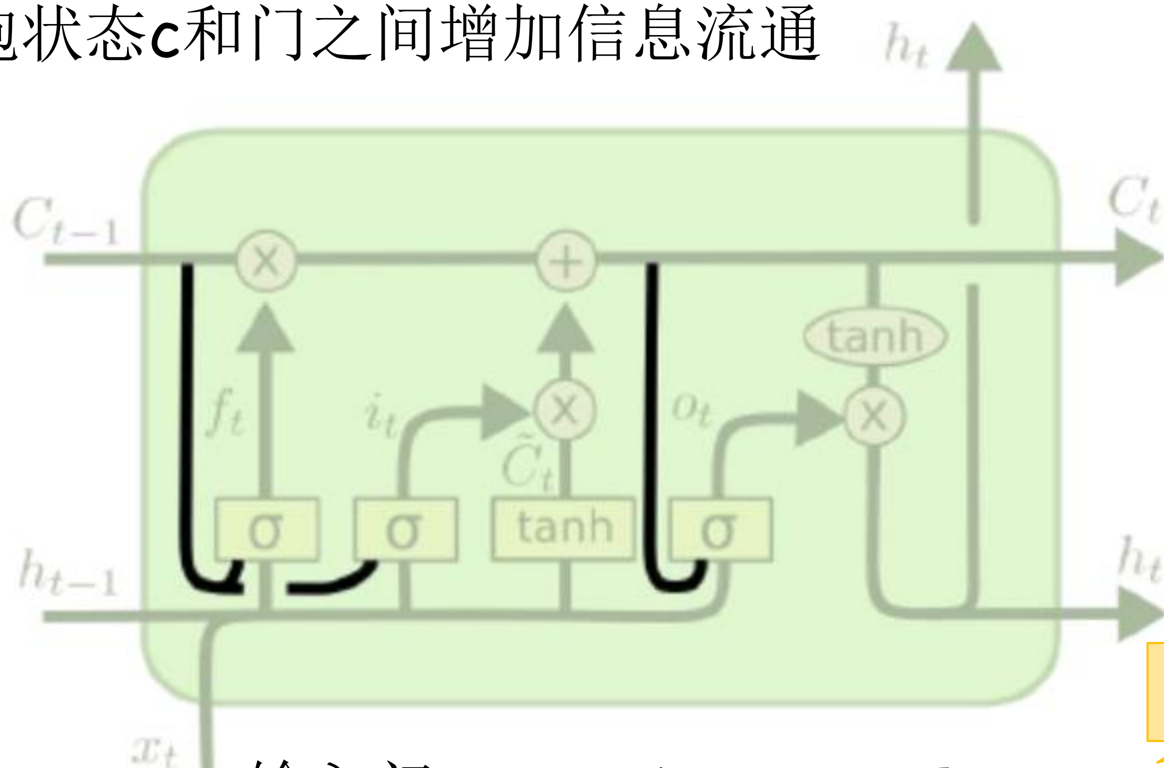
LSTM：变体1

修改门结构：添加窥视孔连接(Peephole connections)

在细胞状态c和门之间增加信息流通



对角矩阵

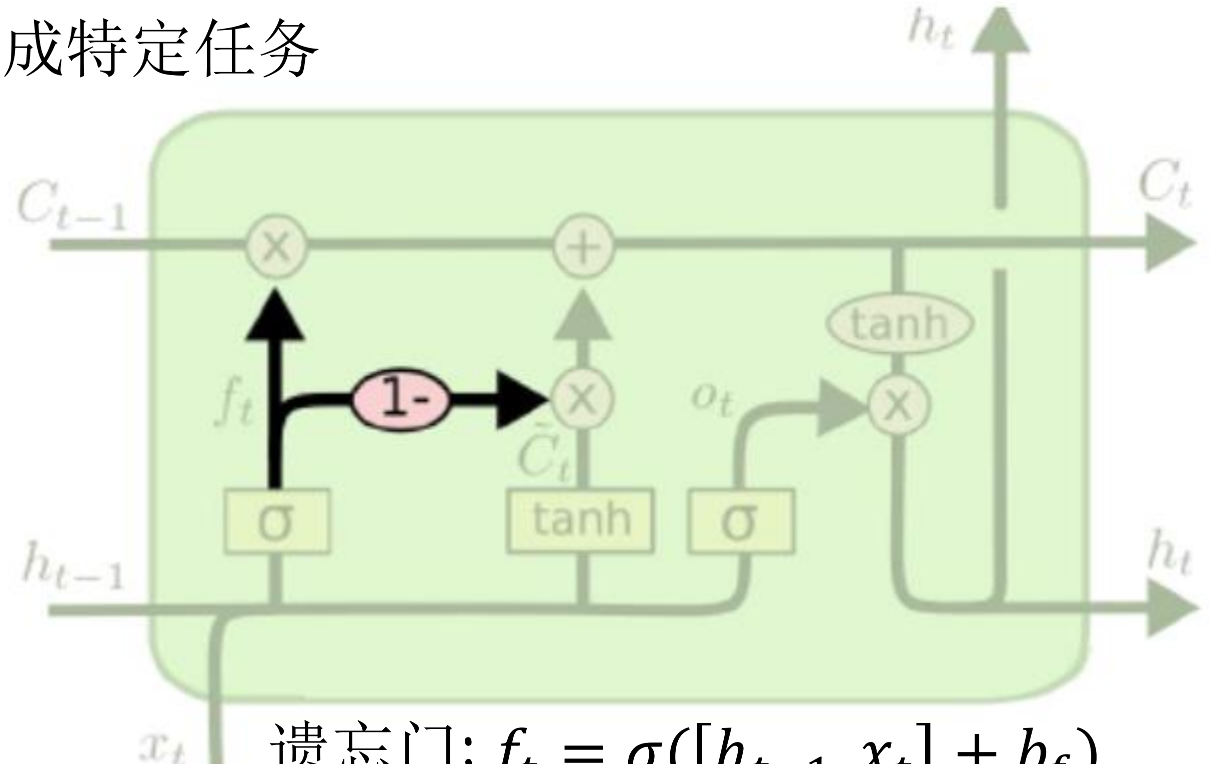
输入门: $i_t = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1} + b_i)$

遗忘门: $f_t = \sigma(W_f x_t + U_f h_{t-1} + V_f c_{t-1} + b_f)$

输出门: $o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o c_t + b_o)$

LSTM：变体2

耦合(coupling)遗忘门和输入门
耦合：多个实体通过各种交互作用而彼此影响，从而联合起来，协同完成特定任务



遗忘门: $f_t = \sigma([h_{t-1}, x_t] + b_f)$

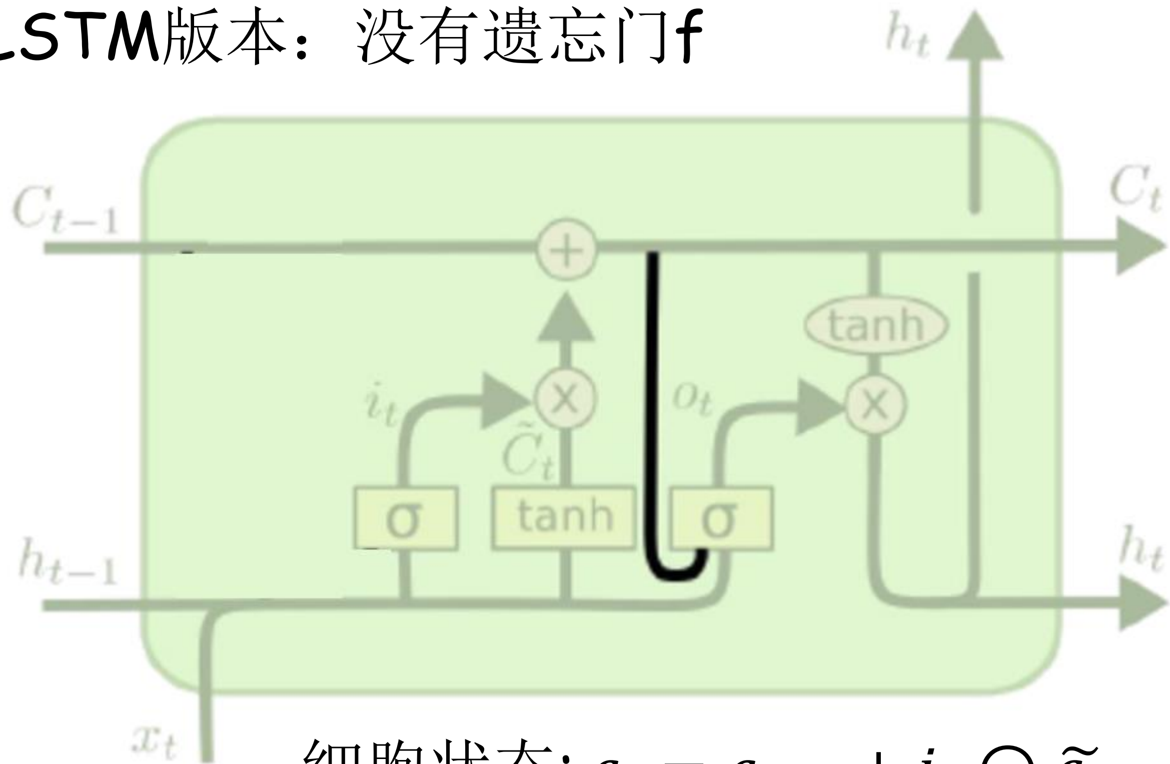
输入门: $i_t = 1 - f_t$

输出门: $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$

LSTM：变体3

经典LSTM：$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$

早期的LSTM版本：没有遗忘门f



细胞状态: $c_t = c_{t-1} + i_t \odot \tilde{c}_t$

缺点：旧细胞状态c_{t-1}全部流入新细胞状态c_t，$\boldsymbol{c}$ 不断增大，当输入序列非常大时，细胞单元的容量会饱和，模型性能下降

LSTM：门机制总结

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

遗忘门f: 决定旧细胞状态的保存或遗忘

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

输入门i: 决定候选细胞状态哪些被写入新细胞

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

输出门o: 决定新细胞状态哪些被输出到隐藏状态

$$\tilde{c}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

候选细胞状态：将写入细胞的新信息

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

新细胞状态: f实现旧细胞状态的保存或遗忘，i筛选候选细胞状态

$$h_t = o_t \odot \tanh(c_t)$$

隐藏状态: 读取了部分细胞状态

LSTM：门机制总结

- LSTM把存储信息的部位由隐藏状态h变成了独立的记忆模块，即细胞状态

  $$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \text{ 其中 } \tilde{c}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

