UNIVERSITY OF CAMBRIDGE

MPhil in Data Intensive Science

Submission: 11:59pm on Thursday the 28th of March

Coursework v1.0
Major Module M2: Application of Machine Learning
Dr. M. Cranmer

The coursework will be submitted via a GitLab repository which we will create for you. You should place all your code and your report in this repository. You should write a report with up to 3000 words that describes your work. Your report should contain figures and tables to support your analysis and discussion. The report should be in PDF format in a folder called "report". You will be provided access to the repository until the stated deadline. After this you will lose access which will constitute submission of your work.

The code associated with the coursework should be written in Python and follow best software development practice as defined by the Research Computing module. This should include:

- *Writing clear readable code that is compliant with a common style guide and uses suitable build management tools.*
- *Providing appropriate documentation that is compatible with auto-documentation tools.*
- *The project must be well structured (sensible folder structure, README.md, licence etc..) following standard best practice.*
- *Uses appropriate version control best practice, including branching for development and testing, and commit hooks to protect ‘main‘.*
- *Appropriate containerisation to ensure portability of the project to other computers and operating systems.*

*Note that extra marks will *not* be awarded for extremely high-performance models trained on large compute resources, so don’t worry about this aspect of things (though it is fine to seek the absolute best possible performance purely for your own personal satisfaction). The main thing considered in the grading of this report is the quality of your report and analysis, and demonstrated understanding of the course material and the problem. In other words, you can do this coursework entirely on a laptop.*

Introduction For this coursework, you will be tasked with building and training different types of diffusion models on MNIST in PyTorch. As part of this, you will be asked to design a custom type of diffusion model using a degradation strategy of your choice. The choice of degradation is quite open-ended; you should be creative in your choice.

Background Your primary resource should be to consult Chapter 18 of the Prince book for a detailed introduction to diffusion models. As a secondary, optional resource, you may find the paper demonstrating denoising diffusion probabilistic models for image generation by Ho et al., (2020) helpful in understanding the motivation of this class of model: `coursework/ho_2020.pdf`. Finally, I also include a blog post by Lilian Weng which describes diffusion models at a higher level: `coursework/weng_2021.pdf`.

***Note**: I break the following two questions into three parts each, a, b, and c. However, this is purely for the sake of breaking up the question itself and does not imply that the subparts are of equal difficulty or should be equivalent length.*

1 Training a Diffusion Model [40]

(a) First, you will train a regular denoising diffusion probabilistic model on MNIST as a warm up. In the coursework folder you will find the notebook `coursework/coursework_start.ipynb` which contains a working implementation of a diffusion model and training loop, on the MNIST dataset. Briefly describe this model and the training algorithm in your report.

(b) Document the process of training this model on the provided dataset, visualizing standard metrics such as the loss curve and the quality of the samples generated.

> *When I trained it with the default hyperparameters, I found it needed about 24 epochs to start consistently generating symbols (though those symbols didn't look like numbers at all yet), and maybe 50 epochs until I started seeing numbers get generated.*

Do this for two different sets of hyperparameters, and discuss the differences in the results.

(c) Present an analysis of each trained model. In this analysis, you should present both high quality and low quality samples from the trained model.

2 Custom Degradation [60]

(a) In the "Cold Diffusion" paper by Bansal et al., (2022), which can be found at `coursework/bansal_2022.pdf`, it was shown that a wide variety of image degradations can be used to train diffusion models, rather than only Gaussian noise.

Design a custom degradation strategy for images that you can use to train a diffusion model. You may use the Bansal et al. paper for guidance on this, but you should try to be creative in your choice of degradation, rather than simply using Gaussian noise or a close variant. Describe your degradation strategy in your report.

(b) Modify the code from the first part of this coursework to train and sample a diffusion model using your custom degradation strategy. (You may also modify the model architecture if desired.) Again, you should train this on the same MNIST dataset. As before, document the process of training this model, and present a detailed analysis of the results.

(c) After this, present a comparison of the two degradation strategies (i.e., the standard Gaussian noise degradation and your custom degradation strategy). Evaluate the fidelity of the samples generated by the two models, discussing any differences between them.

END OF PAPER