Generative Adversarial Networks

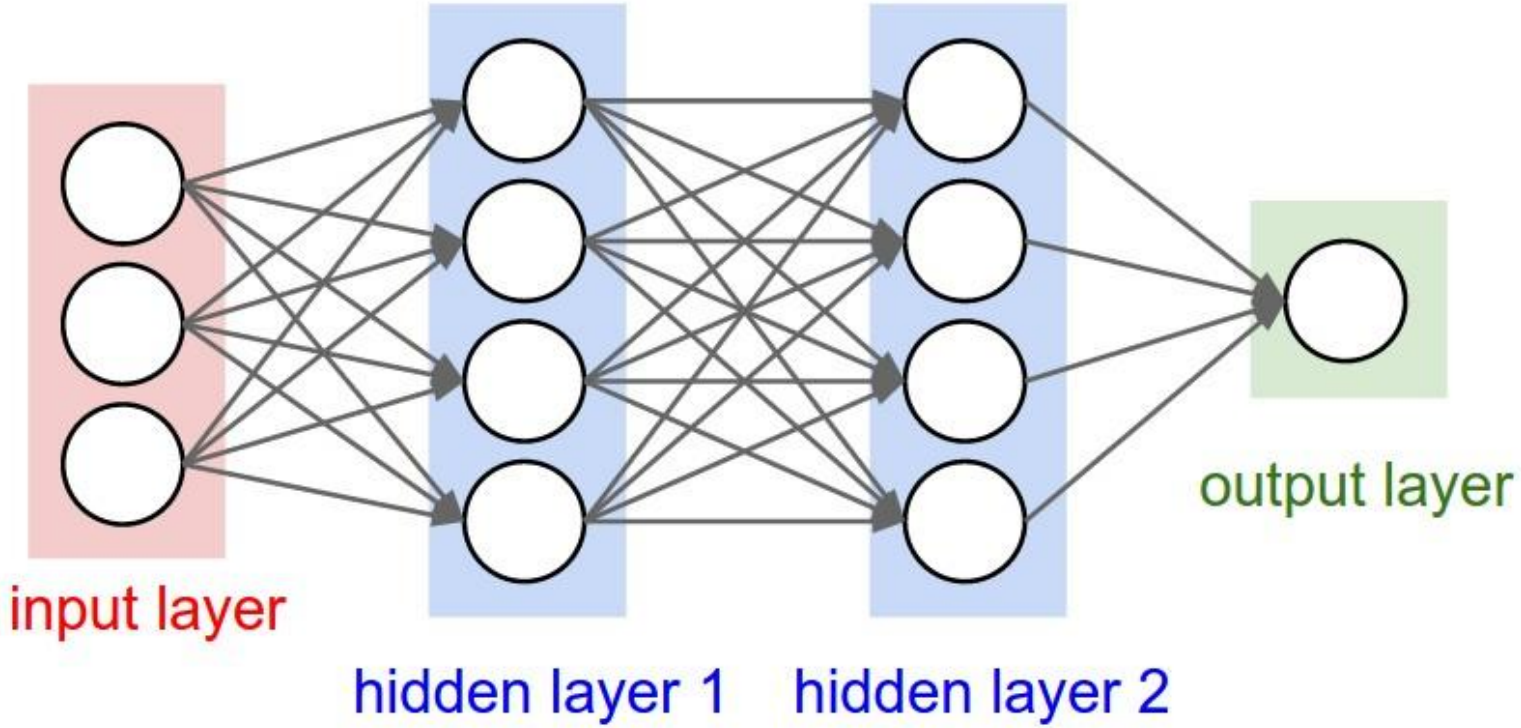
How to draw really ~~small~~ medium pictures of ~~cats~~ celebrities

Aaditya Prakash

10/31/2018

Lot of the slides borrowed from Nick Moran

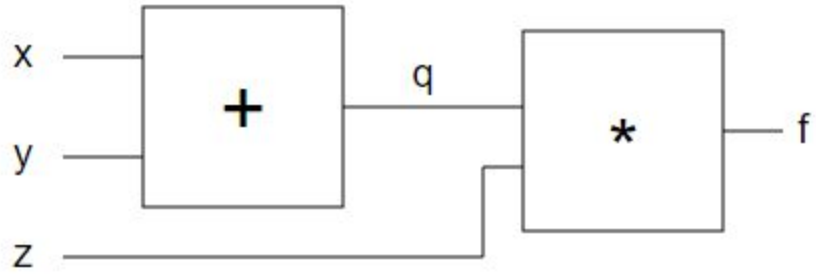
Review: Neural Networks



http://cs231n.github.io/neural-networks-1/

Review: Neural Networks

Neural networks are just very complicated functions.



$$f = z * [q*(x+y)]$$

These functions are differentiable.

Review: Neural Networks

- Neural networks have *parameters*, which we can adjust to make them better.

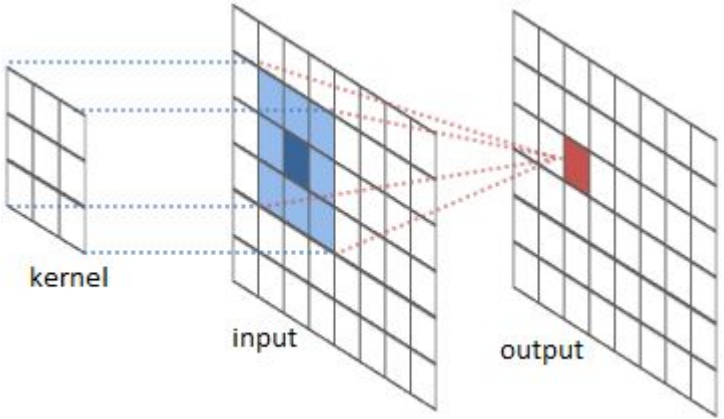
- Given an *loss function*, which measures how bad our network is, we can compute the gradient of our parameters with respect to our loss function.

- We then take small steps down the gradient to minimize our loss.

Review: Convolutions and CNNs

- A CNN works by applying a small filter (kernel) to each section of the input image to produce the output.

- The values of the filter are what we're trying to learn.



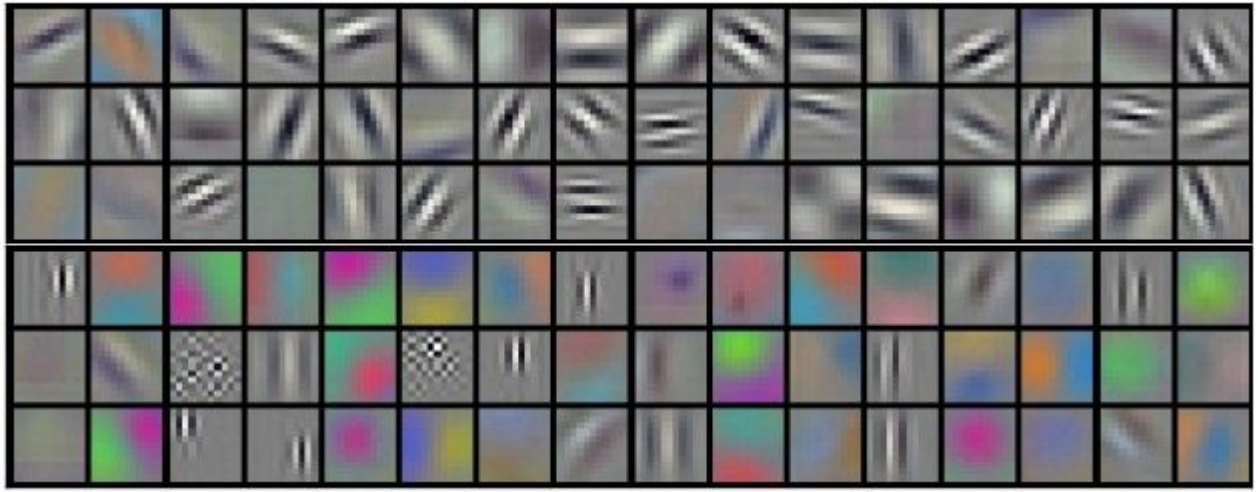
http://colah.github.io/posts/2014-07-Understanding-Convolutions/

