McGill Artificial Intelligence Society

Lecture 7: Generative Adversarial Nets

Slides ~~ripped off~~ inspired by Ian Goodfellow's NIPS 2016 GAN tutorial

Announcements

Exec applications are out! Go Apply!

Flask Workshop for Final Project Deliverables

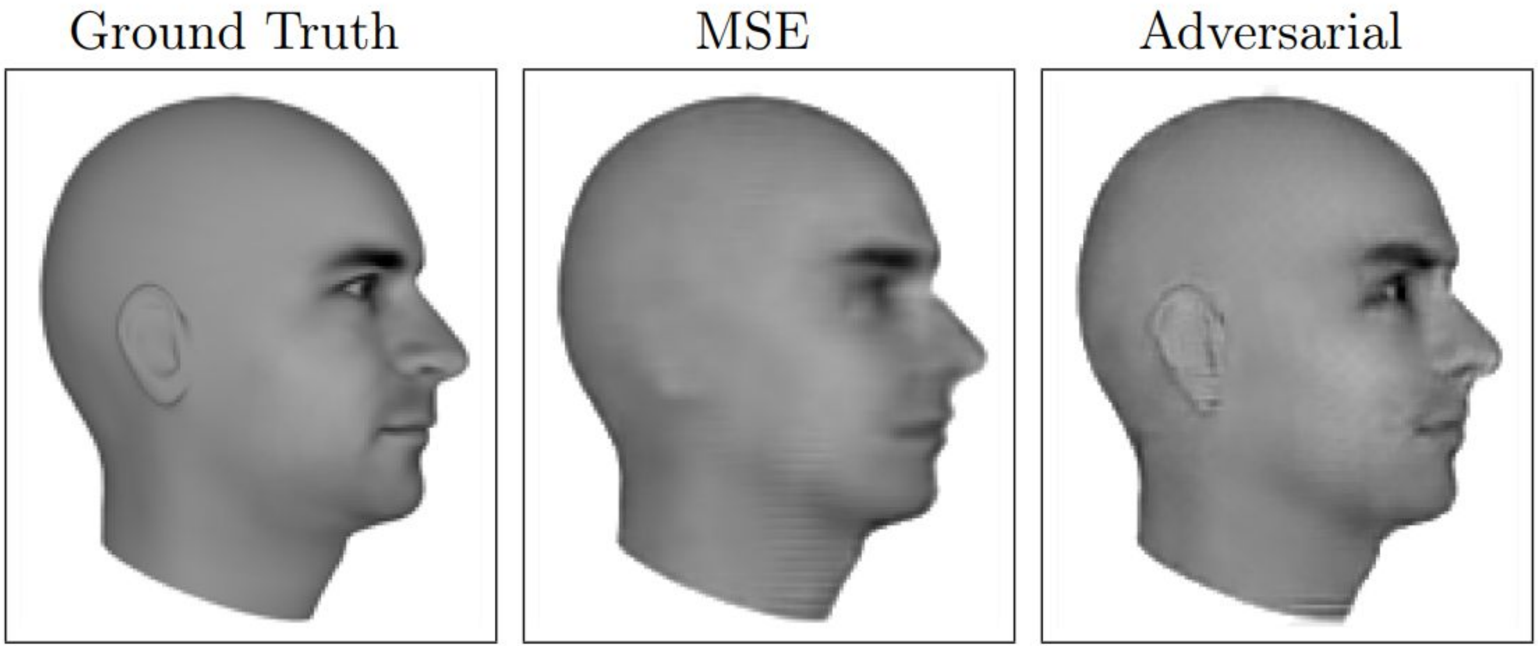
Today’s Lesson Plan

1. Why Study Generative Modelling?
2. Motivating Examples
3. How does Generative Modelling Work?
4. Tips And Tricks
5. Research Frontiers
6. Combining GANs with Other Methods

