# Deep Learning


BITS Pilani
Pilani Campus

**Dr. Sugata Ghosal**
**sugata.ghosal@pilani.bits-pilani.ac.in**

innovate achieve lead

BITS Pilani
Pilani Campus

# Worked Out Problems
# Optimization

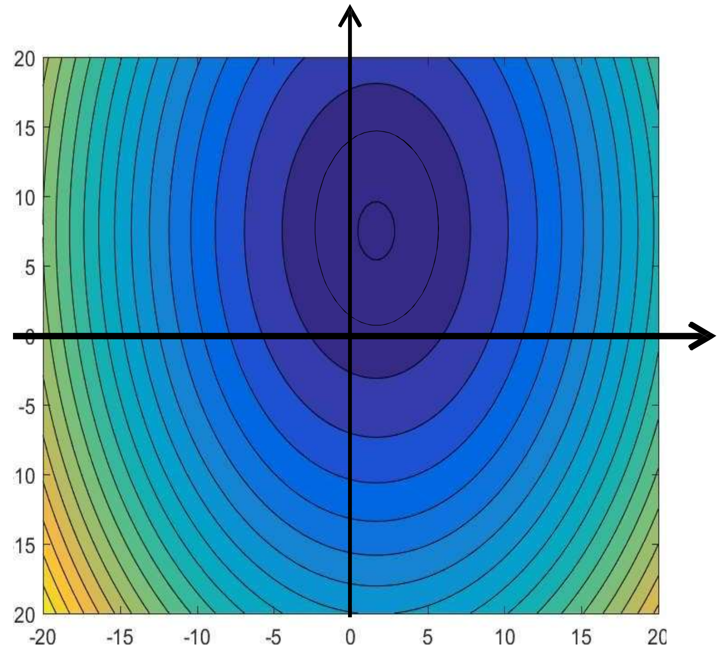
*These slides are assembled by the instructor with grateful acknowledgement of the many others who made their course materials freely available online.*

# Optimal Learning Rate: Multivariate Diagonal Quadratic Error Function

Error surface is given by $E(x,y,z) = 3x^2 + 2y^2 + 4z^2 + 6$. What is the optimal learning rate that leads to fastest convergence to the global minimum?

$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{w}^T\mathbf{b} + c = \frac{1}{2}\sum_i \left(a_{ii}w_i^2 + b_i w_i\right) + c$$

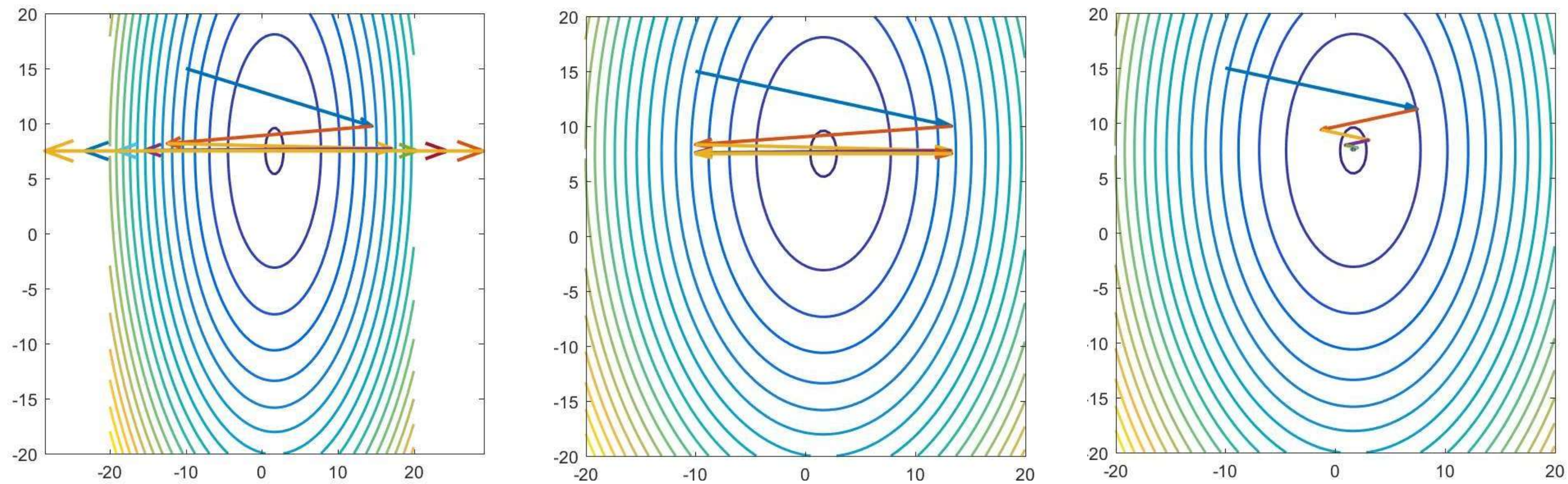
$\eta_{x,opt} = 1/6$
$\eta_{y,opt} = 1/4$
$\eta_{z,opt} = 1/8$

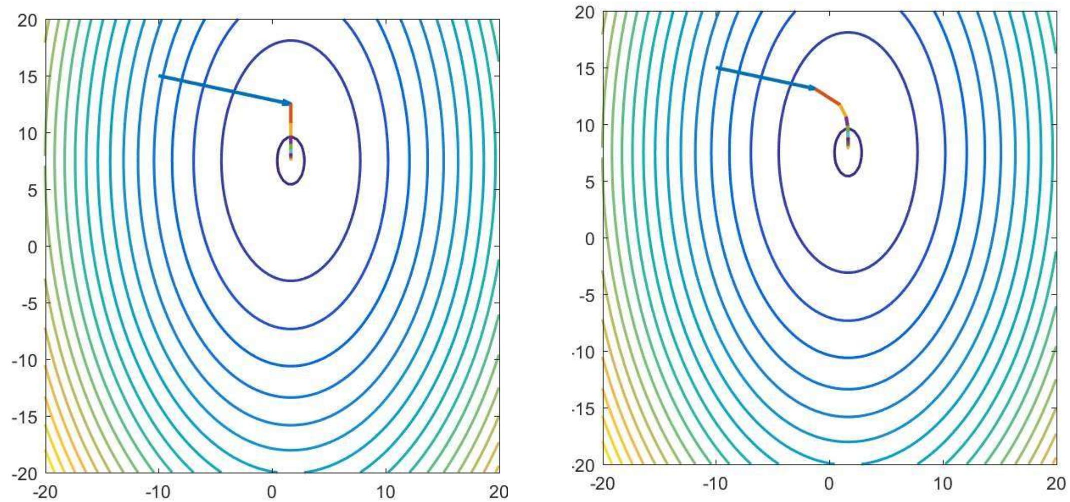



Optimal learning rate = $\min(\eta_{x,opt}, \eta_{y,opt}, \eta_{y,opt}) = 0.125$

Largest learning rate for convergence = $\min(2\eta_{x,opt}, 2\eta_{y,opt}, 2\eta_{y,opt}) = 0.333$

