

三层神经网络分类器实验报告

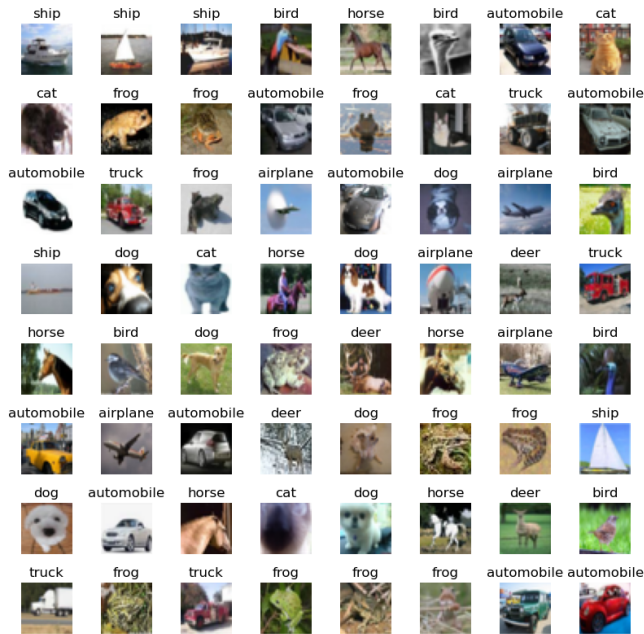
1. 引言

本项目旨在从零开始构建一个三层神经网络分类器实现 CIFAR-10 数据分类任务。该神经网络分类器的构建过程不依赖于任何现成的深度学习框架，仅使用 NumPy 进行数值计算。本报告将详细介绍模型的设计、实现和实验结果。

2. 方法

1) 数据集介绍

本项目使用 CIFAR-10 数据集，该数据集包含 10 个类别的 60000 张 32x32 彩色图像。其中，训练集包含 50000 张，测试集包含 10000 张。



2) 模型架构

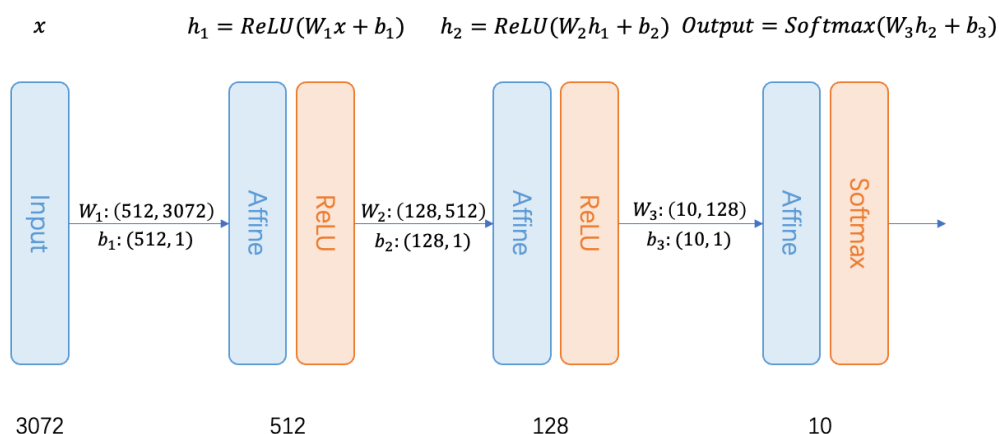
本项目构建了一个三层神经网络，包括一个输入层（3072）、两个隐藏层（[512, 128]）和一个输出层（10）。隐藏层使用 ReLU 激活函数，输出层使用 Softmax 激活函数。

输入层： x

第一个隐藏层输出： $h_1 = \text{ReLU}(W_1x + b_1)$

第二个隐藏层输出： $h_2 = \text{ReLU}(W_2h_1 + b_2)$

输出层： $\text{Output} = \text{Softmax}(W_3h_2 + b_3)$



3) 损失函数

该分类模型采用交叉熵损失函数，用于衡量模型预测的概率分布与真实标签之间的差异。损失函数公式如下：

$$L(y, \hat{y}) = - \sum_{i=1}^n y_i \log(\hat{y}_i)$$

在本次实验中，交叉熵损失函数体现在 `SoftmaxWithLoss` 类的定义中（`Layers.py`）。

4) 防止过拟合

在实验中为防止出现过拟合现象，采用了 L2 正则化、批量标准化、dropout 等设定。

L2 正则化通过在损失函数中添加权重的平方和来防止过拟合。这有助于保持权重值较小，从而减少模型的复杂度。L2 正则化的损失函数表示如下，其中取 $\lambda = 0.1$ 进行模型训练：

