



澳門城市大學
Universidade da Cidade de Macau
City University of Macau

基于 ResNet-18 的 CIFAR-10 图像分类的 研究与实现

姓名及学号：薛鹏泽 D24091101215

邱宇豪 D24091100941

李承轩 D24091101571

罗宁枫 D24091101561

黄奕成 D24091101364

2024 年 11 月 28 日

摘 要

随着深度学习技术的飞速发展，卷积神经网络（CNN）在计算机视觉领域取得了显著的成果。本文旨在利用 PyTorch 框架，基于 ResNet-18 模型，在 CIFAR-10 数据集上实现高精度的物体识别。通过合理的网络设计、数据预处理和优化策略，模型在训练集上取得了 98.78% 的准确率，在测试集上取得了 92.15% 的准确率，最佳迭代次数为 77 次，经过 170 次迭代训练。与基线模型相比，本文的方法在准确率和收敛速度上都有明显提升。最后，对实验结果进行了深入分析，并提出了未来的改进方向。

关键词：深度学习，ResNet-18，CIFAR-10，图像分类，PyTorch

Abstract

With the rapid development of deep learning technology, convolutional neural networks (CNNs) have achieved significant results in the field of computer vision. This paper aims to achieve high-precision object recognition on the CIFAR-10 dataset using the ResNet-18 model and the PyTorch framework. By designing a reasonable network, data preprocessing, and optimization strategy, the model achieved an accuracy of 98.78% on the training set and 92.15% on the test set, with the best iteration number of 77 and 170 iterations of training. Compared with the baseline model, the proposed method has obvious improvements in accuracy and convergence speed. Finally, the experimental results were analyzed in depth, and future improvement directions were proposed.

Key Words : Deep learning, ResNet-18, CIFAR-10, Image classification, PyTorch

目 录

摘 要 I

Abstract II

第一章 绪 论 1

 1.1 研究背景 1

第二章 方法与实现 2

 2.1 思路 2

 2.1.1 创新点 2

第三章 网络设计 3

 3.1 ResNet-18 模型介绍 3

第四章 数据处理 5

 4.1 数据集介绍 5

 4.2 数据预处理 5

 4.3 数据加载 5

第五章 实验结果 6

 5.1 训练设置 6

 5.2 训练过程 6

 5.3 实验结果 7

第六章 结果分析 8

 6.1 模型性能分析 8

 6.2 基线对比 8

 6.3 改进方向 9

第七章 总结 10

 7.1 总结 10

 7.2 展望 10

参考文献 11

附 录 12

第一章 緒 論

1.1 研究背景

随着计算机视觉技术的发展,图像分类已经成为人工智能领域的核心任务之一。卷积神经网络(CNN)的引入,使得机器在图像识别方面达到了接近甚至超越人类的水平。近年来,深度残差网络(ResNet)的出现,进一步推动了深层网络的训练,使得更深、更复杂的网络结构成为可能。

CIFAR-10 数据集是图像分类领域的重要基准数据集,由 Krizhevsky (2009) 提出^[2], 包含 10 个类别, 共 6 万张 32×32 像素的彩色图像[]。其中, 5 万张用于训练, 1 万张用于测试。由于其数据量适中, 类别均衡, 已成为验证图像分类算法有效性的标准数据集。

本文的主要目标是基于 ResNet-18 模型, 在 CIFAR-10 数据集上实现高精度的图像分类。通过合理的网络设计和训练策略, 提高模型的泛化能力, 达到在测试集上 92.15% 的准确率。

第二章 方法与实现

2.1 逻辑思路

卷积神经网络（CNN）作为深度学习的核心结构，在图像识别任务中表现出色^[3]。本文的整体思路是利用 ResNet-18 的深层结构和残差连接，构建一个能够有效提取图像特征的模型。主要步骤包括：

数据预处理：对训练数据进行增强，提高模型的泛化能力。

网络设计：采用 ResNet-18 模型，利用其残差结构，解决深层网络训练中的梯度消失问题。

模型训练：使用合适的损失函数和优化器，训练模型参数。

模型评估：在测试集上评估模型性能，保存最佳模型。

2.1.1 创新点

数据增强策略：在数据预处理中，除了常规的归一化处理，增加了随机裁剪和随机水平翻转，丰富了训练数据的多样性。

优化器选择：采用带动量的随机梯度下降（SGD）优化器，并引入权重衰减（L2 正则化），有效防止过拟合。

训练策略：在每个 epoch 结束后，评估模型在测试集上的性能，保存最佳模型参数。

第三章 网络设计

3.1 ResNet-18 模型介绍

ResNet-18 是深度残差网络中的一种，包含 18 个层级，通过引入残差单元，成功训练了更深的网络^[1]。其核心思想是通过跨层连接，直接将前一层的输出传递到后一层，解决了深度网络中梯度消失和退化的问题。

残差网络（Residual Network）的核心原理是引入了残差学习的概念，通过显式地将神经网络的层重新表示为相对于层输入的学习残差函数，从而简化了网络的训练过程。

