# PyTorch Datasets and Data Loaders

**Lesson** Downloads

**Show TranscriptSummarize Video**

PyTorch makes accessing data for your model a breeze! These tools ensure that the flow of information to our AI is just right, making its learning experience effective and fun.

Technical Terms:

**PyTorch Dataset class:** This is like a recipe that tells your computer how to get the data it needs to learn from, including where to find it and how to parse it, if necessary.

**PyTorch Data Loader:** Think of this as a delivery truck that brings the data to your AI in small, manageable loads called batches; this makes it easier for the AI to process and learn from the data.

**Batches:** Batches are small, evenly divided parts of data that the AI looks at and learns from each step of the way.

**Shuffle:** It means mixing up the data so that it's not in the same order every time, which helps the AI learn better.

Quiz Question

If you have a batch size of three, and your dataset has 11 items, how many batches will the Data Loader produce, and what will be the size of the last batch?

- It will produce 3 batches, with the last batch containing 3 items.
- It will produce 5 batches, with the last batch containing 1 item.
- It will produce 4 batches, with the last batch containing 2 items.

**Submit**

Code Examples

Datasets

```python
from torch.utils.data import Dataset

# Create a toy dataset
class NumberProductDataset(Dataset):
    def __init__(self, data_range=(1, 10)):
        self.numbers = list(range(data_range[0], data_range[1]))

    def __getitem__(self, index):
        number1 = self.numbers[index]
        number2 = self.numbers[index] + 1
        return (number1, number2), number1 * number2

    def __len__(self):
        return len(self.numbers)

# Instantiate the dataset
dataset = NumberProductDataset(
    data_range=(0, 11)
)

# Access a data sample
data_sample = dataset[3]
print(data_sample)
# ((3, 4), 12)
```

Data Loaders

```
from torch.utils.data import DataLoader

# Instantiate the dataset
dataset = NumberProductDataset(data_range=(0, 5))

# Create a DataLoader instance
dataloader = DataLoader(dataset, batch_size=3, shuffle=True)

# Iterating over batches
for (num_pairs, products) in dataloader:
    print(num_pairs, products)
# [tensor([4, 3, 1]), tensor([5, 4, 2])] tensor([20, 12, 2])
# [tensor([2, 0]), tensor([3, 1])] tensor([6, 0])
```

Resources

**PyTorch Dataset documentation**
**PyTorch DataLoader documentation**
**Index of PyTorch data utilities**