A Learning Algorithm:
Error Back Propagation Algorithm

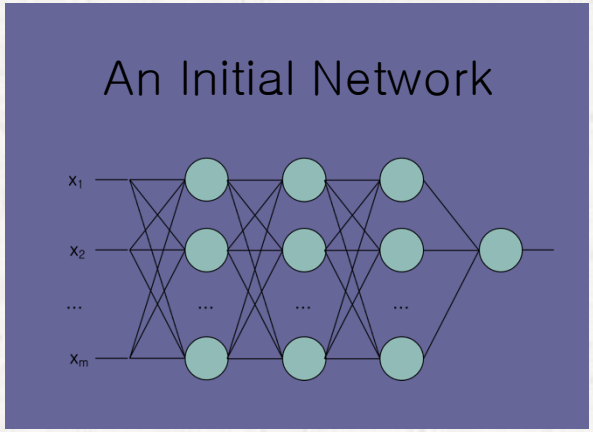
Introduction

- 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

① Given data

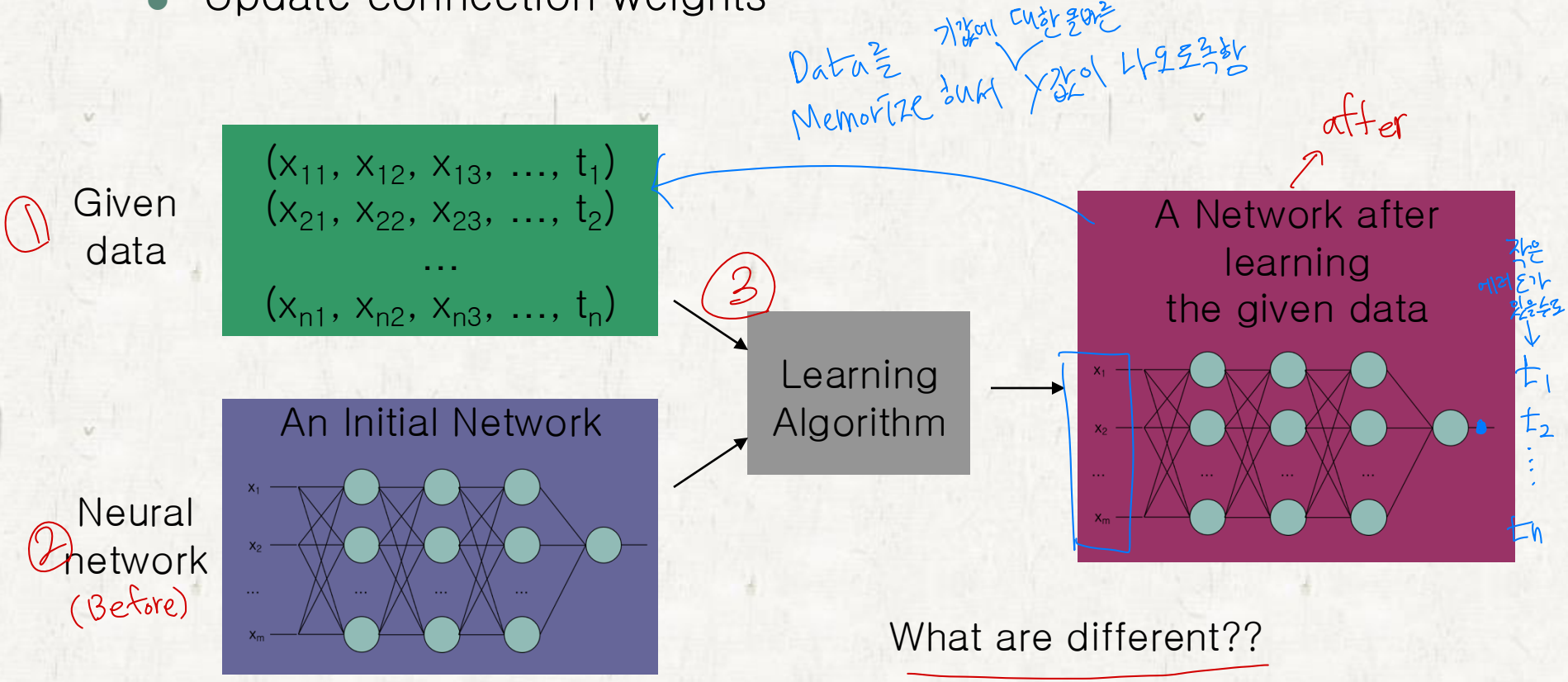
$(x_{11}, x_{12}, x_{13}, \ldots, t_1)$
$(x_{21}, x_{22}, x_{23}, \ldots, t_2)$
$\ldots$
$(x_{n1}, x_{n2}, x_{n3}, \ldots, t_n)$

② Neural network



Introduction

- Learning Algorithm
  - Update connection weights



What are different??

⇒connection weight가 달라진다
(structure of NN은 같음)

Recap: Linear Regression

- Linear regression with gradient descent method
  - Define an error function, E
  - Find **w** to minimize E

3	1	-1
4	3	1
6	1	0.5
2	3	3
5	9	-2
4	8	5
1	2	2
4	4	-2
4	1	-4
5	5	1

Training Data

$$Error = \sum_{(\mathbf{x},t)Data} (t - L(\mathbf{x};\mathbf{w}))^2$$

where

$$L(\mathbf{x};\mathbf{w}) = w_0 + w_1 x_1 + x_2 x_2 + \cdots$$

If E is minimized,
L will output like this!

$L(3,1) \approx -1$
$L(4,3) \approx 1$
$L(6,1) \approx 0.5$
$L(2,3) \approx 3$
$L(5,9) \approx -2$
$L(4,8) \approx 5$
$L(1,2) \approx 2$
$L(4,4) \approx -2$
$L(4,1) \approx -4$
$L(5,5) \approx 1$

Model Output

Recap: Logistic Regression

= classification

- Logistic regression with gradient descent method
  - Define an error function, E
  - Find **w** to minimize E

cross entropy

$$E = -\sum_{(\mathrm{x},t)\in \mathrm{Data}} \Big(t \log L(\mathrm{x};\mathrm{w}) + (1-t)\log(1-L(\mathrm{x};\mathrm{w})) \Big)$$

where

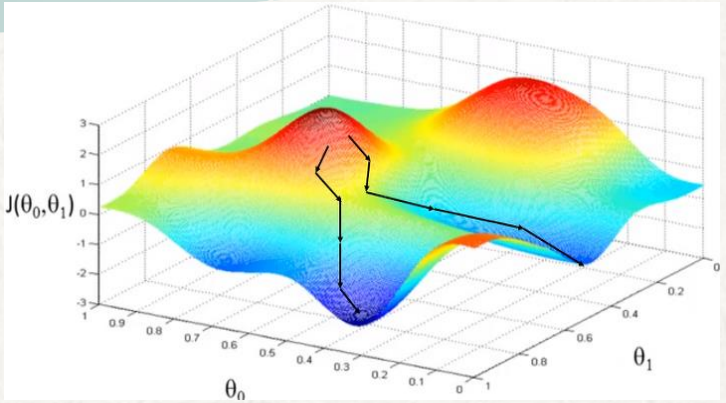
$$L(\mathbf{x};\mathbf{w}) = \frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2+\cdots)}}$$

Training Data		Model Output
(3, 1, 0)		$L(3,1) \approx 0$
(4, 3, 1)		$L(4,3) \approx 1$
(6, 1, 0)		$L(6,1) \approx 0$
(2, 3, 0)		$L(2,3) \approx 0$
(5, 9, 1)		$L(5,9) \approx 1$
(4, 8, 1)		$L(4,8) \approx 1$
(1, 2, 1)		$L(1,2) \approx 1$
(4, 4, 0)		$L(4,4) \approx 0$
(4, 1, 0)		$L(4,1) \approx 0$
(5, 5, 1)		$L(5,5) \approx 1$

If E is minimized,
L will output like this!

