知识卡片 循环神经网络 RNN

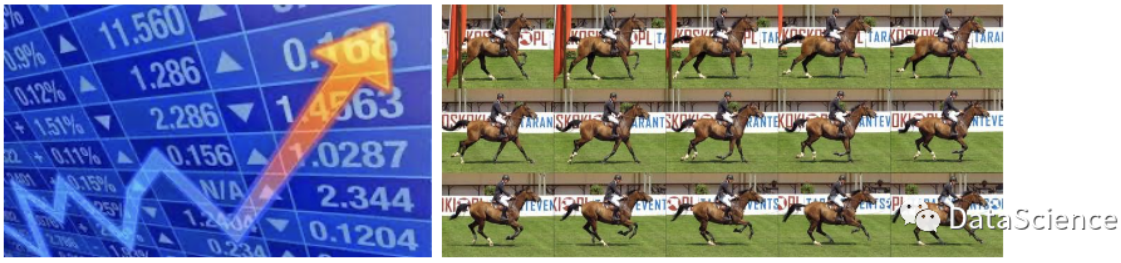
原创 Data君 DataScience 2020-08-14

前言：本文简要介绍了循环神经网络RNN以及其变体长短时记忆LSTM和双向循环网络。

循环神经网络

RNN-Recurrent Neural Network

序列数据

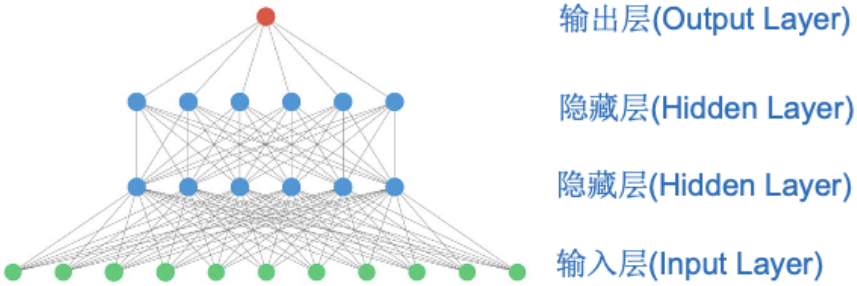


RNN建模的适合于序列数据，例如根据股票价格随时间的走势预测未来；视频中的每一帧属于帧序列，可以预测下一帧的内容，进行动作补偿。

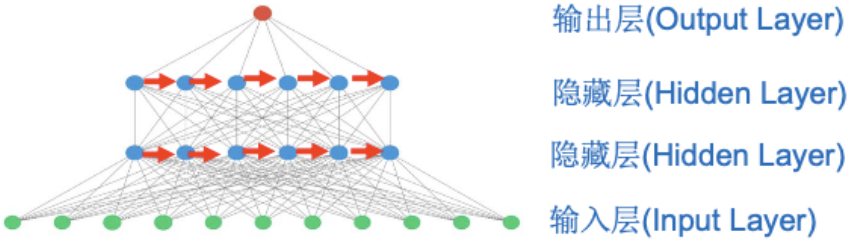
自然语言处理中，如大话西游的台词，这里的括号填什么呢？不可能填写我没有去北京，上海，因为需要上下文的词序列来进行预判，输入法打字也是同样的原理；此外，在机器翻译中，将源语言和目标语言中，也存在着上下文衔接的词序列，因而RNN也可以被用在机器翻译中。

什么是循环神经网络?

- 传统的神经网络模型，隐藏层的节点之间是无连接的。

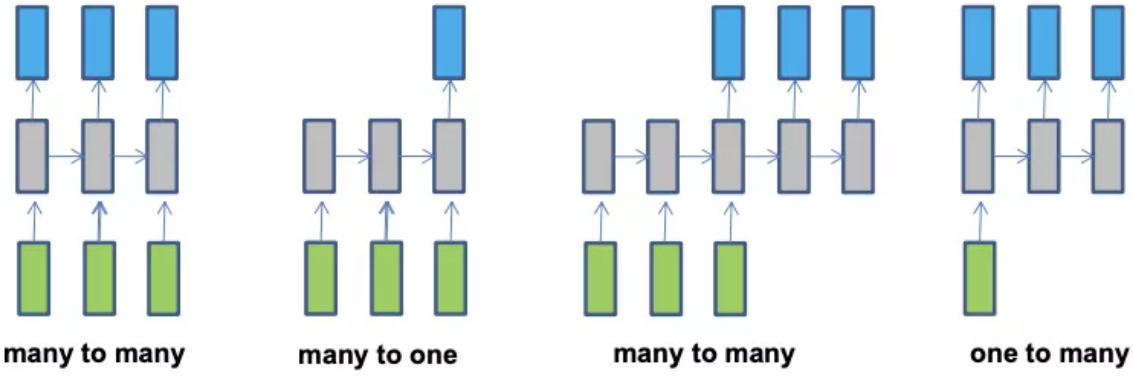


- 循环神经网络 (Recurrent Neural Network，RNN)：隐藏层的节点之间有连接，是主要用于对序列数据进行分类、预测等处理的神经网络。



传统的神经网络模型，层与层之间是全连接，但是隐藏层内的节点没有连接。序列信息中，节点存在被前一刻记忆的影响，隐藏层中的节点接收上一个节点的信息。RNN被称为循环神经网络是其对一组序列的输入进行循环，重复同样的操作。

