重点文章

<http://nlp.seas.harvard.edu/2018/04/03/attention.html>

知乎大佬

<https://zhuanlan.zhihu.com/p/54743941>

参考链接  
[https://zhuanlan.zhihu.com/p/54743941](https://zhuanlan.zhihu.com/p/54743941" \t "https://www.cnblogs.com/chinasoft/p/_blank)  
[http://www.ai-start.com/dl2017/html/lesson5-week1.html#header-n194](http://www.ai-start.com/dl2017/html/lesson5-week1.html" \l "header-n194" \t "https://www.cnblogs.com/chinasoft/p/_blank)  
[https://zhuanlan.zhihu.com/p/46327831](https://zhuanlan.zhihu.com/p/46327831" \t "https://www.cnblogs.com/chinasoft/p/_blank)  
[https://zhuanlan.zhihu.com/p/55386469](https://zhuanlan.zhihu.com/p/55386469" \t "https://www.cnblogs.com/chinasoft/p/_blank)  
[https://kexue.fm/archives/5409](https://kexue.fm/archives/5409" \t "https://www.cnblogs.com/chinasoft/p/_blank)  
[https://zhuanlan.zhihu.com/p/54356280](https://zhuanlan.zhihu.com/p/54356280" \t "https://www.cnblogs.com/chinasoft/p/_blank)  
[http://jalammar.github.io/illustrated-transformer/](http://jalammar.github.io/illustrated-transformer/" \t "https://www.cnblogs.com/chinasoft/p/_blank)  
[https://kexue.fm/archives/4765](https://kexue.fm/archives/4765" \t "https://www.cnblogs.com/chinasoft/p/_blank)

