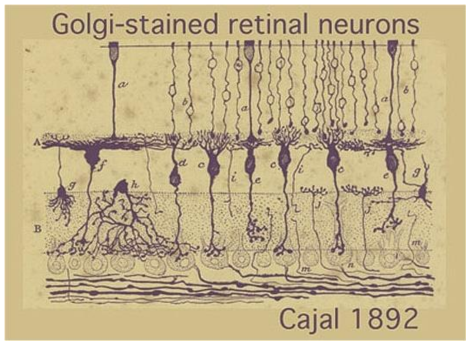
EEEN4/60151: Machine Learning & Opti...

Hujun Yin

Part 6: Neural Networks



Cross-section of retina, drawn by Santiago Ramón y Cajal (1853-1934)

1

Part 6: Neural Networks

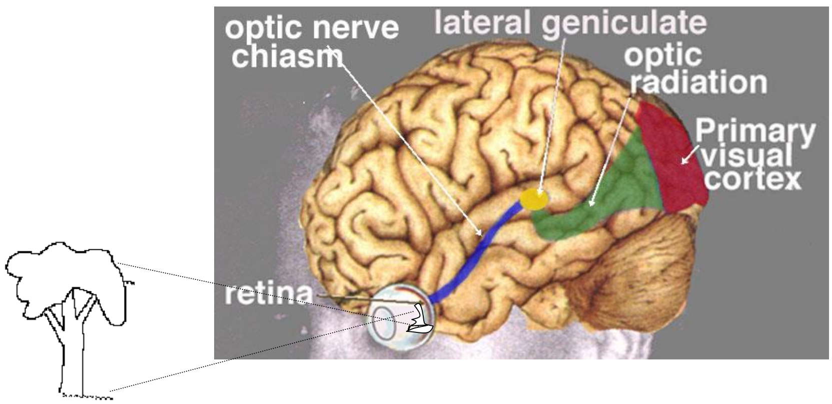
- Perception & cognition*
  - Retina & visual pathway
  - Early artificial intelligence
- Neurons & models
- Learning paradigms & network architectures
- Multilayer perceptrons
- Self-organising networks
- Learning & generalisation

* For knowledge only

2

6.1 Perception & cognition

- Perception & cognition

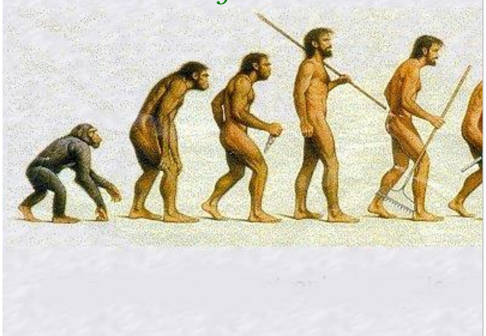


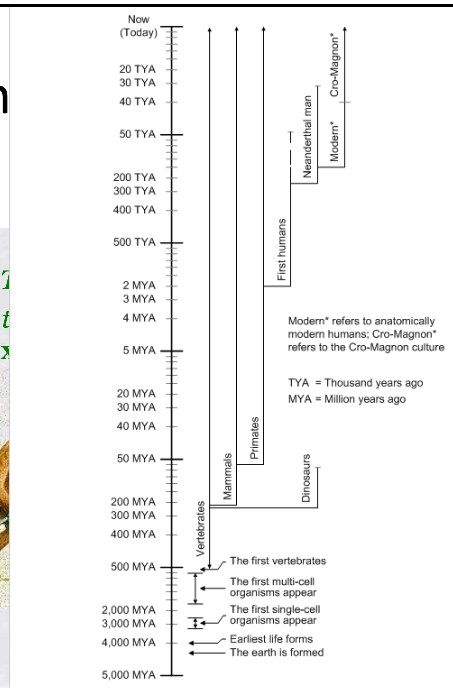
3

6.1 Perception

- Evolution

"Evolution is fascinating to watch. T
interesting when one can observe t
single man." **Shana Alex**



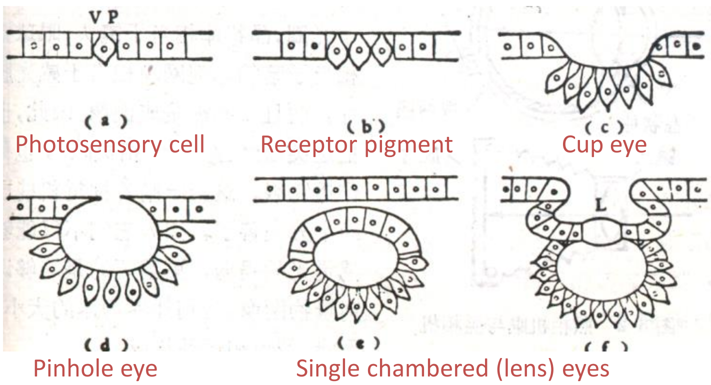


High-level view of evolution in general from a human-centric viewpoint

4

6.1 Perception & cognition

- Evolution of eyes

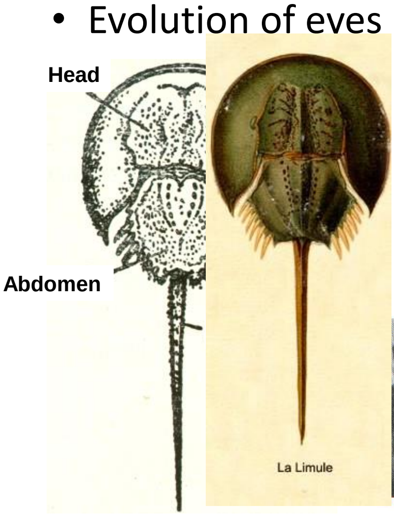


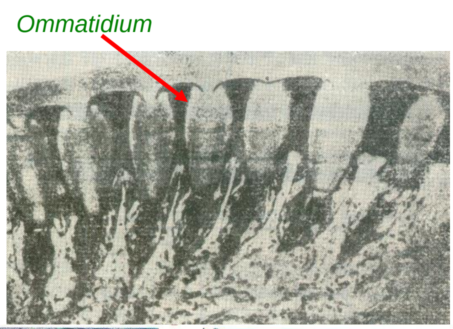
Compound eyes have good time resolving power. Human eyes need 0.05 second to identify objects, while compounds eyes need only 0.01 second.

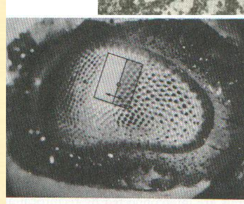
5

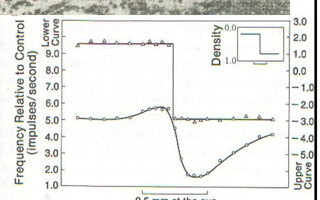
6.1 Perception & cognition

- Evolution of eyes







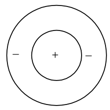


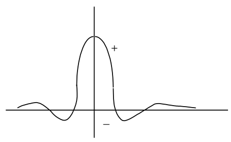
Hartline, et al (1960s) verified receptive field and lateral inhibition in limulus eyes

6

6.1 Perception & cognition

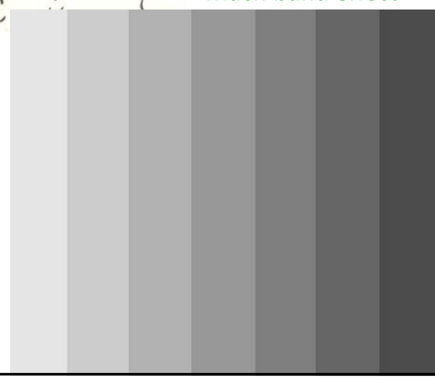
- Eyes



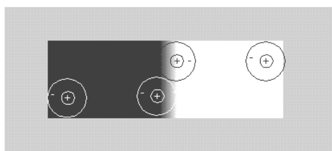


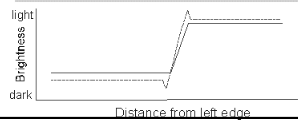
To extract object boundaries!

