机器学习工程师纳米学位

算式识别

毕业项目 曾奕鑫 优达学城
2019 年 7 月 2 日

I. 问题的定义

项目概述

本项目通过深度学习算法，从图片中识别出算式文字序列。这种识别场景常见于图片验证码，是一种序列 OCR 识别问题：OCR 是指光学字符识别（Optical character recognition）；序列是指图片中的字符数量不定，在 OCR 中是一种很有挑战性的问题。

任务提供了 100,000 张图片及其对应的文字标注，本项目将基于这批图片进行训练、验证、测试。

问题陈述

序列 OCR 问题，传统的做法是分两步：首先从图片中切割出单个字符，然后对每个字符进行识别，最后把识别结果组装起来。随着深度学习技术的演进，出现了端到端的解决办法，即输入一张文字图片，没有独立的分割和识别动作，模型返回识别结果，取得了很好的效果，比如 CRNN[1]。
本项目将使用 CRNN 的方法来解决这个问题。CRNN 作为一种深度学习技术，实践过程也将遵循深度学习的流程，分为 3 步：1.在训练数据上学习得到模型；2.在验证数据上验证模型的性能，预估模型的好坏；3.在测试数据上得到最终的模型性能评价结果。

评价指标

本任务最可能的应用场景就是验证码识别，这种场景要求识别出的文字和图片中需完全匹配才能通过。因此本项目采用的评价指标是准确率：

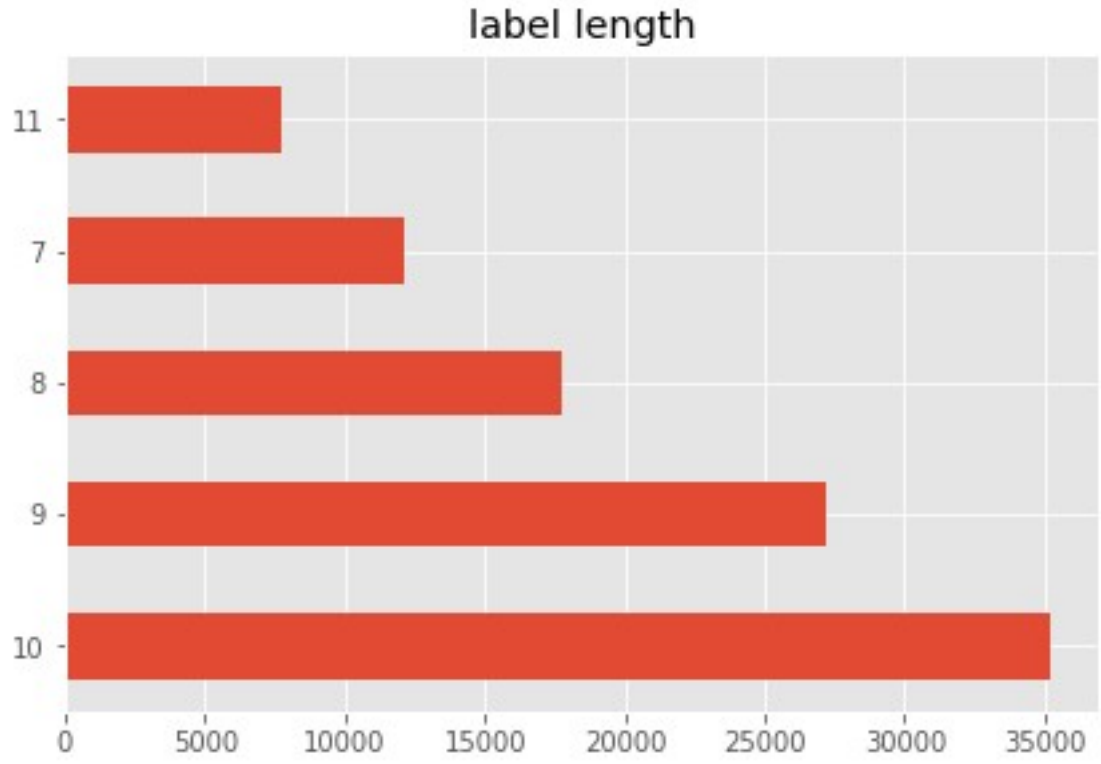
$$准确率 = \frac{识别正确的样本个数}{总样本个数}$$

II. 分析

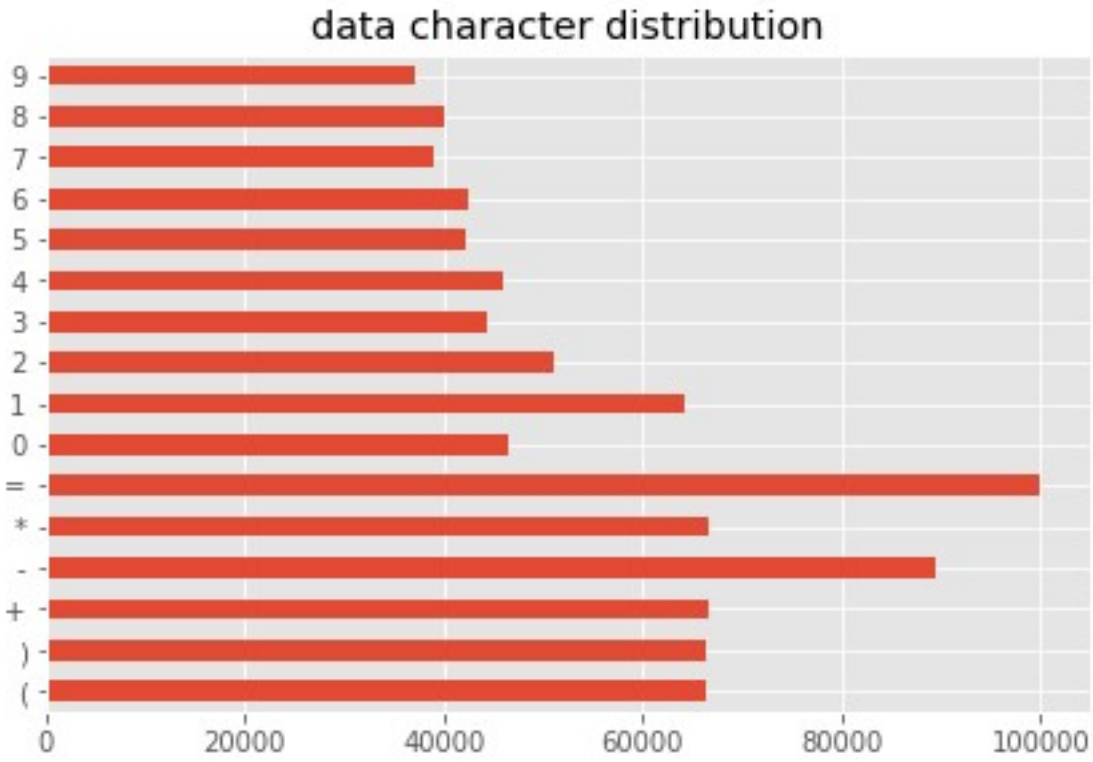
数据的探索与可视化

任务提供的数据集共 100,000 张图片，每张图片分辨率均为 300*64。算式由 3 个数字、2 个运算符、等号及运算结果组成，运算结果位数不定，运算中还有括号，因此文本的长度不是固定的。需要注意的是，乘号只会以“*”的形式出现，所以十字符号只能是“+”。这个数据集将是本次任务中主要使用的数据集，首先会预留 10,000 张图片用作为测试集，查看程序的准确率；剩下的数据中 80,000 张用作训练数据，10,000 张用作验证数据。

下面两图是对数据集的可视化统计。
Label length 直方图说明算式长度不定，分布从 7 到 11 都有，以 10 长度的数量最多，对模型的变长序列识别能力有较高要求。

