Summary

Gradient Descent is an optimization algorithm widely used in machine learning and deep learning. It iteratively adjusts parameters (weights) to minimize a cost function, which measures how far predictions are from actual values. Gradient Descent drives the model to its optimal state by finding the minimum point of the cost function.

Highlights

🔍 Purpose: Gradient Descent minimizes the cost function by adjusting parameters in small steps to make predictions as close to actual values as possible.

⚙️ Cost Function: The cost function (or loss function) measures the error in predictions. Common examples include Mean Squared Error (MSE) for regression and Cross-Entropy for classification.

📈 Learning Rate: The step size taken during each iteration is controlled by the learning rate. A high learning rate may overshoot the minimum, while a low rate can make convergence slow.

Key Insights

🎯 Gradient: The gradient is the slope of the cost function at a given point. It indicates the direction and magnitude of changes to make in the weights to reduce error.

🔁 Iterative Process: Gradient Descent updates the parameters iteratively, gradually moving closer to the minimum cost.

⚖️ Types:

Batch Gradient Descent: Uses the entire dataset to calculate gradients, making it stable but slow for large datasets.

Stochastic Gradient Descent (SGD): Uses one data point at a time, introducing more variability but improving speed.

Mini-batch Gradient Descent: Balances the two, using small batches of data for each iteration.

GitHub Code: https://github.com/NafisAnsari786/Machine-Learning-Algorithms/blob/main/3%20Gradient%20Descent/Gradient%20descent%20with%20formulae.ipynb