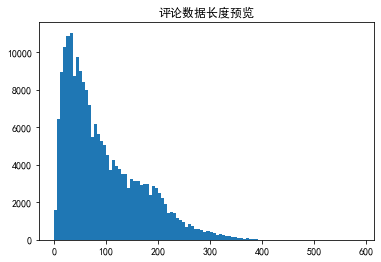
# 基于CNN与LSTM的情感分类的模型

## 一、数据预览与预处理阶段

数据准备的是从网络下载的微博评论数据集，数据集总共分为四类，分别对应的四种情感是喜悦，激动，厌恶与低落。每种情感选择了50000条评论。由于每条评论来自网络，有较多的符号语言，运用正则表达式对评论进行清洗。调用python 的jieba对评论进行分词的处理，并且计算分词过后的每条评论的长度用于下面分词器的建立的工作的准备。



**（图一）**

**（图二）**

count 206000.000000

mean 95.723655

std 72.731714

min 1.000000

25% 38.000000

50% 74.000000

75% 141.000000

max 585.000000

Name: num of fenci, dtype: float64

**（图三）**

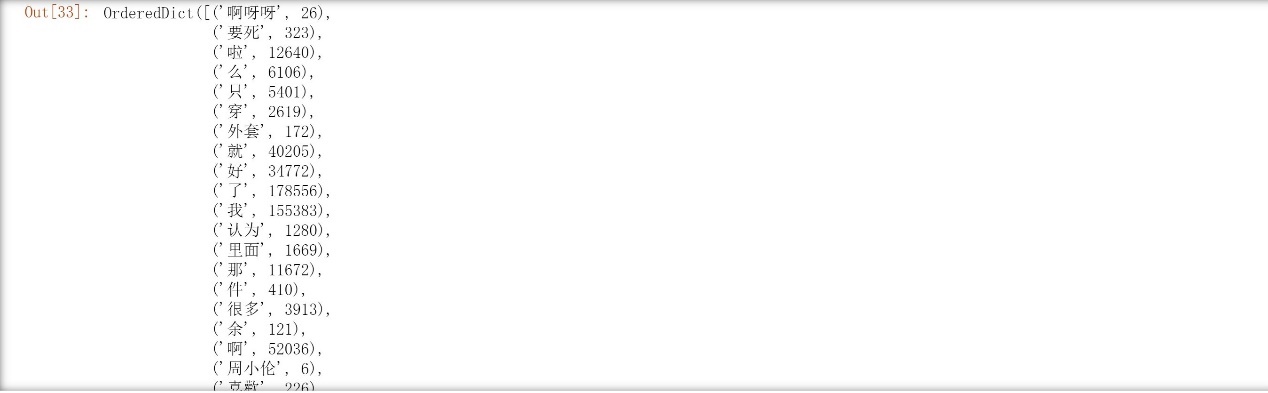
图一展示了选择的数据集的各个种类的评论选择的数量是分布均匀的，图二的频数直方图展示的是每一句话在进行分词过后的长度，可以看到平均的长度是95，且大多数的评论都是在100词以内，因此在接下来的分词器的创建中选择的最大的词语数量就选择了100。

## 二、分词器，词典的创建

经过第一部分的分析，因此已经对于分词器的参数有了认识。运用keras的tokennizer对于已经分完词的数据进行处理，获得了训练数据集，稳定数据集和测试数据集的矩阵，以及一个对应的出现的词语的词频字典和词语的数量的词典。经过分词器处理过的数据的维度是训练集：200000\*100，稳定集：8000\*100以及测试集：2000\*100。到这里，完成了对于模型的数据的预处理的全部的工作。



**（图四）**

****

**（图五）**

图四展示的是所有的数据中出现的词语的频率以及出现的频率的排名，图五展示的是所有的数据中词语的出现的数量。其中总共选择了5000个词语，每一个词语的出现的数量以及出现数量的排名都已经被统计。

## 三、模型的选择与训练

完成了对于数据的预处理之后，开始对于神经网络的模型的搭建以及训练。在经过资料的阅读以及考量，考虑搭建MLP神经网络，TextCNN神经网络以及TextCNN+LSTM神经网络并进行网络的训练。

这三种神经网络各有特点，全连接神经网络的特点是比较简单，但是缺点就是对于数据的预测的效果是比较差的，CNN网络可以对于数据的特征进行比较好的提取，TextCNN模型已经成为了提取文本特征的有力的工具，但是多层的CNN网络会导致梯度的消失，我通过模型的训练对于传统的全连接的CNN模型进行了改良。LSTM网络对于传统的RNN神经网络有一个比较好的提升，对于多变量的问题有比较好的效果。LSTM也已经广泛的运用到了情感分类之中，我也对于传统的LSTM进行了增强的处理。下面我对于这三种神经网络都分别进行了训练和测试，每一次的训练次数都是20次，每1000个训练数据分为一组，所以一次的训练数据需要训练200次。记录了随实验的次数变化的测试集和稳定集的准确率以及损失值的变化。

### （1）全连接神经网络

全连接层的网络的基本结构是经过一层词嵌入层更改输入的词向量的维度之后连接了三层的全连接层，最后一层是输出维度为4的全连接层，模型的激活函数是adam。

Model: "sequential\_7"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_29 (Dense) (None, 32) 3232

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_30 (Dense) (None, 16) 528

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_31 (Dense) (None, 4) 68

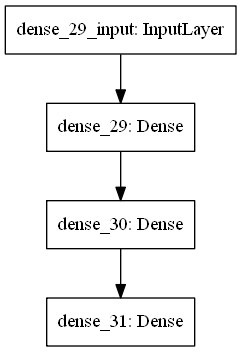
=================================================================

Total params: 3,828

Trainable params: 3,828

Non-trainable params: 0

**（图六）**



**（图七）**

Epoch 1/20

200/200 [==============================] - 0s 2ms/step - loss: 82.1700 - accuracy: 0.2497 - val\_loss: 2.0274 - val\_accuracy: 0.2473

Epoch 2/20

200/200 [==============================] - 0s 2ms/step - loss: 1.6381 - accuracy: 0.2492 - val\_loss: 1.4417 - val\_accuracy: 0.2496

Epoch 3/20

200/200 [==============================] - 0s 2ms/step - loss: 1.4307 - accuracy: 0.2483 - val\_loss: 1.3990 - val\_accuracy: 0.2500

Epoch 4/20

200/200 [==============================] - 0s 2ms/step - loss: 1.4038 - accuracy: 0.2493 - val\_loss: 1.3898 - val\_accuracy: 0.2499

Epoch 5/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3950 - accuracy: 0.2497 - val\_loss: 1.3872 - val\_accuracy: 0.2503

Epoch 6/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3908 - accuracy: 0.2482 - val\_loss: 1.3866 - val\_accuracy: 0.2501

Epoch 7/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3890 - accuracy: 0.2484 - val\_loss: 1.3864 - val\_accuracy: 0.2501

Epoch 8/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3883 - accuracy: 0.2490 - val\_loss: 1.3864 - val\_accuracy: 0.2501

Epoch 9/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3878 - accuracy: 0.2492 - val\_loss: 1.3863 - val\_accuracy: 0.2501

Epoch 10/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3874 - accuracy: 0.2505 - val\_loss: 1.3863 - val\_accuracy: 0.2501

Epoch 11/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3869 - accuracy: 0.2479 - val\_loss: 1.3868 - val\_accuracy: 0.2500

Epoch 12/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3867 - accuracy: 0.2488 - val\_loss: 1.3862 - val\_accuracy: 0.2501

Epoch 13/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3866 - accuracy: 0.2494 - val\_loss: 1.3862 - val\_accuracy: 0.2501

Epoch 14/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3865 - accuracy: 0.2500 - val\_loss: 1.3862 - val\_accuracy: 0.2501

