



南開大學
Nankai University

网络空间安全学院
深度学习（高阶课）实验报告

卷积神经网络实验报告

姓名：王舒瑀

学号：2212777

专业：物联网工程

2025 年 6 月 18 日

目录

1 实验要求	2
2 实验内容	2
2.1 卷积基本原理	2
2.2 原始版本 CNN 网络结构实现分类	2
2.3 ResNet 实现分类	2
2.4 DenseNet 实现分类	3
2.5 SE-ResNet 实现分类	4
2.6 拓展部分	4
2.7 不同网络的比较与特点	5

1 实验要求

1. 掌握卷积的基本原理。
2. 学会使用 PyTorch 搭建简单的 CNN, ResNet, DenseNet, SE-ResNet 实现 Cifar10 数据集分类。

2 实验内容

2.1 卷积基本原理

在卷积神经网络中，**卷积**是一种关键的操作，用于提取输入数据（如图像）的局部特征。相比于传统的全连接神经网络，卷积神经网络利用卷积操作能够更高效地处理图像数据，并且减少参数数量，提高模型的计算效率。

从数学定义来看，在 CNN 中，卷积操作可以表示为：

$$Z(i, j) = \sum_m \sum_n X(i + m, j + n) \cdot K(m, n)$$

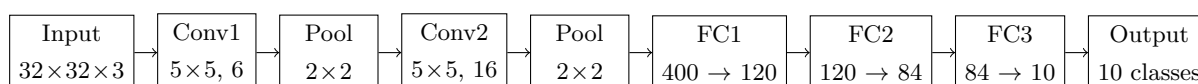
简单来说，卷积操作本质上就是滑动窗口**加权求和**，即用一个小的卷积核（滤波器）扫描输入数据的每个区域，并计算加权和，形成新的特征图。

卷积的优势

在传统的全连接神经网络中，每个神经元都会连接所有输入数据，导致计算量巨大。CNN 通过局部连接和参数共享的方式，让神经网络能够更高效地提取特征，并减少计算量。对于传统神经网络，如果输入是 256×256 的图片，全连接层需要 65536 个输入神经元，参数极多。而 CNN 通过卷积，使用 3×3 的卷积核，只需 9 个参数，就可以提取局部特征，并通过共享参数在整个图片中扫描，极大减少计算量。特别的，因为图像的局部信息是高度相关的，卷积能够有效地保留空间结构，比如边缘、角点、纹理等，而不像全连接网络那样直接展开成一维，丢失空间信息。

2.2 原始版本 CNN 网络结构实现分类

根据给出的实验文件，运行进行数据获取，数据预处理，基础 CNN 搭建，然后训练模型，最后进行模型评估。整体的 CNN 网络结构如下：



训练结果对比会在后续与其他网络的对比中给出

2.3 ResNet 实现分类

ResNet 的基本模块是残差块 (Residual Block)，该模块包含一个或多个卷积层以及一个跳跃连接 (skip connection)。在每个残差块中，网络不仅学习输入与输出之间的映射关系，还通过跳跃连接直接将输入加到输出中，这样网络的学习目标就变成了学习输入和输出之间的“残差”（即两者的差异）。

具体来说，ResNet 中的残差块可以表示为：

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}$$

Stage	Layer	In	Out	Kernel	Notes
Stem	conv1	3	64	3×3 , s1, p1	BN + ReLU
Stem	conv2	64	128	3×3 , s1, p1	BN + ReLU
	MaxPool	128	128	2×2 , s2	—
res1	Layer 0	128	128	3×3 , s1, p1	BN + ReLU
	Layer 1	128	128	3×3 , s1, p1	BN + ReLU
conv3	conv3	128	256	3×3 , s1, p1	BN + ReLU
	MaxPool	256	256	2×2 , s2	—
conv4	conv4	256	512	3×3 , s1, p1	BN + ReLU
	MaxPool	512	512	2×2 , s2	—
res2	Layer 0	512	512	3×3 , s1, p1	BN + ReLU
	Layer 1	512	512	3×3 , s1, p1	BN + ReLU
Classifier	MaxPool	512	512	4×4 , s4	—
	Flatten	512	512	—	—
	FC	512	10	—	bias=True

表 1: ResNet9 的网络结构

其中, \mathbf{x} 为输入, $\mathcal{F}(\mathbf{x}, \{W_i\})$ 为通过卷积层学习到的残差映射, \mathbf{y} 为输出。残差块的引入使得网络即使在层数非常深的情况下也能有效训练, 避免了深层网络难以优化的问题。

2.4 DenseNet 实现分类

DenseNet 核心思想是每一层都与前面所有层相连接, 从而增强特征复用与梯度传播。在 DenseNet 中, 每一层的输入是所有前面层输出的级联 (concatenation) 结果:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}])$$

