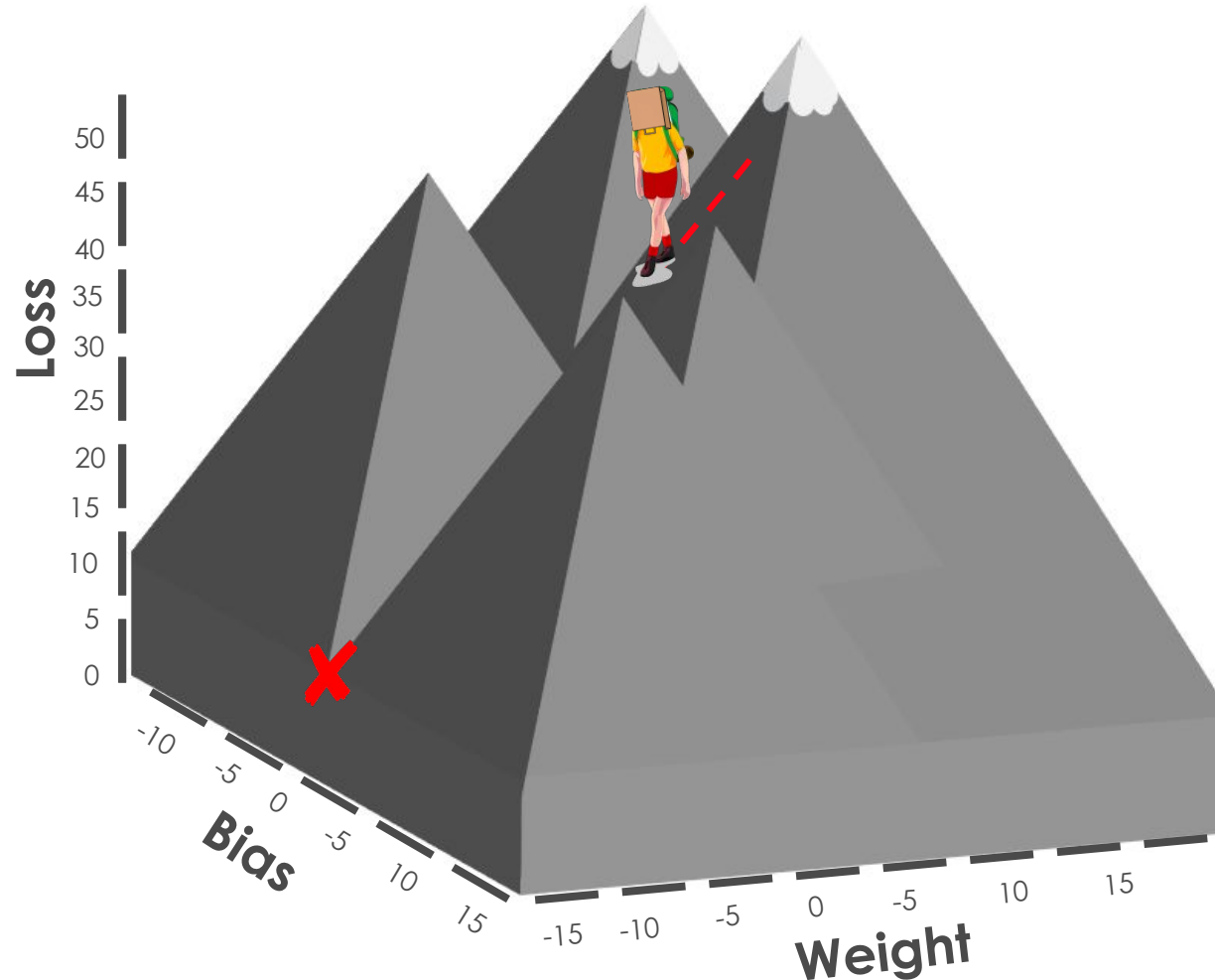




# Calculating the Gradient

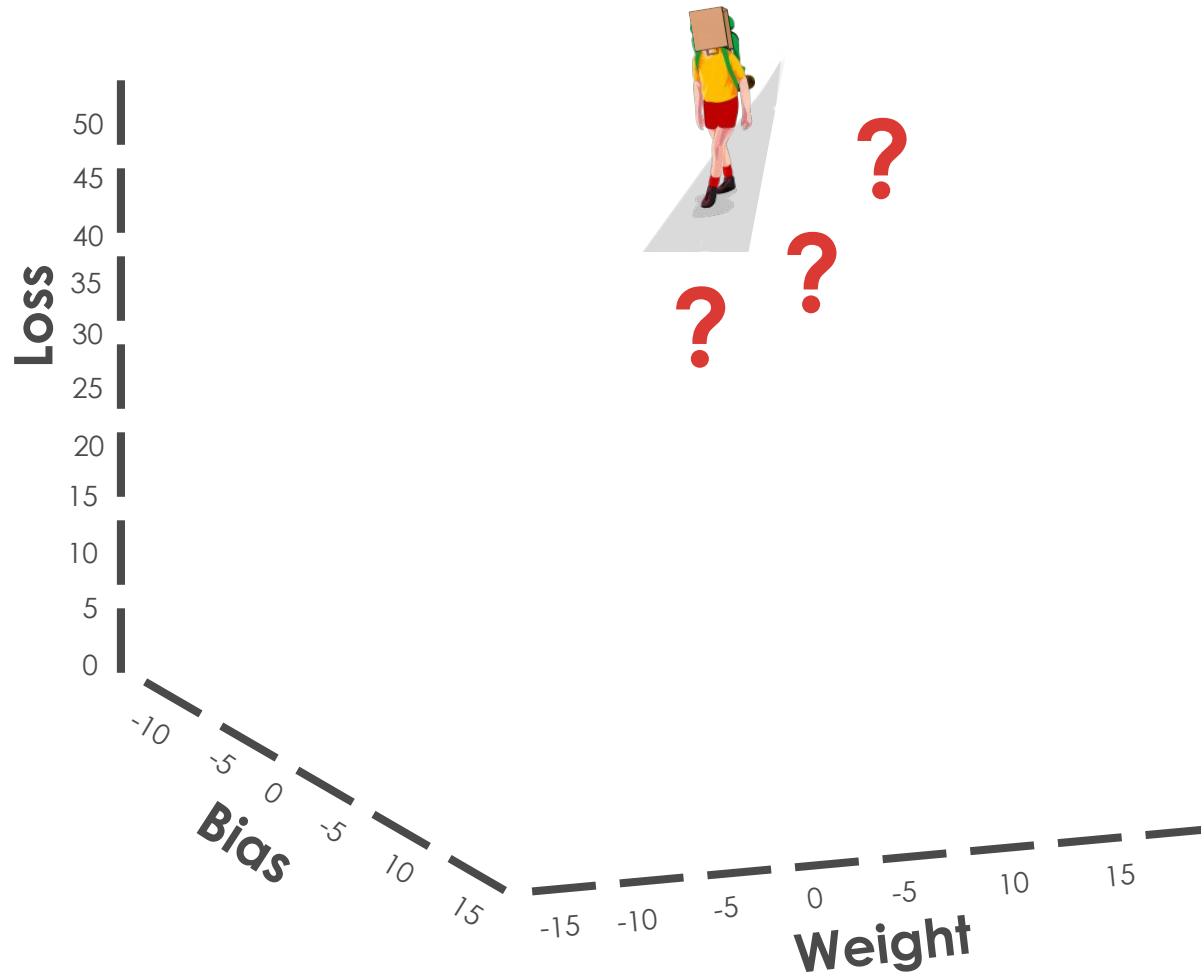
Deep Learning Pre-Work

# Gradient Descent Steps



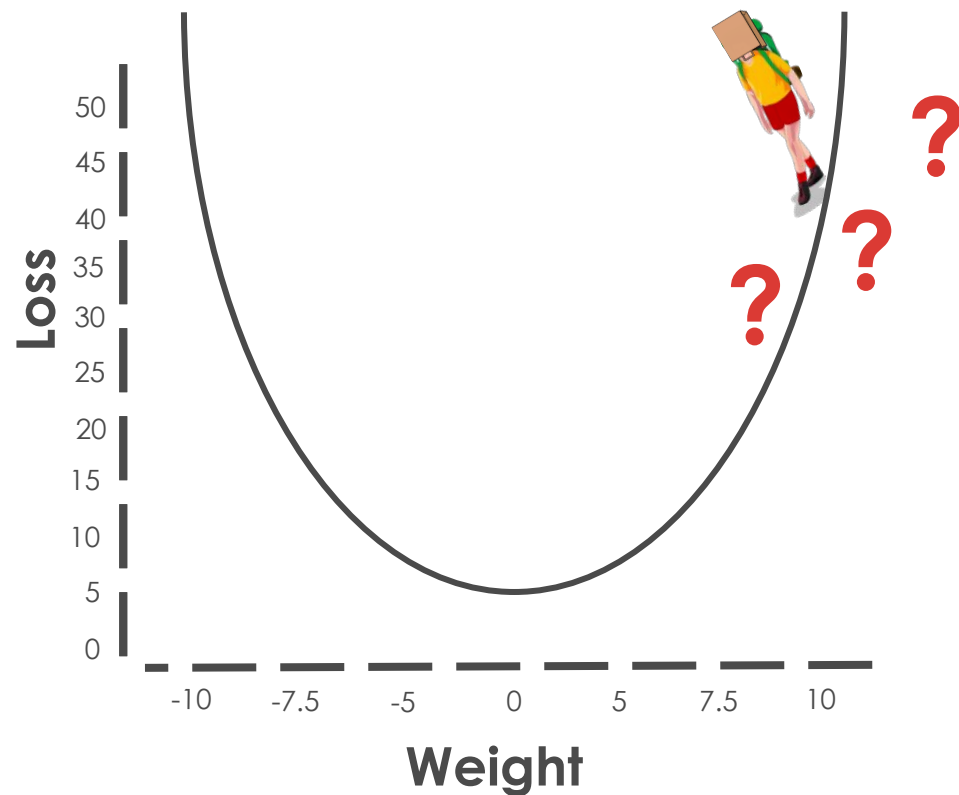
- Step 1** Start at a random bias and weight and calculate the loss
- Step 2** Take a step in the direction with the steepest gradient
- Step 3** Calculate the new loss
- Step 4** Repeat steps 2 and 3

