OPTIMIZATION

INSAID

Course Overview

You are here...

Term	CDF	GCD	GCDAI	PGPDSAI
Term 1	Data Analytics with Python	Data Analytics with Python	Data Analytics with Python	Data Analytics with Python
Term 2	Data Visualization Techniques	Data Visualization Techniques	Data Visualization Techniques	Data Visualization Techniques
Term 3	EDA & Data Storytelling	EDA & Data Storytelling	EDA & Data Storytelling	EDA & Data Storytelling
		Minor Project	Minor Project	Minor Project
Term 4		Machine Learning Foundation	Machine Learning Foundation	Machine Learning Foundation
Term 5		Machine Learning Intermediate	Machine Learning Intermediate	Machine Learning Intermediate
Term 6		Machine Learning Advanced (Mandatory)	Machine Learning Advanced (Mandatory)	Machine Learning Advanced (Mandatory)
		Data Visualization with Tableau (Elective - I)	Data Visualization with Tableau (Elective - I)	Data Visualization with Tableau (Elective - I)
		Data Analytics with R (Elective - II)	Data Analytics with R (Elective - II)	Data Analytics with R (Elective - II)
		Capstone Project	Capstone Project	Capstone Project
Term 7		Bonus: Industrial ML (ML – 4 & 5)	Basics of AI, TensorFlow, and Keras	Basics of AI, TensorFlow, and Keras
Term 8			Deep Learning Foundation	Deep Learning Foundation
Term 9			NPL – I/CV – I	CV – I
Term 10			NLP – II/CV – II	NLP – I
			Capstone Project	Capstone Project
Term 11				CV – II
Term 12				NLP – II
				NLP – III + CV – III
				AutoVision & AutoNLP
				Building AI product

Term Context

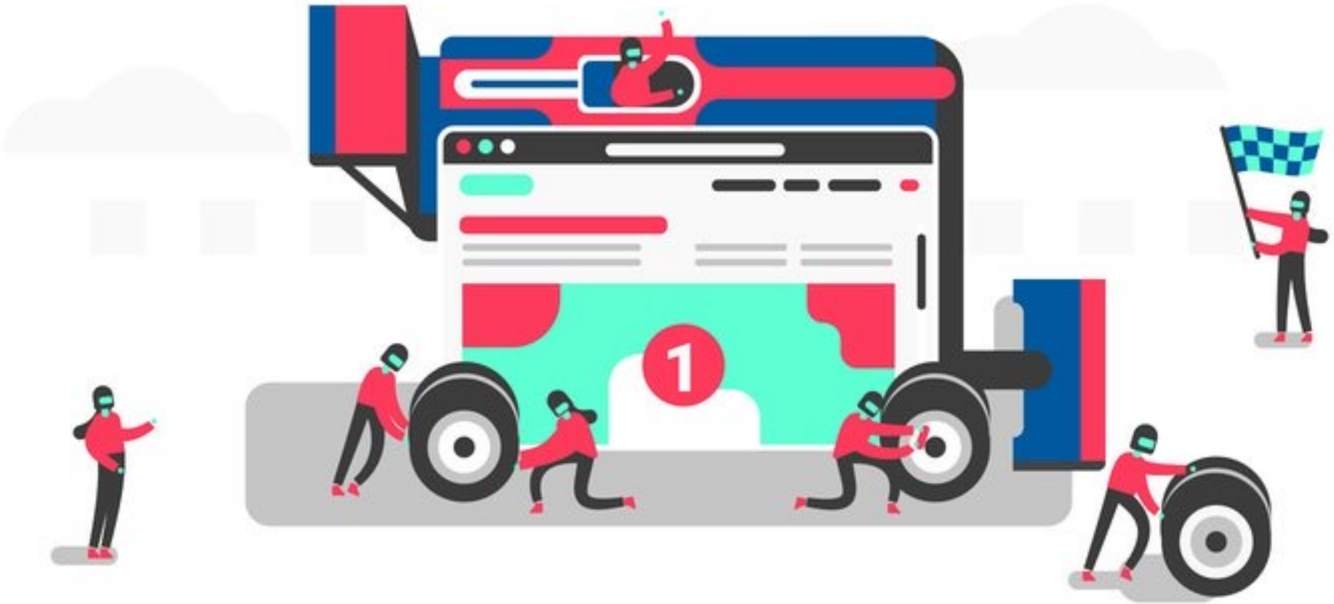
- K – Nearest Neighbor
- K-means Clustering
- Ensemble Learning
- **Optimization** You are here...

Agenda

1. **Optimization**
2. Optimization Techniques
3. Cost Function
4. Working of Gradient Descent
5. Issues with Gradient Descent
6. Types of Gradient Descent

Optimization

- The process of **choosing** the **optimal** solution from all the **feasible** solutions.

