基于神经网络的英语单词识别系统

2024 年 5 月 26 日

摘要

许多成功的神经网络模型被用于手写体识别领域内。为了加快高考英语语篇语法题型的批阅速度，本文构建一个基于神经网络的英语单词识别系统。本文采用了两种神经网络模型，多层感知机和卷积神经网络模型并测试它们在数据集上的表现及改善运行效率。较优者与字母分割模型被用于搭建可运行的识别系统。在初步测试下，该识别系统可以较为准确的完成识别，并将答案框定在一定范围内。

1 问题描述

在日常英语测验中，题型主要分为两大类型：主观题与客观题。为了提高阅卷效率，学生会在特制答题卡上填涂答案。试卷中大部分题目可以用 A，B，C，D 等表示答案，可以使用机器快速阅卷。但是客观题中的语法部分与主观题中的摘要，翻译，写作还需要老师批改，加重了老师的工作负担。

不过经观察，语法部分有使用机器批改减轻工作量的可能。首先，答题卡正面专门有一块区域为语法部分设计，共 10 行，每行一空。下划线规范了学生的书写，使其容易识别。其次，每空的单词数较少，答案通常为 1～3 词。考察的词汇相对较为集中，在高考考纲范围内，可以辅助识别。

本文试图搭建一个可以用于识别单个单词的程序，减轻阅卷老师负担，加快阅卷效率。

2 文献调研

这样一个单词识别系统与手写字体识别密不可分。手写识别 [6] 过程通常包含：手写区域识别，图像降噪，行分割，单词分割与识别等。由于模型目的，这些过程被简化至单词识别和字符识别。

经文献查阅，有一下几种场见方法。其中一种基于 Long Short-Term Memory 模型的多维 Rucurrent neural network 方法 [2] 。相较于普通的一维 RNN，它更加适用于处理二维图片。图片中每个像素点的信息与其上下左右的像素点紧密关联。如果将图片压缩为一维信息，则每个像素点会失去与另一个维度相邻像素点的联系，这会导致识别准确率的下降。文章中提出了一种解决办法，每个格点除了像素信息作为神经元的输入外，还要将它上方与左侧的神经元输出一起输入它的神经元。这样它就与其左上方的所有像素点形成联系。

使用卷积神经网络也是一种常见的方法。一个窗口在图片上从左到右，从上到下滑动。它每移动一次就将窗口内的信息记录为一维向量。在将这些向量传递到剩余神经网络中实现预测。对一个窗口内的所有信息同时加工使其保留了原图中部分联系信息，使得它比将图片直接压缩成一维的模型表现更优。

[5] 给出了一种用于将字符串分割为字符的算法。由于字符与字符之间的分割线应当与字迹距离较近。基于此，首先将照片转换成 0-1 组成的二维序列并将连通分量分离出来，进行初步分割。接着试图根据背景像素点与最近笔迹像素点的距离尝试画出分割线。单个字符识别模型对这个分割结果

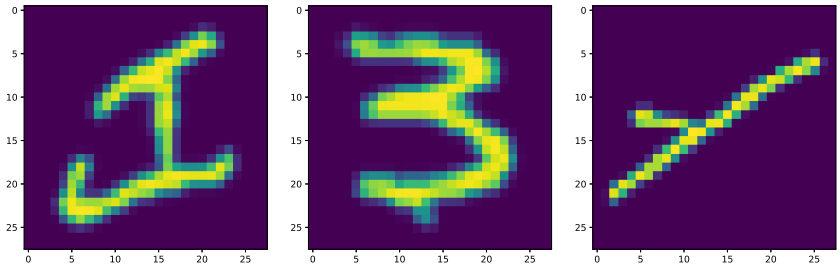
进行测试，用对识别结果的自信度评估分割方式。这个过程不断重复直到自信度达到阀值。

