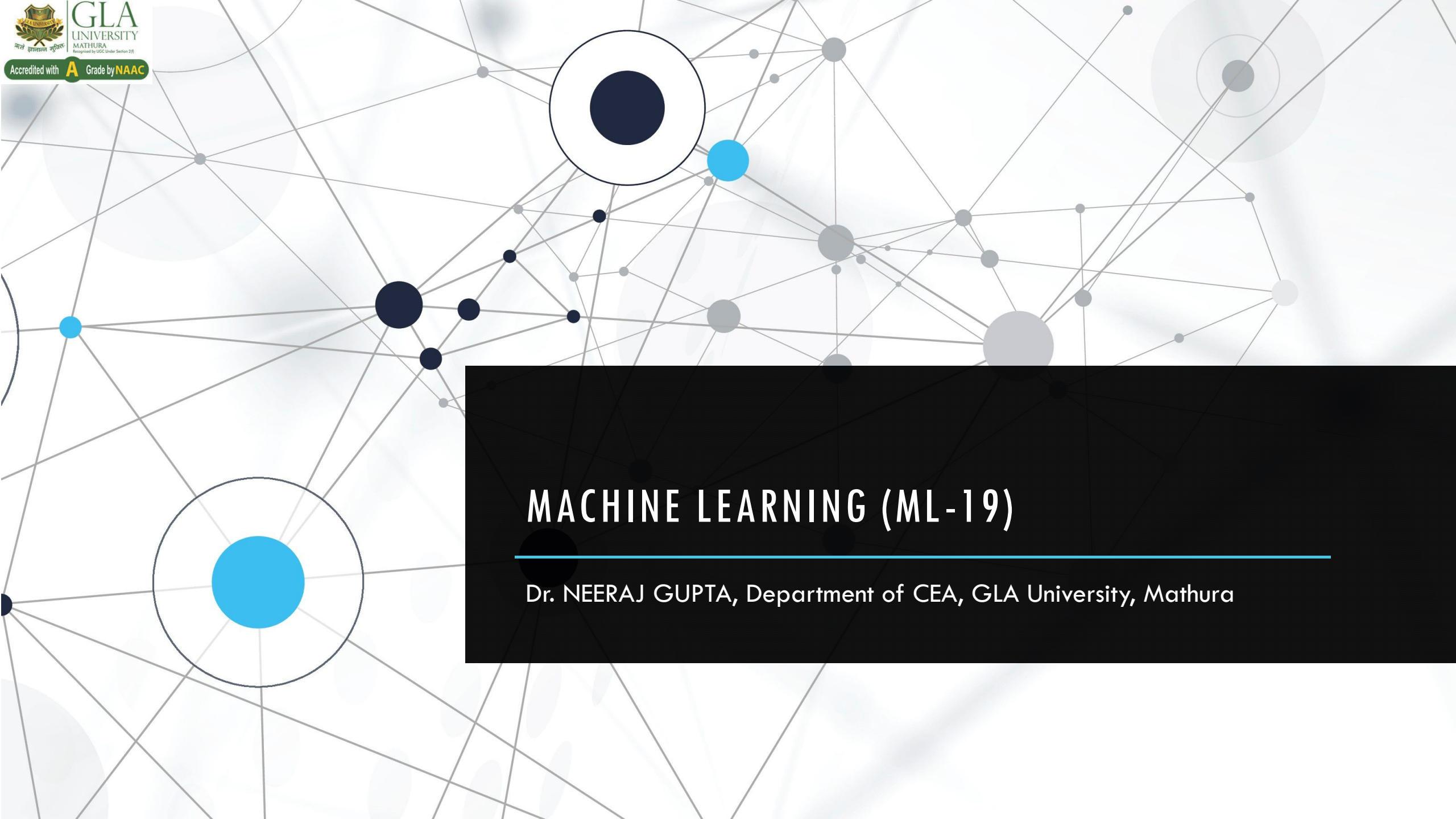


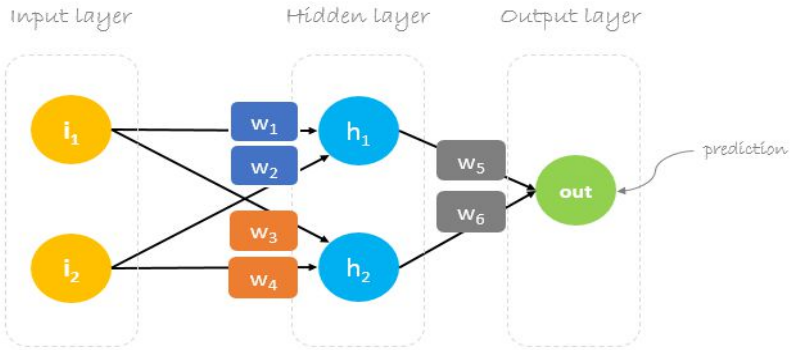
MACHINE LEARNING (ML-19)

