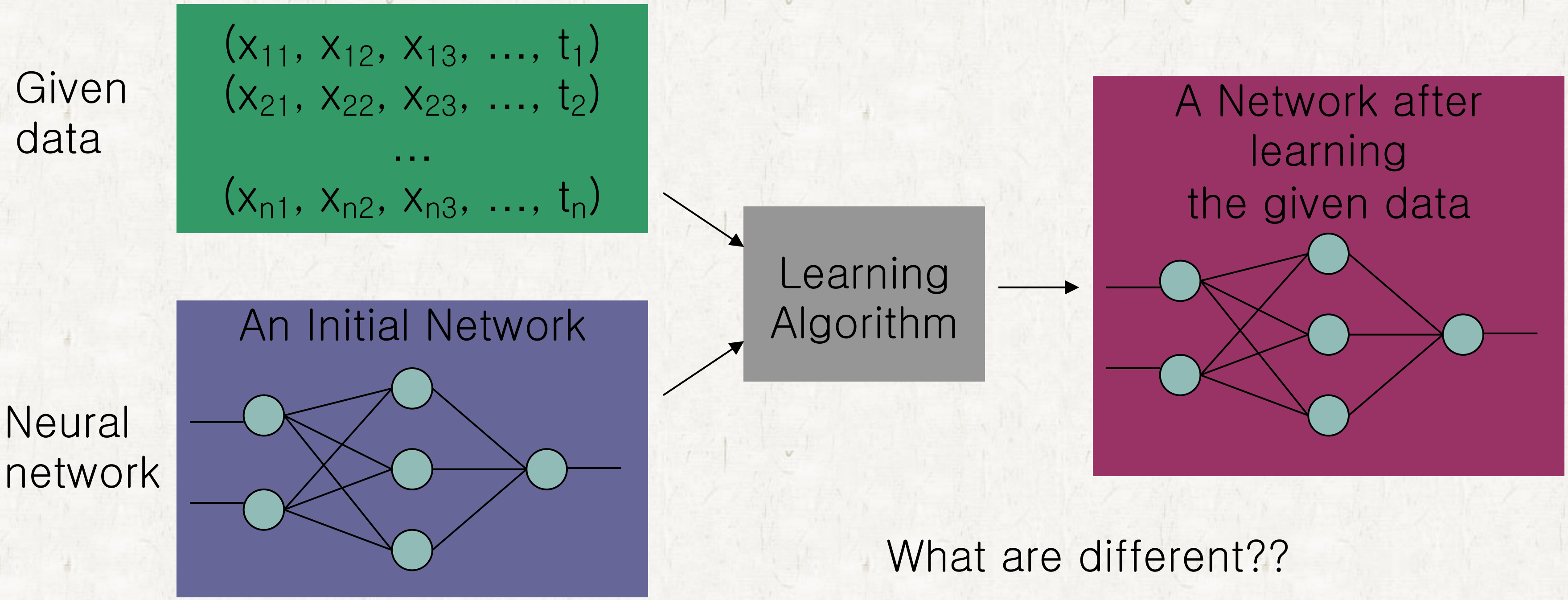
Error Back Propagation

Learning Algorithm (1)

- Preparation for Learning
  - Given input-output data of the target function to learn
  - Given structure of network (# of nodes in hidden layer)
  - Randomly initialized weights



What are different??

Learning Algorithm (2)

- Basic Idea of Learning
  - Find weights $\boldsymbol{w} = (w_1, w_2, \dots, w_n)$ so that

    $$NN(\boldsymbol{w}, \boldsymbol{x}) = \boldsymbol{t} \quad \text{for all } (\boldsymbol{x}, t)$$