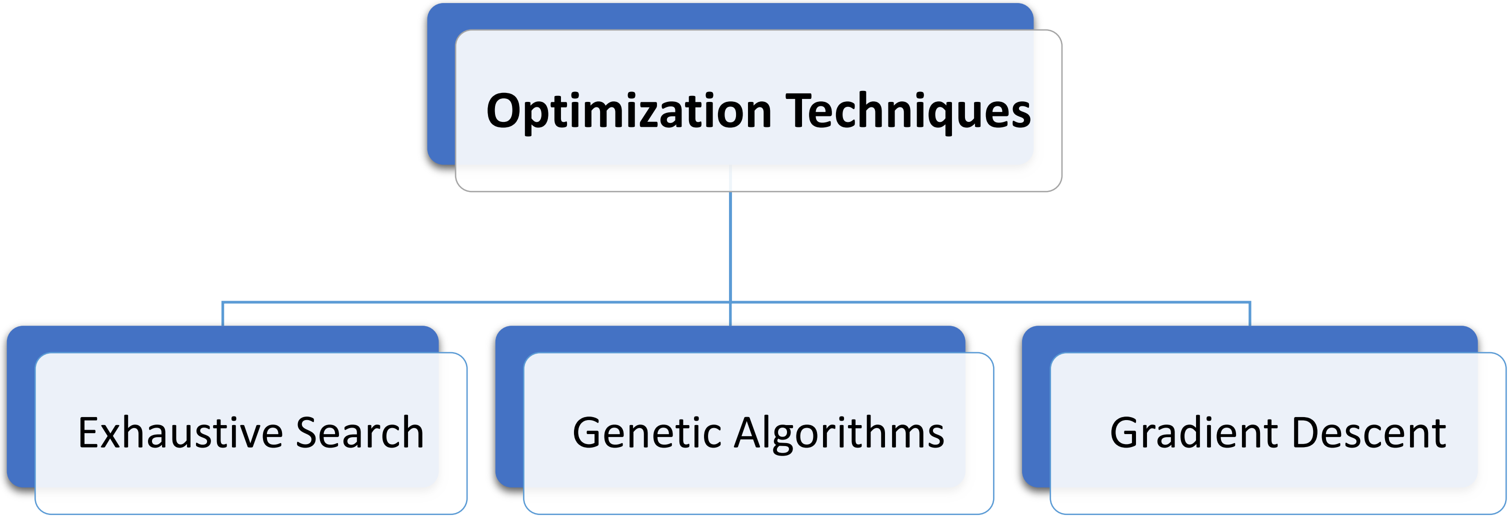


Need Of Optimization

- The **goal** is to create a model that gives **accurate predictions** in a particular set of cases in **less time**.

Agenda

Optimization Techniques

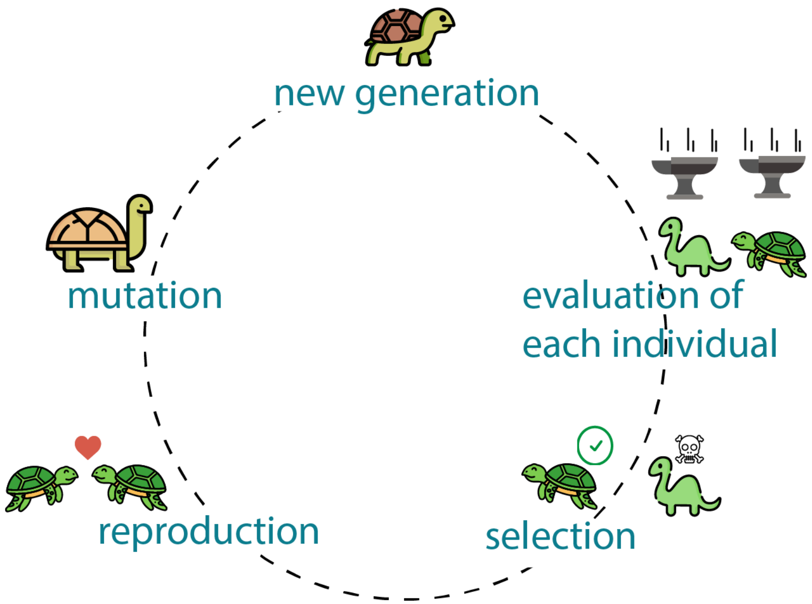


Exhaustive Search

- The process of looking for the **most optimal hyper parameters.**
- It simply checks whether each candidate is a good match or not.
- But if there are thousands of options to consider, it becomes unbearably **heavy and slow**.

