Mini-Batch GD

Mini-batch gradient descent is a variation of the gradient descent algorithm which splits the training dataset into small batches and uses each batch to update the model parameters. This approach combines the advantages of both stochastic gradient descent and batch gradient descent.

Here are the steps involved in mini-batch gradient descent (mini-batch GD):

1. **Iterate Through Mini-batches:**

- For each mini-batch within the current epoch:
  - **Forward Pass:**
    - Feed the data points in the mini-batch through the model to compute predictions.
  - **Error Calculation:**
    - Calculate the error between the model's predictions and the actual target values for each data point in the mini-batch.
  - **Gradient Calculation:**
    - Calculate the gradient of the loss function (typically mean squared error in linear regression) with respect to the model's parameters (weights and biases) using the errors from the mini-batch.
  - **Weight Update:**
    - Update the model's parameters (weights and biases) in the direction opposite to the average gradient calculated from the mini-batch, scaled by the learning rate.

Advantages of Mini Batch Gradient Descent:

1. **Greater Accuracy than Stochastic Gradient Descent (SGD)**:
    - Mini-batch gradient descent strikes a balance between the noisy updates of SGD and the slower convergence of batch gradient descent.
    - It computes gradients using a small subset (mini-batch) of the training data, resulting in more accurate updates compared to pure stochastic updates.
    - The mini-batch size can be adjusted to control the trade-off between accuracy and computational efficiency.

2. **Efficient Handling of Large Datasets**:
    - Mini-batch gradient descent efficiently processes large datasets.
    - By breaking the data into smaller batches, it updates model parameters more frequently, leading to faster convergence.
    - It avoids the memory limitations associated with batch gradient descent, which requires loading the entire dataset into memory.

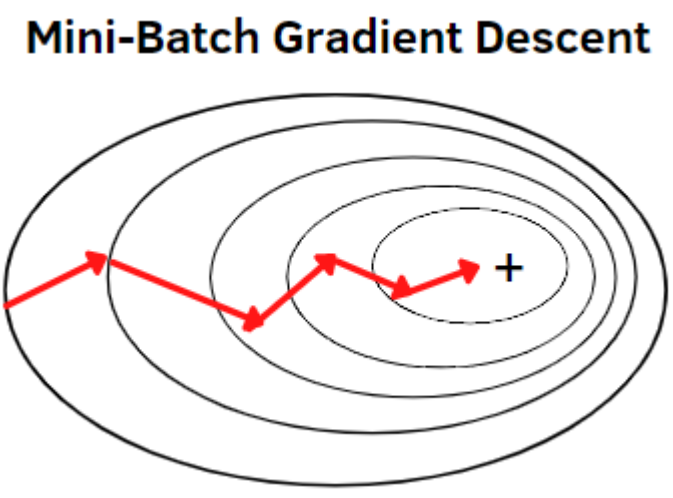
3. **Escape from Local Minima**:
    - Mini-batch updates allow the algorithm to escape local minima more effectively than pure batch gradient descent.
    - The noise introduced by mini-batch updates helps explore different regions of the loss surface, potentially leading to better global minima.

Disadvantages of Mini Batch Gradient Descent:

1. **Complex Algorithm**:
    - Mini-batch gradient descent introduces additional complexity compared to pure batch or stochastic gradient descent.
    - Hyperparameters such as the mini-batch size need to be tuned, which can be challenging.

2. **Additional Hyperparameter**:
    - In addition to the learning rate, mini-batch gradient descent requires tuning the mini-batch size.
    - Choosing an appropriate mini-batch size depends on the problem and available computational resources.

3. **Less Accurate than Batch Gradient Descent**:
    - While mini-batch gradient descent improves efficiency, it sacrifices some accuracy compared to batch gradient descent.
    - The noise introduced by mini-batch updates can lead to suboptimal convergence.



Differences between BGD, SGD, and MBGD

Batch Gradient Descent (BGD):

- Processes the entire training dataset for each update step in the gradient descent algorithm.
- Advantages:
  - Can ensure convergence to a minimum loss value (if the loss function is convex).
  - Less prone to getting stuck in local minima.
- Disadvantages:
  - Computational cost can be high, especially for very large datasets.
  - May require a lot of memory to store the entire dataset.

Stochastic Gradient Descent (SGD):

- Processes a single data point (mini-batch size of 1) for each update step.
- Advantages:
  - Faster than BGD for large datasets.
  - Requires less memory to hold a single data point at a time.
- Disadvantages:
  - Updates can be noisy due to using only one data point.
  - May converge to a less optimal solution due to the noisy updates.

Mini-batch Gradient Descent:

- Processes a small subset (mini-batch) of the training dataset in each update step. This mini-batch size is typically a hyperparameter you can tune.
- Advantages:
  - Faster than BGD, especially for large datasets.
  - Less noisy updates compared to SGD due to using a small group of data points.
  - Can potentially converge to a better solution than SGD due to averaging the gradients across the mini-batch.
  - Requires less memory compared to storing the entire dataset.
- Disadvantages:
  - May not guarantee convergence to a global minimum (like BGD) if the loss function is non-convex.