  - Find weights $\boldsymbol{w} = (w_1, w_2, \dots, w_n)$ which minimize

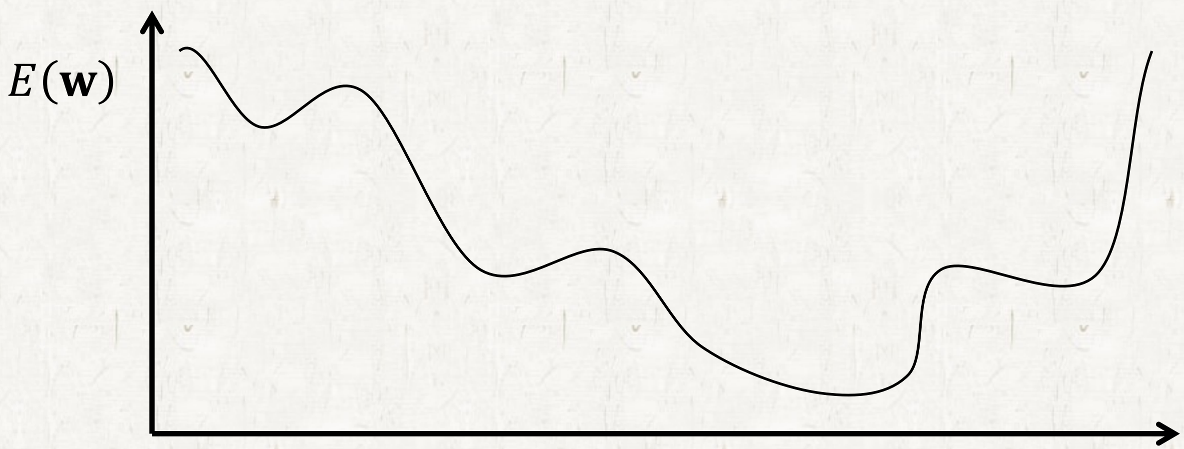
$$E(\boldsymbol{w}) = \sum_{(\boldsymbol{x},t) \in Data} \left(t - NN(\boldsymbol{w}, \boldsymbol{x})\right)^2 \qquad \boldsymbol{w} = (w_1, w_2, \dots, w_n)$$

Gradient Descent Method (1)

- How?

Find weights $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ which minimize

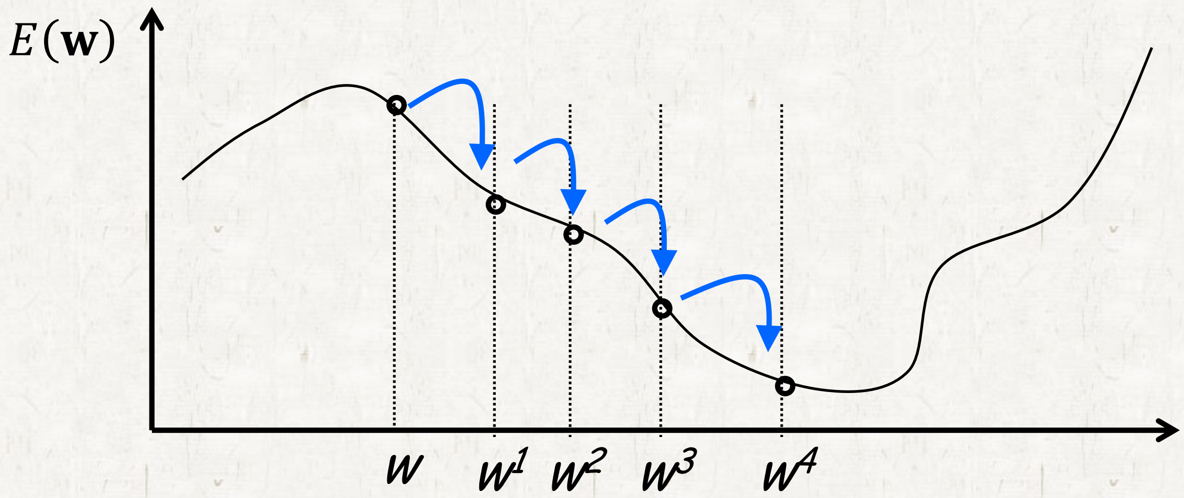
$$E(\mathbf{w}) = \sum_{(\mathbf{x},t)\in Data} \left(y - NN(\mathbf{x};\mathbf{w})\right)^2$$



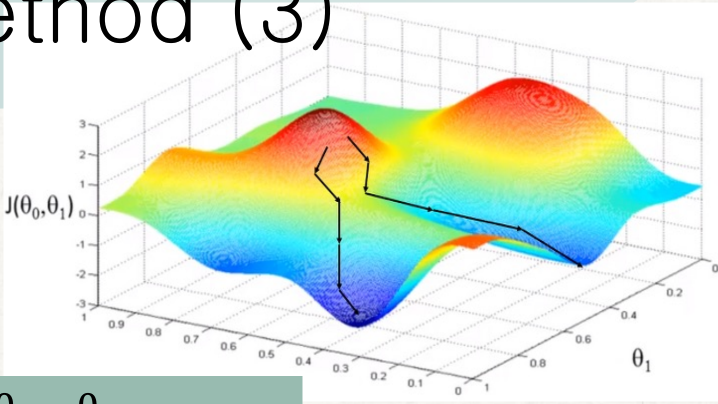
Gradient Descent Method (2)

