MODERN COMPUTER VISION

BY RAJEEV RATAN

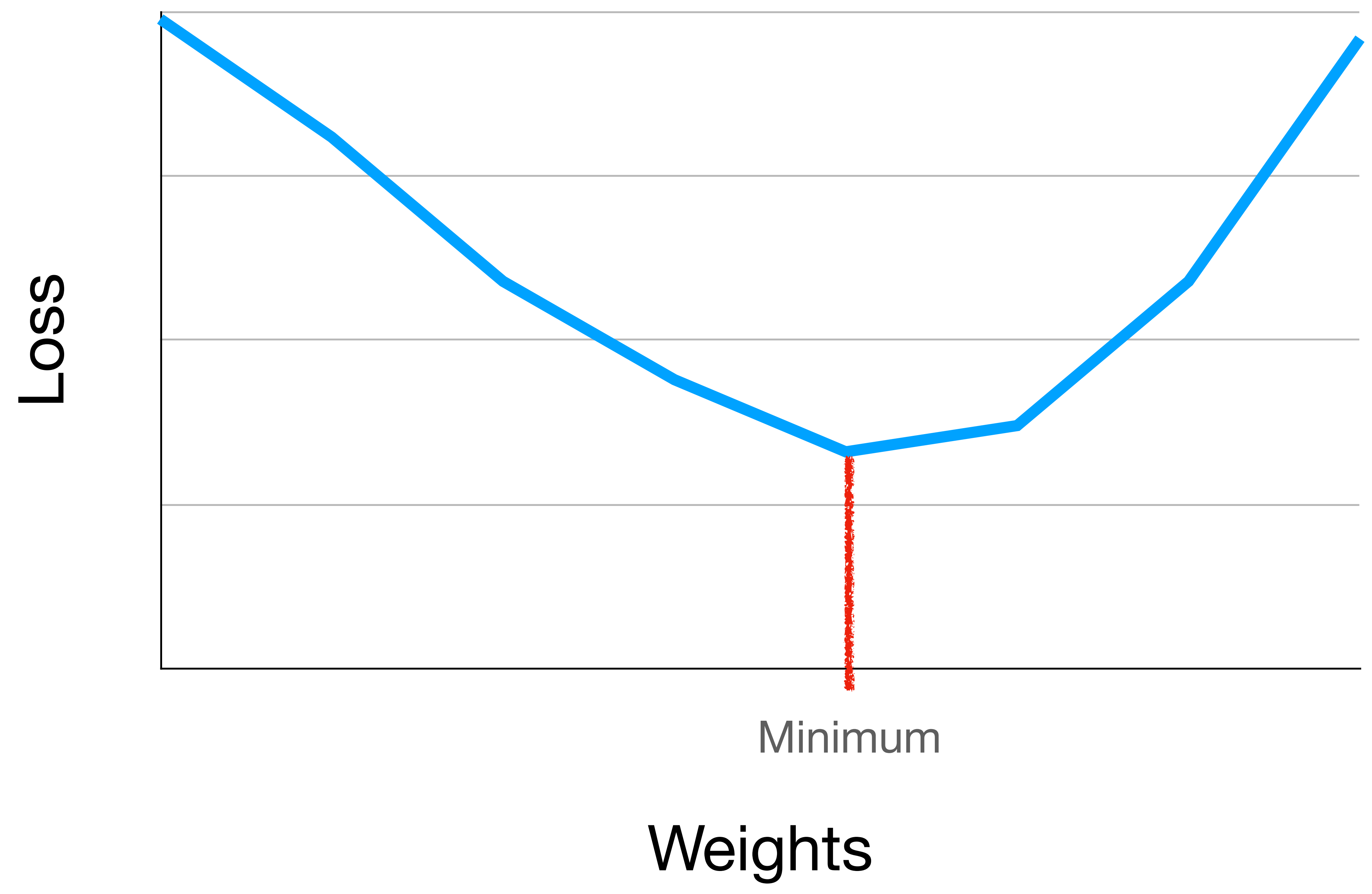
Gradient Descent

Finding the optimal weights

Loss Functions

How do we find the lowest loss?