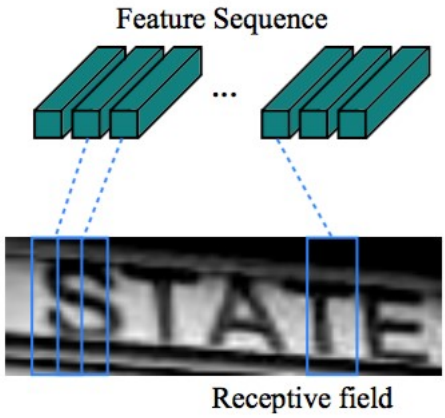


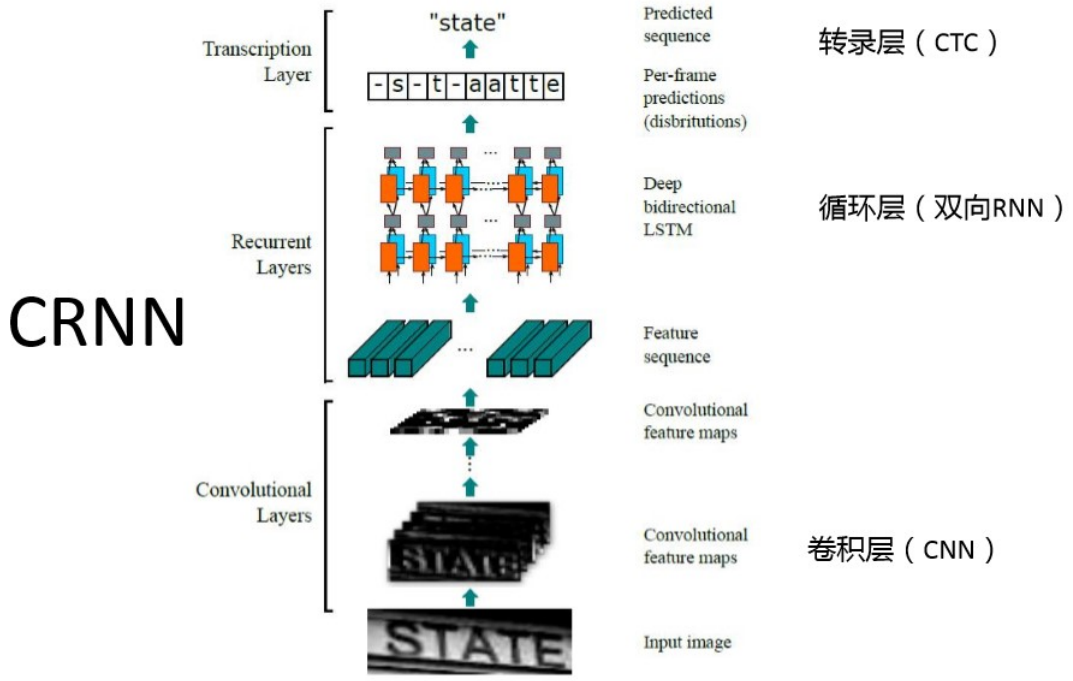
data character distribution 直方图体现了整个数据集的字符分布情况，共 16 种字符，基于字典的算法是可以考虑的。算式结果中“十几”运算结果会多一些，因此“1”的出现次数会多一些，到 60,000 多次，其他数字的分布差不多；运算符号中由于负数的存在，所以“-”的分布多一些，其他符号基本出现次数相同；“=”共 100,000，和数据集总数量一致，这是符合算式的特点的；“(”和“)”数量相等，这也是符合算式的特点的。从分布来看数据比较干净，呈均匀分布。



算法和技术

CRNN 由 3 个部分组成：卷积层、循环层、转录层。
卷积层提取图片的特征，并输出为特征向量，每一个特征向量表征的是高度为图像高度的矩形图像帧，图像帧组合起来就是对原图像的完整特征表述。

