Deep Learning Using TensorFlow

Dr. Ash Pahwa

Lesson 6:

Gradient Descent & Backpropagation Algorithms

Lesson 6.2: Backpropagation Algorithm

Copyright 2019 - Dr. Ash Pahwa

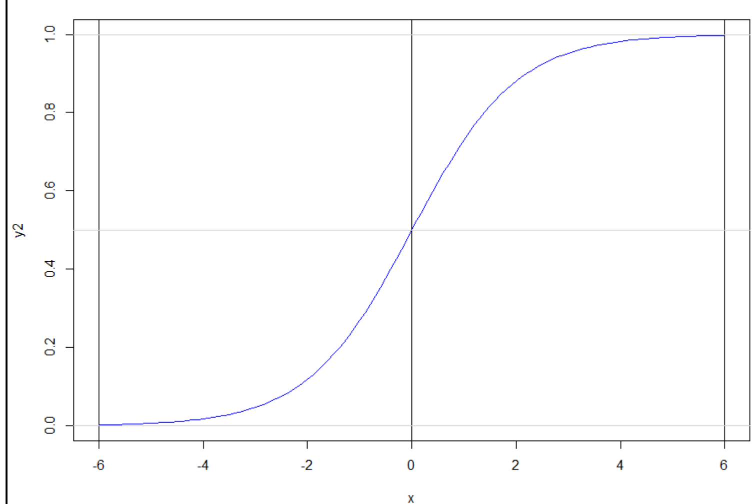
Outline

- Basic Calculus Derivatives
  - Derivative of Sigmoid Function
  - Chain Rule
  - Steepest Descent
- Backpropagation Algorithm
- Example-1
  - Forward Propagation
  - Backpropagation

Activation Functions
Sigmoid Function

- Activation Functions
  - Unit Step
  - Sigmoid
  - ReLU

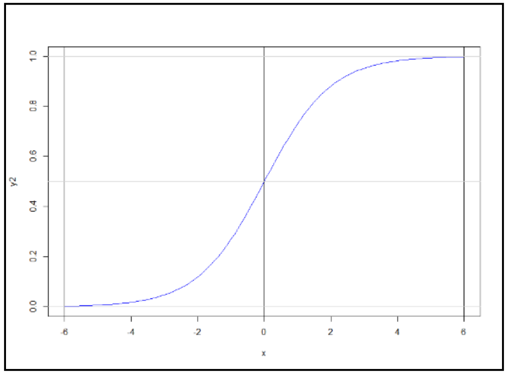
$$f(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$$




```
R Code for Sigmoid Function

> x <- seq(-6,6,0.1)
> y1 <- exp(x)/(1+exp(x))
> y2 <- 1/(1+exp(-x))
> plot(x,y2,type='l',col="blue")
> abline(v=seq(-6,6,6),col="black",lty=1)
> abline(h=seq(0,1,0.5),col="lightgrey",lty=1)
```

Derivative of Sigmoid Function

$$\boxed{f(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}}$$

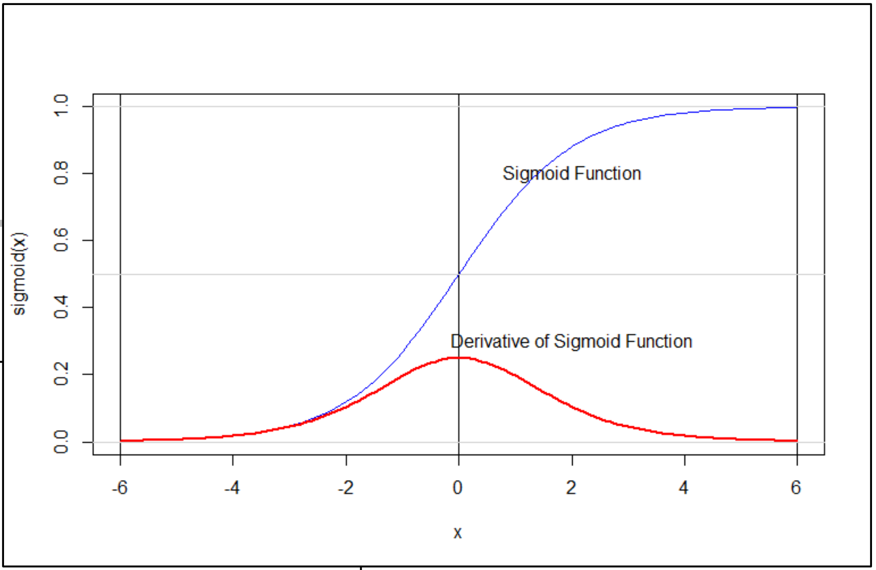


- $\sigma(x) = \frac{1}{1+e^{-x}}$
- $\frac{d\,\sigma(x)}{dx} = \frac{d}{dx}\frac{1}{(1+e^{-x})} = \frac{(1+e^{-x})\frac{d}{dx}(1) - 1\frac{d}{dx}(1+e^{-x})}{(1+e^{-x})^2}$ *Apply Quotient Rule*
- $\frac{d\,\sigma(x)}{dx} = \frac{0-(-1)e^{-x}}{(1+e^{-x})^2} = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{e^{-x}+1-1}{(1+e^{-x})^2} = \frac{1+e^{-x}}{(1+e^{-x})^2} - \frac{1}{(1+e^{-x})^2}$
- $\frac{d\,\sigma(x)}{dx} = \frac{1}{1+e^{-x}} - \left(\frac{1}{1+e^{-x}}\right)^2$
- $\frac{d\,\sigma(x)}{dx} = \sigma(x) - (\sigma(x))^2$
- $\sigma'(x) = \sigma(x)(1 - \sigma(x))$

Plot of Sigmoid Function and its Derivative

$$\sigma(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$$

$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$




```
> ########################################
> # Plot of the derivative of Sigmoid function
> #########################################
>
> sigmoid = function (x) { 1/(1+exp(-x)) }
>
> der.sigmoid = function (x) { sigmoid(x) * ( 1 - sigmoid(x)) }
>
> x <- seq(-6,6,0.1)
>
> plot(x,sigmoid(x), type='l',col="blue")
>
> abline(v=seq(-6,6,6),col="black",lty=1)
> abline(h=seq(0,1,0.5),col="lightgrey",lty=1)
>
> lines(x,der.sigmoid(x), type='l',col="red",lwd=2)
>
> text(2,0.8,"Sigmoid Function")
> text(2,0.3,"Derivative of Sigmoid Function")
```

Chain Rule

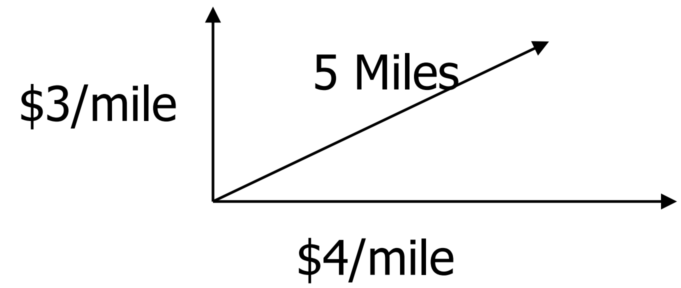
- $$\frac{dz}{dx} = \frac{dy}{dx} * \frac{dz}{dy}$$

B	C	D	E	F
	Chain Rule			
	x	y = f(x)	z = g(y)	
		2*x	3*y	
	2	4	12	
	3	6	18	
	4	8	24	
	5	10	30	
	6	12	36	

$$\frac{dy}{dx} = 2 \qquad \frac{dz}{dy} = 3$$

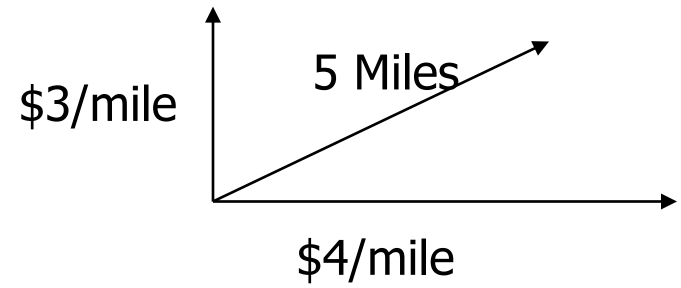
$$\frac{dz}{dx} = \frac{dy}{dx} * \frac{dz}{dy} = 6$$

Steepest Descent



- Suppose you are paid
  - $4/mile to drive East
  - $3/mile to drive North
  - At most you are can drive 5 miles
  - Which direction you should drive?

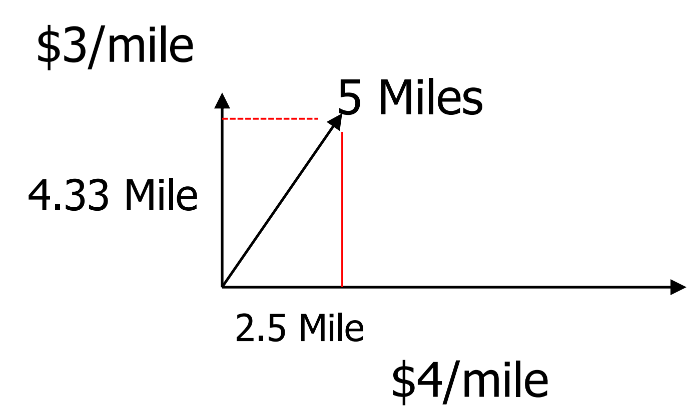
Steepest Descent



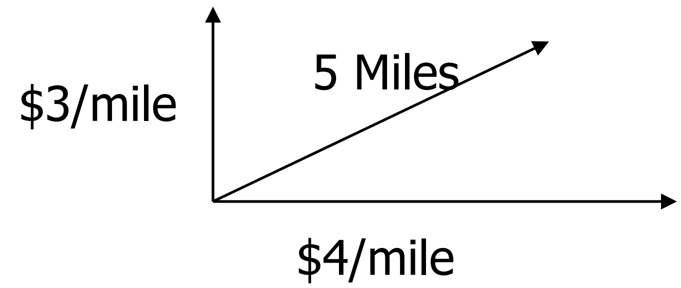
- Suppose you are paid
  - $4/mile to drive East
  - $3/mile to drive North
  - At most you are can drive 5 miles
  - Which direction you should drive?

