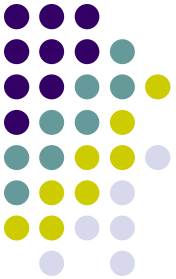
Birkbeck
UNIVERSITY OF LONDON



(Concepts of) Machine Learning- Lab 1

George Magoulas with

Cosmin Stamate (MSc Intelligent Technologies from Birkbeck) is working towards a PhD on machine learning and deep networks for psycho-physiological signals processing, modelling, and classification.

Michal Grochmal (MSc Intelligent Technologies from Birkbeck) is working towards a PhD on bio-inspired machine learning, combining deep learning and statistical mechanics.

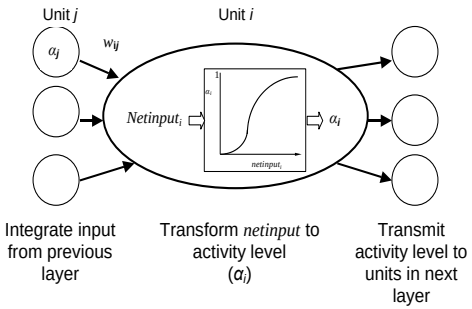
Activities

- Matlab: basic operations, matrices, plots and data management
- Neurons, operation and learning
- Perceptron and Multilayer Networks

What is a neural network?

Neurons, operation and

Learning by weight change: If the response of an output unit is incorrect then the network can be changed so that it is more likely to produce the correct response the next time that the stimulus is presented. This is achieved by changing the connection weights.



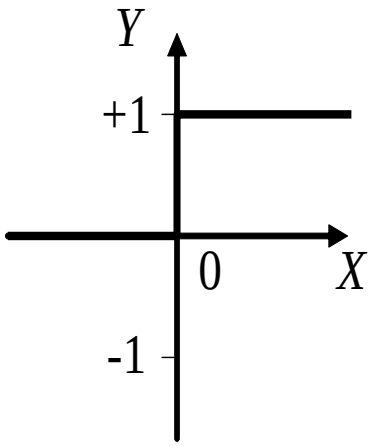
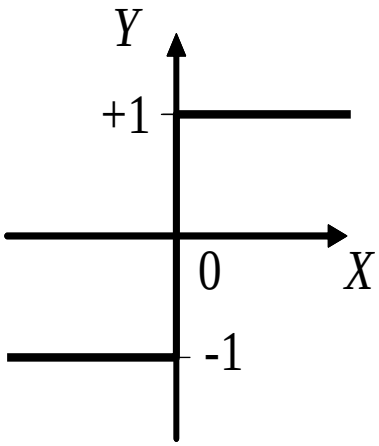
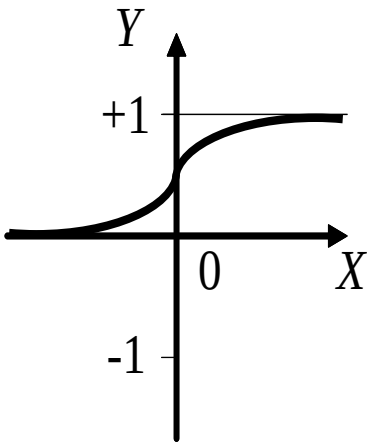
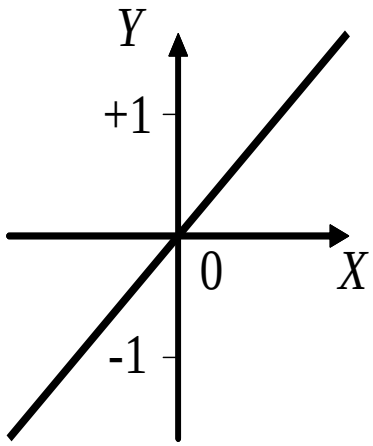
$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij}$$

$$\Delta w_{ij} = -\eta \frac{dE}{dw_{ij}} = -\eta \frac{d\left[a_i^{desired} - a_i^{obtained}\right]^2}{dw_{ij}}$$

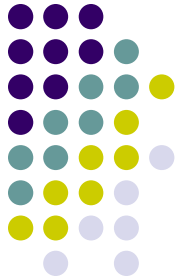
Δw_{ij} is the change in the connection weight w_{ij} from unit *j* to unit *i*

Bias: There is one special input unit, which is called bias unit. The bias unit receives no input itself, and its activity is always set at +1. The weight from the bias unit to the unit of interest can be positive or negative and changes just like any other weight during learning.