# Dependence on learning rate – Error Minimization





- $\eta_{1,\mathrm{opt}} = 1;\ \eta_{2,\mathrm{opt}} = 0.33$
- $\eta = 2.1\eta_{2,\mathrm{opt}}$
- $\eta = 2\eta_{2,\mathrm{opt}}$
- $\eta = 1.5\eta_{2,\mathrm{opt}}$
- $\eta = \eta_{2,\mathrm{opt}}$
- $\eta = 0.75\eta_{2,\mathrm{opt}}$

# Minimization of Quadratic Error Function

Consider an error function $E(w_1, w_2) = 0.05 + \frac{(w_1-3)^2}{4} + \frac{(w_2-4)^2}{9} - \frac{(w_1-3)(w_2-4)}{6}$. Different variants of gradient descent algorithm can be used to minimize this error function w.r.t. $w_1$, $w_2$. Assume $(w_1, w_2) = (1,1)$ at time *(t-1)* and after update $(w_1, w_2) = (1.5, 2.0)$ at time $t$. Assume, learning rate $\eta = 0.3$ and momentum update rate $\beta = 0.9$.

What is the value of $(w_1, w_2)$ that minimizes this error function? What is the minimum possible value of E?

w1=3, w2=4. $E_{min} = 0.05$

# Weight Updates – Ordinary Gradient Descent

Consider an error function $E(w_1, w_2) = 0.05 + \frac{(w_1-3)^2}{4} + \frac{(w_2-4)^2}{9} - \frac{(w_1-3)(w_2-4)}{6}$. Different variants of gradient descent algorithm can be used to minimize this error function w.r.t. $w_1$, $w_2$. Assume $(w_1, w_2) = (1,1)$ at time *(t-1)* and after update $(w_1, w_2) = (1.5, 2.0)$ at time *t*. Assume, learning rate $\eta = 0.3$ and momentum update rate $\beta = 0.9$.

What is the value of $(w_1, w_2)$ at time (t+1) if standard gradient descent is used?

# Weight Updates – Ordinary Gradient Descent

Consider an error function $E(w_1, w_2) = 0.05 + \frac{(w_1-3)^2}{4} + \frac{(w_2-4)^2}{9} - \frac{(w_1-3)(w_2-4)}{6}$. Different variants of gradient descent algorithm can be used to minimize this error function w.r.t. $w_1$, $w_2$. Assume $(w_1, w_2) = (1,1)$ at time *(t-1)* and after update $(w_1, w_2) = (1.5, 2.0)$ at time $t$. Assume, learning rate $\eta = 0.3$ and momentum update rate $\beta = 0.9$.

What is the value of $(w_1, w_2)$ at time (t+1) if standard gradient descent is used?

δE/δw1=0.5*(w1-3)-(w2-4)/6 and δE/δw2=2/9*(w2-4)-(w1-3)/6

So, w1(t+1)=1.5-0.3*0.5(1.5-3)+0.3*(2-4)/6 = 1.625, and

w2(t+1)=2.0-0.3*2*(2-4)/9+0.3(1.5-3)/6=2.058

**Nestorov**

w1_int=1.5+0.9x.5=1.95 w2_inter=2.0+0.9*1=2.9

dE/dw1(w1_int,w2_int)=0.5x(1.95-3)-(2.9-4)/6=-0.342

dE/dw2(w1_int,w2_int)=2/9x(2.9-4)-(1.95-3)/6=-0.0694

W1(t+1)=1.95+0.3*0.342=2.0526 w2(t+1)=2.9+0.3x0.0694=2.92

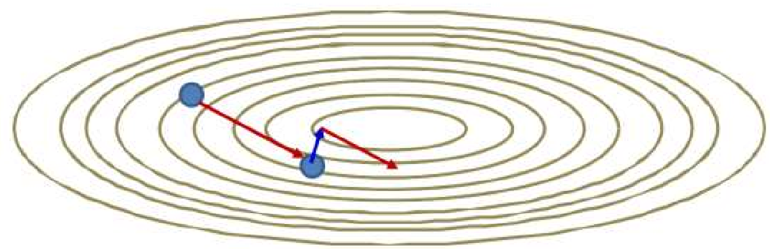
# Weight Updates – Momentum Method

Consider an error function $E(w_1, w_2) = 0.05 + \frac{(w_1-3)^2}{4} + \frac{(w_2-4)^2}{9} - \frac{(w_1-3)(w_2-4)}{6}$. Different variants of gradient descent algorithm can be used to minimize this error function w.r.t. $w_1$, $w_2$. Assume $(w_1, w_2) = (1,1)$ at time *(t-1)* and after update $(w_1, w_2) = (1.5, 2.0)$ at time *t*. Assume, learning rate $\eta = 0.3$ and momentum update rate $\beta = 0.9$.

What is the value of $(w_1, w_2)$ at time (t+1) if momentum based gradient descent is used?

w1(t+1)=1.625+(1.5-1.0)*0.9=2.075

w2(t+1)= 2.058+0.9*(2.0-1.0)=2.958



The momentum method

$$\Delta W^{(k)} = \beta \Delta W^{(k-1)} - \eta \nabla_W Loss(W^{(k-1)})^T$$

**Nestorov**

w1_int=1.5+0.9x.5=1.95 w2_inter=2.0+0.9*1=2.9

dE/dw1(w1_int,w2_int)=0.5x(1.95-3)-(2.9-4)/6=-0.342

dE/dw2(w1_int,w2_int)=2/9x(2.9-4)-(1.95-3)/6=-0.0694

W1(t+1)=1.95+0.3*0.342=2.0526 w2(t+1)=2.9+0.3x0.0694=2.92

# Weight Updates – RProp

Consider an error function $E(w_1, w_2) = 0.05 + \frac{(w_1-3)^2}{4} + \frac{(w_2-4)^2}{9} - \frac{(w_1-3)(w_2-4)}{6}$. Different variants of gradient descent algorithm can be used to minimize this error function w.r.t. $w_1$, $w_2$. Assume $(w_1, w_2) = (1,1)$ at time *(t-1)* and after update $(w_1, w_2) = (1.5, 2.0)$ at time $t$.

Assume, α = 1.5, β = 0.6

What will be (w1,w2) at (t+1)?

