

Gradient Descent and Its Variations

Presented by:

- (1) Chiragee Joshi
- (2) Soham Patel
- (3) Jaya Sai Siva Krishna Payyavula

Instructor Name: Dr. Yasser Alginahi

Date: Sept 29, 2023



University of Windsor

Gradient Descent & Cost Function

What is Gradient Descent ?

Gradient Descent is a fundamental optimization algorithm used in machine learning. The general idea of Gradient Descent is to tweak the parameters iteratively to minimize the cost function.

What is a Cost Function?

A cost function is a correctional function used to measure just how wrong the model is in finding the relation between input and output and how well a model fits the data.

[https://youtu.be/nxijs7_VlGA?si=xqIGeFVYVvOuXSWE\[1\]](https://youtu.be/nxijs7_VlGA?si=xqIGeFVYVvOuXSWE[1])



Gradient Descent

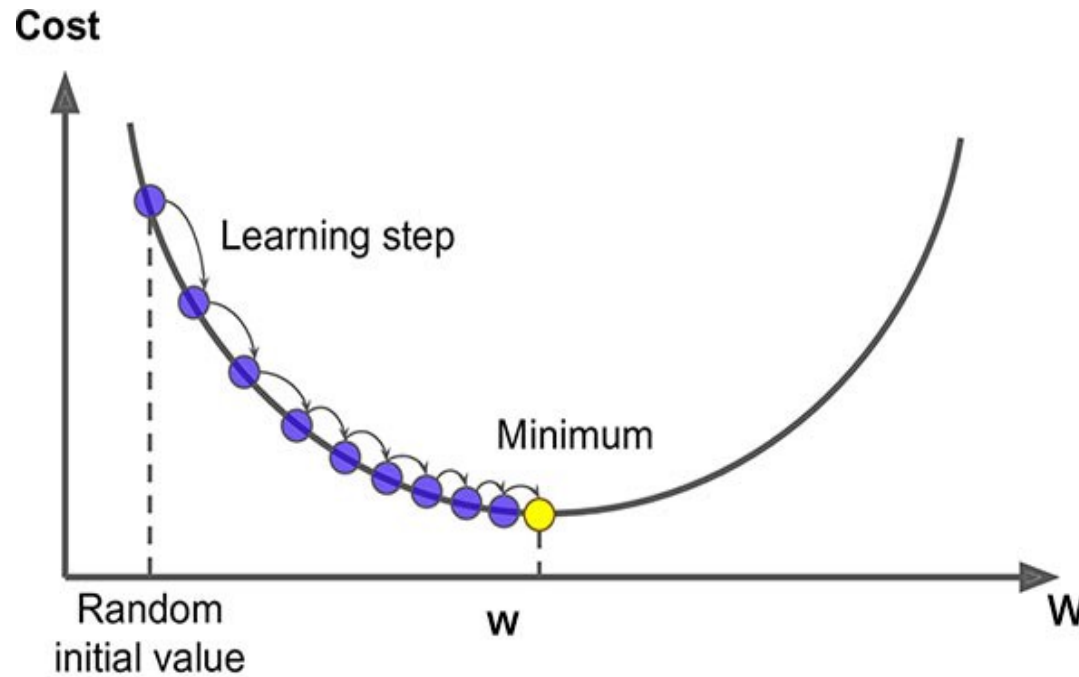


Fig1. Gradient Descent [2]

- Description: - A gradient is nothing but a derivative that defines the effects on outputs of the function with a little bit of variation in inputs.
- Gradient Descent (GD) is a widely used optimization algorithm in deep learning that is used to minimize the cost function of a neural network model during training. It works by iteratively adjusting the weights or parameters of the model in the direction of the negative gradient of the cost function until the minimum of the cost function is reached.
- How the Gradient Descent algorithm works:
- https://youtu.be/gzrQvzYEvYc?si=CHYY_glgmmp6BC34 [3]



Gradient Descent

Description:

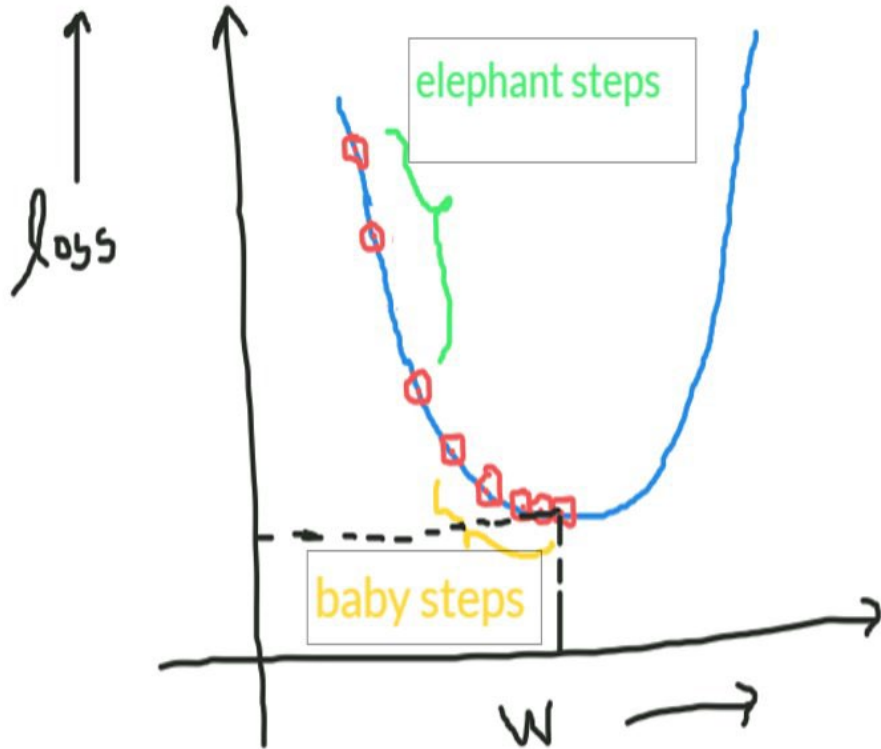


- Imagine we're at the top of a steep mountain, and we have a small ball. Our goal is simple: we want to get the ball to the lowest point in the valley, Gradient Descent is our way of guiding this ball down the mountain towards the lowest point.
- Now think of this Mountain analogy as a mathematical representation of a cost function, and we have to minimise this cost function and bring it to the lowest as possible, So how do we know whether we have to move forward or backwards from this point? How to find the next step? we have a formula for that
- “**New Value=Old Value (Learning Speed*Slope)**”

Fig 2. Gradient Descent analogy. [2]



Gradient Descent



- Now the question is **why the learning rate is put in the equation?** We cannot travel all the points present in between the starting point and the minimal point-
- We need to skip some points-
- We can take elephant steps in the initial phase.
- But while moving near minima, we need to take baby steps, because we may cross over the minima and move to a point where slope increases. So in-order to **control the step size** and the **shift in the graph learning rate was introduced**. Even without the learning rate, we get the minimum value but what we are concerned about is that we want our algorithm to be faster !!![2]

Fig3. Gradient Descent explanation[2]



Epoch

- The one entire passing of training data through the algorithm. It's a hyperparameter that determines the process of training the machine learning model.[4]



Different Variants of Gradient descent

- Stochastic Gradient Descent (SGD)[5]

SGD is an optimization algorithm often used to find the model parameters that correspond to the best fit between predicted and actual outputs.

- Mini-batch Gradient Descent

It divides the training dataset into manageable groups and updates each separately. This strikes a balance between batch gradient descent's effectiveness and stochastic gradient descent's durability.

- Batch gradient Descent

In this calculates the error for each example within the training dataset.[5]



Stochastic Gradient Descent (SGD)

- Working:
- SGD is stochastic in nature i.e it picks up a “random” instance of training data at each step and then computes the gradient making it much faster as there are much fewer data to manipulate at a single time. [6]
- 1 training data example at a time and use forward and backward propagation to update the weight.
- Forward and back propagation:
You tube link: <https://tinyurl.com/bdczd6wk>



Example:

In this the data set used is subject 1 marks, subject 2 marks and study time to predict the marks in subject 3 for one student a time. After completion of the 1st student, the next student's data is taken in consideration.