Review: Convolutions and CNNs

$*$

1	0	-1
2	0	-2
1	0	-1

Review: Convolutions and CNNs

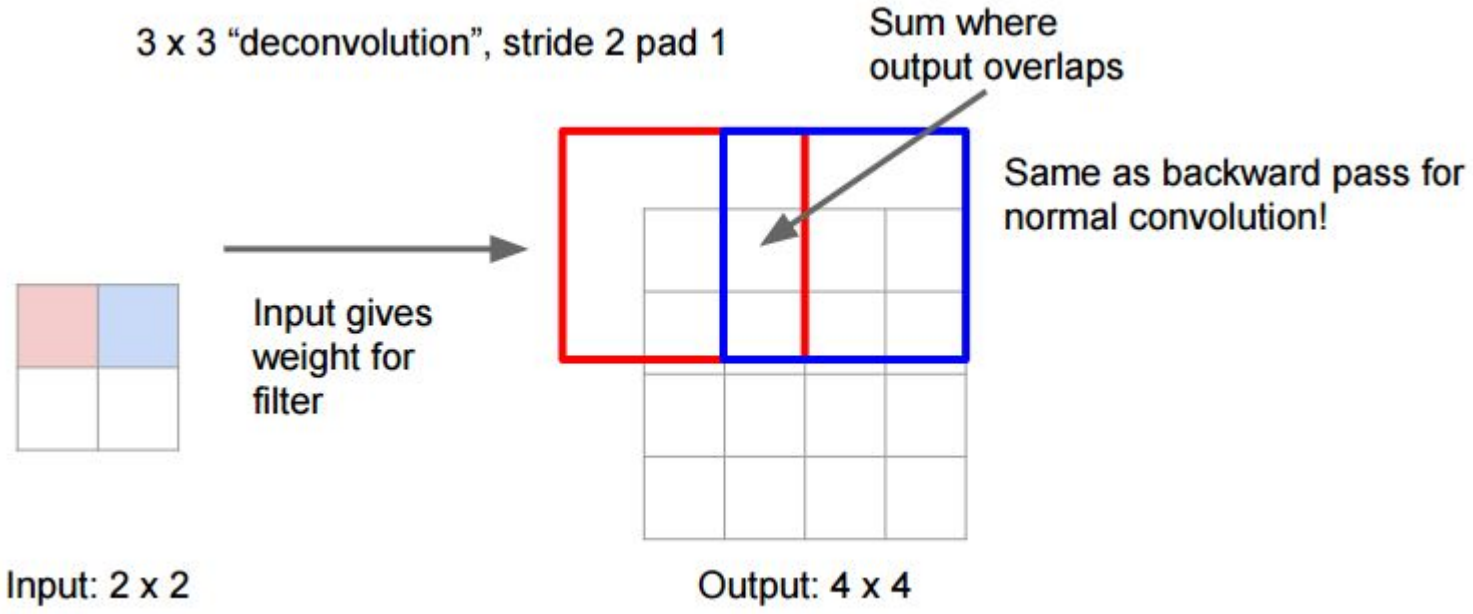


Filters from a trained convnet

http://cs231n.github.io/convolutional-networks/

Deconvolution/Convolution Transpose

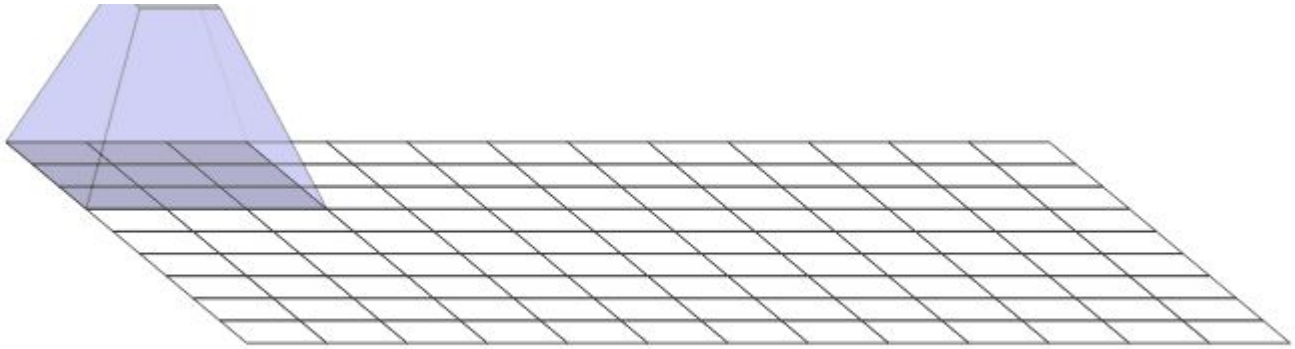
“Deconvolution” is a deceptive name - we’re not really undoing a convolution.



https://leonardoaraujosantos.gitbooks.io/

Deconvolution/Convolution Transpose

We can build an output image by layering different amounts of each filter over top of each other, like patches. The input tells us how much of each patch to use at each position.



http://distill.pub/2016/deconv-checkerboard/

Generative Models vs. Discriminative Models

- A generative model seeks to capture the distribution that generates your data.

  Learn a joint probability distribution $\longrightarrow P(x,y)$

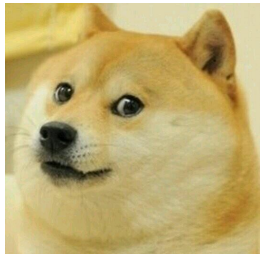
- A discriminative model only tries to tell the difference between two or more distributions.

  Learn a conditional probability distribution $\longrightarrow P(y|x)$

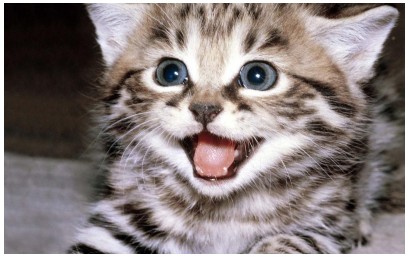
- To generate samples you just have to sample from the joint distribution.

Generative Models vs. Discriminative Models

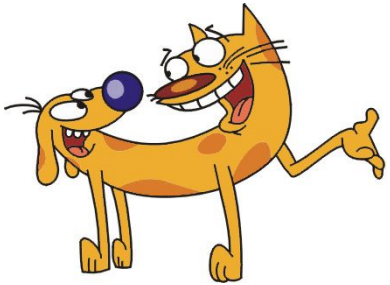
Discriminative model:



Pupper



Kittums



????

Generative Models vs. Discriminative Models

Generative Models:

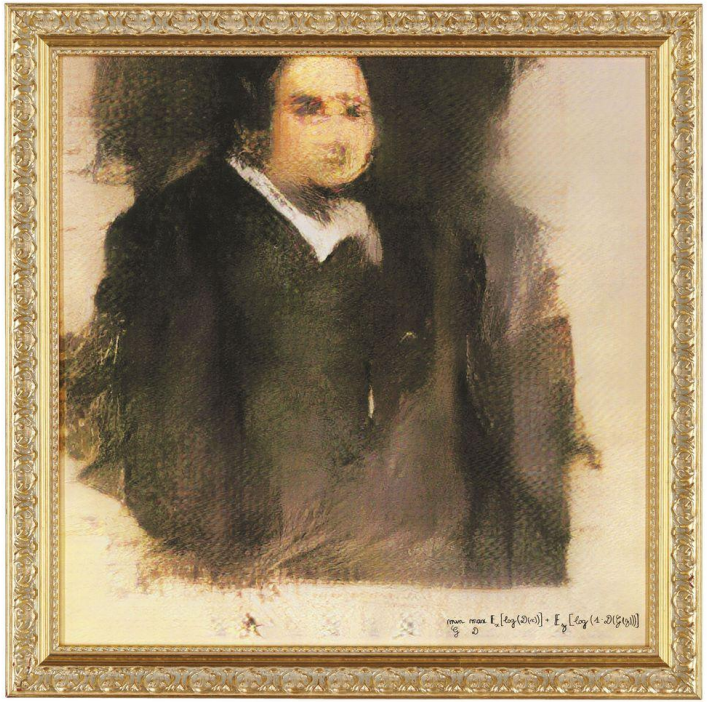
Alternating generated and real cat images

Draw me some cats!

https://github.com/aleju/cat-generator

Generative Models vs. Discriminative Models

Beauty lies in the eye of the beholder



GAN generated image auctioned for $432,500

Adversarial Systems

Sometimes in life, the solution to hard problems is fighting about it.

Adversarial systems work by having two agents with opposing goals that compete with each other to drive continuous improvement.

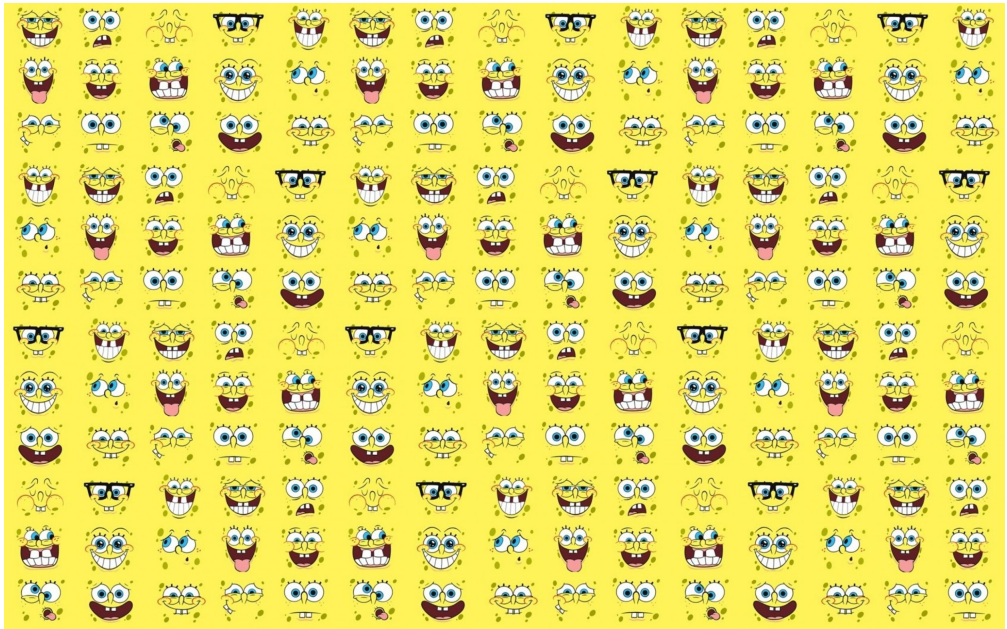
GANs - History

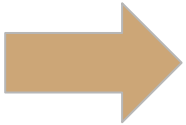
Introduced in the paper “**Generative Adversarial Nets**” by Goodfellow et al., published in NIPS 2014.

