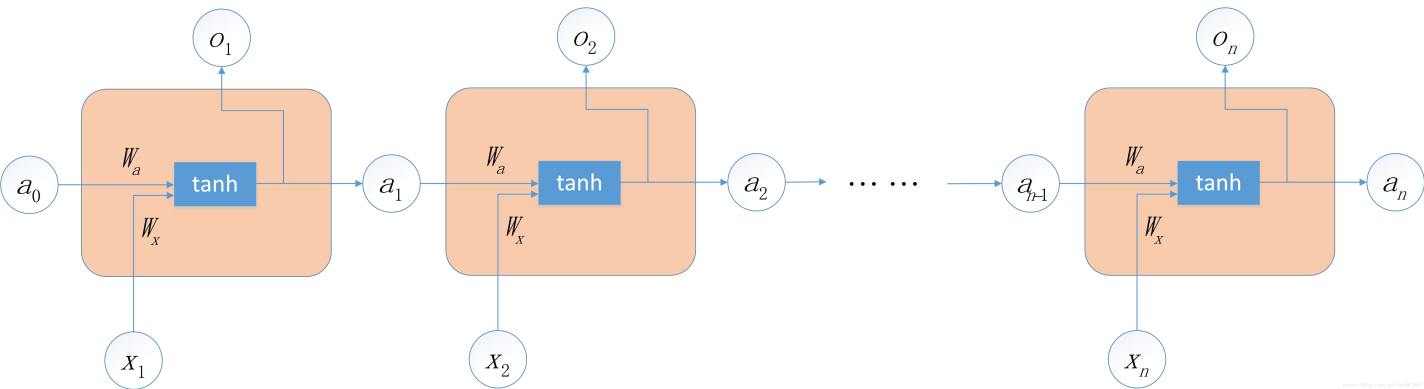


# BiLSTM (双向长短时记忆网络)

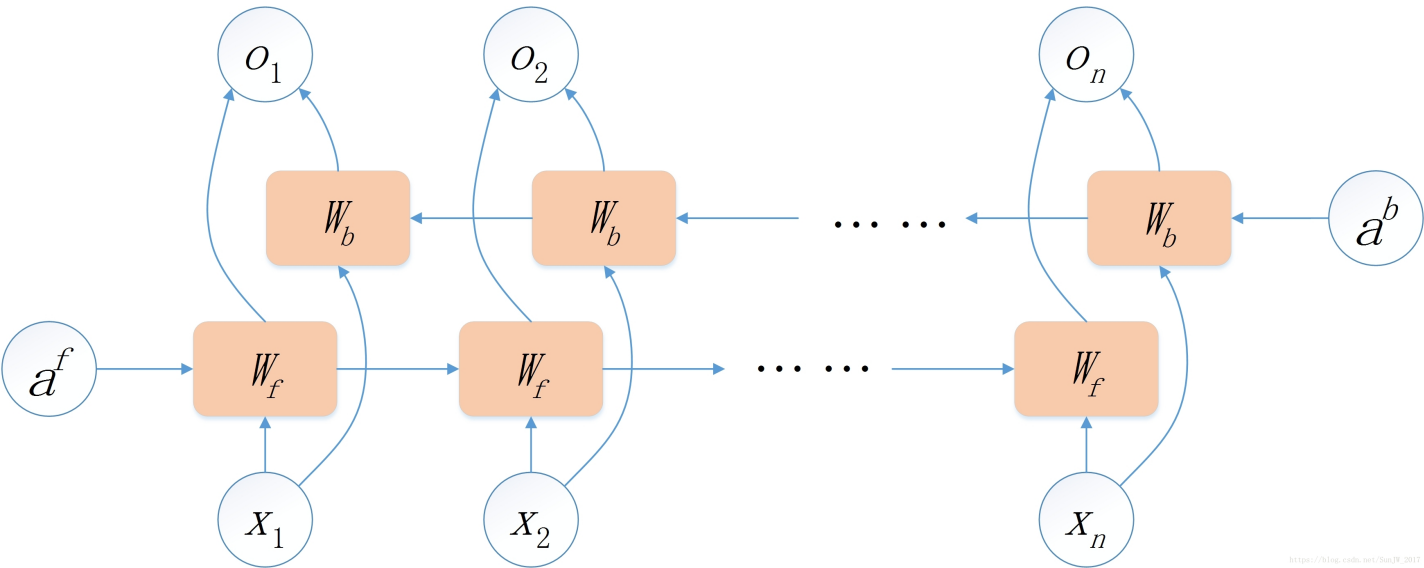
## RNN

普通的神经网络不容易处理训练样本输入为连续序列的情况，比如一段语音、一段文字，于是 RNN 就诞生了。RNN 的运行机制如下图：



从图中看出，与普通神经网络不同的一点在于，神经元接受两个值：一个是当前时刻的输入  $x_t$ ，一个是上一个神经元的输出  $a_{t-1}$ ，通过将前一时刻的运算结果添加到当前的运算中，从而实现“考虑上文信息”的功能。

