Deep Learning
Assignment 7 (Optional)

Diffusion Models

Deadline: 4th June 2024, 11:59 pm

Submission Guidelines:

Submission:

Submit all of your codes and results in a single zip file with the name FirstName_RollNumber_07.zip.

- Submit a single zip file containing.
  (a) codes (b) report (c) Jupyter Notebook (d) Readme.txt
- There should be a Report.pdf detailing your experience and highlighting any interesting results. Kindly don't explain your code in the report, just explain the results. Your report should include your comments on the results of all the steps, with images, for example, what happened when you changed the learning rate, etc.
- Readme.txt should explain how to run your code, preferably it should accept the command line arguments e.g dataset path used for training the model.
- The assignment is accepted in both .py and Jupyter notebooks. You can use Google Collab or Kaggle for GPU.
- In the root directory, there should be **1** python/notebook file, a report, and a folder containing saved models.
- The root directory should be named **FirstName_RollNumber_07**
- Your main code file should be named **rollNumber_07.py**
- Follow all the naming conventions.
- For each convention, there is a 3% penalty if you don't follow it.
- Email the instructor or TA if there are any questions. You cannot look at others' code or use others' code, however, you can discuss it with each other. Plagiarism will lead to a straight zero with additional consequences as well.
- NO LATE SUBMISSION WILL BE ACCEPTED.
- DON'T RESUBMIT THE DATASETS PROVIDED IN YOUR SUBMISSION.
- Report without code/experiment will be rewarded zero.

Due Date: 04/06/2024
Note: For this assignment (and for others in general) you are not allowed to **search online for any kind of implementation.** Do not share code or look at the other's code. You should not have any implementation related to the assignment, other than your own.

In case of any confusion please reach out to the TAs (email them or visit them during their office hours).

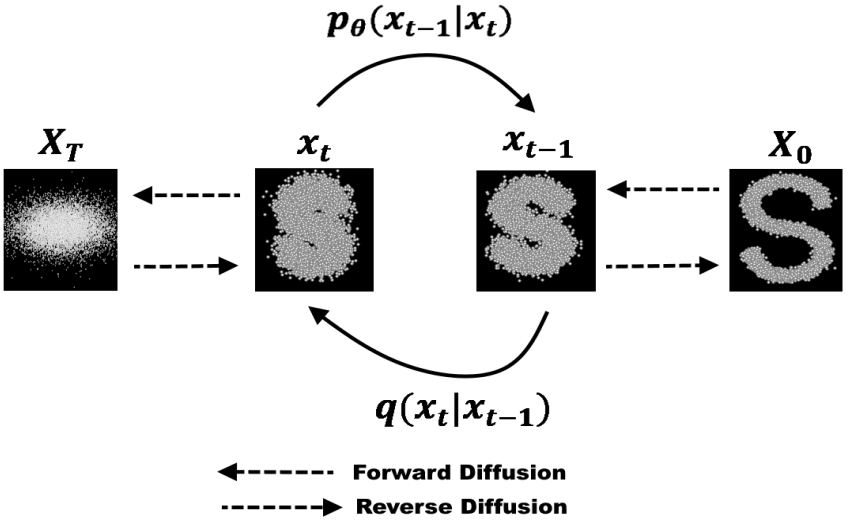
Objectives:

The purpose of this assignment is to provide a comprehensive understanding of diffusion models in machine learning, including their theoretical foundations, applications, and implementation. By the end of this assignment, students should be able to explain the concept of diffusion models, discuss their relevance in various fields, and demonstrate the ability to implement and evaluate these models.

- Understand about Diffusion Models.
- Understand how noise and denoise image.
- Understand how to implement the architecture of Diffusion Model.
- Understand how to generate images using Diffusion Model.

What is Diffusion Model? (Paper Link)

Diffusion models are a class of generative AI models that produce high-resolution images with varying quality. These models operate by progressively adding Gaussian noise to the original data through a forward diffusion process and then learning to remove the noise during the reverse diffusion process. As latent variable models, they reference a hidden continuous feature space and share similarities with Variational Autoencoders (VAEs). Diffusion models are conceptually inspired by principles from non-equilibrium thermodynamics.



Dataset

The dataset is related to the animal classes. There were 15 animal classes, and you can select a portion of images for your training (let's say 20 images) from any 5 classes. Apply necessary transformations on the data before feeding it to the process.

Model Architecture:

You can select any structure of the model for this assignment. Ensure that your structure meets the diffusion models procedure.

Working:

Sample a real image from the distribution of interest.

- Forward:

Over T steps (T=1000) repeatedly inject sample Gaussian noise until you get to a Gaussian distribution. T can be adjusted, but make sure your image looks like noise at the end of forward process.

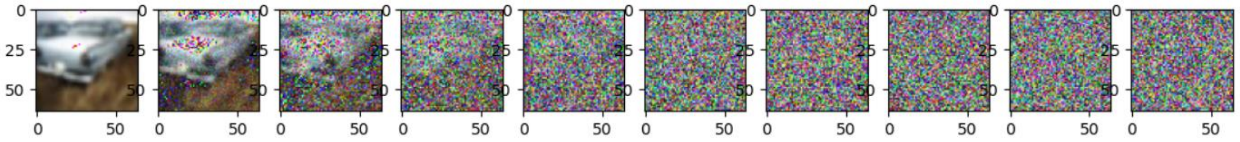


Figure 1: Sample Images of adding noise on T steps.

- Backward:
  - Reverse this process by taking noise away such that image returns to original distribution on original manifold.
  - The output for this process looks like Figure 1 from right to left (Noise to Image).
  - Every step of the model chooses carefully. Which layer, activation function is used (when and where). Misunderstanding about these, and you will lose a major portion of the marks.

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

You'll have to do this via neural network.

What you have to do:

1. Write a DataLoader class which will read images as expected for the model input process. (5)
2. Write a function for the forward process of the model (Adding noise upto T times). (10)
3. Create Model for the denoising process of the model. (10)
4. You can use L1, L2 or any related loss (Loss must be customized function). (5)
5. Once your model is trained, write a test function which will accept a noise and create an image from it.
6. Results generated by the model should be visible, and most of the portion is based on the results.
7. Make a separate file on which you will load your model. During evaluation you will be asked to open "**test_diffusion.ipynb**" and show your results. (10)
8. Make a detailed report about your findings, problems and solution regarding this assignment. Don't forget to add image samples, loss graphs, and added noise on some steps images. (5)
9. While evaluation technical questions related to any topic will be asked. Every portion contained marks of code, viva and results. Besides that, any other related topic questions will be asked as well. (5)

Useful Articles:

1. https://kailashahirwar.medium.com/a-very-short-introduction-to-diffusion-models-a84235e4e9ae
2. https://towardsdatascience.com/diffusion-models-made-easy-8414298ce4da
3. https://www.superannotate.com/blog/diffusion-models#:~:text=Diffusion%20models%20are%20advanced%20machine,learning%20to%20reverse%20this%20process.
4. https://deeplearning.cs.cmu.edu/S24/document/slides/Diffusion_Models.pdf (For mathematical understanding).

Good Luck!