UPPSALA
UNIVERSITET

Natural Computation Methods in Machine Learning (NCML)

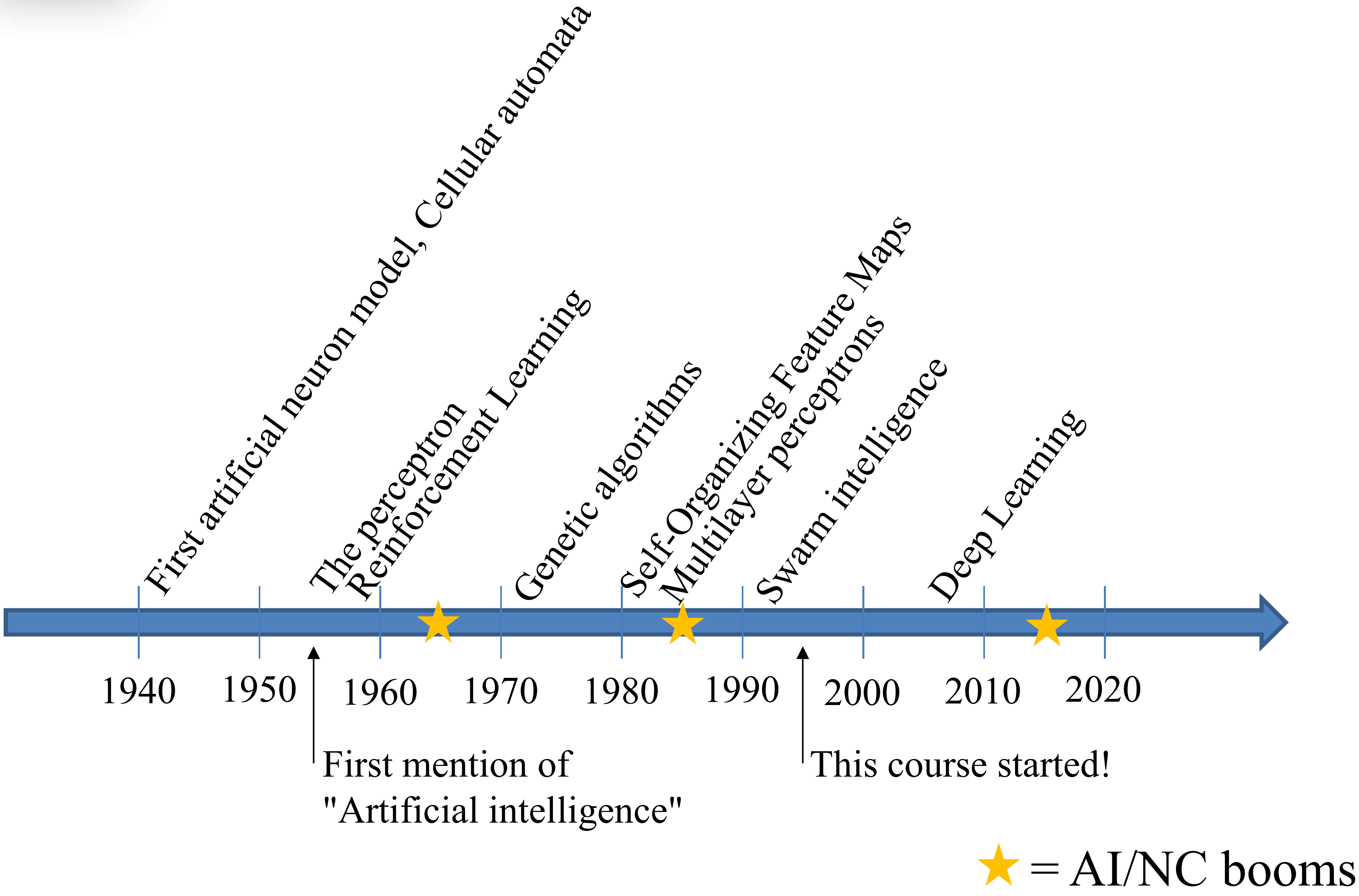
Lecture 2: Introduction to Artificial Neural Networks (ANNs)

UPPSALA UNIVERSITET

What is Artificial Intelligence?

- Modelling human intelligence?
  - No, not really (at least not in computer science)
- Automating tasks which we (previously) thought required intelligence
  - When done, this does not mean that the computer has become intelligent,
  - it more likely shows that the task did not require it
  - The solution is therefore, in hindsight, often no longer considered AI
  - Looking back, it therefore seems as if AI research never lead anywhere!
  - AI is a moving target!

