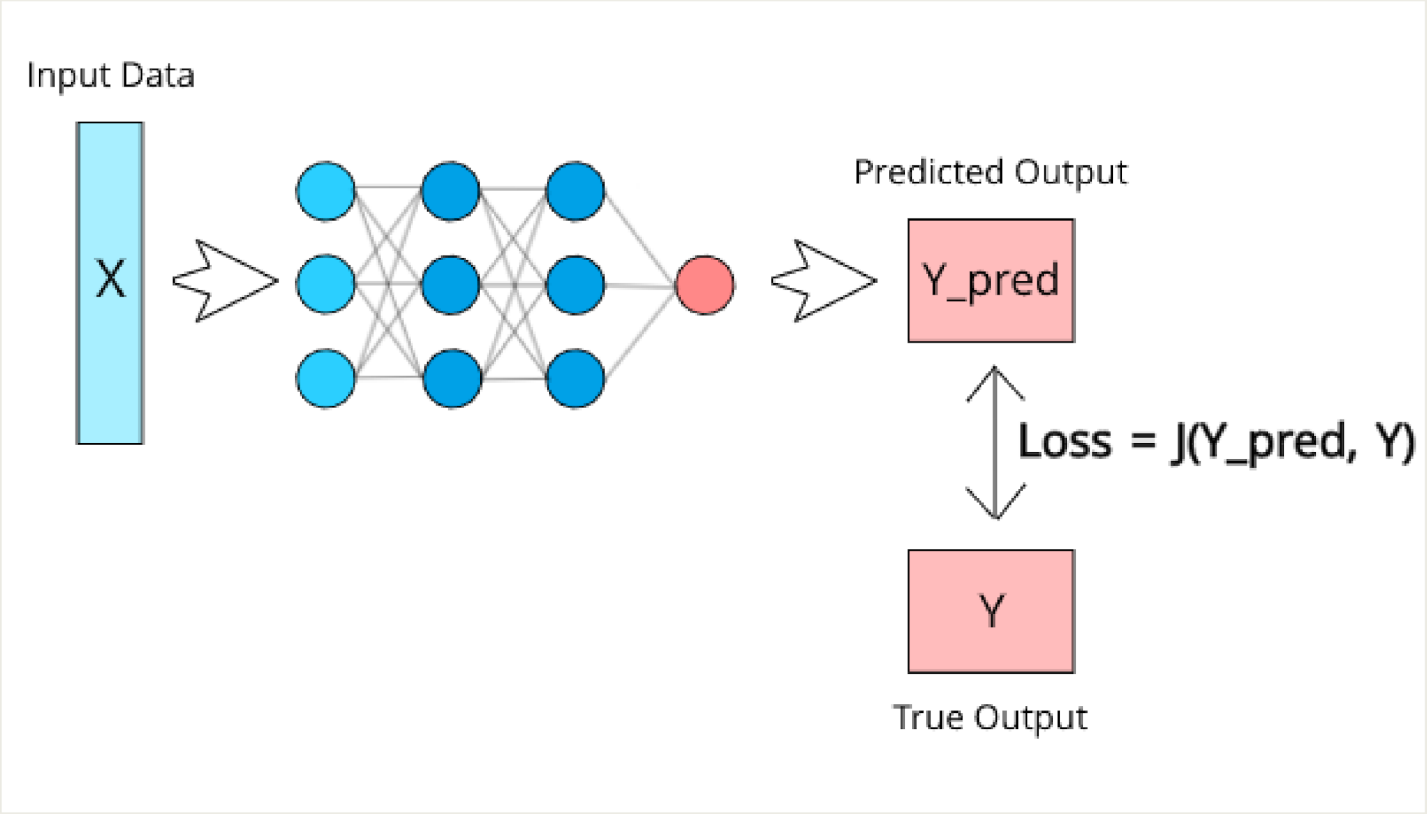
Loss Function



What is a Loss Function

The loss function calculates the error for one output value.

- It shows the difference between the calculated output and the correct value.

$$L(\hat{y}, y)$$

$\bar{y}$: Output value

y : Actual value

- Loss functions are specific to the method and algorithm being applied.

- Different loss functions are applicable to problems like regression, classification etc.

Selecting a loss function for your task requires an understanding of the various functions available, and their characteristics.

Cost Function

While the Loss Function calculates error for one output value, **the Cost Function calculates the error for the entire training dataset.**

$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} \left(L(\hat{y}^i, y^i) \right)$$

with, m = number of training samples, and parameters w : weights, b : biases .

- The cost function J(w,b) represents error of the neural network on the training dataset.

To get the minimum error for the neural network, we need to find the minimum value of cost function J(w,b).

