神经网络分为两种，前馈型神经网络与后馈型神经网络（即递归型神经网络）；  
前馈型神经网络典型的有：卷积神经网络（Convolutional Neural Networks，  
CNN）；后馈型（递归型）神经网络的典型有：循环神经网络（Recurrent  
Neural Network, RNN）；

LSTM是一种改进之后的循环神经网络，即它属于RNN，也属于递归型神经网络（Recursive Neural Network, RNN）。在LSTM的例子中，每一个“门”使用的参数是一致的，可以通过递归调用自身实现。当然，一切递归本质上都是循环，说是循环网络也没问题。不过大家都叫RNN

前馈神经网络（Feedforward Neural Network，缩写为FNN）是一类最简单的神经网络，也是最早出现的神经网络之一。它主要由若干层神经元组成，各层神经元之间没有反馈连接，信息只能向前流动，从而得名“前馈”。通常，输入层负责接收原始数据，中间隐藏层则对数据进行一系列非线性映射和特征提取，输出层则产生神经网络的输出结果。

相比之下，反馈神经网络（Recurrent Neural Network，缩写为RNN），具备反馈（或循环）连接，即输出可以在后续的时间步骤被送回给网络的输入端。这种机制使得RNN能够处理那些需要记忆和上下文信息的复杂任务，例如语音识别、机器翻译、自然语言理解等。然而，由于反馈神经网络的训练相对较为复杂，容易产生梯度消失和爆炸等问题，因此虽然RNN在某些场合具备优势，但并不是适用于所有的神经网络任务。

