对 LeNet-5 模型的激活函数改进

Xiang Chang, Tianjin university, Tianjin, 300220

0. 摘要

本文是天津大学《神经网络与深度学习》（曹兵）的课程实验报告，主要工作是基于 LeNet-5 [1]+ MNIST 手写数据集的优化实验，内容包括：环境安装、数据加载、模型定义、模型训练与测试、通过对 LeNet-5 的层激活函数的改动，对比在不同激活函数下的模型结果。

项目代码见 https://github.com/CassiusXiang/LeNet-5-Optimism

1. 环境配置

本文的环境如下：

Apple MacBook (M3 Silicon)

Python 3.12.8

Anaconda 24.9.2

Pytorch 2.5.1

2. 数据加载

首先，激活 PyTorch 虚拟环境

图 2 - 1 激活 PyTorch 环境

载入 MNIST 数据集，并按照 BATCH_SIZE 进行划分

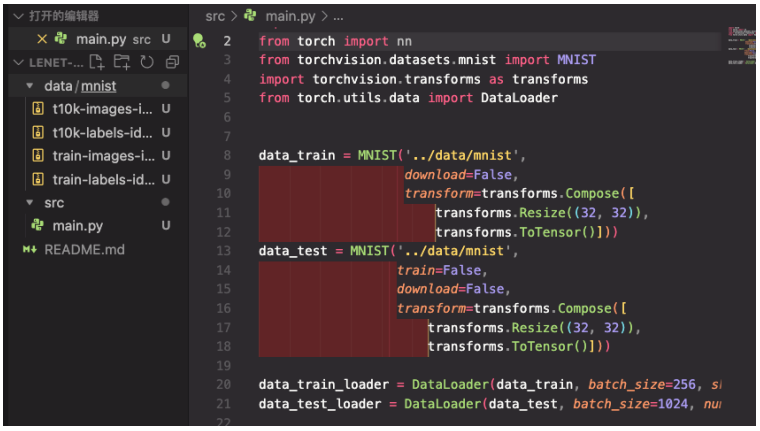

图 2 - 2 MNITST 数据集载入

简单的写了一个 vit.py 库，用来显示一些训练数据，如下图。

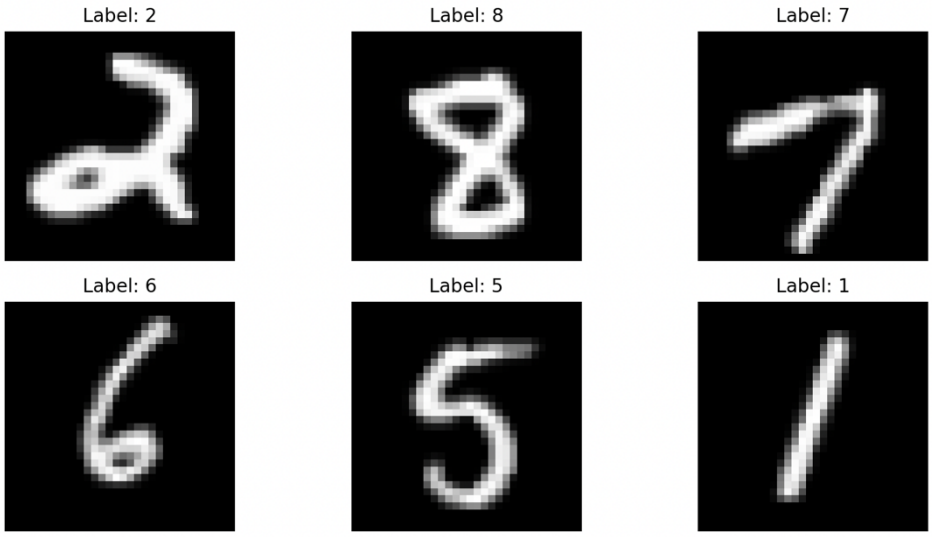

图 2 - 3 训练数据

至此，数据加载成功。

3. 模型建立与训练

LetNet 是由贝尔实验室的研究院 Yann LeCun 在 1989 年提出的，主要用于识别图像中的手写数字。

LetNet-5，是由五个层构成的：卷积层->汇聚层->卷积层->汇聚层->输出层。

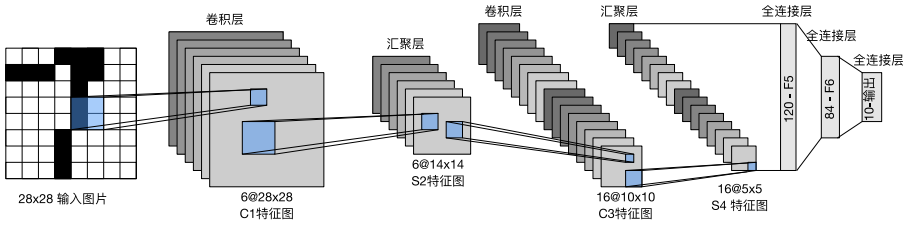

图 2 - 4 DataFlow in LeNet（https://zh.d2l.ai/chapter_convolutional-neural-networks/lenet.html）

