Variational Autoencoders VS Generative Adversarial Networks

Alexander Mennborg
Ismat Halabi
Elliot Härenby Deak

Table of Contents

Introduction

- Generative models, complex data, hard train
- Delimitation: this only considers image generation
- Aims to compare performance of both models
- Related Work
    - Goodfellow I. et al. Generative Adversarial Nets
    - Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes

Taxonomy of Generative Models

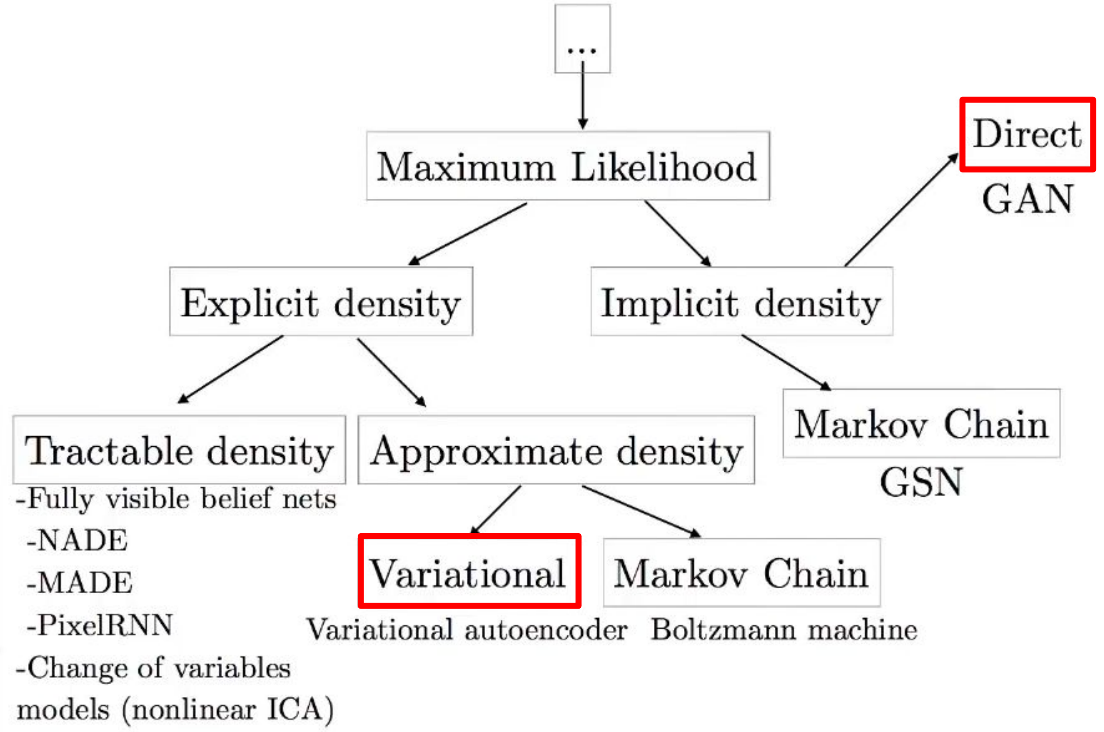


Image taken from a talk by Ian Goodfellow found here https://www.youtube.com/watch?v=HGYYEUSm-0Q

Generative Adversarial Networks

What are GANs?

- Invented by Ian Goodfellow in 2014
- Idea: evaluate real/ fake images
- Training adversary
- Vanilla GANs - MLP

