

# 6.1 Dataset & DataLoader CLASS IN PyTorch

03 September 2025

12:14 AM

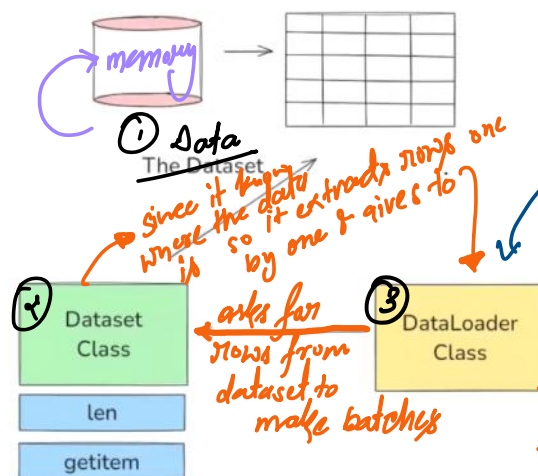
Avinash Yadav

**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.

Components:-

- 1. Data
- 2. Data Set Class
- 3. Data loader Class

All these 3 components together execute this flow, where data is being loaded from source, being divided into batches and putting it in training loop.



- The main component of the flow.
  - The task of creating batches is performed by DataLoader class
  - Decides how many items will be present in each batch
- After making these batches, they are sent to training loop to perform training

Our data resides somewhere in the memory. So, the task of loading the data is performed by dataset class. It knows where the data is present in the memory and it can load the data one by one row wise.

## 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\_\_():

// 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).

Let take a Dataset of 10 rows with below indices:

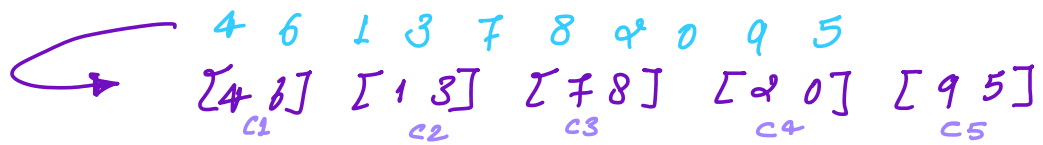
	0	1	2	3	4	5	6	7	8	9
Random shuffle	4	6	1	3	7	8	2	0	9	5

with the help of sampler

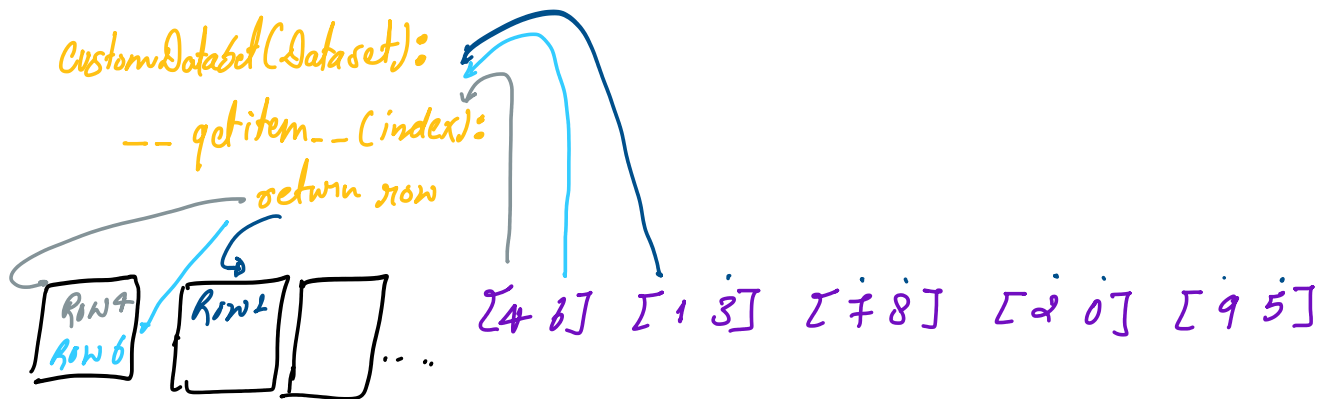
- It divides the indices into chunks of `batch_size`.

Let the batch-size be of 2.

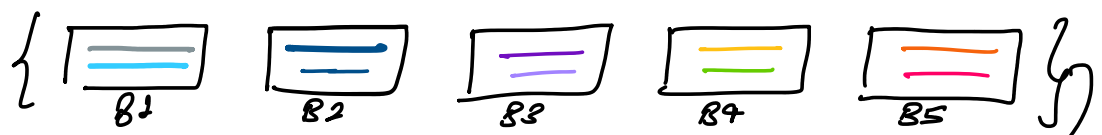
∴ chunks are formed from the shuffled indices



- 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