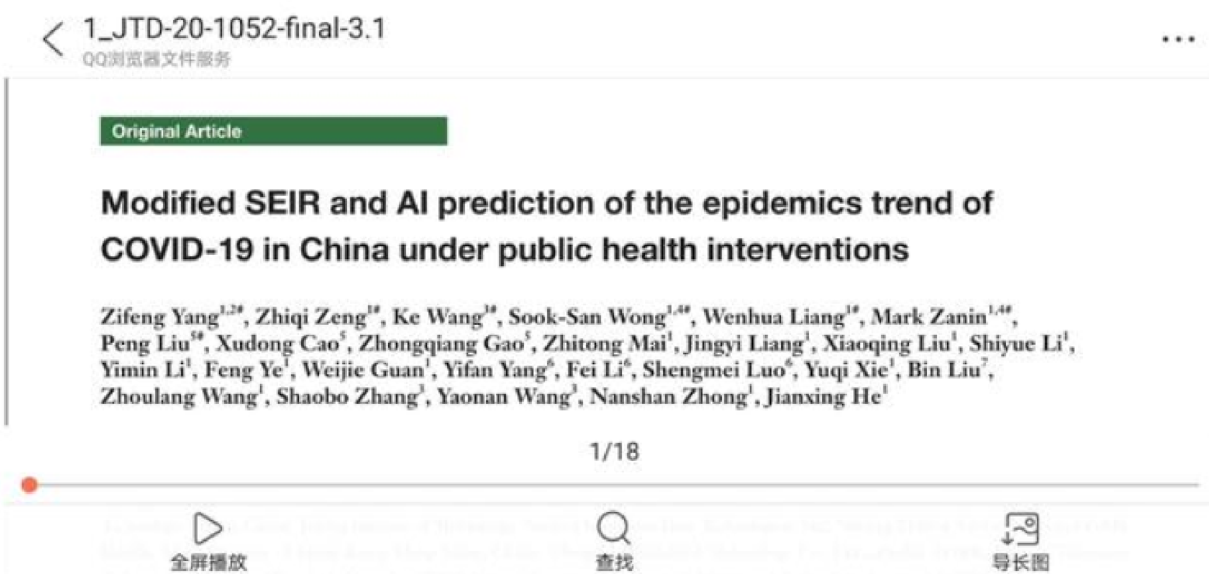
- 细胞状态c只需要进行一些线性的操作，更好地存储记忆信息
- LSTM利用不同的门，来对信息进行读取、遗忘、输出
- 门机制简单来看就是一个sigmoid层加一个逐元素乘操作
- 与RNN相比，LSTM能**减缓**梯度消失
- RNN: h每个时刻被重写→短期记忆(Short-Term Memory)
- LSTM: 细胞状态c可以把信息保存一定的时间间隔，生命周期要长于短期记忆h，长短期记忆指长的"短期记忆"．因此称为长短期记忆（Long Short-Term Memory）

LSTM：真实应用

钟南山院士团队率云创大数据等合写新冠病毒疫情预测论文正式发表

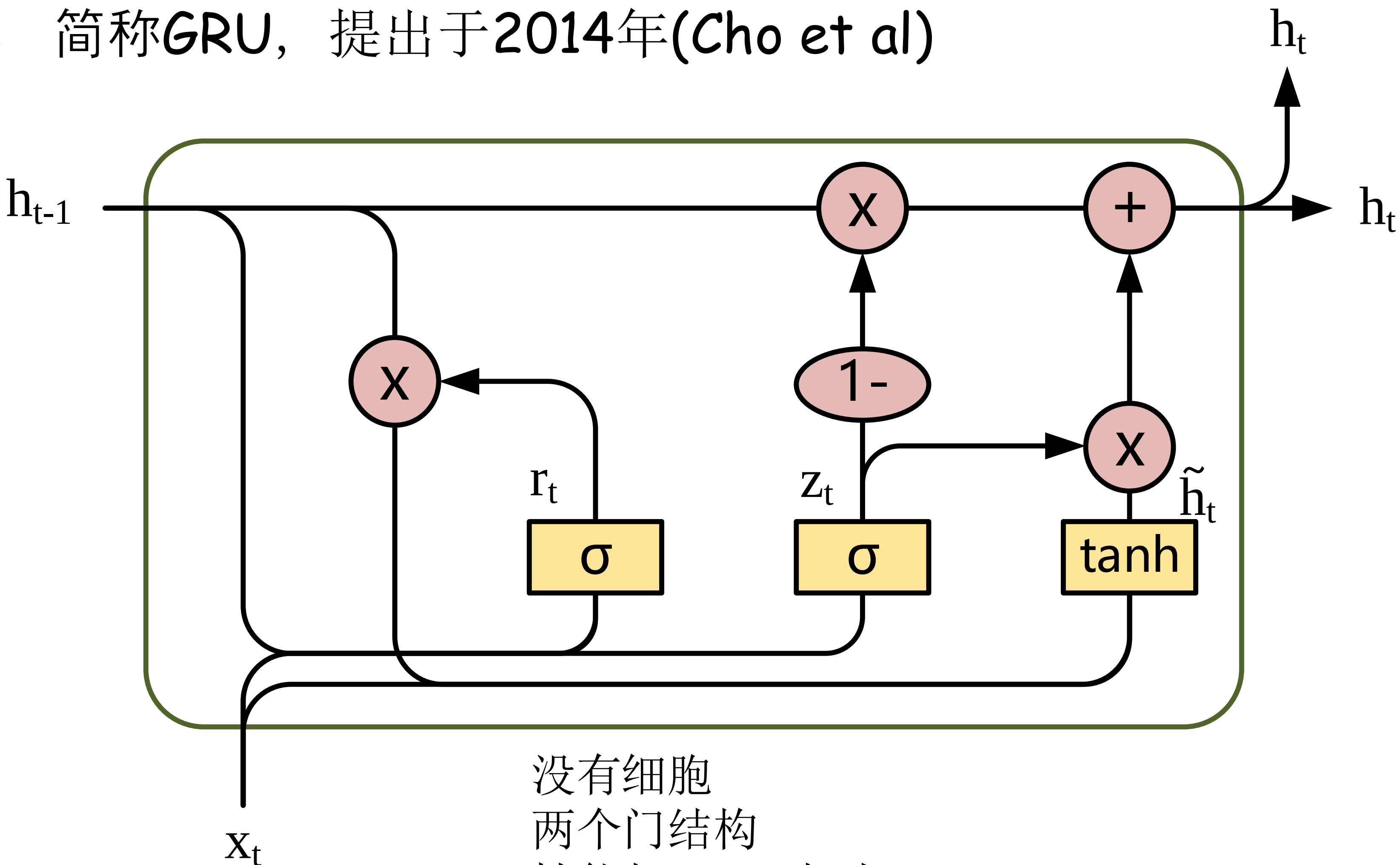
2020-03-04 19:16

近日，云创大数据研发团队在钟南山院士团队、何建行院士、杨子峰教授领导下，合写的新冠病毒疫情预测论文《基于SEIR优化模型和AI对公共卫生干预下的中国COVID-19暴发趋势预测》在《Journal of Thoracic Disease》(《胸部疾病杂志》)上发表。该文章提出了基于大数据的改进SEIR预测模型，并提出了数据受限条件下的人工智能LSTM预测方法。



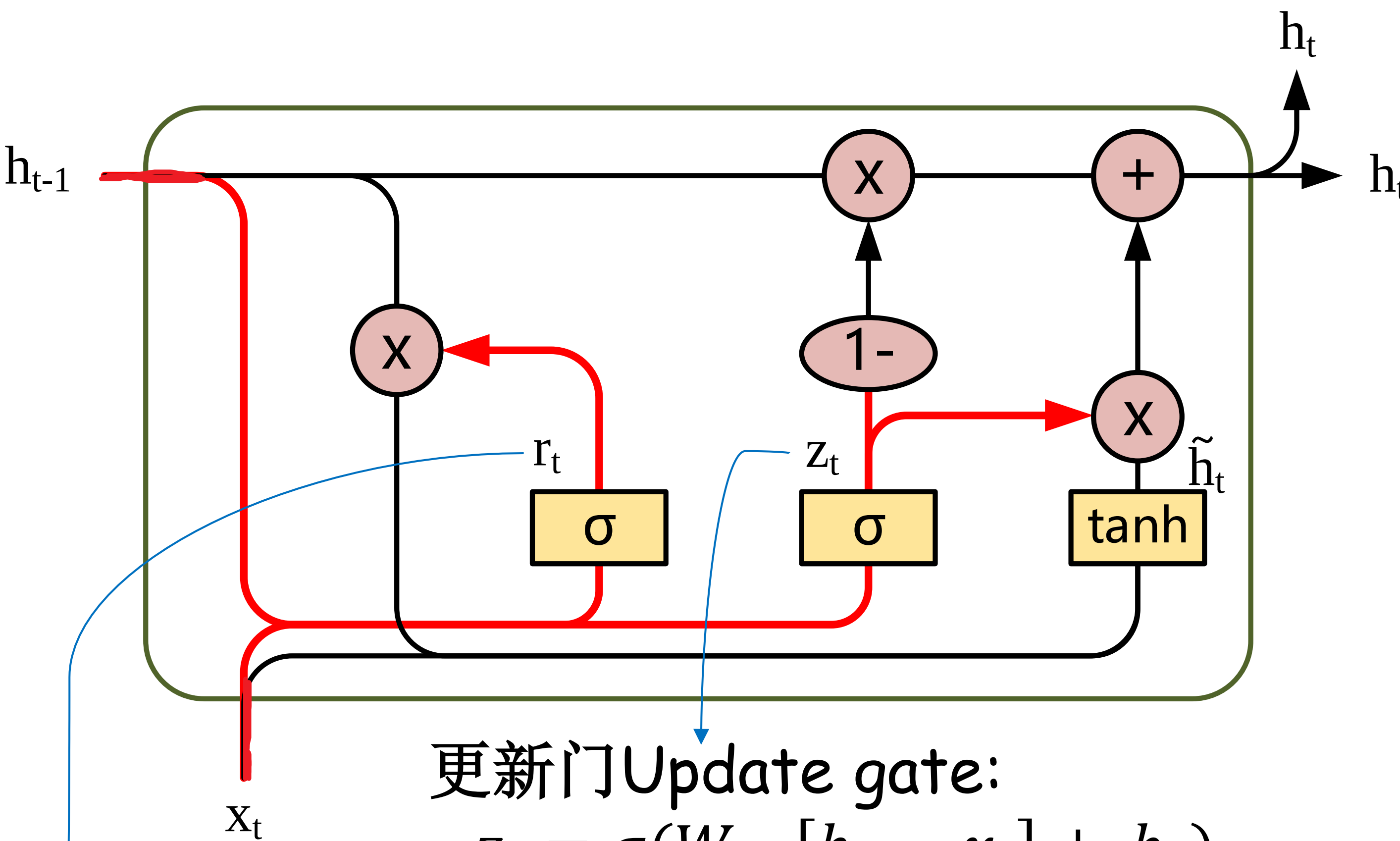
Gated Recurrent Unit

- 简称GRU，提出于2014年(Cho et al)