# 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

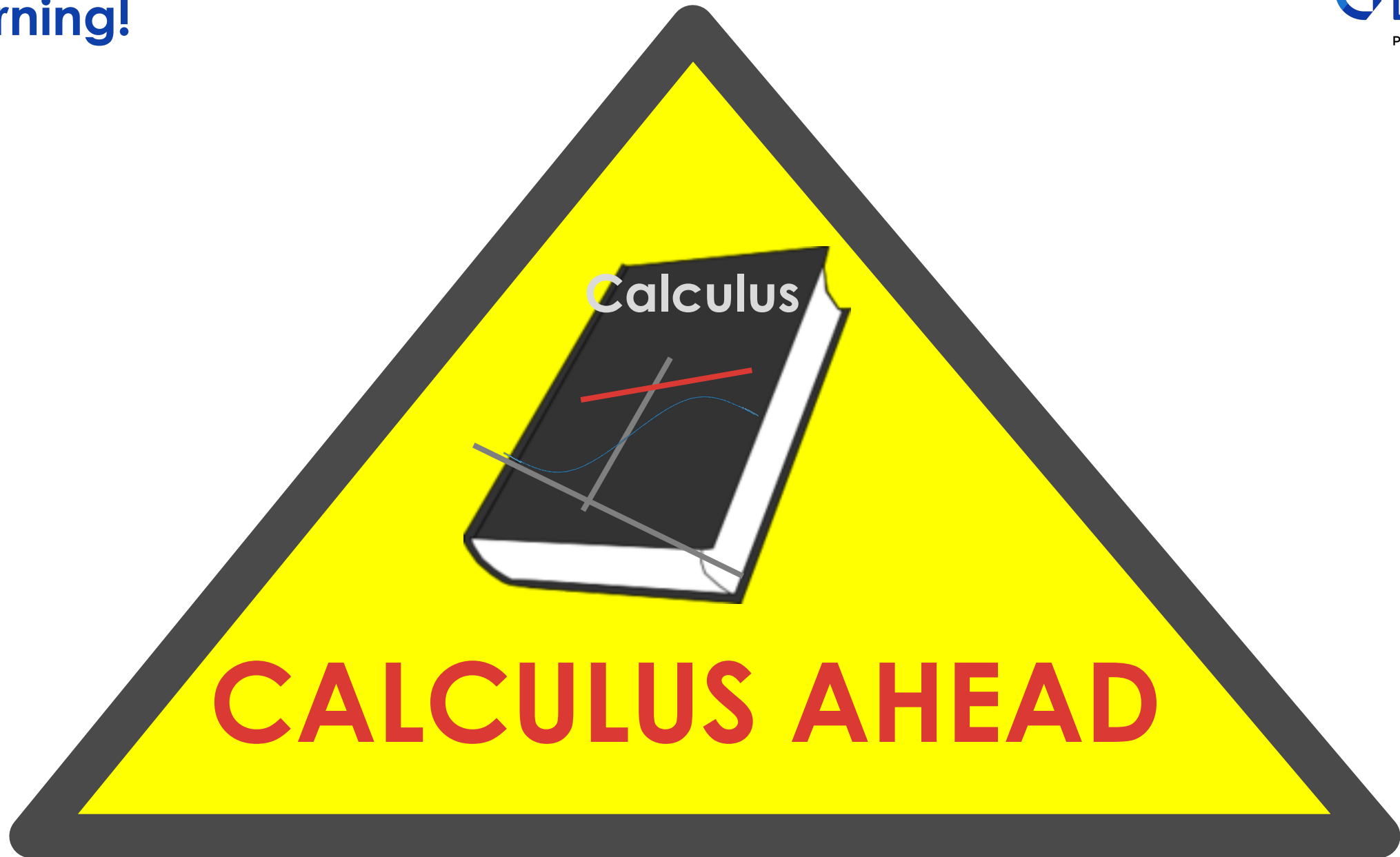


- Step 1 Start at a random bias and weight and calculate the loss
- Step 2 Take a step in the direction with the steepest gradient**
- Step 3 Calculate the new loss
- Step 4 Repeat steps 2 and 3

This file is meant for personal use by muratgguzel77@yahoo.com only.

Sharing or publishing the contents in part or full is liable for legal action.

Warning!



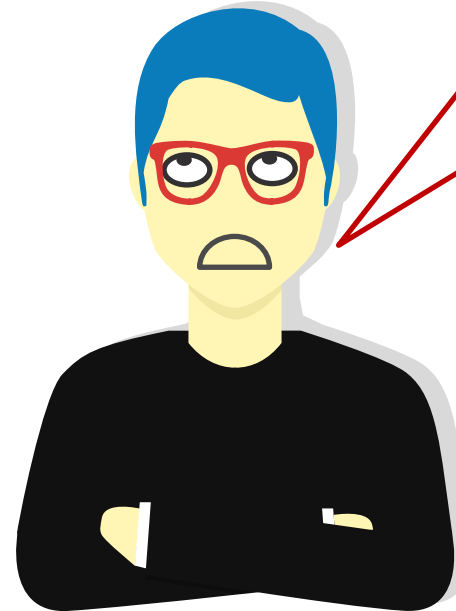
This file is meant for personal use by [muratgguzel77@yahoo.com](mailto:muratgguzel77@yahoo.com) only.

Sharing or publishing the contents in part or full is liable for legal action.

