

情感分类任务 实验报告

计 72 谢兴宇 2017011326

June 2019

目录

1	模型介绍	2
1.1	输入数据处理	2
1.1.1	预训练的词向量库	2
1.1.2	不存在的词	2
1.2	标签数据处理	2
1.3	CNN	3
1.3.1	基本结构	3
1.3.2	正则化	4
1.4	BiRNN	4
2	实验效果	4
3	超参数及模型结构的尝试和调整	6
3.1	初始化	6
3.2	训练方式	6
3.3	超参数	6
3.4	激励函数	6
3.5	损失函数	7
4	问题思考	7

1 模型介绍

情感分析是 NLP 领域的经典问题，本次实验需要解决的是情感分析领域中的情感分类 (sentiment classification) 问题。问题的输入是一篇已经做过分词的新闻，输出是对这篇新闻作情感归类（读者阅读之后的感受），将其归为以下八类之一：喜爱、讨厌、开心、悲伤、惊讶、愤怒。

为了解决情感分类问题，我实现了以下两个模型：

- CNN(Convolutional Neural Network)，参考并改进自 [1]。
- BiRNN(Bidirectional Recurrent Neural Network)，参考并改进自 [3]。

1.1 输入数据处理

首先我们将模型的输入转化为一个 $n \times k$ 的词向量矩阵，其第 i 行是第 i 个词的 k 维的词向量。

1.1.1 预训练的词向量库

在本次实验中，我们使用了由 [2] 预训练的中文词向量库，这些词向量（下载于<https://pan.baidu.com/s/1tUghuTno5y0v0x4LXA9-wg>）：

- 表示方式 (Representation) 为 SGNS(Skip-Gram with Negative Sampling)。
- 使用搜狗新闻 (Sogou News) 语料库 (Corpus) 训练。
- 上下文特征 (Context Features) 为 Word。

1.1.2 不存在的词

上述词向量库拥有 32 万左右的词，但依然会有不在词向量库中的词。对于这部分不在词向量库中的词，直接将其舍弃可能会影响到文章的连贯性，我们采取的策略是在词向量库中随机一个词向量作为它的词向量。

1.2 标签数据处理

这是一个多分类问题，不过提供的数据标签并非分类，而是已有用户的投票。一种非常自然的将投票转为分类的想法是：使用用户投票最多的那个

情感分类作为 ground truth。但这就会导致**多最大值问题**：有的新闻我们是无法确定分类的，因为它可能有多个分类都是投票最多的。

我们这里的解决方案是，先对数据（训练集和测试集）作预处理，筛除掉其中我们无法判断分类的数据。另一种可能的尝试是，在这种情况下我们可以在多最大值中随机指定一个分类。但我们并未作出后者的尝试，因为当多最大值问题出现时，说明这个数据的分类本身便是难以甄别的，即便是人类也无法做到，这样的数据用来学习和测试是不合适的。

1.3 CNN

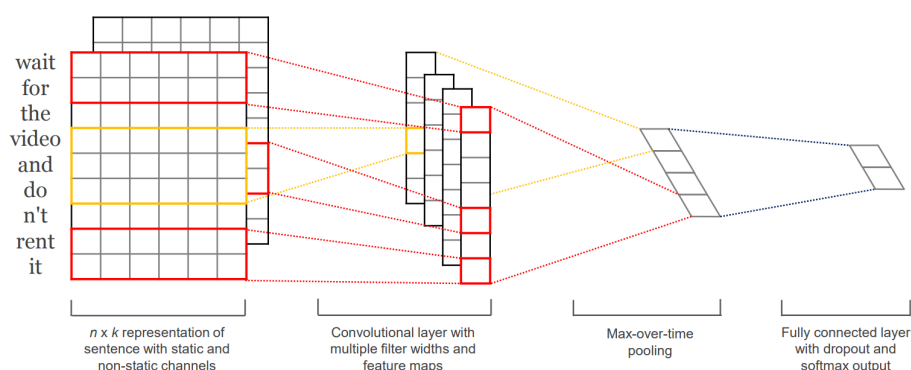


图 1: CNN 模型结构图

CNN 的结构大致如图1所示。

1.3.1 基本结构

在词向量矩阵之上，应用一个卷积层，其卷积核的大小为 $h \times k$ ，生成若干个 $n - h + 1$ 维的 feature map \mathbf{c} 。

对于每一个 feature map \mathbf{c} ，应用 max-over-time pooling，取其最大值 $\hat{c} = \max\{\mathbf{c}\}$ 。通过这种方式，我们便可以解决新闻长度不确定的问题。

将若干特征 \hat{c} 组成的倒数第二层传给一个全连接层，得到一个 8 维向量，再嵌套以一层 softmax 来保证其输出向量是一个概率分布向量。

与图中略有不同之处在于，我们并未采用动态（可学习）的词向量矩阵，而是单纯地使用了静态的词向量矩阵。

1.3.2 正则化

为了作正则化，我们采用了以下三样技术：

- 为倒数第二层的特征向量作 dropout。
- 将全连接层的矩阵向量的 l_2 范数固定为定值。
- 在训练每个 epoch 之前将所有训练数据的顺序随机打乱，再从前往后选择相邻的 50 条新闻数据组成一个 mini-batch 来训练。

1.4 BiRNN

BiRNN 的结构大致如图2所示。

我们将词向量矩阵输入给一个双向 LSTM，将 LSTM 在序列首尾元素的输出相连，输入进一个全连接层，便得到了一个 8 维向量，自然为了将之转为概率，在最末尾还需要一个 softmax 层。

