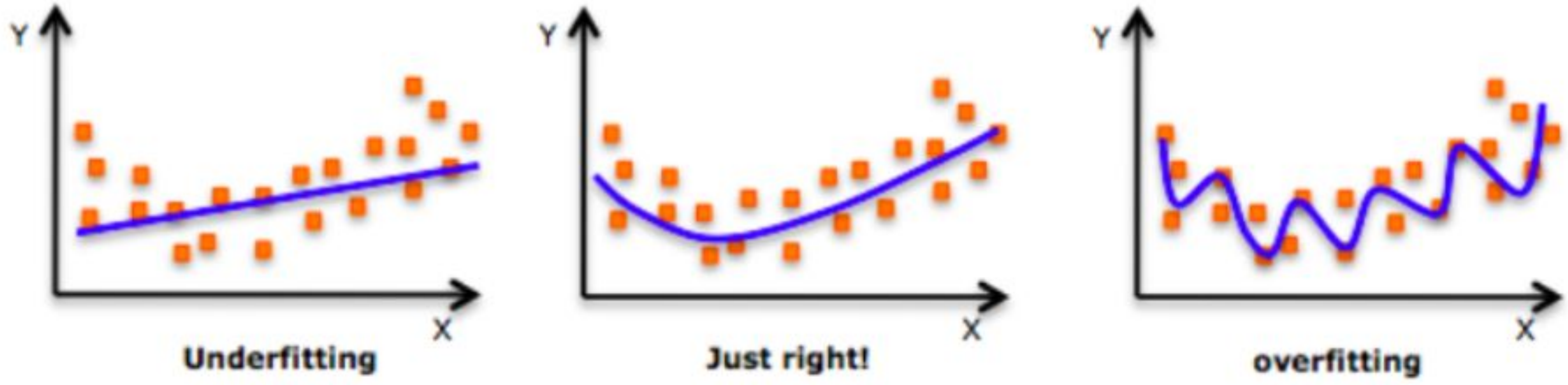
Week 6 Recitation

Krushna Shah
CS 190I Deep Learning

UC SANTA BARBARA

UC SANTA BARBARA
Computer Science

Concept of Overfitting and Regularization



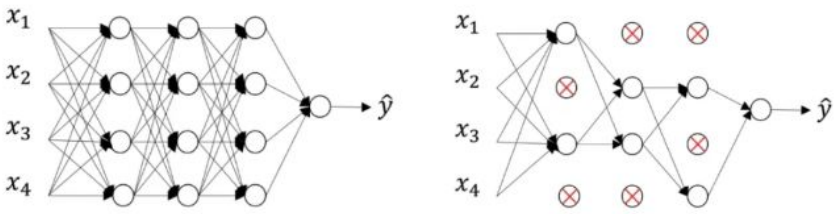
Regularization

- What is regularization?
- L1 and L2 are the most common types of regularization. These update the general cost function by adding another term known as the regularization term.
- Loss function = Loss (say, binary cross entropy) + Regularization term
- The L2 regularization - most common type of all - commonly known as weight decay or Ridge Regression.

$$Cost\,function = Loss + \frac{\lambda}{2m} * \sum \|w\|^2$$

- In the case of L1 regularization (also known as Lasso regression), we simply use the sum of the absolute values of the weight parameters in a weight matrix. $Cost\,function = Loss + \frac{\lambda}{2m} * \sum \|w\|$

- Dropout Layer



Regularization Example

Assume you are training a classification model with 4 output units and the loss function J as defined below. The weight parameters, regularization parameter, expected and predicted outputs for 4 examples are given below. Redefine your loss function J with: L1 Regularization and calculate the loss, L2 Regularization and calculate the loss.

