**武汉大学计算机学院**

**本科生课程设计报告**

**Transformer模型对IMDB电影观后感**

**类别判断**

专 业 名 称 ：软件工程

课 程 名 称 ：商务智能

指 导 教 师 ：朱卫平 副教授

学 生 学 号 ：2016301040050

学 生 姓 名 ：吕若凡

二○一九年十二月

**郑 重 声 明**

本人呈交的设计报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计报告不包含他人享有著作权的内容。对本设计报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计报告的知识产权归属于培养单位。

本人签名： 日期：

摘 要

IMDB观后感类别预测实验的目的是运用描述性分析以及预测性分析方法对在Kaggle平台上得到的IMDB Review原始数据集进行数据理解、数据清洗以及模型建立，对未分类的Review进行类型预测。

实验设计主要遵循python语言规范和tensorflow框架规范，实验过程中基于原始数据特征以及理论知识进行模型搭建，避免主观臆测。

实验内容主要包括：首先载入预览原始数据集进行初步数据理解并根据自己的需求对数据清洗加工，为体现实验创新性并未使用传统自然语言理解模型RNN,而是使用了较新的模型Transformer模型，通过分词、词向量转换、迭代产生正确词向量值、，最后建立简单的预测模型对电影评分进行预测。

实验结论为对原始数据进行数据清洗加工后得到了有价值的数据集，在此基础上结合电影产业的相关需求对得到的数据集进行了一系列数据分析，分析结果详尽合理，最终的预测模型得到的预测结果也具有较高的准确度。

**关键词：**词嵌入；transformer；attention；

目录

[1.实验背景与需求分析 5](#_Toc26033552)

[1.1 实验背景 5](#_Toc26033553)

[1.2 需求分析 5](#_Toc26033554)

[1.3 实验环境 5](#_Toc26033555)

[2. 解决方案和关键代码实现 6](#_Toc26033556)

[2.1 数据预处理 6](#_Toc26033557)

[2.2 数据读取 6](#_Toc26033558)

[2.3 词位置编码 7](#_Toc26033559)

[2.4 Transformer模型 8](#_Toc26033560)

[2.4.1 embedding层 9](#_Toc26033561)

[2.4.2 positional\_encoding层 9](#_Toc26033562)

[2.4.3 Multi-head attention层 10](#_Toc26033563)

[2.4.4 全连接层 12](#_Toc26033564)

[2.5 模型评价 13](#_Toc26033565)

[2.6 模型训练与预测 13](#_Toc26033566)

[2.7 实验结果 14](#_Toc26033567)

[3. 问题解决 14](#_Toc26033568)

[4.实验结论 15](#_Toc26033569)

# 1.实验背景与需求分析

## 1.1 实验背景

实验数据集Kaggle项目地址为https://www.kaggle.com/mantri7/imdb-movie-reviews-dataset,共计25000条训练数据，24999条测试数据。数据分析流程包括：提出问题、理解数据、清洗数据、建立模型、模型评估与预测数据、形成数据分析报告。于是实验思路是通过对数据集的预处理操作以及模型训练，得到一个可以自动分析语言情感类别的模型。并使用该模型对测试数据的影评进行情感分类。

## 1.2 需求分析

对于电影行业来说，影评是对电影最有力的评价，影评里也经常会出现对电影情节的概括，赞赏或批评。但影评的数量实在是太多了，即使是专业数据收集和分析人员也无法对所有电影的影评进行分析和归类。并且，影评不像评分数字那样是可以直接被电脑接受和识别，故这不能用简单的全连接神经网络进行分类，词嵌入是最为重要的一步。

