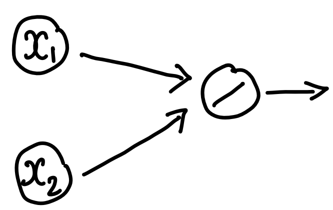
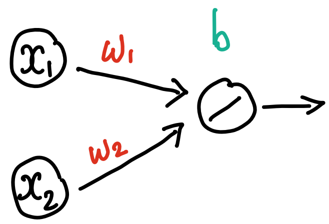
Consider a two-input problem with a continuous dependent variable and no hidden layers.



What are the parameters to be learned?



The model and MSE loss

$$\text{model}(x_1, x_2) = b + w_1 x_1 + w_2 x_2$$

$$\text{Loss} = (\text{model}(x_1, x_2) - y)^2$$

$$\text{Loss} = (b + w_1 x_1 + w_2 x_2 - y)^2$$

Let's calculate $\nabla Loss = \begin{bmatrix} \frac{\partial Loss}{\partial b} & \frac{\partial Loss}{\partial w_1} & \frac{\partial Loss}{w_2} \end{bmatrix}$ the "old-fashioned" way ☺

$$Loss = (b + w_1 x_1 + w_2 x_2 - y)^2$$

$$\frac{\partial Loss}{\partial b} = 2(b + w_1 x_1 + w_2 x_2 - y)$$

$$\frac{\partial Loss}{\partial w_1} = 2(b + w_1 x_1 + w_2 x_2 - y)x_1$$

$$\frac{\partial Loss}{\partial w_2} = 2(b + w_1 x_1 + w_2 x_2 - y)x_2$$

Now, let's organize the calculations a bit differently

Let
$$a_1 = w_1 x_1$$
$$a_2 = w_2 x_2$$
$$\hat{y} = b + a_1 + a_2$$

Plugging ↗ into ↓

$$Loss = (b + w_1 x_1 + w_2 x_2 - y)^2$$

we get:

$$Loss = (\hat{y} - y)^2$$

$$\frac{\partial Loss}{\partial b} = ? \qquad \frac{\partial Loss}{\partial w_1} = ? \qquad \frac{\partial Loss}{\partial w_2} = ?$$

We can apply the Chain Rule!

$$Loss = (\hat{y} - y)^2$$

$$\hat{y} = b + a_1 + a_2$$

$$a_1 = w_1 x_1$$

$$a_2 = w_2 x_2$$

$$\Rightarrow$$

$$\frac{\partial Loss}{\partial b} = \frac{\partial Loss}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b}$$

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1} \cdot \frac{\partial a_1}{\partial w_1}$$

$$\frac{\partial Loss}{\partial w_2} = \frac{\partial Loss}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial w_2}$$

Next, we need to calculate

$$\frac{\partial Loss}{\partial \hat{y}}, \quad \frac{\partial \hat{y}}{\partial b}, \quad \frac{\partial a_1}{\partial w_1} \text{ and } \frac{\partial a_2}{\partial w_2}$$

But that's easy!

$$Loss = (\hat{y}-y)^2 \longrightarrow \frac{\partial Loss}{\partial \hat{y}} = 2(\hat{y}-y)$$

$$\hat{y} = b + a_1 + a_2 \longrightarrow \frac{\partial \hat{y}}{\partial b} = \frac{\partial \hat{y}}{\partial a_1} = \frac{\partial \hat{y}}{\partial a_2} = 1$$

$$a_1 = w_1 x_1 \longrightarrow \frac{\partial a_1}{\partial w_1} = x_1$$

$$a_2 = w_2 x_2 \longrightarrow \frac{\partial a_2}{\partial w_2} = x_2$$

Putting everything together:

$$\frac{\partial Loss}{\partial b} = \frac{\partial Loss}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b} = 2(\hat{y}-y)\cdot 1$$

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1} \cdot \frac{\partial a_1}{\partial w_1} = 2(\hat{y}-y)\cdot 1 \cdot x_1$$

$$\frac{\partial Loss}{\partial w_2} = \frac{\partial Loss}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial w_1} = 2(\hat{y}-y)\cdot 1 \cdot x_2$$

You can check this matches the "old fashioned" calculation!!

