# HW3 DL REPORT

0780819

RASHID ALI

# 1. Standard RNN

## 1.1 Experiment#1

Network architecture

Input layer size = (16,68)
Batch size= 16
Hidden size =14
Sequence length = 1088
Number of unrollings = 10
Output size= 14

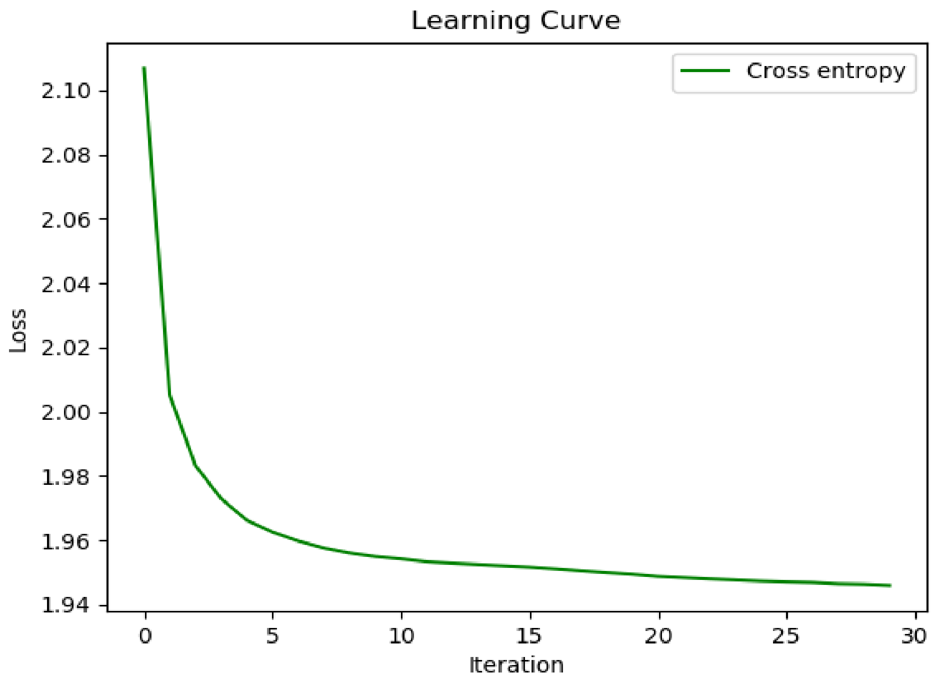
$X_t \rightarrow$ Input Layer $\rightarrow$ Hidden Layer(14) $\rightarrow$ Output layer $\rightarrow$ Flatten output $\rightarrow$ Softmax $\rightarrow$ $Y_t$-hat $\leftarrow$ $Y_t$

Learning Rate = 0.002

Model = RNN

Number of Epochs = 30

Learning Curve



,

Training Error Rate: 1.946

Validation Error Rate: 1.996

## 1.2 Output at Break Points

1. Break Point Execution at epoch number: 5

The input sample text from training text:

Before we proceed any fu

Output text is:

Before we proceed any funt coune, in freak s

2. Break Point Execution at epoch number: 10

The input sample text from training text:

re we proceed any fu

Output text is:

re we proceed any fue whorele here appos

3. Break Point Execution at epoch number: 15

The input sample text from training text:

proceed any fu

Output text is:

proceed any furcape my gome tomtes

4. Break Point Execution at epoch number: 20

The input sample text from training text:

eed any fu

Output text is:

eed any furpeen mart wifage he

5. Break Point Execution at epoch number: 25

The input sample text from training text:

ny fu

Output text is:

ny fure Row

The gmid, wa

## 1.3 Experiment#2

Network architecture

Input layer size = (16,100)
Batch size= 16
Hidden states size =30
Sequence length = 1600
Number of unrollings = 10
Output size= 30

