情感分析作业报告

韩志磊

2019-05-31

一、运行说明

1. 依赖

训练依赖：Keras >=2.0， Theano/Tensorflow 其一

评价依赖：Sklearn

2. 训练

运行 gen_data.py 完成预处理。然后修改 main.py 第 102 行，选择 cnn、rnn 或 mlp 之一，运行之。

默认 256 batchsize，耗费内存较大，可以自行修改

3. 评价

运行 evaluate.py，自动调取 model.h5 内的模型。在 model_pretrained 目录下有训练好的模型，可直接复制测试。

二、网络结构

本次实验中，实现了 CNN，RNN 和 MLP baseline 三种网络。其结构如图所示

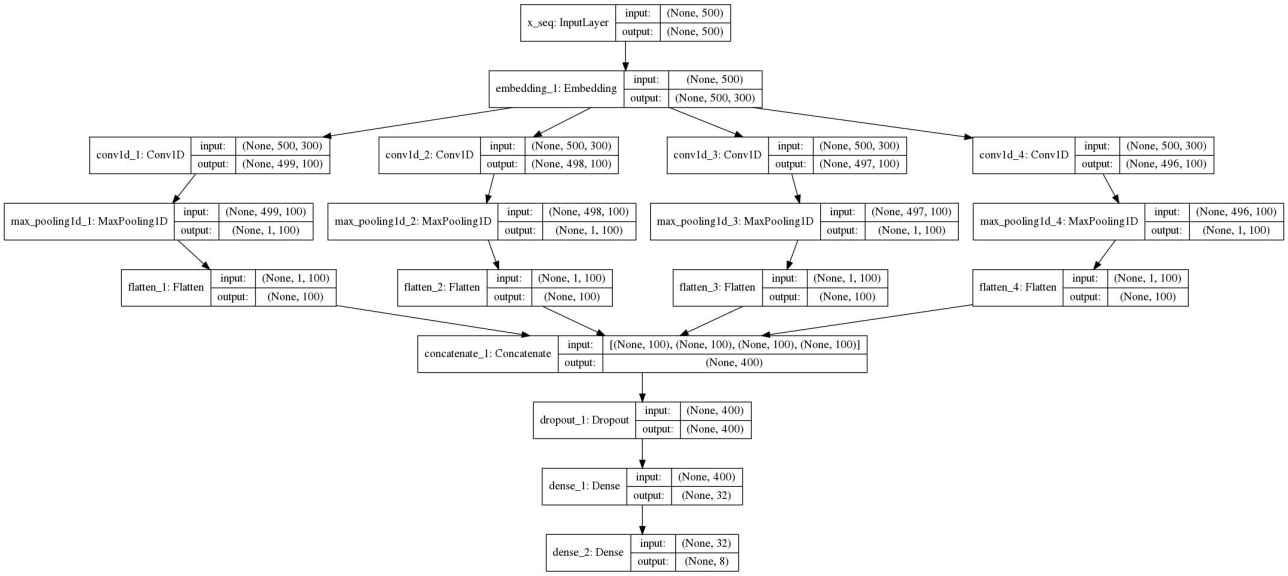

Fig 1. CNN 模型的结构

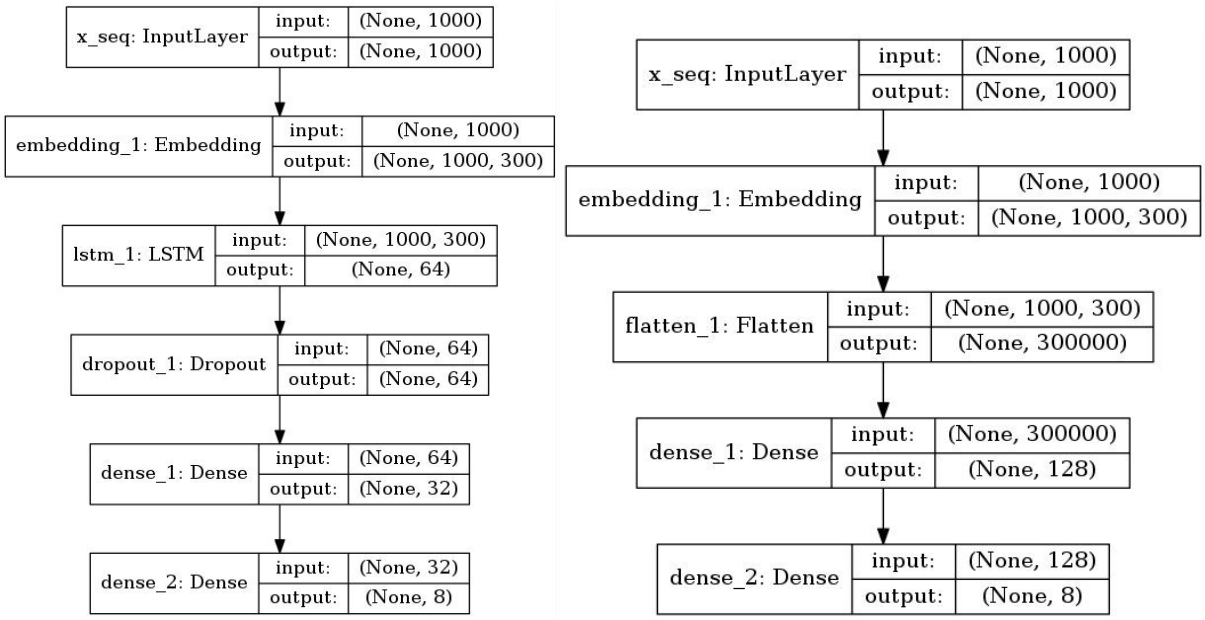

Fig 2. 左：RNN 模型的结构 右：MLP baseline 的结构

CNN 的设计，是基本按照参考论文进行的，将输入的 1000×300 词向量，分别通过高为 2,3,4,5，宽为 300 的卷积核，以及对应的 1-Max Pooling 层，分别得到一个特征值。为了提高效果，每一个卷积层通道数为 100。将总共 400 个特征值连接后，输入全连接层，最后 softmax 输出。基本的思想在于，相邻几个词之间的关系比较大，更有可能有一致的情感。而为了不破坏词的整体性质，选择卷积核的宽度和词向量的维度保持一致。

RNN 与 MLP baseline 基本一样，只是多了一层 LSTM 中间层，两者的输入规模也略有不同。从 RNN 的基本原理上看，此针对情感分析的模型，并没有什么实际的基本思想，更像是一种 MLP 的优化。

三、实验结果与分析

如下：

向量维度=300，最大词数=1000，学习速率=0.001

模型	batchsize	周期	Accuracy	F1-score(macro)	相关系数	Optimizer	Loss
CNN	256	5	58.98%	0.2919288	0.42360196	Adam	交叉熵
RNN	256	10	51.39%	0.1658630	0.3950389	RMSprop	交叉熵
MLP	256	5	47.75%	0.0808019	0.3983701	SGD	交叉熵

在实验前期，对参数做了较多调整。首先是向量维度和最大词数，基本规律是越大越好，但过大会极大地拖慢训练的速度，而且随着增长，对准确率的影响越来越小。最终选择了适中的 300 和 1000。

Batchsize 也是一样，较大的 batchsize 可以避免局部最优值的出现，但是它对内存有较大的要求。256 组数据已经足够避免“误入歧途”，因此选定 batchsize 为 256。