- Drive 5 miles East: Revenues = 5 miles*$4/mile = $20
- Drive 5 Miles North: Revenues = 5 miles*$3/mile = $15

- Drive 2.5 Miles East + 4.33 Miles North
- $\sqrt{2.5^2 + 4.33^2} = 5\ Mile$
- Revenues: 4.33*$3/mile + 2.5*$4/mile = $23



Steepest Descent



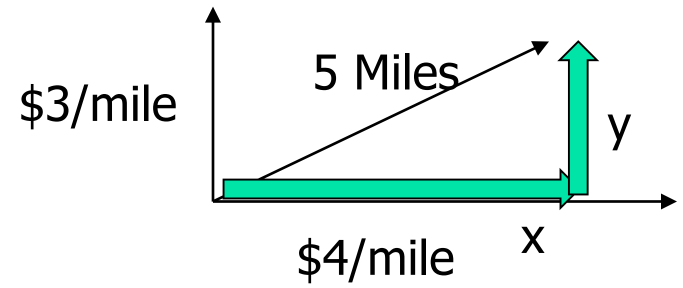
- Suppose you are paid
  - $4/mile to drive East
  - $3/mile to drive North
  - At most you are can drive 5 miles
  - Which direction you should drive?

- Drive 5 miles East: Revenues = 5 miles*$4/mile = $20
- Drive 5 Miles North: Revenues = 5 miles*$3/mile = $15

- For maximum revenues: Drive 4 miles East + 3 miles North
  - Revenues = 4 Miles*$4/mile = $16
  - Revenues = 3 Miles*$3/mile = $9
  - Total Revenues = $16 + $9 = $25

Copyright 2019 - Dr. Ash Pahwa

Gradient



- $Revenues = \frac{\$4}{mile}x + \frac{\$3}{mile}y$
- $\frac{\partial r}{\partial x} = 4$
- $\frac{\partial r}{\partial y} = 3$
- $\nabla r = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$
- Gradient is a vector that points to the steepest increase or decrease

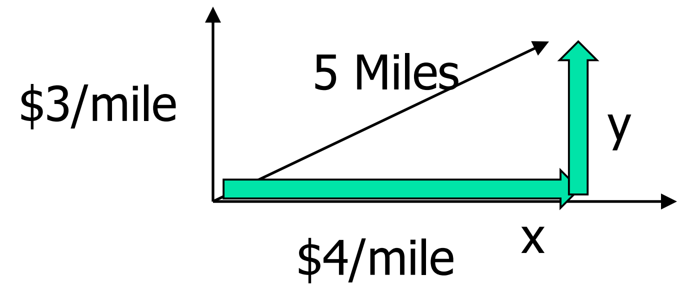
- For maximum revenues: Drive 4 miles East + 3 miles North
  - Revenues = 4 Miles*$4/mile = $16
  - Revenues = 3 Miles*$3/mile = $9
  - Total Revenues = $16 + $9 = $25

Final Result

You go in the direction
in proportion to how profitable that
direction is.

Copyright 2019 - Dr. Ash Pahwa

Another way to solve this problem



- $x^2 + y^2 = 5^2 = 25$
- $y = \sqrt{25 - x^2}$
- $Revenues = \frac{\$4}{mile} x + \frac{\$3}{mile} y$
- $Revenues = 4x + 3 * \sqrt{25 - x^2}$
- $\frac{dr}{dx} = 4 + \frac{3}{2} * \frac{(-2x)}{\sqrt{25 - x^2}} = 0$
- $4 = \frac{3x}{\sqrt{25 - x^2}}$
- $16 = \frac{9x^2}{25 - x^2}$
- $400 - 16x^2 = 9x^2$
- $25x^2 = 400$
- $x^2 = 16$
- $x = 4$
- $y = 3$

$$y = x^n; \qquad \frac{dy}{dx} = nx^{n-1}$$

- For maximum revenues: Drive 4 miles East + 3 miles North
  - Revenues = 4 Miles*$4/mile = $16
  - Revenues = 3 Miles*$3/mile = $9
  - Total Revenues = $16 + $9 = $25

Final Result

You go in the direction
in proportion to how profitable that
direction is.

Backpropagation Algorithm

Backpropagation Algorithm: Procedure of Computing Weights

- Conceptually the back propagation algorithm is very simple

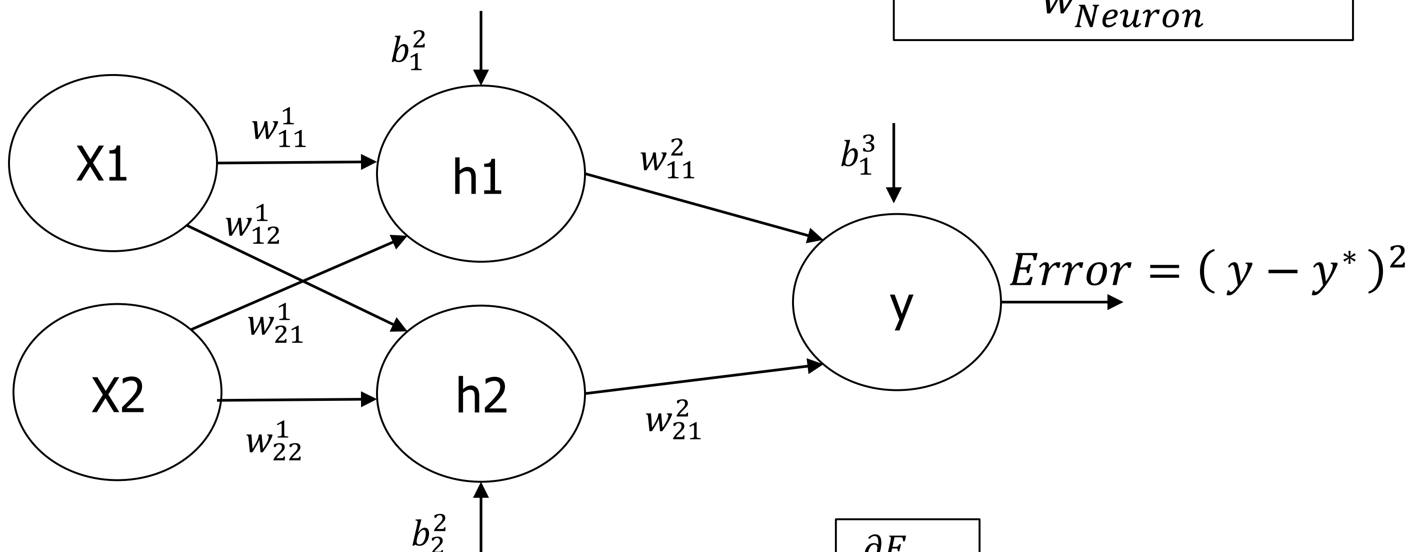
- Algorithm
  - Assign random values to all the weights of the NN
  - Take the first observed data
    - Forward Propagation: Compute Output
    - Compute error = (Computed Output – Observed Output)^2
    - Backpropagation: adjust weights to reduce error
    - Repeat forward, backward propagation, till error is minimized
  - Repeat the previous step for the next sample till all samples are processed
  - The final weights of the NN will be used for prediction

Copyright 2019 - Dr. Ash Pahwa

Compute Partial Derivative

$$w^{Layer}_{Neuron}$$

$$\frac{\partial E}{\partial w^1_{11}} \quad \frac{\partial E}{\partial w^1_{12}} \quad \frac{\partial E}{\partial w^1_{21}} \quad \frac{\partial E}{\partial w^1_{22}}$$



$$\frac{\partial E}{\partial b^2_1} \quad \frac{\partial E}{\partial b^2_2}$$

$$\frac{\partial E}{\partial w^2_{11}} \quad \frac{\partial E}{\partial w^2_{21}}$$

$$\frac{\partial E}{\partial b^3_1}$$

Backpropagation Algorithm

- After computing partial derivatives of error with respect to all weights and biases
  - Adjust the weight and the bias with the steepest change in the output

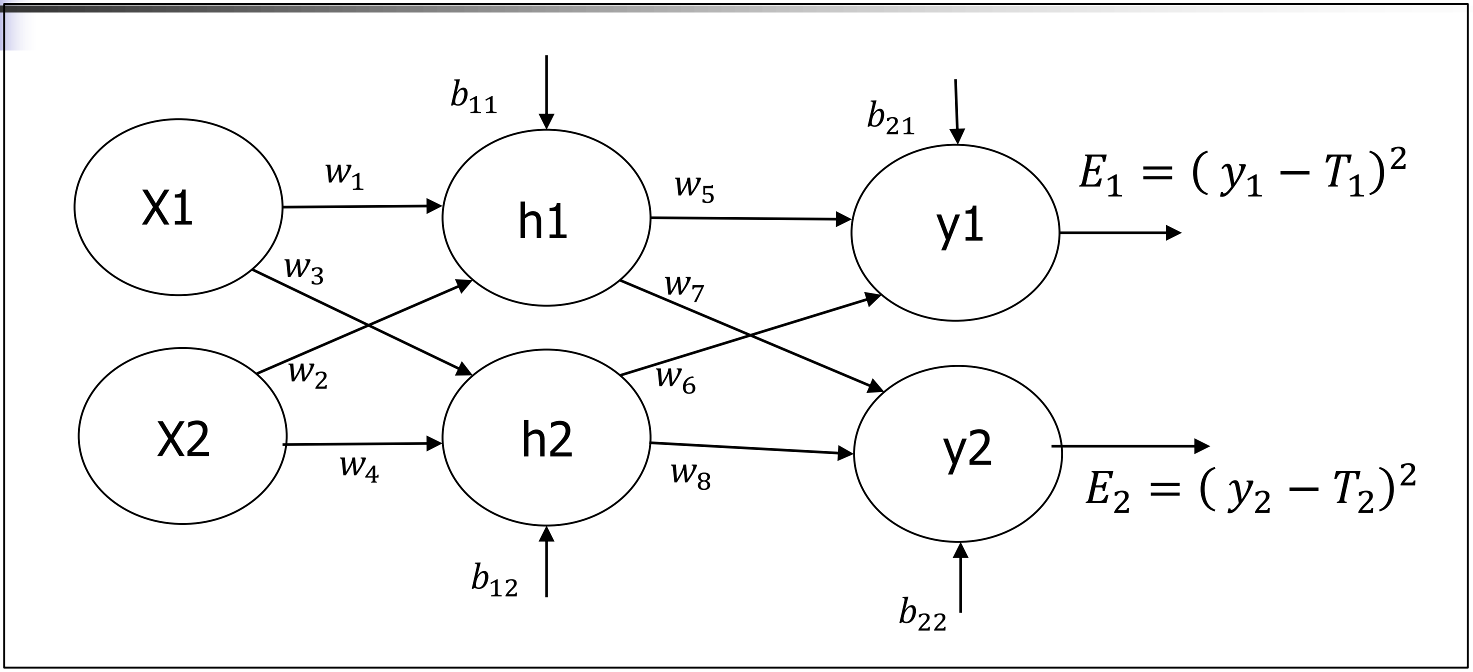
$$\boxed{\begin{matrix}\frac{\partial E}{\partial w_{11}^1} \\ \frac{\partial E}{\partial w_{12}^1} \\ \frac{\partial E}{\partial w_{21}^1} \\ \frac{\partial E}{\partial w_{22}^1}\end{matrix}} \quad \boxed{\begin{matrix}\frac{\partial E}{\partial b_1^2} \\ \frac{\partial E}{\partial b_2^2}\end{matrix}} \quad \boxed{\begin{matrix}\frac{\partial E}{\partial w_{11}^2} \\ \frac{\partial E}{\partial w_{21}^2}\end{matrix}} \quad \boxed{\frac{\partial E}{\partial b_1^3}}$$

