CPSC 540: Machine Learning

VAEs and GANs

Winter 2020

Density Estimation Strikes Back

- One of the hottest topic in machine learning: density estimation?
  - In particular, deep learning for density estimation.

- Very fast-moving, but two most-popular methods are:
  - Variational autoencoders (VAEs).
  - Generative adversarial networks (GANs).

- We previously focused a lot on density estimation for digits:

Figure 3: Digits obtained by linearly interpolating between coordinates in z space of the full model.

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

Density Estimation Strikes Back

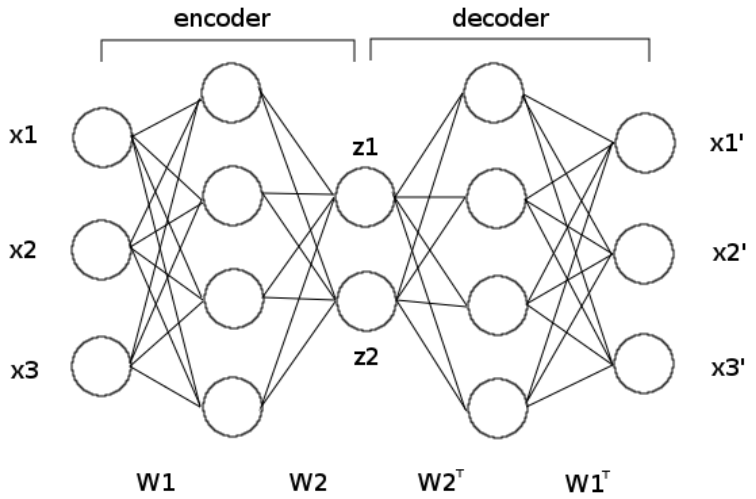
- These models are showing promising results going beyond digits:

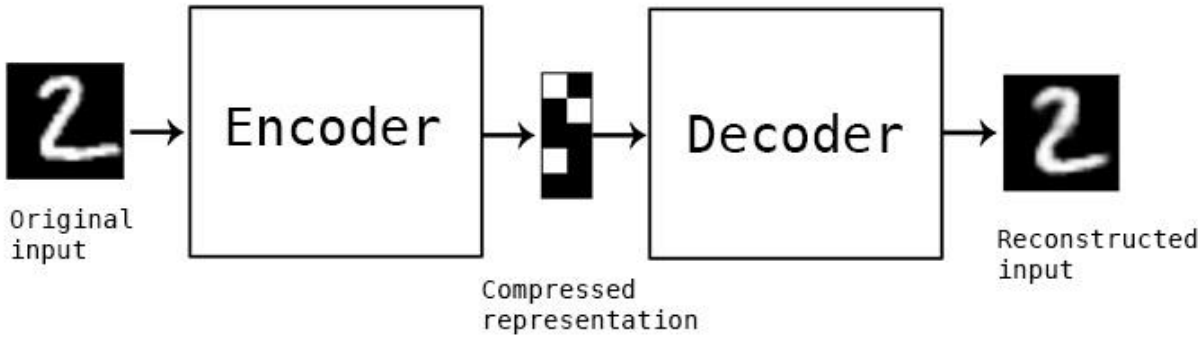
Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative 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)

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

Autoencoders

- Autoencoders are an unsupervised deep learning model:
  - Use the inputs as the output of the neural network.





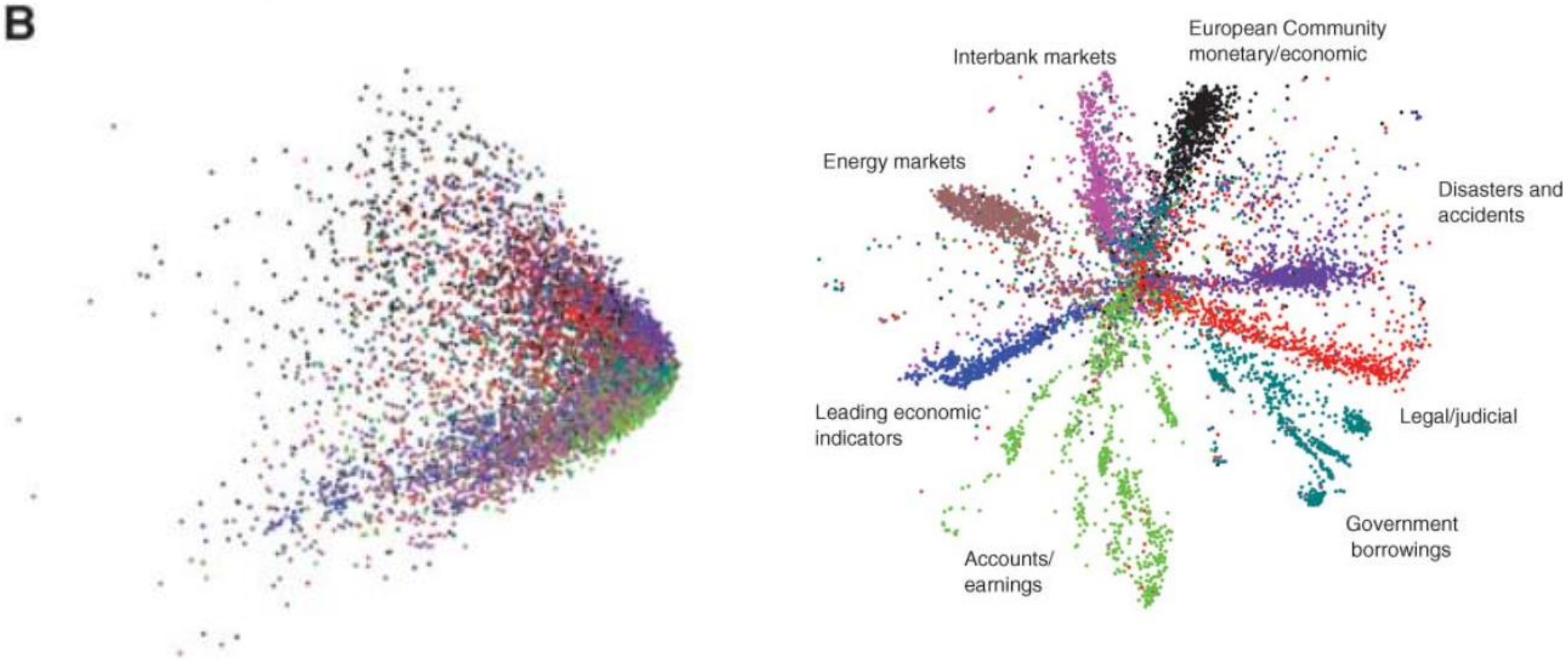
  - Middle layer could be latent features in non-linear latent-factor model.
    - Can do outlier detection, data compression, visualization, etc.
  - A non-linear generalization of PCA (old idea, never really popular).

http://inspirehep.net/record/1252540/plots
https://blog.keras.io/building-autoencoders-in-keras.html

