AI3611 Report of Expt.3 Image Generation with VAE

Zhiwei Ying

521030910376
yingzhiwei@sjtu.edu.cn

1 Introduction

Variational autoencoder (VAE) is a method for generative AI, whose goal is to build the bridge between normal distribution and data distribution. A detailed introduction of VAE is given in Section 2. The main code and the result are shown in Section 3. The source code is available at my Github repository[1].

2 Variational Autoencoder (VAE)

The variational autoencoder[1] is a classical unsupervised learning method. Each sample is matched to a Gaussian distribution by an encoder. This Gaussian distribution is sampled and then passed through a decoder to get a new sample with the hope that the gap between two samples is as small as possible.

2.1 Structure

The specific model can be seen in Figure 1 below.

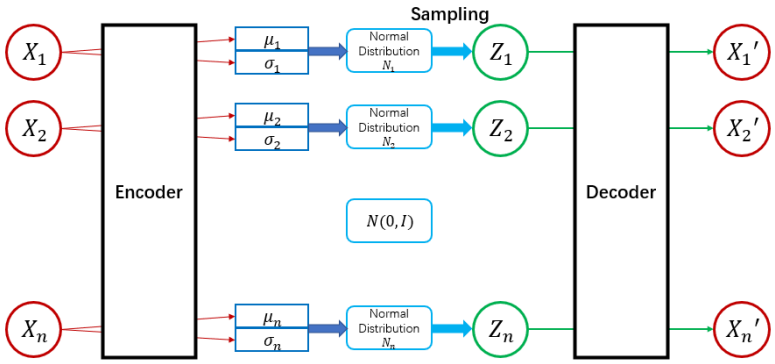
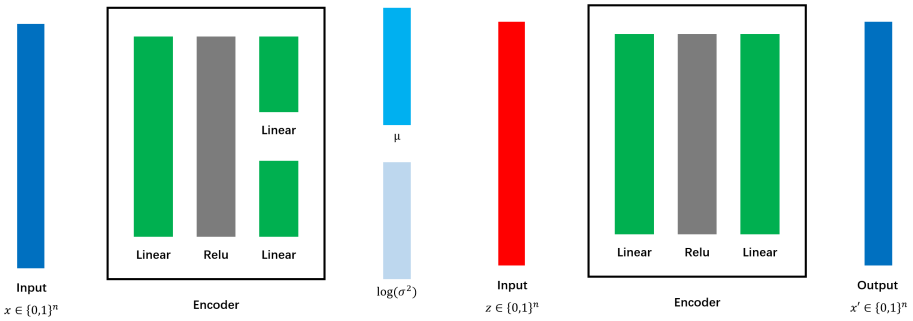


Figure 1: A simple illustration of a VAE model

It is worth noting that variational autoencoder is a kind of end to end learning, where the input is the raw data and the output is the final data result, and the features are automatically learned by the model without additional feature extraction.

[1] https://github.com/yzwykx/AI_Practice/tree/main/Image_Generation



(a) An example of the encoder (b) An example of the decoder

Figure 2: Example of the encoder and decoder

Now, we turn to details for the encoder and the decoder. In our example, the encoder contains three linear layers, one of which is applied straightly on the input and the other two produce the mean value and the variation, and a relu layer. The reason we generate $\log(\sigma^2)$ but straightforward σ is that we require positive and small value for variation. The decoder process can be regarded as a non-linear reconstruction of the sample result z. Sometimes, researchers take Bernoulli Distribution rather than Normal Distribution. The decision depends highly on the experiment result of reconstruction error.

