optimization techniques use in machine learning

optimization techniques in machine learning are used to find the best values for the model's hyperparameters. Hyperparameters are the parameters that control the learning process, such as the learning rate, the number of layers, and the number of neurons in each layer.

There are a number of different optimization techniques that can be used, including:

1.**Gradient Descent**:

Gradient descent is a popular optimization technique that involves iteratively adjusting the model parameters in the direction of the steepest descent of the loss function. This technique is used in many machine learning algorithms, including linear regression, logistic regression, and neural networks.

2.**Stochastic Gradient Descent (SGD)** :

It is a training models that involve large datasets ,It is a variant of gradient descent that updates the model parameters based on a random subset of the training data instead of using the entire dataset for each iteration, only a single random training example (or a small batch) is selected to calculate the gradient and update the model parameters.

SGD is generally noisier than typical Gradient Descent.

it usually took a higher number of iterations to reach the minima, because of the randomness in its descent. Even though it requires a higher number of iterations to reach the minima than typical Gradient Descent, it is still computationally much less expensive than typical Gradient Descent. Hence, in most scenarios, SGD is preferred over Batch Gradient Descent for optimizing a learning algorithm.

the computational cost per iteration is significantly reduced compared to traditional Gradient Descent methods that require processing the entire dataset.

3. Mini-batch Gradient Descent :

updates the model parameters on a small subset, or mini-batch, of the training data at a time, rather than on a single example (as in SGD) or on the entire dataset (as in batch gradient descent).

We Initialize the model parameters with random values then Divide the training dataset into small subsets, or mini-batches, of size m , For each mini-batch, compute the gradient of the loss function with respect to the model parameters on the selected mini-batch.

The advantage of mini-batch gradient descent over SGD and batch gradient descent is that it can balance the benefits of stochastic and batch gradient descent. By updating the model parameters on a small subset of training examples at a time, mini-batch gradient descent can reduce the variance of the parameter updates and achieve better convergence speed and stability than SGD. At the same time, mini-batch gradient descent can still take advantage of vectorized operations and parallel processing to improve the computational efficiency compared to batch gradient descent.

4. **Adagrad optimization (Adaptive Gradient Descent):**

in Adagrad Optimizer the core idea is that each weight has a different learning rate (η).

some of the advantages :

- It is simple and efficient.
- It can achieve fast convergence on a variety of problems.
- It is relatively insensitive to the choice of hyperparameters.

Here are some of the disadvantages:

- It can be sensitive to noise in the gradients.
- It can be slow to converge on problems with sparse gradients.

5. **Adam Optimizer(Adaptive Moment Estimation):**

combines the benefits of momentum and adaptive learning rates. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm**.**

1-Momentum: This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the 'exponentially weighted average' of the gradients. Using averages makes the algorithm converge towards the minima in a faster pace.

2-Root Mean Square Propagation (RMSP): Root mean square prop or RMSprop is an adaptive learning algorithm that tries to improve AdaGrad. Instead of taking the cumulative sum of squared gradients like in AdaGrad, it takes the 'exponential moving average'.

Adam Optimizer inherits the strengths or the positive attributes of the above two methods and builds upon them to give a more optimized gradient descent.

Here are some of the benefits of using optimization techniques in machine learning :

- Improved accuracy: Optimization techniques can help to improve the accuracy of machine learning models. This is because they can help the model to find the best values for its hyperparameters, which can lead to better predictions.
- Increased speed: Optimization techniques can also help to increase the speed of machine learning models. This is because they can help the model to converge more quickly, which means that it can learn from the data more quickly.
- Reduced overfitting: Optimization techniques can help to reduce overfitting in machine learning models. Overfitting occurs when the model learns the training data too well and is not able to generalize to new data. Optimization techniques can help to prevent this by encouraging the model to learn more general features from the data.