CS365: Deep Learning

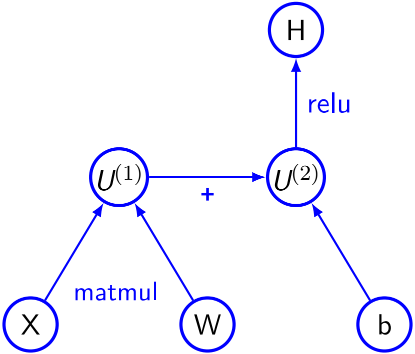
Backpropagation

Arijit Mondal
Dept. of Computer Science & Engineering
Indian Institute of Technology Patna
arijit@iitp.ac.in

Back propagation

- In a feedforward network, an input x is read and produces an output $\hat{\mathsf{y}}$
  - This is forward propagation
- During training forward propagation continues until it produces cost $J(\boldsymbol{\theta})$
- Back-propagation algorithm allows the information to flow backward in the network to compute the gradient
- Computation of analytical expression for gradient is easy
- We need to find out gradient of the cost function with respect to the parameters ie. $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

Computational graph

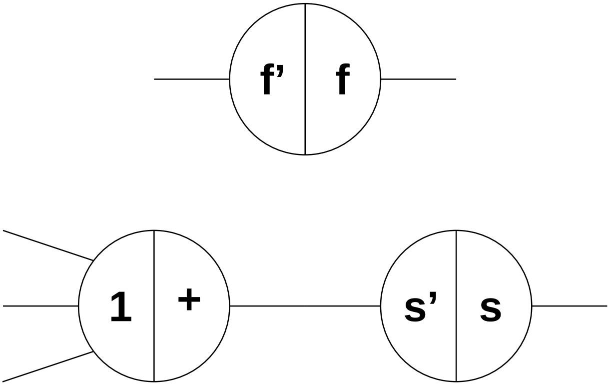


Chain rule of calculus

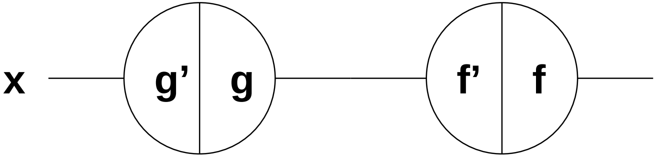
- Back-propagation algorithm heavily depends on it
- Let x be a real number and $y = g(x)$ and $z = f(g(x)) = f(y)$
- Chain rule says $\dfrac{dz}{dx} = \dfrac{dz}{dy}\dfrac{dy}{dx}$
- This can be generalized: Let $\mathrm{x} \in \mathbb{R}^m$, $\mathrm{y} \in \mathbb{R}^n$, $g : \mathbb{R}^m \to \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathrm{y} = g(\mathrm{x})$ and $z = f(\mathrm{y})$ then $\dfrac{\partial z}{\partial x_i} = \sum_j \dfrac{\partial z}{\partial y_j}\dfrac{\partial y_j}{\partial x_i}$
- In vector notation it will be where $\frac{\partial \mathrm{y}}{\partial \mathrm{x}}$ is the $n \times m$ Jacobian matrix of g

$$\nabla_{\mathrm{x}} z = \left(\frac{\partial \mathrm{y}}{\partial \mathrm{x}}\right)^T \nabla_{\mathrm{y}} z$$