At time t-1,
dE/dw1 =0.5*(1-3)-(1-4)/6=-0.5
dE/w2=2/9*(1-4)-(1- 3)/6=-0.333

At time t,
dE/dw1 =0.5*(1.5-3)-(2.0-4)/6=0.4167
dE/w2=2/9*(2-4)-(1.5- 3)/6=-0.194

Delta w1 = 1.5-1 =0.5 Delta w2 = 2-1=1

w1(t+1)= 1+0.5*0.6 = 1.3, sign of derivation became different
w2(t+1)= 2.0+1.5*1=3.5, sign of derivative remained same

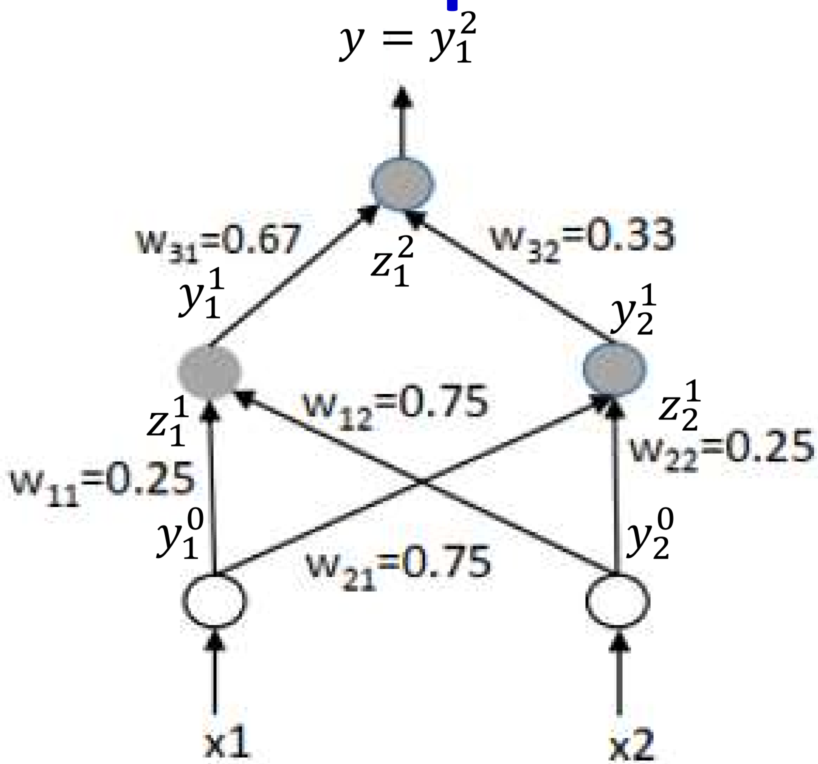
innovate achieve lead

BITS Pilani
Pilani Campus

# Solved Examples

*These slides are assembled by the instructor with grateful acknowledgement of the many others who made their course materials freely available online.*

# Weight Updates in Deep Networks

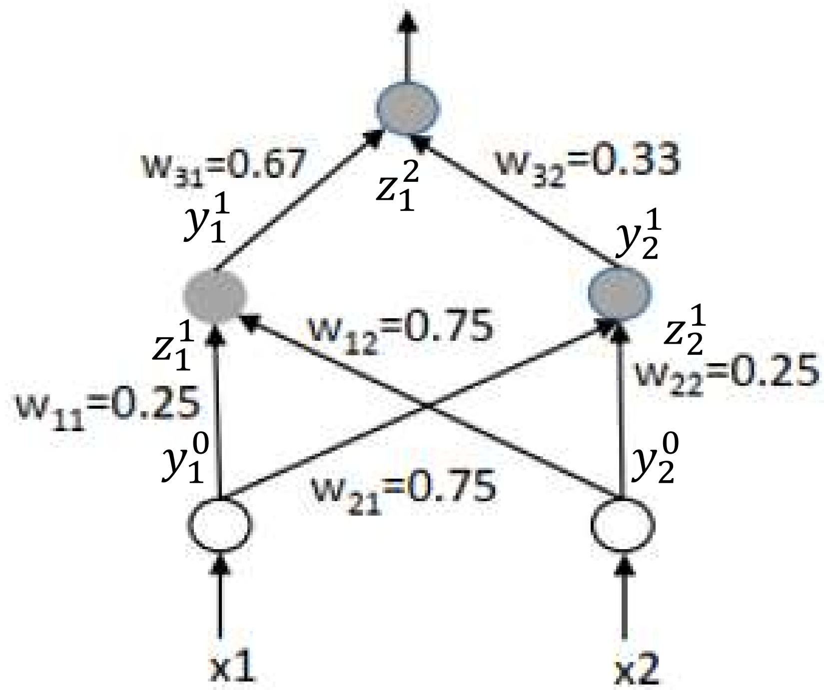



- Training Input: (x1,x2)=(1,1) Target Output: d=0.0
- *div* function: square error
- Activation function *f*( ): ReLU
- Bias: 0 at all nodes
- $\eta = 0.1$
- What are the values of $w_{31}$ and $w_{12}$ in next iteration?

# Weight Updates with ReLU and square error

- Training Input: (x1,x2)=(1,1)
- Target Output: d=0.0

$w_{31}$(t+1) = ?

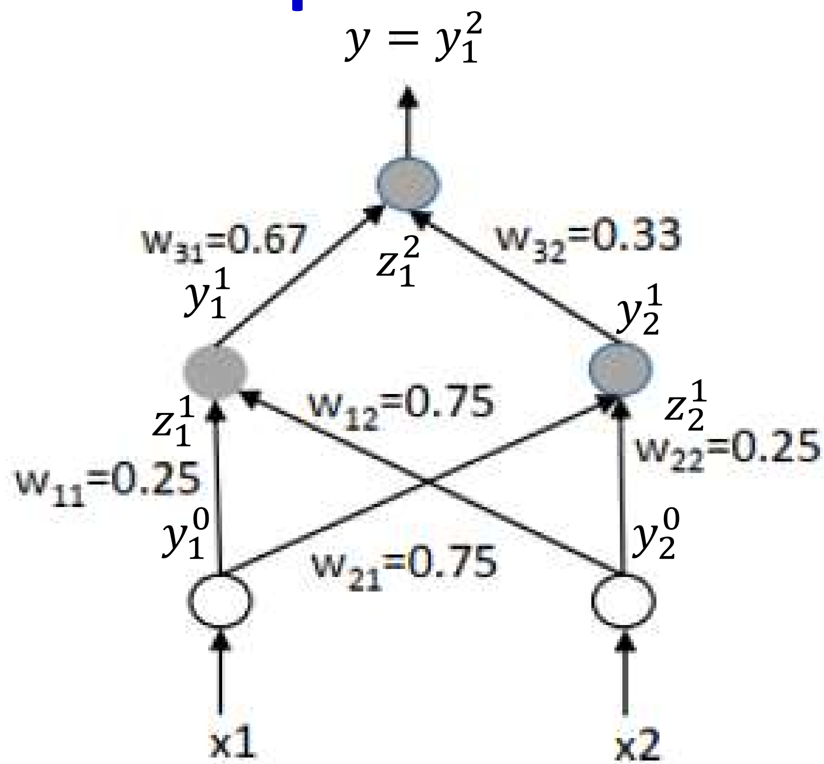



$$w_{31} = w_{31} - \eta * \frac{\delta div}{\delta w_{31}} \text{=0.67-0.1=0.57}$$

