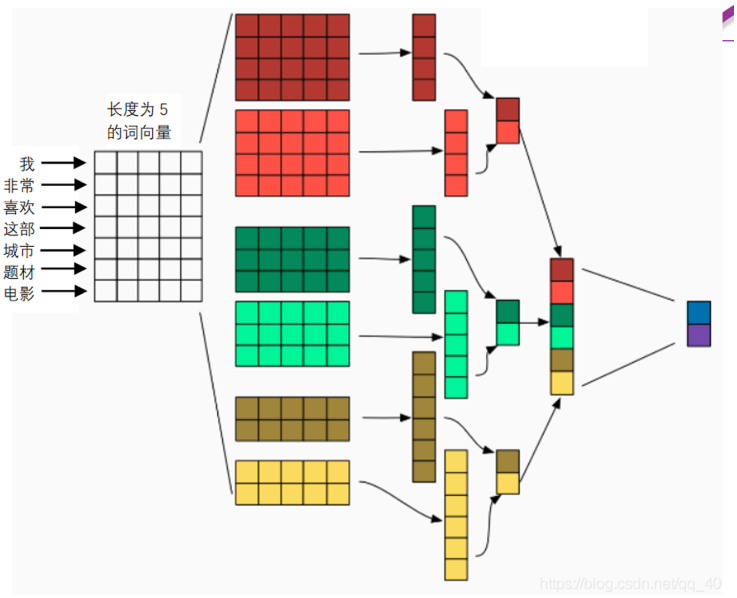
文本情感分类实验报告

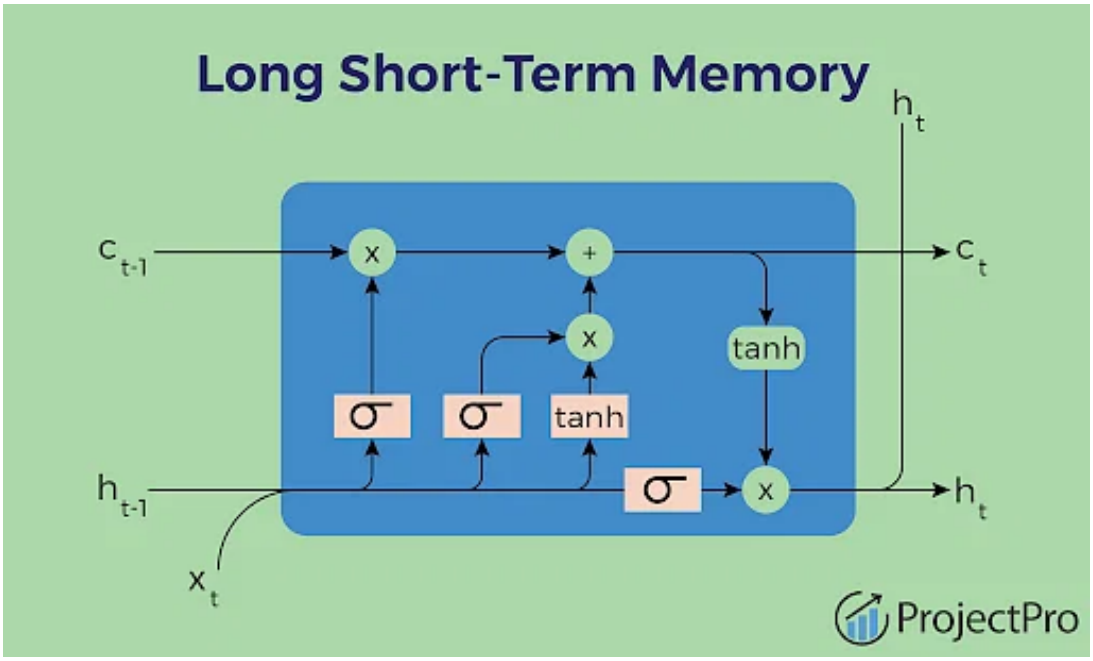
1 模型结构图

1.1 卷积神经网络



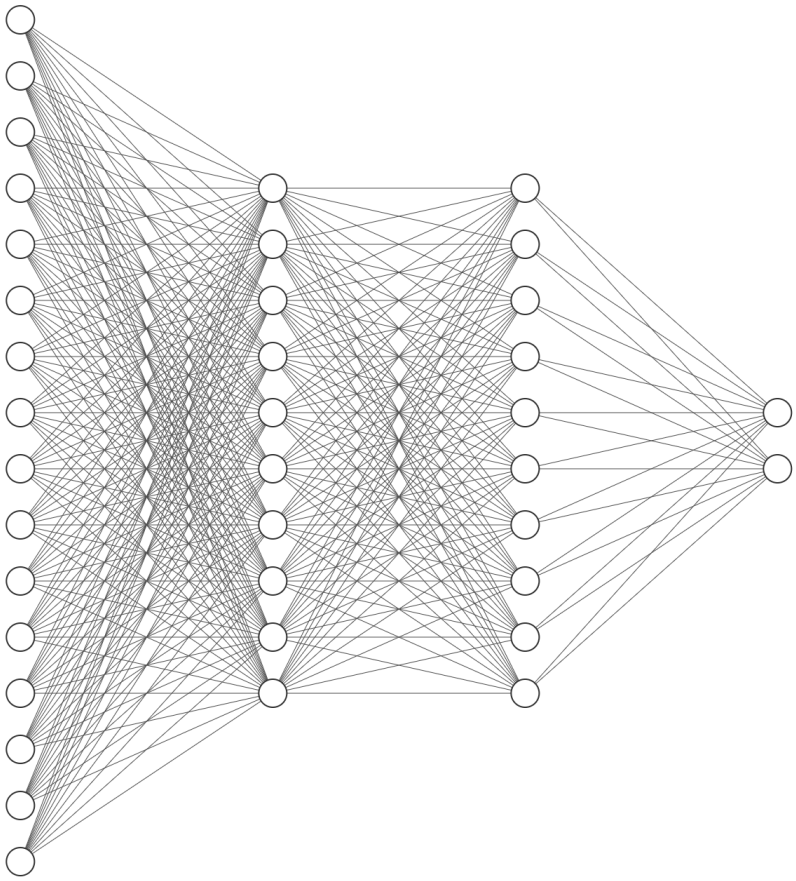
- 流程：将输入的词序号转换为词向量后，拼接为一个二维平面，使用宽度分别为3，4，5的卷积核进行卷积，产生100个通道，然后每个相同大小卷积核通道进行最大池化，拼接后与最后的二值输出进行全连接。

1.2 循环神经网络（LSTM）



- 可以对序列进行处理，序列中每个元素经过如图的LSTM单元后将信息传递给下一个单元。

1.3 全连接神经网络



- 将输入数据拼接后通过两个隐含层与二值输出全连接，并添加dropout。

2 实验结果

2.1 CNN

- 参数：epoch = 6, 卷积核宽度分别为3，4，5，各100个通道
- 准确率 85.637%，f1-score 0.855

2.2 LSTM

- 参数：epoch = 2, 其余默认
- 准确率 83.469%，f1-score 0.825

2.3 MLP

- 参数：epoch = 1，参数默认
- 准确率 83.740%，f1-score 0.832

3 参数分析

3.1 learning rate

- 在 CNN 中实验，最终采用的数值为0.005，调为0.001或更低后损失函数的收敛会变慢，拟合速度变慢，大约15次以上，调为0.1或更高后，每次迭代中损失函数会变得过大，且难以收敛，

3.2 epoch

- 轮次越多拟合效果越好，但数值较大后边际收益降低，且可能过拟合，如lstm，训练10轮后在
- CNN 模型中， epoch = 15时 测试集准确率变为86.18%，相比数值为6时上升0.5%左右。

