CIS 678 Machine Learning

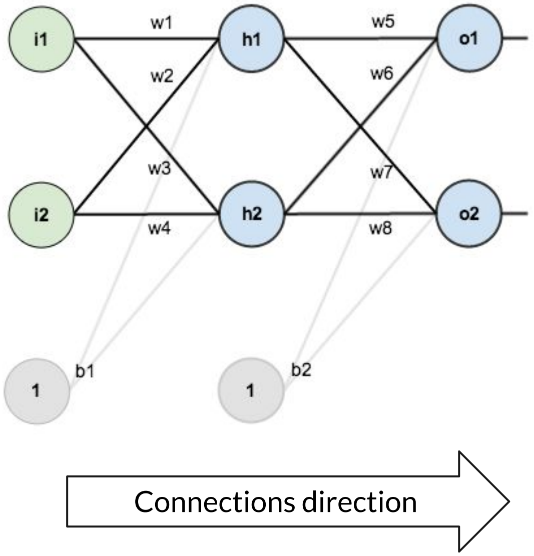
Introduction to Neural Networks (cont.)

Feed forward NNs

- Gradient Descent (Error Back Propagation)
- Challenges
  - How to control overfitting
  - Vanishing Gradient problem

Gradient Descent (Error Back Propagation)

A Basic FF Neural Network

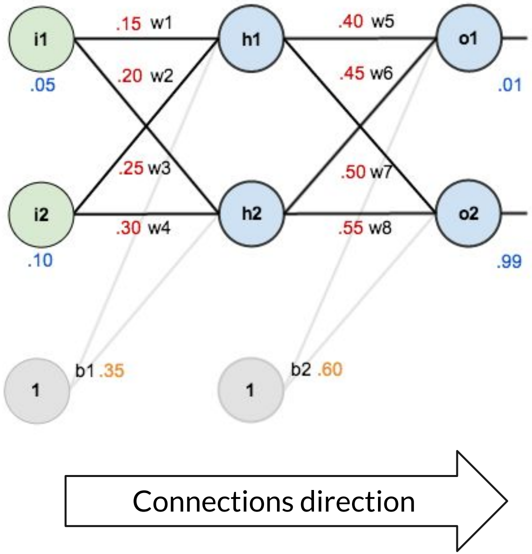


[Adapted from]

Gradient Descent (Error Back Propagation)

A Basic FF Neural Network

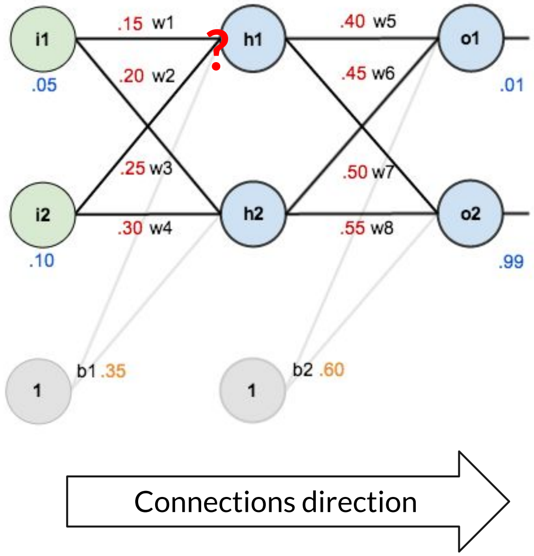
- Let's initialize with some
    - Inputs
    - Network weights including biases, and
    - Outputs (ground truths)



[Adapted from]()

Gradient Descent (Error Back Propagation)

What's the total net input to node: h1?