5-1 本章内容介绍

序列模型：循环神经网络



5-2 什么是序列模型

序列数据sequential data

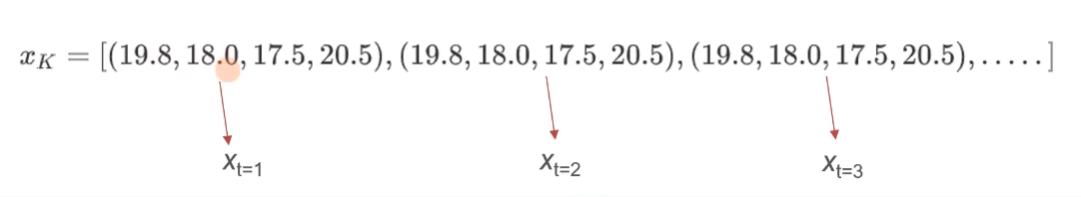
以下都是序列数据

时间步，xt1，xt2，xt3分别对应不同的单词，也可以按照单词去划分时间步，I 空格 l

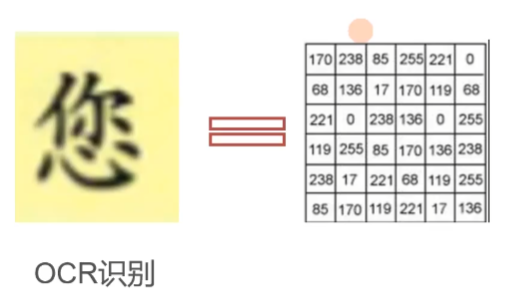


k线图数据的拆分，下面不能再拆分了，因为四个数据是同时出现的。上图每个单词是先后

顺序的，所以可以继续拆分。

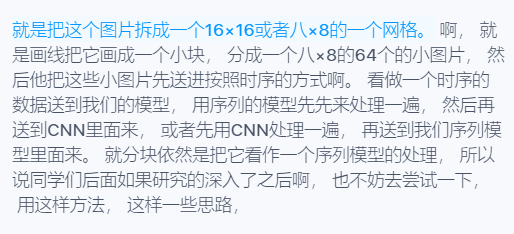


图片的数据也可以当成序列数据，右边一行代表一个实时间步。

