Mathematics of Backpropagation Through Time

vxnuaj (Juan Vera)

January 13, 2025

1 Background

Recurrent Neural Networks (RNNs) are a class of neural networks designed to capture dependencies over sequences of inputs, for instance, a sequence of words over time, $t \in [1, T]$.

They differ from Feed-Forward Neural Networks (FFNNs) in both, it's forward propagation and backward propagation, the change in the latter caused by the change in the former.

1.1 Fully Connected Neural Network

A 3-layer FFNN is typically presented as:

$$H^{(1)} = \phi(XW^{(1)}_{xh_1} + b_{h_1}) \tag{1}$$

$$H^{(2)} = \phi(H^{(1)}W^{(1)}_{h_1h_2} + b_{h_2}) \tag{2}$$

$$A^{(3)} = \text{softmax}(H^{(2)}W^{(2)}_{h_2a} + b_{h_3}) \tag{3}$$

where

1. $H^{(i)}$ is the output of the ith layer
2. A is the output of the final softmax activation
3. b_{h_i} is the bias constant for the ith layer
4. X is the input
5. $W^{(i)}$ is the set of weights for the ith layer
6. ϕ and softmax are activation functions for the hidden and output layers respectively.

1.2 Recurrent Neural Network

Adding in recurrence to each layer, defined as $H_{t-1}^{(i)}W^{(i)}$, through an addition operation, transforms a FFNN into an RNN, presented as:

$$(1) \qquad H_t^{(1)} = \phi(X_t^{(1)}W_{xh_1}^{(1)} + H_{t-1}^{(1)}W_{h_1h_1}^{(1)} + b_{h_1}^{(1)})$$

$$(2) \qquad H_t^{(2)} = \phi(H_t^{(1)}W_{h_1h_2}^{(2)} + H_{t-1}^{(2)}W_{h_2h_2}^{(2)} + b_{h_2}^{(2)})$$

$$(3) \qquad A_t^{(3)} = \text{softmax}(H_t^{(2)}W_{h_2a}^{(3)} + b_a^{(3)})$$

where, regarding the new operands:

1. $H_{t-1}^{(i)}$ is the hidden state for the ith layer
2. $W_{h_ih_i}^{(i)}$ is the set of weights to transform the hidden state, for the ith layer

The 3-Layer RNN can be equivalently referred to as a 2-Layer Stacked RNN, where "2-layer" accounts for the set of two recurrent layers.

The hidden state of an RNN is what allows for the model to effectively summarize information over earlier tokens of an input sequence providing the ability to learn from longer-term dependencies or more generally, time-steps.

RNNs have been more commonly used for language modeling due to their ability to capture long-term dependencies in sequences. Unlike Markov Models, which rely on storing exponentially increasing sets of probabilities as the n-gram size grows, RNNs use a single hidden state matrix $H_{t-1}^{(i)}$ to capture prior context, allowing for some improved efficiency.

2 Backpropagation Through Time (BPTT)

Let's assume a 2-Layer Stacked RNN as:

$$H_t^{(1)} = \phi(X_t^{(1)}W_{xh_1}^{(1)} + H_{t-1}^{(1)}W_{h_1h_1}^{(1)} + b_{h_1}^{(1)})$$

$$H_t^{(2)} = \phi(H_t^{(1)}W_{h_1h_2}^{(2)} + H_{t-1}^{(2)}W_{h_2h_2}^{(2)} + b_{h_2}^{(2)})$$

$$A_t^{(3)} = \text{softmax}(H_t^{(2)}W_{h_2a}^{(3)} + b_a^{(3)})$$

with loss L defined as the cross-entropy loss or equivalently as the negative log likelihood.

Key components and respective dimensions are defined as follows:

- Input $X_t^{(1)} \in \mathbb{R}^{n \times d}$
- First Layer Weights $W_{xh_1}^{(1)} \in \mathbb{R}^{d \times h_1}$
- First Layer Hidden State Weights $W_{h_1h_1}^{(1)} \in \mathbb{R}^{h_1 \times h_1}$
- First Layer Bias $b_{h_1} \in \mathbb{R}^{1 \times h_1}$
- First Layer Output $H_t^{(1)} \in \mathbb{R}^{n \times h_1}$
- First Layer Hidden State $H_{t-1}^{(1)} \in \mathbb{R}^{n \times h_1}$
- Second Layer Weights $W_{h_1h_2}^{(2)} \in \mathbb{R}^{h_1 \times h_2}$
- Second Layer Hidden State Weights $W_{h_2h_2}^{(2)} \in \mathbb{R}^{h_2 \times h_2}$
- Second Layer Bias $b_{h_2} \in \mathbb{R}^{1 \times h_2}$
- Second Layer Output $H_t^{(2)} \in \mathbb{R}^{n \times h_2}$
- Second Layer Hidden State $H_{t-1}^{(2)} \in \mathbb{R}^{n \times h_2}$
- Third Layer Weights $W_{h_2a}^{(3)} \in \mathbb{R}^{h_2 \times a}$
- Third Layer Bias $b_a \in \mathbb{R}^{1 \times a}$
- Third Layer Output $A_t^{(3)} \in \mathbb{R}^{n \times a}$

