

**计算机视觉实验三实验报告**

|  |  |
| --- | --- |
| 姓 名： | 王嘉成 |
| 学 院： | 计算机学院 |
| 专 业： | 计算机科学与技术专业 |
| 班 级： | 计算机2207班 |
| 学 号： | U202215576 |
| 指导教师： | 刘康 |

|  |  |
| --- | --- |
| 分数 |  |
| 教师签名 |  |

2025年 1 月 1 日

目 录

[一、实验要求 1](#_Toc97642866)

[二、实验步骤及内容 2](#_Toc97642870)

[1、数据准备 2](#_Toc97642871)

[2、CNN网络设计和模型训练 2](#_Toc97642872)

[3、可视化平均特征图 4](#_Toc97642873)

[4、测试特征图的激活 5](#_Toc97642873)

[5、剪枝及评估 6](#_Toc97642873)

[三、实验框架 8](#_Toc97642874)

[1、实验环境 8](#_Toc97642875)

[2、卷积神经网络结构 8](#_Toc97642876)

[3、实验结构 9](#_Toc97642877)

[四、实验结果 1](#_Toc97642874)0

[1、训练参数设置 1](#_Toc97642875)0

[2、结果与分析 1](#_Toc97642876)0

# 一、实验要求

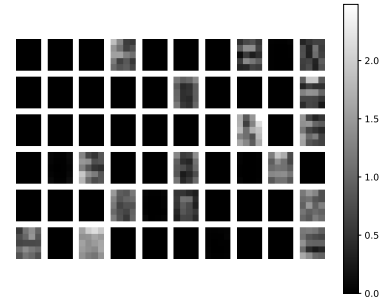
任务要求：对CNN分类神经网络进行权重剪枝实现模型压缩。

实现步骤：

1. 可对最后一层卷积层，依据输出特征图的神经元激活的排序，进行依次剪枝。例如：若最后一层卷积层的权重大小为，输出特征图大小为，在测试数据集上对个输出特征图的神经元激活（*test*）求平均并进行排序。按激活水平由低到高，对前*K*个神经元权重进行剪枝，。
2. 剪枝后的卷积层权重大小为，测试此时神经网络分类准确率。
3. 提示：可将待剪枝的神经元权重、偏置设为0，即相当于神经元剪枝而不用改变网络架构。

报告要求：

1. 画出最后一层卷积层（剪枝前）在整个测试数据集上的平均输出特征图（大小为）。示例如下，共*P*个特征图（如下图为6行10列，），每个特征图的大小为。



1. 画出横坐标为*K*，纵坐标为网络分类*accuracy*的折线图。
2. 实验报告包含网络设计、实验结果图，以及必要的分析等。
3. 代码和实验报告一起打成**ZIP**压缩包，以“姓名-学号-实验报告#”命名，比如“张三-2020XXX-实验报告一.zip”，提交到课程平台（https://smartcourse.hust.edu.cn/）。
4. 截止时间为1月2号下午5:00，逾期视为0分。

二、实验步骤及内容

## 、数据准备

首先明确了数据路径，并选用了 MNIST 数据集作为实验对象，该数据集包含手写数字图像，通过调用 PyTorch 提供的工具加载训练集和测试集，实验在加载的同时对数据进行了基础预处理：将图像数据从原始格式转换为张量，同时归一化到[0, 1]的范围，以便网络能够更有效地处理输入。

为了满足训练和测试阶段的不同需求，实验将训练数据设置为随机排列，以避免模型在训练过程中因数据顺序问题产生依赖性；而测试数据则保持其固定顺序，以确保模型性能的评估具有一致性和可重复性。加载后的数据以批量形式提供，每次传递64个样本，既优化了内存的使用，又提高了训练效率。

