机器学习与数据挖掘 作业一

吴天 19334019

1 Abstract

本次实验实现了线性分类器、非线性分类器、含 L1 或 L2 正则化的非线性分类器，并使用 UCI 数据库中的 Dry Bean Dataset 数据集验证了上述分类器在该数据集上和性能，经过调参，大部分的模型都达到了 90% 以上的准确度。

2 Assignment

1. 实现 Lecture 2 线性分类器（多类分类采用 softmax 函数）.

2. 通过基函数非线性化步骤 1 的线性分类器，得到不含正则化的非线性分类器（基函数的选择不限）.

3. 通过 L1 和 L2 范数分别对步骤 2 的非线性分类器进行正则化，分别得到含 L1 和 L2 正则化的非线性分类器。

4. 在 UCI Machine Learning Repository 找到自己认为合适的数据集对比：线性分类器、不含正则化的非线性分类器、含 L1 正则化的非线性分类器、含 L2 正则化的非线性分类器.

5. 对比指标采用分类精度，即报告每一个分类器在测试集上得到的 ACC.

3 Method

3.1 Dataset

本次实验采用的数据集是 Dry Bean Dataset，共有 7 个类别、16 个特征，13611 个实例。实验数据可以在https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset获取。

随机将数据集中的 70% 数据作为训练集 30% 的数据作为测试集，且训练集和测试集的不同类别的数据分布是与原数据集的分布保持一致。

3.2 Data Standardization

采用了 z-score 标准化 (zero-mean normalization)，给予原始数据的均值（mean）记为 μ 和标准差（standard deviation）记为 σ 进行数据的标准化。经过处理的数据符合

标准正态分布，即均值为 0，标准差为 1，其转化函数为：

$$X = \frac{X - \mu}{\sigma}$$

3.3 Linear Classfier

线性分类器是使用线性的函数表达式即一个超平面，从而对样本进行分类。对于输入的特征向量为 D 维，即 $x \in R^D$，并且分类的类别数为 K，则对应的线性分类器具有如下形式：

$$f(x) = W^T \boldsymbol{x} + \boldsymbol{b}$$

其中 $\boldsymbol{b} \in R^K$ 为偏置向量, $W \in R^{D \times K}$ 。得到的结果为 $\boldsymbol{z} = f(\boldsymbol{x}) \in R^K$ ，其中 $\boldsymbol{z}_i$ 当前样本 $\boldsymbol{x}$ 对应第 i 类的权重。对于 K 分类问题而言，取 $argmax_i \boldsymbol{z}_i$ 作为 $\boldsymbol{x}$ 的预测分类。

简写上述形式，对 W 矩阵进行增广，并对 $\boldsymbol{x}$ 添加第 $D+1$ 维并设置为 1 ，从而替代 $\boldsymbol{b}$ 的作用，此时我们设增广后的 $\boldsymbol{x} = [x_0, x_1, \cdots, x_D, x_{D+1}]^T$，增广后的权重矩阵 $W \in R^{(D+1) \times K}$，则原形式记为：

$$f(\boldsymbol{x}) = W^T \boldsymbol{x}$$

3.4 Softmax Function

线性分类器 $\boldsymbol{z} = f(\boldsymbol{x}) = W^T \boldsymbol{x}$ 的输出是 K 维的向量，但其每一维的值域均为 $[-\infty, +\infty]$ ，我们可以使用 softmax 函数对线性分类器的输出向量进行归一化，将各维的值转换为各类所对应概率（即满足 $\boldsymbol{z}_i \in [0,1], 1 \leq i \leq K$ 且 $\sum_{i=1}^{K} \boldsymbol{z}_i = 1$ ）。即：

$$softmax_i(\boldsymbol{z}) = \frac{e^{\boldsymbol{z}_k}}{\sum_{k=1}^{K} e^{\boldsymbol{z}_k}}$$

z 经过 softmax 层得到的输出 $\hat{\boldsymbol{y}} = softmax(\boldsymbol{z})$ 中各维 $\hat{\boldsymbol{y}}_i$ 代表分类器所预测该样本为第 i 类的概率。

