Generating Images with a Denoising Diffusion Probabilistic Model with input perturbation

Anonymous author

Abstract

This paper proposes using a Denoising Diffusion Probabilistic Model (DDPM) with input perturbation for image generation. DDPMs have shown impressive generation quality, however their long sampling chain leads to an error accumulation issue, similar to the exposure bias problem in auto-regressive language models. Input perturbation is suggested as a solution to the exposure bias problem as it perturbs ground truth samples to simulate inference time prediction errors, resulting in a significant improvement in sample quality.

1 Methodology

The method trains a DDPM to generate images by using a sequence of denoising steps [4]. Images are progressively destroyed using Gaussian noise from the data distribution $q(\boldsymbol{x_0})$. In the forward process (1) a sample is destroyed over T steps $\boldsymbol{x}_0, ..., \boldsymbol{x}_t, ..., \mathbf{x}_T$, using a prefixed noise schedule $\beta_1, ..., \beta_T$,

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}), \tag{1}$$

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \tag{2}$$

until obtaining a completely noisy image. In back propagation, samples are used to invert this forward process using a deep denoising autencoder $\mu(\cdot)$. This is defined by transition probabilities parameterized by θ.

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \sigma_t), \tag{3}$$

Where $\sigma_t = \frac{1-\bar{\sigma}_{t-1}}{1-\bar{\sigma}_t}\beta_t$ with $\bar{\sigma}_t = \prod_t^{i=1} \alpha_i$ and $\alpha_i = 1 - \beta_i$ given $\boldsymbol{x}_0, \boldsymbol{x}_t$ can be obtained by

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{4}$$

The network $\boldsymbol{\epsilon_\theta}(\cdot)$ aims to predict the noise vector $\boldsymbol{\epsilon}$,using a simple L_2 loss function, the training objective becomes

$$\mathbb{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), t \sim \mathbb{U}(\{1,...,T\})} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|^2 \right] \tag{5}$$

Training DDPMs takes lots of steps, T needs to be large as each denoising step is assumed to be Gaussian noise for small step sizes and when lots of noise is added sequentially the prediction network has to solve a harder problem. Errors accumulate in the sampling chain for large values of T as the samples are generated on the results of the previous denoising steps, this problem becomes an issue if the model outputs a bad sample at a certain step, which in turn effects the whole sampling sequence ahead. This problem is similar to the exposure bias problem in auto-regressive language models which refers to the train-test discrepancy that seemingly arises when an autoregressive generative model uses only ground-truth contexts at training time but generated ones at test time [3].

The proposed solution, Denoising Diffusion Probabilistic Model with input perturbation (DDPM-IP), builds upon a Denoising Diffusion Probabilistic Model [1] and aims to alleviate

the exposure bias problem by explicitly modelling the prediction error during training. At training time $\boldsymbol{x}_t$ is perturbed and $\mu(\cdot)$ is fed with a noisy version of $\boldsymbol{x}_t$ so as to simulate the inference time prediction and force the autoencoder to learn to take the error into account. The proposed method is a training regularization method which smooths the prediction function and gets a purturbed version of $\boldsymbol{x}_t$ using a random noise vector $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ called $\boldsymbol{y}_t$.

$$\boldsymbol{y}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}(\boldsymbol{\epsilon} + \gamma_t\boldsymbol{\xi}) \tag{6}$$

The proposed DDPM-IP training cycle uses Eq.5 and Eq.6.

Algorithm 3 DDPM-IP: Training with input perturbation

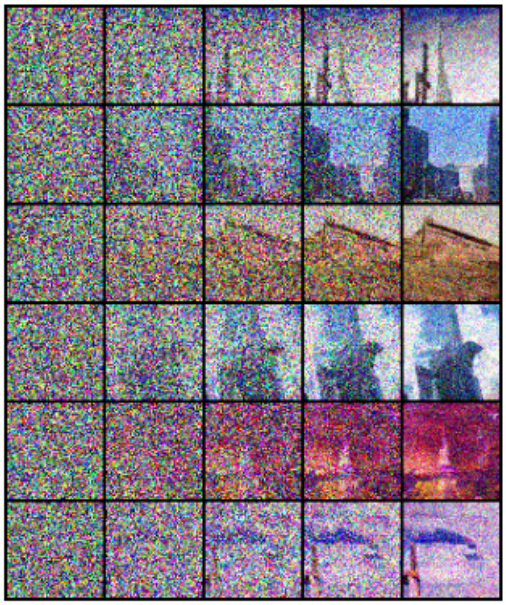
1: **repeat**
2: $\quad \boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), t \sim \mathrm{U}(\{1, ..., T\})$
3: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
4: $\quad$ compute $\boldsymbol{y}_t$ using Eq. 6
5: $\quad$ take a gradient descent step on $\nabla_{\boldsymbol{\theta}}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{y}_t, t)||^2$
6: **until** converged

2 Results

The images were trained on a part of the LSUN dataset, churches, rescaled to 64x64. The non cherry picked results were as follows using code based on [2].

These results are generally good as in most of the images some form of tower can be identified, even in some of the more unclear images a "tower like structure" can be seen.

The next results show the image denoising process.



And here are some cherry-picked samples that show the best outputs the model has generated:

I am really happy with the quality of the cherry-picked samples as the images clearly show a realistic skyline with many different types of towers, built in many different time periods, at different times of day.

3 Limitations

One limitation is that the results mimic the training data that it was trained on, for example the Eiffel tower appears multiple times because it is one of the most photographed towers. Another limitation is that some of the towers have increased lighting and unrealistic sky colours. Furthermore due to the low resolution of the images when zooming in the towers are less clear and it is hard to make out features. Unfortunately due to the amount of processing power for DDPM-IP higher resolution images could not be trained using the resources available.

Bonuses

This submission has a total bonus of +4 marks, as it is trained on LSUN resized to 64x64.

References

[1] Prafulla Dhariwal and Alex Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *ArXiv* abs/2105.05233 (2021).

[2] forever208. *DDPM-IP*. https://github.com/forever208/DDPM-IP. Accessed on: February 15, 2023. 2023.

[3] Florian Schmidt. "Generalization in Generation: A closer look at Exposure Bias". In: *Proceedings of the 3rd Workshop on Neural Generation and Translation*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 157–167. DOI: 10.18653/v1/D19-5616. URL: https://aclanthology.org/D19-5616.

[4] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *ArXiv* abs/2010.02502 (2020).