AIML 425 Assignment 4

James Thompson
Victoria University of Wellington
300680096

I. Introduction

This document reports the implementation and results of Assignment 4. I implement Stochastic Differential Equations and Ordinary Differential Equation models. THe SDE model is a score based generative model to learn a mapping from Gaussian noise to dog images. The ODE model is a normalizing flow model that learns a mapping from Gaussian to dogs and cats to dogs.

II. Theory

I will be looking at two models, a Stochastic Differential Equation (SDE) model and an Ordinary Differential Equation (ODE) model. Both models are generative models that learn a mapping from one distribution to another. The key difference being that the SDE model is score based and stochastic in generation while the ODE model is a normalizing flow model that is deterministic in generation.

A. Flows

B. Stochastic Differential Equations

C. Ordinary Differential Equations

III. Experiments

IV. Conclusion

Statement

The code and report of the Assignment was solely completed by myself (Thompson James). The complete source code can be found here https://gitea.james-server.duckdns.org/james/AIML425_assignment_4, with a link to a colab notebook found here: https://colab.research.google.com/github/1jamesthompson1/AIML425_assignment_4/blob/main/main.ipynb. A complete run through of the notebook takes about 10 minutes on a GPU enabled machine.

I have kept the appendix as concise as possible, however the code is setup to produce many more figures and tables that can be used to understand the models. Most of these figures have been generated and stored in the ‘figures‘ folder.

I completed my work using the following tools:

- **Jupyter Notebook [1] and JupyterText [2]:** For interactive development and hosting.
- **LaTeX**: For writing the report.
- **VSCode [3]:** As IDE, with Copilot to help with plotting and debugging.
- **JAX [4] and Flax [5]:** For implementing the neural network and training logic.
- **Matplotlib [6] and Pandas [7]:** For data visualization and management.

References

[1] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, “Jupyter Notebooks – a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds. IOS Press, 2016, pp. 87–90.
[2] M. Wouts, “Mwouts/jupytext,” Aug. 2025.
[3] “Microsoft/vscode,” Microsoft, Jul. 2025.
[4] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: Composable transformations of Python+NumPy programs,” 2018.
[5] J. Heek, A. Levskaya, A. Oliver, M. Ritter, B. Rondepierre, A. Steiner, and M. van Zee, “Flax: A neural network library and ecosystem for JAX,” 2024.
[6] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
[7] The pandas development team, “Pandas-dev/pandas: Pandas.”