3.5 Loss Function

交叉熵损失函数是分类问题中常用的损失函数，其用来衡量预测类别结果与实际类别结果的相似度差距损失。

为了便于计算，这里引入 onehot 向量，我们将每个样本对应的真实标签转换为 onehot 向量 $\boldsymbol{y}$ ，其中 $\boldsymbol{y}$ 中对应样本真实分类标签的那一维为 1 ，其余维均为 0 。

假设当前训练集中共有 N 个样本，则当前线性分类器模型的分类结果与真实分类

标签的交叉熵损失可以记为:

$$CE = -\frac{1}{2N}\sum_{n=1}^{N}(\boldsymbol{y}_k^{(n)})^T log\hat{\boldsymbol{y}}_k^{(n)}$$

3.6 Model

根据上述理论知识，我们可以得到线性分类器的结构与前向传播过程。假设训练集样本数据共有 N 个样本，其中每个样本的特征为 D 维，则对应的训练集样本组成的矩阵增广后得到 $X \in R^{N\times(D+1)}$，线性分类器的线性增广权重矩阵为 $W \in R^{(D+1)\times K}$ 首先我们将样本通过线性分类器的线性部分，得到输出:

$$Z = XW$$

之后将得到的输出矩阵 $Z \in R^{N\times K}$ 通过 softmax 函数，得到输出:

$$\hat{\boldsymbol{y}} = softmax(Z)$$

最后对于第 n 个样本，我们的预测结果即为 $argmax_i\hat{\boldsymbol{y}}_i^{(n)}$ 类。

3.7 Gradient descent

梯度下降法 (gradient descent) 是一个最优化算法，常用于机器学习和人工智能当中用来递归性地逼近最小偏差模型。交叉熵的损失函数的梯度为:

$$\frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{1}{N}\sum_{i=1}^{N}[\sigma(\boldsymbol{x}^{(i)}\boldsymbol{W}) - y^{(i)}]\boldsymbol{x}^{(i)T}$$

3.8 Parameter update

在训练过程中，当得到当前梯度后，假设学习率为 η ，我们可以根据以下式子更新梯度:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta\frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}}$$

这里我们采用 Adagrad 算法动态调整学习率，即:

$$\eta = \frac{\eta}{\sqrt{\sum grad_i^2}}$$

其中 $\sum grad_i^2$ 为对应参数的到目前为止的所有梯度的平方和。根据公式可以看出，随着训练地不断进行，学习率会以合适的速度逐渐减小，这样可以达到在训练初期有较大学

习率以加快收敛速度、在训练后期有较小的学习率以使得损失能够逐渐收敛稳定。

3.9 Regularize

我们在构造机器学习模型时，最终目的是让模型在面对新数据的时候，可以有很好的表现。当用比较复杂的模型，去拟合数据时，很容易出现过拟合现象 (训练集表现很好，测试集表现较差)，这会导致模型的泛化能力下降，这时候，我们就需要使用正则化，降低模型的复杂度。

本次实验采用了 L1 范数和 L2 范数来进行正则化。另，由于使用 Adagrad 算法动态调整学习率会导致一开始对权重矩阵惩罚过大，因此添加正则化项的模型没有使用动态学习率。

3.9.1 L1 regularization

λ 是一个超参数，用于控制正则化项的影响, $\|W\|_1 = \sum_{k=1}^{K} |w_k|$ 是 $L1$ 范数, 原损失函数加上 $L1$ 正则化表示为:

$$\tilde{L}(\boldsymbol{W}) = L(\boldsymbol{W}) + \lambda \|\boldsymbol{W}\|_1$$

加入正则化项后，参数更新的公式变为:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}} - \eta \lambda \boldsymbol{sign}(\boldsymbol{W})$$