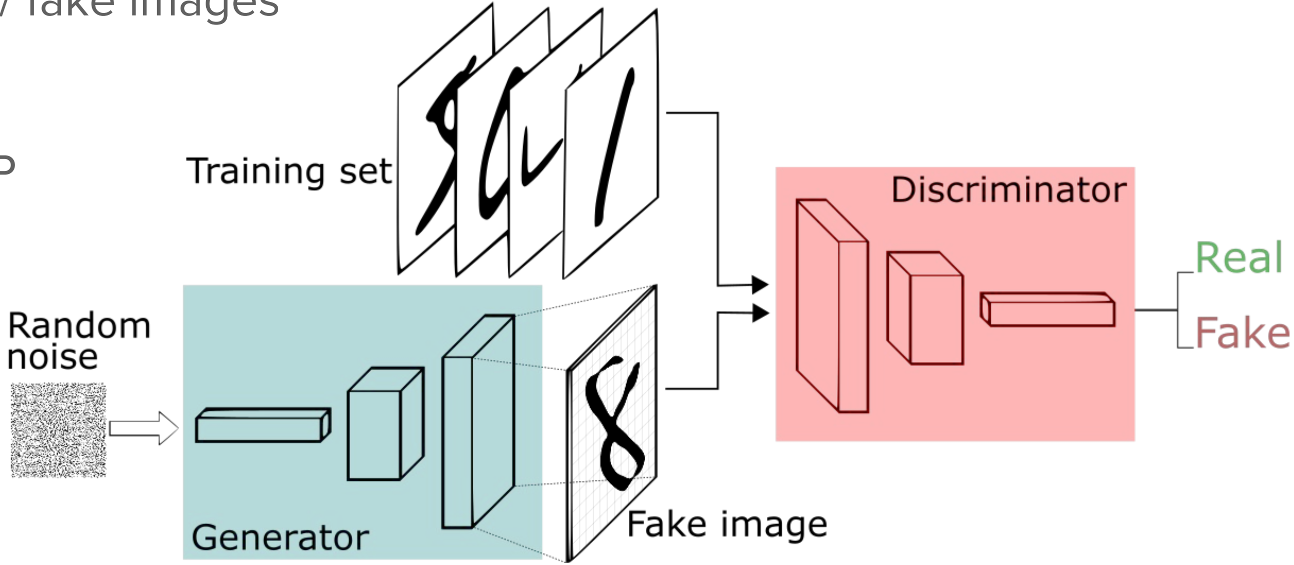
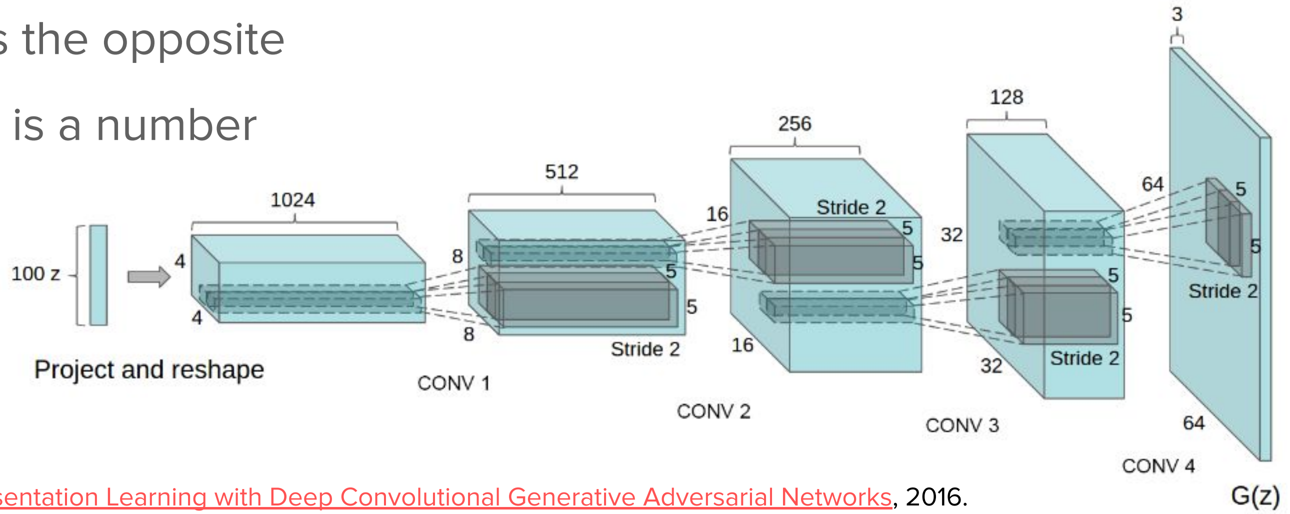


Image credit: Thalles Silva

Deep Convolutional GANs (DCGANs)

- One specific GAN architecture among many
- Generator model inputs latent vector and outputs image
- Conv. layers uses transposed convolutions
- Discriminator is the opposite

  except outputs is a number



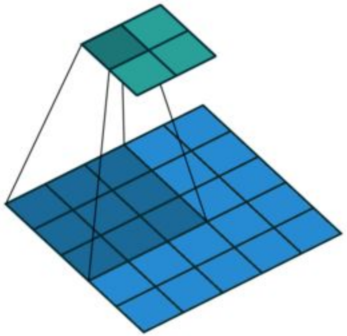
Radford et al. [Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks](), 2016.

