Project 14: Implementation of Denoising Diffusion Probabilistic Models

Caroline Wrist-Jensen (s194349), Christian Schaumburg Jakobsen (s194307),
Niklas Kristian Jensen (s194340) and Vitus Bødker Thomsen (s194331)

DTU Compute, Technical University of Denmark

Introduction

Diffusion models are currently among the most popular methods for generating images. They especially gained popularity after the paper Denoising Diffusion Probabilistic Models by Ho et al. (2020). In this project, we seek to reproduce the results from the DDPM paper and generate synthetic data based on the MNIST and CIFAR-10 datasets. Furthermore, we implement Classifier-Free Guidance which allows us to condition on specific classes when sampling.

Background

Diffusion models use latent variables $\mathbf{x}_1, \dots, \mathbf{x}_T$ which are increasingly noisier versions of the original image $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. The forward process $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ is a Markov chain that gradually adds Gaussian noise to the data with variance defined by $\beta_1, \dots, \beta_T$. The reverse process $p_\theta(\mathbf{x}_{0:T})$ is a Markov chain that learn to denoise the image according to learned Gaussian transitions by approximating the posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$.

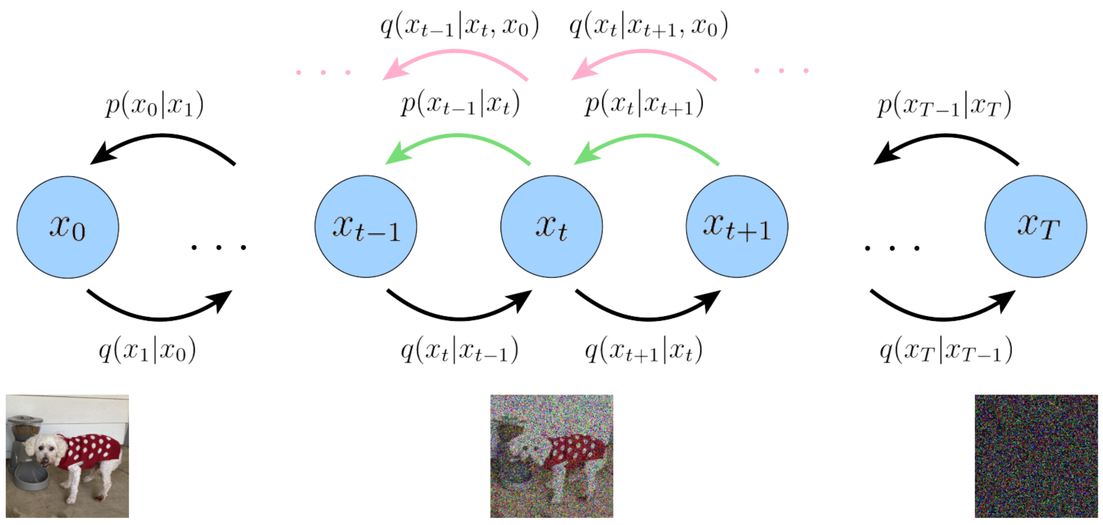


Figure 1: Visual representation of the forward process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$, the approximate denoising process $p_\theta(\mathbf{x}_{t-1}|t)$ and the ground-truth denoising process $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$.

The objective of the model is to maximize the likelihood $p(\mathbf{x})$ which is done by maximizing the ELBO for the log likelihood $\log(p(\mathbf{x})) \geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]$. After some derivation, the following expression for the ELBO can be achieved.

$$\log p(x) \geq \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]}_{\text{reconstruction term}} - \underbrace{D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]}_{\text{denoising matching term}}$$

- **Reconstruction term**: The log probability of the original sample given the first latent image.
- **Prior matching term**: Matches the final latent distribution with the prior. Since we assume that with large T, the final latent distribution is Gaussian, we can use a Gaussian prior, and the term effectively becomes zero.
- **Denoising matching term**: Matches all trained approximate denoising steps with their corresponding ground-truth denoising steps.

