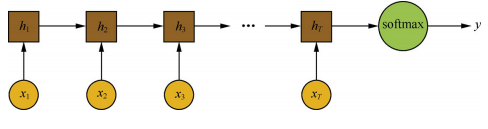
# 循环神经网络

**问题1 简述RNN和前馈神经网络(例如CNN)在处理文本数据时的过程和特点**

首先，在传统的方法中，一般用基于频率的TF-IDF来提取文本的特征向量，但是这显然丢失了位置信息。

CNN在处理文本时通常将文本处理成等长，然后用卷积核池化的方式来提取特征向量，这样可以捕捉到文本中的局部信息，但是单词之间的长距离依赖还是很难学习到。

循环神经网络之所以能很好的处理序列信息，因为在建模的过程中，文本数据是在不同的时间片依次输入到模型，从而使得历史信息通过隐层编码到当前的状态中，这样RNN就拥有了记忆功能（同样带来训练耗时的问题）。

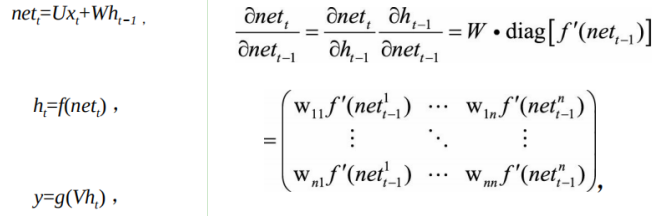


PS:激活函数一般使用tanh或者relu

**问题2 简述梯度消失和梯度爆炸，以及缓解办法。**

在RNN领域，一般采用BPTT算法(基于时间的反向传播)，本质上是将T个时间步的模型等价的看做T层前馈神经网络。

循环神经网络的设计初衷之一就是能够捕获长距离输入之间的依赖。从结构上来看，循环神经网络也理应能够做到这一点，但是实践表明采用BPTT算法的RNN并不能成功的捕获长距离依赖问题，而这主要是由神经网络的梯度消失造成。分析如下：



其中的diag表示对角矩阵，由于f(net)是向量，与矩阵W相乘的过程中，可以等价的处理为对角矩阵，右图的计算结果称为**雅克比矩阵**，n为隐层单元数目。在反向传播的时候，**如果雅克比矩阵的特征值大于1**，则梯度值成指数增加导致梯度爆炸，反之导致梯度消失。

对于普通的前馈网络来说，梯度消失意味着无法通过加深网络层次来改善神经网络的预测效果，因为无论如何加深网络，只有靠近输出的若干层才真正起到学习的作用。这使得循环神经网络模型很难学习到输入序列中的长距离依赖关系

梯度爆炸可以用梯度修剪缓解，梯度消失可以行三个方面考虑：初始化靠近单位矩阵、改进模型（LSTM、GRU）、残差网络

**问题3 在一般的前馈神经网络中，RELU可以有效的改善梯度消失问题，在RNN中可以使用RELU作为激活函数吗？**

1. 首先RELU之所以可以缓解梯度消失问题在于函数在x>0是没有饱和区，所以多层一般不会出现严重的数值问题，但是在RNN中，由于权重参数的连乘，所以多层会存在数值问题。分析如下：

**从前向来考虑**：如果W不是单位矩阵，根据前传公式，最终的结果将会趋于0或者无穷，引发严重的数值问题。卷积神经网络之所以不会出现这样的现象，是因为在卷积神经网络中每一层的权重矩阵W是不同的，并且在初始化时它们是独立同分布的，因此可以相互抵消。

