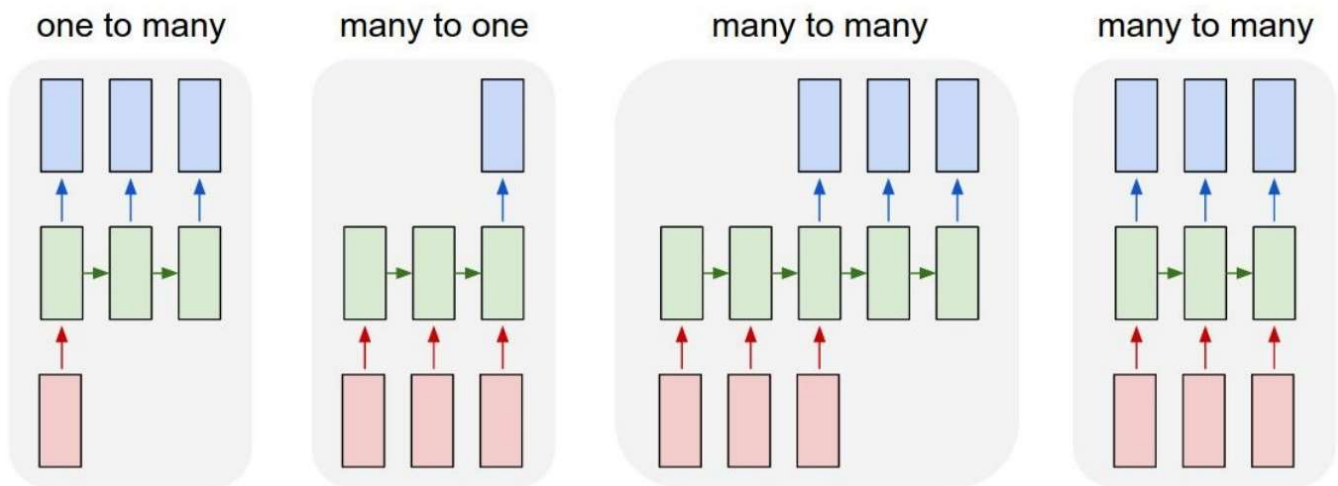


Lecture 10 Recurrent Neural Networks

一、连续样本处理



上图中存在着四种输入输出情况，分别对应四种应用场景：

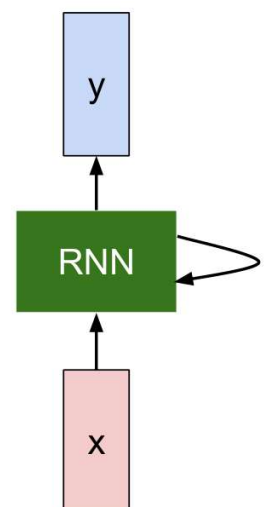
- one to many: 用于image captioning，也就是类似于输入一张图片，然后用一段话做描述。
- many to one: 情感分析，类似于一段话分析话的情感
- many to many(输入输出可变长度): 机器翻译，输入输出都可变长度
- many to many(输入输出不可变长度): 一段视频然后分析人物动作之类的

二、RNN

We can process a sequence of vectors \mathbf{x} by applying a **recurrence formula** at every time step:

$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state some function with parameters W old state input vector at some time step