Copyright 2019 - Dr. Ash Pahwa

Example-1

Example

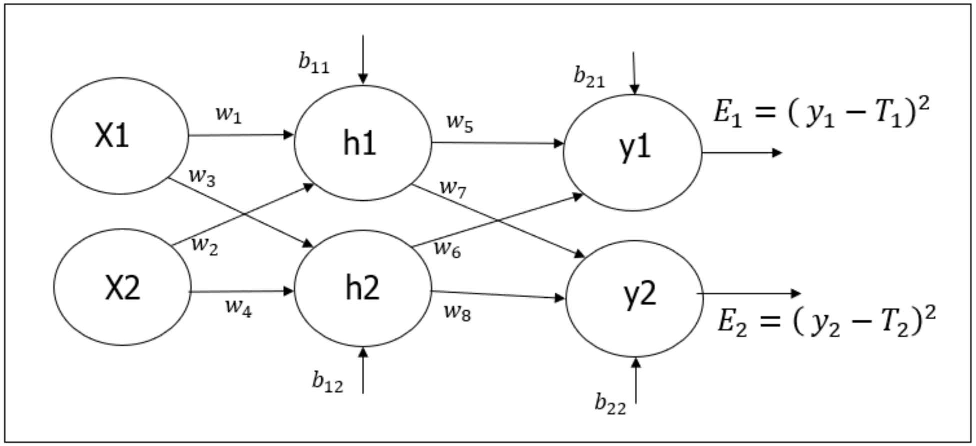
Learning Rate (0-1) = 0.5



Training Data

#	x1	x2	T1	T2
1	0.05	0.10	0.01	0.99
2				

Forward Propagation-1



Activation Function

$$sigmoid(x) = \sigma(x) = \frac{1}{1+e^{-x}}$$

$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$

Initial Values

$w_1 = 0.15,\quad b_{11} = b_{12} = 0.35$

$w_2 = 0.20$

$w_3 = 0.25$

$w_4 = 0.30$

$w_5 = 0.40,\quad b_{21} = b_{22} = 0.60$

$w_6 = 0.45$

$w_7 = 0.50$

$w_8 = 0.55$

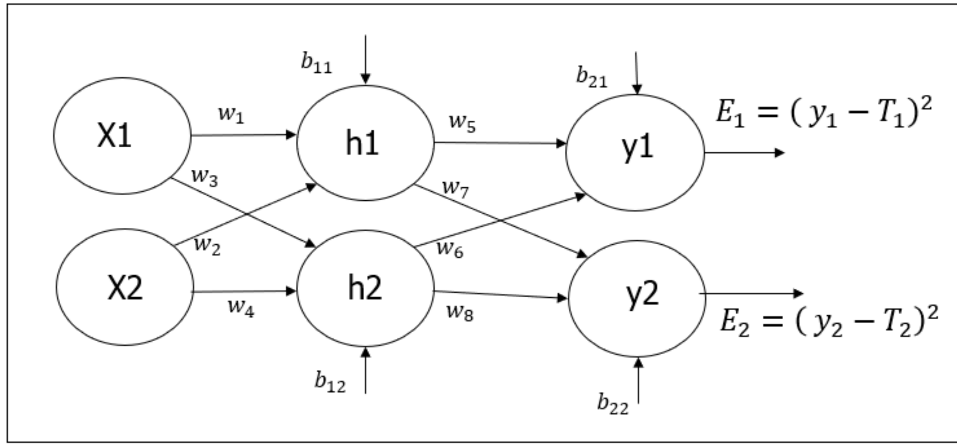
#	x1	x2	T1	T2
1	0.05	0.10	0.01	0.99

- $h1 = x_1 * w_1 + x_2 * w_2 + b_{11}$
- $outH1 = \sigma(h1)$
- $h2 = x_1 * w_3 + x_2 * w_4 + b_{12}$
- $outH2 = \sigma(h2)$

Forward Propagation-1: R Code

```
> ########################
> # Input data
> x1 = 0.05
> x2 = 0.10
> # Output target data
> T1 = 0.01
> T2 = 0.99
> #########################
> # Initial weight and biases
> w1 = 0.15
> w2 = 0.20
> w3 = 0.25
> w4 = 0.30
> ###########
> b11 = b12 = 0.35
> ###########
> w5 = 0.40
> w6 = 0.45
> w7 = 0.50
> w8 = 0.55
> ###########
> b21 = b22 = 0.60
>
```

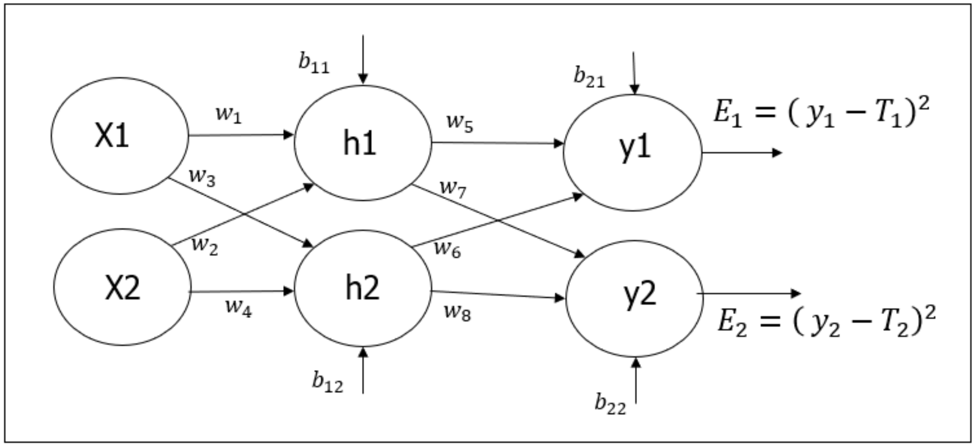
```
##################################
> # Activation Function - sigmoid
> #
> sigmoid <- function(x) {
+   return(1/(1+exp(-x)))
+ }
```




```
> ##########################
> # Forward propagation
> #
> (h1 = x1*w1 + x2*w2 + b11)
[1] 0.3775
> (outH1 = sigmoid(h1))
[1] 0.59327
> (h2 = x1*w3 + x2*w4 + b12)
[1] 0.3925
> (outH2 = sigmoid(h2))
[1] 0.5968844
>
```

- $h1 = x_1 * w_1 + x_2 * w_2 + b_{11}$
- $outH1 = \sigma(h1) = 0.5932$
- $h2 = x_1 * w_3 + x_2 * w_4 + b_{12}$
- $outH2 = \sigma(h2) = 0.5968$

Forward Propagation-2



Activation Function

$$sigmoid(x) = \sigma(x) = \frac{1}{1+e^{-x}}$$

$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$

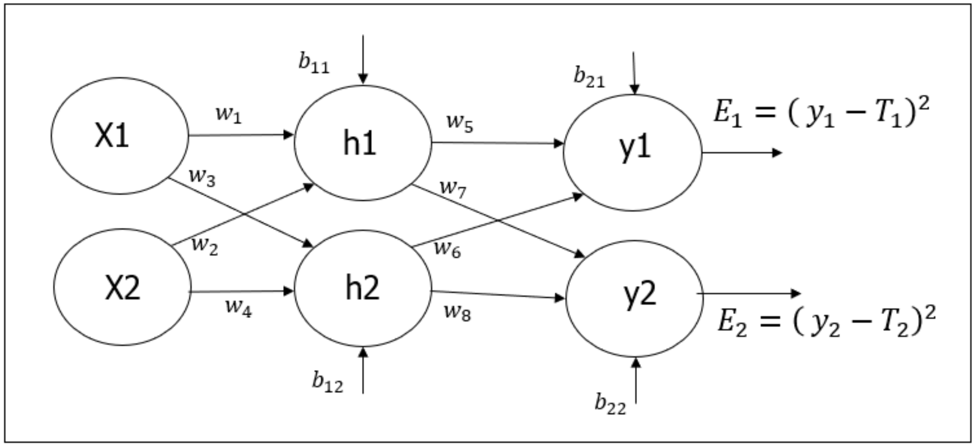
```
##################################
> # Activation Function - sigmoid
> #
> sigmoid <- function(x) {
+   return(1/(1+exp(-x)))
+ }
> #################################
> # Forward propagation
> #
> (h1 = x1*w1 + x2*w2 + b11)
[1] 0.3775
> (outH1 = sigmoid(h1))
[1] 0.59327
> (h2 = x1*w3 + x2*w4 + b12)
[1] 0.3925
> (outH2 = sigmoid(h2))
[1] 0.5968844
>
```