# Weight Updates with ReLU and square error

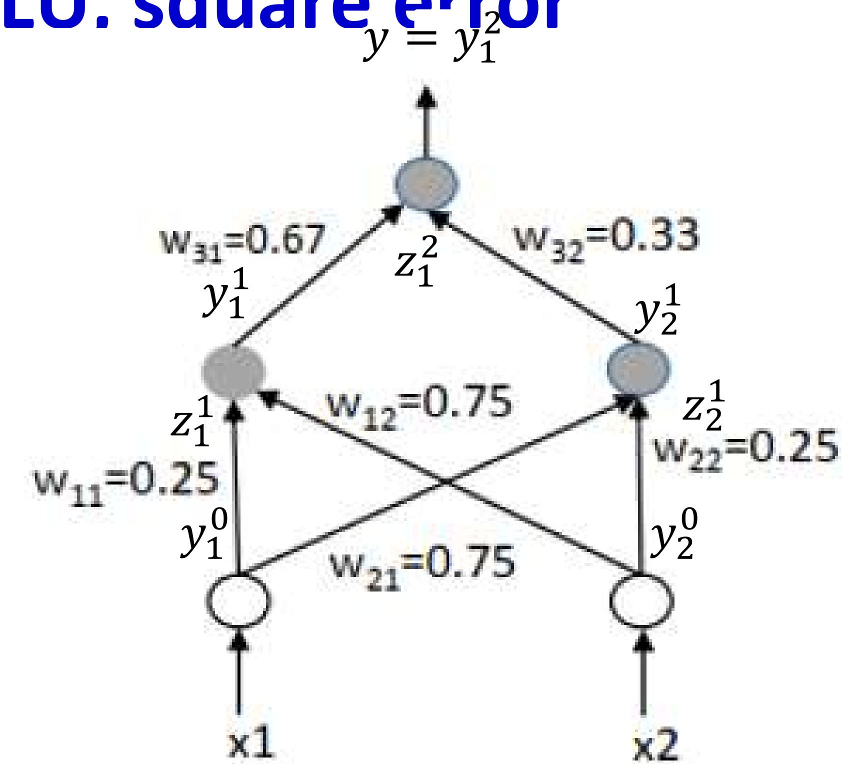
- Training Input: (x1,x2)=(1,1)
- Target Output: d=0.0

$w_{12}$ (t+1) = ?




$$w_{12} = w_{12} - \eta * \frac{\delta div}{\delta w_{12}} = 0.75 - .067 = 0.683$$

# Weight Updates with ReLU. square error




$z_1^1$ = 0.25*1+0.75*1=1.0

$z_2^1$ = 0.75*1+0.25*1=1.0

$y_1^1$ = ReLU($z_1^1$)= 1  $y_2^1$ = ReLU($z_2^1$)= 1

$z_1^2$ = 0.67*1+0.33*1=1.0

$y_1^2$ = ReLU($z_1^2$)= 1.0

$$div = \frac{1}{2}(d - y)^2$$

$$\frac{\delta div}{\delta y} = (\mathrm{y} - \mathrm{d}) = 1.0$$

$$\frac{\delta div}{\delta z_1^2} = \frac{\delta div}{\delta y_1^2} * \frac{\delta y_1^2}{\delta z_1^2} = 1*1=1$$

$$\frac{\delta div}{\delta w_{31}} = \frac{\delta Div}{\delta z_1^2} * \frac{\delta z_1^2}{\delta w_{31}} = 1 * y_1^1 = 1$$

$$\boxed{w_{31} = w_{31} - \eta * \frac{\delta div}{\delta w_{31}} = 0.67 - 0.1 = 0.57}$$

$$\frac{\delta div}{\delta y_1^1} = \frac{\delta Div}{\delta z_1^2} * \frac{\delta z_1^2}{\delta y_1^1} = 1 * w_{31}$$

$$\frac{\delta div}{\delta z_1^1} = \frac{\delta div}{\delta y_1^1} * \frac{\delta y_1^1}{\delta z_1^1} = 0.67*1=0.67$$

$$\frac{\delta div}{\delta w_{12}} = \frac{\delta div}{\delta z_1^1} * \frac{\delta z_1^1}{\delta w_{12}} = 0.67 * y_2^0 = 0.67$$

$$\boxed{w_{12} = w_{12} - \eta * \frac{\delta div}{\delta w_{12}} = 0.75 - .067 = 0.683}$$