Mach band effect



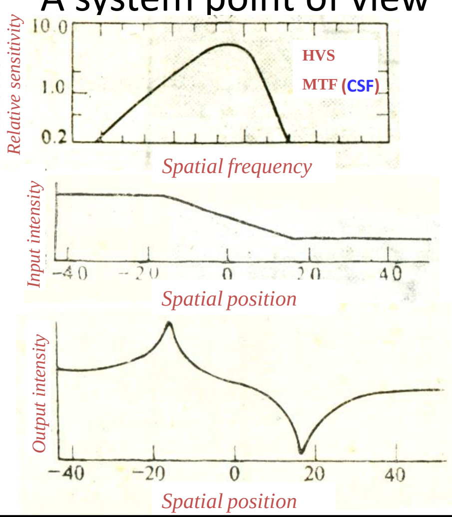
Lateral inhibition

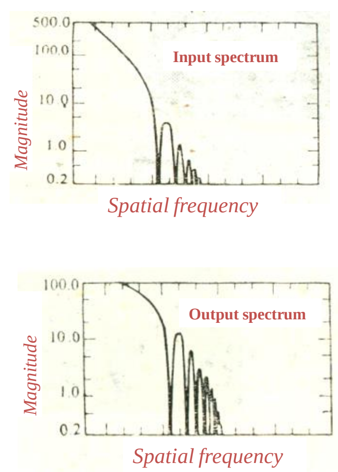




6.1 Perception & cognition

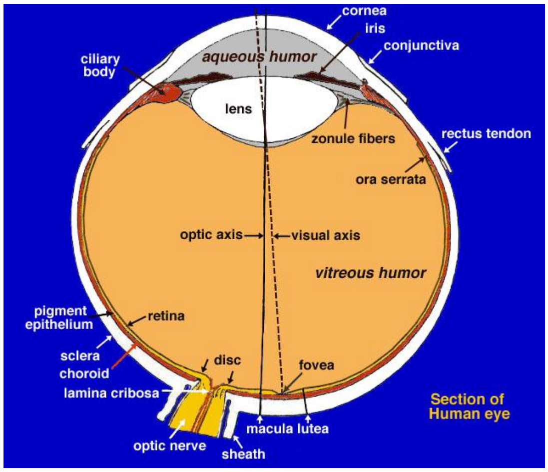
- A system point of view





6.1 Perception & cognition

- Human eye



Cones: 6-7 millions

Red (64%)

Green (32%)

Blue (2%)

Rods: 120 millions

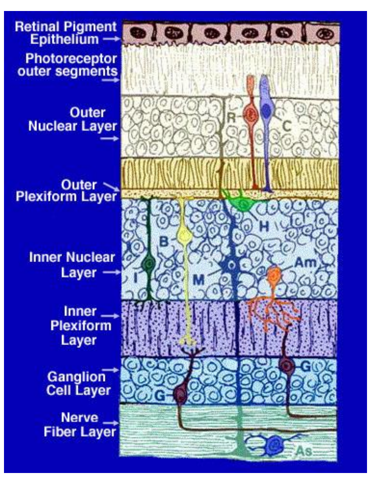
Dynamic range (brightest to darkest)***:*** 20 billion times (i.e. 206 dB)

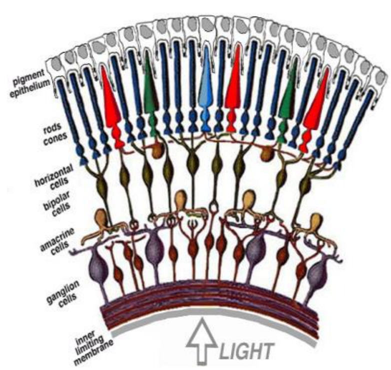
Cameras: 8-16 bits

9

6.1 Perception & cognition

- Retina

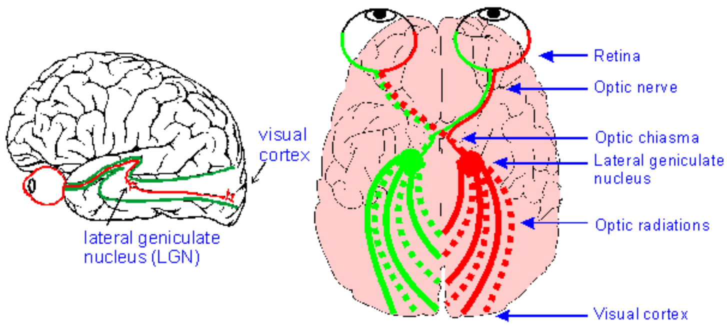




10

6.1 Perception & cognition

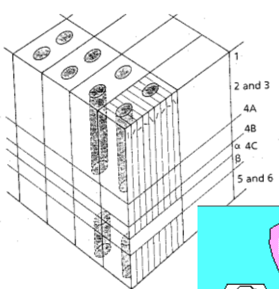
- Visual pathway

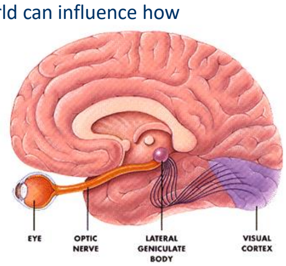


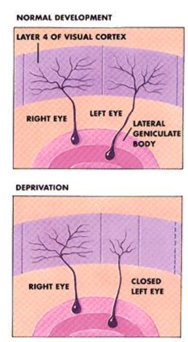
There are about 100 millions of photosensitive cells in human retina,
but only 1 million optic nerves connecting between retina and cortex.

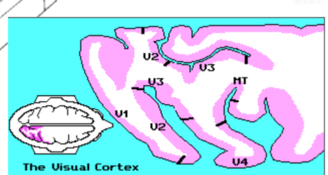
6.1 Perception & cognition

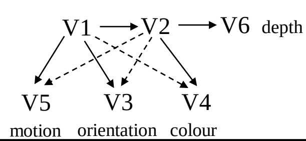
- Visual cortex
  - Many regions have predetermined responses to visual stimuli.
  - Sensory experience from the external world can influence how the brain wires itself up after birth.





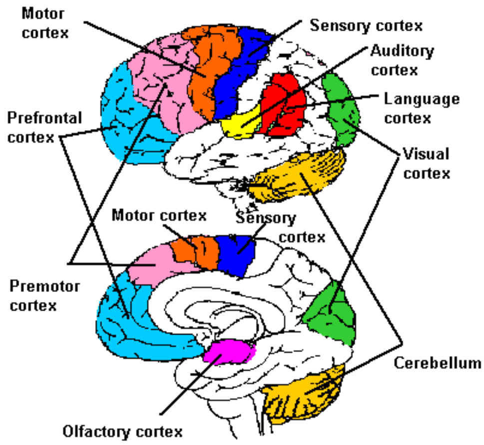






6.1 Perception & cognition

- Cortex



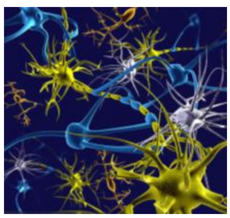
13

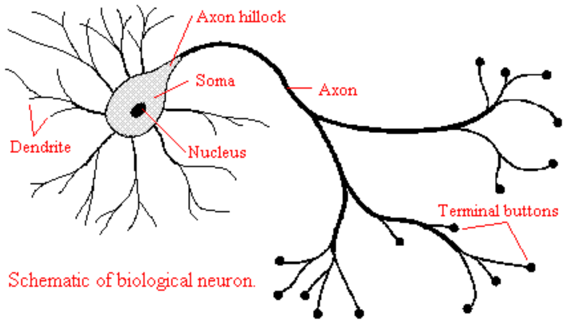
MANCHESTER 1824
The University of Manchester

Neuroscience Stories

Brain

Neurons and their functions





Schematic of biological neuron.

