Introduction to Backpropagation Learning Algorithm

Jeen-Shing Wang
Wei-Chun Chiang
Department of Electrical Engineering
National Cheng Kung University
2016/10/24

NCKU Computational Intelligence & Learning Systems LAB

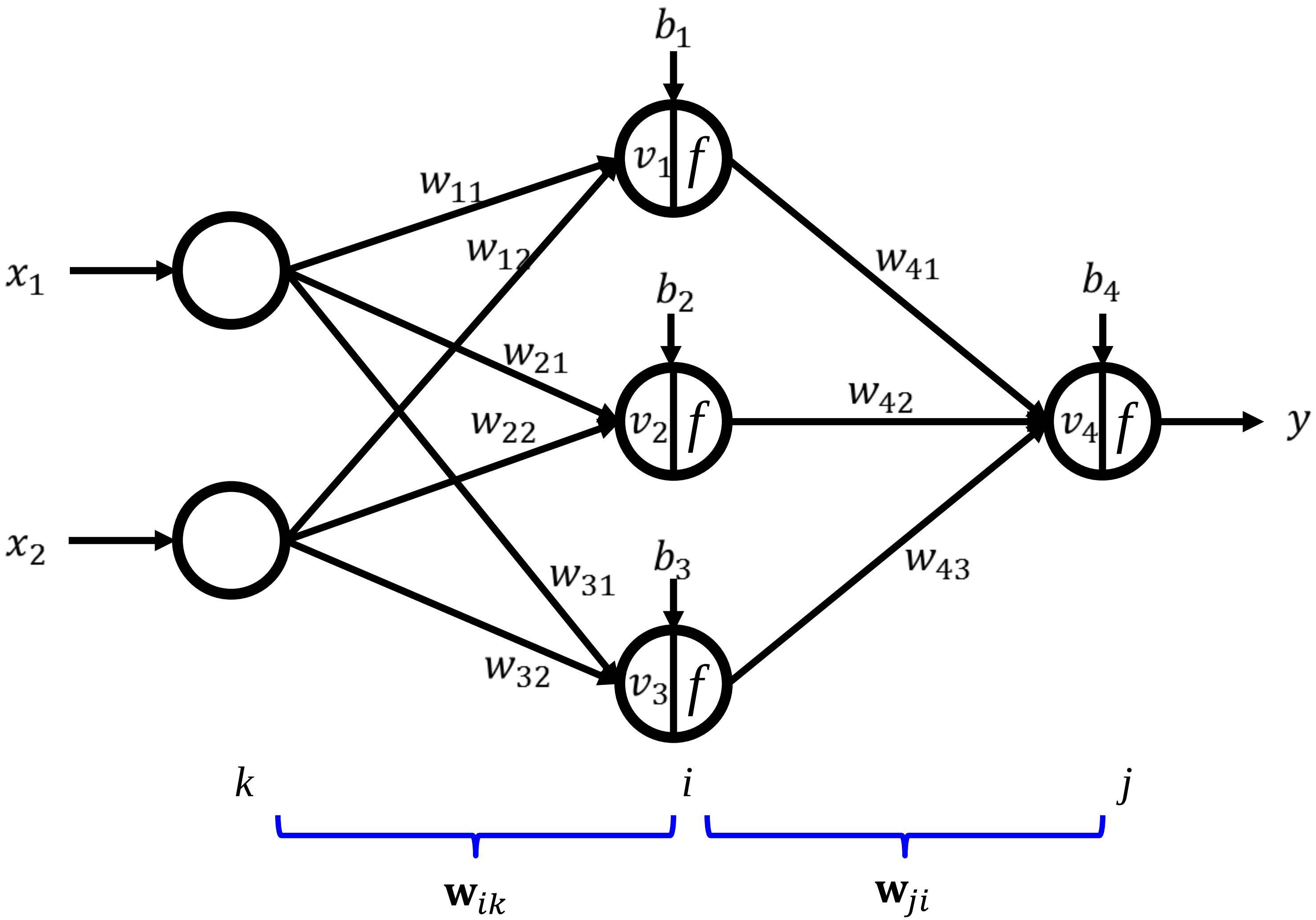
Data Collection

$$y = x_1^2 + x_2^2$$

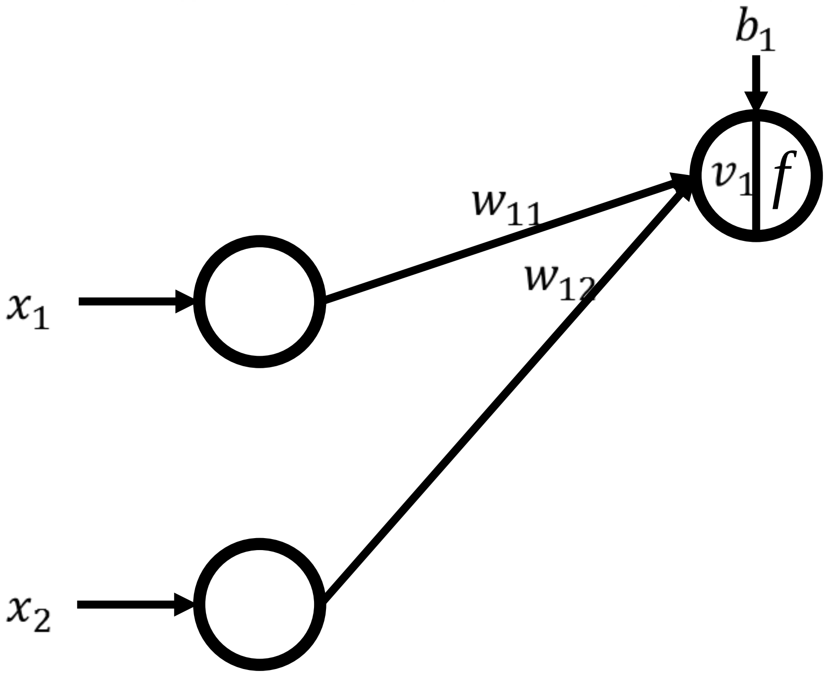
```
clc;close all;clear all;
%% Generate training and testing data
x_1 = rand(1,1000);
x_2 = rand(1,1000);
x = [x_1;x_2];
y = x_1.^2+x_2.^2;

% Training data
train_input = x(:,1:800);
train_output = y(:,1:800);

% Testing data
test_input = x(:,801:end);
test_output = y(:,801:end);
```

```
figure(1)
subplot(6,1,1)
plot(train_input(1,:));
subplot(6,1,2)
plot(train_input(2,:));
subplot(6,1,3)
plot(train_output);
subplot(6,1,4)
plot(test_input(1,:));
subplot(6,1,5)
plot(test_input(2,:));
subplot(6,1,6)
plot(test_output);
```

Topology of a Feedforward Neural Network



Forward Path

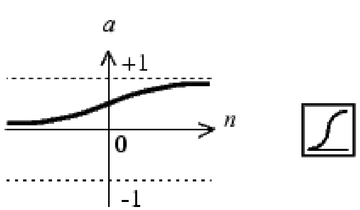


$$\boldsymbol{v_1 = w_{11}x_1 + w_{12}x_2} + b_1$$

Activation function:

$$f(x) = \frac{1}{1+e^{-x}} \qquad f(v_1) = \frac{1}{1+e^{-v_1}}$$

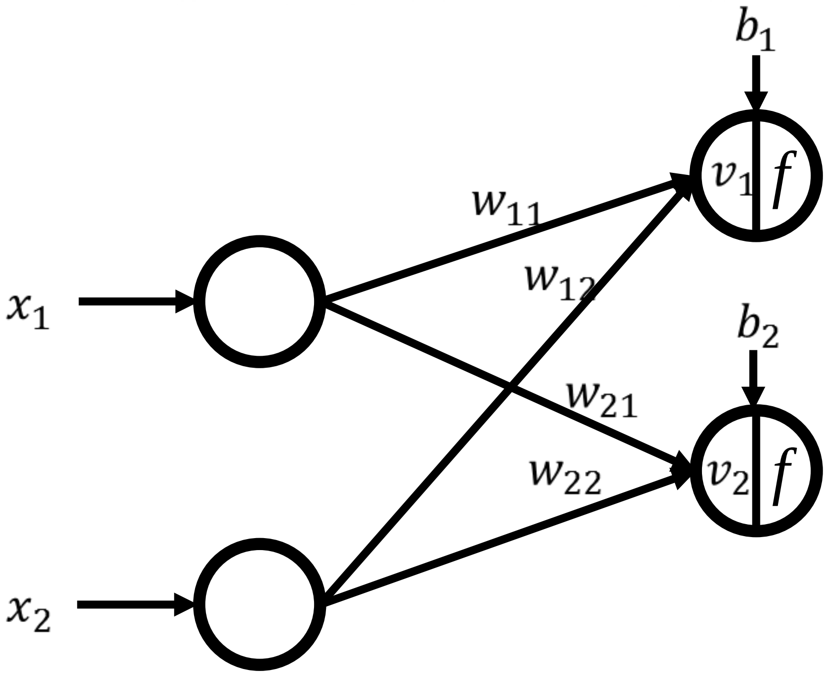
$$f'(x) = f(x)\big(1 - f(x)\big) \qquad f'(v_1) = f(v_1)\big(1 - f(v_1)\big)$$



