Mini-batch Gradient Descent

Batch GD

θ = θ - η∇L(θ)

Batch Size: N (all data)

✓ Stable convergence
✓ Accurate gradient
✗ Slow updates
✗ Memory intensive
✗ May get stuck

Mini-batch GD ⭐

θ = θ - η∇L_B(θ)

Batch Size: B (32, 64, 128...)

✓ Balanced approach
✓ Efficient GPU use
✓ Good convergence
✓ Escapes local minima
✓ Industry standard

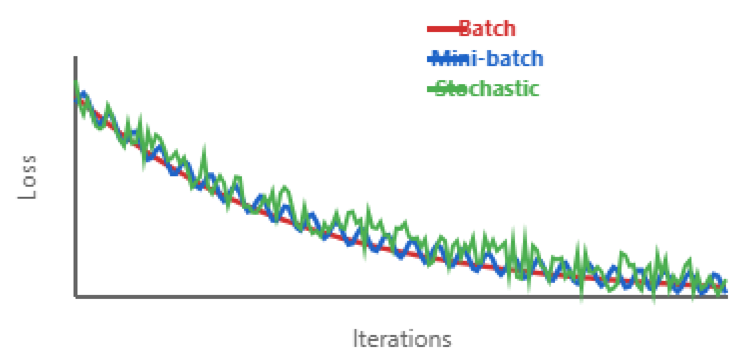
Stochastic GD

θ = θ - η∇L_i(θ)

Batch Size: 1 (single sample)

✓ Fast updates
✓ Low memory
✓ Online learning
✗ Noisy gradients
✗ Unstable convergence

Convergence Behavior



Key Advantages

Efficient: Balances computation and memory

Robust: Noise helps escape poor local minima

Scalable: Works well with large datasets

Hardware: Optimizes GPU parallelization

Common Batch Sizes

Small datasets: `32, 64`

Medium datasets: `128, 256`

Large datasets: `512, 1024`

Training Process

1. Shuffle data
2. Split into mini-batches
3. Update for each batch
4. Repeat for epochs

Best Practice: Mini-batch GD combines the stability of batch GD with the speed of SGD, making it the default choice for training modern neural networks. The batch size is a critical hyperparameter affecting convergence and generalization.