

CNN实验报告

小组成员：

周昱宏 519021910647

钱博闻 519021910581

前言

本篇实验报告按照实际模型训练的过程以及提高准确率进行的所有尝试进行阐述，包括模型设计、参数调节、图像预处理。最终我们得到的模型在测试集上的精度达到了89.48%

模型设计

初选模型

在进行正式训练之前，我们参照官网的例子进行了简单的调参，但是最后的结果只达到70%，于是我们决定参考现有的一些模型。

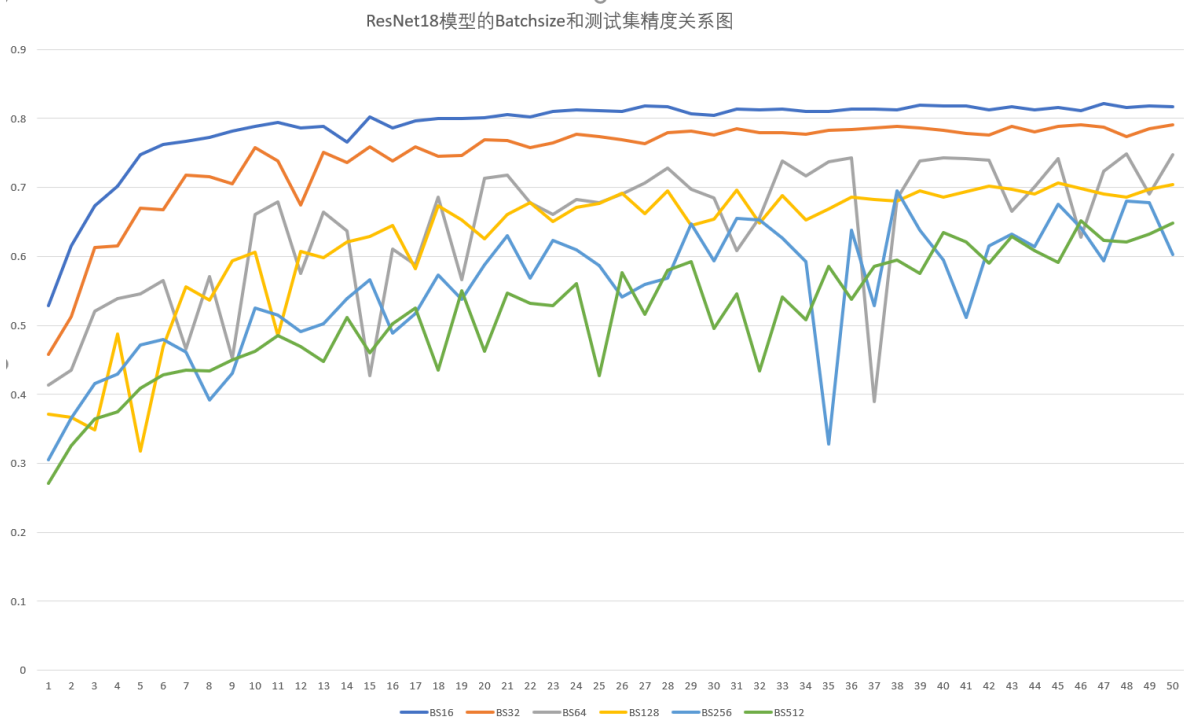
我们选择了**ResNet 18-layer**模型，首先是因为ResNet相比其他神经网络有更深的层数，残差网络在图像分类上有很好的表现；其次是因为ResNet在解决梯度消失和过拟合方面有着优秀的表现。参照resnet 18的论文，我们自己复现了ResNet-18模型。

Batch Size的选择

根据Linear scaling rule，就是说在参数调节时，learning rate和Batch size的比值为定值。对于一个固定的学习率，存在一个最优的Batch size能够最大化测试精度。因此我们固定学习率为0.005，测试不同batch size对于模型准确率的影响。

BatchSize	测试集精度
16	82.17%
32	79.12%
64	74.91%
128	70.60%
256	69.49%
512	65.17%

模型在CIFAR-10数据集的准确率随着Batch size的下降而下降。如下图所示。一般来说，比较大的Batch size比较利于机器学习的效率，节省程序运行时间，并且可以更好的利用内存。



