wHAT IS VANISHING GRADIENT PROBLEM?

SOLUTION-:

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning), the **vanishing gradient problem** is encountered when training [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network" \o "Artificial neural network) with [gradient-based learning methods](https://en.wikipedia.org/wiki/Stochastic_gradient_descent" \o "Stochastic gradient descent) and [backpropagation](https://en.wikipedia.org/wiki/Backpropagation" \o "Backpropagation). In such methods, during each iteration of training each of the neural network's weights receives an update proportional to the [partial derivative](https://en.wikipedia.org/wiki/Partial_derivative" \o "Partial derivative) of the error function with respect to the current weight.[[1]](https://en.wikipedia.org/wiki/Vanishing_gradient_problem" \l "cite_note-Basodi2020-1) The problem is that in some cases, the gradient will be vanishingly small, effectively preventing the weight from changing its value.[[1]](https://en.wikipedia.org/wiki/Vanishing_gradient_problem" \l "cite_note-Basodi2020-1) In the worst case, this may completely stop the neural network from further training.[[1]](https://en.wikipedia.org/wiki/Vanishing_gradient_problem" \l "cite_note-Basodi2020-1)

WHAT IS DYING RELU PROBLEM?

SOLUTIN-:

The dying ReLU problem occurs when several neurons only output a value of zero. This happens primarily when the input is negative. This offers an advantage of network sparsity to ReLU, but it creates a major problem when most of the inputs to the neurons are negative. It basically leads to a worst-case scenario when the entire network dies and only a constant function remains.

When most of the neurons output zero, the gradient fails to flow and the weights stop getting updated. Thus, the network stops learning. As the slope of ReLU activation function is zero in the negative input range, once it becomes dead, it is impossible to recover the network to learn.