Gradient Descent

20CP401T

Himanshu K. Gajera
Department of Computer Science & Engineering
Pandit Deendayal Energy University, Gandhinagar

How Does the Gradient Descent Algorithm Work in Machine Learning?

- Imagine you’re lost in a dense forest with no map or compass.
- What do you do?
- You follow the path of steepest descent, taking steps in the direction that decreases the slope and brings you closer to your destination.

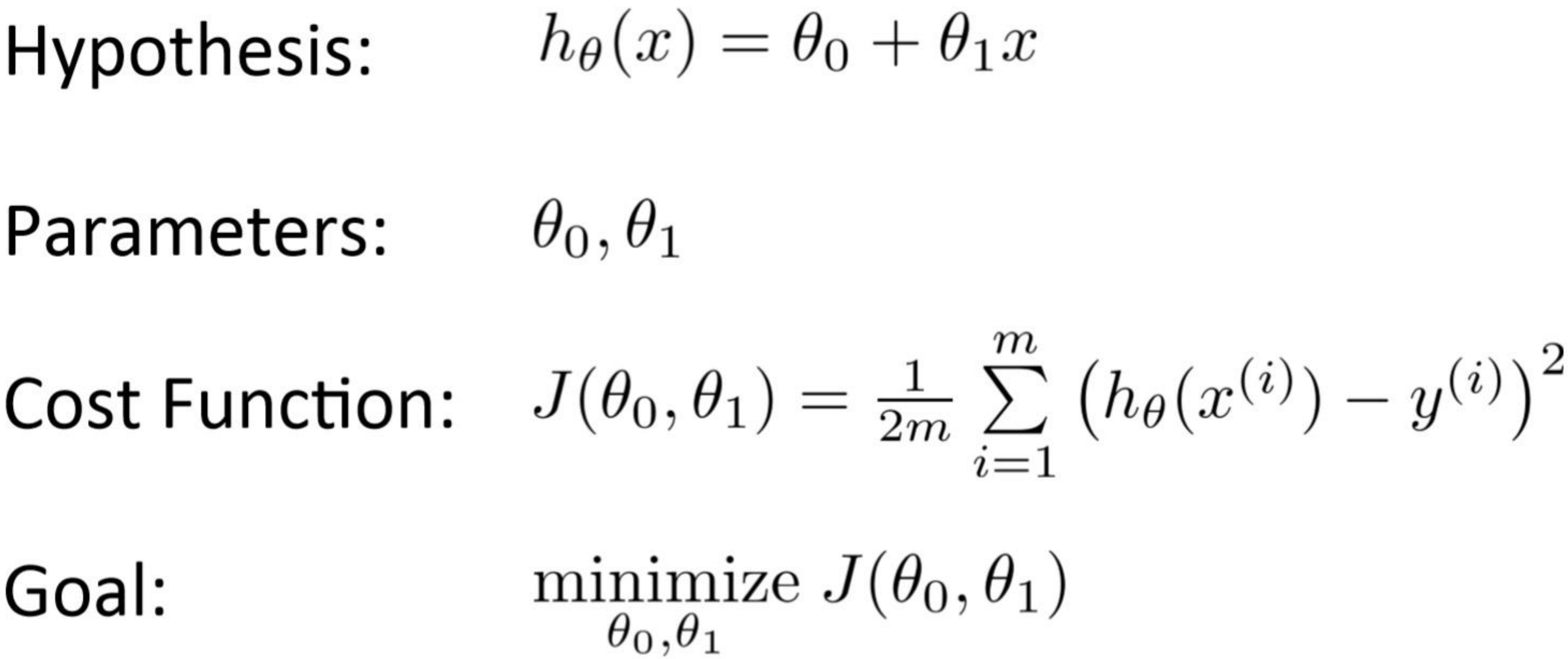
- Similarly, gradient descent is the go-to algorithm for navigating the complex landscape of machine learning.

- It helps models find the optimal set of parameters by iteratively adjusting them in the opposite direction of the gradient.

What is a Cost Function?

- It is a function that measures the performance of a model for any given data.
- Cost Function quantifies the error between predicted values and expected values and presents it in the form of a single real number.
- After making a hypothesis with initial parameters, we calculate the Cost function.
- And with a goal to reduce the cost function, we modify the parameters by using the Gradient descent algorithm over the given data.
- Here’s the mathematical representation for it:

What is a Cost Function?



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

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Source: Coursera

What is Gradient Descent?

- Gradient descent is an optimization algorithm used in machine learning to minimize the cost function by iteratively adjusting parameters in the direction of the negative gradient, aiming to find the optimal set of parameters.

- The cost function represents the discrepancy between the predicted output of the model and the actual output.

