

深度学习快速入门

01Pytorch 加载数据

POET

2024 年 2 月 10 日

1 Transforms 用途

transforms 是 Pytorch 其中一个模块，它包含了一系列常用的数据转换操作，可以对数据进行缩放、裁剪、旋转、翻转、归一化、变换等处理。

transforms 的主要作用是为了让数据适合训练神经网络，比如把 PIL 图像转换为张量，把整数标签转换为 one-hot 编码，把数据范围调整到 [0, 1] 或者 [-1, 1] 等。transforms 也可以用来增加数据的多样性和随机性，比如随机裁剪、随机旋转、随机噪声等，这样可以提高模型的泛化能力和鲁棒性

- Transforms 当成工具箱的话，里面的 class 就是不同的工具。例如像 totensor、resize 这些工具。
- Transforms 拿一些特定格式的图片，经过 Transforms 里面的工具，获得我们想要的结果。

2 Transforms 的用法

totensor 函数是将 PIL 类型或者 numpy 类型的图像转换为 tensor 张量类型因为 totensor 类没有初始化魔方方法，所以必须创建实例化对象传入实参。

```
1 from torchvision import transforms
2 from PIL import Image
3
4 img_path = "dataset/train/bees_image/21399619
5         _3e61e5bb6f.jpg"
6 img = Image.open(img_path)
7
8 tensor_trans = transforms.ToTensor() # 创建
9         transforms.ToTensor类 的实例化对象
10 tensor_img = tensor_trans(img) # 调用 transforms.
11         ToTensor类 的__call__的魔术方法
12 print(tensor_img)
```

3 常见的 Transforms 工具

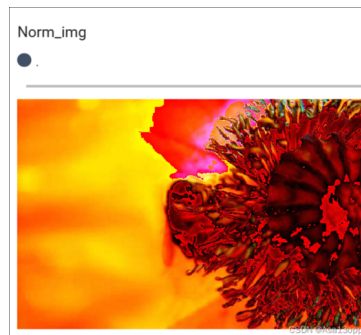
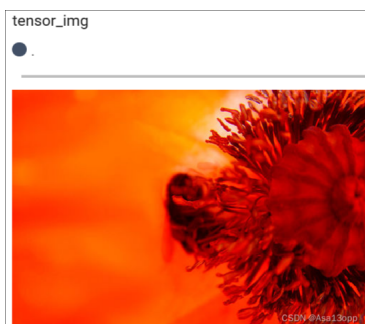
3.1 Normalize 归一化

用均值 (mean value) 和标准差 (standard deviation) 归一化张量图像:

1. 加速训练过程, 可以帮助梯度更快的收敛
2. 提高模型准确率, 将输入数据归一化到相似的尺度
3. 防止梯度消失或者爆炸

```
1   from torch.utils.tensorboard import SummaryWriter
2   from torchvision import transforms
3   from PIL import Image
4
5   img_path = "dataset/train/bees_image/21399619
6       _3e61e5bb6f.jpg"
7   img = Image.open(img_path)
8
9   writer = SummaryWriter("logs")
10
11  tensor_trans = transforms.ToTensor() # 创建
12      transforms.ToTensor类 的实例化对象
13  tensor_img = tensor_trans(img) # 调用 transforms.
14      ToTensor类 的__call__的魔术方法
15
16  #Normalize 操作
17  trans_norm = transforms.Normalize([0.5, 0.5, 0.5],
18      [0.5, 0.5, 0.5])
19  img_norm = trans_norm(tensor_img)
20
21  writer.add_image("tensor_img", tensor_img)
22  writer.add_image("Norm_img", img_norm)
```

```
19 writer.close()
```



3.2 Resize 裁剪

Resize (sequence/int): 期望的输出大小。如果大小是一个序列 sequence(h, w), 输出大小将与此匹配。如果 size 是 int 型, 图像的较小边缘将与此数字匹配。即, 如果高度 > 宽度, 那么图像将被重新缩放为 (尺寸 * 高度/宽度, 尺寸)。

```
1 from torch.utils.tensorboard import SummaryWriter
2 from torchvision import transforms
3 from PIL import Image
4
5 img_path = "dataset/train/bees_image/21399619
6 _3e61e5bb6f.jpg"
7 img = Image.open(img_path)# PIL类型的图片原始比例
8 为 500×464
9
10 writer = SummaryWriter("logs")
11
12 #Tosensor
13 tensor_trans = transforms.ToTensor() # 创建
    transforms.ToTensor类 的实例化对象
```

```
14     print(img.size)
15     trans_resize = transforms.Resize((512, 512))
16     # PIL数据类型的 img -> resize -> PIL数据类型的
        img_resize
17     img_resize = trans_resize(img)
18
19     # PIL 数据类型的 PIL -> totensor -> img_resize
        tensor
20     img_resize = tensor_trans(img_resize) # PIL类型的
        图片原始比例为 3×512×512，3通道
21
22     writer.add_image('Resize_img', img_resize)
23     writer.close()
```

