CS283: Assignment 5

Kerven Durdymyradov, KAUST ID: 187618

March 18, 2024

1 Diffusion Models: Questions (30 points)

1. (5 pt)

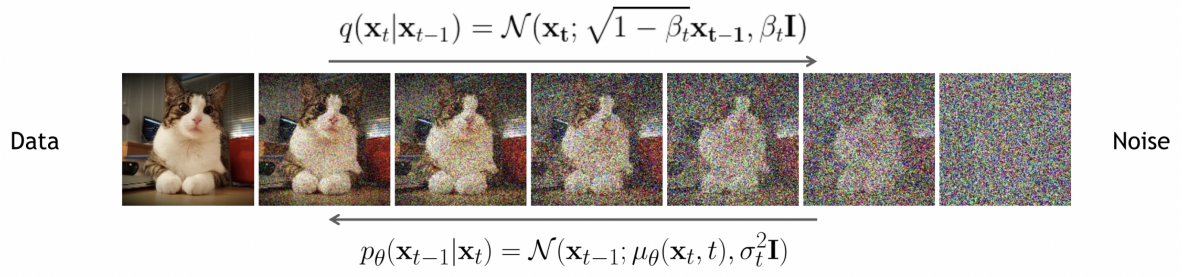

Fig. 1: Taken from tutorial slides of the paper

DDPMs follow the same forward process as traditional DMs, which involves simulating the evolution of a set of particles that represent pixel values of an image over time through simple linear transformations, with the addition of noise to encourage exploration of the image space. In the backward process, the goal is to reconstruct a noise-free image from a noisy input image. To achieve this, particles are initialized with pixel values from the noisy image and the diffusion process is simulated in reverse to remove noise from the image by gradually subtracting noise from the particles. The negative log-likelihood of the model with respect to the pixel values of the image is used to compute gradients during the backward process, which are used to update the parameters of the diffusion process through backpropagation. This helps to learn the parameters of the model that best capture the underlying probability distribution of the noise-free images.

2. (5 pt)

DMs and Hierarchical Variational Autoencoder (HVAEs) differ in their generative process. DMs use a simple linear transformation to simulate the evolution of particles (representing pixel values) over time, while HVAEs use a hierarchical neural network structure. HVAEs first encode the input through encoder networks to obtain latent variables, which are then decoded through decoder networks to generate the output. DMs have an implicit latent space that consists of the evolving particles, while HVAEs explicitly model the latent space through an encoder and decoder networks. Also, the parameters of DMs are learned through maximum likelihood estimation, while HVAEs use a combination of maximum likelihood estimation and variational inference to learn their parameters.

3. (5 pt)

DMs are good for generating high-quality images, but they can be computationally expensive to train and infer due to their direct operation in pixel space. To solve this issue, we can use pretrained autoencoders to apply DMs in the latent space, allowing for more efficient training and inference while still we can achieve good quality and flexibility. The use of cross-attention layers in the model architecture enables DMs to handle general conditioning inputs and produce high-resolution images

in a more efficient manner. This approach involves two training phases, where an autoencoder is first trained to create a lower-dimensional representation of the data space that is perceptually equivalent to the original data. The diffusion models are then trained in the learned latent space, allowing for better scaling properties and reduced complexity in image generation with a single network pass.

4. (5 pt)

In the Classifier approach of image generation using diffusion models, a pre-trained classifier is used to guide the diffusion process and generate images that belong to a desired class. This approach is effective if we know the desired class before and a pre-trained classifier is available. And the drawback of this approach is that it requires a pre-trained classifier to guide the diffusion process. In the , Classifier-Free approach learns to perform class-conditioned image synthesis without relying on any pre-trained classifier or class label information. This approach is more flexible as it can handle any conditioning input and can generate more diverse images. Drawback for it is that it may not be as effective as the Classifier approach in generating images that are faithful to a specific target class. Therefore, while the Classifier approach has the advantage of generating images that belong to a specific target class, it requires a pre-trained classifier. The Classifier-Free approach is more flexible as it does not require a pre-trained classifier, but it may not be as effective in generating images that are faithful to a specific target class.

5. (5 pt)

The main idea is to explore the capabilities of diffusion models in image synthesis. The authors aim to improve the quality of image generation by introducing data augmentation into the conditioning information of super-resolution models. The main challenge they face when connecting the different stages together is the compounding error in cascading pipelines due to train-test mismatch in super-resolution models. To address this challenge, the authors propose practical methods to train and test models amortized over varying levels of conditioning augmentation.

6. (5 pt)

The SDE and SBGMs are connected with the use of the score function. The score function of SBGMs guides the optimization of a particle to obtain a sample from the model distribution and is the gradient of the log-density function of the data distribution. The score function of SDE is the drift term of a stochastic differential equation that provides a continuous-time evolution of the particle towards a sample from the model distribution. SDE achieves better sample quality and more efficient sampling than SBGMs. This is because the continuous-time evolution of the particle in the SDE allows for a more efficient exploration of high-density regions of the distribution, while also providing a way to balance exploration and exploitation. Also, the SDE can handle complex high-dimensional distributions with multimodal or non-smooth structure, which can be difficult for SBGMs because of the high variance of the score function estimator.

4 [Optional/Bonus] Generative Zero-Shot Learning Questions (30 points)

1. (7.5 pt)

Hallucinated features are visual features that are generated for unseen classes using a GAN trained on noisy textual descriptions of the classes. They will enable zero-shot learning, where instances of unseen classes can be classified even though they were not present in the training data. The generated features represent the visual characteristics of the unseen classes and are used as input to a classifier trained on noisy textual descriptions and real visual features of known classes. The GAN is trained to generate diverse and representative features for each class by minimizing the Wasserstein distance between the distribution of the real features and the distribution of the hallucinated features. The generator network takes random noise and class embedding as input to produce features that represent the visual characteristics of the class, while the discriminator network distinguishes between real and

hallucinated features. The goal is to capture the visual characteristics of the class from the textual descriptions.

2. (7.5 pt)

I run this command:
python train CIZSL.py –dataset 'CUB' –splitmode 'easy' –creativity weight 0.0001

```
python train_CIZSL.py --dataset 'CUB' --splitmode 'easy' --creativity_weight 0.0001
```

After running the code, I was able to achieve an Accuracy of 41.19% and Generalized AUC of 38.63%.

```
100%|████████████████████████████████████████| 3001/3001 [56:26<00:00,  1.13s/it]
===============
===============
Reproduce CUB easy
Accuracy is 41.19%, and Generalized AUC is 38.63%
```