Optimizers in Machine Learning

What Are Optimizers?

Optimizers are algorithms used to adjust the parameters of a machine learning model to minimize the loss function. They are essential for training, especially in deep learning, by finding the best model weights.

Why Are Optimizers Important?

1. Efficient Learning
2. Avoiding Local Minima
3. Stability
4. Performance

Common Terminology

- Loss Function: Measures prediction error
- Gradient: Direction of fastest increase in loss
- Learning Rate: Controls weight updates
- Epoch: One full pass through training data

Types of Optimizers

1. Gradient Descent (GD)

Uses the full dataset to compute gradients.

Pros: Accurate

Cons: Slow

2. Stochastic Gradient Descent (SGD)

Uses one sample at a time.

Pros: Fast

Cons: Noisy

Optimizers in Machine Learning

3. Mini-Batch Gradient Descent

Uses small data batches.

Pros: Balanced

Cons: Needs tuning

Advanced Optimizers

4. Momentum

Accelerates updates with prior gradients.

Pros: Escapes local minima

5. RMSProp

Adapts learning rate based on gradient history.

Pros: Good for non-stationary problems

6. Adam

Combines Momentum and RMSProp.

Pros: Fast, adaptive, widely used

7. Nadam

Adam with Nesterov momentum.

Pros: Smooth convergence

Choosing the Right Optimizer

| Optimizer | Best For | Notes |

|-----------|----------|-------|

| SGD | Simplicity | Needs schedule |

| Momentum | Faster | In ravines |

| RMSProp | RNNs | Reduces oscillations |

| Adam | Most use cases | Default choice |

| Nadam | Advanced | Stable convergence |

Challenges in Optimization

- Vanishing/Exploding Gradients
- Overfitting
- Learning Rate Tuning
- Plateaus

Best Practices

- Start with Adam
- Use schedulers
- Early stopping
- Batch norm
- Tune hyperparameters

Conclusion

Optimizers are critical for model training. Understanding different techniques helps build better-performing and faster-learning models.