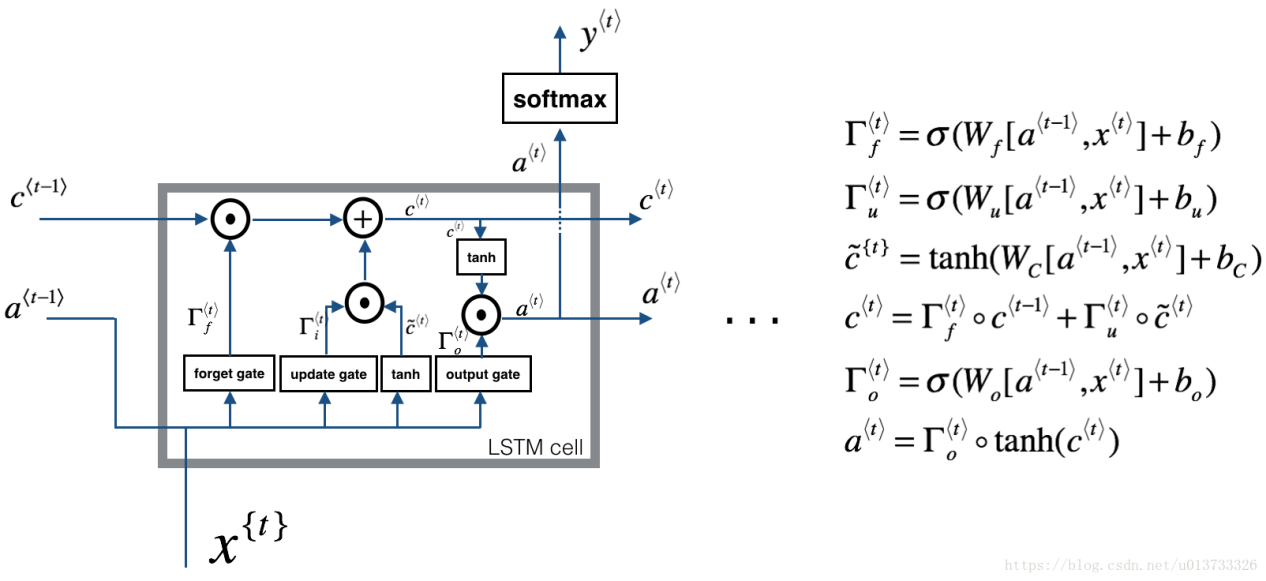
**从反向来考虑**：假如输入在线性区，那么激活函数相当于1，梯度的系数可以达到W的T次方。即使采用了ReLU激活函数，只要W不是单位矩阵，梯度还

是会出现消失或者爆炸的现象。

1. RNN梯度消失不仅受激活函数影响，还在于RNN参数的连乘。
2. **综上所述**，只有将W初始化为单位矩阵时，RELU才能取得不错的效果。实验证明，初始化W为单位矩阵并使用ReLU激活函数在一些应用中取得了与长短期记忆模型相似的结果，并且学习速度比长短期记忆模型更快，是一个值得尝试的小技巧

**问题4 LSTM如何实现长短期记忆功能**

首先，LSTM的结构如下图所示：



其实LSTM的思路很清晰：2条记忆线、一个输入、一个输出。

**历史信号**：进来后，与遗忘门相乘决定多大程度采用历史信息。

**记忆单元**：用双曲激活当前输入后，和更新门相乘再和记忆相加→Ct。

**输出信号**：将选择的历史信息和当前记忆单元求和，然后和输出门相乘决定最后 的输出信号。

所有**门的数值**由历史信息和当前输入决定。

1. 普通的RNN之所以不能捕获长期记忆，是因为W连乘问题导致梯度消失。

2、LSTM和普通RNN的相比之所以能缓解梯度问题，本质上讲在于信号在不同的时间步之间传播时，其数值不在取决于一个单一的矩阵，而是将数值问题的压力分散到更多的矩阵，抽象一层来讲就是由于门控机制的引入，使得理论上而言当遗忘门数值接近1的时候可以顺利的将历史信息引入并传递下去。

PS：关于激活函数的选取：门控值采用sigmoid函数，候选记忆采用双曲正切。原因解释：为了符合门控的物理定义有必要采取“开关函数”，而选用tanh在于大多场景下特征分布以0为中心，而tanh在0附近有较大的梯度，有利于参数的快速收敛

LSTM的演化：

经典的LSTM在计算各门控时，通常使用输入x t 和隐层输出h t−1 参与门控计算，例如对于输入门的更新：。其最常见的变种是加入了窥孔机制，让记忆c t−1 也参与到了门控的计算中 。总而言之，LSTM经历了20年的发展，其核心思想一脉相承，但各个组件都发生了很多演化。



