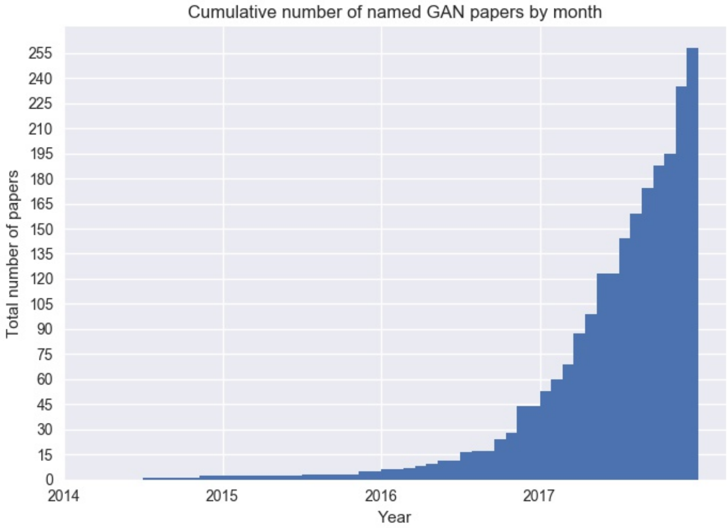
Introduction to GANs

Akshay Kumar

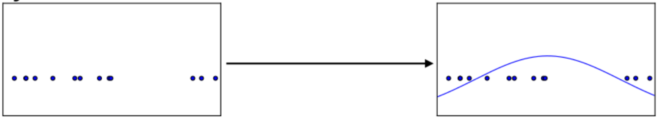
March 3, 2018



Credit: Bruno Gavranovic (https://github.com/bgavran)

Generative Modeling

- Density estimation

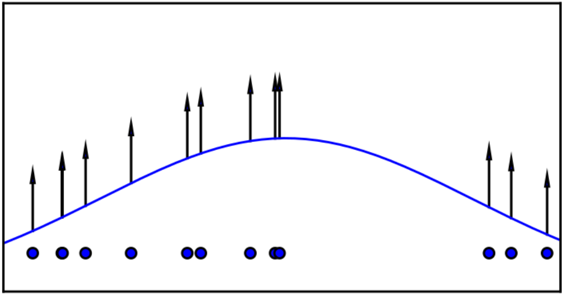


- Sample generation

Training examples Model samples

Credit: Goodfellow NIPS 2016

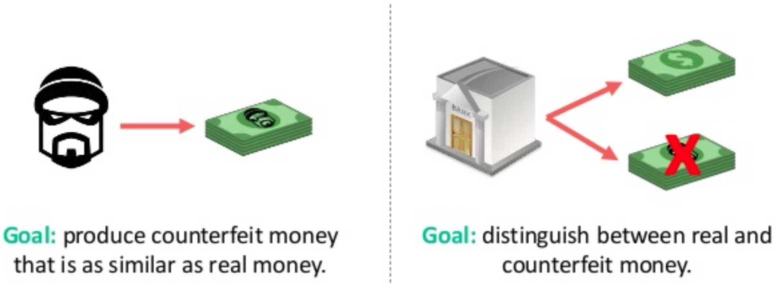
Maximum Likelihood



$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

Credit: Goodfellow NIPS 2016

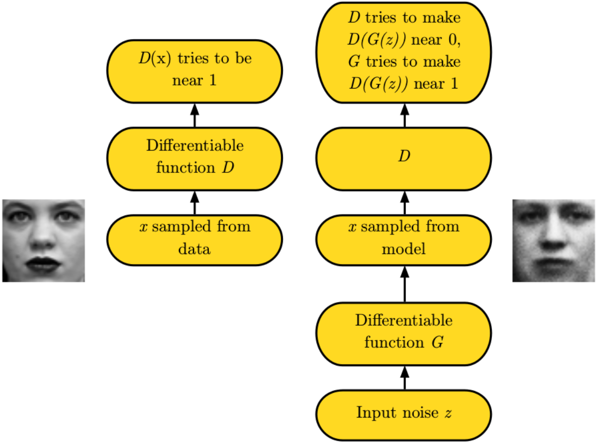
Intuition



Credit: Thomas Paula

Adversarial Networks Framework

- $D(\mathbf{x}; \theta_d)$: Multilayer discriminator
- $G(\mathbf{z}; \theta_g)$: Multilayer generator
- $\mathbf{z}$: Random vector



