神經網路如何運作

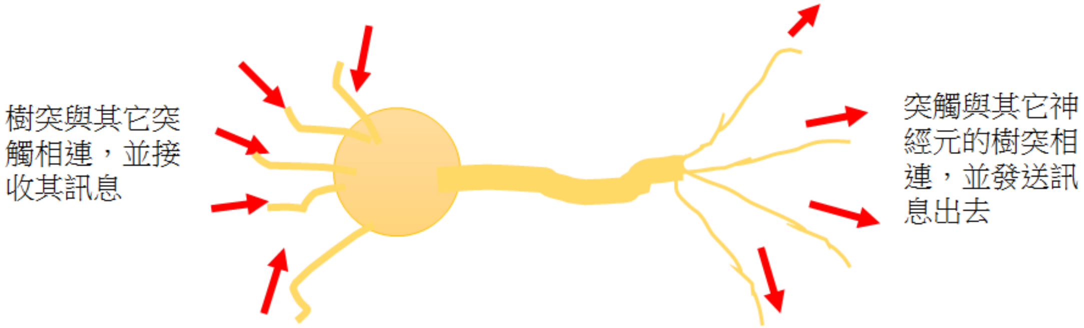
黃志勝 (Tommy Huang)
義隆電子 人工智慧研發部
國立陽明交通大學 AI學院 合聘助理教授
國立台北科技大學 電資學院合聘助理教授

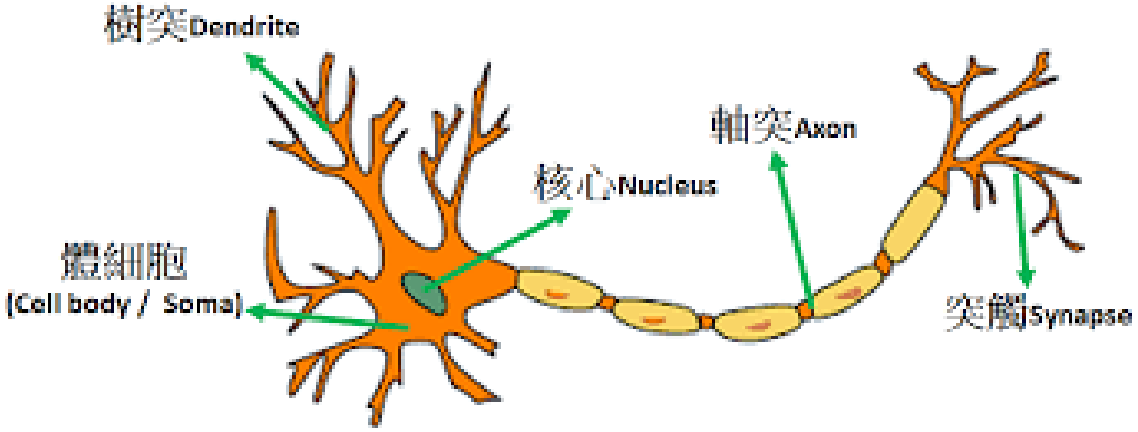
Outline

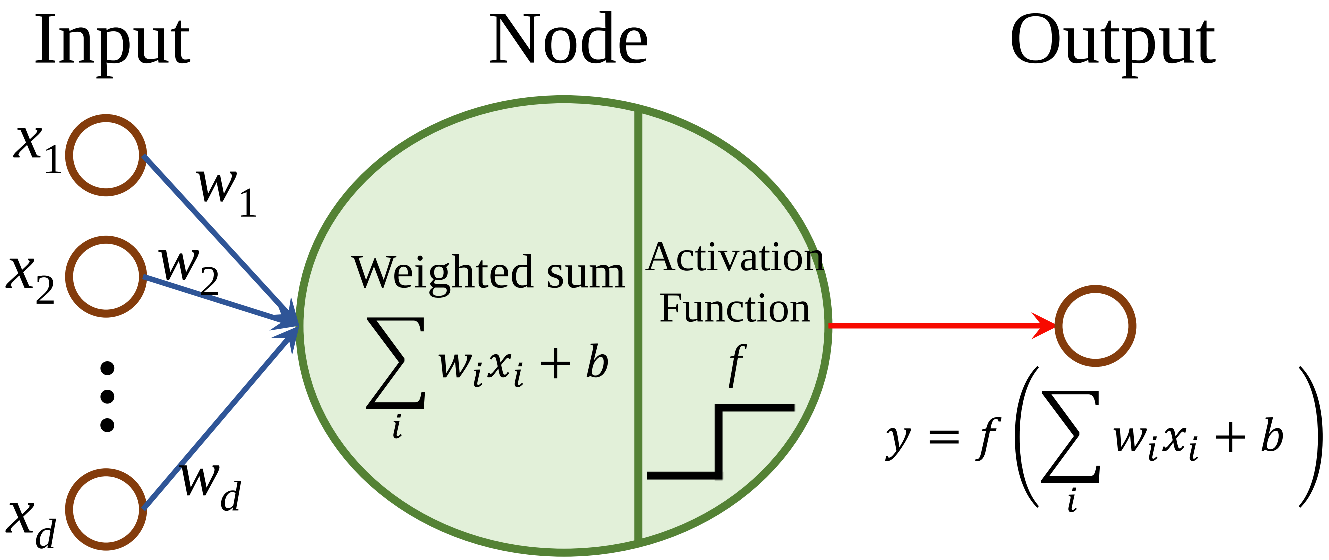
- 1.類神經網路(Neural Network, NN)
- 2.感知機(Perception)
- 3. Multi-layer perception (MLP)
- 4. How NN work?

