Generative Adversarial Networks

Dynamics

Matias Delgadino
UT Austin
July 23, 2024

Creating new people

This person does not exist! Link

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.

Creating new people

This person does not exist! ▸ Link

Starting point 200K samples of HQ headshots: CelebAHQ ▸ Link

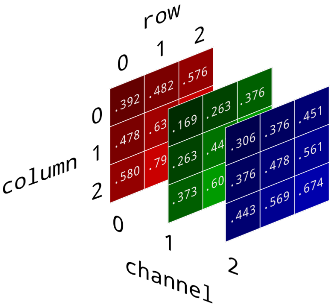
Karras, Laine, and Aila 2019.

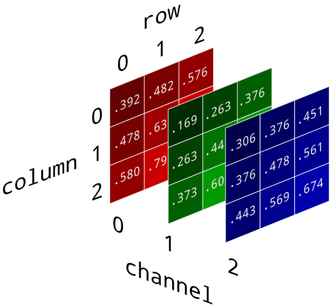
Creating new people

This person does not exist! ▸ Link

Starting point 200K samples of HQ headshots: CelebAHQ ▸ Link

Karras, Laine, and Aila 2019.


row
0
1
2
column
0
1
2
.392
.482
.576
.478
.63
.580
.79
.169
.263
.376
.263
.44
.373
.60
.306
.376
.451
.376
.478
.561
.443
.569
.674
channel
0
1
2



$$\{X_i\}_{i=1}^{200K} \subset \mathbb{R}^{1024 \times 1024 \times 3} = \mathbb{R}^{3145728}$$

Assumptions

1. We have access to infinite data samples that are independent and identically distributed:

$$\{X_i\}_{i=1}^{\infty} \text{ i.i.d. with distribution } P_* \in \mathcal{P}(\mathbb{R}^K)$$

with

$$1 \ll K$$

Assumptions

1. We have access to infinite data samples that are independent and identically distributed:

$$\{X_i\}_{i=1}^{\infty} \text{ i.i.d. with distribution } P_* \in \mathcal{P}(\mathbb{R}^K)$$

   with

$$1 \ll K$$

2. The effective dimension of the problem is more manageable

$$\text{“}L = dim(supp\, P_*)\text{”} \ll K.$$

Assumptions

1. We have access to infinite data samples that are independent and identically distributed:

$$\{X_i\}_{i=1}^{\infty} \text{ i.i.d. with distribution } P_* \in \mathcal{P}(\mathbb{R}^K)$$

with

$$1 \ll K$$

2. The effective dimension of the problem is more manageable

$$\text{“}L = dim(supp\, P_*)\text{”} \ll K.$$

e.g. This person does not exist: $K = 3145728$ and $L = 512$, Parameter size: 310Mb, 40 days of GPU compute time.

Supervised Learning problem

Given a prior distribution which is easy to sample

$$Z \sim \mathcal{N}(0,1) \in \mathcal{P}(\mathbb{R}^L),$$

Supervised Learning problem

Given a prior distribution which is easy to sample

$$Z \sim \mathcal{N}(0,1) \in \mathcal{P}(\mathbb{R}^L),$$

and a continuous function

$$g : \mathbb{R}^L \to \mathbb{R}^K,$$

easy to evaluate, which we call the Generator,

Supervised Learning problem

Given a prior distribution which is easy to sample

$$Z \sim \mathcal{N}(0,1) \in \mathcal{P}(\mathbb{R}^L),$$

and a continuous function

$$g : \mathbb{R}^L \to \mathbb{R}^K,$$

easy to evaluate, which we call the Generator,

we consider the distribution of the composition

$$g(Z) \sim g\#\mathcal{N} \in \mathcal{P}(\mathbb{R}^K)$$

Supervised Learning problem

Objective:

Find $g : \mathbb{R}^L \to \mathbb{R}^K$ easy to evaluate, such that

$$d(g\#\mathcal{N}, P_*) \quad \text{is small,}$$

for some meaningful metric d on $\mathcal{P}(\mathbb{R}^K)$.