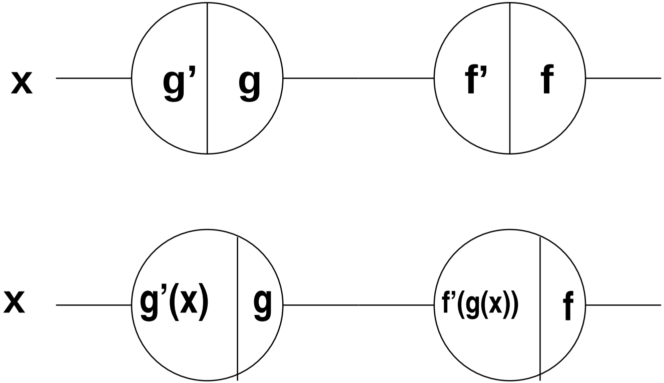
Back propagation - 1



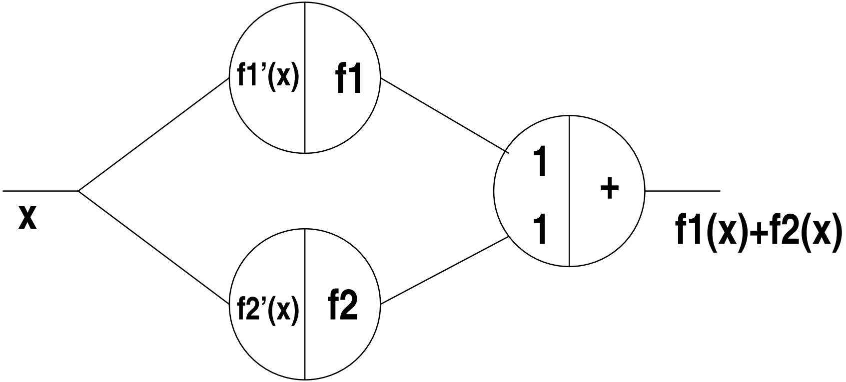
Back propagation - 2



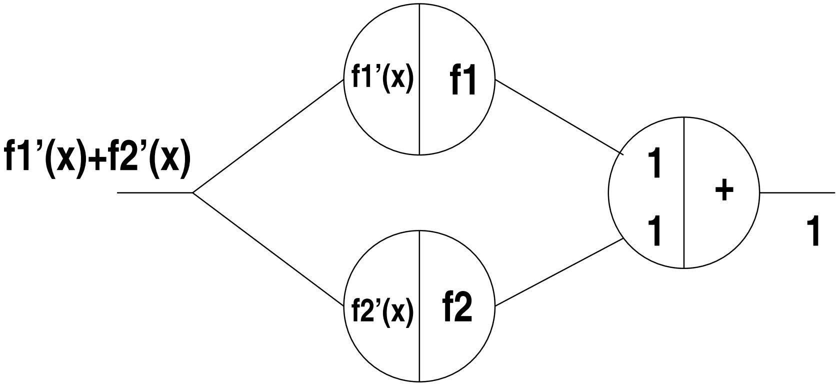
Back propagation - 3



Back propagation - 4



Back propagation - 5

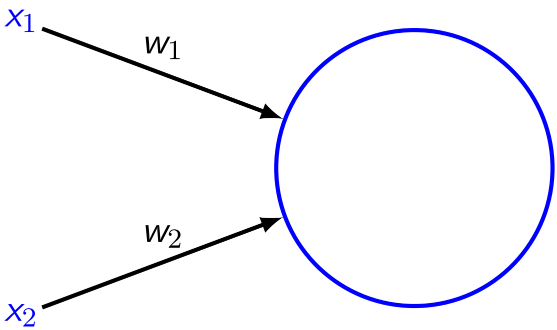


Back propagation - 6

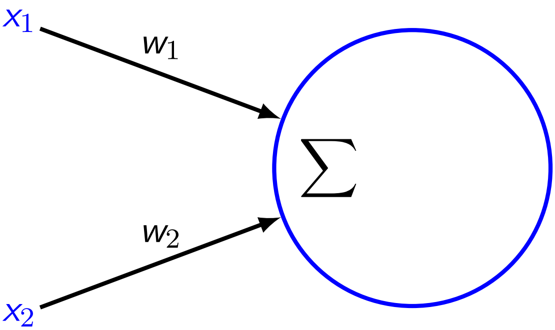
x_1

x_2

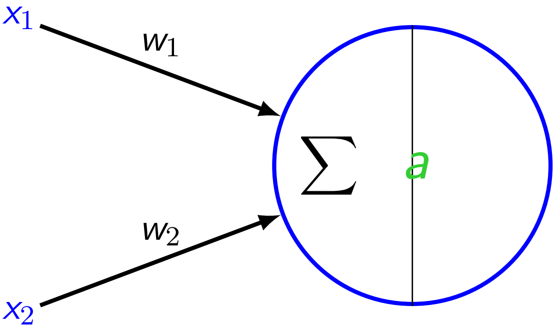
Back propagation - 6



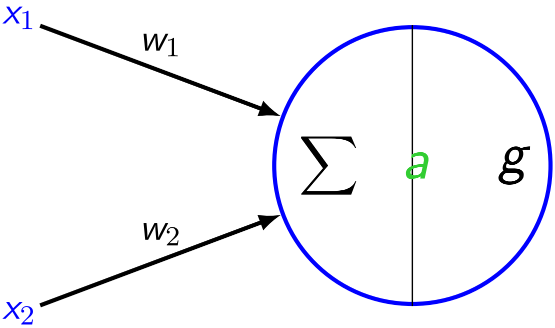
Back propagation - 6



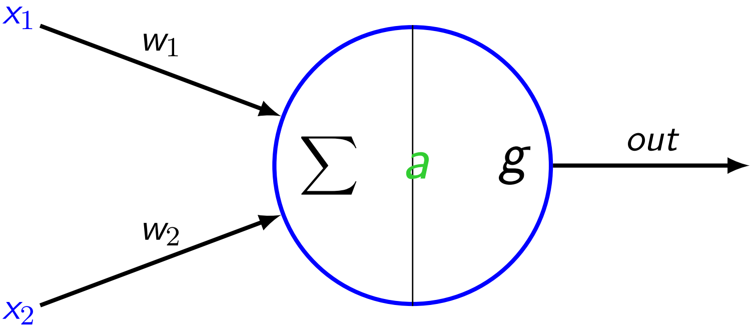
Back propagation - 6



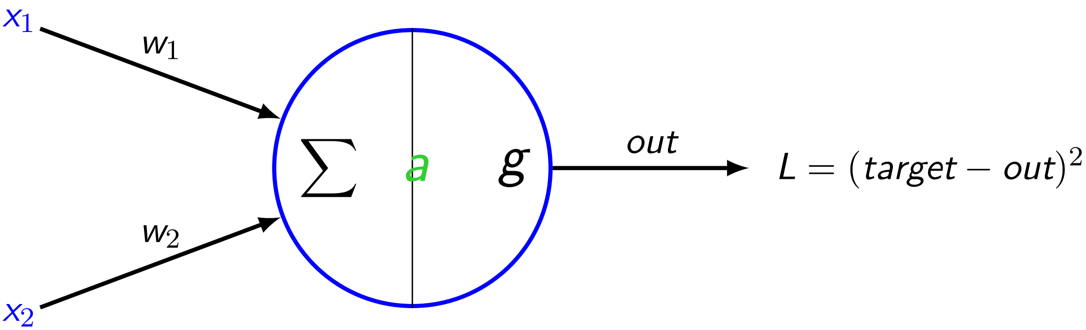
Back propagation - 6



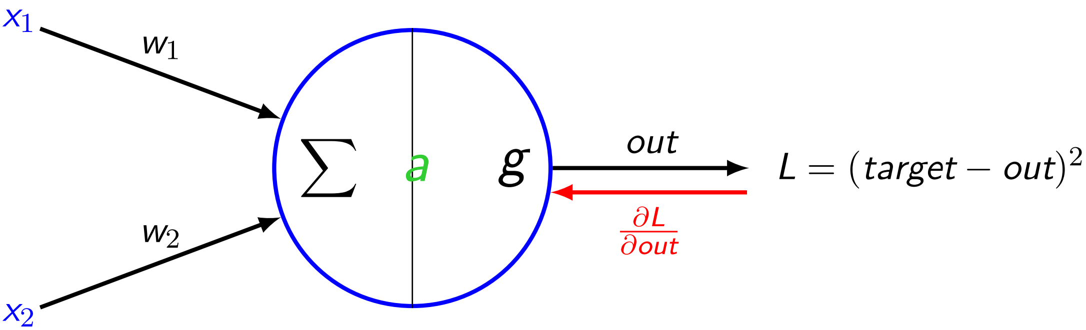
Back propagation - 6



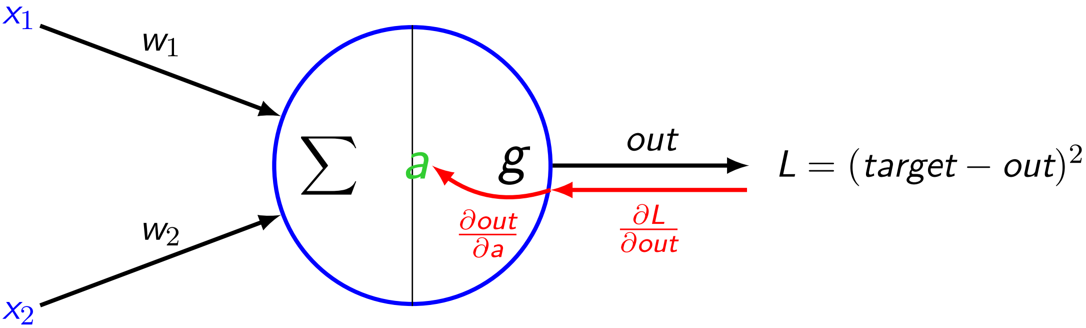
Back propagation - 6



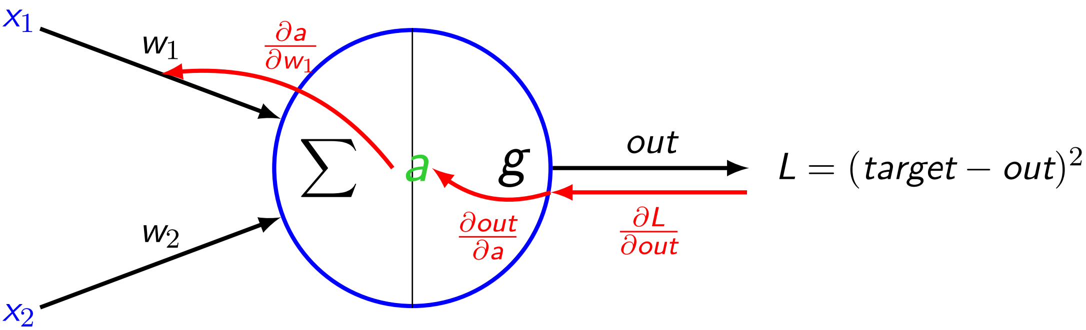
Back propagation - 6



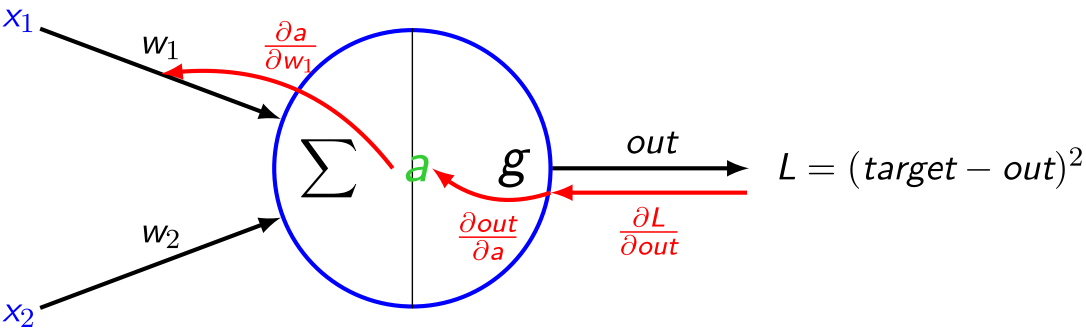
Back propagation - 6



Back propagation - 6



Back propagation - 6

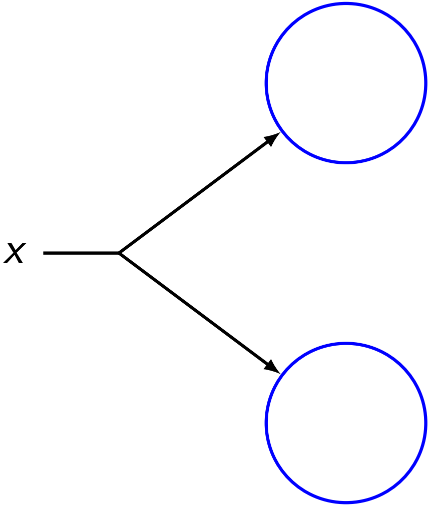


$$\boxed{\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial out}\frac{\partial out}{\partial a}\frac{\partial a}{\partial w_1}}$$