Dr. NEERAJ GUPTA, Department of CEA, GLA University, Mathura

AGENDA

- Artificial Neural Network
  - Gradient Descent
  - Stochastic Gradient Descent
  - Gradient Descent Vs Stochastic Gradient Descent

OPTIMIZATION

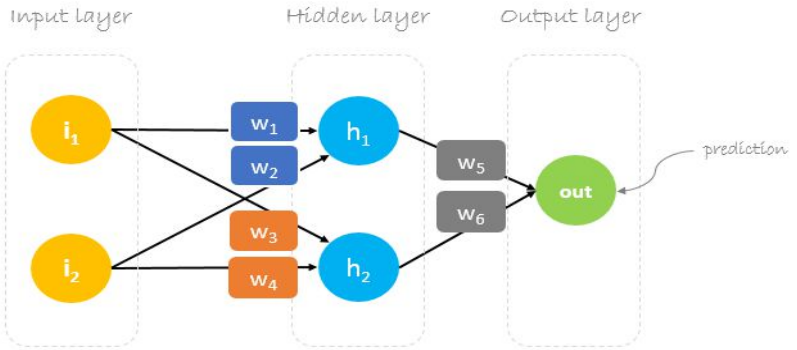


- Optimization is always the ultimate goal whether you are dealing with a real life problem or building a software product.
- Optimization basically means getting the optimal output for your problem.

Broad applications of Optimization

- There are various kinds of optimization techniques which are applied across various domains such as
- **Mechanics –** For eg: In deciding the surface of aerospace design
- **Economics –** For eg: Cost minimization
- **Physics –** For eg: Optimization time in quantum computing
- Optimization has many more advanced applications like deciding optimal route for transportation, shelf-space optimization, etc.