**问题5　什么是Seq2Seq模型？Seq2Seq模型有哪些优点？**

Seq2Seq模型的核心思想是，通过深度神经网络将一个作为输入的序列映射为一个作为输出的序列，这一过程由编码输入与解码输出两个环节构成。

在经典的实现中，编码器和解码器各由一个循环神经网络构成，既可以选择传统循环神经网络结构，也可以使用长短期记忆模型、门控循环单元等。在Seq2Seq模型中，两个循环神经网络是共同训练的。

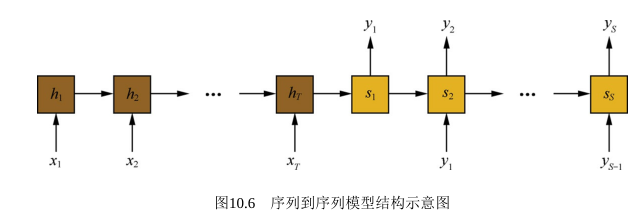
**问题6　Seq2Seq模型在解码时，有哪些常用的办法？**

贪心法：选取一种度量标准后，每次都在当前状态下选择最佳的一个结果，直到结束。贪心法的计算代价低，适合作为基准结果与其他方法相比较。很显然，贪心法获得的是一个局部最优解，由于实际问题的复杂性，该方法往往并不能取得最好的效果。

集束搜索(Beam Search)：一种启发式算法。该方法会保存beamsize（后面简写为b）个当前的较佳选择，然后解码时每一步根据保存的选择进行下一步扩展和排序，接着选择前b个进行保存，循环迭代，直到结束时选择最佳的一个作为解码的结果。

别的优化方法还有：解码时使用堆叠的RNN、增加Dropout机制、与编码器之间建立残差连接、注意力机制还可以采用记忆网络等，从外界获取知识。

**问题7　Seq2Seq模型引入注意力机制是为了解决什么问题？为什么选用了双向的循环神经网络模型？**

****

在Seq2Seq模型中（见图10.6），当前隐状态以及上一个输出词决定了当前输出词，即:





在实际使用中，会发现随着输入序列的增长，模型的性能发生了显著下降。这是因为编码时输入序列的全部信息压缩到了一个向量表示中。随着序列增长，句子越前面的词的信息丢失就越严重。

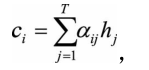
而在解码时，目标语言的第一个词大概率是和源语言的第一个词相对应的，假如翻译一个有100个词的句子，需要将整个句子全部词的语义信息编码在一个向量中。而在解码时，目标语言的第一个词大概率是和源语言的第一个词相对应的，这就意味着第一步的解码就需要考虑100步之前的信息。

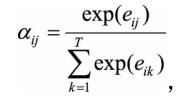
建模时的一个小技巧是将源语言句子逆序输入，或者重复输入两遍来训练模型，以得到一定的性能提升。使用长短期记忆模型能够在一定程度上缓解这个问题，但在实践中对于过长的序列仍然难以有很好的表现。

**Seq2Seq模型中引入注意力机制就是为了解决上述的问题**。在注意力机制中，仍然可以用普通的循环神经网络对输入序列进行编码，得到隐状态h 1 ,h 2 …h T 。但是在解码时，每一个输出词都依赖于前一个隐状态以及输入序列所有的隐状态。



其中语境向量c i 是输入序列全部隐状态h 1 ,h 2 …h T 的一个加权和：







(j为输入编号，i为输出编号) 其中注意力权重参数αij 并不是一个固定权重，而是由另一个神经网络计算得到神经网络a将上一个输出序列隐状态s i−1 和输入序列隐状态h j 作为输入，计算出一个x j ，y i 对齐的值e ij ，再归一化得到权重αij 。

我们可以对此给出一个直观的理解：在生成一个输出词时，会考虑每一个输入词和当前输出词的对齐关系，对齐越好的词，会有越大的权重，对生成当前输出词的影响也就越大。

