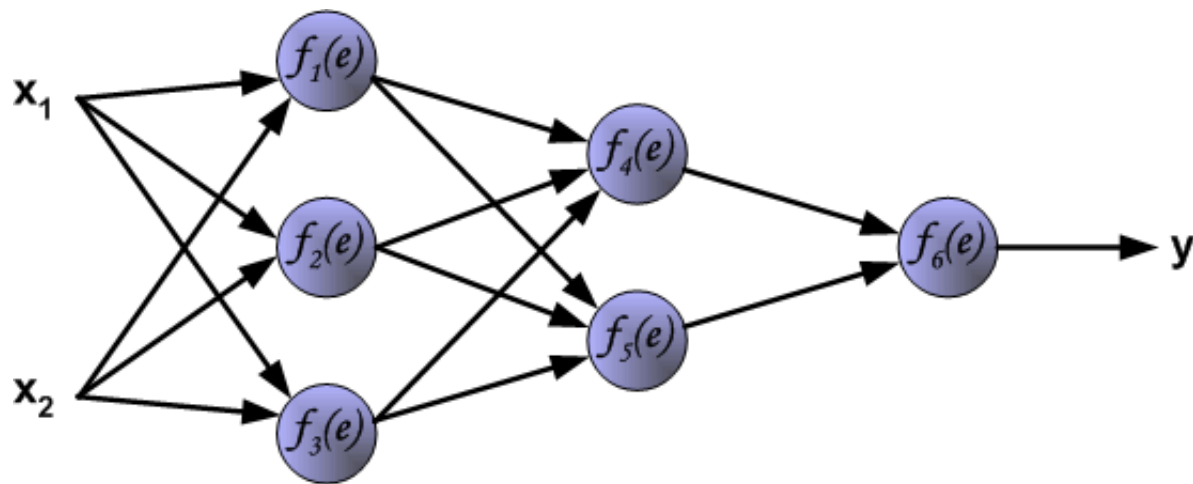# Principles of training multi-layer neural network using backpropagation
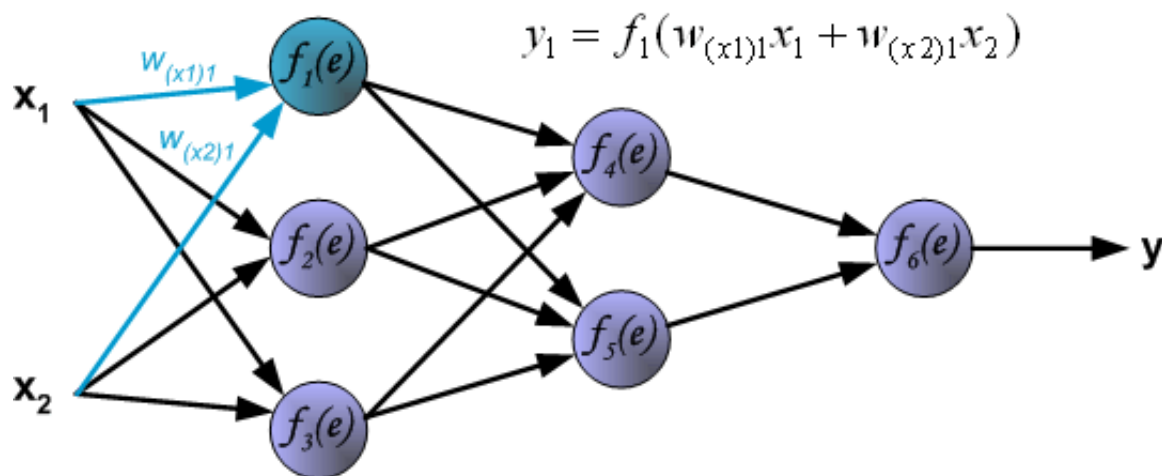