Supervised Learning problem

Objective:

Find $g : \mathbb{R}^L \to \mathbb{R}^K$ easy to evaluate, such that

$$d(g\#\mathcal{N}, P_*) \text{ is small,}$$

for some meaningful metric d on $\mathcal{P}(\mathbb{R}^K)$.

- The eyeball metric rules them all in ML: Amazon Turk ▸ Link

Supervised Learning problem

Objective:

Find $g : \mathbb{R}^L \to \mathbb{R}^K$ easy to evaluate, such that

$$d(g\#\mathcal{N}, P_*) \text{ is small,}$$

for some meaningful metric d on $\mathcal{P}(\mathbb{R}^K)$.

- The eyeball metric rules them all in ML: Amazon Turk Link
- If we consider the family $g_\theta(z)$ of parametric function, we can minimize over θ to get a supervised learning problem.

Supervised Learning problem

Objective:

Find $g : \mathbb{R}^L \to \mathbb{R}^K$ easy to evaluate, such that

$$d(g\#\mathcal{N}, P_*) \text{ is small,}$$

for some meaningful metric d on $\mathcal{P}(\mathbb{R}^K)$.

- The eyeball metric rules them all in ML: Amazon Turk ▸ Link
- If we consider the family $g_\theta(z)$ of parametric function, we can minimize over θ to get a supervised learning problem.
- Catch: We do not have access to the distribution P_*, but only to samples.

Vanilla GAN

Information theory Relative Entropy or Kullback–Leibler divergence

$$\mathcal{H}(g\#\mathcal{N}|P_*) = \begin{cases} \int_{\mathbb{R}^K} \left(\frac{dg\#\mathcal{N}}{dP_*}\right) \log\left(\frac{dg\#\mathcal{N}}{dP_*}\right) dP_* & g\#\mathcal{N} \ll P_* \\ +\infty & g\#\mathcal{N} \not\ll P_* \end{cases}$$

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In: *Advances in neural information processing systems* 27 (2014).

Vanilla GAN

Information theory Relative Entropy or Kullback–Leibler divergence

$$\mathcal{H}(g\#\mathcal{N}|P_*) = \begin{cases} \int_{\mathbb{R}^K} \left(\frac{dg\#\mathcal{N}}{dP_*}\right) \log\left(\frac{dg\#\mathcal{N}}{dP_*}\right) dP_* & g\#\mathcal{N} \ll P_* \\ +\infty & g\#\mathcal{N} \not\ll P_* \end{cases}$$

We need a way to evaluate it using samples.

Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, and Bengio 2014.

Duality

Legendre-Fenchel Transform:

$$\mathcal{H}(g\#\mathcal{N}|P_*) = \sup_{f\in C_b(\mathbb{R}^K)} \int_{\mathbb{R}^L} f(g(z))d\mathcal{N}(z) - \log \int_{\mathbb{R}^L} e^{g(x)} dP_*(x),$$

where

$$f : \mathbb{R}^K \to \mathbb{R}$$

is called the Discriminator.

Sampling

Advantage: For fixed Discriminator $f \in C_b(\mathbb{R}^K)$, we can sample the integrals:

Sampling

Advantage: For fixed Discriminator $f \in C_b(\mathbb{R}^K)$, we can sample the integrals:

Given $m \in \mathbb{N}$ a batch size and $Z_1, ..., Z_m$ i.i.d. with distribution $\mathcal{N}$ and $X_1, ..., X_m$ i.i.d. with distribution P_*

$$\int_{\mathbb{R}^L} f(g(z)) d\mathcal{N}(z) - \log \int_{\mathbb{R}^L} e^{f(x)} dP_*(x)$$

$$\sim$$

$$\frac{1}{m} \sum_{i=1}^{m} f(g(Z_i)) - \log \frac{1}{m} \sum_{i=1}^{m} e^{f(X_i)}$$

For simplicity, we take the batch size $m = 1$ from now on, which is an estimator in expectation.

Degeneracy

If $g\#\mathcal{N} \not\ll P_*$, then $\mathcal{H}(g\#\mathcal{N}|P_*) = \infty$

Degeneracy

If $g\#\mathcal{N} \not\ll P_*$, then $\mathcal{H}(g\#\mathcal{N}|P_*) = \infty$

We will learn nothing if the distributions are not aligned from the start!