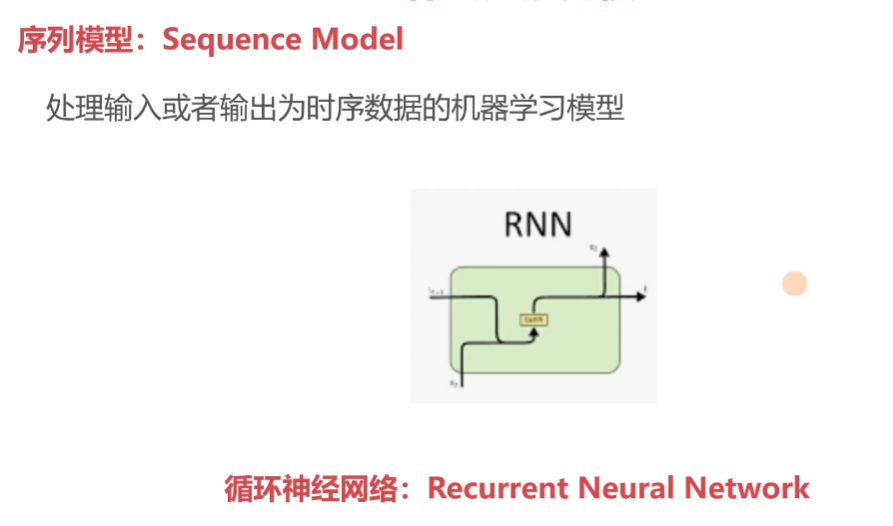


Cnn模型关注结构特征，到底识别图片用哪种吗，取决你关注数据的结构特征还是时序特征。

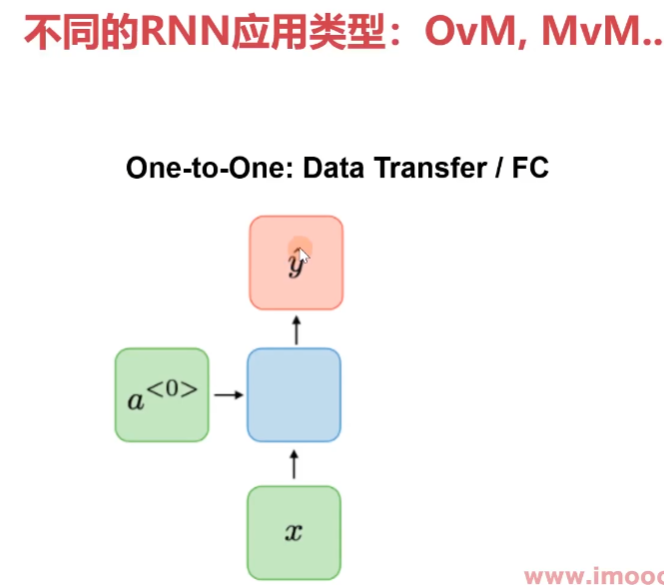
所以图片比较好用cnn模型。

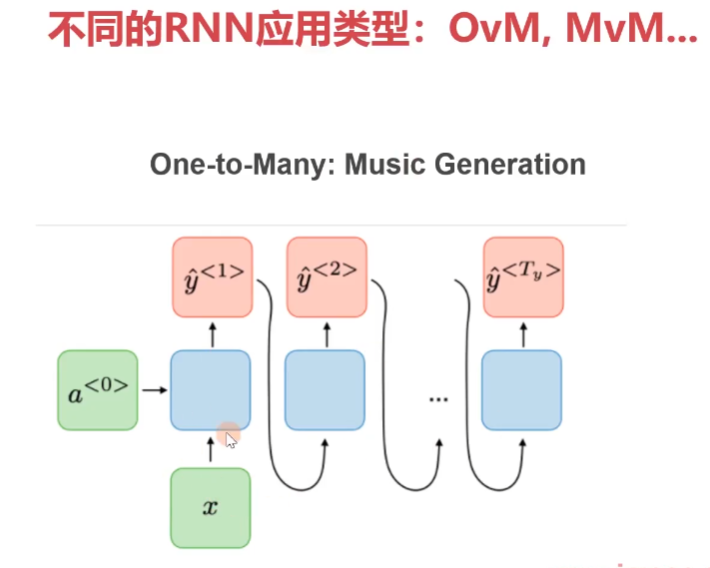


RNN

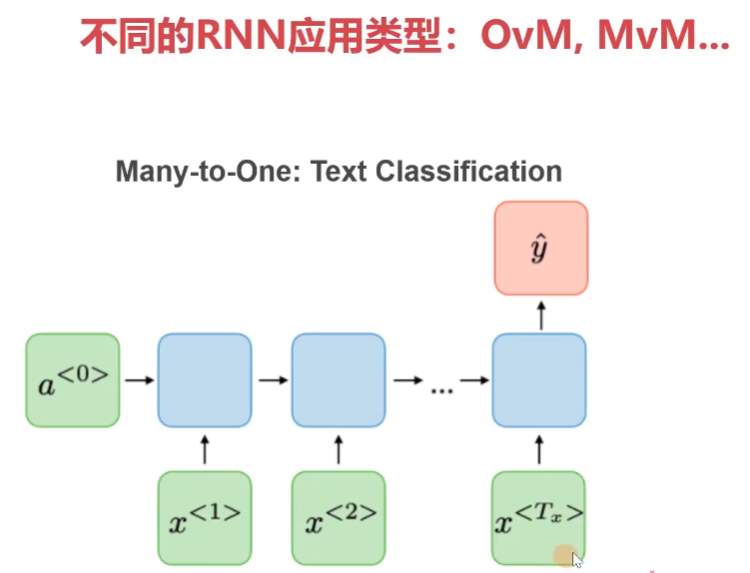


5-3 不同的RNN应用类型：OvM, MvM

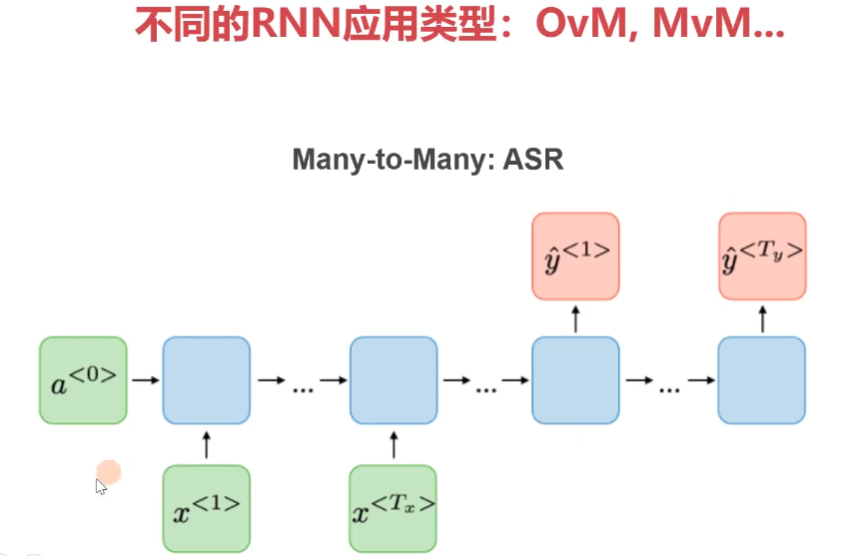




文本分类

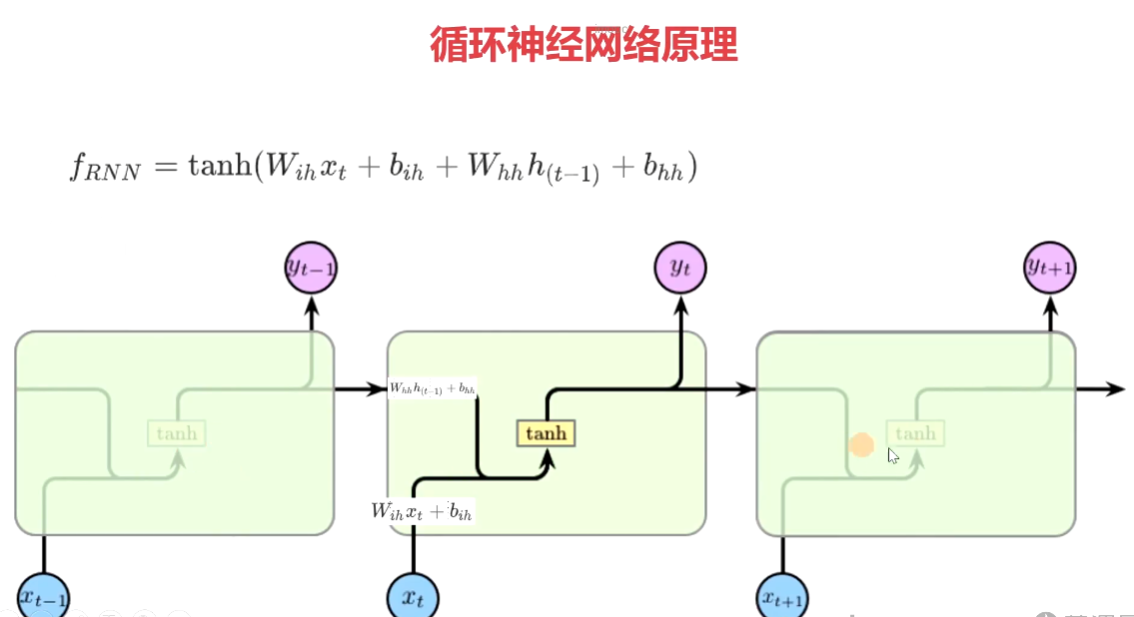


