NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

DT8122 **Project Assignment — Diffusion and BFN**

PROBABILISTIC AI

Summer 2024

1 Tasks

The project assignment is designed to give you hands-on experience with Diffusion and Bayesian Flow Networks. Results from training and evaluating the models on MNIST should be compiled into a report. The report should also cover discussion of results and methods as specified in the specific tasks.

1.1 Task 1: Diffusion

Your first task is implementing Diffusion [3, 2]. You are free to base your implementation on any research you see fit. For those looking for suggestions, [2] offers a nice balance of simplicity and performance. Your implementation should be trained on the MNIST dataset, readily available in most deep learning frameworks.

In addition to the code, you should include the following in your report:

- Plots:
  1. Plot of 9 or more generated digits.
  2. Plot of 1 or more generated digits across 8 or more diffusion steps, starting with an initial sample from the prior and ending with the "noise-less" generated image. See Figure 1
- Briefly discuss the network you used. Cover architecture, what it models, and how the diffusion step t is incorporated (if applicable).
- Discuss any challenges you faced and how you resolved them.

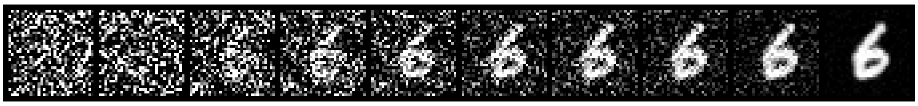

Figure 1. Example of one of the plots you have to generate.

1.1.1 Task 1 Rules and Notes

- You have to implement diffusion-specific functionality yourself, e.g., sampling/generating, loss function, calls to the network, etc. You also need to write all "structuring" code yourself, which means the code used to load data, process data, train the model, plot results, etc. **Note** You can use libraries for plotting and loading data, but you have to write the code that calls these libraries.

- You *are* allowed to use/take inspiration from existing network architectures as well as methods for encoding and incorporating the diffusion step t, e.g., an existing UNET implementation that encodes and adds/concatenates the diffusion step throughout the network.

- Note that some existing implementations may use extremely big networks that can't fit on some laptops and do not train in feasible time. We have ensured that it is possible to achieve acceptable results on MNIST on common laptop hardware (terminate within 3 hours while running on CPU and using 3GB memory during training). So, if you experience any hardware limitations, you should try to reduce the network size or take inspiration from a different implementation.

1.2 Task 2: Bayesian Flow Networks

Your second task is to read and discuss the Bayesian Flow Networks(BFN) paper [1]. You also have to train the model on the MNIST dataset to generate the same plots you did in Task 1, but you are allowed to use existing model implementations.

You should include the following in your report:

- Plots:

  1. Plot of 9 or more generated digits(sample from Output distribution).
  2. Plot of 1 or more generated digits across 8 or more steps, starting with a sample from the prior and ending with the "noise-less" generated image.

- Discuss the similarities of Diffusion and BFN. How do they differ? How do the methods compare when learning discreet distributions?
  - **Note:** We do not expect you to achieve a perfect understanding of the method and all of its bells and whistles. We recommend that you at least get a decent understanding of section 1-3, skim section 4 and 5, and listen/watch a presentation of the paper before writing this part of the paper.
- Compare the BFN results with those you got using Diffusion. How do the different models perform? Can a fair comparison be made based on your work in this project?

1.2.1 Task 2 Repo suggestions and tips

The authors have published their code on GitHub[1]. A popular alternative implementation[2] comes with a Jupyter Notebook for training the model on MNIST and may be easier to use for this project.

The following notes target the alternative implementation:

- If you cannot access GPU processing, you may want to reduce the number of epochs. We found that training for ~ 20 epochs produces satisfactory digits.
- The training procedure attempts to store the trained model in a folder named "models". You can avoid any issues by creating this folder before training.
- The notebook does generate plots akin to those we ask for, but for ease of comparison, we want you to produce identically formatted plots for diffusion and BFN. I.e., you should write your own plotting functions and use them for both models.

2 Submission Requirements

We expect you to submit the following:

- **Code** with your implementation of Task 1.
  - Your code should be readable and include any necessary comments.

[1] https://github.com/nnaisense/bayesian-flow-networks

[2] https://github.com/Algomancer/Bayesian-Flow-Networks

  - You should ensure the code is not dependent on a system-specific configuration. To ensure we can execute your code, please provide a list of packages (such as a requirements.txt file). It should be detailed in a README.md file how to install the packages and reproduce the results in the report.
  - All or part of the code may be a Jupyter notebook.

- **Report** that should include the plots and discussions asked for in tasks 1 and 2. Max. 8 pages (including figures).
  - The report should be a cohesive text with discussion about methods and results. The report should not be a collection of plots and bullet points.
  - If you do any preprocessing or changes to the methods, discuss those in the report.

Both the code and report should be in a private git repository on GitHub. The repository should have a README.md file detailing how to install necessary packages and reproduce the results in the report. Before the deadline, add the following user as a collaborator: DT8122. Do not make any changes after the deadline. Making changes after the deadline will result in an automatic fail. After adding DT8122 as a collaborator, send an email with a link to the repository to bjornar.vassoy@ntnu.no. Please title the email "DT8122 submission <YOUR NAME>". **The deadline is September 15. 23:59 AoE (Anywhere on Earth)**

References

[1] *Graves Alex, Srivastava Rupesh Kumar, Atkinson Timothy, Gomez Faustino.* Bayesian Flow Networks. 2024.

[2] *Ho Jonathan, Jain Ajay, Abbeel Pieter.* Denoising Diffusion Probabilistic Models // Advances in Neural Information Processing Systems. 33. 2020. 6840–6851.

[3] *Sohl-Dickstein Jascha, Weiss Eric, Maheswaranathan Niru, Ganguli Surya.* Deep Unsupervised Learning using Nonequilibrium Thermodynamics // Proceedings of the 32nd International Conference on Machine Learning. 37. Lille, France: PMLR, 07–09 Jul 2015. 2256–2265. (Proceedings of Machine Learning Research).