循环神经网络RNN

作者：凯鲁嘎吉 - 博客园 http://www.cnblogs.com/kailugaji/

在前馈神经网络中，信息的传递是单向的，这种限制虽然使得网络变得更容易学习，但在一定程度上也减弱了神经网络模型的能力，在生物神经网络中，神经元之间的连接关系要复杂得多，前馈神经网络可以看作一个复杂的函数，每次输入都是独立的，即网络的输出只依赖于当前的输入，但是在很多现实任务中网络的输出不仅和当前时刻的输入相关，也和其过去一段时间的输出相关。比如一个有限状态自动机，其下一个时刻的状态(输出)不仅仅和当前输入相关，也和当前状态(上一个时刻的输出)相关，此外，前馈网络难以处理时序数据，比如视频、语音、文本等，时序数据的长度一般是不固定的，而前馈神经网络要求输入和输出的维数都是固定的，不能任意改变因此，当处理这一类和时序数据相关的问题时，就需要一种能力更强的模型。

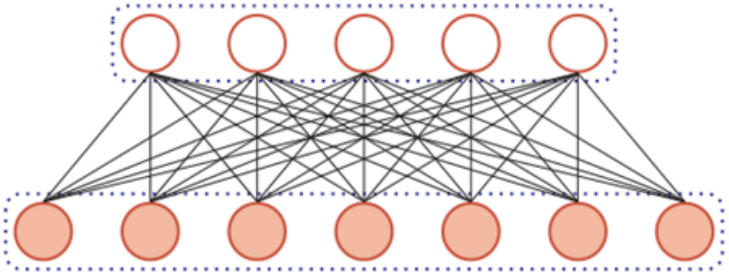
循环神经网络(Recurrent Neural Network，RNN)是一类具有短期记忆能力的神经网络，在循环神经网络中，神经元不但可以接受其他神经元的信息，也可以接受自身的信息，形成具有环路的网络结构，和前馈神经网络相比，循环神经网络更加符合生物神经网络的结构，循环神经网络已经被广泛应用在语音识别、语言模型以及自然语言生成等任务上循环神经网络的参数学习可以通过随时间反向传播算法来学习，随时间反向传播算法即按照时间的逆序将错误信息一步步地往前传递，当输入序列比较长时，会存在梯度爆炸和消失问题，也称为长程依赖问题。为了解决这个问题，人们对循环神经网络进行了很多的改进，其中最有效的改进方式引入门控机制(Gating Mechanism)。

本博文主要介绍循环神经网络，长短期记忆网络LSTM以及门控循环单元网络GRU。

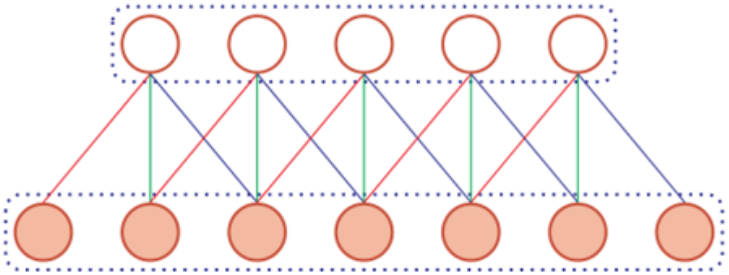
1．前馈网络存在的问题

前馈网络存在的问题

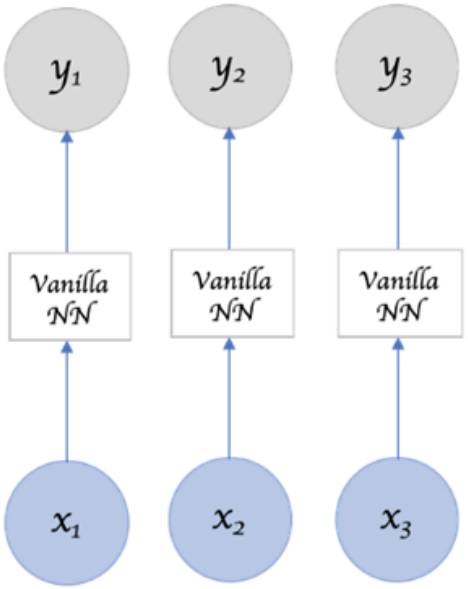
- 连接存在层与层之间，每层的节点之间是无连接的。（无循环）
- 输入和输出的维数都是固定的，不能任意改变。无法处理变长的序列数据。
- 假设每次输入都是独立的，也就是说每次网络的输出只依赖于当前的输入。



(a) 全连接层



(b) 卷积层



邱锡鹏，神经网络与深度学习，机械工业出版社，https://nndl.github.io/，2020.