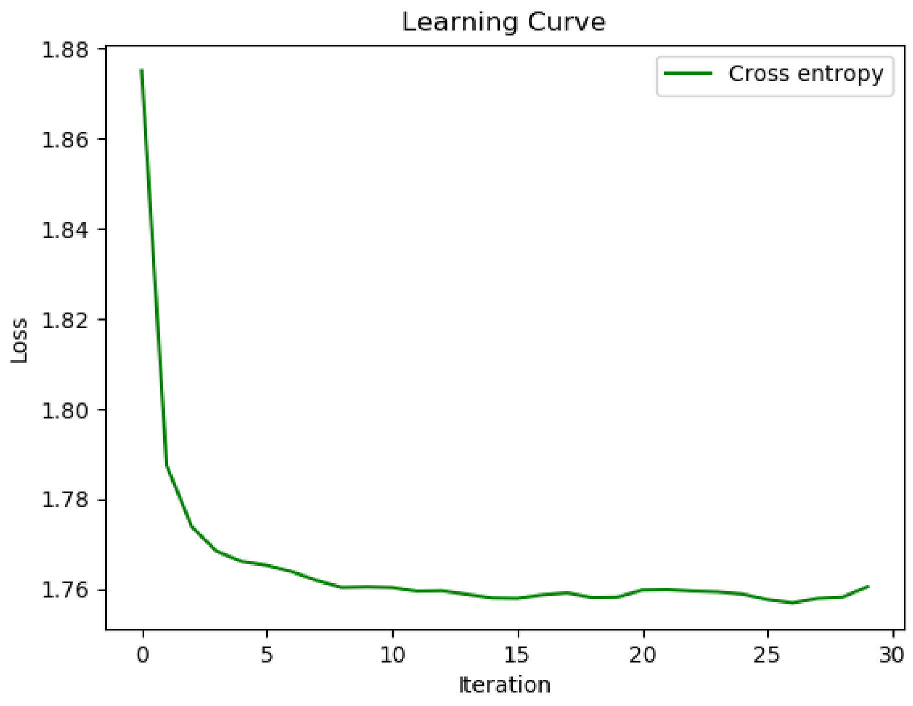
$X_t$→Input Layer→Hidden Layer(30)→Output layer→ Flatten output→Softmax→$Y_t$-hat← $Y_t$

Learning Rate = 0.002

Model = RNN

Number of Epochs = 30

Learning Curve



Training Error Rate: 1.761

Validation Error Rate: 1.831

### 1.4 Output at Break Points

1. Break Point Execution at epoch number: 5

   The input sample text from training text:

   Before we proceed any fu

   Output text is:

   Before we proceed any funch and sust, Not n

2. Break Point Execution at epoch number: 10

   The input sample text from training text:

   re we proceed any fu

   Output text is:

   re we proceed any full your seed, shear,
3. Break Point Execution at epoch number: 15

   The input sample text from training text:

    proceed any fu

   Output text is:

   proceed any fulls tweepence the ma
4. Break Point Execution at epoch number: 20

   The input sample text from training text:

   eed any fu

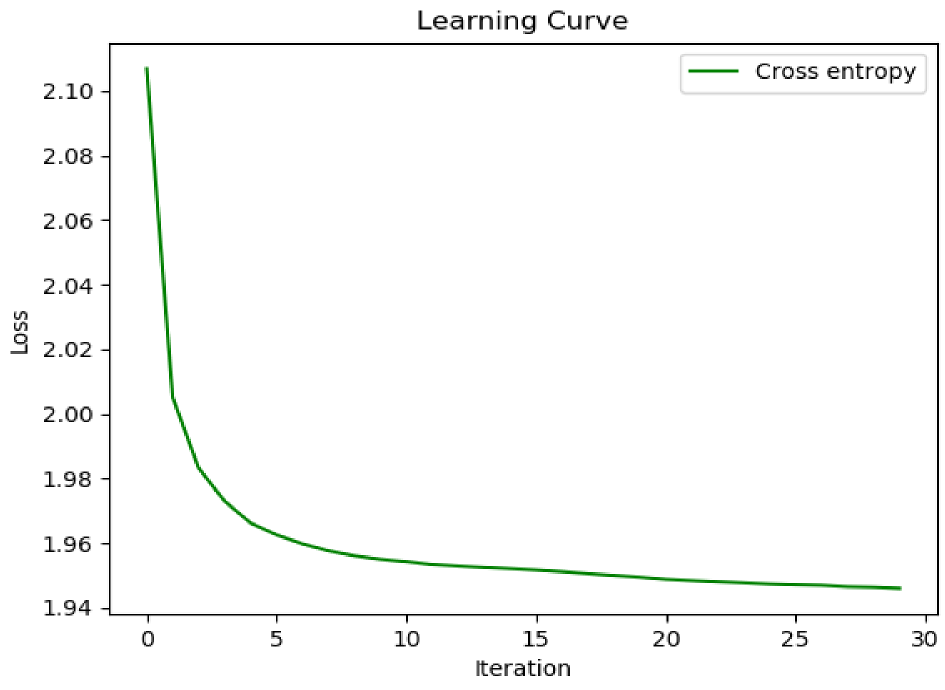
   Output text is:

eed any full; fear agutuss bou

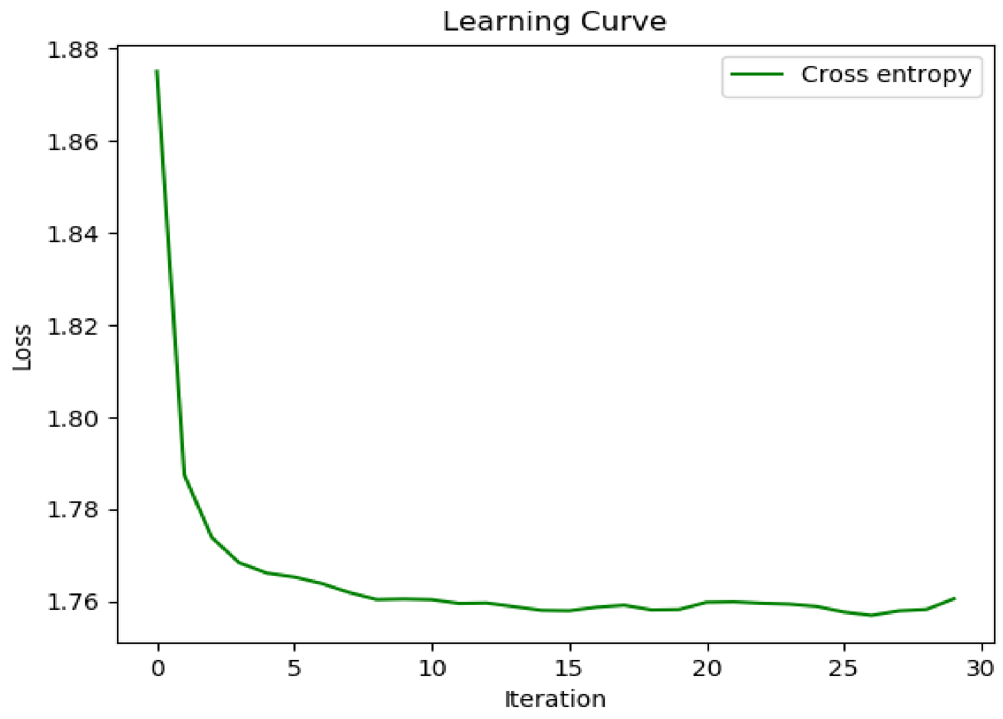
5. Break Point Execution at epoch number: 25
   The input sample text from training text:
   ny fu
   Output text is:
   ny fued blouchey The soo

## 1.5 Comparison of results

1. Hidden size =14, Sequence length = 1088, Model parameter= 7292



2. Hidden size =30, Sequence length = 1600, Model parameter = 31948



The training loss of RNN-2 is optimized to 1.76 when the number of hidden states = 30, sequence length = 1600 and the training loss of RNN-1 is optimized to 1.94 when the hidden state size = 14, sequence length = 1088. Clearly, the model parameters (neurons) are increased when the number of hidden states and sequence length is increased. It means that the capacity of model has increased to remember more information which helped to optimize the training loss when the number of hidden states increased. The output text sequences are longer enough to make some sense but still needs more training to output sensible text.

