HW#3：Learning CNN实验报告

**实验环境：**

Python 3.7

Torch 1.8.1

Torchvision 0.8.1

**任务描述：**

利用CNN进行手写数字识别与物体分类。

1. 实现最基本的卷积神经网络LeNet-5以及一个物体分类的CNN（AlexNet），调用 PyTorch深度学习开发工具中各种构建函数来进行相关训练。

2. 自己用MNIST手写数字数据集（0-9一共十个数字）6万样本实现对LeNet-5的训练，数据集通过torchvision进行下载。对MNIST的1万测试样本进行测试，获得识别率是多少。

3. 自己用CIFAR-10数据库(数据集通过torchvision进行下载)实现CNN物体分类功能的训练与测试。

**实验过程：**

1. LeNet

LeNet包括了两个卷积层以及对应的激活层ReLu和池化层maxpool，和两个全连接层，LeNet的神经网络整体结构图如下所示：

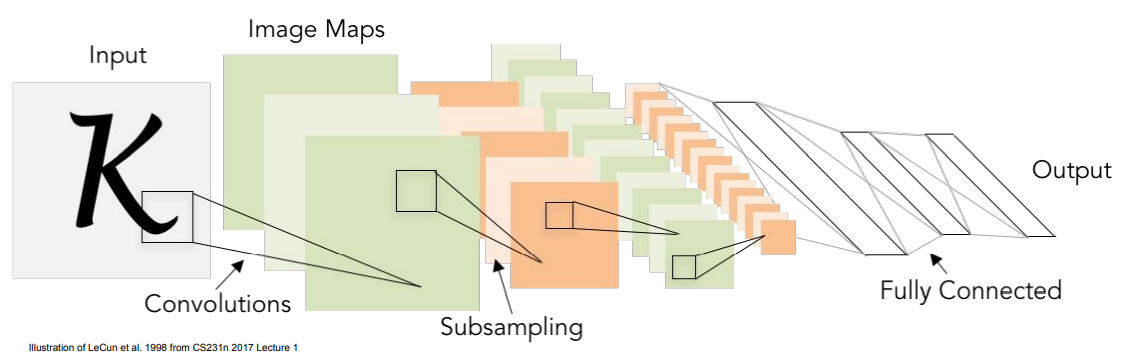


图1 LeNet网路结构

根据LeNet的定义，我利用torch框架定义该神经网络，包括5\*5的卷积核以及步长为2的maxpool层，我设置了epochs为100进行LeNet的训练，发现整体的准确率已经能够达到99%，取得了比较好的结果，最终利用MNIST中6万样本进行训练，并用1万测试样本进行测试得到了相应的**训练损失函数曲线以及测试准确率曲线。**

1. AlexNet

需要训练一个物体分类的CNN，这里我对经典模型AlexNet进行了实现，这也是第一个深度神经网络。整体网络结构增加了卷积层和全连接层的数量，提升了网络的效率。激活函数考虑到传统的tanh()这种饱和的非线性函数在梯度下降的时候要比非饱和的非线性函数慢得多，因此，在AlexNet中实现中使用ReLU函数作为激活函数。这里我在CIFAR10数据集上测试AlexNet的性能，实现了CNN物体分类功能的训练与测试。其中AlexNet的整体网络结果如下所示。