激活函数

- Tansig

$$\tan sig(x) = \frac{2}{1+e^{-2x}} - 1$$

- Sigmoid(σ)

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$$

- Tanh

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

- ReLu

$$\text{ReLU}(x) = \max(x, 0)$$

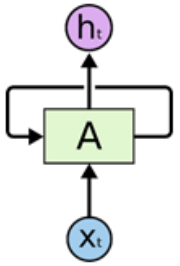
- Linear

$$\text{Linear}(x) = x$$

2. 循环神经网络（Recurrent Neural Network，RNN）

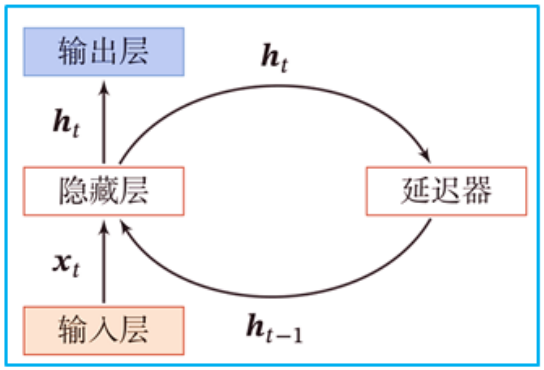
循环神经网络（Recurrent Neural Network，RNN）

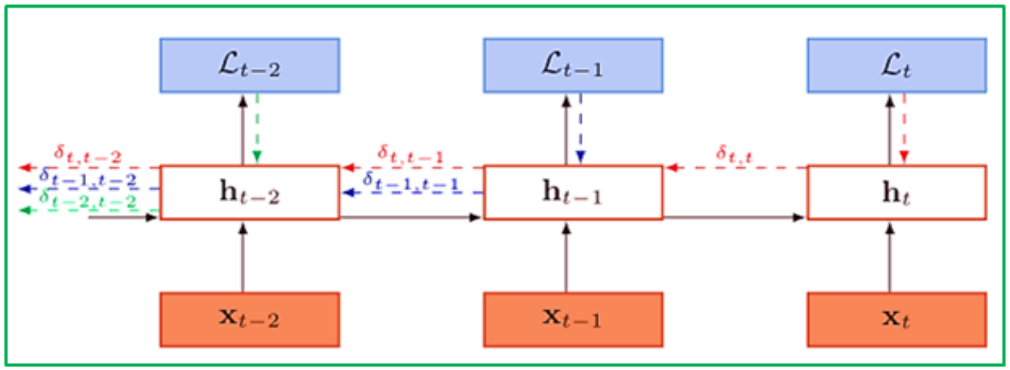
- 循环神经网络通过使用带自反馈的神经元，能够处理任意长度的时序数据。
- 循环神经网络比前馈神经网络更加符合生物神经网络的结构。
- 循环神经网络已经被广泛应用在语音识别、语言模型以及自然语言生成等任务上。



$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) \qquad \boldsymbol{h}_t = f(\boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{b}) \qquad \mathcal{L}_t = \mathcal{L}(\mathbf{y}_t, g(\mathbf{h}_t))$$

活性值状态　　状态更新　　时刻t的损失函数





$$\frac{\partial \mathcal{L}}{\partial U} = \sum_{t=1}^{T}\sum_{k=1}^{t} \delta_{t,k}\mathbf{h}_{k-1}^{\mathrm{T}}$$

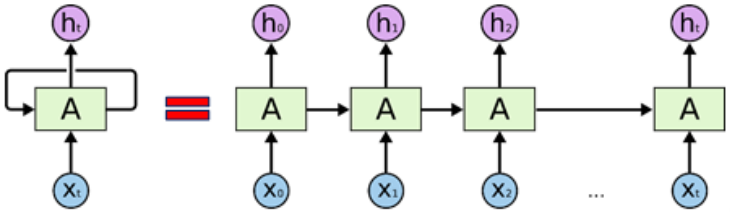
$$\delta_{t,k} = \prod_{\tau=k}^{t-1}\left(\mathrm{diag}(f'(\mathbf{z}_\tau))U^{\mathrm{T}}\right)\delta_{t,t}$$

