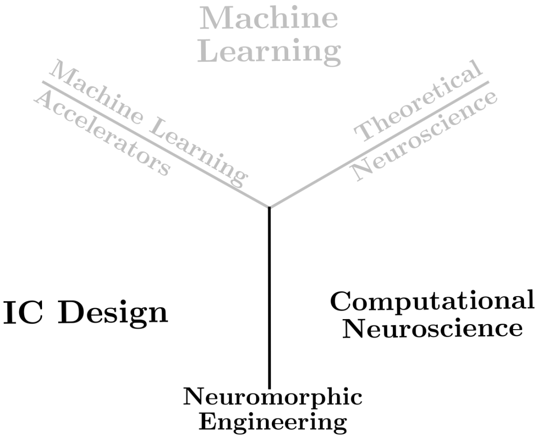
Helmholtz, 1867



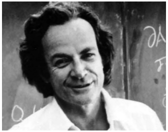
Humans implicitly use Bayesian inference

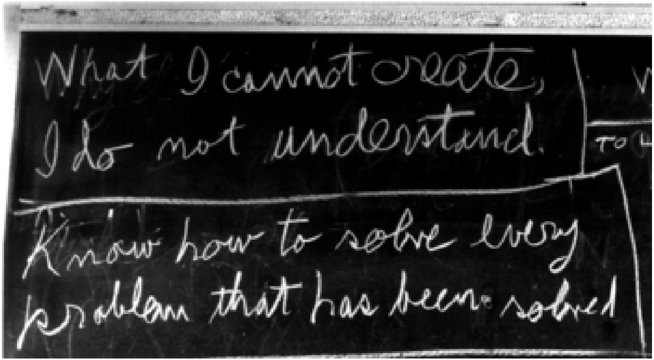
Körding and Wolpert, *Nature*, 2004

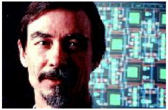
How is (approximate) probabilistic inference instantiated at the neuron level?


Machine Learning
Machine Learning Accelerators
Theoretical Neuroscience
IC Design
Computational Neuroscience
Neuromorphic Engineering

Neuromorphic Engineering



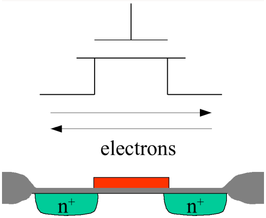




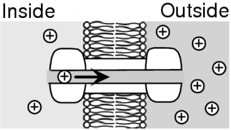
"The constraints on our analog silicon systems are similar to those on neural systems: wire is limited, power is precious, robustness and reliability are essential."

Mead,, 1989

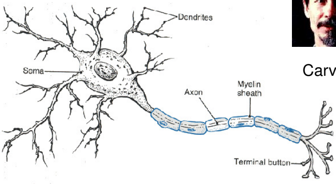
Emulation of the bio-physics of neural systems circuits

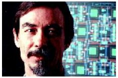


Transistor



Ionic Channel



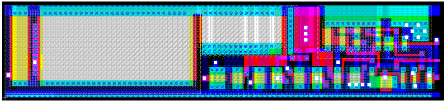


Carver Mead

$$C\frac{\mathrm{d}}{\mathrm{d}t}V_{mem} = g_{lk}(E_{lk} - V_{mem}) + I_{syn},$$

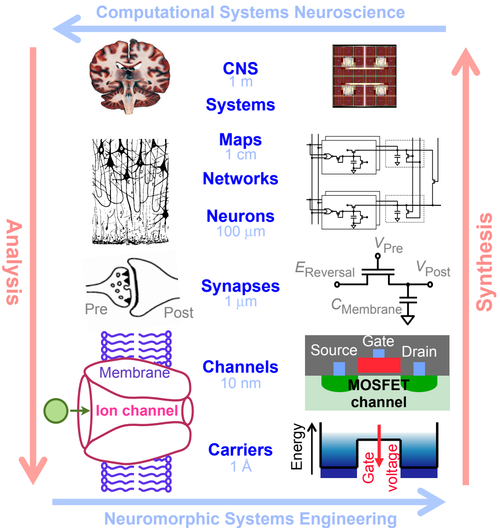
$$\text{if } V_{mem} = \Theta, V_{mem} \leftarrow V_{reset}$$

