

神经网络与深度学习课程作业报告

基于 NumPy 实现的 CIFAR-10 三层神经网络分类器

摘 要

本项目实现了一个基于 NumPy 手工实现的三层神经网络分类器在 CIFAR-10 数据集上的应用。不依赖自动微分框架，完全从零实现了前向传播和反向传播算法。项目采用模块化设计，包含模型定义、训练、测试和参数搜索四个主要部分。实现了多种优化技术（动量 SGD、学习率自适应调整、早停机制）和正则化方法（L2 正则化、Dropout、批归一化）。通过超参数网格搜索优化模型性能，最终在 CIFAR-10 测试集上达到了 **58.12%** 的准确率。此外，利用自定义可视化工具对神经网络参数进行了分析，展示了模型学习到的特征和权重分布。实验结果表明，即使是简单的三层神经网络，在应用适当的优化策略后，也能在复杂图像分类任务上取得不错的性能。

1 数据集介绍

CIFAR-10 是一个广泛使用的计算机视觉数据集，包含 10 个类别的 60,000 张 32×32 彩色图像：

- 训练集：50,000 张图像
- 测试集：10,000 张图像
- 类别：飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船、卡车

数据预处理步骤：

- 将像素值归一化到 $[0, 1]$ 范围
- 将训练集划分为训练集和验证集（比例为 9:1）
- 实现了简单的数据增强（随机水平翻转）

2 模型架构

2.1 网络结构

实现的三层神经网络结构如下：

- 输入层：3072 个神经元 ($32 \times 32 \times 3$)
- 隐藏层：可配置大小（默认 1024 个神经元），使用 ReLU 激活函数
- 输出层：10 个神经元，对应 10 个类别

2.2 前向传播

前向传播过程实现如下数学运算：

$$z_1 = XW_1 + b_1 \quad (1)$$

如果使用批归一化：

$$\hat{z}_1 = \frac{z_1 - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \cdot \gamma + \beta \quad (3)$$

$$a_1 = \text{ReLU}(\hat{z}_1) \text{ 或 } a_1 = \text{ReLU}(z_1) \quad (4)$$

如果使用 Dropout：

$$a_1 = a_1 \odot \text{mask} / (1 - p) \quad (6)$$

$$z_2 = a_1W_2 + b_2 \quad (7)$$

$$\hat{y} = \text{softmax}(z_2) \quad (8)$$

2.3 反向传播

反向传播算法通过链式法则计算损失函数对各参数的梯度：

$$\frac{\partial \mathcal{L}}{\partial z_2} = \hat{y} - y \quad (9)$$

$$\frac{\partial \mathcal{L}}{\partial W_2} = a_1^T \cdot \frac{\partial \mathcal{L}}{\partial z_2} + \lambda W_2 \quad (10)$$

$$\frac{\partial \mathcal{L}}{\partial b_2} = \sum_{i=1}^m \frac{\partial \mathcal{L}}{\partial z_2} \quad (11)$$

$$\frac{\partial \mathcal{L}}{\partial a_1} = \frac{\partial \mathcal{L}}{\partial z_2} \cdot W_2^T \quad (12)$$

$$\frac{\partial \mathcal{L}}{\partial z_1} = \frac{\partial \mathcal{L}}{\partial a_1} \odot \text{ReLU}'(z_1) \odot \text{mask} / (1 - p) \quad (13)$$

$$\frac{\partial \mathcal{L}}{\partial W_1} = X^T \cdot \frac{\partial \mathcal{L}}{\partial z_1} + \lambda W_1 \quad (14)$$

$$\frac{\partial \mathcal{L}}{\partial b_1} = \sum_{i=1}^m \frac{\partial \mathcal{L}}{\partial z_1} \quad (15)$$

