**数智NLP组2022第二阶段考核**

**信息工程学院**

**电子信息类**

**2021级9班**

**3121002477**

**黄维爽**

**2021年4月**

**摘要**：在这个文档里，我将要简要概述第二阶段的考核任务完成情况并进行汇报和总结，任务为使用深度学习框架tensorflow构建神经网络实现自然语言处理（NLP）中的文本分类，包含对yelp数据集进行必要的文本预处理（Text pre\_processing），将处理好的词向量送入神经网络进行训练，通过微调可以进一步提高性能寻找最优参数和最优模型，最后是对模型训练结果进行可视化操作。

关键词：文本分类，神经网络

**Abstract**: In this paper, I will give a brief overview of the completion of the assessment tasks in the second stage and report and summarize them. The task is to use the deep learning framework tensorflow to carry out text classification, including necessary Text pre\_processing for the yelp dataset, The processed word vector is sent to the neural network for training, and the performance can be further improved by fine-tuning to find the optimal parameters and optimal model. The last step is to visualize the model training results.

**Keywords**: NLP, Text Classification，Neural Networks

1. 简介

近年来，深度学习模型在计算机视觉和自然语言处理方面取得了显著成果。自然语言处理（Natural Language Processing，简称NLP）就是用计算机来处理、理解以及运用人类语言(如中文、英文等)， 从研究内容来看，自然语言处理包括语法分析、语义分析、篇章理解等。从应用角度来看，自然语言处理具有广泛的应用前景。特别是在信息时代，自然语言处理的应用包罗万象，例如：机器翻译、手写体和印刷体字符识别、语音识别及文语转换、信息检索、信息抽取与过滤、文本分类与聚类、舆情分析和观点挖掘等，它涉及与语言处理相关的数据挖掘、机器学习、知识获取、知识工程、人工智能研究和与语言计算相关的语言学研究等。

卷积神经网络 (CNN) 利用具有卷积滤波器的层，这些滤波器应用于局部特征CNN 模型最初是为计算机视觉而发明的，后来被证明对 NLP 有效，并在语义解析、信息检索、句子建模和其他传统的 NLP 任务。在目前的工作中，把数据集的文本进行了分词，去除停用词，词干提取，padding等预处理操作后，训练了一个简单的卷积神经网络模型， 把词向量喂入模型训练，预测得到概率。

1. **任务实现流程**
   1. **文本预处理**
      1. 打乱数据样本

默认一定要增加随机采样因素，尽可能使数据分布随机，Shuffle机制可以使的训练结果更加稳定，并且设置随机数种子，保证文本和标签打乱的顺序一致。

* + 1. 剔除数据中非文本部分

这一步主要是针对一些特殊的非英文字符(non-alpha)比如说标点符号，特殊符号等等，可以用Python的正则表达式(re)把非英文字符转换为空字符。

* + 1. 转化为小写

由于英文单词有大小写之分，我们期望统计时像“Home”和“home” 是一个词。因此一般需要将所有的词都转化为小写。这个直接用python 的API就可以搞定。

* + 1. 分词

把一个句子分为若干单词，这个直接用python的API可以搞定。

* + 1. 剔除停用词

在英文文本中有很多无效的词，比如“a”，“to”，这些在句子中没有实 际意义，不想在文本分析的时候引入，因此需要去掉，这些词就 是停用词，从nltk库可直接下载调用。

* + 1. 单词标准化

词干提取(stemming)，是抽取词的词干或词根形式（不一定能够表达完整语义），对于一些变化的词汇，我们必须处理相同单词的不同形式，并且使得计算机能够明白这些不同的单词有相对的形式。例如，单词sing 能够以很多形式出现如sang,singer ,singing等等。这些是有相同意思的一系列单词。人类能够轻松的识别这些单词的基础形式和衍生形式。当分析文本时，提取这些基本形式时很有用的。它将让我们提取有用的统计数来分析输入的文本。提取词根能够做到这一点，目的是通过将单词的不同形式转换为基本形式来减少单词量。nltk.stem.porter可以实现这一操作