$a = logsig(n)$

Log-Sigmoid Transfer Function

Forward Path



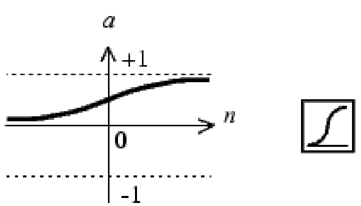
$$v_1 = w_{11}x_1 + w_{12}x_2 + b_1$$
$$y_{h1} = f(v_1) = \frac{1}{1+e^{-v_1}}$$

$$\boldsymbol{v_2 = w_{21}x_1 + w_{22}x_2} + b_2$$

Activation function:

$$f(x) = \frac{1}{1+e^{-x}} \quad\Downarrow\quad f'(x) = f(x)\big(1-f(x)\big)$$

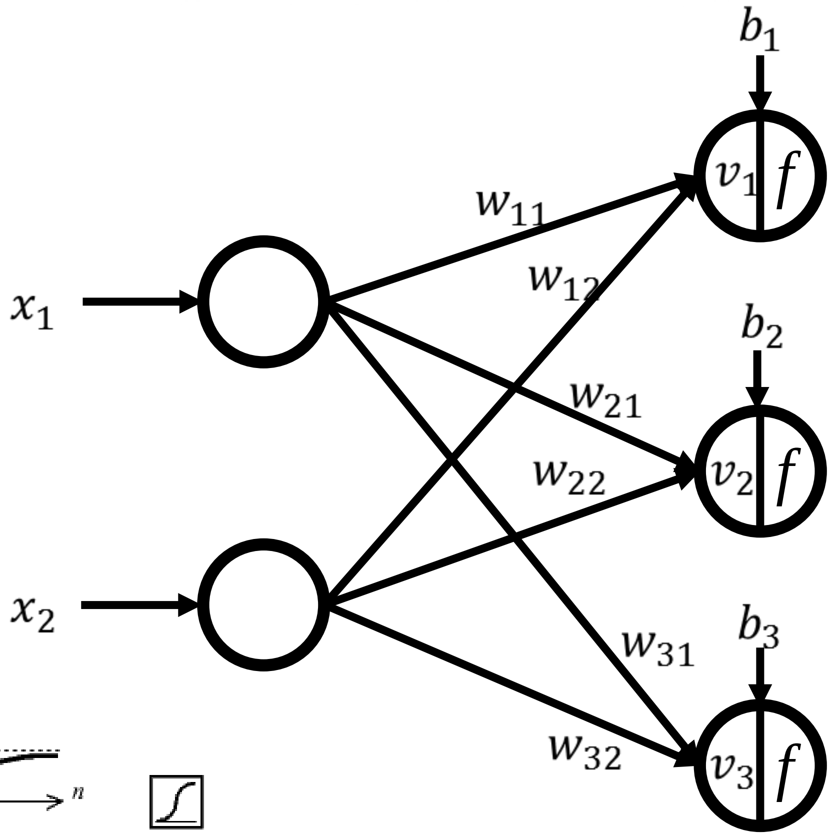
$$f(v_2) = \frac{1}{1+e^{-v_2}} \quad\Downarrow\quad f'(v_2) = f(v_2)\big(1-f(v_2)\big)$$



$a = logsig(n)$

Log-Sigmoid Transfer Function

Forward Path

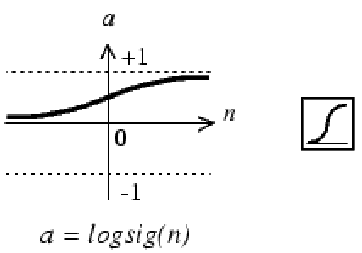


$$v_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

$$y_{h1} = f(v_1) = \frac{1}{1+e^{-v_1}}$$

$$v_2 = w_{21}x_1 + w_{22}x_2 + b_2$$

$$y_{h2} = f(v_2) = \frac{1}{1+e^{-v_2}}$$



Log-Sigmoid Transfer Function

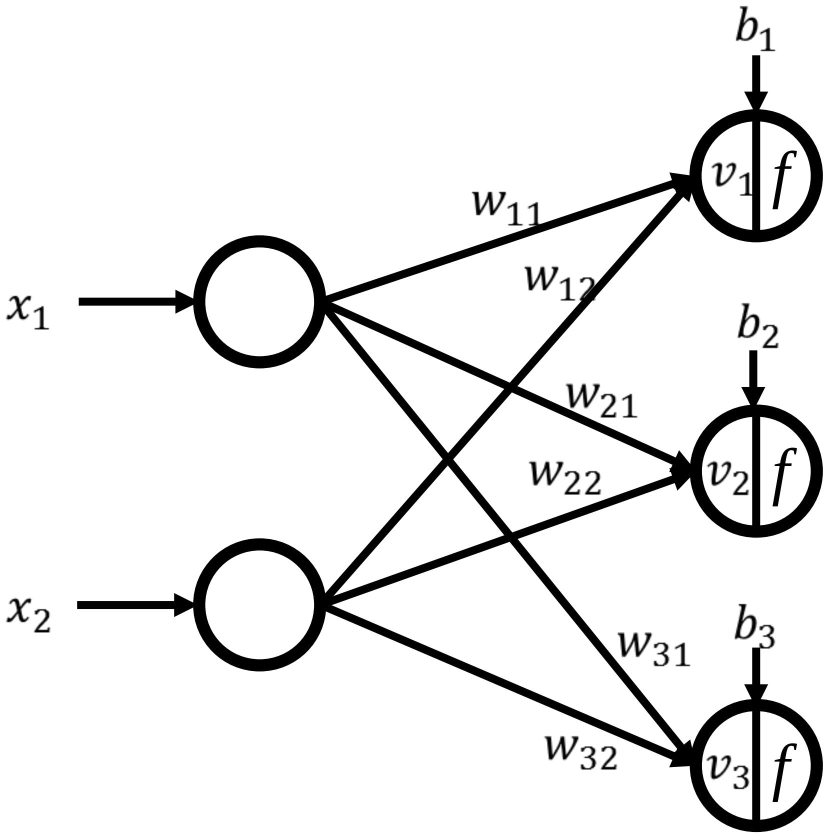
$$\boldsymbol{v_3 = w_{31}x_1} + \boldsymbol{w_{32}x_2} + b_3$$

Activation function:

$$f(x) = \frac{1}{1+e^{-x}} \qquad f(v_3) = \frac{1}{1+e^{-v_3}}$$

$$f'(x) = f(x)\big(1-f(x)\big) \qquad f'(v_3) = f(v_3)\big(1-f(v_3)\big)$$

Forward Path



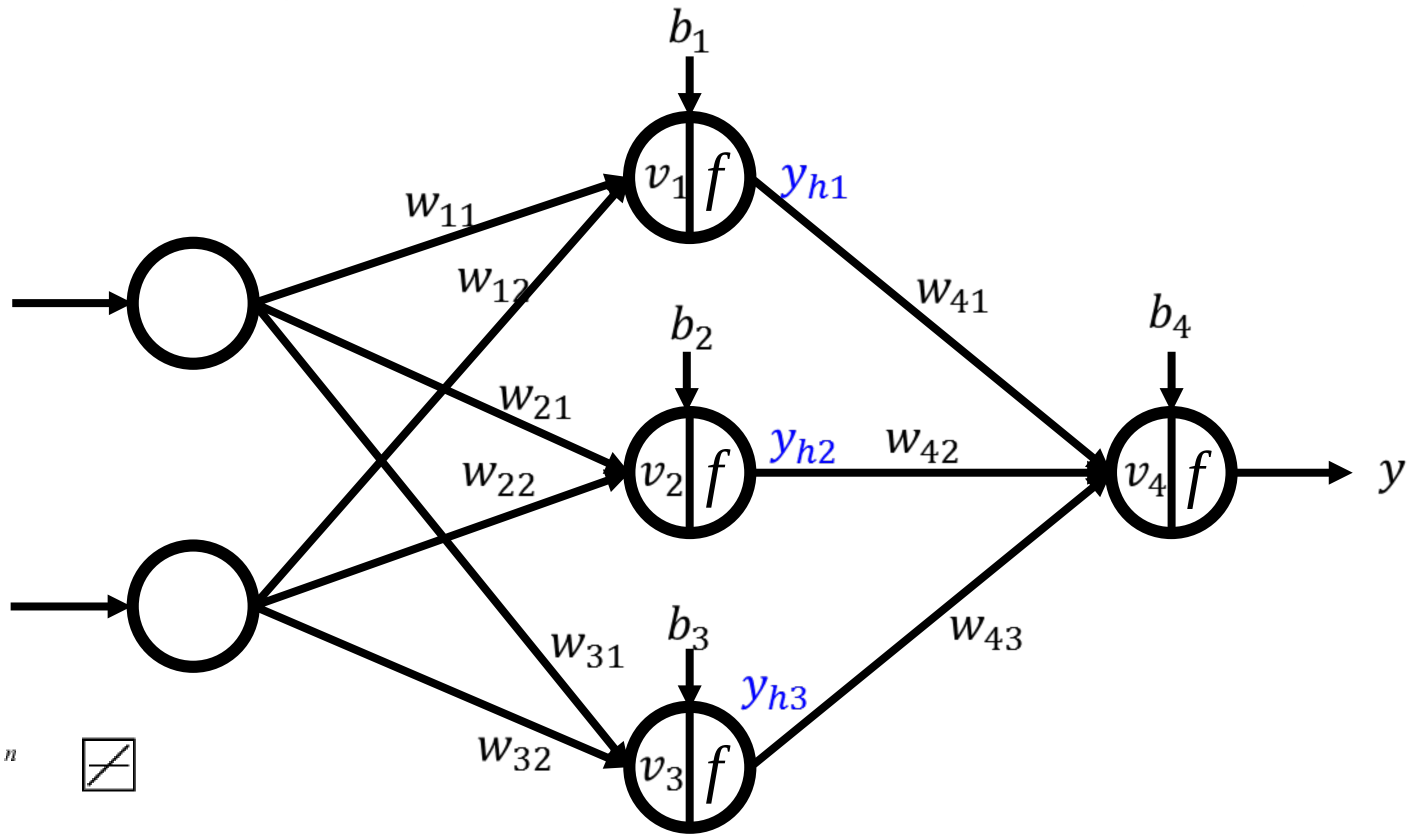
$$v_1 = w_{11}x_1 + w_{12}x_2 + b_1$$
$$y_{h1} = f(v_1) = \frac{1}{1+e^{-v_1}}$$

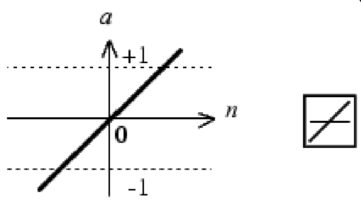
$$v_2 = w_{21}x_1 + w_{22}x_2 + b_2$$
$$y_{h2} = f(v_2) = \frac{1}{1+e^{-v_2}}$$

$$v_3 = w_{31}x_1 + w_{32}x_2 + b_3$$
$$y_{h3} = f(v_3) = \frac{1}{1+e^{-v_3}}$$

$$v_i = \sum_k w_{ik}x_k + b_i \qquad y_{hi} = f(v_i) = \frac{1}{1+e^{-v_i}} \qquad f'(v_i) = f(v_i)\big(1 - f(v_i)\big)$$

