基于卷积神经网络的文本情感分类

付梦锦

北京科技大学 计算机与通信学院

摘 要 我们小组研究了基于卷积神经网络（CNN）的文本情感分类问题。首先，我们了解、对比和分析了多种不同的解决方案的原理和优缺点，并决定采用基于深度学习方法的 CNN 模型。然后，基于开源的文本分类数据集和相关开源项目，以 tensorflow 为框架，训练了基于 CNN 的文本情感分类器，实现了将英文句子分为正面情感和负面情感的功能。此外，还对模型超参数进行了调整，使模型的训练时间缩短为原来的 15%，且准确率提升了 2%。另外，我们还手动收集和处理了测试数据集，以测试模型的泛化性能。

关键词 深度学习 卷积神经网络 文本分类 情感分析

1 引言

互联网给人们提供了宽广平台上来发布交流自己观点和评价，这些言论往往包含有丰富的个人情感，比如对某部大片的影评，对某款商品的用户体验等，其中蕴含着巨大的商业价值。如何从这些文本中分析和获取情感的倾向，已经成为当今商务智能领域关注的热点。利用计算机，利用算法自动对评论进行分析挖掘，是解决这个问题的最有效途径。情感分析(sentiment analysis)的任务包括[1]：识别文本的情感极性（如正面、负面或者中性）和识别文本的情感强度，前者也称为文本情感分类。

针对文本情感分类的问题，我们小组首先了解、对比和分析了传统方法，机器学习与特征工程方法以及基于 CNN、RNN 模型的深度学习的原理和优缺点，决定采用深度学习方法的 CNN 模型。然后，基于开源的电影评论英文文本数据集[2]和相关开源项目[3]，以 tensorflow 为框架，训练了基于 CNN 的文本情感分类器，实现了将英文句子分为正面情感和负面情感的功能。此外，还尝试对模型的超参数进行调整，使模型的训练时间缩短为原来的 15%，且准确率提升了 2.5%。另外，我们还手动收集和处理了测试数据集，以测试模型的泛化性能。

2 相关工作

通过查阅相关的资料，我们了解了文本分类问题解决方案的逐步演进，分析了各种方案的大致原理，对比了各种方案的优缺点。

传统方法基于情感词典来进行文本的情感分类，对人的记忆和判断思维的进行最简单的模拟。首先通过学习来记忆一些基本词汇，如积极词语有“happy”，消极词语有“hate”等，从而获得一个基本的语料库。然后，我们再对输入的句子进行最直接的拆分，查看所记忆的词汇表中是否存在相应的词语，然后根据这个词语的类别来判断情感，比如“I feel so happy!”，“happy”这个词在所记忆的积极词汇表中，所以判断它具有积极的情感。传统方法显然具有一些明显的缺陷，如其采用线性模型准确率低，可扩展性很差。

基于机器学习算法和特征提取的方法在过去的很长一段时间内都是文本情感分类的主流算法。其文本表示通常使用词袋模型，再经由特征提取后，模型可以自动从数据中学习出一个复杂的高维分类模型实现情感分析。机器学习方法在可扩展性和适应性方面，相对于传统方法有着质的飞跃，但也有一定的局限性，其分类的效果主要取决于提取的特征是否能足够很好的区别正面和负面情感。而特征的提取非常依赖于人的先验知识，即需要我们

对数据进行足够深入的观察和分析，把那些对区分正负面情感最有用的特征找出来。且词袋模型会丢失上下文信息，还会导致向量空间特别大，一般都是数十万维。对于短的文本评论，转换成的向量特别稀疏，会造成模型的不稳定性。

近几年，深度学习成为解决文本情感分类的一个很好地方法。相比于传统机器学习方法，其优势包括：（1）可以自动从数据中学习出特征和模型参数，省去了大量繁杂的特征工程工作，对行业先验知识的依赖也降低到最小程度。（2）深度学习在处理文本数据的时候，往往是先把词语转成词向量再进行计算，词向量的生成考虑了一个词语的语义上下文信息，也就解决了词袋模型的局限性。（3）由于使用了词向量，特征维度大幅减少，可以降低到百的量级，同时也使得文本向量变得“稠密”，模型变得更加稳定。

目前，在自然语言处理（NLP）上，深度学习主要包括 CNN 和 RNN(递归神经网络)两大阵营，基于卷积的 CNN 对识别目标任务的结构具有一定的优势，而 RNN 由于其记忆功能对序列识别建模具备优势[4]。针对文本的情感分类问题，带有显著情感极性的词组会对结果有比较关键的影响，由于 CNN 通过卷积可以很好地提取局部特征和关键信息，所以采用 CNN 模型可以取得很好的结果。RNN 模型对于句子或文本作为序列输入更加自然，可以利用到历史信息，将词的顺序也考虑进去[5]。但是，CNN 模型相对于 RNN 来说要简单，训练速度更快，采用 RNN 衍生的 LSTM 模型可以也可以做到不错的效果但是，速度上会比 CNN 慢很多。

