



deepshare.net

深度之眼

木薯叶病分类直播答疑

导师：

本次答疑内容

Course content



1、重难点知识串讲

2、学员问题解答

3、互动答疑

重难点知识串讲

梳理本阶段的重难点知识

学员问题解答

对大家提出的问题进行解答

答疑问题



deepshare.net

深度之眼

@于东阳:

想输出分类的结果，每个类别的概率是多少，应该怎么做呢？

```
#for epoch in range(CFG['epochs']-3):
for i, epoch in enumerate(CFG['used_epochs']):
    print(i)
    print(epoch)
    print(fold)
    model.load_state_dict(torch.load('../input/tf-efficientnet-b4-ns0102/{_fold_}_{_}'.format(CFG['model_arch'], i, epoch)))

    with torch.no_grad():
        for _ in range(CFG['tta']):
            val_preds += [CFG['weights'][i]/sum(CFG['weights'])/CFG['tta']*inference_one_epoch(model, val_loader, device)]
            tst_preds += [CFG['weights'][i]/sum(CFG['weights'])/CFG['tta']*inference_one_epoch(model, tst_loader, device)]

val_preds = np.mean(val_preds, axis=0)
tst_preds = np.mean(tst_preds, axis=0)
```

答疑问题



deepshare.net

深度之眼

@明天去见你：

请问为什么我的训练速度会这么慢， 是没有使用GPU吗， 本机GPU应该怎么使用

```
C:\ProgramData\Anaconda3\lib\site-packages\torch\cuda\amp\grad_scaler.py:115: UserWarning: torch.cuda.amp.GradScaler is enabled, but CUDA is not available. Disabling.
  warnings.warn("torch.cuda.amp.GradScaler is enabled, but CUDA is not available. Disabling.")
0%
| 0/8559 [00:00<?, ?it/s]C:\ProgramData\Anaconda3\lib\site-packages\torch\cuda\amp\autocast_mode.py:114: UserWarning: torch.cuda.amp.autocast only affects CUDA ops, but CUDA is not available. Disabling.
  warnings.warn("torch.cuda.amp.autocast only affects CUDA ops, but CUDA is not available. Disabling.")
epoch 0 loss: 1.0118: 6% █████
| 511/8559 [36:14<9:30:42, 4.25s/it]
```

答疑问题



@马建华:

1. 老师能不能具体再说一下在baseline哪个地方修改fmix, cutmix 这些数据增强?
2. 打卡作业二中提到的“修改训练方法”, 没太懂具体要怎么修改?
3. NNI是不是只能用于.py脚本文件提交任务, 不能用于jupyter notebook?
4. 第一次课和第二次课给的baseline中train_one_epoch函数有不同。with autocast(): 包含的命令行不同, 会导致不同的输出结果吗? (如图所示)

答疑问题



deepshare.net

深度之眼

@马建华:

```
, image_labels) in pbar: # 遍历每个 batch
    imgs = imgs.to(device).float()
    labels = image_labels.to(device).long()

    # 开启自动混精度
    with torch.cuda.amp.autocast():
        preds = model(imgs) # 前向传播, 计算预测值
        loss_fn(preds, labels) # 计算 loss

    scaler.scale(loss).backward() # 对 loss scale, scale梯度

    # 使用指数平均
    if running_loss is None:
        running_loss = loss.item()
    else:
        running_loss = running_loss * .99 + loss.item() * .01

    if (step + 1) % CF['accum_iter'] == 0 or ((step + 1) == len(train_loader)):
        step(
            optimizer) # unscale 梯度, 如果梯度没有 overflow, 使用 opt 更新梯度, 否则不更新
        scaler.update() # 等着下次 scale 梯度
        optimizer.zero_grad() # 梯度清空

    if scheduler is not None and schd_batch_update: # 学习率调整策略
        scheduler.step()

    # 打印
    description = f'epoch {epoch} loss: {running_loss:.4f}'
    pbar.set_description(description)
```

