

使用pytorch实现CTPN的一些心得

• 2018-12-10 Monday

之前使用keras实现的CTPN ([Detecting Text in Natural Image with Connectionist Text Proposal Network](#),[keras-ocr](#)) 在VOC2007_text_detection数据集和icdar2017rctw数据集上训练了一个中文文本检测的网络,感觉效果还不错.当然,一开始看的论文感觉EAST(An Efficient and Accurate Scene Text Detector)和[PixelLink](#)效果都很不错,所以使用预训练的权重测试了一下效果,个人觉得在长文本检测中效果不是很理想,所以就没有折腾了,便主要投入到CTPN上来.

既然都有keras实现的现成的模型了,为什么还要**自寻苦恼**使用pytorch实现呢?主要是keras版本的模型权重太大了(大约有142M),在实际应用中觉得这个模型太大了.使用pytorch实现倒是觉得引人入胜,前景远大,道路却是曲折的.遇到了许许多多的问题,所以在这里一一的记下自己还能记住的一些吧.

(在此更新一下,pytorch的CTPN模型只有**68M**)

一些问题及解决

pytorch中的bi-directional RNN

keras中的双向RNN封装得非常人性化,只需要知道输入(甚至都不需要知道输入,只要是序列就行)输出的维度是什么基本就可以了,比

如: `Bidirectional(GRU(128, return_sequences=True))`,而pytorch中却没有如此高度封装的API,必须要对循环神经网络的结构有深入(至少知道输入输出维度,中间计算过程吧)的了解不可.所以为了扫盲(我刚开始实现的时候,真是碰了太多的钉子了,比如不知道如何将CNN的输出reshape才是将需要的那一维作为序列长度而不是特征或者batch size等),我查阅了大量的资料(分享一下还保存的参考资料,还有一些找不到了😭,至于那张展开的图就不贴了,文章中都有).

- [Understanding LSTM Networks](#)
- [How to Reshape Input Data for Long Short-Term Memory Networks in Keras](#)(虽然是keras,不过感兴趣的主要是对输入数据的处理)
- [Recurrent Neural Networks Tutorial](#)
- [coursera上的Sequence Models](#)(看完之后对于rnn有了很直观的认识,毕竟讲得十分的清楚啊,同时自备一张纸边看边画一画是很不错的体验)
- [LSTMs for Time Series in PyTorch](#)
- [官网的文档](#)

大概就记得这么多了,希望想起来的时候再补充了.

把这些看完,再自己开个notebook一边练习,经过这几轮下来,我不信搞不懂或者不会使用pytorch的LSTM 😊

损失函数:sparse_softmax_cross_entropy_with_logits

keras中使用这个损失函数方便多了,直接调

用 `tf.nn.sparse_softmax_cross_entropy_with_logits` 就行,我翻开pytorch文档一查,这文档没有现成的,我横竖睡不着,查了好几夜,才终于在文档的角落中发现,整篇规规整整都写满了各种基础的loss function.本想调整一下模型的输出和ground truth,然后使用[BCEWithLogitsLoss](#)替代的,但是效果似乎不是很理想.所以最后还是Google了一下,我就解决了.

经过以上的一些探索和尝试,loss保持在0.448左右,这实在是很不理想的

修复以上问题的前三个epoch,loss在后面下降得极慢



还有一些低级错误和粗心导致的问题

这些问题并没有直接导致错误,相反,网络跑得十分欢快, loss一路飙升,洋洋洒洒,就是结果不理想.鉴于这些是低级错误,一方面是怕误导他人,另一方面是觉得好像没有多大参考价值,自己以后肯定尽量努力不再犯. 😊

解决了低级错误之后,loss降到了0.2605,这依然不理想,但依稀可见最终的结果了,训练了120

多个epochs,loss虽然还在降,但是在万分位,或许是lr过小了.

一些尝试和探索

通过掉进及爬出以上几个坑之后,我还是对自己实现的模型心存芥蒂,本想在再瞎鼓捣一通,可是限于时间而又看到了些许效果,所以便尝试了其他的一些方法.

optimizer

使用了Adam和SGD,无论是怎样的组合,个人都觉得在这个任务中SGD工作得很不错,而Adam无论怎样改参数,学习率都只在第一轮稍好些,后面就开始蒙圈了.所以最终使用了SGD

learning rate

使用的learning rate跟基础网络是否使用pretrained的权重,是否freeze(设置requires_grad为False)有关,通常初始采用的是1e-3,最后结果显示,无论是不是使用pretrained或者是不是需要finetune,1e-3都是极好的

是否使用bi-rnn

将bi-rnn(我用的是GRU)那段代码注释掉之后训练模型,发现可以将loss训练到0.38左右,从此再不能降低了.并且论文中也说了使用birnn要好些,所以就没有折腾了,本想尝试一下GRU和LSTM的,鉴于机器的计算能力和个人时间精力有限就没有尝试了.