# Weight Updates with ReLU Hidden, sigmoid output and binary cross-

$z_1^1$ = 0.25*1+0.75*1=1.0

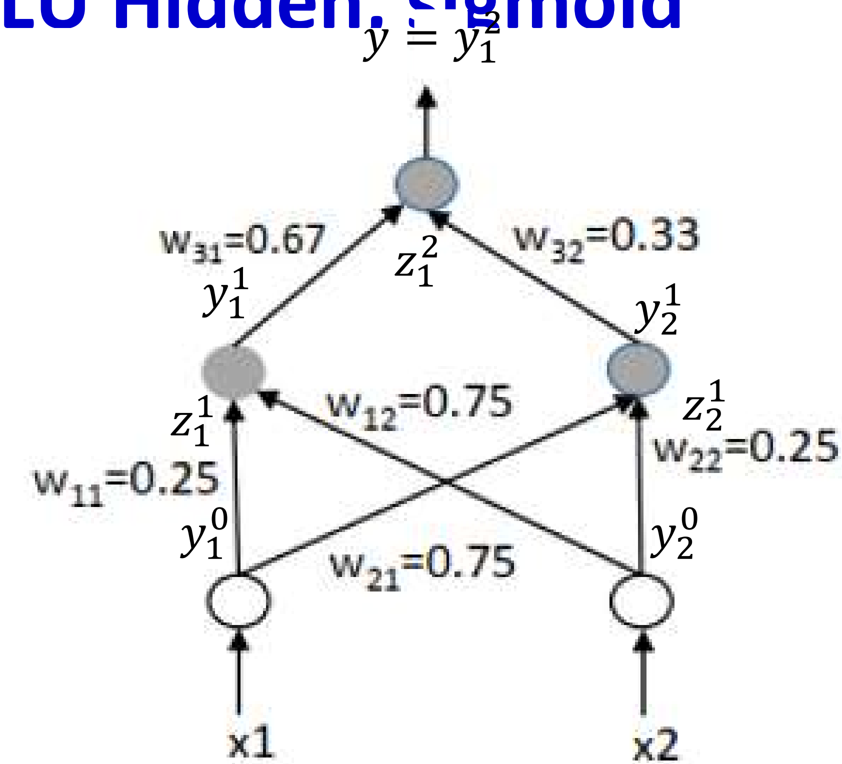
$z_2^1$ = 0.75*1+0.25*1=1.0

$y_1^1$ = ReLU($z_1^1$)= 1   $y_2^1$ = ReLU($z_2^1$)= 1

$z_1^2$ = 0.67*1+0.33*1=1.0

$y_1^2$ = sigmoid($z_1^2$)=e/(1+e)

div = -d*log(y)-(1-d)*log(1-y)



# Weight Updates with ReLU Hidden, sigmoid output and binary cross-

$z_1^1$ = 0.25*1+0.75*1=1.0

$z_2^1$ = 0.75*1+0.25*1=1.0

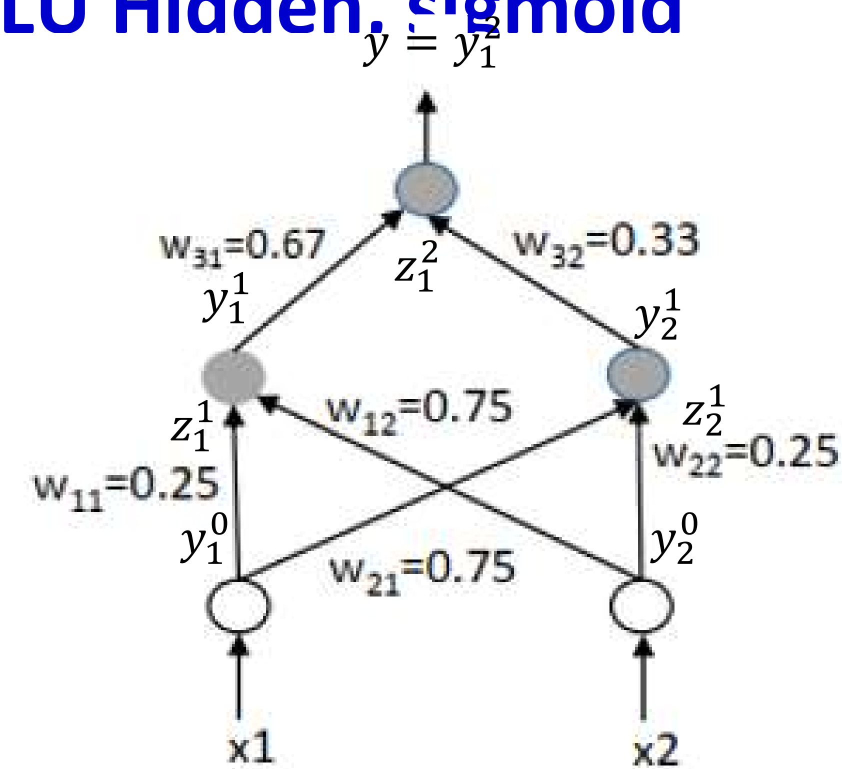
$y_1^1$ = ReLU($z_1^1$)= 1   $y_2^1$ = ReLU($z_2^1$)= 1

$z_1^2$ = 0.67*1+0.33*1=1.0

$y_1^2$ = sigmoid($z_1^2$)=e/(1+e)

div = -d*log(y)-(1-d)*log(1-y)

$\frac{\delta div}{\delta y}$=-d/y+(1-d)/(1-y)=1/(1-y)= (1+e)

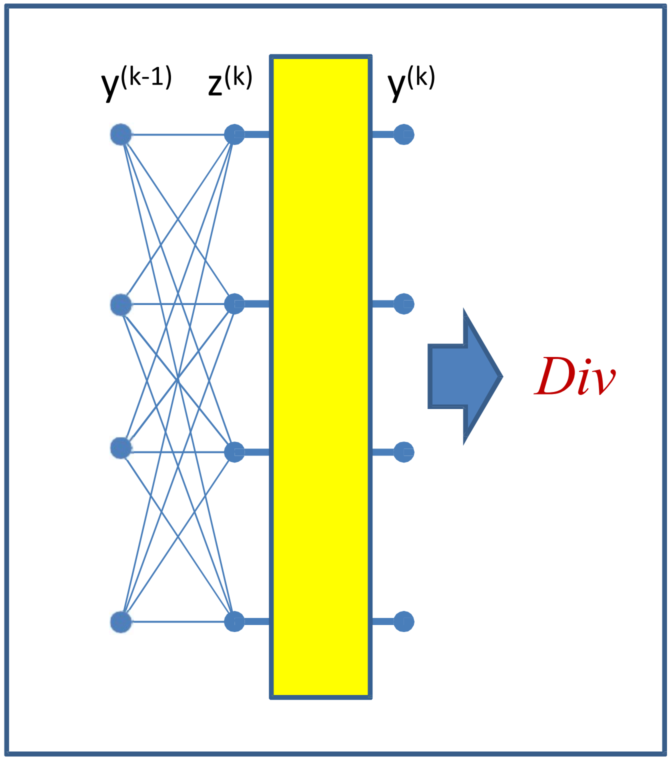



$\frac{\delta div}{\delta z_1^2} = \frac{\delta Div}{\delta y_1^2} * \frac{\delta y_1^2}{\delta z_1^2}$ = (1+e)*sigmoid($z_1^2$)*(1-sigmoid($z_1^2$))=e/(1+e)

$\frac{\delta div}{\delta w_{31}} = \frac{\delta Div}{\delta z_1^2} * \frac{\delta z_1^2}{\delta w_{31}} = \mathrm{e}/(1+\mathrm{e}) * y_1^1$=e/(1+e)

$\boxed{\boldsymbol{w_{31} = w_{31} - \eta^* \frac{\delta div}{\delta w_{31}}} \textbf{=0.67-0.1*e/(1+e)=0.597}}$

# Example Special Case : Softmax




$$y_i^{(k)} = \frac{exp\left(z_i^{(k)}\right)}{\sum_j exp\left(z_j^{(k)}\right)}$$

