Lab 9

Hüseyin Anil Gündüz

2022-07-05

In the first part of the lab, we will analytically derive the backpropagation equations for a simple RNN. Then, in the second part, we will implement forward and backward propagation functions for a simple RNN-model, and train to predict the future temperature based on past weather metrics.

Exercise 1

In this part, we derive the backpropagation equations for a simple RNN from forward propagation equations. For simplicity, we will focus on a single input sequence $\mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[\tau]}$. The forward pass in a RNN with hyperbolic tangent activation at time t is given by:

$$\mathbf{h}^{[t]} = \tanh\left(\mathbf{W}\mathbf{h}^{[t-1]} + \mathbf{U}\mathbf{x}^{[t]} + \mathbf{b}\right) \tag{1}$$

$$\mathbf{y}^{[t]} = \mathbf{V}\mathbf{h}^{[t]} + \mathbf{c} \tag{2}$$

where the parameters are the bias vectors $\mathbf{b}$ and $\mathbf{c}$ along with the weight matrices $\mathbf{U}$,$\mathbf{V}$ and $\mathbf{W}$, respectively, for input-to-hidden, hidden-to-output and hidden-to-hidden connections. As we will use RNN for a regression problem in the of the exercise, we do not use an activation function in order to compute the output $\mathbf{y}^{[t]}$ (at time t).

The loss is defined as:

$$\mathcal{L} = \sum_{t=1}^{\tau} \mathcal{L}\left(\mathbf{y}^{[t]}, \hat{\mathbf{y}}^{[t]}\right) \tag{3}$$

Show that:

$$\nabla_{\mathbf{h}^{[\tau]}}\mathcal{L} = \mathbf{V}^T(\nabla_{\mathbf{y}^{[\tau]}}\mathcal{L}) \tag{4}$$

$$\nabla_{\mathbf{h}^{[t]}}\mathcal{L} = \mathbf{W}^T\mathrm{diag}\left(1 - \left(\mathbf{h}^{[t+1]}\right)^2\right)(\nabla_{\mathbf{h}^{[t+1]}}L) + \mathbf{V}^T(\nabla_{\mathbf{y}^{[t]}}\mathcal{L}) \tag{5}$$

$$\nabla_{\mathbf{c}}\mathcal{L} = \sum_{t=1}^{\tau} \nabla_{\mathbf{y}^{[t]}}\mathcal{L} \tag{6}$$

$$\nabla_{\mathbf{b}}\mathcal{L} = \sum_{t=1}^{\tau} \mathrm{diag}\left(1 - \left(\mathbf{h}^{[t]}\right)^2\right)\nabla_{\mathbf{h}^{[t]}}\mathcal{L} \tag{7}$$

$$\nabla_{\mathbf{V}}\mathcal{L} = \sum_{t=1}^{\tau} (\nabla_{\mathbf{y}^{[t]}}\mathcal{L})\mathbf{h}^{[t]^T} \tag{8}$$

$$\nabla_{\mathbf{W}}\mathcal{L} = \sum_{t=1}^{\tau} \mathrm{diag}\left(1 - \left(\mathbf{h}^{[t]}\right)^2\right)\ (\nabla_{\mathbf{h}^{[t]}}\mathcal{L})\mathbf{h}^{[t-1]^T} \tag{9}$$

$$\nabla_{\mathbf{U}}\mathcal{L} = \sum_{t=1}^{\tau} \mathrm{diag}\left(1 - \left(\mathbf{h}^{[t]}\right)^2\right)(\nabla_{\mathbf{h}^{[t]}}\mathcal{L})\mathbf{x}^{[t]^T} \tag{10}$$

Hint 1 (chain rule for vector calculus): given a vector $\mathbf{x} \in \mathbb{R}^n$ and two functions $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^m \to \mathbb{R}$, call the outputs $\mathbf{y} = f(\mathbf{x})$ and $z = g(\mathbf{y}) = g(f(\mathbf{x}))$, then the following holds:

$$\nabla_{\mathbf{x}} z = \nabla_{\mathbf{x}}\mathbf{y} \cdot \nabla_{\mathbf{y}} z \tag{11}$$

where $\nabla_{\mathbf{y}} z \in \mathbb{R}^m$ and $\nabla_{\mathbf{x}}\mathbf{y} \in \mathbb{R}^n \times \mathbb{R}^m$.

