A Neural Tangent Kernel Perspective of GANs

Jean-Yves Franceschi, Emmanuel de Bézenac, Ibrahim Ayed, Mickaël Chen, Sylvain Lamprier, Patrick Gallinari

A Neural Tangent Kernel Perspective of GANs

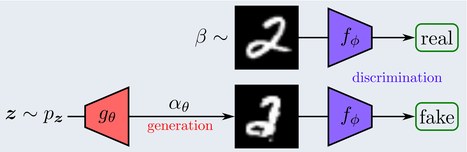
J.-Y. Franceschi,*,[1,2] *E. de Bézenac*,*,[3,2] I. Ayed,*,[2,4] M. Chen,[5] S. Lamprier,[2] P. Gallinari[2,1]

[1]Criteo AI Lab [2]Sorbonne Université, CNRS, ISIR [3]SAM, D-MATH, ETH Zürich [4]ThereSIS Lab, Thales [5]Valeo.ai

Motivation & Outline

- Many analyses cannot explain GAN training as they fail to take into account alternating optimization and model the architecture and implicit biases of the discriminator.
- We propose a theoretical framework solving these issues using the theory of Neural Tangent Kernels.
- We deduce new insights about the flow and convergence of the generated distribution during training.

Generative Adversarial Networks



$$\inf_\theta \sup_\phi \mathcal{L}(g_\theta, f_\phi) = \inf_\theta \mathcal{L}\Big(g_\theta, f_{\phi^*_\theta}\Big) \approx \inf_\theta \mathscr{C}(\alpha_\theta, \beta),$$

$$\mathcal{L}(g_\theta, f_\phi) = \mathbb{E}_{x\sim\alpha_\theta}\Big[(a \circ f_\phi)(x)\Big] - \mathbb{E}_{y\sim\beta}\Big[(b \circ f_\phi)(y)\Big].$$

E.g. for vanilla GAN $\mathscr{C}$ is a Jensen-Shannon, for WGAN it is the 1-Wasserstein, for LSGAN it is a χ^2-divergence.

Issue 1: Alternating Optimization

- In practice, GANs are optimized alternatingly, f_ϕ and g_θ being considered to be independent of each other.

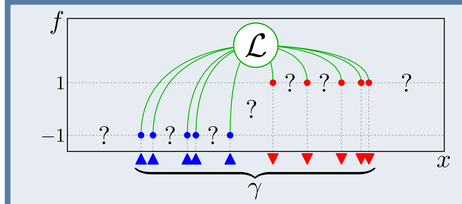
$$\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}(g_\theta, f_\phi);$$
$$\phi \leftarrow \phi + \lambda\nabla_\phi\mathcal{L}(g_\theta, f_\phi).$$

- Gradient received by g_θ:

$$\cancel{\nabla_\theta\mathcal{L}\Big(g_\theta, f_{\phi^*_\theta}\Big)} \quad \Rightarrow \quad \nabla_\theta\mathcal{L}(g_\theta, f_\phi).$$

- This changes the true generator loss $\mathscr{C}$. Hence, sound studies should model alternating optimization.

Issue 2: Ill-Defined Discr. Gradients



In an alternating optimization setting:

$$\nabla_\theta\mathcal{L}(g_\theta, f) = \mathbb{E}_{z\sim p_z}\Big[\nabla_\theta g_\theta(z)^\top \nabla_x(a \circ f)(x)\big|_{x=g_\theta(z)}\Big].$$

- The gradient of the generator requires ∇f (chain rule).
- Without any assumption on the structure of f, as $\mathcal{L}$ is only defined on training points γ, ∇f is not defined.
- The parameter gradient of the generator is thus also ill-defined.
- Analyses need to take into account the structure of f.

Our solution: Modeling discriminator training as a neural network via its NTK.

A Background on NTKs

The Neural Tangent Kernel: For a neural network f_ϕ with parameters ϕ, its NTK k_{f_ϕ} is defined as:

$$k_{f_\phi}(x, y) \triangleq \big\langle \partial_\phi f_\phi(x), \partial_\phi f_\phi(y) \big\rangle.$$

In the infinite-width limit of f, during training:

$$k_{f_\phi}(x, y) = k(x, y).$$

Kernel Integral Operator and RKHS:

$$\mathcal{T}_{k,\gamma}\colon L^2(\gamma) \to \mathcal{H}^\gamma_k,\ h \mapsto \int_x k(\cdot, x)h(x)\,\mathrm{d}\gamma(x),$$

where $\mathcal{H}^\gamma_k \subseteq L^2(\Omega)$ is the RKHS of k generated by γ.