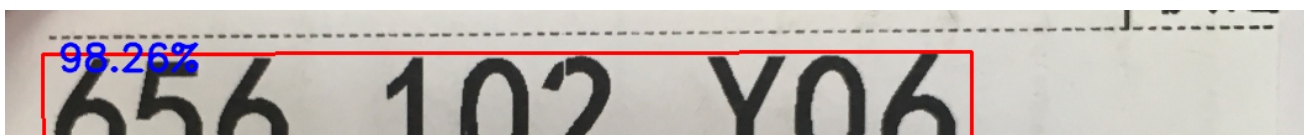
不同的初始化

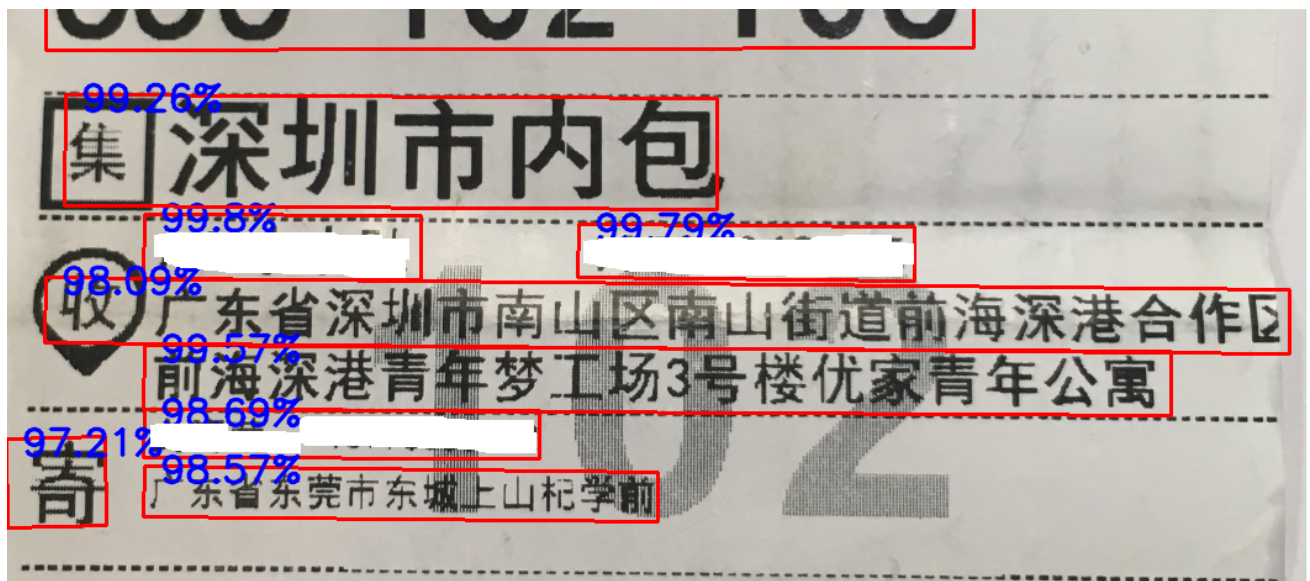
无论是convolution层还是RNN层先是采用默认的初始化方法,发现loss下降得很紊乱,所以还是使用了xavier_normal来初始化kernel,bias都初始化为0(最终发现,模型是否收敛,跟如何初始化关系很大).

使用keras CTPN模型权重初始化pytorch CTPN

这是我最终采用的方法,从keras的模型文件(训练好的hdf5文件格式)中一层层读取权重,然后再一层层的加载到pytorch的CTPN网络中,我先是直接使用这个权重进行检测,发现完全不行,然后基于该权重继续进行训练,第一个epoch loss 就达到了 **0.1440**, 第二个epoch到 **0.0730**,心中大喜.继续训练10轮之后,基本可以部署使用了.

最终是要做一个快递单据的OCR,这只是检测阶段,贴一个效果在下面.