- The goal of gradient descent is to find the set of parameters that minimizes this difference and improves the model’s performance.

- The algorithm operates by calculating the gradient of the cost function, which indicates the direction and magnitude of steepest ascent.

- The objective is to minimize the cost function, gradient descent moves in the opposite direction of the gradient, known as the negative gradient direction.

What is Gradient Descent?

- By iteratively updating the model’s parameters in the negative gradient direction, gradient descent gradually converges towards the optimal set of parameters that yields the lowest cost.

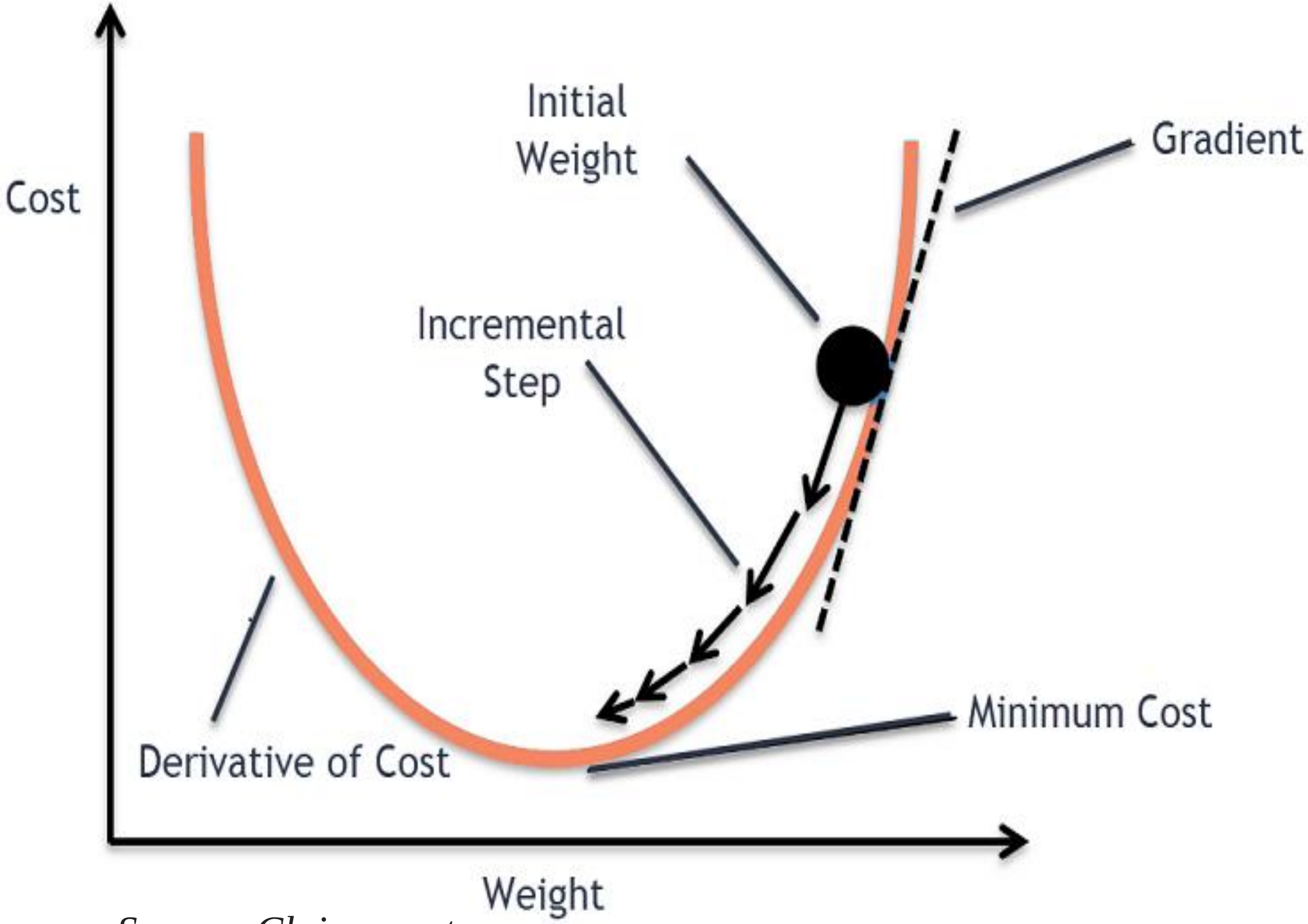
- The learning rate, a hyperparameter, determines the step size taken in each iteration, influencing the speed and stability of convergence.

- Gradient descent can be applied to various machine learning algorithms, including linear regression, logistic regression, neural networks, and support vector machines.

- It provides a general framework for optimizing models by iteratively refining their parameters based on the cost function.