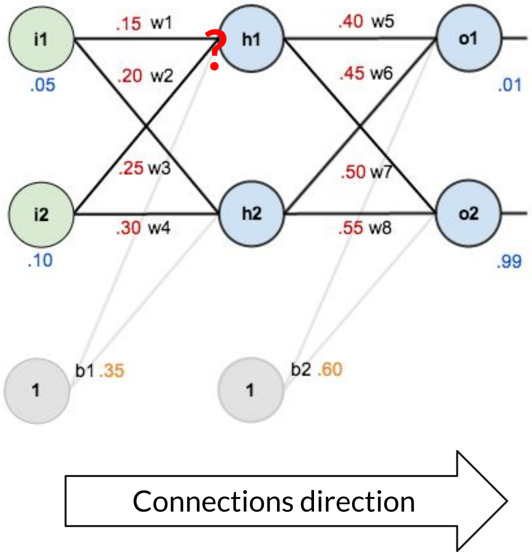


[Adapted from]()

Gradient Descent (Error Back Propagation)

What's the total net input to node: h1?

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$
$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$



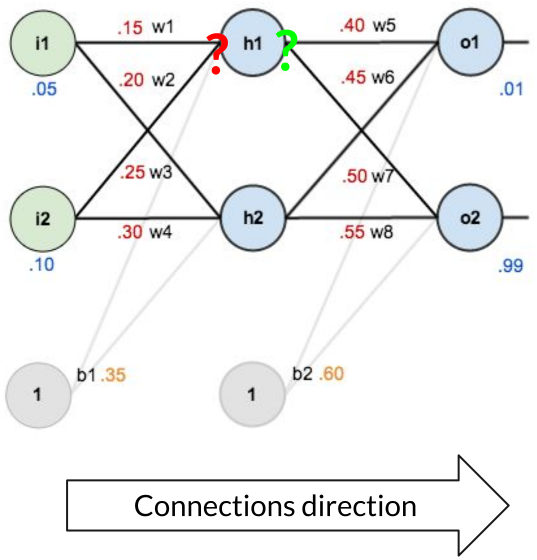
[Adapted from]()

Gradient Descent (Error Back Propagation)

What's the total net input to node: h1?

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$
$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

What will be output of h1 if it uses a Sigmoid activation function?



Adapted from

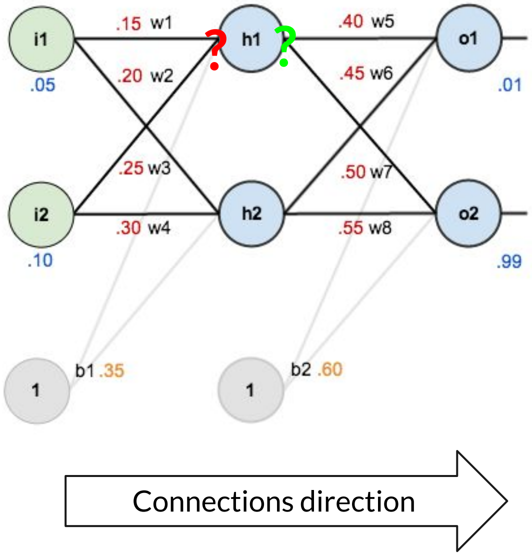
Gradient Descent (Error Back Propagation)

What's the total net input to node: h1?

$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$
$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

What will be output of h1 if it uses a Sigmoid activation function?

$$out_{h1} = \frac{1}{1+e^{-net_{h1}}} = \frac{1}{1+e^{-0.3775}} = 0.593269992$$

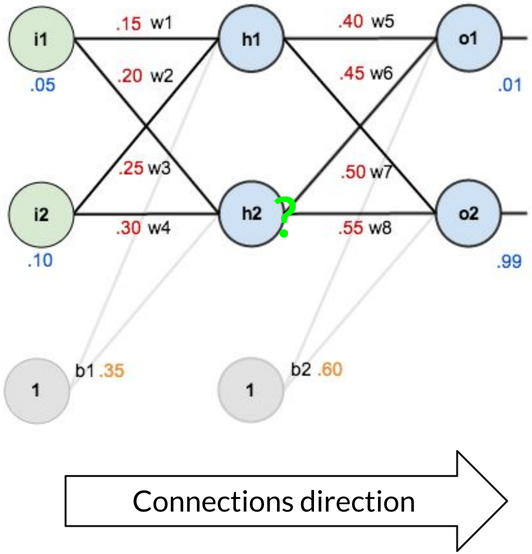


Adapted from

Gradient Descent (Error Back Propagation)

