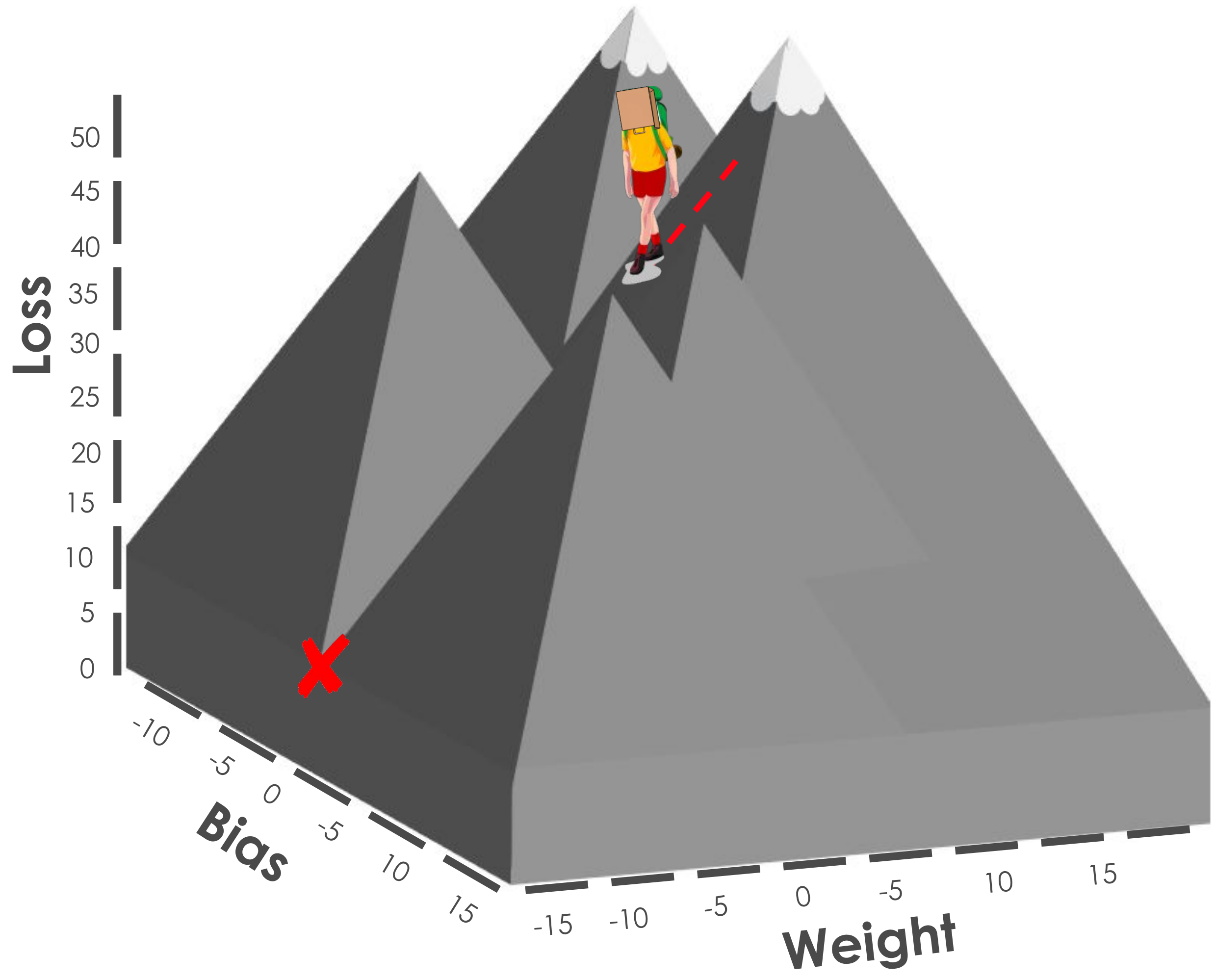
Calculating the Gradient

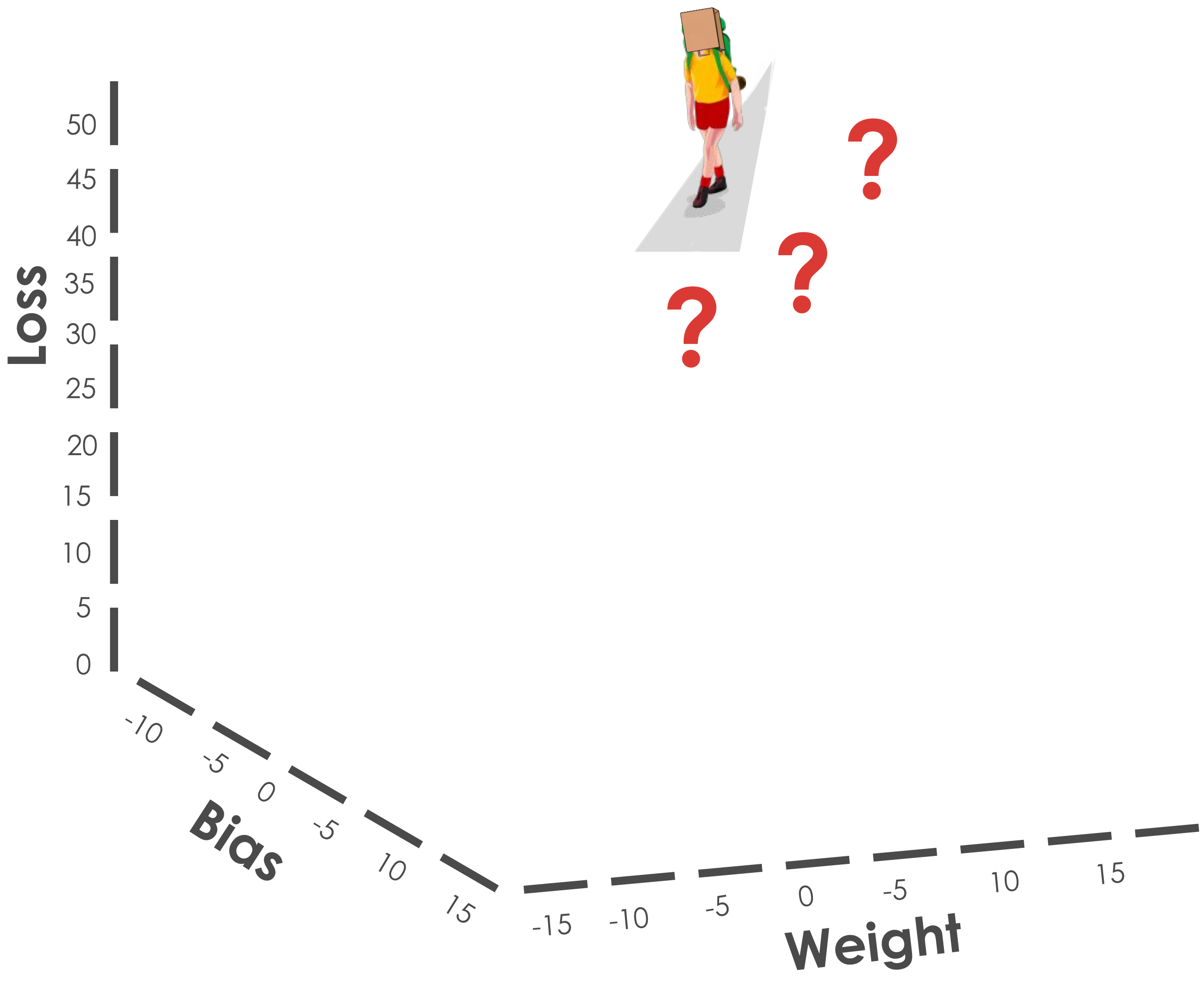
Deep Learning Pre-Work

Gradient Descent Steps



Step 1	Start at a random bias and weight and calculate the loss
Step 2	Take a step in the direction with the steepest gradient
Step 3	Calculate the new loss
Step 4	Repeat steps 2 and 3