Recap: Gradient Descent Method

- Gradient Descent Method



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

Neural Network Training

- **Linear regression**
  - Define an error function, E
  - Find **w** to minimize E

$$E = \sum_{(\mathbf{x},t)Data} MSE(t, y)$$

target value / output of model

where

$$MSE(t, y) = (t - y)^2$$

$$y = LM(\mathbf{x}; \mathbf{w})$$

Linear Model

$$L(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots$$

$w_i x_i$

$$\frac{\partial E}{\partial w_i} = \sum_{(\mathbf{x},t)Data} \frac{\partial MSE}{\partial y} \frac{\partial y}{\partial LM} \frac{\partial LM}{\partial w_i}$$

$= x_i$

$= -2(t-y) \cdot 1$

$\Rightarrow \Sigma -2(t_i - y_i) x_i$

- **NN for Regression**
  - Define an error function, E
  - Find **w** to minimize E

$$E = \sum_{(\mathbf{x},t)Data} MSE(t, y)$$

where

$$MSE(t, y) = (t - y)^2$$

$$y = NN(\mathbf{x}; \mathbf{w})$$

$$NN(\mathbf{x}; \mathbf{w}) = ??$$

$$\frac{\partial E}{\partial w_i} = \sum_{(\mathbf{x},t)Data} \frac{\partial MSE}{\partial y} \frac{\partial y}{\partial NN} \frac{\partial NN}{\partial w_i}$$

Neural Network Training

- **Logistic Regression**
  - Define an error function, E
  - Find **w** to minimize E

$$E = \sum_{(\mathbf{x},t)Data} CE(t,y)$$

where

$$CE(t,y) = t \cdot \log y + (1-t) \cdot \log(1-y)$$

$$y = LR(\mathbf{x};\mathbf{w})$$

$$LR(\mathbf{x};\mathbf{w}) = \frac{1}{1+e^{-f(\mathbf{x};\mathbf{w})}}$$

$$f(\mathbf{x};\mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots$$

$$\frac{\partial E}{\partial w_i} = \sum_{(\mathbf{x},t)Data} \frac{\partial CE}{\partial y}\frac{\partial y}{\partial LR}\frac{\partial LR}{\partial f}\frac{\partial f}{\partial w_i}$$

- **NN for Classification**
  - Define an error function, E
  - Find **w** to minimize E

$$E = \sum_{(\mathbf{x},t)Data} CE(t,y)$$

where

$$CE(t,y) = t \cdot \log y + (1-t) \cdot \log(1-y)$$

$$y = NN(\mathbf{x};\mathbf{w})$$

$$NN(\mathbf{x};\mathbf{w}) = ??$$

$$\frac{\partial E}{\partial w_i} = \sum_{(\mathbf{x},t)Data} \frac{\partial CE}{\partial y}\frac{\partial y}{\partial NN}\frac{\partial NN}{\partial w_i}$$

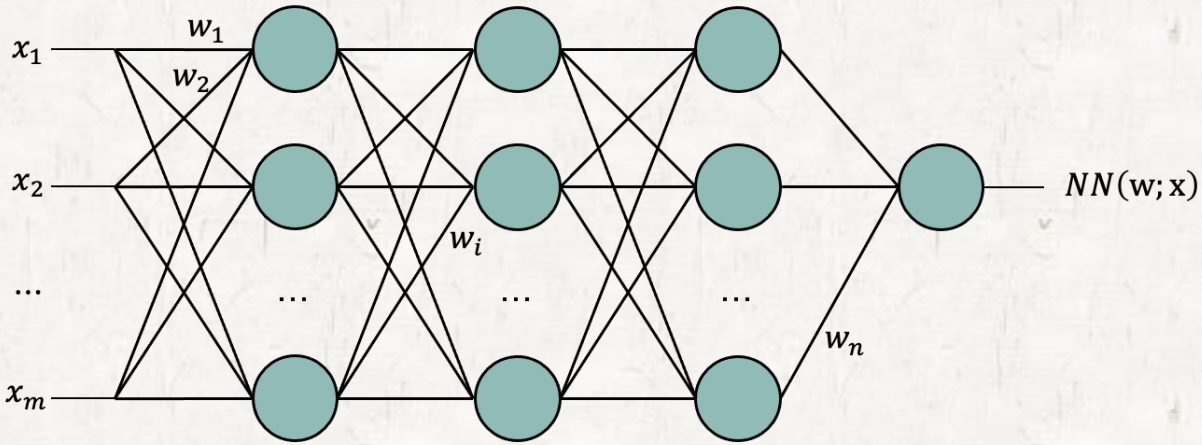
Neural Network Training

- How to differentiate NN

= Fomular식 (너무 복잡해서 식으로 적지않고 Figure로 보여줌)

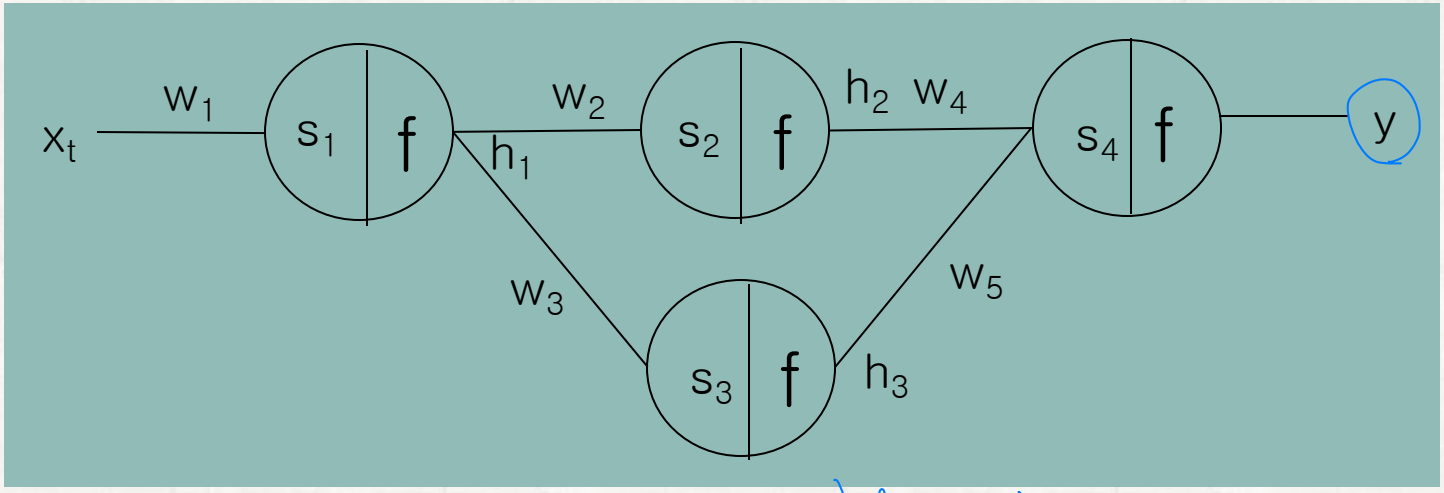
$$\frac{\partial NN}{\partial w_i}$$

계산은 가능하다!



