NOVA IMS Information Management School

AI

Artificial Intelligence

Introduction to neural networks - perspectives from engineering and philosophy

Vitor Santos

vsantos@novaims.unl.pt

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Vitor Santos brief presentation

- Assistant Professor @ NOVAIMS
- Former Invited Professor @ UM & UTAD
- Former Microsoft Portugal Academic Computer Science Program Manager
- Main areas of interest:
  - Artificial Intelligence & Information systems & Culture
  - Creativity and innovation in Information Systems
  - Information systems architectures
  - Software Engineering
  - Data Science Engineering
- Coordinated > 200 MSc
- Scientific Indexed publication > 150
- Coordinated 4 PhDs (ongoing 11)
- 44 Technological Project

vsantos @novaims.unl.pt
https://www.novaims.unl.pt/docentes-investigacao-docentes?d=22
https://www.scopus.com/authid/detail.uri?authorId=49864419600
https://scholar.google.pt/citations?user=n5PoyL0AAAAJ&hl=en

NOVA IMS Information Management School

Vitor Santos small bio

- Vitor Santos, is an Assistant Professor at NOVA Information Management School (NOVA IMS) of Universidade Nova de Lisboa teaching "Information Systems“and "Artificial Intelligence“ in Information Systems Degrees. Before that, he was an invited Professor Trás os Montes e Alto Douro University (UTAD) and Minho University (UM) teaching , “Compiliers”, Artificial Intelligence, programming languages and ”Digital Systems” courses in Computer Science and Informatics Engineering.
- He integrates several national and international conferences scientific committees and has authored several academic publications (>150).
- He was the Microsoft Portugal Academic Computer Science Program Manager for almost a decade. Before that he occupied senior management positions at Santander bank companies and has developed Computer Engineering activities for about 15 years (>40 IS projects).
- Vitor Santos holds a B.Sc. in Informatics Engineering from Cocite, a Postgraduate course in Computer Science from Science Faculty of Lisbon University, a M.Sc. in information Systems Science from University of Minho, a D.E.A. from University of Minho, a Computer Specialist title from polytechnic institutes Guarda, Castelo Branco and Viseu and a PhD. in Science and information and Technology Systems from University of Minho.
- He is working in a second PhD in Culture and Literature

Agenda

1 – Overview and brief history of AI

- Historical Overview of AI and applications
- Paradigms and approaches

2 – An introduction to Artificial Neural Networks

- Neurobiology fundamentals
- McCulloch & Pitts’ Neuron
- Learning - Hebb's Law
- Supervised Neural Networks: Perceptron / Adaline…
- Unsupervised Neural Networks: competitive learning
- Multilayered artificial neural network (Backpropagation, Neocognitron)
  - Deep Learning Introduction (CNN, GAN, …)

& Some questions for discussion will be proposed

Overview and brief history of AI

A Brief Chronology:

1927 - John McCarthy is born in Boston

1935 - Alan Turing defines computer model and shows that there are problems that cannot be solved by a computer.

1943 - McCulloch and Pitts - first artificial neuron model (1946 Von Newmann - first digital computer model)

1949 - Hebb - Hebb Learning Engine

1950 - Turing proposes machine intelligence test

1956 - John McCarthy introduces the term Artificial Intelligence at the first Darthmouth conference.

1957 - John Backus and IBM invent the FORTRAN language

1958 - John McCarthy invents the LISP.

1960 -Newell and Simon… .. PS (General Problem Solver)

1963 -Micro-Worlds - Minsky (Limited Domain Problems)

1968 - The World of Blocks

1969 - Expert Systems

resurgence of enthusiasm

1970 - Prolog Language - Edinburgh / Marseilles

Overview and brief history of AI

70/80 Some famous systems:

- SHRLDU (natural language interface applied to the block world - (SHRDLU was primarily a language parser that allowed user interaction using English terms. The user instructed SHRDLU to move various objects around in the "blocks world" containing various basic objects: blocks, cones, balls, etc)
- MYCIN (medical diagnosis)
- LUNAR (interface for geologists to interrogate about rock samples brought by Appolo on the lunar mission - the first used by people other than the system designers)
- **AI Winter**

1980 - Japan: The Fifth Generation of Computers (IA, Prolog, PLN) repercussions on global financing for AI

1997 - Deep Blue chess machine (IBM) defeats the (then) world chess champion, Garry Kasparov.

- search algorithms
- high performance computers
- Tim Berners-Lee published his Semantic Web Road map paper

Overview and brief history of AI

1999 Sony introduces an improved domestic robot similar to a Furby, the AIBO becomes one of the first artificially intelligent "pets" that is also autonomous.

Late 1990s Web crawlers and other AI-based information extraction programs become essential in widespread use of the World Wide Web.

2000 The Nomad robot explores remote regions of Antarctica looking for meteorite samples.

2004 NASA's robotic exploration rovers Spirit and Opportunity autonomously navigate the surface of Mars.

2005 Honda's ASIMO robot, an artificially intelligent humanoid robot, is able to walk as fast as a human, delivering trays to customers in a restaurant.

2005 Blue Brain is born, a project to simulate the brain at molecular detail.

2006 The Dartmouth Artificial Intelligence Conference: The Next 50 Years (AI@50) AI@50 (14–16 July 2006)

2007 DARPA launches the Urban Challenge for autonomous cars to obey traffic rules and operate in an urban environment.

2009 Google builds autonomous car.

Overview and brief history of AI

2010 Microsoft launched Kinect for Xbox 360, the first gaming device to track human body movement, using just a 3D camera and infra-red detection, enabling users to play their Xbox 360 wirelessly. The award-winning machine learning for human motion capture technology for this device was developed by the Computer Vision group at Microsoft Research, Cambridge.

2011 Mary Lou Maher and Doug Fisher organize the First AAAI Workshop on AI and Sustainability.

2011–2014 Apple's Siri (2011), Google's Google Now (2012) and Microsoft's Cortana (2014) are smartphone apps that use natural language to answer questions, make recommendations and perform actions.

2013 Robot HRP-2 built by SCHAFT Inc of Japan, a subsidiary of Google, defeats 15 teams to win DARPA's Robotics Challenge Trials. HRP-2 scored 27 out of 32 points in 8 tasks needed in disaster response. Tasks are drive a vehicle, walk over debris, climb a ladder, remove debris, walk through doors, cut through a wall, close valves and connect a hose.

2013 NEIL, the Never Ending Image Learner, is released at Carnegie Mellon University to constantly compare and analyse relationships between different images.

2015 Google DeepMind's AlphaGo defeated 3 time European Go champion 2 dan professional Fan Hui by 5 games to 0.

....

Overview and brief history of AI

2017 Asilomar Conference on Beneficial AI was held, to discuss AI ethics and how to bring about beneficial A
while avoiding the existential risk from artificial general intelligence.

2017 Poker AI Libratus individually defeated each of its 4 human opponents—among the best players in the world—at an exceptionally high aggregated winrate, over a statistically significant sample. In contrast to Chess and Go, Poker is an imperfect information game.

2017 Google DeepMind's AlphaGo (version: Master) won 60–0 rounds on two public Go websites including 3
wins against world Go champion Ke Jie.

2017 Google DeepMind revealed that AlphaGo Zero—an improved version of AlphaGo—displayed significan
performance gains while using far fewer tensor processing units (as compared to AlphaGo Lee; it used same amount of TPU's as AlphaGo Master). Unlike previous versions, which learned the game by observing millions of human moves, AlphaGo Zero learned by playing only against itself.

2020 Alibaba language processing AI outscores top humans at a Stanford University reading and comprehension test, scoring 82.44 against 82.304 on a set of 100,000 questions.

2020 The European Lab for Learning and Intelligent Systems (aka Ellis) proposed as a pan-European competitor to American AI efforts, with the aim of staving off a brain drain of talent, along the lines of CERN after World War II.

2020 Announcement of Google Duplex, a service to allow an AI assistant to book appointments over the phone. The LA Times judges the AI's voice to be a "nearly flawless" imitation of human-sounding speech.

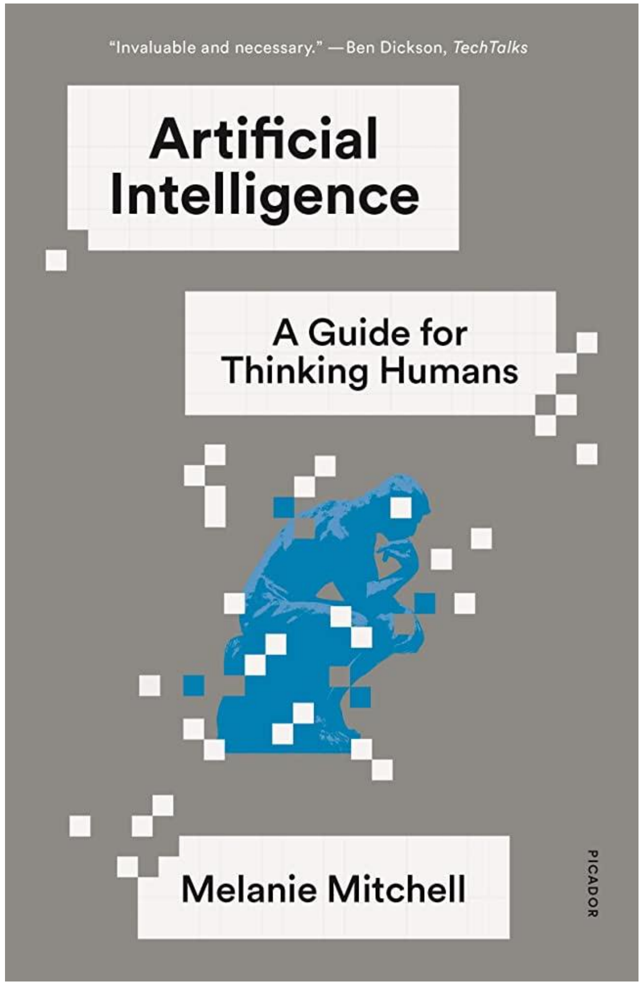
2022 ChatGPT (Chat Generative Pre-trained Transformer - chatbot developed by OpenAI built on top of OpenAI's GPT-3 family of large language models and has been fine-tuned (an approach to transfer learning) using both supervised and reinforcement learning techniques.

https://podcasts.apple.com/us/podcast/hype-vs-realidade-winter-is-coming/id1510665343

Overview and brief history of AI

Sugestion: If you want to know more about AI history →

Melanie Mitchell book - a leading computer scientist, reveals AI's turbulent history and the recent spate of apparent successes, grand hopes, and emerging fears surrounding it

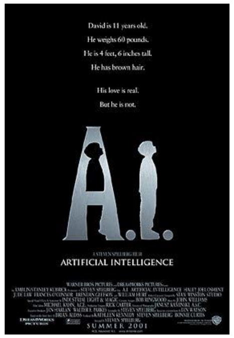

"Invaluable and necessary." —Ben Dickson, TechTalks
Artificial Intelligence
A Guide for Thinking Humans
Melanie Mitchell
PICADOR

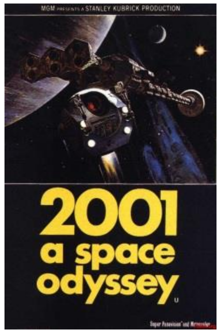
Overview and brief history of AI

- AI is perhaps one of the areas of informatics that arouses the most curiosity and at the same time seems to be one of the most promising.

  Movies such as 2001 Space Odyssey, Solaris, Blade Runner, and Star Wars, Artificial Intelligence, Avatar, Transcendent contribute to the idea of the possible "Machine Intelligence" being widely disseminated.

  However, for the vast majority of people, AI is embroiled in a profound mystery of which only a few Scientists know the intricacies...





Overview and brief history of AI

The magic and mysticism surrounding the knowledge of Artificial intelligence lead us to questions such as:

- What is the object and what are the scientific objectives of Artificial Intelligence?
- What are the paths followed for the definition of AI!
- What is the current state of ”Art”?
- What progress till today has been made that deserves relevant attention?
- What tools does Artificial Intelligence use to carry out its tasks?

https://podcasts.apple.com/us/podcast/vamos-definir-o-que-é-realmente-ai/id1510665343

Vitor Santos

Overview and brief history of AI

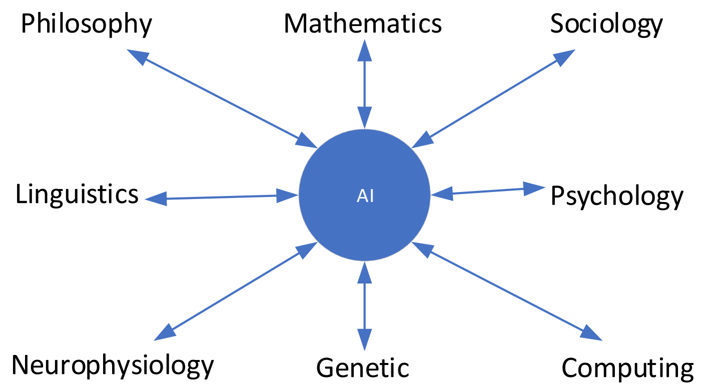
AI Objective: Smart Entities

Objectives: Understand how Smart Entities work

Build smart entities

AI Dependencies:

- Philosophy: Theories about knowledge, mind,….
- Mathematics: Logic, Odds, Decision,…
- Psychology: Perception, Memory,…
- Linguistics: Syntax, Semantics,…
- Computing: Tool.



Paradigms and approaches

Pathways to AI

Symbolic or Computational Approach (Classical School)
Associated subjects: Psychology, Epistomology, Sociology
Nuclear notions: representation, reasoning, search

Connectionist Approach
Associated subjects: Statistics, Neurophysiology
Nuclear notions: learning, pattern recognition

Evolutionary Approach
Associated subjects: Molecular Biology, Theory of Dynamic Systems, Darwinism
Nuclear notions: reproduction, evolution

Biological/ethological Approach
Associated subjects: Biology, Ethology, Control Theory
Nuclear notions: situation, incorporation

Agenda

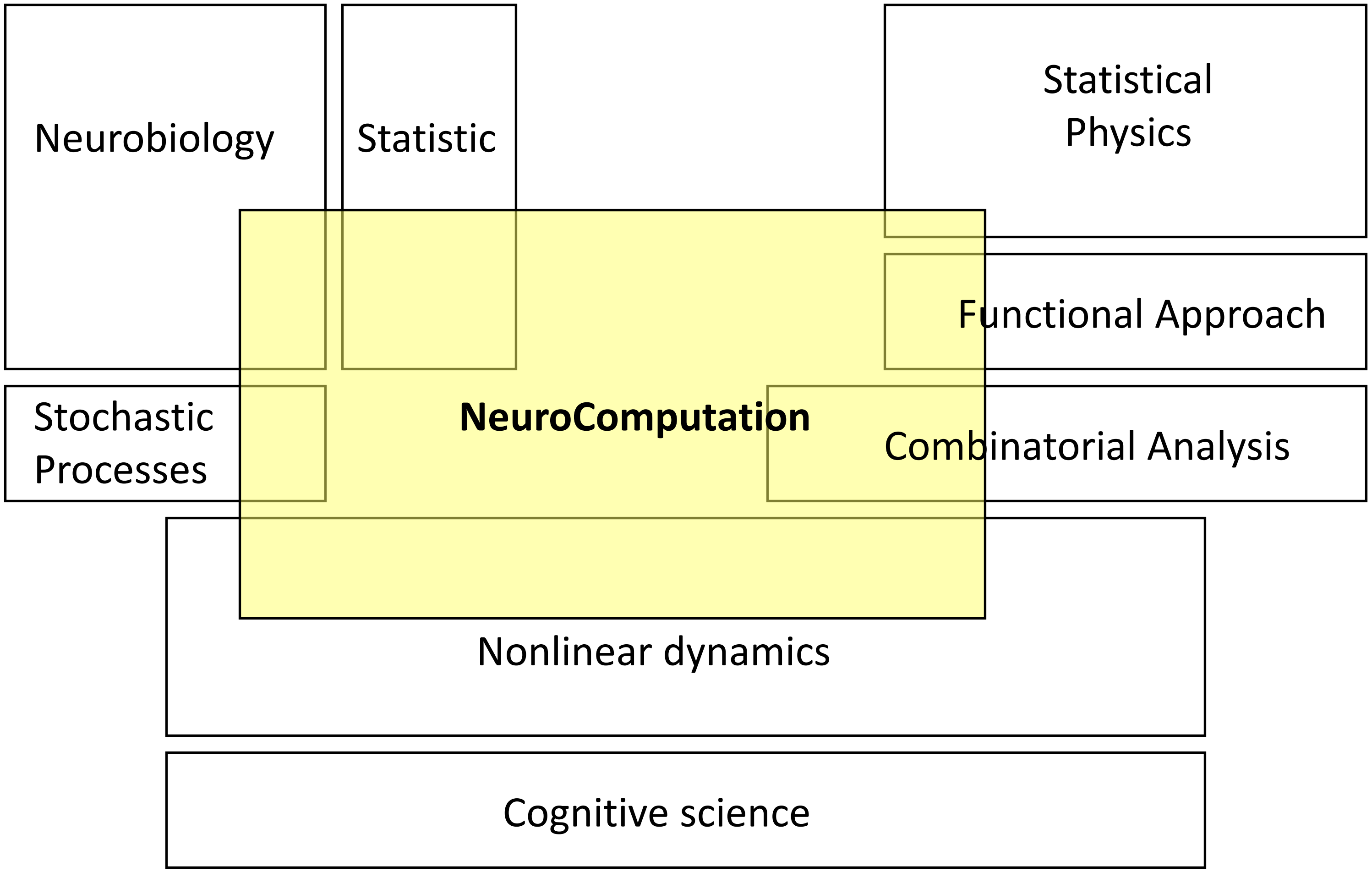
1 – Overview and brief history of AI

- Historical Overview of AI and applications
- Paradigms and approaches

2 – An introduction to Artificial Neural Networks

- Neurobiology fundamentals
- McCulloch & Pitts’ Neuron
- Learning - Hebb's Law
- Supervised Neural Networks: Perceptron / Adaline…
- Unsupervised Neural Networks: competitive learning
- Multilayered artificial neural networks
    - Backpropagation
    - Deep Learning

Artificial Neural Networks



Biological Neural Networks

- Facts :
  - The basic structure of the nervous system, common to most vertebrates, is the same as that of all mammals;
  - 98.8% of human DNA matches a chimpanzee's DNA. 1.2% difference corresponds to structural genes
  - Each human cell contains roughly three billion base pairs, or bits of information. Just 1.2 percent of that equals about 35 million differences. Some of these have a big impact, others don't. And even two identical stretches of DNA can work differently--they can be "turned on" in different amounts, in different places or at different times.

Our brain contains no new hardware (almost → Neurogenesis-)… only rearrangements, dilations, contractions and modifications of the same basic mammalian structure, with a peculiar tuning.

NOVA IMS Information Management School

Question - Superintelligence is possible?

University of Oxford philosopher Nick Bostrom defines superintelligence as

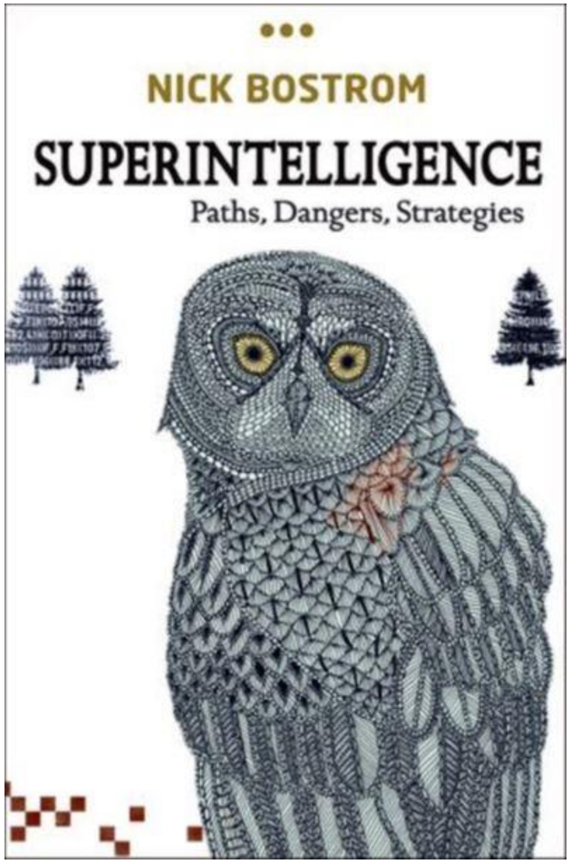
"any intellect that greatly exceeds the cognitive performance of humans in virtually all domains of interest“

Pathways

1. Biological Cognition

   98.8% of X DNA matches a human´s DNA

2. Whole Brain Emulation
3. Artificial Intelligence



Biological Neural Networks

- Facts
    - It is often read and heard that if it were possible to build a computational device in the brain's image, it would become a computer capable of solving all our problems.

This idea is

Biological Neural Networks

- Facts
  - We should not be surprised that brains (or networks of formal neurons) cannot solve all problems, but only find reasonable, approximate solutions to limited but important classes of problems.
  - The networks of formal neurons suggest us many useful ideas, which enchant us and that, in the long run, become more fascinating than magic ...

Biological Neural Networks

- Facts
  - It is **through synapses** that one cell can influence the activity of others.
  - The theory of artificial neural networks tells us that **synapses vary in efficiency and that this is the key to understanding the nature of the computation they perform.**

Biological Neural Networks

◈Hardware

An average adult male brain weighs about 1375 grams (~ 3 pounds)

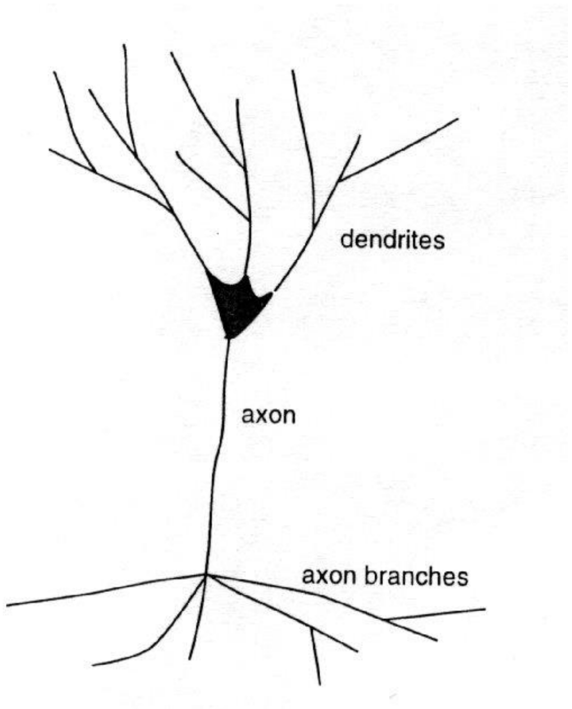
The brain isn't fully formed until age 25. Brain development begins from the back of the brain and works its way to the front. Therefore, the frontal lobes, which control planning and reasoning, are the last to strengthen and structure connections.