Likewise calculated, the output of the h2 would be?

$$out_{h2} = 0.596884378$$



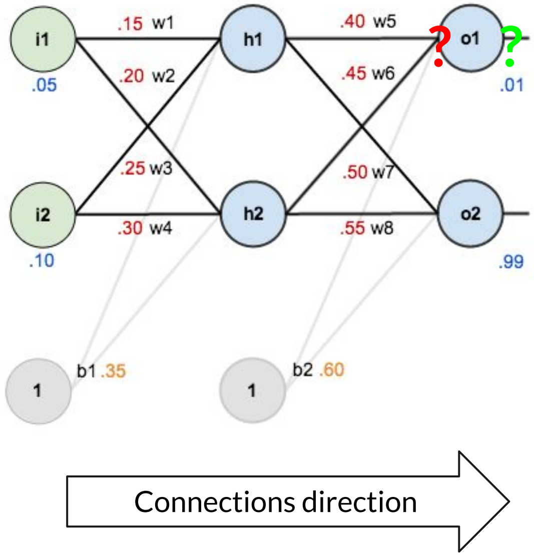
[Adapted from]()

Gradient Descent (Error Back Propagation)

Now we will calculate the input and output of O1.

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$out_{o1} = \frac{1}{1+e^{-net_{o1}}} = \frac{1}{1+e^{-1.105905967}} = 0.75136507$$

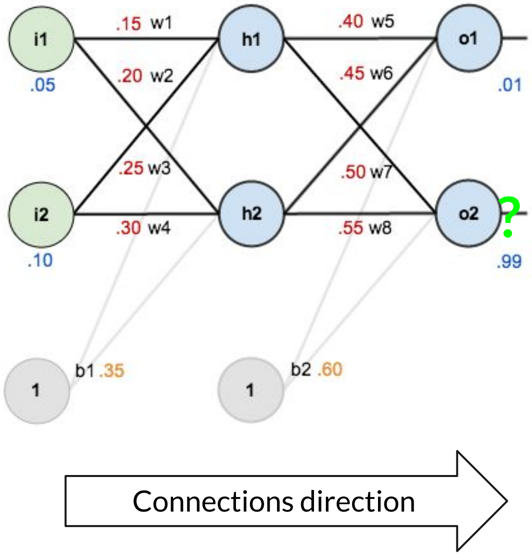


[Adapted from]()

Gradient Descent (Error Back Propagation)

Likewise calculated, the output of the O2 would be?

$out_{o2} = 0.772928465$



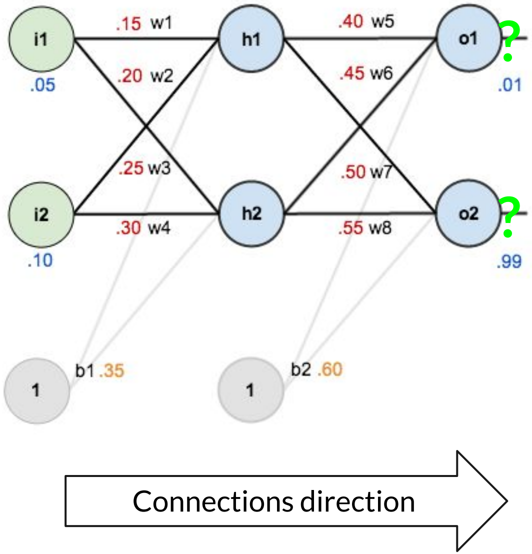
[Adapted from]

Gradient Descent (Error Back Propagation)

We have both O1 and O2 available now. This will allow to calculate the model loss/error?

