人工智能导论
情感分析实验报告

2017011620 计 73 李家昊

2019 年 5 月 31 日

1 数据处理

1.1 文本处理

给定的数据集已经完成了分词，因此这里使用 keras.preprocessing.text 中的 Tokenizer 工具，统计文本中出现的所有词，按照词频从高到低，为每个词建立一个整数索引，然后将文本转换为整数序列。考虑到 CNN 模型要求各数据大小一致，这里将每篇文章的长度限制在 600 个词，对于过长的文章，截取其前 600 个词，对于过短的文章，则在其后补 0，得到文本对应的整数序列，作为神经网络的输入。

1.2 文本表示方法

这里采用词向量（word embedding）的方式表示文本。考虑到给定的数据集较小，不足以训练出较好的词向量，因此我参考了说明文档提供的预训练词向量下载地址，下载了基于搜狗新闻数据集训练的词向量（300 维），将其用到了本次实验的 embedding layer 中，并将其权值标记为 non-trainable。

1.3 标签表示方法

这里采用分类问题的表示方法，将用户打分最高的类别作为整篇新闻的情感类别。

1.4 训练集、验证集、测试集的说明

在给定的训练集中，取 16% 作为验证集（共 375 个样本），其余作为训练集（共 1967 个样本），给定的测试集（共 2228 个样本）仅作为最终测试使用，不参与任何训练过程，不发挥任何验证作用。