如果不仅要考虑上文的信息，还要考虑下文的信息，可以使用 BiRNN，BiRNN 正反向分别计算，然后将正反向结果堆叠，生成最终的结果：



## RNN 存在的问题

对某些简单的问题，可能只需要最后输入的少量时序信息即可解决。但对于某些复杂问题，可能需要更早的

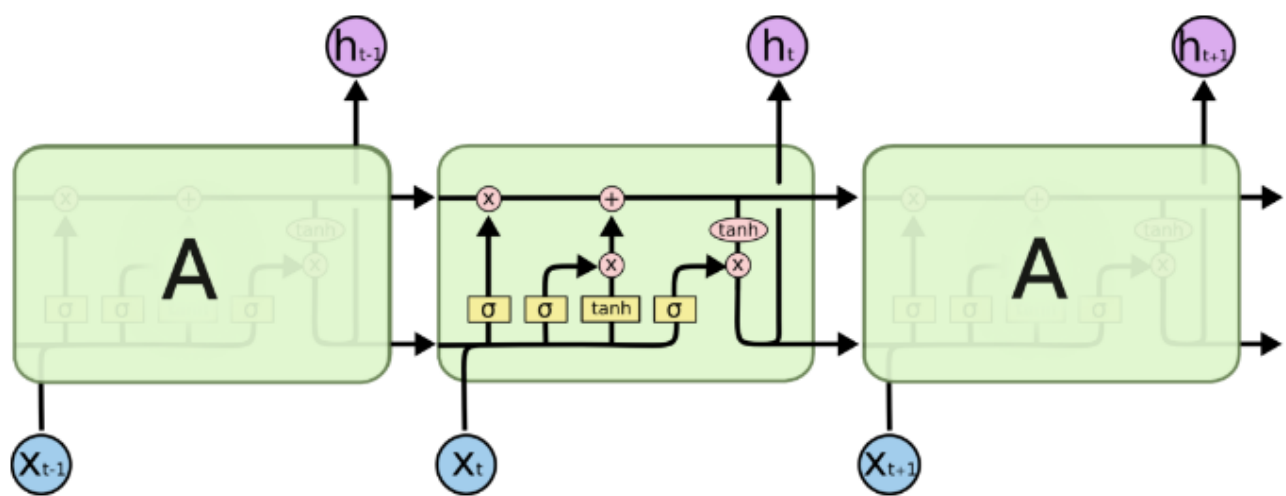
一些信息，而 RNN 难以记忆间隔太久的输入信息（梯度消失），这时候就需要用到 LSTM 了。

# LSTM

LSTM 最核心的是引入了“门”的概念，门实际上是一种全连接层，它的输入是一个向量，输出是一个0到1之间的实数向量。假设  $W$  是门的权重向量， $b$  是偏置项，那么门可以表示为：

$$g(x) = \sigma(Wx + b)$$

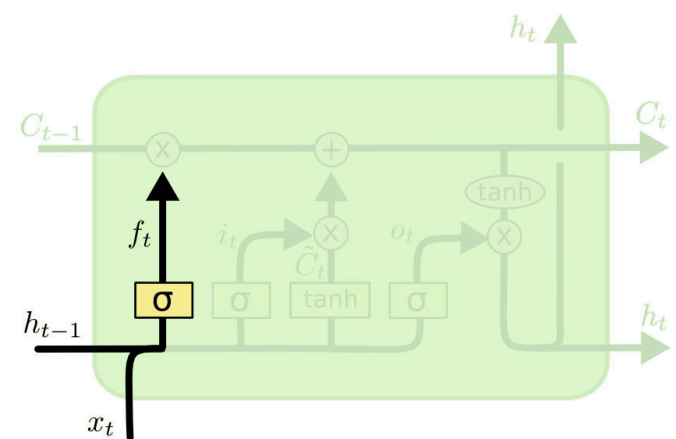
结果表示有多少信息量可以通过这个结构。在 LSTM 中，一共有3种门结构，分别是遗忘门、输入门和输出门，如图所示：



