LSTM和RNN学习问题答案



<https://blog.csdn.net/kymowind/article/details/75145775> 不理解！



梯度消失和梯度爆炸、无法长时记忆，一般只能保存几个，经典的RNN的输入和输出要等长。无法长距离依赖：梯度已经几乎减少到0了。那么，从这个时刻开始再往之前走，得到的梯度（几乎为零）就不会对最终的梯度值有任何贡献，这就相当于无论t-3时刻之前的网络状态h是什么，在训练中都不会对权重数组W的更新产生影响，也就是网络事实上已经忽略了t-3时刻之前的状态。这就是原始RNN无法处理长距离依赖的原因。



可以用在LSTM模型的C的训练



LSTM是一种特殊的RNN，主要为了解决长序列RNN中的梯度爆炸和梯度消失问题。

RNN有两个传输状态，LSTM有两个传输状态：cell state和hidden state（RNN的hidden相当于LSTM的cell state）

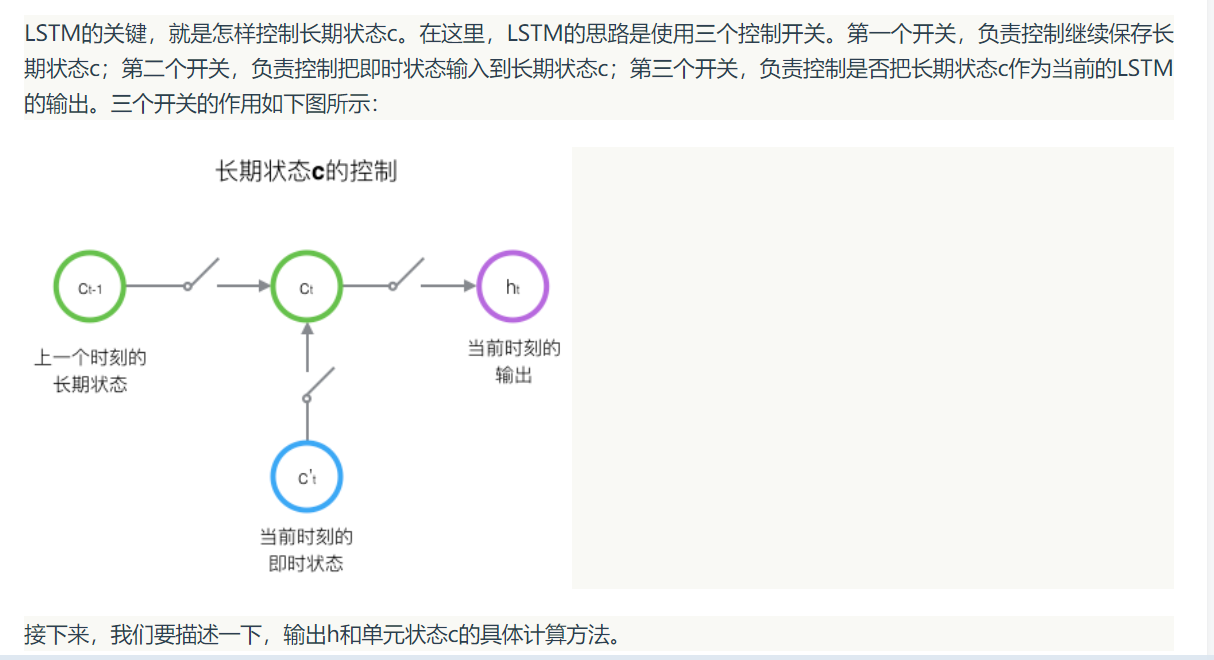


LSTM：忘记门、选择记忆门、输出门。

忘记门：控制上一个阶段的c哪些要留哪些要删除

选择记忆门：控制输入x，判断重要的点

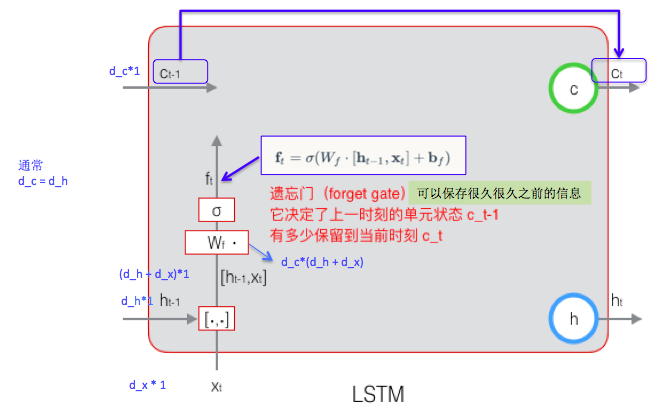
输出阶段：控制哪些会被输出。

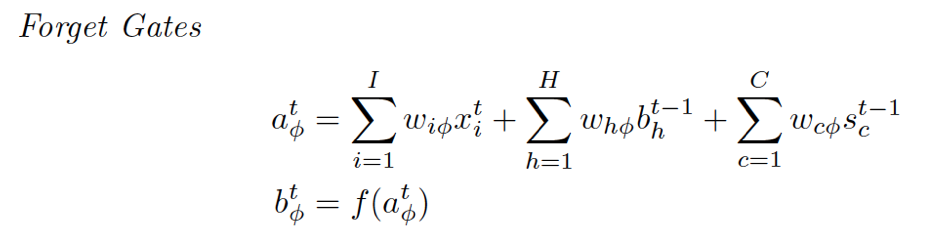


LSTM用两个门来控制单元状态c的内容，一个是**遗忘门（forget gate）**，它决定了上一时刻的单元状态有多少保留到当前时刻；另一个是**输入门（input gate）**，它决定了当前时刻网络的输入有多少保存到单元状态。LSTM用**输出门（output gate）**来控制单元状态有多少输出到LSTM的当前输出值。