The project describes teaching process of multi-layer neural network employing *backpropagation* algorithm. To illustrate this process the three layer neural network with two inputs and one output, which is shown in the picture below, is used:
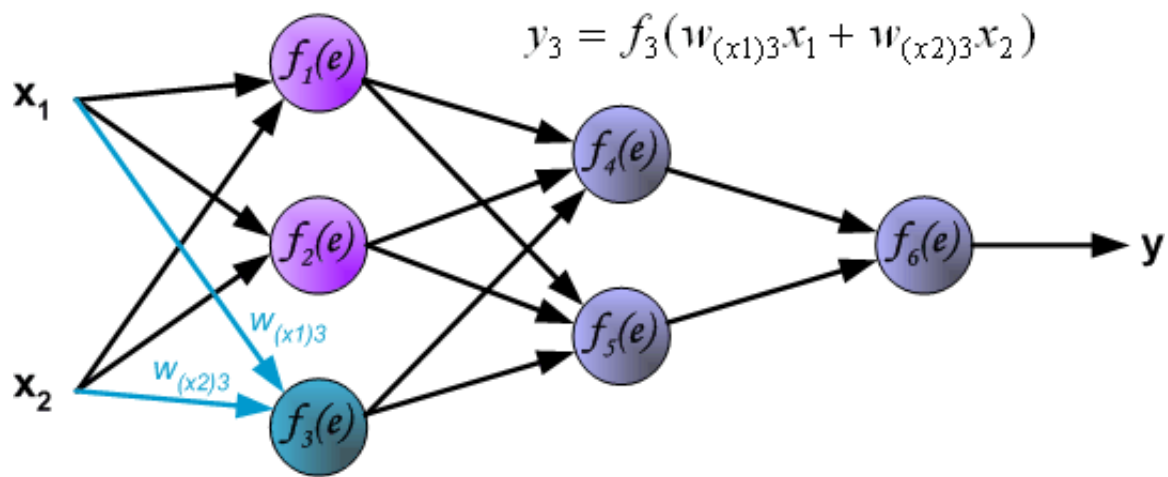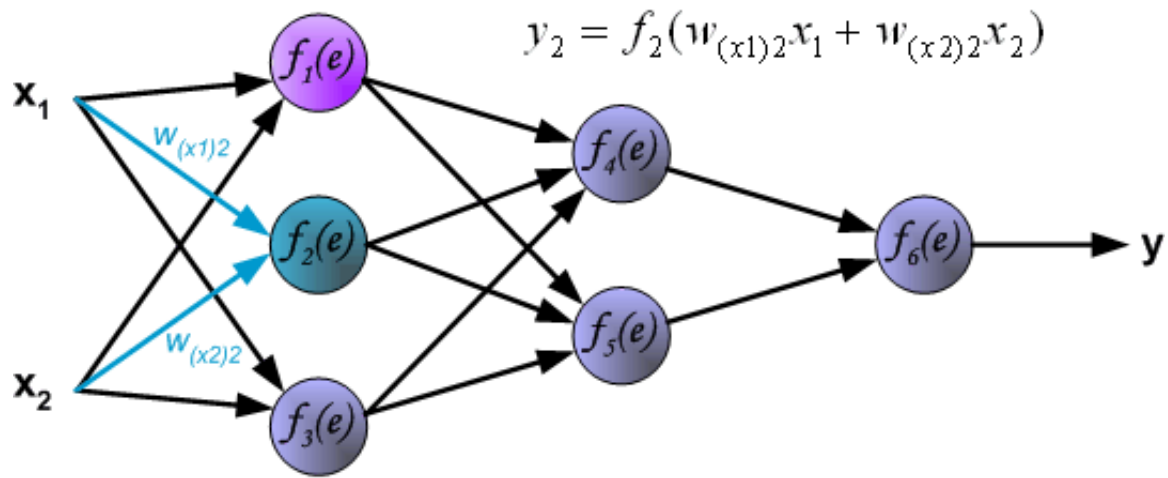

Each neuron is composed of two units. First unit adds products of weights coefficients and input signals. The second unit realise nonlinear function, called neuron activation function. Signal e is adder output signal, and $y = f(e)$ is output signal of nonlinear element. Signal y is also output signal of neuron.
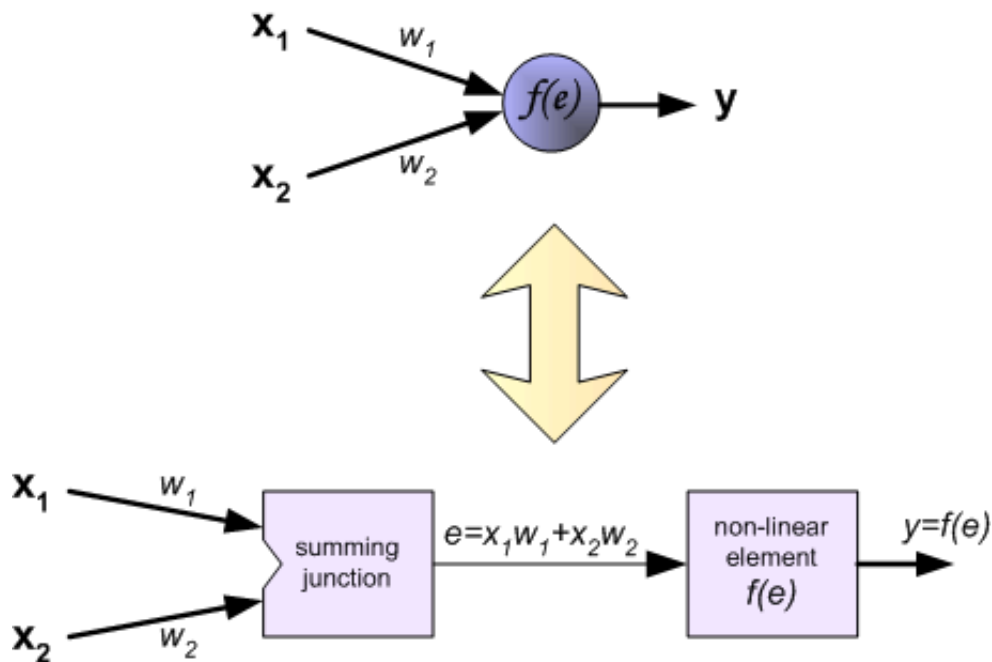
To teach the neural network we need training data set. The training data set consists of input signals (x_1 and x_2) assigned with corresponding target (desired output) z. The network training is an iterative process. In each iteration weights coefficients of nodes are modified using new data from training data set. Modification is calculated using algorithm described below: Each teaching step starts with forcing both input signals from training set. After this stage we can determine output signals values for each neuron in each network layer. Pictures below illustrate how signal is propagating through the network, Symbols $w_{(xm)n}$ represent weights of connections between network input x_m and neuron n in input layer. Symbols y_n represents output signal of neuron n.