Biological Neural Networks

- Hardware
  - The elemental computing units of the nervous system are the **neurons**, or nerve cells.
  - The human brain has between **86 billion** (86 x 10^9) and 10^{11} (one hundred billion) neurons;
  - Each neuron is connected to hundreds or thousands of other neurons.
  - It is thought that the number of connections is between 10^{13} and 10^{15} (1 quadrillion (1,000 trillion) connections)
  - Neurons cooperate and compete with each other.

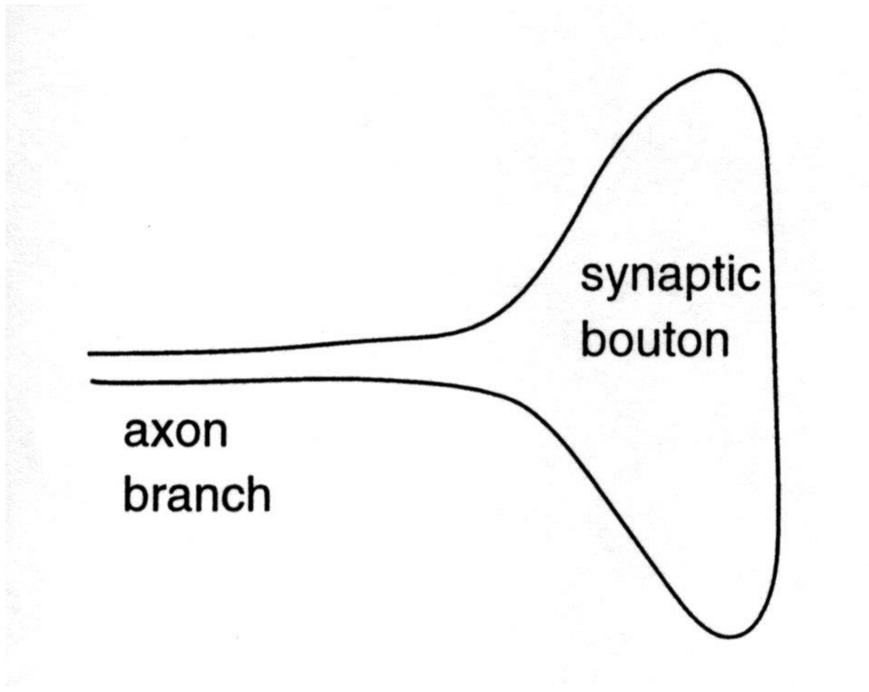
Biological Neural Networks

- Hardware
  - The generic neuron is a model of the spinal cord motor neurons, the most studied in mammals.



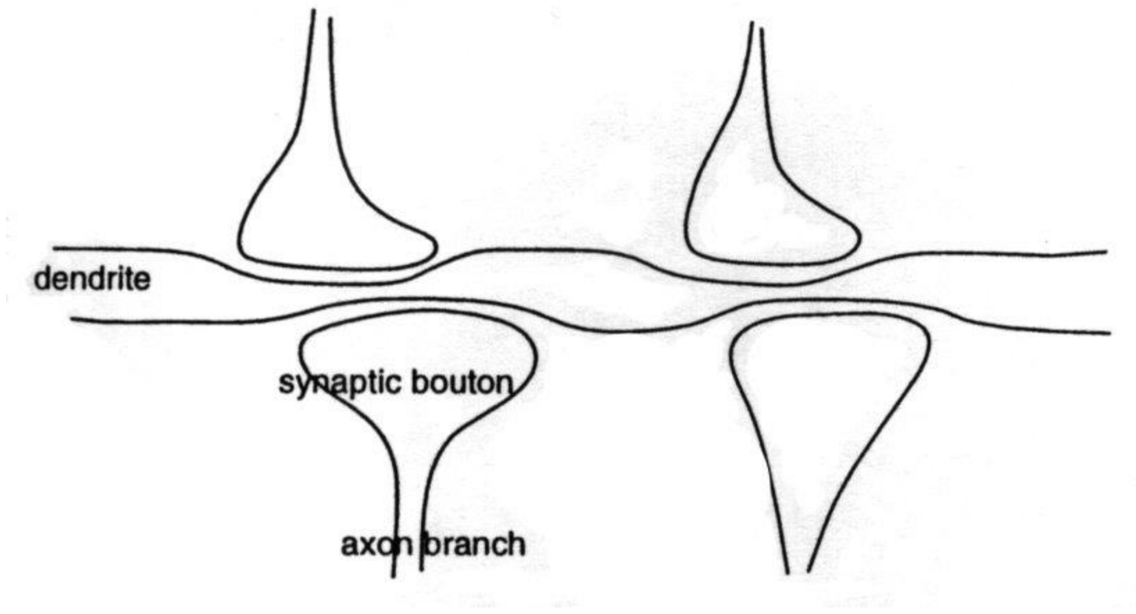
Biological Neural Networks

- Hardware
  - At the extremities of the axon branches are very specialized structures, called synapses.



Biological Neural Networks

- Hardware
  - The dendrites of a neuron are surrounded by the synapses of other neurons.

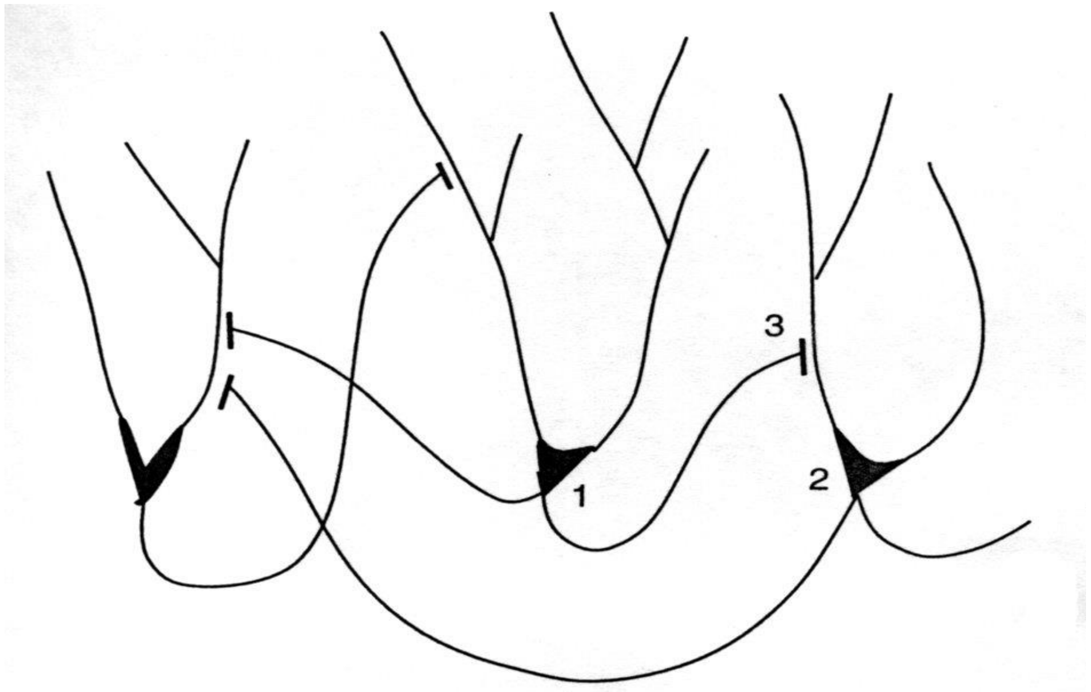


Biological Neural Networks

- Hardware
  - Each neuron is connected to hundreds or thousands of other neurons;
  - The number of connections is estimated between **10^{13} and 10^{15}**;
  - It should be noted that the values presented for neuronal connectionism is much lower than the number that would be obtained if the connectionism in the brain were total.;
  - The human brain can generate about 23 watts of power (enough to power a lightbulb).

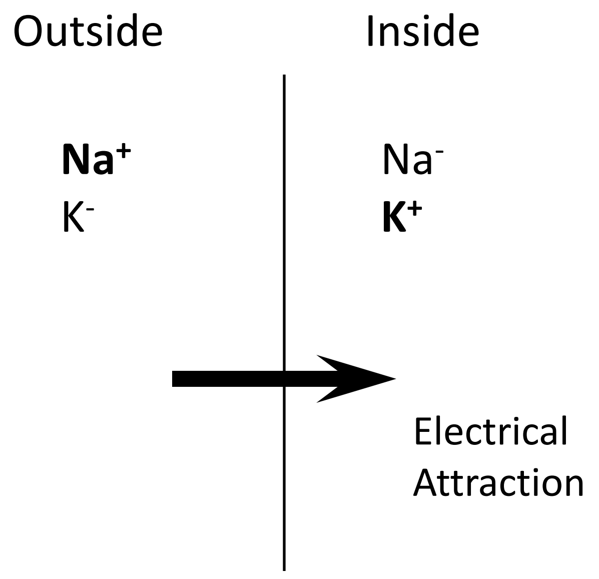
Biological Neural Networks

- Hardware
  - In terms of brain size, each neuron is linked to only an infinitesimal fraction of other neurons.



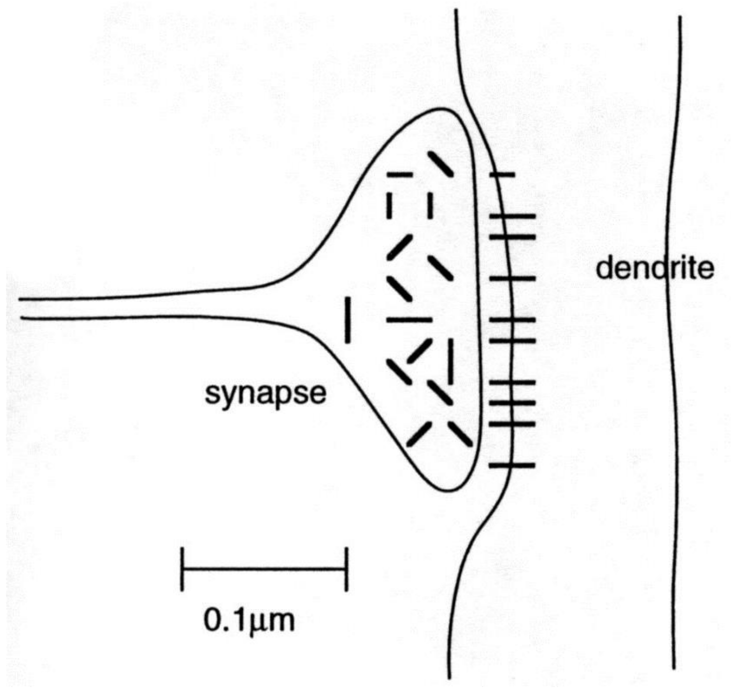
Biological Neural Networks

- Hardware
  - Inside the neuron we find a fluid that contains essentially Na^-, K^+ and Cl^- ions. On the outside, the cerebral fluid contains essentially ions with the opposite sign.
  - Thus, we have:



Biological Neural Networks

- Hardware
  - In the dendritic wall there are large molecules called receptors; in synapses there are large molecules called transmitters.

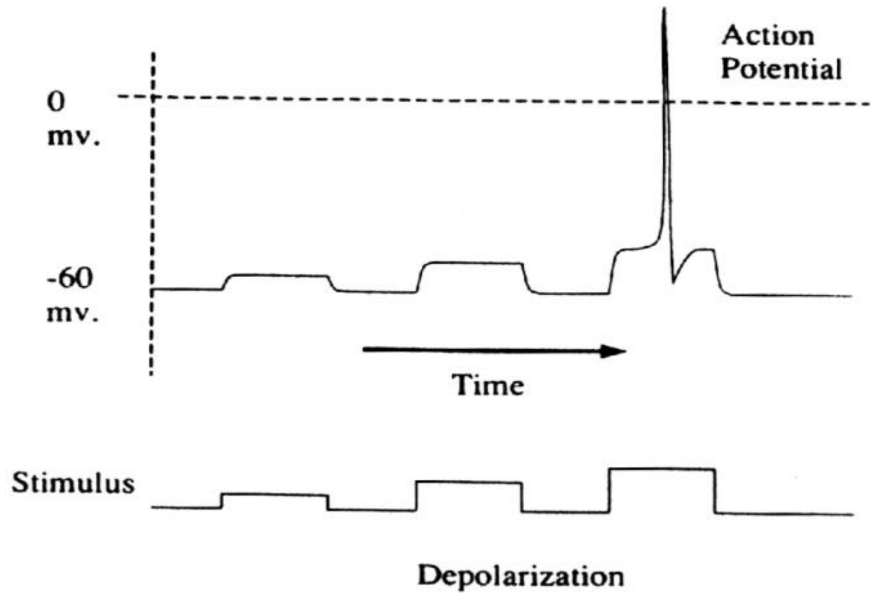


- Transmitters may pass through the synaptic wall as it acts as a membrane;
- Once the internal and external ionic charges in relation to the neuron are known, it is seen that the ions flow from the synapse to the dendrite;
- These ions carry electric charges, causing the change in the ion concentration inside the neurons, causing voltage spikes

NOVA IMS Information Management School

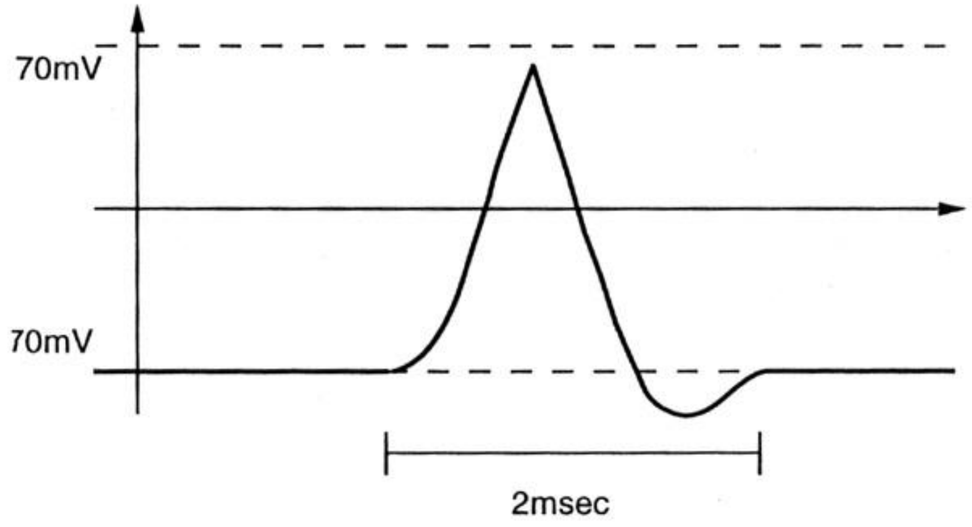
Biological Neural Networks

- Operation
  - Along the axons propagates an action potential, as shown in the figure below :



Biological Neural Networks

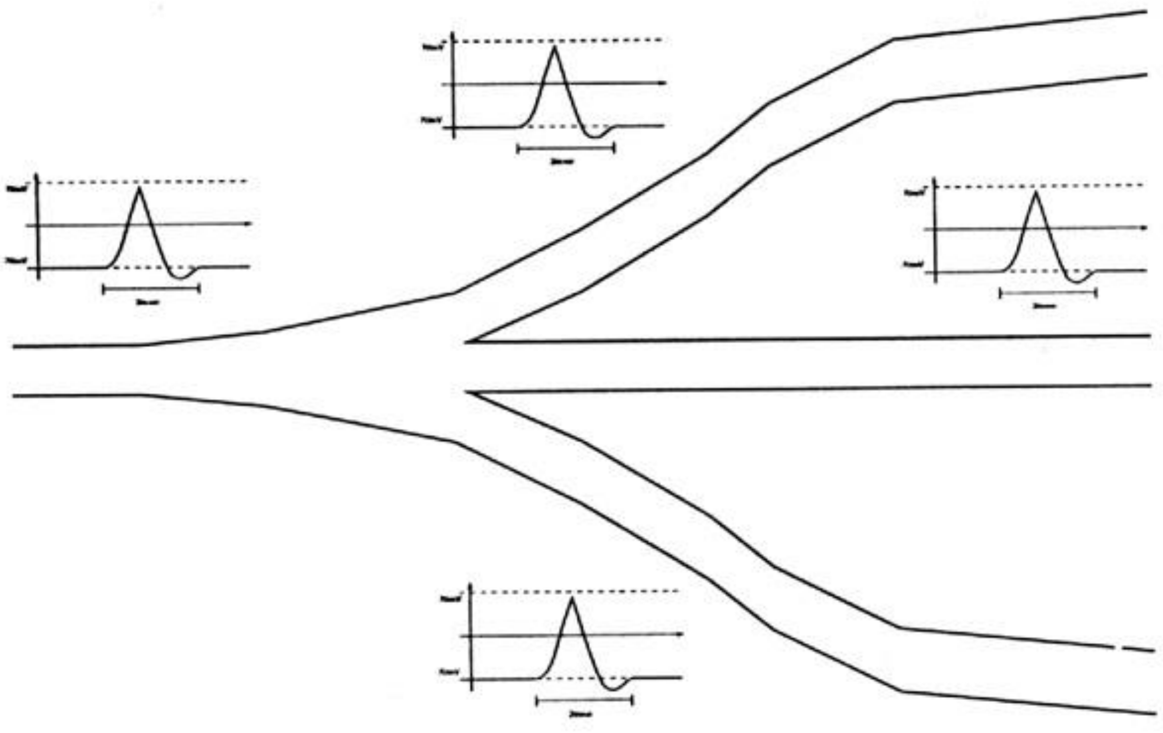
- Operation
  - The action potential is a wave that propagates as shown in the following figure:



Biological Neural Networks

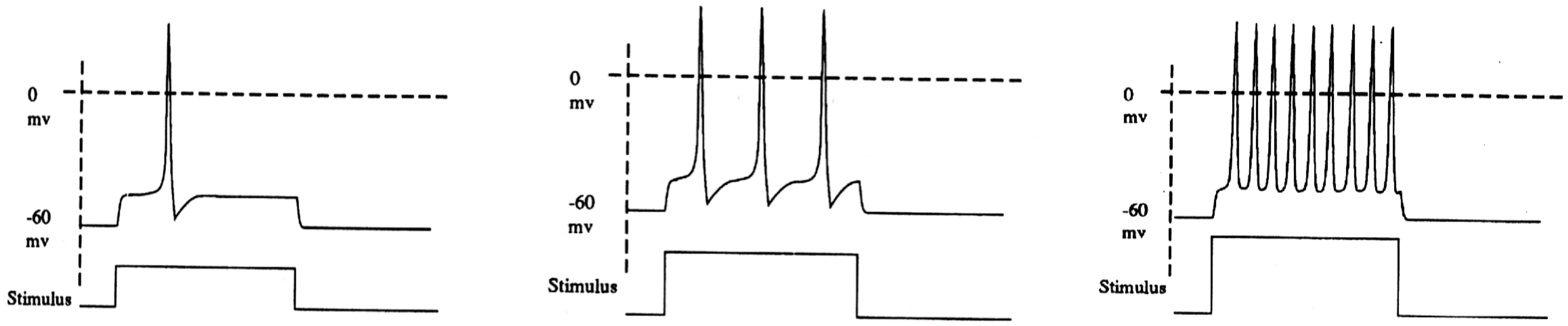
- Operation
  - The current peak is not always present and, if there is no signal, the axon will be at a point called the resting potential of about **-70mV**.
  - During the propagation of the action potential, in axon splitting situations, we witness splits of the signal, while maintaining the amplitude of the signal (remember that the axon does not induce the resistance of an electric cable !!). The following figure illustrates this situation.