现这种情况，**不是因为过拟合**。Batch size较小的时候，在梯度下降时，能够更容易走出鞍点，防止更新停止的情况出现。此外，较小的Batch size会在梯度下降时，收到噪声的影响，从而跳出Loss损失函数的Sharp Minima，最终收敛到Flat Minima。

因此，我们最终选择了Batch size 16作为参数。

但是我们发现此时的模型在训练集准确率接近于1时，测试集的准确率发生剧烈波动。这一方面可能是因为learning rate不合理导致在极小值点附近产生震荡，另一方面准确率不够高说明我们的模型有很大的提升空间。

改进模型

在改进模型之前，我们需要先了解resnet-18模型的结构

图 1

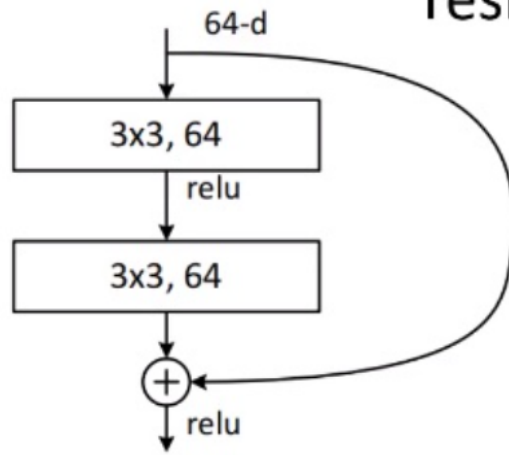
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

上图来自论文，只需关注18-layer这一列即可。

还有一个很重要的概念是Basic Block,是resnet 的residual 模块，也是核心所在。

图2

residual结构



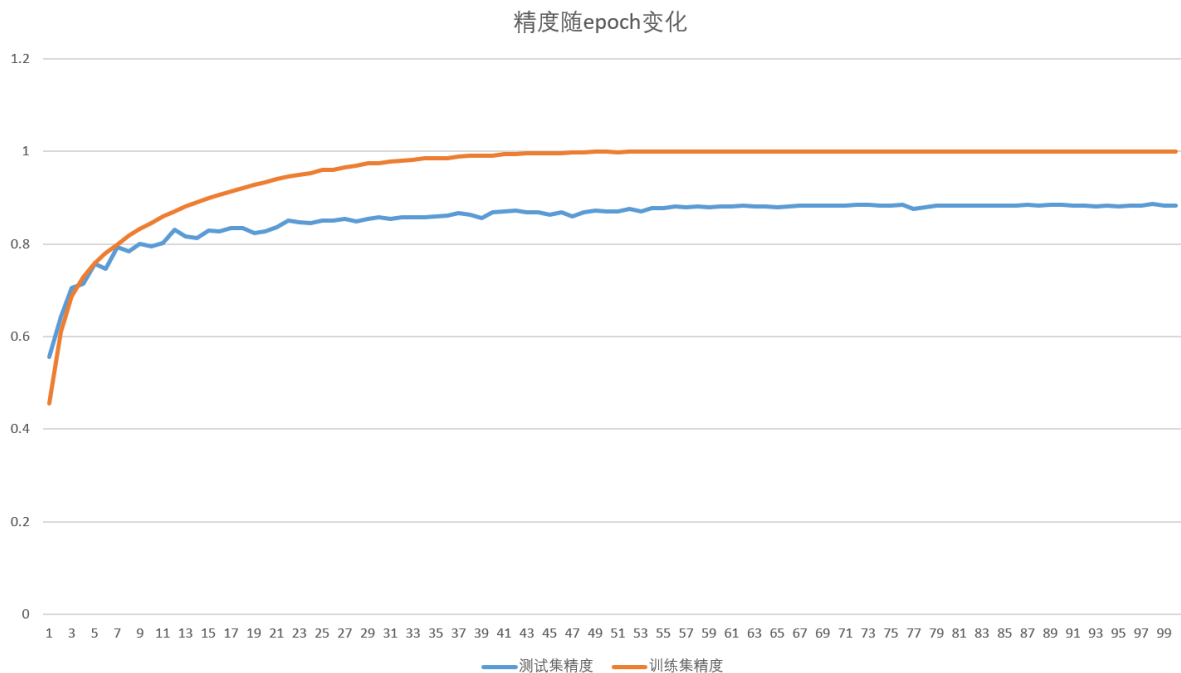
上图conv2x-conv5x分别对应模型中的layer1-layer4，每个layer都由若干个block组成，在resnet18中layer1-layer4的block数都是2，用[2,2,2,2]表示。