- Back Propagation is the process we use to update the individual weights or gradients
    - $wx + b$
- Our goal is finding the right value of weights where the loss is lowest
- The method by which we achieve this goal (i.e. updating all weights to lower the total loss) is called **Gradient Descent**
- It's the point at which we find the **optimal weights** such that **loss is near lowest**



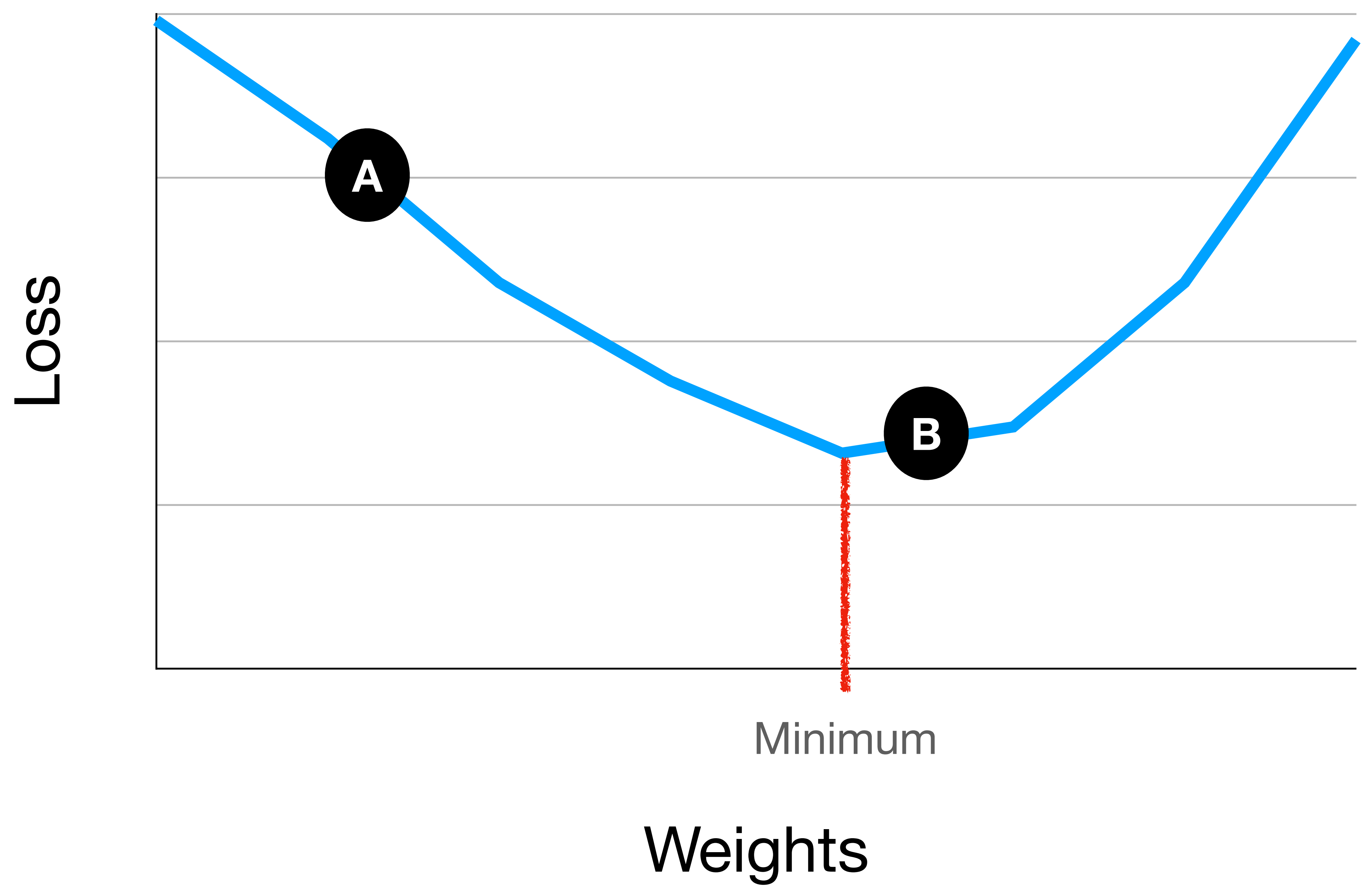
Gradients

Gradients are the derivative of a function

- It tells us the rate of change of one variable with respect to the other e.g.