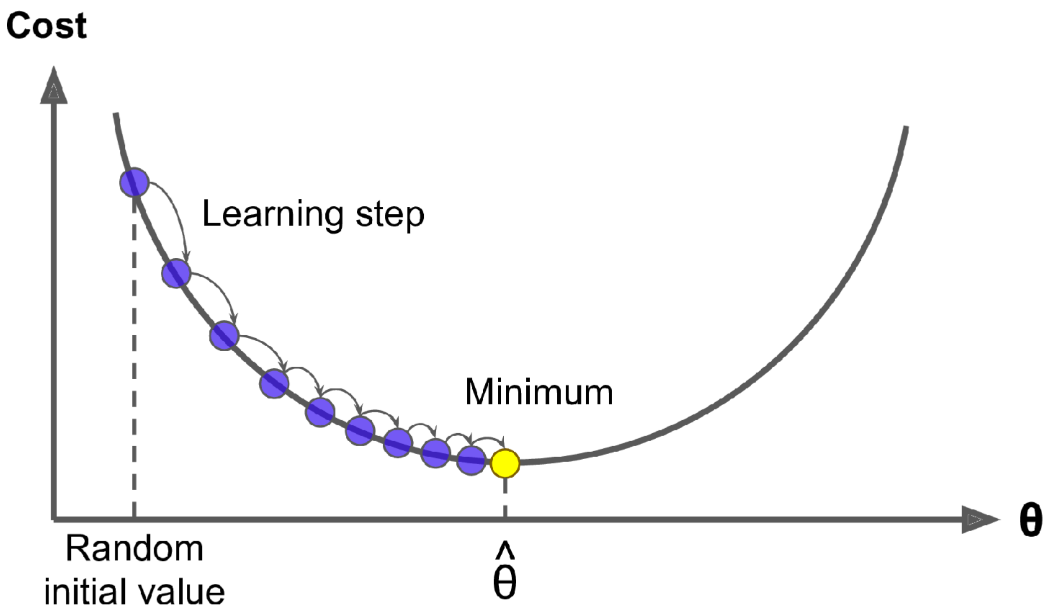
Genetic Algorithms

- A **search heuristic** that is inspired by Charles Darwin's theory of **natural evolution** (a process of natural selection).
- The **fittest** individuals are **selected** for reproduction in order to **produce** offspring of the next generation.
- It is an attempt to apply **theory of natural evolution** to the machine learning.



Gradient Descent

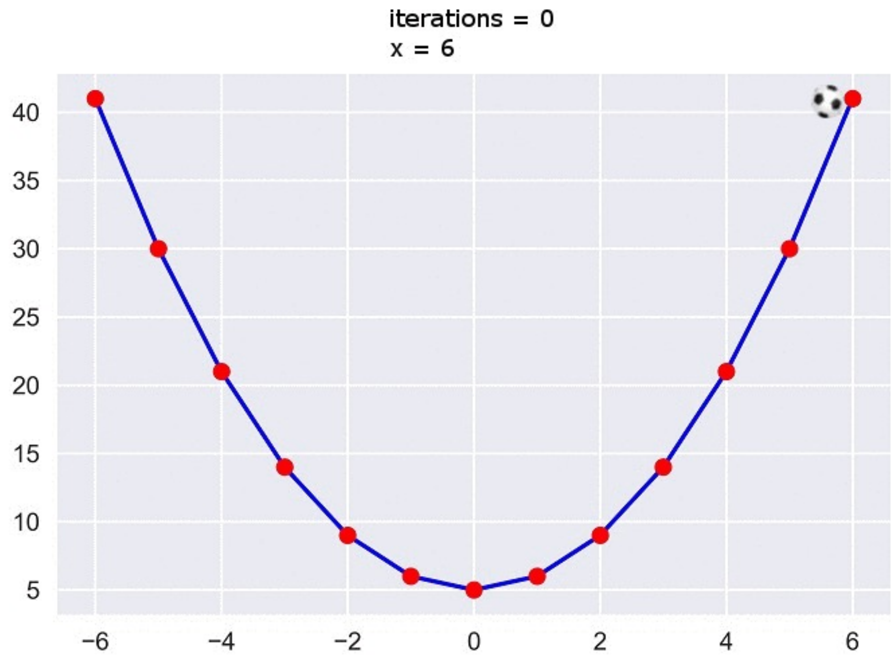
- The most common and **widely used** algorithm to **optimize** the **model** by **minimizing** the **error/cost**.
- It **iterates** over the training dataset while **re-adjusting** the model's **parameters**.



INSAID

Gradient Descent

- It tweak parameters **iteratively** to **minimize** a **cost** function.
- Two things matters **-** **direction** and **step-size.**



Agenda

1. Optimization
2. Optimization Techniques
3. **Cost Function**
4. Working of Gradient Descent
5. Issues with Gradient Descent
6. Types of Gradient Descent

Cost Function

- It is the **average** of the **loss function** for all the training examples.
- There are **several cost functions** that are used to evaluate models.
- **For example**: Mean Squared Error, Mean Absolute Error, etc.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(Y' - Y)^2 \qquad \mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|Y' - Y|$$

- Y’ = Predicted values
- Y = Actual values
- N = No. of data points

Agenda

1. Optimization
2. Optimization Techniques
3. Cost Function
4. **Working of Gradient Descent**
5. Issues with Gradient Descent
6. Types of Gradient Descent

Working of Gradient Descent

- You start by **filling gradient** (θ) with random values, also called **random initialization.**
- Let's say, $h_\theta(x)$ is hypothesis function,

