EN.601.482/682 Deep Learning

Generative Models

Generative Adversarial Networks (GANs)

Mathias Unberath, PhD
Assistant Professor
Dept of Computer Science
Johns Hopkins University

Reminder

Supervised learning

- Ground truth annotations available for every instance (e.g. segmentations, classification, etc)
- Easy to evaluate and compare

Weakly supervised learning

- Annotations, but not for every instance (e.g. only some slices annotated in 3D volume)
- Manageable, may require sophisticated techniques

Unsupervised learning

- No annotations at all
- Complicated (techniques: self-reconstruction, clustering)

Self-supervised learning

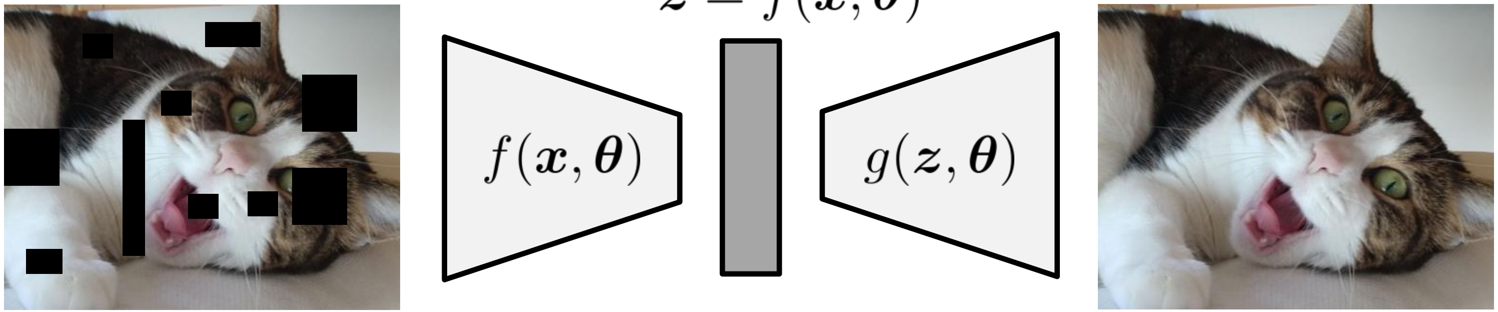
- No annotations, but some signal derived from the images itself
- Difficult to find self-supervision mechanism, but then manageable (even with very good performance!)

Reminder

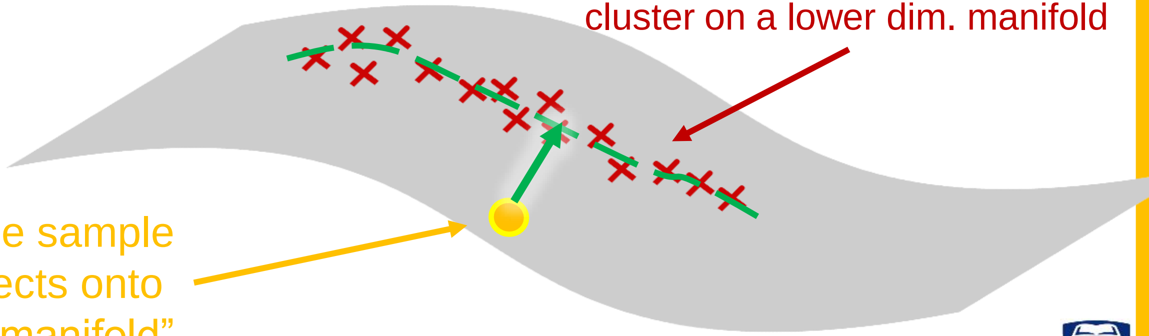
$$\boldsymbol{\theta} = \arg\min_{\hat{\boldsymbol{\theta}}} d\left[\boldsymbol{x}, g(f(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}), \hat{\boldsymbol{\theta}})\right]$$

Denoising autoencoder

$$\boldsymbol{z} = f(\boldsymbol{x}, \boldsymbol{\theta})$$



Valid image sample
→ Hypothesis: Valid images cluster on a lower dim. manifold



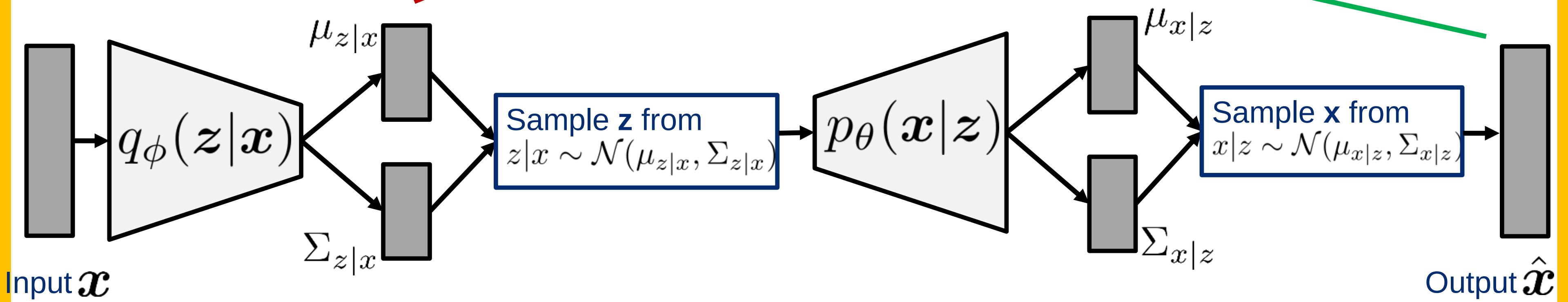
Corrupted image sample
→ Autoencoder projects onto "valid Image manifold"

Variational Autoencoders

Maximizing ELBO $\underbrace{\boldsymbol{E}_{\boldsymbol{z}}\left[\log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z})\right] - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)}), p_\theta(\boldsymbol{z}))}_{\mathcal{L}(\boldsymbol{x}^{(i)},\theta,\phi)}$

Approx. posterior close to prior

Max. likelihood of input being reconstructed!



→ For every minibatch, compute forward pass, then back-prop!

Goal for Today

So far

Image in → Encoding (latent representation) → Image out

We called this (variational) autoencoder

→ For generation, start with latent representation

→ This worked well (also for pre-training), but images still blurry

Now

Random (latent?) representation in → Image out

How to generate sharp samples?

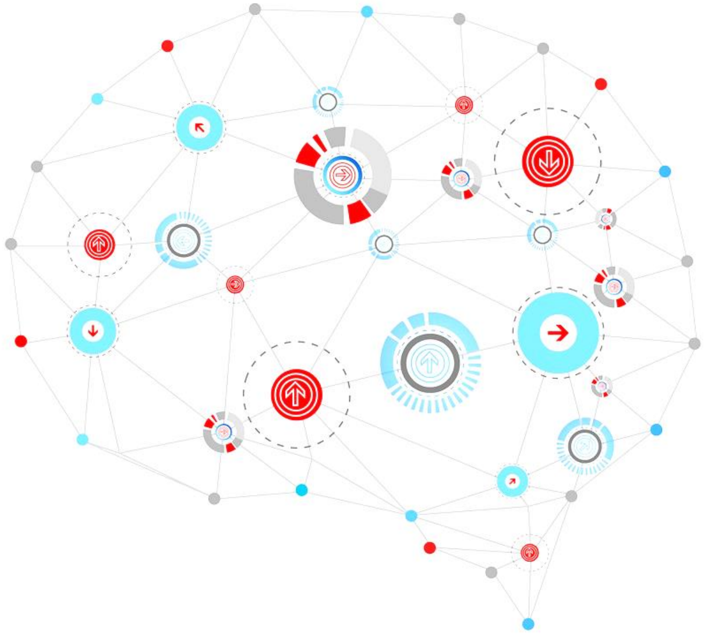
We will see that this strategy works well for a lot of tasks!

Today’s Lecture

Generative Adversarial Networks (GANs)

Conditional GANs

A Small Detour: Adversaries



Generative Adversarial Networks

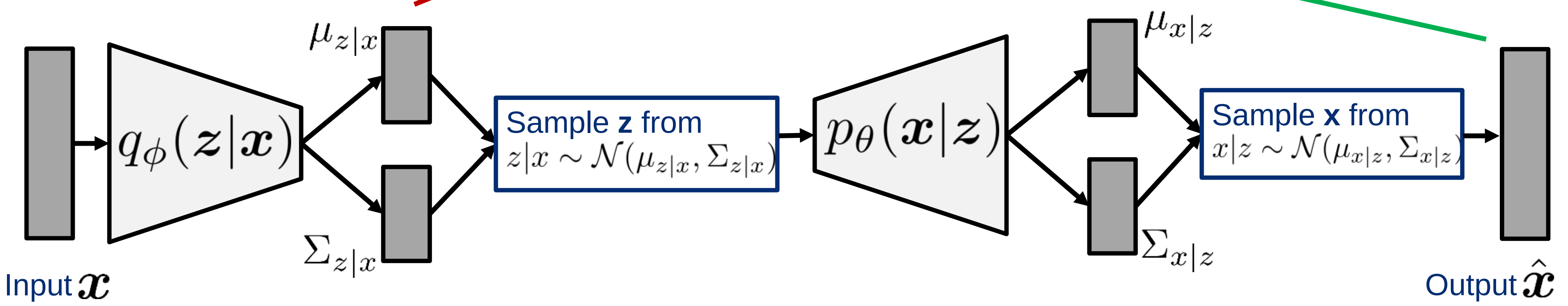
GANs

Variational Autoencoders

Maximizing ELBO $\underbrace{\boldsymbol{E}_{\boldsymbol{z}}\left[\log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z})\right] - D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)}), p_\theta(\boldsymbol{z}))}_{\mathcal{L}(\boldsymbol{x}^{(i)},\theta,\phi)}$

Approx. posterior close to prior

Max. likelihood of input being reconstructed!



→ For every minibatch, compute forward pass, then back-prop!

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. ICLR.

