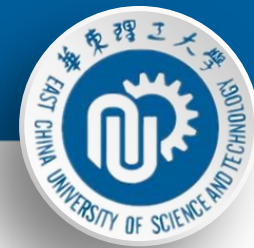


课时内容

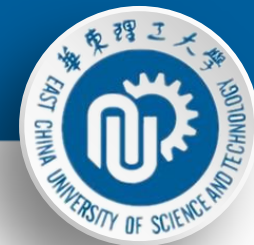
第10章 循环神经网络





■ 循环神经网络

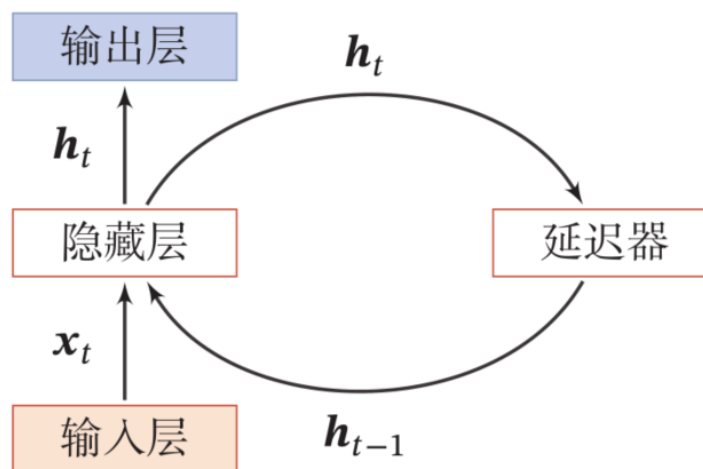
- a. 循环神经网络原理
- b. 标准循环神经网络(RNN)模型
- c. 长短时记忆网络(LSTM)模型
- d. 循环神经网络的应用



