BAM! : Variational AutoEncoder

Mounir Messaoudi, Alik-Serguy Rukubayihunga, Benjamin Sykes

Tuesday 22nd November, 2022

1 Introduction to VAE

1.1 Auto Encoders and the need of regularisation

Autoencoders is a family of neural networks that learn how to compress and encode data efficiently and then reconstruct the data from the reduced encoded representation to a representation ideally identical to the original input. The autoencoder architecture consists of two primary components: the encoder, which reduces the data to a given latent dimension, and the decoder that reconstructs the original data. This specific architecture allowes the generation of random images from the latent space, however, this given latent space lacks regularisation : points from the same class could be mapped to very different and far places in the latent space. To tackle this issue, VAE were proposed [2].

1.2 The two different missions that are carried by VAE : reconstruct images (compress in the latent space) and generate new images

Thus, a variational autoencoder, instead of compressing its input into a fixed projection in the latent space, turns it into the parameters of a statistical distribution: a mean and a variance. This statistical process adds randomness during the encoding and decoding training process. The VAE uses the mean and variance parameters to randomly sample one element of the distribution, and decodes that element back to the original input [2]. The stochasticity of this process improves robustness and forces the latent space to encode meaningful representations everywhere: every point sampled in the latent space is decoded to a valid output. Figure 1 shows the VAE architecture and the notion of latent distribution.

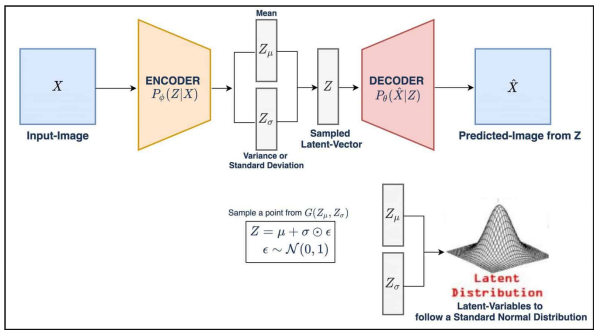


Figure 1: VAE architecture [3]

1.3 The reparameterization trick

The process of sampling from a distribution that is parameterized by this model is not differentiable. It is necessary to be able to make predictions separate from the stochastic sampling element.

This problem can be solved by applying the reparameterization trick to the embedding function [2]. Therefore random sampling is treated as a noise term. A noise term is added and is now independent of and not parameterized by the model. Consequently, the prediction of mean and variance is no longer tied to the stochastic sampling operation [1].

1.4 A double target : reconstruction loss and KL divergence

The VAE was created in order to overcome the weak results in image generation in conventional Auto Encoders. As we saw in previous section, the AE only holds a constraint on the reconstruction loss, which makes it a model good for image compression but struggles to generalise its abilities and thus has week results when generating images.

To that extent, VAE includes -in addition to the reconstruction term- a **regularisation term** that is focused on the shape of the latent space [1]. This regularisation term, known as **KL Divergence** constrains the model's distribution in the latent space to being as close as possible to the a normal distribution $\mathcal{N}(0, I_{z_d})$ where z_d denotes the chosen dimension of latent space. The KL Diverenge becomes bigger when the difference between the empirical and targer distributions is large.

In equation 1, the first part corresponds to the reconscruction term, we decided to use the MSE, though there are other implementations such as intersection over union or binary cross-entropy. The second term corresponds to the KL divergence. θ corresponds to the decoder parameters and ϕ to the encoder.

$$\mathcal{F}(\theta, \phi, k_1, k_2, \boldsymbol{x}, \boldsymbol{z}) = k_1 . \underset{\boldsymbol{z} \sim q_{\phi(\boldsymbol{z}|\boldsymbol{x})}}{E} [p_\theta(\boldsymbol{x}|\boldsymbol{z})] - k_2 . KL(q_\theta(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \tag{1}$$

The parametrisation of both individual losses allowes to strengthen either the reconstruction parameter of the model (closer to conventional AE) or regularisation of the latent space, which will thus improve the generation of images of the VAE. This idea of parametring the losses was proposed in [4] in order to disantangle the features in the latent space.

