TASK DESCRIPTION:

Diffusion models are a powerful class of generative models that can learn complex high-dimensional distributions. But they are limited by the time taken to generate samples. As mentioned in the CVPR tutorial " Tackle the trilemma by accelerating diffusion models ".
The trilemma they are talking about is the balance between

1. Fast sampling
2. Diversity of sample (features)
3. High quality of samples generated

We are exploring techniques to accelerate the diffusion models. Specifically we dive into literature and implementations of

1. Variational diffusion models
2. Denoising diffusion implicit models
3. Denoising diffusion GANs
4. Latent-space diffusion models

CHALLENGES ADDRESSED:

OUTLINE OF METHODS:

Here we describe the literature survey and details about the model architectures .

1.Variational diffusion models

The forward process of diffusion model give by

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t\geq 1} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

The variance in above equation is parametrised using a learned monotonic(non-negative weights) multilayered perceptron denoted by

$$1- \alpha_t = \text{sigmoid}(\gamma_\eta(t))$$

Also the encoder is learnt in VDMs unlike diffusion models where it is fixed. This allows the model to capture more complex distributions and generate more diverse images.Variational diffusion models (VDMs) are faster than traditional diffusion models because they use a learned function to parametrize the variance, rather than sampling from a fixed noise distribution at each step of the diffusion process.

Training objective :

The training objective of VDM is to maximize the variational upper bound with respect to the encoder parameters.It is shown that the variational upper bound in continuous time (T -> ∞) is only related to the signal-to-noise ratio at the endpoints, and is invariant to the noise schedule in-between.

Signal-to-noise ratio(t) = exp(-γ_n(t)) at endpoints

Variational diffusion models are a type of machine learning model used for likelihood estimation, which is a way of measuring how well a model fits a given set of data. The state-of-the-art method for improving the performance of these models is by appending Fourier features to the input of a U-Net, which is a type of neural network.

The problem with likelihood estimation is that good likelihoods require the model to be able to accurately predict even very small changes in the input data. However, neural networks are usually not very good at modeling such small changes.

By appending Fourier features to the input of a U-Net, the model is better able to capture the small changes in the input data, resulting in significant improvements in the log-likelihoods

2. Denoising diffusion implicit models

$$L_\gamma(\epsilon_\theta) \quad := \sum_{t=1}^{T} \gamma_t \mathbb{E}_{x_0 \sim q(x_0), \epsilon_t \sim \mathcal{N}(0,I)} \left[\left\| \epsilon_\theta^{(t)}(\sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t} \epsilon_t) - \epsilon_t \right\|_2^2 \right]$$

Our key observation is that the DDPM objective in the form of Lγ only depends on the "marginals" q(xt|x0), but not directly on the "joint" q(x1:T|x0). Since there are many inference distributions ("joints") with the same "marginals", we explore alternative inference processes that are non-Markovian, which leads to new generative processes (non-Markovian processes)which lead to the same surrogate objective function as DDPMs.

3. Denoising diffusion GANs

Denoising diffusion models are a type of generative model that generate data by gradually perturbing the input data with noise. To generate data, a parametrized reverse process denoises the noisy input data in thousands of iterative steps, starting from random noise. Typically, denoising distributions in reverse processes are modelled using Gaussian distributions. However, this assumption holds only for small denoising steps, requiring a large number of steps in the reverse process. With larger step sizes(i.e., it has fewer denoising steps), a non-Gaussian multimodal distribution is needed to model the denoising distribution, which arises from multiple plausible clean images corresponding to the same noisy image.

To address this issue, denoising diffusion generative adversarial networks (DDGANs) use a multimodal conditional GAN to model each denoising step. DDGANs are designed specifically for fast sampling while maintaining strong mode coverage and sample quality.

Our adversarial training setup is shown below. Given a training image x_0, we use the forward Gaussian diffusion process to sample from x_{t-1} and x_t, the diffused samples at two successive steps.

The forward diffusion process is set up similarly to the diffusion models, with the assumption that T is small (T ≤ 8), and each diffusion step has larger βt.