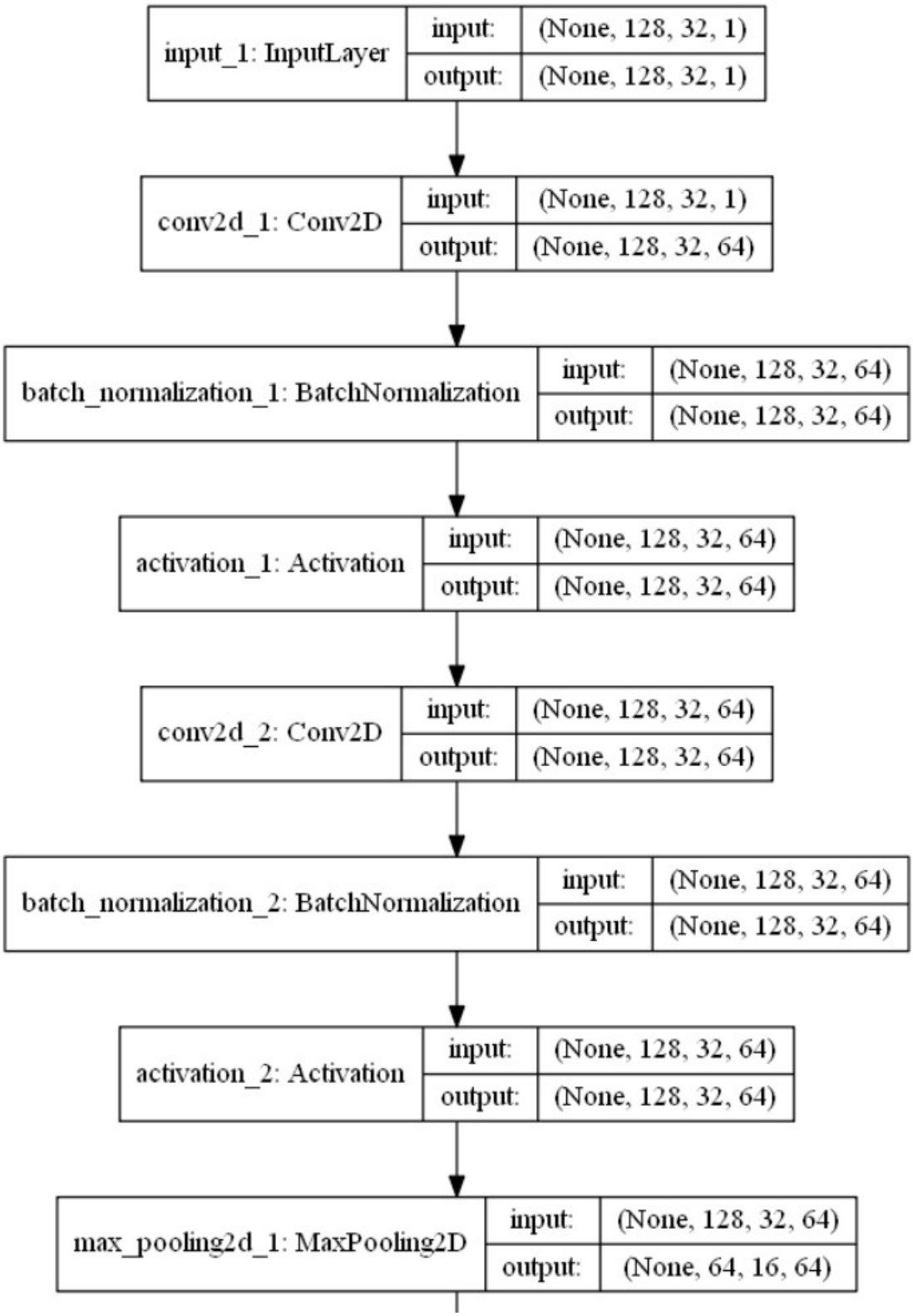




卷积层：

卷积层具体的网络结构设计如下，一共分 4 个部分。

前 3 部分是类似 vgg[2]的卷积神经网络结构，区别在于每 1 部分是在上一部分的基础上滤波器数量逐渐增多，以期在更高层次的感受野上获得更丰富的特征表述。下面着重介绍其中 1 个结构：接受到图像输入后，卷积层 conv2d_1 使用 64 个滤波器对输入进行卷积;batch_normalization_1 是批归一化层[3]，这层通过归一化调整数据的分布，可以有效减小梯度消失，并通过较小 covariate shift，加快模型收敛；activation_1 是激活层，这里用的 relu 函数，也是用于减小梯度小时的现象；接下来重复一次上述三层网络层，进一步解析图片特征；最后使用 2*2 的池化层来降低模型参数，减小过拟合的情况。



