Diffusion project

Exercise 3

Advanced Deep Learning in Computer Vision

September 2023

In this exercise, you are asked to build a diffusion model that generates 16×16 sprites:

Figure 1: Example sprites

You tasks are as follows:

1. Calculate β_t, α_t, and $\bar{\alpha}_t$ in the *__init__* function of the Diffusion class. (See files *ddpm.py* and *playground.py*)
2. Complete the implementation of forward process in the function *q_sample*. (See files *ddpm.py* and *playground.py*)
3. Complete the implementation of the reverse process in the function *p_sample*. (See files *ddpm.py* and *playground.py*)
4. Implement the training function. (See file *ddpm_train.py*). Training takes around 1 hour on a CPU with a reduced dataset size of 40K images. You should be able to see reasonable image generations between epochs 20-30 (tested with SEED=1). Set DATASET_SIZE to None if you want to train on the full dataset.

5. (OPTIONAL) Implement a cosine schedule in the function *get_betas* and analyze all the differences. See Figures 3 and 5 in the paper "Improved Denoising Diffusion Probabilistic Models"

Notation: In the lecture, we follow the notation of the ddpm paper, while in the code, we follow the notation from the OpenAI code repository. Here we provide a mapping between the two.

- T is the total number of diffusion steps
- x_t = image at timestep t
- $x_T \sim \mathcal{N}(0, \mathbf{I})$
- β_t = betas[t]
- α_t = alphas[t]
- $\bar{\alpha}_t$ = alphas_bar[t]
- $q(x_t|x_0)$ = q_sample
- $p_\theta(x_{t-1}|x_t)$ = p_sample