基本的计算过程如下：

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

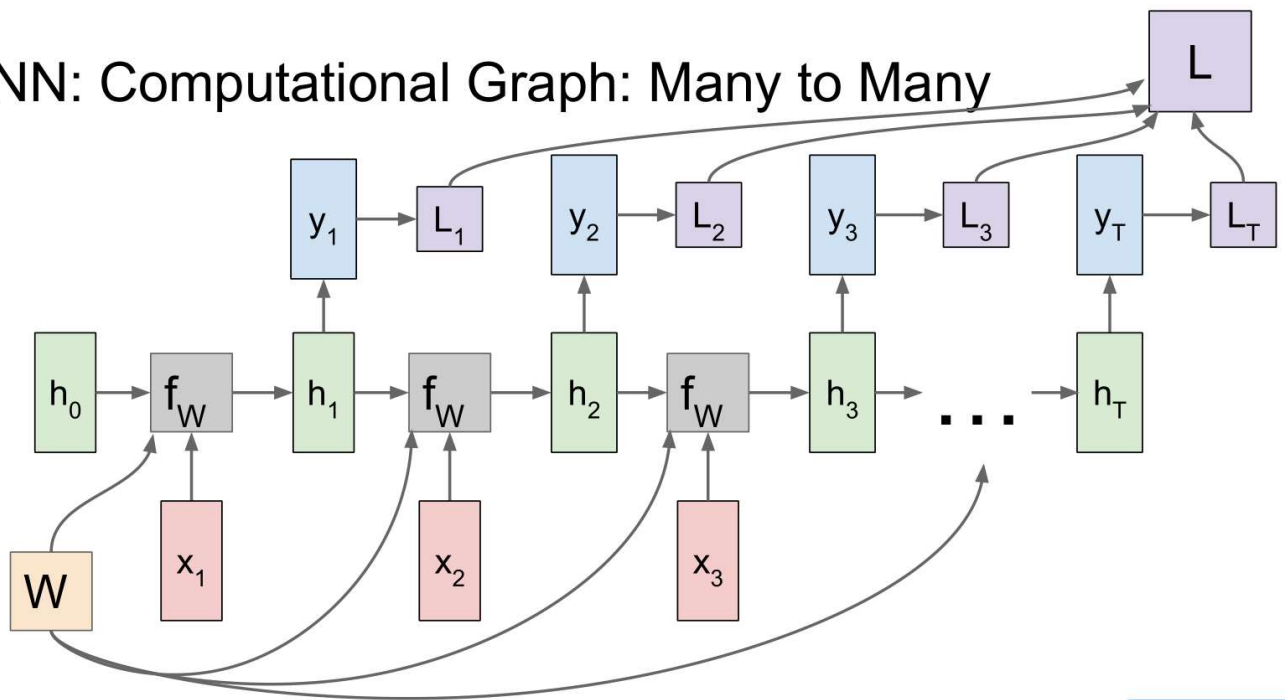


$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

