RAPPORT DE PROJET

Metropolis-Hastings Generative Adversarial Networks

Ouleymatou Kone Kante
Léa Yang
DLIASO

lea yang

In the classical GAN approach, once the training is complete, the discriminator is often no longer used because it is assumed that the generator has become sufficiently effective to operate alone, whereas the generator, even when trained, is not perfect. It may continue to produce errors, artifacts, or low-quality images that end up in the result.
This is where the method studied in this project comes into play, the Metropolis-Hastings algorithm. The central idea is not to forget the discriminator, but to reuse it as a quality filter between the generator and the final user. Instead of blindly accepting all images generated by the generator, we use the discriminator to evaluate their likelihood. Thanks to an iterative process (a Markov chain), the Metropolis-Hastings algorithm will navigate the space of images and preferentially accept those that the discriminator deems most realistic, while rejecting the others. This helps clean the final distribution and artificially increase the quality of the outputs.

I- Metropolis-Hastings Algorithm

- **Using the discriminator as a density guide:**

The MH-GAN approach we implemented does not seek to modify the generator's weights. It aims solely to correct its outputs. The discriminator D is no longer used as an adversary, but as a density estimator. The objective is to build a Markov Chain in which the generator proposes independent candidates at each iteration, and where a probabilistic mechanism decides whether to accept or reject them. To sort the generated images, a mathematical criterion is required to compare their likelihood. Theoretically, as proposed by Turner et al. [2], this criterion is based on the ratio between the real data density $p_{data}$and the generator-induced density p_G. Since these densities are not explicitly accessible, the discriminator is used as an estimator. Under the theoretical assumption of an optimal discriminator, this ratio can be written as:

$$ratio = \frac{D(x)}{1 - D(x)}$$

This quantity acts as a quality score. If the discriminator is very confident that an image is real, then D(x) tends towards 1, the denominator becomes very small, and the ratio takes on a high value. Conversely, if the discriminator considers the image to be fake, D(x) tends towards 0 and the ratio becomes close to zero.

- **Metropolis-Hasting acceptance mechanism:**

The Metropolis-Hastings algorithm uses this ratio to navigate from one image to another. At each iteration, we have a current image x_K, and the generator proposes a new candidate image x'. The decision to accept or reject this proposal is based on the acceptance probability alpha:

$$\alpha = \min\left(1, \frac{ratio(x')}{ratio(x_K)}\right)$$

If the candidate is judged to be of better quality than the current image, then alpha equals 1, and the image is automatically accepted. If the candidate is of lower quality, it is not necessarily rejected; a random number u, between 0 and 1, is drawn, and the candidate is accepted if u exceeds the acceptance threshold. This process is repeated Ktimes. The larger the value of K, the more opportunities the algorithm has to reject poor samples and converge to an image considered optimal by the discriminator.

The implementation of the Metropolis-Hastings algorithm on MNIST was carried out using an iterative approach optimized by batch processing. We initiate the process by generating a large set of 1000 images $K = 0$ from random noise, a sample size chosen to ensure the statistical significance of the Inception Score calculations. The program is driven by a list of target steps $K = [0, 10, 100, 600]$ and a main loop that runs until the maximum of this list is reached. If the loop counter corresponds to one of these values, the current state of the images is automatically saved

From a technical standpoint, this loop integrates two critical utility functions: a numerical safety check to prevent division by zero when calculating the ratio, and a denormalization function. The latter is indispensable for visualization because the final layer of the generator (Tanh activation) produces values between -1 and 1, which are unreadable for standard display; we therefore apply an affine transformation $\frac{x+1}{2}$ to remap the pixels to the $[0, 1]$ interval. Finally, the code systematically records the average acceptance probabilities at each step, data collection that will prove crucial for diagnosing the discriminator's behavior in the following section

II- Evaluation Metric and Improvement

Although visual inspection provides an initial intuition about the improvement in image quality, it remains subjective and difficult to generalize at scale.
To quantitatively validate our approach, we use three metrics that measure the effectiveness of filtering (alpha and $D(x)$) and the quality of the results (IS inception score).

