MACHINE INTELLIGENCE

4^{th} lecture

Dr. Ahmad Al-Mahasneh

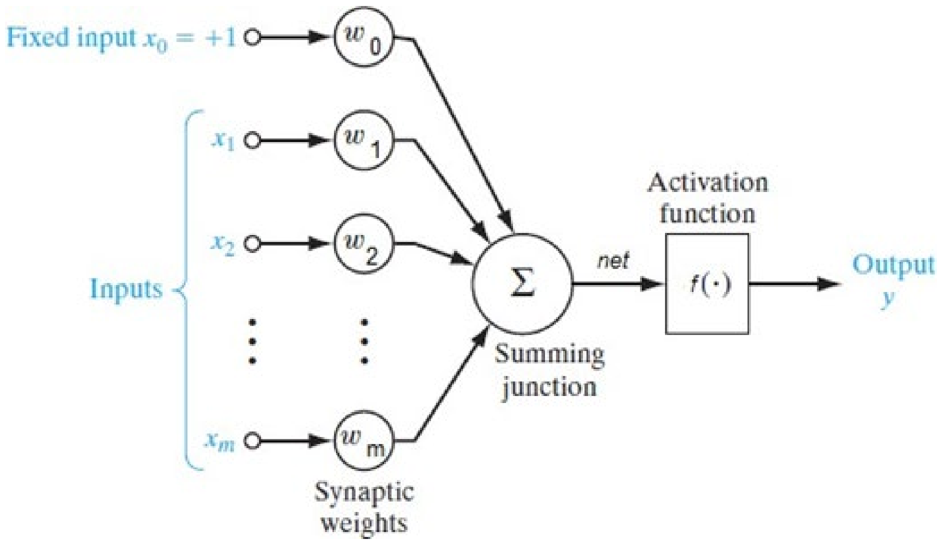
Review of the last lecture

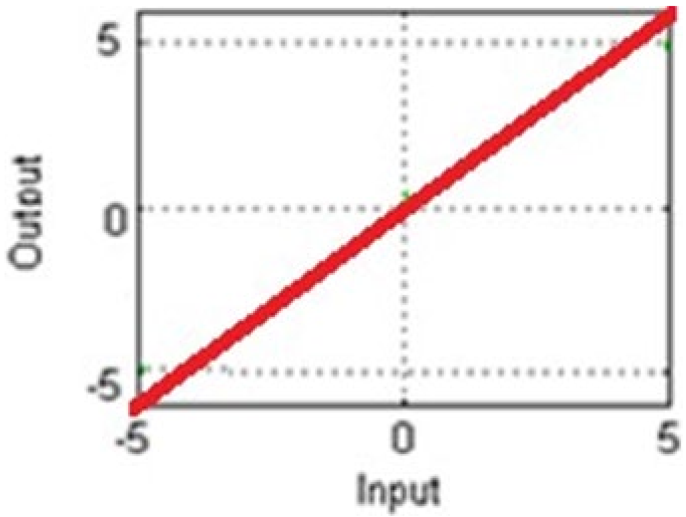
- ❖ Human brain and neural networks.
- ❖ Artificial neural networks properties, elements, activation functions and architecture.
- ❖ Perceptron, OR, AND and XOR problems.

Outline

- Activation functions.
- Feed Forward Neural Networks (FFNNs)
- Radial Basis Functions Neural Network (RBFNNs)

Linear Activation Function





net= $\sum_{i=0}^{m} x_i w_i$

y = k.net

$$\frac{\partial y}{\partial w_i} = K.x$$

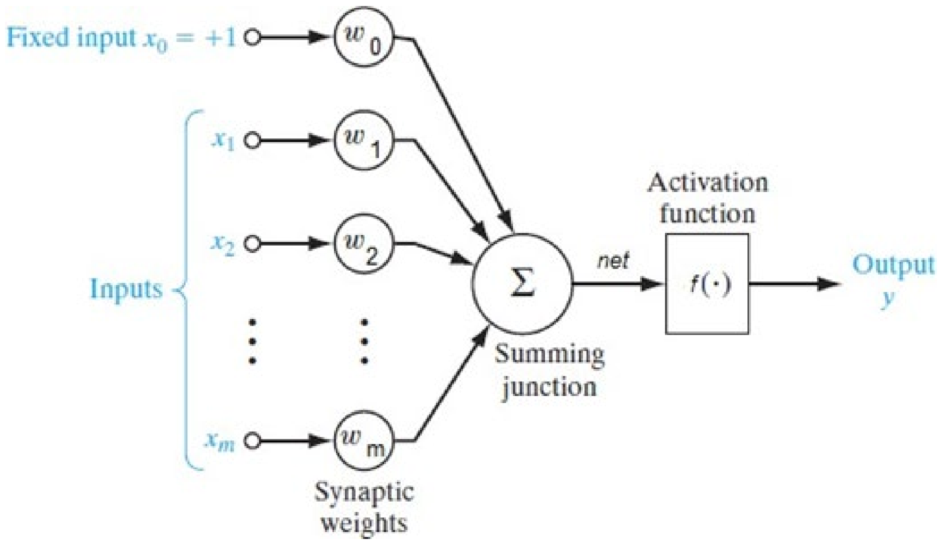
Pros

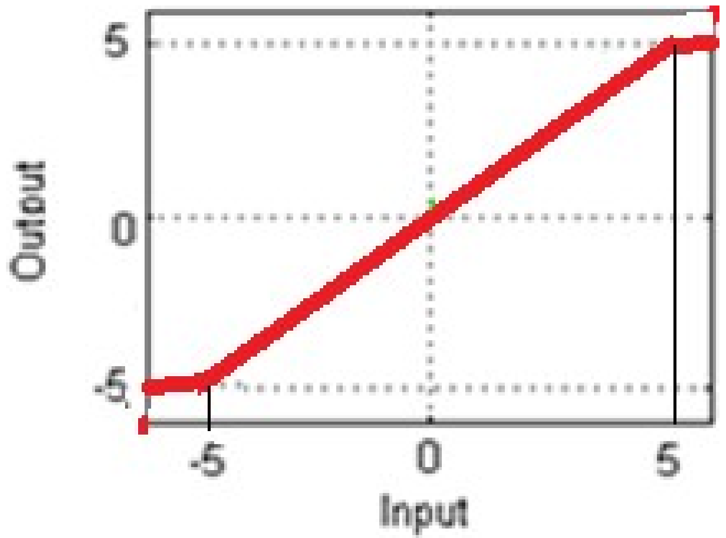
- Simple to use
- Differentiable

Cons:

- Not accurate for approximating nonlinear behavior
- Unbounded

Ramp Activation Function





$$y = f(net) = \begin{cases} max & net > UB \\ K.net & UB < net < LB \\ min & net < LB \end{cases}$$

$$\frac{\partial y}{\partial w_i} = K.x$$

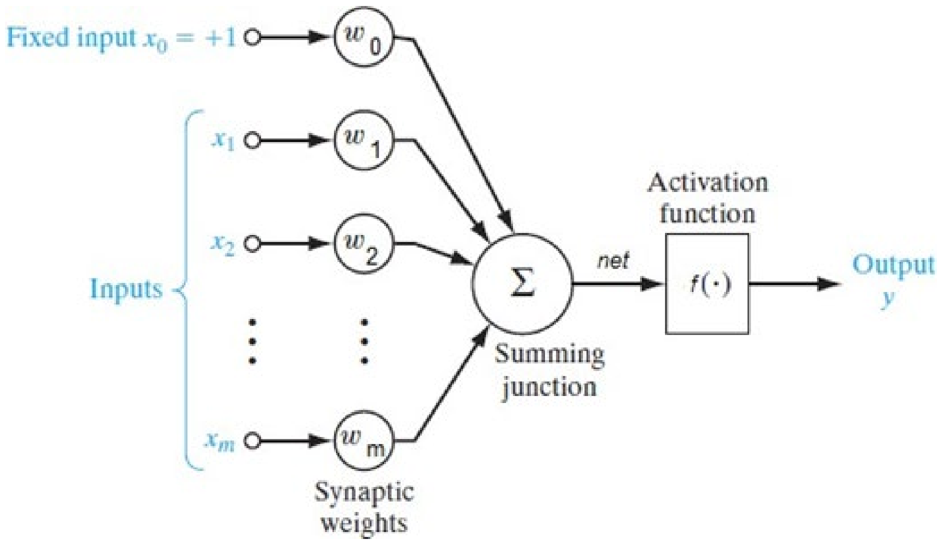
Pros

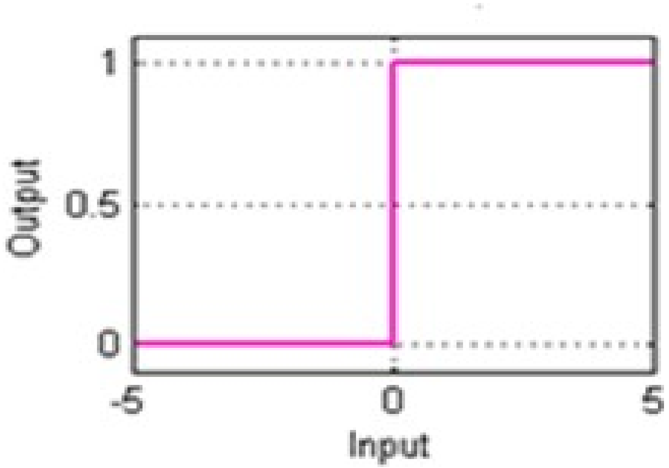
- Simple to use
- Bounded

Cons:

- Not accurate for approximating nonlinear behavior
- Derivative is not found at UB and LB

Threshold Activation Function





$$y = f(net) = \begin{cases} +1 & net >= 0 \\ 0 & net < 0 \end{cases}$$

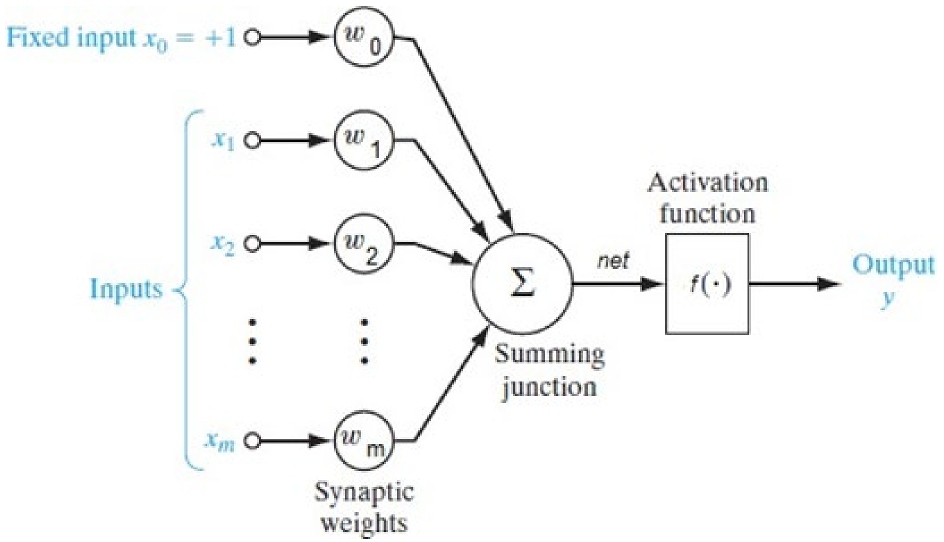
Pros:

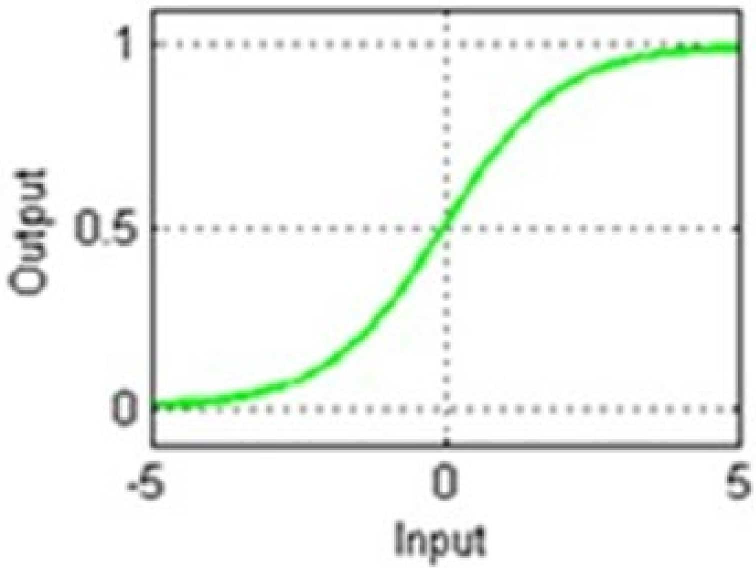
- Simple to use
- Bounded

Cons:

- Not accurate for approximating nonlinear behavior
- Cannot find the derivative at net = 0

Sigmoid Activation Function





$$y = f(net) = \frac{1}{1 + e^{-net}}$$

Pros:

- Simple to use
- Differentiable
- Bounded

Sigmoid Activation Function

Sigmoid derivative

$$y' = f'(net) = \frac{e^{-net}}{(1 + e^{-net})^2}$$

$$y' = f'(net) = \frac{1 + e^{-net} - 1}{(1 + e^{-net})^2} \Rightarrow y' = \frac{1 + e^{-net}}{(1 + e^{-net})^2} - \frac{1}{(1 + e^{-net})^2}$$

$$y' = y - y^2 \Rightarrow y' = y(1 - y)$$

This is a Derivative in a closed-form

Sigmoid Activation Function

Sigmoid partial derivative

$$\frac{\partial y}{\partial net} = y(1 - y) \qquad net = \sum_{i=0}^{m} x_i w_i$$

$$\frac{\partial y}{\partial w_i} = \frac{\partial y}{\partial net}\frac{\partial net}{\partial w_i} \quad \Rightarrow \quad \frac{\partial y}{\partial w_i} = y(1 - y)x_{\mathrm{i}}$$