OK, we are finally ready for backpropagation!! ☺

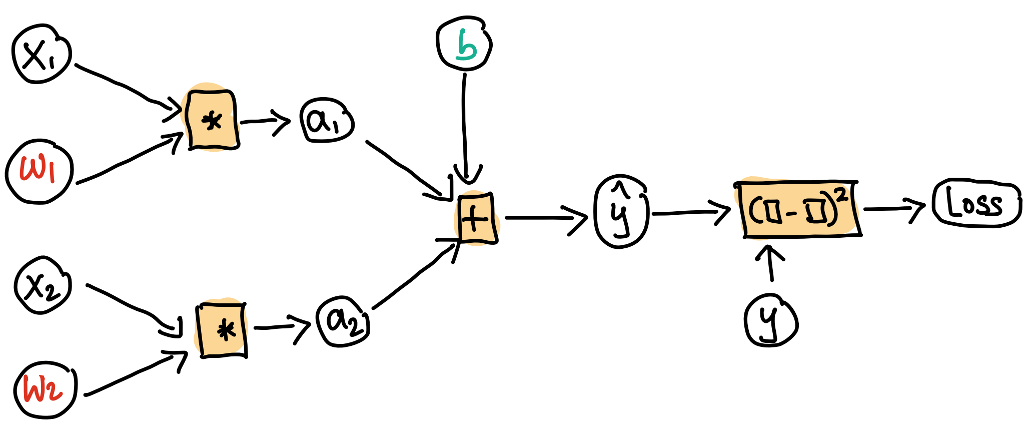
$$a_1 = w_1 x_1$$

$$a_2 = w_2 x_2$$

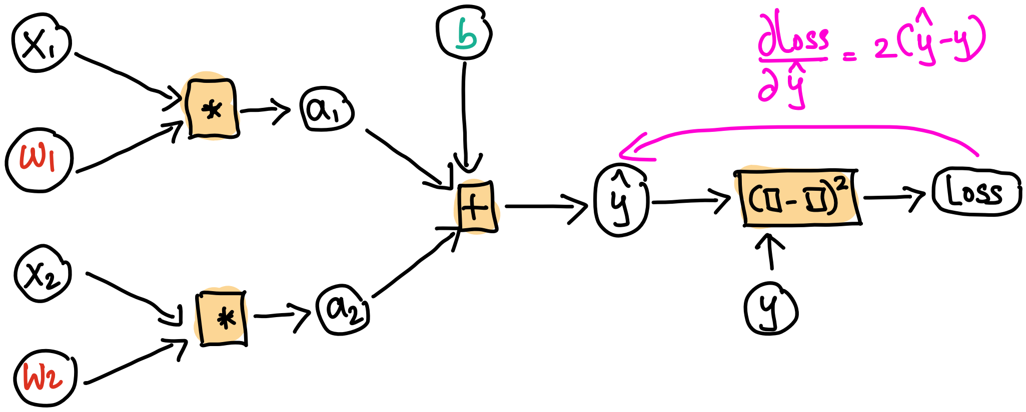
$$\hat{y} = b + a_1 + a_2$$

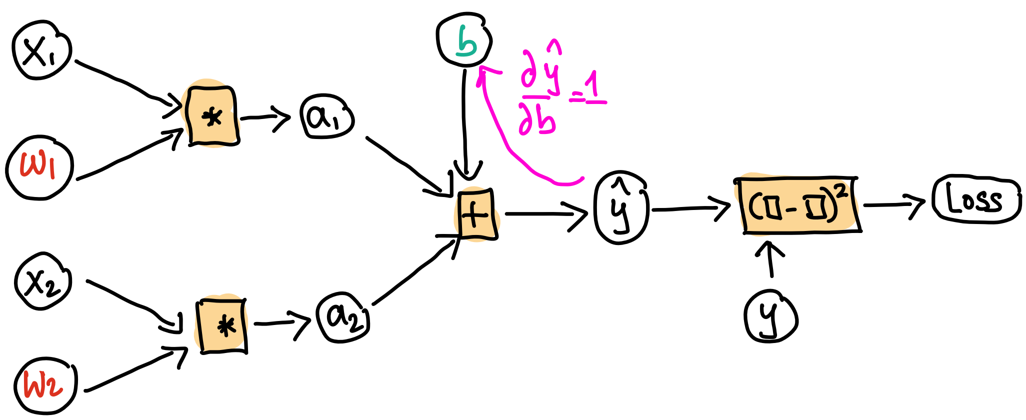
