NEURAL NETWORKS: PROJECT 3

Authors:
Marina Gómez Rey (100472836)
Ángela Durán Pinto (100472766)
María Ángeles Magro Garrote (100472867)

Introduction:
In the realm of generative modeling, the application of Variational Autoencoders (VAEs) has emerged as a promising avenue for synthesizing data that mirrors complex distributions. In this project, we embark on an exploration of VAEs within the context of generating synthetic data from a 3-dimensional Gaussian Mixture Model (GMM) characterized by multiple components spread across the 3D space. Our primary objective is to investigate the VAE's capability to capture the intricate structure of multi-modal distributions and produce samples that closely emulate the ground truth distribution.

By delving into the construction of a VAE architecture, where dense layers substitute Convolutional Neural Networks (CNNs) in both the generative decoder and inference encoding networks, we aim to unveil insights into its performance. Our methodology involves rigorous training and evaluation of the VAE using synthetic data, coupled with an in-depth analysis of its latent space representation through T-SNE visualization. This endeavor promises to shed light on the VAE's efficacy in capturing the nuances of multi-modal distributions and its potential applications in various domains requiring data synthesis.

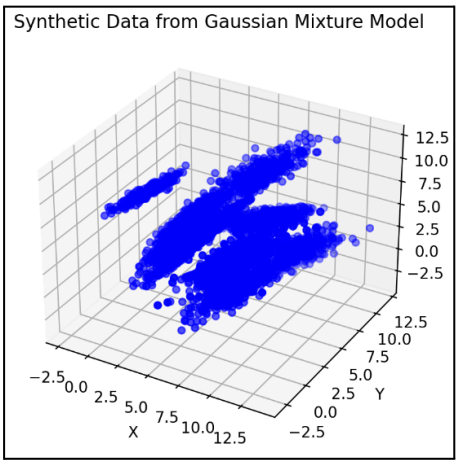
Generate Synthetic Data
In our project, the generation of synthetic data is pivotal for the evaluation and training of our model. To achieve this, we employ a Gaussian Mixture Model (GMM) methodology, a widely-used approach in statistical modeling and machine learning.

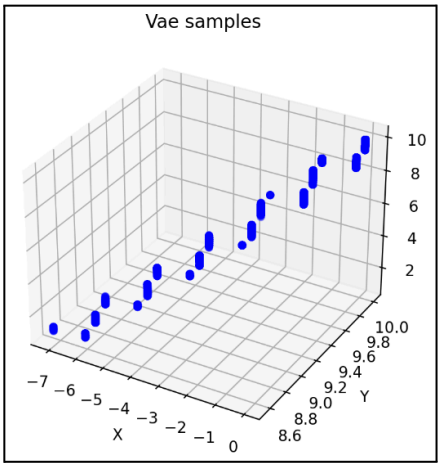
The process begins by defining the characteristics of the GMM components. Specifically, we randomly select the means for each component, representing the centers around which the data points will cluster. These means dictate the location of the data clouds within the multi-dimensional space.

