

Artificial Intelligence and the Singularity

piero scaruffi

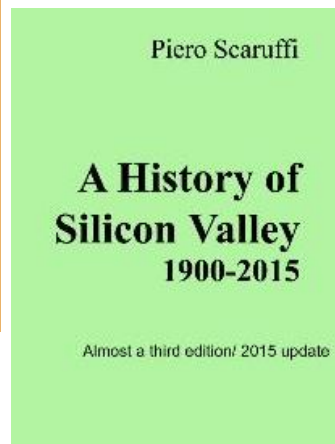
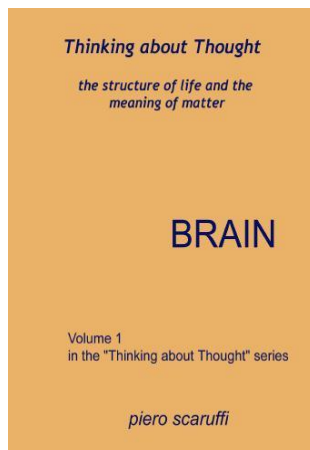
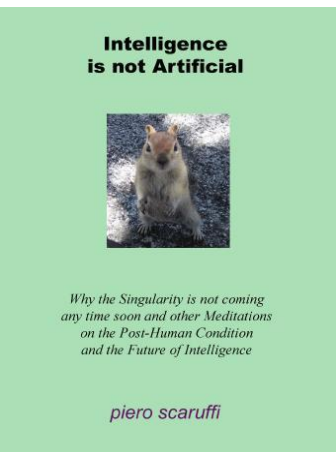
www.scaruffi.com

October 2014 - Revised 2016

*"The person who says it cannot be done should not
interrupt the person doing it" (Chinese proverb)*

Piero Scaruffi

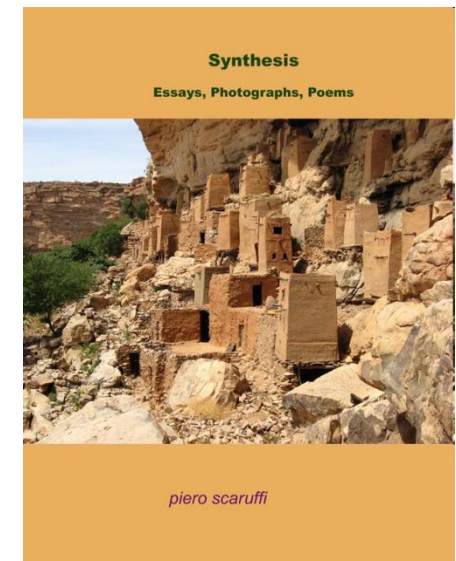
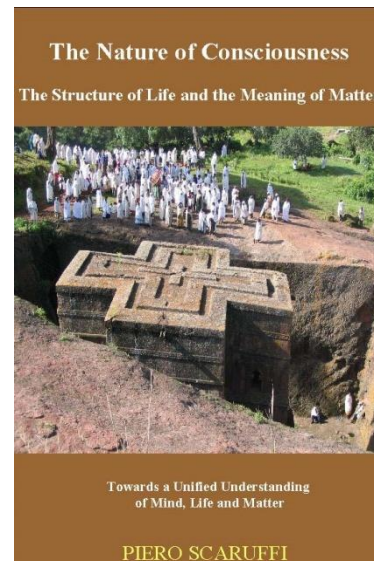
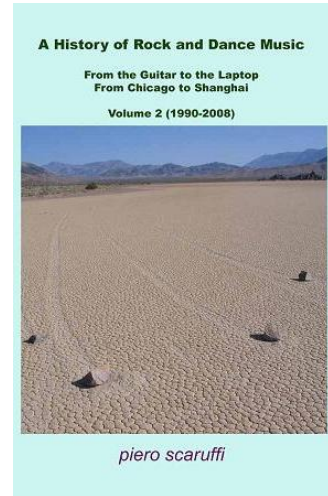
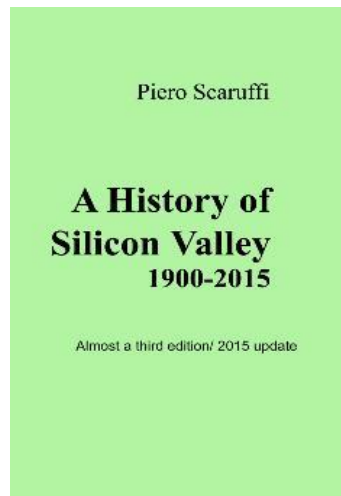
- piero scaruffi
p@scaruffi.com
scaruffi@stanford.edu



Olivetti AI Center, 1987

Piero Scaruffi

- Cultural Historian
- Cognitive Scientist
- Blogger
- Poet
- www.scaruffi.com

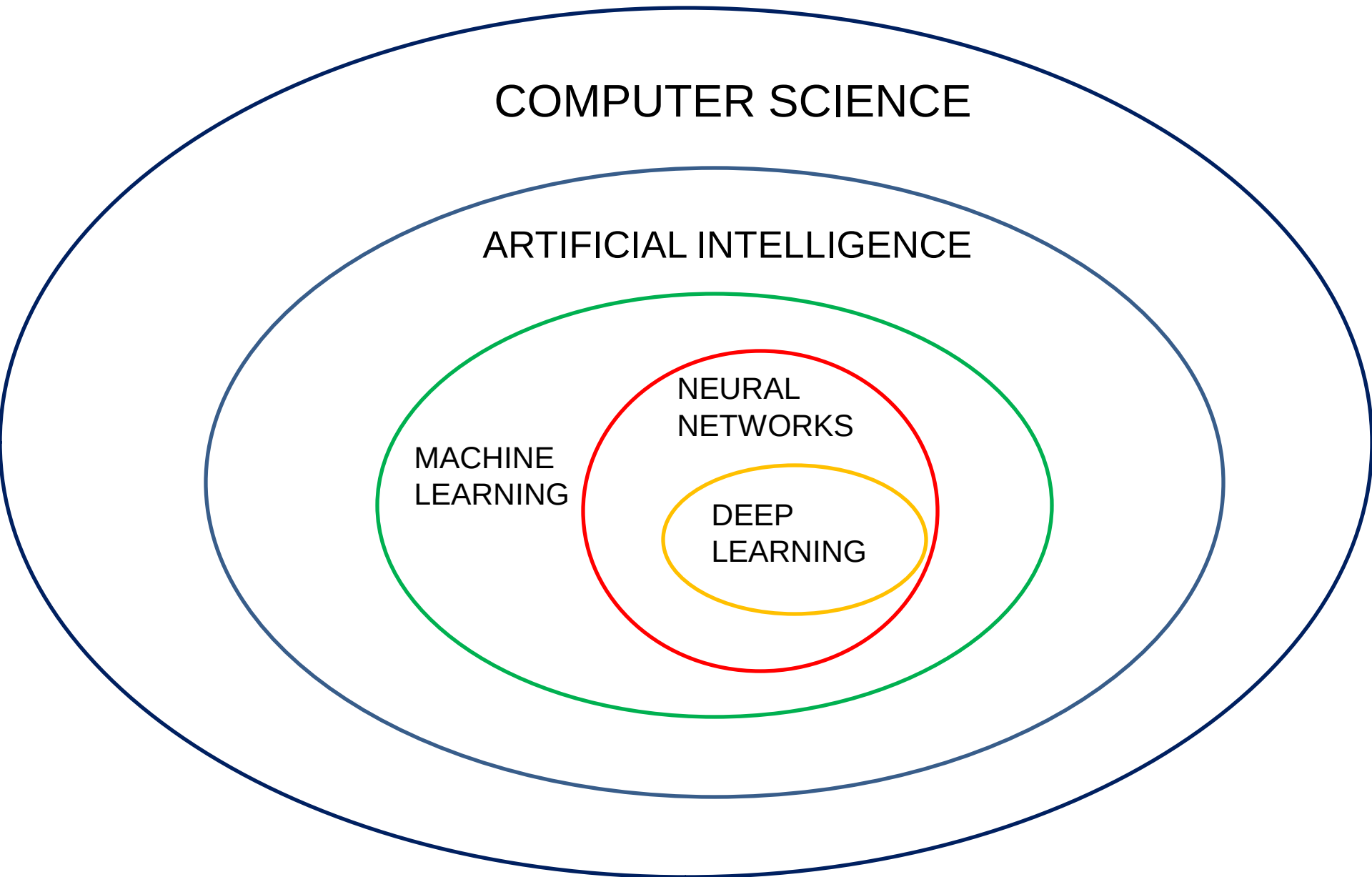


This is Part 2

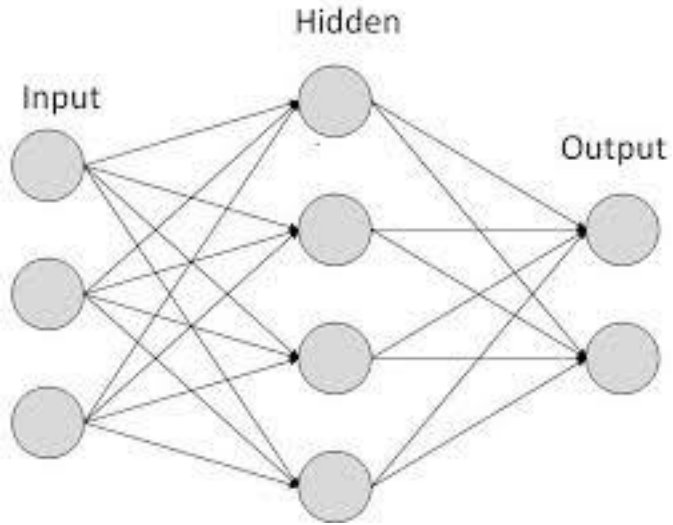
- See <http://www.scaruffi.com/singular> for the index of this Powerpoint presentation and links to the other parts
 1. Classic A.I. - The Age of Expert Systems
 2. The A.I. Winter and the Return of Connectionism
 3. Theory: Knowledge-based Systems and Neural Networks
 4. Robots
 5. Bionics
 6. Singularity
 7. Critique
 8. The Future
 9. Applications
 10. Machine Art
 11. The Age of Deep Learning
 12. Natural Language Processing

The A.I. Winter and the Return of Connectionism

Neural Networks

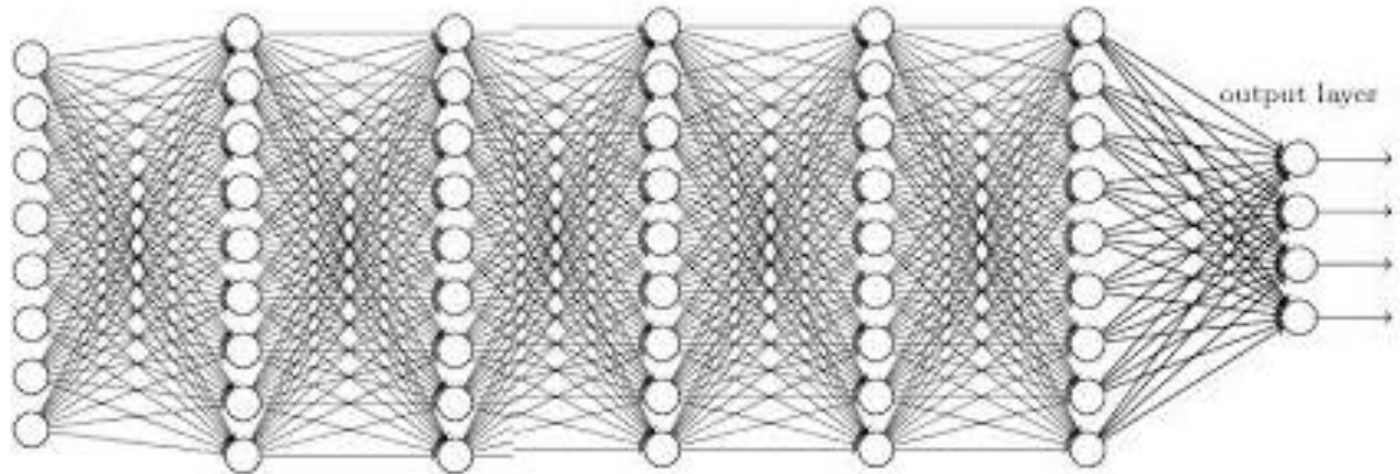


Neural Networks



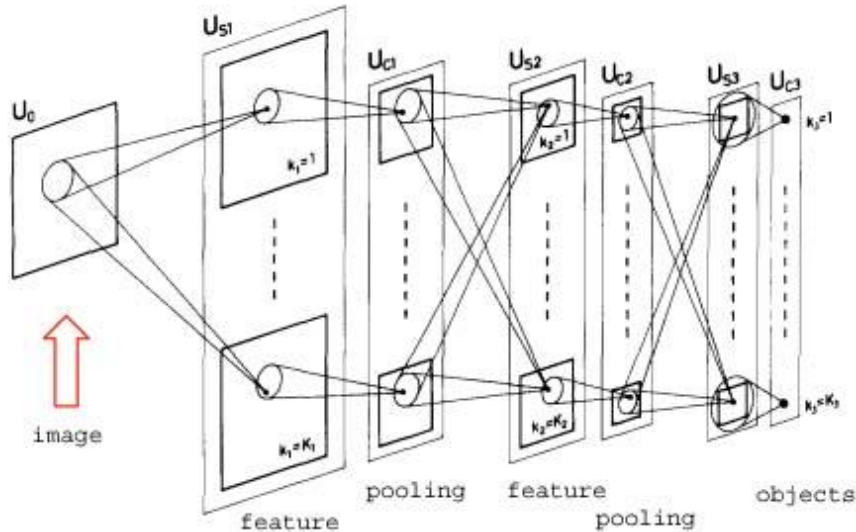
3 LAYERS

MANY LAYERS
("DEEP")



Neural Networks

1979: Kunihiro Fukushima's "Neocognitron": the birth of convolutional neural networks



*J. Physiol. (1962), 160, pp. 106–154
With 2 plates and 20 text-figures
Printed in Great Britain*

Based on the cat's visual system:

**RECEPTIVE FIELDS, BINOCULAR INTERACTION
AND FUNCTIONAL ARCHITECTURE IN
THE CAT'S VISUAL CORTEX**

BY D. H. HUBEL AND T. N. WIESEL

*From the Neurophysiology Laboratory, Department of Pharmacology
Harvard Medical School, Boston, Massachusetts, U.S.A.*