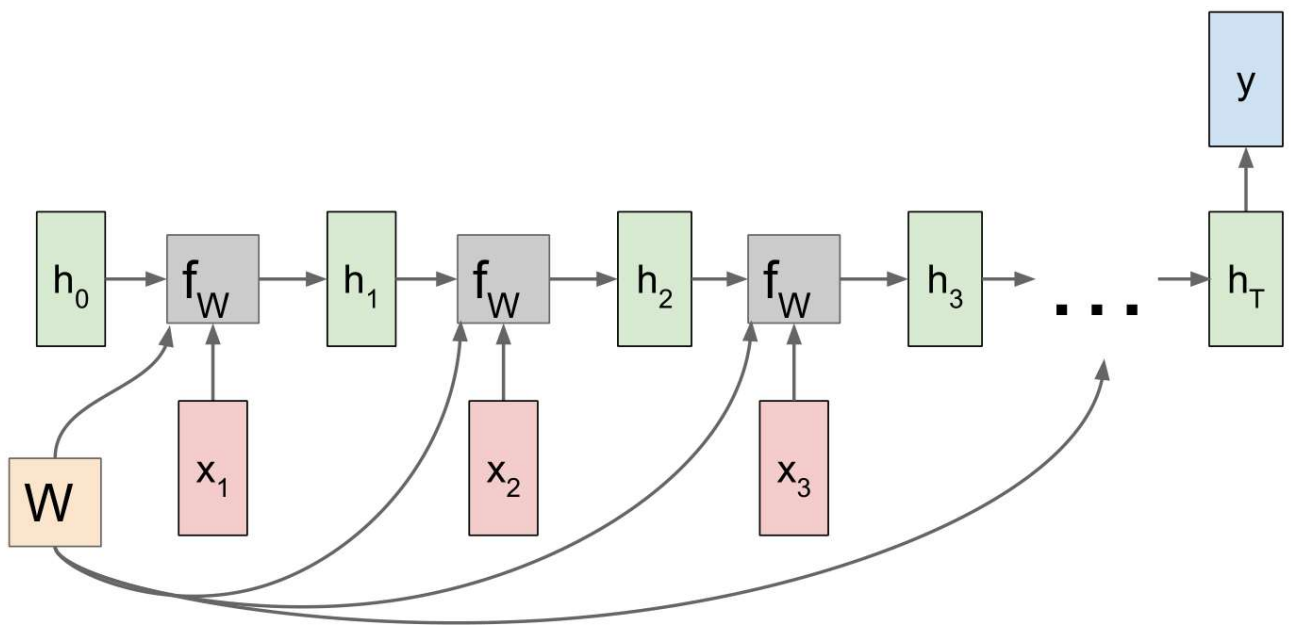
$$y_t = W_{hy}h_t$$

计算图示例

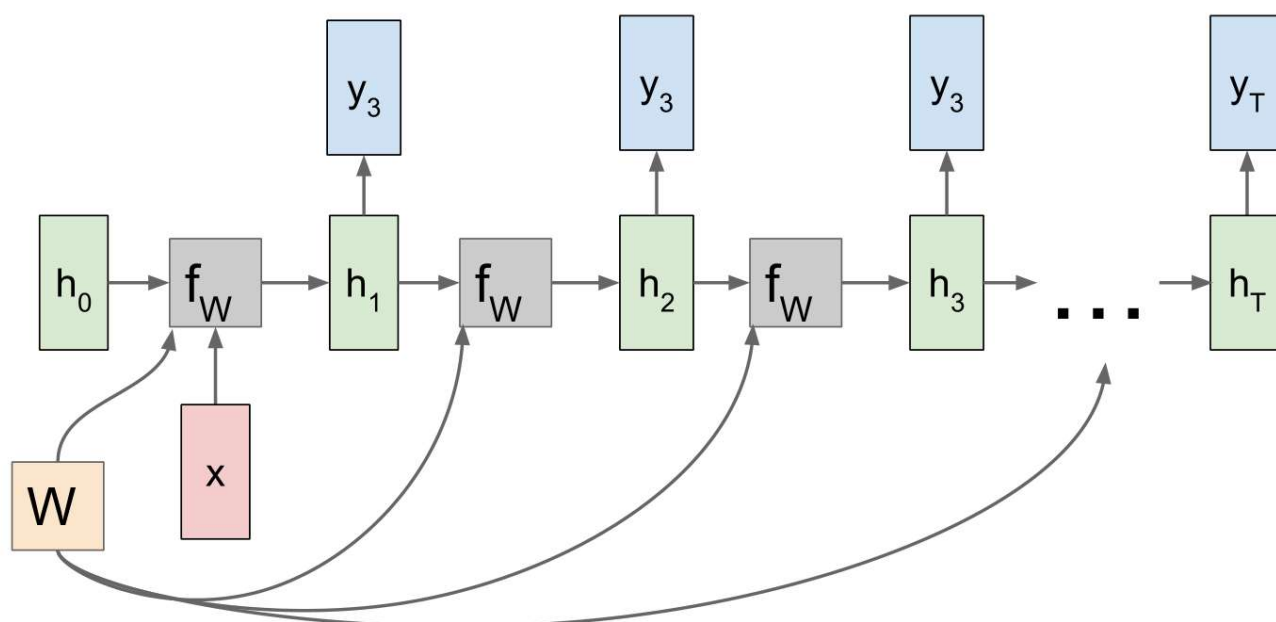
RNN: Computational Graph: Many to Many



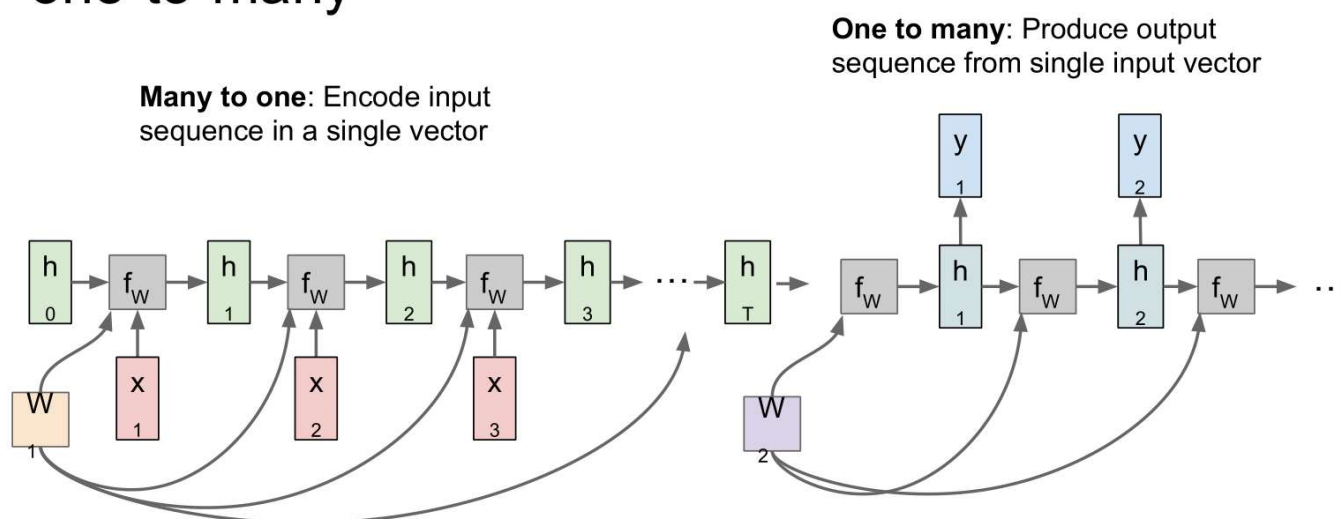