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

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

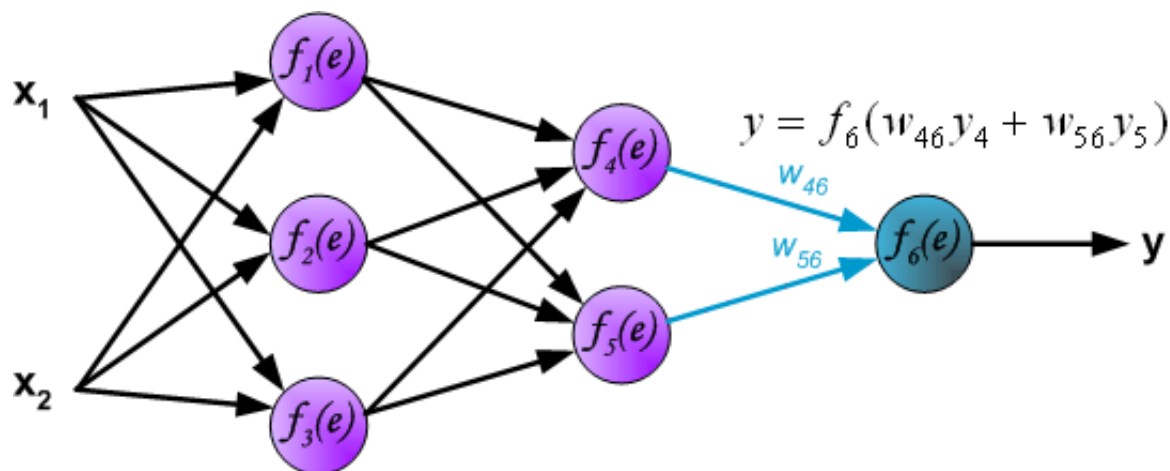$$y_3 = f_3(w_{(x1)3}x_1 + w_{(x2)3}x_2)$$

Propagation of signals through the hidden layer. Symbols w_{mn} represent weights of connections between output of neuron m and input of neuron n in the next layer.

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

$$y_5 = f_5(w_{15}\,y_1 + w_{25}\,y_2 + w_{35}\,y_3)$$
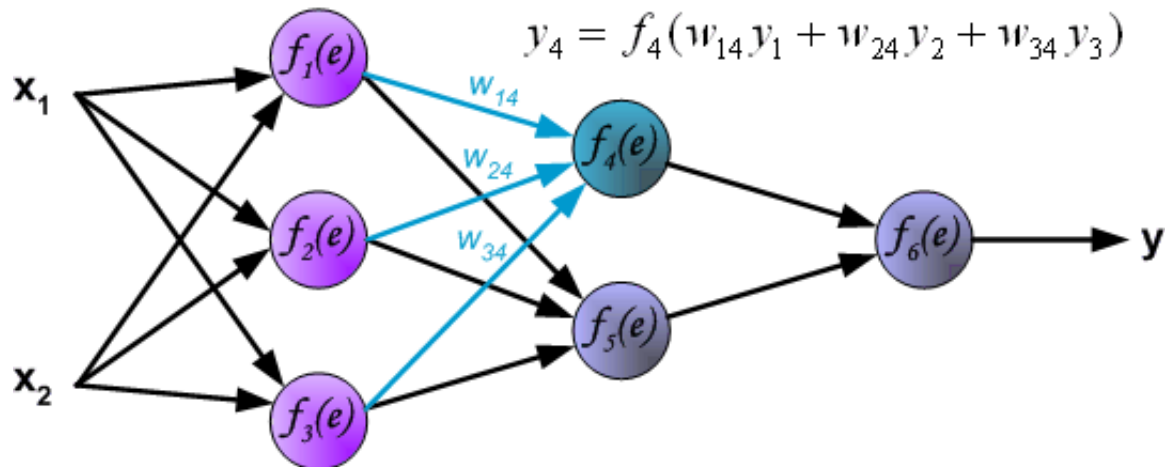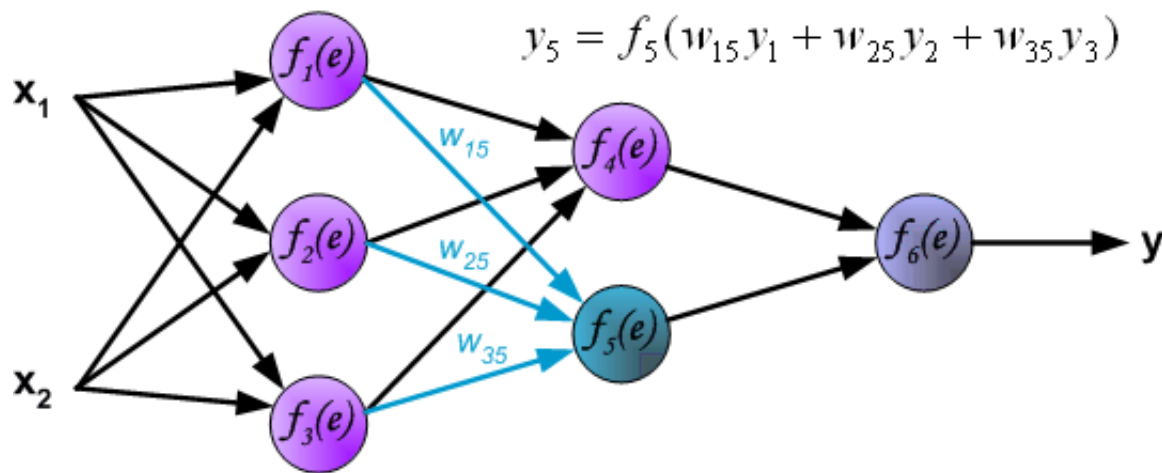
Propagation of signals through the output layer.

$$y = f_6(w_{46}\,y_4 + w_{56}\,y_5)$$

In the next algorithm step the output signal of the network y is compared with the desired output value (the target), which is found in training data set. The difference is called error signal δ of output layer neuron.