周期对模型的影响远远没有想象中那么大，基本上 3~4 个周期后，网络就已经趋于稳定。即使是 RNN，也是在前 5 个周期内上升最快，准确率与周期为 10 时差别不大。

Optimizer 初看觉得不起眼，但实际上大有用处。一开始时 CNN 的准确率只有 40% 出头，但将 SGD 改为 Adam 后，准确率便上升到了 58%。RNN 同样，RMSprop 是据称“特别适用于循环神经网络”的一种训练方式。而 SGD 对于全连接网络又有较好的效果，因此每一种结构都有对应较好地训练方式，至于这到底是为什么，可能更多的还是一种经验。

损失函数有两种备选：MSE 和交叉熵。起初我使用 MSE 作为损失函数，预测的是情感的分布。事实证明，MSE 很难收敛，并且效果奇差。而后我将其转为分类问题，使用交叉熵作为损失函数，得到了较好的结果。

最后，需要把学习速率调低，以减少过拟合的出现。

比较三种模型的效果，CNN 在各方面均最佳，RNN 次之，但与 baseline 差不太多。MLP 的训练时间极短，而 RNN 的训练时间大概在半个多小时（8 核），所以可以说是得不偿失。直觉上，加入词向量后，CNN 的确更适合情感分类的工作，因为在卷积核与输出之间存在比较明显的对应关系，而 RNN 则很难说清 LSTM 层的意义。

四、问题思考

1. 何时停止训练

我采用的是朴素的方法：先在一个较大的周期内训练，然后观察每个周期验证集损失、准确率的变化趋势，选择一个较小的周期，再进行第二次验证。实际上，由于输入数据的顺序随机，变量初始化也是随机的，所以每次训练的效果都不太一样。我这里选定的是平均来看较为稳定的一个周期数。

通过验证集来调整，需要额外的开销，这是肯定的，尤其是把整个测试集当作验证集的时候。而固定迭代次数，则难免会受随机误差的影响。所以两种方法都有各自的优缺点，在不缺时间的情况下，可以选择用验证集来调整。

2. 如何初始化

在模型中，LSTM 、卷积核、全连接层、词向量层的参数都是需要初始化的。经过测试，词向量层采用正态分布的随机变量较好，其他的则是均匀分布较好。至于正交初始化，其效果差别不大。这没有什么特别的理由，纯属经验。

3. 如何避免过拟合

提高 batchsize 是有效的从内部避免方法，但是不能完全避免过拟合。用测试集进行验证，在准确率开始连续下降时终止，这是一种从外部避免的方法。加 Dropout 也是一种办法，可以有效地避免过拟合。

4. 分析 CNN、RNN、MLP 的特点

前面已经谈过一些了。CNN 本来是因为处理图像而诞生的，它最主要的特点是可以高效地从区域中提取特征，尤其是结合 Pooling 时。借用词向量，可以拓展为文字处理的工具。LSTM 是模拟短时的记忆，可以处理变长的文章。MLP 是最基本的神经网络，也基本

适用于所有的场合，一般而言效果都适中，适合作为 baseline，softmax 激活的全连接层也是分类问题中必要的输出层。