Natural computation history



Computers and humans

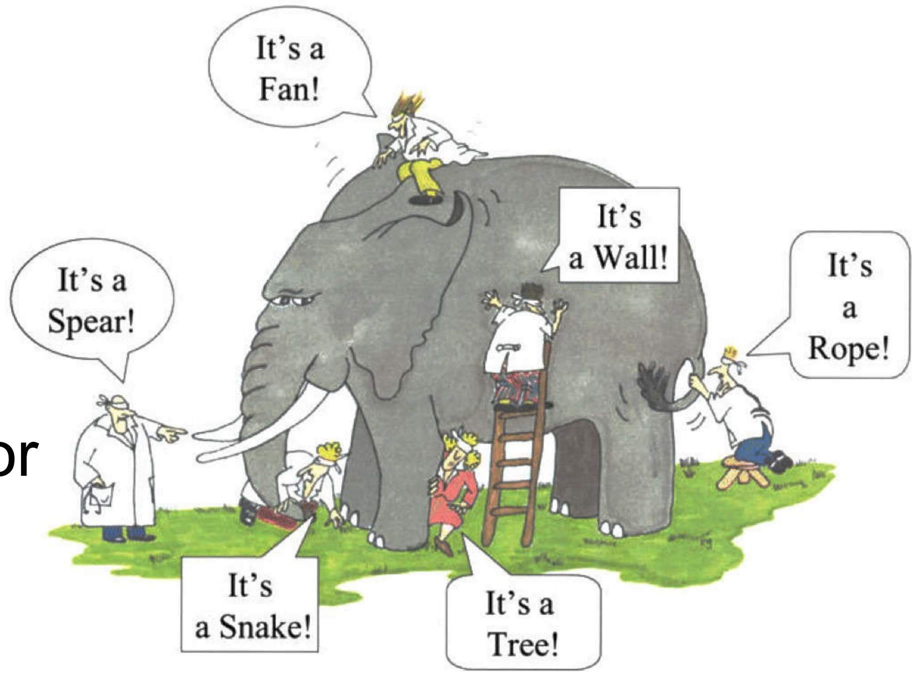
- Generalization
- Dealing with missing information
- Focusing on what is important
- Fault tolerance

Natural Computation

- If we want to solve problems for which we know nature/evolution has found solutions, it should be worth-while studying those solutions
- But don't take modelling too far!
  - It's a tool, not the goal
  - An airplane with flapping wings may be a better model (of birds), but that's not what aircraft engineers strive for. Usually ...

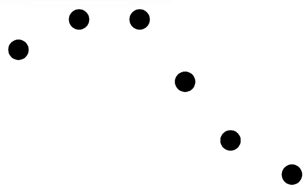
Artificial Neural Networks

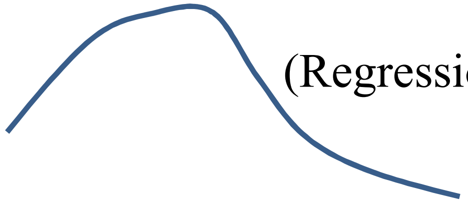
- Interdisciplinary field
  - Many viewpoints
  - The same object may look very different from different angles (subjects)
  - Many different names for the same concepts
  - Sometimes difficult to search online for information (too many alternative keywords)



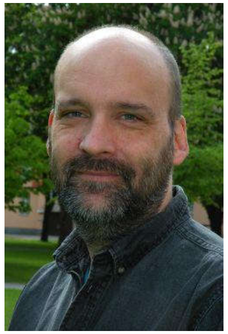
Neural networks

are universal function approximators





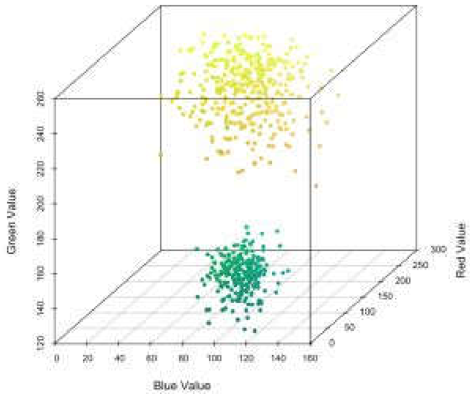
(Regression)



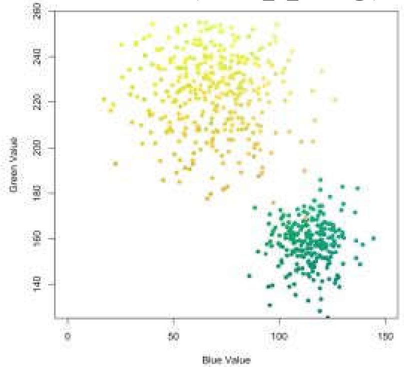
x_1, x_2, . . . , x_n → **Artificial Neural Network** → y_1, y_2, …, y_m

Olle (Classification)

(Mapping)



$$\bar{y} = f(\bar{x})$$



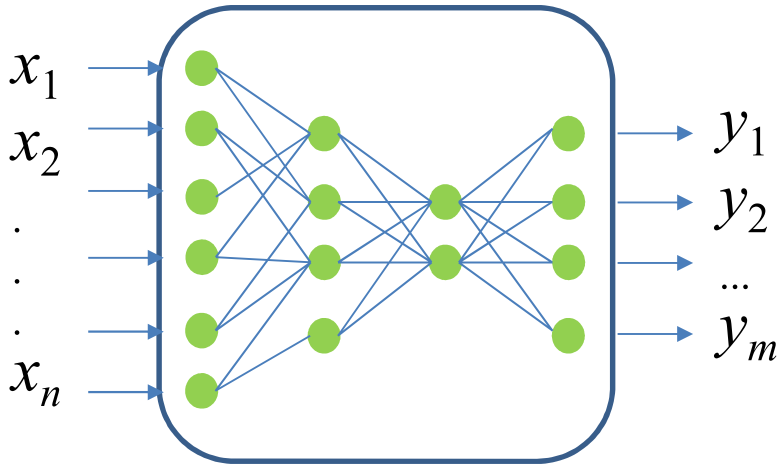
Can approximate any function to any degree of accuracy

