Generative Adversarial Networks

Alexandre Olech
Lucas Rousselet
Kanupriya Jain

Introduction

General problem :

How can we improve the quality and diversity of the generation by working on the latent space ?

We decided to work on gaussian mixtures

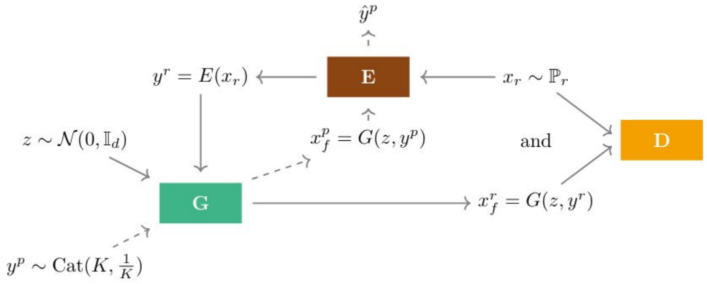
Plan

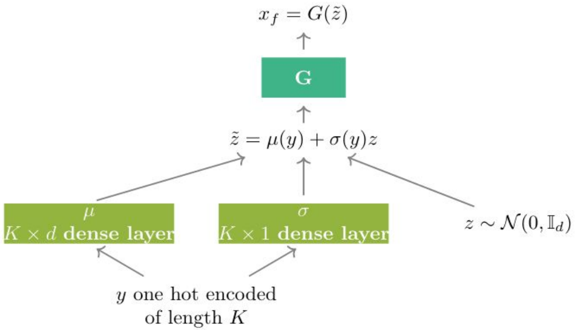
I. MMGan and GMgan
II. Variations of GMgan
III. Results and interpretation

I. MMGan and GMGan

MMGan

Pandeva, T., & Schubert, M. (2019). *MMGAN: Generative Adversarial Networks for Multi-Modal Distributions*. arXiv preprint arXiv:1911.06663.





I. MMGan and GMgan

MMGan

Pandeva, T., & Schubert, M. (2019). *MMGAN: Generative Adversarial Networks for Multi-Modal Distributions*. arXiv preprint arXiv:1911.06663.

Basic Idea :

- **Add an encoder that outputs a distribution of probabilities over classes**

Difficulties :

- **Hard to find the correct balance between the three models, leading to vanishing gradients**
- **Encoder should be strong enough to predict the right clusters, otherwise the generator cannot learn properly and collapse modes**

Algorithm 1 MMGAN

1: **Input:**
  K: number of clusters
  train_iter: number of training iterations
  m: batch size
  d: noise dimension
  α: hyperparameter

2: Initialize $\theta_D^0, \theta_G^0, \theta_E^0$.
3: **for** $t = 1$ to *train_iter* **do**
4:   Sample x_r from data of size m.
5:   $y = E(x_r; \theta_E^{t-1})$
6:   Sample $y^p \sim \text{Cat}(K, \frac{1}{K})$ of size m.
7:   Sample $z \sim \mathcal{N}(0, \mathbb{I}_d)$ of size m.
8:   $x_f = G(z, y; \theta_G^{t-1})$
9:   $x_f^p = G(z, y^p; \theta_G^{t-1})$
10:   $l_D^{t-1} = -\frac{1}{m}\Big(\sum_{i=1}^m \nabla_{\theta_D} \log(s(C(x_r^{(i)}; \theta_D^{t-1}) - C(x_f^{(i)}; \theta_D^{t-1})))\Big)$
11:   $l_{G,E}^{t-1} = -\frac{1}{m}\Big(\sum_{i=1}^m \nabla_{(\theta_G, \theta_E)} \log(s(C(x_f^{(i)}) - C(x_r^{(i)}))) + \alpha \log p_E(y^{p(i)} | x_f^{p(i)})\Big)$
12:   Update θ_D^{t-1} by Adam with gradient l_D^{t-1}.
13:   Update $(\theta_G^{t-1}, \theta_E^{t-1})$ by Adam with gradient $l_{G,E}^{t-1}$.
14: **Output:**
  D, G, E

