Lecture 7: Generative Adversarial Networks

Start Recording!

Reminders

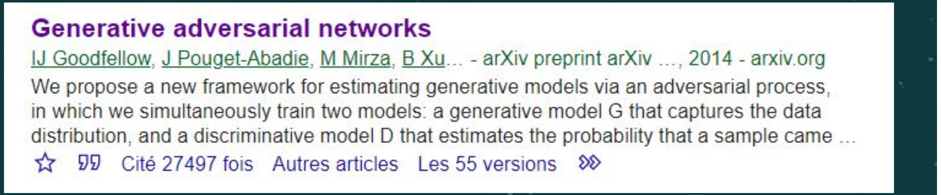
- Office Hours tomorrow (11-12h)
- Talks this Friday. Read the papers
    - https://arxiv.org/abs/1703.10593
    - https://arxiv.org/pdf/1606.00709.pdf

Ask Questions on the papers on TEAMS!

- Time to start to seriously think about your project (if not done yet).
- Deadline for the extended abstract around March 12th.

References related to this course:

1. Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks." *NeurIPS* (2014).
2. Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *ICLR* (2016).

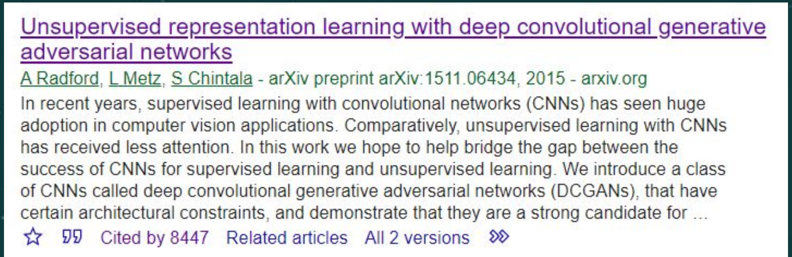


Generative adversarial networks

IJ Goodfellow, J Pouget-Abadie, M Mirza, B Xu... - arXiv preprint arXiv ..., 2014 - arxiv.org

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came ...

☆ ❞ Cité 27497 fois Autres articles Les 55 versions ≫



Unsupervised representation learning with deep convolutional generative adversarial networks

A Radford, L Metz, S Chintala - arXiv preprint arXiv:1511.06434, 2015 - arxiv.org

In recent years, supervised learning with convolutional networks (CNNs) has seen huge adoption in computer vision applications. Comparatively, unsupervised learning with CNNs has received less attention. In this work we hope to help bridge the gap between the success of CNNs for supervised learning and unsupervised learning. We introduce a class of CNNs called deep convolutional generative adversarial networks (DCGANs), that have certain architectural constraints, and demonstrate that they are a strong candidate for ...

☆ ❞ Cited by 8447 Related articles All 2 versions ≫

Supervised Learning

Goal: Given an input *x*, predict and output *y*

What we have: Observations: $(x_i, y_i)\,, i = 1, \ldots, n$

Examples:

- Input: Images, sound, text, video.
- Output: Labels (prediction), real response (regression).

Supervised Learning

Goal: Given an input *x*, predict and output *y*

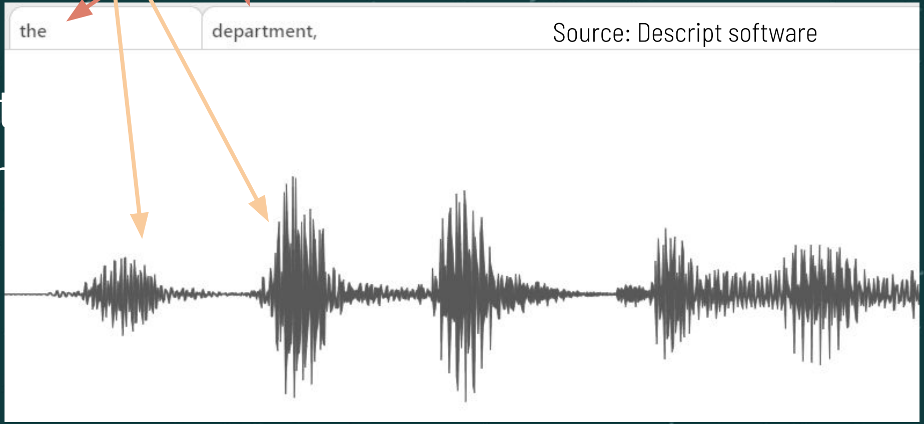
What we have: Observations: $\left(\boxed{x_i}, \boxed{y_i}\right), \; i = 1, \ldots, n$

und, text

rediction

Old label: pier
Real L: dock; pier; speedboat; sandbar; seashore
Old label: hammer
Real L: screwdriver; hammer; power drill; carpenter's kit

[Sagawa et al. 2020]



Source: Descript software

Unsupervised Learning

Goal: ????

