Weight updating in backprop

- Learning in backprop is similar to learning with perceptrons, i.e.,
  - Example inputs are fed to the network.
    - If the network computes an output vector that matches the target, nothing is done.
    - If there is a difference between output and target (i.e., an error), then the weights are adjusted to reduce this error.
    - The key is to assess the blame for the error and divide it among the contributing weights.
- The error term (T - o) is known for the units in the output layer. To adjust the weights between the hidden and the output layer, the gradient descent rule can be applied as done for perceptrons.
- To adjust weights between the input and hidden layer some way of estimating the errors made by the hidden units in needed.

Estimating Error

- Main idea: each hidden node contributes for some fraction of the error in each of the output nodes.
  - This fraction equals the strength of the connection (weight) between the hidden node and the output node.

$$\text{error at hidden node j} = \sum_{i \in outputs} w_{ij} \delta_i$$

where δ_i is the error at output node i.

Back-propagation algorithm for updating weights in a multilayer network

1. Initialize the weights in the network (often randomly)
2. **repeat**

    for each example e in the training set **do**

    i. O = neural-net-output(network, e) ; forward pass

    ii. T = teacher output for e

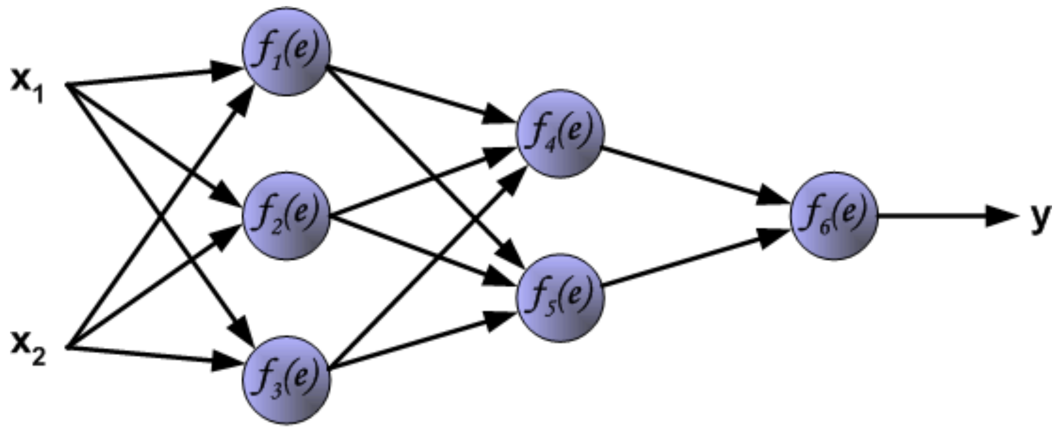
    iii. Calculate error (T - O) at the output units

    iv. Compute **wj = wj +α * Err * Ij** for all weights from hidden layer to output layer;backward pass

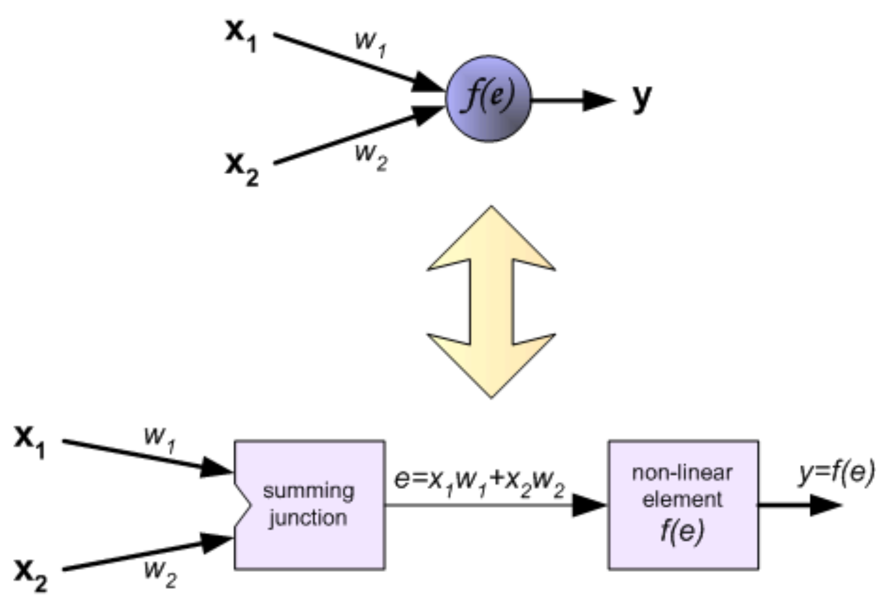
    v. Compute **wj = wj +α * Err * Ij** for all weights from input layer to hidden layer; backward pass continued