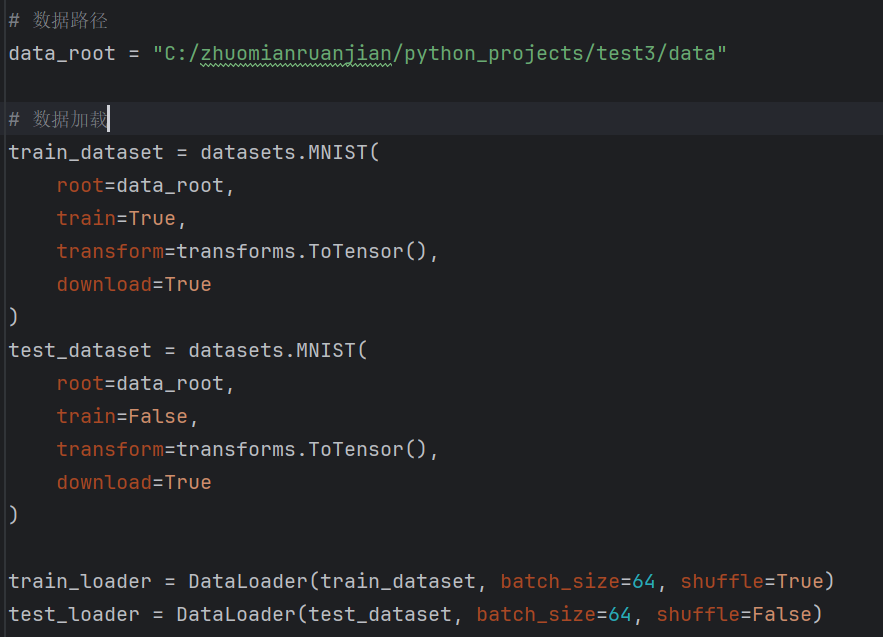


图 1 数据准备代码段

## 、CNN网络设计和模型训练

为了较好地完成本实验，我选择构建了一个简单的卷积神经网络，用于对MNIST数据集进行分类任务，模型包括两层卷积层和一个全连接层，设计上兼顾了特征提取与分类功能。

第一层卷积接收输入的单通道灰度图像，使用32个3x3的卷积核，并通过设置适当的填充保持输出的空间尺寸不变，经过非线性激活函数ReLU的处理，模型能够捕捉输入图像的局部特征。

第二层卷积进一步提取由第一层卷积产生的特征图，使用64个3x3的卷积核，输出更高维度的特征表示，在这之后，模型通过一个2x2的最大池化层对特征图进行下采样，从而降低计算复杂度并增强模型对位置变化的鲁棒性。

特征提取完成后，模型将特征图保存至last\_conv\_output以备后续分析。在进入全连接层之前，模型将特征图展平为一维张量，并通过全连接层输出10个类别的预测分数，用于对输入手写数字图像进行分类。

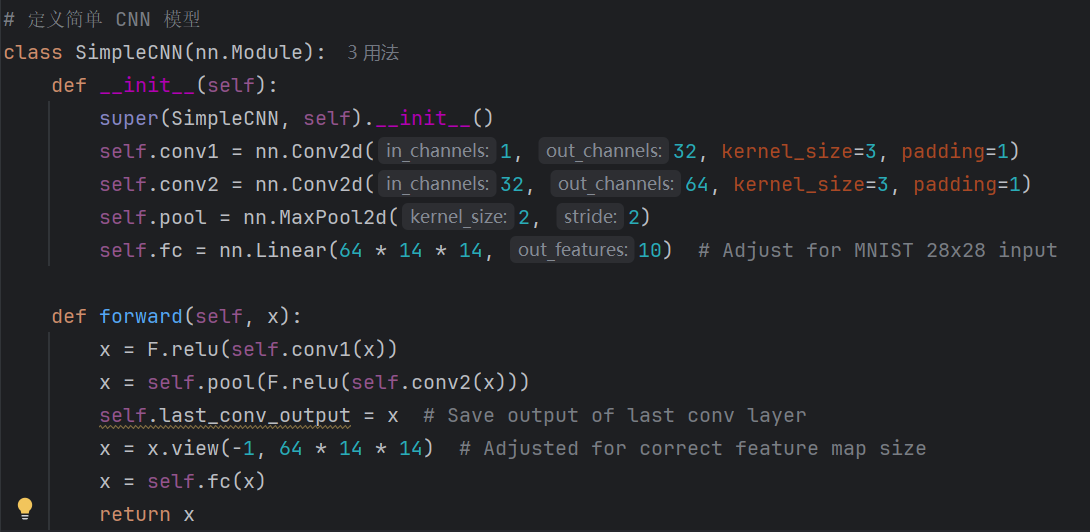


图 2 CNN网络设计部分代码段

接下来就是对模型进行训练，首先，模型被实例化并转移到合适的计算设备，以便充分利用硬件加速进行训练和推理。损失函数选择为交叉熵损失，用于衡量分类结果与实际标签之间的误差，优化器则采用Adam算法，以较快的收敛速度和稳定性调整模型参数。