明确实验需求之后我们便可以对数据进行处理加工得到理想的数据集并展开相关数据分析。

## 1.3 实验环境

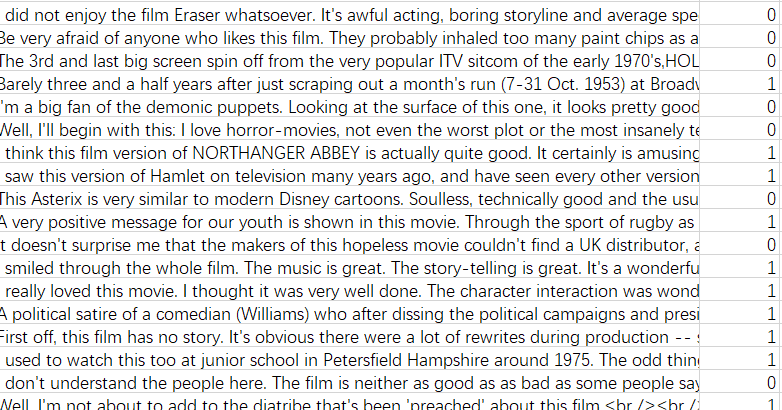
* 操作系统：windows10
* IDE: pycharm
* 编程语言：python3.6
* 深度学习框架：tensorflow2.0
* 其他python包：pandas, numpy, sklearn,kreas

# 2. 解决方案和关键代码实现

## 2.1 数据预处理

在进行数据分析之前，我们首先应该对数据内容和结构，甚至文件类型进行处理。

首先，文件类型是csv类型，即以逗号作为分隔符的文件类型。但打开文件会发现在第一列的影评中含有大量的逗号，使用excel打开虽然显示是正常，影评在第一列，但是在使用python读取时却会出现大量错误，故利用excel将文件保存为.xlsl文件，避免了在读取csv文件时逗号分隔符的影响。

 其次更改每列的header为review和sentiment，便于后续处理时根据列名得到内容。

至此，我们得到了两个xlsx文件，每个文件都由两列组成，第一列是review影评，第二列为sentiment分类。

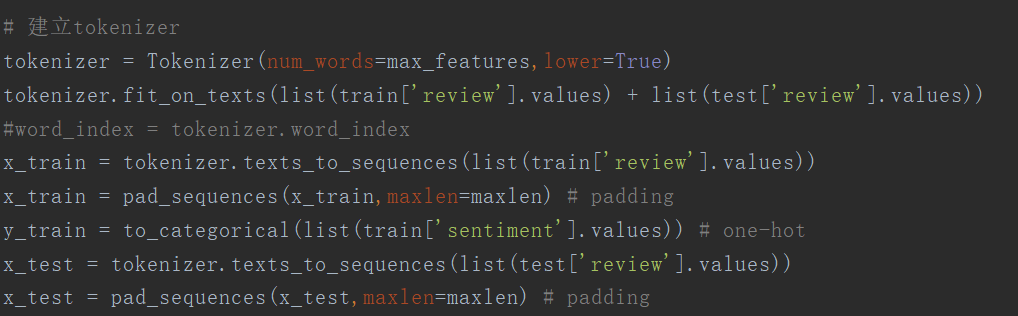
## 2.2 数据读取

直接使用pandas的read\_excel()函数，即可将数据读取至内存中并且以字典的形式读取每一列的值。

## 2.3 词位置编码

首先，我们使用的数据是英文句子或段落，由于英文的特性是每个词用空格分开，句子与句子之间用逗号或句号等标点符号分开，故在进行token划分时可以省去用分词工具划分token的过程。