類神經網路

基本上神經網路是基於感知機(Perceptron)神經網路開始，主要是希望用數學模型去模擬神經細胞的運作模式。


樹突與其它突觸相連，並接收其訊息
突觸與其它神經元的樹突相連，並發送訊息出去





$$y = f\left(\sum_i w_i x_i + b\right)$$

權重(w_i): Dendrite

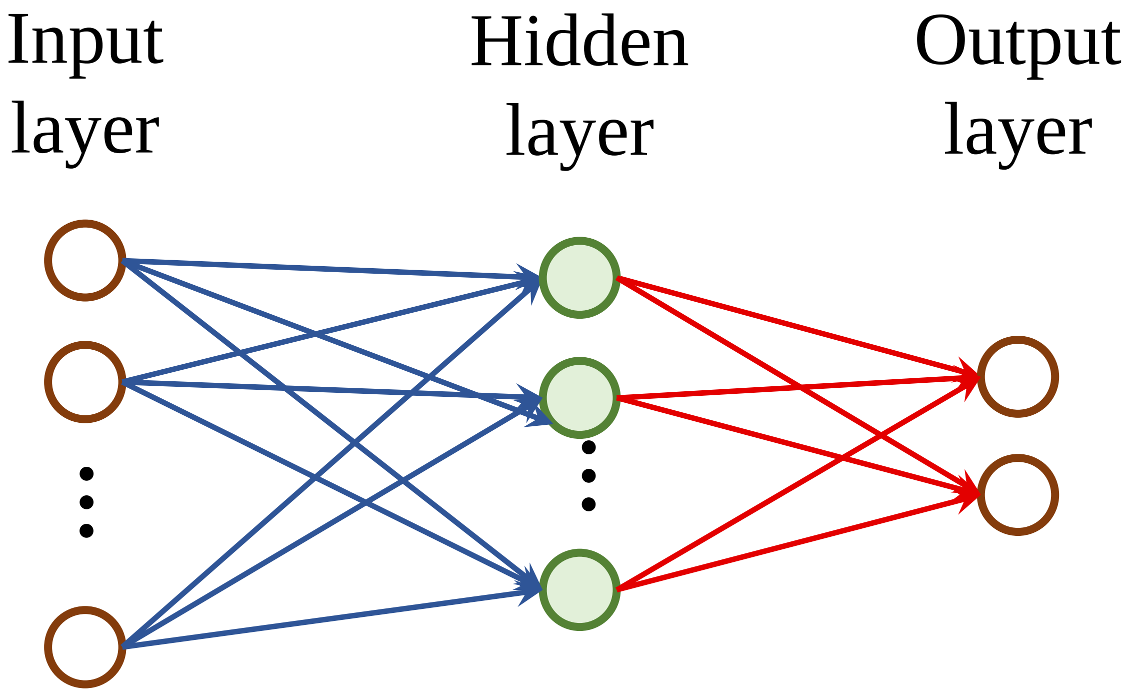
Input(x_i) and output (y) node: Synapse

Node: Cell body

Output: Axon

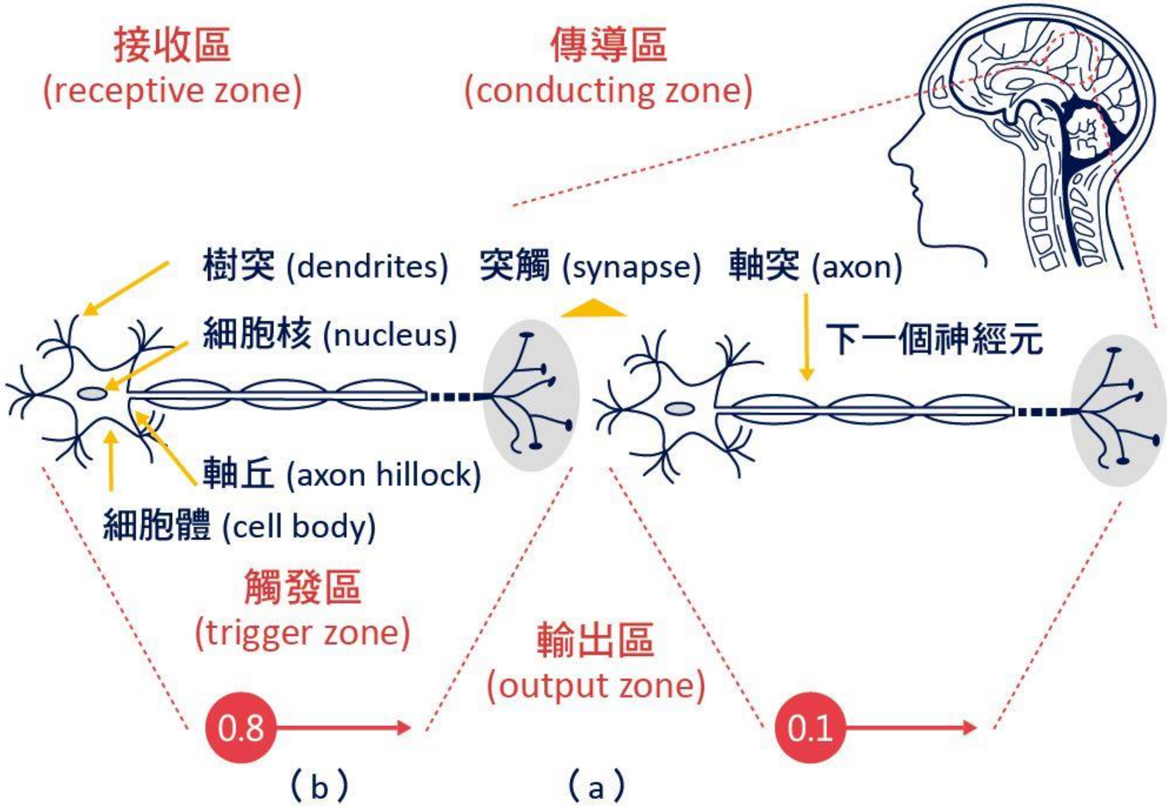
類神經網路

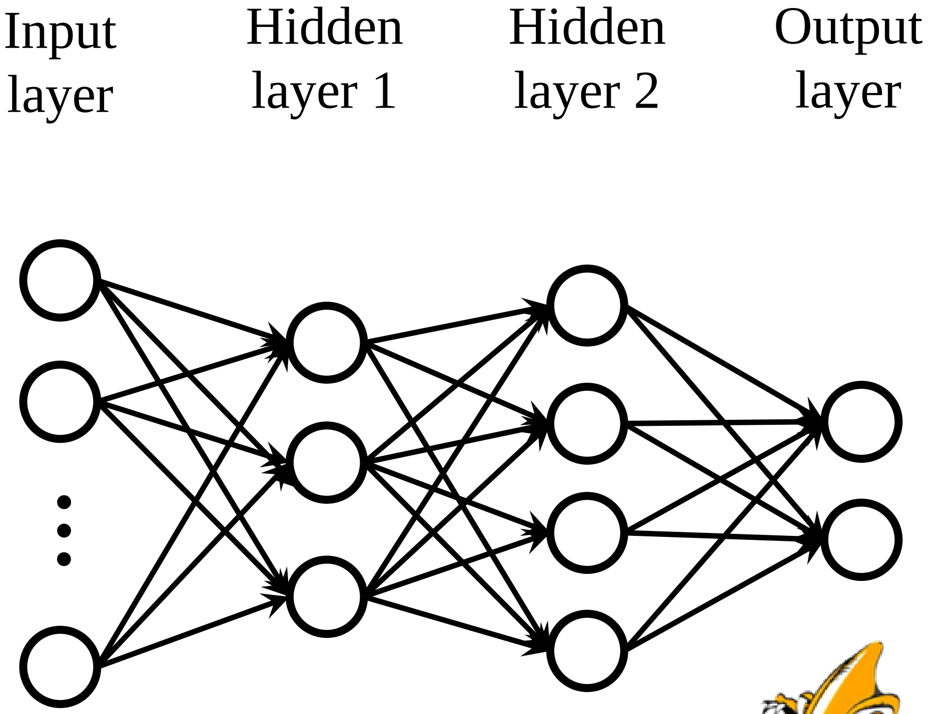
Single node Perceptron，神經運作不會只有一個的細胞訊息傳遞(Single node Perceptron)，因此出現Single layer perceptron



MLP

但神經訊息傳遞不會像只有一層Single layer perceptron運作，神經網路應該是多個細胞部段將訊息傳遞下去運作的模式，這就是Multilayer perception (MLP)，也就是一般認知的類神經網路。





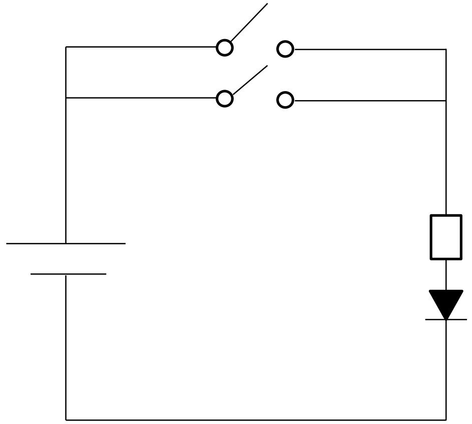
Outline

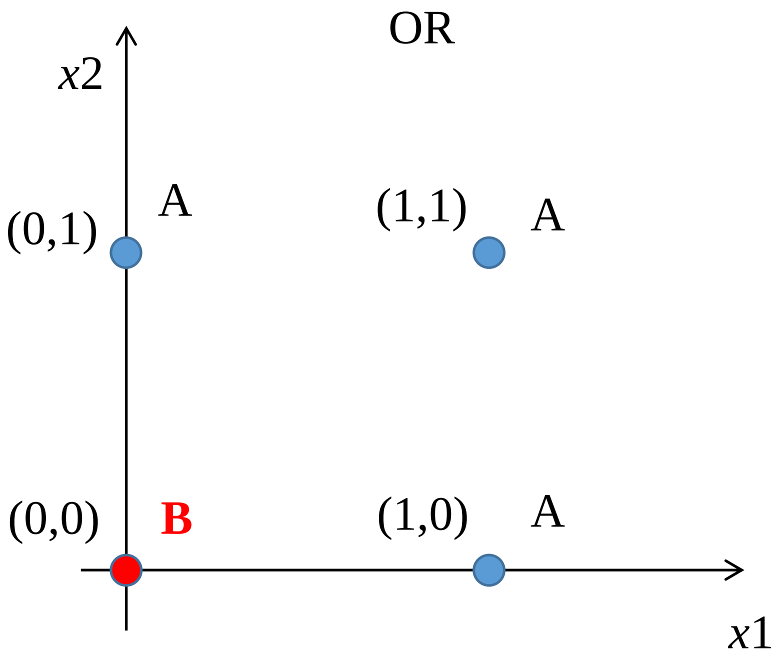
- 1.類神經網路(Neural Network, NN)
- 2.感知機(Perception)
- 3. Multi-layer perception (MLP)
- 4. How does NN work?

