Variational Deep Embedding: Implementation and Evalutation

Adam Czerwoński

January 2025

1 Introduction

In this report, I will describe my implementation and evaluation of Variational Deep Embedding(VaDE) [1]. This work was carried out as part of the Mathematical Underpinnings of Machine Learning course at the Warsaw University of Technology.

2 Variational Deep Embedding

In this section, I will shortly describe how the model works.

2.1 Assumptions

The data for the model is assumed to be generated in the following way (in the case of continuous x):

1. Choose a cluster $c \sim \text{Cat}(\pi)$
2. Choose a latent vector $\mathbf{z} \sim \mathcal{N}\left(\mu_c, \sigma_c^2 \mathbf{I}\right)$
3. Choose a sample $\mathbf{x}$:
   (a) Compute μ_x and σ_x^2: $[\mu_x; \log \sigma_x^2] = f(\mathbf{z}; \theta)$
   (b) Choose a sample $\mathbf{x} \sim \mathcal{N}\left(\mu_x, \sigma_x^2 \mathbf{I}\right)$

where π is the prior probability distribution over clusters, $\text{Cat}(\pi)$ is the categorical distribution parametrized by π, μ_c and σ_c^2 are the mean and the variance of the Gaussian distribution corresponding to cluster c, $\mathbf{I}$ is an identity matrix, $f(\mathbf{z}; \theta)$ is a neural network whose input is $\mathbf{z}$ and is parametrized by θ, $\mathcal{N}(\mu_x, \sigma_x^2)$ is Gaussian distribution parametrized by μ_x, σ_x.

2.2 Model

The model consists of encoder-decoder architecture and also estimations of π_c, μ_c and σ_c for each cluster c.

2.2.1 Encoder

The encoder is a neural network that given x tries to estimate its latent space representation z.

2.2.2 Decoder

The decoder is a neural network that given the latent vector z tries to estimate the original vector x.

2.2.3 ELBO

During training, VaDE is trying to maximize the ELBO function [1] with respect to the encoder and decoder parameters and the π_c, μ_c, σ_c values.

3 Implementation notes

The model was implemented using TensorFlow library in Python. The number of Monte Carlo samples denoted by L in [1] was set to 1. During the calculation of ELBO, in particular while calculating $p(z|c)$, a small value was added for numerical stability.

4 Data

To evaluate the performance of the model, I used the MNIST dataset [2], which contains 70,000 images of handwritten digits.

5 Evaluation

The data was split into training and test sets (60,000 observations and 10,000 observations, respectively). Before training VaDE, the following steps were taken to ensure that the model would not get stuck in a local minimum:

1. The VAE model was trained for 3 epochs on the training data.
2. π_c, μ_c, and σ_c were estimated using a Gaussian mixture model.
3. VaDE was initialized with the weights from the VAE and the parameters from the Gaussian mixture model.

The loss minimized during training was the negative ELBO.

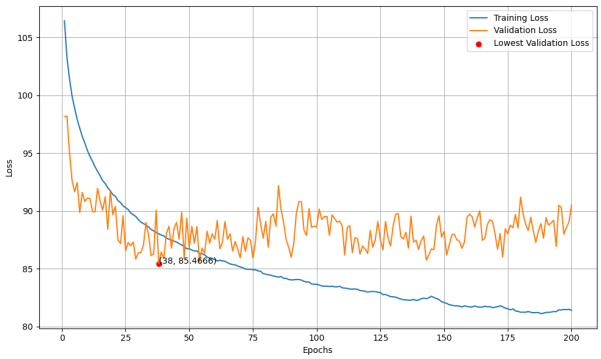

Figure 1: Loss during training of VaDE

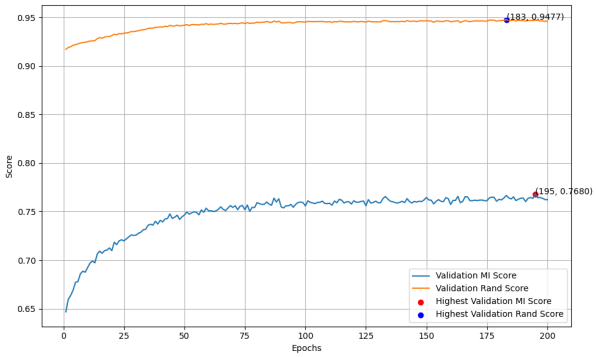

Figure 2: Rand and Normalized Mutual Information scores during training of VaDE

5.1 Loss and clustering performance

In Figure 1, we can see that the validation loss substantially decreased during the first 50 epochs and then plateaued. However, it exhibited significant fluctuations, which might be concerning and may require further investigation.

