GRADIENT DESCENT

- **Optimization** is a big part of **machine learning**. Almost Every machine learning algorithm has an optimization algoritms.
- Gradient Descent is best use when the **parametrs cannot be calculated analytically** or when you need an optimized way to calculate those parameters.
- It is used to **find the values of parameters (coffecient) of a function that minimize the cost function.**
- It is an **iterative algorithm** use in **loss function to find global minima** and gradient descent algorithm is most **popular in machine learning and deep learning**

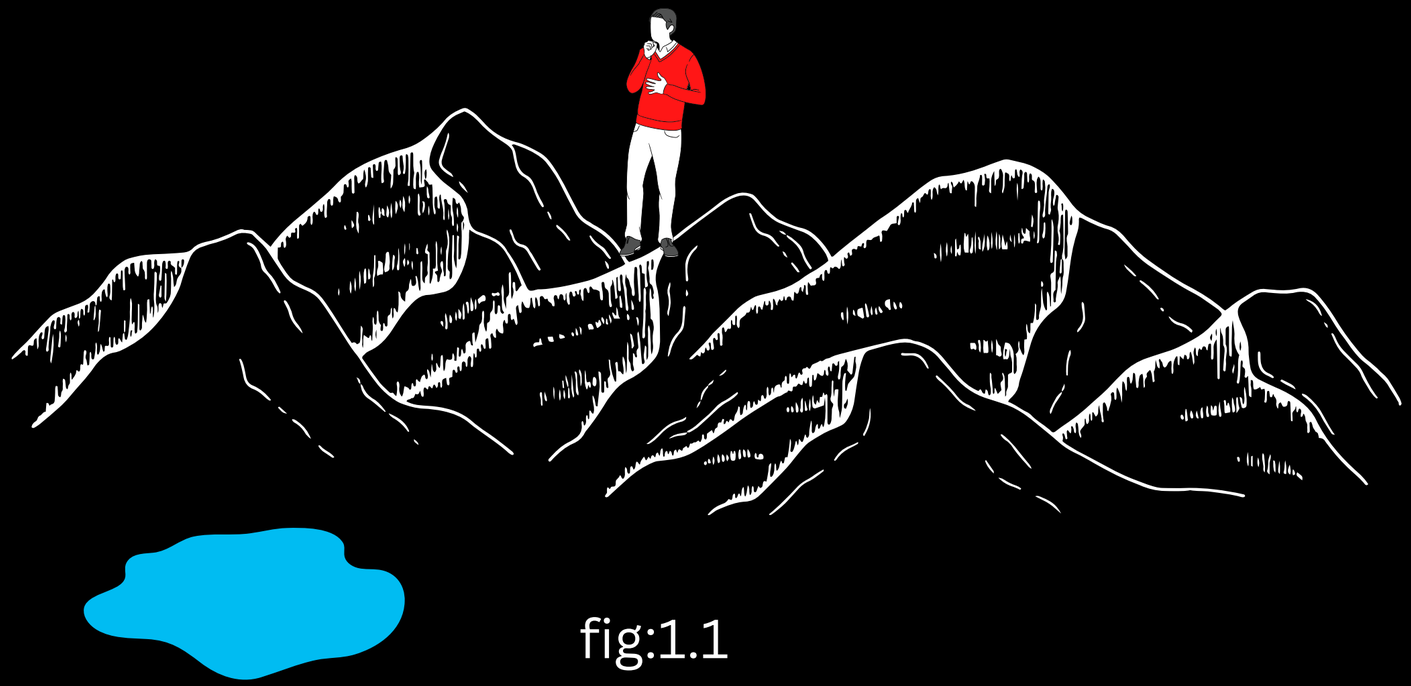

fig:1.1

Above figure let's assume you are at **top of the mountain** and you **eyes was tighed** by cloth your **aim is you need to reach down towards the water**. Generally what we do we try to **step any one side if the side is slope we'll go in that direction else we change**. like gradient also use **same concept to find the global minima gradient descent ultimate goal is finding the gloabal minima** here we called **step** but in **gradient descent algorithm we called as learining rate**.