Biological Neural Networks



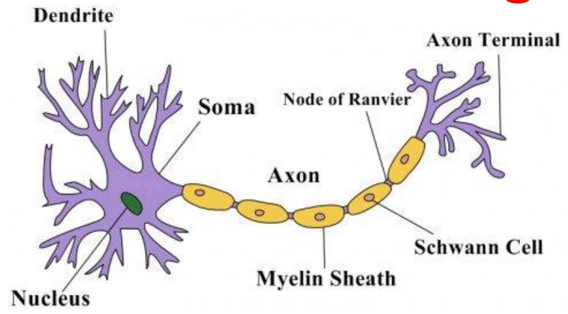
Biological Neural Networks

- Operation
  - If we increase the intensity of the stimulus, we do not get a greater action potential. What will happen is that a regular series of action potentials is triggered whose frequency is a function of the stimulus intensity, as shown in the following images:



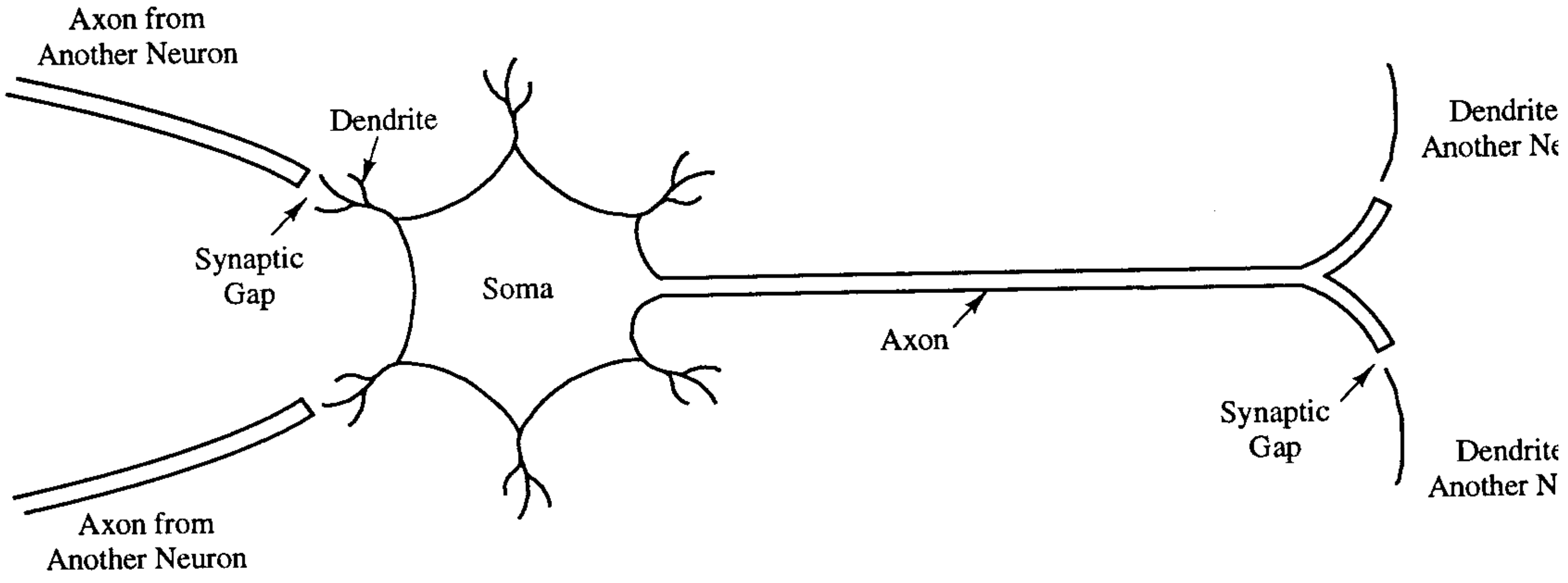
Biological Neural Networks

- Operation
  - Dendrites receive electrical currents from other cells and the nucleus (inside soma) and process and integrate the currents. The resulted computation is propagated along the axon to the synapses.
  - Synapse output currents are input currents from other neurons.
  - Synapses are responsible for the possibility of one neuron influencing another.
  - **The efficiency of the synapses can vary and these efficiencies are the key to the understanding of neuronal computation.**



Biological Neural Networks

- Operation



Biological neuron

Biological Neural Networks

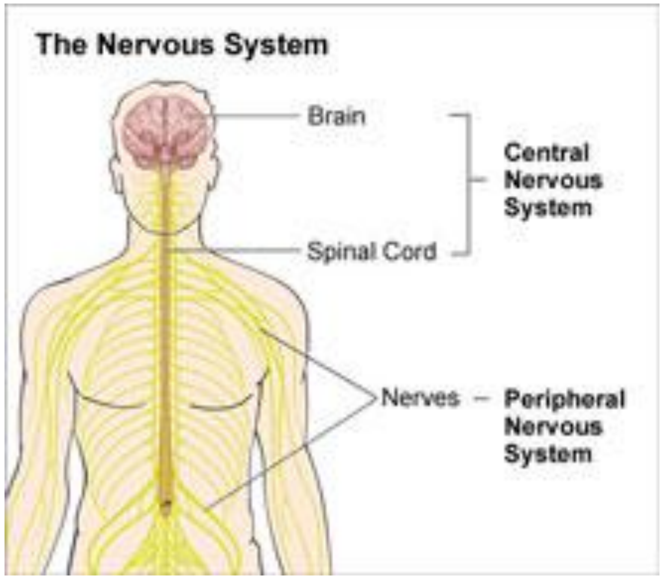
- Operation
  - Depending on the neurotransmitters, these potentials, called postsynaptic potentials, may be excitatory or inhibitory.
  - The excitatory postsynaptic potential is the result of opening an ion channel between the synapse and the dendrite that is permeable to Na + and K + ions.
  - The inhibitory postsynaptic potential acts by preventing the potential of the axon membrane to reach the excitatory plateau. This means that inhibitory neurotransmitters activate Cl- and K + channels.
  - The number of postsynaptic potentials can be as large as the number of dendrites (average 10,000), and they are diffused, or "travel," through the dendrites toward the soma.

Biological Neural Networks

- Operation
  - If there are more excitatory than inhibitory postsynaptic potentials, then the soma may emit an action potential and the whole process begins again.
  - As we have already noted, the emission of action potential occurs only if a certain threshold is reached.
  - After emitting an action potential, the neuron will have a period of 1 to 2 ms, called the absolute refractory period in which it cannot emit new action potential. This implies that the maximum firing frequency of the neuron is between 500Hz and 1000Hz.
  - Experiments have even shown that certain neurons have firing frequencies of 50 Hz.

Biological Neural Networks

- Operation
  - The nervous system makes up only about 1% to 2% of the total weight of the human body. However, electrochemistry of neurons requires high metabolism, and neurons are responsible for consuming about 20% of our body's total energy.



Biological Neural Networks

- Operation
  - Until recently, most neuroscientists (scientists who study the brain) thought we were born with all the neurons we were ever going to have. When a neuron dies, it is not replaced with another neuron…. Even before birth a large portion of nerve cells die (half of the cells may even die).
  - This is supposed to happen because neurons compete with each other to make connections. When the connections are not correct then these cells are "**doomed**" to death.
  - "**Condemnation**" is thought to be transmitted from neighbouring cells that encode death messages.
  - **Neurogenesis-** scientists, like Elizabeth Gould, later found evidence of newborn neurons in a distinct area of the brain in monkeys, and Fred Gage and Peter Eriksson showed that the adult human brain produced new neurons in a similar area
  - **Neurogenesis** in the adult human brain is still tricky for neuroscientists to show, let alone learn about, how it impacts the brain and its functions. Still, scientists are intrigued by current research on neurogenesis and the possible role of new neurons in the adult brain for learning and memory.

Biological Neural Networks

- Operation
  - Nature is in charge of keeping only the necessary neurons since they are metabolically very active and therefore consume a lot of energy.
  - The high metabolism of neurons is due to the fact that the neuron permanently has a pump (sodium pump) expelling sodium and concentrating potassium.
  - By studying the electrical behaviour of the neuron it was found that a positive potential is generated in response to the injection of certain critical depolarizing current values. This potential (action potential), once generated, does not change its shape with increasing current.
  - This fact is almost analogous to the true and false of Boolean logic !!!

Agenda

1 – Overview and brief history of AI

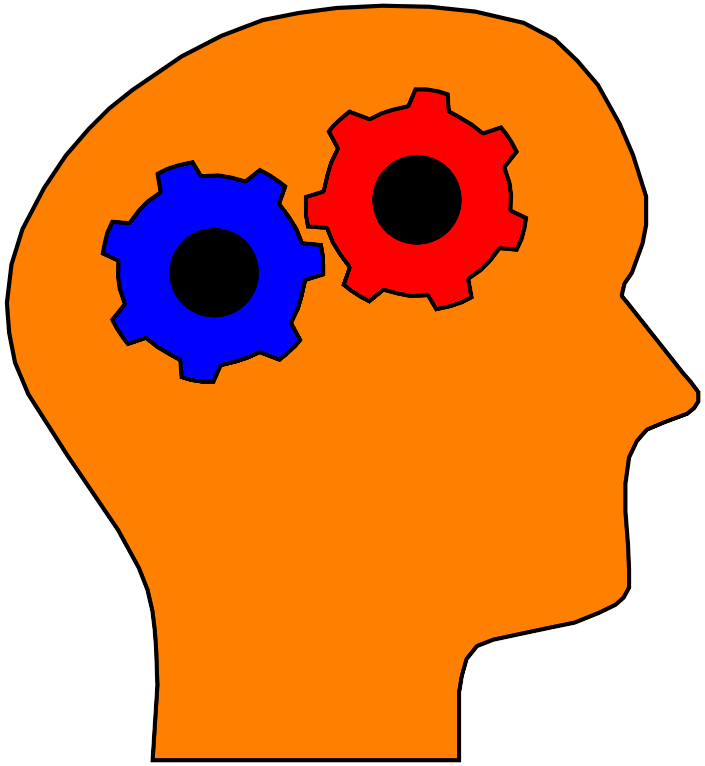
- Historical Overview of AI and applications
- Paradigms and approaches

2 – An introduction to Artificial Neural Networks

- Neurobiology fundamentals
- McCulloch & Pitts' Neuron
- Learning - Hebb's Law
- Supervised Neural Networks: Perceptron / Adaline…
- Unsupervised Neural Networks: competitive learning
- Multilayered artificial neural networks
    - Backpropagation
    - Deep Learning

Neural Networks

- NeuroComputation Model



Artificial Neural Networks

- From everything stated above, it is easy to conclude that if we tried to include in the model everything we know, then neurocomputation would not be possible to implement based on the technology we currently have.

- So let's ignore the details........

Artificial Neural Networks

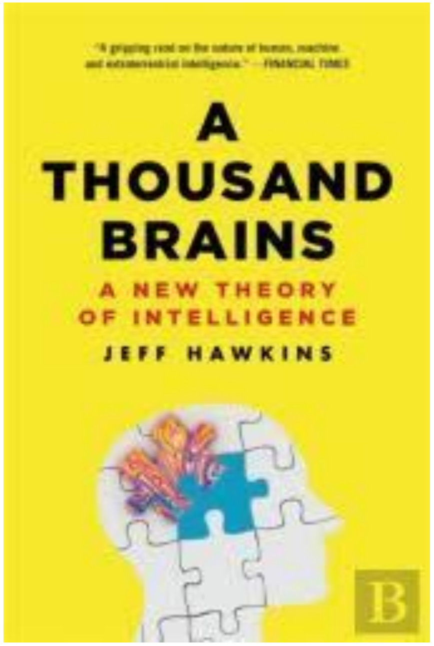
Artificial Neural Networks have been developed as generalizations of mathematical models of human cognition or neurobiology, based on the following assumptions:

- **Information processing occurs in simple units, called neurons**
- **Signals are passed between neurons via connections**
- **Each connection is associated with a weight which in a typical Neuronal network multiplies the transmitted signal.**
- **Each neuron uses an activation function, usually non-linear, that allows to determine its output signal, as a function of the inputs (sum of the weights of the input signals)**

Question – Will he have true AI without first understanding the brain?

Most people in AI don't care too much about the details, says Jeff Hawkins

- Jeff Hawkins and his team discovered that the brain uses maplike structures to build a model of the world-not just one model, but tens of thousands of models of everything we know (new theory)
- This discovery allows Hawkins to find answers to important questions about how we perceive the world, why we have a sense of self, and the origin of high-level thought.



A little of history...

- 1940s - The Beginning
  - 1943 McCulloch and Pitts - first artificial neuron model (1946 Von Newmann - first digital computer model)
  - 1949 Hebb - Hebb Learning Engine
- 50's and 60's - Successful 1st Age
  - 1958 Rosemblat - new model of neuron Perceptron (with one layer)
  - 1960 Bernard Widrow and Marcian Hoff (Stanford) - Introduce Adaline with Least Squares Learning Algorithm
  - 1969 - Minsky and Pappert - Book with Perception Limitations
- 70's - Slightly Busy Period "AI Winter of the 70s"
  - 1972 - Kohonen and Anderson - Associative Linear Networks
  - 1976 - Grossberg - ART (Adaptative Ressonancy Theory)

NOVA IMS Information Management School

A little of history...

- 80's - Renewed Enthusiasm

  1982 - Kohonen - SOM, Self-organized map,

  1982 - Hopfield - Hopfield Network (recurring)

  1986 Rumelhart and McClealland Multilayer Perception

  Rumelhart, Hinton and Williams - Error Re-propagation in Recurrent Multi-Level Networks

  Boltzman Machines

- 90's
  - Vapnik and Others - Vector Support Machines to Solve Regression and Pattern Recognition Problems

NOVA IMS Information Management School

A little of history…

- 2000s - Widespread Application
  - Economy and Finance
  - Signal Processing
  - Computer vision
  - Robotics
  - Automation of expert systems
  - Statistic
  - Data mining
  - …
- 2000s - Deep Learning - The term was introduced in Artificial Neural Networks by Igor Aizenberg and colleagues in 2000.
- …

Artificial Neural Networks

McCulloch-Pitts Formal Neuron – 1943

It is perhaps the first artificial neuron - proposed by Warren McCulloch and Walter Pitts

is considered the first functioning archetype of the nervous system based on abstract neurons and their interconnections.

Its main features can be summarized:

- Has binary activation (does not fire at 0 or fire at 1);
- Neurons are connected by links with their own weights;
- A connection is exciting if the weight of the connection is positive; It is inhibitory if the binding weight is negative. All the weights of the excitatory bonds to a neuron have the same value;
- Each neuron has its own bent. If the input of the network in this neuron is greater than the inclination, the neuron fires.

Architecture

- Two-state machine, characterized by an excitability threshold and driven by synapses of equal efficiency and linear interaction.

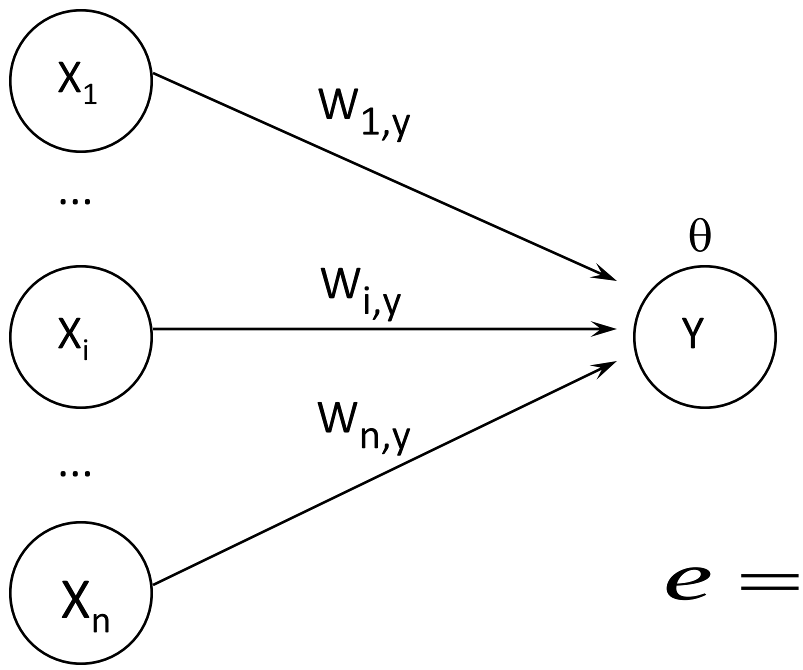
- Inhibitory synapses have an absolute action, ie when an inhibitory synapse is active, the neuron is inactive.

Operation

Over a certain period of time, the neuron responds to the activity of its synapses, which reflect the presynaptic cell states, as follows:

- if there are no active inhibitory synapses, then the neuron compares the sum of the stimuli with the threshold value;
- if there is excess stimulation, then the neuron is active, otherwise it is inactive.

Artificial Neural Networks



$$e = \sum_{i=1}^{n} Xi \; W_{i,y}$$

if $e >= \theta$ then $y_{out} = 1$, else $y_{out} = 0$

NOVA IMS Information Management School

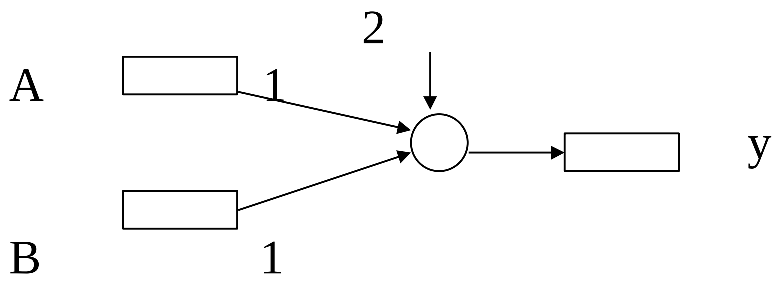
Artificial Neural Networks

- Applications

AND

A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

A.B



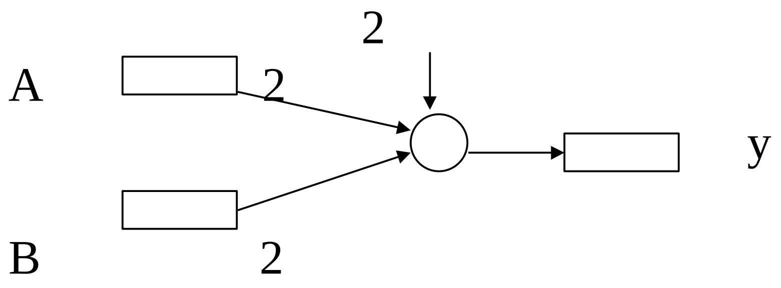
Artificial Neural Networks

- Applications

OR

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

A+B



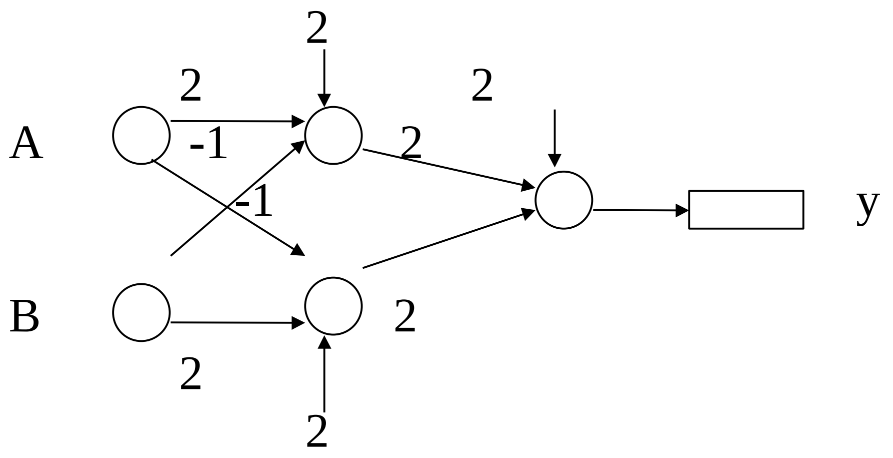
Artificial Neural Networks

- Applications

XOR

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

A⊕B



A⊕B = A . ~ B + ~A. B

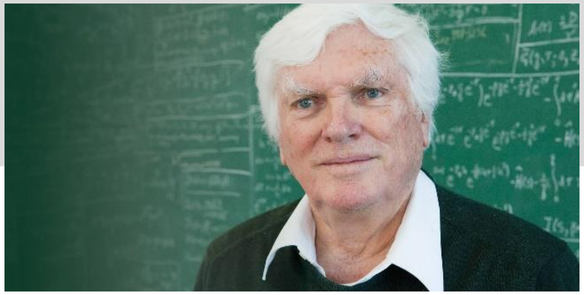
Artificial Neural Networks

- The formal neuron thus presented is an all-or-nothing type.

- Given the level of neurophysiological knowledge in 1943, when the basis of electrical and ionic exchanges of neuronal activity was not yet clear, this type of approximation was very acceptable ☺.

- But our present knowledge allows us to state that neurons are not limited to making the propositions of formal logic.
- Moreover, we are now certain that neurons at all do not behave as mere 2-state machines…

Artificial Neural Networks



- In 1972, Bruce Knight proposes a different model of McCulloch-Pitts formal neuron, and calls it a simple integrative model, although it is also known as an **integrate-and-fire model**.

- **Architecture** :

  imagine a neuron containing an internal variable u (t) that corresponds to the membrane potential. A stimulus, s (t), may correspond to ionic currents and thus:

$$\frac{du}{dt} = s(t) \qquad \text{with } s(t)>0$$

Artificial Neural Networks

- **Architecture**

When **u** reaches a trigger value, θ, a nervous impulse is triggered. The system then resets the initial value of **u**.

Suppose a stimulus is triggered at time $\mathbf{t_1}$ and the current instant is **t**.

So :

$$u = \int_{t_1}^{t} s(t)dt$$

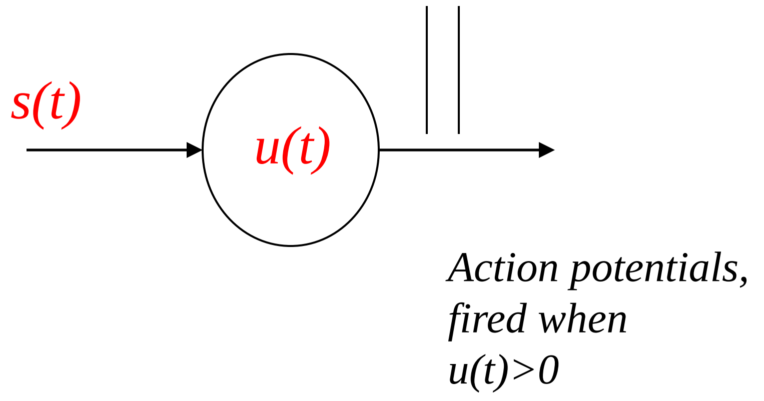
Artificial Neural Networks

- **Architecture**

If the next pulse is fired at time **t₂**, there is a strong relationship between the action potential threshold value, **θ,** and **u** at time **t₂**:

$$\theta = \int_{t_1}^{t_2} s(t)dt$$

The neuron model can be represented as follows:

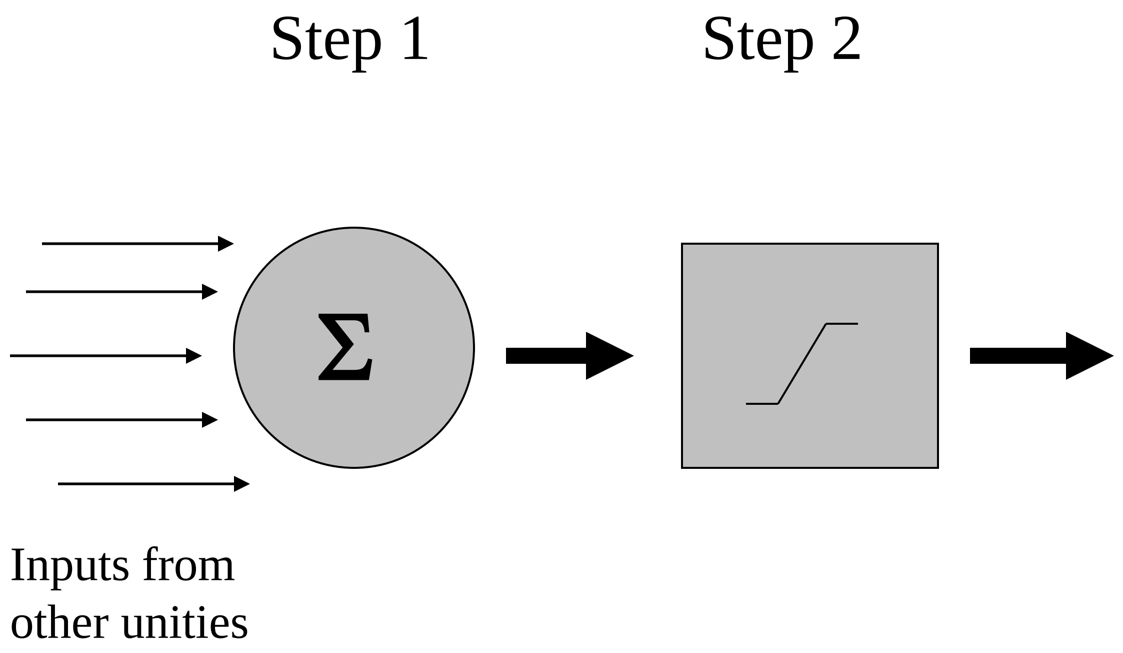


The Generic Neuron

- The most used model for the construction of Neural Networks is the Integrating Neuron, performing detailed computation, based on the efficiency of its connections. This model also contains an important nonlinearity.