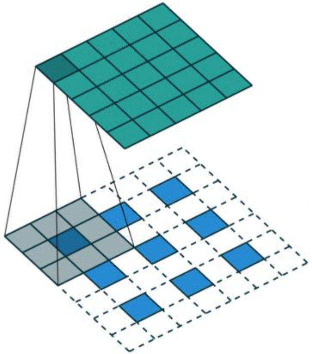
Deep Convolutional GANs (DCGANs) - Tips

- Training can be quite unstable
- Use batch normalization,
- Use strided convolutions instead of pooling layer
- Use ReLU (generator) and LeakyReLU (discriminator)

Downscaling
(strided convolution)

2x2 output
⇑
5x5 input





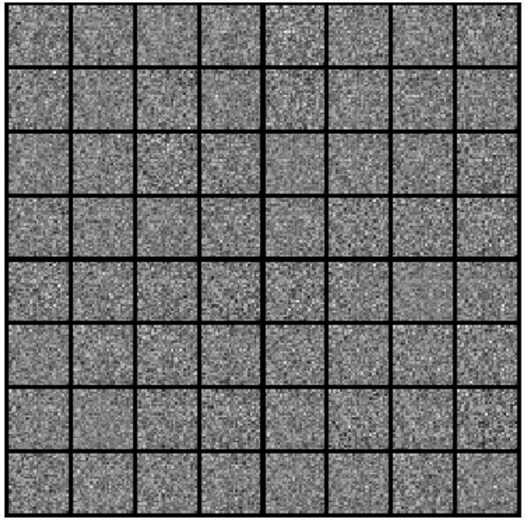
Upscaling
(fractional strided convolution)
5x5 output
⇑
3x3 input (+ padding)

Gifs taken from: https://github.com/vdumoulin/conv_arithmetic

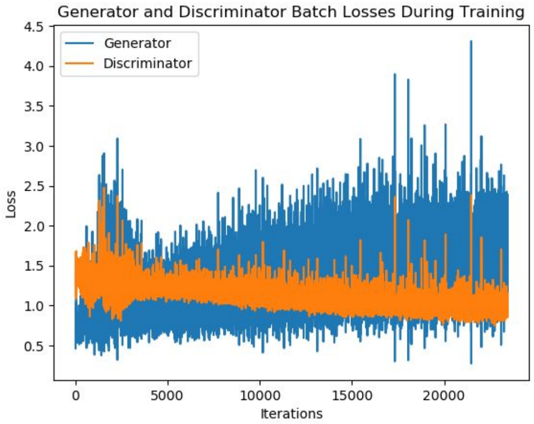
Training Results - MNIST

Real Images

Fake Images
Training Progression



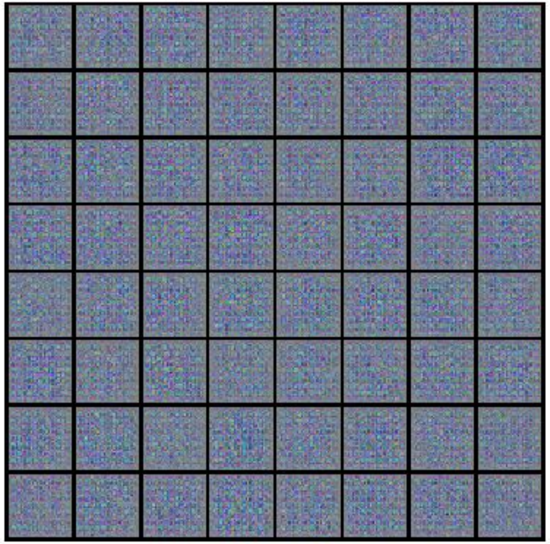
- Uses vanilla GANs
- Learns overall structure quickly
- Some minor denoising happening



