

# 华中科技大学

## 计算机视觉实验二报告

姓 名：

专 业：

班 级：

学 号：

指导教师：

分数	
教师签名	

年月日

# 目录

<b>实验一：基于卷积神经网络的 CIFAR-10 数据集分类 .....</b>	<b>1</b>
1.1 任务要求 .....	1
1.2 实验内容 .....	1
1.3 VGGNet .....	4
1.3.1 vgg11_bn .....	4
1.3.2 vgg13_bn .....	5
1.3.3 vgg16_bn .....	5
1.3.4 vgg19_bn .....	6
1.3.5 分析比较 .....	7
1.4 ResNet .....	8
1.4.1 resnet18 .....	8
1.4.2 resnet34 .....	9
1.4.3 resnet50 .....	9
1.4.4 分析比较 .....	10

# 实验一：基于卷积神经网络的 CIFAR-10 数据集分类

## 1.1 任务要求

设计一个卷积神经网络，在 CIFAR-10 数据集上实现分类任务。

## 1.2 实验内容

我参照 ResNet 中的 building block 构建了一个简单的、9 层的 ResNet（下文中简称 resnet9），实现代码见代码文件中的 modules/mymodel.py。

ResNet 的 building block 结构如下图所示：

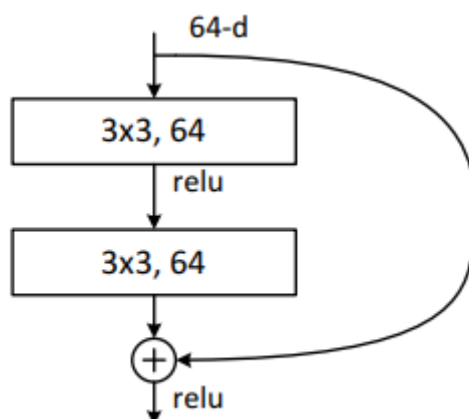


图 1-1: ResNet 的 building block  
resnet9 的网络结构如下图所示：

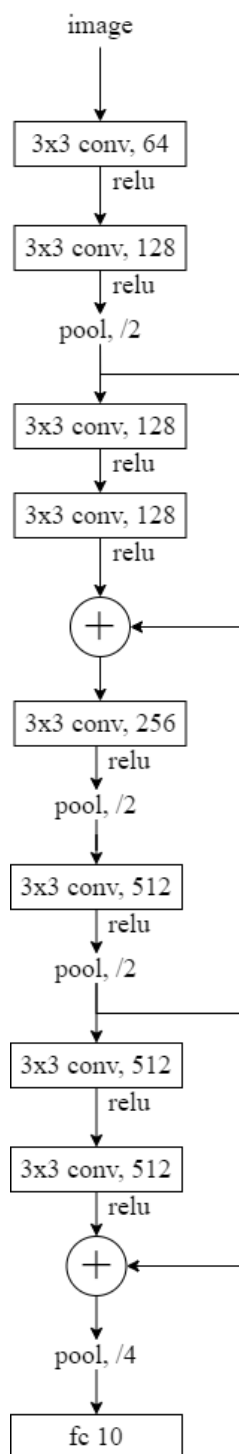


图 1-2: resnet9 网络结构

每一层的输出通道数已在上图中表明。此外，所有卷积层的 kernel size 为 (3,3)，stride 为 1，padding 为 1，不改变输入的大小；所有池化层均采用 MaxPool 进行降采样；每一个卷积层之后都使用 BatchNorm。

resnet9 的训练配置见 configs/resnet9.py。具体如下：

- 1、训练集分割为 49k/1k, train/val, batch\_size 为 128，总共训练 8 个 epoch；
- 2、使用 Adam 优化器，weight\_decay 为 1e-4；

- 3、学习率调整策略为 OneCycleLR，最大学习率为  $1e-2$ ;
- 4、梯度截断为 0.1，避免梯度爆炸;
- 5、数据增强方法参考 ResNet 原论文，先对图片上下左右各填充 4 个像素，然后随机剪裁出大小为(32,32)的图片，再进行随机水平翻转，最后进行归一化处理。