Gradient Descent Steps



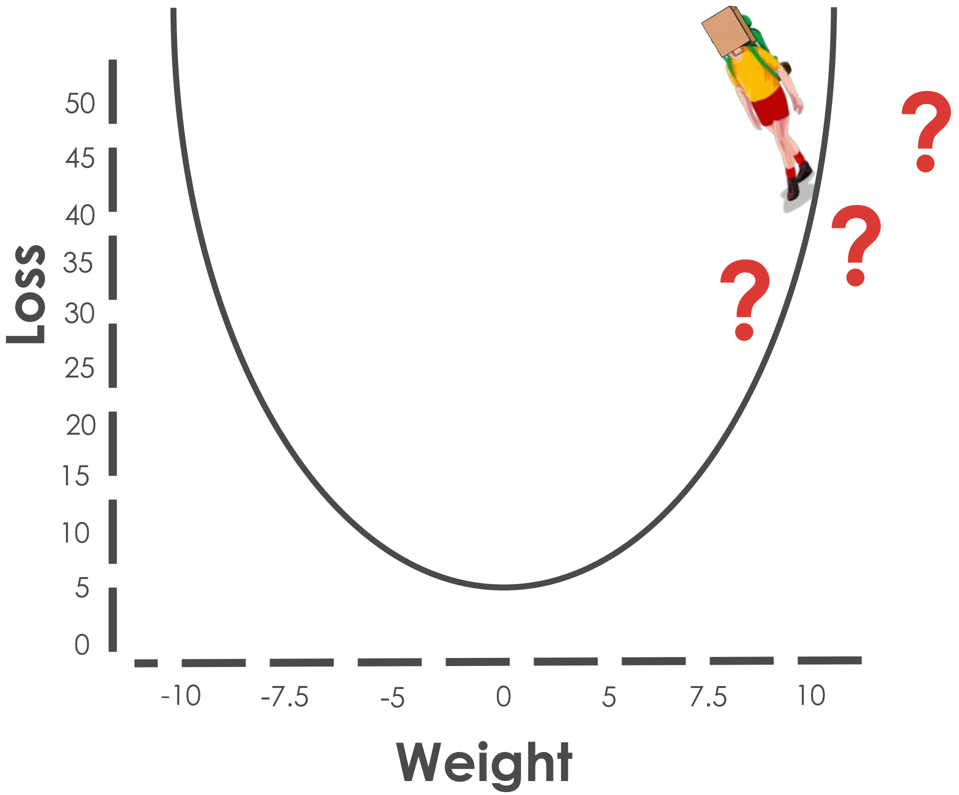
Step 1 Start at a random bias and weight and calculate the loss