(Received 31 July 1961)



Neural Networks

1979: Kunihiro Fukushima's "Neocognitron"



convolution

$$u_{Sl}(k_l, \mathbf{n}) = r_l \cdot \varphi \left[\frac{1 + \sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\mathbf{v} \in S_l} a_l(k_{l-1}, \mathbf{v}, k_l) \cdot u_{Cl-1}(k_{l-1}, \mathbf{n} + \mathbf{v})}{1 + \frac{2r_l}{1+r_l} \cdot b_l(k_l) \cdot v_{Cl-1}(\mathbf{n})} - 1 \right]$$

where

$$\varphi[x] = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Relu

divisive
normalization

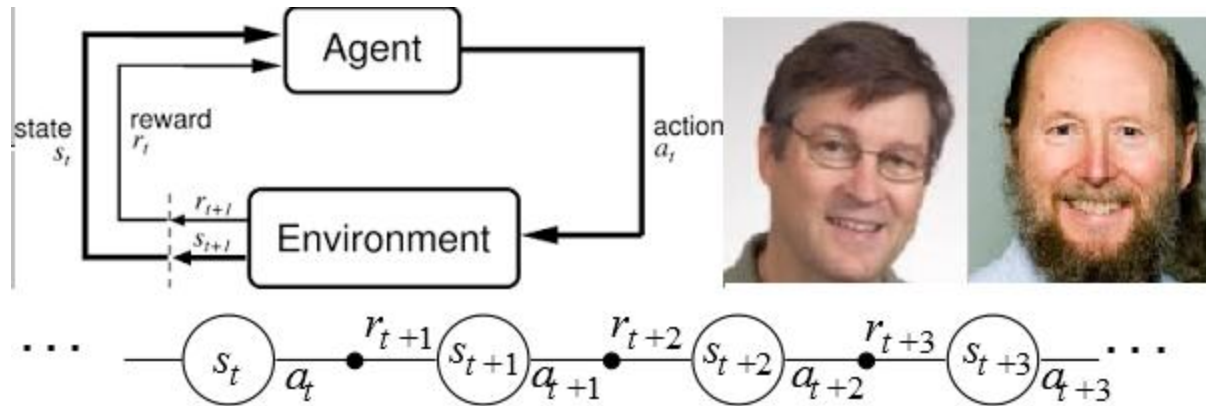
$$v_{Cl-1}(\mathbf{n}) = \sqrt{\sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\mathbf{v} \in S_l} c_{l-1}(\mathbf{v}) \cdot u_{Cl-1}^2(k_{l-1}, \mathbf{n} + \mathbf{v})}$$

$$\Delta a_l(k_{l-1}, \mathbf{v}, \hat{k}_l) = q_l \cdot c_{l-1}(\mathbf{v}) \cdot u_{Cl-1}(k_{l-1}, \hat{\mathbf{n}} + \mathbf{v}),$$

Hebbian learning

Neural Networks

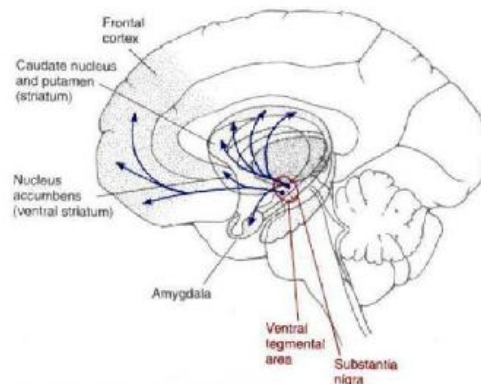
1981: Andrew Barto's & Richard Sutton's reinforcement learning



states are represented with V^π , a state at some time interval, t , to be s_t
the estimate at a given time of our states to be $V(s_t)$.

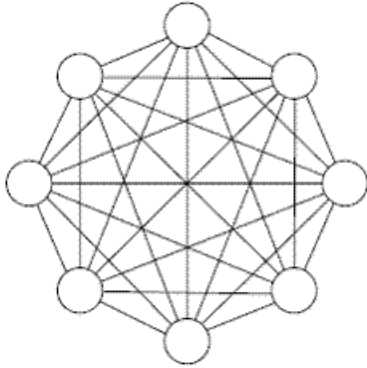
$$V(s_t) = V(s_t) + \alpha \delta \quad \text{where} \quad \delta_t = R_t - V(s_t)$$

Dopamine



Neural Networks

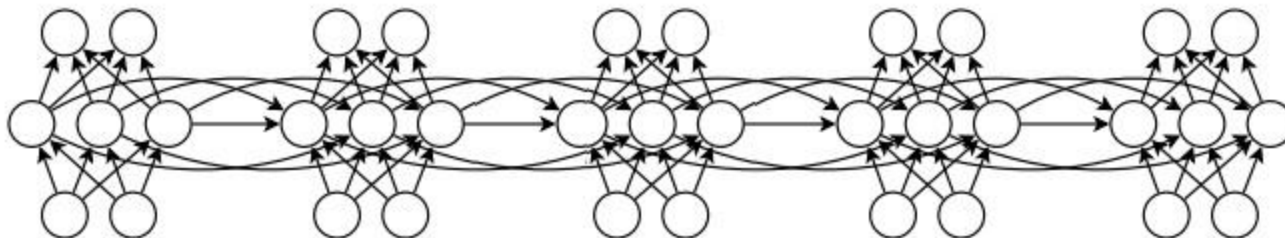
1982: John Hopfield's recurrent neural network



$$E(v) = -\frac{1}{2} \sum_{i \neq j} \sum_{i,j} w_{ij} v_i v_j - \sum_i I_i v_i + \sum_i \frac{1}{R_i} \int_0^{v_i} f_i^{-1}(z) dz$$

$$\nabla_v E(v) = Wv + I - u/R$$

Recurrent Neural Network



Neural Networks

1982-84: John Hopfield's recurrent neural network



Discrete Hopfield Network

Update Equations:

$$V_i(t+1) = \sigma\left(\sum_{j=1}^N w_{ij} V_j(t)\right)$$

Where $\sigma()$ is the sign function, $\text{sign}()$.

Energy function:

$$E = -1/2 \left(\sum_{i=1}^N \sum_{j=1}^N w_{ij} V_i V_j \right)$$

Continuous Hopfield Network

$$V(t+1) \leftarrow V(t)$$

- discrete model

$$dV/dt = f(V)$$

- continuous model

Equations:

$$\frac{du_i}{dt} = -u_i + \sum_{j=1}^N w_{ij} V_j + I_i$$

$$V_i = g(\lambda u_i) = \tanh(\lambda u_i)$$

Lyapunov or “Energy” Function:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} V_i V_j + \sum_{i=1}^N \int_0^{V_i} g^{-1}(V) dV - \sum_{i=1}^N I_i V_i$$

Neural Networks

1983: Terry Sejnowski's and Geoffrey Hinton's Boltzmann machine

$$P(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{Z}$$

> $E(\mathbf{x})$: Energy function

> Z : partition function where $\sum_{\mathbf{x}} P(\mathbf{x}) = 1$



1986: Paul Smolensky's Restricted Boltzmann machine



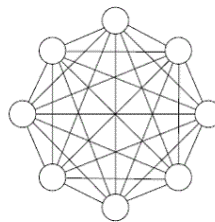
Hopfield networks

Multidirectional data flow

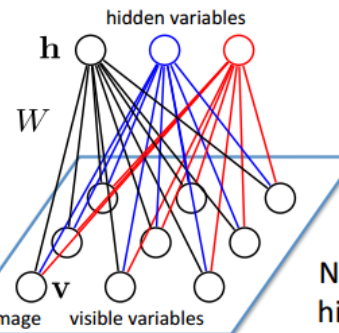
Total integration between input and output data

All neurons are linked between themselves

Trained with or without supervision



Restricted Boltzmann Machines



Boltzmann distribution:

$$P(\mathbf{v} = \mathbf{v}, \mathbf{h} = \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h}))$$

$$Z(\theta) = \sum_{\mathbf{h}, \mathbf{v}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta))$$

No interaction between hidden variables

Neural Networks

1985: Judea Pearl's "Bayesian Networks"
(‘belief networks’)



$$P(C, S, R, W, F) = P(C) P(S|C) P(R|C) P(W|R, S) P(F|R)$$

$$P(C, F) = \sum_S \sum_R \sum_W P(C, S, R, W, F)$$

$$P(F|C) = P(C, F) / P(C)$$

E.g.:

Stanford's Quick Medical Reference-
Decision Theoretic project or QMR-DT (1991)
NASA Vista project (1992)

An Empirical Analysis of
Likelihood-Weighting
Simulation on a Large, Multiply
Connected Medical Belief Network

M Shwe et al. Comput Biomed Res 24 (5), 453-475. 10 1991

We are developing a probabilistic reformulation of the Quick
Medical Reference (QMR) system.

Vista Goes Online: Decision-Analytic Systems for
Real-Time Decision-Making in Mission Control

Matthew Barry
Propulsion Systems Section
NASA/Johnson Space Center DF63
Houston, TX 77058

Eric Horvitz (PI), Corinne Ruokangas, Sampath Srinivas
Information and Decision Sciences
Rockwell International Science Center
444 High Street
Palo Alto, CA 94301

Neural Networks

Backpropagation

1982: David Parker (Stanford): learning-logic

1985: Yann LeCun (France)



COGNITIVA 85

Paris, 4-7 Juin 1985

A LEARNING SCHEME FOR ASSYMETRIC THRESHOLD NETWORK.

UNE PROCEDURE D'APPRENTISSAGE POUR RESEAU A SEUIL ASSYMETRIQUE.

YANN LE CUN

Ecole Supérieure d'Ingénieurs

en Electrotechnique et Electronique



Neural Networks

Backpropagation

1986: David Rumelhart (San Diego)



- Rummelhart network
- Neurons arranged in layers, each neuron linked to neurons of the neighboring layers, but no links within the same layer
- Requires training with supervision

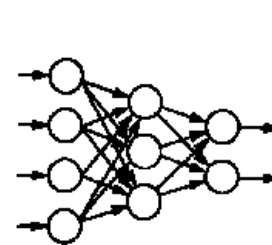
equations of backpropagation

$$\delta^L = \nabla_a C \odot \sigma'(z^L)$$

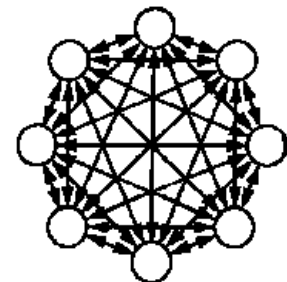
$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$$

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l$$

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$$



RUMMELHART



HOPFIELD

Neural Networks

Gradient Descent Methods

Real Time Recurrent Learning (1987, Tony Robinson & Frank Fallside)

Back Propagation Through Time (1988, Paul Werbos)

Fast Forward Propagation (1991, CalTech)

Green Function (1992, University of Maryland)

Block Update (1989, Ronald Williams)



Real-time recurrent learning (RTRL)

$$\frac{\partial x_i[t]}{\partial W_{kl}^{xx}} = g'(\Xi_i[t]) \left(\delta_{ik} x_l[t] + \sum_{j=1}^{n_X} W_{ij}^{xx} \frac{\partial x_j[t-1]}{\partial W_{kl}^{xx}} \right),$$

Neural Networks

Gradient Descent Methods

Gradient Descent Methods

Method	Description	Advantages	Disadvantages	$O(\cdot)$
RTRL	Computing error gradient after obtaining gradients of the network states w.r.t weights at time t in terms of those at time $t - 1$	- online updating of weights - suitable for online adaption property applications	- large computational complexity	$O(N^4)$
BPTT	Unfolding time iterations into layers with identical weights converts the recurrent network into an equivalent feedforward network, suitable for training with back-propagation method.	- computationally efficient - suitable for offline training	- not practical for online training	$O(N^2)$
FFP	Recursive computing of boundary conditions of back-propagated gradients at time $t = 1$.	- on-line technique - solving the gradient recursion forward in time, rather than backwards.	- more computational complexity than BPTT method	$O(N^3)$
GF	Computing the solution using the sought error gradient based on the recursive equations for the output gradients and a dot product.	- improving RTRL computational complexity - online method	- more computational complexity than BPTT method	$O(N^3)$
BU	Updating the weights every $O(N)$ data points using some aspects of the RTRL and BTT methods.	- online method	- more computational complexity than BPTT method	$O(N^3)$

Hojjat Salehinejad et al

Neural Networks

Gradient Descent Methods

Aiya-Parlos (2000, Amir Atiya & Alexander Parlos)
unifies all of them

$$\gamma(1) = -D^{-1}(0)e^T(1) \quad (60a)$$

$$\gamma(2) = -D^{-1}(1)e^T(2) + We^T(1) \quad (60b)$$

$$\vdots$$

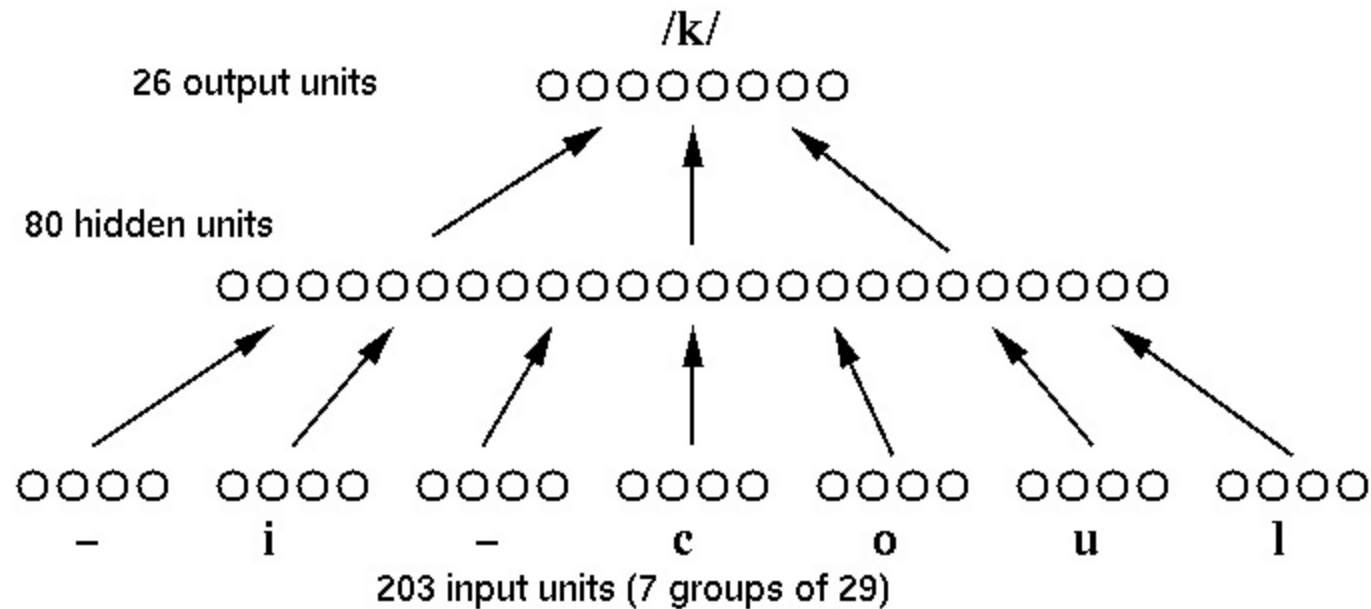
$$\gamma(K) = -D^{-1}(K-1)e^T(K) + We^T(K-1), \quad (60c)$$

$$V(K) = \epsilon I + \sum_{k=0}^{K-1} x(k)x^T(k), \quad (64)$$

$$\Delta W = \eta \left[\sum_{k=1}^K \gamma(k)x^T(k-1) \right] V^{-1}(K). \quad (65)$$