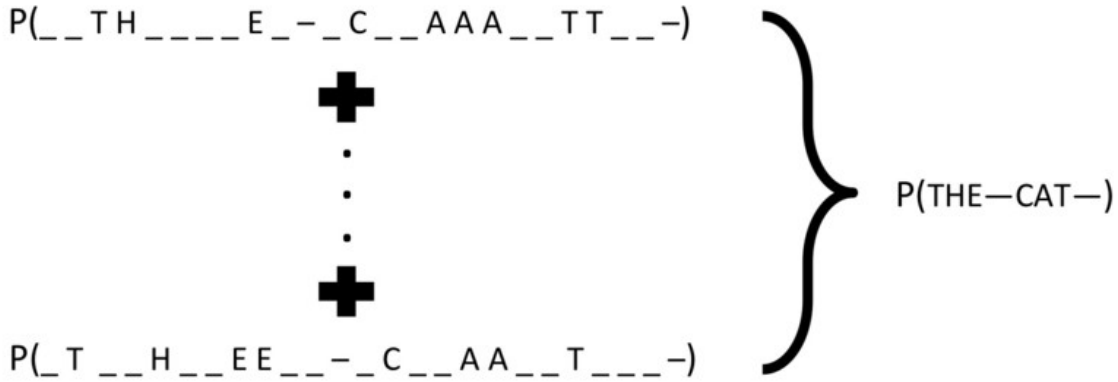
第 4 部分是先把特征展开为特征向量，然后通过全连接层降维。经过前 3 步后，输入图像由(128,32,1)变成(16,4,256)的特征矩阵，展开后变为(16,1024)，经过全连接层降维后，输出(16,256)的特征向量序列，即一张图片从左到右被分成 16 份，每份 256 个特征维数。

循环层：

这里使用了双向的 gru[4]，来对上一层的特征向量序列提取从前往后和从后往前的上下文信息的特征。经过两层双向 gru 后，再使用全连接层降维到(16,17)，实际上 17 是字典的长度，到这里，(i,17)实际上就是模型对图片第 i 份图片帧的 17 个字符的概率估计集合。

转录层：

转录的过程，体现在模型的损失函数的定义上，这里使用的损失函数是 ctc loss[5]，邻近的预测如果属于同一个字符，ctc loss 可以将这些邻近的预测合并。



基准模型

本身模型会要求 99%的准确率。

下表是 18 年以来主流会议中 OCR 识别取得的进展，IC 是 ICDAR（国际文档分析和识别大会）的缩写[6]。可以看出，主流的识别算法都达到 90%以上的识别率，而本任务相对而言场景比较简单，所以准确率定为 99%。另一方面，要达到验证码识别的实用程度，也需要较高的准确率。

Conf.	Date	Title	IC03	IC13
'18-AAAI	18/01/04	Char-Net: A Character-Aware Neural Network for Distorted Scene Text Recognition	0.915	0.908
'18-AAAI	18/01/04	SqueezedText: A Real-time Scene Text Recognition by Binary Convolutional Encoder-decoder Network	0.931	0.929
'18-CVPR	18/05/09	Edit Probability for Scene Text Recognition	0.946	0.944
'18-TPAMI	18/06/25	ASTER: An Attentional Scene Text Recognizer with Flexible Rectification	0.945	0.918
'18-ECCV	18/09/08	Synthetically Supervised Feature Learning for Scene Text Recognition	0.947	0.94

'19-AAAI	18/09/18	Scene Text Recognition from Two-Dimensional Perspective		0.914
'19-CVPR	18/12/14	ESIR: End-to-end Scene Text Recognition via Iterative Image Rectification		0.913
'19-PR	19/01/10	MORAN: A Multi-Object Rectified Attention Network for Scene Text Recognition	0.950	0.924