RNN序列处理



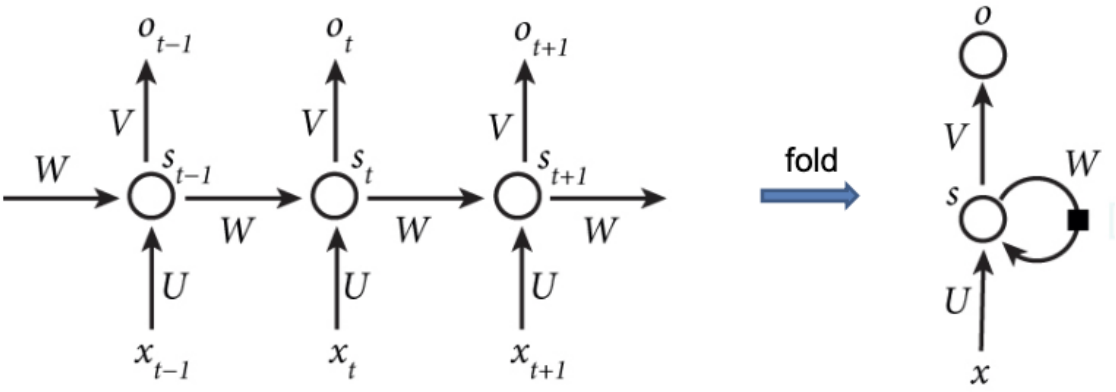
RNN处理序列的类型根据输入和输出的数量，有四种类型。绿色是输入，蓝色是输出，灰色是隐藏层，可捕捉序列前后的信息；并不是每一步都需要输入或者输出，但是隐藏层是不可少的。

同步序列中，Many to many 多对多，输入和输出的数量相同，可用在词性标注，输入一个句子，输出句中每个词的词性；Many to one 多对一，文本的情感分析，输入一句

话，输出这句话表达的情绪是积极还是消极。

非同步序列中，Many to many 多对多，可被用作机器翻译，即输入一种语言的文字，输出另外一种语言的文字；One to many，一对多，输入一张图片，输出对于图片内容的描述。

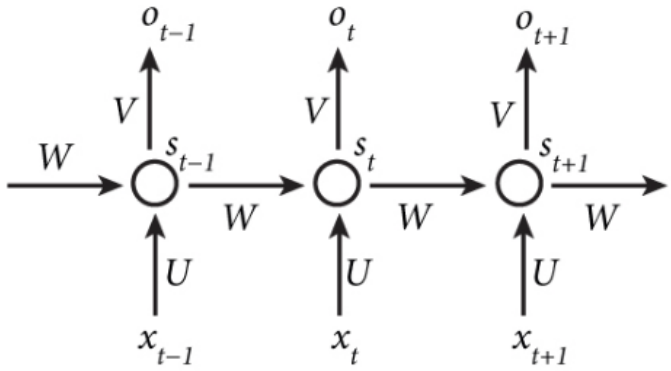
最基本的RNN结构



- 输入单元 (input units)为 $\{x_0, x_1, \dots, x_{t-1}, x_t, x_{t+1}, \dots\}$
- 输出单元 (output units)为 $\{o_0, o_1, \dots, o_{t-1}, o_t, o_{t+1}, \dots\}$
- 隐藏单元 (Hidden units)的输出标记为 $\{s_0, s_1, \dots, s_{t-1}, st, s_{t+1}, \dots\}$

从左往右看，中间的圆圈是隐藏单元为S，x和O是输入和输出，通过折叠S神经单元，旁边加上一个顺时针的箭头，可以简化表示为S循环。

基本RNN的计算过程

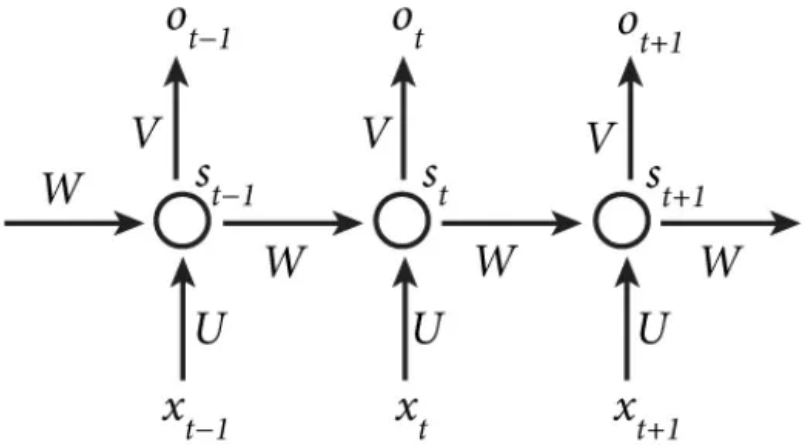


- 输入层：x_t 表示时刻 t 的输入.
- 隐藏层：$s_t = f(Ux_t + Ws_{t-1})$. 其中f是非线性激活函数，如tanh。
- 输出层：$o_t = softmax(Vs_t)$.