Generative Adversarial Networks

VAEs

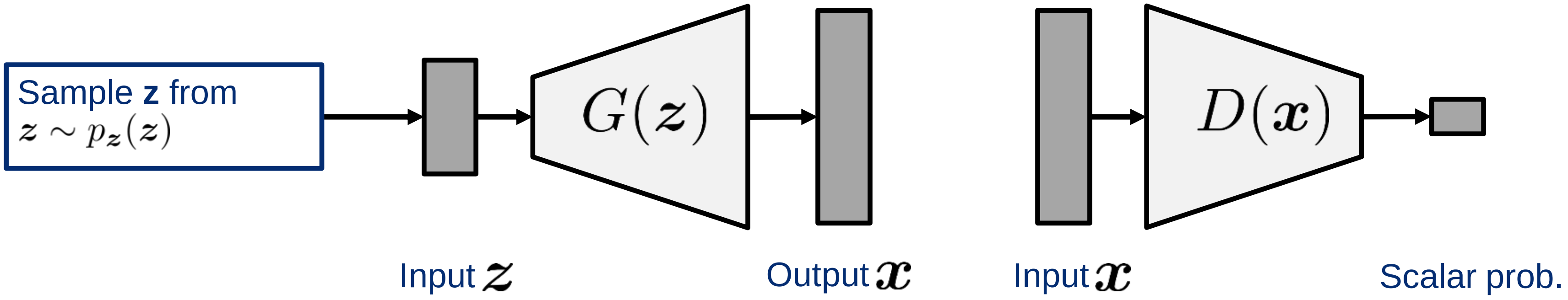
- Maximize density function $p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{z})\, p_\theta(\boldsymbol{x}|\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}$
- Intractable → Maximize lower bound (ELBO)

GANs

- Give up on the idea to explicitly model density functions
- Two player game: Generator vs Discriminator
  - Output is not an image, but a probability (sample from p_{data} or p_G ?)
  - Supervised learning problem
    → This is where backprop, dropout, ReLUs are most successful!

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).

Generative Adversarial Networks



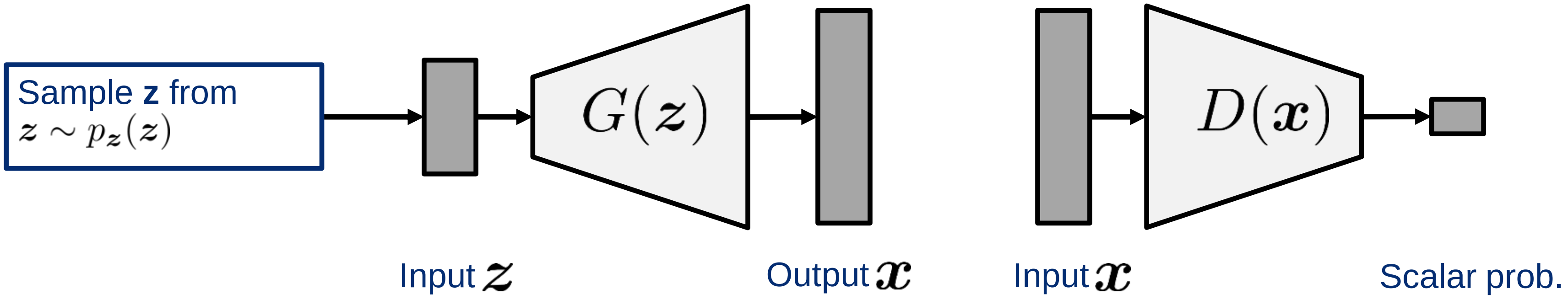
$$\min_G \max_D V(D,G) = \boldsymbol{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \boldsymbol{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G((z))))]$$

GANs: A two-player minimax game with value function V(D,G)

Let’s dissect this a bit more carefully!

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).

Generative Adversarial Networks



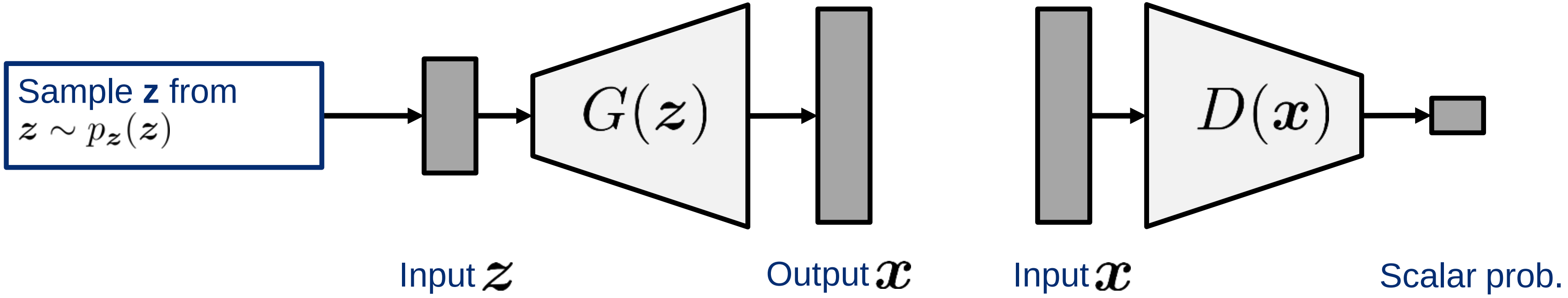
$$\min_{G} \max_{D} V(D,G) = \boldsymbol{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \boldsymbol{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G((z))))]$$

The generator

- Accepts an input noise vector with prior $p_{\boldsymbol{z}}(\boldsymbol{z})$
- Represents mapping $G_{\theta_g}(\boldsymbol{z})$ that generates distribution p_g over the data space
- G is a differentiable function

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).

Generative Adversarial Networks



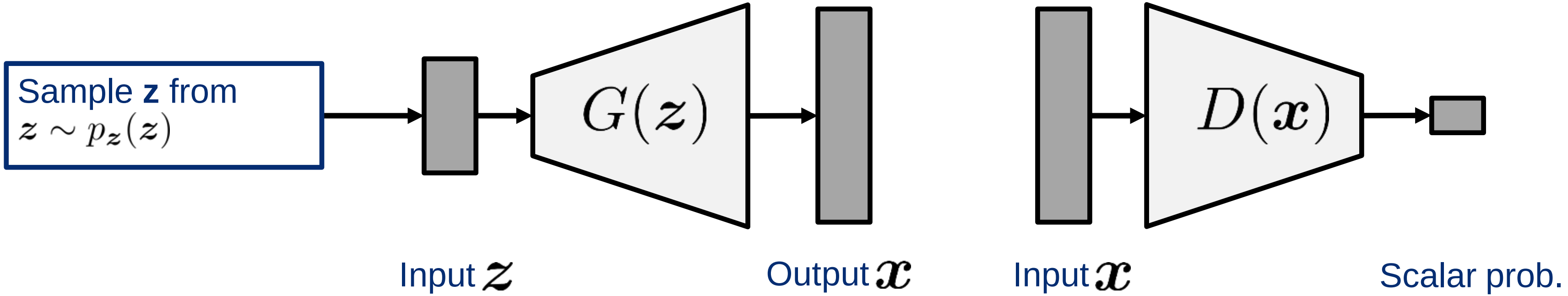
$$\min_{G} \max_{D} V(D,G) = \boldsymbol{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \boldsymbol{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G((z))))]$$

The discriminator

- Accepts an input sample $\boldsymbol{x}$
- Represents mapping $D_{\theta_d}(\boldsymbol{x})$ that yields probability of $\boldsymbol{x} \sim p_{\mathrm{data}}$
- D is also a differentiable function

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).

Generative Adversarial Networks



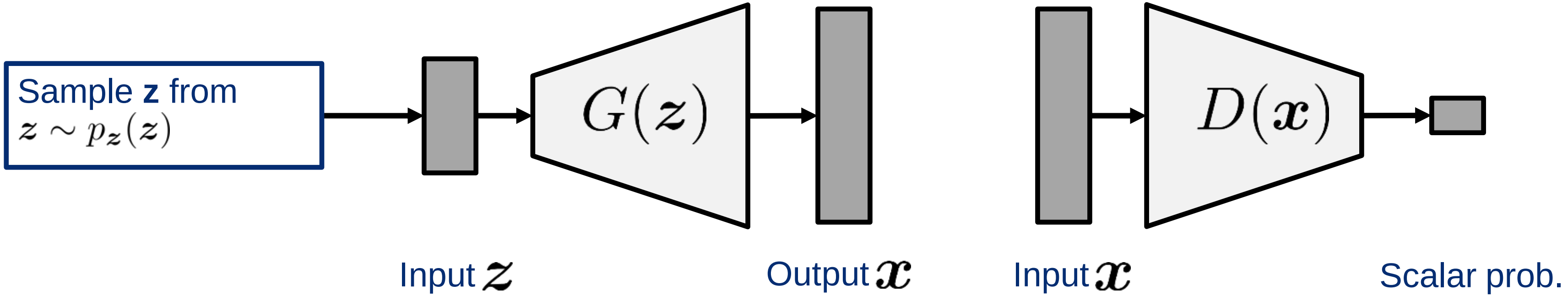
$$\min_G \boxed{\max_D} V(D, G) = \boxed{\boldsymbol{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]} + \boxed{\boldsymbol{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G((z))))]}$$

The value function

- Discriminator $D_{\theta_d}(\boldsymbol{x})$ assigns correct label to any sample it is presented: real / fake
  → Should be maximal: Very good discrimination

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).

Generative Adversarial Networks



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

The value function

- Discriminator $D_{\theta_d}(\boldsymbol{x})$ assigns correct label to any sample it is presented: real / fake
  → Should be maximal: Very good discrimination
- Generator $G_{\theta_g}(\boldsymbol{z})$ attempts to "fool" the discriminator
  → Should be minimal: Poor discrimination

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).

Generative Adversarial Networks

Some practical considerations

- We consider the non-parametric limit
  (both generator and discriminator have sufficient capacity)

- Optimization is done in iterative procedure
  - k-steps of optimizing D, one step of optimizing G

Q: Why?

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).

Generative Adversarial Networks

Some practical considerations

- We consider the non-parametric limit
  (both generator and discriminator have sufficient capacity)

- Optimization is done in iterative procedure
  - k-steps of optimizing D, one step of optimizing G
  - Optimizing D to completion early results in overfitting (Discriminator too strong)
  - Same problem, different spin: Training G, early in learning, G is poor!
    Discrimination is easy and $\log(1 - D(G((z)))$ saturates → No gradients

    Stronger gradients by training G to maximize $\log(D(G((z)))$

[Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).]

