Titile: LSTM 中文
author: Ian
email: stmayue@gmail.com

因为最近接触得比较多，所以一直想尝试一下 LSTM 的基本推导，从 2000 年《Learning to forget-Continual prediction with LSTM》这篇文章的模型入手，现在已经有了新的模型了，就是将 cell 的输出也连接到各个 gate 作为输入 (peephole connection)，如果后面有兴趣的话会完成这部分。

LSTM 对于搞 NLP 的同学来说已经不陌生了，所以这里假设都对该模型有基本的了解，下面会给出必要的公式以及简要的前向过程，主要是对原文公式的详细推导和总结。

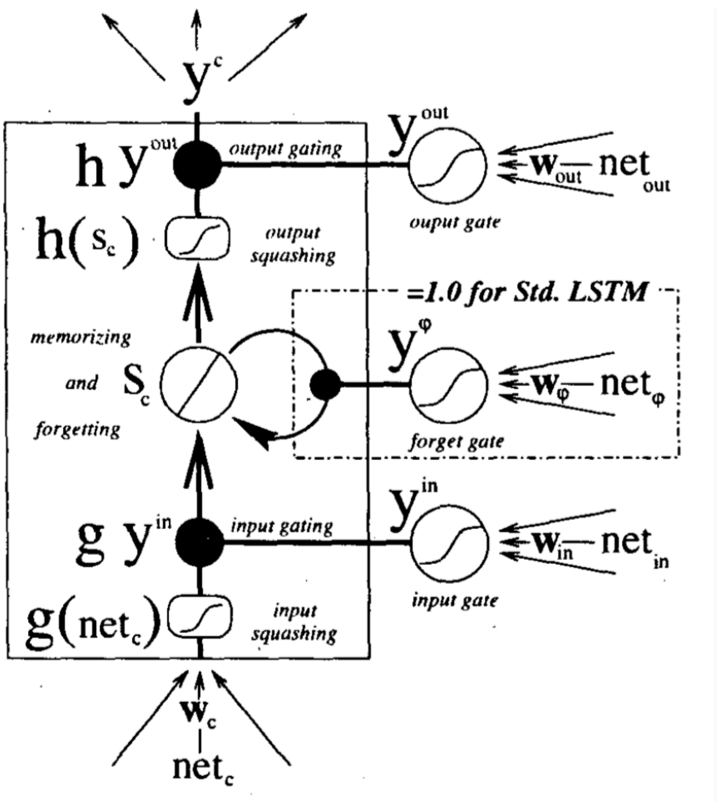

图 1: 基本的 LSTM

图一给出了 LSTM 的基本模型，对于该模型，有以下几个计算过程. 对

于一个 memory cell 从底部向上的计算过程为：首先是计算各个子部分，包括 net_c, output gate, input gate, forget gate

$$net_{c_j^v}(t) = \sum_m \omega_{c_j^v m}\, y_m(t-1) \tag{1}$$

$$y^{out_j}(t) = f_{out_j}(\sum_m \omega_{out_j m}\, y_m(t-1)) \tag{2}$$

$$y^{in_j}(t) = f_{in_j}(\sum_m \omega_{in_j m}\, y_m(t-1)) \tag{3}$$

$$y^{forget_j}(t) = f_{forget_j}(\sum_m \omega_{forget_j m}\, y_m(t-1)) \tag{4}$$

cell 部分的计算为：

$$S_{C_j^v}(t) = y^{forget_j}(t)\, S_{c_j^v} + y^{in_j}(t)\, g(net_{c_j^v}(t)) \tag{5}$$

输出为：

$$y^{c_j^v}(t) = y^{out_j}(t)\, h(S_{c_j^v}(t)) \tag{6}$$

上述公式中 y 为每个节点的输出，j 为第 j 个 memory cell block，v 为第 v 个 memory cell，一般我们是一个 block，memory cell 产生的是标量，组合起来 block 产生的就是向量了。

公式 (1) – (4) 完成了一个 memory cell 的前向过程，为了使后向过程更方便计算，建立一个完整的模型，这里假设 LSTM 是一个 hidden layer，上层接一个 output layer，计算如下：