III. 方法

数据预处理

100,000 张图片的数据集，将随机打散后，80,000 张用作训练集，10,000 张用作验证集，10,000 用作测试集。
数据集提供的图片分辨率为 300*64，经过测试缩放为 128*32 并不影响识别率，而更小的分辨率，可以让我们训练时并行处理的图片数量更多，从而加快训练速度。
由于颜色对识别起的作用不大，图像会从彩色图片转成灰度图，并对图片进行归一化，即让图片的均值为 0，方差为 1。

执行及完善过程

下图是截取的 CRNN 中 CNN 第一层的结构，大部分的参数变化体现在这个结构中：
1.输入。输入最开始是将 300*64 的彩色图片直接输入，由于显卡显存的限制，最多支持并行处理 128 张图片，训练完一轮迭代需要 340 秒左右；输入调整为 128*32 的灰度图后，可以并行处理 256 张图片，一轮迭代的速度加快到 128 秒左右。

```
Epoch 1/20
 - 282s - loss: 6.3710 - acc: 0.3714 - val_loss: 1.7559 - val_acc: 0.4136
Epoch 2/20
 - 282s - loss: 6.3162 - acc: 0.3855 - val_loss: 1.4359 - val_acc: 0.8115
```

（1）300*64 彩色图的迭代速度

```
Epoch 1/20
 - 129s - loss: 0.5433 - val_loss: 0.0953
Epoch 2/20
 - 128s - loss: 0.4217 - val_loss: 0.0805
```

（2）128*64 灰度图的迭代速度

2. dropout 层和卷积核个数的调整。由于 loss 曲线图表明模型有欠拟合的倾向，因此逐渐去掉了部分 dropout 层，并加倍了卷积核个数，期望用更丰富的特征得到对图片更充分的特征提取。从测试集准确率来看，识别率从 0.9721 提升到了 0.9882。
3.模型融合。上两步的调整后测试集准确率达到了 0.9875，通过对错分图片的分析，发现不同参数的模型对图片的错分能力并不一样，有互相弥补的可能性，并且算式本身有一个等号前后的计算关系，可以利用这个关系对模型进行融合。经过融合后，最终测试集准确率达到 0.9941。关键代码如下：

```
try:
    #如果算式从计算结果上就不对，使用模型3预测
    tem_list = pred_texts.split('=')
    if eval(tem_list[0]) != eval(tem_list[1]):
        img_pred3 = cv2.resize(img_pred, (width, height))
        img_pred3 = (img_pred3 / 255.0) * 2.0 - 1.0
        img_pred3 = img_pred3.T
        img_pred3 = np.expand_dims(img_pred3, axis=-1)
        X = np.zeros((1, width, height, 1))
        X[0] = img_pred3
        net_out_value = model_best3.predict(X)
        pred_texts = decode_label2(net_out_value)
except:
    img_pred3 = cv2.resize(img_pred, (width, height))
    img_pred3 = (img_pred3 / 255.0) * 2.0 - 1.0
    img_pred3 = img_pred3.T
    img_pred3 = np.expand_dims(img_pred3, axis=-1)
    X = np.zeros((1, width, height, 1))
    X[0] = img_pred3
    net_out_value = model_best3.predict(X)
    pred_texts = decode_label2(net_out_value)
```