2 Our implementations and detailed results

2.1 Loss function

As described in [2], the ELBO loss we are using can be approximated in the gradient descent by :

$$L(x, x_{recon}, \mu_i, \sigma_i) = -k_2 . \frac{1}{2} \left[\sum_i \left(\log \sigma_i^2 + 1 \right) - \sum_i \sigma_i^2 - \sum_i \mu_i^2 \right] + k_1 . \text{MSE}(x, x_{recon}) \tag{2}$$

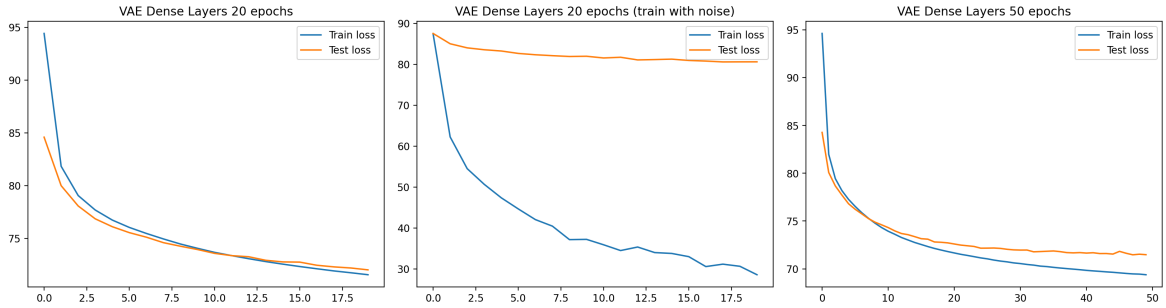
Where we are suming on all dimensions i of the latent space, x is the input image and x_{recon} its reconstruction by the model.

2.2 The CNN model

The CNN model is a well-performing model, with a low number of parameters. In order to improve it, we tuned the number of dimensions of the latent space going from 1 to 2048 dimensions. At the top of its performance (for 16 dimensions), we had a good FID equals to 15.77, with a low accuracy, no more than 30 percent. Finally, we choose to focus on this model, for the 1024 images generation.

2.3 The dense model

We chose to build a model based on dense layers, in order to compare the performance with the convolutional models. It is composed of 3 denses layers, going from 400 000 to 800 000 parameters depending on the layers size. At the top of it performance, we got a FID equals to 25.83. However the accuracy (about number distribution) was not as good, around 35 percent. We tried to tuned the model on : numbers of epochs, number of parameters (depending on layers sizes), adding noised to the training data, trying normalisation (was not a success). Some results are shown in figure 6a.



(a) Without noise, 20 epochs (FID 25.93)

(b) With noise. 20 epochs (FID 28.47)

(c) Without noise, 50 epochs (FID 26.50)

Figure 2: Comparing the train and test losses for varying epochs and training data

2.4 Reconstructions

In order to compare the models, we tried to make a reconstruction based on selected images from MNIST. In figure below are visual comparison of reconstructions on both models.

(a) Reconstructed images from Dense VAE (FID 25.93)

(b) Reconstructed images from CNN VAE

2.5 Latent space visualisations

In order to understand the effects of the training losses functions on the latent space, we visualise the latent space with a certain number of unseen data from the training set. The pytorch implementation indeed allowes us to use the trained encoder by itself, which allowes to get the latent space variables associated to unseen data. This allowed us to visualise the latent space. For 2d latent space, we plot directly the associated latent space, and for a latent space with dimension above 3, in order to illustrate the complete latent space, we used T-SNE in order to plot the latent space in a 2d graph shown in 4