Credit: Goodfellow NIPS 2016

Minimax Game

- $p_{data}(\mathbf{x})$: distribution of input data
- $p_z(\mathbf{z})$: distribution of random noise
- $p_g(\mathbf{x})$: distribution of model data

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

Minimax Game

- $p_{data}(\mathbf{x})$: distribution of input data
- $p_z(\mathbf{z})$: distribution of random noise
- $p_g(\mathbf{x})$: distribution of model data

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

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

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

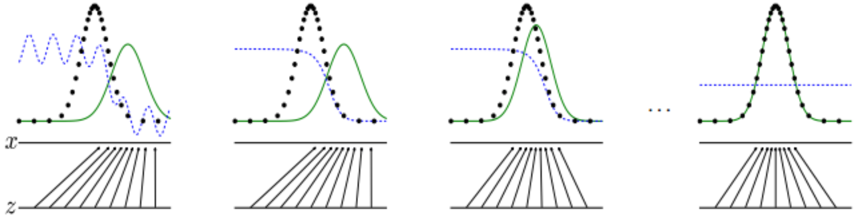
end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Credit: Generative Adversarial Nets, I. Goodfellow et al. NIPS 2014

Minimax Game

- $p_{data}(\mathbf{x})$: distribution of input data - (Black dotted curve)
- $p_z(\mathbf{z})$: distribution of random noise
- $p_g(\mathbf{x})$: distribution of model data - (Green solid curve)

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



(a) (b) (c) (d)

Credit: Generative Adversarial Nets, I. Goodfellow et al. NIPS 2014

Theoretical Results

Proposition 1. For G fixed, the optimal discriminator D is,

$$D_G^*(\mathbf{x}) = \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})} \tag{2}$$

Proof.

$$V(D, G) = \int_x p_{data}(\mathbf{x}) \log D(\mathbf{x}) dx + \int_z p_z(\mathbf{z}) \log(1 - D(G(\mathbf{z}))) dz \tag{3}$$

$$= \int_x p_{data}(\mathbf{x}) \log D(\mathbf{x}) dx + p_g(\mathbf{x}) \log(1 - D(\mathbf{x})) dx \tag{4}$$

For any $(a, b) \in R^2 \backslash \{0, 0\}$, the function y $\rightarrow a \log(y) + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$.

Theoretical Results

Theorem 1. The global minimum of the virtual training criterion $C(G)$ is achieved at $p_g = p_{data}$. At that point, $C(G)$ achieves the value -log(4). where,

$$C(G) = \max_D V(D, G)$$

Proof.

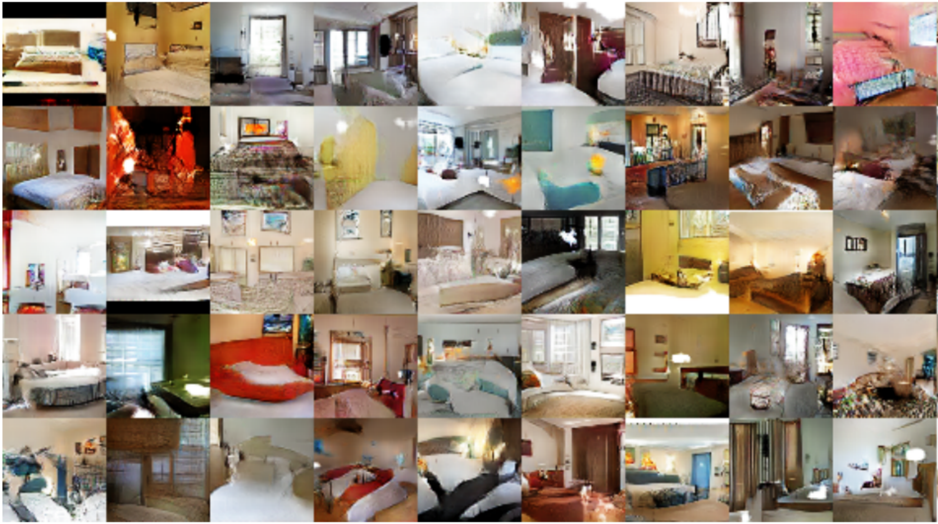
$$
\begin{aligned}
C(G) &= \max_D V(D, G) \\
&= \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[\log(1 - D_G^*(G(\mathbf{z})))] \\
&= \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p_g(\mathbf{x})}[\log(1 - D_G^*((\mathbf{x}))] \\
&= \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}\left[\log \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})}\right] + \mathbb{E}_{\mathbf{x} \sim p_g(\mathbf{x})}\left[\log \frac{p_g(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})}\right] \\
&= -log(4) + KL\left(p_{data} || \frac{p_{data}(\mathbf{x}) + p_g(\mathbf{x})}{2}\right) + KL\left(p_g || \frac{p_{data}(\mathbf{x}) + p_g(\mathbf{x}}{2}\right. \\
&= -log(4) + JSD(p_g || p_{data})
\end{aligned}
$$