Generative Adversarial Networks

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:

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

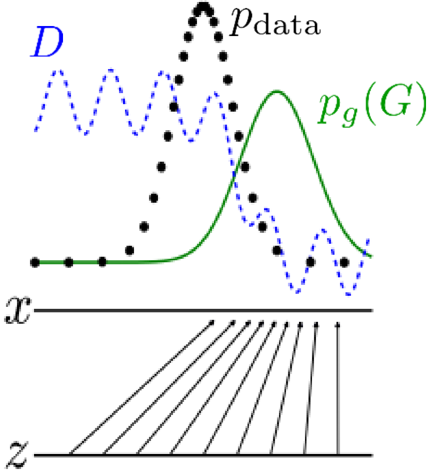
end for

- 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

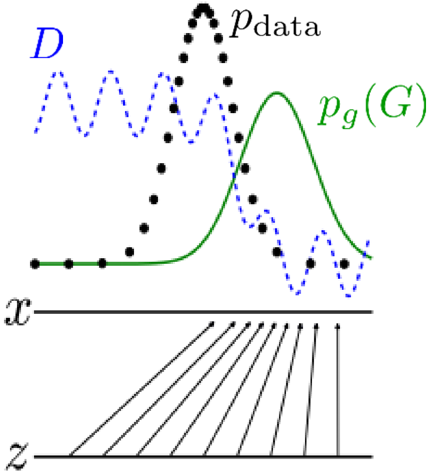
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).



Simultaneous update of discriminative distribution (**D**), to discriminate between samples from the real and the generative distribution

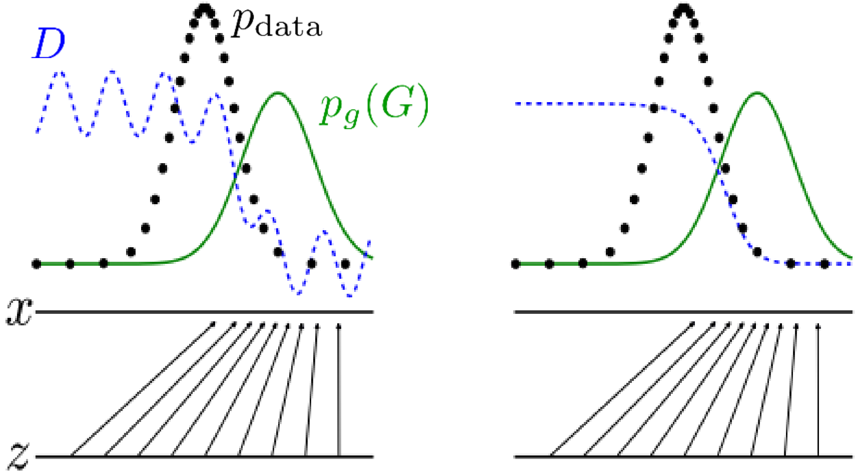
In this case, z is sampled uniformly and $\boldsymbol{x} = G(\boldsymbol{z})$ imposes non-uniform distribution $p_g(G)$

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).



- Near convergence: $p_g(G)$ is similar to p_{data}, and $D(\boldsymbol{x})$ is partially accurate

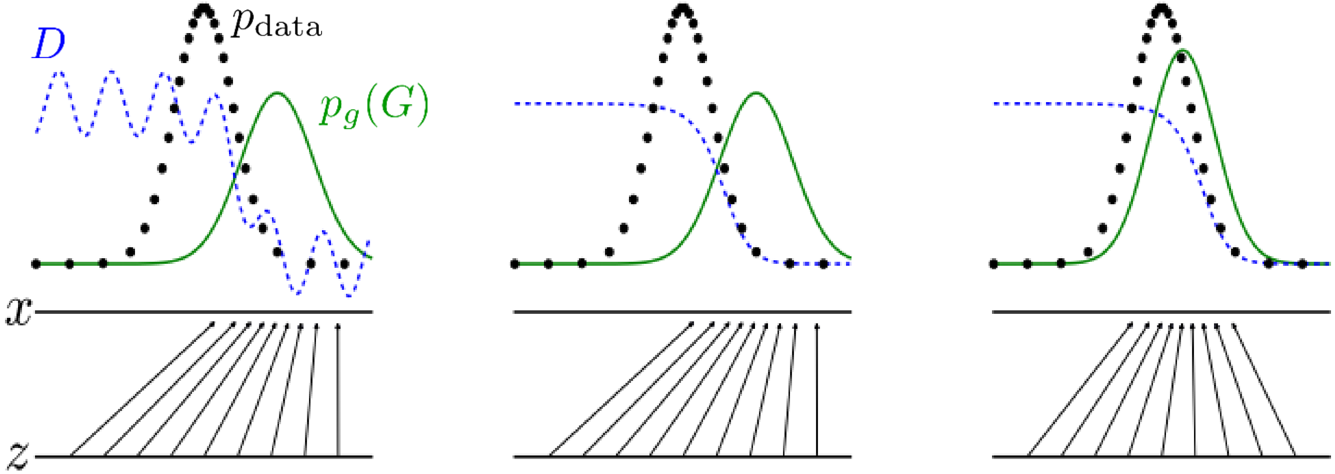
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).



- Near convergence: $p_g(G)$ is similar to p_{data}, and $D(\boldsymbol{x})$ is partially accurate

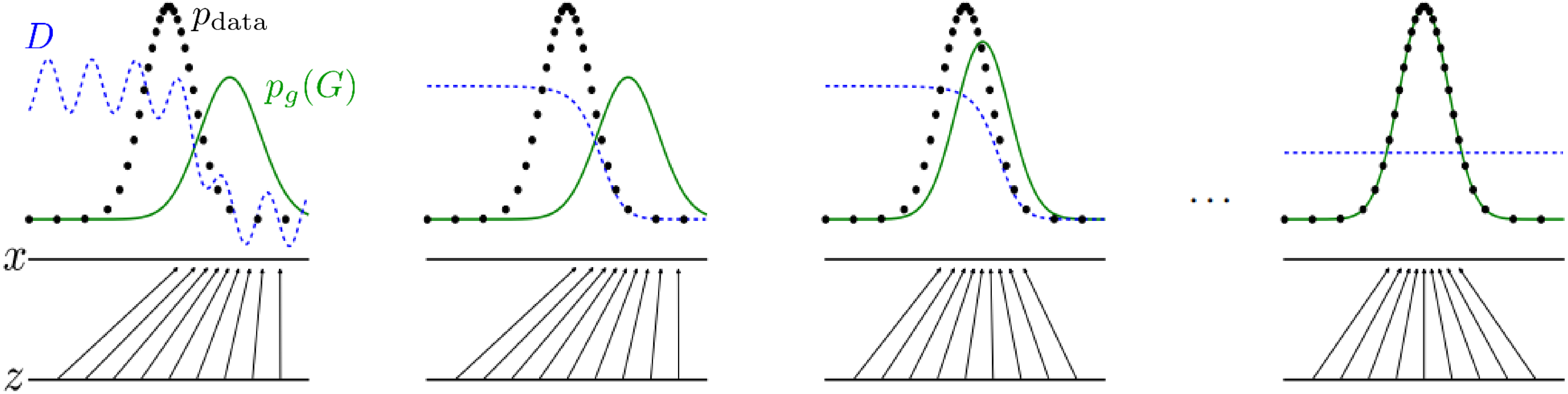
- Inner loop: $D(\boldsymbol{x})$ is trained to better discriminate, converging to $D^{\star}(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).



- Near convergence: $p_g(G)$ is similar to p_{data}, and $D(\boldsymbol{x})$ is partially accurate
- Inner loop: $D(\boldsymbol{x})$ is trained to better discriminate, converging to $D^{\star}(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$
- After G update: Gradient of $D(\boldsymbol{x})$ has guided $G(\boldsymbol{z})$ to be more likely classified as “real”

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).



- Near convergence: $p_g(G)$ is similar to p_{data}, and $D(\boldsymbol{x})$ is partially accurate
- Inner loop: $D(\boldsymbol{x})$ is trained to better discriminate, converging to $D^\star(\boldsymbol{x}) = \frac{p_{\mathrm{data}}(\boldsymbol{x})}{p_{\mathrm{data}}(\boldsymbol{x})+p_g(\boldsymbol{x})}$
- After G update: Gradient of $D(\boldsymbol{x})$ has guided $G(\boldsymbol{z})$ to be more likely classified as “real”
- After multiple iterations, $p_g(G) = p_{\mathrm{data}}$ and $D(\boldsymbol{x}) = \frac{1}{2}$

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).

Generative Adversarial Networks

Some more practical considerations

- Optimization is done in iterative procedure
  - k-steps of optimizing D, one step of optimizing G
  - Discrimination is too strong $\log(1 - D(G((z)))$ saturates → No gradients
  - Another problem: G cannot be trained too much without updating D

Q: Why?

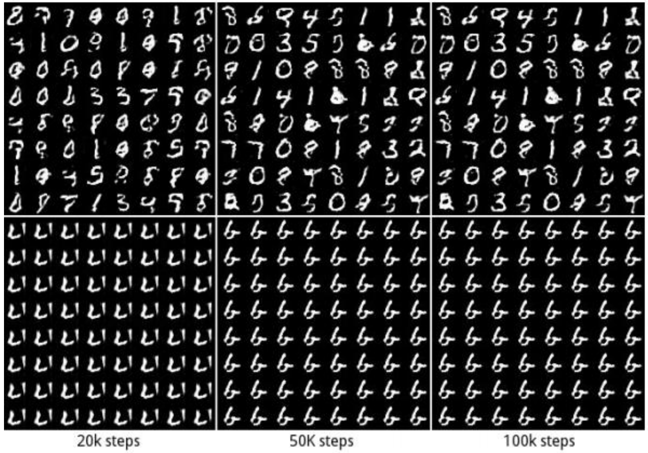
Generative Adversarial Networks

Some more practical considerations

- Optimization is done in iterative procedure
  - k-steps of optimizing D, one step of optimizing G
  - Discrimination is too strong $\log(1 - D(G((z)))$ saturates → No gradients

  - Another problem: G cannot be trained too much without updating D
    G would learn to collapse too many samples of **z** to the same value of **x**
    → **Mode collapse**

  - Gradient descent is good at finding the minimum of a function, not the Nash equilibrium of a game.



