Introduction to GANs

Gokula Krishnan Santhanam
(ETH Zurich)

About Me

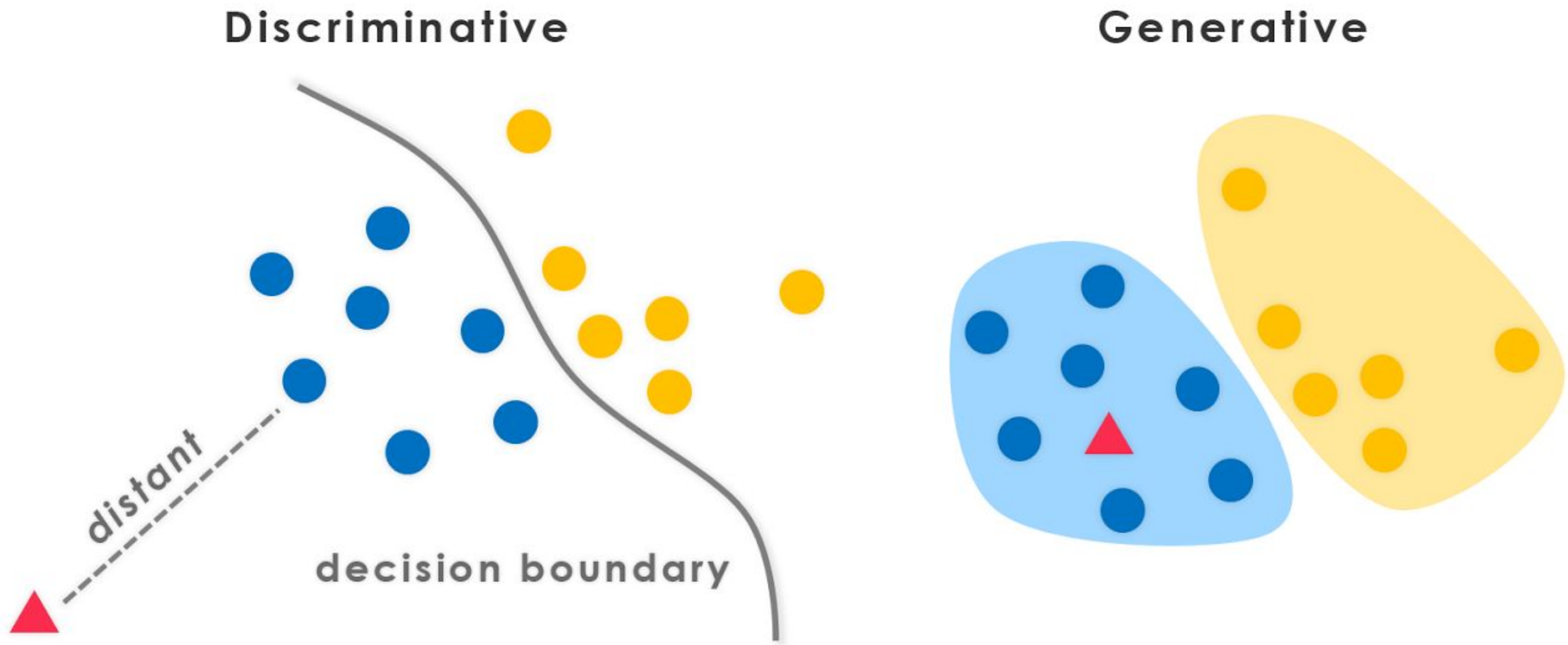
- Masters student @ ETH Zurich
- Google Research Intern
- Work on
  - GANs
  - Adversarial Attacks on CNNs
  - Hyperbolic DL
- space.ml
- Twitter: @gokstudio

Outline

- Introduction to Generative Modelling
- Introduction to GANs
- Known Issues
- Popular GAN Architectures
    - DCGAN
    - f-GAN
    - Pix2Pix / Cycle GAN
- Recent Developments (6 months to 1 year)
- Current Open Problems (under review)
- Q&A

Generative Models

Generative vs Discriminative Models



Why do we need generative models?

- Can do interesting things
  - Image denoising, image inpainting ...
- Fewer labelled data to train
  - Less cost and time
- Data generated from underlying distribution
  - find hidden parameters of that distribution

Common Generative Models

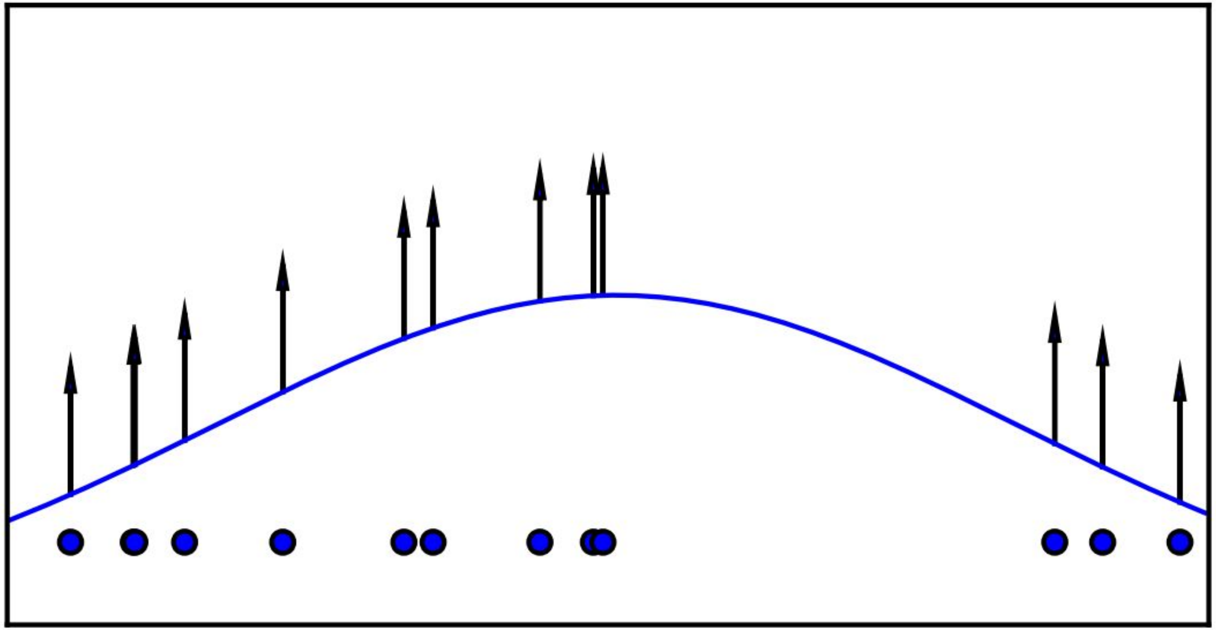
- Gaussian mixture models
- Hidden Markov model
- Naive Bayes
- Latent Dirichlet Allocation
- Variational Autoencoders (VAE)
- **Generative Adversarial Networks**
- ...

