|  |
| --- |
| 电子科技大学  **计算机专业类课程** |
| **实验报告** |
| **课程名称：人工智能综合实验II**  **学院专业：计算机科学与工程学院**  **学生姓名：崔若晨**  **学　　号：2021060902008**  **指导教师：石小爽** |
| **日　　期： 2023 年 9 月 10 日** |
|  |
| 电子科技大学计算机学院实验中心 |

**电 子 科 技 大 学**

**实 验 报 告**

**实验二**

# 一、实验室名称：

电子科技大学清水河校区主楼A2-413-1

# 二、实验项目名称：

人工智能实验II-2：基于卷积神经网络图像分类算法实现

# 三、实验原理：

卷积网络在本质上是一种输入到输出的映射，它能够学习大量的输入与输出之间的映射关系，它主要被用来识别位移、缩放及其他形式扭曲不变性的二维图像。由于CNN的特征检测层通过训练数据进行学习，所以在使用CNN时，避免了显式的特征抽取，而隐式地从训练数据中进行学习；再者由于同一特征映射面上的神经元权值相同，所以网络可以并行学习，这也是卷积网络相对于神经元彼此相连网络的一大优势。卷积神经网络以其局部权值共享的特殊结构在语音识别和图像处理方面有着独特的优越性，其布局更接近于实际的生物神经网络，权值共享降低了网络的复杂性，特别是多维输入向量的图像可以直接输入网络这一特点避免了特征提取和分类过程中数据重建的复杂度。

# 四、实验目的：

通过Pytorch框架搭建LeNet网络，实现在CIFAR-10数据集上的图像分类，并以此掌握：

1. 卷积神经网络中卷积层、池化层、全连接层的使用场景和作用以及它们参数的具体含义。
2. 了解Pytorch框架下模型的训练流程：训练、测试以及评估方法，并亲自代码实践。
3. 实验结果的分析能力，包括两个方面：量化分析与可视化分析。

# 五、实验内容：

1. 问题描述

基于卷积神经网络图像实现分类算法

1. 算法的概要设计与分析

网络采用了卷积算法，池化层算法，Batch Normalization，MLP模型，BP算法等，通过卷积和池化提取图片的特征，然后用MLP对其进行分类预测，并通过BP算法更新网络权重和偏置，实现网络的优化。

1. 核心算法的详细设计与实现

**卷积算法：**卷积算法是构建卷积神经网络的核心。输入层每个通道卷积范围内的每个待征值乘上一个卷积核对应通道的对应权重再全部求和，等于输出层一个通道上的一个特征值。直观地来说，卷积运算会让滤波器学习到某些类型的视觉特征，具体的视觉特征可能是某些方位上的边界，或者在第一层上某些颜色的斑点，甚至可以是网络更高层上的蜂巢状或者车轮状图案。卷积层可以被看做是神经元的一个输出。神经元只观察输入数据中的一小部分，并且和空间上左右两边的所有神经元共享参数。

**池化算法**：池化算法分为最大池化(Maxpooling)和平均池化(Avaragepooling)。最大池化是取输入层在池化核对应权重的最大值，而平均池化则是去对应范围内所有权重的平均值。通常在连续的卷积层之间会周期性地插入一个池化层。它的作用是对图片进行下采样操作，逐渐降低数据体的空间尺寸，这样的话就能减少网络中参数的数量，使得计算资源耗费变少，也能有效控制过拟合。

**全连接层（MLP）：**对于任一个卷积层，都存在一个能实现和它一样的前向传播函数的全连接层。该全连接层的权重是一个巨大的矩阵，除了某些特定块（感受野），其余部分都是零。而在非 0 部分中，大部分元素都是相等的（权值共享）。在全连接层中，我们将最后一个卷积层的输出展平，并将当前层的每个节点与下一层的另一个节点连接起来。它的功能是为了对提取的特征进行分类。

Batch Normalization（BN）：对每一个输入的mini-batch，计算batch的均值和标准差，对batch中的每一个元素作标准化(是一个很小的数，避免由分母等于0带来的系统错误)。BN算法调整了每层网络输出数据的分布，使其进入激活函数的作用区（原点附近的区域），梯度弥散率低，区分率高。同时，BN会在训练过程中，自己调节数据分布，使其“更合理”地进入激活函数。