NN Differentiation Example

- Example: How to differentiate NN
  - One training sample is given (x_t, y_t)

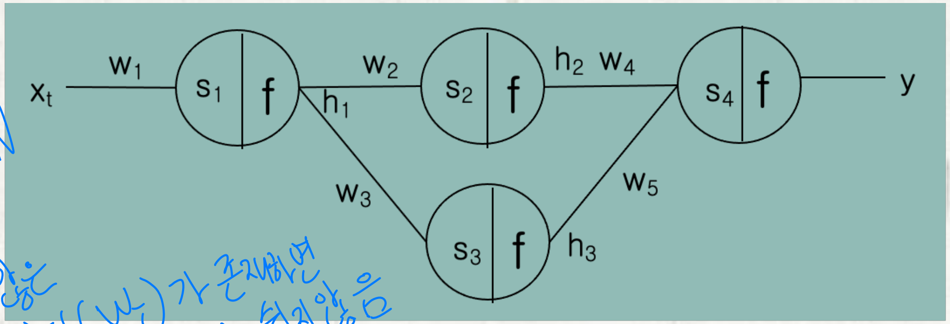


target value
NN output value

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

NN Differentiation Example

- Example: How to differentiate NN



Simple NN

But, 엄청 많은 파라미터(w_i)가 존재하면 변환하기 쉽지않음

$$E = \frac{1}{2}(y_t - y)^2 \qquad E = \frac{1}{2}(y_t - s_4)^2 \qquad E = \frac{1}{2}\left(y_t - (h_2 w_4 + h_3 w_5)\right)^2$$

$$E = \frac{1}{2}\left(y_t - (\mathrm{sigmoid}(s_2) w_4 + \mathrm{sigmoid}(s_3) w_5)\right)^2$$

$$E = \frac{1}{2}\left(y_t - (\mathrm{sigmoid}(h_1 w_2) w_4 + \mathrm{sigmoid}(h_1 w_3) w_5)\right)^2$$

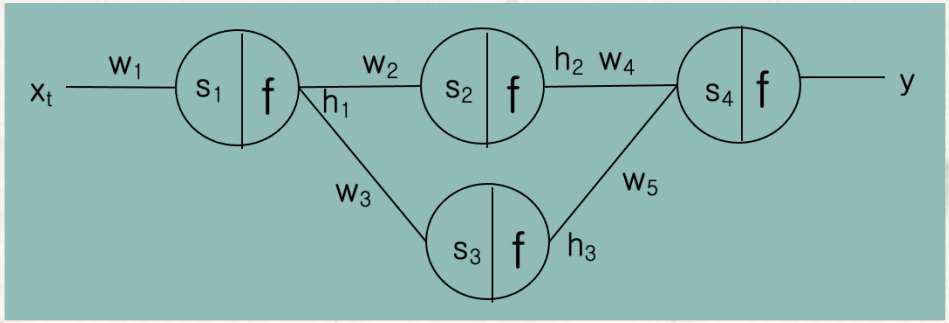
$$E = \frac{1}{2}\left(y_t - (\mathrm{sigmoid}(\mathrm{sigmoid}(s_1) w_2) w_4 + \mathrm{sigmoid}(\mathrm{sigmoid}(s_1) w_3) w_5)\right)^2$$

모든 w_i에 대해 미분가능

$$E = \frac{1}{2}\left(y_t - (\mathrm{sigmoid}(\mathrm{sigmoid}(x_t w_1) w_2) w_4 + \mathrm{sigmoid}(\mathrm{sigmoid}(x_t w_1) w_3) w_5)\right)^2$$

NN Differentiation Example

- How to differentiate NN



$$E = \frac{1}{2}\left(y_t - (\text{sigmoid}(\text{sigmoid}(x_t w_1)w_2)w_4 + \text{sigmoid}(\text{sigmoid}(x_t w_1)w_3)w_5)\right)^2$$

$$E = \frac{1}{2}\left(y_t - \left(\frac{1}{1+e^{-\frac{1}{1+e^{-x_1 w_1}}\cdot w_2}}\cdot w_4 + \frac{1}{1+e^{-\frac{1}{1+e^{-x_1 w_1}}\cdot w_3}}\cdot w_5\right)\right)^2$$

$$\frac{\partial E}{\partial w_i} = ?$$

Instead of converting to a global function,

Let's handle NN as composite functions

We need to remind Composite Function Differentiation

Recap: Composite Function Differentiation

합성 함수 미분

- Evaluate $\left.\frac{\partial y}{\partial w}\right|_{w=0}$ when $y = \frac{1}{x} \quad x = v^2 \quad v = e^w$

Variable Dependency Graph

$w \longrightarrow v \longrightarrow x \longrightarrow y$

체인룰

$$\frac{dy}{dw} = \frac{dy}{dx}\frac{dx}{dv}\frac{dv}{dw}$$

$$\frac{dy}{dw} = -\frac{1}{x^2} \cdot 2v \cdot e^w$$

$w = 0$

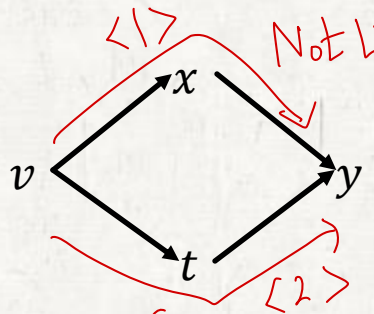
$v = 1$

$x = 1$

$$\left.\frac{\partial y}{\partial w}\right|_{w=0} = -\frac{1}{1^2} \cdot 2 \cdot 1 \cdot e = -e$$

Recap: Composite Function Differentiation

- Evaluate $\left.\frac{\partial y}{\partial v}\right|_{v=0}$ when $y = \frac{1}{(x+t)}$ $x = v^2$ $t = e^v$



가능한 경로만큼을 항을 나눠서

$$\frac{\partial y}{\partial v} = \frac{\partial y}{\partial x}\frac{\partial x}{\partial v} + \frac{\partial y}{\partial t}\frac{\partial t}{\partial v}$$

<방법>
① draw function of graph → variable dependency
② differentiate along each path
③ Σ (summation)

$$\frac{dy}{dv} = -\frac{1}{(x+t)^2}\cdot 2v - \frac{1}{(x+t)^2}\cdot e^v$$

$v = 0$
$t = 1$
$x = 0$

$$\left.\frac{\partial y}{\partial v}\right|_{v=0} = -\frac{1}{1^2}\cdot 2\cdot 0 - \frac{1}{1^2}\cdot e^0 = -1$$