NN for OR problem

Truth Table for AND and OR problem

x1	x2	AND	Class	OR	Class
0	0	0	B	0	B
0	1	0	B	1	A
1	0	0	B	1	A
1	1	1	A	1	A

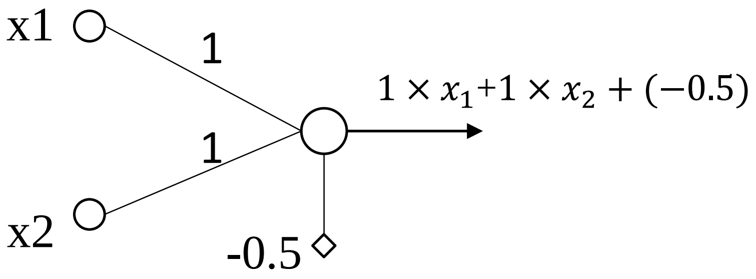


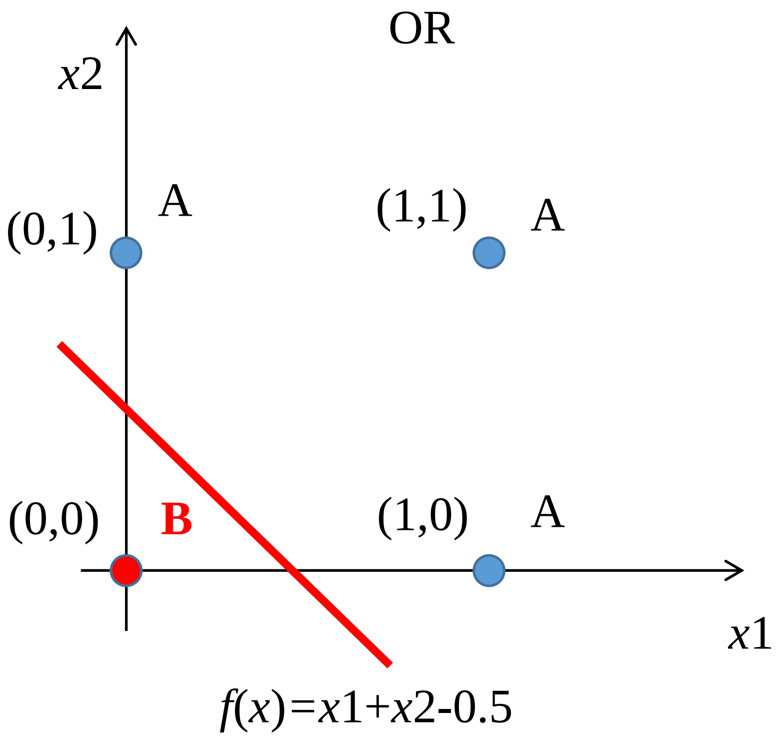


NN for OR problem

Truth Table for AND and OR problem

x1	x2	AND	Class	OR	Class
0	0	0	B	0	B
0	1	0	B	1	A
1	0	0	B	1	A
1	1	1	A	1	A

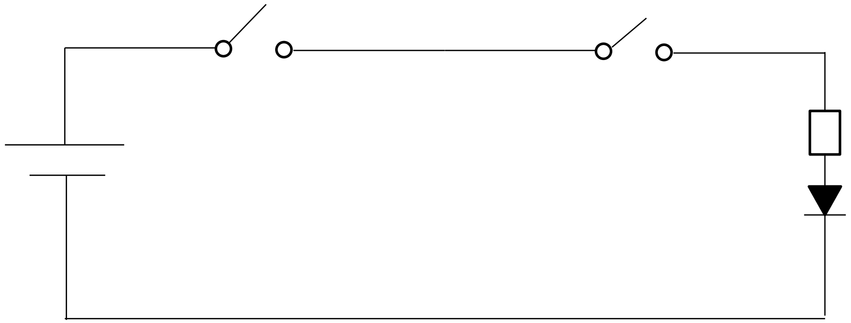


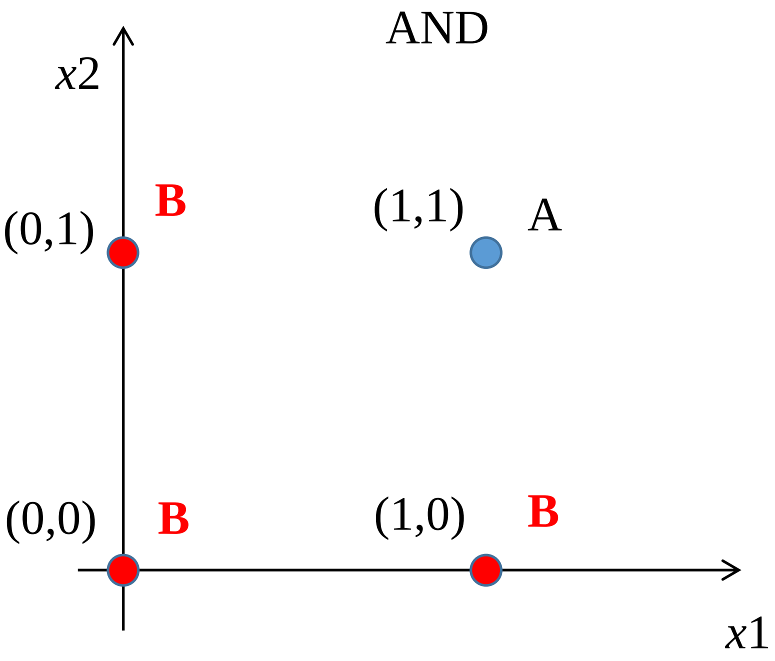


NN for AND problem

Truth Table for AND and OR problem

x1	x2	AND	Class	OR	Class
0	0	0	B	0	B
0	1	0	B	1	A
1	0	0	B	1	A
1	1	1	A	1	A

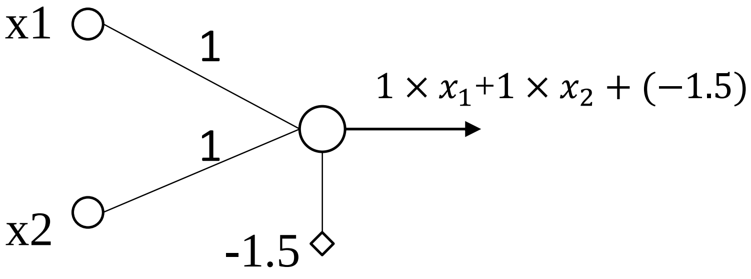


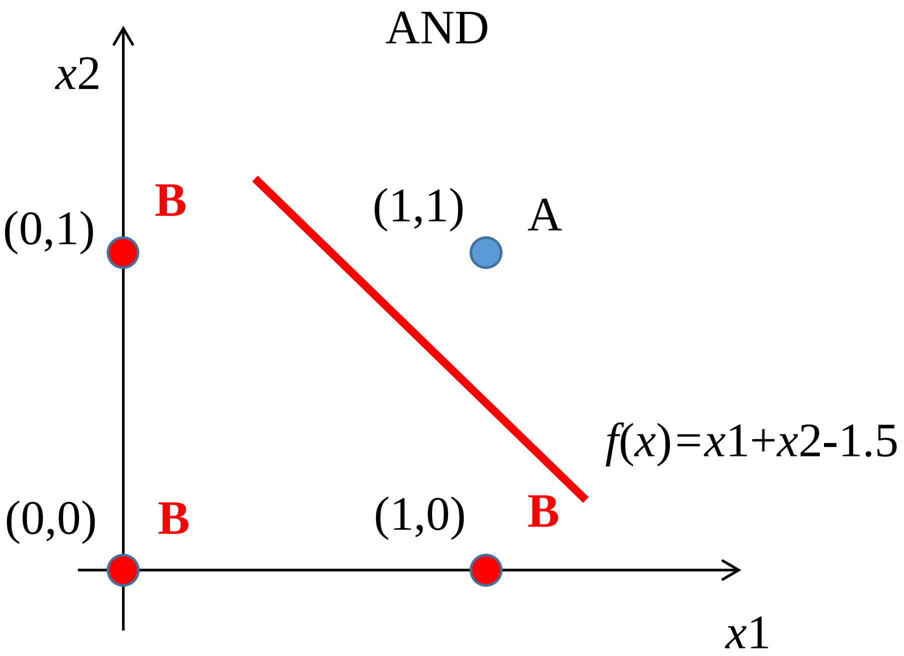


NN for AND problem

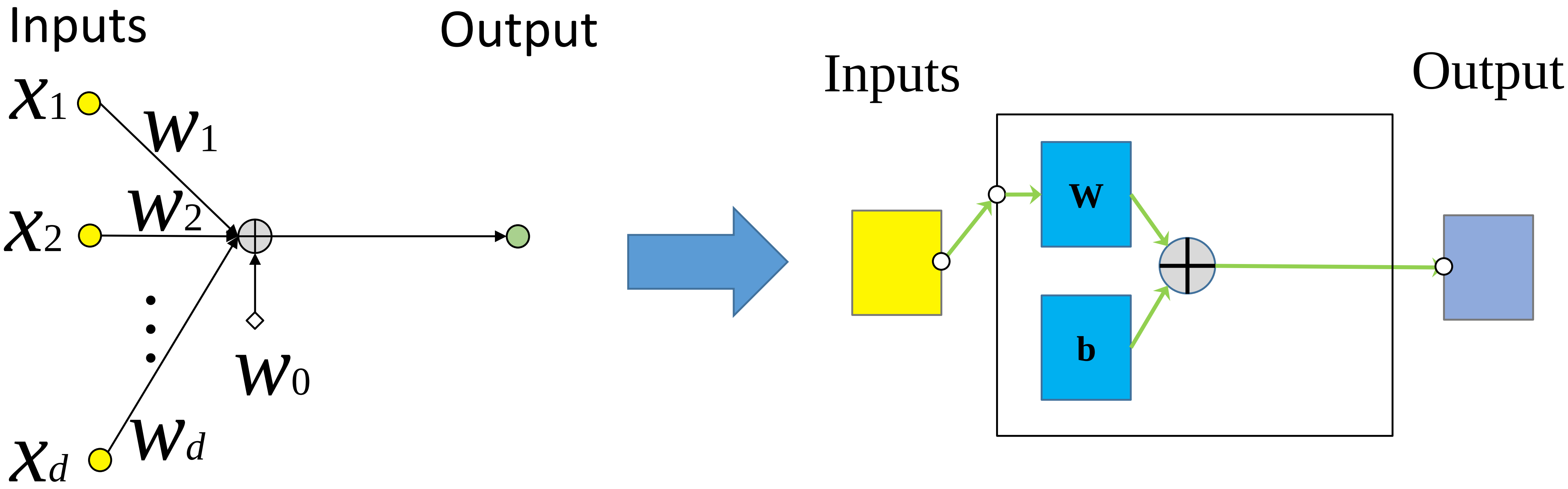
Truth Table for AND and OR problem

x1	x2	AND	Class	OR	Class
0	0	0	B	0	B
0	1	0	B	1	A
1	0	0	B	1	A
1	1	1	A	1	A



