G51FAI
Fundamentals of AI

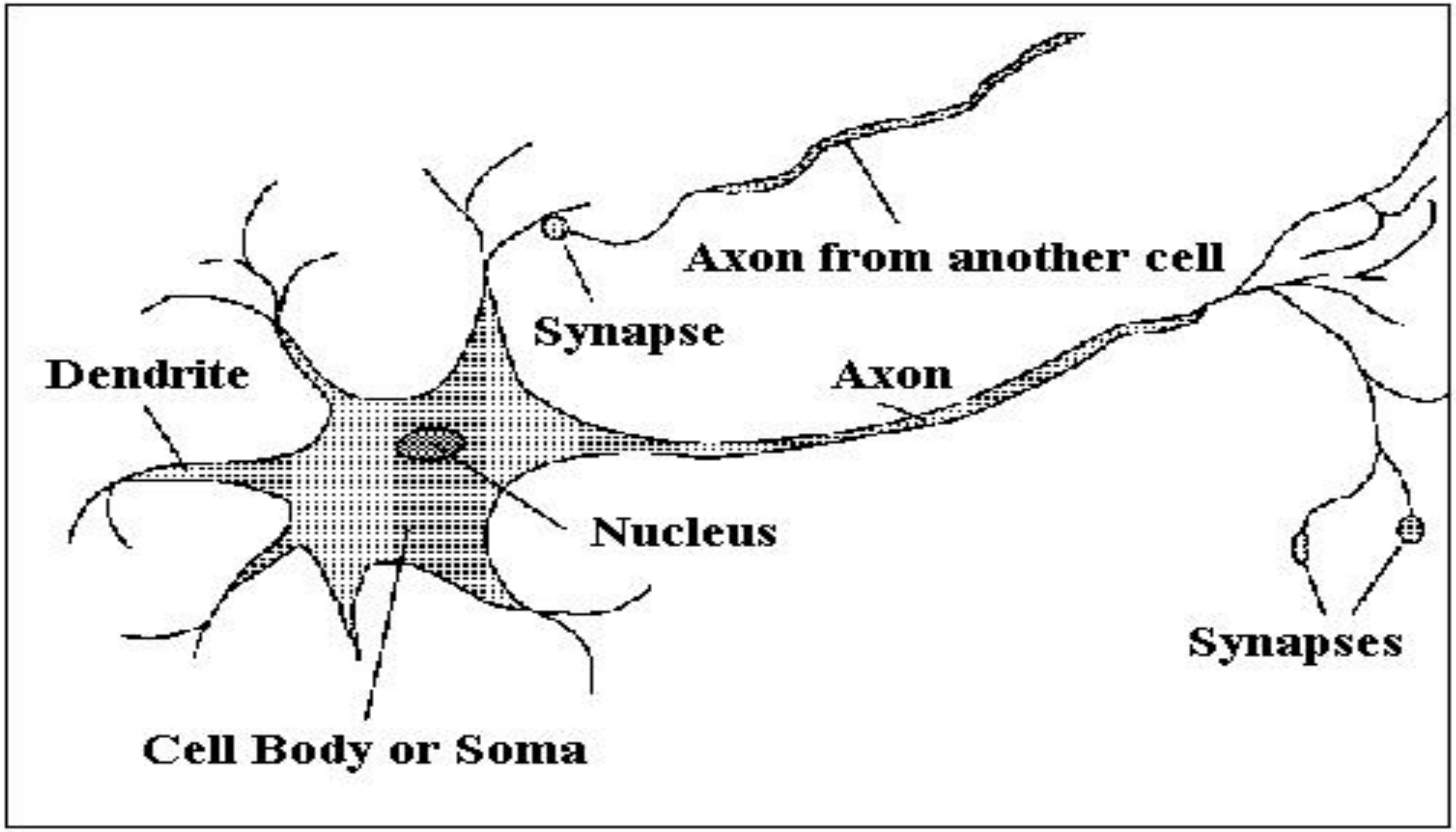
Neural Networks

What is a Neural Network

Computers & Symbols versus Nets & Neurons (Kevin Gurney)

- An interconnected assembly of simple processing elements, ***units or nodes***, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or ***weights***, obtained by a process of adaptation to, or ***learning*** from, a set of training patterns.

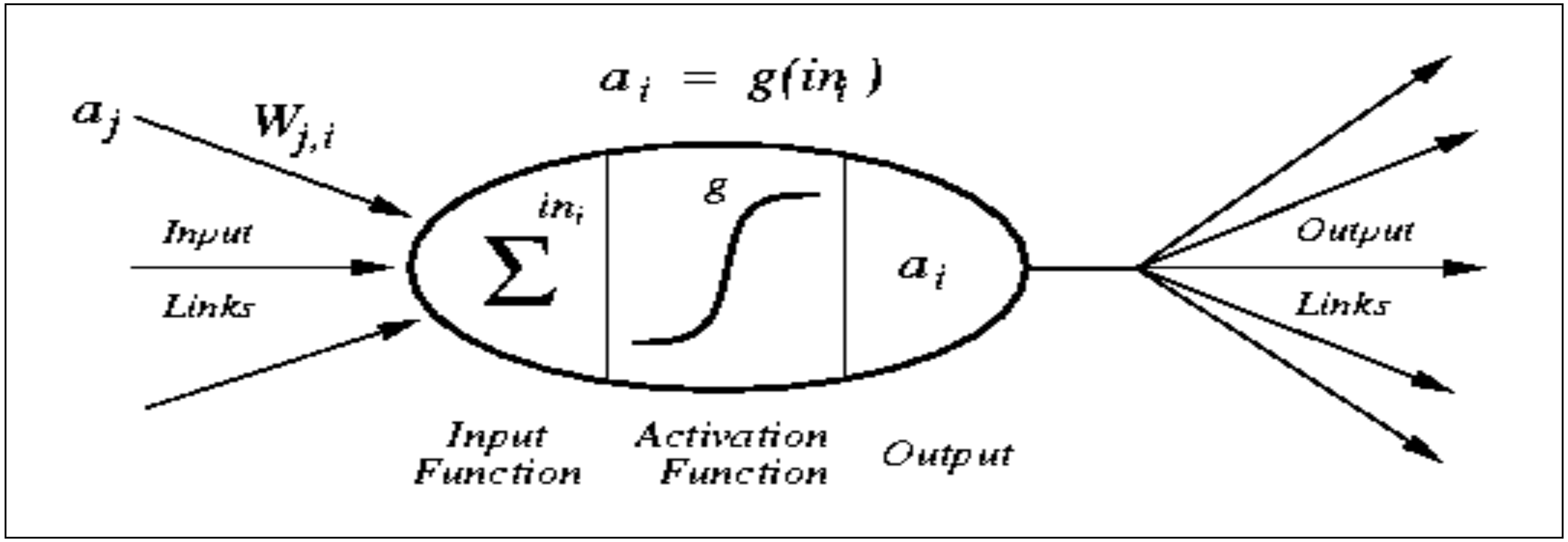
Biological Neuron



Biological Neural Networks

- We are born with about 100 billion neurons
- A neuron may connect to as many as 100,000 other neurons
- Signals “move” via complex electrochemical reactions
- The synapses release a chemical transmitter – the sum of which can cause a threshold to be reached – causing the neuron to “fire”
- Synapses can be inhibitory or excitatory