    vi. Update the weights in the network

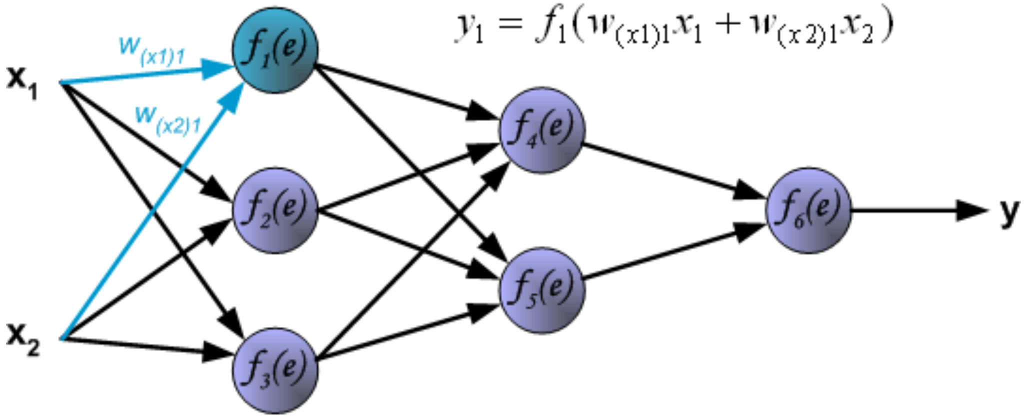
    end
3. **until** all examples classified correctly or stopping criterion met
4. **return**(network)



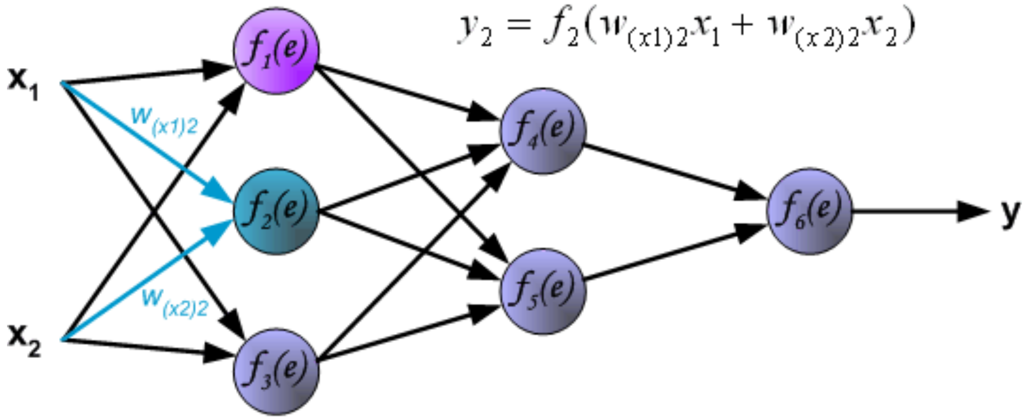
x1
x2
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y


x1
w1
f(e)
y
x2
w2
x1
w1
summing
junction
e=x1w1+x2w2
non-linear
element
f(e)
y=f(e)
x2
w2

