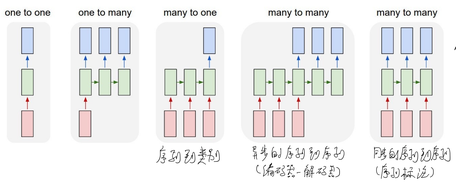
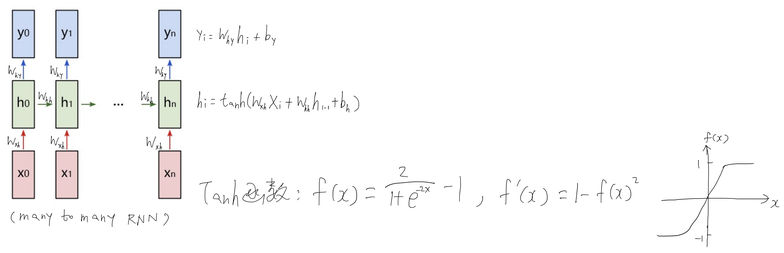
rnn

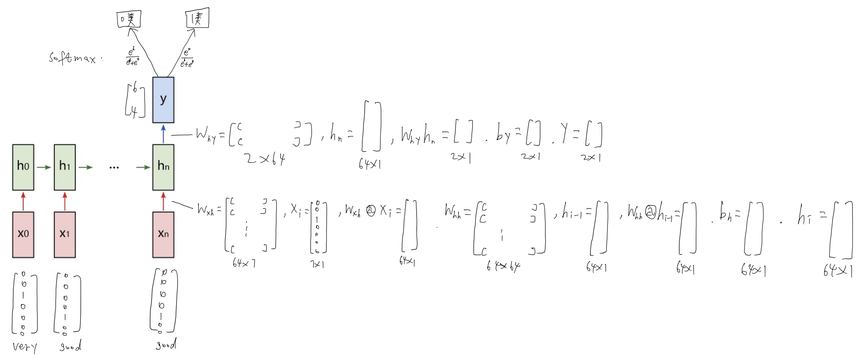
2020年8月23日 星期日　　下午10:58



简单循环网络SRN，只有1个隐藏层 h



$\tanh$函数：$f(x)=\frac{2}{1+e^{-2x}}-1$，$f'(x)=1-f(x)^2$



计算梯度的方式 < 随时间反向传播 BPTT
实时循环学习 RTRL：通过前向传播来计算梯度

长程依赖问题 < 梯度消失
梯度爆炸：权重衰减、梯度截断

$L=-\ln(p_c)=-\ln(\text{softmax}(y_c))$

$\frac{\partial L}{\partial p_c}=-\frac{1}{p_c}$

$\frac{\partial p_c}{\partial y_c}=\frac{e^{y_c}}{\sum e^{y_i}}-\frac{e^{y_c}e^{y_c}}{(\sum e^{y_i})^2}=\frac{e^{y_c}(\sum e^{y_i}-e^{y_c})}{(\sum e^{y_i})^2}=p_c\cdot\frac{\sum e^{y_i}-e^{y_c}}{\sum e^{y_i}}=p_c(1-p_c)$　　$(p_c=\frac{e^{y_c}}{\sum e^{y_i}})$

$\frac{\partial L}{\partial y_c}=\frac{\partial L}{\partial p_c}\cdot\frac{\partial p_c}{\partial y_c}=p_c-1$

$\frac{\partial p_c}{\partial y_i}=-\frac{e^{y_c}e^{y_i}}{(\sum e^{y_i})^2}=-p_c\cdot p_i$

$\frac{\partial L}{\partial y_i}=\frac{\partial L}{\partial p_c}\frac{\partial p_c}{\partial y_i}=p_i$

$\therefore \frac{\partial L}{\partial y_i}=\begin{cases}p_i, & i\neq c\\ p_i-1, & i=c\end{cases}$　　$\frac{\partial L}{\partial y}=\begin{bmatrix}p_1\\ p_2\\ \vdots\\ p_{c-1}\\ p_c-1\\ p_{c+1}\\ \vdots\\ p_n\end{bmatrix}=\begin{bmatrix}\frac{\partial L}{\partial y_1}\\ \frac{\partial L}{\partial y_2}\\ \vdots\\ \frac{\partial L}{\partial y_n}\end{bmatrix}=\begin{bmatrix}p_1\\ p_2\\ \vdots\\ p_n\end{bmatrix}-\begin{bmatrix}0\\ 0\\ \vdots\\ 1\\ \vdots\\ 0\end{bmatrix}$