Neural Networks

1986: Terry Sejnowski's NETtalk (based on backpropagation)



Neural Networks

1986: Hinton and Sejnowski organize the first
"Connectionist Summer School" at CMU



Neural Networks

1987: Bernardo Huberman



Artificial Intelligence

Volume 33, Issue 2, October 1987, Pages 155-171

Phase transitions in artificial intelligence systems

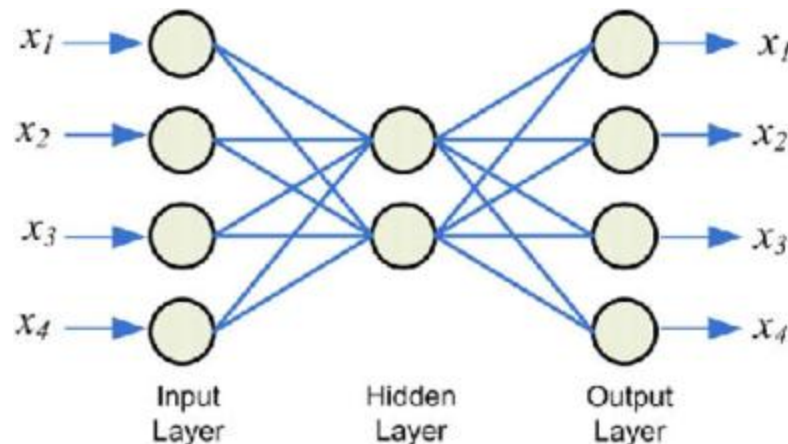
Bernardo A. Huberman, Tad Hogg



Neural Networks

Unsupervised Learning

1986: David Zipser's "autoencoder", an unsupervised neural network that is trained by backpropagation to output the input, or a very close approximation of it: autoencoders are actually powerful models for capturing characteristics of data.



Neural Networks

Unsupervised Learning

1986: Jeanny Hault's and Christian Jutten's independent component analysis



1987: Dana Ballard uses unsupervised learning to build representations layer by layer (deep learning ante-litteram)



1988: Ralph Linsker's infomax method (information theory + neural networks)

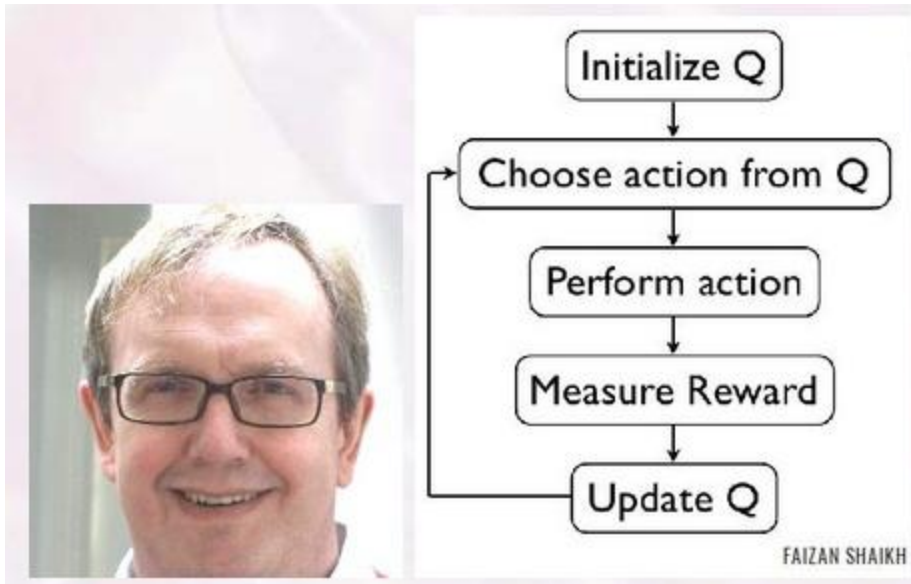


1988: Mark Plumbey (information theory + neural networks)

Neural Networks

Reinforcement Learning

1989: Chris Watkins' Q-learning



Neural Networks

Reinforcement Learning

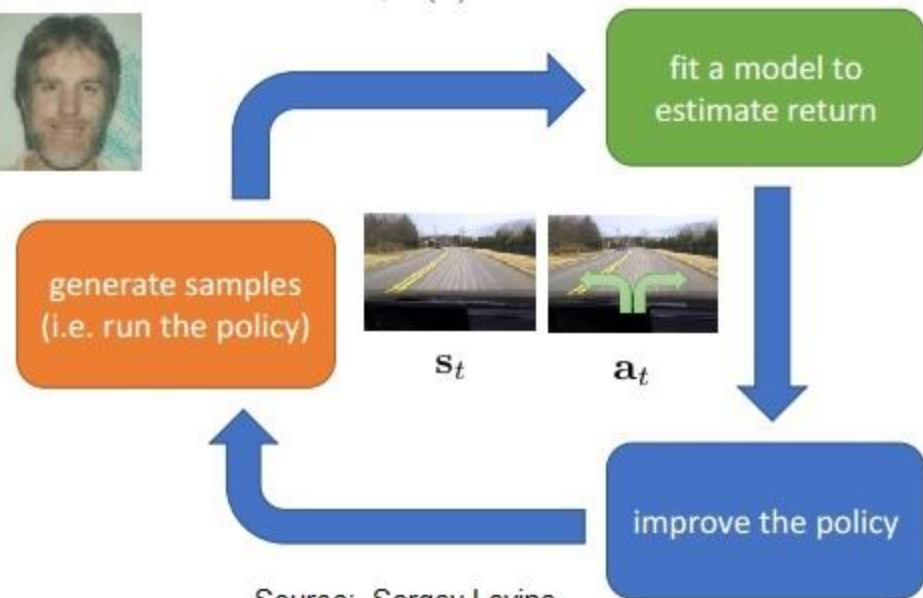
1992: Long-ji Lin's "experience replay"

1992: Ron Williams' REINFORCE, a "policy gradient method"



REINFORCE algorithm:

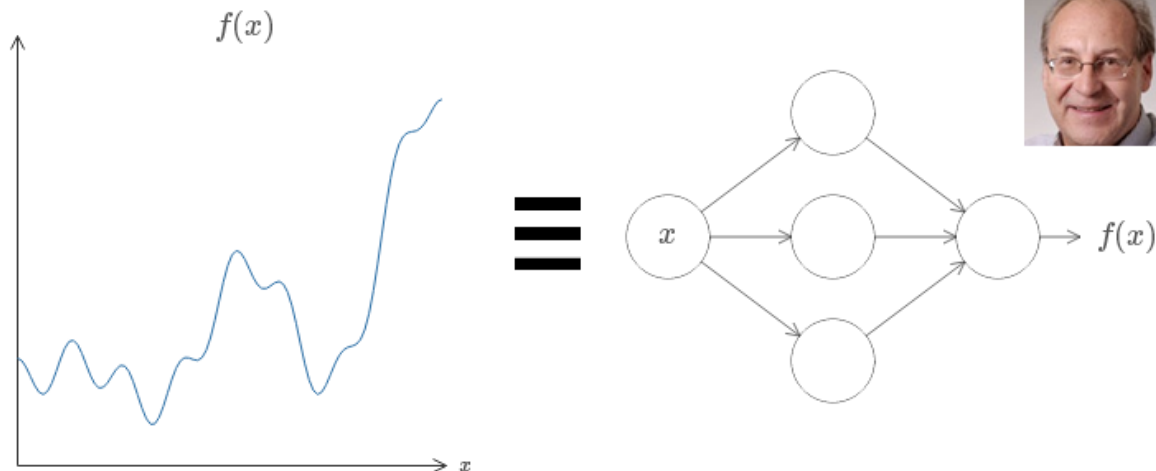
1. sample $\{\tau^i\}$ from $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ (run the policy)
2. $\nabla_\theta J(\theta) \approx \sum_i (\sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)) (\sum_t r(\mathbf{s}_t^i, \mathbf{a}_t^i))$
3. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$



Source: Sergey Levine

Neural Networks

1989: George Cybenko proves that neural networks can approximate continuous functions



Math. Control Signals Systems (1989) 2: 303–314

**Mathematics of Control,
Signals, and Systems**

© 1989 Springer-Verlag New York Inc.

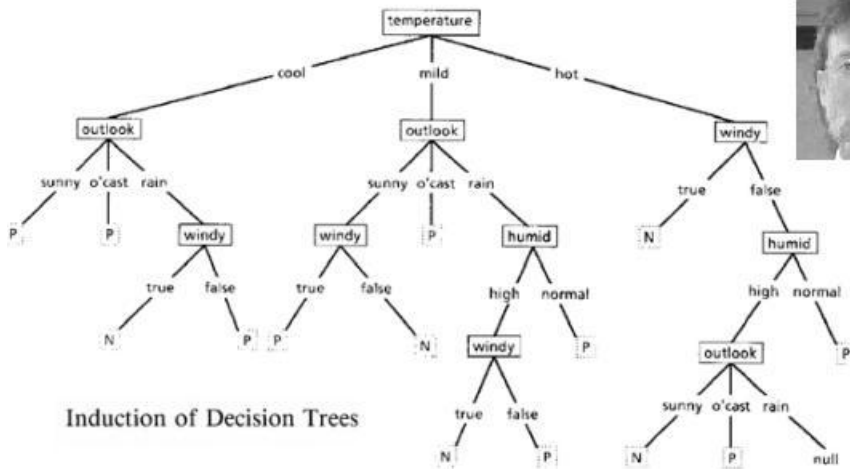
Approximation by Superpositions of a Sigmoidal Function*

G. Cybenko†

Classifiers

1985: Ross Quinlan's ID3

1993: Ross Quinlan's C4.5



Classifiers



Corinna Cortes

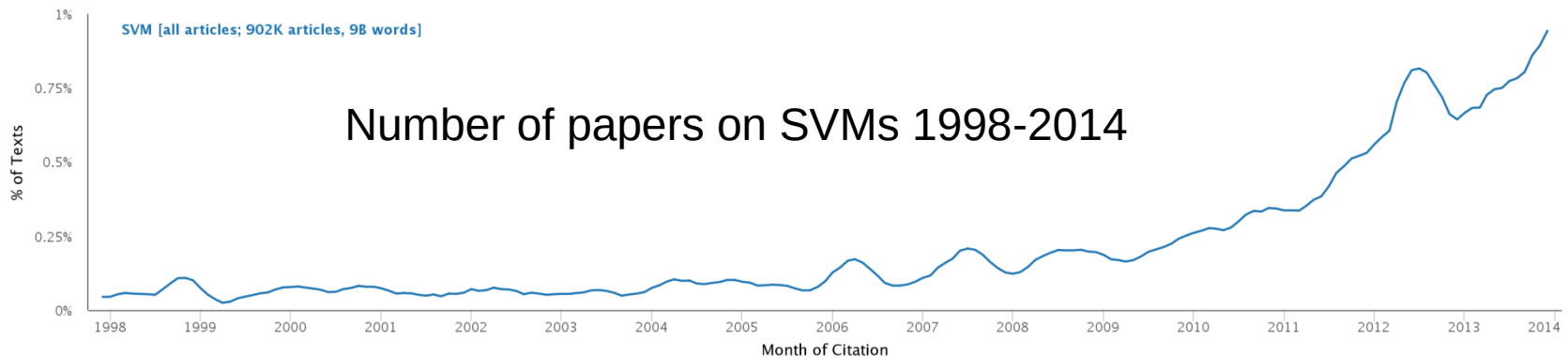
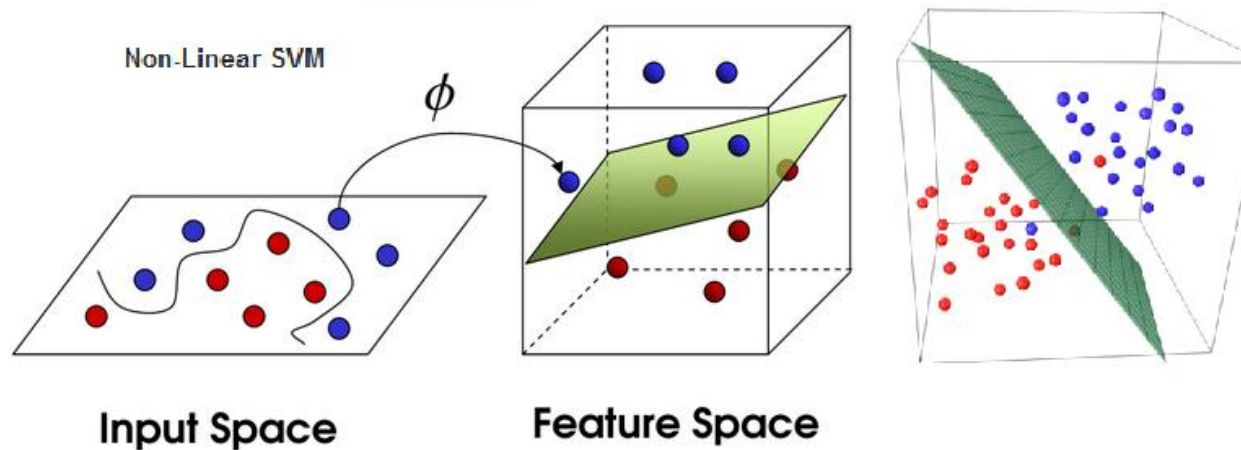
Isabelle Guyon

Vladimir Vapnik

- The support-vector machine
 - 1963: Vladimir Vapnik and Alexey Chervonenkis
 - 1975: Tomaso Poggio's polynomial kernel
 - 1987: Marc Mezard and Werner Krauth's minover
 - 1991: Isabelle Guyon pattern classification with SVMs
 - 1991: Corinna Cortes's soft-margin classifier
 - 1998: Thorsten Joachims text classifier

Classifiers

- The support-vector machine



Classifiers

1990: Robert Schapire's boosting

1993: Ross Quinlan's C4.5

1995: Tin-Kam Ho's random forests

Given: $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = 1/m$.