Since then, GANs have been an extremely fast-moving research area, with over ~~450~~ ~~2,700~~ 5,500 papers citing the original work in just a few years.

GANs: The Goal

Given a set of training images, be able to generate new plausible samples that are similar to the training set.





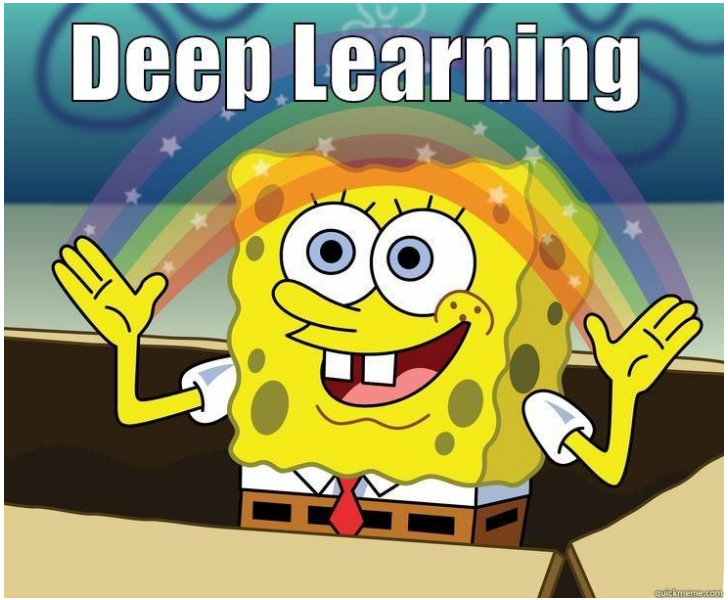
GANs: What is a “plausible sample”?

How do we tell whether we’re actually generating plausible images?

Humans can tell a plausible image when we see it, but I don’t have time to sit at a computer and evaluate millions of samples.

GANs: What is a “plausible sample”?

Idea: Neural networks are magic, so train a neural network to tell what is and is not a plausible sample.



GANs: Discriminating Plausible Samples

Goal: Given a set of real images and a set of fake images, classify new images as fake or real.

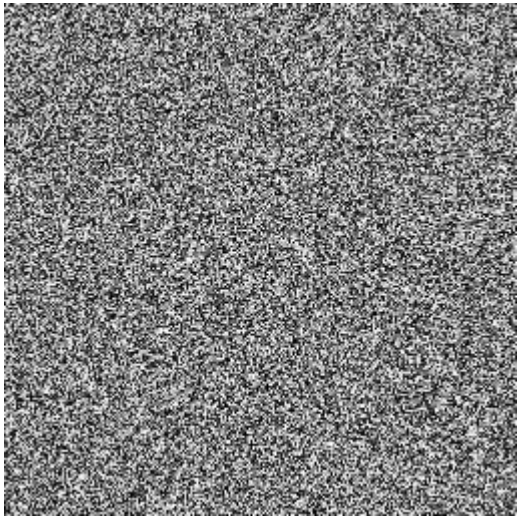
Seems Legit

Suspicious

GANs: Discriminating Plausible Samples

Problem: We need a bunch of fake images to train our discriminator.

It’s easy to generate a lot of images that are definitely not cats.

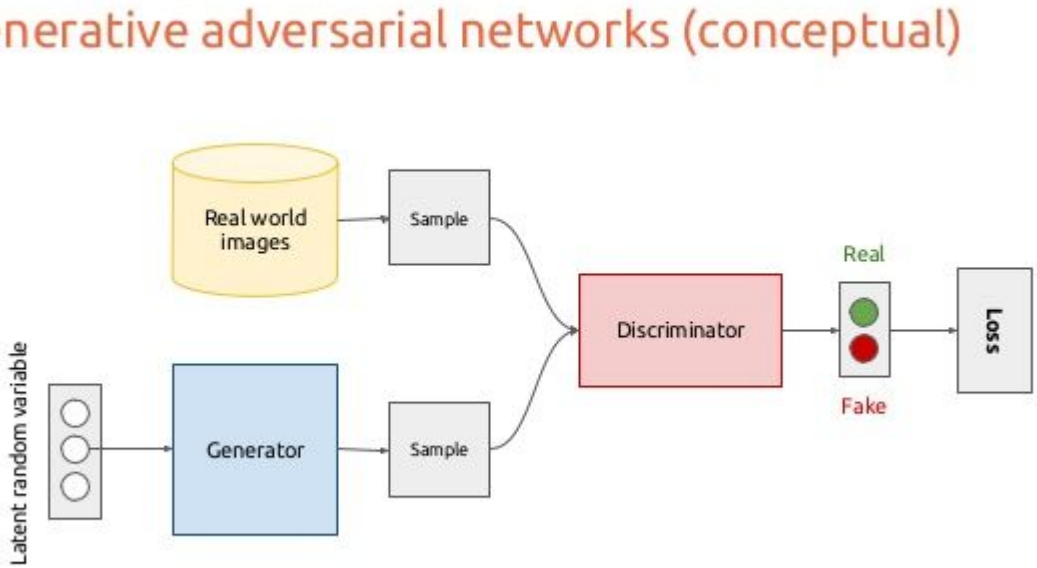


Not a cat

We need images which are pretty close.

Solution: Train the discriminator on the outputs of our generator.

GAN Model

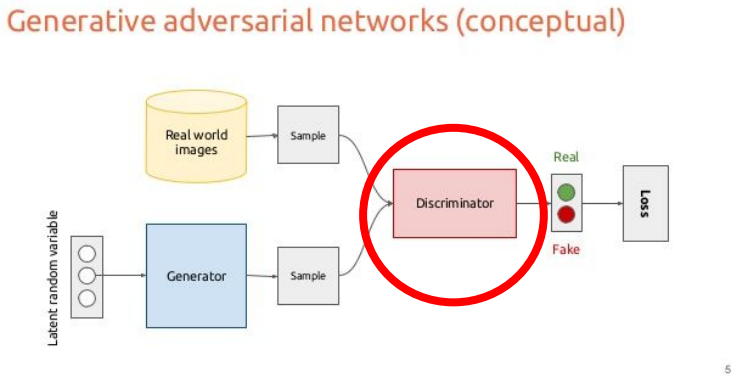


https://wiki.tum.de

GAN Model - Discriminator

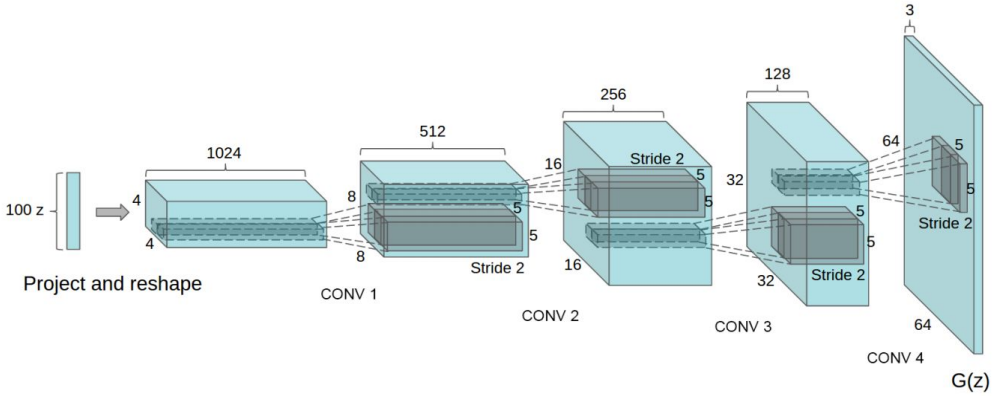
The discriminator is just a standard ConvNet.

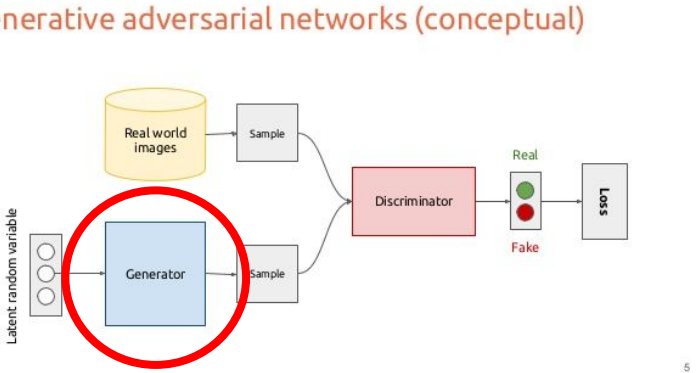
It takes in images, and tries to classify them as real or fake.



GAN Model - Generator

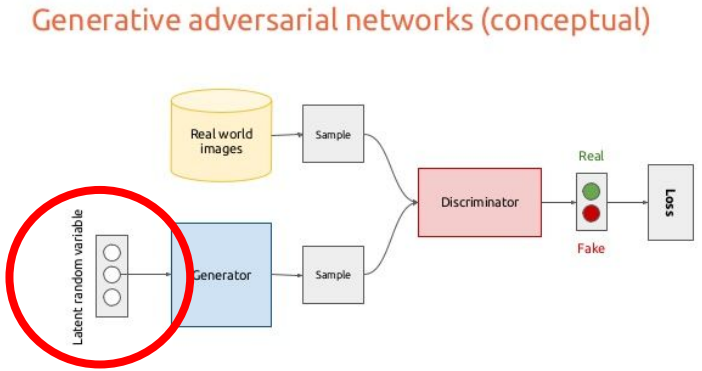
Our generator uses the ~~deconvolution~~ convolution transpose operation.