3.3 dropout

- 能够通过随机关闭神经元的方式，降低过拟合
- 但如果数值过高，会降低准确率

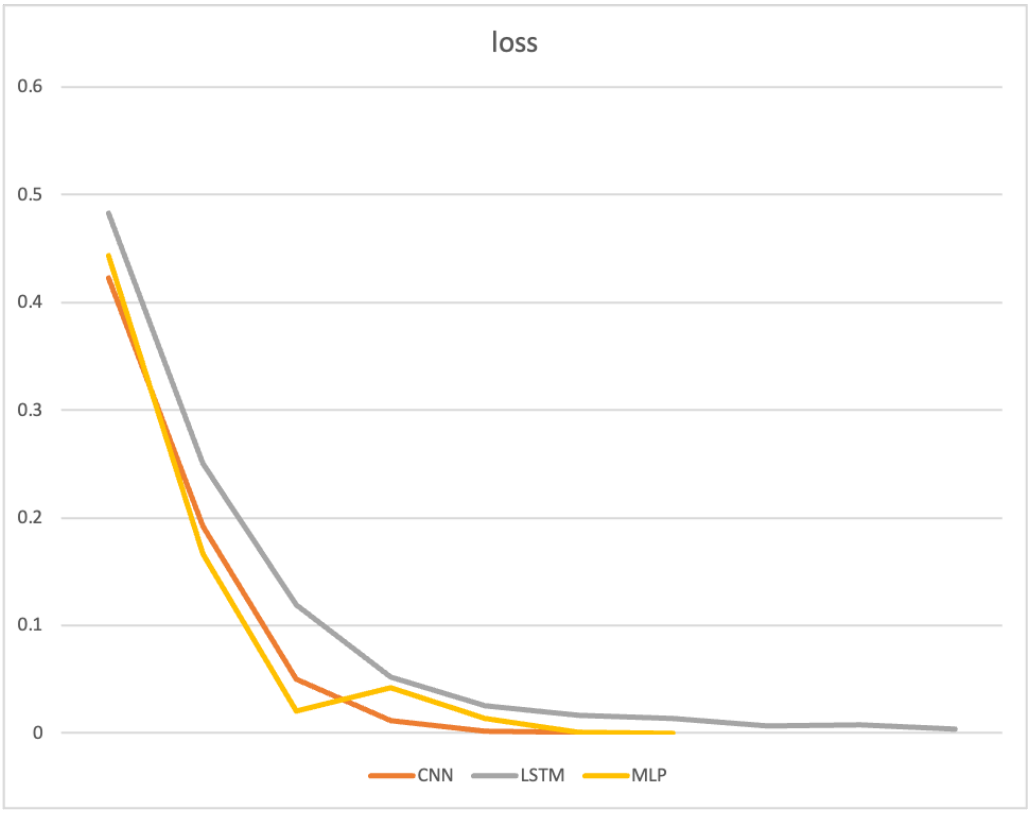
3.4 hidden dim

- 数值较高时会显著降低训练速度
- 数值分别为50，100时，100的准确率比50高出约2%

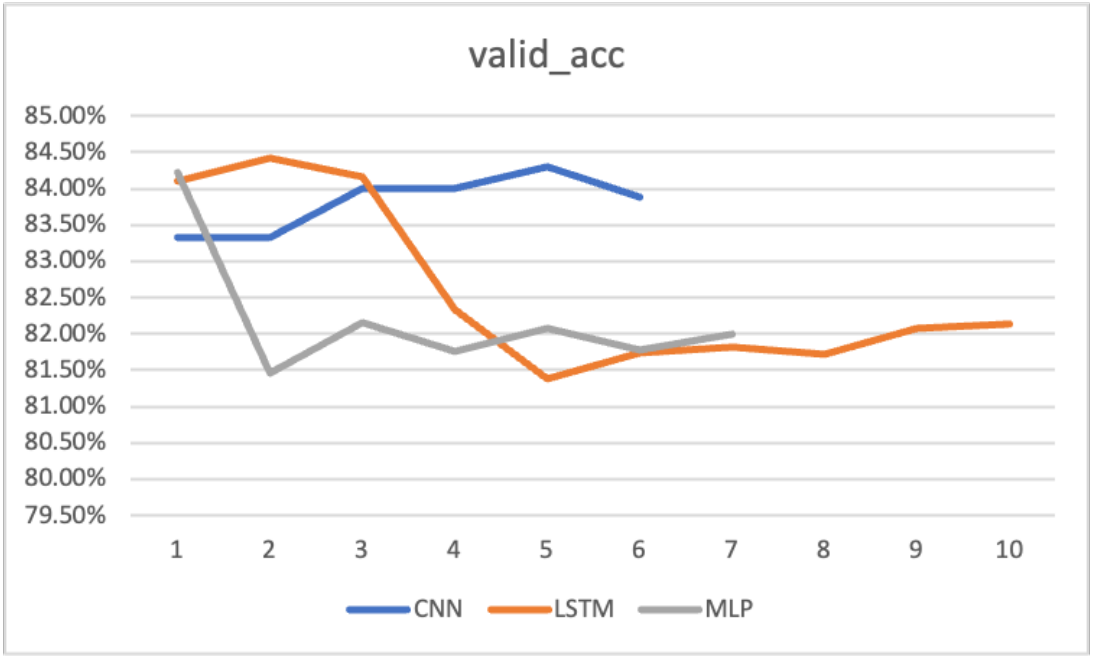
4 模型对比

4.1 训练过程

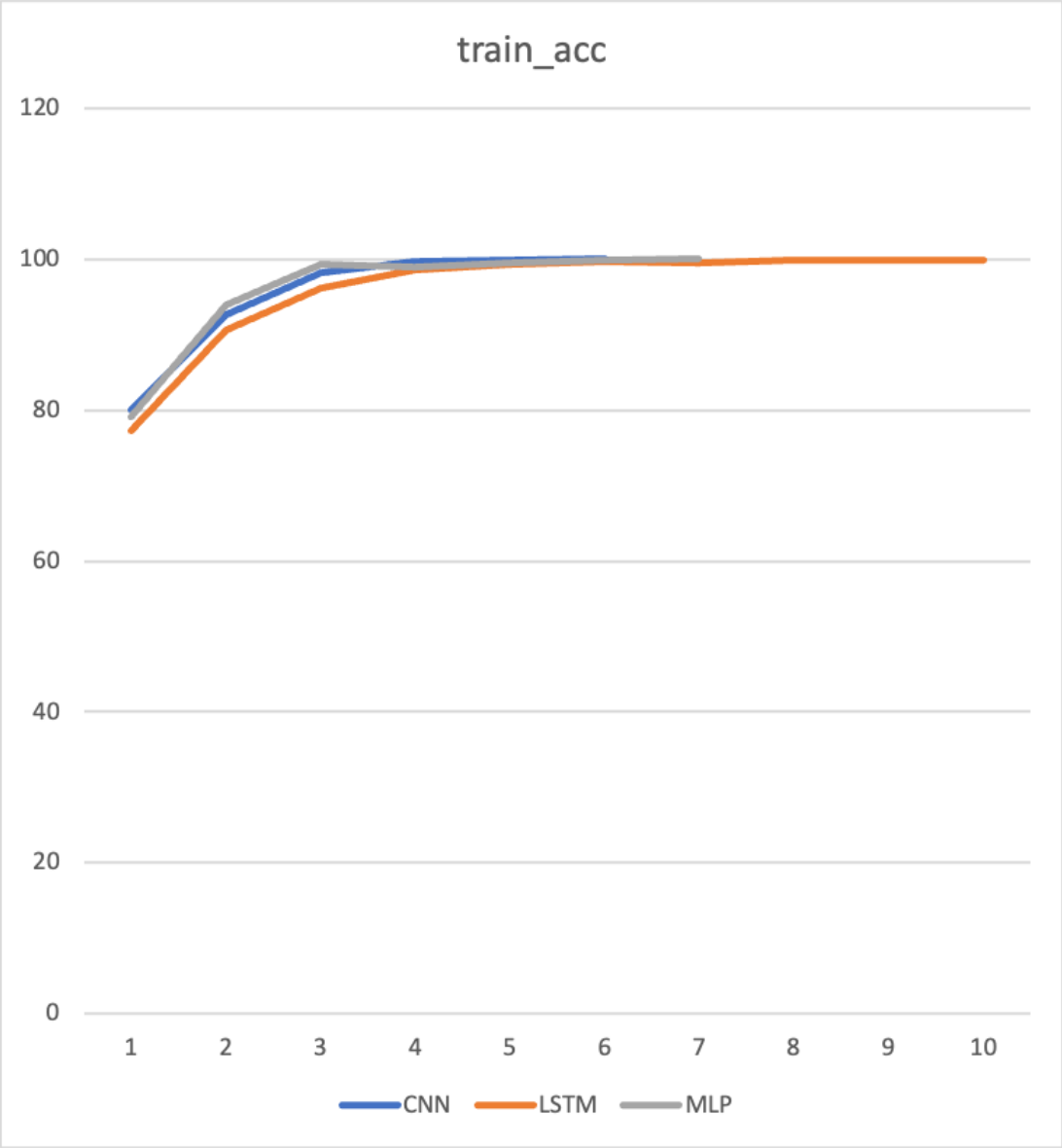
下图为各个变量在训练过程中的变化



- 从损失函数的变化可以看出，cnn收敛快且稳定，而全连接虽然收敛较快，但会出现波动，LSTM收敛较慢



- 从图中可以看出验证集上的准确率一直在波动，但总体来说CNN准确率较高，并且可以观察到LSTM和MLP在loss下降的同时，验证集准确率也发生了下降，说明出现过拟合



