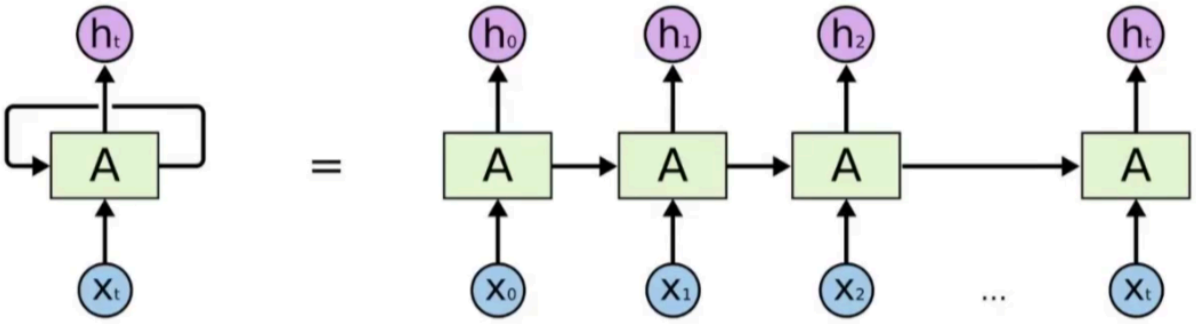
# 第五节 LSTM（长短期记忆网络）及ELMO模型（双向LSTM）

## RNN的问题

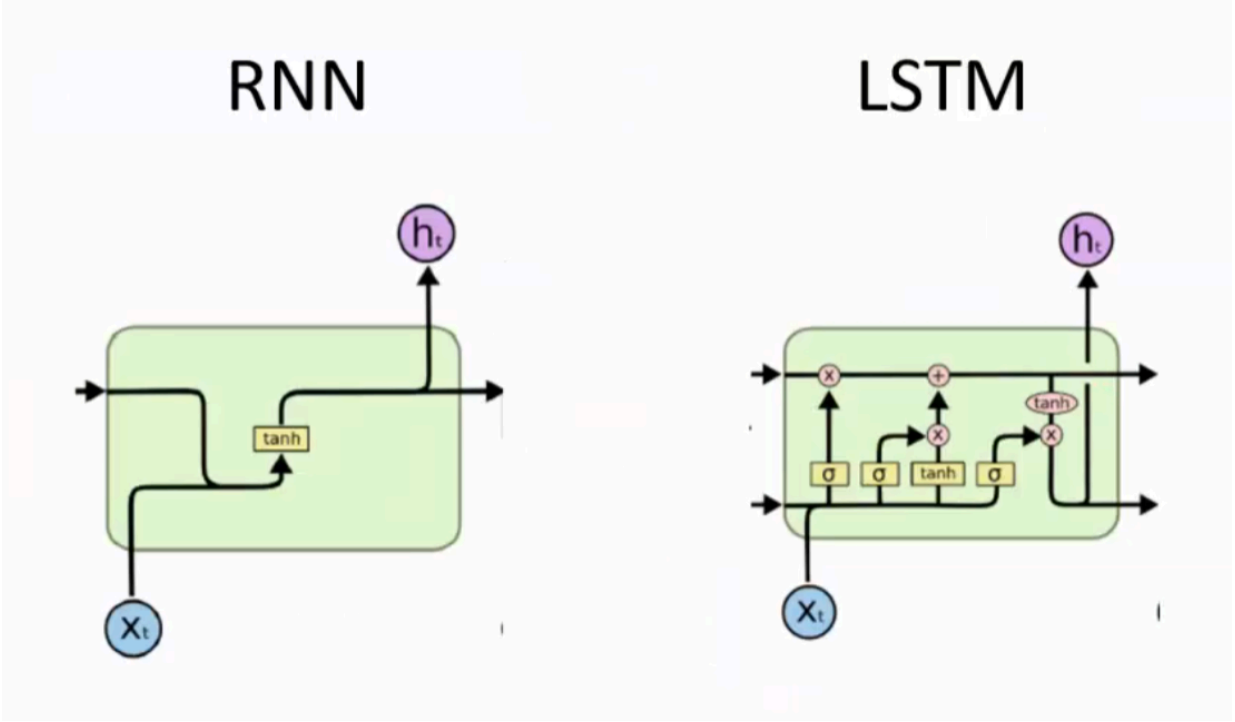



假设我们现在要做一个任务，比如我说了一句话，“我曾经去过日本旅游，也去过巴黎，还去过泰国。但是我是中国人，我爱_____”，就是做一个生成任务，预测一下我爱的后面到底是什么。