1-Wasserstein distance

Alternative, the 1-Wasserstein distance with Kantorovich's duality

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

1-Wasserstein distance

Alternative, the 1-Wasserstein distance with Kantorovich's duality

$$d_1(g\#\mathcal{N}, P_*) = \mathbb{E}_{(X,Z)\sim\pi}[\|X - g(Z)\|]$$

Arjovsky, Chintala, and Bottou 2017.

1-Wasserstein distance

Alternative, the 1-Wasserstein distance with Kantorovich's duality

$$d_1(g\#\mathcal{N}, P_*) = \mathbb{E}_{(X,Z)\sim\pi}[\|X - g(Z)\|]$$

$$= \sup_{f:\|f\|_{lip}\leq 1} \int_{\mathbb{R}^L} f(g(z))d\mathcal{N}(z) - \int_{\mathbb{R}^K} f(x)\, dP_*(x).$$

Arjovsky, Chintala, and Bottou 2017.

1-Wasserstein distance

Alternative, the 1-Wasserstein distance with Kantorovich's duality

$$d_1(g\#\mathcal{N}, P_*) = \mathbb{E}_{(X,Z)\sim\pi}[|X - g(Z)|]$$

$$= \sup_{f:\|f\|_{lip}\le 1} \int_{\mathbb{R}^L} f(g(z))d\mathcal{N}(z) - \int_{\mathbb{R}^K} f(x)\, dP_*(x).$$

$$= \sup_{f:\|f\|_{lip}\le 1} \mathbb{E}f(g(Z)) - \mathbb{E}f(X).$$

Arjovsky, Chintala, and Bottou 2017.

1-Wasserstein distance

Alternative, the 1-Wasserstein distance with Kantorovich's duality

$$d_1(g\#\mathcal{N}, P_*) = \mathbb{E}_{(X,Z)\sim\pi}[|X - g(Z)|]$$

$$= \sup_{f:\|f\|_{lip}\leq 1} \int_{\mathbb{R}^L} f(g(z))d\mathcal{N}(z) - \int_{\mathbb{R}^K} f(x)\, dP_*(x).$$

$$= \sup_{f:\|f\|_{lip}\leq 1} \mathbb{E}f(g(Z)) - \mathbb{E}f(X).$$

The main advantage is that this distance does not degenerate.

Arjovsky, Chintala, and Bottou 2017.

Neural Networks

Introduce, the simplest setting 1 hidden layer Neural Networks:

$$g_\Theta(z) = \frac{1}{N}\sum_{i=1}^{N}\sigma(z;\theta_i) \qquad f_\Omega(x) = \frac{1}{M}\sum_{j=1}^{M}\sigma(x;\omega_i)$$

with $\Theta = (\theta_1, ...\theta_N)$ and $\Omega = (\omega_1, ..., \omega_M)$.

A typical smooth example is the sigmoid

$$\sigma(z;\theta_i) = \begin{pmatrix} \frac{a_i^1}{1+e^{-(b_i^1\cdot z+c_i^1)}} \\ \dots \\ \frac{a_i^K}{1+e^{-(b_i^K\cdot z+c_i^K)}} \end{pmatrix} \in \mathbb{R}^K$$

$$\theta_i = ((a_i^1, b_i^1, c_i^1), \dots, (a_i^K, b_i^K, c_i^K)) \in (\mathbb{R} \times \mathbb{R}^L \times \mathbb{R})^K$$

$$\sigma(x;\omega_j) = \frac{\alpha_j}{1 + e^{-(\beta_j\cdot x+\gamma_j)}} \in \mathbb{R}$$

$$\omega_j = (\alpha_j, \beta_j, \gamma_j) \in \mathbb{R} \times \mathbb{R}^K \times \mathbb{R}$$

Exchangeability

The relative order of the parameters does not affect the output function.

Exchangeability

The relative order of the parameters does not affect the output function.

