LSTM 原理

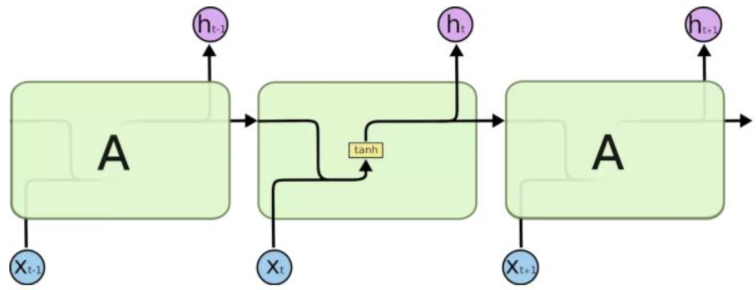
一、LSTM

LSTM (long short term memory)，即长短期记忆网络，是一种 RNN 特殊类型，其目的是解决 RNN 中长时间依赖问题，其核心是通过内置门控达到调节信息流的目的。引用参考文献[3]中一段话说明 LSTM 重要性

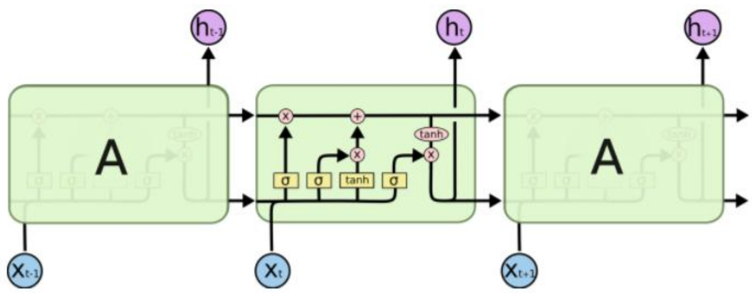
These gates can learn which data in a sequence is important to keep or throw away. By doing that, it can pass relevant information down the long chain of sequences to make predictions. Almost all state of the art results based on recurrent neural networks are achieved with these two networks. LSTM's and GRU's can be found in speech recognition, speech synthesis, and text generation. You can even use them to generate captions for videos.

二、LSTM 结构图

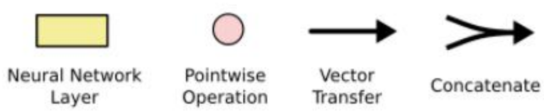
所有的 RNN 都具有一种重复神经网络模块的链式形式。在标准的 RNN 中，这个重复的模块只有一个非常简单的结构，例如一个 tanh 层，一般结构如下



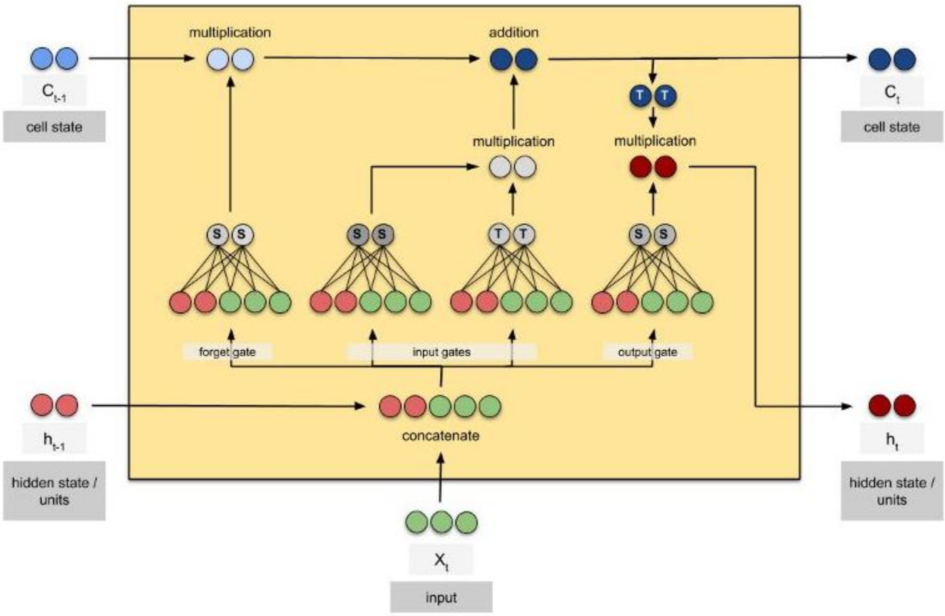
而 LSTM 区别在于设计了不同的重复模块，这个重复模块内部并非单一神经网络层，而是有四层，且不同层间以特殊的方式进行交互，结构如下