训练过程设置为5个周期，每个周期中，模型在训练模式下对数据集进行逐批迭代。每次迭代中，首先将输入图像和对应标签转移到计算设备上，然后通过前向传播计算模型输出，并根据交叉熵损失函数计算误差，接着，通过反向传播更新模型参数。每个周期结束时，累积的平均损失会被输出，用于观察模型在训练过程中的优化情况。

训练完成后，模型的参数状态被保存为文件（model.pth），以便后续实验使用。加载模型时，通过重新实例化和加载权重的方式恢复训练好的状态，同时将模型设置为评估模式，以确保推理过程中不启用dropout等训练时特性。

在评估部分，代码通过对测试集的预测准确性进行计算来衡量模型性能。

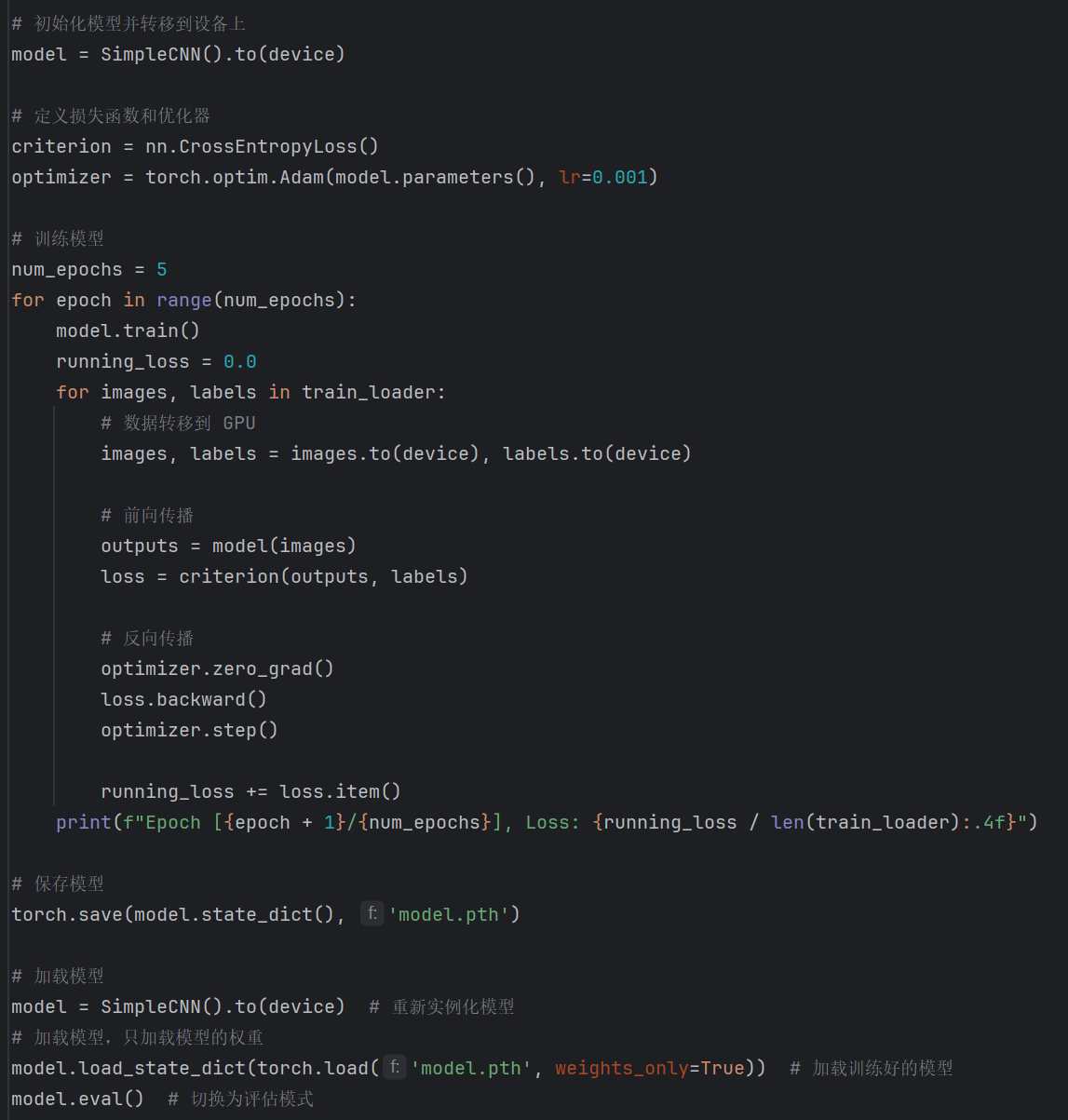


图 3 模型训练部分代码段

## 、可视化平均特征图

为了完成实验目标，画出最后一层卷积层在整个测试数据集上的平均输出特征图，我们首先定义了一个名为plot\_feature\_maps的函数，用于绘制和保存特征图，该函数接收特征图数据、绘制的行列数、以及可选的保存路径作为输入参数。在函数内部，通过matplotlib的子图功能创建一个绘图区，并逐一将特征图数据显示为灰度图。通过循环遍历，将特征图逐一分配到各个子图，并设置其显示为无坐标轴的模式以便更清晰地观察数据模式。如果提供了保存路径，函数会将生成的图像保存到指定位置,同时，生成的可视化图像会被显示在屏幕上。