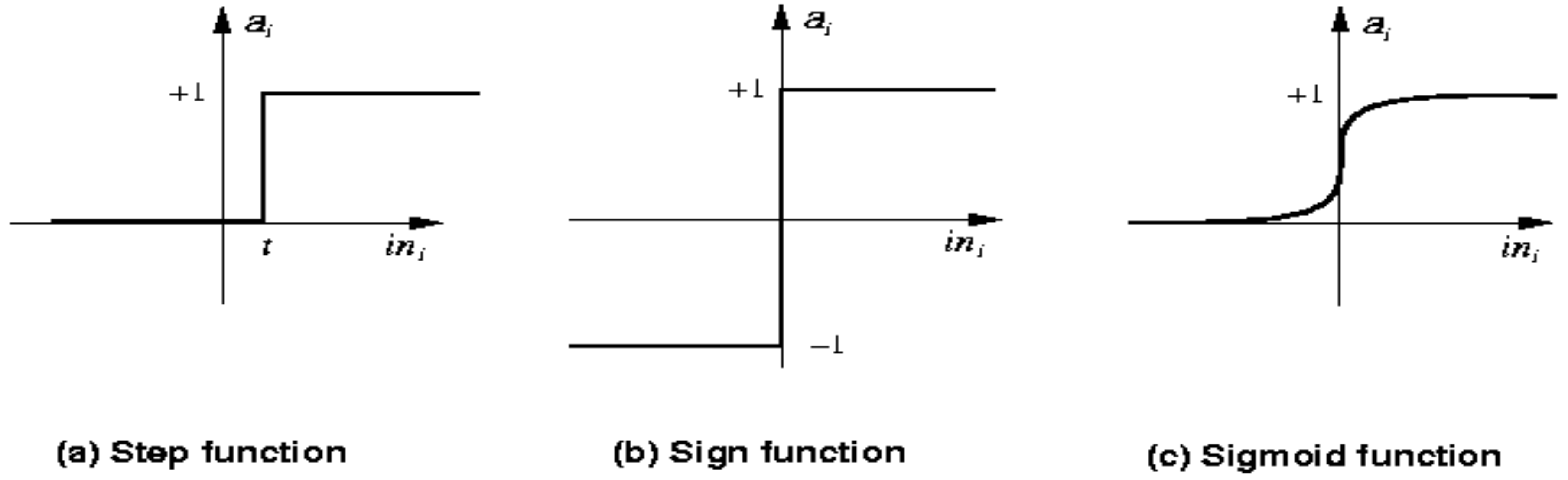
Modeling a Neuron



$$in_i = \sum_j W_{j,i} a_j$$

- a_j :Activation value of unit j
- $w_{j,i}$:Weight on the link from unit j to unit i
- in_i :Weighted sum of inputs to unit i
- a_i :Activation value of unit i
- g :Activation function

