Backpropagation through time of RNN

Hao Ni
University College London
The Alan Turing Institute

Loss function

For a task of predicting the sequential output, the loss function is often in the additive form as follows:

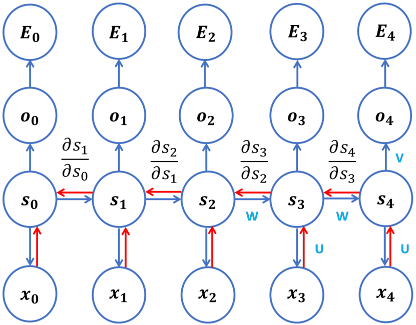
$$\text{Loss Function:} L(\bar{o}, \bar{y}) = \sum_{t=1}^{T} E(o_t, y_t),$$

where y_t and o_t are the actual/estimated output at time t respectively, $\bar{o} = (o_t)_{t=1}^{T}$ and $\bar{y} = (y_t)_{t=1}^{T}$.
For concreteness, we consider $E(o, y) := ||o - y||_2^2$ in the rest of this talk[1], which is commonly used for the regression problem. For ease of notation, we write $E_t := E(o_t, y_t)$.

[1] Let $x = (x^{(1)}, x^{(2)}, \cdots, x^{(d)}) \in \mathbb{R}^d$. $||x||_2^2 = \sum_{i=1}^{d} (x^{(i)})^2$.

Optimization of RNN

- Stochastic/Mini-batch Gradient Decent;
- Gradient calculation: Backpropagation through Time.



Goal: To compute $\frac{dE_t}{dV}, \frac{dE_t}{dU}, \frac{dE_t}{dW}$.

$$\frac{dE_t}{dV} = \frac{\partial E_t}{\partial o_t} \cdot \frac{\partial o_t}{\partial V}$$

$$\frac{dE_t}{dU} = \frac{\partial E_t}{\partial o_t} \cdot \frac{\partial o_t}{\partial U} = \frac{\partial E_t}{\partial o_t}\frac{\partial o_t}{\partial s_t} \cdot \frac{ds_t}{dU}$$

$$\frac{dE_t}{dW} = \frac{\partial E_t}{\partial o_t}\frac{\partial o_t}{\partial s_t} \cdot \frac{ds_t}{dW}$$

Derivative Computation

The computation of total derivatives boils down to

- $\frac{\partial E_t}{\partial o_t}$, $\frac{\partial o_t}{\partial s_t}$, $\frac{\partial o_t}{\partial V}$;
- $\frac{ds_t}{dW}$, $\frac{ds_t}{dU}$.

First let us explain the derivation of the partial derivatives $\frac{\partial E_t}{\partial o_t}$ and $\frac{\partial o_t}{\partial s_t}$. The computation of $\frac{\partial o_t}{\partial V}$ is similar and hence left for the homework.

$$\frac{\partial E_t}{\partial o_t} = \frac{\partial(||o_t - y_t||_2^2)}{\partial o_t} = 2(o_t - y_t).$$

Recall that $o_t = g(Vs_t)$ where $s_t \in \mathbb{R}^{n_1}$, $o_t \in \mathbb{R}^e$ and V is a matrix of size (e, n_1). $\frac{\partial o_t}{\partial s_t}$ is a matrix of size (e, n_1), i.e.

$$\frac{\partial o_t}{\partial s_t} = \left(\frac{\partial o_t^{(i)}}{\partial s_t^{(j)}} \right)_{i \in [e], j \in [n_1]}. \tag{1}$$

Lemma

If $g : \mathbb{R} \to \mathbb{R}$ is differentiable and $o_t = g(Vs_t)$, then it holds that $\forall i \in [e]$ and $j \in [n_1]$,

$$\frac{\partial o_t^{(i)}}{\partial s_t^{(j)}} = V_{ij} g' \left(\sum_{k=1}^{n_1} V_{ik} s_t^{(k)} \right). \tag{2}$$

Proof.

