Lecture 1.2

Closed-form Equation, Type of Gradient Descent

(Batch, Stochastic, Mini-batch) - Definition, properties.

Dr. Mainak Biswas

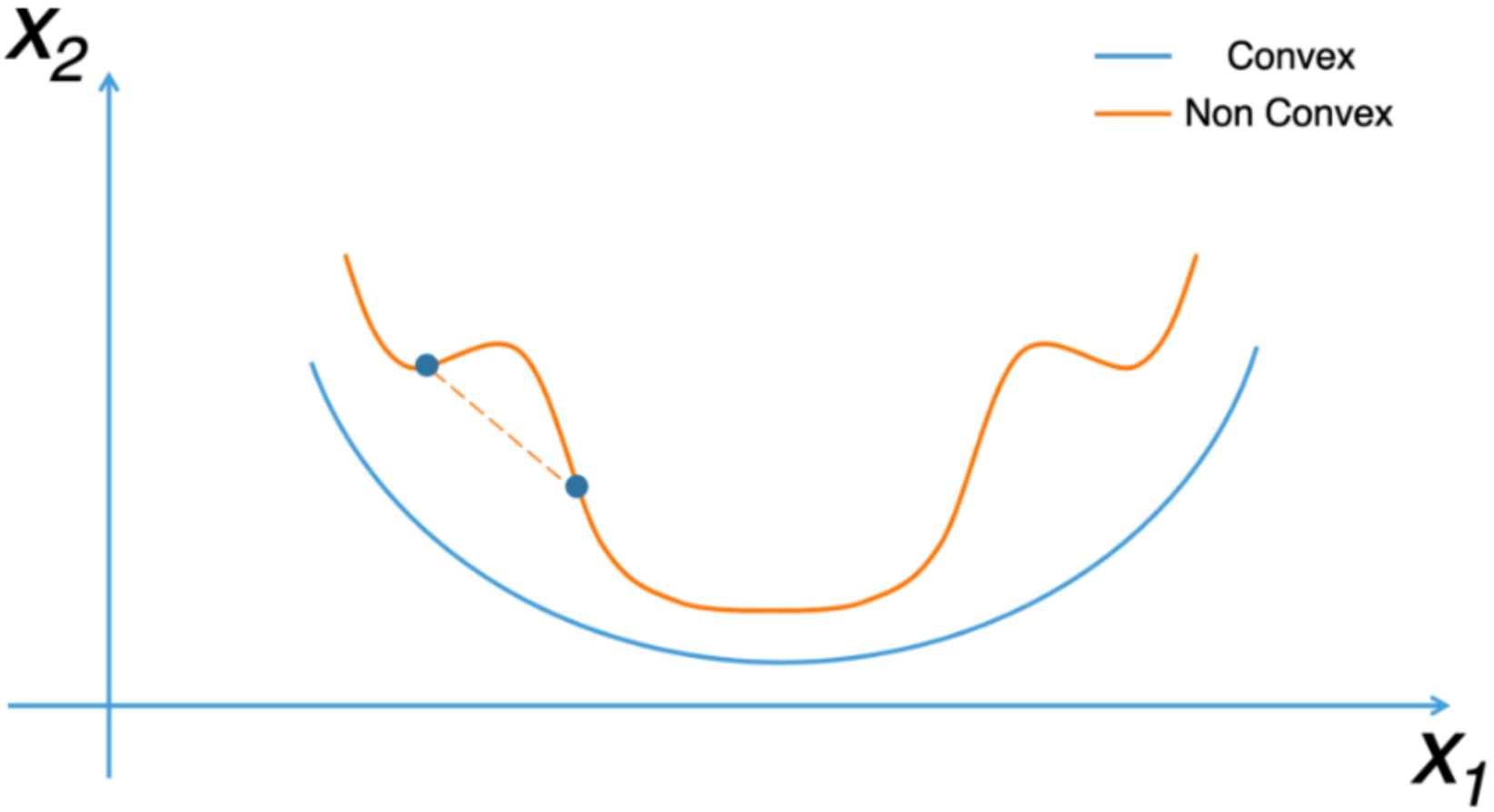
Closed-form Equation

- Closed-form Equation: closed-form equation is a mathematical expression that provides a direct way to compute a value without requiring iterative procedures or infinite series
  - Example:
    - **Sum of an Arithmetic Series**: The sum of the first n terms of an arithmetic series with the first term **a** and common difference d is:

$$S_n = \frac{n}{2}(2a + (n-1)d$$

Gradient Descent

- Gradient Descent is an optimization algorithm used in machine learning and deep learning to minimize the loss function by updating the model's parameters in the direction of the steepest descent
- The type of gradient descent depends on how much data is used to compute the gradient at each iteration
- Gradient descent is also called “the deepest downward slope algorithm”
- It is very important in machine learning, where it is used to minimize a cost function


x_2
Convex
Non Convex
x_1

Loss function

$$E(w) = \frac{1}{2N}\sum_{i=1}^{N}(f(x_i) - y_i)^2$$

- Where $f(x_i) = w^T x_i$, then

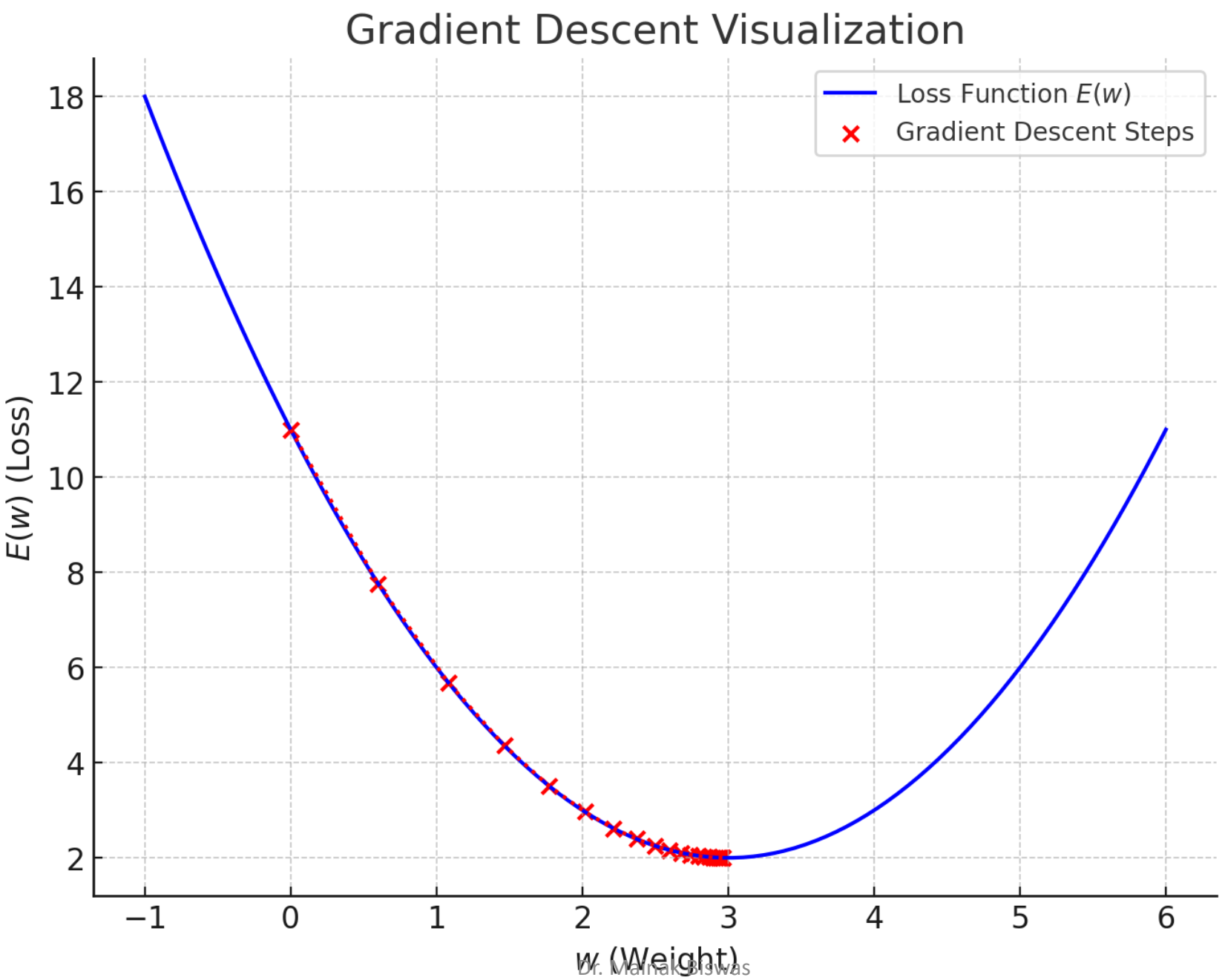
$$\frac{\partial E}{\partial w} = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)x_i$$

