Progressive Distillation for Fast Sampling of Diffusion Models [1]

Minji Kim (minji@snu.ac.kr; 2020-28702), Dept. of Electrical and Computer Engineering, Seoul National University

1. Introduction

Despite the effectiveness of diffusion models in generative tasks, the limitation of diffusion models is their slow sampling time. This paper presents new parameterizations of diffusion models that provide increased stability while using few sample steps. Specifically, this method distills a trained deterministic sampler using many steps into a new diffusion model that takes half of the original sampling steps. The proposed distillation procedure does not take more time than it takes to train the original model, and it can generate an image as few as 4 steps, while still maintaining sample quality competitive with state-of-the-art models which requires thousands of diffusion iteration steps.

2. Progressive Distillation

2.1. Overview

The overview of *Progressive Distillation* is shown in Fig. 1. The proposed method iteratively halves the number of required sampling steps by distilling a slow teacher diffusion model into a faster student model. The iteration process is as follows:

1. Each time the incremental distillation is repeated, the student model is initialized with the copy of the teacher model using the same model parameters and the same model definition.

2. Sample data from the training set and add noise to the data.

3. Process the target $\tilde{x}$ by running two DDIM [2] sampling steps based on the teacher model.

4. Reduce the student step to the half, and set the student model as the new teacher.

The overall algorithm is shown in Fig. 2.

3. Conclusion

This paper has proposed Progressive Distillation, a method that dramatically reduces the number of sampling steps required to produce high quality images, and potentially other data using diffusion models with deterministic samplers such as DDIM [2]. To do so, the authors reparameterize the prediction and define stable weight. By making these models more affordable to run at test time, this work opens a potential that the diffusion model could be applied in environments with time and memory constraints.

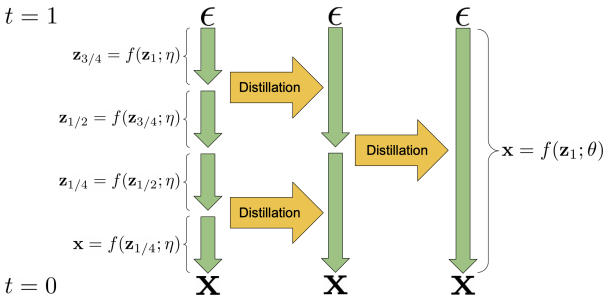

Figure 1. Overview of Progressive Distillation.

Algorithm 1 Standard diffusion training

Require: Model $\hat{\mathbf{x}}_\theta(\mathbf{z}_t)$ to be trained
Require: Data set $\mathcal{D}$
Require: Loss weight function $w()$

while not converged **do**
  $\mathbf{x} \sim \mathcal{D}$ ▷ Sample data
  $t \sim U[0, 1]$ ▷ Sample time
  $\epsilon \sim N(0, I)$ ▷ Sample noise
  $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ ▷ Add noise to data
  $\tilde{\mathbf{x}} = \mathbf{x}$ ▷ Clean data is target for $\hat{\mathbf{x}}$
  $\lambda_t = \log[\alpha_t^2 / \sigma_t^2]$ ▷ log-SNR
  $L_\theta = w(\lambda_t) \|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$ ▷ Loss
  $\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$ ▷ Optimization
end while

Algorithm 2 Progressive distillation

Require: Trained teacher model $\hat{\mathbf{x}}_\eta(\mathbf{z}_t)$
Require: Data set $\mathcal{D}$
Require: Loss weight function $w()$
Require: Student sampling steps N
for K iterations **do**
  $\theta \leftarrow \eta$ ▷ Init student from teacher
  while not converged **do**
    $\mathbf{x} \sim \mathcal{D}$
    $t = i/N, \; i \sim Cat[1, 2, \ldots, N]$
    $\epsilon \sim N(0, I)$
    $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
    `# 2 steps of DDIM with teacher`
    $t' = t - 0.5/N, \;\; t'' = t - 1/N$
    $\mathbf{z}_{t'} = \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_t) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\eta(\mathbf{z}_t))$
    $\mathbf{z}_{t''} = \alpha_{t''} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}_{t'} - \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}))$
    $\tilde{\mathbf{x}} = \frac{\mathbf{z}_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
    $\lambda_t = \log[\alpha_t^2 / \sigma_t^2]$
    $L_\theta = w(\lambda_t) \|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$
    $\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$
  end while
  $\eta \leftarrow \theta$ ▷ Student becomes next teacher
  $N \leftarrow N/2$ ▷ Halve number of sampling steps
end for

Figure 2. Algorithm of standard diffusion training and Progressive Distillation.

References

[1] T. Salimans et al. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022. 1

[2] J. Song, et al. Denoising diffusion implicit models. In *ICLR*, 2021. 1