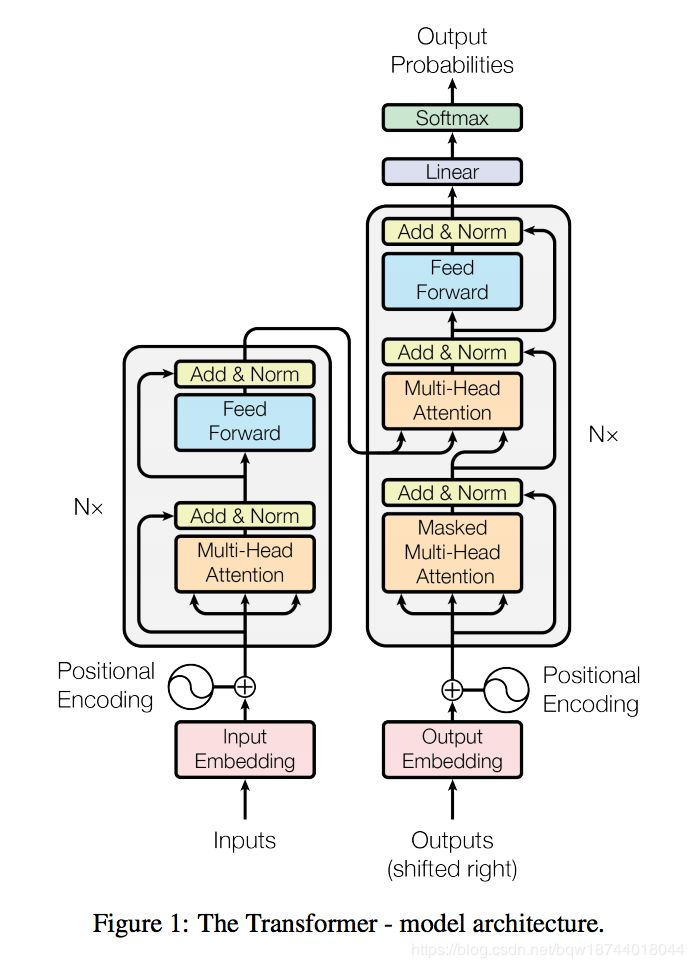
那么接下来就是对词进行位置编码，keras提供了文本处理的词位置编码接口Tokenizer，Tokenizer是一个用于向量化文本，或将文本转换为序列（即单个字词以及对应下标构成的列表，从1算起）的类。是用来文本预处理的。返回的是最常见的、出现频率最高的num\_words个字词。所以我们初始化一个tokenizer对象，对训练和测试数据集的词进行位置编码。

然后对于训练数据进行词位置向量转换，得到训练数据影评的词位置向量，这里值得注意的是需要对位置向量进行填充得到长度相等的词向量以避免出现维度不同导致矩阵乘法错误。使用pad\_sequences()函数以进行填充。该函数是将序列转化为经过填充以后的一个长度相同的新序列新序列，且默认填充0在向量的最前面。

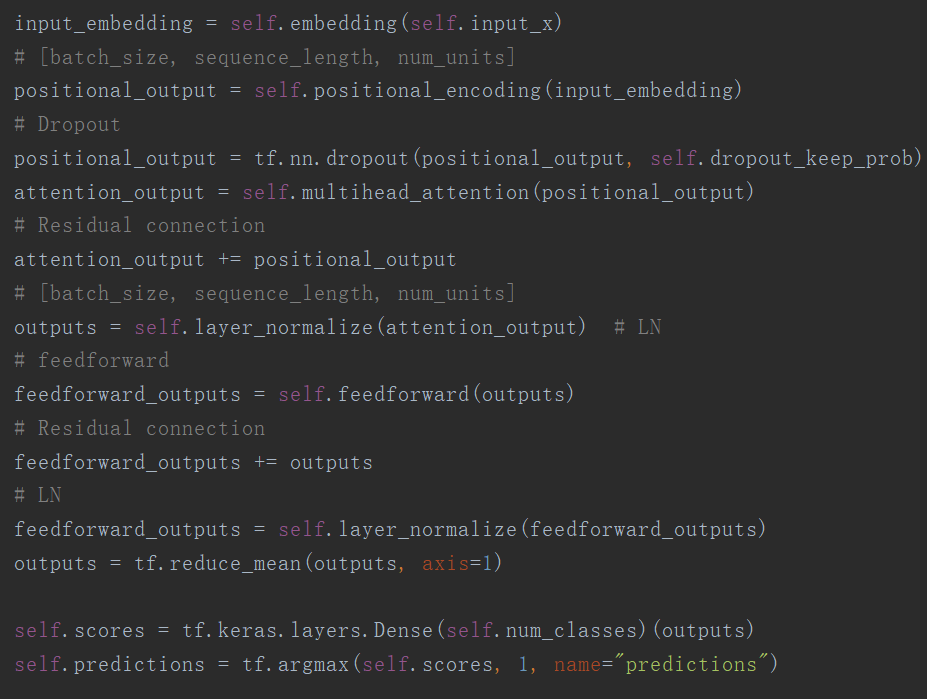
对影评处理完成，那么下面是将标签转换为向量，使用to\_categorical函数处理这一列，该函数就是将类别向量转换为二进制（只有0和1）的矩阵类型表示。其表现为将原有的类别向量转换为独热编码的形式。