可以看出，这个 BiRNN 模型的运算量是远大于 CNN 模型的，300 维的词向量对于 BiRNN 模型来说实在太过庞大了。为了优化计算速度，我们对于原始的词向量做了少许修改，对于每一个词向量只保留其前 30 维。

2 实验效果

Model	Accuracy	Macro F1 score	Pearson Correlation Coefficient
CNN	0.620445	0.310107	0.674281
BiRNN	0.494433	0.082712	0.564566

表 1: 模型评价指标比较

实验效果如表1所示。在计算准确率 (Accuracy) 时，方便起见，我们忽略了模型输出的多最大值问题。为了计算相关系数 (Pearson Correlation Coefficient)，我们重新处理了（无最大值问题的）标签数据，将八个情感分类的用户投票数作为一个 8 维向量输入进一层 softmax，之后得到一个可以表示概率分布的向量，以此作为标注值与预测值计算 Pearson 系数，将其绝对值的平均值作为总体的 Pearson 系数（因为 Pearson 系数可能有正负，而其意义由与 1 的接近程度来决定，故这里取了平均值）。

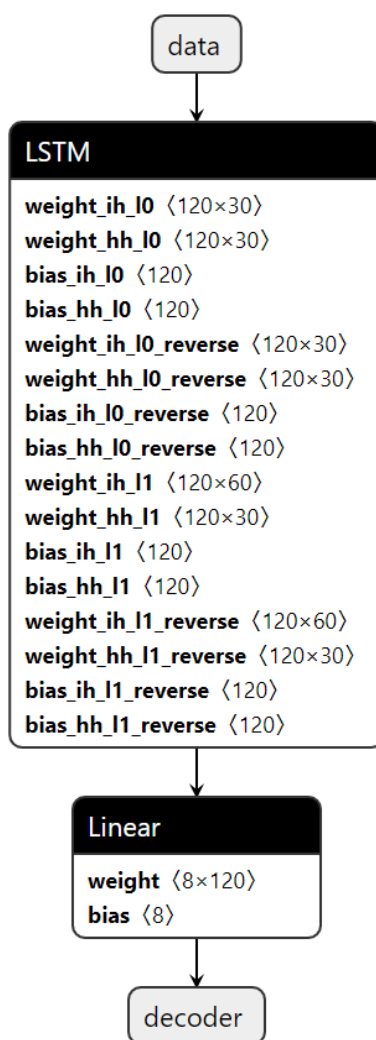


图 2: BiRNN 模型结构图

由于numpy的计算能力限制，在计算 softmax 时可能会得到NaN，为此我们不得不舍弃一部分数据，所幸这只是极少数的数据，并不影响大局。

3 超参数及模型结构的尝试和调整

3.1 初始化

对于 CNN 模型，我们使用了PyTorch的默认初始化方式。

在 BiRNN 模型上，我们对于初始化做了一些尝试。我们使用了 Xavier Initialization 技术初始化全连接层的权重矩阵 \mathbf{w} ，将偏移向量 \mathbf{b} 中的元素均初始化为 0.01。

3.2 训练方式

对于 CNN 模型，我们尝试了 Adadelta 来训练；对于 BiRNN 模型，我们尝试使用 Adam 来训练。

3.3 超参数

经过实验，我们选择了以下超参数：

- CNN 模型：卷积核的窗口高度 $h = 3/4/5$ ，每个生成 100 张 feature maps。
- CNN 模型：drop out 速率 $p = 0.5$ 。
- CNN 模型：mini batch 大小设为 50。
- CNN 模型：将全连接层权重向量的 l_2 范数固定为 3。
- RNN 模型：Adam 的 learning rate 设为 0.01。
- RNN 模型：LSTM 的num_layers设为 2。
- RNN 模型：mini batch 大小设为 50。

3.4 激励函数

在 CNN 模型中，所有的激励函数均使用了 ReLU；RNN 模型没有用到激励函数。

3.5 损失函数

最初我们使用 MSE 作为损失函数，准确率并不高（在 0.1-0.2 左右），之后换成了 Cross Entropy，有了显著提高，只经过很少的几个 epoch 训练，便能达到 0.55 及以上的准确率。

4 问题思考

- 训练时什么时候停止是最合适的？

我是固定训练次数来训练的。

- 实验参数的初始化是怎么做的？不同方法适合哪些地方？

我们使用了 Xavier Initialization 技术初始化全连接层的权重矩阵 \mathbf{w} ，并将偏移向量 \mathbf{b} 中的元素均初始化为 0.01。

- 过拟合是深度学习常见的问题，有什么方法可以防止训练过程陷入过拟合？

我在本次实验中采用的 Dropout、限制权重向量的范数、增大 batch size 都可以防止过拟合。

- 试分析 CNN, RNN, 全连接神经网络 (MLP) 三者的优缺点。

在 NLP 问题中，由于句子长度不确定，并不适合 MLP。MLP 往往只能与 Bags-of-Word-Vectors 来联合使用，不能很好地利用上下文的信息。

RNN 利用历史信息的想法比较自然，但最终结果并不是特别好，并且由于计算量十分巨大，训练和测试都十分耗费时间和资源。

CNN 的想法乍看并不自然，max-over-time pooling 是比较巧妙的操作，最终得到了最好的效果。

参考文献

- [1] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
- [2] Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. Analogical reasoning on chinese morphological and semantic relations. *arXiv preprint arXiv:1805.06504*, 2018.
- [3] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. 2019. <http://www.d2l.ai>.