# Derivatives



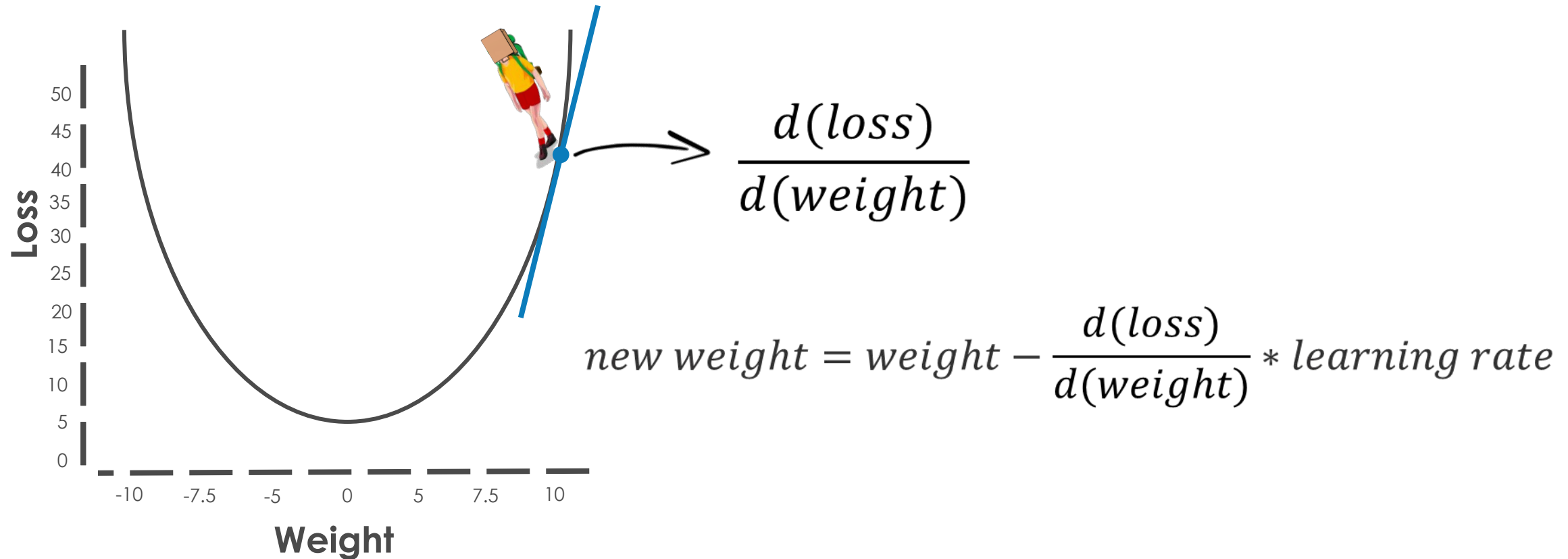
**So derivative**



This file is meant for personal use by muratgguzel77@yahoo.com only.