It is impossible to compute error signal for internal neurons directly, because output values of these neurons are unknown. For many years the effective method for training multiplayer networks has been unknown. Only in the middle eighties the backpropagation algorithm has been worked out. The idea is to propagate error signal δ (computed in single teaching step) back to all neurons, which output signals were input for discussed neuron.

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


$$\delta_5 = w_{56}\delta$$
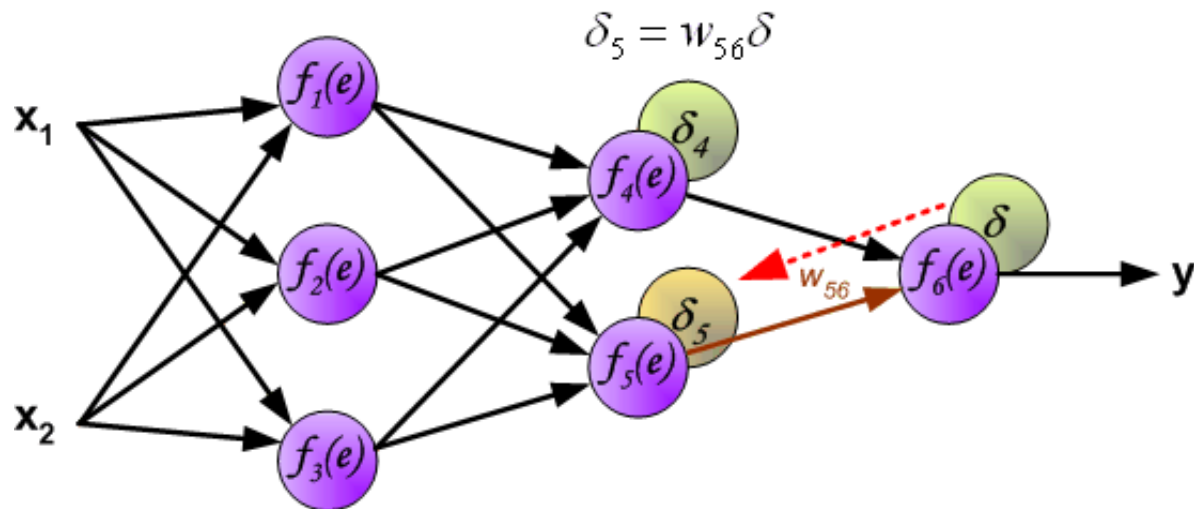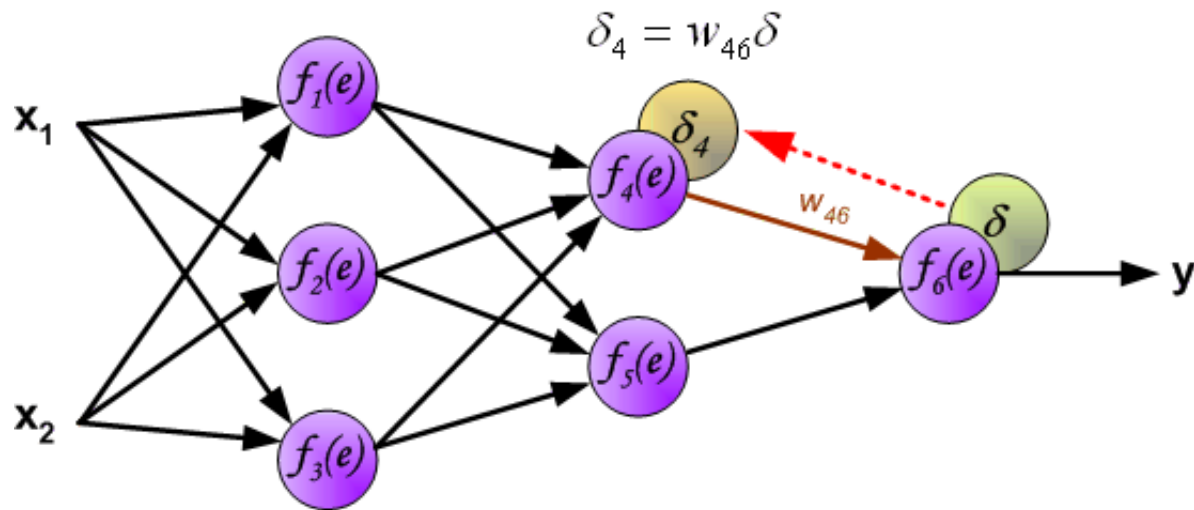
The weights' coefficients w_{mn} used to propagate errors back are equal to this used during computing output value. Only the direction of data flow is changed (signals are propagated from output to inputs one after the other). This technique is used for all network layers. If propagated errors came from few neurons they are added. The illustration is below:
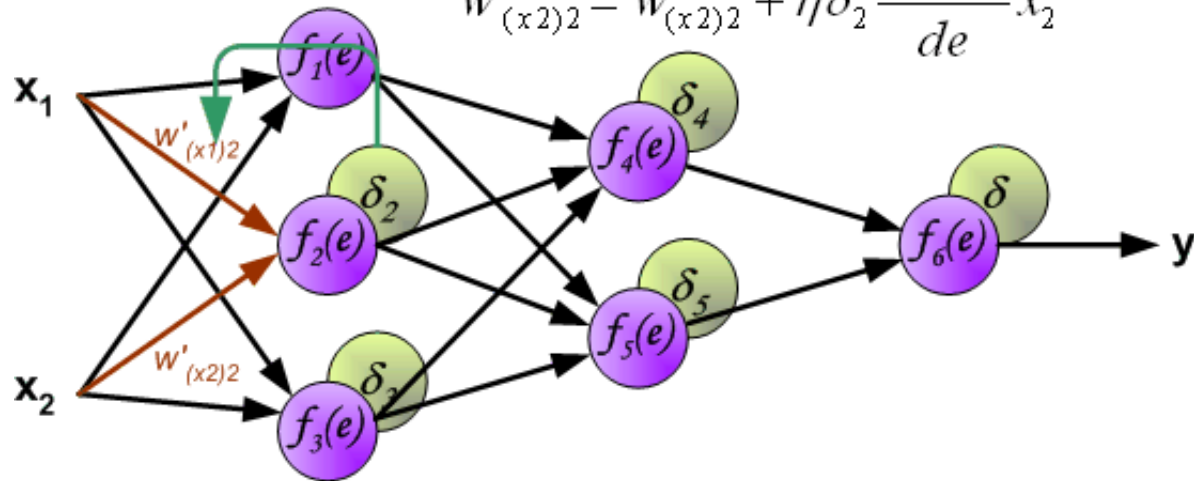
$$\delta_1 = w_{14}\delta_4 + w_{15}\delta_5$$