Epoch 15/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3865 - accuracy: 0.2504 - val\_loss: 1.3862 - val\_accuracy: 0.2501

Epoch 16/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3864 - accuracy: 0.2497 - val\_loss: 1.3862 - val\_accuracy: 0.2501

Epoch 17/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3864 - accuracy: 0.2488 - val\_loss: 1.3861 - val\_accuracy: 0.2501

Epoch 18/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3864 - accuracy: 0.2495 - val\_loss: 1.3861 - val\_accuracy: 0.2501

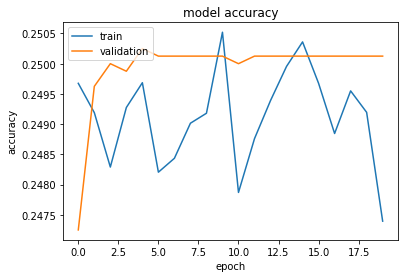
Epoch 19/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3864 - accuracy: 0.2492 - val\_loss: 1.3862 - val\_accuracy: 0.2501

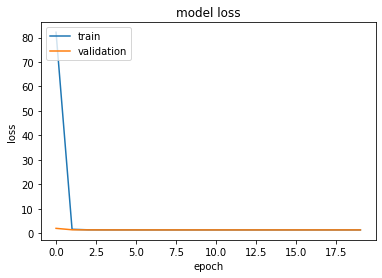
Epoch 20/20

200/200 [==============================] - 0s 2ms/step - loss: 1.3864 - accuracy: 0.2474 - val\_loss: 1.3861 - val\_accuracy: 0.2501

accuracy 25.006500

****

**（图八）**



**（图九）**

图六是模型的概览，图七是模型的结构示意图，图八和图九是模型的准确度和损失值岁实验批次的增加而变化的示意图。实验结果表名全连接层的网络结构简单，但是对于模型的预测效果是非常糟糕的，20论的训练结果准确率只有25%，因此全连接层的网络被我否决了。

### （2）TextCNN模型网络

我构建的TextCNN模型首先是词嵌入层，然后连接的是三层的一维的卷积神经网络与三层的全连接层神经网络，最后输出维度为4的全连接层，模型的激活函数是adam。

Model: "sequential\_13"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

embedding\_8 (Embedding) (None, 100, 256) 1280256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_19 (Conv1D) (None, 100, 80) 61520

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_19 (MaxPooling (None, 50, 80) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_20 (Conv1D) (None, 50, 80) 25680

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_20 (MaxPooling (None, 25, 80) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_21 (Conv1D) (None, 25, 80) 32080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_21 (MaxPooling (None, 12, 80) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 960) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_52 (Dense) (None, 100) 96100

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_53 (Dense) (None, 63) 6363

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_54 (Dense) (None, 32) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_55 (Dense) (None, 4) 132

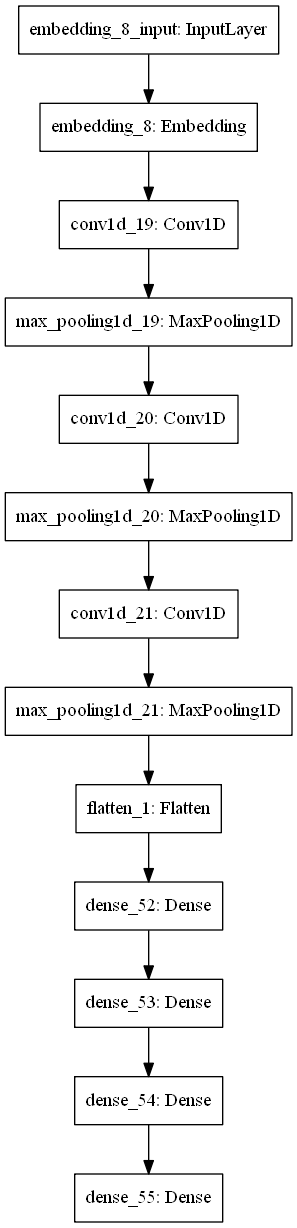
=================================================================

Total params: 1,504,179

Trainable params: 1,504,179

Non-trainable params: 0

**（图十）**



**（图十一）**

Epoch 1/20

200/200 [==============================] - 84s 420ms/step - loss: 1.3863 - accuracy: 0.2530 - val\_loss: 1.3849 - val\_accuracy: 0.2930

Epoch 2/20

200/200 [==============================] - 84s 422ms/step - loss: 1.3388 - accuracy: 0.3376 - val\_loss: 1.2916 - val\_accuracy: 0.3733

Epoch 3/20

200/200 [==============================] - 83s 416ms/step - loss: 1.2785 - accuracy: 0.3786 - val\_loss: 1.2466 - val\_accuracy: 0.3993

Epoch 4/20

200/200 [==============================] - 83s 416ms/step - loss: 1.2318 - accuracy: 0.4107 - val\_loss: 1.2125 - val\_accuracy: 0.4238

Epoch 5/20

200/200 [==============================] - 81s 406ms/step - loss: 1.1678 - accuracy: 0.4561 - val\_loss: 1.1416 - val\_accuracy: 0.4814

Epoch 6/20

200/200 [==============================] - 79s 397ms/step - loss: 1.0946 - accuracy: 0.5018 - val\_loss: 1.0981 - val\_accuracy: 0.5058

Epoch 7/20

200/200 [==============================] - 81s 405ms/step - loss: 1.0233 - accuracy: 0.5366 - val\_loss: 1.0505 - val\_accuracy: 0.5337

Epoch 8/20

200/200 [==============================] - 70s 352ms/step - loss: 0.9615 - accuracy: 0.5638 - val\_loss: 1.0366 - val\_accuracy: 0.5374

Epoch 9/20

200/200 [==============================] - 60s 300ms/step - loss: 0.9137 - accuracy: 0.5843 - val\_loss: 0.9957 - val\_accuracy: 0.5614

Epoch 10/20

200/200 [==============================] - 59s 296ms/step - loss: 0.8736 - accuracy: 0.5998 - val\_loss: 0.9747 - val\_accuracy: 0.5749

Epoch 11/20

200/200 [==============================] - 58s 292ms/step - loss: 0.8389 - accuracy: 0.6140 - val\_loss: 0.9765 - val\_accuracy: 0.5760

Epoch 12/20

200/200 [==============================] - 59s 293ms/step - loss: 0.8127 - accuracy: 0.6257 - val\_loss: 0.9686 - val\_accuracy: 0.5828

Epoch 13/20

200/200 [==============================] - 59s 296ms/step - loss: 0.7878 - accuracy: 0.6363 - val\_loss: 0.9603 - val\_accuracy: 0.5867

Epoch 14/20

200/200 [==============================] - 59s 296ms/step - loss: 0.7685 - accuracy: 0.6427 - val\_loss: 0.9686 - val\_accuracy: 0.5856

Epoch 15/20

200/200 [==============================] - 60s 300ms/step - loss: 0.7534 - accuracy: 0.6496 - val\_loss: 0.9492 - val\_accuracy: 0.5969

Epoch 16/20

200/200 [==============================] - 58s 292ms/step - loss: 0.7402 - accuracy: 0.6527 - val\_loss: 0.9449 - val\_accuracy: 0.5976

Epoch 17/20

200/200 [==============================] - 58s 292ms/step - loss: 0.7253 - accuracy: 0.6576 - val\_loss: 0.9388 - val\_accuracy: 0.6047

Epoch 18/20

200/200 [==============================] - 58s 292ms/step - loss: 0.7177 - accuracy: 0.6605 - val\_loss: 0.9315 - val\_accuracy: 0.6076