其中$softmax$函数的形式$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$。

以第二个神经元单元为例，xt 是向量，表示t时刻的输入，St是t时刻的记忆单元，St = f(U*xt + W*St-1)，f是非线性的激活函数 tanh 双曲线正切函数，作用是将输入的数据规范化，取值在[-1,1]，U和W是矩阵，对应 t时 和 t-1时（左边单元）的权重参数，Ot是t时的输出，用softmax 函数 归一化指数函数对矩阵V和向量st压缩并输出结果。
Softmax函数是逻辑函数Sigmoid的任意推广，将含有任意实数的k维的向量压缩至另外一个k维向量中，使得向量中的每个元素的范围都在[0,1]，并且所有元素的和为1，满足概率的性质。

RNN的参数共享



- 传统神经网络中，每一层的参数是不共享的；
- 而在RNNs中，每一步(每一层)都共享参数U，V，W。

RNN神经网络图中，每一条边都代表一个参数，不同于传统的神经网络，RNN在计算中共享U、V、W参数，即输出值Ot-1,Ot,Ot+1所用的U、V、W参数，这也是循环神经网络的特点，减少了需要学习的参数的数量，并提高了对数据进行训练的效率。

长短时记忆网络

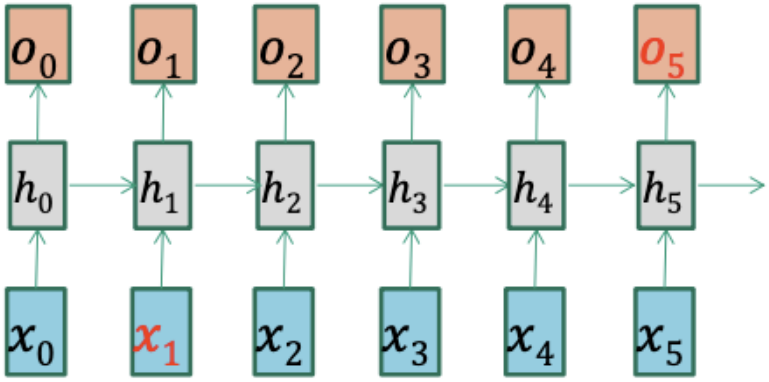
LSTM-Long Short-Term Memory

LSTM是RNN的一种变体，可以有效应对长期依赖的问题。

标准RNN难以应对长期依赖

● 标准RNN可以处理不太长的相关信息间隔：

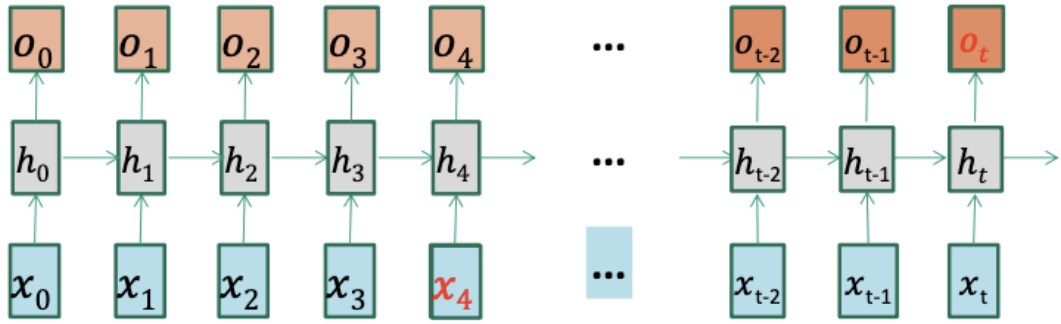
■ 例如，预测 “the clouds are in the ___” 空格中的词。



在文本预测中，空歌词距离先关信息“clouds”的间隔不长，可以填上“sky”。

● 但标准RNN无法处理更长的上下文间隔，即长期依赖问题。

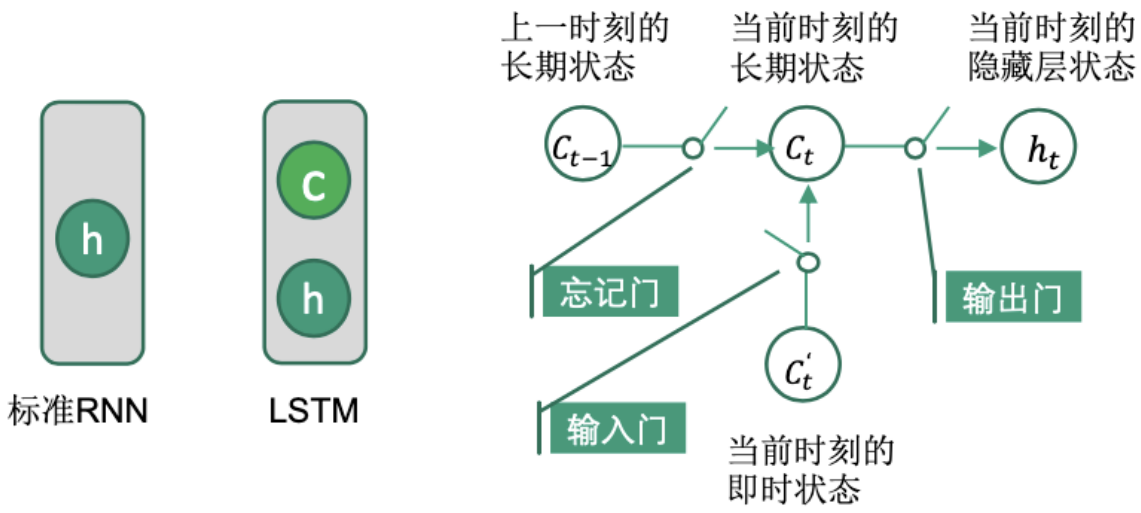
■ 例如，预测“I grew up in France... I speak fluent _____”最后的词。



