

## 实验二 基于 LeNet 的手写字符图像分类实验

### 一、 实验目的

通过复现经典神经网络 LeNet<sup>[1]</sup>，进一步理解深度学习的工作机制和流程；熟悉神经网络模型的基础结构，对神经网络模型的原理与机制有进一步深入的认识和了解。

[1] LECUN Y, BOTTOU L. Gradient-Based Learning Applied to Document Recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

### 二、 实验内容

1. 使用 OpenGADL Model 软件，图形化搭建 LeNet 模型；
2. 使用 OpenGADL Model 软件，使用搭建好的 LeNet 模型在 MNIST 数据集进行训练和验证。

### 三、 实验步骤

1. 打开 OpenGADL Model 软件，新建一个项目，命名为 LeNetProj；选择任务类型为“图像分类”，并选择项目保存路径；



图 1 新建一个项目

2. 切换到【模型设计】界面，搭建 LeNet 模型
  - i. 【增加可读性，重命名页面名，非必要】新建项目默认添加一个设计页面到设计面板，双击设计页面名，弹出一个命名窗口，重命名设计

页面名为 LeNet; 也可以关闭默认设计页面, 新建一个模型设计页面命名为 LeNet;

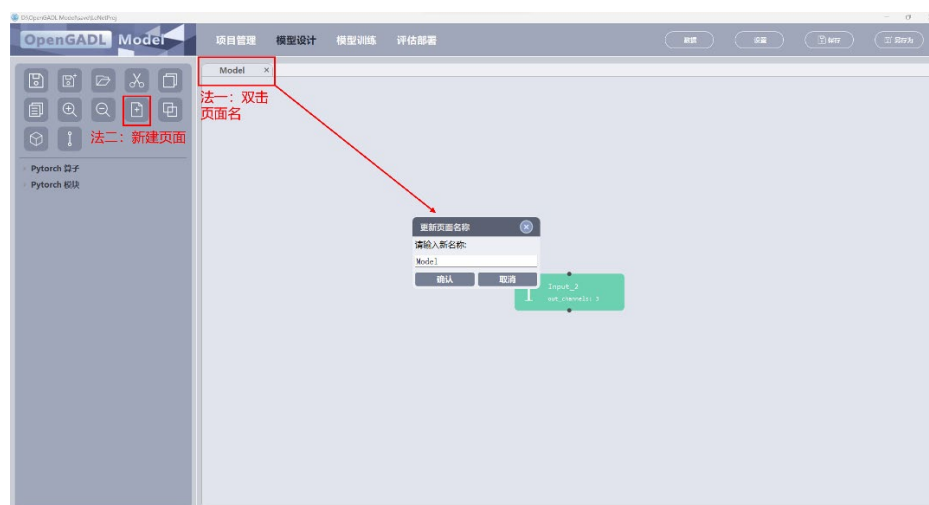


图 2 重命名页面/新建页面

- ii. 如表 1 所示, 根据 LeNet 结构, 从左侧元件树中选择相应的元件添加到设计页面上。

**注意:** 模型的输入必须是输入元件 Input (页面已默认添加), 模型的损失函数也应该在设计页面上进行添加 (这里使用交叉熵损失函数 CrossEntropyLoss);

表 1 LeNet 网络结构

Input	Name	Kernel	Stride	Padding	Out Channels
$1 \times 28 \times 28$	Conv2d	5	1	2	6
$6 \times 28 \times 28$	Pool	2	2	0	6
$6 \times 14 \times 14$	Conv2d	5	1	0	16
$16 \times 10 \times 10$	Pool	2	2	0	16
$16 \times 5 \times 5$	Conv2d	5	1	0	120
120	Linear	-	-	-	84
84	Linear (Output)	-	-	-	10
10	CrossEntropyLoss	-	-	-	-

- iii. 根据 LeNet 模型的参数对页面上的元件参数进行配置: 双击元件, 在页面右侧弹出参数配置界面;

