Machine Learning (521289S)

Multilayer Perceptrons and Artificial Neural Networks

M.Sc. Antti Isosalo

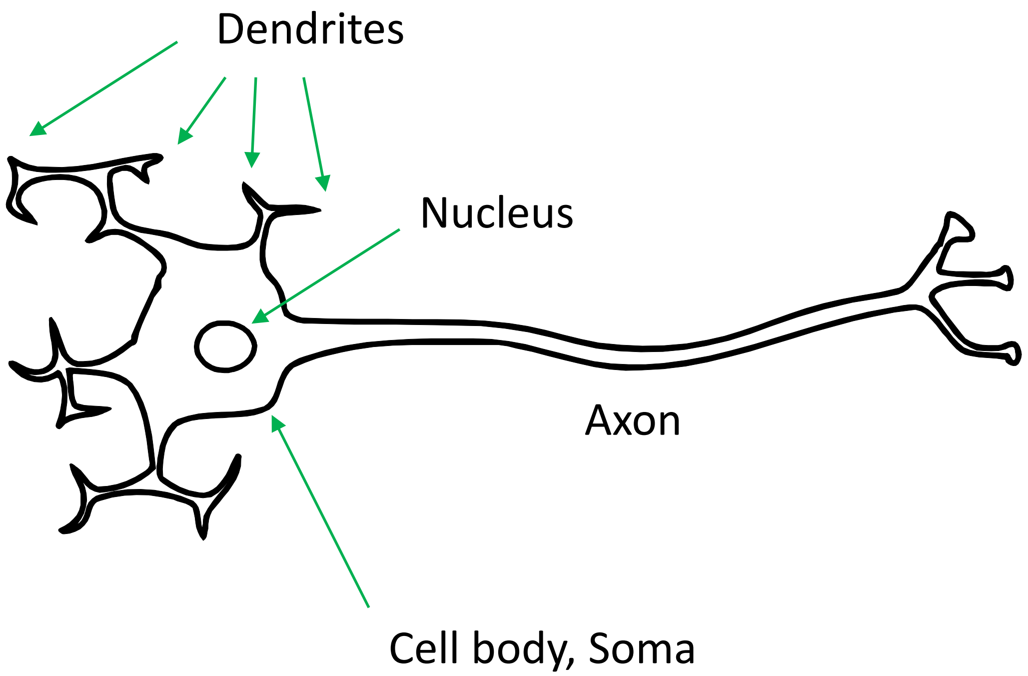
Physiological Signal Analysis Team
Center for Machine Vision and Signal Analysis (CMVS)
University of Oulu

Neural Networks and Artificial Neural Networks

- Humans (and other animals) process information with *neural networks*.
- These neural networks are formed (with large animals) from billions of nerve cells, *neurons*, exchanging brief electrical pulses called action potentials.
- Computer algorithms that simulate these biological structures are formally called *Artificial Neural Networks* (ANN) to differentiate them from the “squishy things” of the animal nervous system.