Autoencoders for Visualization

PCA

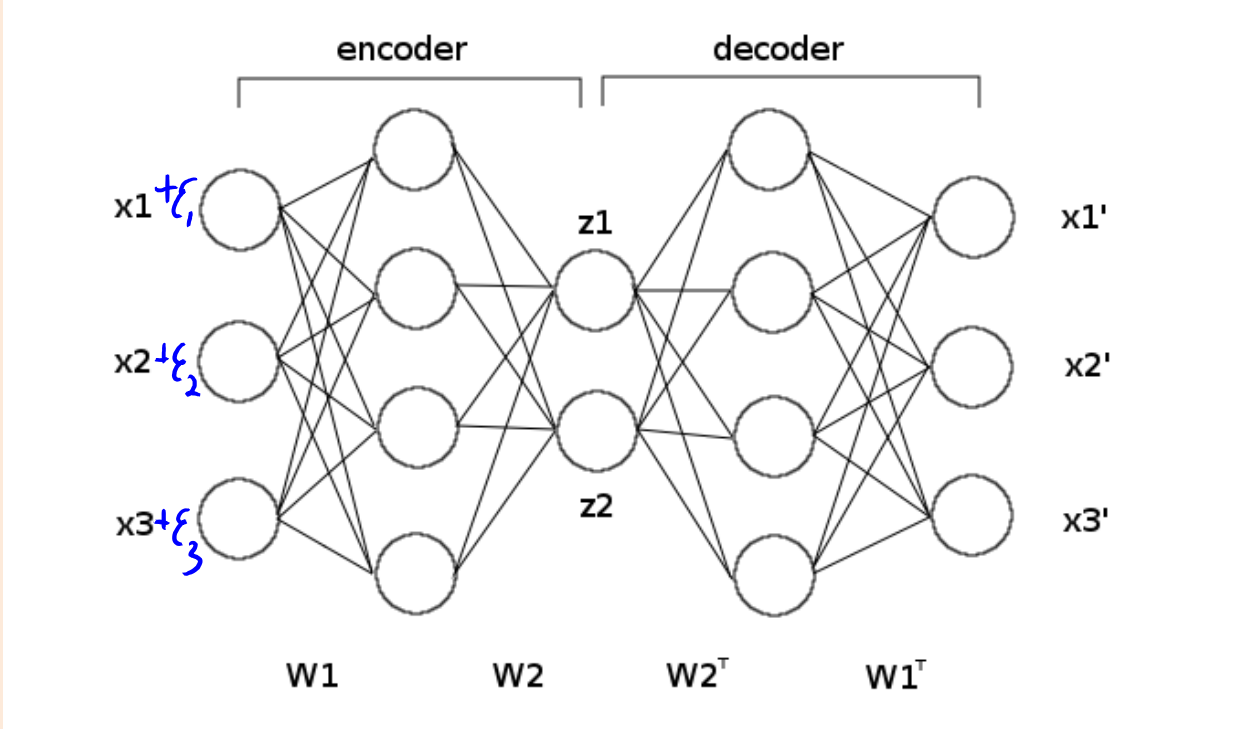
Autoencoder



https://www.cs.toronto.edu/~hinton/science.pdf

Denoising Autoencoder

- Denoising autoencoders add noise to the input:

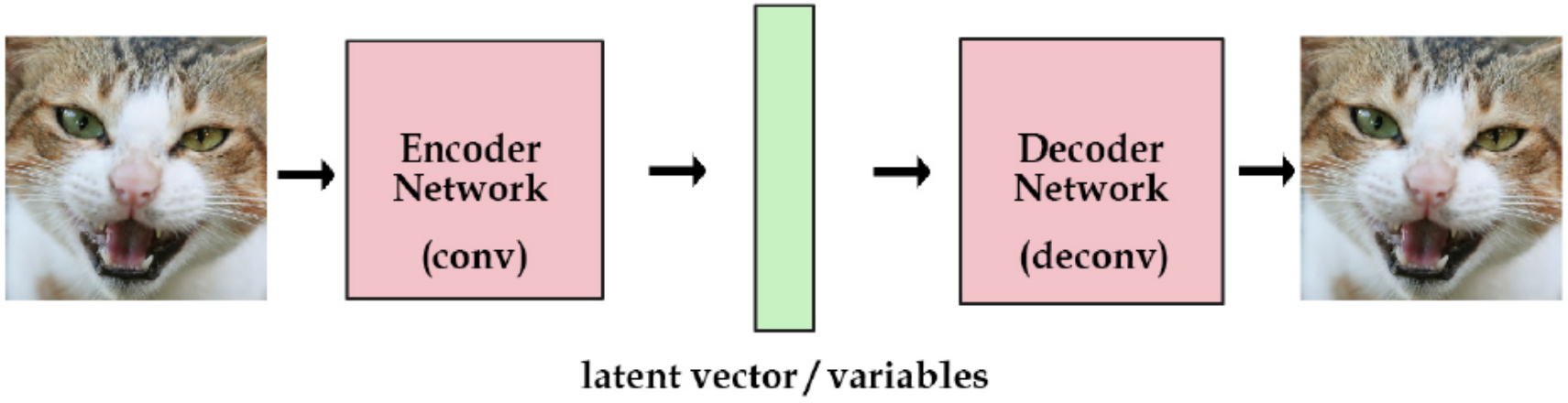


  - Learns a model that can remove the noise.
    - Denoising, filling in parts of the image, etc.

http://inspirehep.net/record/1252540/plots

Autoencoders as a Generative Model

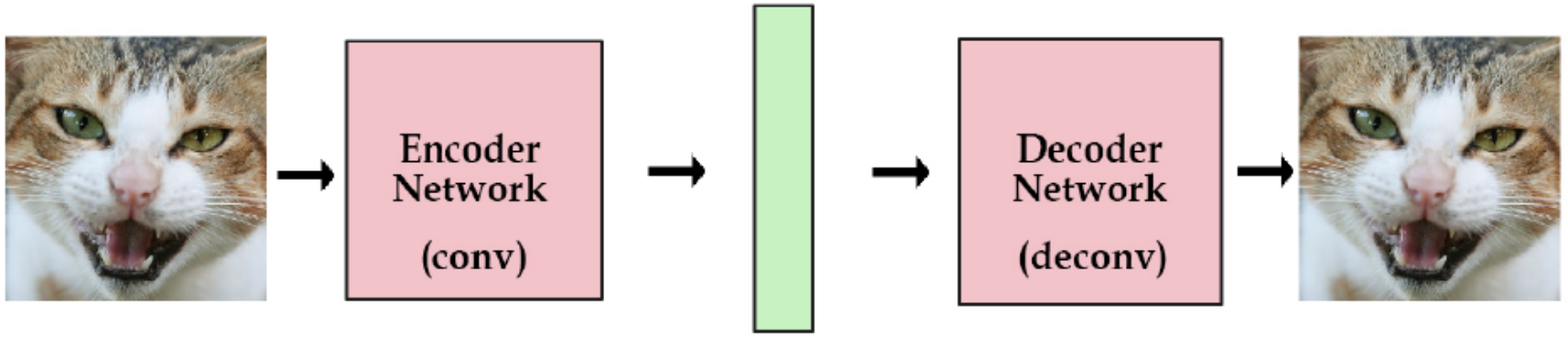
- Good autoencoder would encode any image to latent space ‘z’.
  - Encoder converts image to a continuous space.



  - Decoder converts from any continuous ‘z’ to images.
- We can view the decoder as a generative model:
  - If we sample a ‘z’, decoder should turn this into a realistic sample.

http://kvfrans.com/variational-autoencoders-explained/

Problem with Basic Encoders as Generative Models



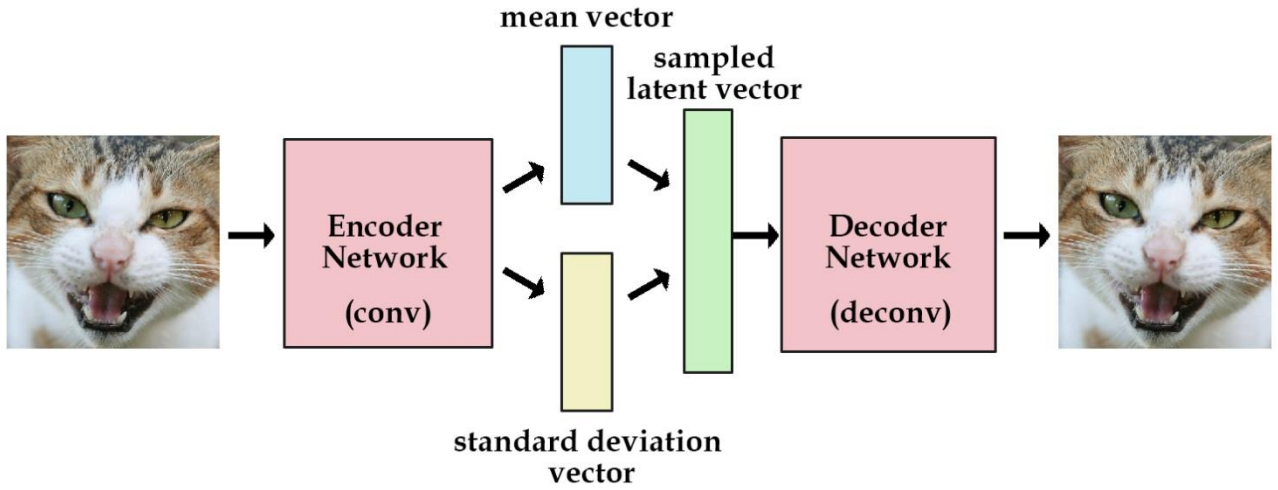
- Unfortunately, there is a problem with training this model.
  - It could "overfit" by mapping each image to a different point in 'z' space.

- Variational autoencoders:
  - Consider marginal likelihood over probabilistic decoding.
  - Add 'z' distribution regularizer, usually encouraging closeness to Gaussian.

http://kvfrans.com/variational-autoencoders-explained/

Variational Autoencoder (VAE)

- Variational autoencoders (VAEs) have the same structure:
  - Encoder network q(z | x), outputting parameters of a distribution.
    - Usually the mean and variance of a Gaussian, so takes ‘x’ and gives a Gaussian.
  - Decoder network p(x | z), same as before (takes a ‘z’ and gives an ‘x’).
  - Prior distribution p(z), usually a N(0,I) distribution.



