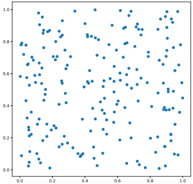
Generative Modelling Challenge
4th Tutorial Session: Generative Adversarial Networks

Michaël Allouche

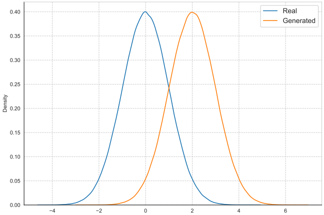
2022, November 15th

Generative model



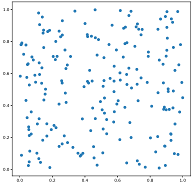
Noise

Generator



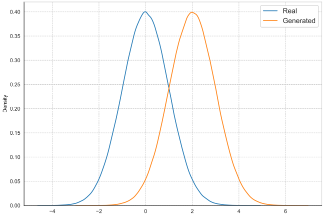
Probability density function

Generative model



Noise

Generator

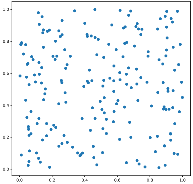


Probability density function

If X denotes the r.v. taking values in some space $\mathcal{X} \subseteq \mathbb{R}^d$ from which we have observations $(X_1, \ldots, X_n)$, the problem is to find a function $G : \mathbb{R}^{d'} \to \mathcal{X}$ and a **latent probability distribution** π on $\mathbb{R}^{d'}$ such that

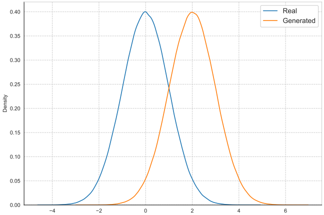
$$X \overset{\mathrm{d}}{=} G(Z) \text{ and } Z \sim \pi$$

Generative model



Noise

→ **Generator** →



Probability density function

If X denotes the r.v. taking values in some space $\mathcal{X} \subseteq \mathbb{R}^d$ from which we have observations $(X_1, \dots, X_n)$, the problem is to find a function $G : \mathbb{R}^{d'} \to \mathcal{X}$ and a **latent probability distribution** π on $\mathbb{R}^{d'}$ such that

$$X \overset{\mathrm{d}}{=} G(Z) \text{ and } Z \sim \pi$$

Existence: see the Kuratowski's Theorem [Bertsekas and Shreve, 1978, Proposition 7.15]

Questions: which π? which d'? how to learn G? dist.$(_{\text{law}}(X),_{\text{law}}(G(Z)))$?

Learning G**:** Generative Adversarial Networks (GANs) [Goodfellow et al., 2014]

Generative Adversarial Networks (GANs)

Data: $X_1, \ldots, X_n$, i.i.d. according to some unknown density p_X on $\chi \subseteq \mathbb{R}^d$

Generative Adversarial Networks (GANs)

Data: $X_1, \ldots, X_n$, i.i.d. according to some unknown density p_X on $\chi \subseteq \mathbb{R}^d$

Generator: parametric family of functions $\mathcal{G} = \left\{ G_\theta : \mathbb{R}^{d'} \to \mathbb{R}^d \right\}_{\theta \in \Theta}$

- *Goal*: learn optimal parameters $\theta^\star$ s.t. $\tilde{X} := G_{\theta^\star}(Z) \stackrel{\mathrm{d}}{=} X$

Generative Adversarial Networks (GANs)

Data: $X_1, \ldots, X_n$, i.i.d. according to some unknown density p_X on $\chi \subseteq \mathbb{R}^d$

Generator: parametric family of functions $\mathcal{G} = \left\{ G_\theta : \mathbb{R}^{d'} \to \mathbb{R}^d \right\}_{\theta \in \Theta}$

- *Goal*: learn optimal parameters $\theta^\star$ s.t. $\tilde{X} := G_{\theta^\star}(Z) \stackrel{\mathrm{d}}{=} X$

Discriminator: parametric family of functions $\mathcal{D} = \left\{ D_\phi : \mathbb{R}^d \to [0,1] \right\}_{\phi \in \Phi}$

- *Goal*: give the probability that an observation x is drawn from p_X

Generative Adversarial Networks (GANs)

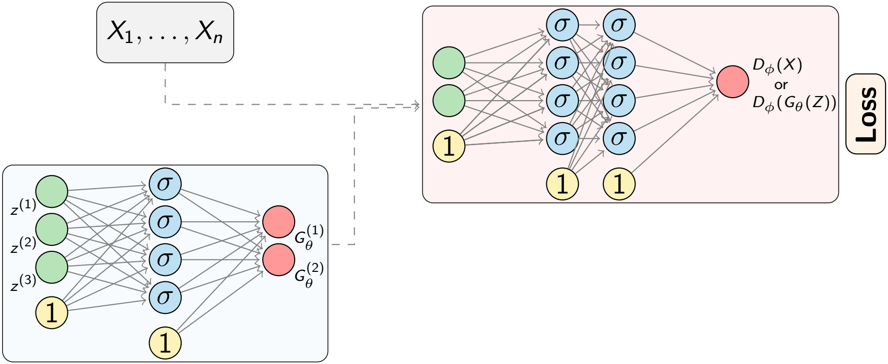
Data: $X_1, \dots, X_n$, i.i.d. according to some unknown density p_X on $\chi \subseteq \mathbb{R}^d$