Example of Gradient Descent

Finding the lowest point in a hilly landscape. (Source: Fisseha Berhane)

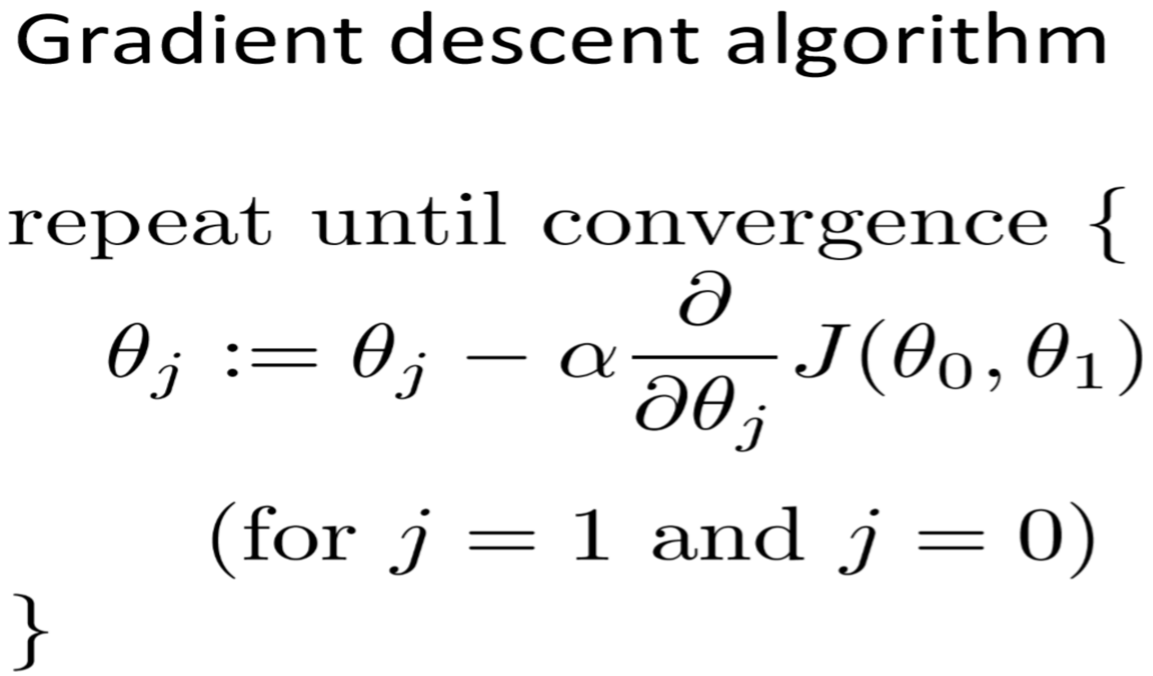


Source: Clairvoyant

What is Gradient Descent?

- The goal of the gradient descent algorithm is to minimize the given function (say cost function). To achieve this goal, it performs two steps iteratively:

1. Compute the gradient (slope), the first order derivative of the function at that point
2. Make a step (move) in the direction opposite to the gradient, opposite direction of slope increase from the current point by alpha times the gradient at that point



Gradient descent algorithm

```
repeat until convergence {
    $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$
        (for $j = 1$ and $j = 0$)
}
```

Alpha is called Learning rate – a tuning parameter in the optimization process.

How Does Gradient Descent Work?

1. Gradient descent is an optimization algorithm used to minimize the cost function of a model.

2. The cost function measures how well the model fits the training data and is defined based on the difference between the predicted and actual values.

3. The gradient of the cost function is the derivative with respect to the model’s parameters and points in the direction of the steepest ascent.

4. The algorithm starts with an initial set of parameters and updates them in small steps to minimize the cost function.

5. In each iteration of the algorithm, the gradient of the cost function with respect to each parameter is computed.

How Does Gradient Descent Work?

6. The gradient tells us the direction of the steepest ascent, and by moving in the opposite direction, we can find the direction of the steepest descent.