$out_{o1} = 0.75136507$

$out_{o2} = 0.772928465$



[Adapted from]

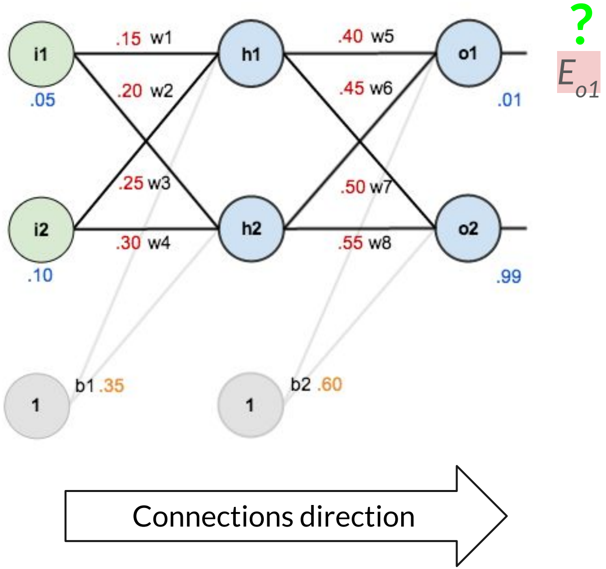
Gradient Descent (Error Back Propagation)

We have both O1 and O2 available now. This will allow to calculate the model loss/error?

$$out_{o1} = 0.75136507$$

$$out_{o2} = 0.772928465$$

$$E_{o1} = \frac{1}{2}(target_{o1} - out_{o1})^2 = \frac{1}{2}(0.01 - 0.75136507)^2 = 0.274811083$$



[Adapted from]

Gradient Descent (Error Back Propagation)

We have both O1 and O2 available now. This will allow to calculate the model loss/error?

$$out_{o1} = 0.75136507$$

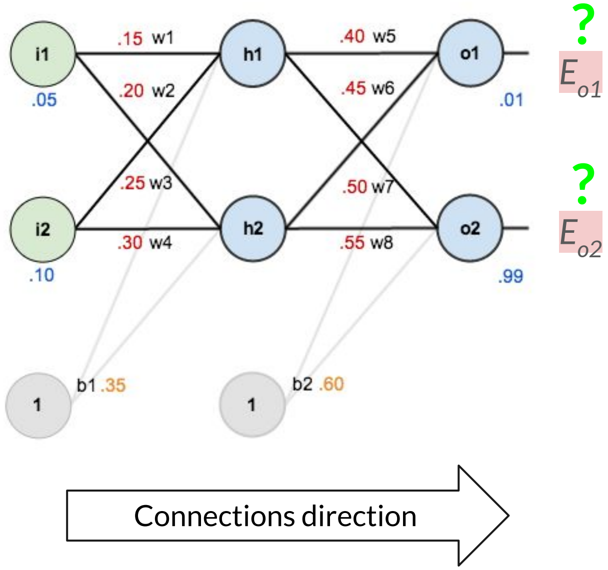
$$out_{o2} = 0.772928465$$

$$E_{o1} = \frac{1}{2}(target_{o1} - out_{o1})^2 = \frac{1}{2}(0.01 - 0.75136507)^2 = 0.274811083$$

Likewise,

$$E_{o2} = 0.023560026$$

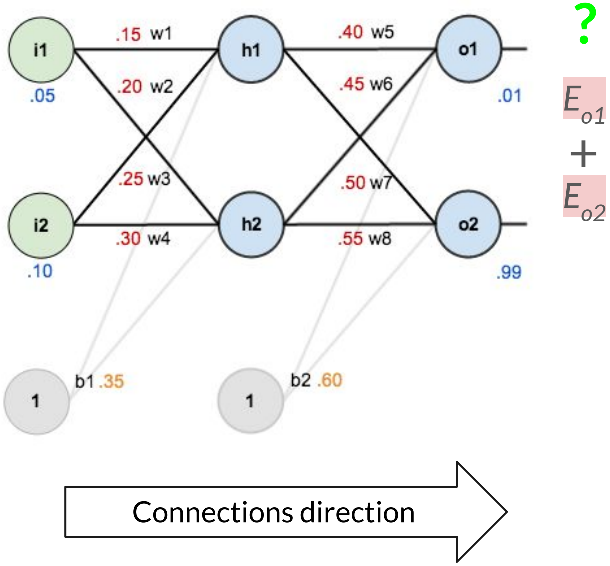
Adapted from



Gradient Descent (Error Back Propagation)

