PyTorch

# DataLoader

torch.utils.data.DataLoader(dataset, batch\_size=1, shuffle=False, sampler=None,

batch\_sampler=None, num\_workers=0, collate\_fn=None,

pin\_memory=False, drop\_last=False, timeout=0,

worker\_init\_fn=None, \*, prefetch\_factor=2,

persistent\_workers=False)

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The DataLoader supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

**Parameters**

dataset (Dataset) – dataset from which to load the data.

batch\_size (int, optional) – how many samples per batch to load (default: 1).

shuffle (bool, optional) – set to True to have the data reshuffled at every epoch (default: False).

sampler (Sampler or Iterable, optional) – defines the strategy to draw samples from the dataset. Can be any Iterable with \_\_len\_\_ implemented. If specified, shuffle must not be specified.

batch\_sampler (Sampler or Iterable, optional) – like sampler, but returns a batch of indices at a time. Mutually exclusive with batch\_size, shuffle, sampler, and drop\_last.

num\_workers (int, optional) – how many subprocesses to use for data loading. 0 means that the data will be loaded in the main process. (default: 0)

collate\_fn (callable, optional) – merges a list of samples to form a mini-batch of Tensor(s). Used when using batched loading from a map-style dataset.

pin\_memory (bool, optional) – If True, the data loader will copy Tensors into CUDA pinned memory before returning them. If your data elements are a custom type, or your collate\_fn returns a batch that is a custom type, see the example below.

drop\_last (bool, optional) – set to True to drop the last incomplete batch, if the dataset size is not divisible by the batch size. If False and the size of dataset is not divisible by the batch size, then the last batch will be smaller. (default: False)

timeout (numeric, optional) – if positive, the timeout value for collecting a batch from workers. Should always be non-negative. (default: 0)

worker\_init\_fn (callable, optional) – If not None, this will be called on each worker subprocess with the worker id (an int in [0, num\_workers - 1]) as input, after seeding and before data loading. (default: None)

generator (torch.Generator, optional) – If not None, this RNG will be used by RandomSampler to generate random indexes and multiprocessing to generate base\_seed for workers. (default: None)

prefetch\_factor (int, optional, keyword-only arg) – Number of samples loaded in advance by each worker. 2 means there will be a total of 2 \* num\_workers samples prefetched across all workers. (default: 2)

persistent\_workers (bool, optional) – If True, the data loader will not shutdown the worker processes after a dataset has been consumed once. This allows to maintain the workers Dataset instances alive. (default: False)

# Dataset

## Map-style

torch.utils.data.Dataset(\*args, \*\*kwds)

An abstract class representing a Dataset.

All datasets that represent a map from keys to data samples should subclass it. All subclasses should overwrite **\_\_getitem\_\_()**, supporting fetching a data sample for a given key. Subclasses could also optionally overwrite **\_\_len\_\_()**, which is expected to return the size of the dataset by many Sampler implementations and the default options of DataLoader.

## Iterable-style