- $y1 = outH1 * w_5 + outH2 * w_6 + b_{21}$
- $outY1 = \sigma(y1)$

- $y2 = outH1 * w_7 + outH2 * w_8 + b_{22}$
- $outY2 = \sigma(y2)$

Copyright 2019 - Dr. Ash Pahwa

Forward Propagation-2: R Code



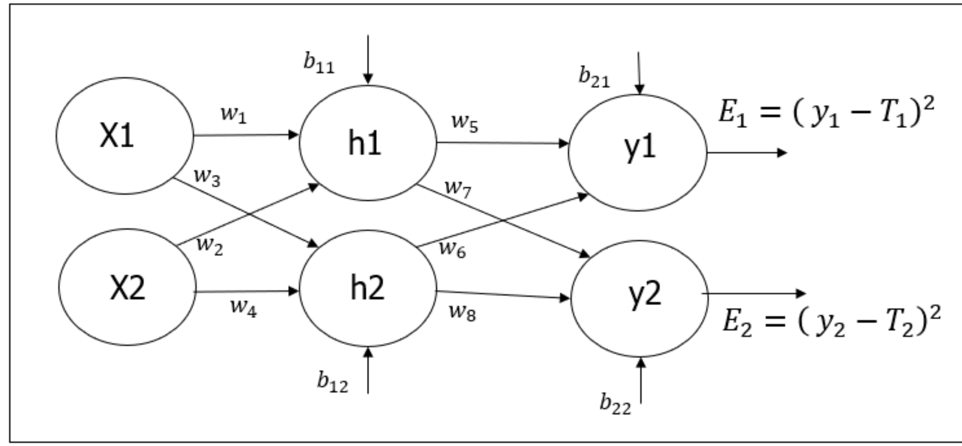

```
###################################
> # Activation Function - sigmoid
> #
> sigmoid <- function(x) {
+   return(1/(1+exp(-x)))
+ }
> ################################
> # Forward propagation
> #
> (h1 = x1*w1 + x2*w2 + b11)
[1] 0.3775
> (outH1 = sigmoid(h1))
[1] 0.59327
> (h2 = x1*w3 + x2*w4 + b12)
[1] 0.3925
> (outH2 = sigmoid(h2))
[1] 0.5968844
>
```

```
> ###################
> #
> (y1 = outH1*w5 + outH2*w6 + b21)
[1] 1.105906
> (outY1 = sigmoid(y1))
[1] 0.7513651
> (y2 = outH1*w7 + outH2*w8 + b22)
[1] 1.224921
> (outY2 = sigmoid(y2))
[1] 0.7729285
>
```

- $y1 = outH1 * w_5 + outH2 * w_6 + b_{21}$
- $outY1 = \sigma(y1) = 0.7513$

- $y2 = outH1 * w_7 + outH2 * w_8 + b_{22}$
- $outY2 = \sigma(y2) = 0.7729$

Error



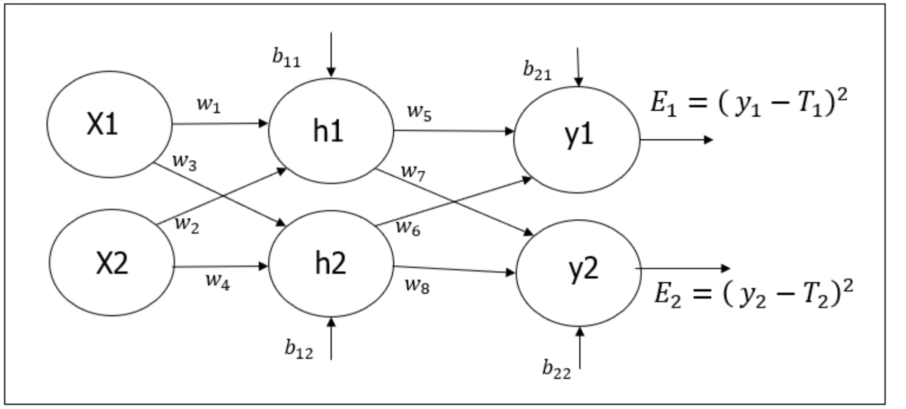

```
> ###################
> #
> (y1 = outH1*w5 + outH2*w6 + b21)
[1] 1.105906
> (outY1 = sigmoid(y1))
[1] 0.7513651
> (y2 = outH1*w7 + outH2*w8 + b22)
[1] 1.224921
> (outY2 = sigmoid(y2))
[1] 0.7729285
>
```

$$Error_1 = \frac{1}{2}(T_1 - outY1)^2$$
$$Error_2 = \frac{1}{2}(T_2 - outY2)^2$$
$$Error_{Total} = Error_1 + Error_2$$
$$Error_{Total} = \frac{1}{2}(T_1 - outY1)^2 + \frac{1}{2}(T_2 - outY2)^2$$

#	x1	x2	T1	T2
1	0.05	0.10	0.01	0.99

```
> ##########################
> # Error
> #
> (E1 = (1/2)*(outY1 - T1)^2)
[1] 0.2748111
> (E2 = (1/2)*(outY2 - T2)^2)
[1] 0.02356003
> (eTotal = E1+E2)
[1] 0.2983711
```

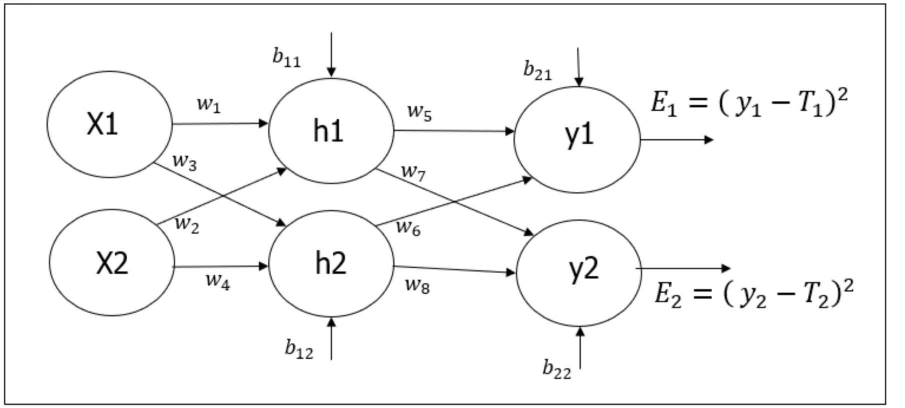
Back Propagation



- The following weights can be changed to reduce the error
  - *Initial Values*
  - $w_1 = 0.15, \;\; b_{11} = b_{12} = 0.35$
  - $w_2 = 0.20$
  - $w_3 = 0.25$
  - $w_4 = 0.30$
  - $w_5 = 0.40, \;\; b_{21} = b_{22} = 0.60$
  - $w_6 = 0.45$
  - $w_7 = 0.50$
  - $w_8 = 0.55$
- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_5}, \frac{\partial E_{Total}}{\partial w_6}, \frac{\partial E_{Total}}{\partial w_7}, \frac{\partial E_{Total}}{\partial w_8}, \frac{\partial\, E_{Total}}{\partial\, b_{21}}, \frac{\partial\, E_{Total}}{\partial\, b_{22}}$
  - $\frac{\partial E_{Total}}{\partial w_1}, \frac{\partial E_{Total}}{\partial w_2}, \frac{\partial E_{Total}}{\partial w_3}, \frac{\partial E_{Total}}{\partial w_4}, \frac{\partial\, E_{Total}}{\partial\, b_{11}}, \frac{\partial\, E_{Total}}{\partial\, b_{12}}$

Backpropagation-1

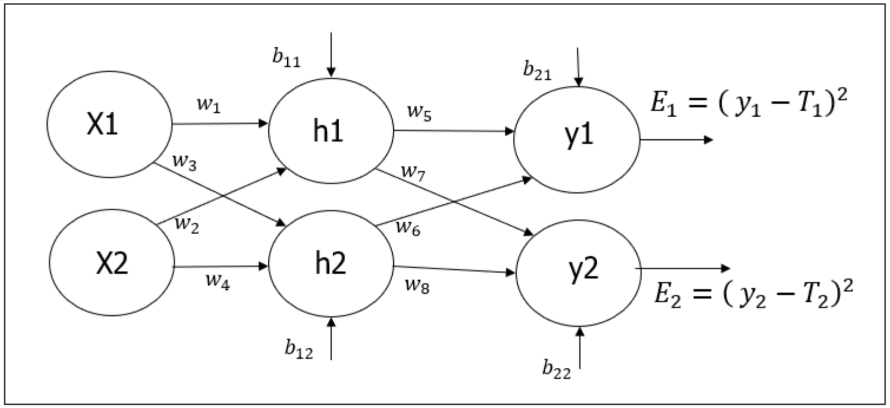
Compute $\frac{\partial E_{Total}}{\partial w_5}$



- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_5} = \frac{\partial E_{Total}}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ w5}$
  - --------------------------------
  - $Error_{Total} = \frac{1}{2}(T_1 - outY1)^2 + \frac{1}{2}(T_2 - outY2)^2$
  - $\frac{\partial E_{Total}}{\partial\ OutY1} = \frac{2}{2}(T_1 - outY1) * (-1) = -(T_1 - outY1)$
  - --------------------------------
  - $\frac{\partial\ OutY1}{\partial\ y1} = outY1(1 - outY1)$, because $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
  - -------------------------------
  - $\text{y1} = outH1 * w_5 + outH2 * w_6 + b_{21}$
  - $\frac{\partial\ y1}{\partial\ w5} = outH1$
  - -----------------------------------
  - $\frac{\partial E_{Total}}{\partial w_5} = -(T_1 - outY1) * outY1(1 - outY1) * outH1$

