Cassy Cormier

Midterm Assignment

6253 Deep Learning 14724

Professor Anna Davarakonda

June 28th, 2025

Creating Images with Diffusion Models

Diffusion models are a major generative model that works by gradually adding and removing noise to the data until it becomes pure random noise, allowing the model to generate image and audio content (Data Preprocessing for Generative AI: Module Four, 2025). Known for their high-quality outputs and stability, this technical report is an analysis of the diffusion model that was trained with the Fashion MNIST dataset.

Understanding Diffusion Models

Progressive noise scheduling is a key function in the forward diffusion process. This forward diffusion process involves slowly adding noise to the data over a series of time. The noise schedule determines the amount of noise added at each stage. This schedule influences the quality of the generated content. A badly chosen schedule can lead to blurry or noisy outputs. The generated photos were quite blurry and not legible, which is indicative of the progressive noise scheduling needed to be fine-tuned (Data Preprocessing for Generative AI: Module Four, 2025).

The noise schedule has a major effect on the quality of the generated data. Gradually adding noise leads to better learning process and thus a better outcome (Data Preprocessing for Generative AI: Module Four, 2025). The image is totally unrecognizable from start to finish. At the beginning, the photos are extremely blurry. The photo becomes less blurry towards the end but it's still unrecognizable hinting at an issue in the noise schedule.

Model Architecture

UNet encoder-decoder architecture shares similar features as traditional autoencoder networks. Prior to 2015, annotated datasets were scarce, and the UNet network fixed this issue. It is major in computer vision as it is great at capturing context, precise localization, and utilizing skip connections in segmentation models with *limited* training data. It's able to understand context in the encoder through convolutional and pooling operations. This is what makes it particularly well-suited for diffusion models. The high-level features created from the encoder are run through the decoder to re-create high-resolution detailed content.

The UNet architecture introduces skip connections, which are connections that pass detailed feature maps directly from layers in the encoder to match layers in the decoder. The importance of skip connections is to ensure fine grained details are preserved and integrated into the final output. Traditional autoencoders have an encoder-decoder function but lack skip connections. Our model utilizes a UNet setup. The U-Net structure contains an:

- **Encoder**: 28x28 → 14x14 → 7x7 (with channels 32→64→128)
- **Decoder**: 7x7 → 14x14 → 28x28 (with skip connections from encoder)

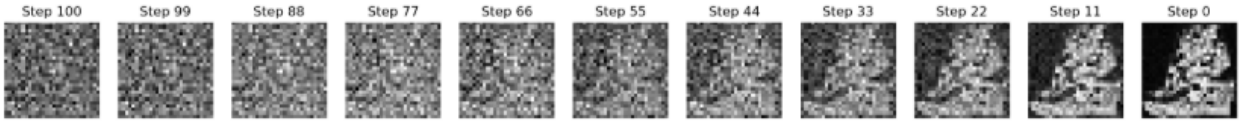
The model is programmed to use one-hot encoded class labels (10 classes) that get embedded into a 10-dimensional space. This is combined with time embeddings to provide both temporal and class conditioning. Skip connections are embedded in the model that match the encoder channels so that fine-grained detailed are preserved.

Training Analysis

The 31.8% improvement in training loss shows the model learned effectively. The final training loss of 0.1076 suggests good convergence. The validation results are particularly encouraging:

- Starting validation loss (0.1321) was lower than training loss (0.1577)
- Final validation loss (0.1083) is very close to final training loss (0.1076)
- Best validation loss (0.1069) is nearly identical to final training loss

This close alignment indicates excellent generalization - the model isn't overfitting and *should* generate diverse, high-quality samples.



- **Step 100** (leftmost): Pure noise - completely random pixel values
- **Steps 99-77**: Still very noisy, but you can start to see something emerging
- **Steps 66-44:** More defined shapes appear, the overall silhouette becomes clearer
- **Steps 33-22**: Recognizable object boundaries, looks like clothing item taking shape
- **Steps 11-0:** Details emerge, final clean result

The forward diffusion model is a Markov chain that describes a sequence of possible events. A Markov chain is a stochastic process where each change depends only on the previous step (Markov Chain, 2021). We need time embedding because at every step in the forward process, random noise is added over time. The steps represent time. By the final step of this initial process, the image is pure static.

Forward Diffusion Equation

$$x_t = \sqrt{1-\beta_t} \cdot x_{t-1} + \sqrt{\beta_t} \cdot \eta_t$$

β_t : Noise level at step t (β_t Controls static).

x_t : Data at step t

η_t : Standard Gaussian random variable

CLIP Evaluation

- What do the CLIP scores tell you about your generated images? Which images got the highest and lowest quality scores? CLIP is a model developed by OpenAI that stands for Contrastive Language-Image Pretrained. This model has a contrastive loss function to optimize the alignment of embeddings (Diffusion Models in Generative AI, 2025). CLIP rated 100% as “good” because it’s comparing very low-quality alternatives. When forced to choose between:

- "A trouser"
- "A clear, well-generated trouser"
- "A blurry or unclear clothing item"

CLIP picked the first options as “better” than the other two, but these look very blurry and like nothing to me. The architecture is too simple and needs more fine tuning. CLIP scores can be used to improve the data in the diffusion model (Diffusion Models in Generative AI, 2025). The CLIP model is working but it’s indicative of something wrong with the diffusion model.

Practical Applications

- How could this type of model be useful in the real world? I am currently interning at TRC in there Distribution Energy Department of the power sector. This diffusion model can be used to generate synthetic power grid plans, load patterns, and operational scenarios. The limitations of our current model include a strong generalization to new prompts, hence the need for the CLIP model.

Bonus Challenge

First, I want to verify that the Fashion MNIST data is loading correctly and try reducing the denoising steps initially. I believe the noisy outputs are indicative of potential data loading issues. Also, I would like to implement reduced steps to try and improve the sampling. My current sampling might be accumulating errors over too many steps. Lastly, I want to try enhanced CLIP evaluation with a ViT-B/16 instead of 32. Though ViT-B/16 requires more computational power, it’s more accurate assessment will help me determine the real issues (ChatGPT, n.d.).

References

ChatGPT. (n.d.). Retrieved from ChatGPT: https://chatgpt.com/

Data Preprocessing for Generative AI: Module Four. (2025, June). *ITAI 2377*. Houston, Texas, USA: HCC.

Diffusion Models in Generative AI. (2025, June). *HCC ITAI 2376 Module 9*. Houston, TX, USA: HCC.

Markov Chain. (2021, December 3). Retrieved from GeeksForGeeks: https://www.geeksforgeeks.org/machine-learning/markov-chain/