2.2 Training

We denote the *evidence lower bound* [2]

$$L(x,\theta,\phi) \triangleq \mathbf{E}_{z\sim q_\phi(\cdot|x)}[\ln p_\theta(x|z)] - KL[q_\phi(z|x)||P(z)].$$

The goal to optimize during training is $\max_{\theta,\phi} F(x,\theta,\rho) = \frac{1}{|X|}\sum_{x\in X} L(x,\theta,\phi)$. In fact, the loss is just the negation of the evidence lower bound. Take two parts into consideration individually. The first term is indeed the negation of cross entropy, so that we can compute it with PyTorch easily. The second term is *Kullback-Leibler Divergence*. By definition,

$$KL[q_\phi(z|x)||P(z)] = \mathbf{E}[\log q_\phi(z|x) - \log P(z)].$$

For Normal Distribution, we have that

$$KL[\mathcal{N}(\mu,\sigma^2)||\mathcal{N}(0,1)] = -\frac{1}{2}\sum_{i=1}^{d}(1+\log(\sigma_i^2(x)) - \mu_i^2(x) - \sigma_i^2(x)).$$

For Bernoulli Distribution, we have that

$$KL[B(\rho)||B(\frac{1}{2})] = \rho\log(2\rho) + (1-\rho)\log(2(1-\rho)) = \log 2 - H(\rho),$$

where $H(\rho)$ is the cross entropy of Bernoulli Distribution with parameter ρ. In an epoch, we should update θ and ρ synchronously. That is,

$$\theta^{(t+1)} = \theta^{(t)} - \eta\nabla_{\theta^{(t)}}F(x,\theta^{(t)},\rho^{(t)}),$$
$$\rho^{(t+1)} = \rho^{(t)} - \eta\nabla_{\rho^{(t)}}F(x,\theta^{(t)},\rho^{(t)}).$$

[2] This subsection refers to the content in https://zhuanlan.zhihu.com/p/572698195.

Note that we cannot do back-propagation from the decoder to the encoder since there is a non-differentiable process- sampling between them. To make the full process differentiable, we can formalize the sampling process as $z = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$. It holds that $z \sim \mathcal{N}(\mu, \sigma^2)$.

With the theoretical analysis above, part of a training pseudocode should be like

Algorithm 1 Part of Training Pseudocode for VAE

```
repeat
    log_prob_p ← reconstruction_loss.sum() - running_avg
    log_prob_q ← -F.binary_cross_entropy(p, z).sum()
    reinforce_loss ← log_prob_p.detach()*log_prob_q - (log_prob_p*log_prob_q).detach()
    loss ← reconstruction_loss + kl_loss + reinforce_loss
    ...
    n_updates++
    running_avg += (reconstruction_loss.sum(dim=1).mean() - running_avg) / n_updates
until converged
```

2.3 Recent Researches

In the paper *Generating Diverse High-Fidelity Images with VQ-VAE-2*[2], researchers explore the use of Vector Quantized Variational AutoEncoder (VQ-VAE) models for large scale image generation and introduce a novel estimator of the variational lower bound, Stochastic Gradient VB (SGVB), for efficient approximate inference with continuous latent variables.

Researchers from ByteDance AI lab published the paper *Dispersed Exponential Family Mixture VAEs for Interpretable Text Generation*[3], in which a class of models for text generation using VAEs with a mixture distribution of exponential family, DEM-VAE, was proposed.

In the paper *Predictive variational autoencoder for learning robust representations of time-series data*[4], Wang et al. shows that standard VAEs are prone to learning spurious features and introduces a novel model selection metric based on smoothness over time of latent representations.

3 Experiments and Results

3.1 Structure of VAE

There are several kinds of encoder and decoder, e.g. linear layers or convolution layers. In my codes, I adopt the convolution layers. The encoder and decoder are defined as follows.

The encoder first use a convolution layer to extract characteristics from 28×28 to 14×14, then after an nonlinear layer Relu is another convolution layer from 14×14 to 7×7, after which is a Relu layer and then characteristics will be flattened and go through a linear layer and a Relu layer, during which it'll be reshaped to hid_{dim} dimensions. Two linear layers fc_{mu} and fc_{logvar} will get the predicted distribution parameters μ and δ from the hid_{dim}-dimension data.