三种门如何计算：

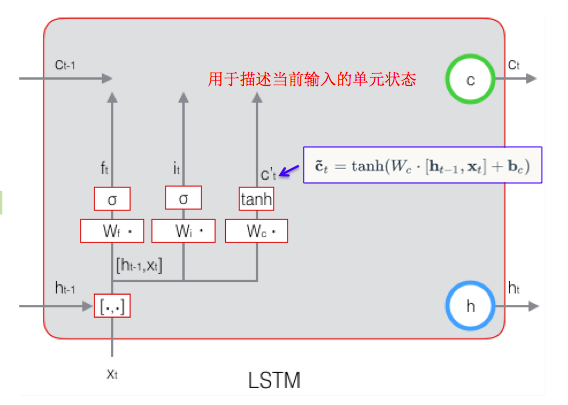
遗忘门：

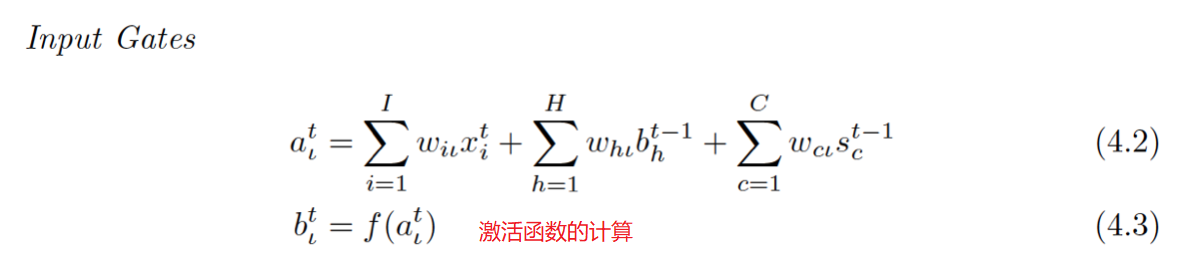




总体来说：使用输入值、激活函数值、隐藏层的值。

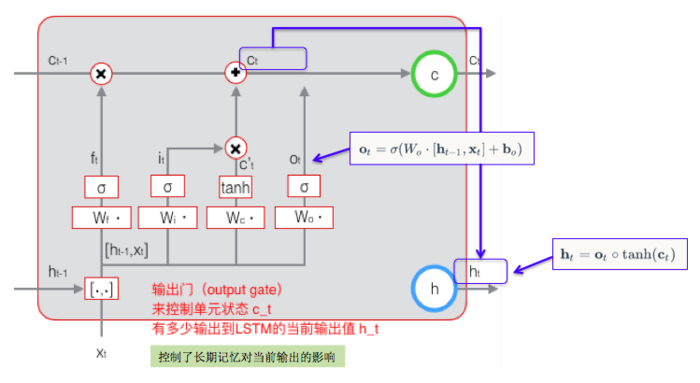
输入门计算：

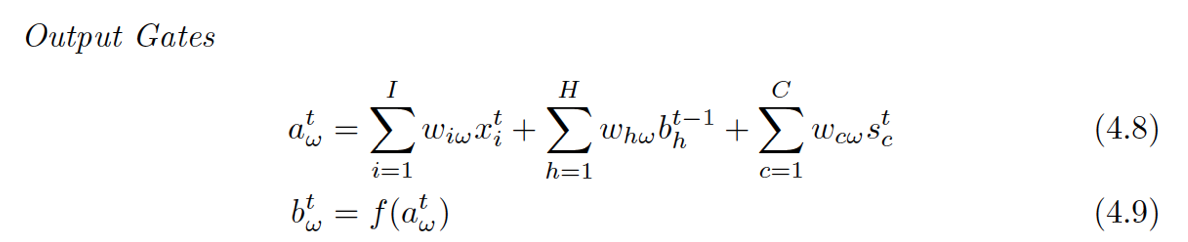




总体来说：上一次输入和本次输入的结果计算当前单元的状态。

输出门计算





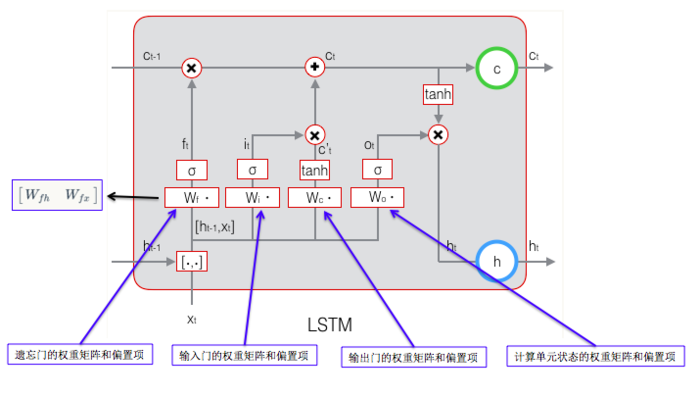
问题：是什么？

Gate的激活函数是：sigmoid函数。

输出的激活函数是：tanh（双曲正切函数），将实数映射到[-1,1]之间。在特征明显的时候想过会更好。

1.sigmod层决定了什么值需要被更新；  
2.tanh层用来生成新的候选信息向量会被更新到细胞状态中。

在过程中要学习8个参数：





使用了sigmoid函数和tanh函数。

激活函数的主要原因是：提高网络的非线性建模能力。如果网络中只有线性卷积和全连接操作，那么网络只能处理线性映射，那么网络的泛化能力就非常弱。

Sigmoid函数的优点：

* + 将数值集中在[-1，1]之间。
  + 将重点区推向中央区，非重点区推向两侧区。
  + 导数性质很好。

Sigmoid函数的缺点：

* + 激活函数的计算量非常大
  + 反向传播求误差梯度时候涉及到除法
  + 反向传播时容易出现梯度爆炸和梯度消失的问题，本质是因为链式求导带来的，可以用RelU函数代替sigmoid函数。