Sharing or publishing the contents in part or full is liable for legal action.

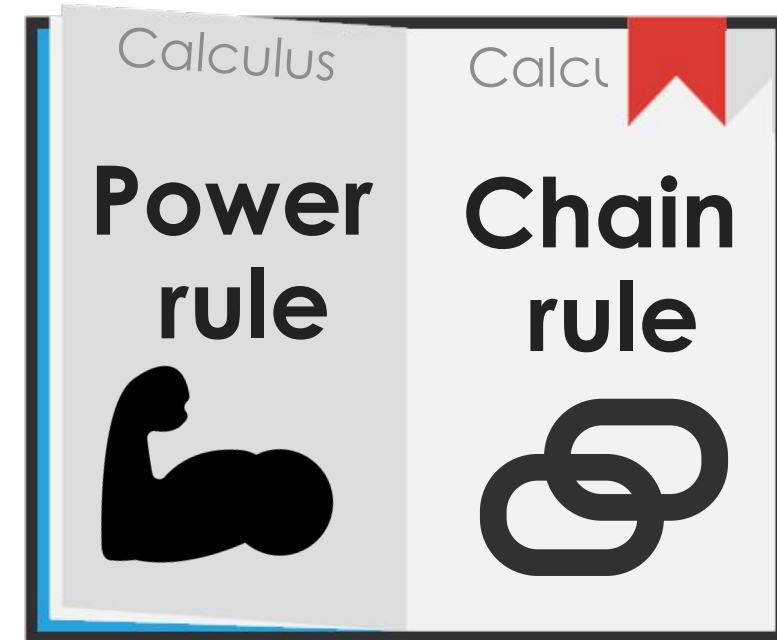
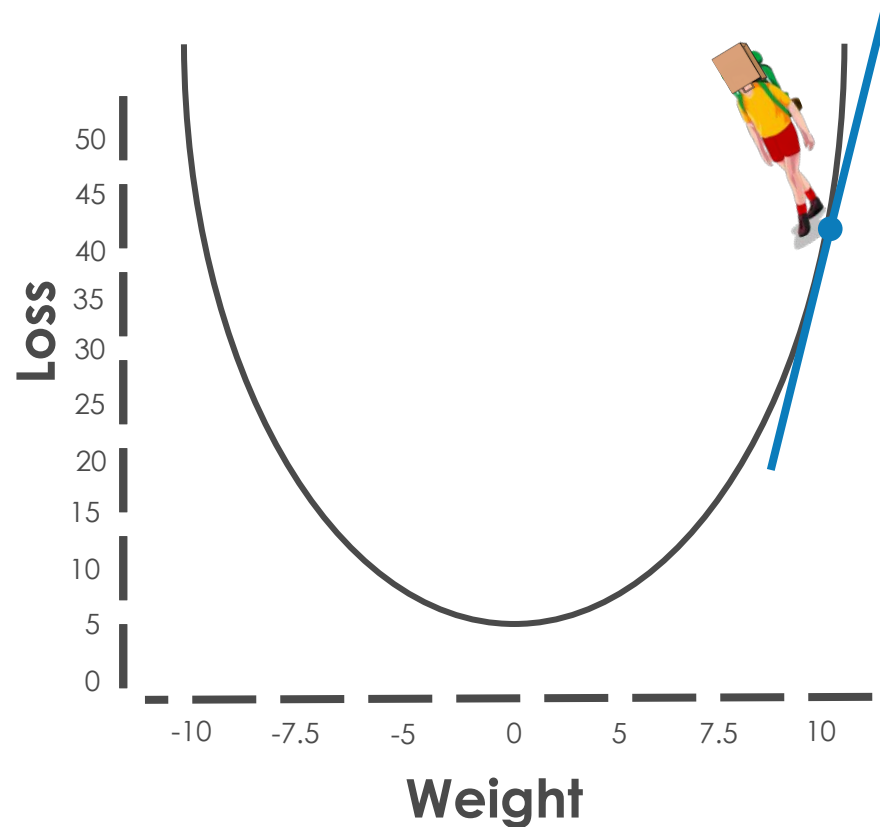
# The derivative



