1. **实验名称**

基于Attention-based Bi-LSTM模型和CNN模型实现关系抽取

1. **摘要**

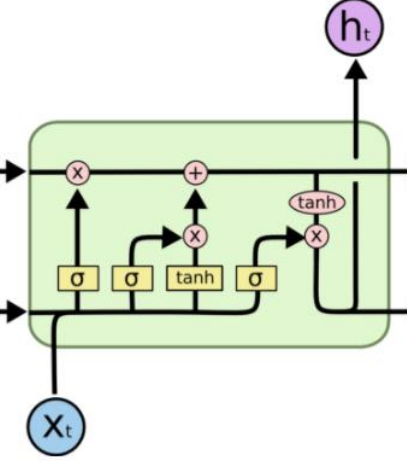
这次作业中，我们组的目标是使用两种模型分别实现关系抽取分别为Attention-based Bi-LSTM和CNN，并最终成功使用前者进行了训练，后者由于数据接口问题未能实现。我们采用SemEval-2010 Task 8的数据集，并进行训练和测试，同时使用预训练词向量GloVe对Word Feature进行表示，然后进一步使用PF和PI完善样本特征，形成一个56维的特征向量。最终实验中所得最好macro-F1可达75.8%，与原论文[1]中模型所得结果差距在5%左右。