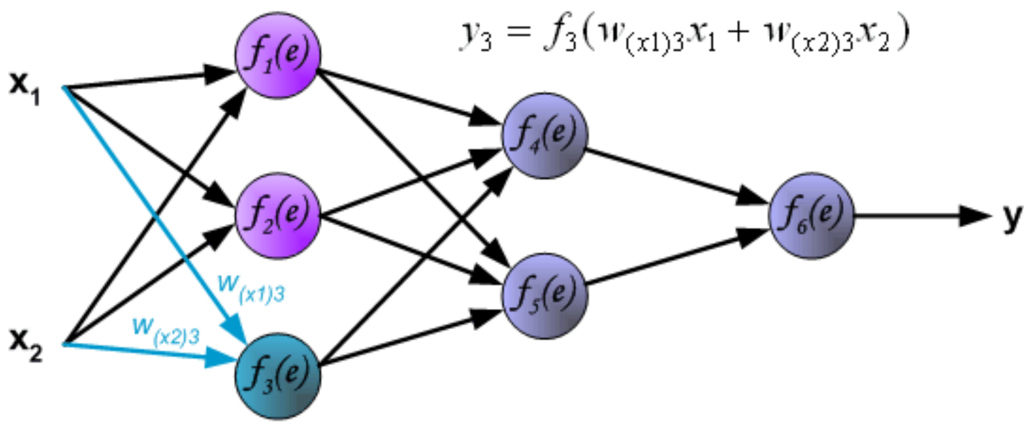
$$y_1 = f_1(w_{(x1)1}x_1 + w_{(x2)1}x_2)$$



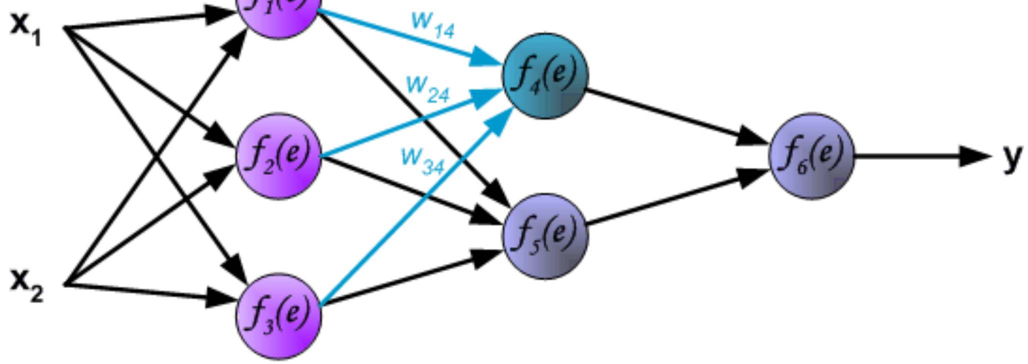
$$y_2 = f_2(w_{(x1)2}x_1 + w_{(x2)2}x_2)$$



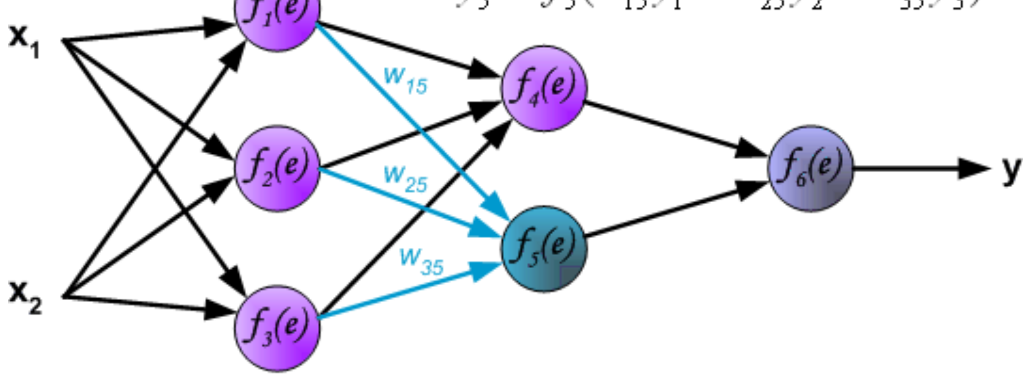
$$y_3 = f_3(w_{(x1)3}x_1 + w_{(x2)3}x_2)$$