$$J = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i$$

$$\theta = \begin{bmatrix} 0.5 \\ -0.4 \\ 0.6 \\ -0.2 \end{bmatrix}, \lambda = 0.1,$$

$$y_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \hat{y}_1 = \begin{bmatrix} 0.10 \\ 0.20 \\ 0.10 \\ 0.60 \end{bmatrix}, y_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \hat{y}_2 = \begin{bmatrix} 0.30 \\ 0.20 \\ 0.45 \\ 0.05 \end{bmatrix},$$

$$y_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \hat{y}_3 = \begin{bmatrix} 0.20 \\ 0.55 \\ 0.10 \\ 0.15 \end{bmatrix}, y_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \hat{y}_4 = \begin{bmatrix} 0.75 \\ 0.10 \\ 0.10 \\ 0.05 \end{bmatrix}$$

Regularization Example

Assume you are training a classification model with 4 output units and the loss function J as defined below. The weight parameters, regularization parameter, expected and predicted outputs for 4 examples are given below. Redefine your loss function J with: L1 Regularization and calculate the loss, L2 Regularization and calculate the loss.

Answer: L1 Regularization

$$
\begin{aligned}
J &= \frac{1}{N}\left(-\sum_{i=1}^{N} y_i \log \hat{y}_i + \lambda|\theta|\right) \\
&= \frac{1}{4}(-(\ln 0.6 + \ln 0.45 + \ln 0.55 + \ln 0.75) + \lambda(|0.5| + |-0.4| + |0.6| + |-0.2|)) \\
&= (2.1948 + 0.17)/4 \\
&= 2.3648/4 = 0.5912
\end{aligned}
$$

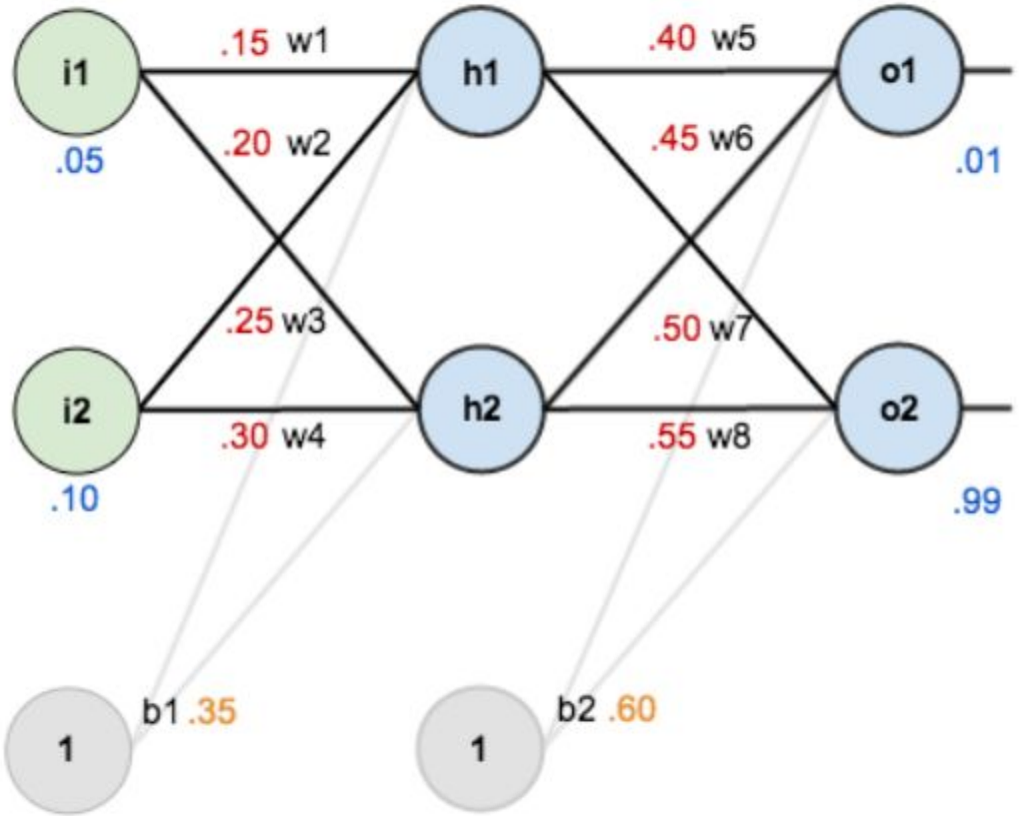
Regularization Example

Assume you are training a classification model with 4 output units and the loss function J as defined below. The weight parameters, regularization parameter, expected and predicted outputs for 4 examples are given below. Redefine your loss function J with: L1 Regularization and calculate the loss, L2 Regularization and calculate the loss.