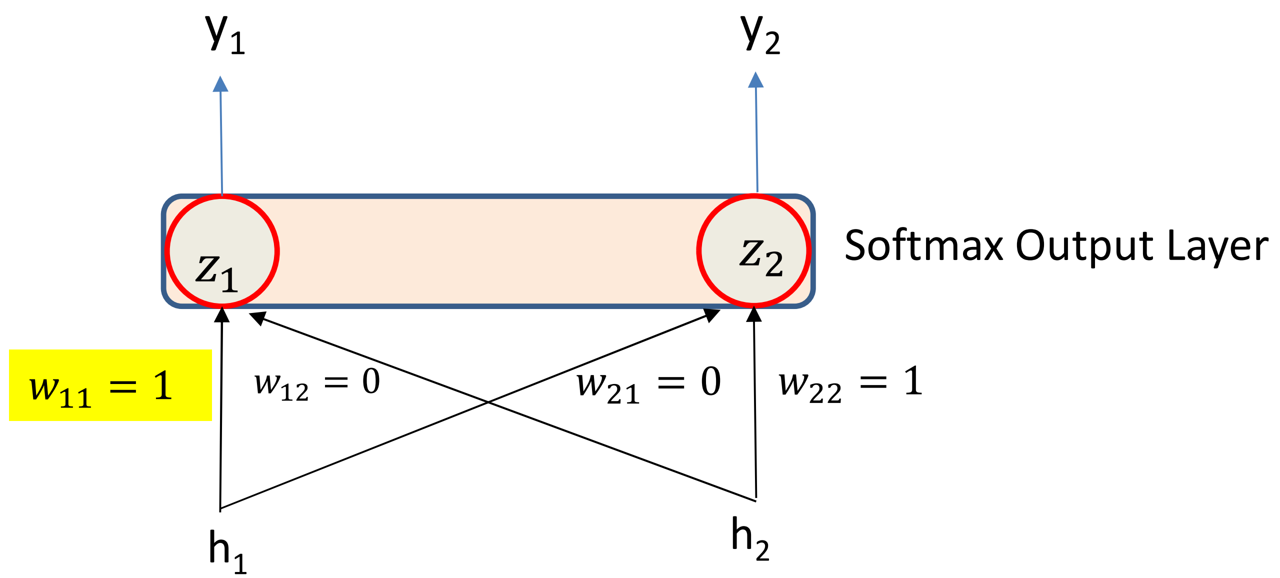
$$\frac{\partial Div}{\partial z_i^{(k)}} = \sum_j \frac{\partial Div}{\partial y_j^{(k)}} \frac{\partial y_j^{(k)}}{\partial z_i^{(k)}}$$

$$\frac{\partial y_j^{(k)}}{\partial z_i^{(k)}} = \begin{cases} y_i^{(k)}\left(1 - y_i^{(k)}\right) & \text{if } i = j \\ -\, y_i^{(k)} y_j^{(k)} & \text{if } i \neq j \end{cases}$$

$$\frac{\partial Div}{\partial z_i^{(k)}} = \sum_j \frac{\partial Div}{\partial y_j^{(k)}} y_i^{(k)} \left(\delta_{ij} - y_j^{(k)}\right)$$

- For future reference
- $\delta_{ij}$ is the Kronecker delta: $\delta_{ij} = 1$ if $i = j$, $\;0$ if $i \neq j$

# Weight Updates with Softmax




- Training input: $(h_1, h_2) = (1, 0)$ target output: $(d_1, d_2) = (0, 1)$
- *div* function: cross-entropy
- Learning rate: 0.1
- Bias = 0
- What will be the value of $w_{11}$ in next iteration?

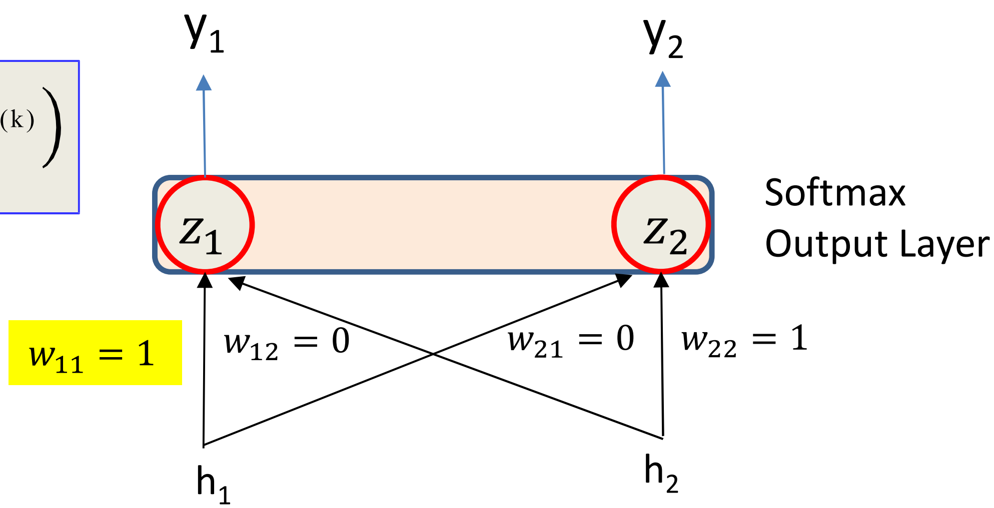
- Input to softmax node $z_1 = w_{11} = 1$; $z_2 = 0$
- $y_1 = e / (1+e)$
- $y_2 = 1 / (1+e)$
- *div* $= -d_1 \log(y_1) - d_2 \log(y_2) = -\log(y_2) = \log(1+e) = 1.313$

# Weight Updates with Softmax …

$$\frac{\partial Div}{\partial z_i^{(k)}} = \sum_j \frac{\partial Div}{\partial y_j^{(k)}} y_i^{(k)} \left(\delta_{ij} - y_j^{(k)}\right)$$