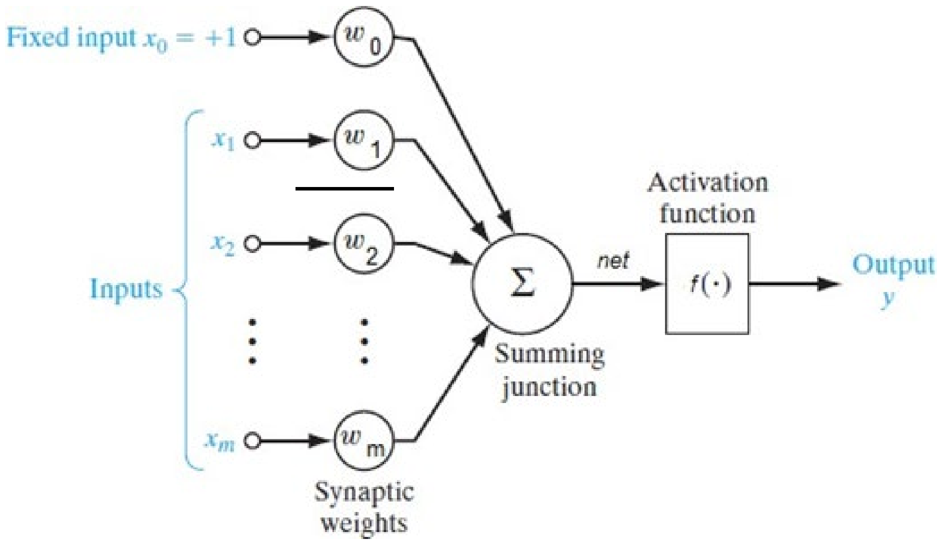
Chain rule:

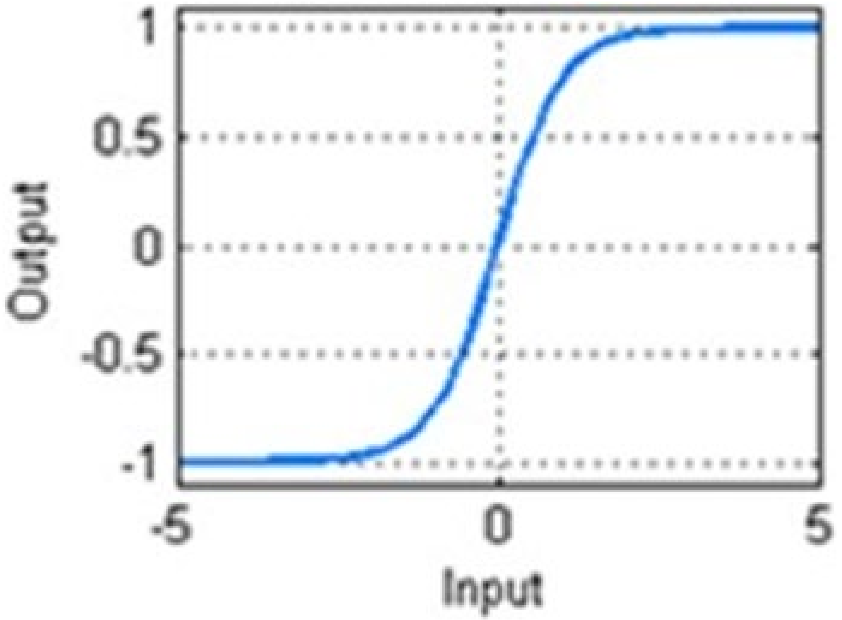
If

$$y = f(g(x))$$

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial f}\frac{\partial f}{\partial g}\frac{\partial g}{\partial x}$$

Hyperbolic Tangent Function





$$y(net) = \frac{(e^{net} - e^{-net})}{(e^{net} + e^{-net})}$$

$$\frac{\partial y}{\partial net} = 1 - y^2$$

Pros:

- Range +1 to -1
- Simple to use
- Differentiable
- Bounded

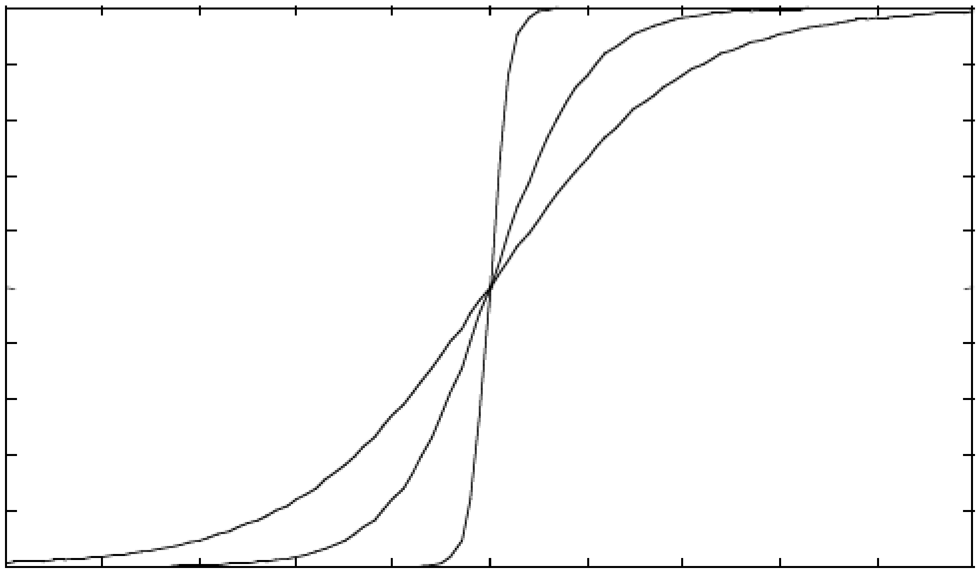
Hyperbolic Tangent Function

Partial derivative

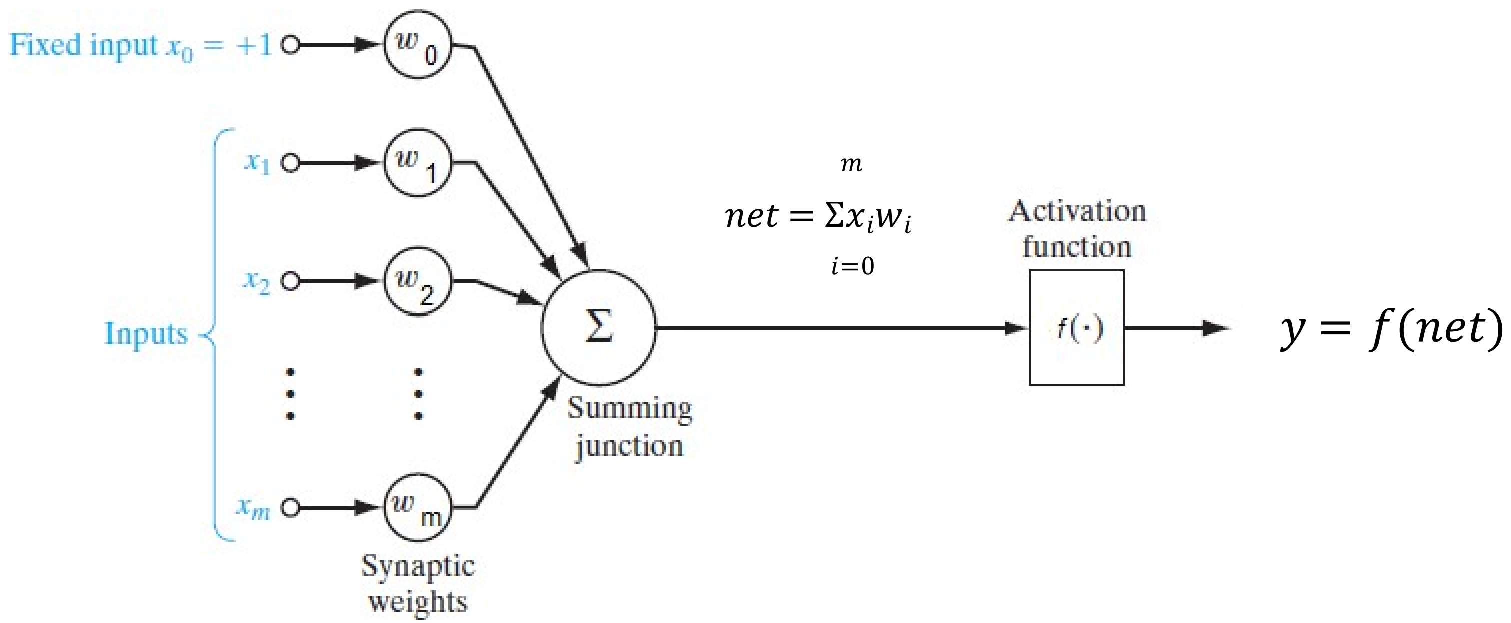
$$\frac{\partial y}{\partial w_i} = \frac{\partial y}{\partial net}\frac{\partial net}{\partial w_i} \quad \Rightarrow \quad \frac{\partial y}{\partial w_i} = (1 - y^2)x_i$$