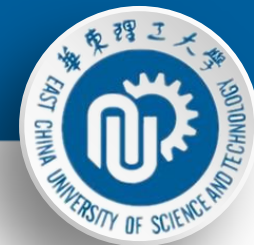
◆ 循环神经网络介绍

循环神经网络通过使用带自反馈的神经元，能够处理任意长度的时序数据。

$$h_t = f(h_{t-1}, x_t)$$

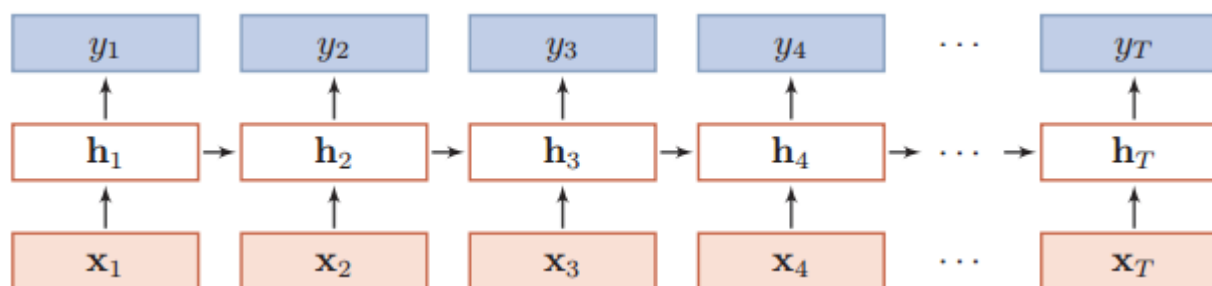


- 循环神经网络比前馈神经网络更加符合生物神经网络的结构。
- 循环神经网络已经被广泛应用于语音识别、语言模型以及自然语言生成等任务。

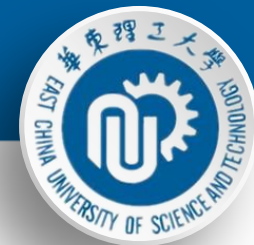


◆ 循环神经网络结构

首先只有一个隐藏层的循环神经网络的结构如下所示：



可以看到，连接不仅存在于相邻的层与层之间（比如输入层-隐藏层），还存在于时间维度上的隐藏层与隐藏层之间（反馈连接， h_1 到 h_T ）。



◆ 循环神经网络结构

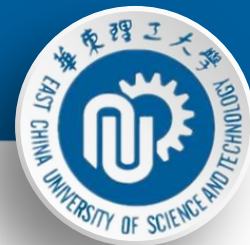
用公式来描述隐状态的计算过程，假设在时刻 t ，网络的输入为 x_t ，隐状态（即隐藏层神经元活性值） h_t 不仅和当前时刻的输入 x_t 相关，也和上一个时刻的隐状态 h_{t-1} 相关，进而与全部过去的输入序列（ $x_1, x_2, \dots, x_{t-1}, x_t$ ）相关。

$$\mathbf{z}_t = U\mathbf{h}_{t-1} + W\mathbf{x}_t + \mathbf{b},$$

$$\mathbf{h}_t = f(\mathbf{z}_t),$$

其中 \mathbf{z}_t 是隐藏层的净输入； $f(\bullet)$ 是非线性激活函数，通常为Sigmoid函数或Tanh函数； U 是状态-状态权重矩阵， W 是状态-输入权重矩阵， \mathbf{b} 为偏置。

注意，在所有的时刻，使用相同参数和相同的激活函数 $f(\bullet)$ 。



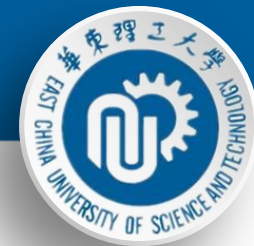
◆ 循环神经网络的模式

1、序列到类别的模式

序列到类别模式主要用于序列数据的分类问题：输入为序列（ T 个数据），输出为类别（一个数据）。典型的例子就是文本分类任务，输入数据为单词的序列（构成一篇文档），输出为该文本的类别。

假设有一个样本 $x_{1:T} = (x_1, x_2, \dots, x_T)$ 为一个长度为 T 的序列，输出为一个类别 $y \in \{1, 2, \dots, C\}$ 。将样本 x 按不同的时刻输入到循环神经网络中去，可以得到不同时刻的隐状态 h_1, h_2, \dots, h_T ，然后将 h_T 看做整个序列的最终表示，输入给分类器 $g(\bullet)$ 做分类。

$$\hat{y} = g(\mathbf{h}_T)$$



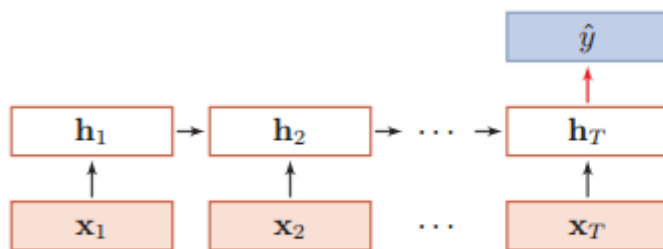
◆ 循环神经网络的模式

1、序列到类别的模式

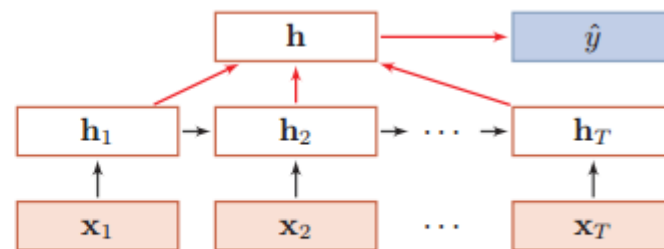
当然除了采用最后时刻的隐状态 h_T 作为序列的表示之外，还可以对整个序列的所有状态进行平均，用平均隐状态来作为整个序列的表示。