Training Variational Autoencoders

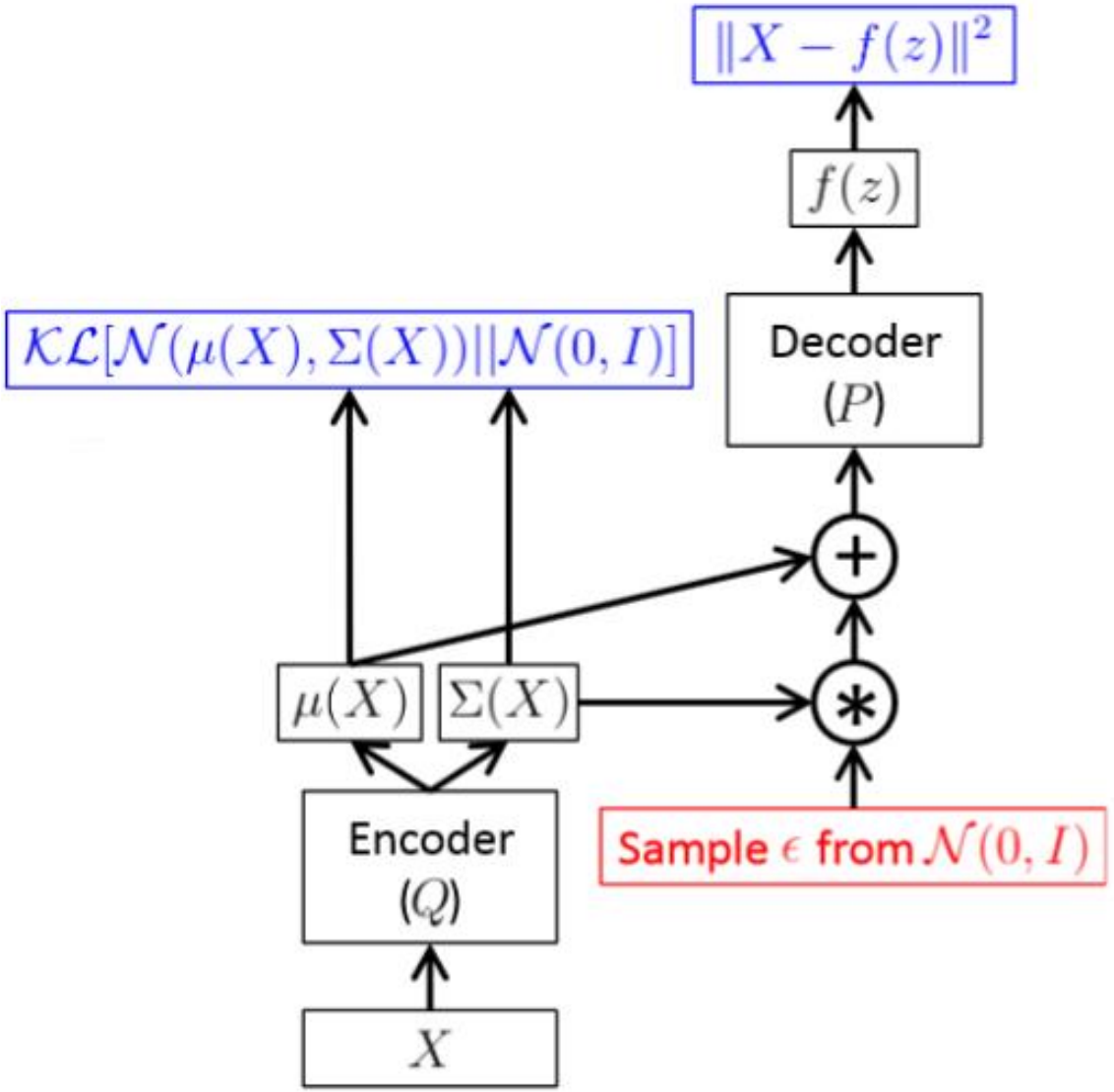
- Training: minimize marginal decoder NLL, regularize by prior:

$$-E_{z\sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)] + KL(q_\theta(z|x_i)||p(z))$$

- Trained using stochastic gradient:
  - "Stochastic" because you choose a training example and sample 'z'.
    - Sampling from encoder network is easy (Gaussian sampling).
      - Can use affine property to get Monte Carlo approximation of gradient ("reparameterization trick").

- Notice again that it's the reverse KL for tractability.
  - Equivalent to variational inference:
    - Using q(z | x) as approximation of posterior p(z | x).

Training Variational Autoencoders



https://arxiv.org/pdf/1606.05908.pdf

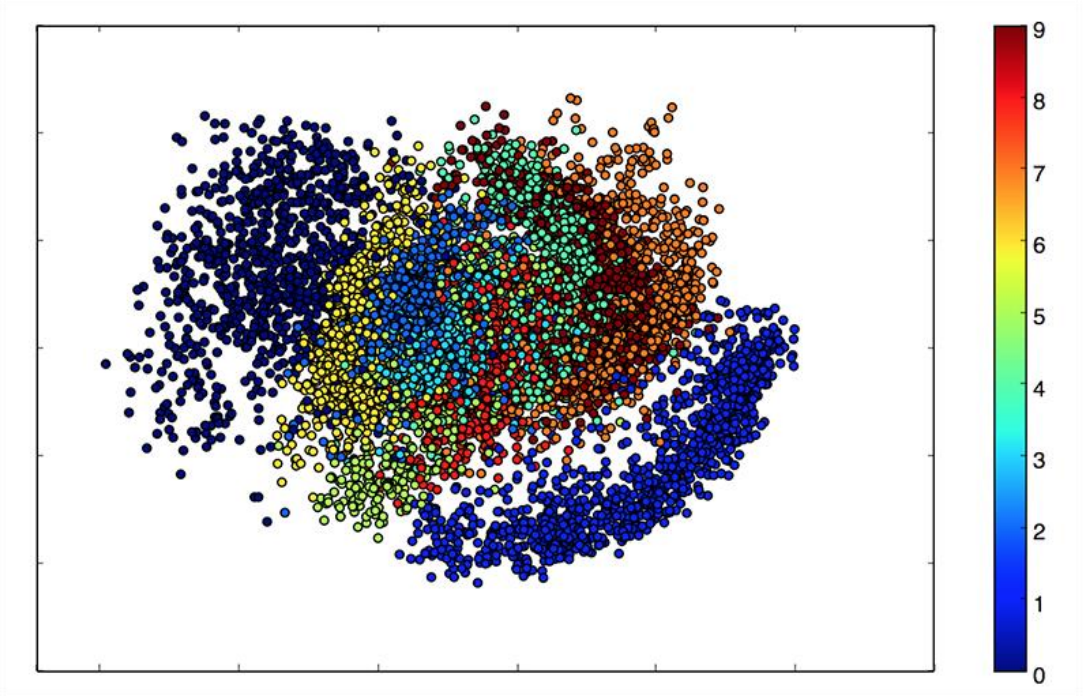
Variational Autoencoder Example: MNIST

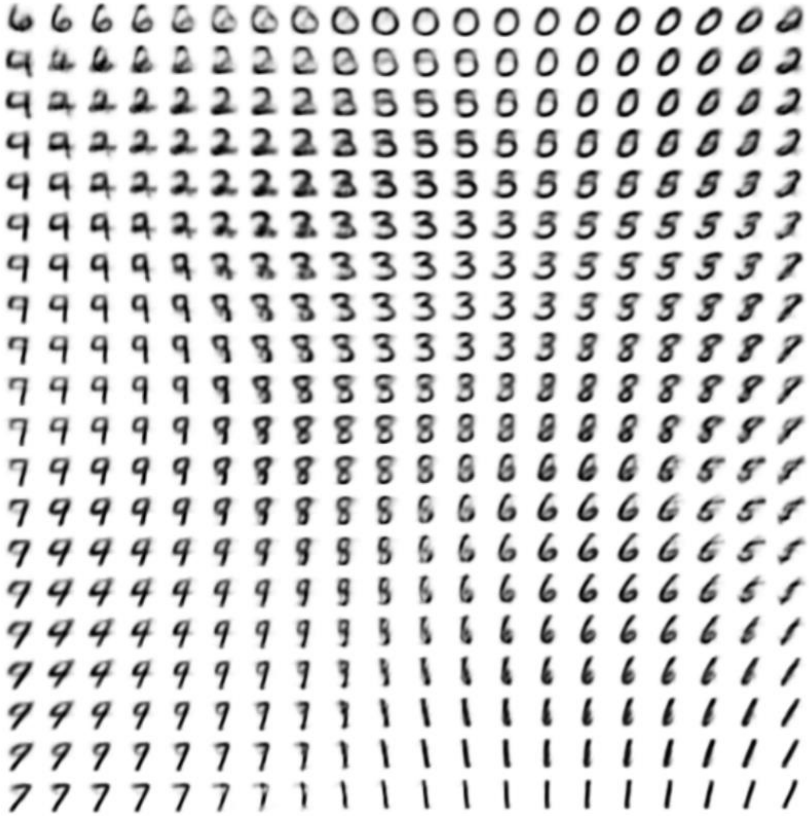
- Samples from model applied to MNIST:

http://kvfrans.com/variational-autoencoders-explained/

Variational Autoencoder Example: MNIST

- Visualizations of latent space:



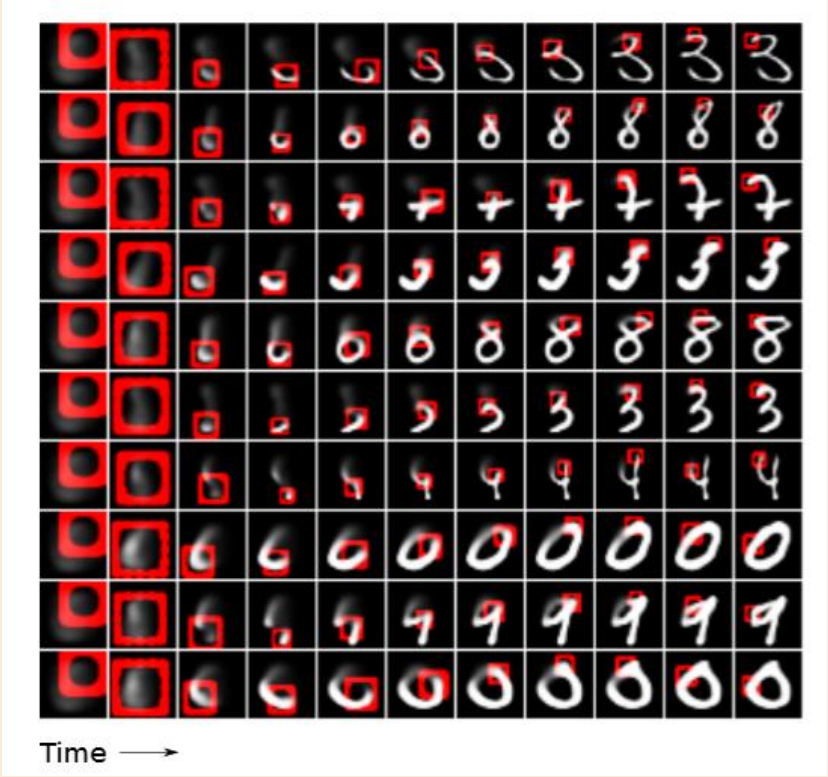


- Non-linear unlike PCA, but visualization is not as nice as t-SNE.
- However, goal was to produce a generative model:
  - Moving through latent space generates realistic digits (video).

https://blog.keras.io/building-autoencoders-in-keras.html
https://arxiv.org/pdf/1312.6114.pdf

DRAW: VAE+RNN+Attention

- Put VAE inside RNN, add attention (next lecture) to “draw” images:
  - https://www.youtube.com/watch?v=Zt-7MI9eKEo



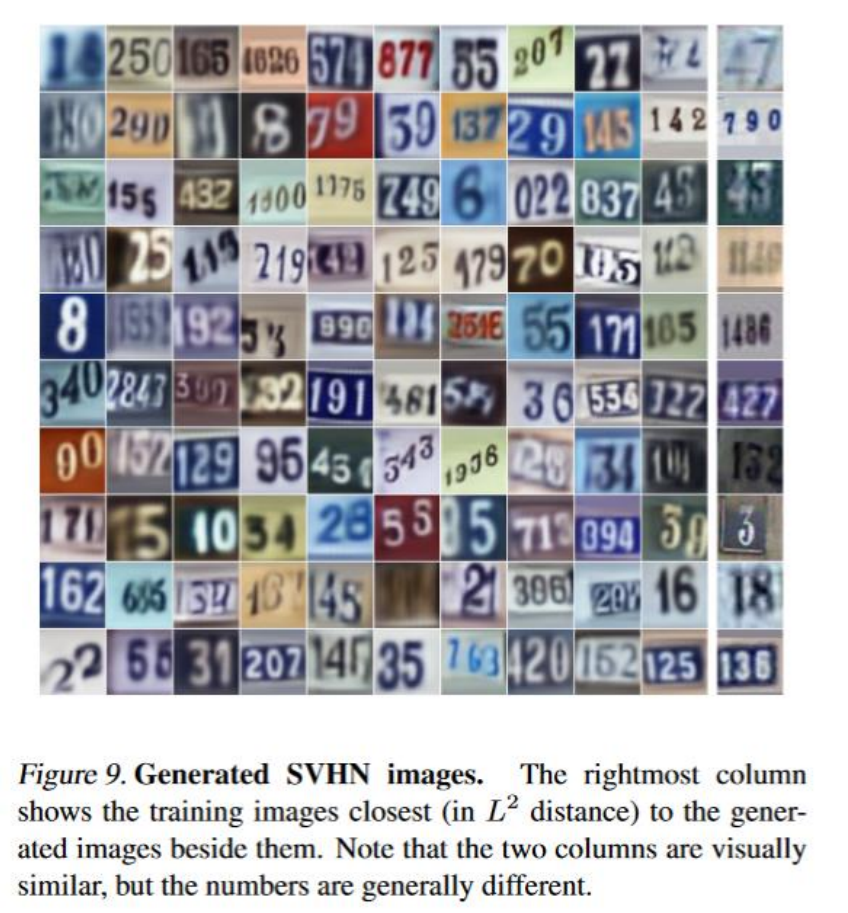


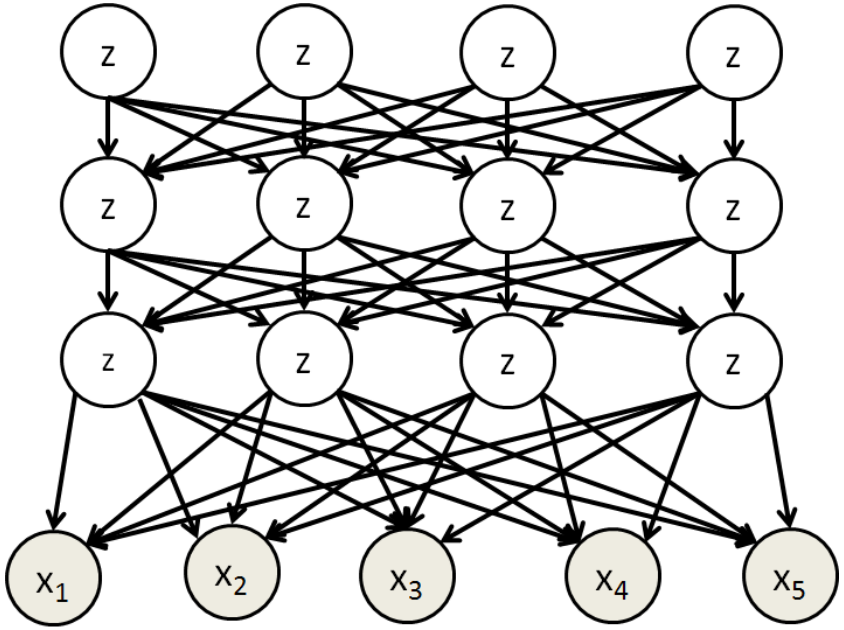
Figure 9. **Generated SVHN images.** The rightmost column shows the training images closest (in L^2 distance) to the generated images beside them. Note that the two columns are visually similar, but the numbers are generally different.

https://arxiv.org/pdf/1502.04623.pdf

(pause)

Neural Network Generative Model

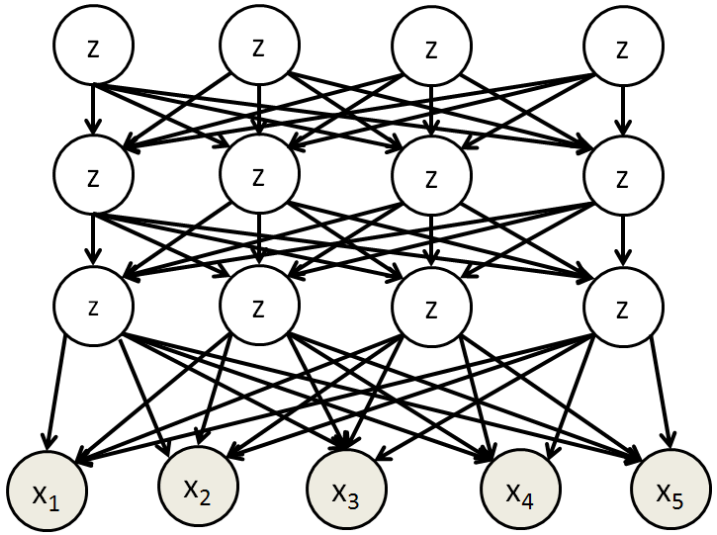
- Recall the structure of a deep belief network and decoder network:



- Notice that the edges are backwards compared to neural networks.
  - We "generate" the features based on the latent 'z' variables.
- Inference is a nightmare: observing 'x' makes everything dependent.