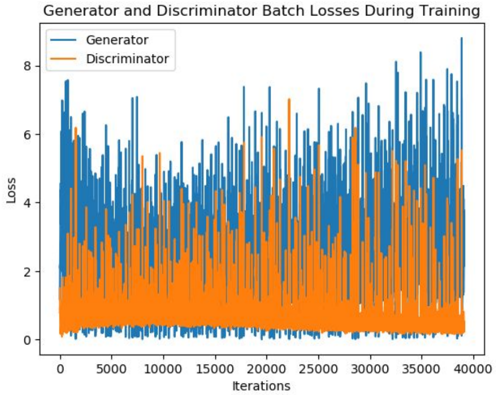
Training Results - CIFAR10

Real Images

Fake Images
Training Progression



- Uses DCGANs
- Fake images, blurry, CIFAR10 like
- Mostly unrecognizable results



Variational Auto-Encoders (VAE)
Bayesian

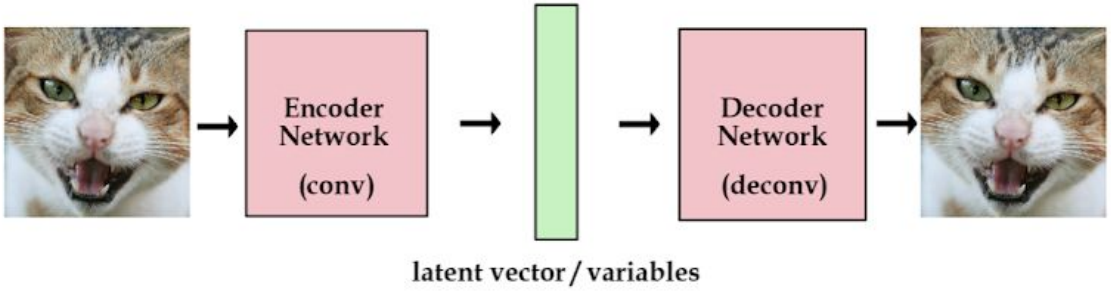
General Idea

- Type of **unsupervised learning**

- UL:
  - No labels, just data
  - Goal: learn some underlying hidden structure of data

- Given observed data points X_n {n=1 to N} distributed according to some (unknown) ground truth distribution p_gt(x), learn a model p that we can sample from, such that p is as similar to p_gt(x) as possible.

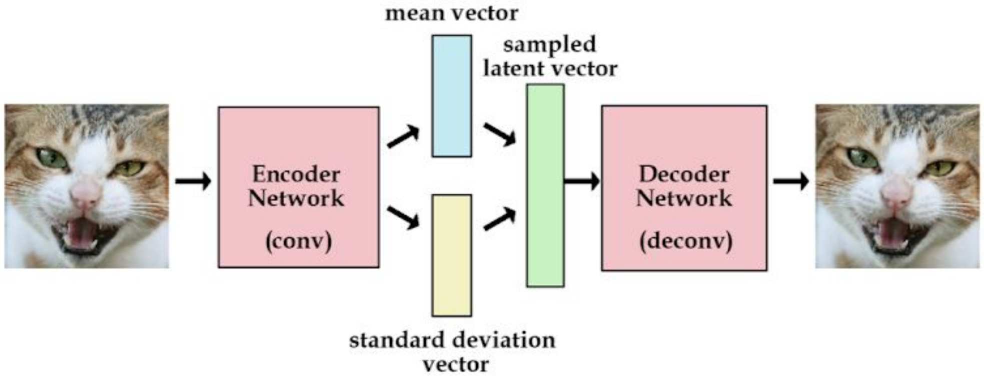
Standard AutoEncoder (AE)



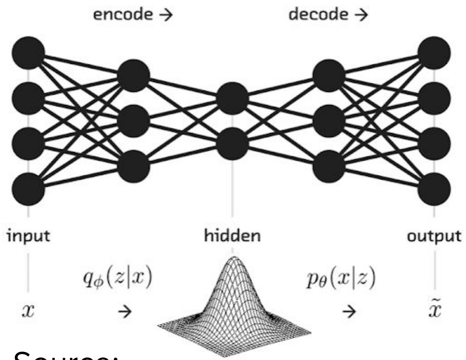
Source: http://kvfrans.com/variational-autoencoders-explained/

- Save encoded image
- Recover it later
- Middle part called bottleneck: dimensionality reduced
- Deterministic encoding/decoding, no new images: output ~ input!