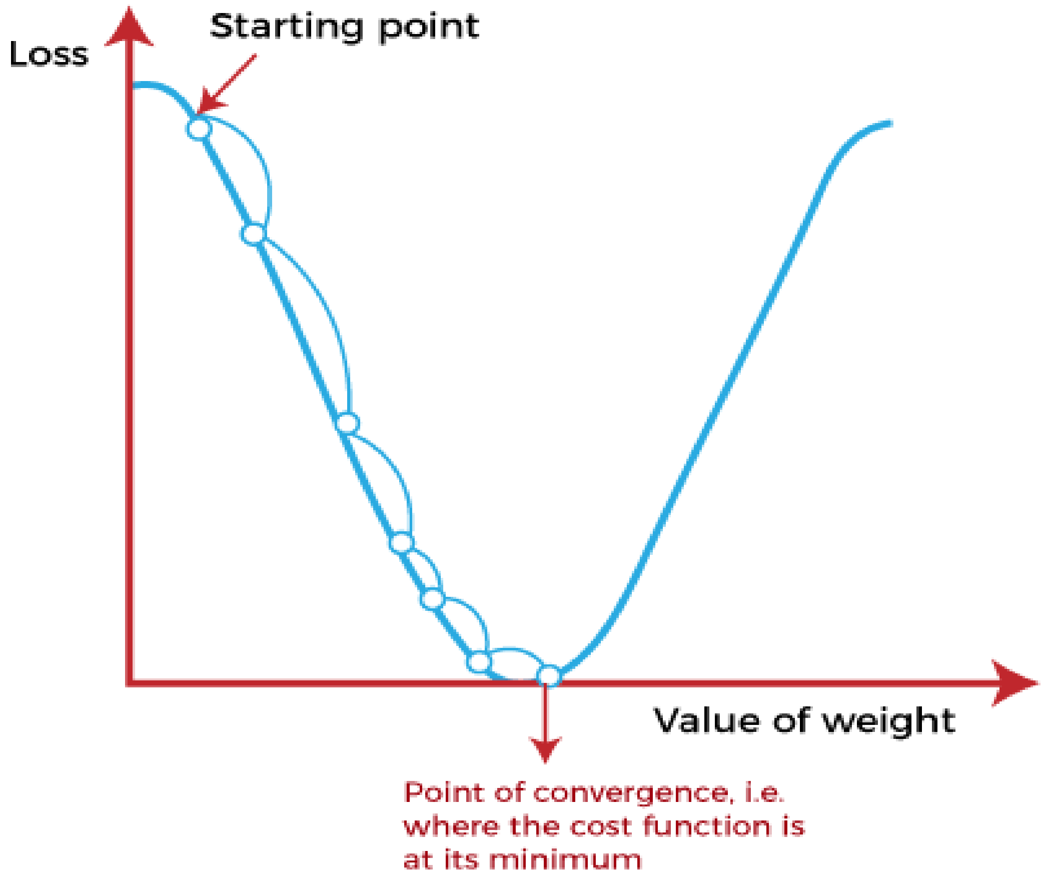
7. The size of the step is controlled by the learning rate, which determines how quickly the algorithm moves towards the minimum.

8. The process is repeated until the cost function converges to a minimum, indicating that the model has reached the optimal set of parameters.

9. There are different variations of gradient descent, including batch gradient descent, stochastic gradient descent, and mini-batch gradient descent, each with its own advantages and limitations.

10. Efficient implementation of gradient descent is essential for achieving good performance in machine learning tasks. The choice of the learning rate and the number of iterations can significantly impact the performance of the algorithm.

How Does Gradient Descent Work?



Learning Rate

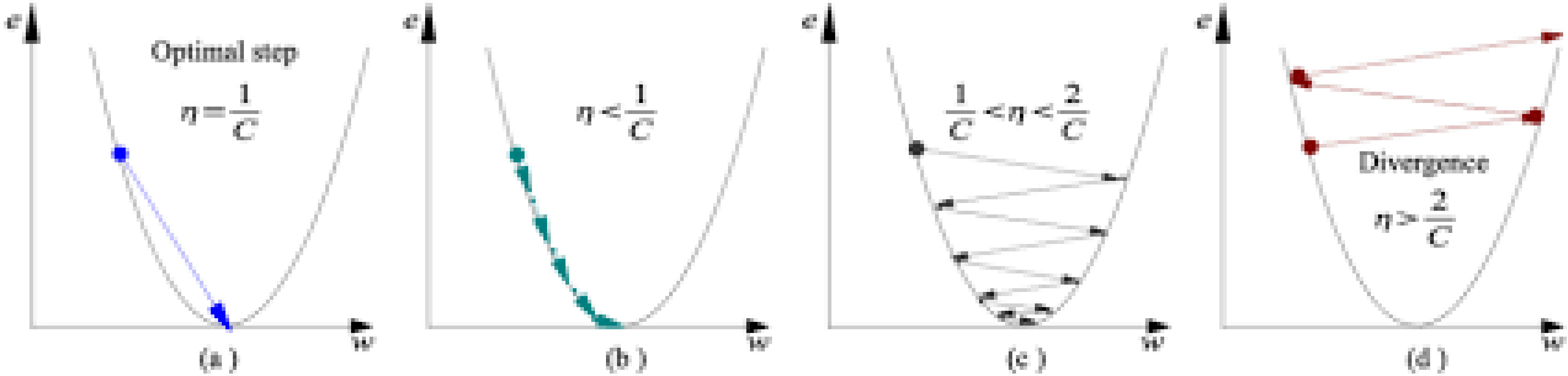
- We have the direction we want to move in, now we must decide the size of the step we must take.