其中 $\boldsymbol{sign}$ 是一个符号函数, 取数字符号（数字前的正负号）

3.9.2 L2 regularization

λ 是一个超参数，用于控制正则化项的影响, $\|W\|_2 = (\sum_{k=1}^{K} w_k^2)^{\frac{1}{2}}$ 是 $L2$ 范数, 原损失函数加上 $L2$ 正则化表示为:

$$\tilde{L}(\boldsymbol{W}) = L(\boldsymbol{W}) + \frac{\lambda}{2} \|\boldsymbol{W}\|_2^2$$

加入正则化项后，参数更新的公式变为:

$$\boldsymbol{W}_{t+1} = (1 - \eta\lambda)\boldsymbol{W}_t - \eta \frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}}$$

4 Result

4.1 Linear Classfier

单独使用线性分类器，虽然模型最后能够得到一个比较好的准确度，但所需要的迭代次数非常多，且在迭代的后期收敛非常缓慢，并有一定程度的振荡，见图 1。

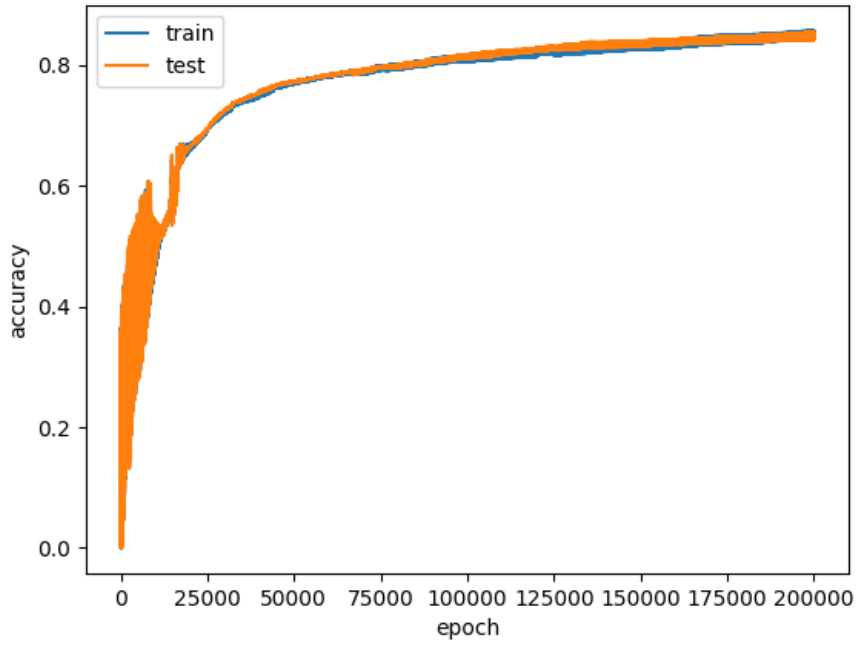


图 1: 学习率为 0.3 时，模型准确度随 epoch 次数的变化

由于迭代后期振荡现象比较明显，虽然提高学习率能够使模型有更好的拟合趋势，但也有可能在迭代结束的时候，模型的准确度并不理想，调整学习率对模型的提高已经意义不大，见图 2。

4.2 Data Standardization

损失函数 $L(\boldsymbol{W})$ 在不同维度下会以不同的速率降低，注意到，数据集中原始数据中的各个特征具有非常不同的数量集，这导致梯度下降的收敛缓慢，因此要将原始数据标准化处理。经过标准化处理的数据能够使模型更快地收敛，上述数据未经过标准化处理的模型 epoch 为 200000 仍未收敛，而经过标准化处理的数据 epoch 不到 500 就已经收敛，见图 3。