VAE: Constrained Stochastic AE => Generative!



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



Source: https://towardsdatascience.com/what-the-heck-are-vae-gans-17b86023588a

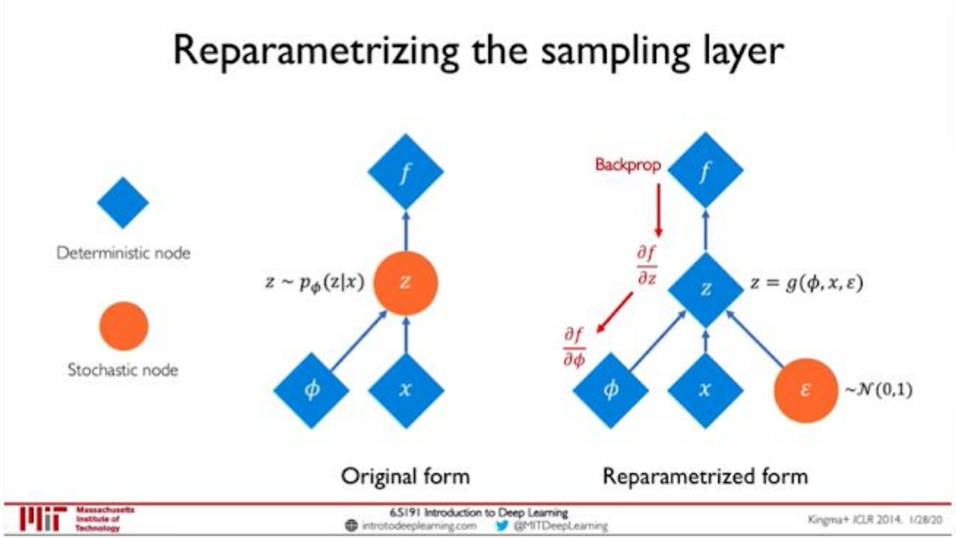
- Encoder constraints: generate latent vectors that roughly follows unit gaussian distribution
- Sample latent vector from unit gaussian
- Parameterization trick: encoder generates vector of means and vector and standard deviations instead of vector of real numbers to optimize KL divergence
- Decoder generates new sample from latent space!

Losses

- generation loss = mean(square(generated_image – real_image))
  - Accuracy in reconstruction images
- latent_loss = KL_Divergence(latent_variable, unit_gaussian)
  - How close z matches unit Gaussian
- total_loss = generation_loss + latent_loss

Generation loss: same as in standard AE

Reparameterization Trick



Source: https://www.youtube.com/watch?v=rZufA635dq4

z = $\mu + \sigma.\epsilon$, only μ and σ trained, ϵ purely stochastic, all Gaussian unit