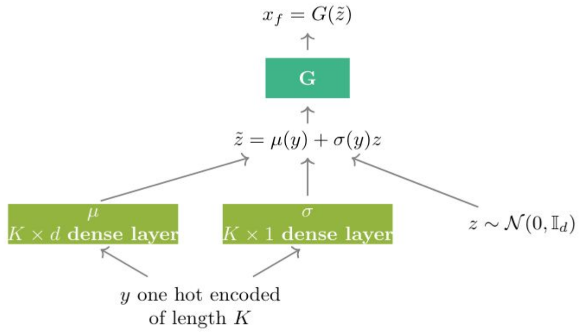
I. MMGan and GMGan

GMGan

Benyosef, M., & Weinshall, D. (2018). *Gaussian Mixture Generative Adversarial Networks for Diverse Datasets, and the Unsupervised Clustering of Images*. arXiv preprint arXiv:1808.10356.

Basic ideas :

- **Add a Gaussian Mixture model that consists in two two Linnears (means and std), that can be learned (dynamic) or not (static)**
- **The discriminator now outputs a distribution of probabilities over classes, replacing the encoder**
- **The losses are cross-entropies on the (softmaxed) outputs of the discriminator, compared with labels**
- **The generator is forced to reach every cluster in the objective space, and cannot collapse modes**



Supervised Static GM GAN

- **Fixed Parameters:** Mixture of Gaussians parameters remain constant during training.
- **Mean Vectors:** μ_k sampled from $U[-c, c]^d$.
- **Covariance:** $\Sigma_k = \sigma \cdot I_{d \times d}$.
- **Discriminator (Supervised):** Outputs vector $o \in R^N$, with N as class count.

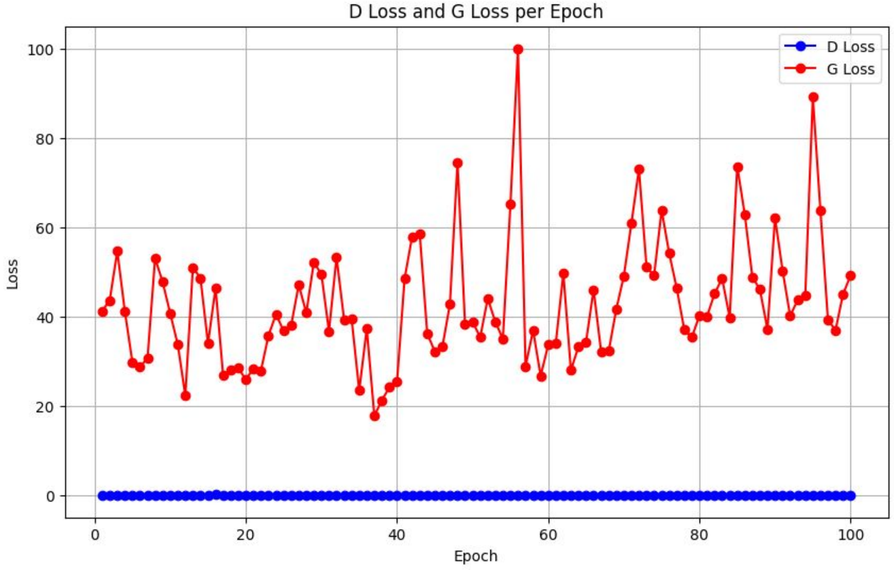
$$L(G) = -\mathop{\mathbb{E}}_{\mathbf{z} \sim p_Z(\mathbf{z})} \left[\log D(G(\mathbf{z}))_{f(y(\mathbf{z}))} + \sum_{\substack{m=1 \\ m \neq f(y(\mathbf{z}))}}^{N} \log(1 - D(G(\mathbf{z}))_m) \right]$$