Metz, L., Poole, B., Pfau, D., & Sohl-Dickstein, J. (2016). Unrolled generative adversarial networks. ICLR 2017.

Note: Only d) are results achieved with a convolutional model!

Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative models, these images show actual samples from the model distributions, not conditional means given samples of hidden units. Moreover, these samples are uncorrelated because the sampling process does not depend on Markov chain mixing. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and "deconvolutional" generator)

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. NeurIPS (pp. 2672-2680).

Improvements in GANs

Nowadays, GANs are SOTA. But slow start due to complicated optimization.
Inception score of **52.5** (Salimans et al. 2016) vs **233** for real data (ImageNet)

BigGAN: Inception scores of **166.5**

- Key contributions
  - GANs benefit from scaling: 2x – 4x increase in parameters and 8x the batch size (**Big**GAN)
  - Regularization scheme to improve conditioning and boost performance (truncation)

- Detailed analysis of failure cases (worth a read!)

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. NeurIPS (pp. 2234-2242).
Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

Improvements in GANs

(a) 128×128 (b) 256×256 (c) 512×512 (d)

Figure 4: Samples from our BigGAN model with truncation threshold 0.5 (a-c) and an example of class leakage in a partially trained model (d).

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. NeurIPS (pp. 2234-2242).
Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

Improvements in GANs

Progressive Growing: Outputs up to 1024²

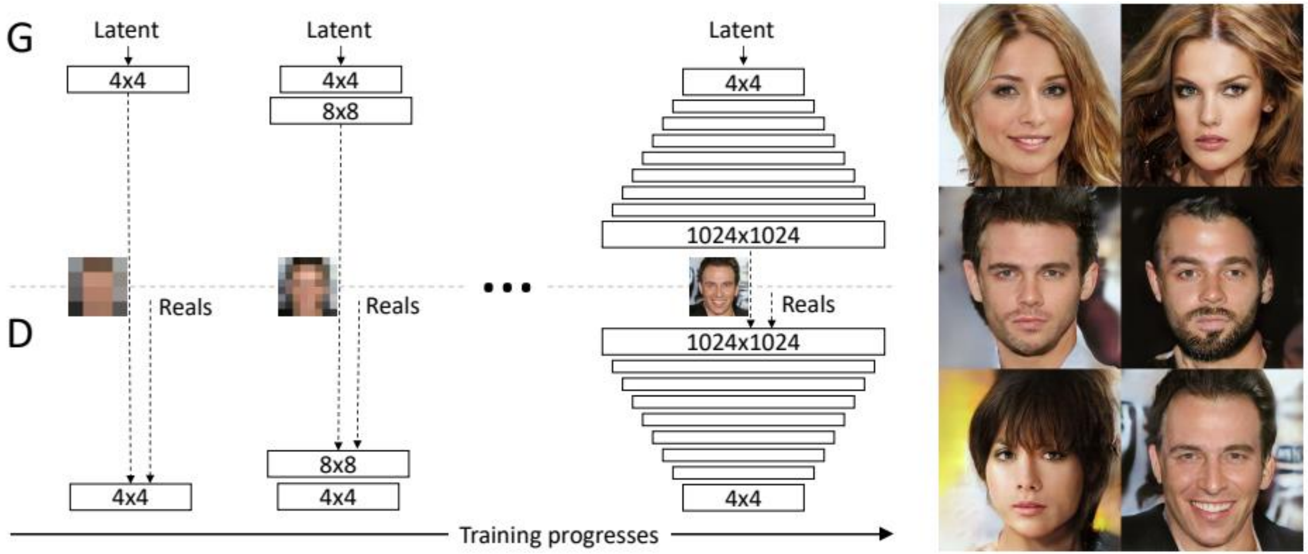


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.
Karras, T., Laine, S., & Aila, T. (2018). A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948.

Improvements in GANs

Progressive Growing: Outputs up to 1024^2

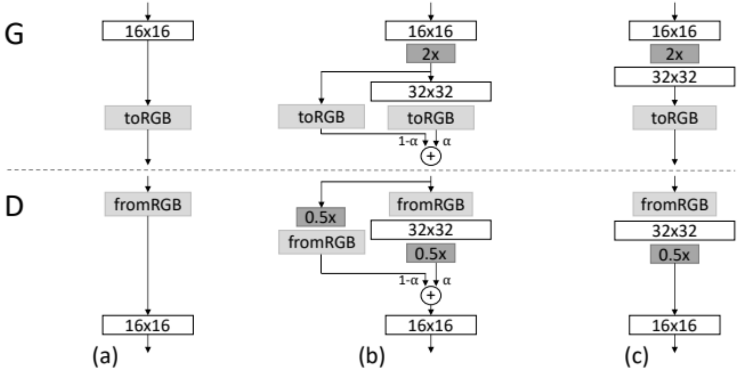
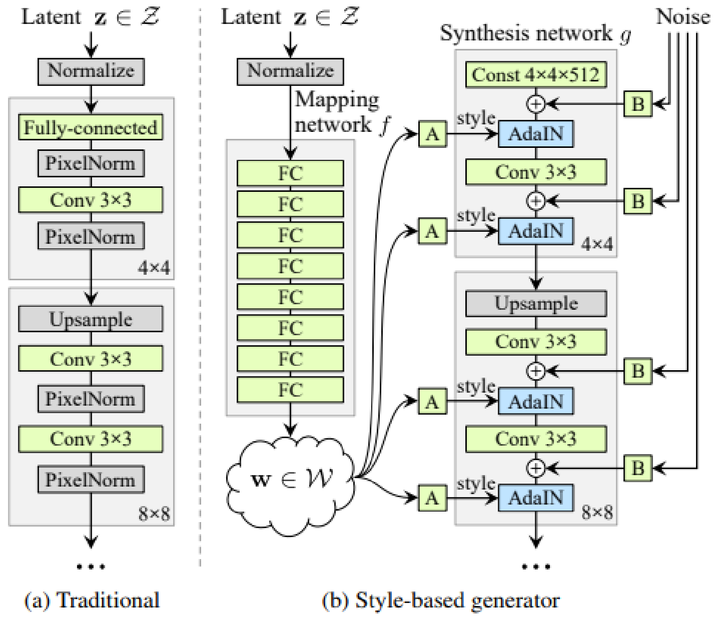


Figure 2: When doubling the resolution of the generator (G) and discriminator (D) we fade in the new layers smoothly. This example illustrates the transition from 16×16 images (a) to 32×32 images (c). During the transition (b) we treat the layers that operate on the higher resolution like a residual block, whose weight α increases linearly from 0 to 1. Here $\boxed{2\times}$ and $\boxed{0.5\times}$ refer to doubling and halving the image resolution using nearest neighbor filtering and average pooling, respectively. The $\boxed{\text{toRGB}}$ represents a layer that projects feature vectors to RGB colors and $\boxed{\text{fromRGB}}$ does the reverse; both use 1×1 convolutions. When training the discriminator, we feed in real images that are downscaled to match the current resolution of the network. During a resolution transition, we interpolate between two resolutions of the real images, similarly to how the generator output combines two resolutions.

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.
Karras, T., Laine, S., & Aila, T. (2018). A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948.

Improvements in GANs

Progressive Growing:
With style-based generator



Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196.*
Karras, T., Laine, S., & Aila, T. (2018). A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948.

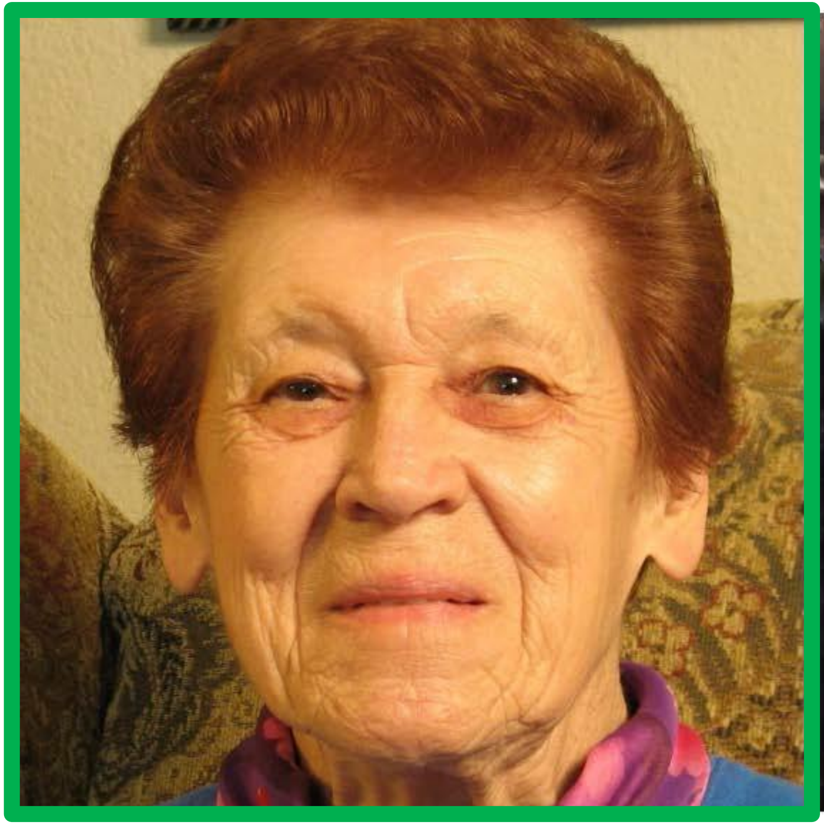
http://www.whichfaceisreal.com/index.php


http://www.whichfaceisreal.com/index.php

http://www.whichfaceisreal.com/index.php

http://www.whichfaceisreal.com/index.php

http://www.whichfaceisreal.com/index.php




http://www.whichfaceisreal.com/index.php

Relativistic Discriminators

One key problem of GANs: Discriminator learning too quickly

Relativistic discriminators:
Prob. of real data being real should decrease as fake data becomes more real

→ Discriminator estimates probability that given real data is more realistic than randomly sampled fake data

Jolicoeur-Martineau, A. (2018). The relativistic discriminator: a key element missing from standard GAN. arXiv preprint arXiv:1807.00734.

