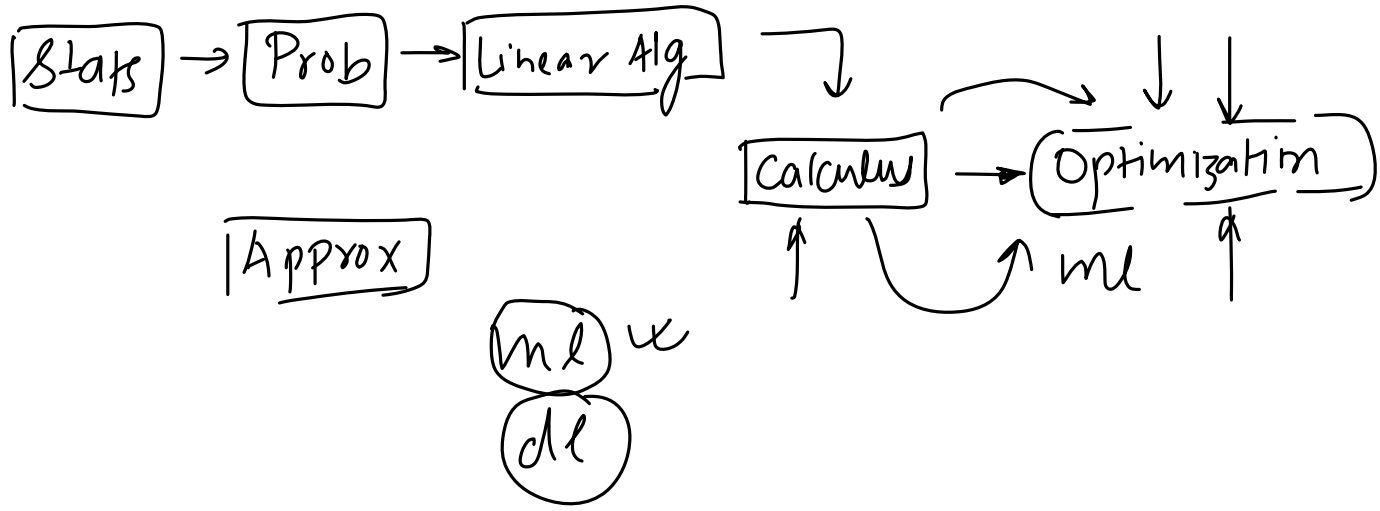
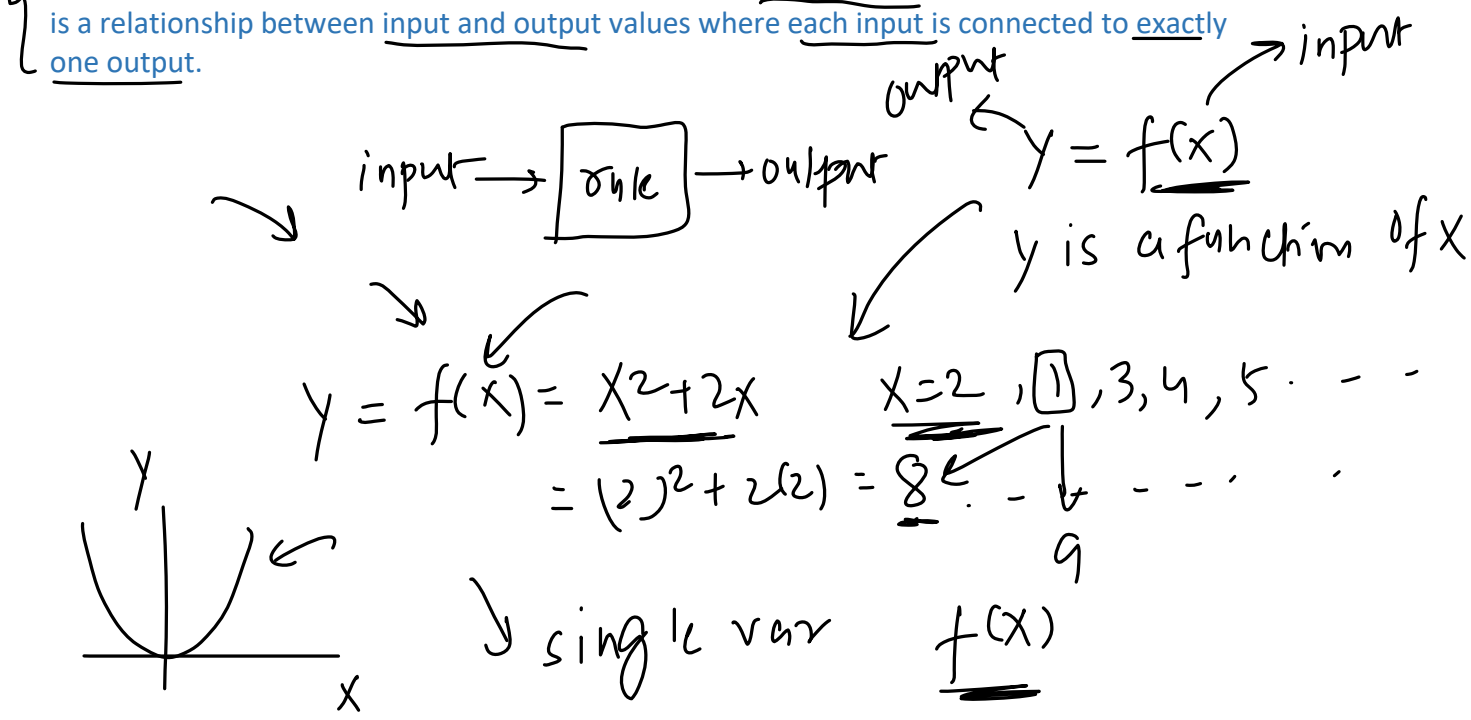
21 April 2023 20:05