Not only for classification

- Some examples of applications which are not classification
  - Continuous control
    - Self-driving cars, for example, where the output (gas pedal and steering wheel) is continuous, even if there are many classification tasks to be solved as well
  - Value prediction
    - Stock market, weather, effects of climate change, probability that a loan applicant will repay the debt, ...
  - Extracting information from noisy data
  - Image analysis (not recognition – that's classification)
    - Compressing/decompressing, segmentation, enhancement, find and mark objects, ...

Neural networks

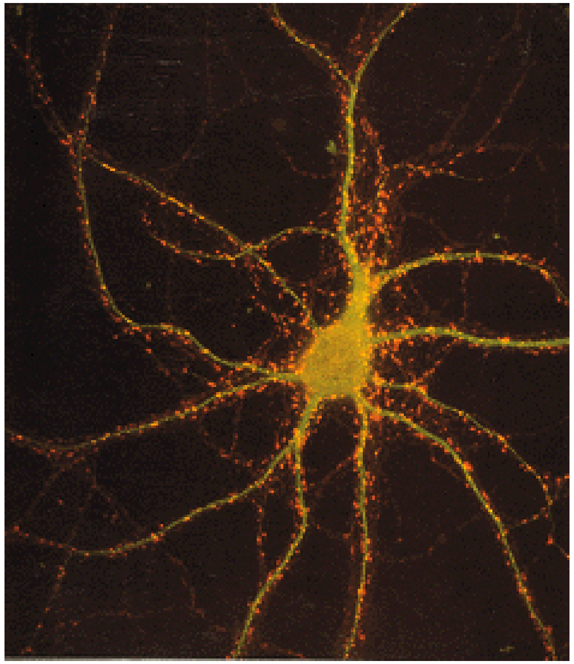
Biologically inspired, massively parallel networks of simple, adaptive, communicating nodes ('neurons')

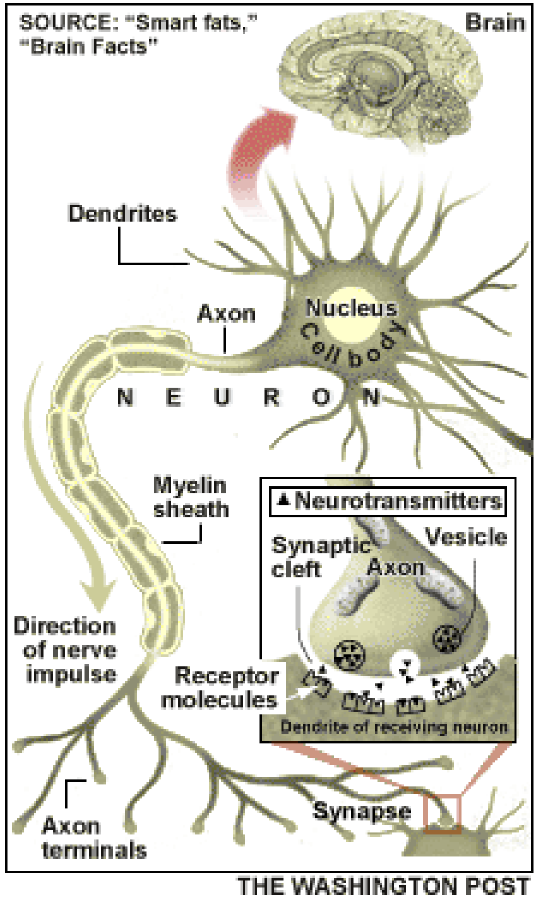


$$\bar{y} = f(\bar{x})$$

(not necessarily 'layered' like this, but most often is)

Neurons from biology

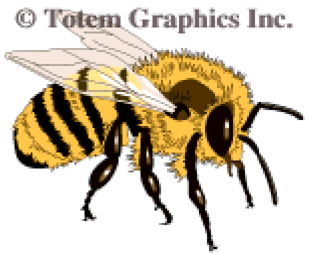




The human body contains about 100 biljon neurons
(≈ number of water drops required to fill an Olympic swimming pool)

UPPSALA UNIVERSITET

The honeybee



- 1 miljon brain cells in a few mm^3
- Approximately 10 TFLOP at 10 μW

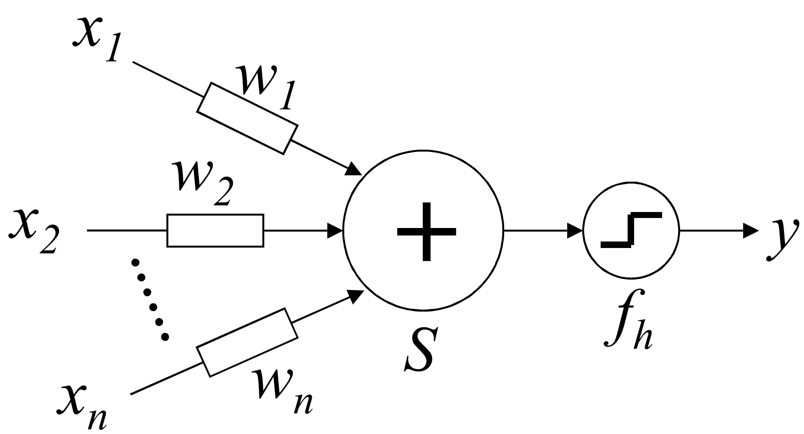
With this computational device, it can

- see, smell, recognize
- walk, fly, maintain balance, navigate
- communicate, find and collect food
- … and more

Autonomously!