Without loss of information we can encode

$$(\theta_1, \dots, \theta_N) \to \mu_N = \frac{1}{N} \sum_{i=1}^{N} \delta_{\theta_i} \in \mathcal{P}\left((\mathbb{R} \times \mathbb{R}^L \times \mathbb{R})^K \right)$$

and

$$(\omega_1, \dots, \omega_N) \to \nu_M = \frac{1}{M} \sum_{i=1}^{M} \delta_{\omega_i} \in \mathcal{P}\left(\mathbb{R} \times \mathbb{R}^K \times \mathbb{R} \right).$$

Algorithm

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**

  for $t = 0, \ldots, n_{\text{critic}}$ **do**

    Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.

    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.

    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$

    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$

    $w \leftarrow \text{clip}(w, -c, c)$

  end for

  Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.

  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$

  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$

end while

Arjovsky, Chintala, and Bottou 2017.

Important parameters

- Learning rate $\alpha = 0.00005$, we consider $\Delta t = \alpha / N$ the fictitious time discretization.

Important parameters

- Learning rate $\alpha = 0.00005$, we consider $\Delta t = \alpha / N$ the fictitious time discretization.
- $n_c = 5$ the critics advantage, we consider $\gamma_c = n_c \frac{N}{M}$ the time re-scaling parameter.

Important parameters

- Learning rate $\alpha = 0.00005$, we consider $\Delta t = \alpha / N$ the fictitious time discretization.
- $n_c = 5$ the critics advantage, we consider $\gamma_c = n_c \frac{N}{M}$ the time re-scaling parameter.
- $c = 0.01$ the clipping parameter that imposes $\|\omega_i\|_\infty \leq c$ to satisfy a uniform Lipschitz bound.

Important parameters

- Learning rate $\alpha = 0.00005$, we consider $\Delta t = \alpha/N$ the fictitious time discretization.
- $n_c = 5$ the critics advantage, we consider $\gamma_c = n_c \frac{N}{M}$ the time re-scaling parameter.
- $c = 0.01$ the clipping parameter that imposes $\|\omega_i\|_\infty \leq c$ to satisfy a uniform Lipschitz bound.
- RMSProp is a version of SGD that normalizes the gradient sizes componentwise to escape plateaus. For some $\beta \in [0, 1]$:

$$
\begin{aligned}
M_k^i &= (1-\beta) M_{k-1}^i + \beta |\partial_{\theta_i} E(\Theta_k)|^2 \\
\theta_{k+1}^i &= \theta_{k+1}^i - \alpha \frac{\partial_{\theta_i} E(\Theta_k)}{\sqrt{M_k^i}}
\end{aligned}
$$

Supervised learning

Supervised learning:

$$\min_{\Theta} E[\Theta] = \min_{\Theta} \int |g_{\Theta}(x) - g_*(x)|^2 \, dP_*(x) = \min_{\Theta} \int e(\Theta, x) \, dP_*(x)$$

Supervised learning

Supervised learning:

$$\min_{\Theta} E[\Theta] = \min_{\Theta} \int |g_\Theta(x) - g_*(x)|^2 \, dP_*(x) = \min_{\Theta} \int e(\Theta, x) \, dP_*(x)$$

Algorithm:
While Θ has not converged:
  Sample $X_k \sim P_*$

$$\Theta_{k+1} = \Theta_k - \alpha \partial_\Theta e[\Theta_k, X_k]$$

Supervised learning

Supervised learning:

$$\min_{\Theta} E[\Theta] = \min_{\Theta} \int |g_{\Theta}(x) - g_*(x)|^2 \, dP_*(x) = \min_{\Theta} \int e(\Theta, x) \, dP_*(x)$$

Algorithm:

While Θ has not converged:

Sample $X_k \sim P_*$

$$\Theta_{k+1} = \Theta_k - \alpha \partial_{\Theta} e[\Theta_k, X_k]$$

SGD is a stochastic discretization of

$$\dot{\Theta} = -\nabla_{\Theta} E[\Theta].$$

SGD as a Stochastic discretization

Using, exchangeability

$$g_{\Theta}(x) = \frac{1}{N}\sum_{i=1}^{N} \sigma(x, \theta_i) = \langle g(x, \cdot), \mu_N \rangle$$

SGD as a Stochastic discretization

Using, exchangeability

$$g_\Theta(x) = \frac{1}{N}\sum_{i=1}^{N}\sigma(x,\theta_i) = \langle g(x,\cdot),\mu_N\rangle$$

we notice

$$\partial_{\theta_i}E[\theta] = \frac{2}{N}\int (g_\Theta(x) - g_*(x))\partial_2\sigma(x,\theta_i)dP_*(x) = \frac{1}{N}V[\mu_N](\theta_i)$$