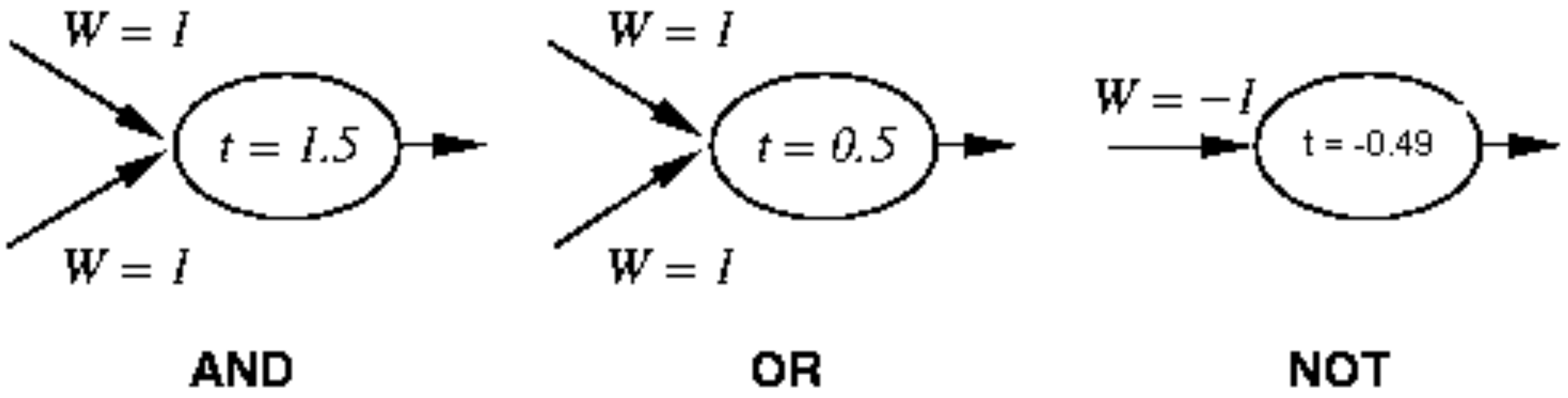
Activation Functions

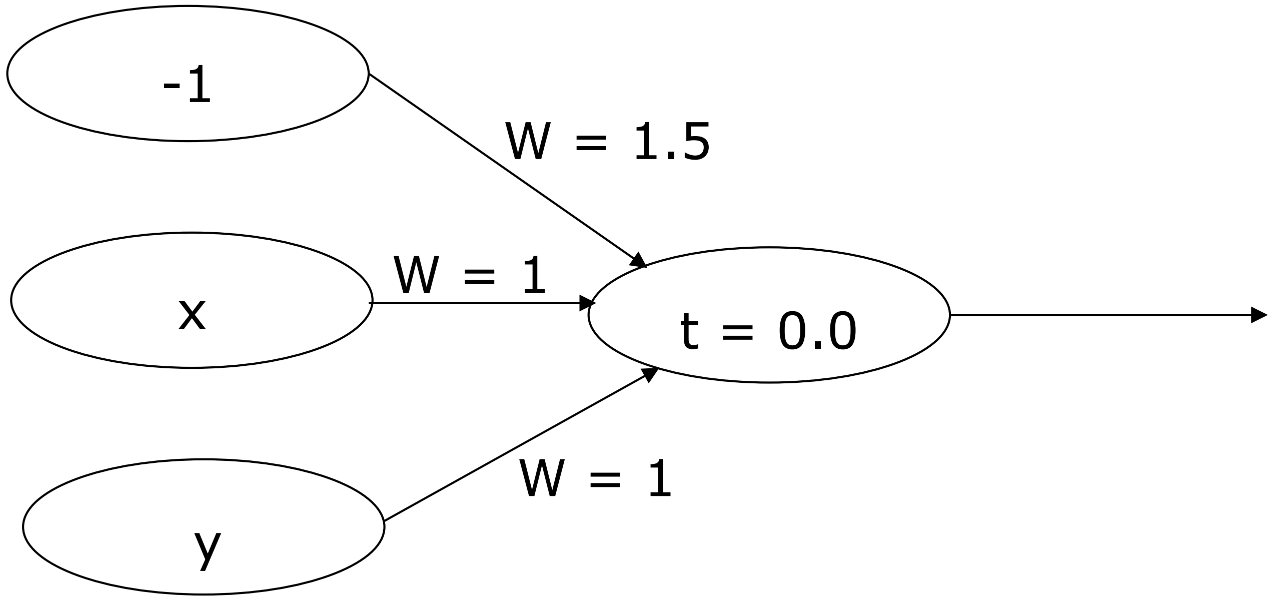


(a) Step function (b) Sign function (c) Sigmoid function

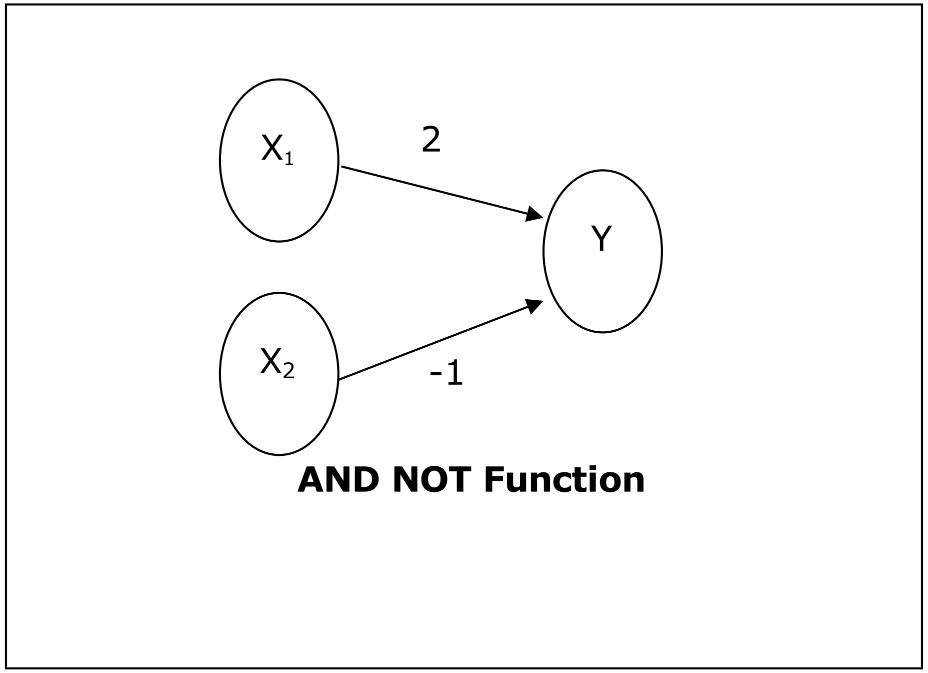
- $\text{Step}_t(x) =$ 1 if x >= t, else 0
- Sign(x) = +1 if x >= 0, else −1
- Sigmoid(x) = $1/(1+e^{-x})$
- Identity Function