4. Repeat until the gradient is zero

$$w^{t+1} = w^t - \eta \left.\frac{\partial E}{\partial w}\right|_{w=w^t}$$



Gradient Descent Method (3)



- Multi-variable case

Randomly choose an initial solution, $w_0^0\ w_1^0$

Repeat

$$w_0^{t+1} = w_0^t - \eta \left.\frac{\partial E}{\partial w_0}\right|_{w_0=w_0^t, w_1=w_1^t}$$

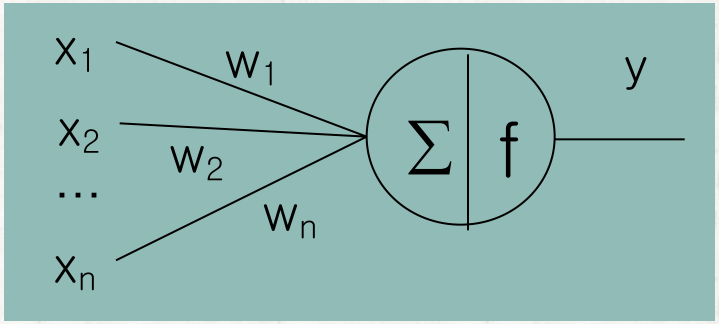
$$w_1^{t+1} = w_1^t - \eta \left.\frac{\partial E}{\partial w_1}\right|_{w_0=w_0^t, w_1=w_1^t}$$

Until stopping condition is satisfied

Activation Function (1)

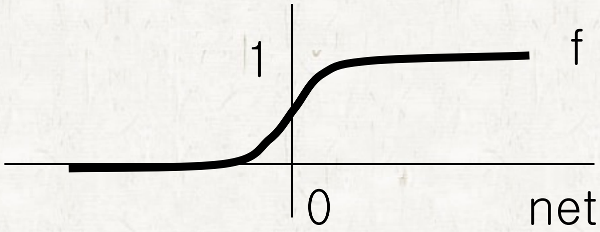
- Error Back Propagation
  - ANN learning algorithm based on gradient descent method
  - Using derivatives to change the weights so that the error is minimized

  - Hard limit is not differentiable. -> Sigmoid



$net = x_1 w_1 + x_2 w_{j2} + \ldots + x_n w_n$

Sigmoid



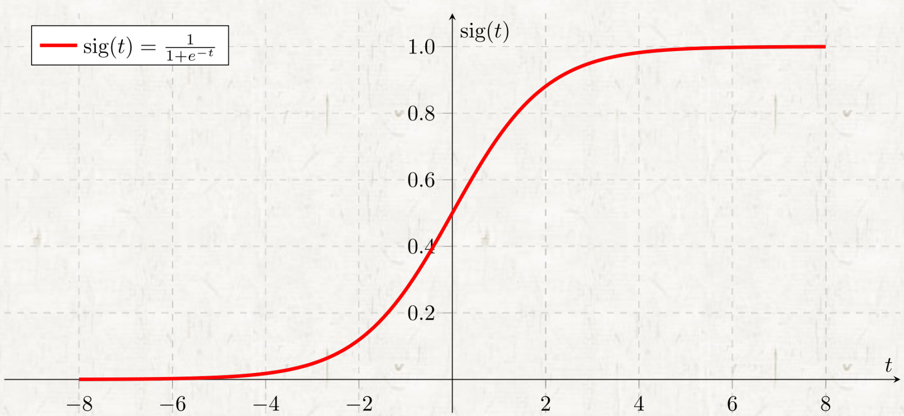
$y = sigmoid(net)$

Activation Function (2)

