Part 3 : Character-level Recurrent Neural Network (RNN).

1. Find and Load Training Text :

(a) Choose the text you want your neural network to learn, but keep in mind that your data set must be quite large in order to learn the structure! RNNs have been trained on highly diverse texts (novels, song lyrics, Linux Kernel, etc.) with success, so you can get creative. **Gutenberg Books** is a source of free books where you may download full novels in a .txt format. Else get the book from the link given

(b) We will use a character-level representation for this model. To do this, you may use extended ASCII with 256 characters. As you read your chosen training set, you will read in the characters one at a time into a one-hot-encoding, that is, each character will map to a vector of ones and zeros, where the one indicates which of the characters is present: char → [0, 0, · · · , 1, · · · , 0, 0] Your RNN will read in these length-256 binary vectors as input.

2. Implement an RNN (without using inbuilt libraries):

(a) **Backpropagation Through Time :** In an RNN with a single hidden layer, you should have three set of weights: W_{xh} (from the input layer to the hidden layer), W_{hh} (the recurrent connection in the hidden layer), and W_{ho} (from the hidden layer to the output layer). Suppose you use Softmax units for the output layer and Tanh units for the hidden layer, show:

i. Write the equation for the activation at time step t in the hidden layer and the equation for the output layer in the forward propagation step.

ii. Write the equation for the weight update rule at time step t for W_{xh} , W_{hh}, and W_{ho} in a vectorized notation. Suppose the backpropagation goes back to time step k ($0 < k < t$).

(b) **Network Training**: Train your recurrent neural network using the dataset you created in 1(b). You are free to choose learning parameters (sequence length, learning rate, stopping criteria, etc.). Complete the following task:

i. Report your training procedure. Plot the training loss vs. the number of training epochs.

ii. During training, choose 5 breaking points (for example, you train the network for 100 epochs and you choose the end of epoch 20,40,60,80,100) and show how well your network learns through time. You can do it by feeding in the network a chunk of your training text and show what is the output of the network. Report your Result.

(c) **Experiment with Network Structure:** As before, we want to explore how the network learns when we change parameters:

i. **Number of hidden units**. Try doubling and halving your number of hidden units. Like in 2(b), plot the training loss vs. the number of training epochs and show your text sampling results. Discuss your findings.

ii. **Sequence length**. Try doubling and halving your length of sequence that feeds into the network. Like in 2(b), plot the training loss vs. the number of training epochs and show your text sampling results. Discuss your findings.

https://drive.google.com/open?id=125fLHzCypUMpM72CSgw_BW59h48zKiYI