Variational Autoencoder for Image Generation

Project of AI3611 Intelligent Perception and Cognition Practice, 2024 Spring, SJTU

Xiangyuan Xue (521030910387)

School of Electronic Information and Electrical Engineering

Abstract

Variational Autoencoder (VAE) is a generative model that is capable of learning the underlying distribution of the data and generating new samples which has been widely used in various fields. In this project, we research on applying VAE for image generation. We derive the mathematical formulation of VAE and implement a simple VAE model for generating handwritten digits on the MNIST dataset. We investigate the influence of the latent space on the generation process and possible ways to minimize the reconstruction error to improve the quality of the generated images. We also present our findings and insights gained from the experiments.

1 Introduction

Variational Autoencoder (VAE) [5] is a generative model that is capable of learning the underlying distribution of the data and generating new samples. It is a powerful tool for unsupervised learning and has been widely used in various fields including image generation, data compression, and representation learning. With the rapid development of deep learning, multiple variants of VAE have been proposed to improve the performance and stability of the model, such as Conditional Variational Autoencoder (CVAE) [9], Beta Variational Autoencoder (β-VAE) [1], and Vector Quantized Variational Autoencoder (VQ-VAE) [10]. With the advent of diffusion models [2], VAE has been one of the most popular network architectures for generative modeling.

VAE is based on the variational inference framework [3] following the well-known encoder-decoder architecture [8]. The encoder maps the input data to a latent space, while the decoder reconstructs the input data from the latent space. The key idea of VAE is to learn the distribution of the latent space by maximizing the evidence lower bound (ELBO) of the data likelihood. Reparameterization trick [5] is also introduced to make the optimization of the ELBO feasible by decoupling the stochastic sampling process from the optimization process. The details will be further discussed in Section 2.

In this project, we will focus on exploring using VAE for image generation, which is one of the mainstream applications of VAE. We will implement a simple VAE model and train it on the MNIST dataset [6] to generate handwritten digits. We will investigate how the latent space influences the generation process and how to minimize the reconstruction error to improve the quality of the generated images. The experiments conducted will be described in Section 3.

2 Method

The variational autoencoder follows the basic structure of the autoencoder [8], consisting of an encoder and a decoder. The encoder compresses the input data into a latent space, while the decoder reconstructs the input data from the latent space. The variational autoencoder introduces a probabilistic model to the latent space, which allows the model to generate new data points by sampling from the latent space. The basic scheme of the variational autoencoder is shown in Figure 1.

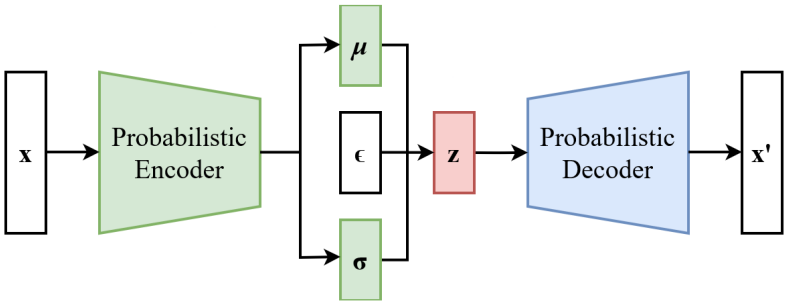

Figure 1: The basic scheme of the variational autoencoder. The model receives x as the input and compresses it into the latent space. The decoder samples z from the latent space as the input and produces x' as the output, which is expected to be as similar as possible to x.

The intuition behind the variational autoencoder is that the data points lie on a low-dimensional manifold in the high-dimensional space [5]. The encoder learns to map the data points to the low-dimensional latent space, thus capturing the underlying structure of the data, so that the decoder can reconstruct the data points from the latent space. Different from the traditional autoencoder, the variational autoencoder introduces randomness to make the model generative.