- The Inception score relies on a pre-trained neural network (Inception v3) that simultaneously evaluates two criteria: the quality and the diversity of the generated images.
  Quality measures the classifier's confidence in its predictions: an image is considered good quality if the network can clearly associate it with a given class (for example, "this digit is a 7" with high probability). Diversity, on the other hand, evaluates the generator's ability to produce all classes in a balanced manner. A model that would produce only one digit, even perfectly, would score poorly. The final score combines these two aspects using Kullback-Leibler (KL) divergence, which measures the distance between the conditional prediction distribution for an image and the average distribution over the entire generated set.
  A high Inception Score thus indicates images that are both realistic and diverse.

$$IS = \exp\left(E_x[KL(p(y \mid x) \;||\; p(y))]\right)$$

The IS is an indicator of convergence: a high and stable value proves that the MCMC process (at each iteration k) has effectively filtered bad samples and improved the GAN's distribution.

- $D(x)$, the probability from the discriminator that the image is real.

$$D(x) = \frac{p_D(x)}{p_D(x) + p_G(x)}$$

  If the image is real: $p_D$is high, $p_G$is small, and $D(x) = 1$.
  If the image is fake: $D(x) = 0$.
- The acceptance rate alpha is a fundamental metric in the Metropolis-Hastings algorithm; it is the probability of accepting a new candidate sample to advance the Markov chain from the current state.

$$\alpha = \min\left(1, \frac{ratio(x')}{ratio(x_K)}\right)$$

  The goal is not to maximize alpha but to optimize it: a very low alpha indicates poor mixing, and a very high alpha indicates a loss of selectivity of the filter.

- **Problem of Overconfidence in the Discriminator:**

Despite the overall improvement in image quality as Kincreases, we observe the persistence of parasitic points or background noise around certain digits. These defects can be explained by the nature of the discriminator trained in a classical manner, which often exhibits overconfidence.
Instead of producing nuanced probabilities (for example, 0.7 or 0.8 for a correct image), the discriminator tends to assign extreme scores, close to 0 or 1 (figure 1 before calibration). This phenomenon poses a critical problem in the acceptance ratio calculation. Indeed, when $D(x)$ is very close to 1, the term $1 - D(x)$ becomes extremely small, which, when divided, causes the ratio to explode. Thus, a slightly noisy

image but classified as "very real" may get an artificially high score, far greater than that of clean but correctly evaluated images. The Metropolis-Hastings algorithm then gets stuck in an absorbing state, refusing to replace this image, which freezes visual defects in the results. This results in a low acceptance rate alpha (figure 2), making the MH ineffective and leading to a loss of diversity and a drop in the IS.

- **Calibration:**

To address the overconfidence issue in the discriminator, we decided to use two different calibration methods:

- Temperature scaling: Initially, $D(x) = \text{Sigmoid}(z)$.
  We then introduce a parameter in the equation $D(x) = \text{Sigmoid}(z/T)$. In our model, we chose $T = 2$. This smoothes the output distribution. It is one of the simplest and quickest calibration methods to implement and does not modify the model's classification outcome (real image or fake).
- Isotonic regression: In the studied paper, it is said that isotonic regression is the most effective calibration method for MH-GAN based on their experiments. It is a non-parametric calibration method used primarily for binary classifications (our case). It groups the discriminator's predictions into intervals (bins) and assigns each interval a new calibrated probability based on the actual observed accuracy in that interval.