This file is meant for personal use by muratgguzel77@yahoo.com only.

Sharing or publishing the contents in part or full is liable for legal action.

# Calculating the derivative

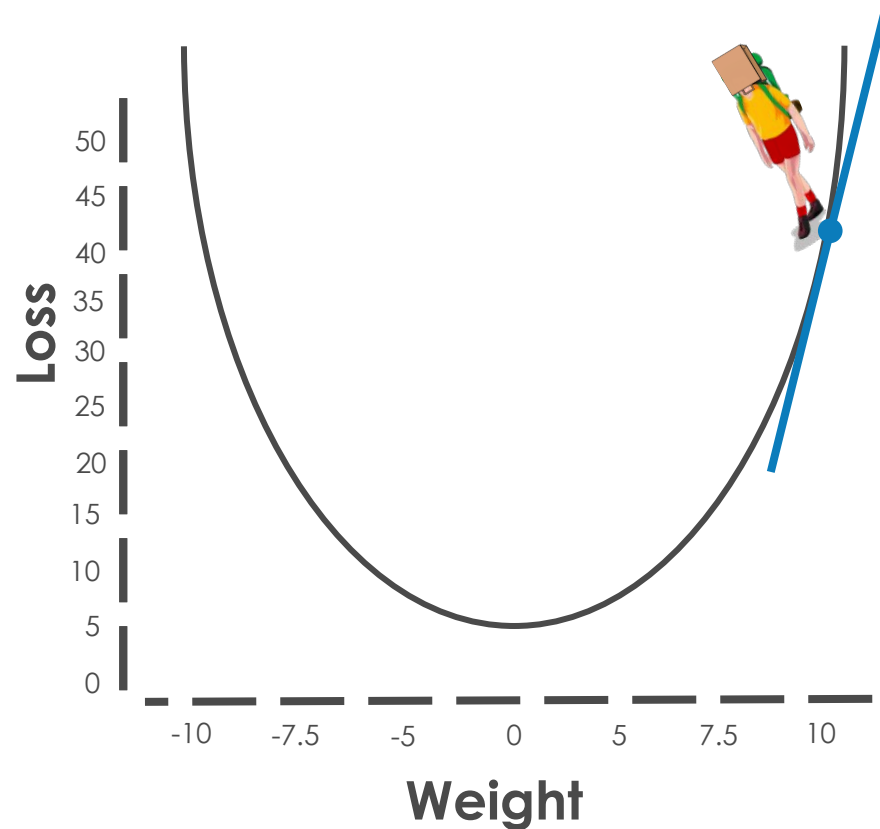


This file is meant for personal use by muratgguzel77@yahoo.com only.