- It must be chosen carefully to end up with local minima.

- If the learning rate is too high, we might OVERSHOOT the minima and keep bouncing, without reaching the minima.

- If the learning rate is too small, the training might turn out to be too long

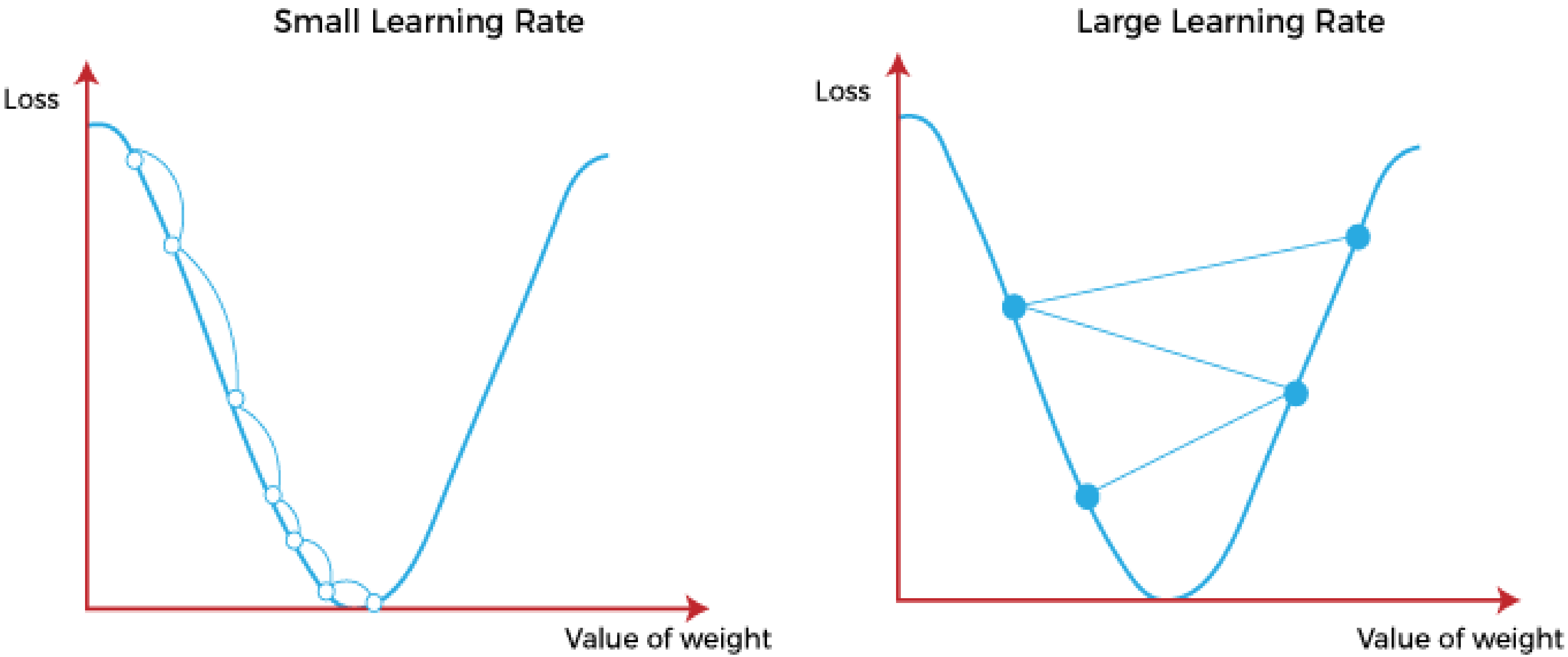
Learning Rate



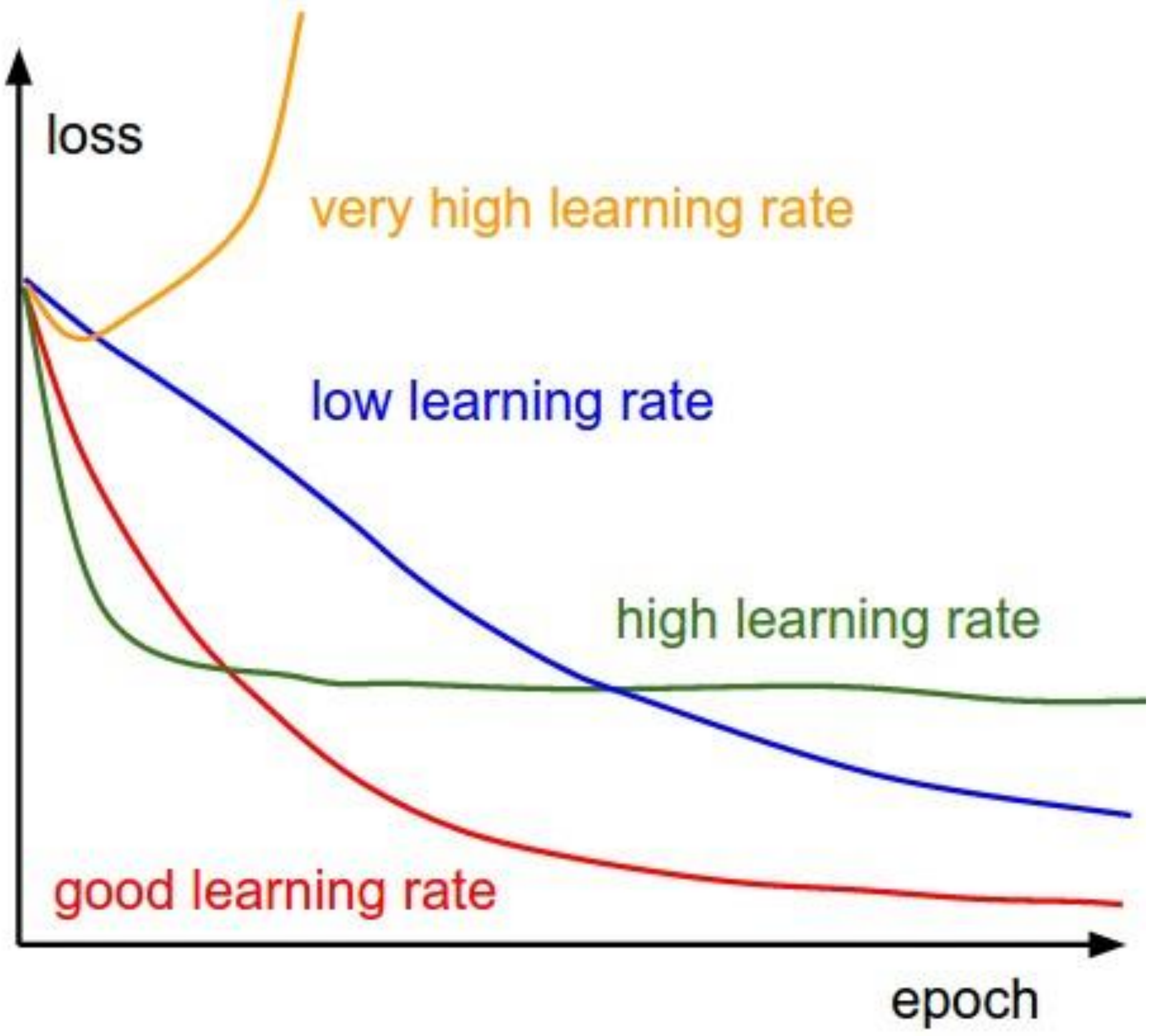
Source: Coursera

a) Learning rate is optimal, model converges to the minimum
b) Learning rate is too small, it takes more time but converges to the minimum
c) Learning rate is higher than the optimal value, it overshoots but converges (1/C < η <2/C)
d) Learning rate is very large, it overshoots and diverges, moves away from the minima, performance decreases on learning

Learning Rate



Learning Rate



Types of Gradient Descent

- The choice of gradient descent algorithm depends on the problem at hand and the size of the dataset.

- Batch gradient descent is suitable for small datasets.

- Stochastic gradient descent is more suitable for large datasets.

- Mini-batch gradient descent is a good compromise between the two and is often used in practice.

Types of Gradient Descent

Batch Gradient Descent

- Batch gradient descent (BGD) is used to find the error for each point in the training set and update the model after evaluating all training examples.
- This procedure is known as the training epoch.
- In simple words, it is a greedy approach where we have to sum over all examples for each update.

Advantages of Batch gradient descent:

- It produces less noise in comparison to other gradient descent.
- It produces stable gradient descent convergence.
- It is Computationally efficient as all resources are used for all training samples.

Types of Gradient Descent

Stochastic gradient descent

- Stochastic gradient descent (SGD) is a type of gradient descent that runs one training example per iteration.
- Or in other words, it processes a training epoch for each example within a dataset and updates each training example's parameters one at a time.
- As it requires only one training example at a time, hence it is easier to store in allocated memory.
- However, it shows some computational efficiency losses in comparison to batch gradient systems

Advantages of Stochastic gradient descent:

- It is easier to allocate in desired memory.
- It is relatively fast to compute than batch gradient descent.
- It is more efficient for large datasets.

