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 95.384%的准确率，在测试集上取得了 92.730%的准确率，最佳迭代次数为 197 次，经过 200 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 95.384% on the training set and 92.730% on the test set, with the best iteration number of 197 and 200 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

第一章 绪 论

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 92.730%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，以及亮度变化及对比度变化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				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	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：

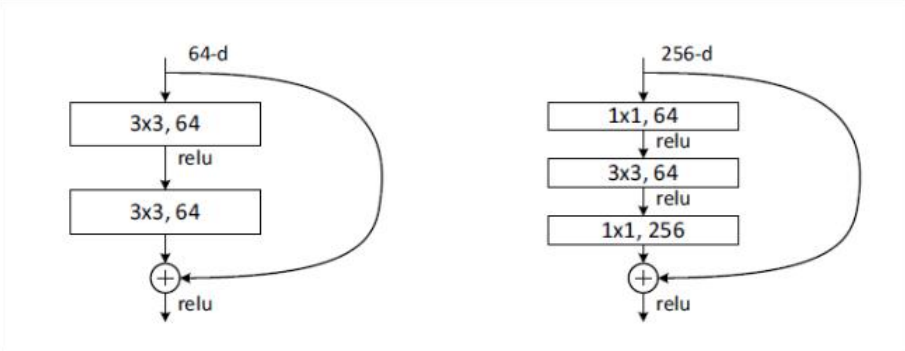


图 3-2

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

在数据预处理阶段主要分为训练集和测试集两部分处理。

对于训练集：

采用了多种数据增强技术：首先对图像进行 32x32 大小的随机裁剪（边缘填充 4 像素），然后应用随机水平翻转增加数据多样性；接着通过调整亮度和对比度进行颜色增强，并进行最大±10 度的随机旋转；在此基础上还应用了随机仿射变换，包括最大 10%范围的平移和 90%-110%范围的缩放；最后将图像转换为张量，进行归一化处理，并以 20%的概率进行随机遮挡。而对于测试集，则只进行了基本的张量转换和归一化处理，没有使用数据增强技术。这些预处理操作的目的是增加训练数据的多样性，提高模型的泛化能力。

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

其中训练集使用了 BATCH_SIZE=128 的批次大小并启用了随机打乱（shuffle=True），而测试集使用了批次大小为 100 且不进行随机打乱（shuffle=False）。两者都设置了 num_workers=2 来启用多进程加载数据，以提高数据加载效率。

4.4 模型与优化器配置

在模型与优化器配置部分，代码使用了 ResNet18 作为模型架构，并将其加载到自动选择的设备（GPU 或 CPU）上。损失函数采用了交叉熵损失（CrossEntropyLoss），用于多分类任务。优化器选择了随机梯度下降（SGD），

设置了初始学习率为 0.01，动量为 0.9，并使用了 5e-4 的权重衰减进行 L2 正则化，以帮助防止过拟合。这些配置共同作用于模型的训练过程，优化模型参数以提高分类准确率。

交叉熵损失函数：交叉熵损失函数是一种常用于分类任务的损失函数，特别是在多分类问题中。它通过衡量预测概率分布与真实标签分布之间的差异来评估模型的性能。具体来说，交叉熵损失会计算预测的概率分布与真实分布之间的负对数似然，惩罚模型对真实类别的低概率预测。损失值越小，表示模型的预测越接近真实标签。交叉熵损失函数能够有效地处理多分类问题，并且在深度学习中与 softmax 激活函数结合使用，能够提供稳定的梯度信号，帮助模型更好地学习。

本文中的运用：

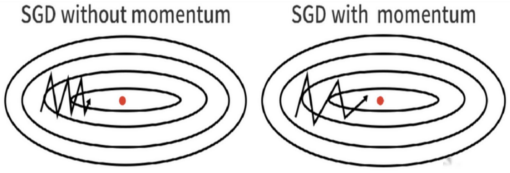
类别索引	0	1	2	3	4	5	6	7	8	9
类别名称	飞机	汽车	鸟	猫	鹿	狗	青蛙	马	船	卡车
Label	0	0	0	0	0	1	0	0	0	0
Pred (logits)	2.0	2.8	1.5	1.2	0.5	3.2	0.9	0.4	0.1	0.3
Softmax 概率	0.117	0.254	0.086	0.065	0.033	0.347	0.046	0.027	0.018	0.026