3 研究思路

为了搭建识别单个字符的神经网络模型，需要从以下几方面进行探究。

3.1 数据库的查找

模型的训练以及最终测试准确度离不开数据库的质量。所以寻找一个优秀的数据库意义重大。经过文献查阅，本次探究使用 EMNIST 数据库。



(a) char a (b) char w (c) char y

图 1: EMNIST 示例

3.2 合适模型的选择

前文提到的三个模型，多维长短期记忆递归神经网络，卷积神经网络，多层感知机需要慎重考虑。模型的选择会直接影响程序的运行时间，从而影响训练效率。复杂的模型固然会给出较为准确的预测，但是会花费大量时间和数据去训练。所以需要对这些模型的效率与准确度予以综合考察。

3.3 模型的训练和测试

需要将所有数据分为两组，测试组与训练组。模型的训练不只是仅仅记住训练数据，而是要从训练数据中找到规律。所以一个好的模型在测试组与训练组的准确率差异应该适中。

4 具体实现

EMNIST 手写字母照片的大小都是 28*28，每个像素点是一个 $[0, 255]$ 间的整数。将照片作为输入，希望得到一个 26 维向量输出，

$$out = (y^1, y^2, \ldots, y^{26})$$

分别代表输入照片是 a，b，…，z 的概率。在通常情况下，神经网络的 26 维向量输出无法满足形成概率分布的条件，即 $\forall y \in \{y^1, y^2, \ldots, y^{26}\}, y \geq 0$ 且 $\sum_{i=1}^{i=26} y^i = 1$。

所以为了满足这些条件，我们对输出向量再进行 softmay 转换。具体来说，为了处理负数元素，需要一个 $\Re \to \Re^+$ 的映射，此处我们选择取指数。最后将其归一，形成概率分布。

$$(p^1, p^2, \ldots, p^{26}) = (\frac{e^{y^1}}{\sum_i e^{y^i}}, \frac{e^{y^2}}{\sum_i e^{y^i}}, \cdots \frac{e^{y^{26}}}{\sum_i e^{y^i}})$$

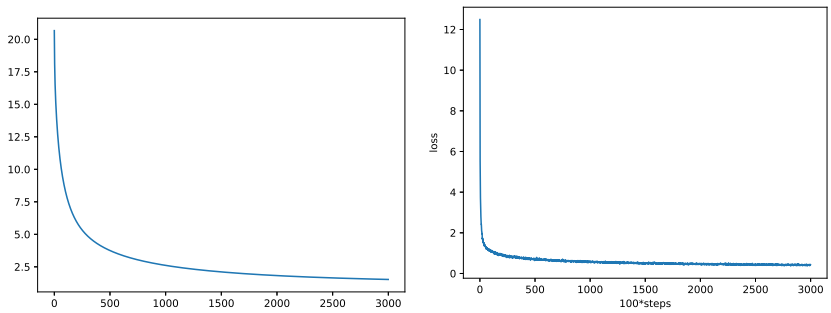
我们还需要一个能描述整个模型表现的参数。准确率在这里就不是特别理想。因为它只是挑选了概率最大的那个字母，具体概率在比较中丢失了。对一组训练照片输入 $input = (x_1, x_2, \ldots, x_m)$ 和目标答案 $target = (t_1, t_2, \ldots, t_m), t_i \in \{1, 2, \ldots, 26\}$，我们定义 $loss = -\frac{1}{m} \ln \prod_{i=1}^{m} p_i^{t_i}$。对于输入 x_i，照片里的字母是 t_i 的概率 $p_i^{t_i}$ 越接近 1，$loss$ 越小，模型越准确。

4.1 多层感知机

我们先设计一个简单的多层感知机模型。其中包括：一层包含 $28 * 28 = 784$ 个神经元，中间隐藏层包含 100 个神经元，用 tanh 作为非线性激发函数，最后输出层包含 26 个神经元。层与层之间都是完全线性连接的，其权重矩阵，偏差分别为 $W_1 \in \mathsf{M}_{100\times784}$，$W_2 \in \mathsf{M}_{26*100}$，$b_1 \in \mathsf{R}^{100}$，$b_2 \in \mathsf{R}^{26}$。

$$x \leftarrow W_1 \times x + b_1$$
$$x \leftarrow \tanh x$$
$$x \leftarrow W_2 \times x + b_2$$