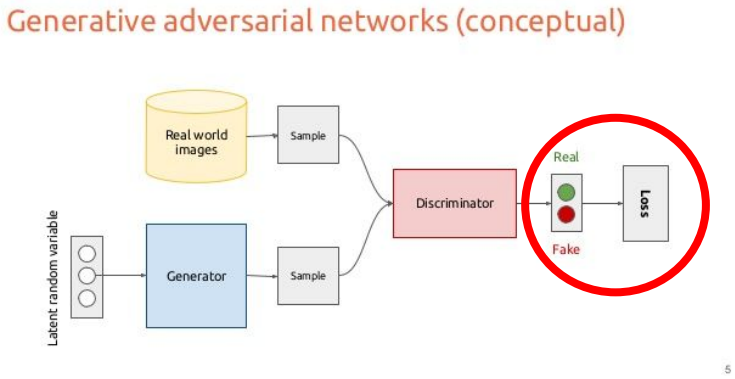
GAN Model - Latent Vector

- It’s no good if our generator always makes the same output image, but our generator is deterministic.
- The input to our generator will be a vector of random numbers drawn from a simple distribution - usually gaussian or uniform.
- Each point from this ‘latent space’ will be mapped by the generator to an output image.



GAN Model - Loss Function

- In order to train our model, we need to define a loss function.
- The discriminator's loss needs to reward it for correctly classifying real images as real and fake images as fake.
- The generator's loss needs to reward it for fooling the discriminator.



GAN Model - Loss Function (Discriminator)

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

https://arxiv.org/abs/1406.2661

GAN Model - Loss Function (Discriminator)

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

- $x^{(i)}$ are samples from our real image dataset.
- $z^{(i)}$ are vectors drawn from our latent space
- D() is the function represented by our discriminator - D = 1 means the input is real, D = 0 means the input is fake.
- G() is the function represented by our generator - $G(z^{(i)})$ will be an image
- *m* is the number of samples we take in a batch

GAN Model - Loss Function (Discriminator)

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

- Our discriminator tries to **maximize** this function.
- Maximizing log D($x^{(i)}$) means that we're outputting values close to 1 for real images.
- Maximizing log(1 - D(G($z^{(i)}$))) means that we're outputting values close to 0 for fake images.

GAN Model - Loss Function (Generator)

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

- Our generator will try to **minimize** this function.

- Minimizing log(1-D(G($z^{(i)}$))) means that we're getting the discriminator to output values close to 1 for our generated samples.

GAN Model - Loss Function

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

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

This is the adversarial part - D and G have competing objectives

GAN Model - Adversarial Training

- If we have two competing objectives, what does it mean to optimize our GAN?

- Idea: Update G to make it a little better against D, then update D to make it a little better against G, and continue alternating.

- Goal: Eventually, both G and D will be really good at their tasks.

GAN Model - Adversarial Training

1. Draw some samples from z, and compute G(z), which are images
2. Compute D(G(z)), which is an evaluation of how good G's images are
3. Compute the gradients of G with respect to the loss function - recall that D and G are differentiable neural networks, so D(G(z)) is differentiable.
4. Adjust the parameters of G using the gradients to make it a little better at fooling D - do not update the parameters of D yet.
5. Draw some samples from X and z, and compute G(z).
6. Feed X and G(z) to D, to compute D(X) and D(G(z)).
7. Compute the gradients of D with respect to its loss function.
8. Adjust the parameters of D to make it a little better at telling real images from fake.

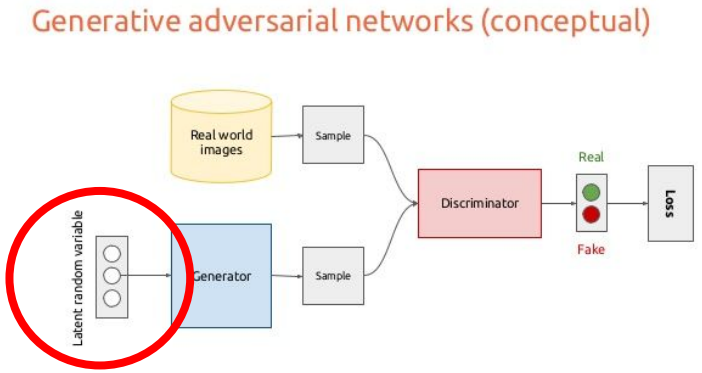
GAN Model - Adversarial Training

https://github.com/carpedm20/DCGAN-tensorflow

GAN Model - Latent Space

- Recall that the input to our generator was random numbers drawn from a simple distribution, Z.

- We can sample from the space of generated images by sampling from Z, but what else can we do?

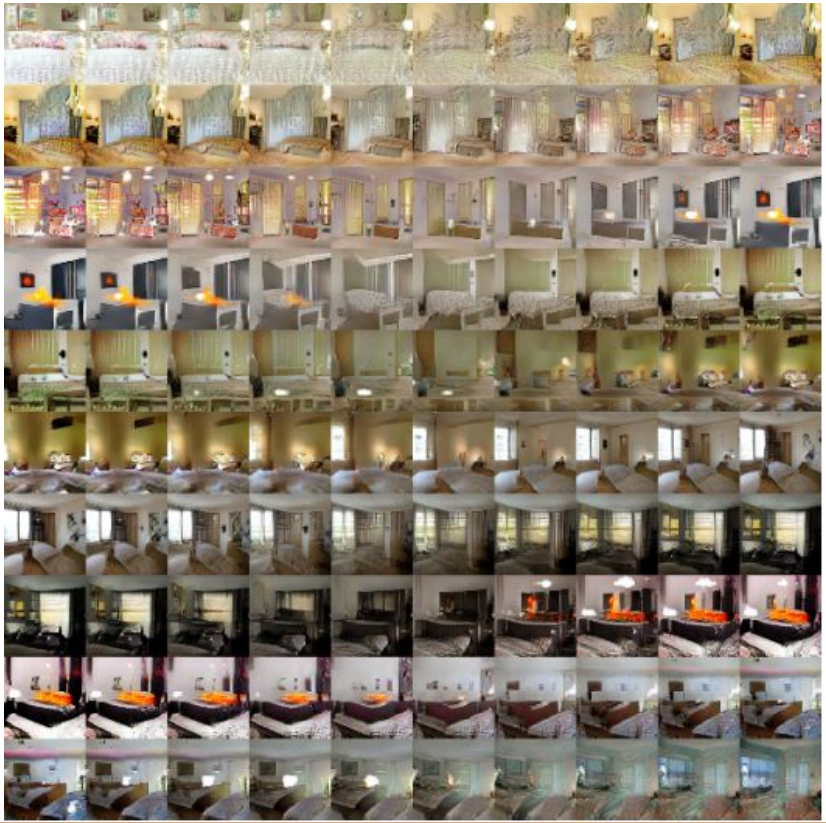


GAN Model - Latent Space

If we draw two samples, z_1 and z_2 from Z, what happens if we interpolate between them?

https://arxiv.org/pdf/1406.2661.pdf

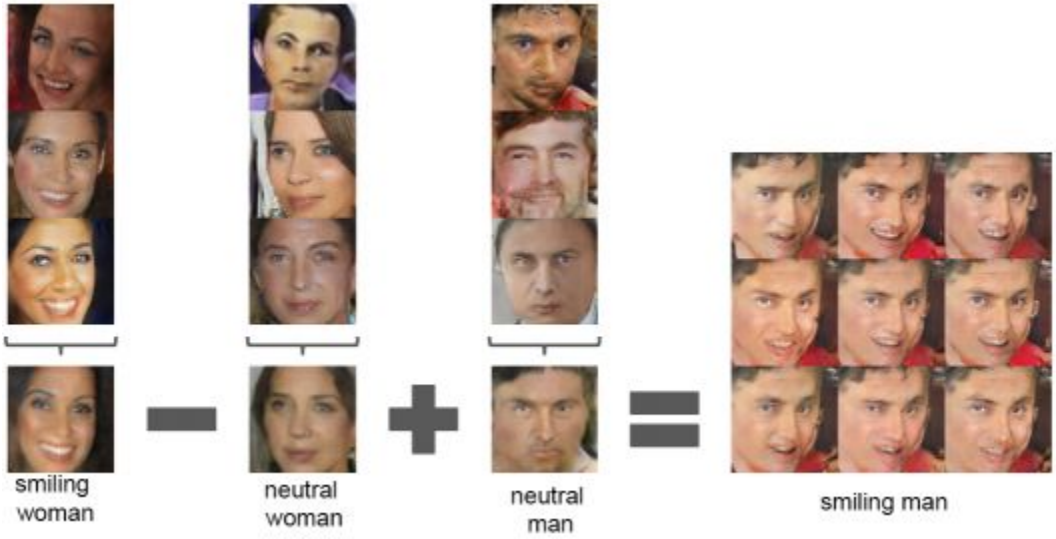
GAN Model - Latent Space



https://arxiv.org/pdf/1511.06434.pdf

GAN Model - Latent Space

We can average the z vectors for multiple images with a particular attribute to find to find a representative z for that sample. We can then do vector arithmetic with these z values.