SGD as a Stochastic discretization

Using, exchangeability

$$g_\Theta(x) = \frac{1}{N}\sum_{i=1}^{N} \sigma(x, \theta_i) = \langle g(x,\cdot), \mu_N \rangle$$

we notice

$$\partial_{\theta_i} E[\theta] = \frac{2}{N}\int (g_\Theta(x) - g_*(x))\partial_2\sigma(x,\theta_i) dP_*(x) = \frac{1}{N} V[\mu_N](\theta_i)$$

Namely $\dot{\Theta}(t) = -\nabla E[\Theta(t)]$, if and only if,

$$\begin{cases} \partial_t \mu_N(t) + \frac{1}{N}\nabla \cdot (\mu_N(t) V[\mu_N(t)]) = 0 \\ \mu_N(0) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_{i,in}} \end{cases}$$

Convergence of the dynamics

Theorem (Law of Large Numbers)

Assume $\{\theta_{i,in}\}$ i.i.d. sampled from μ_{in}. Then, μ_N converges to a deterministic process which concentrates in the unique solution to

$$\begin{cases} \partial_t \mu(t) + \nabla \cdot (\mu(t) V[\mu(t)]) = 0 \\ \mu(0) = \mu_{in} \end{cases} \tag{SGD}$$

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In: *Advances in neural information processing systems* 31 (2018); Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. In: *Proceedings of the National Academy of Sciences* 115.33 (2018), E7665–E7671; Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. In: *SIAM Journal on Applied Mathematics* 80.2 (2020), pp. 725–752; Grant Rotskoff and Eric Vanden-Eijnden. Trainability and accuracy of artificial neural networks: An interacting particle system approach. In: *Communications on Pure and Applied Mathematics* 75.9 (2022), pp. 1889–1935.

Gradient flow interpreation

Considering the energy $\mathsf{E} : \mathcal{P} \to \mathbb{R}$, given by

$$\mathsf{E}[\mu] = \frac{1}{2} \int |g_\mu(x) - g_*(x)|^2 \, dP_*(x)$$

we have that (SGD) is the 2-Wasserstein gradient flow of E.

Aggregation Equation

Moreover, expanding the square we obtain the aggregation equation:

$$\mathsf{E}[\mu] = \frac{1}{2}\int W(\theta_1, \theta_2)d\mu(\theta_1)d\mu(\theta_2) + \int V(\theta)d\mu(\theta) + C,$$

Aggregation Equation

Moreover, expanding the square we obtain the aggregation equation:

$$\mathsf{E}[\mu] = \frac{1}{2} \int W(\theta_1, \theta_2) d\mu(\theta_1) d\mu(\theta_2) + \int V(\theta) d\mu(\theta) + C,$$

where

$$W(\theta_1, \theta_2) = \int \sigma(x; \theta_1) \sigma(x; \theta_2) dP_*(x)$$

and

$$V(\theta) = -\int g_*(x) \sigma(x; \theta) \; dP_*(x).$$

W-GAN as a discretization

Replacing RMSProp by SGD, we have the algorithm

$$\begin{cases} \theta_i^{k+1} = \theta_i^k + \Delta t \; v_\theta[\mu_N, \nu_M](\theta_i; (X_k, Z_k)) \\ \omega_j^{k+1} = Proy_Q(\omega_j^k + \gamma_c \Delta t \; v_\omega[\mu_N, \nu_M](\omega_j^k; (X_k, Z_k))), \end{cases}$$

W-GAN as a discretization

Replacing RMSProp by SGD, we have the algorithm

$$\begin{cases} \theta_i^{k+1} = \theta_i^k + \Delta t \; v_\theta[\mu_N, \nu_M](\theta_i; (X_k, Z_k)) \\ \omega_j^{k+1} = Proy_Q(\omega_j^k + \gamma_c \Delta t \; v_\omega[\mu_N, \nu_M](\omega_j^k; (X_k, Z_k))), \end{cases}$$

where

$$Q = [-c, c]^{1+L+1}, \qquad \gamma_c = n_c \frac{N}{M}$$

and $\{X_k\}_{k=0}^{\infty}$ and $\{Z_k\}$ i.i.d sampled from P_* and $\mathcal{N}$ respectively.