$$\hat{y} = g\left(\frac{1}{T} \sum_{t=1}^T h_t\right)$$

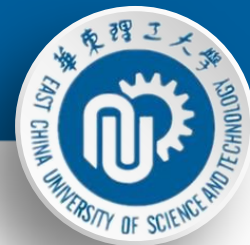
这两种序列到类别模式的图示如下：



(a) 正常模式



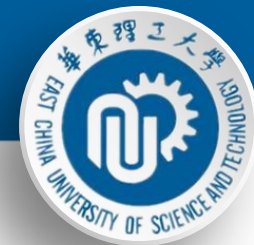
(b) 按时间进行平均采样模式



◆ 循环神经网络的模式

2、同步的序列到序列模式

同步的序列到序列模式主要用于序列标注任务，即每一时刻都有输入和输出，输入序列和输出序列的长度相同。比如词性标注 (Pos Tagging)，每个单词都需要标注它的词性。命名实体识别 (Name Entity Recognition, NER) 也可以看做是序列标注问题，与词性标注的做法类似，特点在于对于命名实体，输出它的命名实体标签来代替词性。

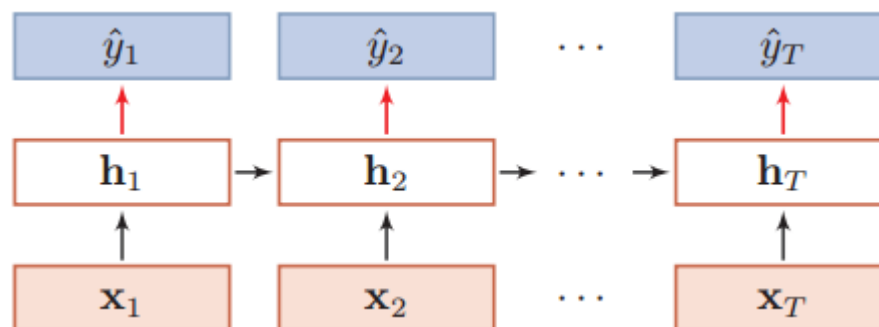


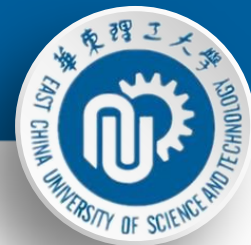
◆ 循环神经网络的模式

2、同步的序列到序列模式

假设有一个样本 $x_{1:T} = (x_1, x_2, \dots, x_T)$ 为一个长度为 T 的序列，输出序列为 $y_{1:T} = (y_1, y_2, \dots, y_T)$ 。将样本 x 按不同的时刻输入到循环神经网络中去，可以得到不同时刻的隐状态 h_1, h_2, \dots, h_T ，然后把每个时刻的隐状态输入给分类器 $g(\cdot)$ ，得到当前时刻的标签。

$$\hat{y}_t = g(h_t), \quad \forall t \in [1, T].$$



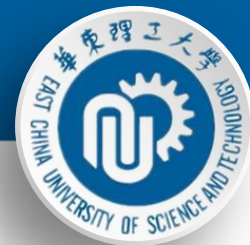


◆ 循环神经网络的模式

3、异步的序列到序列模式

异步的序列到序列模式也称为编码器-解码器 (Encoder-Decoder) 模型，即输入序列和输出序列不需要有严格的对应关系，也不用保持相同的长度。比如机器翻译中，输入为源语言的单词序列，输出为目标语言的单词序列。

在异步的序列到序列模式中，输入为一个长度为 T 的序列： $x_{1:T} = (x_1, x_2, \dots, x_T)$ ，输出一个长度为 M 的序列： $y_{1:M} = (y_1, y_2, \dots, y_M)$ ，通过先编码后解码的方式实现。