Results

- New samples, not memorized

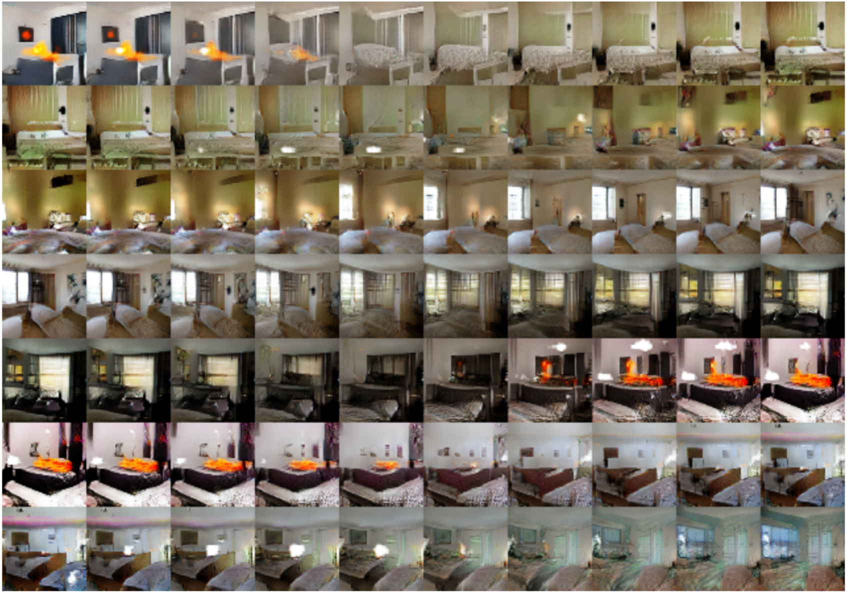
Credit: Generative Adversarial Nets, I. Goodfellow et al. NIPS 2014

Bedroom Dataset

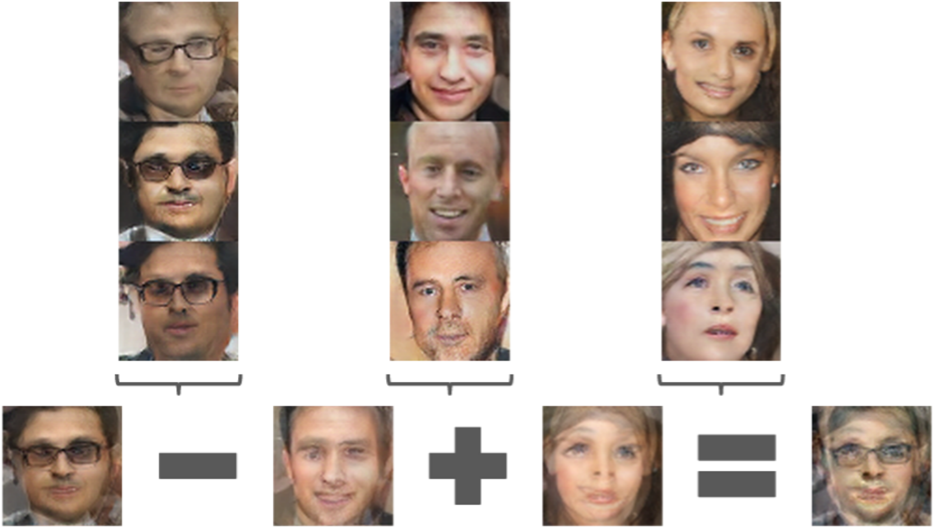


Credit: Unsupervised Representation Learning with DCGAN, Alec Radford et al.