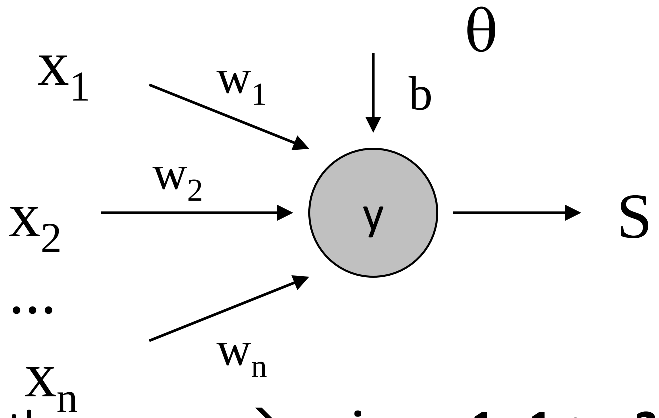
- The operation of this model is a two-step process: in the first, the input of the synapses are added, resulting in a certain level of activity; in the second step this value will be used to generate the output activity of the model, using the result as input of a non-linear function, and thus linking the activity level (membrane potential) to the output value.

Artificial Neural Networks



Artificial Neural Networks

- We can then represent the neuron as follows:



Input of the neuron → **y_in = w1 x1 + w2 x2 + … + wn xn + b**

θ → threshold

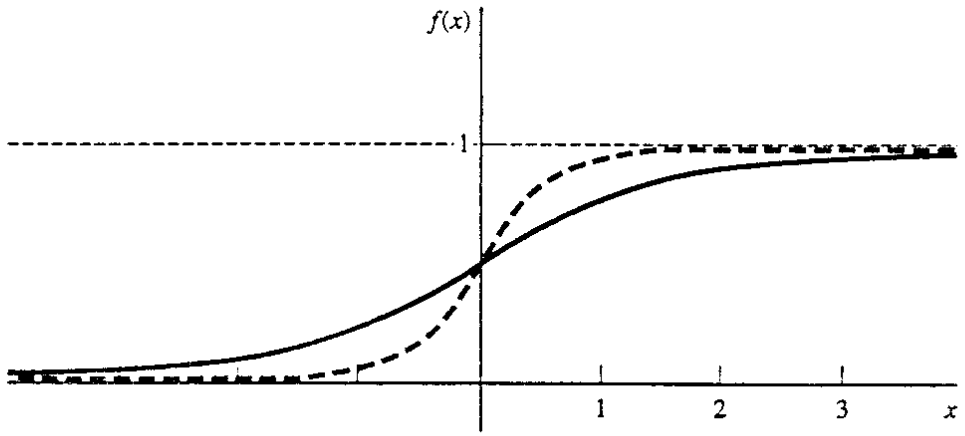
The inputs can be binary (0 and 1) or bipolar (-1 and 1)

Each input has a weight assigned(w_i) that will affect the input signal. As in biological mechanisms there are stronger patterns than others and so their effect must be reflected in the network. The weights are alterable and can be adjusted over the life of the network.

Artificial Neural Networks

- Each neuron has an activation function, which allows determining its output signal (S), depending on the inputs.
- A widely used activation function is the sigmoid function :

$$f = \frac{1}{1+\exp(-\sigma x)}$$



Binary sigmoid. Steepness parameters $\sigma = 1$ and $\sigma = 3$.

NOVA IMS Information Management School

Agenda

1 – Overview and brief history of AI
- Historical Overview of AI and applications
- Paradigms and approaches

2 – An introduction to Artificial Neural Networks
- Neurobiology fundamentals
- McCulloch & Pitts' Neuron
- Learning - Hebb's Law
- Supervised Neural Networks: Perceptron / Adaline…
- Unsupervised Neural Networks: competitive learning
- Multilayered artificial neural networks
  - Backpropagation
  - Deep Learning

Artificial Neural Networks

- **How do Artificial Neural Networks learn?**
  - Supervised - a “teacher” enters the data to learn
  - Unsupervised - The network seeks to detect patterns in the input signals and according to its structure, learns.

A network is taught by presenting cases, giving each one the inputs and the corresponding outputs.

Throughout learning the weights are changed depending on the cases being presented to the network. The rules that determine these weight changes are called learning laws.

One of the first rules to appear was **Hebb's Law**

A system learns an association $\vec{f}: \vec{g}$ when given $\vec{f}$ we obtain $\vec{g}$.

Question– Is Machine Learning <=> Artificial Neural Networks?

Machine Learning Paradigms

- Inductive Learning:
  - Acquisition of concept descriptions from a set of known examples and counterexamples of this concept.
- Analogical Learning:
  - Fundamentally deductive learning from a domain theory and a small number (usually one) of application examples of the theory. Using past problem-solving experience to guide new problem solving or increase the efficiency of pre-existing knowledge application.
- Neural Networks:
  - Learning consisting of adjusting the bond weights of a network of single processing units ("neurons").
- Immunological Networks:
  - Inspired by an analogy with the immune system and its meta-dynamics
- Genetic Algorithms:
  - Inspired by an analogy with mutations in biological reproduction and natural selection.
- …

Artificial Neural Networks

- **Generalized Hebb’s Law**

  (Donald Hebb - 1949)

A system learns by modifying connections A[i,j] by the rule:

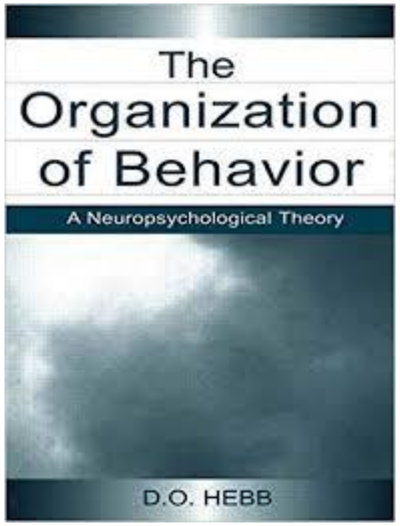
$$\vec{A[i,j]} = \alpha\, \vec{f[j]}\ \vec{g[i]}$$

A - Learning Matrix

f - Vector stimulus

g - Vector answer

α - Learning Constant



Hebb's law

- Consider a network with 4 input neurons and as many output neurons; represent the stimuli by **f** and the responses by **g**. The matrix of associations is called **A**.
  - According to Hebb's law:

$$\vec{A} = \vec{g} \cdot \vec{f}^{\,T}$$

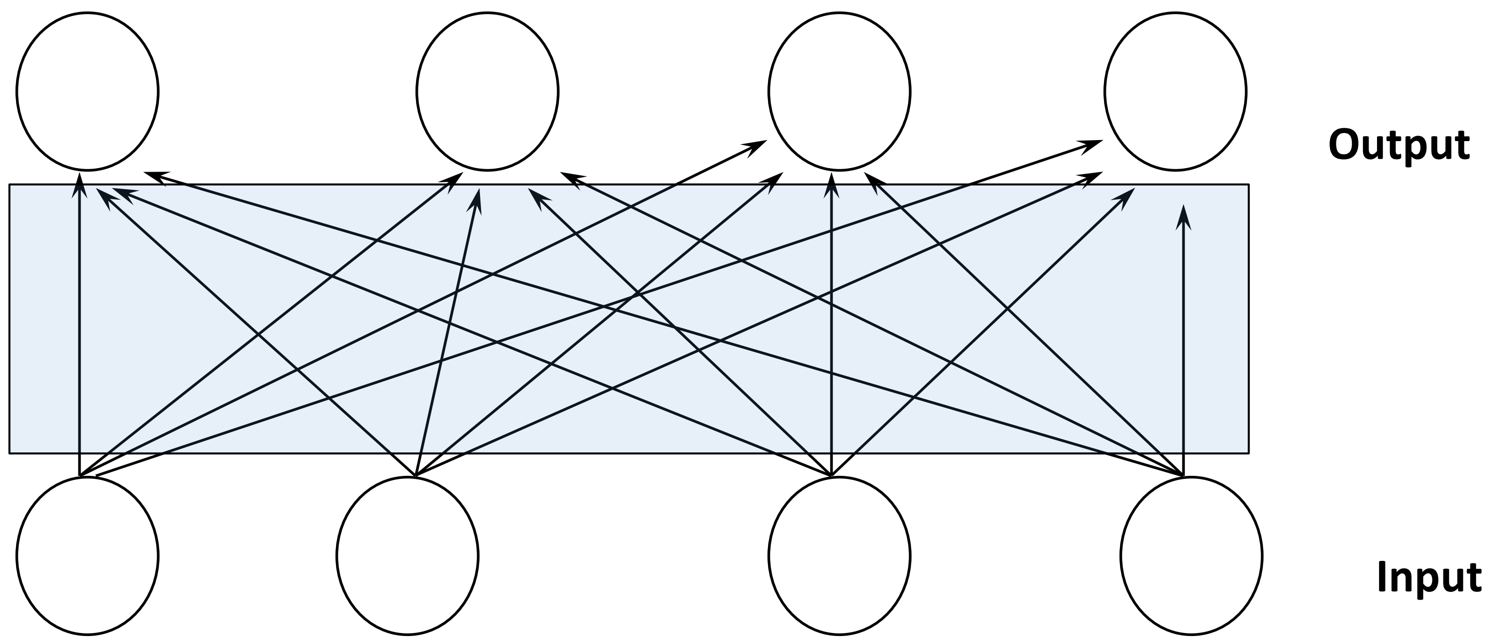
Artificial Neural Networks

- Neuronal Network Architecture (single layer)

$\vec{g}$

$\vec{A}$

$\vec{f}$



Artificial Neural Networks

- Thus, each time the input ***f₁*** appears, the network should respond to ***g₁***.

- Let's do

$$\vec{f}_1 = \frac{1}{2}\begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \qquad \vec{f}_2 = \frac{1}{2}\begin{bmatrix} +1 \\ -1 \\ -1 \\ +1 \end{bmatrix}$$

$$\vec{g}_1 = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \end{bmatrix} \qquad \vec{g}_2 = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

Artificial Neural Networks

- Thus, we have:

$$\vec{A}_1 = \begin{bmatrix} +1 & -1 & +1 & -1 \\ +1 & -1 & +1 & -1 \\ -1 & +1 & -1 & +1 \\ -1 & +1 & -1 & +1 \end{bmatrix} = \vec{g}_1 \cdot \vec{f}_1^{\,T}$$

Artificial Neural Networks

- Let us show the vector ***f₁*** and see what the answer is:

$$\vec{A}_1 \cdot \vec{f}_1 = \frac{1}{2}\begin{bmatrix} +1-1+1-1 \\ +1-1+1-1 \\ -1+1-1+1 \\ -1+1-1+1 \end{bmatrix} \cdot \frac{1}{2}\begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \end{bmatrix} = \vec{g}_1$$

Artificial Neural Networks

- What happens if we input a vector not memorized?

$$\vec{A}_1 \cdot \vec{f}_2 = \frac{1}{2}\begin{bmatrix} +1 -1 +1 -1 \\ +1 -1 +1 -1 \\ -1 +1 -1 +1 \\ -1 +1 -1 +1 \end{bmatrix} \cdot \frac{1}{2}\begin{bmatrix} +1 \\ -1 \\ -1 \\ +1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0$$

Artificial Neural Networks

- Will the network be able to memorize more than one vector?

- The answer is YES ! But how ?

Simply by calculating a new matrix $\overrightarrow{A2}$, now with vectors $\overrightarrow{f2}$ and $\overrightarrow{g2}$ and add it to the matrix $\overrightarrow{A1}$ previously determined.

Artificial Neural Networks

- Thus we have:

$$\vec{A}_2 = \vec{g}_2 \vec{f}_2^{\,T} = \frac{1}{2}\begin{bmatrix} -1 & +1 & +1 & -1 \\ -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

Artificial Neural Networks

- And then, it comes that:

$$\vec{A} = \vec{A}_1 + \vec{A}_2 = \frac{1}{2}\begin{bmatrix} 0 & 0 & +2 & -2 \\ 0 & 0 & +2 & -2 \\ 0 & 0 & -2 & +2 \\ 0 & 0 & -2 & +2 \end{bmatrix}$$

Artificial Neural Networks

- It is now easy to show that:

$$\vec{A} \cdot \vec{f}_1 = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \end{bmatrix} = \vec{g}_1$$

as well as:

$$\vec{A} \cdot \vec{f}_2 = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} = \vec{g}_2$$

Artificial Neural Networks

- Verification of the applicability of Hebb's law to solve the problem of AND
- Adopting Bipolar Notation Using Binary / Bipolar Conversion Function

$0 \rightarrow -1$

$1 \rightarrow 1$

Using the activation function g:= e (if $e_j > 0$ then $g_j = +1$ else $g_j = -1$)

AND Table

A	**B**	b	**A . B**
-1	**-1**	**1**	**-1**
-1	**1**	**1**	**-1**
1	**-1**	**1**	**-1**
1	**1**	**1**	**1**

Artificial Neural Networks

- **Hebb's Law**

$$\vec{A}.\vec{f} = \vec{g} \quad \Leftrightarrow \quad \vec{A} = \vec{g}.\vec{f}^{\,T}$$

$$\delta\vec{A}_1 = [-1]\ [-1,-1,1] = [1,1,-1]$$

$$\delta\vec{A}_2 = [-1]\ [-1,1,1] = [1,-1,-1]$$

$$\delta\vec{A}_3 = [-1]\ [1,-1,1] = [-1,1,-1]$$

$$\delta\vec{A}_4 = [1]\ [1,1,1] = [1,1,1]$$

Artificial Neural Networks

- **Hebb's Law**

Matrix calculation that learns the problem of AND

$$\vec{A} = \delta\vec{A}_1 + \delta\vec{A}_2 + \delta\vec{A}_3 + \delta\vec{A}_4 = [2,2,-2]$$

We can now verify by multiplying each of the transposed vectors that make up the matrix f by the results obtained from the previous equation.

Artificial Neural Networks

- Hebb's Law Verification

$$\vec{g}_1 = \vec{f}_1.\vec{A} = [-1,-1,1]^T [2,2,-2] = [-6] \longrightarrow [-1]$$

$$\vec{g}_2 = \vec{f}_2.\vec{A} = [-1,1,1]^T [2,2,-2] = [-2] \longrightarrow [-1]$$

$$\vec{g}_3 = \vec{f}_3.\vec{A} = [1,-1,1]^T [2,2,-2] = [-2] \longrightarrow [-1]$$

$$\vec{g}_4 = \vec{f}_4.\vec{A} = [1,1,1]^T [2,2,-2] = [2] \longrightarrow [1]$$

The system learned AND !!!!

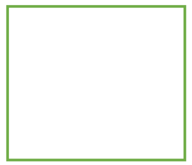
Artificial Neural Networks

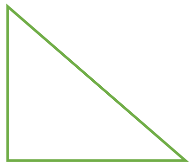
- Another example of application may be the application to recognition of graphics:

Hebb's law

- Choosing as graphics the square and the triangle, represented in a matrix of 4 points we have:

$$\vec{q} = [1,1,1,1] \qquad \vec{t} = [1,1,1,-1]$$





Artificial Neural Networks

Using the activation function g:= e (if ej > 0 then gj = +1 else gj = -1).

it can memorize the square vector:

$$\delta\vec{A}_1 = \vec{q}.\vec{q}^T = [1,1,1,1]^T[1,1,1,1] = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 \end{bmatrix}$$

and also the triangle vector:

$$\delta\vec{A}_2 = \vec{t}.\vec{t}^T = [1,1,1,-1]^T[1,1,1,-1] = \begin{bmatrix} +1 & +1 & +1 & -1 \\ +1 & +1 & +1 & -1 \\ +1 & +1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{bmatrix}$$

Artificial Neural Networks

The matrix that memorizes the 2 vectors is :

$$\delta\vec{A} = \delta\vec{A}_1 + \delta\vec{A}_2 = \begin{bmatrix} 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Applying a stimulus to the Hebbian matrix we have :

$$\vec{g} = \vec{A}.\vec{q} = \begin{bmatrix} 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} . \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 6 & 6 & 2 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

Artificial Neural Networks

The stimulus presented is recognized as a square !!

What happens if we present a stimulus of a triangle?

$$\vec{g} = \vec{A}.\vec{t} = \begin{bmatrix} 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} . \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 6 & 6 & 6 & -2 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix}$$

It is recognized as a triangle !!

Exercise: Try it now with a vector that represents a non memorized image and analyse the result ...

Artificial Neural Networks

- **Algorithm (Hebb's law)**

Step 0

Initialize weights to 0 $w_i = 0$ $(i = 1,2,\ldots,n)$

Step 1

For each pair $\vec{s} : \vec{t}$ (training vector and response) do steps 2, 3 and 4)

Step 2

Enable input units $\vec{x}_i = s_i$ $(i = 1,2,\ldots,n)$

Step 3

Enable output units $\vec{y} = \vec{t}$

Step 4

Adjust the weights

$$w_i(\text{new}) = w_i(\text{old}) + x_i y$$

Adjust bias

$$b(\text{new}) = b(\text{old}) + y$$

Agenda

1 – Overview and brief history of AI

- Historical Overview of AI and applications
- Paradigms and approaches

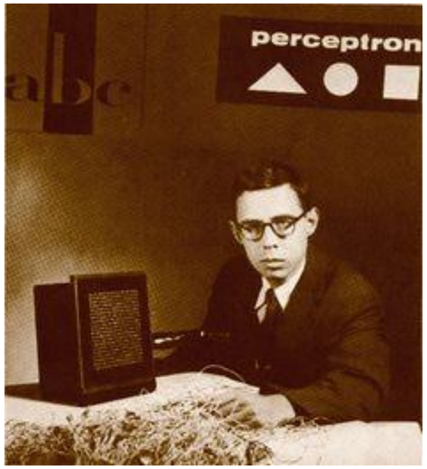
2 – An introduction to Artificial Neural Networks

- Neurobiology fundamentals
- McCulloch & Pitts’ Neuron
- Learning - Hebb's Law
- Supervised Neural Networks: Perceptron / Adaline…
- Unsupervised Neural Networks: competitive learning
- Multilayered artificial neural networks
  - Backpropagation
  - Deep Learning

Artificial Neural Networks

- **The Perceptron**

1958 - Frank Rosenblatt (Psychologist !!!)

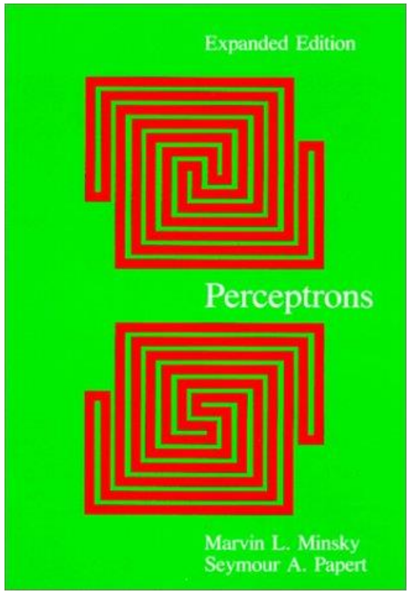


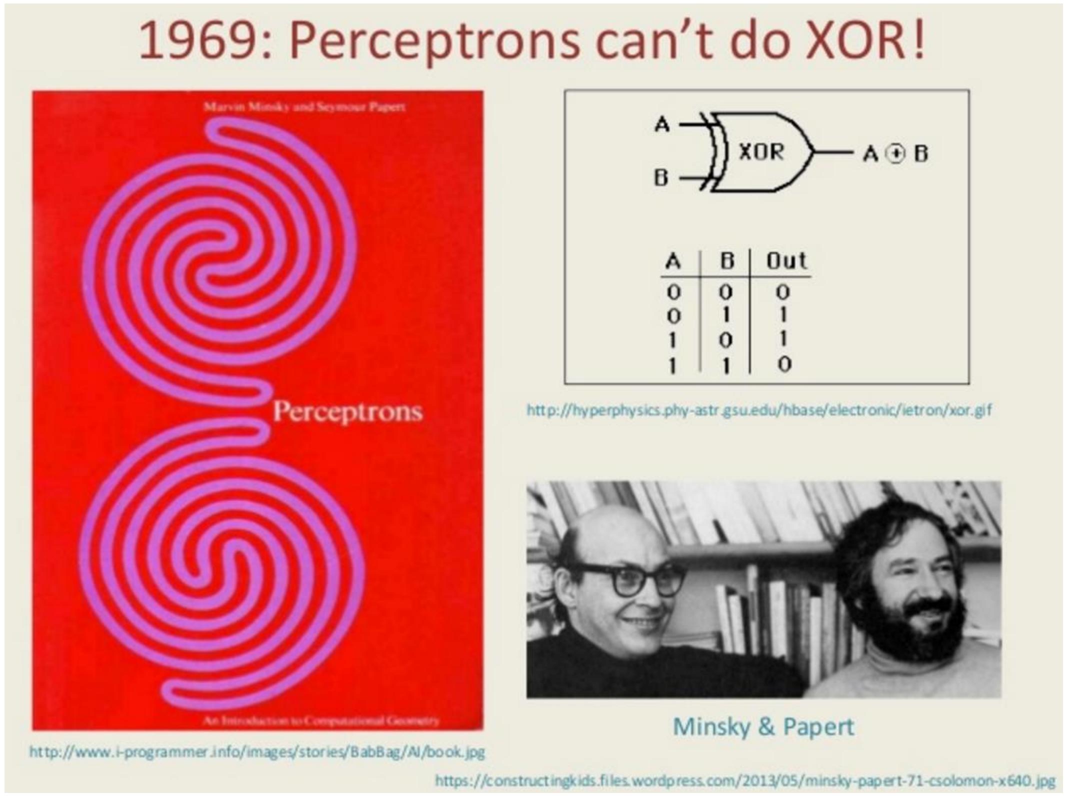
It was the first neuronal network to be "trendy" in scientific circles.