没有细胞
两个门结构
性能与LSTM相当

Gated Recurrent Unit



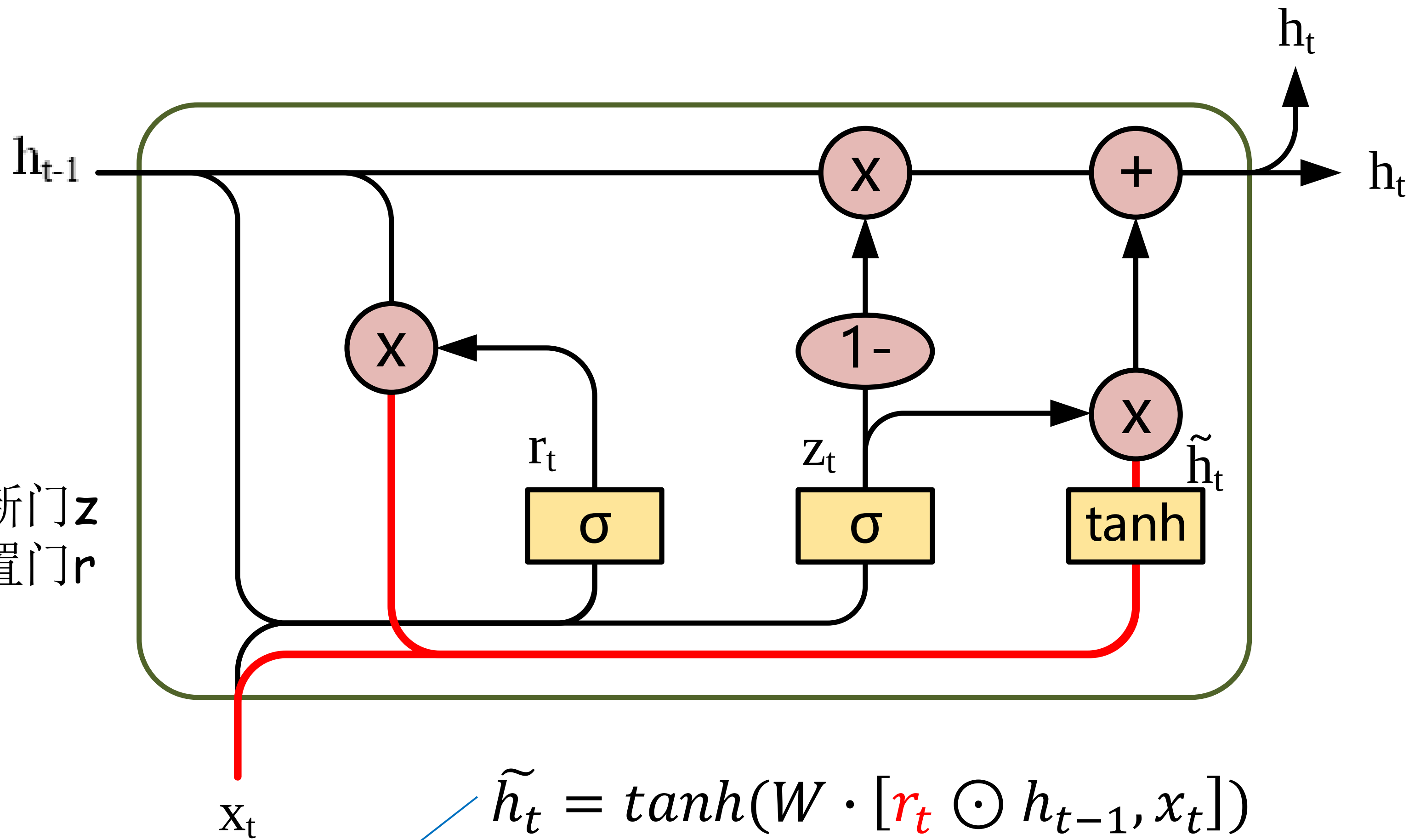
更新门Update gate:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

重置门Reset gate:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

Gated Recurrent Unit



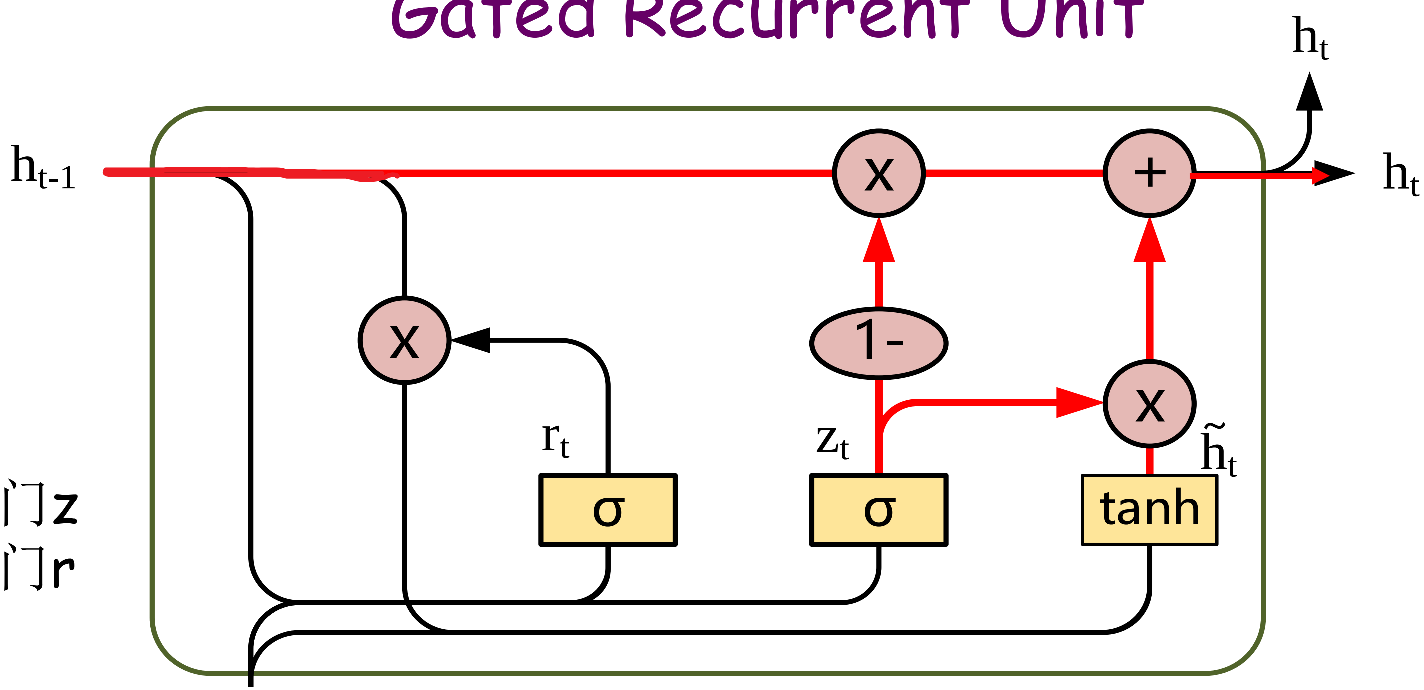
更新门z
重置门r

$$\tilde{h}_t = tanh(W \cdot [r_t \odot h_{t-1}, x_t])$$

new hidden state content

部分内容将被加入到新隐藏状态h_t中

Gated Recurrent Unit



更新门z
重置门r

$$\tilde{h}_t = tanh(W \cdot [r_t \odot h_{t-1}, x_t])$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

z=1, r=1: 退化成RNN
z=1, r=0: 只和当前输入x有关
z=0: 只和上一时刻状态有关

重置r:定义了有多少旧的记忆保存到当前的时间步

更新门z:充当了一个权重系数，决定了有多少新的信息$\tilde{h}_t$被使用，还有多少之前的旧记忆h_{t-1}被使用

Implementation (GRU过程)

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \qquad \widetilde{h_t} = tanh(W \cdot [r_t \odot h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \qquad h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t}$$

```
def gru_forward(input,initial_states,w_ih,w_hh,b_ih,b_hh):
    bs,T,i_size=input.shape
    h_size=initial_states.shape[-1]
    h_0=initial_states

    batch_w_ih=w_ih.unsqueeze(0).tile(bs,1,1)
    batch_w_hh=w_hh.unsqueeze(0).tile(bs,1,1)
    prev_h=h_0

    for t in range(T):
        x=input[:,t,:]
        w_times_x=torch.bmm(batch_w_ih,x.unsqueeze(-1)).squeeze(-1)
        w_times_h=torch.bmm(batch_w_hh,prev_h.unsqueeze(-1)).squeeze(-1)
        r=torch.sigmoid(w_times_x[:,:h_size]+w_times_h[:,:h_size]+b_ih[:h_size]+b_hh[:h_size])
        z=torch.sigmoid(w_times_x[:,h_size:2*h_size]+w_times_h[:,h_size:2*h_size]+\
            b_ih[h_size:2*h_size]+b_hh[h_size:2*h_size])
        n=torch.tanh(w_times_x[:,2*h_size:3*h_size]+b_ih[2*h_size:3*h_size]+\
            r*(w_times_h[:,2*h_size:3*h_size]+b_hh[2*h_size:3*h_size]))
        prev_h=(1-z)*n+z*prev_h
        output[:,t,:]=prev_h
    return output,prev_h
```

Implementation (调用LSTM)

```
class RNN(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, num_classes):
        super(RNN, self).__init__()
        self.hidden_size = hidden_size
        self.num_layers = num_layers
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers, batch_first=True)
        self.fc = nn.Linear(hidden_size, num_classes)

    def forward(self, x):
        # Set initial hidden and cell states
        h0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(device)
        c0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(device)

        # Forward propagate LSTM
        out, _ = self.lstm(x, (h0, c0))  # out: tensor of shape (batch_size, seq_length, hidden_size)

        # Decode the hidden state of the last time step
        out = self.fc(out[:, -1, :])# 此处的-1说明我们只取RNN最后输出的那个hn
        return out

model = RNN(input_size, hidden_size, num_layers, num_classes).to(device)
```

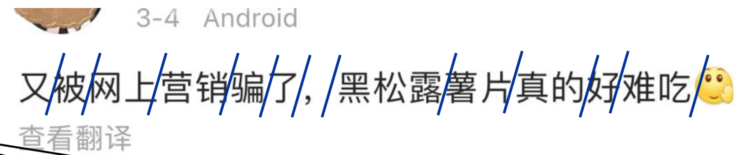
Outline

- 循环神经网络(RNN)
- 反向传播算法
- RNN主流变体(LSTM, GRU)
- 案例分析

LSTM用于文本情感分类

文本情感分类: many to one多对一任务

输入: $x = [x_1, x_2, \ldots x_n]$



预训练词向量序列

⇩

文本特征表示: **LSTM(x)**

⇩

输出: y = 情感标签

e.g. {1 (积极), -1 (消极)}; {高兴, 难过, 惊讶, 生气...}

根据粒度: 文档级, 句子级, 属性级(<u>target/aspect level</u>)

例子

x = Food is amazing, but the **service** is terrible

$y_{service}$ = ?

