

用 Python（也可以用 C 或者 Matlab）实现一个多层感知机（Multi-Layer Perceptron, MLP）来实现手写字符识别分类（建议用 MNIST 数据集），并设计相关的实验来探究影响多层感知机的训练因素。

主要的得分点内容包括：

- 1 构建一个 2 至 4 层的 MLP（给出核心代码，并明确是几层的 MLP），实现用于分类的损失函数，并实现训练算法，包括：（1）实现反向传播获得梯度信息；（2）使用梯度下降（或者基于批量数据的随机梯度下降）算法来更新参数。这个部分占分 50%。（注意一定不能用深度学习框架，一定要自己动手实现反向传播，梯度更新等!!!）；
- 2 设计实验，调整相关的权重初始化方法，学习率，甚至更改非线性函数（如将 Sigmoid 函数改为其他的非线性函数），给出不同的设置下模型的训练和测试损失，以及准确率，形成相关实验说明文档，这个部分占 40%。
- 3 进一步设计实验，构建一个不少于 6 层的 MLP，确保成功训练该 MLP 网络（可以随意更改网络架构，优化算法等），获得至少 95% 的测试准确率，这个部分占 10%。（注意一定不能用深度学习框架，一定要自己动手实现反向传播，梯度更新等!!!）；

作业提交相关材料：

- （1）实现文档说明和实验报告 1 个，包括核心代码及其解释（最好能够响应得分点），以及实验设置及分析（有自己实验心得及比较令人信服的分析的文档会视情况加分）；
- （2）源代码（包括训练和测试代码）文件。