我们在 model.py 中实现这个模型，为了便于后续对网络的增减，我们将每层进行独立实现。

优化前的网络结构如表 3-1 所示。

表 3 - 1 初始神经网络的 5 层结构

序号	名称	实现方法
0	C1	1.卷积 2. Sigmoid 3.池化
1	C2	1.卷积 2. ReLU 3.池化
2	C3	1.卷积 2. ReLU
3	F4	1.线性化 2. ReLU
4	F5	1.线性化 2.Sigmoid

模型建立成功后，进行模型训练与测试，如图 3-1

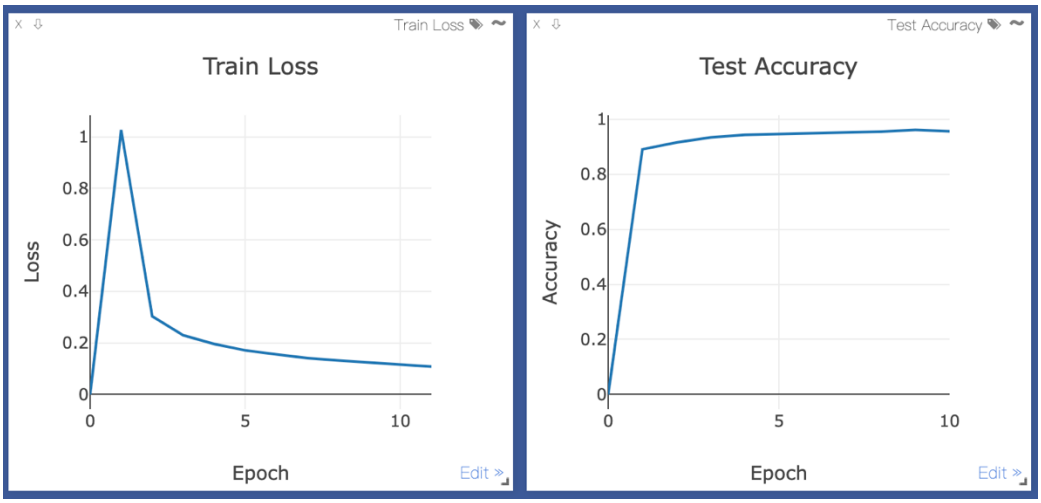

图 3 - 1 训练结果

可以看到，LetNet 的模型拟合效果还是不错的，但是由于 Sigmoid 函数在左右侧的梯度都去趋近于 0，所以我们可以考虑使用不同的激活函数进行层激活。

4. 模型改进与优化

首先我们考虑在除最后层的部分均使用 ReLU 函数。

序号	名称	实现方法
0	C1	1.卷积 2. ReLU 3.池化
1	C2	1.卷积 2. ReLU 3.池化
2	C3	1.卷积 2. ReLU
3	F4	1.线性化 2. ReLU
4	F5	1.线性化 2.Sigmoid