Step 2 Take a step in the direction with the steepest gradient

Step 3 **Calculate the new loss**

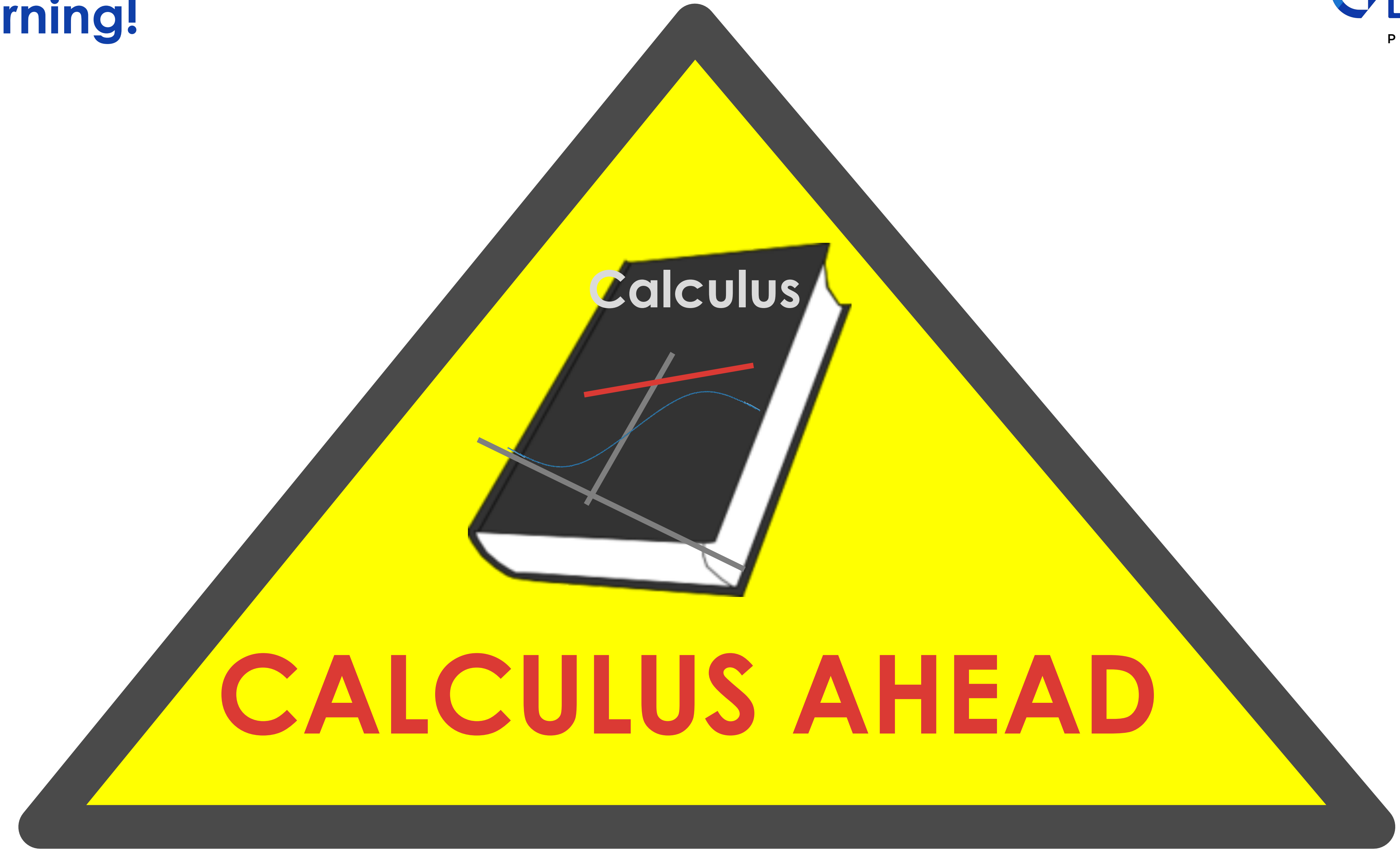
Step 4 **Repeat steps 2 and 3**

Gradient Descent Steps



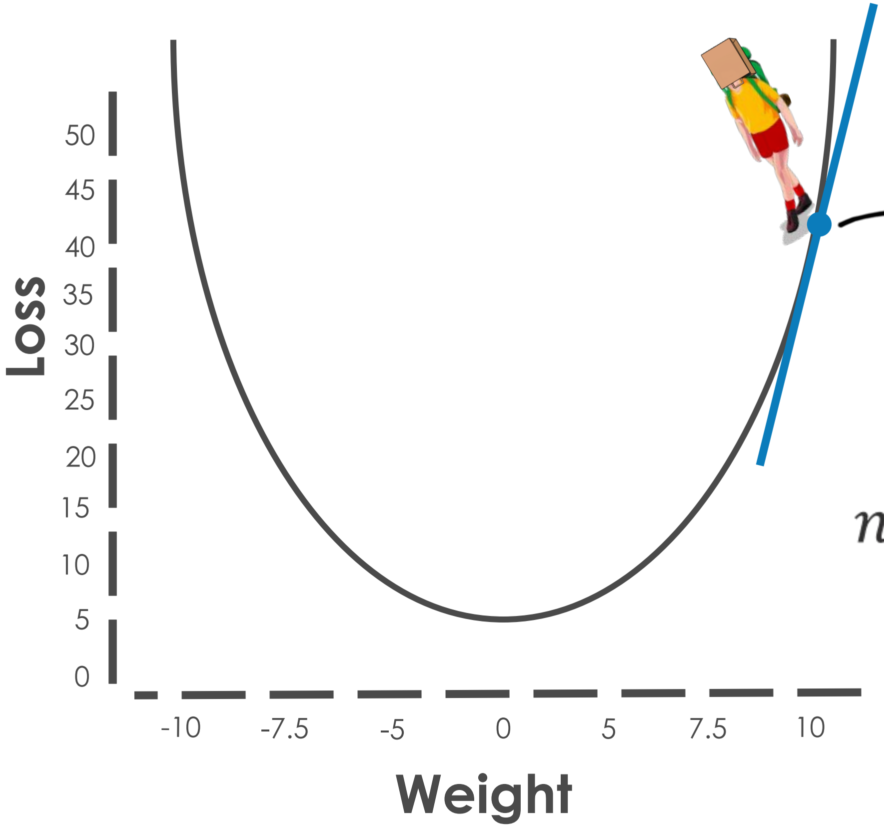
Step 1	Start at a random bias and weight and calculate the loss
Step 2	**Take a step in the direction with the steepest gradient**
Step 3	Calculate the new loss
Step 4	Repeat steps 2 and 3

