Auto-Encoding Variational Bayes

VILHES Samy

2024

Table of contents

1 Introduction

Bayesian inference involves computing the posterior distribution, $p(z \mid x)$, given some data x. The posterior is computed using the following Bayesian formula:

$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

However, in many models, the likelihood and prior distribution can be complex functions. This makes computing the posterior distribution often infeasible.
The "Auto-Encoding Variational Bayes" paper by Diederik Kingma and Max Welling proposes a solution to this Bayesian inference problem. The main idea is to learn an approximation of the true posterior $p(z \mid x)$. They introduce Variational Autoencoders (VAE), a generative model capable of approximating the posterior using variational inference and neural networks.

2 Solving the problem

2.1 Notations

We consider:

- A dataset $X = \{x_1, x_2, \ldots, x_N\}$ of N i.i.d. variables
- z: a latent and continuous random variable that generates x
- The intractable posterior: $p_\theta(z \mid x)$
- The prior: $p_\theta(z)$
- The likelihood: $p_\theta(x \mid z)$
- The evidence or marginal likelihood: $p_\theta(x)$
- The approximate distribution $q_\phi(z \mid x)$

The issue is that the evidence is oftentimes **intractable**, meaning that finding the solution of a problem is infeasible and requires an excessive amount of ressources.

2.2 Variational Lower Bound of the Margins Likelihood

The log marginal likelihood is :

$$log(p_\theta(x)) = D_{KL}(q_\phi(z \mid x) \mid p_\theta(z \mid x)) + \mathcal{L}(\theta, \phi; x)$$

With:

$$\mathcal{L}(\theta, \phi; x) = E_{q_\phi(z|x)}[log(p_\theta(x \mid z)] - D_{KL}(q_\phi(z \mid x) \mid p_\theta(z))$$

$\mathbf{E_{q_\phi(z|x)}}[\mathbf{log}(\mathbf{p}_\theta(\mathbf{x} \mid \mathbf{z})]$ is the **reconstruction cost**.
$\mathbf{D_{KL}}(\mathbf{q}_\phi(\mathbf{z} \mid \mathbf{x}) \mid \mathbf{p}_\theta(\mathbf{z}))$ is the **penalty**. It makes sure that $q_\phi(z \mid x)$ doesn't deviate too much from $p_\theta(z)$).

The KL divergence is non-negative. Therefore:

$$log(p_\theta(x)) \geq E_{q_\phi(z|x)}[log(p_\theta(x \mid z)] - D_{KL}(q_\phi(z \mid x) \mid p_\theta(z)) = \mathcal{L}(\theta, \phi; x)$$

The term on the right $\mathcal{L}(\theta, \phi; x)$ is called the variational lower bound or **ELBO**. The term on the left is the log-likelihood of the datapoint. By maximizing the lower bound, we maximize the log-likelihood since it is intractable.

2.3 Some useful tips

We have the right to do some supposition for our latent space and variables. We can decide that the prior $p_\theta(z)$ and the approximate posterior $q_\phi(z \mid x)$ both follow Gaussians distributions. Moreover, if $p_\theta(z) \sim \mathcal{N}(0, 1)$, we get:

$$-KL(q_\phi(z \mid x) \mid p_\theta(z)) = \sum_{j=1}^{J} \frac{1}{2}\left[1 + log(\sigma_j^2) - \sigma_j^2 - \mu_j^2\right]$$

with J the dimension of the latent space and j the index.

3 Architecture of the Variational Autoencoder

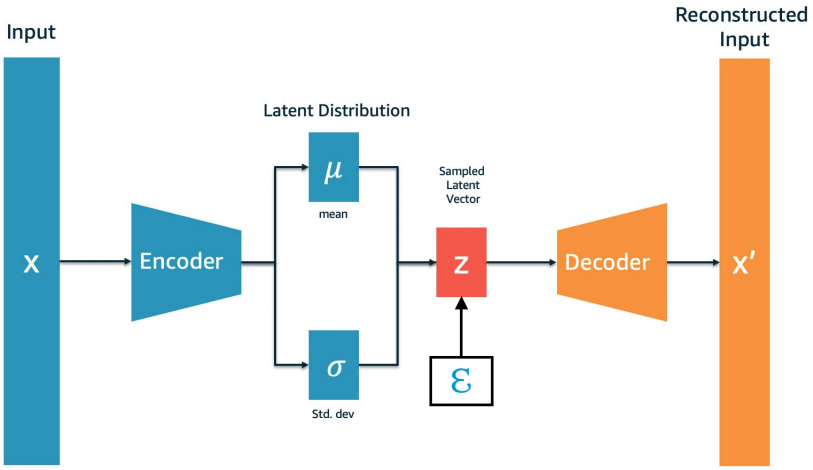


Figure 1: VAE architecture schema

3.1 The encoder and decoder

The encoder is a neural network that aims to reduce the dimensions of the input data. It plays the role of $q_\phi(z \mid x)$. We made earlier the assumption that this distribution $q_\phi(z \mid x^{(i)}) \sim \mathcal{N}(\mu^{(i)}, {\sigma^{(i)}}^2)$. By doing so, the output of the encoder is stochastic. To bypass this problem, we use the reparameterization trick : the output will be μ and σ

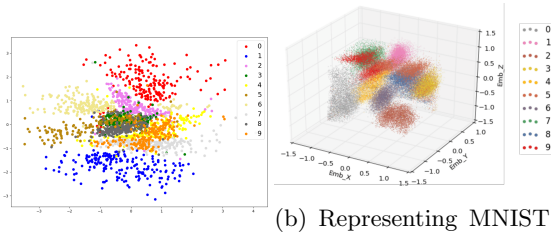
The input of the decoder, which is a neural network, is the latent vector z. Its goal is to generate, from this latent represention, a new output very similar to the input of the encoder.

