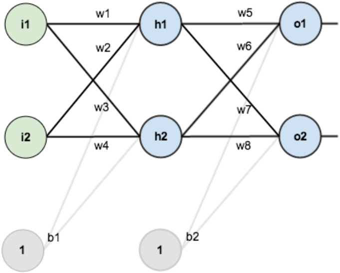
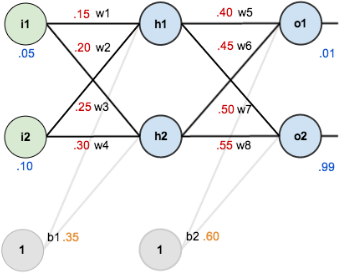
Neural Network: A Running Example

https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/



Neural Network: A Running Example

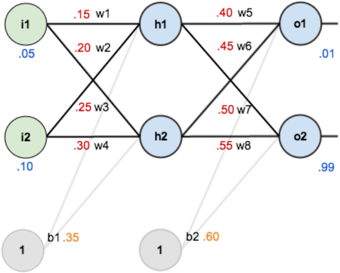
https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/



In order to have some numbers to work with, here are the initial weights, the biases, and training inputs/outputs

The forward pass:

- $net_{h_1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1 = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$
- $out_{h_1} = \frac{1}{1+e^{-net_{h_1}}} = \frac{1}{1+e^{-0.3775}} = 0.5933$
- Similarly, $out_{h_2} = 0.5969$
- $net_{o_1} = w_5 * out_{h_1} + w_6 * out_{h_2} + b_2 * 1 = 0.4 * 0.5933 + 0.45 * 0.5969 + 0.6 * 1 = 1.1059$
- $out_{o_1} = \frac{1}{1+e^{-net_{o_1}}} = \frac{1}{1+e^{-1.1059}} = 0.7514$
- Similarly, $out_{o_2} = 0.7729$



In order to have some numbers to work with, here are the initial weights, the biases, and training inputs/outputs

Recall:

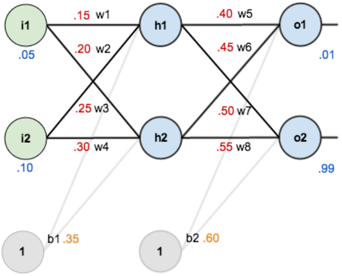
- $out_{o_1} = \frac{1}{1+e^{-net_{o_1}}} = \frac{1}{1+e^{-1.1059}} = 0.7514$
- Similarly, $out_{o_2} = 0.7729$

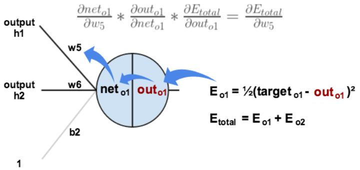
Consider square loss as total error $E_{total} = \sum_i \frac{1}{2}(target - output)^2$

- $E_{o1} = \frac{1}{2}(target_{o1} - output_{o1})^2 = \frac{1}{2}(0.01 - 0.7514)^2 = 0.2748$
- Similarly $E_{o2} = 0.0236$
- $E_{total} = 0.2748 + 0.0236 = 0.2984$

Neural Network: A Running Example

https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/





In the backward pass, we consider the chain rule for w_5:

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

Compute $\frac{\partial E_{total}}{\partial out_{o1}}$:

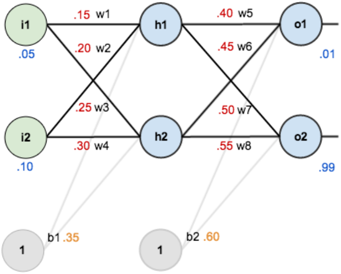
- $E_{total} = \frac{1}{2}(target_{o1} - output_{o1})^2 + \frac{1}{2}(target_{o2} - output_{o2})^2$
- $\frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{o_1}}{\partial out_{o1}} = (output_{o1} - target_{o1}) = 0.7514 - 0.01 = 0.7414$

