MIDTERM - CREATING IMAGES WITH DIFFUSION MODELS

ITAI 2376 - Deep Learning in Artificial Intelligence

Professor: Anna Devarakonda

Iffraah Rehman

ID: W216165700

Table of Contents

Assessment Questions

1. Understanding Diffusion

a. Explain what happens during the forward diffusion process, using your own words and referencing the visualization examples from your notebook.

The MNIST dataset has been used in the attached diffusion model. During the forward diffusion process in the said model random noise has been added gradually in order to clean an image over a series of steps, which in turn transforms it into pure noise. Each step increases the noise slightly, simulating how an image might get blur over time.

In the attached notebook, this is visually demonstrated using grayscale digits from the MNIST dataset, starting from a clear number like "3" or "7" and showing how it fades into an unrecognizable cloud of noise by the final timestep. This explains how the model first learns to degrade images into noise, which is later reversed during generation.

b. Why do we add noise gradually instead of all at once? How does this affect the learning process?

Noise is added noise gradually instead of all at once to make the learning process more stable and structured for the model. When the image is corrupted slowly in many small steps, the model can learn to denoise in manageable stages. Each stage each focuses on removing just a little bit of noise. This helps the model to understand the relationship between clean and noisy images at multiple levels of degradation.

If we added all the noise in one step, the connection between the original image and the noisy version would be lost, making it extremely difficult for the model to learn how to reverse the process. Gradual noise also creates a smooth, predictable transition from structure to randomness, which is very important during the training of the model to reconstruct high-quality images from pure noise in the reverse diffusion (generation) phase.

c. Look at the step-by-step visualization - at what point (approximately what percentage through the denoising process) can you first recognize the image? Does this vary by image?

During the step-by-step visualization in the notebook, an image can typically start to be recognized (for the MNIST dataset) around 70% of the way through the denoising process i.e. 30% of the steps is remaining. At this point (70%), the digit starts to get clearer from the noisy background, with its general shape becoming visible. However, this does vary by image: simpler digits like "1" or "7" (with fewer strokes) may become recognizable earlier, while more complex digits like "8" or "5" may need more steps before their structure is clear. This variation is due to how much detail the digit has and how well that detail survives the earlier noisy stages.

2. Model Architecture

a. Why is the U-Net architecture particularly well-suited for diffusion models? What advantages does it provide over simpler architectures?

The U-Net architecture is well-suited for diffusion models because it outclasses other models at maintaining spatial details while capturing high-level context. These characteristics are crucial for denoising and reconstructing images from noise. In diffusion models, the task is to gradually reverse the noisy image back to its original form. The U-Net's encoder-decoder structure with skip connections allows it to compress the image into abstract features (down sampling) while keeping access to fine-grained spatial information from earlier layers during reconstruction (up sampling). This helps the model accurately recover textures, edges, and digit shapes which is especially important for datasets like MNIST. These features make U-Net an ideal backbone for the reverse diffusion process, where both local pixel accuracy and global structure are needed.

b. What are skip connections and why are they important? Explain them in relations to our model

Skip connections are a crucial feature in diffusion models, that helps to overcome challenges in training deep networks by enabling smoother information flow. They help retain important details throughout the network, resulting in more reliable, efficient, and higher-quality image generation and other diffusion-related tasks.

In our diffusion model, skip connections are especially important because the model's goal is to reconstruct a clean image from noise, step by step. Without skip connections, the model might lose important fine-grained details (like edges or stroke shapes in MNIST digits) during the down sampling process. By connecting encoder and decoder layers at the same resolution, the model can reuse those preserved details during up sampling leading to more accurate and sharper image generation. In the U-Net, each Down Block produces a skip tensor, and each Up Block receives that skip tensor and concatenates it with the current decoder features. This helps the model align structure and detail across the network, making denoising more effective and generation more precise.

c. Describe in detail how our model is conditioned to generate specific images. How does the class conditioning mechanism work?

The attached model is conditioned to generate specific images for the chosen MNIST digit dataset. By using class conditioning, we guide the model to focus on generating a particular class (like a "3" or a "7") instead of random outputs.

Following is the overview of how the class conditioning works in the model:

i. Class Label Input: During both training and generation, we provide the model with a class label (e.g., the number 3).
ii. One-Hot Encoding: This label is converted into a one-hot vector, where the position corresponding to the class is set to 1 and the rest are 0. For example, the digit 3 becomes [0, 0, 0, 1, 0, 0, 0, 0, 0, 0].

iii. Class Embedding Layer: The one-hot vector is passed through an embedding network (Embed Block), which transforms it into a learned class feature representation (a tensor) that the model can use as a condition.
iv. Masking for Conditioning: A c_mask (a tensor of 1s) is applied to control whether the class conditioning is active.
v. Integration into U-Net: The embedded class vector is combined with the time embedding and added into the middle layers of the U-Net. This step allows the model to "know" which digit it's supposed to generate while performing the denoising.
vi. Guided Generation: When we run the reverse diffusion process (denoising), the model uses this class information to guide its predictions, so that it gradually constructs an image that resembles the specified digit.

By feeding the model this class-specific information, we give it the ability to generate controlled, targeted outputs rather than random noise allowing it to generate, for example, a clear digit "7" from pure noise.