Functions

21 April 2023 14:28

A function is a mathematical rule that takes an input value, processes it according to a specific formula or set of instructions, and produces a unique output value. In other words, a function is a relationship between input and output values where each input is connected to exactly one output.



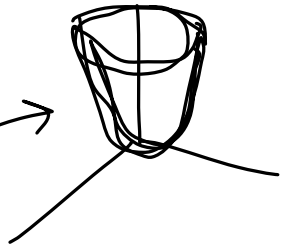
Multivariable Functions

21 April 2023 16:35

$f(x)$ $1+1 \rightarrow 2$ dim

$z = f(x,y) = x^2 + y^2$ $2+1$

output input



$n+1$ dim

$x=1 \quad y=1 \longrightarrow z=2$

$x=1 \quad y=1 \longrightarrow z \neq 3$

$f(x_1, x_2 x_3 \ldots x_n)$

Parameters in a Function

ax^2+bx+c

21 April 2023 16:36

In mathematics, parameters of a function are the variables that are used to define the behaviour of the function. The parameters influence the function's output by determining how the input values are processed.

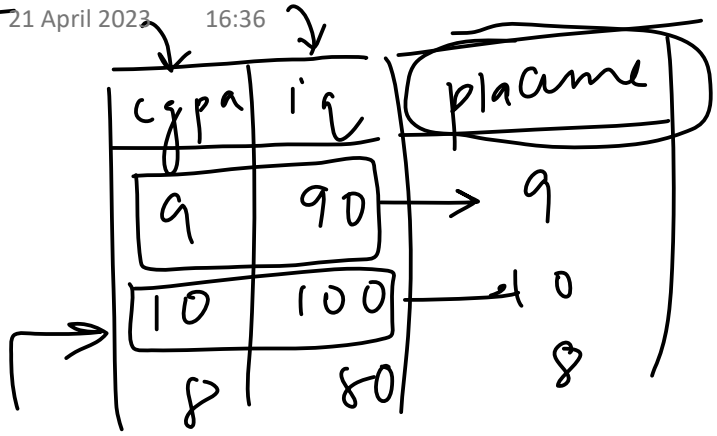
The parameters are the constants or coefficients that appear in the function's formula. For example, in the quadratic function f(x) = ax^2 + bx + c, 'a', 'b', and 'c' are the parameters of the function. By changing the values of these parameters, you can modify the shape and position of the parabola represented by the function.