3.2 The loss function

We want to maximize the log-likelihood which is intractable. To do so, we maximize the variational lower bound $\mathcal{L}(\theta, \phi, x)$. In order to maximize it, we juste have to minimize its opposite (i.e. minimize $-\mathcal{L}(\theta, \phi, x)$). We have an analytical expression of the regularizer term that only needs the output of the encoder μ and σ. For the reconstruction term we can choose a known loss, for example the MSE-loss or the Binary Cross-Entropy Loss.
The loss we used in all of our models is:

$$\mathcal{L} = -\sum_{j=1}^{J} \frac{1}{2}\left[1 + log(\sigma_j^2) - \sigma_j^2 - \mu_j^2\right] + BCELoss(x_{input}, x_{reconstructed})$$

4 Experimentation



(a) 2D representation (b) Representing MNIST digits in 2D and 3D spaces

Figure 2: MNIST

We initially trained the VAE using the architectures outlined in the paper, employing only linear layers without convolution, on both the MNIST and FreyFace datasets. For visualization, we showcased the 2D and 3D representations of digits (equally distributed) from the MNIST dataset.

From this graphic, we can see that the encoder has formed clusters of numbers all by itself. The encoder captured 2 features that distinguish the numbers from one another, but due to the low dimension of the latent space, it has a hard time distinguishing certains numbers. For instance, **8** and **3** are superposed in the graphic, and this problem is reflected when doing tests.
However, when using a 3 dimensional latent space, the formation of distinct clusters is a lot more apparent.

(a) Some examples of Frey face images

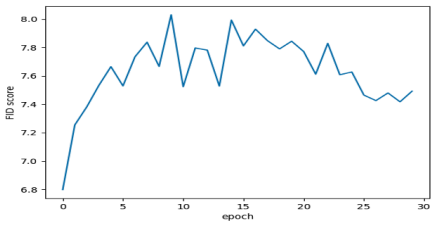
(b) Some faces generated by our VAE

Figure 3: Frey Face

The Frey Face dataset is a series of images of Brendan Frey's face taken from sequential frames of a video. These images represent the face of the same person, but with differents viewing angles and facial expressions.
We found that the VAE was capable of generating realistic face images, but we also noted a slight loss of sharpness compared to the original images. This may be due to the probabilistic nature of the VAE and the latent representation that is used for image generation.

The most recent dataset utilized in our study is the CelebA Dataset, comprising numerous colored facial images. We made slight adjustments to the architecture by employing straightforward CNNs for both the encoder and decoder. The model was trained for 30 epochs. Results are similar as for the Frey Face expermitentation.

(a) Some examples of input (above) images and reconstructed (below) by the VAE



(b) Evolution of the FID Score by epoch

Figure 4: CelebA

Similar to the findings with the Frey Face images, the reconstructed images appear blurry owing to the fundamental architecture of this convolutional VAE. Additionally, upon examining the FID scores across epochs, it is evident that the model's performance plateaus after epoch 9.

5 Bibliography

Diederik P. Kingma and Max Welling, ***Auto-encoding Variational Bayes***, https://arxiv.org/pdf/1312.6114.pdf

Wikipedia, ***Autoencoders***, https://en.wikipedia.org/wiki/Variational_autoencoder

Wikipedia, ***Variational Autoencoders***, https://en.wikipedia.org/wiki/Variational_autoencoder

Wikipedia, ***Kullback-Leibler divergence***, https://en.wikipedia.org/wiki/Kullback-Leibler_divergence

Diederik P. Kingma, ***Auto-Encoding Variational Bayes presentation***, https://www.youtube.com/watch?v=rjZL7aguLAs&t=66s

Sanna Persson, ***Variational Autoencoders***, https://sannaperzon.medium.com/paper-summary-variational-autoencoders-with-pytorch-implementation-1b4b23b1763a

Emma Benjaminson, ***The Reparameterization trick***,https://sassafras13.github.io/ReparamTrick/

Djork-Arné Clevert, ***MNIST representation***,https://www.researchgate.net/figure/Representation-of-the-MNIST-dataset-using-the-3-dimensional-latent-space-learned-by-the_fig3_321307206

Sayak Paul, ***Reparameterization trick in VAE***, https://towardsdatascience.com/reparameterization-trick-126062cfd3c3

Dr Stephen Odaibo, ***Variational Inference and Derivation of the Variational Autoencoder (VAE) Loss Function: A True Story***, https://medium.com/retina-ai-health-inc/variational-inference-derivation-of-the-variational-autoencoder-vae-loss-function-a-true-story-3543a3dc67ee

Jeremy Jordan, ***Variational autoencoders***,https://www.jeremyjordan.me/variational-autoencoders/