多对多，语音识别

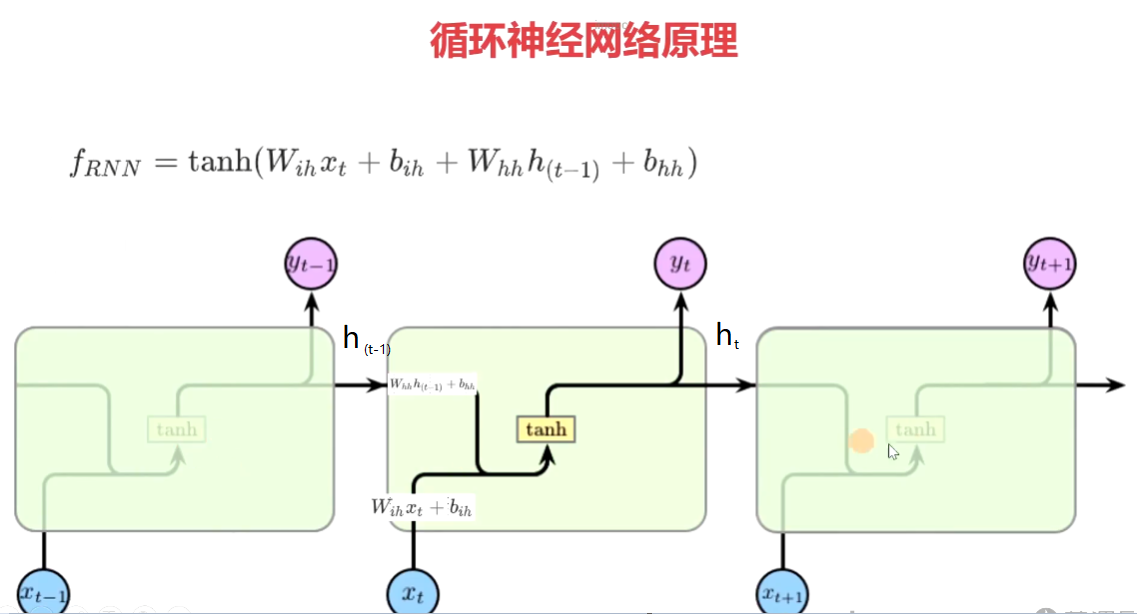


5-4 循环神经网络原理

循环神经网络是一个数据点一个数据点进行处理的，但是它每个时间点的处理并不是独立的，就是只管我当前处理的数据点，不是的。我们从下图可以看到中间时间点的输入依赖于上一个时间点的输出。这个传递的信号相当于循环神经网络的记忆功能，他把当前时间点的一个信号，记忆或者时序关系，时序特征给记忆到了，然后传递到我们当前步。



每个时间步都会有两个输出，y（记录当前步的特征）和h（hidden state，负责记录前面所有步的特征），xt就是我们当前步的特征，可能后面会有一些变体，他的本质思路没有变，就是我要记录的时序特征，也要记录当前的时序特征，这个就是循环神经网络的原理。公式是两个线性相加。我们第一步的话h输入可以使0，也可以是随机数



