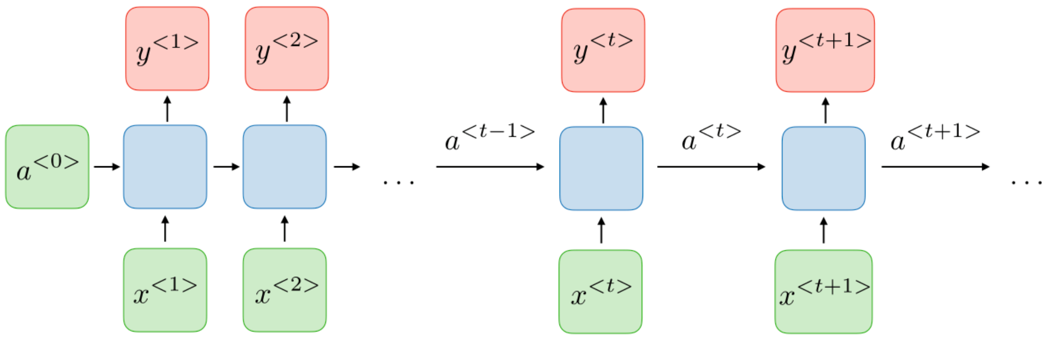
Report OF Assignment Three

RNN for text generation on ShakespeareDataset



1 Introduction

Recurrent Neural Networks are Networks specialized in training sequences of points such as time series, sequence of characters in a text, sequence of speech in an audio, etc.. In this assignment I am trying to implement a network that takes charachters of different texts and try to generate following texts after training the network with a measure of efficiency.

1.1 Data Processing

to have a network compatible input some preprocessing should be performed over the data, The main function in the network Long Short term memomry -will be explained shortly- expects one hot encoded values which are basically values going through a process that converts categorical variables into variable understood by the algorithm and in our case the characters are converted into integers then converted into a column vector where only it's corresponding integer index will have the value of 1 and the rest of the vector will be filled with 0's.

2 Methods

2.1 Network

Network Description: The charachter Recurrent Neural Network takes charachter as inputs to be trained and generates accordingly starts as follows, It has few inputs, charachters, number of hidden units, number of layers, drop probability and learning rate. First dictionaries are created from charcher to integer and vice verse to be used in the encoding process, then the Long short-term memory function is defined to take characters, hidden units, probability and learning rate as input to process and classify the input then try to predict the following text, Followed by the forward function that passes inputs through the network and finally the weights initialization function.

2.2 Training Function

Then to train the network I find it preferable to define a training function so that it's easier if one needs to loop over different values of parameters. This function contains pytorch optimizer and pytorch loss function which is cross Entropy then for each epoch the previously defined one hot encoder function takes in the data and turns them into torch tensor that is fed to the network then back-propagation is done to calculate error.

2.3 Predicting text

an interesting aspect of the Recurrent neural network is the text generation thus a function to predict text is added to the code, this function is composed of: soft max probabilities according to the data -which are as previously explained hot encoded-, then sampling according to these probabilities and the first element of the training text.