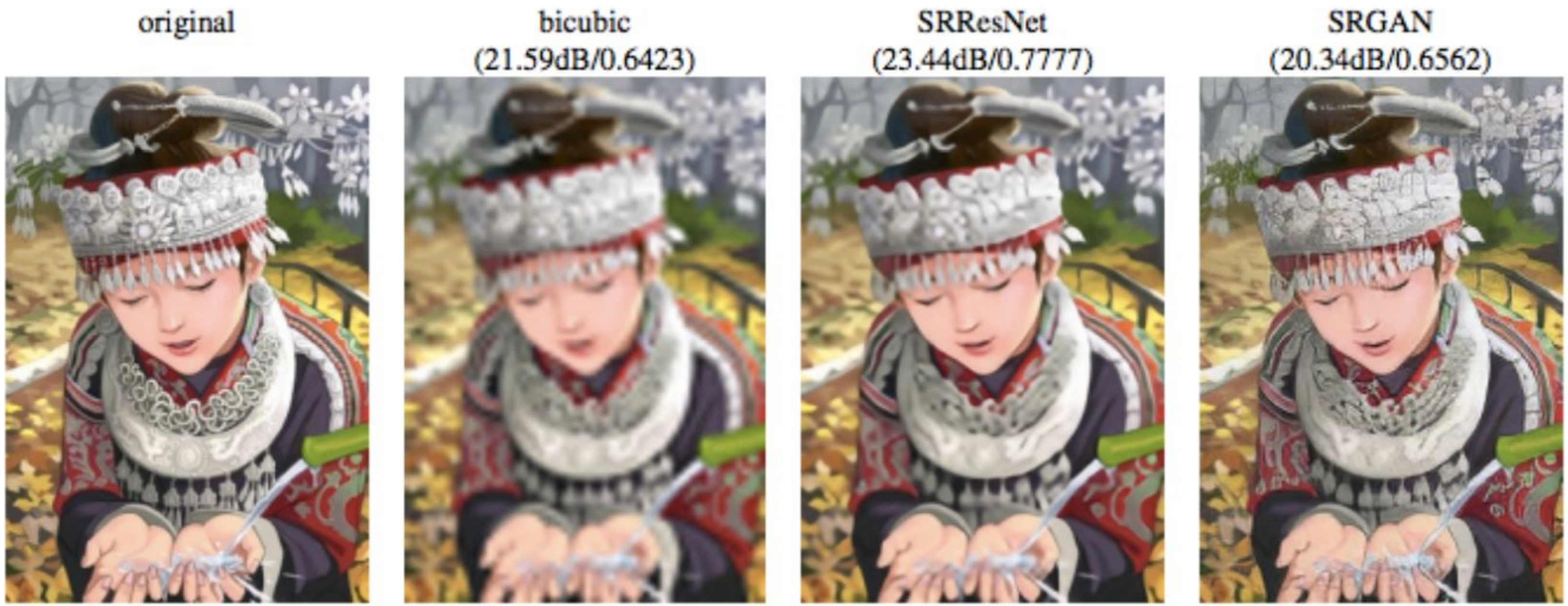
Why Study Generative Models

- A way for us to verify that deep learning does indeed learn accurate representations (and to see where it fails)
- Simulate possible features for planning (especially in RL - we'll see why this doesn't work next lecture)
- Missing Data
  - Semi-supervised learning
- Multi-modal outputs
- Realistic generation tasks