针对CIFAR-10测试集的特点（小，像素点少），为了尽量保留原始图像的信息，我的改进尝试主要有：

- ①更改每个layer的步长stride
- ②更改Basic Block的结构
- ③更改每个layer中Basic Block的数量

论文中的模型是用于224x224大小的图片，因此肯定不适合CIFAR-10集上32x32大小的图片。如果直接使用其模型会造成大量特征的流失，因此，为了尽可能保留图像的原始特征，我先是把layer1到layer4的stride由最初的[1,2,2,2]换成[1,1,1,1]，即每个layer的步长都调成1.得到的结果如下

图三



测试集精度最高达到了**88.56%**

可见效果显著，在这个基础上，我尝试更改block的结构。

如图2所示，原版的Basic Block由两个3x3的卷积组成，用[3x3,3x3]表示，我分别做了将第一个卷积核改成1x1,以及增加一个1x1的卷积核的修改，结果如下

Block 组成	测试集精度
[3X3,3X3]	88.56%
[1X1,3X3]	85.88%
[1X1,3X3,3X3]	87.88%

我们将 3×3 的卷积核改为 1×1 的大小，希望能够提取出更多的特征，但是准确率只有85.880%。这可能是因为卷积核过小时，反而会引入一些多余的特征，或是将噪声当成了特征，导致过拟合的出现。

参照上述结果，仍旧采用原Basic Block

接下来，我们尝试更改每层的Block Num,同时，为了降低由于扩大Block Num的影响，同时调layer4的步长。模型设置以及测试集精度如下：

模型设置	测试集精度
[2-3-3-1]layer4步长为1	87.6%
[2-3-3-2]layer4步长为1	88.4%
[2-3-3-3]layer4步长为1	88.05%
[2-3-3-3]layer4步长为2	89.270%

结果发现，layer1包含2个BasicBlock,layer2包含3个Basic Block,layer3包含3个BasicBlock，layer4包含3个BasicBlock且步长为2的模型效果最好。

进一步优化

学习率优化

我们针对不同学习率下模型的准确率进行测试；此外，我们还测试了learning rate decay时的模型准确率，即学习率按照一定规律变化。结果如下：

学习率	测试集精度
0.004	88.340%
0.005	89.270%
0.006	89.480%
0.007	89.350%
从0.01开始，每5个epoch阶梯状衰减	87.150%
0.005开始，指数衰减	85.350%

我们使用了学习率衰减，希望梯度下降时能迅速收敛到局部最小值，防止震荡的出现，并且无限接近最小值。意外的是，学习率下降并没有使模型得到更高的准确率。我们分析，出现这种情况，**可能是因为学习率衰减过快**，导致梯度下降至不太好的局部最小值或是鞍点，无法跳出这些区域。

图像预处理

图像预处理采用的方法有：增加饱和度、对比度，resize（由于显存限制，没有resize较大），灰度，归一化，水平翻转

结果如下（注意下面的图像都进行过翻转增加训练集样本的预处理）

图像预处理方式	测试集精度
增加色彩饱和度、对比度	87.510%
归一化预处理	89.320%
灰度处理	87.380%
增加色彩饱和度、对比度同时进行归一化	86.880%
resize到35X35	88.360%
resize到40X40	88.610%

可以看到，对图像进行这些常用的预处理效果都不太好，因此图像预处理仅仅采用水平翻转增加样本的方法。可能的解释是上述模型的改进已经十分契合原本图像的特征，因此在进行图像预处理时效果反而变差。

实验结果

我们模型经过调参，最终的准确率为**89.480%**。具体的模型设置、参数处理参见上文。

由于我们的模型过大，训练及调参时间消耗大，以及计算机资源的限制，无法做到进一步准确率的提高。

最终的代码及调参的各个数据也放在文件夹中。