Tanh函数优点：

* + 使输入和输出保持非线性单调上升和下降的关系。
  + 渐进与0，延缓了神经网络的饱和期。
  + 使用效果更好。

总结来说：tanh是sigmoid函数的缩减版。



主要变化是在三个门中增加了细胞状态信息，成为peephole connection。

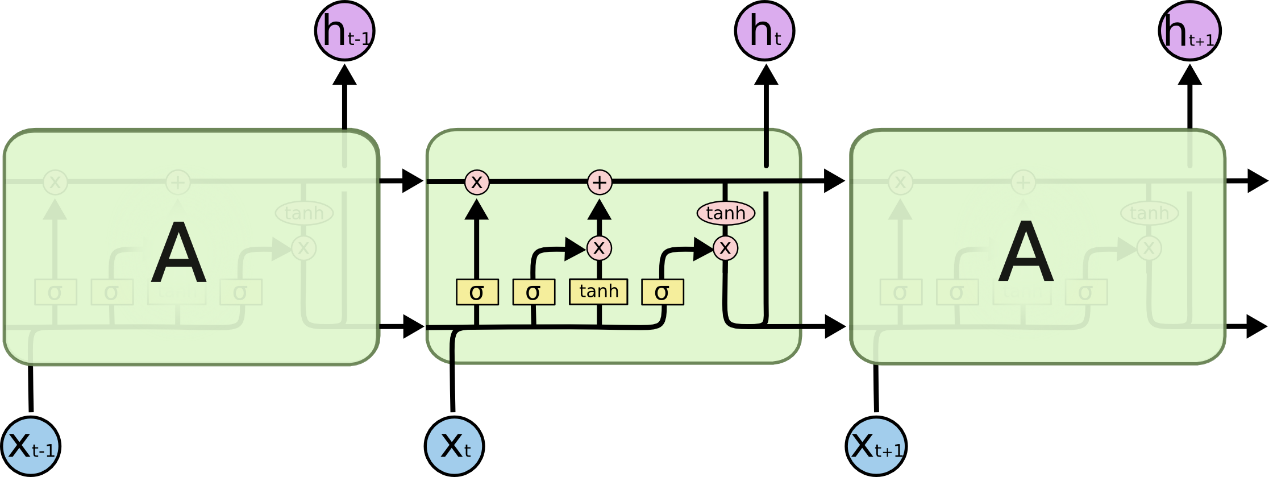
变种一：GRU-LSTM

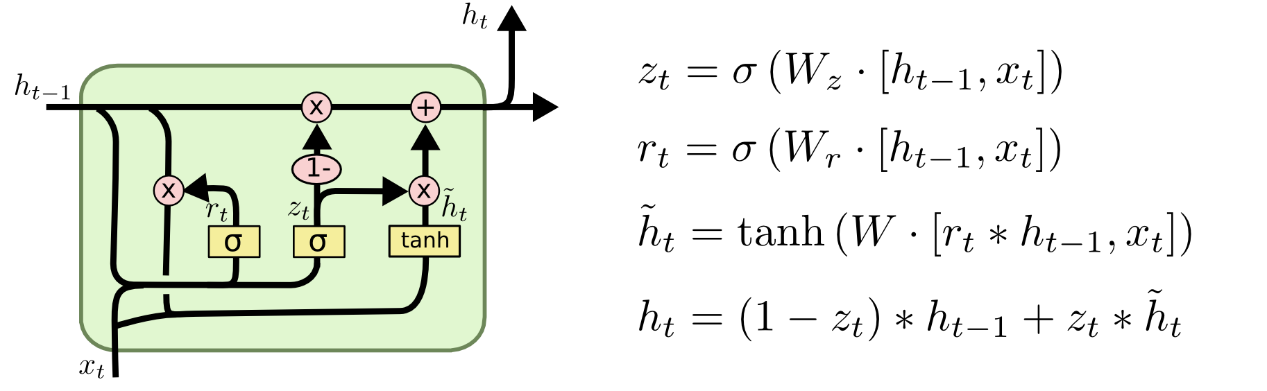
是改动较大也是广为使用的一个版本，将输入门和遗忘门合成一个更新门，细胞状态信息流和隐藏层状态信息流合并为一个信息流。

