

# NLP Unit 2 Evaluation

Sep 27, 2016, 1:30 pm to 6:30 pm

## Objective

In this exercise we will build a Bidirectional RNN based sorting model. Given a sequence of arbitrary length, this model produces an output which is the sorted form of the input.

## Steps

### (a) Dataset creation

- i. Assume our input numbers to be sorted are positive integers in the range 0 to 31. We need to create multiple sequences where the sequence length is a variable and the elements of the sequence are constituted by random integers in the range 0 to 31. Choose the sequence length in the range 2 to 10. They are to be constructed in random order. Elements in a sequence are unique and no duplicate elements. Example: [21, 2, 13], [9, 0], [3, 9, 26, 31, 7, 15], ...
- ii. As per the instruction above, write a script to build this dataset of 10000 sequences. Represent an element of the sequence using a 32 dimensional 1-hot binary vector. The output of the sequence is the sorted list of the given input. The sort order is ascending. Optionally you can implement both ascending and descending. Represent the output element in the same way as the input element is represented.
- iii. You need to form sub sequences of the input sequence and its corresponding target output sequence. Each RNN stage (time step) receives one element. If a sequence has 6 elements, this will need 6 time steps. At any time  $t < T$ , we will have a proper subsequence and you need to generate the corresponding target. For example: assume the input sequence is [3, 9, 26, 31, 7, 15]. At step  $t = 6$ , which is the sequence length  $T$ , the target for RNN is [3, 7, 9, 15, 26, 31]. At any intermediate stage, say,  $t = 2$ , the input will be: [3, 9] and the targets will be [3, 9].

### (b) Building the bidirectional RNN

- i. Modify the code from Andrej Karpathy to take the input and target sequences as per the specification given in part (a) above. That is you will now have a 32 element input layer, 32 element softmax output layer for the RNN. You may choose the hidden layer size to be 8, 16, 24, 32

- ii. Develop a class BiDirectionalRnn that has 2 instances of the RNN from previous step.
  - iii. Write the train() procedure as per the algorithm explained during the class. You may refer my slides at:  
<http://www.slideshare.net/ananth/recurrent-neural-networks-lstm-and-gru>
  - iv. Write the procedure predict() to perform prediction
  - v. Write a procedure to compute the error between the target and actual and print the results. We will use this during validation
- (c) Train the system with 80% of the dataset and use the 20% data to validate.
- (d) Validate, fine tune, improve the results
- (e) Write a simple script to read in a list interactively from the user on command line, perform the Neural Sort and print the results

## Deliverables

Submit the following by 6:30 pm, 27<sup>th</sup> Sep 2016:

1. Source code of BiDirectionalRnn class
2. Source of the test program

Do the following by 10 pm 28<sup>th</sup> Sep 2016:

1. Post your results and analysis on the Facebook. Regardless of the accuracy you got, explain in the document what went right and where you could have taken a better approach to getting better results.
2. Optionally you can include any graphics, visualization etc

Best wishes from the faculty, enjoy NLP development!

Anantharaman Palacode Narayana Iyer, Chitra GM