Allows for end-to-end training

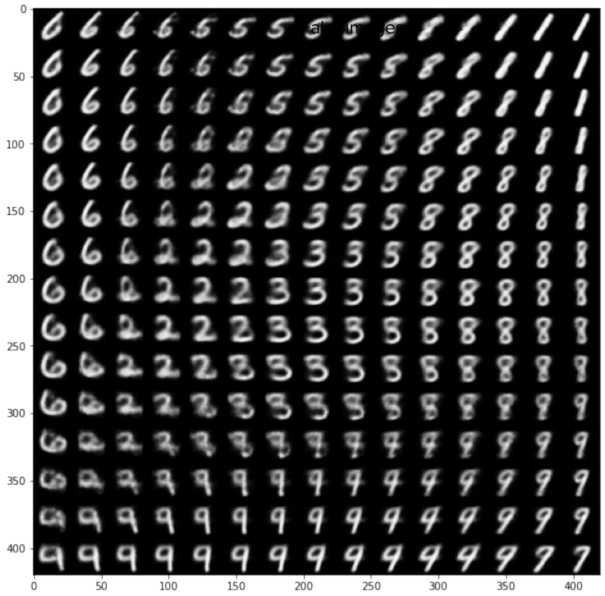
ELBO: Evidence Lower Bound

- Originally, $p(x) = \int p(z).p(x|z)dz$ intractable (decoder)
- Similarly, $p(z|x) = p(x|z)p(z) / p(x)$ also intractable (Bayes' theorem, encoder)

- ELBO(x) = Expected(z|x) [log p(x,z) – log q(z|x)]
- q(z|x): stochastic encoder, p(z|x): true posterior
- Objective:
  - maximize ELBO (data likelihood, tractable)
  - Minimize KL divergence (distance) of stochastic encoder from true posterior