Forward Path





$a = purelin(n)$

Linear Transfer Function

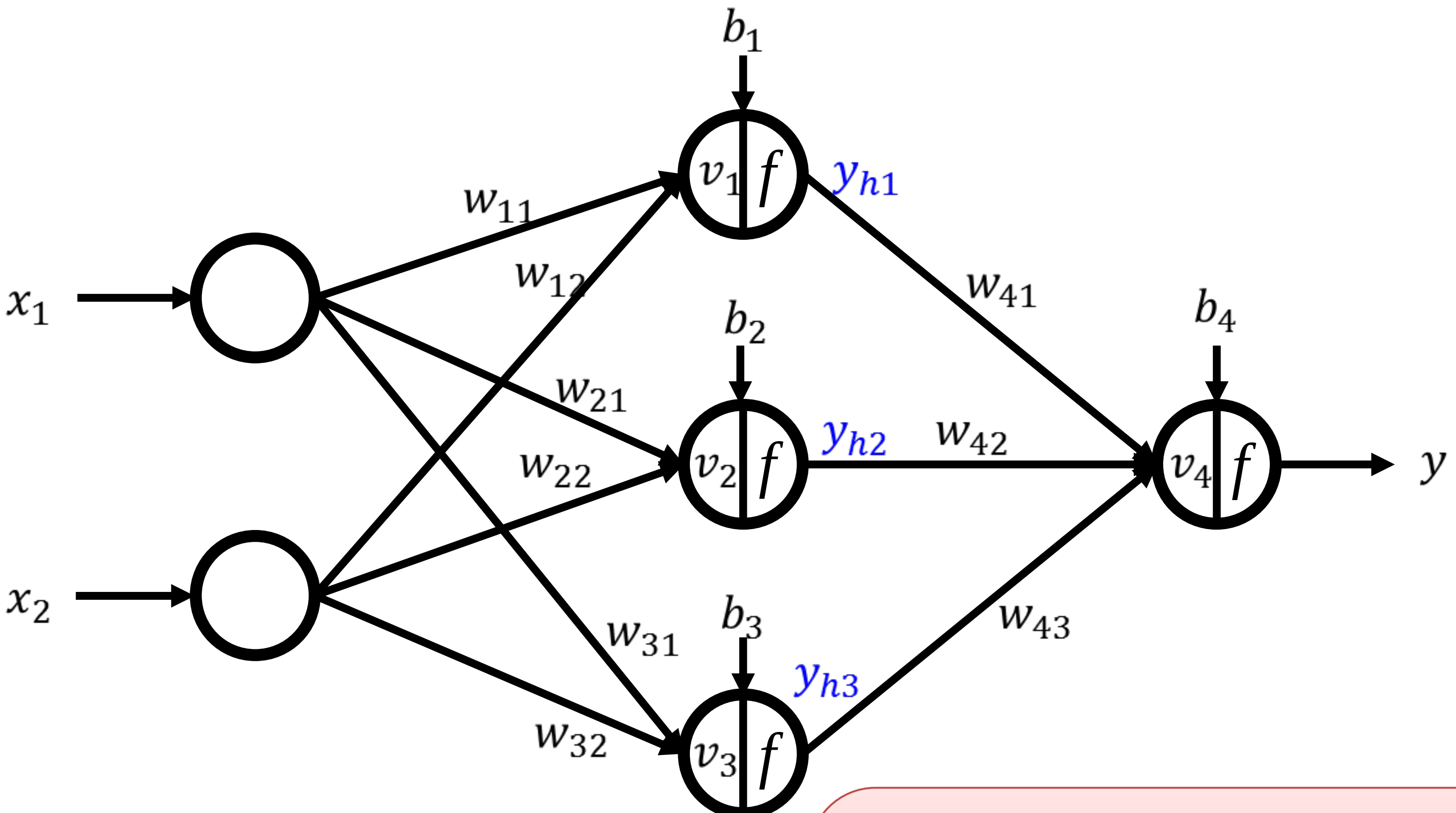
$$\boldsymbol{v_4 = w_{41}y_{h1} + w_{42}y_{h2} + w_{43}y_{h3}} + b_4$$

$$v_4 = w_{41}f(v_1) + w_{42}f(v_2) + w_{43}f(v_3) + b_4$$

Activation function: $f(x) = x$ $\quad f(v_4) = v_4$

$$f'(x) = 1 \qquad f'(v_4) = 1$$

Forward Path



$$v_4 = w_{41} f(v_1) + w_{42} f(v_2) + w_{43} f(v_3) + b_4$$

$$v_4 = w_{41} y_{h1} + w_{42} y_{h2} + w_{43} y_{h3} + b_4$$

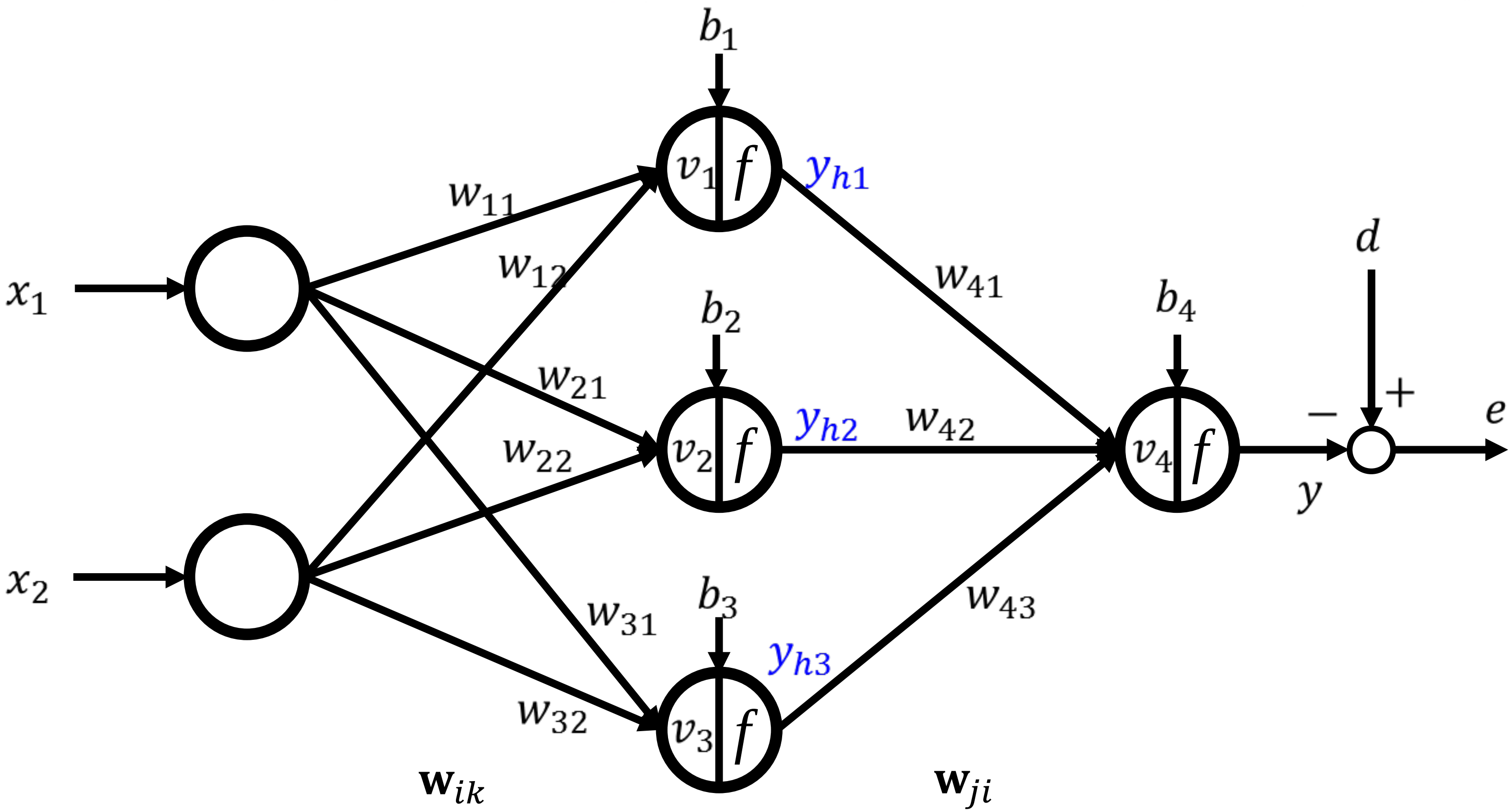
$$y = f(v_4) = v_4$$

$$v_j = \sum_i w_{ji} y_{hi} + b_j = \sum_i w_{ji} f(v_i) + b_j$$

$$y_j = f(v_j) = v_j$$

$$f'(v_j) = 1$$

Forward Path



Cost function:

$$e_j(n) = d_j(n) - y_j(n)$$

$$E_j(n) = \frac{1}{2} e_j(n)^2 = \frac{1}{2}\left(d_j(n) - y_j(n)\right)^2$$

Forward Path

- **Learning condition**
  - ✓ Learning m epochs:
    For all training data
    $$While\ (epoch\ <\ m)$$
  - ✓ Train n training data:
    $$x_k = train\_input_k(n)$$
    $$desire\_output_j$$
    $$= train\_output_j(n)$$
- **Input layer**
  - ✓ Input neurons k, given input x_k:
    For each input neuron
    $$output_k = x_k$$
- **Hidden layer**
  - ✓ Hidden neurons i:
    For each hidden neuron
    $$v_i = \sum_k w_{ik}\, output_k + b_i$$
    $$output_i = logsig(v_i)$$
- **Output layer**
  - ✓ Output neurons j:
    For each output neuron
    $$v_j = \sum_i w_{ji}\, output_i + b_j$$
    $$output_j = v_j$$
- **Output error**
  - ✓ Output neurons j:
    For each output neuron
    $$E_j = \frac{1}{2} error_j^2$$
    $$error_j = desire\_output_j - output_j$$

Backward Path

(1) Update rule for the weights of the output neurons:

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n)$$

$$= w_{ji}(n) - \eta \frac{\partial E_j(n)}{\partial w_{ji}(n)}$$

$$= w_{ji}(n) - \eta \frac{\partial E_j(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

$$= w_{ji}(n) - \eta e_j(n)(-1) f'\left(v_j(n)\right) f\left(v_i(n)\right)$$

$$= w_{ji}(n) - \eta e_j(n)(-1)(1) f\left(v_i(n)\right)$$

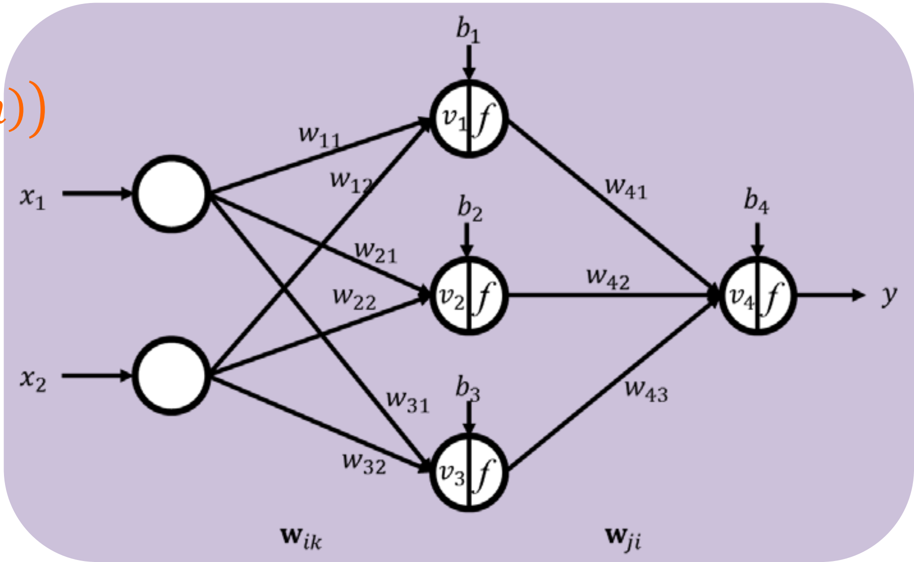
$$= w_{ji}(n) + \eta \left(d_j(n) - y_j(n)\right) f\left(v_i(n)\right)$$

$$E_j(n) = \frac{1}{2} e_j(n)^2$$

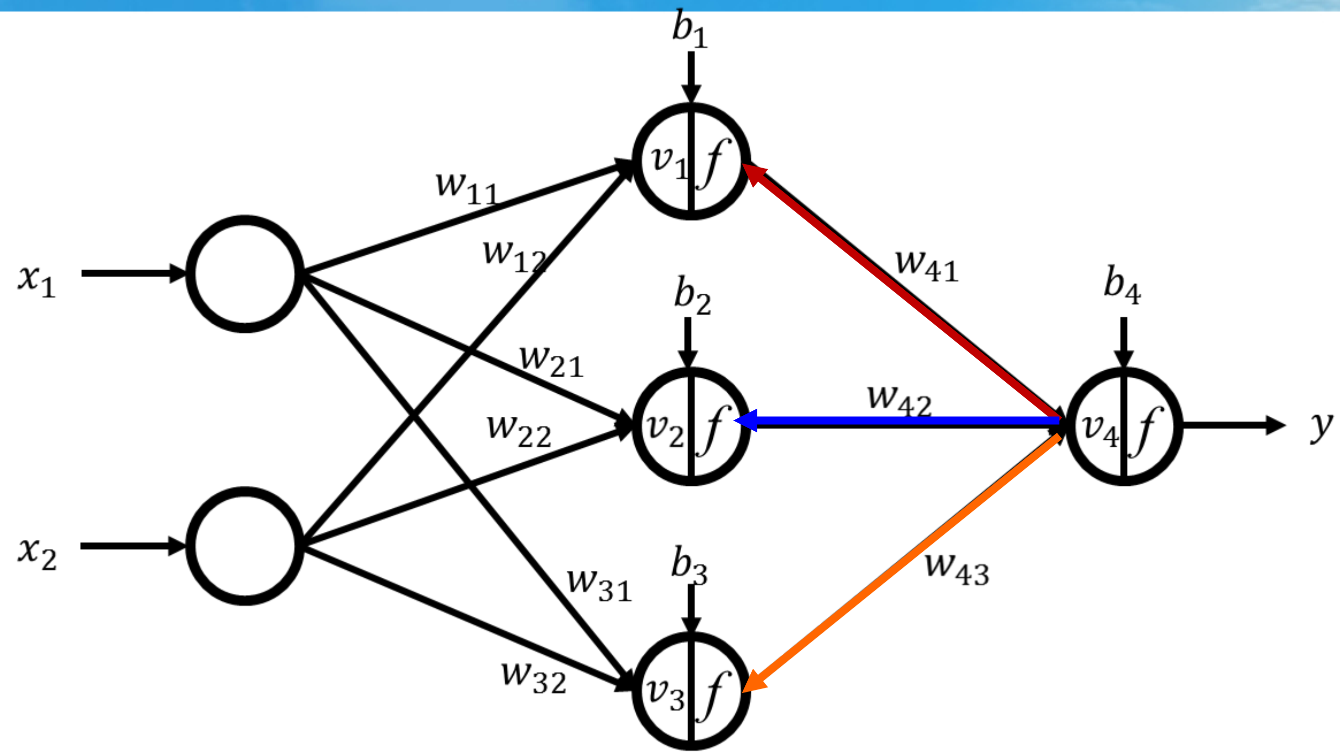
$$e_j(n) = d_j(n) - y_j(n)$$

$$y_j = f(v_j) = v_j$$

$$v_j = \sum_i w_{ji} f(v_i) + b_j$$



Backward Path



$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) = w_{ji}(n) - \eta \frac{\partial E_j(n)}{\partial w_{ji}(n)} = w_{ji}(n) + \eta \left(d_j(n) - y_j(n)\right) f\left(v_i(n)\right)$$

$$w_{41}(n+1) = w_{41}(n) + \Delta w_{41}(n) = w_{41}(n) + \eta\left(d(n) - y(n)\right) f\left(v_1(n)\right)$$

$$w_{42}(n+1) = w_{42}(n) + \Delta w_{42}(n) = w_{42}(n) + \eta\left(d(n) - y(n)\right) f\left(v_2(n)\right)$$

$$w_{43}(n+1) = w_{43}(n) + \Delta w_{43}(n) = w_{43}(n) + \eta\left(d(n) - y(n)\right) f\left(v_3(n)\right)$$

Backward Path

(2) Update rule for the biases of the output neurons:

$$b_j(n+1) = b_j(n) + \Delta b_j(n)$$

$$= b_j(n) - \eta \frac{\partial E_j(n)}{\partial b_j(n)}$$

$$= b_j(n) - \eta \frac{\partial E_j(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial b_j(n)}$$

$$= b_j(n) - \eta e_j(n)(-1) f'\left(v_j(n)\right)(1)$$

$$= b_j(n) - \eta e_j(n)(-1)(1)(1)$$

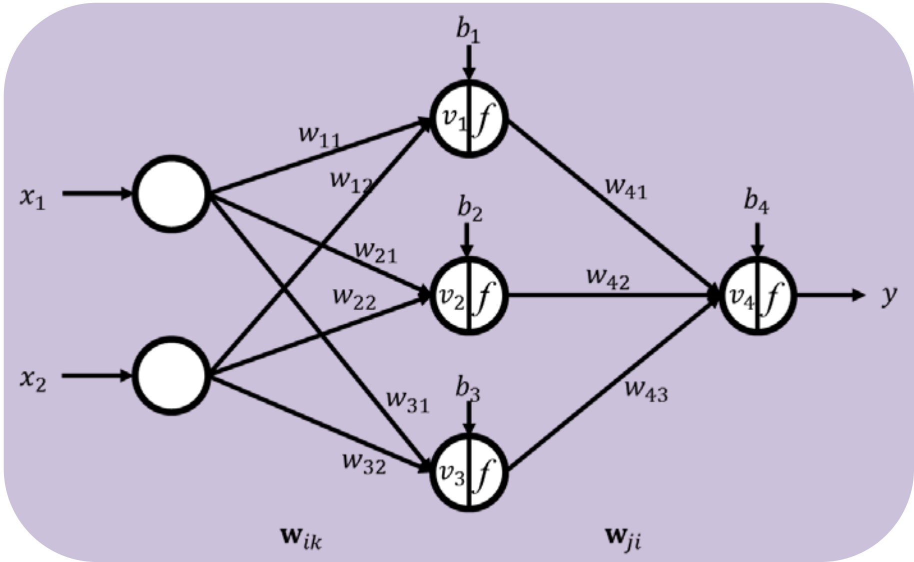
$$= b_j(n) + \eta \left(d_j(n) - y_j(n)\right)$$

$$E_j(n) = \frac{1}{2} e_j(n)^2$$

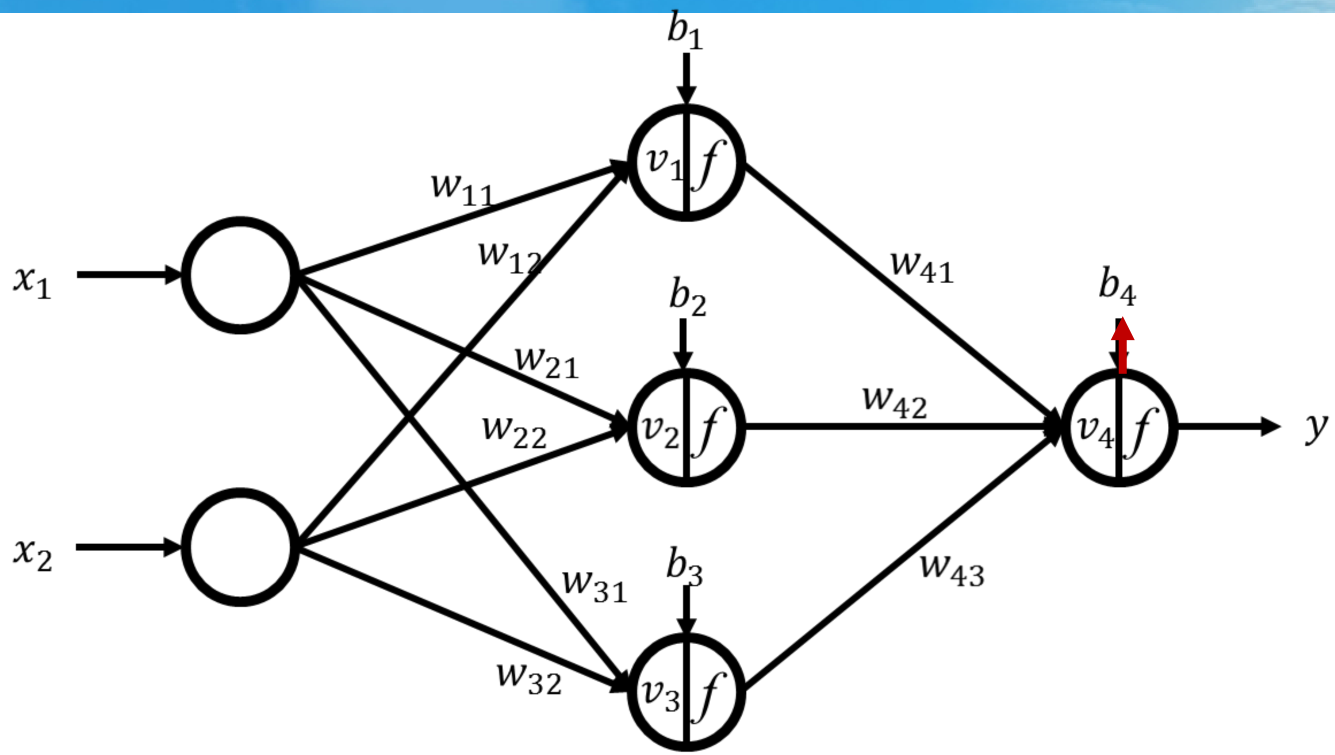
$$e_j(n) = d_j(n) - y_j(n)$$

$$y_j = f\left(v_j\right) = v_j$$

$$v_j = \sum_i w_{ji} f(v_i) + b_j$$



Backward Path



$$b_j(n+1) = b_j(n) + \Delta b_j(n) = b_j(n) - \eta \frac{\partial E_j(n)}{\partial b_j(n)} = b_j(n) + \eta\left(d_j(n) - y_j(n)\right)$$