Sharing or publishing the contents in part or full is liable for legal action.



# Power Rule



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



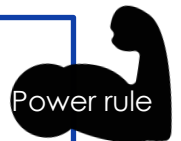
$$\text{loss} = \frac{1}{n} \sum (\hat{y} - y)^2$$

mean ← error → squared

$$\text{loss} = \text{error}^2$$

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

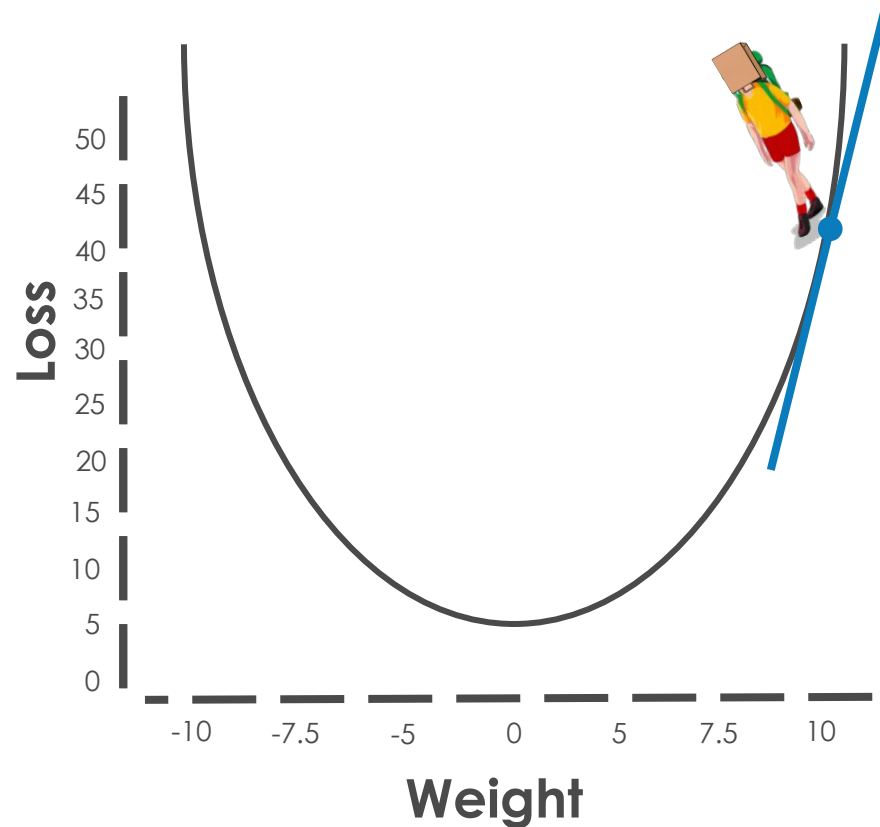
$$\frac{d(\text{loss})}{d(\text{error})} = 2\text{error}$$



This file is meant for personal use by muratgguzel77@yahoo.com only.

Sharing or publishing the contents in part or full is liable for legal action.