WGAN as a PDE

The associated PDE is given by

$$\begin{cases} \partial_t \mu - \nabla \cdot (\partial_\mu \Psi[\mu, \nu] \mu) = 0 \\ \partial_t \nu + \gamma_c \nabla \cdot (\mathbf{Proj}_{\Pi_Q} \partial_\nu \Psi[\mu, \nu] \nu) = 0 \end{cases} \tag{WGAN-PDE}$$

where

$$\Psi[\mu, \nu] = \int_{\mathbb{R}^L} f_\nu(g_\mu(z))\, d\mathcal{N}(z) - \int_{\mathbb{R}^K} f_\nu(x)\, dP_*(x)$$

WGAN as a PDE

The associated PDE is given by

$$\begin{cases} \partial_t \mu - \nabla \cdot (\partial_\mu \Psi[\mu, \nu]\mu) = 0 \\ \partial_t \nu + \gamma_c \nabla \cdot (\mathbf{Proj}_{\Pi_Q} \partial_\nu \Psi[\mu, \nu]\nu) = 0 \end{cases} \tag{WGAN-PDE}$$

where

$$\Psi[\mu, \nu] = \int_{\mathbb{R}^L} f_\nu(g_\mu(z))\, d\mathcal{N}(z) - \int_{\mathbb{R}^K} f_\nu(x)\, dP_*(x)$$

Notice that $\mathbf{Proj}_{\Pi_Q} : Q \times \mathbb{R}^K \to \mathbb{R}^K$ is a discontinuous operator on ∂_Q.

Well Posedness and Coagulation at the Boundary

Proposition (jww R. Cabrera & B. Suassuna)

If the activation function is smooth, then (WGAN-PDE) has a unique stable solution:

$$
\begin{aligned}
&d_2(\mu_1(t), \mu_2(t)) + d_2(\nu_1(t), \nu_2(t)) \\
&\leq C(d_4(\mu_{1,in}, \mu_{2,in}) + d_2(\nu_{1,in}, \nu_{2,in}))
\end{aligned}
$$

for any $t \in [0, T]$.

Well Posedness and Coagulation at the Boundary

Proposition (jww R. Cabrera & B. Suassuna)

If the activation function is smooth, then (WGAN-PDE) has a unique stable solution:

$$d_2(\mu_1(t), \mu_2(t)) + d_2(\nu_1(t), \nu_2(t))$$
$$\leq C(d_4(\mu_{1,in}, \mu_{2,in}) + d_2(\nu_{1,in}, \nu_{2,in}))$$

for any $t \in [0, T]$.

Observation: If the support of ν hits ∂Q it will flatten, and can never fatten back up.

Well Posedness and Coagulation at the Boundary

Proposition (jww R. Cabrera & B. Suassuna)

If the activation function is smooth, then (WGAN-PDE) has a unique stable solution:

$$d_2(\mu_1(t), \mu_2(t)) + d_2(\nu_1(t), \nu_2(t))$$
$$\leq C(d_4(\mu_{1,in}, \mu_{2,in}) + d_2(\nu_{1,in}, \nu_{2,in}))$$

for any $t \in [0, T]$.

Observation: If the support of ν hits ∂Q it will flatten, and can never fatten back up.

In particular, the support it can coagulate to a single point in finite time t_0, and $\nu(t) = \delta_{\omega(t)}$ for any $t > t_0$.

Quantified convergence

Theorem (jww R. Cabrera & B. Suassuna)

Consider $(\mu_N(t), \nu_N(t))$ *the time interpolation of the empirical measures* $\{(\mu_N^k, \nu_N^k)\}_{k=1}^{\infty}$ *given by the WGAN algorithm, then for any fixed time interval* $t \in [0, T]$

$$\mathbb{E} d_2^2((\mu_N(t), \nu_N(t)), (\mu_\infty(t), \nu_\infty(t))) \leq \frac{C}{N}$$

where (μ_∞, ν_∞) *is the unique solution to* (WGAN-PDE) *with initial condition* $\mu_{in} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i}$, $\nu_{in} = \frac{1}{M}\sum_{j=1}^{M} \delta_{\omega_j}$.