## 2. LSTM

### 2.1 Experiment#1

Network architecture

Input layer size = (16,68)
Batch size= 16
Hidden size =10
Sequence length = 1088
Number of unrollings = 10
Output size=10

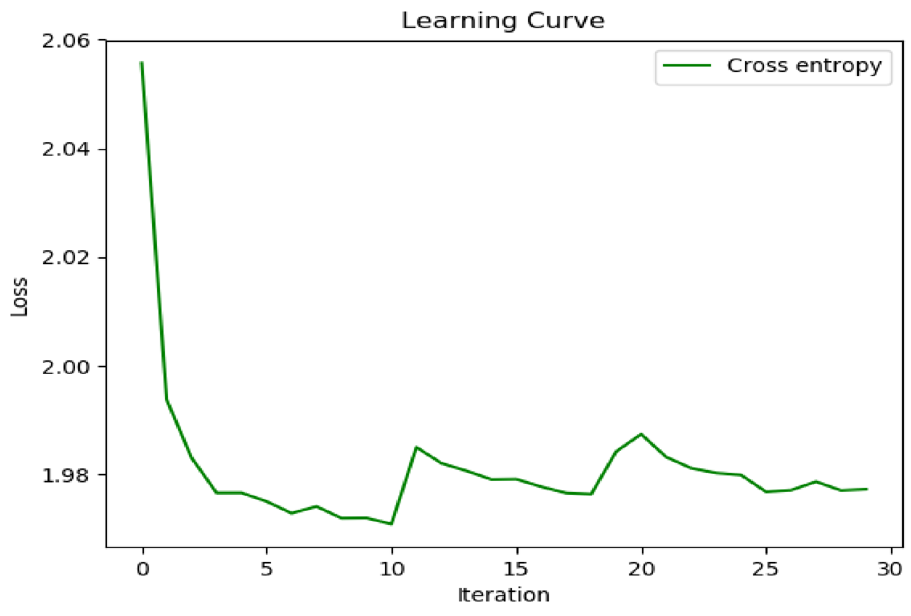
$X_t \rightarrow$ Input Layer $\rightarrow$ Hidden Layer(10 LSTM cell) $\rightarrow$ Output layer $\rightarrow$ Flatten output $\rightarrow$ Softmax $\rightarrow Y_t$-hat $\leftarrow Y_t$

Learning Rate = 0.002

Model = RNN

Number of Epochs = 30

Learning Curve



Training Error Rate: 1.977

Validation Error Rate: 2.051

## 2.2 Output at Break Points

1. Break Point Execution at epoch number: 5

   The input sample text from training text:

   Before we proceed any fu

   Output text is:

   Before we proceed any fuch?

   Make so't aor m
2. Break Point Execution at epoch number: 10

   The input sample text from training text:

   re we proceed any fu

   Output text is:

   re we proceed any funcie the lads het dh
3. Break Point Execution at epoch number: 15

   The input sample text from training text:

   proceed any fu

   Output text is:

   proceed any funpefesbellare?
4. Break Point Execution at epoch number: 20

   The input sample text from training text:

   eed any fu

Output text is:

eed any funle? VUCUNUS: Rh

5. Break Point Execution at epoch number: 25

   The input sample text from training text:

   ny fu

   Output text is:

   ny fur hard.  FANCET:

### 2.3 Experiment#2

Network architecture

Input layer size = (16,100)
Batch size = 16
Hidden size = 30
Sequence length = 1600
Number of unrollings = 10
Output size= 30