What we have: Observations: $(x_i, \cancel{y_i})\,,\, i = 1, \ldots, n$

Examples:

- Input: Images, sound, text, video.
- Output: **NONE!**

Unsupervised Learning

High level: Understand the "world" from data.

Goal: Clustering, Dimensionality reduction, Feature learning, Density estimation.

What we have: Observations: $\left(x_i, \cancel{y_i}\right), i = 1, \ldots, n$

Main body of the CAKE!!!!

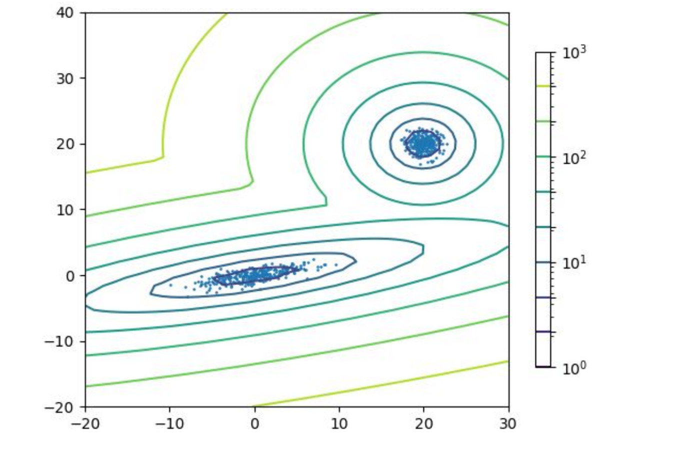
Examples:

- Input: Images, sound, text, video.
- Output: **NONE!**

Generative Modeling

- Unsupervised learning.
- Goal: Estimate the data distribution. Find θ such that $p_\theta(x) \approx p_{data}(x)$

Source: scikit-learn.org



- Standard technique: Maximum Likelihood Estimation

$$\max_\theta \prod_{i=1}^{n} p_\theta(x_i)$$

- Statisticians love it.
- Suffers from curse of dimensionality

Large Log-Likelihood but Poor Samples

- Let us assume we have: (example from [Theis et al. 2016])

$$p_\theta(x) = 0.01 \cdot p_{data}(x) + 0.99 \cdot q(x)$$

Very "good"

Very "bad" (e.g. noise)

"Bad" samples 99% of the time

- But very high log-likelihood:

$$\log(p_\theta(x_i)) \geq \log(p_{data}(x_i)) - \log(100)$$

Scale with d!! (≅ 100 for CIFAR)

≅ 4.61

Large Log-Likelihood and Great Samples

- Let us assume we have: (example from [Theis et al. 2016])

$$p_\theta(x) = 0.01 \cdot p_{data}(x) + 0.99 \cdot q(x)$$

Very "good"

Very ~~"bad"~~ Good

~~"Bad"~~ Good samples ≅100% of the time

- But very high log-likelihood:

$$\log(p_{data}(x_i)) \geq \log(p_\theta(x_i)) \geq \log(p_{data}(x_i)) - \log(100)$$

Scale with d!!

≅ 4.61

Large Log-Likelihood

- L

Conclusion:

- First case: **nonrealistic** samples 99% of the time.
- Second case: **realistic** samples ≅100% of the time.
- In both case when d large log-likelihoods (LL) are **very similar**.

For instance for CIFAR-10 two generative models may have variations in LL to the order of ≅ 100 !!! [van den Oord & Dambre, 2015]

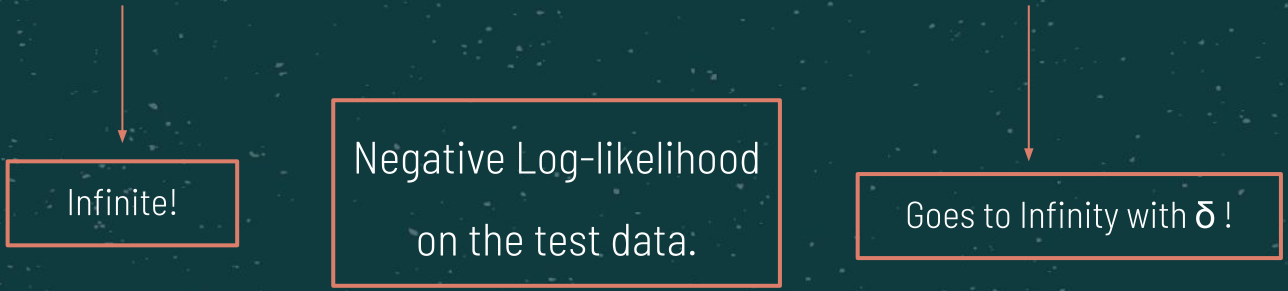
- B

$$\log(p_{data}(x_i)) \geq \log(p_\theta(x_i)) \geq \underbrace{\log(p_{data}(x_i))}_{\text{Scale with d!!}} - \underbrace{\log(100)}_{\cong 4.61}$$