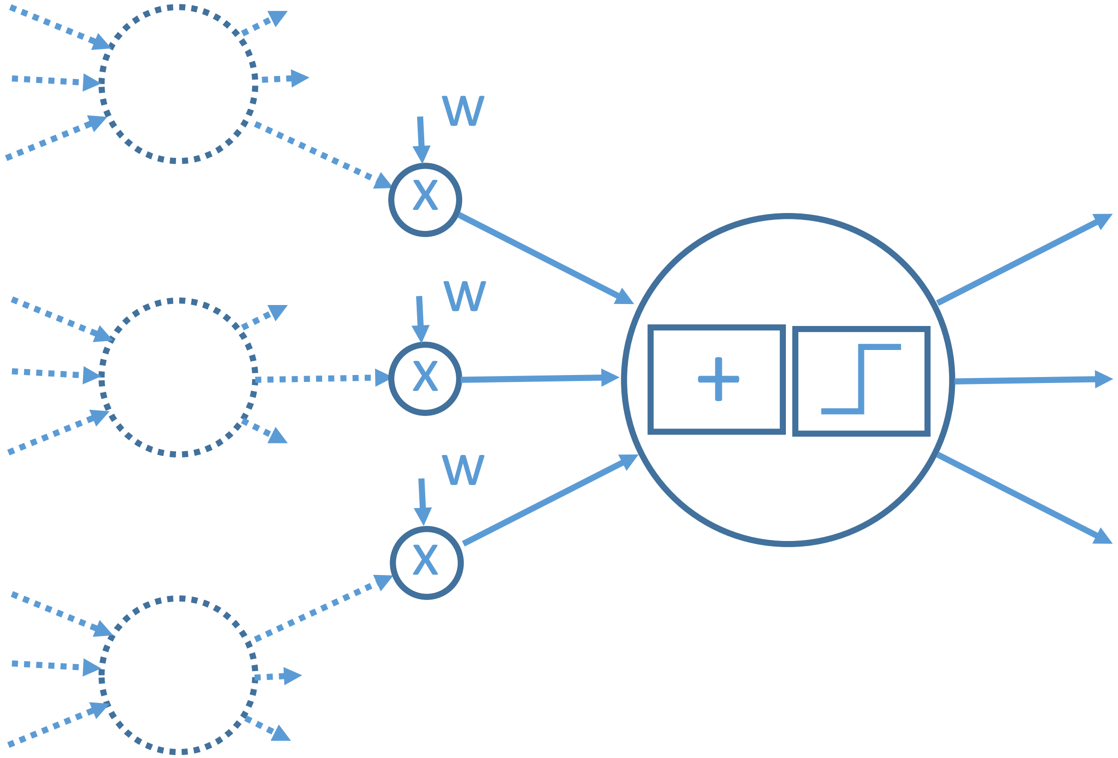
Biological Neuron

- Biological neuron comprises of, for example, an *axon* connecting it to other neurons, *dendrites* connecting to axons of other neurons; these are the ones we are interested in (here).
- Between axons and dendrites there is a small gap called synapse related to learning.
- Neuron is triggered when the stimulus of all inputs exceed a certain threshold, resulting a pulse along the axon.



