Generate mnist synthetic images using varaints of VAE

Medamine Bencheikh lehocine MOHAMMED.BENCHEIKH-LEHOCINE@DAUPHINE.EU

Mohamed Benmaiza MOHAMMED.BENMAIZA@DAUPHINE.EU

Yanis Chikh YANIS.CHIKH@DAUPHINE.EU

Team: MAY_VAE

Tutors: B. NEGREVERGNE, A. VERINE

1. Introduction

An Autoencoder is a deep neural network architecture that aims to find optimal compression for input data, it is composed of an encoder and decoder networks, the encoder learns to map any input vector X to a low latent space vector Y and the decoder learns to generate X' from the original space minimizing is the difference between X and X'. But this type of models is only adapted for image reconstruction and not for image generation, thus, the generated latent space is not regular, so it is possible to sample a point that is meaningless to the decoder. Because of this we will try another architecture, we will explore the use of variational auto-encoders to solve this type of tasks. In what follows in this work we will define and test three different approaches of image generation using a variational auto-encoder, namely, a Vanilla VAE, Beta-VAE and vector Quantized VAE. We will expose the different generated images and latent spaces on MNIST dataset and discuss the different results by citing the advantages and disadvantages of each model. We will then conclude with a comparison between the three models.

2. Vanilla VAE

2.1 Method explanation

Variational auto-encoders represent a function associating to an input value a multivariate latent distribution, which is not directly observed but inferred from a mathematical model from the distribution of other variables Kingma and Welling. (2013). Input data is sampled from a prior distribution, then the encoder and decoder are trained together so that the output minimizes a reconstruction error between the prior and posterior distribution var and et Welling (2019).

Unlike a classic auto-encoder where the generated latent space is not regular, a VAE is adapted to the content generation task, it generates new data by decoding points that are randomly sampled from the latent space. It then overcomes the problem of the latent space irregularity by making the encoder return parameters of a distribution over the latent space

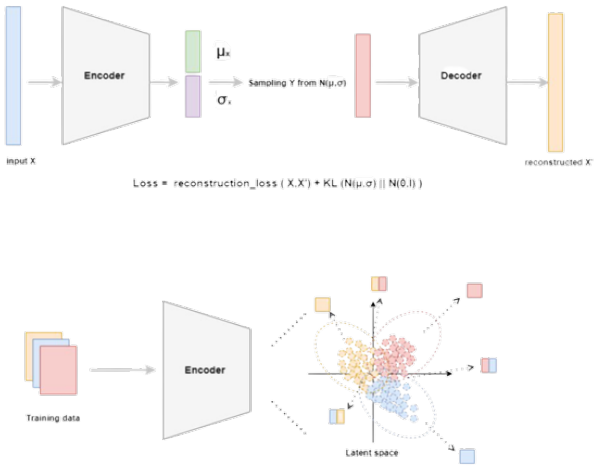

Figure 1: VAE architecture and illustration of its regulated latent space

variables instead of a single point dee. Moreover, VAE adds a regulation term to the loss function called the KL-divergence distance between sample distribution and a normal N(0,I) distribution. The idea is to construct for each input X a distribution and then make all these distributions near to each other (since we try to make them near the N(0,I) distribution), sampling is then done from the N(0,I) distribution.

2.2 Experiment settings and results

For this experiment, we used different dimensions for the latent space (2, 8,16 and 32), with the same architecture (2 Convolution layers and two dense layers for encoder, the decoder is anti-symmetric to the encoder). We only used **30 epochs** and used **128** for the **batch size**.

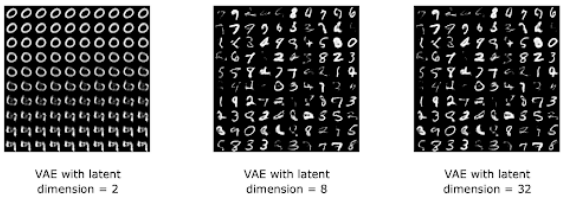

Figure 2: Samples from Vanilla VAE with different latent dimensions

As we can see, samples from vanilla VAE with latent dimension of two are not variants; we generated only 0's and 9's. we can explain this by the fact that the points sampled from $N(0, I_2)$ occur in the region of projected output from the encoder of 0 and 9 digits. This is due to the fact that the KL-divergence between decoder output distributions.

if we augment the dimension of the latent space, we get more and more meaningless samples. It is due mainly to the non-convergence of the KL loss. We think giving more weight to the KL-divergence loss using a hyper-parameter.