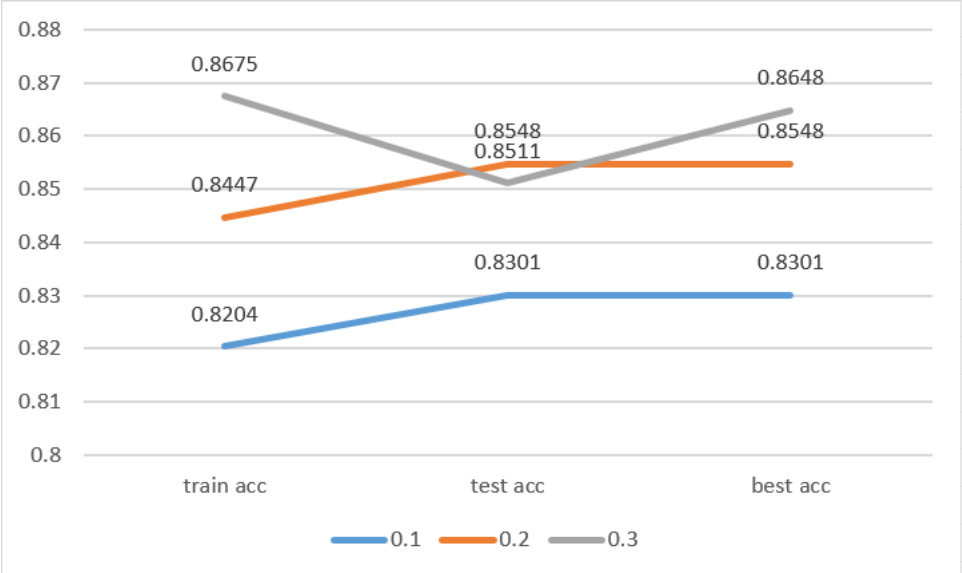


图 2: 学习率分别为 0.1、0.2、0.3 时，epoch 为 200000，模型训练完毕时训练集准确度、测试集准确度以及训练过程中最好的准确度

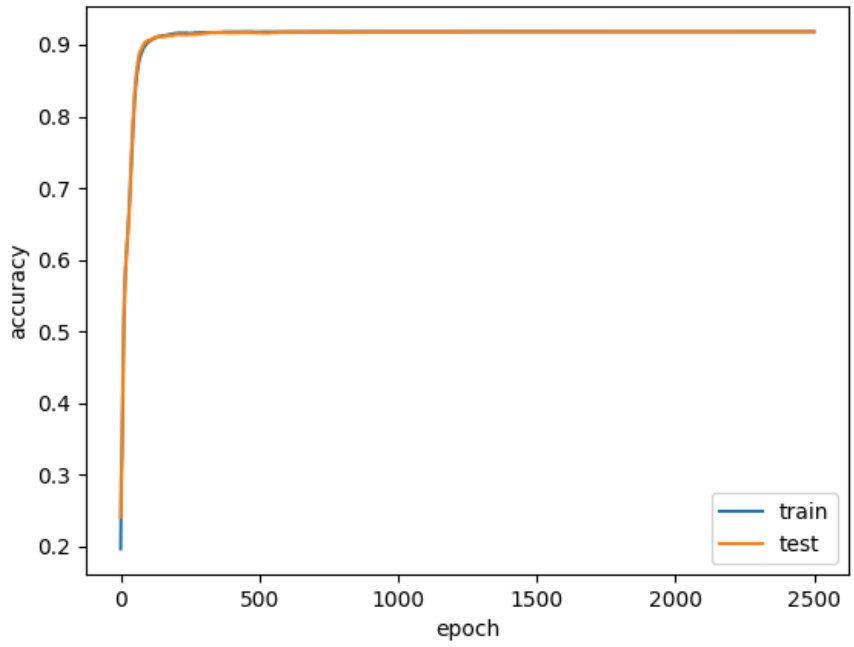
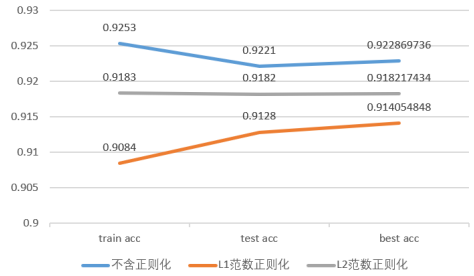