$$L_{regularized} = L + \lambda \sum_{l=1}^L \|W_l\|^2$$

本实验中 L2 正则化体现在 `MultiLayerNet` 类的 `loss` 方法中。

批量标准化通过在每一层的输出上进行标准化处理，以减少内部协变量偏移，从而加速训练过程并防止过拟合。

Dropout 通过在训练过程中随机丢弃一部分神经元来防止过拟合，在训练中设置丢弃比例为 0.2。

5) 优化器

本次实验使用 SGD 随机梯度下降法训练模型，将训练集数据划分成大小为 50 的 mini batch 进行训练。训练中设置学习率为 0.01，学习率下降比例 0.95，每 5 个 epoch 应用一次学习率下降。

6) 数据预处理

从原始训练集中划分训练集与验证集，训练集包含 40000 个样本，而验证集包含 10000 个。

考虑到 CIFAR10 数据集是彩色图片集 (40000, 32, 32, 3)，在输入模型前改造成一维数据。因此输入模型的训练数据大小是 (40000, 3072)。

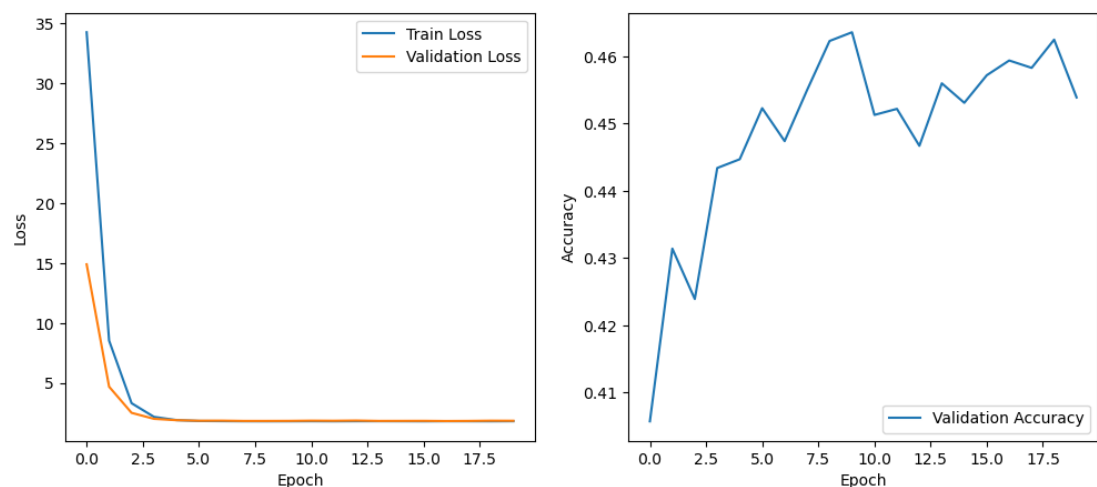
7) 训练过程

训练过程包括前向传播、计算损失、反向传播和参数更新。优化器使用 SGD，并实现了学习率下降。

3. 实验结果

1) 训练

在训练过程中记录了损失和准确率的变化。下图展示了这些指标随训练轮次的变化情况。



- 在训练初期，损失值较高，随着训练的进行，损失值迅速下降，表明模型在学习并逐渐拟合训练数据。
- 在大约 5 个 epoch 之后，训练损失趋于平稳，说明模型已经学到了训练数据中的大部分信息。
- 验证损失在初期也迅速下降，但随后趋于平稳，这表明模型在验证集上的表现与训练集相似，没有明显的过拟合现象。

