JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Introduction to Neural Networks

Johns Hopkins University
Engineering for Professionals Program

605-447/625-438

Dr. Mark Fleischer

Module 5.1: The Feed-forward, Back Propagation Algorithm

This Sub-Module Covers …

- Extends the Perceptron Delta Function to handle multi-layer networks.
- We will derive the feed-forward, back-propagation algorithm.
    - Similar in spirit the the Perceptron Delta Function.
    - Calculus based optimization technique.
- Next video we go through a computational example.

The Perceptron Delta Function

Using the Chain Rule, we get:

$$\frac{\partial E_j}{\partial w_{ij}} = \frac{\partial E_j}{\partial e_j} \times \frac{\partial e_j}{\partial y_j} \times \frac{\partial y_j}{\partial A_j} \times \frac{\partial A_j}{\partial w_{ij}}$$

Factors: 1, 2, 3, 4

The Perceptron Delta Function

$$\frac{\partial E_j}{\partial w_{ij}} = \frac{\partial E_j}{\partial e_j} \times \frac{\partial e_j}{\partial y_j} \times \frac{\partial y_j}{\partial A_j} \times \frac{\partial A_j}{\partial w_{ij}}$$

$$\frac{\partial E}{\partial \omega_{ij}} = -e_j[1 - y_j]y_j x_i$$

From input *i* to output *j.*

The Perceptron Delta Function

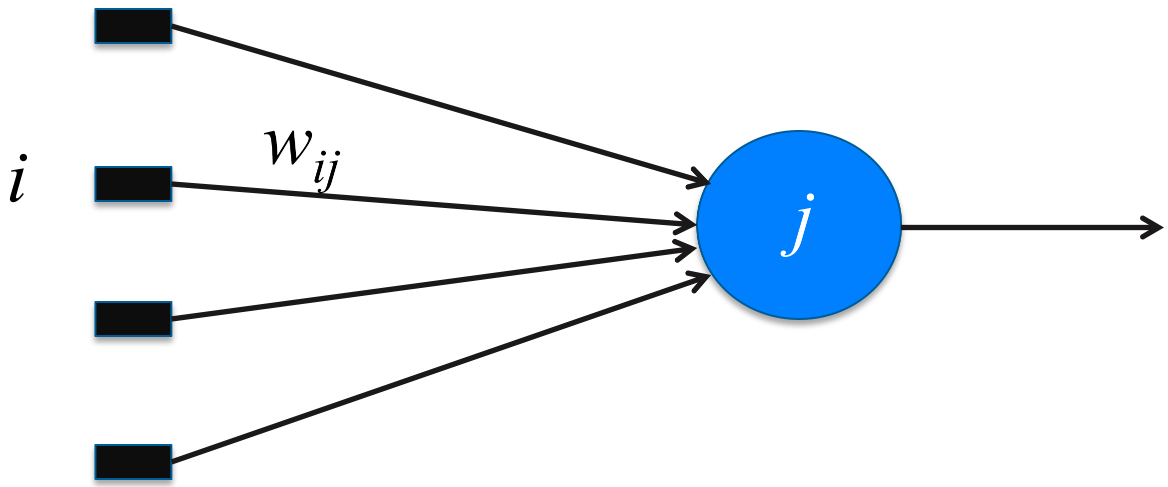
$$\frac{\partial E}{\partial \omega_{ij}} = -e_j[1 - y_j]y_j x_i$$

and letting $\delta_j = \quad e_j[1 - y_j]y_j$ then

$$\Delta\omega_{ij} = \eta\frac{\partial E}{\partial \omega_{ij}} = \eta\delta_j x_i.$$