Total error/loss of the network

$$E_{total} = E_{o1} + E_{o2} = 0.274811083 + 0.023560026 = 0.298371109$$



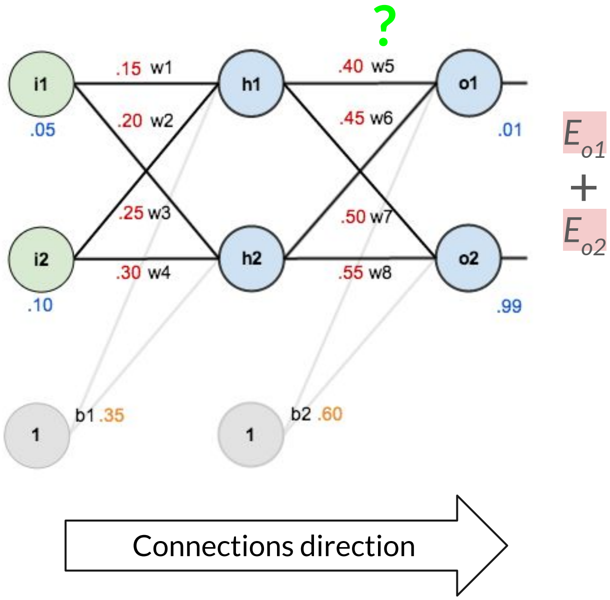
[Adapted from](#)

Gradient Descent (Error Back Propagation)

The Backwards Pass

Let's focus on $\frac{\partial E_{total}}{\partial w_5}$

What would be the gradient update for w5?



[Adapted from]

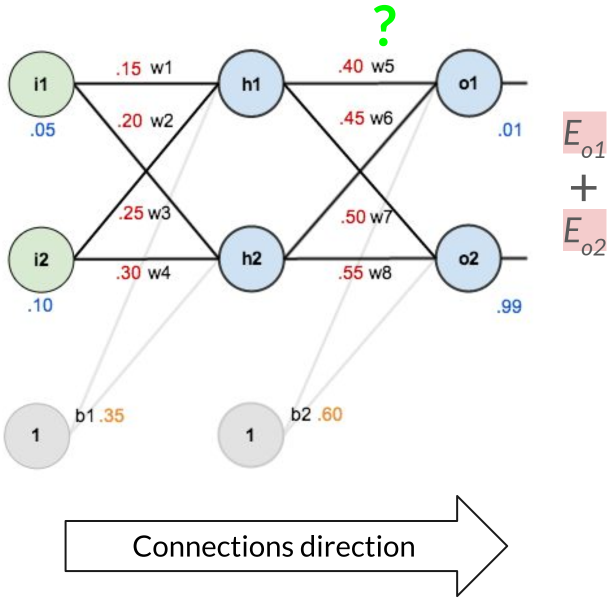
Gradient Descent (Error Back Propagation)

The Backwards Pass

Let’s focus on $\frac{\partial E_{total}}{\partial w_5}$

What would be the gradient update for w5?

We have to apply the chain rule.



Adapted from

Gradient Descent (Error Back Propagation)