Simple Networks





Neural Networks – AND NOT

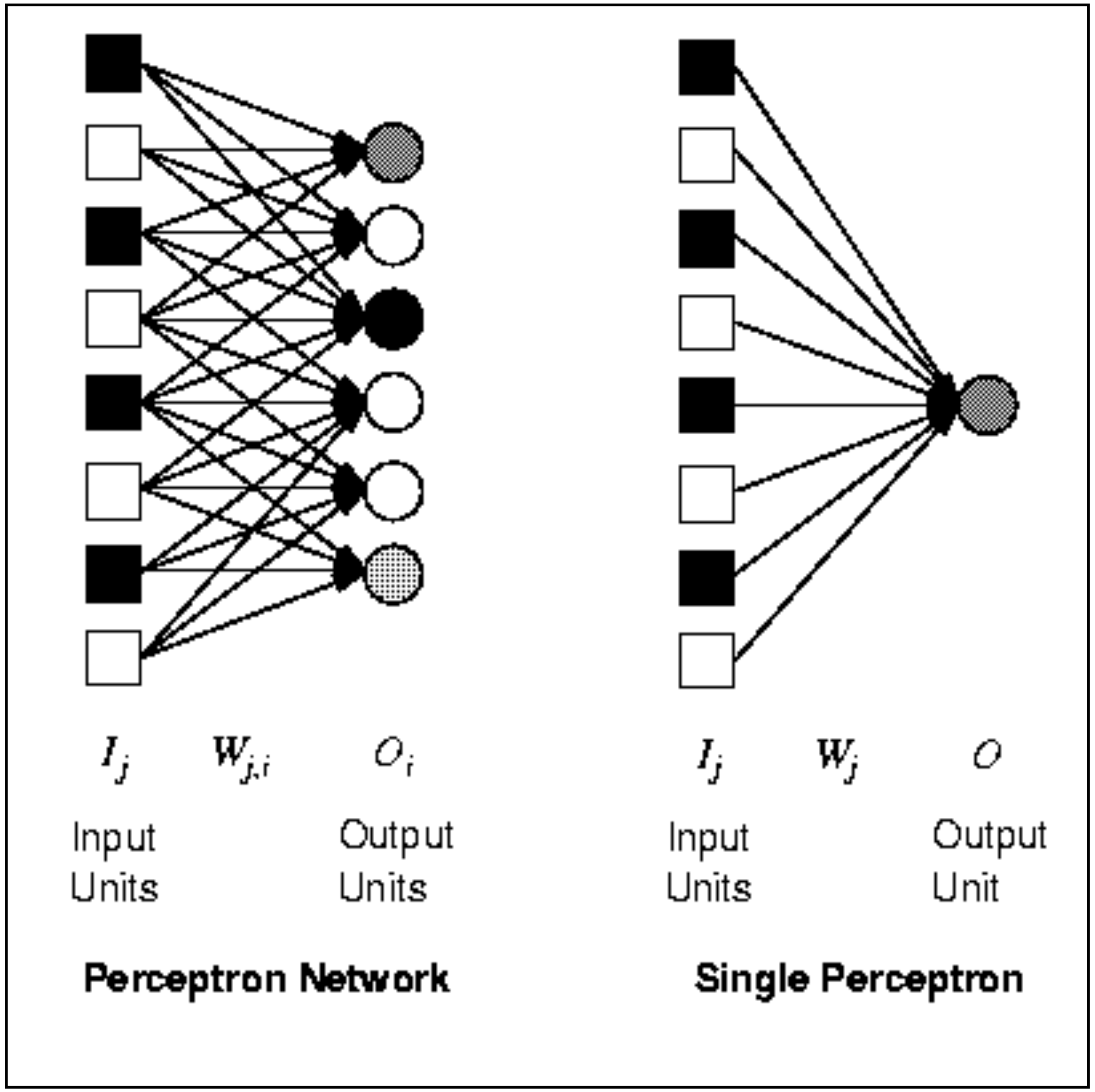


AND NOT Function

threshold(Y) = 1.5

AND NOT		
X1	**X2**	**Y**
1	1	0
1	0	1
0	1	0
0	0	0

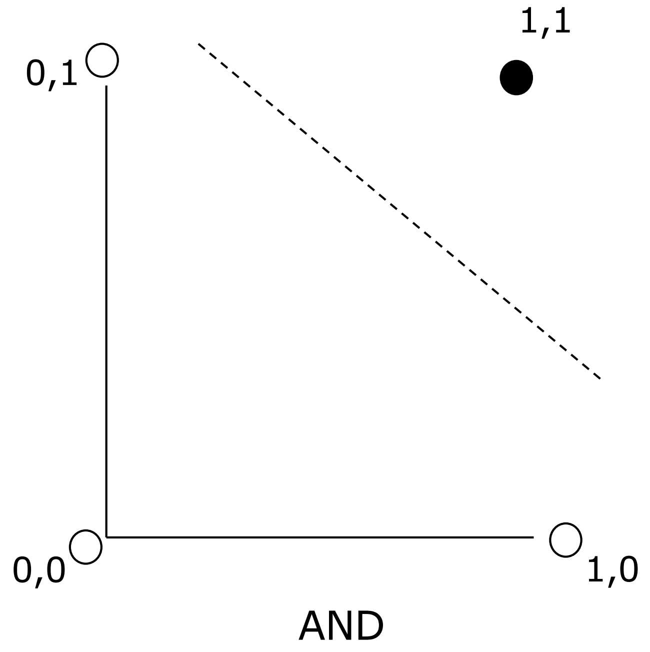
Perceptron

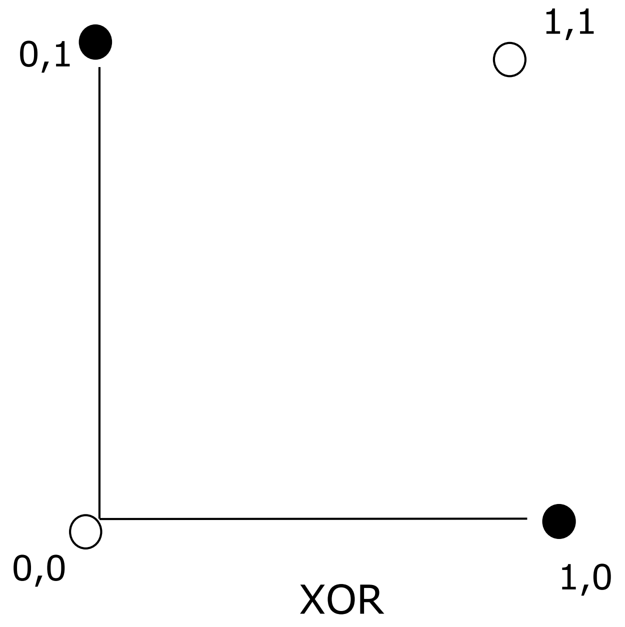


- A single weight only affects one output so we can restrict our investigations to a model as shown on the right
- Notation can be simpler, i.e.

$$O = Step_0 \sum_j WjIj$$

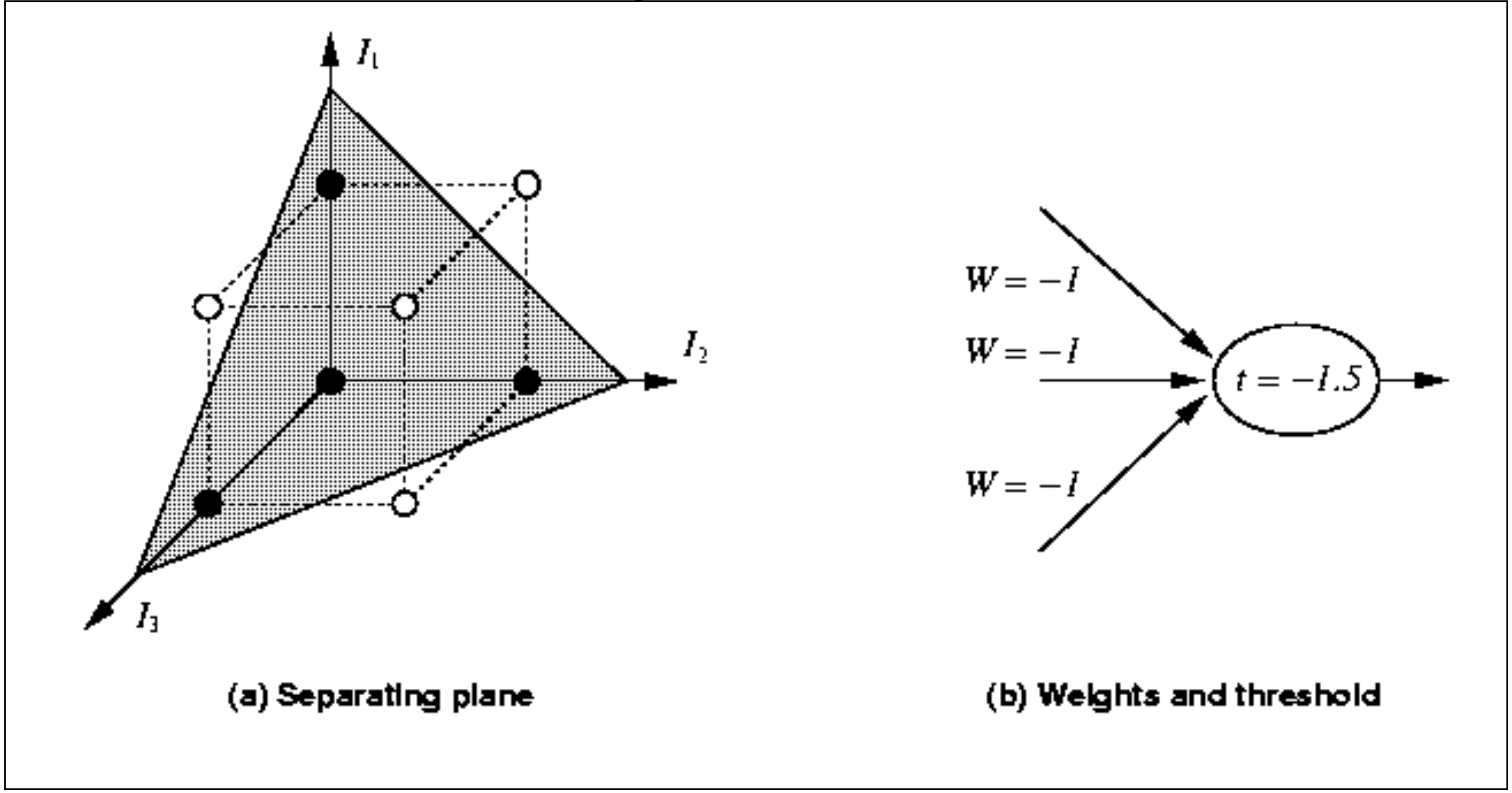
What can Perceptrons Represent?