RNN: Computational Graph: Many to One



RNN: Computational Graph: One to Many



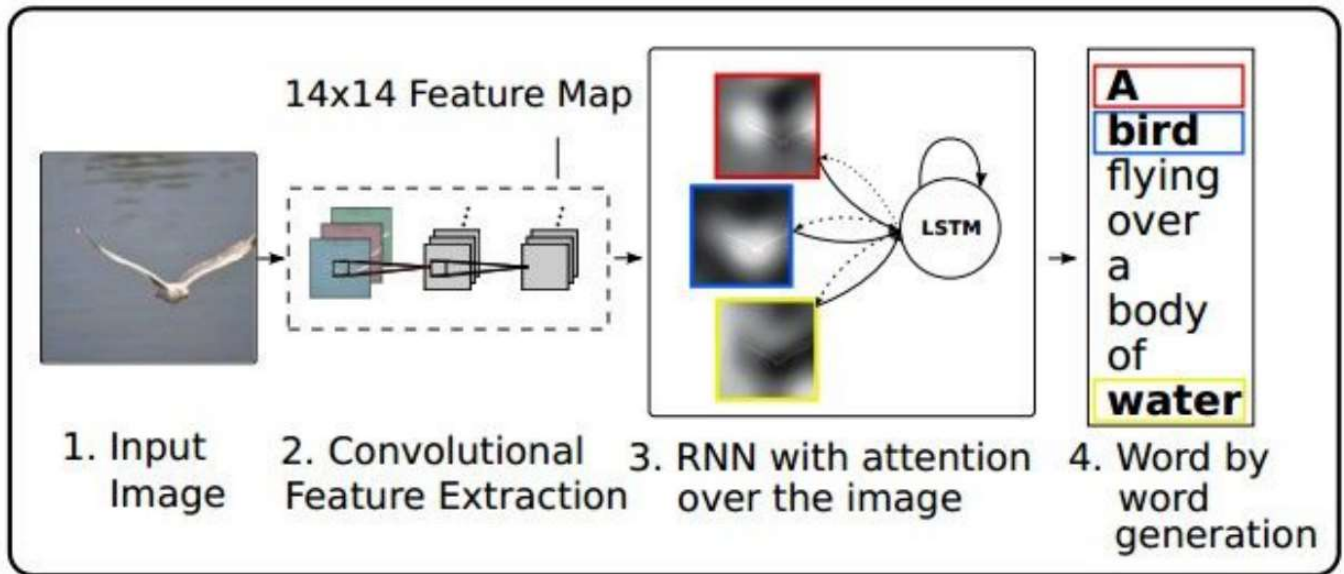
Sequence to Sequence: Many-to-one + one-to-many



