Specification Document

Modeling of Generated Data from Text Documents

Problem Statement:

Deep generative modeling of text documents using GANs and NLP.

Introduction:

In this modern day and age, a large amount of data is collected every day. With this ever-increasing data, it has become more and more difficult to manually process the data and get the desired information. Topic modeling provides us with a method to organize, understand, and summarise large amounts of data. It does this by finding various patterns in the given data. In this project, our main goal is to create a GAN that can successfully classify documents based on generated text.

Dataset:

The dataset used is the 20newsgroups which comprises of 18828 newsgroups posts on 20 topics. All the messages on the newgroups were merged under their respective titles and all the titles from the 20 different topics were merged into a single file. This file was used as the dataset for the model. The dataset consists of two columns: Label and Document

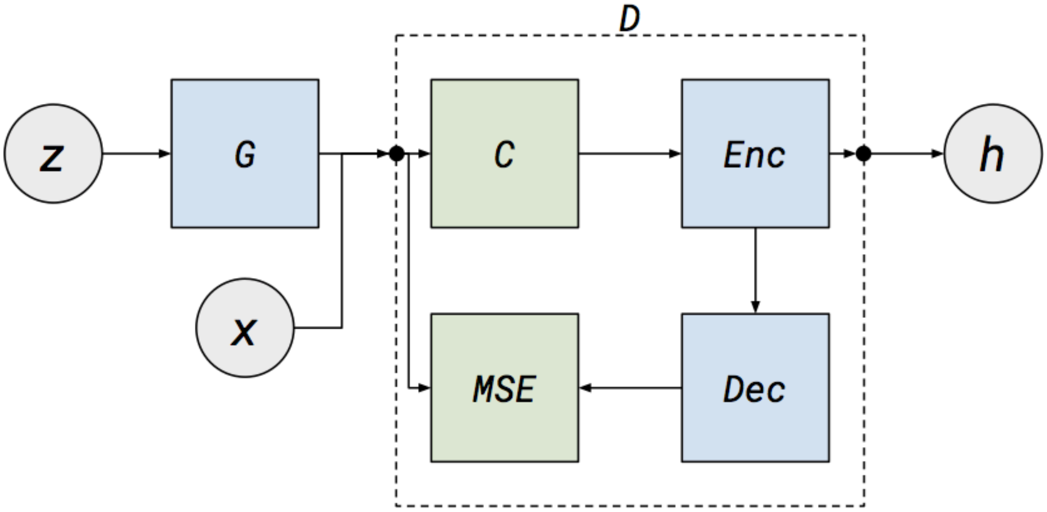
Data preprocessing:

Since the data contained a lot of commonly used words like prepositions and articles, these words had to be removed. After that, the text was converted to lowercase for better efficiency. The text was then tokenized, stemmed, lemmatized. The final step of the preprocessing was converting the result of all the previous steps into a vector and arguments were passed into the program.

Modelling:

Initially, the vectors are masked, gradients are defined, and gradient norm scaling is done. After this, the generator and discriminator are defined.

The generator contains two fully connected ReLU layers and a final sigmoid layer. The discriminator has one leaky ReLU layer and it linearly maps input vector to input space. The basic representation of the full model is shown below:

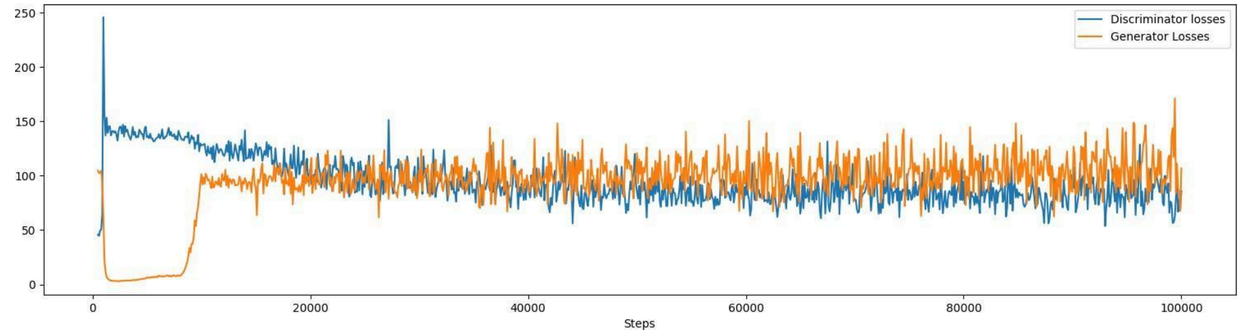


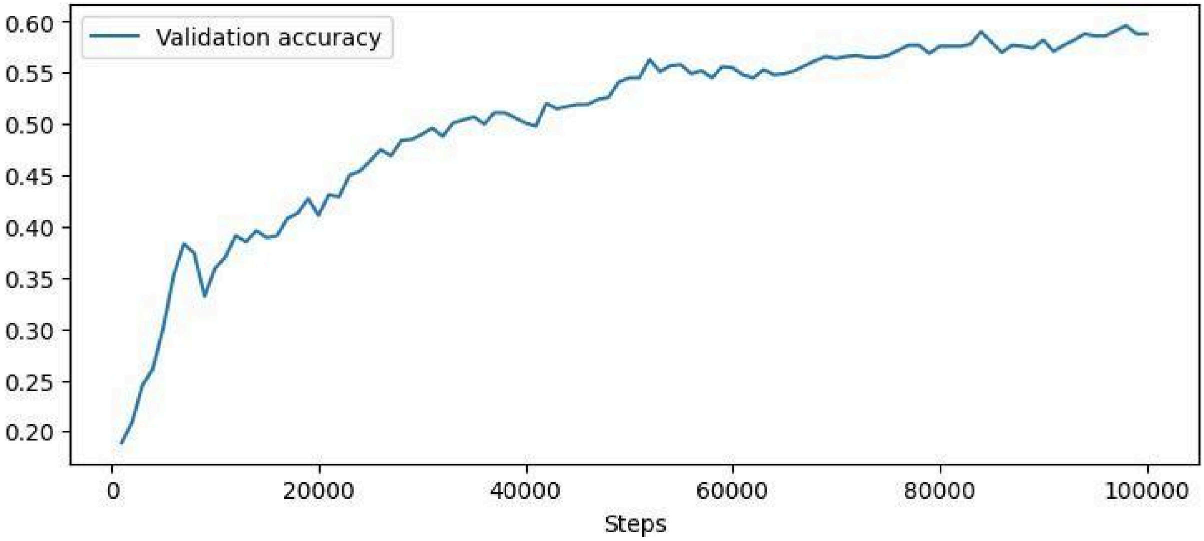
Z is a noise vector, which is passed through the generator G and produces a vector which is the size of the vocabulary. We then pass either this generated vector or a sampled bag of words vector from the data(x) to our denoising autoencoder discriminator D. This vector is masked with noise C, mapped into lower dimensional space by the encoder, mapped back into the data space by the decoder and then finally the loss is taken as the mean squared error between the input to the discriminator and the reconstruction. We can also have the encoded representation (h) for any input document.

Training:

For training, the input dataset and model output directories are passed to the train function. The training data is first divided into mini batches and these mini batches are used for training. Two copies of the generator are created with one network taking the real sample as input from a mini-batch and the other taking the generated samples as input. The update to the discriminator and generator is done separately and at each update, we generate a new noise vector to pass to the generator, and a new noise mask for the denoising autoencoder (the same noise mask is used for each input in the batch). The training was carried on for 10000 steps.’

Training validation and evaluation:





The final validation accuracy at the end of the training was 59.6%. The testing accuracy was 57%.

Future Improvements:

- One of the improvements that could be made is to select a better dataset to improve the accuracy.
- Instead of using batch normalization, spectral normalization can be used which can stabilize the training of our discriminator.

Mentors:

- Agrim Agrawal
- Mohit Goyal

References:

- https://www.ibm.com/cloud/learn/data-modeling#:~:text=Data%20modeling%20is%20the%20process,between%20data%20points%20and%20structures
- https://arxiv.org/abs/1612.09122
- https://arxiv.org/abs/1511.06434
- https://arxiv.org/abs/1609.03126
- https://youtube.com/playlist?list=PLhhyoLH6IjfwIp8bZnzX8QR30TRcHO8Va