其中

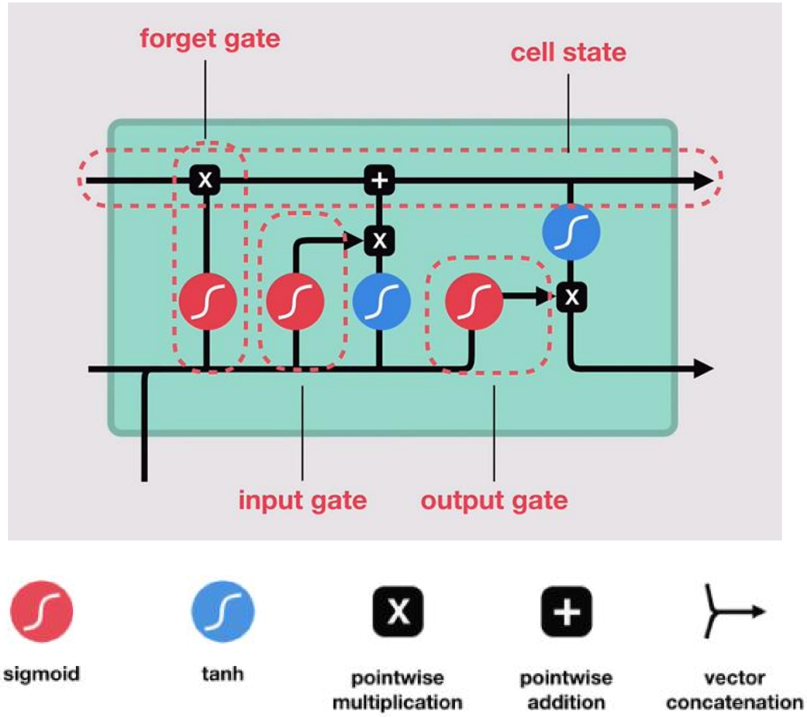


节点图表示如下



multiplication
addition
C_{t-1}
cell state
C_t
cell state
multiplication
multiplication
forget gate
input gates
output gate
h_{t-1}
hidden state / units
concatenate
h_t
hidden state / units
X_t
input

门控示意图

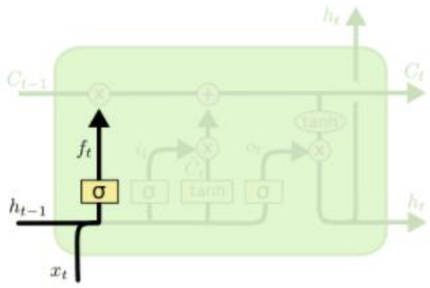


forget gate
cell state
input gate
output gate
sigmoid
tanh
pointwise multiplication
pointwise addition
vector concatenation

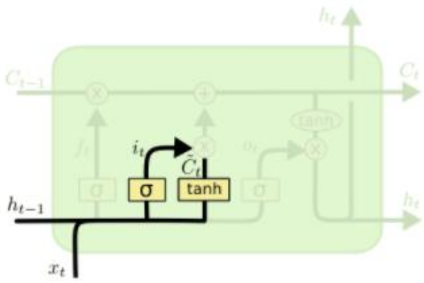
伪码

```python
def LSTMCELL(prev_ct, prev_ht, input):
    combine = prev_ht + input
    ft = forget_layer(combine)
    candidate = candidate_layer(combine)
    it = input_layer(combine)
    Ct = prev_ct * ft + candidate * it
    ot = output_layer(combine)
    ht = ot * tanh(Ct)
    return ht, Ct



ct = [0, 0, 0]
ht = [0, 0, 0]

for input in inputs:
    ct, ht = LSTMCELL(ct, ht, input)
```

