1. 概述

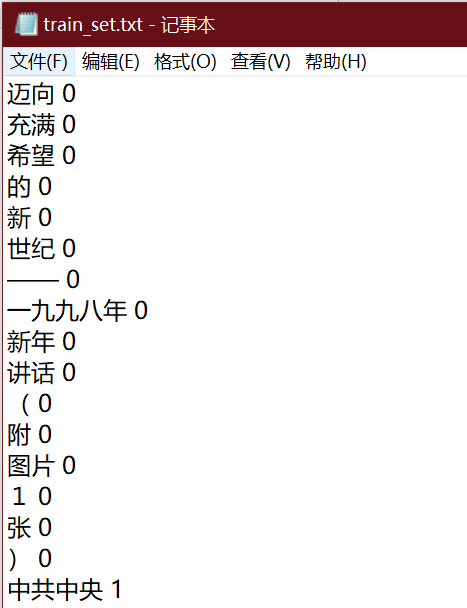
本次作业对1998年1月的人民日报做了命名实体识别，识别对象为机构名，使用预训练好的词向量模型，对词进行BIO三分类，B表示机构实体的首词，I表示机构实体的接续词，O表示非机构实体，使用了双向LSTM模型，并使用softmax作为输出层激活函数。数据集来自北京大学计算语言学教育部重点实验室 (pku.edu.cn)，使用时先进行了预处理，梯度下降使用了小批量梯度下降，最终测试集的f1-measure 达到0.729，比第二次作业有显著提升。

1. 模型原理

此次作业依然使用预训练好的的词向量模型，维数为50维。我们将10个词拼接成一个句子作为样本，每次5个样本为1个batch作为LSTM的输入。LSTM层为双向，2层隐藏层，隐层的维数为100维，对于BIO三种分类，我们使用一个的线性层将输入样本转化成长度为3的输出向量，再使用softmax函数将输出向量转化为和为1的概率分布，完成前向传播的过程，其中概率最大的类别便是预测的结果。然后再通过输出的概率向量与目标向量计算loss，loss\_function采用cross\_entropy函数,再通过反向传播进行梯度下降更新参数，便完成了一次训练。对样本集所有数据遍历一次时为1个epoch，本次共训练20个epoch。

1. 实验流程
2. 处理原始数据集

将原始数据集命名为people\_daily.txt，取19980101-19980120为训练集，19980121-19980125为验证集，19980126-19980131为测试集。调用source目录下collect\_set.py即可完成分割，得到train\_set.txt，validation\_set.txt，test\_set.txt。在分割时，对单个词作为的实体，若为机构名，标为1，否则标为0，对于多个词组成的实体，若为机构名，则将[]中的第一个词标为1，其余词标为2，否则都标为0。处理后的数据集如图所示



1. 获取词向量

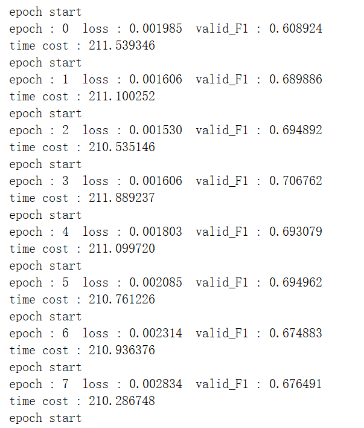
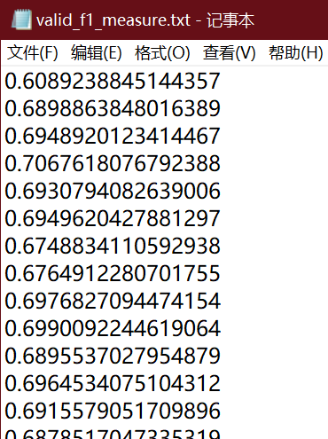
本次使用的词向量为网络资源，已预先训练好，资源链接