The first artificial neuron model

McCulloch & Pitts 1943



$$y = f_h(S)$$

Activation function (here a hard limiter)

$$f_h(S) = \begin{cases} 1, \text{if } S > 0 \\ 0, \text{if } S \leq 0 \end{cases}$$

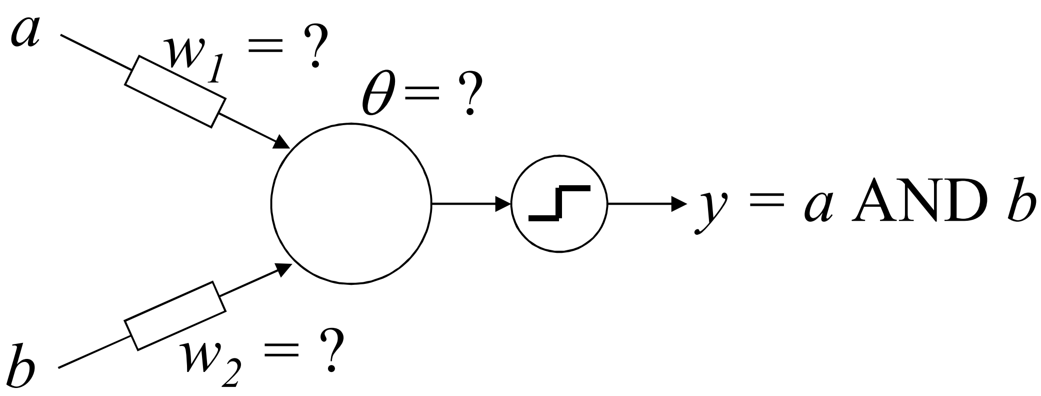
$$S = \sum_{i=1}^{n} w_i x_i - \theta$$

threshold or bias

<u>Very</u> crude biological model, but still useful for computations

Binary neurons = logic gates

Boolean logic



AND

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

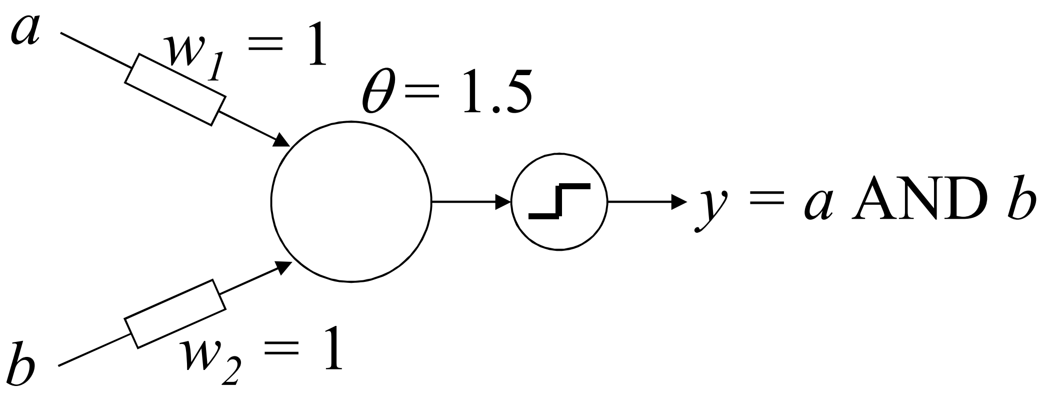
$$y = f_h(S)$$

$$f_h(S) = \begin{cases} 1, \text{if } S > 0 \\ 0, \text{if } S \leq 0 \end{cases}$$

$$S = w_1 a + w_2 b - \theta$$

Binary neurons = logic gates

Boolean logic



AND

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

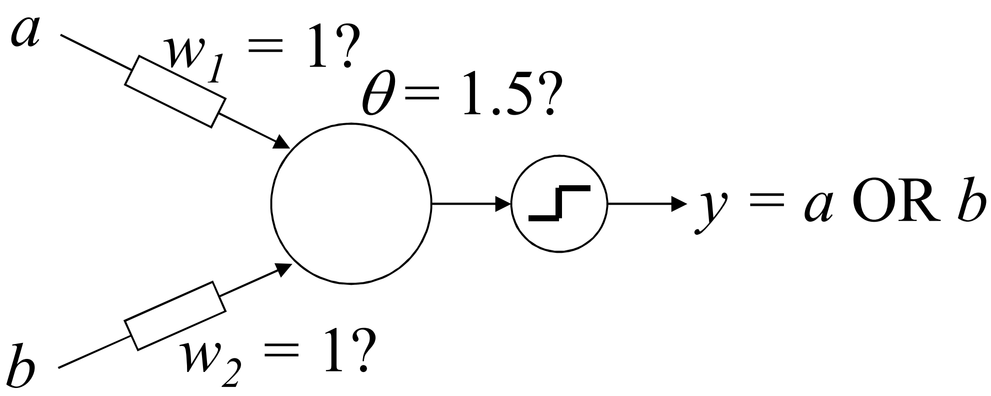
$$y = f_h(S)$$

$$f_h(S) = \begin{cases} 1, \text{if } S > 0 \\ 0, \text{if } S \leq 0 \end{cases}$$