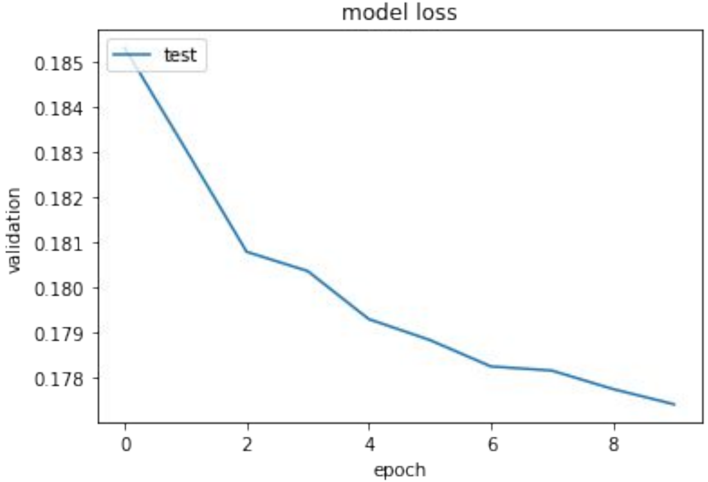
Training Results

Generated images

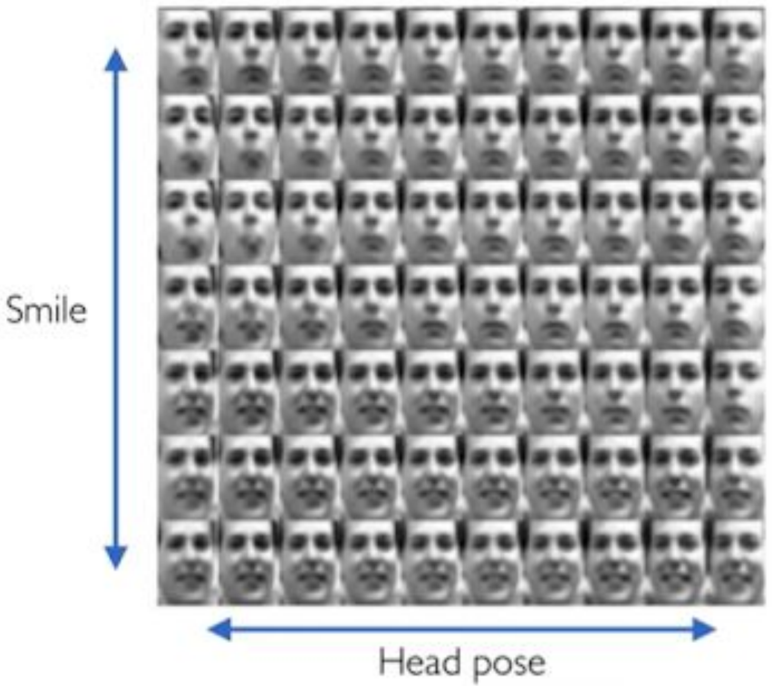


Real images

- 2D variations of z decoded
- CNN / transposed CNN used



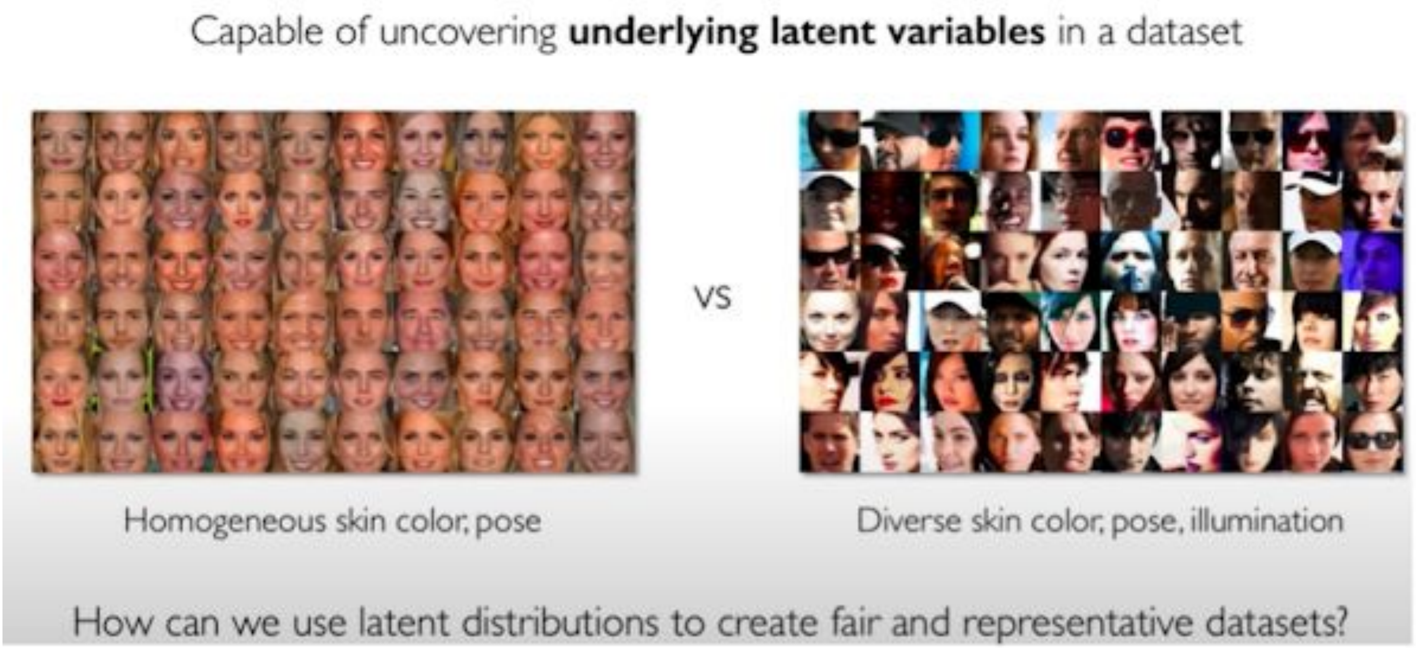
Exploring Latent Space



- Slowly increase/decrease one latent variable leaving others unchanged along each dimension => “continuous" output
- Each dimension encodes different latent feature
- Wanted: uncorrelated latent variables

Source: https://www.youtube.com/watch?v=rZufA635dq4

Application: Debiasing, RL, SL



Source: https://www.youtube.com/watch?v=rZufA635dq4

Experience Gained with VAE

- Used ready code in github
- Difficult to adapt:
    - Compact code but minor changes may break the functioning of the model
    - Understand the intuition but not the maths

Comparison and Differences

Main differences between generators

GAN

- Based on two networks (discriminative/generative)
- Discriminator is not useful after training
- Learns to generate directly from gaussian data