Activation functions of a neuron

Step function	*Sign function*	*Sigmoid function*	*Linear function*
			
$Y^{step} = \begin{cases} 1, \text{ if } X \geq 0 \\ 0, \text{ if } X < 0 \end{cases}$	$Y^{sign} = \begin{cases} +1, \text{if } X \geq 0 \\ -1, \text{if } X < 0 \end{cases}$	$Y^{sigmoid} = \dfrac{1}{1+e^{-X}}$	$Y^{linear} = X$

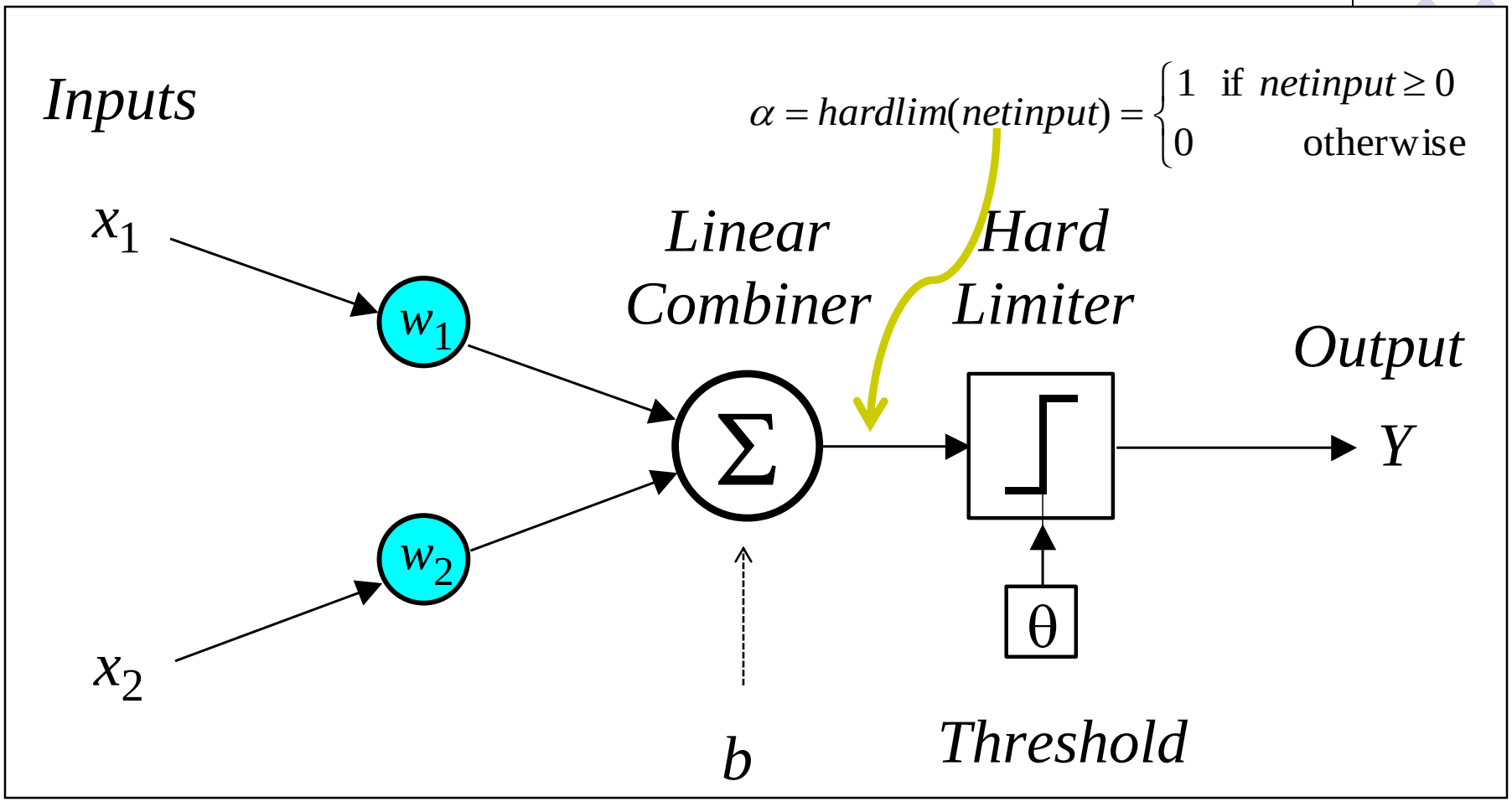
Perceptron and Multilayer Networks



The perceptron and the perceptron rule: can a single neuron learn a task?

- In 1958, **Frank Rosenblatt** introduced a training algorithm that provided the first procedure for training a simple ANN: a **perceptron**.