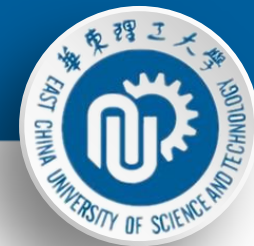
◆ 循环神经网络的模式

3、异步的序列到序列模式

先将样本 x 按不同时刻输入到一个循环神经网络（编码器）中，得到其编码 h_t ，然后在另一个循环神经网络（解码器）中得到输出序列 $\hat{y}_{1:M}$ 。为了建立输出序列之间的依赖关系，在解码器中通常使用非线性的自回归模型。

$$\begin{aligned}h_t &= f_1(h_{t-1}, x_t) & \forall t \in [1, T] \\h_{T+t} &= f_2(h_{T+t-1}, \hat{y}_{t-1}) & \forall t \in [1, M] \\\hat{y}_t &= g(h_{T+t}) & \forall t \in [1, M]\end{aligned}$$

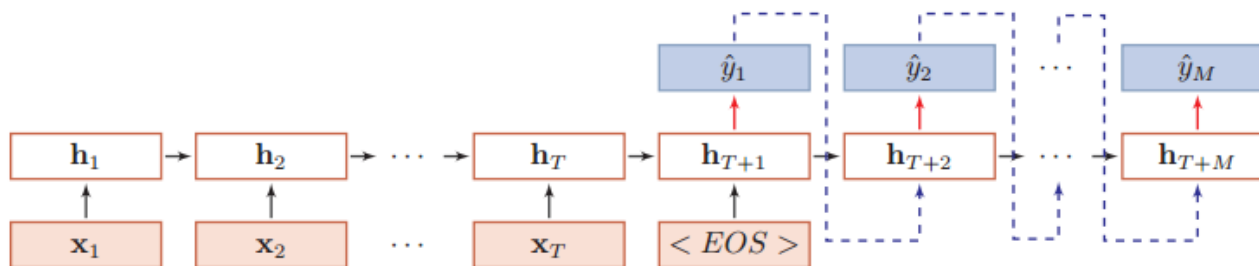
其中 $f_1(\bullet)$ 和 $f_2(\bullet)$ 分别表示用作编码器和解码器的循环神经网络， $g(\bullet)$ 为分类器。