Neural Network Generative Model

- Inference is easier if we make everything deterministic.



  - But we need randomization since otherwise you generate same ‘x’.
- Usual assumption: top layer comes from multivariate Gaussian:
  - So you sample a Gaussian, and neural network tries to convert to image.

Generative Adversarial Network (GAN)

- So ancestral sampling is really easy:
  - Sample from a Gaussian, pass the sample through the network.
- But inference is still hard under the "convert Gaussian to sample".
  - We can't compute the likelihood needed for training.
  - In VAEs we used a variational approximation.
- Seemingly unrelated: we've become really good at image classification.
- Key ideas of generative adversarial networks (GANs):
  - Use ancestral sampling in this "generator" network.
  - Use a second "discriminator" network to decide if samples look real.
- Discriminator "teaches" generator to make real-looking samples.

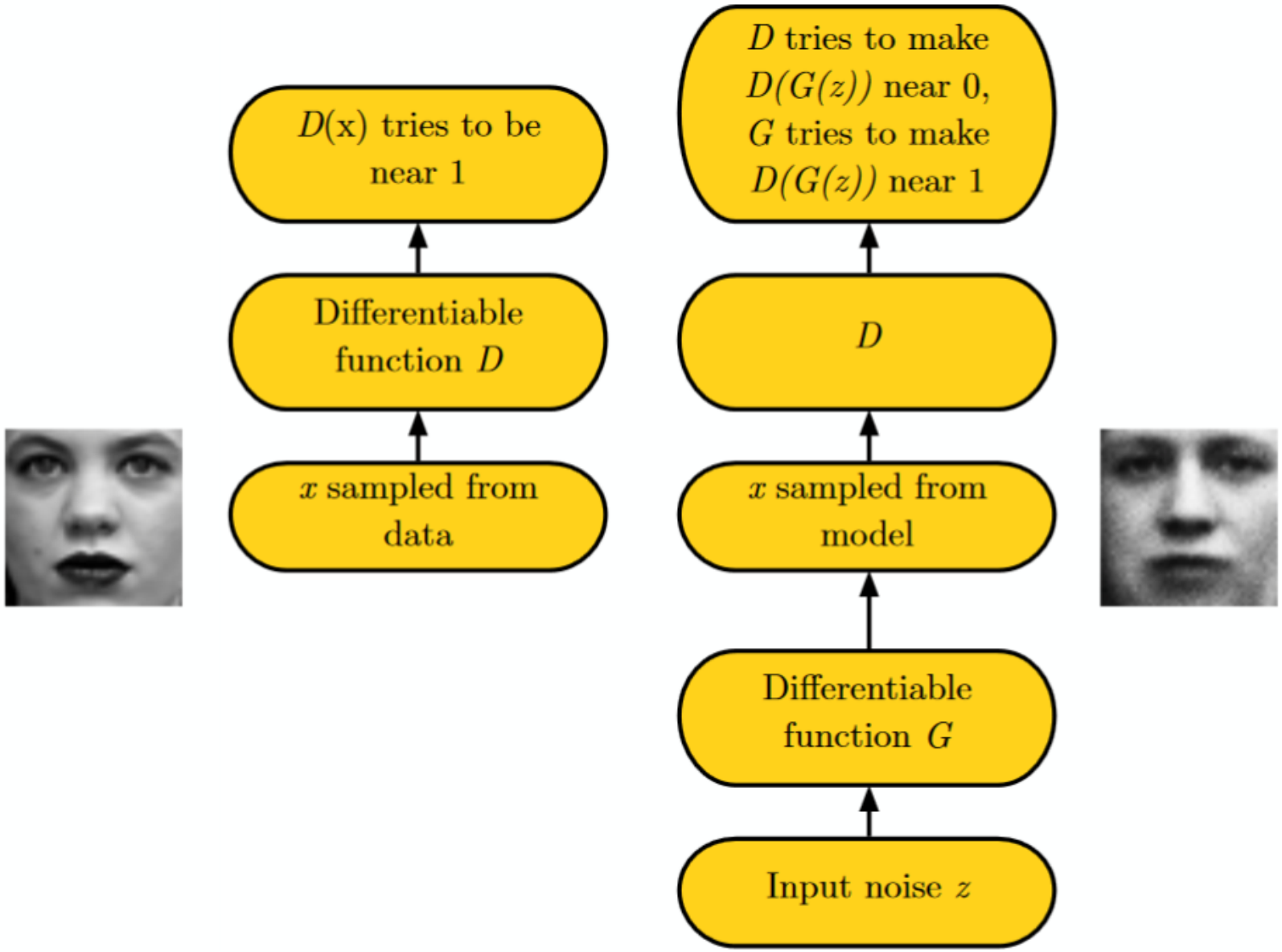
Generative Adversarial Networks

- The generator and discriminator networks compete:
  - Discriminator network trains to classify real vs. generated images.
    - Tries to maximize probability of real images, minimize probability of sampled images.
    - A standard supervised learning problem.
  - Generator network adjusts parameters so samples fool the discriminator.
    - It never sees real data.
    - Trains using the gradient of the discriminator network.
      - Backpropagated through the network so samples look more like real images.
- Can be written as a saddle-point problem:

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

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

Generative Adversarial Network (GAN)

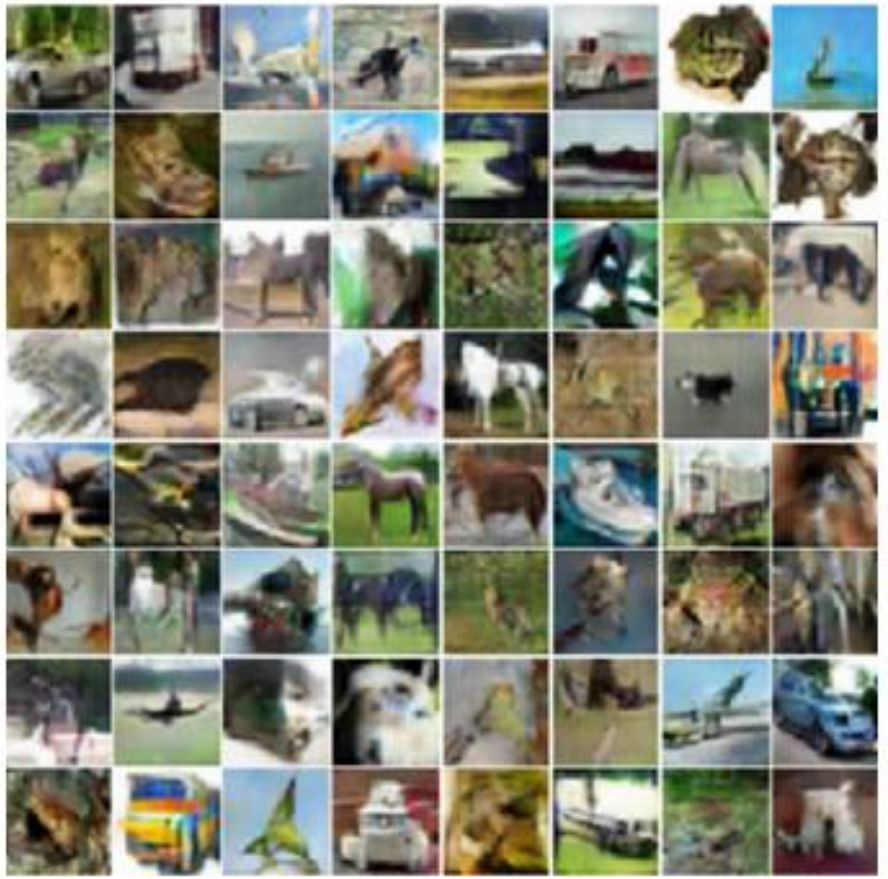


https://arxiv.org/pdf/1701.00160.pdf

Beyond Initial GAN Model

- Improving GANs is an active research area…

Real images (CIFAR-10)