# 六、实验器材（设备、元器件）：

硬件平台：PC端

开发环境：操作系统Windows11

测试环境： Pytorch

# 七、实验步骤：

1. 网络框架结构搭建介绍

设计的卷积神经网络主体框架如图 1。

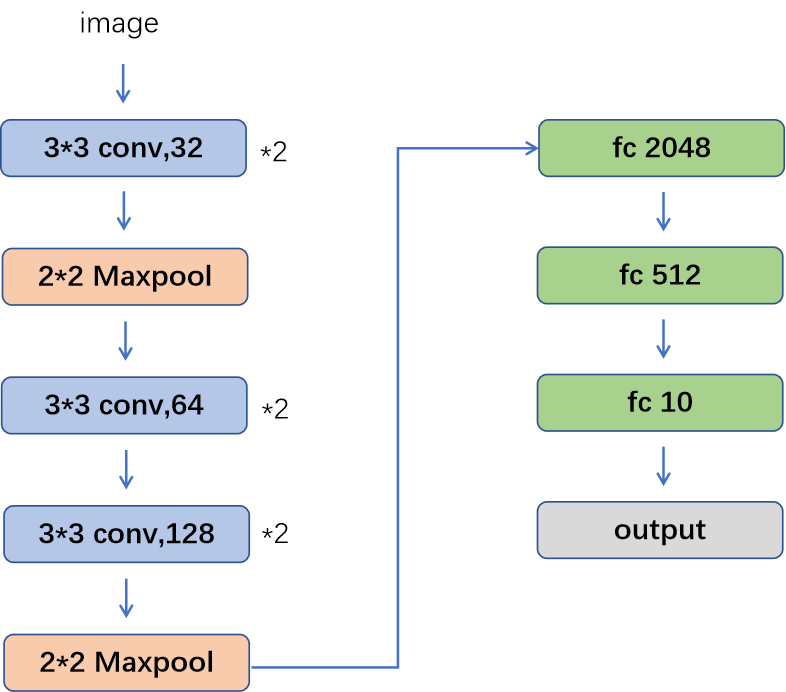


图 1 网络主体框架

网络采用了3个卷积核为33的卷积操作

1. 核心代码介绍

|  |
| --- |
| **输入：**训练集CIFAR10  卷积神经网络net  Adam优化器optim  CrossEntropy损失函数loss\_fun  归一化函数Softmax函数  **过程：**   1. 将数据集传入DataLoader并设置batchsize 2. **for** epoch 1 to 30 3. 从DataLoader中取出img,target 4. 将img传入网络net得到output 5. lossloss\_fun(output,target) 6. accuracy 7. BP算法更新net的权重w和偏置b 8. **end fo**r   **输出：**loss,accuracy,折线图 |

1. 训练执行流程介绍

总共训练30个epoch。每次训练都计算训练集上的损失，训练集上的正确率，测试集上的正确率并通过tensorboard做出相应的折线图。根据训练结果对网络进行一些调整（如卷积核，池化核的大小等）

# 八、实验数据及结果分析：

1. 实验数据

CIFAR-10 是一个包含60000张图片的数据集。如图 2，其中每张照片为32\*32的彩色照片，每个像素点包括RGB三个数值，数值范围 0 ~ 255。所有照片分属10个不同的类别，分别是 'airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck'。其中训练集有50000张图片，测试集有10000张图片。

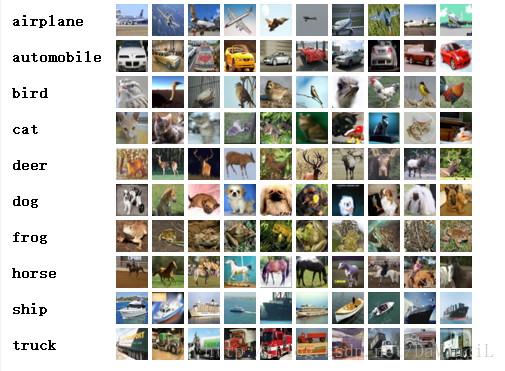


图 2 CIFAR10数据集

1. 结果分析：
   1. 量化展示随着训练迭代数目变化准确率以及其变化曲线

图 3为随着训练迭代数目变化，神经网络在训练集和测试集上的正确率，总共迭代epoch为30。从图中可以看到，网络在训练集上的正确率随着迭代次数的增加增长迅速，并最终达到95%左右。而在测试集上的增速比训练集缓慢，且在epoch=20后振荡，无明显增加，最高正确率为85%左右，比训练集的正确率低很多。原因大概是随着训练迭代次数的增加，网络出现了过拟合，这一点在下文展示的网络分别在训练集和测试集上的损失（图 19,图 20）可以更清晰的看到。