- Sigmoid function

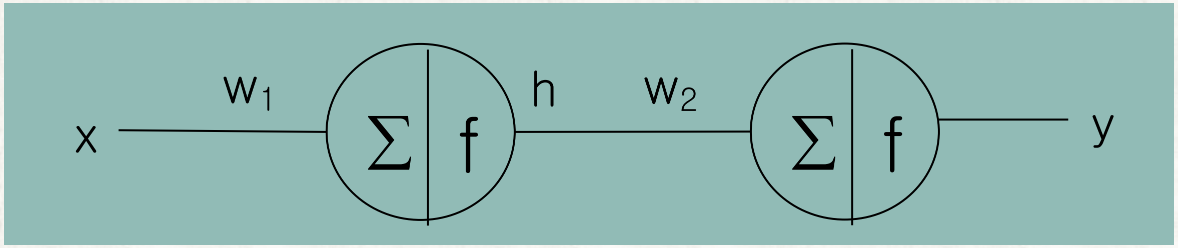
$$y = \frac{1}{1+e^{-x}}$$

$$\begin{aligned}
y' &= \left(\frac{1}{1+e^{-x}}\right)^2 e^{-x} \\
&= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right) \\
&= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{1+e^{-x}-1}{1+e^{-x}}\right) \\
&= \left(\frac{1}{1+e^{-x}}\right)\left(1-\frac{1}{1+e^{-x}}\right) \\
&= y(1-y)
\end{aligned}$$



Simple Examples (1)

- Training of a Simple Neural Network
  - Let's assume that there is one training data (x_t, y_t)



$$s_1 = x_t \cdot w_1$$

$$h = sigmoid(s_1)$$

$$s_2 = h \cdot w_2$$

$$y = sigmoid(s_2)$$

$$E = \frac{1}{2}(y_t - y)^2$$

$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_2}\frac{\partial s_2}{\partial w_2}$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_2}\frac{\partial s_2}{\partial h}\frac{\partial h}{\partial s_1}\frac{\partial s_1}{\partial w_1}$$

Simple Examples (2)

- Training of a Simple Neural Network

$$s_1 = x_t \cdot w_1$$

$$h = sigmoid(s_1)$$

$$s_2 = h \cdot w_2$$

$$y = sigmoid(s_2)$$

$$E = \frac{1}{2}(y_t - y)^2$$

$$\frac{\partial s_1}{\partial w_1} = x_t$$

$$\frac{\partial h}{\partial s_1} = h(1-h)$$

$$\frac{\partial s_2}{\partial w_2} = h \qquad \frac{\partial s_2}{\partial h} = w_2$$

$$\frac{\partial y}{\partial s_2} = y(1-y)$$

$$\frac{\partial E}{\partial y} = -(y_t - y)$$

$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_2}\frac{\partial s_2}{\partial w_2}$$

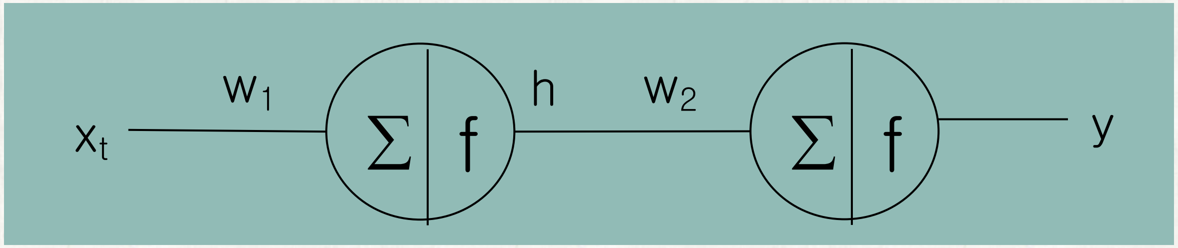
$$= -(y_t - y)y(1-y)h$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_2}\frac{\partial s_2}{\partial h}\frac{\partial h}{\partial s_1}\frac{\partial s_1}{\partial w_1}$$

$$= -(y_t - y)y(1-y)w_2 h(1-h)x_t$$

Simple Examples (3)

- Training of a Simple Neural Network



$$(x_t, y_t) = (1,1), w_1 = 1, w_2 = 1, \eta = 0.1$$

$$s_1 = x_t \cdot w_1 = 1$$

$$h = sigmoid(s_1) = 0.731$$

$$s_2 = h \cdot w_2 = 0.731$$

$$y = sigmoid(s_2) = 0.675$$

$$E = \frac{1}{2}(y_t - y)^2 = \frac{0.343^2}{2}$$

$$\frac{\partial E}{\partial w_2} = -(y_t - y)y(1-y)h$$
$$= -0.325 \cdot 0.675 \cdot 0.325 \cdot 0.731$$

$$\frac{\partial E}{\partial w_1} = -(y_t - y)y(1-y)w_2 h(1-h)x_t$$
$$= -0.325 \cdot 0.675 \cdot 0.325 \cdot 1 \cdot 0.731 \cdot 0.269 \cdot 1$$