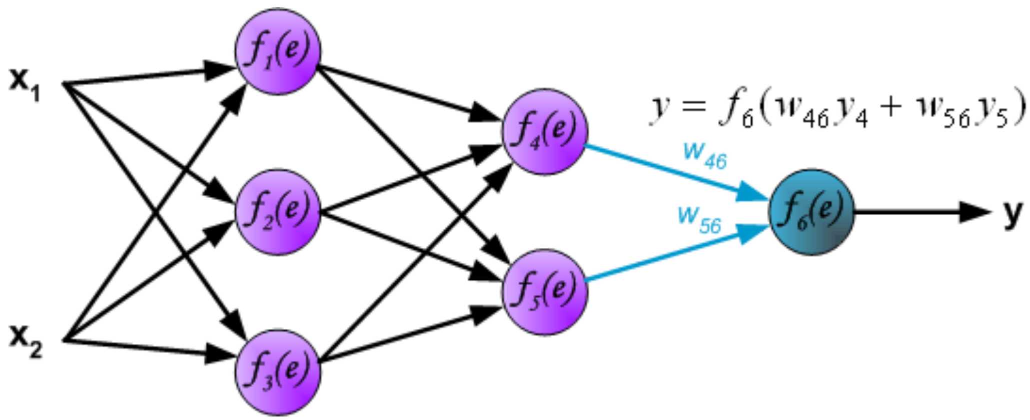


$$y_4 = f_4(w_{14}y_1 + w_{24}y_2 + w_{34}y_3)$$

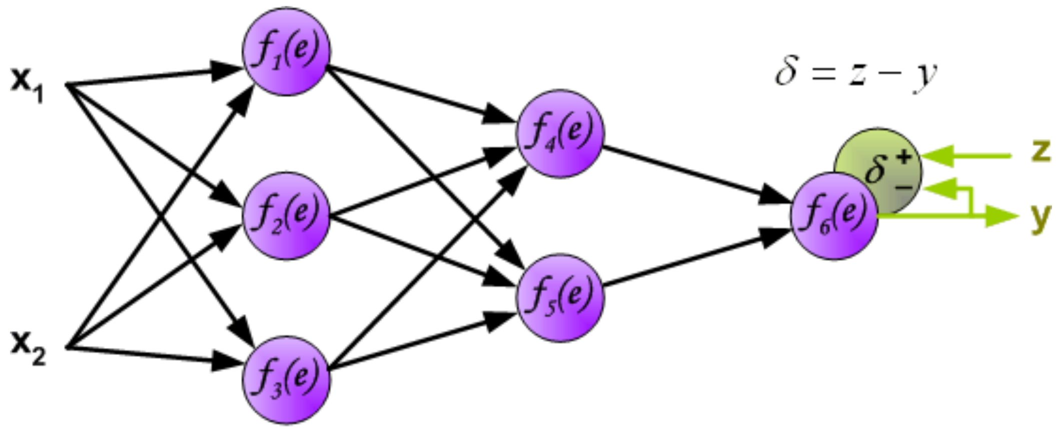


$$y_5 = f_5(w_{15}y_1 + w_{25}y_2 + w_{35}y_3)$$



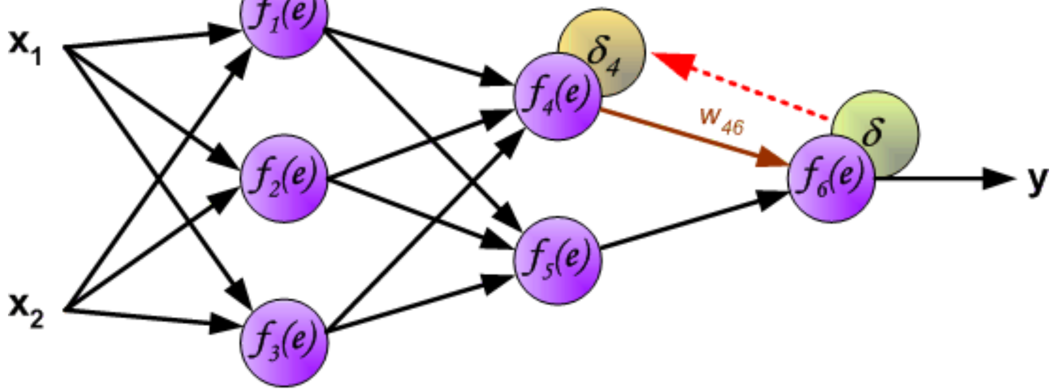


x_1
x_2
$f_1(e)$
$f_2(e)$
$f_3(e)$
$f_4(e)$
$f_5(e)$
$f_6(e)$
$y = f_6(w_{46}y_4 + w_{56}y_5)$
w_{46}
w_{56}
y