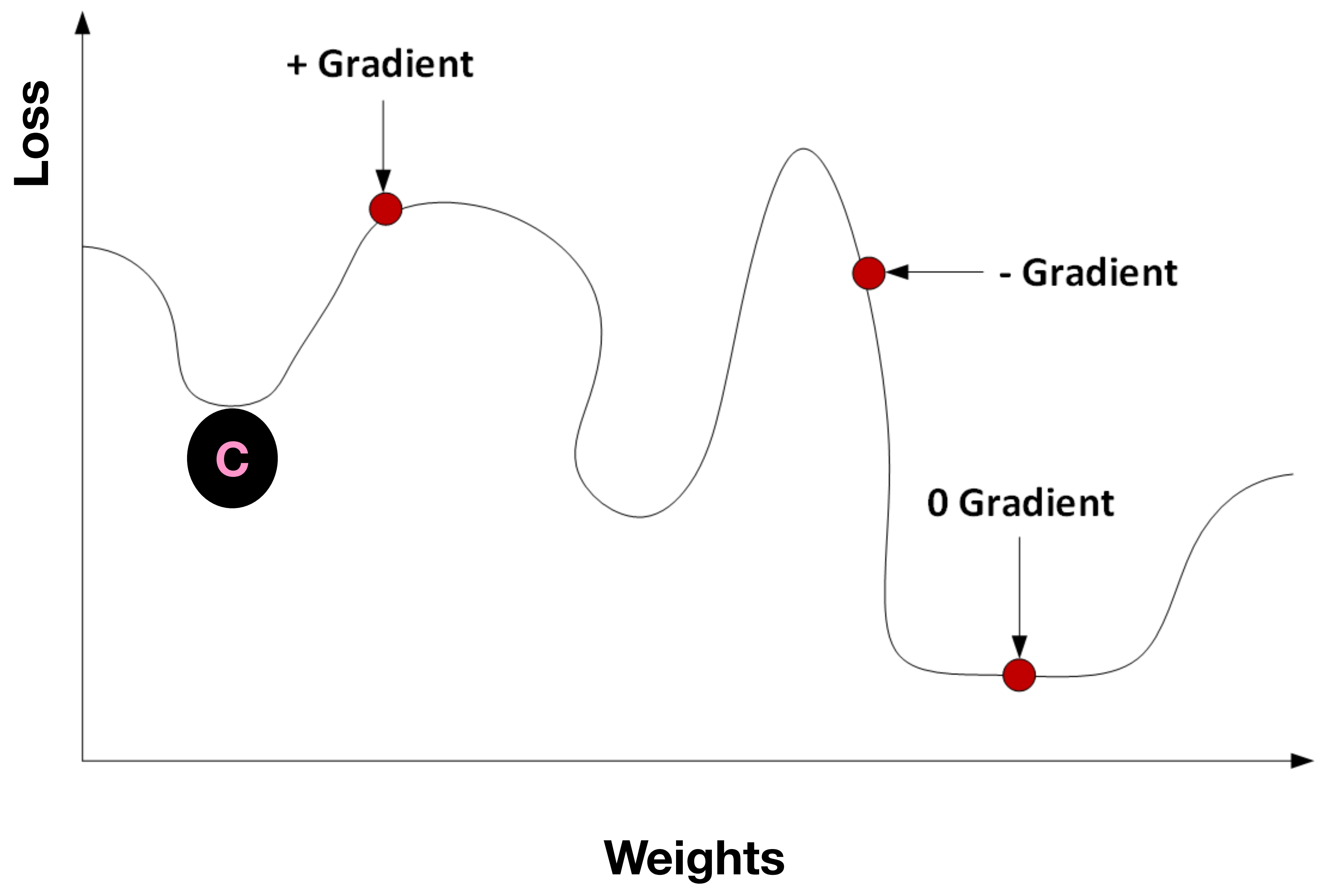
  $Gradient = \dfrac{dE}{dw}$, where E is the Error or Loss and w is the weight

- A positive gradient means, loss increases if weights increase
- A negative gradient means, loss decreases if weights increase
- At point A, moving right increases our weights and decreases our loss, -ve
- A point B, moving right increases our weights and increases our loss, +ve
- Therefore, the **negative** of our gradient tells us the direction we should be moving