Simple Examples (4)

- Training of a Simple Neural Network

$$\frac{\partial E}{\partial w_2} = -(y_t - y)y(1-y)h$$
$$= -0.325 \cdot 0.675 \cdot 0.325 \cdot 0.731 = -0.052$$

$$(x_t, y_t) = (1,1)$$
$$w_1^0 = 1$$
$$w_2^0 = 1$$
$$\eta = 0.1$$

$$\frac{\partial E}{\partial w_1} = -(y_t - y)y(1-y)w_2 h(1-h)x_t$$
$$= -0.325 \cdot 0.675 \cdot 0.325 \cdot 1 \cdot 0.731 \cdot 0.269 \cdot 1 = -0.014$$

$$w_1^1 = w_1^0 - \eta \frac{\partial E}{\partial w_1}$$

$$1.0014 = 1 + 0.1 \cdot 0.014$$

$$w_2^1 = w_2^0 - \eta \frac{\partial E}{\partial w_2}$$

$$1.0052 = 1 + 0.1 \cdot 0.052$$

Randomly choose an initial solution, w_1^0 w_2^0

Repeat

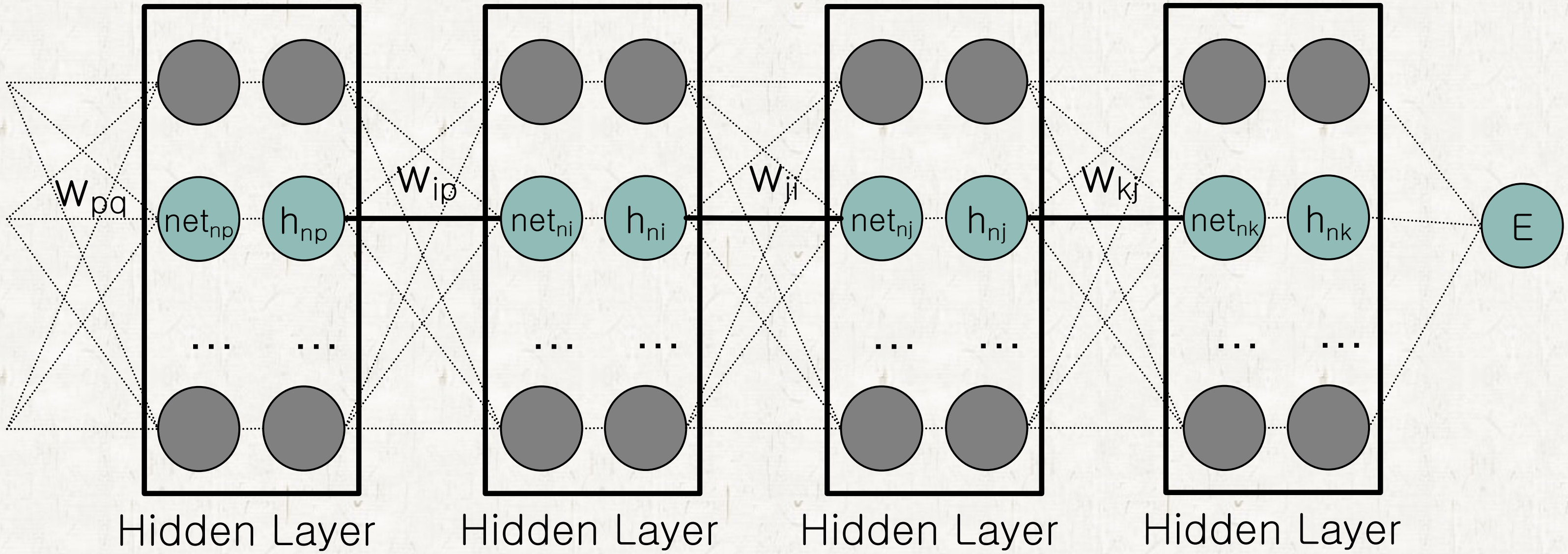
$$w_1^{t+1} = w_1^t - \eta \left.\frac{\partial E}{\partial w_0}\right|_{w_1=w_1^t, w_2=w_2^t}$$

$$w_2^{t+1} = w_2^t - \eta \left.\frac{\partial E}{\partial w_2}\right|_{w_1=w_1^t, w_2=w_2^t}$$