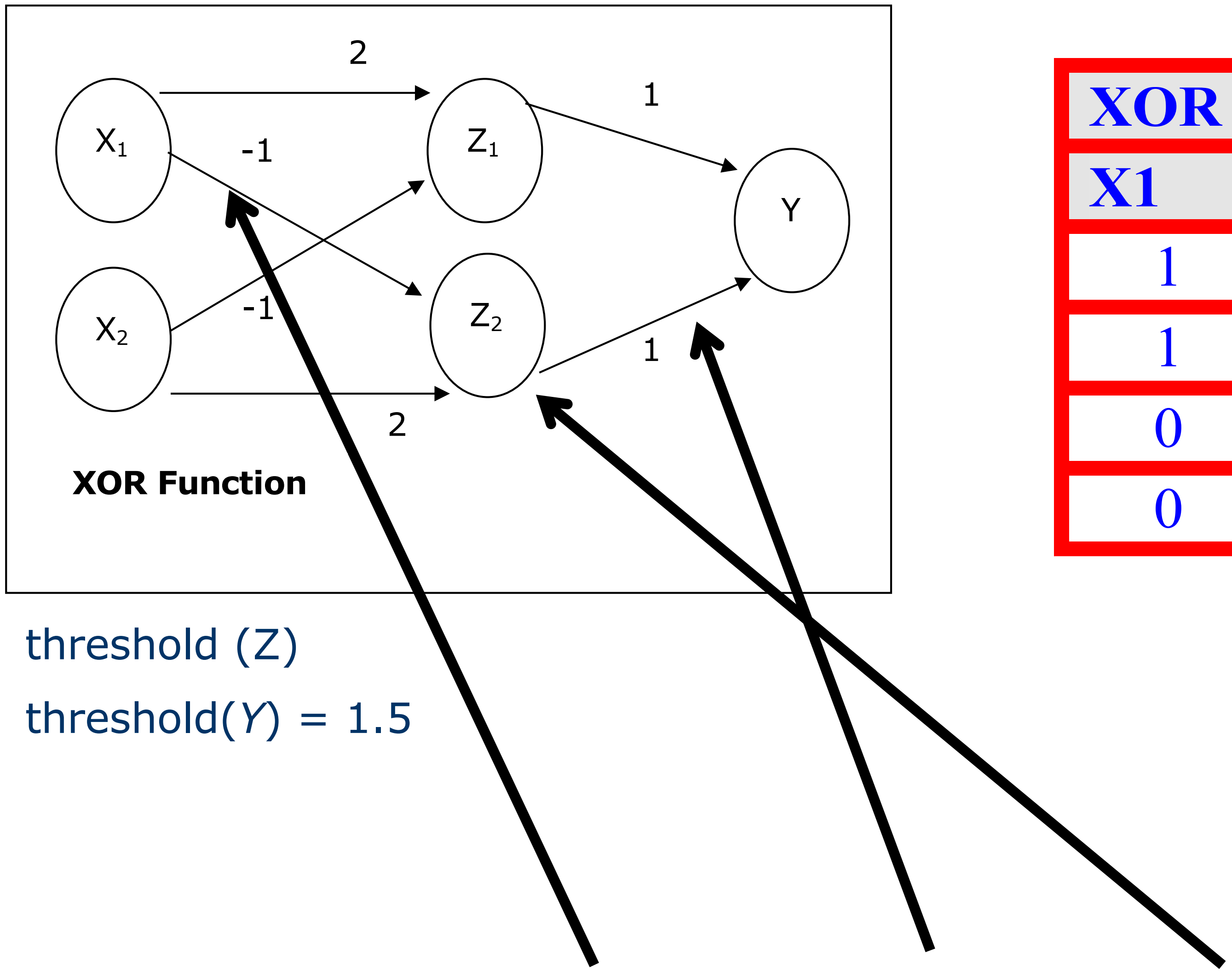
- Only functions which can be separated in this way (*linearly separable)* can be represented by a perceptron
- The dimensionality of function space is equal to the number of inputs, which will typically be higher than 2!
- Separation in this case is by a *hyperplane of* (dimensionality – 1)

What can Perceptrons Represent?



Linear Separability is also possible in more than 3 dimensions – but it is harder to visualise

Neural Networks - XOR



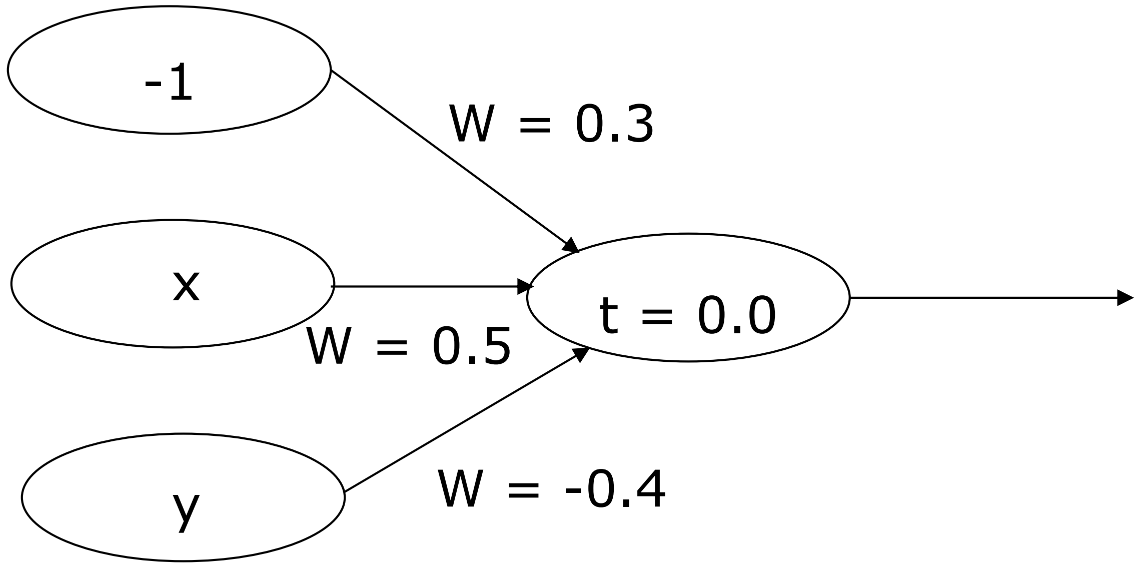
XOR		
X1	**X2**	**Y**
1	1	0
1	0	1
0	1	1
0	0	0

threshold (Z)

threshold(Y) = 1.5

X_1 XOR X_2 = (X_1 AND NOT X_2) OR (X_2 AND NOT X_1)

