

# HW1 实验报告

## CIFAR-10 图像分类与动量优化对比

刘继轩

学号：2400017722

2025 年 9 月 16 日

### 1 实验概述

本实验主要包含两个部分：

1. **Part 1** 使用 PyTorch 框架构建卷积神经网络 (CNN) 对 CIFAR-10 数据集进行图像分类训练。
2. **Part 2** 对比在 SGD 优化器中不同动量 (momentum) 参数值对模型训练效果的影响。

### 2 数据集介绍

CIFAR-10 是一个经典的图像分类数据集，包含 10 个类别的彩色图像：

- 类别：飞机 (plane)、汽车 (car)、鸟 (bird)、猫 (cat)、鹿 (deer)、狗 (dog)、青蛙 (frog)、马 (horse)、船 (ship)、卡车 (truck)
- 图像尺寸：32×32 像素，3 通道 RGB 彩色图像
- 训练集：50,000 张图像
- 测试集：10,000 张图像

### 3 模型架构

#### 3.1 网络结构

本实验采用了经典的 LeNet 变体架构，具体结构如下：

表 1: cifarNet 网络结构

层类型	参数配置	输出尺寸
输入	-	$3 \times 32 \times 32$
卷积层 1	$3 \rightarrow 6$ , kernel= $5 \times 5$	$6 \times 28 \times 28$
最大池化 1	kernel= $2 \times 2$ , stride=2	$6 \times 14 \times 14$
卷积层 2	$6 \rightarrow 16$ , kernel= $5 \times 5$	$16 \times 10 \times 10$
最大池化 2	kernel= $2 \times 2$ , stride=2	$16 \times 5 \times 5$
展平层	-	400
全连接层 1	$400 \rightarrow 120$	120
全连接层 2	$120 \rightarrow 84$	84
全连接层 3	$84 \rightarrow 10$	10

### 3.2 数据预处理

使用 torchvision 中的 datasets 下载 cifar-10 数据集，并利用 transforms 对输入图像进行了以下预处理：

1. 转换为张量格式 (ToTensor)。
2. 像素值归一化 (Normalize)：使用均值 0.5 和标准差 0.5 进行归一化，将像素值范围从  $[0,1]$  映射到  $[-1,1]$ 。

### 3.3 超参数设置

- 学习率 (learning rate): 0.001
- 批量大小 (batch size): 8
- 训练轮数 (epochs): 10
- 优化器: SGD, 动量 (momentum) 参数在实验中进行调整。

### 3.4 其他设计细节

- 所有隐藏层（卷积层和全连接层）使用 ReLU 激活函数。
- 使用交叉熵损失函数 (CrossEntropyLoss) 作为多分类任务的损失函数。
- 使用随机梯度下降 (SGD) 作为优化器。

## 4 Part 1 基础实验结果

在 Part 1 中，我们使用 ‘momentum=0.9’ 进行训练。训练 10 个 epoch 后，模型在测试集上的整体准确率为 62.51%。各类别的准确率如下：

类别	准确率 (%)
plane	56.20
car	81.00
bird	54.60
cat	42.50
deer	66.30
dog	35.50
frog	69.30
horse	69.50
ship	77.10
truck	73.10

图 1: 各类别准确率

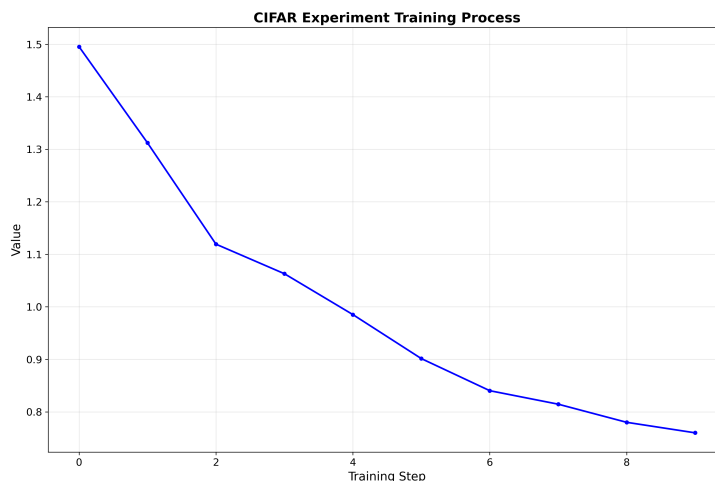


图 2: 训练损失曲线 (momentum=0.9)

从结果可以看出，模型已能有效拟合训练数据，损失函数稳步下降。然而，模型在“猫”和“狗”等类别上表现较差，这可能是由于这些类别内部差异较大，且当前模型结构简单，未采用数据增强、正则化等高级技巧，导致泛化能力有限。