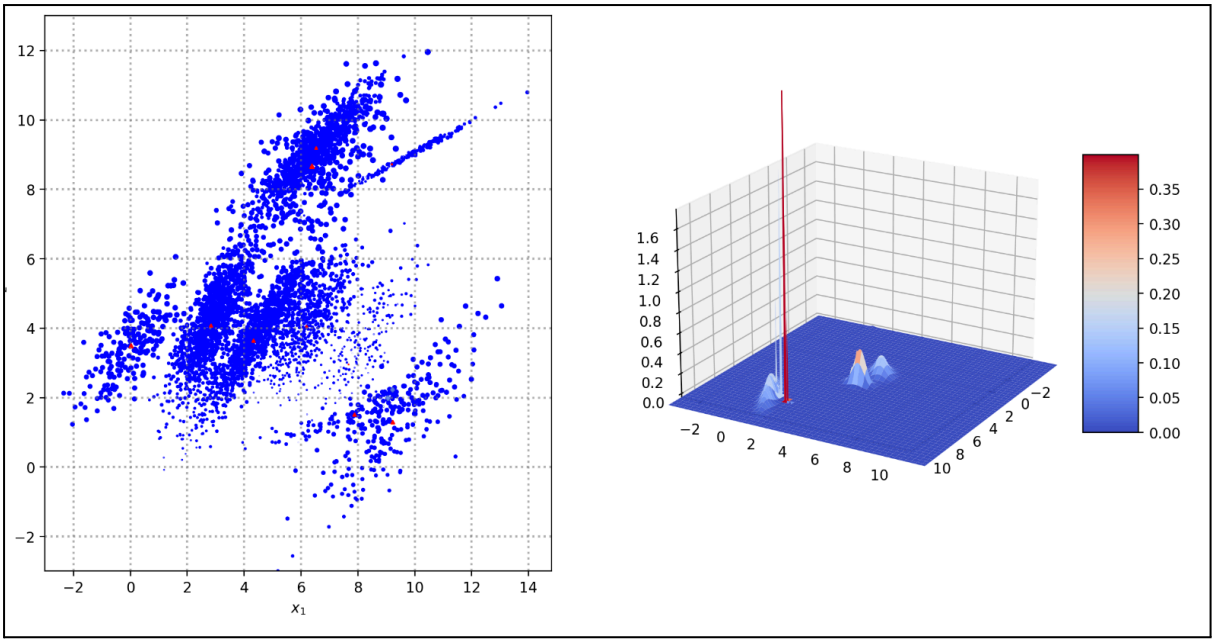
Next, we specify the covariance matrices for each component, defining the spread and orientation of the data points within each cloud. These matrices encapsulate the relationships between different dimensions of the data and play a crucial role in determining the shape and structure of the generated clusters.

Furthermore, to incorporate variability and randomness into our synthetic data, we introduce mixing coefficients. These coefficients determine the proportion of contribution from each component to the overall dataset. By adjusting these coefficients, we can control the relative importance of each cluster, thereby influencing the distribution and composition of the generated data.

By iteratively combining these components, incorporating random variations, and ensuring mathematical coherence in the distribution of data points, we construct a synthetic dataset that closely mirrors the characteristics of real-world data. This dataset serves as a valuable resource for training and evaluating our model's performance in capturing and reproducing the underlying distribution of the data.







VAE Structure

The encoder and decoder are fundamental components of a Variational Autoencoder (VAE), responsible for transforming input data into a latent space representation and reconstructing it back to the original data space, respectively.

The **encoder** network compresses input data into a lower-dimensional latent space representation. In this implementation, the encoder consists of several densely connected layers, each followed by rectified linear unit (ReLU) activation functions, facilitating the extraction of meaningful features from the input data. The network architecture culminates in a linear layer that outputs both the mean and variance parameters of the latent space distribution, enabling stochastic sampling using the reparameterization trick. Additionally, the encode_and_sample method provides functionality for computing the posterior mean and variance, as well as sampling latent space vectors for subsequent decoding.

The **decoder** component reconstructs data from the learned latent space representation. In this implementation, the decoder comprises a series of densely connected layers, each followed by rectified linear unit (ReLU) activation functions, facilitating the transformation of latent space vectors into meaningful data representations. The network architecture culminates in a linear layer with a sigmoid activation function, ensuring that the output values are constrained within a specific range. The decode method encapsulates the decoder's functionality, seamlessly invoking the forward method to generate reconstructions of input data from latent space representations.

The **sample function** generates synthetic data points that adhere to the learned distribution of the input data, GMM. By leveraging random noise vectors sampled from a standard normal distribution, the function decodes these vectors through the decoder network of the VAE. This process effectively synthesizes new data points in the original input space. This functionality is essential for tasks such as data augmentation, where additional synthetic data can be generated to supplement the original dataset, or for evaluating the quality and diversity of the learned latent space.