至此，数据预处理和读取就已经完成，使用自带的验证集和训练集划分函数即可得到训练集和验证集，下面就是模型定义和训练的过程。

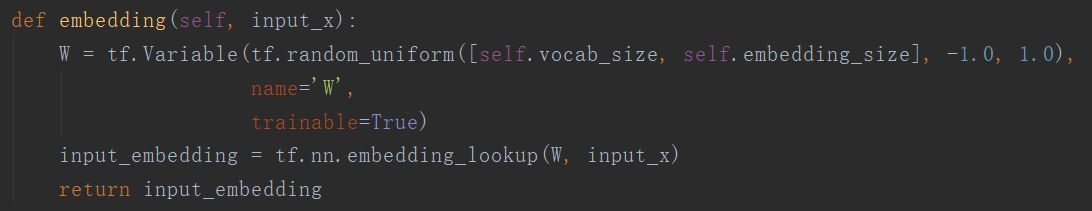
## 2.4 Transformer模型

 transformer模型结构如下图所示，本图是对transformer模型的简化，因为原始的transformer模型的encoder和decoder有6层如图结果。考虑到机器性能问题只实现一层机制。

可知，本文的Transformer模型层次可分为embedding层, positional encoding层， Multi-head attention层，残差连接层，Layer Normalization层和全连接神经网络层。其中最后的全连接神经网络层为计算得到文本的分类。神经网络构造函数代码如下图所示：

接下来详细介绍某些重要的层的作用。

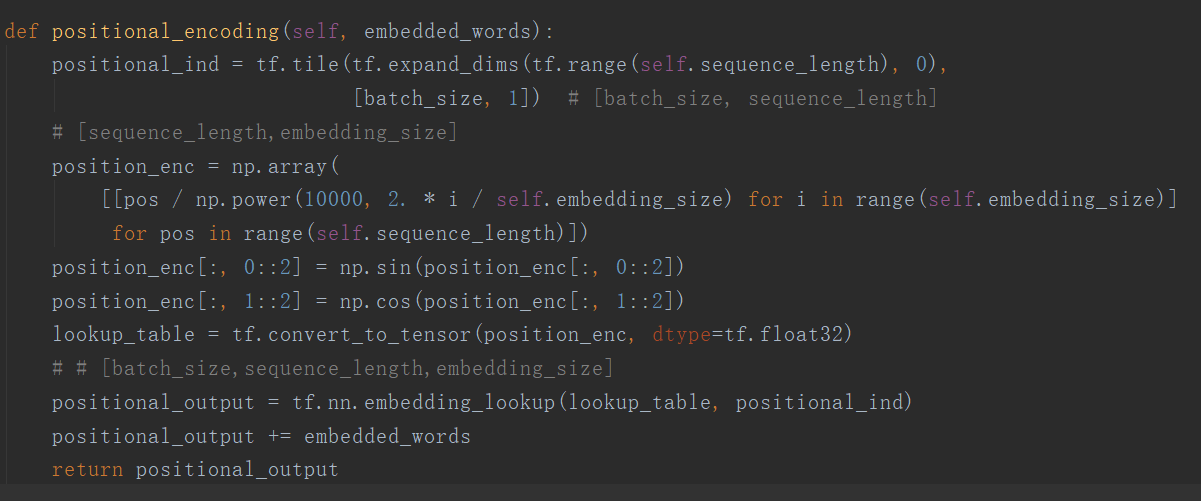
### 2.4.1 embedding层

**** 初始化时，为每个词赋一个随机的初始化向量，后续在训练过程中会不断更新这个向量。tf.nn.embedding\_lookup函数的用法主要是选取一个张量里面索引对应的元素。即寻找embedding层中W对应位置的元素。这样就将token的词位置编码转化为词向量。