Integrate and Fire neuron model



Chicca, Stefanini, and Indiveri, 2013

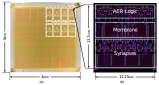
Analysis by Synthesis



Cauwenberghs, *Proceedings of the National Academy of Sciences*, 2013

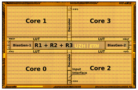
Active Development of Brain-Inspired Computers and Neuromorphic Hardware

Analog processing, digital asynchronous communication



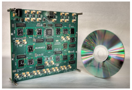
IFAT(Neovision2)

UC San Diego



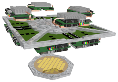
CxQuad

ETH Zurich



Neurogrid

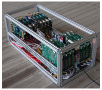
Stanford



FACETS (HBP)

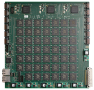
Univ. of Heidelberg

Fully digital processing



Blue Hive

Univ. of Cambridge

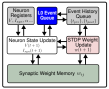


SpiNNaker (HBP)

Univ. of Manchester

TrueNorth (SyNAPSE)

IBM



NSAT

Intel/UCI/UCSD

Software Simulators

CARLsim

UC Irvine

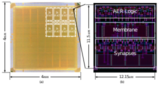
HRLsim

HRL

Compass

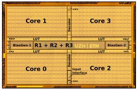
IBM

Analog processing, digital asynchronous communication



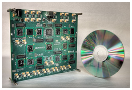
IFAT(Neovision2)

UC San Diego



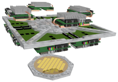
CxQuad

ETH Zurich



Neurogrid

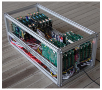
Stanford



FACETS (HBP)

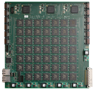
Univ. of Heidelberg

Fully digital processing



Blue Hive

Univ. of Cambridge

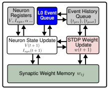


SpiNNaker (HBP)

Univ. of Manchester

TrueNorth (SyNAPSE)

IBM



NSAT

Intel/UCI/UCSD

Software Simulators

CARLsim

UC Irvine

HRLsim

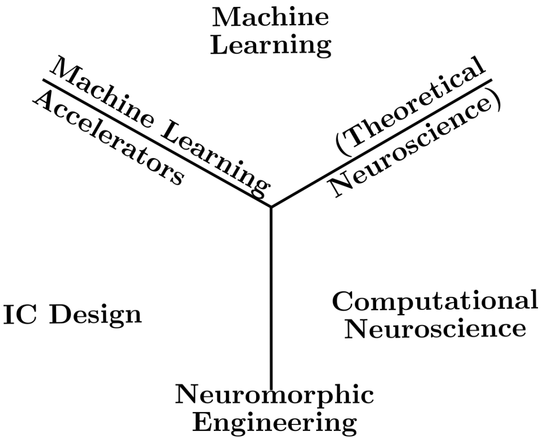
HRL

Compass

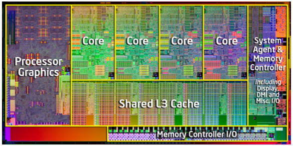
IBM

"Our ability to realize simple neural functions is strictly limited by our understanding of their organizing principles, and not by difficulties in implementation. "

Mead, 1989


Machine
Learning
Machine Learning
Accelerators
(Theoretical
Neuroscience)
IC Design
Computational
Neuroscience
Neuromorphic
Engineering

At the system level, brains are orders of magnitude more efficient than digital processors

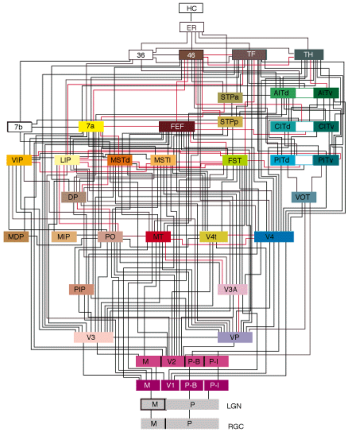


Fast global clock	Self-timed
Distant memory	Local memory
Deterministic, digital states (0,1)	Stochastic, analog states, digital comm.
Devices frozen on fabrication	Constant adaptation and self-modification

- The cost of an elementary operation (turning on transistor or activating synapse) is about the same.

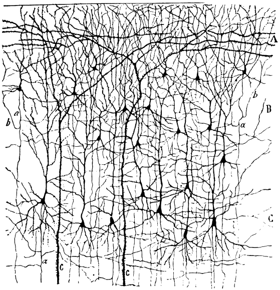
  The brain uses a fundamentally different computational architecture.