https://arxiv.org/pdf/1511.06434.pdf

GAN Model - Latent Space

We can also interpolate along the axis of an average attribute vector.

https://arxiv.org/pdf/1511.06434.pdf

GAN Model - Pitfalls

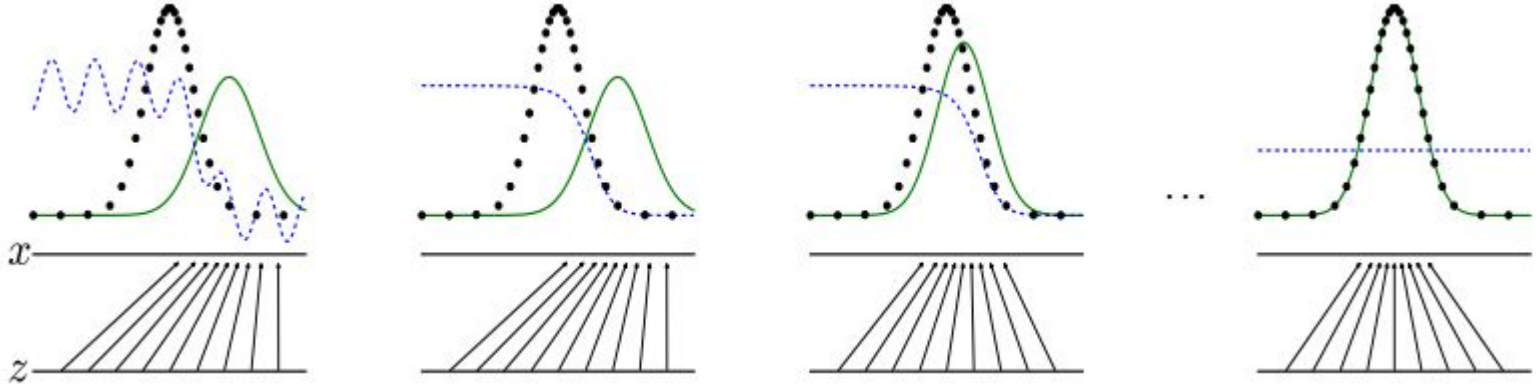
- What if G learns to only make one really accurate sample? Will D just accept that realistic sample?

- Imagine G is trying to make counterfeit money, and D is trying to detect counterfeit money.

- If G can make a perfect counterfeit copy of one bill with a fixed serial number, should D classify bills with that serial number as real or fake?

- Ideally, G needs to approximate the entire distribution of real samples to fool D.

GAN Model - Data Distributions



- Dots are real data, the green line is the output of G, and the blue line is the output of D.
- We can think of D as approximating the difference (or divergence) between the real distribution and the distribution that G outputs.
- We can think of G as transforming the simple z distribution to the target X distribution

https://arxiv.org/pdf/1406.2661.pdf

GAN Model - Pitfalls

- What if one of our networks becomes really good before the other?
- Imagine trying to learn to play chess by repeatedly playing against a grandmaster. Any small change to your strategy will still result in a loss.
- We need G and D to stay in balance or else training might grind to a halt.

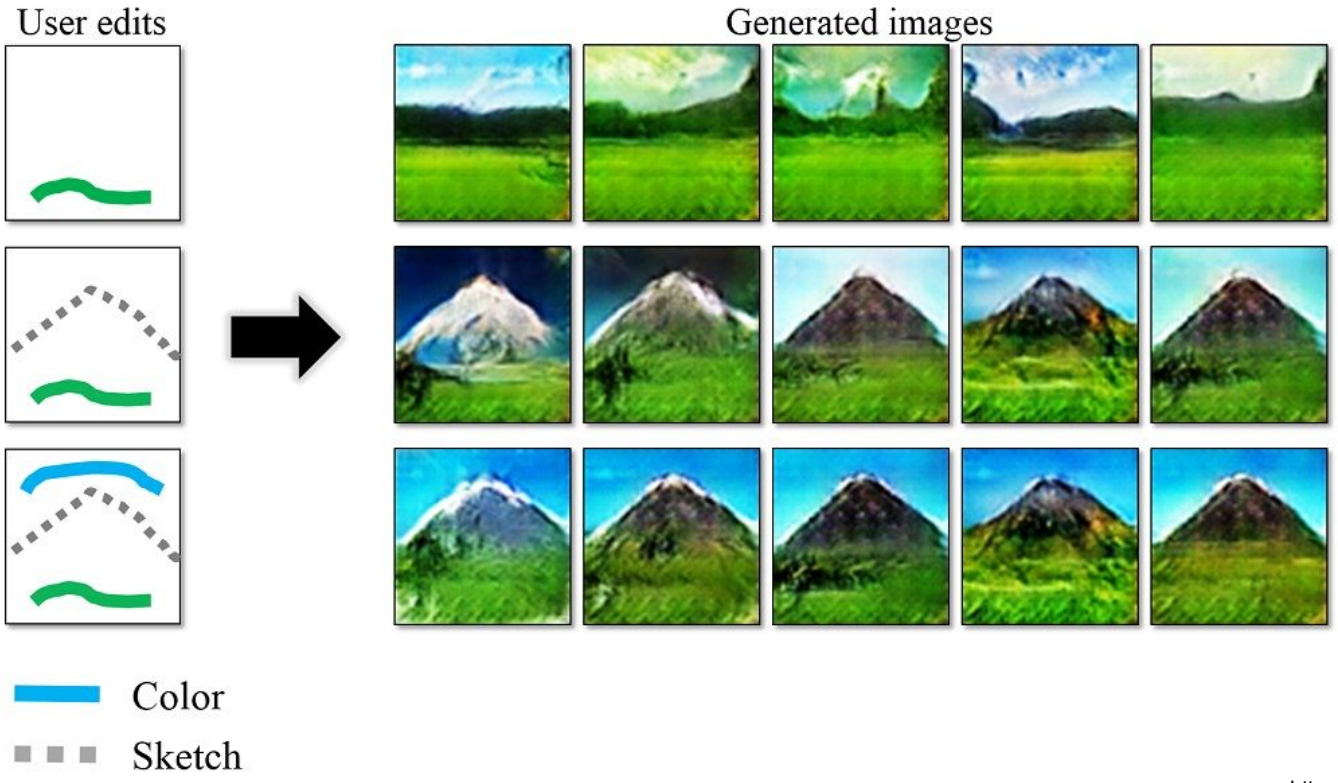
VS

GAN Model - Pitfalls

- GANs tend to struggle when given too wide a variety of real images.
- Most GAN papers focus on one or a few classes - only faces, only digits, only birds, etc.
- GANs also tend to struggle when asked to produce larger outputs. 32x32 and 64x64 work well, 128x128 and larger are difficult.

https://arxiv.org/pdf/1606.03498.pdf

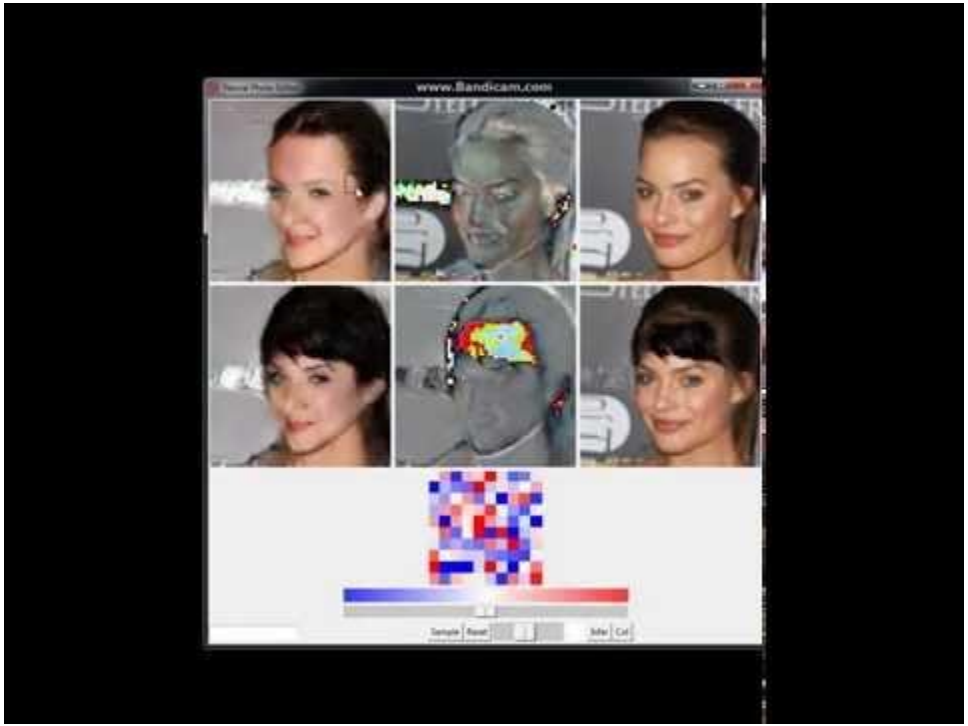
GANs - Nifty Results and Demos (iGAN)



https://github.com/junyanz/iGAN

GANs - Nifty Results and Demos (iGAN)