$$y^k(t) = f_k(\sum_v \omega_{kc_j^v}\, y^{c_j^v}(t)) \tag{7}$$

至此，前向过程已经过了一遍，对应的是利用上面的前向步骤来推导参数的误差，即学习过程。下面为反向的推导。

这里使用 $\frac{1}{2}$ 平方误差来计算，有：

$$E(t) = \frac{1}{2}\sum_k e_k(t)^2 \tag{8}$$

$$e_k(t) = t^k(t) - y^k(t) \tag{9}$$

和 y^k 最相关的是 $\omega_{kc_j^v}$，那么它的权值更新可以通过下面公式计算而来：

$$\delta_k(t) = f'_k(net_k(t))\, e_k(t) \tag{10}$$

$$net_k(t) = \sum_v \omega_{kc_j^v}\, y^{c_j^v}(t) \tag{11}$$

$$\Delta\omega_{kc_j^v} = \alpha\, \delta_k(t)\, y_{c_j^v}(t) \tag{12}$$

接下来，可以看到 $y^{c_j^v}(t)$ 直接和 $y^{out_j}(t)$ 相关，可以先求 output gate 的误差，通过一系列展开式来说明：

$$y^k(t) = f_k(\sum_v \omega_{kc_j^v}\, y^{c_j^v}(t)) \tag{13}$$

$$= f_k(\sum_v \omega_{kc_j^v}\, y^{out_j}(t)\, h(S_{c_j^v}(t))) \tag{14}$$

通过链式法则分步来求其偏导数，可得

$$\frac{\partial E(t)}{\partial y^k(t)} = e_k(t) \tag{15}$$

$$\frac{\partial y^k(t)}{\partial y^{out_j} t} = f'_k(net_k(t)) \sum_v \omega_{kc_j^v}\, h(S_{c_j^v}(t)) \tag{16}$$

$$\frac{\partial y^{out_j}(t)}{\partial \omega_{out_j}} = f'_{out_j}(net_{out_j}(t))\, y^m(t-1) \tag{17}$$

$$\tag{18}$$

连乘并考虑 output layer 的 k 个输出可得

$$\delta_{out_j}(t) = \sum_k e_k(t)\, f'_k(net_k(t)) \sum_v \omega_{kc_j^v}\, h(S_{c_j^v}(t))\, f'_{out_j}(net_{out_j}(t)) \tag{19}$$

$$= f'_{out_j}(net_{out_j}(t))\, (\sum_v h(S_{c_j^v}(t)) \sum_k \omega k c_j^v\, \delta_k(t)) \tag{20}$$

$$\Delta\omega_{out_j} = \alpha\, \delta_{out_j}\, y^m(t-1) \tag{21}$$

接下来求 input gate, forget gate 以及最底层输入 $(\omega_{c_j^v})$ 计算的误差，可以看出，这几个参数只和 $S_{c_j^v}$ 有关，在这之前的误差传播对于这三个都是一样的，我们使用 $e_{s_{c_j^v}}$ 来统一表示，从上面公式（6）以及（20）可以得到

$$e_{s_{c_j^v}} = y^{out_j}(t)\, h'(S_{c_j^v}(t))\, (\sum_k \omega_{kc_j^v}\, \delta_k(t)) \tag{22}$$

重新看一下公式（5），为

$$S_{C_j^v}(t) = y^{forget_j}(t)\, S_{c_j^v} + y^{in_j}(t)\, g(net_{c_j^v}(t))$$

根据上面的式子来求导数，那么对于 input gate，有

$$\frac{\partial S_{c_j^v}(t)}{\partial \omega_{in_j}} = \frac{\partial S_{c_j^v}(t-1)}{\partial \omega_{in_j}}\, y^{forget_j}(t) + g(net_{c_j^v}(t))\, f'{}_{in_j}(net_{in_j}(t))\, y^m(t-1) \tag{23}$$

对于 forget gate，公式（5）只有加号前面的一下和其有关，按照分部积分原则，有

$$\frac{\partial S_{c_j^v}(t)}{\partial \omega_{forget_j}} = \frac{\partial S_{c_j^v}(t-1)}{\partial \omega_{forget_j}}\, y^{forget_j}(t) + S_{c_j^v}(t-1)\, f'{}_{forget_j}(net_{forget_j}(t))\, y^m(t-1) \tag{24}$$

对于 $(\omega_{c_j^v})$，有

$$\frac{\partial S_{c_j^v}(t)}{\partial \omega_{c_j^v}} = \frac{\partial S_{c_j^v}(t-1)}{\partial \omega_{c_j^v}}\, y^{forget_j}(t) + g'(net_{c_j^v}(t))\, y^{in_j}(t)\, y^m(t-1) \tag{25}$$

上述公式中，涉及 $S_{c_j^v}(t-1)$ 的偏导数都能在上一时刻的计算中求得，并且在 $t = 0$ 时刻均初始化为 0。此外涉及 g, f 的求导的部分，依据所选用的激活函数来计算，假设使用了 sigmoid 的函数，那么有

$$f'(x) = x\,(1-x) \tag{26}$$

基本的 LSTM 推导完成。