Recap: Composite Function Differentiation

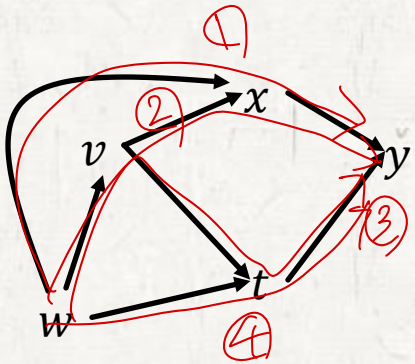
- Evaluate $\left.\frac{\partial y}{\partial w}\right|_{w=0}$ when $y = \frac{1}{x+t}$ $x = v^2 + w$ $t = \sin(w+v)\pi$ $v = e^w$

$w = 0$

$v = 1$

$t = 0$

$x = 1$



$$\frac{\partial y}{\partial w} = \frac{\partial y}{\partial x}\frac{\partial x}{\partial w} + \frac{\partial y}{\partial x}\frac{\partial x}{\partial v}\frac{\partial v}{\partial w} + \frac{\partial y}{\partial t}\frac{\partial t}{\partial v}\frac{\partial v}{\partial w} + \frac{\partial y}{\partial t}\frac{\partial t}{\partial w}$$

① w→x→y ② w→v→x→y ③ w→v→t→y ④ w→t→y

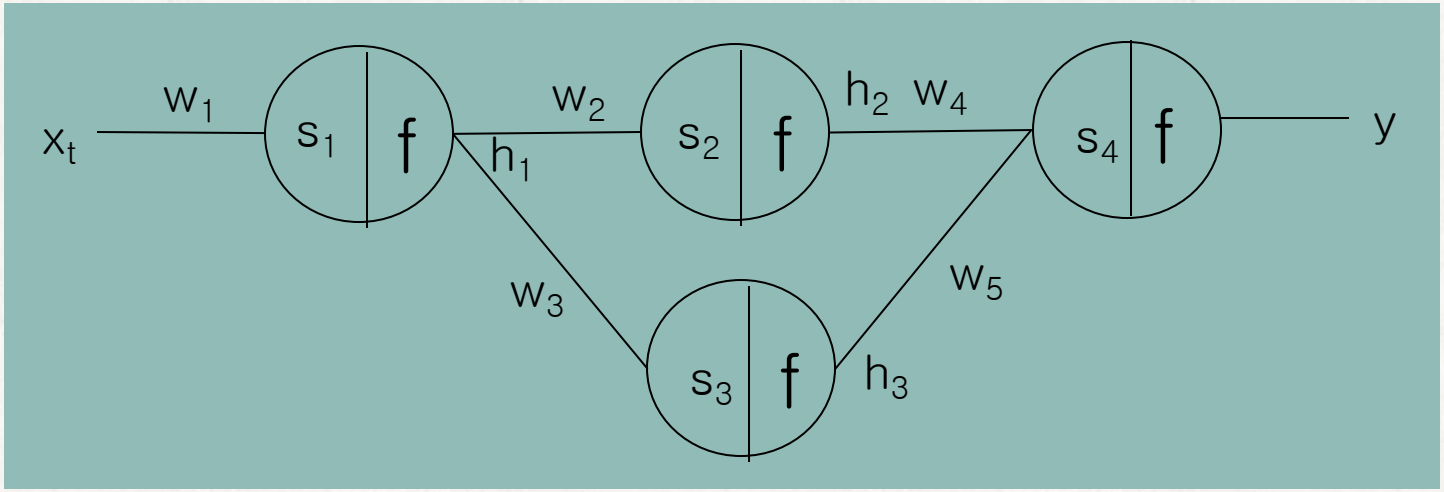
$$\frac{\partial y}{\partial w} = -\frac{1}{(x+t)^2}\cdot 1 - \frac{1}{(x+t)^2}\cdot 2v\cdot e^w - \frac{1}{(x+t)^2}\cdot \pi\cos(w+v)\pi\cdot e^w - \frac{1}{(x+t)^2}\cdot \pi\cos(w+v)\pi$$

$$\left.\frac{\partial y}{\partial w}\right|_{w=0} = -\frac{1}{(1+0)^2} - \frac{1}{(1+0)^2}\cdot 2\cdot 1\cdot e^0 - \frac{1}{(1+0)^2}\cdot \pi\cos(0+1)\pi\cdot e^0 - \frac{1}{(1+0)^2}\cdot \pi\cos(0+1)\pi$$

$$\left.\frac{\partial y}{\partial w}\right|_{w=0} = -1 - 2 - \pi\cdot(-1) - \pi\cdot(-1) = 2\pi - 3$$

NN Differentiation Example

- Example of NN with one training sample (x_t, y_t)



$s_1 = x_t \cdot w_1$

$h_1 = sigmoid(s_1)$

$s_2 = h_1 \cdot w_2$

$h_2 = sigmoid(s_2)$

$s_3 = h_1 \cdot w_3$

$h_3 = sigmoid(s_3)$

$s_4 = h_2 \cdot w_4 + h_3 \cdot w_5$

$y = s_4$

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

NN Differentiation Example

- Example of NN

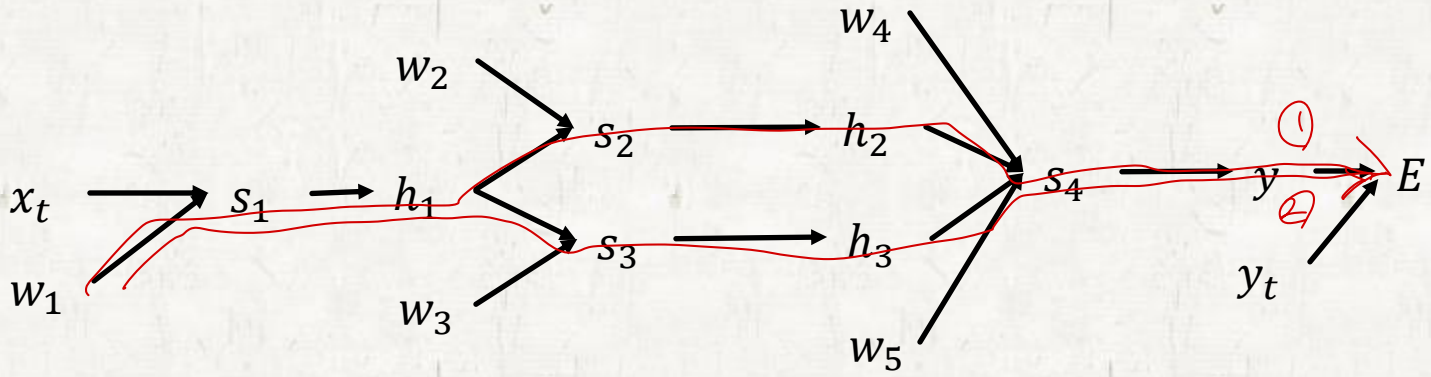
$$s_1 = x_t \cdot w_1$$
$$h_1 = sigmoid(s_1)$$