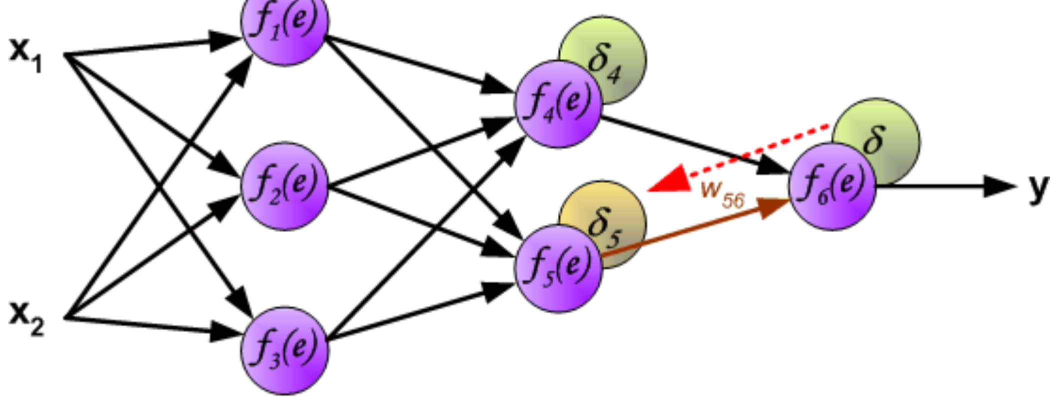


x_1
x_2
$f_1(e)$
$f_2(e)$
$f_3(e)$
$f_4(e)$
$f_5(e)$
$f_6(e)$
$\delta = z - y$
δ
+
–
z
y