2.1 Problem Formulation

Let x and z denote the input data and corresponding latent variable respectively. If we parameterize the distribution by $p_\theta(x)$, we can define the prior $p_\theta(z)$, the likelihood $p_\theta(x|z)$ and the posterior $p_\theta(z|x)$. We expect to maximize the likelihood of the data x, which can be formulated as

$$p_\theta(x) = \int_z p_\theta(x|z)p_\theta(z)\,\mathrm{d}z \tag{1}$$

Actually, Equation 1 describes the optimization target of the decoder. For the encoder, the optimization target is described by the posterior $p_\theta(z|x)$, which can be formulated as

$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{\int_z p_\theta(x|z)p_\theta(z)\,\mathrm{d}z} \tag{2}$$

However, we should note that the integral in Equation 2 is intractable, so we will introduce another parameterized function $q_\phi(z|x)$ to approximate the posterior $p_\theta(z|x)$, namely

$$q_\phi(z|x) \approx p_\theta(z|x) \tag{3}$$

Then the encoder can be described by $q_\phi(z|x)$, and we only need to minimize the Kullback-Leibler divergence between $q_\phi(z|x)$ and $p_\theta(z|x)$, namely $D_{\mathrm{KL}}(q_\phi(z|x)||p_\theta(z|x))$. We will denote the encoder and the decoder by E_ϕ and D_θ respectively in the following sections.

2.2 Evidence Lower Bound

To optimize the autoencoder, we employ the Kullback-Leibler divergence $D_{\mathrm{KL}}(q_\phi(z|x)||p_\theta(z|x))$ as the loss function to squeeze $q_\phi(z|x)$ towards $p_\theta(z)$. We can expand the loss function as

$$\begin{aligned} D_{\mathrm{KL}}(q_\phi(z|x)||p_\theta(z|x)) &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right] \\ &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(z|x)p_\theta(x)}{p_\theta(x,z)}\right] \\ &= \log p_\theta(x) + \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(x,z)}\right] \end{aligned} \tag{4}$$

Then we can define the evidence lower bound (ELBO) as

$$L_{\theta,\phi}(x) \triangleq \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] = \log p_\theta(x) - D_{\mathrm{KL}}(q_\phi(\cdot|x)||p_\theta(\cdot|x)) \tag{5}$$

which indicates that maximizing the ELBO is equivalent to simultaneously maximizing $\log p_\theta(x)$ and minimizing $D_{\mathrm{KL}}(q_\phi(z|x)||p_\theta(z|x))$, that is to say, maximizing the likelihood of the observed data $p_\theta(x)$ and minimizing the divergence between the approximate posterior $q_\phi(z|x)$ and the exact posterior $p_\theta(z|x)$ at the same time. An equivalent but more convenient form is given by

$$L_{\theta,\phi}(x) \triangleq \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x|z)\right] - D_{\mathrm{KL}}(q_\phi(\cdot|x)||p_\theta(\cdot)) \tag{6}$$

It is natural to assume that the input data obey an independent multivariate Gaussian distribution centered at $D_\theta(z)$. In addition, we can specify that the approximate posterior $q_\phi(z|x)$ and the prior $p_\theta(z)$ are also Gaussian distributions, which can be written as

$$\begin{aligned} z|x &\sim \mathcal{N}(\mu_\phi(x), \sigma_\phi^2(x)I) \\ z &\sim \mathcal{N}(0, I) \end{aligned} \tag{7}$$

Then the ELBO can be reformulated into the loss function given by

$$L_{\theta,\phi}(x) = -\frac{1}{2}\mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\|x - D_\theta(z)\|_2^2\right] - \frac{1}{2}\left(\|\mu_\phi(x)\|_2^2 + n\sigma_\phi^2(x) - 2n\log\sigma_\phi(x)\right) + C \tag{8}$$

where the first term represents the reconstruction loss, the second term represents the KL divergence, C is a constant value and d is the dimension of the latent space.