For $t = 1, \dots, T$:

- Train weak learner using distribution D_t .
- Get weak hypothesis $h_t : X \rightarrow \{-1, +1\}$ with error



$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right).$$

AdaBoost



$$P(c|v_j(x)) = \frac{P(c, v_j(x))}{\sum_{l=1}^n P(c_l, v_j(x))}$$

$$g_c(x) = \frac{1}{t} \sum_{j=1}^t \hat{P}(c|v_j(x))$$

Classifiers

1995: No Free Lunch theorems (David Wolpert)
No learning algorithm can excel at learning everything

The Lack of A Priori Distinctions Between Learning Algorithms

David H. Wolpert

*The Santa Fe Institute, 1399 Hyde Park Rd.,
Santa Fe, NM, 87501, USA*



March 1996

Neural Computation 8, 1341–1390 (1996) © 1996 Massachusetts Institute of Technology

No Free Lunch Theorems for Optimization

No Free Lunch Theorems for Search

David H. Wolpert (dhw@santafe.edu)
William G. Macready (wgm@santafe.edu)
The Santa Fe Institute
1399 Hyde Park Rd.
Santa Fe, NM, 87501, USA

February 6, 1995

David H. Wolpert
IBM Almaden Research Center
William G. Macready
Santa Fe Institute

December 31, 1996

Classifiers

No Free Lunch in continuous domains?

Auger/Teytaud: Yes

Vose: No

Continuous lunches are free!

Submitted on 19 Sep 2007

Anne Auger

TAO Team - INRIA Futurs
LRI - Paris-Sud University

Olivier Teytaud

TAO Team - INRIA Futurs
LRI - Paris-Sud University

Reinterpreting No Free Lunch

2009

Jon E. Rowe

School of Computer Science, The University of Birmingham, Edgbaston, Birmingham,
B15 2TT, United Kingdom

J.E.Rowe@cs.bham.ac.uk

M. D. Vose

Department of Electrical Engineering and Computer Science, University of Tennessee,
Knoxville, TN, 37996-3450, USA

vose@eecs.utk.edu

Alden H. Wright

Department of Computer Science, The University of Montana-Missoula Missoula, MT
59812, USA

aldenwright@fastmail.us

Abstract

The result is a general set-theoretic version of NFL
which speaks to symmetries when arbitrary domains and co-domains are involved.

An Artificial Scientist?

- Can an Artificial Intelligence win a Nobel Prize?
- or file a patent for an invention?



UNITED STATES PATENT OFFICE.

THOMAS A. EDISON, OF LLEWELLYN PARK, NEW JERSEY.

PHONOGRAPH.

SPECIFICATION forming part of Letters Patent No. 465,972, dated December 29, 1891.

Application filed November 18, 1889. Serial No. 330,783. (No model.)

An Artificial Scientist?

- Lawrence Hunter (2003 AAAI talk): the publication of a paper in a peer-reviewed scientific journal is a better test for human-level intelligence than the Turing Test.

Discovery Informatics: Can AI Do Science?

AAAI

Robert S. Engelmore
Memorial Lecture Award

2003



Lawrence Hunter,



"I proposed that the creation and publication of a peer-reviewed scientific journal article be used as a test for computational intelligence"

COMMUNICATIONS
OF THE
ACM
Discovery informatics

By William W. Agresti

August 2003 (Vol. 46, No. 8)



JOHNS HOPKINS
CAREY BUSINESS SCHOOL

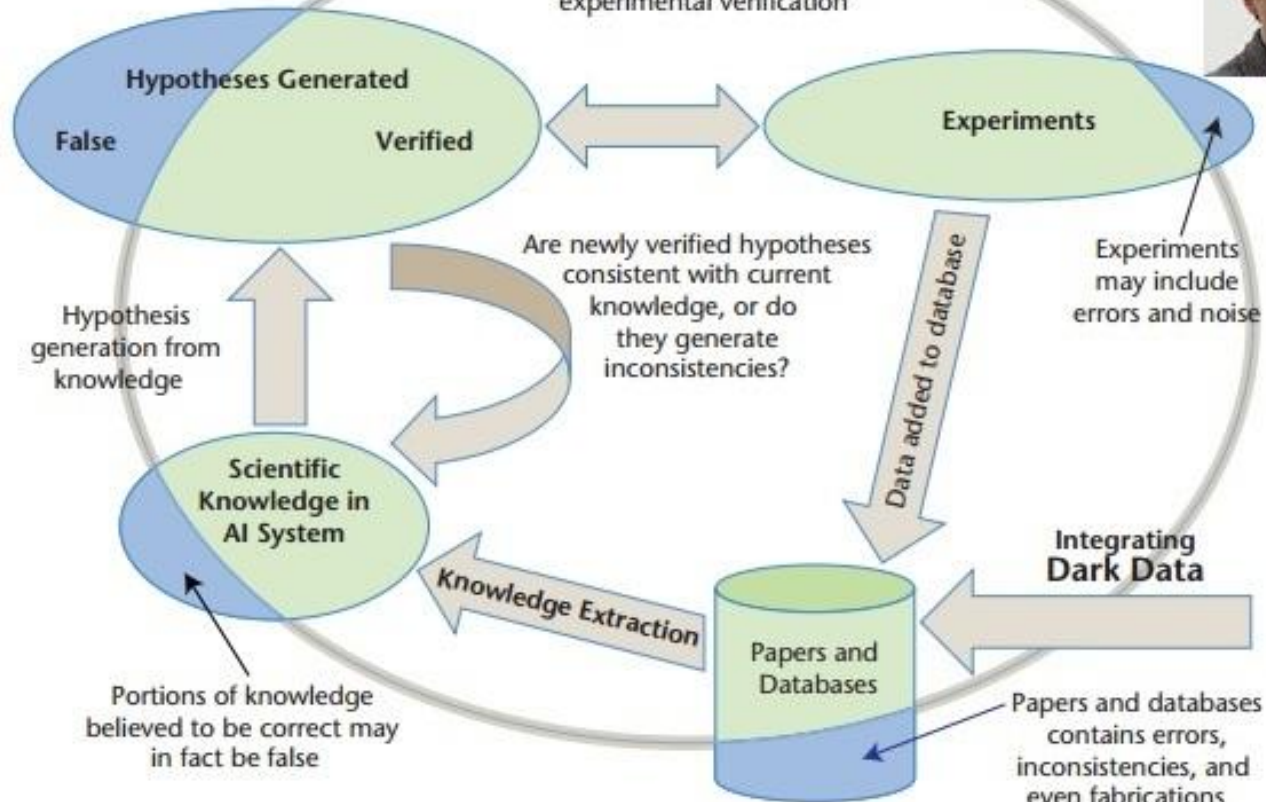
An Artificial Scientist?

- Hiroaki Kitano (October 2016):
"Artificial Intelligence to Win the Nobel Prize and Beyond"
- Grand challenge for AI: to develop a system capable of scientific research in the biomedical sciences that can discover something worthy of the Nobel Prize



An Artificial Scientist?

Entire Hypothetical Body
of Scientific Knowledge



Help!

- 2016: more than 1.2 million papers were published in life science journals alone, on top of the 25 million already in print
- A new article is being published every 30 seconds
- On average a scientist reads about 264 papers per year
- More than 70,000 papers have been published on the tumor suppressor p53



Piero Scaruffi is one of the victims of information overload

What is Scientific Discovery?

1962: Saul Amarel's "An Approach to Automatic Theory Formation"

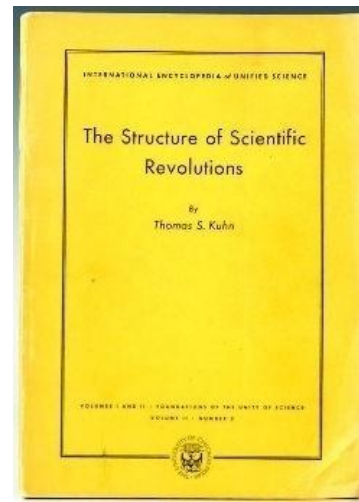
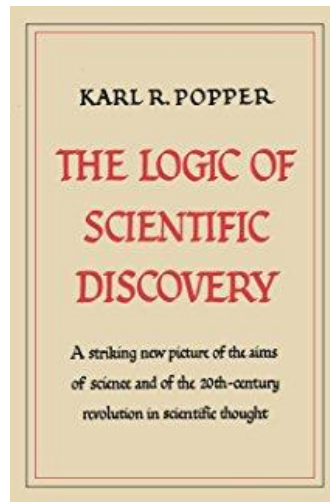


1962: Thomas Kuhn's "The Structure of Scientific Revolutions"



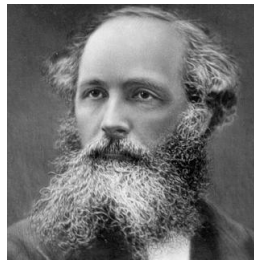
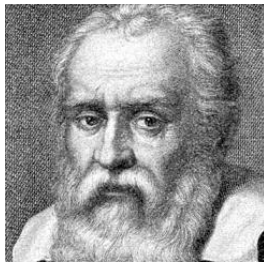
1965: Karl Popper's "The Logic of Scientific Discovery"

1966: Herbert Simon's "Scientific Discovery and the Psychology of Problem Solving"



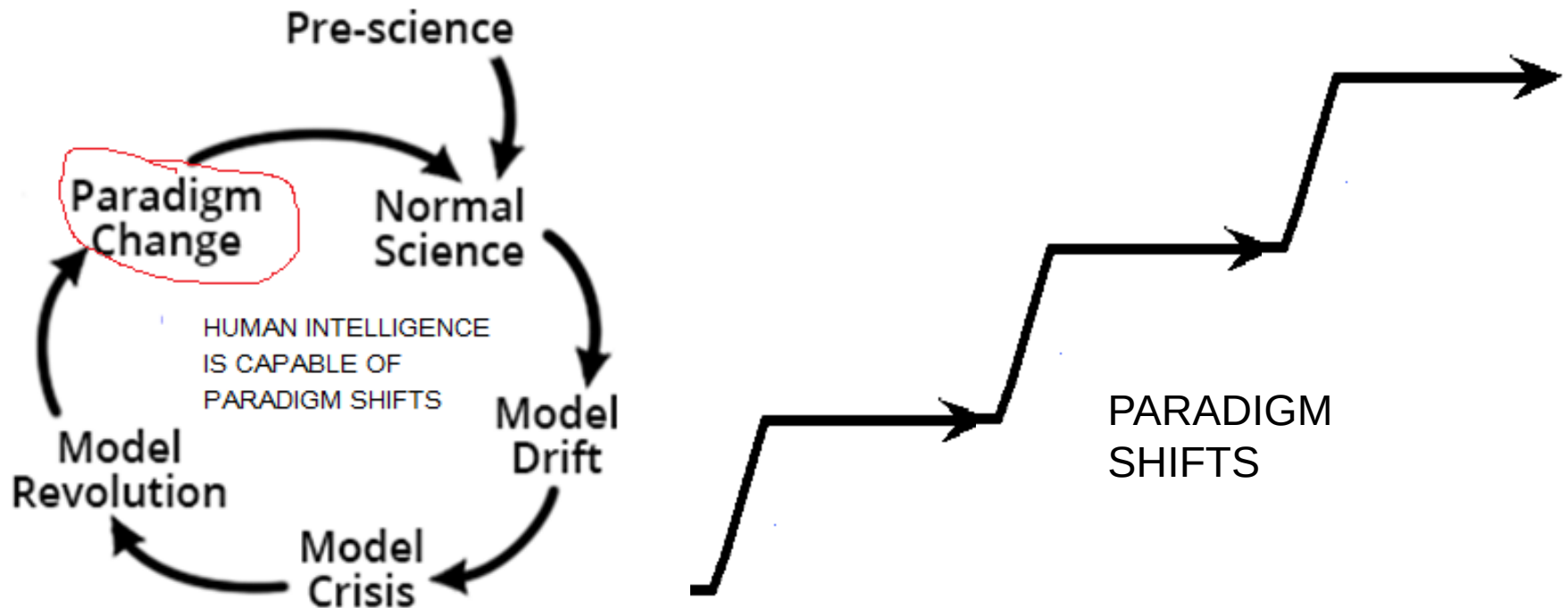
What is Scientific Discovery?

- The cognitive process that leads to novel, creative ideas:
- the **discovery** of natural laws from **experimental data**
- ... i.e. the **formation of theories** that explain those data
- ... and then the **design of experiments** to confirm those theories.



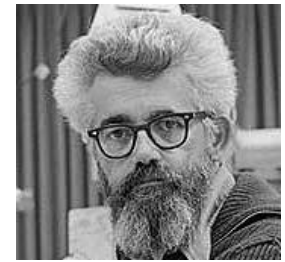
What is Scientific Discovery?

- Kuhn: The history of science is a history of "paradigm shifts", of sudden realizations that old theories were wrong and a whole new way of thinking is required, a conceptual restructuring.



