

哈爾濱工業大學

# 模式识别与机器学习实验报告

## 实验五

题目	卷积神经网络实验
学院	未来技术学院
专业	人工智能
学号	2023112419
学生	陈铠
任课教师	刘扬

哈尔滨工业大学计算机科学与技术学院

2025 年秋季

# 一、实验内容

## (一) 卷积神经网络介绍

CNN 是一种专为处理网格结构数据（如图像）设计的深度学习模型，核心优势是通过局部连接和参数共享，高效提取数据的空间特征。

### 1.核心任务

CNN 的核心任务分为两大阶段，从特征学习到任务落地形成完整流程：

**特征提取：**通过卷积、池化等层，自动从原始数据中学习特征。从低级特征（如边缘、纹理）逐步组合成高级特征（如物体部件、完整物体），无需人工设计特征。

**特征组合与分类：**通过堆叠卷积层和池化层，逐步提取高层抽象特征，并使用全连接层或全局平均池化层进行分类或回归。

在视觉任务中的应用：如图像分类、目标检测、语义分割、人脸识别等。

### 2.数学模型

CNN 的核心由以下几个算子组成：

卷积运算：对输入图像  $X \in \mathbb{R}^{H \times W \times C}$  进行卷积核操作  $K \in \mathbb{R}^{k \times k \times C}$

$$Y_{i,j} = \sum_{m=1}^k \sum_{n=1}^k \sum_{c=1}^C K_{m,n,c} \cdot X_{i+m-1,j+n-1,c} + b$$

作用是提取局部空间相关的特征。

非线性激活：如 Relu, LeakyRelu, ELU, SELU, GELU

$$f(x) = \max(0, x)$$
$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases}$$

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha(e^x - 1) & \text{if } x < 0 \end{cases}$$

$$f(x) = \begin{cases} \lambda x & \text{if } x \geq 0 \\ \lambda \cdot \alpha(e^x - 1) & \text{if } x < 0 \end{cases}$$

$$f(x) = x \times \Phi(x), \Phi(x) = \frac{1}{2} \left( 1 + \operatorname{erf} \left( \frac{x}{\sqrt{2}} \right) \right), \operatorname{erf} \text{ 为误差函数}$$

作用是引入非线性，提高网络表达能力。

池化层：如最大池化，平均池化

$$Y_{i,j} = \max_{(m,n) \in \text{window}} X_{i+m-1,j+n-1}$$

作用是减少参数量，提升平移不变性。

全连接层：将高维特征展开为一维向量，用线性层进行分类

$$y = \text{softmax}(Wx + b)$$

### 3.优化方法

CNN 的优化目标通常是最小化损失函数，例如交叉熵损失：

$$L = - \sum_i y_i \log (\hat{y}_i)$$

优化手段包括：

优化器：SGD、Adam、RMSProp

正则化：L2 权重衰减、Dropout、Batch Normalization

学习率策略：StepLR、Cosine Annealing、Warmup

数据增强：随机翻转、裁剪、颜色扰动

## (二) LeNet-5

LeNet-5 是最早成功应用于手写数字识别（MNIST）的 CNN，由 Yann Le Cun 于 1998 年提出。它验证了 CNN 能够自动提取图像特征并实现端到端训练的思想。

LeNet-5 输入为  $32 \times 32$  灰度图，输出为 10 类分类。

层次	类型	输出尺寸	说明
输入层	-	$32 \times 32 \times 1$	手写数字灰度图
C1	卷积(6 个 $5 \times 5$ 核)	$28 \times 28 \times 6$	提取低层边缘特征
S2	平均池化( $2 \times 2$ )	$14 \times 14 \times 6$	降维、平滑特征
C3	卷积(16 个 $5 \times 5$ 核)	$10 \times 10 \times 16$	提取高层组合特征
S4	平均池化( $2 \times 2$ )	$5 \times 5 \times 16$	降维
C5	卷积层(120 个 $5 \times 5$ 核)	$1 \times 1 \times 120$	全连接特征映射
F6	全连接(84 个神经元)	84	类似隐藏层
输出层	Softmax(10 类)	10	分类输出