2 模型结构

2.1 CNN

这里参考了说明文档提供的 Text CNN 论文，构建 Text CNN 模型，但模型结构与论文中的略有不同，如下图

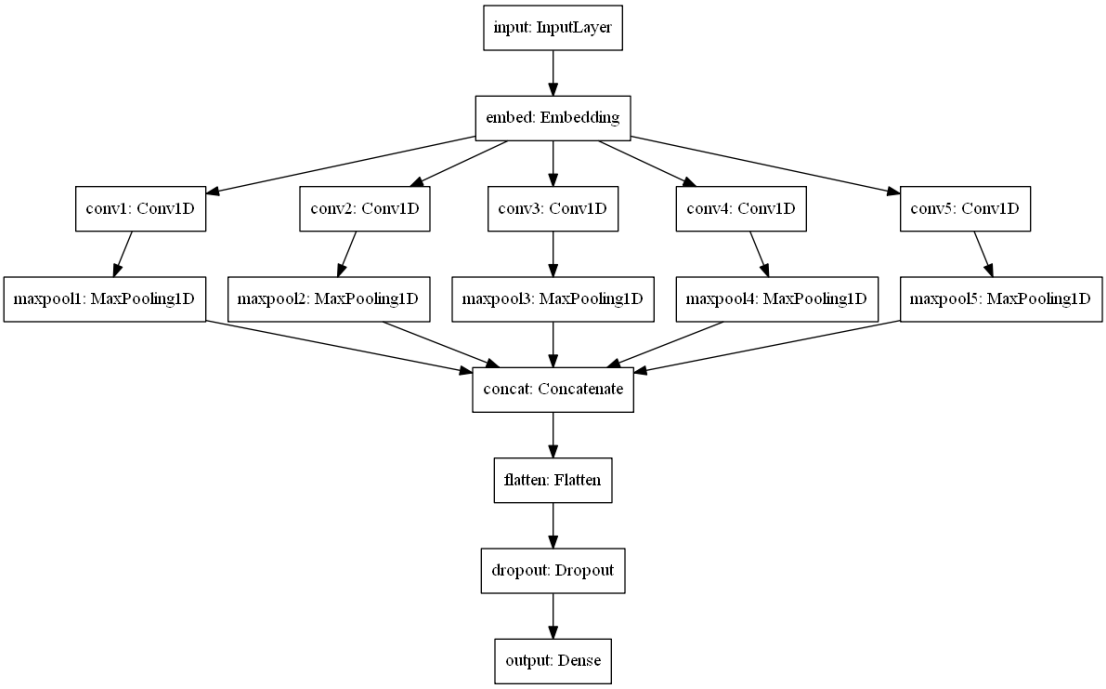

Figure 1: Text CNN Architecture

模型接受一个输入序列，先通过一个 embedding layer，将输入序列转换为预训练好的词向量，然后分别用大小为 $n = 1, 2, 3, 4, 5$ 的卷积核对词向量做一维卷积，对应基于词的 $n = 1, 2, 3, 4, 5$ 元模型，每种大小的卷积核数量为 128 个，然后通过 Max Pooling 层，取出卷积的最大值，然后将每个通道内的五个 pooling 结果连接起来，形成 feature map，将其 flatten 后，以 0.5 的概率进行 dropout，然后用一层全连接层将其连接到输出，采用 softmax 函数计算出每个类别的预测概率。

2.2 RNN

2.2.1 LSTM

构建一个简单单向 LSTM 模型，结构如下图

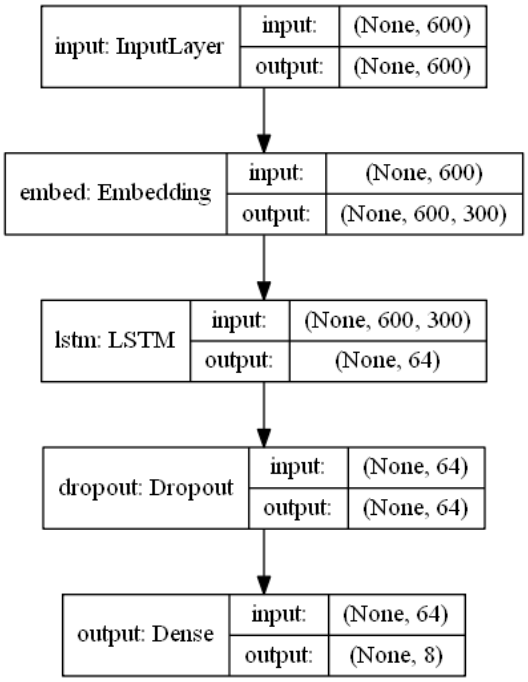

Figure 2: LSTM Architecture

模型接受一个输入序列，先通过一个 embedding layer，将输入序列转换为预训练好的词向量，然后通过一层有 64 个单元的 LSTM 层，再以 0.5 的概率进行 dropout，最后用一层全连接层将其连接到输出，采用 softmax 函数计算出每个类别的预测概率。