Quantified convergence

Corollary (jww R. Cabrera & B. Suassuna)

If $\{\theta_i\}_{i=1}^N$, $\{\omega_j\}_{j=1}^M$ i.i.d. sampled from $\overline{\mu}_{in}$ and $\overline{\nu}_{in}$ respectively, then

$$\mathbb{E}d_2^2((\mu_N(t),\nu_N(t)),(\overline{\mu}_\infty(t),\overline{\nu}_\infty(t))) \leq \frac{C}{N^{\frac{1}{K(2+L)}}}$$

Quantified convergence

Corollary (jww R. Cabrera & B. Suassuna)

If $\{\theta_i\}_{i=1}^N$, $\{\omega_j\}_{j=1}^M$ i.i.d. sampled from $\overline{\mu}_{in}$ and $\overline{\nu}_{in}$ respectively, then

$$\mathbb{E} d_2^2((\mu_N(t), \nu_N(t)), (\overline{\mu}_\infty(t), \overline{\nu}_\infty(t))) \leq \frac{C}{N^{\frac{1}{K(2+L)}}}$$

Remark: The Wasserstein distance suffers from the curse of dimensionality, when we approximate by samples.

Proof

Compare SGD

$$\theta^{k+1} = \theta^k + \Delta t v(\theta^k, X_k)$$

with (Projected) Forward Euler

$$\tilde{\theta}^{k+1} = \tilde{\theta}^k + \Delta t V(\tilde{\theta}^k)$$

Proof

Compare SGD

$$\theta^{k+1} = \theta^k + \Delta t v(\theta^k, X_k)$$

with (Projected) Forward Euler

$$\tilde{\theta}^{k+1} = \tilde{\theta}^k + \Delta t V(\tilde{\theta}^k)$$

$$e_{k+1} = \theta^{k+1} - \tilde{\theta}^{k+1} \leq (1 + \Delta t |V|_{lip}) e_k + \Delta t M_k,$$

with

$$M_k = v(\theta^k, X_k) - V(\theta^k)$$

Proof

Compare SGD

$$\theta^{k+1} = \theta^k + \Delta t v(\theta^k, X_k)$$

with (Projected) Forward Euler

$$\tilde{\theta}^{k+1} = \tilde{\theta}^k + \Delta t V(\tilde{\theta}^k)$$

$$e_{k+1} = \theta^{k+1} - \tilde{\theta}^{k+1} \leq (1 + \Delta t|V|_{lip})e_k + \Delta t M_k,$$

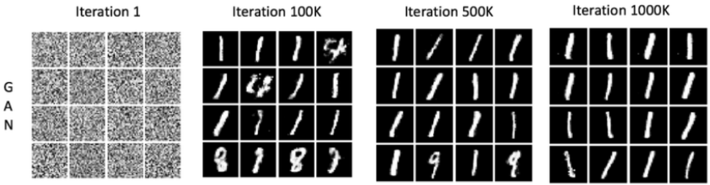
with

$$M_k = v(\theta^k, X_k) - V(\theta^k)$$

Gromwall' inequality, we have

$$\mathbb{E}[|e_k|^2] \leq (\Delta t)^2 \mathbb{E}\left|\sum_{r=0}^{k}(1 + \Delta t|V|_{lip})^{k-r} M_r\right|^2 \leq C\Delta t.$$

Mode Collapse



Mode Collapse

Chat-GPT loves to delve:

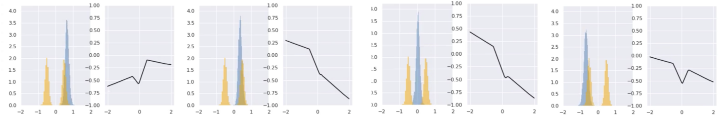
Abstract

Generative Adversarial Networks (GANs) was one of the first Machine Learning algorithms to be able to generate remarkably realistic synthetic images. In this presentation, we **DELVE** into the mechanics of the GAN algorithm and its profound relationship with optimal transport theory. Through a detailed exploration, we illuminate how GAN approximates a system of PDE, particularly evident in shallow network architectures. Furthermore, we investigate the phenomenon of mode collapse, a well-known pathological behavior in GANs, and elucidate its connection to the underlying PDE framework through an illustrative example.