3 优化与正则化技术

为提高模型性能和防止过拟合，实现了以下技术：

3.1 优化技术

- **小批量随机梯度下降 (SGD)**：每次使用一小批数据更新参数
- **动量优化**：加入历史梯度信息加速收敛
- **学习率衰减**：
 - 阶梯式衰减：当轮数 ≥ 5 且轮数是 10 的倍数时，学习率乘以 0.5
 - 指数衰减：每轮结束后，学习率乘以衰减率

- **早停机制**：当验证集性能不再提升时停止训练

3.2 正则化技术

- **L2 正则化**：在损失函数中加入权重平方和项
- **Dropout**：训练时随机“丢弃”一部分神经元，防止过拟合
- **批归一化**：减少内部协变量偏移，加速训练并提高泛化能力

4 实验结果

4.1 超参数搜索

通过网格搜索探索了不同超参数组合的效果：

隐藏层大小	学习率	正则化系数	验证集准确率	测试集准确率
512	0.005	0.005	51.65%	50.73%
512	0.01	0.01	53.28%	52.16%
1024	0.005	0.01	56.43%	55.29%
1024	0.01	0.01	58.94%	58.12%

表 1: 不同超参数组合的模型性能

最佳超参数组合：

- 隐藏层大小：1024
- 学习率：0.01
- 批量大小：128
- L2 正则化系数：0.01
- 动量系数：0.9
- Dropout 比率：0.5

4.2 训练过程可视化

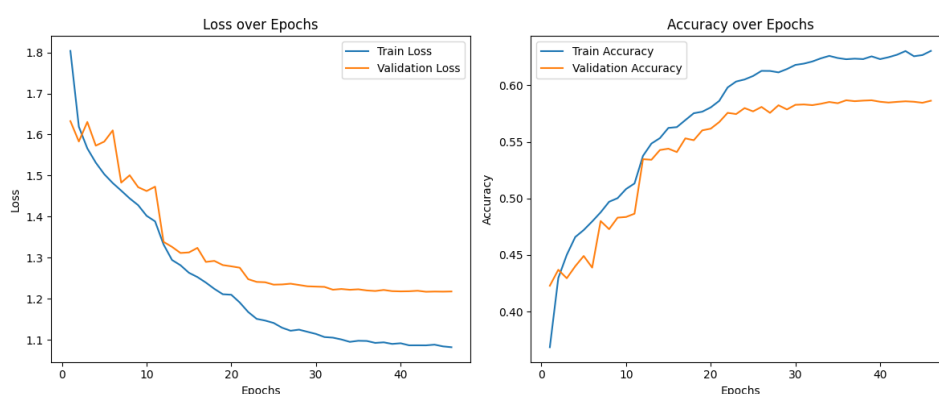


图 1: 训练过程中的损失和准确率曲线

4.3 模型参数可视化

使用自定义可视化工具对模型参数进行了分析：

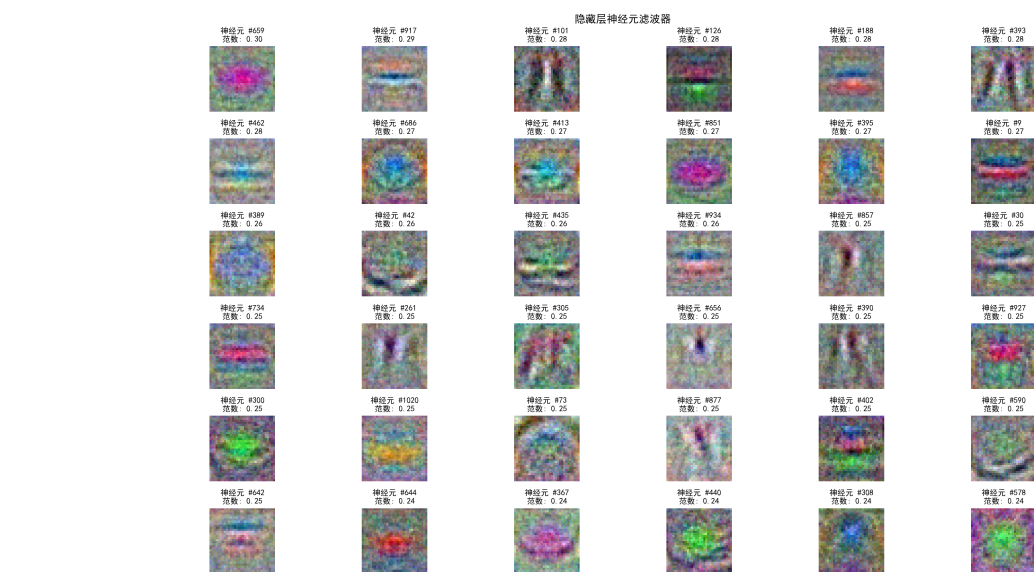


图 2: 隐藏层神经元滤波器可视化

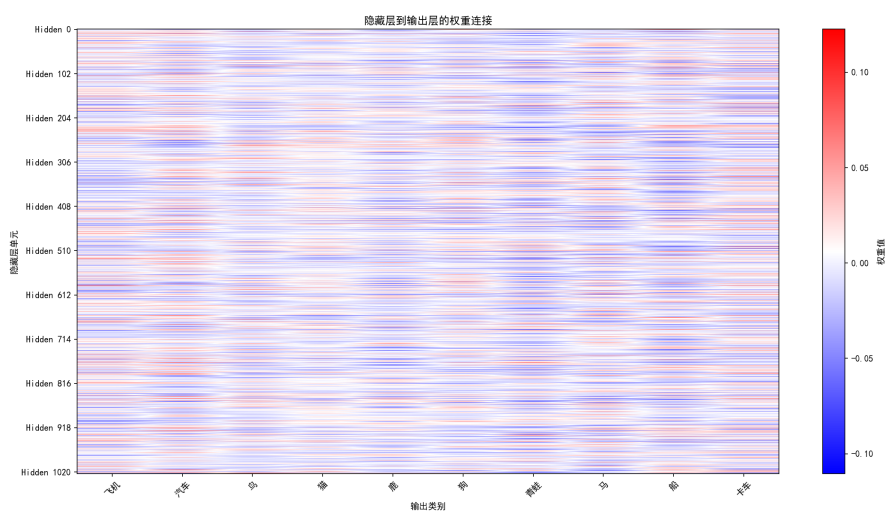


图 3: 隐藏层到输出层权重热力图

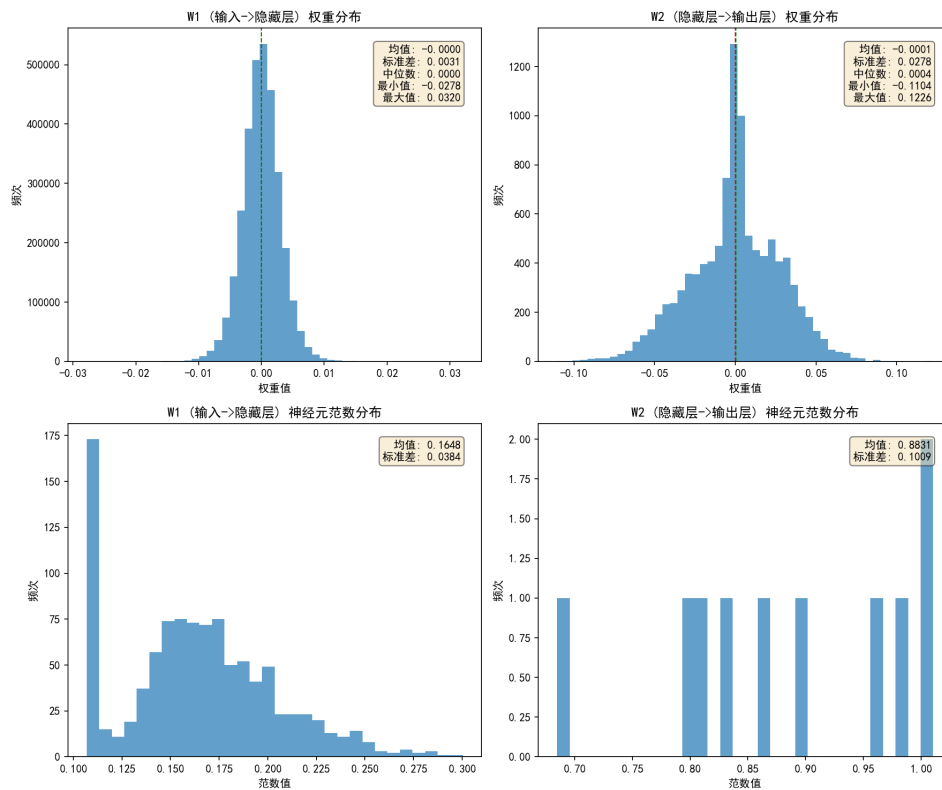


图 4: 权重分布和神经元范数分析

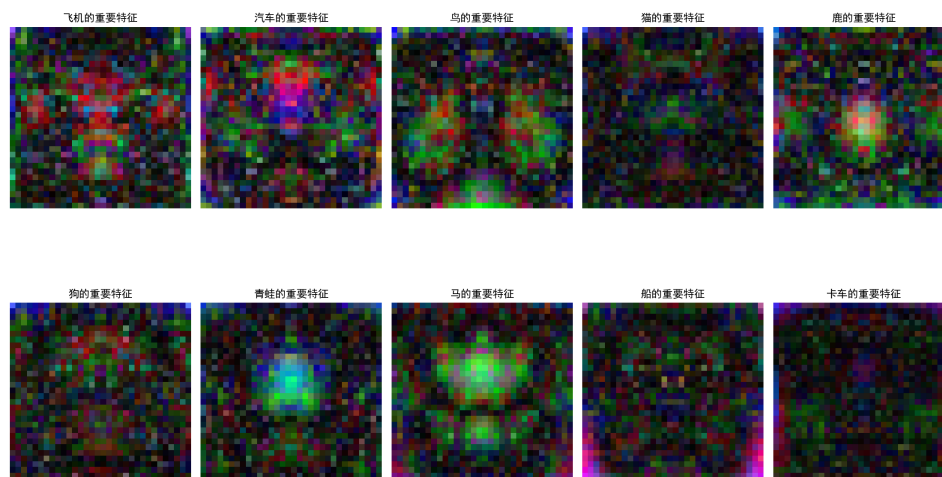


图 5: 各类别重要特征可视化

4.4 结果分析