$$b_4(n+1) = b_4(n) + \Delta b_4(n) = b_4(n) + \eta\big(d(n) - y(n)\big)$$

Backward Path

(3) Update rule for the weights of the hidden neurons:

$$w_{ik}(n+1) = w_{ik}(n) + \Delta w_{ik}(n)$$

$$= w_{ik}(n) - \eta \frac{\partial E_j(n)}{\partial w_{ik}(n)}$$

$$= w_{ik}(n) - \eta \frac{\partial E_j(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial f(v_i(n))} \frac{\partial f(v_i(n))}{\partial v_i(n)} \frac{\partial v_i(n)}{\partial w_{ik}(n)}$$

$$= w_{ik}(n) - \eta e_j(n)(-1) f'\left(v_j(n)\right) w_{ji}(n) f'(v_i(n)) x_k(n)$$

$$= w_{ik}(n) - \eta e_j(n)(-1)(1) w_{ji}(n) \left[f(v_i(n))\left(1 - f(v_i(n))\right)\right] x_k(n)$$

$$= w_{ik}(n) + \eta \left(d_j(n) - y_j(n)\right) w_{ji}(n) \left[f(v_i(n))\left(1 - f(v_i(n))\right)\right] x_k(n)$$

$$E_j(n) = \frac{1}{2} e_j(n)^2$$

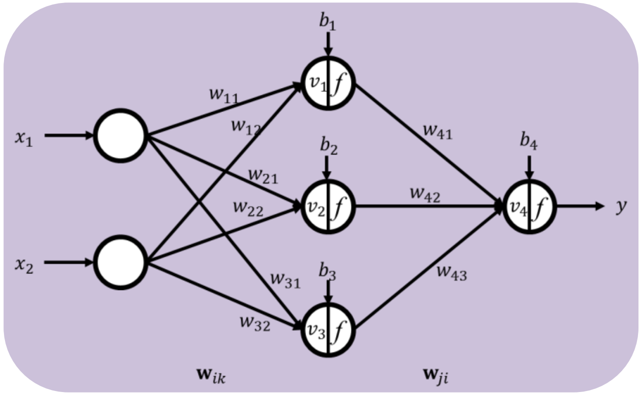
$$e_j(n) = d_j(n) - y_j(n)$$

$$y_j = f(v_j) = v_j$$

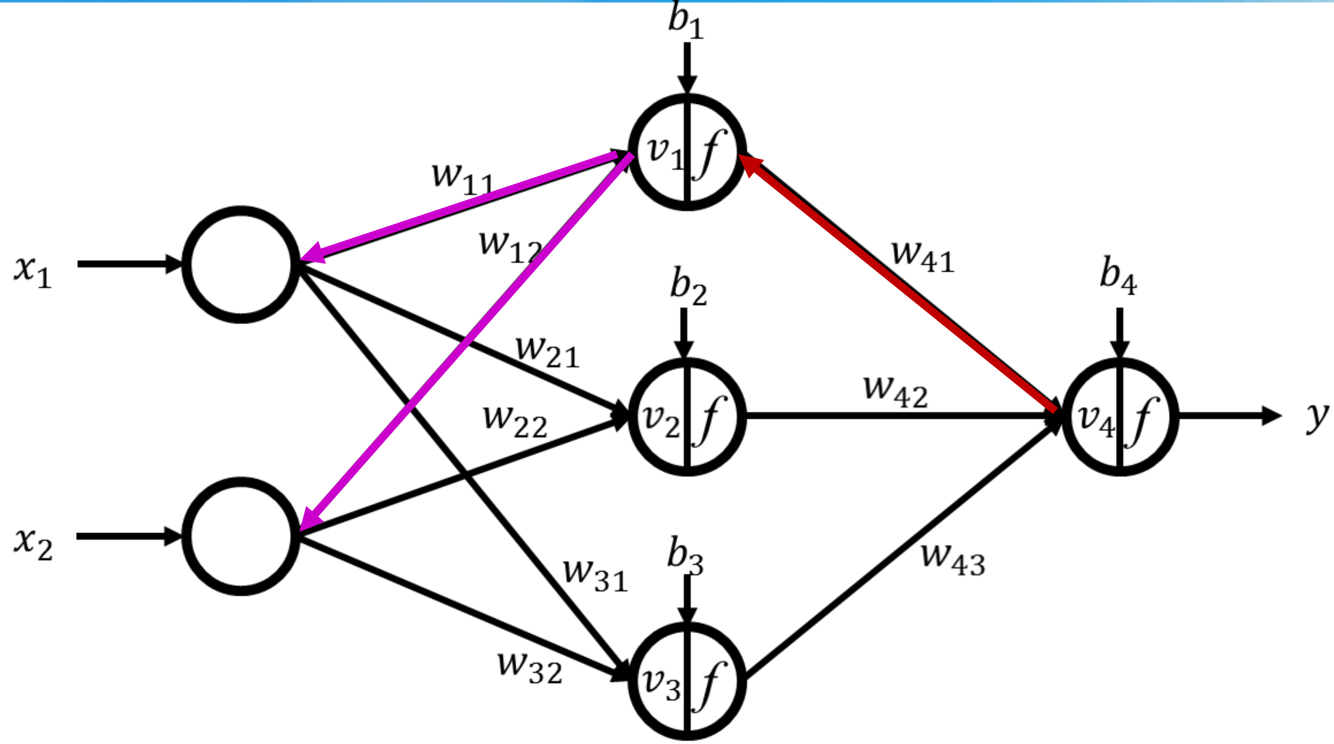
$$v_j = \sum_i w_{ji} f(v_i) + b_j$$

$$y_{hi} = f(v_i) = \frac{1}{1 + e^{-v_i}}$$

$$v_i = \sum_k w_{ik} x_k + b_i$$



Backward Path

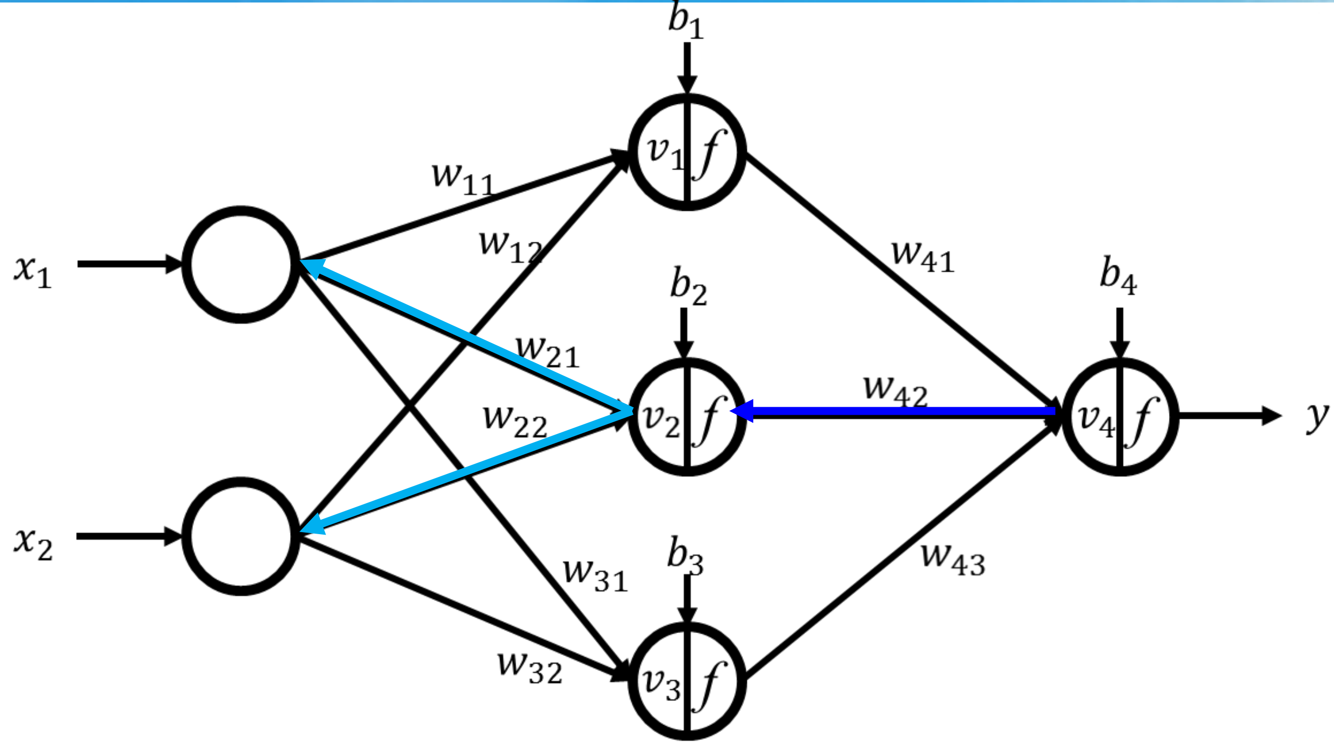


$$w_{ik}(n+1) = w_{ik}(n) + \Delta w_{ik}(n) = w_{ik}(n) + \eta\left(d_j(n) - y_j(n)\right) w_{ji}(n)\left[f\left(v_i(n)\right)\left(1 - f\left(v_i(n)\right)\right)\right] x_k(n)$$

$$w_{11}(n+1) = w_{11}(n) + \Delta w_{11}(n)$$
$$= w_{11}(n) + \eta\big(d(n) - y(n)\big) w_{41}(n) f\big(v_1(n)\big)\left(1 - f\big(v_1(n)\big)\right) x_1(n)$$

$$w_{12}(n+1) = w_{12}(n) + \Delta w_{12}(n)$$
$$= w_{12}(n) + \eta\big(d(n) - y(n)\big) w_{41}(n) f\big(v_1(n)\big)\left(1 - f\big(v_1(n)\big)\right) x_2(n)$$