接着，代码计算了网络对测试数据的平均特征图,通过将之前累积的所有特征图总和除以测试样本数，生成每个特征图的平均值矩阵,打印了该平均特征图的形状信息，便于确认其维度是否与预期一致。

最后，通过调用plot\_feature\_maps函数，将计算出的平均特征图进行可视化，并保存为文件。

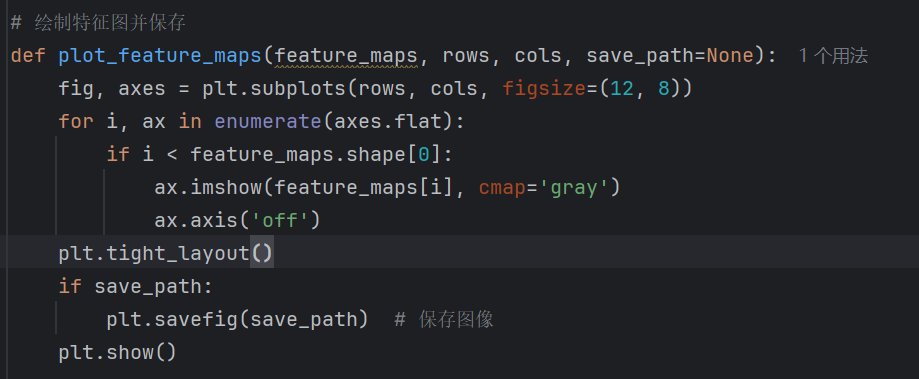


图 4 可视化平均特征图部分代码段

## 4、测试特征图的激活

我们需要通过分析模型对测试数据的输出，提取最后一层卷积层的平均激活值，作为衡量特征的重要性指标。

首先，定义了一个辅助函数get\_average\_activation，该函数通过前向传播计算模型对测试集的输出特征图，并从最后一层卷积层中提取特征图数据。每个特征图的平均激活值通过对其在空间维度上的均值进行计算得到，所有批次的平均激活值被逐一存储，并在循环结束后对批次维度求均值，生成最终的特征图平均激活值，这些结果以张量形式返回，便于后续分析。

接下来进一步累积了测试数据集中每张图像的特征图输出，模型对测试数据的每一批次进行前向传播，从中提取最后一层卷积层的输出特征图，并对批次维度进行平均，从而消除数据加载时的批量影响，累积的结果逐步叠加到sum\_feature\_maps中，用于最终归一化处理。

然后，通过调用get\_average\_activation函数，生成一个基于全测试集的特征图平均激活值向量。该向量可以反映最后一层卷积层中各特征图的重要性，为进一步的特征选择或剪枝操作提供依据。

