数学建模大项目：手写省份与电话号码识别

刘雨萌 王琪 徐逸菲

2017年6月18日

目录

1 摘要 **2**

2 手写省份识别 **3**

2.1 模型的构建 3

2.1.1 模型可视化 3

2.1.2 主要部分说明 4

2.2 数据预处理 4

2.2.1 单字拼接 4

2.2.2 去噪 4

2.2.3 倾斜文字 5

2.3 模型测试结果 6

3 手写数字识别 **6**

3.1 单个手写数字的识别 6

3.2 电话号码的识别问题 7

3.2.1 电话号码数据集的获取 7

3.2.2 数字分割问题 8

3.2.3 手写电话号码分割结果 9

4 收获与总结 **9**

5 不足与展望 **10**

1 摘要

随着电子商务，物流行业蓬勃发展，人工分拣邮件与快递的低效性也日趋显现。本文主要研究的是快递单号上的手写地址中的省份识别以及手写电话号码的识别。

针对省份识别，我们基于深度学习框架 keras ，构造了可识别手写省份的神经网络模型。又采集并构造了60000多的手写省份的图片，包括了全国32个省、市、自治区(不含香港和澳门)，从中选取了900个数据作为测试集，其余作为训练集。在经过1000次迭代后，我们的模型在训练集上的准确率为：88.17%, loss为：0.3799;在测试集上的准确率为：88.75%, loss为：0.5489.

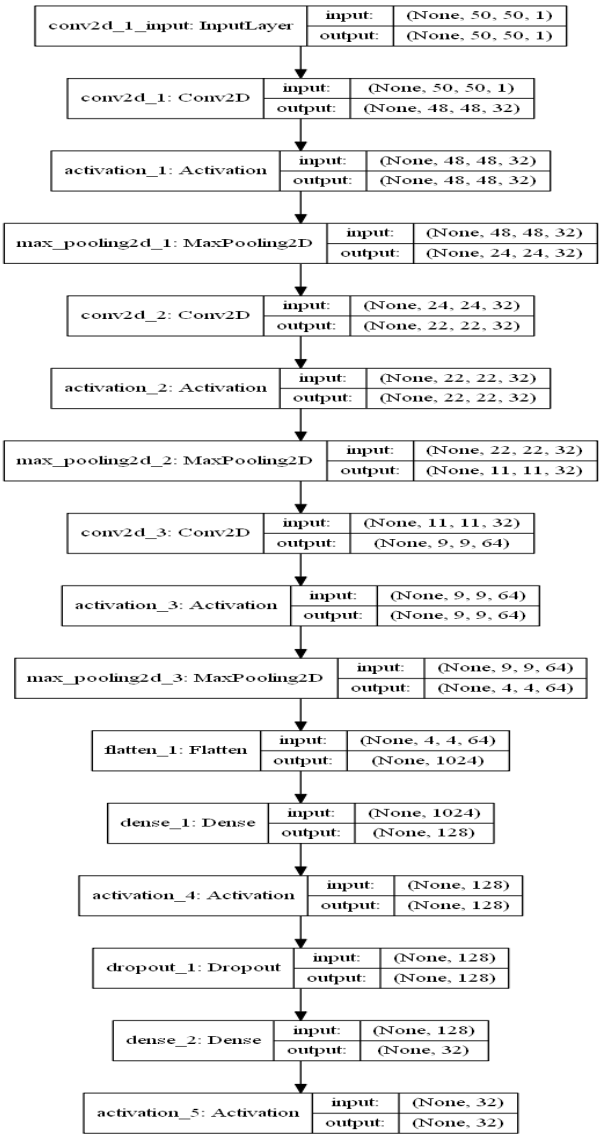
针对手写电话号码识别，我们采用了经典的 MNIST 数据集作为训练集，构造了一个卷积神经网络模型用于识别单个数字，并提出了以连通性为基础的数字分割的方法。

