








成绩

43	 KEUAX07X	0.63878	10	2023-08-05 12:50:31
44	 RUNO0B	0.63857	13	2023-08-08 10:24:24
45	 FwFjedEcN	0.63782	1	2023-08-07 21:49:49
46	 小东西	0.63453	15	2023-08-06 10:46:23
47	 R11111	0.63404	7	2023-08-05 09:05:50
48	 None	0.63375	14	2023-08-07 18:09:14
49	 n10T7d1uc	0.63115	3	2023-08-05 18:11:28

您当前最佳成绩为0.63453提交时间为 2023-08-06 10:46:23 具体成绩请前往提交结果页面,查看提交记录
(注: 以当前所在团队提交的最佳成绩为准)

过程

修改

我的代码是在原来的CNN（深度学习方法）的基础之上修改的，我对这个示例的方法做了下面的修改：

- 1、使用了更多的标准库和第三方库来进行数据加载和图像增强，例如使用了 `albumentations` 库来实现数据增强。
- 2、使用了 `paddle.io.DataLoader` 来构建数据加载器，将数据加载到模型中进行训练、验证和测试。
- 3、定义了一个新的 `xunFeiNet` 类，继承自 `paddle.nn.Layer`，并添加了额外的卷积层和全连接层。
- 4、采用了一种更简洁的方式来训练和验证模型，并且将准确率等信息直接输出在训练和验证过程中。
- 5、可以使用 `paddle.optimizer.lr` 模块中的学习率调度器（learning rate scheduler）来实现动态学习率。一种常用的策略是 `CosineAnnealingDecay`，它会按照余弦退火的方式调整学习率。这样改了之后，成绩就上升了。

采用的网络

我和示例代码一样都是采用的 `resnet18` 作为主干网络，但是我在原有的代码基础之上替换了最后的全连接层，将输出类别数设置为25。同时我还额外地添加了额外的卷积层和全连接层，并对模型进行了修改，添加了更多的参数。

遇到的问题

1、数据格式设置不正确。因为在PaddlePaddle中，默认的数据格式为NCHW（通道数在第二维），而在加载预训练的ResNet-18模型时，默认的数据格式是NHWC（通道数在第四维），所以为了解决这个问题，我们需要在加载预训练模型时，明确指定数据格式为NCHW。

2、预训练模型的全连接层输出维度与新模型的输出维度不匹配。预训练模型是在ImageNet数据集上训练的，其输出维度为[512, 1000]，而你的新模型输出维度是[512, 25]，这导致了形状不匹配的警告。因为这个并不会影响训练过程，所以我直接忽略了这些警告，但是后续还是解决了这个问题。

后续修改

第二次任务是在第一次的基础之上进行了一定的修改，但是实在是想不出其他的方法，试着换了预训练的网络模型，但是不是报错就是收敛的速度过慢，最后是增加了训练的epoch，但是可能是训练的轮数过多，导致后面的预测的结果还不如第一次任务的结果，这导致成绩无法提升。

其实应该可以将ret18的网络更换成为ret50，这样的效果应该更好，但是由于本机的配置跑不起来，飞浆平台的GPU又是限时的，所以没有实际的操作。