Difficulties

Can you look at a painting and recognize it as being the Mona Lisa? You probably can. That's discriminative modeling. Can you paint the Mona Lisa yourself? You probably can't. That's generative modeling.
-- Ian Goodfellow

- Difficult to design flexible models that can capture arbitrary data distributions

How to learn Generative models?



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

GANs

GANs

- Generative Adversarial Networks
- Involves two networks
  - Generator
  - Discriminator
- Min-Max game between the two

Adversarial?

- Two networks
- Try to best each other
- Zero-Sum Game

Nash Equilibrium in GANs

- Generator has learnt the data distribution
- Discriminator is confused
    - Can’t discriminate between generated data pts and data distribution
- Discriminator can effectively cover the data distribution

Caveat: Idealistic Scenario

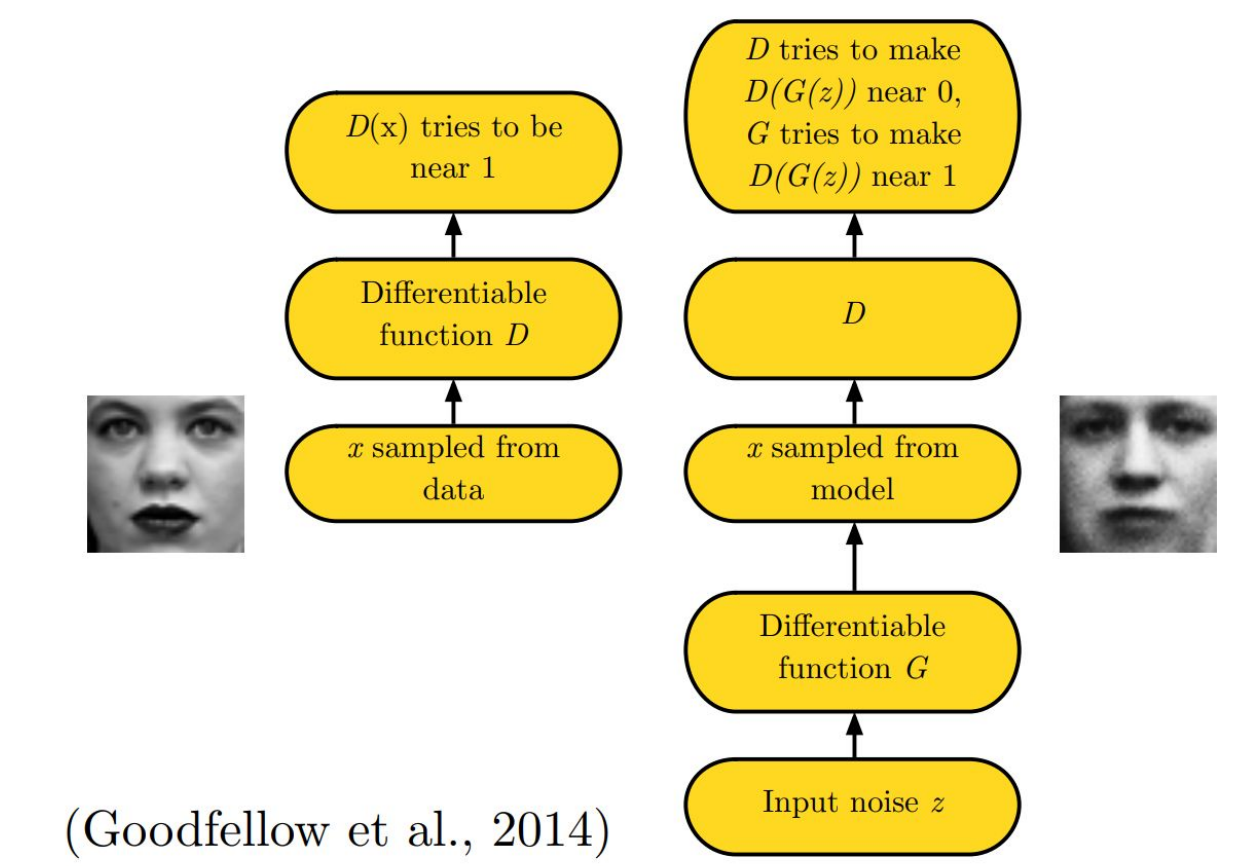
The Hello World equation of GANs

Generator minimizes the log-probability of the discriminator being correct

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

$$J^{(D)}(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)}) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

