

Three-layer Neural Network Classifier on Cifar-10 Dataset

项目主页: <https://github.com/Anderasderry/2025-CV-Project1>

摘要

本项目旨在实现一个三层神经网络，对 CIFAR-10 数据集进行图像分类任务，并通过超参数搜索找到在该数据集上最优的参数组合。

一、实验环境

- **编程语言:** Python 3.13
- **主要库:** NumPy, Matplotlib, argparse, tqdm
- **硬件环境:** Intel Core i9 CPU, 16GB RAM
- **操作系统:** Windows 11

二、数据集介绍

CIFAR-10 (Canadian Institute for Advanced Research - 10 classes) 是一个广泛用于图像分类和深度学习研究的公开数据集，包含 10 个类别的彩色图像，每张图像大小为 32×32 像素，RGB三通道。该数据集分为训练集 (50,000张图像) 和测试集 (10,000张图像)。

三、模型结构

本实验实现了一个三层前馈神经网络，结构如下：

- **输入层:** 3072个神经元 ($32 \times 32 \times 3$)
- **隐藏层:** 可自定义大小 (如128、256、512)，激励函数为 ReLU (支持使用sigmoid和tanh作为激励函数)
- **输出层:** 10 个神经元，激励函数为 Softmax

反向传播

我们用 z 表示所有层的线性输出，用 a 表示所有层的激活输出。模型使用交叉熵作为损失函数，即 $Loss = \frac{1}{n} \sum_{i=1}^n (-\sum_{j=1}^C y_{i,j} \log p_{i,j})$ ，在输出层（第 n 层）采用激励函数 $\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum e^{x_j}}$ ，因此可以说明 $\frac{\partial \mathcal{L}}{\partial z^{(n)}} = a^{[n]} - y$ 。而在之前的层中，有梯度计算公式

$$\frac{\partial \mathcal{L}}{\partial W^{(i)}} = \frac{1}{m} \cdot (a^{(i)})^T \cdot \delta^{[i+1]}, \quad \frac{\partial \mathcal{L}}{\partial b^{[i]}} = \text{mean}(\delta^{[i+1]})$$

其中 m 为 batch size。最后再根据 $\delta^{(i)} = (\delta^{(i+1)} \cdot W^{(i+1)^T}) \odot f'(z^{(i)})$ ，即可通过反向传播算法计算损失的梯度。

训练流程

- 预处理数据时，使用90%的训练集进行训练，剩余的10%用作验证集。
- 训练的主循环 `fit()` 中，每一轮 epoch 开始前将数据打乱（避免过拟合），并分为若干个 batch 进行训练。每个 batch 经历 forward \rightarrow backward 后进行参数更新，采用梯度下降法 $W := W - \eta(\nabla W + \lambda W)$ ，并对权重加入 L2 正则项 $\lambda \|W\|^2$ ，防止过拟合。
- 计算损失函数：使用交叉熵损失函数（含 L2 正则化项）

$$L = -\frac{1}{N} \sum y_i \log(\hat{y}_i) + \frac{\lambda}{2} \sum \|W\|^2$$

- 支持学习率下降，在该实验中，采用每训练5轮，
`learning_rate * 0.9 \rightarrow learning_rate`的策略，减缓后期更新速度。
- 每轮epoch完成后，若当前验证准确率为历史最佳，则保存该模型权重。

四、超参数搜索

对于学习率、隐藏层大小、正则化强度，通过网格搜索尝试不同的组合，其他超参数设置为固定值，记录每次训练的准确率，最终选择在测试集上表现最好的模型。网格设置如下：

```
param_grid = {
    'lr': [0.001, 0.01, 0.1], # 学习率
    'reg_lambda': [1e-4, 1e-3, 1e-2], # 正则化强度
    'hidden_size': [128, 256, 512], # 隐藏层大小
    'epochs': 10, # 训练周期
    'batch_size': 64, # 批次大小
    'decay_every': 5, # 学习率衰减间隔
```

```

'decay_factor': 0.9, # 学习率衰减因子
'data_dir': 'cifar-10-batches-py', # 数据集目录
}

```

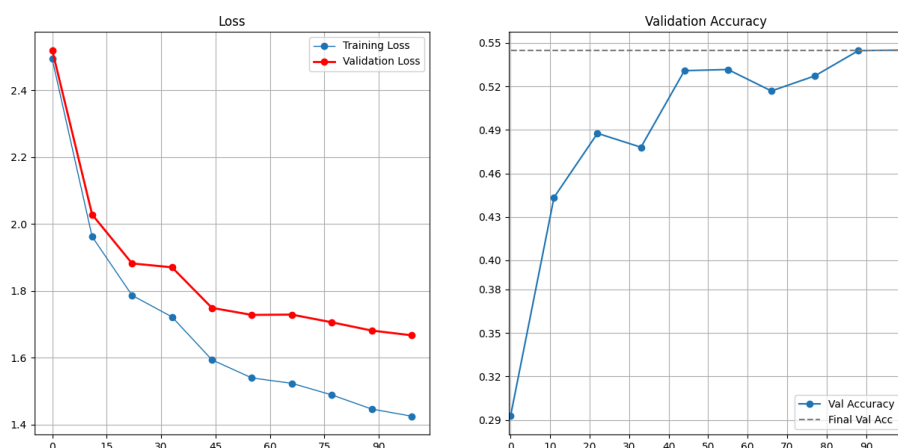
实验结果如下：

Learning Rate	Regularization Lambda	Hidden Size	Validation Accuracy	Test Accuracy
0.001	0.0001	128	0.3716	0.374
		256	0.378	0.3905
		512	0.3788	0.3874
	0.001	128	0.366	0.3696
		256	0.383	0.388
		512	0.3794	0.3798
	0.01	128	0.3636	0.3675
		256	0.3786	0.3796
		512	0.3702	0.3827
0.01	0.0001	128	0.4008	0.4082
		256	0.428	0.4335
		512	0.3674	0.3642
	0.001	128	0.4114	0.4185
		256	0.4264	0.4301
		512	0.4338	0.4396
	0.01	128	0.4142	0.4209
		256	0.4374	0.436
		512	0.4004	0.4014
0.1	0.0001	128	0.2166	0.2226
		256	0.313	0.3038
		512	0.3754	0.3746
	0.001	128	0.2316	0.2311
		256	0.2432	0.2509
		512	0.382	0.3739
	0.01	128	0.3126	0.3118
		256	0.2428	0.2417
		512	0.1174	0.1167

当学习率 = 0.01，正则化强度 = 0.001，隐藏层大小 = 512，模型在测试集上的准确率达到43.96%，为参数搜索网格中所有组合的最大值，因此将其作为模型训练的最优超参数选择。

五、实验结果

根据参数查找环节得到的最佳超参数组合下（lr = 0.01, reg_lam = 0.001, hidden_size = 512），进行 100 轮次训练（epochs = 100），得到的最优模型在测试集上达到的准确率为53.87%，训练过程中在训练集和验证集上的损失和验证集上的准确率变化如下：



可以看到，模型在训练过程中损失逐渐下降，验证准确率呈上升趋势，未出现明显过拟合。约 40 轮后，学习率衰减显著减缓了损失下降速度，有助于收敛。

训练好的模型已上传至百度网盘，可通过

https://pan.baidu.com/s/1IyLk1vmPk_Zi54C17ov3PA 下载，提取码：wwin