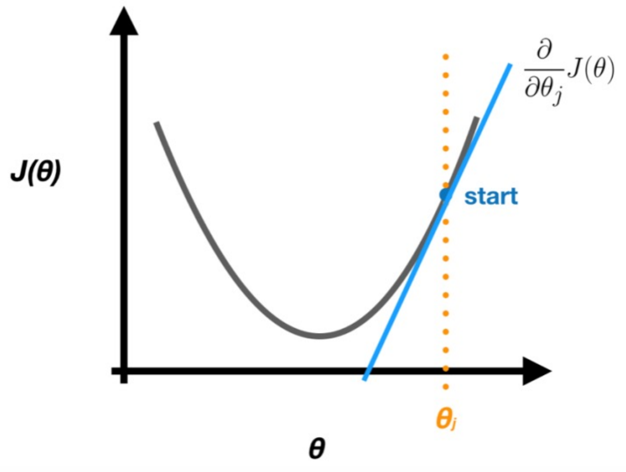
$$h_\theta(x) = \theta_0 + \theta_1 x$$

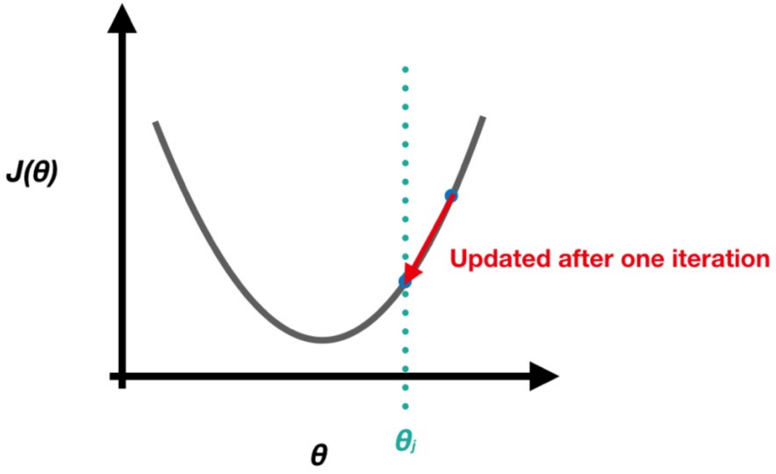
- θ_0 = bias
- θ_1 = weight
- x = independent variable/feature

- Hence, we will initialize $\boldsymbol{\theta_0}$, $\boldsymbol{\theta_1}$ with some **random** values.

Working of Gradient Descent

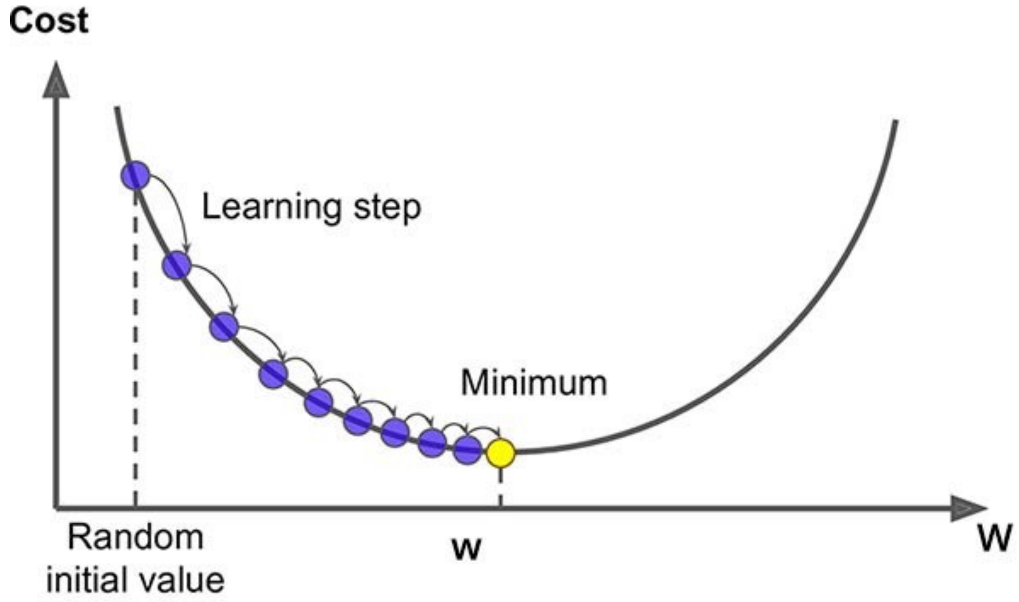
- Each step attempts to **decrease** the **cost** function until the algorithm **converges** to a **minimum.**
- Most common values of learning rate (α) are : 0.001, 0.003, 0.01, 0.03, 0.1, 0.3.





Working of Gradient Descent

- Once the **gradient** is **zero**, you have reached a **minimum**!