Relativistic Discriminators

Relativistic discriminators:
Prob. of real data being real should decrease as fake data becomes more real

Algorithm 1 Training algorithm for non-saturating RGANs with symmetric loss functions

Require: The number of D iterations n_D ($n_D = 1$ unless one seeks to train D to optimality), batch size m, and functions f which determine the objective function of the discriminator (f is f_1 from equation 10 assuming that $f_2(-y) = f_1(y)$, which is true for many GANs).

while θ has not converged **do**

Discriminator loop

- **for** $t = 1, \ldots, n_D$ **do**
  - Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}$
  - Sample $\{z^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_z$
  - Update w using SGD by ascending with $\nabla_w \frac{1}{m} \sum_{i=1}^{m} \boxed{\left[f\left(\boxed{C_w(x^{(i)}) - C_w(G_\theta(z^{(i)}))}\right)\right]}$
- **end for**

f is discriminator objective

This difference is the “real change”

Generator update

- Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}$
- Sample $\{z^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_z$
- Update θ using SGD by ascending with $\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} \left[f(C_w(G_\theta(z^{(i)})) - C_w(x^{(i)}))\right]$

end while

Jolicoeur-Martineau, A. (2018). The relativistic discriminator: a key element missing from standard GAN. arXiv preprint arXiv:1807.00734.

Table 1: A illustrative example of the discriminator's output in standard GAN as traditionally defined ($P(x_r \text{ is real}) = \text{sigmoid}(C(x_r))$) versus the Relativistic average Discriminator (RaD) ($P(x_r \text{ is real}|\overline{C(x_f)}) = \text{sigmoid}(C(x_r) - \overline{C(x_f)})$). Breads represent real images, while dogs represent fake images.

Scenario	Absolute probability (Standard GAN)	Relative probability (Relativistic average Standard GAN)
Real image looks real **and** fake images look fake	 $C(x_r) = 8$ $P(x_r \text{ is bread}) = 1$	 $\overline{C(x_f)} = -5$ $P(x_r \text{ is bread}|\overline{C(x_f)}) = 1$

Table 1: A illustrative example of the discriminator's output in standard GAN as traditionally defined ($P(x_r$ is real$) = \text{sigmoid}(C(x_r))$) versus the Relativistic average Discriminator (RaD) ($P(x_r$ is real$|\overline{C(x_f)}) = \text{sigmoid}(C(x_r) - \overline{C(x_f)})$). Breads represent real images, while dogs represent fake images.

Scenario	Absolute probability (Standard GAN)	Relative probability (Relativistic average Standard GAN)
Real image looks real **but** fake images look similarly real on average	 $C(x_r) = 8$ $P(x_r \text{ is bread}) = 1$	 $\overline{C(x_f)} = 7$ $P(x_r \text{ is bread}\,|\,\overline{C(x_f)}) = .73$

Table 1: A illustrative example of the discriminator's output in standard GAN as traditionally defined ($P(x_r \text{ is real}) = \text{sigmoid}(C(x_r))$) versus the Relativistic average Discriminator (RaD) ($P(x_r \text{ is real}|\overline{C(x_f)}) = \text{sigmoid}(C(x_r) - \overline{C(x_f)})$). Breads represent real images, while dogs represent fake images.

Scenario	Absolute probability (Standard GAN)	Relative probability (Relativistic average Standard GAN)
Real image looks fake **but** fake images look more fake on average	 $C(x_r) = -3$ $P(x_r \text{ is bread}) = .05$	 $\overline{C(x_f)} = -5$ $P(x_r \text{ is bread}\|\overline{C(x_f)}) = .88$

Relativistic Discriminators

One key problem of GANs: Discriminator learning too quickly

Relativistic discriminators:
Prob. of real data being real should decrease as fake data becomes more real

Take away:
- Relativistic assumption is intuitive
- Paper shows improvements, but technique not (yet?) widely adopted

- For your own problems: **May want to give it a try**

Jolicoeur-Martineau, A. (2018). The relativistic discriminator: a key element missing from standard GAN. arXiv preprint arXiv:1807.00734.

Generative Adversarial Networks

Conditional GANs

Image-to-image Translation

So far: GANs can generate realistic-looking images from noise

Exciting (academically) but **a bit lacking in terms of application**

Q: Why?

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR (pp. 1125-1134).

Image-to-image Translation

So far: GANs can generate realistic-looking images from noise

Exciting (academically) but **a bit lacking in terms of application**

Q: Why?

Because most problems are not that open.

→ We would like a way to condition the output of a GAN.

→ **Enter conditional adversarial networks**

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR (pp. 1125-1134).

Image-to-image Translation

Standard GAN: Random noise vector to output image

$$\boldsymbol{E}_{\boldsymbol{x}}[\log D(\boldsymbol{x})] + \boldsymbol{E}_{\boldsymbol{z}}[\log(1 - D(G((z)))]$$

Conditional GAN: Random noise vector + observed image to output image

$$\boldsymbol{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x}, \boldsymbol{y})] + \boldsymbol{E}_{\boldsymbol{x},\boldsymbol{z}}[\log(1 - D(\boldsymbol{x}, G(\boldsymbol{x}, \boldsymbol{z})))]$$

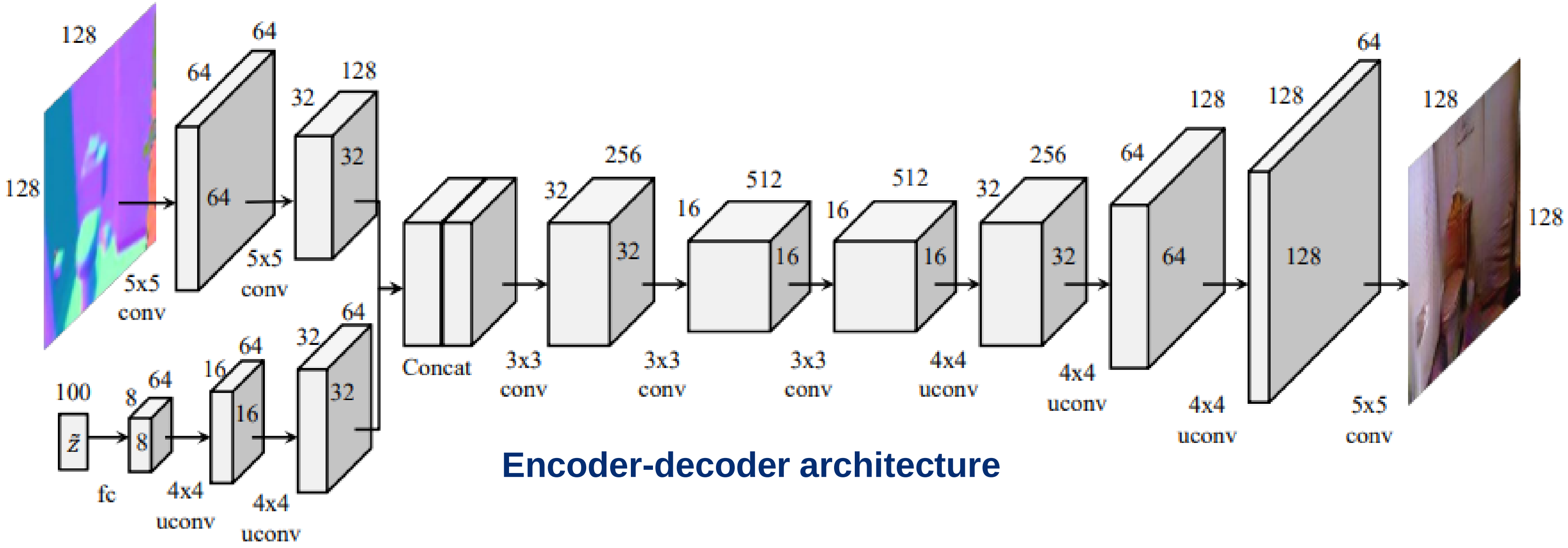
Q: How to mix noise vector with conditioning image?

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR (pp. 1125-1134).

Image-to-image Translation

Conditional GAN: Random noise vector + observed image to output image

$$\boldsymbol{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x},\boldsymbol{y})] + \boldsymbol{E}_{\boldsymbol{x},\boldsymbol{z}}[\log(1 - D(\boldsymbol{x}, G(\boldsymbol{x},\boldsymbol{z})))]$$



Encoder-decoder architecture

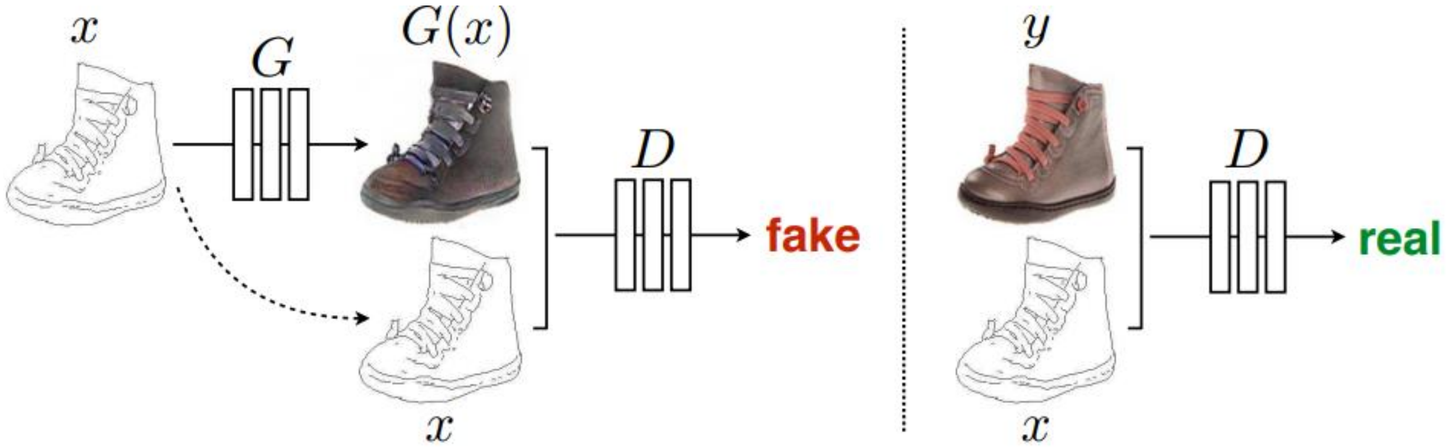
Wang, X., & Gupta, A. (2016, October). Generative image modeling using style and structure adversarial networks. ECCV (pp. 318-335). Springer, Cham.

