Introduction to Generative Adversarial Networks.

Adalberto Claudio Quiros - a.claudio-quiros.1@research.gla.ac.uk

April 22, 2020

Computing Science Department

High Level Perspective

What's a GAN? Where do they fall in the Machine Learning perspective?

- Deep Learning Model.
- Unsupervised Learning: Generative Models.

What's the goal of a GAN then?

- Learn a distribution of data whether if it is images, sound, vectors, anything.
- Learn it in an unsupervised way: no label or initial knowledge about the data, just samples.

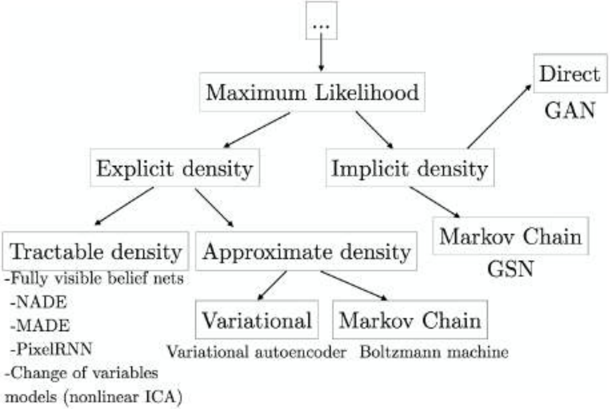
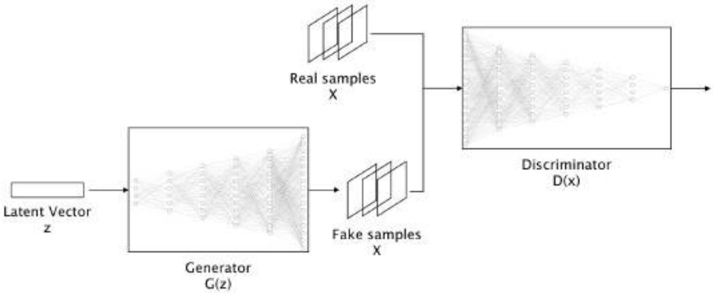


Figure 1: Generative Models - Goodfellow NIPS 2016 Tutorial: Generative Adversarial Networks.

Model Diagram:



Core Idea:

- Two modules: Generator G and Discriminator D.
- The Generator G and Discriminator D compete between each other. G to recreate real data samples and D to distinguish between real and generated samples.

1. **Generator:**
    - **Goal**: Replicate real data.
    - **How**: Modifies an prior distribution $p_z(z)$ (latent space) into a distribution $p_g(x)$, making it as close as possible to the real data distribution $p_{data}(x)$
    - $G(z;\theta_g)$: Function that maps a latent sample $z = \{z_1, ..., z_m\} \sim p_z(z)$ to a data sample $x = \{x_1, ..., x_n\} \sim p_g(x)$.

$$G : z \to x$$

$$\textit{where } z \in \mathfrak{R}^m;\ x \in \mathfrak{R}^n$$

2. **Discriminator:**
    - **Goal**: Distinguish between real and fake samples.
    - **How**: Measures the probability of a sample x_{real} or x_{fake} of coming from the real data distribution $p_{data}(x)$.
    - $D(x;\theta_d)$: Function that maps a data sample x to a scalar d. $d = 0$ if the discriminator assumes $x = x_{fake}$ and $d = 1$ if the discriminator assumes $x = x_{real}$.

$$D : x \to d$$

$$\textit{where } x \in \mathfrak{R}^n;\ d \in \mathfrak{R}$$

3. **Generator G and Discriminator D compete to 'trick' each other. Discriminator pushes Generator to replicate the real data distribution $p_{data}(x)$**

Loss Function:

- Cross entropy loss.
- Generator G tries to minimize accuracy of the Discriminator D, .