2.3 Reparameterization Trick

According to Equation 8, the optimization target is given by

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} L_{\theta,\phi}(x) \tag{9}$$

We expect to solve the problem efficiently by gradient descent. However, the randomness variable z is involved in the optimization, which makes it impossible to back propagate the gradient. Reparameterization trick [5] is introduced to address this issue. Let $\epsilon \sim \mathcal{N}(0, I)$ be a standard Gaussian variable generator, and we can construct z from ϵ, which is formulated as

$$z = \mu_\phi(x) + \sigma_\phi(x) \cdot \epsilon \tag{10}$$

Then the randomness is transferred from z to ϵ, making it possible to back propagate the gradient.

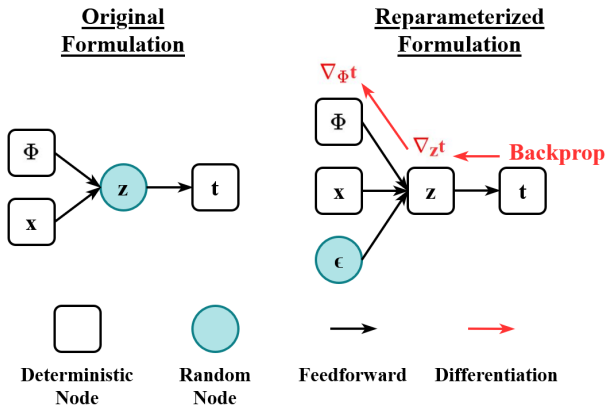

Figure 2: The scheme of the reparameterization trick. The randomness variable ϵ is injected into the latent space z as external input. In this way, it is possible to back propagate the gradient without involving stochastic variable during the update.

Note that the approximate posterior $q_\phi(z|x)$ should be modified, which is specified as

$$q_\phi(z|x) = -\frac{1}{2}\|\epsilon\|_2^2 - \log|\det\sigma_\phi(x)| - \frac{n}{2}\log 2\pi \tag{11}$$

where $\det$ represents the determinant of a matrix and n is the dimension of the latent space.

3 Experiments

We have introduced the overall architecture of VAE in Section 2. In this section, we will implement the model and apply it for image generation. We train the model on the MNIST dataset to make it capable of generating handwritten digits and investigate how the latent space influences the generation process. In addition, we explore using β-VAE to minimize the reconstruction loss, which is expected to improve the quality of generated images.

3.1 Experiment Setup

We employ the MNIST dataset [6] to train the VAE model and expect it to generate handwritten digits. The dataset consists of $60,000$ training images and $10,000$ testing images, where the testing images are only used to generate samples and evaluate the model performance.

In the default setting, we flatten the 28×28 grayscale images into a 784-dimensional vector. Both the encoder and the decoder are implemented as Multi-Layer Perceptrons (MLP) [8] with two hidden layers of 256 units. The dimension of the latent space is set to 64, which is significantly smaller than the input dimension while still preserving the information. We train the model for 300 epochs with a batch size of 256 using Adam optimizer [4] with a learning rate of 10^{-4}.

To investigate how the latent space influences the generation process, we further reduce the dimension of the latent space to 1 and 2 respectively, where other settings remain the same. We present the generated images with different latent vectors and visualize the distribution of latent vectors in the latent space. To minimize the reconstruction loss on the testing set, we apply β-VAE and conduct experiments with $\beta \in \{0.1, 0.5, 1, 2, 10\}$. We compare the reconstruction loss in different settings.

3.2 Latent Space

We set the dimension of the latent space to 1, 2 and 64 and visualize the results in Figure 3, 4 and 5 respectively. We present the generated images of handwritten digits with the latent vectors of different values. For the 64-dimensional latent space, the images are sampled from the testing set. We also visualize the distribution of latent vectors in the latent space categorized by their labels. For the 64-dimensional latent space, we employ t-SNE [11] to reduce the dimensionality of the latent vectors to 2 for visualization. In addition, we present the loss curve during the training process to quantitatively demonstrate the ability of the model to learn the data distribution.