预测文本中，我出生在法国，我说“ ”，可填“法语”，但在文本中因为上下文的距离较长，上文对下文的影响消失或削弱，导致RNN不能预测远处的内容。

LSTM 的基本思路

● LSTM(Long Short-Term Memory），即长短期记忆网络，是RNN的扩展，其通过特殊的结构设计来避免长期依赖问题。



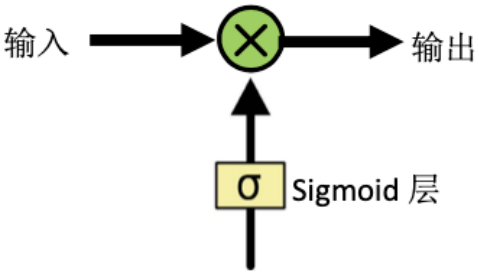
Hochreiter S, Schmidhuber J. Long short-term memory.[J]. Neural Computation, 1997, 9(8):1735-1780.

标准的RNN其隐藏层只有一个h，可以对短期的内容保持敏感，难以捕捉长期的上下文；LSTM在隐藏层的基础上增加一个长时状态c, 也叫 cell state 单元或细胞状态用于保存长期状态，无论是c还是h都是一个向量。

Ct 是当前输入对应的长期状态，由上一时刻的长期状态Ct-1和当前时刻的即时状态C't组成。然而，不能将所有的上一时刻的长期状态都保留，需要选择性的接收，使用一个忘记门，有选择地忘记一些长期信息。

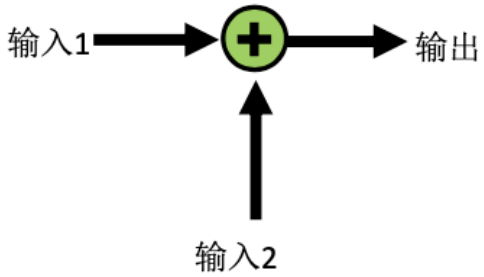
此外，当前时刻的长期状态还需要更新，因此通过输入门输入当前时刻的即时状态来更新。最后还有一个输出门，来控制如何使用当前时刻的长期状态来更新当前时刻的隐藏状态ht，此时ht中保存了一些长期的信息并和标准的RNN兼容；输出Ot时，还是使用当前时刻的ht来计算。

神经网络中的门



乘法门：

- 为了让信息选择性通过；
- sigmoid 层的输出矩阵中每个元素的范围是[0, 1]