训练过程见训练日志 `work_dir\resnet\resnet9\20230402_174749.log`，最后保存训练过程中验证集准确率最高的模型权重参数(`work_dir\resnet\resnet9\epoch8.pth`)。在测试集上测试结果如下图所示：

```
Total Accuracy: 90.5600%
plane: 91.8000%
car: 95.9000%
bird: 86.6000%
cat: 80.4000%
deer: 91.4000%
dog: 85.8000%
frog: 93.3000%
horse: 91.5000%
ship: 94.9000%
truck: 94.0000%
```

图 1-3: resnet9 测试结果

由上图可知：resnet9 的测试准确率达到 90.56%。

根据训练日志，可视化训练过程，如下图所示：

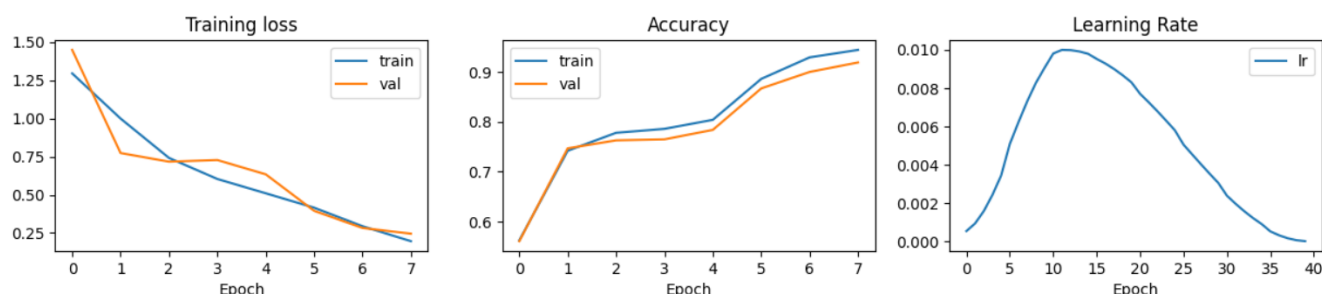


图 1-4: resnet9 训练过程可视化

分析可知：模型的损失仍在下降，准确率仍在上升，并未完全收敛，还有较大的提升空间，可以增加训练轮数、调整学习率进一步提高模型的准确率、降低损失。

本实验由于时间有限，为了提高模型收敛速度、节省训练时间、提高训练效率，只训练了较少的轮数、使用了较大的学习率和 OneCycleLR 学习率调整策略。后续可以继续实验、调参，寻找模型的最优参数。

## 1.3 VGGNet

本实验中，我还使用了 VGGNet。模型的代码实现见 `modules\vgg.py`，参考 PyTorch 源码。为了更好的适配 CIFAR-10 数据集、提高训练效率，我做了如下修改：

1、由于 CIFAR-10 数据集的图片大小仅有(32, 32)，最后的平均池化层和全连接层的输入大小仅有(1,1)，因此将平均池化层的 kernel size 改为(1, 1)，第一个全连接层的 in\_features 大小改为 512 \* 1 \* 1；

2、使用迁移学习。VGGNet 中用于提取输入图片特征的 features block 使用在 ImageNet 上的预训练权重，然后冻结其参数，在训练过程中不更新。因此，只需训练 classifier，即最后的全连接层。

本实验中，我使用了 4 个不同层数的 vgg，分别是 vgg11\_bn、vgg13\_bn、vgg16\_bn、vgg19\_bn。为了比较它们的性能差异，使用相同的训练配置，如下：

- 1、训练集分割为 49k/1k, train/val, batch\_size 为 128，总共训练 30 个 epoch；
- 2、使用 Adam 优化器，学习率为 1e-3，weight\_decay 为 1e-4；
- 3、学习率调整策略为 ReduceLROnPlateau，当模型准确率不再提高时，将学习率减小 1/2；