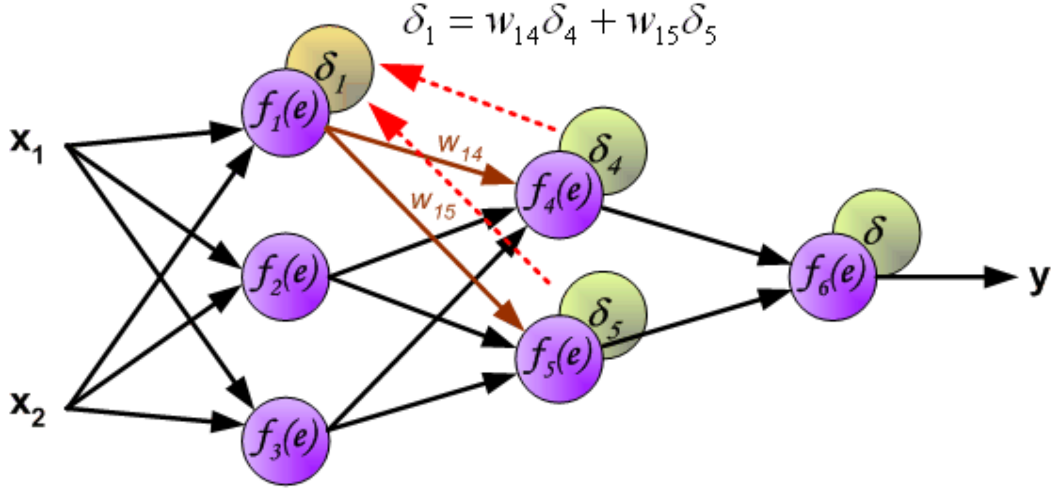
$$\delta_4 = w_{46}\delta$$



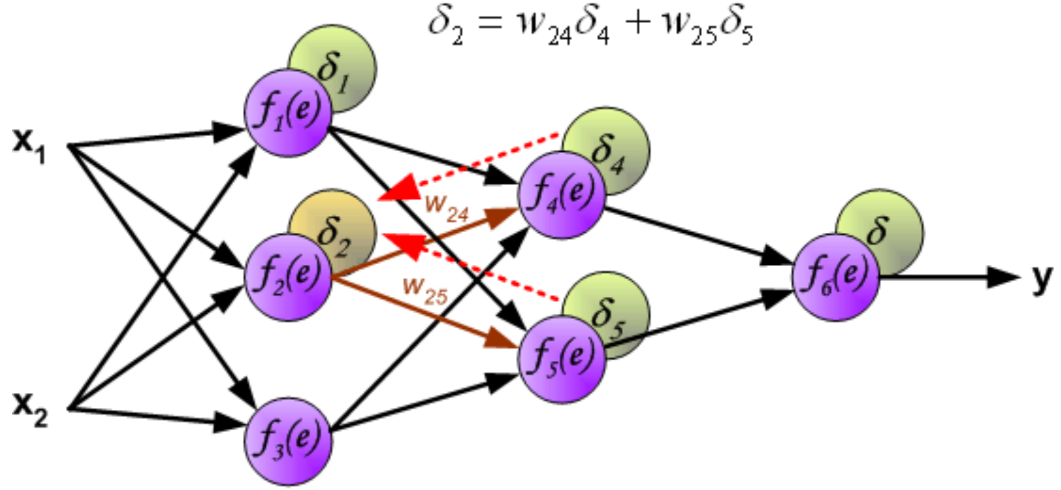
$$\delta_5 = w_{56}\delta$$



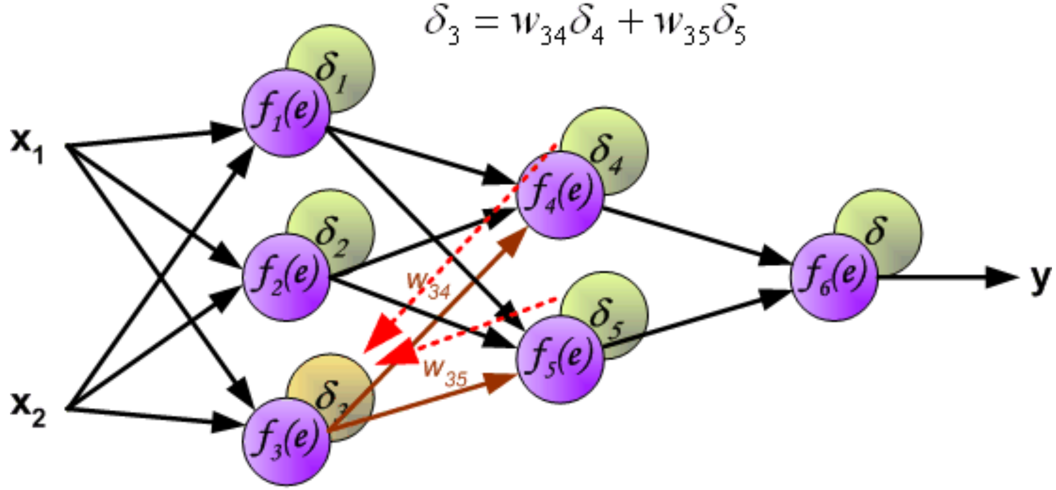
$$\delta_1 = w_{14}\delta_4 + w_{15}\delta_5$$