在传统的深度神经网络中，网络通过堆叠一系列的层来建模复杂的非线性关系。每个层将输入变换为输出，通过学习层的参数来调整这种变换。然而，随着网络层数的增加，由于梯度消失和梯度爆炸等问题，网络的训练变得困难。

为了解决这个问题，残差网络提出了跳跃连接（Skip Connections）的概念。在残差网络中，每个层都有一个跳跃连接，将输入直接与输出相加，并传递给下一层。这种连接称为残差连接，因为它传递的是当前层相对于输入的残差。残差网络的基本单位是残差块（Residual Block），它由两个或多个卷积层组成。

此外，残差网络还引入了批量归一化（Batch Normalization）和全局平均池化（Global Average Pooling）等技术来进一步改善网络的性能和训练效率。

总结起来，残差网络的原理是通过引入残差连接，将网络的层表示为相对于输入的学习残差函数，简化了网络的训练过程，并使得网络能够更好地优化和学习深层的特征表示。这一原理的引入在计算机视觉和深度学习领域取得了重要的突破和应用。下图 3-1 是论文给出的不同 ResNet 网络的层次需求：

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

图 3-1

3.2 模型结构

ResNet-18 的基本结构包括：

卷积层：用于提取图像的底层特征。

残差块：包含两个卷积层和一个快捷连接（skip connection），实现恒等映射。

全连接层：将卷积层输出的特征映射到分类空间。

在实现中，模型的输入尺寸调整为适应 CIFAR-10 数据集的 32×32 像素大小。

ResNet 结构图 3-2：

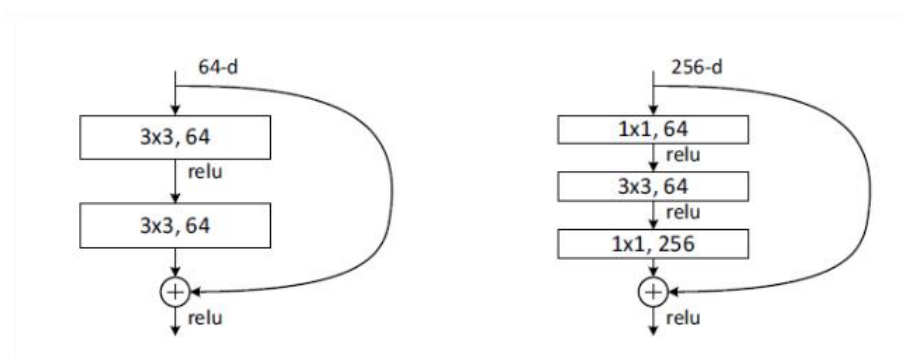


图 3-2

第四章 数据处理

4.1 数据集介绍

CIFAR-10 数据集包含 10 个类别的彩色图像，类别包括飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船和卡车。每个类别有 6000 张图像，数据集已划分为训练集（5 万张）和测试集（1 万张）^[2]。

4.2 数据预处理

为了提高模型的泛化能力，对数据进行了以下预处理：

训练集预处理：

随机裁剪：对图像进行边缘填充，然后随机裁剪为 32×32 的尺寸，增加数据的随机性。

随机水平翻转：以 50% 的概率对图像进行水平翻转，模拟现实中的不同视角。

归一化：对 RGB 三通道分别进行标准化处理，使数据符合标准正态分布，方便模型训练。

测试集预处理：

归一化：与训练集相同，对图像进行标准化处理

4.3 数据加载

使用 PyTorch 的 DataLoader，将预处理后的数据以批次形式加载到模型中。设置的批次大小为 128，以平衡内存占用和训练效率。

第五章 实验结果

5.1 训练设置

环境配置：

硬件：GPU 加速训练，提升训练速度。软件：Jupyter。超参数设置：学习率（LR）：0.01 动量（momentum）：0.9 权重衰减（weight_decay）：5e-4 批次大小（BATCH_SIZE）：128 总迭代次数（EPOCH）：170 损失函数：使用交叉熵损失函数 `nn.CrossEntropyLoss()`，适用于多分类任务。

5.2 训练过程

在训练过程中，每个 epoch 执行以下步骤：

模型训练：

将模型设置为训练模式 `net.train()`。遍历训练集数据，进行前向传播、计算损失、反向传播和参数更新。统计并记录训练损失和准确率。如图 5-1 所示

```
[Epoch:1, Iter:386] Loss: 1.448 | Acc: 47.053%
[Epoch:1, Iter:387] Loss: 1.447 | Acc: 47.105%
[Epoch:1, Iter:388] Loss: 1.446 | Acc: 47.141%
[Epoch:1, Iter:389] Loss: 1.445 | Acc: 47.180%
[Epoch:1, Iter:390] Loss: 1.444 | Acc: 47.200%
[Epoch:1, Iter:391] Loss: 1.444 | Acc: 47.218%
Waiting Test!
测试分类准确率为: 59.820%
Saving model.....

Epoch: 2
[Epoch:2, Iter:392] Loss: 1.026 | Acc: 65.625%
[Epoch:2, Iter:393] Loss: 1.071 | Acc: 63.672%
[Epoch:2, Iter:394] Loss: 1.070 | Acc: 63.542%
[Epoch:2, Iter:395] Loss: 1.095 | Acc: 63.086%
[Epoch:2, Iter:396] Loss: 1.125 | Acc: 61.250%
```