Types of Gradient Descent

MiniBatch Gradient Descent

- Mini Batch gradient descent is the combination of both batch gradient descent and stochastic gradient descent.
- It divides the training datasets into small batch sizes then performs the updates on those batches separately.
- Splitting training datasets into smaller batches make a balance to maintain the computational efficiency of batch gradient descent and speed of stochastic gradient descent.
- Hence, we can achieve a special type of gradient descent with higher computational efficiency and less noisy gradient descent.

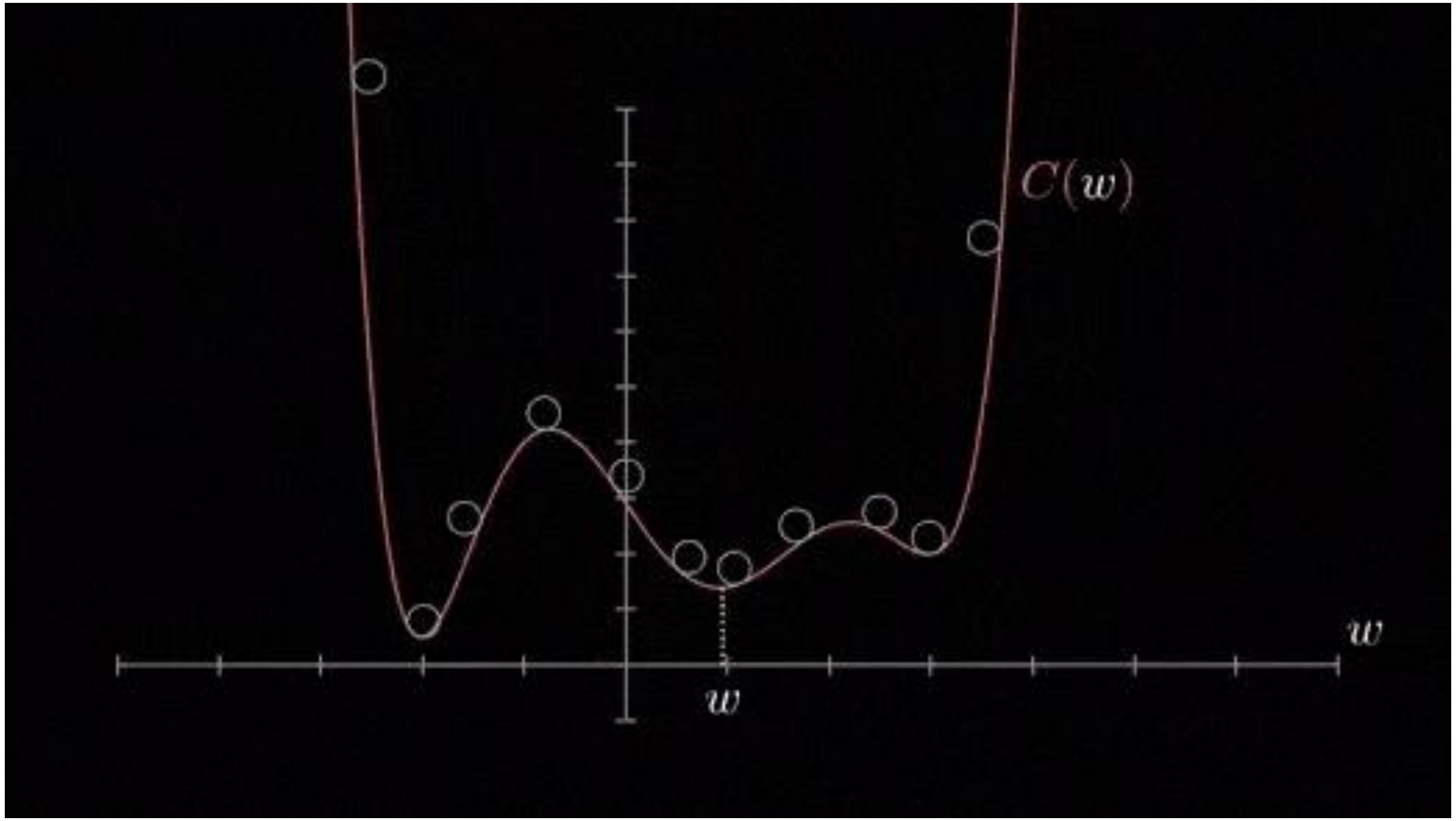
Advantages of Mini Batch gradient descent:

- It is easier to fit in allocated memory.
- It is computationally efficient.
- It produces stable gradient descent convergence.

Challenges of Gradient Descent

Local Optima:

- The cost function may consist of many minimum points. The gradient may settle on any one of the minima, which depends on the initial point (i.e initial parameters(theta)) and the learning rate.
- Gradient descent can converge to local optima instead of the global optimum, especially if the cost function has multiple peaks and valleys.