The decoder is almost the reverse version of encoder. It recover the hid_{dim}-dimension data from the distribution parameters first, and take all reverse function to trying to rebuild a well-recognized image.

3.2 Training of VAE

We take Adam as the optimizer and take the sum of binary cross entropy and KL divergence as the training loss. During the training process, we plot the training loss and validation loss to determine

whether the network function well.

3.3 Reconstruction Error

The reconstruction error in training process and validation process are plotted in Figure 7 and Figure 4.

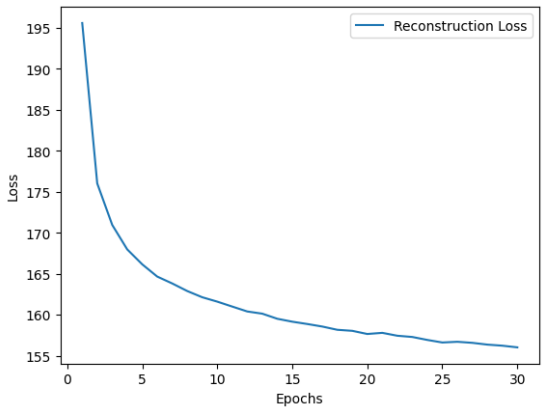


Figure 3: Reconstruction loss for $z_{dim} = 1$

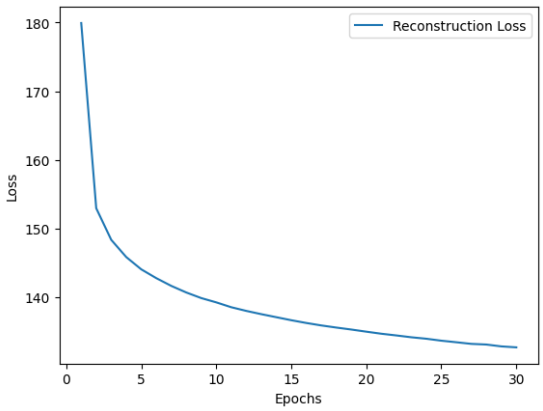


Figure 4: Reconstruction loss for $z_{dim} = 2$

3.4 Sampled Visualization

Set the hidden layer vector z's dimension to 1 and the result of the generated images corresponding to different z-values after the VAE training is as shown below in Figure 5.

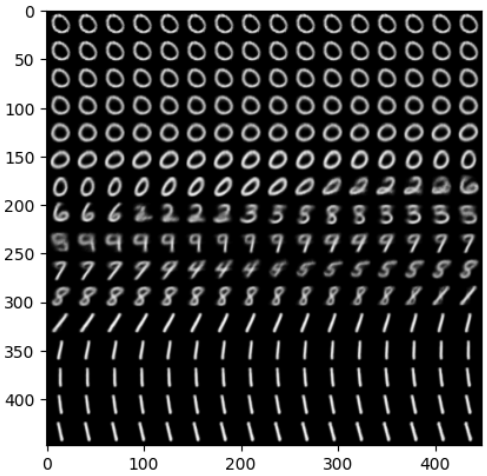


Figure 5: Caption

Then set the dimension of the hidden layer vector z to 2, the corresponding images generated in the interval $[-5, 5]$ of the two dimensions of the hidden layer vector are shown below in Figure 6.