Training a Perceptron



I_1	I_2	I_3	Summation	Output
-1	0	0	(-1*0.3) + (0*0.5) + (0*-0.4) = -0.3	0
-1	0	1	(-1*0.3) + (0*0.5) + (1*-0.4) = -0.7	0
-1	1	0	(-1*0.3) + (1*0.5) + (0*-0.4) = 0.2	1
-1	1	1	(-1*0.3) + (1*0.5) + (1*-0.4) = -0.2	0

Learning

```
While epoch produces an error
    Present network with next inputs
      from epoch
    Err = T – O
    If |Err| > 0 then
        $W_j = W_j + LR * I_j * Err$
    End If
End While
```

Learning

```
While epoch produces an error
    Present network with next inputs
      from epoch
    Err = T – O
    If |Err| > 0 then
        $W_j = W_j + LR * I_j * Err$
    End If
End While
```

Epoch : Presentation of the entire training set to the neural network.
In the case of the AND function an epoch consists of four sets of inputs being presented to the network (i.e. [0,0], [0,1], [1,0], [1,1])

Learning

```
While epoch produces an error
    Present network with next inputs
      from epoch
    Err = T – O
    If |Err| > 0 then
        $W_j = W_j + LR * I_j * Err$
    End If
End While
```

Training Value, T : When we are training a network we not only present it with the input but also with a value that we require the network to produce. For example, if we present the network with [1,1] for the AND function the training value will be 1

Output from Neuron, O : The output value from the neuron

Learning

```
While epoch produces an error
    Present network with next inputs
      from epoch
    Err = T – O
    If |Err| > 0 then
        $W_j = W_j + LR * I_j * Err$
    End If
End While
```

Ij : Inputs being presented to the neuron

Wj : Weight from input neuron (I_j) to the output neuron

LR : The learning rate. This dictates how quickly the network converges. It is set by a matter of experimentation. It is typically 0.1

Error, Err : It is the amount by which the value output by the network differs from the training value. For example, if we required the network to output 0 and it output a 1, then Err = -1

Example

- A perceptron with two inputs, is to learn the AND function below:

x_1	x_2	T
0	0	0
0	1	0
1	0	0
1	1	1

  - Let α=0.25 be the learning rate, initial weights and threshold be 0, 0.4 and 0.3 respectively, T the training output and O the actual output, the weights and threshold should be adjusted as below:

$$\delta w_i = \alpha(T-O)I_i \; ; \qquad \delta\theta = -\,\alpha(T-O)$$

where I_i is the input and is always equal to -1 for the threshold

Perceptron Learning (1)

w_1	w_2	θ	x_1	x_2	a	O	T	α(T-O)	δw_1	δw_2	$\delta\theta$
0.0	0.4	0.3	0	0	0	0	0	0	0	0	0
0.0	0.4	0.3	0	1	0.4	1	0	-0.25	0	-0.25	0.25
0.0	0.15	0.55	1	0	0	0	0	0	0	0	0
0.0	0.15	0.55	1	1	0.15	0	1	0.25	0.25	0.25	-0.25
0.25	0.4	0.3	0	0	0	0	0	0	0	0	0
0.25	0.4	0.3	0	1	0.4	1	0	-0.25	0	-0.25	0.25
0.25	0.15	0.55	1	0	0.15	0	0	0	0	0	0
0.25	0.15	0.55	1	1	0.4	0	1	0.25	0.25	0.25	-0.25
0.5	0.4	0.3	0	0	0	0	0	0	0	0	0
0.5	0.4	0.3	0	1	0.4	1	0	-0.25	0	-0.25	0.25
0.5	0.15	0.55	1	0	0.5	0	0	0	0	0	0
0.5	0.15	0.55	1	1	0.65	1	1	0	0	0	0

Perceptron Learning (2)

w_1	w_2	θ	x_1	x_2	a	O	T	α(T-O)	δw_1	δw_2	$\delta\theta$
0.5	0.15	0.55	0	0	0	0	0	0	0	0	0
0.5	0.15	0.55	0	1	0.15	0	0	0	0	0	0
0.5	0.15	0.55	1	0	0.5	0	0	0	0	0	0
0.5	0.15	0.55	1	1	0.65	1	1	0	0	0	0

$a = \sum_i w_i x_i$ and θ = threshold

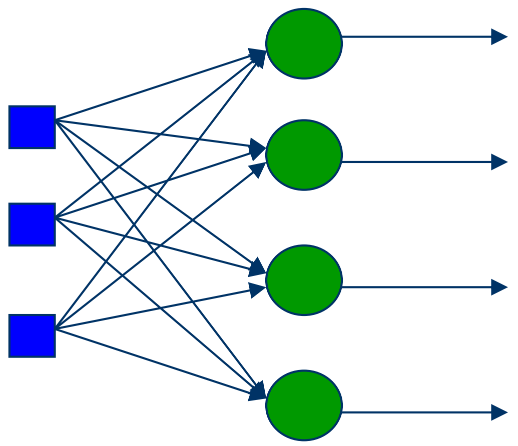
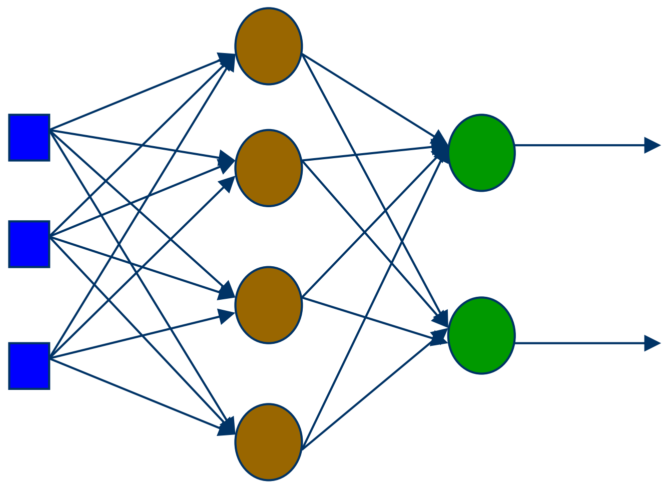
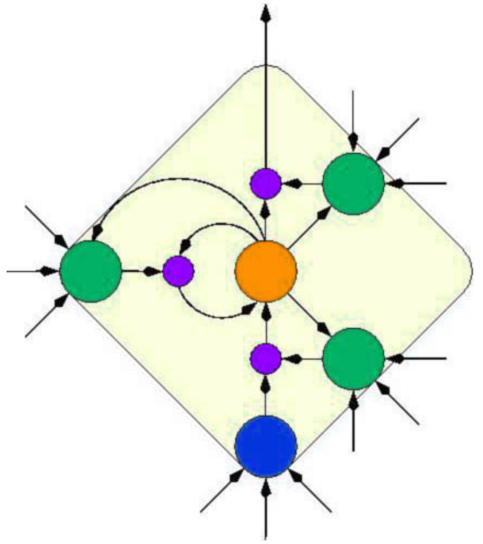
Building a Neural Network

There are many forms of neural networks. Most operate by passing neural 'activations' through a network of connected neurons.

