Report for Programming Assignment-3

April 19, 2019

Checklist :

- ☑ checked.
- ☐ unchecked.
- ☒ not done.

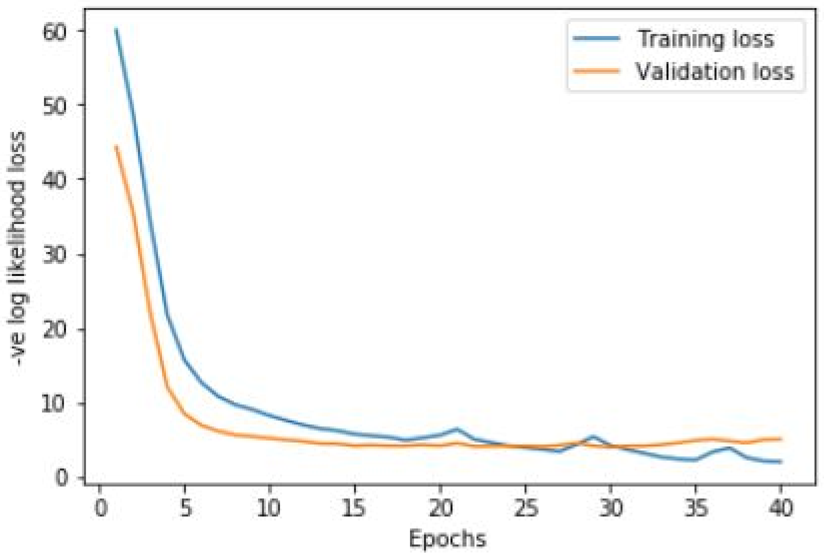
The task you need to ensure before submission.

- ☐ We have read all the instruction carefully and followed them to our best ability.
- ☐ We have written the name, roll no in report.
- ☐ Run sanity_check.sh.
- ☐ We will be submitting only single submission on behalf of our team.
- ☐ We have not included unnecessary text, pages, logos in the assignment.
- ☐ We have not used any high level APIs(Keras, Estimators for e.g.).
- ☐ We have not copied anything for this assignment.

Team Mate 1 : Anshul Kumar
Roll No : ME15B082
Team Mate 2 : Ashutosh Raj
Roll No : ME15B086

Report:

1 Answer 1:



Here, we observe that Validation error is initially lesser than the Training loss. The validation set (about 1000 examples) is much lesser in size as compared to the training set (about 13,000 examples). We believe it is because, the model has sufficiently been able to capture the variations in the small validation dataset at the cost of low -ve likelihood error, during the first 25 epochs of training. However, after training for longer time, the model overfits and validation loss becomes higher than the training loss.

2 Answer 2:

best: Learning rate : 0.001 Batch :128 Drop$_{p}rob : 0.2init : 1(Xavier)Val_{a}cc :$ 37.5
$Learningrate : 0.01 Batch : 128 Drop_{p}rob : 0.2 init : 1 Val_{a}cc : 22.02$
$Learningrate : 0.0001 Batch : 128 Drop_{p}rob : 0.2 init : 1 Val_{a}cc : 17.26$
$Learningrate : 0.001 Batch : 256 Drop_{p}rob : 0.2 init : 1 Val_{a}cc : 30.78$
$Learningrate : 0.001 Batch : 1024 Drop_{p}rob : 0.2 init : 1 Val_{a}cc : 29.87$
$Learningrate : 0.001 Batch : 128 Drop_{p}rob : 0.2 init : 0 Val_{a}cc : 33.2$

3 Answer 4:

Input to Embedding layer : batchsize x time$_steps(128x61)$
$InputofEncodinglayer : batchsizextime_stepsxinembsize(128x61x256)$
$Inputtoattentionlayer : batchsizexencstatesize(128x512(concatenatedfromforwardandbackwarddirectionall$
$Outputofattentionlayer : batch_sizexenc_state_size(128x256)$
$Inputofdecoderlayer : batch_sizexoutembsize, batch_sizexenc_state_size$
$Outputofdecoderlayer(beforesoftmax) : batch_sizexdec_state_size$
$Outputofdecoderlayer(aftersoftmax) : batch_sizextarget_vocab_size(128x62)$