实验中我设置了epochs为200，batch\_size为512进行神经网络的训练，最后我记录相应的训练损失函数曲线以及测试准确率曲线。

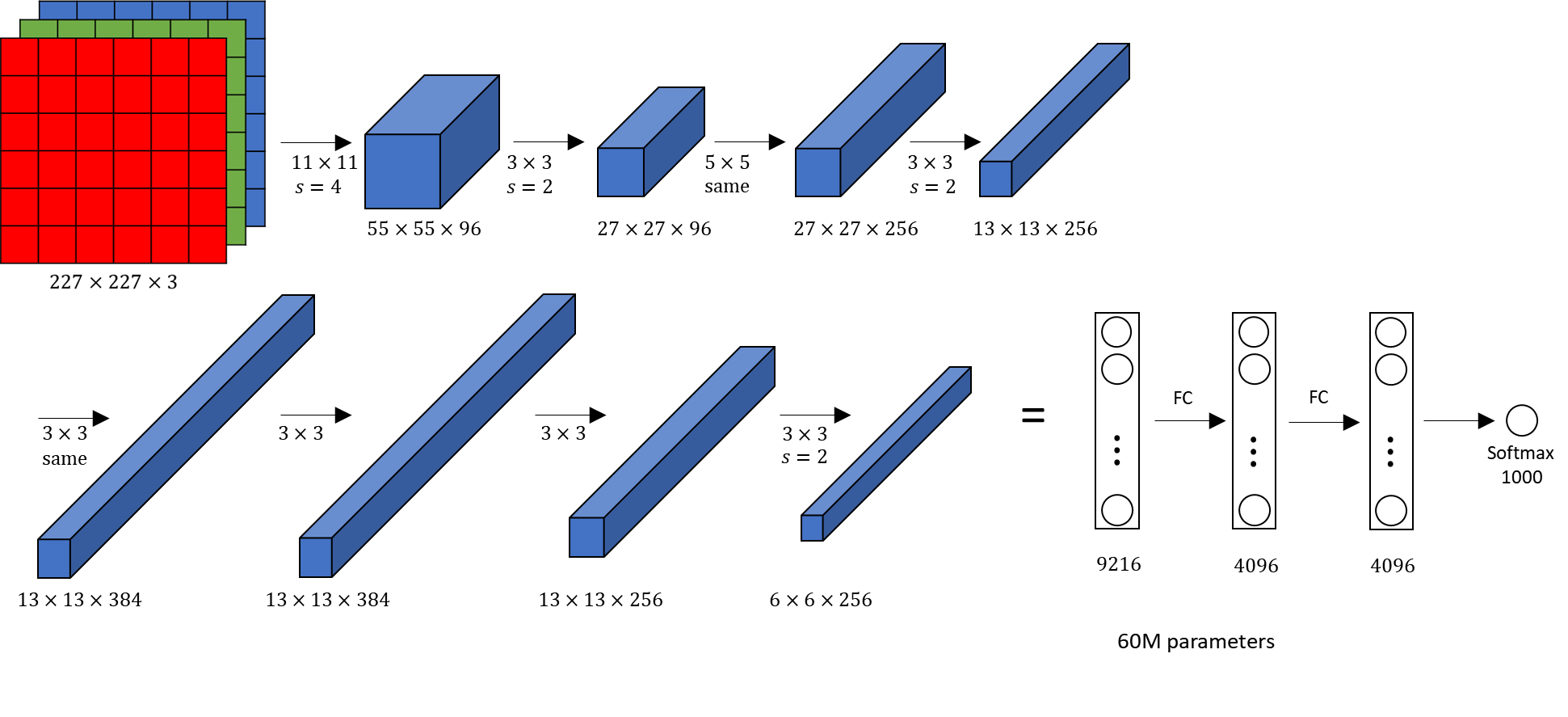


图2 AlexNet网络结构

1. 数据集介绍

MNIST数据集是机器学习领域中非常经典的一个数据集，由60000个训练样本和10000个测试样本组成，每个样本都是一张**28 \* 28**像素的灰度手写数字图片。CIFAR-10 是一个用于识别普适物体的小型数据集。一共包含 10 个类别的 RGB 彩色图 片：飞机（airplane）、汽车（automobile）、鸟类（bird）、猫（cat）、鹿（deer）、狗（dog）、蛙类（frog）、马（horse）、船（ship）和卡车（truck）。图片的尺寸为**32\*32**像素，数据集中一共有50000张训练图片和10000张测试图片。我采用torchvision中的DataLoader函数进行数据集的读取和转化，转到到numpy格式进行处理。