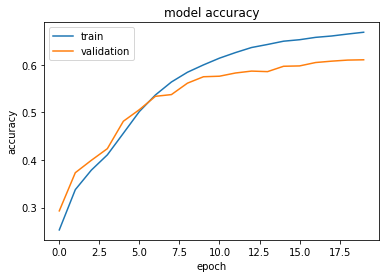
Epoch 19/20

200/200 [==============================] - 59s 297ms/step - loss: 0.7042 - accuracy: 0.6646 - val\_loss: 0.9353 - val\_accuracy: 0.6099

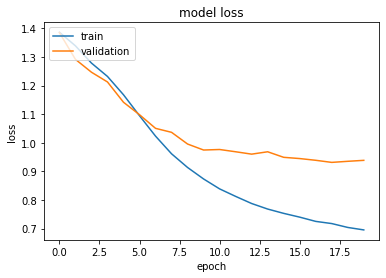
Epoch 20/20

200/200 [==============================] - 59s 296ms/step - loss: 0.6955 - accuracy: 0.6684 - val\_loss: 0.9387 - val\_accuracy: 0.6105

**accuracy 67.447501**

****

**（图十二）**

****

**（图十三）**

图十是模型的概览，图十一是模型的结构示意图，图十二和图十三是模型的准确度和损失值岁实验批次的增加而变化的示意图。实验结果表明卷积神经网络对于文本数据的特征的提取确实是比单纯的全连接层网络强的，但是训练集66%的准确度尚不能让我满意。因此我开始考虑对于普通的卷积神经网络进行加强的处理。

### （3）加强的TextCNN模型网络

考虑运用keras的concatenate对于第二部分的卷积神经网络进行加强的处理，形成Enhanced TextCNN(ETextCNN)，模型的结构是将第二部分的六层卷积神经网络放入concatenate层，从而减少了模型的复杂度，也减少了模型的冗余而带来的梯度消失的情况，也可以增加CNN网络中的输出通道对于数据的特征有更好的提取。

Model: "model\_1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param # Connected to

==================================================================================================

input\_2 (InputLayer) [(None, 100)] 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

embedding\_9 (Embedding) (None, 100, 300) 1500300 input\_2[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_22 (Conv1D) (None, 100, 256) 230656 embedding\_9[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_23 (Conv1D) (None, 100, 256) 307456 embedding\_9[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_24 (Conv1D) (None, 100, 256) 384256 embedding\_9[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_22 (MaxPooling1D) (None, 10, 256) 0 conv1d\_22[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_23 (MaxPooling1D) (None, 10, 256) 0 conv1d\_23[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_24 (MaxPooling1D) (None, 10, 256) 0 conv1d\_24[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

concatenate\_1 (Concatenate) (None, 10, 768) 0 max\_pooling1d\_22[0][0]

max\_pooling1d\_23[0][0]

max\_pooling1d\_24[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_2 (Flatten) (None, 7680) 0 concatenate\_1[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 7680) 0 flatten\_2[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_56 (Dense) (None, 128) 983168 dropout\_1[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_57 (Dense) (None, 64) 8256 dense\_56[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_58 (Dense) (None, 32) 2080 dense\_57[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_59 (Dense) (None, 4) 132 dense\_58[0][0]

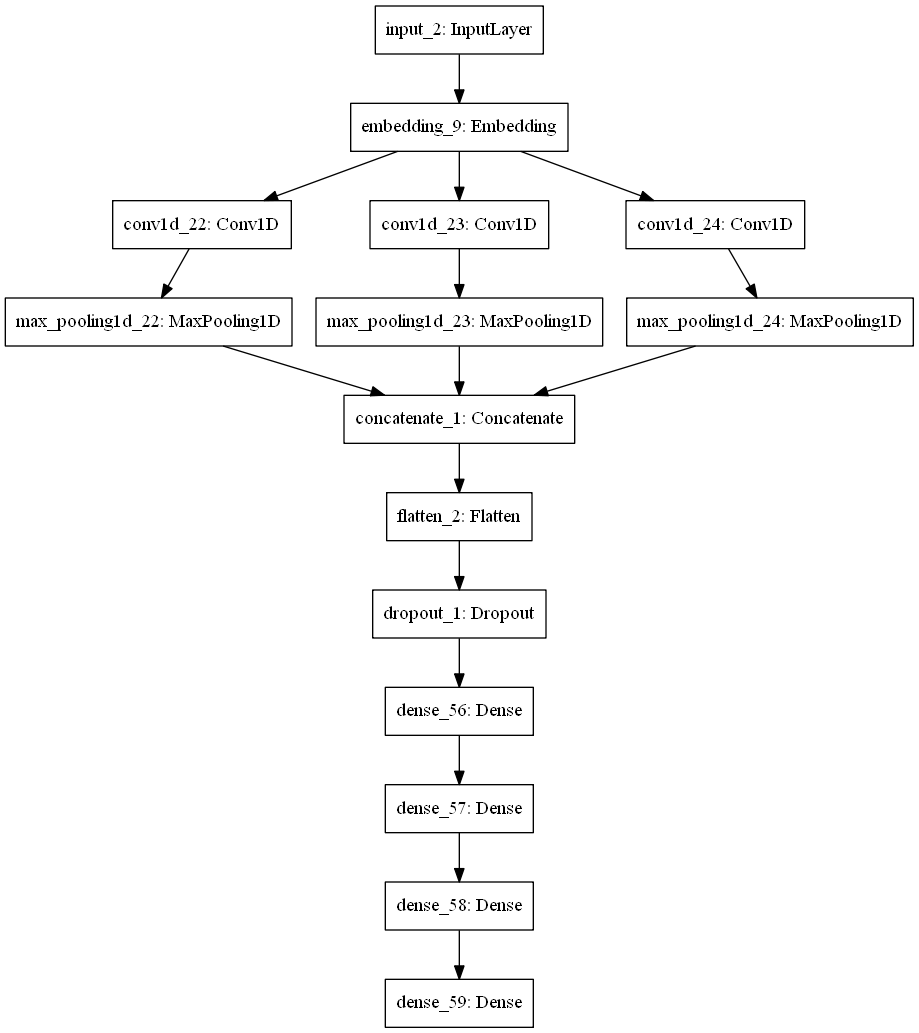
==================================================================================================

Total params: 3,416,304

Trainable params: 1,916,004

Non-trainable params: 1,500,300

**（图十四）**

****

**（图十五）**

Epoch 1/20

200/200 [==============================] - 214s 1s/step - loss: 1.3482 - accuracy: 0.3125 - val\_loss: 1.2658 - val\_accuracy: 0.3911

Epoch 2/20

200/200 [==============================] - 220s 1s/step - loss: 1.2028 - accuracy: 0.4218 - val\_loss: 1.0927 - val\_accuracy: 0.4845

Epoch 3/20

200/200 [==============================] - 222s 1s/step - loss: 1.0259 - accuracy: 0.5005 - val\_loss: 0.9545 - val\_accuracy: 0.5459

Epoch 4/20

200/200 [==============================] - 269s 1s/step - loss: 0.8882 - accuracy: 0.5559 - val\_loss: 0.8890 - val\_accuracy: 0.5780

Epoch 5/20

200/200 [==============================] - 338s 2s/step - loss: 0.7942 - accuracy: 0.5900 - val\_loss: 0.8694 - val\_accuracy: 0.6031

Epoch 6/20

200/200 [==============================] - 341s 2s/step - loss: 0.7290 - accuracy: 0.6155 - val\_loss: 0.9052 - val\_accuracy: 0.6054

Epoch 7/20

200/200 [==============================] - 342s 2s/step - loss: 0.6765 - accuracy: 0.6375 - val\_loss: 0.8477 - val\_accuracy: 0.6245

Epoch 8/20

200/200 [==============================] - 343s 2s/step - loss: 0.6357 - accuracy: 0.6521 - val\_loss: 0.8441 - val\_accuracy: 0.6365

