

# RNN

## 学习长期依赖的困难：

- $\frac{\partial h_T}{\partial h_t} = \frac{\partial h_T}{\partial h_{T-1}} \frac{\partial h_{T-1}}{\partial h_{T-2}} \dots \frac{\partial h_{t+1}}{\partial h_t}$
- 以为RNN用的权重是一样的，所以很有可能造成梯度消失和梯度爆炸
- 长期依赖带来的信号很可能被隐藏，由于短期依赖的小小波动。
- 解决办法：
  - 结合short and long paths in unfolded flow graph
  - Leak units with self-connection:  
$$h_{t+1} = (1 - \frac{1}{\alpha})h_t + \frac{1}{\alpha} \tanh(W_{xh}x_t + W_{hh}h_t + b_n)$$
$$h_{t+1}$$
是 $h_t$ 的线性和非线性结合。
  - 反向传播误差更容易计算，因为有部分是线性的
  - $\alpha$ 控制遗忘的速率，也可以看作以往模型的滑动变化
  - $\alpha$ 可以是固定的也可以是学习的。

## LSTM:

- forgetting rate 希望随着时间的变化是不一样的，根据以前的隐藏层和现在的输入（根据目前的上下文）
  - $f_t = \sigma(W_{xf}X_t + W_{hf}h_{t-1} + b_f)$
  - $i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + b_i)$
  - $o_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + b_o)$
  - $g_t = \tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c)$
  - $c_t = f_t \cdot c_{t-1} + i_t \cdot g_t$
  - $h_t = o_t \cdot \tanh(c_t)$
  - $z_t = \text{softmax}(W_{hz}h_t + b_z)$