For any $\forall i \in [e]$,

$$o_t^{(i)} = (g(Vs_t))^{(i)} = g\left(\sum_{k=1}^{n_1} V_{ik} s_t^{(k)} \right).$$

Then by Chain rule it follows that for any $j \in [n_1]$,

$$\frac{\partial o_t^{(i)}}{\partial s_t^{(j)}} = V_{ij} g' \left(\sum_{k=1}^{n_1} V_{ik} s_t^{(k)} \right).$$

Derivation of $\frac{ds_t}{dU}$ and $\frac{ds_t}{dW}$

By the definition of s_t, the recurrence of $\frac{ds_t}{dU}$ and $\frac{ds_t}{dW}$ holds as follows:

$$s_t = h(Ux_t + Ws_{t-1}) \quad \Longrightarrow \quad \begin{aligned} \frac{ds_t}{dU} &= \frac{\partial s_t}{\partial U} + \frac{\partial s_t}{\partial s_{t-1}} \frac{ds_{t-1}}{dU} \\ \frac{ds_t}{dW} &= \frac{\partial s_t}{\partial W} + \frac{\partial s_t}{\partial s_{t-1}} \frac{ds_{t-1}}{dW}. \end{aligned}$$

Lemma (Recurrence Structure of $\frac{ds_t}{dU}$ and $\frac{ds_t}{dW}$)

For any $t \in \{1, 2, \cdots, T\}$,

$$\frac{ds_t}{dU} = \frac{\partial s_t}{\partial U} + \sum_{k=0}^{t-1} \left(\prod_{j=k+1}^{t} \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial U}, \tag{3}$$

$$\frac{ds_t}{dW} = \frac{\partial s_t}{\partial W} + \sum_{k=0}^{t-1} \left(\prod_{j=k+1}^{t} \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}. \tag{4}$$

Proof.

Since $s_t = h(Ux_t + Ws_{t-1})$ and s_{t-1} also depends on U, it follows:

$$\frac{ds_t}{dU} = \frac{\partial s_t}{\partial U} + \frac{\partial s_t}{\partial s_{t-1}} \frac{ds_{t-1}}{dU}.$$

Applying the above equation for the term $\frac{ds_{t-1}}{dU}$, it follows that

$$\begin{aligned} \frac{ds_t}{dU} &= \frac{\partial s_t}{\partial U} + \frac{\partial s_t}{\partial s_{t-1}} \frac{ds_{t-1}}{dU} \\ &= \frac{\partial s_t}{\partial U} + \frac{\partial s_t}{\partial s_{t-1}} \left(\frac{\partial s_{t-1}}{\partial U} + \frac{\partial s_{t-1}}{\partial s_{t-2}} \frac{ds_{t-2}}{dU} \right) \end{aligned}$$

Repeating this procedure until reaching $t = 0$, we have the formula Equation (3). The rigorous proof can be done as follows by induction. $\square$

Backpropagation through time

Algorithm (Compute $\frac{ds_T}{dU}$ and $\frac{ds_T}{dW}$)

1: Initialize $\frac{ds_T}{dU} \leftarrow \frac{\partial s_T}{\partial U}$;
2: Initialize $z \rightarrow \mathbf{Id}_{n_1 \times n_1}$.
3: **for** $t = T : -1 : 1$ **do**
4: $\quad z \leftarrow z \frac{\partial s_t}{\partial s_{t-1}}$
5: $\quad \frac{ds_T}{dU} \leftarrow \frac{ds_T}{dU} + z \frac{\partial s_t}{\partial U}$
6: $\quad \frac{ds_T}{dW} \leftarrow \frac{ds_T}{dW} + z \frac{\partial s_t}{\partial W}$
7: **end for**
8: Output $\frac{ds_T}{dU}$ and $\frac{ds_T}{dW}$.

Thanks for your attention!