Discriminator Inner Loop

We consider the NNs in the NTK regime. This enables a theoretical study of their evolution w.r.t. training time t:

$$\partial_t f_t = \mathcal{T}_{k,\gamma}\big(\nabla^\gamma\mathcal{L}_\alpha(f_t)\big).$$

Discriminator Structure:

Under mild assumptions, f_t is uniquely defined and:

$$\forall t \in \mathbb{R}_+, f_t = f_0 + \mathcal{T}_{k,\gamma}\left(\int_0^t \nabla^\gamma\mathcal{L}_\alpha(f_s)\,\mathrm{d}s\right) \in f_0 + \mathcal{H}^\gamma_k.$$

- $\mathcal{T}_{k,\gamma}$ *smooths* out gradients over the whole input space by sending them into $\mathcal{H}^\gamma_k$.
- $\mathcal{H}^\gamma_k$ depends on discriminator architecture.

Differentiability of the Discriminator:

The discriminator trained with gradient descent is infinitely differentiable (almost) everywhere.

- The spatial gradient of the discriminator ∇f_t is well-defined.
- The parameter gradient of the generator $\nabla_\theta\mathcal{L}(g_\theta, f_t)$ is well-defined.

Underlying NTK Regularity Results

To prove the above results, we establish novel regularity results on NTKs. Given, for the network f:

- a standard architecture (fully connected, convolutional, residual, etc.),
- a standard activation function ($\tanh$, softplus, ReLU-like, sigmoid, Gaussian, etc.),

we prove that the NTK k of f is:

- smooth almost everywhere if the network has non-null bias terms.
- smooth everywhere if the activation is smooth.

These results, obtained from similar regularity results on the conjugate kernel of f, then transfer to f.

Resulting Convergence Results

Our finer-grained framework allows us to derive novel convergence insights, with highlighted results below.

Gradient Flow of Generated Distribution:

$$\partial_\ell\alpha^z_\ell = -\nabla_{\mathscr{G}_{k_g}}\mathscr{C}(\alpha_\ell)$$
$$= \nabla_x \cdot \left(\alpha^z_\ell \mathcal{T}_{k_g,p_z}\Big(z \mapsto \nabla_x(a \circ f_{\alpha_\ell})(x)\big|_{x=g_\ell(z)}\Big)\right),$$

where α^z_ℓ is the distribution of $\big(z, g_\ell(z)\big)$ under $z \sim p_z$.

- In the non-interacting case, i.e. $\mathcal{T}_{k_g,p_z} = \mathrm{id}$, this corresponds to a Wasserstein gradient flow:

$$\partial_\ell\alpha_\ell = -\nabla_{\mathscr{G}_{k_g}}\mathscr{C}(\alpha_\ell) = -\nabla_{\mathscr{W}}\mathscr{C}(\alpha_\ell).$$

- In the general case, this is a gradient flow in a Stein geometry defined by the generator's NTK k_g.
- $\mathscr{C}(\alpha_\ell)$ is automatically decreasing via this gradient flow, as fast as possible (locally).

IPM GANs ($a = b = \mathrm{id}$) and NTK MMD:

We find $f_t = f_0 + t\Big(\mathbb{E}_{x\sim\alpha}\big[k(x, \cdot)\big] - \mathbb{E}_{y\sim\beta}\big[k(y, \cdot)\big]\Big)$, hence, if $f_0 = 0$:

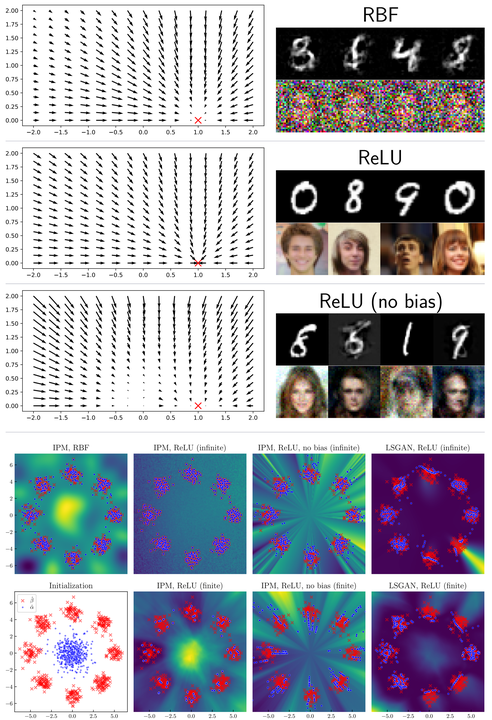
$$\mathscr{C}(\alpha) = \mathcal{L}_\alpha(f_t) \propto \mathrm{MMD}^2_k(\alpha, \beta).$$

Empirical Study

- We assess the adequacy of our framework by observing how close finite and infinite-width regimes are.
- We study the convergence of GANs on empirical distributions in the non-interacting case ($\mathcal{T}_{k_g,p_z} = \mathrm{id}$).
- We discover the singular performance of ReLU architectures for generative modeling and explain it by studying generator gradients with our framework.

Toolkit GAN(TK)^2

- GAN analysis toolkit based on our framework.
- Written in JAX with the Neural Tangents library.
- Link to paper and code ⟶



Thirty-ninth International Conference on Machine Learning — ICML 2022, Baltimore, MD, USA