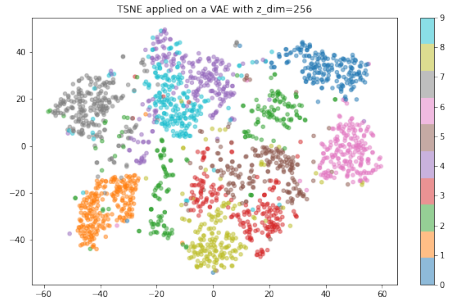
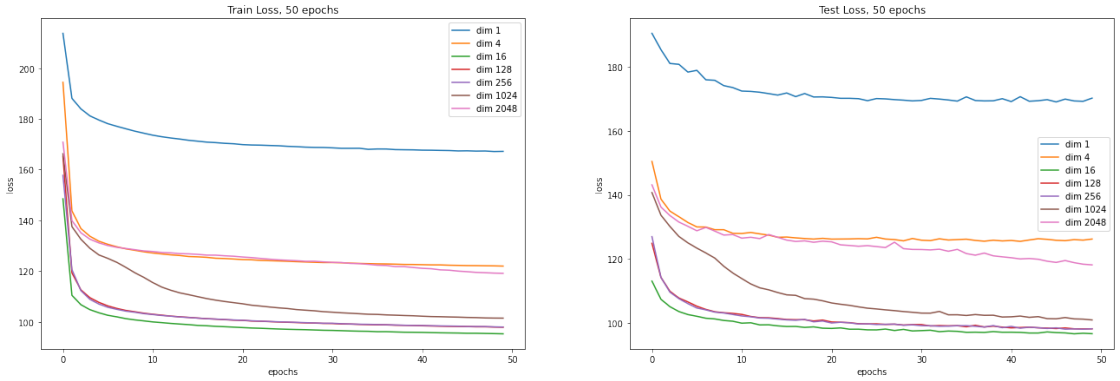


Figure 4: TSNE for a 256 dimensions latent space

2.6 Impact of latent dimension

We were interested in understanding the influence of the models with varying latent dimension z_{dim}. To study the impact of the z_{dim}, we created several models, with $z_{dim} \in [1, 4, 16, 128, 256, 1024, 2048]$. Intuitively, with very low z_{dim} the dimension will not be sufficient to understand the data, and with very high z_{dim} there might be overfitting as the number of parameters increases drasctically.

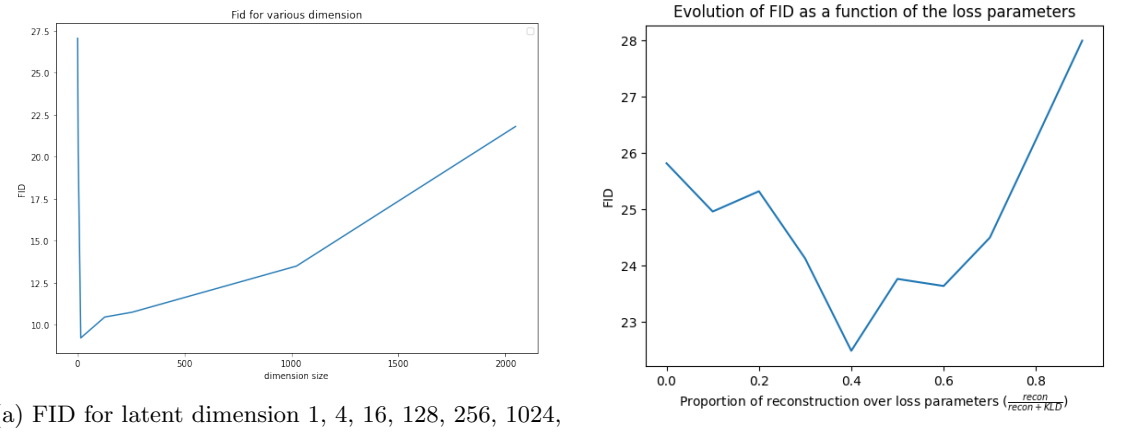
We thus created and trained those models, with $k_1 = k_2 = 0.5$ and evaluted both the losses curves and the FID for the generated images. Figure 6a shows both the train and test loss for the models. It is interesting to see there



