A 15min introduction to GANs

Younes Boulaguiem, PhD
Postdoctoral Researcher in Statistics

University of Geneva – Faculty of Medicine
Clinical Research Center, HUG

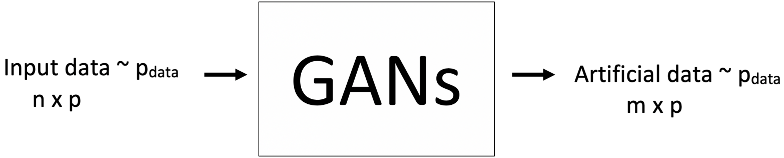
A mini-lecture on Generative Adversarial Networks.
Delivered March 2020

View Our Related Publication

Generative Adversarial Networks

- **Generative Adversarial Networks** (GANs) are a type of generative models, that impletement two **artificial neural networks** (ANNs) whose objectives are at the opposite of one another (thus adversarial).
- The objective is to generate **new samples** that look to some extent similar to the data they have been trained on, but that never existed before (no resampling, no overfitting).
- GANs are designed such that the **underlying distribution** of the data is learnt.

Generative Adversarial Networks



"What I cannot create, I do not understand" - Richard Feynman

Statistical Classification

- **Generative models** and **discriminative models** are the two main approaches in **statistical classification**. Given a continuous observable variable $\mathbf{X}$ and a discrete target variable $\mathbf{Y}$, we are interested in classifying $\mathbf{X}$ into a category determined by the values of $\mathbf{Y}$ (usually referred to as **the label**). These models are used to estimate the probability $\mathbf{P}(\mathbf{Y}|\mathbf{X} = \mathbf{x})$.
- **Generative** models estimate this probability by modelling the joint distribution $\mathbf{P}(\mathbf{X}, \mathbf{Y})$. It also allows one to **generate new instances**, e.g., Gaussian mixtures.
- **Discriminative** models estimate it by directly modelling $\mathbf{P}(\mathbf{Y}|\mathbf{X} = \mathbf{x})$ from the data, e.g., logistic regression.
- What's the use? Applications in a wide variety of domains, such as natural language processing, computer vision, data augmentation...

What is a GAN?

- In appearance, an extremely efficient generative model. It belongs to a new family of generative methods called **Deep Generative Models** (DGMs).
- "Demonstrates the **creative capability** of computers".
- Applications: **computer vision**, but also **finance** (fraud detection), **emotion recognition** (very useful to twitter), **large ensemble simulations** (climate science) and so on...
- Could you tell the real from the fake?

Pros and cons

- The advantages:
  - **No need for a-priory assumptions** about the probabilistic structure of the data or the functional relationship between the data and the target.
  - **An alternative model to Monte-Carlo Markov Chain** or unrolled approximate inference based methods to generate from distributions, as GANs implement ANNs and use **backpropagation** for training.
  - **They are fast!**
- In theory things are pretty, but in practise it can get messy. GANs are notoriously known to suffer from a whole bunch of issues, going from the usual suspects when it comes to training neural networks such as **training instability** and **convergence problems**, to GANs specific ones such as **mode collapse**.

GANs conceptually

- In reality, GANs make the best out of **generative** and **discriminative** models by **turning them against each other in a competitive framework**, thus the term **adversarial**.
- It is a game opposing two agents, **D** and **G**. At each round:
  - **G** produces **artificial** observations.
  - **D** is presented with both **artificial** and **real** data, and needs to guess whether each one is **real** or **fake**.
  - Each agent adapt their strategies in order **to minimise their losses**, or **maximise their opponent's losses**.
  - This translates into **D** learning more about **how real data look like** in order to correctly classify them, and as a consequence leading **G** to **generate more realistic samples**.