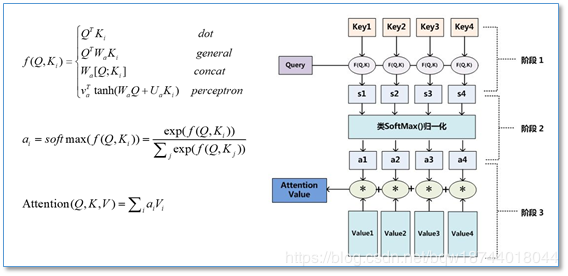
### 2.4.2 positional\_encoding层

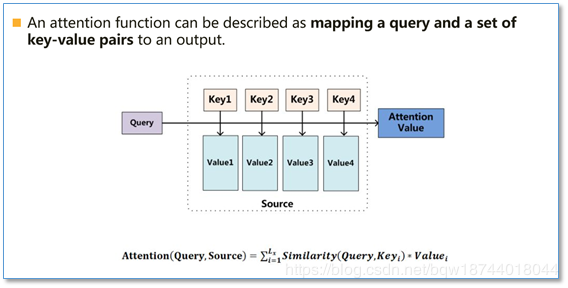
RNN神经网络适合自然语言理解的最大优势在于，RNN可以记住输入的顺序和状态，而自然语言理解中输入顺序会对结果造成很大的影响。考虑到这个因素，transformer模型加入位置编码的网络层。

一种比较经典的位置编码是使用绝对位置作为位置向量，但这种方法没有考虑到词的相对位置。为解决这个问题，模型使用三角函数对奇偶维进行变换得到位置编码。另外使用这个函数可以将当前token的影响传播的更远，这也是本位置编码比RNN优化的地方。

