

Week2实验

实验目标

- 本次实验主要是通过完成一个简单模型的训练任务：
 1. 掌握机器学习模型的基本概念定义。
 2. 熟悉如何用Python实现机器学习模型的训练测试过程。

实验步骤

红酒分类任务

- Wine数据集是一个葡萄酒分类数据集，数据集中包含从3种不同种类的葡萄酒种采样的178杯酒的数据信息。每一杯酒的数据信息包含13个描述葡萄酒客观条件的数据和1个标明葡萄酒具体种类的标签。
- 现已将数据集划分成训练集（150个数据点）和测试集（28个数据点），请按步骤完成：
 1. 将划分好的数据文件和代码上传到服务器端。
 2. 填空完成代码中的TODO内容。
 3. 执行代码完成模型训练。

检查内容

- 对于分类任务：
 1. 代码填空正确，模型正常训练
 2. 测试集上的准确度超过90%
- 超参数验证：
 1. 尝试不同训练轮数和学习速率
 2. 记录训练集和测试集上的准确度

思考题

1. ppt中给了交叉熵最终的求导公式，你能计算出它的完整推导过程吗？
2. 如果不使用CrossEntropy loss，而是直接使用预测结果和数据标签的平方差作为loss，梯度计算会发生什么变化？模型预测准确率会发生什么变化？

附录

- 梯度下降法简易入门教程: <https://zhuanlan.zhihu.com/p/37824148>
- 实验中可能用到的几个关键Numpy函数如下所述:
 1. `np.zeros(shape)`, 生成一个全零多维矩阵
 - `shape`表示多维矩阵每一维度的大小, 比如`shape=(2,3)`等价于创建一个2*3的全零矩阵
 2. `np.exp(x)`, 对输入Numpy多维矩阵`x`中的每一个元素, 计算它的指数函数计算结果
 - 假设`x= [[1, 2], [3, 4]]`, 那么 `np.exp(x)= [[e, e2], [e3, e4]]`
 - `np.log`函数同理
 3. `np.sum(x)`, 计算输入Numpy多维矩阵`x`中所有元素的数值和
 - 假设`x= [[1, 2], [3, 4]]`, 那么 `np.sum(x)= 10`
 4. `np.matmul(a, b)`, 计算两个输入Numpy二维矩阵的矩阵乘积
 - 矩阵大小要求: 矩阵`a`为 $N \times K$, 矩阵`b`为 $K \times M$, 输出大小为 $N \times M$ 的新矩阵
 - 也可支持: 矩阵`a`为 $N \times K$, 而`b`为 K 维向量, 输出大小为 N 维的向量
 5. `np.outer(a, b)`, 计算两个输入Numpy向量的外积
 - 向量`a`是 N 维, 向量`b`是 K 维, 则输出大小为 $N \times K$ 的矩阵
 - 计算规则是: `out[i, j] = a[i] * b[j]`