2.2.2 Bidirectional LSTM

构建一个双向 LSTM 模型，结构如下图

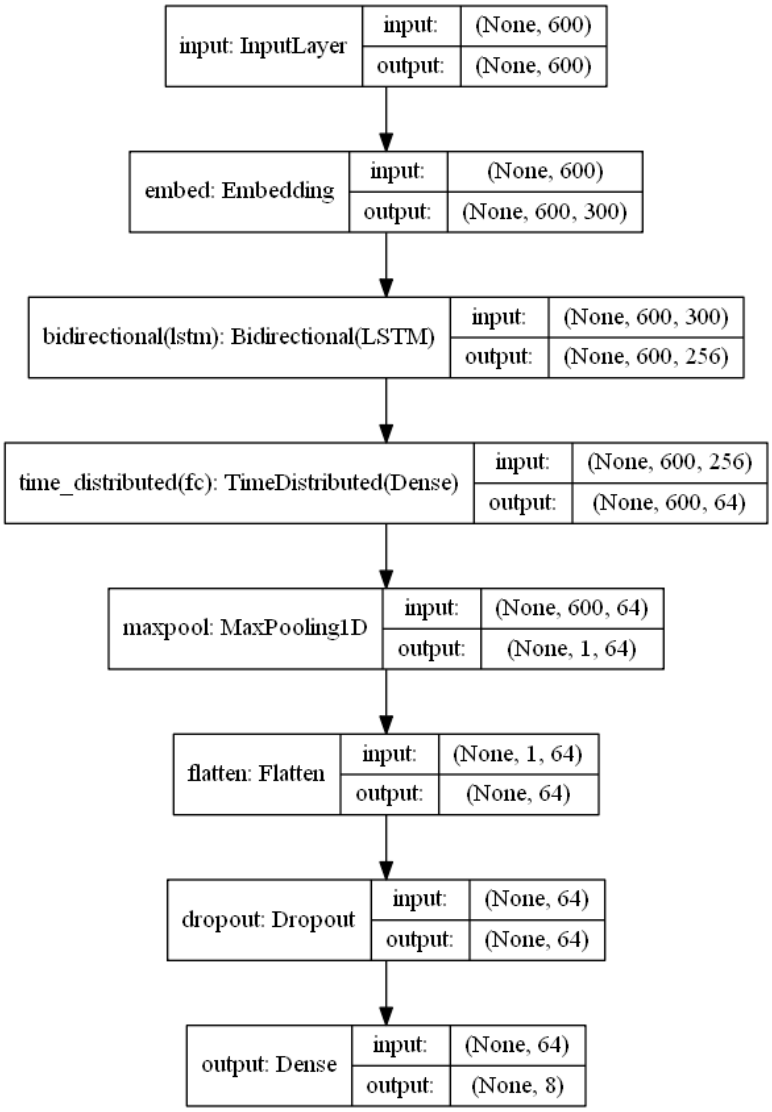

Figure 3: Bidirectional LSTM Architecture

模型接受一个输入序列，先通过一个 embedding layer，将输入序列转换为预训练好的词向量，然后通过一层有 128 个单元的双向 LSTM 层（正向和逆向各 64 个单元），返回一个时间序列，接下来通过一个 time distributed dense 层，在时间维度上进行全连接，再通过 maxpooling 层得到每个通道的最大值，然后 flatten 降维，以 0.5 的概率进行 dropout，最后用一层全连接层将其连接到输出，采用 softmax 函数计算出每个类别的预测概率。

2.3 MLP (Baseline)

构建多层感知机（MLP）作为 baseline，网络结构如下

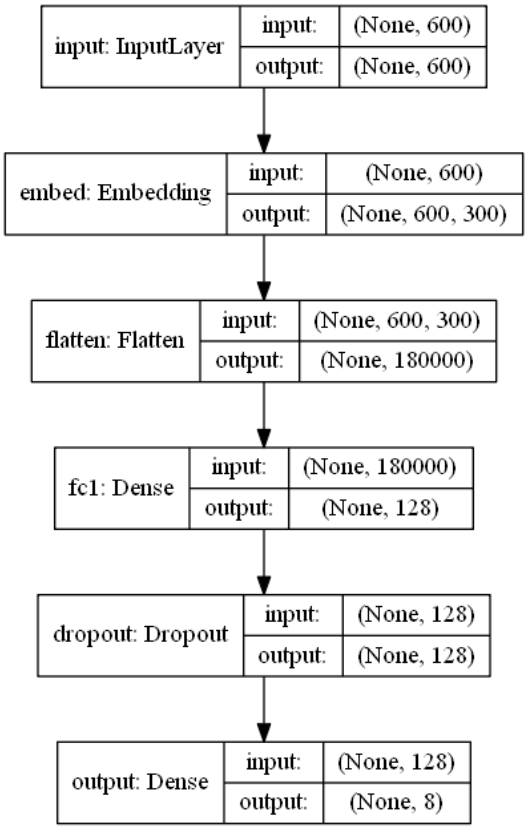

Figure 4: MLP Architecture

输入序列首先经过 embedding 层，转化为词向量，然后将其 Flatten 降维，经过一个全连接层，再以 0.5 的概率进行 dropout，最后用一层全连接层将其连接到输出，采用 softmax 函数计算出每个类别的预测概率。

3 实验结果

训练模型时，采用 Adam 优化器，取 learning rate 为 0.001，损失函数取为交叉熵。取 batch size 为 256，训练 100 个 epochs。训练过程中，总是保存验证集上 loss 最小的一个模型，用于最后的测试。