$$s_2 = h_1 \cdot w_2$$
$$h_2 = sigmoid(s_2)$$

$$s_3 = h_1 \cdot w_3$$
$$h_3 = sigmoid(s_3)$$

$$s_4 = h_2 \cdot w_4 + h_3 \cdot w_5$$
$$y = s_4$$

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



NN Differentiation Example

- Example of NN

$$s_1 = x_t \cdot w_1$$

$$h_1 = sigmoid(s_1)$$

$$s_2 = h_1 \cdot w_2$$

$$h_2 = sigmoid(s_2)$$

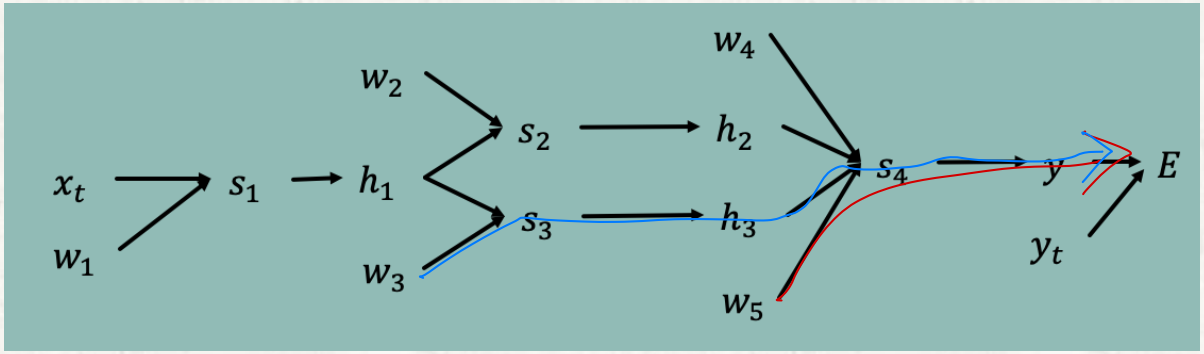
$$s_3 = h_1 \cdot w_3$$

$$h_3 = sigmoid(s_3)$$

$$s_4 = h_2 \cdot w_4 + h_3 \cdot w_5$$

$$y = s_4$$

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



$$\frac{\partial E}{\partial w_5} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_4}\frac{\partial s_4}{\partial w_5} = -(y_t - y) \cdot 1 \cdot h_3$$

$$\frac{\partial E}{\partial w_3} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_4}\frac{\partial s_4}{\partial h_3}\frac{\partial h_3}{\partial s_3}\frac{\partial s_3}{\partial w_3}$$

$$= -(y_t - y) \cdot 1 \cdot w_5 \cdot h_3 \cdot (1 - h_3) \cdot h_1$$

NN Differentiation Example

- Example of NN

$$s_1 = x_t \cdot w_1$$

$$h_1 = sigmoid(s_1)$$

$$s_2 = h_1 \cdot w_2$$

$$h_2 = sigmoid(s_2)$$

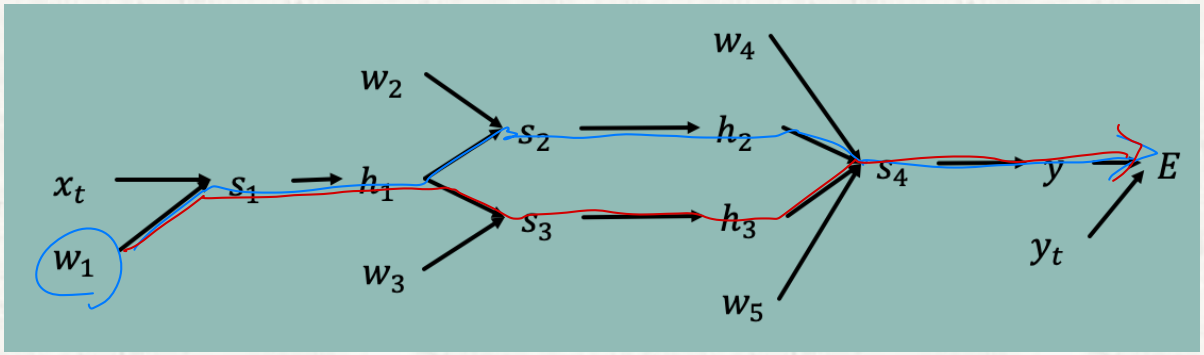
$$s_3 = h_1 \cdot w_3$$

$$h_3 = sigmoid(s_3)$$

$$s_4 = h_2 \cdot w_4 + h_3 \cdot w_5$$

$$y = s_4$$

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



$y = sigmoid(x)$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_4}\frac{\partial s_4}{\partial h_2}\frac{\partial h_2}{\partial s_2}\frac{\partial s_2}{\partial h_1}\frac{\partial h_1}{\partial s_1}\frac{\partial s_1}{\partial w_1} \quad \rightarrow \frac{dy}{dx} = y(1-y)$$

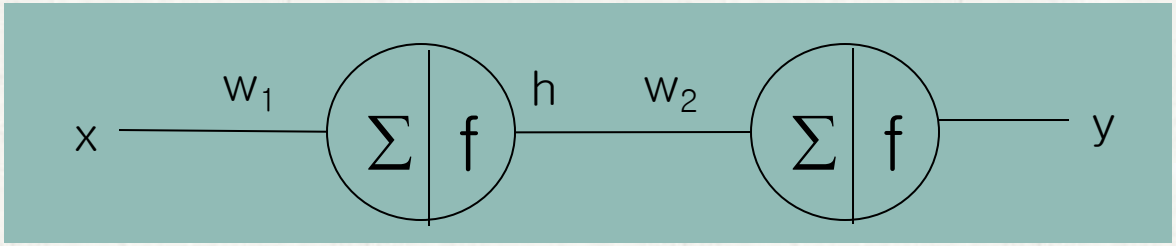
$$+ \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_4}\frac{\partial s_4}{\partial h_3}\frac{\partial h_3}{\partial s_3}\frac{\partial h_1}{\partial s_1}\frac{\partial s_1}{\partial w_1}$$

$$= -(y_t - y)w_4 h_2(1 - h_2)w_2 h_1(1 - h_1)x_t$$

$$+ -(y_t - y)w_5 h_3(1 - h_3)\ w_3 h_1(1 - h_1)x_t$$

Example

- Training of a Simple Neural Network
  - Let's use sigmoid as the activation function
  - Let's assume that there is one training data (x_t, y_t)



$$s_1 = x_t \cdot w_1$$

$$h = sigmoid(s_1)$$

$$s_2 = h \cdot w_2$$

$$y = s_2$$

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

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

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

Example

- Training of a Simple Neural Network

$$s_1 = x_t \cdot w_1$$

$$h = sigmoid(s_1)$$

$$s_2 = h \cdot w_2$$

$$y = s_2$$

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

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

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

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

$$\frac{\partial y}{\partial s_2} = 1$$

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

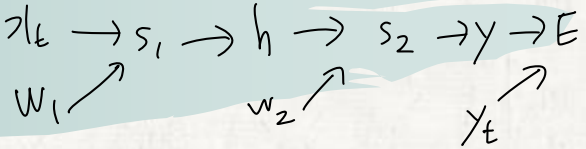
$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_2}\frac{\partial s_2}{\partial w_2}$$

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

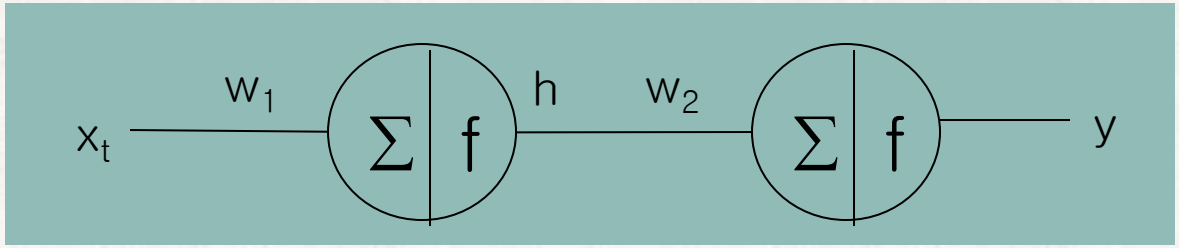
$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_2}\frac{\partial s_2}{\partial h}\frac{\partial h}{\partial s_1}\frac{\partial s_1}{\partial w_1}$$