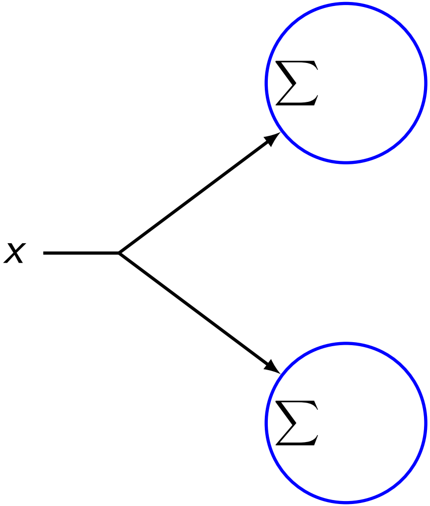
Back propagation - 7

x ——

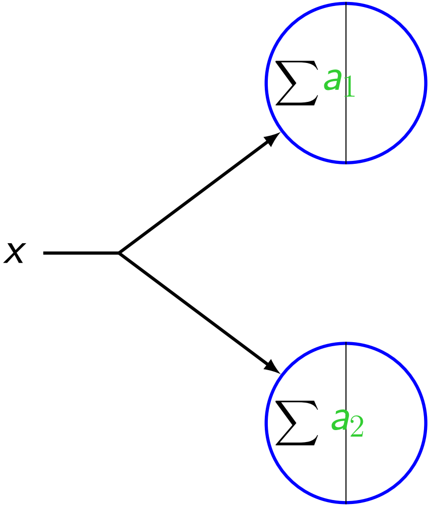
Back propagation - 7



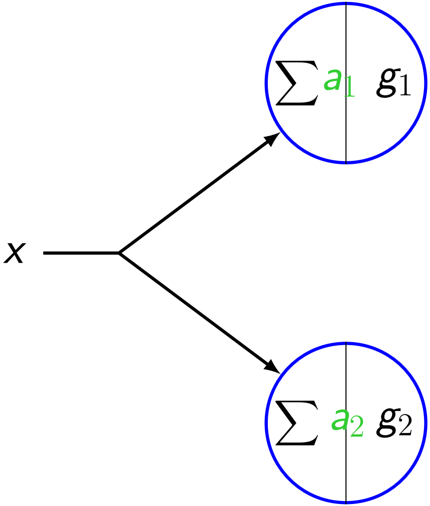
Back propagation - 7



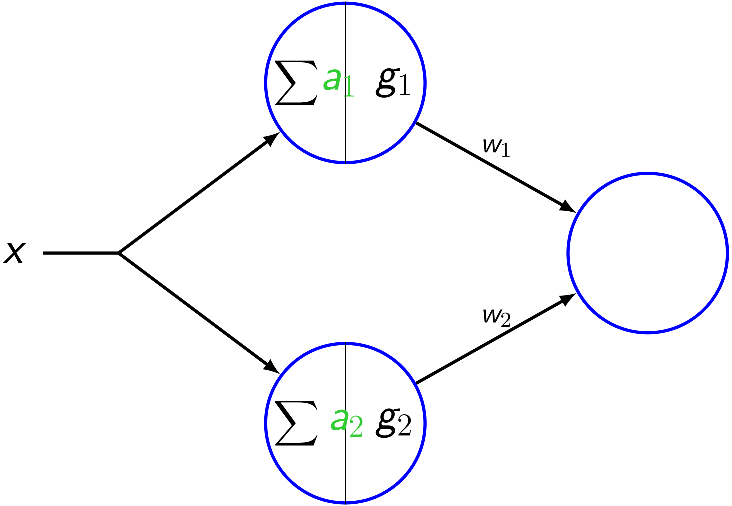
Back propagation - 7



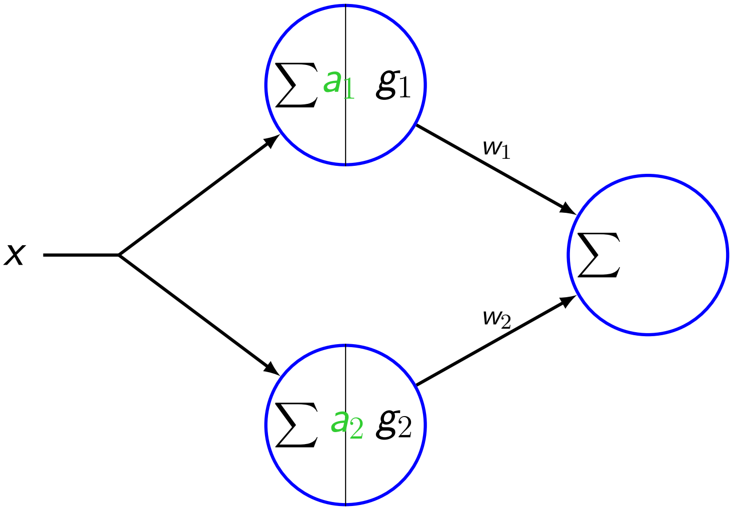
Back propagation - 7



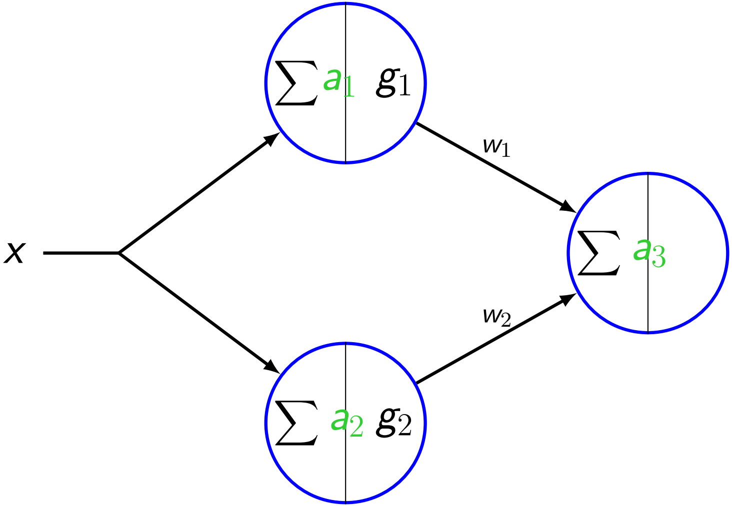
Back propagation - 7



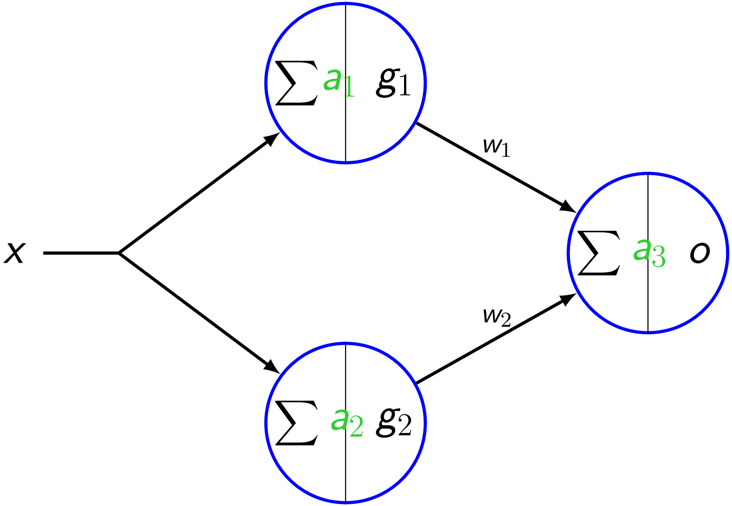
Back propagation - 7



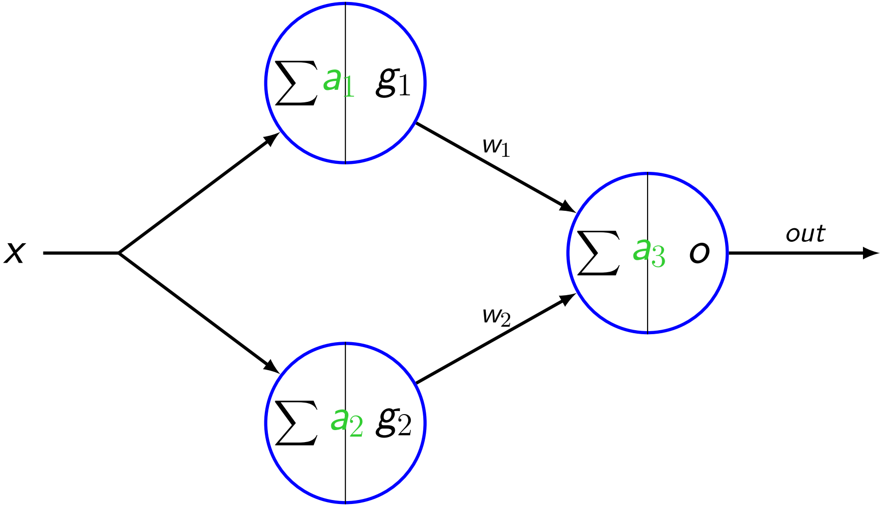
Back propagation - 7



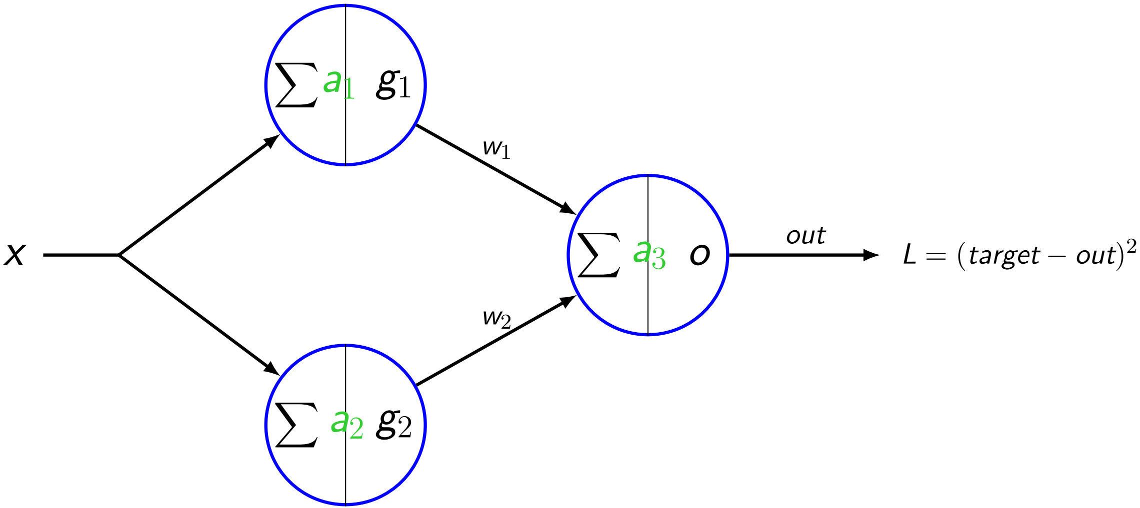
Back propagation - 7



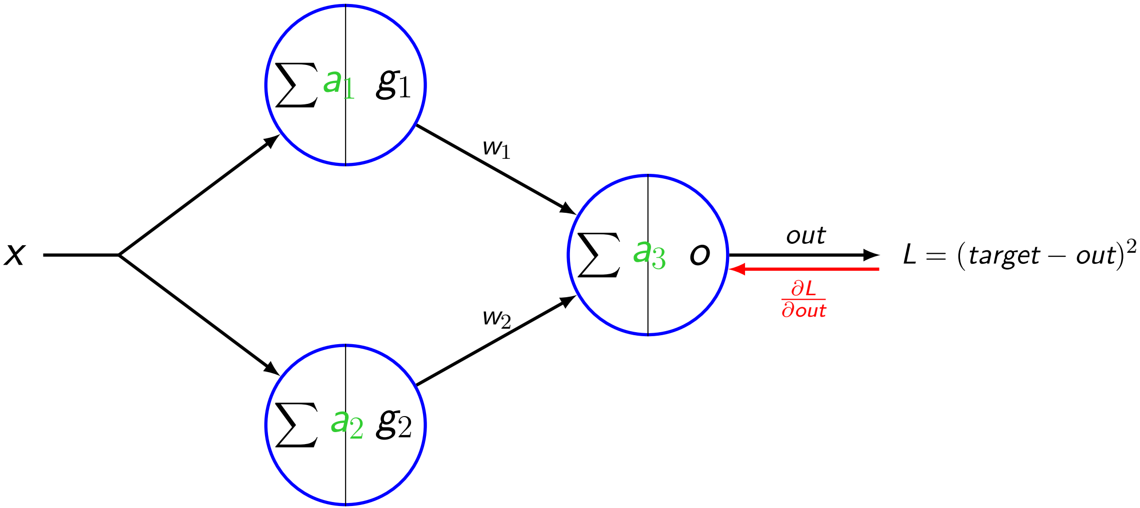
Back propagation - 7



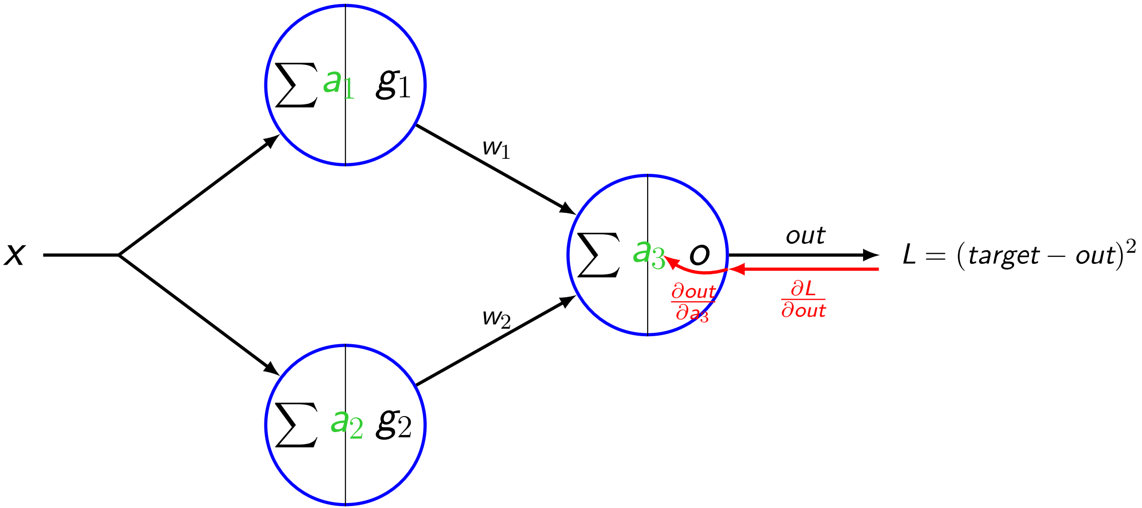
Back propagation - 7



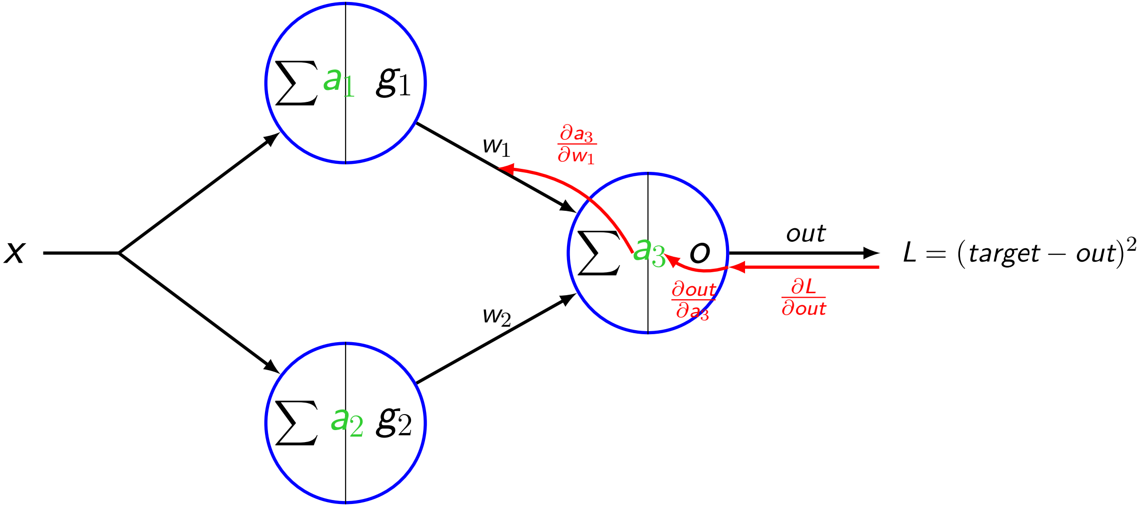
Back propagation - 7



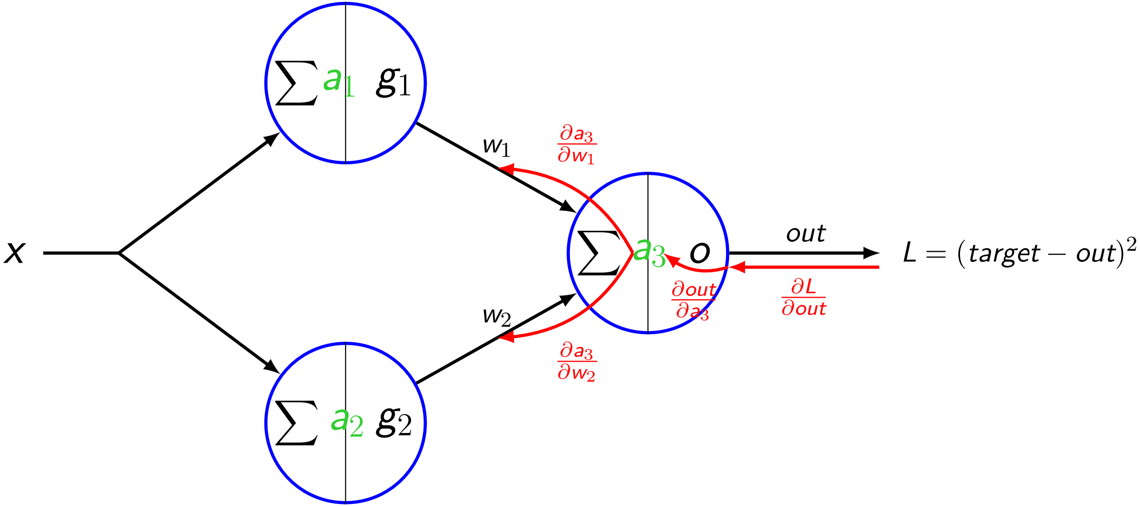
Back propagation - 7



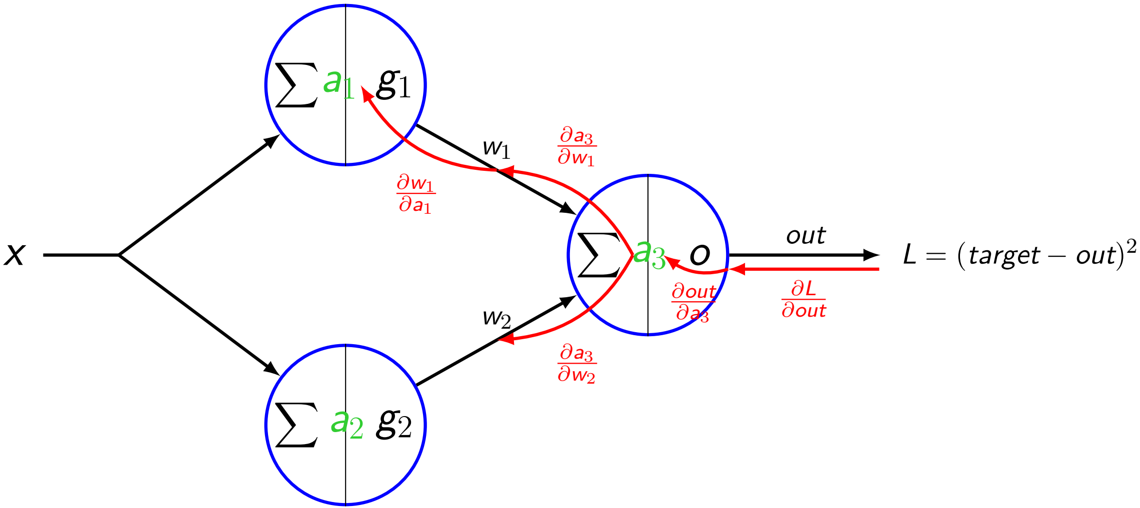
