

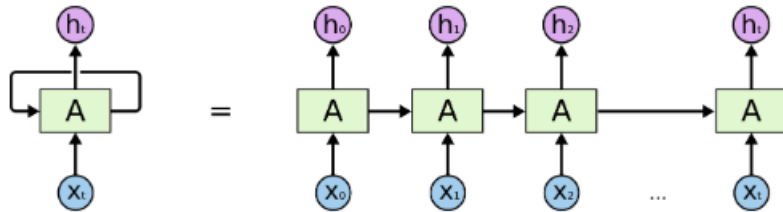
RNN And LSTM

simple-RNN

definition:只考虑RNN的初始架构

main idea:

1. 存在三层 输入 输出 隐藏层
2. 架构



其中 $h(t) = f(W1x(t) + W2h(t-1))$ f 为激活函数

使用 h 来保存之前的信息 由输入得到 h 后

$$y(t) = f(W3h(t))$$

RNN的缺点

main idea:

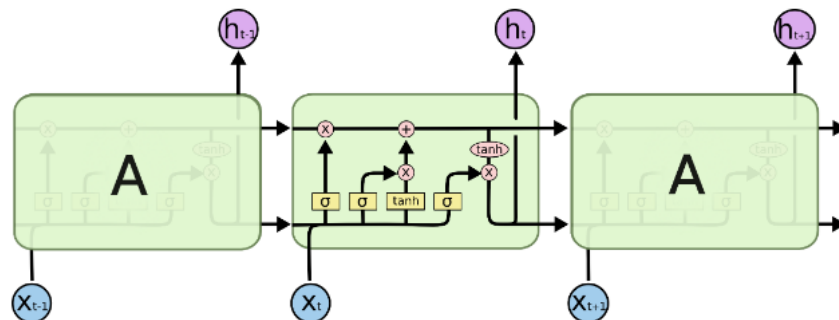
1. 短时间的记忆可以实现 比如 i want to fly to the xxx 这里很容易推测出是sky
2. 长期的记忆 i can speak xxx 这里就不容易简单通过上下文进行推断 这里是什么

LSTM的引入

架构

main idea:

1. 长期记忆是这个架构最基本的特征
2. 基础模型



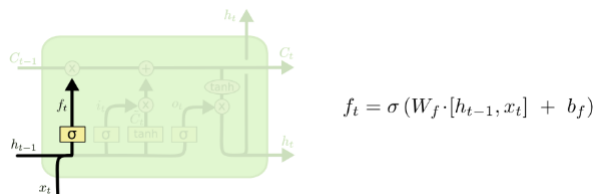
存在三门 遗忘门 输出门 以及细胞状态 内部的操作都是对细胞状态做出改变 输出依赖于细胞状态

数据维度存在三层 与往常的batch feature不一 还外加了一个时间步的维度 即记忆的传递
三层 最外一层是batch 表示参数并行计算 中间一层是时间步 表示训练时的单位 最后一层是feature层

如果时间步为1则和传统的模型类似了 就没有前后的记忆 并且时间步的大小还决定了记忆的距离

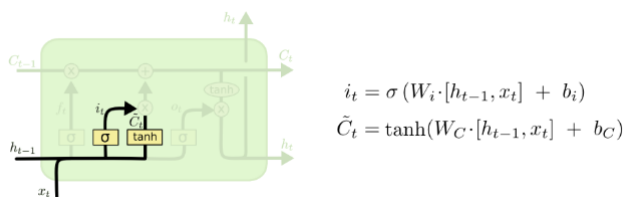
并且因为时间步是在训练层面上 所以预测时不需要考虑

遗忘门

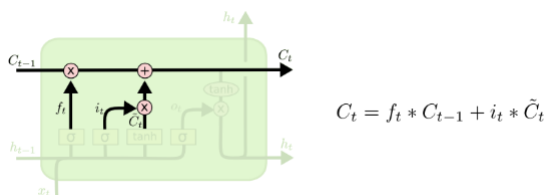


根据输入得到一个缩放系数 作用于上一次时间片的细胞状态

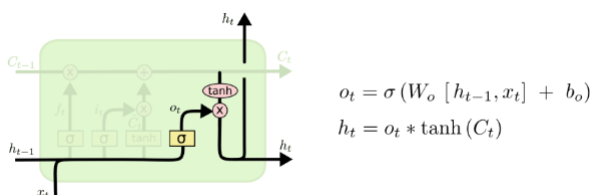
输入门



根据输入得到对上一次细胞状态的更新



输出门



根据细胞状态得到本次h h通过线性层得到目标输出

3. 并行化操作

类似于attention 其隐藏层存在大量复用的内部矩阵 这部分可完全可并行计算

但是attention输入的是一个sample 这里输入的是所有的sample

4. 多层操作

层数:将得到的h作为输入 继续输出到下一层lstm内

隐藏层数:每一层LSTM都存在n个隐藏层 每一个输入都对应n个隐藏状态(h)和细胞状态(这里隐藏层和输入的sequence不一定需要一一对应 隐藏层增加 并行运算加快)

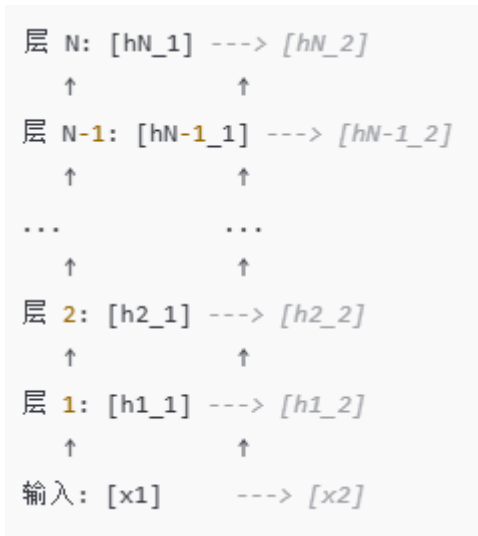
层数代表串接的级数 而隐藏层数相当于CNN中的多图层

问题:这里A的数量不是根据训练sample_num确定死的吗---A是可变的 实际更新的是三门的参数

隐藏层和层数的详细解释

main ideas:

1. 每一个层都根据输入的时间片存在T个黑盒 黑盒之间的相互输出就代表隐藏层的信息传递



- 2. 隐藏层控制h的维度 隐藏层控制网络中中间矩阵的维度
- 3. 实际计算参数h c矩阵维度不变，细胞的个数和训练集，测试集的数据维度不影响实际的模型

Transformer深入