Poor LL and Great Looking Samples

- Let us assume we have memorized the train set: (from [Theis et al. 2016])
- Our density:

$$p_\theta(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x = x_i^{train}\} \quad \text{or} \quad p_\theta(x) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{N}(x, x_i^{train}, \delta I_d)$$



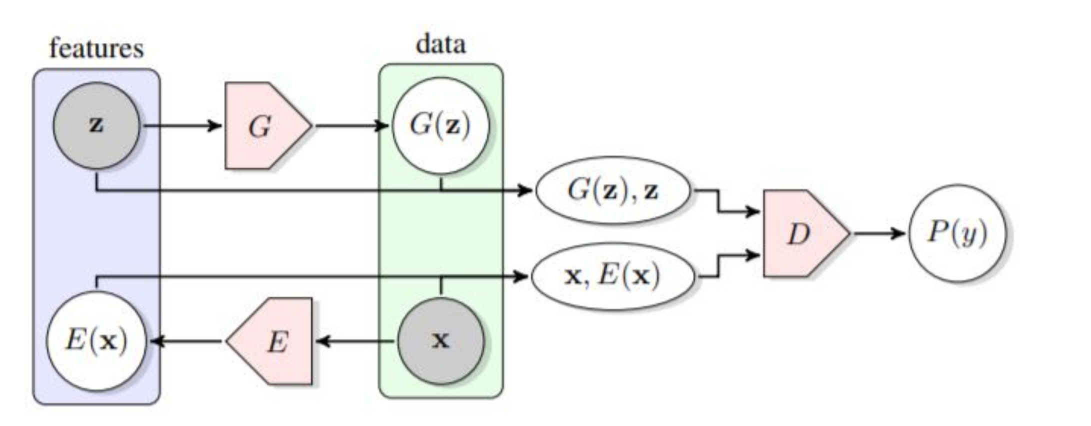
Infinite!

Negative Log-likelihood on the test data.

Goes to Infinity with δ !

Why Generative Modeling?

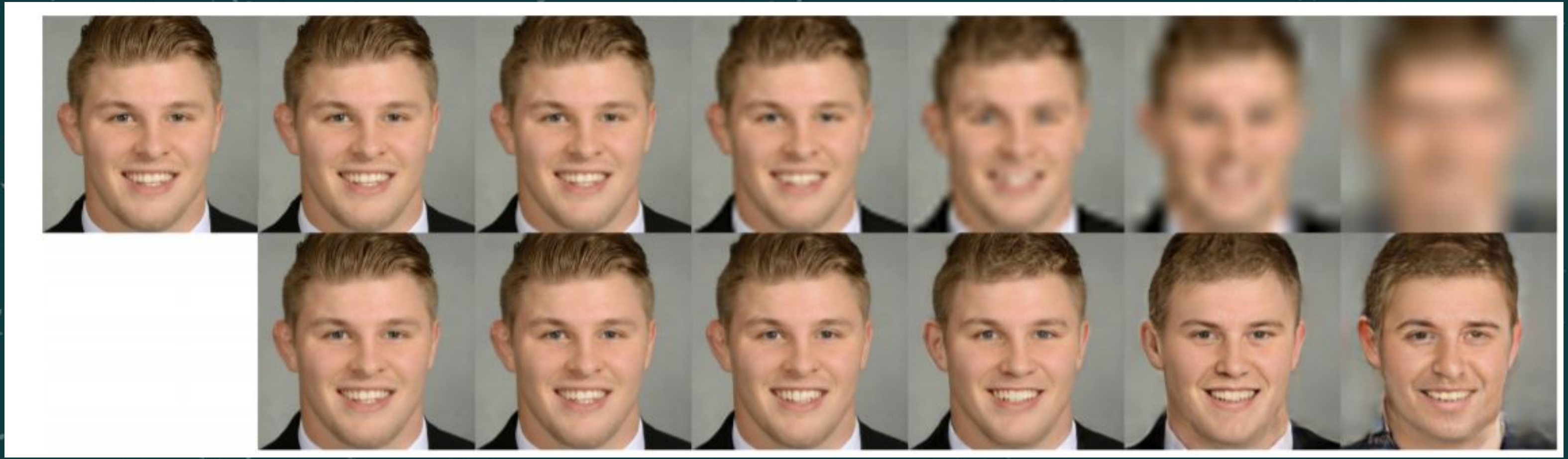
- Unsupervised learning (we can learn meaningful latent features)



[Donahue et al. 2017]