$\delta_{t,k}$为第t时刻的损失对第k步隐藏神经元的净输入z_k的导数

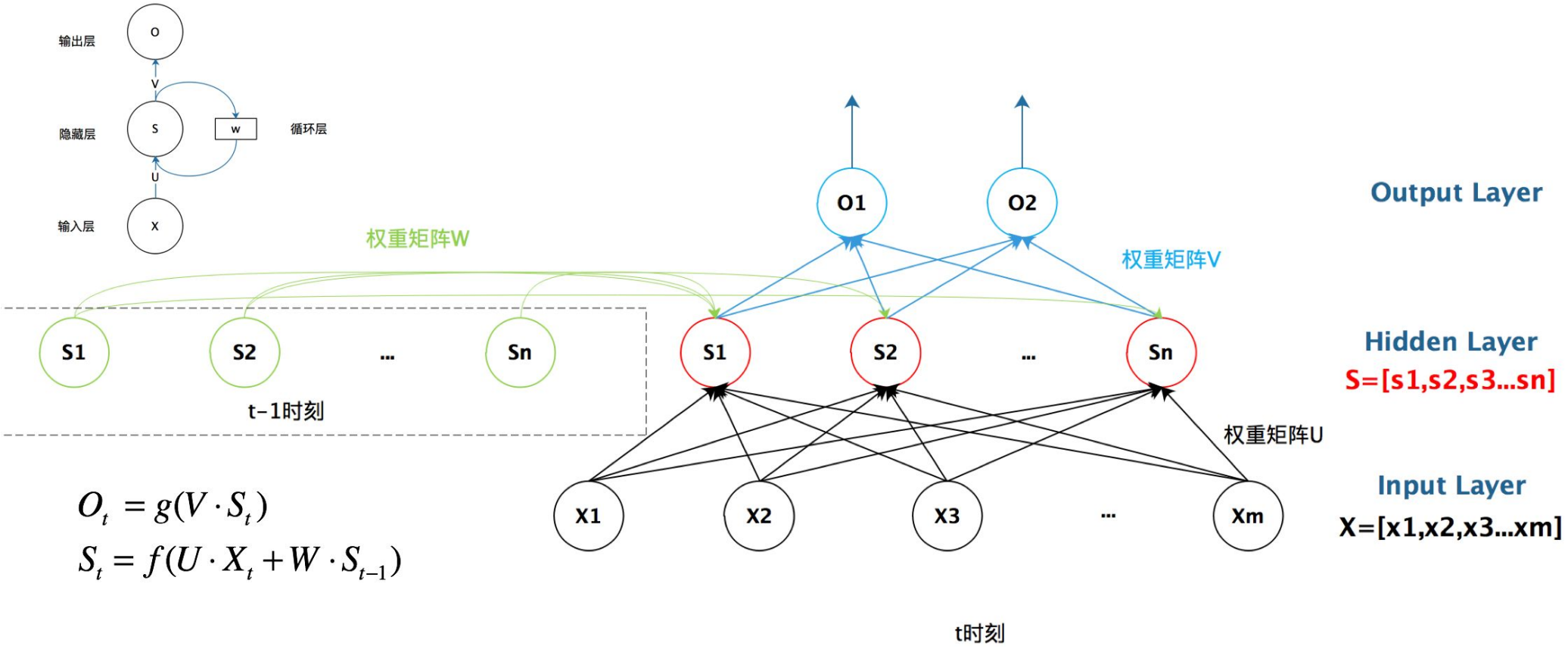
循环神经网络在时间维度上非常深，导致梯度爆炸或消失，实际上只能学习到短周期依赖关系，即长程依赖问题。

长程依赖问题改进方法

- 循环边改为线性依赖关系　$\mathbf{h}_t = \mathbf{h}_{t-1} + g(\mathbf{x}_t; \theta)$
- 增加非线性　$\mathbf{h}_t = \mathbf{h}_{t-1} + g(\mathbf{x}_t, \mathbf{h}_{t-1}; \theta)$



邱锡鹏，神经网络与深度学习，机械工业出版社，https://nndl.github.io/，2020.