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[log\ D(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]$$

Discriminator D:

- First part: Improve the probability of recognizing real data.
- Second part: Improve the probability of recognizing fake data.

$$\max_{D} V(D) = \mathbb{E}_{x \sim p_{data}(x)}[log\ D(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]$$

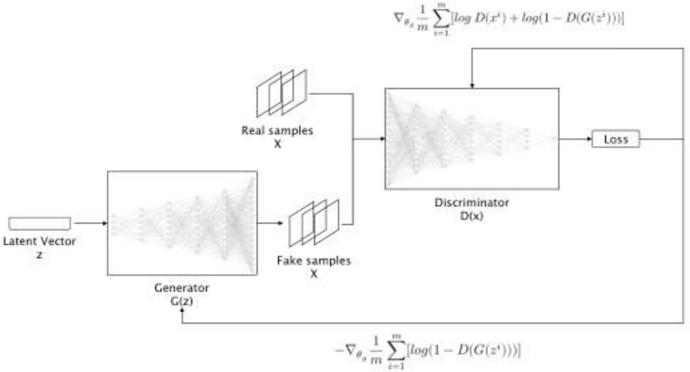
Generator G:

- Improve quality of the generated samples so they will fool the Discriminator D.

$$\min_{G} V(G) = \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]$$

Discriminator D and Generator G gradients:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [log\ D(x^i) + log(1 - D(G(z^i)))] \ \text{and} \ -\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} [log(1 - D(G(z^i)))]$$



Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Figure 2: Training Algorithm Goodfellow et al.[1].

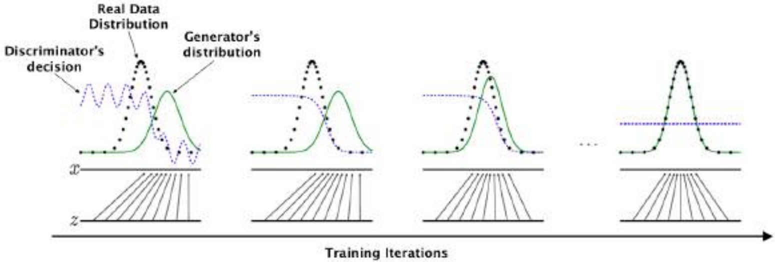


Figure 3: Generated data distribution over training [1].

Let's see it: GAN training

Let's assume a fixed Generator $G(z)$, and find the optimal Discriminator $D^*(x)$, the best Discriminator that we can train for a fixed Generator (**This is what we do in the first part of the algorithm !**):

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[log\ D(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))]$$

If Discriminator $D(x)$ wants to maximize the loss function $V(D, G)$, which Discriminator achieves this maximum:

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[log\ D(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))] =$$

$$\int_x p_{data}(x) log\ D(x) dx + \int_z p_z(z) log(1 - D(G(z))) dz =$$

$$\int_x (p_{data}(x) log\ D(x) dx + p_g(x) log[1 - D(x))]) dx$$

$$\frac{\partial V(D, G)}{\partial D} = 0 \rightarrow p_{data}(x) \frac{1}{D(x)} - p_g(x) \frac{1}{1 - D(x)} = 0$$

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

What's the loss function for the Optimal Discriminator $D^*(x)$?

$$\min_G V(D^*, G) = \int_x (p_{data}(x) log\ D^*(x)dx + p_g(x) log(1 - D^*(x))))dx =$$

$$\int_x (p_{data}(x) log[\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}] + p_g(x) log[\frac{p_g(x)}{p_{data}(x) + p_g(x)}])dx =$$

$$\int_x (p_{data}(x)(-log(2)) + p_g(x)(-log(2)))dx +$$

$$\int_x (p_{data}(x) log[\frac{p_{data}(x)}{\frac{p_{data}(x)+p_g(x)}{2}}] + p_g(x) log[\frac{p_{data}(x)}{\frac{p_{data}(x)+p_g(x)}{2}}])dx =$$

$$\int_x (p_{data}(x) + p_g(x))(-log(2)))dx +$$

$$2[\frac{1}{2} D_{KL}(p_r(x)||\frac{p_{data}(x) + p_g(x)}{2}) \ + \ \frac{1}{2} D_{KL}(p_g(x)||\frac{p_{data}(x) + p_g(x)}{2})] \rightarrow$$

$$\min_{\mathbf{G}} \mathbf{V(D^*, G) = D_{JS}(p_r(x)||p_g(x)) - log(4)}$$

Assuming a fixed $G(z)$, and optimal $D^*(x) \rightarrow$

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

$$\min_G V(D^*, G) = D_{JS}(p_r(x)||p_g(x)) - log(4)$$

So minimizing our GAN cost is the same as minimizing the Jensen-Shannon Distance between $p_r(x)$ and $p_{data}(x)$:

$$D_{JS}(p_r(x)||p_g(x))$$

For our GAN optimal point, our generator $G^*(z)$ matches the real data:
$p_r(x) = p_{data}(x)$

$$V(D^*, G^*) = -log(4)$$

Training-Objective Function:

- WGAN/WGAN-GP.
- Relativistic GAN.
- Geometric GAN.
- f-GAN.
- Spectral Normalization GAN.
- BigGAN.