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

Example



- Training of a Simple Neural Network



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

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

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

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

$$y = s_2 = 0.731$$

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

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

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

Example

- Training of a Simple Neural Network

$$\frac{\partial E}{\partial w_2} = -(y_t - y)h$$
$$= -(1 - 0.731) \cdot 0.731 = -0.197$$

$$\frac{\partial E}{\partial w_1} = -(y_t - y)w_2 h(1-h)x_t$$
$$= -(1 - 0.731) \cdot 1 \cdot 0.731 \cdot 0.269 \cdot 1 = -0.053$$

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

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

$$1.0197 = 1 + 0.1 \cdot 0.197$$

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

$$1.0053 = 1 + 0.1 \cdot 0.053$$

By GDM

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

Repeat

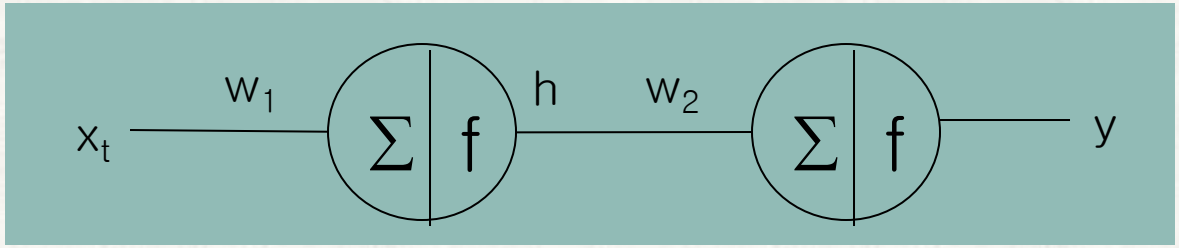
$$w_1^{t+1} = w_1^t - \eta \left.\frac{\partial E}{\partial w_0}\right|_{w_1=w_1^t, w_2=w_2^t}$$

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

Until stopping condition is satisfied

Example

- Training of a Simple Neural Network



$$(x_t, y_t) = (1,1), w_1 = 1.0053, w_2 = 1.0197, \eta = 0.1$$

$$s_1 = x_t \cdot w_1 = 1.0053$$

$$h = sigmoid(s_1) = 0.732$$

$$s_2 = h \cdot w_2 = 0.747$$

$$y = s_2 = 0.747$$

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

$$\frac{\partial E}{\partial w_2} = -(y_t - y)h$$

$$= -(1 - 0.747) \cdot 0.732$$

$$\frac{\partial E}{\partial w_1} = -(y_t - y)w_2 h(1-h)x_t$$

$$= -(1 - 0.747) \cdot 1.0197 \cdot 0.732 \cdot 0.268 \cdot 1$$

Example

w_1, w_2 (1, 1)
↓ update (1.0053, 1.0197)
↓ update (1.0104, 1.0384)

- Training of a Simple Neural Network

$$\frac{\partial E}{\partial w_2} = -(y_t - y)h$$
$$= -(1 - 0.747) \cdot 0.732 = -0.186$$

$$\frac{\partial E}{\partial w_1} = -(y_t - y)w_2 h(1-h)x_t$$
$$= -(1 - 0.747) \cdot 1.0197 \cdot 0.732 \cdot 0.268 \cdot 1 = -0.051$$

$(x_t, y_t) = (1,1)$
$w_1^1 = 1.0053$
$w_2^1 = 1.0197$
$\eta = 0.1$

$$w_2^2 = w_2^1 - \eta \frac{\partial E}{\partial w_2}$$
$$1.0384 = 1.0197 + 0.1 \cdot 0.186$$

$$w_1^2 = w_1^1 - \eta \frac{\partial E}{\partial w_1}$$
$$1.0104 = 1.0053 + 0.1 \cdot 0.051$$

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

Some Questions

- What if we have more than one training examples?
  For example, we have two examples: $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2)$

$$E = \sum_{(\mathbf{x},t)\in Data} (t - NN(\mathbf{x};\mathbf{w}))^2 = (t_1 - NN(\mathbf{x_1};\mathbf{w}))^2 + (t_2 - NN(\mathbf{x_2};\mathbf{w}))^2$$

MSE of (x_1, t_1) MSE of (x_2, t_2)

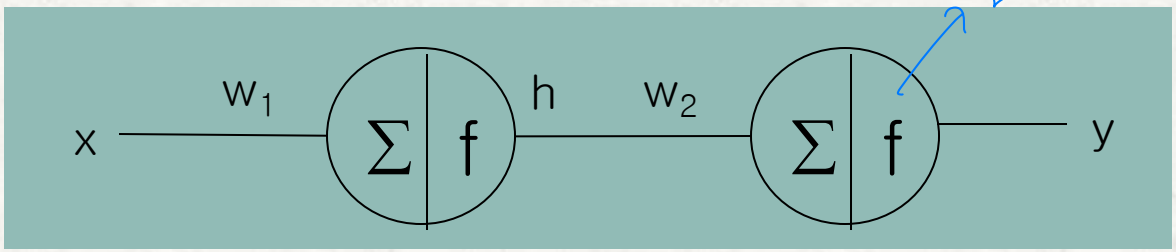
$$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_{(\mathbf{x},t)\in Data} (t - NN(\mathbf{x};\mathbf{w}))^2$$

$$= \frac{\partial}{\partial w_i} (t_1 - NN(\mathbf{x_1};\mathbf{w}))^2 + \frac{\partial}{\partial w_i} (t_2 - NN(\mathbf{x_2};\mathbf{w}))^2$$

$\frac{\partial E}{\partial w_i}$ 값 구하기

Some Questions

- What if we use the regression NN?



$$s_1 = x_t \cdot w_1$$

$$h = ReLU(s_1)$$

$$s_2 = h \cdot w_2$$

$$y = s_2$$

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

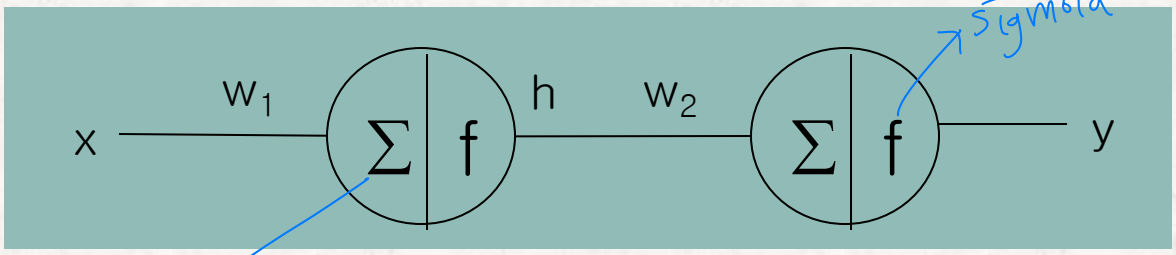
$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial s_2}\frac{\partial s_2}{\partial w_2}$$

