

# 计算机视觉-HW6

xTryer

2024/12/18

## 1 Data augmentation

对输入的图片训练数据进行数据增强，以此来一定程度上防止过拟合 (overfit)。

我使用了对图片进行随机旋转、随机裁剪、随机翻转以及整体的扰动来进行数据增强。

---

```
1 train_transform = transforms.Compose(  
2     [transforms.ToTensor(),  
3       transforms.RandomHorizontalFlip(),#随机水平翻转  
4       transforms.RandomCrop(32,padding=4),#随机裁剪  
5       torchvision.transforms.RandomRotation(10),#随机旋转  
6       transforms.Normalize((0.476, 0.521, 0.489), (0.478, 0.456, 0.498))])#扰动  
7 )
```

---

## 2 VGG

VGG, 主要是通过使用较小的卷积核 (3\*3) 代替较大的卷积核，并增加网络层数来学习更复杂的信息，首先定义一个 vgg\_block 函数用于创建一个 vgg 层。

---

```
1 def vgg_block(self,in_channel,out_channel):  
2     vgg_seq=nn.Sequential(  
3         nn.Conv2d(in_channel,out_channel,3,1,1),  
4         nn.BatchNorm2d(out_channel),  
5         nn.ReLU(inplace=True),  
6         nn.Dropout()  
7     )  
8     return vgg_seq
```

---

然后利用这个函数定义非线性层，最后用一个线性全连接层完成分类。在层间加入 batchnorm 和 dropout, 使梯度下降更稳定并一定程度上防止过拟合。

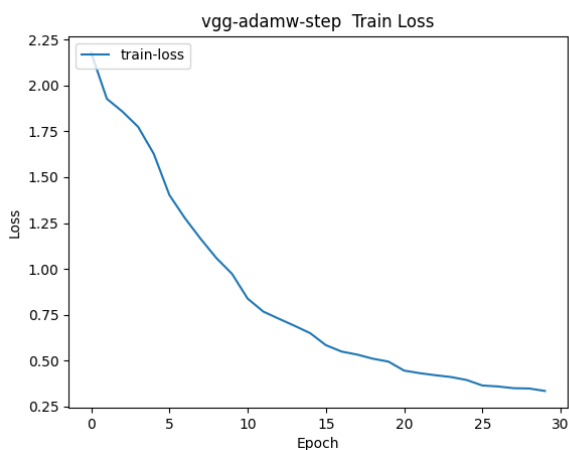
考虑使用 AdamW+StepLR 进行模型训练, 训练参数如下:

- $epochnum = 30$

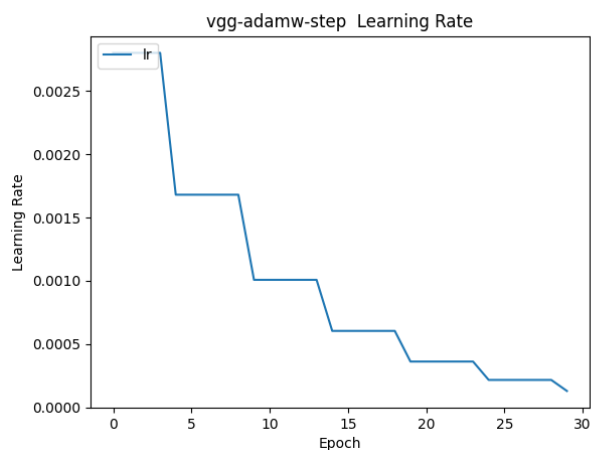
- $batchsize = 64$
- $AdamW\_lr = 0.0028$
- $steplr\_gamma = 0.6$
- $steplr\_stepsize = 5.0$

最后在 epoch=30 时，得到在训练集上正确率为 0.93, 验证集上正确率为 0.86, 测试集上正确率为 0.88

可视化训练结果:

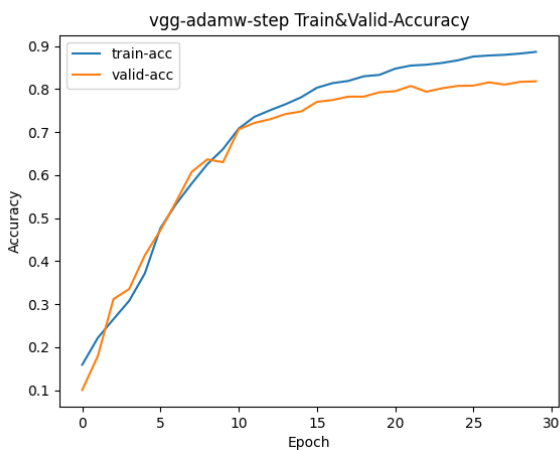


(a) TrainLoss

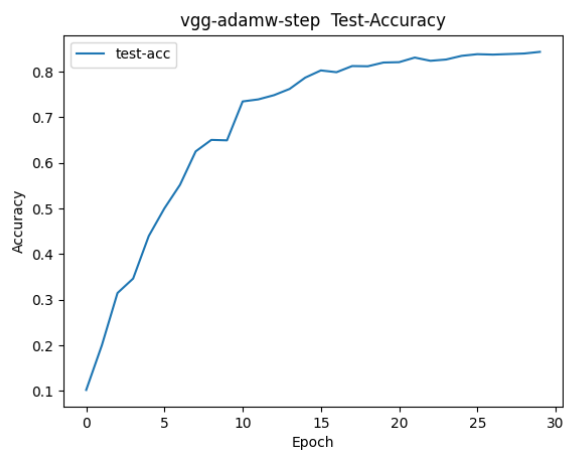


(b) LearningRate

图 1: VGG-Adamw-Steplr



(a) Accuracy on TrainSet & ValidSet



(b) Accuracy on TestSet

图 2: VGG-Adamw-Steplr

### 3 ResNet

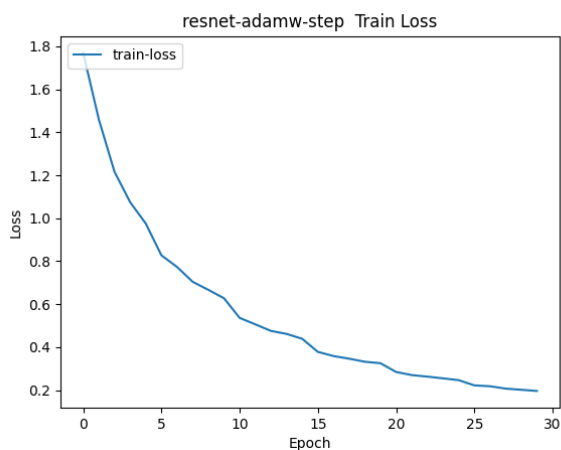
ResNet 引入了一个残差块 (Residual Block), 通过堆叠多个残差块来构建整个网络结构, 避免了深层网络训练中随着网络层数增加梯度消失或爆炸的问题。

考虑使用 AdamW+StepIrr 进行模型训练, 训练参数如下:

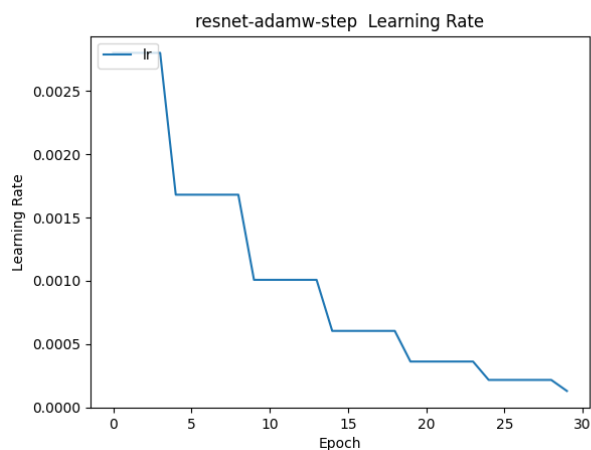
- $epochnum = 30$
- $batchsize = 64$
- $AdamW\_lr = 0.0028$
- $steplr\_gamma = 0.6$
- $steplr\_stepsize = 5.0$

最后在 epoch=30 时, 得到在训练集上正确率为 0.89, 验证集上正确率为 0.82, 测试集上正确率为 0.84

可视化训练结果:



(a) TrainLoss

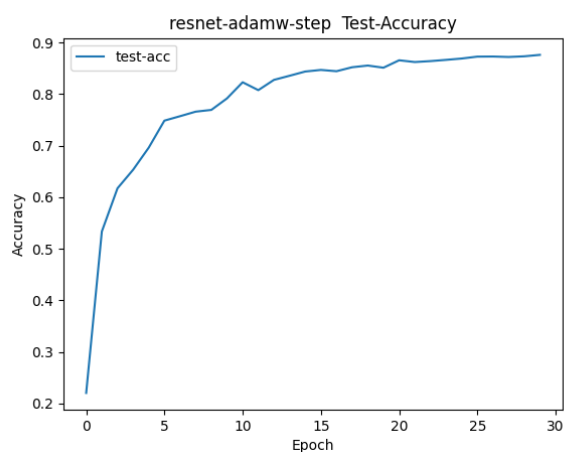


(b) LearningRate

图 3: ResNet-Adamw-StepIrr



(a) Accuracy on TrainSet & ValidSet



(b) Accuracy on TestSet

图 4: ResNet-Adamw-Stepplr

## 4 ResNext

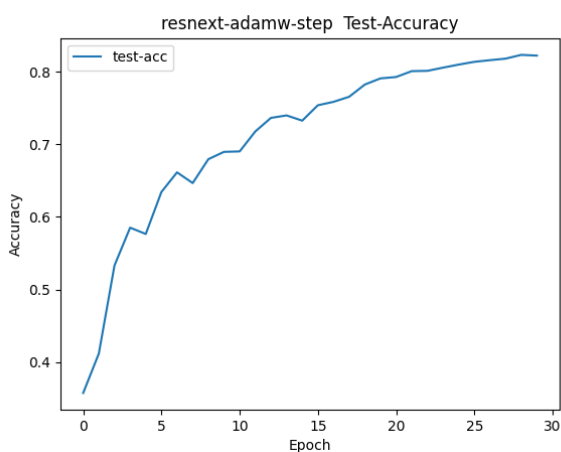
ResNeXt 在 ResNet 的基础上提高了网络的并行路径数量，将每个残差块分成多个并行的小组，在每个小组内部进行卷积操作，然后将结果进行拼接，形成最后的输出。

考虑使用 AdamW+Stepplr 进行模型训练, 训练参数如下:

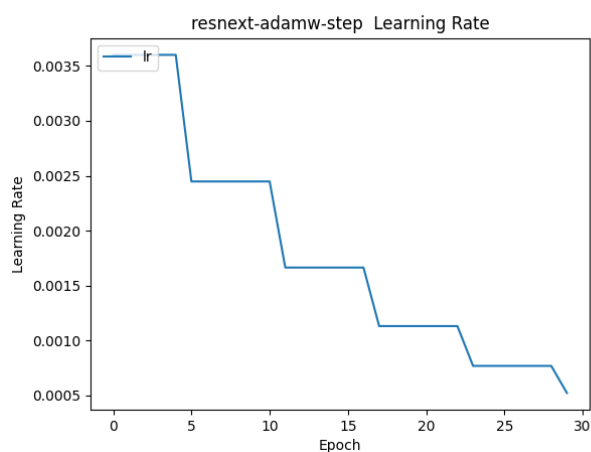
- $epochnum = 30$
- $batchsize = 64$
- $AdamW\_lr = 0.0036$
- $steplr\_gamma = 0.68$
- $steplr\_stepsize = 6.0$

最后在 epoch=30 时，得到在训练集上正确率为 0.84, 验证集上正确率为 0.79, 测试集上正确率为 0.82

可视化训练结果:

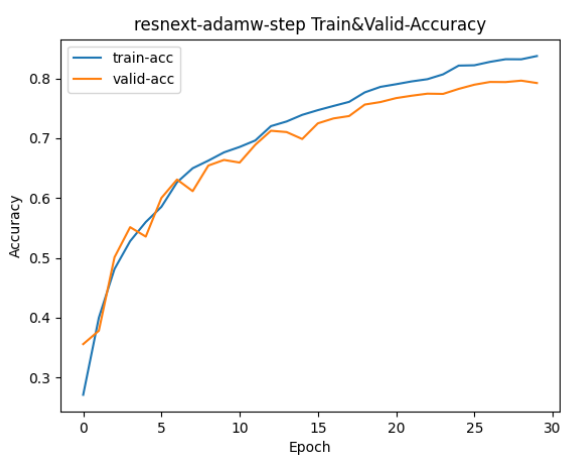


(a) TrainLoss

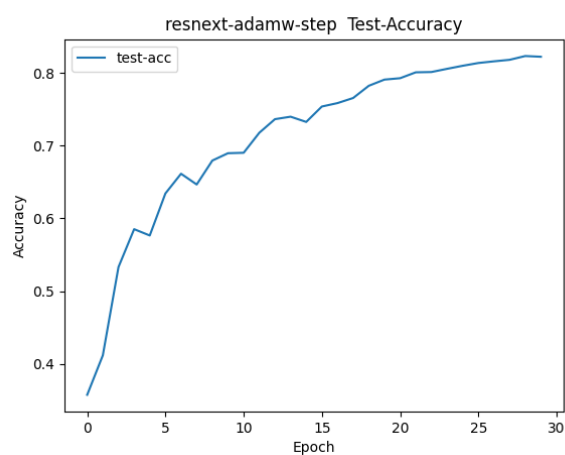


(b) LearningRate

图 5: ResNet-Adamw-StepLr



(a) Accuracy on TrainSet & ValidSet



(b) Accuracy on TestSet

图 6: ResNet-Adamw-StepLr

## 5 Upload

- 最后的模型文件分别为'vgg30.pt','resnet30.pt','resnext30.pt', 可以用 test.py 测试
- runs 文件夹下包含了 tensorboard 文件
- report-data 文件夹下包含了报告中出现的所有图片以及训练日志文件

最终使用 test.py 测试 vgg30.pt,resnet30.pt,resnext30.pt, 分别得到结果:

- vgg30.pt:test-acc: 0.8435999751091003
- resnet30.pt:test-acc: 0.8758999705314636
- resnext30.pt:test-acc: 0.8073999881744385

使用 Tensorboard 得到的可视化结果:

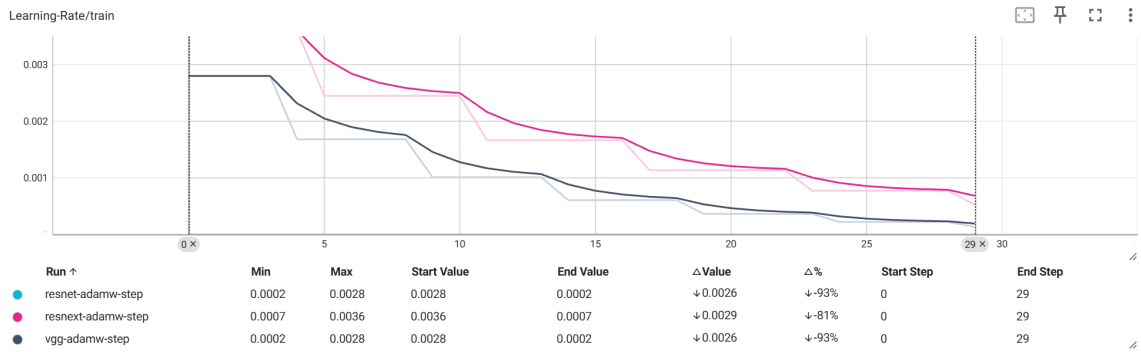


图 7: Visualization-Total-TensorBoard-LearningRate

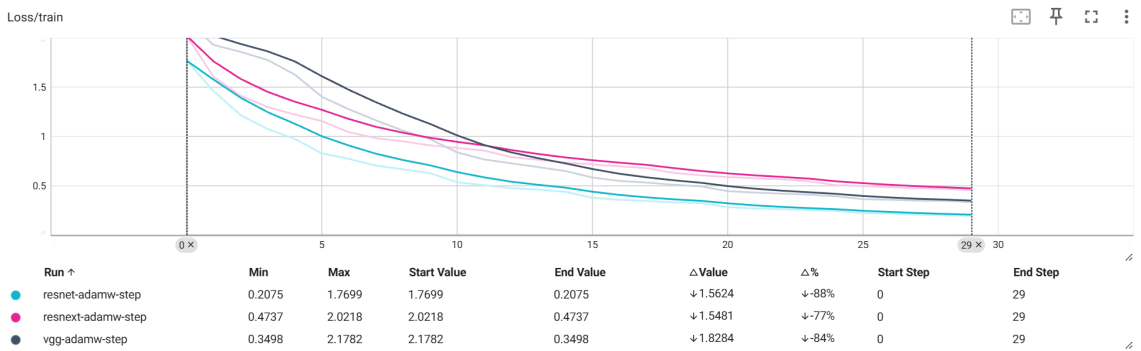


图 8: Visualization-Total-TensorBoard-TrainLoss

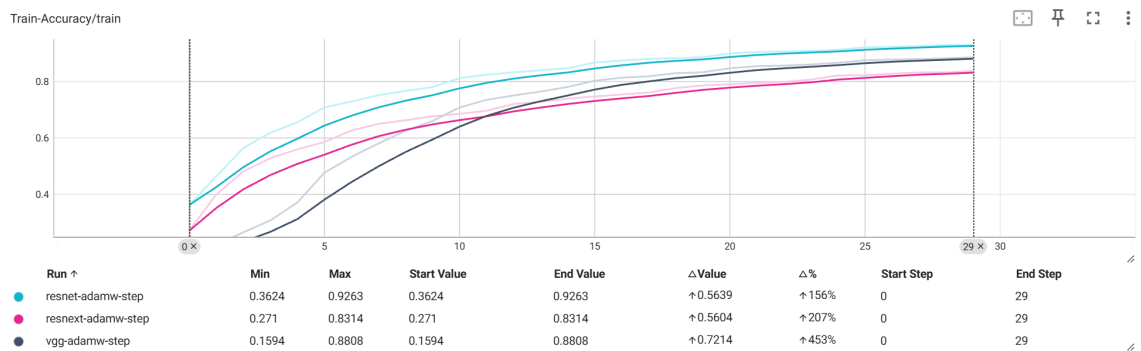


图 9: Visualization-Total-TensorBoard-TrainAccuracy

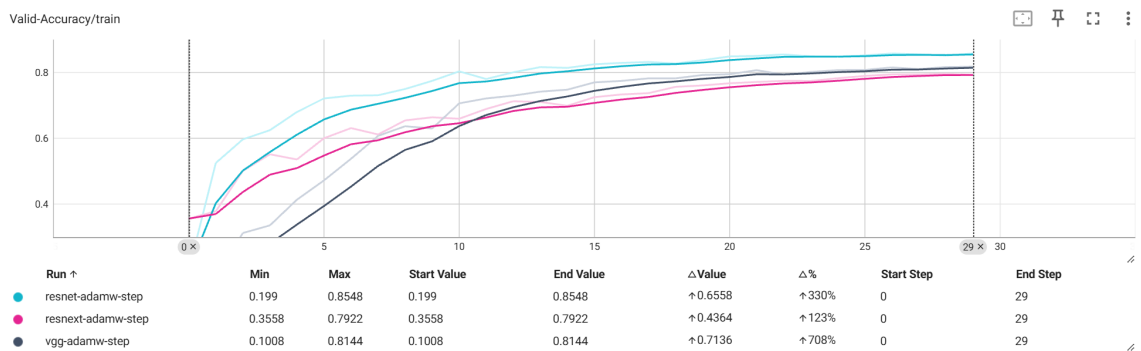


图 10: Visualization-Total-TensorBoard-ValidAccuracy

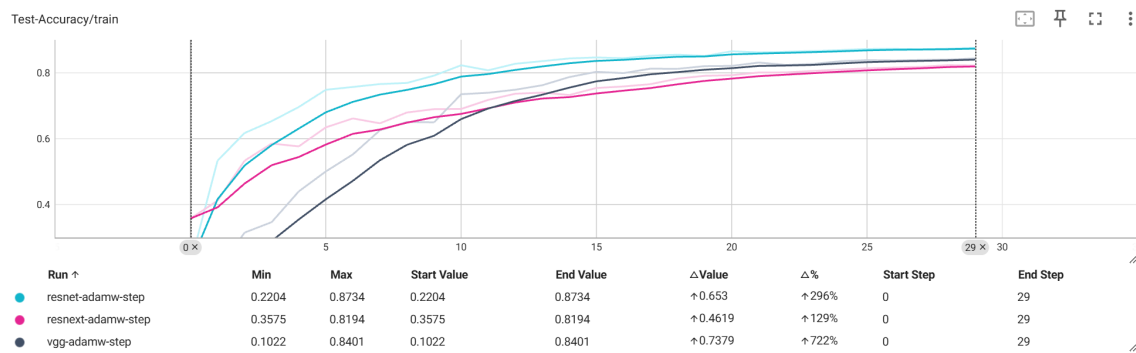


图 11: Visualization-Total-TensorBoard-TestAccuracy