Variational Autoencoders

Sergio Alberto De León Martínez
CIMAT

February 14, 2024

Abstract

This work explores Variational Autoencoders (VAEs) from a probabilistic perspective and demonstrates their application on the MNIST dataset. The theoretical foundations of VAEs are introduced, including variational inference and training, and experiments are presented to evaluate the quality of reconstruction, visualization, interpolation, and robustness. Additionally, conditional generation and anomaly detection are examined. The study highlights the effectiveness of VAEs in various applications.

1 Introduction

A generative model is one that can simulate a random variable from a learned distribution, examples in (Kingma and Welling 2014 [1]). In 2014, the use of Variational Autoencoders (VAE) and Generative Adversarial Networks (GANs) achieved the first practical deep neural networks capable of learning generative models of complex data, such as images [2]. Since then, VAEs have been used in a variety of applications ranging from popular cases such as OpenAI's Dall-E model [3], to applications in physics for classifying phase transitions in the Ising model [4], and even in medicine for inferring the age of a group of patients with magnetic resonance imaging [5]. In this work, we will introduce the Variational Autoencoder model from a probabilistic perspective and then show and explain various experiments with the MNIST dataset.

2 Variational Inference and Neural Networks

Consider a dataset x_i, whose dimensionality is typically very large, making its practical handling very costly. We would like to encode our dataset in such a way that information loss is minimized. Let's denote z_i as the encoded variables, which are often referred to as latent variables. We will assume a probabilistic model for the variables x_i, z_i. The process we would like to follow to obtain samples is as follows, for each data point x_i:

- Write the latent variable $z_i \sim p(z)$
- Generate a sample $x_i \sim p(x|z)$

Recall that the joint probability of the model can be written as $p(x, z) = p(x|z)p(z)$, where the distribution of $x|z$ can be deduced from the data. For example, consider a 28×28 pixel black and white image of handwritten numbers. Then, each pixel can be thought of as a random variable that follows a Bernoulli distribution with a parameter to be determined. Similarly, the latent variables are expected to follow a Gaussian distribution with parameters also to be determined. In this context, we would like to infer the values of the latent variables given the dataset as accurately as possible, this is $p(z|x)$ from Bayes' theorem, we have

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}.$$

The term $p(x)$ is known as the evidence and can be obtained by marginalizing

$$p(x) = \int p(x|z)p(z)\ dz$$

The disadvantage is that this integral is computationally expensive to calculate. Instead, the alternative is to optimize a lower bound.

2.1 Variational Lower Bound

Instead of directly calculating $p(z|x)$, it can be approximated with a family of distributions $q_\phi(z|x)$ where the parameterization ϕ is given by the weights and biases of a neural network. Continuing with this idea, the distribution of $x|z$ can also be learned from a neural network, that is, consider the family of distributions $p_\theta(x|z)$ where again, θ

represents the weights and biases of a neural network. The way the model will be trained is through the paradigm of maximizing the likelihood $p_\theta(x)$ or alternatively optimizing a lower bound. For this, consider the following calculation.

We would like to find a lower bound to $\log p_\theta(x)$:

$$\begin{aligned}\log p_\theta(x) &= \mathbb{E}_{z\sim q_\phi(z|x)} \log p_\theta(x) \\ &= \mathbb{E}_{z\sim q_\phi(z|x)} \log \frac{p_\theta(x,z)}{p_\theta(z|x)} \\ &= \mathbb{E}_{z\sim q_\phi(z|x)} \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \frac{q_\phi(z|x)}{p_\theta(z|x)} \\ &= \mathbb{E}_{z\sim q_\phi(z|x)} \frac{p_\theta(x,z)}{q_\phi(z|x)} + \mathbb{E}_{z\sim q_\phi(z|x)} \log \frac{q_\phi(z|x)}{p_\theta(z|x)}\end{aligned}$$

The first term is known as the variational lower bound or the evidence lower bound (ELBO).

$$\mathcal{L}(\phi,\theta) = \mathbb{E}_{z\sim q_\phi(z|x)} \frac{p_\theta(x,z)}{q_\phi(z|x)}$$