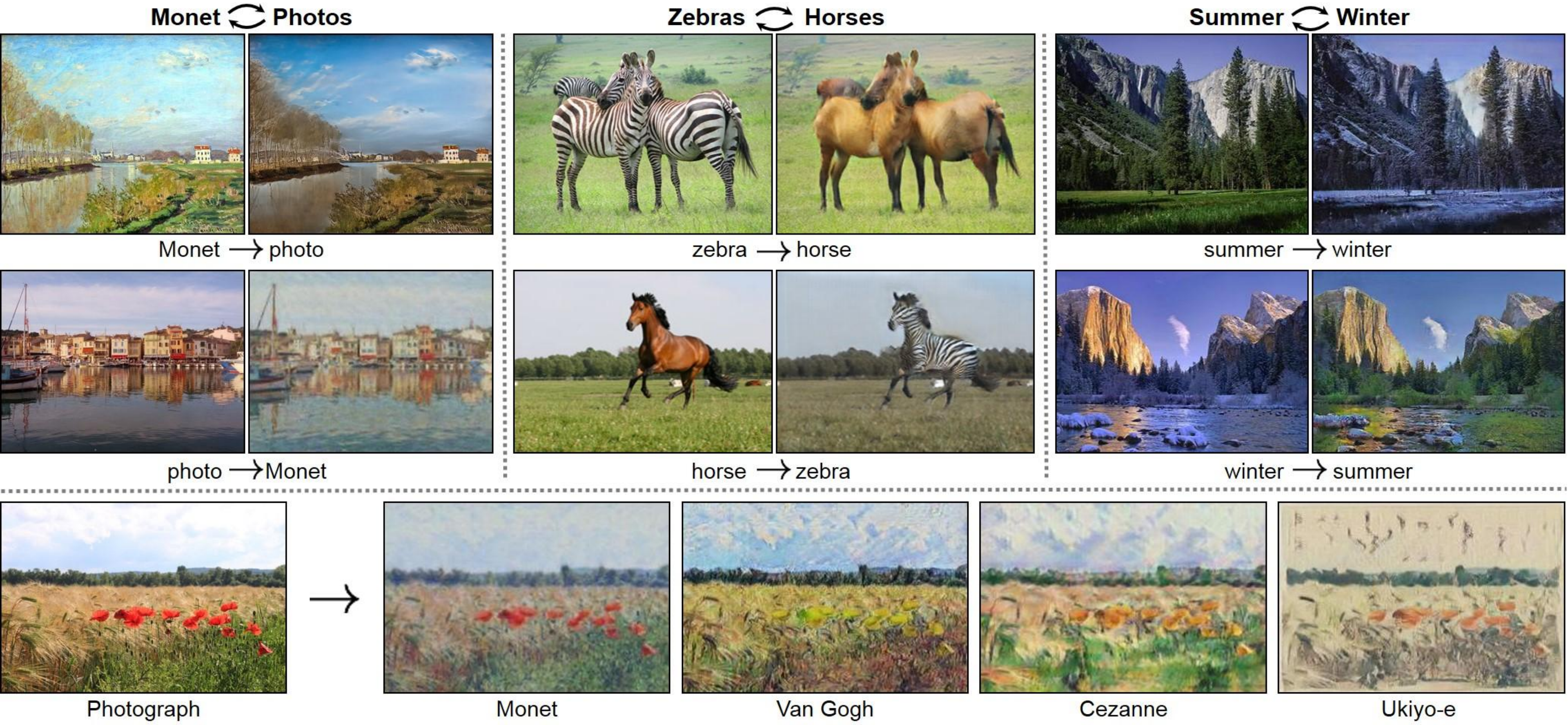
Rotating Heads



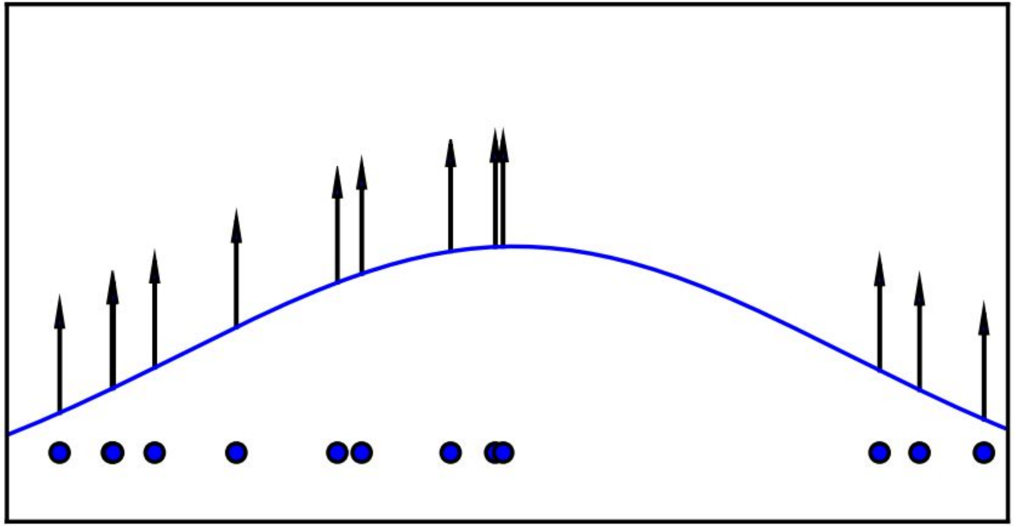
Single Image Super-Resolution



Why Study Generative Models

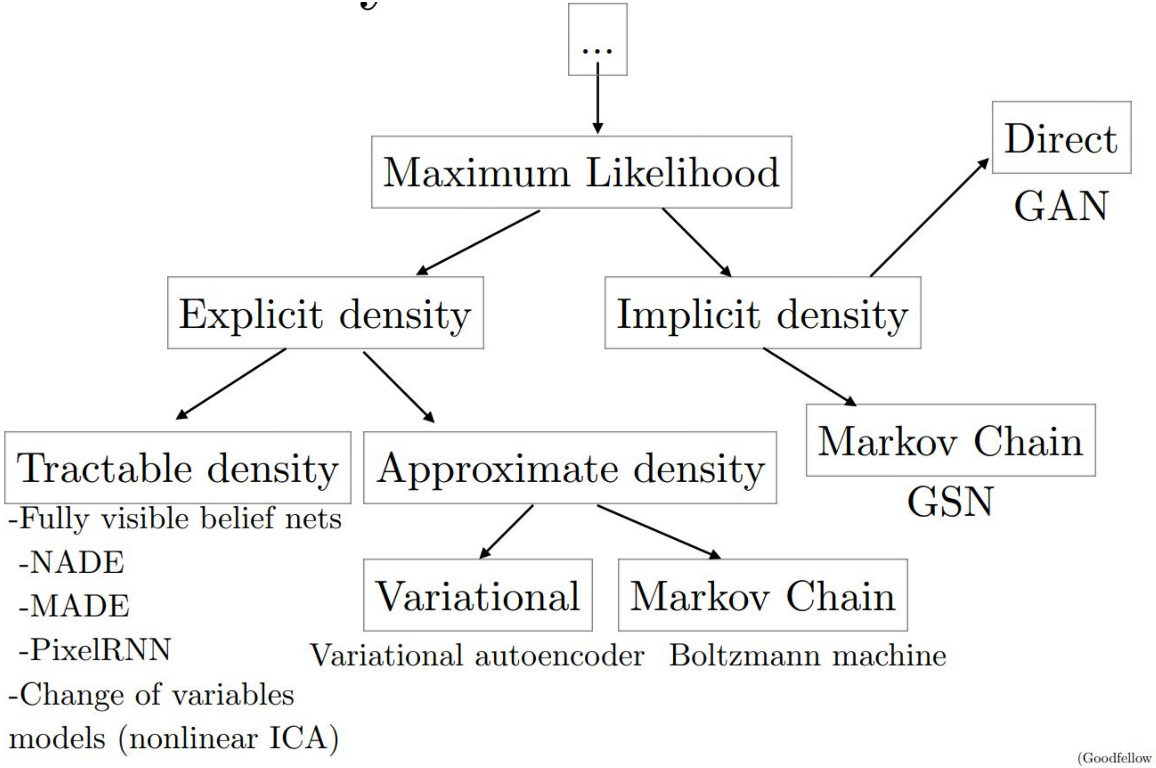

Monet Photos
Zebras Horses
Summer Winter
Monet → photo
zebra → horse
summer → winter
photo →Monet
horse → zebra
winter → summer
Photograph
Monet
Van Gogh
Cezanne
Ukiyo-e

Maximum Likelihood Operation



$$\theta^* = \underset{\theta}{argmax}\, \mathbb{E}_{x \sim p_{data}} \log p_{model}(x|\theta)$$

Taxonomy of Generative Models



(Goodfellow 2016)

Fully Visible Belief Nets

Explicit Formula Based on PGM

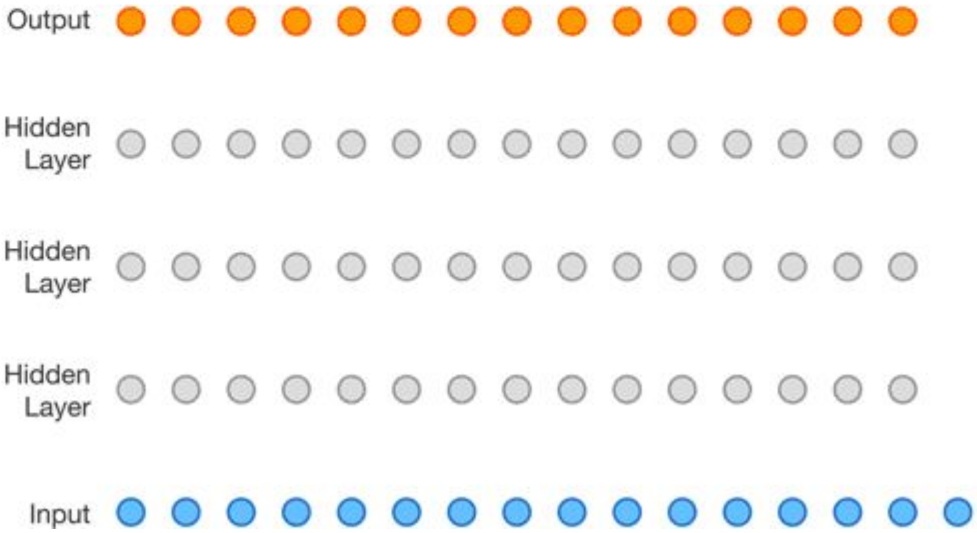
$$p_{model}(x) = p_{model}(x_1)\Pi_{i=2}^{n} p_{model}(x_i | x_1, \ldots, x_{i-1})$$

Disadvantages:

- O(n) sample time
- Generation controlled by latent code

Generated by pixelCNN

WaveNet



Super High Quality Generation, but takes 2 minutes to generate a second of audio

Change of Variables

$$y = g(x) \rightarrow p_x(x) = p_y(g(x))|det\frac{\delta g(x)}{\delta x}|$$

E.g. change of variables

Disadvantages

- Transformation must be invertible
- Latent Dimension must match Visible Dimension

Variational Autoencoder

$$\log p(x) \geq D_{KL}(q(z)||p(z|x) = \mathbb{E}_{z\sim q}\log(p(x,z)) + H(q)$$

Does use a latent code to generate samples

Disadvantages

- Asymptotically inconsistent unless q is perfect - cannot be guaranteed
- In practice, samples generated are worse

CIFAR-10 Samples
(Kingma 2016)

Boltzmann Machines

$$p(x) = \frac{1}{Z} exp(-E(x, z))$$

$$Z = \sum_x \sum_z exp(-E(x, z))$$

Disadvantages

- Relies on Monte Carlo Methods to approximate partition function
- Only perform well on small datasets like MNIST but don’t scale