## 5 Part 2 动量参数对比实验

动量 (momentum) 是 SGD 优化器中的一个重要超参数，它通过积累历史梯度信息来加速收敛并减少优化过程中的振荡。

### 5.1 实验设计

为研究动量参数对训练效果的影响，我们设计了对照实验。在固定随机数种子以保证实验可复现的前提下，我们测试了四种不同的动量值，并观察其训练损失曲线和最终测试准确率：

- momentum = 0.0 (无动量)
- momentum = 0.5 (中等动量)
- momentum = 0.9 (较高动量，常用值)
- momentum = 0.99 (极高动量)

## 5.2 动量对比实验结果

图3展示了不同动量值下的损失函数收敛曲线。

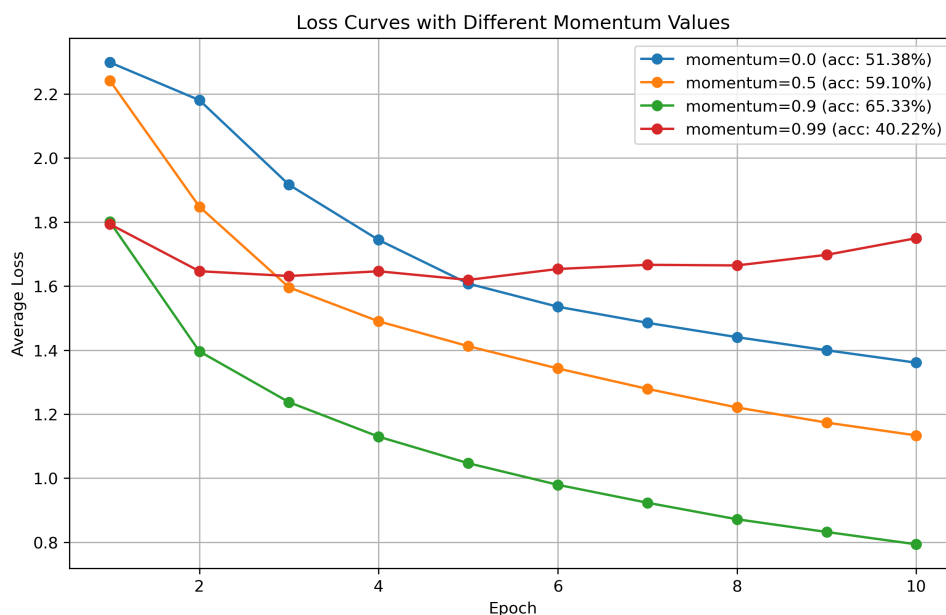


图 3: 不同动量值的损失函数对比

各动量值对应的最终测试准确率如表2所示。

表 2: 不同动量值的测试准确率对比

动量值	测试准确率 (%)
0.0	51.38
0.5	59.10
0.9	65.33
0.99	40.22

## 6 结果分析与讨论

### 6.1 动量对训练的影响

从实验结果可以观察到以下几个重要现象：

- 适中动量效果最佳**：‘momentum=0.9’时获得了最高的测试精度（65.33%），表现最好。
- 无动量收敛较慢**：‘momentum=0.0’时，损失函数下降最为缓慢，最终精度也相对较低（51.38%）。

3. **过高动量可能有害**：‘momentum=0.99’时，虽然早期损失下降最快，但很快出现剧烈振荡，最终精度反而最低（40.22%）。这可能是因为过大的动量导致更新步长过大，使得优化器“冲过”了最优点，并在其附近来回震荡，难以收敛到更精细的局部最小值。
4. **收敛稳定性**：‘momentum=0.5’和‘0.9’都展现出了良好的收敛稳定性，损失函数持续下降。

## 6.2 理论解释

动量机制在 SGD 中的工作原理可以简化为以下公式：

$$v_t = \beta v_{t-1} + \nabla_{\theta} J(\theta_t) \quad (1)$$

$$\theta_{t+1} = \theta_t - \alpha v_t \quad (2)$$

其中， $\beta$  是动量系数 (momentum)， $\alpha$  是学习率， $v_t$  是当前时刻的动量（梯度的指数移动平均）， $\nabla_{\theta} J(\theta_t)$  是当前参数的梯度。

- **动量的优势**：当梯度方向连续保持一致时，动量项会累积，从而加速收敛；当梯度方向变化时，动量项可以抵消部分振荡，使优化路径更平滑。
- **过高动量的问题**：过大的  $\beta$  值意味着历史梯度的影响过重，可能导致更新步长太大而“冲过”最优点 (overshooting)，使其难以在最优解附近稳定下来。
- **最优选择**：通常 0.9 被认为是一个在多种任务中表现稳健的默认值，本实验的结果也支持了这一经验。

# 7 实验总结

## 7.1 主要收获

1. 成功实现了基于 CNN 的 CIFAR-10 图像分类任务，在简单模型下达到了 65.33% 的测试精度。
2. 通过对比实验，深入理解了动量参数对深度学习模型训练速度和最终性能的重要影响。
3. 掌握了 PyTorch 框架的核心使用流程，包括数据加载、模型定义、训练循环和模型评估。
4. 学会了使用 TensorBoard 对训练过程（如损失曲线）进行可视化，为模型调试和分析提供了有力工具。