三、前向传播函数方程

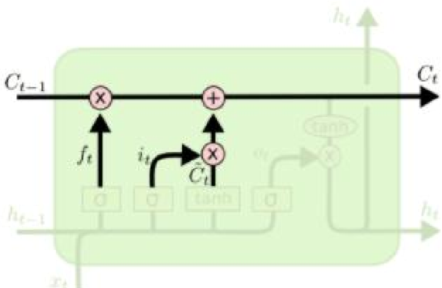


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

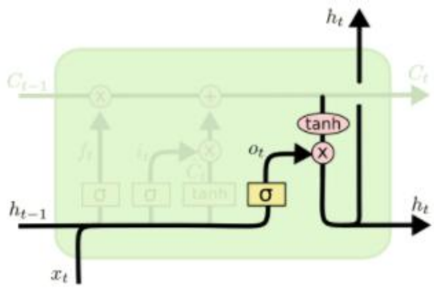


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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



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



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

σ 为 sigmoid 函数， tanh 为 tanh 函数。

四、核心思想—门控原理

The core concept of LSTM's are the cell state, and it's various gates. The cell state act as a transport highway that transfers relative information all the way down the sequence chain. You can think of it as the "memory" of the network. The cell state, in theory, can carry relevant information throughout the processing of the sequence. So even information from the earlier time steps can make it's way to later time steps, reducing the effects of short-term memory. As the cell state goes on its journey, information get's added or removed to the cell state via gates. The gates are different neural networks that decide which information is allowed on the cell state. The gates can learn what information is relevant to keep or forget during training.

1. 遗忘门 (Forget gate)

利用 sigmoid 函数取值范围为[0,1]，越接近 0 遗忘程度越大，与先前单元信息做点积，sigmoid 为 0 时遗忘先前单元中对应部分的信息。

2. 输入门 (Input Gate)

先前单元中隐藏层信息与当前单元输入信息结合，该结合信息经过激活函数为 tanh 网络将其映射到[-1,1]，同时该结合信息经过激活函数 sigmoid 将其映射到[0,1]，前述两者映射后的信息做点积，得到调节后的输入信息，将调节后信息加入当前单元状态。

3. 单元状态 (Cell State)

该单元的信息传递纽带或者枢纽，同时包含了先前单元的信息，含有记忆功能。

4. 输出门 (Output Gate)

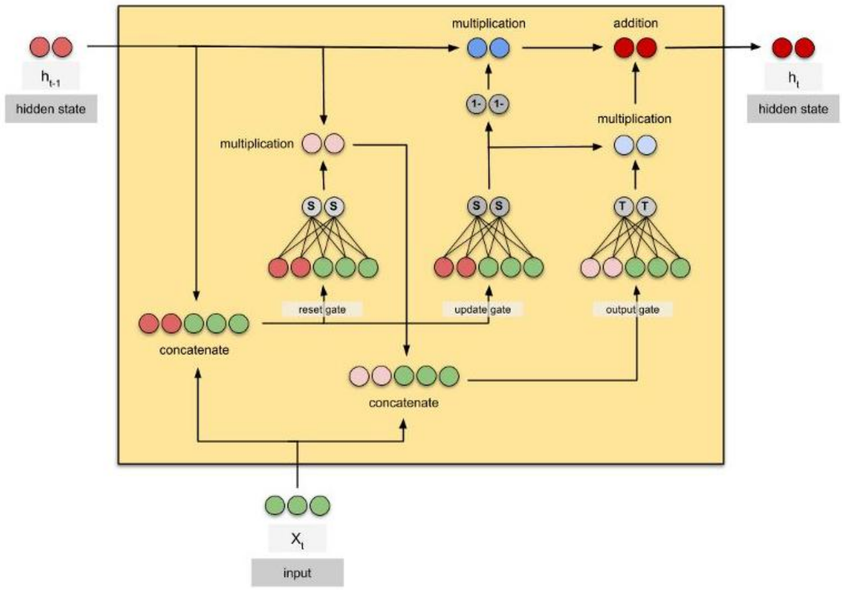
通过 sigmoid 函数调节该单元输出到下一个单元隐藏层的信息，以及该单元预测输出的信息。