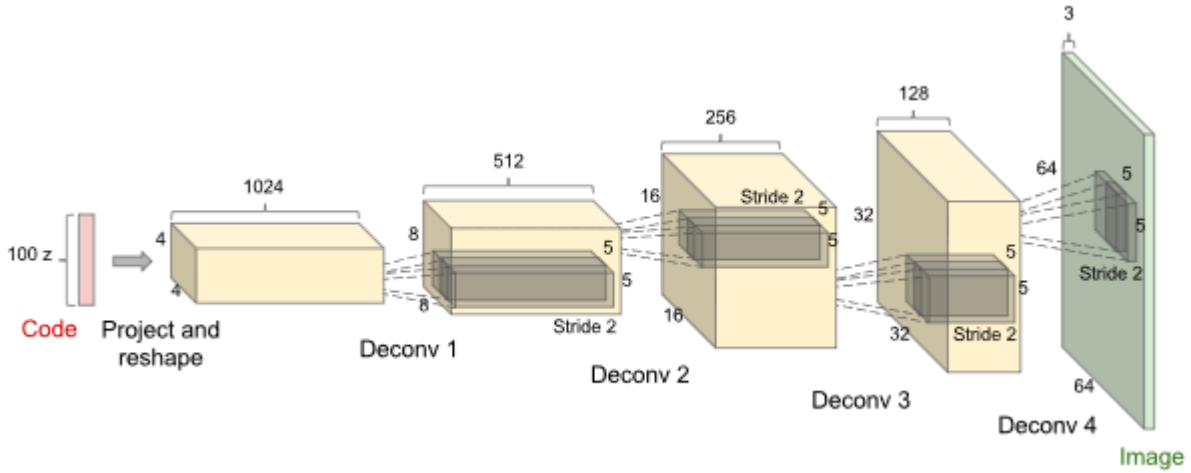


Generated images

https://blog.openai.com/generative-models

Beyond Initial GAN Model

- Generating album covers with convolutional GANs:
  - Used uniform rather than Gaussian.



https://blog.openai.com/generative-models/
https://github.com/Newmu/dcgan_code

GANs for Other Problems

- GANs for super-resolution:

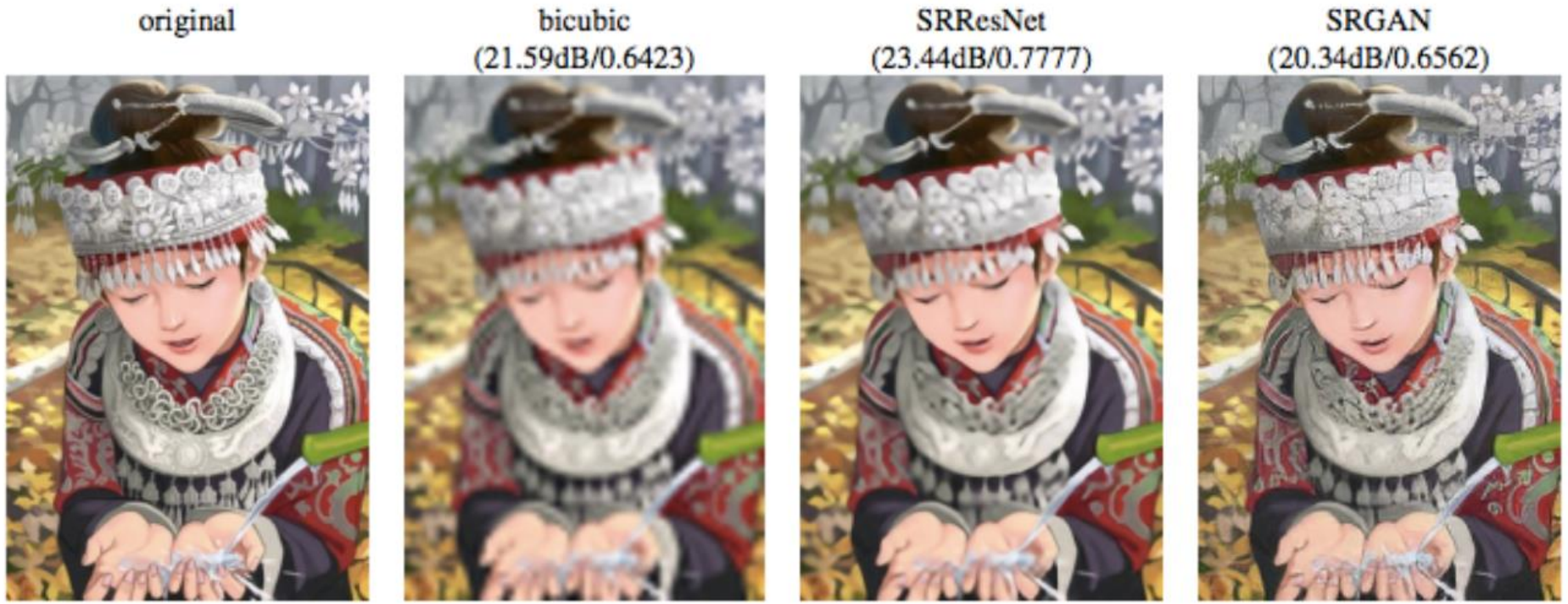


Figure 4: Ledig *et al.* (2016) demonstrate excellent single-image superresolution results that show the benefit of using a generative model trained to generate realistic samples from a multimodal distribution. The leftmost image is an original high-resolution

https://arxiv.org/pdf/1701.00160.pdf

GANs for Other Problems

- GANs for text-to-image translation:

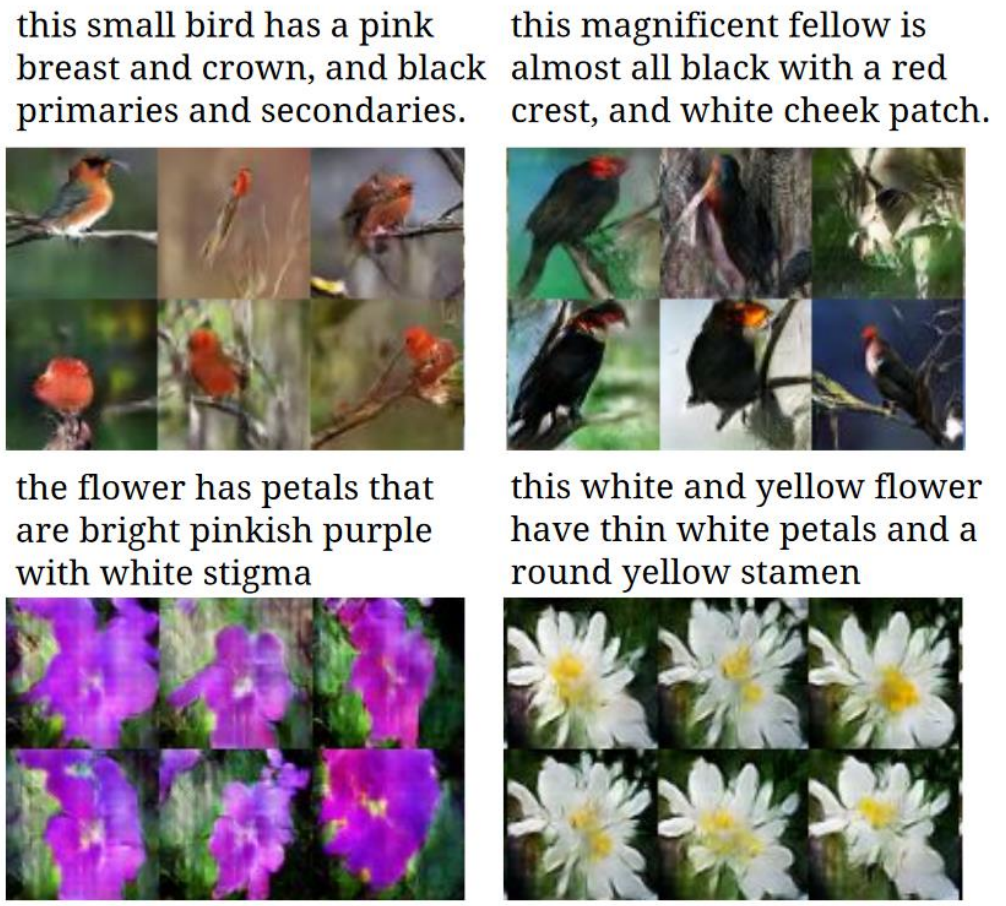


Figure 23: Text-to-image synthesis with GANs. Image reproduced from Reed *et al.* (2016b).

GANs for Other Problems

- GANs for text-to-image translation:

Figure 25: StackGANs are able to achieve higher output diversity than other GAN-based text-to-image models. Image reproduced from Zhang *et al.* (2016).