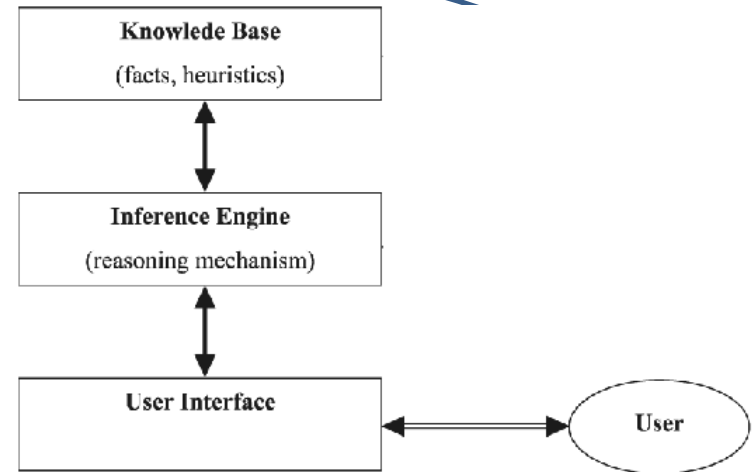


The two schools of A.I.

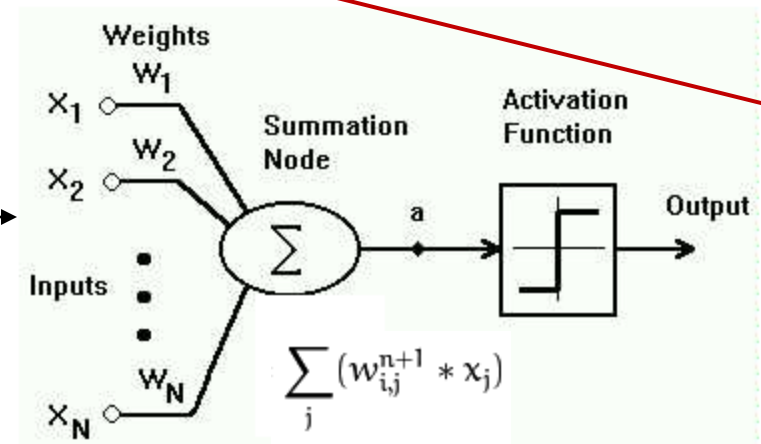
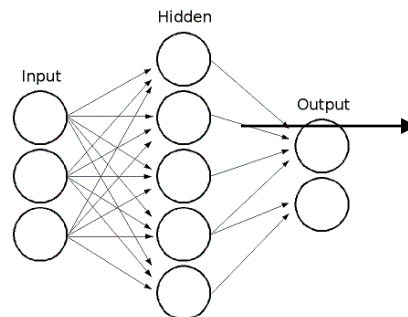
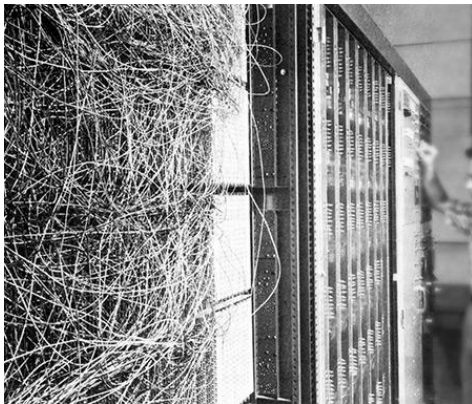


Artificial Intelligence (1956)

- Knowledge-based approach uses mathematical logic to simulate the human mind



- Neural-net approach simulates the structure of the brain



Programs for Theory Formation

1970: Ed Feigenbaum and Bruce Buchanan's Meta-Dendral

1976: Douglas Lenat's AM

1977: Pat Langley's Bacon

1978: Peter Friedland and Mark Stefik's Molgen

1986: Jan Zytkow's Fahrenheit

1986: Ryszard Michalski's Abacus

1988: Deepak Kulkarni's Kekada

1990: Raul Valdes-Perez's Mechem

1990: Michael Sims' IL



Programs for Theory Formation

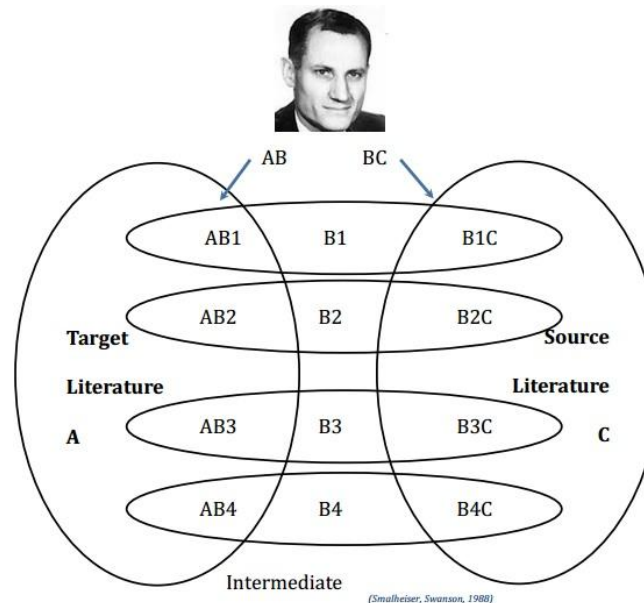
1986: Don Swanson's literature-based knowledge discovery

1,000 papers on fish oil

2,000 papers on Raynaud's syndrome

Conclusion: fish oil may prevent Raynaud's syndrome

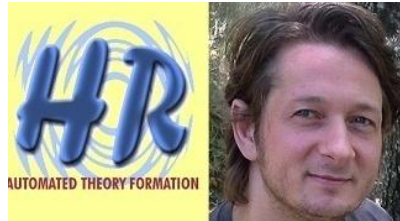
1998: Swanson's Arrowsmith



Programs for Theory Formation

1997: Faye Mitchell's Davicand

1998: Simon Colton's HR



Programs for Theory Formation



2004: Ross King's "robot scientist" Adam

The Automation of Science

WIRED 04.02.09 12:15 PM

ROBOT MAKES SCIENTIFIC DISCOVERY ALL BY ITSELF



Robot Scientist



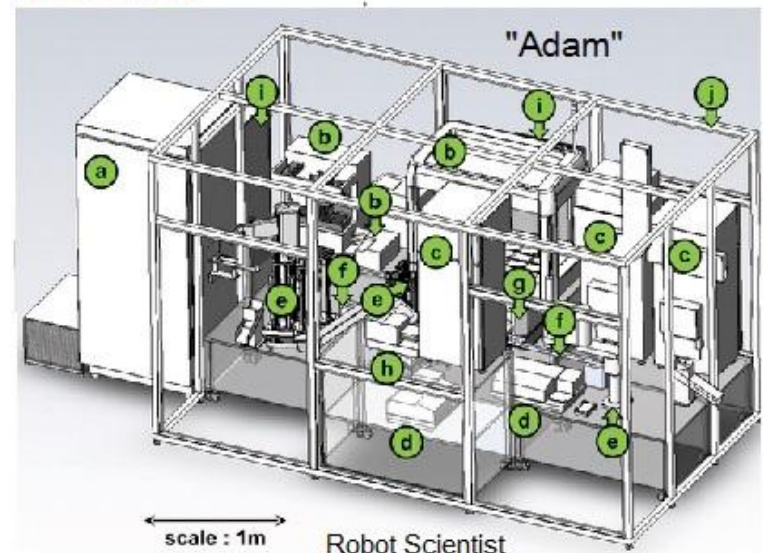
Ross King



Aberystwyth

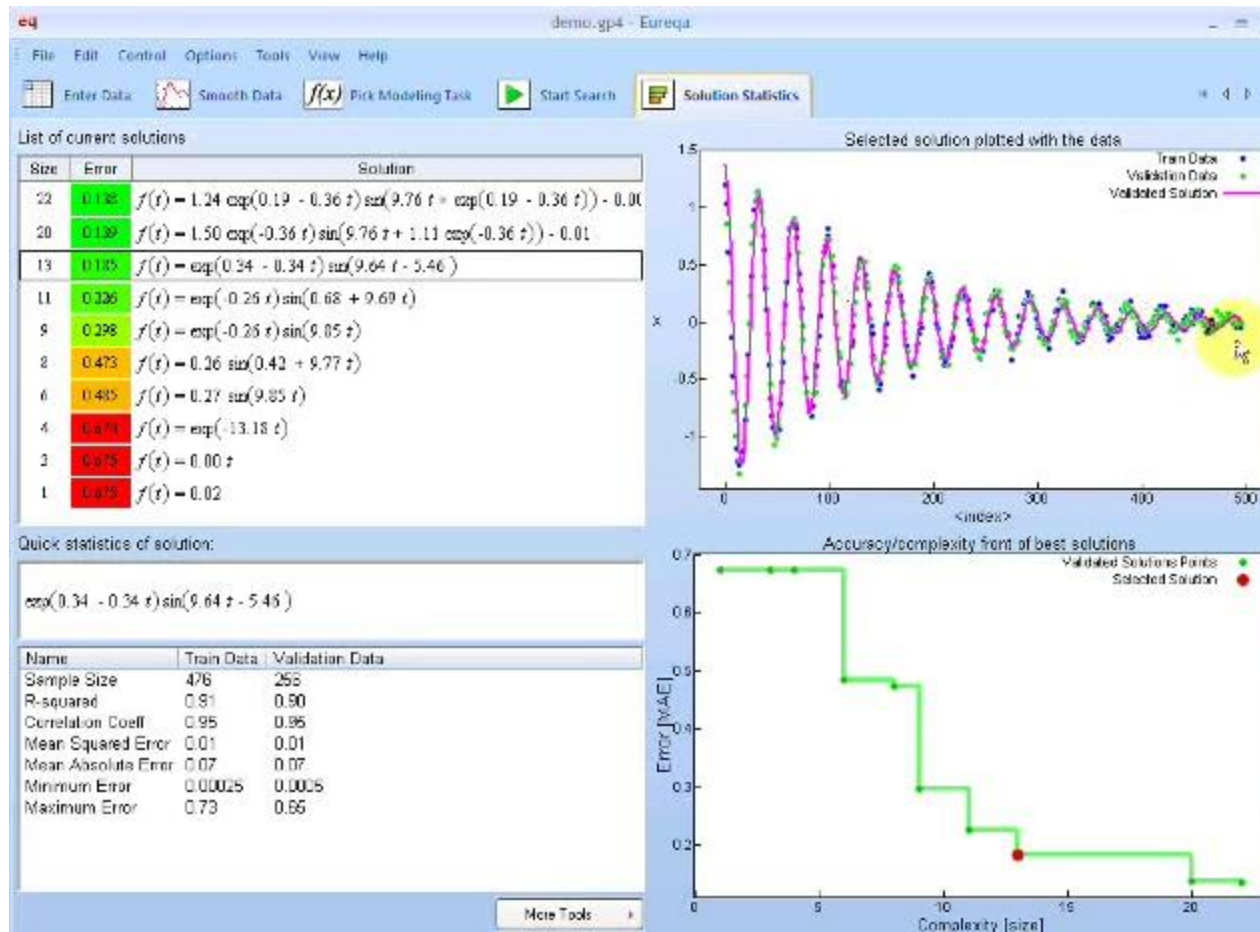
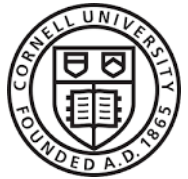
Science

Science 03 Apr 2009



Programs for Theory Formation

2007: Michael Schmidt's Eureqa



Programs for Theory Formation

2009: Andrey Rzhetsky's "machine science"



 **PLOS** | COMPUTATIONAL BIOLOGY November 6, 2009

Looking at Cerebellar Malformations through Interactomes of Mice and Humans

Ivan Iossifov, Raul Rodriguez-Esteban, Ilya Mayzus, Kathleen J. Millen, Andrey Rzhetsky

Science 23 Jul 2010:
Vol. 329, Issue 5990, pp. 399-400

Machine Science

James Evans and Andrey Rzhetsky
University of Chicago



Programs for Theory Formation

2009: Joshua Tenenbaum, Thomas Griffiths, Charles Kemp
Probabilistic model of theory formation



A probabilistic model of theory formation

Received 10 June 2008

Revised 29 June 2009

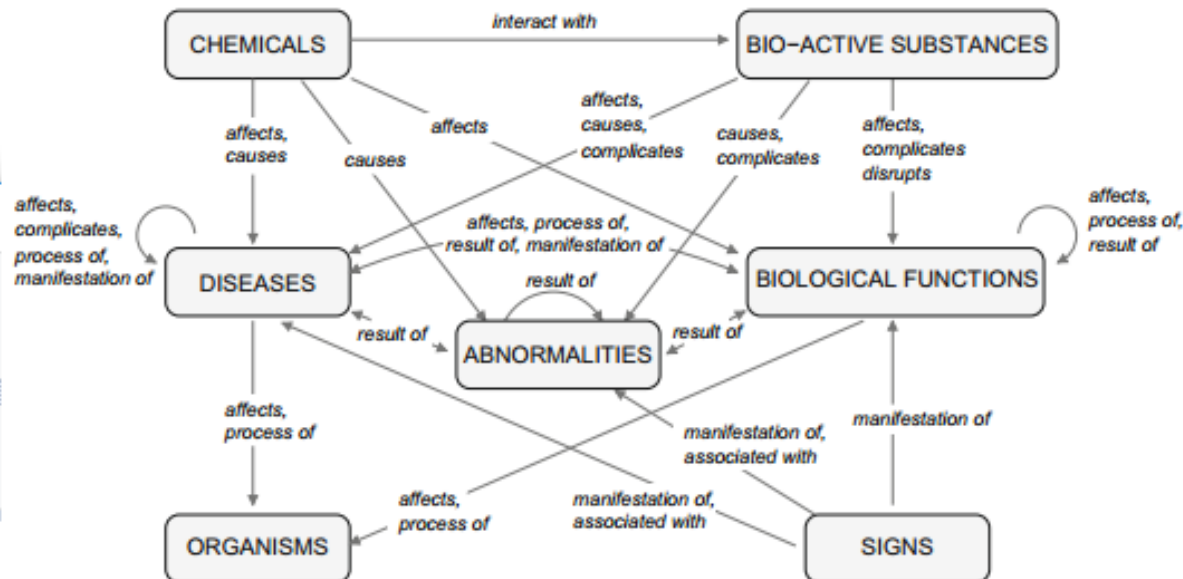
Charles Kemp^{a,*}, Joshua B. Tenenbaum^b, Sourabh Niyogi^c, Thomas L. Griffiths^d

