Basic RNN

Paul Tatasciore

May 19, 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:

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

$$E_t^{(n)} = - \sum_{i=1}^{\#classes} y_i^{(n)} \cdot log_i(\hat{y}_t^{(n)}) \tag{2}$$

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

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

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

3 Backpropagation Through Time (BPTT)

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

$$W_{oh} = W_{oh} - \gamma \frac{dE}{dW_{oh}} \tag{6}$$

$$W_{hh} = W_{hh} - \gamma \frac{dE}{dW_{hh}} \tag{7}$$

$$V_h = V_h - \gamma \frac{dE}{dV_h} \tag{8}$$

$$b_h = b_h - \gamma \frac{dE}{db_h} \tag{9}$$

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

Therefore the following derivatives will need to be calculated:

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

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

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

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

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

The following derivatives still need to be calculated:

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

$$\frac{dE_t^{(n)}}{d\hat{y}_t^{(n)}} = -\sum_{i=1}^{\#classes} \frac{y_i^{(n)}}{\hat{y}_{ti}^{(n)}} \tag{14}$$

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

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

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

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

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

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

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

Plugging into eq(10) gives...

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

$$= y_i^{(n)}(\hat{y}_{ti}^{(n)} - 1) + \sum_{i \neq j}^{\#classes} y_i^{(n)}\hat{y}_{tj}^{(n)} = -y_i^{(n)} + \sum_{j=1}^{\#classes} y_j^{(n)}\hat{y}_{ti}^{(n)} \tag{23}$$

$$= -y_i^{(n)} + \hat{y}_{ti}^{(n)} \sum_{j=1}^{\#classes} y_j^{(n)} = (\hat{y}_t^{(n)} - y^{(n)}) \tag{24}$$

$$\frac{dE}{dW_{oh}} = \frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}(\hat{y}_t^{(n)} - y^{(n)}) \otimes h_t^{(n)T} \tag{25}$$

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

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`