

# TTIC 31230 Problem Set 4

## Win 2017

Hao Jiang

February 2, 2017

### Problem 1

#### Implementation

In this assignment, I implements a LSTM RNN and train it with the given text. My implementation includes two functions: `LSTMCell` and `BuildModel`

The function `LSTMCell` build a single LSTM Cell, which takes three inputs  $x, h, c$  and return two outputs  $h_n, c_n$ .  $x$  is a vector of shape  $[B, ]$ , where  $B$  is the batch size. It represents the input at step  $t$ . Both  $h$  and  $c$  have shape  $[B, H]$ , where  $H$  is the hidden dimension. As described in the lecture, my code first map  $x$  to a matrix of shape  $[B, H]$ , and concatenate it with  $h$  to form a  $[B, 2H]$  matrix. This matrix is then used to generate forget gate, input gate and applied to  $c$  to generate  $c_n$ , which is then used to generate  $h_n$

In the function `BuildModel`, I take a input character array of size  $[B, T]$ , where  $T$  is the sample length, and use a loop to generate  $T$  connected LSTM Cells. For the first cell, the  $h$  and  $c$  are pre-initialized.  $c$  is an all-0 matrix indicating no memory from the beginning, and  $h$  is generated with an all-1 vector, indicating the leading character at the beginning of sentence.  $h$  is then map to  $[B, H]$  by a mapping matrix.

For the  $i$ -th cell, I feed in `input[:, i]` as  $x$ , and obtain  $h_n, c_n$ . The  $h_n$  is used to compute a SoftMax next-char prediction. Together with the real next char `input[:, i+1]`, we compute a LogLoss at step  $i$ . Per-character loss is computed by adding all these step loss together, and divide by both  $B$  and  $T$ .

#### Experiment Results

It takes 14-18 mins to train one epoch on my desktop. With 30 epoches, I obtain a perplexity of 3.41069 and the minimal average loss of 1.20308. The plot of perplexity and loss with each epoch can be seen in Figure 1

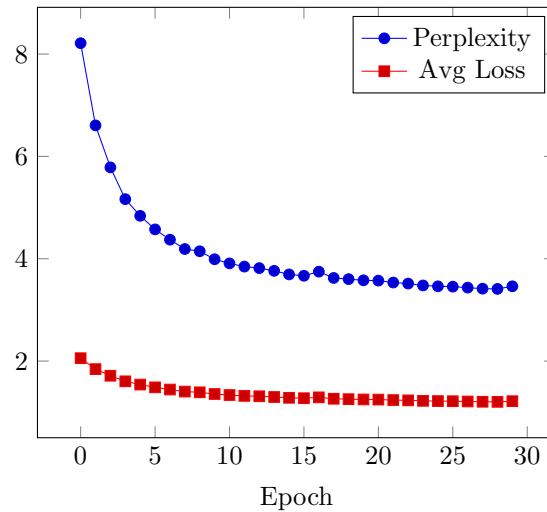


Figure 1: Perplexity and Loss with each Epoch

In some experiments I encounter exponential overflow / underflow error, so I change the the data type declared in edf.py from float32 to float64 to avoid this problem.

## Discussion

The best result of average loss appears at epoch 27, which gives an average loss of 1.20308. However, the predicted sentence doesn't quite make sense. It is

the agreements bring in the new york stock exchange composite trading  
the company 's company 's company 's company 's company 's company  
's company 's company 's company 's company 's company 's company  
's company 's company 's company 's company 's company 's company  
's company 's company 's company 's company 's company 's company  
's company 's company 's company 's company 's company 's company

In epoch 29, I get a sentence that makes more sense

the agreements bring is n't disclosed}

However it gives a slightly higher average loss of 1.21604.