General GAN Architecture



The Training Process

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

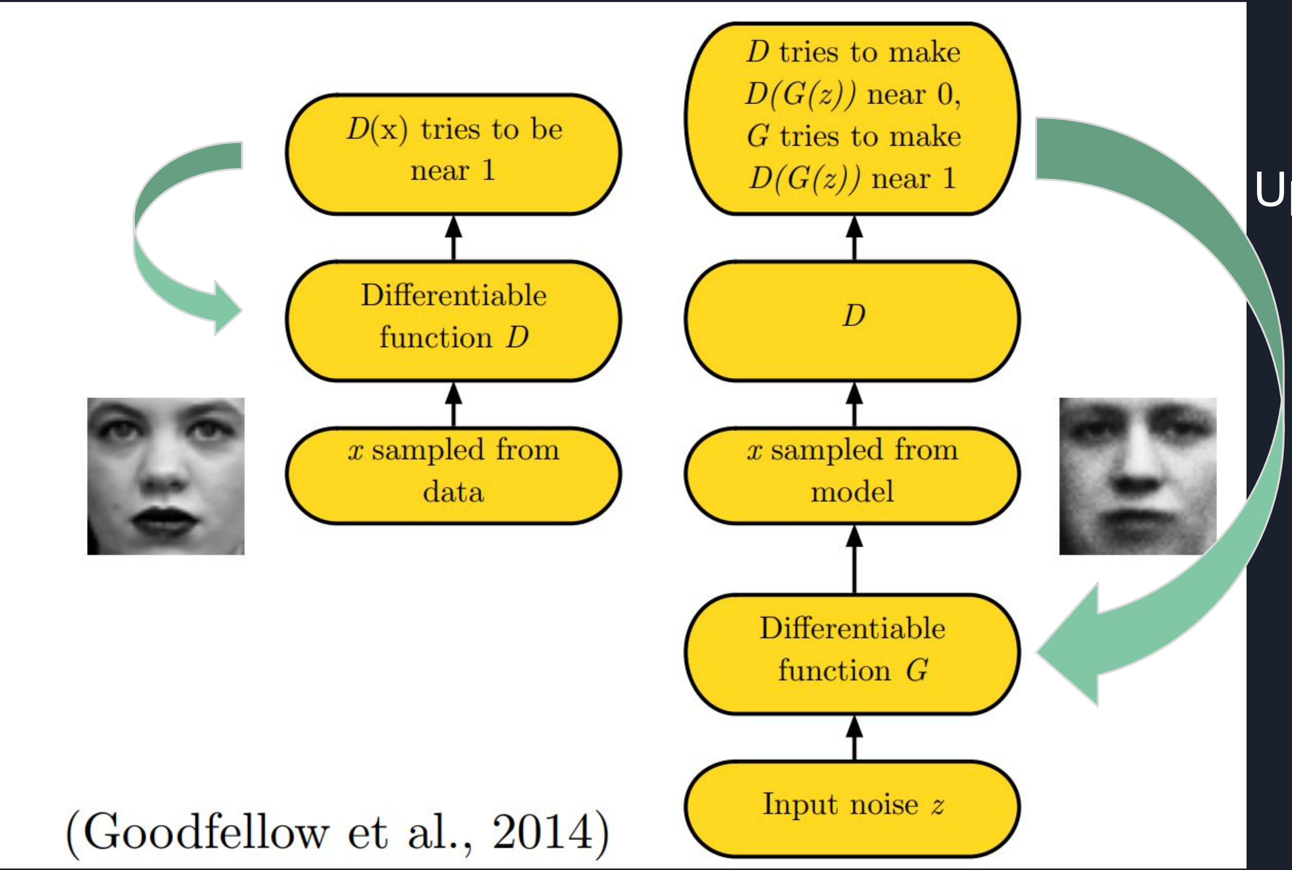
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

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

end for

Gradients

Updates to D



(Goodfellow et al., 2014)

Updates to G

How to update generator?

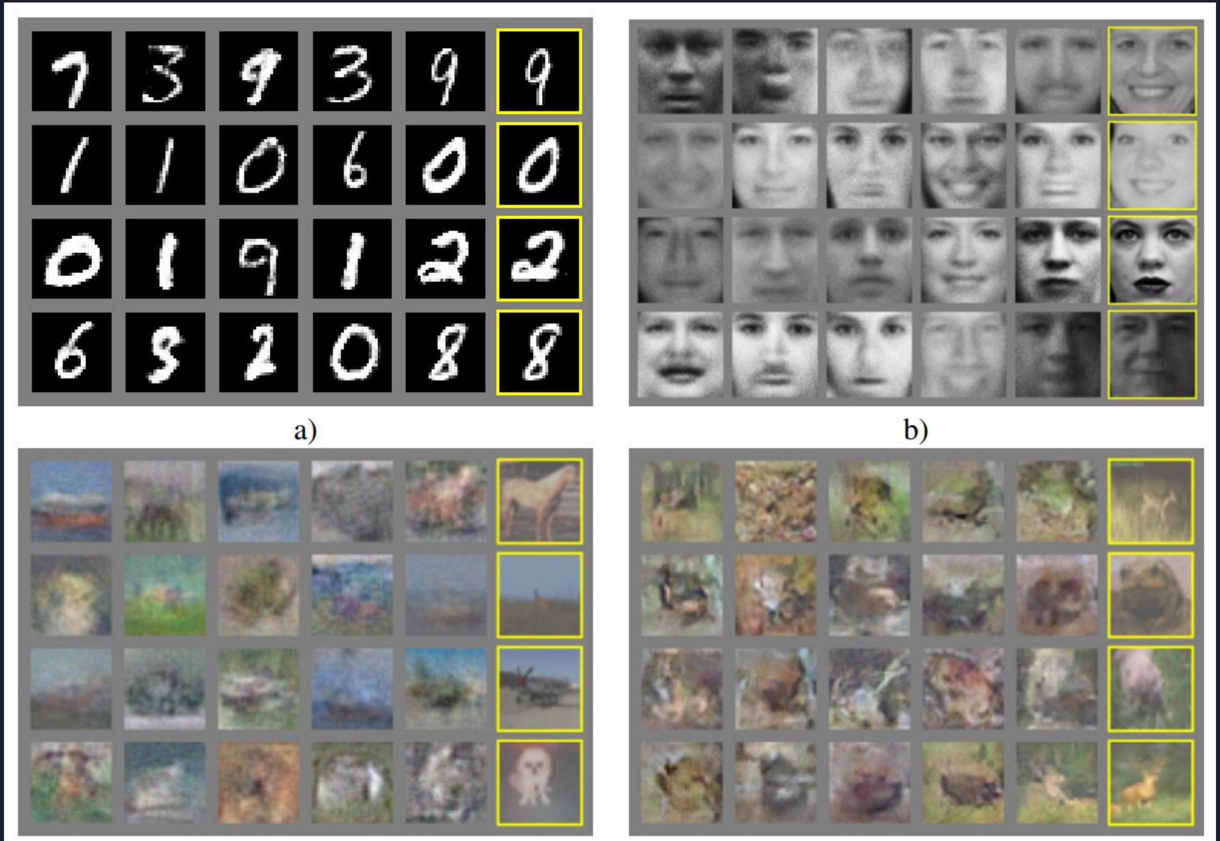
$$\textbf{Minimax} \quad J^{(G)}(G) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} \log[1 - D(G(\boldsymbol{z}))].$$

$$\textbf{Non-saturating} \quad J^{(G)}(G) = -\mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} \log D(G(\boldsymbol{z})).$$