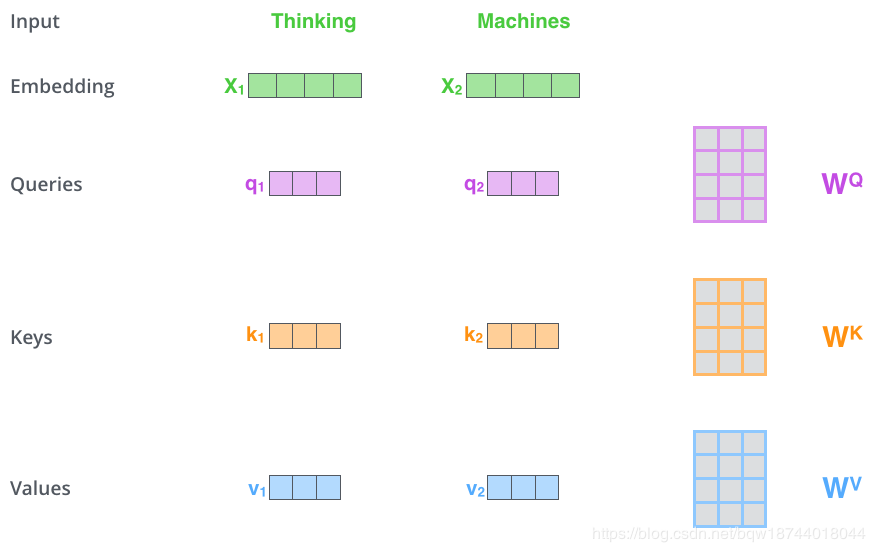


### 2.4.3 Multi-head attention层

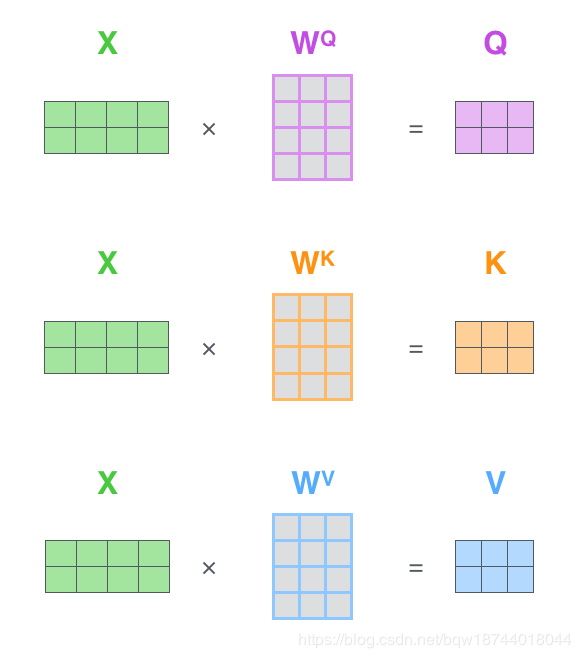
 首先介绍下注意力机制，Attention机制可以看做计算query与key的相似度，然后通过softmax得到每个key对query的贡献度(概率分布)；然后使用这个贡献度做为权重，对value进行加权求和得到Attention的最终输出。在NLP中通常key和value是相同的。上述过程可以用下图表示。

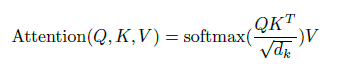


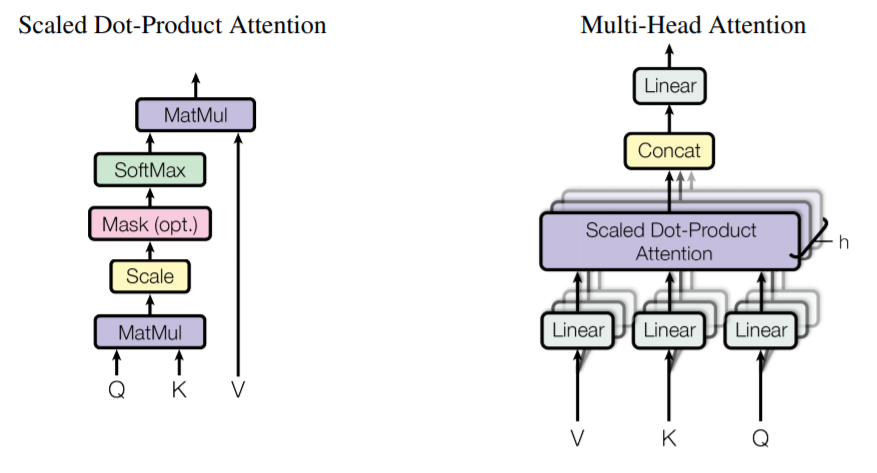
query,key和value的获取过程是从一个输入，乘以3个不同的权重矩阵得到不同的向量值，获取过程如下图所示。