(a) Train losses

(b) Test losses

Figure 5: Comparing the train and test losses for varying latent dimensions, best perfomance for 16 dimensions



(a) FID for latent dimension 1, 4, 16, 128, 256, 1024, 2048

(b) Evolution of FID with different loss parameters

Figure 6: Parameters effects on FID for images generated from random noise

2.7 Impact of the loss parameters

We had decided to make the loss components parametrized as suggested in [4]. The key idea was to force the model to learn key features from the original data distribution. We thus studied the impact of the proportion of KL divergence Loss and reconstruction on the FID of images generated from random noise. Figure 6b shows that the optimal weights are for equally weighed losses of 0.5. We understand that by seeing that in extreme cases, either the reconstruction by itself is considered (standard AE with no latent regularisation), or only the KL regularisation is taken into account which prevents the model from creating relevent images.

3 To go further

3.1 Model robustness

In order to improve our models robustness, we first added random gaussian noise to some training images. We found that this didn't help the model getting better results by a noticable improvement. We also noticed that our model was incapable of correctly treating out of distribution data: the model, with the KL divergence constraint, imposed a strong mapping with a zero average and standard deviation of 1 in the latent space, also mapping in the vicinity of zero any kind of data out of the MNIST distribution. As an illustration, in figure 7 we show that a random gaussian noise image gets reconstructed as a plausible image from MNIST.

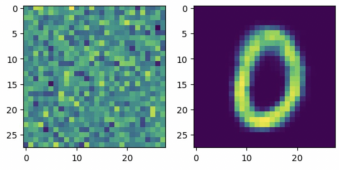


Figure 7: Random image reconstructed to realistic image

3.2 Other VAE implementations

In our work, we implemented the vanilla VAE as well as the β-VAE [4] (parametrized by our parameters k_1 and k_2. We present here alternatives VAE models that we would have liked to implement.

1. **VQVAE** : The key idea in the vq VAE is that the latent space is quantified as opposed to the discrete latent space in conventional VAEs. The latent space is thus even more constrained as there is a finite number of states, helping the generated images to be sharper [**?**].

2. **Conditional VAE** : The idea is here to benefit from the fact our data is labeled, and use this label as a feature value in the the VAE input space. This would help rassembling together similar data. The latent space variable is now $q_\phi(z|x, y)$ where x is the input image and y its label [5].

3. **Introspective VAE** : This variant is hybrid between a GAN and a VAE. GANs are known to produce sharper images than VAEs and the idea here is to train the generator (decoder) along with a discriminator which is the model itself [6]. This method showed to improve the Fréchet Inception Distance over conventional VAEs.

References

[1] Diederik P. Kingma and Max Welling (2019), *An Introduction to Variational Autoencoders*, Foundations and Trends in Machine Learning

[2] Diederik P. Kingma, Max Welling (2014) *Auto-Encoding Variational Bayes*, Machine Learning Group Universiteit van Amsterdam

[3] Aditya Sharma (2021) *Variational Autoencoder in TensorFlow*, accessed 3 November 2022, https://learnopencv.com/variational-autoencoder-in-tensorflow/.

[4] Irina Higgins et al. *beta-VAE: Learning basic visual concepts with a constrained variational framework* Proceedings of the International Conference on Learning Representations, 2016.

[5] Kihyuk Sohn, Xinchen Yan, Honglak Lee. *Learning Structured Output Representation Using Deep Conditional Generative Models*, NEC Laboratories America, Inc. University of Michigan, Ann Arbor 2016

[6] Huaibo Huang, Zhihang Li, Ran He, Zhenan Sun, Tieniu Tan *IntroVAE: Introspective Variational Autoencoders for Photographic Image Synthesis*