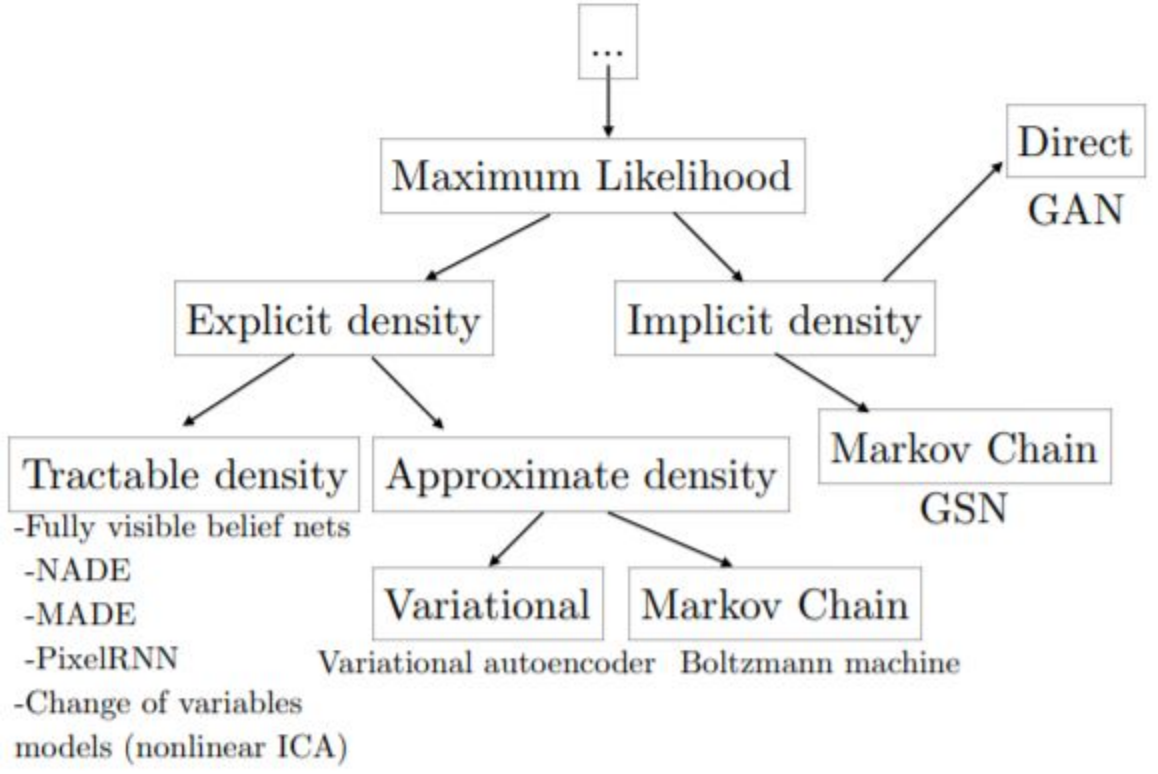
Why Generative Modeling?

- Super-resolution.



[Yang et al. 2020]

Taxonomy of Generative Models



[Goodfellow 2016]

The main Deep Generative Models

- Variational AutoEncoder [Kingma et al. 2013]
- Pixel RNN [van den Oord et al. 2016], Pixel CNN [van den Oord et al. 2016]

Best Paper Award ICML 2016

For VAE you can check the Deep Learning book

- Generative Adversarial Networks. (Today)

This course will only cover GANS

Standard Technique Using Explicit Density

From Bayes' rule:

$$p_{model}(x) = p_{model}(x_1) \prod_{i=2}^{d} p_{model}(x_i | x_1, \ldots, x_{i-1})$$

Problem:

- O(d) generation cost. (slow)
- Not clear what the order we want to set for the variables.
- No latent space. (no features extractions)