* + 1. 构建词表

按照单词出现的频率从大到小排序，建立字典，将每个词或映射到一个唯一的索引(index)将文本从词或者字的序列转换为索引的序列，方便输入模型。

* + 1. Padding与Truncating

对于文本预处理来说，设置一个句子最大长度，padding（填充）使得每个序列都具有相同的长度（最大长度），truncating（截断）使长于最大长度的序列截断到最大长度。

* 1. **模型构建**
     1. 卷积神经网络：

嵌入层（Embedding）：词嵌入使一种语义空间到向量空间得映射，简单说就是把每个词语都转换成固定维数得向量，并且保证语义接近得两个词转化为向量后，这两个向量得相似度也高。嵌入层可以使用预先创建的嵌入矩阵，比如说使用word2vec。数据量较大：可以直接随机初始化embeddings，然后基于语料通过训练模型网络来对embeddings进行更新和学习。数据量较小：可以利用外部语料来预训练(pre-train)词向量，然后输入到Embedding层，用预训练的词向量矩阵初始化embeddings。（通过设置weights=[embedding\_matrix]）。静态(static)方式：训练过程中不再更新embeddings。实质上属于迁移学习，特别是在目标领域数据量比较小的情况下，采用静态的词向量效果也不错。（通过设置trainable=False），非静态(non-static)方式：在训练过程中对embeddings进行更新和微调(fine tune)，能加速收敛。（通过设置trainable=True）。

卷积层（Convolution）：卷积核在划动过程中提取局部特征，减少需要处理的信息，提高计算效率。文本卷积与图像卷积的不同之处在于只在文本序列的垂直方向做卷积，卷积核的宽度固定为词向量的维度d。

池化层（Pooling）：把上一层的数据做下采样（数据压缩），降低特征的维度，保持重要信息

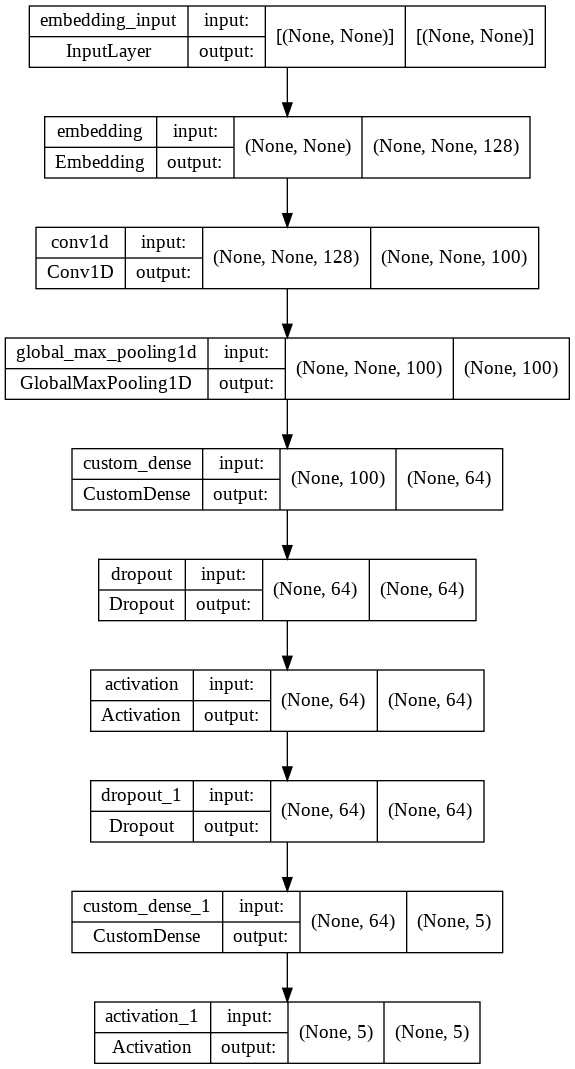
隐藏层（hidden layer）：使神经网络更好地提取特征

Dropout层：随机舍弃一些隐藏神经元，下一轮训练恢复（未舍弃的神经元得到训练），继续随机舍弃，重复这一过程，目的是防止过拟合。

全连接层（dense layer）：一层线性层，通过继承keras.layers.Layer父类，包括\_\_inti\_\_方法初始化自定义线性层所需要的属性，build方法创建可训练的层权重，这个方法使自动调用的，call方法用来编写层的功能性逻辑的地方。