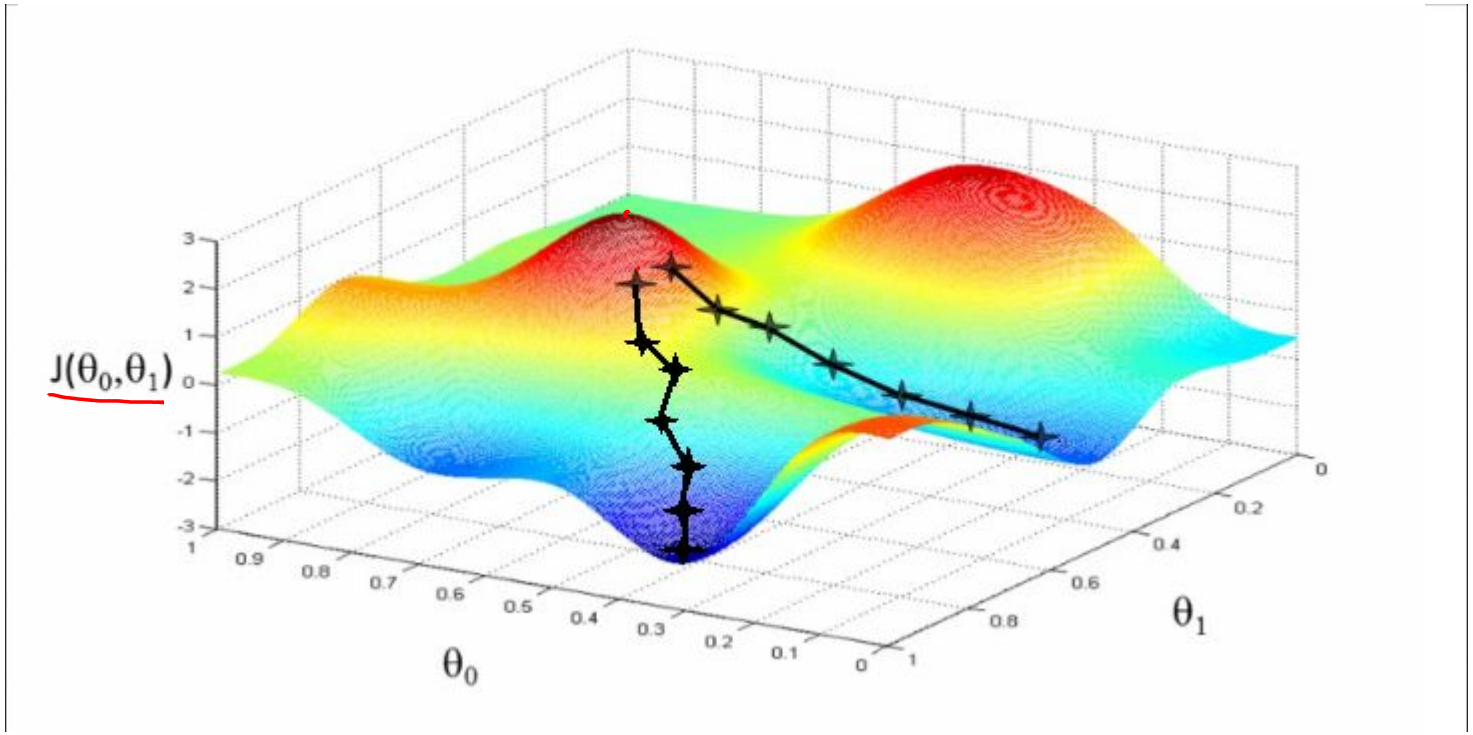
OPTIMIZATION



- Many popular machine algorithms depend upon optimization techniques such as
- linear regression,
- k-nearest neighbors,
- neural networks, etc.
- The applications of optimization are limitless and is a widely researched topic in both academia and industries.

WHAT IS GRADIENT DESCENT?

To explain Gradient Descent I'll use the classic mountaineering example.