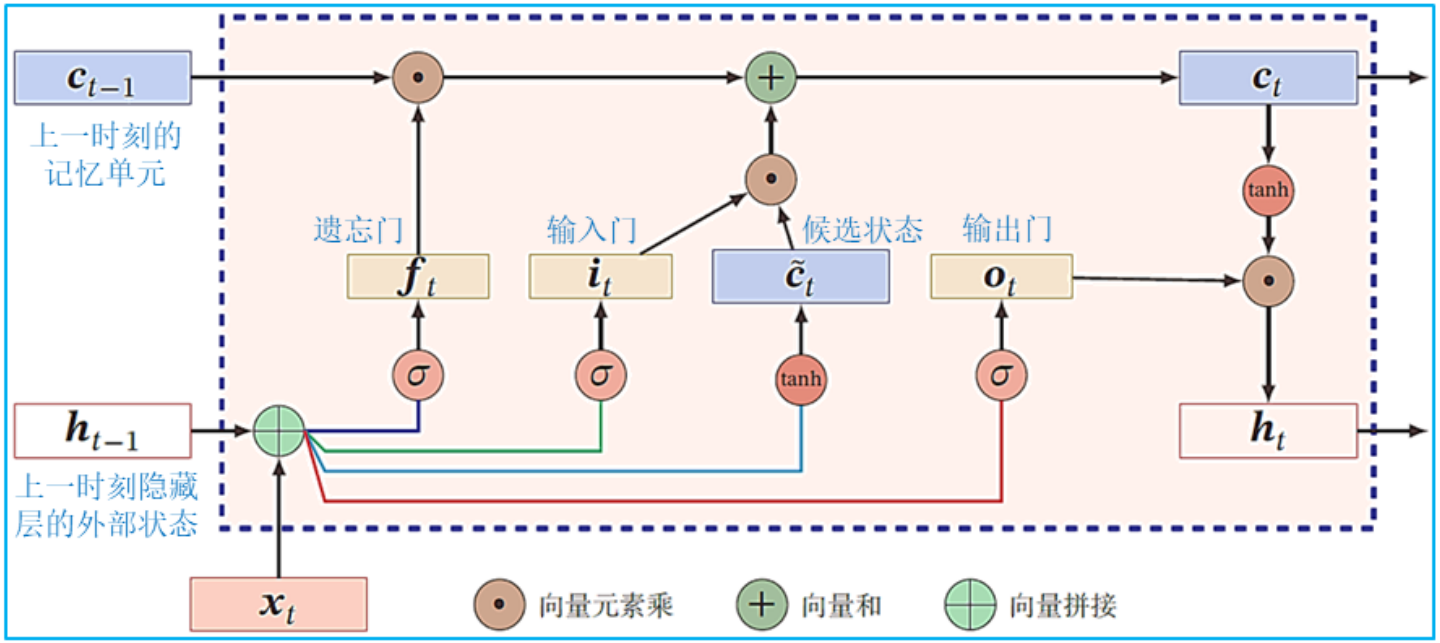


$$O_t = g(V \cdot S_t)$$

$$S_t = f(U \cdot X_t + W \cdot S_{t-1})$$

3. 长短期记忆神经网络（Long Short-Term Memory，LSTM）

➢ LSTM 网络的循环单元结构



LSTM网络引入一个新的内部状态(internal state) c_t 专门进行线性的循环信息传递，同时(非线性地)输出信息给隐藏层的外部状态h_t

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

$$h_t = o_t \odot \tanh(c_t)$$

其中c_{t-1}为上一时刻的记忆单元，⊙为向量元素乘积。

在每个时刻t，LSTM网络的内部状态c_t记录了到当前时刻为止的历史信息。

- 输入门：$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$

控制当前时刻的候选状态有多少信息需要保存

- 遗忘门：$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$

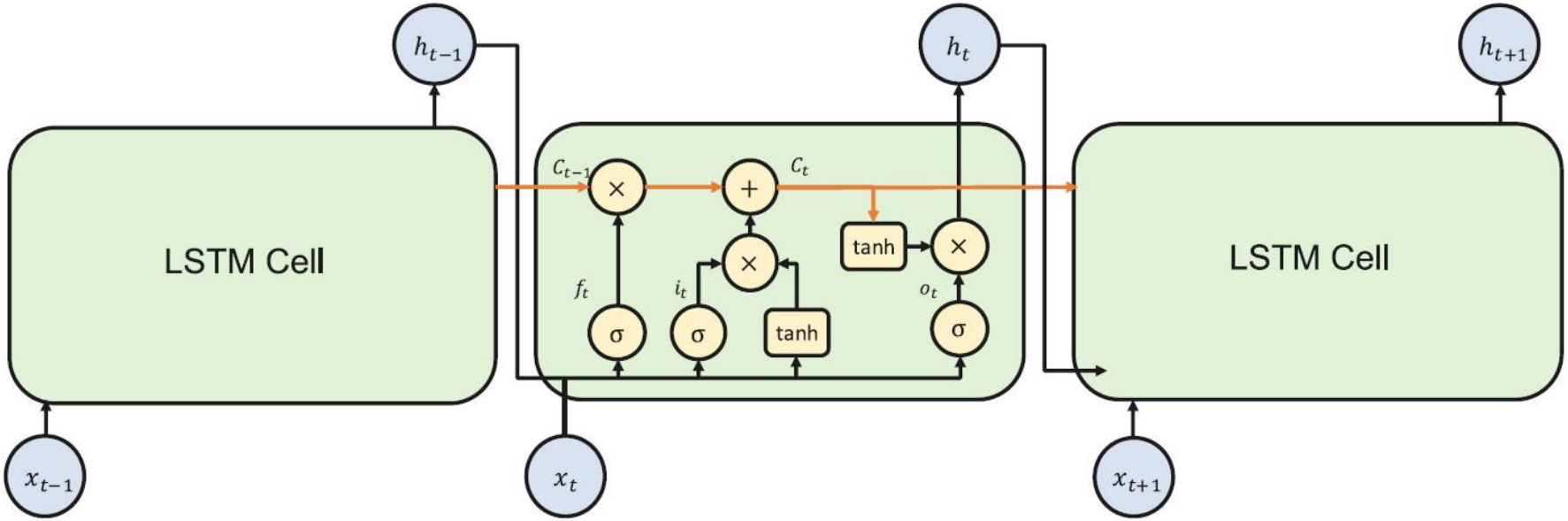
控制上一个时刻的内部状态需要遗忘多少信息

- 输出门：$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$

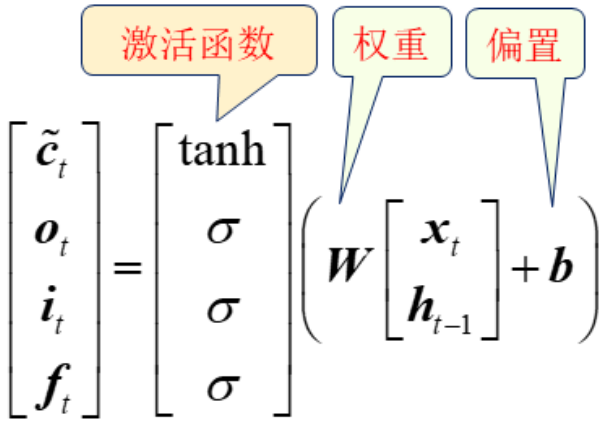
控制当前时刻的内部状态有多少信息需要输出给外部状态

- 候选记忆单元：$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$

通过非线性函数得到的候选状态


h_{t-1}
h_t
h_{t+1}
LSTM Cell
C_{t-1}
C_t
f_t
i_t
o_t
σ
σ
tanh
tanh
σ
LSTM Cell
x_{t-1}
x_t
x_{t+1}

➢ LSTM 网络公式重新整理