属性情感分析：任务大全

表 1　属性级情感分析任务概述

任务	简写	输入	输出	输出示例
属性词抽取	ATE	文本	a	pizza
属性情感分类	ALSC	文本, a	s	positive
观点词抽取	OTE	文本	o	delicious
属性导向的观点词抽取	TOWE	文本, a	o	delicious
类别识别	ACD	文本	c	food
类别情感分类	ACSA	文本, a	s	positive
属性-情感对抽取	ASPE	文本	(a, s)	pizza, positive
属性-观点对抽取	AOPE	文本	(a, o)	pizza, delicious
属性-类别情感分析	CSPE	文本	(c, s)	food, positive
属性情感三元组抽取	ASTE	文本	(a, o, s)	pizza, delicious, positive
目标属性情感检测	TASD	文本	(a, c, s)	pizza, food, positive
属性情感四元组抽取	ASQP	文本	(a, o, c, s)	(pizza, delicious, food quality, positive)

属性情感分析：常用数据

经典数据集: SemEval（国际语义测评大赛） 情感分类任务-公开测评数据集

表 2　属性级情感分析数据集介绍

数据集	领域	标签	语言	样本	链接
CASA [21]	Daily Dialog News	a, o, s	Chinese	3.0k 0.2k	http://www.ukp.tu-darmstadt.de/research/data/sentiment-analysis
Twitter 2014 [22]	Twitter	a, s	English	6.9k	http://goo.gl/5Enpu7
SemEval-2014 [23]	Restaurants Laptops	a, c, s a, s	English	3.8k 3.8k	https://alt.qcri.org/semeval2014/task4/
SemEval-2015 [24]	Restaurants Laptops	a, c, s c, s	English	2.0k 2.5k	https://alt.qcri.org/semeval2015/task12/
SemEval-2016 [25]	Restaurants Laptops	a, c, s c, s	Multilingual	2.6k 3.3k	https://alt.qcri.org/semeval2016/task5/
SemEval-2017 [26]	Twitter	c, s	Multilingual	50k	https://alt.qcri.org/semeval2017/task4/
MAMS [27]	Restaurants	a, c, s	English	22k	https://github.com/siat-nlp/MAMS-for-ABSA
ASTE-data-v2 [28]	Restaurants Laptops	a, o, s c, s	English	4.5k 1.4k	https://github.com/xuuuluuu/Position-Aware-Tagging-for-ASTE
ASC-QA [29]	Electronics Beauty Bags	a, c, s	Chinese	2.4k	https://github.com/jjwangnlp/ASC-QA
TOWE [30]	Restaurants Laptops	a, o	English	4.6k 1.4k	https://github.com/NJUNLP/TOWE
ARTS [31]	Restaurants Laptops	a, s	English	3.5k 1.8k	https://github.com/zhijing-jin/ARTS_TestSet
ACOS [19]	Restaurants Laptops	a, c, o, s	English	4.0k 2.2k	https://github.com/NUSTM/ACOS
ASAP [32]	Restaurants	c, s	Chinese	46.7k	https://github.com/Meituan-Dianping/asap
ABSA-QUAD [20]	Restaurants	a, c, o, s	English	3.7k	https://github.com/IsakZhang/ABSA-QUAD
One-ASQP [33]	Phone Food	a, c, o, s	English Chinese	7.1k 9.5k	https://www.github.com/Datastory-CN/ASQP-Datasets
ABSA-QUAD [34]	Electronics and etc	a, o, s	English	7.5k	https://github.com/NJUNLP/DMASTE

• 数据示例（三分类）


```xml
<sentences>
  <sentence id="457">
    <text>I'd call it an 'italian dinner'.</text>
    <aspectTerms>
      <aspectTerm term="dinner" polarity="neutral" from="24" to="30"/>
```


形成一条样本


```xml
    </aspectTerms>
    <aspectCategories>
      <aspectCategory category="anecdotes/miscellaneous" polarity="neutral"/>
    </aspectCategories>
  </sentence>
  <sentence id="1306">
    <text>my first time here was with my gf for our 12 month anniversary.</text>
```


无属性项， 忽略


```xml
    <aspectCategories>
      <aspectCategory category="anecdotes/miscellaneous" polarity="neutral"/>
    </aspectCategories>
  </sentence>
  <sentence id="3086">
    <text>While the place is not a hotspot hangout, the drinks are unique and pack a lot of bang for the buck.</text>
    <aspectTerms>
      <aspectTerm term="drinks" polarity="positive" from="46" to="52"/>
      <aspectTerm term="place" polarity="negative" from="10" to="15"/>
```

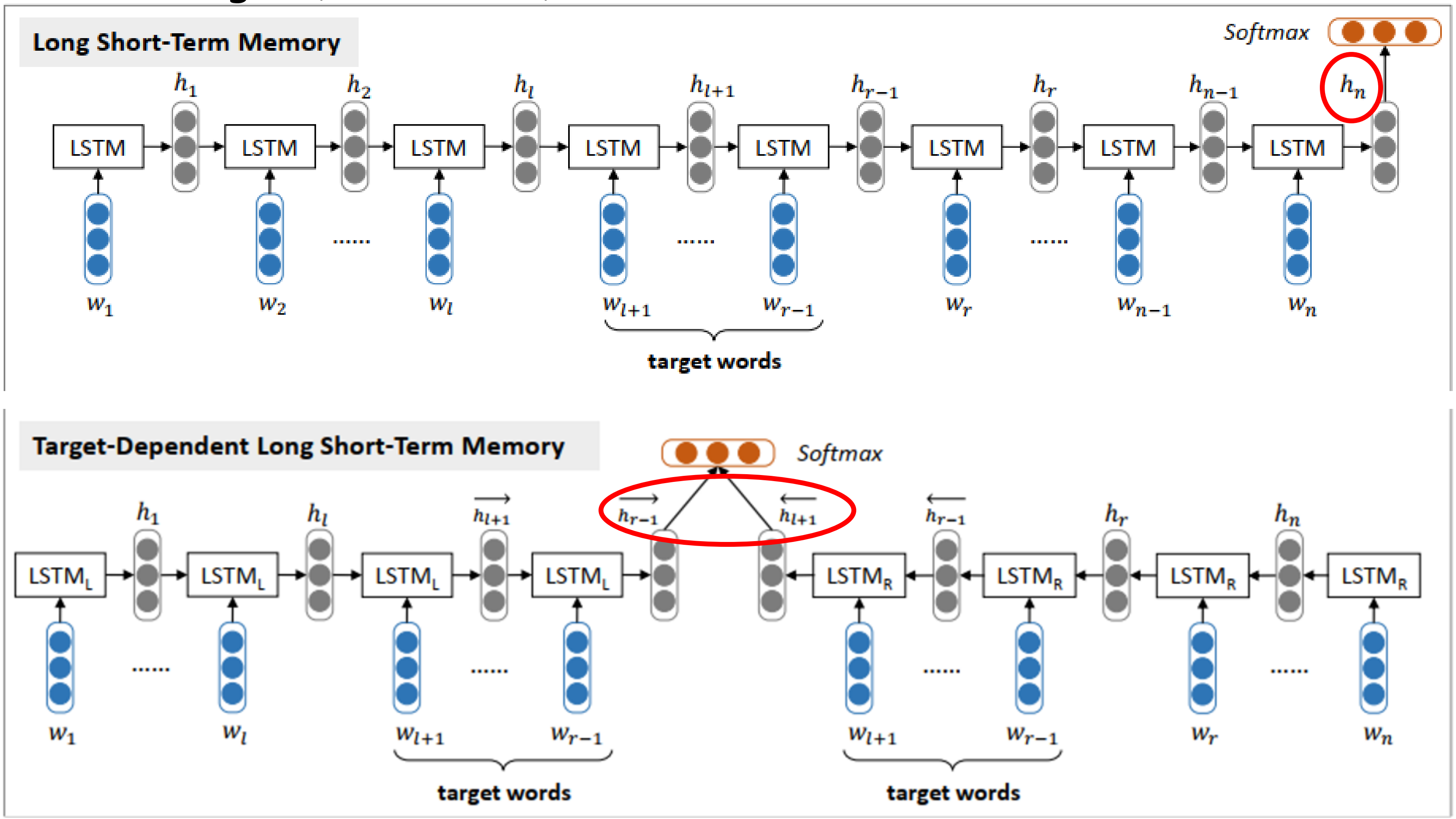

形成两条样本


```xml
    </aspectTerms>
    <aspectCategories>
      <aspectCategory category="food" polarity="positive"/>
      <aspectCategory category="ambience" polarity="negative"/>
      <aspectCategory category="price" polarity="positive"/>
    </aspectCategories>
  </sentence>
  <sentence id="2466">
    <text>I have walked by this place for eons and finally went thanks to a girls' night.</text>
    <aspectCategories>
      <aspectCategory category="anecdotes/miscellaneous" polarity="neutral"/>
    </aspectCategories>
  </sentence>
```

LSTM用于文本情感分类

针对一个target (词语，词组)