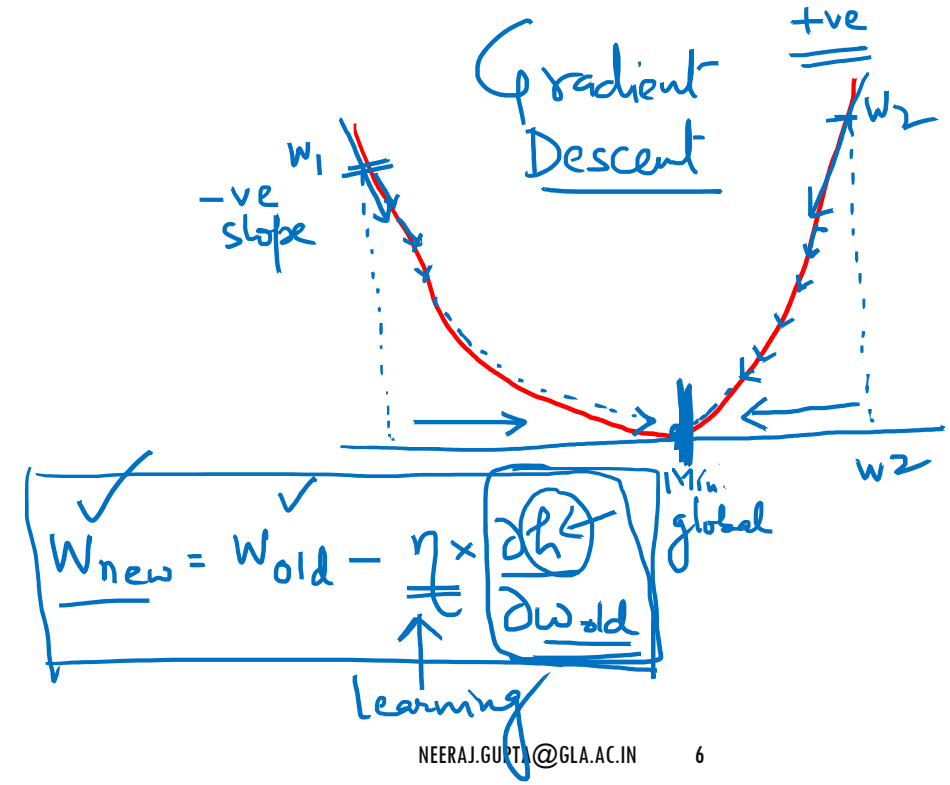


GRADIENT DESCENT ALGORITHM AND ITS VARIANTS

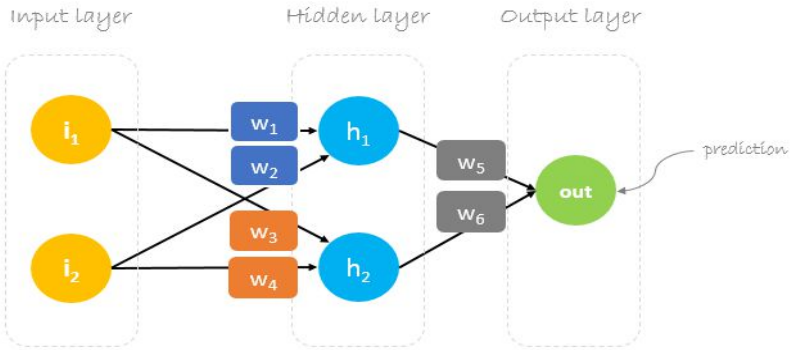
- Gradient Descent is an optimization algorithm used for minimizing the cost function in various machine learning algorithms. It is basically used for updating the parameters of the learning model.

Types of gradient Descent:

1. Batch Gradient Descent
2. Stochastic Gradient Descent
3. Mini Batch gradient descent



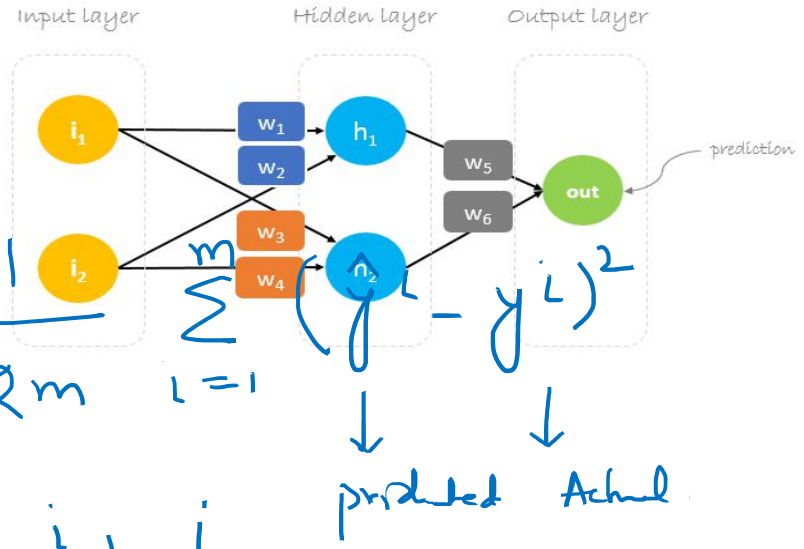
BATCH GRADIENT DESCENT



Batch Gradient Descent:

- This is a type of gradient descent which processes all the training examples for each iteration of gradient descent.
- If the number of training examples is large, then batch gradient descent is computationally very expensive.
- If the number of training examples is large, then batch gradient descent is not preferred.
- Instead, we prefer to use stochastic gradient descent or mini-batch gradient descent.

BATCH GRADIENT DESCENT



Linear Regression Cost function $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}^i - y^i)^2$

($\hat{y}^i$: predicted, y^i: Actual)

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{M} \sum_{i=1}^{m} (\hat{y}^i - y^i) x_j^i$$

Batch G.D (Vanilla)

$$\theta_j^{new} := \theta_j^{old} - \alpha \left[\frac{\partial}{\partial \theta_j} (J(\theta)) \right]_{Batch}$$

$$\frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) x_j^i$$

STOCHASTIC GRADIENT DESCENT

Stochastic Gradient Descent:

- This is a type of gradient descent which processes 1 training example per iteration.
- Hence, the parameters are being updated even after one iteration in which only a single example has been processed.
- Hence this is quite faster than batch gradient descent.
- But again, when the number of training examples is large, even then it processes only one example which can be additional overhead for the system as the number of iterations will be quite large.