2) 测试

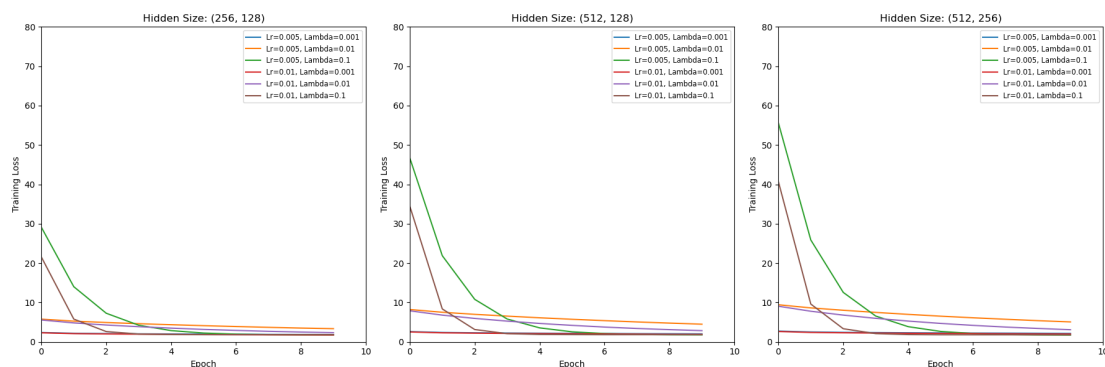
隐藏层大小 [512, 128]、学习率 0.01、L2 正则化强度 0.1 的模型在测试集上的准确率为 46.42%，准确率较低。

3) 超参数查找

在以下参数网格中进行参数查找，并记录最优模型：

- 学习率= $\{0.005, 0.01\}$
- 隐藏层大小= $\{[512, 256], [512, 128], [256, 128]\}$
- L2 正则化强度= $\{0.001, 0.01, 0.1\}$

查找结果如下图：

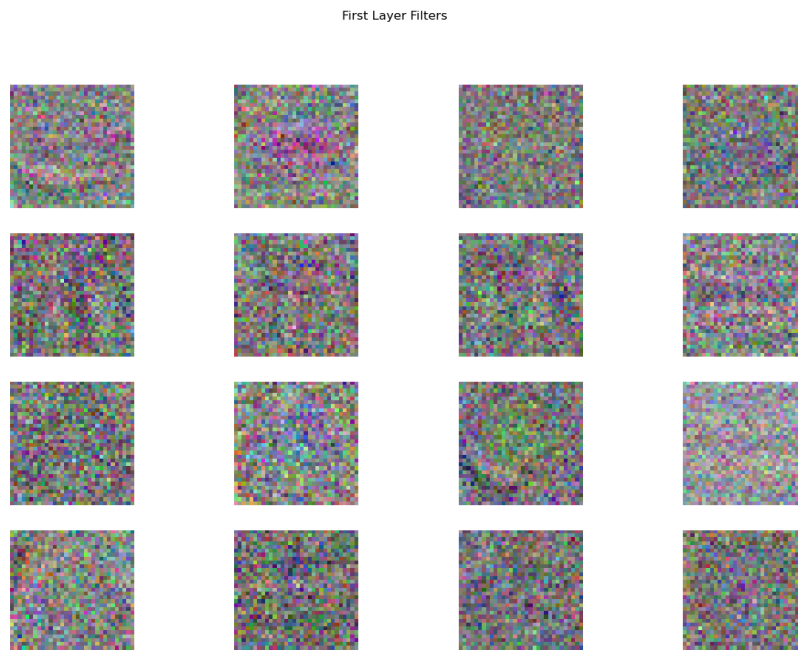


从三种隐藏层配置下的训练损失曲线可见，较低的学习率（如 $lr=0.005$ ）和较高的正则化强度（如 $\text{Lambda}=0.1$ ）组合通常能在训练初期更快地降低损失，表现较好。

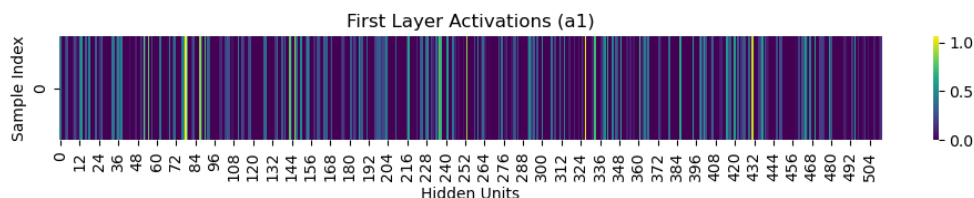
隐藏层的大小对模型的性能有显著影响，较大的隐藏层可能需要更精细的超参数调整来达到最佳性能。