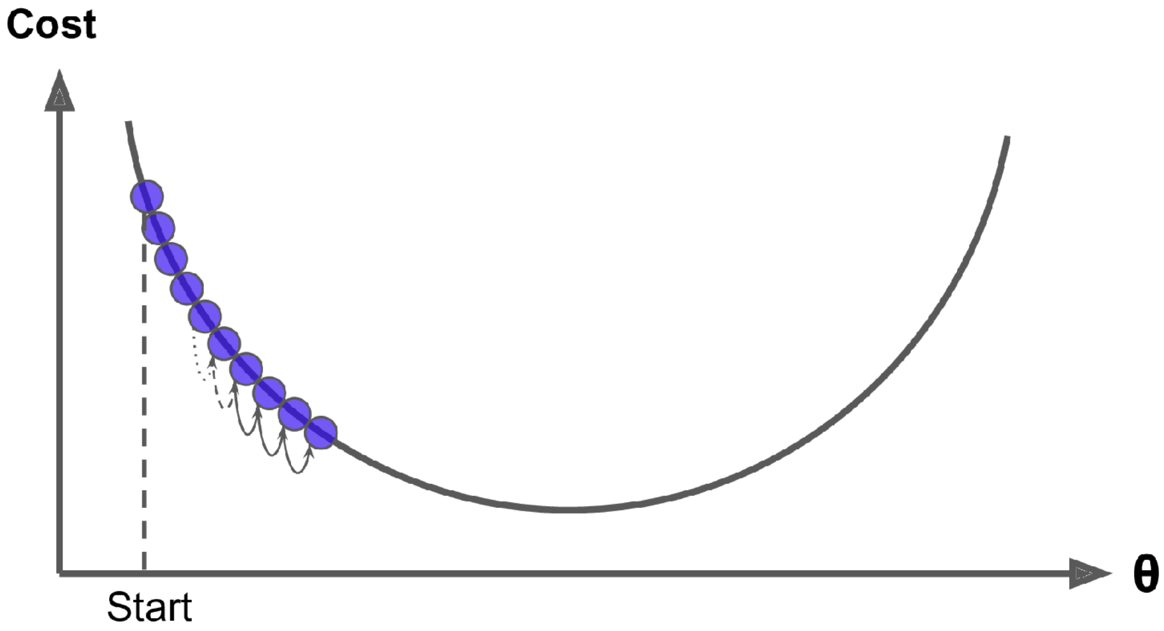


Agenda

1. Optimization
2. Optimization Techniques
3. Cost Function
4. Working of Gradient Descent
5. **Issues with Gradient Descent**
6. Types of Gradient Descent

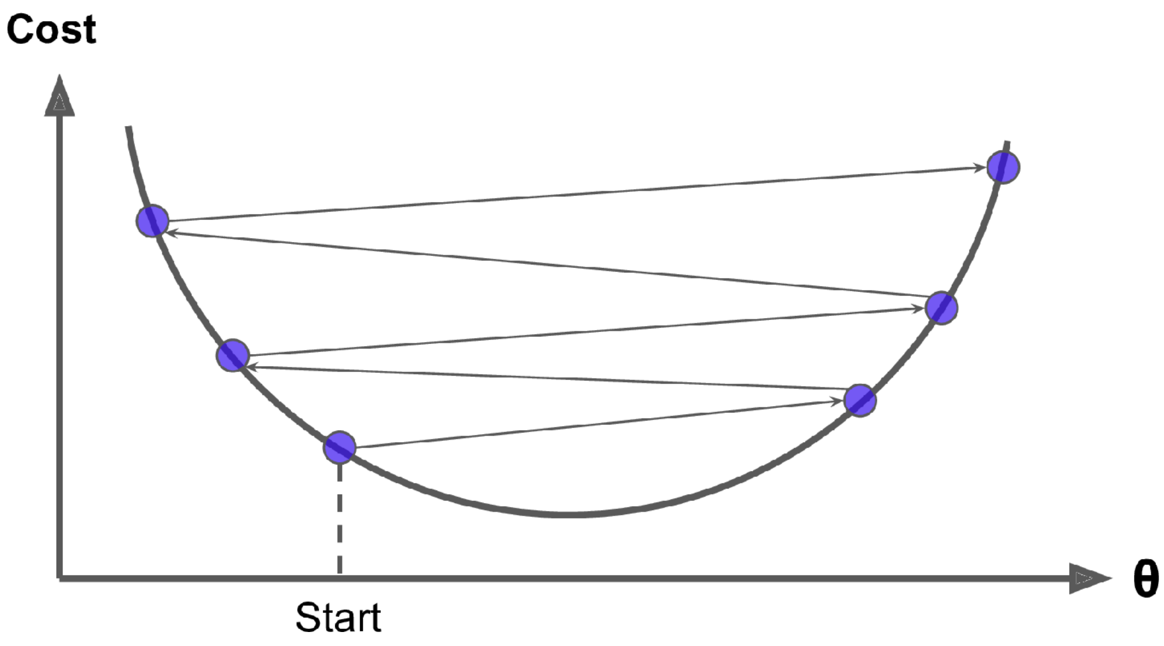
Issues with Gradient Descent

- If the learning rate is **too small**, then it will **iterate too many times** to finally converge, which will take a long time.



