CS 261: Deep Generative Models
Homework 2 - Diffusion Models

Available: 02/22/2024; **Due Date:** 23:59 PM PST, 03/13/2024

---

1 Motivation

In this homework, you will implement a diffusion model with the UNet architecture. You will implement the core components of the model, including the architecture, the noise schedule, the forward diffusion process, the loss function, and the sampling of new images from a trained model. You will then use these as the basis to train a (small) diffusion model on a sprites dataset. The code skeleton is provided at `https://colab.research.google.com/drive/1Rat5t_pTKGflkSgEYMYHLK8UxYlwWc1E?usp=sharing`.

2 Logistics

Colab tutorial Students who are not familiar with Google Colab can checkout the tutorial at `https://colab.research.google.com/?utm_source=scs-index`.

Code structure We provided a code skeleton for the homework. Students are required to complete the code within the green comment blocks. See the figure below for an example. We also provided several simple test functions you can use as a sanity check for your implementation.

```
def construct_ddpm_noise(timesteps, beta1, beta2, device):
    beta_t = None
    alpha_t = None
    alpha_bar_t = None
    ##########################################################################
    # TODO:                                                                  #
    # Implement the noise schedule for the forward diffusion process.        #
    # Should return 3 tensors: beta_t, alpha_t, and alpha_bar_t              #
    # Output shape: [timesteps + 1]                                          #
    # Read the DDPM paper (sections 2 and 4) for how to                      #
    # construct beta_t, and compute alpha_t and alpha_bar_t                  #
    # Reference: https://arxiv.org/pdf/2006.11239.pdf                        #
    ##########################################################################
    ##########################################################################
    #                            END OF YOUR CODE                            #
    ##########################################################################
    alpha_bar_t[0] = 1
    return beta_t, alpha_t, alpha_bar_t
```

Submission After completing the Colab, run the last cell and upload the `my_diffusion_implementation.py` file to GradeScope.

Grading scheme There are 8 unit tests for the implementation of each code block and each unit test is given 1 point.

3 FAQ

Q: What is the relation between `beta_t`, `alpha_t`, `alpha_bar_t`?

A: β_t is the noise level we add to the input at time step t in the forward diffusion process, specifically:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}). \tag{1}$$

$\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ are derived from β_t. In practice, β_t is determined by a noise schedule. In this assignment, we use the noise schedule proposed by the original DDPM paper (`https://arxiv.org/pdf/2006.11239.pdf`), which sets the forward process variances to constants increasing linearly from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$.

Acknowledgements: Sprites by ElvGames, FrootsnVeggies and kyrise
Diffusion model is based on Denoising Diffusion Probabilistic Models