		Student	Subject 1 Marks (Data)	Subject 2 Marks (Data)	Study Time (Hours) (Data)	Subject 3 (Prediction)
Training Examples	→	1	55	77	4	88
	→	2	45	26	8	43
	→	3	73	89	3	88
	→	4	89	56	9	55

Stochastic Gradient Descent (SGD)

- $W_n = W_{old} - \eta \frac{\partial Loss}{\partial W_{old}}$

Where,

W_n = New Weight

W_{old} = Old Weight

η = Learning rate

$\frac{\partial Loss}{\partial W_{old}}$ = Derivative of loss
function wrt old weight

- $Loss = (Y - Y')^2$

Where,

Y = Actual Observation

Y' = Predicted Observation

SGD changes the parameters for each training sample one at a time for each training example in the dataset.

Old weight is randomly considered number in the beginning and the new weight is then considered.

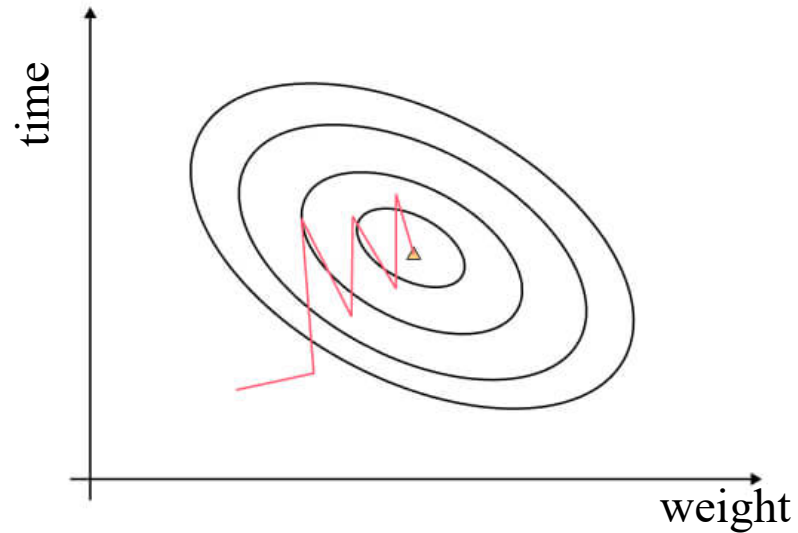
1st predicted observation is randomly considered and upon calculating the new weight, learning of the model occurs till it reaches convergence.



$$W_n = W_{old} - \eta \frac{\partial Loss}{\partial W_{old}}$$

1 training example at a time

Frequent updates can result in noisy gradient signals. This can result in model parameters and cause errors to fly around.[5]



Stochastic Gradient Descent [4]

- P.S.: The meaning of stochastic is “randomly determined”. [7]

Pros / Cons

- Pros

- Instantly see model's performance and improvement rates with frequent updates.
- Easy to understand.
- Faster and less computation power.

- Cons

- Frequent model updates are more computationally intensive than other steepest descent configurations, and it takes considerable time to train the model with large datasets.[5]
- Frequent updates can result in noisy gradient signals. This can result in model parameters and cause errors to oscillate.[5]
- Noisy gradient signals which can be difficult for the algorithm to maintain minimum error for the model.[5]

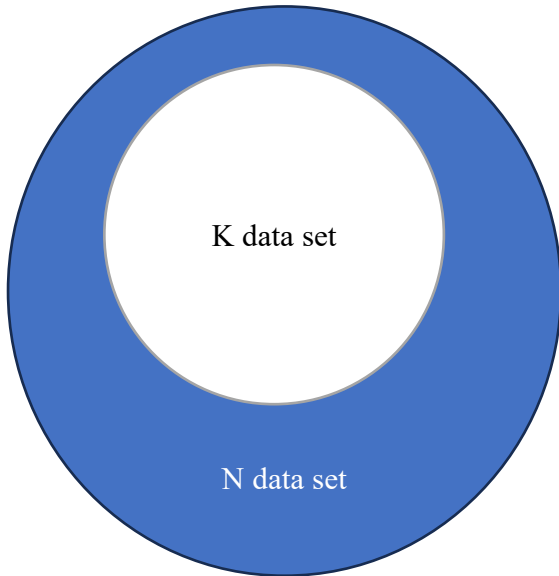
Epoch

Passes for epoch:

- Stochastic – 4

STUDENT	SUBJECT 1 MARKS (DATA)	SUBJECT 2 MARKS (DATA)	STUDY TIME (HOURS) (DATA)	SUBJECT 3 (PREDICTION)
1	55	77	4	88
2	45	26	8	43
3	73	89	3	88
4	89	56	9	55

Mini-batch Gradient Descent

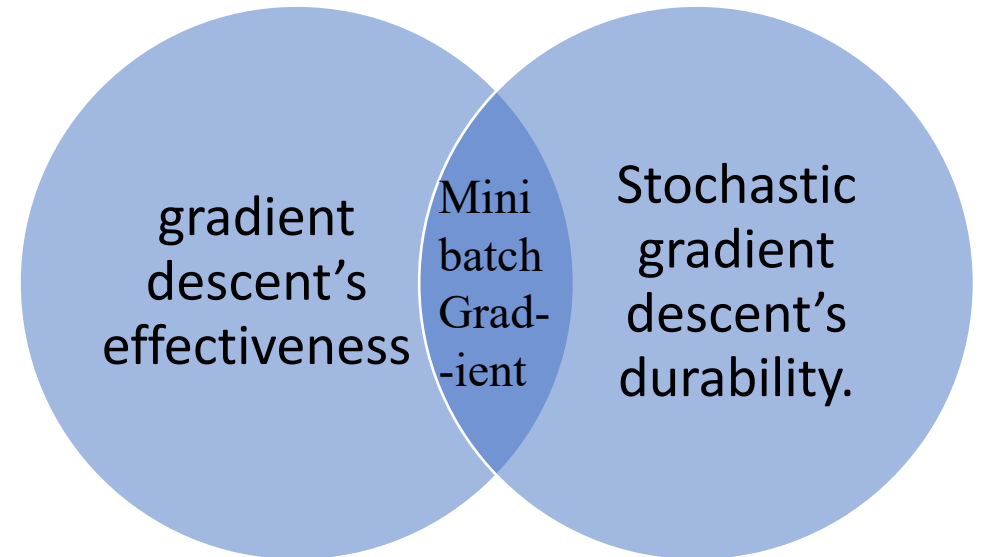


- 'K' train multiple data sets but less than entire data set 'N'.
- $$W_n = W_{old} - \eta \frac{\partial Loss}{\partial W_{old}}$$
- $$Loss = \sum_{i=1}^k (Y - Y')^2$$
- The loss is computed from the dataset under consideration i.e. from $I = 1^{st}$ sample to the k^{th} sample.

Mini-batch Gradient Descent

Working

- We use a batch of a fixed number of training examples which is less than the actual dataset and call it a mini-batch.
- Mini - batch converges faster than gradient descent and with less oscillations than stochastic gradient descent.
Hence, its best of both world.



Example:

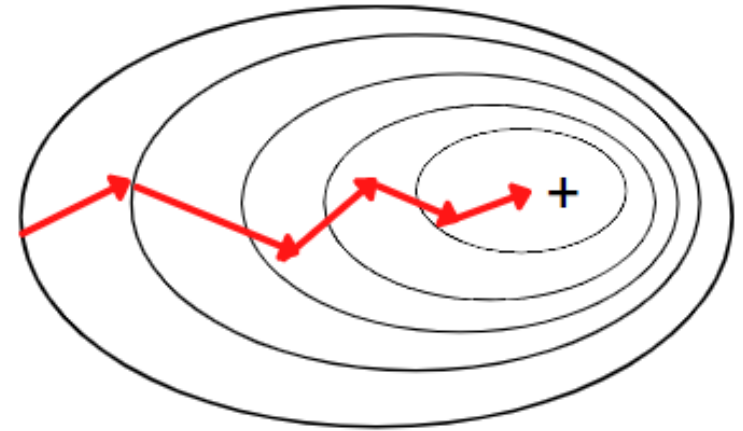
In this the data set used is subject 1 marks, subject 2 marks and study time to predict the marks in subject 3 for two students. After completion of the 1st data set (i.e student 1 & 2) the next data set (Student 3 & 4) is taken in consideration.