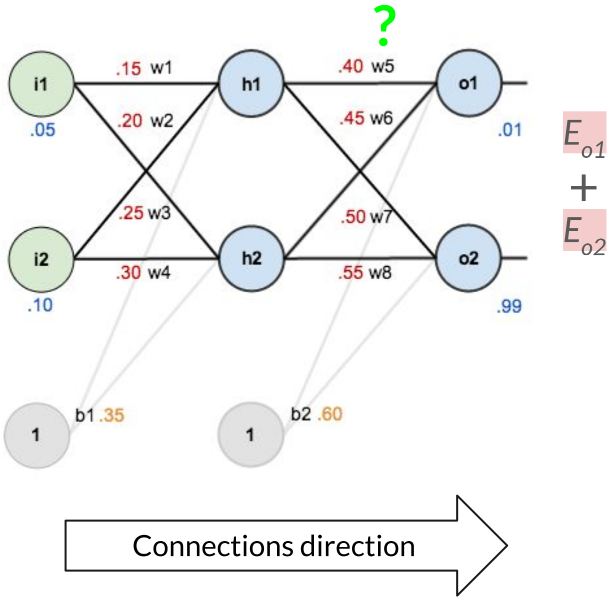
The Backwards Pass

Let's focus on $\frac{\partial E_{total}}{\partial w_5}$

What would be the gradient update for w5?

We have to apply the chain rule.

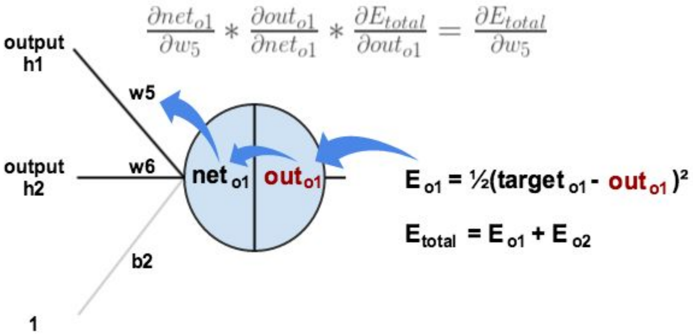
$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

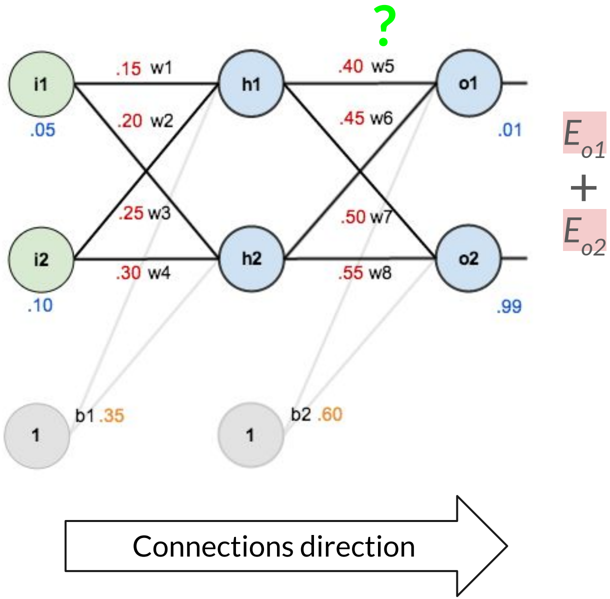


Adapted from

Gradient Descent (Error Back Propagation)

The Backwards Pass

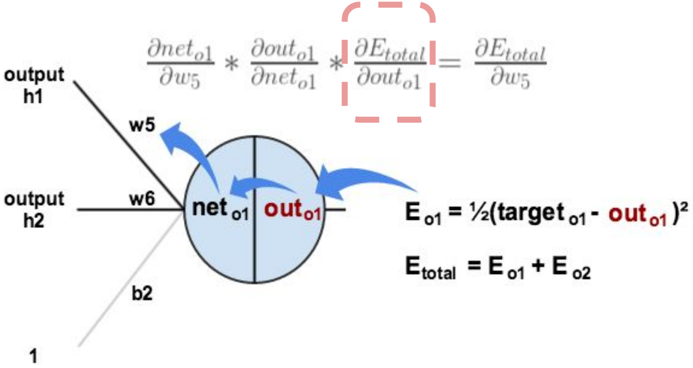




[Adapted from]

Gradient Descent (Error Back Propagation)

The Backwards Pass



$$E_{total} = \frac{1}{2}(target_{o1} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = 2 * \frac{1}{2}(target_{o1} - out_{o1})^{2-1} * -1 + 0$$