Generator: parametric family of functions $\mathcal{G} = \left\{ G_\theta : \mathbb{R}^{d'} \to \mathbb{R}^d \right\}_{\theta \in \Theta}$

- *Goal*: learn optimal parameters $\theta^\star$ s.t. $\tilde{X} := G_{\theta^\star}(Z) \overset{\mathrm{d}}{=} X$

Discriminator: parametric family of functions $\mathcal{D} = \left\{ D_\phi : \mathbb{R}^d \to [0,1] \right\}_{\phi \in \Phi}$

- *Goal*: give the probability that an observation x is drawn from p_X



GAN model with $d' = 3$ and $d = 2$

Generative Adversarial Networks (GANs)

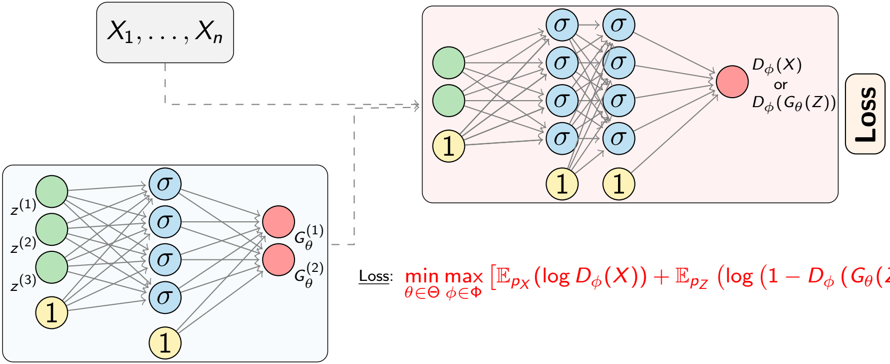
Data: $X_1, \ldots, X_n$, i.i.d. according to some unknown density p_X on $\chi \subseteq \mathbb{R}^d$

Generator: parametric family of functions $\mathcal{G} = \left\{ G_\theta : \mathbb{R}^{d'} \to \mathbb{R}^d \right\}_{\theta \in \Theta}$

- *Goal*: learn optimal parameters $\theta^\star$ s.t. $\tilde{X} := G_{\theta^\star}(Z) \overset{\mathrm{d}}{=} X$

Discriminator: parametric family of functions $\mathcal{D} = \left\{ D_\phi : \mathbb{R}^d \to [0,1] \right\}_{\phi \in \Phi}$

- *Goal*: give the probability that an observation x is drawn from p_X



Loss: $\min_{\theta \in \Theta} \max_{\phi \in \Phi} \left[\mathbb{E}_{p_X} (\log D_\phi(X)) + \mathbb{E}_{p_Z} \left(\log \left(1 - D_\phi \left(G_\theta(Z) \right) \right) \right) \right]$

GAN model with $d' = 3$ and $d = 2$

Results

Proposition ([Goodfellow et al., 2014, Proposition 1])

For any generator G_θ, the optimal discriminator is

$$D_{\phi_\theta^\star}(x) = \frac{p_X(x)}{p_X(x) + p_\theta(x)}.$$

Results

Proposition ([Goodfellow et al., 2014, Proposition 1])

For any generator G_θ, the optimal discriminator is

$$D_{\phi_\theta^\star}(x) = \frac{p_X(x)}{p_X(x) + p_\theta(x)}.$$

Theorem ([Goodfellow et al., 2014, Theorem 1])

For a given optimal discriminator, the optimal generator is achieved for $p_X = p_\theta$, so it entails from Proposition 1

$$D_{\phi_\theta^\star}(x) = \frac{1}{2}.$$

Results

Proposition ([Goodfellow et al., 2014, Proposition 1])

For any generator G_θ, the optimal discriminator is

$$D_{\phi_\theta^\star}(x) = \frac{p_X(x)}{p_X(x) + p_\theta(x)}.$$

Theorem ([Goodfellow et al., 2014, Theorem 1])

For a given optimal discriminator, the optimal generator is achieved for $p_X = p_\theta$, so it entails from Proposition 1

$$D_{\phi_\theta^\star}(x) = \frac{1}{2}.$$

$\Rightarrow$ See [Biau et al., 2020] for more theoretical results

Optimization

Theoretical Loss:

$$\min_{\theta \in \Theta} \max_{\phi \in \Phi} \left[\mathbb{E}_{p_X} \left(\log D_\phi(X) \right) + \mathbb{E}_{p_Z} \left(\log \left(1 - D_\phi \left(G_\theta(Z) \right) \right) \right) \right]$$

Optimization

Theoretical Loss:

$$\min_{\theta\in\Theta}\max_{\phi\in\Phi}\left[\mathbb{E}_{p_X}(\log D_\phi(X)) + \mathbb{E}_{p_Z}(\log(1 - D_\phi(G_\theta(Z))))\right]$$