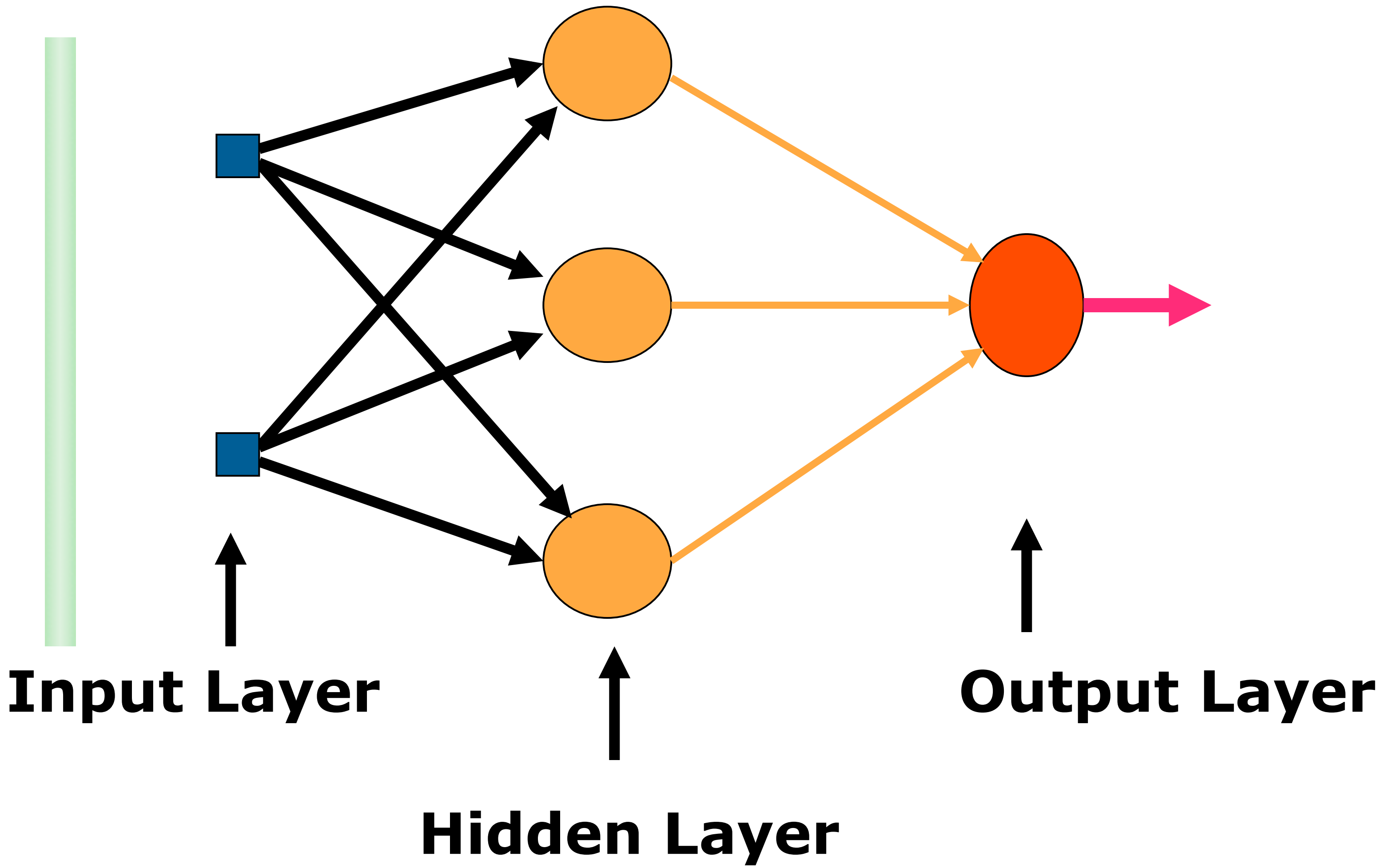
From input *i* to output *j*.

The Perceptron



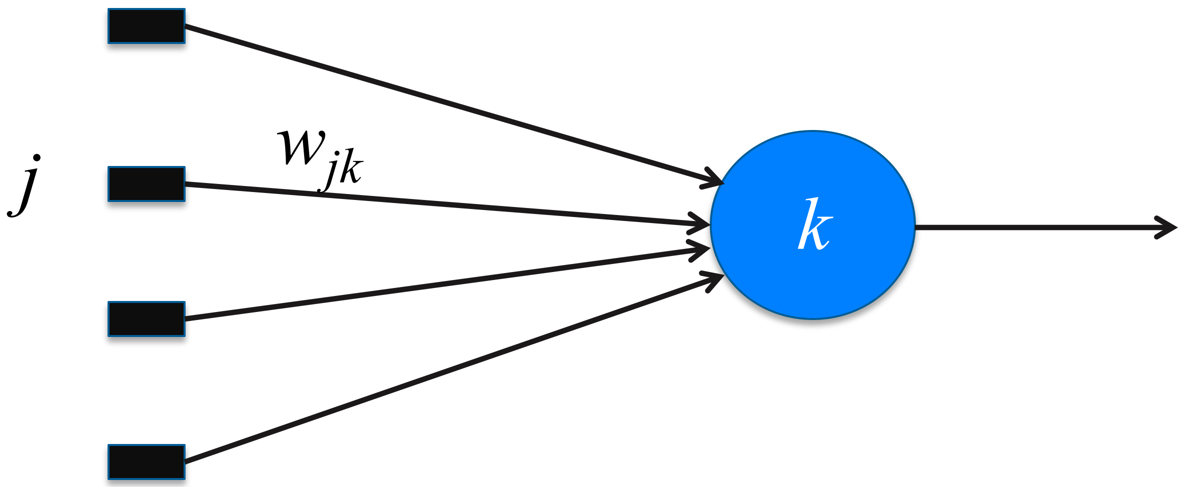
$$\frac{\partial E_j}{\partial w_{ij}}$$