- The perceptron is the simplest form of a neural network. It consists of a single neuron with *adjustable* synaptic weights and a *hard limiter*.

Single-layer two-input perceptron



The node divides the input space into two regions because it can only be in one of two states (i.e. 1 or 0)

(1) Assume a node with only two inputs : $w_{11}x_1 + w_{12}x_2 = 0$

(2) Assume the node has a bias term *b:* $w_{11}x_1 + w_{12}x_2 + b = 0$

(3) Assume w_{11}=1; w_{12}=1; b=-1

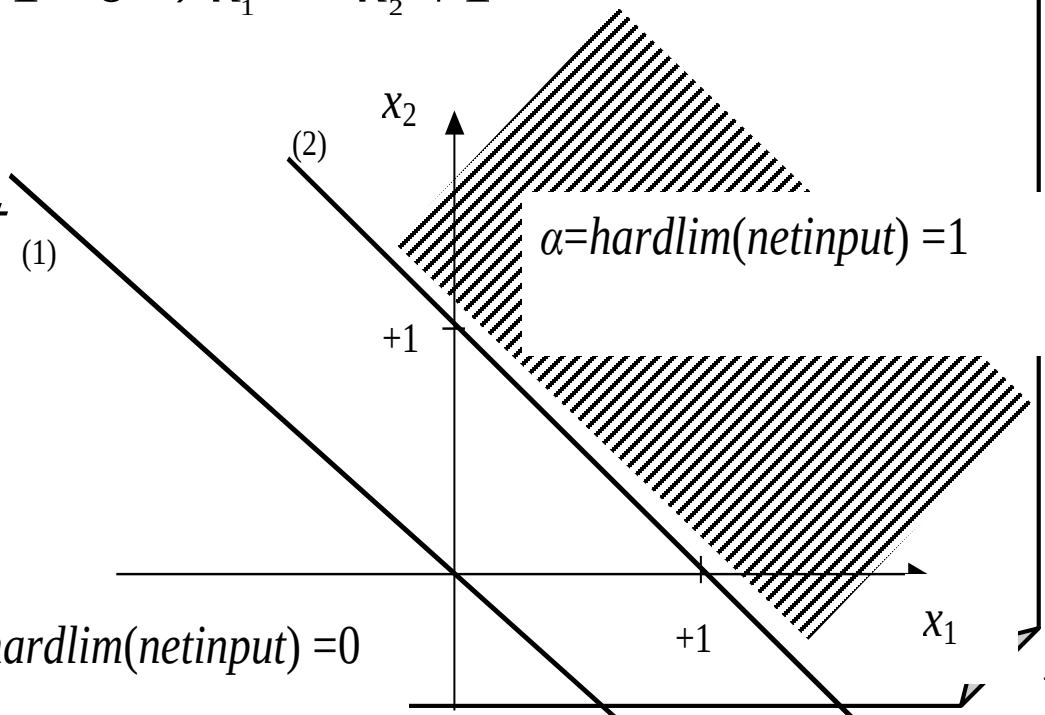
Then:

These are line equations→

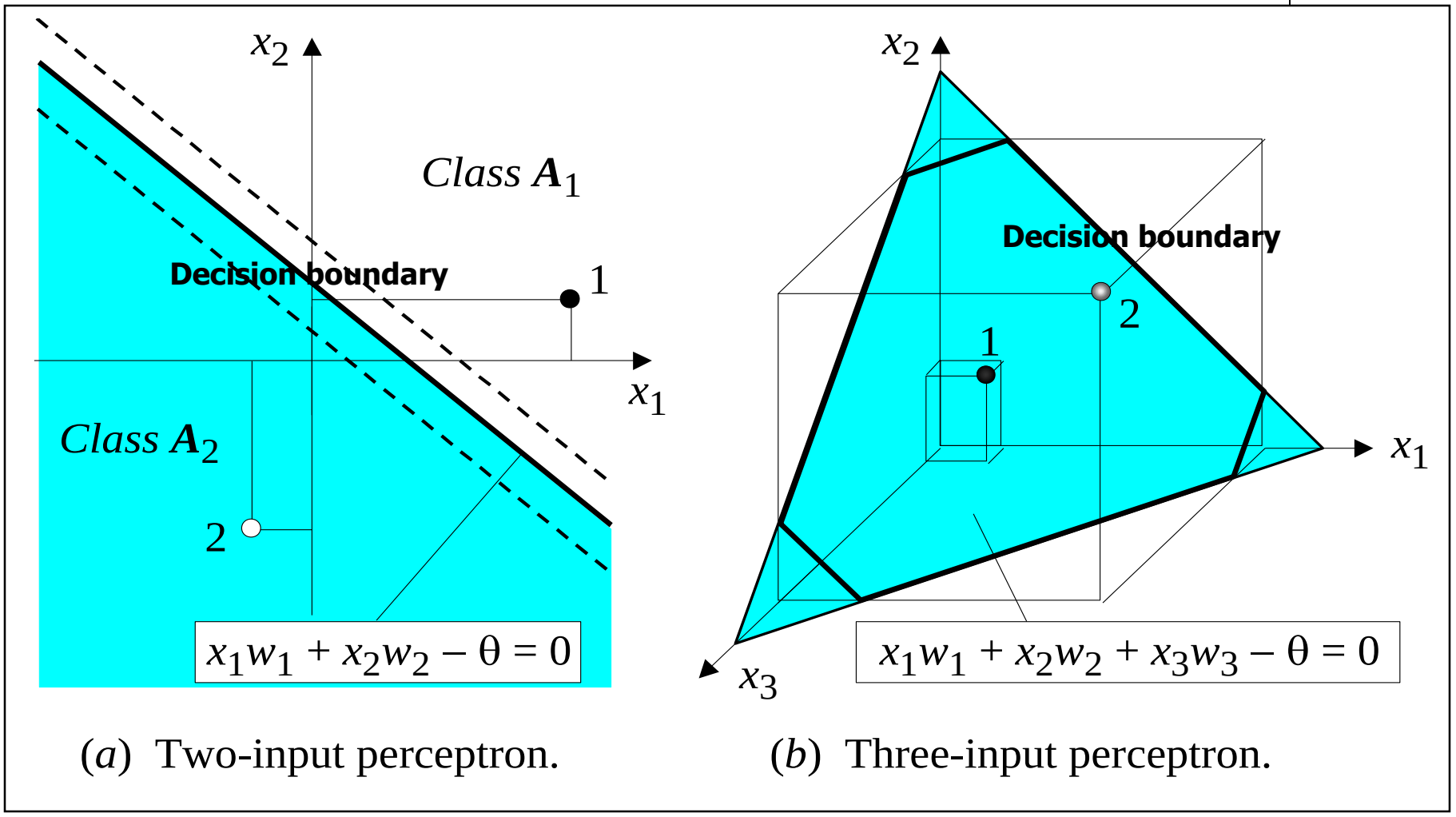
$$x_1 + x_2 = 0 \Rightarrow x_1 = -x_2$$ (1) No bias

$$x_1 + x_2 - 1 = 0 \Rightarrow x_1 = -x_2 + 1$$ (2) With a bias

The line defines the boundary between regions where the input pattern produces a positive response (output) and regions where the response will be negative or zero. A line of this kind is also called *decision boundary*



Linear separability in the perceptrons



(*a*) Two-input perceptron. (*b*) Three-input perceptron.

