PES UNIVERSITY ONLINE

ARTIFICIAL NEURAL NETWORK

Unit-2: Perceptron

Ms. Swetha R.
Department of Electronics and
Communication Engineering
PES University

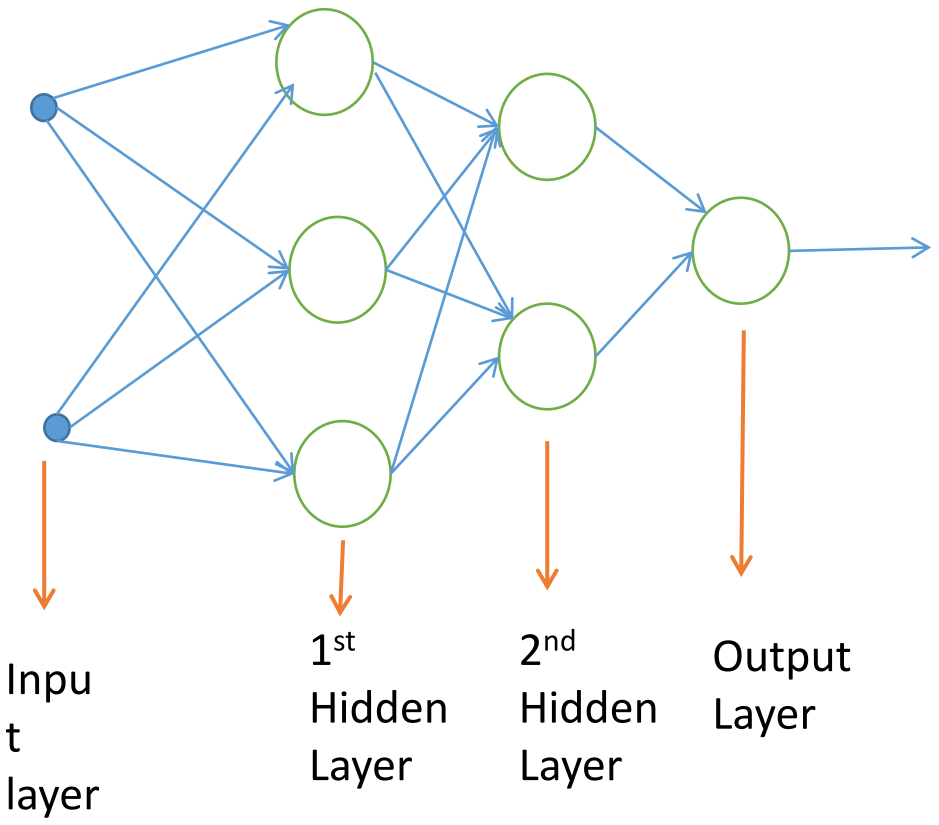
CONTENT:Part-1

1. *Perceptron*
   1. *Introduction-Linearly Separable*
   2. *Rosenblatt Algorithm with example*
   3. *Perceptron Convergence Theorem*
2. *Single Layer Perceptron*

   DrawBack: Xor Logic Gate
3. *Multilayer Perceptron*
   1. *Backpropogation Algorithm*
   2. *Example: XOR Logic Gate*

Artificial Neural Network: Multi-layer Perceptron

Back-Propogation Algorithm (BPA)