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

$$\delta_3 = w_{34}\delta_4 + w_{35}\delta_5$$
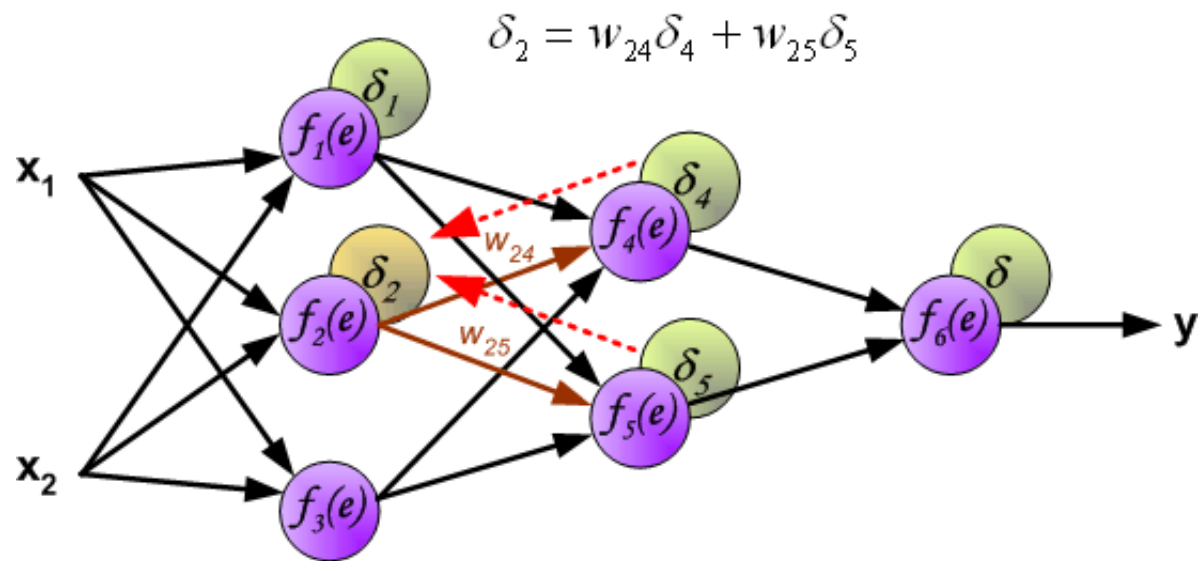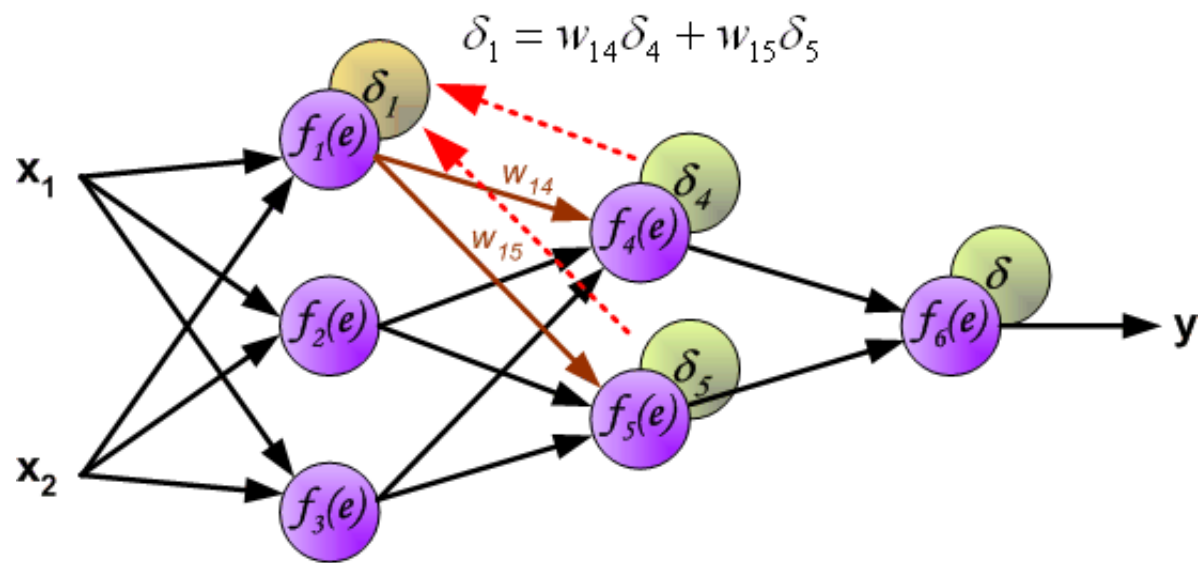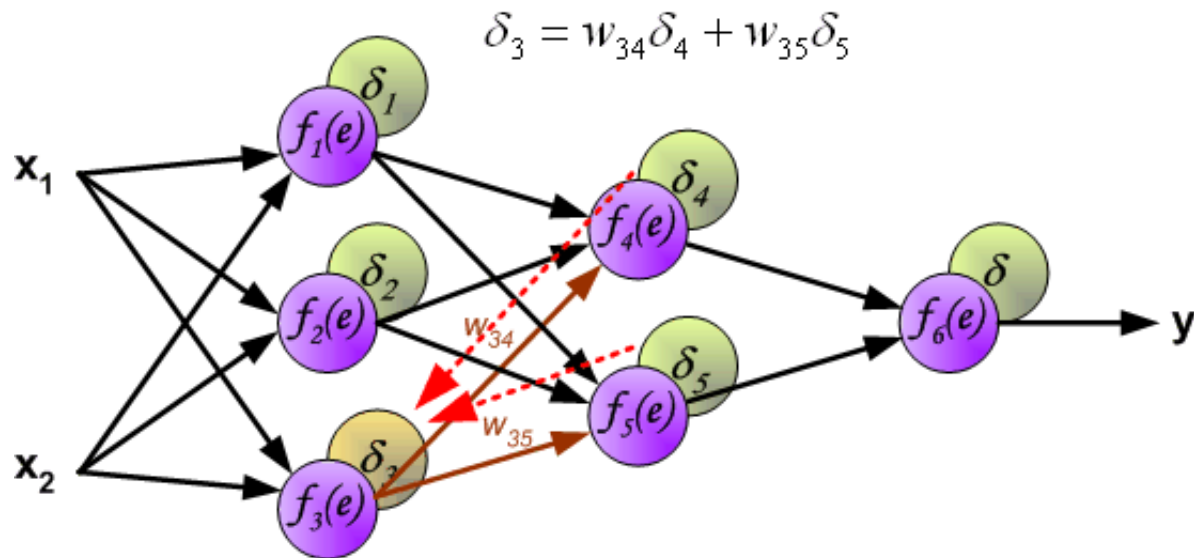
When the error signal for each neuron is computed, the weights coefficients of each neuron input node may be modified. In formulas below *df(e)/de* represents derivative of neuron activation function (which weights are modified).