$$\begin{bmatrix} \tilde{\boldsymbol{c}}_t \\ \boldsymbol{o}_t \\ \boldsymbol{i}_t \\ \boldsymbol{f}_t \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} \left(\boldsymbol{W} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{h}_{t-1} \end{bmatrix} + \boldsymbol{b} \right)$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t)$$

其中$\boldsymbol{x}_t$为当前时刻的输入，$\boldsymbol{W}$和$\boldsymbol{b}$为网络参数.

➢ 记忆

- 循环神经网络中的隐状态$\boldsymbol{h}$ 存储了历史信息，可以看作一种记忆(Memory). 在简单循环网络中，隐状态每个时刻都会被重写，因此可以看作一种短期记忆(Short-Term Memory). 在神经网络中，长期记忆(Long-Term Memory)可以看作网络参数，隐含了从训练数据中学到的经验，其更新周期要远远慢于短期记忆. 而在LSTM 网络中，记忆单元$\boldsymbol{c}$ 可以在某个时刻捕捉到某个关键信息，并有能力将此关键信息保存一定的时间间隔. 记忆单元$\boldsymbol{c}$ 中保存信息的生命周期要长于短期记忆$\boldsymbol{h}$，但又远远短于长期记忆，因此称为长短期记忆(Long Short-Term Memory).
- 长短期记忆是指长的“短期记忆”.
- 一般在深度网络参数学习时，参数初始化的值一般都比较小. 但是在训练LSTM网络时，过小的值会使得遗忘门的值比较小. 这意味着前一时刻的信息大部分都丢失了，这样网络很难捕捉到长距离的依赖信息. 并且相邻时间间隔的梯度会非常小，这会导致梯度弥散问题. 因此遗忘的参数初始值一般都设得比较大，其偏置向量$\boldsymbol{b}_f$ 设为1 或2.