Image-to-image Translation

Conditional GAN: Random noise vector + observed image to output image

$$\boldsymbol{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x},\boldsymbol{y})] + \boldsymbol{E}_{\boldsymbol{x},\boldsymbol{z}}[\log(1 - D(\boldsymbol{x}, G(\boldsymbol{x},\boldsymbol{z})))]$$

Adding noise may neither be necessary nor effective. Generator may learn to ignore added noise (Isola et al 2018). → **Add noise via dropout!**



Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR (pp. 1125-1134).

Image-to-image Translation

Conditional GAN: Random noise vector + observed image to output image

$$\boldsymbol{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x},\boldsymbol{y})] + \boldsymbol{E}_{\boldsymbol{x},\boldsymbol{z}}[\log(1 - D(\boldsymbol{x}, G(\boldsymbol{x},\boldsymbol{z})))]$$

Improving localization and sharpness

- Previous generators: **Encoder – decoder**
- All information must pass through bottleneck
- Problematic for low-level information

→ **Skip connections** to directly shuttle this information across

→ **U-net-like** architecture

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR (pp. 1125-1134).

Image-to-image Translation

Overall loss function

cGAN $L_{cGAN}(G, D) = \boldsymbol{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x}, \boldsymbol{y})] + \boldsymbol{E}_{\boldsymbol{x},\boldsymbol{z}}[\log(1 - D(\boldsymbol{x}, G(\boldsymbol{x}, \boldsymbol{z})))]$

But, for now, image data is paired. For every **x** we know **y**

→ **G** should not only fool **D**, but also **yield image similar to real image**

→ Additional, more traditional loss

$$L_{L1}(G) = \boldsymbol{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}[\|\boldsymbol{y} - G(\boldsymbol{x}, \boldsymbol{z})\|_1]$$

Overall objective: $\boxed{G^\star = \arg\min_G \max_D \; L_{cGAN}(G, D) + \lambda L_{L1}(G)}$

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR (pp. 1125-1134).

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR (pp. 1125-1134).

Image-to-image Translation

Overall loss function $\boxed{G^\star = \arg\min_G \max_D \; L_{\mathrm{cGAN}}(G, D) + \lambda L_{\mathrm{L1}}(G)}$

PatchGAN – A restricted discriminator

- L1 (or L2) loss produce blurry images in vision tasks
- **But**: Accurately capture low frequencies!

→ Restrict GAN discriminator to high-frequency structure!

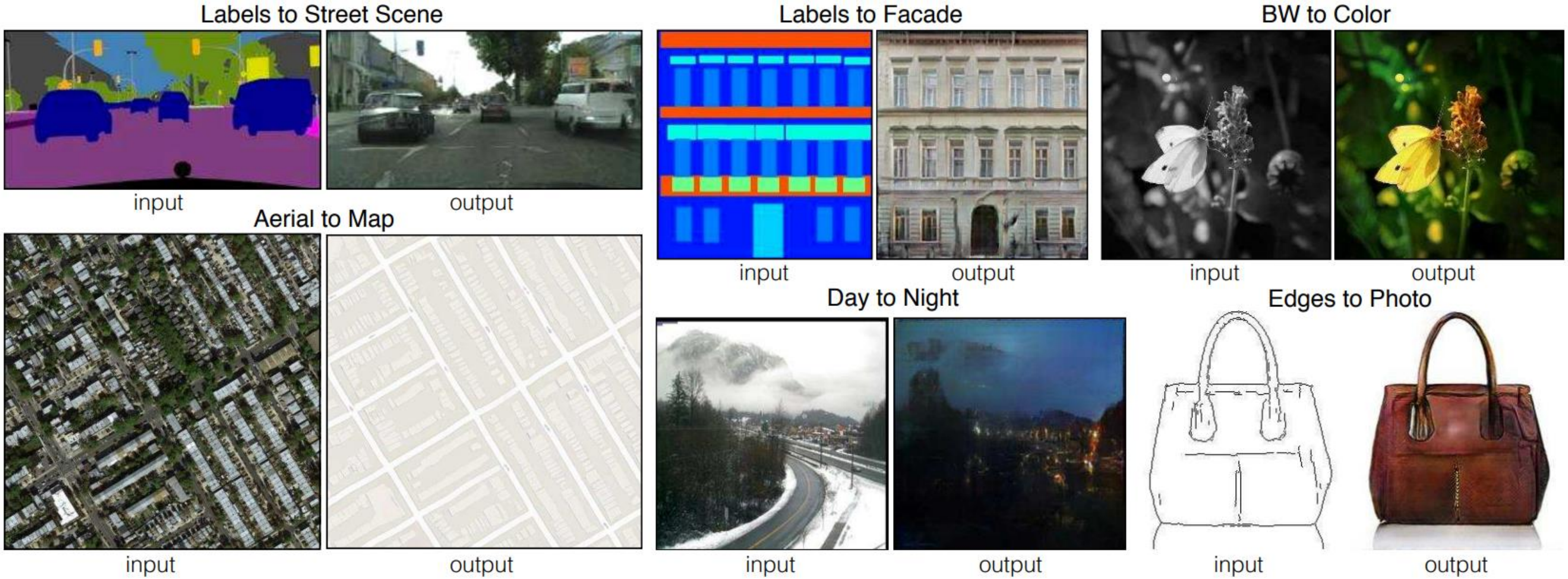
→ **D** is evaluated on N x N (N ~ 70 px) patches (FCN)

→ Final result: Average over all patches

→ Can be understood as "texture/style loss"

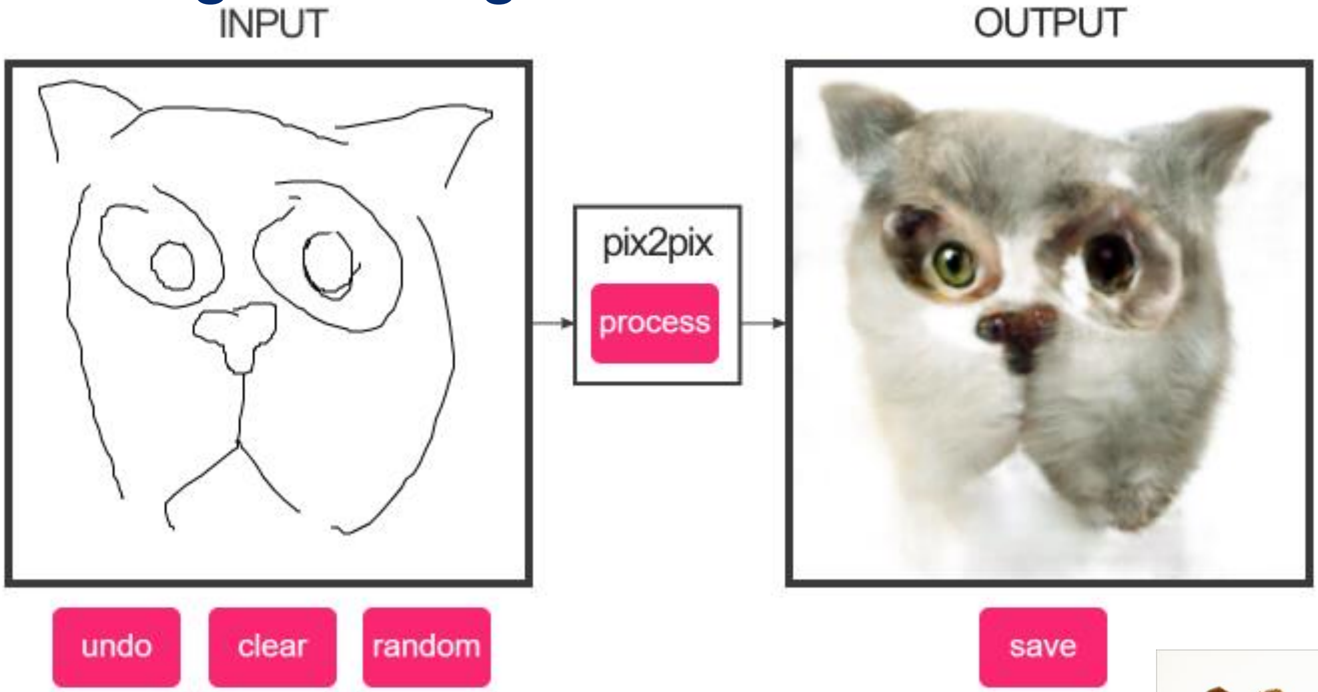
Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR (pp. 1125-1134).

Image-to-image Translation



Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR (pp. 1125-1134).

Image-to-image Translation



Disclaimer: I made this wonderful cat.
Don't you think it has some similarity with Wes Anderson's Isle of Dogs?

Pix2pix cat generator

[Pix2pix online demo](#).

Unpaired Image-to-image Translation

From paired to unpaired data

- Paired data is expensive
- Paired data may be unavailable, or even impossible to obtain

Unpaired data: Much more common!