2 手写省份识别

2.1 模型的构建

2.1.1 模型可视化

利用 keras 的模型可视化函数 plot_model 将我们的模型示意图绘制出来，如下图：



2.1.2 主要部分说明

模型中首先是3层卷积3层池化，然后接了两个全连接网络，并以32个神经元和softmax激活结束模型。使用的损失函数为交叉熵函数，对应的优化方法为随机梯度下降法，并以预测的准确度作为度量。(模型的具体构建可见我们的代码 UseKerasForProvince.py)。keras 为我们提供了函数，可以很方便的从图片中直接产生数据和标签。并且可以用一些随机变换对数据进行提升。由于搜集到的数据集并不算大，而需要划分的类别又相当多，所以这个提升操作可以增加最终模型预测的准确性。我们选择的随机变换有剪切变换和随机放大。

2.2 数据预处理

我们通过邀请身边同学书写，加上从网上搜集艺术、书法字体以及同学们分享的数据，获得了最原始的数据集，每个省份大约有100多张，但这些数据远远不够，所以我们采用了一些方法来丰富数据集。

2.2.1 单字拼接

我们从中科院网站上下载了汉字数据集，使用MATLAB解码后，挑选出其中可以组合成省份的汉字，并利用MATLAB的矩阵拼接函数cat将两个汉字拼接成一个省份。这样拼接的数据也占了数据集的一大部分。

2.2.2 去噪

- **对二维double型图像实现去噪**:
  mynoise(x_noise)函数实现对二维double型图像x_noise去噪。首先设置阈值，将小于阈值的小波系数置为0，从而达到去噪的目的。过程中分别应用db小波和haar小波，两种小波去噪过程类似，均先分解低频图像，再分别对高、宽、对角线三个方向的高频系数进行阈值处理，最后进行小波重构。大致过程如下图：

图 1: double型图像去噪流程

- **对彩色图像去噪**:
  将彩色图像分为rgb三色，均为二维图像（彩色图像也要求为double型，否则做相应的转换）。对r、g、b分别应用mydenoise函数实现去噪，最后将三色重新组合为彩色图像，得到的便是原彩色图像的去噪结果。

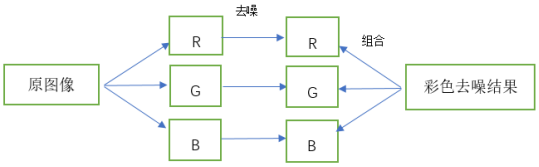
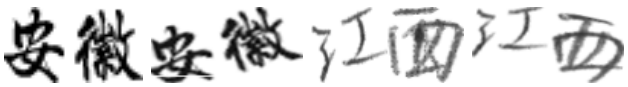


图 2: 彩色图像去噪流程

2.2.3 倾斜文字

现实生活中有很多人习惯向左或向右倾斜书写汉字，所以可以将图像向左或向右倾斜合理的角度以模拟这一类人的书写习惯，同时增大数据集。

倾斜文字的算法为：若待处理的图像为rgb图像，则分别处理R,G,B三个二维图像后再整合，处理方法与灰度图像类似。而对于二维灰度图像，我们先对即将旋转的图像进行预处理，将取值为0的像素点置为20，这样做是为了在旋转后能较好的区分图像原有的黑色笔记和旋转后添加到黑色背景。为了方便，我们先预处理这些图片的大小，使其均为50*100。为了保证旋转之后的图像不会有像素的损失，且旋转后的汉字图像与旋转得到图像边界没有交集，我们将图像的长宽各增加一倍，且使原图像置于增加后图像的正中央，增加的部分均置为白色，即像素值均为255。这样一来，新的图像大小为100*200，即使图像90度旋转，也不会超出范围导致像素点损失。



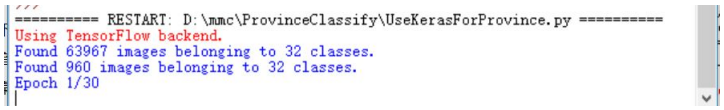
(a) 未处理艺术字 (b) 处理后的艺术字 (c) 未处理手写字 (d) 处理后的手写字

图 3: 汉字旋转结果

依照人们的书写习惯，我们将待处理图像向左或向右使用imrotate函数旋转15度，得到的新图像会有一块黑色的背景，我们将黑色背景设为白色以适应原图像。其中在旋转后的原图与背景交界处，一般会出现灰色矩形边界。前一部分的预处理使得此边界与内部汉字像素无交集，故易将这些边界均置为白色。最终得到了仅旋转汉字部分的图像，再对其取包围便可得到一张旋转后的字体。

2.3 模型测试结果

在经过1000次迭代后，我们的模型在训练集上的准确率为：88.17%, loss为：0.3799;在测试集上的准确率为：88.75%, loss为：0.5489.