GANs for Other Problems

- GANs for image manipulation:
  - https://www.youtube.com/watch?v=9c4z6YsBGQ0
  - https://www.youtube.com/watch?v=FDELBFSeqQs

GANs for Other Problems

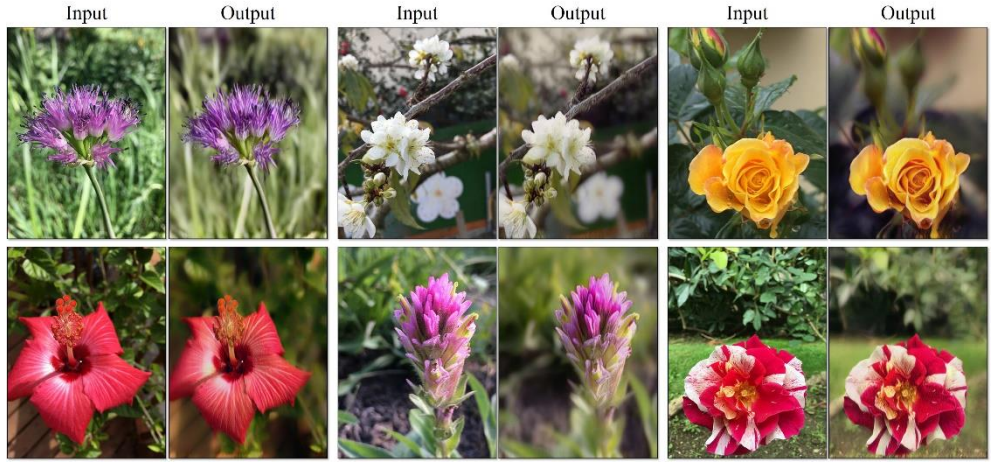
- GANs for image-to-image translation:
  - https://affinelayer.com/pixsrv

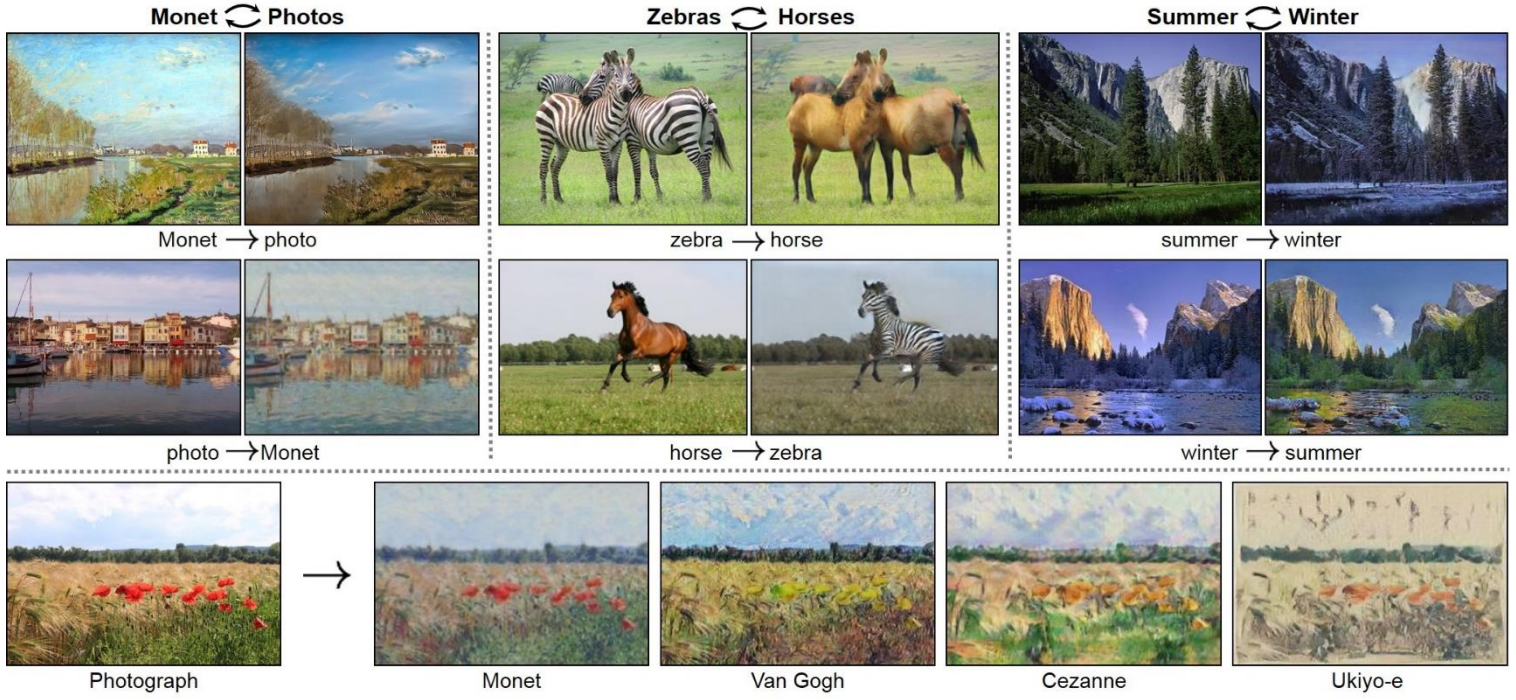
Figure 7: Isola *et al.* (2016) created a concept they called image to image translation, encompassing many kinds of transformations of an image: converting a satellite photo into a map, coverting a sketch into a photorealistic image, etc. Because many of these conversion processes have multiple correct outputs for each input, it is necessary to use generative modeling to train the model correctly. In particular, Isola *et al.* (2016) use a GAN. Image to image translation provides many examples of how a creative algorithm designer can find several unanticipated uses for generative models. In the future, presumably many more such creative uses will be found.

https://arxiv.org/pdf/1701.00160.pdf

GANs for Other Problems

- CycleGANs: recent works try to avoid needing to have image pairs:
  - Adds extra part regularizing mapping in both directions.





https://github.com/junyanz/CycleGAN

In Progress…

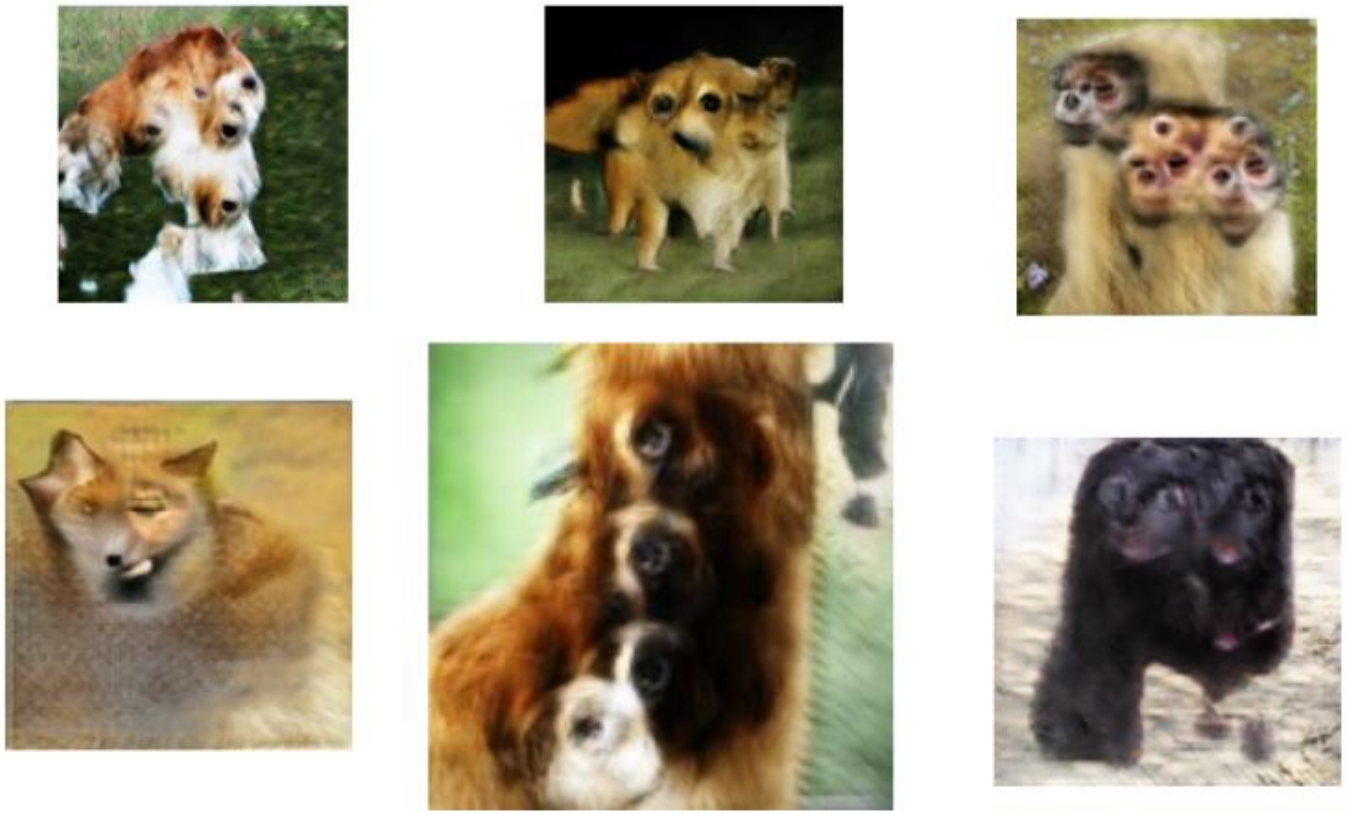


Figure 29: GANs on 128 × 128 ImageNet seem to have trouble with counting, often generating animals with the wrong number of body parts.