可以看出，LSTM 每个神经元都有三个输入， $X_t$  是该时刻新加入的信息， $h_{t-1}$  和  $C_{t-1}$  是上文信息的表示，输出  $C_t$  和  $h_t$ 。

## 1. 遗忘门

遗忘门在 LSTM 中以一定概率控制是否遗忘上一层的隐藏细胞状态，遗忘门子结构如下图所示：

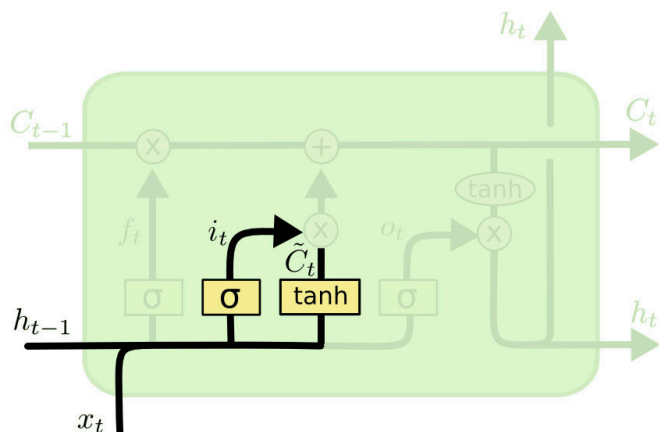


$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

## 2. 输入门

两个输入合并的向量分别进行了  $\sigma$  变换和  $\tanh$  变换，成为  $i_t$  和  $\tilde{C}_t$  进入输入门。其中  $i_t$  的值用于决定是否

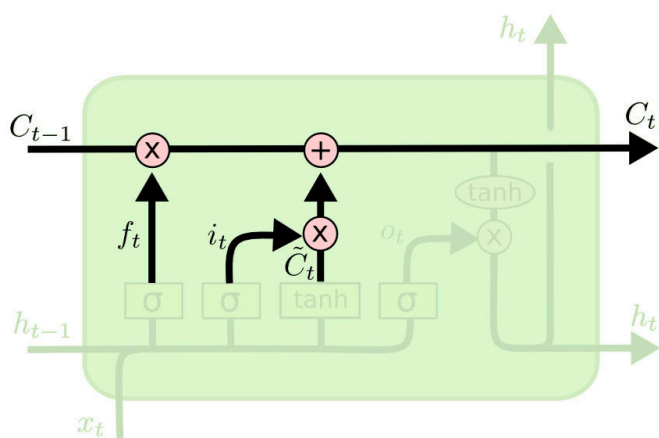
要接受输入信息（即 $\tilde{C}_t$ ）。



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

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

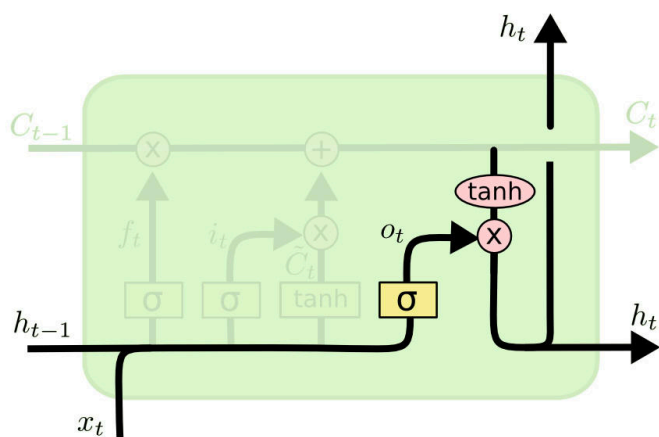
之后，根据 $f_t$ 的值决定是否保留上一时刻隐藏层的输出( $C_{t-1}$ )，再将经过 $i_t$ 缩放之后的输入 $\tilde{C}_t$ 累加到 $C_{t-1}$ 中，成为 $C_t$ 。



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

### 3. 输出门

在输出门中， $o_t$ 决定是否将经过 $\tanh$ 变换的 $C_t$ 输出，作为下一时刻输入的一部分。



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

$$h_t = o_t * \tanh(C_t)$$