FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Advanced Machine Learning
2024/2025 - 2nd Semester

Generating blood cells with ML

Nuno Lourenço
João Macedo
João Correia

1 Introduction

We present here the second practical project, part of the students' evaluation process of the Advanced Machine Learning course of the Master in Engineering and Data Science of the University of Coimbra. This work is to be done autonomously by a group of **two students**. The **deadline** for delivering the work is **2nd of June 2025** via Inforestudante. The quality of your work will be judged as a function of the value of the technical work, the written description, and the public discussion. All sources used to perform the work (including the code) must be clearly identified. If you use AI tools in the production of this work (e.g. ChatGPT), you must clearly identify all the parts in which the tool was involved. Please note that, during the defence, you will be required to demonstrate a deep understanding of the content generated by the tool, and this knowledge will be subject to evaluation. The document may be written in Portuguese or in English, using a word processor of your choice[1]. The **written report** is limited to **8 pages long**. The document should be well structured, including a general **introduction**, **a description of the problem**, the **approach**, the **experimental setup**, an **analysis of the results**, and a **conclusion**. The report should follow the **Springer LNCS format**. The Latex and Word templates are available in the Support Material of the course. The final mark will be given to each member of the group individually. To do the work the student may consult any source he/she wants. Nevertheless, plagiarism will not be allowed and, if detected, it will imply failing the course. While doing the work and when submitting it, you should pay particular attention to the following aspects (whose relative importance depends on the type of work done):

- description of the approach to the problem
- description of the general architecture of the methods used;
- description of the experiment, including a table with the parameters used which should allow full replication;
- description of the evaluation metrics used for the validation: quality of the final result, efficacy, efficiency, diversity, or any other most appropriate;

Do not forget, besides what was just said, that it is fundamental: (1) to do a correct experimental analysis; (2) to do an informed discussion about the results obtained; (3) to put in evidence the advantages of the chosen alternative.

[1] Latex is preferred

2 Problem Statement

Discriminative models focus on learning the boundary between classes by identifying distinguishing features—for example, classifying photographs of animals as either "cat" or "dog." In contrast, generative models aim to learn the joint probability distribution of the input data and its labels (or just the input distribution in an unsupervised setting). This allows them to generate entirely new data instances that resemble those in the training set. For example, a generative model can create a realistic image of an animal that was never seen during training. Popular generative models include Generative Adversarial Networks (GANs), which use a game-theoretic approach to synthesize high-quality data, and Variational Autoencoders (VAEs), which use probabilistic latent variables to reconstruct and generate data. By capturing the underlying structure and variability in the data, generative models can produce novel outputs that are statistically consistent with the original distribution.

In this work, you are tasked with developing generative models capable of producing high-quality samples that accurately reflect the underlying real-world data distribution. This involves not only implementing and training the models effectively, but also evaluating their performance using appropriate metrics. The ultimate goal is to generate samples that are both realistic and useful for downstream tasks or applications, such as data augmentation, simulation, or content generation.

3 Objective

The main objective is to analyse the dataset, prepare and train generative approaches that learn to generate more data for it dataset. To do that you should attend to the following objectives:

- Load and analyse the data;
- Prepare the machine learning pipeline for image generation;
- **Image Generation**
  Develop and experiment with generative machine learning models to synthesize images that follow the statistical distribution of the provided dataset. The goal is to produce realistic and diverse samples that closely resemble the original data.

- **Model Comparison**
  Implement and compare the performance of at least the following types of generative models:
  - **Autoencoders** (e.g., Variational Autoencoders, Denoising Autoencoders)
  - **Generative Adversarial Networks (GANs)** (e.g., DCGAN, Conditional GAN)
  - **Diffusion Models** (e.g., Denoising Diffusion Probabilistic Models)
- Compare the results of the previous models visually and analytically (see section 3.2).

Exploring alternative generative models beyond the ones listed, provided that they are suitable for the task, will be considered as additional work. These efforts may serve as compensation for any shortcomings or issues in the implementation or results of the required models.