总之，前馈神经网络和反馈神经网络在结构和任务上都存在一些差别，需要根据具体情况灵活选择。

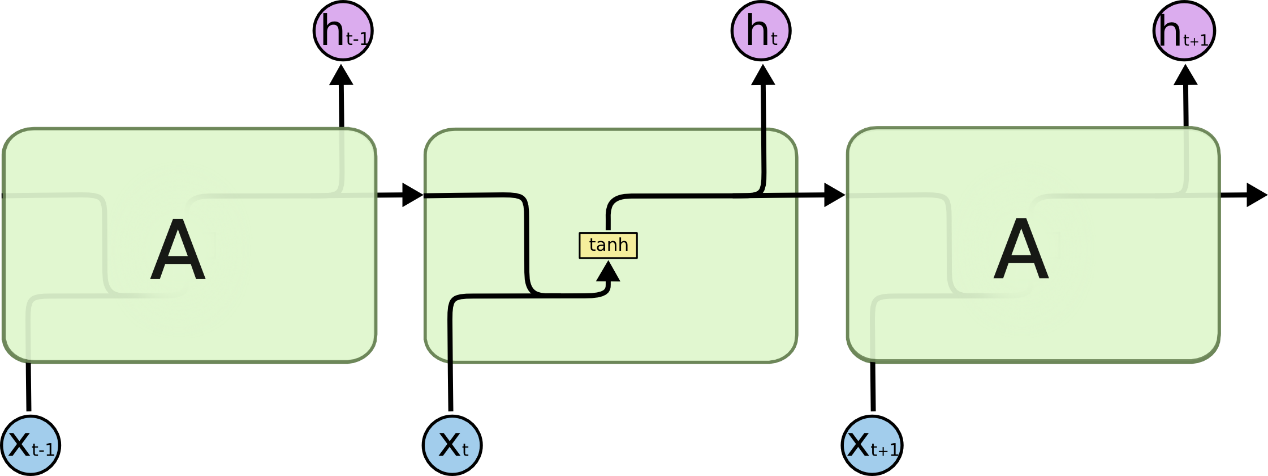
**LSTM**

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

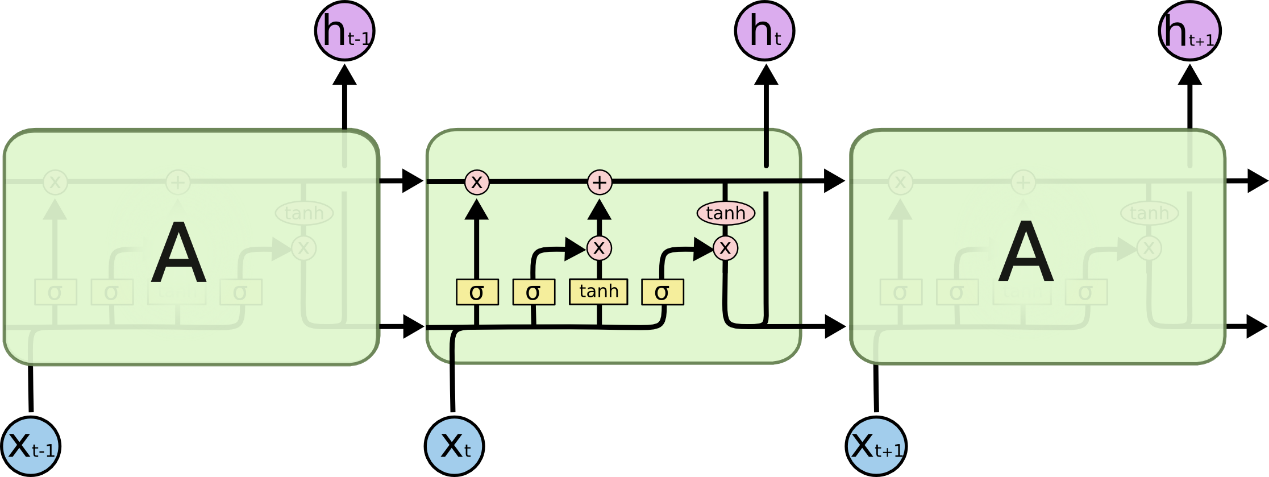
LSTM，全称 Long Short Term Memory (长短期记忆) 是一种特殊的**递归神经网络**。这种网络与一般的前馈神经网络不同，LSTM可以利用时间序列对输入进行分析

LSTM从被设计之初就被用于解决一般递归神经网络中普遍存在的**长期依赖问题**，使用LSTM可以有效的传递和表达长时间序列中的信息并且不会导致长时间前的有用信息被忽略（遗忘）。与此同时，LSTM还可以解决RNN中的梯度消失/爆炸问题。

一个普通的，使用tanh函数的RNN可以这么表示：



与上图朴素的RNN相比，单个LSTM单元拥有更加复杂的内部结构和输入输出：



在上图中，每一个红色圆形代表对向量做出的操作（pointwise operation， 对位操作），而黄色的矩形代表一个神经网络层，上面的字符代表神经网络所使用的激活函数

**point-wise operation 对位操作**

如果我要对向量<1, 2, 3> 和 <1, 3, 5>进行逐分量的相乘操作，会获得结果 <1, 6, 15>

**layer 函数层**

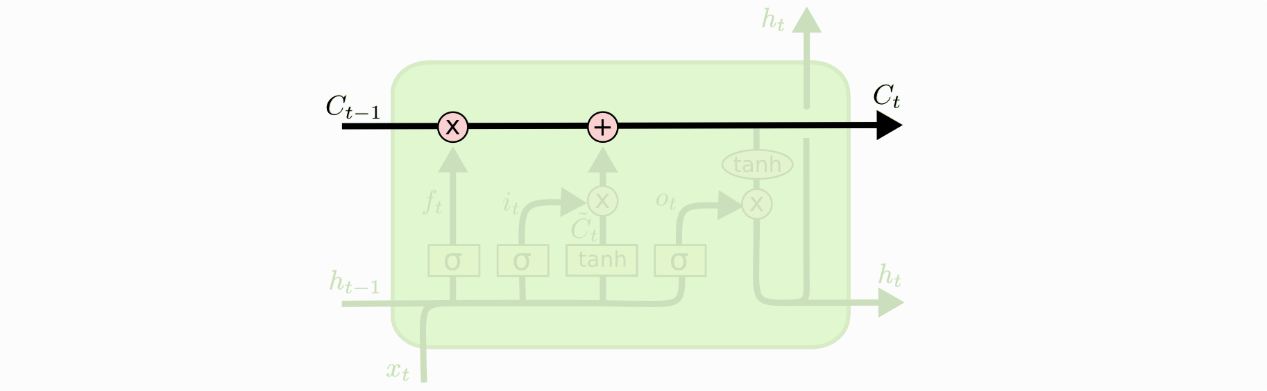
一个函数层拥有两个属性：权重向量(Weight) 和 偏置向量(bias)，对于输入向量***A***的每一个分量*i*， 函数层会对其进行以下操作(假设激活函数为**F(x)**):



常见的激活函数（也就是套在最外面的**F(x)**）有ReLU(线性修正单元)，sigmoid（写作），和tanh

LSTM 的关键是单元状态，即贯穿图表顶部的水平线。

细胞状态有点像传送带。它直接沿着整个链条向下运行，只有一些轻微的线性相互作用。信息很容易原封不动地沿着它流动。



LSTM确实能够删除或添加信息到细胞状态，由称为门的结构仔细调节。

门是一种选择性地让信息通过的方式。它们由 sigmoid 神经网络层和逐点乘法运算组成。



sigmoid 层输出介于 0 和 1 之间的数字，描述每个组件应通过多少。值为零表示“不让任何东西通过”，而值为一表示“让一切都通过！

LSTM 具有三个这样的门，用于保护和控制单元状态。