Hint 2: draw a computational graph representing the computation performed by the RNN unrolled over time, then use this graph to compute the gradients: multiply gradients via the chain rule when traversing edges, and sum the gradients obtained along each path from the loss to the item you are differentiating against.

Solution

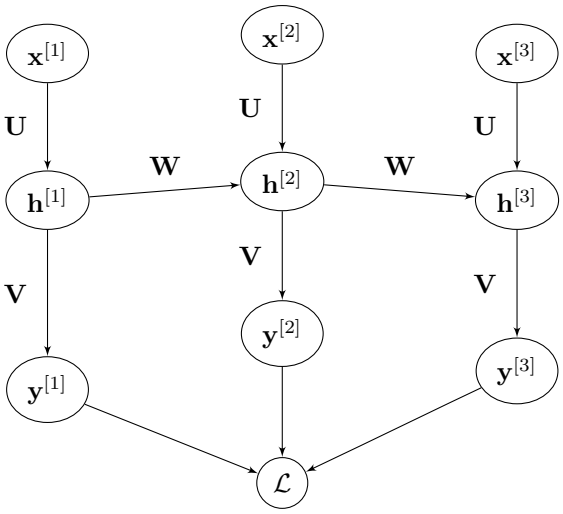

Figure 1: A simplified computational graph for three steps of a RNN. Biases omitted for simplicity.

The computational graph is shown in Figure 1. There is only one path connecting $\mathbf{h}^{[\tau]}$ to the loss:

$$\nabla_{\mathbf{h}^{[\tau]}}\mathcal{L} = \nabla_{\mathbf{h}^{[\tau]}}\mathbf{y}^{[\tau]} \cdot \nabla_{\mathbf{y}^{[\tau]}}\mathcal{L} = \mathbf{V}^T \cdot \nabla_{\mathbf{y}^{[\tau]}}\mathcal{L} \tag{12}$$

while every other hidden activation influences the loss via its associated output and the following hidden activation, thus:

$$\nabla_{\mathbf{h}^{[t]}}\mathcal{L} = \nabla_{\mathbf{h}^{[t]}}\mathbf{y}^{[t]} \cdot \nabla_{\mathbf{y}^{[t]}}\mathcal{L} + \nabla_{\mathbf{h}^{[t]}}\mathbf{h}^{[t+1]} \cdot \nabla_{\mathbf{h}^{[t+1]}}\mathcal{L} \tag{13}$$

The first term is analogous to Eq. 12, while to find $\nabla_{\mathbf{h}^{[t]}}\mathbf{h}^{[t+1]}$ we need to apply the chain rule again:

$$\nabla_{\mathbf{h}^{[t]}}\mathbf{h}^{[t+1]} = \nabla_{\mathbf{h}^{[t]}} \tanh\left(\mathbf{W}\mathbf{h}^{[t]} + \mathbf{U}\mathbf{x}^{[t+1]} + \mathbf{b}\right) = \mathbf{W}^T \cdot \operatorname{diag}\left(1 - {\mathbf{h}^{[t]}}^2\right) \tag{14}$$

Therefore,

$$\nabla_{\mathbf{h}^{[t]}}\mathcal{L} = \mathbf{V}^T \cdot \nabla_{\mathbf{y}^{[t]}}\mathcal{L} + \mathbf{W}^T \cdot \operatorname{diag}\left(1 - {\mathbf{h}^{[t]}}^2\right) \cdot \nabla_{\mathbf{h}^{[t+1]}}\mathcal{L} \tag{15}$$

where we do not expand $\nabla_{\mathbf{h}^{[t+1]}}\mathcal{L}$ further as that is carried over during backpropagation (it corresponds to the δ in lab 4).

We now compute the gradients with respect to the parameters of the network, starting with the easy biases. $\mathbf{c}$ is used to compute $\mathbf{y}^{[t]}$ for every t, thus:

$$\nabla_{\mathbf{c}}\mathcal{L} = \sum_{t=1}^{\tau} \nabla_{\mathbf{c}}\mathbf{y}^{[t]} \cdot \nabla_{\mathbf{y}^{[t]}}\mathcal{L} = \sum_{t=1}^{\tau} \nabla_{\mathbf{y}^{[t]}}\mathcal{L} \tag{16}$$