$$\delta_2 = w_{24}\delta_4 + w_{25}\delta_5$$

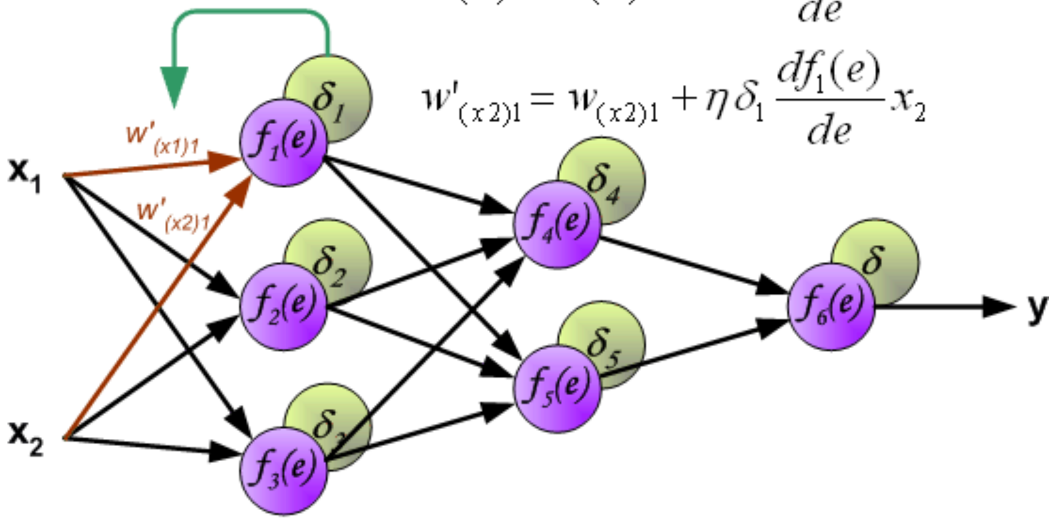


$$\delta_3 = w_{34}\delta_4 + w_{35}\delta_5$$



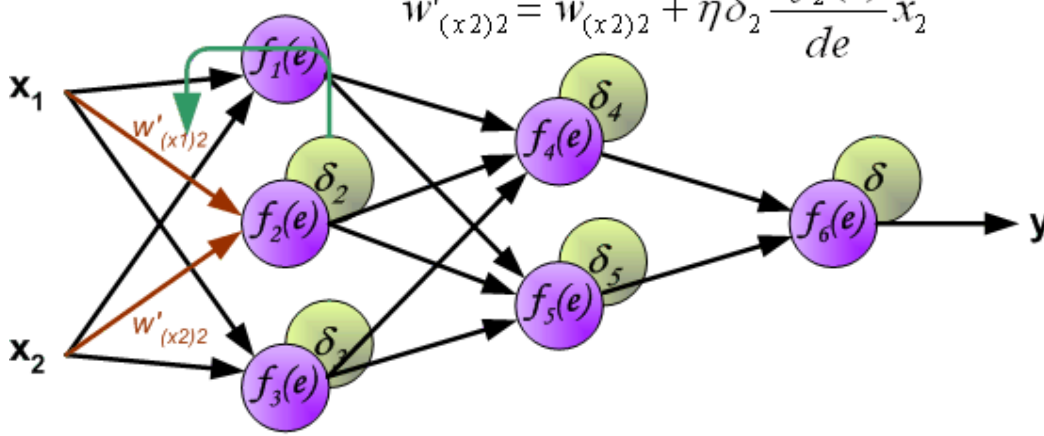
$$w'_{(x1)1} = w_{(x1)1} + \eta \delta_1 \frac{df_1(e)}{de} x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta \delta_1 \frac{df_1(e)}{de} x_2$$



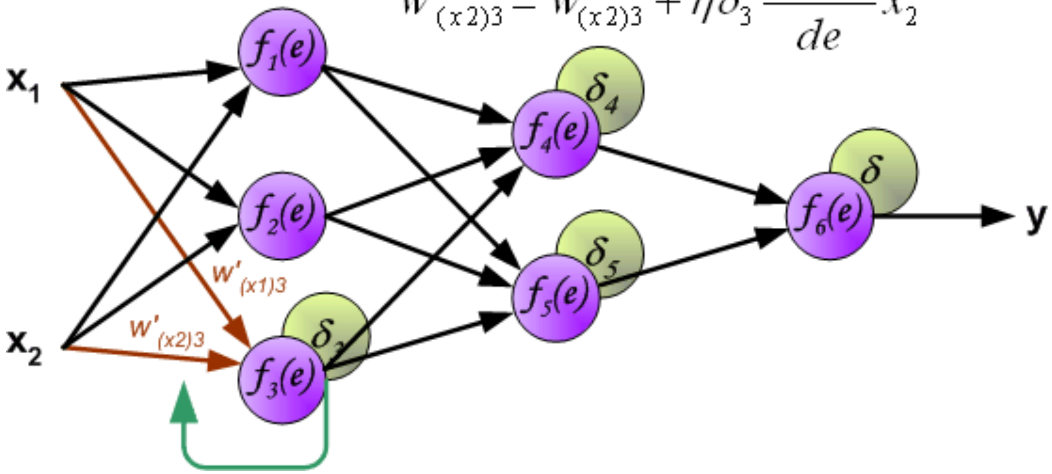
$$w'_{(x1)2} = w_{(x1)2} + \eta \delta_2 \frac{df_2(e)}{de} x_1$$