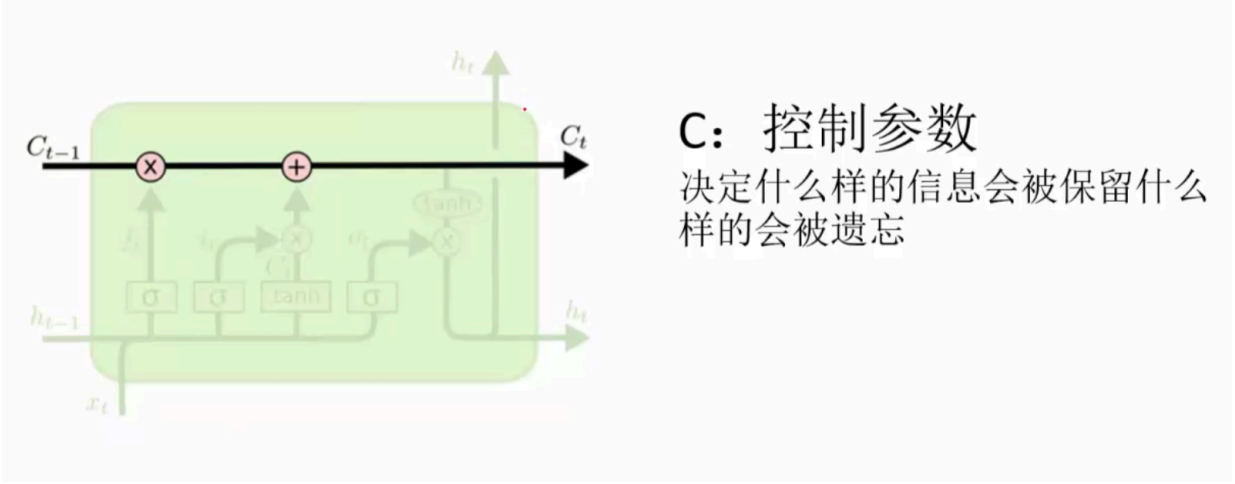
假如让RNN来做这个任务，它会把所有的空格前面的信息都考虑进去，但是前面的那些内容对这个填空其实没有作用，甚至会导致误判。而且这还是比较短的句子，一旦达到几百几千个字，把所有的信息都同时融入到最终的输出结果里面显然是不合理的，所以RNN模型就存在着这么一个缺陷。