^aDepartment of Psychology, Carnegie Mellon University, United States

^bDepartment of Brain and Cognitive Sciences, Massachusetts Institute of Technology, United States

^cDepartment of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, United States

^dDepartment of Psychology, University of California, Berkeley, United States



Programs for Theory Formation

Joshua Tenenbaum

RESEARCH ARTICLES

COGNITIVE SCIENCE

Human-level concept learning through probabilistic program induction

Brenden M. Lake,^{1*} Ruslan Salakhutdinov,² Joshua B. Tenenbaum³

December 10, 2015

Programs for Theory Formation

2014: IBM and Baylor College: KnIT

2015: Michael Levin's program



Computer A.I. Solves 120-Year-Old Biological Mystery

JUNE 9, 2015

A computer has developed a new scientific theory using only its artificial intelligence, without any help from humans. Continue reading →



Michael Levin

Tufts

Programs for Theory Formation

The search for relevance

Sparrho

Meta

Sparrho

**Find the science that
matters to you**

Discover the latest and most relevant research from
more than 60,000,000 articles in over 45,000
journals

**MACHINE LEARNING APPLIED TO
SCIENTIFIC DISCOVERY – META &
THE CHAN ZUCKERBERG
INITIATIVE**

SHANGHAI 1

SEPTEMBER 12, 2017, 15:30-16:30



Jan 23, 2017



Chan Zuckerberg Initiative acquires and will
free up science search engine Meta

Neural Networks

1989: Alex Waibel's time-delay networks

IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. 37, NO. 3, MARCH 1989

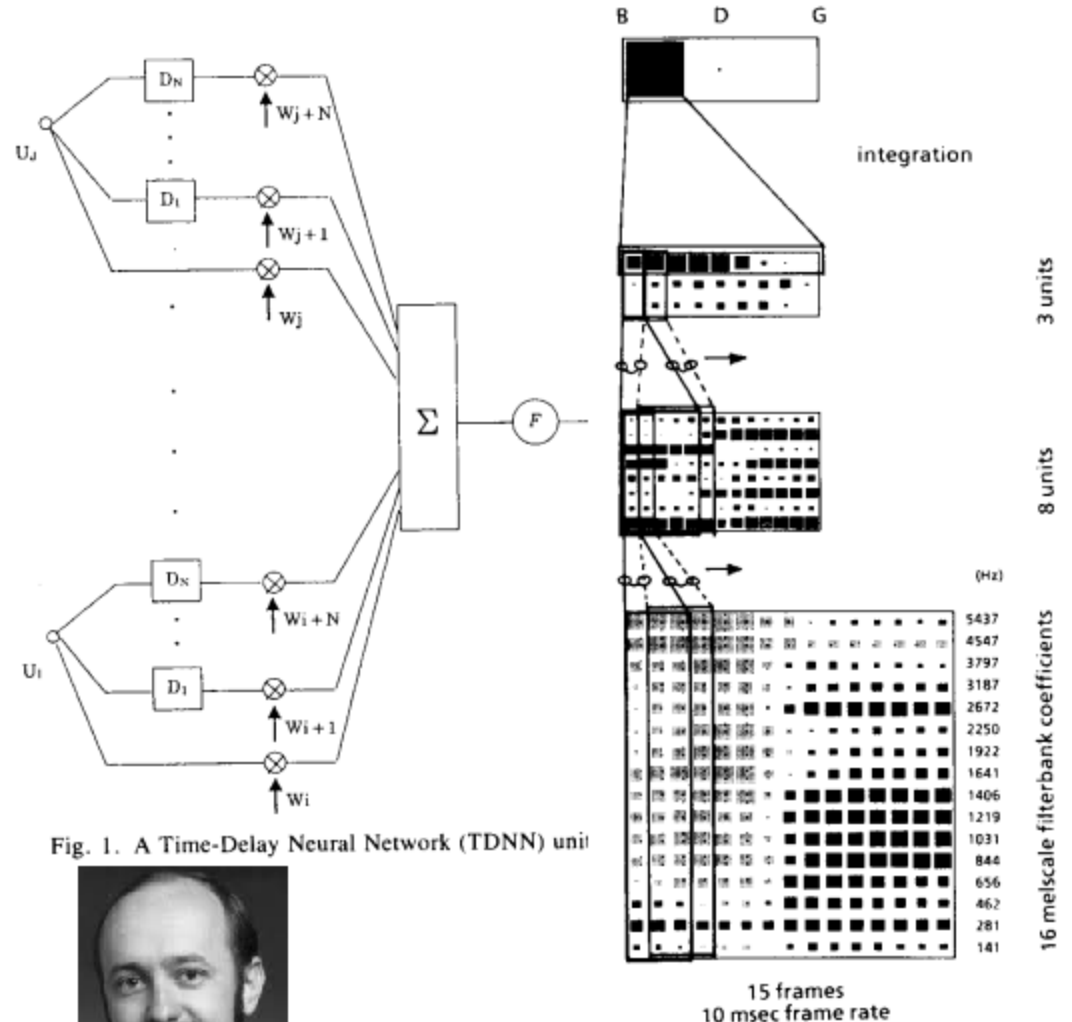


Fig. 1. A Time-Delay Neural Network (TDNN) unit



Alexander Waibel

Fig. 2. The architecture of the TDNN.

Neural Networks

1989: Yann LeCun 's convolutional neural network LeNet-1
(backpropagation applied to convolutional networks)

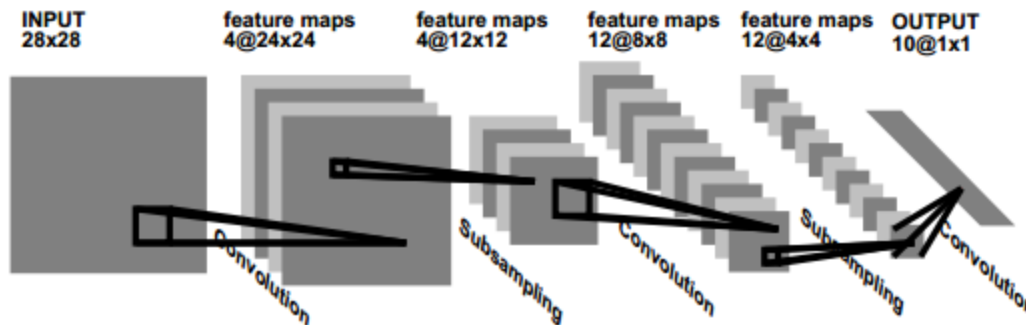


Figure 1: Architecture of LeNet 1

*Handwritten Digit Recognition with a
Back-Propagation Network*

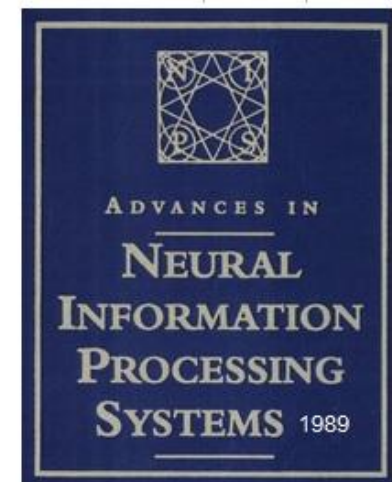
Y. Le Cun, B. Boser, J. S. Denker, D. Henderson,
R. E. Howard, W. Hubbard, and L. D. Jackel
AT&T Bell Laboratories, Holmdel, N. J. 07733

Given functions $x(t)$ and $w(t)$, their convolution is a function $s(t)$

$$s(t) = \int x(a)w(t - a)da$$

Written as

$$s = (x * w) \quad \text{or} \quad s(t) = (x * w)(t)$$



Neural Networks

1998: Yann LeCun 's LeNet-5

PROC. OF THE IEEE, NOVEMBER 1998

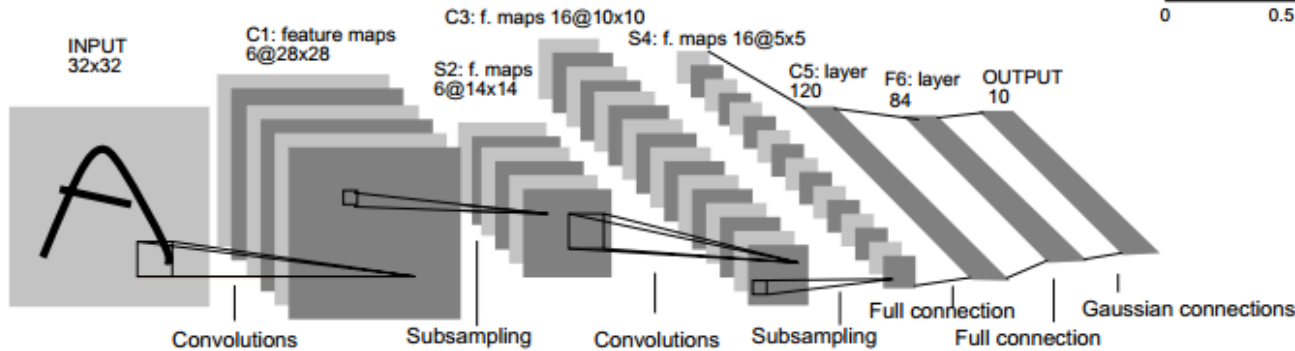
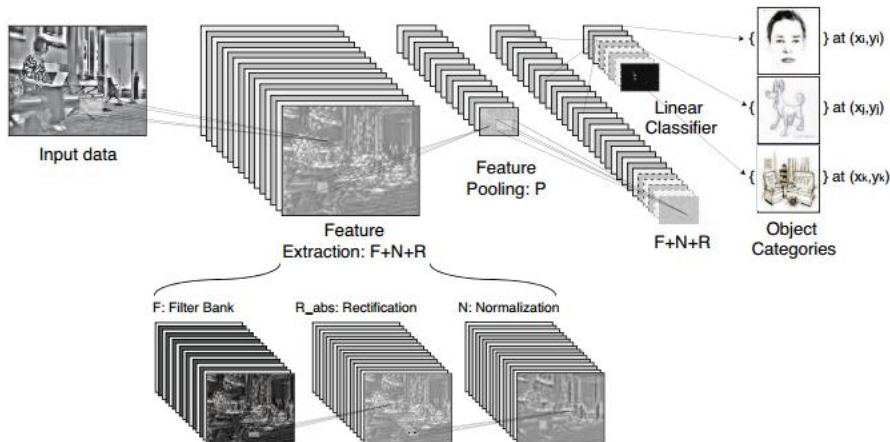


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition.

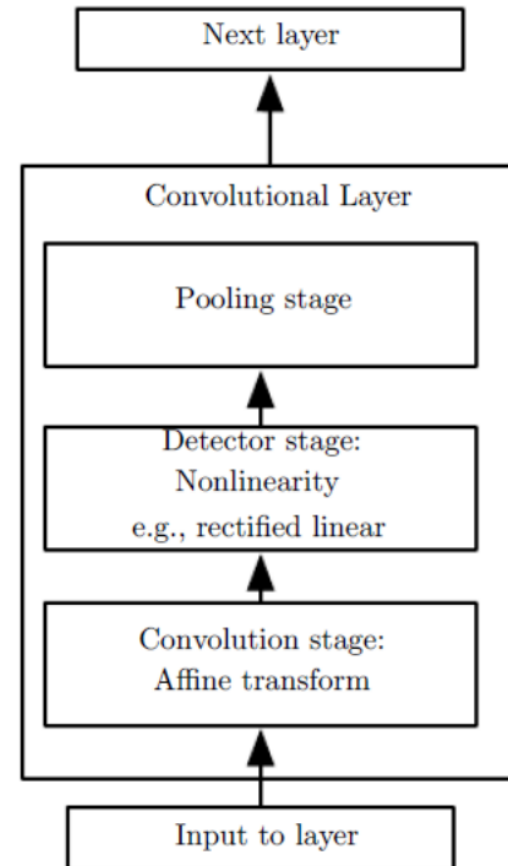
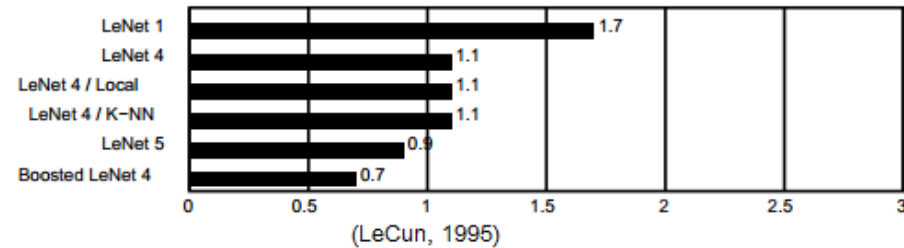
Convolutional Network



Architecture of a typical convolutional network for object recognition

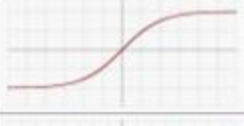
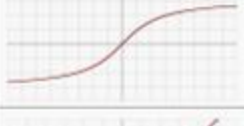
From a paper by Yann LeCun

Figure 2: error rate on the test set (%).



Neural Networks

Activation functions: binary, sigmoid, tanh, ReLu, etc

Identity		$f(x) = x$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$
ArcTan		$f(x) = \tan^{-1}(x)$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$

Neural Networks

Stateful networks: RNN

- 1991: Hava Siegelmann's and Eduardo Sontag's theorem (RNN=Turing Machine)

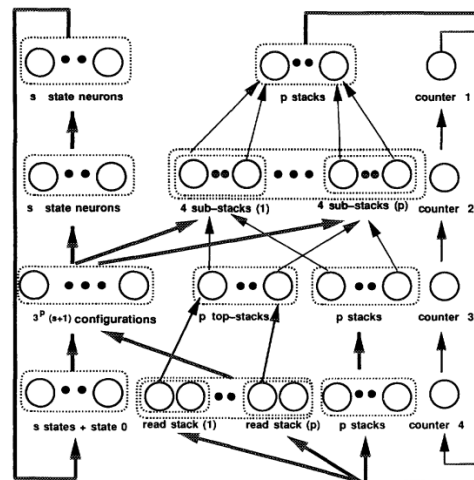


FIG. 1. The universal network.

