
```python
def tng_dataloader():
     d = MNIST()
     # 4: Add distributed sampler
     # sampler sends a portion of tng data to each machine
     dist_sampler = DistributedSampler(dataset)
     dataloader = DataLoader(d, shuffle=False, sampler=dist_sampler)
def main_process_entrypoint(gpu_nb):
     # 2: set up connections  between all gpus across all machines
     # all gpus connect to a single GPU "root"
     # the default uses env://
     world = nb_gpus * nb_nodes
     dist.init_process_group("nccl", rank=gpu_nb, world_size=world)
     # 3: wrap model in DPP
     torch.cuda.set_device(gpu_nb)
     model.cuda(gpu_nb)
     model=DistributedDataParallel(model,device_ids=[gpu_nb])
     # train your model now...
if  __name__ == '__main__':
     # 1: spawn number of processes
     # your cluster will call main for each machine
     mp.spawn(main_process_entrypoint, nprocs=8)
```


[公开]训练一个alexnet

pytorch实现alexnet:https://blog.csdn.net/weixin_44023658/article/details/105798326

本代码若要运行成功，需要先将数据下载下来并做处理。

数据下载路径为：http://download.tensorflow.org/example_images/flower_photos.tgz

处理代码为：week10_210708/alexnet/splitdata.py

处理办法：

1 在week10_210708/alexnet/内建立文件夹flower_data,将flower_photos.tgz解压到flower_data内

2 在alexnet内，运行python splitdata.py 处理数据，处理完成后，flower_data文件夹内会新文件夹train。

```
cd week10_210708/alexnet/
#下载数据:
wget http://download.tensorflow.org/example_images/flower_photos.tgz
#解压缩,并将数据文件放在flower_data内
tar -xf flower_photos.tgz
mkdir flower_data
mv flower_photos flower_data/
```

```
(python3.6.7) zhaomingming@x86_64-apple-darwin13 alexnet % ls flower_data
flower_photos
```

把解压缩得到的flower_photos放在flower_data内

```
#运行数据处理程序, 将数据分成train,split两部分
python splitdata.py
```

处理过程以及，处理后的文件夹：

```
(python3.6.7) zhaomingming@x86_64-apple-darwin13 alexnet % python splitdata.py
[roses] processing [641/641]
[sunflowers] processing [699/699]
[daisy] processing [633/633]
[dandelion] processing [898/898]
[tulips] processing [799/799]
processing done!
(python3.6.7) zhaomingming@x86_64-apple-darwin13 alexnet % ls flower_data
flower_photos   train           val
```

处理过程以及处理后的文件夹

然后运行：

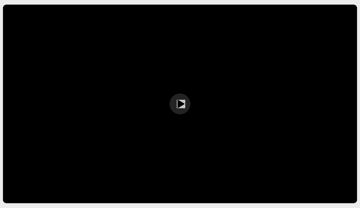
```
python train.py
```

得到：

```
(python3.6.7) zhaomingming@x86_64-apple-darwin13 alexnet % python train.py
cpu
train loss: 10 %[*****->.........................................]1.606_
```

训练命令以及训练过程

训练过程如视频所示【8倍速】：



训练完成后，会发现文件夹内多了AlexNet.pth,class_indices.json两个文件

AlexNet.pth是模型参数保存成的文件，像卷积中的 weights和bias的值就保存在了这里。

class_indces.json是类别号和类别名字对应关系表

有了这两个文件，我们就可以使用reference.py 去使用模型进行分类了。

使用如下命令：

daisy就是图片的类别

0.9925为置信度

我们看下reference.py的源码：

```python
1 #predict.py
2 
3 import torch
4 from model import AlexNet
5 from PIL import Image
6 from torchvision import transforms
7 import matplotlib.pyplot as plt
8 import json
9 
10 data_transform = transforms.Compose(
11     [transforms.Resize((224, 224)),
12      transforms.ToTensor(),
13      transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])
14 
15 # load image
16 imgpath="./flower_data/flower_photos/daisy/18023717391_e2c9089e10.jpg"
17 imgpath="./flower_data/flower_photos/daisy/4851353993_2cbbbd1040_n.jpg"
18 
19 img = Image.open(imgpath)  #验证
20 #img = Image.open("./roses.jpg")     #验证玫瑰花
21 plt.imshow(img)
22 # [N, C, H, W]
23 img = data_transform(img)
24 # expand batch dimension
25 img = torch.unsqueeze(img, dim=0)
26 
27 # read class_indict
28 try:
29     json_file = open('./class_indices.json', 'r')
30     class_indict = json.load(json_file)
31 except Exception as e:
32     print(e)
33     exit(-1)
34 
35 # create model
36 model = AlexNet(num_classes=5)
37 # load model weights
38 model_weight_path = "./AlexNet.pth"
39 model.load_state_dict(torch.load(model_weight_path))
40 model.eval()
41 with torch.no_grad():
42     # predict class
43     output = torch.squeeze(model(img))
44     predict = torch.softmax(output, dim=0)
45     predict_cla = torch.argmax(predict).numpy()
46 print(class_indict[str(predict_cla)], predict[predict_cla].item())
47 plt.show()
```

修改第16行，可修改测试的图片。

有任何问题，联系明明老师

明润 (安江)

北京 朝阳

扫一扫上面的二维码图案，加我微信

评论区

栾川

很详细；还想问一下、如何部署？

2021-07-10 16:34:01删除

admin

退出

B *I*

说两句心得？

提交