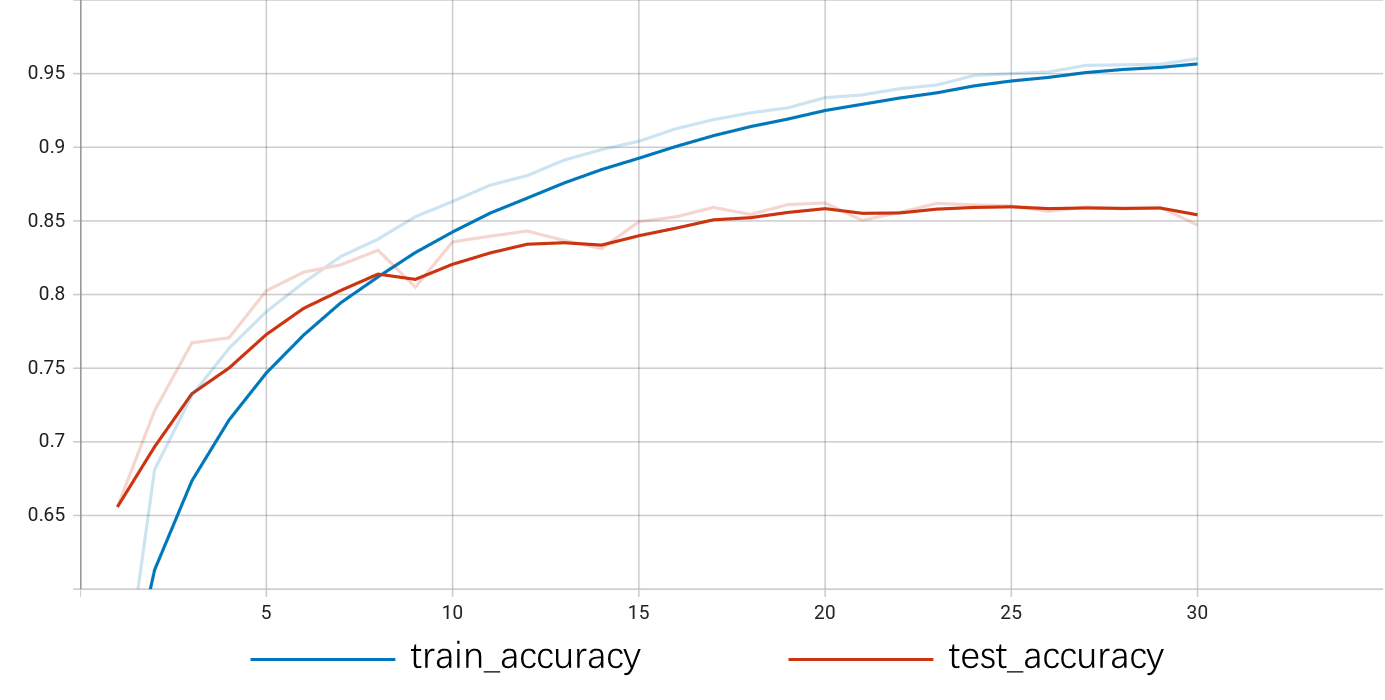


图 3训练30个epoch的正确率

1. 量化展示随着训练数据集大小对于准确率的影响极其变化曲线

图 4展示了模型在不同大小数据集训练和测试的正确率。数据集的大小有训练集和测试集各5k,10k,20k（由于测试集一共只有10k张图片，因此此时取测试集全集10k即可）以及全集50k张图片的训练集和10k张图片的测试集。

|  |  |
| --- | --- |
|  |  |

图 4不同数据集大小对正确率的影响

从图中可以看出，4种大小训练集的正确率整体都呈上升趋势，开始时正确率由低到高依次为5k,10k,20k,fullset。在epoch=30的时候，4者的正确率几乎相当。4种大小测试集的正确率也整体都呈上升趋势，在整个测试过程中，正确率由低到高依次为5k,10k,20k,fullset，且彼此之间差异较大并都低于训练集的正确率。原因是因为训练集数据量越小越容易出现过拟合现象，模型泛化能力较差，在训练集上的表现较差。因而增加数据集大小是一种常见的预防过拟合的方法。

