

# Table of Contents

- [1 BiLSTM双向长短期记忆网络模型](#)
- [2 CRF条件随机场模型](#)
- [3 BiLSTM-CRF模型](#)
- [4 Pytorch实现](#)

## 命名实体识别任务的BiLSTM-CRF模型

### 1 BiLSTM双向长短期记忆网络模型

输入向量序列:

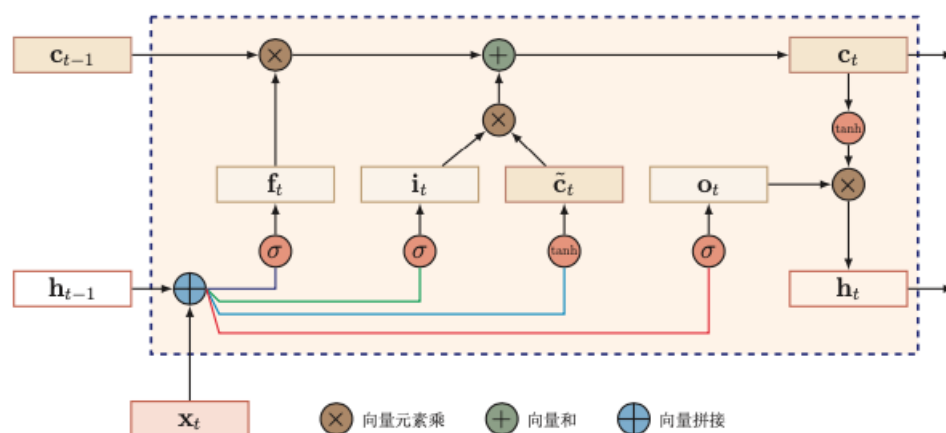
$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t, \dots, \mathbf{x}_n)$$

其中 $\mathbf{x}_t \in \mathbb{R}^m$ 为t时刻输入向量。

输出序列:

$$\mathbf{y} = (y_1, y_2, \dots, y_t, \dots, y_n)$$

其中 $y_t \in \{1, \dots, k\}$ 为t时刻输出。



LSTM引入门控机制来控制信息传递的路径。三个门分别为遗忘门 $\mathbf{f}_t$ ，输入门 $\mathbf{i}_t$ 和输出门 $\mathbf{o}_t$ ，取值在 $(0, 1)$ 之间，表示以一定的比例通过信息。

门控作用:

- 遗忘门 $\mathbf{f}_t$ 控制上一个时刻的内部状态 $\mathbf{c}_{t-1}$ 需要遗忘多少信息。
- 输入门 $\mathbf{i}_t$ 控制当前时刻的候选状态 $\tilde{\mathbf{c}}_t$ 需要保存多少信息。
- 输出门 $\mathbf{o}_t$ 控制当前时刻的内部状态 $\mathbf{c}_t$ 需要输出多少信息给外部状态 $\mathbf{h}_t$ 。

门控计算:

$$\mathbf{f}_t = \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f)$$

$$\mathbf{i}_t = \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i)$$

$$\mathbf{o}_t = \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o)$$

其中 $\sigma(\cdot)$ 为logistic函数，其输出区间为 $(0, 1)$ ， $\mathbf{x}_t$ 为当前时刻输入， $\mathbf{h}_{t-1}$ 为上一时刻的外部状态。

$\tilde{\mathbf{c}}_t$ 是通过非线性函数得到的候选状态。

候选状态计算:

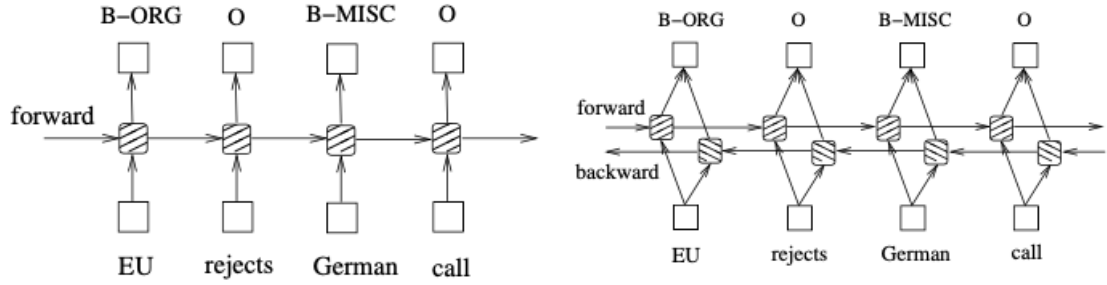
$$\tilde{\mathbf{c}}_t = \tanh(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c)$$

$\mathbf{c}_t$ 是内部状态，进行线性的循环信息传递；

$\mathbf{h}_t$ 是外部状态，进行非线性的循环信息传递。

输出计算:

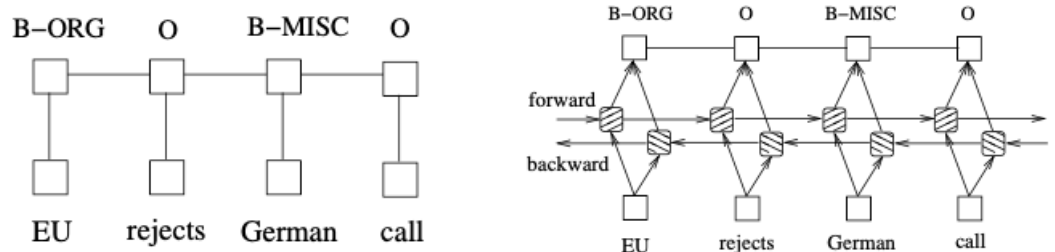
$$\begin{aligned} \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \end{aligned}$$



t时刻前向序列的输出为  $\vec{\mathbf{h}}_t$ , t时刻后向序列的输出为  $\overleftarrow{\mathbf{h}}_t$ , BiLSTM在t时刻的输出为

$$\begin{aligned} \mathbf{h}_t &= \vec{\mathbf{h}}_t \oplus \overleftarrow{\mathbf{h}}_t = \begin{bmatrix} \vec{\mathbf{h}}_t \\ \overleftarrow{\mathbf{h}}_t \end{bmatrix} \\ \mathbf{y}_t &= V\mathbf{h}_t \end{aligned}$$