$$\text{Loss} = (\hat{y} - y)^2$$

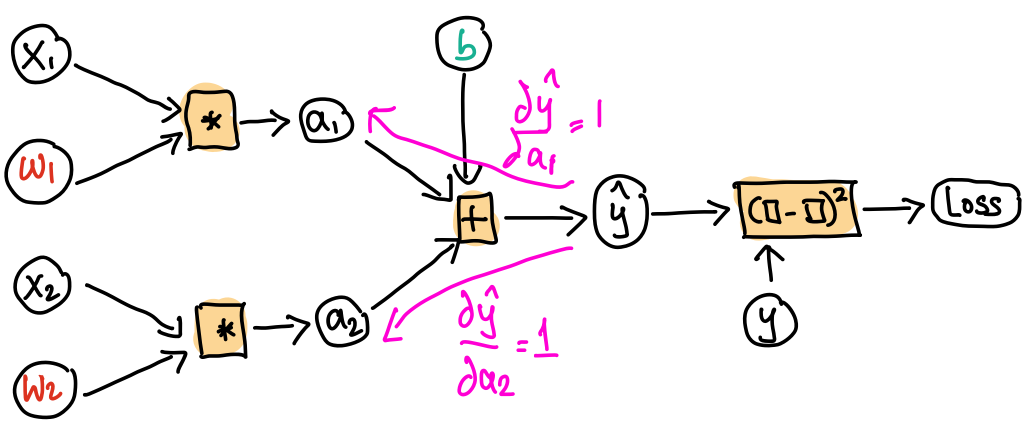
We will rewrite these equations as a computational graph

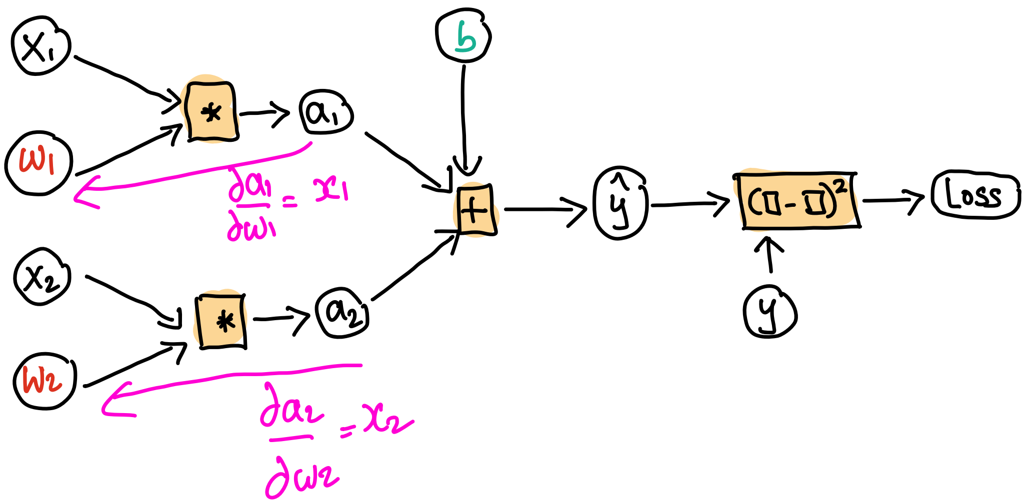


We can "attach" each of those little derivatives we calculated earlier to the graph.