- 训练过程中，对训练数据的拟合速度基本类似。

5 问题思考

- Q: 实验训练什么时候停止是最合适的？简要陈述你的实现方式，并试分析固定迭代次数与通过验证集调整等方法的优缺点。

  A: 当损失函数几乎不再下降的时候停止训练时最合适的。我的实现方式是指定最大迭代次数，每次迭代时输出loss的改变状况与验证集的准确率，选择loss下降变慢，或者验证集准确率下降时停止训练。固定迭代次数较为方便，但是次数较少时无法达到效果，次数较多时浪费资源，并且容易出现过拟合，通过验证集调整能更好地防止过拟合。

- Q：实验参数的初始化是怎么做的？不同的方法适合哪些地方？

A：初始化采用了pytorch所默认的kaiming_uniform初始化方式。零初始化容易导致模型对称性问题，大部分时候不使用；kaiming_uniform适用于ReLU等非饱和激活函数的网络结构；Xavier正态分布初始化适合tanh激活函数。

- Q：过拟合是深度学习常见的问题，有什么方法可以方式训练过程陷入过拟合?

  A：可以通过：

  1. 数据增强，如每次训练时随机变换或者删除一些原始数据
  2. 检测验证集准确率，在验证集准确率降低时停止训练，从而防止过拟合，并且保留较好的性能

- Q： 试分析CNN，RNN，全连接神经网络（MLP）三者的优缺点。

  A：如前文所说，CNN较为稳定，没有明显的过拟合，并且准确率最高；LSTM准确率最快达到峰值，但也很快出现了过拟合，后续虽然损失函数仍在下降，但准确率没有提高；MLP在一轮训练后准确率立刻下降，发生过拟合，但是训练速度较快，准确率与RNN相近。

6 心得体会

本次实验是我第一次接触机器学习和pytorch，实验过程中我学到了pytorch框架的基础用法，学习了tensor的一些变换，了解如何自定义一个神经网络模型。同时，在调整参数的过程中，发现原来不能一味地追求训练数据的准确率，有时候虽然训练数据准确率提高，但是出现了过拟合，在验证集上的准确率反而会降低。