**LSTM简介（Long Short Term Memory networks）**

它是一种特殊的RNN，可以学习长期依赖。

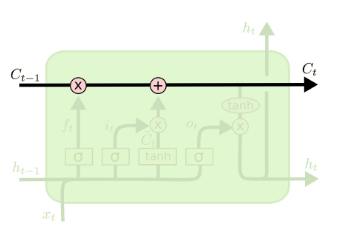
LSTM被设计用来规避长期依赖问题。对于长期的信息记忆是它默认的本质行为，而不是它努力在学习的东西。

**对比标准RNN：**

所有的递归神经网络都有一个神经网络的重复模块的链式形式。在标准RNN中，重复的模块是一个非常简单的结构，例如单层的tanh层。

LSTM同样有这样类似的结构，但是重复的模块不同。不同于单个神经网络层，LSTM有4个以一种非常特殊的方式交互的部件。

**LSTM的核心思想：**

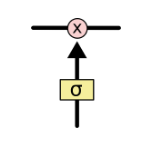
****

LSTM的关键是**单元状态**(cell state)，通过图片上方的水平线。

单元状态是一种传动带。它笔直贯穿整个链式结构，在这个过程中只与少数单元线性交互。

LSTM通过一种被称为门的控制结构可以为单元状态添加或移除信息。

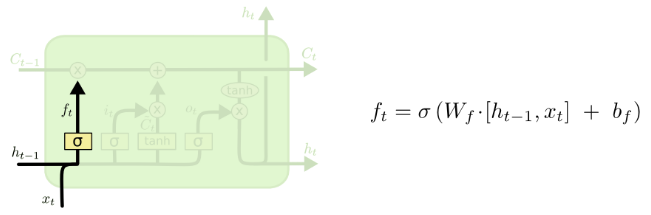
这些门是一种有选择的让信息通过的方式。他们是由sigmoid神经网络层和点乘法运算组成的。如下图所示：



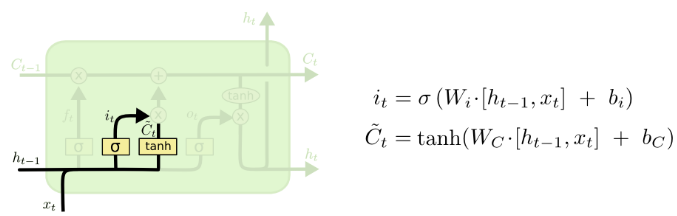
Sigmoid层输出的值在0-1之间，用以描述每一个组件应该通过多少信息。0表示“都不通过”，1表示“全都通过”。一个LSTM拥有3个这样的门来保护和控制单元状态。

**逐步进入LSTM**

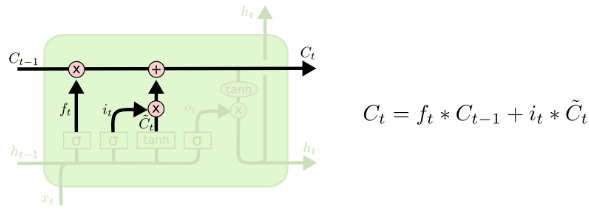
在LSTM中，第一步是要确定哪些信息需要从单元状态中丢弃。这个决定由一个被称为“忘记门”(forget gate layer)的sigmoid层决定。这一层会观察ht-1和xt并为t-1阶段的每一个单元状态输出一个在0-1之间的数。1表示全部保留，0表示全部丢弃。



第二步是要确定那些信我们需要存储在单元状态中。这一部分由2部分组成。首先，一个被称为“输入门”(input gate layer)的sigmoid层决定哪些值将被更新。然后，一个tanh层会创建一个新的候选值向量，，这些值将加到状态上。下一步，我们合并这两个值去构建一个新的更新值去更新状态。



接下来要做的，就是真正的更新旧的产生新的。将乘以ft来决定前一阶段哪些应该忘记。然后加上it \* 。这就是新的候选值，我们决定更新多少单元状态值。



最后，我们要决定我们要输出什么。这个输出将基于我们的单元状态，但是一个过滤版的。首先，我们执行一个sigmoid层，由它决定单元状态的哪些部分要输出。然后，我们把单元状态通过tanh(把值压缩在-1到1)sigmoid门的输出，以便我们仅输出我们想要输出的那部分。

