AIML425 ASSIGNMENT 3

Benjamin McEvoy

300579954

1. INTRODUCTION

This document revises the concepts of Auto Encoders, Variational Autoencoder (VAEs), Mean Squared Error (MSE), and Signal-to-noise ratio (SNR). The code is located in this repository or this google drive link: https://drive.google.com/file/d/1UrWSKobb3WOsHa4c5QHOnsJtF2AVmKoq/view?usp=sharing.

2. THEORY

This section outlines the core knowledge of the assignment, and course for this program.

2.1. Autoencoders (AE)

An Autoencoder is a type of neural network primarily utilised in unsupervised learning tasks, data compression, noise reduction, anomaly detection and image generation. In basic form, it reiterates the principles of:

1. Encoder: maps data onto low-dimensional space ("latent layer")
2. Bottleneck/Latent layer: low-dimensional description of data
3. Decoder: It's attempt to reconstruct the data

However, the problems with a basic AE, are that it's not particularly good for generation as its distribution of the latent variable is not controlled and that the relation input, output, and latent layer have no reason to be meaningful, other than the network map being smooth.

2.1.1. Variational Autoencoder (VAE)

A Variational Autoencoder (VAE) is a generative model that learns to encode input data into a lower-dimensional latent space while ensuring that this latent space follows a specified distribution, typically Gaussian. It consists of an encoder that compresses data and a decoder that reconstructs it from the latent representation. VAE minimises the reconstruction loss during training to encourage the learned distribution to be close to the prior distribution. This enables the VAE to generate new, similar data samples by sampling from the learned latent space.

2.2. Signal-to-Noise Ratio (SNR)

In neural networks, Signal-to-Noise Ratio (SNR) refers to the relationship between the meaningful data (signal) and the irrelevant variations (noise) present in the input. A high SNR indicates that the relevant features significantly outweigh any noise, allowing the model to learn effectively and generalize well to unseen data. Conversely, a low SNR can lead to overfitting, as the model may capture noise instead of the underlying patterns, resulting in poor performance.

In AEs, the latent variable captures a compressed representation of the input data; introducing iid (independent and identically distributed) Gaussian noise, in the latent space simulation of real-world scenarios where data may be corrupted is applicable.

3. EXPERIMENTS

This outlines the methodical approach to obtain the results, and explanations and conclusions about said results.

3.1. Methods

The explanation of the methods used in the code base, and results they provided.

3.1.1. 3D Data Generation

The method creates a dataset of points distributed on the faces of a cube by generating random coordinates in three-dimensional space. It iterates six times, once for each face of the cube. For each iteration, one of the three dimensions (x, y, or z) is fixed at either 0 or 1, depending on which face is being represented, while the other two dimensions vary randomly. After generating points for all six faces, the function combines these points into a single array, resulting in a uniform distribution of points on the cube's surface. **Figure 1**. visualises the output.

3.1.2. Basic Autoencoder and MSE Objective Function

The basic autoencoder compresses the input into a latent space and reconstructs the data. The MSE objective function helps the network learn how to minimise the difference between input and output, treating the error probabilistically as

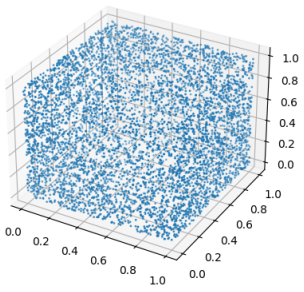


Fig. 1: 3D Data Generation

normally distributed noise. Specifically, the MSE is equivalent to maximising the likelihood of the data assuming that the reconstruction error follows a Gaussian distribution with zero mean and constant variance. **Figure 2**, visualises the output.

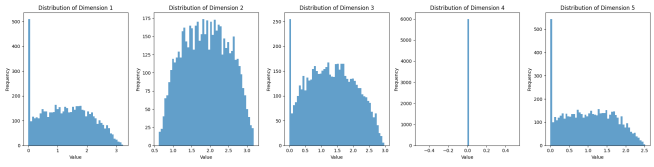


Fig. 2: Dimension Distribution - Basic Autoencoder

3.1.3. Distribution Control Method

In this implementation of a Variational Autoencoder (VAE), an adjustable Signal-to-Noise Ratio (SNR) is introduced to control the distribution of the latent (bottleneck) variable. The model consists of an encoder that learns to compress input data into a latent representation, along with parameters for the mean (μ) and the logarithm of the variance ($\log(\sigma^2)$). In the reparameterization step, Gaussian noise is added to the latent variable to ensure variability in the encoding process. This noise is generated with a specific variance determined by the SNR; specifically, the noise power is set as the signal power divided by the SNR. This approach allows for controlled noise levels in the latent space, helping to balance the information retained from the input against the variability introduced by the noise, thereby influencing the model's generative capabilities. The VAE is trained using MSE loss and KL divergence to encourage the latent representation to follow a desired distribution, facilitating the reconstruction of the input data. **Figure 3**, visualises the changes in the distribution.