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

$$\frac{\partial h}{\partial s_1} = \begin{cases} 1 & s_1 > 0 \\ 0 & s_1 \leq 0 \end{cases}$$

Some Questions

- What if we use the classification NN?



$$s_1 = x_t \cdot w_1$$

$$h = ReLU(s_1)$$

$$s_2 = h \cdot w_2$$

$$y = sigmoid(s_2)$$

$$E = -y_t \cdot \log y - (1 - y_t) \cdot \log(1 - y)$$

Cross Entropy

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

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

$$\frac{\partial E}{\partial y} = -\frac{y_t}{y} + \frac{1 - y_t}{1 - y}$$

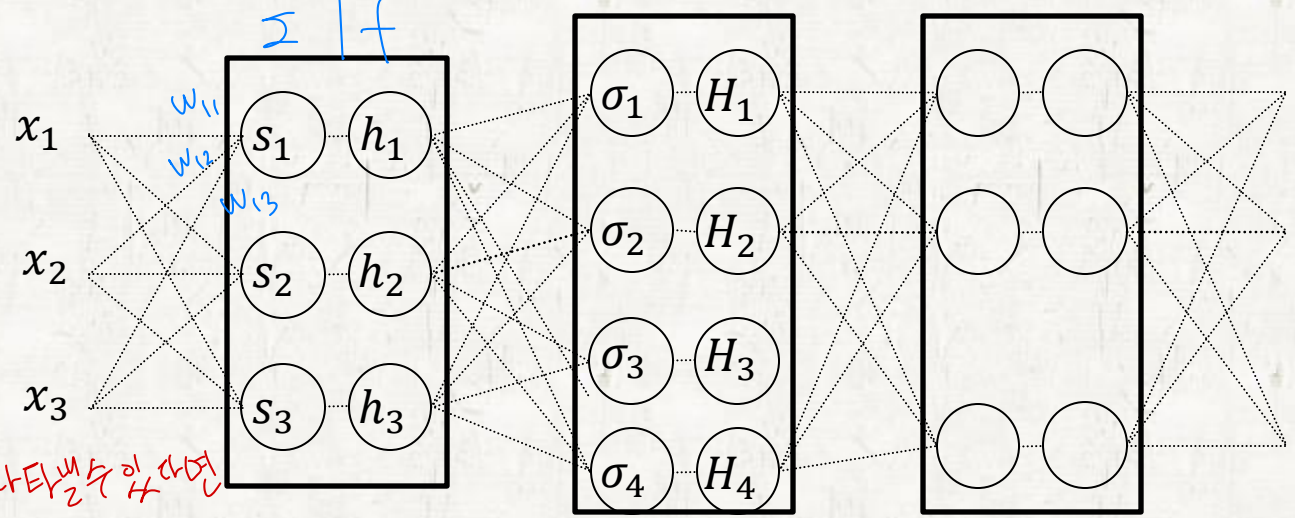
$$\frac{\partial y}{\partial s_2} = y(1 - y)$$

Other Considerations

- What is the role of “Activation Function”?
- Why “ReLU” in hidden layers? instead sigmoid

Other Considerations

- What is the role of "Activation Function"?

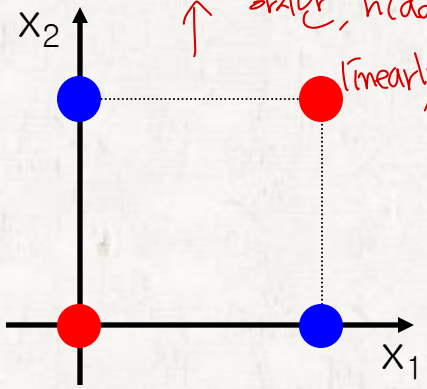


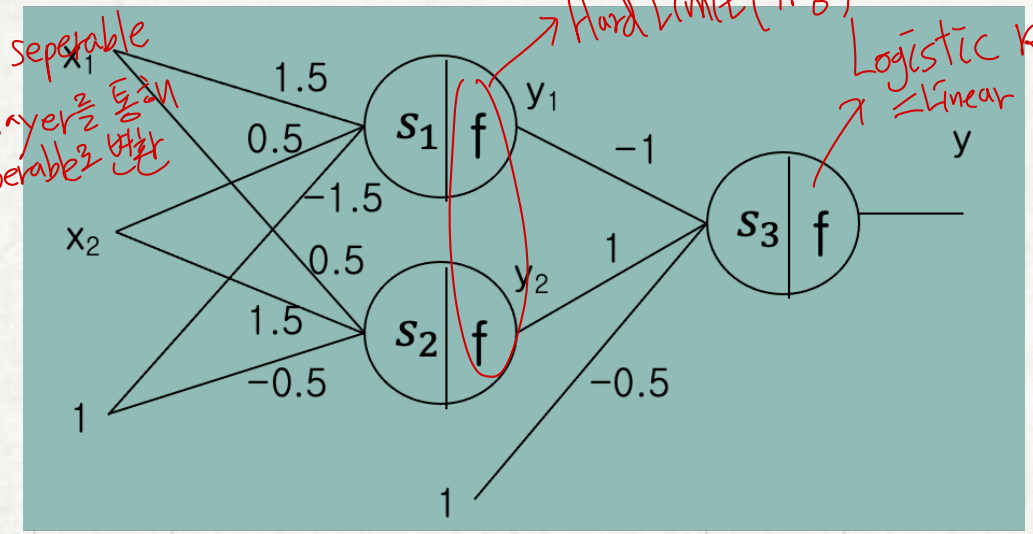
$$(x_1, x_2, x_3) \rightarrow (s_1, s_2, s_3) \rightarrow (h_1, h_2, h_3) \rightarrow (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \rightarrow (H_1, H_2, H_3, H_4)$$

$$(s_1, s_2, s_3) = (x_1, x_2, x_3)\begin{pmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \\ w_{13} & w_{23} & w_{33} \end{pmatrix} \qquad (h_1, h_2, h_3) = \begin{pmatrix} ReLU(s_1) \\ ReLU(s_2) \\ ReLU(s_3) \end{pmatrix}^T$$

Other Considerations

- What is the role of "Activation Function"?





