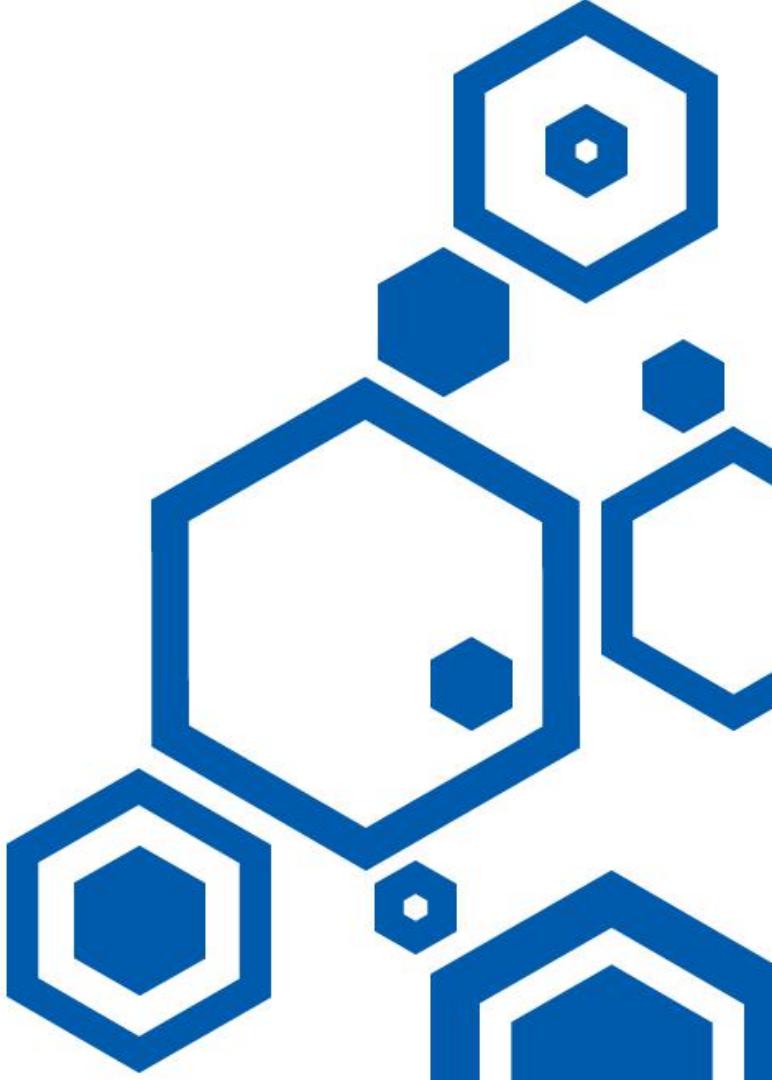




深蓝学院
shenlanxueyuan.com

第五章作业分享

主讲人 冯震



作业内容

口 实现dataset (读取)

数据集格式为list，每行内容为图像路径+label，请实现一个新的dataset，从此格式读取数据，并训练minist模型。

口 实现transform

随机旋转数据增广，图像可随机进行0,90,180,270度旋转，加入到minist模型训练中。

一、实现dataset(读取)

主要修改minist.py文件中的`_init_()`函数以及`_getitem_()`函数实现。

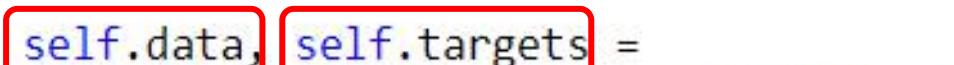
- Step1 解压数据集到工程目录中。在minist.py和minist-main.py文件中的对应位置将数据集的相关名称改成作业数据集的格式并修改`_check_exists()`函数。

```
training_file = 'train.txt'  
test_file = 'test.txt'  
  
def _check_exists(self):  
    return os.path.exists(os.path.join(  
        self.root, self.training_file)) and \  
        os.path.exists(os.path.join(self.  
            root, self.test_file))  
  
datasets.MNIST('dataset',
```

一、实现dataset(读取)

主要修改minist.py文件中的`__init__()`函数以及`__getitem__()`函数实现。

- Step2 在minist.py文件中的`__init__()`函数中使用pandas将data和targets数据读取并保存为np.array的形式。

```
import pandas as pd
dataset = pd.read_table(os.path.join
(self.root, data_file), header =
None, sep = ' ')
dataset_array = np.array(dataset)

self.data, self.targets =
dataset_array[:, 0], dataset_array[:, 1]
```

一、实现dataset(读取)

主要修改minist.py文件中的`_init_()`函数以及`_getitem_()`函数实现。

- Step3 在minist.py文件中的`_getitem_()`函数中使用cv2将对应的图片内容以及图片标签进行读取。

```
img_pth, target = self.data[index],  
int(self.targets[index])
```

```
img = cv2.imread(os.path.join(self.  
root, img_pth), flags = cv2.  
IMREAD_UNCHANGED)
```

```
img = Image.fromarray(img, mode='L')
```

Test set: Average loss: 0.0482, Accuracy: 9851/10000 (99%)

- Step4 进行模型训练

二、实现transform

通过在transforms.py文件中创建相应的旋转函数my_rotate()实现。

- Step1 在transforms.py文件中仿照其他函数形式创建旋转函数，并编写图片旋转程序，这里我使用了np.rot90()函数进行实现。

```
class my_rotate(object):
    # 注意这里有个PIL image格式的转换
    def __call__(self, img):
        im = np.array(img)
        for _ in range(np.random.randint(0, 4)):
            im = np.rot90(im)
        return Image.fromarray(im, mode='L')

    def __repr__(self):
        return self.__class__.__name__ + '()'
```

二、实现transform

通过在transforms.py文件中创建相应的旋转函数my_rotate()实现。

- Step2 在transforms.py文件中__all__中将自己的函数名进行添加，以便import模块时导入my_rotate()函数。

```
__all__ = ["Compose", "ToTensor", "ToPILImage",
"Normalize", "Resize", "Scale", "CenterCrop", "Pad",
"Lambda", "RandomApply", "RandomChoice",
"RandomOrder", "RandomCrop",
"RandomHorizontalFlip",
"RandomVerticalFlip", "RandomResizedCrop",
"RandomSizedCrop", "FiveCrop", "TenCrop",
"LinearTransformation",
"ColorJitter", "RandomRotation",
"RandomAffine", "Grayscale",
"RandomGrayscale", "my_rotate"]
```

二、实现transform

通过在transforms.py文件中创建相应的旋转函数my_rotate()实现。

- Step3 在minist-main.py文件中的transforms.Compose函数处添加自己的旋转函数并进行模型的训练。

```
train_loader = torch.utils.data.DataLoader(  
    datasets.MNIST('dataset', train=True, download=True,  
    transform=transforms.Compose([  
        transforms.my_rotate(),  
        transforms.ToTensor(),  
        transforms.Normalize((0.1307,),  
        (0.3081,))  
    ])),  
    batch_size=args.batch_size, shuffle=True, **kwargs)
```

Test set: Average loss: 0.2492, Accuracy: 9297/10000 (93%)

Q&A



感谢各位聆听 !
Thanks for Listening !