Until stopping condition is satisfied

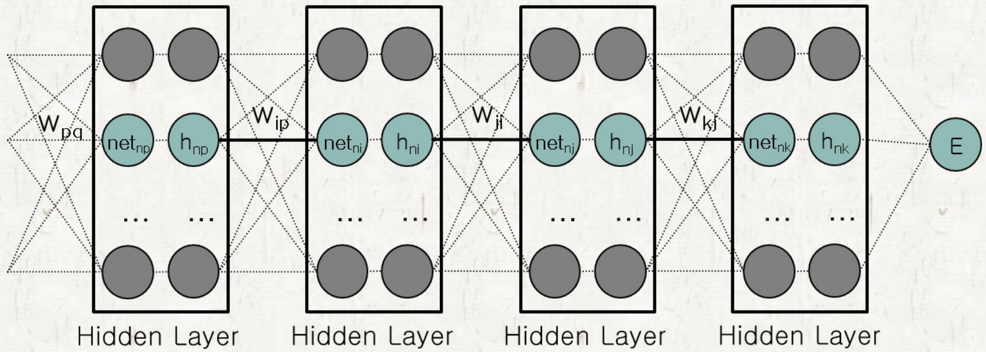
Error Back Propagation (1)

- Weights between deep layers
  - For D_n=(x_{n1}, x_{n2}, …, x_{nd}, t_{n1} , t_{n2}, …, t_{nm})



Error Back Propagation (2)

- Weights between deep layers



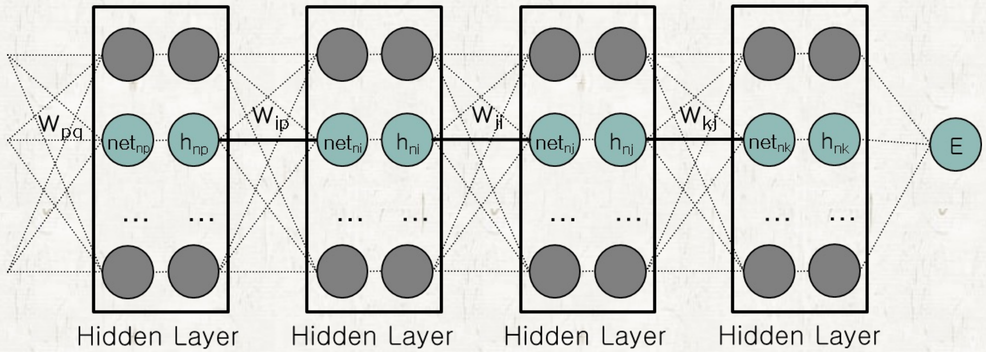
$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial net_{nk}} \frac{\partial net_{nk}}{\partial w_{kj}} = \delta_{nk} h_{nj} \qquad \delta_{nk} = \frac{\partial E}{\partial net_{nk}}$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial net_{nj}} \frac{\partial net_{nj}}{\partial w_{ji}} = \delta_{nj} h_{ni} \qquad \delta_{nj} = \frac{\partial E}{\partial net_{nj}}$$

$$\frac{\partial E}{\partial w_{ip}} = \frac{\partial E}{\partial net_{ni}} \frac{\partial net_{ni}}{\partial w_{ip}} = \delta_{ni} h_{np} \qquad \delta_{ni} = \frac{\partial E}{\partial net_{ni}}$$

Error Back Propagation (3)

- Weights between deep layers



$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial net_{nk}} \frac{\partial net_{nk}}{\partial w_{kj}} = \delta_{nk} h_{nj} \qquad \delta_{nk} = \frac{\partial E}{\partial h_{nk}} \frac{\partial h_{nk}}{\partial net_{nk}}$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial net_{nj}} \frac{\partial net_{nj}}{\partial w_{ji}} = \delta_{nj} h_{ni} \qquad \delta_{nj} = \left(\sum_{k=1}^{K} \delta_{nk} w_{kj} \right) \frac{\partial h_{nj}}{\partial net_{nj}}$$

$$\frac{\partial E}{\partial w_{ip}} = \frac{\partial E}{\partial net_{ni}} \frac{\partial net_{ni}}{\partial w_{ip}} = \delta_{ni} h_{np} \qquad \delta_{ni} = \left(\sum_{j=1}^{J} \delta_{nj} w_{ji} \right) \frac{\partial h_{ni}}{\partial net_{ni}}$$