Back propagation - 7



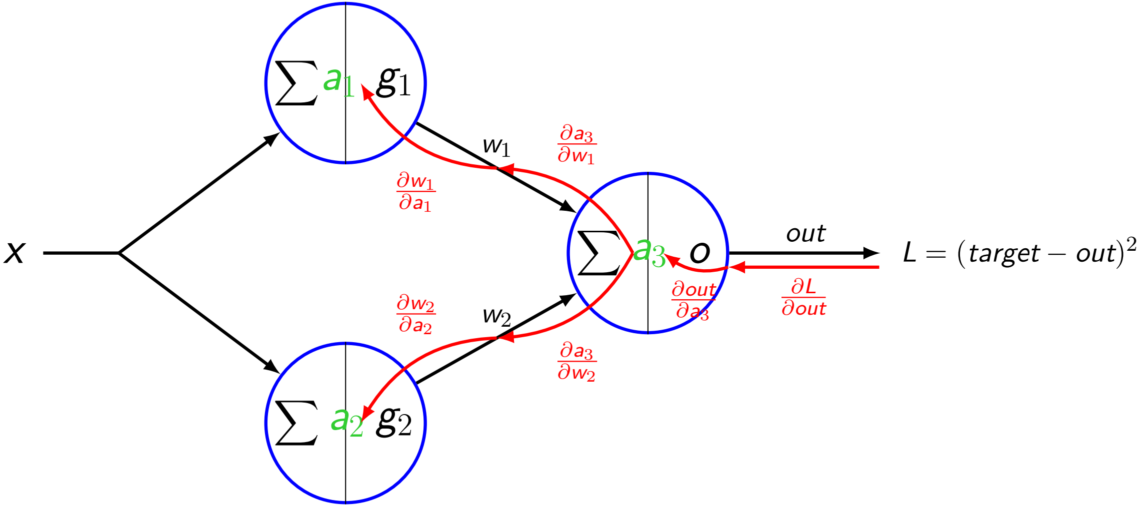
Back propagation - 7



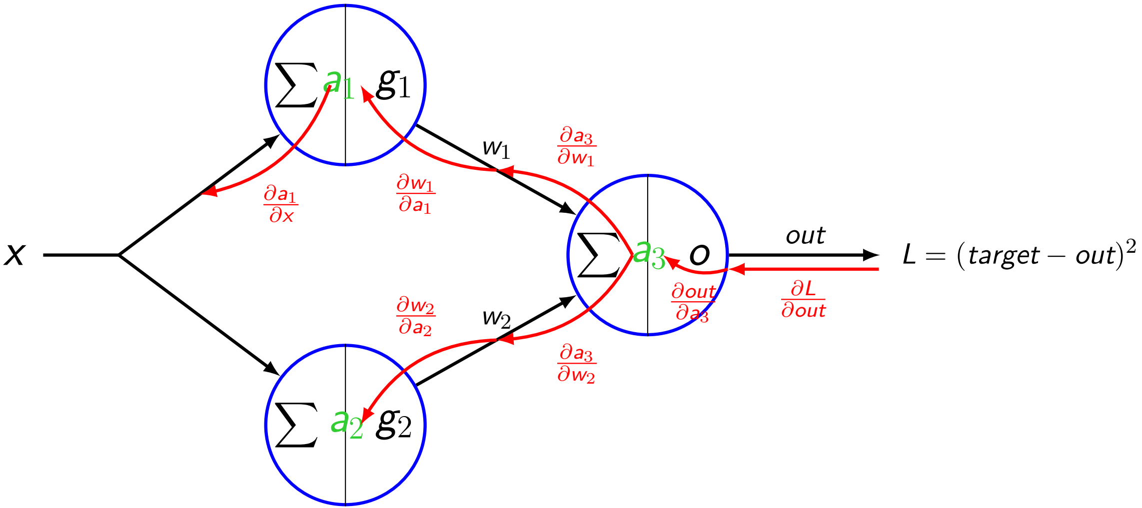
Back propagation - 7



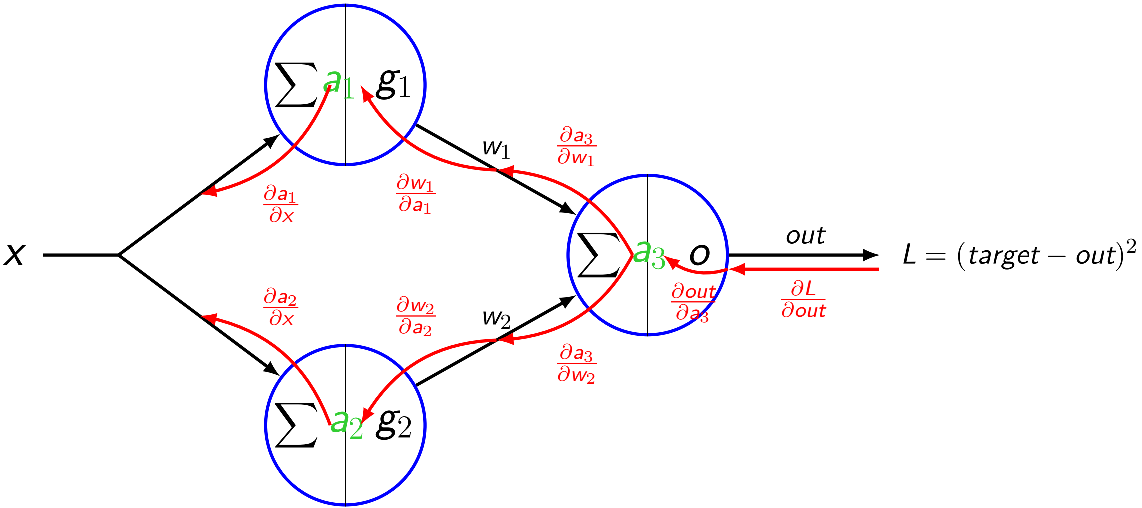
Back propagation - 7



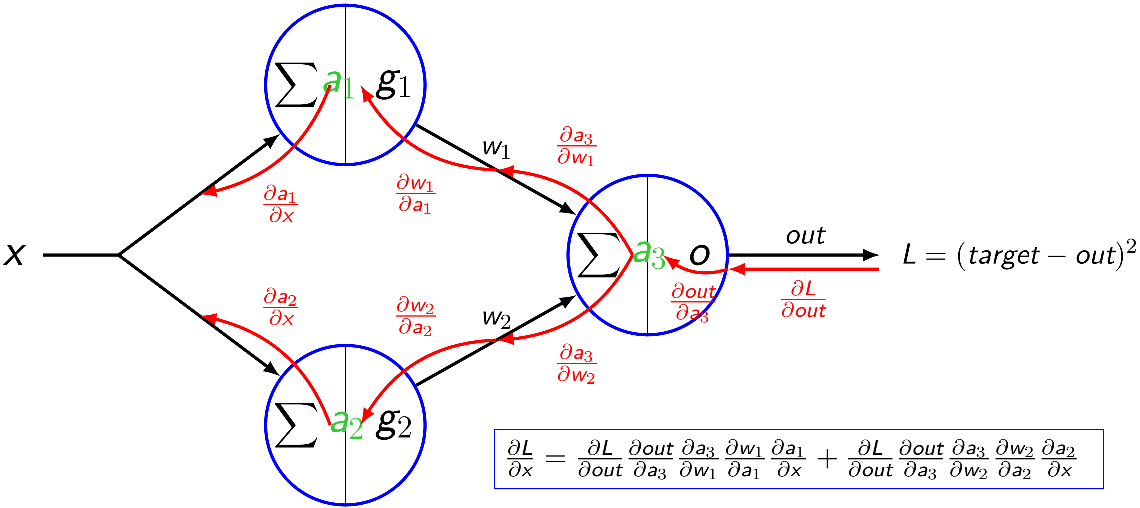
Back propagation - 7



Back propagation - 7

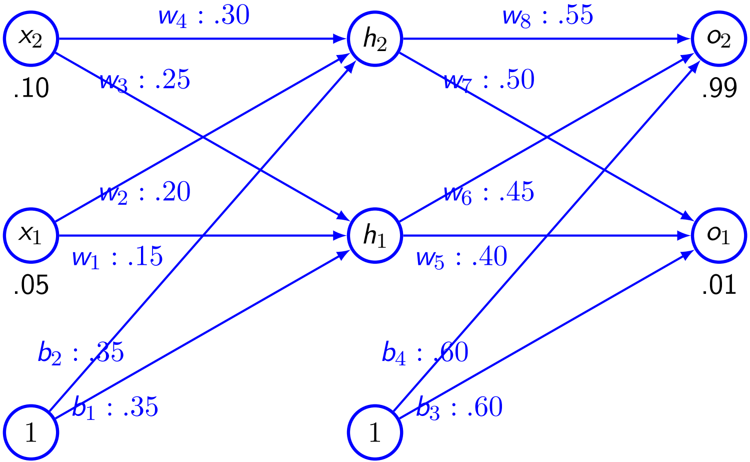


Back propagation - 7



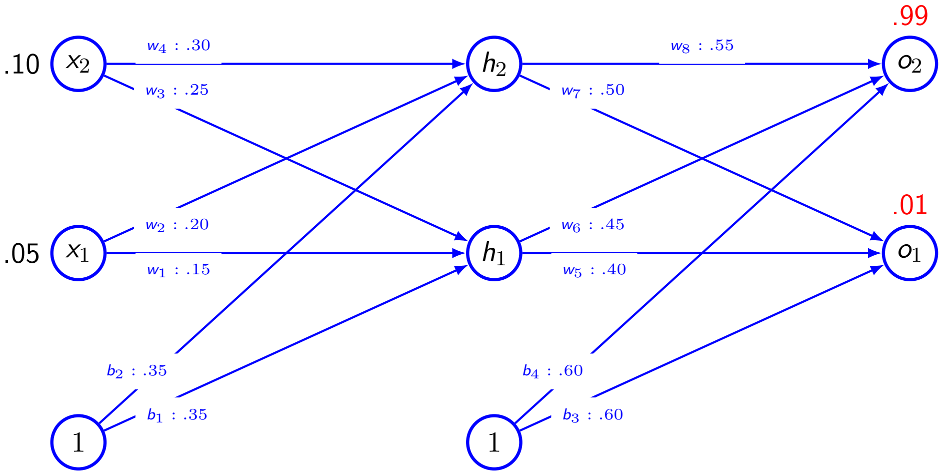
$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial out}\frac{\partial out}{\partial a_3}\frac{\partial a_3}{\partial w_1}\frac{\partial w_1}{\partial a_1}\frac{\partial a_1}{\partial x} + \frac{\partial L}{\partial out}\frac{\partial out}{\partial a_3}\frac{\partial a_3}{\partial w_2}\frac{\partial w_2}{\partial a_2}\frac{\partial a_2}{\partial x}$$

Example



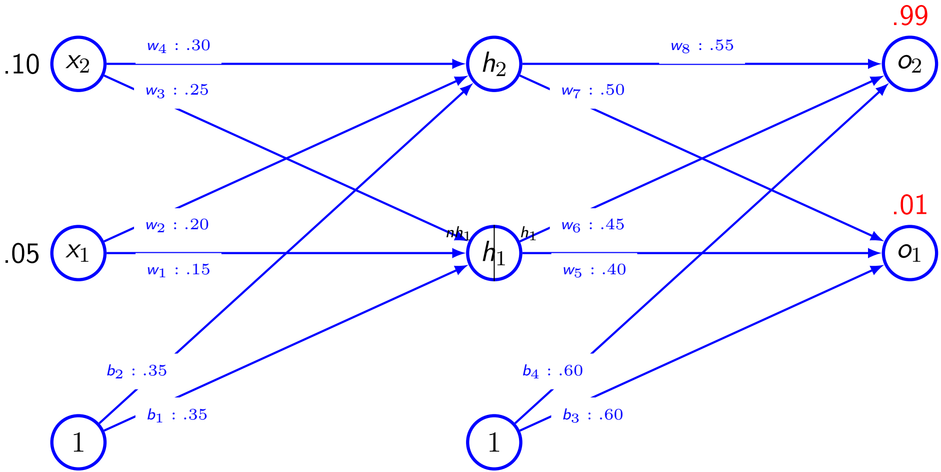
Hidden and output layer have sigmoid activation function. Loss function - MSE.