Failure to converge

Example: $K = 1$, $L = 1$

$$P_* = \frac{1}{2}\mathcal{N}(0,-1) + \frac{1}{2}\mathcal{N}(0,1)$$



Video

Toy Example

$K = 1$, $L = 1$, $P_* = \frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1$ and activation functions

$$g(z;\theta) = \begin{cases} -1 & \text{if } z < \theta \\ 1 & \text{if } z > \theta \end{cases} \qquad f(x,\omega) = (\omega x)_+.$$

Toy Example

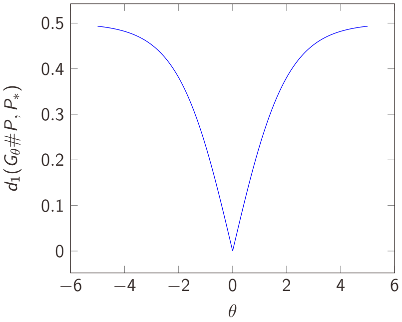
$K = 1$, $L = 1$, $P_* = \frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1$ and activation functions

$$g(z;\theta) = \begin{cases} -1 & \text{if } z < \theta \\ 1 & \text{if } z > \theta \end{cases} \qquad f(x,\omega) = (\omega x)_+.$$

$$g_\theta \# \mathcal{N} = \Phi(\theta)\delta_{-1} + (1 - \Phi(\theta))\delta_1$$

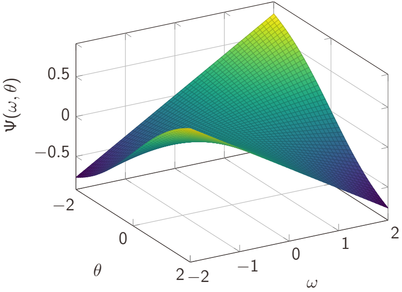
Graphs

$$d_1(g_\theta \# \mathcal{N}, P_*) = \max_{\omega \in [-1,1]} \int f_\omega(g_\theta(z)) d\mathcal{N}(z) - \int f_\omega(x) dP_*(x)$$



Graphs

$$\begin{aligned}\Psi(\omega, \theta) &= \int_{\mathbb{R}} D_\omega(g_\theta(z))\, dP(z) - \int_{\mathbb{R}} D_\omega(x)\, dP_*(x) \\ &= \left(\frac{1}{2} - \Phi(\theta)\right)\omega.\end{aligned}$$



Toy Example:ODE dynamics

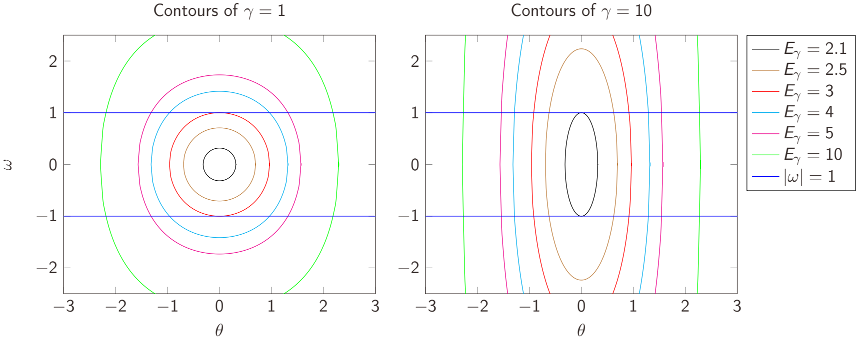
Gradient descent/ascent gives rise to periodic orbits. If we consider

$$E_\gamma[\theta, \omega] = \cosh(\theta) + \frac{1}{\gamma}|\omega|^2,$$

then for all $t > 0$

$$E_\gamma[\theta(t), \omega(t)] = E_\gamma[\theta_{in}, \omega_{in}]$$

Periodic Orbits



Questions?

Thank you!