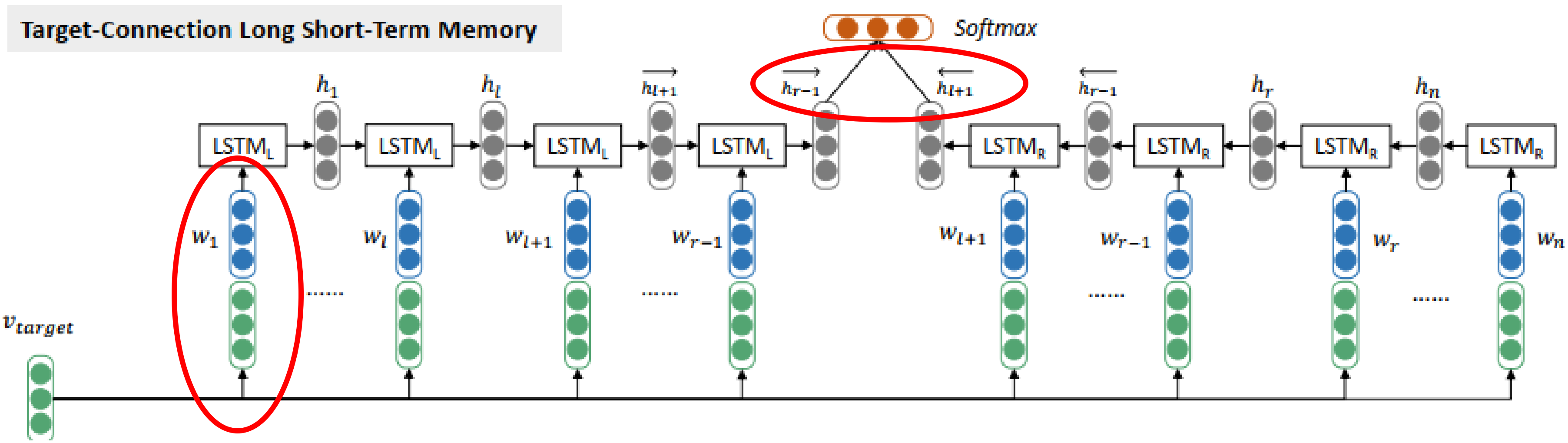
Tang et al, 2016. COLING. pdf



LSTM用于文本情感分类

针对一个target (词语，词组)

Tang et al, 2016. COLING. pdf



类别数

句子s预测为类别c的概率

$$loss = -\sum_{s\in S}\sum_{c=1}^{C} P_c^g(s)\cdot log(P_c(s))$$

训练集

样本s的类别是否是c

Q: 其他方式?

其他分类：e.g. 依存分析中的依存关系分类…

LSTM用于NER (1)

命名实体识别(Name Entity Recognition, NER):

序列标注任务，同步的many to many

输入:	张	三	出	生	于	南	京
输出:	B-PER	I-PER	O	O	O	B-LOC	I-LOC

输入:	Mark	Watney	visited	Mars
输出:	B-PER	I-PER	O	B-LOC

标注说明:

B=beginning, I=middle, O=not entity (E=end, S=single entity)

PER=person, LOC=location...

LSTM用于NER (1)

经典思路：基于机器学习模型，根据数据条件的不同，采用全监督/半监督/无监督/混合的方法

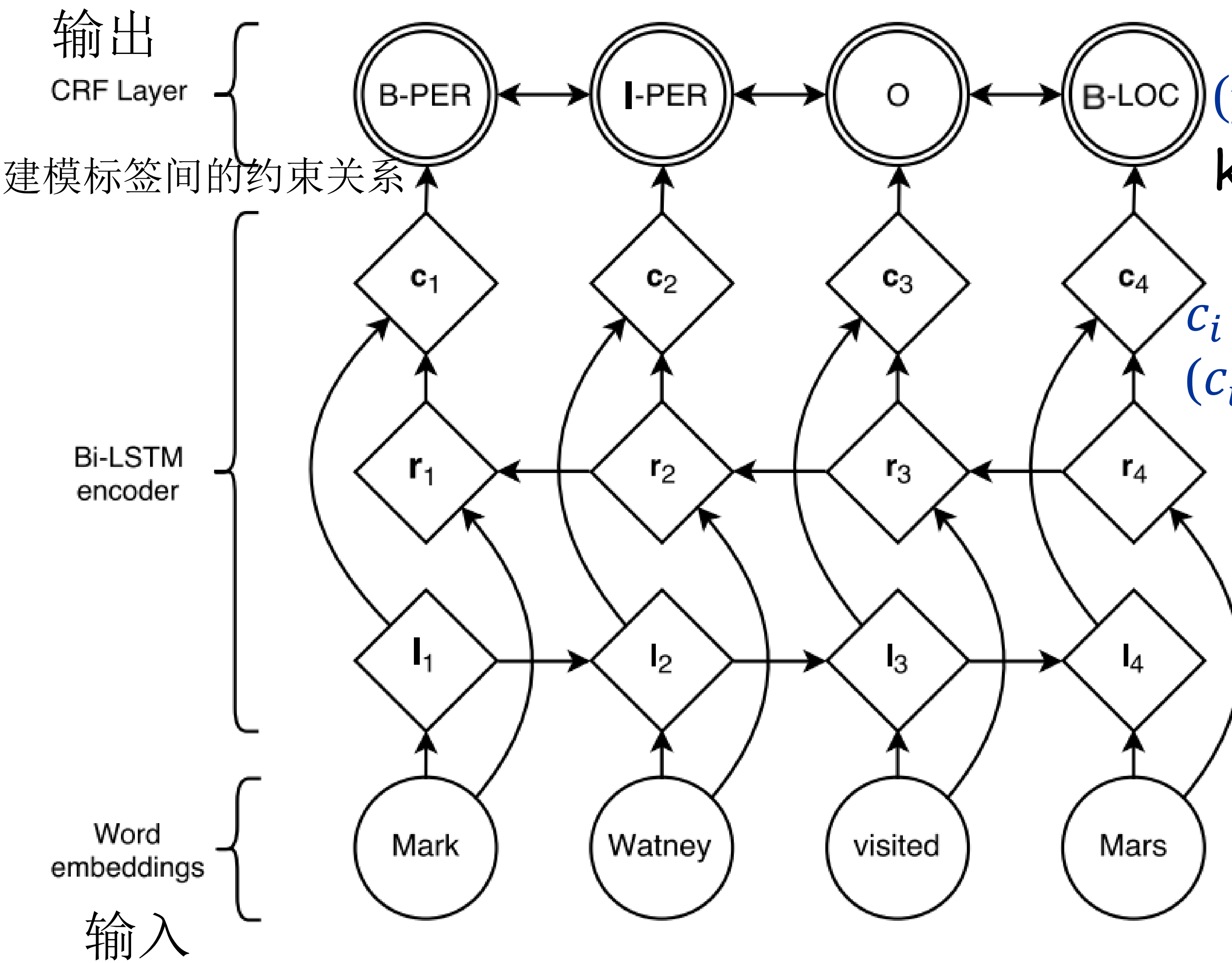
e.g. Stanford NER (CRF模型)，BANNER (CRF)，MALLET (HMM, CRF, ME)，BaseNER(CRF++)...

主要采用CRF (条件随机场) why?

CRF能显式捕获标签间的转移规律

特征模板（当前位置的前后n个位置上的token）→ CRF → 预测结果

LSTM用于NER (1)



$(p_i \ \dots \ p_n) \in R^{n*k}$

k: 标签类别数

$c_i = [\overrightarrow{r_i}, \overleftarrow{l_i}] \in R^m$

$(c_i \ \dots \ c_n) \in R^{n*m}$

n: 序列长度

$x_i \in R^d$

LSTM用于NER (1)

- 对于输入句 X=(x_1, x_2, …, x_n), 预测得到输出序列Y=(y_1, y_2, …, y_n), 定义X和Y的得分函数:

$$s(X,Y)=\sum_{i=0}^{n} A_{y_i,y_{i+1}}+\sum_{i=1}^{n} P_{i,y_i}$$

  - ✓ A 是转移矩阵，用于刻画相邻位置标签的转移、依赖关系，其中$A_{i,j}$ 代表 了从标签i 转移到j的得分
  - ✓ P 是LSTM层的输出，存储了n个k维向量，其中P_{i,y_i}代表了句中第i个字(词)的标签是y_i的分数

LSTM用于NER (1)

- 输出序列y的概率:

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

所有可能的序列

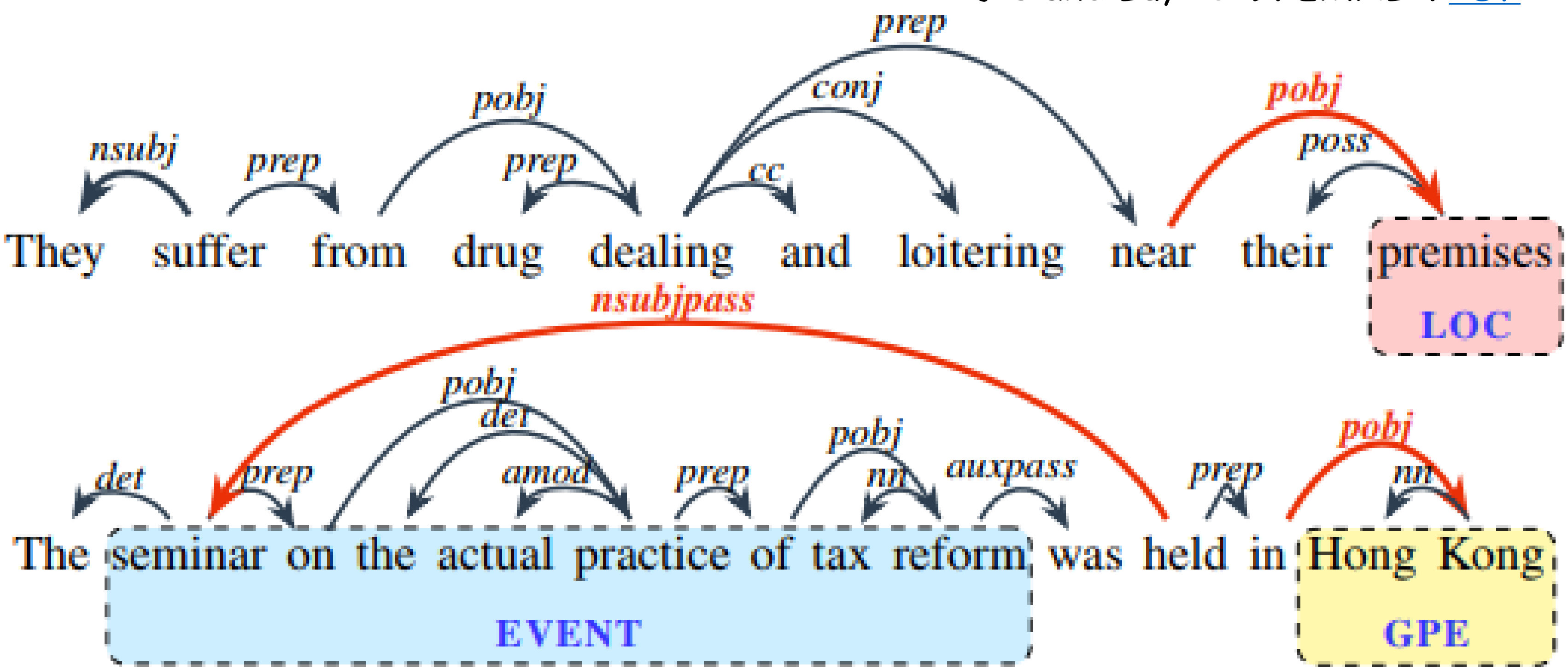
- 目标函数:

$$\log(p(\mathbf{y}|\mathbf{X})) = s(\mathbf{X},\mathbf{y}) - \log\left(\sum_{\widetilde{\mathbf{y}}\in\mathbf{Y_X}} e^{s(\mathbf{X},\widetilde{\mathbf{y}})}\right)$$