$y=f(x)=5x^2$ $6x^2$ $y=f(x)=ax^2$

parameter input parameter

ML models as Mathematical Function

21 April 2023 16:36



ml model

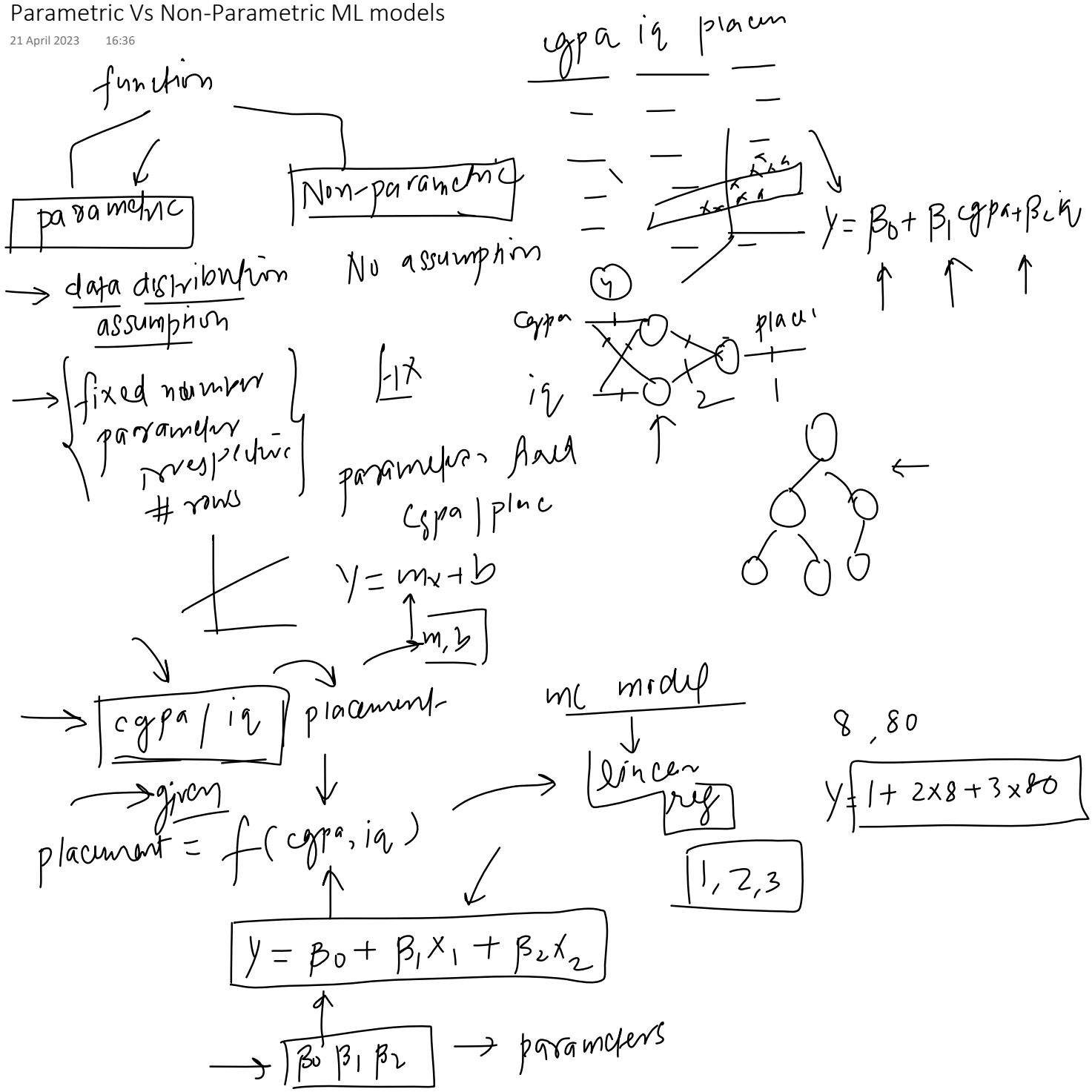
y → placement

x → cgpa, iq

placement = f(cgpa, iq)