A Multi-layered Network



The Perceptron

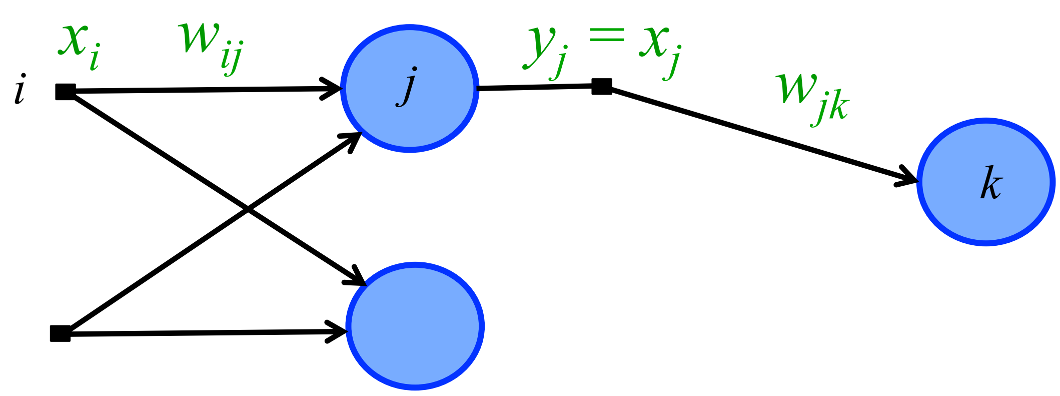
New naming conventions for the output nodes.



$$\frac{\partial E_k}{\partial w_{jk}}$$

The Feed-forward Back-propagation Algorithm

Notational Conventions for a multi-layered network



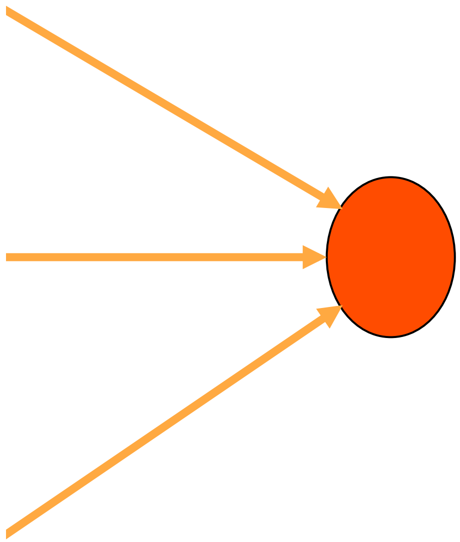
The Gradient Vector

was just $\left(\frac{\partial E_k}{\partial w_{1k}}, \frac{\partial E_k}{\partial w_{2k}}, \ldots, \frac{\partial E_k}{\partial w_{nk}} \right)$

but in a multi-layer network, it becomes

$$\left(\underbrace{\frac{\partial E_k}{\partial w_{1k}}, \frac{\partial E_k}{\partial w_{2k}}, \ldots, \frac{\partial E_k}{\partial w_{nk}}}_{\text{Output Layer Node}}, \underbrace{\frac{\partial E_k}{\partial w_{1j_1}}, \frac{\partial E_k}{\partial w_{2j_1}}, \ldots, \frac{\partial E_k}{\partial w_{mj_1}}}_{\text{Hidden Layer Node 1}}, \underbrace{\frac{\partial E_k}{\partial w_{1j_2}}, \frac{\partial E_k}{\partial w_{2j_2}}, \ldots, \frac{\partial E_k}{\partial w_{mj_2}}}_{\text{Hidden Layer Node 2}}, \ldots \right)$$

A Multi-layered Network



The Feed-forward Back-Propagation Algorithm

From this

$$\frac{\partial E_k}{\partial w_{jk}} = \frac{\partial E_k}{\partial e_k} \times \frac{\partial e_k}{\partial y_k} \times \frac{\partial y_k}{\partial A_k} \times \frac{\partial A_k}{\partial w_{jk}}$$

to this

$$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial x_j} \times \frac{\partial x_j}{\partial A_j} \times \frac{\partial A_j}{\partial w_{ij}}$$