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

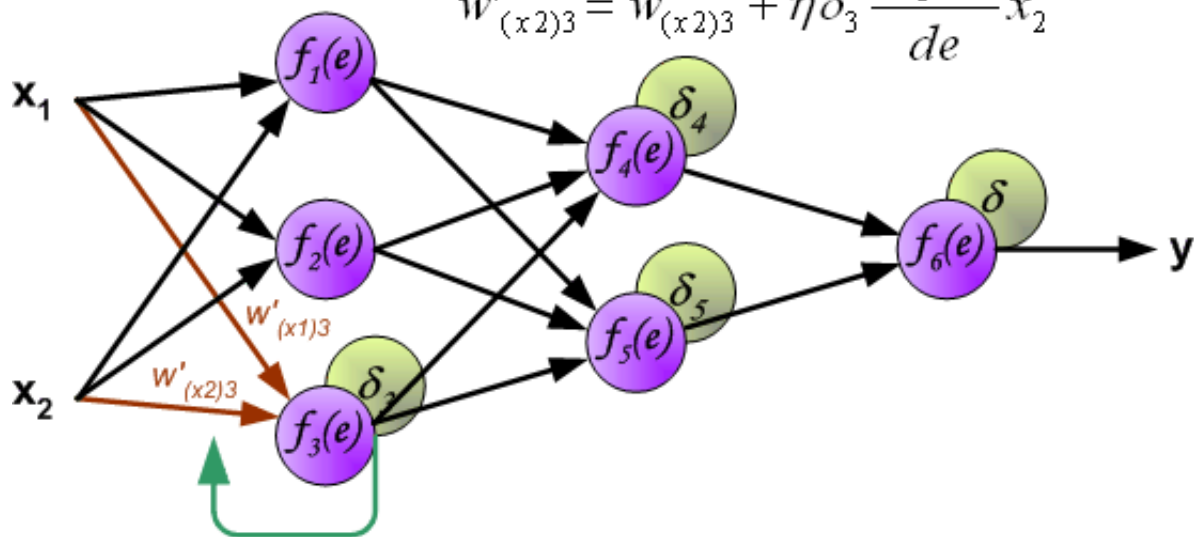$$w'_{(x2)1} = w_{(x2)1} + \eta\delta_1 \frac{df_1(e)}{de} x_2$$
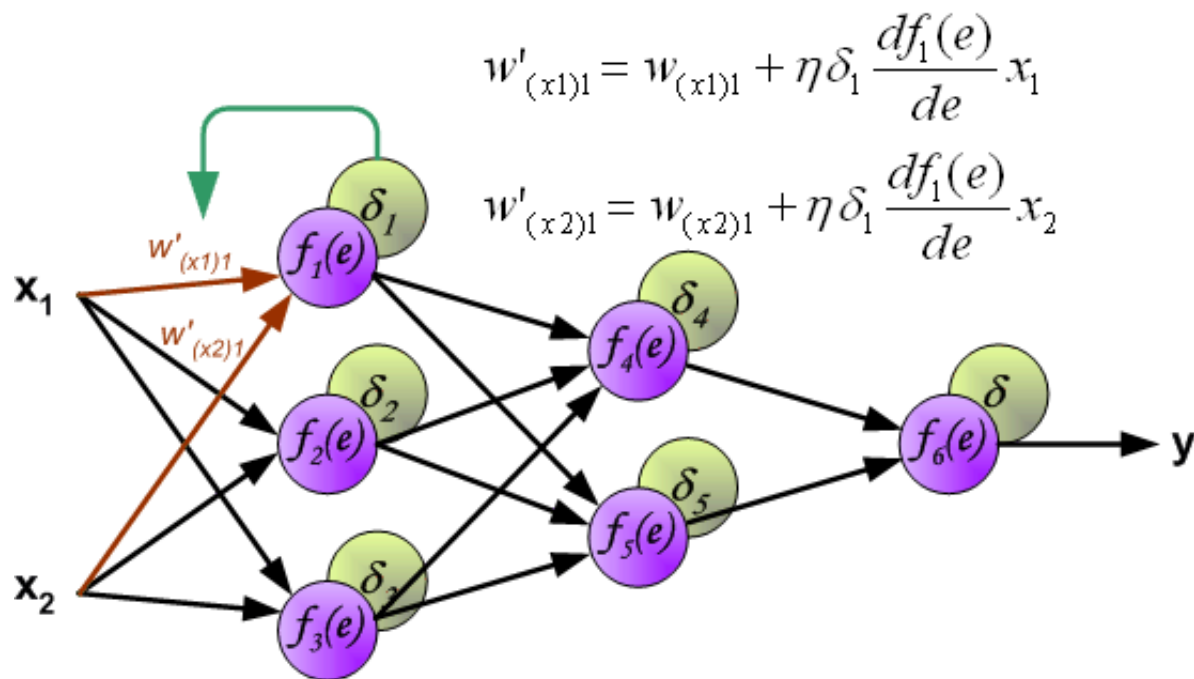
$$w'_{(x1)2} = w_{(x1)2} + \eta \delta_2 \frac{df_2(e)}{de} x_1$$