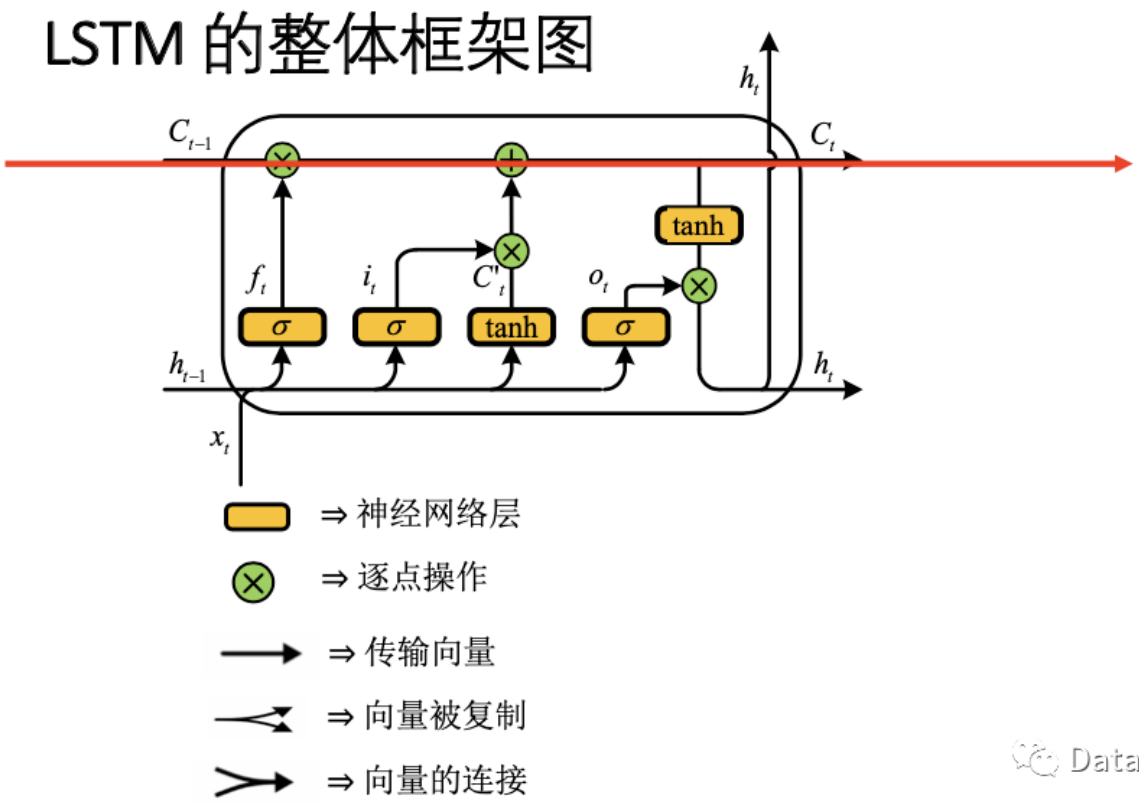
加法门：

- 在输入1基础上更新输入2的信息

因此，LSTM中忘记门和输出门要用到乘法门。输入门要用到加法门。

输入和输出都是尺寸相同的矩阵，对于其中的每个元素进行逐点操作。

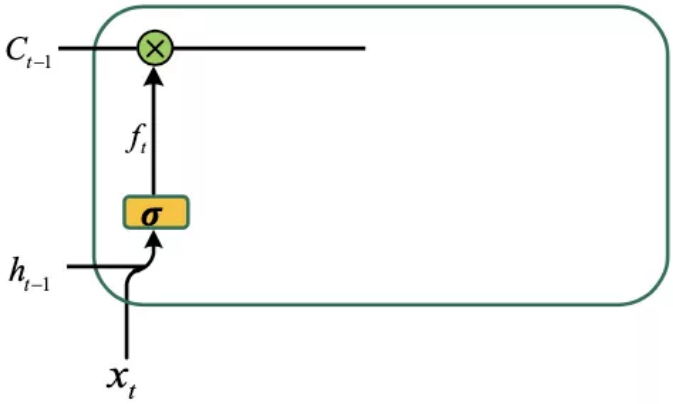
LSTM 的整体框架图



LSTM的难点是如何计算Ct，红色的水平线表示了长期信息的计算。

LSTM的计算过程

忘记信息：从长期状态中**丢弃某些信息**。

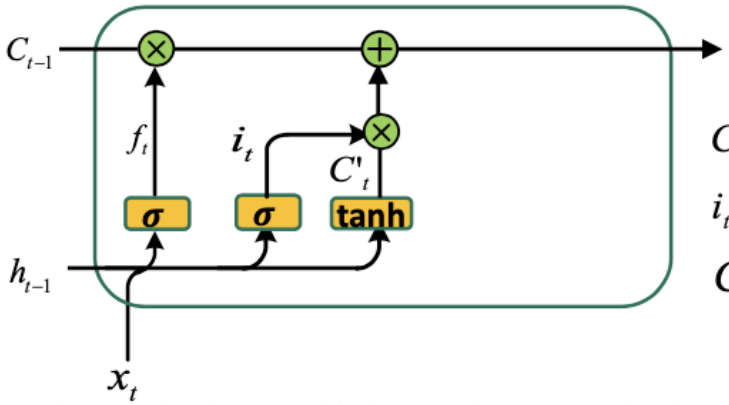


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

- 忘记门层 f_t 的输入为h_{t-1} 和 x_t，输出的矩阵中每个元素为 0 到 1 之间的数值，并与细胞状态矩阵C_{t-1}中的每个对应位置元素相乘。
- 语言模型例子：... Germany .... I grew up in France... I speak fluent ____。

σ是sigmoid函数，对应[0,1]，选择忘记还是记忆；语言模型中，Germany是距离远的长期信息，尝试忘记。

新记忆信息：将新信息存放在长期状态中。

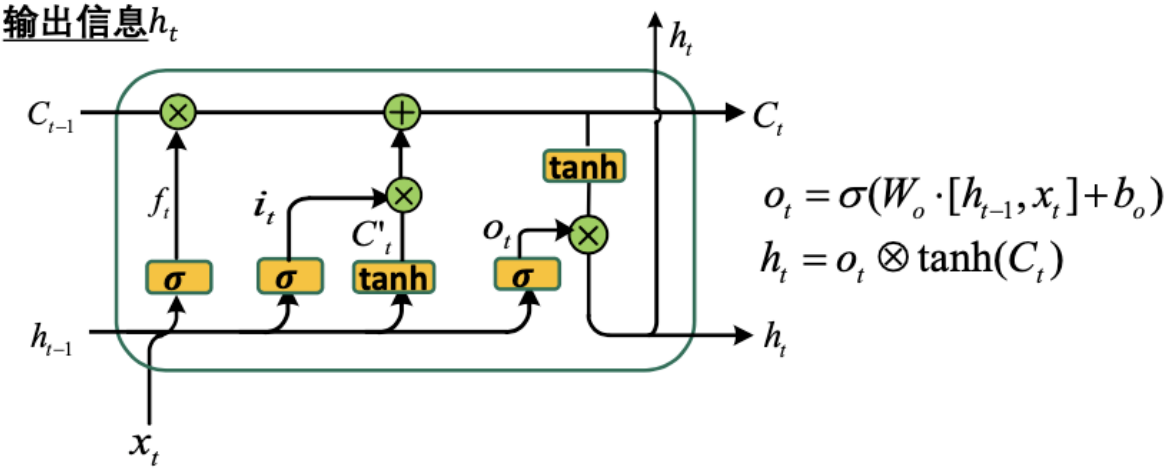


$$C_t' = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$C_t = f_t \otimes C_{t-1} + i_t \otimes C_t'$$

