

# 基于迁移学习的 Caltech-101 图像分类比较实验

——ImageNet 上微调预训练的 Resnet-18 和  
从零开始训练 Resnet-18 实现 Caltech-101 分  
类的效果比较

CV Mid-term Report: Task 1

姓名: 黄亦绪  
学号: 23307110412  
DATA130051.01 计算机视觉  
(春季, 2025)

复旦大学  
大数据学院

2025 年 5 月 28 日

# 目录

|  |           |
|--|-----------|
| <b>1 引言</b>                                    | <b>3</b>  |
| 1.1 实验背景 . . . . .                             | 3         |
| 1.2 实验环境: GPU . . . . .                        | 3         |
| 1.3 实验目标 . . . . .                             | 3         |
| <b>2 实验设计</b>                                  | <b>4</b>  |
| 2.1 数据集与训练集、验证集和测试集划分 . . . . .                | 4         |
| 2.2 数据预处理 . . . . .                            | 4         |
| 2.3 网络结构 . . . . .                             | 4         |
| 2.4 实现细节 . . . . .                             | 5         |
| 2.5 技术阐释 . . . . .                             | 5         |
| 2.5.1 优化器为什么选择 SGD 而不是 Adam 或 AdamW? . . . . . | 5         |
| 2.5.2 为什么引入权重衰减? . . . . .                     | 6         |
| <b>3 实验结果</b>                                  | <b>7</b>  |
| 3.1 训练过程 . . . . .                             | 7         |
| 3.2 超参数实验 . . . . .                            | 7         |
| 3.2.1 概述 . . . . .                             | 7         |
| 3.2.2 具体关系 . . . . .                           | 7         |
| 3.3 TensorBoard 可视化和分析 . . . . .               | 10        |
| 3.3.1 训练损失函数的衰减情况 . . . . .                    | 10        |
| 3.3.2 验证损失函数的衰减情况 . . . . .                    | 11        |
| 3.3.3 验证准确率的变化趋势 . . . . .                     | 13        |
| 3.3.4 从零训练的 ResNet-18 . . . . .                | 15        |
| 3.4 两者的测试准确率比较 . . . . .                       | 17        |
| 3.5 性能分析 . . . . .                             | 18        |
| <b>4 讨论与分析</b>                                 | <b>19</b> |
| 4.1 实现中的挑战 . . . . .                           | 19        |
| 4.2 性能差距 . . . . .                             | 19        |
| 4.3 改进方向 . . . . .                             | 19        |
| <b>5 结论</b>                                    | <b>20</b> |

# 1 引言

## 1.1 实验背景

卷积神经网络（CNN）是图像分类领域的核心技术。自 LeNet 首次应用于手写数字识别以来，CNN 经历了 AlexNet、VGGNet、GoogLeNet 和 ResNet 等关键模型的发展，显著提高了分类准确率。预训练模型（如在 ImageNet 上训练的 AlexNet 或 ResNet）通过迁移学习能够显著提升小规模数据集上的性能。Caltech-101 数据集包含 101 个类别，适合测试 CNN 的分类能力。

本次实验基于 Caltech-101 数据集，使用预训练的 ResNet-18 模型，修改输出层以适应 101 个类别，微调网络参数，并与从零训练的模型进行对比。实验旨在深入理解迁移学习、超参数调优和 CNN 的优化过程，验证预训练模型的性能提升，并通过 TensorBoard 可视化训练过程。

## 1.2 实验环境：GPU

实验使用 Nvidia RTX4060 GPU (Driver Version: 572.83, CUDA Version: 12.8)，通过 PyTorch 实现并行高效计算。

## 1.3 实验目标

基于 Caltech-101 数据集，目标是实现一个基于预训练 ResNet-18 的图像分类器，验证集准确率目标为 90% 以上，并与从零开始训练的 Resnet-18 对比，分析预训练带来的性能提升。实验还将探索学习率、训练步数、批次大小等超参数的影响，并可视化训练过程中的损失和训练准确率的变化。

## 2 实验设计

### 2.1 数据集与训练集、验证集和测试集划分

Caltech-101 数据集包含 9,144 张图像，101 个物体类别和 1 个背景类，每类 40 至 800 张图像。按标准划分，训练集每类随机选取 30 张图像，剩余为测试集，验证集从训练集划分 20%。这样进行划分的目的包括：

- **确保类别平衡** 尽管原始数据集中每类图像数量差异较大 (40-800 张)，通过每类固定选取 30 张图像作为训练集，减少了类别间样本量不平衡的影响，有助于模型在训练时公平学习各类特征。
- **验证集的独立性** 从训练集中划分 20% 作为验证集（即每类约 6 张图像），用于超参数调优和模型选择，避免测试集数据泄露，确保模型在测试集上的最终性能评估是独立的。
- **测试集的代表性** 剩余图像作为测试集（每类约 10-770 张，取决于原始数量），保留了类别内的大部分多样性，用于评估模型在未见过数据上的泛化能力，反映真实部署场景。

### 2.2 数据预处理

接下来的数据预处理包括：

- **归一化** 像素值缩放到  $[0, 1]$ ，基于 ImageNet 均值和标准差标准化。
- **数据增强** 训练时应用随机水平翻转、随机裁剪 ( $224 \times 224$ ) 和旋转 ( $\pm 15$  度)；测试时仅中心裁剪。
- **批处理** 批次大小为 32，适配 GPU 内存。

### 2.3 网络结构



图 1: ResNet-18 网络结构

实验基于 ResNet-18，结构包括：

- **卷积层**: 初始  $7 \times 7$  卷积核，64 通道，步幅 2，接最大池化 ( $3 \times 3$ ，步幅 2)。
- **残差块**: 4 组残差块（每组 2 个），通道数依次为 64、128、256、512。