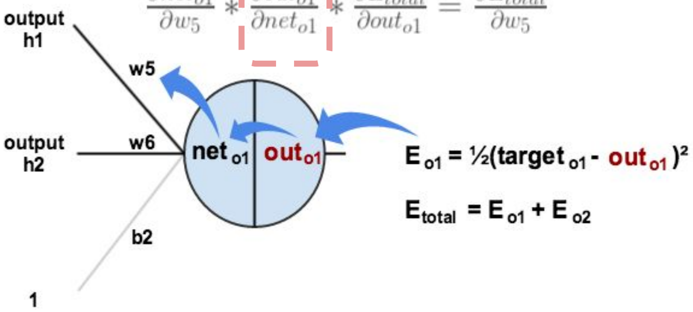
$$\frac{\partial E_{total}}{\partial out_{o1}} = -(target_{o1} - out_{o1}) = -(0.01 - 0.75136507) = 0.74136507$$

[Adapted from]()

Gradient Descent (Error Back Propagation)

The Backwards Pass

$$\frac{\partial net_{o1}}{\partial w_5} * \boxed{\frac{\partial out_{o1}}{\partial net_{o1}}} * \frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{total}}{\partial w_5}$$



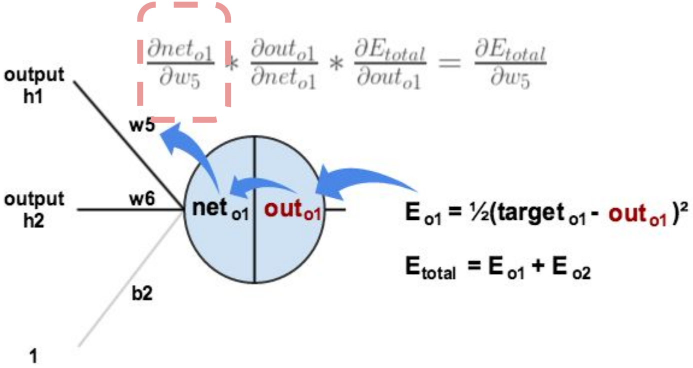
$$out_{o1} = \frac{1}{1+e^{-net_{o1}}}$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}(1 - out_{o1})$$
$$= 0.75136507(1 - 0.75136507)$$
$$= 0.186815602$$

[Adapted from]()

Gradient Descent (Error Back Propagation)

The Backwards Pass



$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

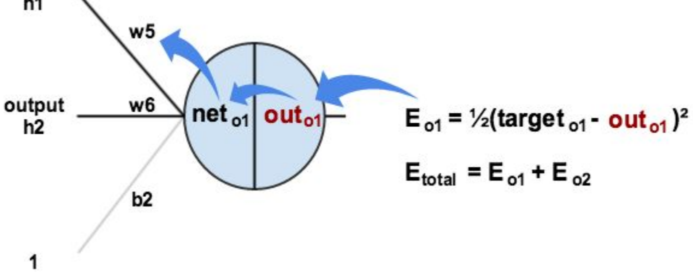
$$\frac{\partial net_{o1}}{\partial w_5} = 1 * out_{h1} * w_5^{(1-1)} + 0 + 0 = out_{h1} = 0.593269992$$

[Adapted from]

Gradient Descent (Error Back Propagation)

The Backwards Pass

$$\frac{\partial net_{o1}}{\partial w_5} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{total}}{\partial w_5}$$



Putting it all together:

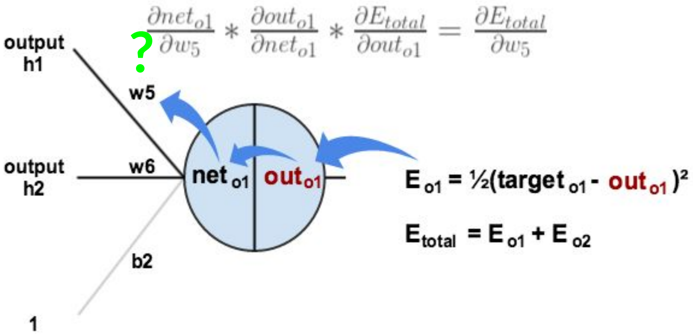
$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.74136507 * 0.186815602 * 0.593269992 = 0.082167041$$