**实验结果：**

实验结果用训练集上损失函数值的变化和在测试集上分类的准确率来进行表示，可以发现LeNet-5在MNIST上达到近乎99%的分类准确率，同时我测试发现LeNet-5在CIFAR10上只能达到65%的准确率，说明其对于复杂的图片分类任务适应的不是很好；而AlexNet通过对CNN分类网络加深提升了分类的性能，在CIFAR10上能够取得接近75%的准确率，而从损失函数曲线发现没有发生过拟合，实验结果表示如下：

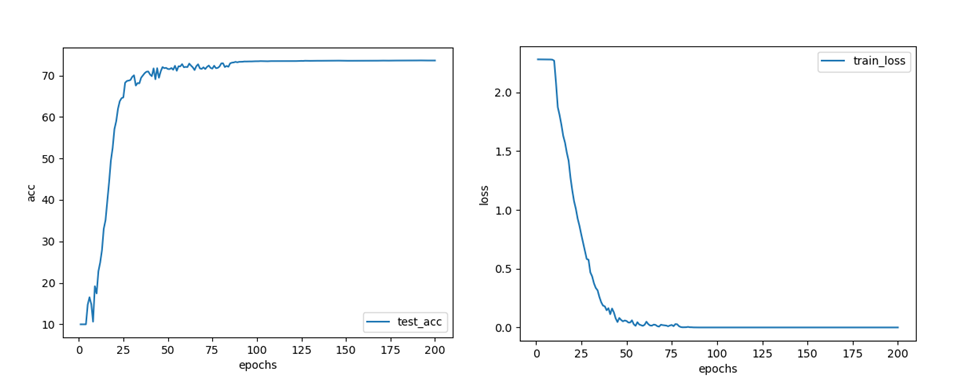


图3 AlexNet测试CIFAR10准确率曲线 图4 AlexNet训练CIFAR10分类损失值曲线

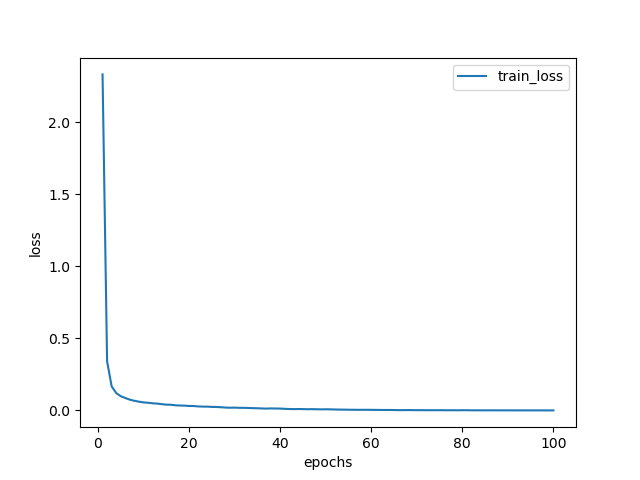
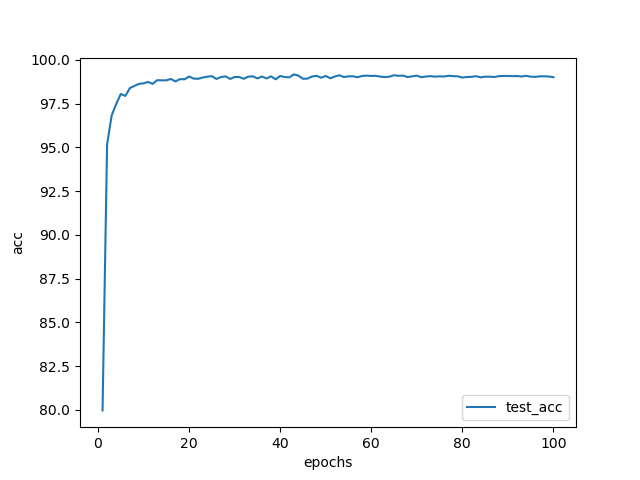


图5 LeNet测试MNIST准确率曲线 图6 LeNet训练MNIST分类损失值曲线