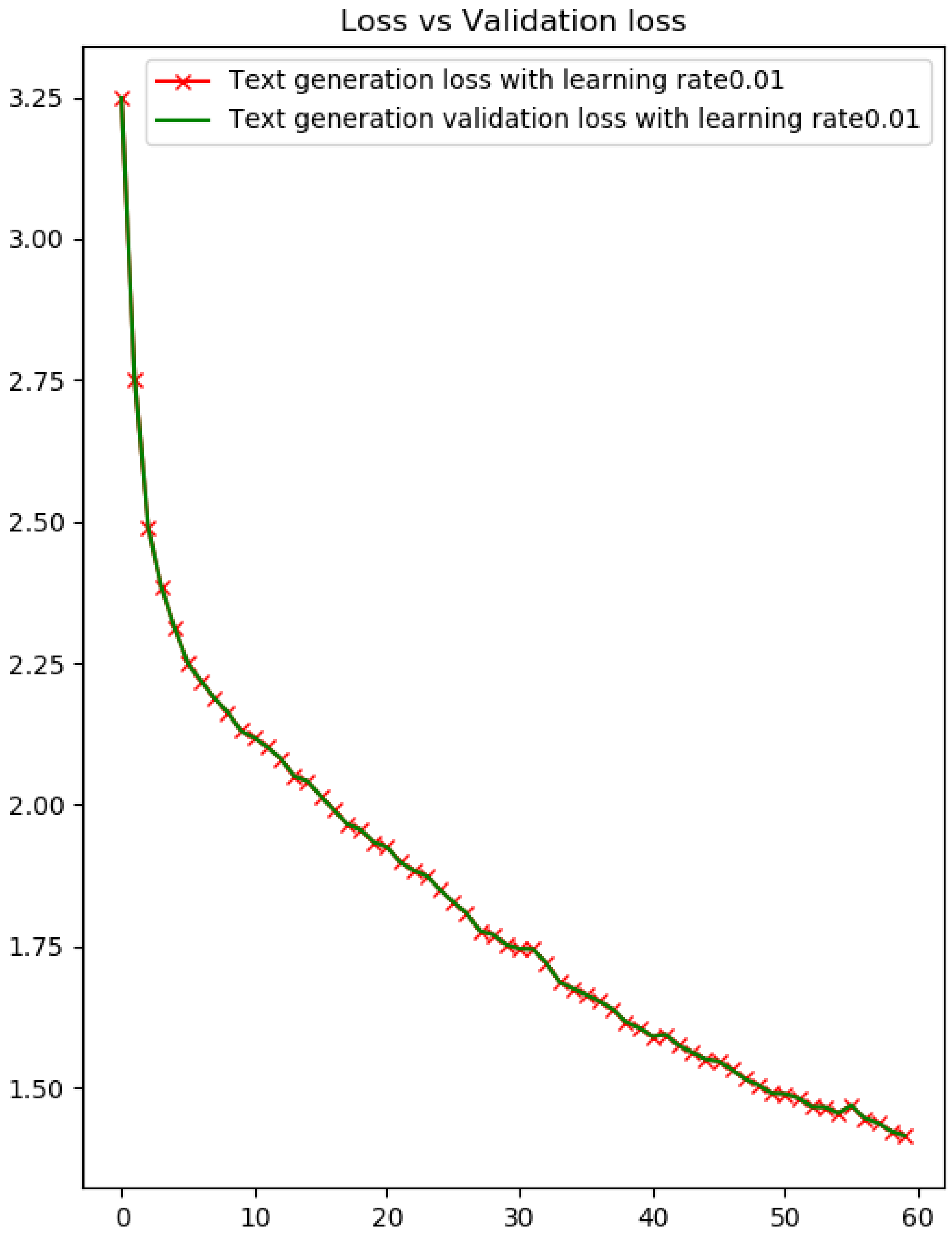


Figure 1: Loss plots vs validation loss with learning rate 0.01

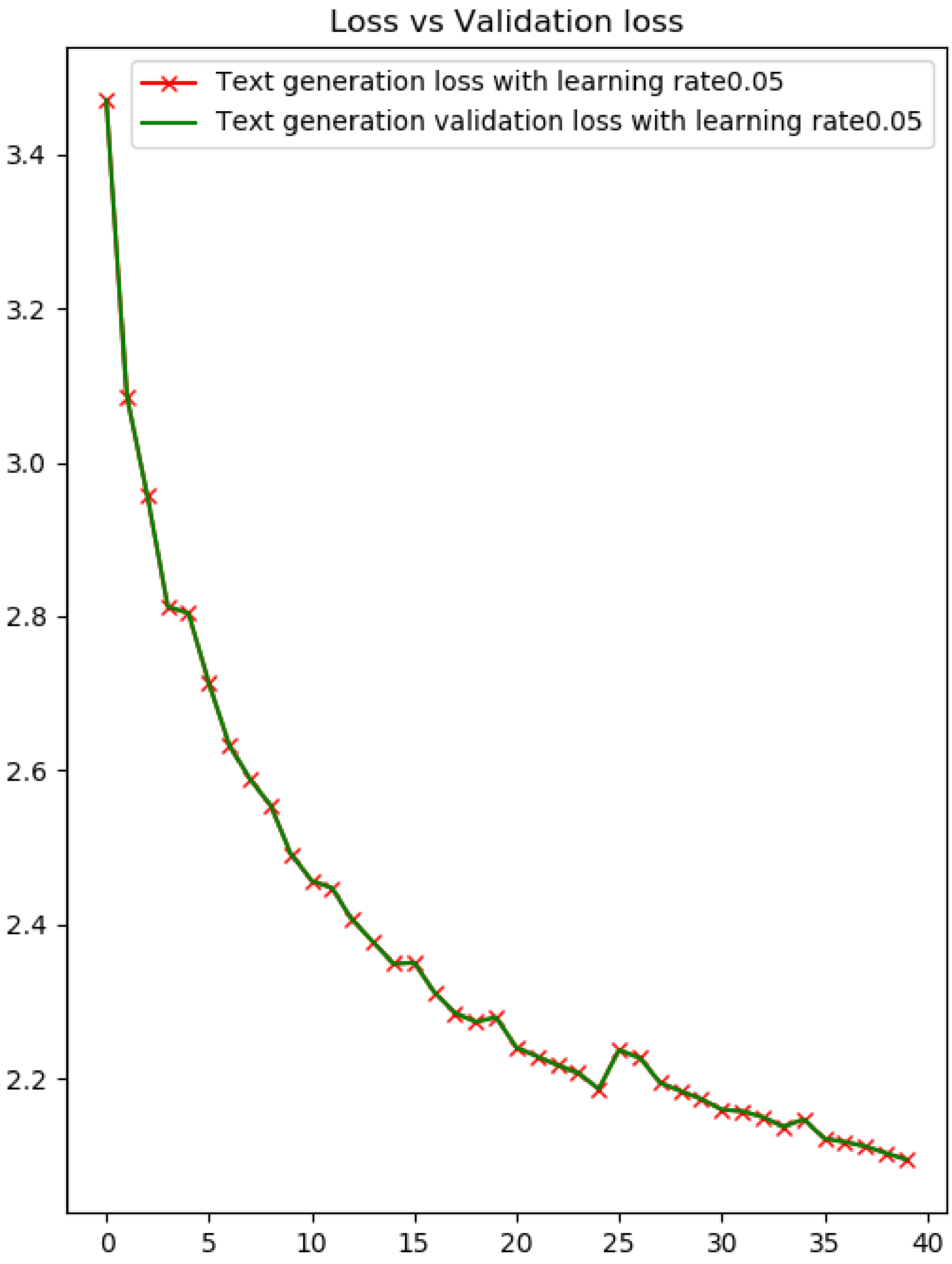


Figure 2: Loss plots vs validation loss with learning rate 0.05