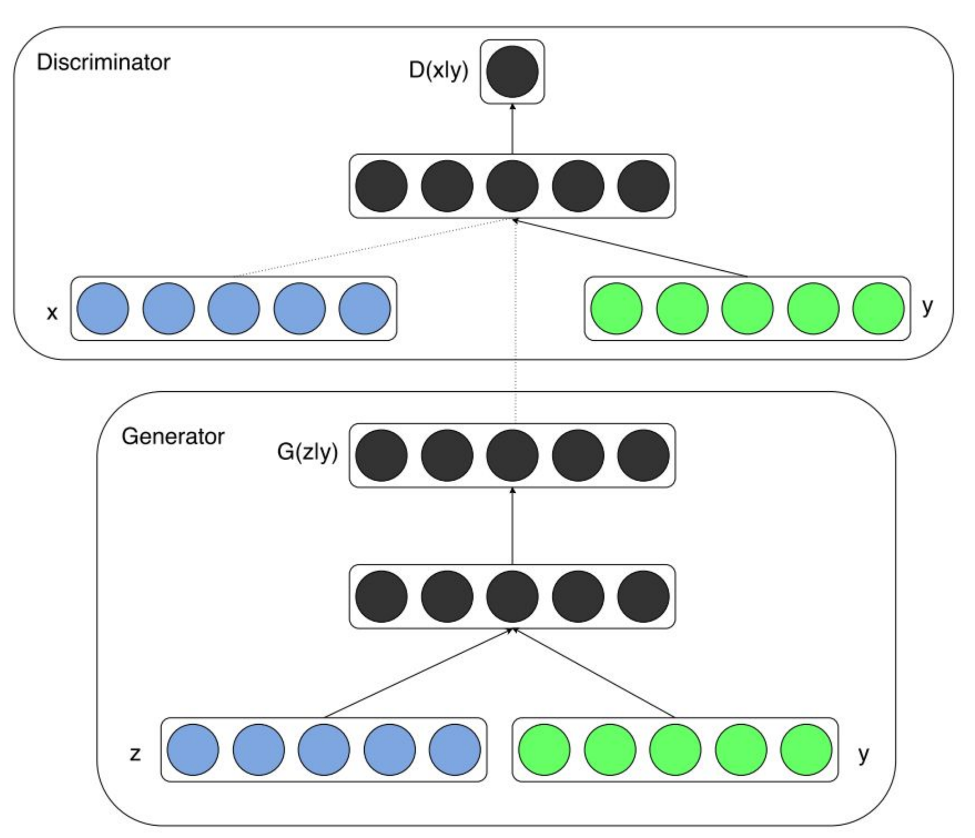
Quick Example in TensorFlow

```
#Inputs
images = tf.placeholder(shape = [None, 32, 32, 3], dtype = tf.float32)
noise = tf.placeholder(shape = [None, noise_dime], dtype = tf.float32)

gen_op = generator(noise)

real_logits = discriminator(images)
fake_logits = discriminator(gen_op)

d_loss_real = tf.softmax_cross_entropy_with_logits(logits = real_logits,
labels = tf.ones_like(real_logits)
d_loss_fake = tf.softmax_cross_entropy_with_logits(logits = fake_logits,
labels = tf.zeroes_like(real_logits)

D_loss = d_loss_real + d_loss_fake

G_loss = tf.softmax_cross_entropy_with_logits(logits = fake_logits, labels =
tf.ones_like(fake_logits))
```

Initial Results

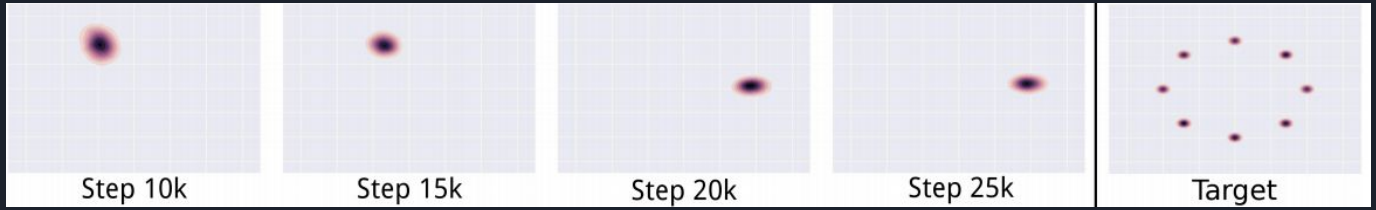


Conditional GANs

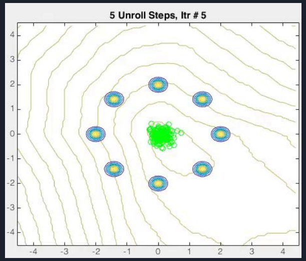


Are we done?

