An Implementation of Denoising Diffusion Probabilistic Models

Yug Ajmera
University of Pennsylvania

***Abstract*—In this project, we will implement a Denoising Diffusion Probabilistic Model(DDPM) for image generation. DDPMs are a relatively new class of generative neural networks which have recently shown to produce better results compared to Generative Adversarial Networks(GANs). As the name suggests, DDPMs try to emulate the process of diffusion wherein, normally distributed noise ϵ is progressively added to the input $\mathbf{x}$ to produce a sequence of noised representation of the input $\tilde{\mathbf{x}}$. We then try to learn the reverse mapping from the noised representations $\tilde{\mathbf{x}} \rightarrow \mathbf{x}$ using Deep Neural Networks(DNNs).**

I. Goals and Objectives

The currently conventional models for generative deep learning on images are Generative Adversarial Networks(GANs). In practice, however, GANs are notoriously difficult to train [6]. If the balance between the Generator and discriminator is not maintained, the training loss can diverge or become stagnant. DDPMs are an alternative class of generative models which use a different mechanism of learning. In recent studies DDPMs have shown superior results when compared to GANs for generative deep learning. DDPMs propose a diffusion based process for image generation whereby small incremental noise is added to the true samples $\mathbf{x}$ for T steps as part of the forward diffusion process. We then try to reverse this process using a Deep Neural Network(DNN) which is typically a Variational Auto Encoder(VAE). Informally, the benefit of carring out this incremental process is that the reverse mapping can be better defined mathematically. In general, it is important to note that DDPMs make the assumption that the noised sample obtained after T timesteps of noising is approximately sampled from a multivariate Gaussian distribution. This assumption allows a better mathematical analysis of the reverse mapping (from the noised representation to the true samples) which we are ultimately interested in.

Diffusion models can complete various tasks, including image generation, image denoising, inpainting, outpainting, and bit diffusion. Popular diffusion models include Open AI's Dall-E 2, Google's Imagen, and Stability AI's Stable Diffusion. These models cannot be trained/used without a big GPU. The goal of this project is to implement a simpler denoising diffusion probabilistic model that can be trained on google colab and can potentially generate new images that are visually appealing.

II. Related Works

Diffusion models typically use Markov chains to gradually convert one distribution into another, an idea which has it's roots in non-equilibrium statistical physics [3]. In the context of generative modelling, we formulate a Markov chain that transforms a known distribution into a target distribution. Typically, the known distribution in our case is a Multivariate Gaussian and the unknown distribution is that of the input data. The idea behind using a Markov chain for the formulation is that every step has an analytically evaluable probability and hence the full chain is also analytically evaluable (under the assumption that the final probability in the ultimate timestep of the forward process can be analytically evaluated). The main advantage that this has over GANs is that it is in general easier to evaluate small perturbations in the diffusion process, than it is to describe a full distribution with a single non-analytically-normalizable potential function [9].

The simple idea of Diffusion Models is then to incrementally destroy information in a distribution of data in forward diffusion and train a DNN to reverse this process and recover information. [2] showed that instead of predicting the data distribution in the chain, if an estimate of the added Gaussian noise is predicted, the produced samples have better quality. This makes sense since the noise is explicitly Gaussian, whereas the data distribution is in general non-analytic [7]. [1] improved the performance of the model proposed by [2] by improving on the baseline model (UNet [8] in [2]'s work); they added a global attention layer and added a temporal embedding to the model, so as to include the specific timestep in the training process. The temporal embedding allows the model to estimate the amount of noise removal more accurate since, different timesteps in general have a different amount of noise (this is made more significant due to non-linear noising schedules introduced by [1]).

III. Proposed Approach

In this project, we intend to expand upon the original DDPM implementation by [2], and train it with limited compute resources. Note that there are several perspectives on diffusion models. Here, we employ the discrete-time (latent variable model) perspective as it is simpler to train.

A fixed (or predefined) forward diffusion process q of our choosing, that gradually adds Gaussian noise to an image, until you end up with pure noise. A learned reverse denoising diffusion process $p\theta$, where a neural network is trained to gradually denoise an image starting from pure noise, until you end up with an actual image. Steps involved in the diffusion process:

1) we take a random sample $\mathbf{x}_0$ from the real unknown and possibily complex data distribution $q(\mathbf{x}_0)$
2) we sample a noise level t uniformally between 1 and T (i.e., a random time step)
3) we sample some noise from a Gaussian distribution and corrupt the input by this noise at level t
4) the neural network is trained to predict this noise based on the corrupted image $\mathbf{x}_t$, i.e. noise applied on $\mathbf{x}_0$ based on known schedule β_t

To derive an objective function to learn the mean of the backward process, the authors observe that the combination of q and $p\theta$ can be seen as a variational auto-encoder (VAE). Hence, the variational lower bound (also called ELBO) can be used to minimize the negative log-likelihood with respect to ground truth data sample $\mathbf{x}_0$. The neural network is optimized using a simple mean squared error (MSE) between the true and the predicted Gaussian noise.

The neural network needs to take in a noised image at a particular time step and return the predicted noise. What is typically used here is very similar to that of an Autoencoder, where an encoder first encodes an image into a smaller hidden representation called the "bottleneck", and the decoder then decodes that hidden representation back into an actual image. In terms of architecture, the DDPM authors went for a U-Net, which like any autoencoder, consists of a bottleneck in the middle that makes sure the network learns only the most important information. Importantly, it introduced residual connections between the encoder and decoder, greatly improving gradient flow.

We will test the developed model incrementally, hence we will first start from simpler datasets and progress towards more complex datasets. First we intend to implement a DDPM on the simple MNIST Fashion dataset [10], we will then progress to the relatively more difficult CIFAR-10 [5] dataset. Finally, we will use the Stanford Cars dataset ([4]) that contains 8k images of cars. Since ours is a purely generative task, we do not require semantic labels associated with the images. We will train our DDPM to produce realistic samples from this dataset, we will further test different backbones and compare the performances.

IV. Timeline

- Week 1 (6th Nov - 13th Nov): Understanding the nuances of the paper
- Week 2 (13th Nov - 20th Nov): Implementing the original DDPM architecture
- Week 3 (20th Nov - 27th Nov): Training the network on MNIST Dataset and CIFAR-10 Dataset
- Week 6 (27th Nov - 3rd Dec): Trying different backbones
- Week 7 (4th Dec - 10th Dec): Report writing and finalizing submission

References

[1] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[3] C. Jarzynski. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E*, 56(5):5018–5035, nov 1997.

[4] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[5] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.

[6] Lars M. Mescheder. On the convergence properties of GAN training. *CoRR*, abs/1801.04406, 2018.

[7] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *CoRR*, abs/2102.09672, 2021.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[9] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585, 2015.

[10] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017.