Machine Learning

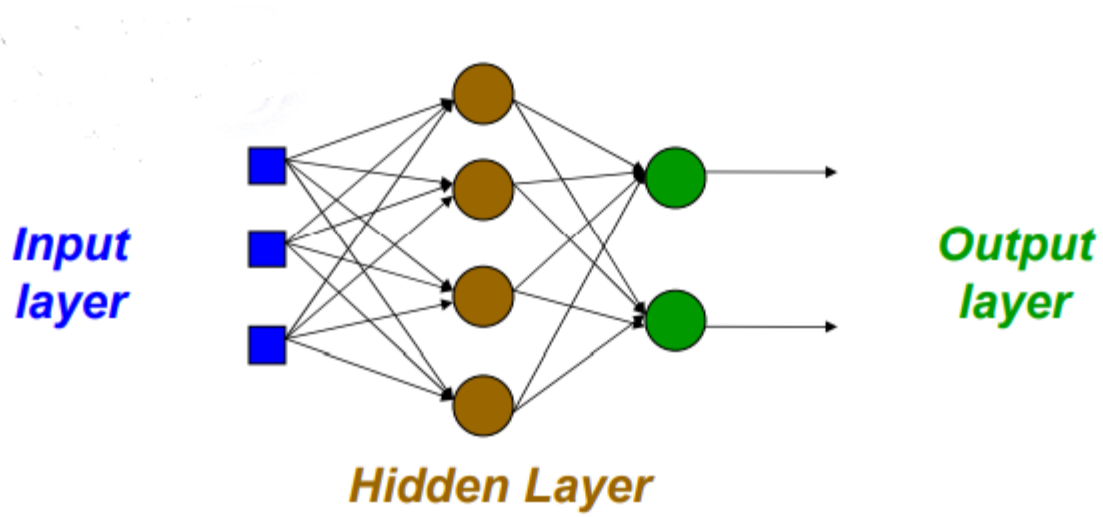
Multi layer feed-forward NN

Computer & Health Informatics Department

Introduction FNN

- We consider a more general network architecture: between the input and output layers there are hidden layers, as illustrated below.
- Hidden nodes do not directly receive inputs nor send outputs to the external environment.
- FNNs overcome the limitation of single-layer NN: they can handle non-linearly separable learning tasks.

FNN



XOR problem

- A typical example of non-linealy separable function is the XOR.
- This function takes two input arguments with values in $\{-1,1\}$ and returns one output in $\{-1,1\}$, as specified in the following table:

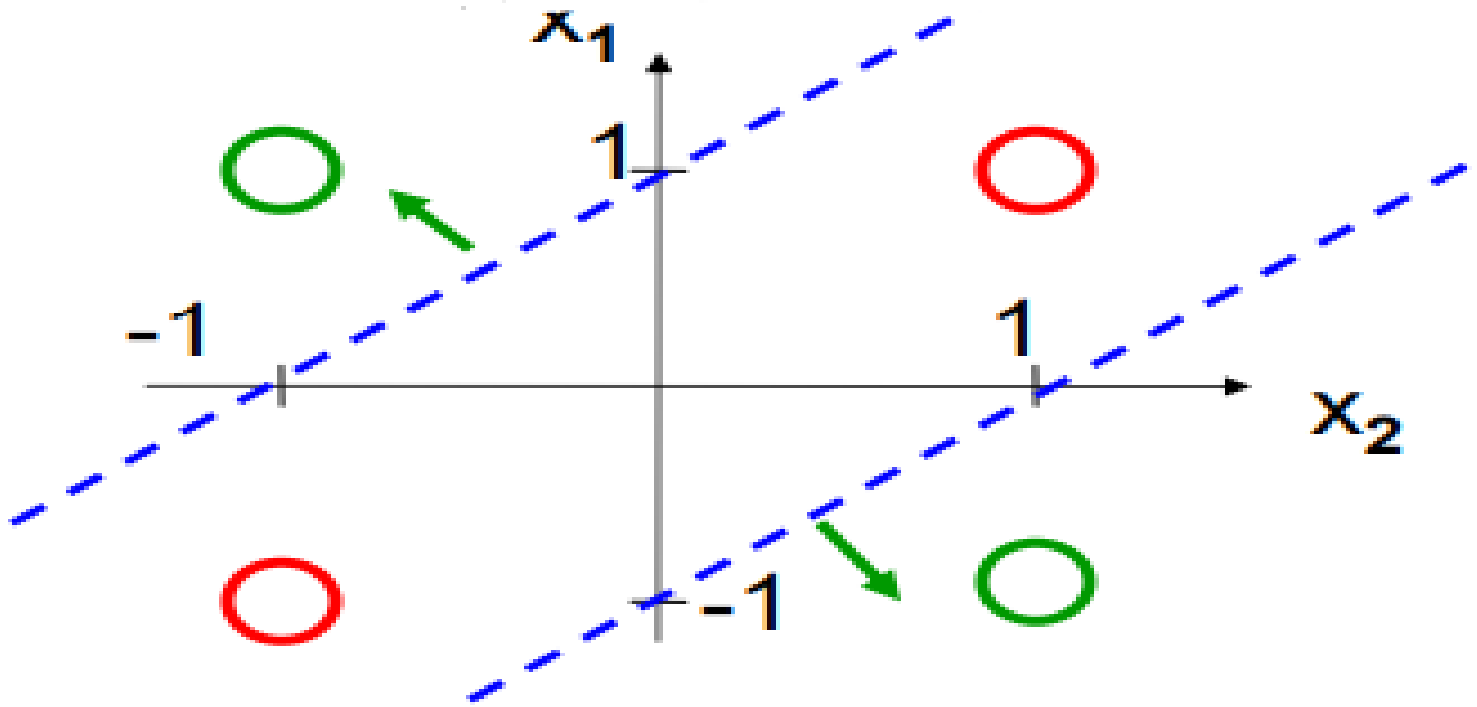
XOR problem

x_1	x_2	x_1 xor x_2
-1	-1	**-1**
-1	1	**1**
1	-1	**1**
1	1	**-1**

XOR problem

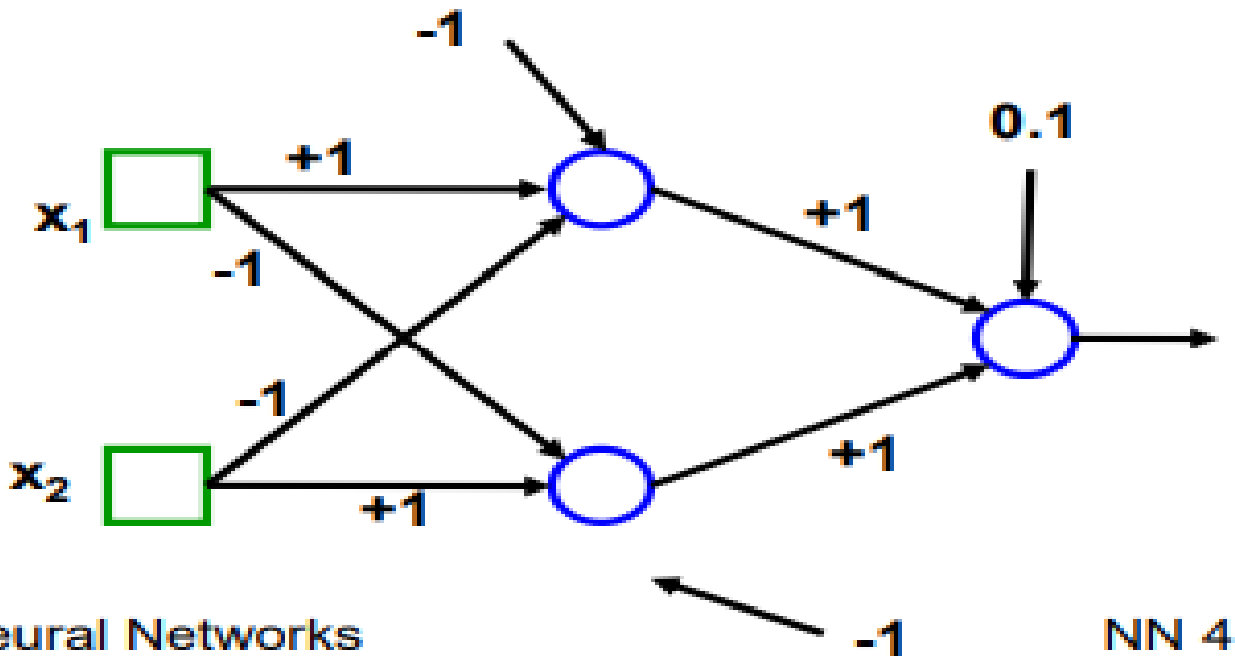
- If we think at -1 and 1 as encoding of the truth values false and true, respectively, then XOR computes the logical exclusive or, which yields true if and only if the two inputs have different truth values.
- In the following graph of the XOR, input pairs giving output equal to 1 and -1 are shown.
- These two classes cannot be separated using a line. We have to use two lines.