On the Computational Power of Neural Nets

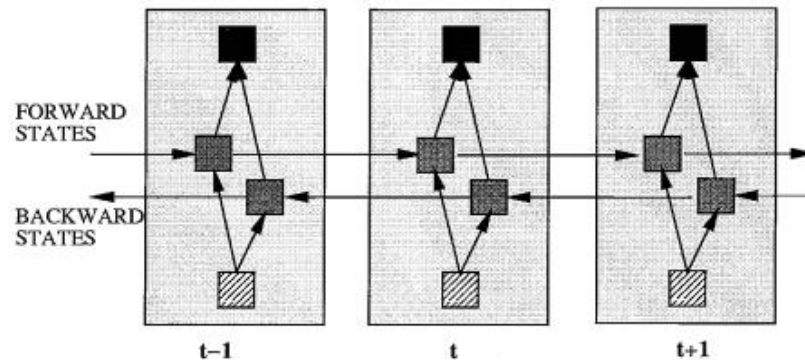
HAVA T. SIEGELMANN¹ EDUARDO D. SONTAG²

Received February 4, 1992; revised May 24, 1993

Neural Networks

Stateful networks: RNN

- 1996: Mike Schuster and Kuldip Paliwal's bidirectional recurrent neural networks



$$y(t) = \frac{1}{10} \sum_{\Delta t=-10}^{-1} x(t + \Delta t) \cdot \left(1 - \frac{|\Delta t|}{10}\right) + \frac{1}{20} \sum_{\Delta t=0}^{19} x(t + \Delta t) \cdot \left(1 - \frac{|\Delta t|}{20}\right).$$



Neural Networks

Stateful networks: LSTM

- 1997: Sepp Hochreiter's and Jeurgen Schmidhuber's Long Short Term Memory (LSTM) model

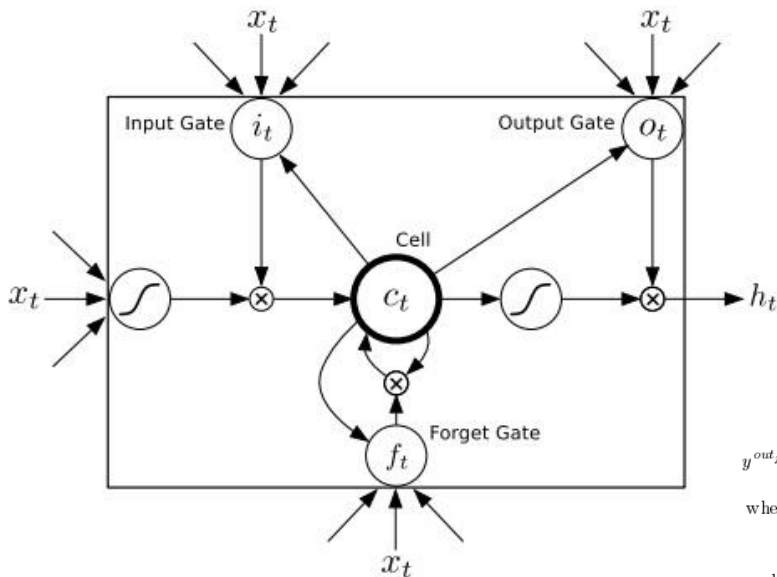


Figure 2: Long Short-term Memory Cell

Alex Graves

$$y^{out_j}(t) = f_{out_j}(net_{out_j}(t)); y^{in_j}(t) = f_{in_j}(net_{in_j}(t));$$

where

$$net_{out_j}(t) = \sum_u w_{out_j u} y^u(t-1),$$

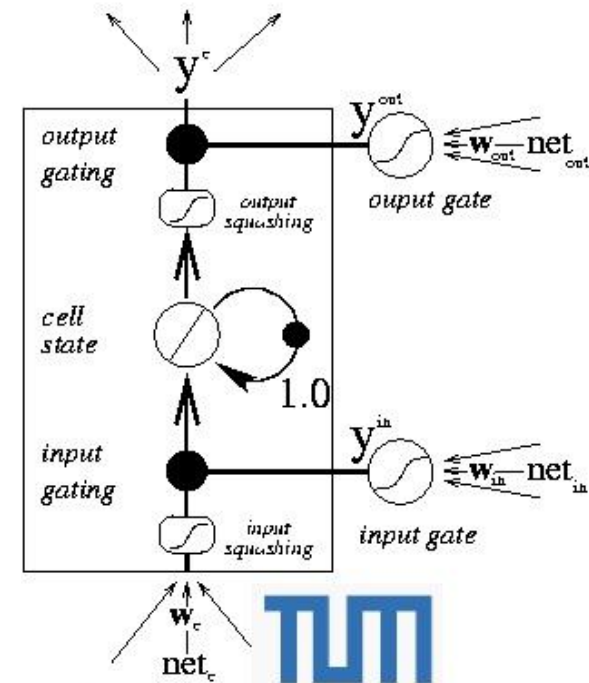
and

$$net_{in_j}(t) = \sum_u w_{in_j u} y^u(t-1).$$

We also have

$$net_{c_j}(t) = \sum_u w_{c_j u} y^u(t-1).$$

Long Short-Term Memory



Neural Networks

Stateful networks: LSTM

- 2002: Music composition
- 2006: Alex Graves' CTC for speech recognition

A First Look at Music Composition using LSTM Recurrent Neural Networks

2002

Douglas Eck
doug@idsia.ch



Jürgen Schmidhuber
juergen@idsia.ch



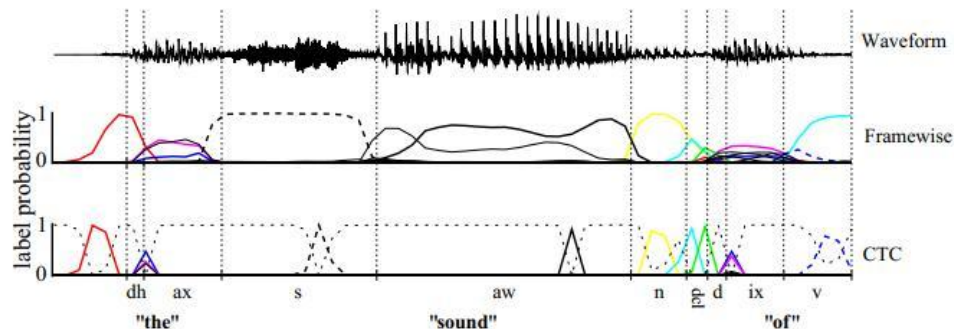
Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks

2006

Alex Graves¹
Santiago Fernández¹
Faustino Gomez¹
Jürgen Schmidhuber^{1,2}



ALEX@IDSIA.CH
SANTIAGO@IDSIA.CH
TINO@IDSIA.CH
JUERGEN@IDSIA.CH



Neural Networks

- CNN vs RNN
 - CNN: a feed forward neural network with 4 kinds of layers: convolution layers (that extract features), ReLU layer, pooling (that reduces the dimensionality) and a fully-connected layer (that classifies)
 - RNN saves the output of a layer and feeds it back to the input, i.e. RNN considers the current input and also the previously received inputs
 - CNN takes a fixed -size inputs (typically a two-dimensional matrix) and generates fixed-size outputs (for example, a number)
 - RNN can handle arbitrary input/output lengths
 - CNNs are especially good at classification problems (e.g. object recognition)
 - RNNs are especially good at processing sequential data (e.g. speech recognition)
 - Example of a hybrid: generating the caption (description) of an image (the CNN classifies the image and the RNN generates the words)

Neural Networks

- Unsupervised learning
 - 1996: Bruno Olshausen's sparse coding for autoencoders
 - 1992: Suzanna Becker's imax
 - 1995: Geoffrey Hinton's "wake-sleep" algorithm
 - 1995: Anthony Bell's improved infomax
 - 1996: Shunichi Amari's infomax with natural gradient

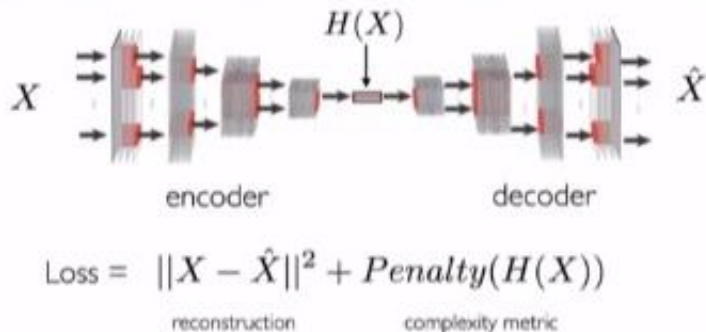


Neural Networks

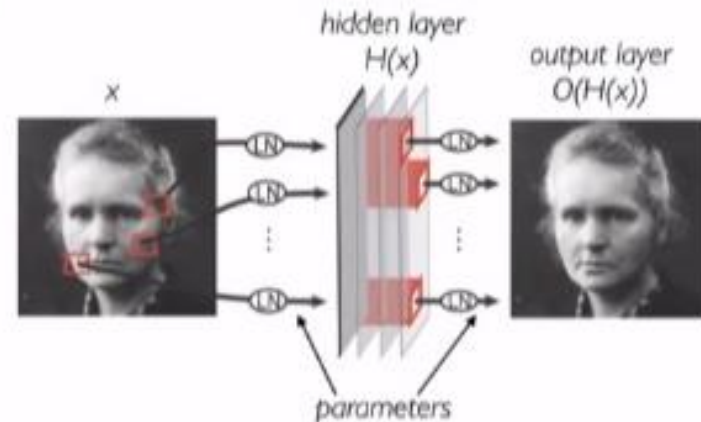
- Unsupervised learning
 - 1996: Bruno Olshausen's sparse coding for autoencoders



Unsupervised Methods: **Autoencoders**



Unsupervised Methods: **Sparse** Autoencoders



$$L(x) = |x - O(H(x))|^2 + \lambda \cdot |H(x)|$$

Sparse Coding Foldiak, Olshausen → neurons have to represent their environment, as efficiently as possible

Neural Networks

- Semi-supervised learning
 - David Yarowsky's self-training (1995)
 - Avrim Blum's and Tom Mitchell's Co-training (1998)
 - Zhi-Hua Zhou's Tri-training (2005)



Neural Networks

Reservoir Computing: deep learning made practical

- Herbert Jaeger's echo state networks (2001)
- Wolfgang Maass and Henry Markram's liquid state machines (2002)

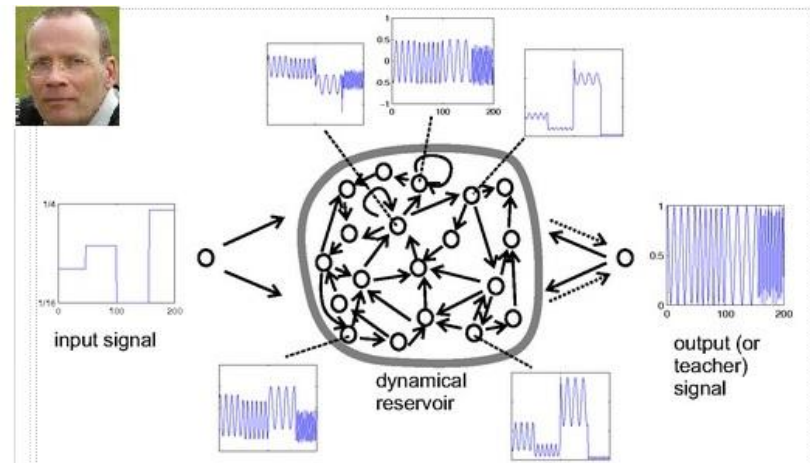


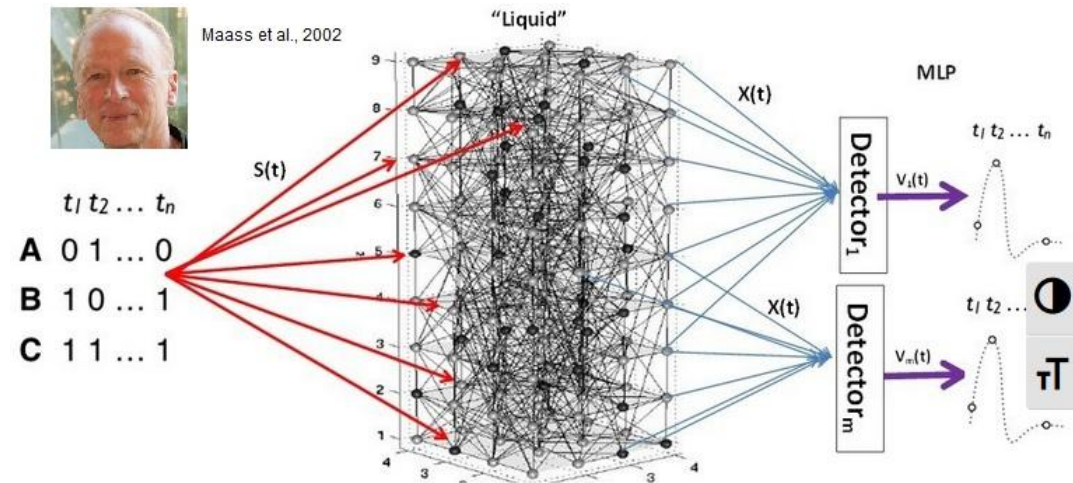
Figure 1: The basic schema of an ESN, illustrated with a tuneable frequency generator task. Solid arrows indicate fixed, random connections; dotted arrows trainable connections.

