Structured Latent Space in Variational Autoencoders (VAEs) - Task 1 Documentation

Akshay Shukla (2022A7PS0087P)
Deepan Roy (2022A7PS0004P)

April 2025

0.1 Introduction

Through this project, we aim to build a custom variational autoencoder (VAE) architecture and successfully train it to create a well-structured latent space with disentangled dimensions for content and style. We will sample from this latent space to generate diverse and meaningful images.

0.2 Omniglot Dataset

The Omniglot dataset contains images of 1623 different handwritten characters from 50 different alphabets. These characters were written by 20 writers, altering each character's style. We attempt to model both the content and style of the characters in this dataset with our VAE network.

0.3 VAE Architecture

We build our VAE model with convolutional layers for the encoder and convolutional transpose for the decoder. We try to ensure separation of content and style features by generating two separate latent vectors, one for content and the other for style, from the encoder.

The decoder then uses these two vectors to generate images. The style and content vectors are concatenated before inputting it to the decoder network. Thus, style and content are separated to preserve the latent space structure.

We use the reparameterisation trick to maintain tractability in our network. The style and content vectors are generated as mean and variance of the latent space, then are applied to a sample from the normal distribution to generate the inputs for the decoder.

0.4 Maximum Mean Discrepancy (MMD)

Vanilla VAEs use the KL Divergence metric to train the network to model the latent space according to a prior distribution (usually Gaussian). We replace this with another metric: Maximum Mean Discrepancy (MMD). MMD is similar to KL Divergence in that it also helps the model reduce the difference between two distributions. However, MMD instead measures the difference between the distributions by measuring the difference between the means of two distributions after they've been embedded in a reproducing kernel Hilbert space (RKHS). MMD is calculated as follows:

$$\text{MMD}^2(P, Q) = E_{x,x' \sim P}[k(x, x')] + E_{y,y' \sim Q}[k(y, y')] - 2E_{x \sim P, y \sim Q}[k(x, y)] \quad (1)$$

0.5 Evaluation Results

We evaluated the latent space structure using KL divergence. Our final KL divergence value was **63.1432**. Our final clustering accuracy was **0.0017**. We also generated a t-SNE plot as shown below:

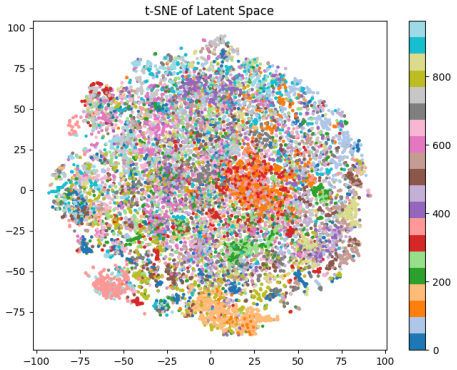

Figure 1: t-SNE Plot for VAE network

0.6 Future Work

There remains scope for improvement in the latent space structuring, as both KL divergence and clustering accuracy remain worse than expected results. Also, the subpar clustering is visualised in the t-SNE plot. Thus, our primary goal in task 2 will be to improve the structuring of the latent space. While there are many approaches to accomplishing this task, the most suitable method we have reviewed is to use an adversarial training method, as described in "**Adversarial Autoencoders**", by Makhzani et al. A discriminator network will attempt to distinguish between a prior distribution and the latent space distribution, while the generator (VAE encoder) will iteratively improve on the latent space distribution, allowing for better structuring.