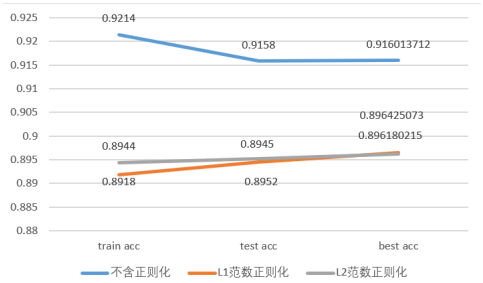
图 3: 学习率为 0.3 时，模型准确度随 epoch 次数的变化

4.3 Nonlinear Classfier and Regularize

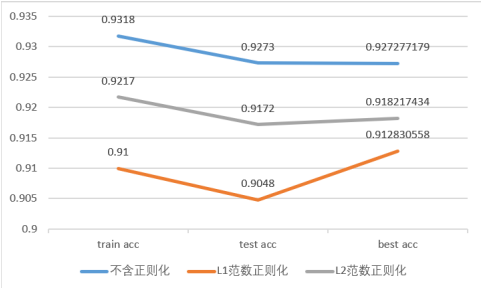
本次实验在 Dry Bean Dataset 上训练了线性分类器、基函数为 $X \to sin(X)$ 的分类器，基函数为 $X \to X, X^2$ 的分类器、基函数为 $X \to X, X^2, X^3$ 的分类器，并且每种模型还训练了带 L1、L2 正则化项的分类器，学习率均为 0.5，前三个分类器的正则系数 $\lambda = 0.01$, 第四个分类器的正则系数 $\lambda = 0.001$ 迭代次数均为 2500 次，得到的效果如图 4 所示：



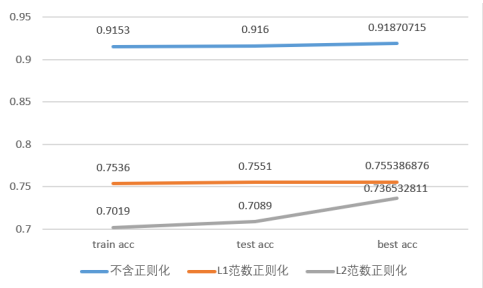
(a) 线性分类器和带正则化项的线性分类器的模型表现



(b) 基函数为 $X \to sin(X)$ 和带正则化项的分类器的模型表现



(c) 基函数为 $X \to X, X^2$ 和带正则化项的分类器的模型表现



(d) 基函数为 $X \to X, X^2, X^3$ 和带正则化项的分类器的模型表现

图 4: 线性分类器和不同基函数的分类器以及不带正则化项和带正则化项的模型表现

在 Dry Bean Dataset 上，线性分类器已经获得了非常好的效果 (图 4(a)), 添加各类不同的基函数，有的提升了模型的准确度 (图 4(c))，有的降低了模型的准确度 (图 4(b、d))。此外，所有模型的 Acc 在训练集和测试集上得到了较为接近的表现（训练集上正确率略高于测试集）(图 3)，所以并未发生过拟合的问题，加了正则化项反而使模型的准确度下降 (图 4(abcd))。

5 Summary

在本次实验中，我重新对线性分类器中涉及到的各项公式进行了一遍推导，并且动手将算法实现。通过这次实验，我对老师课堂上讲到的有关线性分类器、优化和训练技巧等理论知识的掌握更加深入了，并且提高了我把理论知识实现为可行代码的能力，感

觉收获很大。

除此之外，整个实验的过程走一遍下来，寻找数据集、数据处理、实现算法、选择不同基函数、查看结果、调参、换基函数换参数、不断实验，最后终于让模型在这个数据集上表现不错。对于刚接触机器学习来说，整个流程走下来实属不易，总算是摸到了机器学习的门槛。

6 Supplementary material

本实验的代码已上传至 GitHub，可以在https://github.com/PointerA/ML_DM_course获得。