$$S = w_1 a + w_2 b - \theta$$

Binary neurons = logic gates

Boolean logic



OR

a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

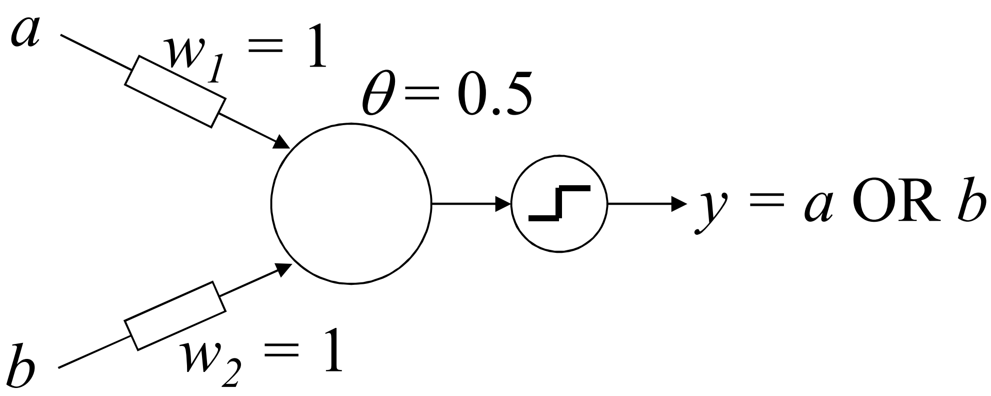
$$y = f_h(S)$$

$$f_h(S) = \begin{cases} 1, \text{if } S > 0 \\ 0, \text{if } S \leq 0 \end{cases}$$

$$S = w_1 a + w_2 b - \theta$$

Binary neurons = logic gates

Boolean logic



OR

a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

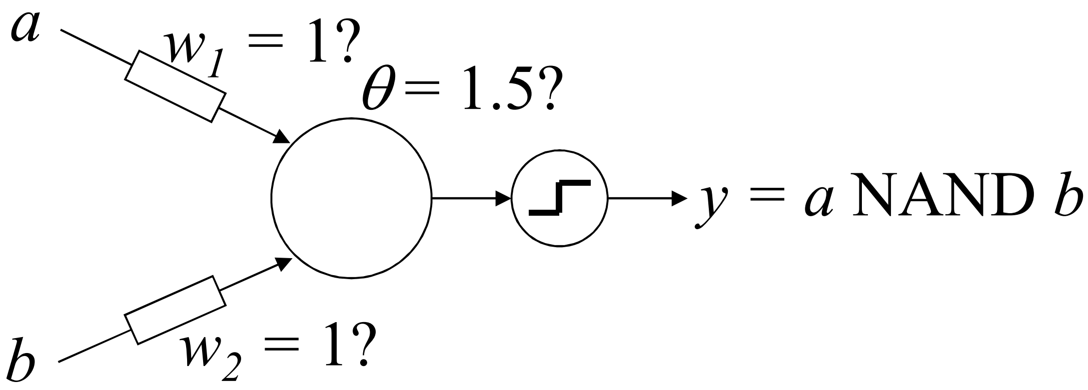
$$y = f_h(S)$$

$$f_h(S) = \begin{cases} 1, \text{if S} > 0 \\ 0, \text{if S} \leq 0 \end{cases}$$

$$S = w_1 a + w_2 b - \theta$$

Binary neurons = logic gates

Boolean logic



NAND

a	b	y
0	0	1
0	1	1
1	0	1
1	1	0

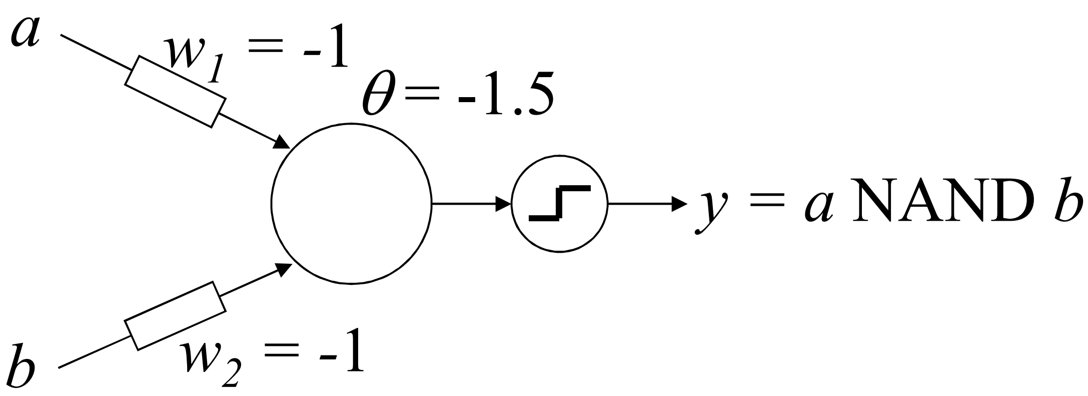
$$y = f_h(S)$$

$$f_h(S) = \begin{cases} 1, \text{if } S > 0 \\ 0, \text{if } S \leq 0 \end{cases}$$