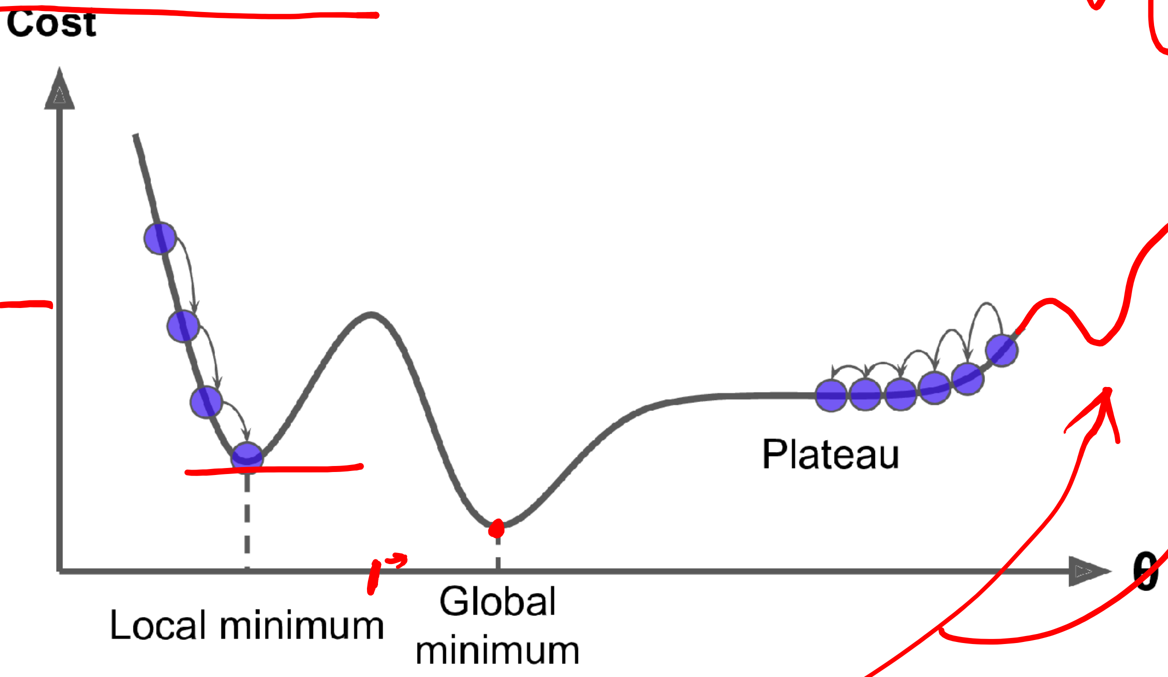
Issues with Gradient Descent

- If the learning rate is **too high**, then we may **overshoot the minima** and possibly keep bouncing.



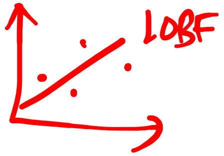
Issues with Gradient Descent

- Not all cost functions look like nice, regular bowls.
- There may be holes, ridges, plateaus, and all sorts of irregular terrains.
- This makes **convergence** to the **minimum difficult**.



Agenda

Types of Gradient Descent

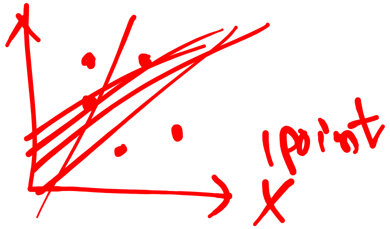


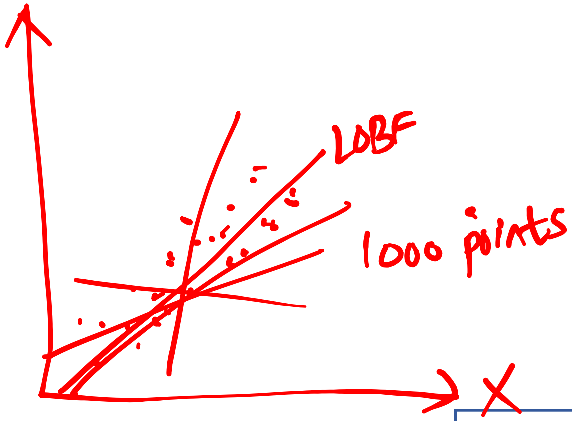
$$\frac{(A_1-P_1)+(A_2-P_2)+(A_3-P_3)+(A_4-P_4)}{4}$$

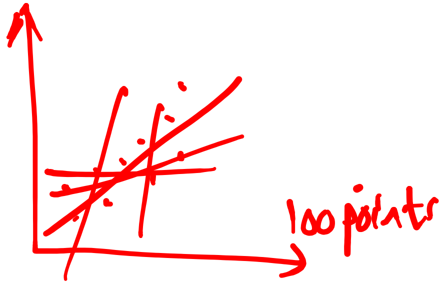
Total no of computations = 4.

500 Lines X 1

= 500







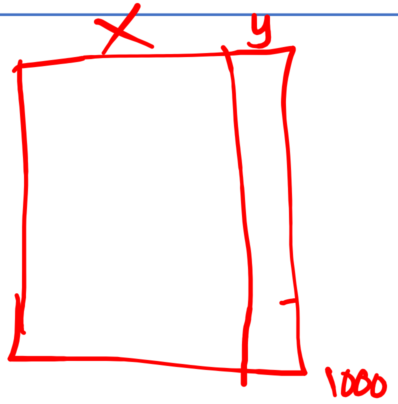
Gradient Descent Types

1. Batch Gradient Descent

500 Lines X 1000

= 5,00,000

i.e. 5 Lakh



3. Stochastic Gradient Descent

∴ Batch size = 1

We will take any 1 point & plot LOBF for that

Keep repeating this process.