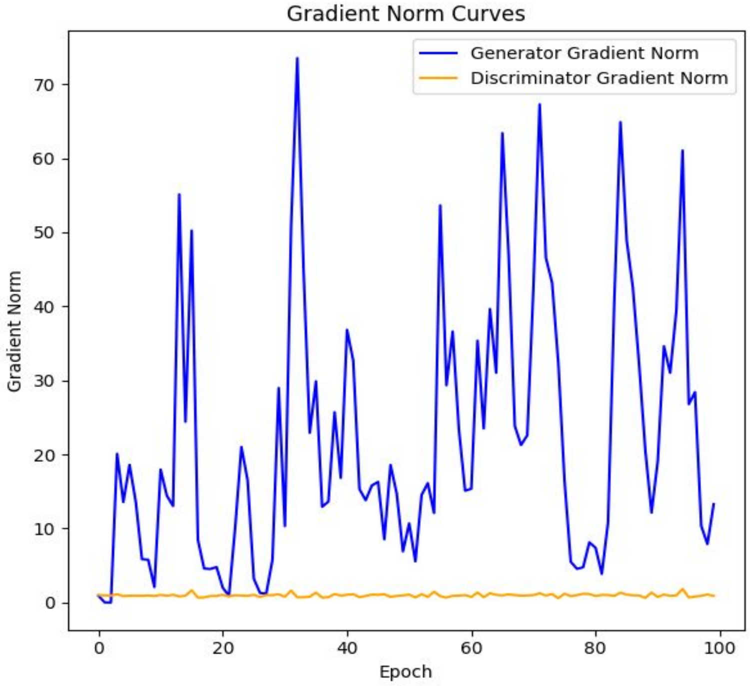
$$L(D) = -\mathop{\mathbb{E}}_{\mathbf{z} \sim p_Z(\mathbf{z})} \left[\sum_{m=1}^{N} \log(1 - D(G(\mathbf{z}))_m) \right] - \mathop{\mathbb{E}}_{\mathbf{x} \sim p_{\mathcal{X}}(\mathbf{x})} \left[\log D(\mathbf{x})_{y(\mathbf{x})} + \sum_{\substack{m=1 \\ m \neq f(y(\mathbf{x}))}}^{N} \log(1 - D(\mathbf{x})_m) \right]$$

Supervised Static GM GAN

sigma = 1.0, c = 1, K = 11, d = 200

FID Score: 258.607





Static GM GAN with WGAN-GP

Why WGAN-GP?

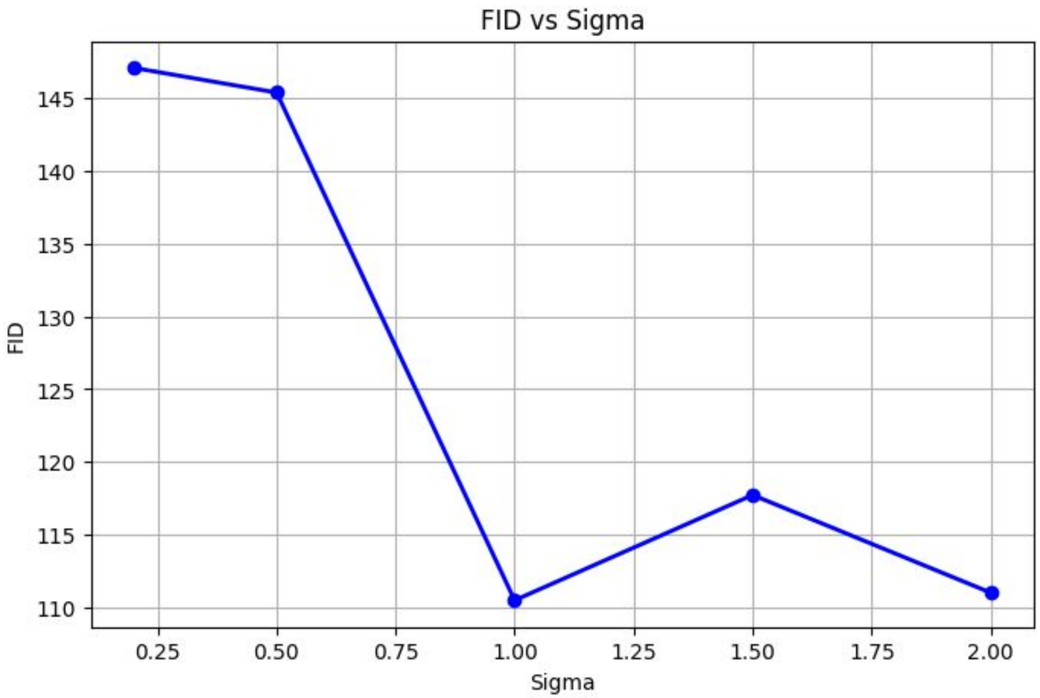
- GM-GAN struggles with mode collapse and unstable training.
- WGAN-GP mitigates these issues by enforcing a smooth discriminator response, encouraging better convergence.

