Generative Adversarial Networks

Solomon HARVEY, Alexandros KOUVATSEAS, Théo COURTY
Team GANdalf
Data Science Lab
Paris Dauphine University

6 November 2024

Plan

- Objective: Training and Optimization of Generative Adversarial Networks (GANs), to be applied to data generation using MNIST Dataset
- Methods:
  - VanillaGAN
  - f-GAN (Variational divergence minimization)
  - W-GAN (Emphasis on Wasserstein Distance)

Figure: Example of a label, and a generated prediction

- f-GANs extend the traditional GAN framework, by leveraging f-divergence for training. They allow any f-divergence $\mathcal{D}_f(\mathcal{P}||\mathcal{Q})$ defined as:

$$\mathcal{D}_f(\mathcal{P}||\mathcal{Q}) = \int_{\mathcal{X}} q(x)f(\frac{p(x)}{q(x)})dx$$

  Where f must be a convex function

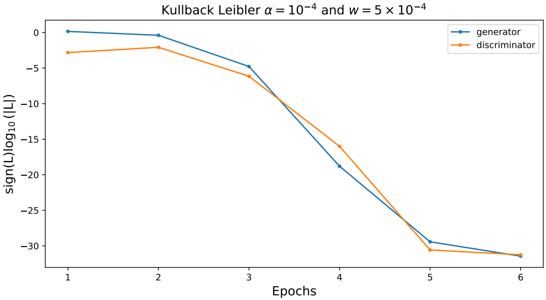
- **Training overview:** The importance of the training comes with the use of a *variational function* $T_\omega(x)$ and a generator $\mathcal{Q}_\theta$ that are trained:

$$F(\theta,\omega) = \mathbb{E}_{x\sim P}[T_\omega(x)] - \mathbb{E}_{x\sim Q_\theta}[f^*(T_\omega(x))]$$

  - f^* represents a convex conjugate of the function f
  - The variational function can be analyzed like: $T_\omega(x) = g_f(V_\omega(x))$, where g_f represents an activation function.

Optimization for f-GAN : Choice of optimizer

- Problem : Huge losses values for some f-divergence (especially KL) during the first epochs. Can even go to NaN !
- Optimizer used : Adam
  - Momentum orders $\beta_1 = 0.5$ and $\beta_2 = 0.999$
  - Learning rate $\alpha = 2 \times 10^{-4}$
  - Use of weight decay $w = 10^{-3} - 10^{-5}$ for the discriminator (form of L2 regularization)
- Gradient clipping to prevent exploding gradients
  - If $\|\nabla L\| \geq c$ then $\nabla L = c \frac{\nabla L}{\|\nabla L\|}$
  - We use $c = 1$ (arbitrary)
- These choices of optimization do not always work for $\alpha \sim 10^{-4}$.



Optimization for f-GAN : Pre-training

- Or, as jensen-shannon f-divergence results in stable convergence, we use it to pre-train the model $(4 - 6$ epochs$)$ and then switch to other f-divergence.
- It helps stabilize the convergence
- Allows for higher α $(\sim 10^{-3} - 10^{-4})$ and lower w $(\sim 10^{-5} - 10^{-6})$

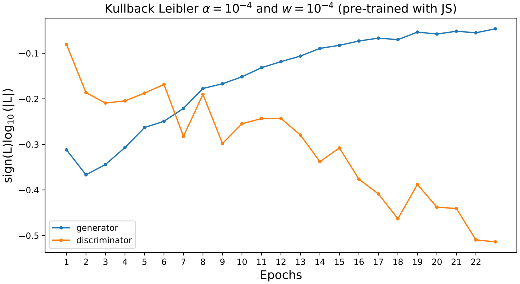


Figure: Convergence for KL f-GAN

Results for f-GAN : Pre-training

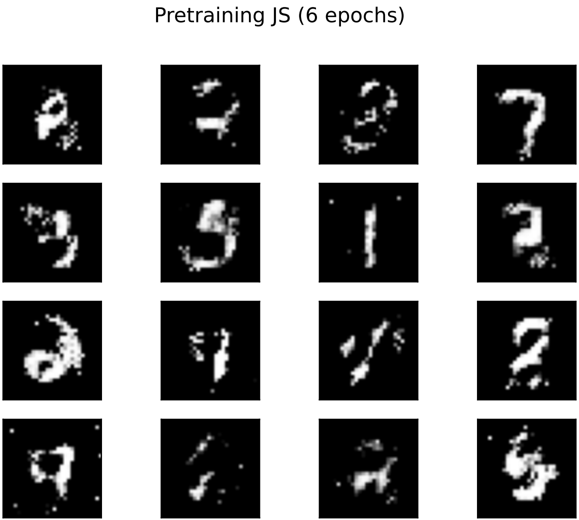


Figure: 6 epochs Jensen-Shannon f-GAN used for pre-train the other models

KL pretrained with JS

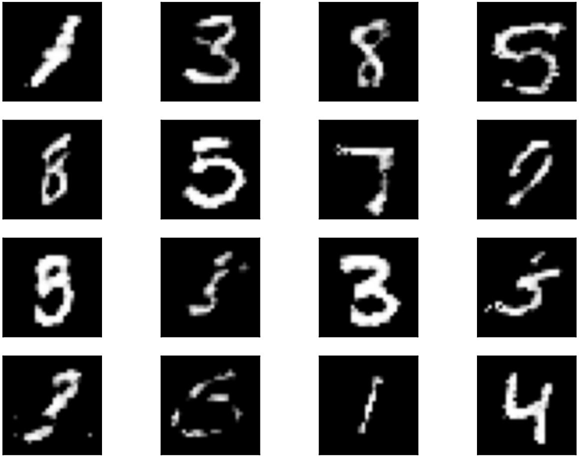


Figure: f-GAN using Kullback-Leibler f-divergence pretrained with 6 epochs of Jensen-Shannon f-GAN

Future Experimentation

- Optimizer Tuning
  - Experiment with different optimizers, such as RMSProp or SAdjust learning rates and momentum parameters for both the generator and the discriminator to observe their impact on convergence.
  - Adjust learning rates and momentum parameters for both the generator and the discriminator.
- Divergence Choice
  - Try different pre-training techniques, with different divergence functions, to stabilize the convergence and manage to get better results.
- Regularization Techniques
  - Increase the range of weight decay values (L2 regularization) and observe if it prevents large fluctuations in loss.
  - Apply gradient penalty (used in WGAN) to avoid issues with weight clipping, which could stabilize the training process.