Alternative optimization program with SGD:

1. **Discriminator update**

$$\phi^{t+1} = \phi^t + \eta_D^t \frac{1}{M}\sum_{m=1}^{M} \nabla_\phi \left[\log D_{\phi^t}(x^{(m)}) + \log\left(1 - D_{\phi^t}(G_{\theta^t}(Z^{(m)}))\right)\right]$$

Optimization

Theoretical Loss:

$$\min_{\theta \in \Theta} \max_{\phi \in \Phi} \left[\mathbb{E}_{p_X}(\log D_\phi(X)) + \mathbb{E}_{p_Z}\left(\log\left(1 - D_\phi\left(G_\theta(Z)\right)\right)\right)\right]$$

Alternative optimization program with SGD:

1. **Discriminator update**

$$\phi^{t+1} = \phi^t + \eta_D^t \frac{1}{M} \sum_{m=1}^{M} \nabla_\phi \left[\log D_{\phi^t}(x^{(m)}) + \log\left(1 - D_{\phi^t}(G_{\theta^t}(Z^{(m)}))\right)\right]$$

2. **Generator update**

$$\theta^{t+1} = \theta^t - \eta_G^t \frac{1}{M} \sum_{m=1}^{M} \nabla_\theta \log\left(1 - D_{\phi^{t+1}}(G_{\theta^t}(Z^{(m)}))\right)$$

Optimization

Theoretical Loss:

$$\min_{\theta \in \Theta} \max_{\phi \in \Phi} \left[\mathbb{E}_{p_X}\left(\log D_\phi(X)\right) + \mathbb{E}_{p_Z}\left(\log\left(1 - D_\phi\left(G_\theta(Z)\right)\right)\right)\right]$$

Alternative optimization program with SGD:

1. **Discriminator update**

$$\phi^{t+1} = \phi^t + \eta_D^t \frac{1}{M} \sum_{m=1}^{M} \nabla_\phi \left[\log D_{\phi^t}(x^{(m)}) + \log\left(1 - D_{\phi^t}(G_{\theta^t}(Z^{(m)}))\right)\right]$$

2. **Generator update**

$$\theta^{t+1} = \theta^t - \eta_G^t \frac{1}{M} \sum_{m=1}^{M} \underbrace{\nabla_\theta \log\left(1 - D_{\phi^{t+1}}(G_{\theta^t}(Z^{(m)}))\right)}_{\to 0(\triangle \mathit{VanishingGradient} \triangle)}$$

Optimization

Theoretical Loss:

$$\min_{\theta\in\Theta}\max_{\phi\in\Phi}\left[\mathbb{E}_{p_X}(\log D_\phi(X)) + \mathbb{E}_{p_Z}\left(\log\left(1 - D_\phi\left(G_\theta(Z)\right)\right)\right)\right]$$

Alternative optimization program with SGD:

1. **Discriminator update**

$$\phi^{t+1} = \phi^t + \eta_D^t \frac{1}{M}\sum_{m=1}^{M}\nabla_\phi\left[\log D_{\phi^t}(x^{(m)}) + \log\left(1 - D_{\phi^t}(G_{\theta^t}(Z^{(m)}))\right)\right]$$

2. **Generator update**

$$\theta^{t+1} = \theta^t - \eta_G^t \frac{1}{M}\sum_{m=1}^{M}\underbrace{\nabla_\theta \log\left(1 - D_{\phi^{t+1}}(G_{\theta^t}(Z^{(m)}))\right)}_{\to 0(\triangle\!\!!\,\textit{VanishingGradient}\,\triangle\!\!!)}$$

Numerical Loss:

$$\max_{\theta\in\Theta}\max_{\phi\in\Phi}\left[\mathbb{E}_{p_X}(\log D_\phi(X)) + \mathbb{E}_{p_Z}\left(\log\left(D_\phi\left(G_\theta(Z)\right)\right)\right)\right]$$

Numerical issues

Challenges:

- **Oscillating Loss.** Non-convergence with large oscillations during the iterations.
- **Hyperparameter sensitive.** GANs require large number of hyperparameters to tune (batch-size, latent space, learning rate, ...) and they are very sensitive to small changes.
- **Mode collapse.** The Generator is overfitted and collapses to a single mode, *i.e.* it simulates the same objects (no diversification).

Attempt to remedy:

- Consider other loss functions (WGAN, EBGAN, ...)
- Consider penalization terms (Spectral normalization, Gradient penalty, ...)

References I

Bertsekas, D. P. and Shreve, S. E. (1978).
Stochastic optimal control, volume 139 of *Mathematics in Science and Engineering*.
Academic Press, Inc., New York-London.

Biau, G., Cadre, B., Sangnier, M., Tanielian, U., et al. (2020).
Some theoretical properties of gans.
Annals of Statistics, 48(3):1539–1566.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
Generative adversarial nets.
In *Advances in neural information processing systems*, pages 2672–2680.