- 包含三个部分：1）首先，一个 tanh 层创建一个新的候选值向量；2）然后，sigmoid 层即输入门层 i_t 控制候选向量的哪些元素被更新；3）新的信息被加入到状态中。
- 语言模型例子： ... Germay .... I grew up in France... I speak fluent ____。

语言模型中，应将当前词 France，更新到Ct中。

输出信息h_t

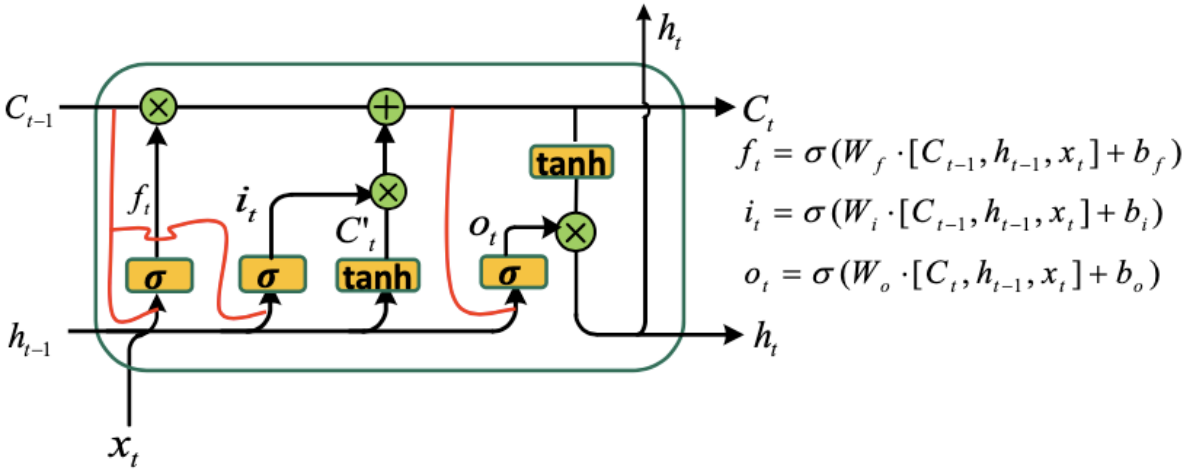


- 通过 sigmoid 层，来确定将输出哪些信息，即得到输出门o_t。
- 然后把长期状态通过 tanh 层进行处理，然后将其与经输出门过滤后的信息相乘，得到要输出的h_t。

得到输出的结果ht，经过复制后去往上方和下方，上方为通过后续的softmax函数计算，输出结果Ot；下方的ht可以被送入下一个单元进行计算。

LSTM 的变体-1

- 由 Gers & Schmidhuber (2000) 提出，增加了 “peephole connection”。门层也接受长期状态的输入。

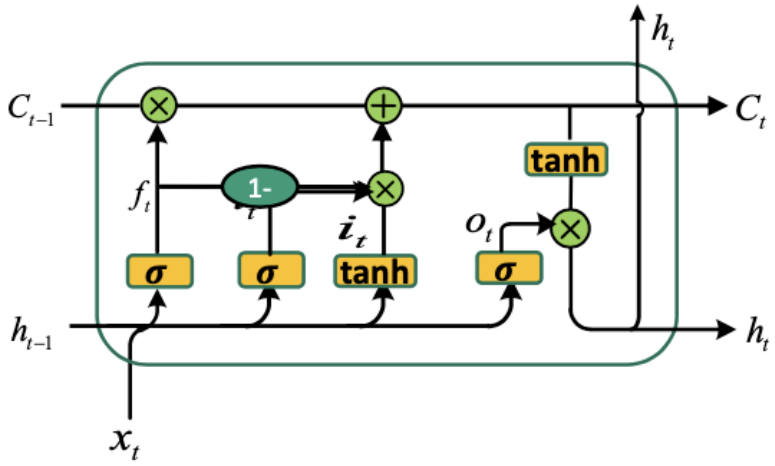


Gers, F. A., & Schmidhuber, J. (2000). Recurrent Nets that Time and Count. Ieee-Inns-Enns International Joint Conference on Neural Networks (Vol.3, pp.189-194 vol.3). IEEE.