经过比较，最佳的超参数组合为{学习率 0.01，隐藏层[512, 128]，正则化强度 0.01}。该组合下模型在测试集上准确率达到 49.48%。

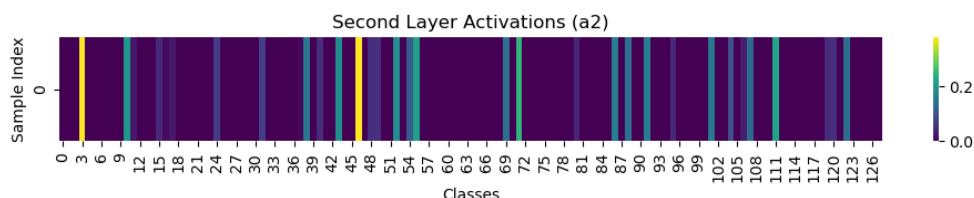
4) 参数可视化



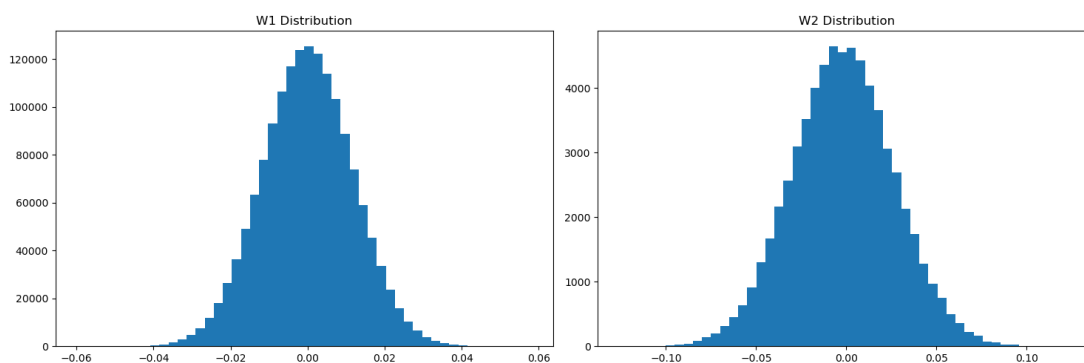
这张图展示了神经网络第一层的滤波器（或权重矩阵），这些滤波器用于从输入数据中提取特征。图中的每个小方块代表一个滤波器，而滤波器的权重决定了它将如何响应输入数据的特定部分。



在第一个隐藏层中，大部分隐藏单元的激活值较低（接近 0），只有少数几个隐藏单元的激活值较高（接近 1）。这可能表明只有少数神经元对输入样本有较强的响应，而大多数神经元对输入样本不太敏感。



在第二个隐藏层中，大部分类别的激活值较低（接近 0），只有少数几个类别的激活值较高（接近 1）。这可能表明模型对输入样本的预测较为集中于几个特定的类别。



这些直方图展示了第一层和第二层权重的分布情况。权重值主要集中在-0.05 到 0.05 之间，表明权重值较小，这可能有助于防止过拟合，因为大权重可能导致梯度消失或梯度爆炸问题。权重分布的对称性和集中性表明权重初始化和训练过程可能是合理的。

4. 结论

本项目成功地从零开始构建了一个三层神经网络分类器，并在 CIFAR-10 数据集上进行了训练和测试。通过超参数优化，我们找到了提高模型性能的有效组合：学习率为 0.01，隐藏层大小为[512, 128]，L2 正则化强度为 0.01，使得测试准确略达到 49.48%。

然而，这一准确率表明三层神经网络结构可能在捕捉 CIFAR-10 数据集图像的复杂信息方面存在局限性。

为了进一步提高分类性能，未来的工作可以考虑以下几个方向：

- **增加网络复杂度：**尝试构建层数更多、结构更复杂的模型，这可能有助于提高模型对图像特征的学习能力。
- **采用卷积神经网络（CNN）：**CNN 能够自动和有效地提取图像的空间层次特征，这可能是更适合图片数据集分类任务的网络架构。
- **超参数优化：**继续调整和优化学习率、批大小、正则化强度等超参数，以找到更适合复杂模型的配置。
- **数据增强：**利用数据增强技术增加训练数据的多样性，提高模型对新图像的泛化能力。

5. 参考资料

斋藤康毅，著，陆宇杰，译. 《深度学习入门：基于 Python 的理论与实现》. 北京：人民邮电出版社，[2018].

6. 代码链接

Github:

<https://github.com/Lilac-xxq/3layernetforcifar>

百度网盘

链接: <https://pan.baidu.com/s/1scAXpqJUv1lxg4LH8ATMJg?pwd=d3y3>

提取码: d3y3