When optimizing, the denoising matching term dominates the reconstruction term, therefore we only need to minimize the denoising matching term. If we set the variances of the two distributions to be exactly the same, minimizing the KL term reduces to

$$\arg\min_\theta \frac{1}{2\sigma_q^2} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} \left[\|\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|_2^2\right],$$

where $\alpha_t := 1 - \beta_t$,, $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, and $\hat{x}_\theta(\mathbf{x}_t, t)$ is the model's prediction of $\mathbf{x}_0$ when only looking at the noise image at step t. This final optimization term essentially teaches the neural network to predict the ground-truth image from any latent version of it.
Equivalently, we can teach the neural network to predict the noise $\boldsymbol{\epsilon}$ instead of the ground-truth image $\mathbf{x}_0$. This means we can minimize the following expression

$$\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_q^2\alpha_t(1-\bar{\alpha}_t)}\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t)\|_2^2\right].$$

where $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$. If we ignore the weightings we arrive at the following simple loss:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t)\|_2^2],$$

i.e. simply the MSE between the true and predicted noise. Ignoring the weights causes us to focus more on denoising at the higher t values, which has been shown to give better performance.
To sample from the model, we start with $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then for $t = T, \dots, 1$ we do

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}}}\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z},$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = 0$.

Model architecture and experiments

The network $\hat{\boldsymbol{\epsilon}}_\theta(x_t, t)$ that predicts the noise is chosen as a pre-implemented U-Net which includes residual blocks and multi-head attention. It also takes positional embeddings of the timestep t.
We use $T = 1000$ timesteps and choose a linear schedule for the β_t values going from $\beta_1 = 0.0001$ to $\beta_T = 0.02$. We use the Adam optimizer with an initial learning rate of 2×10^{-5} which decays by 0.5 on certain milestone epochs. On MNIST, we train for 60 epochs and decay on epochs 20 and 40. On CIFAR-10, we train for 90 epochs and decay on epochs 30 and 60. We use a batch size of 128. After training we sample 12800 images. We repeat the experiment 3 times for each dataset.

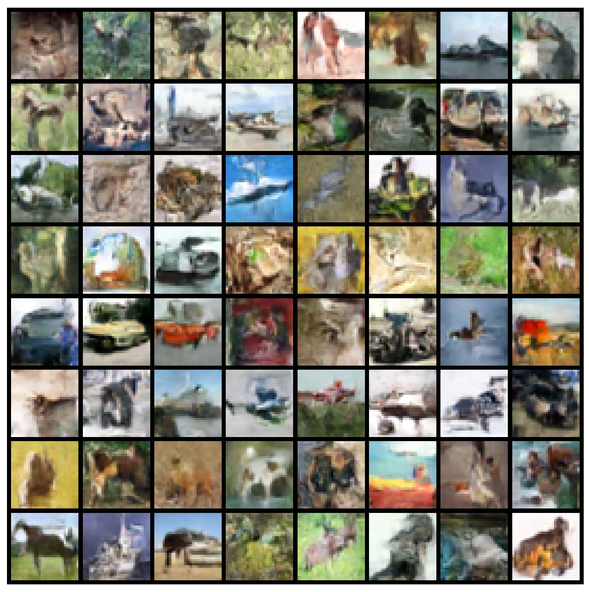
Results

We compute the Fréchet Inception Distance (FID) between the generated samples and the ground-truth dataset. We see satisfactory results, especially for MNIST.

Dataset	FID	Loss
MNIST	6.06 ($\pm$0.70)	0.0175
CIFAR-10	49.5 ($\pm$2.6)	0.0336

Table 1: Mean FID scores (with standard errors) and training loss values obtained

(a) MNIST generated samples



(b) CIFAR-10 generated samples

Figure 2: Visualization of some of the generated samples.

We can visualize the sampling process (reverse process) by viewing the latent image $\mathbf{x}_t$ at different time steps t, going from pure Gaussian noise to a clear image.

(a) MNIST reverse process

(b) CIFAR-10 reverse process

Figure 3: Visualization of the sampling process (reverse process) for $t = 1000, 900, \dots, 100, 0$.