$y=W_{hy}h+b_y=\begin{bmatrix}y_1\\ y_2\end{bmatrix}$　　$y_1=W_{hy_1}h+b_{y_1}$，$y_2=W_{hy_2}h+b_{y_2}$

$\frac{\partial y}{\partial W_{hy}}=\begin{bmatrix}\frac{\partial y_1}{\partial W_{hy_1}}\\ \frac{\partial y_2}{\partial W_{hy_2}}\end{bmatrix}=\begin{bmatrix}c & h & \end{bmatrix}$

$\frac{\partial L}{\partial W_{hy}}=\frac{\partial L}{\partial y}\frac{\partial y}{\partial W_{hy}}=\begin{bmatrix}\frac{\partial L}{\partial W_{hy_1}}\\ \frac{\partial L}{\partial W_{hy_2}}\end{bmatrix}=\begin{bmatrix}p_1-1 & h & \end{bmatrix}=\frac{\partial L}{\partial y}$ 乘 $\frac{\partial y}{\partial W_{hy}}=\begin{bmatrix}p_1-1\\ p_2\end{bmatrix}$ @ $\begin{bmatrix}h & \end{bmatrix}$

$\frac{\partial y}{\partial b_y}=1$

$\frac{\partial L}{\partial b_y}=\frac{\partial L}{\partial y}\frac{\partial y}{\partial b_y}=\begin{bmatrix}p_1\\ p_2-1\end{bmatrix}$

$\frac{\partial y}{\partial h}=W_{hy}=\begin{bmatrix}\frac{\partial y_1}{\partial h}\\ \frac{\partial y_2}{\partial h}\end{bmatrix}=\begin{bmatrix}c & \\ c & \end{bmatrix}=\frac{\partial y}{\partial h_n}$　　2×64

$h_i=\tanh(W_{xh}X_i+W_{hh}h_{i-1}+b_h)$

$h_n=\tanh(W_{xh}X_n+W_{hh}h_{n-1}+b_h)$，$h_{ni}=\tanh(W_{xh_i}X_n+W_{hh_i}h_{n-1}+b_{h_i})=\begin{bmatrix}\\ \\ \\ \end{bmatrix}$　　64×1

$\frac{\partial h_n}{\partial b_h}=1-h_n^2=\begin{bmatrix}\frac{\partial h_{n_1}}{\partial b_{h_1}}\\ \frac{\partial h_{n_2}}{\partial b_{h_2}}\\ \vdots\end{bmatrix}=\begin{bmatrix}1-h_{n_1}^2\\ 1-h_{n_2}^2\\ \vdots\end{bmatrix}$　　64×1

$\frac{\partial L}{\partial b_h}=\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_n}\frac{\partial h_n}{\partial b_h}=p_i\hat{\delta}(i\neq c-1)\begin{bmatrix}W_{hy_1}\\ W_{hy_2}\end{bmatrix}\cdot\begin{bmatrix}1-h_{n_1}^2\\ 1-h_{n_2}^2\\ \vdots\end{bmatrix}=\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_n}\frac{\partial h_n}{\partial b_h}=\frac{1}{2}(p_i\cdot\begin{bmatrix}W_{hy_1}\\ W_{hy_2}\end{bmatrix}\cdot\begin{bmatrix}1-h_{n_1}^2\\ 1-h_{n_2}^2\\ \vdots\end{bmatrix}+(p_i-1)\cdot\begin{bmatrix}W_{hy_1}\\ W_{hy_2}\end{bmatrix}\cdot\begin{bmatrix}1-h_{n_1}^2\\ 1-h_{n_2}^2\\ \vdots\end{bmatrix})=\frac{1}{2}\left((p_i\ \ p_i-1\ ]\ @\begin{bmatrix}W_{hy_1}\\ W_{hy_2}\end{bmatrix}\right)\cdot\begin{bmatrix}1-h_{n_1}^2\\ 1-h_{n_2}^2\\ \vdots\end{bmatrix}$　　64×1