Back Propagation-1

Compute $\frac{\partial E_{Total}}{\partial w_5}$



- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_5} = \frac{\partial E_{Total}}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ w5}$
  - $\frac{\partial E_{Total}}{\partial w_5} = -(T_1 - outY1) * outY1(1 - outY1) * outH1$

```
> d.sigmoid <- function(x) {
+   return( x*(1-x) )
+ }
> (d.eTotal_d.outY1 = -(T1-outY1))
[1] 0.7413651
> (d.outY1_d.y1 = d.sigmoid(outY1))
[1] 0.1868156
> (d.y1_d.w5 = outH1)
[1] 0.59327

> (d.eTotal_d.w5 = d.eTotal_d.outY1 * d.outY1_d.y1 * d.y1_d.w5)
[1] 0.08216704
> (newW5 = w5 - LearningRate*d.eTotal_d.w5)
[1] 0.3589165
>
```

$Learning\ Rate(LR) = 0.5$

$Old\ w_5 = 0.40$

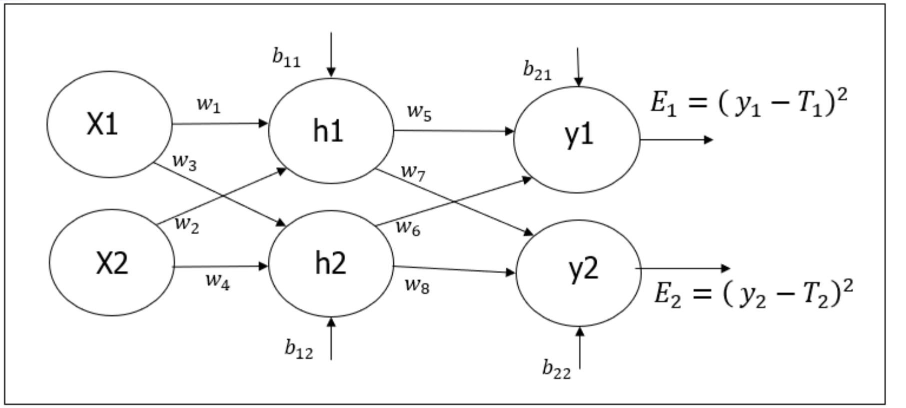
$New\ w_5 = Old\ w_5 - LR * \frac{\partial E_{Total}}{\partial w_5}$

$New\ w_5 = 0.3589$

Copyright 2019 - Dr. Ash Pahwa

Back Propagation-2

Compute $\frac{\partial E_{Total}}{\partial w_6}$

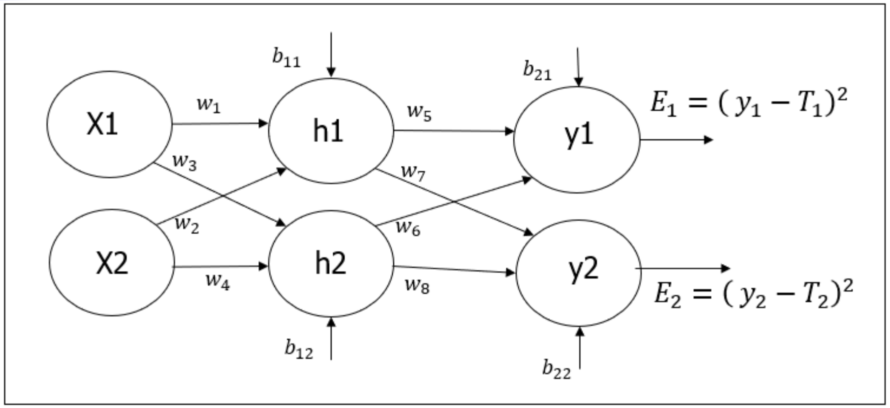


- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_6} = \frac{\partial E_{Total}}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ w6}$
  - -------------------------------
  - $Error_{Total} = \frac{1}{2}(T_1 - outY1)^2 + \frac{1}{2}(T_2 - outY2)^2$
  - $\frac{\partial E_{Total}}{\partial\ OutY1} = \frac{2}{2}(T_1 - outY1) * (-1) = -(T_1 - outY1)$
  - -------------------------------
  - $\frac{\partial\ OutY1}{\partial\ y1} = outY1(1 - outY1)$, because $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
  - ------------------------------
  - $\mathrm{y1} = outH1 * w_5 + outH2 * w_6 + b_{21}$
  - $\frac{\partial\ y1}{\partial\ w6} = outH2$
  - ---------------------------------
  - $\frac{\partial E_{Total}}{\partial w_6} = -(T_1 - outY1) * outY1(1 - outY1) * outH2$

Copyright 2019 - Dr. Ash Pahwa

Back Propagation-2

Compute $\frac{\partial E_{Total}}{\partial w_6}$



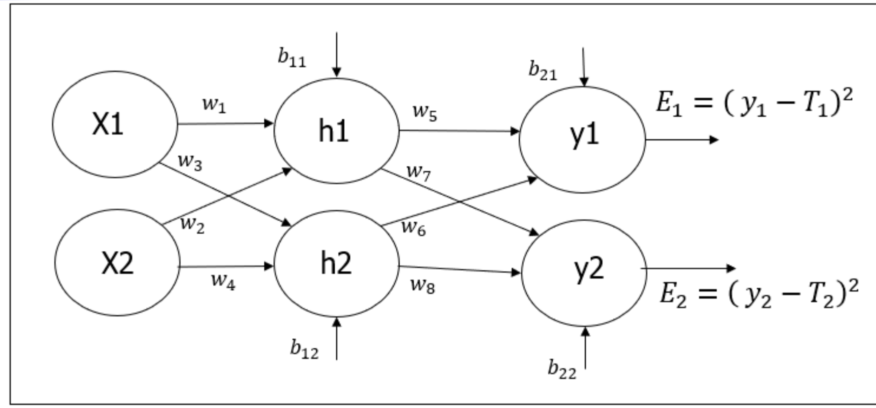
- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_6} = \frac{\partial E_{Total}}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ w6}$
  - $\frac{\partial E_{Total}}{\partial w_6} = -(T_1 - outY1) * outY1(1 - outY1) * outH2$

```
> (d.eTotal_d.outY1 = -(T1-outY1))
[1] 0.7413651
> (d.outY1_d.y1 = d.sigmoid(outY1))
[1] 0.1868156
> (d.y1_d.w6 = outH2)
[1] 0.5968844



> (d.eTotal_d.w6 = d.eTotal_d.outY1 * d.outY1_d.y1 * d.y1_d.w6)
[1] 0.08266763
> (newW6 = w6 - LearningRate*d.eTotal_d.w6)
[1] 0.4086662
>
```

$$Learning\ Rate(LR) = 0.5$$
$$Old\ w_6 = 0.45$$
$$New\ w_6 = Old\ w_6 - LR * \frac{\partial E_{Total}}{\partial w_6}$$
$$New\ w_6 = 0.40866$$

Back Propagation-3

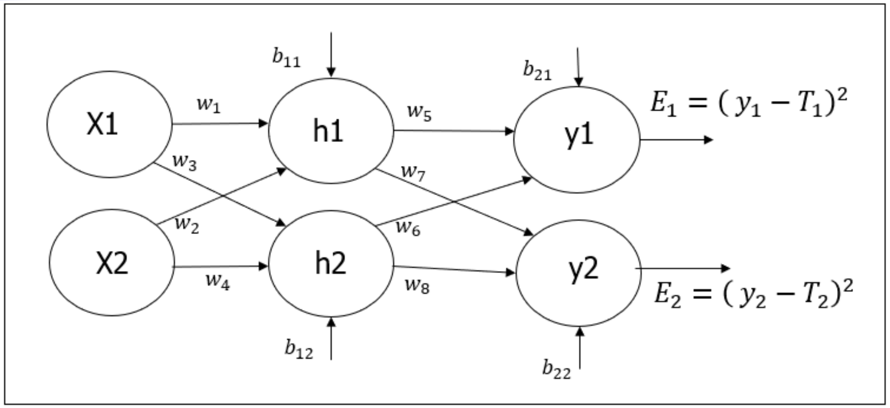
Compute $\frac{\partial E_{Total}}{\partial w_7}$



- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_7} = \frac{\partial E_{Total}}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ w7}$
  - ------------------------------
  - $Error_{Total} = \frac{1}{2}(T_1 - outY1)^2 + \frac{1}{2}(T_2 - outY2)^2$
  - $\frac{\partial E_{Total}}{\partial\ OutY2} = \frac{2}{2}(T_2 - outY2) * (-1) = -(T_2 - outY2)$
  - ------------------------------
  - $\frac{\partial\ OutY2}{\partial\ y2} = outY2(1 - outY2)$, because $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
  - ------------------------------
  - $y2 = outH1 * w_7 + outH2 * w_8 + b_{22}$
  - $\frac{\partial\ y2}{\partial\ w7} = outH1$
  - ---------------------------------
  - $\frac{\partial E_{Total}}{\partial w_7} = -(T_2 - outY2) * outY2(1 - outY2) * outH1$

Back Propagation-3

Compute $\frac{\partial E_{Total}}{\partial w_7}$



- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_7} = \frac{\partial E_{Total}}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ w7}$
  - $\frac{\partial E_{Total}}{\partial w_7} = -(T_2 - outY2) * outY2(1 - outY2) * outH1$

```
> (d.eTotal_d.outY2 = -(T2-outY2))
[1] -0.2170715
> (d.outY2_d.y2 = d.sigmoid(outY2))
[1] 0.1755101
> (d.y2_d.w7 = outH1)
[1] 0.59327


> (d.eTotal_d.w7 = d.eTotal_d.outY2 * d.outY2_d.y2 * d.y2_d.w7)
[1] -0.02260254
> (newW7 = w7 - LearningRate*d.eTotal_d.w7)
[1] 0.5113013
> >
```

