Spam or Not Spam Data Generation using LSTM-VAE

Tejas Asija[1] and Deevyansh Khadria[2]

[1]Department of Mathematics ,Indian Institute of Technology, Delhi
[2]Department of Electrical Engineering ,Indian Institute of Technology, Delhi

November 24, 2024

Abstract

This project focuses on generating synthetic spam and not-spam text data using advanced deep learning models. By leveraging a Long Short-Term Memory (LSTM) based Variational Autoencoder (VAE), it aims to learn and replicate patterns from the "Spam or Not Spam Dataset" on Kaggle. The process involves preprocessing the dataset, building an LSTM VAE with a Gaussian latent space, and training it to reconstruct input text while learning meaningful representations. The quality of generated synthetic data will be evaluated using reconstruction accuracy (by the GNB Classifier) and KL divergence loss. This Project discusses all the architecture, training procedure, evaluation results.

Keywords
LSTM, VAE, GNB

1 Introduction

This project focuses on generating synthetic spam and not-spam text data. It is very beneficial in the cases when the dataset size is small, generating synthetic data increases the dataset size, which helps in training more robust machine learning models. Synthetic data adds variability and diversity to the dataset, allowing the training models to generalize better.Models trained on larger and more diverse datasets are less prone to overfitting. The project uses the LSTM based VAE to learn a probabilistic latent space of the dataset to enable controlled and diverse text generation using the decoder.This synthetic data can augment the original dataset, mitigating issues such as class imbalance, limited data availability, and privacy concerns.

2 Materials & Methods

The dataset is taken from the Use the "Spam or Not Spam Dataset" from Kaggle labeled as spam or not spam. It consist of the 3000 emails(2500 Spam, 500 Ham) making the dataset imbalance.Missing enteries were taken care of.

Converted the text to lower case and removed the stop words(eg. "the","is") using the NLTK. Tokenized the emails using the keras Tokenizer. The special characters has been removed. The NLTK based **stemming** is done to consider "running", "run" , "ran" as the similar words.

The text data is tokenized using the **Keras Tokenizer**, which maps each word in the dataset to a unique numerical index, effectively transforming the text into a sequence of integers.These integer sequences are then mapped to dense vector representations using the **GloVe (Global Vectors for Word Representation)** embedding. This step converts the tokenized words into fixed-dimensional, pre-trained word embeddings, where each word is represented as a probability distribution in a continuous vector space.

3 Model Architecture

3.1 LSTM based VAE

The purpose of the VAE is to learn the compressed latent representation of the email embeddings and reconstruct them.

- Encoder LSTM: (64+32) hidden dimensions.
- Mean Layer (Z mean): Dense layer outputing the latent vector mean.

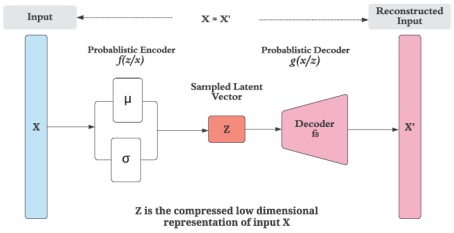

Figure 1: LSTM architecture

- Log Variance(Z sigma):Dense layer outputing the log variance for the latent space.
- Latent Space dimension: 16
- Sampling Layer : Used Reparametrization trick to sample latent vectors.
- Decoder: (32+64) hidden dimension
- Reconstruction Loss: Mean squared error (MSE) between the input and reconstructed embeddings.
- KL Divergence loss: Regularized the latent space using the Kullback-Leibler divergence
- Combined Loss= KL Divergence loss+ Reconstruction Loss

  $L_{KL} = -\frac{1}{2}\sum_{i=1}^{n}\left(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2\right)$

The model is trained using the Adam optimizer with a learning rate of 0.003 and a batch size of 1 for 50 epochs. This setup ensures efficient training while encouraging generalization by penalizing large weights.

3.2 Classifier Naive Bayes

The model classifies emails using the synthetic "spam" and "not spam" dataset generated by the LSTM-based VAE, leveraging Gaussian Naive Bayes (GNB) for classification. The Naive Bayes classifiers, including Gaussian Naive Bayes (GNB), Multinomial Naive Bayes (MNB), and Bernoulli Naive Bayes (BNB), are trained on the original Kaggle dataset to evaluate the accuracy and performance. The classifiers model the conditional probability distributions of the features given the class labels, with each variant utilizing different assumptions.

