ECS795P Deep Learning and Computer Vision, 2024

Submitted by: Ruthwik Ganesh (230702930) – ec23759@qmul.ac.uk

Coursework 2: Diffusion Models.

Contributed by Kaviraj Gosaye.

1. Differences between Auto-Encoders, GANs (Generative Adversarial Networks), and Diffusion Models:

- Model Structure
  - Autoencoders' architecture consists of two parts: an encoder and a decoder. The task of the encoder is to compress the input data into a latent space representation. On the other hand, the decoder will use this latent space representation to reconstruct into the original input. The output of the encoder is used as input to the decoder.
  - GANs architecture consists of two convolutional neural networks: the Generator (G) and the Discriminator (D). The task of generator G is to create new data sampled from random noise. On the other hand, the task of the discriminator is to distinguish between the new data created by the generator G and the real data. Simply put, we can say that the generator's task is to generate fake data such that the discriminator cannot detect whether it is false or not.
  - Diffusion models are based on the laws of thermodynamics where an image is gradually destroyed with random noise in stepwise additions. This process is called the diffusion or forward process. The reverse process involves removing the noise from the noisy generated image at the end of the forward process.

- Objective Functions:
  - Autoencoder: The objective function of the autoencoder is to minimize the error between the input data and the output data of the decoder. This objective function can use metrics like binary cross-entropy loss or mean squared error (MSE).
  - GANs: The generator G of the model will learn the distribution of input images and has as objective to minimize the probability that the discriminator D identifies or detects that the image is synthetic one. On the other hand, the objective of the discriminator is to maximize this probability of detecting the fake image.
  - Diffusion Model: The objective function makes use of maximum likelihood estimation (MLE) for observing the data under this model. For diffusion models, the negative-log likelihood with regularization is typically used.

- Training of Components:
    - Autoencoder: The inputs are encoded to a latent representation during the forward pass by the encoder part. The output of the decoder is used for loss computation using MSE or binary cross-entropy and the gradients computed using backpropagation. The optimization algorithm for updating the model weights can be Adam, Stochastic Gradient Descent (SGD) or RMSprop.
    - GANs: The loss computation of generator G is calculated from the output of the discriminator. The loss function is usually a minimax loss or adversarial loss. The weight of the generator is updated using gradient descent algorithm. The loss computation of discriminator is to classify the fake and real data. This is computed as a binary cross-entropy loss.
    - Diffusion Model: The generator or the reverse process is denoising an image to reconstruct a noise-less image. During this process, learning occurs. The difference between the original image and the output of the model can be computed with Mean-Squared Error (MSE), perceptual loss or adversarial loss. Stochastic gradient descent (SGD) is used for training the model and the gradients computed using backpropagation.

Contributed by: Kaviraj Gosaye, Naveen Raj Govindaraj and Ruthwik Ganesh

2. UNet in Unconditional Diffusion Models:

a) Intermediate Feature Map Dimensions:
   i) Downsample Blocks:
      (1) 1st Block: [256, 256, 128] (from the hint)
      (2) 2nd Block: [128, 128, 256] (256 = 2 * 128)
      (3) 3rd Block: [64, 64, 512] (512 = 4 * 128)
   ii) Middle Block: Retains the dimensions of the last downsample block: [64, 64, 512].
   iii) Upsample Blocks:
      (1) 1st Block: [128, 128, 256] (halving channels)
      (2) 2nd Block: [256, 256, 128] (further halving channels)
      (3) 3rd Block: [512, 512, 128] (returning to original dimensions)
b) Integration with Attention Modules and Time-Step Embeddings:
   i) Attention Modules: In a UNet, attention modules are integrated to selectively focus on certain features of the input data at different scales. They are particularly useful in capturing long-range dependencies and subtle details that are crucial for the denoising process in diffusion models.
   ii) Time-Step Embeddings: Time-step embeddings provide information about the specific step in the diffusion process. They are integrated into the UNet to adapt its behavior based on the current stage of noise addition or removal. This temporal information is crucial for the model to understand how much noise to add or remove at each step.

Contributed by: Ruthwik Ganesh.

3. Noise Addition and Denoising Process in Diffusion Models:

1. Resultant Image Identity: After undergoing a process of noise addition and subsequent denoising, the resultant image is unlikely to be identical to the original.
2. Reason for Differences: Each step in the diffusion process involves adding noise and then attempting to reverse it. However, this process is imperfect, and some original details may be lost or altered during these transformations. The denoising process approximates the reverse of noise addition but does not perfectly invert it.
3. Effect of More Steps: Increasing the number of noising and denoising steps results in a more complex transformation process. While it can potentially lead to a more refined result due to the model having more steps to adjust the noise, it also increases the risk of deviation from the original image, as each step introduces potential for slight errors or changes.

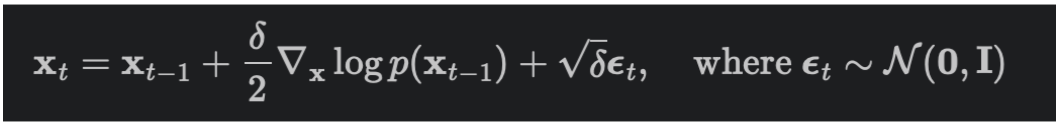
Contributed by: Naveen Raj Govindaraj.

4. Samplers in Diffusion Models:

1. Sampler Used for Training: Langevin-like sampler was used, this is a type of Markov Chain Monte Carlo (MCMC) sampler inspired by Langevin dynamics, a mathematical model that describes the behavior of particles subject to random forces. In the context of sampling from probability distributions, Langevin-like samplers are particularly useful for exploring high-dimensional and complex spaces.

   The Langevin-like sampler operates by simulating the dynamics of a particle moving through the parameter space of the target distribution. Like Langevin dynamics, the particle is subject to a drift term, which encourages movement towards regions of high probability density, and a diffusion term, which introduces randomness to explore space.

   Mathematically, the Langevin-like sampler updates the current state of the particle iteratively using the Langevin equation:



$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{\delta}{2}\nabla_{\mathbf{x}} \log p(\mathbf{x}_{t-1}) + \sqrt{\delta}\boldsymbol{\epsilon}_t, \quad \text{where } \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

2. DDPM (Denoising Diffusion Probabilistic Models) vs. DDIM (Denoising Diffusion Implicit Models):
   a. DDPM: This is a stochastic model where each step in the denoising process involves a probabilistic component, making the process inherently random.

b. DDIM: The Denoising Diffusion Implicit Models modify the sampling process to be less stochastic and more deterministic, allowing for faster and more predictable generation of samples.
c. Benefits of DDIM: DDIM offers increased efficiency with faster inference times. It generates samples in fewer steps without compromising much on the quality of the generated samples. This deterministic approach can be advantageous in applications where predictability and speed are crucial.