$Learning\ Rate(LR) = 0.5$

$Old\ w_7 = 0.50$

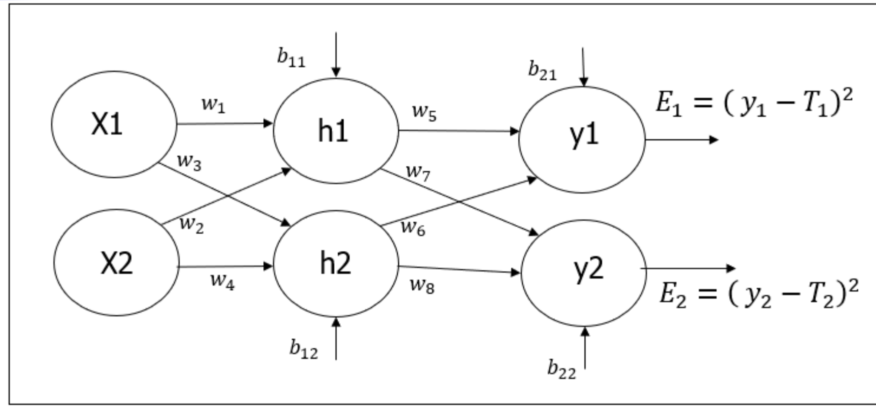
$New\ w_7 = Old\ w_7 - LR * \frac{\partial E_{Total}}{\partial w_7}$

$New\ w_7 = 0.5113$

Copyright 2019 - Dr. Ash Pahwa

Back Propagation-4

Compute $\frac{\partial E_{Total}}{\partial w_8}$



- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_8} = \frac{\partial E_{Total}}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ w8}$
  - $\frac{\partial E_{Total}}{\partial w_8} = -(T_2 - outY2) * outY2(1 - outY2) * outH2$

```
> (d.eTotal_d.outY2 = -(T2-outY2))
[1] -0.2170715
> (d.outY2_d.y2 = d.sigmoid(outY2))
[1] 0.1755101
> (d.y2_d.w8 = outH2)
[1] 0.5968844


> (d.eTotal_d.w8 = d.eTotal_d.outY2 * d.outY2_d.y2 * d.y2_d.w8)
[1] -0.02274024
> (newW8 = w8 - LearningRate*d.eTotal_d.w8)
[1] 0.5613701
>
```

$$Learning\ Rate(LR) = 0.5$$
$$Old\ w_8 = 0.55$$
$$New\ w_8 = Old\ w_8 - LR * \frac{\partial E_{Total}}{\partial w_8}$$
$$New\ w_8 = 0.5613$$

New Weight Values

- *Initial Values*
- $w_5 = 0.40$
- $w_6 = 0.45$
- $w_7 = 0.50$
- $w_8 = 0.55$

$$\frac{\partial E_{Total}}{\partial w_5} = 0.0821$$
$$\frac{\partial E_{Total}}{\partial w_6} = 0.0826$$
$$\frac{\partial E_{Total}}{\partial w_7} = -0.0226$$
$$\frac{\partial E_{Total}}{\partial w_8} = -0.0227$$

$$Learning\ Rate(LR) = 0.5$$
$$Old\ w_5 = 0.40$$
$$New\ w_5 = Old\ w_5 - LR * \frac{\partial E_{Total}}{\partial w_5}$$
$$New\ w_5 = 0.3589$$

$$Learning\ Rate(LR) = 0.5$$
$$Old\ w_6 = 0.45$$
$$New\ w_6 = Old\ w_6 - LR * \frac{\partial E_{Total}}{\partial w_6}$$
$$New\ w_6 = 0.4086$$

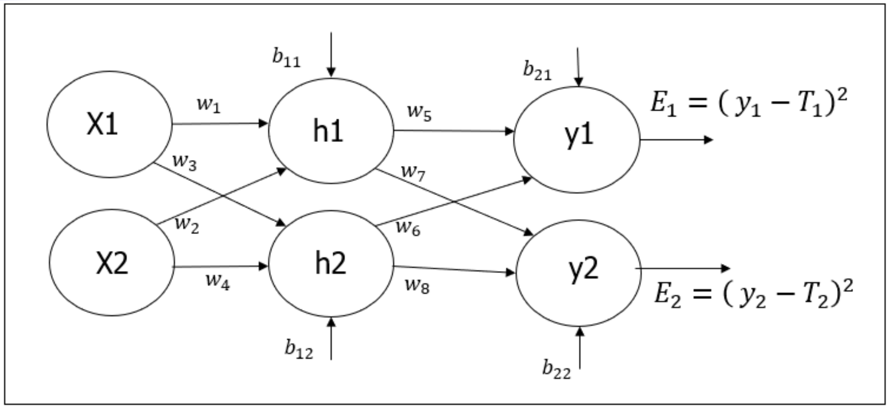
$$Learning\ Rate(LR) = 0.5$$
$$Old\ w_7 = 0.50$$
$$New\ w_7 = Old\ w_7 - LR * \frac{\partial E_{Total}}{\partial w_7}$$
$$New\ w_7 = 0.5113$$

$$Learning\ Rate(LR) = 0.5$$
$$Old\ w_8 = 0.55$$
$$New\ w_8 = Old\ w_8 - LR * \frac{\partial E_{Total}}{\partial w_8}$$
$$New\ w_8 = 0.5613$$

- *New Values*
- $w_5 = 0.3589$
- $w_6 = 0.4086$
- $w_7 = 0.5113$
- $w_8 = 0.5613$

Back Propagation-5

Compute $\frac{\partial E_{Total}}{\partial w_1}$

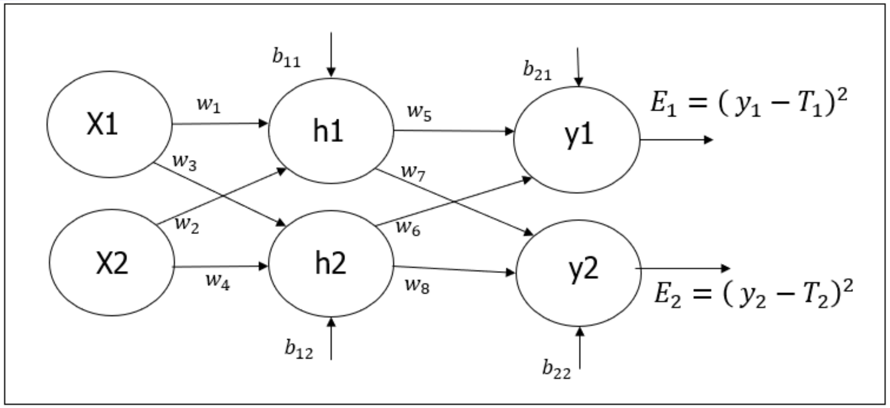


- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_1} = \frac{\partial E_{Total}}{\partial\ OutH1} * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w1}$
  - --------------------------------
  - $\frac{\partial E_{Total}}{\partial\ OutH1} = \frac{\partial E_1}{\partial\ OutH1} + \frac{\partial E_2}{\partial\ OutH1}\ because\ E_{Total} = E_1 + E_2$
  - $\frac{\partial E_1}{\partial\ OutH1} = \frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}$
  - $\frac{\partial E_2}{\partial\ OutH1} = \frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}$
  - --------------------------------
  - $\frac{\partial E_{Total}}{\partial w_1} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w1}$

Copyright 2019 - Dr. Ash Pahwa

Back Propagation-5

Compute $\frac{\partial E_{Total}}{\partial w_1}$



- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_1} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w1}$

- $E_1 = \frac{1}{2}(T_1 - outY1)^2$
- $\frac{\partial E_1}{\partial\ OutY1} = (-1) * (T_1 - outY1)$
- --
- $outY1 = \sigma(y1)$
- $\frac{\partial\ OutY1}{\partial\ y1} = outY1(1 - outY1),$
  - because $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
- --
- $y1 = outH1 * w_5 + outH2 * w_6 + b_{21}$
- $\frac{\partial\ y1}{\partial\ OutH1} = w_5$

- $E_2 = \frac{1}{2}(T_2 - outY2)^2$
- $\frac{\partial E_2}{\partial\ OutY2} = (-1) * (T_2 - outY2)$
- --
- $outY2 = \sigma(y2)$
- $\frac{\partial\ OutY2}{\partial\ y2} = outY2(1 - outY2),$
  - because $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
- --
- $y2 = outH1 * w_7 + outH2 * w_8 + b_{22}$
- $\frac{\partial\ y2}{\partial\ OutH1} = w_7$

- $outH1 = \sigma(h1)$
- $\frac{\partial\ outH1}{\partial\ h1} = outH1(1 - outH1)$

- $h1 = x_1 * w_1 + x_2 * w_2 + b_{11}$
- $\frac{\partial\ h1}{\partial\ w1} = x_1$

Back Propagation-5: Compute $\frac{\partial E_{Total}}{\partial w_1}$

- Need to Compute
  - $\frac{\partial E_{Total}}{\partial w_1} = \left(\left(\frac{\partial E_1}{\partial\, OutY1} * \frac{\partial\, OutY1}{\partial\, y1} * \frac{\partial\, y1}{\partial\, OutH1}\right) + \left(\frac{\partial E_2}{\partial\, OutY2} * \frac{\partial\, OutY2}{\partial\, y2} * \frac{\partial\, y2}{\partial\, OutH1}\right)\right) * \frac{\partial\, OutH1}{\partial\, h1} * \frac{\partial\, h1}{\partial\, w1}$

- $\frac{\partial E_1}{\partial\, OutY1} = (-1) * (T_1 - outY1)$
- $\frac{\partial\, OutY1}{\partial\, y1} = outY1(1 - outY1)$
- $\frac{\partial\, y1}{\partial\, OutH1} = w_5$

- $\frac{\partial E_2}{\partial\, OutY2} = (-1) * (T_2 - outY2)$
- $\frac{\partial\, OutY2}{\partial\, y2} = outY2(1 - outY2)$
- $\frac{\partial\, y2}{\partial\, OutH1} = w_7$