Distributed Organization of the Primate Visual Cortex



Felleman and Van Essen, *Cerebral cortex*, 1991

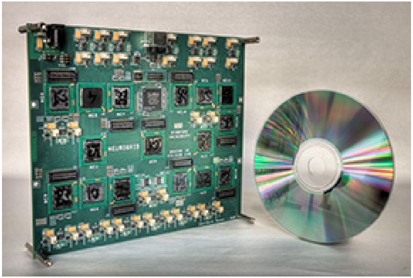
Pyramidal cells (Neurons) in the Cortex



Ramon y Cajal, 1911

The cortex computes in a massively parallel fashion using “simple”, slow and imprecise units

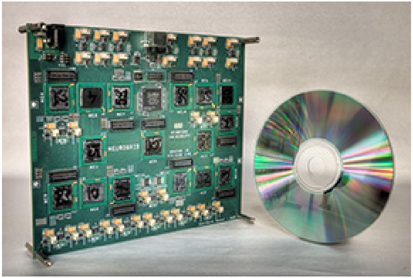
BlueGene (IBM): 10 MW, 131072 processors in parallel, needs one floor of an entire building AND one floor for cooling devices.



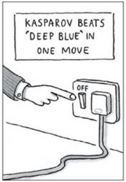
Neurogrid (Stanford): 5 W, 10^6 neurons, “rivaling BlueGene’s performance”.

Power dissipation in Digital Computers

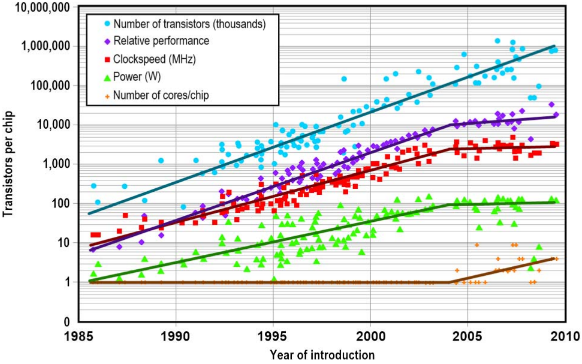
BlueGene (IBM): $10\,\mathrm{MW}$, 131072 processors in parallel, needs one floor of an entire building AND one floor for cooling devices.



Neurogrid (Stanford): $5\,\mathrm{W}$, 10^6 neurons, “rivaling BlueGene’s performance”.



Moore’s law: denisty of transistors in an integrated circuit doubles every two years



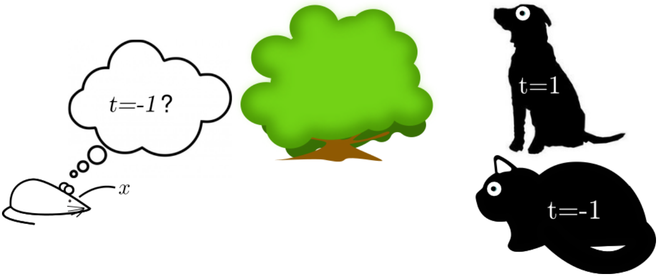
S. H. Fuller and L. I. Millet, 2011; T. Hamilton et al. 2014

Need for new computing technologies. Brain-like computing for a subclass of problems

The State of Brain-inspired Computing

Brain-Inspired Machine Learning


μ
?


t=-1?
x
t=1
t=-1

Given a training dataset D, the goal of machine learning and pattern recognition is to determine t given a new observation of x. Different approaches to achieve this can be used.

- **Discriminant model:** Learn a function $f(x)$ that maps data x to target t, without *explicitely* taking probabilities into account.