(a) 模型训练起始

(b) 模型训练结果

图 4: 模型训练

3 手写数字识别

3.1 单个手写数字的识别

在解决单个手写数字的识别问题中，我们选择了经典的MNIST手写数字集作为训练集，也利用了它的测试集做交叉测试。我们在TensorFlow中读取数据集，整个训练集可以看成一个$N \times n^2$维矩阵，其中每个列向量均描述了一份数据。为了方便存储，我们可以将每个标签转化为各位取值为0或1的10维列向量。例如，用[0,0,0,0,1,0,0,0,0,0]表示数字4。最终得到$N\times$10维矩阵作为监督学习的标签。

其次我们用到了softmax回归模型，softmax函数为：

$$P(i) = \frac{\exp f_i(x)}{\sum_{k=1}^{K} \exp f_i(x)}$$

其中$f_i(x)$为关于x的函数(一般为线性)，我们的目标便是通过优化$f_i(x)$中的参数使$P(i)$最大。这里的$P(i)$可以看做样本属于第i类的概率。

为了利用图片中各个像素点的信息，我们将图片中的各个像素的值与一定权值相乘并累加，再加上偏差，即为下式：

$$f_i(x) = W_{ij}x_j + b_i$$

除此之外，我们的代价函数选为交叉熵函数:

$$H_{y'} = -\sum_i y'_i \log(y_i)$$

由于：

$$\sum_i y'_i \log \frac{y'_i}{y_i} = \sum_i y'_i \log y'_i + H_{y'}(y)$$

其中 y 和 y' 分别为预测的概率分布以及真实的概率分布。可以发现交叉熵函数越小，真实的概率分布与预测的概率分布越接近。

我们选择的训练模型是采用梯度下降算法，得到使代价函数最小时相应的权值W和偏置$\mathbf{b}$, 然后采用多次循环重复以上操作，每次循环随机选取100个样本，且每次进行梯度下降优化时将上一次运行得到的参数作为输入值。

最终在测试集上的准确率为90%以上。

```
======= RESTART: E:\Catmint\MathModelingHW\FinalProject\test_mnist.py =======
Extracting MNIST_data/train-images-idx3-ubyte.gz
Extracting MNIST_data/train-labels-idx1-ubyte.gz
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
Download Done!
Accuarcy on Test-dataset: 0.9168
>>>
```

图 5: 单个手写数字测试结果

3.2 电话号码的识别问题

3.2.1 电话号码数据集的获取

电话号码的数据集来源主要有两部分：

- 我们从 MNIST 中解码得到大量单个手写数字图片，从中随机选出11个组合拼接成一串数字，拼接时两个数字的间距从0 3中随机产生。
- 对真实的手写电话号码照片进行类似于手写省份的预处理。

3.2.2 数字分割问题

首先根据像素点的连通性对电话号码进行粗分割。

若电话号码两两数字之间无笔画相连，而又因为每个数字个体是连通的，所以理论上会分割得到11个数字，即为正确分割。如下图，每个数字为一个连通分量。

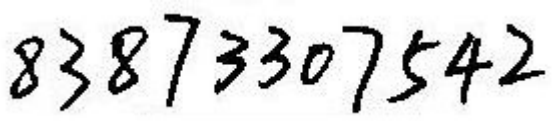


图 6: 每个数字均为连通分量

然而，现实生活中人们手写的电话号码往往字迹潦草，难免会在两个数字之间有笔画的连接，这样一来，连接的两个字便会被粗分割操作认为是一个数字。从而得到的分割结果个数少于11。如下图的7和3.