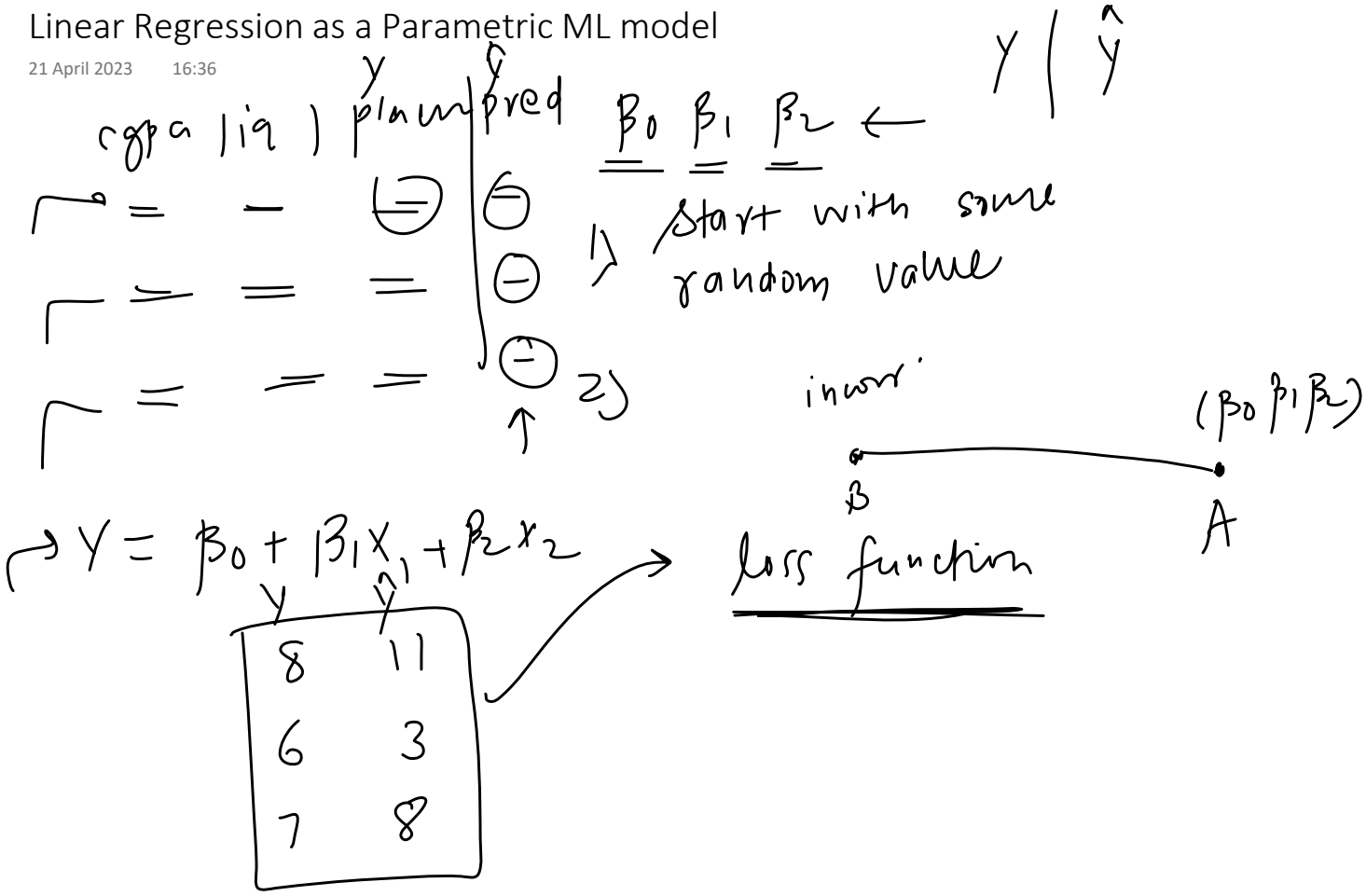
Parametric Vs Non-Parametric ML models

21 April 2023 16:36



Linear Regression as a Parametric ML model

21 April 2023 16:36

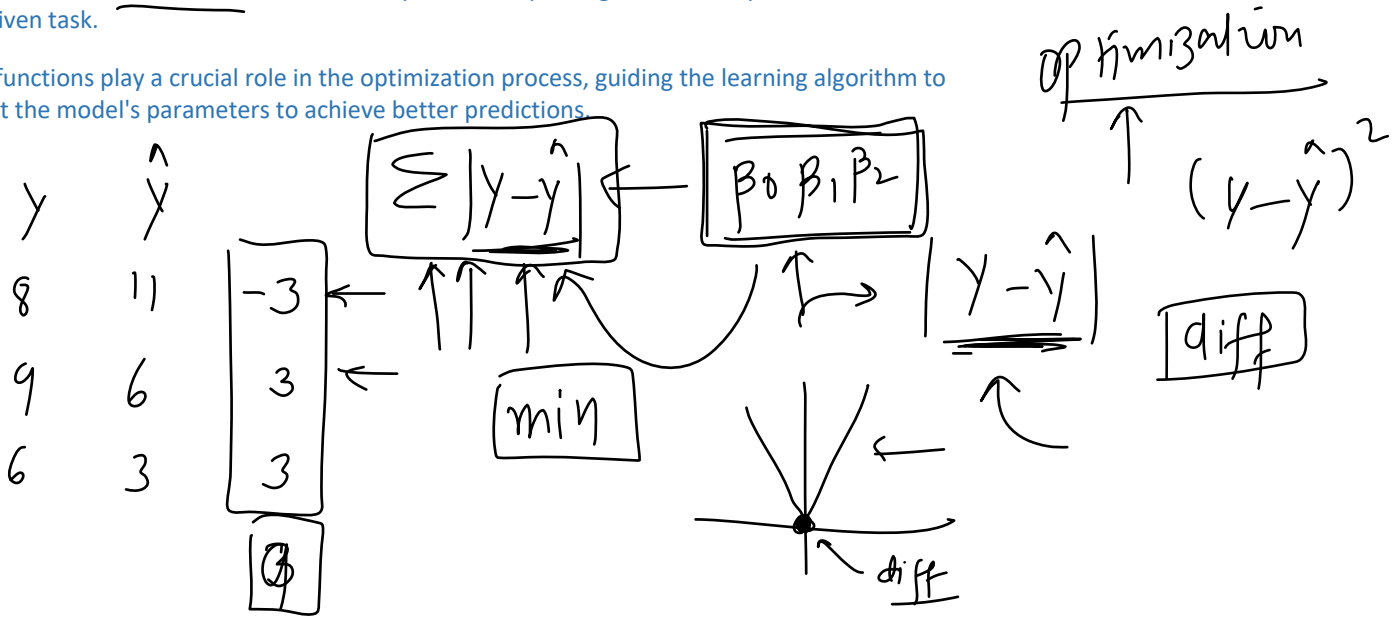


Loss Function

21 April 2023 16:37

A loss function, also known as a cost function or objective function, is a mathematical function that measures the difference between the predicted output and the actual target values in a machine learning model. The primary goal of training a machine learning model is to minimize the value of the loss function, which corresponds to improving the model's performance on the given task.