我们将整个训练组作为输入，循环 3000 次，学习率为 0.1，并计算 $loss$，执行反向传播算法，以及计算在测试组上的准确率。这个程序总共执行了 8m1.5s，在测试组中的准确率为 57.83% 执行效果如左图所示。可以看到其效率有待提升。我们使用随机抽样，从训练组中每次挑选 32 张图片作为输入。循环 300000 次，学习率 0.1。程序执行时间 1m50.9s，在测试组中的准确率为 81.63%。为了更好比较两者，随机抽样每训练 100 次取 $loss$ 的平均，执行效果如右上所示。



(a) 整个训练组输入　　(b) 随机抽样输入

图 2: 不同输入训练表现对比

随机抽样虽然会带来一些波动，但是它的效率与准确率都大大超过了将整个训练组作为输入。

但是对图表进一步观察，可以发现两者在训练刚开始时 *loss* 都很大。如果我们让模型从 26 个字母均匀随机挑选一个，它的 $loss = \ln(26) \approx 3.26$。

选用的激发函数为 $\tanh x$，它将全体实数不均匀映射到 $(-1, 1)$ 内，如图 3a，3b所示。可以看到激发后大部分神经元都分布在接近 ± 1 范围内。在执行反向传播算法时，神经元的修改正比于其 tanh 的导数。

$$\frac{\mathrm{d}\tanh x}{\mathrm{d}x} = 1 - \tanh^2 x$$

所以当 $\tanh x$ 接近 ± 1 时，神经元的参数将几乎不会得到任何修改。图 3c中每一列代表一个中间层的神经元，当 $\tanh x > 0.99$ 时为白色格子。可以看到在图 3c中存在一些神经元没有改动。

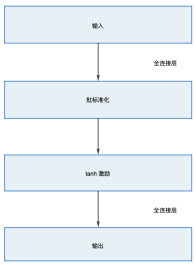
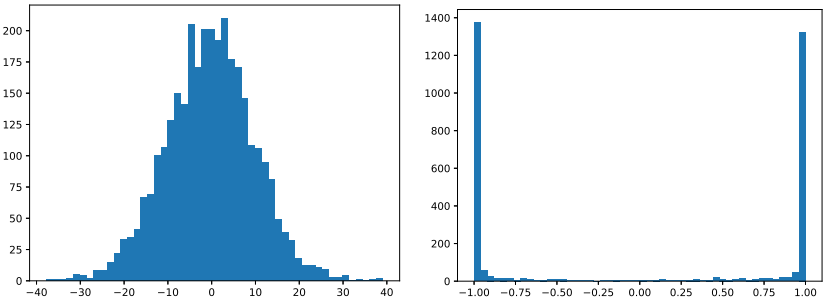


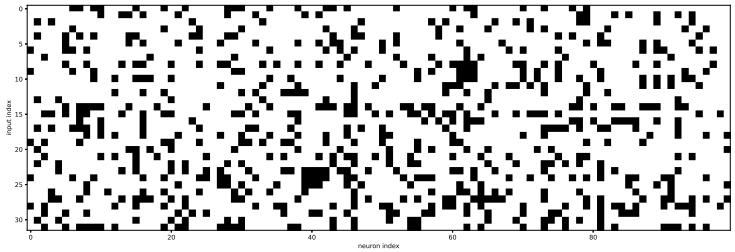
图 4: 程序示意图

为了解决这两个问题，我们使用批标准化方法 [4] 和 Kaiming 权值初始化方法 [3] 这两者的存在使在进入激励函数过后计算结果的分布维持与之前的分布大致类似，并且在层与层之间传播时方差保持一致。每个区域都有分布会让数据更有价值，达到更好的训练效果。