$$S = w_1 a + w_2 b - \theta$$

Binary neurons = logic gates

Boolean logic



NAND

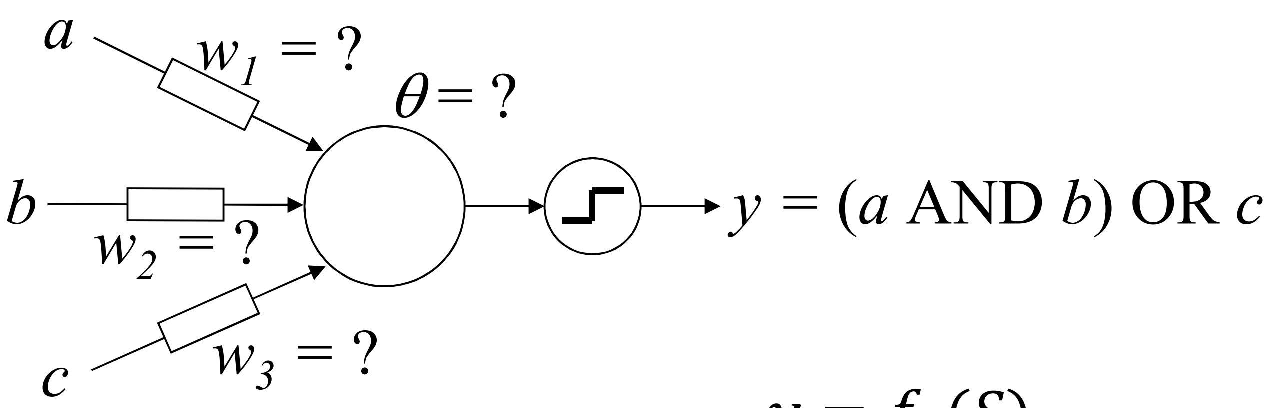
a	b	y
0	0	1
0	1	1
1	0	1
1	1	0

NAND is functionally complete – any Boolean function can be reformulated using NANDs only!

➔ A network of binary neurons is Turing complete

Challenge

- Set the weights of a binary neuron, so that it implements (*a* AND *b*) OR *c*



$$y = f_h(S)$$

$$f_h(S) = \begin{cases} 0, \text{ if } S \leq 0 \\ 1, \text{ if } S > 0 \end{cases}$$

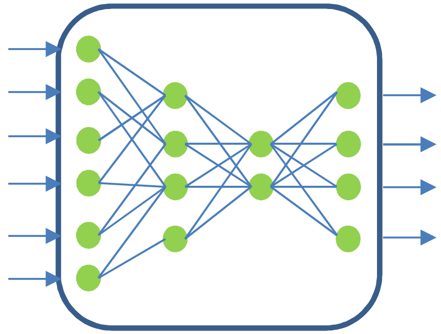
$$S = w_1 a + w_2 b + w_3 c - \theta$$

a	b	c	y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

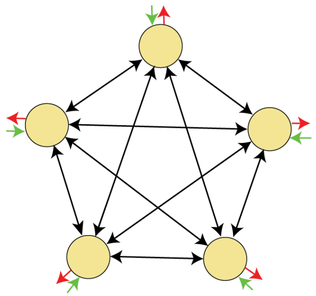
UPPSALA UNIVERSITET

Common properties of ANNs

- ANNs store information in the connections (weights), not in the nodes
- ANNs are fault tolerant
- ANNs are trained (by modifying the weights), not programmed
- ANNs can generalize – work in situations slightly different than before (without retraining)
- ANNs are adaptive – can be retrained if generalization does not suffice
- ANNs are concurrent



Learning

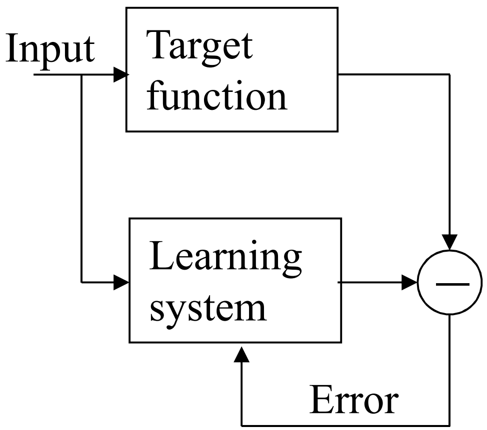


- Hebb's rule (1949)
  - Memory model for (biological) neural networks
  - If two nodes are active simultaneously, reinforce the connection between them
- Rosenblatt's Perceptron Convergence Procedure (PCP, 1958)
  - Started the 1960's boom
  - Next lecture