General Hyperbolic Tangent Function

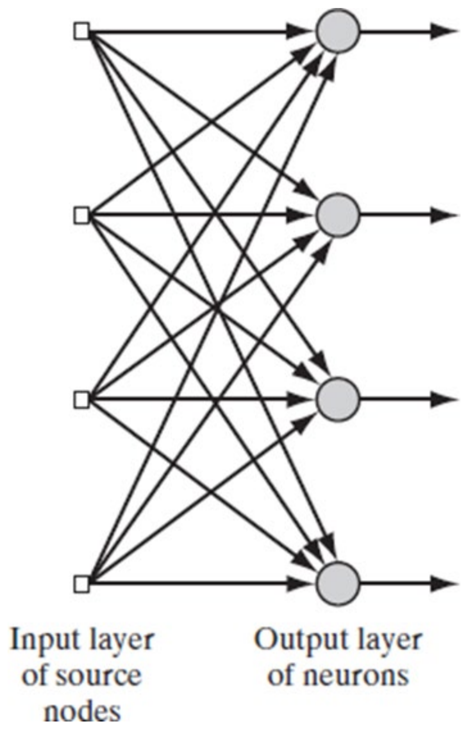
$$f(x) = \frac{e^{\alpha x} - e^{-\alpha x}}{e^{\alpha x} + e^{-\alpha x}} \qquad -1 \le f(x) \le 1$$

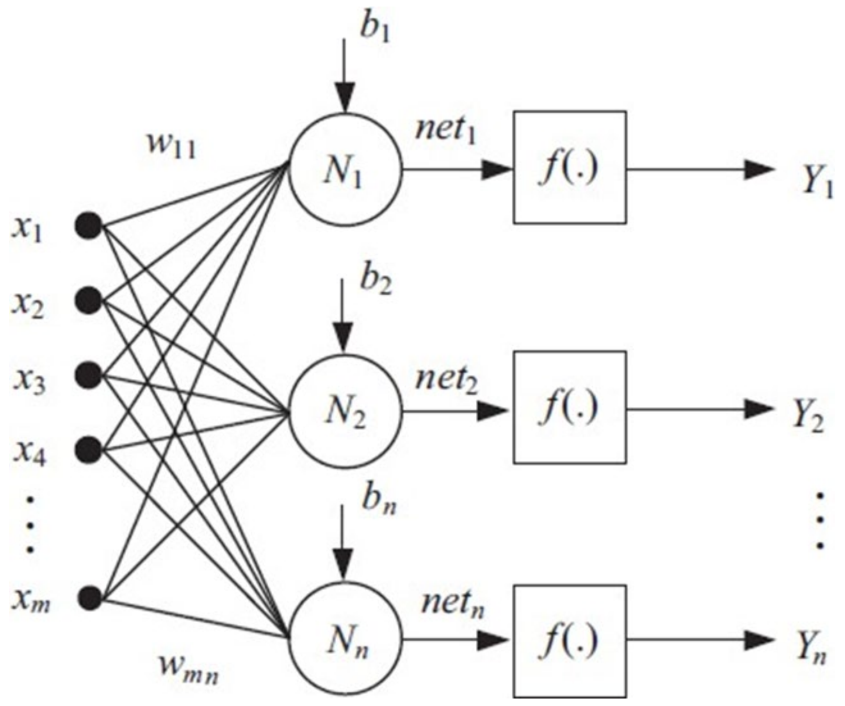


Multi-Layer Feed Forward NNs (FFNNs)

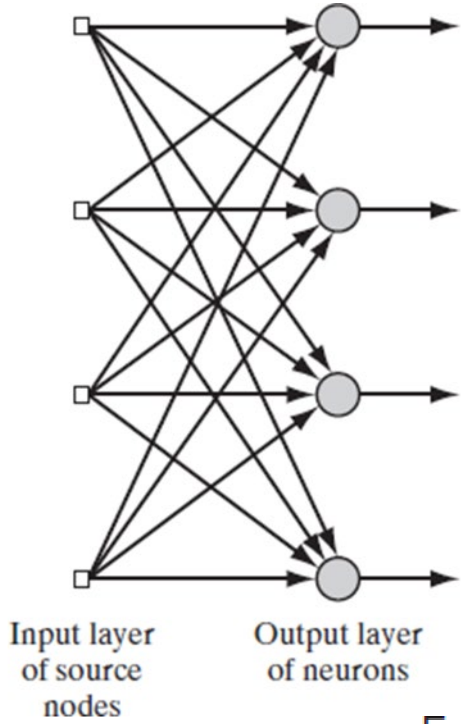


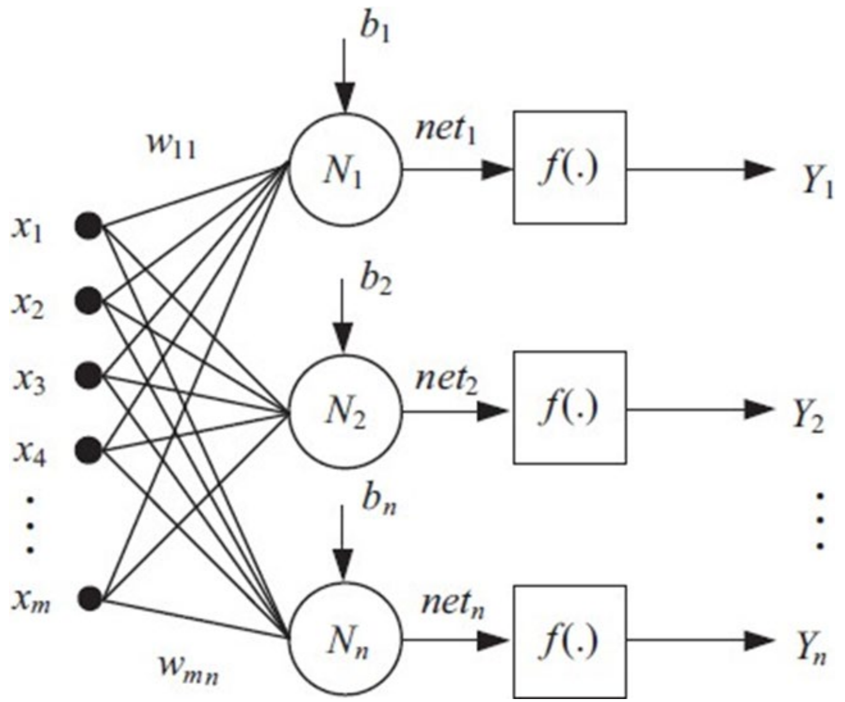
Multi-Layer Feed Forward NNs (FFNNs)