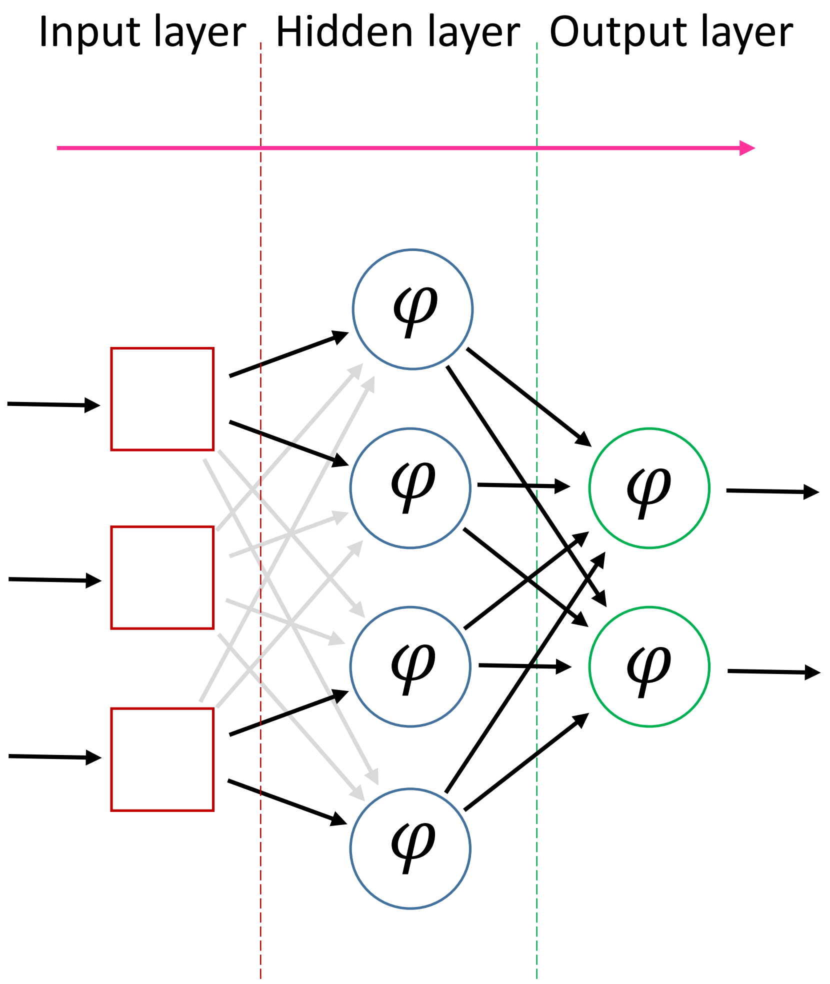
Artificial Neuron

- Artificial (simulated) neurons are usually depicted as nodes which are connected to other nodes with links (arrows in the figure) corresponding to axon-synapse-dendrite connections.
- Only some of the features of biological neurons are included in the artificial neurons.
- Artificial neuron is comprised of multipliers, summation unit and thresholding element.
    - First model of neurone was introduced 1943
- We usually refer to these artificial neurons as *Perceptrons*.



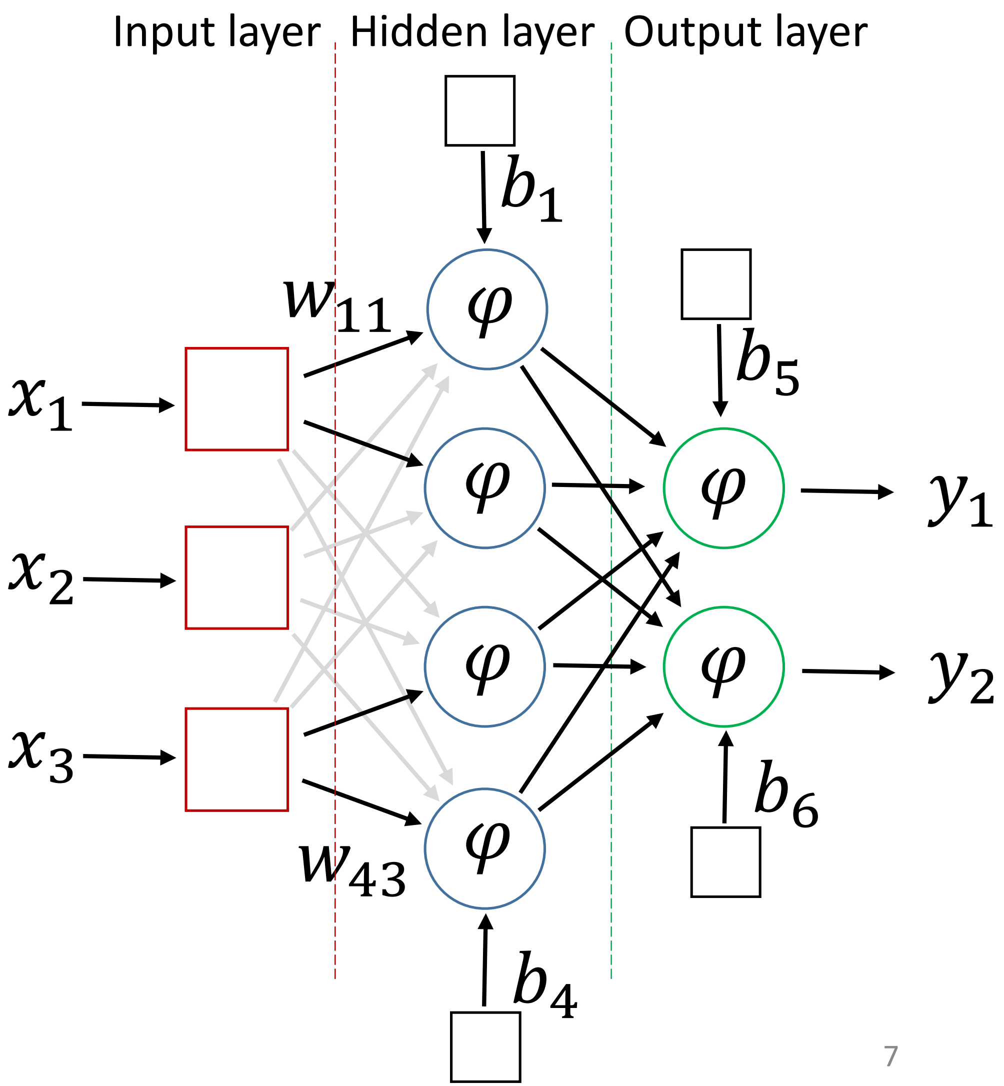
Multilayer Perceptron: Input, Hidden and Output Layer