Epoch 9/20

200/200 [==============================] - 342s 2s/step - loss: 0.6007 - accuracy: 0.6655 - val\_loss: 0.8295 - val\_accuracy: 0.6378

Epoch 10/20

200/200 [==============================] - 340s 2s/step - loss: 0.5760 - accuracy: 0.6747 - val\_loss: 0.8407 - val\_accuracy: 0.6400

Epoch 11/20

200/200 [==============================] - 341s 2s/step - loss: 0.5566 - accuracy: 0.6804 - val\_loss: 0.8290 - val\_accuracy: 0.6459

Epoch 12/20

200/200 [==============================] - 340s 2s/step - loss: 0.5411 - accuracy: 0.6869 - val\_loss: 0.8337 - val\_accuracy: 0.6482

Epoch 13/20

200/200 [==============================] - 344s 2s/step - loss: 0.5270 - accuracy: 0.6920 - val\_loss: 0.8761 - val\_accuracy: 0.6470

Epoch 14/20

200/200 [==============================] - 344s 2s/step - loss: 0.5124 - accuracy: 0.6972 - val\_loss: 0.8633 - val\_accuracy: 0.6494

Epoch 15/20

200/200 [==============================] - 353s 2s/step - loss: 0.5049 - accuracy: 0.6972 - val\_loss: 0.8595 - val\_accuracy: 0.6500

Epoch 16/20

200/200 [==============================] - 346s 2s/step - loss: 0.4965 - accuracy: 0.7035 - val\_loss: 0.8621 - val\_accuracy: 0.6530

Epoch 17/20

200/200 [==============================] - 338s 2s/step - loss: 0.4865 - accuracy: 0.7050 - val\_loss: 0.8975 - val\_accuracy: 0.6530

Epoch 18/20

200/200 [==============================] - 342s 2s/step - loss: 0.4796 - accuracy: 0.7065 - val\_loss: 0.8837 - val\_accuracy: 0.6576

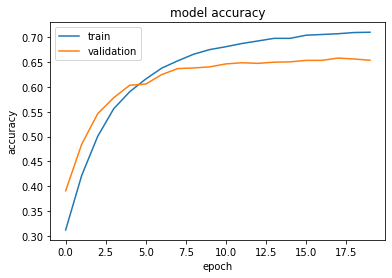
Epoch 19/20

200/200 [==============================] - 339s 2s/step - loss: 0.4741 - accuracy: 0.7089 - val\_loss: 0.9008 - val\_accuracy: 0.6557

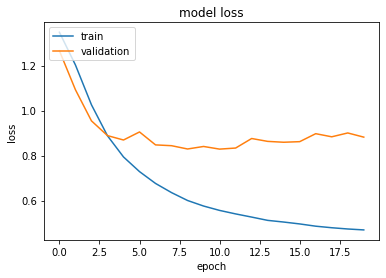
Epoch 20/20

200/200 [==============================] - 338s 2s/step - loss: 0.4699 - accuracy: 0.7095 - val\_loss: 0.8822 - val\_accuracy: 0.6532

**accuracy 71.01**

****

**（图十六）**

****

**（图十七）**

图十四是模型的概览，图十五是模型的结构示意图，图十六和图十七是模型的准确度和损失值岁实验批次的增加而变化的示意图。实验结果表明加强的卷积神经网络对于文本数据的特征的提取确实是比普通的卷积神经网络更强，准确度已经提高了5%。因此在接下来的模型中我舍弃了传统的CNN模型使用加强的CNN模型。

### （4）LSTM-TextCNN模型网络

在加强的卷积神经网络的基础上我增加了LSTM模型，在情感分析的领域LSTM模型已经广为使用，因此在我的 第三部分的模型的基础上，又在加强卷积神经网络与全连接层神经网络之间又增加了一层LSTM网络。

Model: "model\_2"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param # Connected to

==================================================================================================

input\_3 (InputLayer) [(None, 100)] 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

embedding\_10 (Embedding) (None, 100, 256) 1280256 input\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_25 (Conv1D) (None, 100, 256) 196864 embedding\_10[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_26 (Conv1D) (None, 100, 256) 262400 embedding\_10[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_27 (Conv1D) (None, 100, 256) 327936 embedding\_10[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_28 (Conv1D) (None, 100, 256) 393472 embedding\_10[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_29 (Conv1D) (None, 100, 256) 459008 embedding\_10[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_25 (MaxPooling1D) (None, 10, 256) 0 conv1d\_25[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_26 (MaxPooling1D) (None, 10, 256) 0 conv1d\_26[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_27 (MaxPooling1D) (None, 10, 256) 0 conv1d\_27[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_28 (MaxPooling1D) (None, 10, 256) 0 conv1d\_28[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_29 (MaxPooling1D) (None, 10, 256) 0 conv1d\_29[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

concatenate\_2 (Concatenate) (None, 10, 1280) 0 max\_pooling1d\_25[0][0]

max\_pooling1d\_26[0][0]

max\_pooling1d\_27[0][0]

max\_pooling1d\_28[0][0]

max\_pooling1d\_29[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_1 (LSTM) (None, 256) 1573888 concatenate\_2[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_3 (Flatten) (None, 256) 0 lstm\_1[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_60 (Dense) (None, 256) 65792 flatten\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_61 (Dense) (None, 128) 32896 dense\_60[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_62 (Dense) (None, 64) 8256 dense\_61[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_63 (Dense) (None, 32) 2080 dense\_62[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_64 (Dense) (None, 4) 132 dense\_63[0][0]

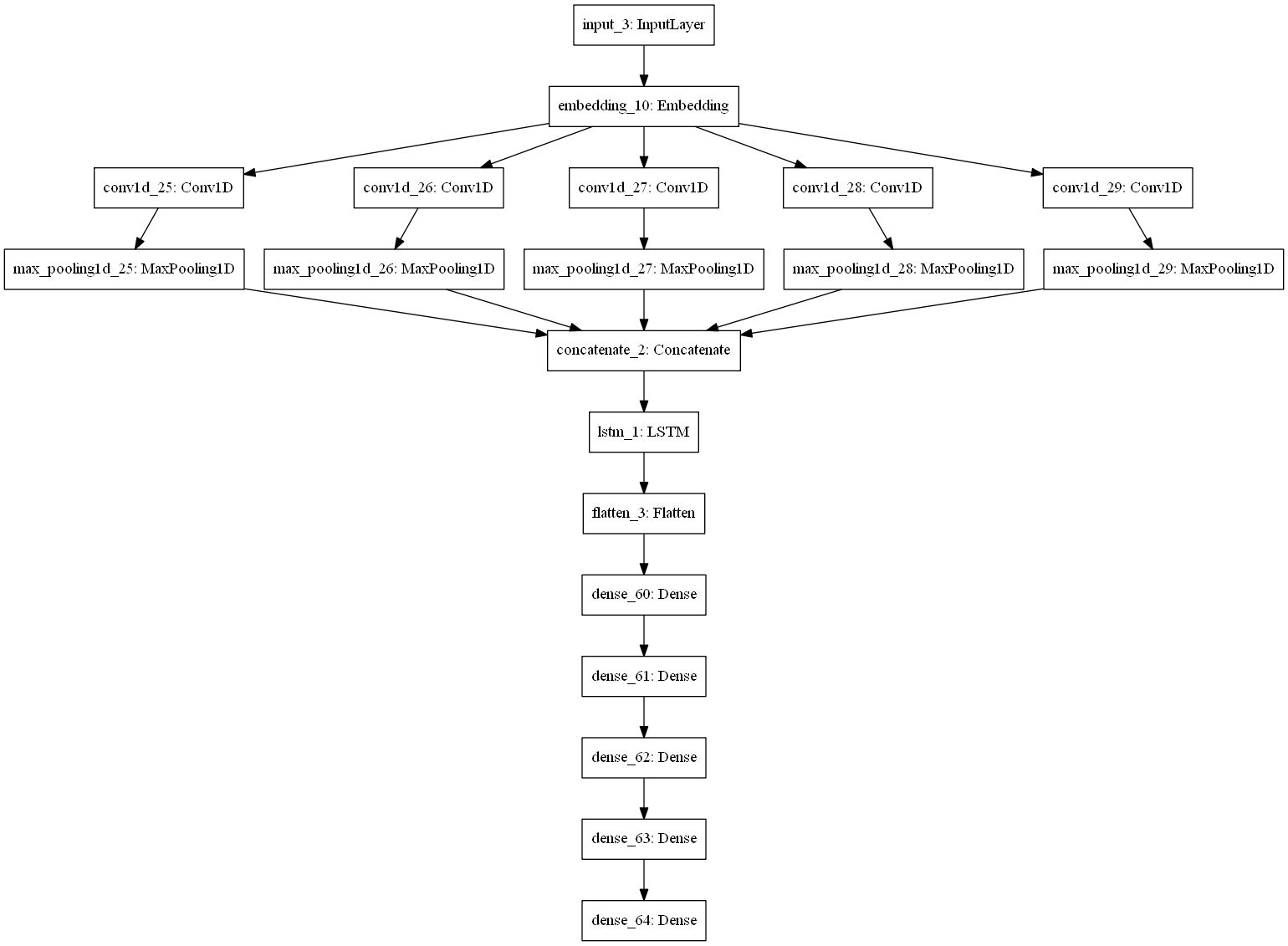
==================================================================================================