GANs - Nifty Results and Demos (Photo Editor)



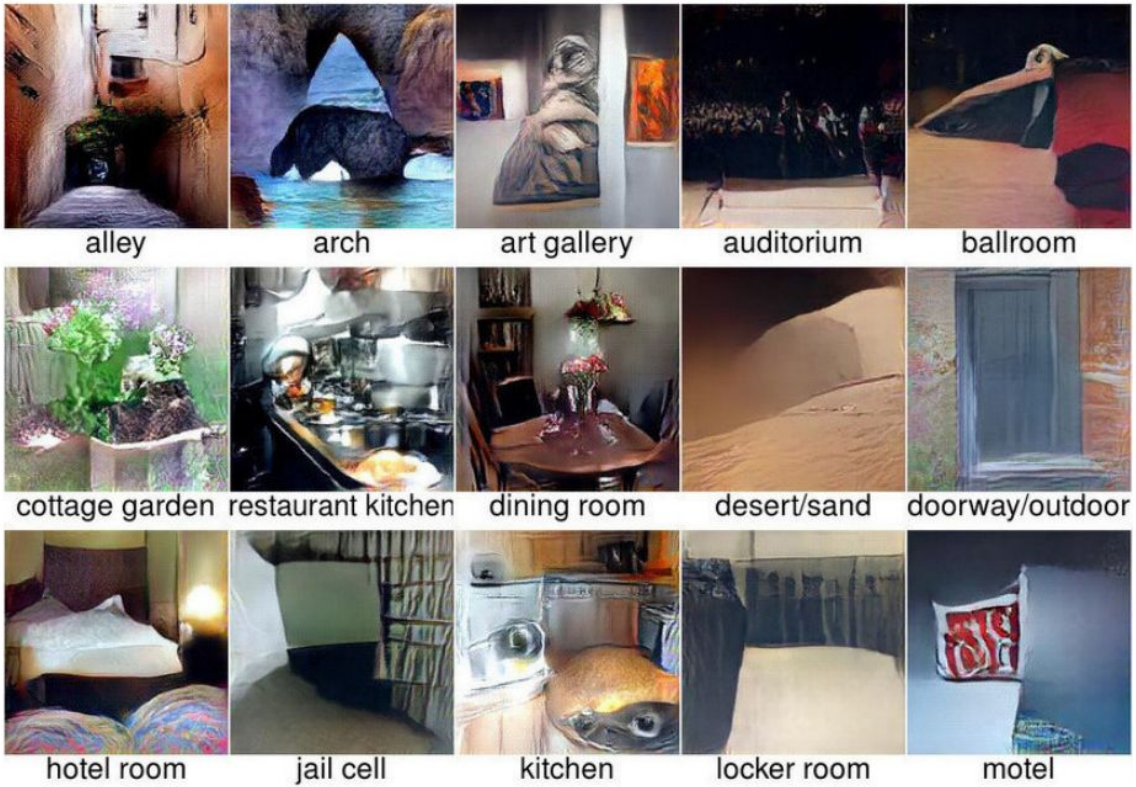
GANs - Nifty Results and Demos (StackGAN)

https://arxiv.org/pdf/1612.03242.pdf

GANs - Nifty Results and Demos (PPGN)

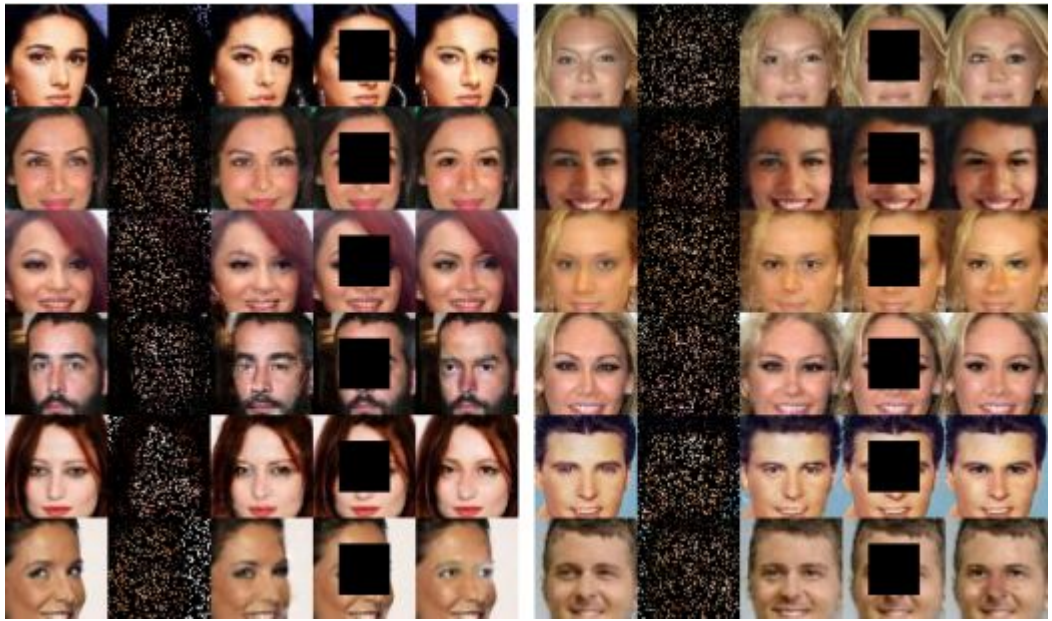
http://www.evolvingai.org/files/nguyen2016ppgn__v1.pdf

GANs - Nifty Results and Demos (PPGN)



http://www.evolvingai.org/files/nguyen2016ppgn__v1.pdf

GANs - Nifty Results and Demos (inpainting)



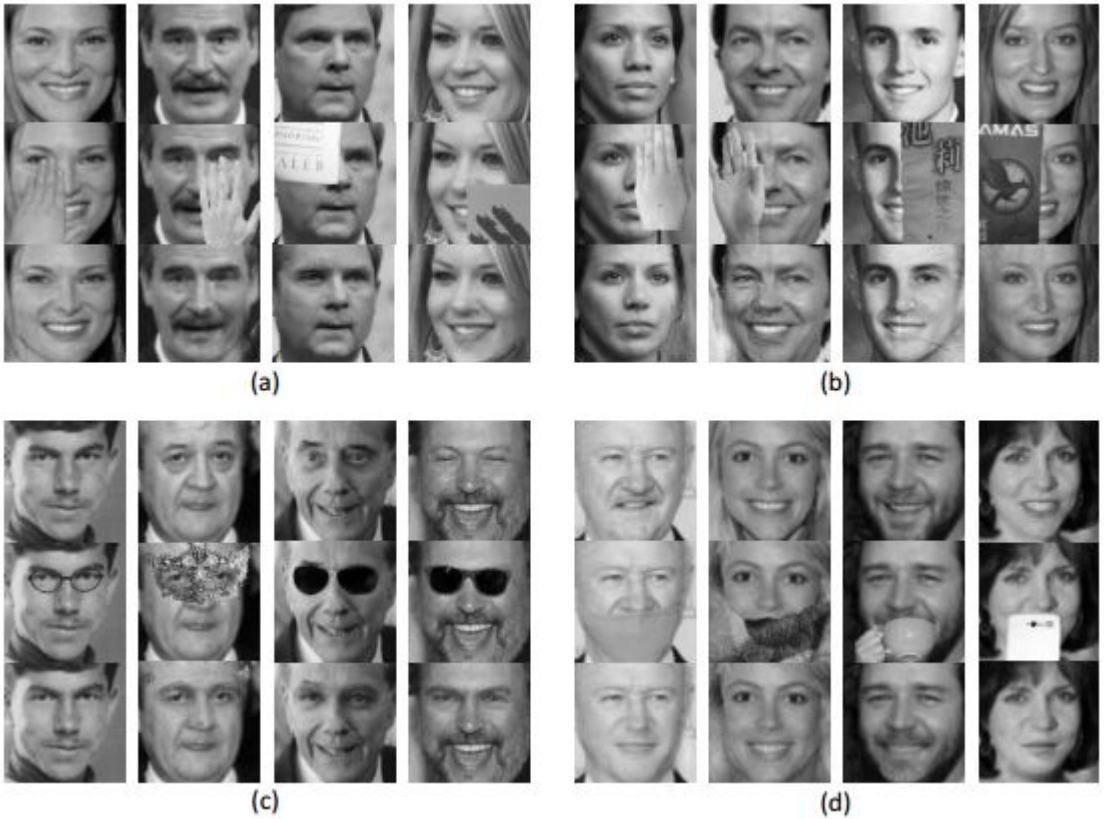
https://arxiv.org/pdf/1607.07539.pdf

GANs - Nifty Results and Demos (super-res)