训练过程见代码文件中的训练日志，训练结束后保存训练过程中验证集准确率最高的模型参数权重。

各模型训练过程可视化和测试结果如下：

### 1.3.1 vgg11\_bn

训练时长：8 分 35 秒。

测试结果如下图所示：

```
Total Accuracy: 90.4200%
plane: 93.0000%
car: 92.5000%
bird: 85.7000%
cat: 84.2000%
deer: 89.8000%
dog: 82.4000%
frog: 93.3000%
horse: 93.5000%
ship: 95.9000%
truck: 93.9000%
```

图 1-5: vgg11\_bn 测试结果

训练过程可视化如下图所示：

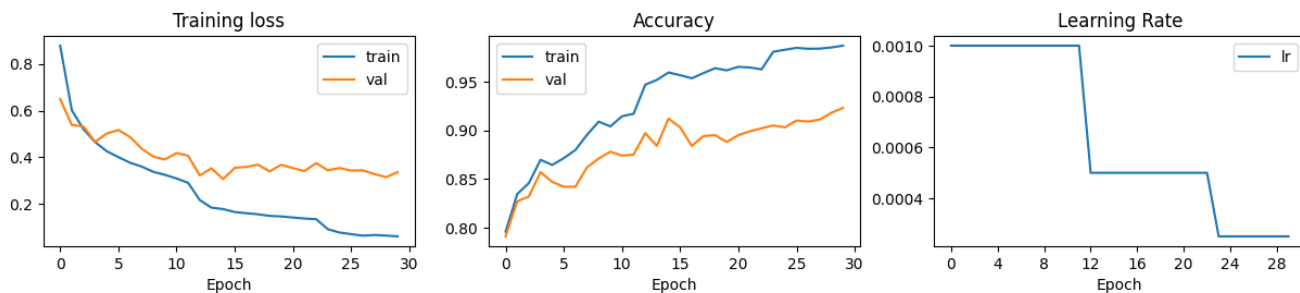


图 1-6: vgg11\_bn 训练过程可视化

### 1.3.2 vgg13\_bn

训练时长: 10 分 18 秒。

测试结果如下图所示:

```
Total Accuracy: 92.1800%
plane: 93.7000%
car: 96.1000%
bird: 91.1000%
cat: 79.5000%
deer: 93.6000%
dog: 87.3000%
frog: 94.7000%
horse: 95.6000%
ship: 95.4000%
truck: 94.8000%
```

图 1-7: vgg13\_bn 测试结果

训练过程可视化如下图所示:

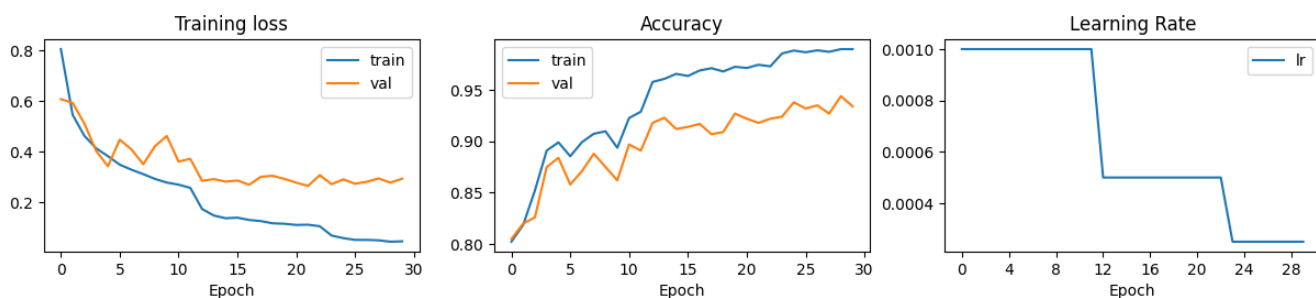


图 1-8: vgg13\_bn 训练过程可视化

### 1.3.3 vgg16\_bn

训练时长: 12 分 26 秒。

测试结果如下图所示:

Total Accuracy: 91.9100%  
plane: 94.9000%  
car: 96.7000%  
bird: 90.4000%  
cat: 81.7000%  
deer: 95.1000%  
dog: 86.7000%  
frog: 93.6000%  
horse: 92.5000%  
ship: 92.5000%  
truck: 95.0000%

图 1-9: vgg16\_bn 测试结果

训练过程可视化如下图所示:

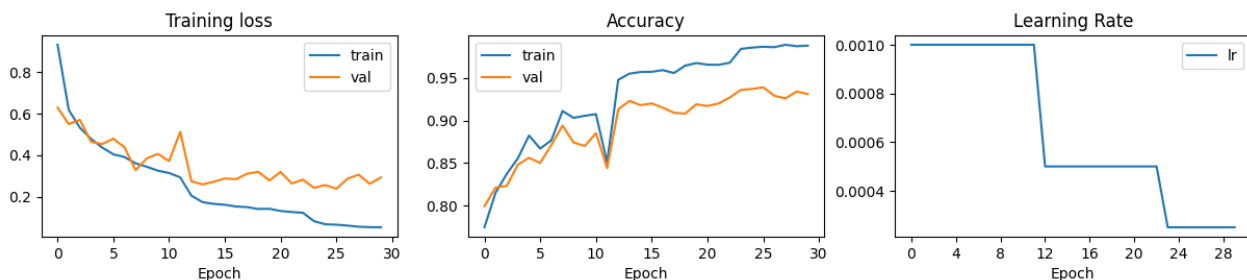


图 1-10: vgg16\_bn 训练过程可视化

### 1.3.4 vgg19\_bn

训练时长: 14 分 35 秒。

测试结果如下图所示:

Total Accuracy: 91.8100%  
plane: 93.1000%  
car: 97.2000%  
bird: 88.4000%  
cat: 85.4000%  
deer: 92.3000%  
dog: 87.1000%  
frog: 91.4000%  
horse: 94.0000%  
ship: 95.0000%  
truck: 94.2000%

图 1-11: vgg19\_bn 测试结果

训练过程可视化如下图所示:

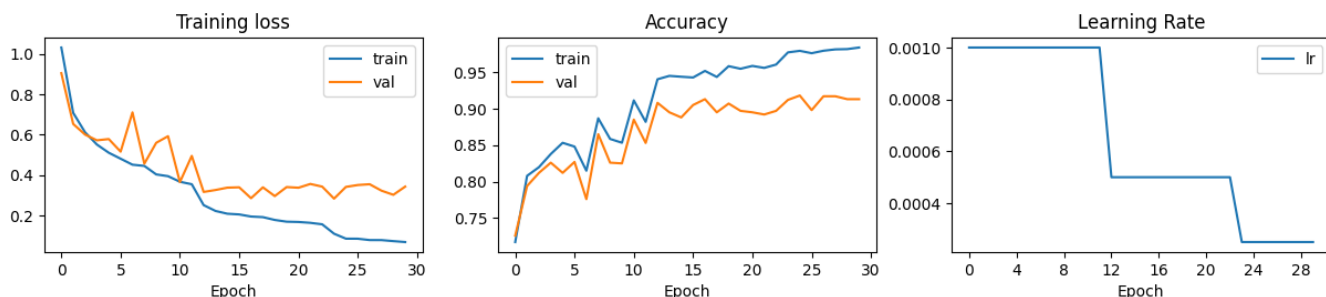


图 1-12: vgg19\_bn 训练过程可视化



### 1.3.5 分析比较

通过训练时长可以看出层数越多，模型参数越多，计算量越大，训练时长越长。

下面将 4 个模型的可视化结果放在一起比较，如下：

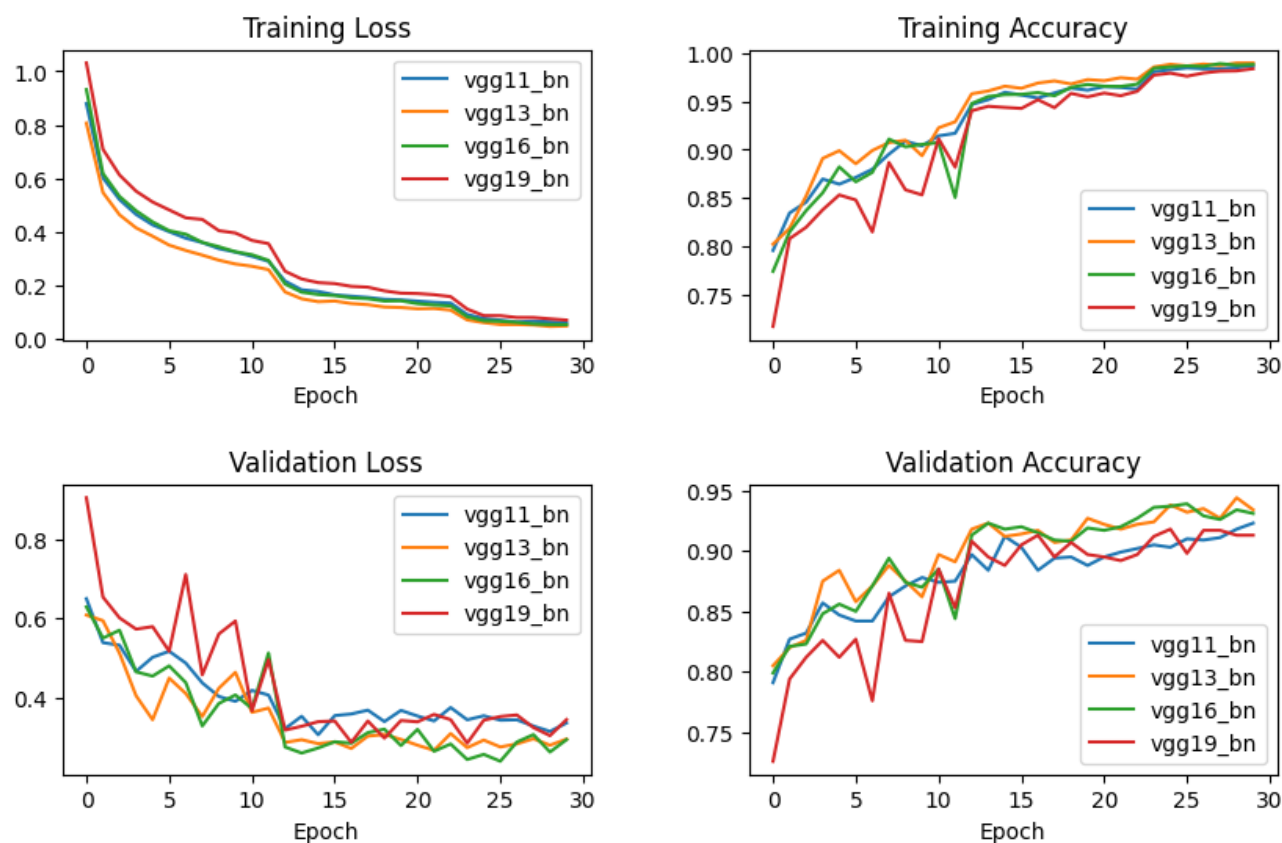


图 1-13: vgg 模型可视化比较

分析：

1、从训练损失图和训练准确率图可以明显看出，每当学习率减小一半时，训练损失会陡然下降，训练准确率也会提高较多，因此曲线呈现阶梯状。但是由于验证集上的波动较大，不太明显，只有大概的趋势和轮廓；