每个卷积层：

$$h^{(l)} = f(W^{(l)} * h^{(l-1)} + b^{(l)})$$

池化层：

$$p^{(l)} = \frac{1}{|R|} \sum_{(i,j) \in R} h_{i,j}^{(l)}$$

最后通过 Softmax 预测类别。

## (三) VGG-11

VGG 网络由 Oxford 的 Visual Geometry Group 提出（2014），在 ImageNet 分类任务上表现突出。

核心思想是：使用多个小卷积核 ( $3 \times 3$ ) 堆叠，代替大卷积核，从而在参数不爆炸的前提下加深网络深度，提升特征表达能力。

VGG-11 是 VGG 系列中结构最简单的一种版本，适用于中小型数据集实验。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

## 二、实验环境

google colab 云平台

jupyter notebook

## 三、实验结果及分析

未进行优化的 LeNet-5 和 VGG-11 在 MNIST 和 CIFAR10 的表现:

Epoch/MNIST	LeNet5	LeNet5	VGG11	VGG11
	TrainLoss	TestAcc	TrainLoss	TestAcc
1	0.37	96.86	0.72	97.84
2	0.10	98.04	0.25	98.68
3	0.07	98.27	0.21	98.93
4	0.06	98.02	0.17	98.46
5	<b>0.05</b>	<b>98.54</b>	<b>0.16</b>	<b>99.11</b>

Epoch/CIFAR10	LeNet5	LeNet5	VGG11	VGG11
	TrainLoss	TestAcc	TrainLoss	TestAcc
1	1.76	46.40	1.88	35.01

2	1.43	51.50	1.56	40.16
3	1.30	52.94	1.39	52.51
4	1.22	56.61	1.25	59.08
5	<b>1.15</b>	<b>57.60</b>	<b>1.12</b>	<b>59.26</b>

在 MNIST 手写数字数据集（简单灰度图像任务）上，未优化的 LeNet-5 与 VGG-11 均呈现出 TrainLoss 逐步下降、TestAcc 稳步提升的趋势，表明模型均能有效学习数据特征并实现收敛。其中 VGG-11 的最终性能更优，TrainLoss 降至 0.16，TestAcc 达到 99.11%，显著高于 LeNet-5 的 TrainLoss 0.05 与 TestAcc 98.54%，这得益于 VGG-11 更深的网络结构与更多的卷积层，能够提取更抽象、更具判别性的高层特征，更适配 MNIST 的特征学习需求。

在 CIFAR10 彩色图像数据集（复杂小尺寸图像任务）上，两者性能均大幅低于 MNIST，LeNet-5 最终 TrainLoss 1.15、TestAcc 57.60%，VGG-11 TrainLoss 1.12、TestAcc 59.26%，仅略优于 LeNet-5，差距远小于 MNIST 上的表现。这是因为 CIFAR10 包含更多纹理、颜色等复杂特征，且图像尺寸小导致细节信息密集，未优化的模型（无正则化、无数据增强等）难以充分捕捉复杂特征，同时可能存在轻微过拟合，表现为 TrainLoss 持续下降但 TestAcc 提升逐渐放缓，模型泛化能力受限。

### 激活函数对比

activation	LeNet5+CIFAR10	VGG11+CIFAR10
ReLU	1.2111/55.75%	1.1647/62.46%
LeakyRelu	1.1597/57.36%	<b>1.0959/64.14%</b>
ELU	1.1187/58.16%	1.1631/61.62%
SELU	1.3102/53.28%	1.2420/59.43%
GELU	<b>1.0751/58.55%</b>	<b>1.1115/66.77%</b>

在 CIFAR10 数据集上，不同激活函数对 LeNet-5 与 VGG-11 的性能影响呈现一致趋势：GELU 激活函数均表现最优。

对 LeNet-5 而言，其 TrainLoss 1.0751 为所有激活函数中最低，TestAcc 58.55% 显著高于 ReLU (55.75%)、LeakyReLU (57.36%) 与 ELU (58.16%)，仅 SELU 因对数据分布敏感表现最差 (TestAcc 53.28%)；

对 VGG-11，GELU 的 TestAcc 66.77% 同样最高，LeakyReLU (64.14%) 次之，ReLU (62.46%) 与 SELU (59.43%) 依次降低。这一结果源于 GELU 结合了高斯误差函数的平滑非线性特性，能够对输入信号进行更细腻的响应，避免 ReLU 的“死亡神经元”问题与 LeakyReLU 的固定斜率局限，更适配 CIFAR10 复杂数据的特征学习过程，而 SELU 的自归一化特性需依赖特定数据分布，在 CIFAR10 上难以发挥作用，导致性能不佳。