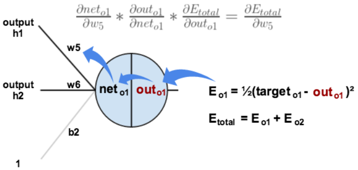
Compute $\frac{\partial out_{o1}}{\partial net_{o1}}$:

- $out_{o1} = \frac{1}{1+e^{-net_{o_1}}} \doteq \sigma(net_{o_1})$
- $\frac{\partial out_{o1}}{\partial net_{o_1}} = \frac{e^{-net_{o_1}}}{(1+e^{-net_{o_1}})^2} = (1 - \sigma(net_{o_1}))\sigma(net_{o_1}) = 0.7514(1 - 0.7514) = 0.1868$

Neural Network: A Running Example

https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/





Compute $\frac{\partial net_{o1}}{\partial w_5}$:

- $net_{o1} = w_5 * out_{h_1} + w6 * out_{h_2} + b_2 * 1$
- $\frac{\partial net_{o1}}{\partial w_5} = out_{h_1} = 0.5933$

Put all together:

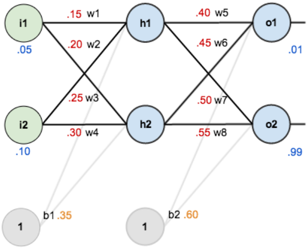
- $\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \frac{\partial out_{o1}}{\partial net_{o1}} \frac{\partial net_{o1}}{\partial w_5} =$
  $0.7414 * 0.1868 * 0.5932 = 0.0822$

We update w_5 as:

- $w_5' = w_5 - \eta \frac{\partial E_{total}}{\partial w_5} =$
  $0.4 - 0.5 * 0.0822 = 0.3589$

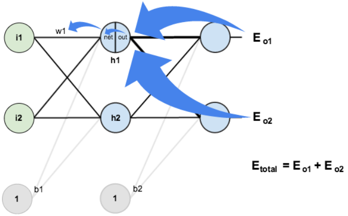
Neural Network: A Running Example

https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/



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

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

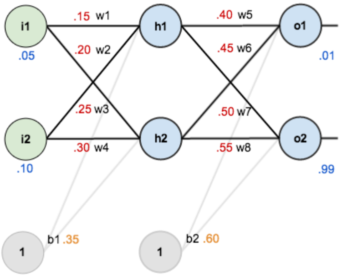


The chain rule for w_1:

- $\frac{\partial E_{total}}{\partial w_1} = \left(\frac{\partial E_{o_1}}{\partial out_{h1}} + \frac{\partial E_{o_2}}{\partial out_{h1}}\right)\frac{\partial out_{h1}}{\partial net_{h1}}\frac{\partial net_{h1}}{\partial w_1}$
  where
  $\frac{\partial E_{o_1}}{\partial out_{h1}} = \frac{\partial E_{o_1}}{\partial out_{o1}}\frac{\partial out_{o1}}{\partial net_{o1}}\frac{\partial net_{o1}}{\partial out_{h1}}$
  and
  $\frac{\partial E_{o_2}}{\partial out_{h1}} = \frac{\partial E_{o_2}}{\partial out_{o2}}\frac{\partial out_{o2}}{\partial net_{o2}}\frac{\partial net_{o2}}{\partial out_{h1}}$
- Note that here $\frac{\partial E_{o_1}}{\partial out_{o1}}\frac{\partial out_{o1}}{\partial net_{o1}}$ have been computed before and can be reused
- After simple calculation as before, we have $\frac{\partial E_{total}}{\partial w_1} = 0.0004$
- $w_1' = w_1 - \eta\frac{\partial E_{total}}{\partial w_1} = 0.15 - 0.5 * 0.0004 = 0.1498$

Neural Network: A Running Example

https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/



Similarly we can compute all the weights:

- $w_1' = 0.1498$
- $w_2' = 0.1996$
- $w_3' = 0.2498$
- $w_4' = 0.2995$
- $w_5' = 0.3589$
- $w_6' = 0.4087$
- $w_7' = 0.5113$
- $w_8' = 0.5614$

The total error is then: 0.2910 ($<$0.2984 as initial)

http://playground.tensorflow.org/

References I