The second term is known as the Kullback-Leibler divergence. In general, we have that

$$D_{KL}(q||p) = \mathbb{E}_{z\sim q} \log \frac{q(z)}{p(z)}$$

Remarks:

- $D_{KL}(q||p) \neq D_{KL}(p||q)$
- Mide qué tan lejos está p de q
- $D_{KL}(q||p) \geq 0$ (y 0 si y sólo si $p = q$)

The last observation follows from Jensen's inequality, which states the following

$$\varphi\left(\mathbb{E}\{X\}\right) \leq \mathbb{E}\{\varphi(X)\}.$$

For φ a convex function. Since the K-L divergence is defined positive, then the evidence lower bound (ELBO) is indeed a lower bound of the likelihood:

$$\log p_\theta(x) \geq \mathbb{E}_{z\sim q_\phi(z|x)} \log \frac{p_\theta(x,z)}{q_\phi(z|x)} = \mathcal{L}_{\theta,\phi}(x)$$

The ELBO function, $\mathcal{L}_{\theta,\phi}(x)$, can be broken down as follows:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(x) &= \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log p_\theta(x|z)\right] + \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log \frac{p(z)}{q_\phi(z|x)}\right]\end{aligned}$$

Consider the first term, if $p_\theta(x|z) \sim \mathcal{N}(x; G_\theta(z), \eta I)$, then

$$\begin{aligned}\log p_\theta(x|z) &= \log \left[\frac{1}{\sqrt{(2\pi)^d \eta^d}} \exp\left(-\frac{1}{2}(x - G_\theta(z))^T \eta I (x - G_\theta(z))\right)\right] \\ &= -\frac{1}{2}||x - G_\theta(z)||^2 + \log\left[\frac{1}{\sqrt{(2\pi)^d \eta^d}}\right]\end{aligned}$$

So, the expectation of the logarithm of $p_\theta(x|z)$ is the mean of the mean squared error of reconstruction under the encoder model. The second term can be thought of as a regularization term, and furthermore, let's see that

$$\mathbb{E}_{z\sim q_\phi(z|x)} \log \frac{p(z)}{q_\phi(z|x)} = -D_{KL}(q_\phi(z|x)||p(z))$$

Thus, maximizing ELBO $\mathcal{L}_{\theta,\phi}$ implies:

- Approximately maximizing $\log p_\theta(x|z)$.
- Minimizing the K-L divergence between $q_\phi(z|x)$ and $p_\theta(z|x)$ making q_ϕ better.

Here is the English translation of your subsection:

2.2 Training

Instead of optimizing

$$\sum_{i=1}^{n} \log p_\theta(x_i)$$

Optimize $\sum_i \mathcal{L}_{\theta,\phi}(x_i)$ with

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$$

The parameters of the inference model are shared among the data. Now, calculating $\nabla_{\theta,\phi}\mathcal{L}_{\theta,\phi}(x_i)$ is intractable, but some unbiased estimators can be deduced, for θ it is simple.

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\theta,\phi}(x) &= \nabla_\theta \mathbb{E}_{z\sim q_\theta(z|x)} \left[\log p_\theta(x,z) - \log q_\theta(z|x)\right] \\ &= \mathbb{E}_{z\sim q_\theta(z|x)} \nabla_\theta \log p_\theta(x) \\ &\approx \nabla_\theta \log p_\theta(x) \quad \text{with} \quad z \sim q_\theta(z|x)\end{aligned}$$

But for ϕ it's not the same, since as it changes, $q_\phi(z|x)$ changes as well. However, remember that $q_\phi(z|x) \sim \mathcal{N}(z; \mu(x), \sigma(x)\mathbf{I})$, then

$$q_\phi(z|x) = \mu_(x) + \sigma_(x) \cdot \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

therefore

$$\begin{aligned}\mathcal{L}_\theta(x) &= \mathbb{E}_{z\sim q_\theta(z|x)}[\log p_\theta(x|z) - \log q_\phi(z|x)] \\ &= \mathbb{E}_{\epsilon\sim p(\epsilon)}[\log p_\theta(x|z) - \log q_\phi(z|x)]\end{aligned}$$

where in the last expression we have separated the random sources from the differentiable quantities to take it as an unbiased estimator for $\nabla_\phi \mathcal{L}_{\theta,\phi}(x)$.