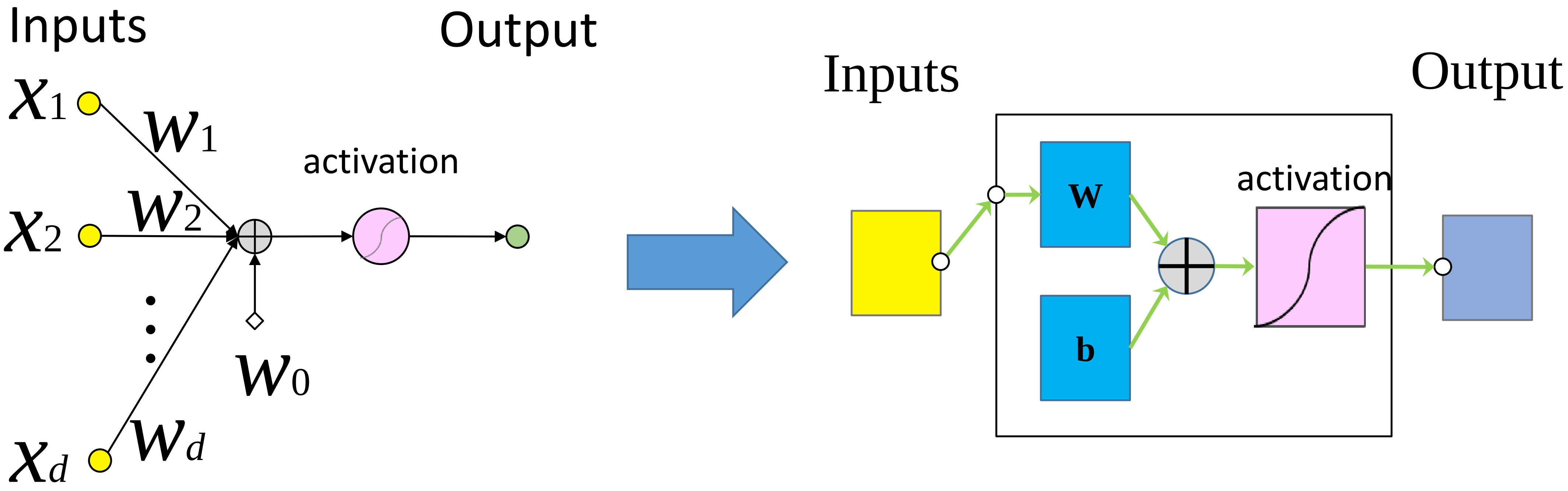


Perception

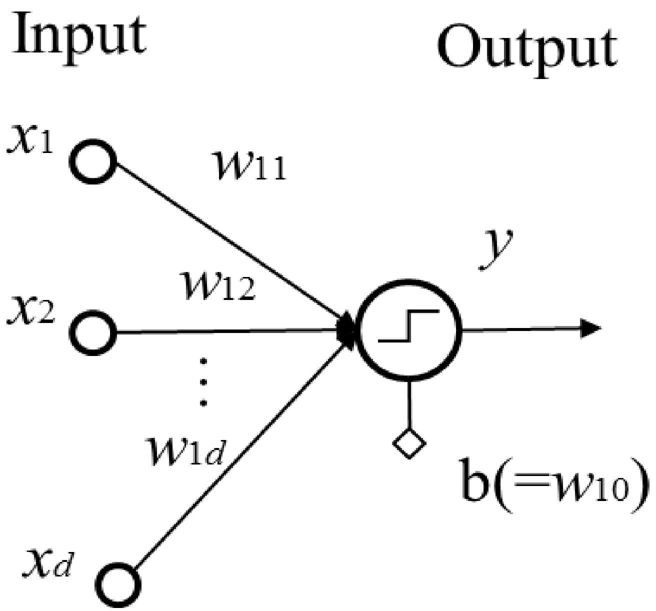


Perception

Activation function is a very import in NN, it lets the perception is learning by nonlinear representation.



Perception



$$y = f(w_{10} + w_{11}x_{1+}w_{12}x_2 + \cdots + w_{1d}x_d)$$
$$= f(\boldsymbol{W_1^T x} + w_{10})$$

Classification: $f = \begin{cases} 1 & \boldsymbol{W_1^T x} + w_{10} \geq 0 \\ 0 & O.W. \end{cases}$

Regression:$f(\boldsymbol{W_1^T x} + w_{10})$

Perception with linear output is linear regression.

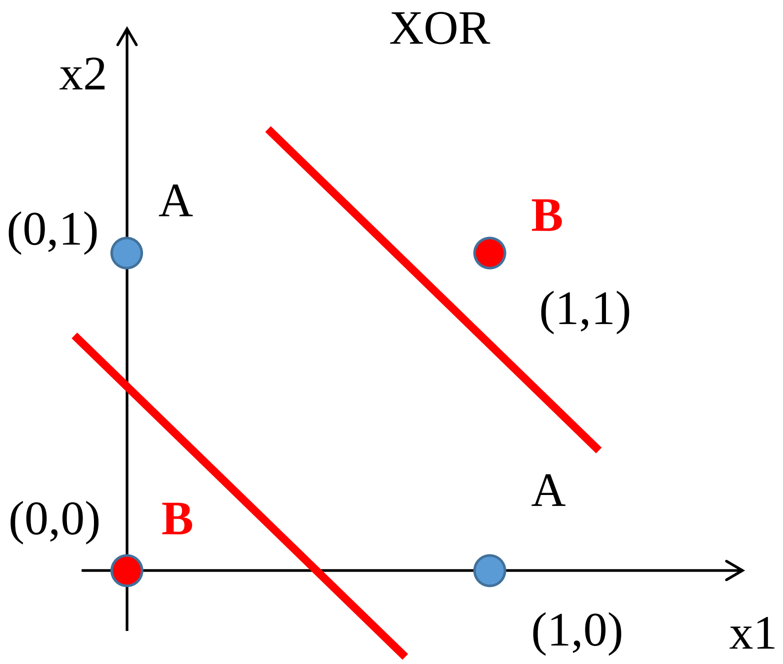
The difference is the parameter search for NN is backpropagation, regression is OLSE.

NN for XOR problem

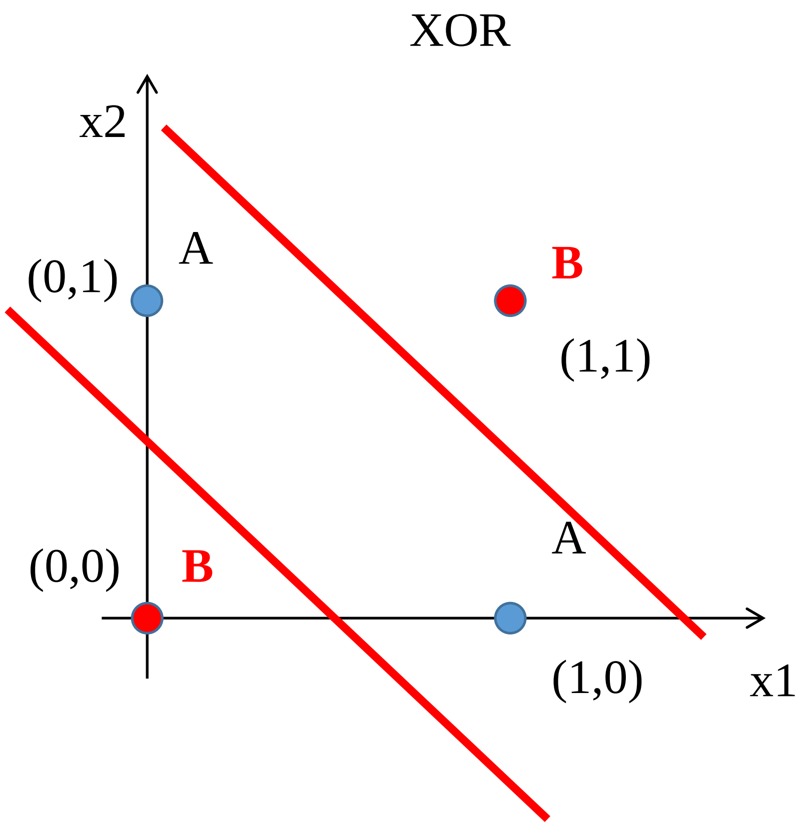
- Exclusive OR (XOR Boolean function)
- It’s impossible to find a single straight line to separate two classes.