Warning!



Derivatives

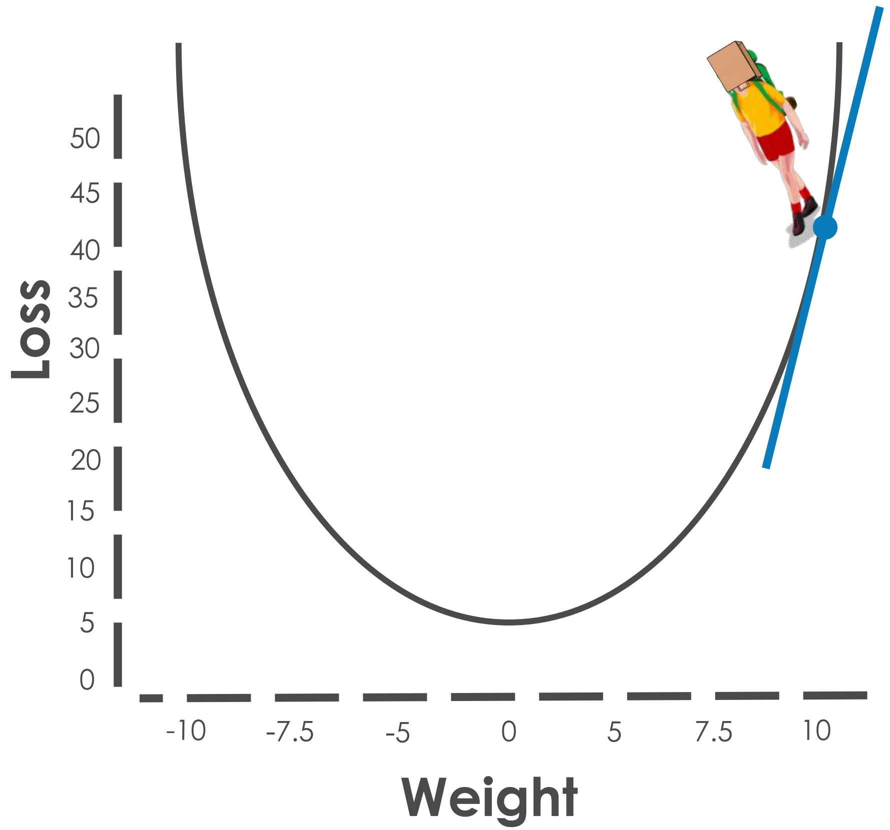
The derivative

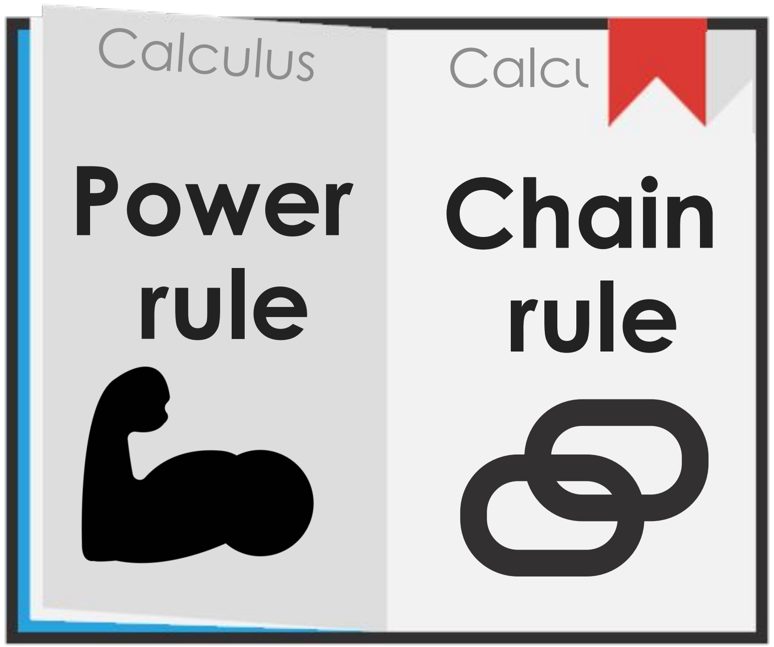