我们将这两种方法运用在程序中，执行 300000 次，学习率 0.1。如图 5a，可以看到加入两种优化方法后在相同条件下达成了更好的训练效果，测试组正确率 88.88% 程序执行时间 2m52.4s。程序执行情况如图 5所示。可以看到尽管程序有较高的正确率，但还是会分辨不清一些比较相近的字符，如 g 与 q 和 i 与 l。

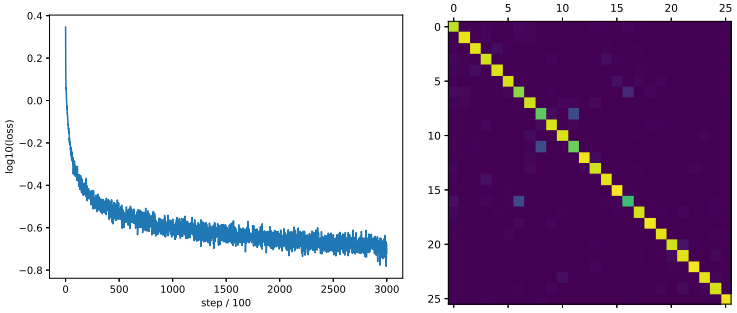


(a) 激发前分布　　(b) 激发后分布

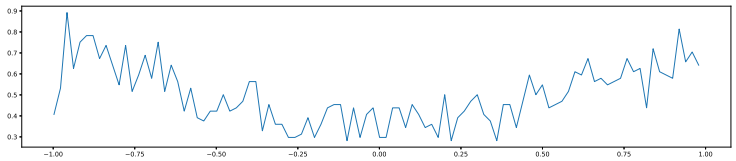


(c) 神经元激发状态

图 3: 中间层状态



(a) 加入优化方法　　(b) 混淆矩阵



(c) 激励后分布

图 5

4.2 卷积神经网络

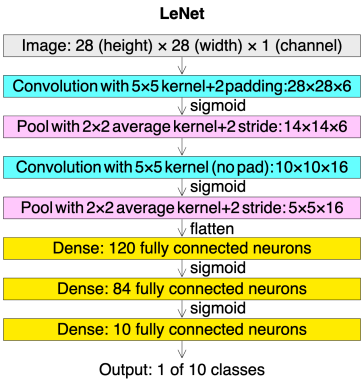
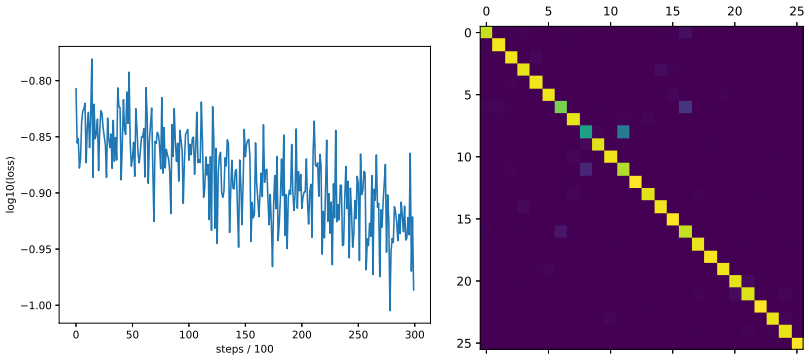


图 6: 程序示意图

这里我们采用类似 **LeNet** 的模型。28*28 的图片进入卷积层，卷积核大小为 5*5，并在图片的四周填补两行或两列空白。接着用 *sigmoid* 函数进行整流，再进行平均池化。如此重复两遍，输入大小为 28*28 的照片被转换为了 400 维向量。由于硬件配置问题，我们的模型只能在此之前做到相同。接着它输入进含有 100 个神经元的全连接隐藏层，最后进入含 26 个神经元的输出层。程序训练 30000 次，运行时间 6m33.2s，在测试组中的准确率高达 93.37%。可以看到比前面明显运行的慢，这是因为随着神经元层数增多，尽管变量个数保持大致一致，反向传播算法将耗费更多算力。执行效果如图 7所示。同样可以看到，模型不太能够分清 g，q 与 a 和 i 与 l。