测试模型时，先加载上述方式保存的模型，然后在测试集上测试，最终测试结果如下表所示

Model	Accuracy	F1-Score(Macro)	Coef.
Text CNN	62.97%	27.03%	61.61%
LSTM	57.09%	16.62%	52.81%
Bidirectional LSTM	**63.42%**	**31.07%**	**62.17%**
MLP	55.57%	17.86%	50.56%

Table 1: Results of implemented models

特别说明：经统计，在测试集中有 234 条数据有多个最大标签，占总标签数的 10.5%。因此，在计算上表中的准确率时，只要模型预测出来的类别为最大标签之一时，即判定为预测正确。

此外，按照实验要求，F1-Score 需要用 Macro Average 计算，但是由于数据集的缺陷，某些类别从未被模型预测过，其 F1-Score 被置为 0，拉低了总体的 F1-Score。因此，这里的 F1-Score 不具有参考意义。

4 调整参数

本次实验对 Text CNN 有所创新，受到 GoogLeNet 中 Inception module 的 1×1 卷积核的启发，我也将大小为 1 的一维卷积核用到了 Text CNN 中，对应于词的一元模型。这么做的原因是，某些词的情感色彩非常明显，只要这些词出现在文章内，基本就可以确定这篇文章的情感类别。

下面进行对照实验，实现 Baseline 为包含长度为 $n = 2, 3, 4, 5$ 的卷积核的 Text CNN，与包含长度为 $n = 1, 2, 3, 4, 5$ 的卷积核的 Text CNN 做对比，在测试集上的测试结果如下

Model	Accuracy	F1-Score(Macro)	Coef.
CNN(with 1-conv)	**62.97%**	**27.03%**	**61.61%**
CNN(baseline)	62.06%	25.97%	60.21%

Table 2: Results of 1-conv CNN against baseline

可见，增加了大小为 1 的一维卷积核后，准确率上升了约 0.9%，F1-Score 与相关系数均有相应提升。

5 问题思考

5.1 停止训练的时机

我的做法：在训练集上以 16% 的比例划分出验证集，训练过程中，总是保存验证集上 loss 最小的一个模型。然后使模型充分训练，当看到 loss 明显回升，且不会下降到更低点时，停止训练。然后取保存下来的模型进行测试。

固定迭代次数的方式：优点是方便实现，缺点是不太灵活，需要针对不同的模型选择不同的迭代次数，不能避免过拟合现象，无法预测测试集上的准确率。

通过验证集调整的方式：优点是容易观察到过拟合现象，以及模型的泛化能力，缺点是需要额外消耗计算资源，延长训练时间。

5.2 参数初始化

本实验中参数初始化方式是均匀分布初始化（uniform initialization）。初始化参数只要不是太大，一般来说对训练结果影响不大，最终都能收敛到合适的值。

零均值初始化能防止梯度爆炸问题，若初始值均值不为 0，则可能产生梯度爆炸问题，导致模型无法训练。

高斯分布初始化能防止梯度消失问题，高斯分布初始化的权重集中在 0 点附近，多次经过 sigmoid 激活函数后，仍然能保持相应梯度，防止梯度消失。

正交初始化主要用在 RNN 的初始化，避免梯度爆炸和梯度消失的问题。

5.3 防止过拟合的方式

- 增加 Dropout Layer。
- 增加 L1/L2 Regularization Layer。
- 增加 Batch Normalization Layer。

