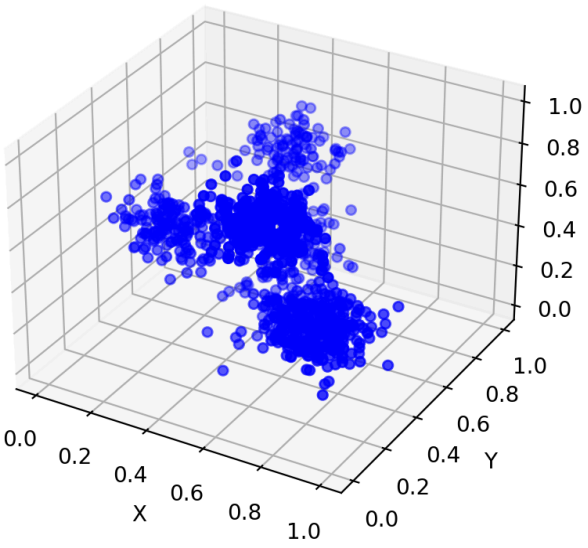
Neural networks - Project 4

Olga Bonachera del Pozo - Daniel Kwapien - Alejandro Sánchez Díez

1. 3D GMM synthetical data generation

Following the statement, we first defined a 3-dimensional Gaussian Mixture Model (GMM) data generation function. This algorithm creates an array of dimensions: *number of dimensions* x *number of components*. Then, assigns to each component a **mean**, which will determine the center of the Gaussian (G) in the 3-dimensional space, a **covariance**, securing a variance = 1 and a covariance = 0 for all components, translating into their mutual independence, also controlling the form and spread of the G, which, because of the use of identity matrix, will be a sphere, and a **weight**, as the probability of being chosen for a data entry, summing all up to 1. We used 10 for the number of components feature and 1000 for the number of samples, obtaining a complex dataset with multiple overlapping and distinct clusters.



This initial plot shows some initial clustering of the points, which might indicate the different components of the mixture, although is early to draw conclusions on them.
Finishing with the initial steps, we used *DataLoader* function from *pytorch* to mix the whole dataset in batches of 32 samples at a time, shuffling it for each epoch, in order to avoid the Neural Network learning any inadvertent ordering, leading to a better convergence

2. Replacing CNNs with Dense Layers

The Variational Autoencoder (VAE) using Dense Layers (DL) instead of Convolutional Neural Networks (CNN), is a strategic choice based on the structure of the data. We are focused on capturing the internal distribution and correlations of the data at a global level, not assuming any spatial correlations, treating each input dimension independently, but with full connectivity to the subsequent layer.

Our VAE's *forward* method uses the encoder to obtain the mean, variance, and a sample of the latent space, then passes the sample through the decoder to generate the reconstructed output. We ran the VAE's *extended* function, fixing the **channels** = 3 (input dimension), **dimz** = 2, returning a compressed representation better for visualization and representation, with *mu* and *var* the learning rate for *Adam's optimizer* to **lr** = 1e-3, a typical value for deep learning applications, providing balance between convergence speed and stability, and finally, **epochs** = 100, allowing the model to adjust its weights iteratively to minimize the loss function.

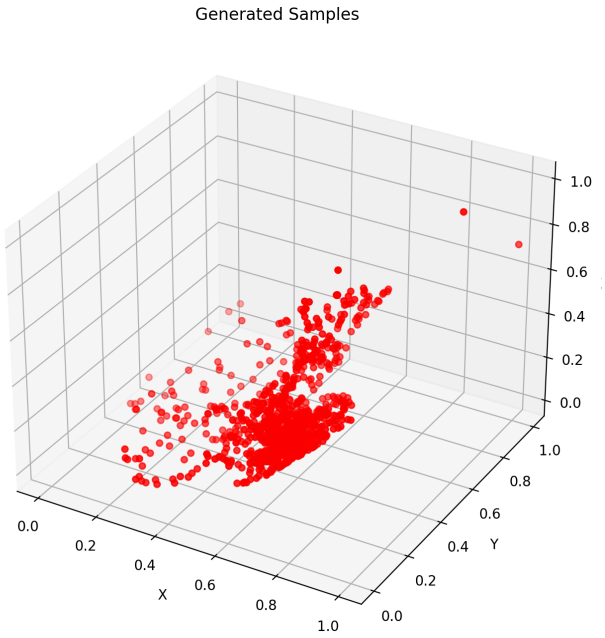
Training epoch	#0	# 24	# 49	# 74	# 99
Loss value	0.654027	0.052035	0.051890	0.051855	0.051320

