

# Basic RNN

Paul Tatasciore

May 18, 2020

## 1 Introduction

Implementing a basic recurrent neural network on the MNIST dataset for classification.

Here I plan to implement the backpropagation through time (BPTT) algorithm from scratch and observe the networks accuracy as long-term dependencies become more important. (i.e as more time steps are taken)

## 2 Basic RNN

Let  $N$  be the number of training examples inputed into the network,  $T$  be the number of time steps for each example, and  $y^{(n)}$  be a one-hot vector containing the correct class for example  $n$ .

The Basic RNN can then be defined as follows:

$$E = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T E_t^{(n)} \quad (1)$$

$$E_t^{(n)} = -y^{(n)} \cdot \log(\hat{y}_t^{(n)}) \quad (2)$$

$$\hat{y}_t^{(n)} = \text{softmax}(\hat{o}_t^{(n)}) \quad (3)$$

$$\hat{o}_t^{(n)} = W_{oh} h_t^{(n)} \quad (4)$$

$$h_t^{(n)} = \tanh(W_{hh} h_{t-1}^{(n)} + V_h x^{(n)} + b_h) \quad (5)$$

### 3 Backpropagation Through Time (BPTT)

We want to make the following updates to each of the weights:

$$W_{oh} = W_{oh} - \gamma \frac{dE}{dW_{oh}} \quad (6)$$

$$W_{hh} = W_{hh} - \gamma \frac{dE}{dW_{hh}} \quad (7)$$

$$V_h = V_h - \gamma \frac{dE}{dV_h} \quad (8)$$

$$b_h = b_h - \gamma \frac{dE}{db_h} \quad (9)$$

where  $\gamma \in [0, 1]$  is referred to as the learning rate.

Therefore the following derivatives will need to be calculated:

- $\frac{dE}{dW_{oh}}, \frac{dE}{dW_{hh}}, \frac{dE}{dV_h}, \frac{dE}{db_h}$

$$\frac{dE}{dW_{oh}} = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T \frac{dE_t^{(n)}}{d\hat{y}_t^{(n)}} \frac{d\hat{y}_t^{(n)}}{d\hat{o}_t^{(n)}} \frac{d\hat{o}_t^{(n)}}{dW_{oh}} \quad (10)$$

$$\frac{dE}{dW_{hh}} = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T \frac{dE_t^{(n)}}{d\hat{y}_t^{(n)}} \frac{d\hat{y}_t^{(n)}}{d\hat{o}_t^{(n)}} \frac{d\hat{o}_t^{(n)}}{dh_t^{(n)}} \left( \prod_{k=t+1}^T \frac{dh_k^{(n)}}{dh_{k-1}^{(n)}} \right) \frac{dh_t^{(n)}}{dW_{hh}} \quad (11)$$

$$\frac{dE}{dV_h} = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T \frac{dE_t^{(n)}}{d\hat{y}_t^{(n)}} \frac{d\hat{y}_t^{(n)}}{d\hat{o}_t^{(n)}} \frac{d\hat{o}_t^{(n)}}{dh_t^{(n)}} \left( \prod_{k=t+1}^T \frac{dh_k^{(n)}}{dh_{k-1}^{(n)}} \right) \frac{dh_t^{(n)}}{dV_h} \quad (12)$$

$$\frac{dE}{db_h} = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T \frac{dE_t^{(n)}}{d\hat{y}_t^{(n)}} \frac{d\hat{y}_t^{(n)}}{d\hat{o}_t^{(n)}} \frac{d\hat{o}_t^{(n)}}{dh_t^{(n)}} \left( \prod_{k=t+1}^T \frac{dh_k^{(n)}}{dh_{k-1}^{(n)}} \right) \frac{dh_t^{(n)}}{db_h} \quad (13)$$

The following derivatives still need to be calculated:

- $\frac{dE_t^{(n)}}{d\hat{y}_t^{(n)}}, \frac{d\hat{y}_t^{(n)}}{d\hat{o}_t^{(n)}}, \frac{d\hat{o}_t^{(n)}}{dW_{oh}}, \frac{d\hat{o}_t^{(n)}}{dh_t^{(n)}}, \frac{dh_k^{(n)}}{dh_{k-1}^{(n)}}, \frac{dh_t^{(n)}}{dW_{hh}}, \frac{dh_t^{(n)}}{dV_h}, \frac{dh_t^{(n)}}{db_h}$

$$\frac{dE_t^{(n)}}{d\hat{y}_t^{(n)}} = -\frac{y^{(n)}}{\hat{y}_t^{(n)}} \quad (14)$$

$$\frac{d\hat{y}_{ti}^{(n)}}{d\hat{o}_{tj}^{(n)}} = \begin{cases} \hat{y}_t^{(n)}(1 - \hat{y}_t^{(n)}) & i = j \\ -\hat{y}_t^{(n)}\hat{y}_t^{(n)} & i \neq j \end{cases} \quad (15)$$

$$\frac{d\hat{o}_t^{(n)}}{dW_{oh}} = h_t^{(n)} \quad (16)$$

$$\frac{d\hat{o}_t^{(n)}}{dh_t^{(n)}} = W_{oh} \quad (17)$$

$$\frac{dh_k^{(n)}}{dh_{k-1}^{(n)}} = W_{hh}^T [1 - \tanh^2(W_{hh}h_{k-1}^{(n)} + V_h x^{(n)} + b_h)] \quad (18)$$

$$\frac{dh_t^{(n)}}{dW_{hh}} = [1 - \tanh^2(W_{hh}h_{t-1}^{(n)} + V_h x^{(n)} + b_h)] \otimes h_{t-1}^{(n)} \quad (19)$$

$$\frac{dh_t^{(n)}}{dV_h} = [1 - \tanh^2(W_{hh}h_{t-1}^{(n)} + V_h x^{(n)} + b_h)] \otimes x^{(n)} \quad (20)$$

$$\frac{dh_t^{(n)}}{db_h} = 1 - \tanh^2(W_{hh}h_{t-1}^{(n)} + V_h x^{(n)} + b_h) \quad (21)$$

After bit of simplifing...

$$\frac{dE_t^{(n)}}{d\hat{o}_t^{(n)}} = \frac{dE_t^{(n)}}{d\hat{y}_t^{(n)}} \frac{d\hat{y}_t^{(n)}}{d\hat{o}_t^{(n)}} = \begin{cases} y^{(n)}(\hat{y}_t^{(n)} - 1) & i = j \\ y^{(n)}\hat{y}_t^{(n)} & i \neq j \end{cases} = (\hat{y}_t^{(n)} - y^{(n)}) \quad (22)$$

$$\frac{dE}{dW_{oh}} = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (\hat{y}_t^{(n)} - y^{(n)}) \otimes h_t^{(n)} \quad (23)$$

$$\frac{dE}{dW_{hh}} = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T (\hat{y}_t^{(n)} - y^{(n)})^T W_{oh} \left( \prod_{k=t+1}^T \frac{dh_k^{(n)}}{dh_{k-1}^{(n)}} \right) \frac{dh_t^{(n)}}{dW_{hh}} \quad (24)$$

## References

- [1] Denny Britz, *Recurrent Neural Networks Tutorial, Part 3 – Backpropagation Through Time and Vanishing Gradients*  
<http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-/backpropagation-through-time-and-vanishing-gradients/>
  
- [2] Carter Brown, *Gradients for RNN*  
<https://github.com/go2carter/nn-learn/blob/master/grad-deriv-tex/rnn-grad-deriv.pdf>