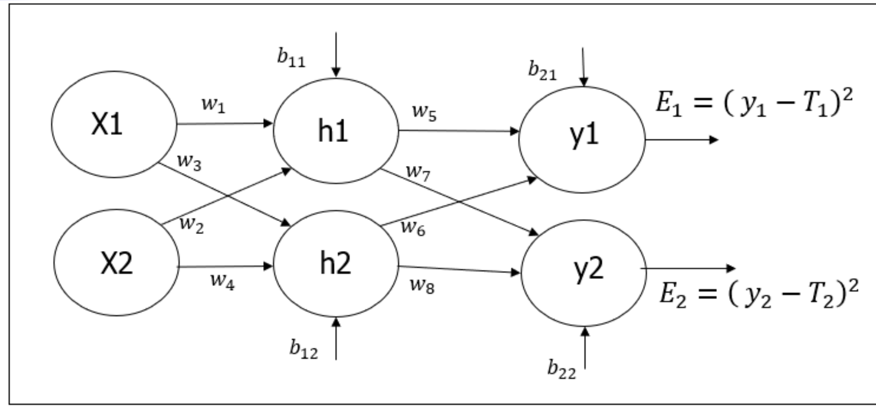
```
> (d.E1_d.outY1 = (-1)*(T1 - outY1))
[1] 0.7413651
> (d.outY1_d.y1 = d.sigmoid(outY1))
[1] 0.1868156
> (d.y1_d.outH1 = w5)
[1] 0.4
> (d.E1_d.outH1 = d.E1_d.outY1 * d.outY1_d.y1 * d.y1_d.outH1)
[1] 0.05539942
```

```
> (d.E2_d.outY2 = (-1)*(T2 - outY2))
[1] -0.2170715
> (d.outY2_d.y2 = d.sigmoid(outY2))
[1] 0.1755101
> (d.y2_d.outH1 = w7)
[1] 0.5
> (d.E2_d.outH1 = d.E2_d.outY2 * d.outY2_d.y2 * d.y2_d.outH1)
[1] -0.01904912
```

Copyright 2019 - Dr. Ash Pahwa

Back Propagation-5

Compute $\frac{\partial E_{Total}}{\partial w_1}$



- Need to Compute
  - $$\frac{\partial E_{Total}}{\partial w_1} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w1}$$

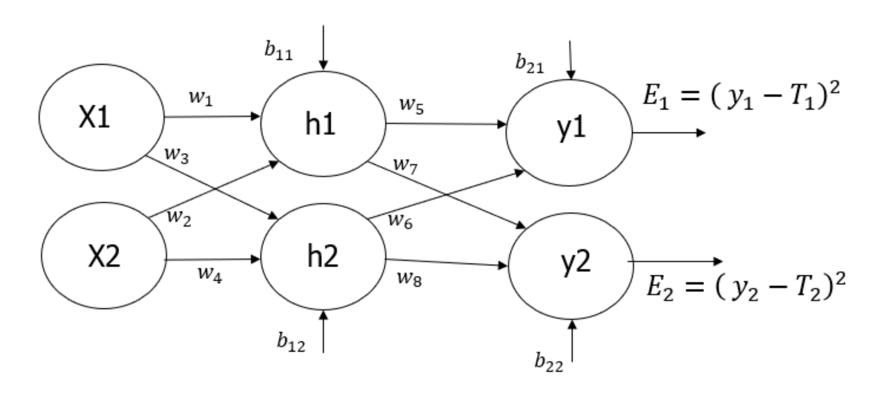
```
> (d.ETotal_d.outH1 = d.E1_d.outH1 + d.E2_d.outH1)
[1] 0.03635031
```

```
> (d.outH1_d.h1 = d.sigmoid(outH1))
[1] 0.2413007
> (d.h1_d.w1 = x1)
[1] 0.05

> (d.Etotal_d.w1 = d.ETotal_d.outH1 * d.outH1_d.h1 * d.h1_d.w1)
[1] 0.0004385677
>
> (newW1 = w1 - LearningRate*d.Etotal_d.w1)
[1] 0.1497807
```

$$Learning\ Rate(LR) = 0.5$$
$$Old\ w_1 = 0.15$$
$$New\ w_1 = Old\ w_1 - LR * \frac{\partial E_{Total}}{\partial w_1}$$
$$New\ w_1 = 0.1497$$

Back Propagation-6

Compute $\frac{\partial E_{Total}}{\partial w_2}$



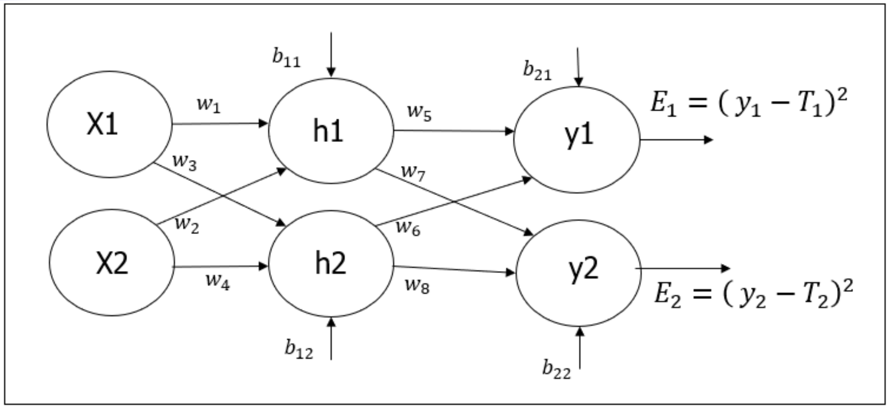
- $$\frac{\partial E_{Total}}{\partial w_1} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w1}$$

- $$\frac{\partial E_{Total}}{\partial w_2} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w2}$$

To compute $\frac{\partial E_{Total}}{\partial w_2}$,
only change
- from $\frac{\partial E_{Total}}{\partial w_1}$
- to $\frac{\partial E_{Total}}{\partial w_2}$
- is the last term.

Back Propagation-6

Compute $\frac{\partial E_{Total}}{\partial w_2}$



- $\frac{\partial E_{Total}}{\partial w_2} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w2}$

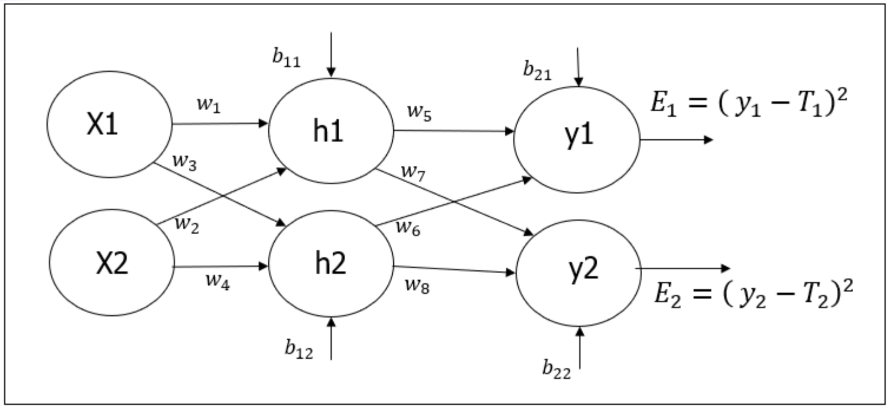
- $E_1 = \frac{1}{2}(T_1 - outY1)^2$
- $\frac{\partial E_1}{\partial\ OutY1} = (-1) * (T_1 - outY1)$
- --
- $outY1 = \sigma(y1)$
- $\frac{\partial\ OutY1}{\partial\ y1} = outY1(1 - outY1),$
  - because $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
- ---------------------------------------
- $y1 = outH1 * w_5 + outH2 * w_6 + b_{21}$
- $\frac{\partial\ y1}{\partial\ OutH1} = w_5$

- $E_2 = \frac{1}{2}(T_2 - outY2)^2$
- $\frac{\partial E_2}{\partial\ OutY2} = (-1) * (T_2 - outY2)$
- --
- $outY2 = \sigma(y2)$
- $\frac{\partial\ OutY2}{\partial\ y2} = outY2(1 - outY2),$
  - because $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
- ---------------------------------------
- $y2 = outH1 * w_7 + outH2 * w_8 + b_{22}$
- $\frac{\partial\ y2}{\partial\ OutH1} = w_7$

- $outH1 = \sigma(h1)$
- $\frac{\partial\ outH1}{\partial\ h1} = outH1(1 - outH1)$

- $h1 = x_1 * w_1 + x_2 * w_2 + b_{11}$
- $\frac{\partial\ h1}{\partial\ w2} = x_2$

Back Propagation-6
Compute $\frac{\partial E_{Total}}{\partial w_2}$



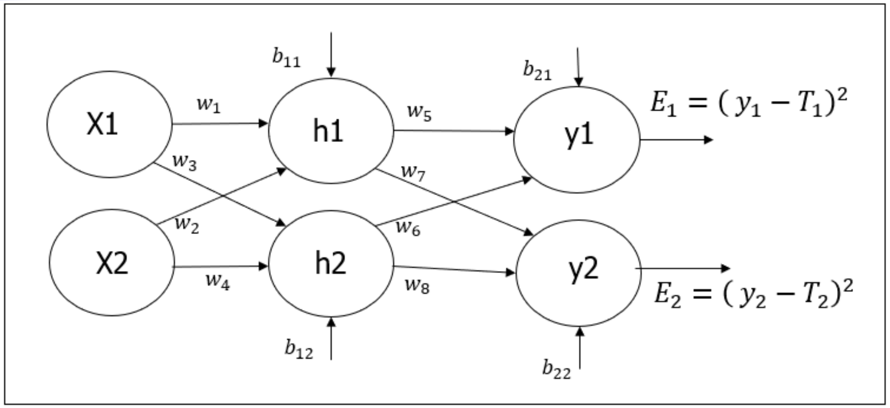
- $$\frac{\partial E_{Total}}{\partial w_2} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w2}$$

```
> (d.E1_d.outY1 = (-1)*(T1 - outY1))
[1] 0.7413651
> (d.outY1_d.y1 = d.sigmoid(outY1))
[1] 0.1868156
> (d.y1_d.outH1 = w5)
[1] 0.4
➢ (d.E1_d.outH1 = d.E1_d.outY1 *
➢          d.outY1_d.y1 * d.y1_d.outH1)
[1] 0.05539942


> (d.E2_d.outY2 = (-1)*(T2 - outY2))
[1] -0.2170715
> (d.outY2_d.y2 = d.sigmoid(outY2))
[1] 0.1755101
> (d.y2_d.outH1 = w7)
[1] 0.5
➢ (d.E2_d.outH1 = d.E2_d.outY2 *
➢           d.outY2_d.y2 * d.y2_d.outH1)
[1] -0.01904912
```

```
> (d.ETotal_d.outH1 = d.E1_d.outH1 + d.E2_d.outH1)
[1] 0.03635031
```