$$\frac{d(loss)}{d(weight)}$$

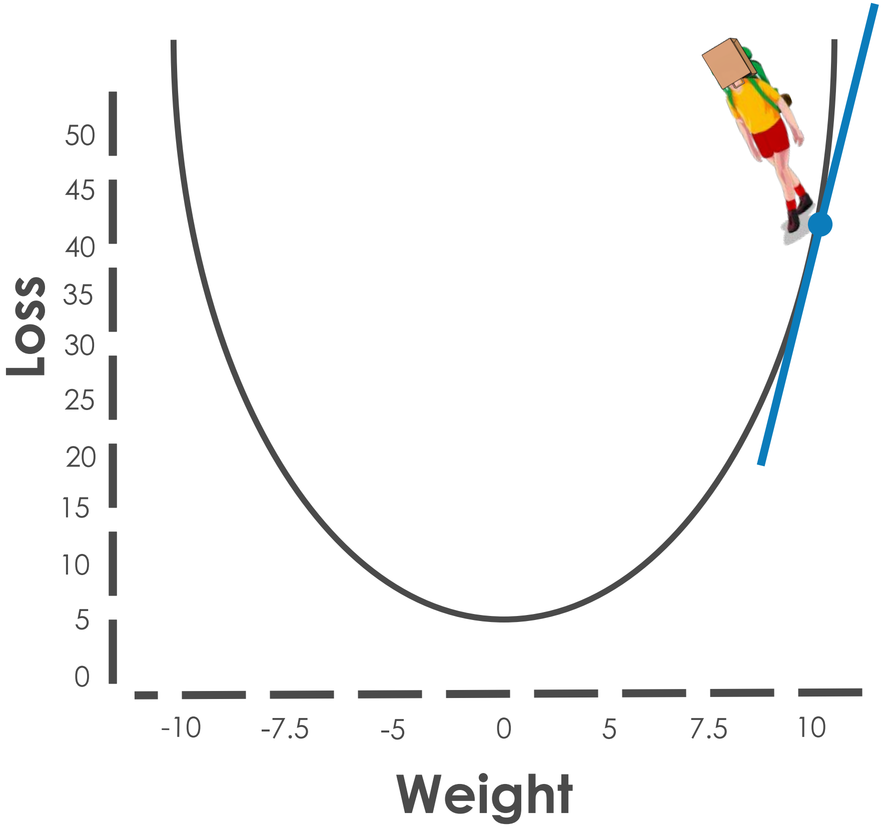
$$new\ weight = weight - \frac{d(loss)}{d(weight)} * learning\ rate$$