[Adapted from]()

Gradient Descent (Error Back Propagation)

The Backwards Pass



[Adapted from]()

Putting it all together:

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial net_{o1}}{\partial w_5}$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.74136507 * 0.186815602 * 0.593269992 = 0.082167041$$

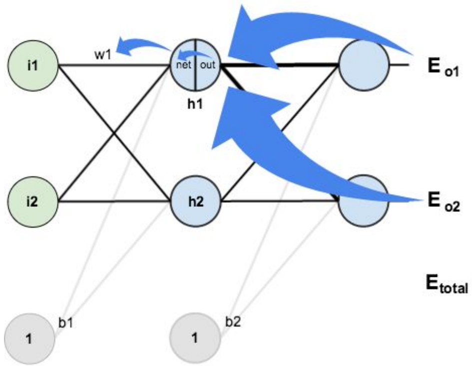
So, we can now update w5 (gradient descent)

Gradient Descent (Error Back Propagation)

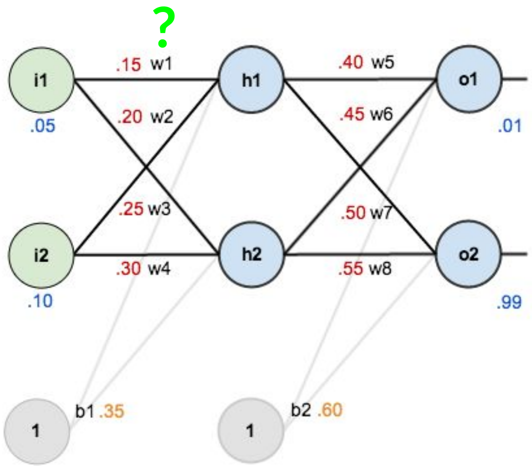
Hidden Layer

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$E_{total} = E_{o1} + E_{o2}$



E_{o1} $+$ E_{o2}

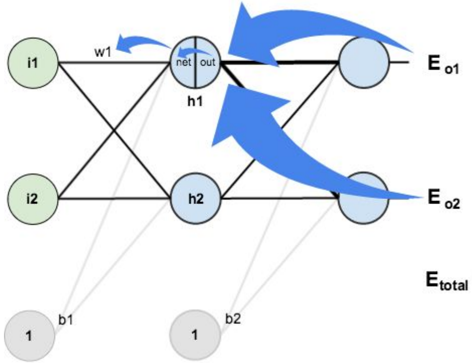
- While changing w5 affects only O1, a change in w1 will change both O1 and O2

Gradient Descent (Error Back Propagation)

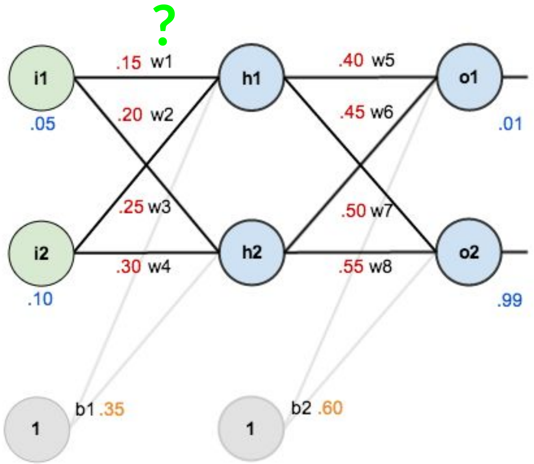
Hidden Layer

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



$E_{total} = E_{o1} + E_{o2}$



E_{o1}
$+$
E_{o2}

- Can you think of an arbitrary node in a giant and complex NN? What challenges we may encounter?

QA