## LSTM

针对上面所说的RNN的问题，就有人提出了LSTM。



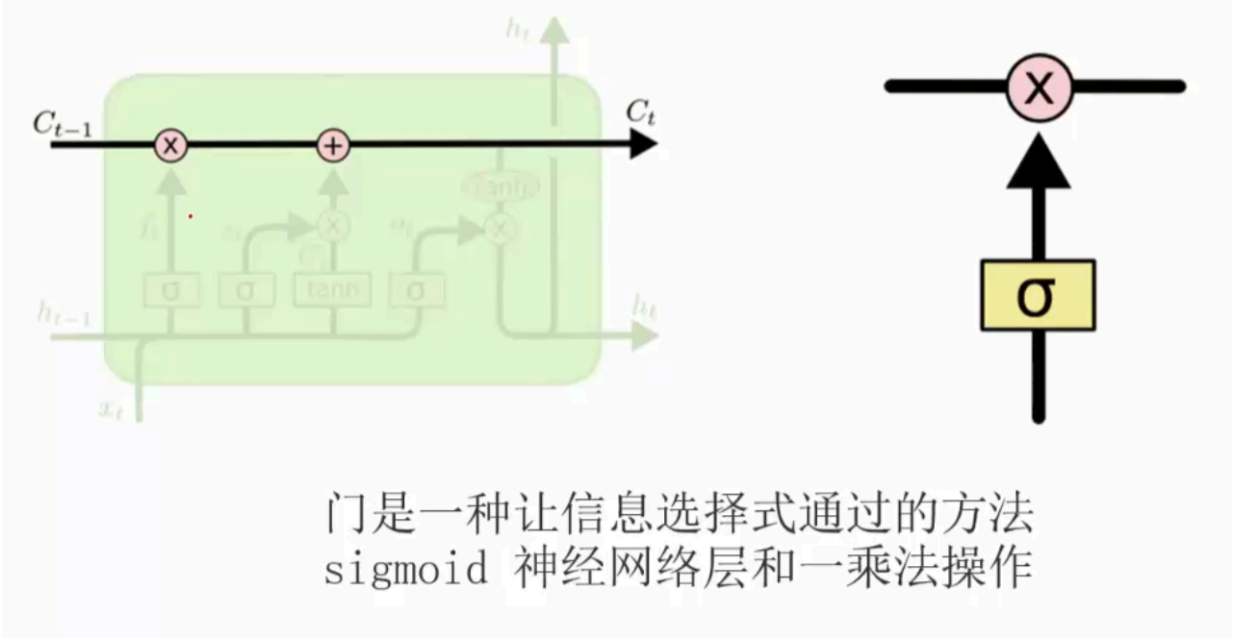

之前我们说RNN的问题在于它记录了太多的信息，这些信息对我们当下要执行的任务不一定是有用的，所以LSTM的目的就是让RNN忘掉一部分冗余的信息，从而提升RNN的实际效果。

