

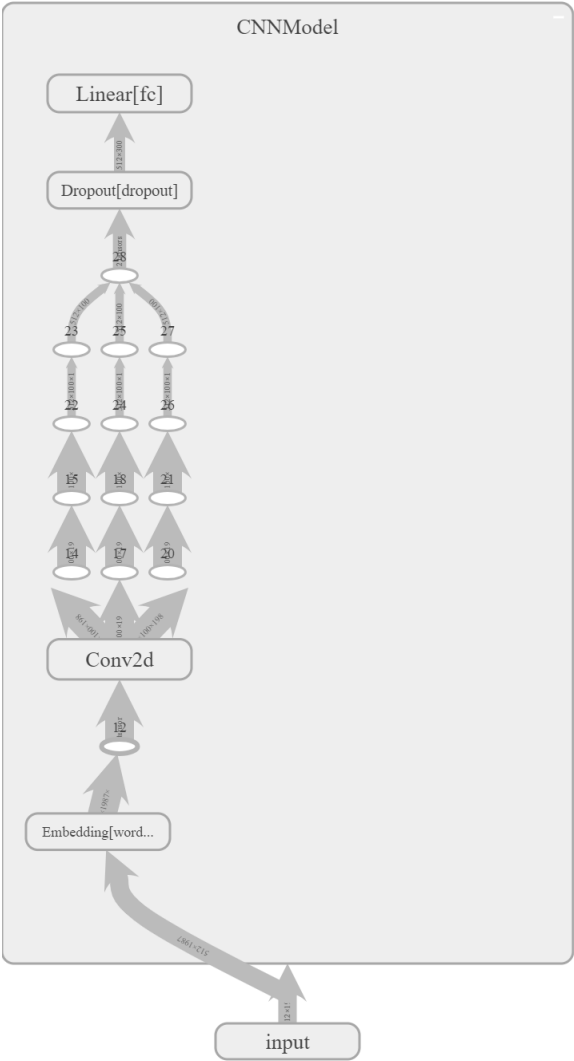
# 情感分析作业报告

2017011421 计75 顾煜贤

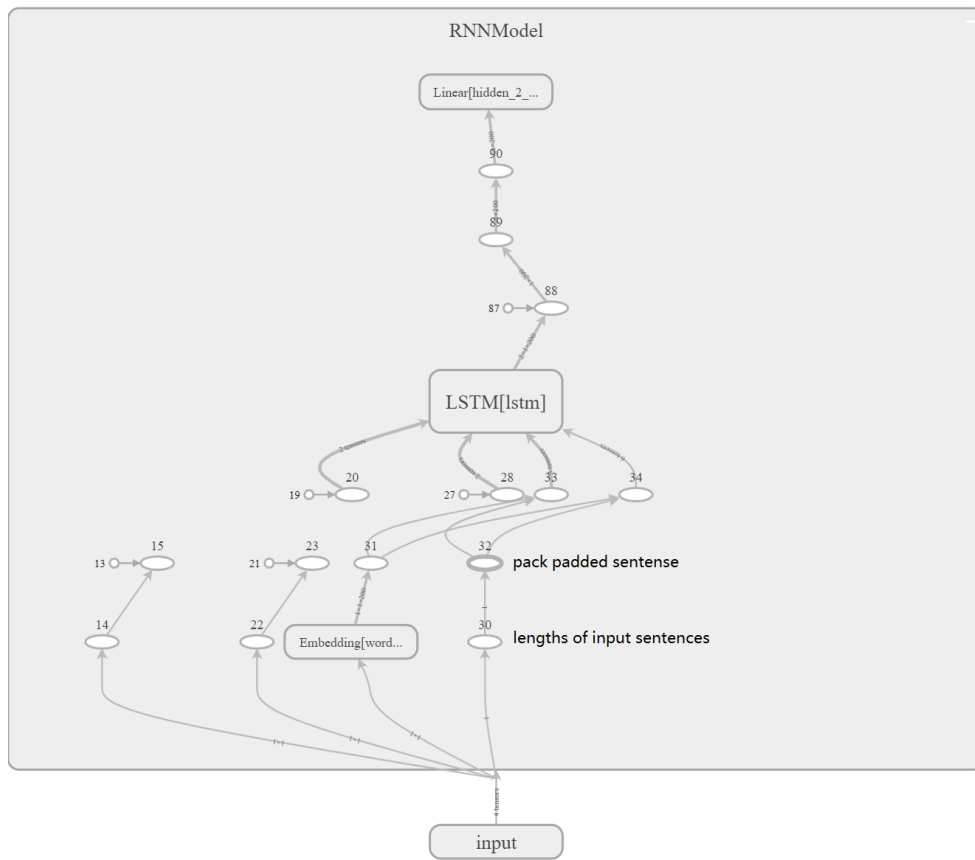
## 1 模型结构

本次作业实现了卷积神经网络（CNN），循环神经网络（RNN），以及全连接网络（MLP）。

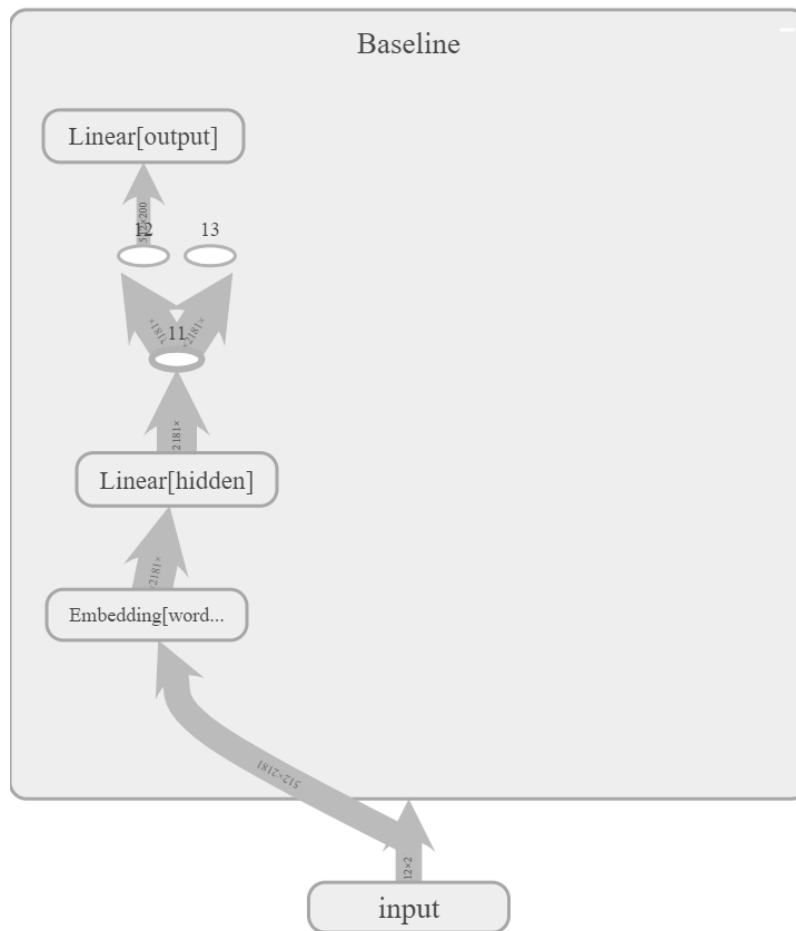
### 1.1 卷积神经网络



### 1.2 循环神经网络



### 1.3 全连接网络



## 2 实验结果

### 2.1 综合性能对比

|           | Accuracy | F-score | Correlation |
|-----------|----------|---------|-------------|
| CNN(best) | 60.5     | 0.320   | 0.590       |
| RNN(best) | 57.7     | 0.265   | 0.543       |
| MLP(best) | 59.4     | 0.354   | 0.577       |

### 2.2 CNN 各种参数及性能

|                     | CEL      |         |             | MSE      |         |             |
|---------------------|----------|---------|-------------|----------|---------|-------------|
|                     | Accuracy | F-score | Correlation | Accuracy | F-score | Correlation |
| CNN                 | 55.5     | 0.217   | 0.566       | 44.7     | 0.168   | 0.362       |
| CNN(+vec)           | 59.6     | 0.302   | 0.585       | 58.6     | 0.244   | 0.600       |
| CNN(+vec drop)      | 60.5     | 0.320   | 0.590       | 56.4     | 0.213   | 0.578       |
| CNN(+vec drop orth) | 57.5     | 0.295   | 0.571       | 53.6     | 0.190   | 0.538       |
| CNN(+vec drop norm) | 60.4     | 0.326   | 0.613       | 58.7     | 0.242   | 0.599       |