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

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

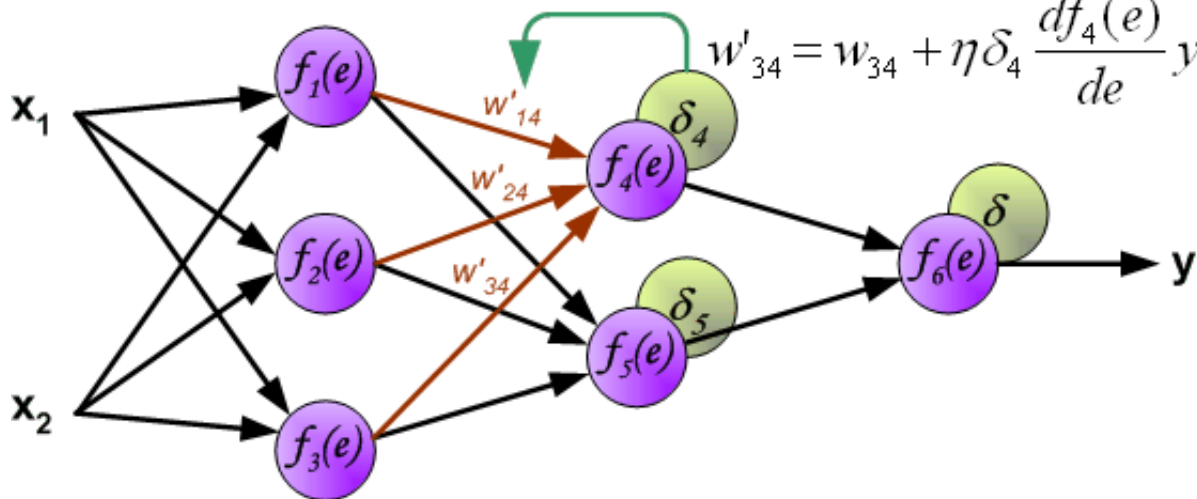$$w'_{(x2)3} = w_{(x2)3} + \eta \delta_3 \frac{df_3(e)}{de} x_2$$