*hidden\_size就是h的size*

*可以把input\_size看为原始特征个数，hidden\_size就是特征提取之后的特征个数（CNN中就是feature\_maps的数量）*

*与CNN或者一般FNN不同的是，CNN每层的feature\_maps的数量可以改变，比如我最开始输入的特征数量是28\*28=784*

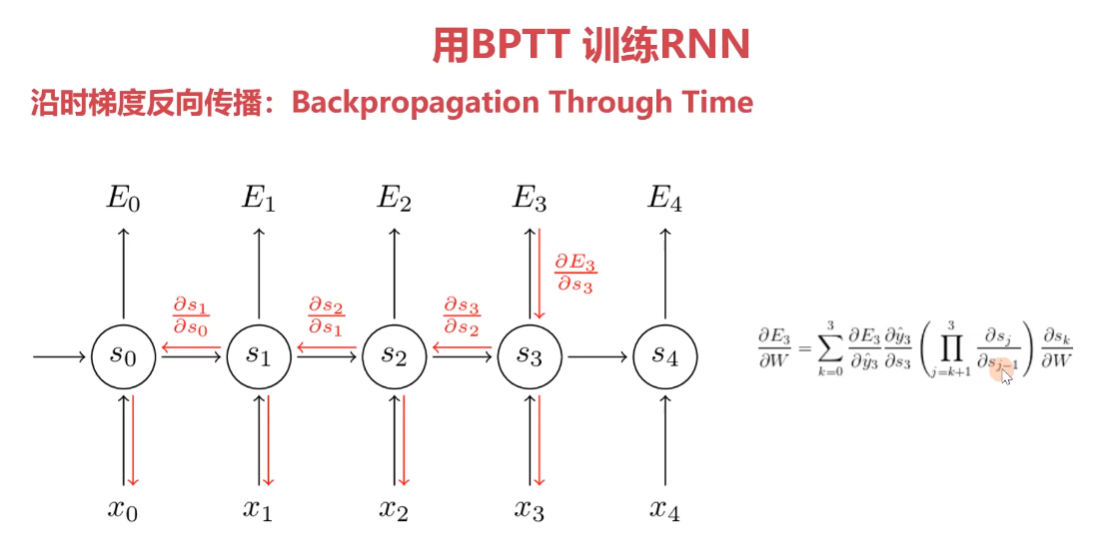
*通过设置第一层的feature\_maps的数量为100，我把特征个数改变到了100；再通过设置第二层的feature\_maps的数量为50，我又可以把特征个数变成50...*