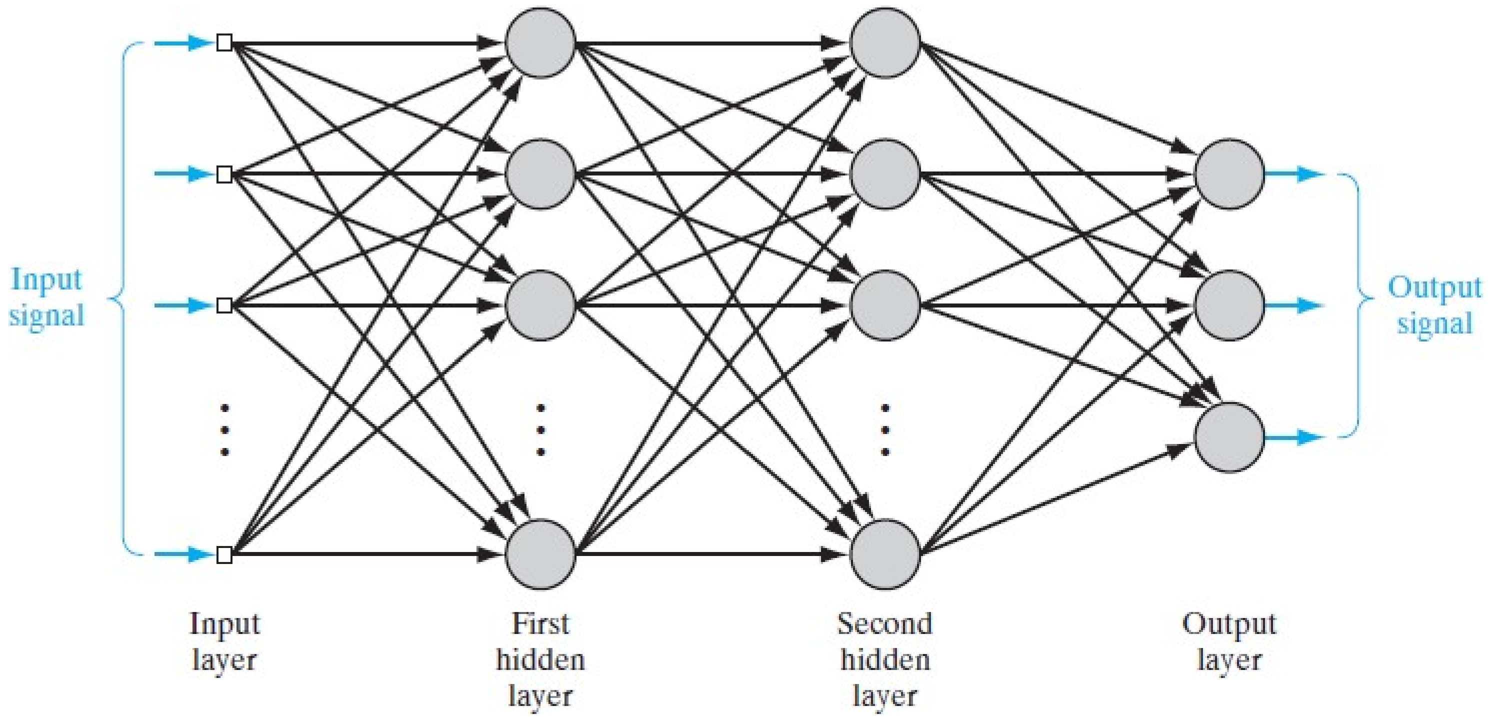
Multi-Layer Feed Forward NNs (FFNNs)



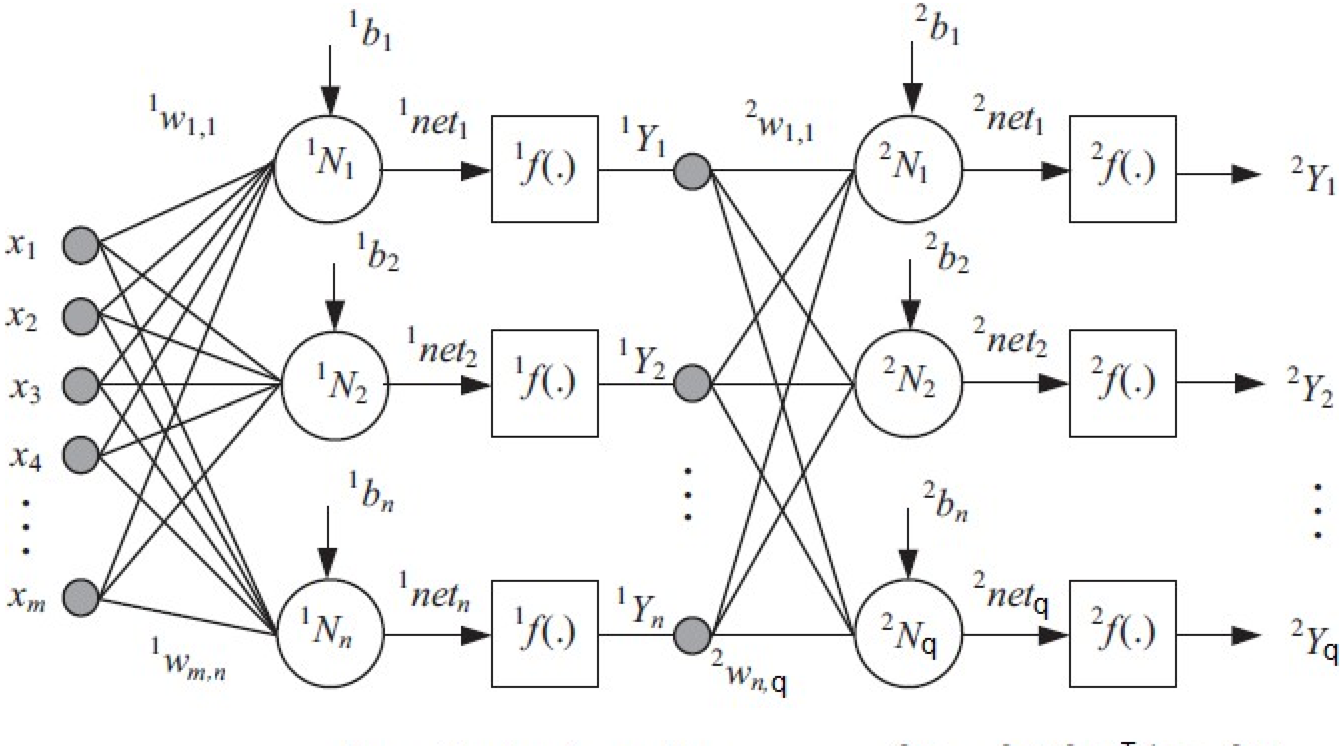


$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \qquad W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & \\ \vdots & \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \qquad Y = f(W^{\mathsf{T}} \cdot x + b)$$