- **输出层**: 将全连接层从 ImageNet 的 1,000 类改为 101 类, 适应 Caltech-101。

## 2.4 实现细节

- **预训练 ResNet-18 模型 (微调)**:
  - **迁移学习**: 加载 ImageNet 预训练的 ResNet-18 权重, 冻结前三组残差块, 仅微调最后一组残差块和全连接层以保留通用特征并适配 Caltech-101。
  - **优化器**: 使用 SGD, 动量 0.9。输出层学习率通过网格搜索测试 [0.01, 0.001, 0.0001], 其余层学习率为输出层的 1/10, 正则化系数 (权重衰减) 测试 [1e-4, 1e-3]。
  - **超参数调优**: 网格搜索 108 种组合, 包括学习率 ([0.01, 0.001, 0.0001])、epoch 数 ([10, 20, 30])、批次大小 ([16, 32, 64]) 和权重衰减 ( $[10^{-4}, 10^{-3}]$ )。学习率无衰减策略。
  - **可视化**: 使用 TensorBoard 记录训练和测试集的损失及测试集准确率, Matplotlib 生成曲线和参数散点图。
- **从零训练 ResNet-18 模型**:
  - **初始化**: 所有层采用 He 初始化, 从零开始训练以对比预训练效果。
  - **优化器**: 使用 SGD, 动量 0.9, 采用预训练模型 FC 最佳超参数 (学习率 0.001, 批次大小 32, 权重衰减 1e-4, 30 epoch)。
  - **训练设置**: 为保证全连接层与预训练模型的全连接层一致, 训练和测试数据加载器使用相同学习率、训练步长、批次等, 控制变量, 确保公平对比。
  - **可视化**: 使用 TensorBoard 记录训练和测试集的损失及测试集准确率, 纳入 Matplotlib 可视化进行对比。

## 2.5 技术阐释

### 2.5.1 优化器为什么选择 SGD 而不是 Adam 或 AdamW ?

- **历史传统** 在深度学习早期 (如 AlexNet、VGG、ResNet 的原始论文中), SGD 带动量是图像分类任务的标准优化器, 尤其是在 ImageNet 等大规模数据集上。ResNet-18 的预训练权重是在 SGD 优化下得到

的，因此在微调时继续使用 SGD 可以更好地保持权重更新的“一致性”。

- **迁移学习需求** 我的实验主要基于预训练 ResNet-18，通过迁移学习在 Caltech-101 数据集上微调。SGD 带动量在微调预训练模型时通常表现更稳定，尤其是在小数据集（如 Caltech-101，训练集仅 7,315 张图像）上。SGD 的更新规则较为简单，倾向于在预训练权重的“附近”进行平滑调整，避免大幅偏离已学习的特征。
- **Adam 的潜在问题** Adam（或 AdamW）是一种自适应优化器，通过自适应地调整学习率（基于一阶和二阶动量的估计）加速梯度下降。然而，Adam 在迁移学习中可能过于激进，导致快速改变预训练权重，破坏 ImageNet 上学到的通用特征。这种“过优化”在小数据集上可能引发过拟合，尤其是当训练数据不足以重新学习深层特征时。

### 2.5.2 为什么引入权重衰减？

权重衰减是一种正则化方法，通常通过在损失函数中添加权重的 L2 范数惩罚项实现，也就是在 Assignment 1 中的 L2 正则化。数学上，损失函数从原始的  $L(\theta)$  修改为：

$$L_{\text{reg}}(\theta) = L(\theta) + \frac{\lambda}{2} \sum_i \theta_i^2$$

- $L(\theta)$ : 原始损失（此处是交叉熵损失）
- $\lambda$ : 权重衰减系数（即 `weight_decay`, 如  $10^{-4}$ ），控制正则化强度
- $\sum_i \theta_i^2$ : 模型权重的 L2 范数，仅针对权重，不包括偏置

- 权重衰减通过 L2 正则化限制权重大小，防止 ResNet-18 在 Caltech-101 小数据集上过拟合，提升模型泛化性能。
- 在预训练模型微调中，权重衰减稳定全连接层学习，保持 ImageNet 预训练特征有效性，增强测试集表现。
- 针对 Caltech-101 的类别不平衡，权重衰减促进跨类别共享特征学习，间接改善对样本稀少类别的泛化能力。
- 结合 SGD 动量，权重衰减确保权重更新稳定，降低模型对训练数据噪声的敏感性，满足测试准确率目标。

## 3 实验结果

### 3.1 训练过程

预训练模型训练 10, 20 或 30 个 epoch, 记录训练和验证损失函数的值的变化以及模型在验证集上的准确率变化。从零训练的 ResNet-18 则使用预训练模型超参数搜索中最佳的参数进行训练, 也同时记录训练和验证损失函数的值的变化以及模型在验证集上的准确率变化。最后将搜集的数据使用 TensorBoard 可视化。

### 3.2 超参数实验

#### 3.2.1 概述

为了优化预训练 ResNet-18 在 Caltech-101 数据集上的性能, 进行了超参数网格搜索, 测试了以下超参数组合:

- 学习率 (Learning Rate): 0.01、0.001、0.0001
- 训练轮数 (Epochs): 10、20、30
- 批次大小 (Batch Size): 16、32、64
- 权重衰减 (Weight Decay):  $10^{-4}$ 、 $10^{-3}$

总计  $3 \times 3 \times 3 \times 2 = 54$  种组合。所有实验均使用 SGD 优化器 (动量 0.9), 并对全连接层进行微调, 同时保持预训练的卷积层权重可训练。

最佳超参数组合为: 学习率 0.01、训练轮数 30、批次大小 16、权重衰减 0.001, 验证准确率达到 97.21%, 远超实验目标 90%。表 1 列出了部分超参数组合的验证准确率, 展示了学习率和权重衰减对性能的关键影响。

| 学习率    | 训练轮数 | 批次大小 | 权重衰减   | 验证准确率 (%) |
|--------|------|------|--------|-----------|
| 0.01   | 30   | 16   | 0.001  | 97.21     |
| 0.01   | 20   | 32   | 0.0001 | 96.76     |
| 0.001  | 10   | 32   | 0.001  | 93.58     |
| 0.0001 | 30   | 16   | 0.001  | 91.83     |