表 4-1

那么在这个识别出狗的任务中损失函数为：L = -log(0.347)=1.058。

SGD 优化：优化器被用来调整模型的参数，优化器的目的是调整模型权重以最小化损失函数。

没有动量的 SGD 直接使用当前批次的梯度来更新参数，容易在优化过程中出现震荡，特别是在遇到峡谷型损失曲面时，可能会在两侧来回振荡，导致收敛速度慢。而加入动量（本代码中设置为 0.9）的 SGD 会累积之前的梯度信息，类似于物理学中的动量概念，它会记住过去的梯度方向并在当前更新中保持一定的惯性，这样可以帮助模型在优化过程中保持一致的更新方向，减少震荡，加快收敛速度，并且更容易跳出局部最小值。特别是在处理曲面不规则的损失函数时，带动量的 SGD 表现更好。



第五章 实验结果

5.1 训练设置

在超参数方面，设置了总训练轮数 EPOCH=200，预训练轮数 pre_epoch 从 0 开始，批次大小 BATCH_SIZE 为 128，初始学习率 LR 设为 0.01。模型选择了 ResNet18 架构，损失函数使用交叉熵损失（CrossEntropyLoss），优化器采用 SGD（随机梯度下降），其中动量参数 momentum 设为 0.9，权重衰减参数 weight_decay 设为 5e-4 用于 L2 正则化。在模型保存策略上，每个 epoch 都会保存一次模型，同时设置了 best_acc=85 作为初始最佳准确率阈值，当测试准确率超过这个阈值时，会将其作为最佳模型保存到专门的 best 文件夹中，并记录相关信息。整个训练过程中会实时打印训练损失和准确率，并在每个 epoch 结束后进行测试评估。

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: 197 ====================
[Batch: 100/391] Loss: 0.128 | Acc: 95.602%
[Batch: 200/391] Loss: 0.128 | Acc: 95.664%
[Batch: 300/391] Loss: 0.133 | Acc: 95.474%

Epoch 197 训练完成:
平均损失: 0.135
训练准确率: 95.384%

正在测试 Epoch 197 ...

Epoch 197 测试结果:
测试准确率: 92.730%

正在保存 Epoch 197 模型...
【新记录】最佳准确率: 92.730%
最佳模型已保存至: ./best/best_model_92.73.pth
最佳准确率记录已保存至: ./best/best_acc.txt

==================================================


==================== Epoch: 198 ====================
[Batch: 100/391] Loss: 0.125 | Acc: 95.781%
[Batch: 200/391] Loss: 0.127 | Acc: 95.715%
[Batch: 300/391] Loss: 0.133 | Acc: 95.482%
```

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

分批次训练集准确率：95.384%

总训练集准确率：98.774%，如图 5-2 所示。

测试集准确率：92.730%，如图 5-3 所示。

最佳迭代次数：第 197 次

总迭代次数：200 次

模型在第 197 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_197.pth
Calculating training set accuracy...
Processing Batches: 100%|██████████| 500/500 [00:07<00:00, 70.97it/s]
训练集准确率: 98.77%
```

图 5-2

```
1 EPOCH=197, best_acc=92.730%
2 Model saved as: ./best/best_model_92.73.pth
```

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。

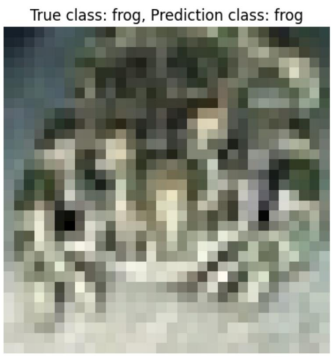


图 5-4

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：

在起初我们只是对训练集进行简单的处理在训练完模型后训练集准确率达到了 98.340%，但测试集准确率为 91.300%如图 6-1 所示，略微存在一定的过拟合。同时最佳次数在总训练次数的中后部分，后续训练基本无意义，而在我们对训练集进行进一步的更改后，改善了过拟合现象最佳训练次数很大程度接近与总共训练次数，说明后续训练为有效训练所以如果继续增大训练次数或稍稍调低学习率可能会有更好的测试集准确率得出更优越的模型。

```
==================== Epoch: 78 ====================
[Batch: 100/391] Loss: 0.044 | Acc: 98.547%
[Batch: 200/391] Loss: 0.049 | Acc: 98.402%
[Batch: 300/391] Loss: 0.050 | Acc: 98.354%

Epoch 78 训练完成:
平均损失: 0.050
训练准确率: 98.340%

正在测试 Epoch 78 ...

Epoch 78 测试结果:
测试准确率: 91.300%

正在保存 Epoch 78 模型...

================================================
```

图 6-1

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：调高训练次数，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.