1. 损失下降曲线，是否出现过拟合或者欠拟合现象，考虑如何解决

图 5，图 6为网络分别在训练集和测试集上的损失，可以看到随着训练迭代次数的增加，网络在训练集上的损失整体呈现下降趋势，与网路在训练集上的正确率（图 17）变化相吻合。而网络在测试集上的损失在epoch=0~20时呈下降趋势，在epoch=20后有较明显的上升。这同样与网络在训练集上的正确率（图 17）变化相吻合。损失在测试集的上升与网络在测试集上的正确率波动共同表明网络训练出现了过拟合，进一步验证了上文的猜想。

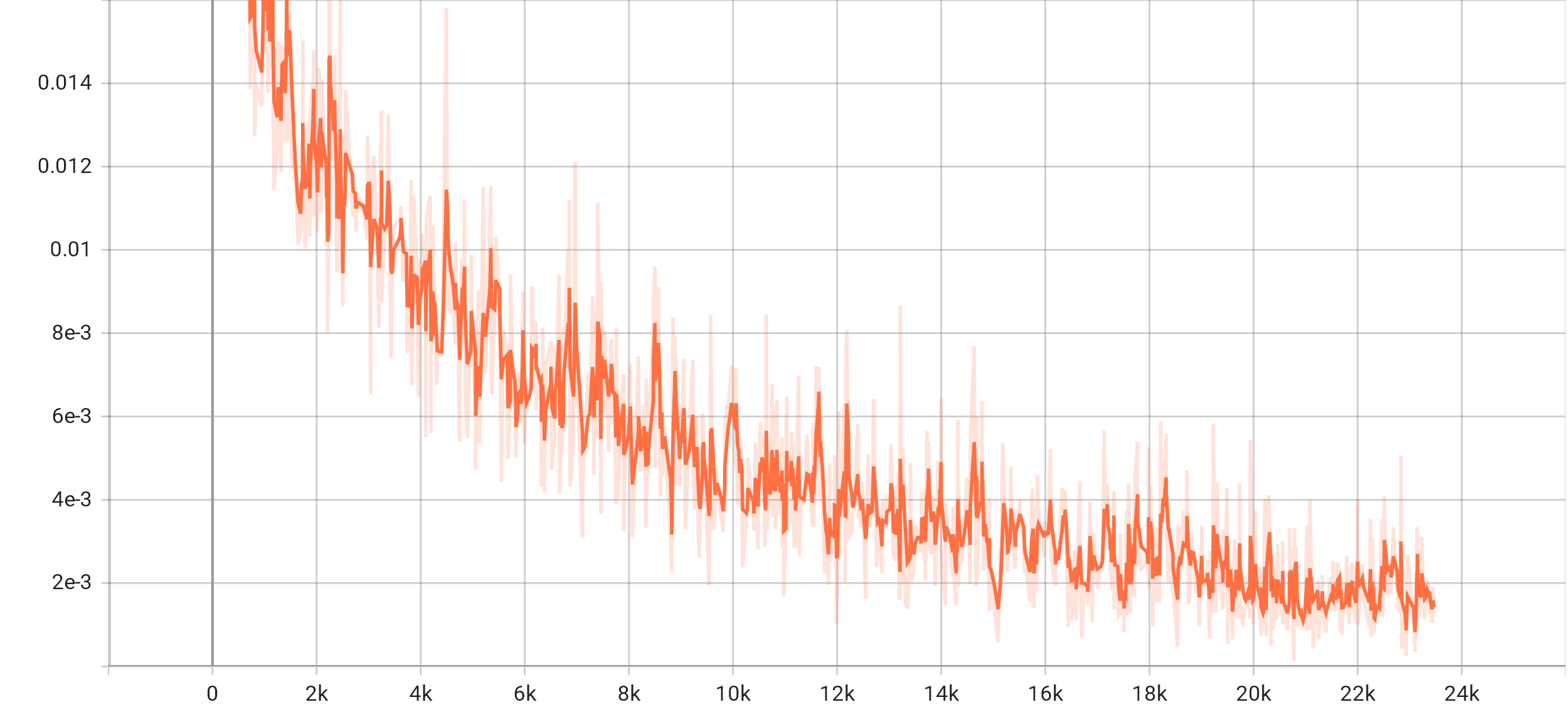


图 5 training loss

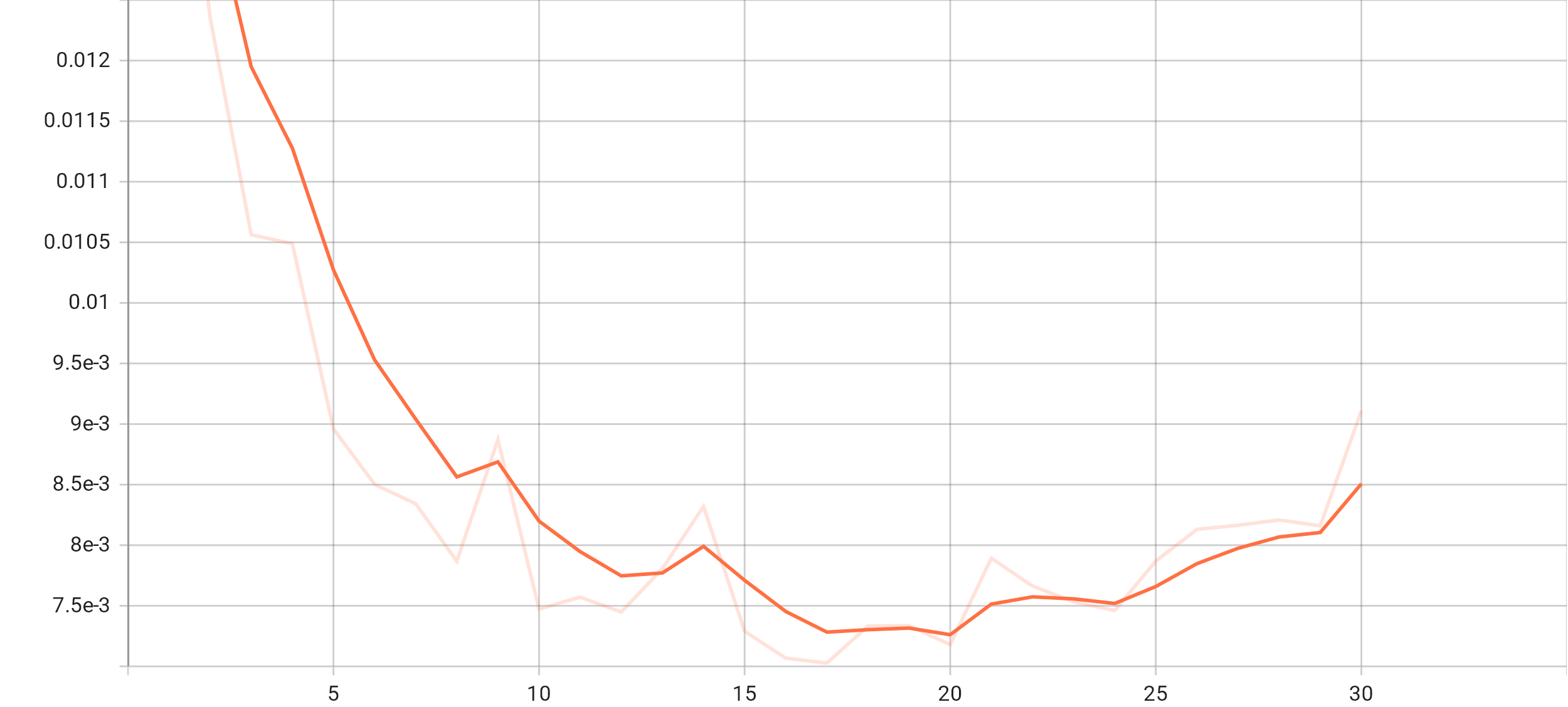


图 6 test loss

常见的解决过拟合方法如下：

正则化：正则化是机器学习中最常见的过拟合解决方法，在损失函数中加入正则项来惩罚模型的参数，以此来降低模型的复杂度，常见的添加正则项的正则化技术有L1，L2正则化。L1正则化是基于L1范数的，在损失函数中加入L1范数修正项，修正后的损失函数为。L2正则化是基于L2范数的，损失函数为。