Implicit distribution

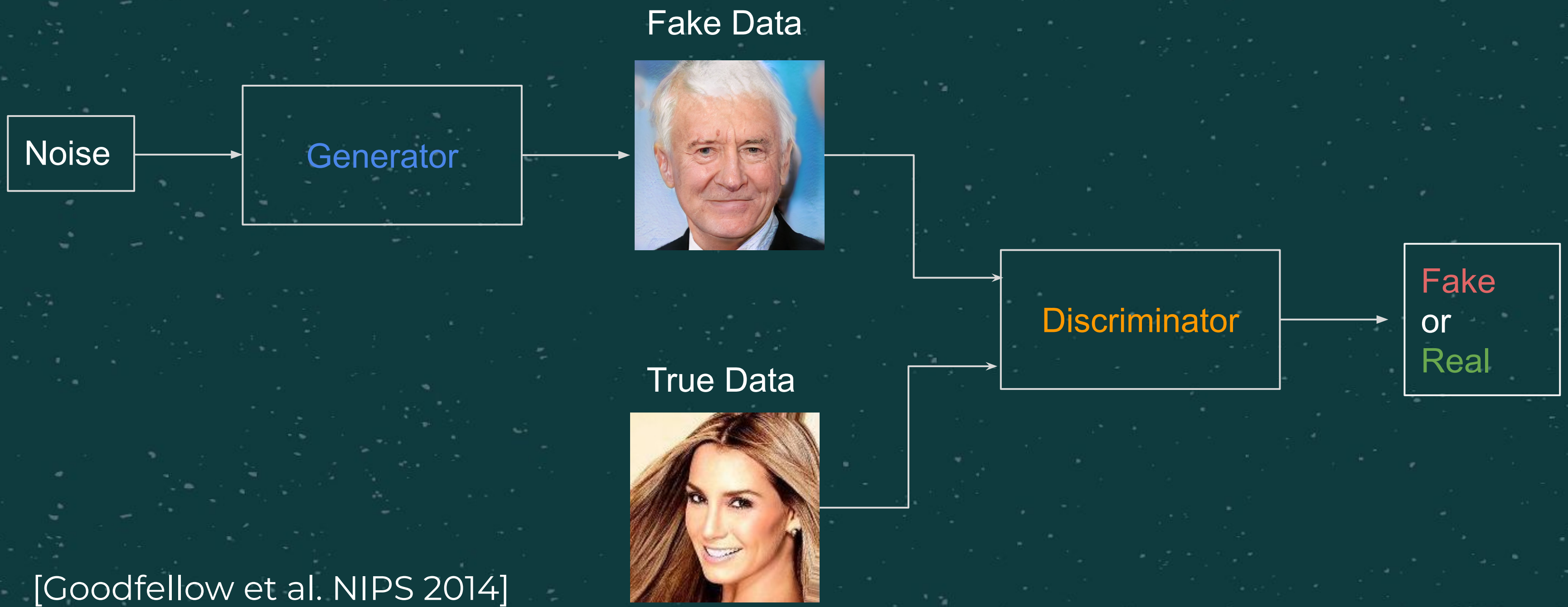
Idea: Transform an easy to sample distribution into something else:

$$x \sim p_\theta \Leftrightarrow x = g_\theta(z)\ ,\ z \sim p_z$$

Challenging to compute but easy to sample from

Usually Gaussian

GANs



Fake Data

Noise

Generator

Discriminator

Fake
or
Real

True Data

[Goodfellow et al. NIPS 2014]

GANs

Given p_g, binary classification with cross-entropy:

$$\mathbb{E}_{x \sim p_{data}}[\log(D(x))] + \mathbb{E}_{x_{fake} \sim p_g}[\log(1 - D(x_{fake}))]$$

Depends on G!!!

For a fixed Generator, the Discriminator wants to:

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

GANs

- Game!!!

$$\underbrace{\mathbb{E}_{x\sim p_{data}}[\log(D(x))] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]}_{\boxed{\text{D wants to maximize that while G want to minimize it}}}$$

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

GANs

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

Questions: (not easy)

1. Does this game have a pure Nash Equilibrium? (under which assumption)

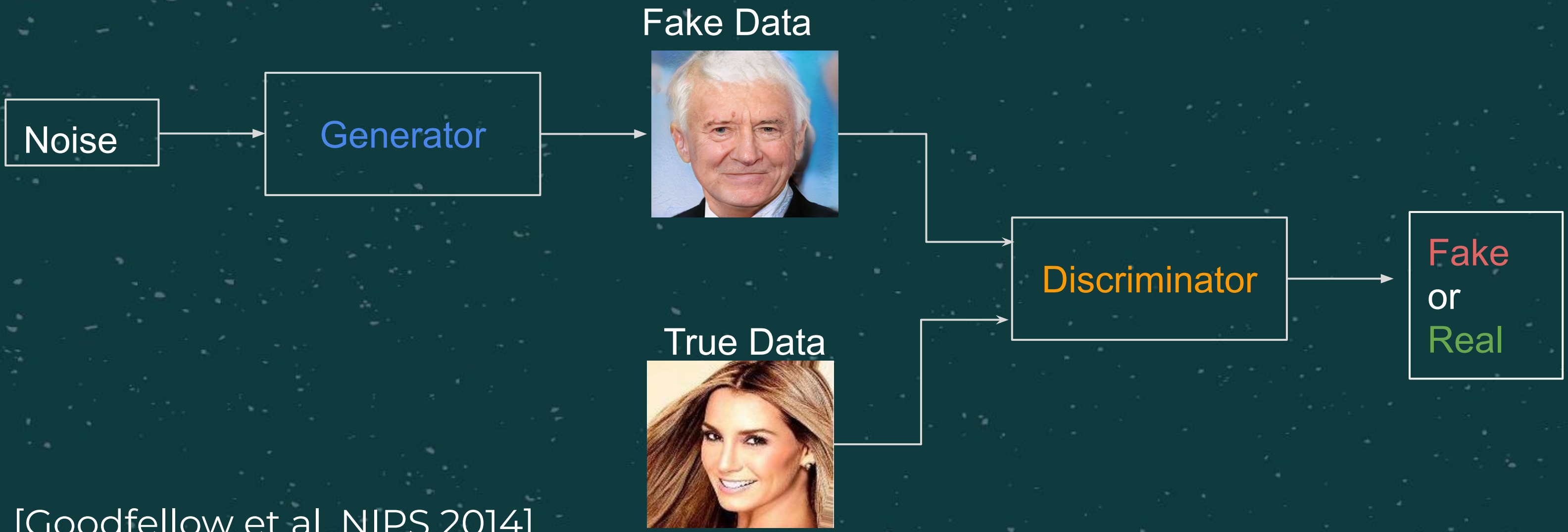
$$\varphi(D, G^*) \leq \varphi(D^*, G^*) \leq \varphi(D^*, G)\,, \quad \forall D, G$$

2. Do we have a simple formula for the Nash?

GAN: a Trivial Game?