NOTE: THIS EXAMPLE THAT EXPLAINED WHAT I UNDERSTOOD

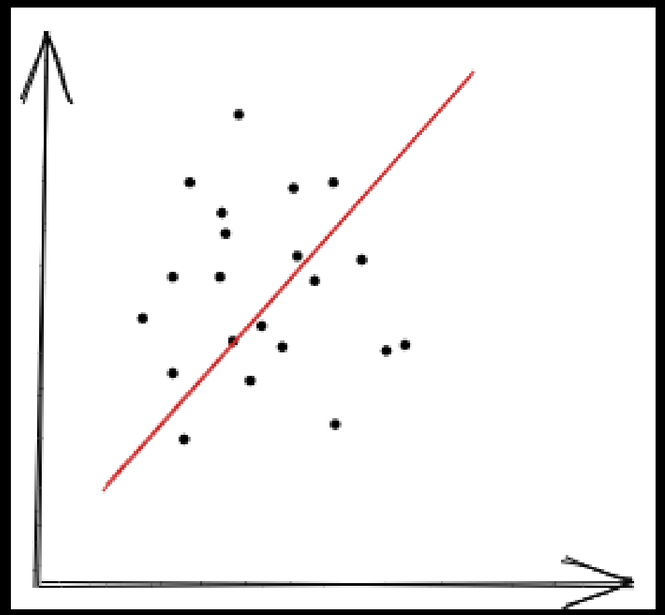

fig1.2

In fig1.2 here the fit line is simple to **find minimum cost using the mx+c** is the general equation of any straight line where m is the gradient of the line (how steep the line is) and c is the y -intercept (the point in which the line crosses the y -axis)

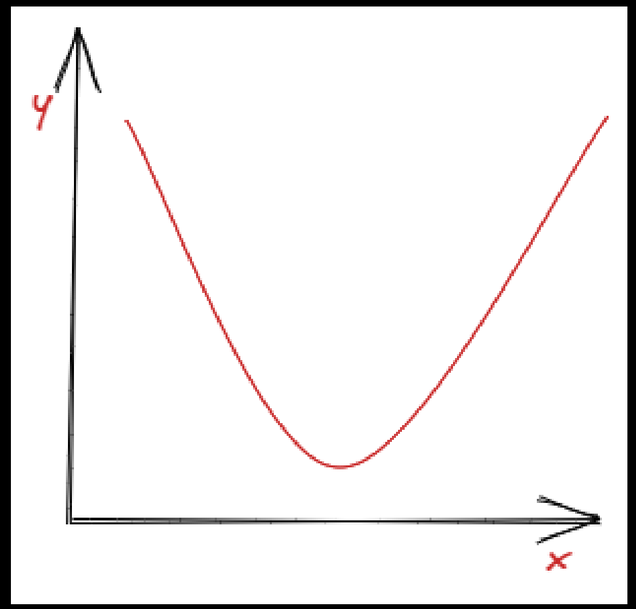

fig1.3

In fig1.2 if fit line is like this it is **very hard to find the minimum cost** to solve this we use **Gradient descent is an optimization algorithm** Training data helps these models learn over time, and the cost function within gradient descent specifically **acts as a barometer, gauging its accuracy with each iteration of parameter updates.** Until the function is **close to or equal to zero**, the model will continue to adjust its parameters to yield the smallest possible error.