The **Loss function** used has two main components. The reconstruction loss measures how well the VAE reconstructs the input data, while the KL divergence loss encourages the learned latent space to match a prior distribution. By minimizing this combined loss during training, the VAE aims to balance accurate data reconstruction with regularization of the latent space.

Clustering and modes identification

On one hand, in order to compute the clusters, and compare the correspondence between clusters identified in the latent space and those derived from ground truth data a function called compare_clusters was computed. Initially, it calculates the latent representations of the input data using the VAE's encoder module. After that, K-Means clustering is applied separately to both the VAE representations and the original data, with the aim of partitioning them. The function then obtains the optimal number of clusters. It is called for both the vae samples and the ground truth ones.

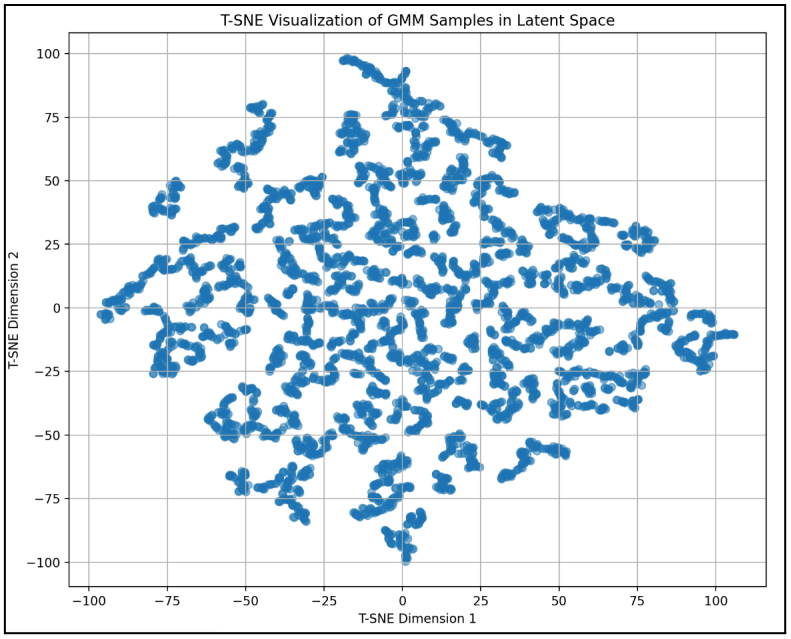
On the other hand, in order to identify the modes (peaks) a function called identify_modes has been obtained. Initially, it computes the density distribution of the data using a histogram with 50 bins, thus creating a smoothed representation of the data's distribution. After that, peaks in this density function are identified using the find_peaks function, where only peaks surpassing a specified threshold in height and separated by a minimum distance are considered significant. By tallying the number of identified peaks, the function determines the count of modes within the dataset. This count reflects the distinct clusters or prominent features present in the data distribution.

The results that were obtained using both algorithms are of 2 clusters in the vae samples and of 4 in the ground truth data, which means that the vae is not able to represent and capture the whole structure. Regarding the modes, 5 have been obtained for vae and 1 for the ground truth data, this means that the alignment is again not correct, and probably there are errors in the training.

T-SNE visualization

In this last part of the lab, T-SNE will be used to project the representations into a two-dimensional space, but the similarities between data points are kept.

As a result, a scatter plot is obtained where the different points can be interpreted in the following way: the proximity indicates similarity in the original high-dimensional space, so, by examining the spatial arrangement of points, it is possible to discern clusters and patterns, offering insights into the learned representation's organization. This visualization help us in understanding how well the VAE captures the structure of the data and enables the identification of distinct clusters or relationships within the latent space.



An interesting insight we have thought about is that, as the obtained results are not good as the vae samples are not representing the structure of the Gaussian Mixture Models. For that reason, an interesting option would be to make the number of epochs bigger to see if the training is done better, and better results are obtained.