- Exactly memorizing the train set is optimal for the Generator.

Question: Why it does not happen?



Noise

Generator

Fake Data

True Data

Discriminator

Fake
or
Real

[Goodfellow et al. NIPS 2014]

Non-Zero Sum Game

A different loss of each player:

$$\begin{cases} \min\limits_{D} -\mathbb{E}_{x\sim p_{data}}[\log(D(x))] - \mathbb{E}_{z\sim p_z}[\log(1-D(G(z)))] \\ \min\limits_{G} -\mathbb{E}_{z\sim p_z}[\log(D(G(z))] \end{cases}$$

Supposed to give stronger Gradient early in learning

V.s

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

Non-Zero Sum Game

A different loss of each player:

$$\begin{cases} \min\limits_{D} -\mathbb{E}_{x\sim p_{data}}[\log(D(x))] - \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))] \\ \min\limits_{G} -\mathbb{E}_{z\sim p_z}[\log(D(G(z)))] \end{cases}$$

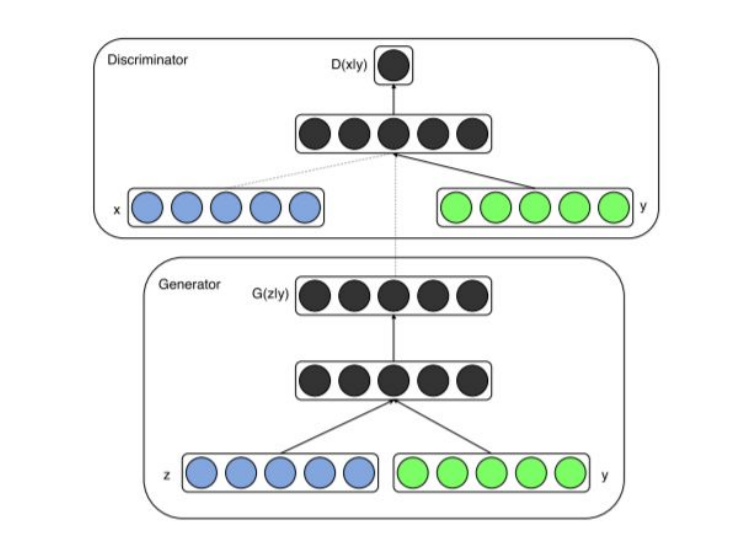
Question:
What is the Equilibrium of this new Game?

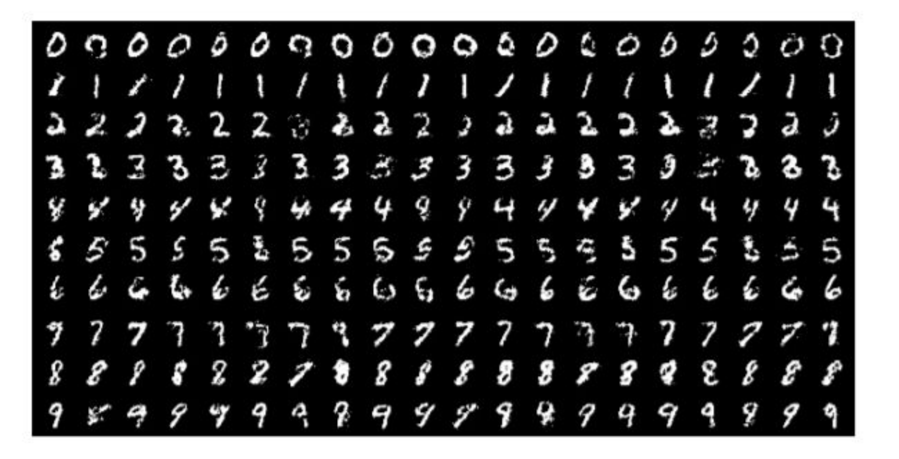
$$\min_{G} \quad {}_{D} \quad (z)))]$$

Issues with GAN Training

Conditional GANs

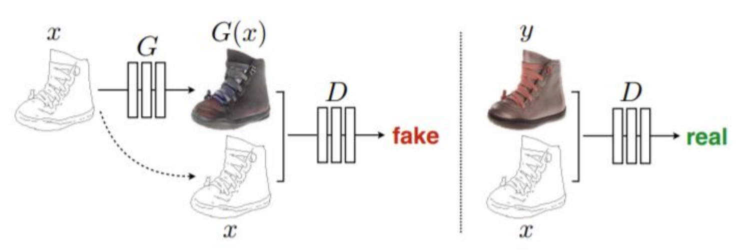
- Idea: condition the generator AND the discriminator with labels. [Mirza et al. 2014]