3.1 The Dataset

BloodMNIST Dataset BloodMNIST is a medical imaging dataset composed of RGB images of peripheral blood smear cells, curated specifically for machine learning applications in hematology. It contains over 17,000 labeled images spanning 8 different blood cell types. These are:

- Neutrophils
- Eosinophils
- Basophils
- Lymphocytes
- Monocytes
- Immature Granulocytes
- Erythroblasts
- Platelets

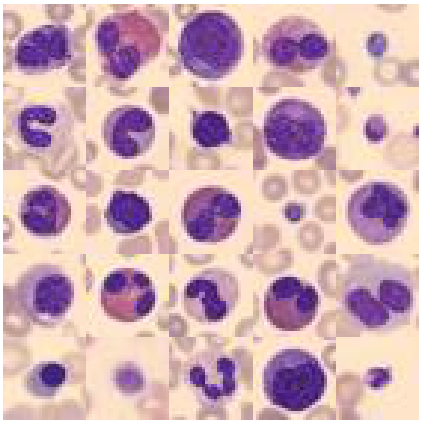

Figura 1: Example of the images from the BloodMNIST dataset.

Figure 1 shows representative examples from the dataset.

The dataset is originally divided into training, validation, and test sets following a 7:1:2 ratio. Each image is a 28×28 pixel color scan centered on a single blood cell. However, for this generative modeling task, you are allowed to use the entire dataset for training purposes.

3.2 Evaluation

The evaluation of each generative model must follow the methodology outlined below:

- Randomly sample 10000 real data points from the full dataset.
- Generate 10000 synthetic samples using the generative models.
- Compute the Fréchet Inception Distance (FID) between the real and generated samples to assess similarity.

Since generative models are inherently stochastic, you must perform **5 independent runs** using different random initialization seeds. Report the average FID score across these runs to assess the model's overall performance.

3.2.1 Fréchet Inception Distance (FID)

The Fréchet Inception Distance (FID)[2] is a widely used metric for evaluating the quality of images generated by generative models. It compares

[2]The FID metric was introduced by Heusel et al. in *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*, NeurIPS 2017.

the distributions of real and generated image features, which are extracted using a pre-trained InceptionV3 network. The FID score is computed as the Fréchet distance between two multivariate Gaussians fitted to these feature representations.

Formally, let $\mathcal{N}(\mu_r, \Sigma_r)$ and $\mathcal{N}(\mu_g, \Sigma_g)$ be the Gaussian distributions fitted to the real and generated data, respectively. The FID [3] is given by:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2}\right)$$

where μ_r, Σ_r are the mean and covariance of the real image features, and μ_g, Σ_g are the mean and covariance of the generated image features. A lower FID indicates that the generated samples are more similar to the real data in terms of both visual fidelity and diversity.

In this assignment, FID should be computed between 10000 real samples randomly selected from the dataset and 10000 generated samples. The evaluation must be repeated over 5 independent runs with different random seeds, and the average FID score should be reported.

For implementation in PyTorch, we recommend using the `pytorch-fid` library, specifically `fid_score` function. Alternatively, for custom pipelines, feature vectors can be extracted manually using a pre-trained InceptionV3 model from `torchvision`, and FID can be computed using NumPy and `scikit-learn`'s mean and covariance functions.

4 Conclusion

A few short comments. First, the control of the progression of your work will be done during the classes (T and PL). Moreover, you can discuss eventual problems by presenting yourself during office hours. Second, the projects reflect for the most part your current knowledge. The rest will be the object of lecturing soon. Third, we try to balance the difficulty of all the work, but we are aware that this is not an easy task and it is somehow a subjective matter. Fourth, we try to ask for a workload compatible with the value of the work for the final mark.

Methodological issues, like the statistical background, were elucidated during the previous lectures. You may use the statistical tool you feel at ease with, including the Python code that was provided. Finally, even if this is a work that asks you to do simulations and analyse the results, i.e., it has

[3] The trace operator, denoted $\text{Tr}(\cdot)$, returns the sum of the diagonal elements of a square matrix. In the FID formula, it reflects the difference in shape and scale between the covariance matrices of real and generated data.

a practical flavour, there is however a theory behind the work, and you are advised to consult the necessary literature.

Good luck!