总之：

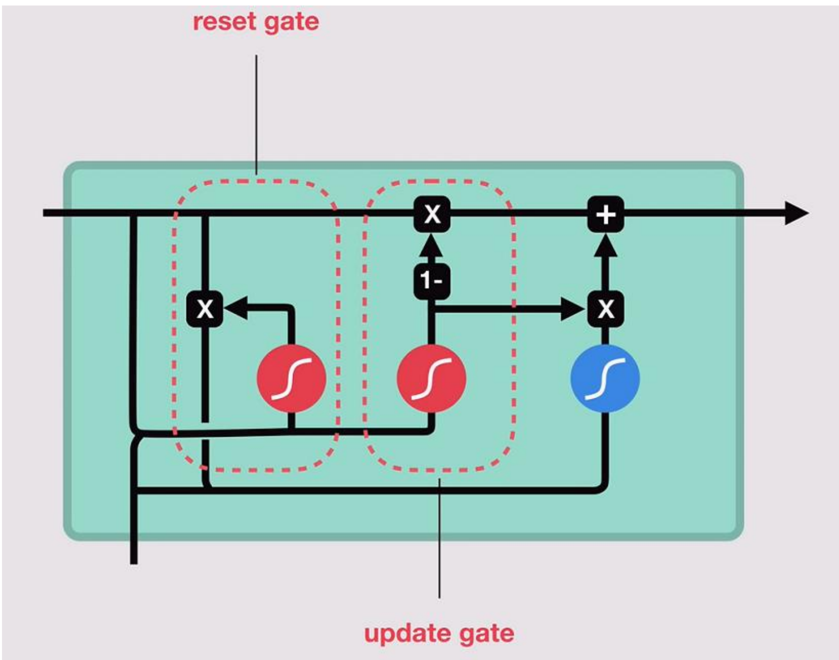
To review, the Forget gate decides what is relevant to keep from prior steps. The input gate decides what information is relevant to add from the current step. The output gate determines what the next hidden state should be.

附录 1

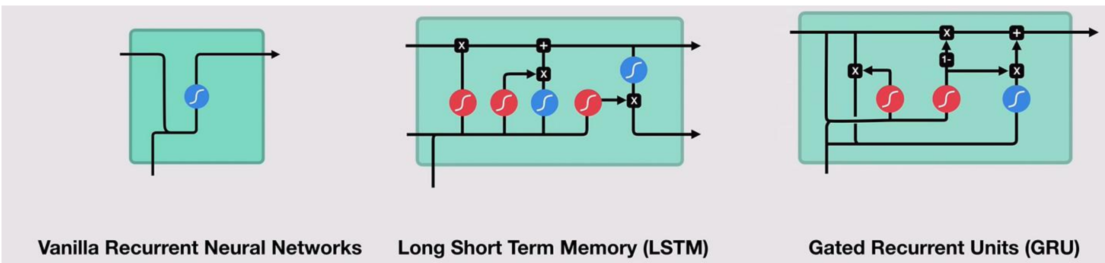
GRU 节点图



GRU 示意图



Vanilla RNN 、LSTM 与 GRU



参考文献

[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[2] https://www.lizenghai.com/archives/24286.html

[3] https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21