$=\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_j}\frac{\partial h_j}{\partial b_h}=\frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_j}\frac{\partial h_j}{\partial b_h}\right)$

$\frac{\partial h_n}{\partial W_{xh}}=(1-h_n^2)X_n=\begin{bmatrix}\frac{\partial h_{n_1}}{\partial W_{xh}}X_n\\ (1-h_{n_2}^2)X_n\\ \vdots\end{bmatrix}=\begin{bmatrix}c & \\ c & \\ \vdots & \end{bmatrix}$　　64×1

$\frac{\partial L}{\partial W_{xh}}=\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_n}\frac{\partial h_n}{\partial W_{xh}}=\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_n}\frac{\partial h_n}{\partial W_{xh}}=\frac{1}{2}\left((p_i\ \ p_i-1\ ]\ @\begin{bmatrix}W_{hy_1}\\ W_{hy_2}\end{bmatrix}\right)\ @\begin{bmatrix}(1-h_{n_1}^2)X_n\\ (1-h_{n_2}^2)X_n\\ \vdots\end{bmatrix}$　　64×1

$=\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_j}\frac{\partial h_j}{\partial W_{xh}}=\frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_j}\frac{\partial h_j}{\partial W_{xh}}\right)$

$\frac{\partial h_n}{\partial W_{hh}}=(1-h_n^2)h_{n-1}=\begin{bmatrix}\frac{\partial h_{n_1}}{\partial W_{hh}}\\ \frac{\partial h_{n_2}}{\partial W_{hh}}\\ \vdots\end{bmatrix}=\begin{bmatrix}(1-h_{n_1}^2)h_{n-1}\\ (1-h_{n_2}^2)h_{n-1}\\ \vdots\end{bmatrix}=\begin{bmatrix}c & \\ c & \\ \vdots & \end{bmatrix}$　　64×1

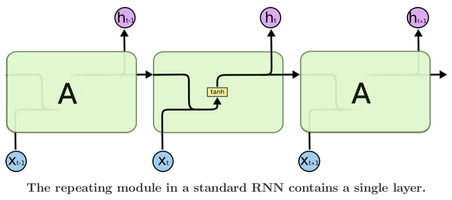
$\frac{\partial L}{\partial W_{hh}}=\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_n}\frac{\partial h_n}{\partial W_{hh}}=\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_n}\frac{\partial h_n}{\partial W_{hh}}=\frac{1}{2}\left((p_i\ \ p_i-1\ ]\ @\begin{bmatrix}W_{hy_1}\\ W_{hy_2}\end{bmatrix}\right)\ @\begin{bmatrix}(1-h_{n_1}^2)h_{n-1}\\ (1-h_{n_2}^2)h_{n-1}\\ \vdots\end{bmatrix}$　　64×1

$=\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_j}\frac{\partial h_j}{\partial W_{hh}}=\frac{1}{N}\sum_{j=1}^{N}\left(\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_j}\frac{\partial h_j}{\partial W_{hh}}\right)$

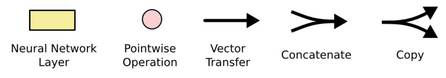
$\frac{\partial L}{\partial h_{n-1}}=\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial h_n}\frac{\partial h_n}{\partial h_{n-1}}=\frac{\partial L}{\partial h_n}(1-h_n^2)W_{hh}$