where

- h_i is the count of hidden units in the ith hidden layer
- n is the batch size
- a is the count of output units, equivalently the size of the one-hot target vector or vocabulary size.

At first glance, given that an output to the ith layer is dependent on hidden states at $t-1$ and those are dependent on the hidden state at $t-2$, it's clear that computing partial derivatives to attain $\frac{\partial L}{\partial H^{(i)}}$ and $\frac{\partial L}{\partial W_{h_ih_i}^{(i)}}$ will require computing the chain rule through multiple time steps t such that we end up computing partial derivatives across $H_t^{(i)}, H_{t-1}^{(i)} \dots H_{t-(T-1)}^{(i)}$ (where T is the total count of time steps, or equivalently sequence length), for each layer $i \in [1, 3]$, or more generally, $i \in [1, \mathcal{L}]$ where $\mathcal{L}$ is the total number of layers in the RNN.
This results in an exponentially increasing count of gradient factors in the chain rule as the sequence length, or equivalently total time steps T, increases. Hence, in the derived equations (Section 2.1) T is constrained to $T = 2$, to avoid complexity.

2.1 BPTT at $t = 2 = T$

Backpropagation through the stacked RNN at time step $t = 2 = T$, can be computed as:

$$(1) \qquad \frac{\partial L}{\partial Z_t^{(3)}} = A_t^{(3)} - \text{one-hot}(y) \ \in \mathbb{R}^{n \times a}$$

$$(2) \qquad \frac{\partial L}{\partial W_{h_2 a}^{(3)}} = \left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial W_{h_2 a}^{(3)}}\right) = (H_t^{(2)})^T \cdot \frac{\partial L}{\partial Z_t^{(3)}} \ \in \mathbb{R}^{h_2 \times a}$$

$$(3) \qquad \frac{\partial L}{\partial Z_t^{(2)}} = \left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right) = \left(\frac{\partial L}{\partial Z_t^{(3)}} \cdot (W_{h_2 a}^{(3)})^T\right) \odot \phi'(Z_t^{(2)}) \ \in \mathbb{R}^{n \times h_2}$$

$$(4) \qquad \frac{\partial L}{\partial W_{h_1 h_2}^{(2)}} = \left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right)\left(\frac{\partial Z_t^{(2)}}{\partial W_{h_1 h_2}^{(2)}}\right) +$$

$$\left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right)\left(\frac{\partial Z_t^{(2)}}{\partial H_{t-1}^{(2)}}\right)\left(\frac{\partial H_{t-1}^{(2)}}{\partial Z_{t-1}^{(2)}}\right)\left(\frac{\partial Z_{t-1}^{(2)}}{\partial W_{h_1 h_2}^{(2)}}\right)$$

$$= ((H_t^{(1)})^T \cdot \frac{\partial L}{\partial Z_t^{(2)}}) + (H_{t-1}^{(1)})^T \cdot \left(\left(\frac{\partial L}{\partial Z_t^{(2)}} \cdot (W_{h_2 h_2}^{(2)})\right) \odot (\phi'(Z_{t-1}^{(2)}))\right) \ \in \mathbb{R}^{h \times h_2}$$

$$(5) \qquad \frac{\partial L}{\partial W_{h_2 h_2}^{(2)}} = (H_{t-1}^{(2)})^T \cdot \left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right)\left(\frac{\partial Z_t^{(2)}}{\partial W_{h_2 h_2}^{(2)}}\right)$$

$$+(W_{h_2 h_2}^{(2)})^T \cdot \left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right)\left(\frac{\partial Z_t^{(2)}}{\partial H_{t-1}^{(2)}}\right)\left(\frac{\partial H_{t-1}^{(2)}}{\partial Z_{t-1}^{(2)}}\right)\left(\frac{\partial Z_{t-1}^{(2)}}{\partial W_{h_2 h_2}^{(2)}}\right)$$