- Notoriously hard to train
- Mode collapse

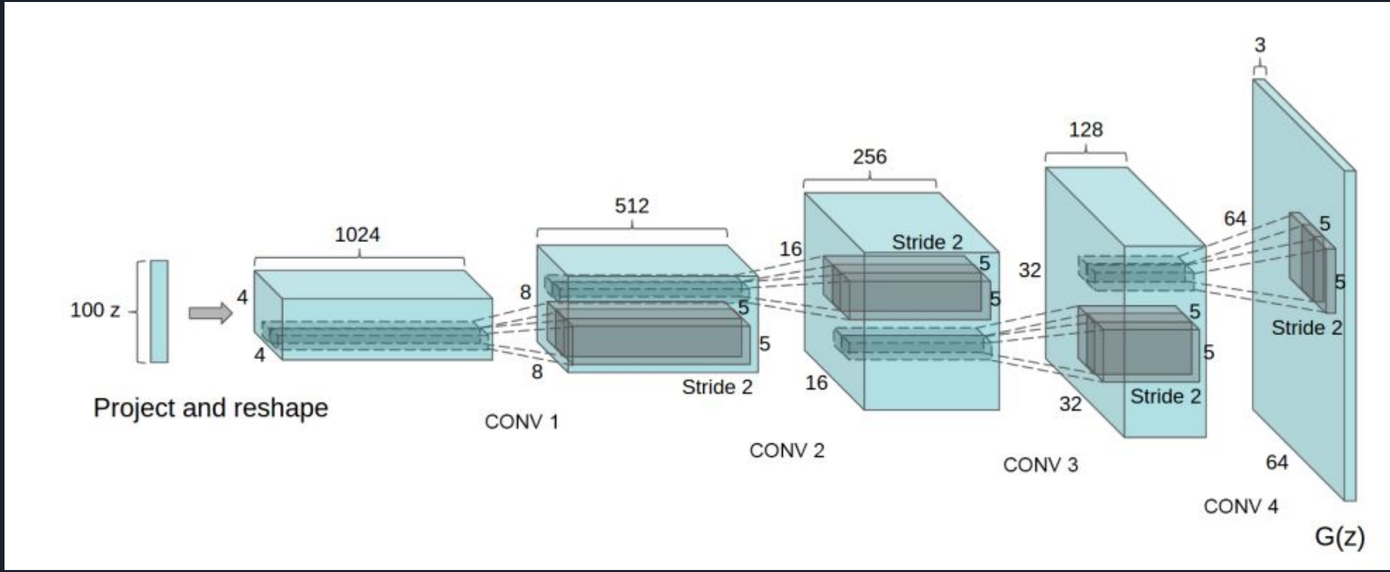


- Non-convergence



- Catastrophic degradation over prolonged training
- No objective performance metric

DCGANs

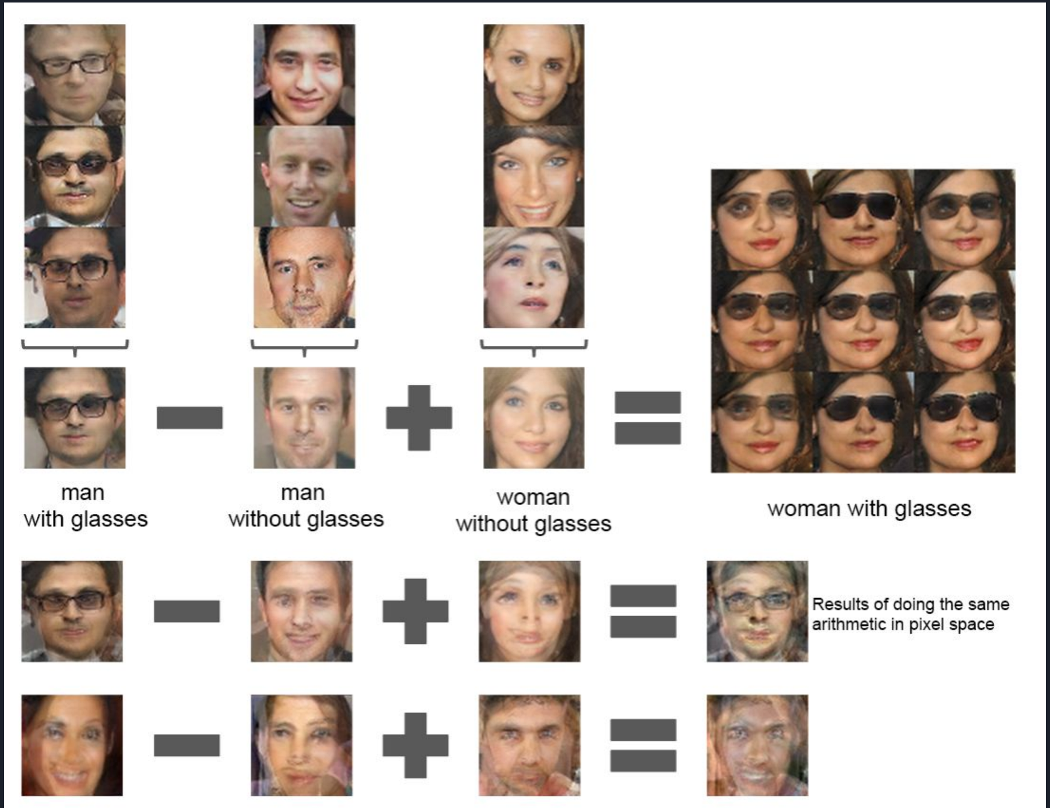


100 z
Project and reshape
1024
4
4
512
8
8
5
5
Stride 2
CONV 1
256
16
16
Stride 2
5
5
CONV 2
128
32
32
5
5
Stride 2
CONV 3
3
64
64
5
5
Stride 2
CONV 4
G(z)

