第七天

2019年6月15日 9:38

fasttext的原理

1. 架构
    a. 输入层：词语+N-gram之后的词语 ---》embedding
    b. 隐藏层
    c. 输出层
2. 层次化的softmax（对传统softmax的优化）
    a. 哈夫曼树：树的带权路径最短的二叉树
    b. 带权路径长度：从根结点到该结点之间的路径长度与该结点的权的**乘积**
    c. 哈夫曼树的构造方法：
        i. 每次选择根节点权值最小的两棵树进行合并
    d. 哈夫曼编码：
        i. 满足前缀编码的要求
        ii. 把字母的出现次数叶子节点的权值，构造哈夫曼树
    e. 好处：减少计算的时间复杂度，提高效率
3. fasttext的负采样（对传统的softmax的优化）
    a. 采样：
        i. 数据根据label的数量分为V份，每一份是和数量的3/4次方成比例
        ii. 之后从数据中随机的选择neg个负样本
    b. 训练：
        i. 整个训练过程通过二分类的方式进行训练，提高当前正样本的概率
        ii. 损失=正样本的损失+neg个负样本的损失
    c. 好处：
        i. 提高训练速度
            1）训练数据少
            2）把多分类转化为2分类进行进行训练
        ii. 提高模型的稳健性，负样本可以模拟噪声数据
4. Seq2seq模型的原理
    a. 编码器：RNN
        i. 去理解句子
    b. 解码器：RNN
        i. 根据句子向量，得到输出

ii. 为什么可以有多个输出
    1）把前一次的输出和hidden_state作为下一次的输入
iii. 什么时候停止多个输出
    1）在训练的时候训练数据的target最后添加<EOS>
    2）在预测的时候预测结果为EOS，则停止

4. seq2seq的DEMO
    a. 文本转化为序列（数字序列，torch.LongTensor）
    b. 使用序列，准备数据集，准备Dataloader
        i. 随机种子，random.seed(9)
            1）使用同一个随机种子，每次随机的结果是一样的
        ii. add_eos:
            1）训练集中，target需要有EOS，告诉模型EOS是结束
            2）预测的时候，预测出EOS，意味着该结束
    c. 完成编码器
        i. 为了加速gru、lstm的计算，可以对句子进行打包和解包的操作
            1）torch.nn.utils.rnn.pack_paded_sequence
            2）nn.utils.rnn.pad_packed_sequence
            3）打包的方法需要batch的数据按照句子的长度降序排序
    d. 完成解码器
        i. decoder最开始的输入是一个[batch_size,1]的SOS，
        ii. 是一个在每个时间步循环
            1）output_t,hidden = forward_step(input,hidden)
            2）input是[batch_size,1],hidden和encoder hidden相同
            3）input先进行embedding
            4）结果通过gru处理得到output,[batch_size,1,hidden_size]
            5）结果进行变形为，【batch_size,vocab_Size】,进行log_softmax转化为概率
        iii. 把每个时间步的输出保存，用来和target计算loss
        iv. 当前时间步的输出，取概率最大的位置，作为预测值，作为下一个时间步的输入
    e. 完成seq2seq模型
        i. encode和decoder放到一起使用
    f. 完成模型训练的逻辑，进行训练
        i. loss怎么定义

1) 每个时间步是在进行多分类，类别数量=词典数量
2) outputs:[batch_Szie,max_Len,vocab_size],target:[batch_size,max_len]
3) 对outputs,和target进行变形，让batch_size*max_len,
4) 之后的损失计算和普通多分类一样

g. 完成模型评估的逻辑，进行模型评估
    i. 需要保存每个时间步的预测结果，供后续进行准确率的计算

作业：
使用对联数据，训练模型，能够输入上联，预测下联
数据地址：https://github.com/wb14123/couplet-dataset