LSTM-CRF 模型在序列标注问题上的应用

FudanNLP-nlp-beginner 任务四探究报告

戴宁

1 Introduction

序列标注问题，即给定观测序列 $x = (x_1, x_2, ..., x_n)$，需要预测相应的标注序列 $y = (y_1, y_2, ..., y_n)$，其中 n 是序列长度。具体的，学习系统需要建模 $p(y|x)$，在给定 x 的情况下，学习系统会将 $y^* = argmax_y\ p(y|x)$ 作为对当前观测序列所对应标注序列的预测。

1.1 LSTM-CRF

条件随机场 (CRF) 用于建模条件概率 $p(y|x)$，在序列标注问题中通常使用线性链条件随机场，其形式如下:

$$p(y|x;w) = \frac{exp(w \cdot F(x,y))}{\sum_{y'} exp(w \cdot F(x,y'))}$$

其中 w 为需要学习的参数向量，而 $F(x,y) = \sum_{i=1}^{n} f(x,i,y_{i-1},y_i)$ 是局部特征向量之和。

传统的机器学习就需要对特征函数 $f(x,i,y_{i-},y_i)$ 进行精心的设计从而达到较好的效果。

对于深度学习来说，由于其拥有较强的学习特征表示的能力，故常考虑用神经网络代替人工设计的特征。在使用神经网络时，考虑如下方式定义的 CRF：

$$p(y|x) = \frac{exp(\sum_{i=1}^{n} A_{y_{i-1},y_i} + P_{i,y_i}(x))}{\sum_{y'} exp(\sum_{i=1}^{n} A_{y'_{i-1},y'_i} + P_{i,y'_i}(x))}$$

其中 A 为转移矩阵，A_{y_{i-1},y_i} 用于评价从标记 y_{i-1} 转移到标记 y_i 的优劣，而 $P_{i,y_i}(x)$ 表示观测序列为 x 的前提下，在第 i 个位置使用标记 y_i 的优劣。其中 A 为通过学习得到的参数矩阵，$P(x)$ 为神经网络需要学习到的映射。

虽然 $P(x)$ 可以为任意类型的神经网络，但考虑到 x 是序列型输入，故主要构架使用 RNN(LSTM、GRU) 进行建模。

2 Model

模型参照 [3] 中提出的 char-level-feature + LSTM-CRF 模型，本文中 char-level-feature 的提取使用的是 GRU 网络，同时使用 IOBES 格式的标记。

2.1 embedding

2.1.1 word level

通过 [2] 中的对比，发现 GloVe 算法得到的单词的词嵌入更加适合命名实体识别任务，故使用与 [2] 中相同的由 GloVe 预训练得到 100 维词嵌入向量。同时通过 [2] 中的单词分布分析得到，在开发集和测试集中有大量训练集中未出现的单词，故使用所有 400000 个预训练过的单词作为词表，同时忽略大小写。

对于未出现在词表中的词，用标识符 <UNK> 代替。<UNK> 的 embedding 由词表中所有词的 embedding 取平均得到。

以上 embedding 在训练过程中可以被梯度下降调整。

2.1.2 char level

由于英语的语言特点，很多词型上的特征会对 NER 任务提供很多重要的信息，故考虑使用一个额外的 GRU 网络来提取每个单词词型的特征。

字符级的词表为训练集中所有出现过的字符，区分大小写。这样一来在单词级忽略掉的大小写信息可以由字符级的特征刻画。每个字符的词嵌入向量在 $[-1.0, 1.0]$ 直接由均匀分布生成，并且设置为可调整，通过训练过程中进一步学习得到。

字符级词表不设 <UNK> 标识，未出现在词表中的字符直接忽略不计。

每个单词的字符通过一个隐状态数为 25 的双向 GRU 网络，其前向与后向最终状态一起作为当前单词 embedding 的一部分。

通过拼接 word level 和 char level 的特征，最终每个单词由 150 维特征表示。

2.2 LSTM-CRF

每个句子由前面的步骤得到单词的向量表示，作为双向 LSTM 单元的输入。然后将前向与后向得到的输出拼接后通过一个全连接层，投影成标记对应词表的大小，作为对每个位置使用每个标记的打分。

之后通过 CRF 层与转移矩阵一起对当前的标记序列打分。

3 Training Detail

3.1 hyperparameters

字符词嵌入向量为 30 维，单词词嵌入向量为 100 维

GRU 的隐单元数为 25，LSTM 的隐单元数为 256。

mini batch 的大小为 128，使用 Adam 算法计算梯度，其中 $\beta_1 = 0.9, \beta_2 = 0.999$。

最终选择在开发集上 F1 分数最高的参数作为模型最终参数。

3.2 dropout

在字符特征输入到 GRU 之前、单词特征输入到 LSTM 之前、LSTM 输出进入全连接层之前都进行了 dropout 操作，keep_prob 为 0.4 或 0.5。

3.3 gradient clipping

为了防止 RNN 在训练过程中出现梯度爆炸的问题，在训练过程中对模型进行了梯度截断，公式如下：

$$dW_{cliped} = \frac{dW_{raw} * max\_norm}{max(||dW_{raw}||_2,\ max\_norm)}$$

其中 dW_{raw} 代表所有变量的原始梯度，dW_{cliped} 代表所有变量的被截断后梯度，$||dW_{raw}||_2$ 代表 dW_{raw} 的 L2 范数，max_norm 为截断的阈值 (设置为 5)。

3.4 learning rate decay

为了防止训练后期模型震荡，在模型训练整个过程的前 1/2 学习率为初始设定的学习率，而训练过程的后 1/2，采用线性衰减的策略，从初始学习率 0.002 一直衰减到 0 为止。

4 Result

本文分别实验了不加 char-level-feature 和附加 char-level-feature 的 LSTM-CRF 模型

4.1 without char-level-feature

一开始没有注意到 [2] 中对单词分布的分析，直接将训练集出现过的单词作为词表，模型的泛化能力很差，在训练集上能够到达接近 100% 的 F1 分数，但是在开发集上就只有 85% 左右的 F1，在测试集上更是 80% 都不到。

将所有 400000 个单词作为词表之后，在测试集上的 F1 分数能够达到 85.54%（使用 IOB1 格式标记）。

将标记改成 IOBES 格式后，模型的 F1 分数能够达到 87.25%

4.2 with char-level-feature

在前面的基础上，加上了 char-level-feature，模型在测试集上的 F1 分数直接升到了 90.51%

尝试对 LSTM 部分进行叠加，在 2 层 LSTM 的情况下达到了 90.70% 的 F1 分数。

最后考虑到，由于训练集单词数量较小，在训练过程中对 GloVe 向量的再调整可能造成模型的过拟合，故尝试将单词的词嵌入部分设置为不可被调整。

实验后发现对最终结果没有太大影响，在相同参数下 F1 分数反而稍微下降了些。

References

[1] Zhiheng Huang. ; Wei Xu. ; & Kai Yu. (2015) *Bidirectional LSTM-CRF Models for Sequence Tagging.* Retrieved from https://arxiv.org/abs/1508.01991

[2] Xuezhe Ma. ; & Eduard Hovy. (2016) *End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF.* Retrieved from https://arxiv.org/abs/1603.01354

[3] Guillaume Lample. ; Miguel Ballesteros. ; Sandeep Subramanian. ; Kazuya Kawakami. ; & Chris Dyer. (2016) *Neural Architectures for Named Entity Recognition.* Retrieved from https://arxiv.org/abs/1603.01360