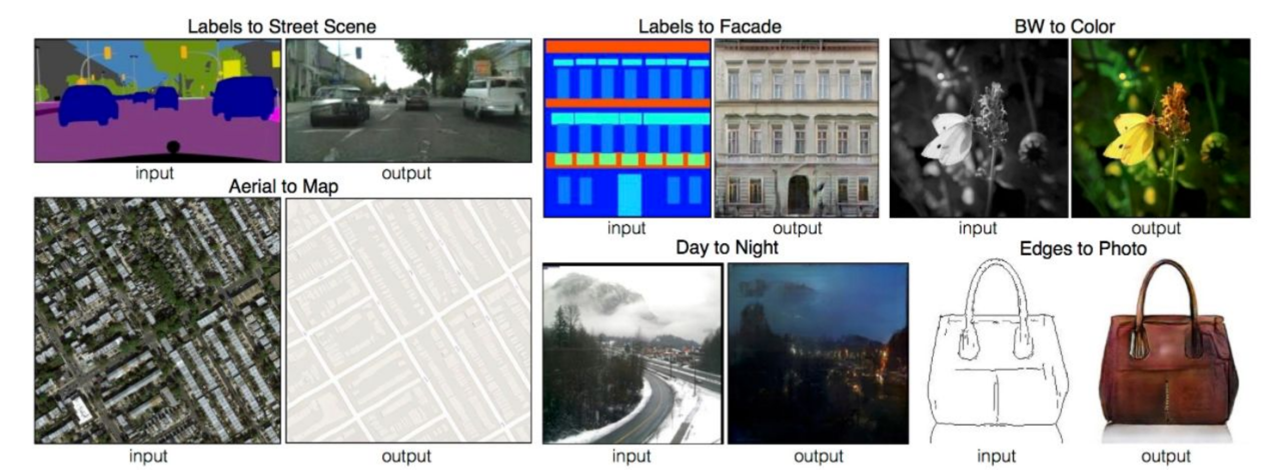




Conditional GANs

- Can condition on anything:
  (as long as we have data)

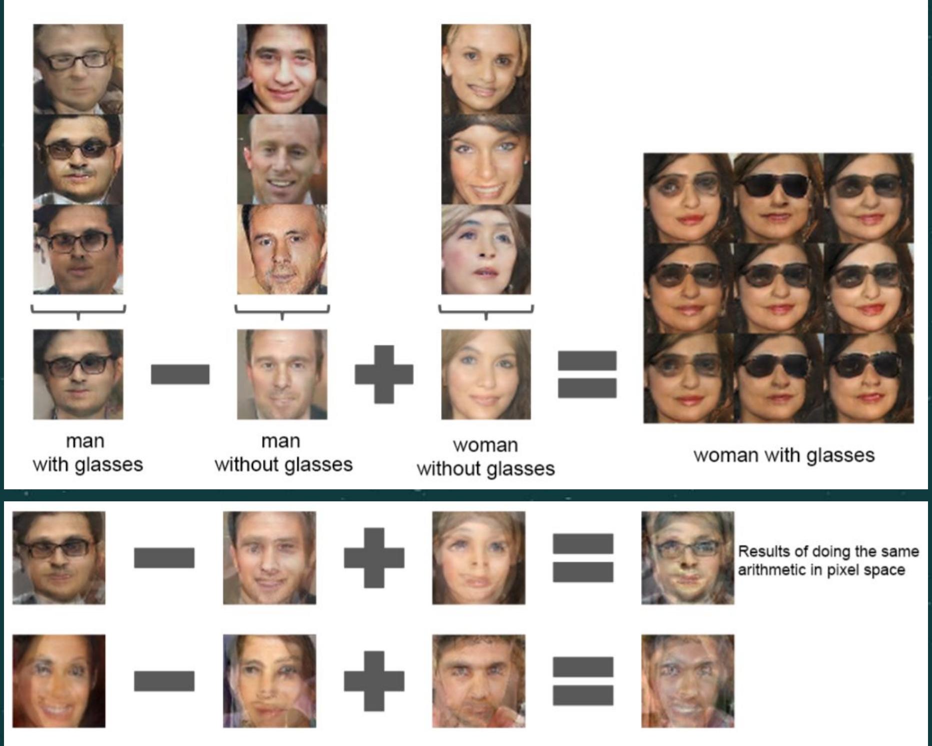




https://phillipi.github.io/pix2pix/ [Isola et al 2017]

Arithmetic in the Latent Space

- Initial idea from Radford et al [2016]