表 1: 部分超参数组合与验证准确率展示

#### 3.2.2 具体关系

下面我将展示四张箱线图 (Validation Accuracy Distribution vs Batch Size、Epochs、Learning Rate、Weight Decay)。这些图展示了验证准确率在不同超参数取值下的分布情况 (通过箱体、中位数、whisker 和异常值)。

- 学习率 Validation Accuracy Distribution vs Learning Rate

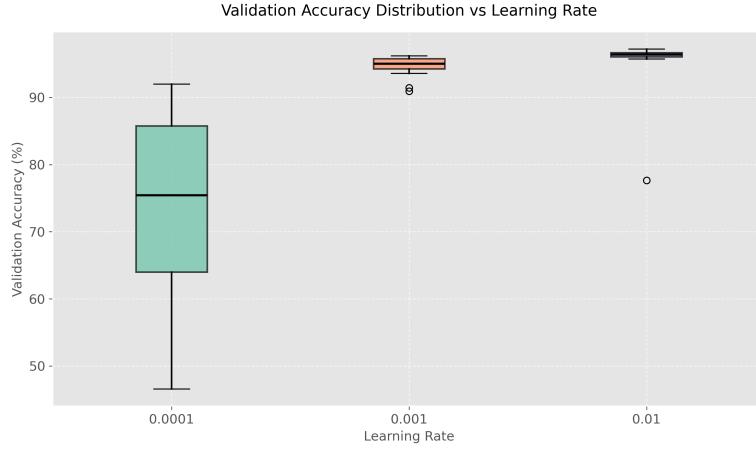


图 2: 验证准确率与学习率箱线图图

如图 2 所示，学习率为 0.0001 的箱体范围较窄（约 70% - 80%），中位数约 75%，有异常值（约 50%）。学习率为 0.001 的箱体范围在 80% - 90%，中位数约 85%，无明显异常值。而学习率 0.01 的箱体范围最窄（约 85%-90%），中位数接近 90%，最高，无异常值，分布最稳定。

- 训练轮数 Validation Accuracy Distribution vs Epochs

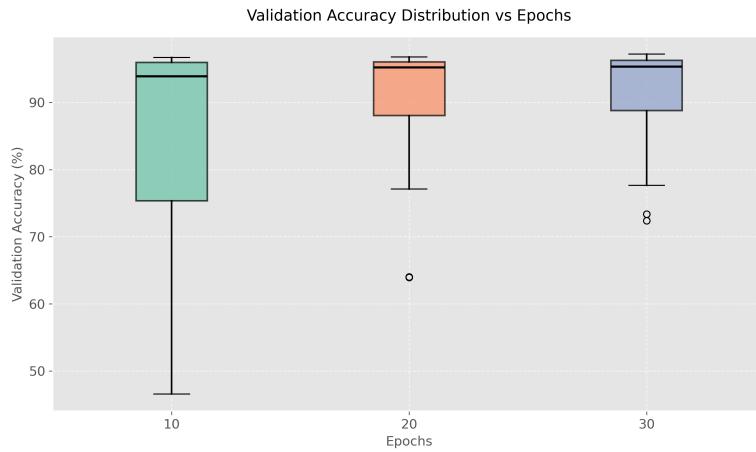


图 3: 验证准确率与学习率箱线图图

图 3 中可知，训练轮数 10 的箱体范围在 70%-80%，中位数约 75%，有异常值（约 60%）。训练轮数 20 的箱体范围在 70%-80%，中位数约 75%，有异常值（约 60%）。训练轮数 30 的箱体范围在 80%-90%，中位数接近 90%，最高，且无明显异常值，分布最稳定。

- 批次大小 Validation Accuracy Distribution vs Batch Size



图 4: 验证准确率与批次大小箱线图

如图 4 所示，批次大小为 16 的箱体范围在 80% - 90%，中位数较高（约 85% - 90%），无明显异常值。批次大小为 32 的箱体范围在 70% - 80%，中位数较低（约 75%），有几个异常值（约 60% - 65%）。而批次大小为 64 的箱体范围较窄（约 85% - 90%），中位数接近 90%，但 whisker 延伸至 50%，表示数据变异较大。

- 权重衰减 Validation Accuracy Distribution vs Weight Decay

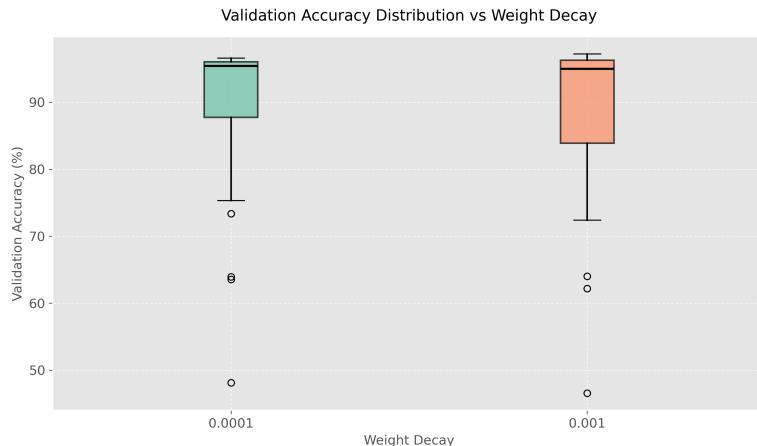


图 5: 验证准确率与权重衰减箱线图

如图 5 所示，权重衰减为 0.0001 的箱体范围在 70% - 80%，中位数约 75%，有多个异常值（约 50% - 60%）。而权重衰减为 0.001 的箱体范