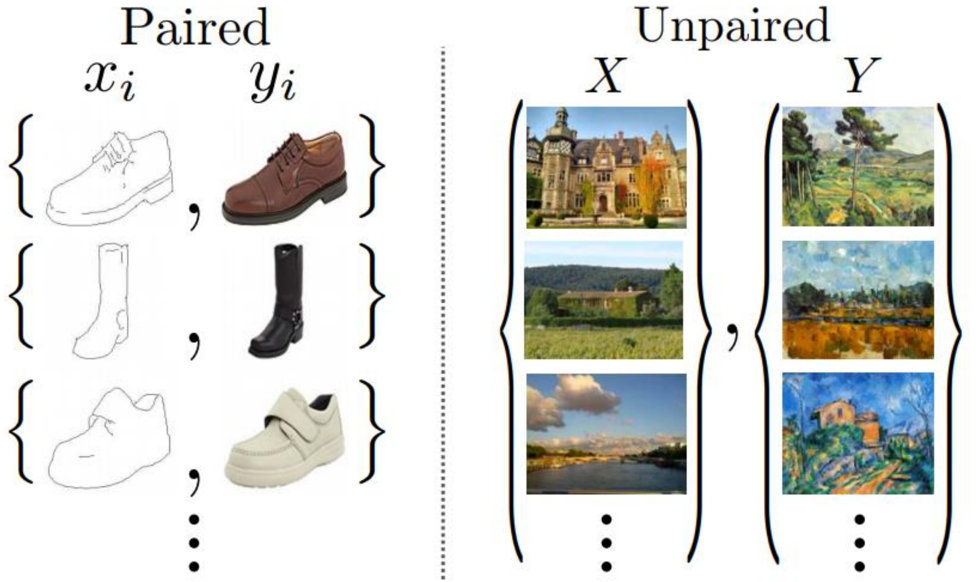
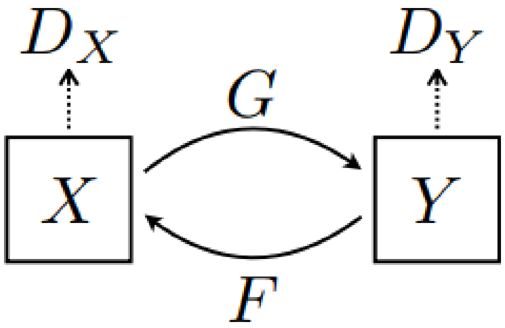


Figure 2: *Paired* training data (left) consists of training examples $\{x_i, y_i\}_{i=1}^{N}$, where the y_i that corresponds to each x_i is given [20]. We instead consider *unpaired* training data (right), consisting of a source set $\{x_i\}_{i=1}^{N} \in X$ and a target set $\{y_j\}_{j=1}^{M} \in Y$, with no information provided as to which x_i matches which y_j.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. CVPR (pp. 2223-2232).

Unpaired Image-to-image Translation

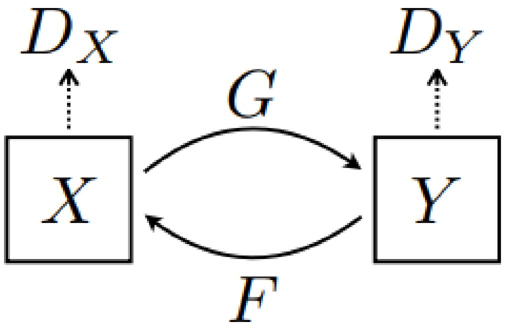


- Two style domains: **X** and **Y**
- Two generator functions **G(X) → Y** and **F(Y) → X**
- Two discriminator functions **D(X)** and D**(Y)**

Q: Any potential problem with this?

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. CVPR (pp. 2223-2232).

Unpaired Image-to-image Translation



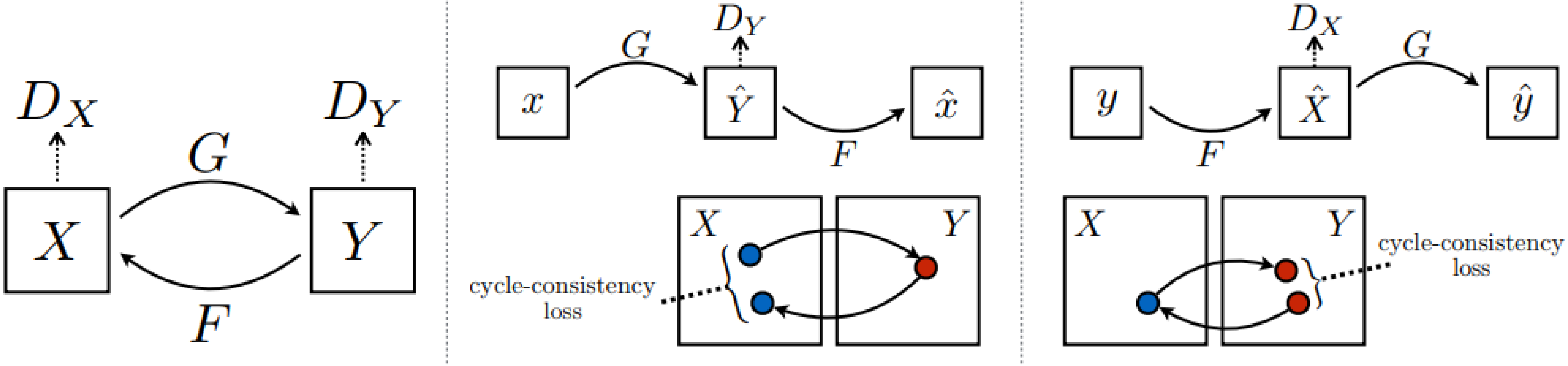
- Two style domains: **X** and **Y**
- Two generator functions **G(X) → Y** and **F(Y) → X**
- Two discriminator functions **D(X)** and D**(Y)**

Q: Any potential problem with this?

Networks with large enough capacity can, in principle, directly map input images to a random permutation of output images. No learning, just memorization!

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. CVPR (pp. 2223-2232).

Unpaired Image-to-image Translation



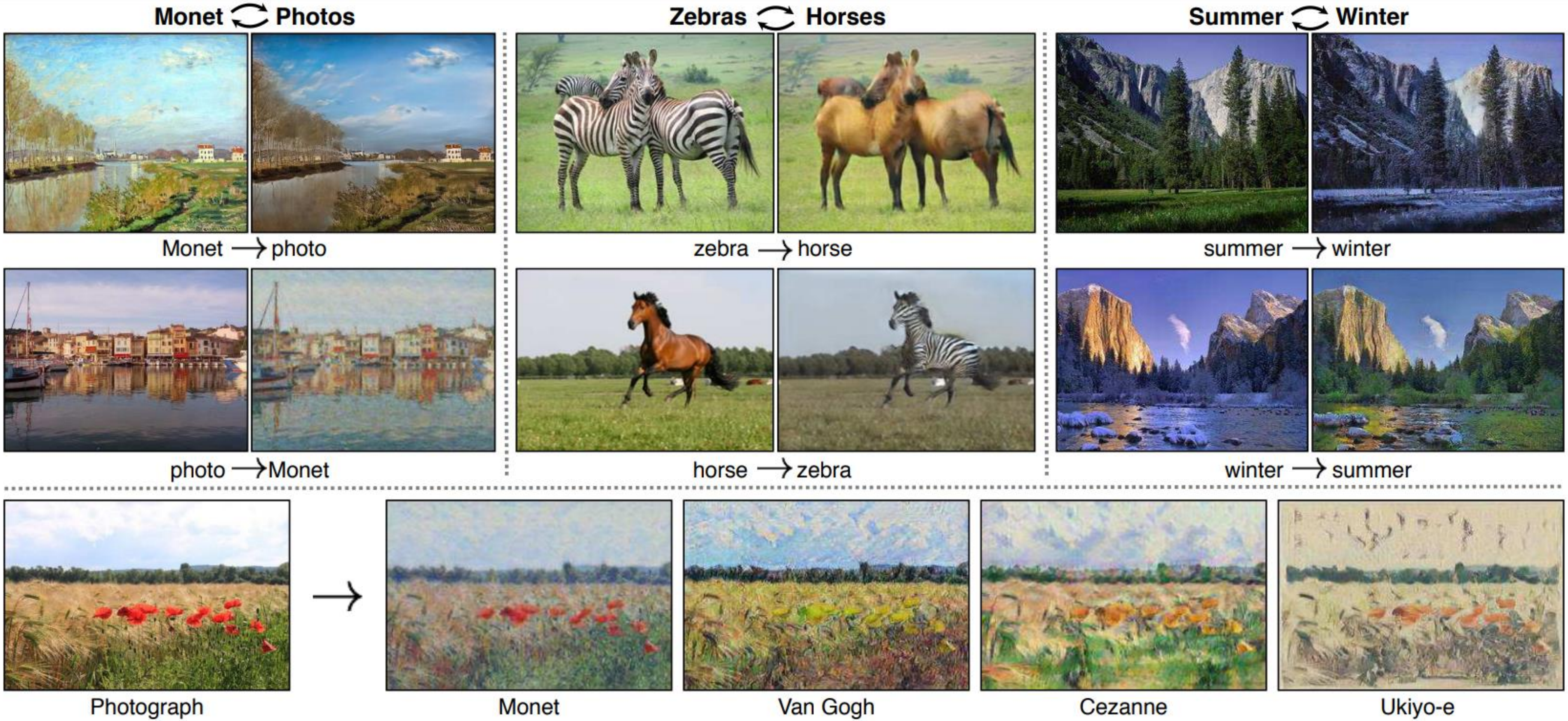
→ **Enforce cycle consistency to reduce space of possible mappings**

$$L_{\text{cyc}}(G, F) = \boldsymbol{E}_{\boldsymbol{x}}[\|\boldsymbol{x} - F(G(\boldsymbol{x}))\|_1] + \boldsymbol{E}_{\boldsymbol{y}}[\|\boldsymbol{y} - G(F(\boldsymbol{y}))\|_1]$$

Interestingly: Can be understood as training two autoencoders F(G()) and G(F()) with special internal structure

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. CVPR (pp. 2223-2232).

Unpaired Image-to-image Translation



Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. CVPR (pp. 2223-2232).

Unpaired Image-to-image Translation

Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [28]	0.40	0.10	0.06
BiGAN/ALI [7, 6]	0.19	0.06	0.02
Pixel loss + GAN [42]	0.20	0.10	0.04
Feature loss + GAN	0.06	0.04	0.01
CycleGAN (ours)	**0.52**	**0.17**	**0.11**
pix2pix [20]	0.71	0.25	0.18

Table 2: FCN-scores for different methods, evaluated on Cityscapes labels→photos.

Impressive results, but not yet comparable to paired performance.

Figure 12: Some failure cases of our method.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. CVPR (pp. 2223-2232).