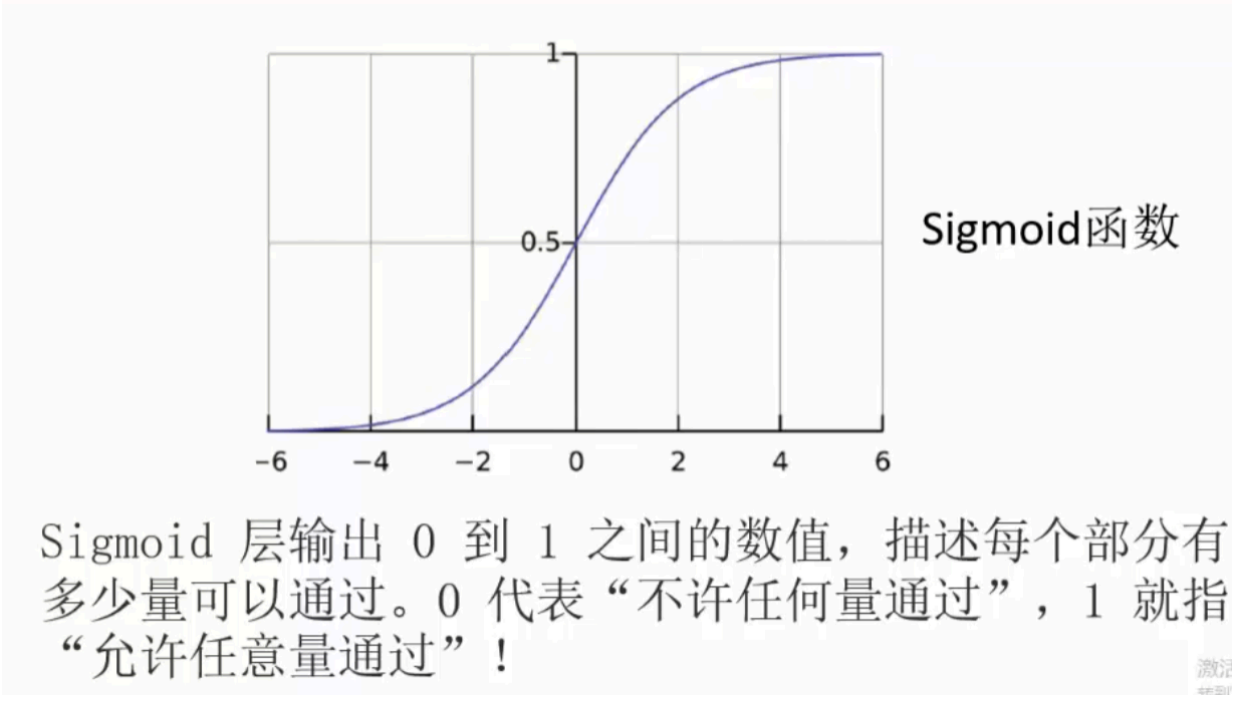


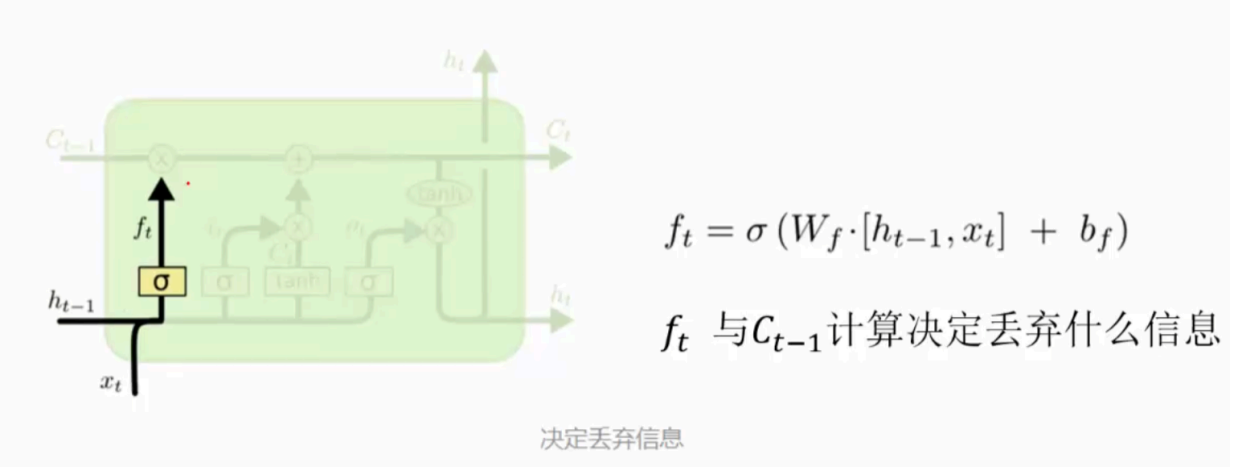

LSTM里面有一个重要的控制参数C，这个参数的作用就是决定什么样的信息会被保留，什么样的信息会被遗忘，这个参数不是一成不变的，而是随着神经网络的学习不断更新的。

LSTM实际上就是通过各种门去决定信息的去留

**遗忘门**

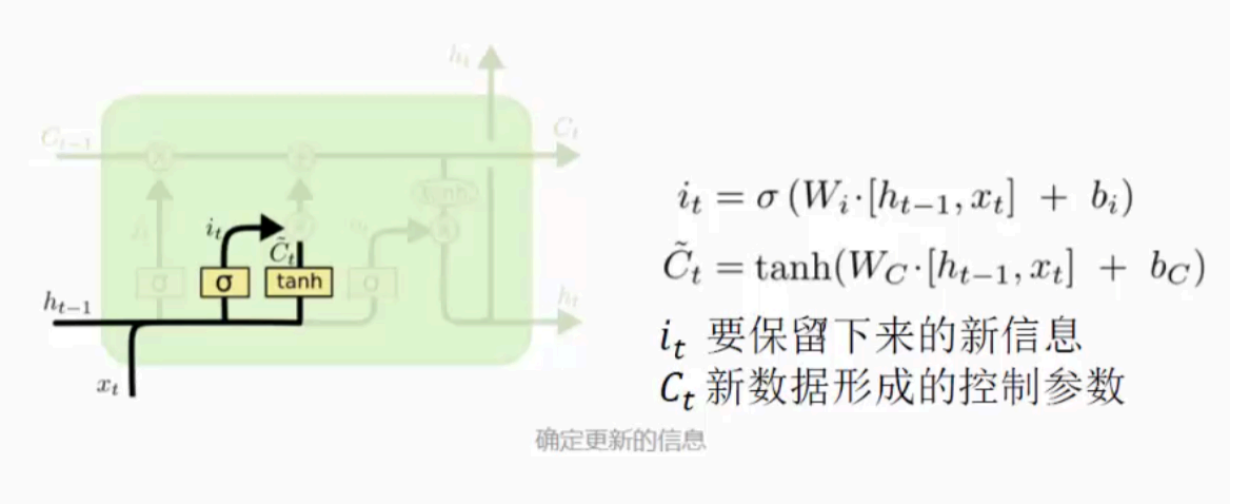





首先第一种门就是遗忘门，遗忘门是通过一个激活函数（一般是sigmoid）和一个乘法去实现的。