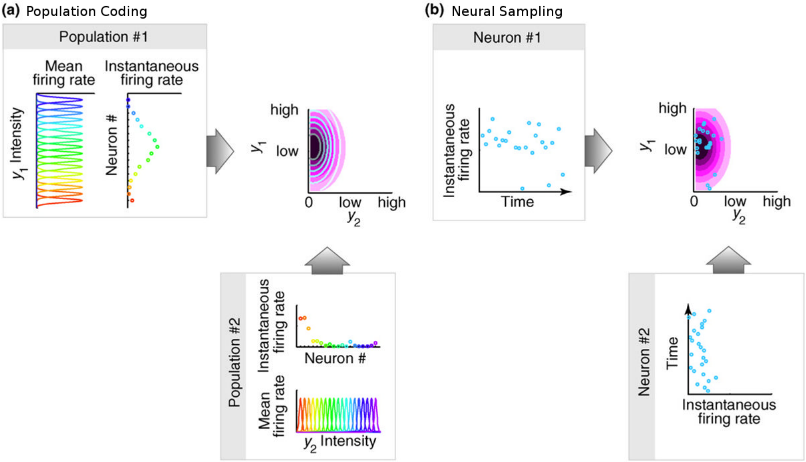
- **Discriminative models**: Learn posterior class probabilities: $p(t|x)$. Given this distribution, we can classify and have a degree of certainty.

- **Generative models**: Learn the joint distribution $p(x, t)$. Given this distribution, we can compute any of the quantities of interest.

Two methods to compute $P(y_1, y_2)$:

$$P(y_1, y_2) \in [0, 1]$$

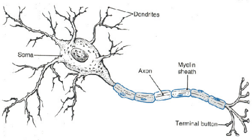
$$y_i(\omega) \in \mathbb{R}^2$$
$$hist((y_1, y_2))$$

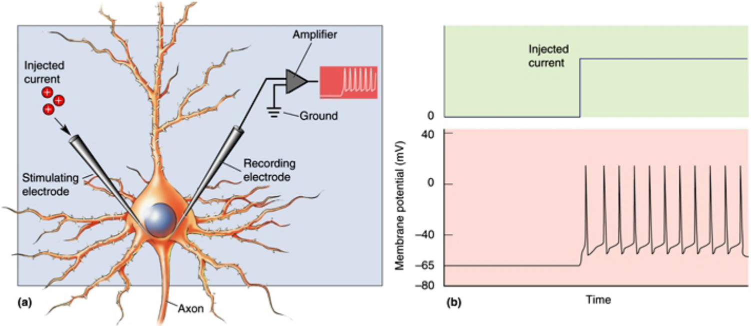


Fiser, Berkes, Orbán, and Lengyel, *Trends in cognitive sciences*, 2010

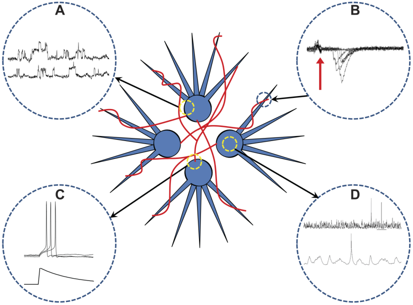
Course Overview

Biological Neuron Models

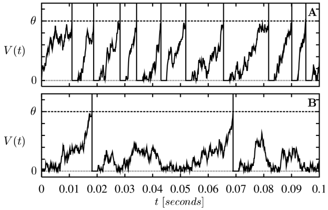


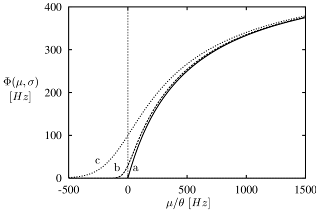


Mean-field Models for Stochastic Spiking Neurons

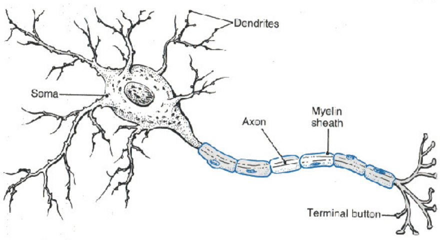


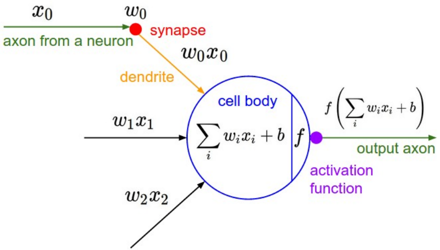
Yarom and Hounsgaard, *Physiological Reviews*, 2011