### 正则化对比

dropout+batchnorm	LeNet5+CIFAR10	VGG11+CIFAR10
0.0/no	<b>1.1802/57.06%</b>	1.1303/ <b>66.43%</b>
0.3/no	1.2900/56.19%	1.1936/66.41%
0.5/no	1.3283/55.36%	1.2268/59.09%
0.0/yes	1.2216/54.64%	1.3037/52.54%
0.3/yes	1.2543/56.55%	1.1274/65.60%
0.5/yes	1.2894/ <b>57.53%</b>	<b>1.1255/63.94%</b>

以 CIFAR10 为实验对象，通过调整 dropout 比例 (0.0、0.3、0.5) 与是否启用 batchnorm，

发现正则化策略的效果因模型结构而异。

对 LeNet-5 (浅网络), 无单一最优组合: 0.5dropout+batchnorm 时 TestAcc57.53% 略高于无正则化 (0.0/no, 57.06%), 但 TrainLoss1.2894 高于无正则化的 1.1802, 说明较高 dropout 虽可能缓解过拟合, 但增加了浅网络的训练难度; 单独启用 batchnorm(0.0/yes) 时 TestAcc 降至 54.64%, 反而抑制特征学习, 推测因 LeNet-5 结构简单, batchnorm 的规范化作用过度约束了模型表达能力。

对 VGG-11 (深网络), 无正则化 (0.0/no) 时 TestAcc66.43% 最高, 0.3dropout+batchnorm 时 65.60% 接近最优, 而 0.5dropout (无论是否用 batchnorm) 均导致 TestAcc 大幅下降 (最低 59.09%), 表明 VGG-11 未优化时过拟合程度较低, 过高 dropout 会导致深层网络的特征信息丢失过多, 反而损害性能, 仅低比例 dropout 结合 batchnorm 可在不显著牺牲性能的前提下增强泛化能力。

## 优化器对比

optimizer	LeNet5+CIFAR10	VGG11+CIFAR10
Adam	1.1971/ <b>56.69%</b>	<b>1.1120/65.80%</b>
SGD	2.3009/10.00%	1.8295/45.68%
RMSprop	<b>1.1799</b> /56.22%	2.3028/10.00%

在 CIFAR10 数据集上, 不同优化器对模型训练的稳定性与最终性能影响显著。

对 LeNet-5, Adam 优化器表现最佳, TestAcc56.69%, RMSprop 次之 (56.22%), 而 SGD 的 TrainLoss 高达 2.3009, TestAcc 仅 10.00% (接近随机猜测), 这是因为 SGD 无自适应学习率机制, 在 LeNet-5 训练中难以平衡收敛速度与参数更新幅度, 导致收敛极慢;

对 VGG-11, Adam 同样表现最优 (TestAcc65.80%), SGD 虽 TrainLoss1.8295 较高, 但 TestAcc45.68% 仍优于 RMSprop (TestAcc10.00%), 推测 RMSprop 在深层网络中易出现梯度消失或学习率失控, 导致训练过程崩溃。整体而言, Adam 的动量项与自适应学习率机制能够更好适配浅、深两种网络结构, 尤其在 VGG-11 这类深层模型中, 可有效稳定参数更新过程, 避免训练异常, 保障模型收敛。

## 学习率对比

learningrate	LeNet5+CIFAR10	VGG11+CIFAR10
0.0001	1.6336/42.78%	<b>0.7158/75.52%</b>
0.001	1.2176/55.90%	1.2464/57.81%
0.003	1.0257/60.94%	1.5960/40.19%
0.005	<b>0.9928/62.47%</b>	1.6761/27.38%
0.008	1.0952/60.40%	-
0.010	1.0497/60.68%	-

以 CIFAR10 为实验背景, 学习率的调整需与模型深度高度匹配。

对 LeNet-5 (浅网络), 学习率从 0.0001 提升至 0.005 时, TestAcc 从 42.78% 持续升至 62.47%, TrainLoss 从 1.6336 降至 0.9928, 表明浅网络结构简单、参数较少, 较高学习率可加快参数更新速度, 促进模型快速捕捉数据特征, 低学习率则因更新幅度过小导致训练效率低下、特征学习不充分。

对 VGG-11 (深网络) 则呈现相反趋势: 学习率 0.0001 时 TestAcc75.52% 最高、TrainLoss0.7158 最低, 随着学习率升至 0.005, TestAcc 急剧降至 27.38%, TrainLoss 升至 1.6761, 这是因为深层网络参数规模大、梯度传播路径长, 高学习率易导致梯度爆炸