综合以上的分析和比较，我们最终决定采用 CNN 模型。

3 模型

基于 CNN 文本情感分类模型结构大致如图 3-1 所示。

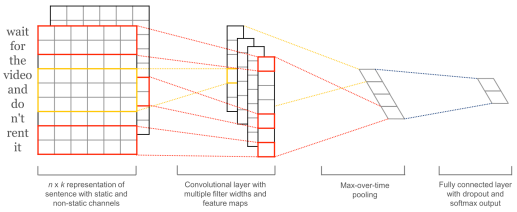

图 3-1[8] 句子分类的 CNN 模型结构图

第一层是输入层，输入以矩阵形式表示的句子或者文本，其中每一行都是一个词向量，代表着一个单词。词向量可以由是 word2vec 或 GloVe 这样的词嵌入（word embedding），也可以用 one-hot 向量表示，将单词转换成词汇表[6]。我们采用的是 128 维的词向量表示。为使问题简化，我们只用一个 channel，并且不使用 word2vec 预训练的词向量初始化词嵌入，而是使用训练集从头开始学习。

第二层是卷积层。在 NLP 中，通常使用与输入矩阵等宽的 filters 进行卷积，但是一般一次只滑过 2-5 个字以上的窗口。每个 filter 可以提取一个特征，我们采用的是尺寸分别为 3,4,5 的 filters，每种 128 个，来提取多个特征。另外，还有一个细节在于输入矩阵边缘的处理。我们使用零填充（对于我们的数据集，每个句子都填充到长度为 59），以使 filter 可以提取到于输入矩阵的每个元素的信息，并获得更大或相同大小的输出。因为卷积是线性运算，所以之后要经过非线性激励，我们采用的是二元文本分类问题最常采用的 Relu 函数。

第三层是池化层。池化可以向分类器提供固定大小的输出，同时降低维度并尽可能保留最重要信息[7]。我们采用的是最大池化，得到一个长的特征向量。

最后将特征结果传给全连接的 softmax 层，输出分类结果。并加上一层 dropout，以防止过拟合。

4 实验部分

4.1 模型的实现

使用 tensorflow 框架，Python 语言。

在 TextCNN 类的 init 函数中生成模型图。我们定义的第一层为嵌入层（embedding layer），其将词汇索引映射为低维向量表示，本质上是从数据中学习的一个可供查找的表。随机初始化模型参数 W。然后是卷积层和最大池化层。因为每个卷积产生不同形状的张量（tensor），我们需要对其进行最大池化，得到特征向量，filter 的数量对应着提取的特征数。当我们从中获得了所有的输出张量，就将它们组合成一个长的特征向量。Dropout 是防止卷积神经网络产生过拟合的最流行的方法。Dropout 层随机地“禁用”部分神经元，这可以迫使神经元学习不同的有用特征。在训练期间，我们设置 dropout 值为 0.5，在测试期间禁用。使用 max-pooling 层输出的特征向量，生成预测，并挑选具有最高 score

的类。还可以应用 softmax 函数将原始分数转换为归一化概率，但这不会影响我们的最终预测的结果。最后，定义损失函数和准确率。分类问题的标准损失函数是交叉熵损失函数（cross-entropy loss）[9]，我们使用损失函数平均值，便于比较。

整个模型使用 tensorboard 可视化的结构示意图如图 4-1 所示。

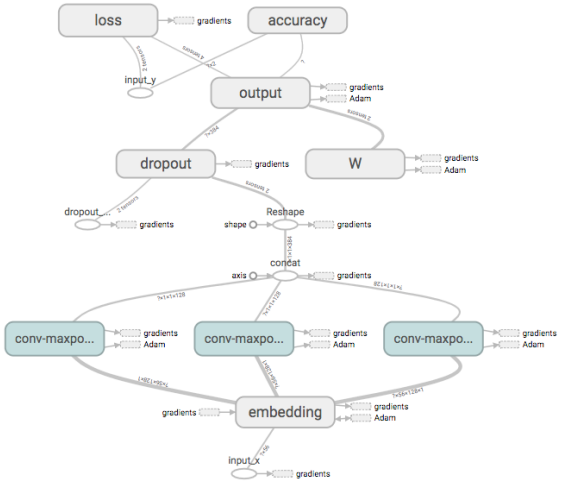

图 4-1 CNN 模型实现示意图

4.2 训练过程

首先是数据的预处理。我们使用的数据集是 Rotten Tomatoes 的 Movie Review 数据，该数据集包含 10,662 个电影评论句子，正负面情感各占一半。由于这个数据集很小，我们使用复杂的模型去学习，很容易产生过拟合。此外，我们将 10%的数据用作验证集。数据预处理主要包括：数据加载；数据清洗；将每个句子扩充至最大长度；构建数据索引。