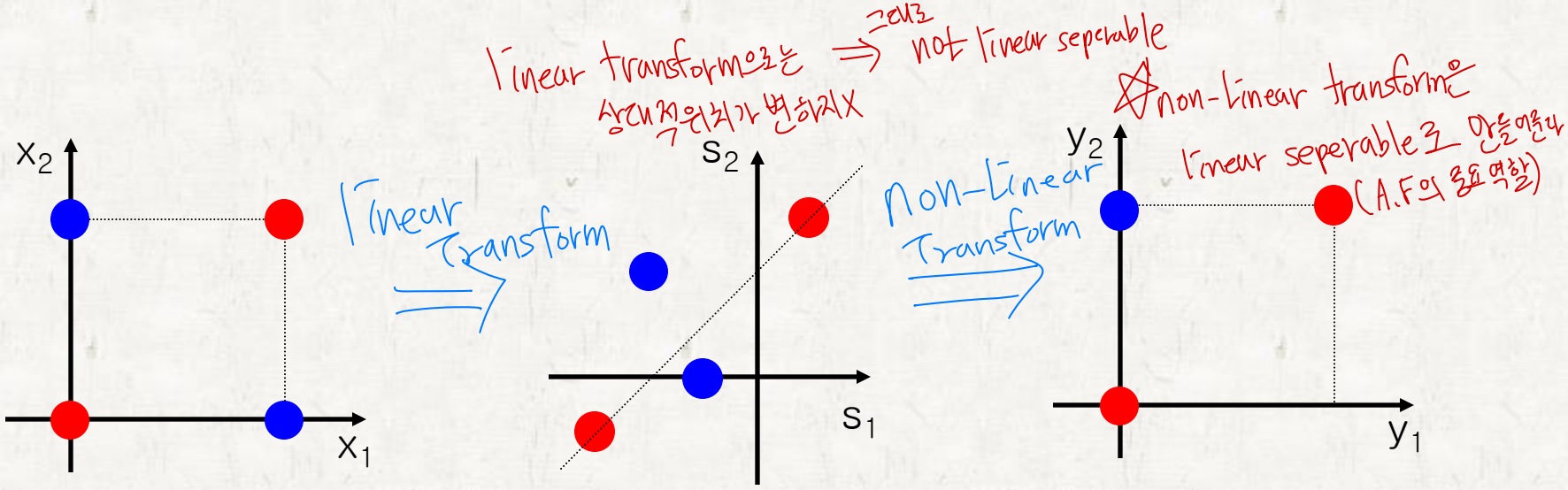
x_1	x_2
0	0
0	1
1	0
1	1

s_1	s_2
−1.5	−0.5
−1.0	1.0
−0.5	0.0
0.5	1.5

y_1	y_2
0	0
0	1
0	1
1	1

Other Considerations

- What is the role of "Activation Function"?



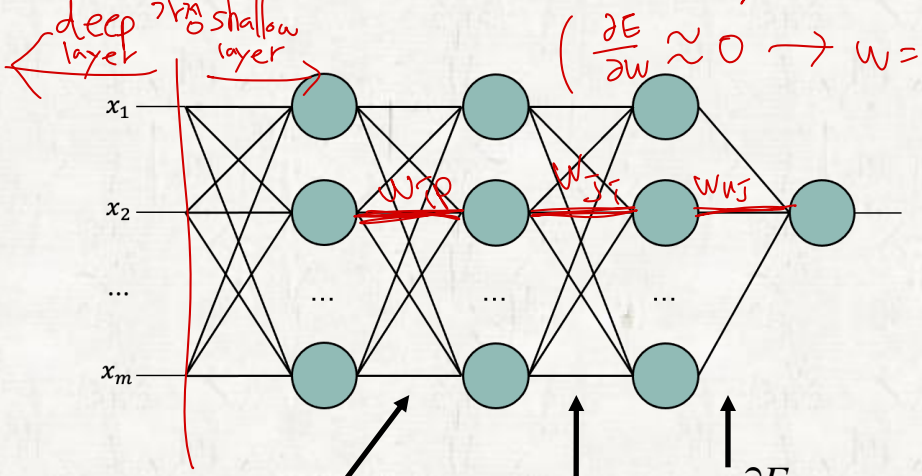
x_1	x_2
0	0
0	1
1	0
1	1

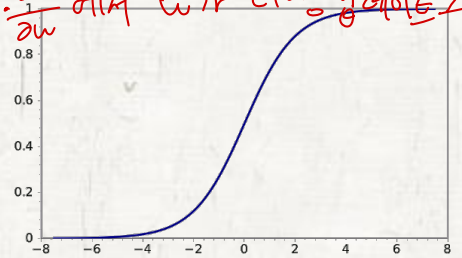
s_1	s_2
−1.5	−0.5
−1.0	1.0
−0.5	0.0
0.5	1.5

y_1	y_2
0	0
0	1
0	1
1	1

Other Considerations

- Why "ReLU" in Hidden Layers? not Sigmoid?
  - Vanishing Gradient → deep layer에 있는 w값에 대한 도함수가 0에 근접하게 되어 더이상 update할 수 없다





Sigmoid derivative

$$\frac{\partial \text{Sigmoid}}{\partial w} \leq \frac{1}{4}$$

$$\frac{\partial E_n}{\partial w_{kj}} = -(t_{nk} - o_{nk}) o_{nk}(1-o_{nk}) h_{nj}$$

$$\frac{\partial E_n}{\partial w_{ji}} = -h_{nj}(1-h_{nj}) x_{ni} \sum_{k=1}^{m} w_{kj}(t_{nk} - o_{nk}) o_{nk}(1-o_{nk})$$

$$\frac{\partial E}{\partial w_{ip}} = \left(\sum_{j=1}^{J} \left(\sum_{k=1}^{K} -(t_n - o_{nk}) o_{nk}(1-o_{nk}) w_{kj} \right) h_{nj}(1-h_{nj}) w_{ji} \right) h_{ni}(1-h_{ni}) h_{np}$$

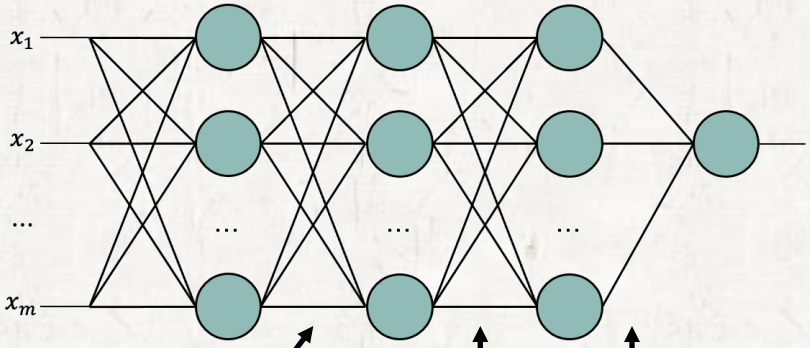
$\leq \frac{1}{64}$

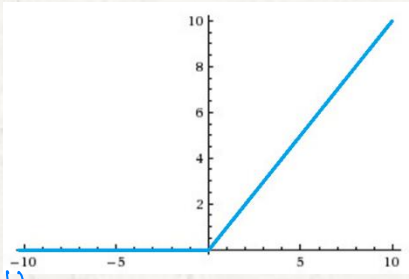
∴ deep layer에 있는 w에 대한 도함수가 점점 작아지게 됨

deep layer에 있는 w값에 대한 도함수는 점점 작아지고 학습이 느려진다

Other Considerations

- Why "ReLU" in Hidden Layers?
  - Vanishing Gradient





ReLU(x)

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

1 or 0 중 값이 대입되고

$$\frac{\partial E_n}{\partial w_{kj}} = -(t_{nk} - o_{nk}) \boxed{1} h_{nj}$$

$$\frac{\partial E_n}{\partial w_{ji}} = -\boxed{1} x_{ni} \sum_{k=1}^{m} w_{kj} (t_{nk} - o_{nk}) \boxed{1}$$

$$\frac{\partial E}{\partial w_{ip}} = \left(\sum_{j=1}^{J} \left(\sum_{k=1}^{K} -(t_n - o_{nk}) \boxed{1} w_{kj} \right) \boxed{1} w_{ji} \right) \boxed{1} h_{np}$$

↳ 잘 안 없어지지 않는다 ⇒ deeplayer의 w값의 조합은 0에 가까워지지 않는다 ⇒ trainable!!