國立陽明交通大學

NATIONAL YANG MING CHIAO TUNG UNIVERSITY

[EEEE30034] Cloud Computing and Big Data Analytics

Final Project -
HDD:Hierarchical Denoising Diffusion model

(Chia Kai, Yeh), (Xiang-Jun, You)
Student ID : 311511046, 311511056
kai.ee11@nycu.edu.tw, davidyou0121.ee11@nycu.edu.tw
National Yang Ming Chiao Tung University

Contents

Abstract

The goal of this work is to reduce the time required to train and generate high-resolution images using a denoising diffusion model. To achieve this, we propose a method that samples from low-resolution to high-resolution by splitting the training and sampling steps into two separate models: a low-resolution U-Net model and a high-resolution U-Net model. During training, the low-resolution model is fed resized versions of the original images, while the high-resolution model is fed the original size images. During sampling, the low-resolution model generates a low-resolution image from pure noise, which is then used as input for the high-resolution model. The ratio of clean input image to noise is controlled using a hyperparameter γ. We evaluate our method on the MNIST and Anime Face Datasets and show that it significantly reduces the time required to train and generate high-resolution images while maintaining good performance.

1 Motivation

Denoising diffusion model is a popular topic nowadays, when we do our homework, we find that if working on high-resolution images, it using a lot of time to train and generate. So we want to find a way to decrease the time by sampling from low-resolution to high-resolution.

Because of the high-resolution model's input is not pure noise, it is something from low-resolution image, so we don't need too many time steps on high-resolution model, this can reduce the time when training and sampling.

2 Method

2.1 HDD:Hierarchical Denoising Diffusion Model

Inspire by [2], **we split training and sampling step to two different models**. One is the low-resolution U-net model, and another is high-resolution U-net model(e.g. 14×14, 28×28). In low-resolution model(Figure 1), we only pooling image from 14x14 to 7x7 once in order to prevent too much information lost, and the high-resolution model(Figure 2) is similar to [1], we only modified the input argument to we need.

2.2 Training

When training, we use the standard way as same as DDPM [1] , and the data of low-resolution is just get from the resize of original images, and high-resolution is the original size images. Figure 3 shows the whole forward process.