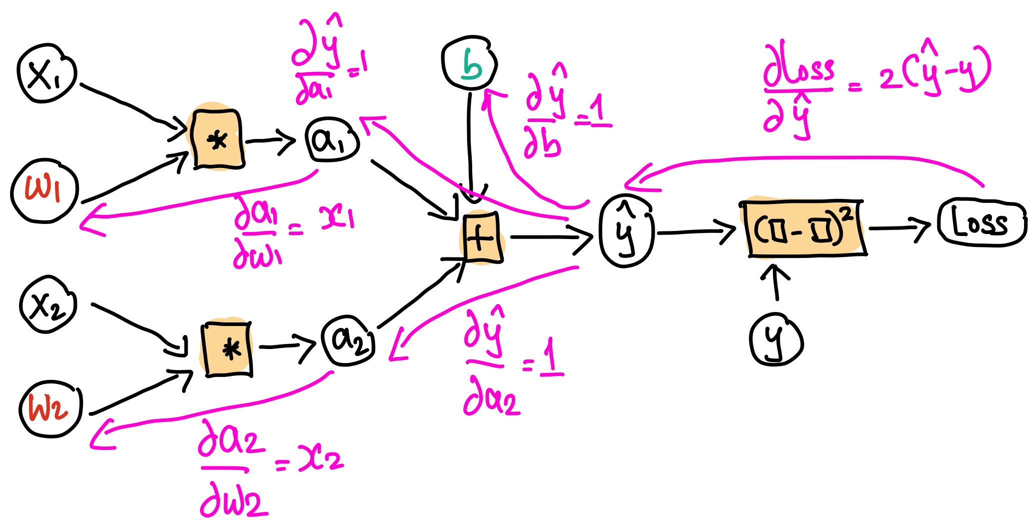








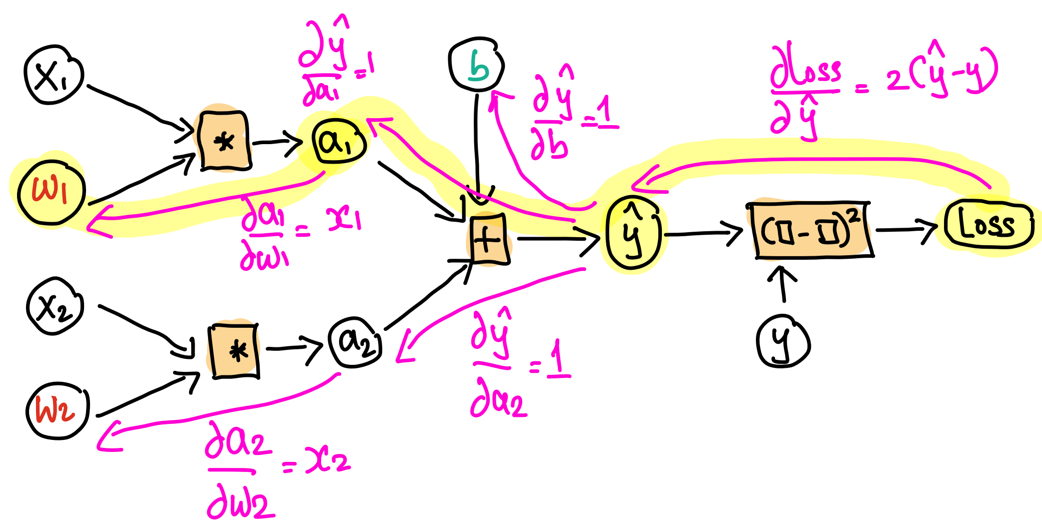
Putting everything together:



To calculate $\frac{\partial \text{Loss}}{\partial \text{ any parameter}}$, start from the loss and travel <u>backwards</u> to the parameter, multiplying the partial derivatives as you go.

This is called **BACKPROPAGATION**

To calculate $\frac{\partial Loss}{\partial w_1}$, for example:



Multiplying all the partial derivative on the yellow path, we get the answer:

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial loss}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1} \cdot \frac{\partial a_1}{\partial w_1} = 2(\hat{y} - y) \cdot 1 \cdot x_1$$

Does this match what we calculated earlier?

$\boxed{\text{YES!!}}$

Backprop is very efficient

- Calculate once and use many times (e.g. $\frac{d\text{Loss}}{d\hat{y}}$)
- (When more than one neuron in the layer) traversing backward is a series of matrix multiplications
- GPUs are perfect for matrix multiplications!!

$\boxtimes$