(a) loss function

(b) 混淆矩阵

图 7: 卷积神经网络

4.3 成果展示

根据高考英语答题卡的设计，我们假设图片已经完成了对空白部分的裁剪。查阅 EMNIST 数据库对图像的处理方法 [1] 后，我们将照片进行相应操作并裁减输入训练好的模型。通过统计照片中每列的黑色格子数，将一个单词拆分为字母。然后按照 8 进行处理，并输入卷积神经网络模型，得到相应的概率分布。

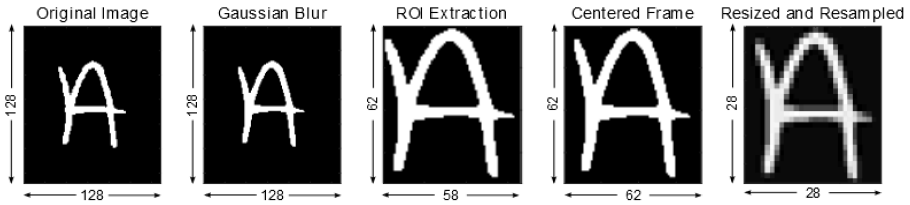


图 8: EMNIST 照片处理

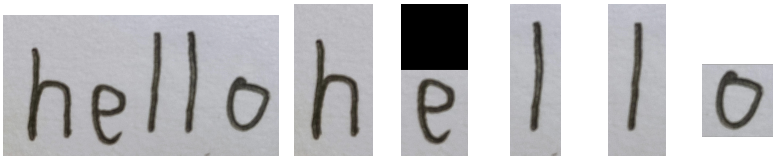


图 9: 处理前图片

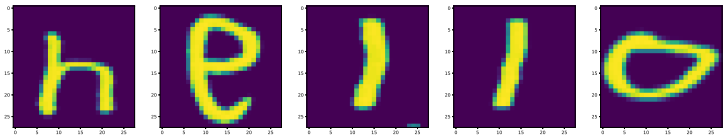


图 10: 处理后图片

编号	最可能	概率	次可能	概率
1	h	99.66%	n	0.27%
2	p	57.75%	e	33.83%
3	i	64.93%	l	34.18%
4	l	60.68%	i	39.18%
5	o	98.71 %	d	0.64%

我们可以观察到，当最大概率和次大概率相差几个数量级时，模型的预测是准确的。当最大概率与次大概率在相同数量级时，答案包含于两者。我们可以使用枚举法来尝试复原单词。

5 展望

5.1 模型选择

在本次探究中，我们没有使用多维长短期记忆递归神经网络方法。这是因为它的训练需要特使的程序设计，并且需要更多算力支持，在作者能力范围之外。它的表现呢应该优于卷积神经网络模型的表现，能更准确的对单个字母完成识别。

5.2 单词分割

在我们的模型中，通过统计每列中颜色较深的格子数量来决定是否该列处于字符与字符之间，进而完成划分。但是考生可能字体歪斜，有连笔的情况出现这种简单分割方法出错概率将会大大提高。如果可能，我们应该使用更准确的，与字体依赖度不大的分割方法来确保模型稳定性。

5.3 字符识别

这个模型对照片的预处理依赖很高。经过尝试，只有完全按照 EMNIST 处理方法才能获得较好的准确率。这对扫描仪器要求较高。

当模型识别结果出现多种较大概率可能时，我们可以设计通过枚举并联系高考考纲词汇的方式来加强识别结果。

参考文献

[1] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017.

[2] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[5] Yi Lu and M. Shridhar. Character segmentation in handwritten words—an overview. *Pattern Recognition*, 29(1):77–96, January 1996.

[6] U.-V. Marti and H. Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, November 2002.