Loss functions play a crucial role in the optimization process, guiding the learning algorithm to adjust the model's parameters to achieve better predictions.



How to select a good Loss Function

21 April 2023 16:37

1. Problem type: The choice of a loss function depends on the type of problem you are solving. For example, in regression tasks, mean squared error (MSE) or mean absolute error (MAE) are commonly used. For binary classification, cross-entropy loss or hinge loss can be employed. For multi-class classification, categorical cross-entropy or multi-class hinge loss can be used. Choose a loss function that aligns with the objectives of the specific problem you are addressing.

2. Robustness to outliers: Some loss functions, like mean squared error, are more sensitive to outliers, which can lead to a model that is overly influenced by extreme values. If your dataset contains outliers or is prone to noise, consider using a loss function that is more robust to outliers, such as mean absolute error (MAE) or Huber loss.

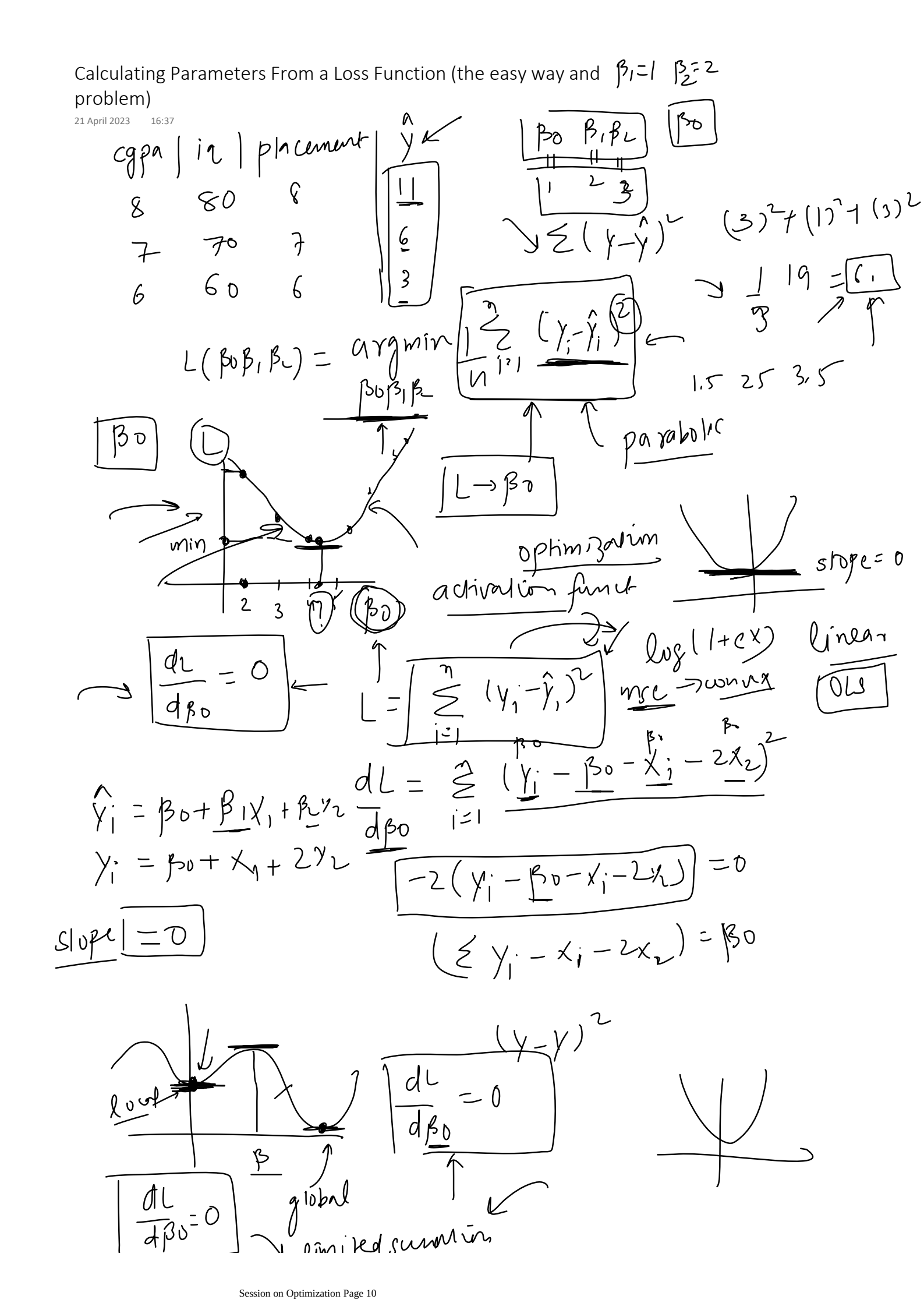
3. Interpretability and ease of use: A good loss function should be interpretable and easy to implement. Simple loss functions like mean squared error or cross-entropy loss are widely used because they are easy to understand, compute, and differentiate. When possible, opt for a loss function that is easy to work with and can be easily incorporated into your optimization process.