Multi-Layer Feed Forward NNs (FFNNs)

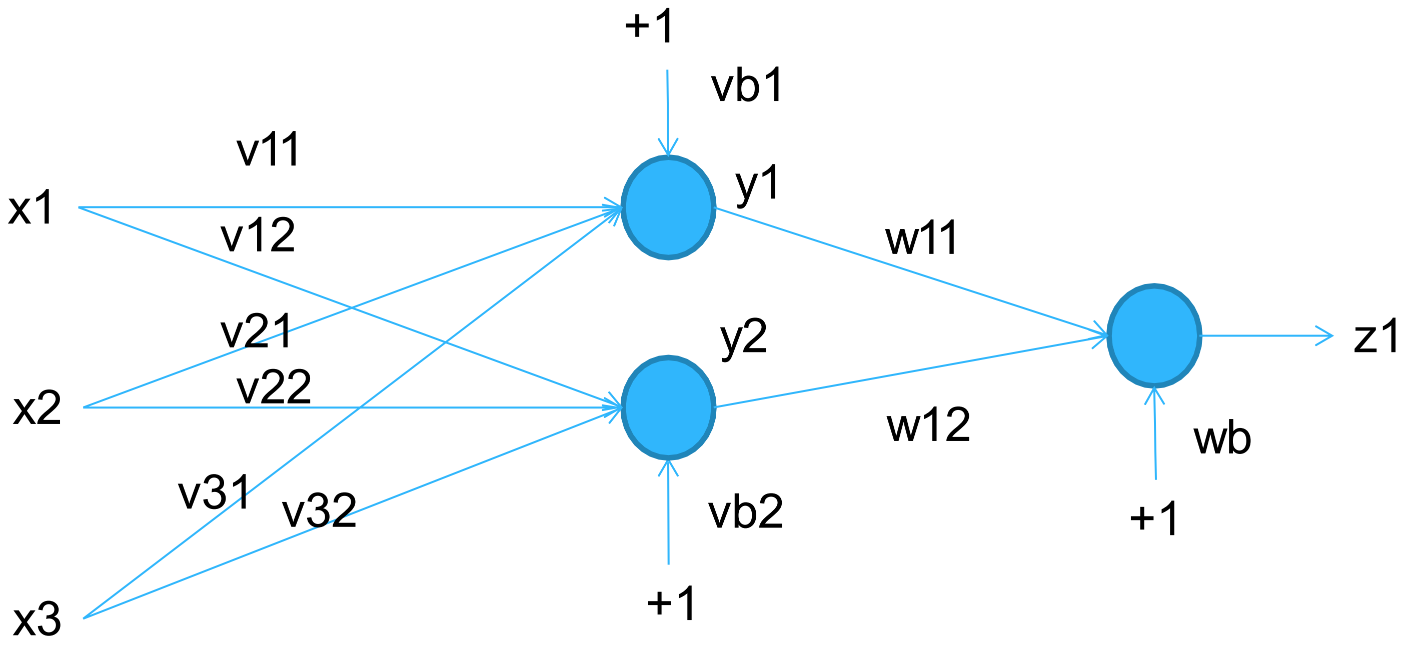


Multi-Layer Feed Forward NNs (FFNNs)



$^1Y = {}^1f({}^1W^T \cdot X + {}^1b)$ $\qquad$ $^2Y = {}^2f({}^2W^T \cdot {}^1Y + {}^2b)$

Multi-Layer Feed Forward NNs (FFNNs)

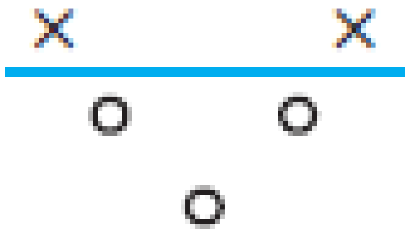


$$y_1 = f(v_{11}x_1 + v_{21}x_2 + v_{31}x_1 + v_{b1})$$

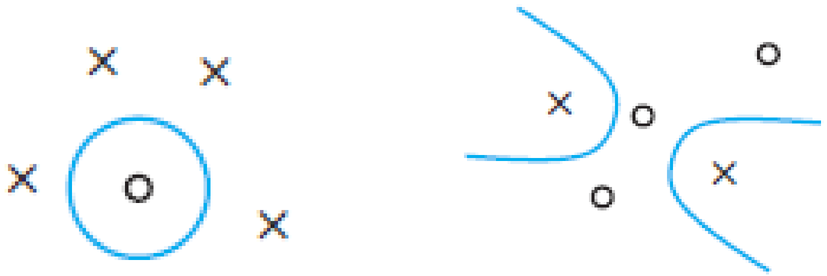
$$y_2 = f(v_{12}x_1 + v_{22}x_2 + v_{32}x_1 + v_{b2})$$

$$z_1 = f(w_{11}y_1 + w_{21}y_2 + w_b)$$

Linear vs. nonlinear separation



Linear Separation



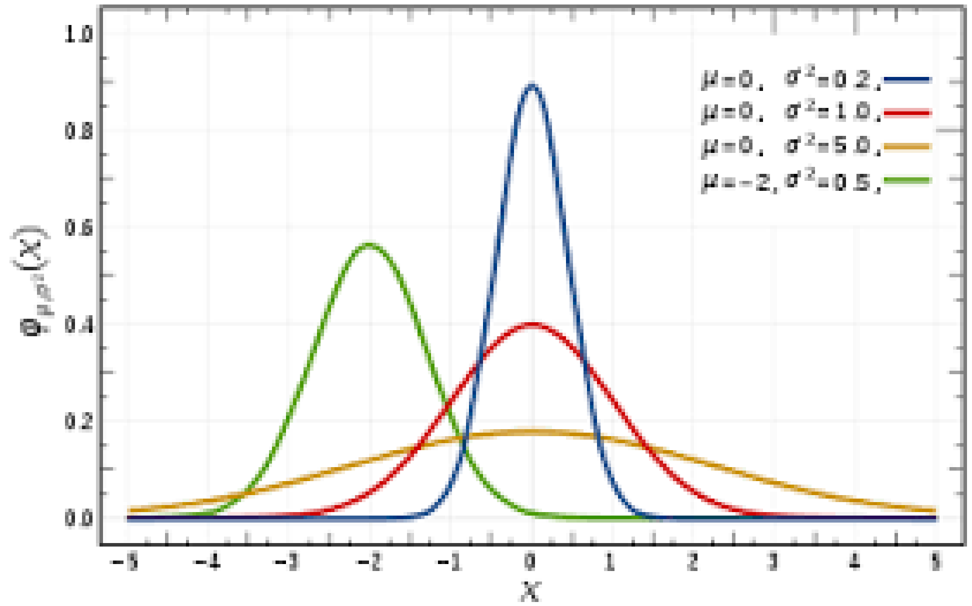
Nonlinear Separation

XOR is linearly separable ?