➢ LSTM网络的各种变体

目前主流的LSTM网络用三个门来动态地控制内部状态应该遗忘多少历史信息，输入多少新信息，以及输出多少信息．我们可以对门控机制进行改进并获得LSTM网络的不同变体。

◘ 无遗忘门的LSTM网络

$$\boldsymbol{c}_t = \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t.$$

◘ 耦合输入门和遗忘门

为了减少LSTM 网络的计算复杂度，将输入门与遗忘门合并为一个门

$$\boldsymbol{c}_t = (1 - \boldsymbol{i}_t) \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t.$$

◘ peephole连接

三个门不仅依赖于输入$\boldsymbol{x}_t$和上一时刻的隐状态$\boldsymbol{h}_{t-1}$，也依赖于上一个时刻的记忆单元$\boldsymbol{c}_{t-1}$

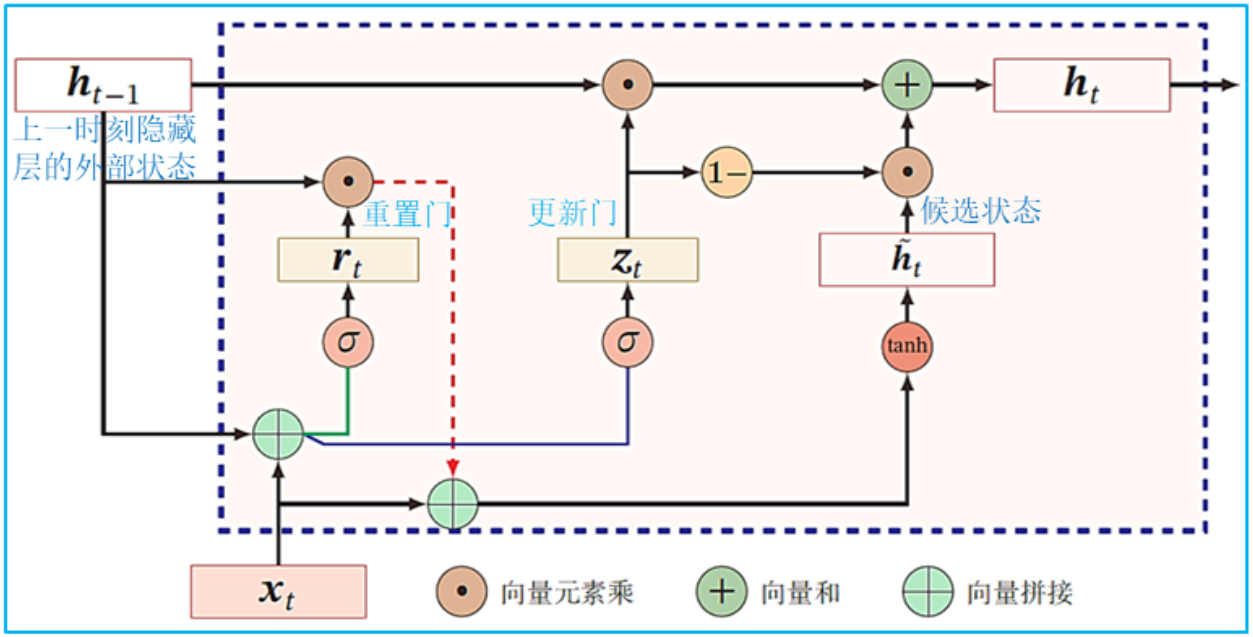
$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i\boldsymbol{x}_t + \boldsymbol{U}_i\boldsymbol{h}_{t-1} + \boldsymbol{V}_i\boldsymbol{c}_{t-1} + \boldsymbol{b}_i),$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f\boldsymbol{x}_t + \boldsymbol{U}_f\boldsymbol{h}_{t-1} + \boldsymbol{V}_f\boldsymbol{c}_{t-1} + \boldsymbol{b}_f),$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o\boldsymbol{x}_t + \boldsymbol{U}_o\boldsymbol{h}_{t-1} + \boldsymbol{V}_o\boldsymbol{c}_t + \boldsymbol{b}_o),$$

4. 门控循环单元网络（Gated Recurrent Unit，GRU）

➢ GRU网络的循环单元结构



GRU 网络引入一个更新门(Update Gate)来控制当前状态需要从历史状态中保留多少信息(不经过非线性变换)，以及需要从候选状态中接受多少新信息，GRU 网络的状态更新方式为

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{z}_t) \odot \tilde{\boldsymbol{h}}_t$$

其中$\boldsymbol{z}_t$为更新门

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z)$$

当前时刻的候选状态

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W}_h \boldsymbol{x}_t + \boldsymbol{U}_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}_h)$$

其中$\boldsymbol{r}_t$为重置门

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r \boldsymbol{x}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r)$$