Truth Table for the XOR problem

x1	x2	AND	Class
0	0	0	B
0	1	1	A
1	0	1	A
1	1	0	B



NN for XOR problem



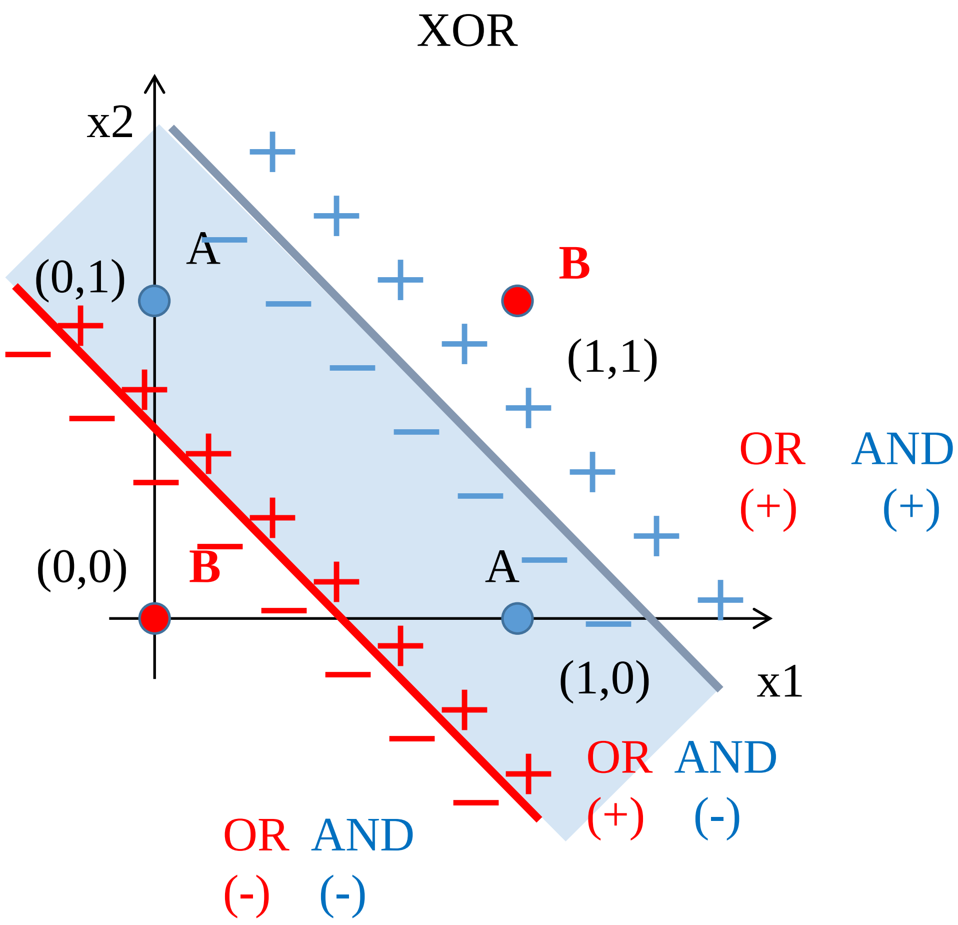
OR

$$h_1(x) = x_1 + x_2 - 0.5 = 0$$

AND

$$h_2(x) = x_1 + x_2 - 1.5 = 0$$

NN for XOR problem



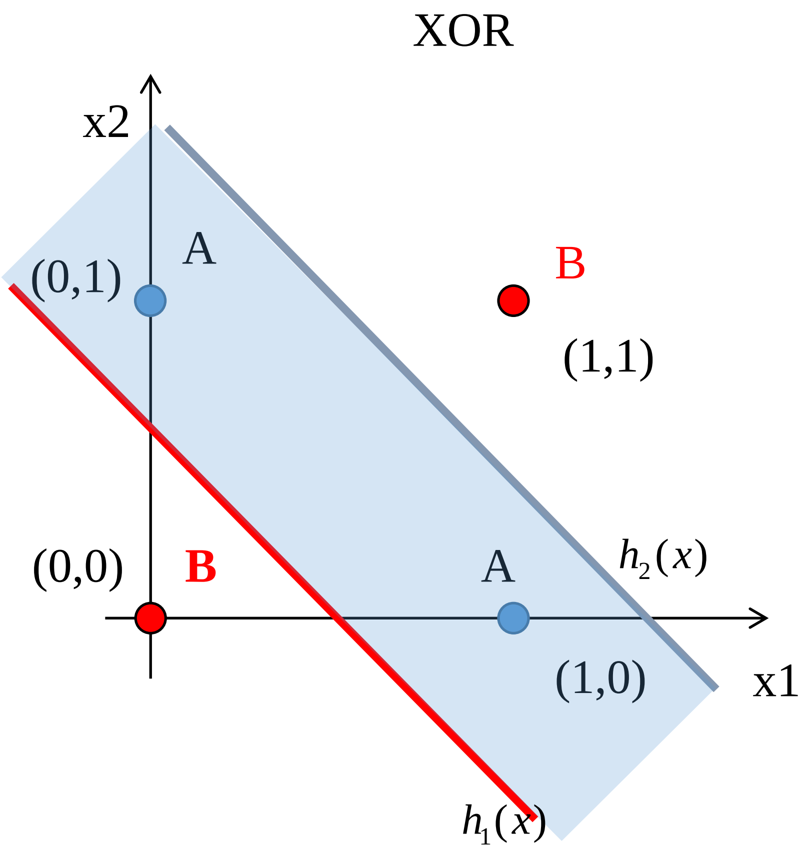
OR

$$h_1(x) = x_1 + x_2 - 0.5 = 0$$

AND

$$h_2(x) = x_1 + x_2 - 1.5 = 0$$

NN for XOR problem



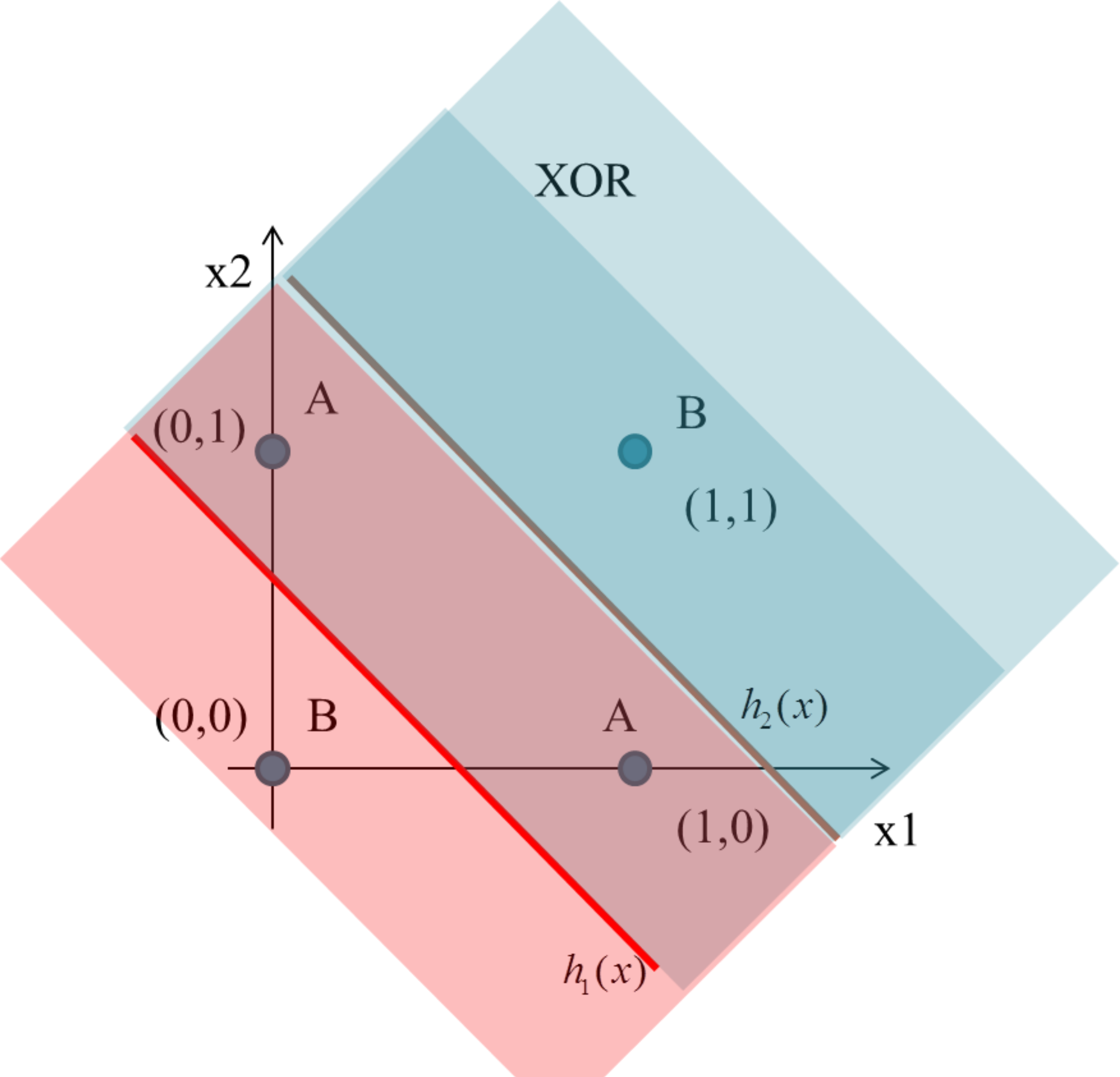
OR

$$h_1(x) = x_1 + x_2 - 0.5 > 0$$

AND

$$h_2(x) = x_1 + x_2 - 1.5 < 0$$

NN for XOR problem



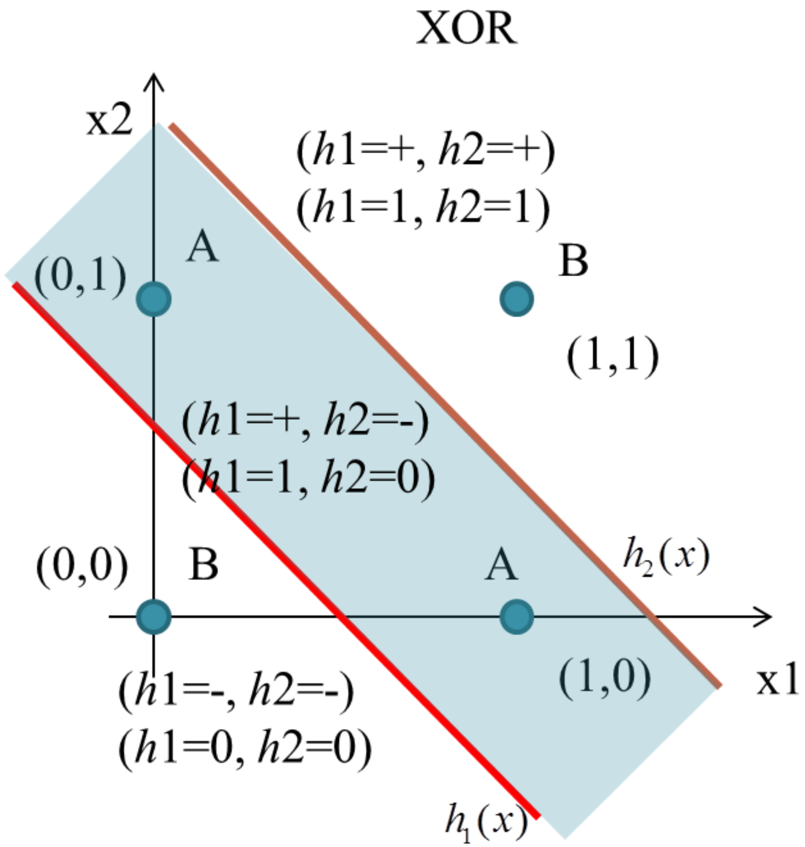
OR

$$h_1(x) = x_1 + x_2 - 0.5 \lessgtr 0$$

AND

$$h_2(x) = x_1 + x_2 - 1.5 \lessgtr 0$$

NN for XOR problem



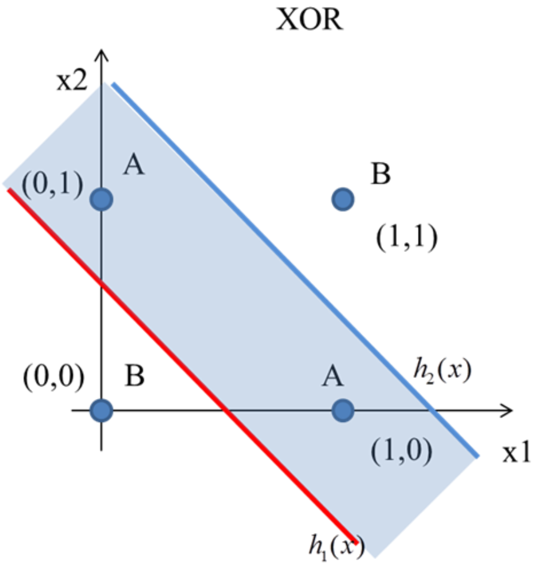
OR

$$h_1(x) = x_1 + x_2 - 0.5 < 0$$