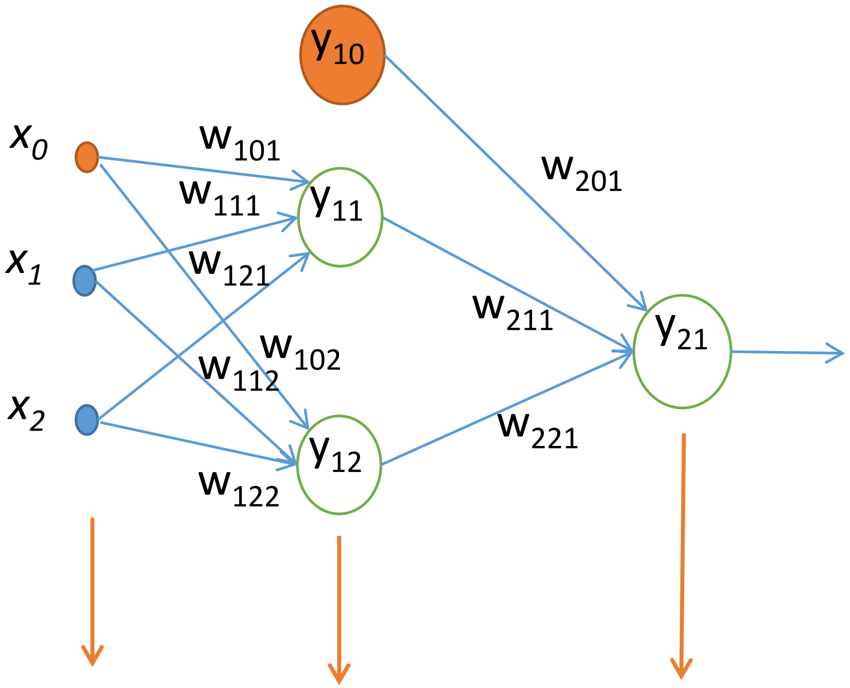


⇒ This Multilayer Perceptron is represented as 2:3:2:1

Artificial Neural Network: Multi-layer Perceptron

Back-Propogation Algorithm (BPA)

Consider a neural network with 2:2:1 as shown below

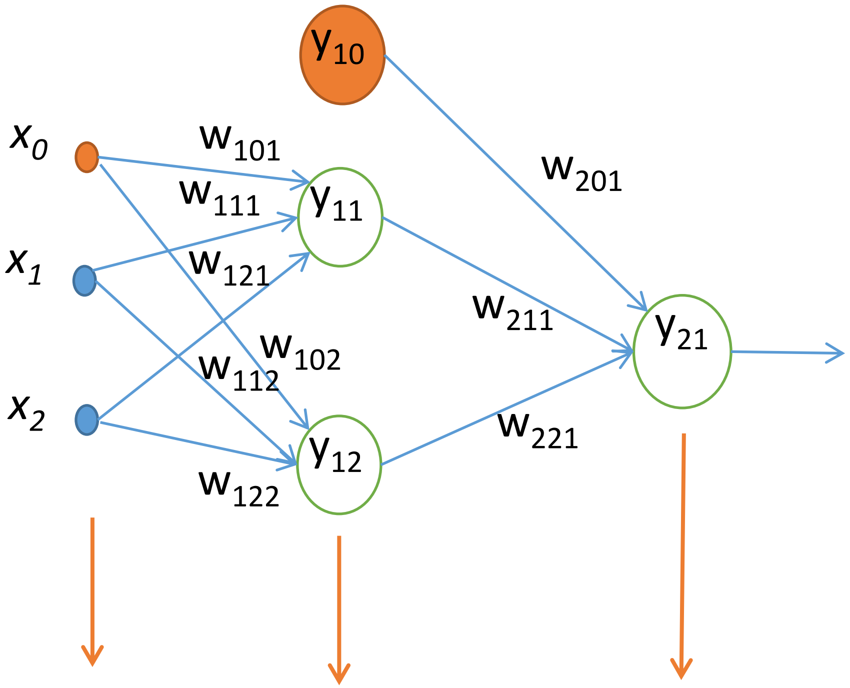


Layer	0	1	2
Index	m_0	m_1	m_2

Artificial Neural Network: Multi-layer Perceptron

Back-Propogation Algorithm (BPA)

Consider a neural network with 2:2:1 as shown below



Layer	0	1	2
Index	m_0=2	m_1=2	m_2=1

Artificial Neural Network: Multi-layer Perceptron

Back-Propogation Algorithm (BPA)

- Procedure:

Step 1: Initialize the weight vector all layers

Step 2: Computation takes place at 2 stages

- Feedforward Pass
  - Compute output
- Backward Pass
  - Compute local gradient

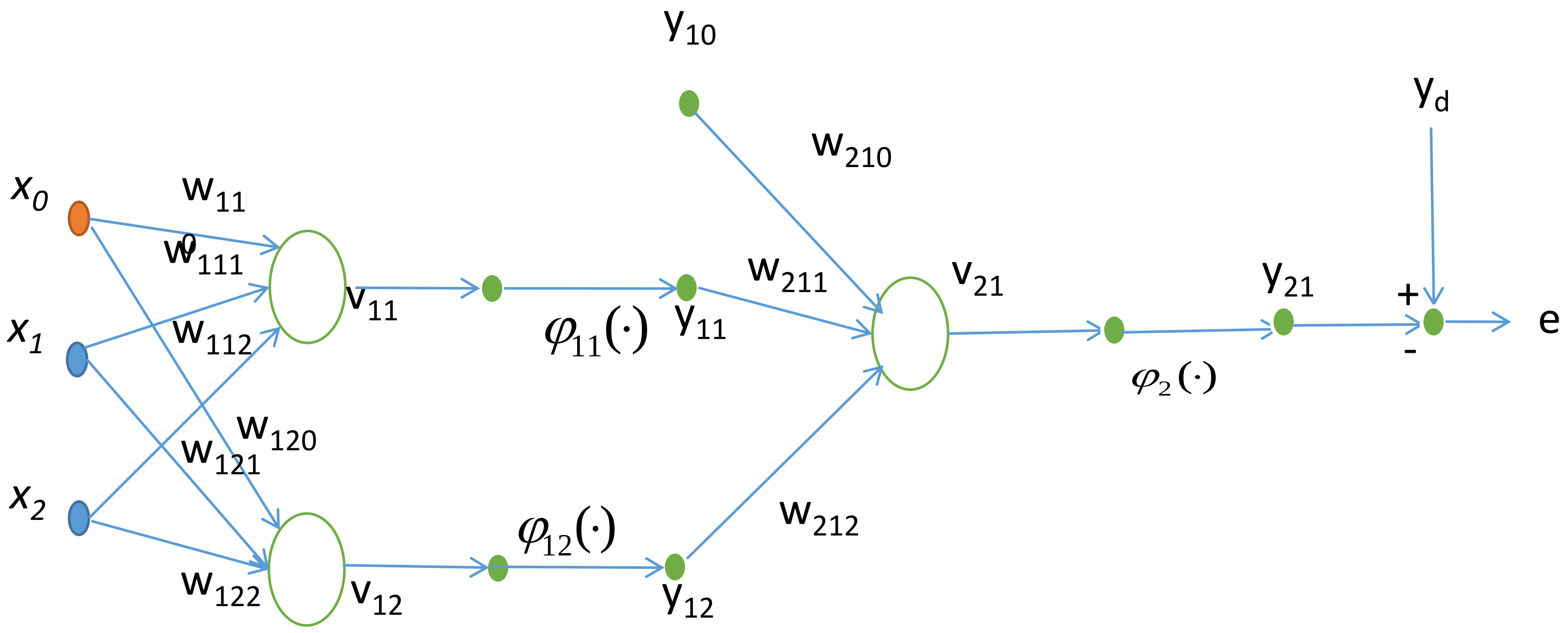
$$W = \begin{pmatrix} w \\ \\ \end{pmatrix}$$

Step 3: Update the weight vectors if required

Artificial Neural Network: Multi-layer Perceptron

Back-Propogation Algorithm (BPA)

Feedforward Pass:



Layer: p	0	1	2
Index	i--> 0:(m_0=2)	j--> 0:(m_1=2)	l--> 0:(m_2=1)

Artificial Neural Network: Multi-layer Perceptron

Back-Propogation Algorithm (BPA)

Layer 0: Input Layer

$$X(k) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad y_0(k) = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$$

Layer 1: 1st Hidden Layer

Input : $y_0(k) = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$

Weight Vector: $W_1 = \begin{pmatrix} w_{110} & w_{111} & w_{112} \\ w_{120} & w_{121} & w_{122} \end{pmatrix}$

Induced local field:

$$v_1 = \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} = W_1(k) y_0(k)$$

Output of activation block:

$$\varphi_1(v_1) = \begin{pmatrix} \varphi_{11}(v_{11}) \\ \varphi_{12}(v_{12}) \end{pmatrix}$$

Output of 1st Hidden Layer:

$$\overline{y}_1 = \begin{pmatrix} \varphi_{11}(v_{11}) \\ \varphi_{12}(v_{12}) \end{pmatrix} = \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix}$$

Artificial Neural Network: Multi-layer Perceptron

Back-Propogation Algorithm (BPA)

Layer 2: Output Layer

Input : $y_1(k) = \begin{pmatrix} 1 \\ \bar{y}_1(k) \end{pmatrix}$

Weight Vector: $W_2 = \begin{pmatrix} w_{210} & w_{211} & w_{211} \end{pmatrix}$

Induced local field:

$$v_2(k) = w_2(k)y_1(k)$$

Output of activation block:

$$\varphi_2(v_2(k)) = \varphi_{21}(v_2(k))$$

Output:

$$y_2 = \varphi_2(v_2(k))$$

Error:

$$e_1 = y_d - y_{21}$$

PES UNIVERSITY ONLINE

THANK YOU

Ms. Swetha R.

Department of Electronics and Communication Engineering

swethar@pes.edu

+91 80 2672 1983 Extn 753