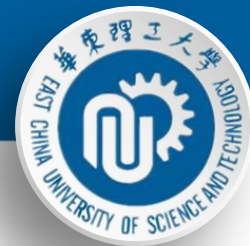


◆ 循环神经网络的模式

3、异步的序列到序列模式

基于上述描述，编码器和解码器的工作过程如下图所示：

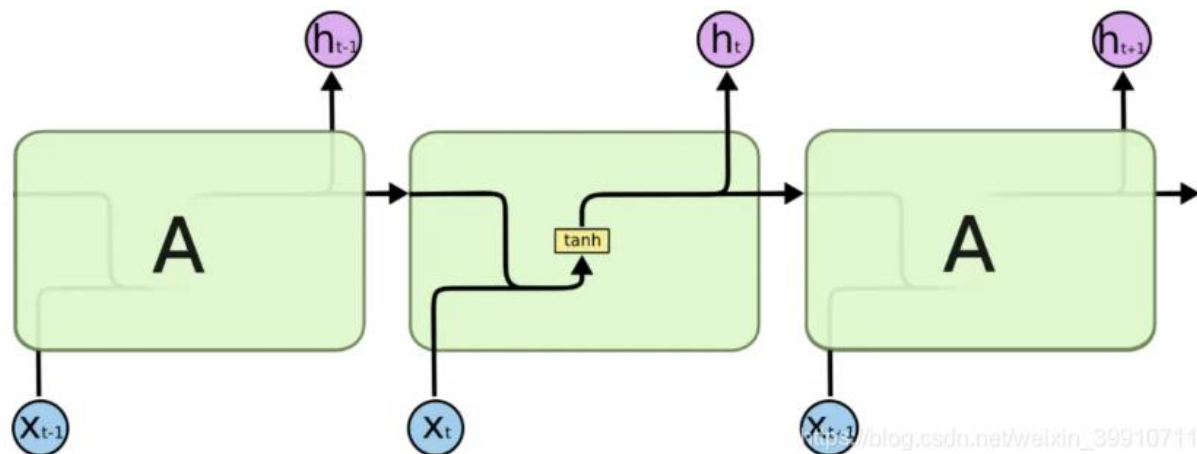




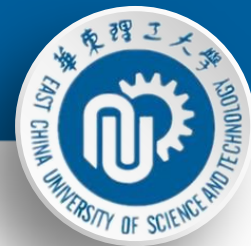
◆LSTM 与传统RNN的区别

所有循环神经网络都具有神经网络重复模块链的形式。在标准的RNN中，该重复模块将具有非常简单的结构。

标准 RNN 只有两个主要操作：将先前的隐藏状态与新输入组合并将其传递给激活函数：

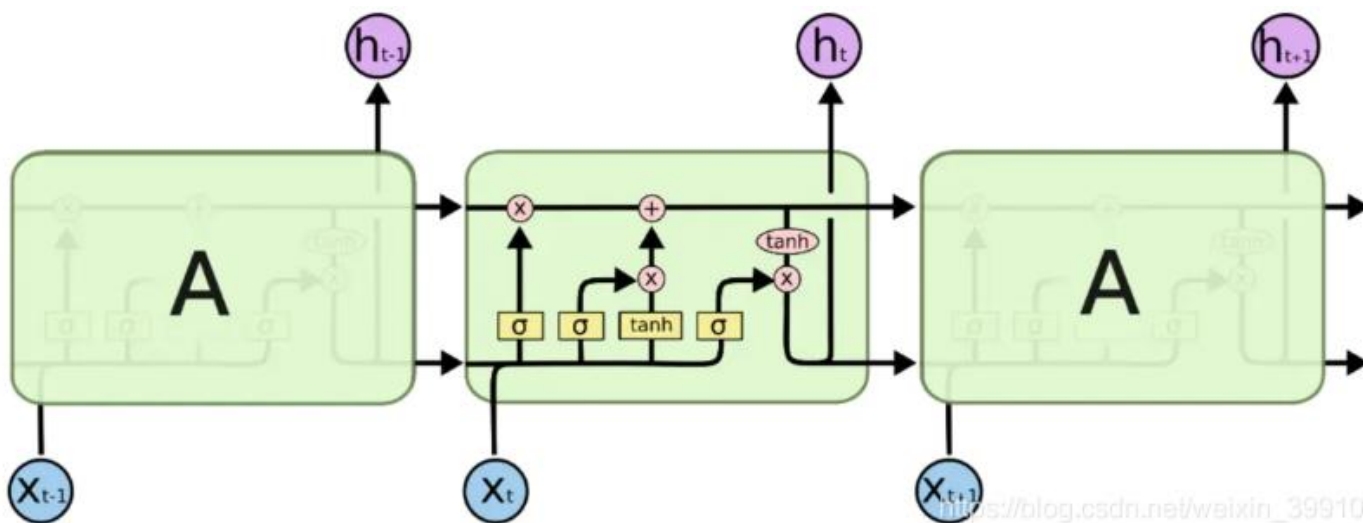


标准RNN结构图

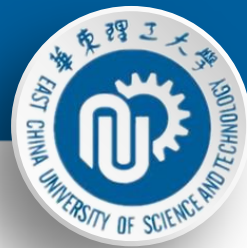


◆ LSTM 与传统 RNN 的区别

LSTM 的控制流程与 RNN 相似，它们都是在前向传播的过程中处理流经的数据，不同之处在于 LSTM 中细胞的结构和运算有所变化。



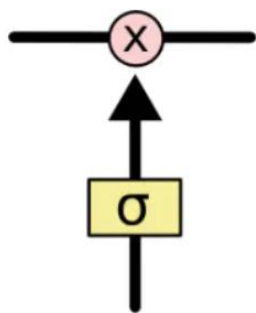
LSTM中结构图



◆ LSTM 门控制机制

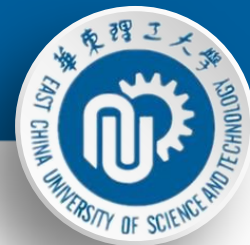
LSTM具有删除或添加信息到细胞状态的能力，这个能力是由被称为门(Gate)的结构所赋予的。为了控制记忆元，我们需要许多门。