当一个输入值经过sigmoid函数后，会得到0到1之间的一个概率值，假如这个值接近1，那么它的信息就保留得越多（按比例保留），假如越接近0，那么它的信息被遗忘得就越多。所以通过遗忘门，我们可以做到对信息的一个筛选。保留有用的信息，去除冗余的信息。这里的乘法就是f_t和c_{t-1}做一个乘法，来更新C参数。

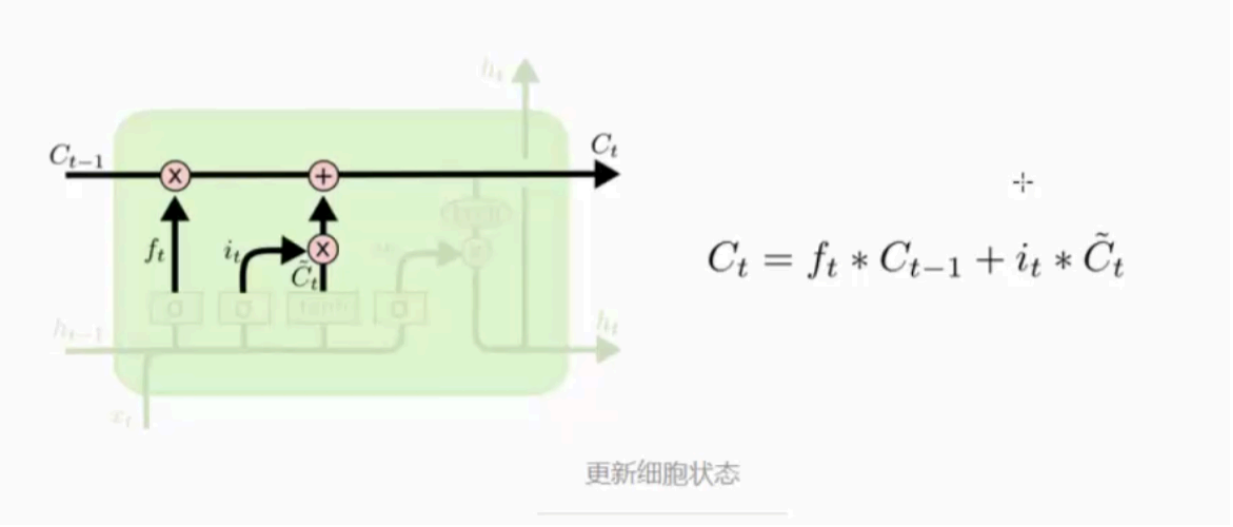
## 输入门




前面遗忘门决定了信息的保留程度，但是新的输入信息也要能够保留下来，所以这里其实起到的作用就是保留新的输入信息。

一部分用sigmoid函数决定哪些特征要更新，另一部分使用tanh函数生成一个新的候选值向量，这个向量就是输入的信息，两个参数相乘就能得到保留了有用特征的输入信息。