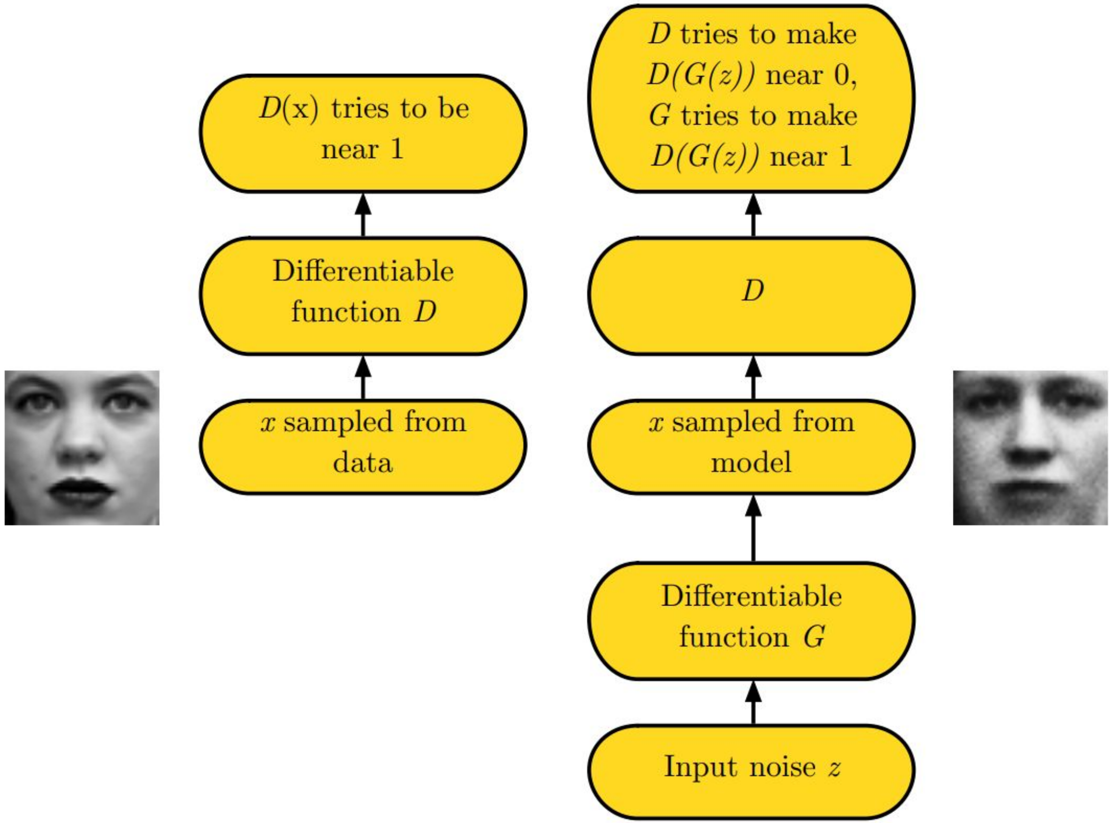
Why Use GANs

- Does use a latent code to generate samples
- Asymptotically Consistent
- No Markov Chains Needed

Break

Break

How do Adversarial Nets Work?



Generator Network

$$x = G(z; \theta^{(G)})$$

- Must be differentiable
- No invertibility requirement
- Trainable for any size of z
- Some guarantees require x to have higher dimension than z

Training Procedure

- Use SGD-esque algorithm on two minibatches simultaneously
    - Minibatch of training examples
    - Minibatch of test examples

- Optional: Run k-steps of one player per step of the other player

Minimax Game

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

$$J^{G} = -J^{D}$$

- Equilibrium is at saddle point of discriminator loss
- Resembles Jensen-Shannon divergence
- Generator minimizes log probability of discriminator being correct

Discriminator Strategy

Optimal D(x) for any p_{model} and p_{data} is $D^*(x) = \frac{p_{data}}{p_{model} + p_{data}}$

