RNN backpropogation

Input : $x \in \mathbb{R}^f$

Assuming the cell is d dimensional i.e. internal state of RNN is d dimensional $h_t \in \mathbb{R}^d$ and predicted output $\hat{y} \in \mathbb{R}^n$

At any time t, the equations in RNN are

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

$$y_t = W_{hy}h_t \tag{2}$$

Applicable for $t = 1, 2, , .s$ where s is sequence length.
h_1 computation requires h_0 which is set to 0 or some other pre defined distribution.
Thus, we have inputs for the entire sequence $x_1, x_2, ..., x_s$ and we could have corresponding output as just y_s or $(y_1, y_2, ..., y_s)$ as per single output or multiple outputs case.

Given one particular W_{xh} and W_{hh} (initialised randomly), we can start first round of forward propogation.

For backpropogation, we first need to define loss function between y_t and $\hat{y}_t$. Consider the example of softmax loss.

Step 1: Compute $\hat{y}_t$
Actual output $\hat{y}_t$ would be n dimensional but we'll show here computation for 3 dimensional output and then generalize to n dimensions. So, for our 3-d case,

$$\hat{y}_t = \begin{bmatrix} \hat{y}_{t1} \\ \hat{y}_{t2} \\ \hat{y}_{t3} \end{bmatrix} \tag{3}$$

$$loss = -\sum_{i=1}^{3} y_i log(P_i) \tag{4}$$

where $P_i = \dfrac{e^{\hat{y}_{ti}}}{\sum_{j=1}^{3} e^{\hat{y}_{tj}}}$

This exponential over sum of exponentials is equivalent to probability. Also, since y_t is one hot encoded output, only one of the values in y_t would be 1 and rest would be 0. So, the above loss equation reduces to

$$loss = -log(P_{correct\_class}) \tag{5}$$

Learning outcomes:
Author(s): Falak Shah

Also, note that since $y_t \in \mathbb{R}^n$ and loss is scalar, $\dfrac{\partial loss}{\partial y} \in \mathbb{R}^n$

For our 3d case, say the correct class is 2. In this case, we'll have

$$y_t = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{6}$$

and $loss = -log(P_2)$ where $P_2 = \dfrac{e^{\hat{y}_{t2}}}{e^{\hat{y}_{t1}} + e^{\hat{y}_{t2}} + e^{\hat{y}_{t3}}}$
After some computation, we have

$$\frac{\partial loss}{\partial \hat{y}_t} = \begin{bmatrix} \frac{\partial loss}{\partial \hat{y}_{1t}} \\ \frac{\partial loss}{\partial \hat{y}_{2t}} \\ \frac{\partial loss}{\partial \hat{y}_{3t}} \end{bmatrix} \tag{7}$$

$$\frac{\partial loss}{\partial \hat{y}_t} = \begin{bmatrix} P_1 \\ P_2 - 1 \\ P_3 \end{bmatrix} \tag{8}$$

In general, for an n dimensional predicted output $\hat{y}_t$ and where c is the correct class, we would have

$$\frac{\partial loss}{\partial \hat{y}_t} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_c - 1 \\ \vdots \\ P_n \end{bmatrix} \tag{9}$$

Let us denote the term $\dfrac{\partial loss}{\partial \hat{y}_t}$ as dy from now on. Thus from equation 2, we have

$$\begin{aligned} y_t &= W_{hy} h_t \\ \frac{\partial loss}{\partial W_{hy}} &= \frac{\partial loss}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial W_{hy}} \\ \frac{\partial loss}{\partial W_{hy}} &= dy \cdot h^T \\ dy[n \times 1] \cdot h^T[1 \times d] - - > [n \times d] W_{hh} \end{aligned} \tag{10}$$

Next, from equation 1, we have

$$
\begin{aligned}
h_t &= tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\
\frac{\partial loss}{\partial W_{hh}} &= \frac{\partial loss}{\partial h_t}\frac{\partial h_t}{\partial W_{hh}} \\
\frac{\partial loss}{\partial W_{xh}} &= \frac{\partial loss}{\partial h_t}\frac{\partial h_t}{\partial W_{xh}}
\end{aligned}
\tag{11}
$$

Denoting $\frac{\partial loss}{\partial h_t}$ by dh_t which we'll compute soon, we have for $\frac{\partial loss}{\partial W_{hh}}$

$$
\begin{aligned}
h_t &= tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\
\frac{\partial loss}{\partial W_{hh}} &= dh_t \frac{\partial h_t}{\partial W_{hh}} \\
\frac{\partial loss}{\partial W_{hh}} &= dh_t * (1 - tanh^2(W_{hh}h_{t-1} + W_{xh}x_t)) \cdot h_{t-1}^T \\
\frac{\partial loss}{\partial W_{hh}} &= dh_t * (1 - h_t^2) \cdot h_{t-1}^T
\end{aligned}
\tag{12}
$$

Similarly, for $\frac{\partial loss}{\partial W_{xh}}$

$$
\frac{\partial loss}{\partial W_{xh}} = dh_t * (1 - h_t^2) \cdot x_t^T \tag{13}
$$

Finally, let us compute $\frac{\partial loss}{\partial h}$ or dh. Since it impacts not just loss at time t, but also loss at time $t+1$ since it impacts h_{t+1} too. We have

$$
\frac{\partial loss}{\partial h_t} = \frac{\partial loss_t}{\partial h_t} + \frac{\partial loss_{t+1}}{\partial h_t} \tag{14}
$$

$$
\begin{aligned}
\frac{\partial loss}{\partial h_t} &= \frac{\partial loss_t}{\partial y_t}\frac{\partial y_t}{\partial h_t} + \frac{\partial loss_{t+1}}{\partial h_{t+1}}\frac{\partial h_{t+1}}{\partial h_t} \\
\frac{\partial loss}{\partial h_t} &= W_{hy}^T dy + W_{hh}^T(1 - h_{t+1}^2) * dh_{t+1}
\end{aligned}
\tag{15}
$$

Thus, we have

$$
dh_t = W_{hy}^T dy + W_{hh}^T(1 - h_{t+1}^2) * dh_{t+1} \tag{16}
$$

Replacing dh_t value from equation 16 into equations 13 and 12, we can compute the values for $\frac{\partial loss}{\partial W_{xh}}$ and $\frac{\partial loss}{\partial W_{hh}}$, thus completing the backpropogation step for a single time step.