**注意：**输入层 Input 配置输入尺寸用于模型校验，这里 LeNet 配置的输入尺寸为 (1, 28, 28)，对应 MNIST 数据集的数据尺寸；



图 3 搭建好的 LeNet 结构

3. 配置完所有参数后，点击“模型校验”按钮，校验通过后，在模型保存文件夹下生成设计模型对应的代码；

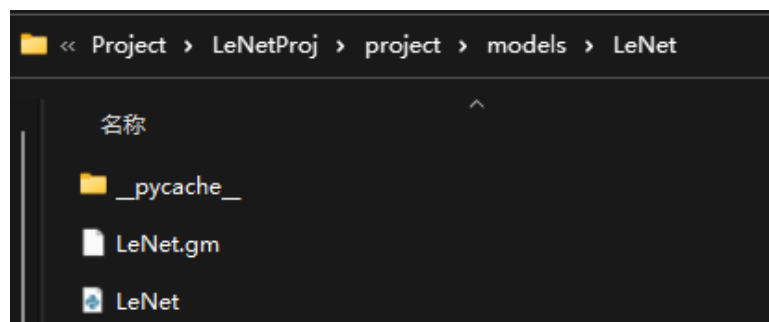
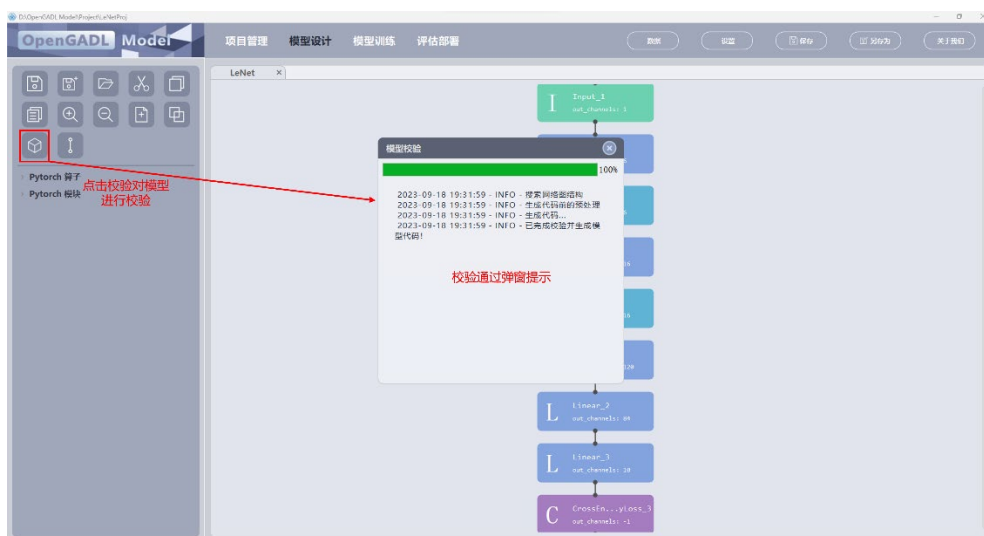


图 4 生成模型代码

#### 4. 切换到【模型训练】界面，进行训练参数的配置

- i. 选择模型路径：点击打开文件按钮，在弹出的文件对话框中选择当前项目/project/models/LeNet;
- ii. 选择数据集：本例使用公开数据集 MNIST，点击“使用公开数据集”按钮，在下拉框中选择数据集 MNIST;
- iii. 选择预训练模型：本例不使用预训练权重，因此点击按钮“不使用预训练模型”;
- iv. GPU 加速：本例使用单张 GPU 进行训练，因此点击按钮“使用单个 GPU 训练”;
- v. 输入尺寸：选择公开数据集的输入尺寸将自动匹配，无需设置;
- vi. 训练参数设置界面：配置迭代轮次数为 50，批量大小（Batch Size）为 8（取决于 GPU 显存大小）；图像均值和方差由于是公开数据集，已自动配置；优化器选用 Adam 算法（参数默认）;
- vii. 配置完成后，点击“训练”按钮，开始训练;
- viii. 训练结束后，在当前项目文件夹/project/train 得到一个单次训练的文件;