The perceptron learning rule

$$w_i(p+1) = w_i(p) + \alpha \cdot x_i(p) \cdot e(p)$$

where $p = 1, 2, 3, \ldots$

α is the **learning rate**, a positive constant less than unity.

The perceptron learning rule was first proposed by **Rosenblatt** in 1960. Using this rule we can derive the perceptron training algorithm for classification tasks.

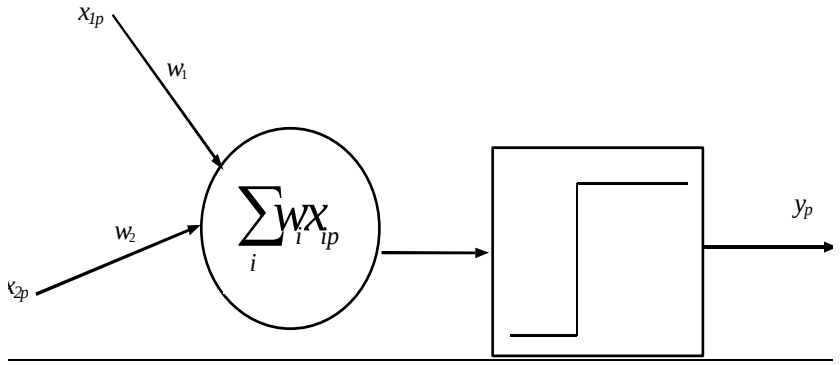
Numeric example: A single node with two weights

A single node with two weights is trained using the following rule:

$$w_i^{t+1} = w_i^t + \sum_{p=1}^{P} x_{ip} y_p, \quad i = 1,2,$$

where t denotes iterations, P is the number of patterns in the training set and the inputs, x_{ip}, and outputs, y_p, for each pattern, p=1,…,4, are shown in the following truth table

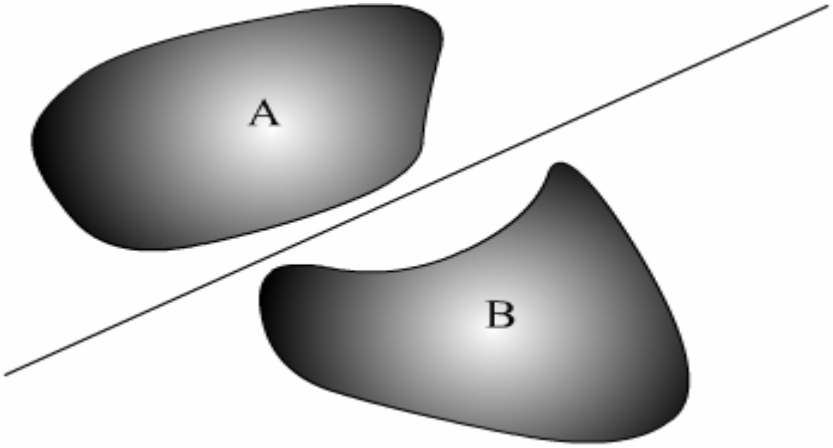
p	x_{1p}	x_{2p}	y_p
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1



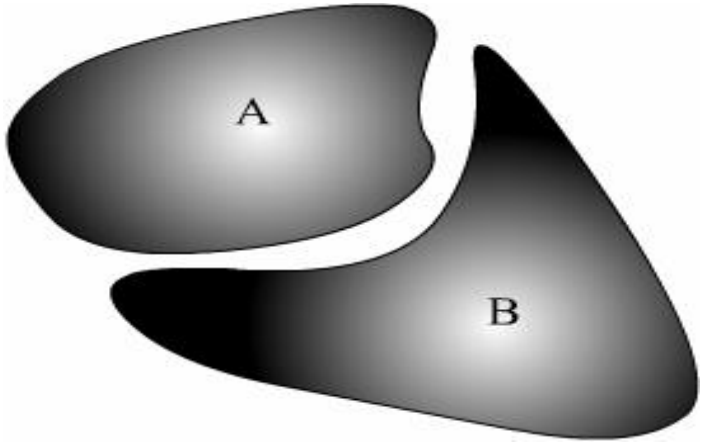
The weights are all zero at the start, i.e. $w_i^0 = 0$, for all $i = 1,2$. The node is trained by presenting it with input and output pairs in the same order as the truth table, starting form the top, then the first input pattern that is used (p=1) is [0, 0] and the corresponding output pattern is 0.

