Gradient Descent

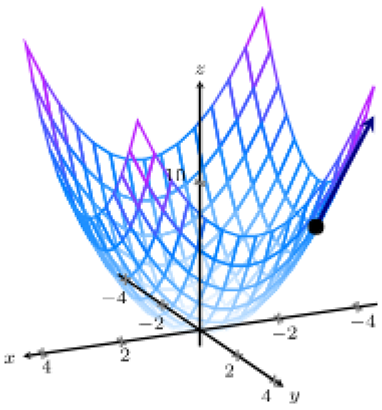
- What is a Gradient in math
- What is Gradient Descent
- Gradient Descent Algorithm
- Why Gradient Descent

What is Gradient in math

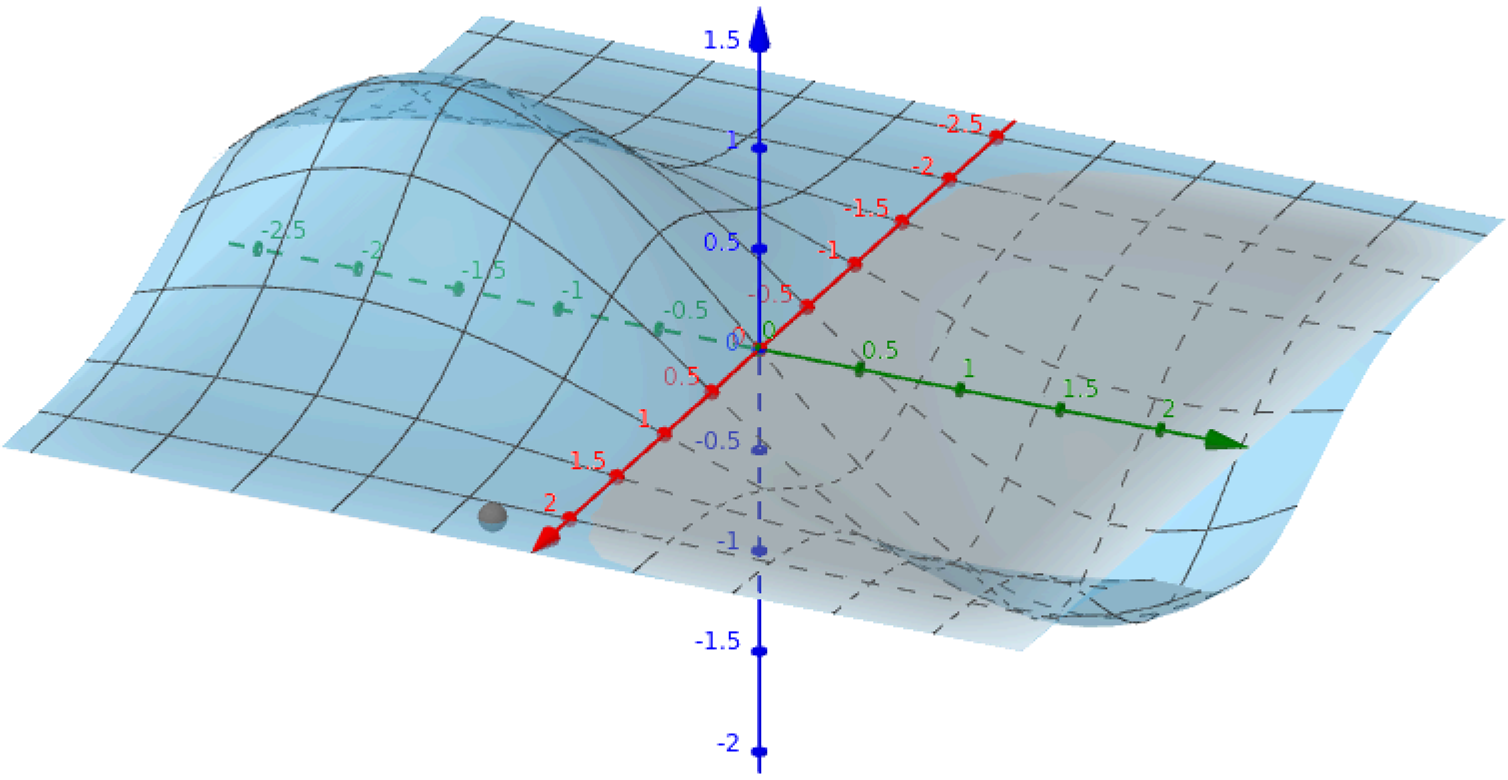
A gradient is a **vector** of partial derivatives for a **multivariate** function $f(x_1, x_2, \dots, x_n)$ with respect for each variable x_i

$$\nabla f = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots \frac{\partial f}{\partial x_n})$$