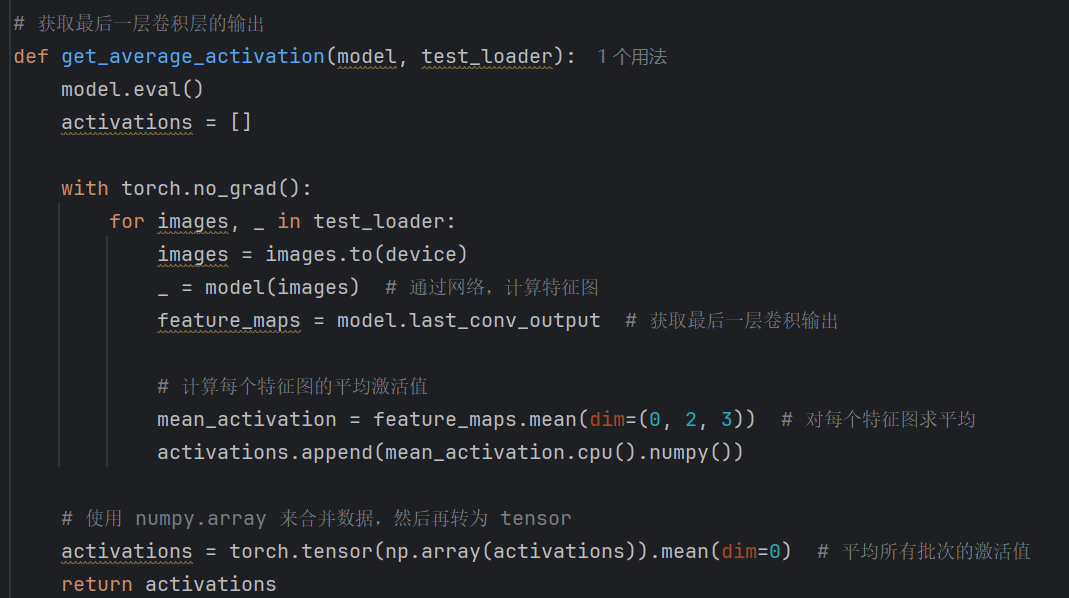


图 5 特征图激活中部分代码段

## 剪枝及评估

实现剪枝，我们要结合前面获取特征图平均激活值的部分，平均激活值较低的特征图被认为在网络的表达中贡献较小，可以作为剪枝的目标。

我们首先定义了一个prune函数，从模型中提取第一层卷积层的权重和偏置，并根据输入的特征图平均激活值对其进行排序，选取激活值最低的前K个特征图作为剪枝目标。为了避免越界错误，索引会被限制在权重张量的范围内。接着，将这些低激活值的卷积核权重和偏置设置为零，以模拟剪枝操作，并将修改后的权重和偏置赋值回模型中，从而完成剪枝。

绘制剪枝实验结果的函数plot\_pruning\_accuracy则负责生成剪枝量与测试准确率之间的折线图。图表以剪枝量为横轴，以测试准确率为纵轴，并通过标记点和网格线等细节使数据呈现更加直观，绘制完成后，图像会被保存为文件，便于记录和分析实验结果。

在实验中，代码逐步增加剪枝量，从剪去1个特征图到接近特征图总数，通过循环调用剪枝函数依次完成不同剪枝量下的操作。每次剪枝后，会对模型在测试集上的性能进行评估，将测试准确率存储在一个列表中，并打印当前剪枝量及其对应的测试准确率。完成所有剪枝实验后，调用绘图函数生成剪枝量与测试准确率的关系图。



图 6 剪枝及评估部分代码段

# 三、实验框架

## 、实验环境

操作系统：windows 11

CUDA：11.8

显卡：NVIDIA GeForce RTX 3070 Ti Laptop GPU

深度学习框架：PyTorch

开发工具：PyCharm Community Edition 2024.3.1

## 、卷积神经网络结构

本实验中设计的卷积神经网络结构如图所示，整体由卷积层、激活函数、池化层和全连接层组成，旨在对28x28大小的手写数字图像进行分类。网络的输入是单通道的28x28灰度图像，经过两层卷积层进行特征提取，最终通过全连接层输出分类结果。

