分类器实现汇总作业

## 问题

## Based on the MNIST dataset, design and implement classifiers including: least squares with regularization, Fisher discriminant analysis (with kernels), Perceptron (with kernels), logistic regression, SVM (with kernels), MLP-NN with two different error functions.

## 数据集

MNIST数据集由60000张训练图片和10000张测试图片组成，为手写0-9的灰度图像，分辨率28\*28。由于需要控制运行时间，因此将训练和测试集都缩小到1000。

## one-vs-one和one-vs-all

由于很多算法只能进行二类划分，在进行多类划分时有两种策略。one-vs-one训练n\*(n-1)/2个分类器，对于一个元素，在每个分类器中都能得到一个类别，最后将投票次数最多的类别作为其最终类别。one-vs-all训练n个分类器，用于分类某个元素和其他所有元素。如果一个元素在某个分类器被判定为真那么就认为这个元素属于该类。在one-vs-one中，由于元素会在很多和自己类别无关的分类器中判定，在one-vs-all中则是可能在多个分类器中被认为属于一类；或是被所有分类器拒绝，因此两种方法都各有优劣。

## least squares with regularization

最小二乘法使用正则化，采用了L2正则化，即岭回归。其目标函数为

求导结果为

令导数为零，则

使用sklearn中的岭回归函数。同时使用最小二乘法找到平面仅能做二类划分，测试后使用one-vs-one方法来做多类划分。最终分类准确率77.8%

## Kernel LDA

Kernel LDA是在普通的LDA上运用核方法，使得LDA能够处理非线性问题。使用核方法后，对于每次询问都需要和训练数据集中每个元素进行核函数计算，因此相比传统LDA测试开销很大。

为1类的平均值，为2类的平均值，为1类的协方差矩阵，为2类的协方差矩阵，LDA的目标函数是

其解为

使用核方法，核函数为，升维结果为，目标函数变为

其中和为经过核函数以后的协方差矩阵。在这里，我们使用来表示，即

那么我们要求。考虑原式的解，和每项均为，因此直接计算即可。对于和，根据上式拆分可得

因此可求解出。然后当输入一个值时，我们可以计算

核函数采用了RBF核。测试后使用one-vs-one的方法做多类划分。最终分类准确率29.1%

## Kernel Perceptron

核函数感知器使用感知器的对偶问题，这样使用核方法不会带来不同的时间复杂度。

感知器对偶问题中，

初始时均为零。每次训练取出一个样本，如果该样本分错那么。当输入一个值时，我们计算

因此在训练时如果我们计算出矩阵就可以减少计算量。

使用核方法，核函数为，升维结果为，函数变为

变为

矩阵变为

同样使用RBF核，由于训练速度过慢，因此采用训练分类器较少的one-vs-all方法。最终分类准确率72.6%

## Logistic Regression

Logistic Regression和最小二乘法相比由于使用了Logistic函数使得距离分类平面很远，也就是分得很对的点对分类平面的影响减小，因此可以取得比最小二乘法更好的效果。

其最大似然函数为

取对数为

然后使用梯度下降求解。

使用sklearn中的LogisticRegression，最终分类准确率81.9%

## SVM

设平面的法向量是，则一个点，类别到平面的距离是

如果间隔越远那么越大，同时所有点都要分对类别，所以目标函数是

加入拉格朗日算子，目标函数为

目标是

同时

则有。

使用支持向量机来进行分类，使用sklearn中的SVM，并采用默认参数。由于直接使用原始数据无法运作，将输入的灰度图像根据阈值128转换为01黑白图像作为输入。最终分类准确率78.3%

## MLP

使用多层感知器，即神经网络进行分类。将图片拉成一条向量作为输入，隐层有80个神经元，输出10个神经元。三层之间采用全连接，并使用ReLU作为中间层的激活函数。输出10维表示10个类别。训练的batch size取100，并进行10000个epoch的训练。

首先采用交叉熵函数作为损失函数，最终分类准确率83.3%

然后采用KLDivLoss，需要在输出结果上加上一层softmax。最终分类准确率84.1%

## CNN

使用卷积神经网络，LeNet5结构。由于LeNet5需要输入32\*32，因此在手写数字图片边缘添加2像素的边距。经过5\*5的卷积后变成28\*28\*6，下采样后14\*14\*6，继续5\*5卷积后10\*10\*16，分别与120，84，10个神经元的隐层全连接得到结果。全连接隐层使用ReLU作为激活函数。输出10维表示10个类别。训练的batch size取100，并进行1000个epoch的训练。采用交叉熵函数作为损失函数，最终分类准确率91.0%

## 代码说明

代码使用Python3编写，Kernel LDA代码使用了<https://github.com/bhaktipriya/Kernel-PCA-and-LDA/blob/master/klda.py>的代码。Ridge, Logistic Regression, SVM使用sklearn; MLP使用pytorch; Kernel Perceptron，one-vs-one，one-vs-all，数据处理，RBF自行编写。

代码共有main.py和klda.py两个文件，执行python3 ./main.py即可。

## 文件说明

共有如下文件/文件夹：

mnist：mnist数据集文件

main.py/klda.py：代码文件。

output.txt：main.py执行的输出。

report.docx/report.pdf：报告文档及其pdf。