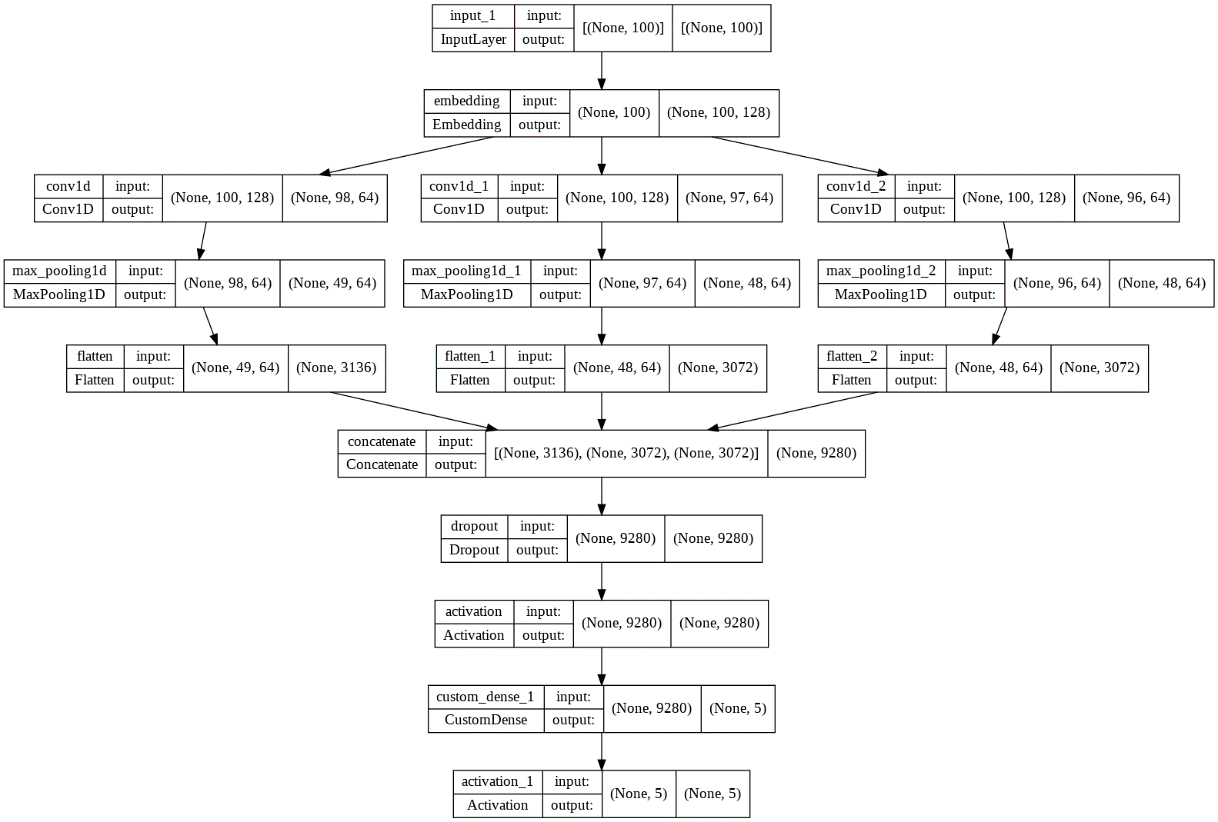
激活层：把前面层的结果做非线性映射，relu、softmax、than等为常见的激活函数。