To evaluate clustering performance, the Rand Index and Normalized Mutual Information were calculated on the test data using the true labels. Most of the improvement in both metrics occurred during the first 75 epochs. The maximum Rand Index achieved was 0.947, while the maximum Normalized Mutual Information was 0.768.

5.2 Comparison with VAE

To compare the VAE and VaDE models, their latent spaces were analyzed, specifically focusing on how well the models separated different digits. The testing data was transformed into the latent space using both VaDE and VAE, then projected onto a 2-dimensional space using t-SNE. The results are shown in Figure 3 and Figure 4. While it appears that the VAE separated the digits more distinctly, the calculated Silhouette and Calinski-Harabasz scores (Table 1) indicated that VaDE actually performed better. These scores evaluate the compactness of observations within clusters and the separation between clusters.

5.3 Generative capabilities

To evaluate the generative capabilities of VaDE, 10 examples were generated for each digit. The results, shown in Figure 5, are not perfect. The model strug-

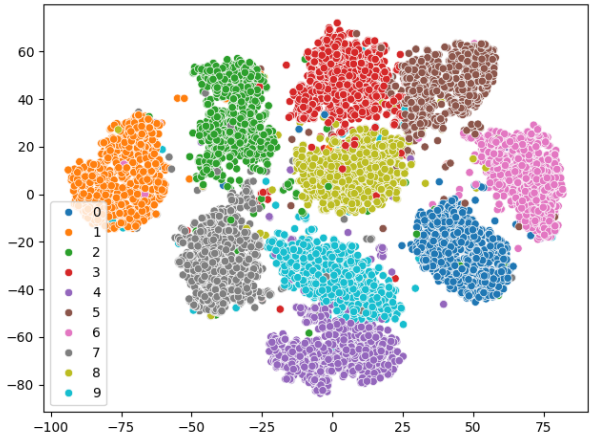

Figure 3: Observations from the test set in latent VAE space

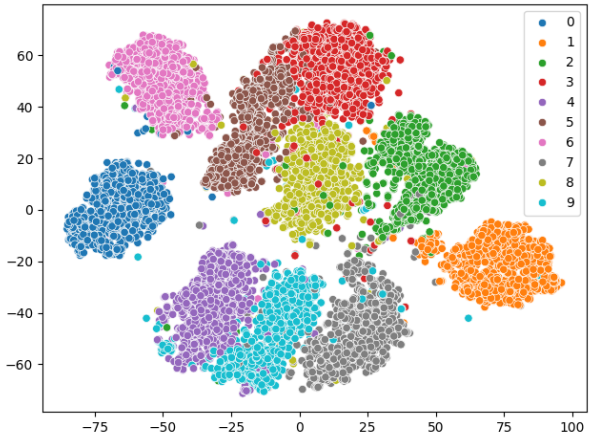

Figure 4: Observations from the test set in VaDE latent space

Model	Silhouette Score	Calinski-Harabasz Score
VaDE	0.158	1169
VAE	0.109	646

Table 1: Comparison of clustering performance metrics for VaDE and VAE models.

gles the most with two groups of digits: 4, 7, 9, and 3, 5. This issue was also observed in the previous section, where these digits appeared close to each other in the latent space.

In Figure 6, a comparison is shown between the generated and real digits. For example, the color in row 0, column 1 represents the distance between the digit 0 generated by VaDE and the closest real example of 1. Each digit type was generated 10 times, so these distances are averaged and then scaled. We expect to see small distances between the same digits and large distances between different ones. There is, once again, evidence that the model struggles with 4 and 9, as the distance between them is small. Interestingly, the ones are quite close to all of the other digits.

6 Summary

Although VaDE performed slightly better than VAE, the results are somewhat underwhelming. The authors in [1] achieved much better results, particularly when generating new digits. To bridge this gap, a different training setup and potential implementation discrepancies should be investigated.

References

[1] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering, 2017.

[2] Yoshua Bengio Yann LeCun, Léon Bottou and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Figure 5: Digits generated by VaDE

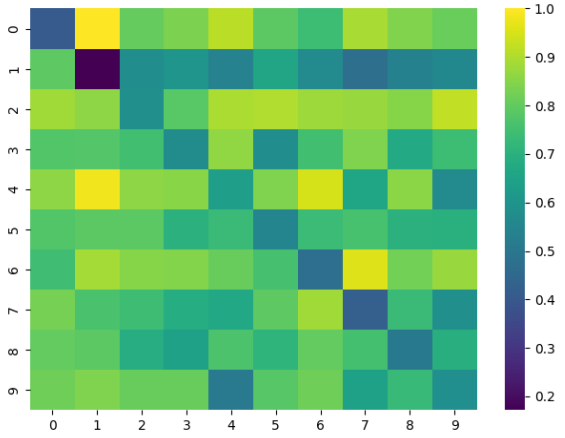

Figure 6: Distance between generated and real digits