Calculating the derivative





Power Rule



$$\frac{d}{d(x)} x^n = nx^{n-1}$$

Power rule

mean, error, squared

$$loss = \frac{1}{n} \sum (\hat{y} - y)^2$$

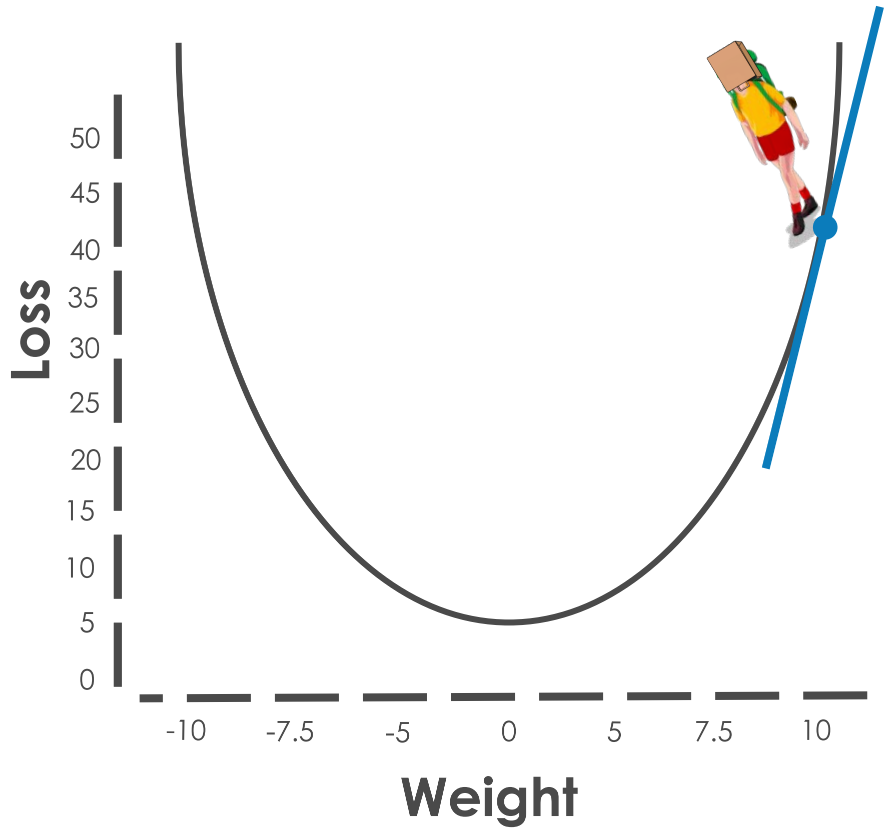
$$loss = error^2$$

$$\frac{d(loss)}{d(error)} = 2erro$$

$$\frac{d(loss)}{d(error)} = 2error$$

Power rule

Chain Rule



$$y = x$$
$$x = z$$

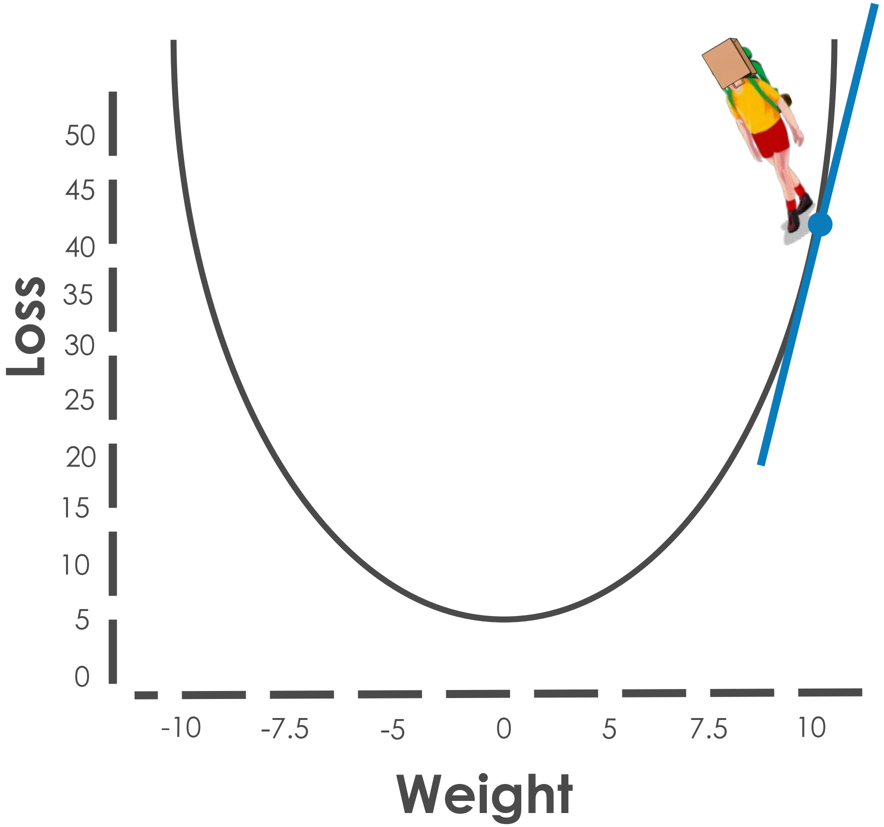
$$\frac{d(y)}{d(z)} = \frac{d(y)}{d(x)} * \frac{d(x)}{d(z)}$$