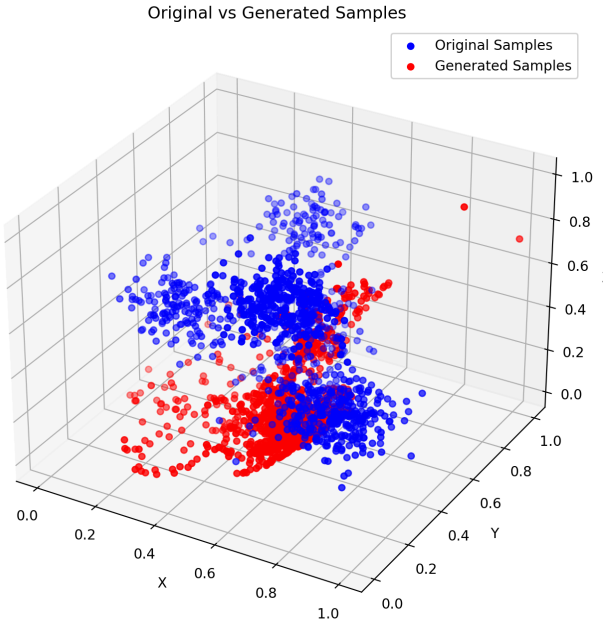
The rapid drop of the loss function (in epoch #1, loss value already 0.072070) indicates a fast learning rate by the model. The ending flattening of the learning curve suggests that the model is approaching a point of convergence where further training results in diminishing improvements.

3. Comparison: VAE samples vs ground truth distribution

Then we set the model to evaluation mode and generate 1000 random from a standard Gaussian distribution vectors from the trained VAE. Then we apply Min-Max scaling independently for each column.

This visualization provides an immediate way to evaluate how well our VAE model has learned to mimic the original data distribution. Ideally, the generated samples should cover the same space as the original data and exhibit similar clustering or dispersion characteristics.

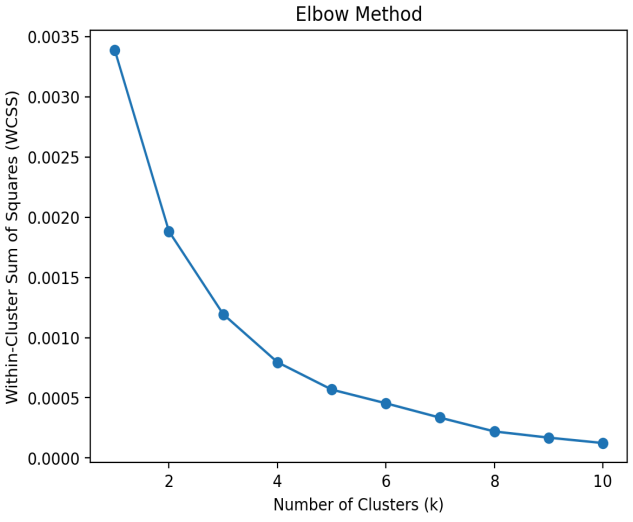




The original samples seem to be more densely packed and cover a broader area of the plot, especially in the lower regions of all 3 dimensions. On the other hand, generated samples are less densely distributed and do not cover as broad a range of the plot. There is some overlap, particularly in the center, suggesting that the model has learned to extent the central tendency of the data distribution. However, the extremities, especially the higher values along all axes, are less well represented by the generated samples, the model might be underfitting and not capturing neither the tails of the distribution nor the full diversity of the original data.