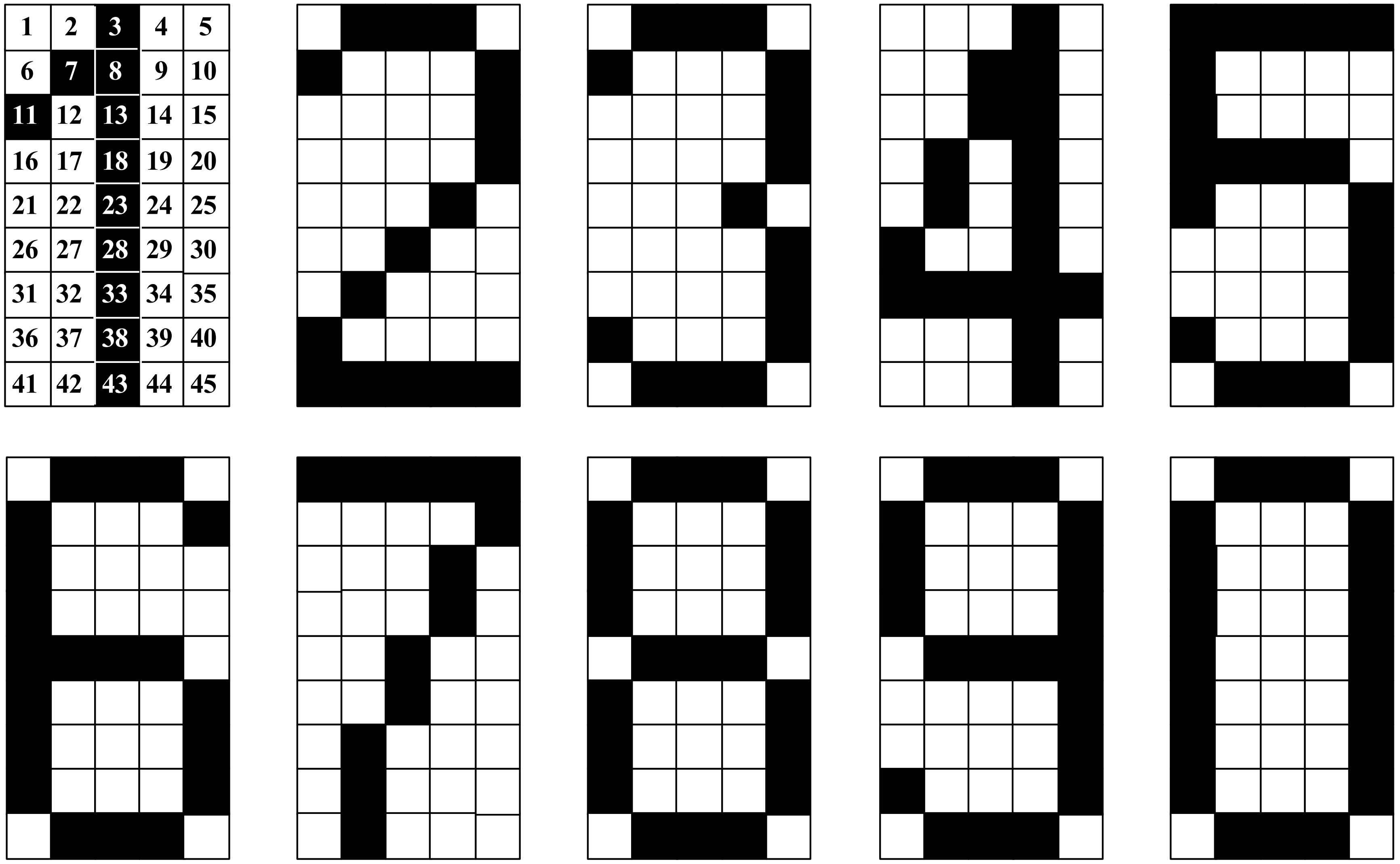
1.“Select Structure”: design the way that the neurons are interconnected

- three main classes of network architectures
  - ❖ single-layer feed-forward
  - ❖ multi-layer feed-forward
  - ❖ recurrent