Mathematical Formulation of Gradient Descent

$$w = w - \eta \nabla E(w)$$

- w : Model parameters (weights)
- η : Learning rate
- $\nabla E(w)$: Gradient of the loss function $E(w)$ with respect to w
- It can be also written as:

$$w = w - \eta \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i) x_i$$


Gradient Descent Visualization
Loss Function $E(w)$
Gradient Descent Steps
$E(w)$ (Loss)
w (Weight)
18
16
14
12
10
8
6
4
2
−1
0
1
2
3
4
5
6
Dr. Manab Biswas

Numerical Problem

- Let $E(w) = (w-3)^2 + 2, \eta = 0.1, w = 0$, then find w and E(w) for five iterations:
  - So, we see $x_i = 1$, therefore iterations can be solved in terms of w only
  - $\frac{\delta E}{\delta w} = 2(w-3)$
  - $w_{new} = w_{old} - \eta \nabla E(w_{old}) \Rightarrow 0.8 w_{old} + 0.6$

Sl	w	$E(w)$
1	0	11
2	0.6	7.76

Sl	w	$E(w)$
1	0	11
2	0.6	7.76
2	1.0800	5.6864
2	1.4640	4.3593
2	1.7712	3.5099

Batch Gradient Descent

- **Batch Gradient Descent** is an optimization algorithm used to minimize a loss function by iteratively updating the model's parameters using the entire dataset to calculate the gradient
- Advantages:
  - Computes the gradient with high precision using the entire dataset
  - Converges steadily towards the minimum
  - Suitable for smooth and convex loss functions
- Disadvantages:
  - Memory-intensive when the dataset is large
  - Requires processing the entire dataset for each iteration

Stochastic Gradient Descent

- **Stochastic Gradient Descent (SGD)** is a variant of gradient descent where the model parameters are updated using only a single training example at a time, rather than the entire dataset
- This leads to faster updates and can help the algorithm escape local minima, making it suitable for large datasets
- Advantages
  - Faster Updates
  - Escaping Local Minima
  - Scalability
- Disadvantages
  - Noisy Convergence
  - Requires More Iterations

Mini-Batch Gradient Descent

- **Mini-batch Gradient Descent** is a hybrid approach between Batch Gradient Descent and Stochastic Gradient Descent. It aims to combine the advantages of both by updating the model parameters using a subset (mini-batch) of the training data rather than the entire dataset (batch) or just one data point (stochastic)
  - **Mini-batch**: The dataset is divided into small batches, each containing a fixed number of training examples (The size of each mini-batch (denoted as b) is a hyper-parameter)
  - **Gradient Calculation:** For each mini-batch, the gradient is calculated based on the average of the training examples in that batch
  - **Weight Update**: The model parameters are updated using the computed gradient for the mini-batch
  - **Repeat** for all mini-batches until convergence
- **Advantages**: Faster than Batch GD, Less Noisy than SGD
- **Disadvantages**: Choosing the Right Batch Size, Memory Considerations