III- Results and Challenges

Applying Temperature Scaling, we obtained the following graphs:

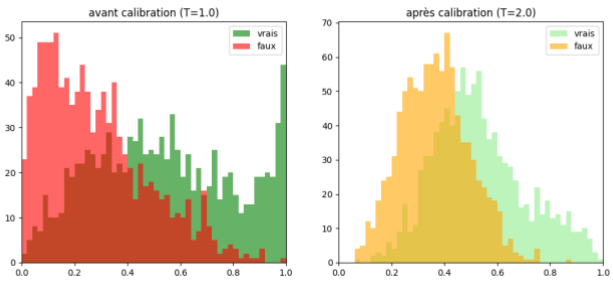

Figure 1: Discriminator scores distribution before and after calibration with Temperature Scaling (T=1.0 and T=2.0).

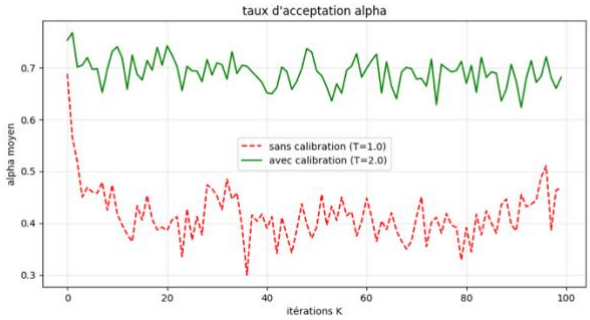

Figure 2 : Average acceptance rate alpha as a function of iterations (K) with and without calibration (T=1.0 and T=2.0)

These graphs confirm what we previously mentioned about the importance of calibration and the overconfidence of the discriminator: in figure 1, the probability from the discriminator that the images after calibration are centered around 0.5, showing that calibration has resolved the overconfidence issue, leading to a higher and more stable average acceptance rate alpha, oscillating between 0.6 and 0.8. Before calibration, alpha varies considerably throughout the iterations, indicating instability in the Metropolis-Hastings process. This can lead to convergence issues in the model.

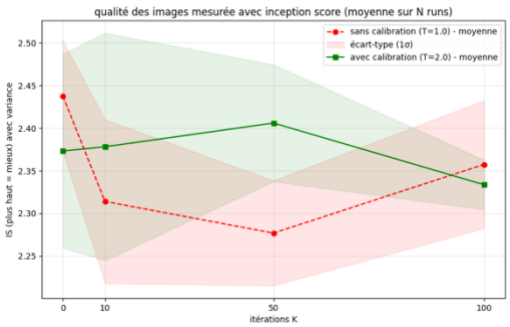

Figure 3: Quality of generated images measured by Inception Score (average over N runs) without calibration (T=1.0) and with calibration (T=2.0) (on 50 runs) with batchsize =100.

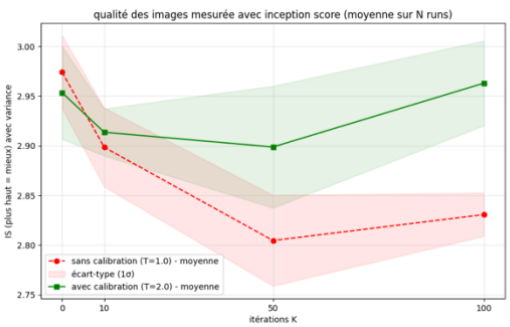

Figure 4: Quality of generated images measured by Inception Score (average over N runs) without calibration (T=1.0) and with calibration (T=2.0) (on 50 runs) with batchsize =1000.

However, we noticed that each time we reran the code, the average Inception Score curves behaved differently. We therefore implemented the standard deviation of the IS curves based on the batch size (batch size = 100 initially, then batch size = 1000) and observed a lower variation in the standard deviation of the average Inception Score curves (across 50 runs of the code) for a batch size of 1000 (especially before calibration). In both figures, we can generally say that calibration increases the IS score, but we see that in figure (3) with a batch size of 100, the IS curve with calibration is decreasing, while in figure (4) with a batch size of 1000, it is increasing. The acceptance rate alpha is also more stable when the batch size increases. (figures 5 and figure 6)