Three forms of learning

(you tried all three in the intro-lab)

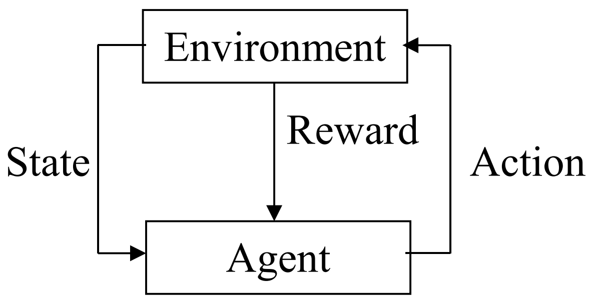
Supervised



Imitation

PCP, Backprop, Rprop, ...

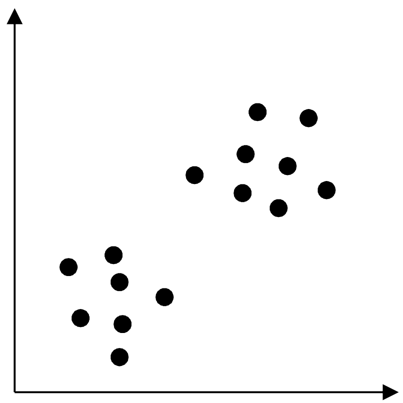
Reinforcement



Trial-and-error

TD(λ), Sarsa, Q-Learning, ...

Unsupervised

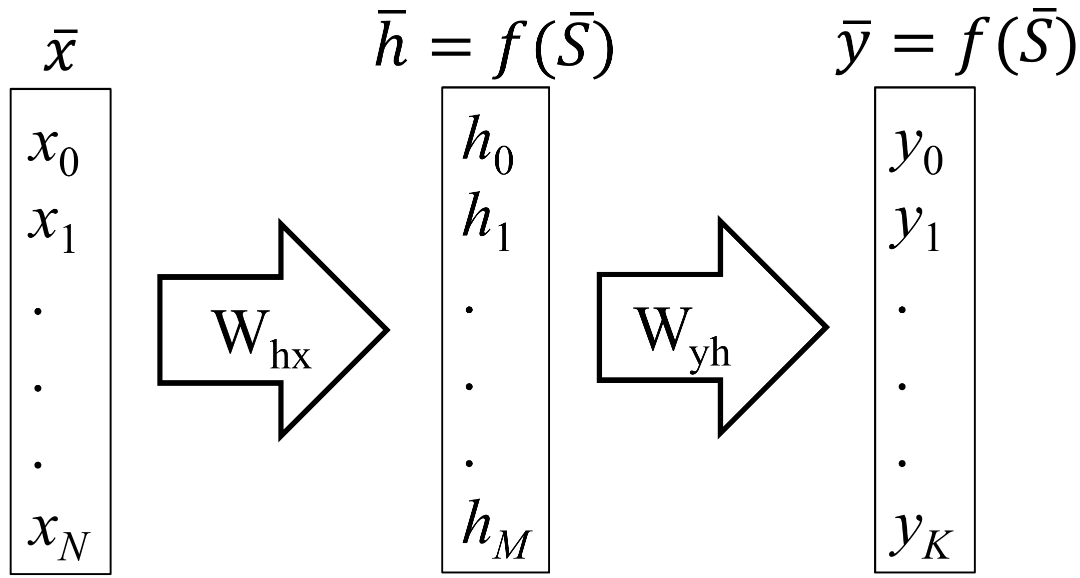


Structure

Hebb, SCL, K-Means, SOFM

Network architectures

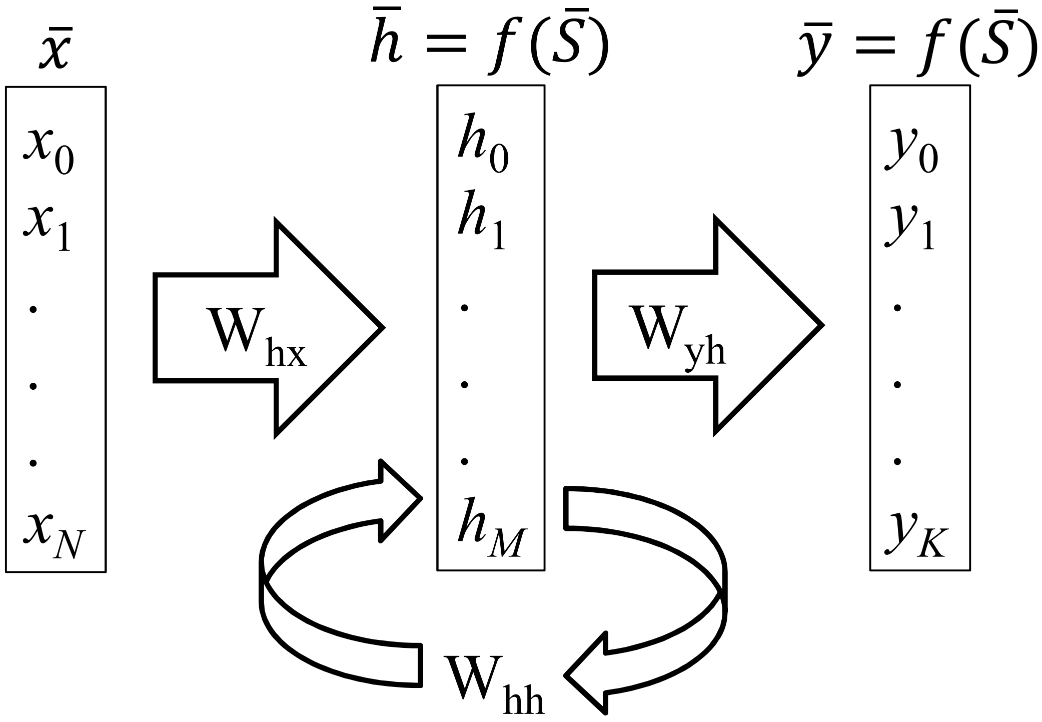
Feed-forward – information flows in one direction