然后开始模型的训练。实例化 CNN 模型，设置好默认参数，并随机初始化所有变量，采用 Adam Optimizer 进行优化。将训练过程中损失函数和准确率的变化写入 summaries，将模型参数的变化写入 checkpoint。

4.3 超参数调整

默认参数下，模型的训练时间较长（约 2h），模型的准确很早就已经趋于稳定，且出现了比较严重的过拟合，训练后期验证集的 loss 一直在增大。表 4-1 列出了一些关键的默认参数。

表 4-1 一些关键的默认参数

参数	embedding_dim	filter_sizes	num_filters
	128	64	128
参数	dropout_keep_prob	batch_size	num_epochs
	0.5	64	200

表 4-2 显示了调整超参数 epoch 和加入 L2 正则化项对训练结果的影响。

表 4-2 超参数的调整

	num_epochs	L2	训练时间/s	准确率
默认	200	0	7200	72.3
1	200	0.3	7200	71.6
2	200	3	7200	71.4
3	50	0	1900	71.9
4	30	0	1200	72.2
5	30	3	1200	74.3
6	30	10	1200	**75.0**
7	20	0	780	73.3
8	20	3	780	74.6
9	20	10	780	74.3

分析结果可知，将 epoch 大幅减小后，对模型的准确率并没有明显的影响，而训练时间则可以大大缩短，如 epoch 减少至 30 后，训练时间仅为约原来的 15%。而在减少 epoch 之后，再加入 L2 正则化项进一步防止过拟合的发生，则可以使模型的准确率得到提高，模型的最高准确率达到 75.0%。

4.4 训练结果测试

最后，采用测试集进行模型泛化性能的测试。为此，从网络上收集并手动整理了测试语料库，包含总共 4280 个英文语句，内容主要包括对电影和商品的评价，正面和负面情绪各约一半。图 4-2 展示了测试集的部分内容。

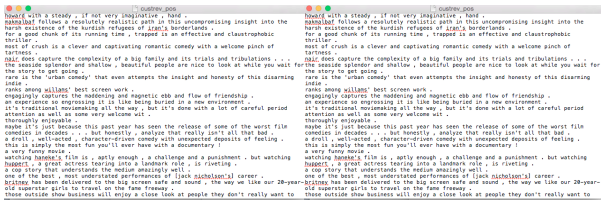

图 4-2 测试集展示

对训练结果准确率最高的模型进行测试后，测试准确率为 71.9%，这表明训练后得到的模型的泛化性能和迁移性都还取得了不错的效果。

5 结论

通过学习和研究文本情感分类问题，我们了解到了文本情感分类方法的演进，了解了最新也最热门的几种深度学习方法的基本原理和区别。在此基础之上，我们重点研究了基于 CNN 模型的文本情感分类分类问题，采用 CNN 实现了一个对英文句子进行情感二分类的分类器。在这个过程中，我们深入理解了 CNN 的结构，实现和构建 CNN 模型的方法，以及 CNN 训练过程，并尝试对部分超参数进行优化。调整超参数后的模型，训练时间大大缩短，且准确率最高提升了约 2.5%。最后，还尝试自己收集并整理了测试数据集，对训练好的模型进行泛化性和迁移性的测试，并得到了较好的测试结果。

参 考 文 献

[1] 何炎祥, 孙松涛, 牛菲菲,等. 用于微博情感分析的一种情感语义增强的深度学习模型[J]. 计算机学报, 2017, 40(4):773-790.

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[3] https://github.com/dennybritz/cnn-text-classification-tf/

[4] Yin W, Kann K, Yu M, et al. Comparative Study of CNN and RNN for Natural Language Processing[J]. arXiv preprint arXiv:1702.01923, 2017.

[5] https://zhuanlan.zhihu.com/p/25928551

[6] Johnson R, Zhang T. Semi-supervised convolutional neural networks for text categorization via region embedding[C]//Advances in neural information processing systems. 2015: 919-927.

[7] Zhang Y, Wallace B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1510.03820, 2015.

[8] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.

[9]http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/