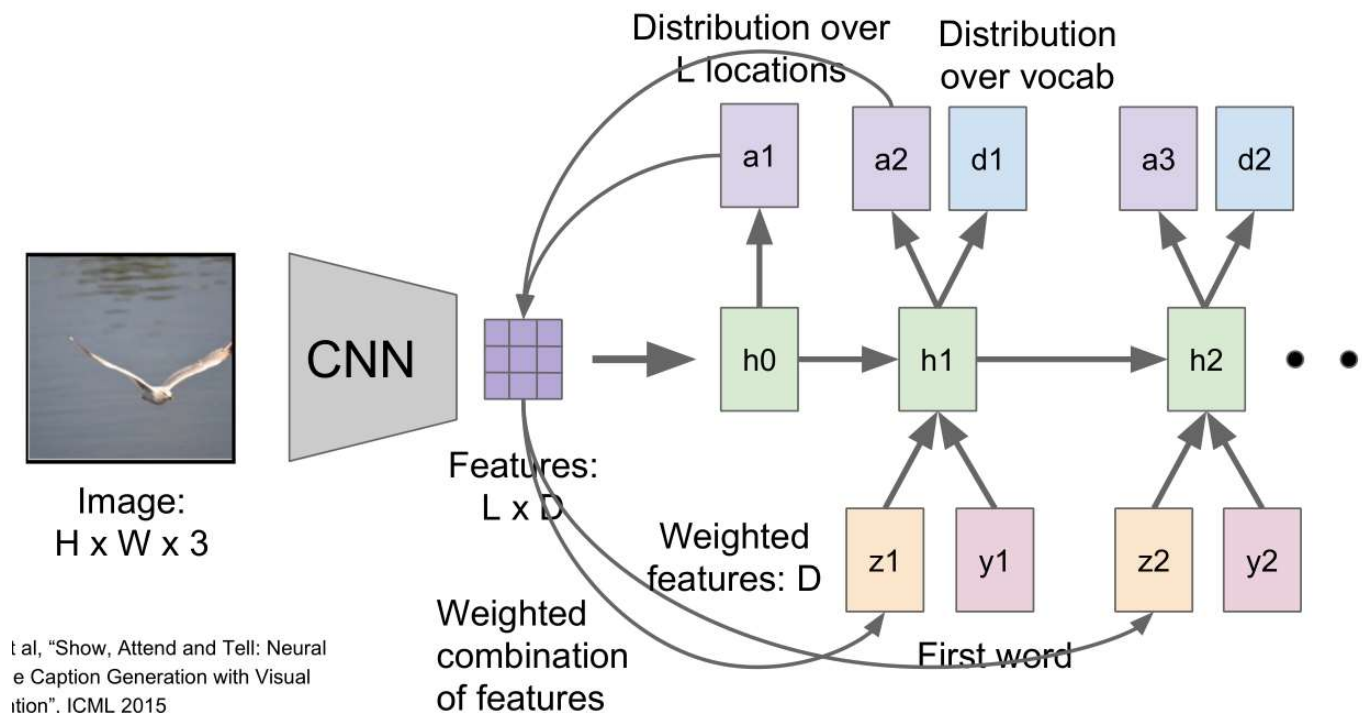
三、 Attention

这里的attention主要指的是图像信息捕捉过程中的一种技术和图像分析方式。

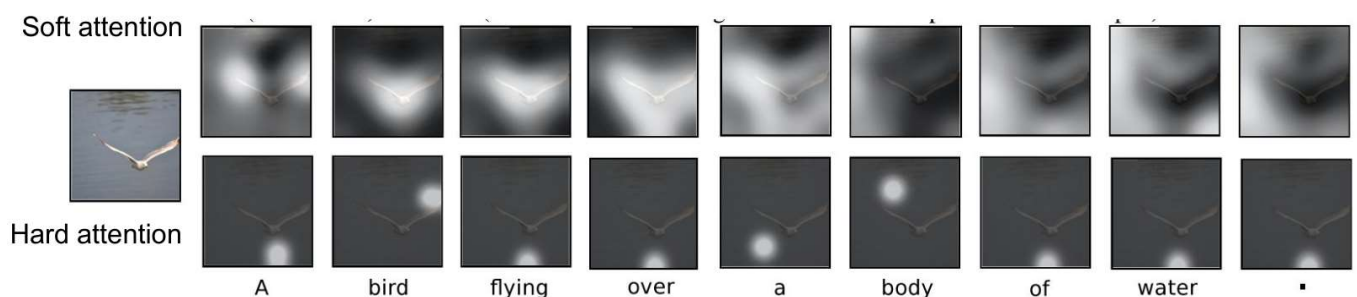
RNN focuses its attention at a different spatial location when generating each word