Loss Function

$$\mathcal{L} = \underset{\tilde{\mathbf{x}} \sim \mathbb{P}_g}{\mathbb{E}} [f(\tilde{\mathbf{x}})] - \underset{\mathbf{x} \sim \mathbb{P}_r}{\mathbb{E}} [f(\mathbf{x})] + \lambda \underset{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{x}}}{\mathbb{E}} [(||\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}})||_2 - 1)^2]$$

Static GM GAN with WGAN-GP

Changing the values of sigma



Best Results:

sigma = 1.0,
K = 10,
d = 200

FID Score: 110.47

Dynamic GM GAN

From static to dynamic

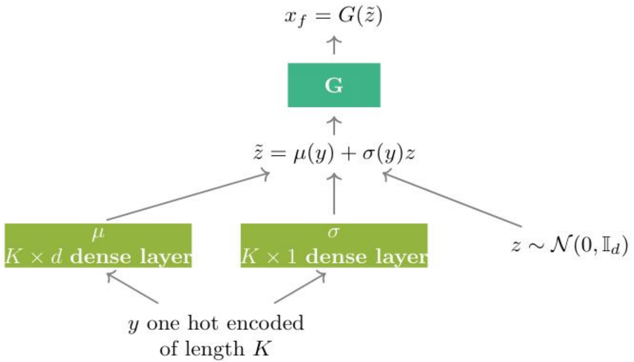
Basic Idea :

Learn the gaussian parameters !

Best Results:

sigma = 1.4,
K = 10,
d = 100

FID Score: 15,67



III. Results and Interpretation

Best model vs. Baseline (on DSLAB Platform)

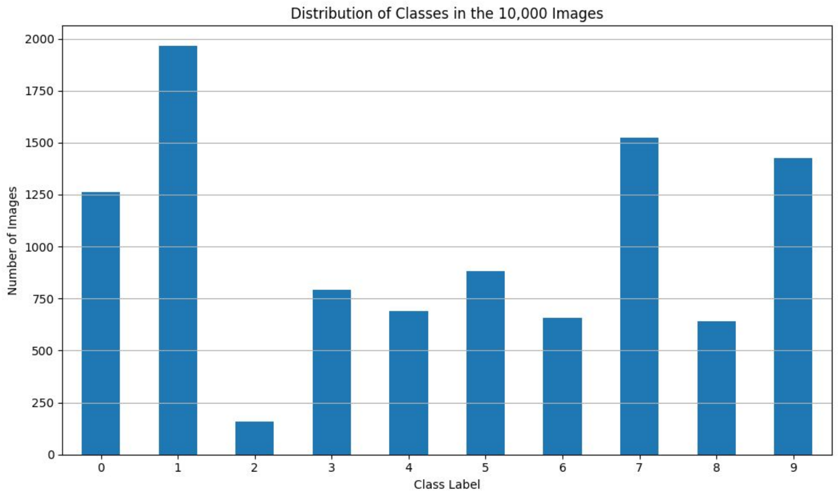
Model	FID	Precision	Recall
VanillaGAN (dim=200)	27.73	0.52	0.2
GM GAN (dynamic)	15.67	0.55	0.29

How does this improved FID translates to image generation?