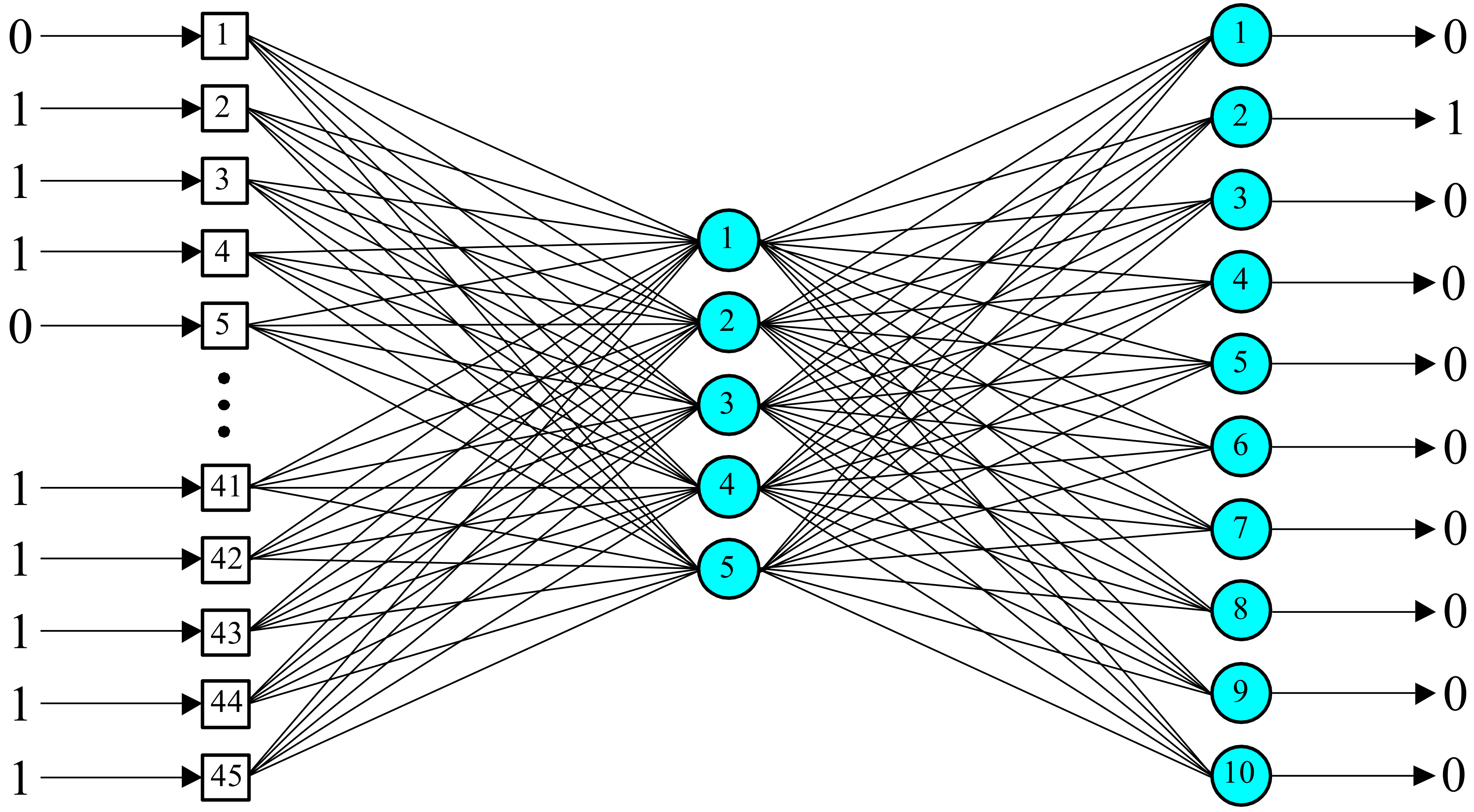
$\left.\begin{matrix} \text{single-layer feed-forward} \\ \text{multi-layer feed-forward} \end{matrix}\right\}$ neurons are organised in acyclic layers

Bit Maps for Digit Recognition

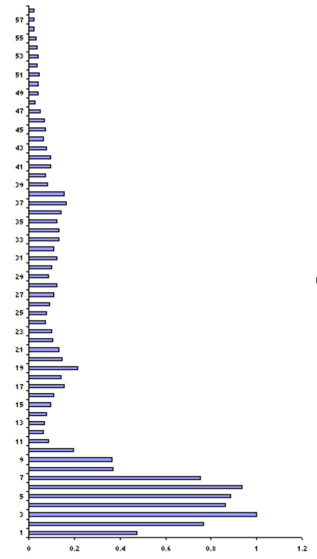


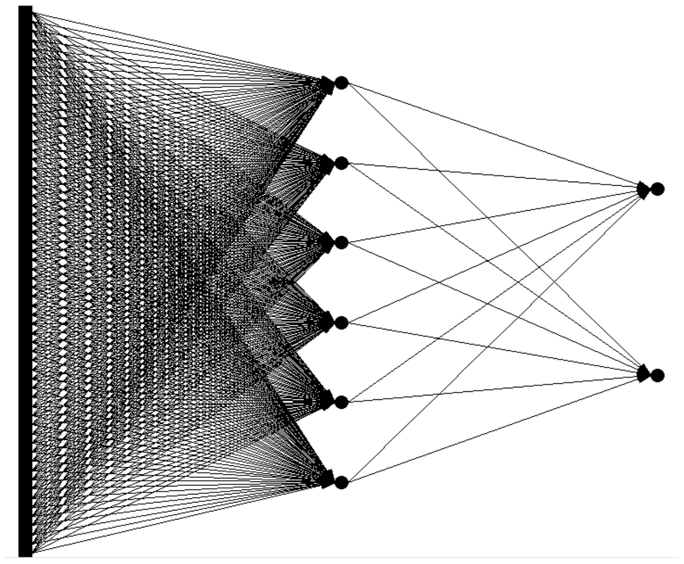
Neural Network for Printed Digit Recognition



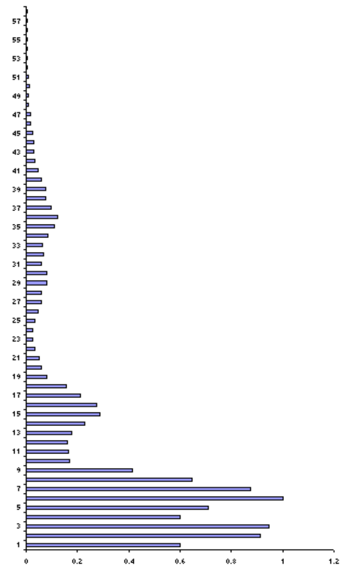
Voice Recognition

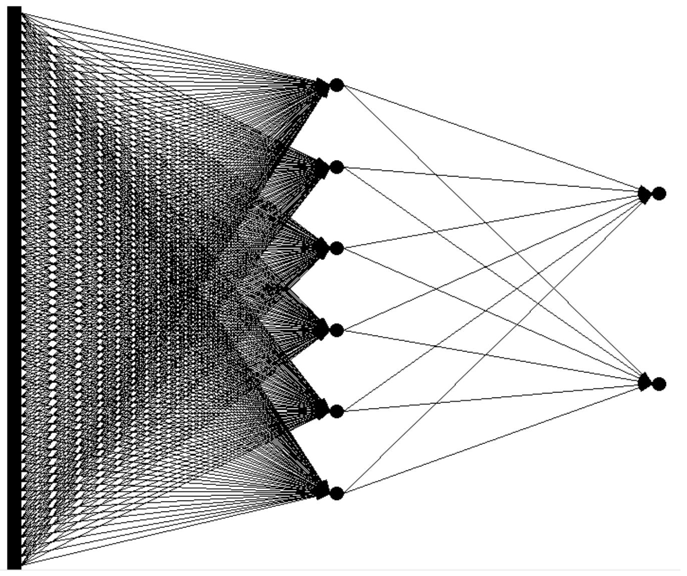
Steve





David





Network Architecture

Feedforward Network

60 inputs (one for each frequency bin);

6 hidden;

2 output (0-1 for "Steve", 1-0 for "David")

Summary of Neural Network

- Definitions
- Modelling Functions
  - weights
  - Activation functions
- Simple Networks
  - perceptron
- Network Architecture
- Learning Algorithm
  - delta rule

Acknowledgements

Most of the lecture slides are adapted from the same module taught in Nottingham UK
by
Professor Graham Kendall,
Dr. Rong Qu
and
Dr. Andrew Parker