变种的优点：

* + 解决了梯度爆炸和梯度消失的问题
  + 少了几个矩阵，节约了训练时间
  + 构造更简单
  + 结果更好，一般能提高2.点左右。

传统的LSTM和GRU-LSTM比较：





ingenious传统的LSTM比较：

只有更新门和重置门。更新门负责控制前一个时刻的状态信息被带入到当前时刻的程度，数值越大，越多信息被利用。重置门负责控制前一个时刻的状态信息的忽略程度，数值越小，忽略越多。



Encoder的输入：输入序列 输出：输入序列对应的特征向量，所有特征向量的长度是相等的。

Decoder的输入：特征向量 输出：序列，根据当前语句预测的下一句结果。



损失函数：交叉熵函数

其他损失函数：最大似然条件概率

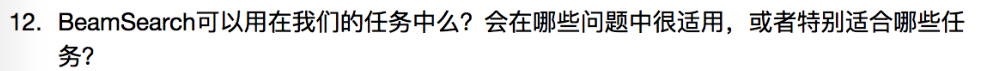




编码常用方法：最后一个时刻的隐藏层状态作为整个序列的特征向量c

解码常用方法：

* + 简单解码模式：把编码端得到的编码向量做为解码模型的每时刻输入特征。
  + 带输出回馈的解码模式：只需要上一个时刻的输出。
  + 带编码向量的解码模式：需要上一刻的输出和解码向量。
  + 带注意力的解码模式：需要某些重点的解码向量。



BeamSearch算法

1. 简介

Beam Search（集束搜索）是一种启发式图搜索算法，在解空间很大的情况下，减掉一些质量较差的节点。

1. 优点

减少空间消耗、提高时间效率。

1. 缺点

是贪心算法，只能找到局部最优解，同时可能最优方案被减掉。

1. 流程

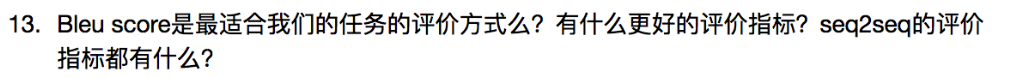
Beam Search（集束搜索）使用广度优先策略建立搜索树，在树的每一层，按照启发代价对节点进行排序，然后仅留下预先确定的个数（Beam Width-集束宽度）的节点，仅这些节点在下一层次继续扩展，其他节点就被剪掉了。将初始节点插入到list中，将给节点出堆，如果该节点是目标节点，则算法结束；否则扩展该节点，取集束宽度的节点入堆。然后到第二步继续循环。算法结束的条件是找到最优解或者堆为空。集束宽度可以是预先定好的，也可以是变动的，可以先按照一个最小的集束宽度进行搜索，如果没有找到合适的解，再扩大集束宽度再找一遍。

1. 应用Beam Search（集束搜索）多用在一些大型系统中，比如机器翻译系统，语音识别系统等，因为这些系统中的数据集可能非常大，而且结果也没有唯一正确的解，系统用最快的方式找到最接近正确的解才是系统的目标。
2. 和贪心算法的区别是：找前B个/找最优的一个？

4.Seq2Seq模型中的使用：sequence2sequence模型[[深度学习：Seq2seq模型](http://blog.csdn.net/pipisorry/article/details/78258198)]中，beam search的方法只用在测试的情况（decoder解码的时候），因为在训练过程中，每一个decoder的输出是有正确答案的，也就不需要beam search去加大输出的准确率。

BeamSearch是否可以应用在我们的任务中：可以的。

* + 可以在训练神经网络的时候使用。找到可能性最大的几个主题，在主题范围内使用歌词，减小神经网络的复杂度。
  + 可以在预测，预测的时候找到可能性最大的几个词，减小神经网络的训练范围。
  + 可以在判断预测结果的使用，不同预测结果的最后得分可以选择可能性最大的结果。