Example



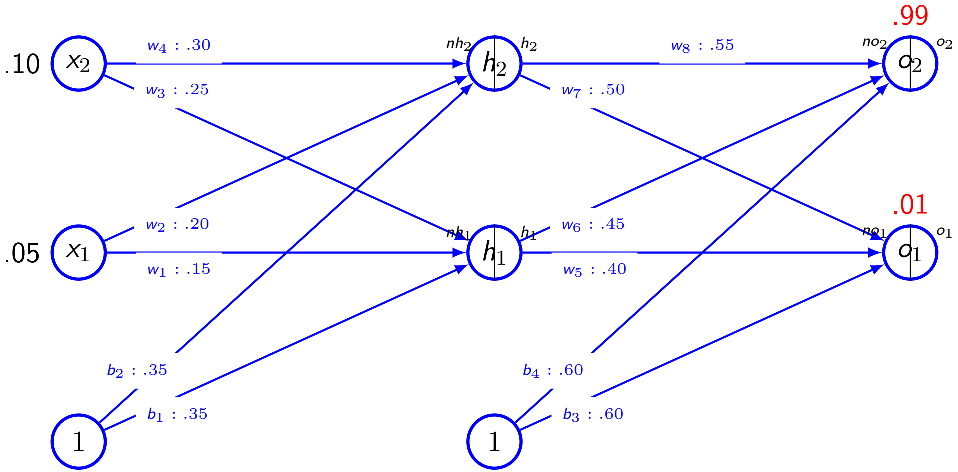
.10
x_2
.05
x_1
1
w_4 : .30
w_3 : .25
w_2 : .20
w_1 : .15
b_2 : .35
b_1 : .35
h_2
h_1
1
w_8 : .55
w_7 : .50
w_6 : .45
w_5 : .40
b_4 : .60
b_3 : .60
.99
o_2
.01
o_1

Example



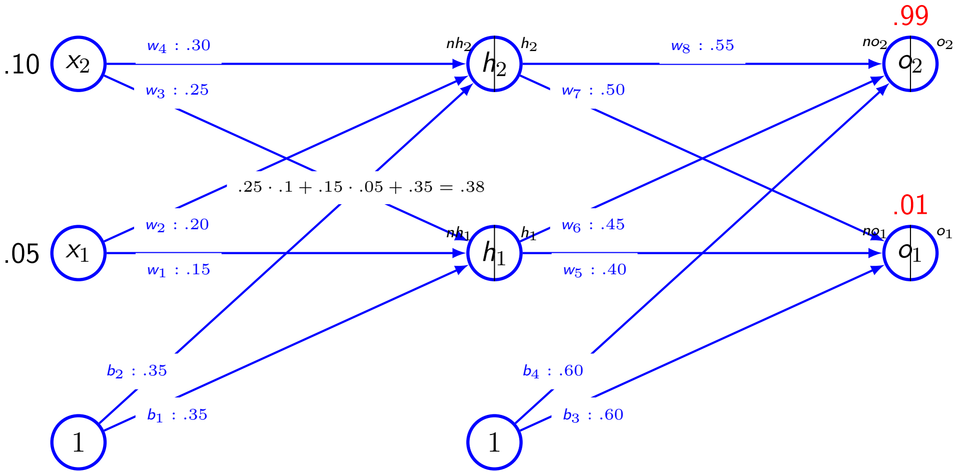
.99
.01
.10
.05
x_2
x_1
1
h_2
h_1
1
o_2
o_1
w_4 : .30
w_3 : .25
w_2 : .20
w_1 : .15
b_2 : .35
b_1 : .35
w_8 : .55
w_7 : .50
w_6 : .45
w_5 : .40
b_4 : .60
b_3 : .60

Example



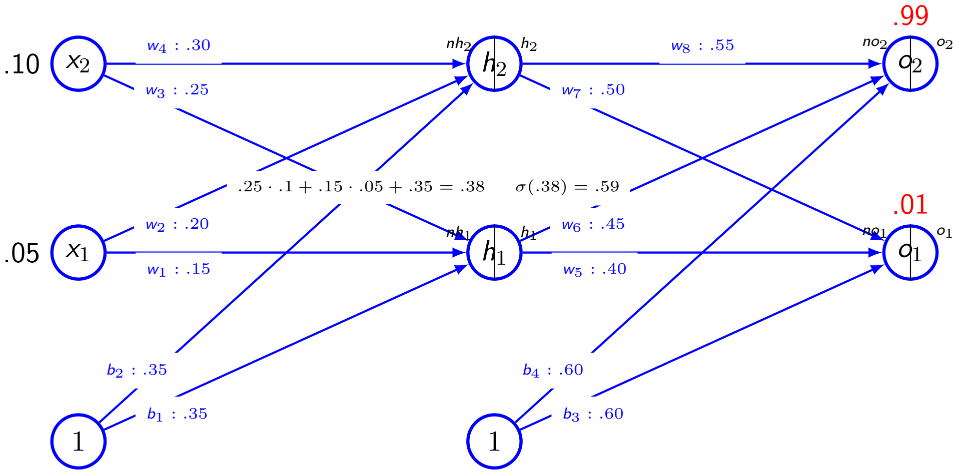
.99
.01
.10
.05
x_2
x_1
1
1
$w_4 : .30$
$w_3 : .25$
$w_2 : .20$
$w_1 : .15$
$b_2 : .35$
$b_1 : .35$
nh_2
h_2
nh_1
h_1
$w_8 : .55$
$w_7 : .50$
$w_6 : .45$
$w_5 : .40$
$b_4 : .60$
$b_3 : .60$
no_2
o_2
no_1
o_1

Example



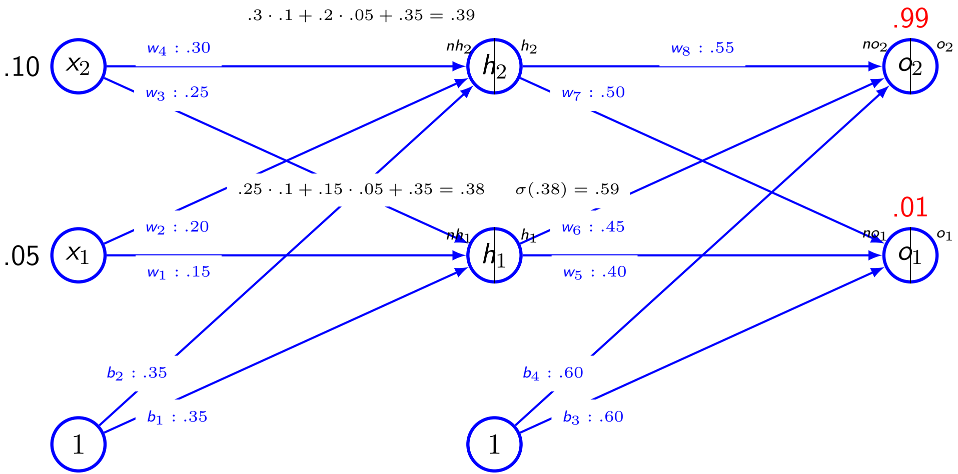
.99
.01
.10
.05
x_2
x_1
$w_4 : .30$
$w_3 : .25$
$w_2 : .20$
$w_1 : .15$
nh_2
h_2
nh_1
h_1
$w_8 : .55$
$w_7 : .50$
$w_6 : .45$
$w_5 : .40$
no_2
o_2
no_1
o_1
$.25 \cdot .1 + .15 \cdot .05 + .35 = .38$
$b_2 : .35$
$b_1 : .35$
$b_4 : .60$
$b_3 : .60$
1
1

Example



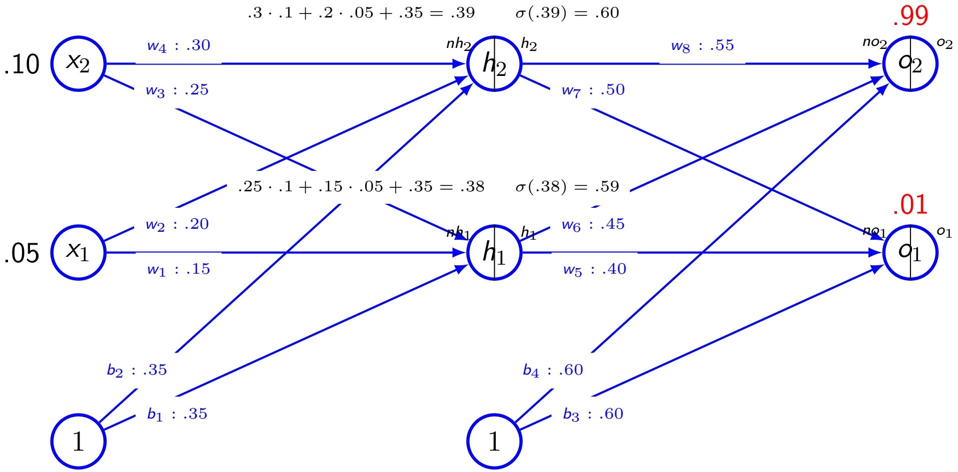
.99
.01
.10
.05
x_2
x_1
1
1
w_4 : .30
w_3 : .25
w_2 : .20
w_1 : .15
b_2 : .35
b_1 : .35
nh_2
h_2
nh_1
h_1
w_8 : .55
w_7 : .50
w_6 : .45
w_5 : .40
b_4 : .60
b_3 : .60
no_2
o_2
no_1
o_1
$.25 \cdot .1 + .15 \cdot .05 + .35 = .38$
$\sigma(.38) = .59$