1. **任务概述**

模型输入为一个句子以及句子中已标注的两个实体，需要从给定的关系集合中判断两个实体之间关系的类型。

关系类型由SemEval2010-Task8给出，共包含10种关系。抽取结果为每个句子中实体间关系类型名称。

1. **模型介绍**
2. LSTM

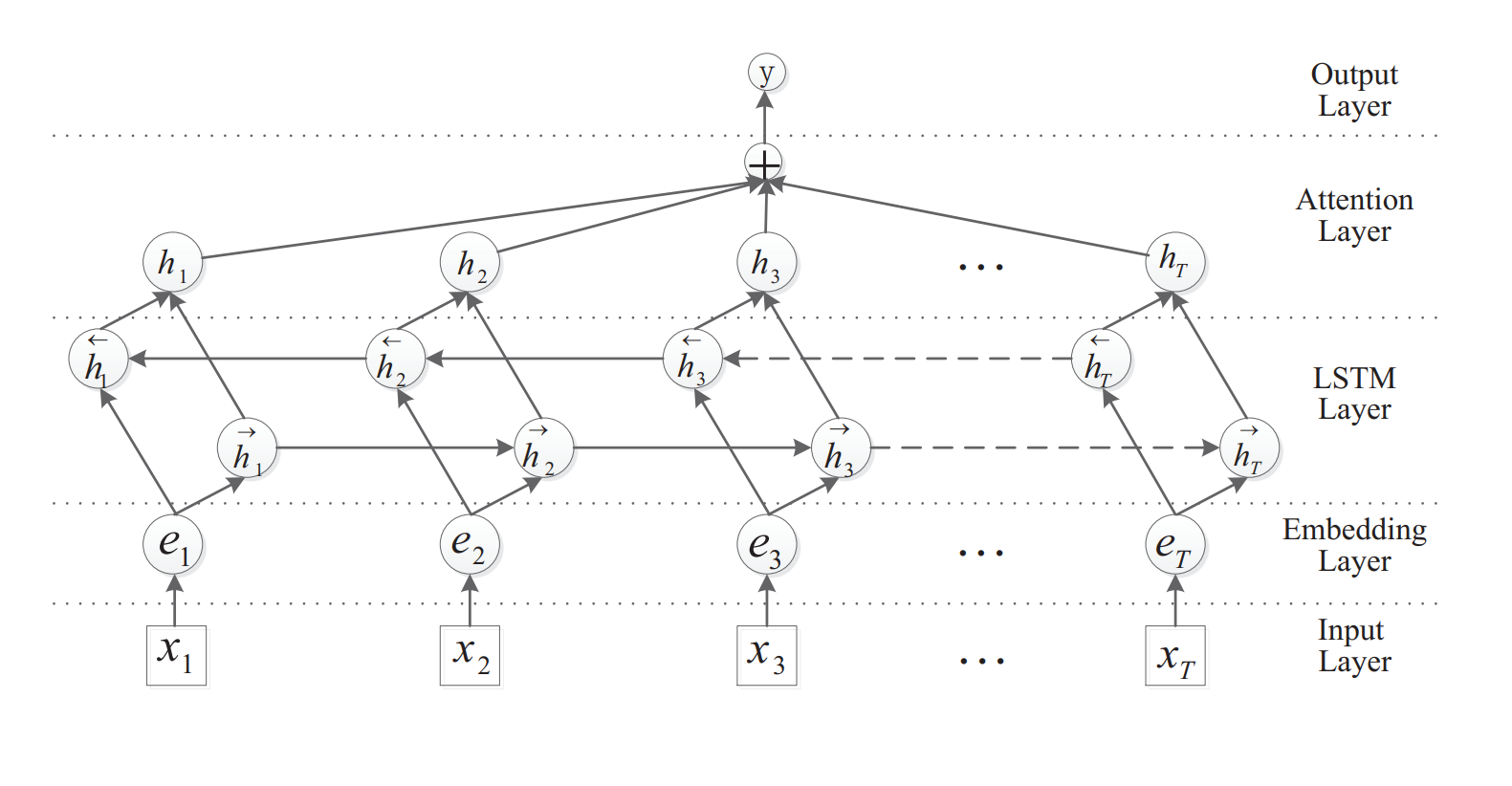


由从前往后的三个阶段组成，分别是遗忘阶段，选择记忆阶段和输出阶段组成，分别对应如下

遗忘门：

选择记忆门：

汇总输出：

1. ****Attention-based Bi-LSTM

与Bi-LSTM相比，在模型上增加了一个Attention层，使LSTM层的输出向量经过以下处理

其中，为词向量的维度，是一个训练学习得到的参数向量的转置。最终得到输出向量

1. **算法步骤**
2. 预处理
3. 数据清洗与数据划分
4. 使用正则表达式对语句中的不规范部分做规范处理。
5. 将SemEval2010中”train.txt”的数据随机分为两部分，分别是train部分和valid部分，两部分的比例是1：9。
6. 词向量构建

词向量由50WF+4PI+2PF组成。其中前50个数出自GloVe预训练的结果。中间四位为关键位置 “<e1>, </e1>, <e2>, </e2>”位置的one-hot表示，分别有<e1>对应1000，</e1>对应0100，<e2>对应0010，</e2>对应0001，其他对应0000。最后两位为该词在句子中的位置特征，分别对应与e11和e21的位置的差。

1. 梯度下降

使用Bi-LSTM，从而保证词的特征与前文、后文均相关，从而获得更高的准确率。使用Attention机制，从而为重要特征附较高的权值，进而实现突出重要特征的作用。

建立Attention+Bi-LSTM模型：以torch.optim.Adam为优化器；以torch.nn. functional.cross\_entropy为损失函数；以torch.nn.LSTM为正向传播模型；以torch.nn.Embedding实现词与对应词嵌入表示的快速对应；使用torch.autograd库中zero\_grad()梯度清零、backward()反向传播、step()更新参数，实现自动求导与梯度下降。

1. 模型评估

用macro-f1进行结果的评估，对每一个类分别求其对应的F1-score，然后对于出现过对应实例的F1-score进行求平均值，得到目标macro-F1，以此对训练的结果进行评测。