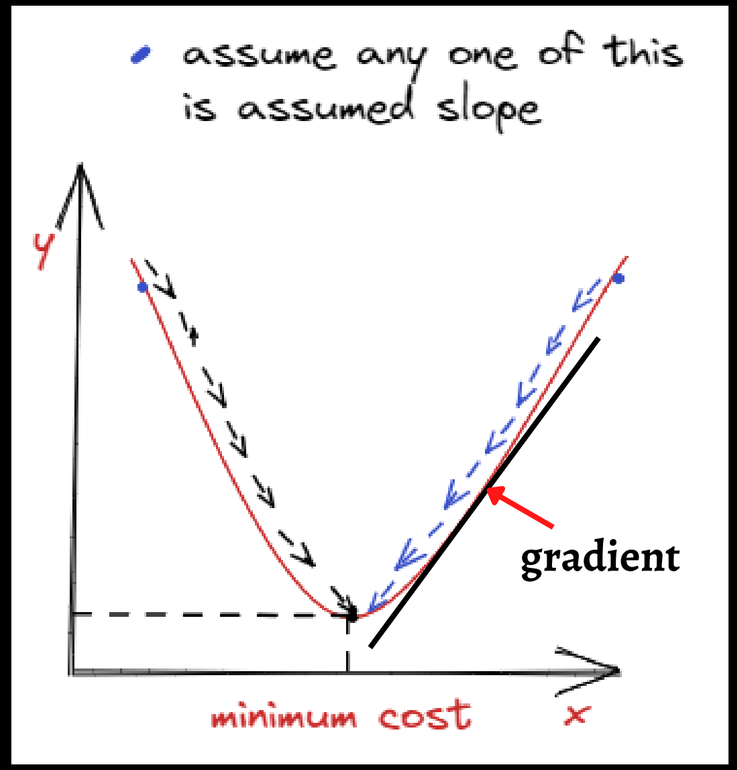

fig:1.4

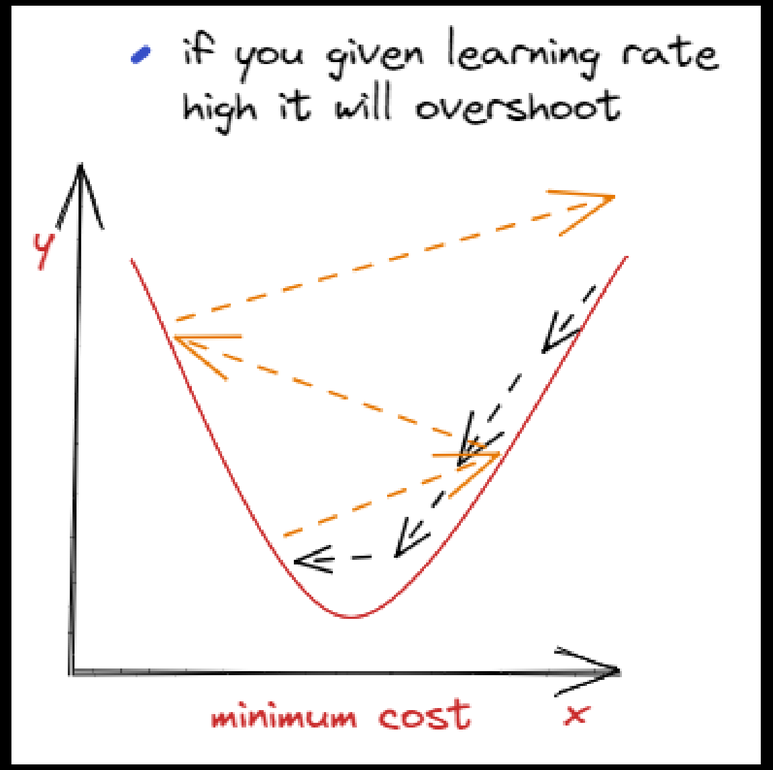

fig:1.5

In above figures you can find the **gradient line** first we **initial weight randomly** after that check for **slope** and taking the **small steps(learning rate)** towards the global minima, taking the large steps leads to over-shooting in fig:1.5 here **"m"** and **"c"** will **update** using **gradient descent** to find the best fit line best line nothing but minima or loss function.