XOR problem



XOR problem

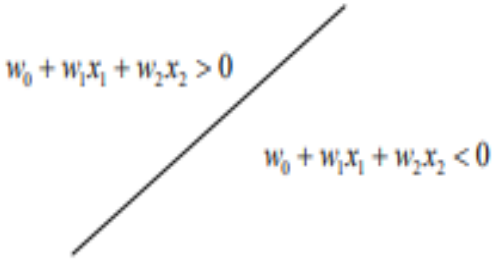
- The following NN with two hidden nodes realizes this non-linear separation, where each hidden node describes one of the two lines

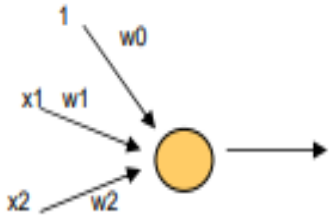


XOR problem

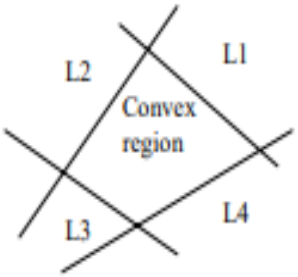
- This NN uses the sign activation function. The two arrows indicate the regions where the network output will be 1.
- The output node is used to combine the outputs of the two hidden nodes.

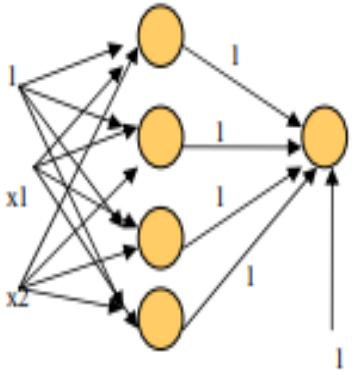
Types of decision regions





Network with a single node





One-hidden layer network that realizes the convex region: each hidden node realizes one of the lines bounding the convex region

Fnn Neuron Model

- The classical learning algorithm of FFNN is based on the gradient descent method.
- For this reason the activation function used in FFNN are continuous functions of the weights, differentiable everywhere.
- A typical activation function that can be viewed as a continuous approximation of the step (threshold) function is the Sigmoid Function.

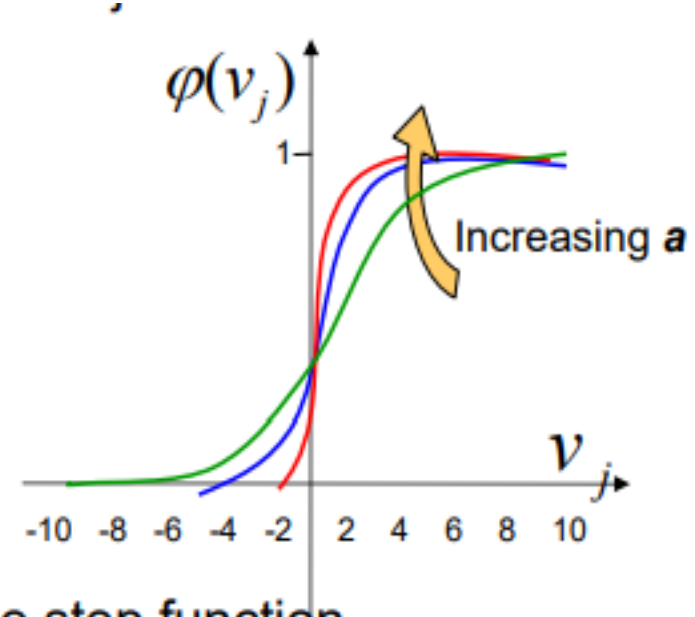
Fnn Neuron Model

- The activation function for node j is:

$$\varphi(v_j) = \frac{1}{1+e^{-av_j}} \quad \text{with} \quad a > 0$$

where $v_j = \sum_i w_{ji} y_i$

with w_{ji} weight of link from node i to node j and y_i output of node i



- when $a \to \infty$, φ 'becomes' the step function

Deep Neural Networks (multilayer perceptrons)

- When we use more than one layer/ level of perceptrons then such a network is called deep neural networks.
- The term 'Deep Learning' comes from here which refers to networks that have hidden layers.
- To solve the XOR problem we will go for a three-layer model rather than the two-layer model.

Deep Neural Networks (multilayer perceptrons)

- Here we will have the input layer and output layer. In addition to this, we will have two additional perceptrons in the hidden layer which will lie between the input and output layer.
- We as such never see the workings of the hidden layer and that is why they are called so as we only see what we input and what we get as the output while the rest of the data processing is done in these hidden layers.

Deep Neural Networks (multilayer perceptrons)

- If we try to solve the XOR problem, this time using more than one layer of perceptrons we can be successful in correctly classifying the problem.

Practical Exercise

- If we have denoted True for 1 and False for 0 and we have the following dataset then we can perform a multilayer perceptrons.

x_1	x_2
0	0
1	1
0	1
1	0

Practical Exercise

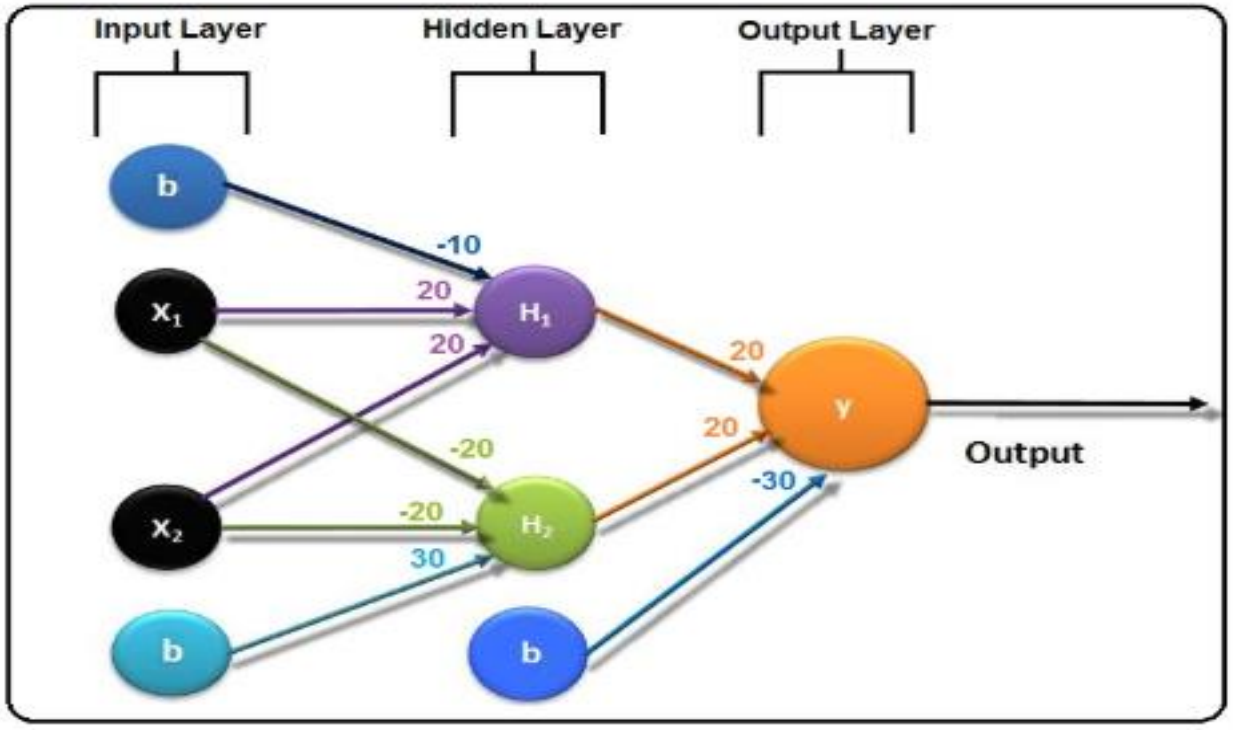
- We use a three-layer model wherein the hidden layer we will have two perceptrons h1 and h2 and another perceptron y in the output layer. We provide the following weights and bias to these perceptrons.

