

Week3实验

实验目标

本次实验需要在手写数字数据集上完成识别分类任务。

1. 掌握MLP模型结构
2. 掌握PyTorch库的自动微分操作，实现模型训练与梯度更新
3. 掌握PyTorch内置模块的使用方法，学习模型结构的定义、优化器的使用

实验内容

Task 1: 基于自动微分的手写数字识别

PyTorch库对NumPy做了进一步的封装，能结合链式求导法则，实现自动微分。

代码见Week3_Task1_Question.ipynb，请根据TODO内容依次补全：

1. 模型的正向过程(forward函数)
2. 完成正向过程后，调用反向传播，获取参数梯度，并实现参数的SGD更新(train函数)
3. 利用已经实现的正向过程，实现模型的预测(predict函数)
4. 实现模型的预测准确率统计(test函数)

补全所有代码后，完成模型的训练、测试、预测样本可视化。

Task 1 Bonus: 手动推导参数梯度

Task 1中直接使用了Pytorch框架进行自动微分，而链式法则可以帮助我们计算出各参数的偏导数。

尝试手动推导 W_h, b_h, W_o, b_o 的梯度，通过矩阵运算来实现梯度计算 (在Week3_Task1_Question.ipynb中自行补充train_handcraft函数)。

Task 2: 基于PyTorch框架的手写数字识别

为了便于模型的定义与训练，PyTorch还提供神经网络模块接口(torch.nn.Module类)，通过重写类函数，我们可以自定义模型的结构、参数初始化方式与正向计算过程。

代码见 Week3_Task2_Question.ipynb, 请根据TODO内容依次补全:

1. 定义要求的MLP模型结构(__init__函数)
2. 实现MLP模型的前向过程(forward函数)
3. 基于优化器的使用方法, 完成反向梯度传播、参数更新(《训练模型》部分)

补全所有代码后, 完成模型的训练、测试、保存, 与Task1的结果比较。

检查内容

Task 1: 基于自动微分的手写数字识别

1. Week3_Task1_Question.ipynb中的代码实现正确
2. 再次运行代码, 代码能正常执行
3. 默认参数下的测试集准确度(90%以上)
4. 自行尝试的3组不同的(超参数-准确度)数据

Task 1 Bonus: 手动推导参数梯度

1. 手动推导 W_h, b_h, W_o, b_o 的梯度
2. 通过矩阵运算来实现梯度计算(自行补充train_handcraft函数)
3. 再次运行代码, 代码能正常执行
4. 默认参数下的测试集准确度

Task 2: 基于PyTorch框架的手写数字识别

1. Week3_Task2_Question.ipynb中的代码实现正确
2. 再次运行代码, 代码能正常执行
3. 默认参数下的测试集准确度, 与Task1的结果比较
4. (可选, 额外尝试)torch.optim中其他优化器的结果对比