Example

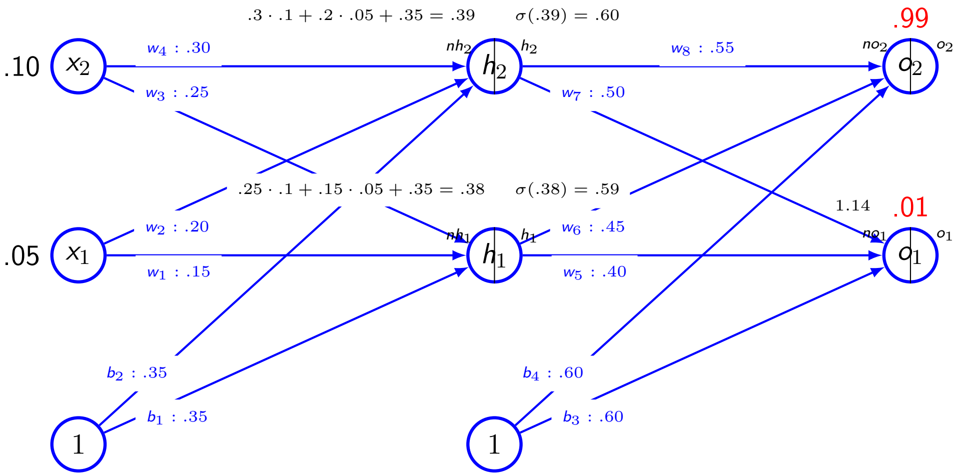


$.3 \cdot .1 + .2 \cdot .05 + .35 = .39$
.99
.01
.10
.05
x_2
x_1
$w_4 : .30$
$w_3 : .25$
$w_2 : .20$
$w_1 : .15$
nh_2
h_2
nh_1
h_1
$w_8 : .55$
$w_7 : .50$
$w_6 : .45$
$w_5 : .40$
no_2
o_2
no_1
o_1
$.25 \cdot .1 + .15 \cdot .05 + .35 = .38$
$\sigma(.38) = .59$
$b_2 : .35$
$b_1 : .35$
$b_4 : .60$
$b_3 : .60$
1
1

Example

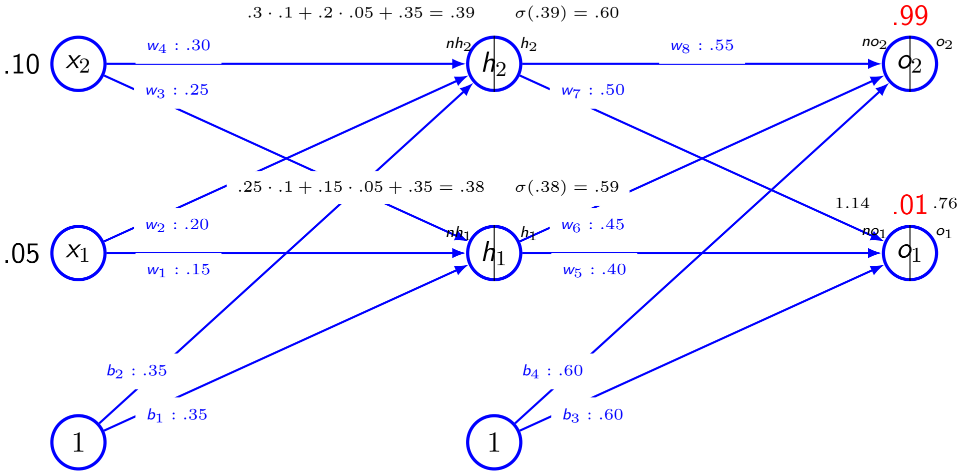

.3 · .1 + .2 · .05 + .35 = .39
σ(.39) = .60
.99
.10
x2
w4 : .30
nh2
h2
h2
w8 : .55
no2
o2
o2
w3 : .25
w7 : .50
.25 · .1 + .15 · .05 + .35 = .38
σ(.38) = .59
.01
.05
x1
w2 : .20
nh1
h1
h1
w6 : .45
no1
o1
o1
w1 : .15
w5 : .40
b2 : .35
b4 : .60
b1 : .35
b3 : .60
1
1

Example



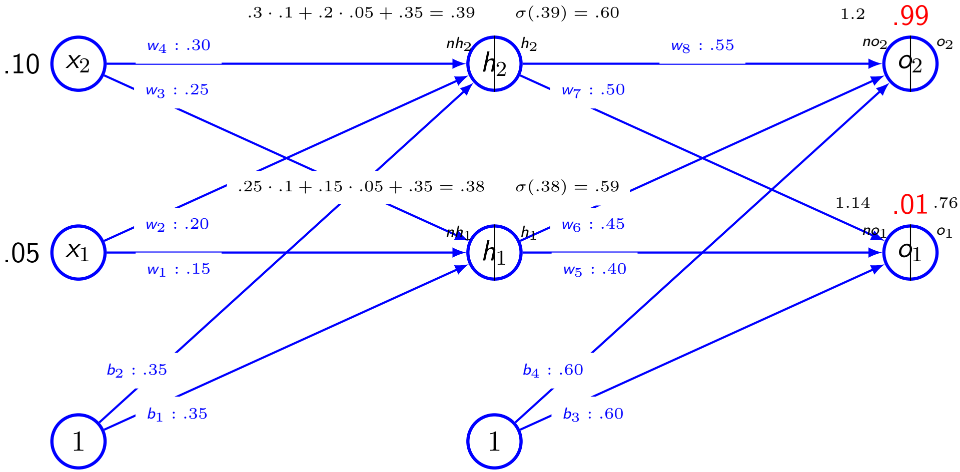
$.3 \cdot .1 + .2 \cdot .05 + .35 = .39$
$\sigma(.39) = .60$
.99
.10
x_2
$w_4 : .30$
nh_2
h_2
h_2
$w_8 : .55$
no_2
o_2
o_2
$w_3 : .25$
$w_7 : .50$
$.25 \cdot .1 + .15 \cdot .05 + .35 = .38$
$\sigma(.38) = .59$
1.14
.01
.05
x_1
$w_2 : .20$
nh_1
h_1
h_1
$w_6 : .45$
no_1
o_1
o_1
$w_1 : .15$
$w_5 : .40$
$b_2 : .35$
$b_4 : .60$
$b_1 : .35$
$b_3 : .60$
1
1

Example



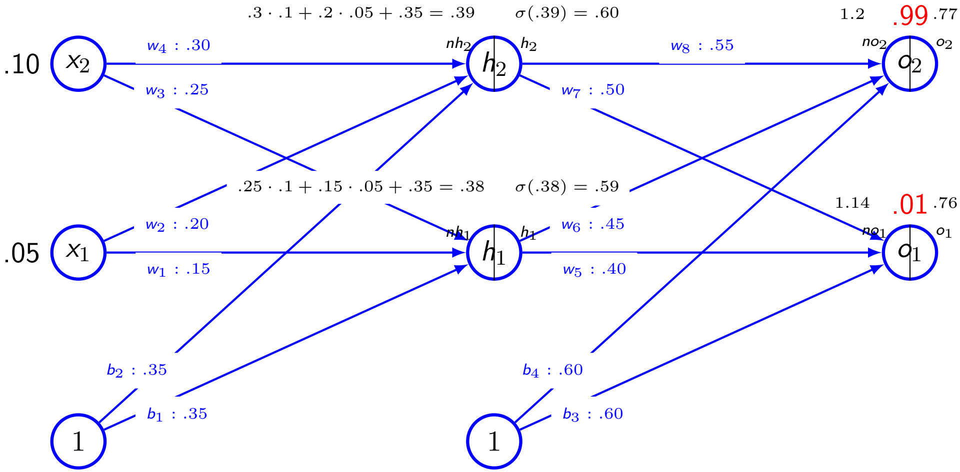
$.3 \cdot .1 + .2 \cdot .05 + .35 = .39$
$\sigma(.39) = .60$
.99
.10
x_2
$w_4 : .30$
nh_2
h_2
$w_8 : .55$
no_2
o_2
$w_3 : .25$
$w_7 : .50$
$.25 \cdot .1 + .15 \cdot .05 + .35 = .38$
$\sigma(.38) = .59$
1.14
.01
.76
$w_2 : .20$
$w_6 : .45$
nh_1
h_1
no_1
o_1
.05
x_1
$w_1 : .15$
$w_5 : .40$
$b_2 : .35$
$b_4 : .60$
$b_1 : .35$
$b_3 : .60$
1
1

Example

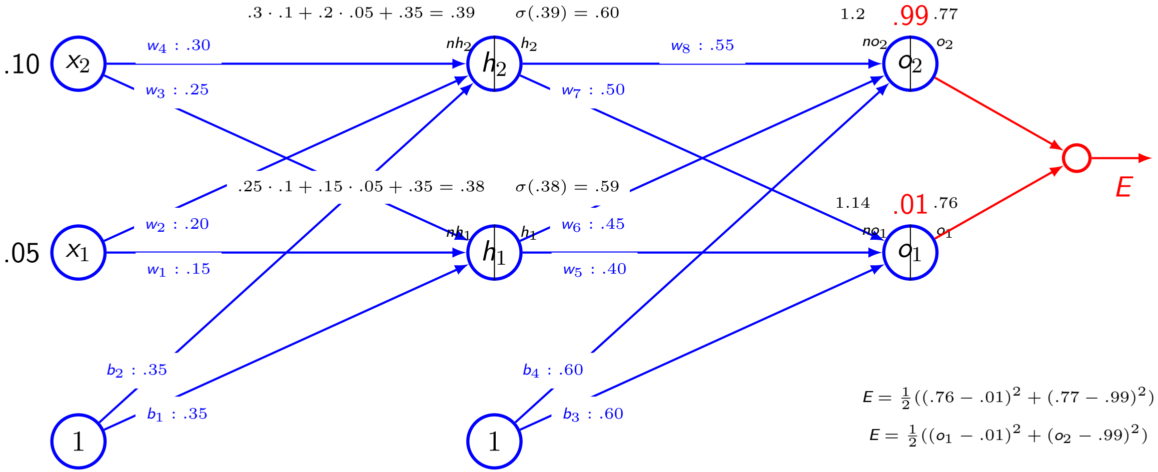


.3 · .1 + .2 · .05 + .35 = .39
σ(.39) = .60
1.2
.99
.10
x_2
w_4 : .30
nh_2
h_2
h_2
w_8 : .55
no_2
o_2
o_2
w_3 : .25
w_7 : .50
.25 · .1 + .15 · .05 + .35 = .38
σ(.38) = .59
1.14
.01
.76
w_2 : .20
w_6 : .45
nh_1
h_1
no_1
o_1
.05
x_1
h_1
o_1
w_1 : .15
w_5 : .40
b_2 : .35
b_4 : .60
b_1 : .35
b_3 : .60
1
1

Example


.3 · .1 + .2 · .05 + .35 = .39
σ(.39) = .60
1.2
.99
.77
.10
x2
w4 : .30
nh2
h2
h2
w8 : .55
no2
o2
o2
w3 : .25
w7 : .50
.25 · .1 + .15 · .05 + .35 = .38
σ(.38) = .59
1.14
.01
.76
w2 : .20
w6 : .45
.05
x1
nh1
h1
h1
no1
o1
o1
w1 : .15
w5 : .40
b2 : .35
b4 : .60
b1 : .35
b3 : .60
1
1