第二次课baseline的train_one_epoch函数

```
for step, (imgs, image_labels) in pbar:
    imgs = imgs.to(device).float()
    image_labels = image_labels.to(device).long()

    # print(image_labels.shape, exam_label.shape)
    with autocast():
        image_preds = model(imgs) # output = model(input)
        # print(image_preds.shape, exam_pred.shape)

        loss = loss_fn(image_preds, image_labels)

        scaler.scale(loss).backward()

    if running_loss is None:
        running_loss = loss.item()
    else:
        running_loss = running_loss * .99 + loss.item() * .01

    if ((step + 1) % CF['accum_iter'] == 0 or ((step + 1) == len(train_loader))):
        # may unscale_ here if desired (e.g., to allow clipping unscaled gradients)

        scaler.step(optimizer)
        scaler.update()
        optimizer.zero_grad()

        if scheduler is not None and schd_batch_update:
            scheduler.step()

    if ((step + 1) % CF['verbose_step'] == 0 or ((step + 1) == len(train_loader))):
        description = f'epoch {epoch} loss: {running_loss:.4f}'

        pbar.set_description(description)

if scheduler is not None and not schd_batch_update:
    scheduler.step()
```

第一次baseline的train_one_epoch函数

答疑问题



@正心，正己：

1:先不考虑网络更改网络后要重新训练，请问代码上如何实现从网络中抽取某一层的特征图输出，接入到一个模块；

举例而言，比如代码上如何实现，把SiLu()的输出特征图，接入到RFB模块中输出后再接入到classifier中，我直接如图将RFB_small()这个类写进去，Pycharm中可以跑通，但是kaggle kernel上出错；

这是RFB模块---

<https://github.com/pprp/SimpleCVReproduction/blob/master/Plug-and-play%20module/ReceptiveFieldModule.py>

这是我改的方法：<https://www.kaggle.com/chuyunxinlan/cassava-rfb-train> 请老师您帮忙看一下，代码上我该如何实现将网络中某一层的特征输出，接入到RFB模块这个类中，然后再经过他输出；谢谢；

答疑问题



deepshare.net

深度之眼

@正心，正己：

```
BatchNorm2d(2304, eps=1e-05, momentum=0.1, affine=True)
): SiLU(inplace=True)
SqueezeExcite(
nv_reduce): Conv2d(2304, 96, kernel_size=(1, 1), stride=
t1): SiLU(inplace=True)
nv_expand): Conv2d(96, 2304, kernel_size=(1, 1), stride=
_pwl): Conv2d(2304, 384, kernel_size=(1, 1), stride=(1,
): BatchNorm2d(384, eps=1e-05, momentum=0.1, affine=True,
): Conv2d(384, 1536, kernel_size=(1, 1), stride=(1, 1), b
nNorm2d(1536, eps=1e-05, momentum=0.1, affine=True, trac
U(inplace=True)
1): SelectAdaptivePool2d (pool_type=avg, flatten=True)
): Linear(in_features=1536, out_features=1000, bias=True
```

SRTB

```
M4
# 模型构建
class CassvaImgClassifier(nn.Module):
    def __init__(self, model_arch, n_class, pretrained=False):
        super().__init__()
        self.model = timm.create_model(model_arch,
pretrained=pretrained)
        # n_features = self.model.classifier.in_features

        self.model.global_pool = nn.Sequential(
            BasicFRB_small(in_planes=1792, out_planes=512,
stride=1, scale=0.1),
            SelectAdaptivePool2d(output_size=1, flatten=True)
        )

        self.model.classifier = nn.Linear(512, n_class)

    def forward(self, x):
        x = self.model(x)
        return x
```


答疑问题



deepshare.net

深度之眼

@嘉Tarus:

1. 关于梯度累加的问题 图片1是 `accum_iter=2`, `loss`没有除以2 图片2是 `accum_iter=4` 并且`loss`除以4 但是感觉最后结果都差不多呢? 但是`loss`差了4倍啊
2. 看了下其他kaggle上的baseline 为什么他们batchsize 32 都可以, 我到16就不行了 还是用了梯度累加。都是在kaggle自带的GPU运行。我们的baseline哪里特别吃显存呢?