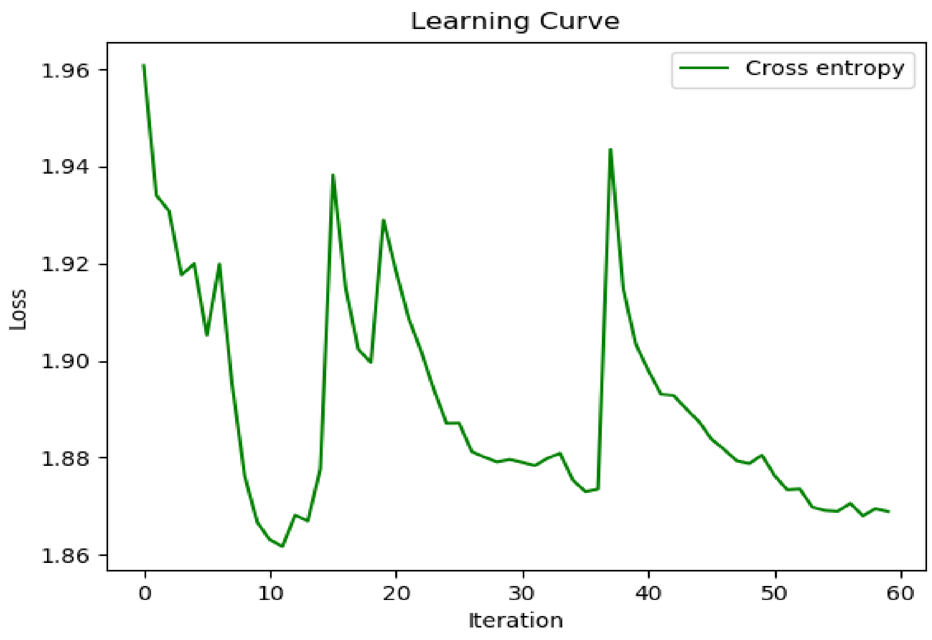
$X_t$→Input Layer→Hidden Layer(30 LSTM cell)→Output layer→ Flatten output→Softmax→$Y_t$-hat← $Y_t$

Learning Rate = 0.002

Model = LSTM

Number of Epochs = 60

Learning Curve



Training Error Rate: 1.869

Validation Error Rate: 1.939

## 2.4 Output at Break Points

1. Break Point Execution at epoch number: 35

   The input sample text from training text:

   , hear me speak. All:

   Output text is:

   , hear me speak. All: Sale; You wadana,

2. Break Point Execution at epoch number: 40

   The input sample text from training text:

   r me speak. All:

   Output text is:

r me speak. All: I worce Belad may

3. Break Point Execution at epoch number: 45

   The input sample text from training text:

   speak.All:

   Output text is:

   speak. All: Mad cofmort your pe

4. Break Point Execution at epoch number: 50

   The input sample text from training text:

   . All:

   Output text is:

   .All: Hilers of our ping.

5. Break Point Execution at epoch number: 55

   The input sample text from training text:

   All:

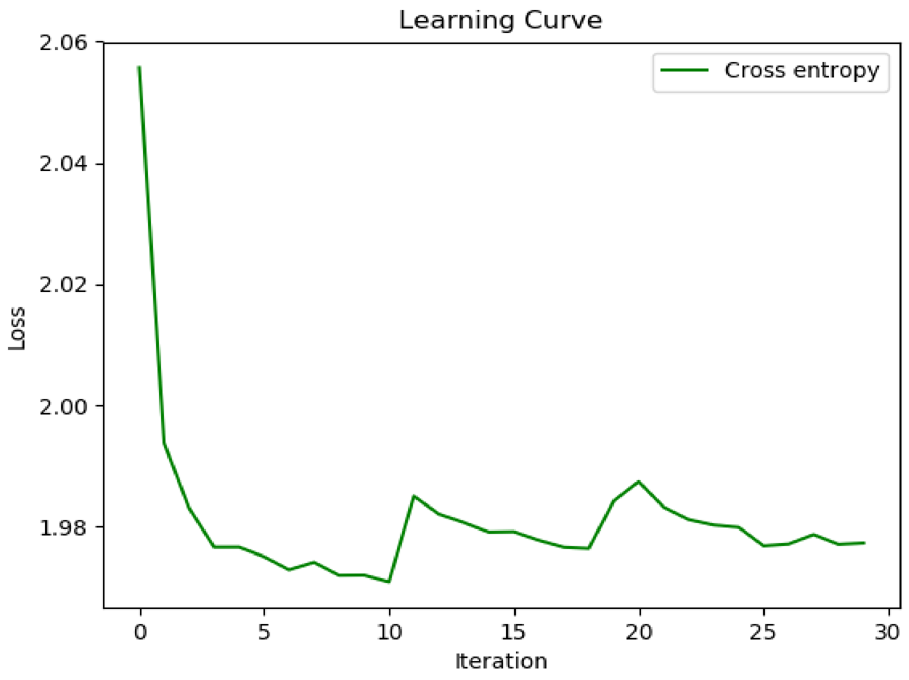
   Output text is:

   All:

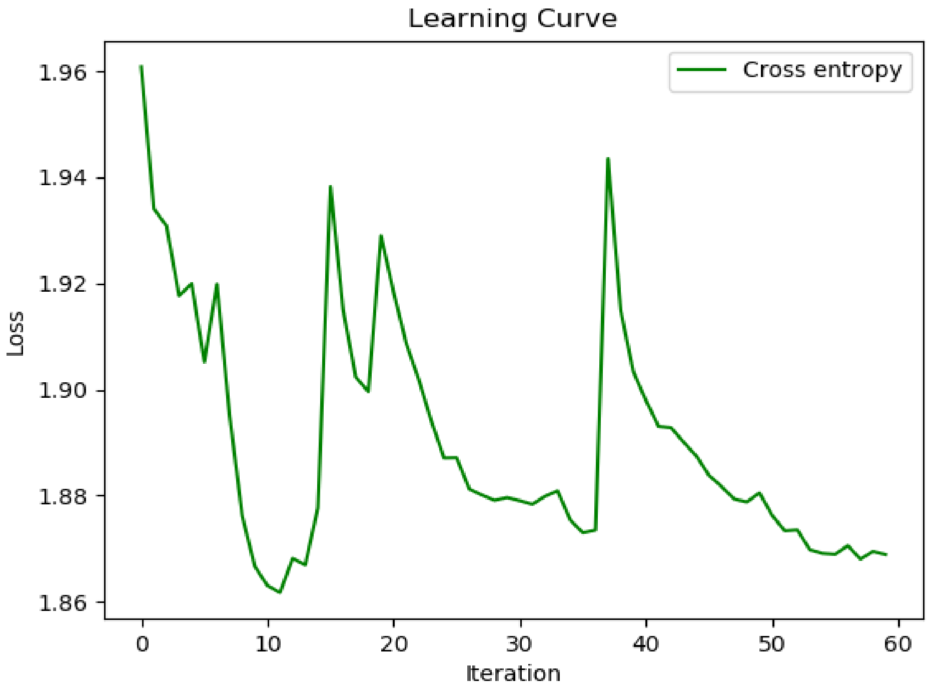
   Now, as it him, thy

## 2.5 Comparison of results

1. Hidden size =10, Sequence length = 1088, Model parameter= 4767



2. Hidden size =30, Sequence length = 1600, Model parameter = 31948



The training loss of LSTM is optimized to 1.869 when the number of hidden states = 30, sequence length = 1600 and the training loss of LSTM is optimized to 1.939 when the hidden state size = 10, sequence length = 1088. The text generated at the last break point of LSTM-2 training makes more sense than that of LSTM-1 and standard RNN. The reason is that LSTM-2 is deeper network with 30 LSTM cells. It has more capacity to remember longer sequences which helps to generate sensible text. Therefore, LSTM-2 performance is better than LSTM-1 and RNN.

## 3. Words Generation using LSTM

Input text: “JULIET”

Generated text is: JULIETZE: If to logats! what:

The let's, applardes; Spendeers is queter? why, sear as a his a are for h

Input text: “JULIET”

Generated text is: JULIETU: My nothsens

Input text:

spirit of love!

Generated text is: spirit of love! Lonk your, all sshews?

Swell, I binst, my the maingnus. I quidood Wias his wrong, friarArA ESEUN

## References

Ideas, concepts and some code blocks are used from following links to accomplish the tasks of this homework.

http://adventuresinmachinelearning.com/recurrent-neural-networks-lstm-tutorial-tensorflow/

https://www.tensorflow.org/tutorials/sequences/text_generation

https://github.com/sherjilozair/char-rnn-tensorflow

https://github.com/martin-gorner/tensorflow-rnn-shakespeare

https://github.com/crazydonkey200/tensorflow-char-rnn

https://github.com/pangolulu/char-rnnlm-tensorflow