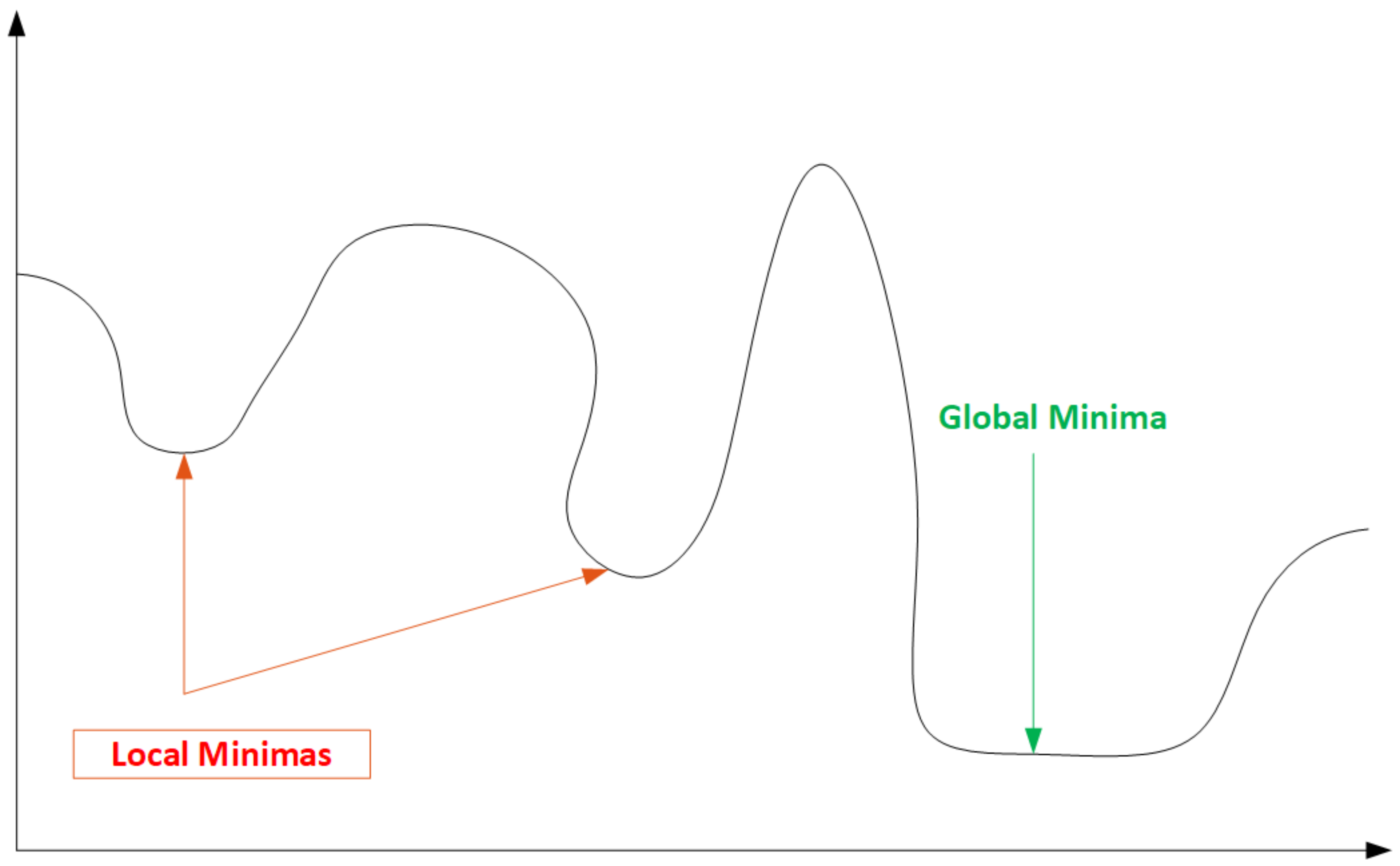
More on Gradients

- The point at which a Gradient is zero means that small changes to the left or right don't change the loss
- In training Neural Networks this is good and bad
- At point, C, very small changes to the left or right don't change the Loss
- This means, our network gets stuck during training.
- This is called getting **stuck in a Local Minima**