2、前 12 个 epoch 由于学习率较大，模型损失和准确率波动也较大，但是随着训练，学习率逐渐减小，模型逐渐收敛，趋于稳定；

4、由于 4 个模型训练使用的优化器、学习率调整策略等相同，所以收敛速度基本相同；

5、模型并非层数越多越好，本实验中，由上图可以看出，模型损失和准确率由好到坏排序，大致如下：vgg13\_bn、vgg16\_bn、vgg19\_bn、vgg11\_bn。由于相差不大，所以此排序不一定准确。当然，此结果仅限于本实验中上述的训练配置下，是否具有普适性还需进一步实验探究。

## 1.4 ResNet

本实验中，我还使用了 ResNet。模型的代码实现见 `modules\resnet.py`，参考 PyTorch 源码。为了更好的适配 CIFAR-10 数据集、提高训练效率，我做了如下修改：

1、由于 CIFAR-10 数据集的图片大小仅有(32, 32)，第一个卷积层改为 `kernel_size=3、stride=1、padding=1`，其后的第一个 maxpool 层改为 `kernel_size=2、stride=2`；

2、使用迁移学习。ResNet 中的所有 building block 使用在 ImageNet 上的预训练权重，然后冻结其参数，在训练过程中不更新。

本实验中，我使用了 3 个不同层数的 resnet，分别是 resnet18、resnet34、resnet50。为了比较它们的性能差异，使用相同的训练配置，如下：

- 1、训练集分割为 49k/1k, train/val, batch\_size 为 128, 总共训练 30 个 epoch;
- 2、使用 Adam 优化器，学习率为 1e-3, weight\_decay 为 1e-4;
- 3、学习率调整策略为 ReduceLROnPlateau，当模型准确率不再提高时，将学习率减小 1/2;

训练过程见代码文件中的训练日志，训练结束后保存训练过程中验证集准确率最高的模型参数权重。

各模型训练过程可视化和测试结果如下：

### 1.4.1 resnet18

训练时长：28 分 19 秒。

测试结果如下图所示：

```
Total Accuracy: 90.7600%
plane: 93.2000%
car: 95.2000%
bird: 84.5000%
cat: 76.3000%
deer: 91.8000%
dog: 89.6000%
frog: 93.5000%
horse: 93.9000%
ship: 95.6000%
truck: 94.0000%
```

图 1-14: resnet18 测试结果

训练过程可视化如下图所示：

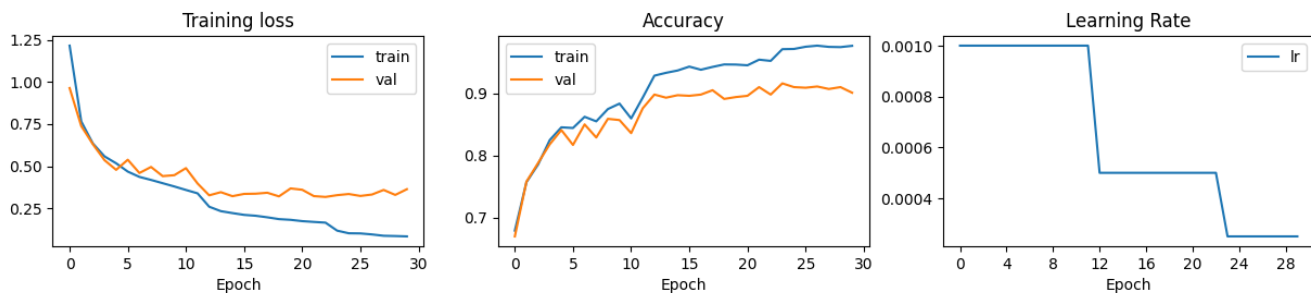


图 1-15: resnet18 训练过程可视化

## 1.4.2 resnet34

训练时长: 40 分 46 秒。

测试结果如下图所示:

```
Total Accuracy: 90.1300%
plane: 91.9000%
car: 95.8000%
bird: 84.4000%
cat: 77.5000%
deer: 93.3000%
dog: 83.2000%
frog: 92.6000%
horse: 94.2000%
ship: 95.0000%
truck: 93.4000%
```

图 1-16: resnet34 测试结果

训练过程可视化如下图所示:

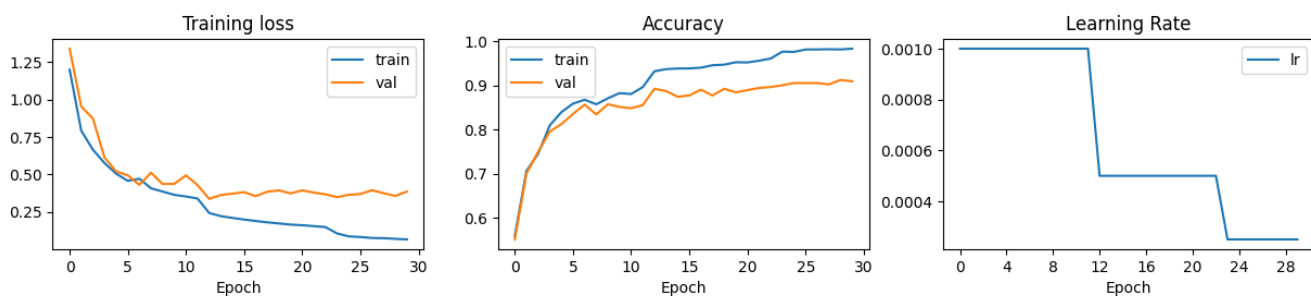


图 1-17: resnet34 训练过程可视化

## 1.4.3 resnet50

训练时长: 46 分 9 秒。

测试结果如下图所示:

```

Total Accuracy: 91.4100%
plane: 93.2000%
car: 95.5000%
bird: 88.2000%
cat: 80.5000%
deer: 92.3000%
dog: 85.5000%
frog: 96.2000%
horse: 92.7000%
ship: 95.6000%
truck: 94.4000%

```

图 1-18: resnet50 测试结果

训练过程可视化如下图所示:

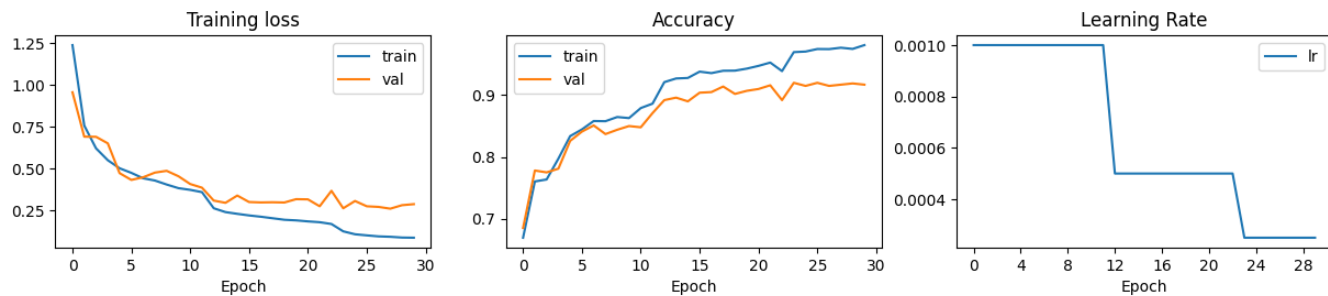


图 1-19: resnet50 训练过程可视化

## 1.4.4 分析比较

由于 3 个模型是在单 GPU 下同时训练的, 而 vgg 的 4 个模型是分别单独在单 GPU 下训练的, 所以训练时长比 vgg 要长许多。

通过训练时长可以看出层数越多, 模型参数越多, 计算量越大, 训练时长越长。

下面将 4 个模型的可视化结果放在一起比较, 如下:

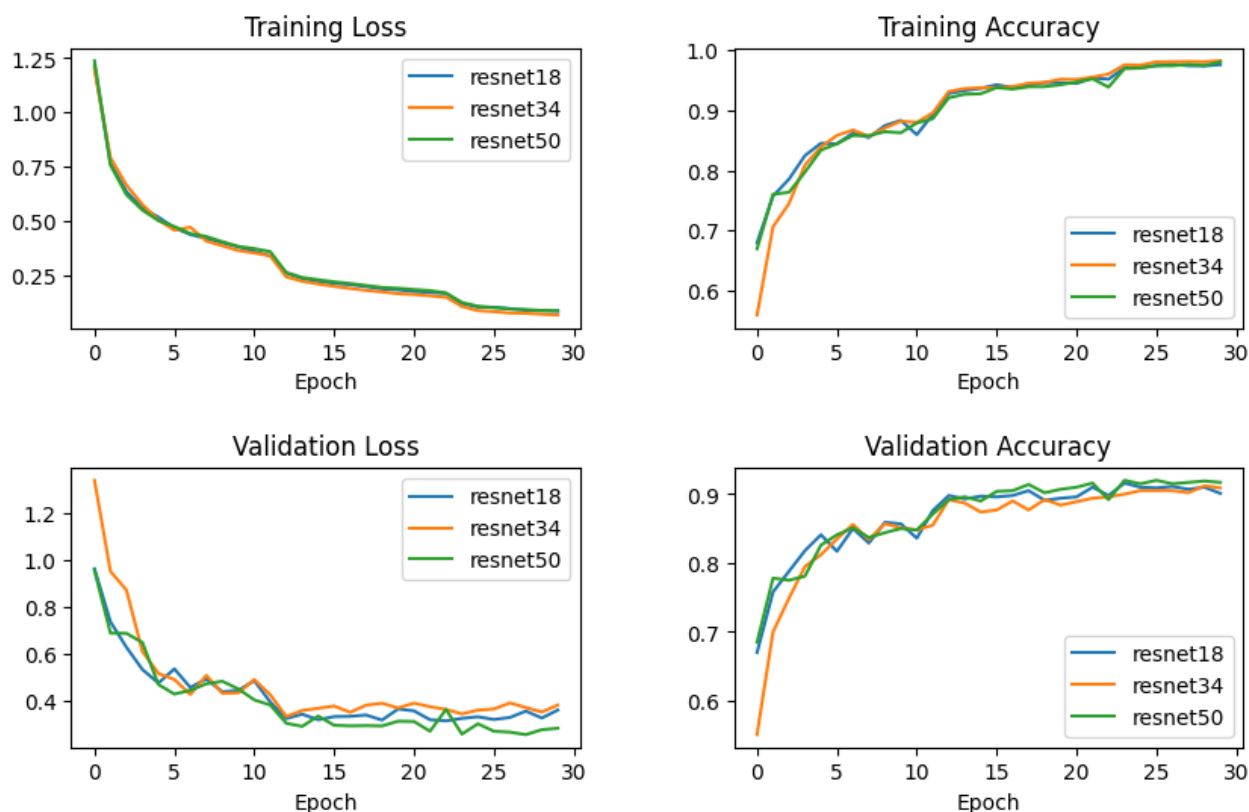


图 1-20: resnet 模型可视化比较

分析:

1、从训练损失图和训练准确率图可以明显看出，每当学习率减小一半时，训练损失会陡然下降，训练准确率也会提高较多，因此曲线呈现阶梯状。但是由于验证集上的波动较大，不太明显，只有大概的趋势和轮廓；

2、前 12 个 epoch 由于学习率较大，模型损失和准确率波动也较大，但是随着训练，学习率逐渐减小，模型逐渐收敛，趋于稳定；

4、由于 3 个模型训练使用的优化器、学习率调整策略等相同，所以收敛速度基本相同；

5、模型并非层数越多越好，本实验中，由上图可以看出，模型损失和准确率由好到坏排序，大致如下：resnet50、resnet18、resnet34。由于相差不大，所以此排序不一定准确。当然，此结果仅限于本实验中上述的训练配置下，是否具有普适性还需进一步实验探究。

由于时间有限，为了训练效率，训练不够充分，因此未能复现 ResNet 原论文中的结果，后续有待进一步实验。