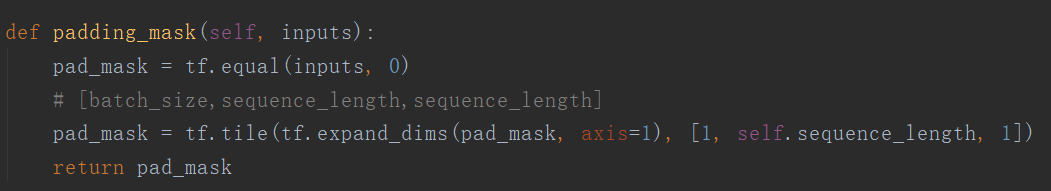


可以观测到q,k,v的维度是由权重矩阵的维度决定的，而这个维度由我们自己决定，我们设计维度为128的query,key和value向量。

有了q与各个k的相似度以后，使用这些相似度对v进行加权求和，得到当前query的输出。在实际运算中，是使用矩阵运算来加速运算。

计算公式可以用下图表示：

而多头注意力是为了再次加速运算，将不同的query同时计算出注意力值，其差别可以用下图表示。

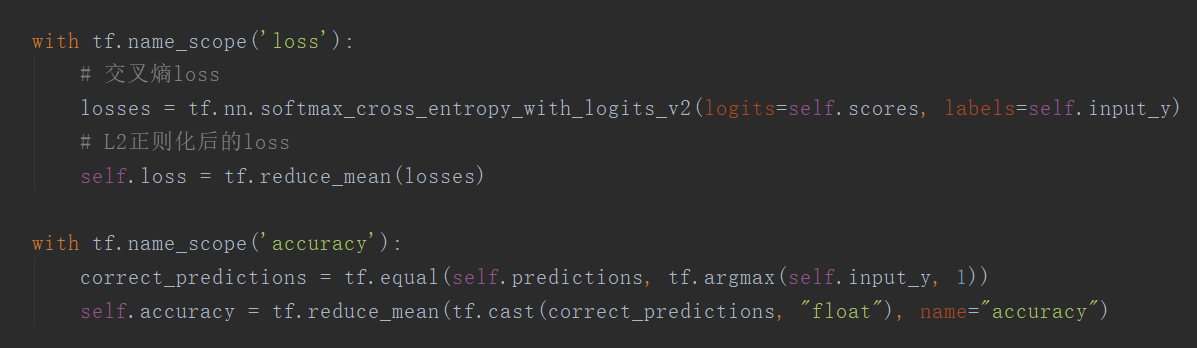
 除此之外，本模型对输入还有paddingMask处理。因为文本的长度不同，所以我们会在处理时对其进行padding。但是在进行attention机制时，注意力不应该放在padding的部分。因此将这些位置设置为非常大的负数，这样经过softmax后的概率接近与0。对于矩阵QKT的第i行第j列的元素表示的是第i个query与第j个key的相似度，为了进行padding mask，那么必须要将所有与padding key的相似度设为负无穷。因此生成一个形状与QKT相同的padding 矩阵，其中所有padding key对应的列设为false,其余为true。

### 2.4.4 全连接层

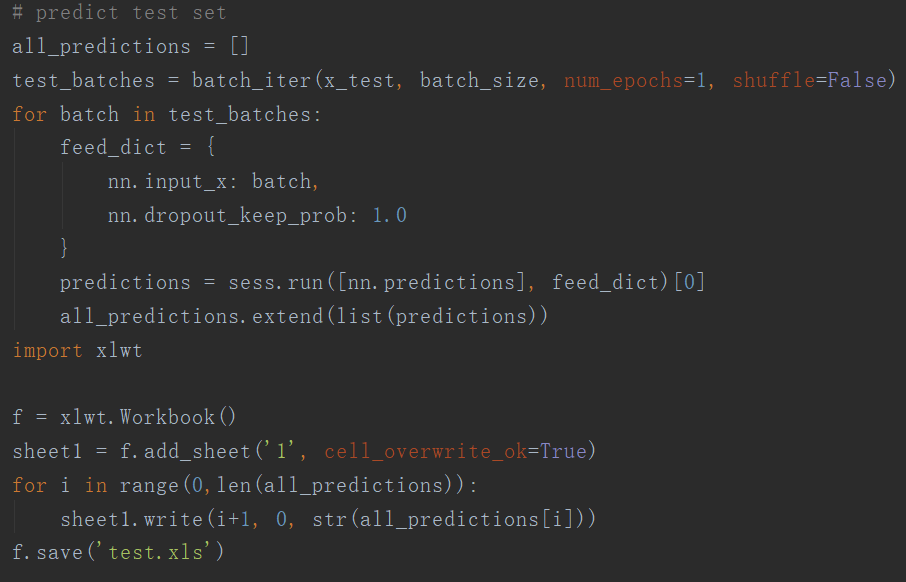
在2.4.3节中我们得到了各个词的注意力的值，再经过残差连接和LN计算方差后，最后进入一个全连接层得到一个二维向量。全连接层，包含两个线性变换和一个非线性激活函数ReLU，公式为FFN(x) = max(0, xW1 + b1)W2 + b2 。这个公式可以用两个大小为1的一维卷积来实现，其中中间层的维度为2048。

