LSTM 的反向传播推导：
摘自 CSDN 博客
链接：https://www.cnblogs.com/pinard/p/6519110.html

有了LSTM前向传播算法，推导反向传播算法就很容易了， 思路和RNN的反向传播算法思路一致，也是通过梯度下降法迭代更新我们所有的参数，关键点在于计算所有参数基于损失函数的偏导数。

在RNN中，为了反向传播误差，我们通过隐藏状态$h^{(t)}$的梯度$\delta^{(t)}$一步步向前传播。在LSTM这里也类似。只不过我们这里有两个隐藏状态$h^{(t)}$和$C^{(t)}$。这里我们定义两个δ，即：

$$\delta_h^{(t)} = \frac{\partial L}{\partial h^{(t)}}$$

$$\delta_C^{(t)} = \frac{\partial L}{\partial C^{(t)}}$$

为了便于推导，我们将损失函数$L(t)$分成两块，一块是时刻t位置的损失$l(t)$，另一块是时刻t之后损失$L(t+1)$，即：

$$L(t) = \begin{cases} l(t) + L(t+1) & \text{if}\, t < \tau \\ l(t) & \text{if}\, t = \tau \end{cases}$$

而在最后的序列索引位置τ的$\delta_h^{(\tau)}$和 $\delta_C^{(\tau)}$为：

$$\delta_h^{(\tau)} = (\frac{\partial O^{(\tau)}}{\partial h^{(\tau)}})^T \frac{\partial L^{(\tau)}}{\partial O^{(\tau)}} = V^T(\hat{y}^{(\tau)} - y^{(\tau)})$$

$$\delta_C^{(\tau)} = (\frac{\partial h^{(\tau)}}{\partial C^{(\tau)}})^T \frac{\partial L^{(\tau)}}{\partial h^{(\tau)}} = \delta_h^{(\tau)} \odot o^{(\tau)} \odot (1 - tanh^2(C^{(\tau)}))$$

接着我们由$\delta_C^{(t+1)}, \delta_h^{(t+1)}$反向推导$\delta_h^{(t)}, \delta_C^{(t)}$。

$\delta_h^{(t)}$的梯度由本层t时刻的输出梯度误差和大于t时刻的误差两部分决定，即：

$$\delta_h^{(t)} = \frac{\partial L}{\partial h^{(t)}} = \frac{\partial l(t)}{\partial h^{(t)}} + (\frac{\partial h^{(t+1)}}{\partial h^{(t)}})^T \frac{\partial L(t+1)}{\partial h^{(t+1)}} = V^T(\hat{y}^{(t)} - y^{(t)}) + (\frac{\partial h^{(t+1)}}{\partial h^{(t)}})^T \delta_h^{(t+1)}$$

整个LSTM反向传播的难点就在于$\frac{\partial h^{(t+1)}}{\partial h^{(t)}}$这部分的计算。仔细观察，由于$h^{(t)} = o^{(t)} \odot tanh(C^{(t)})$，在第一项$o^{(t)}$中，包含一个$h$的递推关系，第二项$tanh(C^{(t)})$就复杂了，$tanh$函数里面又可以表示成：

$$C^{(t)} = C^{(t-1)} \odot f^{(t)} + i^{(t)} \odot a^{(t)}$$

$tanh$函数的第一项中，$f^{(t)}$包含一个h的递推关系，在$tanh$函数的第二项中，$i^{(t)}$和$a^{(t)}$都包含h的递推关系，因此，最终$\frac{\partial h^{(t+1)}}{\partial h^{(t)}}$这部分的计算结果由四部分组成。即：

$$\Delta C = o^{(t+1)} \odot [1 - tanh^2(C^{(t+1)})]$$

$$\frac{\partial h^{(t+1)}}{\partial h^{(t)}} = diag[o^{(t+1)} \odot (1 - o^{(t+1)}) \odot tanh(C^{(t+1)})]W_o + diag[\Delta C \odot f^{(t+1)} \odot (1 - f^{(t+1)}) \odot C^{(t)}]W_f \\ + diag\{\Delta C \odot i^{(t+1)} \odot [1 - (a^{(t+1)})^2]\}W_a + diag[\Delta C \odot a^{(t+1)} \odot i^{(t+1)} \odot (1 - i^{(t+1)})]W_i$$

而$\delta_C^{(t)}$的反向梯度误差由前一层$\delta_C^{(t+1)}$的梯度误差和本层的从$h^{(t)}$传回来的梯度误差两部分组成，即：

$$\delta_C^{(t)} = (\frac{\partial C^{(t+1)}}{\partial C^{(t)}})^T \frac{\partial L}{\partial C^{(t+1)}} + (\frac{\partial h^{(t)}}{\partial C^{(t)}})^T \frac{\partial L}{\partial h^{(t)}} = (\frac{\partial C^{(t+1)}}{\partial C^{(t)}})^T \delta_C^{(t+1)} + \delta_h^{(t)} \odot o^{(t)} \odot (1 - tanh^2(C^{(t)})) \\ = \delta_C^{(t+1)} \odot f^{(t+1)} + \delta_h^{(t)} \odot o^{(t)} \odot (1 - tanh^2(C^{(t)}))$$

有了$\delta_h^{(t)}$和$\delta_C^{(t)}$， 计算这一大堆参数的梯度就很容易了，这里只给出W_f的梯度计算过程，其他的$U_f, b_f, W_a, U_a, b_a, W_i, U_i, b_i, W_o, U_o, b_o$，$V, c$的梯度大家只要照搬就可以了。

$$\frac{\partial L}{\partial W_f} = \sum_{t=1}^{\tau} [\delta_C^{(t)} \odot C^{(t-1)} \odot f^{(t)} \odot (1 - f^{(t)})](h^{(t-1)})^T$$