Classifier-Free Guidance

In classifier-free guidance we allow the model to be conditioned on the class c without needing to train a separate classifier network. We have two models: an unconditional diffusion model $\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t)$ and a conditional diffusion model $\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, c)$. These models are learned together as a single conditional model, where the unconditional model is learned by replacing the conditioning information with zeros. When training, the conditioning information is randomly removed with probability p_{uncond}. This is equivalent to performing random dropout on the conditioning information. We use $p_{\text{uncond}} = 0.2$.
When sampling, we replace $\hat{\boldsymbol{\epsilon}}_\theta(x_t, t)$ by a linear combination of the noise predicted from the unconditional and the conditional model:

$$\tilde{\boldsymbol{\epsilon}}(\mathbf{x}_t, t, c) = (1+w)\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, c) - w\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t)$$

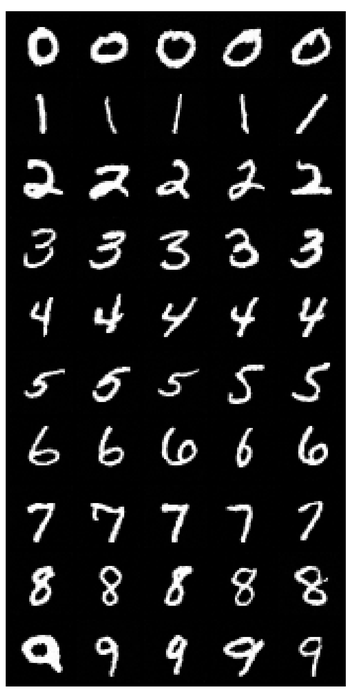
For $w = 0$, this is equivalent to just using the conditional model. For $w > 0$, we step away from the unconditional prediction to explicitly favor the conditional prediction. This can make the samples more closely match the conditional information at the cost of less sample diversity.

Results for Classifier-Free Guidance

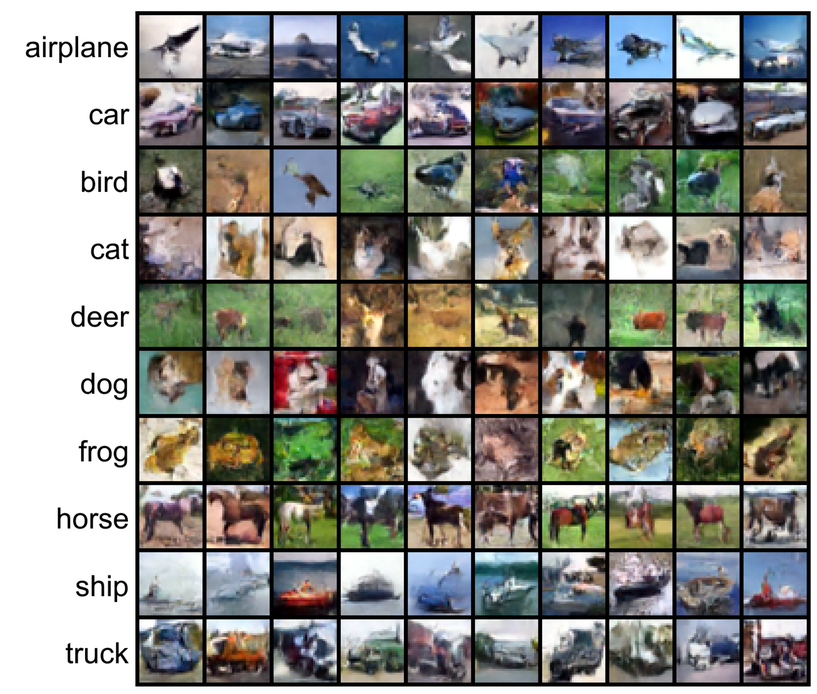
We sample 10% from each class using $w \in \{0, 0.5, 1\}$. We are able to improve the FID score using the conditional model. Interestingly, a low w is best for MNIST while a high w is best for CIFAR-10. Visually, the samples are more consistent with the class for high w.

Dataset	$w = 0$	$w = 0.5$	$w = 1$
MNIST	5.56	7.27	9.74
CIFAR-10	47.18	42.73	39.52

Table 2: FID scores using Classifier-Free Guidance.



(a) MNIST



(b) CIFAR-10

Figure 4: Guided conditional samples using Classifier-Free Guidance with $w = 1$.