Walking through the latent space

Face Pose modeled linearly in Latent Space

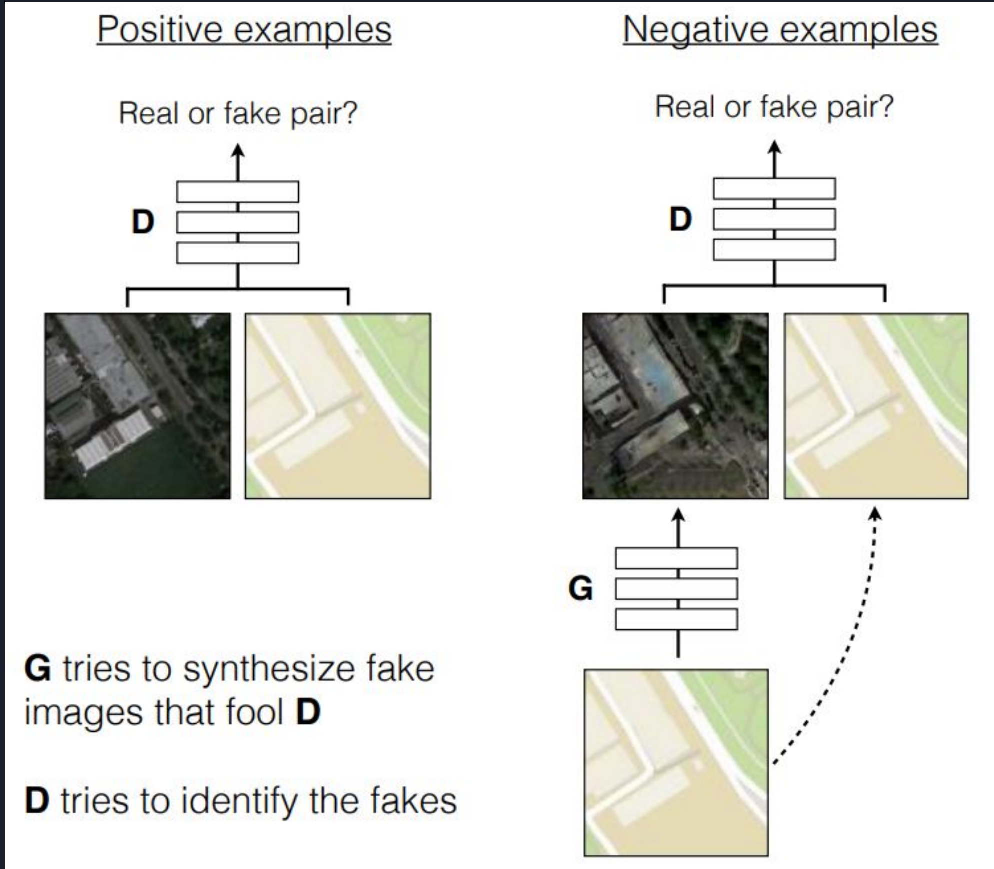
Vector Arithmetic on the Latent Space



The Divergence View

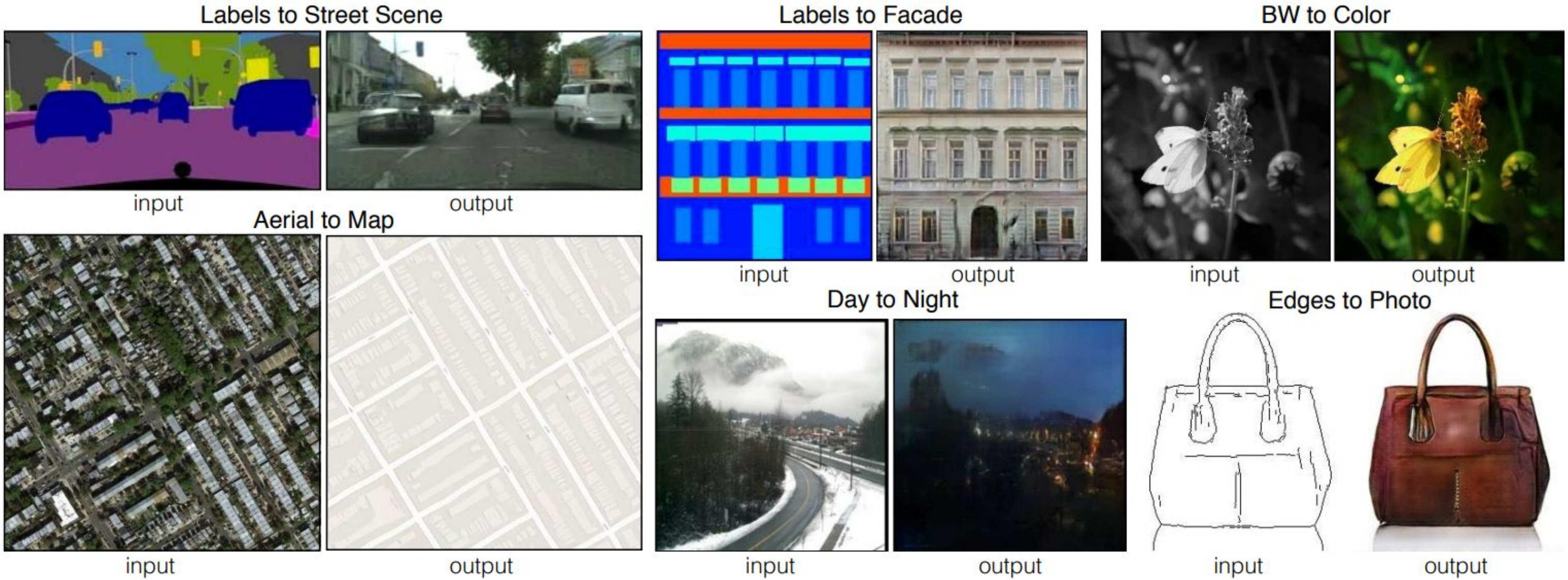
- Training reduces divergence between data distribution and Generator distribution
- Could be any divergence
  - Jensen Shannon divergence for WGAN
  - F-divergence for F-GANs
- Divergence minimized at convergence

Learning Image Transformations: Pix2Pix

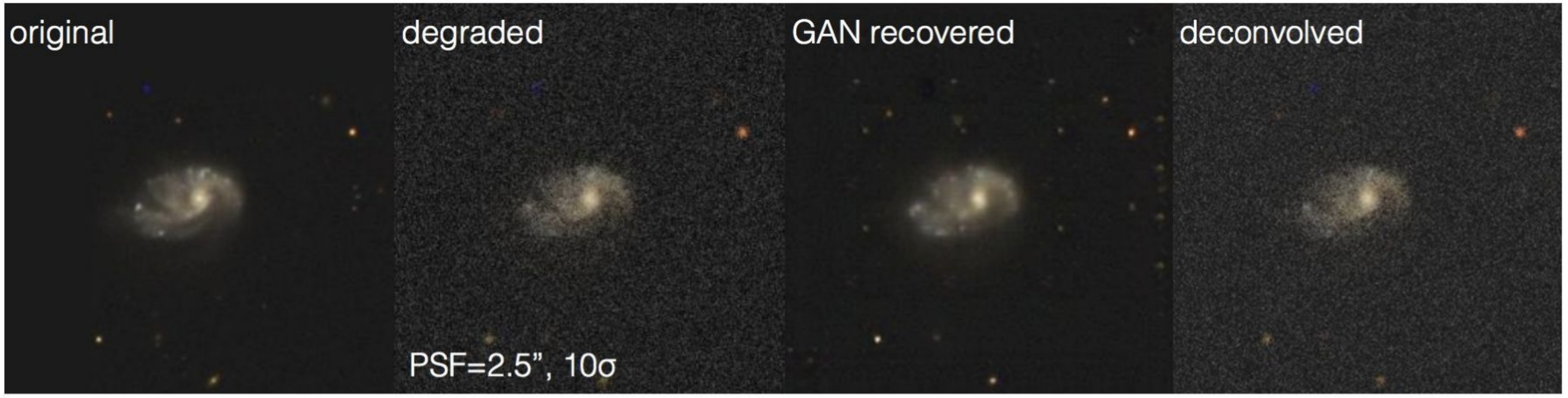


Cycle-GAN

Applications



GANs in Astronomy



Generative Adversarial Networks recover features in astrophysical images of galaxies beyond the deconvolution limit (https://arxiv.org/abs/1702.00403)

(Covered by WIRED, The Atlantic, Science, The Register, phys.org)