## 2 CRF条件随机场模型



CRF模型:

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i)\right)$$

其中,

$$Z(x) = \sum_y \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i)\right)$$

转移得分:

$$p(y_i | y_{i-1}) = \frac{p(y_{i-1}, y_i)}{p(y_{i-1})}$$

其中

$$\begin{aligned} p(y_i | x) &= \frac{\alpha_i^\top(y_i | x) \beta_i(y_i | x)}{Z(x)} \\ p(y_{i-1}, y_i | x) &= \frac{\alpha_{i-1}^\top(y_{i-1} | x) M_i(y_{i-1}, y_i) \beta_i(y_i | x)}{Z(x)} \end{aligned}$$

## 3 BiLSTM-CRF模型

BiLSTM输出分值矩阵:

$$\mathbf{P} = [P_{i,j}]_{n \times k}$$

其中  $P_{i,j}$  为第i个单词对应第j个标签的分数。

转移分值矩阵:

$$\mathbf{A} = [A_{i,j}]_{(k+1) \times (k+1)}$$

其中  $A_{i,j} = p(y_i | y_{i-1})$ 。

模型分值：

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=1}^{n+1} A_{i,j} + \sum_{i=1}^n P_{i,j}$$

所有可能的标签序列上的softmax产生序列 $\mathbf{y}$ 的概率

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X}, \mathbf{y})}}{\sum_{\tilde{\mathbf{y}} \in \mathbf{Y}} e^{s(\mathbf{X}, \tilde{\mathbf{y}})}}$$

对数似然函数：

$$\begin{aligned} \log(p(\mathbf{y}|\mathbf{X})) &= s(\mathbf{X}, \mathbf{y}) - \log \left( \sum_{\tilde{\mathbf{y}} \in \mathbf{Y}} e^{s(\mathbf{X}, \tilde{\mathbf{y}})} \right) \\ &= s(\mathbf{X}, \mathbf{y}) - \log \text{add}_{\tilde{\mathbf{y}} \in \mathbf{Y}} s(\mathbf{X}, \tilde{\mathbf{y}}) \end{aligned}$$

预测输出：

$$\begin{aligned} \mathbf{y}^* &= \arg \max_{\tilde{\mathbf{y}} \in \mathbf{Y}} p(\mathbf{y}|\mathbf{X}) \\ &= \arg \max_{\tilde{\mathbf{y}} \in \mathbf{Y}} \log p(\mathbf{y}|\mathbf{X}) \\ &= \arg \max_{\tilde{\mathbf{y}} \in \mathbf{Y}} s(\mathbf{X}, \tilde{\mathbf{y}}) \\ &= - \arg \min_{\tilde{\mathbf{y}} \in \mathbf{Y}} -s(\mathbf{X}, \tilde{\mathbf{y}}) \end{aligned}$$

## ▼ 4 Pytorch实现

数据样例：

迈/O 向/O 充/O 满/O 希/O 望/O 的/O 新/O 世/O 纪/O -/O -/O -/O 九/O 九/O 八/O 年/O 新/O 年/O 讲/O 话/O (/O 附/O 图/O 片/O 1/O 张/O ) /O

中/B\_nt 共/M\_nt 中/M\_nt 央/E\_nt 总/O 书/O 记/O

国/O 家/O 主/O 席/O 江/B\_nr 泽/M\_nr 民/E\_nr

(/O -/O 九/O 九/O 七/O 年/O 十/O 二/O 月/O 三/O 十/O -/O 日/O ) /O

1/O 2/O 月/O 3/O 1/O 日/O

中/B\_nt 共/M\_nt 中/M\_nt 央/E\_nt 总/O 书/O 记/O

国/O 家/O 主/O 席/O 江/B\_nr 泽/M\_nr 民/E\_nr 发/O 表/O 1/O 9/O 9/O 8/O 年/O 新/O 年/O 讲/O 话/O 《/O 迈/O 向/O 充/O 满/O 希/O 望/O 的/O 新/O 世/O 纪/O 》/O

(/O 新/B\_nt 华/M\_nt 社/E\_nt 记/O 者/O 兰/B\_nr 红/M\_nr 光/E\_nr 摄/O ) /O

模型：

In [ ]:

```
1 #BiLSTM
2 self.lstm = nn.LSTM(self.embedding_dim, self.hidden_dim // 2,
3                     num_layers=1, bidirectional=True, batch_first=False)
4 self.hidden2tag = nn.Linear(self.hidden_dim, self.tagset_size)
5 #CRF
6 self.crf = CRF(self.tagset_size)
```

前向过程：

In [ ]:

```
1 #BiLSTM
2 embedding = self.word_embeds(x)
3 outputs, hidden = self.lstm(embedding)
4 outputs = self.dropout(outputs)
5 outputs = self.hidden2tag(outputs)
6 #CRF
7 outputs = self.crf.decode(outputs)
```

In [ ]:

```
1 预测输出:  
2 word:兰红光->type:nr
```