图 7: 两个数字为一个连通分量

假设分割所得的数字共有 num 份，且 num 小于11，那么必有一些数字被当做是一个数字分割到了一组。我们这里考虑两个数字连通的情况。

一般来说两个连通的数字只有一条连通边，这是由写字连笔习惯所导致，而多条连通边的情况较为少见。若两个数字仅有一条连通边，或者说仅有一点连接着两个数字。那么，将图像在高度上分为上下两份，则连接俩数字的那一点必然属于其中一份（如上图）。我们不妨对高度做平分，且假设连接点位于上半部分。那么下半部分二者由于没有连接点，故不连通。而单个数字上下两部分一般均为但连通。所以一般来说，上下两部分连通子集的个数不同，可由此大致判断是否为两个数字。

然而，实验中我们发现，有些单个数字在做完上下分割时，由于书写习惯，导致上下部分连通子集个数并不相同，比如4的手写体上半部分常常是不连通的。所以我们还需加上宽度的判断，一般地，两个数字的宽度小于一个数字。所以我们以电话号码总长11为原则，按照宽度排列各个满足

上述条件的分割子集并有宽到窄选择相应的个数，作为带分割的数字对。

对于上一步得到的数字对，从数字组像素宽度的一半对数字对进行分割一般可以得到较好的结果，实验结果如下所示

3.2.3 手写电话号码分割结果

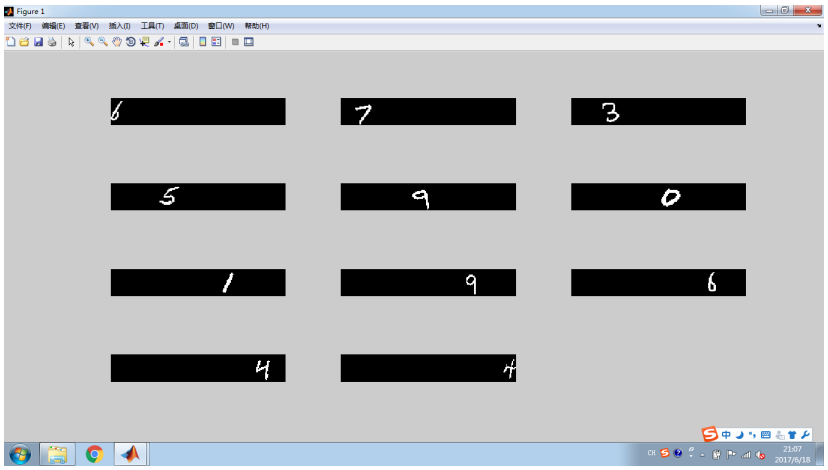


图 8: 上一张图分割的结果

4 收获与总结

在完成本次大作业前，机器学习在我们眼中一直是一个高深的理论。尽管为了解决问题查阅了各种文档、请教同学、多次为代码中的bug焦头烂额，但我们仍在学习机器学习的过程体会到很多乐趣。从装好TensorFlow，跑通了训练 MNIST 的程序，再到渐渐熟悉 keras 的各种接口和神经网络的构造，再到终于利用选择的模型跑出了满意的结果。感到自己也终于算是步入了机器学习的大门，也更深刻地意识到还有很多知识要去学习。在完成作业的过程中，和队友分工合作，积极讨论也让我们体会到了团体的重要性，受益匪浅。

5 不足与展望

虽然花了不少时间精力去准备，本次大作业的完成仍有很多不足：比

如训练手写省份名的数据集独立性较差，大部分是通过变换生成。其次是，原本我们解决数字分割问题的设想是利用拼接好的电话号码记录正确的分割范围，以此为粗分割点打上label,来训练可以识别正确分割点与错误分割点的模型。但试验了之后，模型无法收敛，再加上时间紧张，所以这个设想也就未能实现。而尝试利用连通性解决分割问题，对于一些特殊的数据也仍然会有错误分割。希望以后可以想出更好的算法来解决这个问题。