3. Training Analysis (20 points)

a. What does the loss value tell of your model tell us?

The loss value in attached notebook's diffusion model tells us how well the model is learning to predict the noise added to images during the forward diffusion process.

i. Mean squared error (MSE) loss, which measures the difference between the model's predicted noise and the actual noise that was added at each timestep.
ii. A lower loss value means the model is accurately predicting the noise, which indicates it's learning how to denoise the image effectively. This is crucial because the better the model can predict and remove noise, the more realistic and accurate the generated images will be during the reverse diffusion process.

The evaluation of loss over training helps us understand, whether the model is improving (loss should decrease over time), and how well it's capturing the structure of the data (e.g., MNIST digits). If the loss remains high or doesn't improve, it suggests the model is struggling to learn the denoising task, possibly due to issues like insufficient training, poor conditioning, or architectural problems.

b. How did the quality of your generated images change throughout the training process?

Throughout the training process, the quality of the generated images gradually improved as the model learned to better predict and remove noise. In the early epochs, the generated MNIST digits appeared blurry, incomplete, or noisy, since the model hadn't yet learned the structure of the digits or how to effectively denoise. As training progressed and the loss decreased, the images became clearer and more defined. We could start to recognize digit shapes like "3", "7", or "0" more confidently.

c. Why do we need the time embedding in diffusion models? How does it help the model understand where it is in the denoising process?

We need the time embedding in diffusion models to tell the model which step of the denoising process its currently in. Since diffusion models work by gradually removing noise over many time steps, the model must know how much noise to remove at each specific step.

The time embedding encodes this step number into a format the neural network can understand, typically using sinusoidal embeddings. This embedding is then injected into the network, often in the middle of the U-Net, so the model can adjust its denoising behaviour based on how far along it is in the reverse process

Without time embeddings, the model wouldn't know whether it's supposed to remove a lot of noise (early steps) or just fine-tune small details (later steps), leading to poor or inconsistent results. So, time embeddings are essential for giving the model a sense of "where it is" in the step-by-step reconstruction of the image.

4. CLIP Evaluation (20 points)

a. What do the CLIP scores tell you about your generated images? Which images got the highest and lowest quality scores?

The CLIP scores in the attached notebook measures how well the generated images match their target labels semantically. A higher score indicates that the generated image resembles the proposed digit. In the notebook results, the digit with the highest CLIP score was '0', showing clear and accurate generation. The lowest CLIP score was for digit '4' and '9', which may have appeared ambiguous or less visually distinct. Overall, these scores provide a quantitative way to evaluate the realism and fidelity of our diffusion model's output.

b. Develop a hypothesis explaining why certain images might be easier or harder for the model to generate convincingly.

Easier digits (like 1, 0, or 7) have simple, consistent shapes with minimal strokes. These digits exhibit low variability, making it easier for the model to learn and reproduce them.

Harder digits (like 5, 8, or 3) have more curves, loops, or similar forms to other digits. These digits can overlap visually with other classes, increasing confusion during generation.

c. How could CLIP scores be used to improve the diffusion model's generation process? Propose a specific technique.

Based on our diffusion model used in notebook, the technique of CLIP-guided classifier-free guidance can be used to improve the diffusion model's output. In this technique in each denoising step, two candidate images (conditioned and unconditioned) are generated. CLIP is used to evaluate which aligns better with the desired label. The model

then blends these using a CLIP-weighted factor, refining the generation path toward semantically correct outputs without relying on explicit classifiers."

5. Practical Applications (20 points)

a. How could this type of model be useful in the real world?

The diffusion model showcases how AI can learn to construct structured images from noise based on class labels. In the real world, such models are useful in fields like:

i. Medical Imaging – for generate synthetic medical scans (e.g., MRIs, X-rays) for rare conditions to augment training datasets. It can help in training diagnostic models when real patient data is limited or imbalanced.
ii. Creative Design - Artists and designers can use diffusion models to generate sketches, logos, or concept art.
iii. Data Augmentation - Automatically generate labeled training data for machine learning models.
iv. Handwriting synthesis - Generate synthetic handwriting for personalized fonts or data augmentation in OCR training

While the notebook focuses on handwritten digits, the same architecture can scale to generate faces, clothing, or even complex scenes.

b. What are the limitations of our current model?

The model is trained solely on the MNIST dataset (28×28 grayscale digits). While MNIST is great for learning, it lacks complexity. The model cannot generalize to real-world images with color, texture, or higher resolution. The model wouldn't work well for more diverse image tasks like face generation or natural scene synthesis without major retraining.

c. If you were to continue developing this project, what three specific improvements would you make and why?

i. Add Classifier-Free Guidance for More Accurate Digit Control

Currently, the model uses class embeddings for conditioning, but doesn't implement classifier-free guidance, a technique where the model learns to generate both conditioned and unconditioned samples and blends them during generation to boost fidelity.

ii. Reduce Sampling Time with Faster Inference (e.g., DDIM)

The model uses 100+ denoising steps for every image, which is computationally expensive and slow.

iii. Expand to a More Complex Dataset (e.g., Fashion-MNIST or CIFAR-10)

MNIST digits are too simple and don't reflect real-world image challenges.