## 输出门




最终我们讲之前得到的两个参数相加，就能得到更新后的C_t。

通过上面所说的那些门，我们可以有效地剔除冗余信息，保留有效信息，LSTM相比RNN是一次很大的改进。

## LSTM的缺点

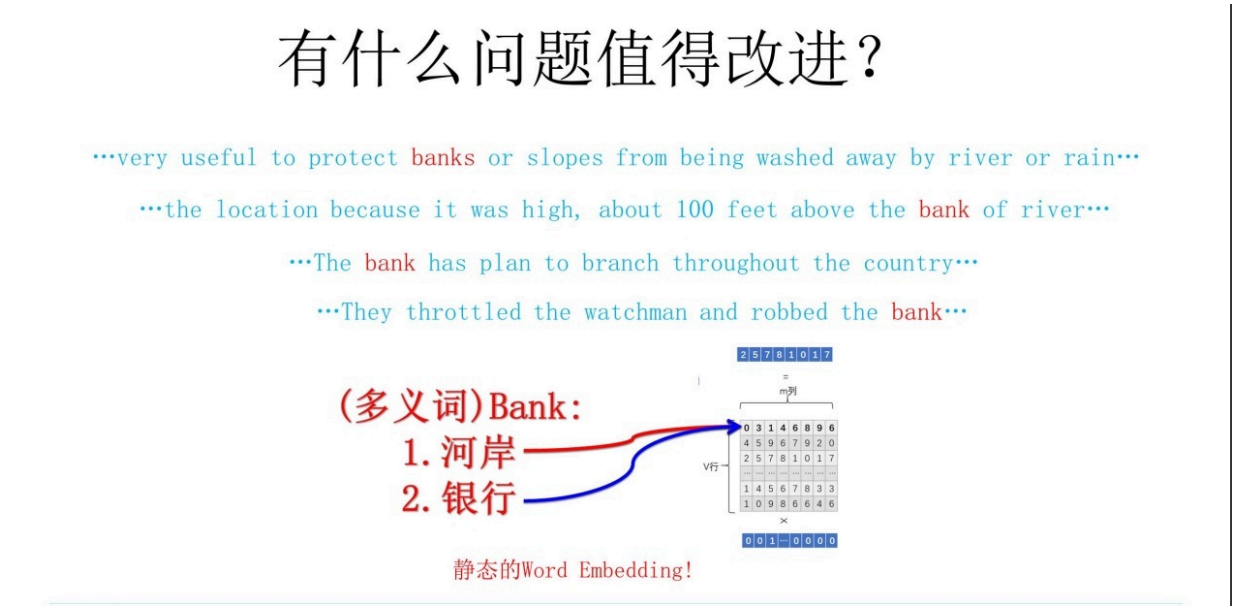
LSTM只是缓解了梯度弥散或者梯度爆炸的问题（或者说长期依赖问题），但是在网络层数比较深的时候，这样的问题依然存在，没有得到根本缓解