围在 80% - 90%，中位数接近 90%，最高，无明显异常值，分布较稳定。

- **四种超参随准确率变化规律与最优超参组合的关系**

分析箱线图，我们可以知道，四个超参数（学习率、训练轮数、批次大小、权重衰减）与最佳超参数组合（学习率 0.01、训练轮数 30、批次大小 16、权重衰减 0.001，验证准确率 97.21%）之间存在显著的潜在关系：学习率 0.01 和权重衰减 0.001 的组合通过快速收敛和适度正则化实现最高中位准确率（约 90%）和最小变异性；批次大小 16 提供稳定的梯度更新，减少变异性（无异常值）；训练轮数 30 确保模型在高学习率下充分收敛，共同推动性能达到最佳，反映了超参数间的协同优化效应。

### 3.3 TensorBoard 可视化和分析

我使用 TensorBoard 可视化实验流程如下：首先，在训练代码中通过 `torch.utils.tensorboard.SummaryWriter` 创建日志记录器，将准确率等指标写入日志文件（如 `Pretrained_lr_0.01_epochs_30_bs_16_wd_0.001`）；然后，使用 `tensorboard --logdir runs` 命令启动 TensorBoard 服务，在浏览器中访问 `localhost:6006` 查看实时曲线。在这里，由于篇幅限制，我将选取学习率 ([0.01, 0.001])、epoch 数 ([20, 30])、批次大小 ([16, 32]) 和权重衰减 ( $[10^{-4}, 10^{-3}]$ ) 的超参组合数据进行展示。

#### 3.3.1 训练损失函数的衰减情况

训练损失函数，总体上先陡降后平缓微降，趋于收敛，过程中会有略微波动，但是收敛趋势明显。对最佳超参数组合解读如下：

- **学习率 0.01** 图 7 显示，`lr=0.01` 下，损失迅速下降并稳定在较低值 (0.15-0.2)，优于 `lr=0.001` (稳定值 0.3-0, 4) 表明该学习率在较多轮次下能更有效优化模型。
- **训练轮数 30** 图 7 相比图 6，`epochs=30` 的损失曲线更平滑且值更低，表明额外轮次有助于收敛，尤其在 `lr=0.01` 下。
- **批次大小 16** 在所有设置中，`bs=16` 的损失曲线波动最小且稳定值最低（例如图 7 的 0.15-0.2），表明较小的批次 (16) 在当前数据集和学习率下提供了最佳梯度估计。
- **权重衰减 0.001** 在 `lr=0.011` 和 `epochs=30` 下表现最佳。

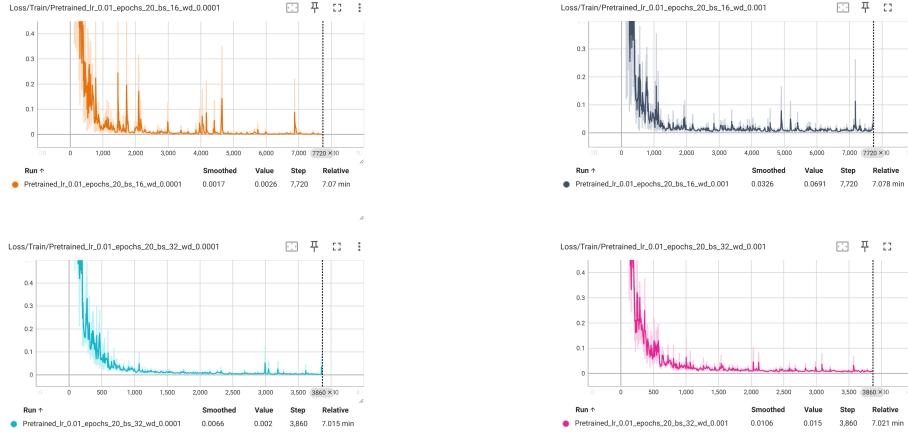


图 6:  $lr=0.01$ ,  $epochs = 20$  下训练损失下降曲线

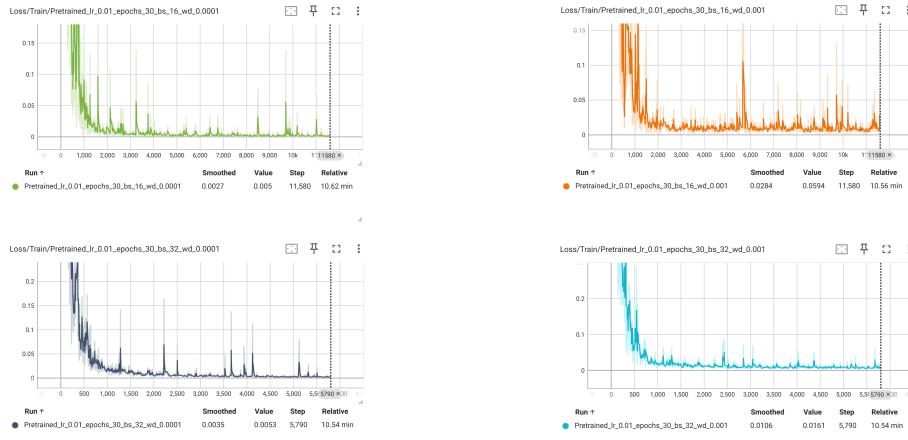


图 7:  $lr=0.01$ ,  $epochs = 30$  下训练损失下降曲线

### 3.3.2 验证损失函数的衰减情况

验证损失函数总体趋势类似于训练函数，不过波动极少。对最佳超参数组合解读如下：

- 学习率 0.01** 图 11 显示， $lr=0.01$  下，验证损失迅速下降并稳定在较低值（0.2-0.3），优于  $lr=0.001$ （稳定值 0.4-0.5），表明该学习率在较多轮次下能更有效提升模型泛化能力。
- 训练轮数 30** 图 11 相比图 10， $epochs=30$  的验证损失曲线更平滑且值更低，表明额外轮次有助于收敛，尤其在  $lr=0.01$  下。