DenseNet 同样缓解了梯度消失问题, 通过密集连接, 梯度可以直接从后层传播到前层; 同时特征重用高效, 所有层共享前面层的特征, 有效利用已学习的信息。相比于 ResNet, DenseNet 在性能相近甚至更优时使用了更少的参数, 由于特征复用与有效信息流动, 模型收敛更迅速。

Step	Operation	Bottleneck	Transition
1	BatchNorm2d	$C \rightarrow C$	$C \rightarrow C$
2	ReLU	$C \rightarrow C$	$C \rightarrow C$
3	Conv 1×1	$C \rightarrow 4C$	$C \rightarrow \lfloor \theta C \rfloor$
4	BatchNorm2d	$4C \rightarrow 4C$	—
5	ReLU	$4C \rightarrow 4C$	—
6	Conv 3×3 , p=1	$4C \rightarrow k$	—
7	AvgPool2d(2×2)	—	$\lfloor \theta C \rfloor \rightarrow \lfloor \theta C \rfloor$

其中, Bottleneck 中 $k = 12$, DenseBlock 增长率; Transition 中 $\theta = 0.5$, 压缩系数, $\lfloor \theta C \rfloor$ 表示向下取整。

表 2: DenseNet 中 Bottleneck 与 Transition 模块

但同时内存消耗也更大, 由于连接方式为拼接 (concat), 特征图维度随层数线性增长, 同时训练时间比 ResNet 长, 在本实验中, ResNet 一个 epoch 需要 10s 左右, 而 DenseNet 需要 60s 左右, 前向与反向传播中计算量略大。

Stage	Layer	In	Out	Count	备注
Stem	Conv3×3, s1, p1	3	24	1	bias=False
DenseBlock 1	Bottleneck (见表 2)	24 → 84 (累加)	+12/ch	6	每层输出叠加到通道
Transition 1	Transition (见表 2)	96	48	1	下采样 ×2
DenseBlock 2	Bottleneck (见表 2)	48 → 48+6×12=120	+12/ch	6	无下采样
Classifier	AdaptiveAvgPool2d(output_size=1)	120	120	1	—
	Linear(120 → 10)	120	10	1	bias=True

表 3: DenseNet 变体的整体网络结构

2.5 SE-ResNet 实现分类

SE-ResNet 基于 ResNet 网络架构, 结合了 Squeeze-and-Excitation (SE) 模块, 其核心目标是显式建模通道间的关系, 从而提升网络对特征重要性的选择性关注

SE 模块可以无缝嵌入现有的主干网络, 如 ResNet、DenseNet 等, 构成如 SE-ResNet、SE-DenseNet 等。

Step	Operation	SEResBlock	SEResBlockDown
1	Conv 3 × 3, s=1, p=1 + BN	$C \rightarrow C$	$C \rightarrow C'$
2	Conv 3 × 3, s=1, p=1 + BN	$C \rightarrow C$	$C' \rightarrow C'$
3	SE: AvgPool → FC($C \rightarrow C'/r \rightarrow C$) + Sigmoid	$C \rightarrow C$	$C' \rightarrow C'$
4	ReLU	$C \rightarrow C$	$C' \rightarrow C'$
5	Downsample: Conv 1 × 1, s=2 + BN	—	$C \rightarrow C'$
6	ReLU	—	$C' \rightarrow C'$

表 4: SEResNet 中两种残差模块 (含与不含下采样)

Stage	Layer / Module	In	Out	Count	Notes
Stem	Conv3×3, s1, p1	3	64	1	bias=False
	BatchNorm2d, ReLU	64	64	1	—
Layer1	SEResBlock (Tab. 4)	64	64	2	—
Layer2	SEResBlockDown (Tab. 4)	64	128	1	下采样 ×2
	SEResBlock (Tab. 4)	128	128	1	—
Layer3	SEResBlockDown (Tab. 4)	128	256	1	下采样 ×2
	SEResBlock (Tab. 4)	256	256	1	—
Layer4	SEResBlockDown (Tab. 4)	256	512	1	下采样 ×2
	SEResBlock (Tab. 4)	512	512	1	—
Classifier	AdaptiveAvgPool2d(1)	512	512	1	—
	Linear(512 → 10)	512	10	1	bias=True

表 5: SEResNet 模块化网络结构

2.6 拓展部分

传统 ResNet 中的残差块结构是 $\text{Output} = F(x) + x$ 其中 $F(x)$ 是一系列卷积操作。而 Res2Net 对 $F(x)$ 的结构进行细化, 通过将通道维度划分为多个分支子集, 逐级堆叠形成类似金字塔的特征组合。将输入特征 x 分为 s 个子特征映射 x_1, x_2, \dots, x_s , 第一个子特征 $y_1 = x_1$ 保持不变, 后续每个 y_i