## 2.3 RNN 各种参数及性能

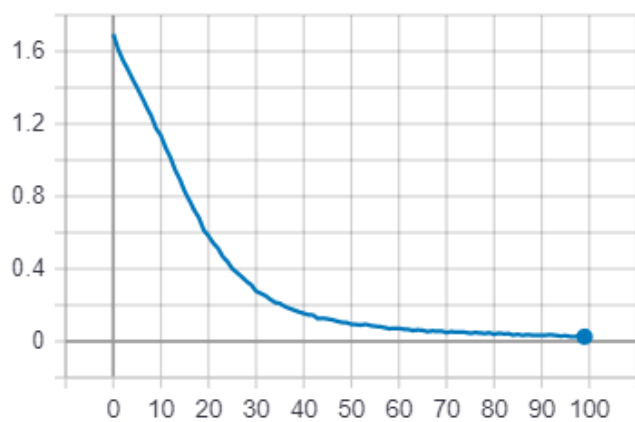
|                     | CEL      |         |             | MSE      |         |             |
|---------------------|----------|---------|-------------|----------|---------|-------------|
|                     | Accuracy | F-score | Correlation | Accuracy | F-score | Correlation |
| RNN                 | 47.7     | 0.081   | 0.415       | 47.7     | 0.081   | 0.415       |
| RNN(+vec)           | 47.7     | 0.081   | 0.415       | 47.7     | 0.081   | 0.415       |
| RNN(+vec drop)      | 56.1     | 0.234   | 0.526       | 50.0     | 0.120   | 0.504       |
| RNN(+vec drop orth) | 57.7     | 0.265   | 0.543       | 56.4     | 0.167   | 0.560       |
| RNN(+vec drop norm) | 57.2     | 0.233   | 0.537       | 57.7     | 0.200   | 0.582       |

## 2.4 MLP 各种参数及性能

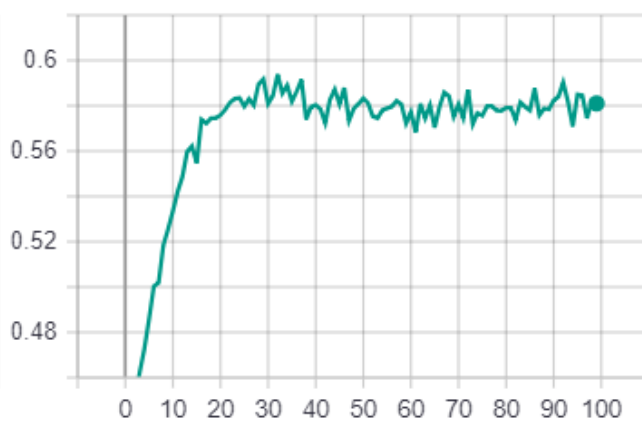
|                     | CEL      |         |             | MSE      |         |             |
|---------------------|----------|---------|-------------|----------|---------|-------------|
|                     | Accuracy | F-score | Correlation | Accuracy | F-score | Correlation |
| MLP                 | 47.3     | 0.243   | 0.501       | 45.6     | 0.262   | 0.498       |
| MLP(+vec)           | 53.9     | 0.261   | 0.538       | 51.4     | 0.199   | 0.488       |
| MLP(+vec drop)      | 59.4     | 0.354   | 0.577       | 58.6     | 0.226   | 0.609       |
| MLP(+vec drop orth) | 58.5     | 0.277   | 0.593       | 56.6     | 0.245   | 0.541       |
| MLP(+vec drop norm) | 58.7     | 0.317   | 0.595       | 53.6     | 0.254   | 0.485       |