而且LSTM仍然保留了RNN的老毛病，就是无法并行

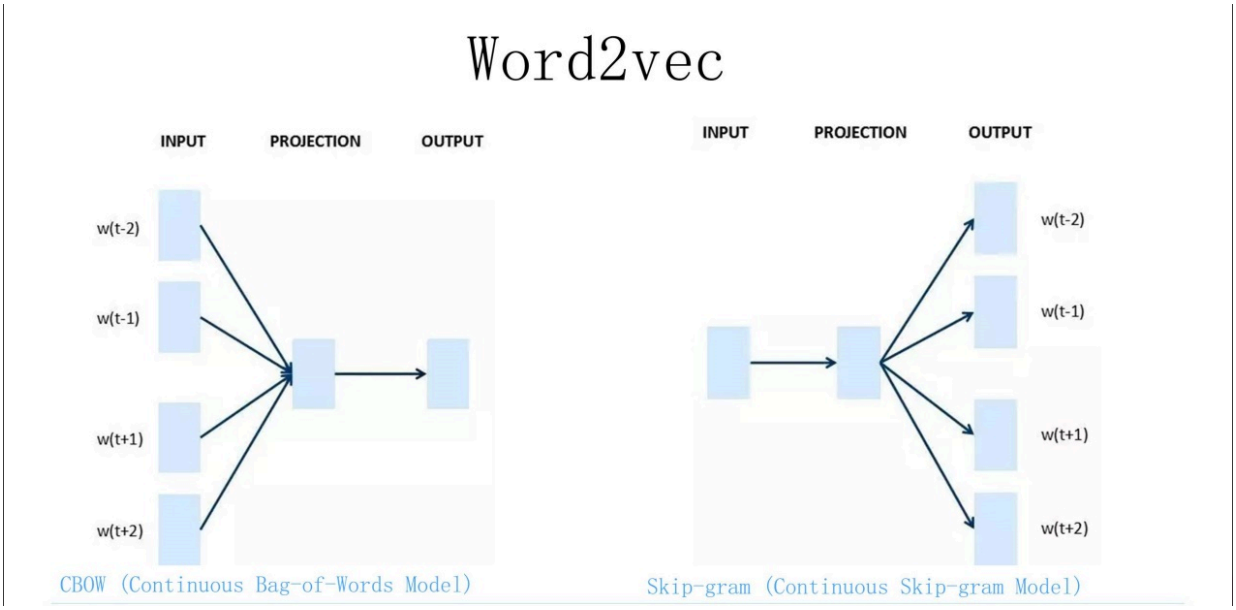
后面的attention会比较好地解决掉梯度弥散或者梯度爆炸问题

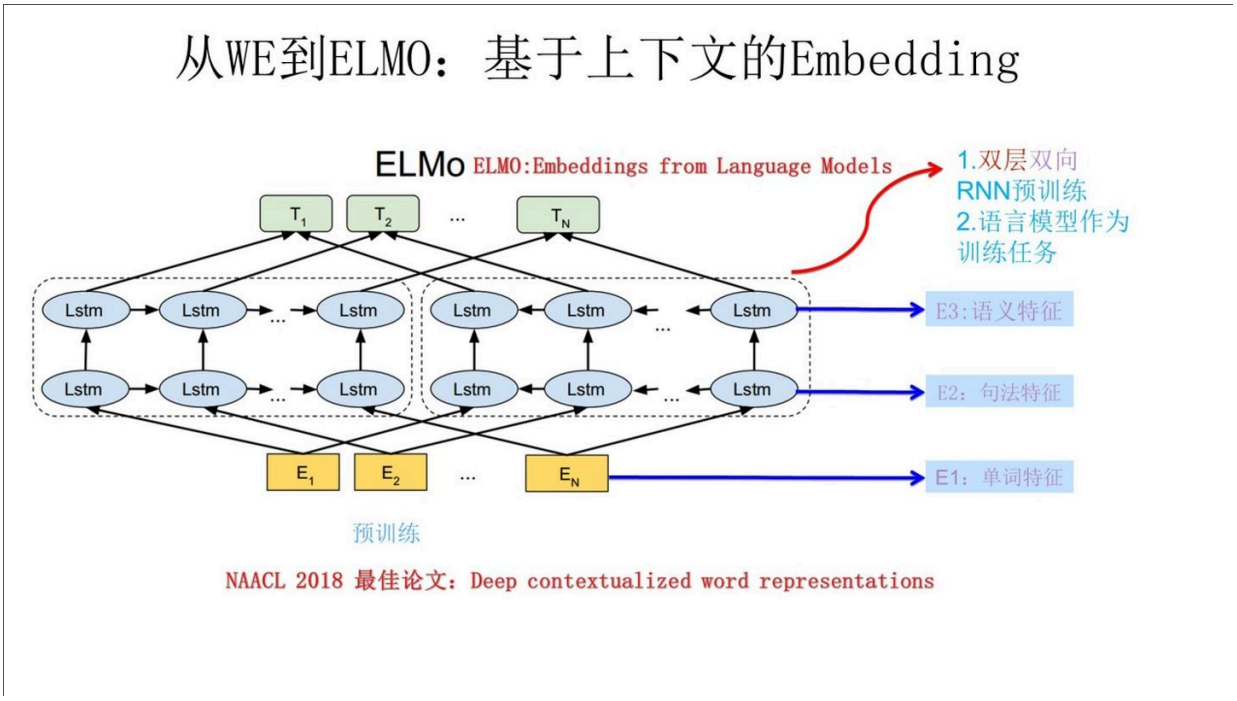
# ELMO（双向LSTM）

前面讲到的词向量模型无法表达多义词的情况，ELMO提供了有效解决这个问题的思路。




ELMO解决多义词问题






**ELMO这个模型是专门做更好的词向量（融入了上下文信息的词向量）的，也就是一个预训练词向量模型**

ELMO不只是训练一个参数矩阵Q，我还可以把这个次的上下文信息融入到这个 Q 矩阵中

**比如E2，左边的 LSTM 获取 E2 的上文信息，右边获取下文信息**

$$x_1, x_2, , x_4, x_5 - - > Word2Vec x_1 + x_2 + x_4 + x_5 - -- > \text{预测那一个词} x_3$$