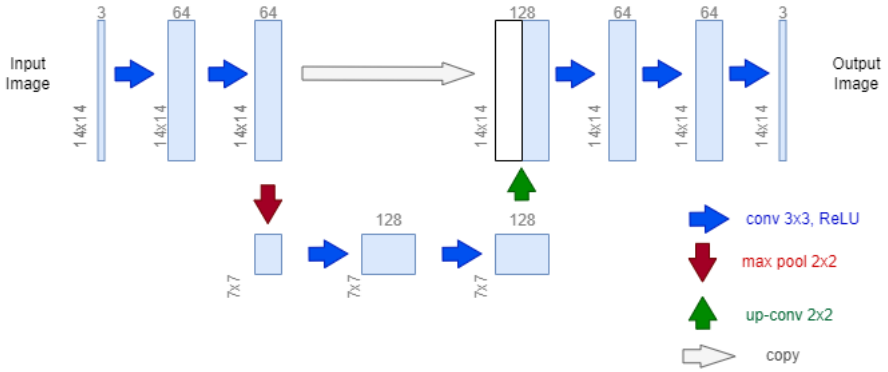

Figure 1: Low-resolution U-net

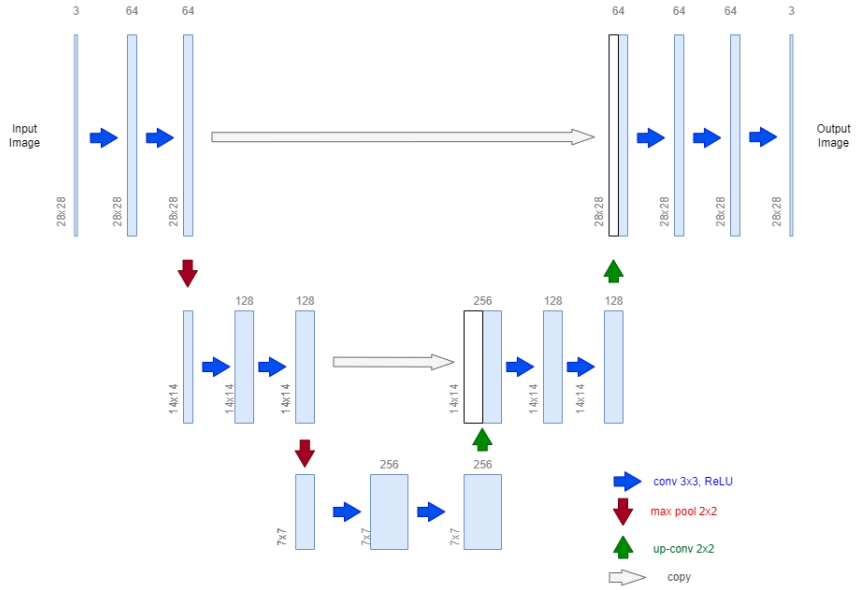

Figure 2: High-resolution U-net

2.3 Sampling

When sampling, first using low-resolution model from pure noise to generate low-resolution image, and using low-resolution model's output to be high-resolution model's input. Figure4 shows the whole backward process. Instead of adding clean image generated from low-resolution model, **we use a hyper-parameter** γ, which is the ratio of the clean input image and noise before we send to next sampling stage, the process can be written as:

$$\mathbf{x}_{HT} = \gamma \mathbf{x}_l + (1 - \gamma)\mathbf{z} \tag{1}$$

, where $\mathbf{x}_l$ is the resized images sampled from low-resolution model, and noise $\mathbf{z}$ is $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, and the output images $\mathbf{x}_{HT}$ then send to the next high-resolution model.

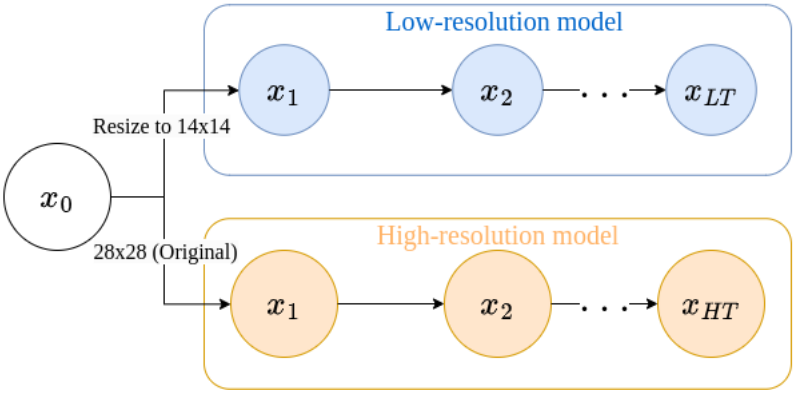

Figure 3: Forward process.

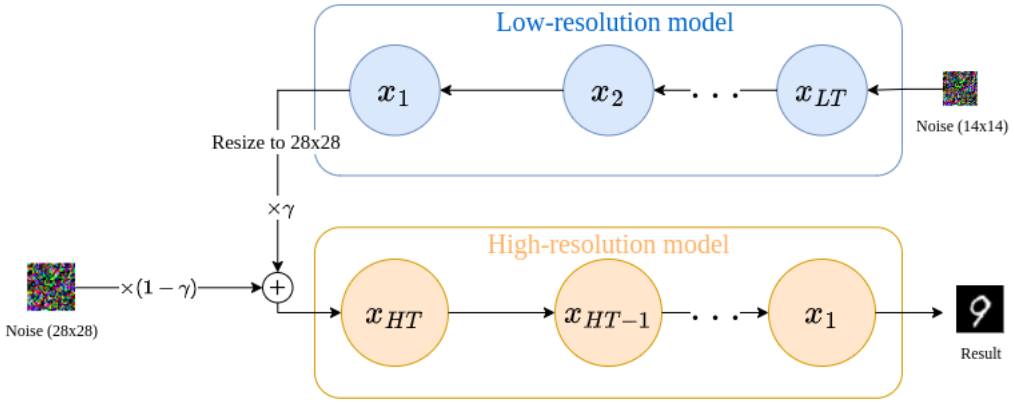

Figure 4: Backward process.