- Typically estimate this with a supervised learning procedure

Non-Saturating Game

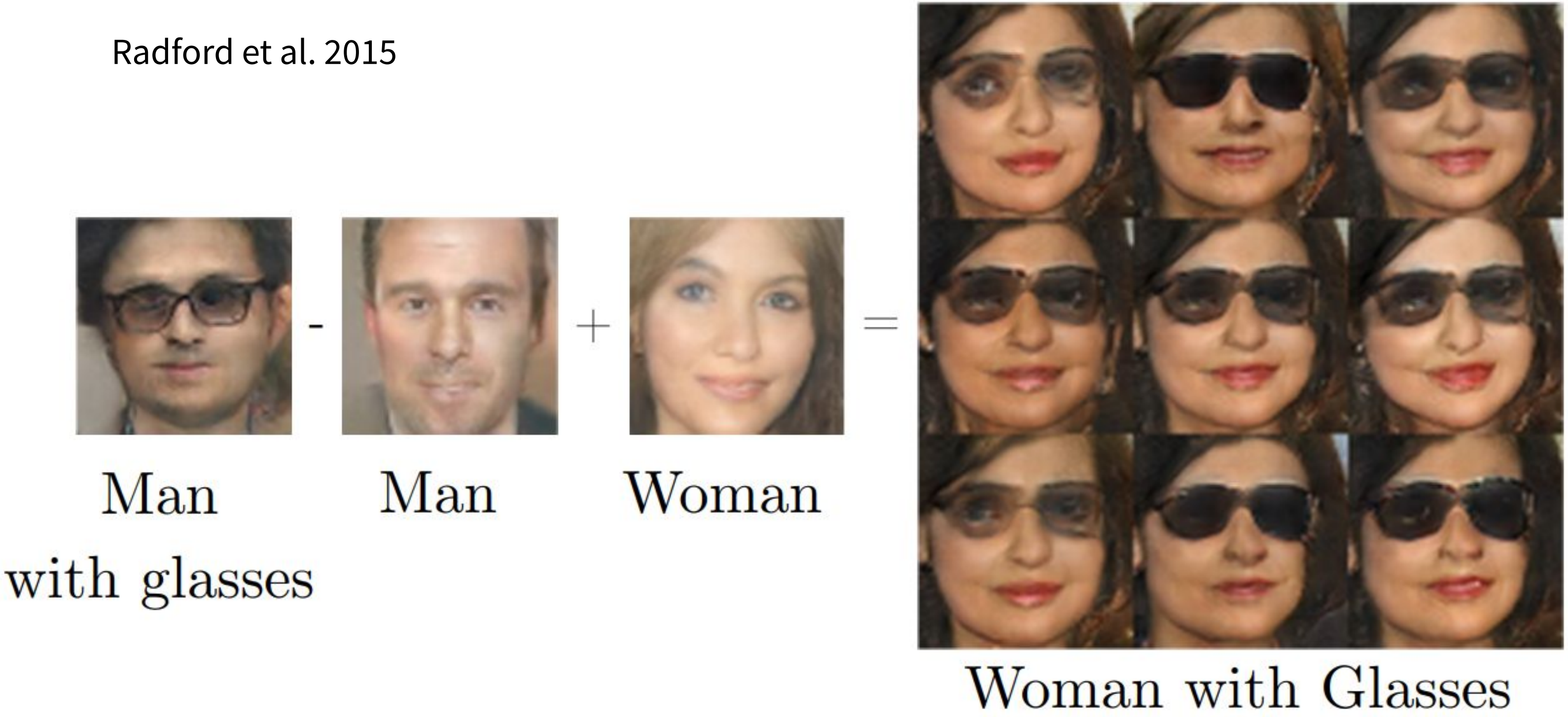
$$J^D = -\frac{1}{2}\mathbb{E}_{x \sim p_{data}} \log D(x) - \frac{1}{2}\mathbb{E}\log(1 - D(G(z)))$$

$$J^G = \frac{1}{2}\mathbb{E} log(D(G(z)))$$

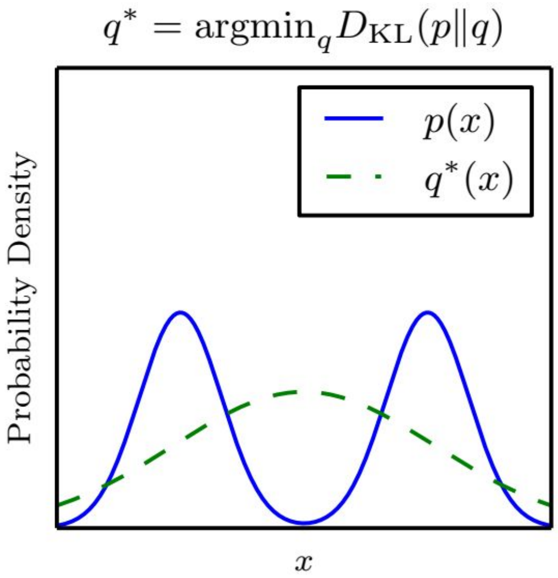
- Equilibrium no longer describable with a single loss
- Generator maximizes log-probability of discriminator being mistaken
- Heuristically, generator can still learn even when discriminator rejects all generator examples