4. Differentiability: Most optimization algorithms, like gradient descent, require the loss function to be differentiable. Choose a loss function that has continuous first-order derivatives, which makes it easier to compute the gradients needed for optimization.

5. Compatibility with the model: Ensure that the chosen loss function is compatible with the model architecture you are using. Some models have specific requirements or assumptions about the loss function. For example, linear regression assumes a Gaussian noise distribution, which is why mean squared error is a suitable loss function in that case.

Calculating Parameters From a Loss Function (the easy way and problem)

21 April 2023 16:37

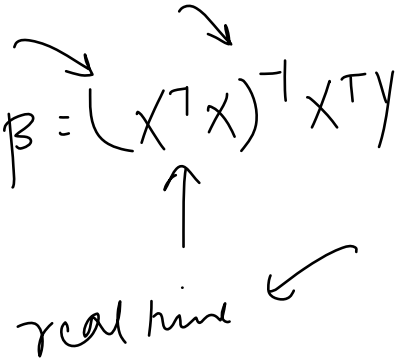


$\frac{\partial L}{\partial \beta_0} = 0$ → global → limited summation

Problem with the easy way

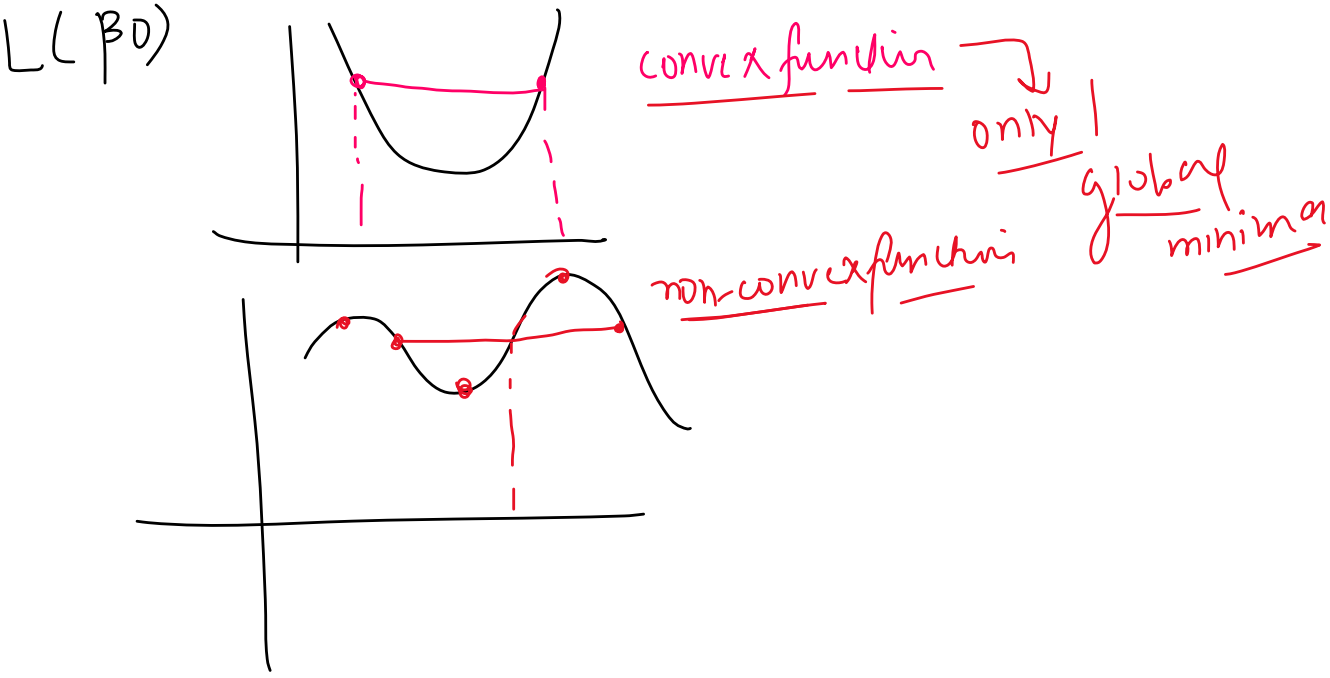
21 April 2023 16:44

1. Non-convexity: The loss function may not always be convex, meaning that it might have multiple local minima and maxima. In such cases, setting the gradient to zero might lead to a local minimum or maximum, which is not necessarily the global minimum (the optimal solution).
2. Complexity: For some models, the loss function can be highly complex, and finding the analytical solution by setting the gradient to zero might be computationally expensive or even impossible. This is particularly true for deep learning models, where the loss functions involve a large number of parameters and complex relationships between them.
3. Scalability: In large-scale machine learning problems with massive amounts of data or high-dimensional feature spaces, computing the analytical solution by setting the gradient to zero can be computationally prohibitive due to the high cost of processing and storing the data.
4. Online learning and streaming data: In some applications, the data is not available all at once but arrives in a continuous stream. In these scenarios, models need to be updated incrementally as new data arrives, and an analytical solution would not be practical. Gradient descent and its variants, such as stochastic gradient descent, are well-suited for online learning and handling streaming data.