WGAN Motivation

- Many f-divergences require the two distributions to overlap (which is not automatic, especially when considering a high-dimensional manifold), causing vanishing gradients that halt the generator's progress.
- Thus we look towards a metric that measures distance in a more broad sense, not necessarily requiring distribution overlap.

Definition (Wasserstein Distance)

Let P and Q be two probability distributions and denote by $\Pi(P, Q)$ the set of probability distributions whose marginals are P and Q. We call Wasserstein distance between P and Q the distance

$$W(P, Q) = \inf_{\gamma \in \Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|.$$

Intuitively, $W(P, Q)$ measures the minimal effort required to move mass around to change P into Q, or Q into P.

Theorem (Kantorovich-Rubinstein Duality)

For all probability distributions P and Q, we have

$$W(P, Q) = \sup_{\|f\|_L \leqslant 1} \left\{ \mathbb{E}_{x\sim P}[f(x)] - \mathbb{E}_{x\sim Q}[f(x)] \right\}$$

where $\|\cdot\|_L$ denotes the Lipschitz seminorm.

Definition (Wasserstein GAN)

- A *Wasserstein GAN* is a GAN whose goal is to solve the min-max problem

$$\inf_{\theta} \sup_{-c \leqslant w \leqslant c} \left\{ \mathbb{E}_{x \sim P_{\text{data}}}[f_w(x)] - \mathbb{E}_{z \sim Z}[f_w \circ g_\theta(z)] \right\}$$

  where c is clipping constant, simulating a Lipschtiz constant.

- By the Kantorovich-Rubinstein duality, the problem can be reformulated as first approximating up to a multiplicative constant

$$W(P_{\text{data}}, P_\theta) \approx \sup_{-c \leqslant w \leqslant c} \left\{ \mathbb{E}_{x \sim P_{\text{data}}}[f_w(x)] - \mathbb{E}_{z \sim Z}[f_w \circ g_\theta(z)] \right\}$$

  and then solving

$$\inf_{\theta} W(P_{\text{data}}, P_\theta).$$

Definition (Wasserstein GAN) (Continued)

- Hence the critic loss is equal to

$$\begin{aligned} L_{\text{crit}}(w) &= \mathbb{E}_{z \sim Z}[f_w \circ g_\theta(z)] - \mathbb{E}_{x \sim P_{\text{data}}}[f_w(x)] \\ &= \text{mean(fake data scores)} - \text{mean(real data scores)}. \end{aligned}$$

- And the generator loss is equal to

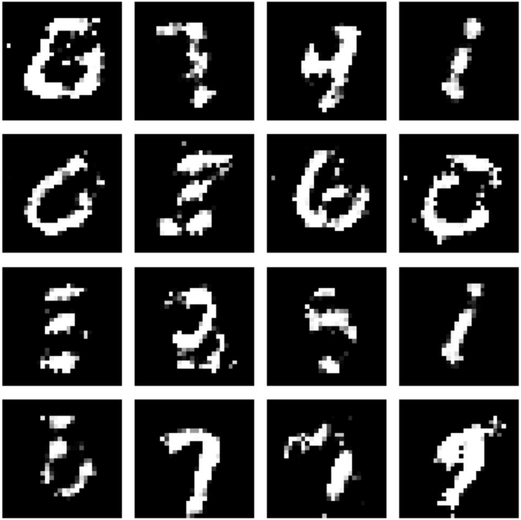
$$\begin{aligned} L_{\text{gen}}(\theta) &= -\mathbb{E}_{z \sim Z}[f_w \circ g_\theta(z)] \\ &= -\text{mean(fake data scores)}. \end{aligned}$$

WGAN Algorithm

require α (learning rate), c (clipping parameter), m (batch size), n_{critic} (number of critic iterations per generator iteration)
initialize w_0 (critic parameters), θ_0 (generator parameters)
while θ has not converged **do**
  for $0 \leqslant t \leqslant n_{\text{critic}}$ **do**
    Sample a batch $\{x_i\}_{i=1}^m \sim P_{\text{data}}$
    Sample a batch $\{z_i\}_{i=1}^m \sim Z$
    $\delta_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z_i)) - \frac{1}{m}\sum_{i=1}^m f_w(x_i)\right]$
    $w \leftarrow w - \alpha \cdot \text{RMSProp}(w, \delta_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample a batch $\{z_i\}_{i=1}^m \sim Z$
  $\delta_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z_i))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, \delta_\theta)$
end while

Preliminary WGAN Results

Preliminary WGAN Results (Continued)



Future Experimentation

- WGAN
  - Use more epochs or pretrain.
  - Use gradient penalty instead of weight clipping.
  - Use Adam optimizer for generator.
  - Modify critic architecture (not recommended by literature).

Model	VanillaGAN	f-GAN	W-GAN
FID	50.14	44.36	94.79
Precision	0.47	0.54	0.63
Recall	0.15	0.19	0.16
Time	97.38	103.84	106.05

Table: Results between the models