Similarly, $\mathbf{b}$ is used to compute $\mathbf{h}^{[t]}$, therefore:

$$\nabla_{\mathbf{b}}\mathcal{L} = \sum_{t=1}^{\tau} \nabla_{\mathbf{b}}\mathbf{h}^{[t]} \cdot \nabla_{\mathbf{h}^{[t]}}\mathcal{L} = \sum_{t=1}^{\tau} \operatorname{diag}\left(1 - {\mathbf{h}^{[t]}}^2\right) \cdot \nabla_{\mathbf{h}^{[t]}}\mathcal{L} \tag{17}$$

Moving on to the three weight matrices, we have:

$$\nabla_{\mathbf{V}}\mathcal{L} = \sum_{t=1}^{\tau} \nabla_{\mathbf{V}^{[t]}}\mathbf{y}^{[t]} \cdot \nabla_{\mathbf{y}^{[t]}}\mathcal{L} \tag{18}$$

where we use $\nabla_{\mathbf{V}^{[t]}}\mathbf{y}^{[t]}$ to denote the gradient of $\mathbf{y}^{[t]}$ with respect to $\mathbf{V}$ *without* backpropagating, i.e., the contribution of $\mathbf{V}$ only at time t. In other words, you can think of $\mathbf{V}^{[1]}, \ldots, \mathbf{V}^{[t]}$ as dummy variables that all equal $\mathbf{V}$. Note that we must now deal with tensors: let $\mathbf{V}^{[t]} \in \mathbb{R}^{n\times m}$, then $\nabla_{\mathbf{V}^{[t]}}\mathbf{y}^{[t]} \in \mathbb{R}^{n\times m\times n}$, so that, since $\nabla_{\mathbf{y}^{[t]}}\mathcal{L} \in \mathbb{R}^n$, $\nabla_{\mathbf{V}}\mathcal{L} \in \mathbb{R}^{n\times m}$ (the last dimension disappears due to the dot products, just like normal matrix multiplication). Let's analyze each item of the final gradient:

$$
\begin{aligned}
(\nabla_V \mathcal{L})_{ij} &= \frac{\partial}{\partial \mathbf{V}_{ij}}\mathcal{L} && (19)\\
&= \sum_{t=1}^{\tau}\sum_{k=1}^{n} \frac{\partial \mathcal{L}}{\partial y_k^{[t]}} \cdot \frac{\partial y_k^{[t]}}{\partial V_{ij}^{[t]}} && (20)\\
&= \sum_{t=1}^{\tau}\sum_{k=1}^{n} \frac{\partial \mathcal{L}}{\partial y_k^{[t]}} \cdot \frac{\partial}{\partial V_{ij}^{[t]}} \sum_{\ell=1}^{m} V_{k\ell}^{[t]} h_\ell^{[t]} && (21)\\
&= \sum_{t=1}^{\tau}\sum_{k=1}^{n} \frac{\partial \mathcal{L}}{\partial y_k^{[t]}} \cdot \delta_{ik} h_j^{[t]} && (22)\\
&= \sum_{t=1}^{\tau} \frac{\partial \mathcal{L}}{\partial y_i^{[t]}} \cdot h_j^{[t]} && (23)
\end{aligned}
$$

Therefore, via the outer product:

$$
\nabla_{\mathbf{V}}\mathcal{L} = \sum_{t=1}^{\tau} \nabla_{\mathbf{y}^{[t]}}\mathcal{L} \cdot {\mathbf{h}^{[t]}}^T \tag{24}
$$

A faster way of reaching the same result is via:

$$
\nabla_{\mathbf{V}}\mathcal{L} = \sum_{t=1}^{\tau}\sum_{i=1}^{n} \nabla_{\mathbf{V}} y_i^{[t]} \cdot \nabla_{y_i^{[t]}}\mathcal{L} \tag{25}
$$

and noticing that $\nabla_{\mathbf{V}} y_i^{[t]}$ is a matrix with all zeros except for row i which equals ${\mathbf{h}^{[t]}}^T$.