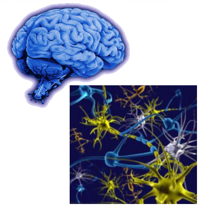
The brain has ~100 billion interconnected neurons, each connecting to 1000~10000 neighbouring neurons. These connections are called synapses.

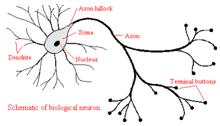
14

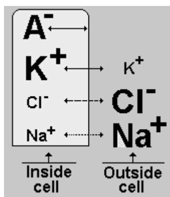
Neuroscience Stories

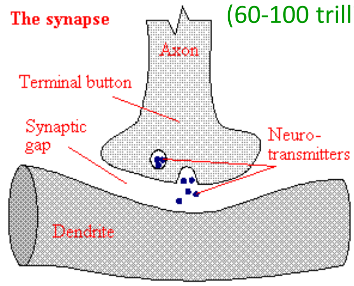
Brain

Neurons and their functions

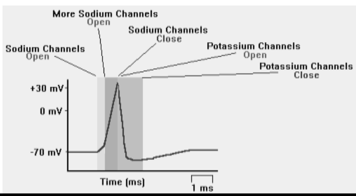








(60-100 trillion) synapses



Axon's action potential opens ion channel, producing Ca^{2+} that leads to release of neurotransmitter, causing ion-conducting channels to open. Synapses can have either excitatory (depolarizing) or inhibitory (hyperpolarizing) effect on postsynaptic neurons.

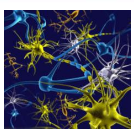
15

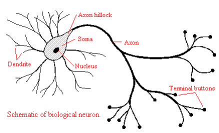
Neuroscience Stories

Cortex

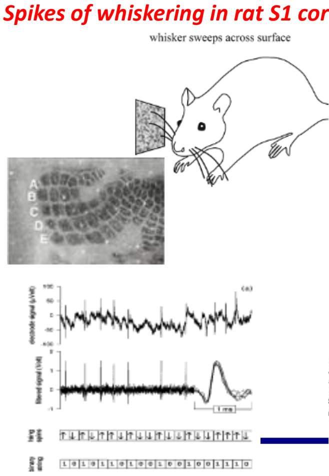
Neurons and their functions

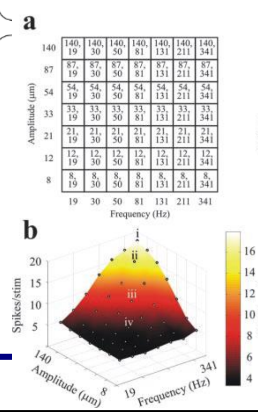
An example



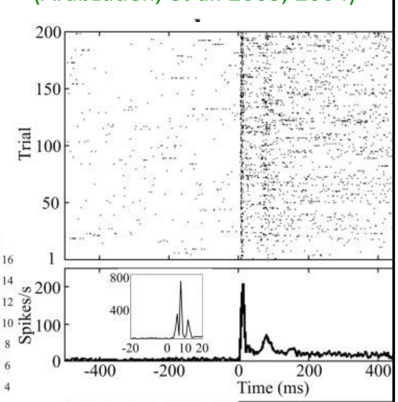


Spikes of whiskering in rat S1 cortex





(Arabzadeh, et al. 2003, 2004)



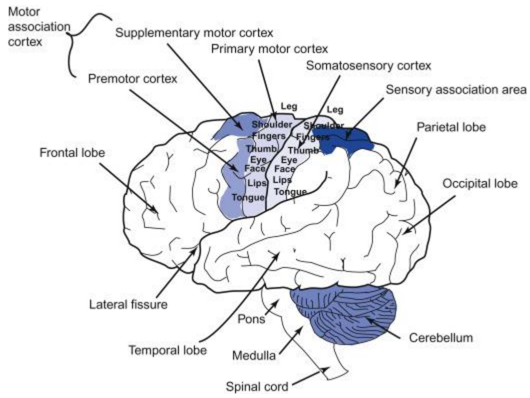
16

MANCHESTER 1824

The University of Manchester

Neuroscience Stories

Cortex

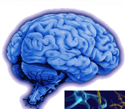


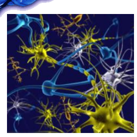
Courtesy of ScienceDirect

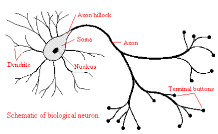
17

MANCHESTER 1824

The University of Manchester







Neuroscience Stories

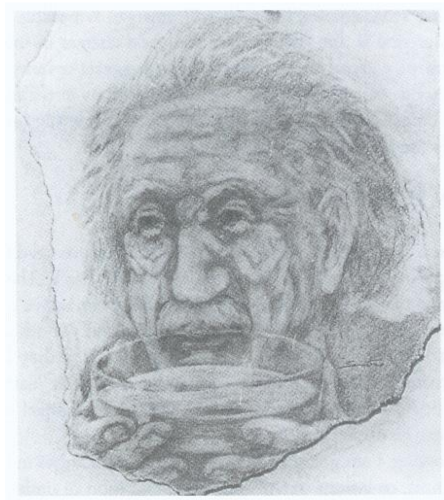
Learning and memory

- ***Neurobiology***: Learning is either formation of receptor evoked (neuronal) connections or strengthening an existing connection given an event", while "Memory is a process by which that knowledge of the world is encoded, stored and later retrieved. (*Eric Kandel*)

18

6.1 Perception & cognition

- Perception



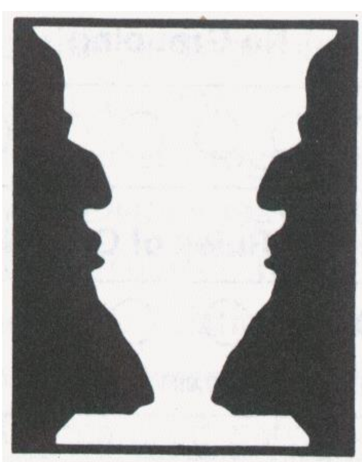
Einstein and ??

- from Kelso (1997)

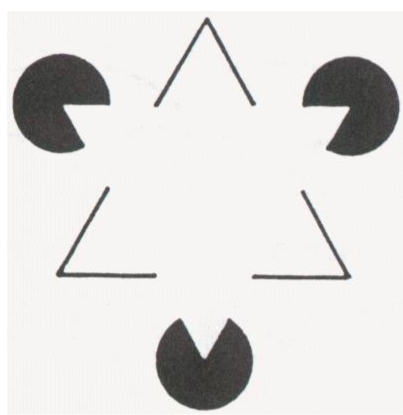
19

6.1 Perception & cognition

- Perception



Robin's vase

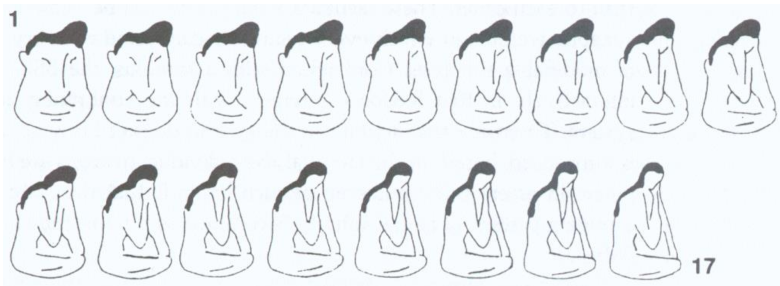


Kanisza triangle

20

6.1 Perception & cognition

- Cognition



Cognition is a result of dynamic pattern forming process of the brain, in which activities in the brain and behaviours appear self-organising into coherent statues or patterns and self-organisation provide a paradigm for behaviour and cognition, as well as the structure and function of nervous system - *from Kelso (1997)*