- This is a general multilayer *feedforward* neural network
- We have distinct layers and the information flows only from the input to the output, i.e. it is *acyclic*



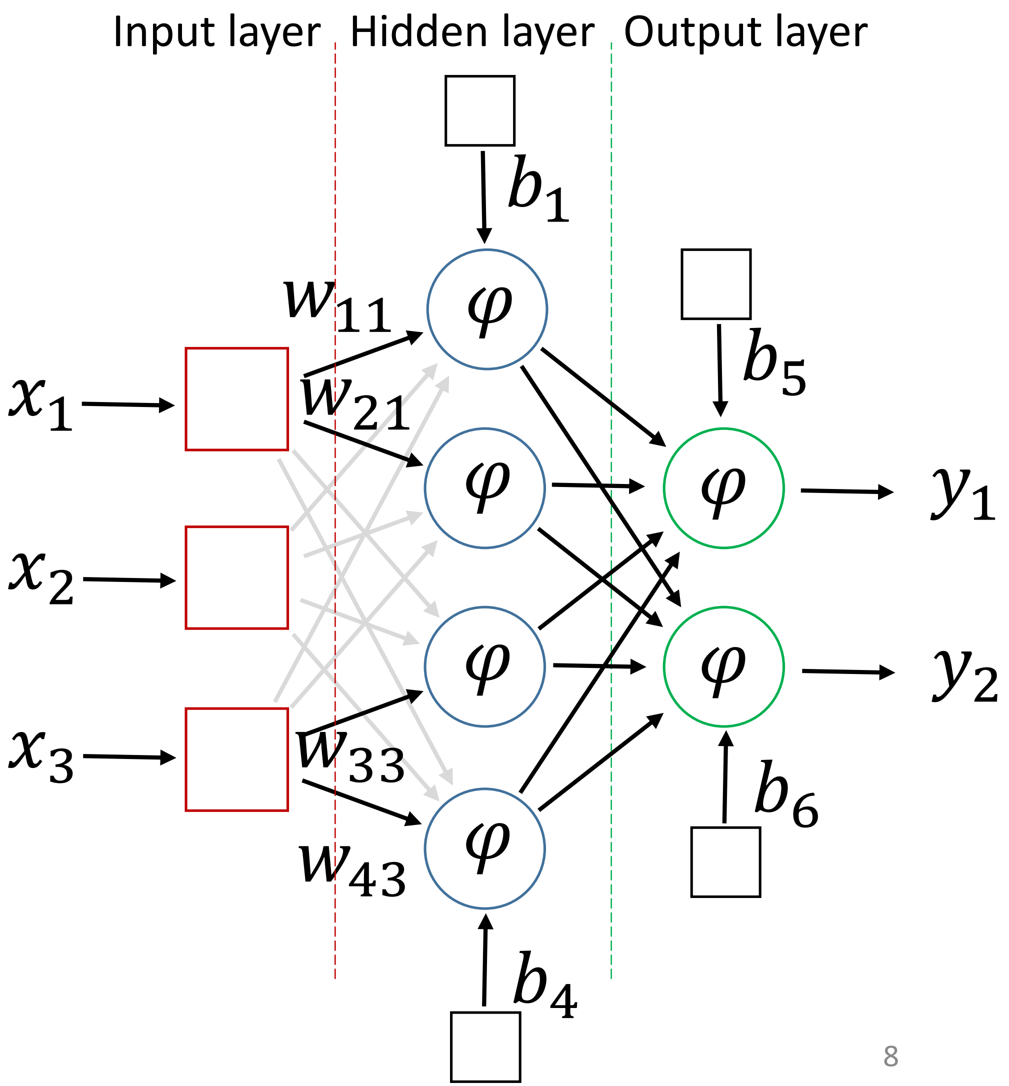
Multilayer Perceptron: Input, Hidden and Output Layer (cont.)