图像信息捕捉，主要做的是输入图像，先由CNN做处理，得到处理好的图像信息，然后交由RNN处理，最终得到图像信息的文字描述。这是一种RNN与CNN结合的处理办法。

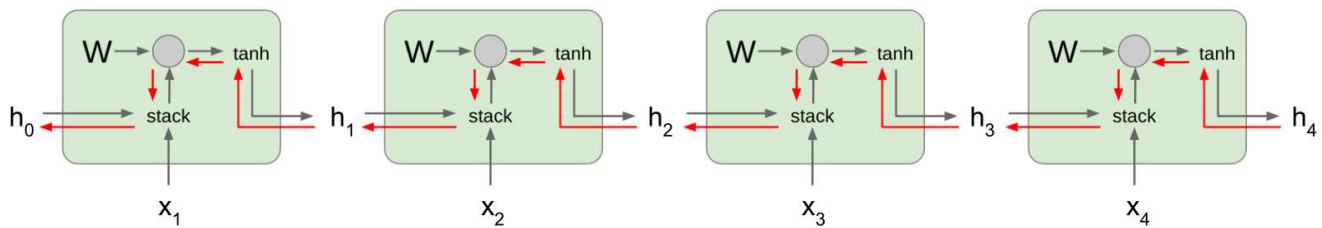


但是存在着一种新的处理办法就是attention办法，主要的目的就是让RNN每次处理的时候只专注于图片的一部分，这样可以使得描述更加准确。



四、RNN梯度流问题与LSTM

1.RNN梯度消失和梯度爆炸问题



Computing gradient of h_0 involves many factors of W (and repeated \tanh)

Largest singular value > 1 :
Exploding gradients

Largest singular value < 1 :
Vanishing gradients

Gradient clipping: Scale gradient if its norm is too big

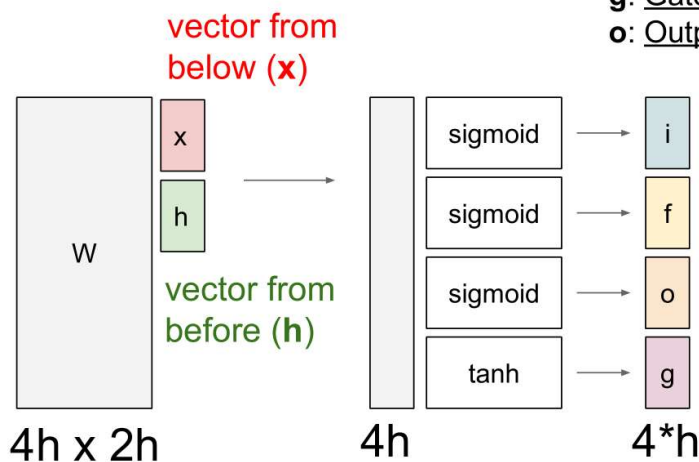
```
grad_norm = np.sum(grad * grad)
if grad_norm > threshold:
    grad *= (threshold / grad_norm)
```

2.梯度消失问题解决方法 —— LSTM

Long Short Term Memory (LSTM)

[Hochreiter et al., 1997]

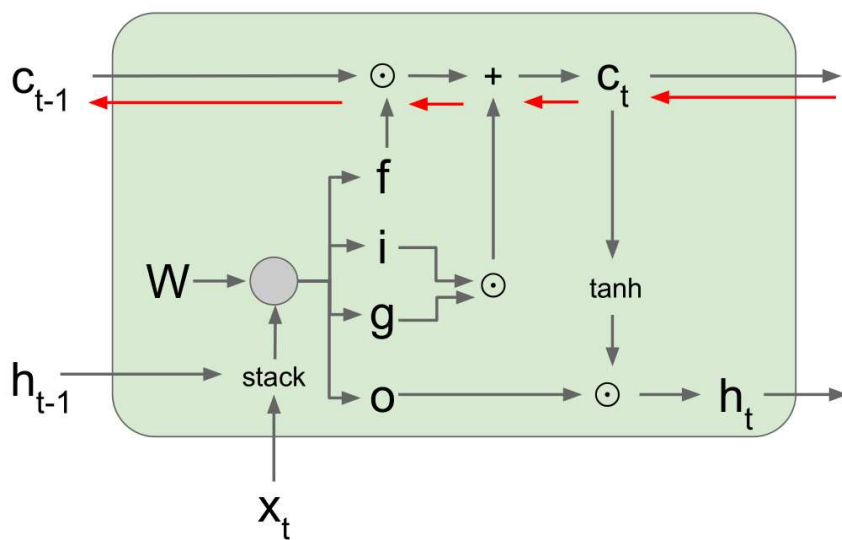
f: Forget gate, Whether to erase cell
i: Input gate, whether to write to cell
g: Gate gate (?), How much to write to cell
o: Output gate, How much to reveal cell



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

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



Backpropagation from c_t to c_{t-1} only elementwise multiplication by f , no matrix multiply by W

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

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

具体解释可以看这个博客: https://blog.csdn.net/jcsyl_mshot/article/details/80712110