Total params: 4,602,980

Trainable params: 3,322,724

Non-trainable params: 1,280,256

**（图十八）**



**（图十九）**

Epoch 1/20

200/200 [==============================] - 696s 3s/step - loss: 1.3433 - accuracy: 0.3219 - val\_loss: 1.2505 - val\_accuracy: 0.4046

Epoch 2/20

200/200 [==============================] - 676s 3s/step - loss: 1.1748 - accuracy: 0.4390 - val\_loss: 1.0810 - val\_accuracy: 0.5006

Epoch 3/20

200/200 [==============================] - 674s 3s/step - loss: 0.9278 - accuracy: 0.5458 - val\_loss: 0.8899 - val\_accuracy: 0.5820

Epoch 4/20

200/200 [==============================] - 677s 3s/step - loss: 0.7078 - accuracy: 0.6284 - val\_loss: 0.9237 - val\_accuracy: 0.6090

Epoch 5/20

200/200 [==============================] - 675s 3s/step - loss: 0.6052 - accuracy: 0.6671 - val\_loss: 0.9310 - val\_accuracy: 0.6237

Epoch 6/20

200/200 [==============================] - 675s 3s/step - loss: 0.5338 - accuracy: 0.6927 - val\_loss: 0.9180 - val\_accuracy: 0.6355

Epoch 7/20

200/200 [==============================] - 680s 3s/step - loss: 0.5240 - accuracy: 0.6967 - val\_loss: 0.9035 - val\_accuracy: 0.6331

Epoch 8/20

200/200 [==============================] - 675s 3s/step - loss: 0.4641 - accuracy: 0.7153 - val\_loss: 1.0202 - val\_accuracy: 0.6395

Epoch 9/20

200/200 [==============================] - 676s 3s/step - loss: 0.4491 - accuracy: 0.7192 - val\_loss: 0.9688 - val\_accuracy: 0.6398

Epoch 10/20

200/200 [==============================] - 674s 3s/step - loss: 0.4538 - accuracy: 0.7192 - val\_loss: 0.9428 - val\_accuracy: 0.6425

Epoch 11/20

200/200 [==============================] - 676s 3s/step - loss: 0.4315 - accuracy: 0.7220 - val\_loss: 0.9416 - val\_accuracy: 0.6435

Epoch 12/20

200/200 [==============================] - 674s 3s/step - loss: 0.4408 - accuracy: 0.7221 - val\_loss: 0.9912 - val\_accuracy: 0.6406

Epoch 13/20

200/200 [==============================] - 672s 3s/step - loss: 0.4180 - accuracy: 0.7274 - val\_loss: 1.0413 - val\_accuracy: 0.6501

Epoch 14/20

200/200 [==============================] - 673s 3s/step - loss: 0.4090 - accuracy: 0.7300 - val\_loss: 0.9522 - val\_accuracy: 0.6389

Epoch 15/20

200/200 [==============================] - 674s 3s/step - loss: 0.4122 - accuracy: 0.7288 - val\_loss: 1.0387 - val\_accuracy: 0.6486

Epoch 16/20

200/200 [==============================] - 669s 3s/step - loss: 0.4155 - accuracy: 0.7281 - val\_loss: 1.0538 - val\_accuracy: 0.6478

Epoch 17/20

200/200 [==============================] - 671s 3s/step - loss: 0.4019 - accuracy: 0.7314 - val\_loss: 1.0336 - val\_accuracy: 0.6505

Epoch 18/20

200/200 [==============================] - 673s 3s/step - loss: 0.3970 - accuracy: 0.7339 - val\_loss: 1.0747 - val\_accuracy: 0.6414

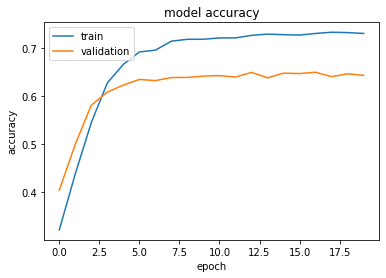
Epoch 19/20

200/200 [==============================] - 672s 3s/step - loss: 0.3984 - accuracy: 0.7331 - val\_loss: 1.1051 - val\_accuracy: 0.6472

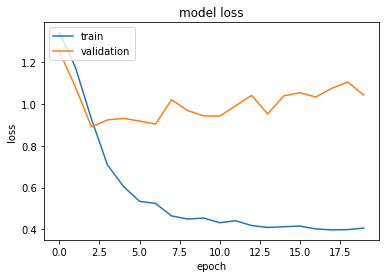
Epoch 20/20

200/200 [==============================] - 671s 3s/step - loss: 0.4058 - accuracy: 0.7314 - val\_loss: 1.0423 - val\_accuracy: 0.6440

**accuracy 73.445499**



**（图二十）**



**（图二十一）**

图十八是模型的概览，图十九是模型的结构示意图，图二十和图二十一是模型的准确度和损失值岁实验批次的增加而变化的示意图。实验结果表明增加了一层的LSTM的模型对于模型的准确度提高是有作用的，准确度又提高了2% 。在接下来的模型中选择CNN与LSTM相连接的传统的CNN模型使用加强的CNN模型。

### （5）对LSTM-TextCNN模型网络的改进

第五部分是在通过前四部分的对于整个模型的主要框架进行了确定的基础上，对于现有的模型进行改进。首先考虑的是增加concatenate层增加卷积神经网络的数量，其次是增加LSTM的复杂度，可以考虑在LSTM的基础上增加一层全连接层和一层Dropout层，实现数据特征的更好的提取。

基于这样的考量，我构造了这样的模型，首先是一层词嵌入层，接下来的一层是由6个卷积神经网络组成的concatenate层，接下来的是一层LSTM层，我增加了LSTM层的隐藏神经元的个数增加到了1000个。再下面是全连接层的神经网络和上面的模型是一样的。

Model: "model\_3"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param # Connected to