Recent Developments

Broad Themes

- Evaluating GANs
- Tackling failure modes
- Altering optimization process
- Application of GANs

Quantitative Metrics (Inception Score)

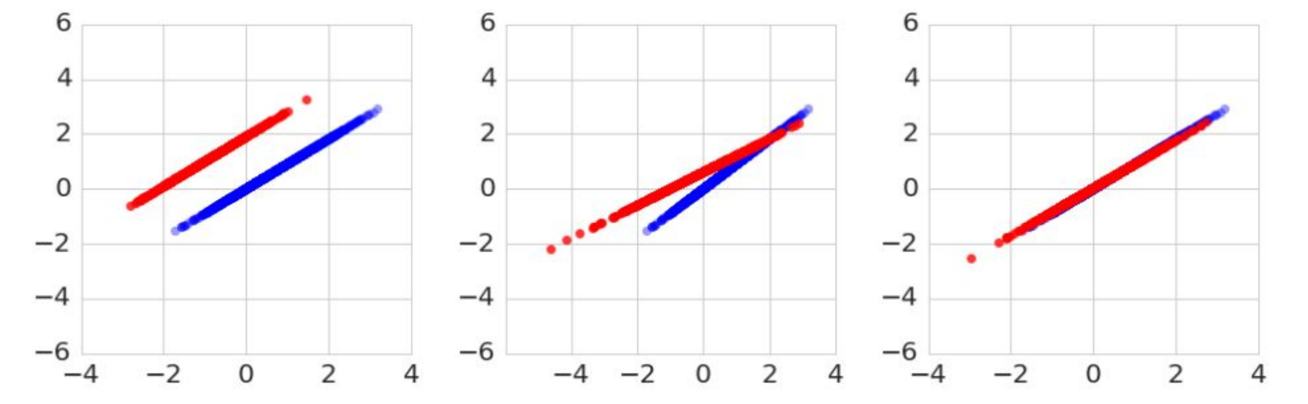
"My GAN's better than yours and I have numbers to prove it"

- Apply Inception model to generated images
- Get conditional label distribution $p(y|x)$
- Images with meaningful objects have conditional label distribution $p(y|x)$ with low entropy
- Higher is better

Do we need to reduce divergences?

Many Paths to Equilibrium (https://arxiv.org/abs/1710.08446)

- Show that divergence minimization is not the complete picture
- GANs converge even when divergence reduction says it shouldn't

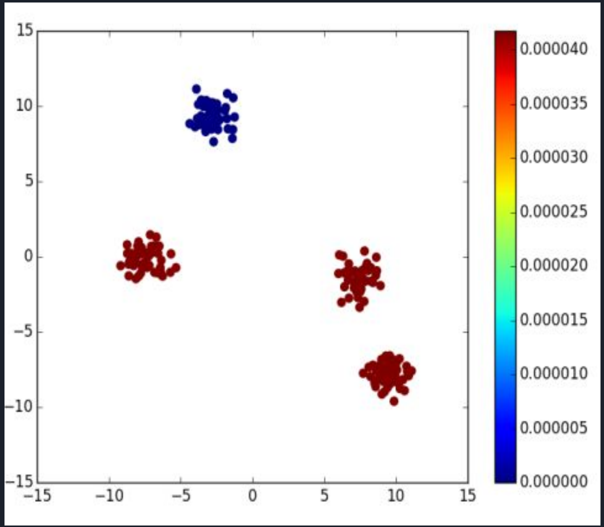


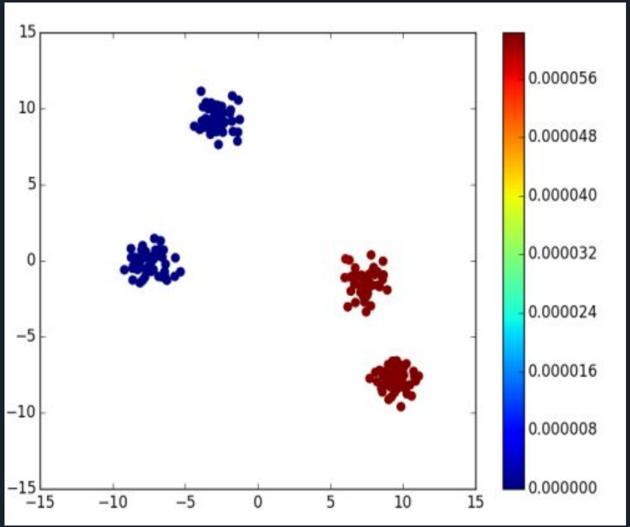
(a) Step 0 (b) Step 5000 (c) Step 12500

Progress in handling Mode-Collapse

Ada-GAN (https://arxiv.org/abs/1701.02386)

Like AdaBoost, ensemble of weak generators + reweighting of samples





Provable Convergence

- Cherkov GAN
  - GAN Training: find a mixed strategy in a zero-sum game.
  - Method provably converges to an equilibrium for semi-shallow GAN architectures

GANs for Text

- Not been applied to NLP
  - GANs are only defined for real-valued data
- All NLP based on discrete values (words, characters, or bytes)
  - no way to make a slight change.
- No one really knows how to apply GANs to NLP (yet)
- Could use the REINFORCE algorithm

Progressive GANs

High-resolution imag

Questions?