$$= (H_{t-1}^{(2)})^T \left((H_{t-1}^{(2)})^T \cdot \frac{\partial L}{\partial Z_t^{(2)}}\right) + (W_{h_2 h_2}^{(2)})^T \cdot \left((H_{t-2}^{(2)})^T \cdot \left(\left(\frac{\partial L}{\partial Z_t^{(2)}} \cdot (W_{h_2 h_2}^{(2)})\right) \odot (\phi'(Z_{t-1}^{(2)}))\right)\right) \ \in \mathbb{R}^{h_2 \times h_2}$$

$$(6) \qquad \frac{\partial L}{\partial Z_t^{(1)}} = \left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right)\left(\frac{\partial Z_t^{(2)}}{\partial H_t^{(1)}}\right)\left(\frac{\partial H_t^{(1)}}{\partial Z_t^{(1)}}\right)$$

$$= \left(\frac{\partial L}{\partial Z_t^{(2)}} \cdot (W_{h_1h_2}^{(2)})^T\right) \odot \phi'(Z_t^{(1)}) \ \in \mathbb{R}^{n \times h}$$

$$(7) \qquad \frac{\partial L}{\partial W_{xh_1}^{(1)}} = \left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right)\left(\frac{\partial Z_t^{(2)}}{\partial H_t^{(1)}}\right)\left(\frac{\partial H_t^{(1)}}{\partial Z_t^{(1)}}\right)\left(\frac{\partial Z_t^{(1)}}{\partial W_{xh_1}^{(1)}}\right)$$

$$+\left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right)\left(\frac{\partial Z_t^{(2)}}{\partial H_t^{(1)}}\right)\left(\frac{\partial H_t^{(1)}}{\partial Z_t^{(1)}}\right)\left(\frac{\partial Z_t^{(1)}}{\partial H_{t-1}^{(1)}}\right)\left(\frac{\partial H_{t-1}^{(1)}}{\partial Z_{t-1}^{(1)}}\right)\left(\frac{\partial Z_{t-1}^{(1)}}{\partial W_{xh_1}^{(1)}}\right)$$

$$= (X_t^{(1)})^T \cdot \frac{\partial L}{\partial Z_t^{(1)}} + (X_{t-1}^{(1)})^T \cdot \left(\left(\frac{\partial L}{\partial Z_t^{(1)}} \cdot (W_{h_1h_1}^{(1)})\right) \odot (\phi'(Z_{t-1}^{(1)}))\right) \in \mathbb{R}^{d \times h_1}$$

$$(8) \qquad \frac{\partial L}{\partial W_{h_1h_1}^{(1)}} = (H_{t-1}^{(1)})^T \cdot \left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right)\left(\frac{\partial Z_t^{(2)}}{\partial H_t^{(1)}}\right)\left(\frac{\partial H_t^{(1)}}{\partial Z_t^{(1)}}\right)\left(\frac{\partial Z_t^{(1)}}{\partial W_{h_1h_1}^{(1)}}\right)$$

$$+(W_{h_1h_1}^{(1)})^T \cdot \left(\frac{\partial L}{\partial Z_t^{(3)}}\right)\left(\frac{\partial Z_t^{(3)}}{\partial H_t^{(2)}}\right)\left(\frac{\partial H_t^{(2)}}{\partial Z_t^{(2)}}\right)\left(\frac{\partial Z_t^{(2)}}{\partial H_t^{(1)}}\right)\left(\frac{\partial H_t^{(1)}}{\partial Z_t^{(1)}}\right)\left(\frac{\partial Z_t^{(1)}}{\partial H_{t-1}^{(1)}}\right)\left(\frac{\partial H_{t-1}^{(1)}}{\partial Z_{t-1}^{(1)}}\right)\left(\frac{\partial Z_{t-1}^{(1)}}{\partial W_{h_1h_1}^{(1)}}\right)$$

$$= (H_{t-1}^{(1)})^T \cdot \left((H_{t-1}^{(1)})^T \cdot \frac{\partial L}{\partial Z_t^{(1)}}\right) + (W_{h_1h_1}^{(1)})^T \cdot \left((H_{t-2}^{(1)})^T \left(\left(\frac{\partial L}{\partial Z_t^{(1)}} \cdot (W_{h_1h_1}^{(1)})\right) \odot (\phi'(Z_{t-1}^{(1)}))\right)\right) \in \mathbb{R}^{h_1 \times h_1}$$

It's clear that there is a high count of gradient factors in the chain rule for computing the partials. For merely 2 recurrent layers, with $T = 2$, it's clear that a problem of vanishing gradients or exploding gradients will come to fruition fairly quickly as we scale T or the depth of the RNN.