Vector Arithmetic

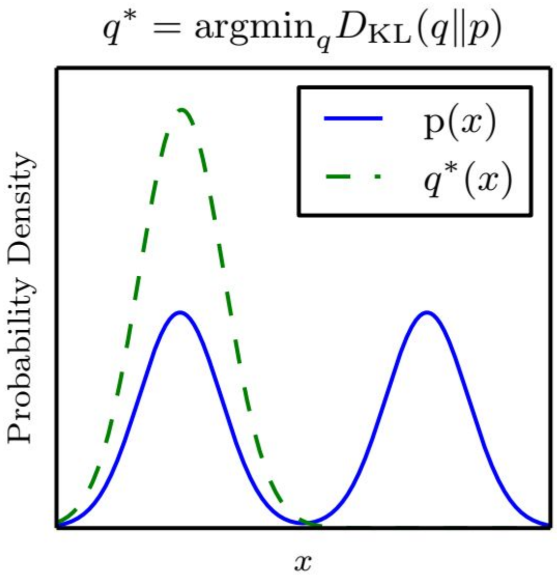
Radford et al. 2015



Original argument for using reverse-KL



Maximum likelihood

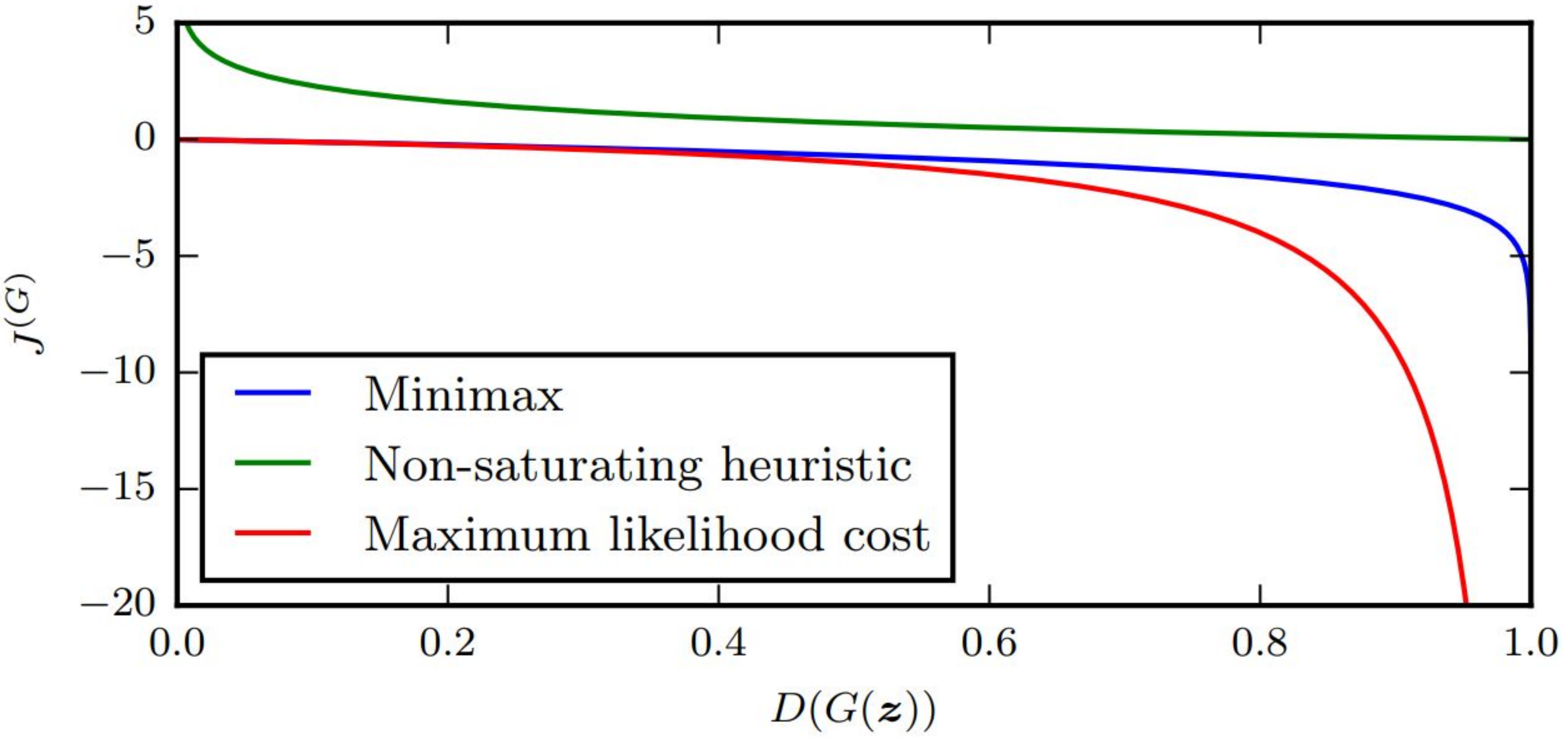


Reverse KL

Reducing GANs to RL

- Generator makes a sample
- Discriminator evaluates sample and gives a reward
- Generator’s cost (negative reward) is a function of D(G(z))

GAN Costs



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

Comparison to Noise Contrastive Estimation

$$V(G, D) = \mathbb{E}_{p_{\text{data}}} \log D(\boldsymbol{x}) + \mathbb{E}_{p_{\text{generator}}} \left(\log\left(1 - D(\boldsymbol{x})\right)\right)$$

	NCE (Gutmann and Hyvärinen 2010)	MLE	GAN
D	$D(x) = \frac{p_{\text{model}}(\boldsymbol{x})}{p_{\text{model}}(\boldsymbol{x}) + p_{\text{generator}}(\boldsymbol{x})}$		Neural network
Goal	Learn p_{model}		Learn $p_{\text{generator}}$
G update rule	None (G is fixed)	Copy p_{model} parameters	Gradient descent on V
D update rule	Gradient ascent on V		