表 4 - 1 全 ReLU 内层激活函数

将新模型与老模型进行比较，如下图所示，其中 Model 1 是新模型。

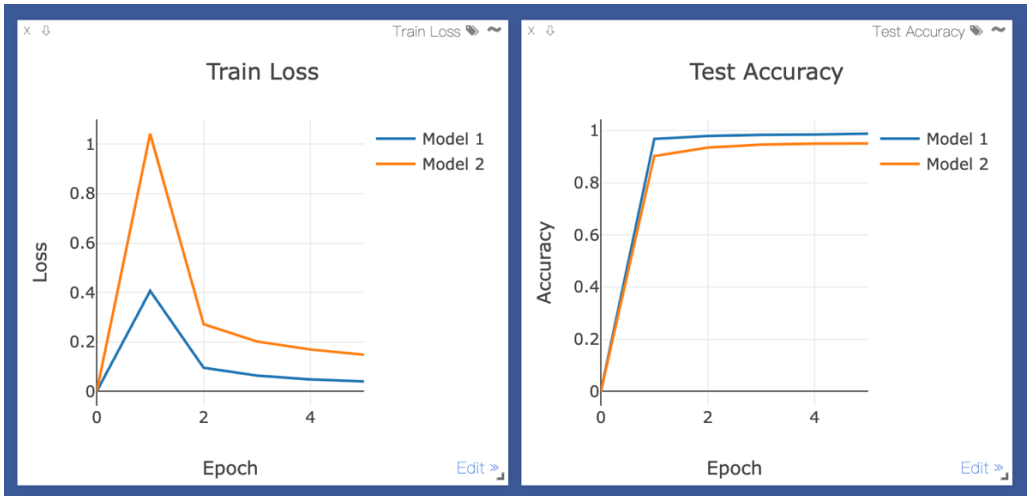

图 4 - 1 **模型比较**

可以看到，全 ReLU 的训练效果好了不少。

接下来，我们进一步使用其他激活函数，但是由于训练准确度上升过快，不方便我们观察训练效果，将每个 Epoch 的训练步数调小。

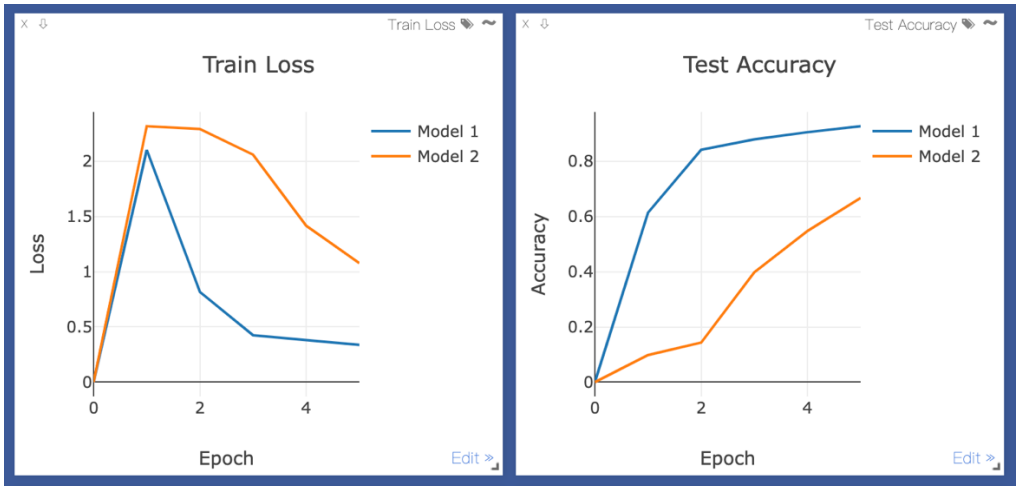

图 4 - 2 max_batch = 30

这下就能看到训练细节了，太棒了！

接下来我们考虑使用 GELU、Tanh 作为激活函数来观察收敛特性。层结构设计分别见表 4-2 和表 4-3。

序号	名称	实现方法
0	C1	1.卷积 2. GELU 3.池化
1	C2	1.卷积 2. GELU 3.池化
2	C3	1.卷积 2. GELU
3	F4	1.线性化 2. GELU
4	F5	1.线性化 2.Sigmoid

表 4 - 2 GELU 激活

序号	名称	实现方法
0	C1	1.卷积 2. Tanh 3.池化
1	C2	1.卷积 2. Tanh 3.池化
2	C3	1.卷积 2. Tanh
3	F4	1.线性化 2. Tanh
4	F5	1.线性化 2.Sigmoid

表 4 - 3 Tanh 激活

我们将 ReLU、GELU、Tanh，放到一起来观察收敛特性。如下图所示。

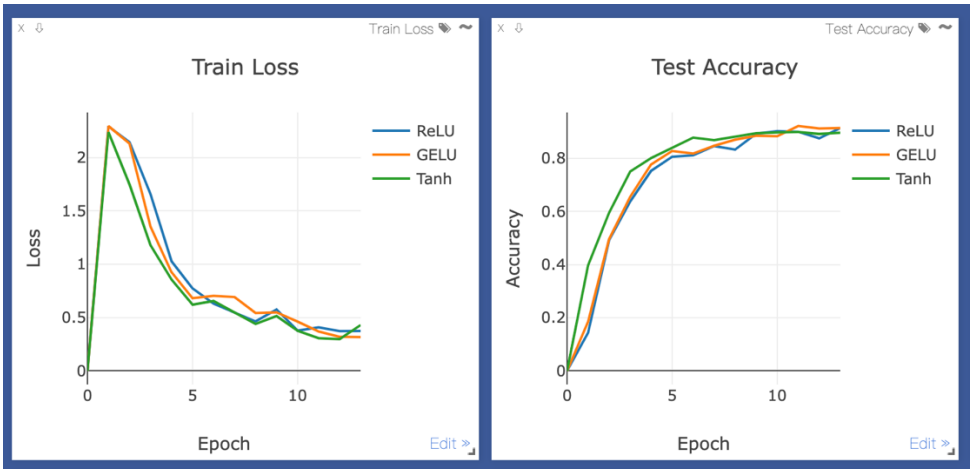

图 4 - 3 不同激活函数的训练曲线

R. 参考文献

[1] Lécun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.