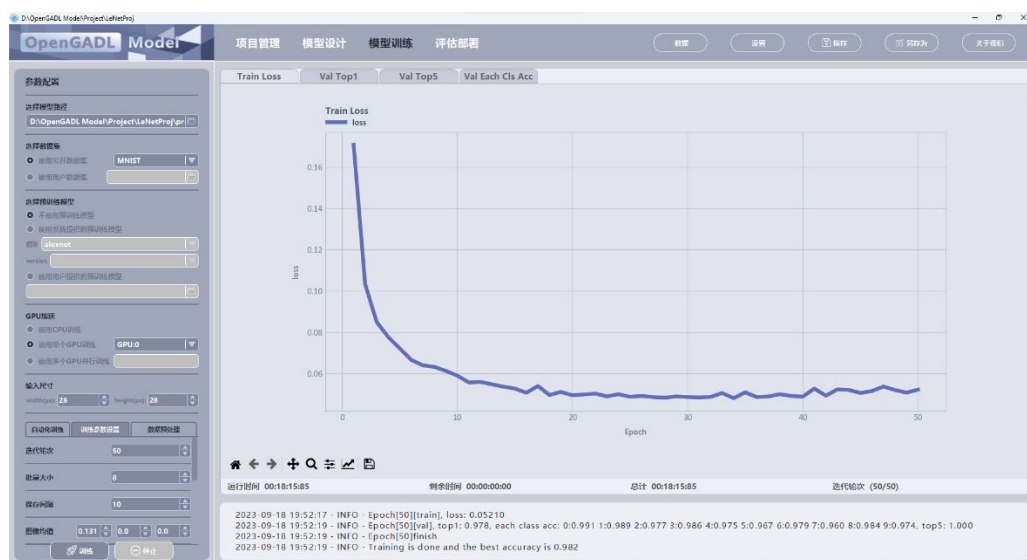


图 5 模型训练

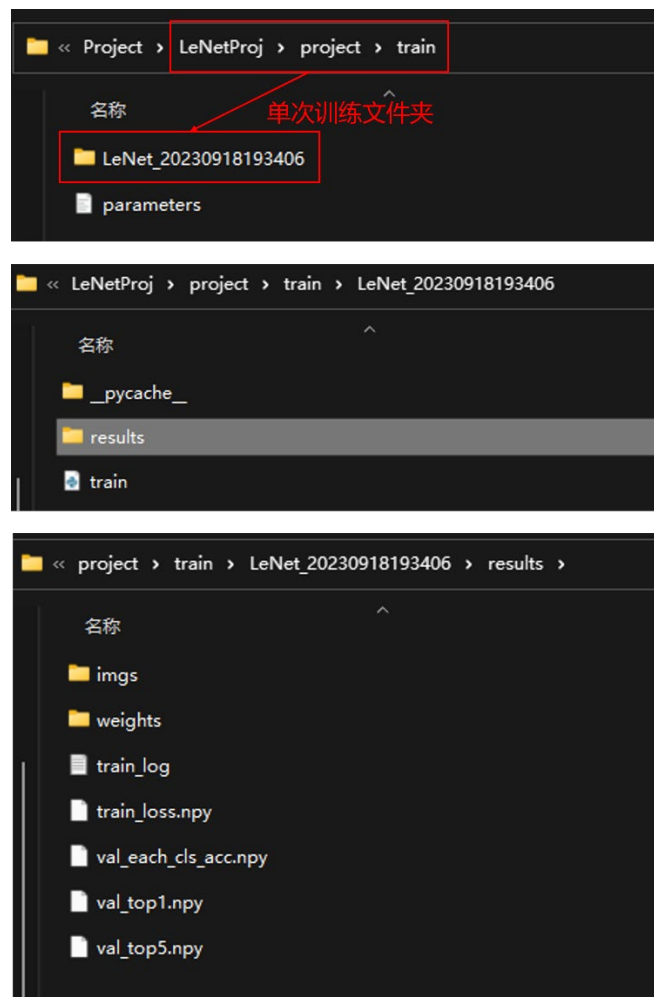


图 6 训练结果文件夹