为 3×3 卷积作用于前一个 y_{i-1} 与当前 x_i 的加和结果。最终将所有 y_i 沿通道维度拼接并通过 1×1 卷积融合，形成输出。这种跨层堆叠的方式，使得一个残差块内部即可同时捕捉多种尺度的上下文信息。

Step	Operation	Res2NetBlock	Res2NetBlockDown
Conv1	Conv 3×3 , stride=1, padding=1	$C \rightarrow w$	$C \rightarrow w$ (stride=2)
	BatchNorm2d + ReLU	$w \rightarrow w$	$w \rightarrow w$
	$w = C/\text{scales}$ (默认 scales=4)	—	—
Conv2	Conv 3×3 , stride=1, padding=1	$w \rightarrow w$	$w \rightarrow w$ (stride=2 if down)
	BatchNorm2d + ReLU	$w \rightarrow w$	$w \rightarrow w$
Conv3	Conv 3×3 , stride=1, padding=1	$w \rightarrow C'$	$w \rightarrow C'$ (stride=2)
	BatchNorm2d	$C' \rightarrow C'$	$C' \rightarrow C'$
Shortcut	identity	$C \rightarrow C$	Conv 1×1 , stride=2 + BN ($C \rightarrow C'$)
ReLU	最终激活	$C \rightarrow C$	$C' \rightarrow C'$

表 6: Res2NetBlock 与 Res2NetBlockDown 模块

Res2Net 引入了高效的多尺度特征建模机制，提升了网络对局部与全局信息的协同建模能力。在不显著增加参数量的前提下实现性能提升，尤其适合图像识别等多尺度敏感的任务。在本实验设置下，其测试表现优于传统 ResNet 和 DenseNet，但训练稳定性略低于 SE-ResNet，表明仍有优化空间，如可进一步结合 SE 模块形成 SE-Res2Net 架构以增强其鲁棒性。

Stage	Layer / Module	In	Out	Count	Notes
Stem	Conv7×7, s2, p3	3	64	1	bias=False
	BatchNorm2d + ReLU	64	64	1	—
Layer1	Res2NetBlock (Tab. 6)	64	64	3	scales=4, $w = 16$
Layer2	Res2NetBlockDown (Tab. 6)	64	128	1	$w = 32$
	Res2NetBlock (Tab. 6)	128	128	3	$w = 32$
Layer3	Res2NetBlockDown (Tab. 6)	128	256	1	$w = 64$
	Res2NetBlock (Tab. 6)	256	256	3	$w = 64$
Layer4	Res2NetBlockDown (Tab. 6)	256	512	1	$w = 128$
	Res2NetBlock (Tab. 6)	512	512	3	$w = 128$
Classifier	AdaptiveAvgPool2d(1)	512	512	1	—
	Linear(512 \rightarrow 10)	512	10	1	bias=True

表 7: Res2Net 在 CIFAR-10 上的模块化网络结构

可以看到 SE-Resnet 从训练准确率和损失来看，略优于 ResNet 和 DenseNet，表明 SE 模块有助于提升特征表达能力，强调关键通道，使模型能够更好地拟合训练数据。。测试准确率和损失略差，可能是由于 SE 模块增强了通道特征建模能力，使模型在训练集上拟合更充分，但也可能引入更多复杂度，从而在测试集上产生轻微过拟合，同时 SE 模块引入了新的超参数（如压缩率 r ），未经过充分调优，可能导致模型在测试时未能充分发挥其优势。

2.7 不同网络的比较与特点

首先，对于没有跳跃连接的卷积网络（CNN），由于使用了传统的逐层堆叠卷积层每一层都只接收前一层的输出作为输入，所以存在典型的训练问题，包括梯度消失或梯度爆炸，随着层数加深，反向