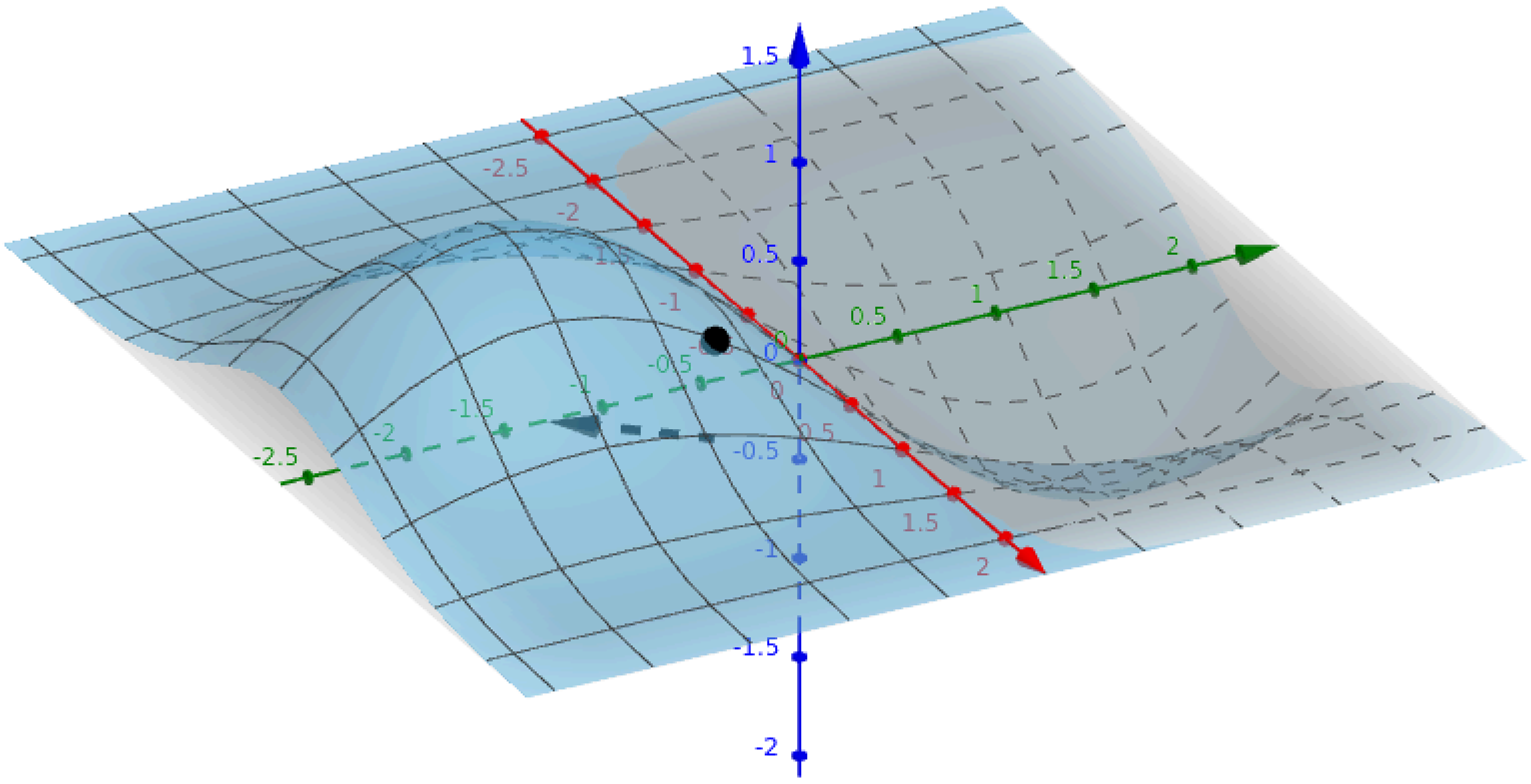
- The Gradient vector points in the direction of the **maximum** change (increase), formally **steepest ascent**
- Its magnitude is equal to the **maximum rate** of change



- A zero gradient $\nabla f = 0$ indicates a **critical point**



- The Gradient measures the "slop" in all directions our point here is on a *flat surface* the gradient vector is zero



- Unlike here where there is a slight slop the gradient vector ∇f points to the direction of the **steepest ascent**