网络结构的第一层是一个卷积层，其输入为单通道的图像，输出为32个特征图。卷积操作使用的是3x3的卷积核，并通过1的填充方式保持图像尺寸不变。卷积层后接一个ReLU激活函数，带来非线性变换，从而增强网络的表示能力。接下来，经过一个2x2的最大池化层，池化操作将特征图的空间尺寸减半，从而减少计算量和过拟合的风险。

第二层是另一个卷积层conv2，输入是32个特征图，输出64个特征图，卷积核仍为3x3，且使用相同的填充和池化方式，通过这一层，网络进一步提取图像中的高层次特征。

在特征提取完成后，通过x.view(-1, 64 \* 14 \* 14)对输出进行展平，将二维特征图转换为一维向量，准备输入全连接层，全连接层接收展平后的特征向量，并将其映射到一个大小为10的输出层，以表示每个类别的分数。

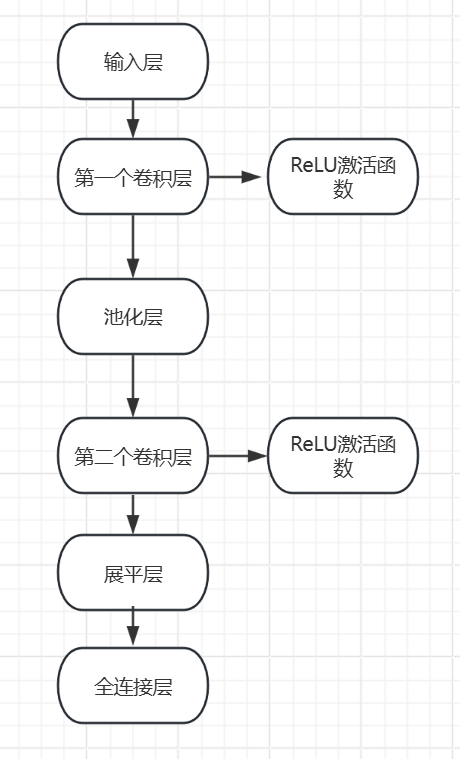


图 7 卷积神经网络结构图

## 、实验结构

实验工程中，main为主函数，data为数据集MNIST，model为实验保存的训练完的模型，其他png文件为实验生成的结果图像。

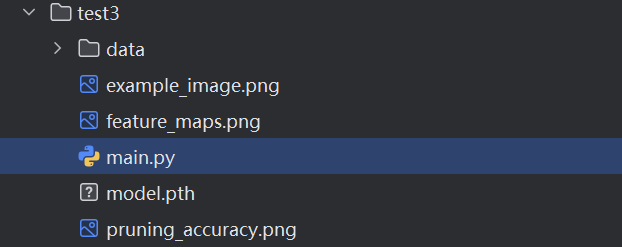


图 8 实验结构图

# 实验结果

## 1、训练参数设置

在这个实验中，训练模型的参数设置如下：在网格搜索的过程中，最佳的超参数配置为：

* 激活函数：ReLU
* 学习率 ：0.001，使用的是Adam优化器，通常是一个较常用的学习率。
* 训练轮数：5轮，指的是整个训练数据集将会被模型处理的次数。
* 批量大小：64，表示每次梯度更新时使用的训练样本数量。
* 损失函数：使用的是二分类交叉熵损失函数BCELoss，适用于输出概率值并进行二分类的任务。