2. Mini-Batch Gradient Descent

500 Lines X 100

= 50,000

i.e. 50 Thousand

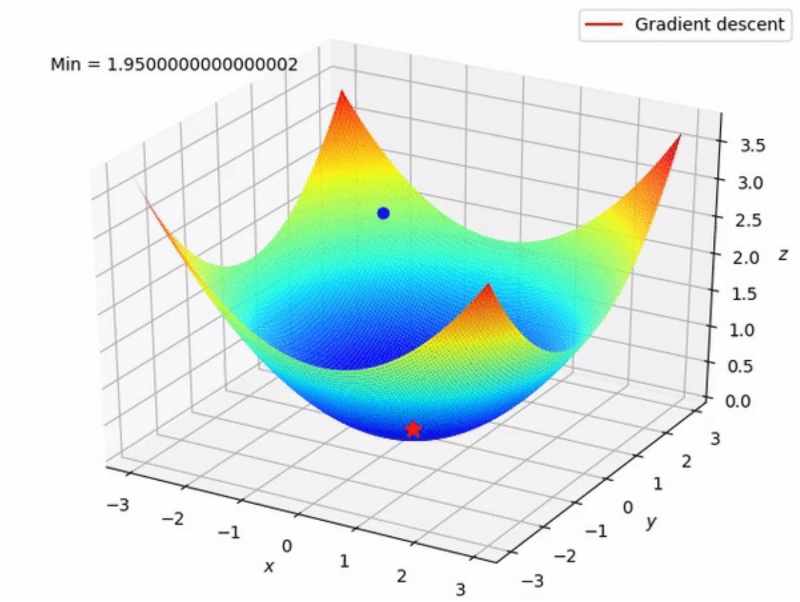
Batch Size = 100

∴ No of Batches $= \frac{1000}{100} = 10$

Take any one of the batch instead of all 1000 points

Batch Gradient Descent

- Here calculations are **involved** over the **full training set** i.e. at each gradient descent step.
- We take the **average** of the **gradients** of all the training examples.
- Then use **mean** gradient to update our parameters.



Pros/Cons of Batch Gradient Descent

Pros

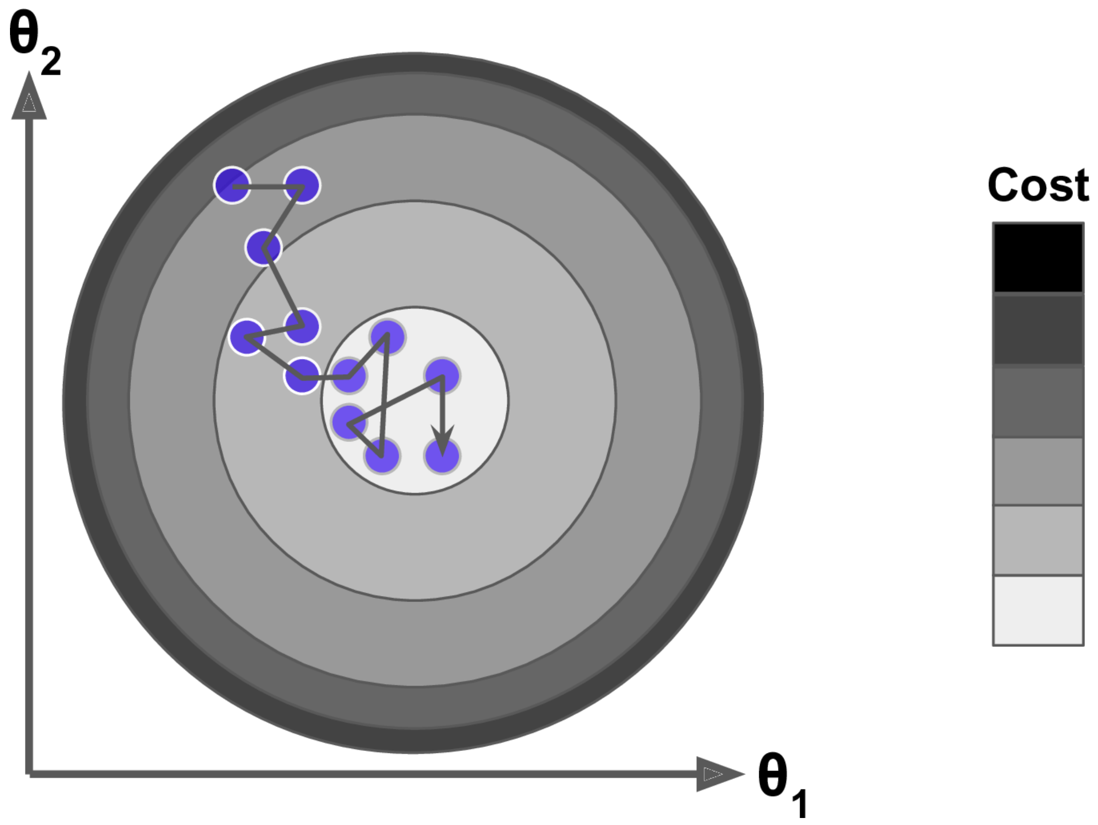
- It is **computationally efficient**.
- **No updates** are required after each sample.
- It produces a **stable** gradient descent **convergence**.
- It benefits from the **vectorization**, which increases the speed of processing.