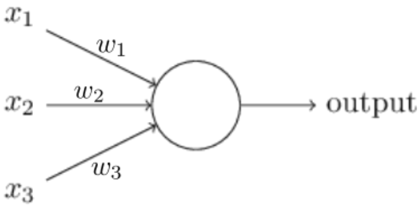




Fusi and Mattia, *Neural Computation*, 1999


Dendrites
Soma
Axon
Myelin sheath
Terminal button

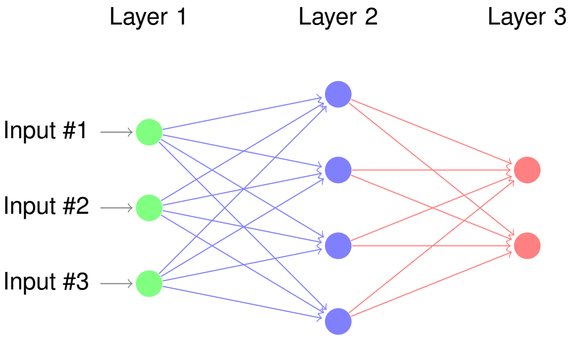

x_0
w_0
synapse
axon from a neuron
$w_0 x_0$
dendrite
cell body
$f\left(\sum_i w_i x_i + b\right)$
$w_1 x_1$
$\sum_i w_i x_i + b$
f
output axon
activation function
$w_2 x_2$



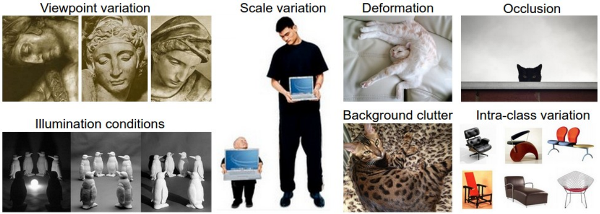
$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases} \tag{1}$$

Learning to recognize (simple) patterns with Perceptrons

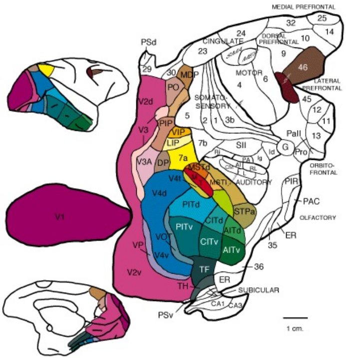
w_{ij}^{L}: means weight j of unit i in layer $L = 1, 2, 3, ...$

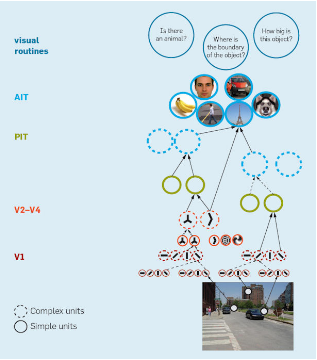


Challenges in computer vision



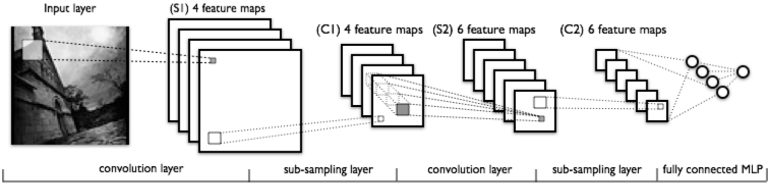
Hierarchical Organization of the Visual Pathway





Felleman and Van Essen, 1991 (left), Cerebral Cortex 1:1-47. Serre and Poggio, 2007 (right)

HMAX - A computational model of object recognition in cortex


Input layer
(S1) 4 feature maps
(C1) 4 feature maps
(S2) 6 feature maps
(C2) 6 feature maps
convolution layer
sub-sampling layer
convolution layer
sub-sampling layer
fully connected MLP

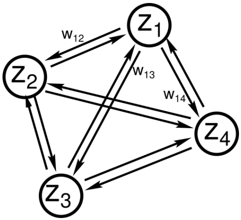
(a) Stored Memories

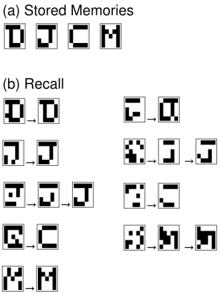
moscow------russia
lima----------peru
london-----england
tokyo--------japan
edinburgh-scotland
ottawa------canada
oslo--------norway
stockholm---sweden
paris-------france