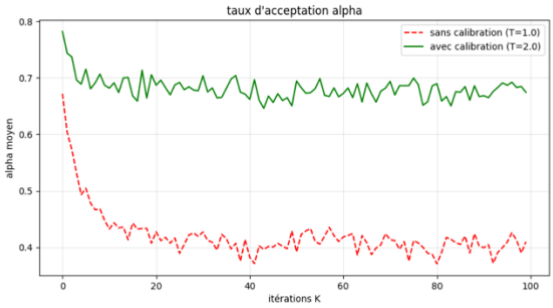

Figure 5 : Average acceptance rate alpha as a function of iterations K, with and without calibration, batchsize= 100

Figure 6 : Average acceptance rate alpha as a function of iterations K, with and without calibration, batchsize= 1000

The calibrated IS generally achieves a better Inception Score than the basic unmodified GAN, confirming what the paper suggests. We also wanted to see if isotonic regression improved the results, so we increased the number of iterations *K* to see how things

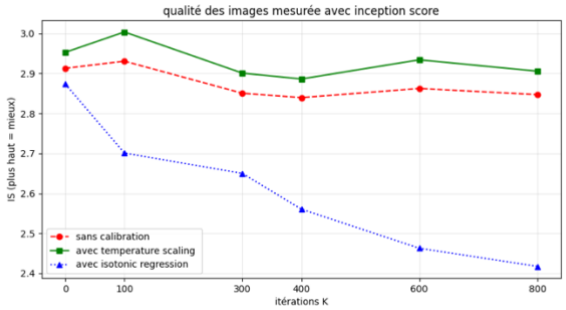

Figure 7 : Quality of generated images measured by Inception Score with and without calibration (T=1.0, T=2.0) and with isotonic regression

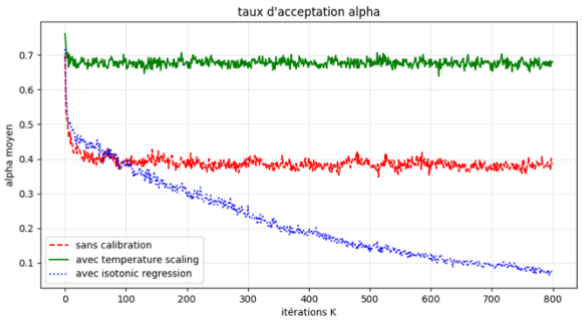

Figure 8 : Average acceptance rate alpha as a function of iterations K, with and without calibration and with isotonic regression

Contrary to our expectations and what the paper suggests, isotonic regression did not yield good results for our Inception Score and alpha:

- Isotonic regression seems to have reduced the quality of the generated samples; hence the decreasing IS curve. This suggests that, while isotonic regression improves the calibration of the discriminator's scores, it does not necessarily lead to an improvement in the perceived quality of the generated images. One hypothesis is that isotonic regression, by adjusting the output function too strongly, reduces the diversity of the generated samples, negatively affecting the overall quality of the generative model, as measured by the Inception Score.
- The IS was decreasing at each iteration, whereas with Temperature Scaling, it increased compared to the curve without calibration. Similarly, the acceptance rate alpha decreased with isotonic regression, tending toward 0 as the iterations progressed. It is possible that this modification of the distribution was not enough to compensate for the other negative effects of isotonic regression on image generation.

The loss in image quality could be caused by several factors, here are our hypotheses:
Isotonic regression works less well with a small dataset like ours, making it more difficult for it to find a stable monotonic relationship between the discriminator's scores and the real or fake classes of the images. This could lead to imprecise calibration, preventing the model from producing high-quality images, as shown by the decrease in the IS. Moreover, the reduction in acceptance rate alpha suggests that the model became too confident, rejecting a large portion of the generated images, which hindered continuous improvement in image quality.

