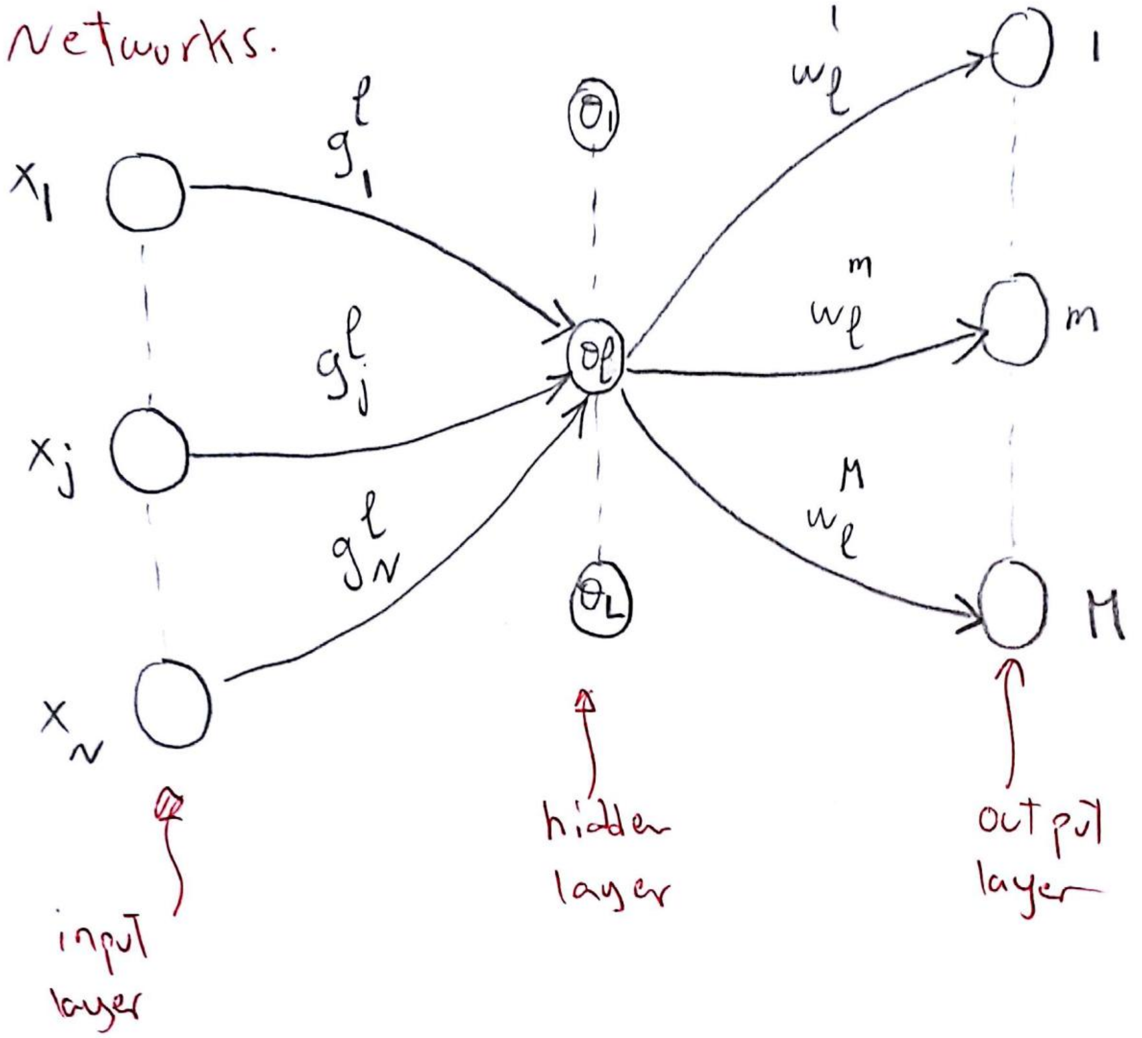
Back-Propagation Learning Algorithm for Multi-layer Feed-Forward Artificial Neural Networks.



Define:

F_h: activation function for hidden units

F_o: activation function for output units

O_h^ℓ: output of the ℓ^{th} hidden unit.

O_o^m: output of the m^{th} output unit.

Define:

$$net_h^l(x) := \sum_{j=1}^{N} g_j^l \cdot x_j - \theta_l$$

$$net_o^m(x) := \sum_{l=1}^{L} w_l^m \, o_h^l(x)$$

$$o_h^l(x) = f_h(net_h^l(x))$$

$$o_o^m(x) = f_o(net_h^l(x))$$

Now define the cost function:

$$E = \frac{1}{2} \sum_{p \in P} \sum_{m=1}^{M} \left[d^m(p) - o_o^m(p) \right]$$

P : training set. (p , $d(p)$ are known)

input desired

Apply the gradiant decent (steepest decent) algorithm to minimize E.

$$\frac{\partial E}{\partial g_k^t} = - \sum_{p \in P} \left[d^m(p) - O_o^m(p) \right] . f_o^{\prime}\left(net_o^m(p)\right) w_t^m f_h^{\prime}\left(net_h^t(p)\right) . p_k$$

$$\frac{dE}{d\theta_t} = \sum_{p \in P} \sum_{m=1}^{M} \left[d^m(p) - O_o^m(p) \right] f_o^{\prime}\left(net_o^m(p)\right) w_t^m f_h^{\prime}\left(net_h^t(p)\right)$$

$$\frac{dE}{dw_t^s} = - \sum_{p \in P} \left[d^s(p) - O_o^s(p) \right] . f_o^{\prime}\left(net_o^s(p)\right) O_h^t(p)$$

Algorithm:

(1) Initialize g_k^t, w_k^s, θ_t to small random values.

(2) Choose a training pattern p.

(3) compute : $net_h^l(p) = \sum_{j=1}^{N} g_j^l p_j - \theta_l$

$$O_h^l(p) = F_h\left(net_h^l(p)\right)$$

(4) compute: $net_o^m(p) = \sum_{l=1}^{L} w_l^m \, o_h^l(p)$

$$O_o^m(p) = F_o(net_o^m(p))$$

(5) compute:

$$\delta_o^s(p) = (d^s(p) - o_o^s(p)) \, f_o'(net_o^s(p))$$

$$s = 1, \ldots M$$

(6) compute: $\delta_h^t(p) = \delta_o^m(p) . w_t^m$

$$t = 1, \ldots, L$$

(7) compute: $\Delta w_t^s, \Delta \theta_t, \Delta g_k^t$ as:

$$\Delta w_t^s = \eta \, \delta_o^s(p) . o_h^t(p)$$

$$\Delta \theta_t = -\eta \, F_h'(net_h^t(p) . \delta_h^t(p))$$

$$\Delta g_k^t = \eta \, F_h'(net_h^t(p)) . P_k . \delta_h^t(p)$$

$t = 1, \ldots L$, $S = 1 \ldots M$, $k = 1, \ldots N$

(8) apply the updates:

$$w_t^s \leftarrow w_t^s + \Delta w_t^s$$

$$\theta_t \leftarrow \theta_t + \Delta \theta_t$$

$$g_k^t \leftarrow g_k^t + \Delta g_k^t$$

(9) Go to step 2, and repeat for all training patterns.

Activation functions can be as:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

or: $$f(x) = \frac{1}{1+e^{-x}} - \frac{1}{2}$$