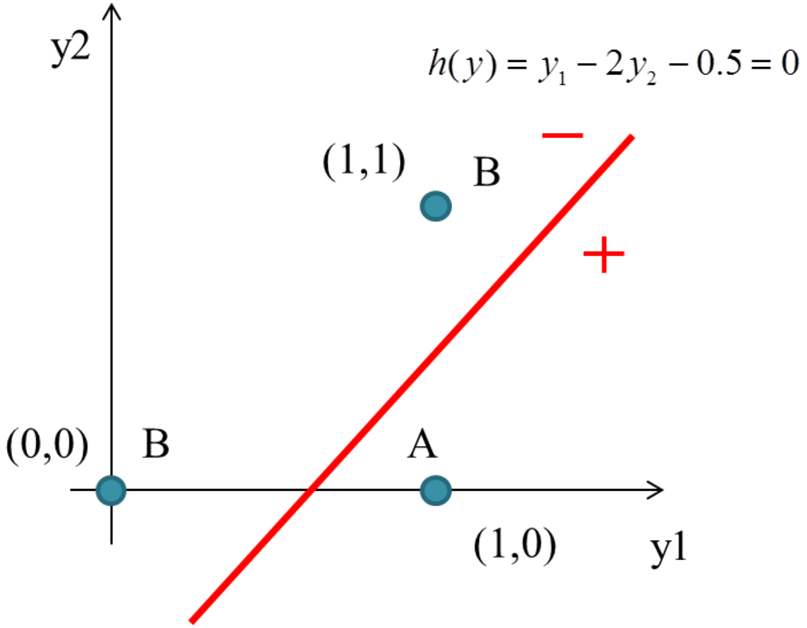
AND

$$h_2(x) = x_1 + x_2 - 1.5 > 0$$

NN for XOR problem

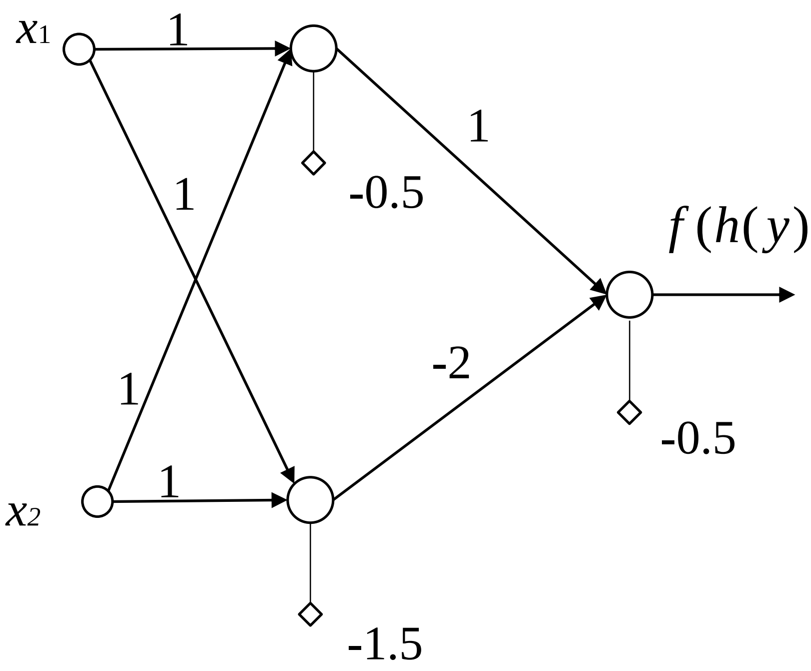


Truth Table for XOR problem				
x1	x2	y1	y2	Class
0	0	0(-)	0(-)	B
0	1	1(+)	0(-)	A
1	0	1(+)	0(-)	A
1	1	1(+)	1(+)	B



Two Layer Perception

$$y_1 = f(h_1(x)) = f(x_1 + x_2 - 0.5)$$



$$f(h(y)) = f(y_1 - 2y_2 - 0.5)$$

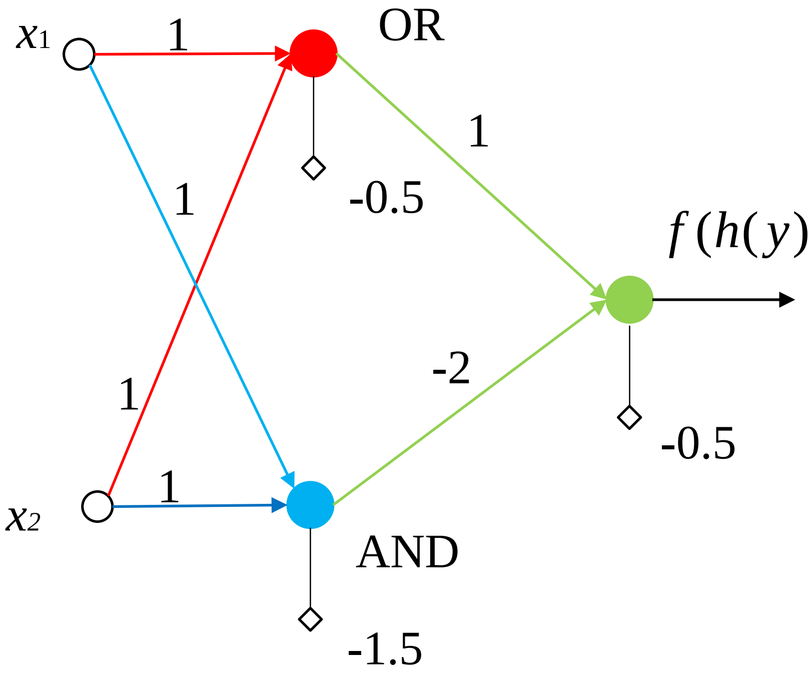
$$y_2 = f(h_2(x)) = f(x_1 + x_2 - 1.5)$$

$$f(v) = \begin{cases} 1 & if \;\; v \geq 0 \\ o & if \;\; v < 0 \end{cases}$$

Step activation function

Two Layer Perception

$$y_1 = f(h_1(x)) = f(x_1 + x_2 - 0.5)$$



$$f(h(y)) = f(y_1 - 2y_2 - 0.5)$$

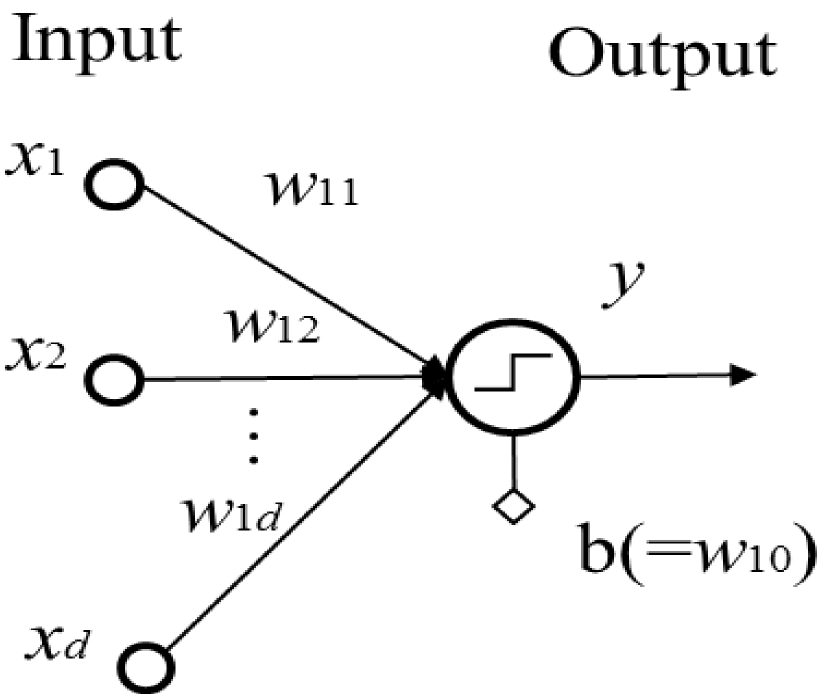
$$y_2 = f(h_2(x)) = f(x_1 + x_2 - 1.5)$$