变体将Ct-1放入了ft，it和Ot中，使得门层接收长期状态的输入。

LSTM 的变体-2

●耦合(coupled)遗忘和输入单元：将遗忘和新记忆两个过程耦合，即只遗忘那些有新元素来填充的元素。



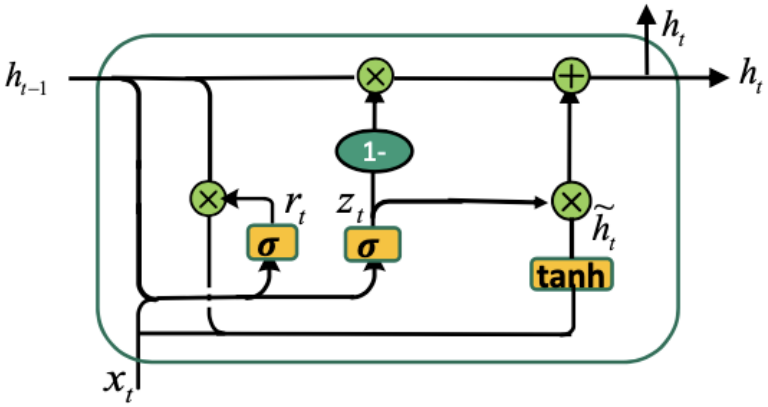
$$C_t = f_t \otimes C_{t-1} + i_t \otimes C_t'$$

$$C_t = f_t \otimes C_{t-1} + (1 - f_t) \otimes C_t'$$

将遗忘的记忆(1-ft)和新记忆C't进行耦合，将只有新元素来填充的元素遗忘。

LSTM 的变体-3

●即Gated Recurrent Unit Cho, et al. (2014)，混合了长期状态和隐藏状态。



$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \otimes h_{t-1}, x_t])$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$h_t = z_t \otimes \tilde{h}_t + (1 - z_t) \otimes h_{t-1}$$

- GRU只有两个门: 重置（reset）门r和更新（update）门z，取消了LSTM中的output门。r和z共同控制了如何从之前的隐藏状态（h_{t-1}）计算获得新的隐藏状态（h_t）。

Cho K, Van Merrienboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Eprint Arxiv, 2014.

简单的理解，GRU通过重置门R和更新门U，将隐藏状态（ht-1 上一个时刻的ht）与长期状态~ht进行混合得到新的隐藏状态ht。

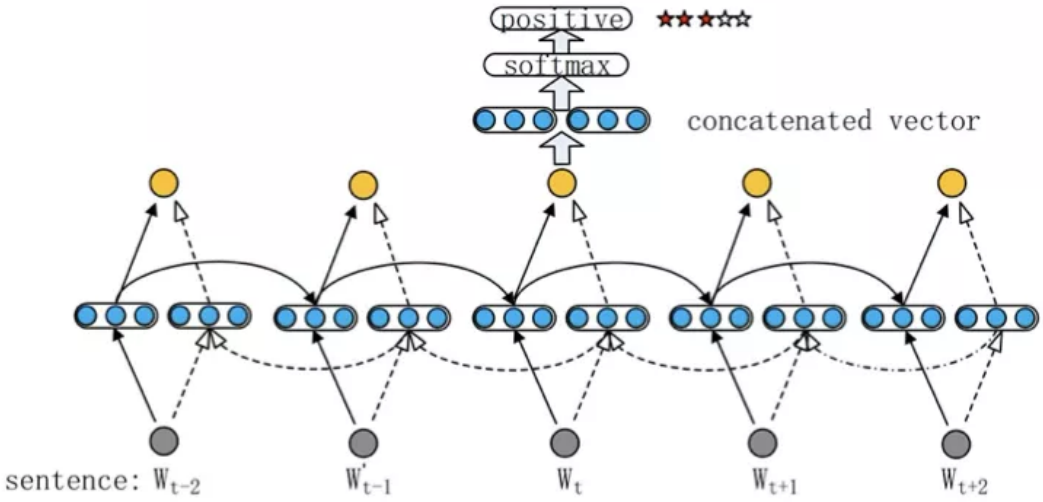
双向循环神经网络和注意力机制

Bidirectional RNN and Attention Mechanism

双向RNN(Bidirectional RNNs)

- 在很多应用中，当前步，即第t步的输出与前面的序列和后面的序列都有关。

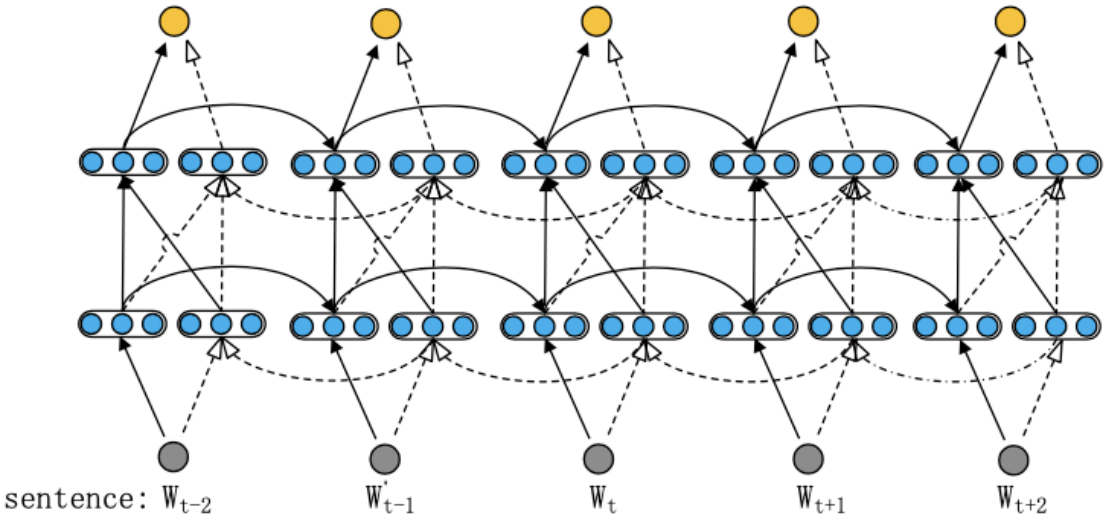
例如：“我喜欢宠物，家里养了一（zhi）可爱的小花猫。”，则括号内填“只”还是“支”？