1. **实验设置**

OS：Ubuntu 18.04 64bit

CPU: 2vCPUs

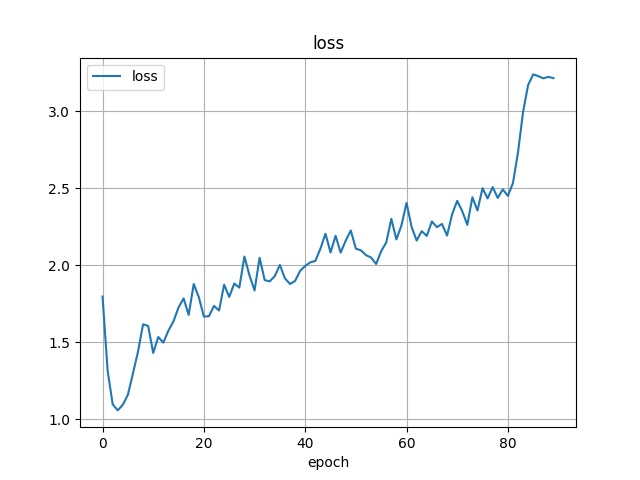
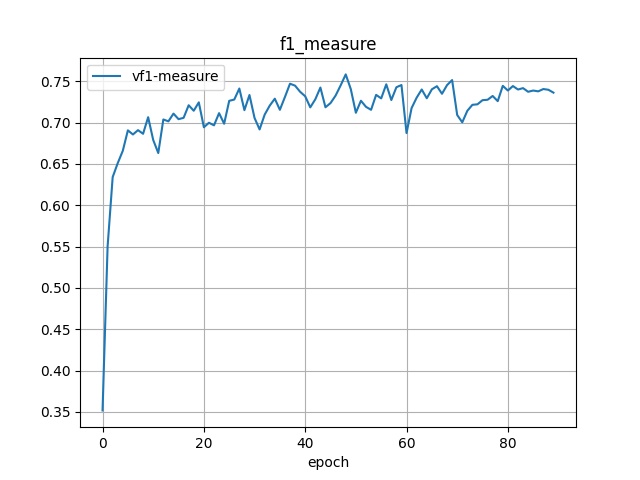
Date:20210708

Language:Python 3.9 torch 1.8.1

Dataset: SemEval2010-Task8

1. **实验结果**

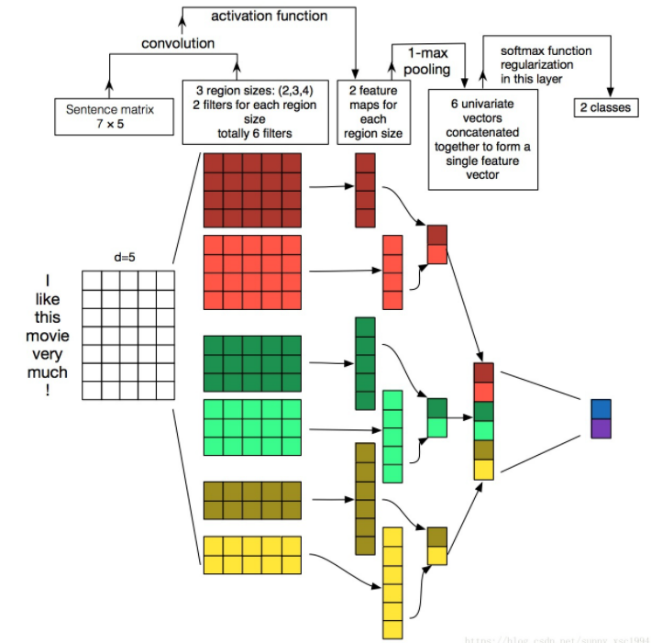
实验共在数据集上训练了91个epoch，loss和macro-F1变化如下图

****

1. **总结与收获**

此次代码中的ClearString函数参考了学长(@1120181891)的代码，其余部分均为小组成员独立完成。本次实验成功复现了Attention-based Bi-LSTM并在词特征表示上创新性地同时加入了WF、PF、PI，以期获得更高的macro-F1结果。最终在验证集上最高可达75.8%。

最终的macro-F1仍与论文结果有5%左右的差距，我们认为可以从词向量的训练以及LSTM层的参数设置上进行调整和改进。

1. **未完成的CNN**

CNN是一种典型的卷积网络，有卷积层、池化层、全连接层、softmax层组成。其中，卷积层与池化层配合，可形成多个卷积组，从而实现逐层提取信息，通过全连接层连接，并最终通过softmax函数完成最终分类。其中卷积层实现对局部特征的提取，池化层提取出当前局部的最主要特征，全连接层将训练出的特征做连接，相当于神经网络的隐藏层，softmax层输出预测的输出值，从而实现需要的多分类任务。

当前torch.nn库针对1维、2维、3维情况，给出了对应的卷积函数。Conv1d主要解决一维分类问题，Conv2d主要解决二维分类、识别问题，Conv3d主要解决三维分类、识别问题。

我们也尝试了用CNN模型解决关系抽取问题，并尝试为CNN与Attention-based Bi-LSTM设计相同的接口，以优化模型的封装性，但在实际操作中遇到了函数接口不一致问题，强行重写接口会带来很多不必要的麻烦。考虑到已有实验结果中CNN模型弱于Bi-LSTM模型，因此只实现了Bi-LSTM模型。

1. **参考文献**

[1] Peng Z , Wei S , Tian J , et al. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016.