Moving in Latent space



Vector arithmetic



Vector arithmetic

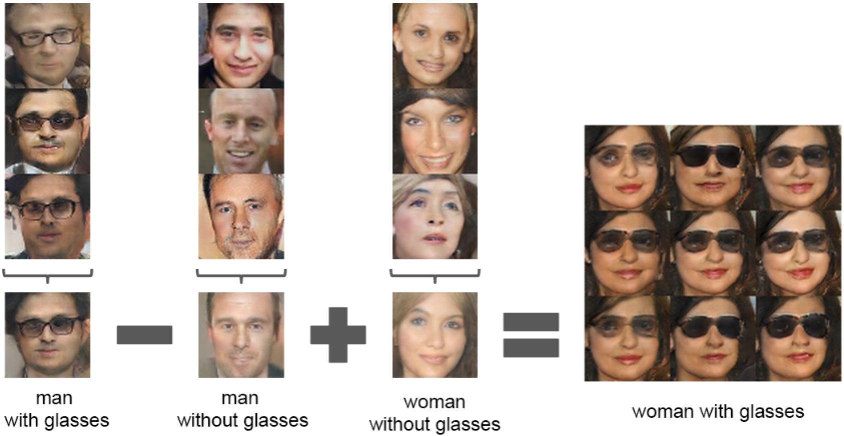
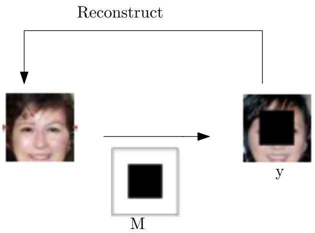


Image inpainting



- Train GAN on face dataset to get G and D
- Find the closest point in the latent space

$$\hat{\mathbf{z}} = \min_{z} \mathcal{L}_c(\mathbf{z}|\mathbf{y}, \mathbf{M}) + \mathcal{L}_p(\mathbf{z}) \tag{5}$$

where, $\mathcal{L}_c(\mathbf{z}|\mathbf{y}, \mathbf{M}) = ||\mathbf{W} \odot (G(\mathbf{z}) - \mathbf{y})||_1$ and $\mathcal{L}_p(\mathbf{z}) = \lambda \log(1 - D(G(\mathbf{z})))$

$$\mathbf{W}_i = \begin{cases} \sum_{j \in N(i)} \frac{1-\mathbf{M}_j}{|N(i)|} & \textit{if} \ \ \mathbf{M}_i \neq 0 \\ 0 & \textit{if} \ \ \mathbf{M}_i = 0 \end{cases}$$

Semantic Image Inpainting with Deep Generative Models, Raymond A. Yeh et al.

Image inpainting

- Train GAN on face dataset to get G and D
- Find the closest point in the latent space

$$\hat{\mathbf{z}} = \min_{z} \mathcal{L}_c(\mathbf{z}|\mathbf{y}, \mathbf{M}) + \mathcal{L}_p(\mathbf{z})$$

where, $\mathcal{L}_c(\mathbf{z}|\mathbf{y}, \mathbf{M}) = ||\mathbf{W} \odot (G(\mathbf{z}) - \mathbf{y})||_1$ and $\mathcal{L}_p(\mathbf{z}) = \lambda \log(1 - D(G(\mathbf{z})))$

$$\mathbf{W}_i = \begin{cases} \sum_{j \in N(i)} \frac{1 - \mathbf{M}_j}{|N(i)|} & if \;\; \mathbf{M}_i \neq 0 \\ 0 & if \;\; \mathbf{M}_i = 0 \end{cases}$$

Image inpainting

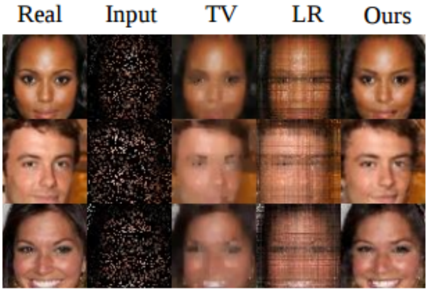


Figure 6. Comparisons with local inpainting methods TV and LR inpainting on examples with random 80% missing.

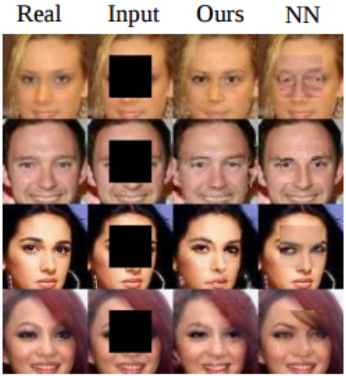


Figure 7. Comparisons with nearest patch retrieval.