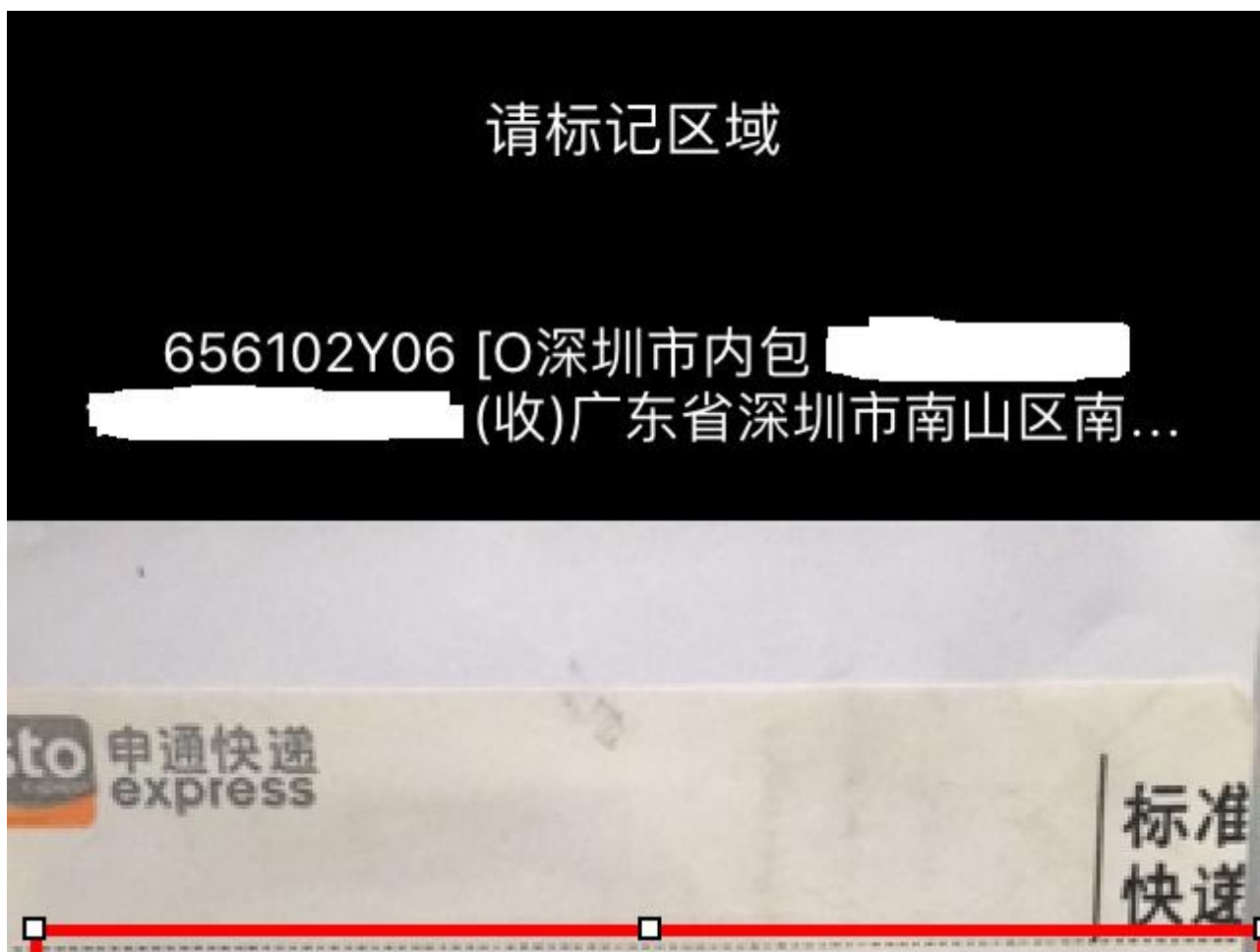




更新一下:图中的百分数,是text的分类概率

附

结合后面训练的CRNN用于识别检测出来的文字,做成一个移动端的OCR应用.对感兴趣区域进行识别.



656 102 Y06

集

深圳市内包

收

广东省深圳市南山区南山街道前海深港合作区
前海深港青年梦工场3号楼优家青年公寓

寄

广东省东莞市东城上山杞学前



3376372046052

2018/09/18

16:27:27

3/12

快件送达收件人地址。经收件人或收件人（寄件人）
允许的代收人签字，视为送达。您的签字代表您已经
签收此包裹。并已确认商品信息无误、包装完好，没
有损坏。如有疑问，请及时联系快递员。



重拍



完成

请标记区域

品名·A5缝线本货号:FA5400 规格210x148mm
页数:40页 简约·CITY REMEMBER，MEMOR...

deli得力

品名:A5缝线本 货号:FA54003

规格:210x148mm 页数:40页

简约·CITY

REMEMBER·MEMORY

得力集团有限公司

地址:浙江宁海得力工业园

传真:0574-65273660 [Http://www.nbdeli.com](http://www.nbdeli.com)

全国服务热线:400-185-0555 MADE IN CHINA

执行标准:QB/T1438 合格



重拍



完成