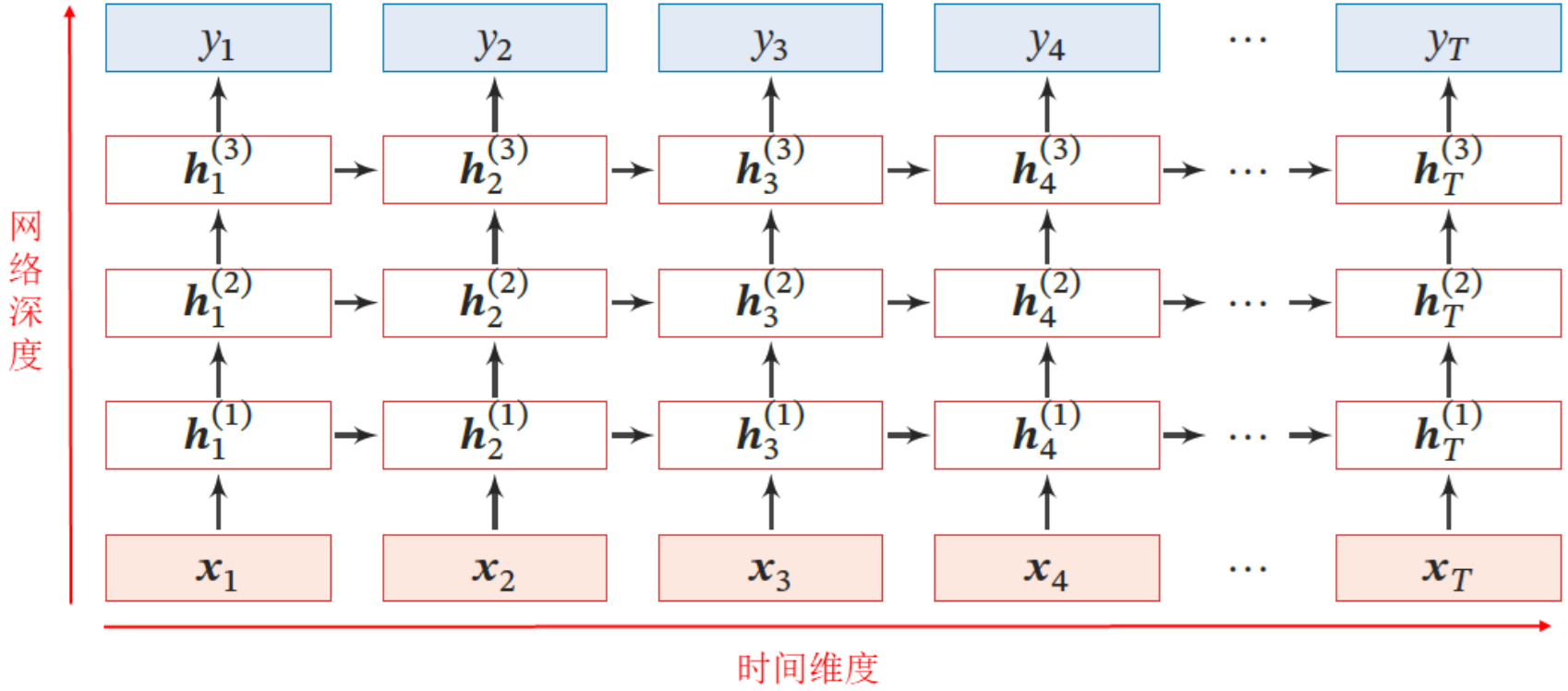
- 当$\boldsymbol{z}_t$=0, $\boldsymbol{r}_t$=1时，GRU 网络退化为简单循环网络；
- 当$\boldsymbol{z}_t$=0, $\boldsymbol{r}_t$=0时，当前状态$\boldsymbol{h}_t$只和当前输入$\boldsymbol{x}_t$相关，和历史状态$\boldsymbol{h}_{t-1}$无关.
- 当$\boldsymbol{z}_t$=1时，当前状态$\boldsymbol{h}_t$=$\boldsymbol{h}_{t-1}$等于上一时刻状态$\boldsymbol{h}_{t-1}$，和当前输入$\boldsymbol{x}_t$无关.

5. 深层循环神经网络

- 按时间展开的堆叠循环神经网络(Stacked Recurrent Neural Network, SRNN)

  将多个循环网络堆叠起来，第l层网络的输入是第l-1层网络的输出

$$\boldsymbol{h}_t^{(l)} = f(\boldsymbol{U}^{(l)}\boldsymbol{h}_{t-1}^{(l)} + \boldsymbol{W}^{(l)}\boldsymbol{h}_t^{(l-1)} + \boldsymbol{b}^{(l)}) \qquad \boldsymbol{h}_t^{(0)} = \boldsymbol{x}_t$$



邱锡鹏，神经网络与深度学习，机械工业出版社，https://nndl.github.io/，2020.