More formally

- The generator **G** is defined as a set of **parametric functions** satisfying $G = \{g_\theta(z) : \mathbb{R}^d \rightarrow \mathbb{R}^p\}$, where $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^q$ and $\boldsymbol{z} \in Z \subseteq \mathbb{R}^d$ is a latent variable with known distribution $\mathbf{p_z}$.
- The discriminator **D** is defined as a set of **parametric functions** satisfying $D = \{f_\phi(x) : \mathbb{R}^p \rightarrow [0, 1]\}$, where $\phi \in \Phi \subset \mathbb{R}^m$.
- Direct implications:
  - $\boldsymbol{g_\theta}(.)$ is deterministic but $\boldsymbol{g_\theta}(\boldsymbol{z})$ is **random** because $\boldsymbol{z}$ is random. We therefore define $\boldsymbol{p_{g_\theta}}$ as the distribution of $\boldsymbol{g_\theta}(\boldsymbol{z})$.
  - The discriminator evaluates an observation $\boldsymbol{x} \in \mathbb{R}^p$ where sometimes $\boldsymbol{x} \sim \boldsymbol{p_{data}}$ and sometimes $\boldsymbol{x} \sim \boldsymbol{p_{g_\theta}}$, and outputs an estimation of $\mathbf{P}(\mathbf{Y} = \text{“}\mathbf{real}\text{”}|\mathbf{x})$ than we denote $\boldsymbol{p_{d_\phi}}$.
  - The generator and discriminator enter an appropriate optimisation process to find the set of optimal parameters $\boldsymbol{\theta}^*$ and ϕ^* such that $\boldsymbol{p_{g_\theta}} = \boldsymbol{p_{data}}$
  - What would be the value of $\boldsymbol{p_{d_\phi}}(\boldsymbol{x})$ at optimum regardless from where $\mathbf{x}$ is sampled?

Training process

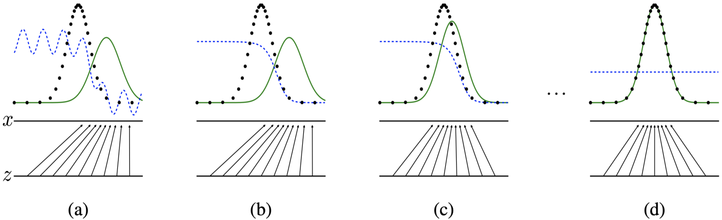


Figure: Taken from *Goodfellow et al., 2014*. The **black dotted curve** corresponds to the **distribution of the data**. The **green curve** to the **distribution of the generator p_{g_θ}**, and the **blue dashed curve** to the **discriminative distribution p_{d_ϕ}** . The **upward arrows** represent the **generator's mapping** of the latent variable $\boldsymbol{z}$ to the data space $\boldsymbol{x}$. From (a) to (d) we see the evolution of learning during the training process.

GANs and divergence minimisation

- GANs' **optimisation problem** belongs to the family of **divergence minimisation** problems represented generally as follows:

$$\min_{\theta} \mathrm{Div}(p_{data}||p_{\theta}).$$

- It can be shown that this representation can also be written as follows under certain conditions:

$$\min_{\theta} \max_{\phi} \mathcal{L}(\theta, \phi).$$

- This formulation is useful because the optimisation problem can be viewed as a **two-player zero-sum game**, where the strategy of each player is characterised by $\boldsymbol{\theta}$ and ϕ that belong to **finite sets of strategies** Θ and Φ and suffer losses $\boldsymbol{\mathcal{L}}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \phi)$ and $\boldsymbol{\mathcal{L}}^{(\boldsymbol{\phi})}(\boldsymbol{\theta}, \phi)$ respectively.
- It is **zero-sum** as $\boldsymbol{\mathcal{L}}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \phi) = -\boldsymbol{\mathcal{L}}^{(\boldsymbol{\phi})}(\boldsymbol{\theta}, \phi)$.

Divergence minimisation

- If $\mathcal{L}$ is **convex-concave** and Θ and Φ are **compact** subsets of a linear topological space, **Von-Neumann's minimax** theorem applies guaranteeing the **existence** of a **unique** solution, corresponding to the **Nash-equilibrium** of the game.
- At this point, neither player has an **incentive to deviate** from the current strategy. In other words, each player **minimises the maximum possible pay-off** for the other, and since **the game is zero-sum**, they also **minimise their own maximum loss** (or maximise their minimum pay-off).
- Consequently, **no one benefits from moving away from the Nash-equilibrium**.