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t\geq 1} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

Given x_t, our conditional denoising GAN first stochastically generates x'_o and then uses the tractable posterior distribution $q(x_{t-1}|x_t, x_o)$ to generate x'_{t-1}. The discriminator is trained to distinguish between the real (x_{t-1}, x_t) vs. fake (x'_{t-1}, x_t) pairs. We train both generator and discriminator using the non-saturated GAN objective simultaneously for all time steps t. Generators and discriminators for different steps t share parameters, while additionally conditioning on t-embeddings, similar to regular denoising diffusion models

Discriminator trained on:

$$\min_{\phi} \sum_{t\geq 1} \mathbb{E}_{q(\mathbf{x}_t)} \left[\mathbb{E}_{q(\mathbf{x}_{t-1}|\mathbf{x}_t)}[-\log(D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, t)] + \mathbb{E}_{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}[-\log(1-D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, t))]\right]$$

Generator trained on :

$$\max_\theta \sum_{t\geq 1} \mathbb{E}_{q(\mathbf{x}_t)} \mathbb{E}_{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}[\log(D_\phi(\mathbf{x}_{t-1}, \mathbf{x}_t, t))],$$

In contrast to DDPM, where x0 is predicted as a deterministic mapping of xt, in DDGANs, x0 is produced by the generator with a random latent variable z. This key difference allows the denoising distribution pθ(xt-1|xt) to become multimodal and complex in contrast to the unimodal denoising model in DDPM.

After training, we generate novel instances by sampling from noise and iteratively denoising it in a few steps(greatly reducing sampling time) using our denoising diffusion GAN generator.

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \int p_\theta(\mathbf{x}_0|\mathbf{x}_t) q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) d\mathbf{x}_0 = \int p(\mathbf{z}) q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0 = G_\theta(\mathbf{x}_t, \mathbf{z}, t)) d\mathbf{z}$$

4.Latent Space Diffusion Model

The Latent Space Diffusion Model (LSDM) is a generative model that operates in the latent space of a pre-trained deep neural network. The model is trained to transform a random noise vector into a sample that resembles the distribution of the training data.

Starting with a noise vector $z_0 \sim p(z)$, apply T diffusion steps to obtain a final sample x_T:

$z_{t+1} = z_t + \text{sqrt}(2 * \eta) * \varepsilon$

$x_t = g(z_t)$ for $t = 0,1,2,...T-1$

LSDM uses a diffusion process to generate samples. In this process, a set of diffusion steps is applied to the noise vector to gradually transform it into a sample. Each diffusion step involves updating the noise vector by adding a scaled Gaussian noise vector, and then passing the result through the decoder network to obtain a sample.

The model is trained to minimize a contrastive loss function, which encourages the encoder network to learn a representation that captures the underlying structure of the data.

By repeating this process for a sufficient number of steps, the noise vector is transformed into a realistic sample from the target distribution.

The advantage of using a diffusion process is that it allows LSDM to generate high-quality samples with fewer steps compared to other generative models like Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs). This is because the diffusion process does not require the model to optimize a complex objective function or to compute expensive gradient updates. Instead, the model can simply apply a sequence of simple transformations to the input noise vector to generate a sample.

Another advantage of LSDM is that it can be trained using a contrastive loss function, which is computationally efficient and easy to optimize. This makes it easier to train large-scale models on large datasets, which can be challenging with other generative models.

The model is trained to minimize a contrastive loss function, which encourages the encoder network to learn a representation that captures the underlying structure of the data.The loss function is defined as follows:

$L = E_x[||z - e(x)||\text{\^{}}2] + \alpha * E_z[||g(z) - x||^2]$

Here, e(x) is the output of the encoder network, which maps the data samples x to the latent space z, and alpha is a hyperparameter that controls the balance between the reconstruction loss and the contrastive loss.

Overall, the combination of the diffusion process and the contrastive loss function make LSDM a promising approach for generative modeling, with potential applications in image synthesis, video prediction, and other areas of machine learning.

In LSDM, The diffusion process is designed to ensure that the generated samples are diverse, by gradually increasing the complexity of the noise vector as it progresses through the diffusion process.LSDMs can be trained using a contrastive loss function, therefore they are based on a meaningful representation of the data. This results in samples that are different from one another, but still high-quality. LSDMs are computationally efficient, which allows them to generate samples quickly without sacrificing quality or diversity. This makes LSDMs a promising approach to tackle the trilemma of fast sampling, diversity, and quality in generative modeling.

Experiment Details and Main Results

Related Work

Conclusion

2.Denoising diffusion implicit models

- GANS > VANS,autoregressive models and normalizing flows (in terms of quality)
- DDPMs>GANS(similar quality but GANS donot need adversarial training)
- DDIMs>DDPMs(lesser iterations,basically faster).Because we are able to use *non-Markovian* diffusion processes which lead to ”short” generative Markov chains.

Work split up amongst the team members

References to existing code or libraries used for the project.

https://openreview.net/pdf?id=2LdBqxc1Yv

🗎 slides_v2.pdf