Backward Path

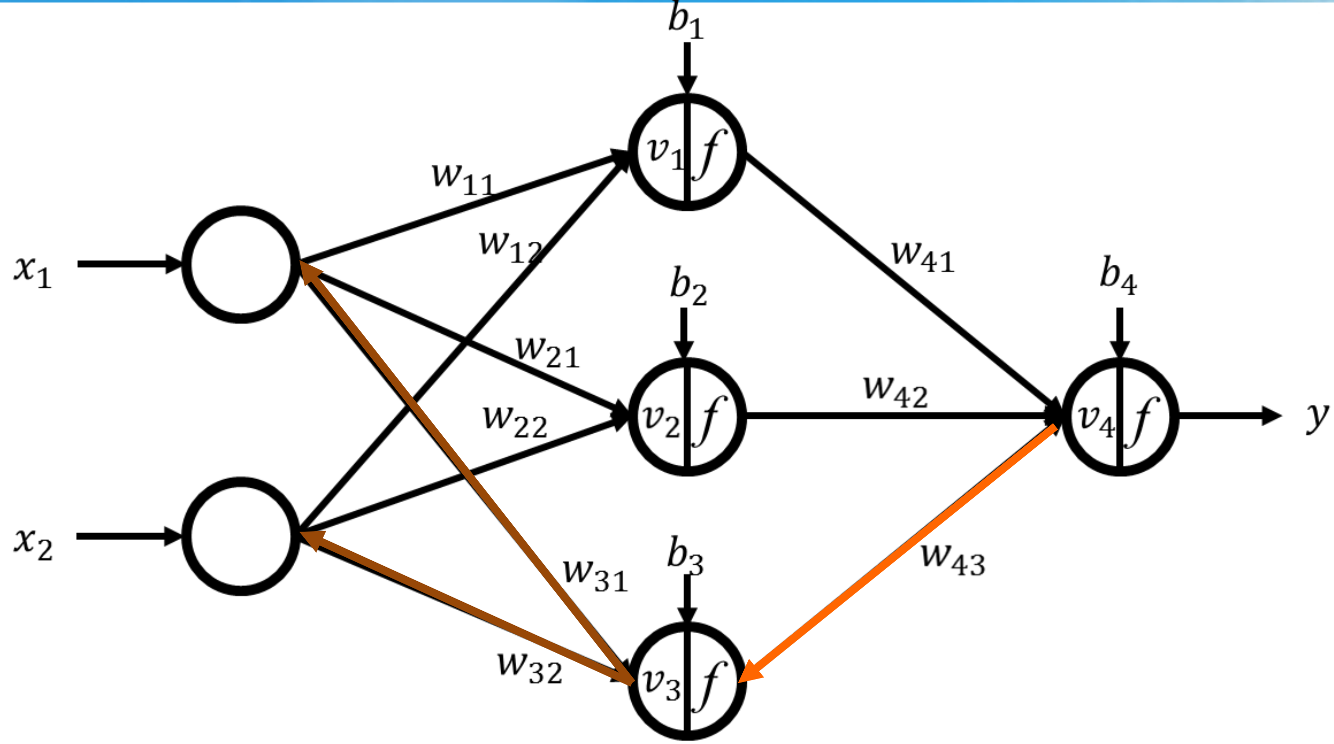


$$w_{ik}(n+1) = w_{ik}(n) + \Delta w_{ik}(n) = w_{ik}(n) + \eta\left(d_j(n) - y_j(n)\right) w_{ji}(n) \left[f\left(v_i(n)\right)\left(1 - f\left(v_i(n)\right)\right)\right] x_k(n)$$

$$w_{21}(n+1) = w_{21}(n) + \Delta w_{21}(n)$$
$$= w_{21}(n) + \eta\left(d(n) - y(n)\right) w_{42}(n) f\left(v_2(n)\right)\left(1 - f\left(v_2(n)\right)\right) x_1(n)$$

$$w_{22}(n+1) = w_{22}(n) + \Delta w_{22}(n)$$
$$= w_{22}(n) + \eta\left(d(n) - y(n)\right) w_{42}(n) f\left(v_2(n)\right)\left(1 - f\left(v_2(n)\right)\right) x_2(n)$$

Backward Path



$$w_{ik}(n+1) = w_{ik}(n) + \Delta w_{ik}(n) = w_{ik}(n) + \eta\left(d_j(n) - y_j(n)\right) w_{ji}(n)\left[f\left(v_i(n)\right)\left(1 - f\left(v_i(n)\right)\right)\right] x_k(n)$$

$$\begin{aligned} w_{31}(n+1) &= w_{31}(n) + \Delta w_{31}(n) \\ &= w_{31}(n) + \eta\big(d(n) - y(n)\big) w_{43}(n) f\big(v_3(n)\big)\left(1 - f\big(v_3(n)\big)\right) x_1(n) \end{aligned}$$

$$\begin{aligned} w_{32}(n+1) &= w_{32}(n) + \Delta w_{32}(n) \\ &= w_{32}(n) + \eta\big(d(n) - y(n)\big) w_{43}(n) f\big(v_3(n)\big)\left(1 - f\big(v_3(n)\big)\right) x_2(n) \end{aligned}$$

Backward Path

(4) Update rule for the biases of the hidden neurons:

$$b_i(n+1) = b_i(n) + \Delta b_i(n)$$

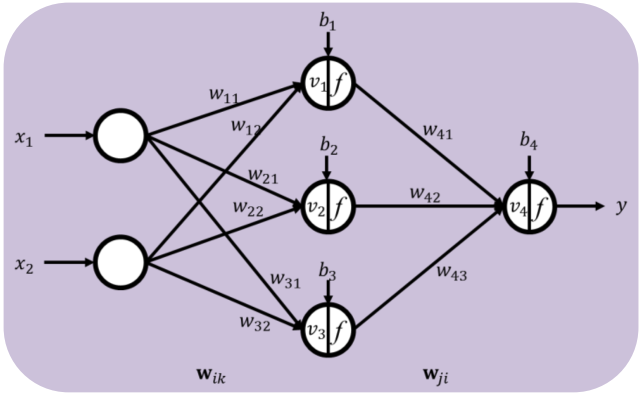
$$= b_i(n) - \eta \frac{\partial E_j(n)}{\partial b_i(n)}$$

$$= b_i(n) - \eta \frac{\partial E_j(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial f\big(v_i(n)\big)} \frac{\partial f\big(v_i(n)\big)}{\partial v_i(n)} \frac{\partial v_i(n)}{\partial b_i(n)}$$

$$= b_i(n) - \eta e_j(n)(-1) f'\Big(v_j(n)\Big) w_{ji}(n) f'\big(v_i(n)\big)(1)$$

$$= b_i(n) - \eta e_j(n)(-1)(1) w_{ji}(n) \Big[f\big(v_i(n)\big)\Big(1 - f\big(v_i(n)\big)\Big)\Big](1)$$

$$= b_i(n) + \eta \Big(d_j(n) - y_j(n)\Big) w_{ji}(n) \Big[f\big(v_i(n)\big)\Big(1 - f\big(v_i(n)\big)\Big)\Big]$$



$$E_j(n) = \frac{1}{2} e_j(n)^2$$

$$e_j(n) = d_j(n) - y_j(n)$$

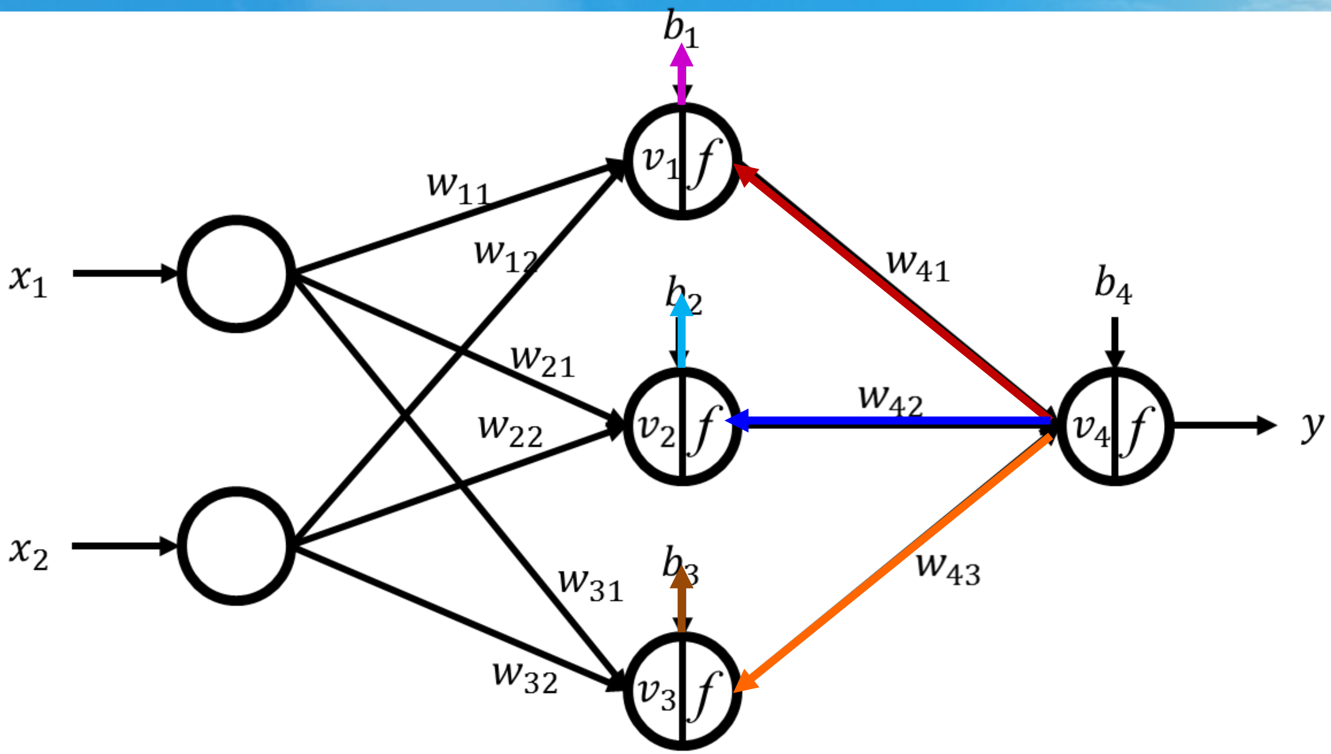
$$y_j = f(v_j) = v_j$$

$$v_j = \sum_i w_{ji} f(v_i) + b_j$$

$$y_{hi} = f(v_i) = \frac{1}{1 + e^{-v_i}}$$

$$v_i = \sum_k w_{ik} x_k + b_i$$

Backward Path



$$b_i(n+1) = b_i(n) + \Delta b_i(n) = b_i(n) + \eta\left(d_j(n) - y_j(n)\right) w_{ji}(n)\left[f\left(v_i(n)\right)\left(1 - f\left(v_i(n)\right)\right)\right]$$

