Data Classes (Pytorch, FastAI)

Pytorch Dataset - A collection of tuples of X(independent features) and Y(dependent or target) for each item. Say there are 100 such tuples.

Pytorch DataLoader – Pass the above dataset and a batch size to Dataloader. Dataloader returns an iterator of multiple batches of tuples of X and Y.

For the above dataset and batch size =16, DataLoader will return 7 batches of 16 tuples each in the first 6 batches and remaining 4 tuples in the 7th batch. (16\*6=96 and remaining 4)

FastAI Datasets – A training set collection and a validation set collection of tuples of X(independent features) and Y(dependent or target) for each item. This is an iterator as well.

FastAI DataLoaders – This is a FastAI class which can take multiple Pytorch Dataloaders as input. Generally a training pytorch dataloader and validation pytorch dataloader are passed as inputs to this class.

# Fast AI DataBlock – This API is used to create the FastAI Dataloaders. Below is an example.

“dls” is the dataloaders class that we created using the Datablock API.

* This is like a template/blueprint which we defined first.
* The bold text is where we create the FastAI Dataloaders using the blueprint defined and passing a dataframe as input.
* splitter\_function is a function we are passing as input for splitting the dataframe into train and valid.
* get\_x\_function and get\_y\_function are functions we pass as inputs for getting the X and Y from the dataframe.
* [Fast AI Docs](https://docs.fast.ai/data.block) has more details.
* show\_batch() can be used to see if the Datablock that we defined is correct. For example, if we missed the Item Transformation of RandomResizeCrop and the input images are of different sizes, then the show\_batch() is going to fail and provide a meaningful message.

dblock = DataBlock(blocks=(ImageBlock, MultiCategoryBlock),

splitter=splitter\_funtion,

get\_x=get\_x\_function,

get\_y=get\_y\_funtion,

item\_tfms = RandomResizedCrop(128, min\_scale=0.35))

**dls = dblock.dataloaders(df)**

**dls.show\_batch(nrows=2, ncols=3)**