门(Gate)是一种**可选地让信息通过**的方式。它由**一个Sigmoid神经网络层和一个点乘法运算**组成。

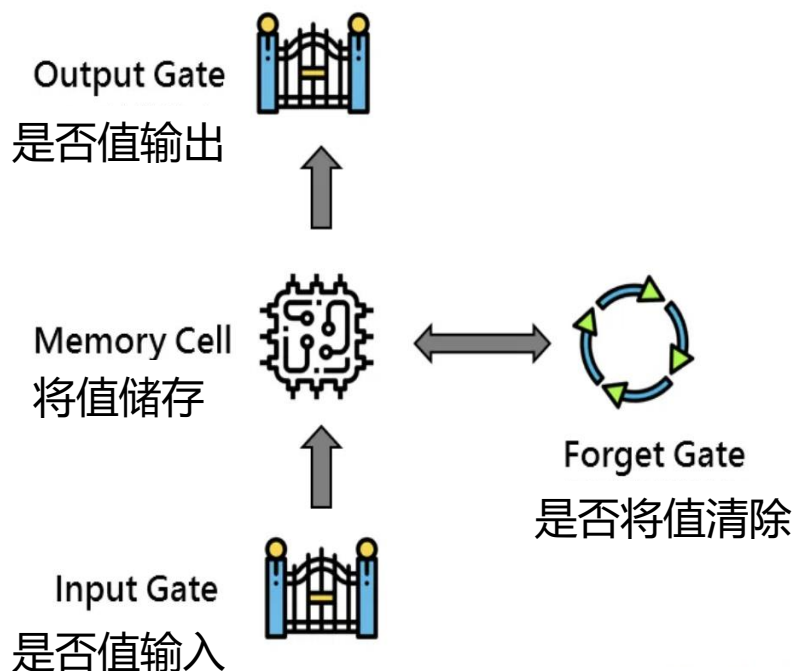


门结构示意图

Sigmoid神经网络层**输出0和1之间**的数字，这个数字描述每个组件有多少信息可以通过，**0表示不通过任何信息，1表示全部通过。**



◆ LSTM 门控制机制



LSTM 具有三个门，
以保护和控制单元状态。

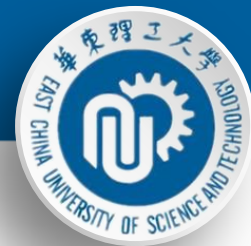
• **Input Gate:** 当将feature输入时，input gate会去控制是否将这次的值输入

• **Memory Cell:** 将计算出的值储存起来，以下个阶段拿出来使用

• **Output Gate:** 控制是否将这次计算出来的值output

• **Forget Gate:** 是否将Memory清掉(format)，类似restart

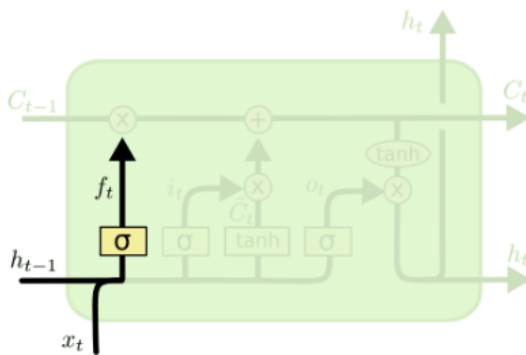
其中，“是否”这件事情，是可以通过神经网络进行学习



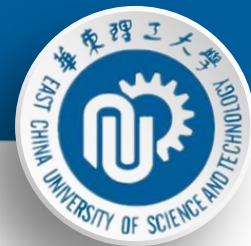
◆ Forget Gate 遗忘门

LSTM的第一步是决定我们要从细胞状态中丢弃什么信息，由称为**“遗忘门”**的**Sigmoid层**实现。它查看**前一个输出**和**当前输入**，并为单元格状态(上一个状态)中的每个数字输出0和1之间的数字。1代表完全保留，而0代表彻底删除。

