Optimizers — Short Notes

Optimizers are algorithms that update a neural network's weights using the gradients computed during backpropagation.

They control *how fast* and *how effectively* the model learns.

Why Optimizers Are Needed

- Training is a difficult optimization problem.
- Loss surface has many slopes, flat regions, and noisy gradients.
- Optimizers help models converge faster, avoid getting stuck, and improve stability.

Main Types of Optimizers

1. SGD (Stochastic Gradient Descent)

- Updates weights using mini-batches.
- Fast and simple but sensitive to learning rate.

2. SGD + Momentum

- Adds "inertia" to updates.
- Reduces oscillations and accelerates learning.

3. RMSProp

- Uses moving average of past squared gradients.

- Gives each parameter its own adaptive learning rate.
- Helps with unstable or noisy gradients.

4. Adam (Most Popular)

- Combines Momentum + RMSProp.
- Adaptive learning rates + smooth updates.
- Fast, stable, and works well for most deep learning tasks.

5. AdamW

- Adam with proper weight decay.
- Better generalization, used in modern models (Transformers, large CNNs).

Summary

Optimizers decide how weights change during training. SGD is simple, Momentum stabilizes it, RMSProp adapts learning rates, and Adam/AdamW give fast and reliable convergence.