$$b_1(n+1) = b_1(n) + \Delta b_1(n) = b_1(n) + \eta\big(d(n) - y(n)\big) w_{41}(n) f\big(v_1(n)\big)\left(1 - f\big(v_1(n)\big)\right)$$

$$b_2(n+1) = b_2(n) + \Delta b_2(n) = b_2(n) + \eta\big(d(n) - y(n)\big) w_{42}(n) f\big(v_2(n)\big)\left(1 - f\big(v_2(n)\big)\right)$$

$$b_3(n+1) = b_3(n) + \Delta b_3(n) = b_3(n) + \eta\big(d(n) - y(n)\big) w_{43}(n) f\big(v_3(n)\big)\left(1 - f\big(v_3(n)\big)\right)$$

HW 1

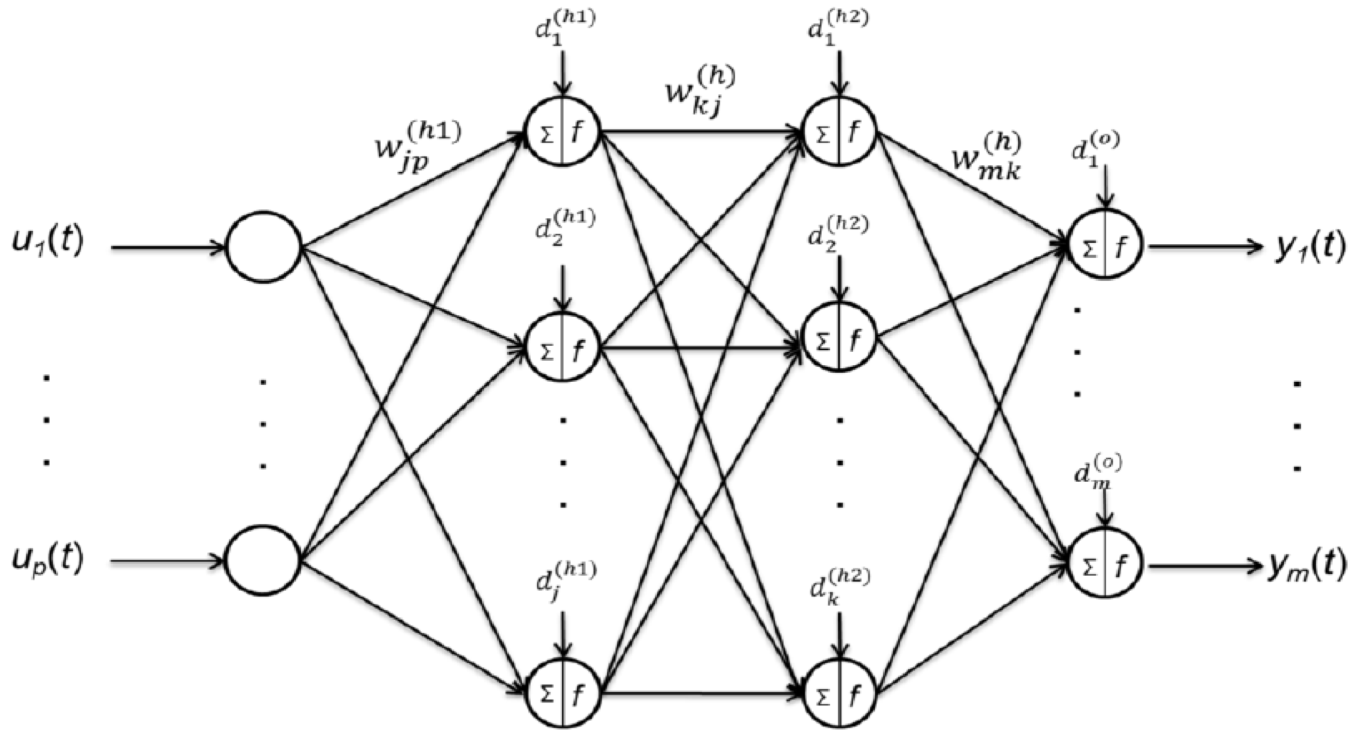


Fig. 1. Structure of three-layer feedforward neural network.

$f^{h1}(v_j) = \text{sigmoid}(v_j) \quad f^{h2}(v_k) = \tanh(v_k) \quad y_m = f^{out}(v_m) = v_m$

Backward Path

(1) Update rule for the weights of the output neurons:

Let: bias=b
design=d

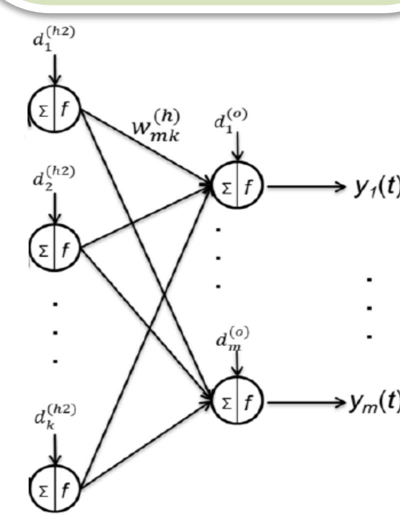
$$
\begin{aligned}
w_{mk}(t+1) &= w_{mk}(t) + \Delta w_{mk}(t) \\
&= w_{mk}(t) - \eta \frac{\partial E_m(t)}{\partial w_{mk}(t)} \\
&= w_{mk}(t) - \eta \frac{\partial E_m(t)}{\partial e_m(t)} \frac{\partial e_m(t)}{\partial y_m(t)} \frac{\partial y_m(t)}{\partial v_m(t)} \frac{\partial v_m(t)}{\partial w_{mk}(t)} \\
&= w_{mk}(t) - \eta e_m(t)(-1) f^{out'}\big(v_m(t)\big) f^{h2}\big(v_k(t)\big) \\
&= w_{mk}(t) - \eta e_m(t)(-1)(1) f^{h2}\big(v_k(t)\big) \\
&= w_{mk}(t) - \eta \big(d_m(t) - y_m(t)\big)(-1)(1) f^{h2}\big(v_k(t)\big) \\
&= w_{mk}(t) + \eta \big(d_m(t) - y_m(t)\big) \frac{e^{v_k} - e^{-v_k}}{e^{v_k} + e^{-v_k}}
\end{aligned}
$$

$$E_m(t) = \frac{1}{2}\left(\sum e_m(t)^2\right)$$

$$e_m(t) = d_m(t) - y_m(t)$$

$$y_m = \boldsymbol{f^{out}}(v_m) = v_m$$

$$v_m = \sum_m w_{mk} f^{h2}(v_k) + b_m$$



Backward Path

(2) Update rule for the biases of the output neurons:

$$b_m(t+1) = b_m(t) + \Delta b_m(t) \quad = b_m(t) - \eta \frac{\partial E_m(t)}{\partial b_m(t)}$$

$$= b_m(t) - \eta \frac{\partial E_m(t)}{\partial e_m(t)} \frac{\partial e_m(t)}{\partial y_m(t)} \frac{\partial y_m(t)}{\partial v_m(t)} \frac{\partial v_m(t)}{\partial b_m(t)}$$

$$= b_m(t) - \eta e_m(t)(-1) f^{out'}\big(v_m(t)\big)(1)$$

$$= b_m(t) - \eta e_m(t)(-1)(1)(1)$$

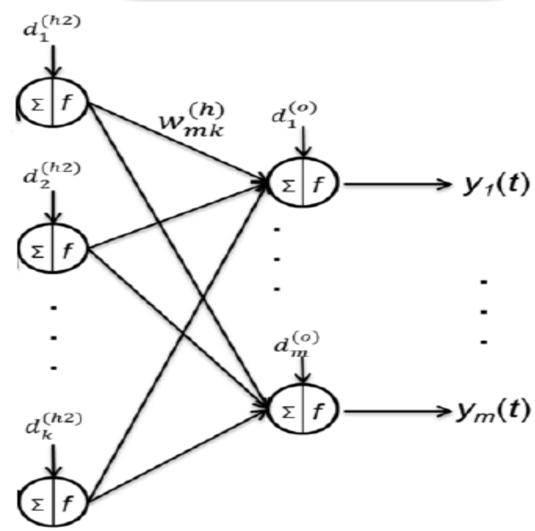
$$= b_m(t) + \eta\big(d_m(t) - y_m(t)\big)$$

$$E_m(t) = \frac{1}{2}\left(\sum e_m(t)^2\right)$$

$$e_m(t) = d_j(t) - y_j(t)$$

$$y_m = f^{out}(v_m) = v_m$$

$$v_m = \sum_m w_{mk} f^{h2}(v_k) + b_m$$



Backward Path

$$E_m(t) = \frac{1}{2}\left(\sum e_m(t)^2\right)$$

$$e_m(t) = d_m(t) - y_m(t)$$

$$y_m = f^{out}(v_m) = v_m$$

$$v_m = \sum_m w_{mk} f^{h2}(v_k) + b_m$$

$$f^{h2}(v_k) = \frac{e^{v_k} - e^{-v_k}}{e^{v_k} + e^{-v_k}}$$

$$v_k = \sum_k w_{kj} f^{h1}(v_j(t)) + b_k$$

$$f^{h1}(v_j) = \frac{1}{1 + e^{-v_j}}$$

(3) Update rule for the weights of the 2nd hidden neurons:

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t)$$

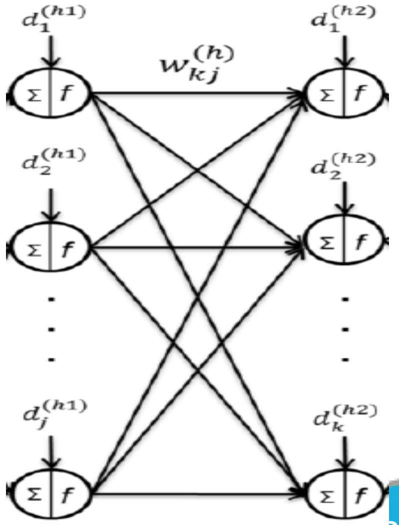
$$= w_{kj}(t) - \eta \frac{\partial E_m(t)}{\partial w_{kj}(t)}$$

$$= w_{kj}(t) - \eta \frac{\partial E_m(t)}{\partial e_m(t)} \frac{\partial e_m(t)}{\partial y_m(t)} \frac{\partial y_m(t)}{\partial v_m(t)} \frac{\partial v_m(t)}{\partial f^{h2}(v_k(t))} \frac{\partial f^{h2}(v_k(t))}{\partial v_k(t)} \frac{\partial v_k(t)}{\partial w_{kj}(t)}$$

$$= w_{kj}(t) - \eta \sum_m \{e_m(t)(-1) f^{out'}(v_m(t)) w_{mk}(t) f^{h2'}(v_k(t)) f^{h1}(v_j(t))\}$$

$$= w_{kj}(t) - \eta \sum_m \{e_m(t)(-1)(1) w_{mk}(t)(1 - (f^{h2}(v_k(t)))^2) f^{h1}(v_j(t))\}$$

$$= w_{kj}(t) + \eta \sum_m \{(d_m(t) - y_m(t)) w_{mk}(t)[1 - (tanh v_k(t))^2] \frac{1}{1 + e^{-v_j}}\}$$



Backward Path

(4) Update rule for the biases of the 2nd hidden neurons:

$$b_k(t+1) = b_k(t) + \Delta b_k t \quad = b_k(t) - \eta \frac{\partial E_m(t)}{\partial b_k(t)}$$

$$= b_k(t) - \eta \frac{\partial E_m(t)}{\partial e_m(t)} \frac{\partial e_m(t)}{\partial y_m(t)} \frac{\partial y_m(t)}{\partial v_m(t)} \frac{\partial v_m(t)}{\partial f^{h2}(v_k(t))} \frac{f^{h2}(v_k(t))}{\partial v_k(t)} \frac{\partial v_k(t)}{\partial b_k(t)}$$

$$= b_k(t) - \eta \sum_m \{e_m(t)(-1) f^{out'}(v_m(t)) w_{mk}(t) f^{h2'}(v_k(t)) 1\}$$

$$= b_k(t) - \eta \sum_m \{e_m(t)(-1) f^{out'}(v_m(t)) w_{mk}(t) \left[\left(1 - f^{h2}(v_k(t))^2\right)\right]\}$$

$$= b_k(t) - \eta \sum_m \{(d_m(t) - y_m(t))(-1) 1 w_{mk}(t) \left[\left(1 - f^{h2}(v_k(t))^2\right)\right]\}$$

$$= b_k(t) + \eta \sum_m \{(d_m(t) - y_m(t)) w_{mk}(t) [(1 - (tanh v_k(t))^2)]\}$$

$$E_m(t) = \frac{1}{2}(\sum e_m(t)^2)$$

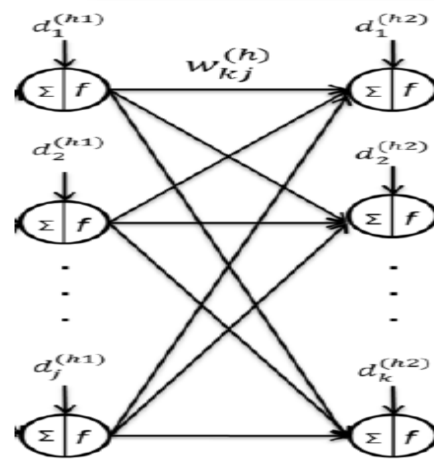
$$e_m(t) = d_m(t) - y_m(t)$$

$$y_m = f^{out}(v_m) = v_m$$

$$v_m = \sum_m w_{mk} f^{h2}(v_k) + b_m$$

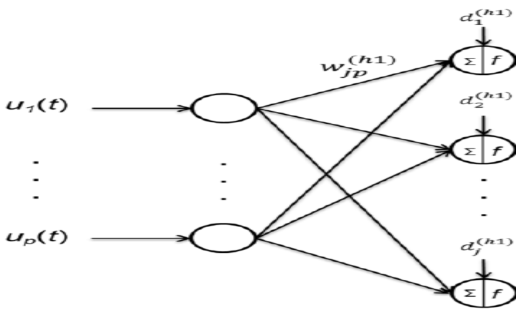
$$f^{h2}(v_k) = \frac{e^{v_k} - e^{-v_k}}{e^{v_k} + e^{-v_k}}$$

$$v_k = \sum_k w_{kj} f^{h1}(v_j) + b_k$$



Backward Path

(5) Update rule for the weights of the 1st hidden neurons:



$$E_m(t) = \frac{1}{2}\left(\sum e_m(t)^2\right)$$

$$f^{h2}(v_k) = \frac{e^{v_k} - e^{-v_k}}{e^{v_k} + e^{-v_k}}$$

$$e_m(t) = d_m(t) - y_m(t)$$

$$v_k = \sum_k w_{kj} f^{h1}(v_j) + b_k$$

$$y_m = f^{out}(v_m) = v_m$$

$$f^{h1}(v_j) = \frac{1}{1 + e^{-v_j}}$$

$$v_m = \sum_m w_{mk} f^{h2}(v_k) + b_m$$

$$v_j = \sum_j w_{jp} x_p + b_j$$

$$w_{jp}(t+1) = w_{jp}(t) + \Delta w_{jp}(t)$$

$$= w_{jp}(t) - \eta \frac{\partial E_m(t)}{\partial w_{jp}(t)}$$

$$= w_{jp}(t) - \eta \frac{\partial E_m(t)}{\partial e_m(t)} \frac{\partial e_m(t)}{\partial y_m(t)} \frac{\partial y_m(t)}{\partial v_m(t)} \frac{\partial v_m(t)}{\partial f^{h2}(v_k(t))} \frac{\partial f^{h2}(v_k(t))}{\partial v_k(t)} \frac{\partial v_k(t)}{\partial f^{h1}(v_j(t))} \frac{\partial f^{h1}(v_j(t))}{\partial v_j(t)} \frac{\partial v_j(t)}{\partial w_{jp}(t)}$$

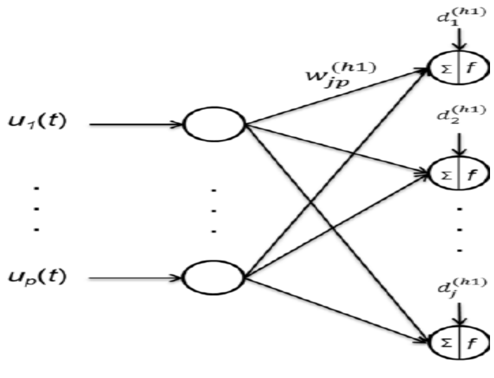
$$= w_{jp}(t) - \eta \sum_k \sum_m \{e_m(t)(-1) f^{out\prime}(v_m(t)) w_{mk}(t) f^{h2\prime}(v_k(t)) w_{kj}(t)\, f^{h1\prime}(v_j(t))\, u_p(t)\}$$

$$= w_{jp}(t) - \eta \sum_k \sum_m \{e_m(t)(-1)(1) w_{mk}(t) \left[\left(1 - f^{h2}(v_k(t))^2\right)\right] w_{kj}(t) \left[f^{h1}(v_j(t))\left(1 - f^{h1}(v_j(t))\right)\right] u_p(t)\}$$

$$= w_{jp}(t) + \eta \sum_k \sum_m \{(d_m(t) - y_m(t)) w_{mk}(t) \left[\left(1 - f^{h2}(v_k(t))^2\right)\right] w_{kj}(t) \left[f^{h1}(v_j(t))\left(1 - f^{h1}(v_j(t))\right)\right] u_p(t)\}$$

Backward Path

(6) Update rule for the biases of the 1st hidden neurons:



$$E_m(t) = \frac{1}{2}\left(\sum e_m(t)^2\right)$$

$$e_m(t) = d_m(t) - y_m(t)$$

$$y_m = f^{out}(v_m) = v_m$$

$$v_m = \sum_m w_{mk} f^{h2}(v_k) + b_m$$

$$f^{h2}(v_k) = \frac{e^{v_k} - e^{-v_k}}{e^{v_k} + e^{-v_k}}$$

$$v_k = \sum_k w_{kj} f^{h1}(v_j) + b_k$$

$$f^{h1}(v_j) = \frac{1}{1 + e^{-v_j}}$$

$$v_j = \sum_j w_{jp} u_p + b_j$$

$$b_j(t+1) = b_j(t) + \Delta b_j(t) \quad = b_j(t) - \eta \frac{\partial E_m(t)}{\partial b_j(t)}$$

$$= b_j(t) - \eta \frac{\partial E_m(t)}{\partial e_m(t)} \frac{\partial e_m(t)}{\partial y_m(t)} \frac{\partial y_m(t)}{\partial v_m(t)} \frac{\partial v_m(t)}{\partial f^{h2}(v_k(t))} \frac{\partial f^{h2}(v_k(t))}{\partial v_k(t)} \frac{\partial v_k(t)}{\partial f^{h1}(v_j(t))} \frac{\partial f^{h1}(v_j(t))}{\partial v_j(t)} \frac{\partial v_j(t)}{\partial b_j(t)}$$

$$= b_j(t) - \eta \sum_k \sum_m \{e_m(t)(-1) f^{out\prime}(v_m(t)) w_{mk}(t) f^{h2\prime}(v_k(t)) w_{kj}(t)\, f^{h1\prime}(v_j(t))\,(1)\}$$

$$= b_j(t) - \eta \sum_k \sum_m \{e_m(t)(-1)(1) w_{mk}(t) \left[\left(1 - f^{h2}(v_k(t))^2\right)\right] w_{kj}(t) \left[f^{h1}(v_j(t))\left(1 - f^{h1}(v_j(t))\right)\right](1)\}$$

$$= b_j(t) + \eta \sum_k \sum_m \{(d_m(t) - y_m(t)) w_{mk}(t) \left[\left(1 - f^{h2}(v_k(t))^2\right)\right] w_{kj}(t) \left[f^{h1}(v_j(t))\left(1 - f^{h1}(v_j(t))\right)\right](1)$$

Thanks for your attention !!

Any Question?