- We have here two layers.
  - We can have one or more hidden layers.
  - Some authors count also the input layer as one layer.
- Output layer is visible to the system.
- Hidden layer is visible only to neurons of output layer and the input layer nodes.
- The hidden layer transforms the inputs into something that the output layer can use.
- The output layer scales the hidden layer "products" into a scale that the output needs to be.
- There are usually as many outputs as there are classes.



Multilayer Perceptron: Input, Hidden and Output Layer

- In the schematic figure
  - input nodes (x_j) are described as squares (no processing done, signal is only copied to several nodes),
  - actual neurons as circles (processing done, activation function φ limits the output of the weighted sum of its inputs to a certain range),
  - bias unit (b_k in the picture) to shift the activation function to the left or right and
  - the values entering a node are multiplied by weights (predetermined numbers, w_{kj}).
  - Outputs are marked with y_k
  - Some of the symbols **are left undrawn for not overcomplicating the figure**.



Multilayer Perceptron: Input, Hidden and Output Layer (cont.)

- Output y_k of a neuron k can be described by writing

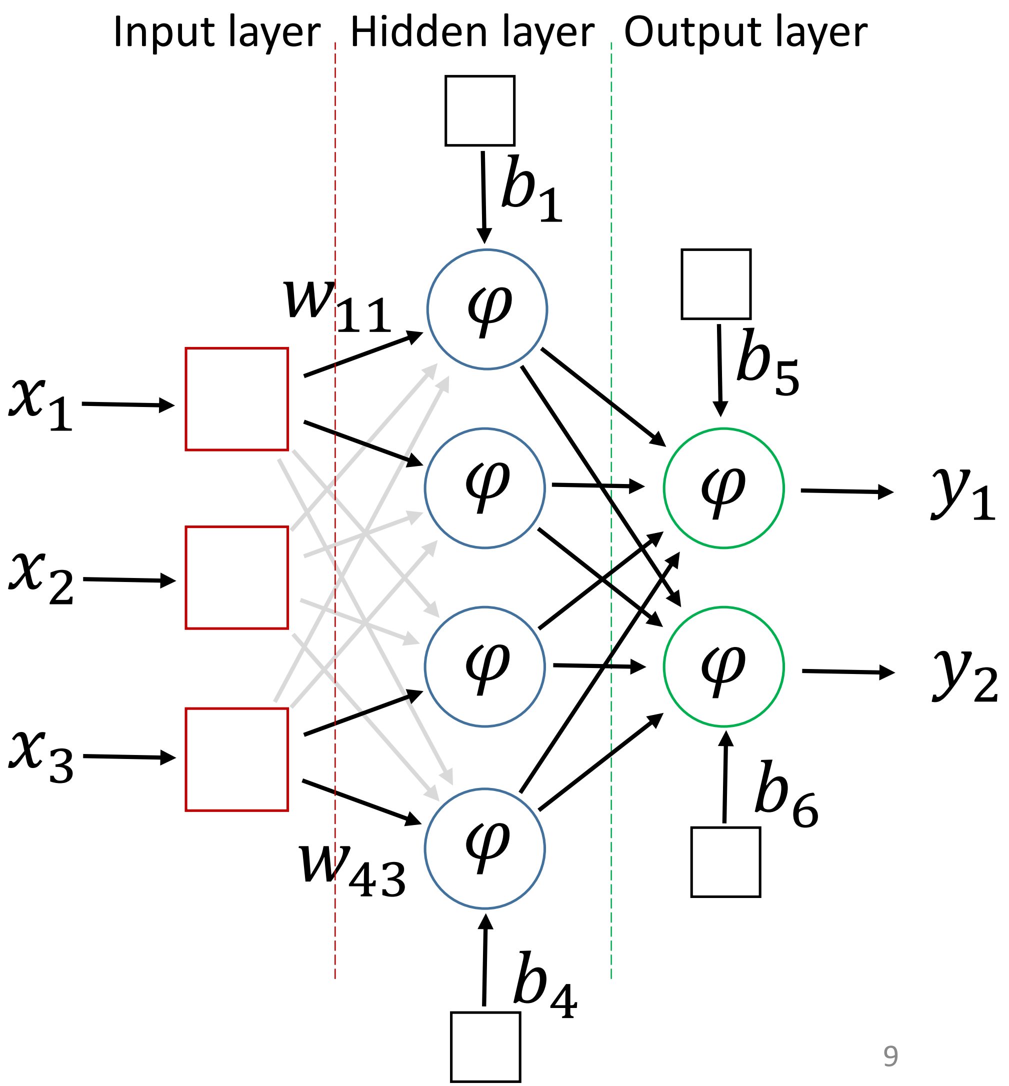
$$y_k = \varphi(\underbrace{b_k + u_k}_{=:v_k}) = \varphi\left(b_k + \sum_{j=1}^{m} w_{kj}x_j\right),$$