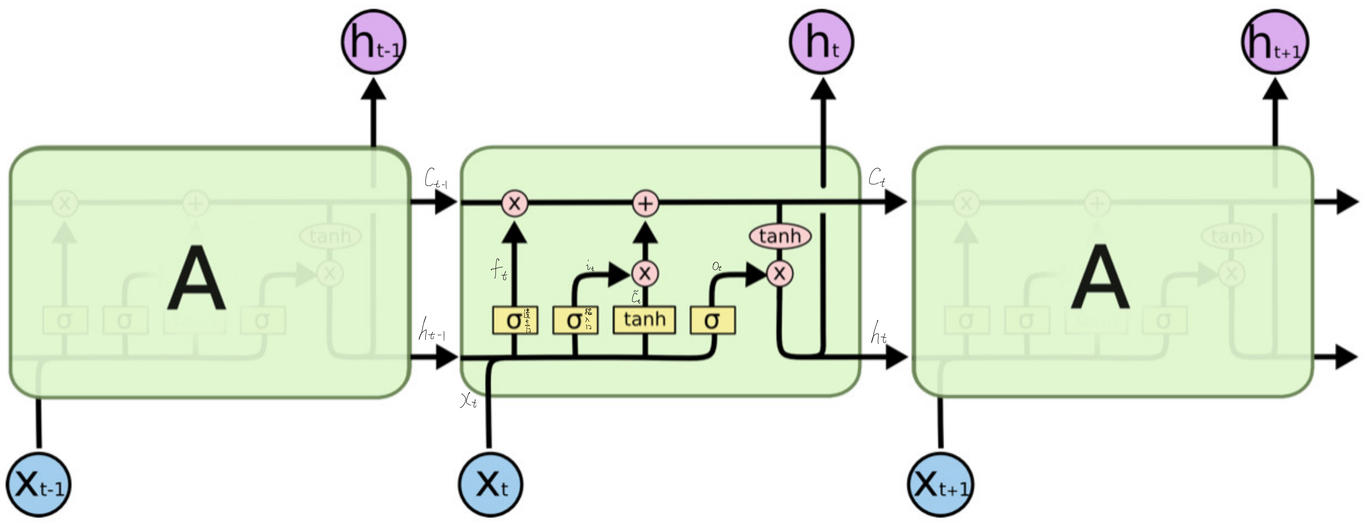
随时间反向传播算法 BPTT
Backpropagation Through Time

基于门控的循环神经网络 < 长短期记忆网络 LSTM，Long Short-Term Memory Network
(Gated RNN)　　门控循环网络 GRU，Gated Recurrent Unit



The repeating module in a standard RNN contains a single layer.





The repeating module in an LSTM contains four interacting layers.

$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f)=\sigma(\begin{bmatrix}c & W_f & \end{bmatrix}\begin{bmatrix}h_{t-1}\\ x_t\end{bmatrix}+[b_f])=\sigma(\lambda_t)=\frac{1}{1+e^{-\lambda_t}}$　　(遗忘门)

$i_t=\sigma(W_i\cdot[h_{t-1},x_t]+b_i)=\sigma(\begin{bmatrix}c & W_i & \end{bmatrix}\begin{bmatrix}h_{t-1}\\ x_t\end{bmatrix}+[b_i])=\sigma(\lambda_i)=\frac{1}{1+e^{-\lambda_i}}$　　(输入门)

$\tilde{C}_t=\tanh(W_{\tilde{c}}\cdot[h_{t-1},x_t]+b_{\tilde{c}})=\tanh(\begin{bmatrix}c & W_{\tilde{c}} & \end{bmatrix}\begin{bmatrix}h_{t-1}\\ x_t\end{bmatrix}+[b_{\tilde{c}}])=\tanh(\lambda_{\tilde{c}})=\frac{2}{1+e^{-2\lambda_{\tilde{c}}}}-1$

$C_t=f_t*C_{t-1}+i_t*\tilde{C}_t=[\ ]*[\ ]+[\ ]*[\ ]=[\ ]$

$O_t=\sigma(W_o[h_{t-1},x_t]+b_o)=\sigma(\begin{bmatrix}c & W_o & \end{bmatrix}\begin{bmatrix}h_{t-1}\\ x_t\end{bmatrix}+[b_o])=\sigma(\lambda_o)=\frac{1}{1+e^{-\lambda_o}}$　　(输出门)

$h_t=O_t*\tanh(C_t)$

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

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

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

门控机制：
（1）遗忘门 $\boldsymbol{f}_t$ 控制上一个时刻的内部状态 $\boldsymbol{c}_{t-1}$ 需要遗忘多少信息。
（2）输入门 $\boldsymbol{i}_t$ 控制当前时刻的候选状态 $\tilde{\boldsymbol{c}}_t$ 有多少信息需要保存。
（3）输出门 $\boldsymbol{o}_t$ 控制当前时刻的内部状态 $\boldsymbol{c}_t$ 有多少信息需要输出给外部状态 $\boldsymbol{h}_t$。

$C_t=f_t*C_{t-1}+i_t*\tilde{C}_t$

$h_t=O_t*\tanh(C_t)$

门取值在 (0, 1) 之间，表示以一定的比例 允许信息通过