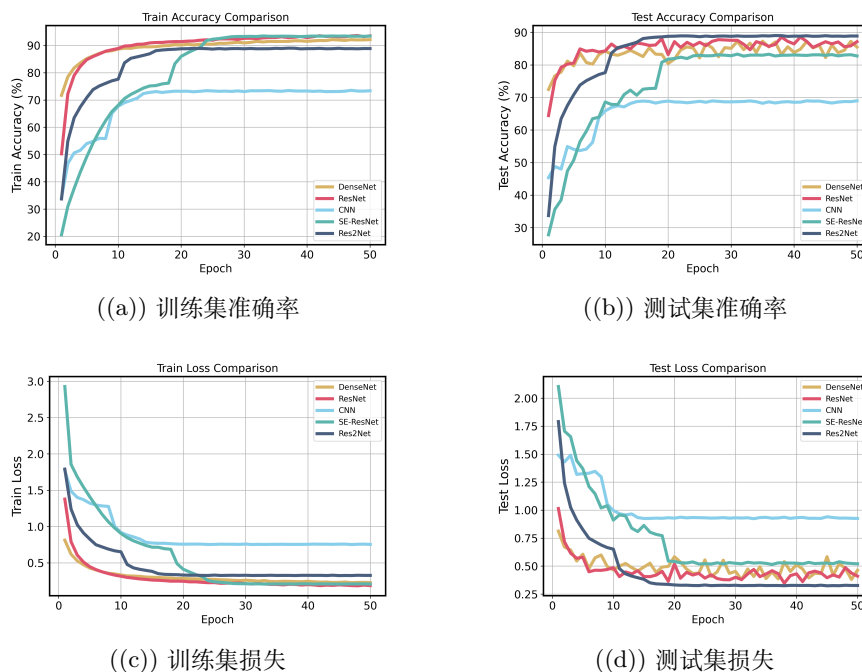
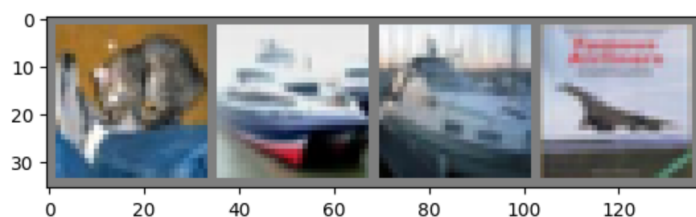


图 2.1: SE-ResNet 在 CIFAR-10 上的准确率及损失曲线图

传播时梯度逐渐变小或变大，导致模型很难训练。同时，由于这一问题，前面层的特征难以有效传递到后面层，导致收敛慢，准确率低，尤其在深层网络中这一缺点非常明显。



```
GroundTruth:  cat ship ship plane
Predicted:    cat ship car plane
Accuracy of the network on the 10000 test images: 90 %
Accuracy of plane : 93 %
Accuracy of car : 96 %
Accuracy of bird : 84 %
Accuracy of cat : 83 %
Accuracy of deer : 93 %
Accuracy of dog : 87 %
Accuracy of frog : 94 %
Accuracy of horse : 90 %
Accuracy of ship : 91 %
Accuracy of truck : 96 %
```

((a)) ResNet 分类测试结果

```
Accuracy for class: plane is 81.5 %
Accuracy for class: car is 94.8 %
Accuracy for class: bird is 67.4 %
Accuracy for class: cat is 58.2 %
Accuracy for class: deer is 79.2 %
Accuracy for class: dog is 69.9 %
Accuracy for class: frog is 90.4 %
Accuracy for class: horse is 87.0 %
Accuracy for class: ship is 90.0 %
Accuracy for class: truck is 85.5 %
```

((b)) SE-ResNet 分类测试结果

图 2.2: 两种模型在 CIFAR-10 上的分类效果对比

在 CNN 基础上 ResNet 引入了残差连接的方式，缓解了梯度消失的问题，残差连接提供了梯度的“捷径”，让梯度可以直接传回前层，也由此可以训练更深的网络，因为残差结构使得模型能更快收敛，使得优化问题更接近于恒等映射，更容易训练

在 Resnet 基础上改变得到的 DenseNet 所有层之间进行了全连接，每一层都接收前面所有层的输出作为输入，这样的改变使得特征复用率更高，早期层的特征会直接被后面所有层使用，同时相比 ResNet，DenseNet 的参数更少，且优化更容易，因为梯度可以直接流向前面所有层，训练更加稳定，

但同时缺点是计算和内存开销相对较大。

对于 SE-ResNet 而言，在 ResNet 基础上加入了 Squeeze-and-Excitation 模块，该模块对每个通道进行权重重标定。形象的来看就是如果一张图以红色系为主，那么自然对于 R 通道的关注就要相对应的多一些。SE 模块提升了模型的特征表示能力，通过学习通道间的依赖关系，强调“重要”的特征通道，从而提高模型泛化能力和准确率，使得训练表现更稳定。SE-ResNet 增加非线性能力，但保留了 ResNet 的梯度优势。

表 8: 各种网络在关键特性上的对比

特性	Plain	ResNet	DenseNet	SE-ResNet	Res2Net
连接方式	Seq. 堆叠	Add 残差	Concat 拼接	Add+SE 注意力	Add+ 多尺度
梯度传播	一般	强	更强	强	强（多尺度）
特征复用	无	一定程度	高度复用	一定 + 加权	多尺度复用
通道建模	无	无	无	有 (SE)	无
多尺度建模	无	无	无	无	有
收敛速度	中	快	更快	快	略慢于单尺度
参数量	少	中	少	略高于 ResNet	略高于 ResNet
模型复杂度	低	中	高	中	中高
提升方式	深度/宽度	残差	拼接	通道重标定	多尺度融合