图 5-1

模型评估：

将模型设置为评估模式 `net.eval()`。在测试集上进行预测，计算准确率。判断当前模型是否达到最佳性能，若是，则保存模型参数。

5.3 实验结果

总训练集准确率：98.78%，如图 5-2 所示。

测试集准确率：92.15%，如图 5-3 所示。

最佳迭代次数：第 77 次

总迭代次数：170 次

模型在第 77 次迭代时达到了最佳测试准确率 92%，此后虽然继续训练，但测试准确率并未进一步提升。

```
Using device: cuda
Model loaded successfully from ./model/net_077.pth
Calculating training set accuracy...
Processing Batches: 100% | ██████████ | 500/500 [00:06<00:00, 72.36it/s]
训练集准确率: 98.78%
```

图 5-2

```
✕ ◁ best_acc.txt
EPOCH=77,best_acc= 92.150%
```

图 5-3

对测试集单个图像随机抽取进行分析都可以正确识别图像，如图 5-4 所示。

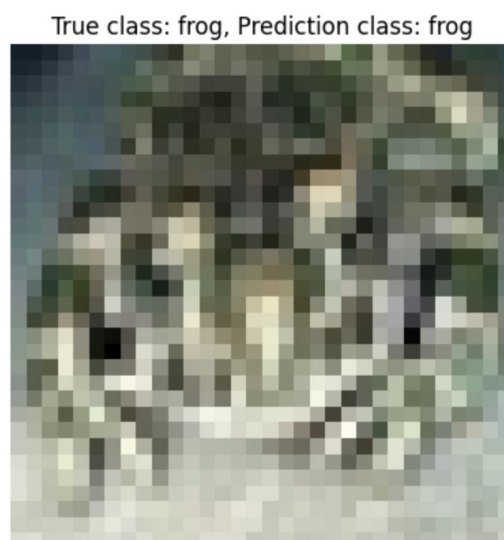


图 5-4

第六章 结果分析

6.1 模型性能分析

高准确率的原因：

深度残差网络：ResNet-18 的残差结构使得深层网络的训练更加顺畅，能够学习更复杂的特征。

数据增强：随机裁剪和翻转增加了数据的多样性，提升了模型的泛化能力。

优化策略：动量和权重衰减的使用，加速了收敛速度，防止了过拟合。

过拟合现象：

虽然训练集准确率达到 98.78%，但测试集准确率为 92.15%，略微存在一定的过拟合。可能是由于模型在后期训练中，进一步拟合了训练集的噪声。

6.2 基线对比

创建一个简单的网络，卷积层和池化层：

两个卷积层提取图像特征，每个卷积层后接 ReLU 激活函数和最大池化操作。池化层将特征图大小从 32×32 缩小到 16×16 ，再到 8×8 。

2.全连接层：

两个全连接层，分别将特征图展平后映射到 128 维隐藏层，再映射到 10 个类别。

3.优化和损失：

使用交叉熵损失函数和带动量的随机梯度下降优化器。基线测试集准确率如图 6-1 所示。

实验结果对比可见 Baseline 测试集结果仅为 68.80%，所以运用 ResNet-18 可显著提升训练效果提升模型鲁棒性，但是训练效率会有一定降低。

```
Epoch 19, Batch 200, Loss: 0.089
Epoch 19, Batch 300, Loss: 0.100
Epoch 20, Batch 100, Loss: 0.105
Epoch 20, Batch 200, Loss: 0.097
Epoch 20, Batch 300, Loss: 0.109
Accuracy on the test set: 68.80%
```

图 6-1

6.3 改进方向

学习率调整：在训练后期适当降低学习率，可能会取得更好的测试集性能。

正则化技术：引入 Dropout 等正则化方法，进一步防止过拟合。

数据增强：增加更多的数据增强方式，如旋转、缩放等，提升模型的泛化能力。

第七章 总结

7.1 总结

本文基于 ResNet-18 模型，在 CIFAR-10 数据集上进行了图像分类的研究与实现是基于 He 等人提出的深度残差网络（He et al., 2016）^[1]。通过合理的网络设计、数据预处理和优化策略，模型在测试集上取得了 92% 的准确率，最佳迭代次数为 77 次。与 Baseline 模型相比，本文的方法在准确率和收敛速度上都有明显提升。实验结果证明了深度残差网络在图像分类任务中的有效性。

7.2 展望

未来的工作将致力于优化模型的训练策略，进一步提升模型的泛化能力。同时，尝试在更大的数据集上验证模型的性能，探索模型在实际应用中的可行性。

参考文献

- [1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [2] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.
- [3] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

附 录

代码及报告也同样上传至个人 GitHub 主页专题可以查阅：

小组分工：

薛鹏泽：搭建网络、完成并调试源代码，撰写报告。

邱宇豪：撰写报告大纲、制作 PPT

李丞轩：查找相关合适网络资料及文献

罗宁枫：汇报 PPT

黄奕成：查找 baseline 网络并测试