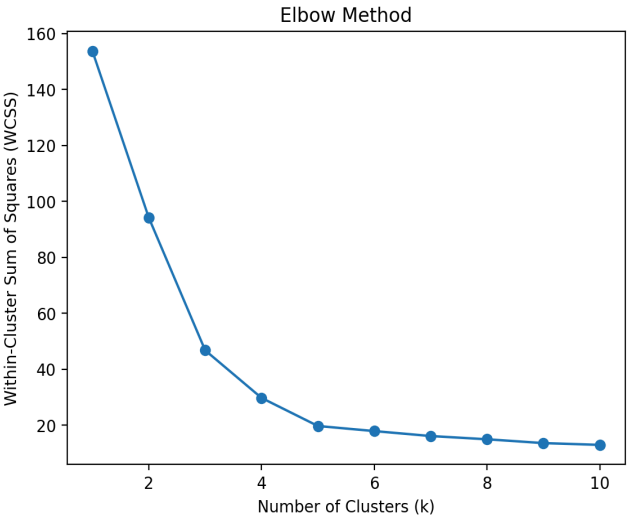
3.1. Q: How many components are effectively captured

Through a KMeans algorithm we cluster the VAE generated samples. With a maximum number of clusters of 10, related to the 10 original components, and a 20 samples batch, we obtained an optimal number of clusters k = 2 (first plot), accepting up to k = 3 and k = 4, while in the non scaled data, the optimal number of clusters found is again k = 2 (second plot), maintaining the coherence. Our final decision is k = 3, which means it is only capturing 3 effectively components from the 10 original ones. This reiterates our statement of the underfitting model or open new gates as lack of variability among the dimensions or too much significant overlapping between the data points.



3.2. Q: How many modes are missing

The difference between the number of components used in the synthethic data and the number of clusters in the VAE generated one, gives the count of 7 missing modes. In an effort to think about a solution, we state that a larger dataset or a different more complex clustering technique like another GMM could capture the totality of the components, however, we would then have to search for overfitting issues.

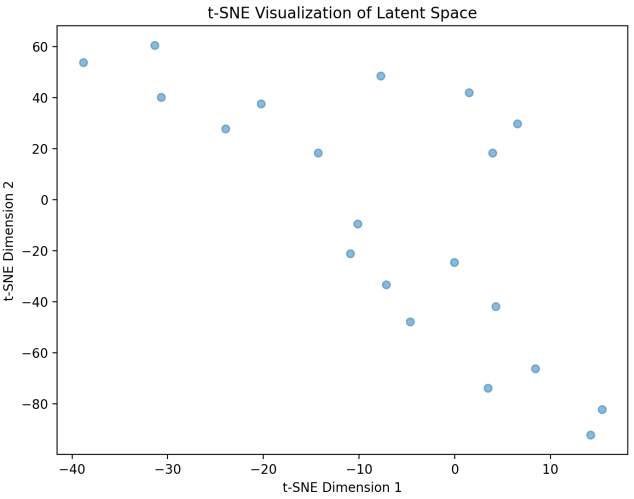


4. Using T-SNE for Visualization

We finally performed a t-SNE visualization of the latent space. This is a useful approach to inspect how the VAE represents and organizes data in a 2-dimension space.

4.1. Results analysis

As we see, the data points are spread relatively widely across the plot, doing it with respect to both dimensions implies that the two t-SNE components are effectively capturing variance in the data. There appears to be no clear or dense clustering, which can suggest several things: that the data does not naturally form tight clusters, which was our initial idea analyzing the 3-dimensional data plot, or the perplexity parameter (chosen as 5), might be too low to capture more nuanced clusters, possibly emphasizing local data structures at the expense of global ones. The absence of clear clusters can also be due to the overlapping categories, where data points might not belong to sharply defined categories, but that is a condition we are aware of from the beginning.



4.2. Q: Number of components from number of clusters in latent space

From this visualization, it's challenging to conclusively determine a specific number of components without additional analysis. The absence of clear, well-defined clusters might indicate that the dataset does not contain distinctly separable categories as captured by our model, or that the t-SNE perplexity and dimensionality reduction parameters may not be optimally set to reveal the clusters.