Image inpainting

Figure 9. Comparisons with CE on the CelebA dataset.

Compressed Sensing using Generative Models

Image compression using sparse signal processing:

- Sparse in DCT or wavelet basis
- Pre-multiply by random (Gaussian) matrix
- Reconstruct using sparse recovery
- Exact recovery $m = O(s \log(n))$

Image compression using generative models:

- Train a GAN network to get G
- $G : R^k \rightarrow R^n$ (k is fixed)
- Minimize $||y - AG(z)||$ using gradient descent

A. Bora, A. Jalal, E. Price, and A. Dimakis, Compressed sensing using generative models, Proc. Int. Conf. Machine Learning, 2017.

Theorem

Let $G : R^k \rightarrow R^n$ *be a generative model from a d-layer neural network using ReLU activations. Let* $A \in R^{m \times n}$ *be a random Gaussian matrix for* $m = O(kd\log(n))$*, scaled so* $A_{i,j} \sim \mathcal{N}(0, \frac{1}{m})$*. For any* $x^* \in R^n$ *and any observations* $y = Ax^* + n$*, let* $||y - AG(\hat{z})||_2 \leq \epsilon$*. Then with* $1 - e^{-\Omega(m)}$ *probability,*

$$||G(\hat{z}) - x^*||_2 \leq 6 \min_{z^* \in R^k} ||G(z) - x^*||_2 + 3||n||_2 + 2\epsilon$$

CelebA Dataset

- 200,000 face images. $64 \times 64 \times 3 = 12288$ inputs per image, k=100.

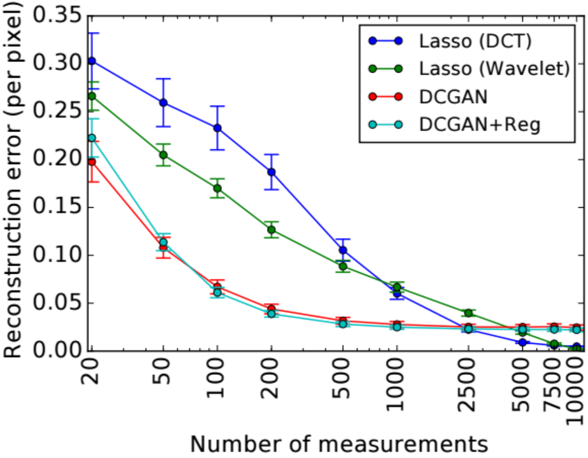


Figure 1 : Plot of per pixel reconstruction error as we vary the number of measurements. The vertical bars indicate 95% confidence intervals.

CelebA Dataset

- 200,000 face images. $64 \times 64 \times 3 = 12288$ inputs per image, k=100.

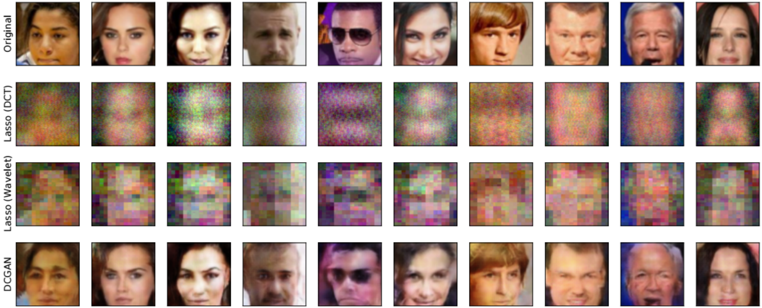


Figure 3. Reconstruction results on celebA with $m = 500$ measurements (of $n = 12288$ dimensional vector). We show original images (top row), and reconstructions by Lasso with DCT basis (second row), Lasso with wavelet basis (third row), and our algorithm (last row).

Other works

- **Generalization and Equilibrium in Generative Adversarial Nets (GANs)** - Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma and, Yi Zhang
- **Learning to Protect Communications with Adversarial Neural Cryptography** - Martin Abadi and David G. Andersen
- **Deep Generative Adversarial Networks for Compressed Sensing (GANCS) Automates MRI** - Morteza Mardani, Enhao Gong, Joseph Y. Cheng, Shreyas Vasanawala, Greg Zaharchuk, Marcus Alley, Neil Thakur, Song Han, William Dally, John M. Pauly, and Lei Xing