

# 深度学习快速入门

05Dataloader 的使用

POET

2024 年 2 月 11 日

## 1 Dataloader 使用

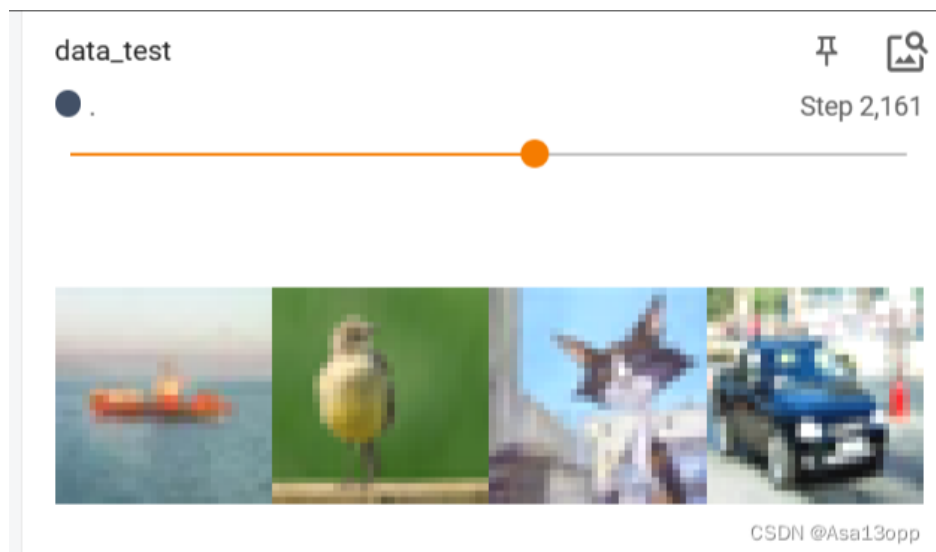
- Dataset 只是去告诉我们程序，我们的数据集在什么位置，数据集第一个数据给它一个索引 0，它对应的是哪一个数据。
- Dataloader 就是把数据加载到神经网络当中，Dataloader 所做的事就是每次从 Dataset 中取数据，至于怎么取，是由 Dataloader 中的参数决定的。

```
1 import torchvision
2 from torch.utils.data import DataLoader
3
4 # 准备的测试数据集
5 test_data = torchvision.datasets.CIFAR10("./
    dataset", train=False, transform=torchvision.
    transforms.ToTensor())
6 img, target = test_data[0]
7 print(img.shape)
8 print(img)
9
10 # batch_size=4 使得 img0, target0 = dataset[0]、
    img1, target1 = dataset[1]、img2, target2 =
    dataset[2]、img3, target3 = dataset[3]，然后这
    四个数据作为Dataloader的一个返回
11 test_loader = DataLoader(dataset=test_data,
    batch_size=4, shuffle=True, num_workers=0,
    drop_last=False)
12 # 用for循环取出DataLoader打包好的四个数据
13 for data in test_loader:
14     imgs, targets = data # 每个data都是由4张图片组
        成，imgs.size 为 [4,3,32,32]，四张32×32图片
        三通道，targets由四个标签组成
```

```
15     print(imgs.shape)
16     print(targets)
```

## 2 Tensorboard 展示

```
1     import torchvision
2     from torch.utils.data import DataLoader
3     from torch.utils.tensorboard import SummaryWriter
4
5     # 准备的测试数据集
6     test_data = torchvision.datasets.CIFAR10("./
        dataset", train=False, transform=torchvision.
        transforms.ToTensor())
7     # batch_size=4 使得 img0, target0 = dataset[0]、
        img1, target1 = dataset[1]、img2, target2 =
        dataset[2]、img3, target3 = dataset[3]，然后这
        四个数据作为Dataloader的一个返回
8     test_loader = DataLoader(dataset=test_data,
        batch_size=64, shuffle=True, num_workers=0,
        drop_last=False)
9     # 用for循环取出DataLoader打包好的四个数据
10    writer = SummaryWriter("logs")
11    step = 0
12    for data in test_loader:
13        imgs, targets = data # 每个data都是由4张图片组
            成，imgs.size 为 [4,3,32,32]，四张32×32图片
            三通道，targets由四个标签组成
14        writer.add_images("test_data", imgs, step)
15        step = step + 1
16
```



```
17 writer.close()
```

### 3 Dataloader 多轮次

```
1 import torchvision
2 from torch.utils.data import DataLoader
3 from torch.utils.tensorboard import SummaryWriter
4
5 # 准备的测试数据集
6 test_data = torchvision.datasets.CIFAR10("./
    dataset", train=False, transform=torchvision.
    transforms.ToTensor())
7 # batch_size=4 使得 img0, target0 = dataset[0]、
    img1, target1 = dataset[1]、img2, target2 =
    dataset[2]、img3, target3 = dataset[3], 然后这
    四个数据作为Dataloader的一个返回
8 test_loader = DataLoader(dataset=test_data,
    batch_size=64, shuffle=True, num_workers=0,
    drop_last=True)
```

```
9      # 用for循环取出DataLoader打包好的四个数据
10      writer = SummaryWriter("logs")
11      for epoch in range(2):
12          step = 0
13          for data in test_loader:
14              imgs, targets = data # 每个data都是由4张图片组成, imgs.size 为 [4,3,32,32], 四张
                                   # 32×32图片三通道, targets由四个标签组成
15              writer.add_images("Epoch: {}".format(epoch), imgs, step)
16              step = step + 1
17
18      writer.close()
```