21

6.1 Perception & cognition

- Cognition

The human brain has trillions of cells, large enough for holding all events we have ever experienced [as claimed], but why can't we remember all of them in details? Pinker believes that our brains are compelled to organise, without categories, the mental lift would be a chaos. Categorisations are for inference of objects having the same or similar properties.- *from Pinker (1997)*

22

6.1 Perception & cognition

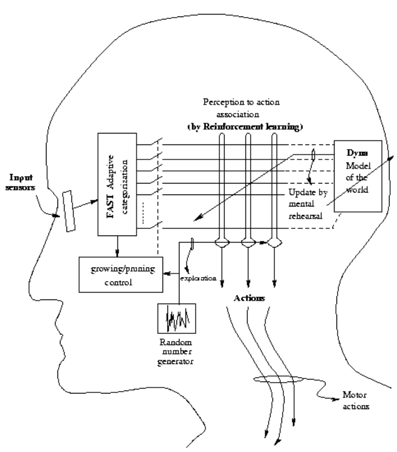
- Cognition

Psychology
Behaviourism
Neurobiology
Cognitive Science
Computationism
Connectionism
……

23

6.1 Perception & cognition

- Artificial system

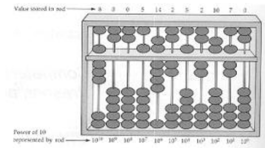


Cognition is a complex information processing process
(David Marr, 1945-1980)

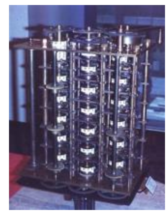
24

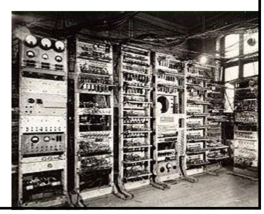
6.1 Perception & cognition

- Computing history



China		ancient Abaci
Blaise Pascale	1642	Pascaline or Adding Machine
Joseph Jacquard	1804	Punch card controlled loom
Charles Babbage	1823	Difference Engine
Augusta Ada	1836	Programming
Herman Hollerith	1880	Data Processing Devices (IBM)
UK team	1942	Colossus
USA team	1946	ENIAC
UK teams	1949	EDSAC and Manchester Mark I





25

6.1 Perception & cognition

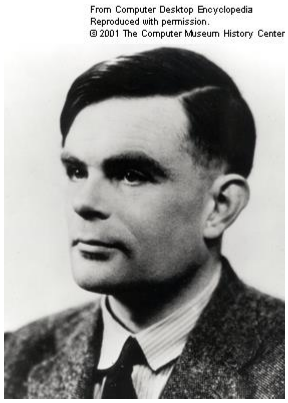
- Artificial intelligence

Alan Turing: Turing Machine,
Universal Turing Machine
Turing Test (Can Machines Think?)

Top-down: Representation/Symbols
Reasoning/Rules/Grammars
Learning/Logical Operations

26

Turing test

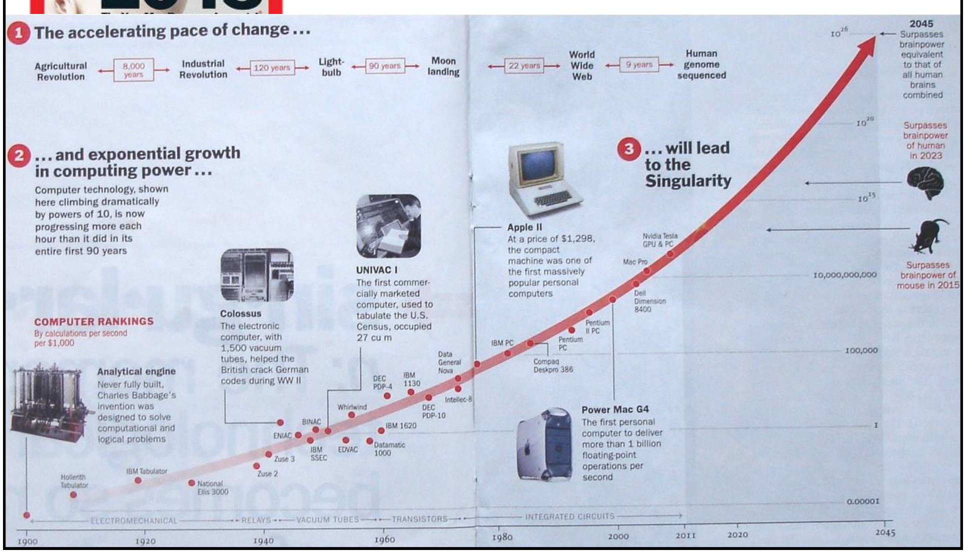


"Can machines think?"

A seminal paper, *Computing Machinery and Intelligence*, Alan Turing published in 1950 in *Mind*.

27

Singularity?



28

6.2 Neurons

- Models of neural networks

(Artificial) Neural Networks Approach *(bottom-up approach)*

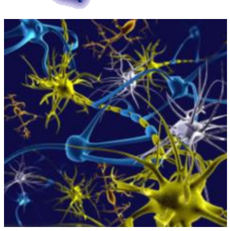
° NNs started on findings and discoveries in neurobilogy and neuroscience. We gradually realise that human brains bear little resemblance to the von Neumann type of computers .

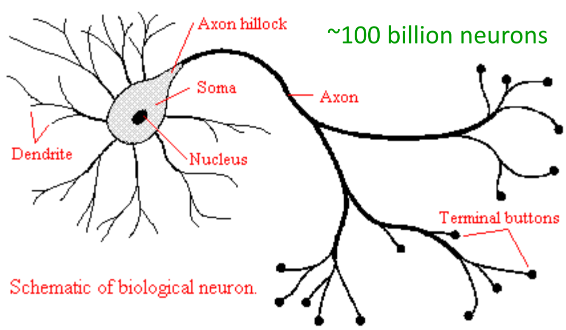
° The human brain has about 100 billion interconneted neurons, Each is a cell that uses biochemical reactions to receive, process and transmit stimulus. Each neuron is a simple processing unit, which receives (through dendrites and synapses) stimuli (input) from roughly 1000 neighbouring neurons (axons), cumulates these stimuli (in soma) and fires through its axon to its output neurons if the aggregated input is over a threshold (hillock). Networks of these cells form the basis of information processing.

29

6.2 Neurons

- Biological neurons





Schematic of biological neuron.

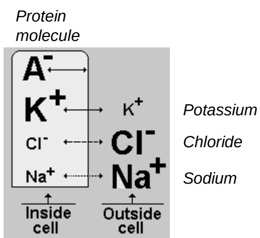
~100 billion neurons

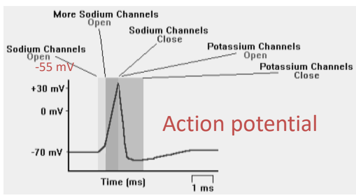
The brain has ~100 billion interconnected neurons, each connecting to ~1000 neighbouring neurons.