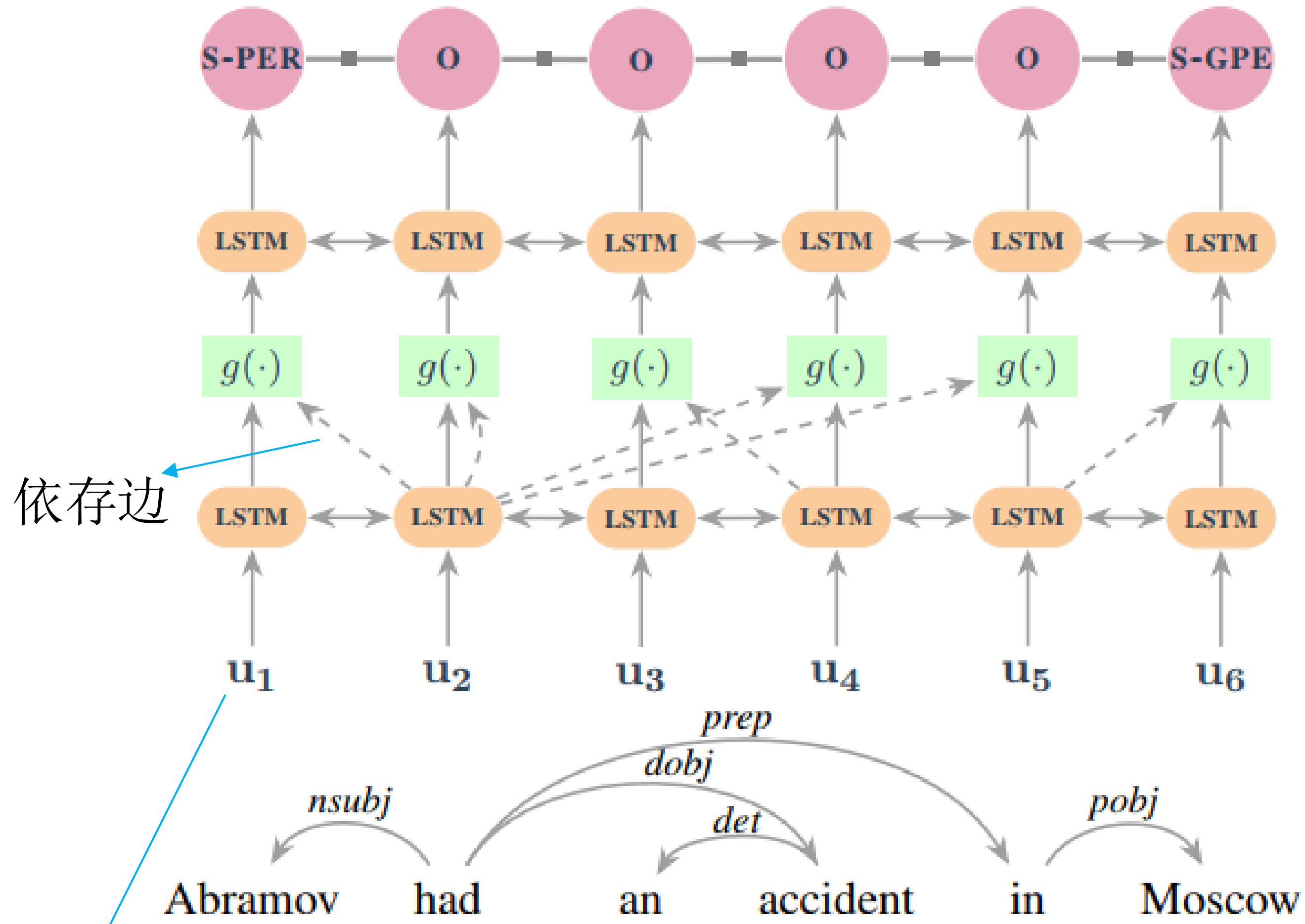
LSTM用于NER (2)

- 句法关系很可能暗示了命名实体的存在(主要句法关系?)
- 依存树能提供词和词之间的长距离依赖

Jie and Lu, 2019. EMNLP. [PDF]



LSTM用于NER (2)



Q：这样的设计可能有什么缺点？

$g(\mathbf{h}_i, \mathbf{h}_{p_i})$ 交互函数

g: 除了每个时间步的隐向量，还包含了某些<u>其他位置</u>的隐向量。

<u>其他位置</u>：和当前词存在依存关系的词所在位置

$$\mathbf{u}_i = [\mathbf{w}_i; \mathbf{w}_h; \mathbf{v}_r], \quad x_h = parent(x_i)$$

父节点向量

依存关系向量随机初始化

LSTM用于NER (2)

- 可以继续堆叠LSTM层

$$\mathbf{H}^{(l+1)} = \mathrm{BiLSTM}\left(f\left(\mathbf{H}^{(l)}\right)\right)$$

$$\mathbf{H}^{(l)} = \left[\mathbf{h}_1^{(l)}, \mathbf{h}_2^{(l)}, \cdots, \mathbf{h}_n^{(l)}\right]$$

$$f\left(\mathbf{H}^{(l)}\right) = \left[g(\mathbf{h}_1^{(l)}, \mathbf{h}_{p1}^{(l)}), \cdots, g(\mathbf{h}_n^{(l)}, \mathbf{h}_{pn}^{(l)})\right]$$

- g函数选择

Interaction Function	$g(\mathbf{h}_i, \mathbf{h}_{p_i})$
Self connection	$\mathbf{h}_i$
Concatenation	$\mathbf{h}_i \bigoplus \mathbf{h}_{p_i}$
Addition	$\mathbf{h}_i + \mathbf{h}_{p_i}$
MLP	$\mathrm{ReLU}(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{h}_{p_i})$

还有哪些函数可以用？

实验作业

- 基于LSTM-CRF的NER，复现2019EMNLP论文
- 基于OntoNotes英文数据集

https://cemantix.org/data/ontonotes.html
https://huggingface.co/datasets/ontonotes/conll2012_ontonotesv5

Welcome   a2e_0015.gold_conll

Users > klmini > Downloads > conll-2012 > v12 > data > train > data > english > annotations > wb > a2e > 00 > a2e_0015.gold_conll

NER标签 涉及到论元/共指 列数有多有少

```
#begin document (wb/a2e/00/a2e_0015); part 000
wb/a2e/00/a2e_0015   0   0          Police    NN   (TOP(FRAG(NP(NP*)    -     -   -  -   *   (ARG0*   (ARG1*    -
wb/a2e/00/a2e_0015   0   1             and    CC                *    -     -   -  -   *        *        *    -
wb/a2e/00/a2e_0015   0   2   Investigation   NNP              (NP*    -     -   -  -   *        *        *    -
wb/a2e/00/a2e_0015   0   3       Authority   NNP              *))    -     -   -  -   *       *)       *)    -
wb/a2e/00/a2e_0015   0   4          Agreed   VBN              (VP*  agree  01   3  -   *      (V*)       *    -
wb/a2e/00/a2e_0015   0   5              to    TO            (S(VP*    -     -   -  -   *   (ARG1*        *    -
wb/a2e/00/a2e_0015   0   6              be    VB              (VP*   be    01   1  -   *        *      (V*)   -
wb/a2e/00/a2e_0015   0   7          Unjust    JJ            (ADJP*    -     -   -  -   *        *   (ARG2*    -
wb/a2e/00/a2e_0015   0   8              to    IN              (PP*    -     -   -  -   *        *        *    -
wb/a2e/00/a2e_0015   0   9            this    DT              (NP*    -     -   -  -   *        *        *   (8
wb/a2e/00/a2e_0015   0  10          Sheikh   NNP         *))))))    -     -   -  -   *       *)       *)   8)
wb/a2e/00/a2e_0015   0  11               /   NFP                *    -     -   -  -   *        *        *    -
wb/a2e/00/a2e_0015   0  12             His  PRP$              (NP*    -     -   -  -   *        *        *  (8)
wb/a2e/00/a2e_0015   0  13         Picture    NN              *)))    -     -   -  -   *        *        *    -

wb/a2e/00/a2e_0015   0   0       Abd   NNP   (TOP(FRAG(NP*   -  -  -  -   (PERSON*   (93
wb/a2e/00/a2e_0015   0   1        al   NNP               *   -  -  -  -          *     -
wb/a2e/00/a2e_0015   0   2         -  HYPH               *   -  -  -  -          *     -
wb/a2e/00/a2e_0015   0   3    Rahman   NNP               *   -  -  -  -          *     -
wb/a2e/00/a2e_0015   0   4      2002    CD            *)))   -  -  -  -         *)   93)
```

NER标签

短语结构树
可以通过
stanfordnlp转化为
依存树

Column	Type	Description
1	Document ID	This is a variation on the document filename
2	Part number	Some files are divided into multiple parts numbered as 000, 001, 002, ... etc.
3	Word number	This is the word index of the word in that sentence.
4	Word itself	This is the token as segmented/tokenized in the Treebank. Initially the file contain the placeholder which gets replaced by the actual token from the Treebank which is part of the OntoNotes release.`*_skel[WORD]`
5	Part-of-Speech	This is the Penn Treebank style part of speech. When parse information is missing, all part of speeches except the one for which there is some sense or proposition annotation are marked with a XX tag. The verb is marked with just a VERB tag.
6	Parse bit	This is the bracketed structure broken before the first open parenthesis in the parse, and the word/part-of-speech leaf replaced with a *. The full parse can be created by substituting the asterix with the "([pos] [word])" string (or leaf) and concatenating the items in the rows of that column. When the parse information is missing, the first word of a sentence is tagged as "(TOP*" and the last word is tagged as "*)" and all intermediate words are tagged with a "*".
7	Predicate lemma	The predicate lemma is mentioned for the rows for which we have semantic role information or word sense information. All other rows are marked with a "-".
8	Predicate Frameset ID	This is the PropBank frameset ID of the predicate in Column 7.
9	Word sense	This is the word sense of the word in Column 3.
10	Speaker/Author	This is the speaker or author name where available. Mostly in Broadcast Conversation and Web Log data. When not available the rows are marked with an "-".
11	Named Entities	These columns identifies the spans representing various named entities. For documents which do not have named entity annotation, each line is represented with an "*".
12:N	Predicate Arguments	There is one column each of predicate argument structure information for the predicate mentioned in Column 7. If there are no predicates tagged in a sentence this is a single column with all rows marked with an "*".
N	Coreference	Coreference chain information encoded in a parenthesis structure. For documents that do not have coreference annotations, each line is represented with a "-".