It seemed to be a learning machine! - Rosenblatt himself and his team said Perceptions could do everything !!!! The enthusiasm and fascination was such that it produced a kind of magic about the Artificial Neural Networks that still lasts to this day.

- **The Perceptron**

In 1969 Minsk and Papert showed in their historical book Perceptrons that they also have important theoretical limitations.





1969: Perceptrons can't do XOR!

Marvin Minsky and Seymour Papert
Perceptrons
An Introduction to Computational Geometry
A
B
XOR
A ⊕ B

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/ietron/xor.gif

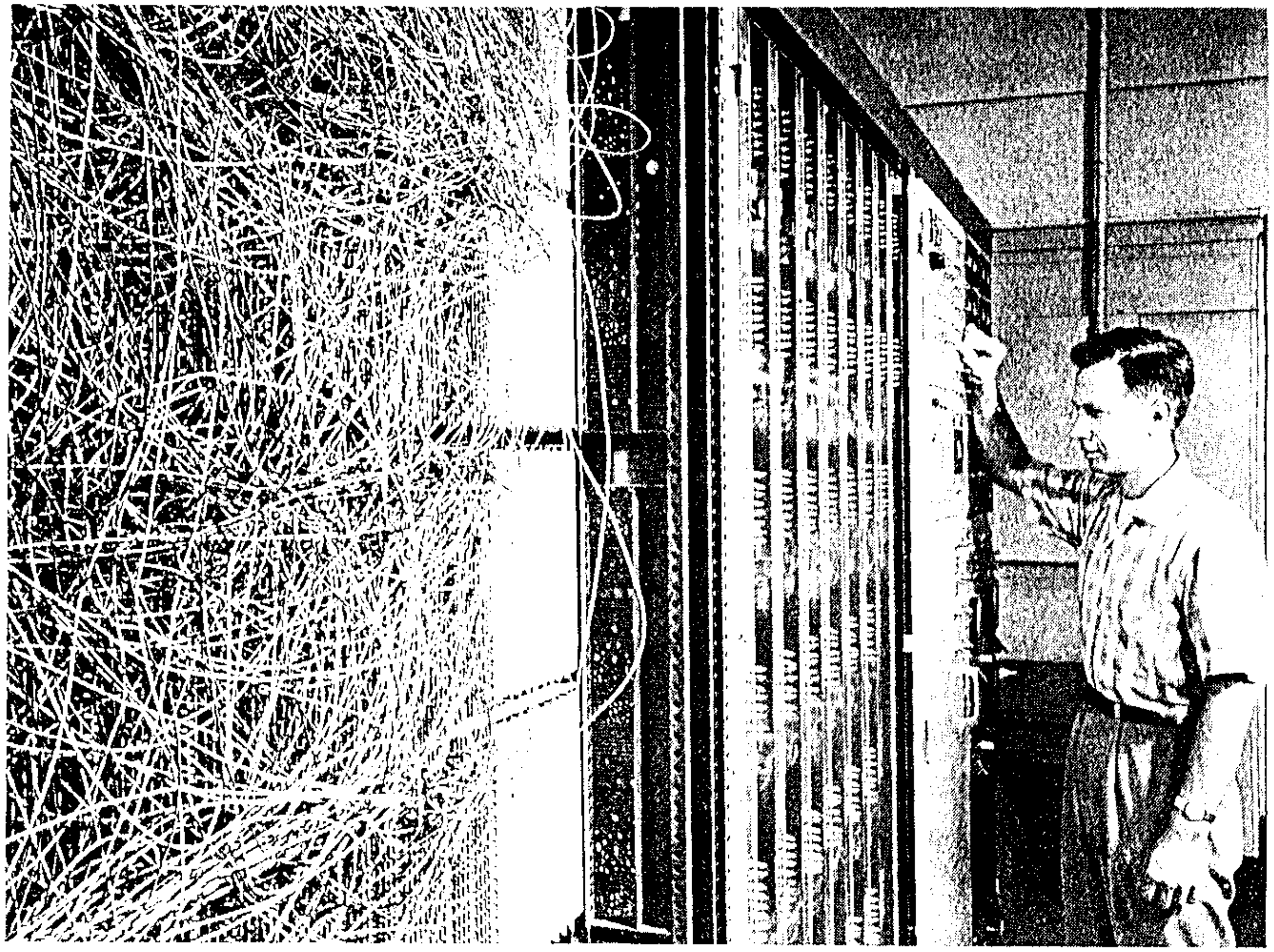
Minsky & Papert

http://www.i-programmer.info/images/stories/BabBag/AI/book.jpg

https://constructingkids.files.wordpress.com/2013/05/minsky-papert-71-csolomon-x640.jpg

Artificial Neural Networks

Perception was not just a theoretical machine…



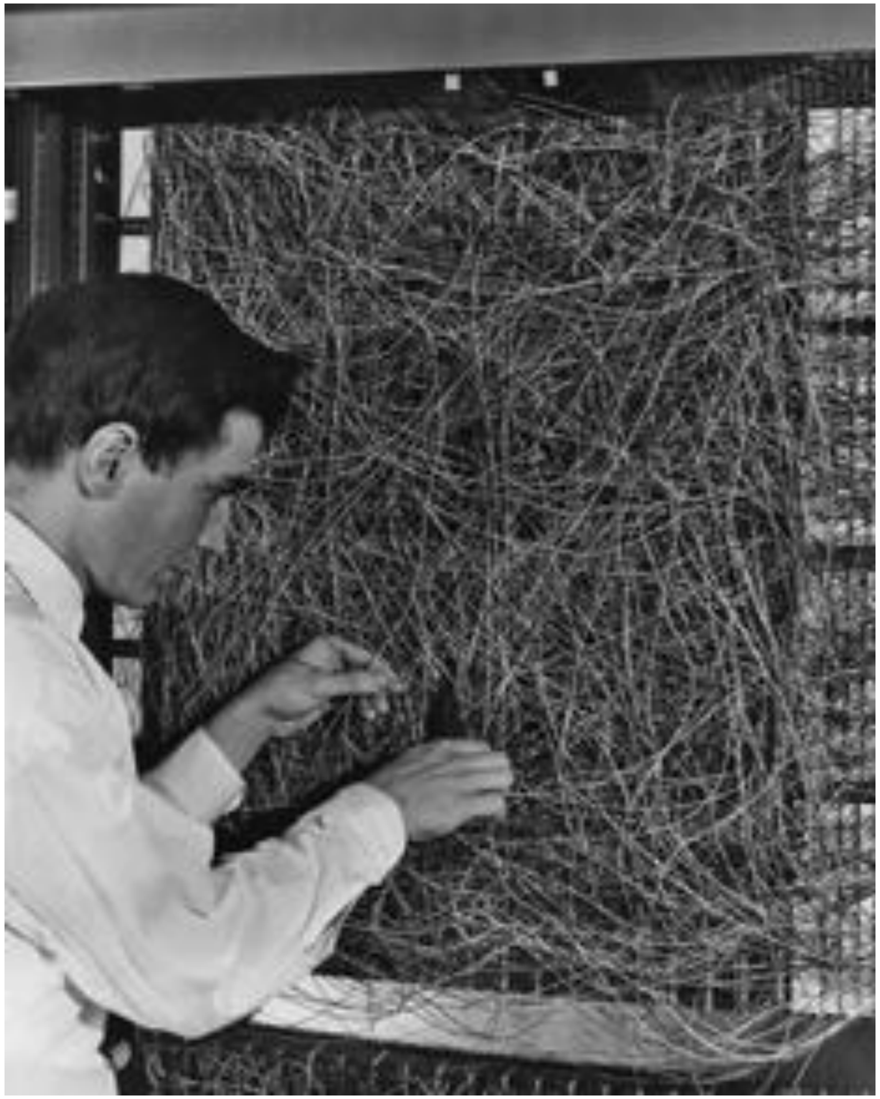
Artificial Neural Networks

- In the previous slide we can see the photograph of the first built perceptron (photo by Alvin Calspan)

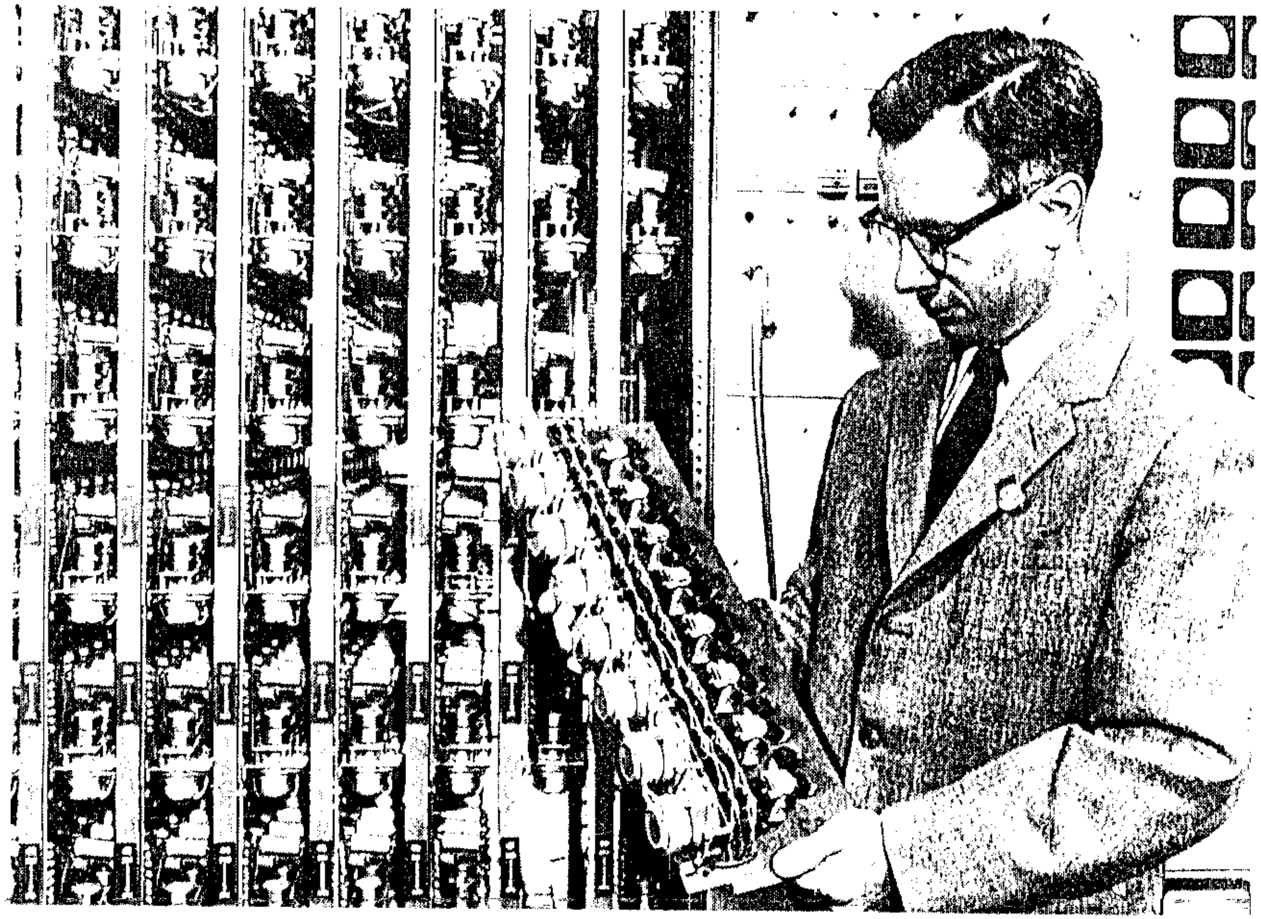
- Mark I Perception Dashboard The pattern of the connections was typically “random,” to illustrate the ability of the perceptron to learn the desired pattern without needing precise connections (as opposed to a computer).

NOVA IMS Information Management School

Artificial Neural Networks



Artificial Neural Networks



Charles Wightman holding a plate of 8 pairs of motors / potentiometers. Each motor / potentiometer pair functioned as a single weight value adapter. The perceptron learning law was implemented with analogic circuits that controlled the motor of each potentiometer (the resistance that worked to vary the weights).

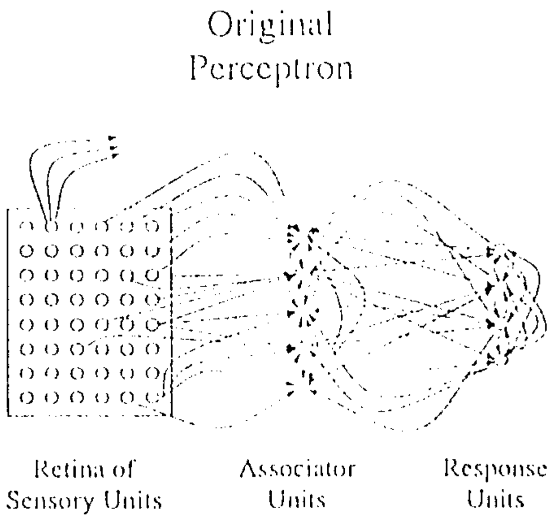
NOVA IMS Information Management School

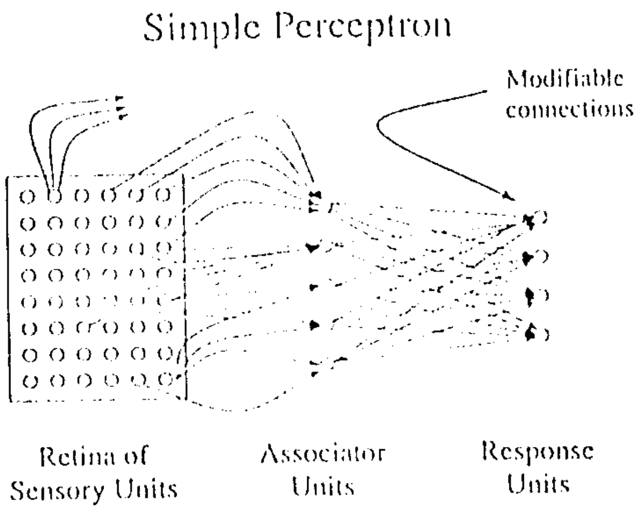
Artificial Neural Networks

NOVA IMS Information Management School

Artificial Neural Networks

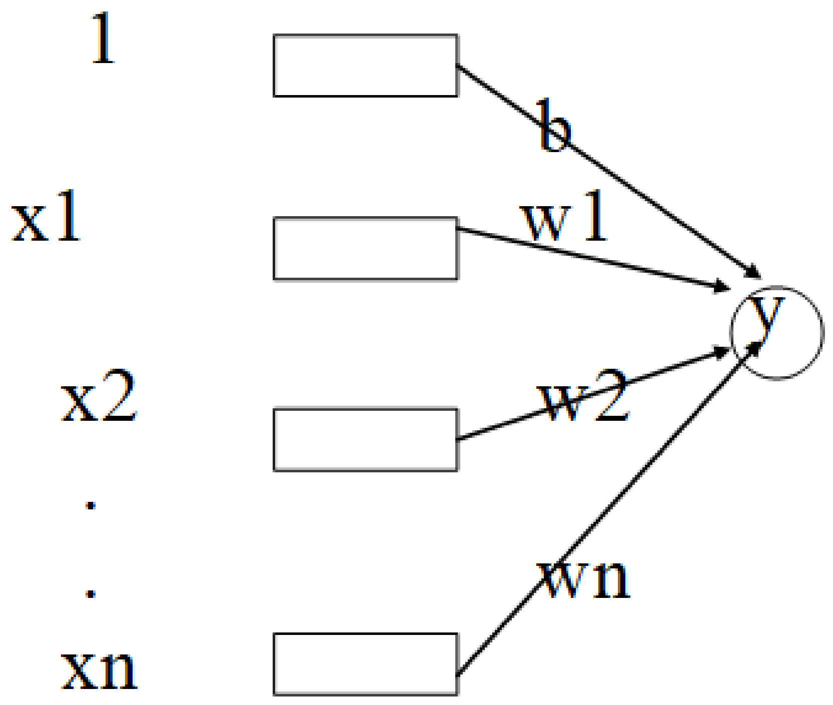
The original Perception consisted of three layers of neurons: sensor layer, association layer, and response layer.





Artificial Neural Networks

Elemental Perception Architecture



$$\overrightarrow{y\_in} = b + \sum_i \overrightarrow{xi}.\overrightarrow{wi}$$

Artificial Neural Networks

The perceptron responses are coded as follows :

$$y_{out} = \begin{cases} +1, y\_in > \theta \\ 0, -\theta \le y\_in \le \theta \\ -1, y\_in < -\theta \end{cases}$$

Artificial Neural Networks

- Perception Algorithm (Teach)
  - **Step 0**

Initialize the weights and the bias (usually b=0 and $w_i = 0$ (i = 1,2,…,n))

Initialize the learning coefficient α ($0 < \alpha <= 1$) (usually = 1)

- **Step 1**

While there is no stopping condition do steps 2, 3,4,5 and 6

- **Step 2**

For each pair s:t (training vector and its response) do steps 2, 3, 4,5

- **Step 3**

Enable input units $x_i = s_i$ (i = 1,2,…,n)

- **Step 4**

Calculate unit response

$$y\_in = b + \sum_i xi.wi$$

Artificial Neural Networks

- Perception Algorithm(cont.)

- **Step 5**

Calculate Y_{out}

$$\vec{y_{out}} = \begin{cases} +1, y\_in > \theta \\ \_0, -\theta \le y\_in \le \theta \\ -1, y\_in < -\theta \end{cases}$$

- **Step 6**

if error exists adjust bias and weights ($y \neq t$)

w_i(new) = w_i(old) + $\alpha t x_i$

b(new) = b(old) + αt

otherwise keep the old values

Artificial Neural Networks

- Perception Algorithm(Ask)

1 - Enable input units $x_i = s_i$ (i = 1,2,…,n)

2 - Calculate response of each unit

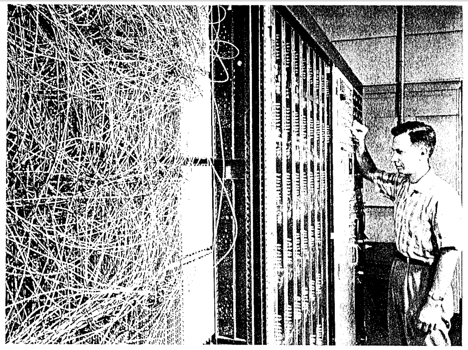
$$\overrightarrow{y_{in}} = b + \sum_i \overrightarrow{xi}.wi$$

3. Calculate $\overrightarrow{Y_{out}}$

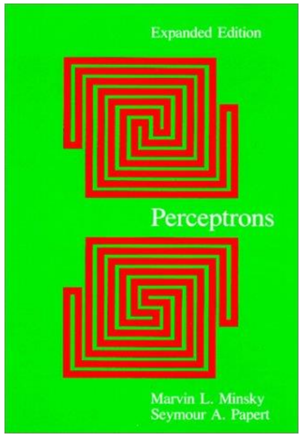
$$y_{out} = \begin{cases} +1, y\_in > \theta \\ \_0, -\theta \le y\_in \le \theta \\ -1, y\_in < -\theta \end{cases}$$

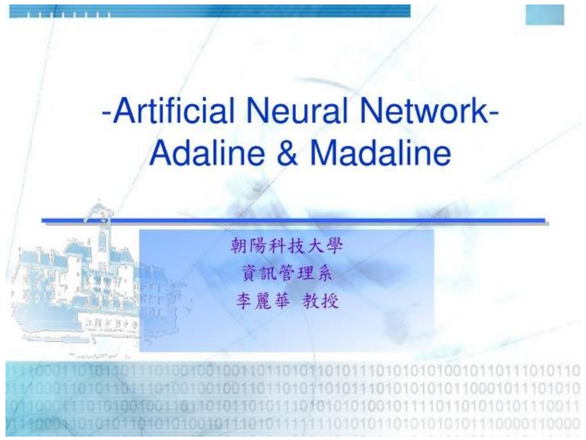
NOVA IMS Information Management School

Simple Pattern Recognition Networks



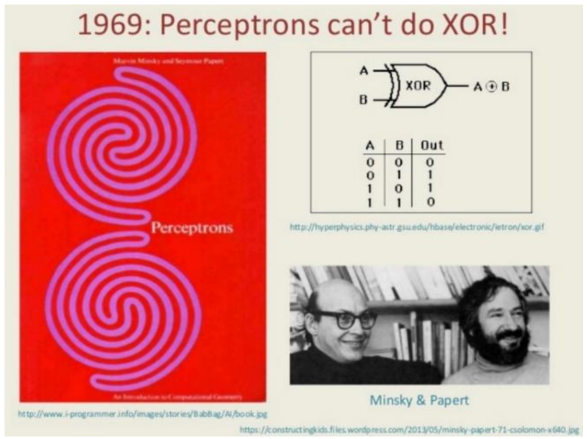
- Hebbian Network
- Perceptron
- Adaline
- Madaline (A multilayer network of ADALINE units is known as a MADALINE)
- …





-Artificial Neural Network-
Adaline & Madaline

朝陽科技大學
資訊管理系
李麗華 教授



1969: Perceptrons can't do XOR!

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/ietron/xor.gif

Minsky & Papert

http://www.i-programmer.info/images/stories/BabBag/AI/book.jpg

https://constructingkids.files.wordpress.com/2013/05/minsky-papert-71-csolomon-x640.jpg

NOVA IMS Information Management School

Simple Pattern Recognition Networks

- Adaline (Delta Rule)

Bernard Widrow
Stanford University

Marcian Hoff (**Ted Hoff)**

https://en.wikipedia.org/wiki/Marcian_Hoff

https://prezi.com/0jqghsgnvqal/marcian-edward-hoff/

NOVA IMS Information Management School

Widrow-Hoff Learning - ADALINE (1960)

- It assumes that correcting a weight requires only the information accessible to each adaptive unit: the stimulus, the actual response, and the intended response.

The processing of each unit is independent of the activity of the other units.

To associate $\vec{f}$ with $\vec{t}$, we have so far built the Hebb matrix:

$$\vec{A} = \alpha\, t\, f$$

However, t is only reproduced in case the final matrix $\vec{A}$ involves few associations.

Widrow-Hoff Learning - ADALINE

- We can increase learning accuracy by using the Widrow-Hoff algorithm. Suppose we intend to reproduce an association f: t, learned but forgotten by the excess information stored in $\vec{A}$:

$\vec{A}\,\vec{f} = \vec{g}$, where g is not exactly the correct association t.