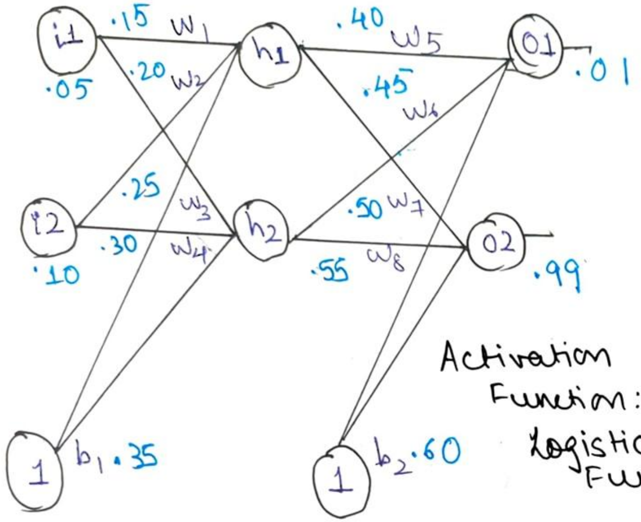
Answer: L2 Regularization

$$
\begin{aligned}
J &= \frac{1}{N}\left(-\sum_{i=1}^{m} y_i \log \hat{y}_i + \frac{1}{2}\lambda||\theta||^2\right) \\
&= \frac{1}{4}\left(-(\ln 0.6 + \ln 0.45 + \ln 0.55 + \ln 0.75) + \frac{1}{2}\lambda((0.5)^2 + (-0.4)^2 + (0.6)^2 + (-0.2)^2)\right) \\
&= (2.1948 + 0.0405)/4 \\
&= 2.2353/4 = 0.5588
\end{aligned}
$$

A Step by Step Backpropagation Example



A Step by Step Backpropagation Example



Activation Function: Logistic Function

The Forward Pass:

Hidden Layer:

$net_{h_1} = w_1 \cdot i_1 + w_2 \cdot i_2 + b_1 \cdot 1$

$= 0.15 \times 0.05 + 0.2 \times 0.1 + 0.35 \times 1$

$= 0.3775$

Applying Activation Function:

$out_{h_1} = \frac{1}{1+e^{-net_{h_1}}} = 0.593269992$

Similarly, $out_{h_2} = 0.596884378$

Repeat the process for output layer:

$net_{o_1} = 0.4 \times 0.593269992 + 0.45 \times 0.596884378 + 0.6 \times 1$

$= 1.105905967$

$out_{o_1} = \frac{1}{1+e^{-net_{o_1}}} = 0.75136507$

Similarly, $out_{o_2} = 0.772928465$

Calculating Total Error:

$E_{total} = \sum \frac{1}{2}(target - output)^2$

$E_{o_1} = \frac{1}{2}(target_{o_1} - out_{o_1})^2$

$= 0.274811083$

$E_{o_2} = 0.023560026$

$E_{total} = E_{o_1} + E_{o_2} = 0.298371109$

A Step by Step Backpropagation Example

The Backward Pass

Output Layer:

Consider w_5, we want to know

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o_1}} \times \frac{\partial out_{o_1}}{\partial net_{o_1}} \times \frac{\partial net_{o_1}}{\partial w_5}$$