## 2.5 各指标在训练过程中的变化

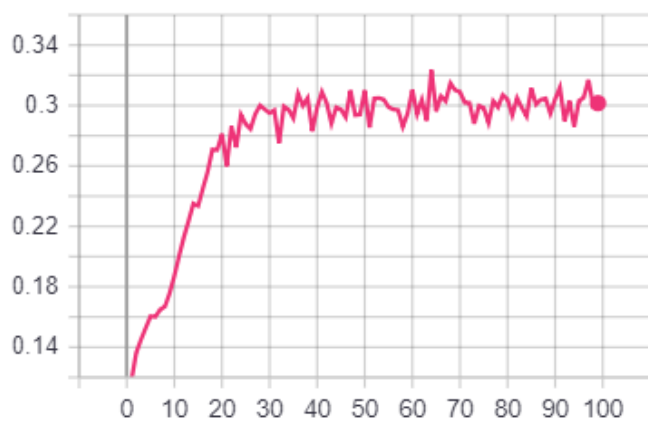
### 2.5.1 CNN



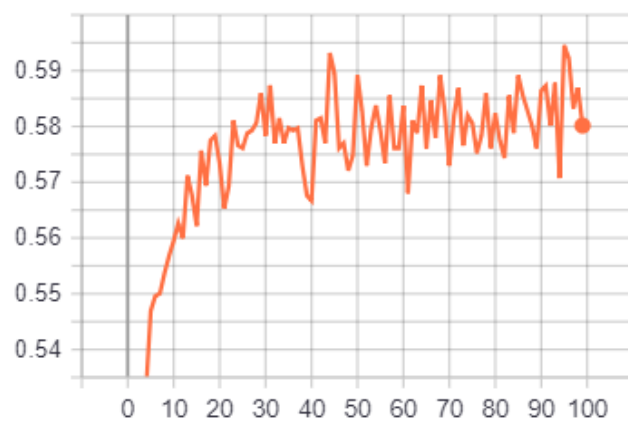
Training Loss



Covariance

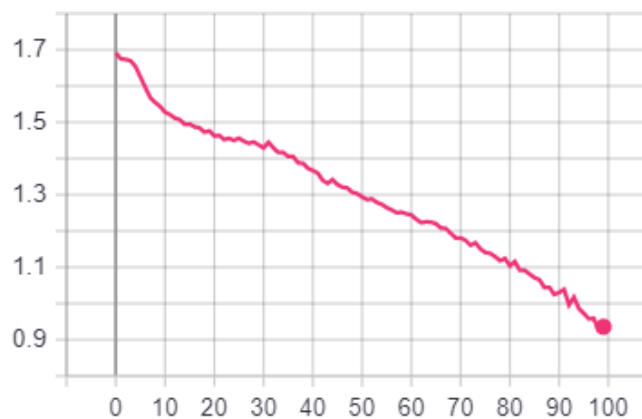


F1\_score

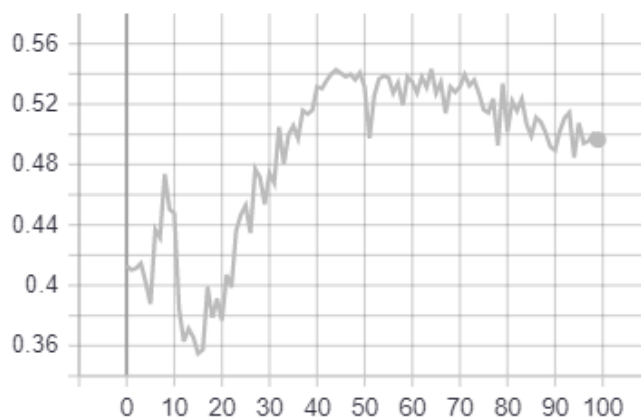


Accuracy

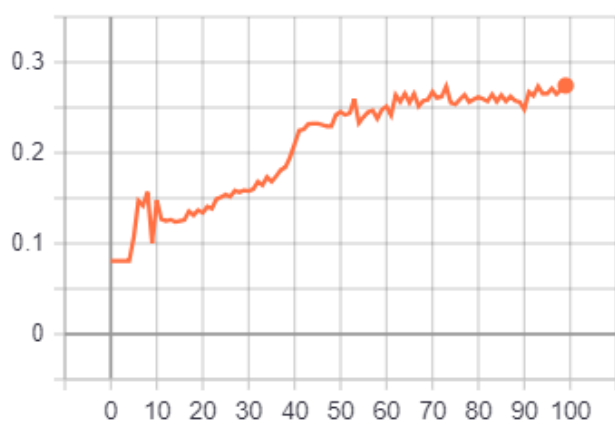