1. Batch Normalization：BM算法是一种非常有用的正则化方法，可以让大型的卷积神经网络快速收敛，同时还能提高分类的准确率。BM算法将每一层的输入值做归一化处理，并且会重构归一化处理之后的数据，确保数据的分布不会发生变化。
2. Dropout是深度学习中最常用的控制过拟合的方法，主要用在全连接层。Dropout方法是在一定的概率上（通常设置为0.5，原因是此时随机生成的网络结构最多）隐式的去除网络中的神经元，提高网络的泛化能力。
3. 可视化分析实验结果，错误分析

选取表现最佳的模型（测试集正确率为87.92%）分别在10个类别上做测试，其结果由高到低如表 1，并进行可视化，如图 7所示。模型分别在10个类别上的准确率

表 1模型分别在10个类别上的准确率

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类别 | automobile | frog | airplane | truck | ship |
| 正确率 | 94.5% | 93.0% | 92.2% | 91.6% | 91.5% |

续表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类别 | horse | deer | dog | bird | cat |
| 正确率 | 89.9% | 86.4% | 82.0% | 81.6% | 76.5% |

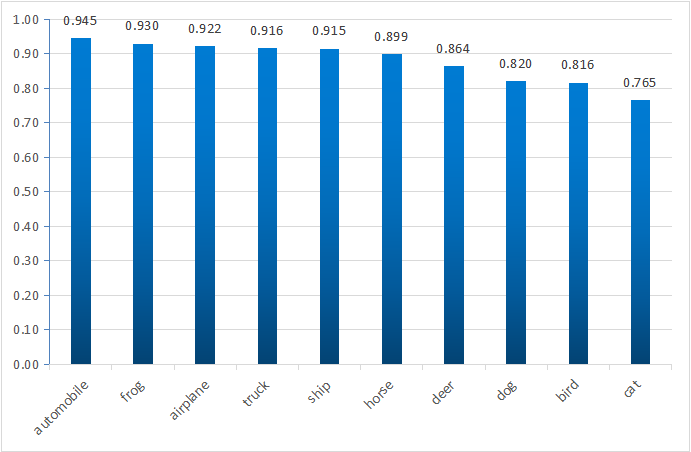


图 7 模型分别在10个类别上的准确率

表现最好的类别为automobile，表现最差的类别为cat。大体上在人造物体的表现优于自然生物的表现。这大概是因为自然生物相比人造物体，亚类更多，形态差异性更大，干扰更大，学习难度更大的缘故。

**拓展实验：基于Resnet实现图像分类**

残差网络（Resnet）的主要创新是在卷积网络中增加了直连通道，此前的网络结构是性能输入做一个非线性变换，而Resnet则允许保留之前网络层的一定比例的输出即允许原始输入信息直接传到后面的层中。这样的话这一层的神经网络可以不用学习整个的输出，而是学习上一个网络输出的残差（如图 8）

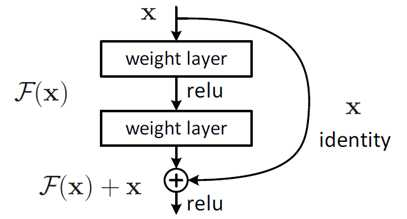


图 8 Resnet残差学习模块

将残差模块堆叠在一起实现不同层数的Resnet，常见的有Resnet18，Resnet50，Resnet101等。本次实验选择了Resnet18，网络框架结果如图 9。由于使用的CIFAR10数据集，因而在原始Resnet18做了部分调整，包括输入图片尺寸为3232，输出为10个特征值。