==================================================================================================

input\_4 (InputLayer) [(None, 100)] 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

embedding\_11 (Embedding) (None, 100, 256) 1280256 input\_4[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_30 (Conv1D) (None, 50, 256) 196864 embedding\_11[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_31 (Conv1D) (None, 50, 256) 262400 embedding\_11[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_32 (Conv1D) (None, 50, 256) 327936 embedding\_11[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_33 (Conv1D) (None, 50, 256) 393472 embedding\_11[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_34 (Conv1D) (None, 50, 256) 459008 embedding\_11[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_35 (Conv1D) (None, 50, 256) 524544 embedding\_11[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_30 (MaxPooling1D) (None, 5, 256) 0 conv1d\_30[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_31 (MaxPooling1D) (None, 5, 256) 0 conv1d\_31[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_32 (MaxPooling1D) (None, 5, 256) 0 conv1d\_32[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_33 (MaxPooling1D) (None, 5, 256) 0 conv1d\_33[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_34 (MaxPooling1D) (None, 5, 256) 0 conv1d\_34[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_35 (MaxPooling1D) (None, 5, 256) 0 conv1d\_35[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

concatenate\_3 (Concatenate) (None, 5, 1536) 0 max\_pooling1d\_30[0][0]

max\_pooling1d\_31[0][0]

max\_pooling1d\_32[0][0]

max\_pooling1d\_33[0][0]

max\_pooling1d\_34[0][0]

max\_pooling1d\_35[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_2 (LSTM) (None, 1000) 10148000 concatenate\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_4 (Flatten) (None, 1000) 0 lstm\_2[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_65 (Dense) (None, 256) 256256 flatten\_4[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_66 (Dense) (None, 128) 32896 dense\_65[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_67 (Dense) (None, 64) 8256 dense\_66[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_68 (Dense) (None, 32) 2080 dense\_67[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_69 (Dense) (None, 4) 132 dense\_68[0][0]

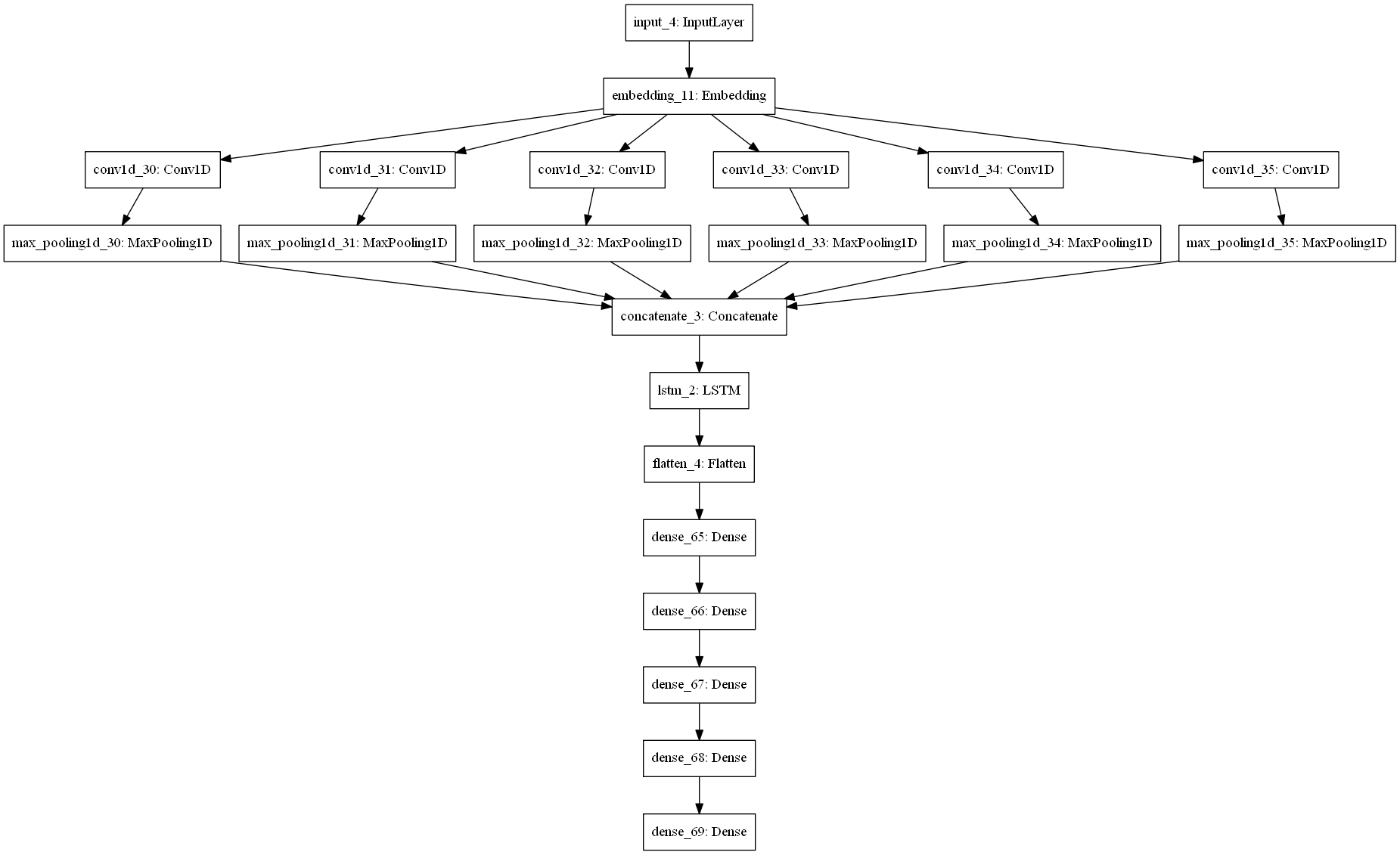
==================================================================================================

Total params: 13,892,100

Trainable params: 12,611,844

Non-trainable params: 1,280,256

**（图二十二）**

****

**（图二十三）**

Epoch 1/20

200/200 [==============================] - 629s 3s/step - loss: 1.3432 - accuracy: 0.3161 - val\_loss: 1.2313 - val\_accuracy: 0.4218

Epoch 2/20

200/200 [==============================] - 654s 3s/step - loss: 1.1175 - accuracy: 0.4627 - val\_loss: 0.9585 - val\_accuracy: 0.5353

Epoch 3/20

200/200 [==============================] - 629s 3s/step - loss: 0.8065 - accuracy: 0.5879 - val\_loss: 0.9250 - val\_accuracy: 0.5938

Epoch 4/20

200/200 [==============================] - 628s 3s/step - loss: 0.6370 - accuracy: 0.6539 - val\_loss: 0.8929 - val\_accuracy: 0.6248

Epoch 5/20

200/200 [==============================] - 633s 3s/step - loss: 0.5456 - accuracy: 0.6871 - val\_loss: 0.8607 - val\_accuracy: 0.6286

Epoch 6/20

200/200 [==============================] - 629s 3s/step - loss: 0.4959 - accuracy: 0.7039 - val\_loss: 0.9047 - val\_accuracy: 0.6311

Epoch 7/20

200/200 [==============================] - 631s 3s/step - loss: 0.4688 - accuracy: 0.7127 - val\_loss: 0.8869 - val\_accuracy: 0.6439

Epoch 8/20

200/200 [==============================] - 629s 3s/step - loss: 0.4471 - accuracy: 0.7190 - val\_loss: 0.9384 - val\_accuracy: 0.6394

Epoch 9/20

200/200 [==============================] - 629s 3s/step - loss: 0.4582 - accuracy: 0.7193 - val\_loss: 0.8930 - val\_accuracy: 0.6456

Epoch 10/20

200/200 [==============================] - 628s 3s/step - loss: 0.4158 - accuracy: 0.7287 - val\_loss: 0.9990 - val\_accuracy: 0.6451