Structural Networks Changes:

- Self-Attention GAN.
- StyleGAN.
- ProGAN.
- DCGAN.

Image Quality and Diversity:

- ProGAN/StyleGAN.
- BigGAN.
- Self-Attention GAN.

Representation Learning:

- InfoGAN.
- BiGAN/BigBiGAN.
- StyleGAN/StyleGAN2.

Transfer Learning:

- Star-GAN.
- Cycle-GAN.

ImageNet: 1M images and 20K classes.

Figure 4: Generated and uncurated images from a BigGAN. [2]

This person does not exist.

Figure 5: Generated and uncurated images from a StyleGAN2. Karras et al. 2020.

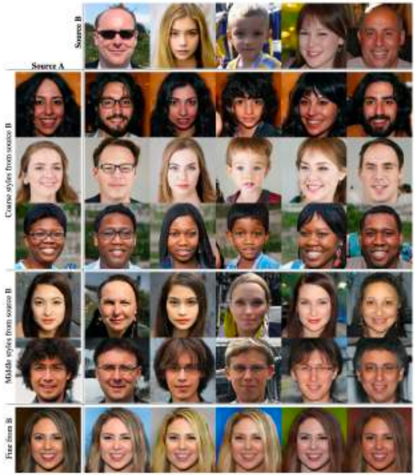


Figure 6: Style mixing on StyleGAN. Karras et al. 2018

Representation learning on ImageNet (1M images and 20K classes)

Architecture	Top-1	Top-5
ResNet-50	76.3	93.1
ResNet-101	77.8	93.8
RevNet-50	71.8	90.5
RevNet-50 ×2	74.9	92.2
RevNet-50 ×4	76.6	93.1

Method	Architecture	Feature	Top-1	Top-5
BigBiGAN (ours)	ResNet-50	AvePool	55.4	77.4
	ResNet-50	BN+CReLU	56.6	78.6
	RevNet-50 ×4	AvePool	60.8	81.4
	RevNet-50 ×4	BN+CReLU	61.3	81.9

Figure 7: Image enconding/reconstruction and representation learning accuracy on BigBiGAN. Donahue et al. 2019. []

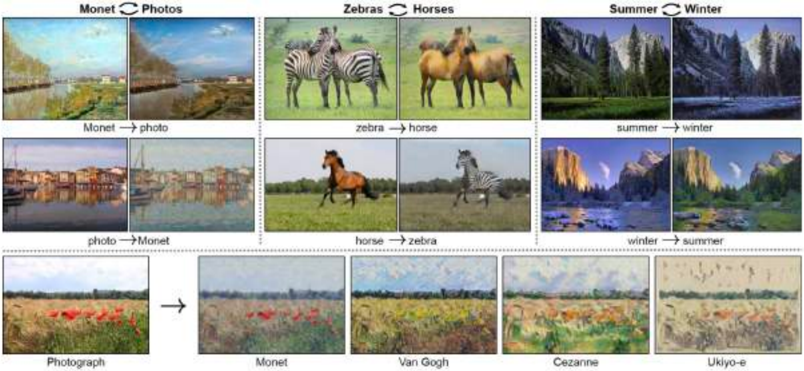


Figure 8: Cycle-GAN Zhu et al. 2017

Further Dip Into GANs

- GANs:
  - Maximum Likelihood Models vs GANs.
- GAN Problems:
  - Non-convergence.
  - Diminished Gradient.
  - Unstable Gradients.
  - Mode Collapse.
- Evaluation:
  - Inception Score.
  - Frechet Inception Distance.
- General lines of research:
  - Stabilizing training and Improving learning.
  - Image Quality and Variety.
  - Representation learning.
  - Structural Changes.
  - Transfer Learning.

- **Further dip into GANs:** Myself - 2020
- **Another GAN introduction and models review:** Myself - 2020
- **Brief introduction to GANs:** Jonathan Hui - Medium
- **GAN models reading list, this is a good introduction to relevant models:** Jonathan Hui - Medium
- **Goodfellow NIPS 2016 Tutorial:**
  - Paper
  - Video
- **Play with Generative Adversarial Networks in your browser:** Link
- **Pytorch DCGAN Tutorial:** Link

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio.
Generative adversarial nets.
In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan.
Large scale GAN training for high fidelity natural image synthesis.
CoRR, abs/1809.11096, 2018.