The difference $(\vec{t}-\vec{g})$ constitutes the mistake made Δ.

The Widrow-Hoff method consists of learning this error: the final matrix will eventually be the sum of the desired associations, plus a number of corrective terms that tend to correct the error.

Widrow-Hoff Learning - ADALINE

- Let's look at 1st iteration:

  Suppose that matrix A is corrected by the term α A:

  $= \overrightarrow{A} = \alpha (\vec{t} - \vec{g})\, \vec{f}^T$, where α is the, or:

  learning coefficient $\overrightarrow{A} = \alpha (\vec{t} - \overrightarrow{A}\vec{f})\, f^T$

  The correction is not perfect however. Corrections are continued until the system produces the smallest possible error

Widrow-Hoff Learning - ADALINE

- Algorithm
  - 1. Random extraction of a pair $\vec{f}$: $\vec{t}$ from the training set;
    2. Operation of $\vec{A}$ over $\vec{f}$, producing $\vec{g}$;
    3. Determination of the difference (t-g);
    4. Correction of matrix $\vec{A}$ with a term proportional to the outer product, $(t-g)\,\vec{f}^T$, between error and stimulus;
    5. Repeat (1-4) until desired accuracy is achieved.

Widrow-Hoff Learning - ADALINE

Realizing,

α - learning coefficient

x - input vector

t - output vector for input vector x

y - output vector obtained by input of vector

$$\overrightarrow{y}_j = \sum_i x_i w_{ij}$$

We know that an error is being made, which is given by the expression:

$$\Delta w_{ij} = \alpha.(t_j - y_j).x_i$$

(Why ??)

Widrow-Hoff Learning - ADALINE

(Why ??) The error made for a particular training pattern is given by:

$E=(t - y\_in)^2$, where y_in is the activation of output neuron Y

Where **E** is a function of all weights w_i, i = 1,…, n. The gradient of E will be a vector consisting of the partial derivatives of **E** with respect to each of the weights. Thus, this gradient gives us the direction of the fastest increment of **E**. The opposite direction will give the largest decrease of the error. That is, the error can be reduced by adjusting the weight w_i in the direction of

Once y_in = $\sum_{i=1}^{n} x_i w_i$, comes: $-\frac{\partial E}{\partial w_I}$

Widrow-Hoff Learning - ADALINE

$$\frac{\partial E}{\partial w_I} = -2(t - y\_in)\frac{\partial y\_in}{\partial w_I}$$

$$\frac{\partial E}{\partial w_I} = -2(t - y\_in)x_I$$

Thus, the local error will be reduced faster (for a given learning coefficient) by adjusting the weights according to the delta rule (name also given to Widrow-Hoff learning):

$$\Delta w_I = \alpha.(t - y\_in).x_I$$

Widrow-Hoff Learning - ADALINE

Now that we know why, we can say that the changes that will have to be made to the weights are given by the expression:

$$w_{ij}(new) = w_{ij}(old) + \alpha.(t_j - y_j).x_i \qquad (i = 1,\dots,n;\ j = 1,\dots,m)$$

Widrow-Hoff Learning - ADALINE

- Coefficient of Learning

  α **large** Then the value of corrections grows indefinitely (easily detectable problem).

  α **small** If too small, then the algorithm will converge very slowly to the correct answer.

  α **constant** So matrix A may not converge; will swing around the best solution.

  α **variable** If α decreases as learning progresses, then matrix converges, possibly before full error correction.

Coefficient of Learning

Often one takes α = 1 / n, where n is the number of learning attempts.

Widrow-Hoff Learning - ADALINE

- Even with the ADALINE (delta-rule), it turns out that the classification of standards does not become universal, that is, there are still situations of error.
- This demonstration of the limitations of the so - called single - layered neuron networks was a determining factor in the decline in interest in the same networks of the 1970s.
- This interest is reborn with Rumelhart (1986), who present a method for training multi-layer networks.
- This method is called backpropagation or generalized delta-rule

Agenda

1 – Overview and brief history of AI

- Historical Overview of AI and applications
- Paradigms and approaches

2 – An introduction to Artificial Neural Networks

- Neurobiology fundamentals
- McCulloch & Pitts' Neuron
- Learning - Hebb's Law
- Supervised Neural Networks: Perceptron / Adaline…
- Unsupervised Neural Networks: competitive learning
- Multilayered artificial neural networks
  - Backpropagation
  - Deep Learning

Competitive Artificial Neural Networks

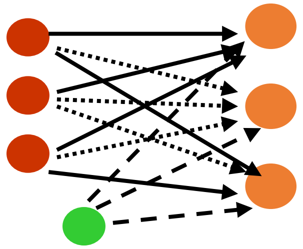
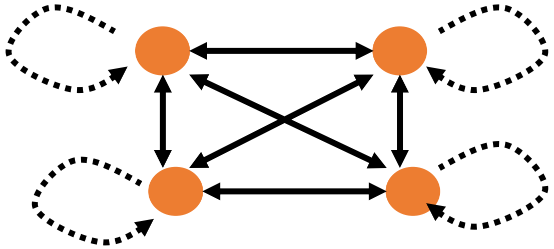
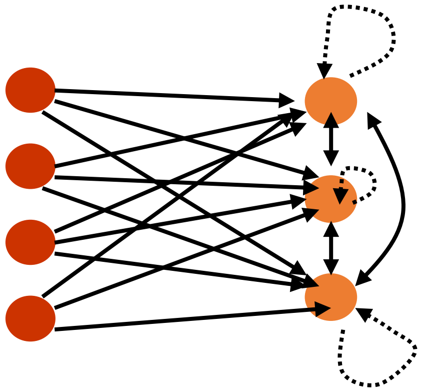
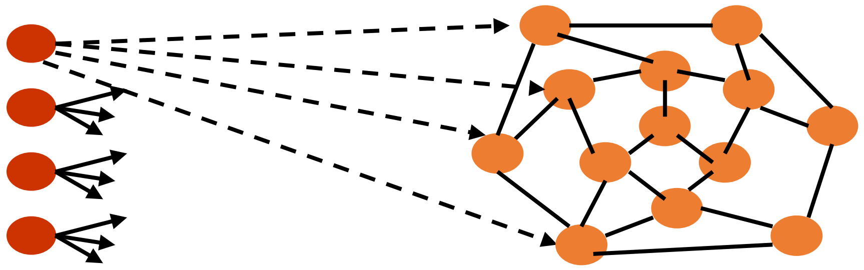
- Until now the networks studied had many active outputs, in competitive learning only one output unit can be active at any given time.
- Output units compete with each other to be active in response to a given input.
- The neurons in a competitive network receive identical information at the entrance and compete for being the only one active.

Competition

- Competition can be of two types:
  - Hard competition, only one neuron can stay active (Winner-take-all);
    - Usually, the squared Euclidean distance is used to determine the closest weight vector to a pattern vector
    - Only the neuron with the smallest Euclidean distance from the input vector is allowed to update
  - Soft competition, there is a winner and your “neighbours” share a small percentage of this activation.

Network Types for Clustering

- Winner-Take-All Networks
  - Hamming Networks
  - Maxnet
  - Simple Competitive Learning Networks
- Topologically Organized Networks
  - Winner & its neighbors “take some”

SOM Neural Networks

Teuvo Kalevi Kohonen
(11 July 1934 – 13 December 2021)

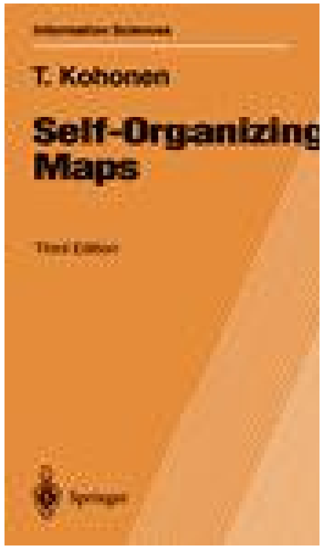
The SOM Kohonen algorithm was developed by **Teuvo Kohonen** in 1982, being considered relatively simple and with the capacity of dimensionally organizing complex data into clusters, according to their relations.

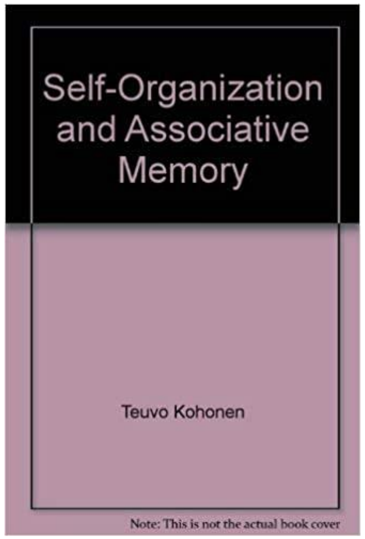
This method requires only the input parameters being ideal for problems whose patterns are unknown or indeterminate.

SOM Neural Networks

- SOM = Self- Organized Map
  - Self-organized map
  - KOHONEN Maps, or KOHONEN Neural Networks

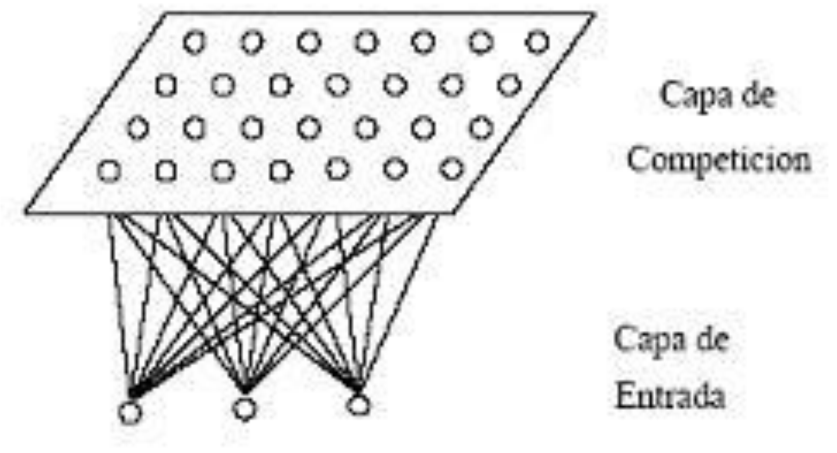
- Neural network for unsupervised learning
  - Visualization of multidimensional data, projection of data on a space of smaller dimension, clustering, detection of novelties




Self-Organization and Associative Memory
Teuvo Kohonen
Note: This is not the actual book cover

SOM Neural Networks

- The network consists of two layers of neurons connected by weights.
- The input layer is connected to an input vector of the data set and the output layer forms a map consisting of a grid where several neurons are arranged.



SOM Neural Networks

- The neurons in the Kohonen network are placed on the nodes of a grid (which has a particular topology, which may be rectangular, hexagonal, etc.), which is usually one or two dimensions. Larger maps are also possible, but more difficult to apply and understand.

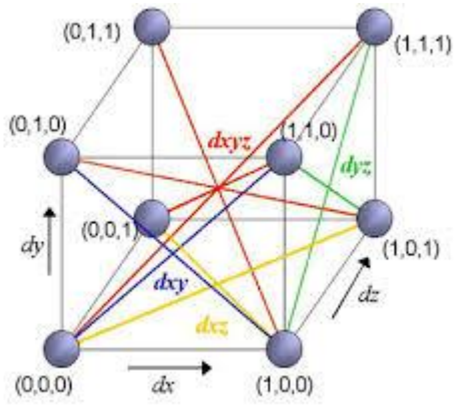
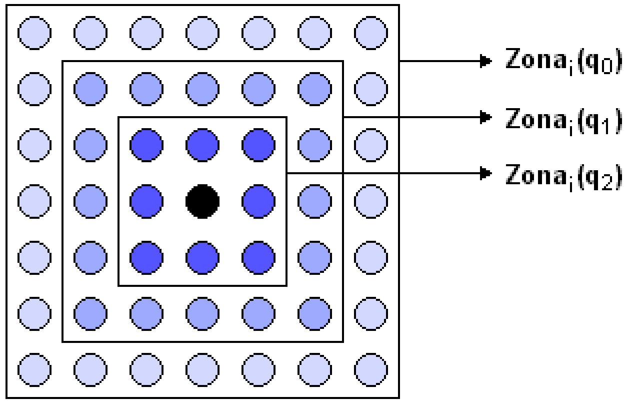
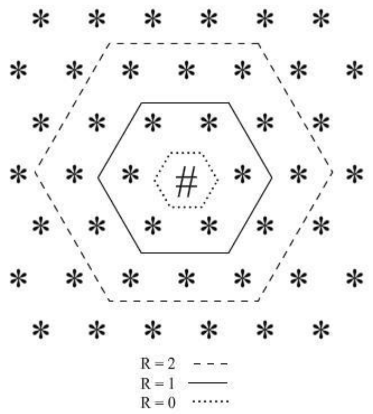


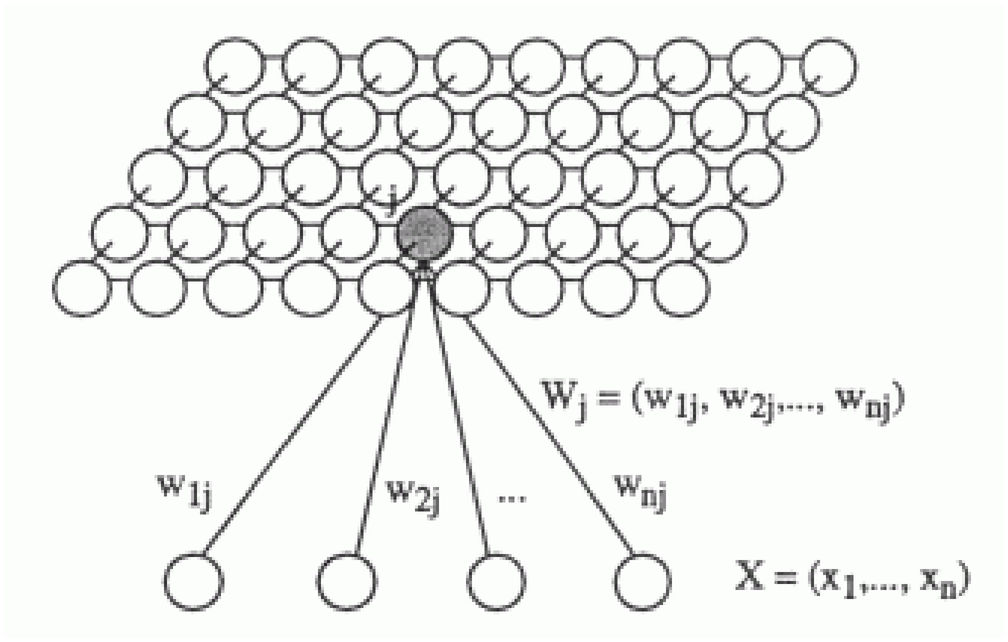
Figura 6: Ilustração do mapa de tamanho 2×2×2.





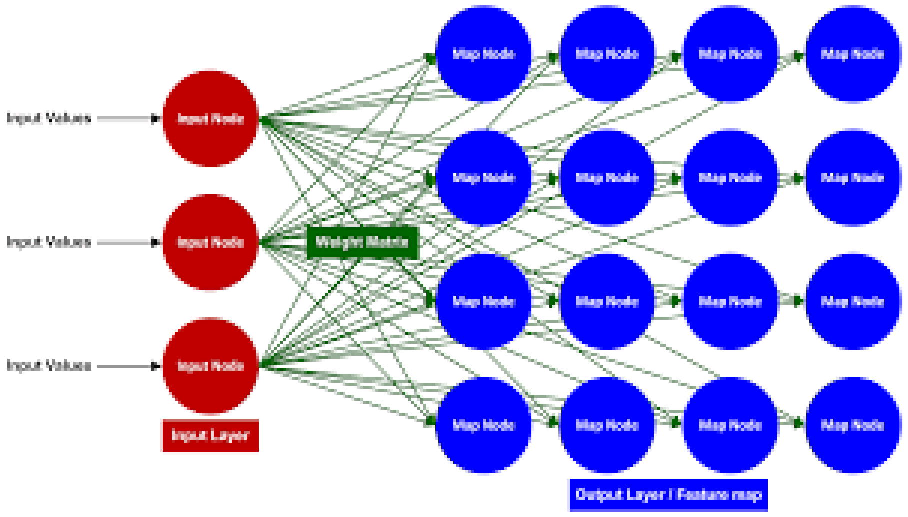
Kohonen Neural Networks

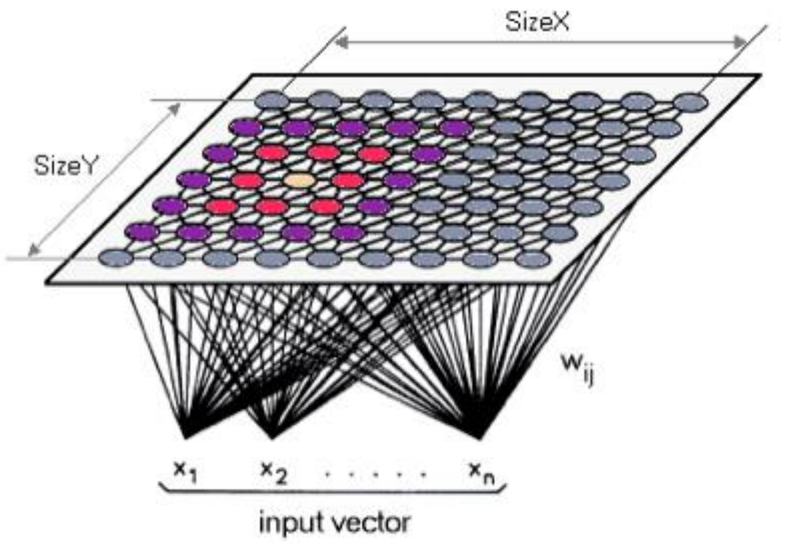
- Each neuron in the grid is connected to all neurons in the input layer.
- The input patterns are presented to the grid and for each pattern presented you get an activity region in the grid



SOM Neural Networks

- The location and nature of a particular region vary from one entry pattern to another.
- Therefore, all network neurons must be exposed to a sufficient number of different input patterns, thus ensuring that the process of self-organization occurs properly.





SOM Neural Networks

For each input pattern, the distance between the data pattern and all neurons is calculated.

Distance function :

- Usually the Euclidean distance is used
- Other measures of distance (or similarity)
  - Measures of Minkowski (blocks of city or Manhattan, of 3rd order, etc.)
  - Internal products
  - Hamming
  - Tanimoto Coefficients
  - Angle between vectors
  - Distances from Hausdorff
  - …

Euclidean $$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan $$\sum_{i=1}^{k}\left|x_i - y_i\right|$$

Minkowski $$\left(\sum_{i=1}^{k}\left(\left|x_i - y_i\right|\right)^q\right)^{1/q}$$

Kohonen's algorithm, step-by-step (CHON, 1996):

- Step 0: INITIALIZATION
    - Initialize the initial learning coefficient (η =)
    - Initialize the topological radius (r =)
    - Initialize weights: Matrix W i_x j_x k (where: i = number of columns of the data file, j = number of columns of the map and k = number of rows of the map), with random values between 0 and 1
- Step 1: STOP CRITERION
    - Set the maximum number of iterations to perform (e =)
    - Compare the maximum number of iterations and the change in the values of the weight matrix.
- Step 2: TRAINING For each vector xi:
    - <u>Competitive Phase</u>
    - Calculate the distances (D (j) = sum ($w_{ij-}x_i$) 2) from each row of the data matrix with all columns of the weight matrix
    - Find the index j such that D (j) is minimal (it is the winning neuron)

Kohonen Neural Networks

Kohonen's algorithm, step-by-step (CHON, 1996):

Step 2: TRAINING (continued). For each vector x_i:

Cooperative Phase

- For all columns of the weight matrix:
- Find the neighbours of the winning neuron (j) by the topological radius (if the neuron analysed is part of the neighbourhood, do z (k) = 1, otherwise z (k) = 0)

Adaptive Phase

- To update the entire matrix weight: $w_{ij}(t + 1) = w_{ij}(t) + \eta (x_i - w_{ij}(t)) * z(k)$

- Repeat Step 2: with the input of a new x_i

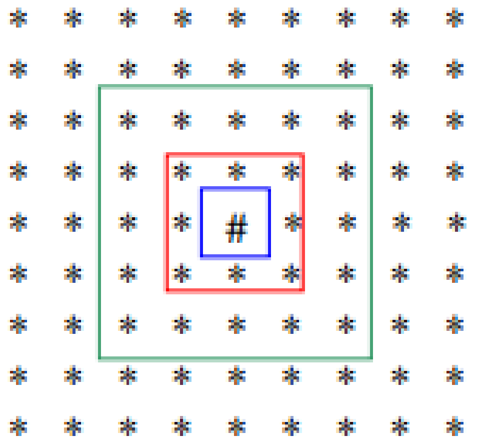
Kohonen Neural Networks

Kohonen's algorithm, step-by-step (CHON, 1996):

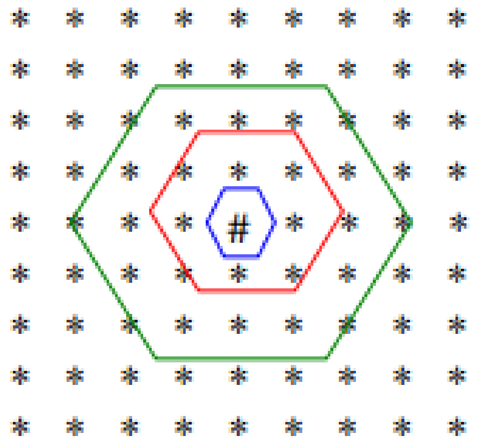
- Step 3: UPDATE
  - Update learning rate: η(t) (a linear, exponential or geometric decreasing function depending on the iterations)
  - Update topological radius: (defined as a monotonous function decreasing as a function of the iterations)
  - Repeat Step 1
- END
- DATA OUTPUT

Learning rate generally decreases with time
η(t)

Kohonen Neural Networks



neighborhood with rectangular grid



neighborhood with hex grid

# - winning neuron

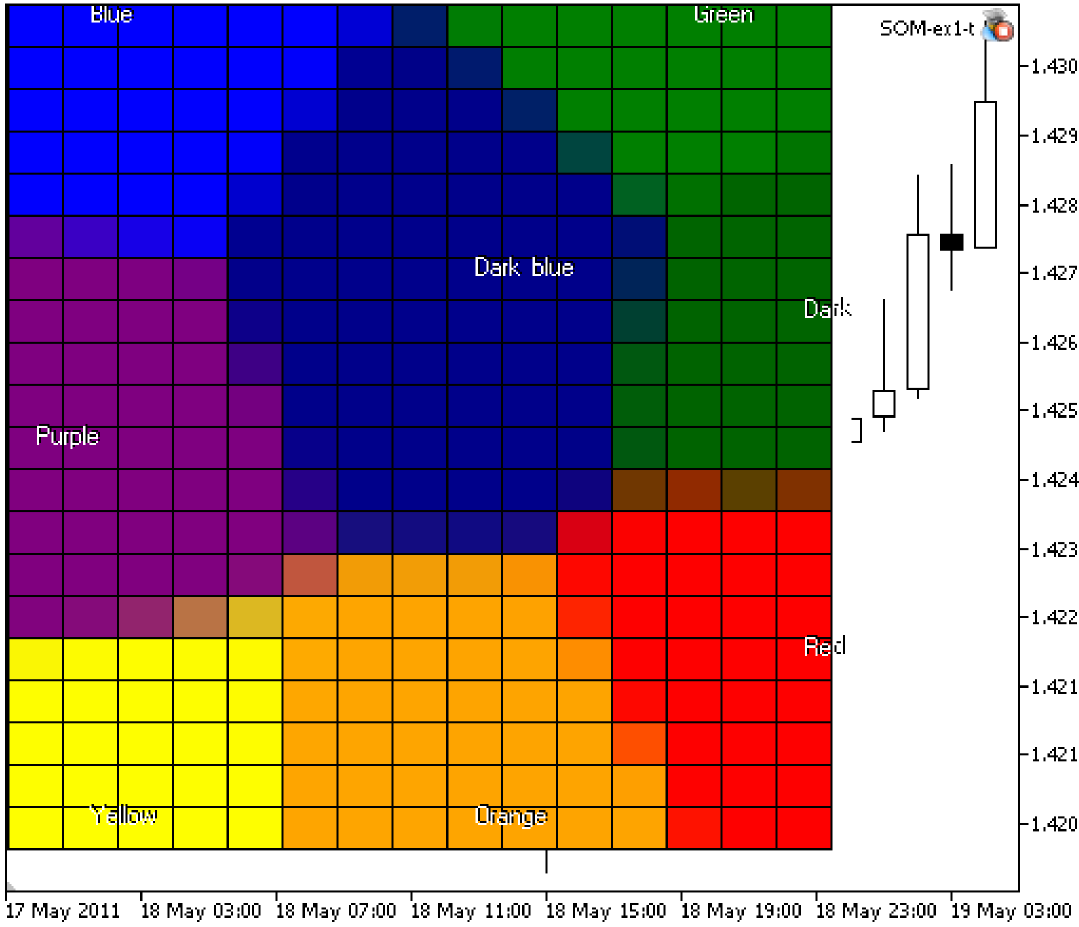
🟩	Radius = 2
🟥	Radius = 1
🟦	Radius = 0

Topological parameters of neighborhood and topological radius

Kohonen Neural Networks

The classic Kohonen map example is a color-matching problem.
Let's say we have a set of 8 colors. Each of these is represented as a three-dimensional vector in the RGB color model.