Back Propagation-6

Compute $\frac{\partial E_{Total}}{\partial w_2}$



- $$\frac{\partial E_{Total}}{\partial w_2} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w2}$$

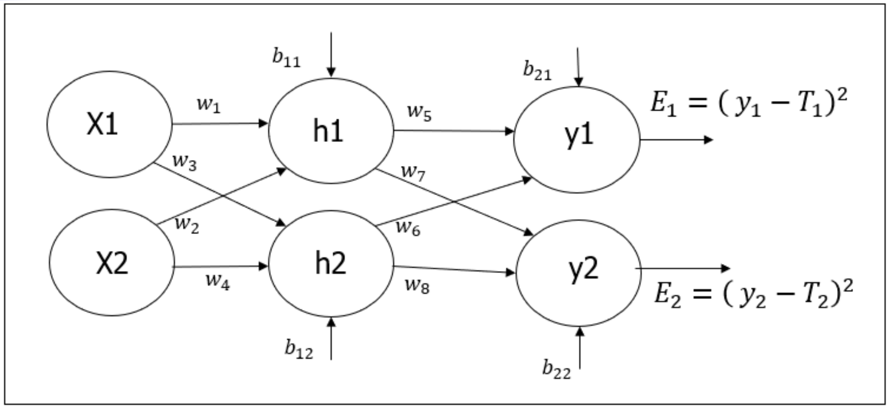
```
> (d.ETotal_d.outH1 = d.E1_d.outH1 + d.E2_d.outH1)
[1] 0.03635031
```

```
> ###################
> (d.outH1_d.h1 = d.sigmoid(outH1))
[1] 0.2413007
>
> (d.h1_d.w2 = x2)
[1] 0.1
>
> ##################
➢ (d.Etotal_d.w2 = d.ETotal_d.outH1 *
➢         d.outH1_d.h1 * d.h1_d.w2)
[1] 0.0008771355
>
> (newW2 = w2 - LearningRate*d.Etotal_d.w2)
[1] 0.1995614
```

$$Learning\ Rate(LR) = 0.5$$
$$Old\ w_2 = 0.20$$
$$New\ w_2 = Old\ w_2 - LR * \frac{\partial E_{Total}}{\partial w_2}$$
$$New\ w_2 = 0.1995$$

Back Propagation-7

Compute $\frac{\partial E_{Total}}{\partial w_3}$



- $$\frac{\partial E_{Total}}{\partial w_1} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w1}$$

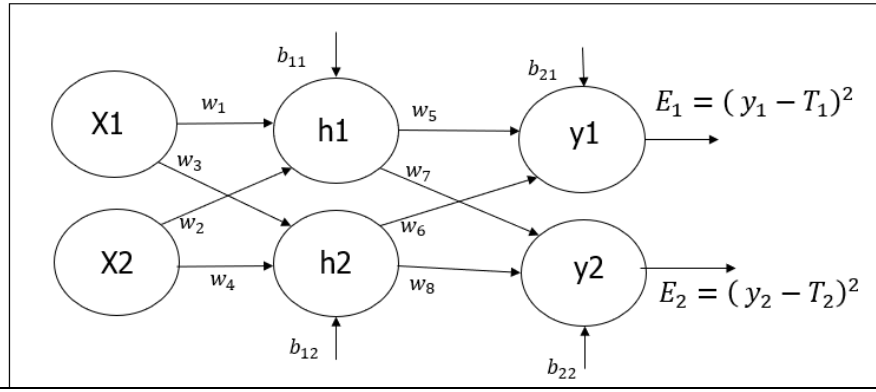
- $$\frac{\partial E_{Total}}{\partial w_2} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w2}$$

- $$\frac{\partial E_{Total}}{\partial w_3} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH2}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH2}\right)\right) * \frac{\partial\ OutH2}{\partial\ h2} * \frac{\partial\ h2}{\partial\ w3}$$

Copyright 2019 - Dr. Ash Pahwa

Back Propagation-7

Compute $\frac{\partial E_{Total}}{\partial w_3}$



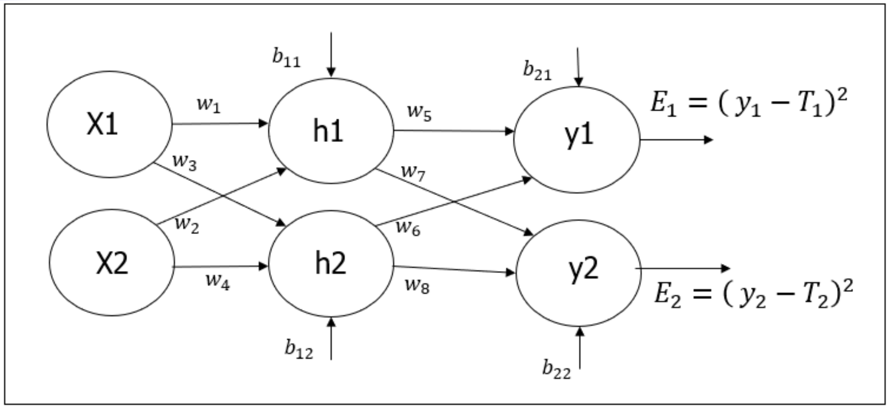
- $$\frac{\partial E_{Total}}{\partial w_3} = \left(\left(\frac{\partial E_1}{\partial\, OutY1} * \frac{\partial\, OutY1}{\partial\, y1} * \frac{\partial\, y1}{\partial\, OutH2}\right) + \left(\frac{\partial E_2}{\partial\, OutY2} * \frac{\partial\, OutY2}{\partial\, y2} * \frac{\partial\, y2}{\partial\, OutH2}\right)\right) * \frac{\partial\, OutH2}{\partial\, h2} * \frac{\partial\, h2}{\partial\, w3}$$

```
> (d.E1_d.outY1 = (-1)*(T1 - outY1))
[1] 0.7413651
> (d.outY1_d.y1 = d.sigmoid(outY1))
[1] 0.1868156
> (d.y1_d.outH2 = w6)
[1] 0.45
➢ (d.E1_d.outH1 = d.E1_d.outY1 *
➢        d.outY1_d.y1 * d.y1_d.outH1)
[1] 0.05539942
>
> (d.E2_d.outY2 = (-1)*(T2 - outY2))
[1] -0.2170715
> (d.outY2_d.y2 = d.sigmoid(outY2))
[1] 0.1755101
> (d.y2_d.outH2 = w8)
[1] 0.55
➢ (d.E2_d.outH1 = d.E2_d.outY2 *
➢         d.outY2_d.y2 * d.y2_d.outH1)
[1] -0.01904912
>
> (d.ETotal_d.outH1 = d.E1_d.outH1 + d.E2_d.outH1)
[1] 0.03635031
```

```
> ###################
> (d.outH1_d.h2 = d.sigmoid(outH2))
[1] 0.2406134
>
> (d.h2_d.w3 = x1)
[1] 0.05
>
> ##################
>
➢ (d.Etotal_d.w3 = d.ETotal_d.outH1 *
➢            d.outH1_d.h2 * d.h2_d.w3)
[1] 0.0004373186
>
> (newW3 = w3 - LearningRate*d.Etotal_d.w3)
[1] 0.2497813
>
```

Back Propagation-8

Compute $\frac{\partial E_{Total}}{\partial w_4}$



- $$\frac{\partial E_{Total}}{\partial w_1} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w1}$$

- $$\frac{\partial E_{Total}}{\partial w_2} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH1}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH1}\right)\right) * \frac{\partial\ OutH1}{\partial\ h1} * \frac{\partial\ h1}{\partial\ w2}$$

- $$\frac{\partial E_{Total}}{\partial w_3} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH2}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH2}\right)\right) * \frac{\partial\ OutH2}{\partial\ h2} * \frac{\partial\ h2}{\partial\ w3}$$

- $$\frac{\partial E_{Total}}{\partial w_4} = \left(\left(\frac{\partial E_1}{\partial\ OutY1} * \frac{\partial\ OutY1}{\partial\ y1} * \frac{\partial\ y1}{\partial\ OutH2}\right) + \left(\frac{\partial E_2}{\partial\ OutY2} * \frac{\partial\ OutY2}{\partial\ y2} * \frac{\partial\ y2}{\partial\ OutH2}\right)\right) * \frac{\partial\ OutH2}{\partial\ h2} * \frac{\partial\ h2}{\partial\ w4}$$

- $$\frac{\partial\ E_{Total}}{\partial\ b_{21}}, \frac{\partial\ E_{Total}}{\partial\ b_{22}}, \frac{\partial\ E_{Total}}{\partial\ b_{11}}, \frac{\partial\ E_{Total}}{\partial\ b_{12}}\ ???$$

Copyright 2019 - Dr. Ash Pahwa

New Weight Values

- $Initial\ Values$
- $w_5 = 0.40$
- $w_6 = 0.45$
- $w_7 = 0.50$
- $w_8 = 0.55$
- $b_{21} = b_{22} = 0.60$

- $New\ Values$
- $w_5 = 0.3589$
- $w_6 = 0.4086$
- $w_7 = 0.5113$
- $w_8 = 0.5613$
- b_{21}, b_{22}

- $Initial\ Values$
- $w_1 = 0.15$
- $w_2 = 0.20$
- $w_3 = 0.25$
- $w_4 = 0.30$
- $b_{11} = b_{12} = 0.35$

- $New\ Values$
- $w_1 = 0.1497$
- $w_2 = 0.1995$
- $w_3 = 0.2497$
- $w_4 = 0.29956$
- b_{11}, b_{12}

Next Step

- Algorithm
  - Assign random values to all the weights of the NN
  - Take the first observed data
    - Forward Propagation: Compute Output
    - Compute error = (Computed Output – Observed Output)^2
    - Backpropagation: adjust weights to reduce error
    - Repeat forward, backward propagation, till error is minimized
  - Repeat the previous step for the next sample till all samples are processed
  - The final weights of the NN will be used for prediction

Summary

- Basic Calculus Derivatives
  - Derivative of Sigmoid Function
  - Chain Rule
  - Steepest Descent
- Backpropagation Algorithm
- Example-1
  - Forward Propagation
  - Backpropagation