Bleu score是机器翻译的评价指标。**BLEU(bilingual evaluation understudy)** 中文名称为双语互译质量辅助工具, 计算这个指标, 需要使用机器翻译好的文本(称作candidate docs)以及一些专业翻译人员翻译的文本(称作reference docs)。 本质上讲BLEU 就是用来衡量机器翻译文本与参考文本之间的相似程度的指标,取值范围在0-1, 取值越靠近1表示机器翻译结果越好。

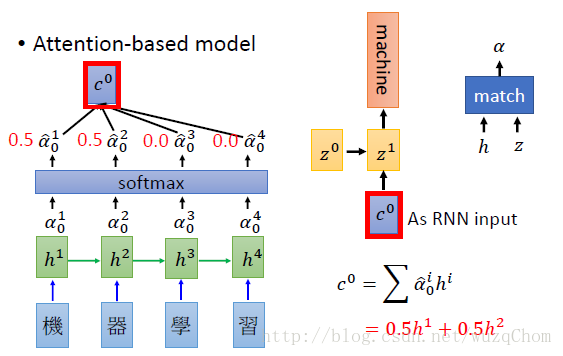
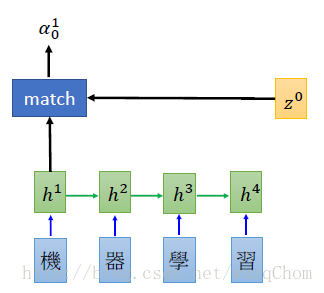
总结：BLEU是一个非常简单快速粗略的评估指标, 当面对多个翻译模型且需要快速选择模型的场景, 可以使用这个指标来评估模型的好坏, 但是在需要精确评估翻译文本质量的场景, 这个指标就不是那么适用了。

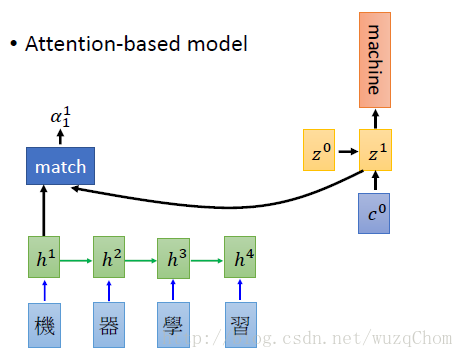
更好的评价指标：



Attention机制是保留LSTM对输入序列的中间输出结果，通过一个模型对这些结果进行学习并在输出时将学习结果与输出序列关联。本质是相似性度量。当前输入和当前目标输出的相似度越大，当前目标的输出越依赖当前输入。

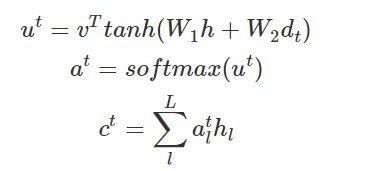
一般情况下，如果没有attention机制，会把最后一个state当做Eecoder的结果传入Decoder。