# Chain Rule



$$\begin{aligned} y &= x \\ x &= z \end{aligned}$$

Chain rule



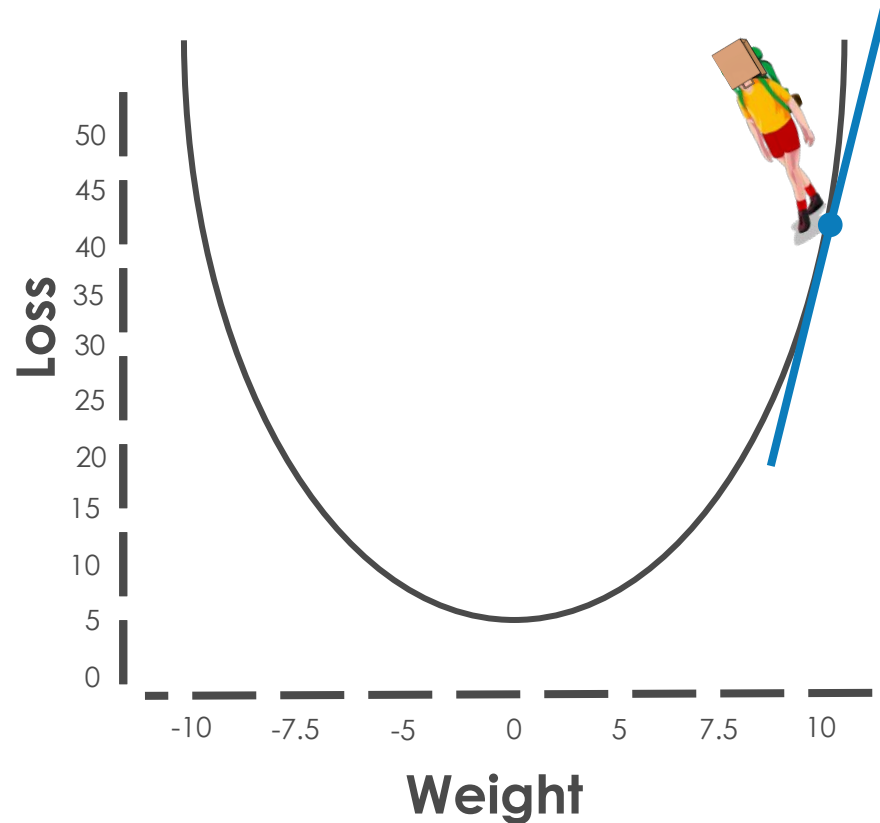
$$\frac{d(y)}{d(z)} = \frac{d(y)}{d(x)} * \frac{d(x)}{d(z)}$$

$$loss = error$$

$$error = weight$$

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

# Calculating the gradient



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

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

$x * \text{weight}^1$

**Power rule**       $1 * x * \text{weight}^0$

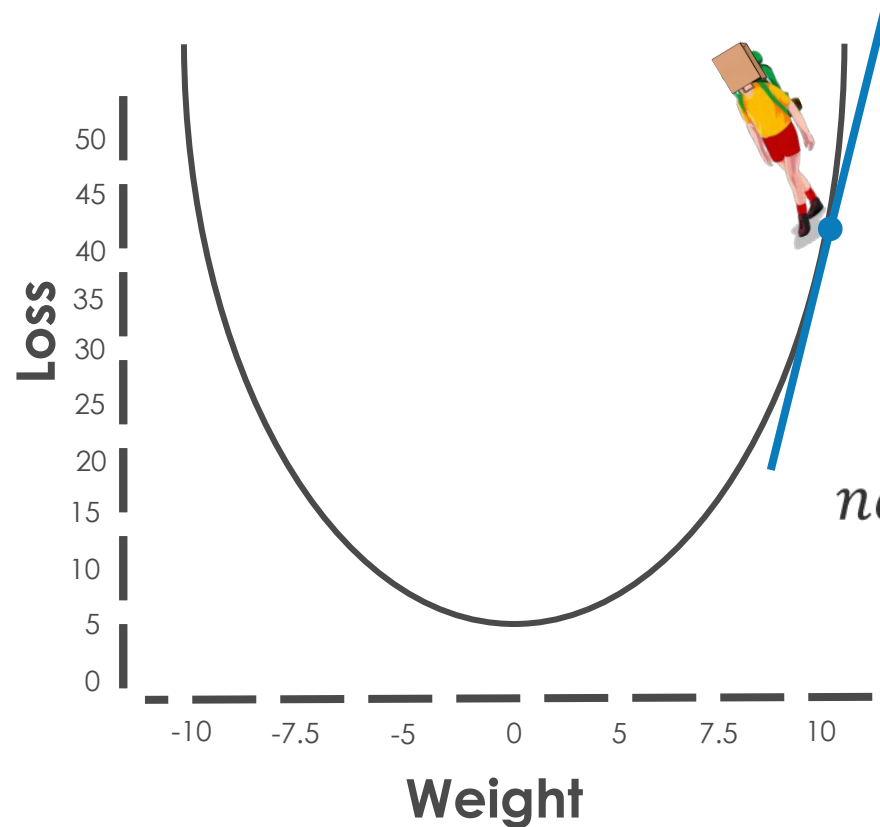
$x$

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

This file is meant for personal use by muratgguzel77@yahoo.com only.

Sharing or publishing the contents in part or full is liable for legal action.

# Updating the weight



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

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

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