## 2.5.2 RNN



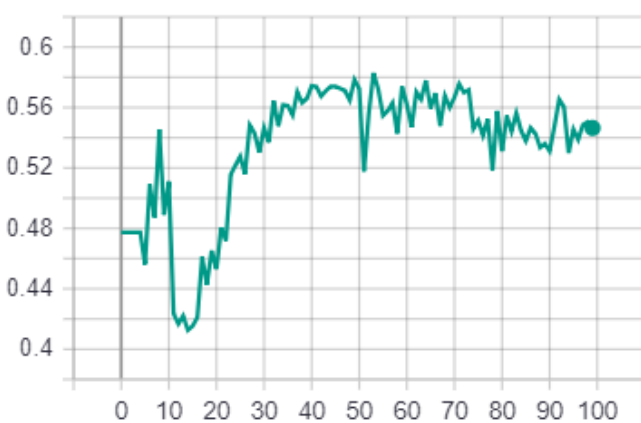
Training Loss



Covariance

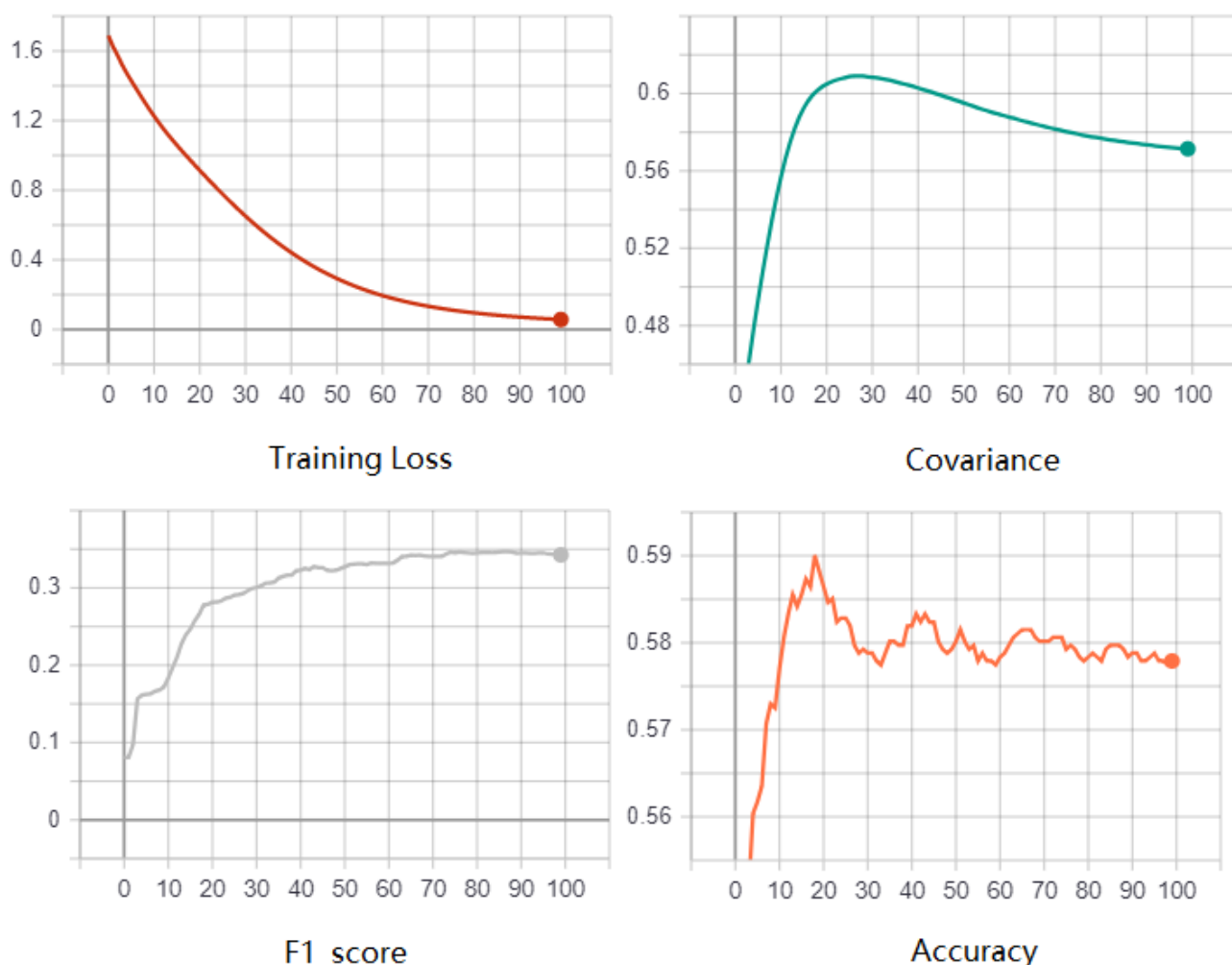


F1\_score



Accuracy

### 2.5.3 MLP



### 3 不同参数结果比较

#### 3.1 dropout

由上面实验数据可知，各模型在加入dropout后性能都有所提升，并且从test loss随时间的变化曲线中来看，未加dropout的模型在test loss值较早上升（RNN甚至一直不上升），此时出现过拟合现象。实验中均取dropout=0.5。