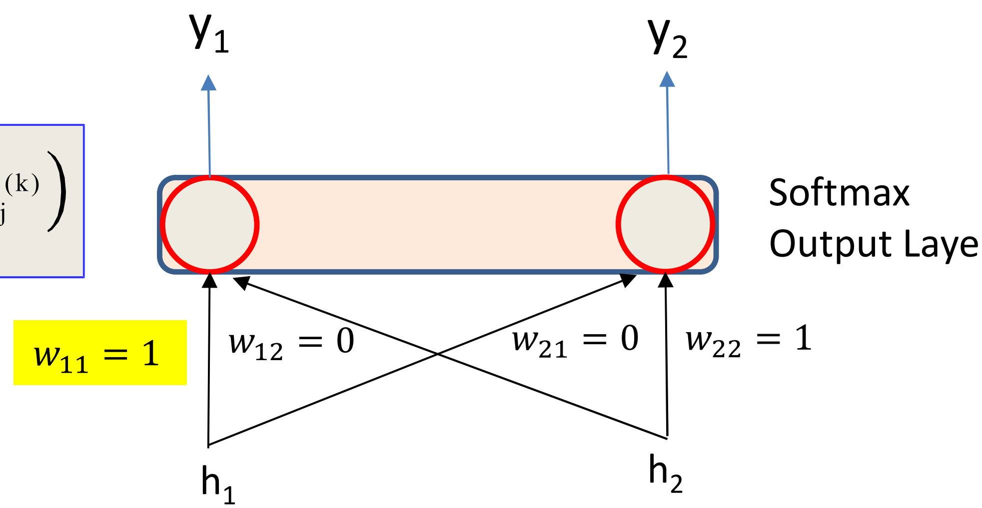
$$w_{11}(\mathrm{t}+1) = ?$$

Training input: $(h_1, h_2) = (1, 0)$
Target output: $(d_1, d_2) = (0, 1)$



# Weight Updates with Softmax …

$$\frac{\partial Div}{\partial z_i^{(k)}} = \sum_j \frac{\partial Div}{\partial y_j^{(k)}} y_i^{(k)} \left(\delta_{ij} - y_j^{(k)}\right)$$

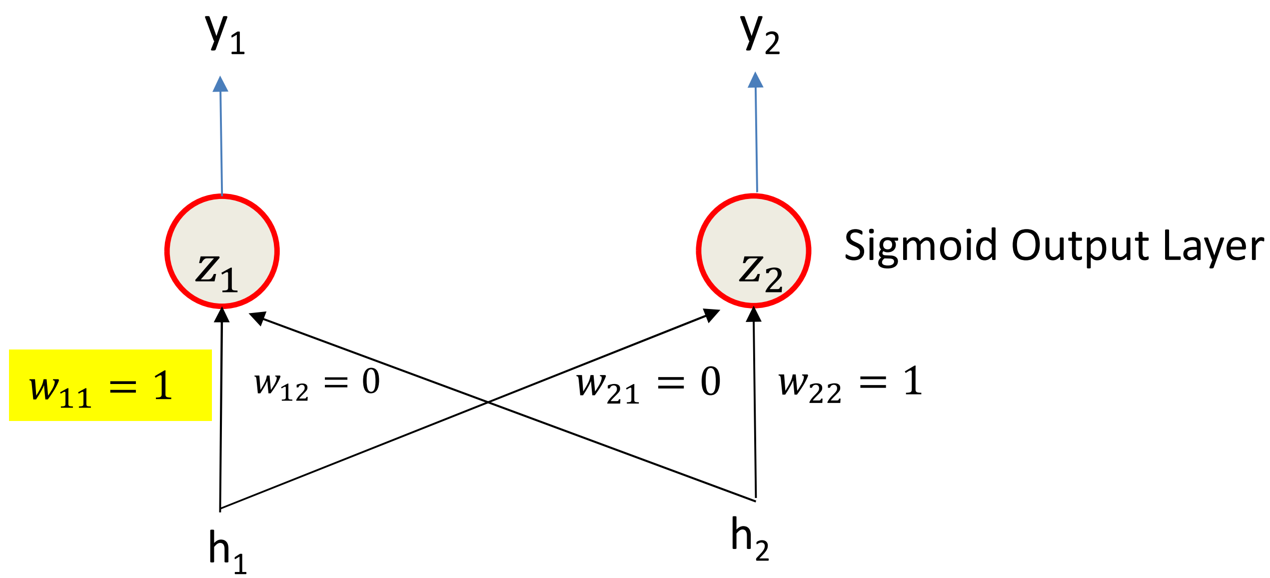



- Change in $w_{11}$ = $\delta w_{11}$ = -0.1 * d*div*/d$w_{11}$
  = -0.1 * d*div*/d$z_1$ * d$z_1$/d$w_{11}$ = -0.1 * d*div*/d$z_1$ * $h_1$

- d*div*/d$y_1$ = -$d_1$/$y_1$ = 0 d*div*/d$y_2$=-$d_2$/$y_2$ = -1/$y_2$
- d*div*/d$z_1$ = d*div*/d$y_1$*$y_1$(1-$y_1$) + d*div*/d$y_2$*$y_1$(-$y_2$) = - $y_1$ $y_2$ / $y_2$ = -$y_1$

  - Input to softmax node $z_1$ = $w_{11}$=1; $z_2$ =0 ; $y_1$ = e / (1+e)
  - $\delta w_{11}$ = $-0.1$ * e/(1+e) = -0.0731

# Weight Updates with Sigmoid Output



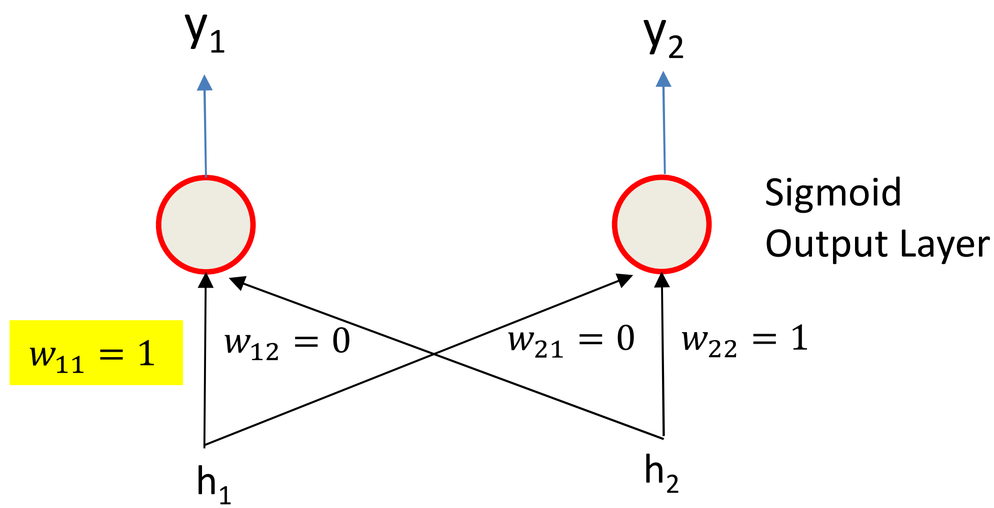

- Training input: $(h_1, h_2) = (1, 0)$ target output: $(d_1,d_2)=(0,1)$
- Activation: sigmoid, Bias=0
- *div* function: cross-entropy
- Learning rate: 0.1
- What will be the value of $w_{11}$ in next iteration?