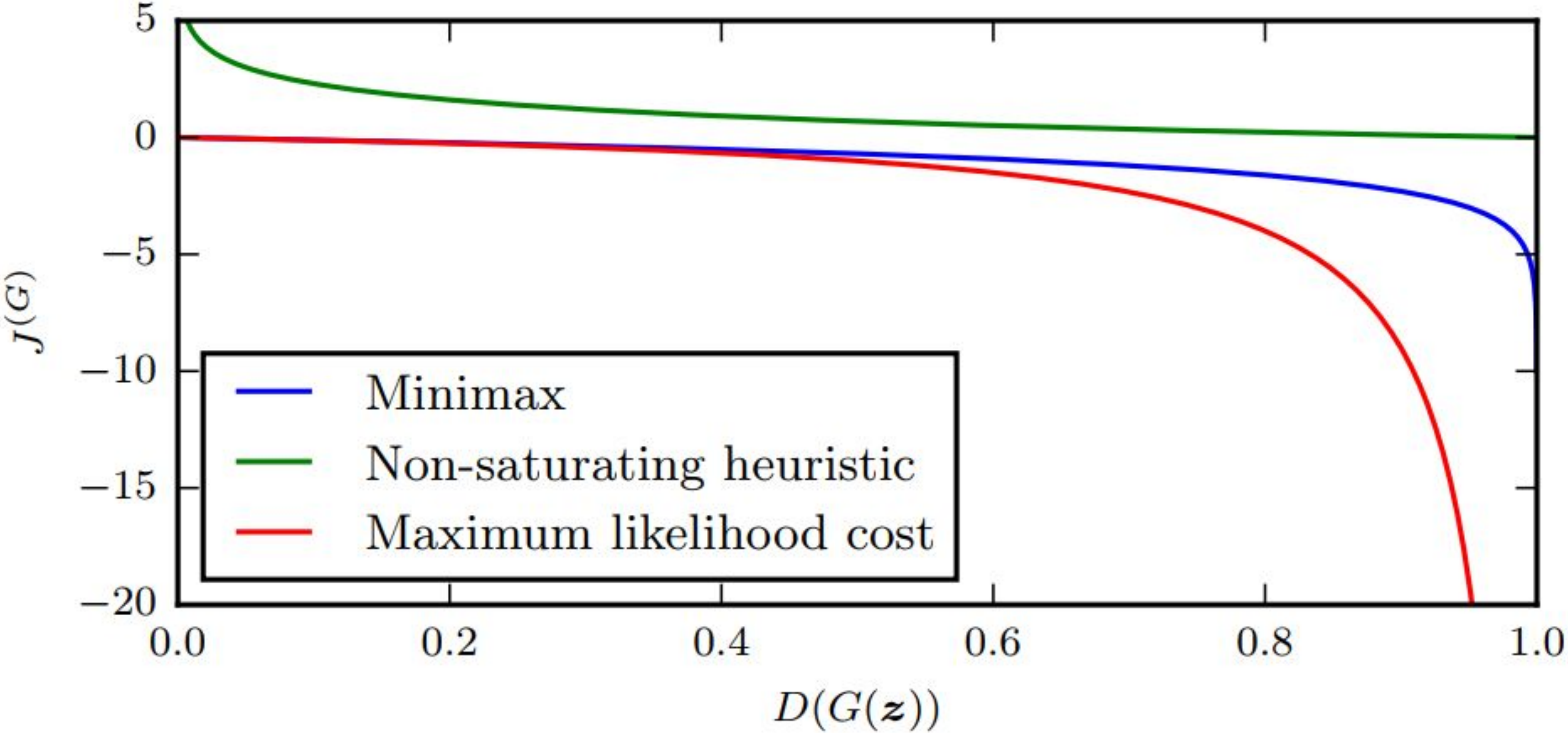
Thanks!

It’s Hot!

“

{ There are many interesting recent development in deep learning...The most important one, in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks). This, and the variations that are now being proposed, is the most interesting idea in the last 10 years in ML. }

Yann LeCun



5
0
−5
−10
−15
−20
$J(G)$
0.0
0.2
0.4
0.6
0.8
1.0
$D(G(\boldsymbol{z}))$
Minimax
Non-saturating heuristic
Maximum likelihood cost

f-GANs

Name	$D_f(P\|Q)$	Generator $f(u)$	$T^*(x)$
Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)}\, dx$	$u \log u$	$1 + \log \frac{p(x)}{q(x)}$
Reverse KL	$\int q(x) \log \frac{q(x)}{p(x)}\, dx$	$-\log u$	$-\frac{q(x)}{p(x)}$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)}\, dx$	$(u-1)^2$	$2(\frac{p(x)}{q(x)} - 1)$
Squared Hellinger	$\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx$	$(\sqrt{u} - 1)^2$	$(\sqrt{\frac{p(x)}{q(x)}} - 1) \cdot \sqrt{\frac{q(x)}{p(x)}}$
Jensen-Shannon	$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)}\, dx$	$-(u+1) \log \frac{1+u}{2} + u \log u$	$\log \frac{2p(x)}{p(x)+q(x)}$
GAN	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)}\, dx - \log(4)$	$u \log u - (u+1) \log(u+1)$	$\log \frac{p(x)}{p(x)+q(x)}$

$$D_f(P\|Q) = \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

$$T^*(x) = f'\left(\frac{p(x)}{q(x)}\right)$$

(tight variational lower-bound)

What's next?

- Applications
  - Healthcare
  - Drug discovery
  - Malware detection
  - Program synthesis
- Core research directions
  - Convergence bounds
  - Explanation for failure modes
  - Divergence view is incomplete, what's missing?
  - GANs for discrete data

How to get started?

- The original GAN paper
- GAN tutorial by Ian Goodfellow
- Soumith Chintala’s GAN Hacks (https://github.com/soumith/ganhacks)
- DCGAN implementation (https://github.com/gokul-uf/TF-DCGAN)
- Toy Examples - Mixture of Gaussians
- WGAN / f-GAN / WGAN-GP
- ICLR / NIPS / ICML / CVPR / ICCV

The Criminal Police Game

- Counterfeiters tries to counterfeit currency notes
- Police tries to catch them
- Police crack down on counterfeiting
- Counterfeiters improves the quality of counterfeits
- Police improve their detection techniques
- ...

Game Theory 101

Definition

- Study of mathematical models of conflict and cooperation between intelligent rational decision-makers
- Game: Situations in which at least one player can only act to maximize his utility through anticipating the responses to his actions by one or more other players
- Players: Individual rational decision-makers
  - Could be
    - Individuals (Resource Sharing)
    - Corporations (Free Markets)
    - Nation States (Diplomacy)

Nash Equilibrium

- Solution of a non-cooperative game involving two or more players in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only his own strategy.
- Mutual assured destruction
  - A full-scale use of nuclear weapons by two or more opposing sides would cause the complete annihilation of both the attacker and the defender

Back to GANs

Applications

- In Physics
  - Learning Particle Physics by Example
  - Simulating 3D High Energy Particle Showers
- In Chemistry
  - Generate compounds with certain attributes
  - https://chemrxiv.org/articles/ORGANIC_1_pdf/5309668
- In Biology
  - Cleaning Cryo-EM images
  - DeepCancer: Detecting Cancer through Gene Expressions via Deep Generative Learning
  - Medical Image Synthesis
- In medicine
  - Generating “fake” patient records