(b) Recall

moscow---:::::::::: ⟹ moscow------russia
::::::::::--canada ⟹ ottawa------canada

otowa-------canada ⟹ ottawa------canada
egindurrh-sxotland ⟹ edinburgh-scotland



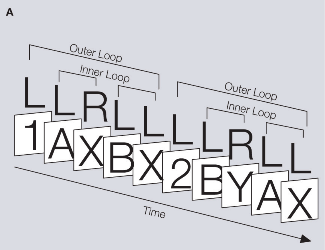


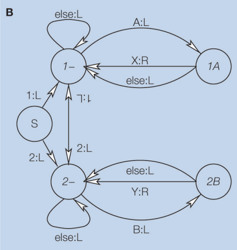
MacKay,, 2003

Working Memory

12AX Task

State machine solving 12AX





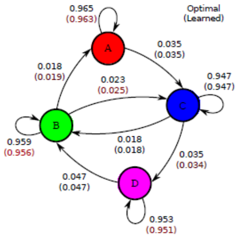
O’Reilly and Frank, 2006

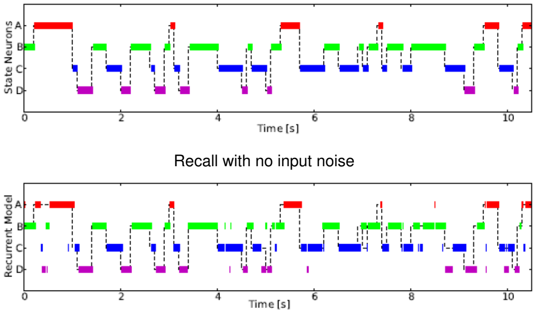
Dayan, 2008

- Inference with neural sampling
- On-line learning with *spiking Expectation Maximization*
  - Learning based on a form of Spike Time Dependent Plasticity (STDP)

Nessler, Pfeiffer, and Maass, 2013

State Diagram





Recall with no input noise

Recall with 70% input noise

Corneil, Neftci, Indiveri, Pfeiffer, 2015

../l01/code/lif.py

```
import numpy as np
import pylab

#Parameters
alpha = .975
b = .0251
T = 1000
theta = 1

#Array for saving intermediate variables
V = np.zeros([T])

#Main loop
v = 0
for t in range(T):
    v = alpha*v + b
    if v >= theta:
        v = 0
    V[t] = v

#Plot
pylab.plot(V)
pylab.xlabel('t [au]')
pylab.ylabel('V [au]')
pylab.show()
```

Class Organization

Fridays 1:00-3:50PM, Sept. 23 - Dec. 2
9 Classes, 11 weeks

- Introduction to the course . . . .
- Mathematical Models of Biological Neurons . . . .
- Basic Concepts in Machine Learning and Neural Networks . . . .
- Spike-Based Neural Networks and Deep Learning . . . .
- Probabilistic Population Codes for Bayesian Inference . . . .
- Neural models of Associative memory and Spike-based Monte Carlo Sampling . .
- Self-Organizing Models, Competitive learning and Expectation Maximization . . . . .
- Sequence Learning* . . . .
- Reservoir Computing: Liquid State Machines and the Neural Engineering Framework . . . .
- Dimensionality Reduction Algorithms* . . . .
- Group Projects . . . .
- Final Projects Presentation . . . .

No single textbook that covers all the material, but the following books provide useful complementary information on some of the covered topics.

Key References:

- *Pattern recognition and machine learning*, Bishop, 2006
- *Neuronal dynamics: From single neurons to networks and models of cognition*, Gerstner, Kistler, Naud, and Paninski, 2014
- *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, Dayan and Abbott, 2001
- *Neural Networks and Deep Learning*, Nielsen, 2015
- *Neuroscience*, Bear, Connors, and Paradiso, 2007

Programming References:

- `http://www.scipy-lectures.org/intro/index.html` (sections 1.1 through 1.4).

Software:

- Python (numpy, scipy, scikits)
- TensorFlow (For deep learning)

Grading Plan

1. Assignments (50%)
2. Projects (50%)

Important Dates:

Final Presentation .... December 2, 2016

No make-up examinations.

Reports and assignments must be submitted before the deadline posted with each assignment sheet.

Biweekly Assigments:

- Pencil and Paper
- Python Programming

Programming

- Why Python?
  - Versatile
  - Free and Open source
  - Rich collection of modules
  - Basic are easy to learn
- No previous programming experience is OK

Register on `https://canvas.eee.uci.edu/courses/3021`