获取上下文信息后，把三层的信息进行一个叠加

$$E1 + E2 + E3 = K1 \text{一个新的词向量} \approx\approx E1\text{，但是同时包含了} E_2 \text{本身和它的上下文信息}$$

$$E1, E3 \text{相当于两个上下文信息}$$

$$K1 \text{包含了第一个词的词向量，还包含单词特征、句法特征、语义特征}$$

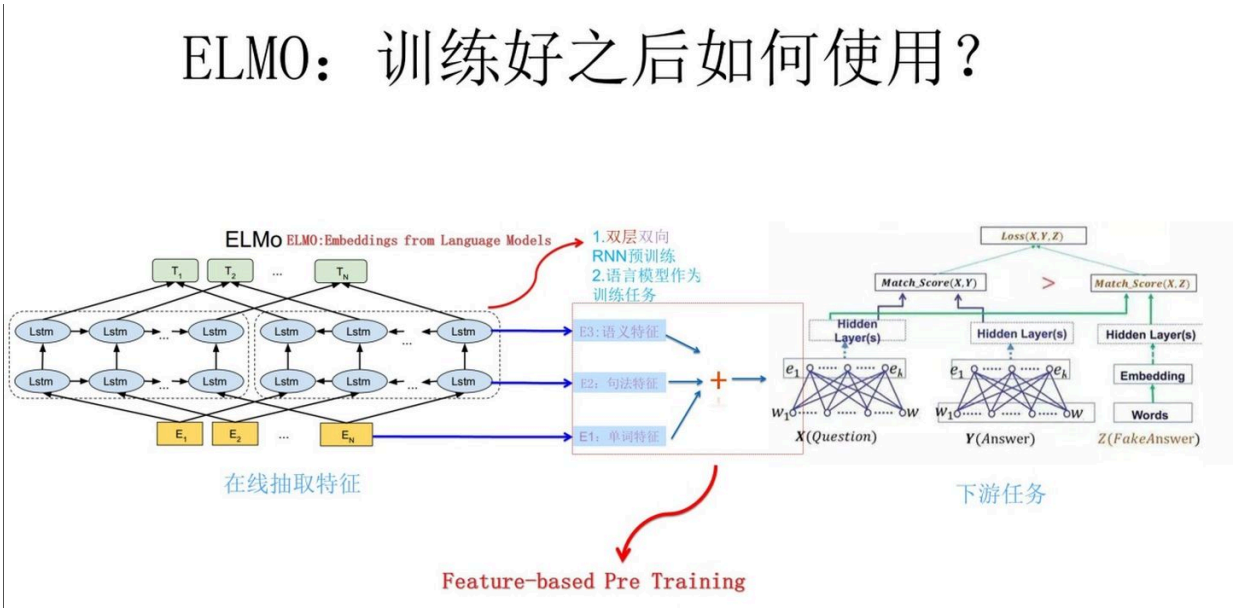
怎么用?

$$E2\text{，}E3 \text{不同，} E1 + E2 + E3 \text{不同}$$

$$apple - -》\text{我吃了一个苹果} - -》[1, 20, 10]$$

$$apple - -》\text{我在用苹果手机} - -》[1, 10, 20]$$

下面的词向量比上面的词向量更客观地描述了苹果这个词，包含了更多的含义，也可以理解为更好地描述了苹果这个词在高维空间的分布




得到更好的词向量后可以完成我们之前所说的那些下游任务，而且实际表现会比之前的word2vec更好

## ELMO的缺陷

因为仍然使用了LSTM，而LSTM 无法并行，还有长期依赖的问题

Attention以及衍生的transformer仍然很好地解决了这个问题