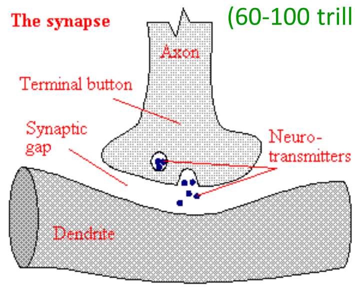
30

6.2 Neurons

- Biological neurons







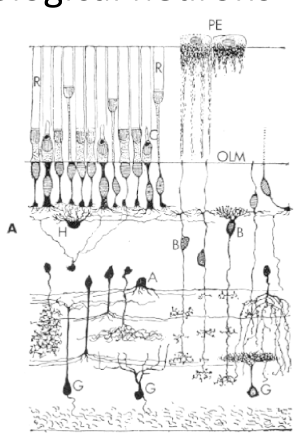
(60-100 trillion) synapses

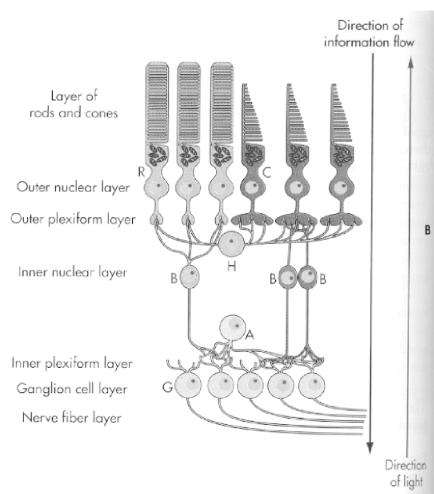
Axon's action potential opens ion channel, producing Ca^{2+} that leads to release of neurotransmitter, causing ion-conducting channels to open. Synapses can have either excitatory (depolarizing) or inhibitory (hyperpolarizing) effect on postsynaptic neurons.

31

6.2 Neurons

- Biological neurons



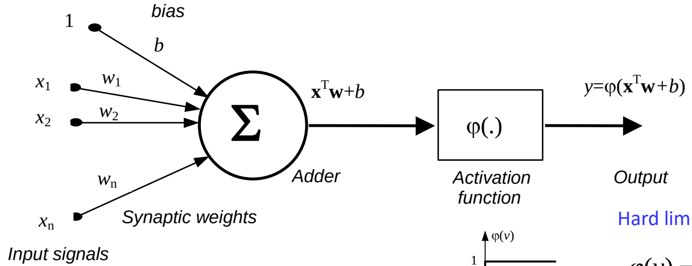


32

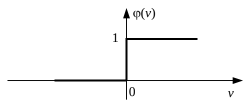
6.2 Neurons

- Models of neurons

Perceptron

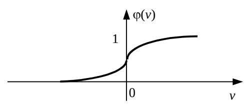


Hard limit function



$$\varphi(v)=\begin{cases}1 & \text{if } v>0\\ 0 & \text{otherwise}\end{cases}$$

Sigmoid function

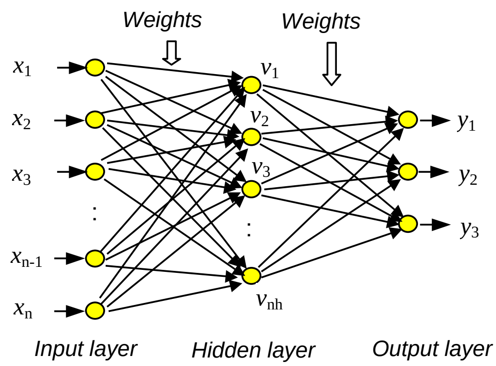


$$\varphi(v)=\frac{1}{1+e^{-v}}$$

33

6.2 Neurons

- Models of neural networks



Multilayer Perceptron (MLP)

34

6.2 Neurons

- Artificial neural networks: a history

McCulloch and Pitts	1942:	Linear Threshold Gate (LTG)
Wiener	1948:	"Cybernetics"
Hebb	1949:	*"Organisation of Behaviour"*
Rosenblatt	1958:	Perceptron
Minsky and Papert	1969:	"Perceptron"
Werbos	*1974:*	*Back-propagation algorithm*
Hopfield	1982:	Hopfield Recurrent Network
Kohonen	1982:	Self-Organising Map
Rumelhart, et. al.	1986:	Multilayer Perceptron

35

Hopfield net

Step 1: Assign Connection Weights

$$t_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s, & i \neq j \\ 0, & i = j,\ 0 \leq i, j \leq N-1 \end{cases}$$

t_{ij} is the connection weight from node i to node j and x_i^s which can be +1 or -1 is element i of the exemplar class s.

Step 2: Initialize with Unknown Input Pattern

$$\mu_i(0) = x_i,\ 0 \leq i \leq N\text{-}1,$$

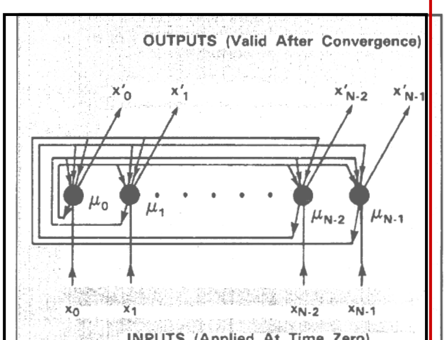
$\mu_i(t)$ is the output of node i at time t and x_i is the element i of the input pattern.

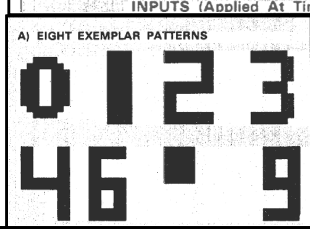
Step 3: Iterate Until Convergence

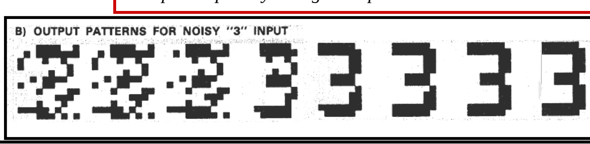
$$\mu_j(t+1) = f[\sum_{i=0}^{N-1} t_{ij}\mu_i(t)], \quad 0 \leq j \leq N-1$$

f is a activation function such as hard limit. The process is repeated until the node outputs remain unchanged with further iterations. The node outputs represent the exemplar pattern that best matches the input.

Step 4: Repeat by Going to Step 2







36

6.3 Learning paradigms

- Learning

The process which leads to the modification of behaviour
_Oxford Dictionary

"Some growth process or metabolic change takes place"
_ Donald Hebb

Learning is a process by which the free parameters of neural networks are adapted through a process of simulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place.
_Simon Haykin

37

6.3 Learning paradigms

- Learning

Learning, in neural network terms, is the process of modification of connection weights.

- Supervised Learning
  - Error-correction learning $\min \| d - y(x,w) \|^2$

$$\Delta w \approx \alpha e(d, y, x)$$

- Reinforcement learning $\max_a \Sigma reward\{s,a\}$ s: state, a: action

$$\Delta w \approx reward$$

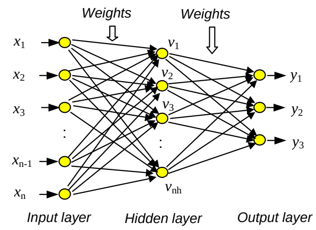
- Unsupervised Learning
  - Hebbian learning $\Delta w \approx \alpha y x$
  - Competitive learning $\Delta w_i \approx \begin{cases} \alpha f(x, y_i), & \text{if neuron } i \text{ wins} \\ 0, & \text{otherwise} \end{cases}$
  - Self-organisation $\Delta w_i \approx \begin{cases} \alpha y_i x, & \text{if neuron } i \text{ close to winner} \\ 0, & \text{otherwise} \end{cases}$

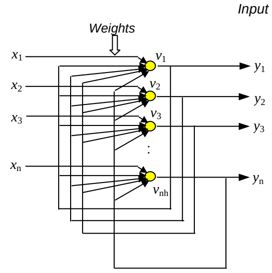
Global order can arise from local interactions (Turing 1952)

