Algorithm	Description	Use Cases	Advantages	Disadvantages
Stochastic Gradient Descent (SGD)	Updates parameters using the gradient of loss computed on a single training example or a small subset.	Large datasets, online learning, non-convex optimization	Computationally efficient, easy to implement	High variance, noisy updates, slower convergence
Stochastic Gradient Descent using Momentum (SGDM)	Adds a momentum term to standard SGD to smooth optimization trajectory and accelerate convergence.	Large datasets, noisy or sparse gradients	Accelerates convergence, stabilizes optimization trajectory	Hyperparameter tuning, potential overshooting
Gradient Descent	Updates parameters using the gradient of loss computed on the entire training dataset.	Small to medium-sized datasets, convex optimization	Guaranteed convergence to global minimum (in convex cases)	Computationally expensive for large datasets
AdaGrad	Adapts the learning rate for each parameter based on the historical sum of squared gradients.	Sparse data, features with different frequencies	Adaptive learning rates, effective for sparse data	Diminishing learning rates, accumulation of squared gradients
RMSProp	Adapts the learning rate for each parameter based on the exponentially decaying average of past squared gradients.	Deep neural networks, non-stationary environments	Adaptive learning rates, stability, efficiency	Hyperparameter tuning, memory requirement
Adam	Combines momentum optimization and adaptive learning rate methods by estimating first and second moments of gradients.	Deep neural networks, noisy or sparse gradients	Adaptive learning rates, efficiency, stability	Hyperparameter tuning, memory requirement, convergence to sharp minima