- Used for classification, function approximation (regression), perception
- e.g. Multilayer perceptrons, CNNs

Network architectures

Recurrent – layered networks with loops

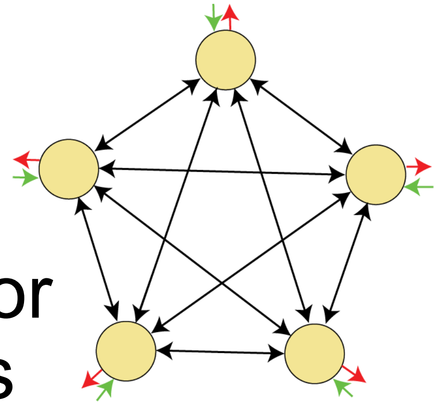


- Used for recognizing/generating sequences
- e.g. Jordan networks, Elman networks, LSTMs

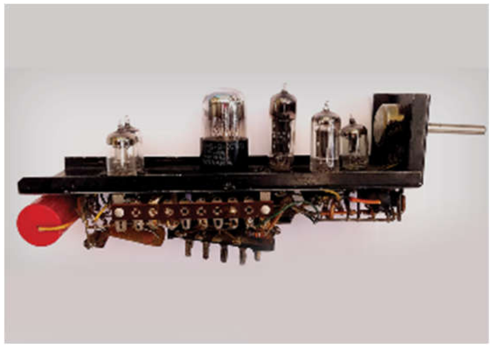
(LSTM layers are very different though)

Network architectures

Fully interconnected recurrent



- Finite state machine!
- Used as associative memories, or for combinatorial optimization problems
- Training – often some extension of Hebb's rule
- The first neurocomputer, SNARC (1951) was based on this



A SNARC neuron

UPPSALA UNIVERSITET

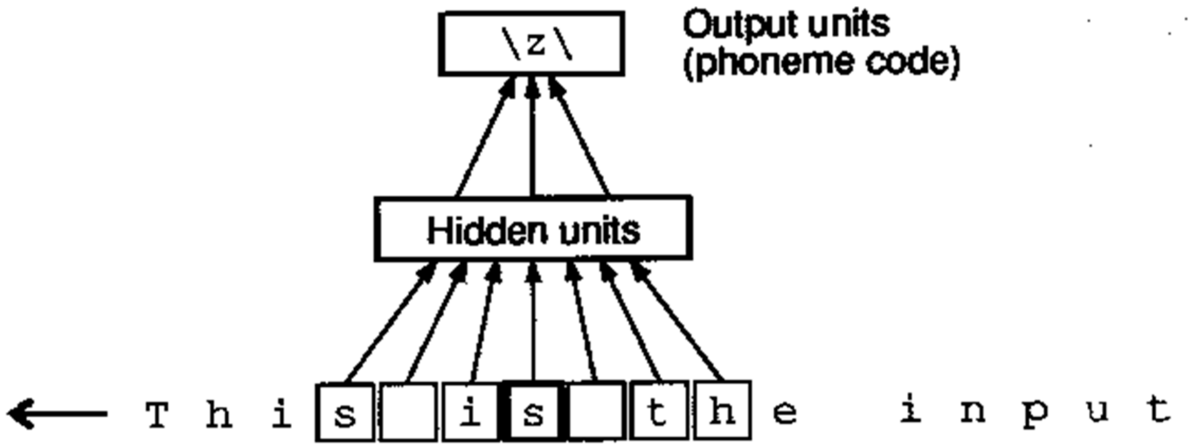
Why neural networks?

- Why not use statistical methods?
- Neural networks are statistical methods! (not as "model free" as sometimes claimed)
- Today, neural networks excel in many application areas, but they have been used for a long time for other reasons as well:
  – Rapid response (feed-forward networks)
    - hardware
    - trade-off training/recall
  – Rapid prototyping, e.g. NetTalk

NetTalk

Sejnowski & Rosenberg, 1987

- Learns to pronounce written text
  - Produces phonemes



  - Did not beat state-of-the-art at the time, but close, and at a very small fraction of R&D time

ANN application classics

Tool boxes, adaptive filters, OCR, Image analysis, signal processing, touch pads, fraud detection, recommendation systems, ...