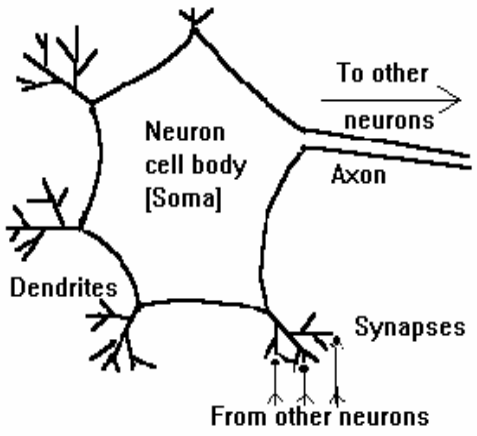
Connectionist Learning
(Artificial Neural Networks)

- The human brain contains about 1 billion neurons
- Each neuron is connected to thousands of others
- Neurons can be either excitatory or inhibitory
- Neurons perform very simple computations
- The computational power of the brain is derived from the complexity of the connections

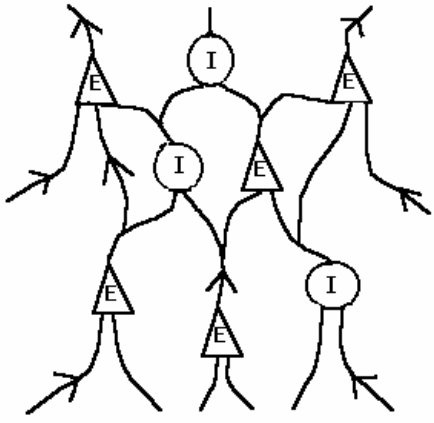
A Neuron



Axons and Synapses

- The primary mechanism for information transmission in the nervous system is the axon
- An axon relays all-or-nothing (binary) impulses
- Signal strength is determined from the frequency of the impulses
- An axon signal eventually arrives at a synapse
- A synapse may either attenuate or amplify the signal whilst transmitting it to a neuron
- A neuron accumulates the modified signals and produces an impulse on its own axon if the total synaptic input strength is sufficient

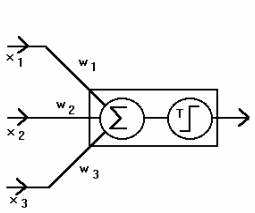
Assemblies of Neurons



Modifications to neuron assemblies can only be achieved by adjusting the attenuation or amplification which is applied at the synapses

Model of a Neuron (1943)

- Due to McCulloch and Pitts
- Organisation



x_i are inputs, w_i are weights

- Firing Rule

$$\sum_i x_i w_i > T$$

T is the threshold

Hebb Rule (1949)

- Due to Donald Hebb
- Pre-empted by William James (1890)
- Hebb Rule

  If a particular input is always active when a neuron fires then its weight should be increased

- Interpretation

  Classical conditioning -

  When two things happen within a short time of each other, a re-occurrence of one should make the other more likely to occur

Delta Rule (1960)

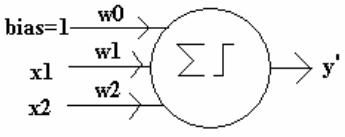
- Due to Widrow and Hoff
- Also known as -
  - Widrow-Hoff Rule
  - Least Mean Square Rule
- General algorithm
  1. Determine an *error* between the desired and actual outputs
  2. Modify all of the "live" weights and the threshold so as to remove a *portion* of the error

$$\Delta^p w_{ij} = \eta \delta_i^p O_j^p$$

  3. Repeat until the error is reduced to an acceptable level

Perceptron (1957-1962)

- Due to Rosenblatt, Wightman, et al.
- Organisation



x_i are inputs, w_i are weights,
y is desired output, y' is actual output

- Firing Rule

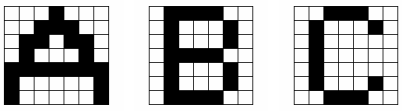
$$\sum_i x_i w_i \geq 0$$

- Learning Rule

$$\underline{w}^{new} = \underline{w}^{old} + \eta \cdot (y - y') \cdot \underline{x}$$

A Perceptron Application

- An early and influential success story of the Perceptron was recognising letters of the alphabet



- A 20x20 input grid was used
- The 400 binary input values were weighted and summed into output units representing different letters of the alphabet
  - Output 1 meant “A”
  - Output 2 meant “B”
  - Etc.

Perceptron Convergence Theorem

- Theorem

  If a solution to a classification problem exists then the general Perceptron learning rule will find it in a finite number of steps.