where $\varphi(.)$ is the so called *activation function*.

- Activation function can be, for example, a threshold function

$$\varphi(v_k) = \begin{cases} 1 & \text{if } v_k > 0 \\ 0 & \text{if } v_k \leq 0 \end{cases}.$$

- Hidden and output layer neurons can have the same structure (i.e., same activation function)
- Output layer neurons usually have linear activation function, but it is not a rule.



Training Neural Networks: Backpropagation

- The error *backpropagation* algorithm is a popular method of training artificial neural networks and used in conjunction with some *optimization method*, e.g. gradient descent method.
- The motivation for developing the backpropagation algorithm: to find a way to train a multilayered neural network so that the hidden layer(s) will contain the characteristics, features, common to certain patterns in the interactions of its nodes, and can then produce output for arbitrary input pattern that is the same or close to the desired one.
- This is accomplished by learning the weights for the network.
  - Thus, getting a representation for the knowledge that we model
- Some details of the backpropagation algorithm are found in Section 6.3 (Duda, Hart & Stork 2001, Section 6.8.7).
- The backpropagation algorithm is perhaps better explained in Haykin's book (Haykin 1999, Section 4.3. and 4.4.).

How to Determine the proper amount of Hidden Layers and Number of Nodes?

- A feedforward network *without* a hidden layer can only solve linearly separable problems.
- A feedforward network *with single hidden layer* can approximate accurately, for example, any continuous function by adding more nodes, see *Universal Approximation Theorem* (Haykin 1999, pp. 230-231).
- Every Boolean function can be represented exactly by a network with *two layers of nodes*.
- Networks with *two hidden layers* can represent functions with any kind of shape.
- If the patterns are well separated or linearly separable, then only few nodes are needed (Duda, Hart & Stork 2001, Section 6.8.7).
- If we have complex data and we add too few nodes we are not able to describe well enough the different characteristics of the data.
- When we have small amount of data and we add too many neurons, we risk *overfitting*.
- There are manly rules-of-thumb to determining the correct number neurons, see Section 6.8.7 (Duda, Hart & Stork 2001).

Literature

- Haykin, S. (1999). “Neural networks,” MacMillan College Publishing Company, 2nd ed.
- Duda, R., Hart, P. & Stork, D. (2001). “Pattern classification,” John Wiley & Sons Inc., 2nd ed.
- Rumelhart, D., Hinton, G. & Williams, R. (1986). “Learning representations by back-propagating errors,” Nature, 323(6088), pp. 533-536.