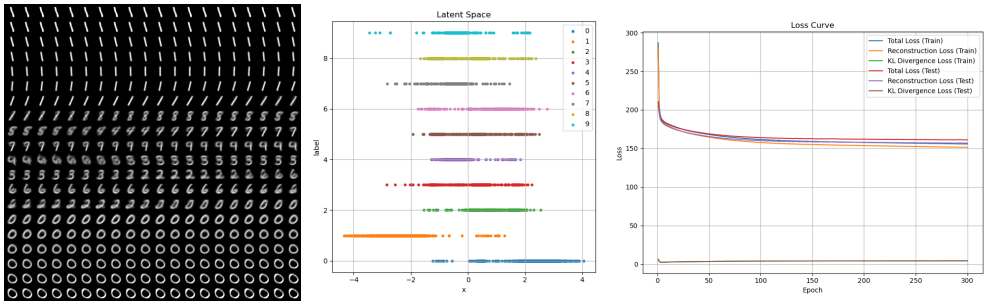

Figure 3: Visualization of the results with the dimension of the latent space set to 1. The left figure shows the generated images of handwritten digits, where the value of z ranges in $[-5, 5]$ from the top left to the bottom right. The middle figure shows the distribution of latent vectors in the latent space categorized by their labels. The right figure shows the loss curve during training.

From the figures we can see that with the dimension of the latent space increasing, the model is capable of generating handwritten digits with higher quality. With the dimension of the latent space set to 1, the reconstruction loss is relatively high and the generated images are blurry. We can see obvious aliasing over different labels in the latent space distribution, so some digits such as 4 and 5 cannot be clearly generated. With the dimension of the latent space set to 2, we can see higher quality and more diversity in the generated images, though the reconstruction loss is still not satisfactory. The latent space distribution shows that there are still some overlaps between different labels. With the dimension of the latent space set to 64, the model can generate images with high quality and

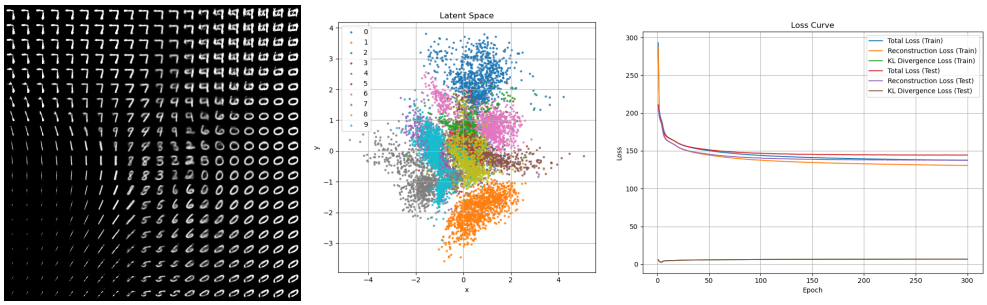

Figure 4: Visualization of the results with the dimension of the latent space set to 2. The left figure shows the generated images of handwritten digits, where the value of both axes ranges in $[-5, 5]$ from the top left to the bottom right. The middle figure shows the distribution of latent vectors in the latent space categorized by their labels. The right figure shows the loss curve during training.

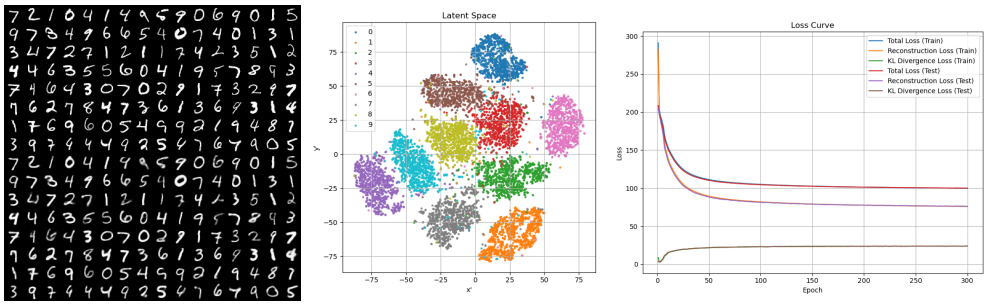

Figure 5: Visualization of the results with the dimension of the latent space set to 64. The left figure shows the generated images of handwritten digits where the latent vectors are sampled from the testing set. The middle figure shows the distribution of latent vectors whose dimensionality is reduced to 2 by t-SNE. The right figure shows the loss curve during training.

strong diversity and the reconstruction loss is significantly reduced. With dimensionality reduction in the latent space, we can see that the latent vectors are well separated and the distribution is more concentrated, which is close to a mixture of Gaussian distributions.