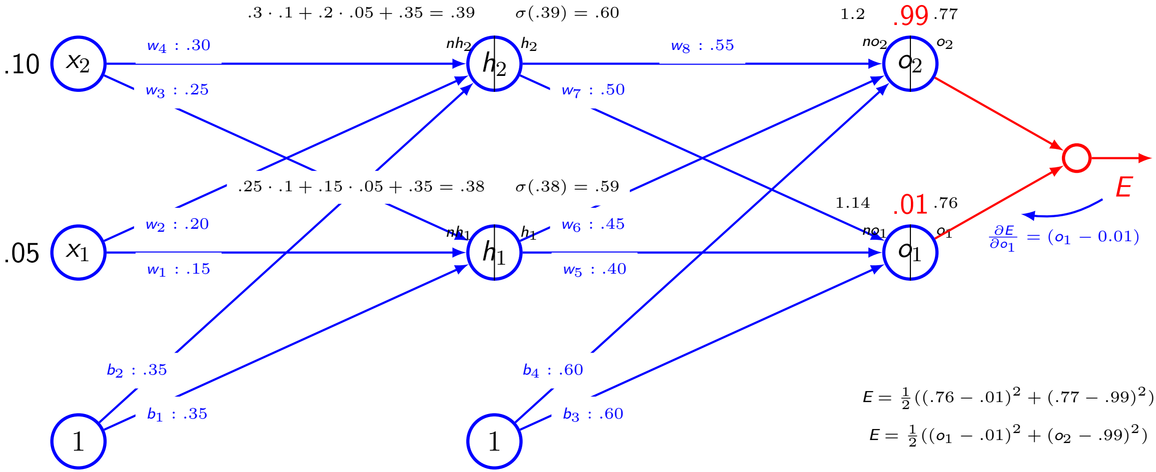
Example



$$E = \frac{1}{2}((.76 - .01)^2 + (.77 - .99)^2)$$

$$E = \frac{1}{2}((o_1 - .01)^2 + (o_2 - .99)^2)$$

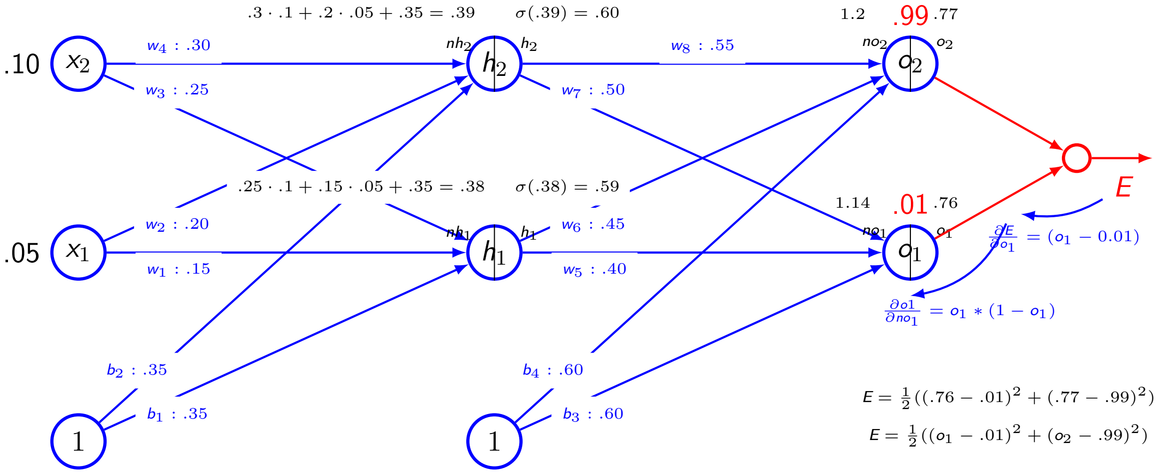
Example



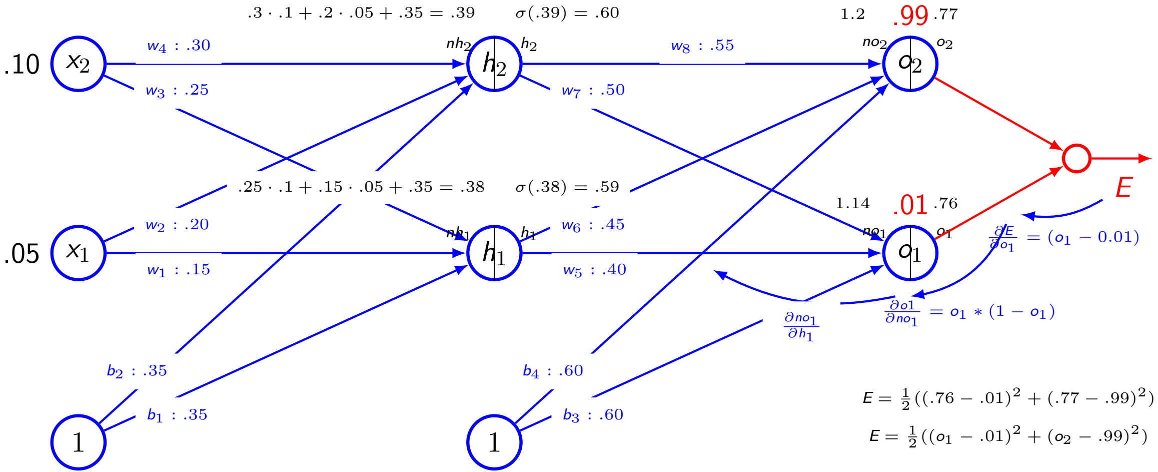
$$E = \frac{1}{2}((.76 - .01)^2 + (.77 - .99)^2)$$

$$E = \frac{1}{2}((o_1 - .01)^2 + (o_2 - .99)^2)$$

Example

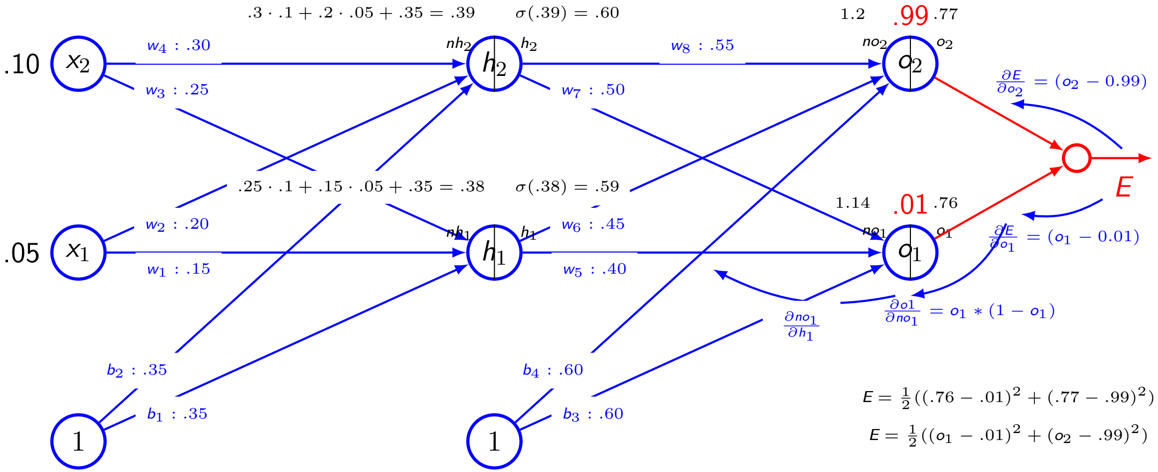

.3 · .1 + .2 · .05 + .35 = .39
σ(.39) = .60
.25 · .1 + .15 · .05 + .35 = .38
σ(.38) = .59
1.2
.99
.77
1.14
.01
.76
.10
.05
x2
x1
h2
h1
o2
o1
nh2
nh1
no2
no1
w4 : .30
w3 : .25
w2 : .20
w1 : .15
w8 : .55
w7 : .50
w6 : .45
w5 : .40
b2 : .35
b1 : .35
b4 : .60
b3 : .60
1
1
E
∂E/∂o1 = (o1 − 0.01)
∂o1/∂no1 = o1 ∗ (1 − o1)
E = 1/2((.76 − .01)^2 + (.77 − .99)^2)
E = 1/2((o1 − .01)^2 + (o2 − .99)^2)

Example

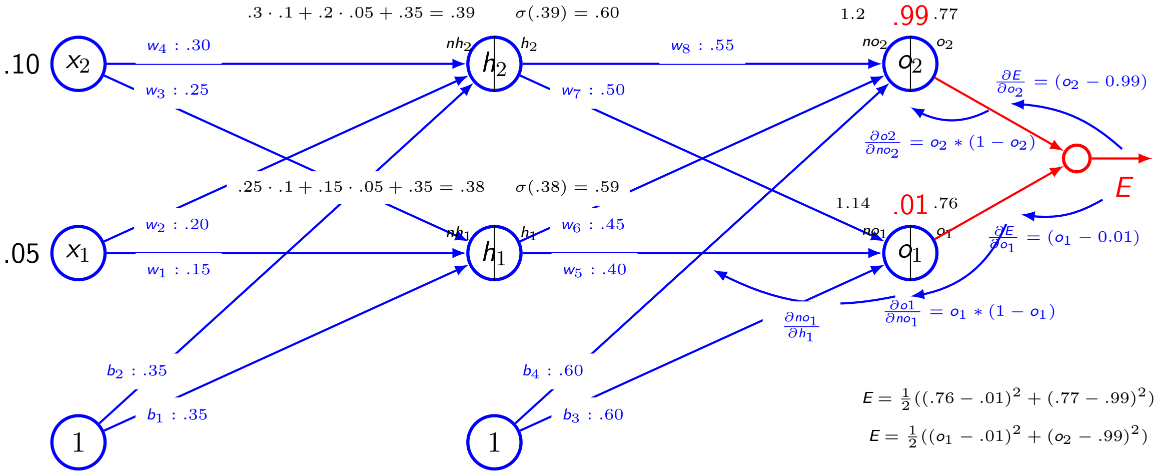


.3 · .1 + .2 · .05 + .35 = .39
σ(.39) = .60
1.2
.99
.77
.10
x2
w4 : .30
nh2
h2
h2
w8 : .55
no2
o2
o2
w3 : .25
w7 : .50
.25 · .1 + .15 · .05 + .35 = .38
σ(.38) = .59
1.14
.01
.76
E
.05
x1
w2 : .20
nh1
h1
h1
w6 : .45
no1
o1
o1
∂E/∂o1 = (o1 − 0.01)
w1 : .15
w5 : .40
∂no1/∂h1
∂o1/∂no1 = o1 ∗ (1 − o1)
b2 : .35
b4 : .60
b1 : .35
b3 : .60
1
1
E = 1/2((.76 − .01)^2 + (.77 − .99)^2)
E = 1/2((o1 − .01)^2 + (o2 − .99)^2)

Example


.3 · .1 + .2 · .05 + .35 = .39
σ(.39) = .60
1.2
.99
.77
.10
x2
w4 : .30
w3 : .25
nh2
h2
w8 : .55
w7 : .50
no2
o2
∂E/∂o2 = (o2 − 0.99)
.25 · .1 + .15 · .05 + .35 = .38
σ(.38) = .59
1.14
.01
.76
E
.05
x1
w2 : .20
w1 : .15
nh1
h1
w6 : .45
w5 : .40
no1
o1
∂E/∂o1 = (o1 − 0.01)
∂o1/∂no1 = o1 ∗ (1 − o1)
∂no1/∂h1
b2 : .35
b1 : .35
b4 : .60
b3 : .60
1
1
E = 1/2((.76 − .01)^2 + (.77 − .99)^2)
E = 1/2((o1 − .01)^2 + (o2 − .99)^2)