答疑问题



deepshare.net

深度之眼

@嘉Tarus:

```
validation multi-class accuracy = 0.8879

epoch 2 loss: 0.3791: 100%|██████████| 1070/1070 [18:25<00:00, 1.03s/it]
epoch 2 loss: 0.3358: 100%|██████████| 268/268 [01:55<00:00, 2.32it/s]
0%|          | 0/1070 [00:00<?, ?it/s]

validation multi-class accuracy = 0.8895

epoch 3 loss: 0.3484: 100%|██████████| 1070/1070 [18:26<00:00, 1.03s/it]
epoch 3 loss: 0.3367: 100%|██████████| 268/268 [01:54<00:00, 2.34it/s]
0%|          | 0/1070 [00:00<?, ?it/s]

validation multi-class accuracy = 0.8909

epoch 4 loss: 0.3226: 100%|██████████| 1070/1070 [18:25<00:00, 1.03s/it]
epoch 4 loss: 0.3395: 100%|██████████| 268/268 [01:55<00:00, 2.33it/s]
0%|          | 0/1070 [00:00<?, ?it/s]

validation multi-class accuracy = 0.8909

epoch 5 loss: 0.3070: 100%|██████████| 1070/1070 [18:29<00:00, 1.04s/it]
epoch 5 loss: 0.3439: 100%|██████████| 268/268 [01:54<00:00, 2.34it/s]
0%|          | 0/1070 [00:00<?, ?it/s]

validation multi-class accuracy = 0.8897

epoch 6 loss: 0.2956: 100%|██████████| 1070/1070 [18:26<00:00, 1.03s/it]
epoch 6 loss: 0.3427: 100%|██████████| 268/268 [01:54<00:00, 2.33it/s]
0%|          | 0/1070 [00:00<?, ?it/s]
```

```
validation multi-class accuracy = 0.8729

epoch 1 loss: 0.1063: 100%|██████████| 1070/1070 [17:57<00:00, 1.01s/it]
epoch 1 loss: 0.3666: 100%|██████████| 268/268 [01:59<00:00, 2.25it/s]
0%|          | 0/1070 [00:00<?, ?it/s]

validation multi-class accuracy = 0.8776

epoch 2 loss: 0.0973: 100%|██████████| 1070/1070 [17:29<00:00, 1.02it/s]
epoch 2 loss: 0.3439: 100%|██████████| 268/268 [01:52<00:00, 2.38it/s]
0%|          | 0/1070 [00:00<?, ?it/s]

validation multi-class accuracy = 0.8843

epoch 3 loss: 0.0899: 100%|██████████| 1070/1070 [17:30<00:00, 1.02it/s]
epoch 3 loss: 0.3340: 100%|██████████| 268/268 [01:53<00:00, 2.37it/s]
0%|          | 0/1070 [00:00<?, ?it/s]

validation multi-class accuracy = 0.8951

epoch 4 loss: 0.0818: 100%|██████████| 1070/1070 [17:30<00:00, 1.02it/s]
epoch 4 loss: 0.3364: 100%|██████████| 268/268 [01:52<00:00, 2.38it/s]
0%|          | 0/1070 [00:00<?, ?it/s]

validation multi-class accuracy = 0.8925

epoch 5 loss: 0.0786: 100%|██████████| 1070/1070 [17:30<00:00, 1.02it/s]
epoch 5 loss: 0.3386: 100%|██████████| 268/268 [01:52<00:00, 2.38it/s]

validation multi-class accuracy = 0.8902
```


互动时间

结语

—— 结 语 ——

感谢同学们参加今晚的直播答疑！

课下，请好好**总结和回顾知识点**