## 结果与分析

**2.1 平均输出特征图**

在这个实验中，我设计的CNN模型最终输出共有64个通道，特征图的尺寸是14x14x64，所以我设置为8行8列，因为64个特征图适合安排成8x8的网格。

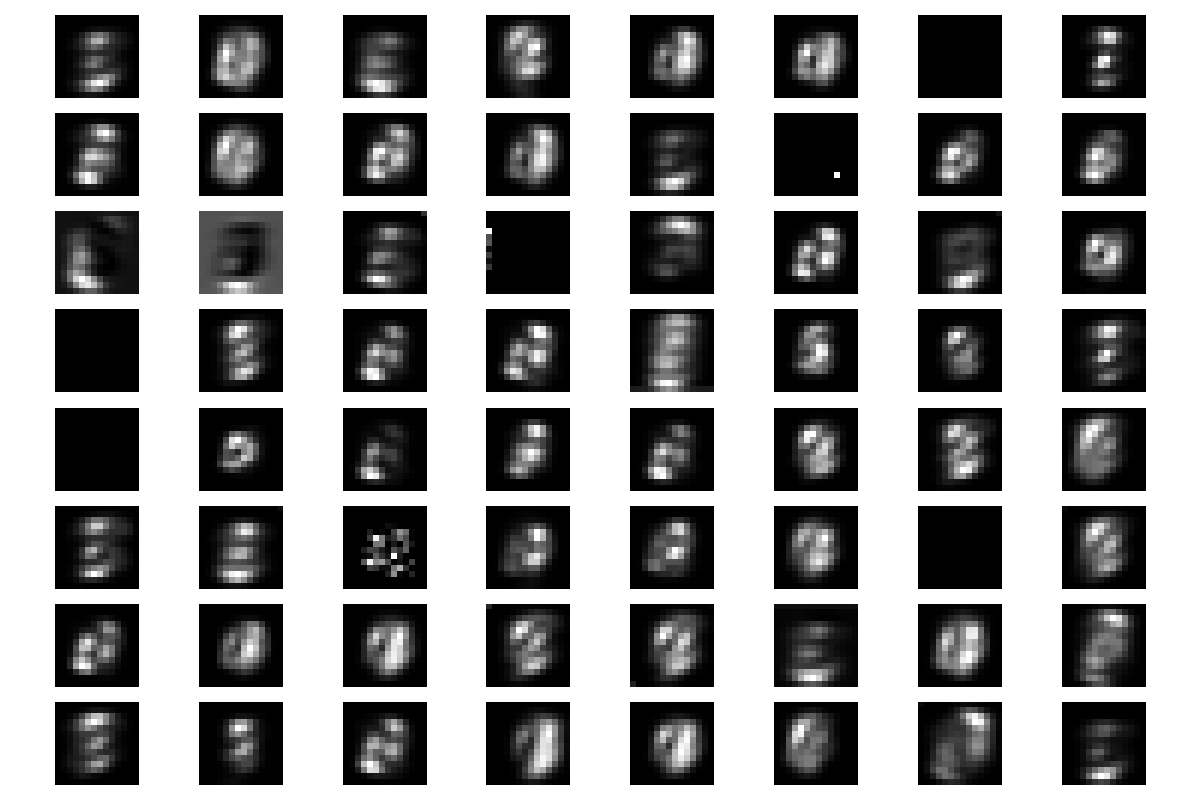
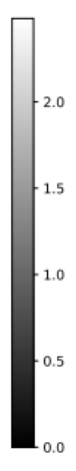


图 9 实验结构图

图 9 可视化特征图

从上到下，从左往右，可以发现更多特征被提取出来，边缘更加清晰。

**2.2 剪枝作用**

我们可以发现，在一开始减去一些不重要的特征图时，准确率并没有明显下降，而当剪枝数量增大到50时，准确率开始急速下降。

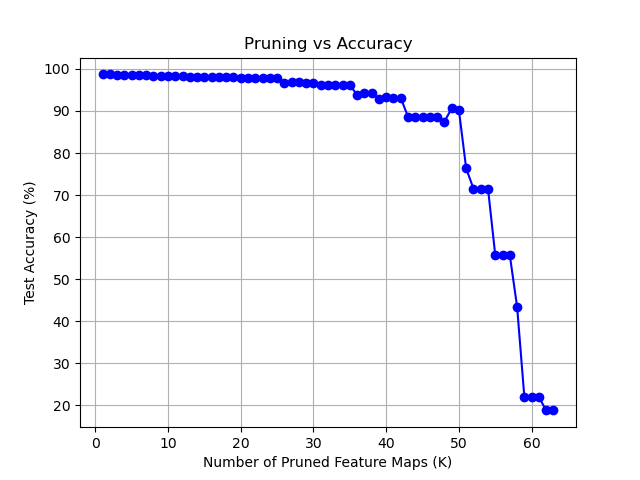


图 10 准确率与剪枝数量折线图