Local and Global Minimas

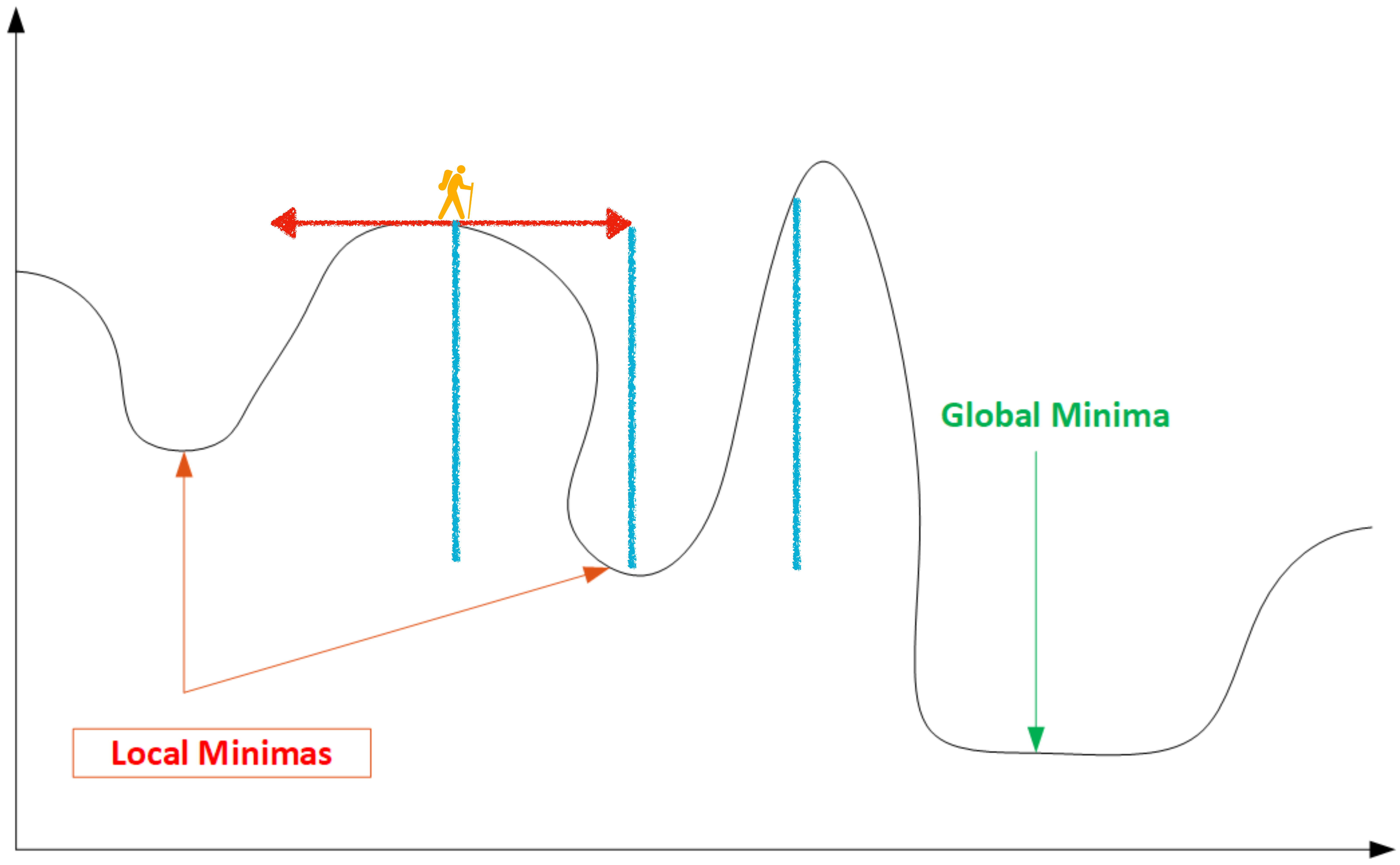
- We always want to find the Global Minima during training.
- That is the point where the combination of weights give the lowest loss



Gradient Descent

- Imagine being a really tiny person and you're traversing down the slope of this old, rough bowl.
- There'd be peaks, valleys, troughs etc.
- How do you know when you're truly at the bottom?
- Possibly take large steps so you don't get stuck in a valley
- But then you risk jumping over the Global Minima

Step Size is Important



Learning Rates

Recall our Back Propagation Weight Update Formula

- $W_5 = -\lambda \times \frac{dE_T}{dW_5}$
- λ is our learning rate
- Learning Rates allow us to adjust the magnitude of jumps in weights
- Finding an optimal value will avoid us finding Local Minimums and while preventing us from jumping over the Global Minimum.

Gradient Descent Methods

- **Naive Gradient Descent** - Passes the entire dataset through our network then updates the weights.
    - It is computationally expensive and slow.
- **Stochastic Gradient Descent (SGD)** - Updates weights after each data sample (image) is forward propagated through our network.
    - This leads to noisy fluctuating loss values and is also slow to train
- **Mini-Batch Gradient Descent** - Combines both methods, it takes a batch of data points (images) and forward propagates all, then updates the gradients.
    - This leads to faster training and convergence to the Global Minima
    - Batches are typically 8 to 256 in size

Next…

Optimisers