3.3 Training Environment

For testing, cretead a new virtual environment and install the following required versions of required libraries:
Numpy: 1.19.2
Pandas: 1.1.4
Genism Verison: 4.0.1
NLTK Version: 3.6.5
Tensorflow version: 1.10.0
Kera version: 2.3.2

3.4 HyperParameters Tuning:

Max len: Maximum allowed length for the email. Emails shorter than this length are padded, while those longer are truncated.=200
N dim: The dimension of the embedding vector.=100
Intermediate dim: The number of LSTM units (cells) in the layer=(64+32)
Z-dim: The size of the latent space dimension= 16

4 Results

4.1 LSTM-VAE:

- **Reconstruction Loss** (0.1231) measures the difference between the original and reconstructed data, indicating how well the model captures the data's structure.
- **KL Divergence** (0.032) quantifies the divergence between the learned latent distribution and the prior Gaussian distribution, ensuring smoothness in the latent space.
- **Reconstruction Accuracy** (0.8769) reflects the percentage of correctly predicted outputs, indicating how effectively the model reconstructs the original input data.

4.2 Classifier Accuracy:

- **Accuracy** (0.926) represents the overall proportion of correctly classified samples.
- **Precision** (0.846) measures the proportion of true positive predictions among all positive predictions.
- **F1 Score** (0.814) is the harmonic mean of precision and recall, balancing the two metrics for an overall measure of performance.

The classifier performed well in identifying spam emails, indicating that reconstructed embeddings from the VAE preserved meaningful semantic information.

5 Synthetic Data Generation

The VAE generates synthetic data by sampling from its latent space and decoding latent vectors. The quality of the generated data is assessed using reconstruction loss and KL diver-

gence, which indicate meaningful latent representations. Further validation can be done by classifying the generated embeddings using the LSTM classifier, with successful alignment to spam or non-spam categories suggesting high-quality data. Future evaluations may include qualitative inspection of generated sentences by mapping embeddings back to words.

6 Discussion

6.1 Advantages

- **Variational Autoencoder (VAE):** It enables the generation of diverse, high-quality synthetic data that can augment the dataset and improve model generalization.
- **Long Short-Term Memory (LSTM):** LSTMs excel at capturing long-range dependencies in sequential data, making them ideal for processing and generating meaningful text sequences

6.2 Limitations

- **Training Instability:** VAEs can sometimes suffer from training instability, especially when the balance between the reconstruction loss and KL divergence is not properly managed.
- **Limited Expressiveness of Latent Space:** The latent space in VAEs is constrained by the Gaussian prior, which can limit the model's ability to capture highly complex or multimodal distributions.

6.3 Future Works

- **Improved Latent Space Regularization:** Future work could involve experimenting with more advanced latent space priors (e.g., normalizing flows) to enhance the expressiveness of the latent space
- **Cross-Domain Data Generation:** Extending the VAE-LSTM model to generate synthetic data across different domains (e.g., text-to-image or multi-modal generation)

Conclusions

- Improving the word embedding and removing stop words, has a huge impact on the kullback divergence loss.
- VAE and LSTM can undestand any distribution due to temporal dependencies of the LSTM and gaussian probability interpetation of the VAE.

Acknowledgements

I would like to express my sincere gratitude to everyone who supported and contributed to the successful completion of this project. First and foremost, I would like to thank my supervisor, Ashwini, for their invaluable guidance and expertise.I also want to acknowledge the developers and contributors of the Keras and TensorFlow libraries for providing the tools and resources that made the implementation of the VAE-LSTM model possible.

References

[1] Use the "Spam or Not Spam Dataset" from Kaggle (https://www.kaggle.com/datasets/ozlerhakan/spam-or-not-spam-dataset/data)

[2] Building a Neural Network Zoo From Scratch: The Long Short-Term Memory Network ("https://medium.com/building-a-neural-network-zoo-from-scratch-the-long-short-term-memory-network-1cec5cf31b7")

[3] Building the VAE ("https://www.youtube.com/watch?v=h94ITdb$_p$7I")