

Datasets and Dataloaders

Datasets and Dataloaders

Introduction

- Model training ideally should be separated from data preprocessing
 - Better readability and modularity
- Dataset and Dataloader
 - Interface to Pre-loaded datasets
 - Interface to custom datasets
- **Dataset**
 - Stores samples and labels
- **Dataloader**
 - Iterable wrapped around Dataset

Datasets and Dataloaders

Custom Dataset

- Requires three function implementations:
 - `__init__`
 - Runs once during instantiating the object
 - `__len__`
 - Returns number of samples
 - `__getitem__`
 - Loads samples from dataset, pre-processes them and returns them for given index

```
from torch.utils.data import Dataset, DataLoader
```

Run Cell | Run Above | Debug Cell

Dataset and DataLoader

```
class LinearRegressionDataset(Dataset):  
    def __init__(self, X, y):  
        self.X = X  
        self.y = y  
  
    def __len__(self):  
        return len(self.X)  
  
    def __getitem__(self, idx):  
        return self.X[idx], self.y[idx]
```

Datasets and Dataloaders

Dataloader

- Dataloader iterates through dataset
- Iterations return batches of data
- Features
 - allows for shuffling the data
 - custom sampling strategies

```
train_loader = DataLoader(dataset = LinearRegressionDataset  
(X_np, y_np), batch_size=2)
```