2.1.1 Generalizing BPTT to $t \in [1, T = 2]$

For multiple timesteps, $t \in [1, T = 2]$, we can compute the total gradients as a summation of the gradients at each t:

$$(1) \qquad \frac{\partial L}{\partial Z^{(3)}} = \frac{1}{T}\sum_{t=1}^{T}\frac{\partial L}{\partial Z_t^3}$$

$$(2) \qquad \frac{\partial L}{\partial W_{h_2a}^{(3)}} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial L}{\partial W_{h_2a}^{(3)}}\right)_{(t)}$$

$$(3) \qquad \frac{\partial L}{\partial Z^{(2)}} = \frac{1}{T}\sum_{t=1}^{T}\frac{\partial L}{\partial Z_t^2}$$

$$(4) \qquad \frac{\partial L}{\partial W_{hh_2}^{(2)}} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial L}{\partial W_{hh_2}^{(2)}}\right)_{(t)}$$

$$(5) \qquad \frac{\partial L}{\partial W_{h_2h_2}^{(2)}} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial L}{\partial W_{h_2h_2}^{(2)}}\right)_{(t)}$$

$$(6) \qquad \frac{\partial L}{\partial Z^{(1)}} = \frac{1}{T}\sum_{t=1}^{T}\frac{\partial L}{\partial Z_t^1}$$

$$(7) \qquad \frac{\partial L}{\partial W_{xh}^{(1)}} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial L}{\partial W_{xh}^{(1)}}\right)_{(t)}$$

$$(8) \qquad \frac{\partial L}{\partial W_{hh}^{(1)}} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial L}{\partial W_{hh}^{(1)}}\right)_{(t)}$$

These computations are all generalizable to any $T \in [1, \infty]$ but the original forward pass (Sec 1.2, Eq. 1-3) will begin to rely on a larger set of t such that there will be more gradient factors in the chain rule during backpropagation purely for a single time step t, as we'll have to backpropate through a larger set of hidden states H_t to get the gradients for a single Z or W.

The count of matrix products in the chain rule will continue to scale when backpropagating through multiple t (see Sec 2.2.1, Eq. 1-8) and multiple layers of the RNN, ultimately increasing the probability for vanishing and exploding gradients.

Slightly more formally, assuming we can define $\frac{\partial L}{\partial W^{(i)}}$ as a Jacobian Matrix, if it's eigenvalue, λ_i, is > 1, the matrix product will scale up the gradients. Otherwise, it'll shrink the gradients.

As we backpropagate through more layers with more matrix products, if λ is consistently > 1 or < 1, the magnitude of the gradients will exponentially increase or decrease respectively.

2.2 BPTT at $t \in [1, T \to \infty]$

As mentioned earlier, the count of matrix products in the chain rule during BPTT tends to scale exponentially for a larger T or equivalently a larger input sequence length.

More explicitly, to compute $\frac{\partial L}{\partial W_{hh}^{(1)}}$ for any $T \in [1, \infty]$:

$$\frac{\partial L}{\partial W_{hh}^{(1)}} = \sum_{t=1}^{T} \left(H_{t-1}^{(1)}\right)^T \cdot \left(\prod_{k=t}^{T} \frac{\partial Z_k^{(3)}}{\partial H_k^{(2)}} \cdot \frac{\partial H_k^{(2)}}{\partial Z_k^{(2)}} \cdot \frac{\partial Z_k^{(2)}}{\partial H_k^{(1)}} \cdot \frac{\partial H_k^{(1)}}{\partial Z_k^{(1)}} \right) \cdot \frac{\partial Z_t^{(1)}}{\partial W_{hh}^{(1)}}$$

As can be seen, we begin to rely not only on a $\sum$ operation but a $\prod$ which is terribly counterproductive for stable gradients.

A general solution to these shattered gradients is truncating the computation of gradients to be through $T - \tau$ time steps, where $\tau < T$. This leads to a decrease in complexity of the backpropagation and also serves as a means to mitigate overfitting, acting as a form of regularization, as the model is only learning based on gradients up to time $T - \tau$.

One can also opt for random truncation, as was proposed by Tallec and Ollivier [1].

References

[1] Tallec, Corentin, and Yann Ollivier. "Unbiasing Truncated Backpropagation Through Time." arXiv, 2017.