VAE

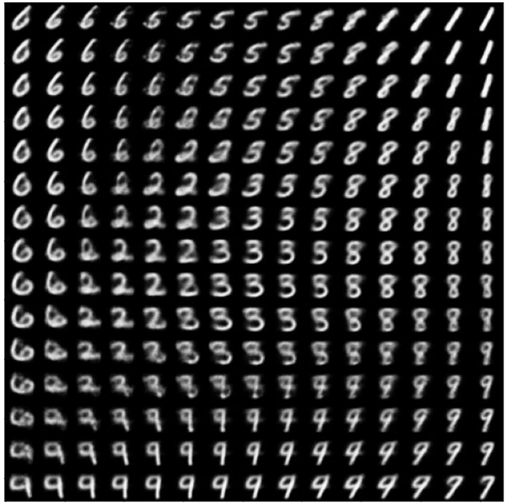
- Based on one network
- Encoder not used after training, but could be
- Learns to compress data to gaussian distribution and decompress it back to image, gaussian noise yields new data

Comparison by results

Real images

GAN fake images

VAE fake images



Comparison of generators - “pros” and “cons”

GAN

- Convincing data
- Sharp

- Somewhat noisy
- Difficult to navigate in latent space
- May suffer from Mode Collapse
- Only real/fake basis
- Difficult to evaluate - no one to rule the all

VAE

- Convincing data
- Easy to navigate in latent space

- Fuzzy-looking

Supervised? Unsupervised? Semi/self-supervised?

Most probably unsupervised, according to most litterature

GANs - Unsupervised with supervised loss

VAEs do not have labels - Unsupervised according to MIT and Stanford

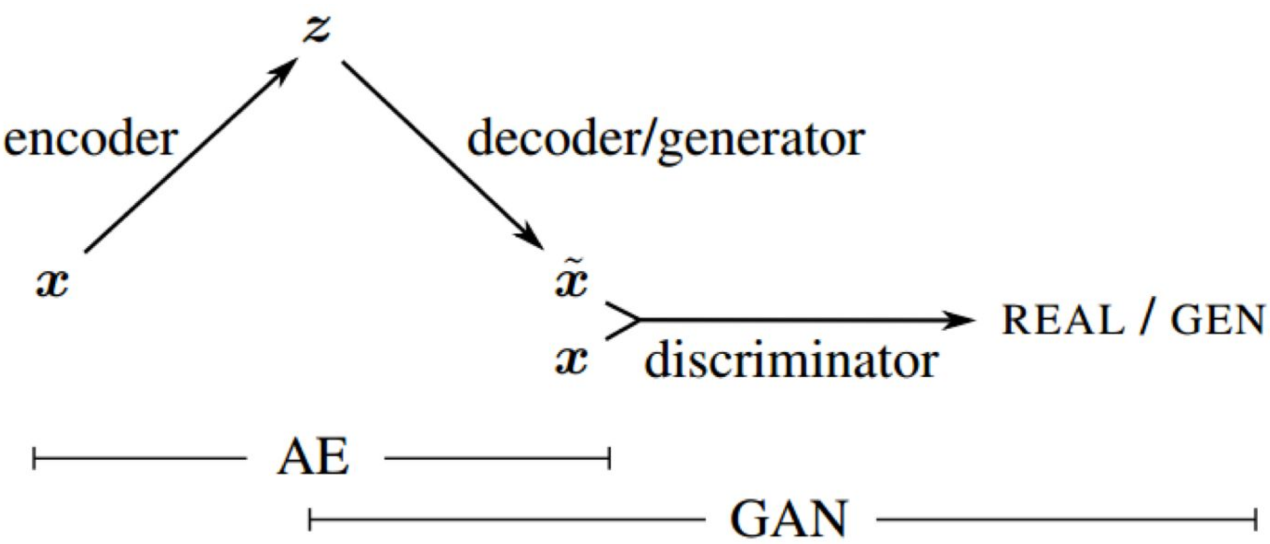
- Input is used as reference to the output

Why not combine them?

Future work

VAE-GAN

- Combination of VAE and GAN
- Basically merge decoder and generator
- Use discriminator during training



From “Autoencoding beyond pixels using a learned similarity metric” A. Larsen

References

Goodfellow I. et al. Generative Adversarial Nets, 2014

Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2016.

Kingma, D. P. and Welling, M. Auto-encoding Variational Bayes. Proceedings of the 2nd International Conference on Learning Representations (ICLR), Banff, Canada, 2014

A. Larsen Autoencoding beyond pixels using a learned similarity metric, 2016