Generating Images with a Score-Based Diffusion Model

Abstract

This paper proposes using a diffusion model to generate images by investigating and building on the concept of score from recent generative models. The main aim of this paper is to generate images from noise using stochastic differential equations to model potential pathways between realistic images and noisy data in our data space. This is achieved through a combination of Langevin dynamics for a forward time diffusion to noise, and deep neural networks to represent a quantity we define as score. This work introduces novel methods of retracing steps back to the data distribution using the Millstein method to sample in reverse time, whilst also improving the performance of score networks through features such as the exponential moving average.

1 Methodology

A common problem with likelihood-based generative models is the calculation of the normalisation coefficient. If architectures are left unrestricted, this could lead to intractable normalisation coefficients. For example, energy based models that rely on some probability density function,

$$p(\mathbf{x}) = \frac{e^{-E(\mathbf{x})}}{\int_{\hat{\mathbf{x}} \sim \mathcal{X}} e^{-E(\mathbf{x})}}, \tag{1}$$

where $E(\mathbf{x})$ is an energy function [3], and the integral in the denominator could be intractable. While implicit generative models such as Generative Adverserial Networks (GANs) had been known to produce much better results through rather hackish, unnatural energy functions, recent advancements such as Denoising Diffusion Probabilistic Models (DDPMs) and its derivatives such as Score-Based Models (SBMs) have began to rival GANs. These models work on the concept of score, s, which is related to the gradient of the distribution in data space, rather than the parameter space that is common in most deep learning works. It is defined as:

$$s = \nabla_{\mathbf{x}} \log p(\mathbf{x}). \tag{2}$$

This is implemented as a deep neural network $s_\theta(\mathbf{x})$, with a U-Net architecture by Song et al. [4]. Since $\mathbf{x}, s(\mathbf{x}) \in \mathbb{R}^n$, the input and output to of $s_\theta(\mathbf{x})$ is of the same shape. SBMs allow us to create generative models that seek to minimise the Fisher Divergence as their objective function (cite this), through a process known as score-matching.

$$FD(\mathbf{x}) = \mathbb{E}_{p(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - s_\theta(\mathbf{x})\|_2^2] \tag{3}$$

Several architectural changes were made to s_θ from Song's original code (which was for the MNIST dataset). Initially the convolution configurations were changed to try out different architectures such as 4,2,1 convolutions and 3,1,1 convolutions. 3,1,1 were found to work best and implemented in the network. Next, the dimensionality of the latent space was experimented with heavily. After trying both 512 and 256 sizes in the latent space, 256 was found to work best, as depicted in the loss results section.

Furthermore, various alternative activation functions were tried to replace $\tanh(x)$ for this diffusion model, such as $\frac{1}{2}(1 +$

$textrmerf(x)$), however $\tanh x$ showed the best results. Song's architecture was also enhanced through the introduction of an Exponential Moving Average procedure for stochastic trajectory optimization during training [2].

Next, the mathematics of Langevin Dynamics is used to sample from our model. This relies on two stochastic differential equations (SDEs). The forward equation that allows our image to drift and diffuse through data space through a time step schedule $\{\mathbf{x}(t) \in \mathbb{R}^n\}_{t=0}^{T}$, perturbing our distribution p_0 to an eventual prior distribution of noise p_T,

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}\mathbf{x} + g(t)\mathrm{d}\mathbf{w}, \tag{4}$$

where $\mathbf{f}(\mathbf{x}, t)$ is the drift coefficient, $\mathbf{w}$ from the Weiner process (cite this), $g(t)$is responsible for diffusion, and $\mathrm{d}t$ is a step forward in time. In this paper, we tried various mappings from our exploration standard deviation, σ, and found $g(t) = \sigma^t$ to perform consistently well as adrift coefficient. It also lends itself nicely to sampling methods developed in this work, discussed later. A reverse time SDE allows us to retrieve a realistic image back from the noise, for a well-trained model. This reverse SDE has closed form [4]:

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log(p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \tag{5}$$

with $\mathrm{d}t$ now representing a step backward in time. While Song et al. presents 3 ways to sample this SBM (the Euler-Mayurama (EM), Predictor-Corrector(PC) and the ODE solver(ODE)), in this work we show, test and experiment a new method untested in the literature for this kind of model.

SDEs like (4) can be solved numerically by taking the Taylor-Ito Series expansion of $\mathbf{x}$ [1]:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + f(\mathbf{x}_t, t)\Delta t + g(\mathbf{x}_t)\Delta\mathbf{w}_t + \frac{1}{2}g(\mathbf{x}_t)\frac{\mathrm{d}g}{\mathrm{d}\mathbf{x}}[\Delta\mathbf{w}_t^2 - \Delta t] + R, \tag{6}$$

where R is the remainder terms. The Euler-Maruyama method does exactly this to obtain a numerical solution for our reverse SDE, but only up to the first term of the series expansion (6). In this paper, we extend Song et al.'s work and implement a Millstein SDE solver (cite this) to sample images using the first two terms of the Taylor-Ito expansion. The Millstein method assumes $f, \frac{\mathrm{d}f}{\mathrm{d}t}, g, \frac{\mathrm{d}g}{\mathrm{d}t}$ all satisfy a uniform Lipschitz condition (cite this), and f, g are twice continuously differentiable. It is worth nothing that the diffusion step $\Delta\mathbf{w}$ in (6) is implemented as $\Delta\mathbf{w} = \sqrt{\Delta t}\mathbf{z}_t$ in this work, where $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_n)$.

Finally, all 4 samplers were experimented with to rank them in terms of their diversity of samples generated. These results are elaborate in the following section.