Cons

- It's learning process is very **slow**.
- The entire training set can be **too large** to **process** in the memory.
- We may get **stuck** in a **local minimum** of the loss function and never reach the global optimum.

Stochastic Gradient Descent

- This variant **picks** a **random instance** in the training set at **every step**.
- **Computes** the **gradient** based only on a **single instance**.



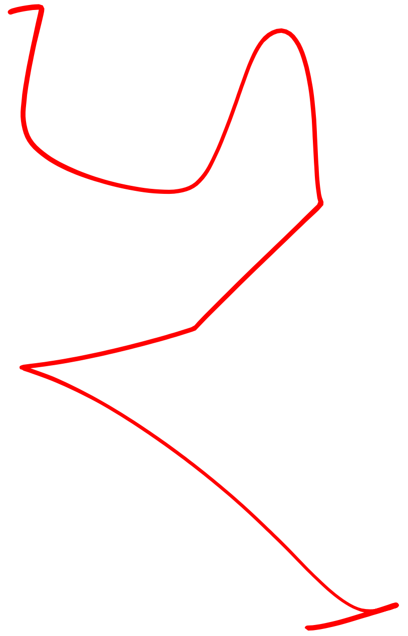
Pros/Cons of Stochastic Gradient Descent

Pros

- It can **easily fit** the data into **memory**.
- It is **faster** on a large dataset and better than Batch Gradient Descent.
- It **immediately** gives us an **insight** into the performance of the model.

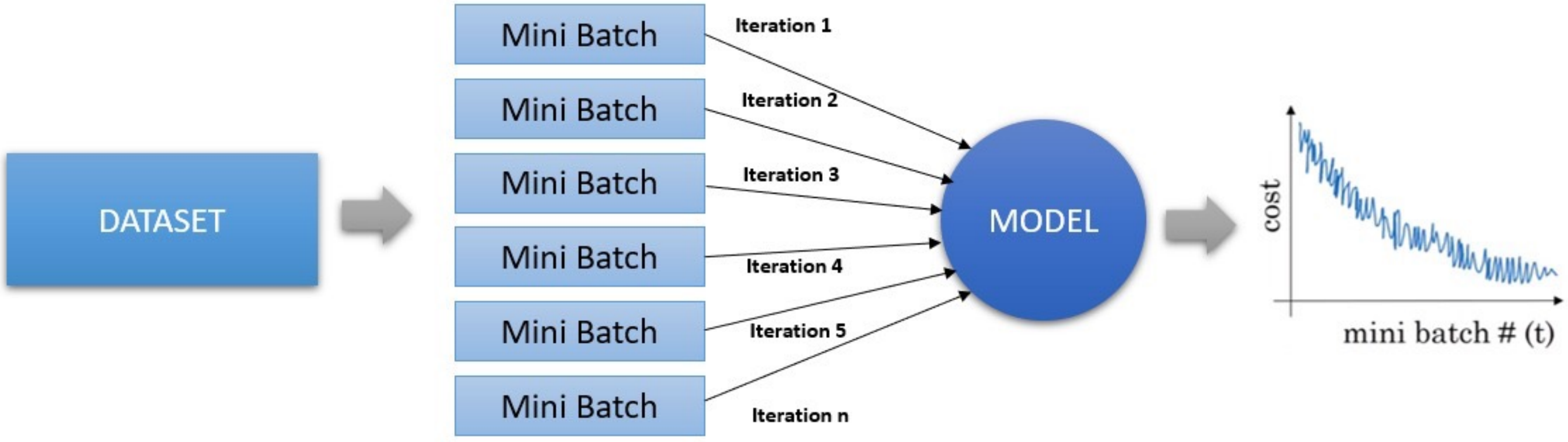
Cons

- More **computationally intensive** than the batch gradient descent
- **Lose** the benefits of **vectorization** since we process one observation per time
- Due to the **noisiness**, it is more **difficult** to find and stay at a **global minimum**.



Mini-Batch Gradient Descent

- Here gradients are computed on **small random sets** of instances called **mini-batches**.
- Finds a **balance** between the robustness of **stochastic gradient descent** and the efficiency of **batch gradient descent.**



Pros/Cons of Mini Batch Gradient Descent

Pros

- It is **computationally efficient**.
- It is a **fast learner** since we perform more updates.
- The has a more **stable convergence** than Stochastic Gradient Descent.

Cons

- It is more **time consuming**.
- It requires **configuring** of **mini-batch size** as hyperparameter.

Thank you