Gaussian Functions

In probability theory, a normal distribution is a continuous probability distribution for a real- valued random variable.

The general form of its probability density **function** is ... A random variable with a **Gaussian** distribution is said to be normally distributed and is called a normal deviate.



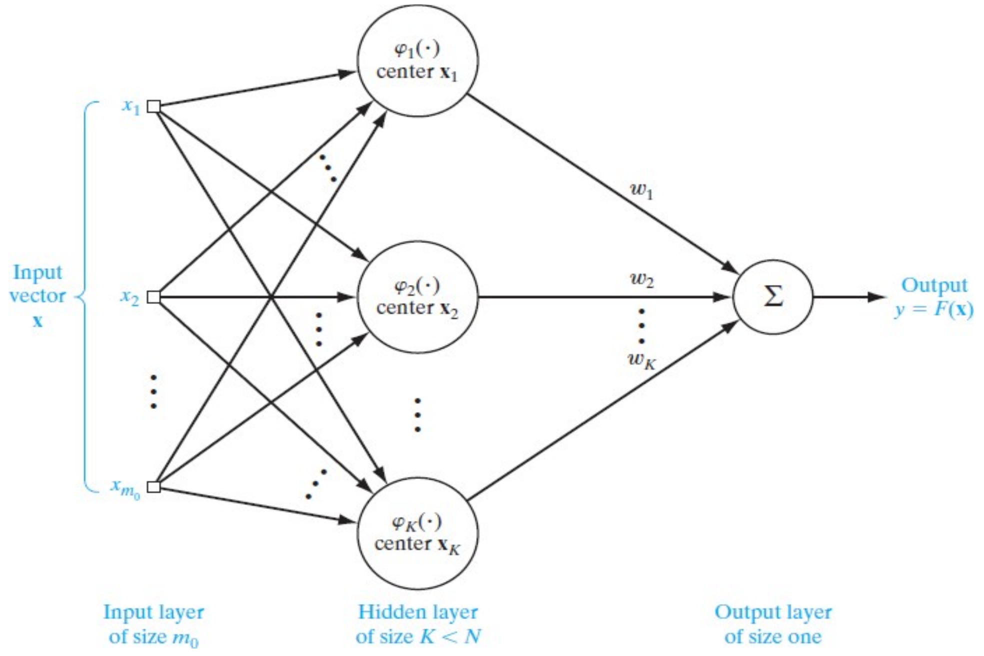
$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Radial Basis Functions

- RBF Functions use Gaussian-like functions
- RBF network can be used to perform complex pattern classification task and regression tasks also.
- Pattern Classification for nonlinear separation can be performed in two stages:
  - **Transform** nonlinear patterns into new linearly separated space
  - **Separate** the data using least-squares estimation

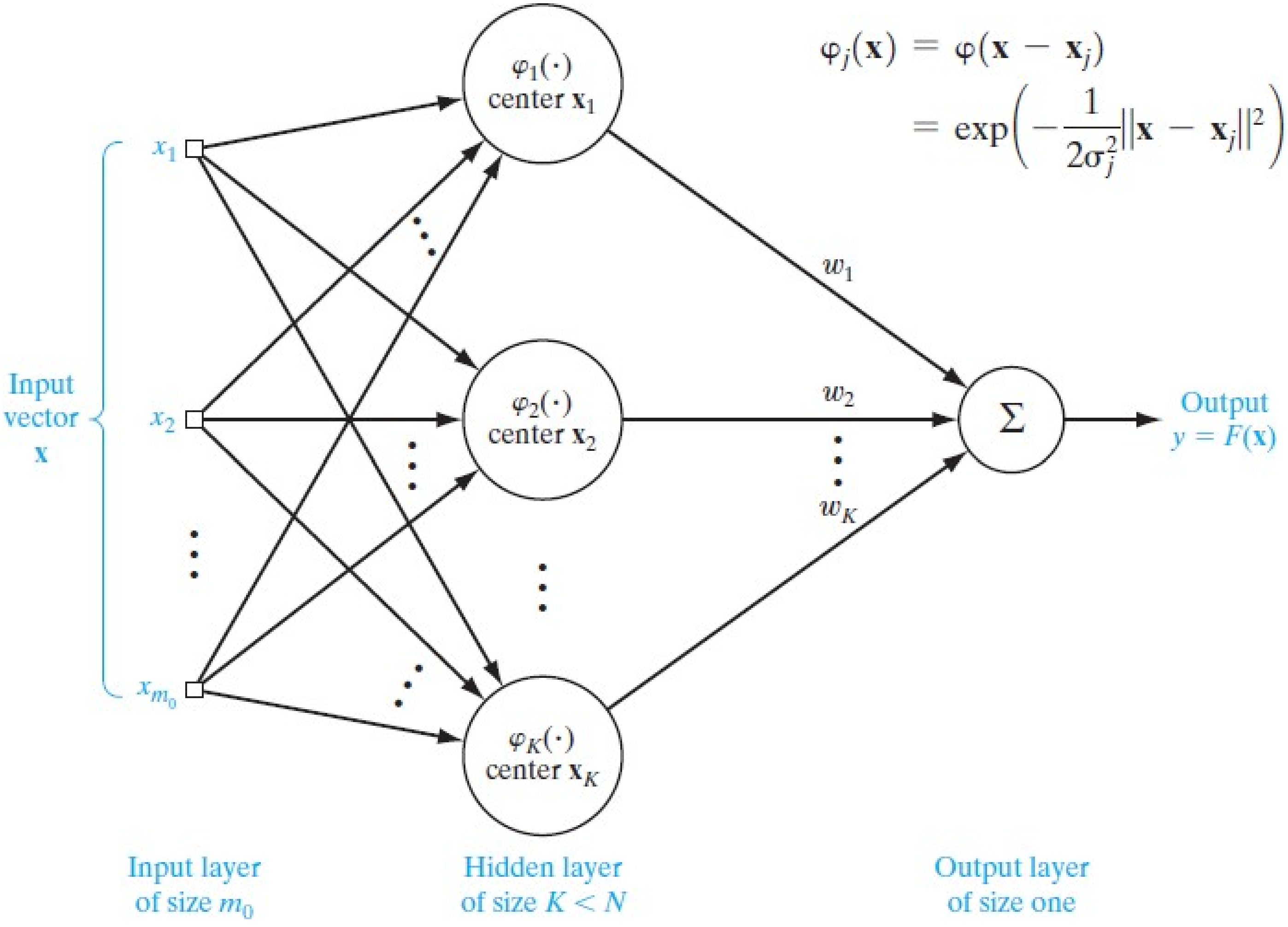
RBF Network

- RBF Network is composed of **three** layers
  - **Input layer** is made up of sensor inputs
  - **Hidden layer** applies nonlinear transformation from input space to hidden space
  - **Output layer** is linear

