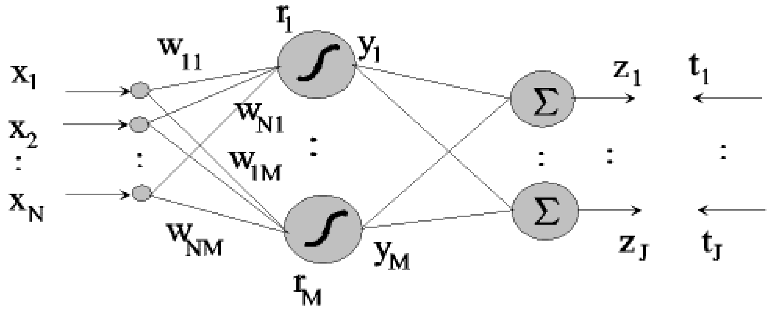
Derivation of the Backpropagation Algorithm for Feedforward Neural Networks



$$r_m = \sum_{n=1}^{N} x_n w_{nm} \qquad y_m = 1 / \{1 + \exp[-(r_m - b)]\}$$

$$z_j = \sum_{m=1}^{M} y_m u_{mj} \qquad E = \sum_{q=1}^{Q} \sum_{j=1}^{J} (z_j^{(q)} - t_j^{(q)})^2$$

The method of *steepest descent* from differential calculus is used for the derivation. To solve respectively for the weights $\{u_{mj}\}$ and $\{w_{nm}\}$, we use the standard formulation

$$u_{mj} \leftarrow u_{mj} - \eta_1[\partial E/\partial u_{mj}], \qquad w_{nm} \leftarrow w_{nm} - \eta_2[\partial E/\partial w_{nm}]$$

where the factors η_k, k = 1,2, are the step sizes, or *learning rates*.

Upon using the equations below the diagram above, we derive the two partial derivatives. To keep the values small for less error build-up, we divide E by the factor QJ.

$$\partial E/\partial u_{mj} = (2/QJ)\textstyle\sum_{(q=1,Q)}(z_j^{(q)} - t_j^{(q)})[\partial z_j^{(q)}/\partial u_{mj}] = (2/QJ)\textstyle\sum_{(q=1,Q)}(z_j^{(q)} - t_j^{(q)})[y_m] \qquad (1)$$

$$\begin{aligned}\partial E/\partial w_{nm} &= (2/QJ)\textstyle\sum_{(q=1,Q)}(z_j^{(q)} - t_j^{(q)})[\partial z_j^{(q)}/\partial y_m^{(q)}][\partial y_m^{(q)}/\partial r_m][\partial r_m/\partial w_{nm}] \\ &= (2/QJ)\textstyle\sum_{(q=1,Q)}(z_j^{(q)} - t_j^{(q)})[u_{mj}][\partial y_m^{(q)}/\partial r_m][x_n] \\ &= (2/QJ)\textstyle\sum_{(q=1,Q)}(z_j^{(q)} - t_j^{(q)})[u_{mj}][y_m^{(q)}(1-y_m^{(q)})][x_n] \qquad (2)\end{aligned}$$

The only part that needs more explanation is that $\partial y_m^{(q)}/\partial r_m = y_m^{(q)}(1-y_m^{(q)})$. We show that now.

$$\begin{aligned}\partial y_m^{(q)}/\partial r_m &= \partial/\partial r_m \{1 + \exp[-r_m + b]\}^{-1} = (-1)\{1 + \exp[-r_m + b]\}^{-2} \exp[-r_m + b](-1) \\ &= [y_m^{(q)}]^2\{1 + \exp[-r_m + b] - 1\} \quad \text{(upon adding 1 and then subtrating 1)} \\ &= [y_m^{(q)}]^2\{(1/y_m^{(q)}) - 1) = [y_m^{(q)}]^2\{(1-y_m^{(q)})/y_m^{(q)}\} = y_m^{(q)}(1 - y_m^{(q)}) \qquad (3)\end{aligned}$$