最后通过tf.argmax(input,axis)函数获取最终标签。

## 2.5 模型评价

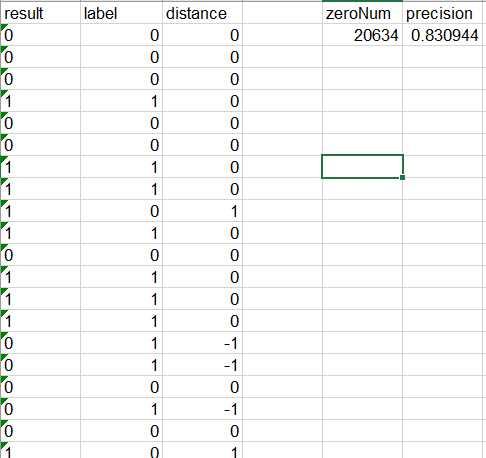
 使用交叉熵作为模型的损失函数，使用精确率作为评估模型的精确度。

## 2.6 模型训练与预测

 受限于电脑性能，在迭代到1600步时，已经花了8个小时。最终 loss为 0.400088, acc 为0.841191。这对于自然语言理解已经是较好的结果，于是停止训练，在下次运行程序使用保存好的模型对测试数据进行预测。并使用xlwt包将预测结果保存至excel文件中。

## 2.7 实验结果

第一列是预测结果，第二列是真实标签，第三列是预测结果与真实标签的差距，那么可知当distance为0时是正确标签，计算可以得到正确率是0.83。

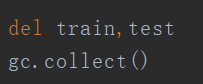


# 3. 问题解决

1. 数据集读取问题

由于kaggle下载数据是csv格式，但其第一列影评有很多逗号，这给pandas读取带来了困扰。我尝试很多方法例如指定read\_csv()函数中quoting=3和忽略错误继续读等，但都会出现报错。最后只好将csv文件转换为xlsx文件，直接读取第一列就是影评文本，没有出现错误。

1. 内存溢出问题

由于数据集较大，并且当还有对数据集的位置编码转换等等，导致有重复且无用的数组占用较多内存。为解决这个问题将不需要的变量手动删除并且主动调用垃圾回收方法。

1. 机器性能差

由于本机器没有GPU,只能使用较慢的cpu进行运算，在运行8小时后只进行到1600步，只好终断程序并保存当前最好的模型最为训练结果，并使用该模型进行测评。

1. 原始模型的简化与实现

transformer模型是最近的新成果，其思想和理念都是世界领先的，并且也打破了自然语言理解多项任务的世界记录，其神经网络层数较深且神经连接复杂，为保证得到较好的结果，对其进行简化和优化。其实现论文为英文论文，地址为：<http://arxiv.org/abs/1706.03762> ，也参考了很多博主的实现，最终完成。

1. RNN与Transformer模型对比

通常认为RNN适合进行自然语言理解，这是因为NLP中的一个理论：一个词的含义可以使用他的上下文context表示，并且距离越近，其表达能力越强。而RNN就是一个局部感知模型，他的t+1时刻的状态由前k个状态决定，如果使用双向RNN，就实现了NLP中的context理论。但是这种方法也有一定的缺陷，他只对位置敏感而对真实含义敏感度低，所以Transformer模型引入attention机制，不仅考虑位置因素，还将相似度纳入考虑范围，attention在这个模型中的理解为self-attention，即对与自己相似的单词更加注意。故Transformer模型在自然语言理解中比RNN更胜一筹。

# 4.实验结论

通过对原始数据集进行数据清洗、加工，成功获得了比较有代表意义且在后续分析中能够合理使用的数据；

实验过程中具体分析了如何使用transformer模型对自然语言进行处理和预测，并通过介绍各个层的功能和思想，对比RNN更加成熟和成功的处理方法。文章重点介绍了attention机制的原理，具有一定的前沿性和创新性。

最终合理的使用模型对影评分类取得了较好的结果。完成的要求有单独完成本实验报告，在transformer思想的基础上，简化其网络层次并将其运用在实际应用上完成本次实验，使用tensorflow的基础框架且没有使用其提供的transformer模块完成本次实验。

教师评语评分

评语：

评分：

评阅人：

年 月 日

（备注：对该实验报告给予优点和不足的评价，并给出百分之评分。）