<https://github.com/jiesutd/LatticeLSTM>

在使用过程中，数据集中存在但词向量集中没有的词标为“-unknown-”。

1. 训练模型

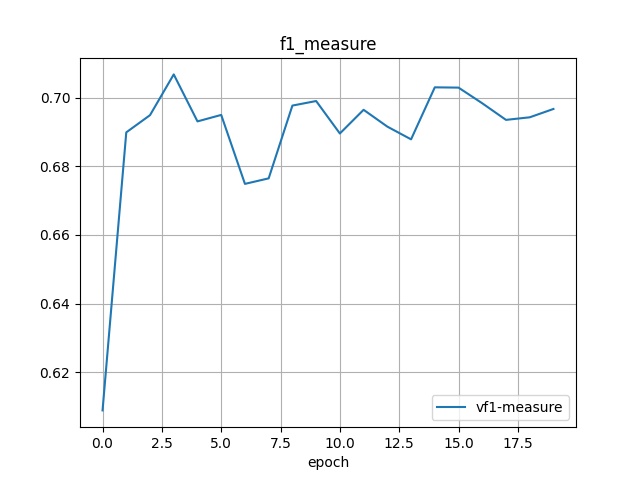
调用main.py，将词向量，训练集，验证集读入，存入自定义数据集MyDataSet中，MyDataSet继承自torch.nn的TensorDataset类，代码中自定义了数据接口，配合DataLoader将数据集分割成batch大小的样本集。同时定义了MyLSTM模型，使用nn.Embedding将词向量表存入，创建所需网络层，每次将MyDataSet传来的样本集映射到词向量，再用网络层进行前向传播。若首次进行训练，则使用初始化的随机网络层，否则从protected data/lstm.pth中读取神经网络参数，并在训练后存入protected data/lstm.pth中。在训练时使用小批量梯度下降，使用Adam优化器，学习率默认0.001。一次epoch训练完成后，计算训练集上的loss以及验证集上的f1-measure，并写入文件储存。loss.txt和valid\_f1\_measure.txt以及程序输出nohup.out如图所示

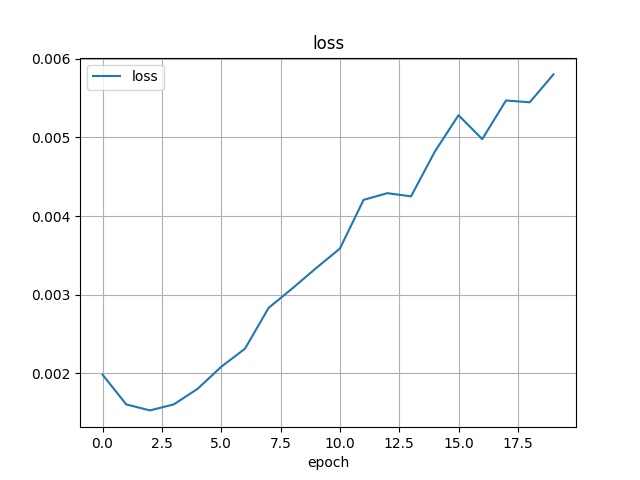


1. 结果分析

与上次作业的700个epoch相比，此次仅训练了20个epoch，模型的loss先快速下降而后逐步上升，验证集的f1-measure在第1个epoch即达到0.608，随后快速上升到0.70附近并小幅上下波动。测试集的f1-measure达到了0.729。

模型训练过后做出的预测效果较上次有显著提升，也符合采用双向LSTM模型的预期。



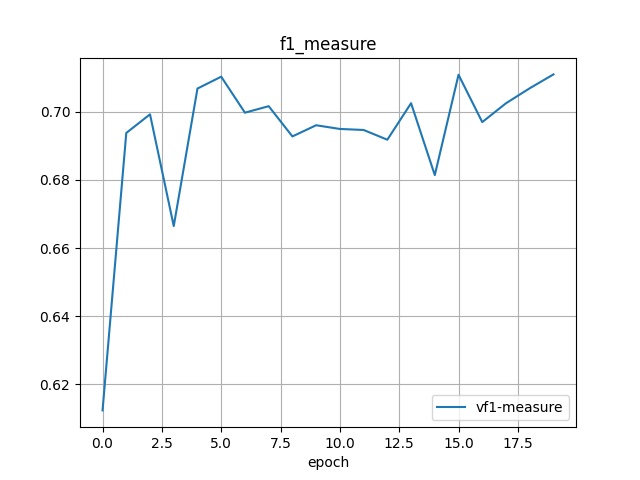


1. 总结与收获

这次作业写代码所用时间不多，仅需要在上次的模型中更改MyDataSet的接口和神经网络模型，但是花了很多时间调整batch，学习率等参数，并尝试了两种不同的优化器SGD和Adam。

在尝试中，我发现超参数的选取对程序的训练速度和最终效果影响是显著的，我对SGD选取了两种不同的参数搭配，对Adam选取了5种，并在最后一种中加上了学习率的指数衰减optim.lr\_scheduler.ExponentialLR，使f1-measure的变化更稳定，波动更小。

未使用学习率衰减：



使用学习率衰减：