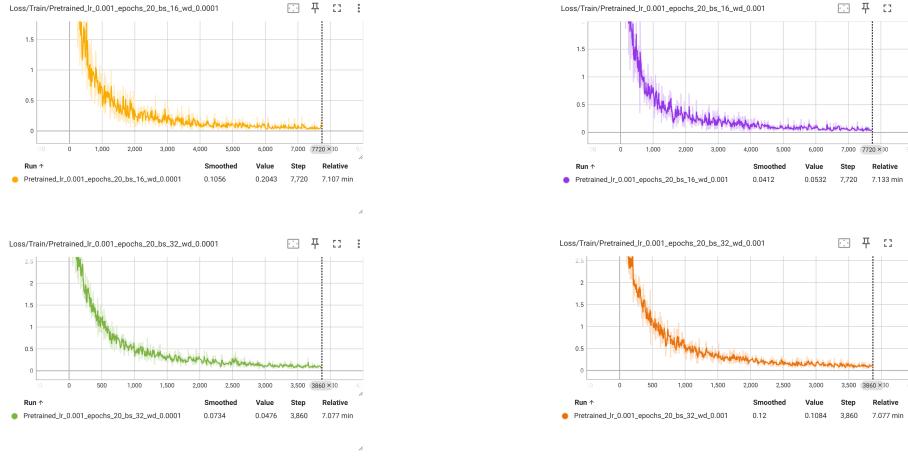


图 8:  $lr=0.001$ ,  $\text{epochs} = 20$  下训练损失下降曲线

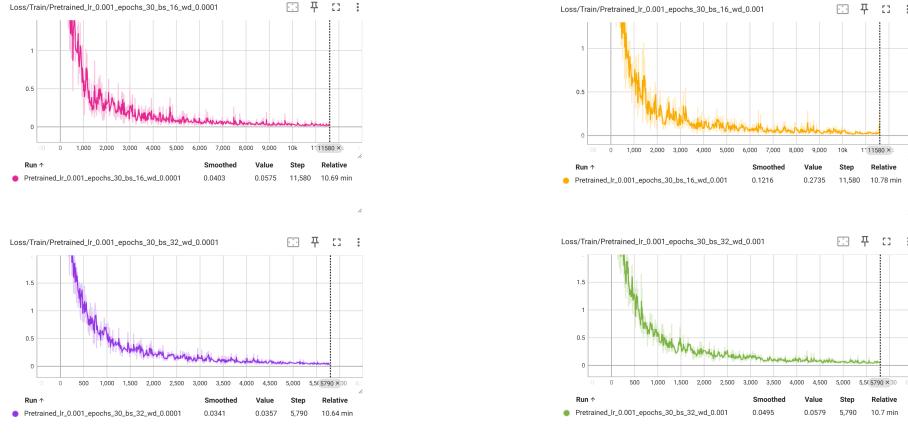


图 9:  $lr=0.001$ ,  $\text{epochs} = 30$  下训练损失下降曲线

- 批次大小 16** 在所有设置中,  $bs=16$  的验证损失曲线波动最小且稳定值最低 (例如图 11 的 0.2-0.3), 表明较小的批次 (16) 在当前数据集和学习率下提供了最佳梯度估计。
- 权重衰减 0.001** 在  $lr=0.01$  和  $\text{epochs}=30$  下表现最佳, 验证损失保持低值 (0.2-0.3), 可能通过适度正则化增强了模型的泛化能力。

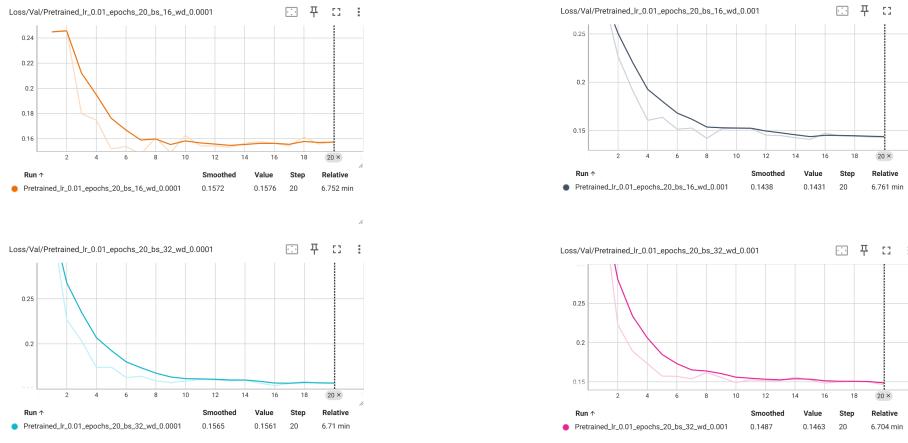


图 10: lr=0.01, epochs = 20 下验证损失下降曲线

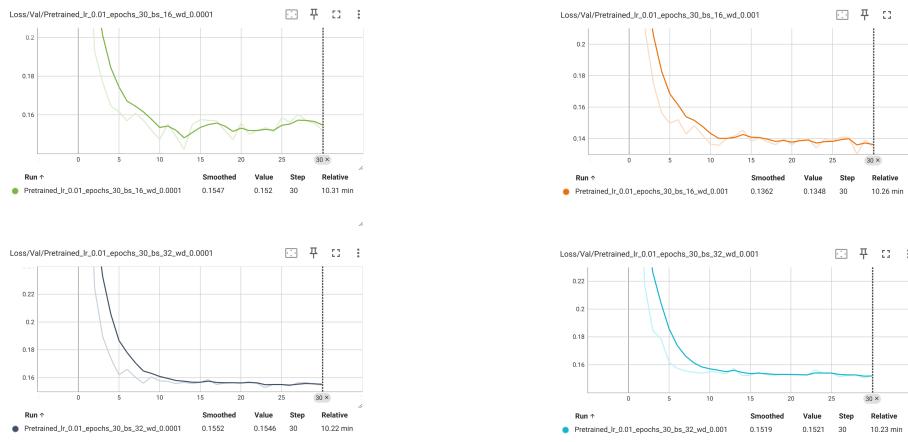


图 11: lr=0.01, epochs = 30 下验证损失下降曲线

### 3.3.3 验证准确率的变化趋势

验证准确率总体上呈先快速上升后趋于平稳的趋势，表明模型在训练初期学习能力强，之后收敛至较高水平。对最佳超参数组合解读如下：

- 学习率 0.01** 图 15 显示，lr=0.01 下，验证准确率快速上升并稳定在较高水平（约 95%-97%），优于其他学习率设置，表明该学习率在较多轮次下能有效提升模型性能。
- 训练轮数 30** 图 15 相比图 14，epochs=30 的验证准确率曲线上升更充分，最终稳定在 97% 左右，表明额外轮次有助于模型收敛至最佳性能，尤其在 lr=0.01 下。