结构如下：



1. textcnn模型：

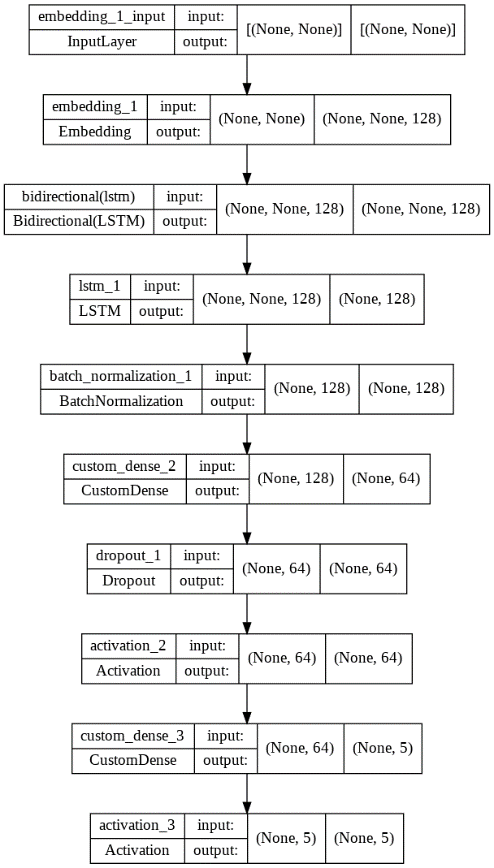
在普通的cnn模型基础上，还包括了不同窗口大小的卷积核， kernel\_size通常为3，4，5；卷积计算出的特征经过MaxPooling保留 最大特征，然后拼接成一个向量作为文本表示。



1. bi-lstm模型：

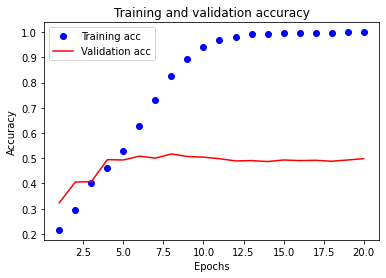
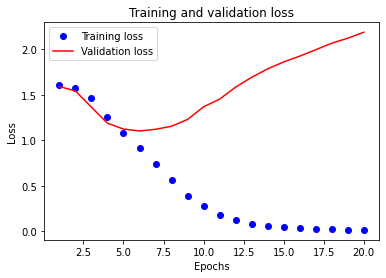
是rnn循环神经网络的一种，是为了处理有序的数据，预测序列化的数据，内部包括遗忘门，输入门，输出门。针对记忆的长短，以及应该遗忘何种信息，记住何种信息。

模型结构：



* + 1. 训练效果可视化：

其中损失为the sparse categorical crossentropy loss

评价指标为the sparse categorical crossentropy accuracy

通过画图操作，将损失函数的值作为y轴，迭代轮数作为x轴，可以得到训练集损失函数值随着迭代次数的增加而减少，最后曲线变得比较平缓，可得训练集损失函数已经收敛，而验证集的曲线先下降再上升，得不到有效的收敛。

右图将准确率作为y轴，迭代轮次作为x轴，可以得到训练集的准确率随着迭代轮数的增加而增加，验证集的准确率增加到一定程度就不再增加，有下降的趋势。

1. 实验
2. 寻找词向量的对于此次分类任务的最佳维度

要将自然语言交给机器学习中的算法来处理，通常需要首先将语言数学化，词向量就是用来将语言中的词进行数学化的一种方式。可以用向量的距离来衡量词相似度。如维度太小，对于词与词之间的关系的表示效果就会较差，太大则会影响性能，计算效率低。

业界对于词向量的选择一般是固定公式，比如说通常是128维，也总结出了一些经验公式：embedding\_dim= vocab\_size\*\*0.25、embedding\_dim>8.33log(vocab\_size)

实验通过设置单一变量，调节不同的embedding\_dim参数，研究对模型训练的影响程度。

实验结果如下：

|  |  |
| --- | --- |
| **embedding\_size** | **val\_s\_c\_acc** |
| 32 | 0.4805 |
| 64 | 0.5019 |
| 96 | 0.5054 |
| 128 | 0.5293 |
| 256 | 0.5206 |
| 512 | 0.5468 |
| 1024 | 0.5381 |