4 Answer 6:

Attention mechanism basically forms a context vector by taking all the cells outputs as inputs to compute the source language words for each single word which decoder wants to generate. And by utilizing this, it becomes possible for the decoder to have a global context with more importance(attention) to certain words in the context, for predicting the target language word/character. The equations for the attention model we have used:

$$\alpha_{ts} = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \qquad \text{[Attention weights]} \qquad (1)$$

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \qquad \text{[Context vector]} \qquad (2)$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W_c}[\boldsymbol{c}_t; \boldsymbol{h}_t]) \qquad \text{[Attention vector]} \qquad (3)$$

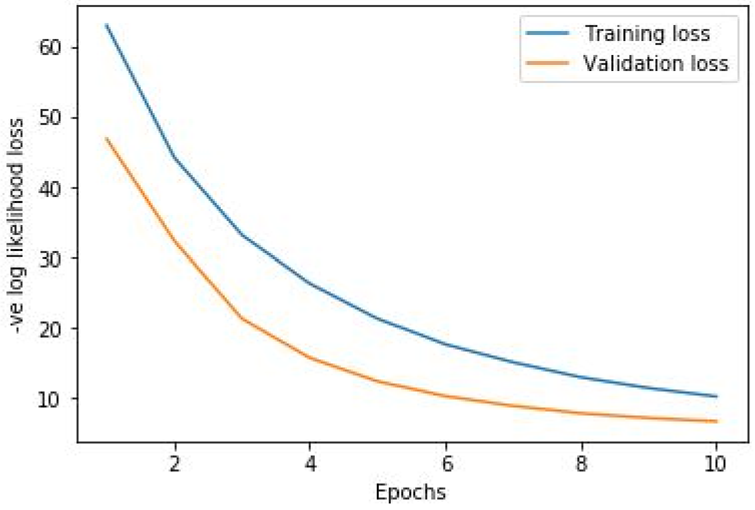
During decoding, context vectors are computed for every output word. So we will have a 2D matrix whose size is number of target words multiplied by number of source words. Equation (1) demonstrates how to compute a single value given one target word and a set of source word. Once context vector is computed, attention vector could be computed by context vector, target word, and attention function f. We have used Luong's attention.

5 Answer 9:

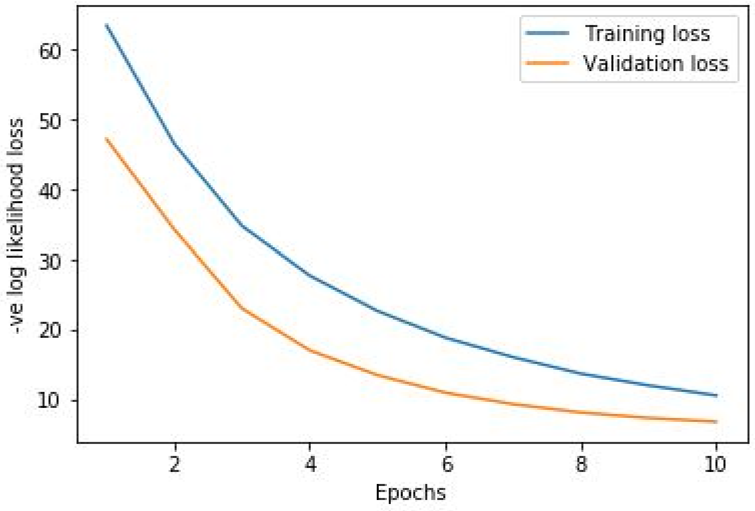
The effect of using two-layered decoder over one-layered decoder can be seen as contributing more to increasing memory rather than to hierarchical processing. This is also the reason we don't see much difference in the learning curves for the two cases (as shown below). One advantage that a multilayered RNN offeres over a single layered RNN is that we are able to process the input at various time scales. For example, the hidden state of the i-th layer is given by :

$$a_i^t = tanh(W_i a_i^{t-1} + a_{i-1}^t Z_i)$$

, which inherently captures different time scales.



This plot above shows the Loss variation over the epochs for one-layered encoder-decoder architecture.



This plot above shows the Loss variation over the epochs for the one -layered encoder and two-layered decoder architecture.

6 Answer 7:

The model with attention gave a lower validation error.