$$w'_{(x2)2} = w_{(x2)2} + \eta \delta_2 \frac{df_2(e)}{de} x_2$$



$$w'_{(x1)3} = w_{(x1)3} + \eta \delta_3 \frac{df_3(e)}{de} x_1$$

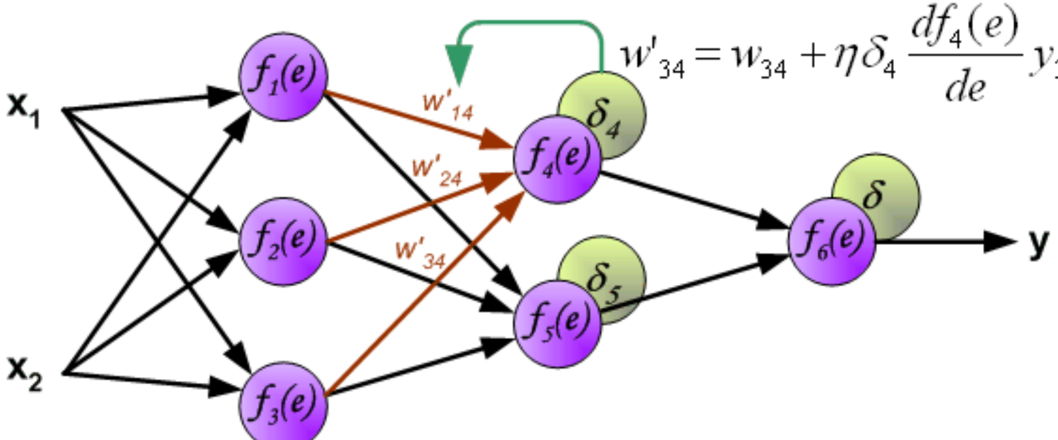
$$w'_{(x2)3} = w_{(x2)3} + \eta \delta_3 \frac{df_3(e)}{de} x_2$$



$$w'_{14} = w_{14} + \eta \delta_4 \frac{df_4(e)}{de} y_1$$

$$w'_{24} = w_{24} + \eta \delta_4 \frac{df_4(e)}{de} y_2$$

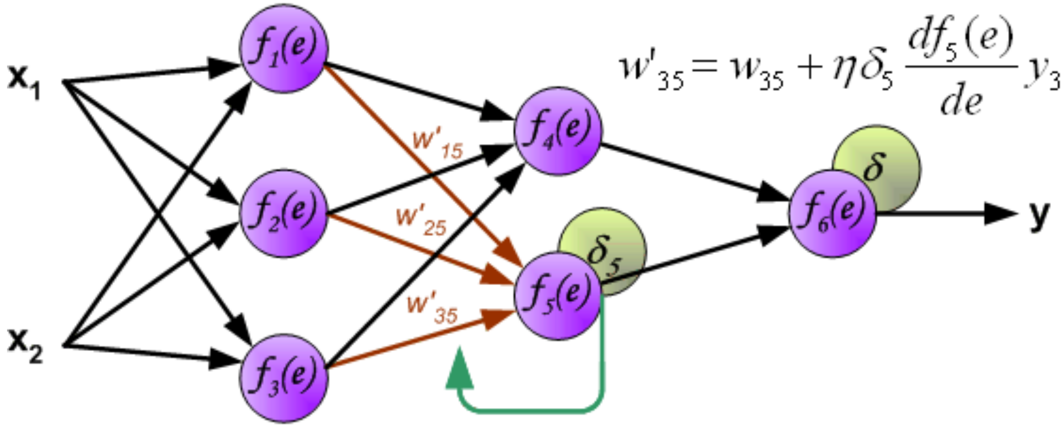
$$w'_{34} = w_{34} + \eta \delta_4 \frac{df_4(e)}{de} y_3$$



$$w'_{15} = w_{15} + \eta \delta_5 \frac{df_5(e)}{de} y_1$$

$$w'_{25} = w_{25} + \eta \delta_5 \frac{df_5(e)}{de} y_2$$

$$w'_{35} = w_{35} + \eta \delta_5 \frac{df_5(e)}{de} y_3$$



$$w'_{46} = w_{46} + \eta \delta \frac{df_6(e)}{de} y_4$$

$$w'_{56} = w_{56} + \eta \delta \frac{df_6(e)}{de} y_5$$