Convex And Non Convex Loss Functions

21 April 2023 16:38



Gradient Descent

21 April 2023 16:38

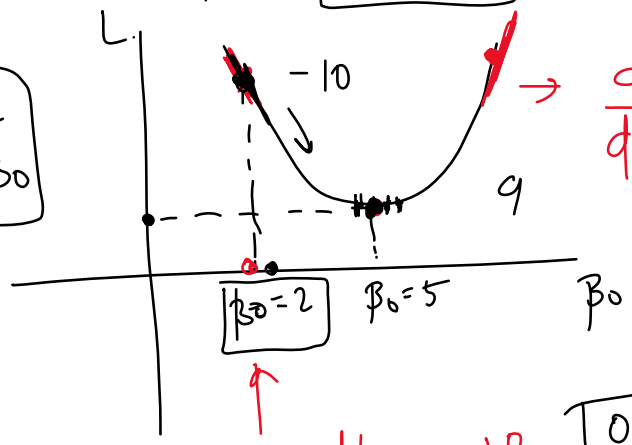
optimization techniques (first order deri.)

- linear reg
- logistic
- perceptron
- Deep learning → ANN, CNN, RNN

SVM → quadratic prog
PCA

$\frac{dL}{d\beta_0}$ $f'(x)$ $f''(x)$ $\frac{d^2L}{d\beta_0^2}$

β_0 ? $\beta_1 = \beta_0$



$\frac{dL}{d\beta_0}$ → $\frac{dL}{d\beta}$

convex

10 maxima

$\beta_{0\,new} = \beta_{0\,old} - (1)\frac{dL}{d\beta}$

learning

$\beta_0 = 2$ $\frac{dL}{d\beta} = -10$ 0.01

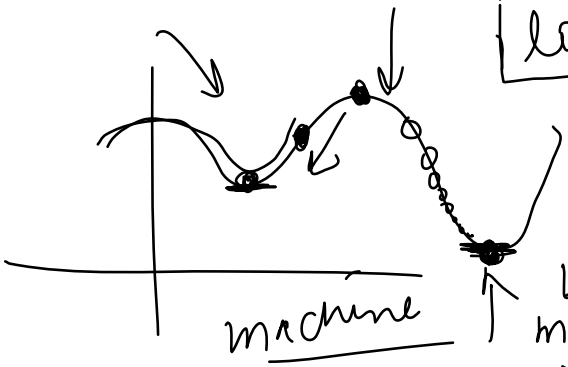
$2 - (-10) = 12$

$12 - 10 = 2$

$\beta_0 = \beta_{0\,old} - \eta \frac{dL}{d\beta_0}$

$\beta_0 = 2 + 0.01 \times 10 = 2.1$

$2.1 + 0.01 \times 9 = 2.$



local minima

gradient

variants

machine

momentum

NAG | RMP / Adam

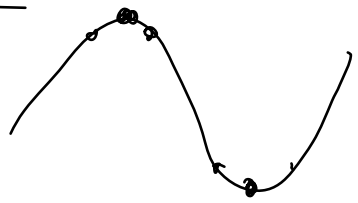
deep learning

12 β_{N+1} — slope = 0 $\beta_{new} = \beta_{old}$

$\beta_{new} = \beta_{old}$

$\beta_{0_{new}} = \beta_{old} - \text{slope} = 0$

epochs



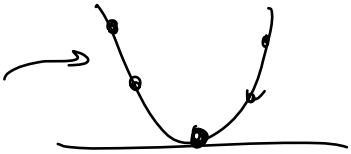
linear reg

$\beta = 5$ min max

$f(\beta) = \beta^2 + 2$

$= 27$

$\beta = 4.9$



convex

1 minima epochs $(4.9)^2 + 2 = 26$

$\beta_n = \beta_0 - \eta \frac{\partial L}{\partial \beta}$

Gradient Descent with multiple Parameters

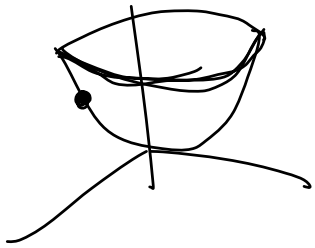
21 April 2023 16:39

$L(\beta_0) \rightarrow$ cols $x_1\ x_2 \cdots x_n$

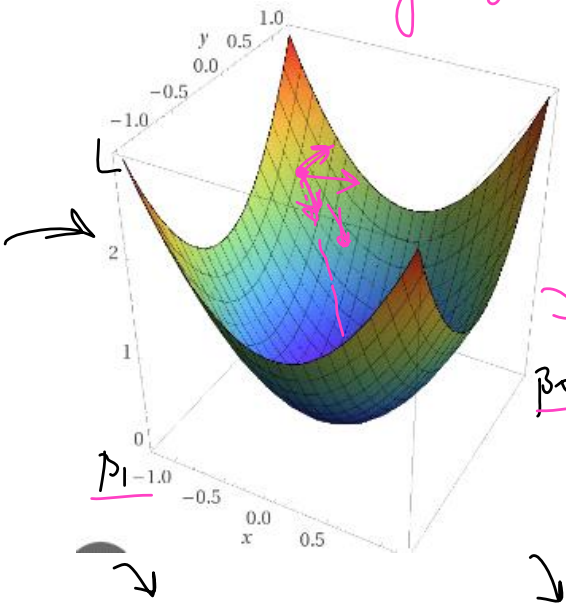
$\beta_0\ \beta_1\ \beta_2 \cdots \beta_n$

$(n+1)$

$L\ \beta_0\ \beta_1$



gradient descent

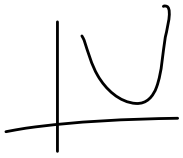


$\frac{\partial L}{\partial \beta_1}\ \frac{\partial L}{\partial \beta_0}$ $\beta_0\ \beta_1\ \beta_2$

$$\beta_0 = \beta_0 - \eta \frac{\partial L}{\partial \beta_0}$$

$$\beta_1 = \beta_1 - \eta \frac{\partial L}{\partial \beta_0}$$

$$\beta_2 = \beta_2 - \eta \frac{\partial L}{\partial \beta_2}$$

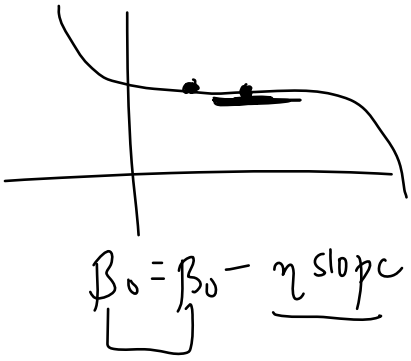