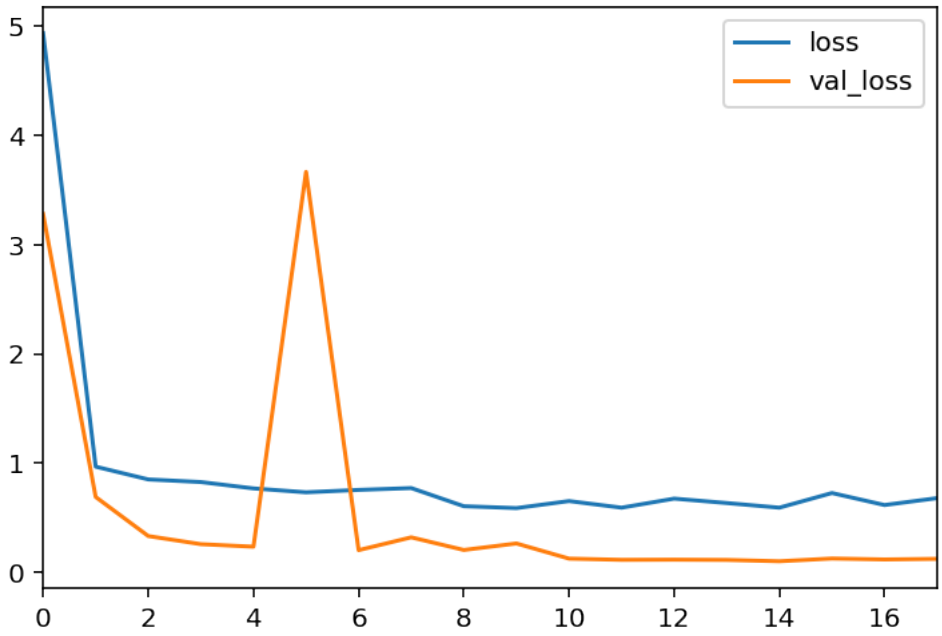
IV. 结果

模型的评价与验证

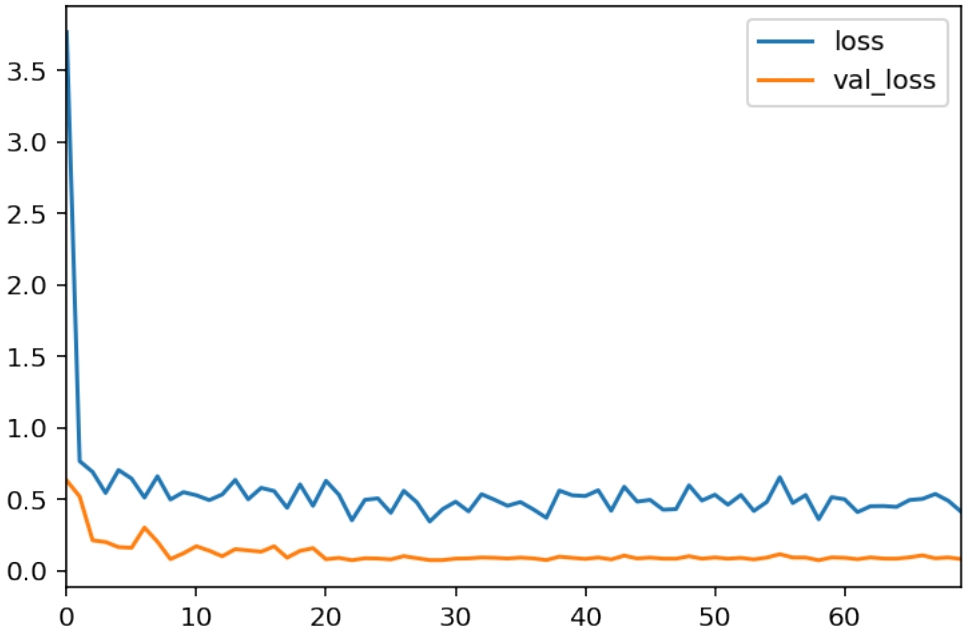
最终的模型是利用算式的特点对两个训练中较优的模型进行了融合。在单个模型的训练中，保存了迭代中的最低 loss 的模型。通过两个模型的训练 loss 曲线图可以看出，开始 loss 急剧下降，在 10 个迭代后，曲线趋于水平稳定，说明模型已经过充分训练达到收敛。

在测试集上的准确率，两个模型的准确率分别为 0.9882、0.9833，可以佐证，模型没有过拟合或者欠拟合。在这两个模型的基础上，利用算式需计算相等的特点对模型进行了融合，得到最终的模型，在测试集上准确率达到 0.9941。

基准指标为 0.99 的准确率，本项目的模型准确率为 0.9941，超过了基准指标的水平。如果把模型实际应用于验证码识别中，平均 10000 次识别，9941 次可以通过，只有 59 次会失败，是可以接受的。