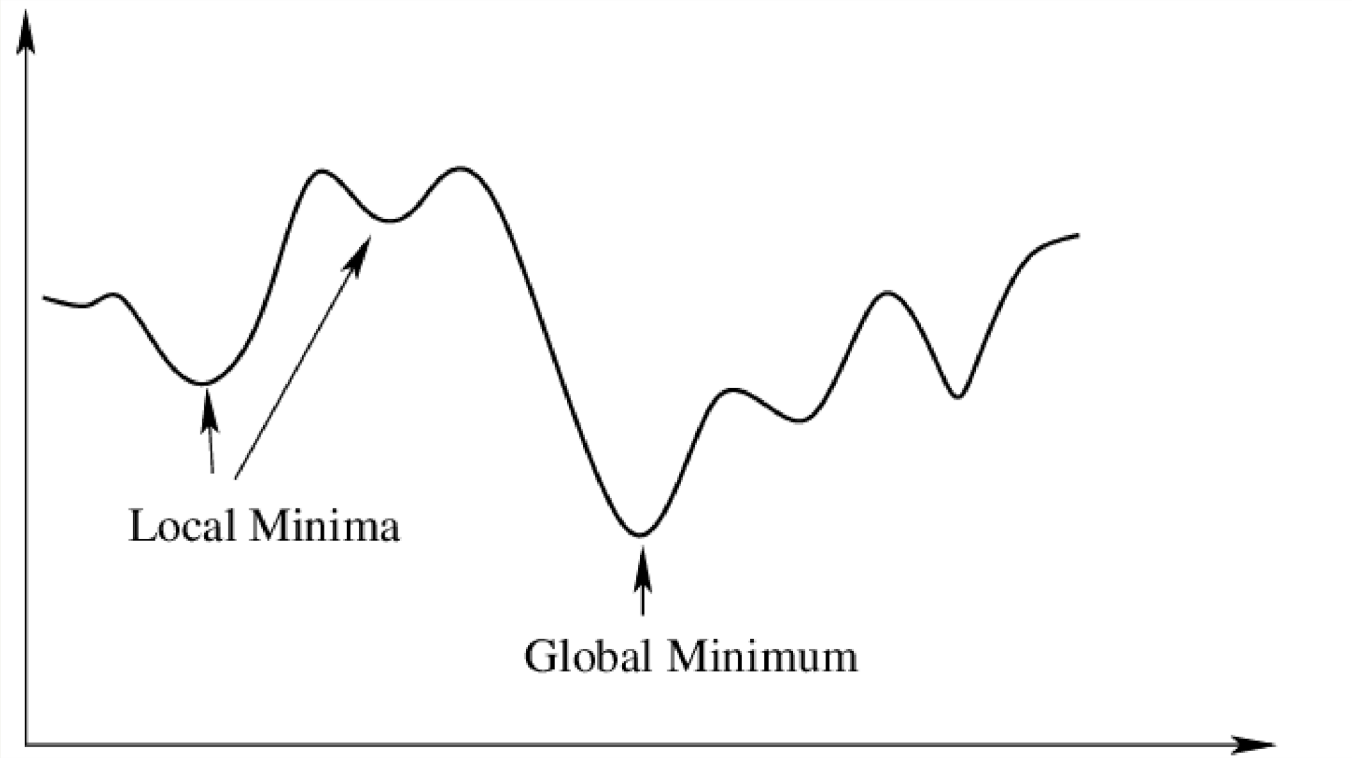
From the minimum value of J(w,b), we can find the **optimum value of w (weights) and b (biases).**

Gradient Descent

Gradient descent is an **optimization method** which works by adjusting model parameters iteratively.

We start with initial values of w and b for J(w,b).

To find the optimum values, we start at the initial values and *take steps in downhill direction*, for one or more iterations, until we converge on the global minima.