3.3 Reconstruction Loss

As a key metric to evaluate the quality of the generated images, the reconstruction loss is expected to be minimized. We have revealed that the dimension of the latent space can strongly affect the reconstruction loss by deciding the representation capability of the model. With a larger latent space, the information bottleneck is alleviated and the model can better capture the data distribution, which at the same time increases the complexity of the model. In this experiment, we try to reduce the reconstruction loss from another perspective by applying β-VAE [1], which introduces a hyperparameter β to balance the reconstruction loss and the KL divergence. To be specific, we can split the loss function of the vanilla VAE into the reconstruction loss and the KL divergence. Now we assign a weight β to the KL divergence term, which is specified as

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{BCE}} + \beta \cdot \mathcal{L}_{\text{KL}}, \tag{12}$$

If $\beta = 1$, the model is exactly the same as the vanilla VAE. A smaller β means a looser constraint on the latent space, which can be beneficial for reducing the reconstruction loss. We conduct experiments with $\beta \in \{0.1, 0.5, 1, 2, 10\}$ and compare the reconstruction loss in different settings. The results are presented in Table 1, including different latent space dimensions and β values.

From the table we can see that a larger latent space can significantly reduce the reconstruction loss, which is consistent with the results in the previous experiment. With β-VAE, we can further reduce the reconstruction loss by assigning a smaller weight to the KL divergence term. With the dimension of the latent space set to 64, the reconstruction loss decreases monotonically as β decreases, which indicates that the a smaller reconstruction loss can be achieved by loosening the constraint on the

Table 1: The reconstruction loss, the KL divergence and the total loss of the models with different latent space dimensions and β values computed on the testing set of MNIST dataset, where the reconstruction loss is minimized with $n = 64$ and $\beta = 0.1$.

n (Latent)	β (Weight)	$\mathcal{L}$ (Loss)		
		$\mathcal{L}_{\text{BCE}} \downarrow$	$\mathcal{L}_{\text{KL}} \downarrow$	$\mathcal{L}_{\text{VAE}} \downarrow$
1	1	156.82	4.22	161.04
2	1	137.70	6.80	144.49
64	1	76.23	23.80	100.03
64	0.1	**59.18**	77.68	**66.94**
64	0.5	69.50	34.30	86.65
64	2	87.29	15.46	118.20
64	10	143.37	**3.69**	180.22

latent space. However, if β becomes too small, the model will lose its diversity and fall back to the traditional autoencoder, which is contrary to the original intention of generation. Note that directly comparing the total loss is unfair since β modifies the scale of the KL divergence term.

4 Conclusion

In this project, we research on applying VAE for image generation. We derive the mathematical formulation of VAE and implement a simple VAE model for generating handwritten digits on the MNIST dataset. We investigate the influence of the latent space on the generation process and possible ways to minimize the reconstruction error to improve the quality of the generated images. The results show that VAE can generate realistic images with an interpretable latent space. We find that the dimension of the latent space significantly determines the quality of the generated images by controlling the information bottleneck. We also find that the reconstruction error can be further reduced by introducing β-VAE, where the trade-off between the reconstruction error and the KL divergence can be adjusted by the hyperparameter β.

Besides, we also have some interesting observations. From the generated images in Figure 4 we can see that the latent space to some extent has a disentangled structure, which means that different dimensions of the latent space control different aspects of the generated images. This result partly reveals the interpretability of the latent space. From the loss curves in Figure 3, 4 and 5 we can see that the losses on the training set and the validation set are extremely close, which indicates the strong generalization ability of the model. Such property is crucial for the model to be applied in the large-scale industrial scenarios such as Stable Diffusion [7].

References

[1] Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[3] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[8] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[9] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[10] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[11] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.