$$\nabla L(\beta_0 \cdots \beta_n) = \left(\frac{\partial L}{\partial \beta_0}, \frac{\partial L}{\partial \beta_1} \cdots \frac{\partial L}{\partial \beta_n}\right)$$

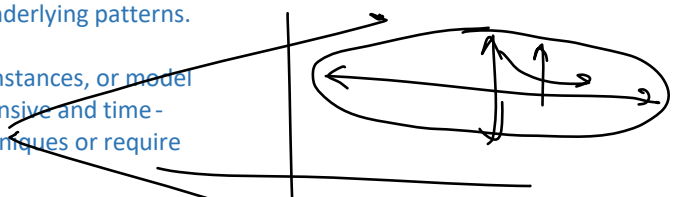
Problems faced in Optimization → GD

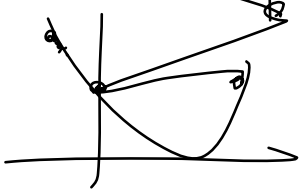
21 April 2023 16:39

1. Non-convexity: For many machine learning models, such as artificial neural networks, the loss function is non-convex, which means it has a complex landscape with multiple local minima, maxima, and saddle points. This makes it difficult for optimization algorithms to find the global minimum and can result in suboptimal solutions.
2. Ill-conditioning: The loss function may be ill-conditioned, meaning the gradients in some dimensions are much larger than in others. This can cause gradient-based optimization algorithms, such as gradient descent, to oscillate and converge slowly.
3. Vanishing and exploding gradients: In deep neural networks, the gradients can become very small (vanish) or very large (explode) as they propagate through the layers. This can lead to slow convergence or unstable training dynamics, making it difficult to optimize the loss function.
4. Overfitting: When optimizing the loss function, the algorithm may overfit the training data, resulting in a model that performs poorly on unseen data. This occurs when the model is too complex and learns the noise in the training data instead of the underlying patterns.
5. Scalability: For large-scale problems with a high number of features, instances, or model parameters, optimizing the loss function can be computationally expensive and time-consuming. This can limit the applicability of certain optimization techniques or require significant computational resources.



$\beta_0 = \beta_0 - \eta \text{ slope}$





$\frac{dL}{d\beta}$ → small big

new β

(m, b) → 10 thousand → diff

60 billion

Other optimization techniques

21 April 2023 16:39