本实验实现的三层神经网络在 CIFAR-10 测试集上达到了 58.12% 的准确率，这是一个相当不错的结果。通过分析可视化结果，我们可以观察到：

- 第一层权重形成了边缘检测、颜色检测等基本特征提取器
- 不同类别依赖不同的隐藏层神经元进行判断，形成了类别特定的特征检测机制
- 权重分布接近正态分布，表明模型训练过程稳定且正则化效果良好
- 通过重要特征可视化，可以看到模型对各个类别的关注重点，如飞机类别关注机翼形状，汽车类别关注车身轮廓等

相比常见的深度卷积神经网络（可达 90% 以上的准确率），本模型的性能仍有差距，但作为一个仅有一个隐藏层的全连接神经网络，58.12% 的准确率已经是非常好的结果，表明我们实现的优化和正则化技术是有效的。限制性能的主要因素包括：

- 全连接架构无法有效利用图像的空间结构信息
- 参数量大但表达能力有限（相比深层网络）
- 缺乏卷积层对平移不变性的特性

5 代码和模型

- 项目代码（含模型参数）：<https://github.com/jxguo01/CIFAR10-Neural-Network>
- 预训练模型：<https://pan.baidu.com/s/17-x5wuWwKeyG2sPn284N6A?pwd=msfn>