#### 3.2 初始化

不同模型初始化的影响不同。使用特定的初始化（正交初始化或高斯分布初始化）有利于RNN模型取得更高的准确率，但是对于MLP和CNN模型都会降低准确率。

#### 3.3 batch size

实验中，batch size 主要影响训练速度。当batch size为1时每一个epoch训练较快，但是最终可以用较少的epoch达到较好的效果。相反，batch size增大会让每一个epoch训练加快，但是同时也会需要更多epoch达到最好的效果。

#### 3.4 词向量维数

下面列出不同词向量维数对结果的影响。可见，当词向量维数在100量级时，对结果的影响不大（RNN模型在不加载词向量的情况下一直停留在47.7，即情感全选“愤怒”的状态）。2中的实验里，对于未加载预训练词向量的情况，词向量维数取200。

|     | 10   | 50   | 100  | 200  | 300  |
|-----|------|------|------|------|------|
| RNN | 47.7 | 47.7 | 47.7 | 47.7 | 47.7 |
| CNN | 48.3 | 47.3 | 52.4 | 55.5 | 54.9 |
| MLP | 50.8 | 51.3 | 53.6 | 53.9 | 54.0 |

### 3.5 预训练词向量

本次作业采用了sogou新闻的[sgns.sogou.bigram](#)词向量，词向量维数300，未在词向量中出现的词用随机向量代替。对比不同词向量的效果，此种词向量的效果最好。

### 3.6 loss

由上述数据可见Cross Entropy的效果好于Mean Square Error。可能的原因是，该数据集对于情感分布的标注并不好，因为有些文章只有一种情感有人标注。因此，很难用模型拟合出其真是分布。但是，这样的标注对于只用考虑最大值的Cross Entropy loss已经足够，因此后者跟适合此次作业的数据集。

## 4 不同模型效果比较

1. 评价指标：CNN模型不论在Accuracy、F1\_score、Correlation 四个统计指标上都达到了最高的准确率，MLP次之，RNN再次。
2. 训练过程：CNN模型和MLP模型在训练中都有明显的loss下降，并且准确率上升的过程，但是RNN模型一开始大约5个epoch中准确率始终保持在0.477，观察输出发现，此时的模型将所有情感都输出为“愤怒”，因此F1\_score和Correlation都非常低。在第6个epoch时，准确率才开始上升。MLP模型用很短的时间达到了比较好的效果，但是此后过拟合现象比较严重，Accuracy, Covariance 在第20个epoch后开始明显下降。
3. 训练时间：CNN模型训练100轮大约用时60min，MLP模型大约用时40min，RNN大约用时120min，可见在本次作业的数据集上，MLP可以用最少的时间达到较好的效果。

## 5 问题思考

### 5.1 实验训练停止时间

实验训练停止时间主要根据模型在验证集上的表现判断。如下图所示，一般情况下，随着训练loss的下降，验证集的loss以及各种评分指标会被逐渐优化。但是，当训练loss下降当一定程度时，验证集上的loss会开始上升，这时产生了过拟合现象。此时，训练loss并未完全收敛。

因此，训练停止时间可以选择“early stop”策略，即当验证集上loss开始回升，或者基本不变时，就让训练停止。

但是，我的实现方式有所不同。我的实现方式为：让模型训练足够长时间（固定epoch），直到训练的loss收敛，每轮保存当前为止最好的模型。因为本次作业训练一个epoch的时间大约只有不到1min，因此训练成本实际上非常小，可以做更加充分的训练。

下面比较两种方法的优缺点：

- 通过验证集调整：



- 优点：节省计算资源，当模型一个epoch训练时间较长或者模型文件较大不方便频繁读写时比较节省时间
  - 缺点：在验证集上收敛的时候不方便确定停止的时间，尤其是当模型loss波动较大时，有可能训练不充分就停止了。
- 固定训练次数：
  - 优点：可以充分训练模型，在训练速度较快的情况下，可以将训练轮数设计得很大，从而保证模型效果的最佳状态。
  - 缺点：比较耗费计算资源，当一个epoch训练时间比较长时，训练轮数不好确定，因为如果轮数过小则训练不充分，轮数过大则耗费资源。并且，随着各种参数、数据集的改变往往收敛速度不尽相同，此时训练轮数更加不好确定。