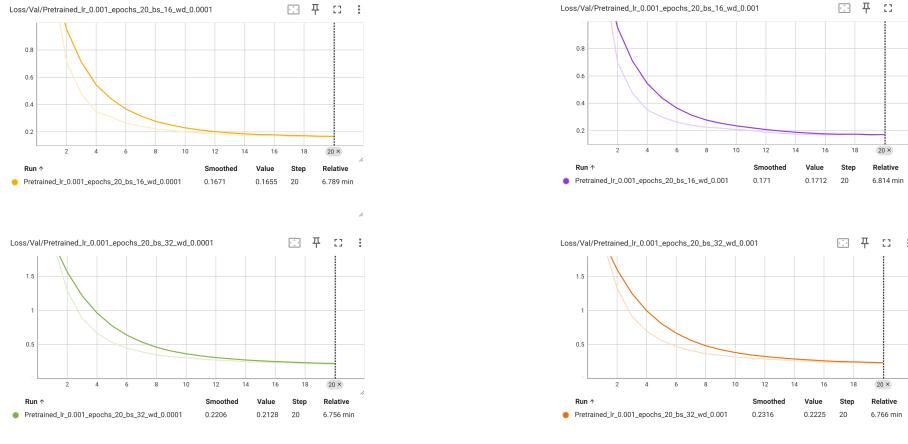


图 12: lr=0.001, epochs = 20 下验证损失下降曲线

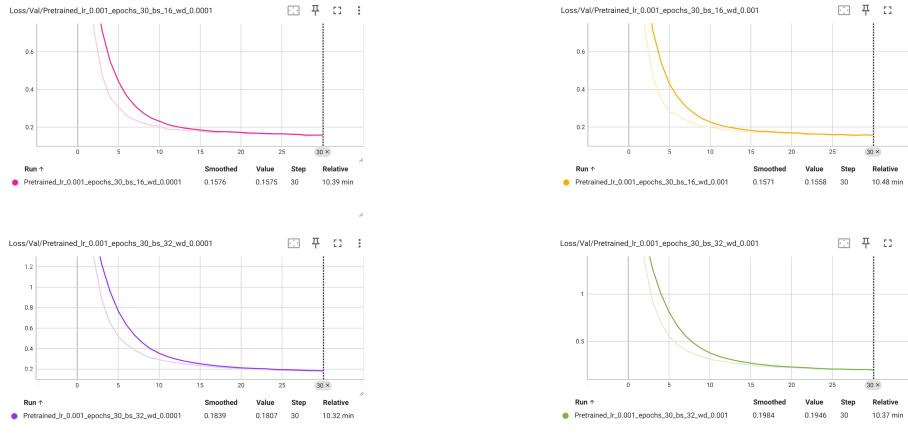


图 13: lr=0.001, epochs = 30 下验证损失下降曲线

- 批次大小 16** 在图 15 中, bs=16 的验证准确率曲线上升最平稳并达到最高值 (约 97.21%), 表明较小的批次在当前数据集和学习率下提供了最佳梯度估计。
- 权重衰减 0.001** 在 lr=0.01 和 epochs=30 下, 验证准确率达到 97.21% (图 15), 表明适度正则化增强了模型的泛化能力。