Low-res, bicubic, GAN, ground truth

https://github.com/david-gpu/srez

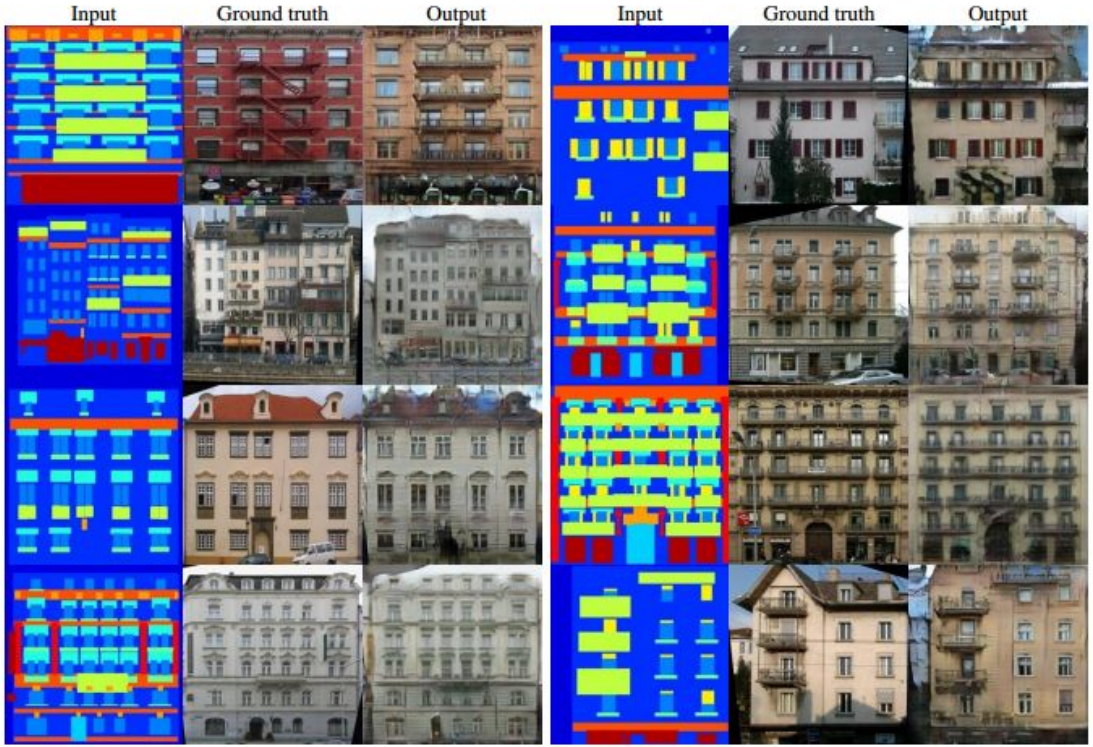
GANs - Nifty Results and Demos (de-occlusion)



GANs - Nifty Results and Demos (img-to-img)

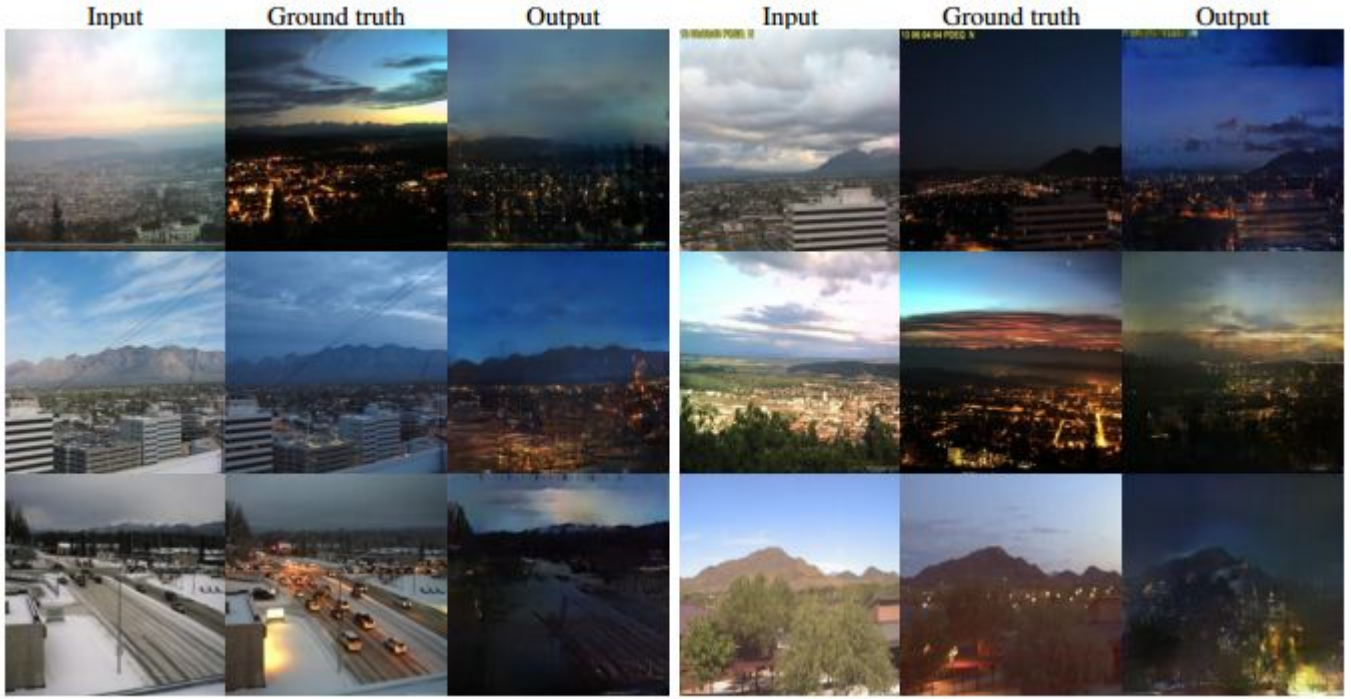
https://arxiv.org/pdf/1611.07004v1.pdf

GANs - Nifty Results and Demos (img-to-img)



https://arxiv.org/pdf/1611.07004v1.pdf

GANs - Nifty Results and Demos (img-to-img)



https://arxiv.org/pdf/1611.07004v1.pdf

GANs - Nifty Results and Demos (sketch->img)

https://affinelayer.com/pixsrv/

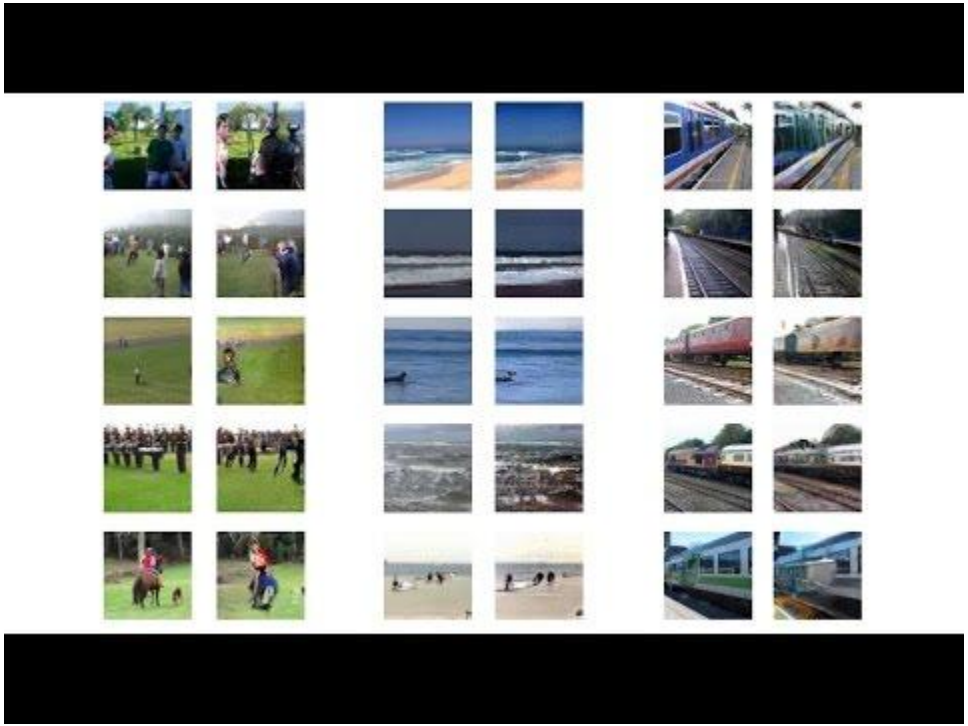
GANs - Nifty Results and Demos (CycleGAN)

winter Yosemite → summer Yosemite

summer Yosemite → winter Yosemite

Horse → Zebra

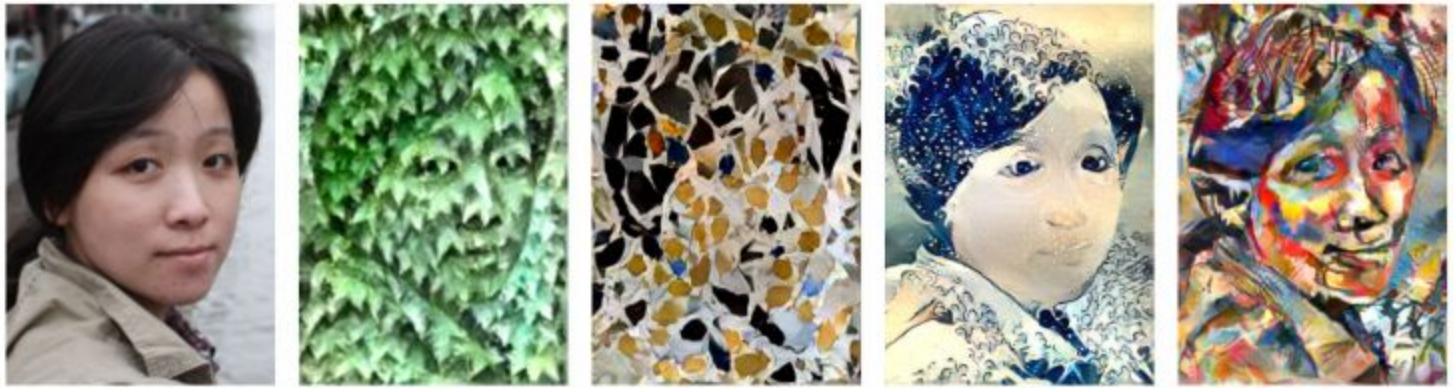
GANs - Nifty Results and Demos (video prediction)