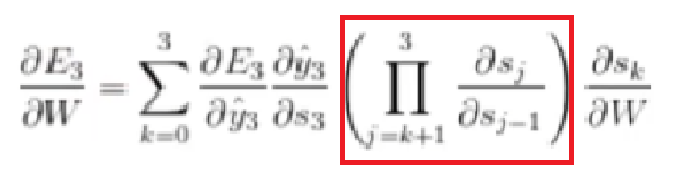
5-5 用BPTT 训练RNN

不管是cnn，还是rnn，他的训练都是采用求导数，然后求出梯度的办法去调整权重，也就是叫梯度反向传播。

下图每个节点都有两个权重，一个是求出E的权重，从上到下直接求导立马可以求出导数。一个是求出记忆的权重，从右到左一步步求导，有点像全链接层从最后一个节点往前面求导。



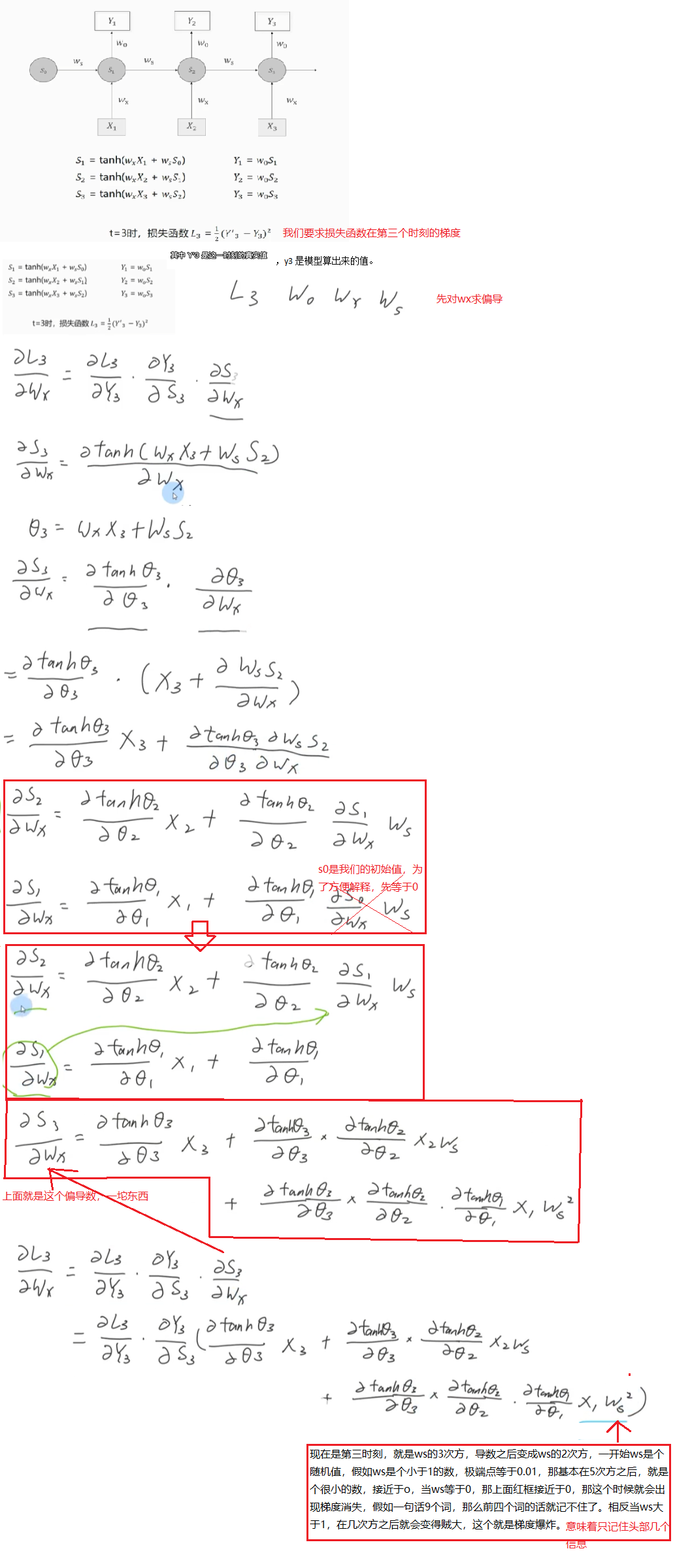
红圈代表递归，代表向左一步步求导数，过长的时间步会导致梯度消失，就是假如我们一篇文章几千个字一下子塞到模型里面，就是记不住一句话太靠前的信息。还有一种情况就是不当的初始值会导致梯度的爆炸，就是我们的hidden state选择得不对的时候。