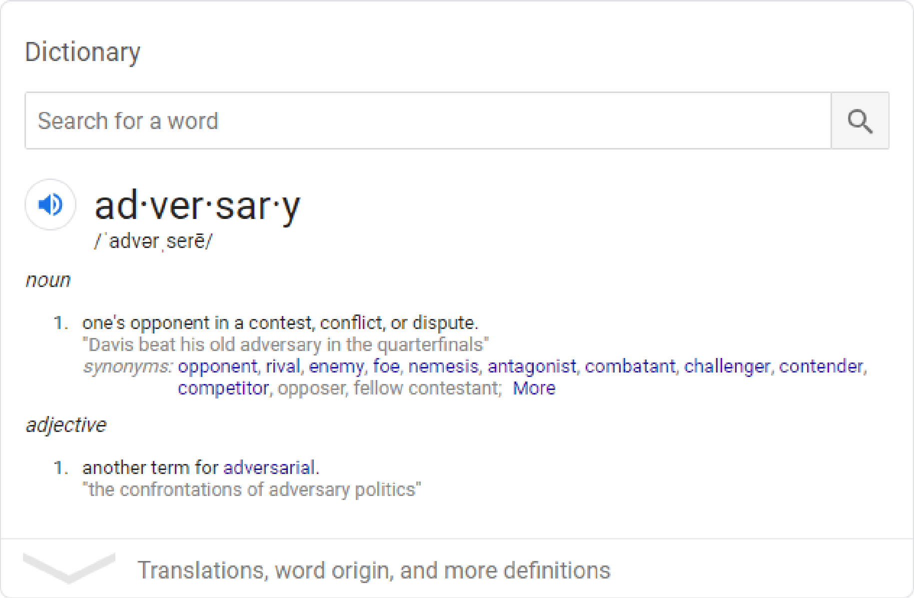
A Small Detour: Adversaries

Adversaries

This lecture is concerned with generative **adversarial** networks.

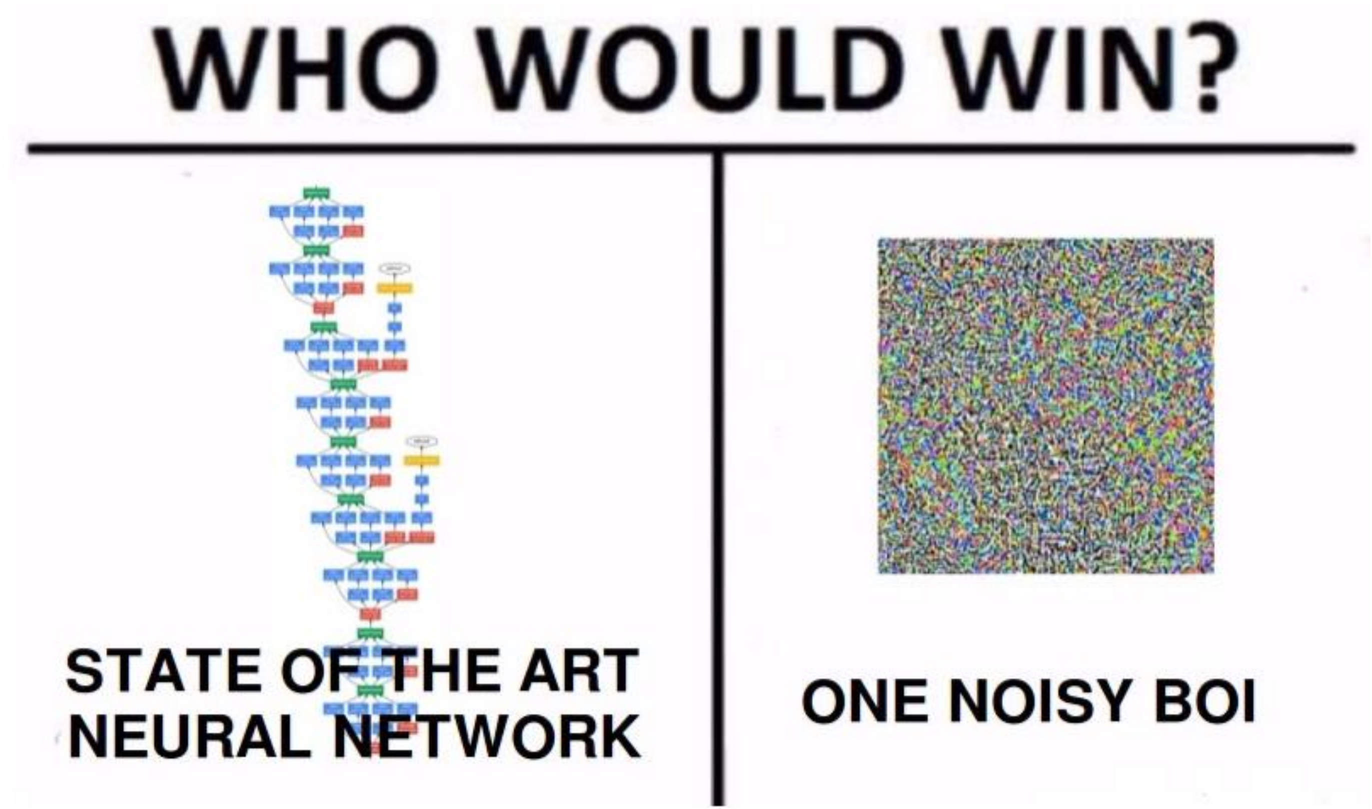
Adversaries

- Generator vs. discriminator
- Two player game
- Competing interests (minimax prob.)

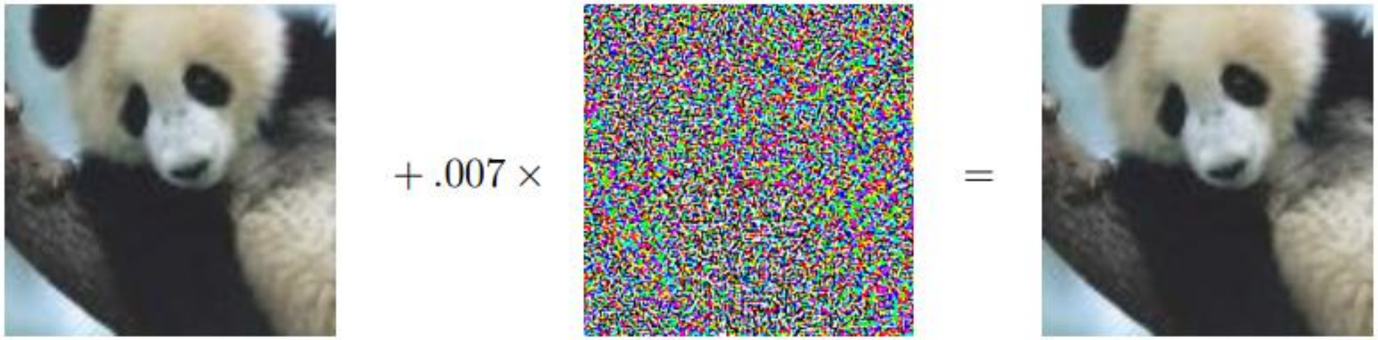


GANs are not to be confused with adversarial attacks!

Adversarial Attacks



Adversarial Attacks



$\boldsymbol{x}$ "panda" 57.7% confidence

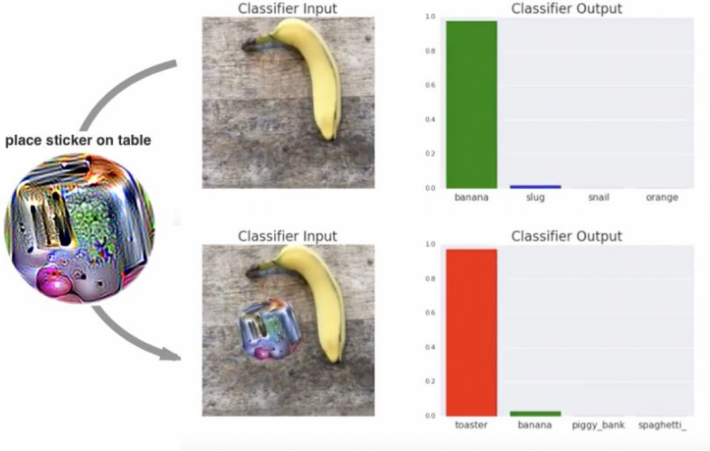
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "nematode" 8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ "gibbon" 99.3 % confidence

Figure 1: A demonstration of fast adversarial example generation applied to GoogLeNet (Szegedy et al., 2014a) on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet's classification of the image. Here our ϵ of .007 corresponds to the magnitude of the smallest bit of an 8 bit image encoding after GoogLeNet's conversion to real numbers.

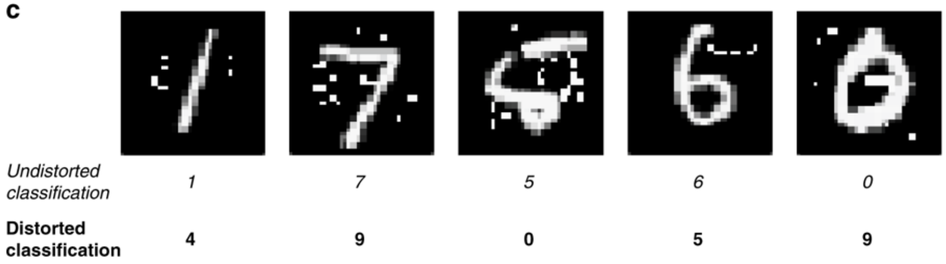
Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Adversarial Attacks



Some thoughts on adversarial attacks

- Interesting area to study (see sticker)
- Defense mechanisms not yet well known
  Best mechanism at this time: Feature denoising
- A bit of fuzz (also in medical)



But susceptibility to these attacks just reveals a larger problem:

→ **Despite breakthroughs, we are still in the infancy of machine learning**

Brown, T. B., Mané, D., Roy, A., Abadi, M., & Gilmer, J. (2017). Adversarial patch. *arXiv preprint arXiv:1712.09665*.
Zhou, Z., & Firestone, C. (2019). Humans can decipher adversarial images. Nature communications, 10(1), 1334.

Generative Adversarial Networks

Remarks

Concluding Remarks

Problem 1	What are the trade-offs between GANs and other generative models?
Problem 2	What sorts of distributions can GANs model?
Problem 3	How can we Scale GANs beyond image synthesis?
Problem 4	What can we say about the global convergence of the training dynamics?
Problem 5	How should we evaluate GANs and when should we use them?
Problem 6	How does GAN training scale with batch size?
Problem 7	What is the relationship between GANs and adversarial examples?

https://distill.pub/2019/gan-open-problems/, Augustus Odena – Google Brain Team

Generative Adversarial Networks

Questions?