Task_4

Types of Gradient Descent

1. Batch Gradient Descent

Definition: Computes the gradient using the entire dataset before updating parameters.
Pros:

- More stable convergence.
- Moves directly toward the minimum.

Cons:

- Slow for large datasets.
- High memory usage.

2. Stochastic Gradient Descent (SGD)

Definition: Updates parameters after computing the gradient for each training example.
Pros:

- Faster updates and can handle large datasets.
- Escapes local minima due to noise in updates.

Cons:

- More variance in updates, making convergence noisy.
- May not converge to the exact minimum.

3. Mini-Batch Gradient Descent

Definition: Uses a small subset (mini-batch) of data to compute the gradient and update parameters.
Pros:

- Balances stability (like Batch) and speed (like SGD).
- Efficient for deep learning tasks.

Cons:

- Requires tuning batch size for optimal performance.
- Still has some noise in updates.

4. Momentum-based Gradient Descent

Definition: Uses past gradients to accelerate convergence and reduce oscillations.
Pros:

- Helps overcome saddle points.

- Faster convergence than standard gradient descent.

Cons:

- Requires careful tuning of the momentum parameter.
- Can overshoot the optimal point.

5. Adam (Adaptive Moment Estimation)

Definition: Uses adaptive learning rates for each parameter by computing both first and second moments of gradients.

Pros:

- Works well with noisy gradients and sparse data.
- Adaptive learning rates improves performance.

Cons:

- Requires more memory.
- May not always converge optimally.

Types of Optimizers

1. Momentum Optimizer

Definition: Uses an exponentially weighted moving average of past gradients to smooth updates and accelerate convergence.

Pros:

- Helps escape local minima.
- Faster convergence than vanilla gradient descent.

Cons:

- Requires tuning the momentum hyperparameter.
- Can overshoot the optimal point.

2. AdaGrad (Adaptive Gradient Algorithm)

Definition: Adapts the learning rate for each parameter individually based on past gradients, giving smaller updates to frequently updated parameters.

Pros:

- Suitable for sparse data.
- Reduces the need for manual learning rate tuning.

Cons:

- Learning rate decreases too aggressively, leading to premature stopping.
- May not generalize well to non-convex problems.

3. RMSprop (Root Mean Square Propagation)

Definition: An adaptive learning rate method that normalizes the gradient using a moving average of squared past gradients.

Pros:

- Works well for non-stationary data.
- Solves AdaGrad’s aggressive learning rate decay issue.

Cons:

- Still requires careful tuning of the learning rate.
- Can be sensitive to hyperparameter choices.