🟥 1. Red: (255.0,0);
🟩 2. Green: (0.128.0);
🟦 3. Blue: (0.0255);
🟩 4. Dark green: (0.100.0);
🟦 5. Dark blue: (0.0139);
🟨 6. Yellow: (255,255.0);
🟧 7. Orange: (255,165.0);
🟪 8. Purple: (128.0,128).



https://www.youtube.com/watch?v=3UOnOpUeZwk

https://www.youtube.com/watch?v=2fRcfk0Gevs

Agenda

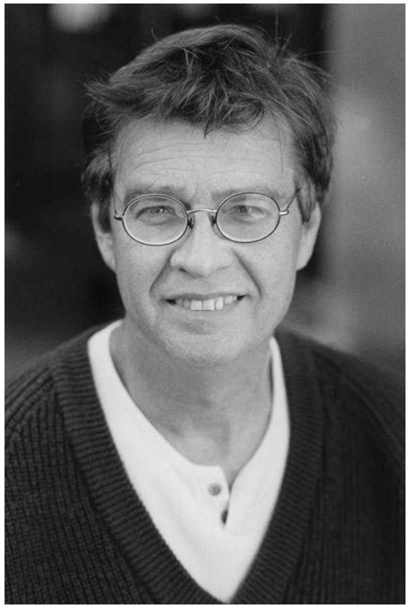
1 – Overview and brief history of AI
- Historical Overview of AI and applications
- Paradigms and approaches

2 – An introduction to Artificial Neural Networks
- Neurobiology fundamentals
- McCulloch & Pitts’ Neuron
- Learning - Hebb's Law
- Supervised Neural Networks: Perceptron / Adaline…
- Unsupervised Neural Networks: competitive learning
- Multilayered artificial neural networks
  - Backpropagation
  - Deep Learning

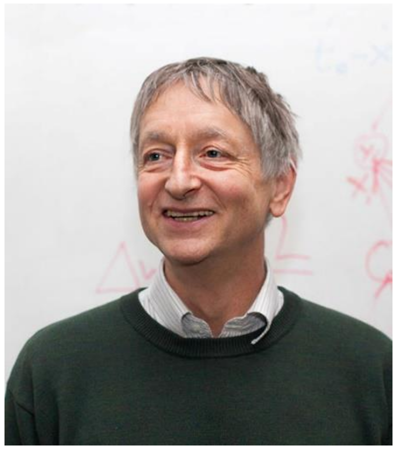
Backpropagation

A method for training multi-layer networks.

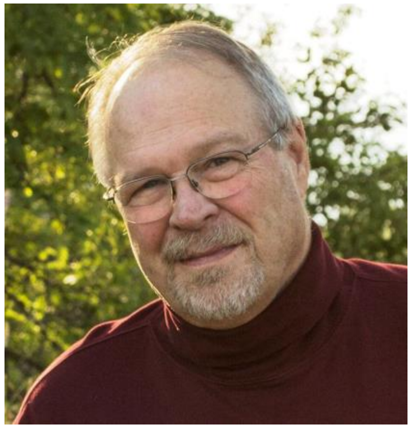
David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams (1986)



David Everett Rumelhart
(June 12, 1942 – March 13, 2011)
Stanford University



Geoffrey E. Hinton
Carnegie-Mellon University,
Pittsburgh, Philadelphia



Ronald J. Williams
Northeastern University

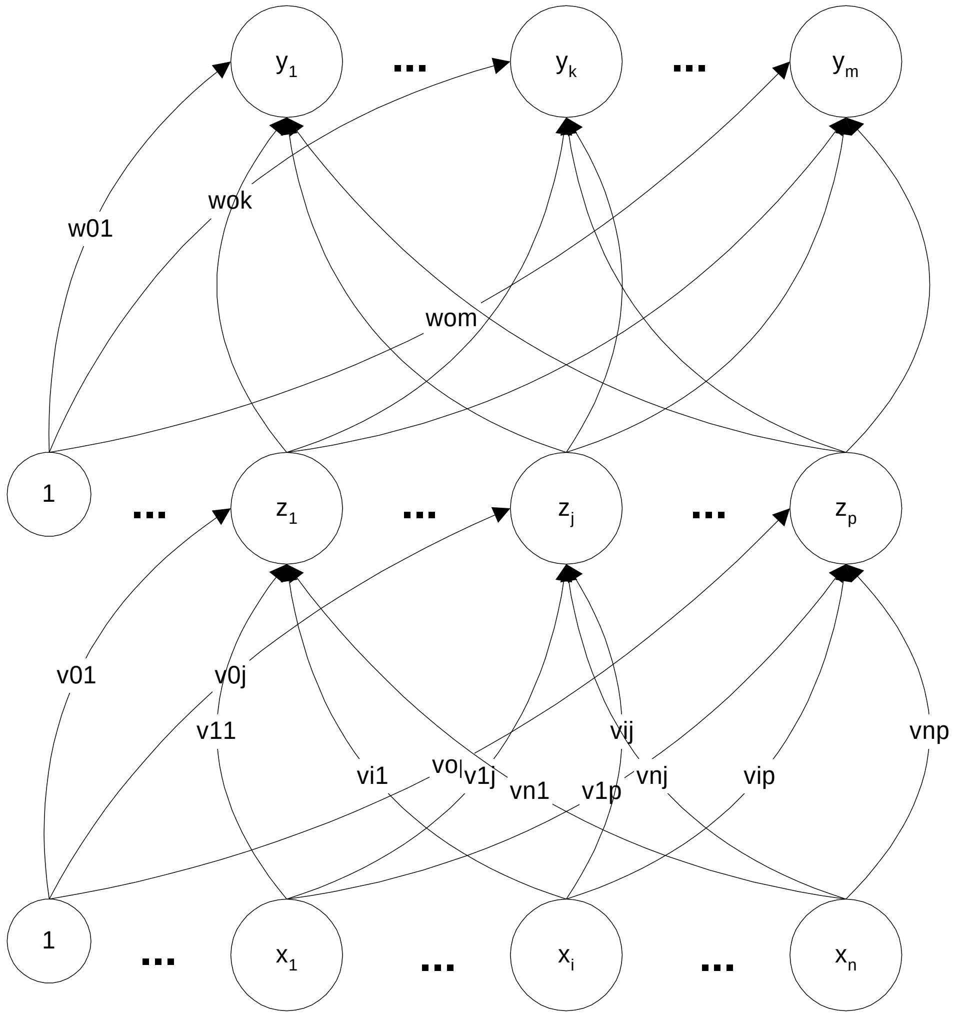
David E. Rumelhart, Geoffrey E. Hinton und Ronald J. Williams (1986). Learning representations by back-propagating errors., Nature (London)

Backpropagation/multi-layer perceptron

- Architecture

The training of a multi-layer network by backpropagation is carried out in 3 stages:

- input of the training input pattern
- Associated Error Backpropagation
- adjustment of weights

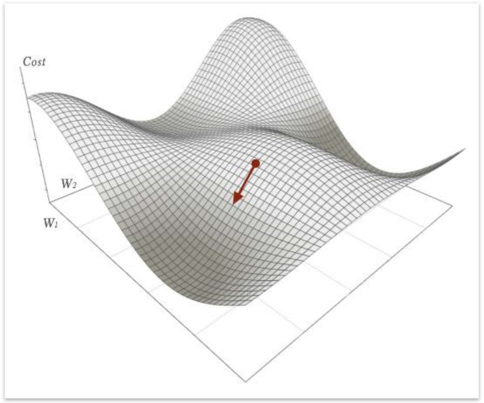


Backpropagation/multi-layer perceptron

- Algorithm

The training of a multilayer network by back propagation is done in 3 steps:

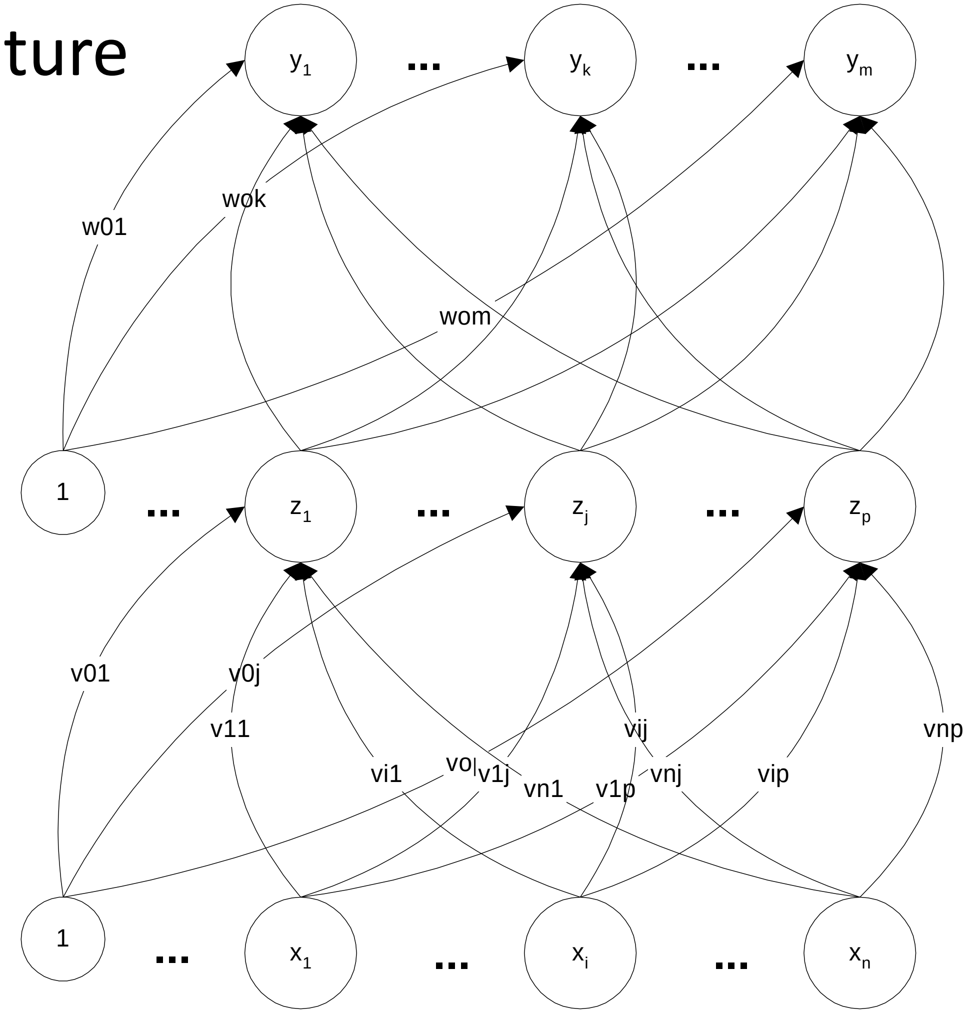
- **training input pattern**
- **backpropagation of associated error**
- **weight adjustment**
  - The adjustment of weights is performed using a non-linear optimization technique called stochastic gradient descent.



Backpropagation/multi-layer perceptron

- Algorithm - Nomenclature

  - x training vector (input)
  - t training vector (output)
  - δ_k setting value for weight wjk, due to error in unit Yk
  - δ_j setting value for weight v_{ij}, due to the backpropagation of error information from the output layer to the hidden layer Z_j
  - α learning coefficient
  - X_i input unit I
  - v_{0j} hidden unit pendant j
  - Z_j hidden unit j
  - w_{0k} output unit slope k
  - Y_k output unit k



Backpropagation/multi-layer perceptron

- Algorithm - Activation Function

When using binary encoding, use binary sigmoid function, defined by :

$$f(x) = \frac{1}{1+\exp(-x)}$$

In the use of bipolar coding, use bipolar sigmoid function, defined by:

$$f(x) = \frac{2}{1+\exp(-x)} - 1$$

But we can used other functions if they are:

- Continuous
- Differentiable
- Monotonous growing

and have:

- Simple derivation calculus
- An asymptotic behaviour for the extreme values of the action potential

Backpropagation/multi-layer perceptron

Algorithm

Step 0. Initialize weights (random values - but not indifferent → Random assignment or Assign values to weights between -0.5 and +0.5.)

Step 1. While stopping condition is false, repeat Steps 2-9.

Step 2. For each x: t pair, perform Steps 3-8.

(FeedForward)

Step 3. Xi=xi

Step 4. $z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$

$z\_out_j = f(z\_in_j)$ (f is the activation function – ep sigmoid example)

Step 5. $y\_in_k = w_{0k} + \sum_{j=1}^{p} z\_out_j w_{jk}$

$y\_out_k = f(y\_in_k)$

Backpropagation/multi-layer perceptron

- Algorithm (cont.)

(Backpropagation of error)

Step 6. Calculate error in unit Y_k as well

the corrections to weights at the 2nd tier layer :

$\delta_k=(t_k\text{-}y\_out_k).f'(y\_in_k)$

$\Delta w_{jk}=\alpha\delta_k z\_out_j$

$\Delta w_{0k}=\alpha\delta_k$

Step 7. Calculate error in unit Z_j, as well as

corrections to the weights and threshold,

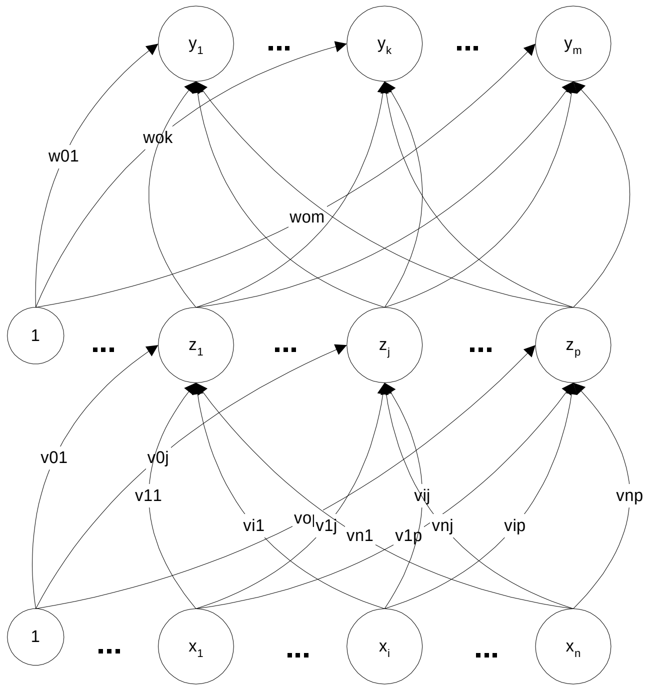
at the 1st layer

$\delta\_in_j= \sum_{k=1}^{m}\delta_k w_{jk}$

$\delta_j = \delta\_in_j.f'(z\_in_j)$

$\Delta v_{ij}=\alpha\delta_j x_i$

$\Delta v_{0j}=\alpha\delta_j$



NOVA IMS Information Management School

Backpropagation/multi-layer perceptron

- Algorithm (cont.)

```
(Update weights and slopes)
Step 8. For each $Y_k$ do:
        $w_{jk}$ (new) = $w_{jk}$ (old) + $\delta w_{jk}$, with k = 1, ..., m and j = 0, ..., p
    For each unit $Z_j$:
        $v_{ij}$ (new) = $v_{ij}$ (old) + $\delta v_{ij}$, with j = 1, ..., p and i = 0, ..., n
Step 9. Test stop condition
```

Backpropagation/multi-layer perceptron

- Number of Hidden Layers

  Although a hidden layer is sufficient to solve any problem of functional approximation, some solve more easily with 2 or more hidden layers.

  The algorithm is in all identical to that already studied. Thus, it is possible to generalize the delta rule for backpropagation with M layers, assuming the name generalized-delta-rule.

Backpropagation/multi-layer perceptron

- Selection of weights and initial bias

  The choice of weight values and bias synaptic links will influence the attainment of a global (or only local) error minimum and the speed of this convergence.

  - High value weights - Derivative values approach zero and the total stimulus in one unit will be small;
  - Low value weights - Total stimulus in one unit will be equally small.

  In either case the learning is very slow !!!

NOVA IMS Information Management School

Backpropagation/multi-layer perceptron

- Selection of weights and initial bias

  How to proceed?

  - Random assignment
  - Assign values to weights between -0.5 and +0.5

  - Nguyen-Widrow Initialization
    Based on geometric analysis of the response of hidden layer neurons to a simple input. This analysis is then extended to the situation of several inputs using the Fourier transforms.

Backpropagation/multi-layer perceptron

- Nguyen-Widrow Initialization

Where n = number of input units

p = number of hidden units

β = scale factor, the algorithm comes:

Step 1. Assign random values, between -0.5 and +0.5, to the weights: w_{kj}

Step 2. Calculate $\beta = 0.7\sqrt[n]{p}$

Step 3. Assign random values between - β and + β to the weights v_{j0} (bias)

Step 4. Assign random values between -0.5 e +0.5 to the weights v_{ji}

Step 5. Renormalize the weights v_{ji}

$$v_{ji}(novo) = \beta \frac{v_{ji}(velho)}{|v_j(velho)|}$$

Backpropagation/multi-layer perceptron

- And after training the network?

Step 1. For each input vector, execute Step 2-4.

Step 2. For i = 1, ..., n: activate the sensory units X_i;

Step 3. For j=1,…,p :

$$z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$$

$$z_j = f(z\_in_j)$$

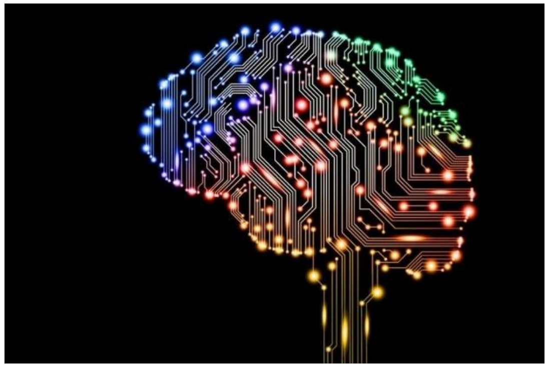
Step 4. For k=1,…,m :

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk}$$

$$y_k = f(y\_in_k)$$

Deep Learning

- What exactly is Deep Learning?
- Why is it generally better than other methods of image recognition, speech, and other data types?



"Deep Learning" means using an artificial neural network with multiple layers of nodes between input and output

The use of multiple layers between input and output allows for the identification and processing of multi-step characteristics, just as our brains do.

NOVA IMS Information Management School

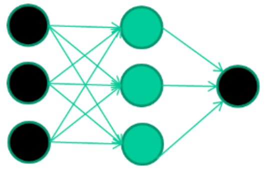
Deep Learning

- The neuronal network is composed of a cascade of many layers of nonlinear processing units for feature extraction and transformation.
- Each successive layer uses the output of the previous layer as input.
- Algorithms can be supervised or unsupervised and applications include pattern analysis (unsupervised) and classification (supervised).

Deep Learning

Right! but what's new? multi-layer neural networks have been around since the 1980s. What's really new?

there have always been good algorithms to learn weights in networks with 1 hidden layer...

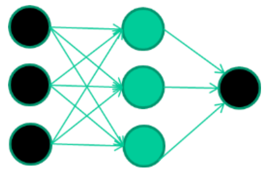


- For instance, multilayer perceptrons (MLP) represent the most general and powerful feedforward neural network model possible; They are arranged in layers so that all neurons within a layer receive input from all outputs of the previous layer at their input.