3 Experiment

3.1 Training settings

In our model, we set the time step $T = 500$ for each of our two models, which each model's time step is half of the original denoising diffusion process[1]. Also, the epochs is half of the process, so the total epochs of our two model is equals to original process(e.g. if original is 1000 epochs, out low-resolution and high-resolution model is 500 and 500.), then we can give a fair comparison. Specially, since the high-resolution images we sampled are not from a pure Gaussian noise,we set the low-resolution forward process variances to constants increasing linearly from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$ and high-resolution forward process variances to constants increasing linearly from $\beta_1 = 10^{-4}$ to $\beta_T = \mathbf{0.01}$. [1]

We set the noise ratio in Eq.(1) $\gamma = 0.3$. All models are train on NVIDIA RTX-2060 super GPU.

3.2 Dataset

3.2.1 MNIST

The MNIST (Modified National Institute of Standards and Technology) dataset is a widely used dataset for the training and testing of image processing systems. It consists of a training set of 60,000 grayscale images and a test set of 10,000 grayscale images. Each image is 28 pixels by 28 pixels, for a total of 784 pixels. The images are of handwritten digits, and the task is to classify them as one of the 10 digits (0 through 9). The MNIST dataset is often used as a benchmark for machine learning algorithms, and it is also a popular choice for beginners to learn image processing and machine learning.

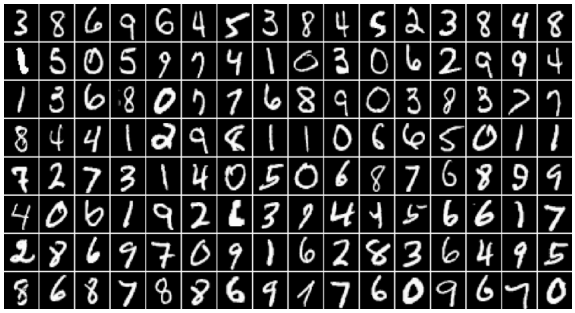

Figure 5: MNIST

3.2.2 Anime Face Dataset(AFD)

The Anime Face Dataset (AFD) is a collection of 63,632 images of anime characters' faces(64×64 resolution) that can be used for training machine learning algorithms for tasks such as facial recognition and expression recognition.

Figure 6: Anime Face Dataset

3.3 Evalution

In this section, we compare the model training time, sampling time and FID. The result in Table 1 shows that our model perform a better training and sampling time and the generative image quality is very close to DDPM[1].

Dataset	Method	training time(sec)	sampling time(sec)	FID
MNIST	DDPM	1924	1026	7.95
	ours(Low-resolution 14×14)	**400**	**134**	**6.59**
	ours(High-resolution 28×28)	**979**	**523**	
AFD	DDPM	25064	12558	29.23
	ours(Low-resolution 32×32)	**3195**	**939**	**27.94**
	ours(High-resolution 64×64)	**12546**	**6292**	

Table 1: Training comparison. Note that we show the detail of low and high-resolution model training and sampling time separately(For example: On MNIST dataset, DPPM training time is 1924, and ours method total time is $400 + 979 =$

Furthermore, we try different γ in Eq(1) to verify our method, in Table 2, the result show that different γ can effect the result intensely, so we can fine-tune the γ to get a best model.

γ	FID
0	28.97
0.3	**6.59**
0.5	8.77
1	11.35

Table 2: Different γ comparison on MNIST

Result

Figure 7: MNIST generative image by DDPM

Figure 8: MNIST generative image by HDD(ours)

Figure 9: AFD generative image by DDPM

Figure 10: AFD generative image by HDD(ours)

4 Conclusion

Using DDPM model to generate high resolution images requires a lot of time to training and sampling.We propose **a new efficient HDD model**, using low-resolution model to generate low resolution images, and depending on these images, using high-resolution model to generate high resolution images.The experiments show our model can **reduce** training and sampling time, and even get lower FID value.

Future work

First, Even though adjusting beta and gamma can effectively reduce the distance between the two domains, this method is simple and crude and lacks theoretical basis. Perhaps we can approach from a more theoretical angle and adjust the training algorithm to reduce the distance between the two domains.

Second, We have found that once the images generated by the low-resolution model are not very good, the images generated by the high-resolution model based on these images will not be very good either. Perhaps we can add a discriminator to judge the images generated by the low-resolution model, and only feed the images with scores above a certain threshold to the high-resolution model. This can effectively reduce the time spent generating poor images.

Third, In order to generate higher resolution images, perhaps we can add more models rather than just two, such as low-resolution, medium-resolution and high-resolution. Or even more, four, five, six and so on.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020.

[2] Tero Karras et al. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. 2017.