作业任务

本次大作业要求大家使用机器学习方法，对手写数字图片进行分类。针对本次大作业的问题，附录中提供了参考材料，大家可以采用但不限于里面的方法。

本次作业分成基础任务和提高任务：

1、基础任务：

使用机器学习算法，对黑白手写数字图像进行分类。利用训练集中的数据和标签（train\_data1,和train\_labels1），通过机器学习方法得到分类器，并在同为黑白的手写数字图像（valid\_data1）下验证，最后助教会在额外留出的测试集下测试；

2、提高任务：

使用机器学习算法，对手写数字图像进行分类。利用训练集中黑白图像的数据和标签（train\_data1和train\_labels1）和彩色手写图像的数据（train\_data2），通过机器学习方法得到分类器，并在彩色手写数字图像验证集（valid\_data2）下验证，最后助教会在额外留出的测试集下测试；

|  |  |
| --- | --- |
| 黑白图像示例 | 彩色图像示例 |
|  |  |

数据

我们的数据储存在文件train\_data.pkl中，读取数据的示例代码见read\_train\_data.py，数据文件中包括：

train\_data1, train\_labels1:训练集，50000张黑白手写数字的图片和对应标签（从0到9），其中train\_data形状大小为50000\*28\*28，train\_labels长度为50000；

train\_data2:训练集，50000张彩色手写数字的图片，其中train\_data2形状大小为50000\*28\*28\*3；

valid\_data1, valid\_labels1:验证集，10000张黑白手写数字的图片和对应标签（从0到9），其中valid\_data1形状大小为50000\*28\*28，valid\_labels1长度为50000；

valid\_data2, valid\_labels2:验证集，5139张彩色手写数字的图片和对应标签，其中valid\_data1形状大小为5139\*28\*28\*3，valid\_labels1长度为5139；

要求

1. 程序应保证能在以下环境之一运行：Ubuntu14.04、Win 8、Win 10、OS X EI Caption。代码如需编译，报告中应清楚描述源代码需要的编译环境，并能在必要的时候在助教面前复现数据处理和模型训练的过程。

2. 数据集中的验证集用来在完成作业的过程中对比不同算法的性能和选择最优的模型参数，最终得分同时取决于额外留出的测试数据。代码中必须留有测试接口，方便助教使用额外的数据进行测试，测试数据格式如下：

测试文件：.pickle文件

要求你的程序读取.pickle文件，然后将所有预测结果输出到predict.txt，每个样本的预测结果单独一行，预测顺序保证与.pickle文件一致。测试接口的使用应在报告中详细说明。如果同时完成了基础任务和提高任务，应对两个任务都分别提供相应的测试接口。

3. 允许使用网上相关的代码包，但在报告中必须清楚说明程序中哪些部分是参考或是参考了别人的代码，哪些部分是由自己完成。

4. 鼓励实现多种算法，做充分的对比、分析和尝试；如果有，请简要说明相应设计和原理，并展示相应结果。

给分

大作业由个人单独完成，评分标准如下（满分10分）：

报告2分，基础任务6分（算法实现和现象分析4分，接口满足运行要求并能在测试集上取得较好分类效果2分），提高任务2分。

如果各部分中有完成得非常突出的，酌情加1到2分。

附录：参考方法及资料（针对基础任务）

可以把每张图像（28\*28）直接展开成784维的向量，直接用传统的算法进行分类。

以下介绍的方法均为经典的分类算法，网上有大量的阅读材料，下面列举的只是一部分参考

1. SVM（支持向量机）

介绍

https://en.wikipedia.org/wiki/Support\_vector\_machine

现有工具包（支持各种编程语言）

http://www.csie.ntu.edu.tw/~cjlin/libsvm/

matlab里面也有内置的工具包，在matlab中输入help svmtrain即可查看相关教程。

2. 神经网络multi-layer perceptron

介绍

https://en.wikipedia.org/wiki/Multilayer\_perceptron

http://www.iro.umontreal.ca/~pift6266/H10/notes/mlp.html

基于theano工具包的代码示例

http://deeplearning.net/tutorial/mlp.html

3. softmax

介绍

http://ufldl.stanford.edu/wiki/index.php/Softmax\_Regression

https://en.wikipedia.org/wiki/Softmax\_function

4. kNN（k最近邻）

介绍

https://en.wikipedia.org/wiki/K-nearest\_neighbors\_algorithm

另外对于kNN算法，可以通过Metric Learning来对kNN中的距离函数进行学习从而提升kNN的性能，有兴趣的同学可以参考。

也可以在图像上采用深度学习的框架进行学习，以下是一些工具包

Caffe: http://caffe.berkeleyvision.org/

Keras:http://keras.io/

Lasagna:http://lasagne.readthedocs.org/en/latest/

MatConvNet：http://www.vlfeat.org/matconvnet/