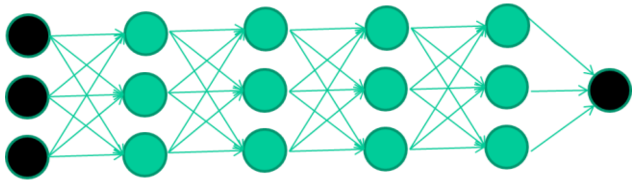
- This type of model has shown to be suitable for a certain type of problem with a fixed number of (more or less) unstructured parameters.

Deep Learning

Also these algorithms are mediocre at learning weights for networks with more than one hidden layer

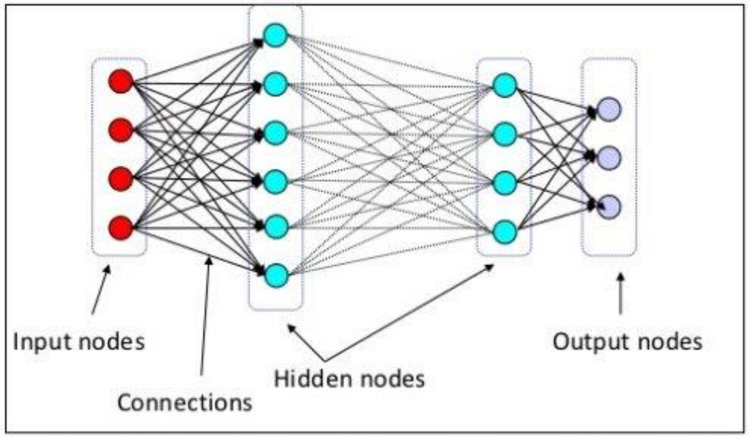


- What is New: finally we have competent algorithms to train networks with more than one hidden layers ☺ !



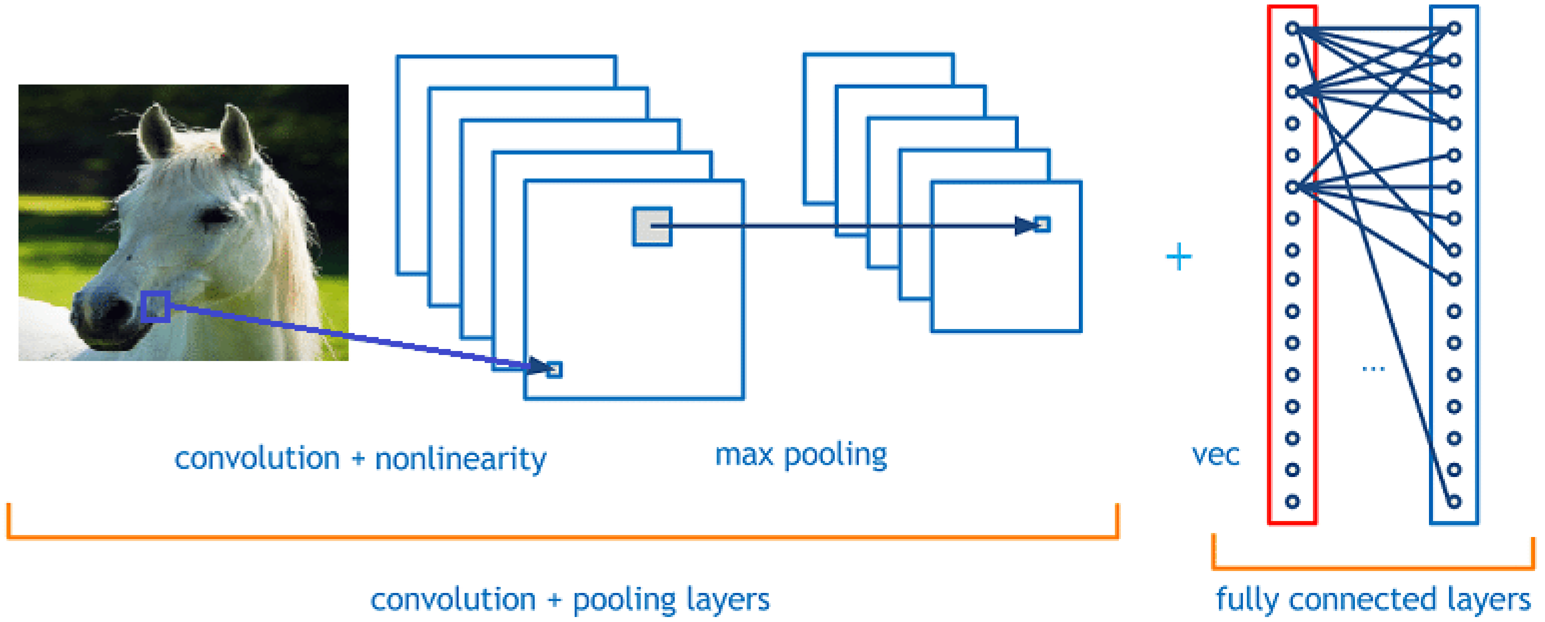
Deep Learning

- Are based on learning of multiple levels of resources or data representations. Top-level resources are derived from lower-level resources to form a hierarchical representation.

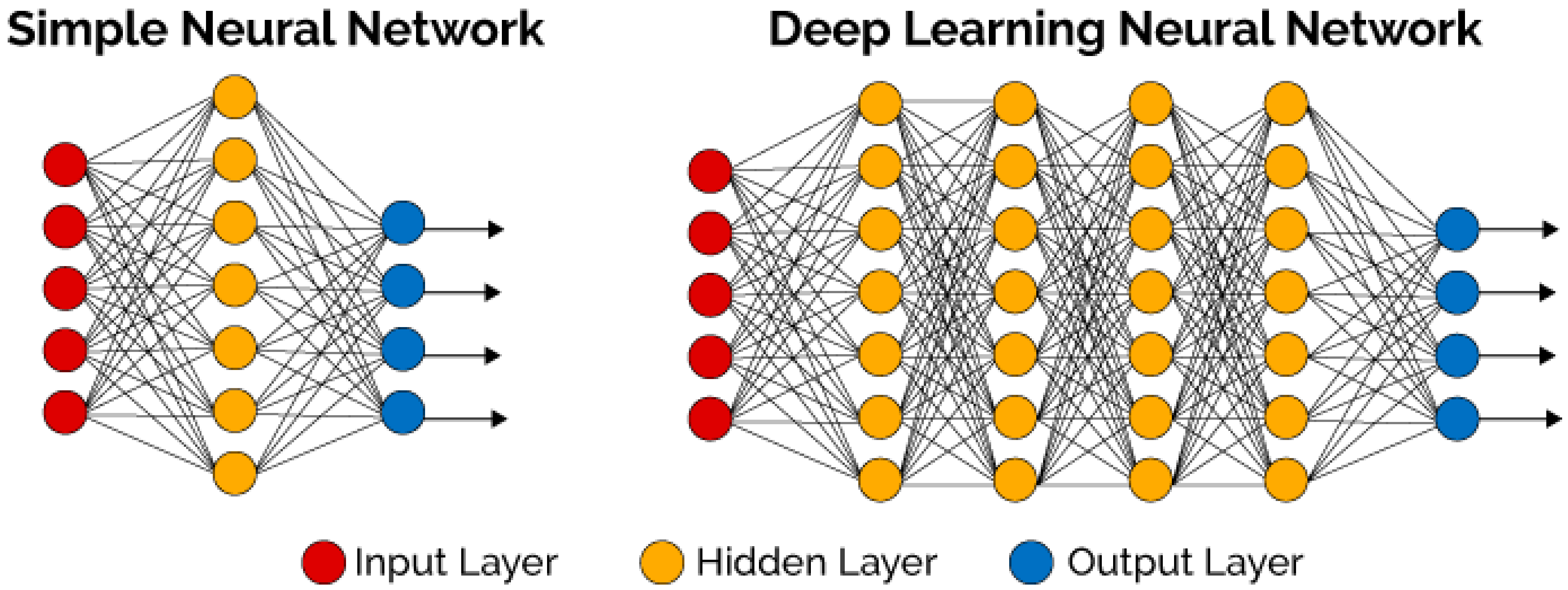


- Learn multiple levels of representations that correspond to different levels of abstraction => The levels form a hierarchy of concepts.

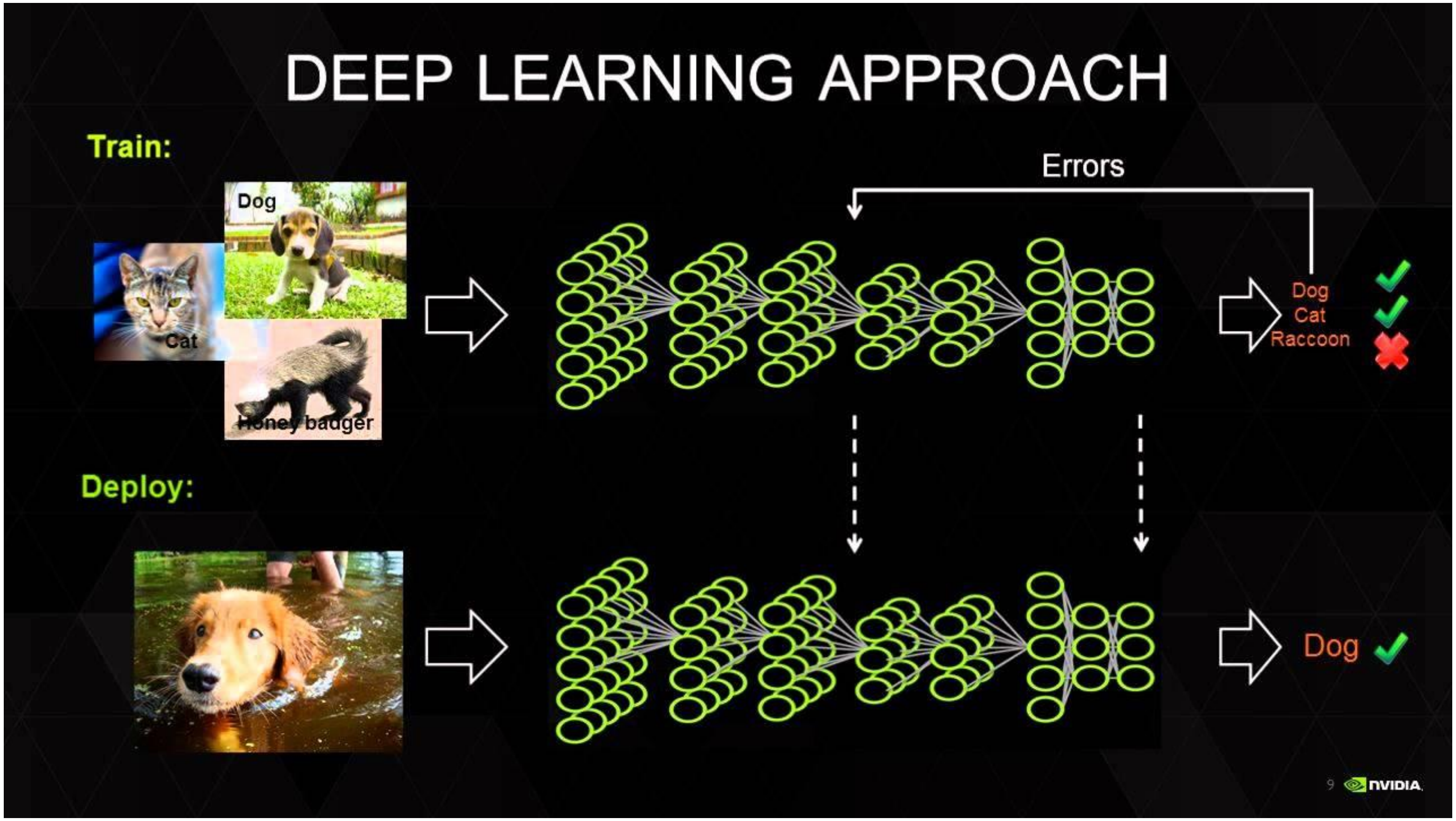
Convolution Algorithms



Deep Learning



Deep Learning



Deep Learning

However, if the number of parameters (weights) of this MLP models increases greatly - for example when receiving raw image data´

CIFAR-10, for example, contains 32 × 32 × 3 color images: if we want to treat each channel of each pixel as an independent input to an MLP, each neuron in the first hidden layer adds 3000 new parameters to the model!

The **CIFAR-10** dataset is a collection of images that are commonly used to train machine learning and computer vision algorithms

So, even before reaching the kind of images that people usually want to work with in real applications, the situation becomes unmanageable as the size of the images grows rapidly and becomes impossible to behave on such networks.

Toward Truly Intelligent Artificial Intelligences

- On of the most important research areas will be hybrid systems that combine the advantages of systems capable of reasoning on the basis of knowledge and memory use with those of AI based on the analysis of massive amounts of data (eg: deep learning).
- Today, deep-learning systems are significantly limited by what is designated as "catastrophic forgetting."
- It means that if they have been trained to carry out one task (playing Go, for example) and are then trained to do something different (distinguishing between images of dogs and cats, for example) they completely forget what they learned for the previous task (in this case, playing Go).
- <u>This limitation is powerful proof that those systems do not learn anything, at least in the human sense of learning</u>.

Toward Truly Intelligent Artificial Intelligences

- To have cognitive architectures (that integrate these components adequately. Integrated systems are a fundamental first step in someday achieving general AI.

- To design systems that combine perception, representation, reasoning, action, and learning. - those systems will have to be able to learn continuously throughout their existence - we still do not know how to integrate all of these components of intelligence.

- Strong Week AI – today we have a week Strong AI
- Week Strong AI – today we have a strong Week AI

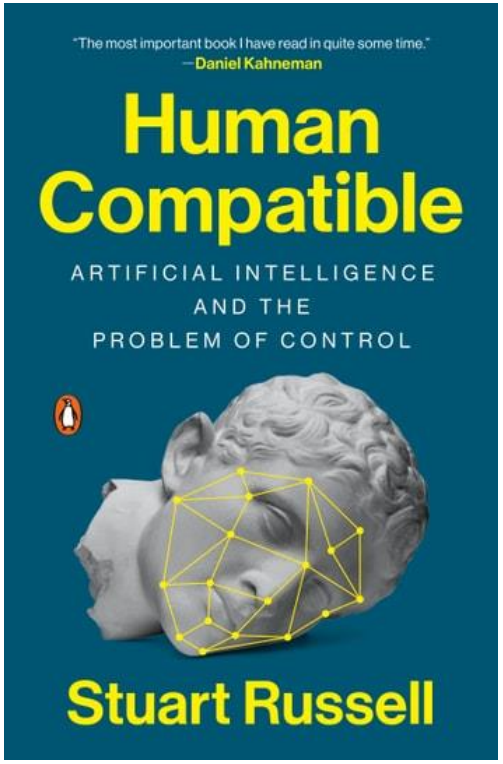
Question - Is the future conflict between humans and machines inevitable ?

If the predicted breakthroughs occur and superhuman AI emerges, we will have created entities far more powerful than ourselves.

How can we ensure they never, ever, have power over us? Can we ??

Russell suggests that we can rebuild AI on a new foundation, according to which machines are designed to be inherently uncertain about the human preferences they are required to satisfy.

Such machines would be humble, altruistic, and committed to pursue our objectives, not theirs.


"The most important book I have read in quite some time." —Daniel Kahneman
Human Compatible
ARTIFICIAL INTELLIGENCE AND THE PROBLEM OF CONTROL
Stuart Russell

NOVA IMS Information Management School

Discussion

Bibliography

Russell, Stuart, and Norvig, Peter. Artificial Intelligence: a Modern Approach, 4th Edition, Prentice Hall, 2020.

Buchanan, Bruce G. (Winter 2005), "A (Very) Brief History of Artificial Intelligence" (PDF), AI Magazine, pp. 53–60

Chitta Baral, "Knowledge Representation, Reasoning and Declarative Problem Solving", Cambridge University Press, 2003

"The Description Logic Handbook: Theory, Implementation and Applications, 2nd Edition", Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider, Cambridge University Press, 2007

Minsky, Marvin "The Society of Mind" Simon and Schuster

Costa, Ernesto & Simões, A. Inteligência Artificial. Fundamentos e Aplicações. Lisboa: FCA.

Bibliography (extra)

Gordon M. Shepherd, M.D. Phil. “Neurobiology”. Oxford University Press

Haykin, Simon. “Neural Network - A comprehensive foundation”. Macmillan Press

Fausett, Laura. “Fundamentals of Neural Networks”. Prentice Hall

Elaine Rich, Kevin Knight; Artificial intelligence. ISBN: 0-07-100894-2

Minky, Marvin L.; Papert, A. “Perceptrons”. MIT Press

Gurney, Kevin (1997). An introduction to neural networks. UCL Press. ISBN 978-1857286731.

Ripley, Brian D. (2007). Pattern Recognition and Neural Networks. Cambridge University Press. ISBN 978-0-521-71770-0.

Artificial Neural Networks

- A possible classification (adapted from Simon Haykin)
  - Simple Pattern Recognition Networks
    - Hebb Network, Perceptron, Adaline
  - Association of Standards
    - Heteroassociative, Self-associative, Bidirectional (BAM)
  - Competition-based networks
    - With fixed weights (Maxnet, Mexican Hat, Hamming Net)
    - Kohonen Self organization Maps (SOM)
  - Learning vectors Quantization
  - Counterpropagation
  - Adaptive resonance theory
    - ART1 and ART2
  - Backpropagation

Artificial Neural Networks

- Others
  - Fixed weight networks for restriction optimization
    - Boltzmann machine
    - Continuous Hopfield Net
    - Cauchy machine
    - Gaussian machine
  - Learning
    - Modified Hebbian Learning
    - Boltzmann Machine Learning
    - Simple recurrent network
    - Backpropagation in time
    - Backpropagation Training for Fully recurrent networks
  - Adaptive architectures
    - Probabilistic Neural Net
    - Cascade Correlation
  - Neocognitron
  - Deep Learning
  - ….

Deep Learning

Some important algorithms used in deep learning architectures:

1. Multilayer Perceptron Neural Network (MLPNN) - (image verification and reconstruction, Speech recognition, Machine translation, data classification, e-commerce, where many parameters are involved)
2. Backpropagation - (can be used in image and speech recognition, to improve the accuracy of predictions in data mining and machine learning, and in projects where derivatives must be calculated quickly)
3. Convolutional Neural Network (CNN) – (can be used with Image processing, recognition, and classification, Video recognition, Natural language-processing tasks, Pattern recognition, Recommendation engines, Medical image analysis)
4. Recurrent Neural Network (RNN) – (are useful for Sentiment classification, Image captioning, Speech recognition, Natural language processing, Machine translation, Search prediction, Video classification)
   1. Long Short-Term Memory (LSTM) - (ideal for Captioning of images and videos, Language translation and modelling, Sentiment analysis, Stock market predictions)
   2. Gated Recurrent Unit (**GRU**)
   3. Bidirectional Long-Short Term Memory (BLSTM).
5. Generative Adversarial Network (GAN) – (are useful for Cyber security, Health diagnostics, Natural language processing, Speech processing)
6. Restricted Boltzmann Machine (RBM) – (useful for Recommender systems, Filtering, Feature learning, Dimensionality reduction, Topic modelling)
7. Deep Belief Network (DBN) - (useful for Image and face recognition, Video-sequence recognition, Motion-capture data, Classifying high-resolution satellite image data)
   https://www.simplilearn.com/deep-learning-algorithms-article

NOVA IMS Information Management School

Convolution Algorithms

There are various architectures of CNNs available which have been key in building algorithms which power and shall power AI as a whole in the foreseeable future.

Some examples listed below:

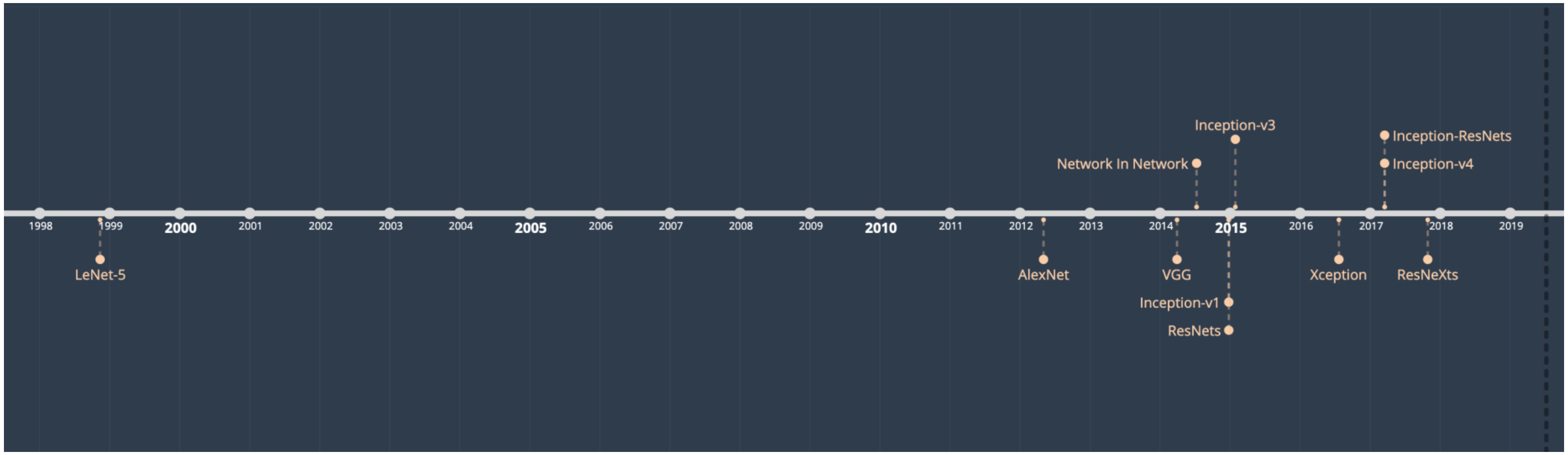
- LeNet-5 (1998)
- AlexNet (2012)
- ZFNet (2013 - modified version of AlexNet)
- VGG-16 (2014)
- GoogLeNet (Inception-v1) (2014)
- Inception-v3 (2015)
- ResNet-50 (2015)
- Xception (2016)
- Inception-v4 (2016)
- Inception-ResNets (2016)
- ResNeXt-50 (2017)
- ….

Alex Krizhevsky,

ResNet = Residual Networks)

https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

Convolution Algorithms


LeNet-5
AlexNet
Network In Network
Inception-v3
VGG
Inception-v1
ResNets
Xception
Inception-ResNets
Inception-v4
ResNeXts