

# 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:

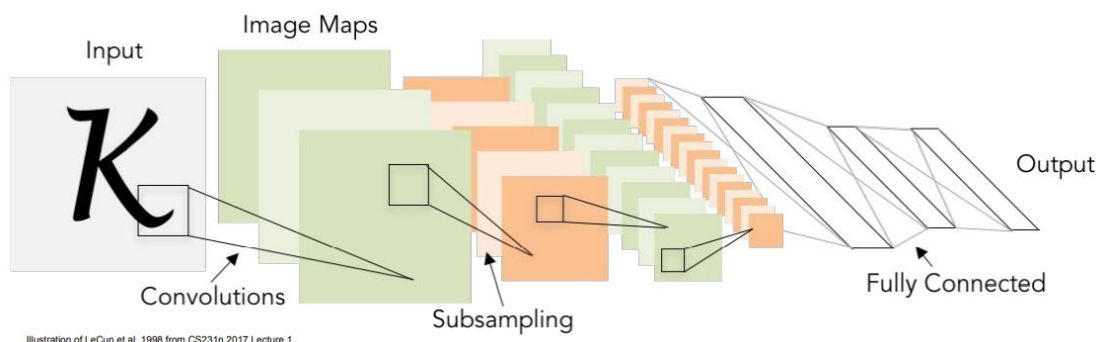


图 1 LeNet 网路结构

根据 LeNet 的定义, 我利用 torch 框架定义该神经网络, 包括  $5 \times 5$  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**。

### 2. 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。

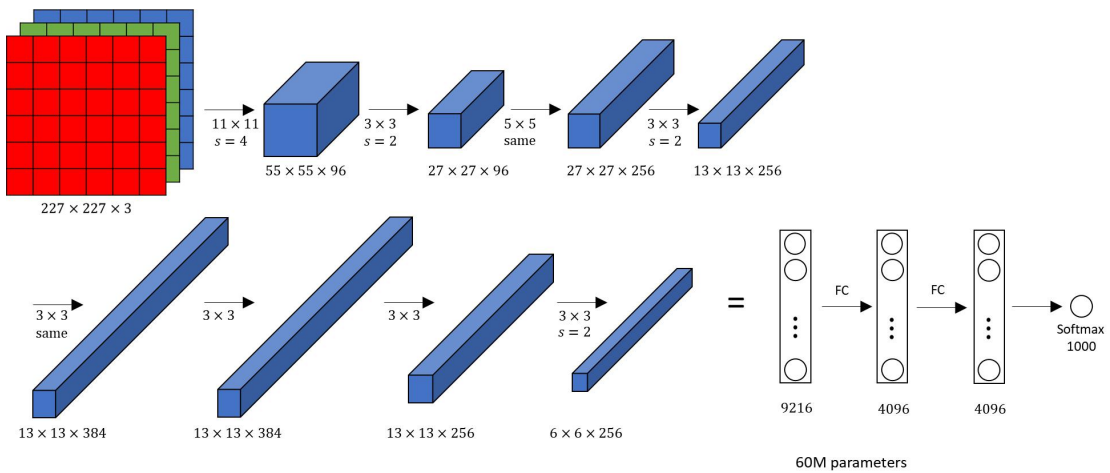


图 2 AlexNet 网络结构

### 3. 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  $28 \times 28$  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  $32 \times 32$  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：

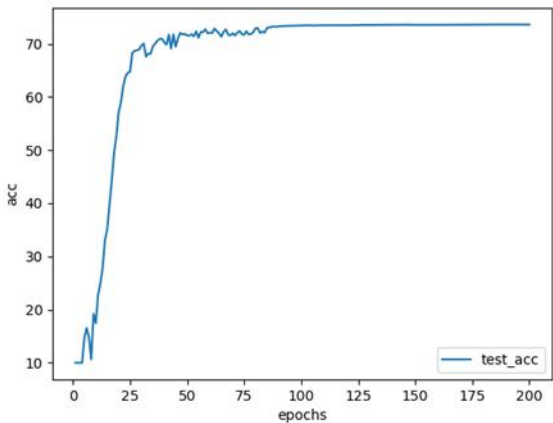


图 3 AlexNet 测试 CIFAR10 准确率曲线

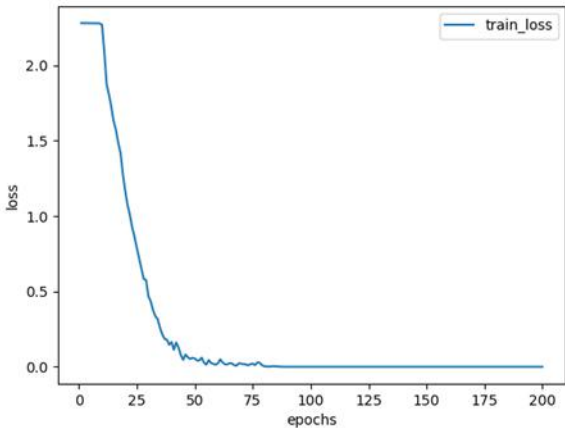


图 4 AlexNet 训练 CIFAR10 分类损失值曲线

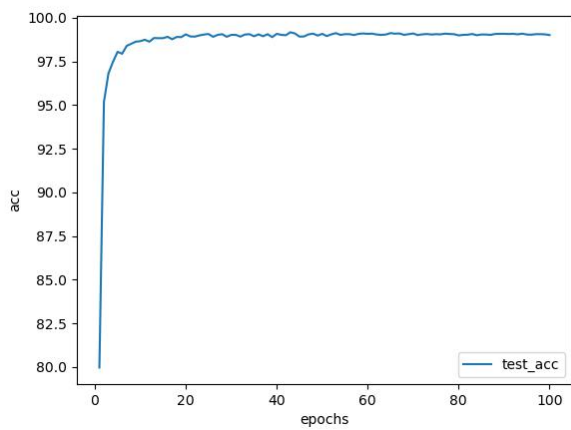


图 5 LeNet 测试 MNIST 准确率曲线

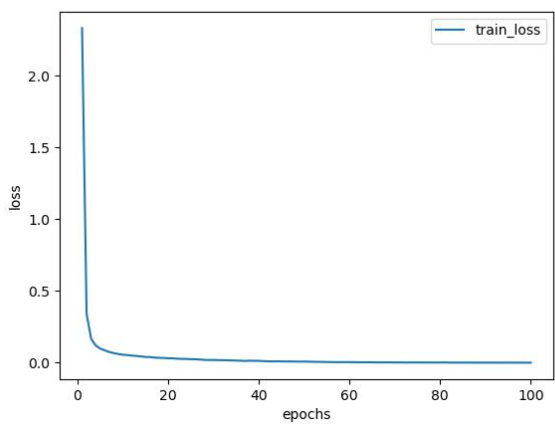


图 6 LeNet 训练 MNIST 分类损失值曲线