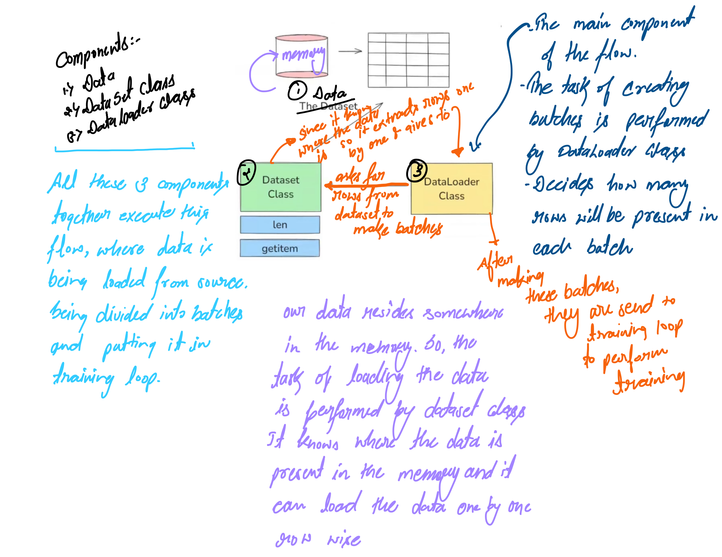
6.1 Dataset & DataLoader CLASS IN PyTorch

03 September 2025 12:14 AM Avinash Yadav

Dataset and **DataLoader** are core abstractions in PyTorch that **decouple how you define your data from how you efficiently iterate over it in training loops.**



1. Dataset Class

The Dataset class is essentially a blueprint. When you create a custom Dataset, you decide how data is loaded and returned.

```
CustomDataset(Dataset):
    __init__():
        # from where the data should be loaded
        # ex. csv, folder containing images etc.

    __len__():
        # returns length/no of rows presents
        # in our dataset → with the help of
        # this we can determine how many
        # batches will be made i.e. len/batch_size

    __getitem__(index):
        # returns a particular row based on
        # the provided index.
```

It defines:

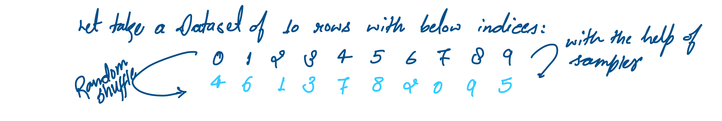
- *__init__()* which tells how data should be loaded.
- *__len__()* which returns the total number of samples.
- *__getitem__(index)* which returns the data (and label) at the given index.

2. DataLoader Class

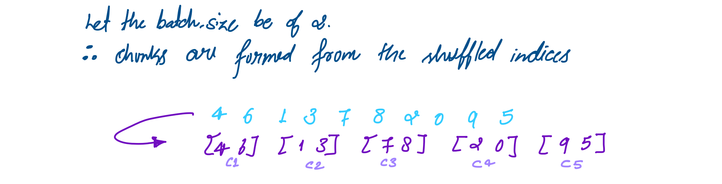
The DataLoader **wraps a Dataset** and **handles batching, shuffling, and parallel loading** for you.

DataLoader Control Flow:

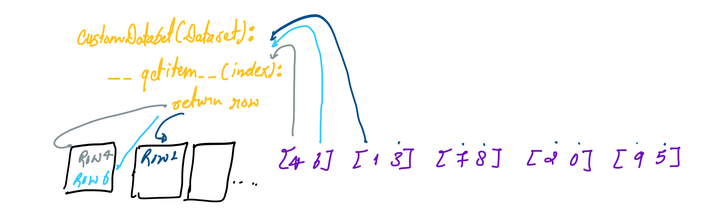
- **At the start of each epoch, the DataLoader (if *shuffle=True*) shuffles indices(using a sampler).**



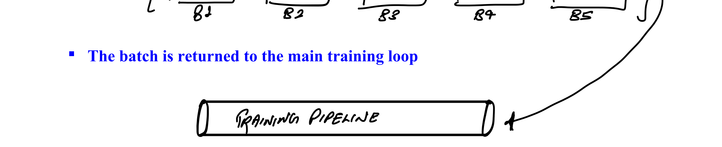
- **It divides the indices into chunks of *batch_size*.**



- **For each index in the chunk, data samples are fetched from the Dataset object**



- **The samples are then collected and combined into a batch (using *collate_fn*)**



- **The batch is returned to the main training loop**