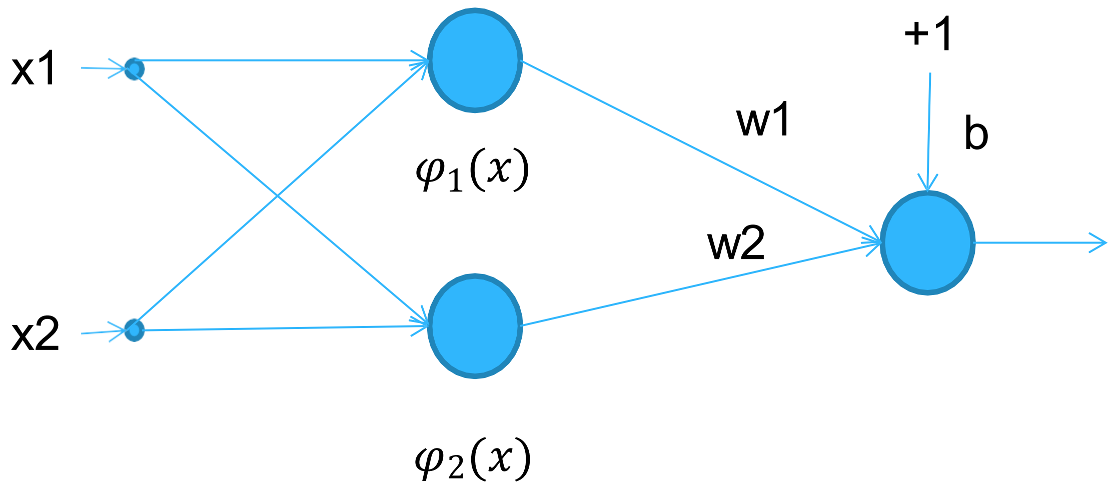


RBF Network Structure

$$\varphi_j(\mathbf{x}) = \varphi(\mathbf{x} - \mathbf{x}_j) = \exp\left(-\frac{1}{2\sigma_j^2}\|\mathbf{x} - \mathbf{x}_j\|^2\right), \qquad j = 1, 2, ..., N$$



Simple RBF Network

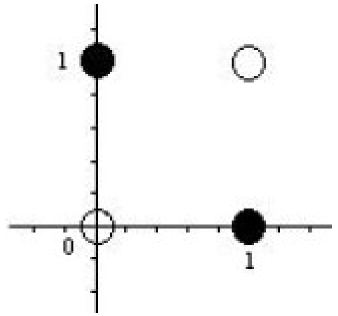


$$\varphi_i(x) = exp\left(-\frac{\|x - c_i\|^2}{2\sigma^2}\right)$$

$$y = w_1\varphi_1(x) + w_2\varphi_2(\mathrm{x}) + \mathrm{b}$$

XOR Problem Revisited

x2	x1	y
0	0	0
0	1	1
1	0	1
1	1	0



Linearly unseparable

$\varphi_1(x) = exp(-\|x - c_1\|^2)$ with centers *c1* = (1,1)

$\varphi_2(x) = exp(-\|x - c_2\|^2)$ with center *c2* = (0,0)

Pattern x	φ_1	φ_2
(0,0)	0.1353	1
(0,1)	0.3678	0.3678
(1,0)	0.3678	0.3678
(1,1)	1	0.1353

XOR Problem Revisited

Let w1 = w2 = -2.5, b = 2.84
c1=(1,1), c2=(0,0)

$$\varphi_i(x) = exp\left(-\frac{\|x - c_i\|^2}{2(0.5)}\right)$$

$$y = -2.5\ \varphi_1(x) - 2.5\ \varphi_2(\mathrm{x}) + 2.84$$

Input x=(0,1) or x=(1,0)

$$d_1 = (0-1)^2 + (1-1)^2 = 1$$
$$\varphi_1(x) = e^{-1} = 0.3678$$

$$d_2 = (0-0)^2 + (1-0)^2 = 1$$
$$\varphi_2(x) = e^{-1} = 0.3678$$

y = 2.84 - 2.5(0.3679) - 2.5(0.367) = **1**

Input x=(0,0) or x=(1,1)

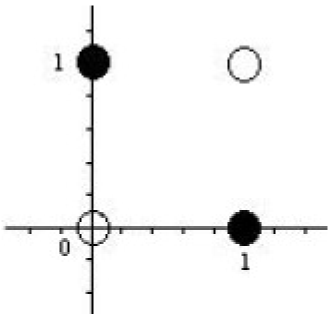
$$d_1 = (0-1)^2 + (0-1)^2 = 2$$
$$\varphi_1(x) = e^{-2} = 0.1353$$

$$d_2 = (0-0)^2 + (0-0)^2 = 0$$
$$\varphi_2(x) = e^0 = 1$$

y = 2.84 – 2.5(1) -2.5(0.1353) = **0**

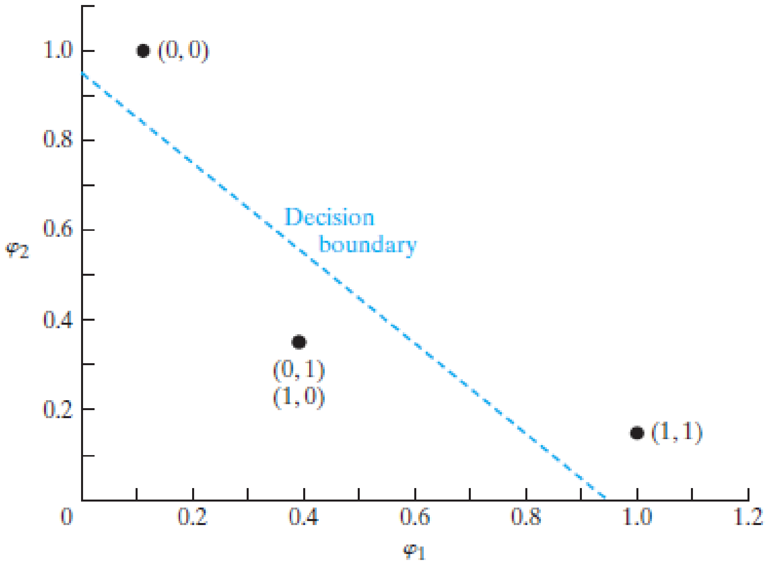
XOR Problem Re-visted

x2	x1	y
0	0	0
0	1	1
1	0	1
1	1	0



Transformation from input space to hidden space

Pattern x	φ_1	φ_2
(0,0)	0.1353	1
(0,1)	0.3678	0.3678
(1,0)	0.3678	0.3678
(1,1)	1	0.1353



conclusions

- Activation functions are used in neural networks to map data between different dimensions in nonlinear fashion

- The most commonly-used activation functions are: sigmoid and hyperbolic tangent

- Pros:
- Easy to differentiate
- Derivative can be formulated in closed-form
- Bounded
- Easy to adapt to different ranges

conclusions

- Feed-Forward Networks (FFN) have signals that flow in one direction only from input to output.
- A Multi-Layer FFN can have several hidden layers
- A Gaussian function is a normaly distributed function that is used in probability theory
- Radial Basis Functions use Gaussian-like functions
- RBF Network is composed of three layers: input, hidden, and output
- The input layer is the input signals
- hidden layer transforms the input-space to a hidden-space
- The output layer uses a linear activation function

References

- Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks, and Evolutionary Computing (Chapter 4), by N. Siddique and H. Adeli. Wiley Publication 2013

- Neural Networks and Learning Machine (Chapter 5) by Simon Haykin 3rd Edition. Pearson 2009