The Feed-forward Back-Propagation Algorithm

$$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial x_j} \times \frac{\partial x_j}{\partial A_j} \times \frac{\partial A_j}{\partial w_{ij}}$$

Factors: 1, 2, 3

The Feed-forward Back-Propagation Algorithm

Factor 1:

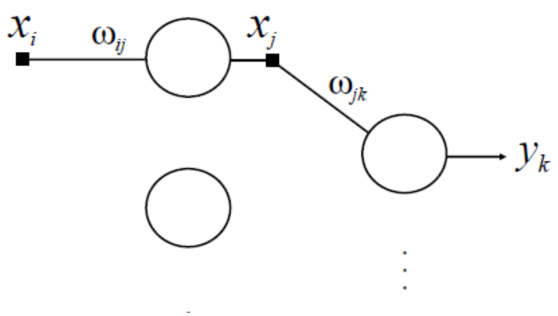
$$\frac{\partial E_k}{\partial x_j} = \frac{\partial}{\partial x_j}\frac{1}{2}\sum_k e_k^2$$

$$= \frac{1}{2}\sum_k \frac{\partial}{\partial x_j} e_k^2$$

$$= \sum_k e_k \frac{\partial e_k}{\partial x_j}$$

$$= \sum_k e_k \frac{\partial e_k}{\partial A_k} \cdot \frac{\partial A_k}{\partial x_j}$$

The Feed-forward Back-propagation Algorithm



$$\sum_k \boxed{e_k \frac{\partial e_k}{\partial A_k}} \cdot \boxed{\frac{\partial A_k}{\partial x_j}}$$

$$e_k = d_k - y_k$$
$$= d_k - f_k(A_k)$$
$$\therefore \frac{\partial e_k}{\partial A_k} = -f_k'(A_k)$$

Since $A_k = \sum_{j=1}^{M} x_j w_{jk}$

then $$\frac{\partial A_k}{\partial x_j} = \frac{\partial \sum_{j=1}^{M} x_j w_{jk}}{\partial x_j}$$
$$= w_{jk}$$

The Feed-forward Back-propagation Algorithm

Factor 1:

$$\frac{\partial E_k}{\partial x_j} = \frac{\partial}{\partial x_j}\frac{1}{2}\sum_k e_k^2$$

$$= \frac{1}{2}\sum_k \frac{\partial}{\partial x_j} e_k^2$$

$$= \sum_k e_k \frac{\partial e_k}{\partial x_j}$$

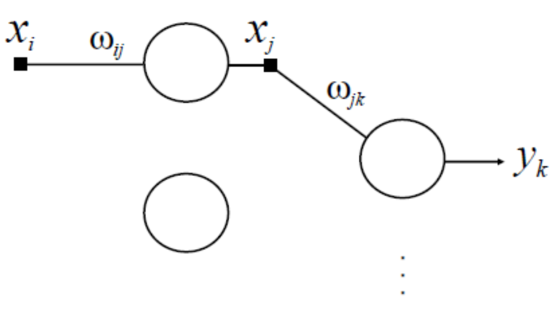
$$= \sum_k e_k \frac{\partial e_k}{\partial A_k} \cdot \frac{\partial A_k}{\partial x_j}$$

$$\frac{\partial E_k}{\partial x_j} = -\sum_k e_k f_k'(A_k) w_{jk}$$

$$= \sum_k \delta_k w_{jk}$$

FFBP: Factors 2 and 3

$$\frac{\partial x_j}{\partial A_j} \times \frac{\partial A_j}{\partial w_{ij}} = [1 - x_j]x_j x_i$$



$$A_j = \sum_i w_{ij} x_i$$

FFBP --- All Together Now

$$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial x_j} \times \frac{\partial x_j}{\partial A_j} \times \frac{\partial A_j}{\partial w_{ij}}$$

$$\sum_k \delta_k w_{jk} \quad \times \quad [1 - x_j]x_j \quad \times \quad x_i$$

FFBP --- All Together Now

$$\frac{\partial E}{\partial w_{ij}} = \boxed{[1 - x_j]x_j\left(\sum_k \delta_k w_{jk}\right)} x_i = \delta_j x_i$$

$$\Delta w_{ij} = \eta \delta_j x_i$$

A Multi-Layered Network