Moving on to $\mathbf{W}$, using the same insight, we have:

$$
\nabla_{\mathbf{W}}\mathcal{L} = \sum_{t=1}^{\tau} \nabla_{\mathbf{W}}\mathbf{h}^{[t]} \cdot \nabla_{\mathbf{h}^{[t]}}\mathcal{L} = \sum_{t=1}^{\tau}\sum_{i=1}^{n} \nabla_{\mathbf{W}} h_i^{[t]} \cdot \nabla_{h_i^{[t]}}\mathcal{L} \tag{26}
$$

where the i-th row of $\nabla_{\mathbf{W}} h_i^{[t]}$ equals, by the chain rule,

$$
\nabla_{\mathbf{W}} h_i^{[t]} = \left(1 - {h_i^{[t]}}^2\right) \cdot {\mathbf{h}^{[t-1]}}^T \tag{27}
$$

therefore:

$$
\nabla_{\mathbf{W}}\mathcal{L} = \sum_{t=1}^{\tau}\sum_{i=1}^{n} \nabla_{\mathbf{W}} h_i^{[t]} \cdot \nabla_{h_i^{[t]}}\mathcal{L} = \sum_{t=1}^{\tau} \operatorname{diag}\left(1 - {\mathbf{h}^{[t]}}^2\right) \cdot \nabla_{\mathbf{y}^{[t]}}\mathcal{L} \cdot {\mathbf{h}^{[t-1]}}^T \tag{28}
$$

Finally, in a similar way,

$$
\nabla_{\mathbf{U}}\mathcal{L} = \sum_{t=1}^{\tau} \nabla_{\mathbf{U}}\mathbf{h}^{[t]} \cdot \nabla_{\mathbf{h}^{[t]}}\mathcal{L} = \sum_{t=1}^{\tau}\sum_{i=1}^{n} \nabla_{\mathbf{U}} h_i^{[t]} \cdot \nabla_{h_i^{[t]}}\mathcal{L} = \sum_{t=1}^{\tau} \operatorname{diag}\left(1 - {\mathbf{h}^{[t]}}^2\right) \cdot \nabla_{\mathbf{y}^{[t]}}\mathcal{L} \cdot {\mathbf{x}^{[t]}}^T \tag{29}
$$

Exercise 2

In the third exercise, we are going to be estimating only the temperature value of the next hour from the given past 24 hours of weather-related information. Thus we will not be computing any intermediate output from the RNN and only one scalar value at the final step. Additionally, we will use mean square error as a loss function.

Given this information, show that:

$$\nabla_{\mathbf{h}^{[\tau]}}\mathcal{L} = 2(\hat{y} - y)\mathbf{V}^T \tag{30}$$

$$\nabla_{\mathbf{h}^{[t]}}\mathcal{L} = \mathbf{W}^T \cdot \text{diag}\left(1 - \mathbf{h}^{[t+1]^2}\right) \cdot \nabla_{\mathbf{h}^{[t+1]}}\mathcal{L} \tag{31}$$

$$\nabla_{\mathbf{c}}\mathcal{L} = 2(\hat{y} - y) \tag{32}$$

$$\nabla_{\mathbf{V}}\mathcal{L} = 2(\hat{y} - y)\mathbf{h}^{[\tau]^T} \tag{33}$$

Solution

In the first formula we can directly expand the gradient of the loss:

$$\nabla_{\mathbf{h}^{[\tau]}}\mathcal{L} = \mathbf{V}^T \cdot \nabla_{\mathbf{y}^{[\tau]}}\mathcal{L} = \mathbf{V}^T \cdot 2(y - \hat{y}) \tag{34}$$

In the other cases, since only the last output is connected to the loss, the formulas developed above do not need to consider the paths connecting intermediate outputs. Therefore,

$$\nabla_{\mathbf{h}^{[t]}}\mathcal{L} = \nabla_{\mathbf{h}^{[t]}}\mathbf{h}^{[t+1]} \cdot \nabla_{\mathbf{h}^{[t+1]}}\mathcal{L} = \mathbf{W}^T \cdot \text{diag}\left(1 - \mathbf{h}^{[t]^2}\right) \cdot \nabla_{\mathbf{h}^{[t+1]}}\mathcal{L} \tag{35}$$

for the bias:

$$\nabla_{\mathbf{c}}\mathcal{L} = \nabla_{\mathbf{c}}\mathbf{y}^{[\tau]} \cdot \nabla_{\mathbf{y}^{[\tau]}}\mathcal{L} = 2(\hat{y} - y) \tag{36}$$

and for the last weight matrix:

$$\nabla_{\mathbf{V}}\mathcal{L} = \nabla_{\mathbf{V}}\mathbf{y}^{[\tau]} \cdot \nabla_{\mathbf{y}^{[\tau]}}\mathcal{L} = 2(\hat{y} - y)\mathbf{h}^{[\tau]^T} \tag{37}$$

where $\mathbf{V}$ now has only one row since the network outputs scalars.