The original GANs optimisation problem

- GANs objective function is the **negative-cross entropy**,
$$\min_{\theta} \max_{\phi} L(g_\theta, f_\phi) =_{x \sim p_{data}} [\log(f_\phi(x))] +_{x \sim p_z} [\log(1 - f_\phi(g_\theta(z)))].$$
- The optimisation problem **in the space of the probability density functions** corresponds to:
$$\min_{p_g} \max_{p_d} L(p_g, p_d) =_{x \sim p_{data}} [\log(p_d(x))] +_{x \sim p_g} [\log(1 - p_d(x))].$$
- It can be shown that this function **satisfies the necessary conditions for the minimax theorem to apply**, which guarantees the existence of a unique solution.
- Optimising for $\boldsymbol{p_d}$ yields $\boldsymbol{p_d^* = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}}$, also known as the **Bayes-optimal classifier**.
- The objective function under **the optimal discriminator** can be shown to be equivalent to:
$$L(p_g, p_d^*) = -log(4) + 2JSD(p_{data}||p_g),$$
where **JSD** stands for the **Jensen-Shannon divergence**, which implies a **global minimum** of $-log(4)$ at $\boldsymbol{p_g = p_{data}}$ (JSD is non-negative).

GANs in practise

- The original GANs model approximates $G = \{g_\theta(z) : \mathbb{R}^d \to \mathbb{R}^p\}$ and $D = \{f_\phi(x) : \mathbb{R}^p \to [0,1]\}$ with **ANNs** making use of the **universal approximation theorem**, stating (roughly):

  Given any continuous function on a compact subset of $\mathbb{R}^k$, one can always find a neural network that can approximate it.

- However, a neural network's parameter space is **not compact** since it is unbounded, and the **convex-concave property** of the objective function **no longer holds** as neural networks are **neither convex or concave** with respect to their parameter space, which results in **non-convex and infinite games**, and consequently providing **no guarantee of the existence of a unique solution**.

  $\implies$ **GANs in practise is messy...**

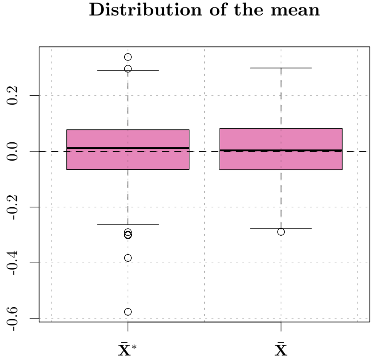
GANs for density estimation and inference?

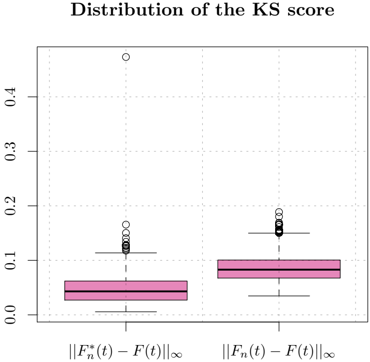
- ...**but the results are still impressive!**
- So in practise, GANs do not learn $\boldsymbol{p_{data}}$ but a"descent" **estimation** of $\boldsymbol{p_{data}}$.
- Let me denote the **true CDF** of the data $\boldsymbol{F}$, the **empirical CDF** $\boldsymbol{F_n}$ and the **GANs CDF** $\boldsymbol{F_n^*}$.
- Natural follow up questions that I do not yet know the answers to:
  - Is this true: $\forall n, \ \sup_t ||F_n(t) - F(t)|| \geq \sup_t ||F_n^*(t) - F(t)||$,
  - How fast does F_n^* converge to F: $\sup_t ||F_n^*(t) - F(t)|| = \mathcal{O}(?)$,
  - Can F_n^* be used to make inference, and what are the properties of an estimator $\widehat{\theta}^* = T(F_n^*)$, where T is an appropriate functional?
  - Can it/when does it outperform standard methods like the bootstrap?

Quick proof of concept simulation study

- for $B = 10'000$ MC simulations:
  - Generate $n = 100$ observations from a $\mathcal{N}(0, 1)$, compute $\bar{X}$ and $||F_n(t) - \Phi(t)||_\infty$.
  - Train a GAN and generate $m = 10'000$ observations, compute $\bar{X}^*$ and $||F_n^*(t) - \Phi(t)||_\infty$.

Simulation results





$mean(\bar{X}^*)$	$mean(\bar{X})$	$\widehat{var}(\bar{X}^*)$	$\widehat{var}(\bar{X})$	Cramér-Rao (σ^2/n)
0.0053	0.0043	0.0124	0.0104	0.01

- $coverage(\bar{X}^*) = 0.9509$, $coverage(\bar{X}) = 0.9508$
- $tol = [0.9457, 0.9542]$

References

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

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*, volume 27, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, volume 29, 2016.