Multilayer networks

Non linear separable problems: Training patterns belonging to one output class cannot be separated from training patterns belonging to another class by a straight line, plane or hyperplane.

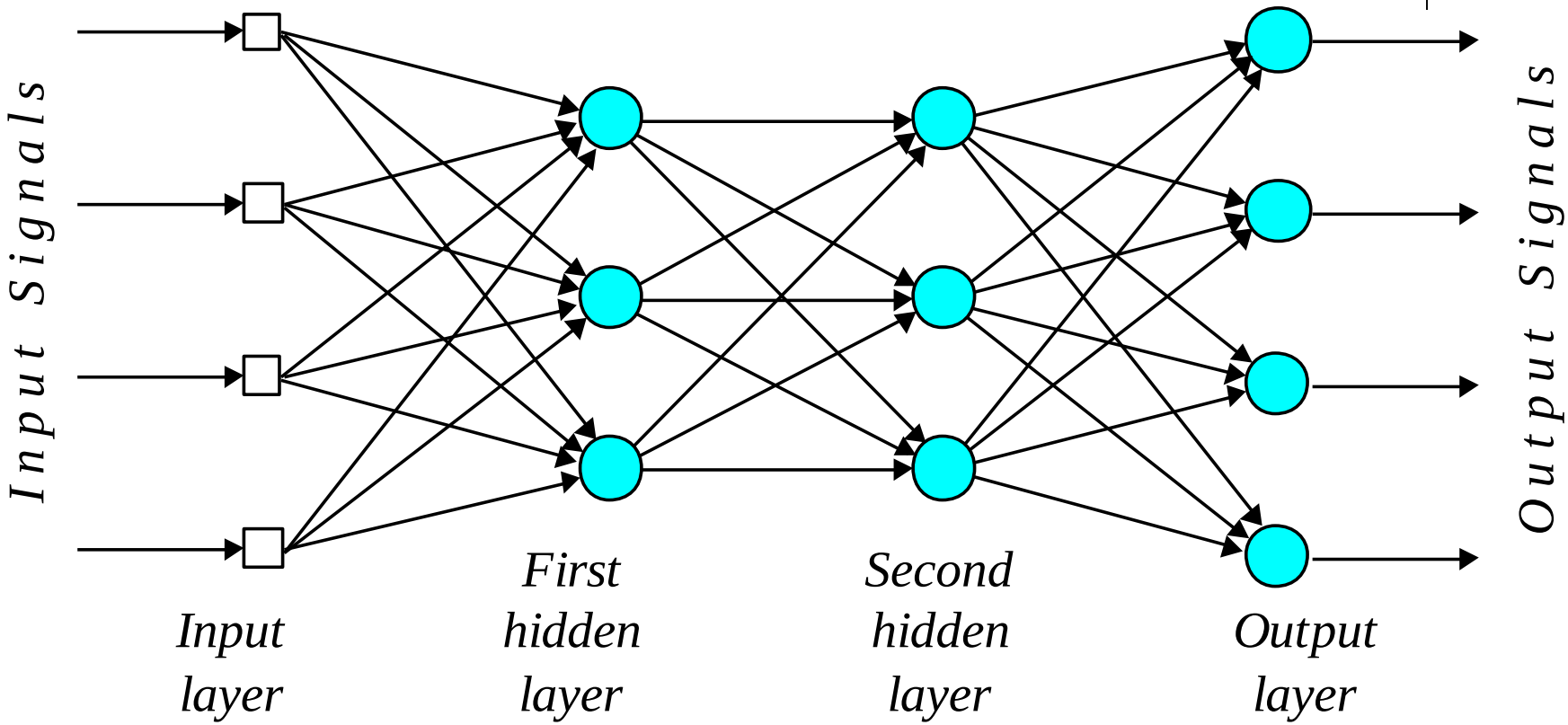


Linear separable



Nonlinear separable

Multilayer perceptron with two (or more) hidden layers



This form of training is called **Supervised learning:** The response that the backpropagation network is required to learn is presented to the network during training. The desired response of the network acts as an explicit teacher signal.