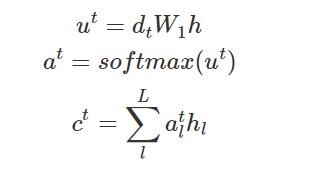


Python实现的两种Attention方式

* BahdanauAttention



* LuongAttention



Attention方式

* Soft Attention

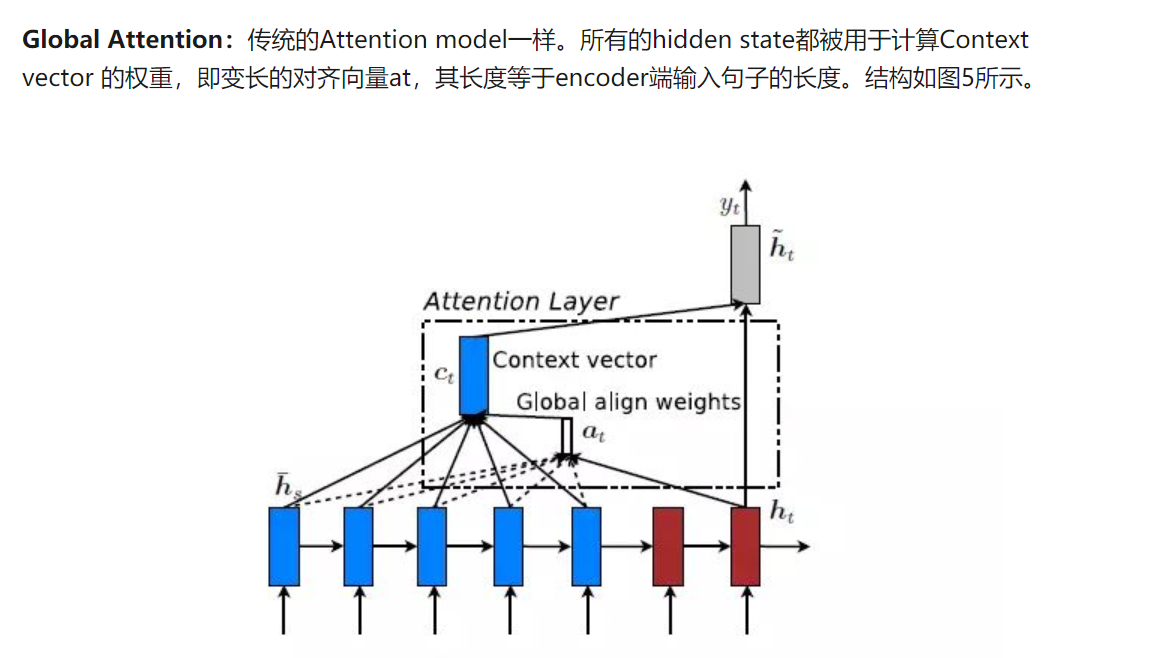
Soft Attention是参数化的（Parameterization），因此可导，可以被嵌入到模型中去，直接训练。梯度可以经过Attention Mechanism模块，反向传播到模型其他部分。

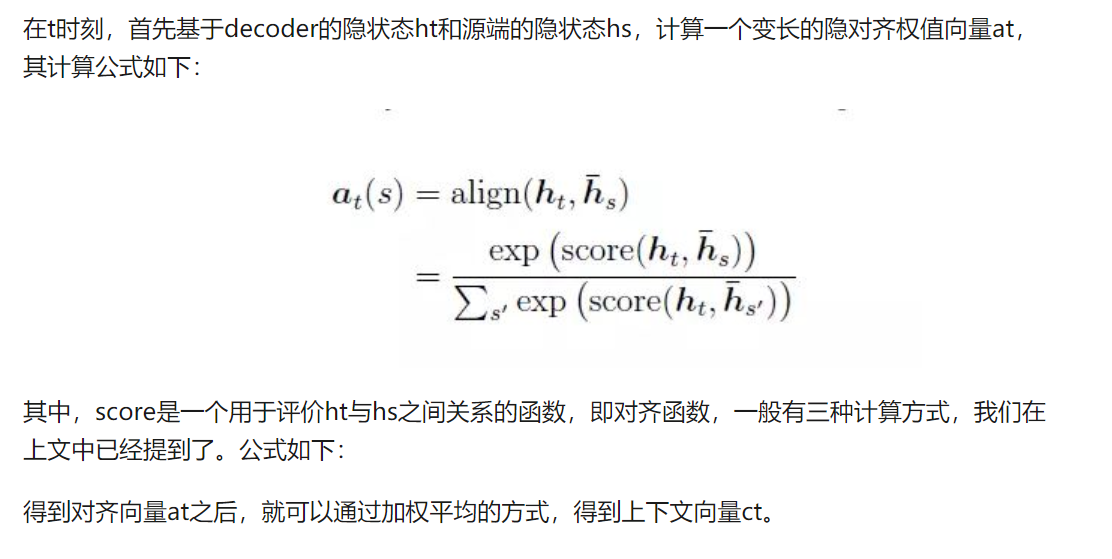
Hard Attention

* Hard Attention是一个随机的过程。Hard Attention不会选择整个encoder的输出做为其输入，Hard Attention会依概率Si来采样输入端的隐状态一部分来进行计算，而不是整个encoder的隐状态。为了实现梯度的反向传播，需要采用蒙特卡洛采样的方法来估计模块的梯度。

