Report on

“Analysis on three different gradient descent algorithms and its performance.”

Introduction

This report evaluates the performance of three gradient descent algorithms-Gradient Descent (GD), Stochastic Gradient Descent (SGD), and Adam Optimizer used to train a basic Artificial Neural Network (ANN) for predicting California house prices. The ANN has three layers: an input layer with one neuron, a hidden layer with 10 neurons (using the sigmoid activation function), and an output layer with one neuron.

We assess each algorithm's convergence speed, accuracy, and computational efficiency.

Overview of Gradient Descent Algorithms

1. **Gradient Descent (GD)**: Updates model parameters using the entire training dataset in each iteration. It has stable but slower convergence due to its reliance on the full dataset.
   a. **Learning Rate**: 0.01
   b. **Batch Size**: Entire training dataset
2. **Stochastic Gradient Descent (SGD)**: Updates parameters for each training example individually, introducing noise that helps escape local minima and converge faster but less stably.
   a. **Learning Rate**: 0.01
   b. **Batch Size**: 1
3. **Adam Optimizer**: Uses adaptive learning rates and combines the benefits of RMSProp and Momentum, leading to faster and more stable convergence.
   a. **Learning Rate**: 0.01
   b. **Batch Size**: Entire training dataset

Performance Evaluation

- **Actual vs. Predicted Prices**:

   a. **GD**: Predicted values align with actual prices but show slower response to fluctuations.
   b. **SGD**: Predictions have higher variance around actual prices due to its stochastic nature.
   c. **Adam**: Predicted values closely match actual prices with minimal deviation, indicating superior convergence.
- **Learning Curves**:
   a. **GD**: Loss decreases steadily but slowly across epochs.
   b. **SGD**: Loss shows fluctuations due to noisy updates but generally trends toward convergence.
   c. **Adam**: Loss decreases rapidly and stabilizes faster, showing quick convergence.
- **Weights and Cost During Iterations**:
   a. **GD**: Weights move toward optimal values slowly but consistently.
   b. **SGD**: Weights exhibit erratic movements due to high variance in updates.
   c. **Adam**: Weights adjust quickly and more effectively toward optimal values.

Analysis and Comparison

1. **Convergence Speed**:
   a. **Fastest**: Adam Optimizer
   b. **Moderate**: SGD
   c. **Slowest**: GD
2. **Stability**:
   a. **Most Stable**: GD
   b. **Balanced**: Adam
   c. **Least Stable**: SGD
3. **Accuracy**:
   a. **Highest**: Adam
   b. **Moderate**: GD
   c. **Lowest**: SGD
4. **Computational Efficiency**:
   a. **Most Efficient per Iteration**: SGD
   b. **Balanced**: Adam
   c. **Least Efficient per Iteration**: GD

Conclusion

1. **Adam Optimizer** is the best performer overall, with the fastest convergence, high stability, and accuracy. It is ideal for large datasets and complex models.
2. **SGD** is useful when computational resources are limited or in online learning tasks, but it may require careful tuning of hyperparameters.
3. **GD** is reliable for small datasets where stability is crucial, but its slow convergence makes it less suitable for large-scale problems.