- Input to output node $z_1 = w_{11}=1$; $z_2 =0$
- $y_1 = 1 / (1+e^{-1}) =e/(1+e)$ $y_2 = 1/ (1+e^0)=1/2$
- *div* $= -d_1 \log(y_1) – d_2 \log(y_2) = -\log(y_2) = \log 2$

# Weight Updates with Sigmoid Outputs …

$w_{11}$(t+1)=?

Input (h1, h2)=(1.0, 0.0)
target output: ($d_1$,$d_2$)=(0,1)

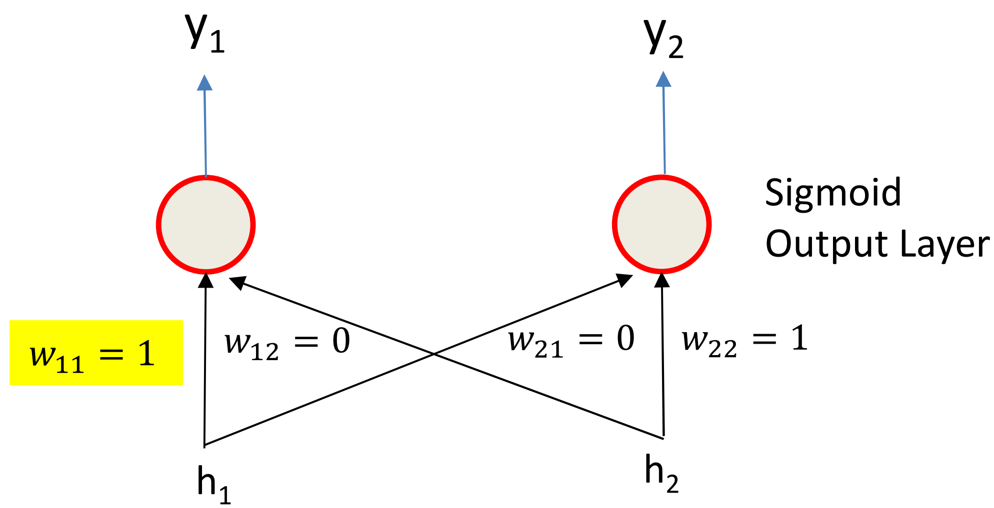


# Weight Updates with Sigmoid Outputs …

Input (h1, h2)=(1.0, 0.0)
target output: $(d_1,d_2)=(0,1)$

$w_{11}(t+1)=?$

$div = -d_1 \log(y_1) - d_2 \log(y_2)$




$$\frac{\delta div}{\delta y_1} = -\frac{d_1}{y_1}$$

$$\frac{\delta y_1}{\delta z_1} = \frac{1}{1+e^{-1}} * \left(1 - \frac{1}{1+e^{-1}}\right) = \frac{e}{(1+e)^2}$$

$$w_{11}(t+1) = w_{11}(t) - \eta \frac{\delta div}{\delta w_{11}}$$

$$= w_{11}(t) - 0.1 * \frac{\delta div}{\delta y_1} \frac{\delta y_1}{\delta z_1} \frac{\delta z_1}{\delta w_{11}}$$

$$= w_{11}(t) - 0.1 * 0 * \frac{e}{(1+e)^2} * h_1$$

$$= 1.0$$

# L2 Regularization

Loss$(x,y)=3x^2+6xy+y^2$
What is the value of ($x_{opt}$ , $y_{opt}$) that minimizes the loss, if L2 regularization constant 0.1 is applied?

$$\theta_R^* \approx (H + 0.1\, I)^{-1} H\theta^*$$

Now, for regularized case, optimal $\theta_R^*$ is given by

where H is the Hessian of Loss(x,y) and $\theta^*$is the optimal value in the unregularized case.

# L2 Regularization

Loss($x,y$)$=3x^2+6xy+y^2$ . What is the value of ($x_{opt}$ , $y_{opt}$) that minimizes the loss, if L2 regularization constant 0.1 is applied?

d Loss / dx = 6x + 6y = 0 and d Loss / dy = 6x+2y = 0 at ($x_{opt}$ , $y_{opt}$). Solving these two equations, we get, in unregularized case, $x_{opt}$ = $y_{opt}$ = 0.

Now, for regularized case, optimal $\theta_R^*$ is given by

$$\theta_R^* \approx (H + 0.1\ I)^{-1} H\theta^*$$

where H is the Hessian of Loss(x,y) and $\theta^*$is the optimal value in the unregularized case.

Since in unregularized case, $x_{opt}$=$y_{opt}$=0, $\theta_R^*$is also a zero vector.

# L1 Regularization

$\text{Loss}(x, y) = 3x^2 - 6x + 3 + y^2$

What is the value of ($x_{opt}$ , $y_{opt}$) that minimizes the loss, if L1 regularization constant 0.1 is applied?

$$(\theta_R^*)_i \approx \begin{cases} \max\left\{\theta^* - \frac{\alpha}{H_{ii}}, 0\right\} & \text{if } \theta^* \geq 0 \\ \min\left\{\theta^* + \frac{\alpha}{H_{ii}}, 0\right\} & \text{if } \theta^* < 0 \end{cases}$$

So, in regularized case, $x_{opt}$ = 1 – 0.1/6 = 0.9833 and $y_{opt}$ = max (0-0.1/2,0)=0.

# L1 Regularization

Loss$(x, y) = 3x^2 - 6x + 3 + y^2$ . What is the value of ($\underline{x}_{opt}$ , $\underline{y}_{opt}$) that minimizes the loss, if L1 regularization constant 0.1 is applied?

$H_{11}$= $d^2$ Loss (x,y) / $dx^2$ = 6, $H_{12}$= $H_{21}$ = $d^2$ Loss (x,y) / dxdy = 0 and $H_{22}$= $d^2$ Loss (x,y) / $dy^2$ = 2 . So, H is diagonal and +ve definite. Hence

$$(\theta_R^*)_i \approx \begin{cases} \max\left\{\theta^* - \dfrac{\alpha}{H_{ii}}, 0\right\} & \text{if } \theta^* \geq 0 \\ \min\left\{\theta^* + \dfrac{\alpha}{H_{ii}}, 0\right\} & \text{if } \theta^* < 0 \end{cases}$$

d Loss(x,y)/dx = 6x -6 = 0, and d Loss(x,y)/dy = 2y=0 at unregularized ($x_{opt}$ , $y_{opt}$). Solving these we get, in unregularized case, $x_{opt}$ = 1 and $y_{opt}$ = 0.

So, in regularized case, $x_{opt}$ = max(1 – 0.1/6,0) = 0.9833 and $y_{opt}$ = max (0-0.1/2,0)=0.