Aims

Section aim:
- To teach the fundamental theory and practical applications of artificial neural networks.

Session Aim:
- This will entail an introduction to the biological background of these methods and implementation issues relevant to the development of practical systems.

Biological neuron

http://hepunx.rl.ac.uk/~candreop/minos/NeuralNets/neuralNetIntro.html
The human brain consists of approx. **10 billion neurons** interconnected with about **10 trillion synapses**.

A neuron is a specialized cell for **receiving, processing and transmitting informations** by biochemical means (**neurotransmitters**).

Electric charge from the activity of neighboring neurons reaches the neuron through its **input synaptic tree** and it is added. The summed signal is passed to the soma that processing this information that it simply forwards to the axis hillock that applies a signal threshold.

If the summed signal > threshold, the neuron fires and a constant (independent from the input signal) output signal is transmitted through the axon to the output synaptic tree and, from there, to other neurons.
The strength and polarity of the signal passed to the other neuron depends on the neurochemical features of each synapse and it varies so as to adapt the network behavior.

Simplified neuron

Taken from http://www.geog.leeds.ac.uk/people/a.turner/projects/medalus3/Task1.htm

Exercise 1

In groups of 2-3, as a group:

Write down one question about this topic?
McCulloch-Pitts model

\[ Y = 1 \text{ if } W_1X_1 + W_2X_2 + W_3X_3 \geq T \]
\[ Y = 0 \text{ if } W_1X_1 + W_2X_2 + W_3X_3 < T \]

Logic functions - OR

\[ Y = X_1 \text{ OR } X_2 \]

Logic functions - AND

\[ Y = X_1 \text{ AND } X_2 \]
Logic functions - NOT

\[ Y = \text{NOT} \ X \]

McCulloch-Pitts model

\[ Y = 1 \text{ if } W_1X_1 + W_2X_2 + W_3X_3 \geq T \]
\[ Y = 0 \text{ if } W_1X_1 + W_2X_2 + W_3X_3 < T \]

Introduce the bias

Take the threshold over to the other side of the equation and replace it with a weight \( W_0 \) which equals \(-T\), and include a constant input \( X_0 \) which equals 1.

\[ Y = 1 \text{ if } W_1X_1 + W_2X_2 + W_3X_3 - T \geq 0 \]
\[ Y = 0 \text{ if } W_1X_1 + W_2X_2 + W_3X_3 - T < 0 \]
\[ Y = 1 \text{ if } W_1X_1 + W_2X_2 + W_3X_3 + W_0X_0 \geq 0 \]
\[ Y = 0 \text{ if } W_1X_1 + W_2X_2 + W_3X_3 + W_0X_0 < 0 \]
### Short-hand notation

Instead of writing all the terms in the summation, replace with a Greek sigma $\Sigma$

$Y = 1$ if $W_1X_1 + W_2X_2 + W_3X_3 + W_0X_0 \geq 0$

$Y = 0$ if $W_1X_1 + W_2X_2 + W_3X_3 + W_0X_0 < 0$

becomes

$$y = \begin{cases} 
W_1X_1 + W_2X_2 + W_3X_3 + W_0X_0 \geq 0 \\
W_1X_1 + W_2X_2 + W_3X_3 + W_0X_0 < 0
\end{cases}$$

### Revision

$$\sum_{i=0}^{4} i = 0 + 1 + 2 + 3 + 4 = 10$$

$$\sum_{i=0}^{4} X_i = X_0 + X_1 + X_2 + X_3 + X_4 = 14$$

address | 0 | 1 | 2 | 3 | 4  
array    | 2 | 3 | 7 | 0 | 2  

### Logic functions - OR

$Y = X_1 \text{ OR } X_2$

Y = X1 OR X2
**Logic functions - AND**

\[
\begin{array}{c}
X_0 & X_1 & X_2 & Y \\
-2 & 1 & 1 & Y = X_1 \text{ AND } X_2 \\
\end{array}
\]

**Logic functions - NOT**

\[
\begin{array}{c}
X_0 & X_1 & Y \\
0 & -1 & Y = \text{NOT } X_1 \\
\end{array}
\]

**The weighted sum**

- The weighted sum, \( \Sigma W_i X_i \) is called the “net” sum.
- Net = \( \Sigma W_i X_i \)
- \( y = 1 \) if net \( \geq 0 \)
- \( y = 0 \) if net < 0
**Hopfield network**
- Feedback network
- Easy to train
- Single layer of neurons
- Neurons fire in a random sequence
- Good as content-addressable memory

**Radial basis function network**
- Feedforward network
- Has 3 layers
- Hidden layer uses statistical clustering techniques to train
- Output layer is like an MLP
- Good at pattern recognition
Radial basis function networks

Input layer → Hidden layer → Output layer

Kohonen network
- Neurons arranged in a square
- All neurons connected to the inputs
- Neurons not connected to each other
- Often uses a MLP as an output layer
- Neurons are self-organising
- Trained using "winner-takes all"

What can they do?
- Perform tasks that conventional software cannot do
- For example, reading text, understanding speech, recognising faces
Neural network approach
- Set up examples of numerals
- Train a network
- Done, in a matter of seconds

Learning and generalising
- Neural networks can do this easily because they have the ability to learn and to generalise from examples
- Learning is achieved by adjusting the weights
- Generalisation is achieved because similar patterns will produce an output

Summary
- Neural networks have a long history but are now a major part of computer systems
- They can perform tasks (not perfectly) that conventional software finds difficult
- Introduced
  - McCulloch-Pitts model and logic
  - Multi-layer perceptrons
  - Wisard
  - Hopfield network
  - Kohenen network

- Neural networks can
  - Classify
  - Learn and generalise.
  - Learning is achieved by adjusting the weights
  - Generalisation is achieved because similar patterns will produce an output