

## A. Project

"Character-Level Language Model (CLLM)" through stacking LSTM in Tensorflow"

## B. Description

The CLLM is implemented in Tensorflow and compares a change of loss by a range of LSTM stacking level such as 1, 2 and 3

## C. Date & Author

July 2017, Gonsoo Moon

## D. Environment

1. Mac Pro 2.6 GHz Intel Core i5, 16 GB RAM
2. Tensorflow 1.0.0
3. Python 2.7

## E. How to run

Refer to Readme.txt

## F. Result

### 1. Input Process

The input file data/simple.txt is fed from the command line and is read as the <Figure1> shown.

<Figure1>

```
[Info] sentence: If you want to build a ship, don't drum up people together to collect wood and don't assign them tasks and work, but rather teach them to long for the endless immensity of the sea.
```

After that, the sentence is split into input and target batches by sequence length and moving window. In the <Figure2>, I used 10 of sequence length and 10 of moving window. the 181 characters of the sentence is converted into 18 input batches (dataX) and 18 target batches (dataY).

<Figure2>

```
[Info] Input and Target by batch: 0 If you wan -> f you want
[Info] Input and Target by batch: 10 t to build -> to build
[Info] Input and Target by batch: 20 a ship, d -> a ship, do
[Info] Input and Target by batch: 30 on't drum -> n't drum u
[Info] Input and Target by batch: 40 up people -> p people t
[Info] Input and Target by batch: 50 together t -> ogether to
[Info] Input and Target by batch: 60 o collect -> collect w
[Info] Input and Target by batch: 70 wood and d -> ood and do
[Info] Input and Target by batch: 80 on't assign -> n't assign
[Info] Input and Target by batch: 90 n them tas -> them task
[Info] Input and Target by batch: 100 ks and wor -> s and work
[Info] Input and Target by batch: 110 k, but rat -> , but rath
[Info] Input and Target by batch: 120 her teach -> er teach t
[Info] Input and Target by batch: 130 them to lo -> hem to lon
[Info] Input and Target by batch: 140 ng for the -> g for the
[Info] Input and Target by batch: 150 endless i -> endless im
[Info] Input and Target by batch: 160 mmensity o -> mensity of
[Info] Input and Target by batch: 170 f the sea. -> the sea.
```

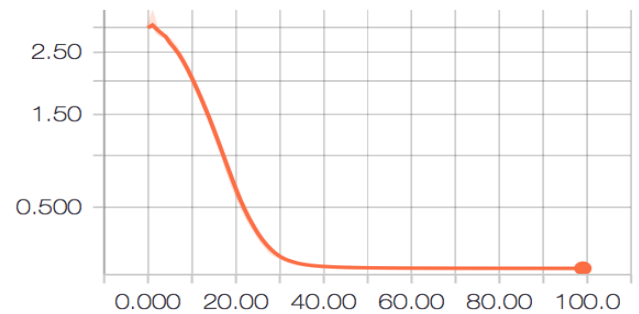
### 2. The Change of a Convergence of Loss by LSTM Level

With one LSTM level (hyperparameters - data size: 181, vocabulary size: 26, hidden size (# of neurons): 26, # of output classes: 26, sequence length: 10, moving window: 10, batch size: 18, epoch: 100), in the 43 of epoch, the loss

started to converge as 0.068 and reached to 0.060 in the 100 epoch in the <Figure3>.

<Figure3>

■ Mean\_loss



With the model of <Figure3>, the output in the <Figure4> was generated. Please compare the input sentence in the <Figure1> with <Figure4> that is almost similar to it.

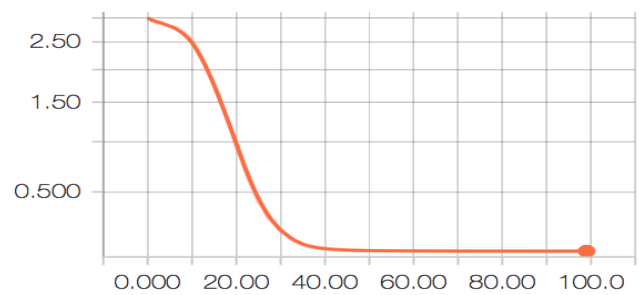
<Figure4>

```
=====Prediction=====
f you wantoto build a ship, don't arum up people together ton
collect wood and don't assign them task, and work, but rather
teach toem to lon for the andless immensity of the sea.
```

With two LSTM levels and the same hyperparameters, in the 44 of epoch, the loss started to converge as 0.068 and reached to 0.060 in the 100 of epoch shown in <Figure5>. The prediction on the model of the two LSTM level, shown in <Figure6>, was very similar to the one LSTM level model.

<Figure5>

■ Mean\_loss



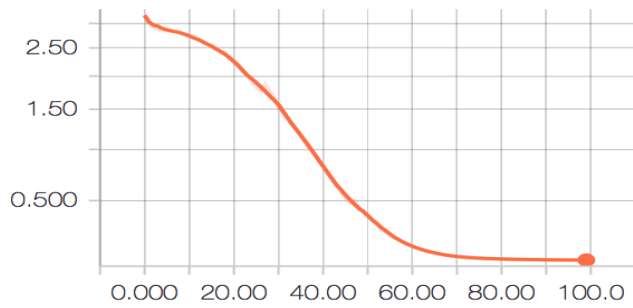
<Figure6>

```
=====Prediction=====
f you want to build a ship, don't arum up people t ge
ther toncollect wood and don't assign them tasks and
works but rather teach t em to lon for the andless i
mmensity of the sea.
```

With three LSTM levels and the same ones, in the 83 of epoch, the loss did to converge as 0.069 and did to 0.06 in the 100 of epoch shown in <Figure7>. The <Figure8> shows the prediction on the model with three LSTM levels.

<Figure7>

## Mean\_loss



<Figure8>

```
=====Prediction=====
f you want to build a ship, don't arum up people thgether toncol
lect wood and don't assigngthem tasks and works but rather teach
them to long for the andless immensity of the sea.
```

For details, refer to the full resulting logs in the  
result/result\_1\_10\_10\_100.txt (1 LSTM),  
result\_2\_10\_10\_100.txt (2 LSTMs) and  
result\_3\_10\_10\_100.txt (3 LSTMs)

### 3. Analysis

With the very simple text, 181 characters using the same hyperparameters, one LSTM level and two LSTM ones did not mostly affect the loss as epoch increases. However, three LSTM level was slow for the loss to converge rather than others.

## G. Reference

[1] Sung Kim, DeepLearningZeroToAll  
(2016), <https://github.com/hunkim/DeepLearningZeroToAll>  
,lab-12-4-rnn\_long\_char.py