Figure 30: GANs on 128 × 128 ImageNet seem to have trouble with the idea of three-dimensional perspective, often generating images of objects that are too flat or highly axis-aligned. As a test of the reader's discriminator network, one of these images is actually real.

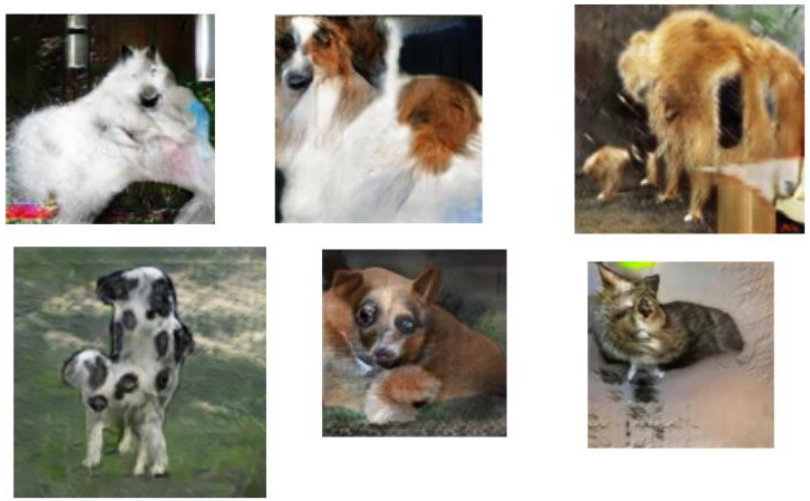


Figure 31: GANs on 128 × 128 ImageNet seem to have trouble coordinating global structure, for example, drawing "Fallout Cow," an animal that has both quadrupedal and bipedal structure.

May not work as well in real life as in papers:
https://twitter.com/search?q=edges2cats

https://arxiv.org/pdf/1701.00160.pdf

Improving Resolution

- New generative models are appearing at a very-fast rate:

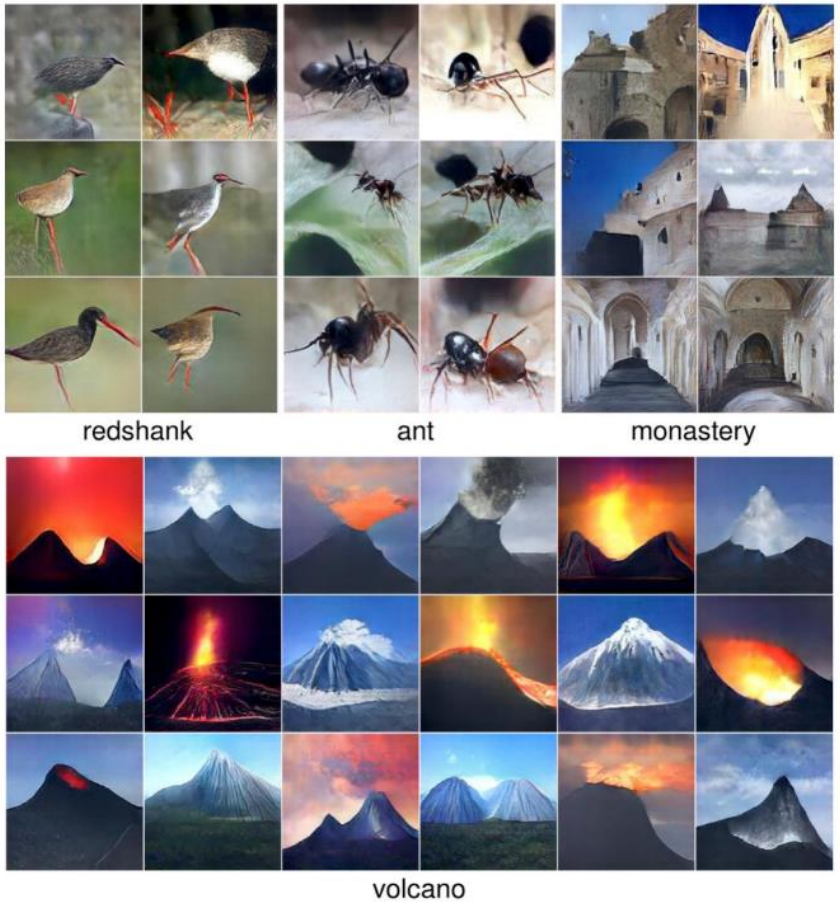


Figure 33: PPGNs are able to generate diverse, high resolution images from ImageNet classes. Image reproduced from Nguyen *et al.* (2016).

https://arxiv.org/pdf/1701.00160.pdf

Improving Resolution

- A lot of work on trying to improve resolution:
  - Fake celebrities: https://www.youtube.com/watch?v=VrgYtFhVGmg

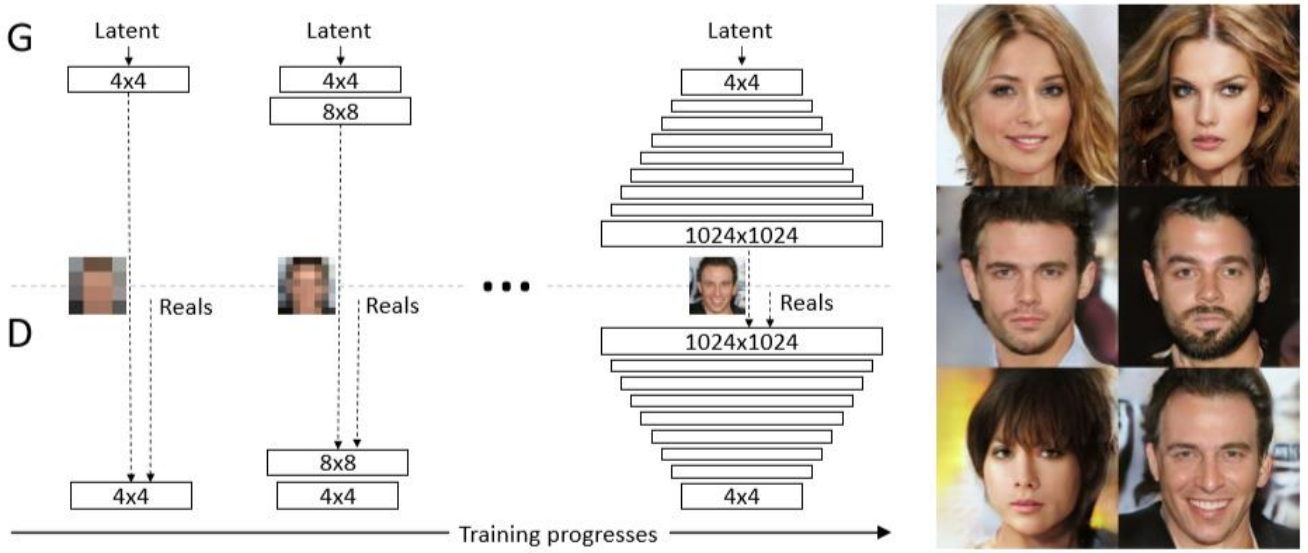


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $N \times N$ refers to convolutional layers operating on $N \times N$ spatial resolution. This allows stable synthesis in high resolutions and also speeds up training considerably. One the right we show six example images generated using progressive growing at 1024×1024.

http://research.nvidia.com/sites/default/files/pubs/2017-10_Progressive-Growing-of/karras2018iclr-paper.pdf

Summary

- Autoencoders:
  - Latent-factor model, where we training output of network to reproduce input.
- Deep generative models:
  - Sample from a simple distribution,
    use neural network to transform to real sample.
  - Likelihood is intractable.
- Variational autoencoders (VAEs):
  - Autoencoder with the simple distribution in the model.
  - Use variational approximation to deal with intractable marginal likelihood.
- Generative adversarial networks (GANs):
  - Replace likelihood with a “discriminator” that trains to detect fake samples.