3 Experiments

Next, we will show the results of a series of experiments that we conducted with the MNIST dataset, which consists of a series of images of 28×28 pixels. The number of hidden layers used and the dimension of the latent space were 256 and 200 respectively. The training was done over 50 epochs.

3.1 Quality of Reconstruction

The first thing we did was to pass a set of test images through the VAE and then calculate the average mean squared error between the original and reconstructed images, obtaining the following value:

Average MSE: 0.017317143331368 13

Below are some images to make a visual comparison.

Figure 1: Up: original image. Below: reconstructed image.

3.2 Visualization

For this experiment, we set the dimension of the latent space to 2, with the purpose of being able to visualize it and detect patterns that allow us to better understand how the network learns to distinguish between different classes.

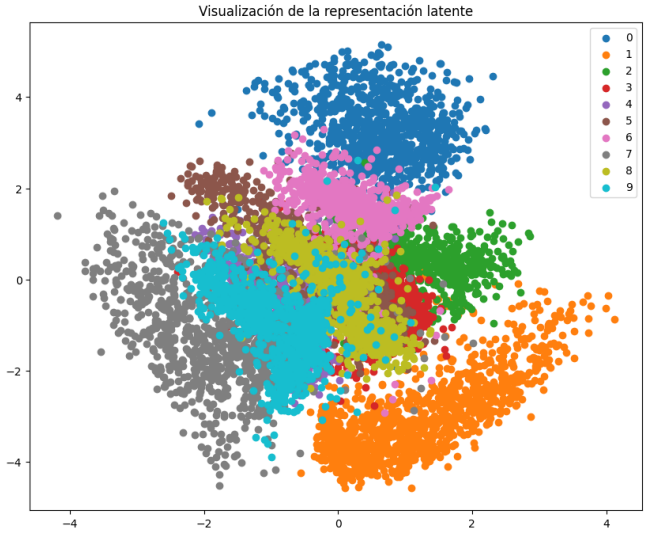

Figure 2: Latent space visualization.

As we can appreciate in Figure 2, the representations associated with the same class tend to cluster in the same region. However, we can also see how different clusters overlap, which motivates the following example.

3.3 Interpolation

Let's consider two images of different classes, for example, zero and one. Then, we pass these two images through the encoder to obtain their respective representations in the latent space (μ). Next, we interpolate linearly and use the decoder to obtain a series of images that allow us to see how the network transitions smoothly between classes, as shown in the image below.

Figure 3: Interpolation between classes.

3.4 Robustness

To test the robustness of the model, we considered a set of test images and applied Gaussian noise to them. Then, we passed them through the VAE for reconstruction and calculated the average mean squared error between the noise-free images and the reconstructed ones, obtaining the following result:

Average MSE (with noise): 0.021153034076999172

Figure 4 compares the noisy images with their respective reconstructed image.

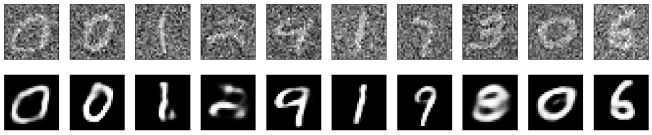

Figure 4: Images with Gaussian noise and their corresponding reconstruction.

As can be seen from the MSE calculation and the visual comparison, the model is relatively efficient even with the noise factor.

3.5 Anomaly Detection

To test the model's efficiency in detecting anomalous data, we considered a dataset significantly different from the MNIST dataset, for example, Fashion-MNIST. First, we took a test dataset from MNIST and calculated the empirical distribution of the mean squared error, setting a threshold to determine if a data point is anomalous at $\mu \pm 2\sigma$.