Chain rule

$$loss = error$$
$$error = weight$$

$$\frac{d(loss)}{d(weight)} = \frac{d(loss)}{d(error)} * \frac{d(error)}{d(weight)}$$

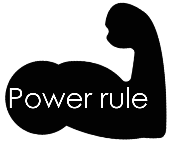
Calculating the gradient



$$\frac{d(loss)}{d(weight)} = \frac{d(loss)}{d(error)} * \frac{d(error)}{d(weight)}$$

$$2error \qquad error = x * weight \cancel{- y}$$

$$x * weight^1$$

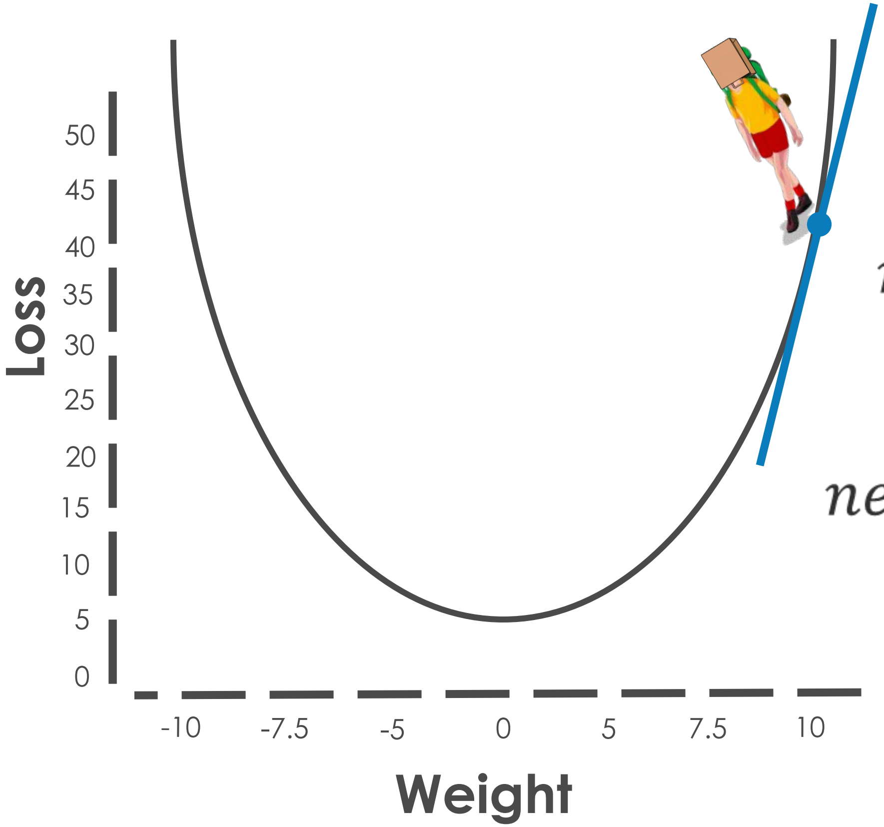


$$1 * x * weight^0$$

$$x$$

$$\frac{d(loss)}{d(weight)} = 2error * x$$

Updating the weight



$$\frac{d(loss)}{d(weight)} = 2error \; * \; x$$

$$new\ weight = weight - \frac{d(loss)}{d(weight)} * learning\ rate$$

$$new\ weight = weight - 2error \; * x \; * learning\ rate$$