$$f(v) = \begin{cases} 1 & if \;\; v \geq 0 \\ o & if \;\; v < 0 \end{cases}$$

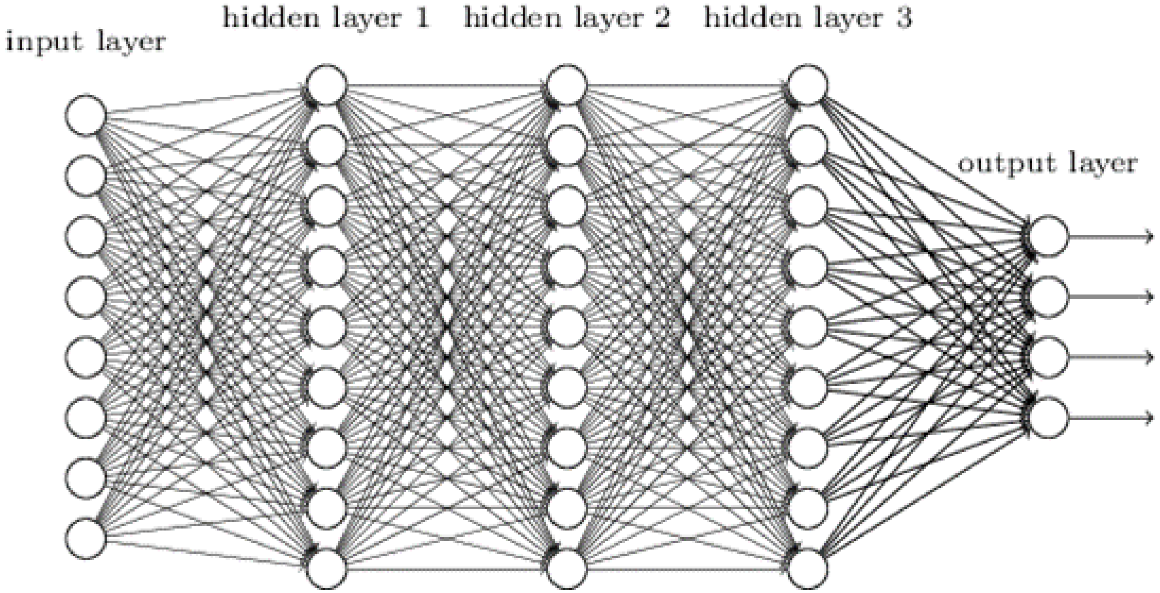
Step activation function

“Perception” and “multi-layer perception (MLP)”

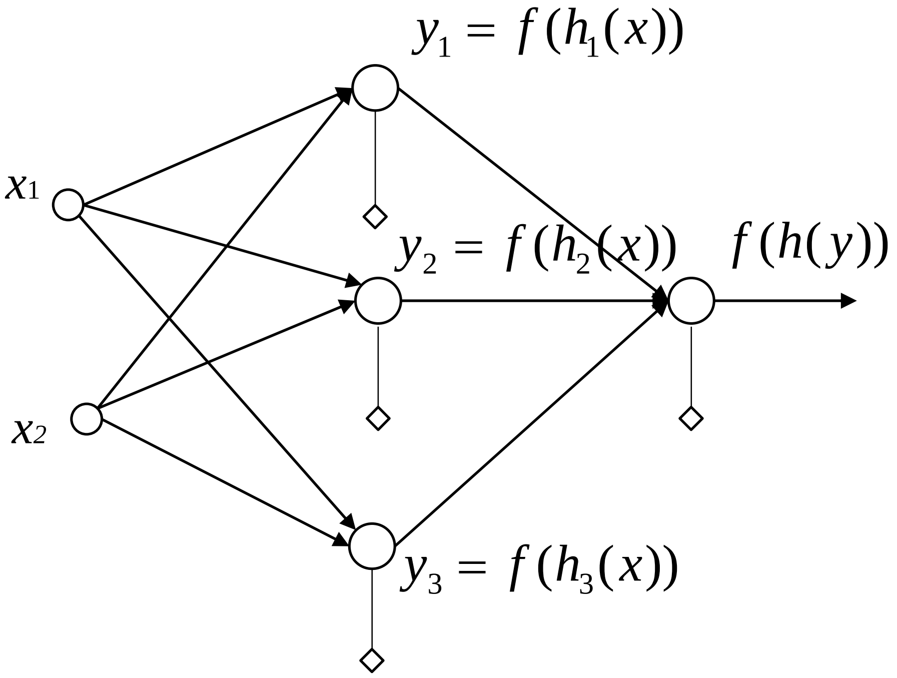
Perception



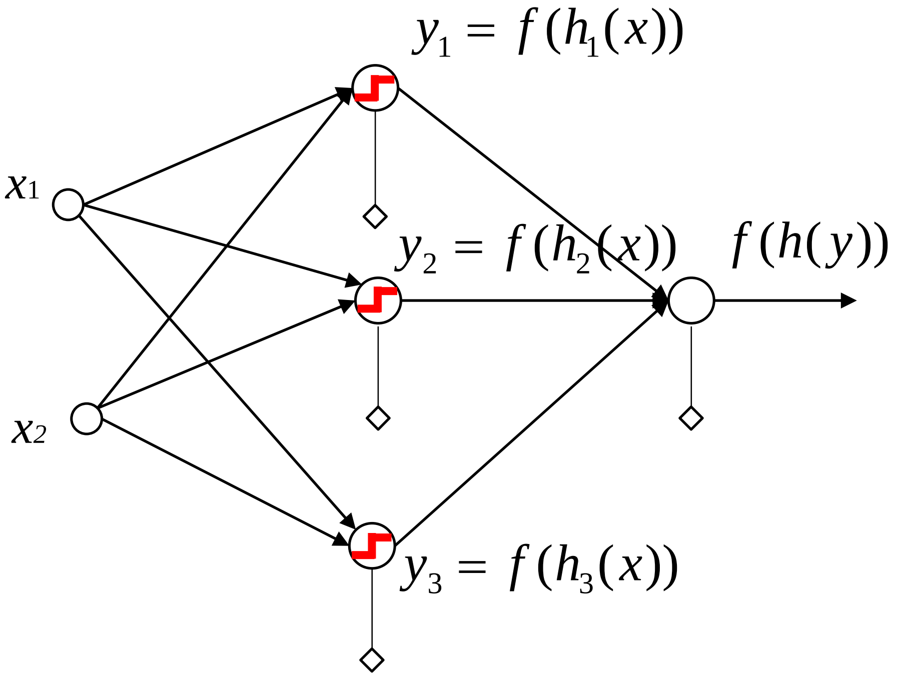
MLP



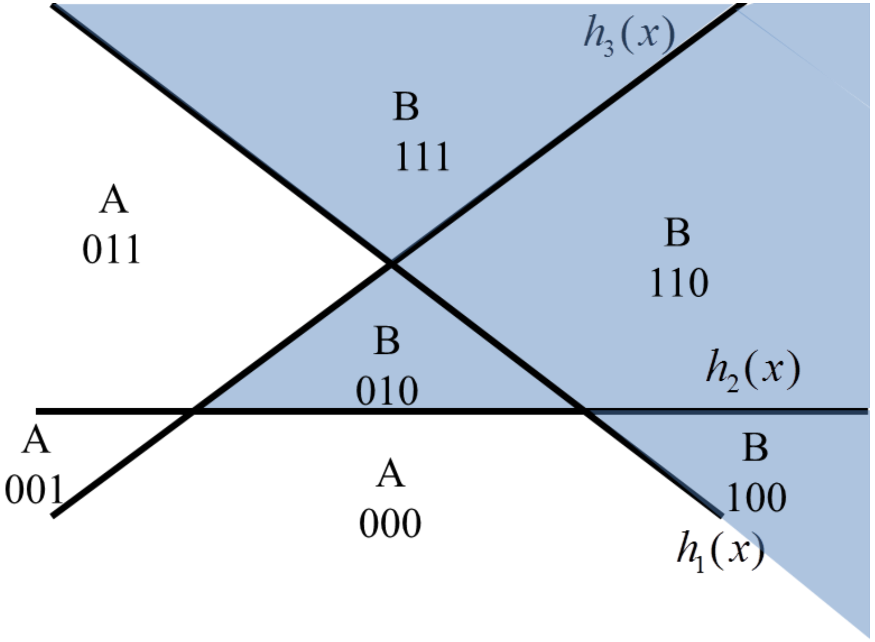
Polyhedral Regions



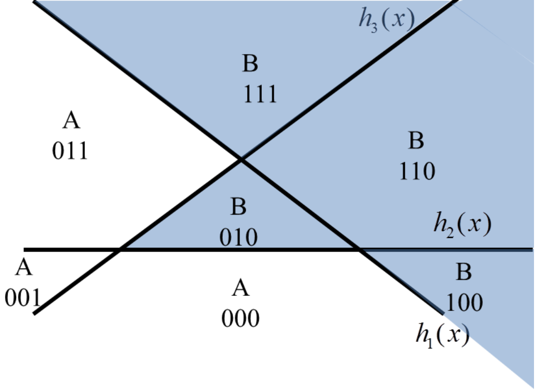
Polyhedral Regions



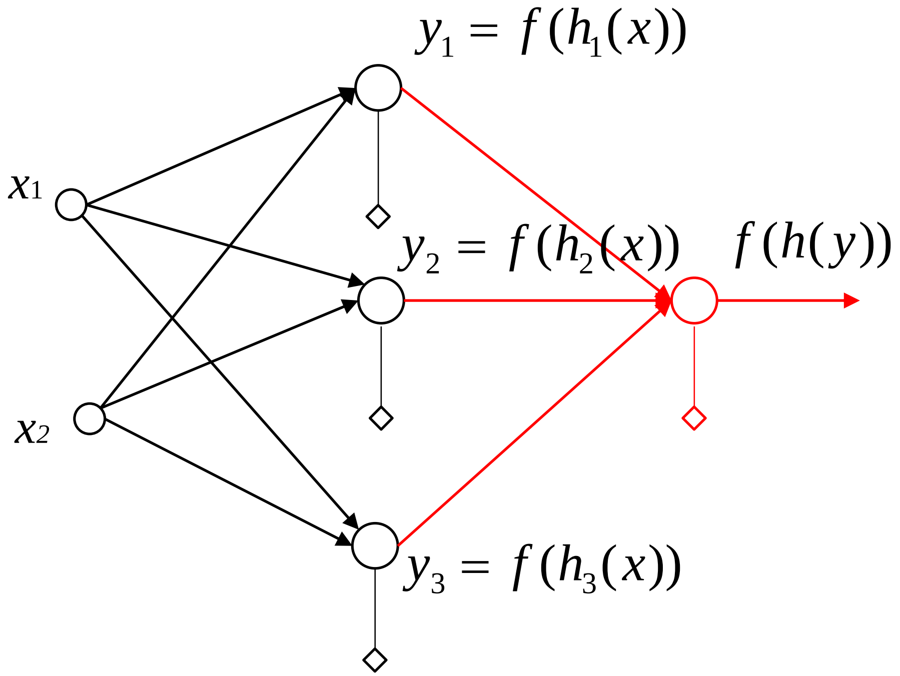
The first layer of neurons divides the input d-dimensional space into polyhedral, which are formed by hyperplane intersections.