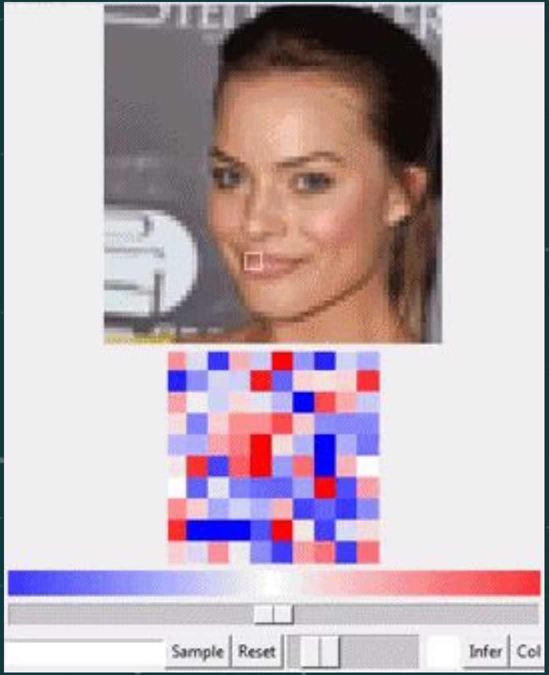


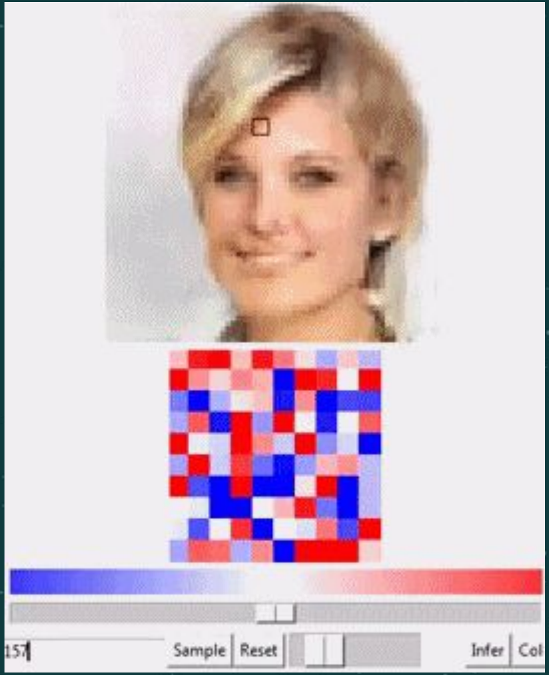
Arithmetic in latent space

Arithmetic in pixel space

Arithmetic in the Latent Space

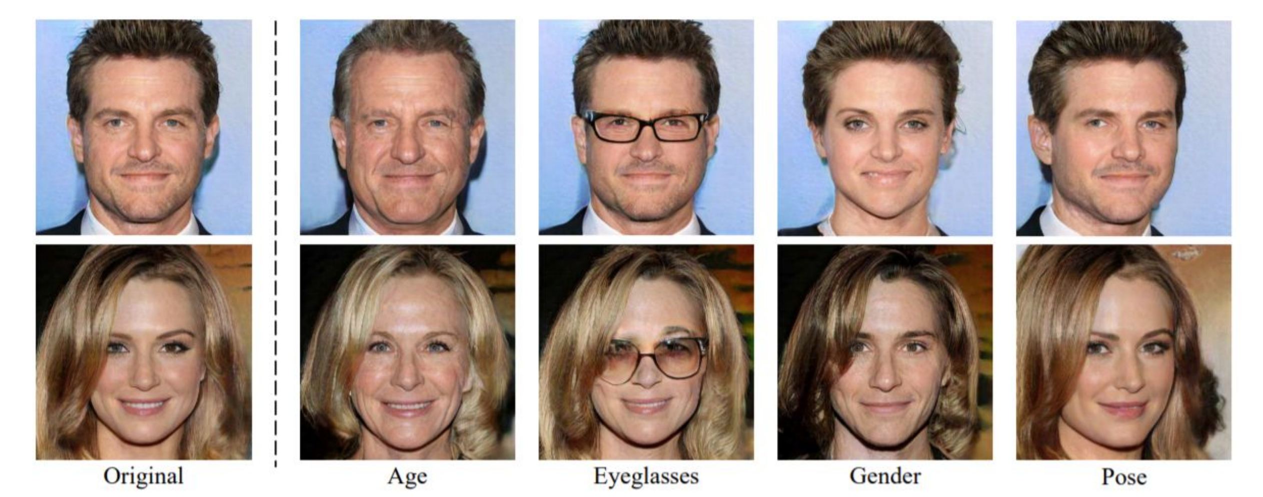
https://github.com/ajbrock/Neural-Photo-Editor (2016)





Arithmetic in the Latent Space

Idea: learn the "latent directions" of these features(Age, Eyeglasses, Gender, pose).



https://genforce.github.io/interfacegan/ (2020)