*Schuster M, Paliwal K K. **Bidirectional recurrent neural networks[J]**. Signal Processing, IEEE Transactions on, 1997, 45(11): 2673-2681.*

在文本中，一个词的预测不仅与上文有关，也与下文有关，因此采用双向的RNN来进行预测更为准确，图中Wt由正反向的两个向量拼接组成拼接向量concatenated vector，再经过softmax函数进行归一化，输出结果。

深层双向RNN(Deep Bidirectional RNNs)



Graves A, Mohamed A R, Hinton G. Speech Recognition with Deep Recurrent Neural Networks[J]. Acoustics Speech & Signal Processing . icassp. international Conference on, 2013:66[illegible]5 - 6649.

深层双向RNN与RNN类似，增加了更多的隐藏层，具有更强大的学习和表达能力，同时也需要更多的数据来进行训练。

注意力模型(Attention model)

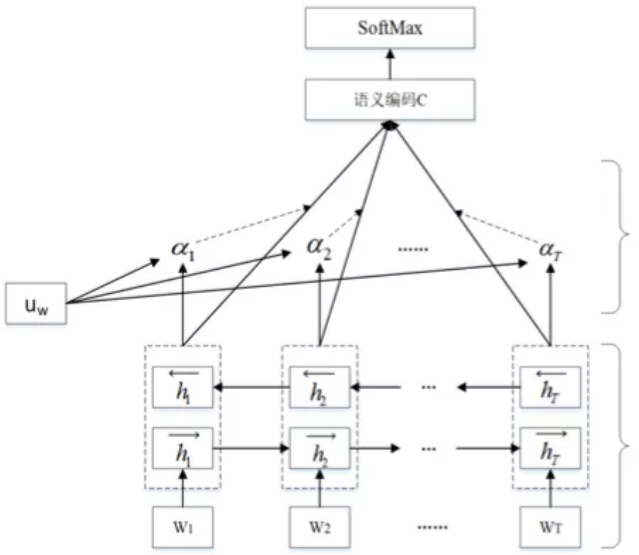
- 注意力模型（机制）是受到了人类注意力机制的启发。

- Google mind团队[1]在RNN模型上使用了attention机制进行图像分类。
- 后来Bahdanau等人[2]将attention机制应用到NLP领域中。如问答系统、自动文摘、文本分类等。

1. *Mnih, V., Heess, N., Graves, A., et al. Recurrent models of visual attention. NIPS 2014*
2. *Bahdanau, D., Cho, K., & Bengio, Y. Neural machine translation by jointly learning to align and translate. NIPS 2014*

注意力机制的简单描述，人类会将注意力集中在有特点的位置，下次遇到类似的场景会注意相同特点的位置。

注意力模型基本原理



1） $\theta_t = \tanh(W_w h_t + b_w)$

其中$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$,

W_w， b_w为权重参数和偏置参数

2） $\alpha_t = \frac{\exp(\theta_t{}^T u_w)}{\sum_{t=1}^{T} \exp(\theta_t{}^T u_w)}$

其中u_w为上下文向量

α_t为不同词的注意力概率分布

3) $C = \sum_{t=1}^{T} \alpha_t h_t$

其中 C为带有注意分布的语义编码

https://github.com/richliao/textClassifier

Yang Z, Yang D, Dyer C, et al. Hierarchical attention networks for document classification[C]//Proceedings of NAACL-HLT. 2016: 1480-1489.

上图左边部分以文本分类为例，输入用W表示为一个语句连续的若干个词。总体上来看，采用的是双向RNN，不同点在于对每个词都加入一个权重α，在获取语义编码C的时候，不同的词的权重不同。αt的取值由Uw决定，可以看做哪一个词是关键词的抽象表示。在训练过程中随机初始化，逐渐更新。

具体的更新形式，参考上图右边的公式：

（1) 将拼接层的隐藏节点通过双曲拼接层的变化得到θt

（2）将θt与uw点乘，得到归一化的αt，即不同词的注意力概率分布。

（3）αt和ht点乘求和，得到带注意分布的语义编码。

带有注意力机制的文本分词的好处是可以直观地看到每个词对分类的重要性。

案例推荐：

https://blog.csdn.net/qq_33431368/article/details/85288590

此文讲解RNN和LSTM的原理，可阅读加深对其理解，并用LSTM模型进行实战训练PTB(Penn Treebank Dataset) 宾州树库数据集。

好文章，我 在看♡

文章已于2020-08-14修改

喜欢此内容的人还喜欢

数据可视化 艺术字