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

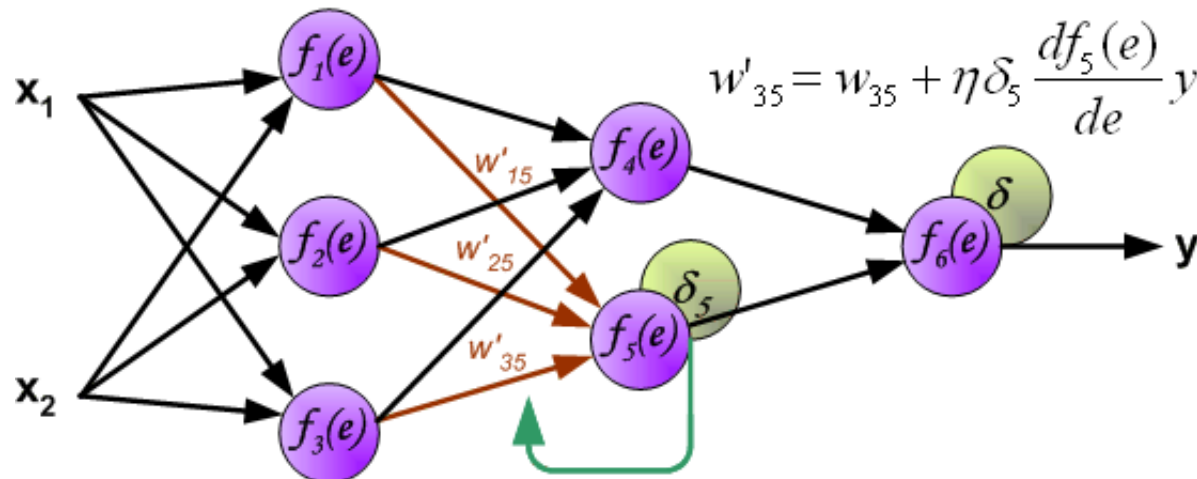$$w'_{24} = w_{24} + \eta \delta_4 \frac{df_4(e)}{de} y_2$$

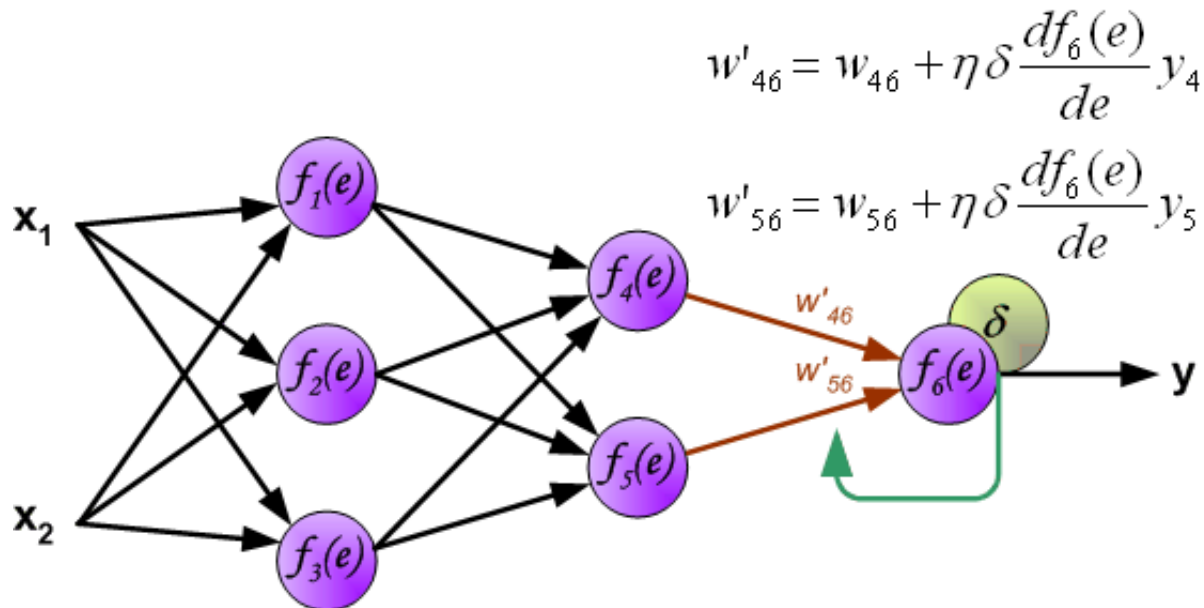$$w'_{34} = w_{34} + \eta \delta_4 \frac{df_4(e)}{de} y_3$$

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

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

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

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

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

Coefficient η affects network teaching speed. There are a few techniques to select this parameter. The first method is to start teaching process with large value of the parameter. While weights coefficients are being established the parameter is being decreased gradually. The second, more complicated, method starts teaching with small parameter value. During the teaching process the parameter is being increased when the teaching is advanced and then decreased again in the final stage. Starting teaching process with low parameter value enables to determine weights coefficients signs.

References
Ryszard Tadeusiewcz "Sieci neuronowe", Kraków 1992

Mariusz Bernacki
Przemysław Włodarczyk
mgr inż. Adam Gołda (2005)
Katedra Elektroniki AGH

Last modified: 06.09.2004

Webmaster