Challenges of Gradient Descent

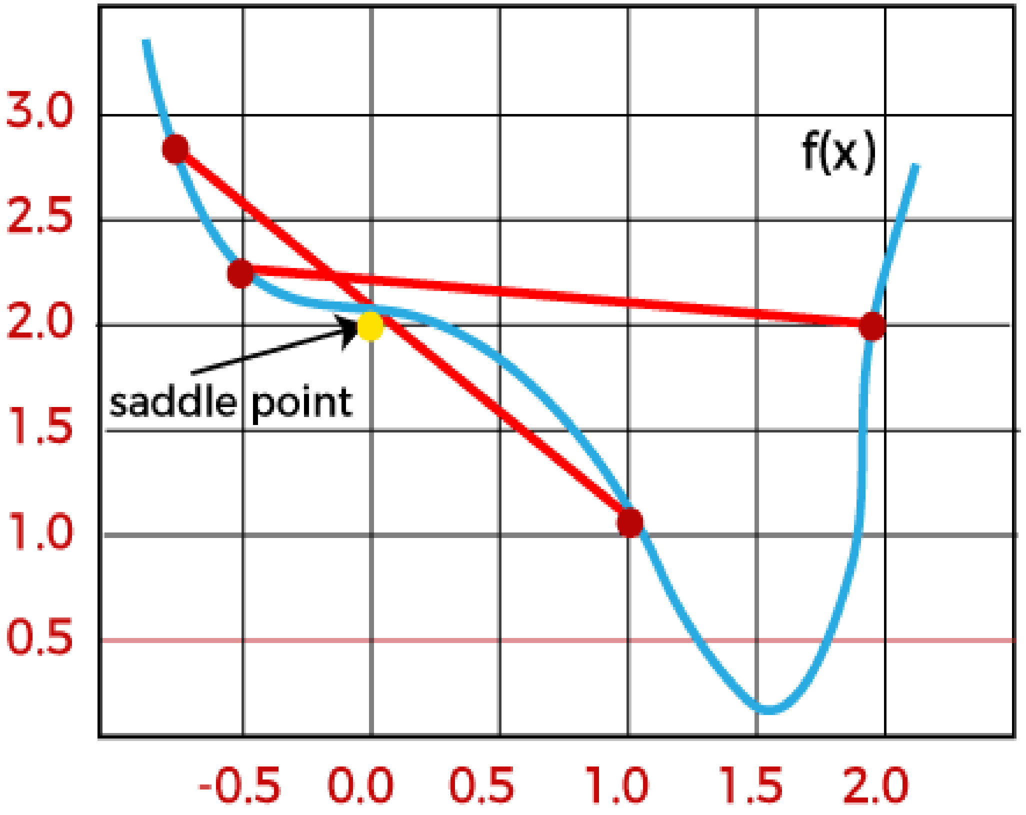
- **Learning Rate Selection**: The choice of learning rate can significantly impact the performance of gradient descent. If the learning rate is too high, the algorithm may overshoot the minimum, and if it is too low, the algorithm may take too long to converge.

- **Overfitting**: Gradient descent can overfit the training data if the model is too complex or the learning rate is too high. This can lead to poor generalization performance on new data.

- **Convergence Rate**: The convergence rate of gradient descent can be slow for large datasets or high-dimensional spaces, which can make the algorithm computationally expensive.

Challenges of Gradient Descent

- **Saddle Points**: In high-dimensional spaces, the gradient of the cost function can have saddle points, which can cause gradient descent to get stuck in a plateau instead of converging to a minimum.



Challenges of Gradient Descent

Vanishing and Exploding Gradient:

- In a deep neural network, if the model is trained with gradient descent and backpropagation, there can occur two more issues other than local minima and saddle point.

Vanishing Gradients:

- Vanishing Gradient occurs when the gradient is smaller than expected. During backpropagation, this gradient becomes smaller that causing the decrease in the learning rate of earlier layers than the later layer of the network. Once this happens, the weight parameters update until they become insignificant.

Exploding Gradient:

- Exploding gradient is just opposite to the vanishing gradient as it occurs when the Gradient is too large and creates a stable model. Further, in this scenario, model weight increases, and they will be represented as NaN.
- This problem can be solved using the dimensionality reduction technique, which helps to minimize complexity within the model.

Challenges of Gradient Descent

- To overcome these challenges, several variations of gradient descent have been developed.
  - Adaptive learning rate methods
  - Momentum-based methods
  - Second-order methods.
- Additionally, choosing the right regularization method, model architecture, and hyperparameters can also help improve the performance of gradient descent.