实体标签及含义

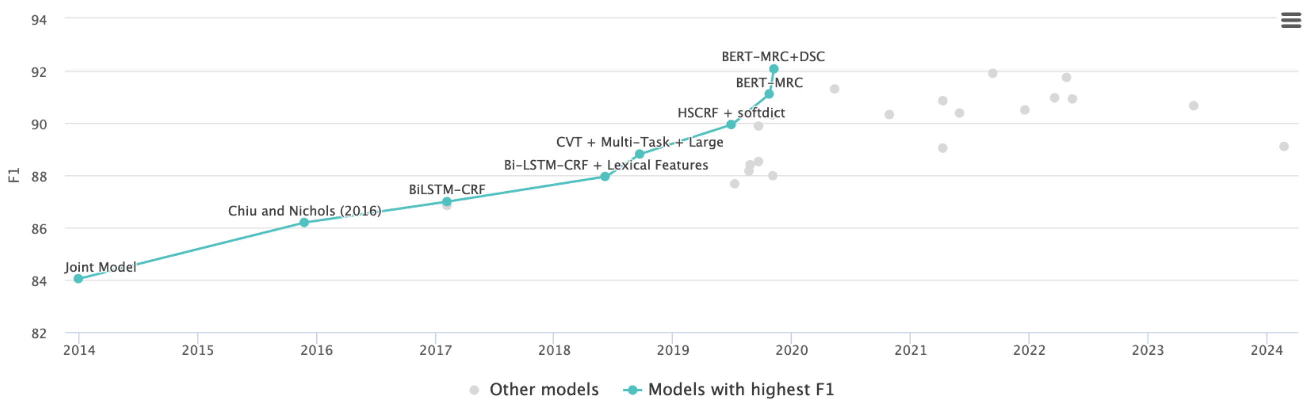
CARDINAL：基数值
DATE：日期值
EVENT：事件名称
FAC：建筑物名称
GPE：地缘政治实体
LANGUAGE：语言名称
LAW：法律名称
LOC：地点名称
MONEY：货币名称
NORP：组织关系
ORDINAL：序数值
ORG：组织名称
PERCENT：百分比值
PERSON：人名
PRODUCT：产品名称
QUANTITY：数量值
TIME：时间值
WORK_OF_ART：艺术作品名称

查看数据集上某任务的SOTA：https://paperswithcode.com/sota/named-entity-recognition-ner-on-ontonotes-v5

Named Entity Recognition (NER) on Ontonotes v5 (English)

Leaderboard　Dataset

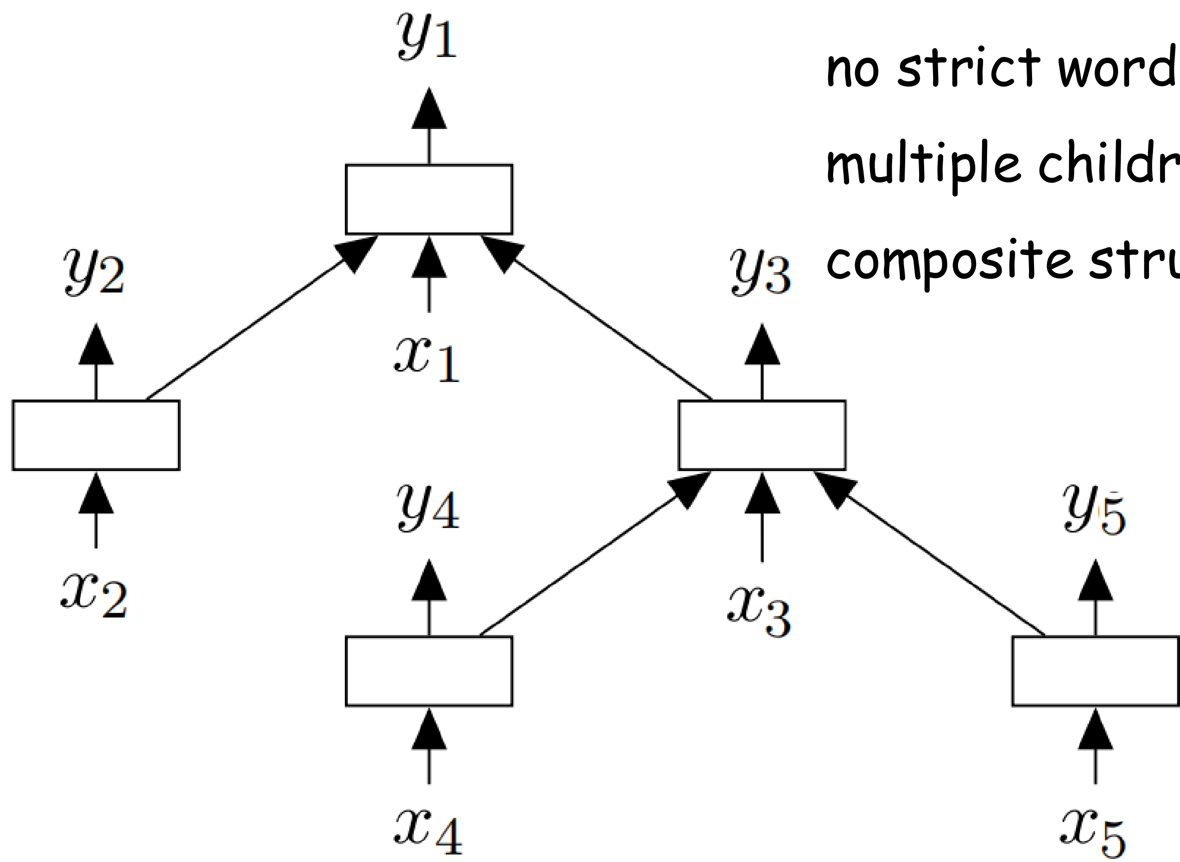
View F1 by Date for All models



Filter: Transformer LSTM MRC-based CNN untagged　Edit Leaderboard

Rank	Model	F1	Precision	Recall	Extra Training Data	Paper	Code	Result	Year	Tags
1	**BERT-MRC+DSC**	92.07			✓	Dice Loss for Data-imbalanced NLP Tasks			2019	MRC-based Transformer
2	**PL-Marker**	91.9	92.0	91.7	×	Packed Levitated Marker for Entity and Relation Extraction			2021	
3	**Baseline + BS**	91.74			×	Boundary Smoothing for Named Entity Recognition			2022	

改进：Tree-Structured LSTMs



no strict word ordering

multiple children

composite structure

具有任意分支的树LSTM

Tai, Socher and Manning, 2015. ACL. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks

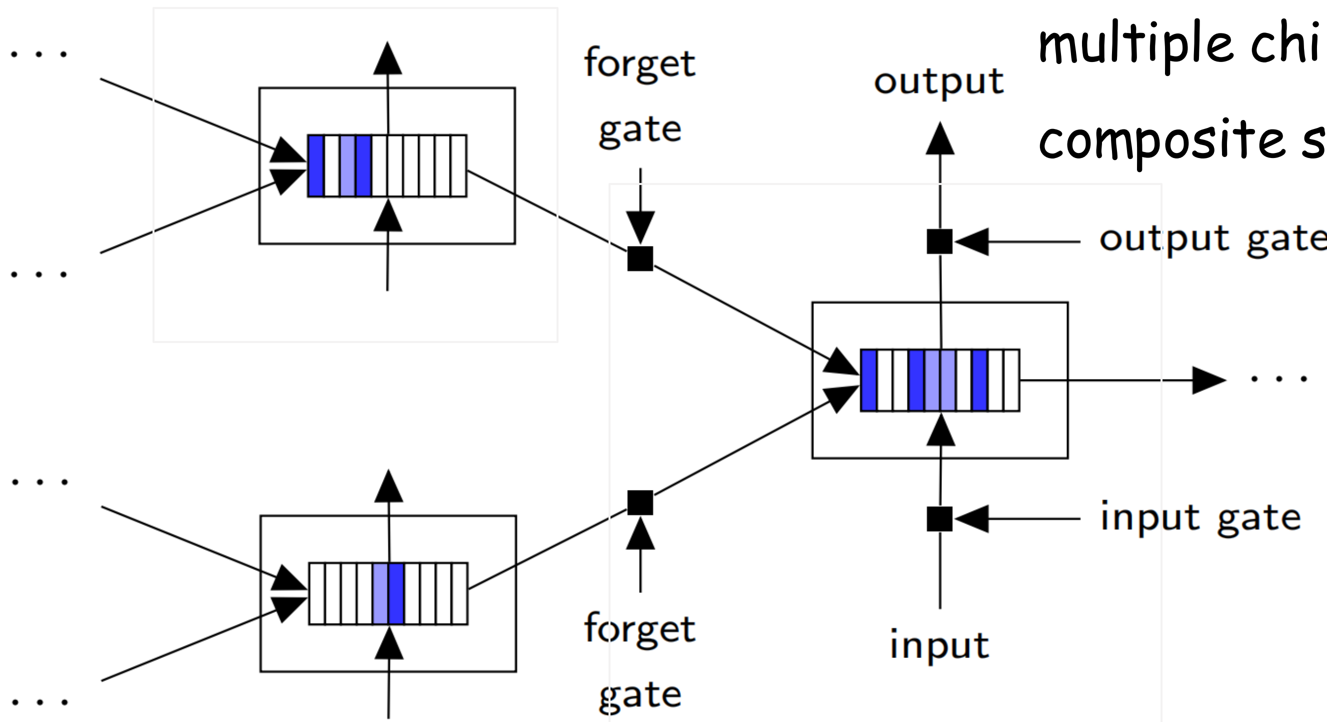
改进：Tree-Structured LSTMs

One cell in lstm

no strict word ordering

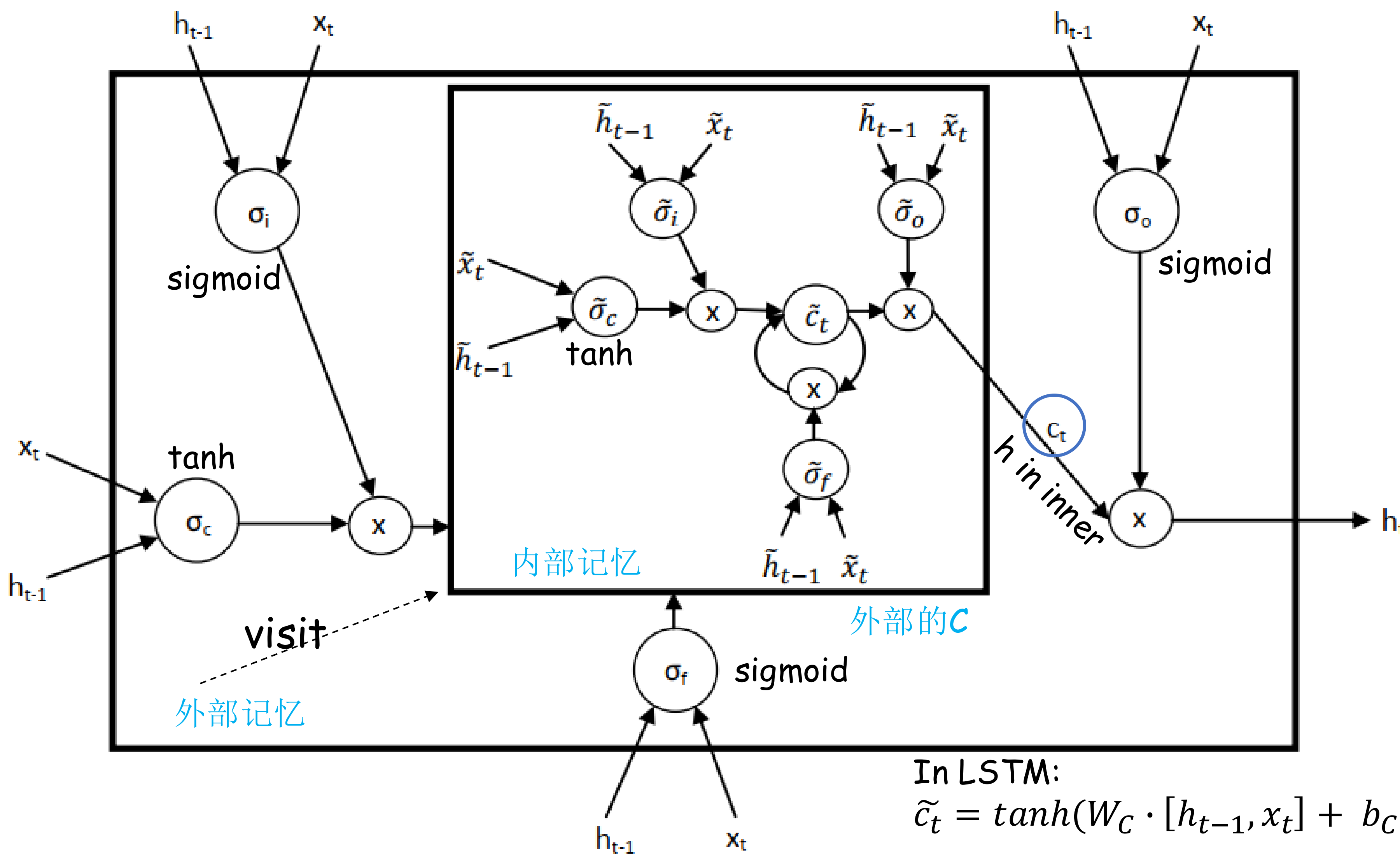
multiple children

composite structure



- 树LSTM允许任意数量的子节点
- 链式LSTM可以看成一个特例

改进：Nested LSTMs (NLSTM)



In LSTM:

$$\tilde{c}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

$$h_t = o_t \odot \tanh(c_t)$$

Moniz and Krueger, 2017. ACML. Nested LSTMs

改进：Nested LSTMs (NLSTM)

- NLSTM

Recall in normal LSTM:

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

- inner: similar to normal LSTM

$$\widetilde{\boldsymbol{i}}_t = \widetilde{\boldsymbol{\sigma}}_i(\widetilde{\boldsymbol{x}}_t \widetilde{\boldsymbol{W}}_{xi} + \widetilde{\boldsymbol{h}}_{t-1} \widetilde{\boldsymbol{W}}_{hi} + \widetilde{\boldsymbol{b}}_i)$$

$$\widetilde{\boldsymbol{f}}_t = \widetilde{\boldsymbol{\sigma}}_f(\widetilde{\boldsymbol{x}}_t \widetilde{\boldsymbol{W}}_{xf} + \widetilde{\boldsymbol{h}}_{t-1} \widetilde{\boldsymbol{W}}_{hf} + \widetilde{\boldsymbol{b}}_f)$$

$$\widetilde{\boldsymbol{c}}_t = \widetilde{\boldsymbol{f}}_t \odot \widetilde{\boldsymbol{c}}_{t-1} + \widetilde{\boldsymbol{i}}_t \odot \widetilde{\boldsymbol{\sigma}}_c(\widetilde{\boldsymbol{x}}_t \widetilde{\boldsymbol{W}}_{xc} + \widetilde{\boldsymbol{h}}_{t-1} \widetilde{\boldsymbol{W}}_{hc} + \widetilde{\boldsymbol{b}}_c)$$

$$\widetilde{\boldsymbol{o}}_t = \widetilde{\boldsymbol{\sigma}}_o(\widetilde{\boldsymbol{x}}_t \widetilde{\boldsymbol{W}}_{xo} + \widetilde{\boldsymbol{h}}_{t-1} \widetilde{\boldsymbol{W}}_{ho} + \widetilde{\boldsymbol{b}}_o)$$

$$\widetilde{\boldsymbol{h}}_t = \widetilde{\boldsymbol{o}}_t \odot \widetilde{\boldsymbol{\sigma}}_h(\widetilde{\boldsymbol{c}}_t)$$

Input of inner comes from outer:

$$\widetilde{\boldsymbol{h}}_{t-1} = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1}$$

$$\widetilde{\boldsymbol{x}}_t = \boldsymbol{i}_t \odot \underline{\boldsymbol{\sigma}_c(\boldsymbol{x}_t \boldsymbol{W}_{xc} + \boldsymbol{h}_{t-1} \boldsymbol{W}_{hc} + \boldsymbol{b}_c)}$$

- outer:

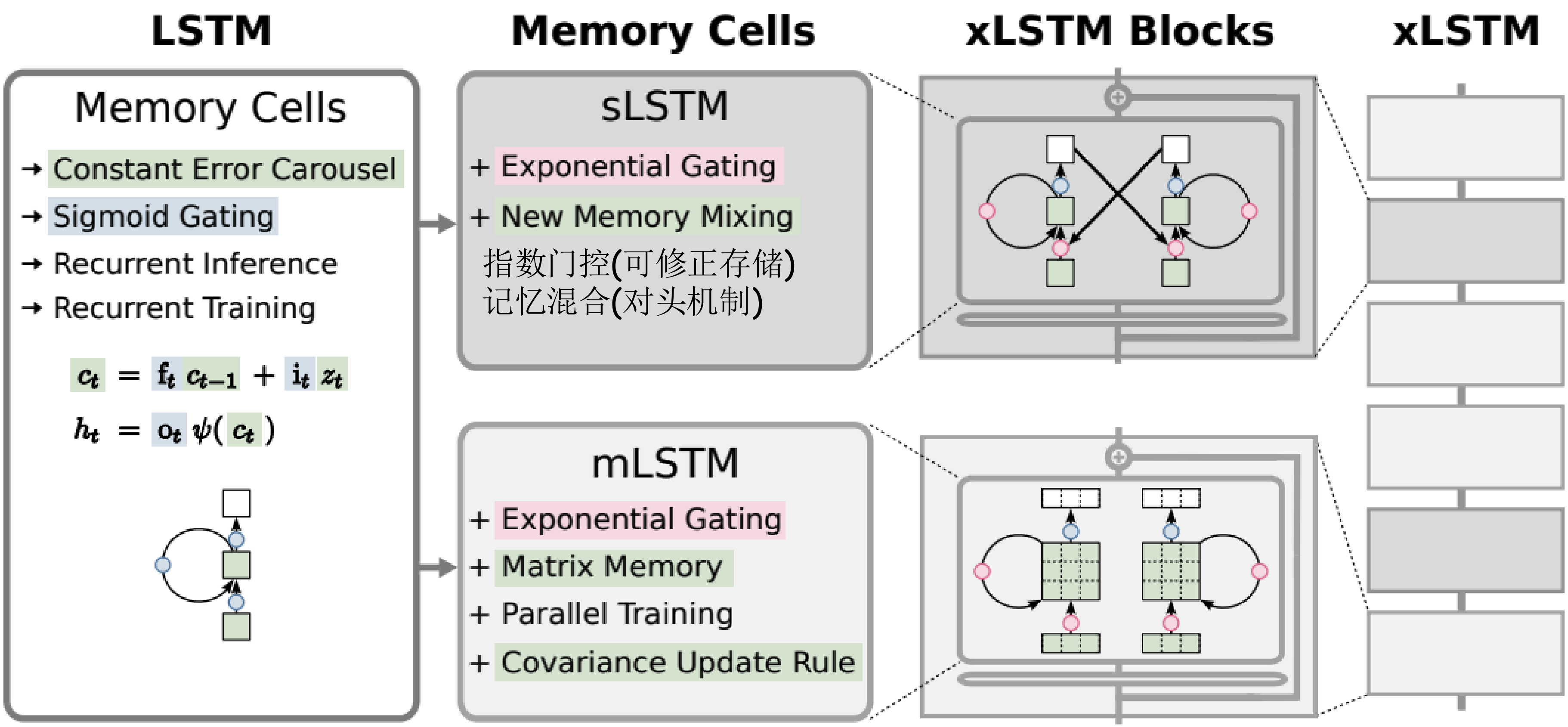
$$\boldsymbol{c}_t = \widetilde{\boldsymbol{h}}_t$$

In other words:

$$\underbrace{c_t}_{\text{普通LSTM}} = \underbrace{m_t}_{\text{内部记忆函数}}(f_t \odot c_{t-1}, i_t \odot \underline{g_t})$$

改进：xLSTM

- xLSTM: Extended Long Short-Term Memory



用矩阵代替标量记忆

将 sLSTM 和 mLSTM 集成到残差块中，构建深层网络

https://arxiv.org/pdf/2405.04517