在之前语言模型的例子，细胞状态可能包括当前主语的性别，从而决定使用正确的代词。当我们看到一个新主语时，我们想要忘记旧主语的性别。



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

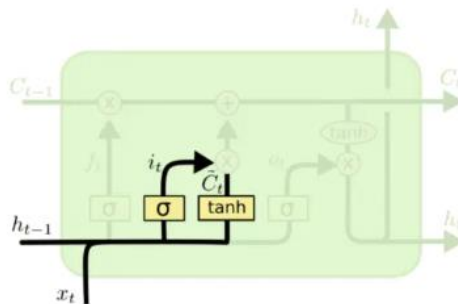


◆ Input Gate 输入门

输入门的作用是决定当前时间步长输入信息的哪些部分需要被更新到当前时间步长的状态信息中。可以分为两步：

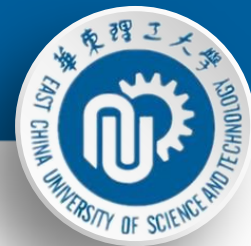
1. “输入门层”的Sigmoid层决定了我们将更新哪些值。
2. 一个tanh层创建候选向量 C_t ，该向量将会被加到细胞的状态中。

在语言模型的例子中，我们希望将新主语添加到单元格状态，以替换我们忘记的旧主语。



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

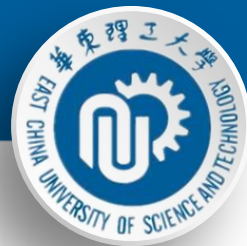
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$



◆ Memory Cell 记忆元

长短期记忆网络引入了记忆元 (memory cell) , 或简称为单元 (cell) 。

长短期记忆网络的隐藏层输出包括“隐状态”和“记忆元”。只有隐状态会传递到输出层, 而记忆元完全属于内部信息。

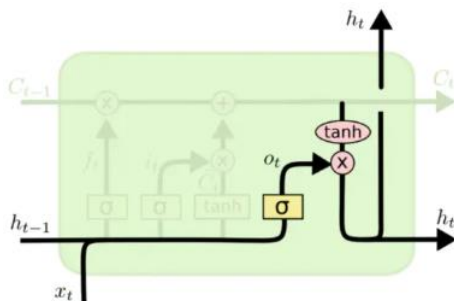


◆ Output Gate 输出门

首先，运行一个sigmoid层，它决定了我们要输出的细胞状态的哪些部分。

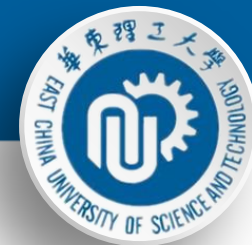
再将单元格状态通过tanh（将值规范化到-1和1之间），并将其乘以Sigmoid门的输出，至此我们只输出了我们决定的那些部分。