- Note that this theorem is qualified by the phrase

  if a solution to a classification problem exists

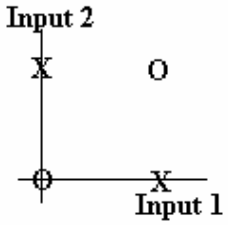
- Not all classification problems are solvable with a Perceptron
- BUT, for a problem that is solvable, the learning rule will definitely find it in a finite time

XOR Problem (I)

- Truth table

Input 1	*Input 2*	*Output*
0	0	0
1	0	1
0	1	1
1	1	0

- Linearly inseparable (Graphically)



XOR Problem (II)

- Truth table

Input 1	*Input 2*	*Output*
0	0	0
1	0	1
0	1	1
1	1	0

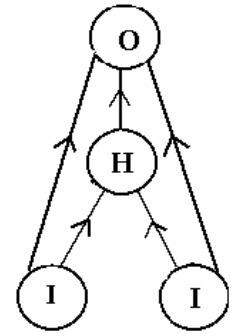
- Linearly inseparable (Algebraically)

0 0 0 =>	w0 < 0	(A)
1 0 1 =>	w0 + w1 >= 0	(B)
0 1 1 =>	w0 + w2 >= 0	(C)
1 1 0 =>	w0 + w1 + w2 < 0	(D)

Statement D is incompatible with Statements B and C since it requires w1+w2 to be less than a positive amount (w0 is -ve, see A) which neither is less than individually

Multi-Layered Perceptrons (MLPs)

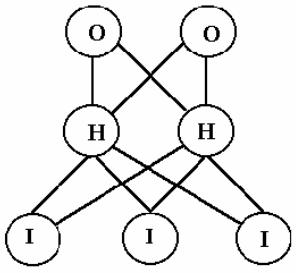
- The XOR problem is solvable if we add an extra "node" to a Perceptron



- A set of weights can be found for the above 5 connections which will enable the XOR of the inputs to be computed

MLPs Formalised

- MLPs become more manageable, mathematically and computationally, if we formalise them into a standard structure (or *topology* or *architecture*)



- Each node is connected to EVERY node in the adjacent layers and NO nodes in the same or any other layers

Weight finding in MLPs

- Although it has been known since the 1960's that Multi-Layered Perceptrons are not limited to linearly separable problems there remained a big problem which blocked their development and use
  - How do we find the weights needed to perform a particular function?
- The problem lies in determining an error at the hidden nodes
  - We have no desired value at the hidden nodes with which to compare their actual output and determine an error
  - We have a desired output which can deliver an error at the output nodes but how should this error be divided up amongst the hidden nodes?

MLP Learning Rule

- In 1986 Rumelhart, Hinton and Williams proposed a Generalised Delta Rule
  - Also known as Error Back-Propagation or Gradient Descent Learning
- This rule, as its name implies, is an extension of the good old Delta Rule

$$\Delta^p w_{ji} = \eta \delta_j^p o_i^p$$

- The extension appears in the way we determine the δ values
  - For an output node we have -

$$\delta_j^p = \left(T_j^p - o_j^p\right) \cdot f_j'\left(net_j^p\right)$$

  - For a hidden node we have -

$$\delta_j^p = f_j'\left(net_j^p\right) \cdot \sum_k \delta_k^p w_{kj}$$

MLP Training Regime

- The back-propagation algorithm
  1. Feed inputs forward through network
  2. Determine error at outputs
  3. Feed error backwards towards inputs
  4. Determine weight adjustments
  5. Repeat for next input pattern
  6. Repeat until all errors acceptably small
- Pattern based training
  - Update weights as each input pattern is presented
- Epoch based training
  - Sum the weight updates for each input pattern and apply them after a complete set of training patterns has been presented (after one **epoch** of training)

ANN Bibliography

- Hebb, D.O., 1949, *The Organization of Behaviour*, John Wiley.
- McCulloch, W.S. & Pitts, W.H., 1943, A Logical Calculus of The Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, **5**, pp.115-133.
- Rosenblatt, F., 1958, The Perceptron – A Probabilistic Model for Information Storage and Organisation in the Brain, *Psychological Review*, **65**, pp. 386-408.
- Rumelhart, D.E., Hinton, G.E. & Williams, R.J., 1986, Learning Internal Representations by Error Propagation, in Rumelhart, D.E. & McClelland, J.L., (Eds), 1986, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press.
- Widrow, B. & Hoff, M.E., 1960, Adaptive Switching Circuits, *IRE WESTCON Convention*, pp. 96-104.