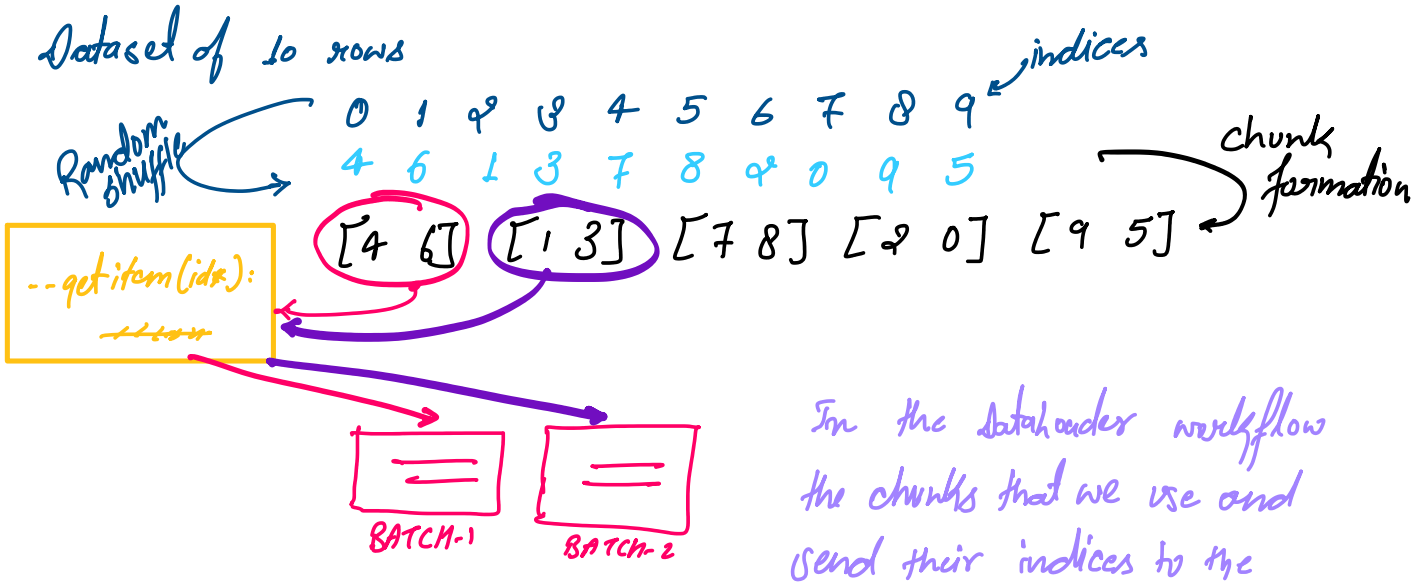
6.6 NOTE ABOUT `collate_fn`

03 September 2025 03:08 AM

Avinash Yadav



In the dataloader workflow the chunks that we use and send their indices to the --getitem-- and in return we get the data of rows and then data of rows are combined to formed batches and all this task of combining the rows is performed by 'collate_fn'

The ***collate_fn*** in PyTorch's ***DataLoader*** is a function that ***specifies how to combine a list of samples from a dataset into a single batch.***

By default, the ***DataLoader*** uses a simple batch collation mechanism, but ***collate_fn*** allows you to customize how the data should be processed and batched.

So when do we need customization to collate_fn:

Suppose we have a textual dataset as given below

Here we have 4 rows and let the batch size be of 2 that means in each batch we'll have 2 rows or 2 sentences.

chunk 1

chunk 1

	Sentence	Tokenized(Integer IDs)	Label
R1	"I Love Coding"	[1, 2, 3]	0
R2	"Deep Learning Rocks"	[4, 5, 6]	1
R3	"Transformers Are Fun"	[7, 8, 9, 10]	1
R4	"Hello World"	[11, 12]	0

Chunk 2

suppose the chunk1 is formed with R1 & R2. So we can simply merge R1 and R2 very easily because they are of same size

But if we look in the chunk2. and we go for stacking those 2 tensors, we see they are of different size. so they won't stack easily because of different sizes.

Therefore the solution to this problem of merging of chunks of different sizes will be achieved by writing a manual code to perform padding

Therefore we can add zeroes before or after to make all the tensors of equal size and then we can perform stacking/ merging to form the batch.

Sentence	Tokenized(Integer IDs)	Label
"I Love Coding"	[1, 2, 3]	0
"Deep Learning Rocks"	[4, 5, 6]	1
"Transformers Are Fun"	[7, 8, 9, 10]	1
"Hello World"	[**0**, **0**, 11, 12]	0

	or [11, 12, **0**, **0**]	

Therefore the logic of padding is written inside the 'collate_fn'