3.2. Results

The main experiment consisted of data evaluation over a range of latent variables and SNR values and their results.

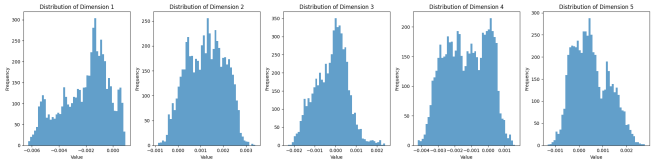


Fig. 3: Dimension Distribution - Controlled Latent Variable

Compilation of a snippet of the generated data and reconstructed data in the appendix figure 4, shows a trend in the distribution of the reconstructed model based on the latent variable. Figure 4 illustrates the comparison of latent dimensionality and its correlation to its increase of information present, the lower the dimension the less information it can't gather to efficiently reconstruct the given data.

4. TOOLS

This assignment used the tools of Numpy [1], Matplotlib [2], Seaborn [3], SKLearn [4] and Pytorch [5] to carry out experimentation.

Google Colab [6] is where the code base is hosted and is provided with any available machine during runtime allocation.

Besides the utilisation of these tools [7], I declare that the code and paper provided is my work.

5. CONCLUSION

Autoencoders are a versatile tool in machine learning, their application is used in a variance of different unsupervised tasks. However, limitations in basic AEs are apparent in the results as the distribution of generation isn't up to par without sufficient means for latent variable control and regularisation. While the VAE framework, Gaussian distribution and SNR were welcome to the improvement of the generation, there is still difficulty producing noteworthy results to be compared to the original data. For results to truly indicate the effectiveness of AEs, the regularisation techniques and AE complexity need to be considered to provide further results for comparison.

If we were to extend the study of AEs further, an application of AEs could be Conditional Variational Autoeconders (CVAEs), it enhances traditional VAEs by including conditional inputs, allowing for the model to generate data based on specific attributes. As image generation can be a particular application to be focused on, specifics can range from class labels or descriptive text to generate certain data samples.

6. REFERENCES

[1] Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al., "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[2] John D Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

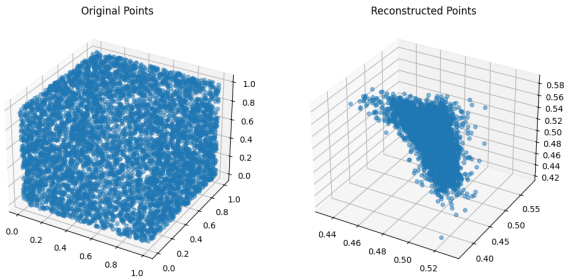
[3] Michael L. Waskom, "Seaborn: statistical data visualization," https://seaborn.pydata.org, 2021, Accessed: 2024-09-13.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

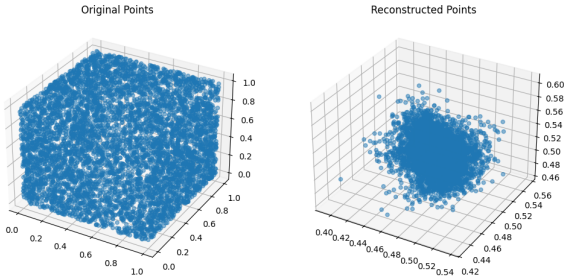
[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[6] Google, "Google colaboratory," Online, 2023, Available at: https://colab.research.google.com.

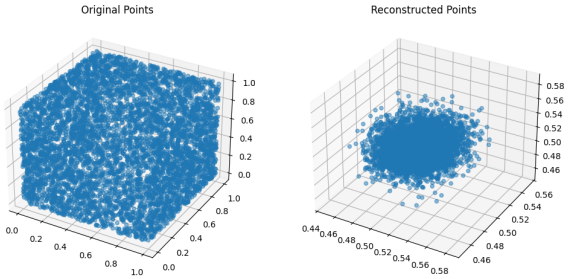
[7] Inc. Grammarly, "Grammarly," https://www.grammarly.com, 2024, Accessed: 2024-08-12.



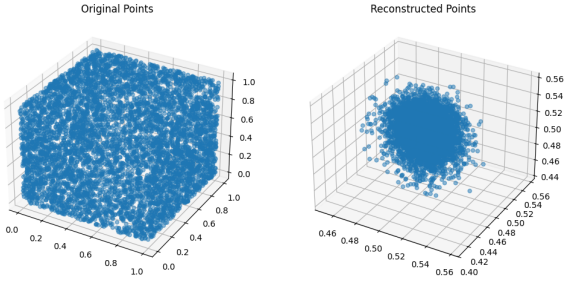
(a) Latent Variable:3, SNR:20



(b) Latent Variable:5, SNR:30



(c) Latent Variable:10, SNR:30



(d) Latent Variable:20, SNR:30

Fig. 4: Original Points against Generated Points

Appendix