Epoch 11/20

200/200 [==============================] - 627s 3s/step - loss: 0.4186 - accuracy: 0.7280 - val\_loss: 1.0592 - val\_accuracy: 0.6469

Epoch 12/20

200/200 [==============================] - 627s 3s/step - loss: 0.4124 - accuracy: 0.7294 - val\_loss: 1.0227 - val\_accuracy: 0.6472

Epoch 13/20

200/200 [==============================] - 626s 3s/step - loss: 0.4321 - accuracy: 0.7255 - val\_loss: 0.9963 - val\_accuracy: 0.6472

Epoch 14/20

200/200 [==============================] - 627s 3s/step - loss: 0.3994 - accuracy: 0.7321 - val\_loss: 1.0549 - val\_accuracy: 0.6501

Epoch 15/20

200/200 [==============================] - 626s 3s/step - loss: 0.3984 - accuracy: 0.7331 - val\_loss: 1.1143 - val\_accuracy: 0.6434

Epoch 16/20

200/200 [==============================] - 628s 3s/step - loss: 0.3980 - accuracy: 0.7329 - val\_loss: 1.0597 - val\_accuracy: 0.6534

Epoch 17/20

200/200 [==============================] - 628s 3s/step - loss: 0.3946 - accuracy: 0.7350 - val\_loss: 1.1460 - val\_accuracy: 0.6505

Epoch 18/20

200/200 [==============================] - 632s 3s/step - loss: 0.4175 - accuracy: 0.7291 - val\_loss: 1.0004 - val\_accuracy: 0.6511

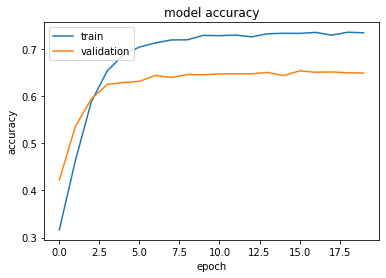
Epoch 19/20

200/200 [==============================] - 627s 3s/step - loss: 0.3917 - accuracy: 0.7353 - val\_loss: 1.1294 - val\_accuracy: 0.6494

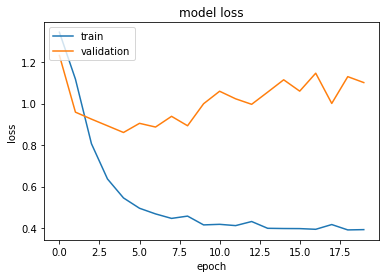
Epoch 20/20

200/200 [==============================] - 643s 3s/step - loss: 0.3931 - accuracy: 0.7342 - **val\_loss: 1.1009** - val\_accuracy: 0.6488

**accuracy 73.684502**

****

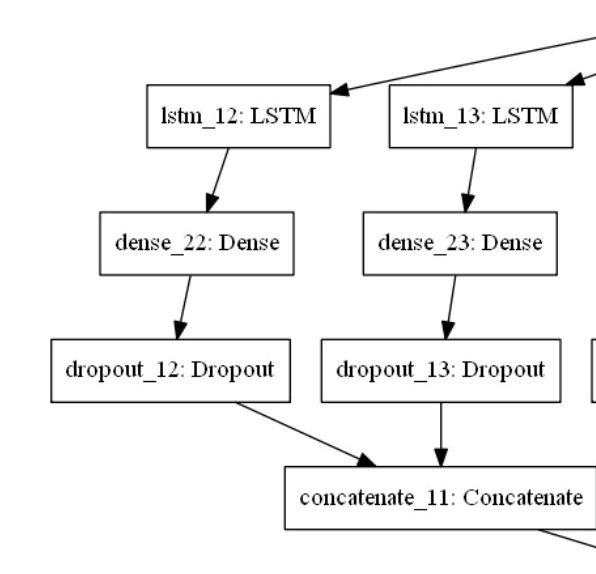
**（图二十四）**

****

**（图二十五）**

图二十二是模型的概览，图二十三是模型的结构示意图，图二十四和图二十五是模型的准确度和损失值岁实验批次的增加而变化的示意图，准确度又提高了0.5%。准确度的提升是有限的，而且可以看到稳定集的误差在训练的过程中波动起伏较大，这说明模型的训练过程中预测稳定集的能力是比较糟糕的，因此考虑对于模型进行改进。

拟打算采用这样的改进方式，改进传统的LSTM神经网络，首先将传统的LSTM神经网络后面再接上一层全连接层网络和一层Dropout神经网络防止模型在训练过程中出现过拟合的现象，然后将这样的两个神经网络放入concatenate层形成加强的LSTM神经网络，再用这三个加强的神经网络放入concatenate层实现LSTM神经网络的加强。由于神经网络的复杂性，我减少了一次训练的batch\_size到100，减少了训练的次数到10次防止模型出现过拟合的现象。



**（图二十六）**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param # Connected to

==================================================================================================

input\_4 (InputLayer) [(None, 100)] 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

embedding\_3 (Embedding) (None, 100, 256) 1280256 input\_4[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_18 (Conv1D) (None, 50, 256) 196864 embedding\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_19 (Conv1D) (None, 50, 256) 262400 embedding\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_20 (Conv1D) (None, 50, 256) 327936 embedding\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_21 (Conv1D) (None, 50, 256) 393472 embedding\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_22 (Conv1D) (None, 50, 256) 459008 embedding\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_23 (Conv1D) (None, 50, 256) 524544 embedding\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_18 (MaxPooling1D) (None, 5, 256) 0 conv1d\_18[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_19 (MaxPooling1D) (None, 5, 256) 0 conv1d\_19[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_20 (MaxPooling1D) (None, 5, 256) 0 conv1d\_20[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_21 (MaxPooling1D) (None, 5, 256) 0 conv1d\_21[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_22 (MaxPooling1D) (None, 5, 256) 0 conv1d\_22[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling1d\_23 (MaxPooling1D) (None, 5, 256) 0 conv1d\_23[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

concatenate\_15 (Concatenate) (None, 5, 1536) 0 max\_pooling1d\_18[0][0]

max\_pooling1d\_19[0][0]

max\_pooling1d\_20[0][0]

max\_pooling1d\_21[0][0]

max\_pooling1d\_22[0][0]

max\_pooling1d\_23[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_18 (LSTM) (None, 1000) 10148000 concatenate\_15[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_19 (LSTM) (None, 1000) 10148000 concatenate\_15[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_20 (LSTM) (None, 1000) 10148000 concatenate\_15[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_21 (LSTM) (None, 1000) 10148000 concatenate\_15[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_22 (LSTM) (None, 1000) 10148000 concatenate\_15[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_23 (LSTM) (None, 1000) 10148000 concatenate\_15[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_33 (Dense) (None, 16) 16016 lstm\_18[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_34 (Dense) (None, 16) 16016 lstm\_19[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_35 (Dense) (None, 16) 16016 lstm\_20[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_36 (Dense) (None, 16) 16016 lstm\_21[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_37 (Dense) (None, 16) 16016 lstm\_22[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_38 (Dense) (None, 16) 16016 lstm\_23[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_18 (Dropout) (None, 16) 0 dense\_33[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_19 (Dropout) (None, 16) 0 dense\_34[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_20 (Dropout) (None, 16) 0 dense\_35[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_21 (Dropout) (None, 16) 0 dense\_36[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_22 (Dropout) (None, 16) 0 dense\_37[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_23 (Dropout) (None, 16) 0 dense\_38[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

concatenate\_16 (Concatenate) (None, 32) 0 dropout\_18[0][0]

dropout\_19[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

concatenate\_17 (Concatenate) (None, 32) 0 dropout\_20[0][0]

dropout\_21[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

concatenate\_18 (Concatenate) (None, 32) 0 dropout\_22[0][0]

dropout\_23[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

concatenate\_19 (Concatenate) (None, 96) 0 concatenate\_16[0][0]

concatenate\_17[0][0]

concatenate\_18[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_3 (Flatten) (None, 96) 0 concatenate\_19[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_39 (Dense) (None, 256) 24832 flatten\_3[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_40 (Dense) (None, 128) 32896 dense\_39[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_41 (Dense) (None, 64) 8256 dense\_40[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_42 (Dense) (None, 32) 2080 dense\_41[0][0]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