或参数更新幅度过大，破坏模型的稳定学习状态，而低学习率可让模型精细调整参数，逐步优化深层特征的表达能力。

#### 数据增强对比

augment	LeNet5+CIFAR10	VGG11+CIFAR10
-	1.5590/51.42%	1.6548/39.07%
随机裁剪	1.4310/51.87%	1.4437/54.75%
随机裁剪+随机旋转、擦除	<b>1.1927/56.53%</b>	<b>1.2621/55.98%</b>

在 CIFAR10 数据集上，数据增强策略（随机裁剪、随机裁剪+旋转+擦除）均能提升模型性能，且增强手段越丰富，效果越显著。对 LeNet-5，无增强时 TestAcc 51.42%、TrainLoss 1.5590，仅使用随机裁剪时 TestAcc 提升至 51.87%（提升微弱），而结合随机裁剪、旋转与擦除后，TestAcc 升至 56.53%、TrainLoss 降至 1.1927，表明多种增强手段可更有效扩充训练数据多样性，缓解浅网络的过拟合倾向；

对 VGG-11，数据增强的提升效果更明显：无增强时 TestAcc 仅 39.07%、TrainLoss 1.6548，随机裁剪后 TestAcc 升至 54.75%，进一步结合旋转与擦除后达 55.98%，TrainLoss 降至 1.2621，这是因为 VGG-11 参数更多、模型容量更大，更易对训练数据过拟合，数据增强通过生成“新样本”扩展数据分布，有效提升模型泛化能力，弥补了未优化模型在复杂数据上的性能短板。

## 四、结论

本次实验通过对 LeNet-5 与 VGG-11 在 MNIST、CIFAR10 数据集上的性能测试及关键超参数（激活函数、正则化、优化器、学习率、数据增强）的对比，得出以下核心结论：

其一，模型深度与数据集复杂度共同决定性能上限，VGG-11 因更深的结构在两类数据集上整体优于 LeNet-5，但在 CIFAR10 这类复杂数据上，未优化模型的性能均大幅下降，表明模型需结合优化策略才能适配复杂任务；

其二，激活函数选择需匹配数据特性，GELU 的平滑非线性特性在 CIFAR10 上表现最优，是两类模型的最优选择，SELU 则因对数据分布敏感适配性最差；

其三，优化器与学习率需与网络深度适配，Adam 的自适应机制是两类模型的最优优化器，浅网络（LeNet-5）适合较高学习率（0.005）以加快收敛，深网络（VGG-11）适合低学习率（0.0001）以保障稳定学习；

其四，数据增强是提升复杂数据泛化能力的有效手段，多种增强手段结合效果优于单一手段，且对深层模型（VGG-11）的性能提升更显著；

其五，正则化效果因模型而异，浅网络对正则化敏感，深层网络需控制 dropout 比例（避免过高导致特征丢失），单独使用 batchnorm 反而可能抑制浅网络性能。

整体而言，未优化的卷积神经网络在简单任务（MNIST）上可实现较高精度，但在复杂任务（CIFAR10）上性能受限，需通过“最优激活函数（GELU）+适配优化器（Adam）+匹配学习率+多手段数据增强”的组合策略提升性能，后续可进一步结合迁移学习、更复杂的正则化（如权重衰减）或更深的网络结构（如 ResNet），进一步突破 CIFAR10 等复杂数据集上的性能瓶颈。

## 五、参考文献

- [1]周志华著.机器学习[M].北京：清华大学出版社,2016.01.
- [2]李航著.统计学习方法第2版[M].北京：清华大学出版社,2019.05.
- [3]LeCun,Y.,Bottou,L.,Bengio,Y.,&Haffner,P.(1998).**Gradient-Based Learning Applied to Document Recognition**.Proceedings of the IEEE.
- [4]Krizhevsky,A.,&Hinton,G.(2009).**Learning Multiple Layers of Features from Tiny Images**.Technical Report.,A.(2009).Learning Multiple Layers of Features from Tiny Images.
- [5]Nair,V.,&Hinton,G.E.(2010).**Rectified Linear Units Improve Restricted Boltzmann Machines**.ICML.
- [6]Clevert,D.A.,Unterthiner,T.,&Hochreiter,S.(2016).**Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)**.ICLR.
- [7]Srivastava,N.,Hinton,G.,Krizhevsky,A.,Sutskever,I.,&Salakhutdinov,R.(2014).**Dropout: A Simple Way to Prevent Neural Networks from Overfitting**.JMLR.