Now, figuring out each piece:

$$E_{total} = \frac{1}{2}(target_{o_1} - out_{o_1})^2 + \frac{1}{2}(target_{o_2} - out_{o_2})^2$$

$$\frac{\partial E_{total}}{\partial out_{o_1}} = 2 \times \frac{1}{2}(target_{o_1} - out_{o_1})^{2-1} \times (-1) + 0$$

$$= -(target_{o_1} - out_{o_1})$$

$$= -(0.01 - 0.75136507)$$

$$= +0.74136507$$

Next, how much does the output of o_1 change w.r.t its total net input.

$$out_{o_1} = \frac{1}{1 + e^{-net_{o_1}}}$$

$$\frac{\partial out_{o_1}}{\partial net_{o_1}} = out_{o_1}(1 - out_{o_1})$$

$$= 0.75136507(1 - 0.75136507)$$

$$= 0.186815602$$

Finally, how much does total net input of o_1 change w.r.t w_5?

$$net_{o_1} = w_5 \cdot out_{h_1} + w_6 \cdot out_{h_2} + b_2 * 1$$

$$\therefore \frac{\partial net_{o_1}}{\partial w_5} = 1 \times out_{h_1} + 0 + 0$$

$$= out_{h_1} = 0.59326999$$

Putting it all together:

$$\frac{\partial E_{total}}{\partial w_5} = 0.082167041$$

To decrease error,

$$w_5^+ = w_5 - \eta \times \frac{\partial E_{total}}{\partial w_5} = 0.4 - 0.5 \times \frac{\partial E_{total}}{\partial w_5}$$

$$= 0.35891648$$

η → learning rate $(= 0.5)$

A Step by Step Backpropagation Example

Similarly,

$w_6^+ = 0.408666186$

$w_7^+ = 0.511301270$

$w_8^+ = 0.561370121$

Hidden Layer:

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h_1}} \times \frac{\partial out_{h_1}}{\partial net_{h_1}} \times \frac{\partial net_{h_1}}{\partial w_1}$$

out_{h_1} affects both out_{o_1} and out_{o_2}, therefore $\frac{\partial E_{total}}{\partial out_{h_1}}$ needs to take into consideration its effect on both output neurons.

$$\frac{\partial E_{total}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial out_{h_1}} + \frac{\partial E_{o_2}}{\partial out_{h_1}}$$

$$\frac{\partial E_{o_1}}{\partial out_{h_1}} = \frac{\partial E_{o_1}}{\partial net_{o_1}} \times \frac{\partial net_{o_1}}{\partial out_{h_1}}$$

$$\frac{\partial E_{o_1}}{\partial net_{o_1}} = \frac{\partial E_{o_1}}{\partial out_{o_1}} \times \frac{\partial out_{o_1}}{\partial net_{o_1}} = 0.138498562$$

And,

$$\frac{\partial net_{o_1}}{\partial out_{h_1}} = w_5$$

$[net_{o_1} = w_5 \times out_{h_1} + w_6 \times out_{h_2} + b_2 \times 1]$

$$\therefore \frac{\partial net_{o_1}}{\partial out_{h_1}} = w_5 = 0.4$$

Plugging them in:

$$\frac{\partial E_{o_1}}{\partial out_{h_1}} = 0.055399425$$

$$\frac{\partial E_{o_2}}{\partial out_{h_1}} = -0.019049119$$

$$\frac{\partial E_{total}}{\partial out_{h_1}} = 0.036350306$$

Now,

$$\frac{\partial out_{h_1}}{\partial net_{h_1}} = out_{h_1}(1 - out_{h_1}) = 0.241300709$$

$$\frac{\partial net_{h_1}}{\partial w_1} = i_1 = 0.05$$

$[net_{h_1} = w_1 \times i_1 + w_3 \times i_2 + b_1 \times 1]$

A Step by Step Backpropagation Example

Putting it all together:

$$\frac{\partial E_{total}}{\partial w_1} = 0.000438568$$

Updating w_1:

$$w_1^+ = w_1 - \eta \times \frac{\partial E_{total}}{\partial w_1}$$

$$= 0.15 - 0.5(0.000438568)$$

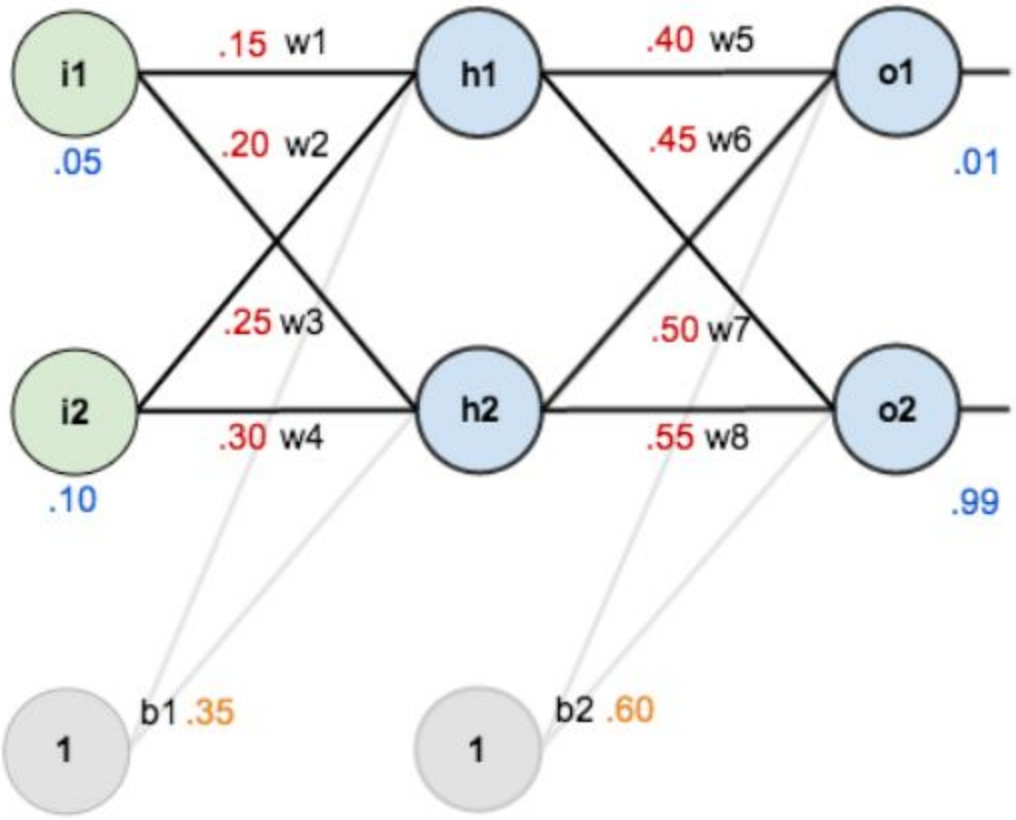
$$= 0.149780716$$

$w_2^+ = 0.19956143$

$w_3^+ = 0.24975114$

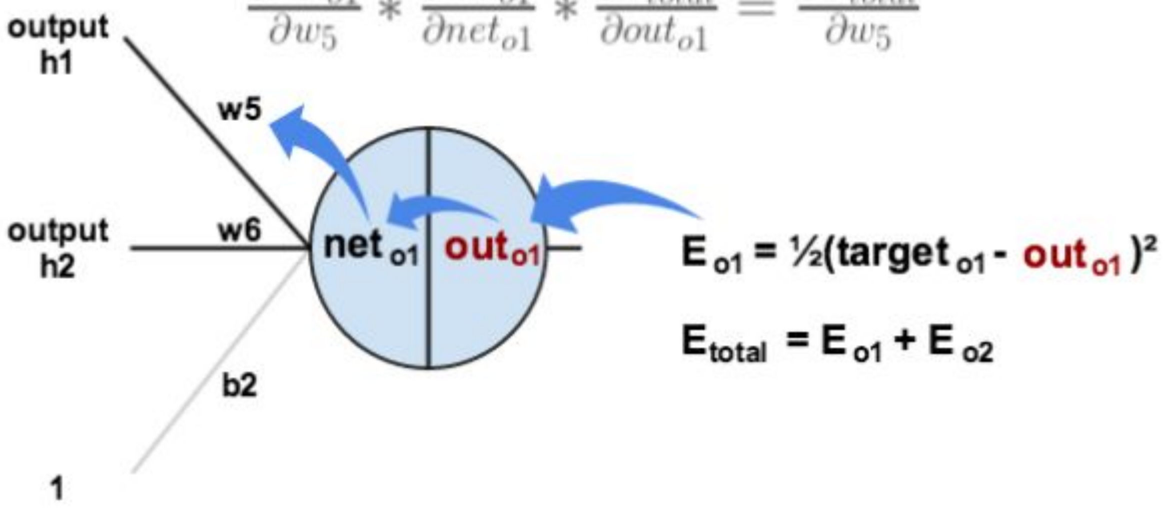
$w_4^+ = 0.29950229$

A Step by Step Backpropagation Example

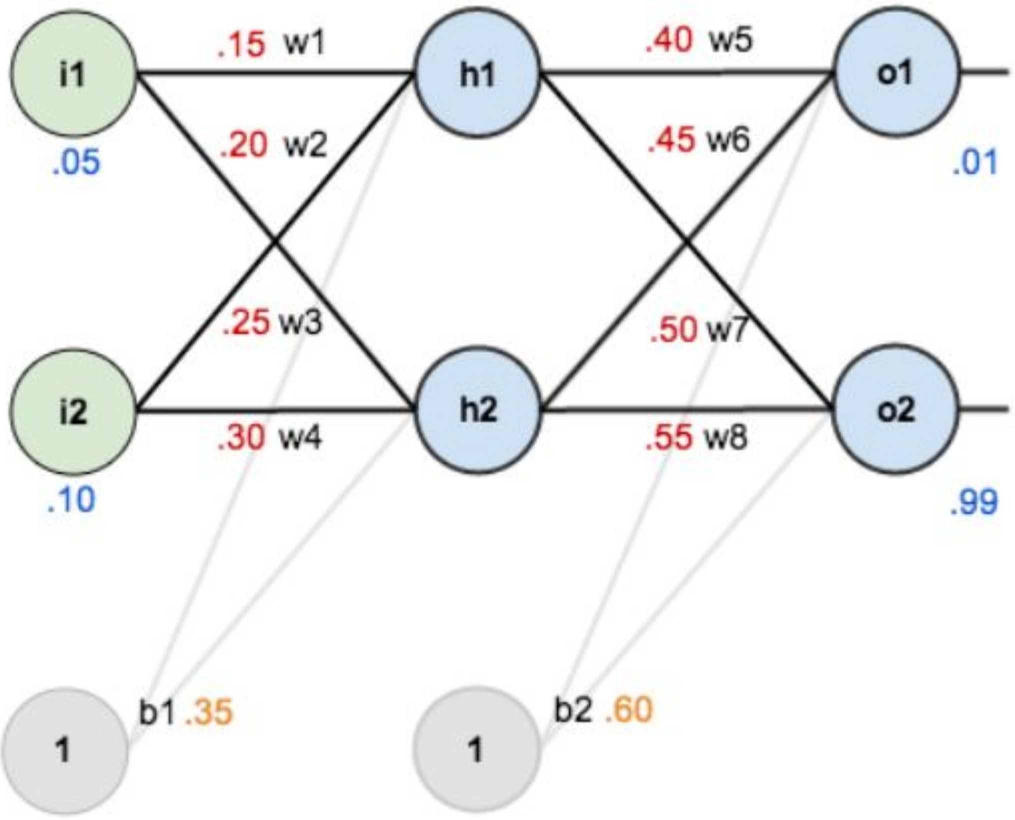


Consider w5

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

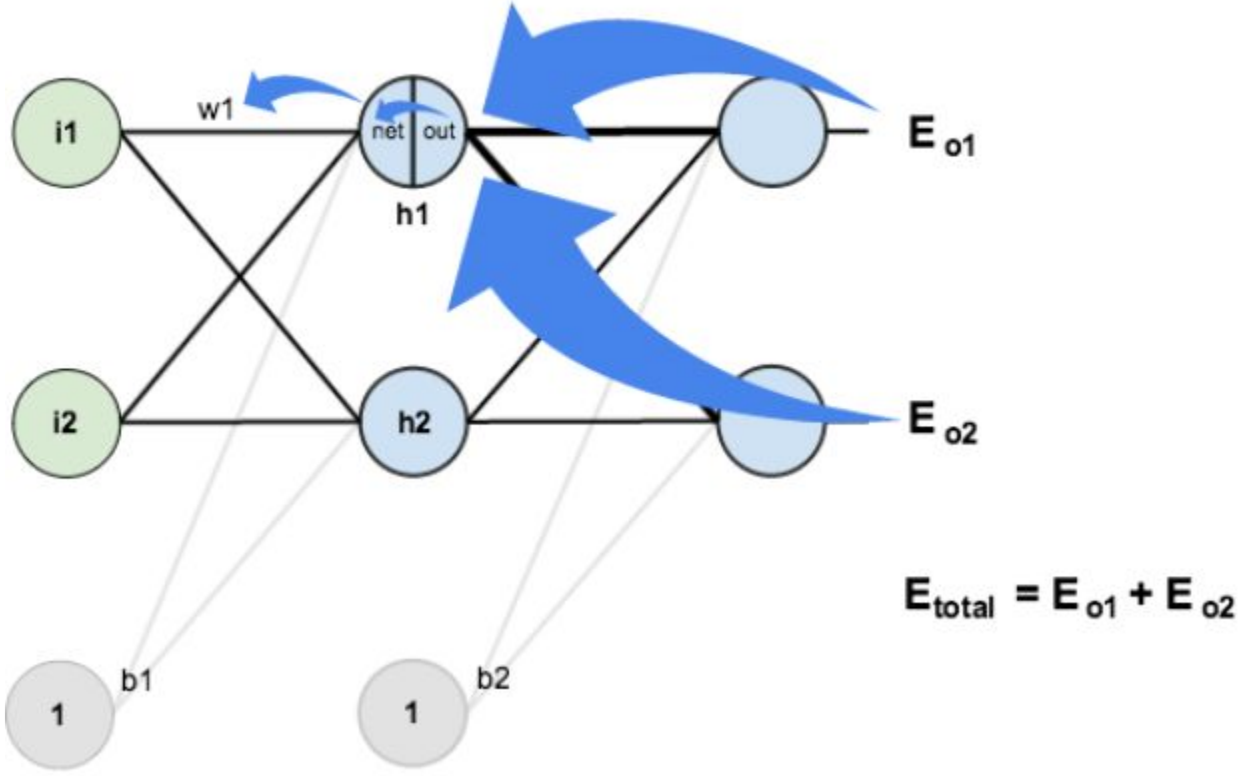


A Step by Step Backpropagation Example



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

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



CNN Example

Suppose we have one batch 100 input images each of size $3 \times 64 \times 64$. Consider a convolutional layer with 2 output channels, kernel size 5×5, no padding, and stride of 2. Answer the following questions and given brief explanations for your answers.

(a) (5') What is the shape of the weight parameters for the convolutional layer?

(b) (5') What is the output size after we feed the whole batch of input images through the convolutional layer?

(c) (5') We decide to add a linear layer after the convolutional layer to make a prediction of whether the image contains a pedestrian, what would be the input dimension for the linear layer?

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

Any Questions?