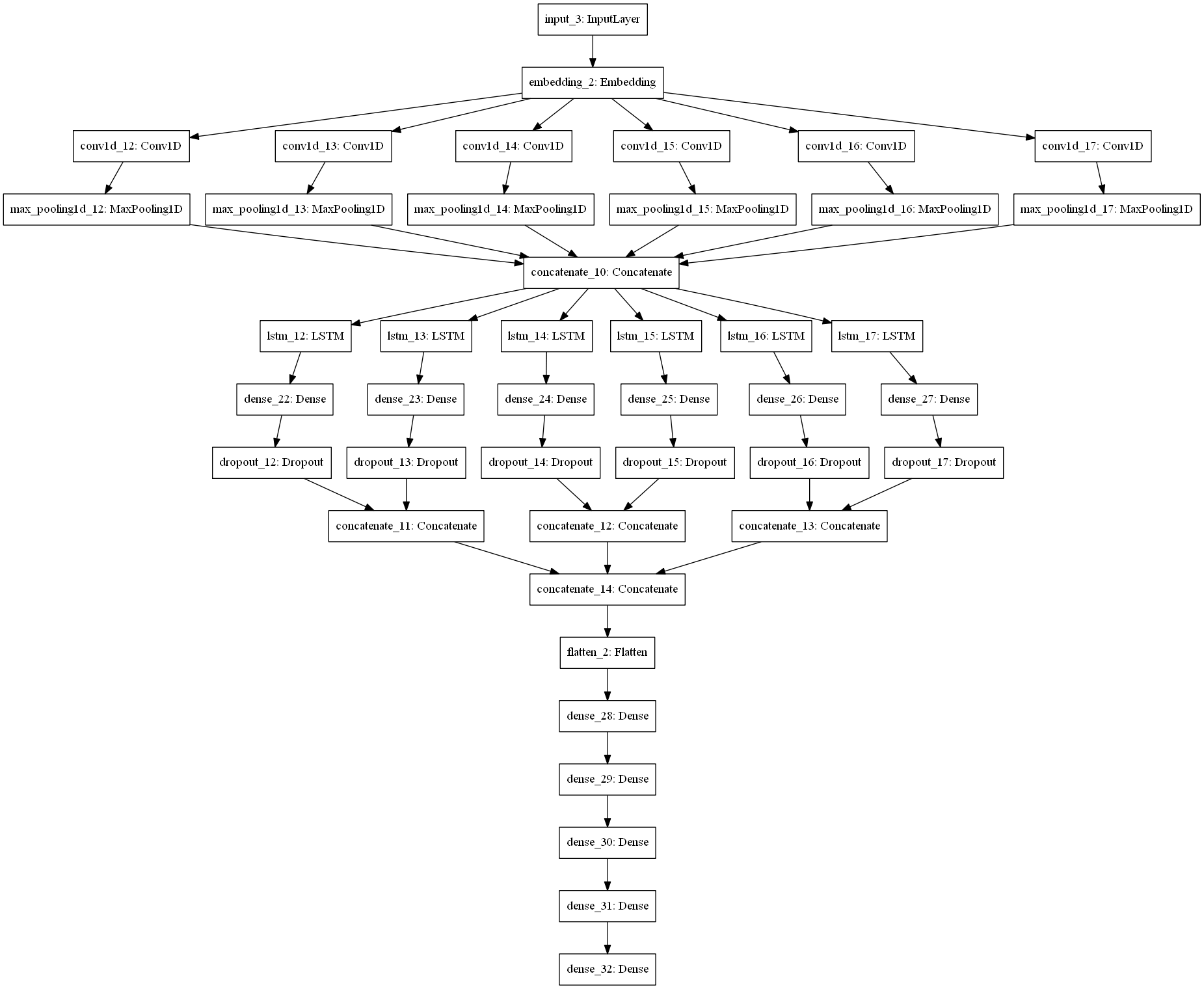
dense\_43 (Dense) (None, 4) 132 dense\_42[0][0]

==================================================================================================

Total params: 64,496,772

Trainable params: 63,216,516

Non-trainable params: 1,280,256

**（图二十七）**

**（图二十八）**

Epoch 1/10

2000/2000 [==============================] - 4814s 2s/step - loss: 1.3205 - accuracy: 0.3362 - val\_loss: 1.2070 - val\_accuracy: 0.4270

Epoch 2/10

2000/2000 [==============================] - 4990s 2s/step - loss: 1.1085 - accuracy: 0.4651 - val\_loss: 0.9736 - val\_accuracy: 0.5400

Epoch 3/10

2000/2000 [==============================] - 4737s 2s/step - loss: 0.8232 - accuracy: 0.5797 - val\_loss: 0.8979 - val\_accuracy: 0.5828

Epoch 4/10

2000/2000 [==============================] - 5231s 3s/step - loss: 0.6624 - accuracy: 0.6407 - val\_loss: 0.8744 - val\_accuracy: 0.6154

Epoch 5/10

2000/2000 [==============================] - 4366s 2s/step - loss: 0.5738 - accuracy: 0.6761 - val\_loss: 0.9100 - val\_accuracy: 0.6180

Epoch 6/10

2000/2000 [==============================] - 4361s 2s/step - loss: 0.5236 - accuracy: 0.6919 - val\_loss: 0.9048 - val\_accuracy: 0.6335

Epoch 7/10

2000/2000 [==============================] - 4381s 2s/step - loss: 0.4926 - accuracy: 0.7041 - val\_loss: 0.8991 - val\_accuracy: 0.6388

Epoch 8/10

2000/2000 [==============================] - 4638s 2s/step - loss: 0.4731 - accuracy: 0.7085 - val\_loss: 0.9970 - val\_accuracy: 0.6421

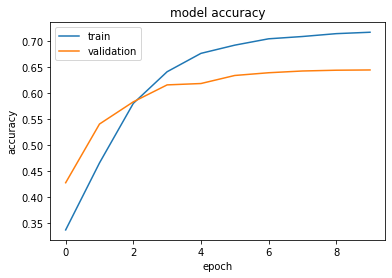
Epoch 9/10

2000/2000 [==============================] - 4606s 2s/step - loss: 0.4588 - accuracy: 0.7140 - val\_loss: 0.9295 - val\_accuracy: 0.6436

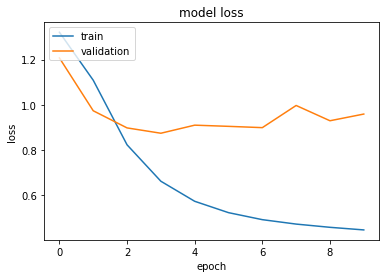
Epoch 10/10

2000/2000 [==============================] - 4625s 2s/step - loss: 0.4473 - accuracy: 0.7167 - **val\_loss: 0.9591** - val\_accuracy: 0.6441

**accuracy 72.961003**

****

**（图二十九）**

****

**（图三十）**

图二十六是加强的LSTM的模型的结构，图二十七是模型的概览，图二十八是模型的结构示意图，图二十九和图三十是模型的准确度和损失值岁实验批次的增加而变化的示意图，准确度虽然小有下降。但是可以看到模型的稳定集的损失值是有下降的趋势的而且可以看到模型在预测稳定集的损失值要比前面的模型要低，这说明这个模型在普适性上要比前面的一个模型强。因此最终选择了这个模型作为情感分类的模型。