This means if the mean squared error of a data point falls outside this interval, it is considered anomalous. To validate this threshold, we took another subset of MNIST and obtained an accuracy of 98.4

3.6 Conditional Generation

Finally, we would like to generate images conditioned on a specific class. For this, we need to slightly modify the model to accept not only the image but also its corresponding label encoded in one-hot. Once this is done, we generate random vectors in the latent space and pass them through the encoder but conditioned to a particular class. The result is as follows.

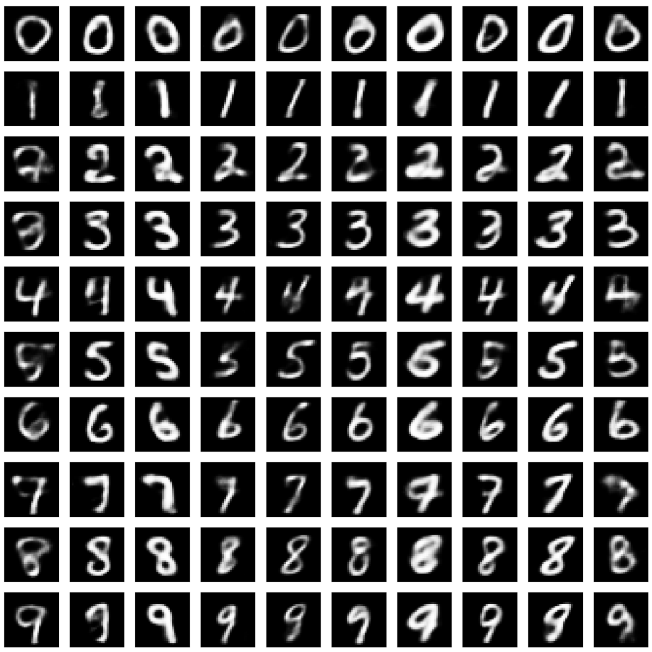

Figure 5: Conditioned images.

In Figure 5, different samples conditioned to each of the classes are shown, so we see different examples of 1's, 2's, etc. As a result, we can see an acceptable quality in the images.

4 Conclusion

The study on Variational Autoencoders highlights their versatility and efficacy in a wide range of applications. Their ability to generate useful and robust representations is emphasized, demonstrating their potential in areas such as image generation and anomaly detection. The success in these applications suggests that VAEs are powerful and flexible tools for the analysis and interpretation of complex data, paving the way for future research and developments in the field of artificial intelligence and machine learning.

References

[1] Kingma, Diederik P.; Welling, Max. (2013). *Auto-Encoding Variational Bayes*. Disponible en: https://doi.org/10.48550/arXiv.1312.6114.

[2] Wikipedia. *Inteligencia artificial generativa*. Disponible en: https://es.wikipedia.org/wiki/Inteligencia_artificial_generativa. Último acceso: 2023-12-08.

[3] Finetwork. *DALL-E: Inteligencia Artificial para la creación de imágenes*. Disponible en: https://blog.finetwork.com/dall-e-inteligencia-artificial/. Último acceso: 2023-12-08.

[4] Wetzel, Sebastian Johann. (2017). *Unsupervised learning of phase transitions: from principal component analysis to variational autoencoders. Phys. Rev. E 96, 022140*. Disponible en: https://doi.org/10.1103/PhysRevE.96.022140.

[5] Zhao, Qingyu; Adeli, Ehsan; Honnorat, Nicolas; Leng, Tuo; Pohl, Kilian M. (2019). *Variational AutoEncoder For Regression: Application to Brain Aging Analysis*. Disponible en: https://doi.org/10.48550/arXiv.1904.05948.

[6] Kingma, Diederik P.; Welling, Max. (2019). *An Introduction to Variational Autoencoders*. Foundations and Trends in Machine Learning. Vol. xx, No. xx, pp 1–18. DOI: https://doi.org/10.1561/XXXXXXXXX.

[7] Altosaar, Jaan. *What is a Variational Autoencoder (VAE)?*. Disponible en: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/. Último acceso: [2023-12-08].