3. Beta Variational AutoEncoder (beta-VAE)

3.1 Method explanation

One of the drawbacks of VAE is that the learned latent representation is very entangled. An entangled representation is when a single latent unit is sensitive to one or more generative factors. We assume that our images are generated by some model or process that takes as input some generative factors. A disentangled latent representation is thus defined by the fact that each latent unit is sensitive and maps only one single generative factor. The beta VAE is able to learn a disentangled representation by introducing a beta factor in the total loss shown in the equation below:

$$L(\theta, \phi; x, z) = E_{q_{\phi(z|x)}} Log(p_\theta(z)) - \beta D_{KL}(q(x)||p(z))$$

The capacity of the latent information channel is constrained by Beta and combined with the pressure to maximize the log likelihood of the training data x under the model which will control the emphasis on learning statistically independent latent factors. With β ¿ 1 the model is pushed to learn a more efficient latent representation of the data, which is disentangled if the data contains at least some underlying factors of variation that are independent Higgins et al. (2017). Beta is a hyper-parameter that must be chosen carefully because setting beta to be very high then the model will focus on minimizing the kl divergence but will ignore the reconstruction loss.

3.2 Experiment settings and results

For beta-VAE, we used different dimensions for the latent space (2, 8 and 16), with the similar architecture used in Vanilla. We only used **30 epochs** and used **128** for the **batch size**. we tried different values for the β hyper-parameter.

Figure 3: projections of latent vectors with beta = 2 and dimension = 8 on the right side dimension = 2 on the left one.

We can see that for higher dimension, the latent space is well organized in contrast to what we had seen in vanilla model.

Figure 4: samples generated from beta-VAE with latent dimension = 8

4. Vector-Quantized Variational AutoEncoder (VQ-VAE)

4.1 Method explanation

As we explained in the previous sections; giving more importance to the regularization term of the latent space prevents the reconstruction error from vanishing in an acceptable rate which can lead to generate some deformed digits samples. lower weight for the KL-divergence term in the loss function makes the sampler generate the same few samples many times with no variety (only one or two digits at most as we had seen in vanilla VAE). An interesting idea to avoid this trade-off between variety and quality in sampled digits is to use dictionary-based latent space vqv ; instead of forwarding directly the re-parameterized latent vector z_e to the decoder, it is mapped to an embedding dictionary then passed to the decoder Oord and Kavukcuoglu (2017).

The posterior distribution $q(Z|X)$ is a one-hot distribution* and if we define $P(z)$ as uniform categorical distribution, the KL-divergence term will be constant Oord and Kavukcuoglu (2017). Then, No need for KL divergence in the loss function. VQ-VAE takes for the loss function, instead of minimizing the KL-divergence term, a combination of the embedding loss (how the embedding vectors are distant from encoder output) and commitment loss (how the encoder outputs commit to the embedding dictionary).

$$L = log[p(X|Z_q)] + \|sg(Z_e) - e\|_2^2 + \beta\|Z_e - sg(e)\|_2^2$$

Where sg means stop gradients operation, β is learning parameter used to balance the importance between commitment and embedding terms vqv.

It is important to notice that the encoded vector z_e is of shape $(H_l, W_l, latent_dim)$ where H_l and W_l are smaller dimension of the input image (we used $H_l = W_l = 7$). The encoder compress the input images into feature matrices then each feature vector is mapped to a vector of the dictionary. We train an auto-regressive model like **pixelCNN** to learn the mapping from encoded output of training data and corresponding embeddings Razavi and Vinyals (2019).

4.2 Experiment settings

In our experience, we used a straight through backward pass, meaning that we pass directly the gradients to the encoder output without passing by the dictionary (we used the suggestion of this ker). For the choice of hyper parameters we used Latent dimension in $[8, 16, 32]$, Size of the embedding vector in $[10, 20, 128]$, β parameter we used the value 0.25 as mentioned in Oord and Kavukcuoglu (2017) and an exponential decreasing learning rate scheduler.