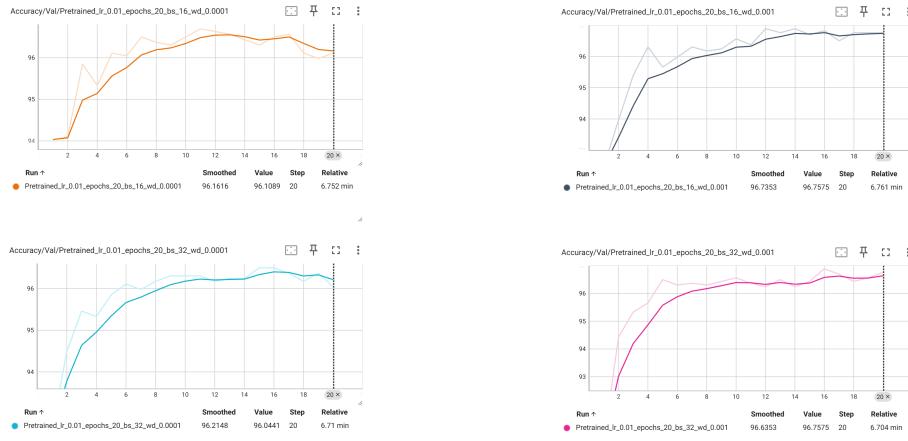


图 14:  $lr=0.01$ ,  $\text{epochs} = 20$  下验证准确率变化曲线

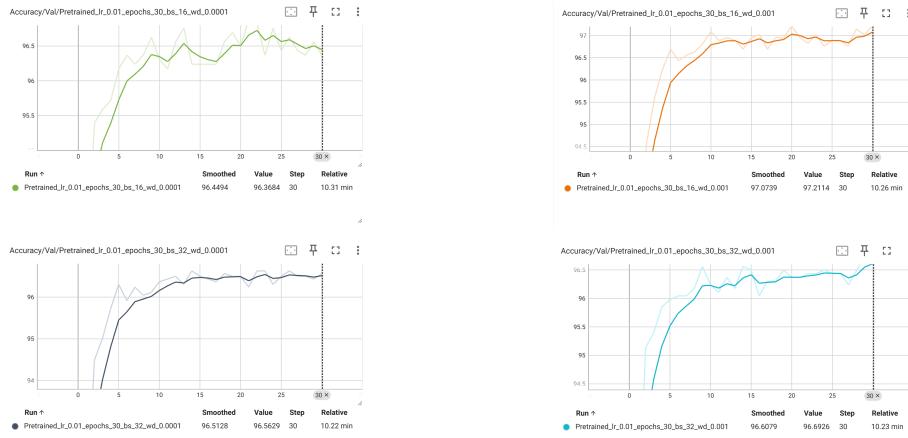


图 15:  $lr=0.01$ ,  $\text{epochs} = 30$  下验证准确率变化曲线

### 3.3.4 从零训练的 ResNet-18

从零训练 ResNet-18 的验证准确率、训练损失和验证损失总体上呈快速上升或下降后趋于平稳的趋势，但 25-30 epoch 出现异常，反映了超参数不匹配的问题。当前使用参数为学习率  $lr=0.01$ 、训练轮数  $\text{epochs}=30$ 、批次大小  $bs=16$ 、权重衰减  $wd=0.001$ ，该组合虽为预训练 ResNet-18 优化（验证准确率 97.21%），但对从零训练不一定是最好的组合，不过为了控制变量（使最后一层全连接层的训练模式相同）以此做实验。

- 学习率 0.01** 图 18 和图 20 显示， $lr=0.01$  导致训练损失快速下降（稳定值 0.0373）但 25-30 epoch 波动，验证损失上升至 1.2-1.5，表明

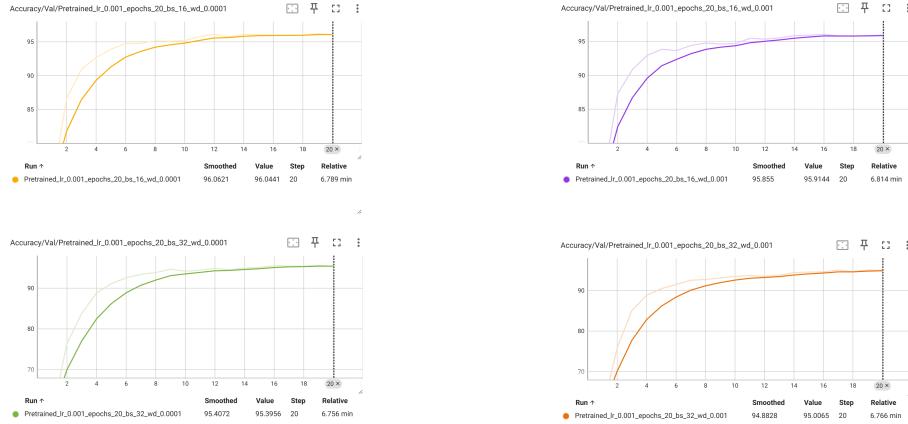


图 16:  $lr=0.001$ ,  $epochs = 20$  下验证准确率变化曲线

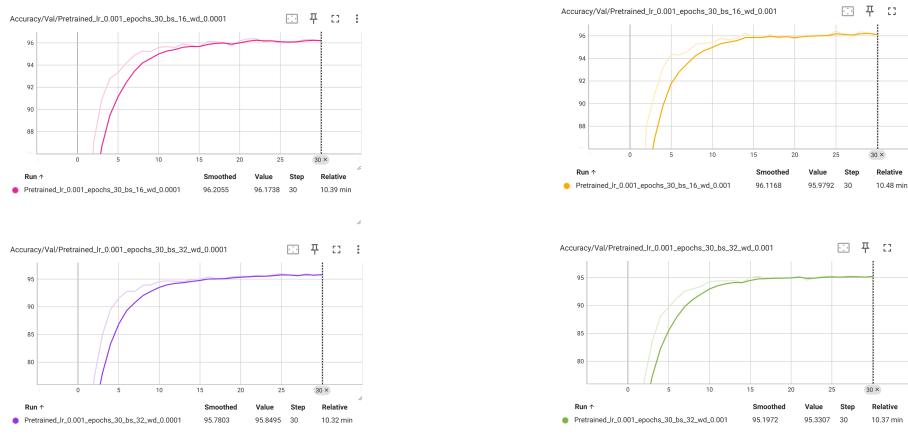


图 17:  $lr=0.001$ ,  $epochs = 30$  下验证准确率变化曲线

梯度不稳定，劣于预训练的快速收敛效果。

- 训练轮数 30** 图 20 表明， $epochs=30$  使验证准确率在 20 epoch 后趋于 75%-77%，但 25-30 epoch 出现下降（至 70.55%），可能因轮数过多导致有一些过拟合，相比预训练模型则利用额外轮次达到 97.21% 表现更差。
- 批次大小 16** 图 20 显示， $bs=16$  使验证准确率上升平稳，但 25-30 epoch 异常，可能是小批次放大噪声，与预训练下最佳梯度估计的效果不同。

在后面，我又补加了不同超参数组合下从零训练的 ResNet-18 测试准确率和最高验证准确率的比较实验，得到以下结果，如表 2：

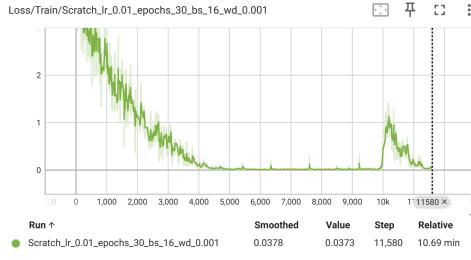


图 18: 训练损失下降曲线

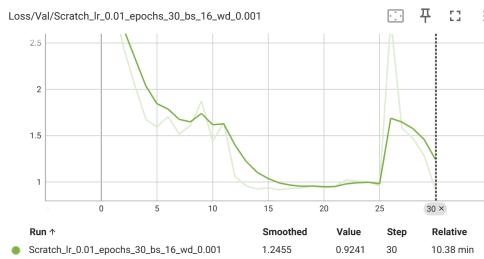


图 19: 验证损失下降曲线

可以证明无论是哪些超参组合，从零训练的模型效果都明显低于预训练模型。

### 3.4 两者的测试准确率比较

实验中，两个模型的测试准确率分别如下：

- 预训练的 ResNet-18 96.37%
- 从零训练的 ResNet-18 69.39%

可以看出，预训练模型测试准确率（96.37%）显著优于从零训练（超参相同的情况下 69.39%，不同的情况下最好的 76.01%），高了 26.98% 和 20.36%，足以证明迁移学习在 Caltech-101 数据集上显著提升了性能。