Student	Subject 1 Marks (Data)	Subject 2 Marks (Data)	Study Time (Hours) (Data)	Subject 3 (Prediction)
1	55	77	4	88
2	45	26	8	43
3	73	89	3	88
4	89	56	9	55

Training Examples

Pros / Cons & Diagrammatic representation

- Pros
 - Compared to the stochastic gradient descent approach, the model is updated more often, allowing for more reliable convergence and avoiding local minima.
 - Batch updates provide a more computationally efficient process than stochastic gradient descent.
 - Batch processing allows for both the efficiency of not having all the training data in memory and implementing the algorithm.[5]
- Cons
 - Mini-batch requires additional hyperparameters “mini-batch size” to be set for the learning algorithm.[5]
 - Error information should be accumulated over a mini-batch of training samples, such as batch gradient descent.[5]
 - It will generate complex functions.



Mini – Batch Gradient Descent [5]

Epoch

Passes for epoch:

- Mini Batch – 2

STUDENT	SUBJECT 1 MARKS (DATA)	SUBJECT 2 MARKS (DATA)	STUDY TIME (HOURS) (DATA)	SUBJECT 3 (PREDICTION)
1	55	77	4	88
2	45	26	8	43
3	73	89	3	88
4	89	56	9	55



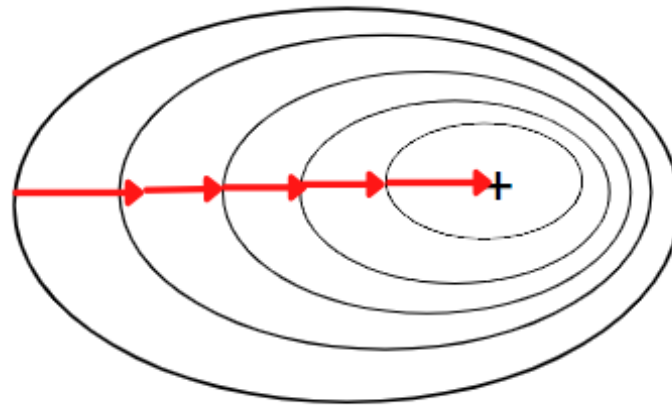
Batch 'vanilla' gradient descent

- Entire training data set 'N'.

- $W_n = W_{old} - \eta \frac{\partial Loss}{\partial W_{old}}$

$$Loss = \sum_{i=1}^n (Y - Y')^2$$

The loss is computed from the dataset under consideration i.e.
from $I = 1^{st}$ sample to the N^{th} sample.