5.4 CNN, RNN, MLP 优缺点分析

5.4.1 CNN

优点：训练速度快，参数数量少，不容易产生过拟合现象，考虑了上下文信息，大小为 n 的一维卷积核对应了基于词的 n 元模型。

缺点：需要预先确定矩阵的大小，训练期间不能发生变化，因此只能通过截长补短固定文本长度，但可能使长文本的重要信息丢失，无法达到训练效果。

5.4.2 RNN

优点：参数数量少，不容易产生过拟合现象，考虑了上下文信息，通过输入门、遗忘门、输出门完成对上下文信息的处理，不需要固定文本长度。

缺点：训练速度慢，实现复杂。

5.4.3 MLP

优点：实现简单，训练速度快。

缺点：直接将文本的词向量矩阵降维，会丢失上下文的信息，可能无法达到训练效果。此外，其参数数量较多，容易产生过拟合现象。

6 对数据集的分析与建议

本次实验的数据集在数量上和质量上都不尽人意，因此，我想对本实验的数据集提出如下建议：

- 扩大数据集的规模

  本次实验中，真正能用到训练中的样本仅为 2000 个左右，训练数据严重不足，非常容易出现过拟合现象。建议将训练集规模增大到 10000 个样本以上。

- 提高数据集的标注质量

  我猜想这个数据集是直接从新浪网上用爬虫爬下来的，新浪网上选择情感分类的都是普通网友，并非专门的标注人士，网友的选择随机性很大，而且有些文章仅有几个网友投票，不能代表整篇文章的情感分类。

- 使类别分配更加平均

  经过统计得出，训练集和测试集上各标签的数量如下图所示

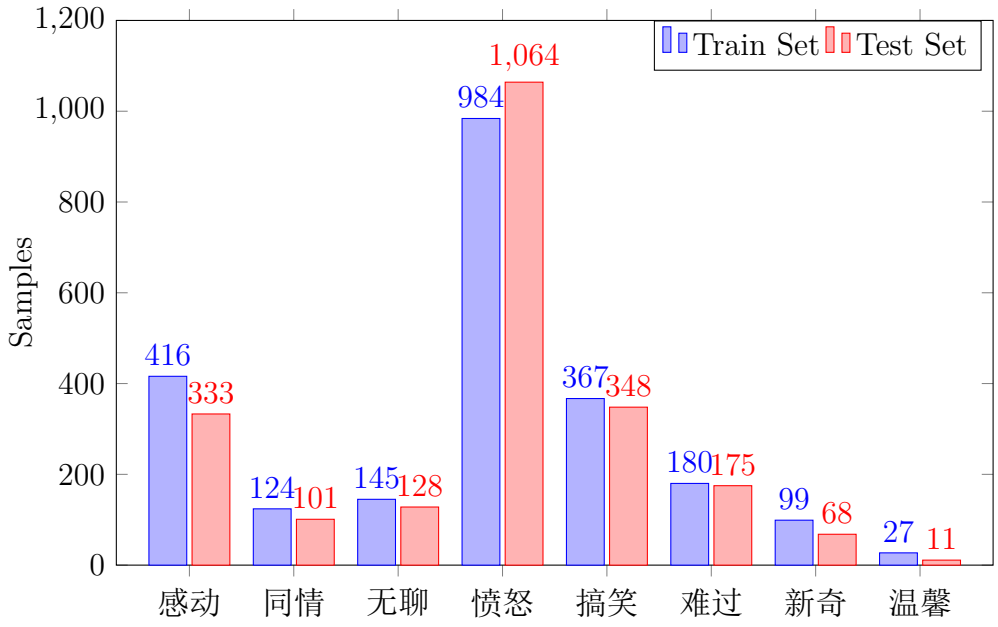

Figure 5: Summary statistics of train set and test set

其中“愤怒”标签数占总标签数的 44.81%，而“温馨”只占所有标签的 0.83%，标签分布极不均衡，导致某些类别从未被模型预测过，拉低了总体的 F1-Score。

若能提升数据集的质量，将进一步提升训练效果，使训练结果更有说服力。

7 实验总结

1. 通过本次实验，我实现了 MLP, Text CNN, LSTM 等文本处理模型，实现了文本的多分类任务，对神经网络的工作机制有了更深的理解，对 F1-Score, Correlation Coefficient 等评价指标更加熟悉。

2. 通过调参实验培养了耐心和毅力，同时领悟了不少调参的经验，例如事先预估可训练参数的数量、通过 dropout 防止过拟合、通过调整 learning rate 实现精细调整等等，这些经验有效地提升了模型效果，其中双向 LSTM 效果最佳，准确率达到了 63.42%。

3. 感谢助教的耐心指导！