## 5.2 实验参数初始化

本次作业分别实现了全0初始化，高斯分布初始化，正交初始化。在网上查阅资料，一般全0初始化会使得较复杂的模型遇到较多问题，如梯度消失，等等，因此对于比较复杂的模型（如RNN），应该采用不同的初始化方式。

- 高斯分布初始化：高斯分布初始化用于替代全0初始化，在需要使用比较小的随机数进行初始化的时候使用。
- 正交初始化：正交初始化为将权重矩阵初始化为正交矩阵，一般用于避免梯度消失与梯度爆炸现象，在RNN中使用较多。经过一些推导，可知RNN的loss对参数的倒数正比于参数矩阵特征值的 $t$ 次方（ $t$ 为经过的RNN Cell的个数）。因此，当 $t$ 很大时，若特征值 $\lambda > 1$ ，则发生梯度爆炸，很难收敛，若 $\lambda < 1$ ，则发生梯度消失，之前Cell中的信息无法传递过来。所以，若将权重矩阵初始化为正交阵，由于正交矩阵的特征值为1或-1，使得训练过程中不会一开始就发生梯度爆炸或者梯度消失的问题。

## 5.3 防止过拟合

本次实验中主要采用了两种方法防止过拟合：“early stop”策略和“dropout”层。

- “early stop”策略在5.1中已经分析过，实际上其本质是将模型在与训练集不相交的验证集上的表现作为评判模型的标准，从而防止模型在训练集上的过拟合。
- dropout层：dropout层的实验结果在2中给出。可见dropout层在一定程度上防止了模型的过拟合。

## 5.4 CNN、RNN、MLP优缺点

- CNN:
  - 优点：训练速度适中，收敛较快。对于比较短的文本，各种情感词汇分布比较集中，CNN效果更好，因为CNN的卷积核相当于一个窗口，每次提取一个文本序列一段长度的特征，如果情感词汇分布较为集中，CNN可以根据它们提取出更强的特征。
  - 缺点：对于较长的文本，因为每次CNN的卷积核总是对一段区域做卷积，因此较难根据两个相距较远的情感词汇提取出文本的特征。
- RNN:
  - 优点：对于较长的文章序列，RNN的效果会更好。因为RNN对于整个输入序列处理，每一次输入下一个cell的数据实际上包含了之前所有数据的信息。因此即使两个情感词汇（或其他特征）相隔较远，RNN也能通过cell之间传递的数据将其“记住”。
  - 缺点：训练速度较慢，而且收敛比较慢。在较短的文章上表现不如RNN。
- MLP:
  - 优点：训练速度很快。模型十分简单，实现方便，并且效果不差。
  - 缺点：本实验中没有体现出明显的缺点，这一点比较反直观，因为多数文献表示全连接神经网络效果应该是最差的。但本实验中MLP甚至比RNN表现还要好。

经过分析，原因可能是数据集的问题。本次作业数据集新闻量为2000+，但是一些常用的这类新闻分类数据集数据量都在几十万量级，如THUCNews数据集（76万），今日头条数据集（38万）。因此，在小数据集的情况下，一些比较复杂的网络可能并不能体现出其优势。

## 6 心得体会

---

1. 这次作业让我更加熟悉了pytorch的使用，并且将整个机器学习领域的实验过程完整地走了一遍，从处理数据集到编写模型代码、训练测试代码，再到调参、选择测试样例、撰写实验报告。虽然之前这些过程都做过，但是这次是第一次完整地走下来。
2. pytorch框架我之前会用一点，但是这次实验让我学会了更多pytorch的特性，比如使用TorchText处理数据，使用tensorBoard绘制实验曲线、模型结构，参数初始化、更多的loss函数等。
3. 这次实验让我对RNN与CNN模型有了更深的理解，原来我一直觉得CNN模型只能用在CV领域，RNN用在NLP领域，因此在NLP领域，RNN效果一定比CNN好。但是这次实验不仅让我看到了CNN在NLP领域的一个应用，还让我明白，RNN和CNN都有不同的适用范围，不一定哪个效果一定更好，甚至在特定的情况下，全连接网络的效果还是最好的。