38

6.4 Network architectures

- Feedforward
  - Feed-forward Networks
    - Perceptron and multilayer perceptron
    - Radial basis function
    - Support vector machine
  - Recurrent Networks
    - Hopfield networks
    - Boltzmann machine



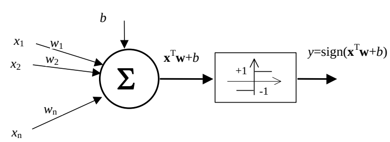


39

6.5 Feedforward networks

- Perceptron learning

  Least (mean) square method:



$$y(t) = \sum_{k=0}^{n} x_k(t) w_k(t) = \mathbf{x}(t)^T \mathbf{w}(t) \qquad e = d(t) - y(t)$$

$$J = \frac{1}{2} E[e^2(t)] \qquad \text{(without thresholding or activation function)}$$

$$J(t) = \frac{1}{2} e^2(t) \qquad \frac{\partial J(t)}{\partial \mathbf{w}} = e(t) \frac{\partial e(t)}{\partial \mathbf{w}} = -\mathbf{x}(t) e(t)$$

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \alpha \frac{\partial J(t)}{\partial \mathbf{w}} = \mathbf{w}(t) + \alpha \mathbf{x}(t) e(t) = \mathbf{w}(t) + \alpha \mathbf{x}(t)[d(t) - \mathbf{x}(t)^T \mathbf{w}(t)]$$

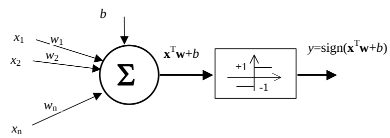
$$b(t+1) = b(t) + \alpha [d(t) - \mathbf{x}(t)^T \mathbf{w}(t)]$$

Can you derive learning rules for case with sigmoid activation function?

40

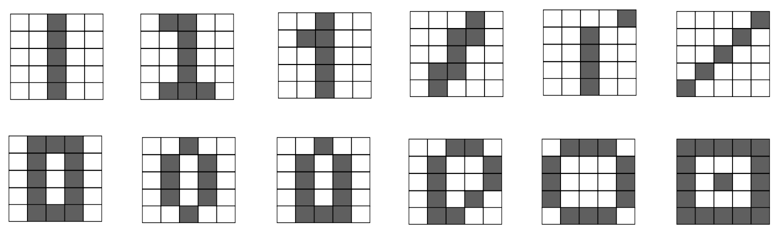
6.5 Feedforward networks

- Perceptron learning



Example: learn to classify hand written digits: '0' and '1', each of 5x5 pixels.

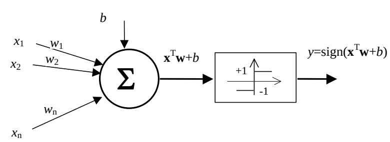
Use 4 for training and remaining 2 for testing.



41

6.5 Feedforward networks

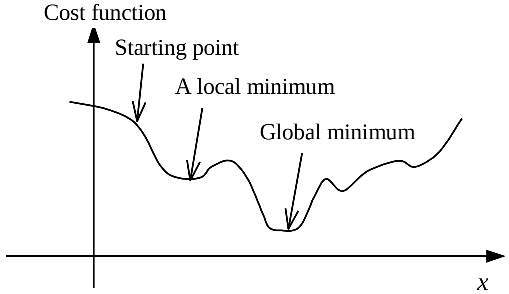
- Perceptron limitations



Linear separation

Sensitive to initial states

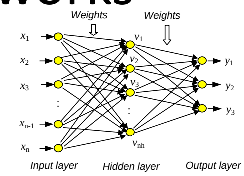
Local minima



42

$$\varphi(v) = \frac{1}{1+e^{-v}}$$

6.5 Feedforward networks



- Multiplayer perceptron

Back-propagation algorithm

Step 1: *Initialisation*. Set all weights and nodes' biases to small random numbers.

Step 2: *Presentation of training examples*. input $\mathbf{x}=[x_1, x_2, \ldots x_n]^T$ and desired output $\mathbf{d}=[d_1, d_2,..d_m]^T$.

Step 3: *Forward computation*. Calculate the outputs of the hidden and output layer,

$$y_k = \varphi_k^o\left(\sum_{j=1}^{n_h} w_{jk}^o v_j + b_k^o\right) \qquad v_j = \varphi_j^h\left(\sum_{i=1}^{n} w_{ij}^h x_i + b_j^h\right)$$

Step 4: *Backward computation and updating weights*. Compute the error terms,

$$\delta_k^o = e_k^o(\varphi_k^o)' = e_k^o y_k(1-y_k) = y_k(1-y_k)(d_k - y_k) \qquad w_{jk}^o = w_{jk}^o + \alpha\delta_k^o v_j$$

b_k ...

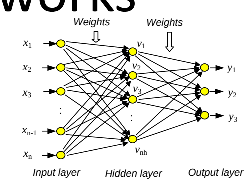
$$\delta_j^h = (\varphi_j^h)'\sum_{k=1}^{m}\delta_k^o w_{jk}^o = v_j(1-v_j)\sum_{k=1}^{m}\delta_k^o w_{jk}^o \qquad w_{ij}^h = w_{ij}^h + \alpha\delta_j^h x_i$$

b_j ...

Step 5: *Iteration*. Repeat steps 3 and 4 and stop when the total error reaches the required level).

$$\varphi(v) = \frac{1}{1+e^{-v}}$$

6.5 Feedforward networks

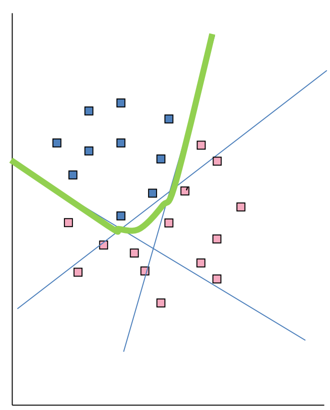


- Multiplayer perceptron

How does MLP form nonlinear separations?

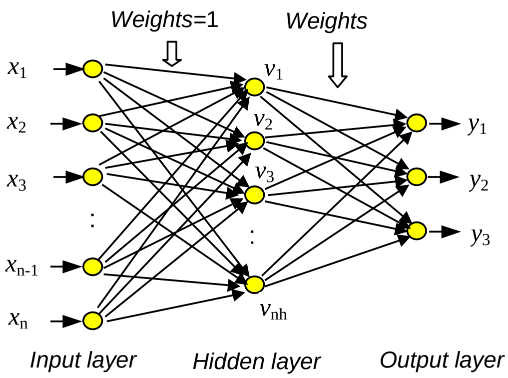
Key points:

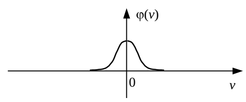
- Each hidden node forms a linear separate boundary;
- An output node is a combination of all hidden nodes, in effect forming a piecewise linear (or nonlinear) separation boundary.



6.5 Feedforward networks

- Radial basis functions





Properties:

•***Hidden neurons are Gaussian***

$$v_k = \exp\left(-\frac{\left\|\mathbf{x}-\mathbf{m}_k\right\|^2}{2\sigma^2}\right)$$

•***Output neurons are linear***
similar to Single Layer Perceptrons.

•***Easy to train***
Hidden nodes can be chosen first by clustering; output layer is trained as single layer perceptrons.

•***Fast to converge***

6.5 Feedforward networks

- Radial basis functions

The RBF model came from two fundamental theories

The Universal Approximation Theory: any function can be approximated with arbitrary precision by a weighted sum of a set of non-constant, bounded and monotone-increasing continuous functions,

$$\hat{f} = \sum_{i=1}^{M} w_i \varphi(\mathbf{x}, \xi)$$

Cover's Separability Theory: A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low high dimensional space.