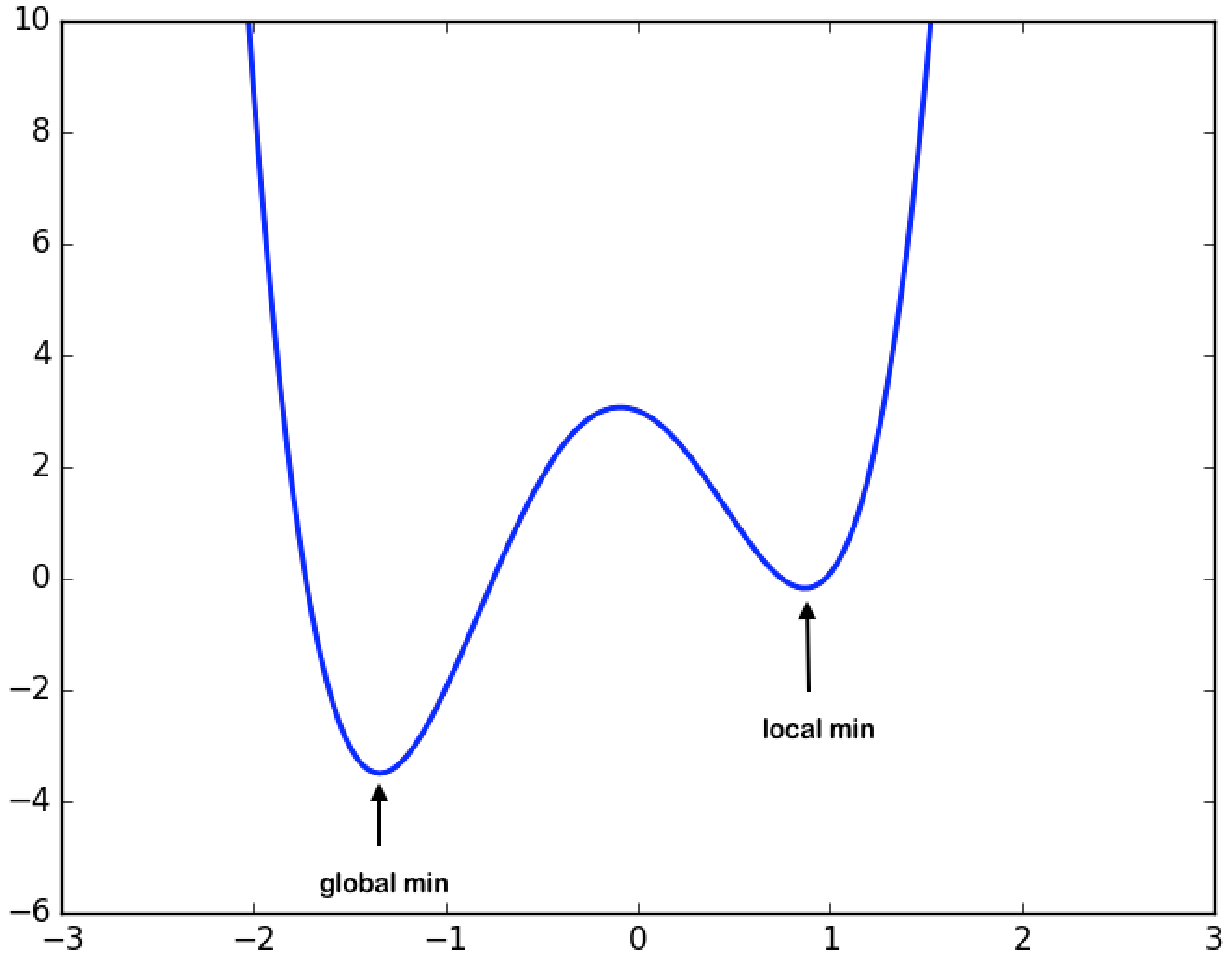
Gradient Rules

1. Product Rule
   $\nabla(f.g) = f\nabla g + g\nabla f$
2. Quotient Rule
   $\nabla\left(\frac{f}{g}\right) = \frac{g\nabla f - f\nabla g}{g^2}$
3. Gradient of a Norm
   $\nabla||x|| = \frac{x}{||x||}$
4. Directional Derivative Connection
   $D_{\hat{u}}G = \nabla G.\hat{u}$

What is Gradient Descent ?

Simply its an **optimization** algorithm used to **train machine learning** models, by optimizing the **parameters** of the model **iteratively** to find the **global minimum** of a loss function curve

More formally the Gradient Descent is an algorithm that essentially computes the gradient ∇ of the cost loss function (MSE in Linear Regression) $*h*$



- That is the lowest point in a function curve

Intuition :

Imagine a person(**gradient descent algorithm**) is stuck in a foggy mountain(**loss function curve**) and he is trying to get down (finding the **global minimum**). Therefore the person need to use local information and **calculations** and what's visible to descent down , using the gradient descent which says look at your current position and goes into the direction of the steepest descent

- **Fog** $\rightarrow$ Limited, local information
- **Slop** $\rightarrow$ Gradient
- **Step size** $\rightarrow$ Learning rate

Gradient Descent Algorithm

Generally the Gradient Descent follows These steps :

1. Initialize θ (randomly)
2. While not converged :
   1. Computer gradient : $\nabla J(\theta)$
   2. Update parameters : $\theta^{+} = \theta^{-} - \alpha \nabla_{\theta} J(\theta)$
   3. Check convergence **optional**
3. Return optimized θ

Note :

- $J(\theta)$ is a the **Loss Function**
- α is the **Learning Rate** which is the step size
- Batch size $\rightarrow$ its the trade off between speed and stability