STOCHASTIC GRADIENT DESCENT

for i in range (m):

$\theta_j = \theta_j - \alpha \cdot$ Only One Example

$(\hat{y}^i - y^i) \cdot x_j^i$

SGD

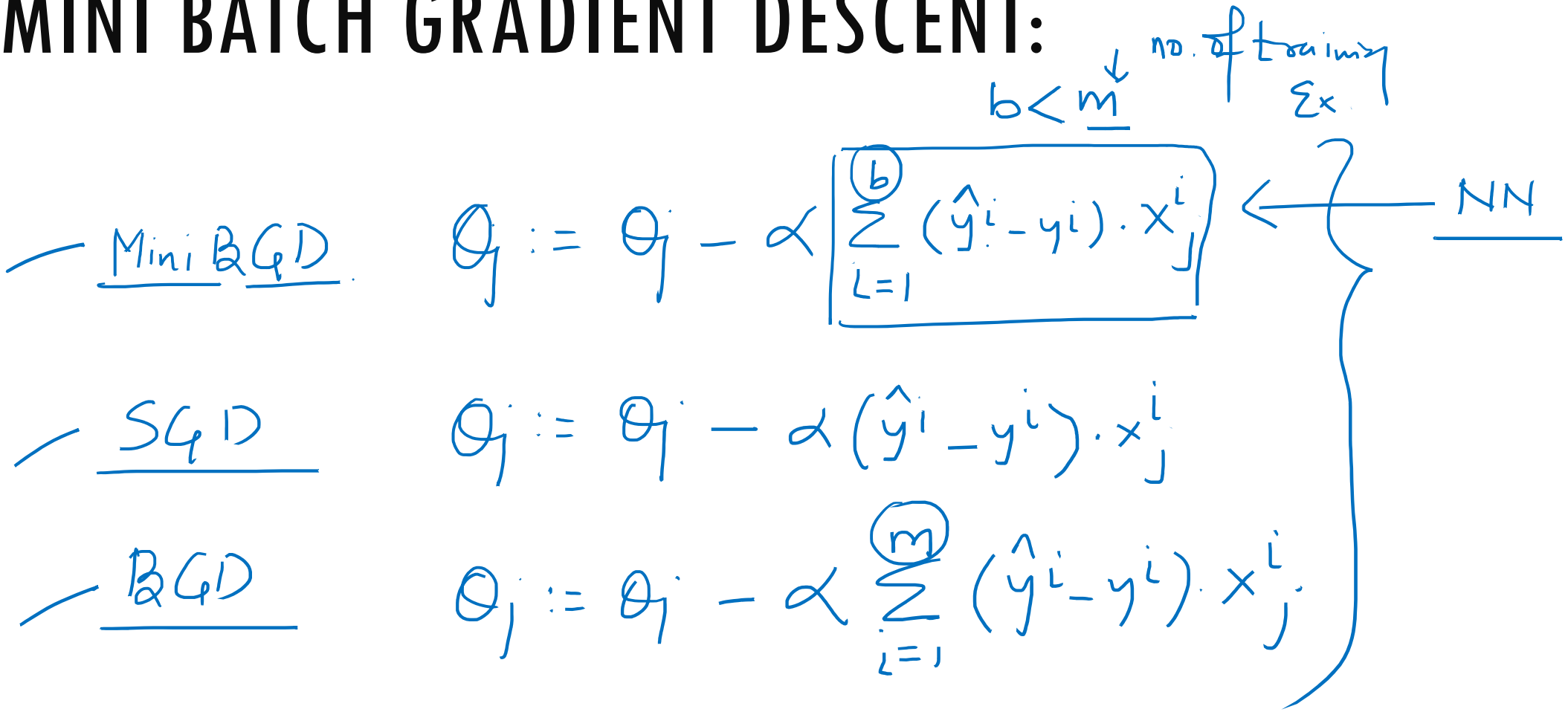
MINI BATCH GRADIENT DESCENT:

$k < m$

Mini Batch gradient descent:

- This is a type of gradient descent which works faster than both batch gradient descent and stochastic gradient descent.
- Here *b* examples where $b<m$ are processed per iteration.
- So even if the number of training examples is large, it is processed in batches of b training examples in one go.
- Thus, it works for larger training examples and that too with lesser number of iterations.

MINI BATCH GRADIENT DESCENT:



THANKS


Keep Learning
Keep Growing


Dr. Neeraj Gupta
Assistant Professor, Dept. of CEA
neeraj.gupta@gla.ac.in