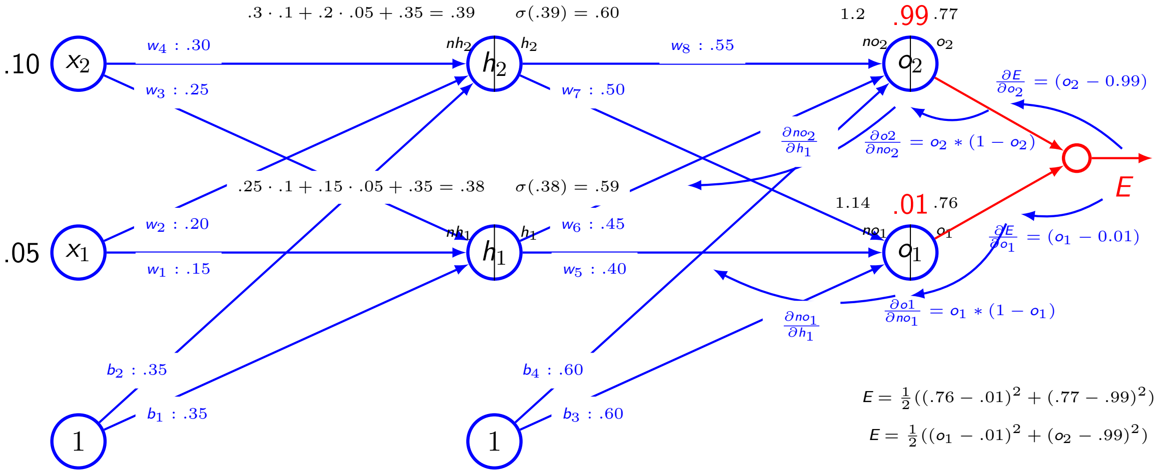
Example



$$E = \frac{1}{2}((.76 - .01)^2 + (.77 - .99)^2)$$

$$E = \frac{1}{2}((o_1 - .01)^2 + (o_2 - .99)^2)$$

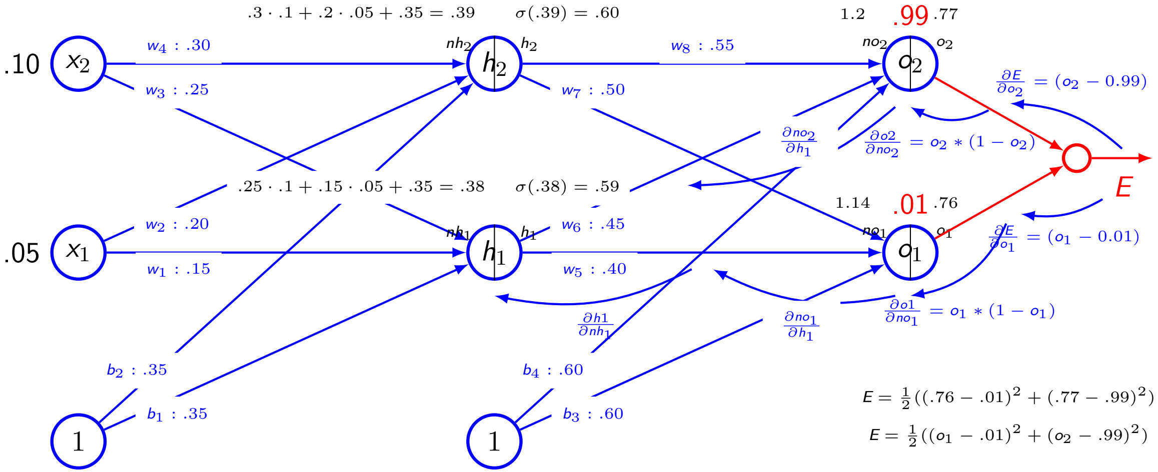
Example



$$E = \frac{1}{2}((.76 - .01)^2 + (.77 - .99)^2)$$

$$E = \frac{1}{2}((o_1 - .01)^2 + (o_2 - .99)^2)$$

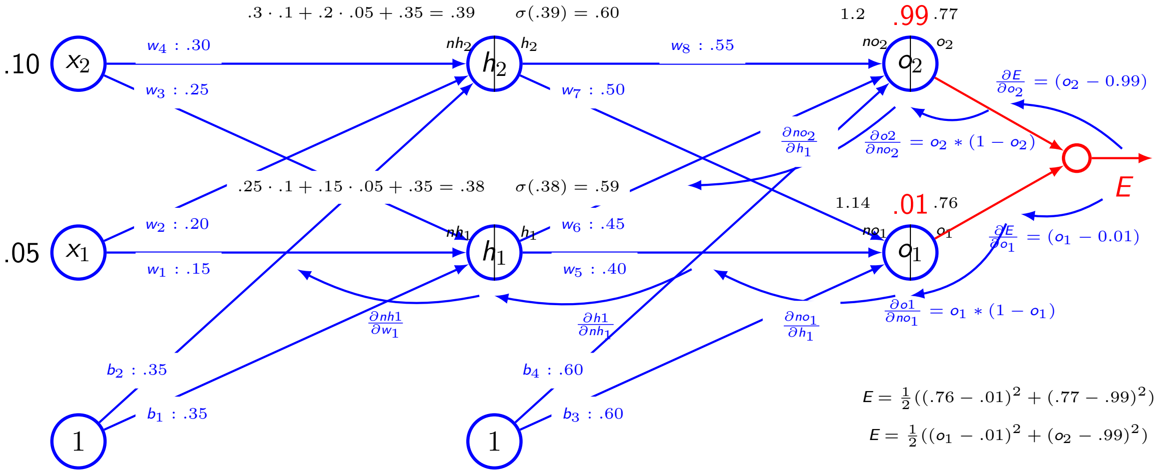
Example



$$E = \frac{1}{2}((.76 - .01)^2 + (.77 - .99)^2)$$

$$E = \frac{1}{2}((o_1 - .01)^2 + (o_2 - .99)^2)$$

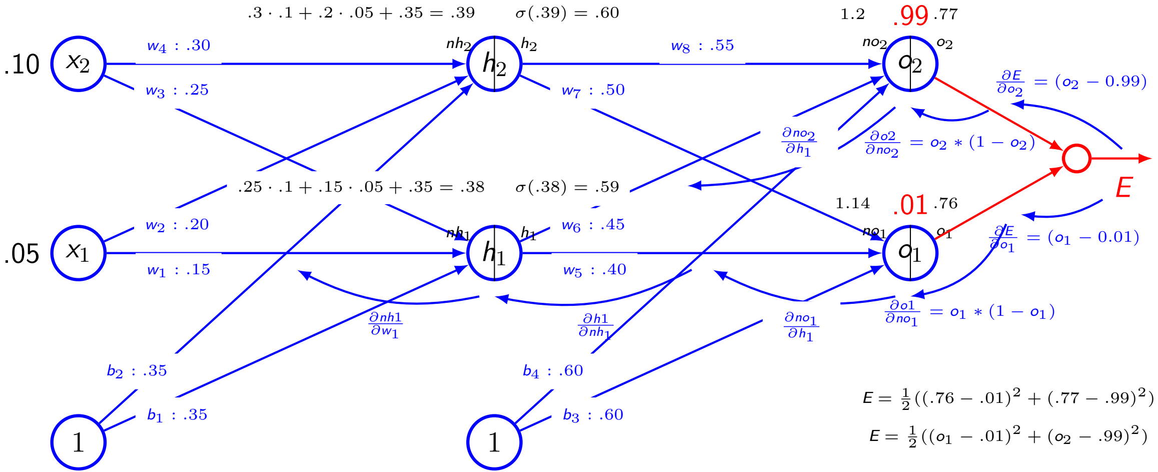
Example



$$E = \frac{1}{2}((.76 - .01)^2 + (.77 - .99)^2)$$

$$E = \frac{1}{2}((o_1 - .01)^2 + (o_2 - .99)^2)$$

Example



$$E = \frac{1}{2}((.76 - .01)^2 + (.77 - .99)^2)$$

$$E = \frac{1}{2}((o_1 - .01)^2 + (o_2 - .99)^2)$$

$$\frac{\partial E}{\partial w_1} = \left(\frac{\partial E}{\partial o_1} \times \frac{\partial o_1}{\partial no_1} \times \frac{\partial no_1}{\partial h1} + \frac{\partial E}{\partial o_2} \times \frac{\partial o_2}{\partial no_2} \times \frac{\partial no_2}{\partial h1}\right) \times \frac{\partial h1}{\partial nh1} \times \frac{\partial nh1}{\partial w_1}$$

Application of chain rule

- Let us consider $u^{(n)}$ be the loss quantity. Need to find out the gradient for this.
- Let $u^{(1)}$ to $u^{(n_i)}$ are the inputs
- Therefore, we wish to compute $\frac{\partial u^{(n)}}{\partial u^{(i)}}$ where $i = 1, 2, \ldots, n_i$
- Let us assume the nodes are ordered so that we can compute one after another
- Each $u^{(i)}$ is associated with an operation $f^{(i)}$ ie. $u^{(i)} = f(\mathbb{A}^{(i)})$

Algorithm for forward pass

for $i = 1, \ldots, n_i$ **do**

$u^{(i)} \leftarrow x_i$

end for

for $i = n_i + 1, \ldots, n$ **do**

$\mathbb{A}^{(i)} \leftarrow \{u^{(j)} | j \in Pa(u^{(i)})\}$
$u^{(i)} \leftarrow f^{(i)}(\mathbb{A}^{(i)})$

end for

return $u^{(n)}$

Algorithm for backward pass

$\texttt{grad\_table}[u^{(n)}] \leftarrow 1$

for $j = n - 1$ down to 1 **do**

$$\texttt{grad\_table}[u^{(j)}] \leftarrow \sum_{i: j \in Pa(u^{(i)})} \texttt{grad\_table}[u^{(i)}] \frac{\partial u^{(i)}}{\partial u^{(j)}}$$

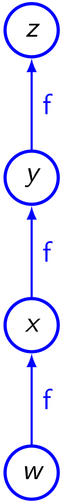
end for

return $\texttt{grad\_table}$

Computational graph & subexpression

- We have $x = f(w)$, $y = f(x)$, $z = f(y)$

$$
\begin{aligned}
& \frac{\partial z}{\partial w} \\
= \quad & \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\frac{\partial x}{\partial w} \\
= \quad & f'(y)f'(x)f'(w) \\
= \quad & f'(f(f(w)))f'(f(w))f'(w)
\end{aligned}
$$



Forward propagation in MLP

- Input
  - $\mathsf{h}^{(0)} = \mathsf{x}$
- Computation for each layer $k = 1, \ldots, l$
  - $\mathsf{a}^{(k)} = \mathsf{b}^{(k)} + \mathsf{W}^{(k)}\mathsf{h}^{(k-1)}$
  - $\mathsf{h}^{(k)} = f(\mathsf{a}^{(k)})$
- Computation of output and loss function
  - $\hat{\mathsf{y}} = \mathsf{h}^{(l)}$
  - $J = L(\hat{\mathsf{y}}, \mathsf{y}) + \lambda\Omega(\theta)$

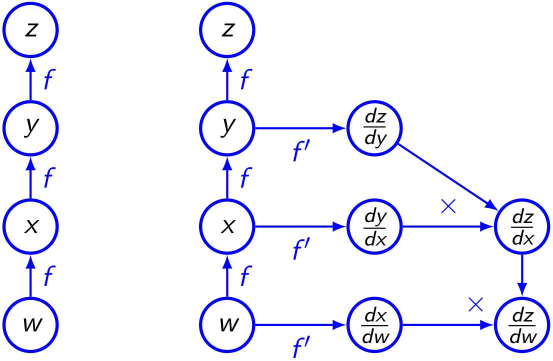
Backward computation in MLP

- Compute gradient at the output
  - $\mathrm{g} \leftarrow \nabla_{\hat{\mathrm{y}}} J = \nabla_{\hat{\mathrm{y}}} L(\hat{\mathrm{y}}, \mathrm{y})$
- Convert the gradient at output layer into gradient of pre-activation
  - $\mathrm{g} \leftarrow \nabla_{\mathrm{a}^{(k)}} J = \mathrm{g} \odot f'(\mathrm{a}^{(k)})$
- Compute gradient on weights and biases
  - $\nabla_{\mathrm{b}^{(k)}} J = \mathrm{g} + \lambda \nabla_{\mathrm{b}^{(k)}} \Omega(\theta)$
  - $\nabla_{\mathrm{W}^{(k)}} J = \mathrm{g}\mathrm{h}^{(k-1)T} + \lambda \nabla_{\mathrm{W}^{(k)}} \Omega(\theta)$
- Propagate the gradients wrt the next lower level activation
  - $g \leftarrow \nabla_{\mathrm{h}^{(k-1)}} J = \mathrm{W}^{(k)T} \mathrm{g}$

Computation of derivatives

- Takes a computational graph and a set of numerical values for the inputs, then return a set of numerical values
  - Symbol-to-number differentiation
  - Torch, Caffe
- Takes computational graph and add additional nodes to the graph that provide symbolic description of derivative
  - Symbol-to-symbol derivative
  - Theano, TensorFlow

Example



Summary

- Writing gradient for each parameter is difficult
- Recursive application of chain rule along the computational graph help to compute the gradients
- Forward pass - compute the value of the operations and store the necessary information
- Backward pass - uses the loss function, computes the gradient, updates the parameters.