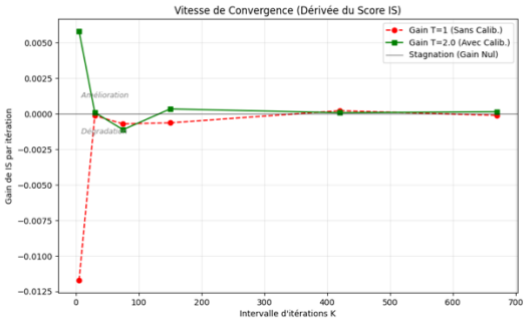

Figure 9 : Speed of convergence (derivative of Inception Score) as a function of iterations K with and without calibration (T=1.0 and T=2.0).

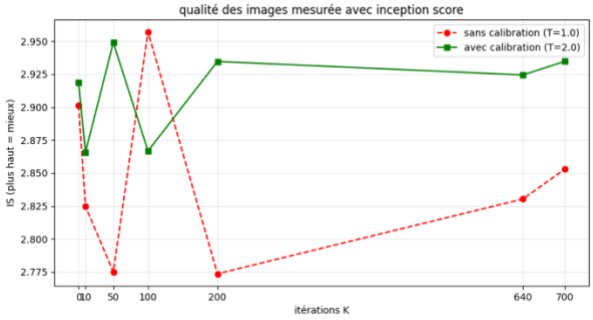

Figure 10 : Quality of generated images measured by Inception Score (IS) as a function of iterations K with and without calibration (T=1.0 and T=2.0).

As previously mentioned, the Inception Score is imperfect and sensitive to variations in the test samples or the absence of fine details in the images. It does not capture all the quality of the generated images, as stated in the paper. Moreover, the paper also mentions that gains in IS are not always monotone as the training iterations progress. There can be oscillations or periods where the IS does not continue to increase, which means the Markov chain is still adapting and exploring. According to the paper, most of the gains in Inception Score occur within the first 100 iterations with improvements becoming smaller beyond 400 iterations. This is observed in the convergence speed graph (figure 9), which shows the derivative of the IS score over iterations, reflecting the evolution of the model's gains at each iteration. We observe that at the beginning of the training, the model makes rapid progress, with a high positive gain, especially in the early stages. However, after a certain point, around 100 iterations, the gains begin to diminish, and the model reaches a phase of stagnation. It indicates that the gains become smaller, and the Markov chain approaches an equilibrium where it stops generating significant improvements. In summary, the graph confirms that the model quickly reaches a convergence phase after the first iterations. Calibration (T=2.0) seems to improve this convergence, offering a reduction in fluctuations and faster stabilization of gains.

This project explored the use of Metropolis-Hastings to improve the quality of images generated by a GAN, with a focus on optimizing the discriminator and calibrating the scores. The results showed that isotonic regression led to a reduction in diversity and overconfidence problems, negatively impacting image quality, while Temperature Scaling showed more stable and beneficial results. The current results show a FID of 25,15, a precision of 0.53, and a recall of 0.21. These metrics suggest good overall quality but also highlight areas for improvement in fidelity and the model's ability to capture all data variations. We could try other calibration methods, increase the dataset size, and improve the generator before passing it through a Metropolis-Hastings algorithm. We could also try optimizing the model's parameters, such as the batch size or the number of iterations, using techniques like random search or grid search, though these would be more costly to implement due to the need for multiple iterations of the Markov chain, making it impractical for real-time applications. We could also try to improve the generator's quality using others methods because MH doesn't improve the generator and if it suffers from mode collapse, the MH algorithm can only select from low-quality samples. Future improvements could involve gradient-based methods like MALA or HMC, which would refine the image generation process by using discriminator gradients, potentially offering smoother transitions and higher acceptance rates."

Références

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative Adversarial Networks. Advances in Neural Information Processing Systems (NeurIPS).

[2] Turner, R., Hung, J., Frank, E., Saatchi, Y., & Yosinski, J. (2019). Metropolis-Hastings Generative Adversarial Networks. International Conference on Machine Learning (ICML).

[3] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved Techniques for Training GANs. Advances in Neural Information Processing Systems (NeurIPS).

[4] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. International Conference on Machine Learning (ICML).