深层循环神经网络

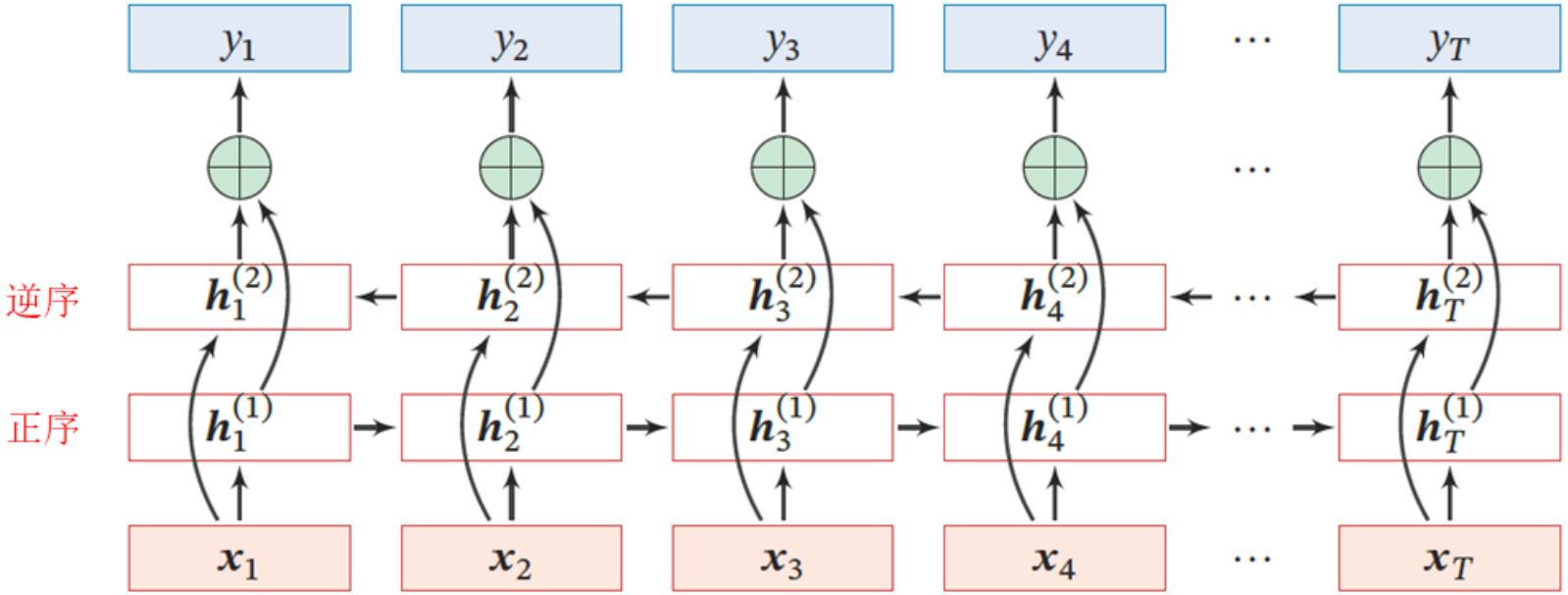
➢ 按时间展开的双向循环神经网络(Bidirectional Recurrent Neural Network, Bi-RNN)

增加一个按照时间的逆序来传递信息的网络层，来增强网络的能力．由两层循环神经网络组成，它们的输入相同，只是信息传递的方向不同．假设第1层按时间顺序，第2层按时间逆序

$$\boldsymbol{h}_t^{(1)} = f(\boldsymbol{U}^{(1)}\boldsymbol{h}_{t-1}^{(1)} + \boldsymbol{W}^{(1)}\boldsymbol{x}_t + \boldsymbol{b}^{(1)}), \qquad \boldsymbol{h}_t = \boldsymbol{h}_t^{(1)} \oplus \boldsymbol{h}_t^{(2)}$$

$$\boldsymbol{h}_t^{(2)} = f(\boldsymbol{U}^{(2)}\boldsymbol{h}_{t+1}^{(2)} + \boldsymbol{W}^{(2)}\boldsymbol{x}_t + \boldsymbol{b}^{(2)}),$$

其中⊕为向量拼接操作



邱锡鹏，神经网络与深度学习，机械工业出版社，https://nndl.github.io/，2020.

6．参考文献

[1] 邱锡鹏，神经网络与深度学习，机械工业出版社，https://nndl.github.io/, 2020.

[2] The Unreasonable Effectiveness of Recurrent Neural Networks http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[3] Understanding LSTM Networks -- colah's blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[4] [干货]深入浅出LSTM及其Python代码实现 - 知乎 https://zhuanlan.zhihu.com/p/104475016

[5] 一文搞懂RNN（循环神经网络）基础篇 - 知乎 https://zhuanlan.zhihu.com/p/30844905