（1）去掉部分 dropout 层后的 loss 曲线图

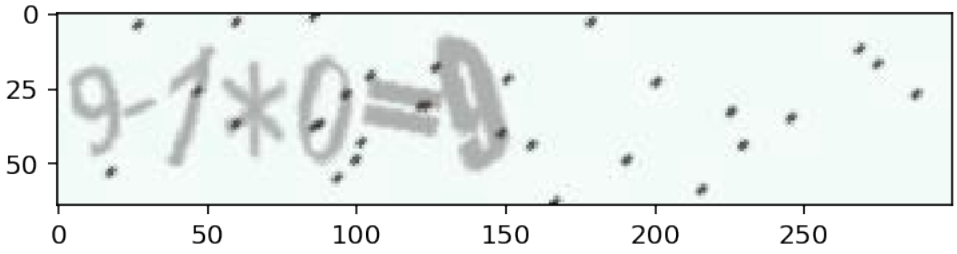


（2）增大卷积核个数后的 loss 曲线图

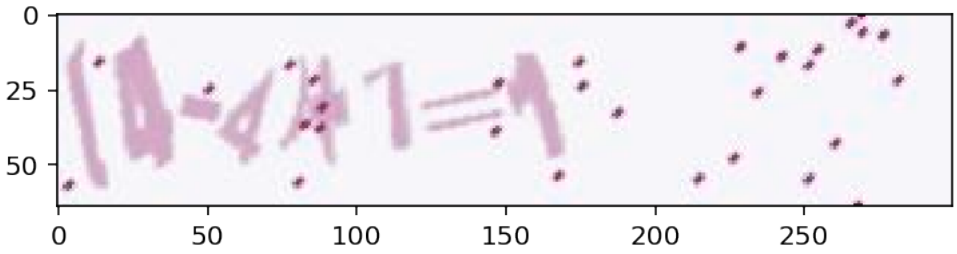
V．项目结论

结果可视化

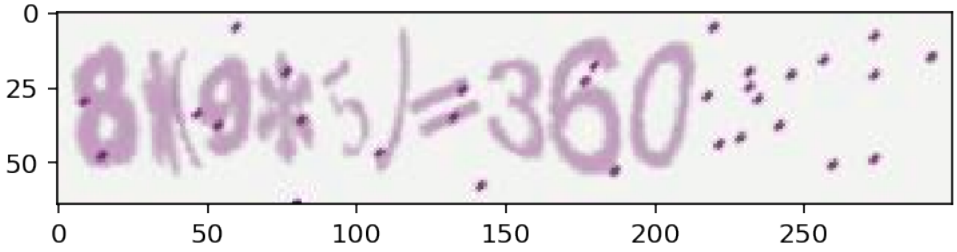
对于模型错分的数据，非常值得留意的。



（1）错误预测值为：9-7*0=9



（2）错误预测值为：(4-4)*1=1



（3）错误预测值为：8*(0*5)=360

从上图中的（1）可以看到，模型将“1”误识别为“7”了，事实上，肉眼去识别的话，也不一定能够准确区分，因为“1”或“7”算式都能成立；图中（2）加号和乘号区分问题，但是这个样本，人可以通过对算式的推导，得出应为加号，这说明模型对于上下文的结合还有提升空间；（3）的问题是噪点的影响，使得“9”被误认作了“0”，这引发出预处理还可以做得更多。

对项目的思考

模型初始的准确率已经达到 0.9721 了，在此基础上进行的提升，影响都很细微，在调参过程中，有些参数的调整并不能直接看到结果或者影响没有通过正确的维度可视化出来，最后采用的参数调整相信只是发挥了模型的部分潜力，模型还有其他的调优空间，这是遇到比较困难的地方。但是最后还是达到了预设的基准指标要求。

需要做出的改进

模型的 CNN 部分还有调整的思路，比如近年很火的 densenet[7]，对图像更好的特征表征与提取，应该可以进一步提升模型性能。

引用

[1] Shi B , Bai X , Yao C . An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39(11):2298-2304.

[2] K. Simonyan and A. Zisserman. Very deep convolu- tional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014. 5

[3] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.

[4] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[5] Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks[C]//Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 369-376.

[6] hwalsuklee. https://github.com/hwalsuklee/awesome-deep-text-detection-recognition[Z], 2019

[7] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.