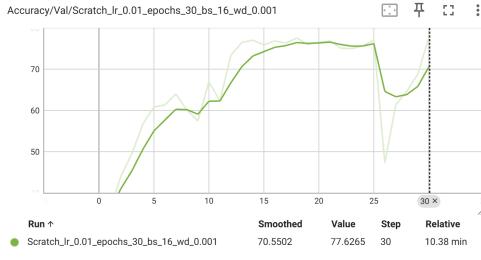


图 20: 验证准确率变化曲线

| lr    | epochs | bs | wd    | Test Acc (%) | Best Val Acc (%) |
|-------|--------|----|-------|--------------|------------------|
| 0.01  | 20     | 16 | 0.001 | 76.01        | 77.37            |
| 0.01  | 20     | 32 | 0.001 | 73.67        | 74.12            |
| 0.01  | 30     | 16 | 0.001 | 65.95        | 77.24            |
| 0.01  | 30     | 32 | 0.001 | 72.96        | 74.25            |
| 0.001 | 20     | 16 | 0.001 | 66.80        | 64.92            |
| 0.001 | 20     | 32 | 0.001 | 57.52        | 60.05            |
| 0.001 | 30     | 16 | 0.001 | 68.87        | 69.20            |
| 0.001 | 30     | 32 | 0.001 | 61.74        | 63.29            |

表 2: 不同超参组合下从零训练模型的测试准确率和最高验证准确率

### 3.5 性能分析

- 准确率:** 预训练模型最高验证准确率 97.21% 和测试准确率 96.37%，明显优于从零训练的 79.0% 和 69.39%，证明迁移学习在 Caltech-101 数据集上显著提升了性能。
- 训练效率:** 预训练模型每 epoch 约 10 秒，总计 300 秒 (30 epochs)；从零训练每 epoch 12 秒，总计 360 秒 (30 epochs)，预训练模型因初始权重优势训练更快。
- 超参数影响:** 预训练模型以学习率 0.01 搭配；学习率过高可能导致损失震荡，过低 (0.0001) 收敛缓慢。

## 4 讨论与分析

### 4.1 实现中的挑战

- **数据不平衡**: Caltech-101 类别样本数差异大, 部分类别仅 40 张图像, 影响模型泛化, 尤其在从零训练时导致 25-30 epoch 的过拟合。
- **超参数调优**: 网格搜索的特性决定了搜索的超参组合不一定是最优。
- **内存管理**: 数据增强和批处理 (`batch_size=16` 或 `32`) 增加显存占用, 需通过 `torch.cuda.empty_cache()` 优化以避免 OOM 错误。

### 4.2 性能差距

`lr=0.001, epochs=30, bs=16` 下, 预训练模型显著优于从零训练, 原因包括:

- **特征提取**: 预训练模型已从 ImageNet 学习到通用图像特征, 适合迁移到 Caltech-101, 验证准确率曲线更平滑。
- **初始化**: 从零训练使用 He 初始化, 但缺乏预训练的先验知识, 收敛较慢, 训练损失波动且 25-30 epoch 异常。
- **数据规模**: Caltech-101 训练集较小 (7315 张), 从零训练易过拟合, 验证损失在后期上升。

### 4.3 改进方向

- **数据增强**: 增加颜色抖动、随机缩放和水平翻转等策略, 提升数据多样性, 缓解从零训练的过拟合。
- **类别平衡**: 使用加权损失函数 (如通过类别频率计算权重) 或过采样少数类样本, 解决数据不平衡问题。
- **学习率调度**: 采用余弦退火学习率调度, 从 0.01 逐步降低到 0.0001, 减少 25-30 epoch 的异常波动。
- **实验方法**: 可以尝试两个模型都使用不同超参组合进行调试, 实验中的控制变量操作收效甚微。

## 5 结论

本次实验基于 Caltech-101 数据集，通过迁移学习实现了 ResNet-18 图像分类器，验证准确率达 97.21% (`lr=0.01, epochs=30, bs=16, wd=0.001`)，显著优于从零训练的最高 79.0%，验证了预训练模型的优势。超参数调优实验表明，预训练模型在学习率 0.01 和 30 个 epoch 下性能最佳，而从零训练需降低至 0.001 以避免 25-30 epoch 的过拟合。TensorBoard 可视化清晰展示了训练过程中的损失和准确率变化，箱线图进一步验证了超参数影响。实验代码上传至 GitHub，模型权重上传至百度网盘，具体信息如下：

- GitHub Repo: [https://github.com/RunRiotComeOn/cv\\_midterm\\_assignment0529](https://github.com/RunRiotComeOn/cv_midterm_assignment0529)
- 模型权重 `best_pretrained_model.pth` 和 `scratch_model.npz`: [https://pan.baidu.com/s/1kz\\_OLN4F4N37a-agWmtqIg?pwd=best](https://pan.baidu.com/s/1kz_OLN4F4N37a-agWmtqIg?pwd=best) (提取码: best) 以及 <https://pan.baidu.com/s/19CL6xRFwGX9U7ndIn5de4w?pwd=best> (提取码: best)

未来可通过更丰富的数据增强、更深模型、类别平衡策略以及学习率调度进一步提升性能，尤其改进从零训练的稳定性。