在语言模型中，输出的主语是单数还是复数，会影响我们后续输出动词的时态形式。

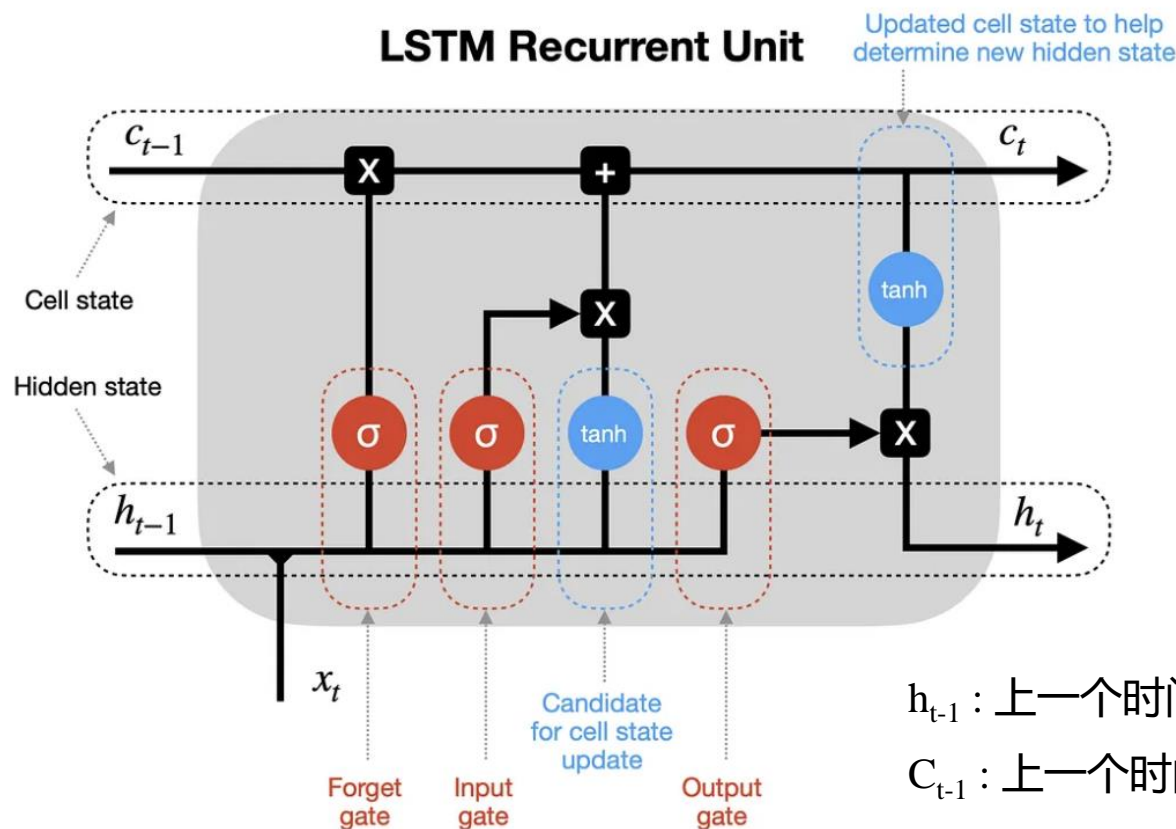


$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$



LSTM Recurrent Unit



LSTM基本结构总览图

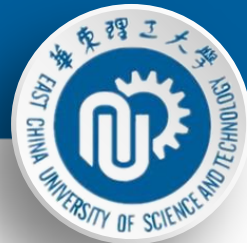
h_{t-1} : 上一个时间步骤 $t-1$ 的隐藏状态 (短期记忆)

C_{t-1} : 上一个时间步骤 $t-1$ 的细胞状态 (长期记忆)

x_t : 当前时间步骤 t 的输入向量 h ,

h_t : 当前时间步骤 t 的隐藏状态

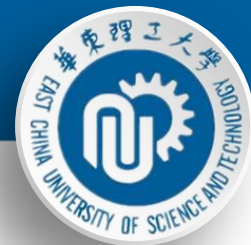
C_t : 当前时间步骤 t 的细胞状态



聚类分析：根据各类别之间的相似性对样本进行划分，对一组未知类别的样本集合，将相似的划分为一类，不相似的划分为另一类。

相似性：利用样本的几何特性，若以特征空间中两观测样本（看作两点）的距离函数作为模型相似性的测量，以“距离”作为分类的依据，则距离越小越相似。在此过程中，特征向量的选取就成为非常重要的一个环节。

例如：要识别并分开不同种类、不同品牌的雪碧和白醋，若以味道为特征，则很容易分别；但若以颜色为特征，则很难分开。



- **聚类准则：**根据相似性测度确定的样本之间是否相似的标准，即两样本向量之间的距离小于某一程度时，就认定两样本间是相似的，可归于同一类。常用的聚类准则有阈值准则和函数准则。
- **阈值准则：**根据规定距离阈值（实际情况中通常根据经验直接确定）进行分类。
- **函数准则：**用聚类分析函数来表示类别之间的相似性。通过定义一个准则函数，寻找该函数极值来解决聚类分析问题。