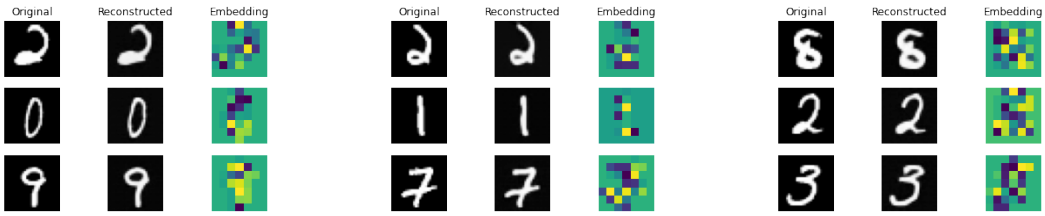

Figure 5: reconstructed images from mnist test and their corresponding embedding vectors

The results after training the VQ-VAE can be seen in the followings figures, using human-meaningless criterion to evaluate sampled images, we can say that more then 80% of images represent hand-written digits (although its subjectivity, we will compare implemented methods in more rigorous way in the next section).

Figure 6: 200 example of sampled images using the VQ-VAE generator

5. Comparative results

To have a more accurate view about the performance of the different implemented approaches and the quality of generated samples we use the inception score metric. to measure this metric we used pre-trained Resnet-18 model already trained on MNIST dataset.

We can see that VQ-VAE and beta-VAE with latent dimension 8 gives high inception score (we should mention also that we sampled only 800 image from VQ-VAE due to computation res-sources constraints). Vanilla VAE with latent dimension 2 gives high results with different sampling method (we used a random selection of region in the latent space then normal sampling which is quite not the same approach we used earlier in this document).

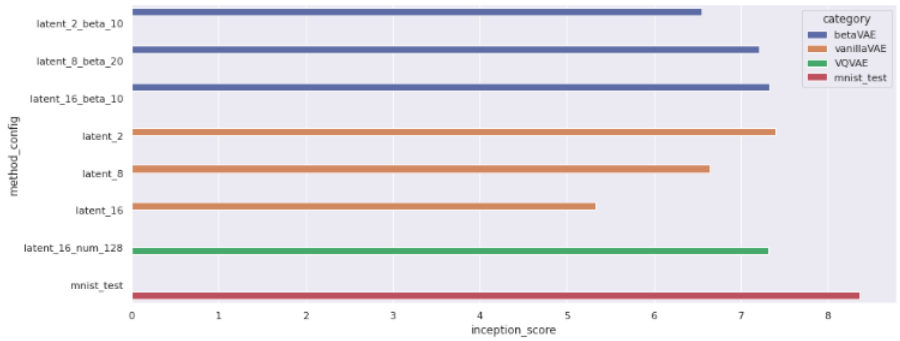

Figure 7: Inception score measured using 1K sample using pretrained Resnet-18

6. Conclusions

After training the three models on the MNIST dataset , we used them to new images and For numerical results, we used the inception score (IS) to measure the quality and the variety of the generated samples. For further work , we aim to test other variations of VAE such as including the Wasserstein distance in VAE and Deep Feature Consistent VAE. For evaluation, we tried to implement an agnostic metric that evaluate the utility of generated sampled with regard to the training of classification models and compare the accuracy of models using different synthetic datasets Ravuri (2019).

References

Stanford course of deep generative models. https://deepgenerativemodels.github.io/notes. Accessed: 2022-11-26.

An implementation in keras of vq-vae. https://keras.io/examples/generative/vq_vae/. Accessed: 2022-11-26.

A beginner's guide to variational methods: Mean-field approximation. https://blog.evjang.com/2016/08/variational-bayes.html. Accessed: 2022-11-26.

vector quantized variational autoencoder. https://ameroyer.github.io/portfolio/2019-08-15-VQVAE/. Accessed: 2022-11-26.

Kingma et Welling. An introduction to variational autoencoders. 2019.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Diederik P. Kingma and Max Welling. Vae: Auto-encoding variational bayes. 2013. URL https://doi.org/10.48550/arXiv.1312.6114.

Vinyals Oriol Oord, Aaron V. and Koray Kavukcuoglu. Neural discrete representation learning. 2017. URL https://doi.org/10.48550/arXiv.1711.00937.

Vinyals O. Ravuri, S. Cas: Classification accuracy score for conditional generative models. 2019. URL https://doi.org/10.48550/arXiv.1905.10887.

Oord Aaron V. Razavi, Ali and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. 2019. URL https://doi.org/10.48550/arXiv.1906.00446.