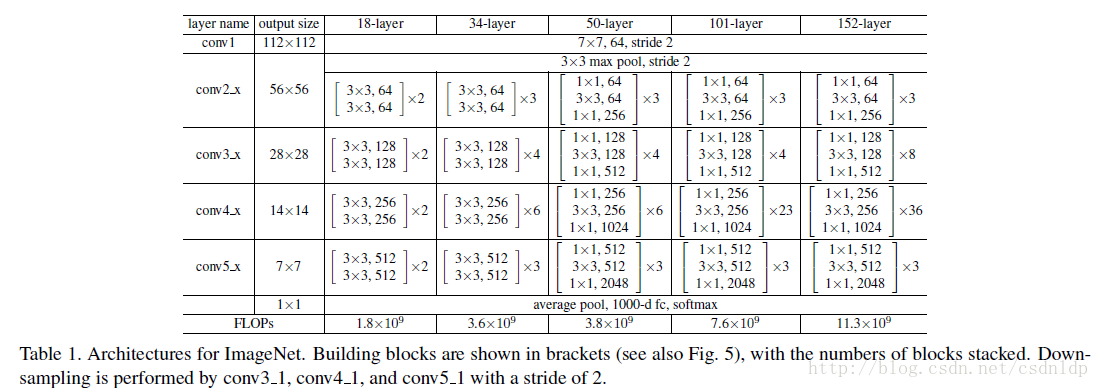


图 9 Resnet不同层数的网络架构

将搭建好的Resnet18在CIFAR10数据集上训练30个epoch，并做出网络在训练集和测试集上的正确率。同时将结果与之前搭建的CNN网络以及pytorch提供的训练好的Resnet18进行比较，做出折线图如图 10。其中Resnet(tr)表示pytorch提供的训练好的Resnet18网络，Resnet(ut)表示自己搭建的Resnet18网络，Mynet表示上文搭建的CNN网络。三者在训练集上的正确率由高到低依次为Resnet(tr)，Mynet，Resnet(ut)。三者最终在测试集上的表现由高到低为Mynet，Resnet(tr)，Resnet(ut)。分析原因大概是因为训练epoch太少，Resnet网络较深，优化较慢，并没有优化到一个特别好的效果。另外，Resnet标准输入的图片尺寸为224224，而本实验使用的图片尺寸为3232，图片尺寸较小，对于较深的网络并不是特别合适。

|  |  |
| --- | --- |
|  |  |

图 10 3种网络分别在训练集和测试集上的正确率

# 九、总结及心得体会：

通过本次实验，我对卷积神经网络的工作原理与实现有了很好的了解。

卷积神经网络与普通神经网络的区别在于，卷积神经网络包含了一个由卷积层和子采样层（池化层）构成的特征抽取器。在卷积神经网络的卷积层中，一个神经元只与部分邻层神经元连接。在CNN的一个卷积层中，通常包含若干个特征图(featureMap)，每个特征图由一些矩形排列的的神经元组成，同一特征图的神经元共享权值，这里共享的权值就是卷积核。卷积核一般以随机小数矩阵的形式初始化，在网络的训练过程中卷积核将学习得到合理的权值。共享权值（卷积核）带来的直接好处是减少网络各层之间的连接，同时又降低了过拟合的风险。子采样也叫做池化（pooling），通常有均值子采样（mean pooling）和最大值子采样（max pooling）两种形式。子采样可以看作一种特殊的卷积过程。卷积和子采样大大简化了模型复杂度，减少了模型的参数。 卷积神经网络（CNN）具有共享卷积核，对高维数据处理无压力，通过卷积运算建立了局部连接的稀疏交互网络，相比于全连接前馈神经网络少了很多参数。卷积神经网络无需手动选取特征，训练好权重，即得特征分类效果好的优点，除此之外，卷积网络还具有平移不变性，平移不变性意味着系统产生完全相同的响应（输出），不管它的输入是如何平移的。

深度学习网络在网络很深的时候，效果却越来越差了。其中的原因之一即是网络越深，梯度消失的现象就越来越明显。对于深度卷积神经网络，由于每做一次卷积（包括对应的激活操作）都会浪费掉一些信息：比如卷积核参数的随机性（盲目性）、激活函数的抑制作用等等。ResNet通过shortcut把以前处理过的信息直接再拿到现在一并处理，起到了减损的效果。因而非常深的残差网络能够很容易的优化，能够容易地从增加的深度中得到精度收益，比先前的网络产生了更好的效果。Resnet使信息更容易在各层之间流动，包括在前向传播时提供特征重用，在反向传播时缓解梯度信号消失，它没有带来额外的参数和计算开销，便于和具有相同结构的“平常”网络进行对比。

通过本次实验，我对python基本语法更加熟悉，对pytorch框架更加熟悉。在实验一的基础上，我能够进一步较熟练地使用pytorch实现卷积操作，池化操作，搭建基本的卷积网络框架。

# 十、对本实验过程及方法、手段的改进建议及展望：

无

**报告评分：**

**指导教师签字：**