两种Attention Mechanism都有各自的优势，但目前更多的研究和应用还是更倾向于使用Soft Attention，因为其可以直接求导，进行梯度反向传播。

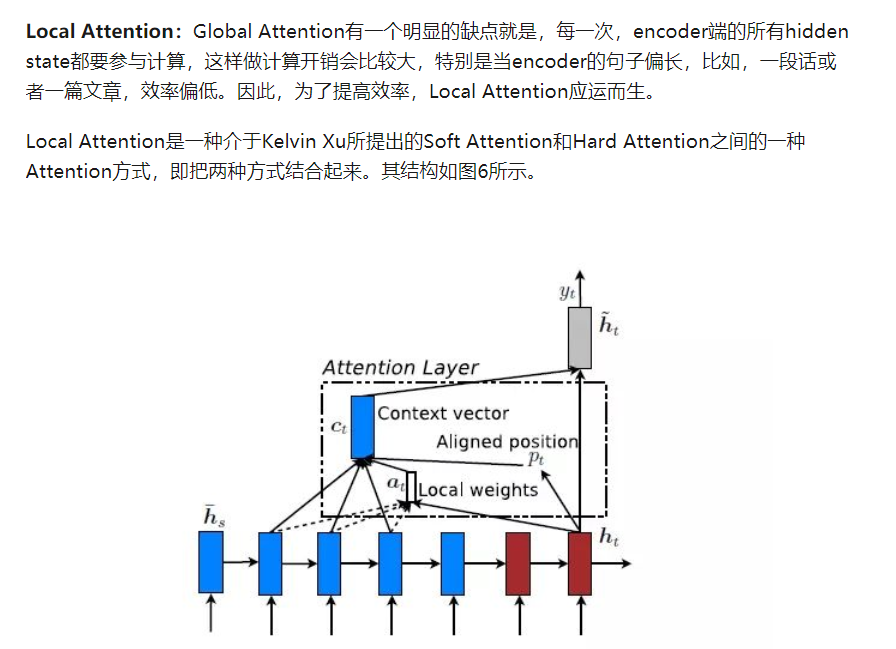
* Global Attention

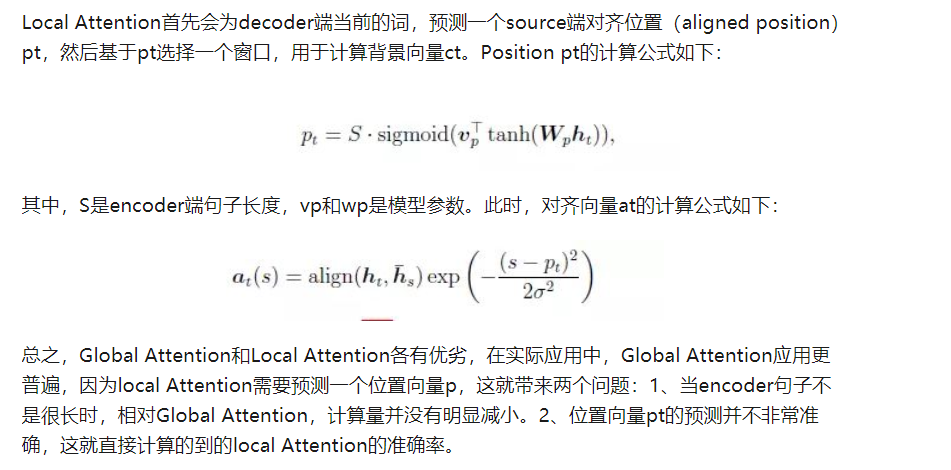




这种方法的重点是：计算变长的隐对齐权值at。

* Local Attention





就是在Global Attention上加了pt（source postiton 对应到target position）

SlefAttention

Source Attention + Souce Attention



使歌词的每一句通顺，避免每句出现相同词语大量重复的情况。

+

