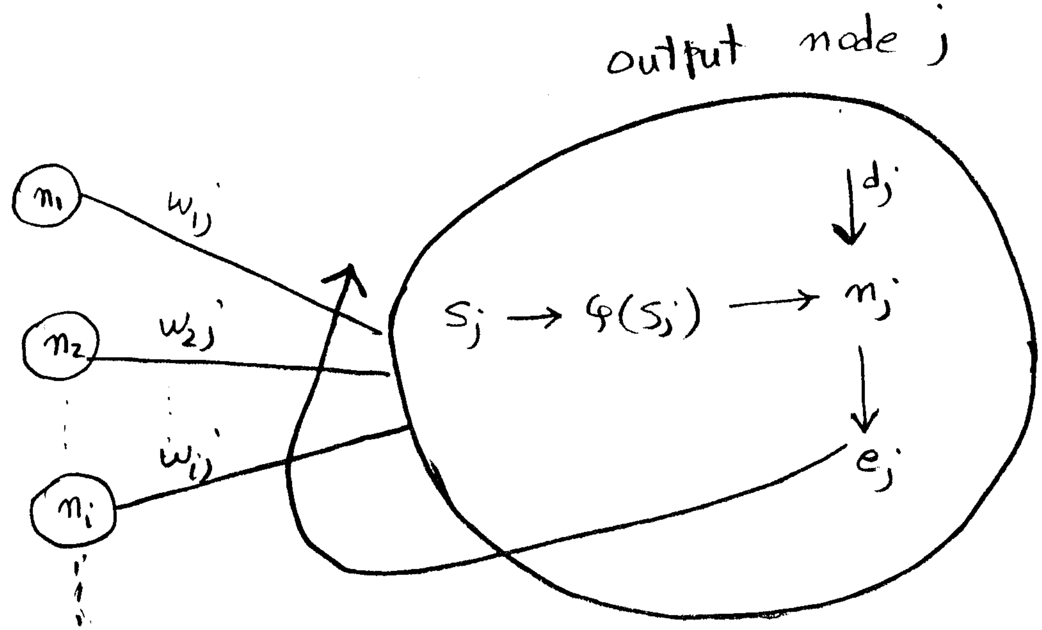
Neural Networks

Error - Correction Learning (Memory based learning; Hebbian learning; Competitive learn

The Back Propagation (Backprop) Algorithm Boltzmann learning)

Consider a node j. Suppose n_i is the input from the i^{th} node of the previous layer to the j^{th} node of the present layer.



Where:

Neural Networks

$$s_j = \sum_i n_i w_{ij}$$

$$n_j = \varphi(s_j) = \frac{1}{1 + e^{-s_j}}$$, predicted output of node j

d_j = desired output of node j

$e_j = n_j - d_j$, error of the output node j

The overall error for all output nodes is:

$$\sum_{\text{all output nodes}} e_j = \sum (n_j - d_j)$$

Neural Networks

Define the "error function", "cost function", "loss function":

$$E = \frac{1}{2} \sum_{\text{all output nodes}} e_j^2$$

$$= \frac{1}{2} \sum (n_j - d_j)^2$$

The goal is to minimize E.

The backprop algorithm is a step-by-step weight adjustment procedure designed to make the predicted output n_j closer and closer to the desired output d_j.

Neural Networks

The correction (adjustment) to the weights w_{ij}'s at step $t+1$ is proportional to the partial derivative of E with respect to weights, w_{ij}'s, at step t, i.e.:

$$w_{ij}(t+1) = w_{ij}(t) + \lambda \frac{\partial E}{\partial w_{ij}(t)},$$

where λ, the proportionality constant, is called the learning (-rate) parameter.

The correction $\lambda \dfrac{\partial E}{\partial w_{ij}(t)}$ is called "delta rule" denoted by (widrow-Hoff rule)

$$\Delta w_{ij}(t) = \lambda \frac{\partial E}{\partial w_{ij}(t)}.$$

Neural Networks

Now, according to the chain rule:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial n_j} \cdot \frac{\partial n_j}{\partial s_j} \cdot \frac{\partial s_j}{\partial w_{ij}}$$

$$\frac{\partial E}{\partial n_j} = (n_j - d_j),$$

$$\frac{\partial n_j}{\partial s_j} = \frac{e^{-s_j}}{(1+e^{-s_j})^2} = e^{-s_j} n_j^2 = (1-n_j)\, n_j$$

$$\frac{\partial s_j}{\partial w_{ij}} = n_i$$

$$\therefore \quad w_{ij}(t+1) = w_{ij}(t) + \lambda \frac{\partial E}{\partial n_j} \cdot \frac{\partial n_j}{\partial s_j} \cdot \frac{\partial s_j}{\partial w_{ij}}$$

Neural Networks

Notice that

$\dfrac{\partial E}{\partial n_j}$ represents the change in error function, E, due to the change in output node, n_j.

$\dfrac{\partial E}{\partial n_j} \cdot \dfrac{\partial n_j}{\partial s_j} = \dfrac{\partial E}{\partial s_j}$ represents the change in error function, E, due to the change in input, s_j to a given node j.

$\dfrac{\partial E}{\partial n_j} \cdot \dfrac{\partial n_j}{\partial s_j} \cdot \dfrac{\partial s_j}{\partial w_{ij}} = \dfrac{\partial E}{\partial w_{ij}}$ represents the change in error function, E, due to the change in weight, w_{ij}, from a given node i on the previous layer to the node j of the current layer.