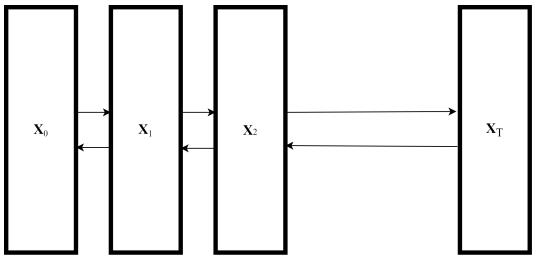

Figure 1: Drift of the image with subsequent timesteps, to reach a prior noise distribution

2 Results

The following are sets of 64 randomly picked samples, generated after different amounts of sampling using our Millstein Sampler.

The quality of samples for such methods are proportional to not just the training time, but the amount of sampling used.

Figure 2: Random sample of 64 images

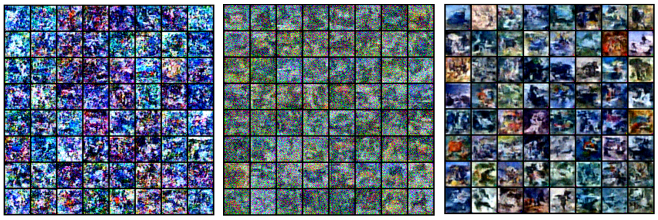

Figure 3: Interpolating images form prioir distribution in latent space

Similarly, the quality of samples is also strongly dependant on the type of sampler used. The 3 samplers by [4] were experimented with at various parameters, such as step count, signal to noise ratio and exploration σ to generate sampled of 8 images and study how these influence the models sampled from the same ScoreNet model (i.e no further training was performed between each sampling trial).

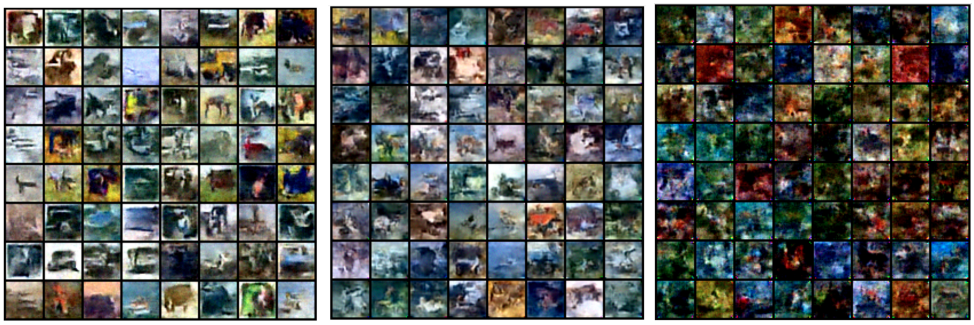

Figure 4: (left to right) PC sampler, EM sampler and ODE sampler all sampling images from the same pytorch model

The following loss curves were obtained when training the models over 10000 epochs, using different dimensional spaces:

Some Cherry Picked samples:

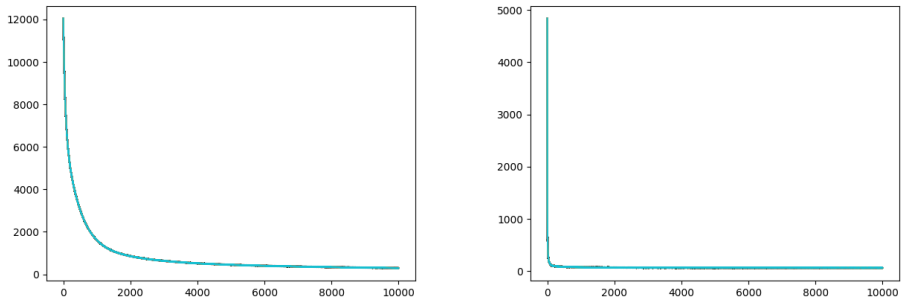

Figure 5: (left to right) Loss in original latent space and 256 dimensional latent space after modifiying convolutions

Figure 6: (left to right) Loss in original latent space and 256 dimensional latent space after modifiying convolutions

It was also noted that, while experimenting with the PC Sampler, taking a very large nuber of steps led ot the opposite effect of numerical samplers like EM or Millstein: the images looked warmer!

Figure 7: (left to right) Loss in original latent space and 256 dimensional latent space after modifiying convolutions

3 Limitations

the images are vaguely distinguishable but still very blurry.

Fig 3 Suggests numerical sampling methods often end up much more blue and cooler than other images in the data set, and the PC sampler shows the most variety. Due to GPU

constraints on Colab the augmented PC sampler was not able to run long enough to generate realsitic images. There is also scope for further improvements in the PC sampler method.

Bonuses

This submission has a total bonus of -2 marks (a penalty), as it is trained only on CIFAR-10.

References

[1] Mustafa Bayram, Tugcem Partal, and Gulsen Orucova Buyukoz. "Numerical methods for simulation of stochastic differential equations". In: *Advances in Difference Equations* 2018.1 (2018), pp. 1–10.

[2] Yanhao Jin, Tesi Xiao, and Krishnakumar Balasubramanian. "Statistical Inference for Polyak-Ruppert Averaged Zeroth-order Stochastic Gradient Algorithm". In: *arXiv preprint arXiv:2102.05198* (2021).

[3] Taesup Kim and Yoshua Bengio. "Deep directed generative models with energy-based probability estimation". In: *arXiv preprint arXiv:1606.03439* (2016).

[4] Yang Song et al. "Score-based generative modeling through stochastic differential equations". In: *arXiv preprint arXiv:2011.13456* (2020).