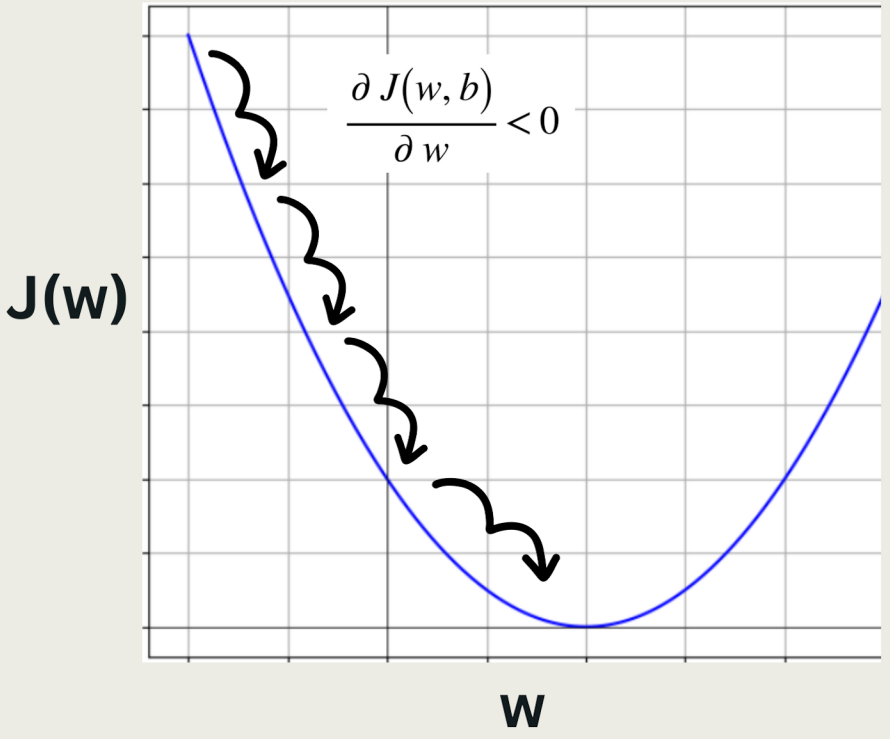
The global minima of the cost function will give us our optimal weights and biases.

Updating the weights

Using partial derivatives, the optimum value of w (weights) is calculated.

$$\partial w = \frac{\partial J(w,b)}{\partial w}$$

$$w = w - \alpha \cdot \frac{\partial (w,b)}{\partial w}$$



The parameter alpha α is the learning rate.

Updating the bias value

Using partial derivatives, the optimum value of b (bias) is calculated similar to the value of w.

$$db = \frac{\partial J(w,b)}{\partial b}$$

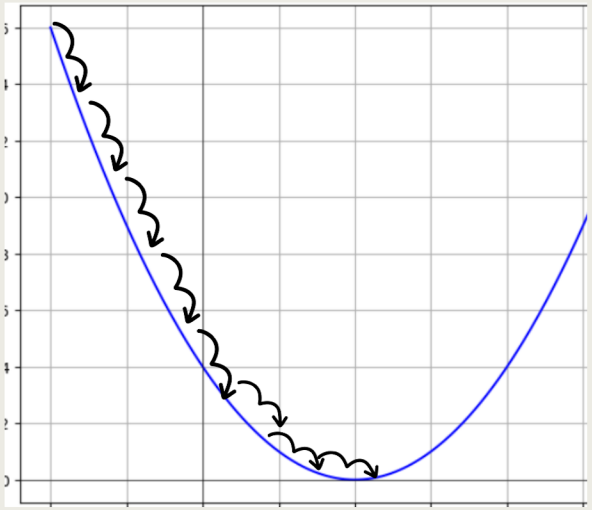
$$b = b - \alpha \cdot \frac{\partial J(w,b)}{\partial b}$$

The learning rate

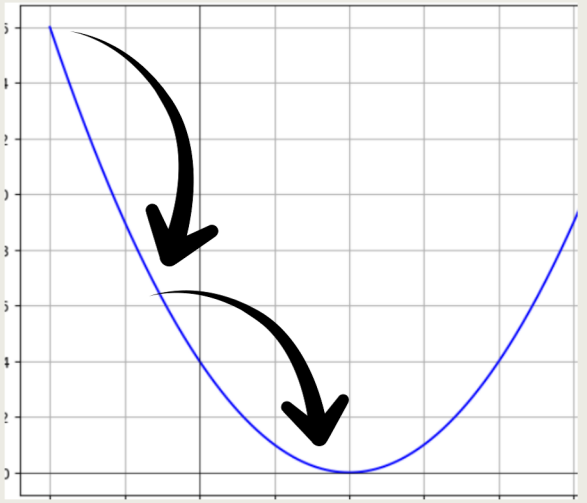
The learning rate (alpha) is a parameter, which defines the number of iterations.

- It directly affects how fast or how slowly the algorithm *converges* to the minimum of the cost function.

small value of α



large value of α



- If it is *too high*, the algorithm might *overshoot* the global minima and cause divergence.

- It is *too low*, the algorithm will make very small steps, making convergence very slow.

The learning rate is either

set to a specific value for the entire working of the neural networks, or

an algorithm is used to adjust its values dynamically.

Thank you !

Join me in understanding data science clearly :

https://www.linkedin.com/in/shivang-kainthola-2835151b9/

https://github.com/HeadHunter28