Problem with using JS-Divergence

- Generator Gradient disappears as discriminator approaches optimality
- As $||D - D^*|| < \epsilon$

$$||\nabla_\theta \mathbb{E}_{z\sim p(z)}[\log(1 - D(g_\theta(z)))]||_2 < M\frac{\epsilon}{1-\epsilon}$$

Or

$$\lim_{||D-D^*||\to 0} \nabla_\theta \mathbb{E}_{z\sim p(z)}[\log(1 - D(g_\theta(z)))] = 0$$

So we can see that as discriminator approaches optimality, generator goes to 0

-log(D) alternative

- Alternative gradient step

$$\Delta\theta = \nabla_\theta \mathbb{E}_{z\sim p(z)}[-\log D(g_\theta(z))]$$

We obtain

$$\mathbb{E}_{z\sim p(z)}[-\nabla_\theta log D^*(g_\theta(z))|_{\theta=\theta_0}] = \nabla_\theta[KL(P_{g_\theta}||P_r) - 2JSD(P_{g_\theta}||P_r)]|_{\theta=\theta_0}$$

Problems: 2 JSD are minus sign, so they are pushing distributions to be different
Reverse KL assigns an extremely high cost to fake looking samples and extremely low cost to mode dropping
Also HIGHLY unstable generator gradients

$$\mathbb{E}_{z\sim p(z)}\left[\frac{-J_\theta g_\theta(z) r(z)}{\epsilon(z)}\right]$$

Kantorovich-Rubinstein (Wasserstein) Metric

$$W(P,Q) = \inf_{\gamma \in \Gamma} \int_{\chi \times \chi} ||x - y||_2 \, d\gamma(x,y)$$

- Result in Optimal Transport

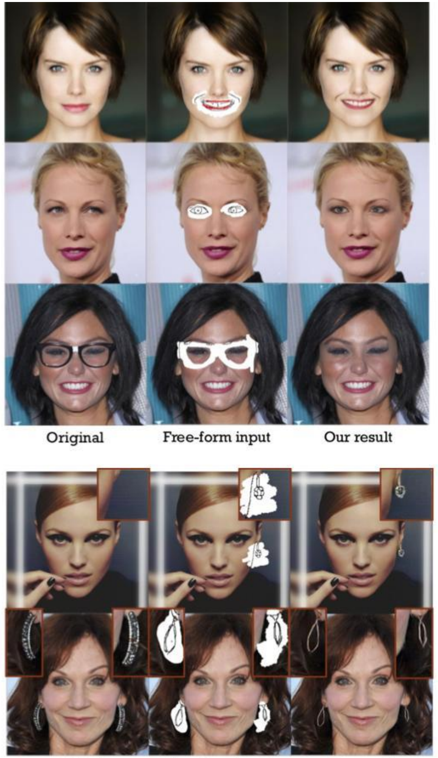
$$W(P_r, P_g) \leq 2V^{\frac{1}{2}} + 2C\sqrt{JSD(P_{r+\epsilon}||P_{g+\epsilon})}$$

Cool New Advancements

Cool New Advancements

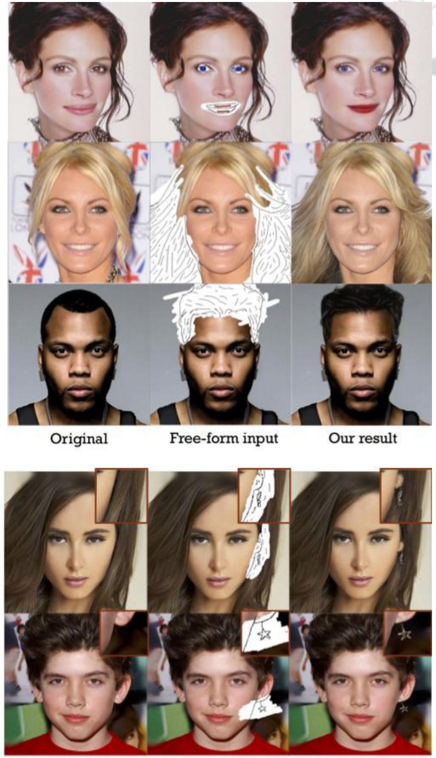
Jo 2019

Original Free-form input Output



Original Free-form input Our result

Original Free-form input Our result



Original Free-form input Our result

Original Free-form input Our result

Further Readings

- (Should Read, but mathematically highly challenging) Why WGANs work: https://arxiv.org/pdf/1701.13851.pdf
- Language GANs falling short: https://arxiv.org/pdf/1811.02549.pdf
- WGAN-GP: https://arxiv.org/pdf/1704.00028.pdf