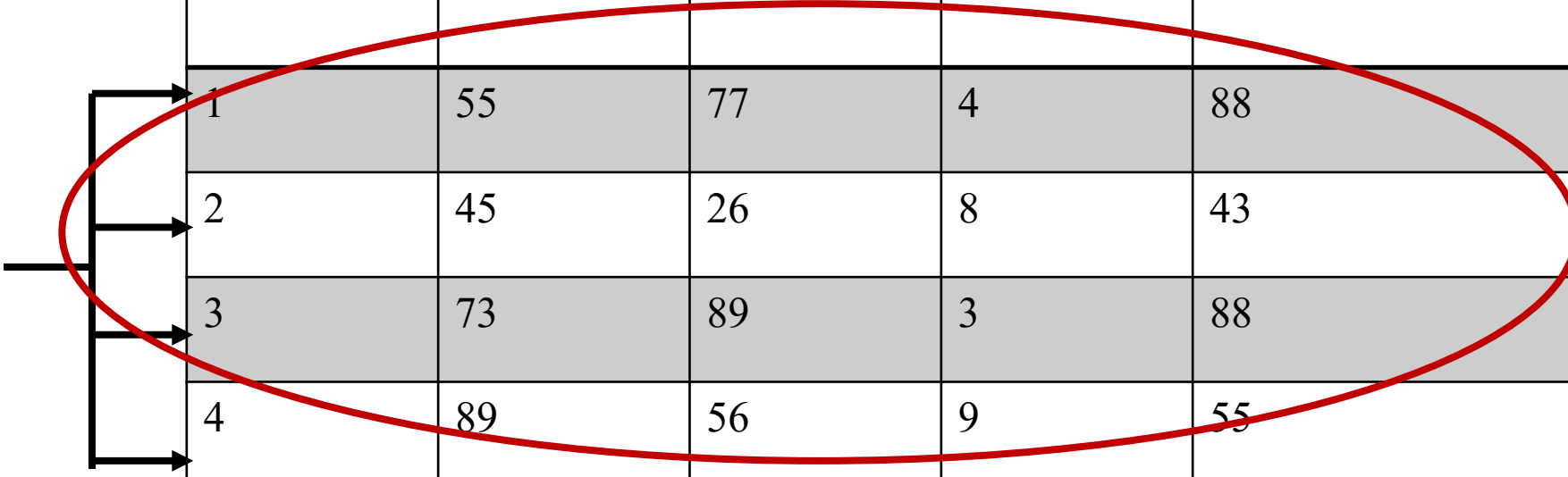
**Batch Gradient Descent [4]
in a complete 'N' data set**



Example

Training Examples

Student	Subject 1 Marks (Data)	Subject 2 Marks (Data)	Study Time (Hours) (Data)	Subject 3 (Prediction)
1	55	77	4	88
2	45	26	8	43
3	73	89	3	88
4	89	56	9	55



Pros / Cons

- Pros

- Fewer model updates mean that this variant of the steepest descent method is more computationally efficient.[5]
- Reducing the update frequency provides a more stable error gradient and a more stable convergence.[5]
- Separating forecast error calculations and model updates provides a parallel processing-based algorithm implementation.

- Cons

- High computation power.
- Costly.
- Slow updates and training speed.



Epoch

Passes for epoch:

- Batch – 1
- Point to remember:
Multiple epochs for a complete training.

STUDENT	SUBJECT 1 MARKS (DATA)	SUBJECT 2 MARKS (DATA)	STUDY TIME (HOURS) (DATA)	SUBJECT 3 (PREDICTION)
1	55	77	4	88
2	45	26	8	43
3	73	89	3	88
4	89	56	9	55

Examples:

- SGD : Regressor (Using Random Values)

<https://tinyurl.com/5yndwxn9>

- SGD : Regressor (Using Boston Housing Dataset)

<https://tinyurl.com/yeyxxxya>

- SGD : Classifier (Using SKlearn Iris dataset)

<https://tinyurl.com/bdf9n85m>



References: -

- [1] M. Banoula, “What is cost function in machine learning [updated]: Simplilearn,” Simplilearn.com, <https://www.simplilearn.com/tutorials/machine-learning-tutorial/cost-function-in-machine-learning> (accessed Oct. 5, 2023).
- [2] P. Ratan, “What does gradient descent actually mean,” Analytics Vidhya, <https://www.analyticsvidhya.com/blog/2020/10/what-does-gradient-descent-actually-mean/> (accessed Oct. 5, 2023).
- [3] “Gradient descent simple explanation|gradient descent machine learning|gradient descent algorithm,” YouTube, <https://www.youtube.com/watch?v=gzrQvzYEvYc&t=1s> (accessed Oct. 5, 2023).
- [4] Simplilearn, “What is epoch in machine learning?: Simplilearn,” Simplilearn.com, <https://tinyurl.com/2r86me9d> (accessed Oct. 5, 2023).
- [5] “Gradient Descent,” <https://www.analyticsvidhya.com/blog/2020/10/what-does-gradient-descent-actually-mean/> (accessed on sep28,2023).
- [6] “Stochastic gradient descent,” Engati, <https://tinyurl.com/ye6cdn85>. (accessed Oct. 10, 2023).
- [7] Oxford English Dictionary, <https://www.oed.com/search/dictionary/?scope=Entries&q=motor> (accessed Sep. 28, 2023).