6.6 Self-organising networks

- Hebbian learning

When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased.

In mathematical term: $\Delta w = \alpha x y$

Oja's rule:

$$w_i(t+1) = \frac{w_i(t) + \alpha x_i(t) y(t)}{\{\sum_{j=1}^{n} [w_j(t) + \alpha x_j(t) y(t)]^2\}^{1/2}} \approx w_i(t) + \alpha y(t)[x_i(t) - y(t) w_i(t)] + O(\alpha^2)$$

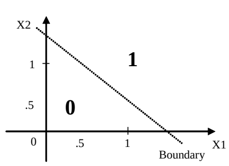
47

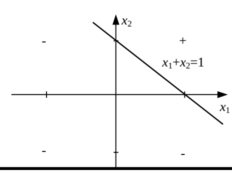
6.6 Self-organising networks

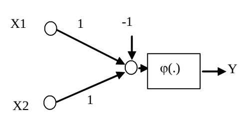
- Hebbian learning

train a perceptron for an AND function, $y = b + \Sigma x_i w_i$ $\Delta w = xy$

Input			**Output**	**Weight Changes**			**Weights**		
x_1	x_2	1	y	Δw_1	Δw_2	Δb	w_1	w_2	b
							0	0	0
1	1	1	1	1	1	1	1	1	1
1	-1	1	-1	-1	1	-1	0	2	0
-1	1	1	-1	1	-1	-1	1	1	-1
-1	-1	1	-1	1	1	-1	2	2	-2



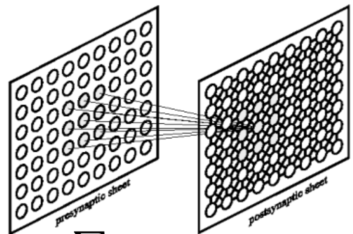




48

6.6 Self-organising networks

- Von der Malsburg and Willshaw model (1973,1976)



$$\frac{\partial y_i(t)}{\partial t} + c y_i(t) = \sum_j w_{ij}(t) x_i(t) + \sum_k e_{ik} y_k(t) - \sum_{k'} b_{ik'} y_{k'}(t)$$

$$y_j^*(t) = \begin{cases} y_j^*(t) - \theta, & \text{if } y_j^*(t) > \theta \\ 0 & \text{otherwise} \end{cases}$$

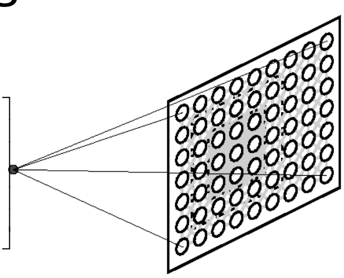
$$\frac{\partial w_{ij}(t)}{\partial t} = \alpha x_i(t) y_j^*(t), \quad \text{subject to } \sum w_{ij} = \text{constant}$$

49

6.6 Self-organising networks

- Kohonen‘s SOM

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ : \\ : \\ x_n \end{bmatrix}$$



$$y_j(t+1) = \varphi[\mathbf{w}_j^T \mathbf{x}(t) + \sum_i h_{ij} y_i(t)]$$

$$\|\mathbf{w}_i(t) - \mathbf{x}(t)\| = \min_j \|\mathbf{w}_j(t) - \mathbf{x}(t)\|$$

$$y_j(t+1) = \begin{cases} 1, & \text{If neuron j is inside the bubble} \\ 0, & \text{Otherwise} \end{cases}$$

$$\frac{\partial w_{ij}(t)}{\partial t} = \alpha y_j(t) x_i(t) - \beta y_j(t) w_{ij}(t) = \alpha[x_i(t) - w_{ij}(t)] y_j(t) = \begin{cases} \alpha[x_i(t) - w_{ij}(t)], & \text{if } j \in \eta(t) \\ 0 & \text{if } j \notin \eta(t) \end{cases}$$

50

6.6 Self-organising networks

- SOM algorithm
  - *At each time t, present an input,* $\mathbf{x}(t)$*, select the winner.*

$$v = \arg\min_{c \in \Omega} \left\| \mathbf{x}(t) - \mathbf{w}_c \right\|$$

  - *Updating the weights of winner and its neighbours.*

$$\Delta \mathbf{w}_k(t) = \alpha(t)\eta(v,k,t)[\mathbf{x}(t) - \mathbf{w}_v(t)]$$

  - *Repeat until the map converges.*

Typical neighbourhood function: $\eta(v,k,t) \propto \exp[-\frac{\|v-k\|^2}{2\sigma(t)^2}]$

6.6 Self-organising networks

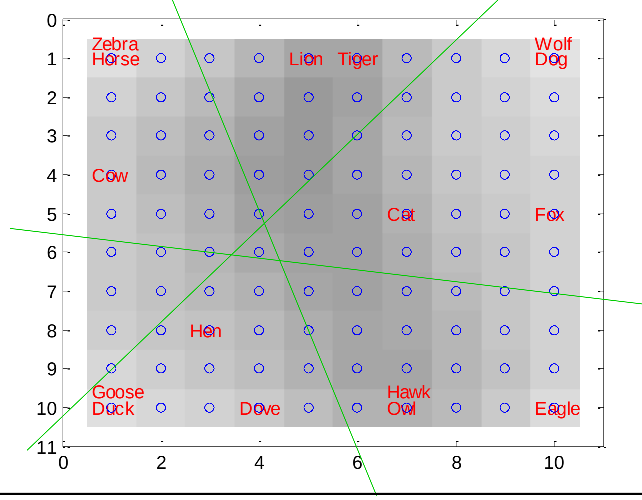
- SOM clustering objects (animals) according to attributes

	is			has						likes to			
	small	medium	big	2legs	4legs	hair	hooves	mane	feather	hunt	run	fly	swim
Dove	1	0	0	1	0	0	0	0	1	0	0	1	0
Hen	1	0	0	1	0	0	0	0	1	0	0	0	0
Duck	1	0	0	1	0	0	0	0	1	0	0	1	1
Goose	1	0	0	1	0	0	0	0	1	0	0	1	1
Owl	1	0	0	1	0	0	0	0	1	1	0	1	0
Hawk	1	0	0	1	0	0	0	0	1	1	0	1	0
Eagle	0	1	0	1	0	0	0	0	1	1	0	1	0
Fox	0	1	0	0	1	1	0	0	0	1	0	0	0
Dog	0	1	0	0	1	1	0	0	0	0	1	0	0
Wolf	0	1	0	0	1	1	0	1	0	1	1	0	0
Cat	1	0	0	0	1	1	0	0	0	1	0	0	0
Tiger	0	0	1	0	1	1	0	0	0	1	1	0	0
Lion	0	0	1	0	1	1	0	1	0	1	1	0	0
Horse	0	0	1	0	1	1	1	1	0	0	1	0	0
Zebra	0	0	1	0	1	1	1	1	0	0	1	0	0
Cow	0	0	1	0	1	1	1	0	0	0	0	0	0

6.6 Self-organising networks

- SOM

clustering result:

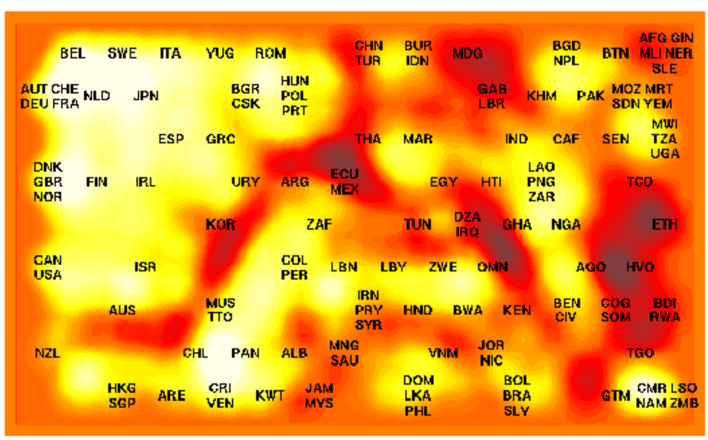


53

6.6 Self-organising networks

- SOM

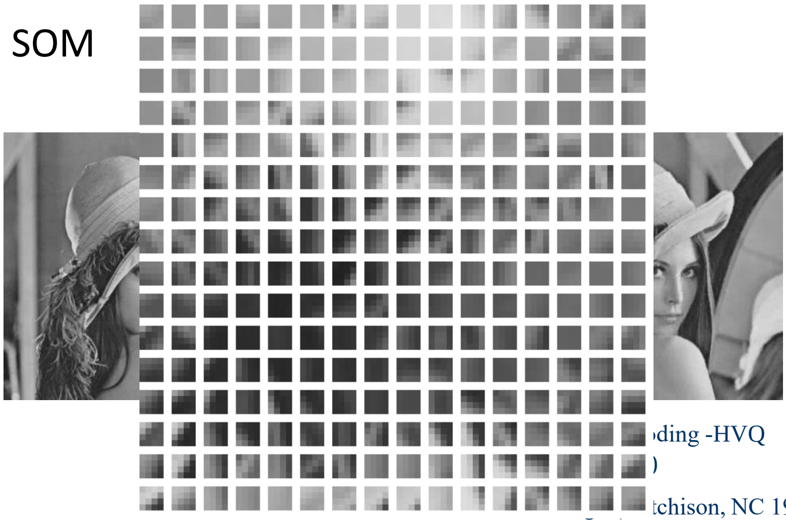
Data visualisation/management



54

6.6 Self-organising networks

- SOM



ding -HVQ

)

tchison, NC 1995),

(Durbin&Mitchson, Nature1990)

55

6.6 Self-organising networks

- SOM extensions
  - **HSOM** (Miikkulainen 1990), **DISLEX** (1990, 1997)
  - **PSOM** (Ritter 1993), **Hyperbolic SOM** (1999), **H^{2}SOM**
  - **Temporal Kohonen Map** (Chappell &Taylor 1993)
  - **Neural Gas** (Martinetz et al.1991), **Growing Grid** (Fritzke1995)
  - **ASSOM** (Kohonen 1997)
  - **Recurrent SOM** (Koskela, 1997), RecursiveSOM (Voegtlin2001)
  - **SOAR** (Lampinen&Oja 1989), **SOMAR** (Ni&Yin, 2007)
  - **Bayesian SOM** & **SOMN** (Yin&Allinson 1995,1997; Utsugi 1997)
  - **GTM** (Bishop et al. 1998)
  - **GHSOM** (Merkl et al. 2000), **TOC**(TreeSOM) (Freeman&Yin2004)
  - **PicSOM** (Laaksonen, Oja, et al., 2000)
  - **ViSOM** (Yin 2001, 2002), **gViSOM** (Yin 2007)

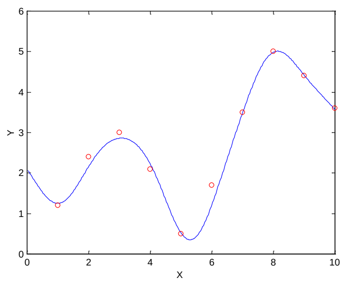
56

6.7 Learning & generalisation

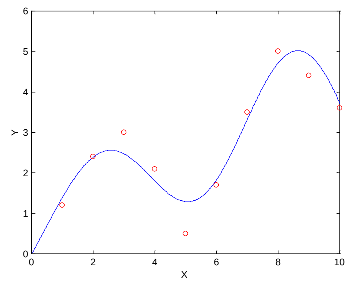
- Learning (fitting) & generalisation

On one hand, we want to get as high classification (or low error) rate as possible for the training data set.

On the other hand, we would also like the trained network to have good performance on unseen (testing) data.



Which is better?



57

6.7 Learning & generalisation

- Bias &Variance dilemma (finite data set D)

$$\Phi = \frac{1}{2}\int [y(x) - d(x)]^2 p(x)dx$$

$$[y(x) - d(x)]^2 = [y(x) - E\{y(x)\} + E\{y(x)\} - d(x)]^2$$
$$= [y(x) - E\{y(x)\}]^2 + [E\{y(x)\} - d(x)]^2$$
$$+ 2[y(x) - E\{y(x)\}][E\{y(x)\} - d(x)]$$

$$E_D\{[y(x) - d(x)]^2\} = [E_D\{y(x)\} - d(x)]^2 + E_D\{[y(x) - E\{y(x)\}]\}^2$$

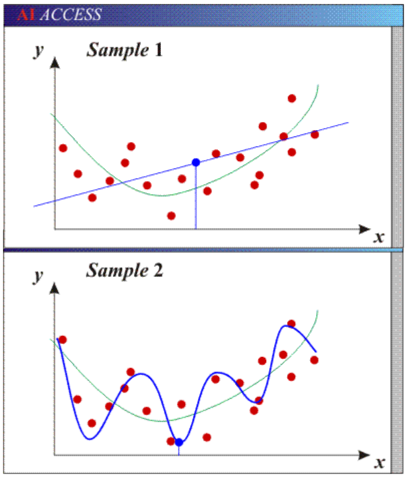
(bias)^2 (variance)

A close fitter/model yields small bias but large variance, while a loose fitter (or a wild guess) gives small variance but large bias.

58

6.7 Learning & generalisation

- Bias &Variance dilemma (finite data set D)



***g*(*x*): generating curve**

***dots*: sample**

For $h_1(x)$: $E_D\{[y(x)-E\{y(x)\}]\}^2 \to 0$

$as\ E\{y(x)\} = h_1(x) = y(x)$

Low variance, but large bias (error) !

For $h_2(x)$: $[E_D\{y(x)\}-d(x)]^2 \to 0$

$as\ E_D\{y(x)\} = E\{h_2(x)\}$

$= E\{g(x)+\varepsilon\} = g(x) = d(x)$

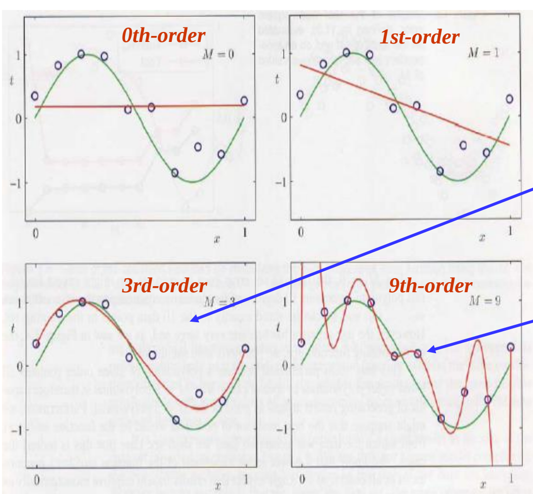
Low bias but large variance ! Poor generalisation !
That is, different samples give different $h_2(x)$, thus large variance! Or if for Sample 2, one still uses $h_2(x)$ obtained from Sample 1, the errors are large.

** Courtesy of http://www.aiaccess.net/*

59

6.7 Learning & generalisation

- Bias &Variance dilemma (finite data set D)



***sin*(2π*x*): generating curve**

***dots*: samples**

Best representation

Over-fitting !

60

Summary

- Introduction to perception & cognition*
- Artificial intelligence
- Retina & visual pathway
- Neurons & models
- Learning paradigms & network architectures
- Multilayer perceptron
- Self-organising networks
- Learning & generalisation

* For knowledge only

61