Practical Exercise

Perceptron	Weight	Bias
h_1	20	-10
h_2	-20	30
y	20	-30

Practical Exercise

- We can now visualise a three-level neural network.



Structure of 3-Level Artificial Neural Network

Practical Exercise

- For each perceptron, we will have the following calculations.

h_1	h_2	y
$(20_{x1} + 20_{x2} - 10)$	$(-20_{x1} - 20_{x2} + 30)$	$(20_{h1} - 20_{h2} - 30)$

Practical Exercise

- For the four data entries, we will multiply the data with the weights and add bias.
- As an activation function, we take a very simple step function where a negative value outputs 0 while a positive value outputs 1.
- Using the weights, bias and activation function we get the following results.

Practical Exercise

x_1	x_2	h_1	h_2	y	Resut
0	0	$((20 \times 0) + (20 \times 0)) - 10) = 0$	$((-20 \times 0) + (-20 \times 0)) + 30) = 1$	$((20 \times 0) + (20 \times 1)) - 30) = 0$	FALSE
1	1	$((20 \times 1) + (20 \times 1)) - 10) = 1$	$((-20 \times 1) + (-20 \times 1)) + 30) = 0$	$((20 \times 1) + (20 \times 0)) - 30) = 0$	FALSE
0	1	$((20 \times 0) + (20 \times 1)) - 10) = 1$	$((-20 \times 0) + (-20 \times 1)) + 30) = 1$	$((20 \times 1) + (20 \times 1)) - 30) = 1$	TRUE
1	0	$((20 \times 1) + (20 \times 0)) - 10) = 1$	$((-20 \times 1) + (-20 \times 0)) + 30) = 1$	$((20 \times 1) + (20 \times 0)) - 30) = 1$	TRUE

Practical Exercise

- We can see that for 0,1 and 1,0 we are able to get True as result.

- Thus, the idea behind Multilayer perceptron is linking multiple perceptrons together in order to solve non-linearly separable and very complex problems.

Practical Exercise

- Also, it is important to understand that so far we have discussed the feed forward process where all the connections are in one direction without any cycle.
- In Feedforward, the data from the input unit is passed to the next layer of the unit and after due computation, the result is passed to the next layer and so forth until the results are computed by the output layer.

Advantages of Multilayer Perceptrons

- The following two features characterise multilayer perceptrons and artificial neural networks in general.
- They are mainly responsible for the "edge" these networks have over conventional computing systems.

Advantages of Multilayer Perceptrons

- Generalisation

Neural networks are capable of generalisation, that is, they classify an unknown pattern with other known patterns that share the same distinguishing features. This means noisy or incomplete inputs will be classified because of their similarity with pure and complete inputs.

Advantages of Multilayer Perceptrons

- Fault Tolerance

Neural networks are highly fault tolerant. This characteristic is also known as "graceful degradation". Because of its distributed nature, a neural network keeps on working even when a significant fraction of its neurons and interconnections fail. Also, relearning after damage can be relatively quick.

Applications of Multilayer Perceptrons

- The multilayer perceptron with backpropagation has been applied in numerous applications ranging from OCR (Optical Character Recognition) to medicine. Brief accounts of a few are given below.
- Pattern Recognition
- Financial applications
- Speech synthesis

Limitations of Multilayer Perceptrons

- Computationally expensive learning process

Large number of iterations required for learning, not suitable for real-time learning

- No guaranteed solution

- Scaling problem

The End
?