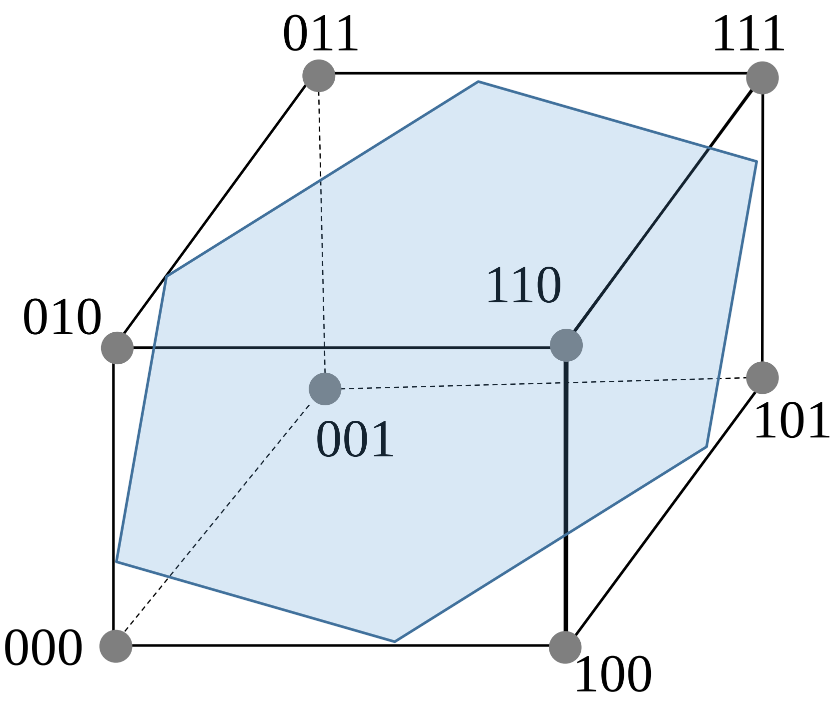


Polyhedral Regions



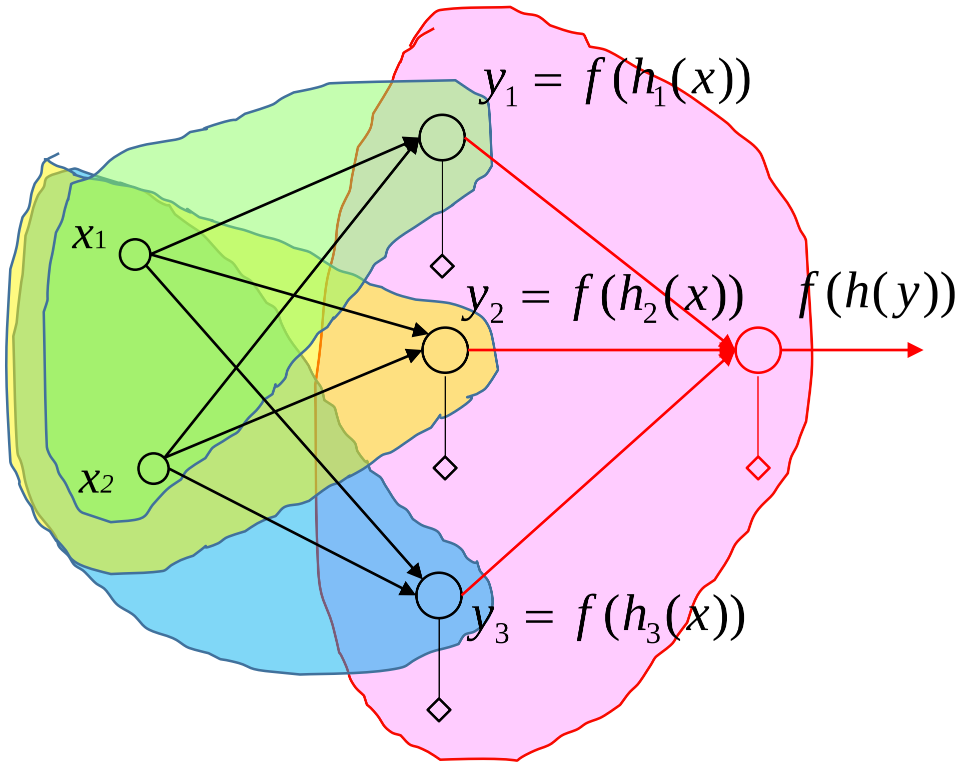
All vectors located within one of these polyhedral regions are mapped onto a specific vertex of the unit hypercube.



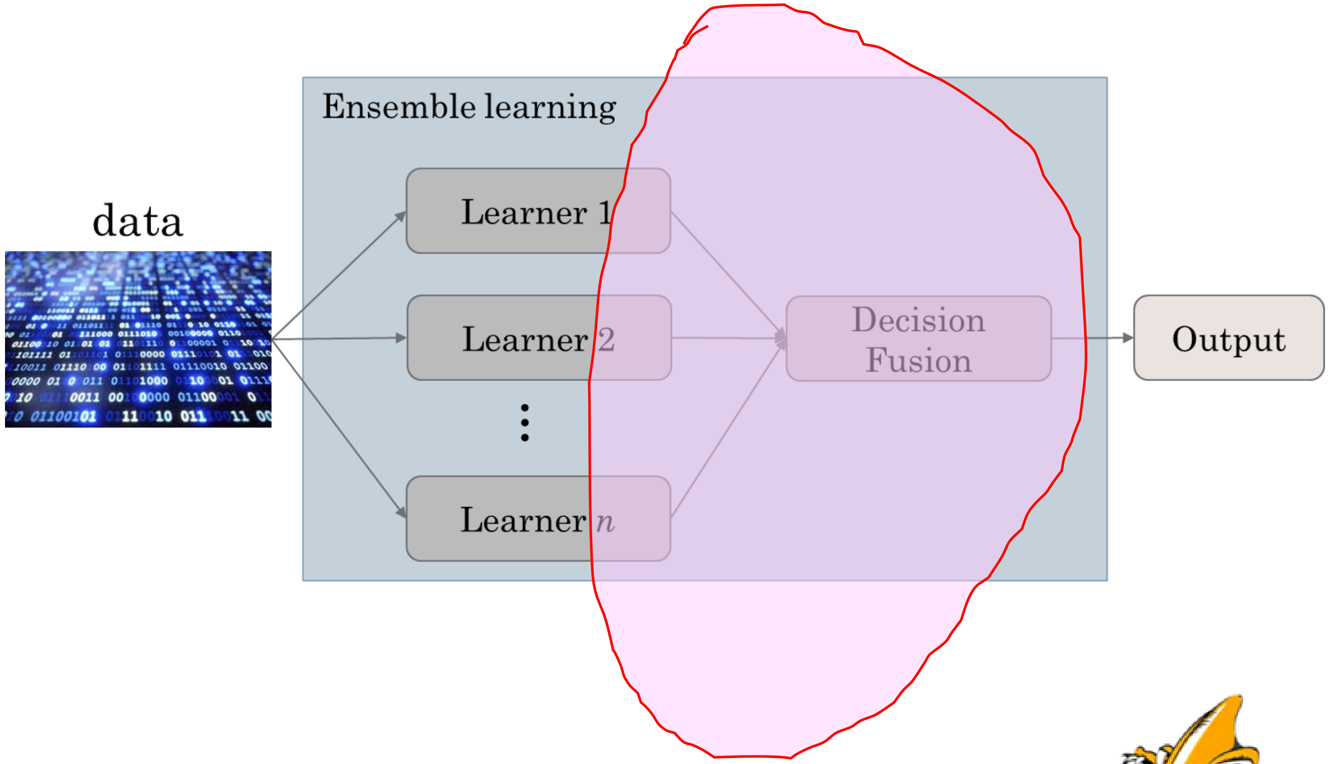


NN and Ensemble Learning

NN



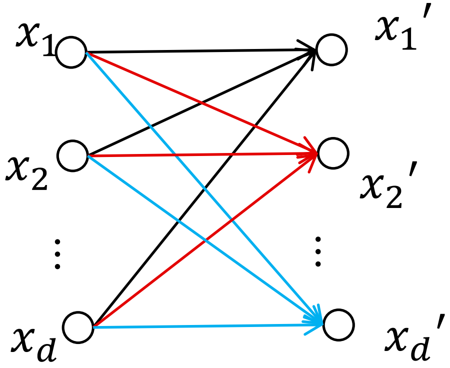
Ensemble Learning



NN and Dimension Reduction

- Principle Component Analysis

$$\boldsymbol{x}' = \begin{bmatrix} x_1{}' \\ \vdots \\ x_d{}' \end{bmatrix} = W^T\boldsymbol{x} = \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \vdots & \ddots & \vdots \\ w_{d1} & \cdots & w_{dd} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$


x_1
x_2
⋮
x_d
$x_1{}'$
$x_2{}'$
⋮
$x_d{}'$