GANs - Nifty Results and Demos (style transfer)

https://arxiv.org/pdf/1604.04382.pdf

GANs - Nifty Results and Demos (style transfer)



https://arxiv.org/pdf/1604.04382.pdf

GANs - Nifty Results and Demos (attribute manip)

(a) *Glasses*

(b) *No_beard*

(c) *Mouth_open*

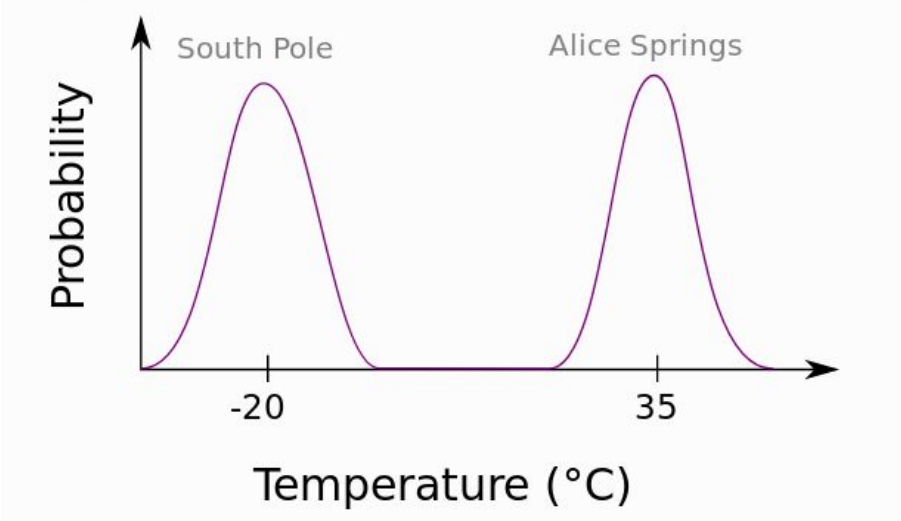
(d) *Smile*

(e) *Male*

(f) *Young*

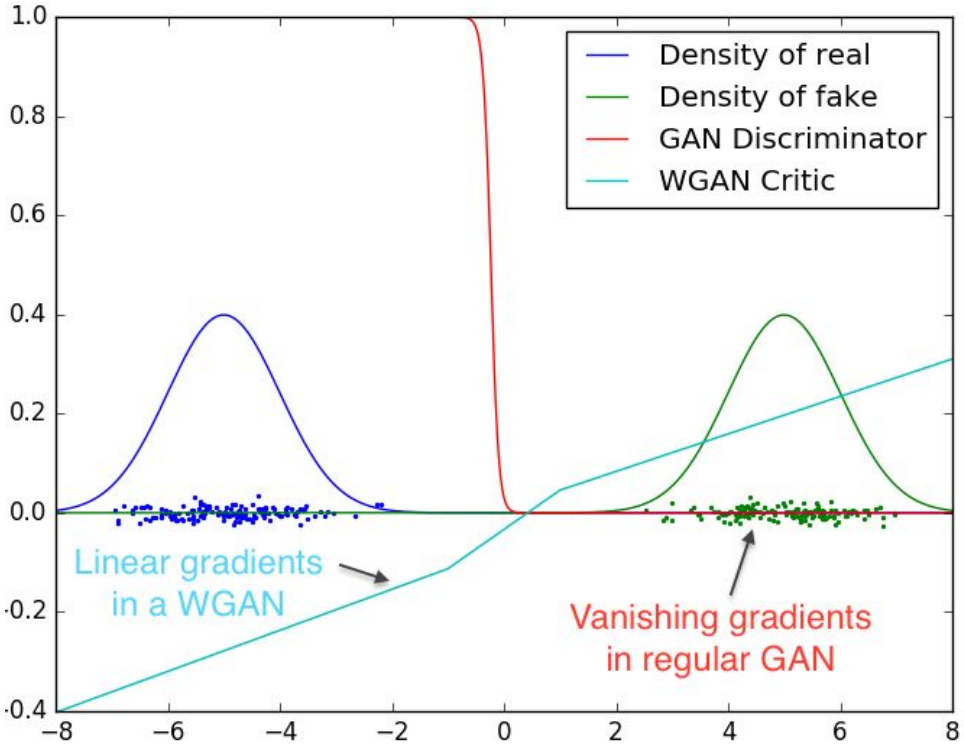
https://arxiv.org/pdf/1612.05363.pdf

Mode Collapse



http://aiden.nibali.org/blog/2017-01-18-mode-collapse-gans/

Problems - Wasserstein GAN

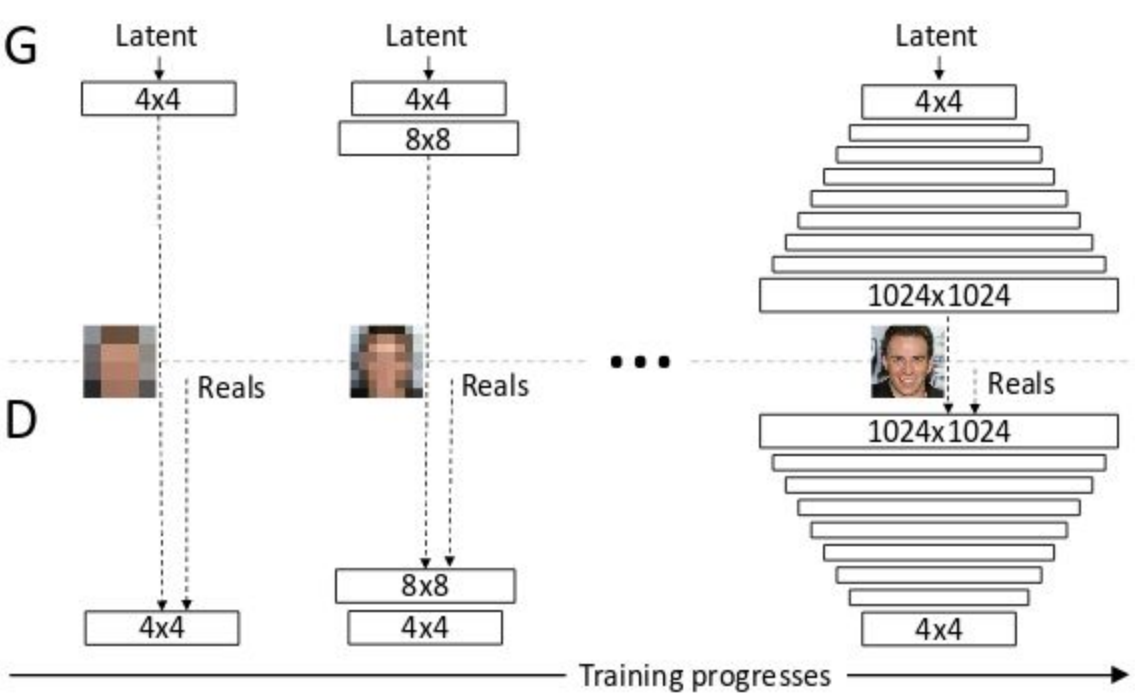


https://www.alexirpan.com/2017/02/22/wasserstein-gan.html

Progressive Growing of GANs (NVIDIA)

Learn coarse features well before attempting to learn fine features.

Interpolate between sizes (nn filter, avg pooling)

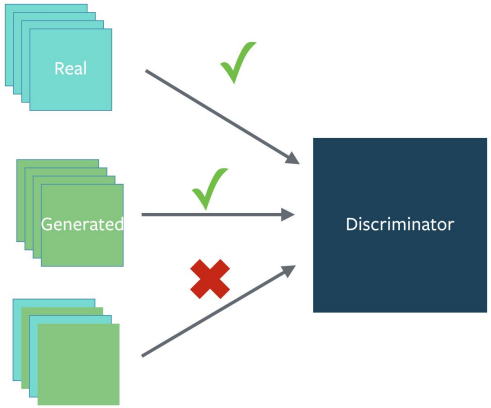

G
Latent
4x4
Latent
4x4
8x4
Latent
4x4
1024x1024
Reals
Reals
Reals
D
1024x1024
8x8
4x4
4x4
4x4
Training progresses

All Together:

Figure 10: Top: Our CELEBA-HQ results. Bottom: The nearest neighbors found from the training data, using the center crop (half vertically, half horizontally) and L_1 distance in pixel space.

GAN Hacks

1. Normalize the inputs
   a. Normalize the images between -1 and 1
   b. Tanh as the last layer of the generator output
2. Use a spherical Z
   a. Dont sample from a Uniform distribution
   b. Sample from a gaussian distribution
3. Batch Norm
   a. Different mini-batches for real and fake images
4. Optimizer
   a. Use SGD for Discriminator
   b. Use Adam for Generator



Thanks !

More hacks: https://github.com/soumith/ganhacks