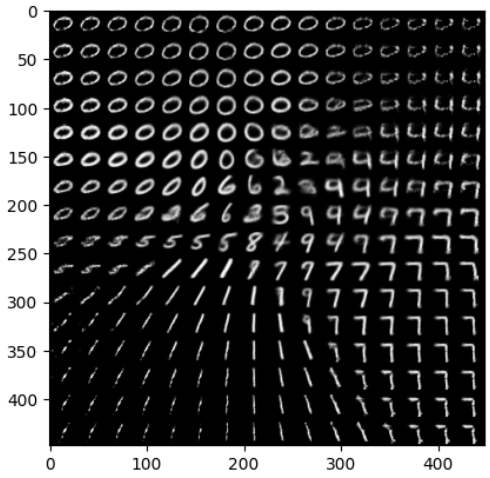


Figure 6: Caption

3.5 Best Hyper-parameters and Explorations

3.5.1 Data Augmentation

Data augmentation is an ordinary method to improve the effectiveness of models. Here, I introduce white noise to MNIST dataset and do some experiments. The reconstruction loss is as shown in Figure 8. We can see that whether adding white noise or not does not influence much on the result. There is slightly improvement but is not very important.

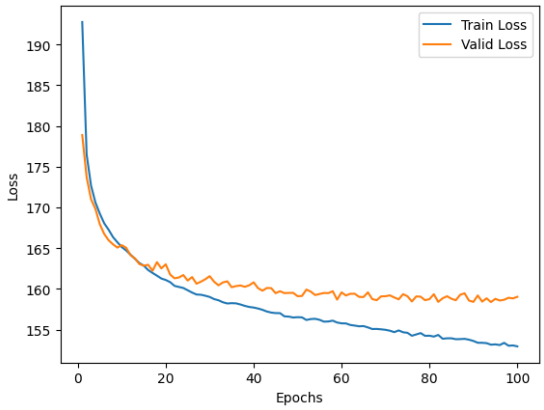


Figure 7: Loss for original data ($z_{dim} = 1$)

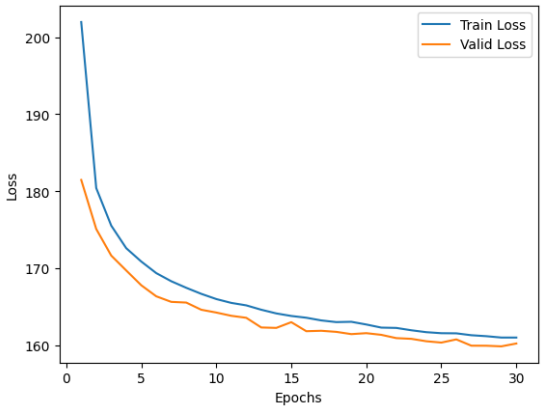


Figure 8: Loss for noised data ($z_{dim} = 1$)

3.5.2 Ratio of Binary Cross Entropy and KL Divergence

I test different ratio $[0.1, 0.5, 1.0, 2, 10]$ and the result are plotted below. The performance of $ratio = 0.5$ is the best.

References

[1] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 1

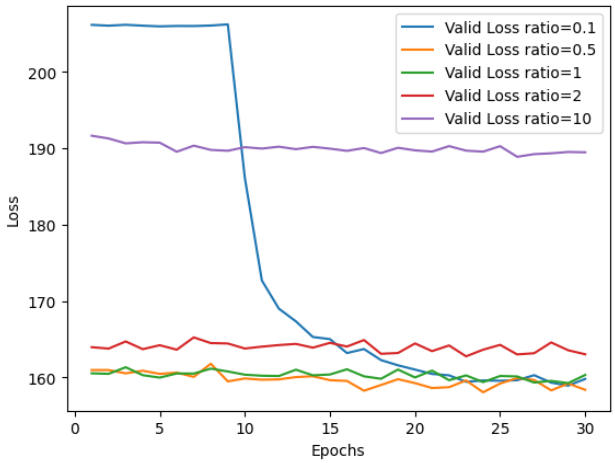


Figure 9: Affect of Different Ratio

[2] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2, 2019. 3

[3] Wenxian Shi, Hao Zhou, Ning Miao, and Lei Li. Dispersed exponential family mixture vaes for interpretable text generation, 2020. 3

[4] Julia Huiming Wang, Dexter Tsin, and Tatiana Engel. Predictive variational autoencoder for learning robust representations of time-series data, 2023. 3