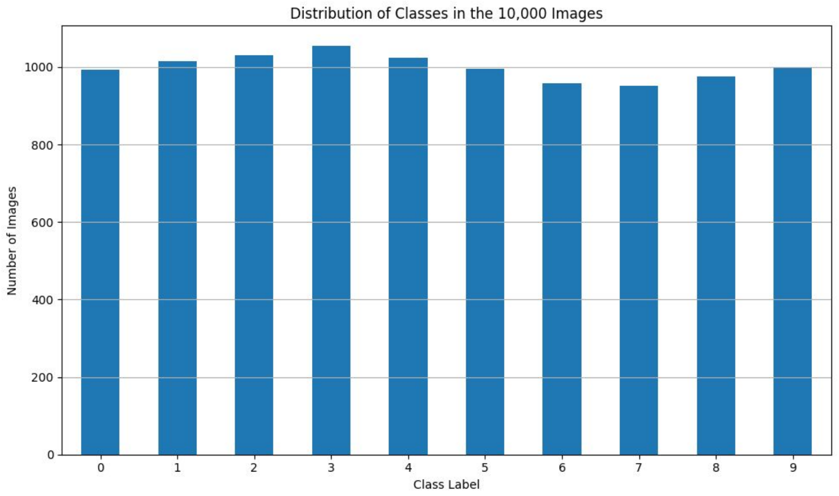
- In what aspects is it better than the baseline ?
- How can we explain it with the architecture of the GMGAN ?