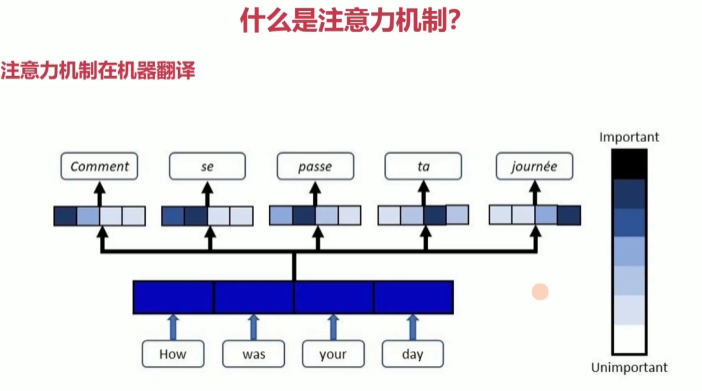
L3损失函数要对ws，w0，wx求偏导数，这三个都是因变量，因为我们调的就是w这些权重，对某一个因变量求偏导数，那么其他变量就当成是常数。意思就是我们有多少个因变量（权重），我们就需要对多少个因变量求偏导数，也就是算出在对应偏导数下该点的斜率，有多少个权重，我们就求出多少个斜率，斜率\*学习率=梯度。

我们先对wx求偏导数，依据链式法则：

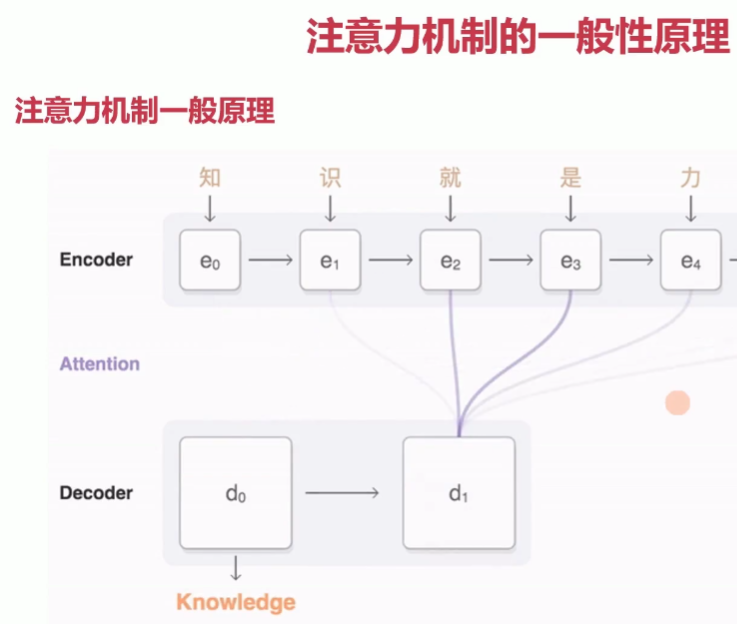
求复合函数的导数就是外面函数的导数\*里面函数的导数



7-2 什么是注意力机制？



7-3 注意力机制的一般性原理



线越深代表越关注

句子和句子之间不是意义映射

数学原理