Herbert Jaeger (2007)

$$\vec{s}_n = (1 - \alpha)\vec{s}_{n-1} + \alpha f_{in}(\mathbf{W}_{in}[b_{in}; \vec{u}_n] + \mathbf{W}\vec{s}_{n-1})$$



Maass et al., 2002



1

THE LIQUID STATE MACHINE

Image Recognition

2003: Gabriela Csurka's "Visual Categorization With Bags Of Keypoints" (submitted in 2003, published in 2004)



Image Recognition

2005: Dileep George & Jeff Hawkins'
hierarchical probabilistic model of the visual
cortex

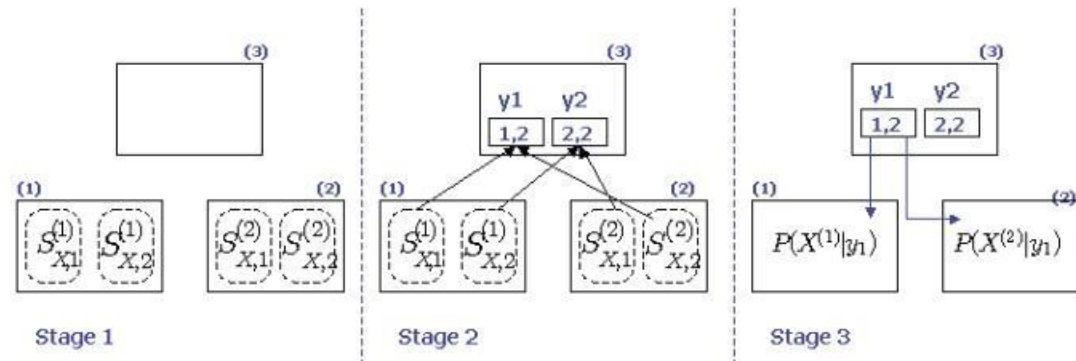
A Hierarchical Bayesian Model of Invariant Pattern Recognition in the Visual Cortex

2005



Dileep George
Department of Electrical Engineering
Stanford University and
Redwood Neuroscience Institute
Menlo Park, CA 94305

Jeff Hawkins
Redwood Neuroscience Institute
Menlo Park, CA 94305
E-mail: jhawkins@rni.org



No need for Neural Nets

1991: The Beier-Neely method to generate fake images
(used in Michael Jackson's video "Black and White")

Computer Graphics, 26, 2, July 1992

Feature-Based Image Metamorphosis

Thaddaus Beier

Silicon Graphics Computer Systems
2011 Shoreline Blvd, Mountain View CA 94043

Shawn Neely

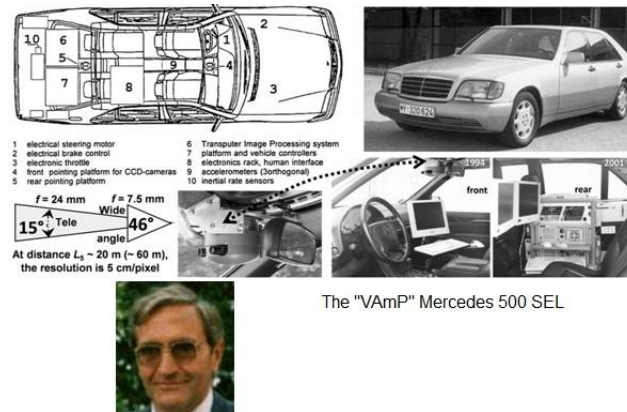
Pacific Data Images
1111 Karlstad Drive, Sunnyvale CA 94089



Figure 17
A sequence from Michael Jackson's *Black or White*

No need for Neural Nets

1994: Ernst Dickmanns' self-driving car drives more than 1,000 kms near the airport Charles-de-Gaulle in Paris



1997: IBM's "Deep Blue" chess machine beats the world's chess champion, Garry Kasparov



No need for Neural Nets

1998: Two Stanford students, Sergey Brin and Larry Page, publish the paper "The Anatomy of a Large-scale Hypertextual Web Search Engine" and launch the search engine Google

The anatomy of a large-scale hypertextual Web search engine ¹



Sergey Brin ², Lawrence Page ^{*,2}

Computer Science Department, Stanford University, Stanford, CA 94305, USA



In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at <http://google.stanford.edu/>

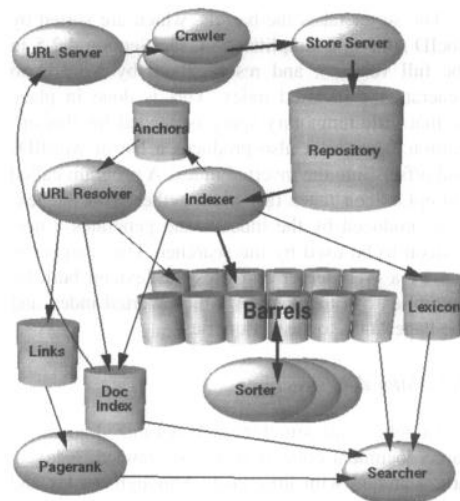
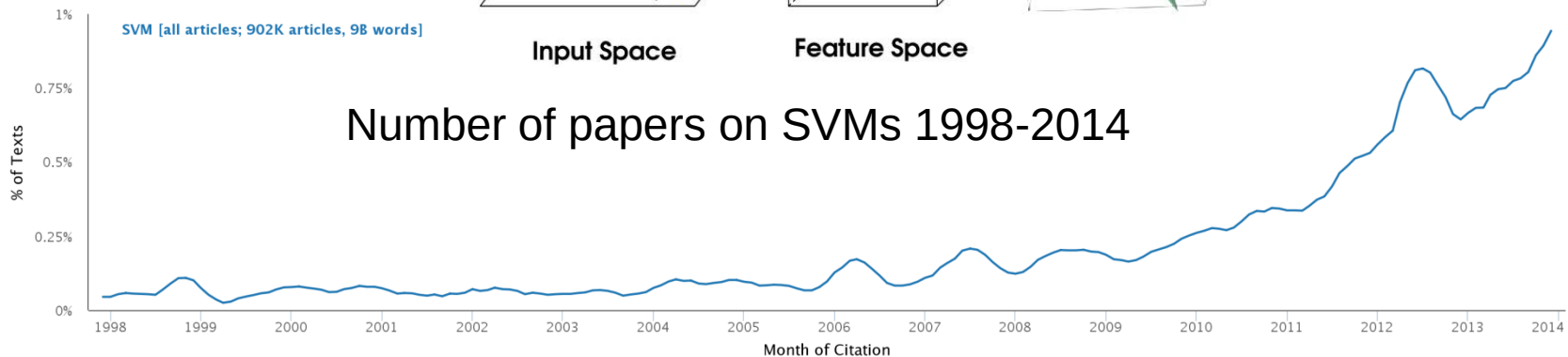
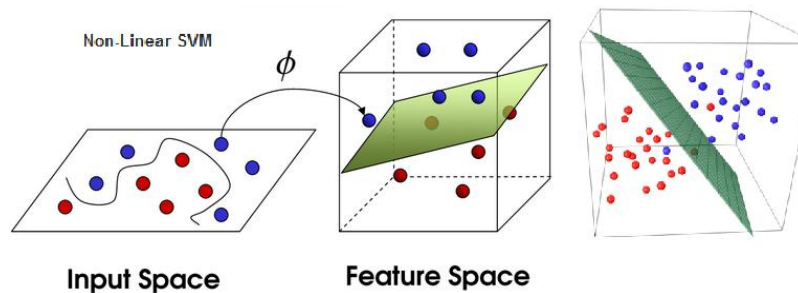


Fig. 1. High level Google architecture.



No need for Neural Nets

- Statistical machine learning
 - 1990: Robert Schapire's boosting
 - 1991: Vladimir Vapnik's SVM
 - 1993: Ross Quinlan's C4.5
 - 1995: Tin-Kam Ho's random forests



No need for Neural Nets

- Statistical machine learning (2016 survey)

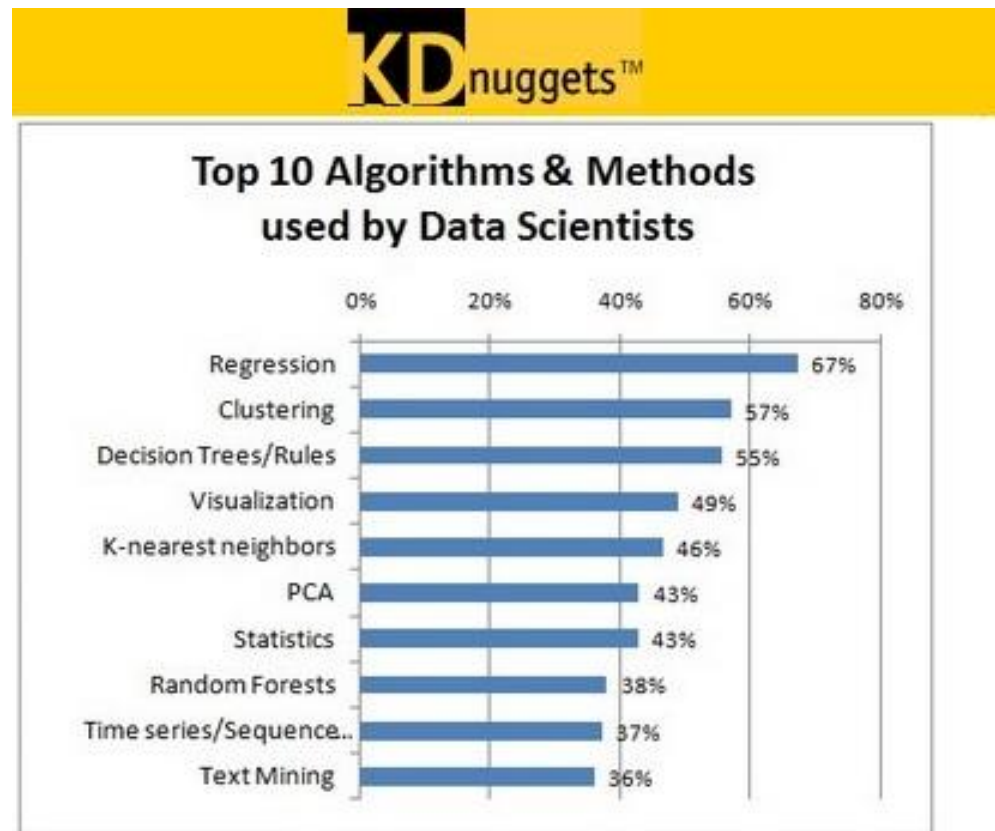


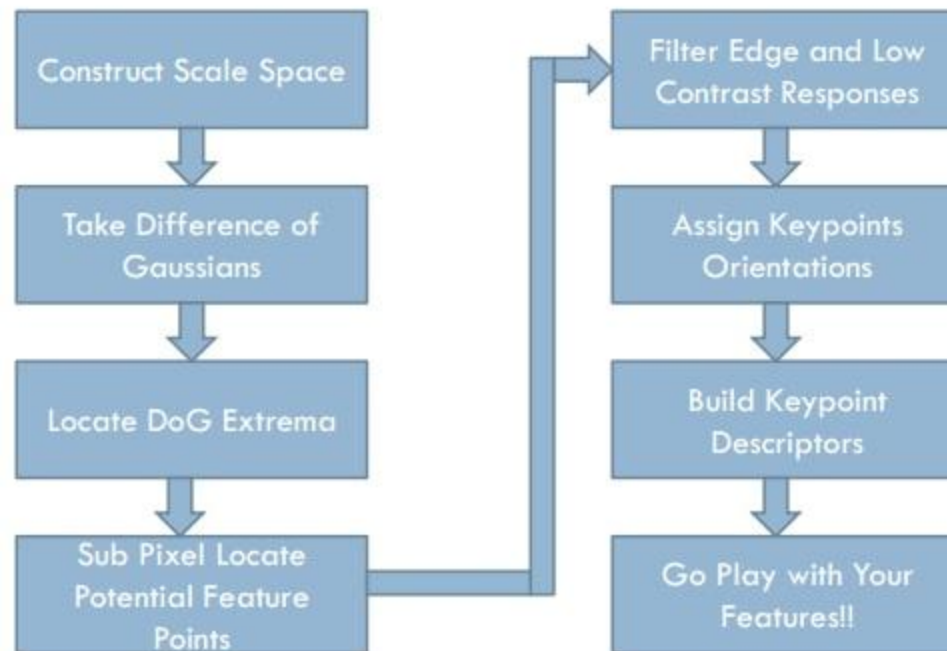
Fig. 1: Top 10 algorithms & methods used by Data Scientists

No need for Neural Nets

1999: David Lowe's SIFT for computer vision

Scale Invariant Feature Transform

David Lowe (University of British Columbia)
1999



No need for Neural Nets

ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola

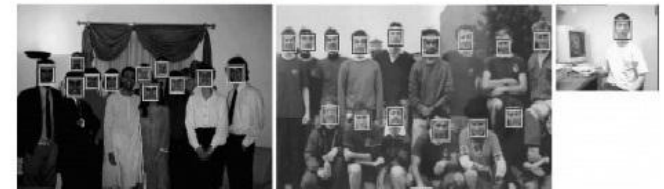


Mitsubishi Electric Research Labs



Michael Jones

Compaq CRL



- Computer Vision
 - Viola-Jones (2001)
 - HOG (2005)
 - DPM (2008)

A Discriminatively Trained, Multiscale, Deformable Part Model

Computer Vision and Pattern Recognition, 2008.

Pedro Felzenszwalb

David McAllester

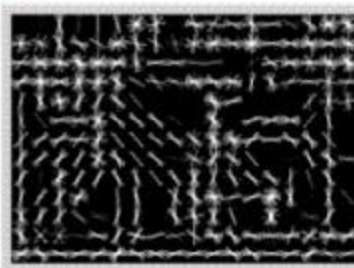


Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs

2005

INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France



No need for Neural Nets

2005: Sebastian Thrun's driverless car Stanley wins DARPA's Grand Challenge



2011: IBM's Watson debuts on a tv show



No need for Neural Nets

2011: Apple Siri (2011)



2014: Vladimir Veselov's and Eugene Demchenko's program Eugene Goostman, which simulates a 13-year-old Ukrainian boy



Unsung Heroes: The datasets

- 1990: Switchboard-1 Telephone Speech Corpus (TI)
- 1991: Continuous Speech Recognition (CSR) Corpus
- 1993: FERET (Army Research Lab)
- 1994: ORL face dataset (Olivetti)
- 1996: Broadcast News corpus
- 1999: MNIST handwritten-digit dataset (NYU)
- 2006: PASCAL VOC
- 2007: Tiny Images Dataset (MIT)
- 2007: Labeled Faces in the Wild (University of Massachusetts)
- 2009: ImageNet
- 2013: dataset of Atari games (University of Alberta)
- 2014: COCO (Microsoft)
- 2016: SQuAD (Stanford)
- 2016: MARCO (Microsoft)

Unsung Heroes: The datasets

Human Affective Datasets

Human Affective Datasets.
SEMAINE (2008)
EAGER (2013)
Cam3D (2011)
MAHNOB-HCI (2011)
RECOLA (2013)

Next...

- See <http://www.scaruffi.com/singular> for the index of this Powerpoint presentation and links to the other parts