Improved inter-class diversity

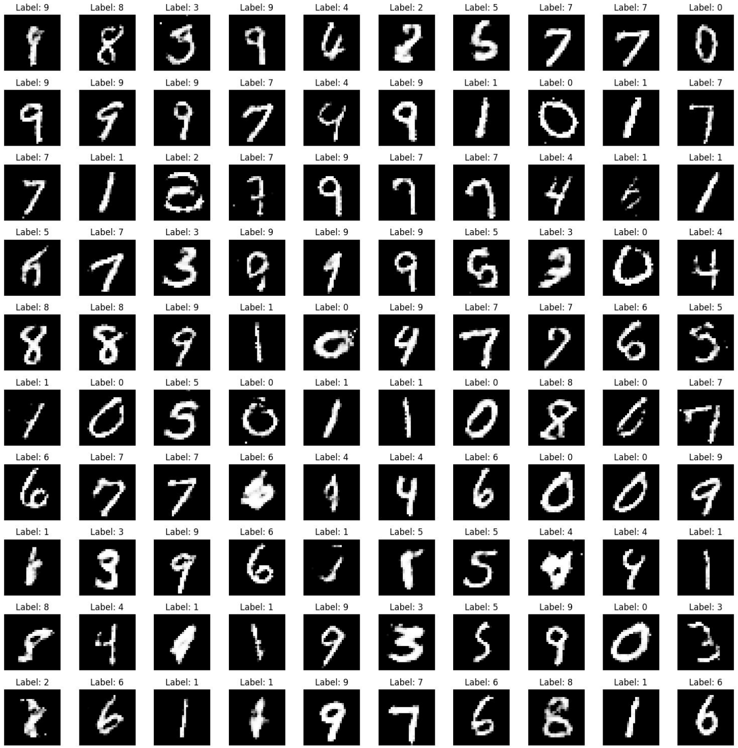
Baseline



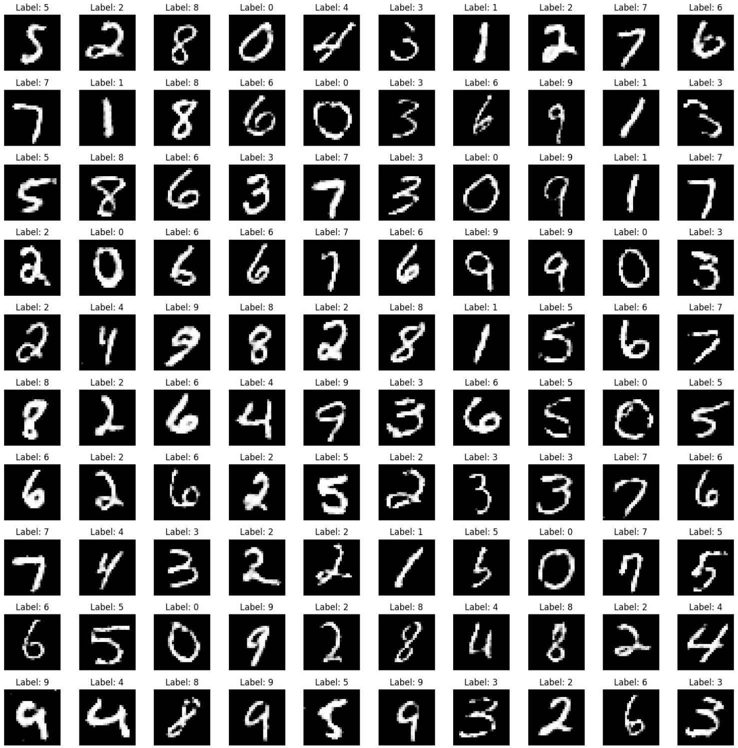
GM GAN



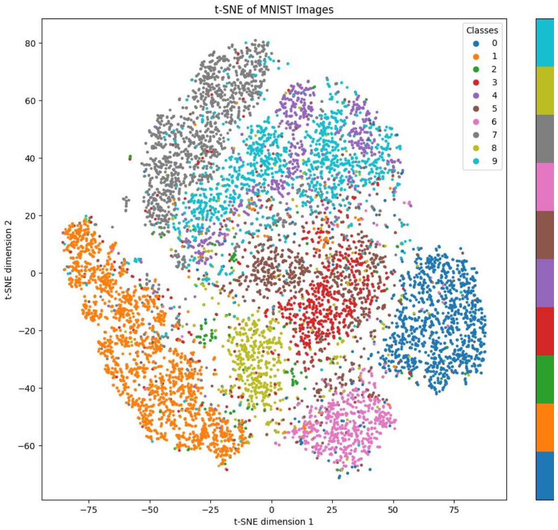
Baseline



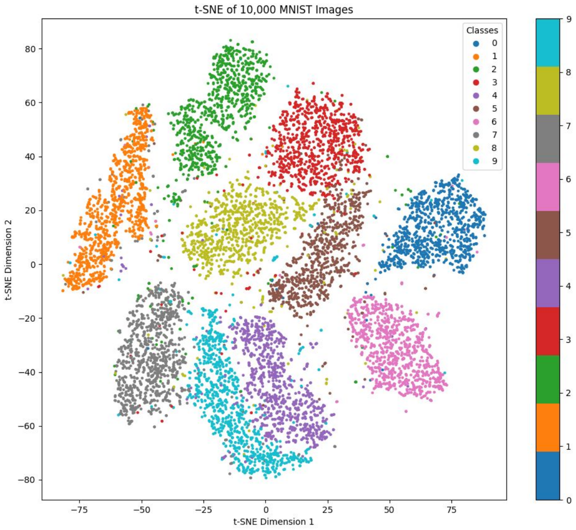
GM GAN



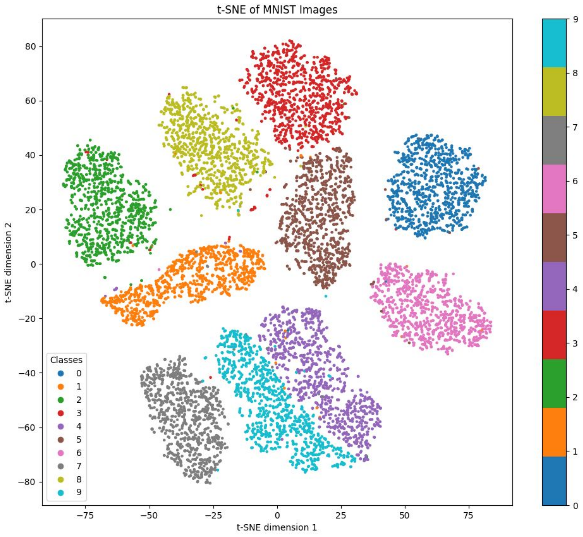
Improved separation



Baseline



Real MNIST



GM GAN

Future experimentations

- Increase the number of clusters, to try to map clusters to “sub–classes” in the real data space

- Optimize over the hyperparameters, and the initialization of means and variances, to reach higher recall / better precision

References

- **Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017).** *Improved Training of Wasserstein GANs.* arXiv preprint arXiv:1704.00028.
- **Pandeva, T., & Schubert, M. (2019).** *MMGAN: Generative Adversarial Networks for Multi-Modal Distributions*. arXiv preprint arXiv:1911.06663.
- **Benyosef, M., & Weinshall, D. (2018).** *Gaussian Mixture Generative Adversarial Networks for Diverse Datasets, and the Unsupervised Clustering of Images*. arXiv preprint arXiv:1808.10356.