小结：128维对于模型就已经足够了，再增加维度效果不是很明显。

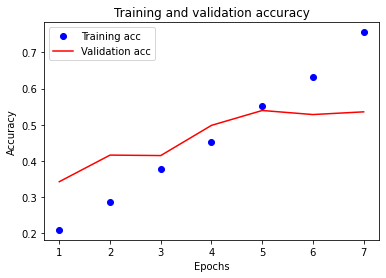
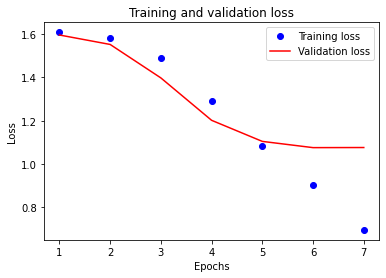
1. 探究Earlystopping的效果

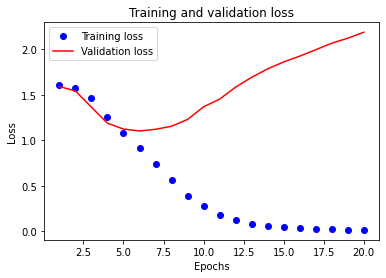
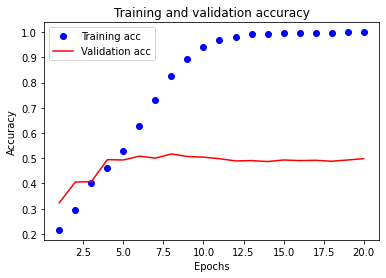
为了获得性能良好的神经网络，网络定型过程中需要进行许多关于所用设置（超参数）的决策。超参数之一是epoch的数量：应当完整遍历数据集多少次（一次为一个epoch）如果epoch数量太少，网络有可能发生欠拟合（即对于定型数据的学习不够充分）；如果epoch数量太多，则有可能发生过拟合（即网络对定型数据中的“噪声”而非信号拟合）。早停法旨在解决epoch数量需要手动设置的问题。

开始时,将训练的数据分为训练集和验证集。每个epoch结束后（或每N个epoch后)： 在验证集上获取测试结果，记录目前为止最好的验证集精度, 而随着epoch的增加，如果在验证集上发现测试误差上升，则停止训练；

将之前处理测试集时准确率最高时的权重作为网络的最终参数。

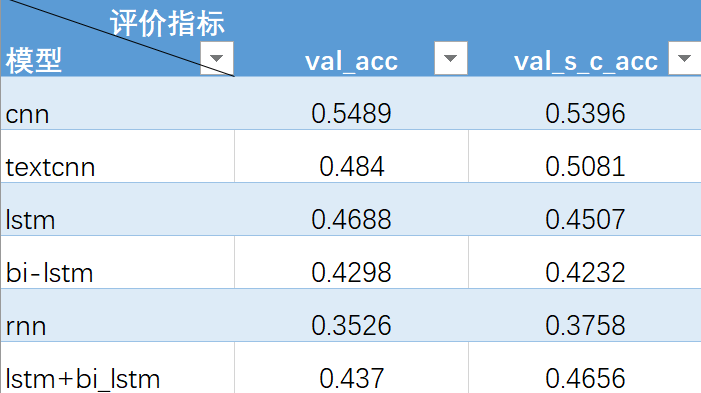
效果对比：未使用早停法epoch为20轮，验证集loss曲线先呈下降后上升趋势，acc曲线先升后小范围下降，发生了过拟合。使用早停法当epoch为7的时候停止训练，一定程度上防止过拟合。



1. 寻找最优模型

设置合适的超参数，使用不同的多分类评价指标，以指标最优为标准选取最优模型：



综合两个评价指标，cnn模型整体上训练效果最好。

1. **总结**

一个月的考核二就此落下帷幕，这个月里实现了文本分类的项目，打榜也取得一定的成绩，但是仍不太满意，未能完成自己的目标。新的征程开始了，继续努力吧。感谢师兄，感谢学校。

1. **参考文献与资料**

[1] keras官方文档<https://keras.io/api/>

[2] <https://www.zhihu.com/question/361999946/answer/953795432>

[3]刘建平. 英文文本挖掘预处理流程总结[OL] <https://www.cnblogs.com/pinard/p/6756534.html>

[4] 萨姆大叔.文本情感分类[OL] <https://www.cnblogs.com/sumuncle/p/6370686.html>