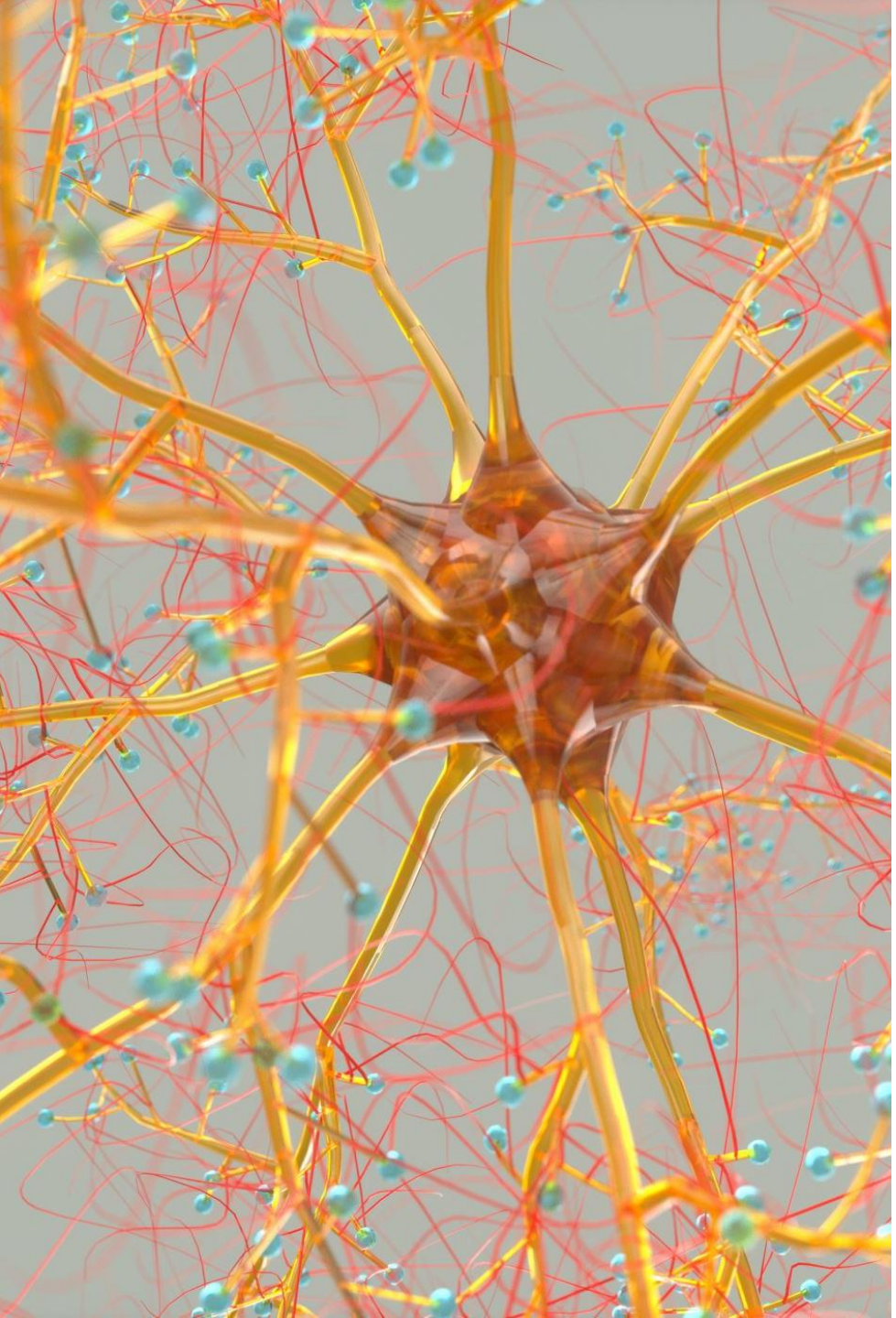


Advanced Topics in Neural Networks

Course 14

GENERATIVE ADVERSARIAL NEURAL NETWORKS

MIHAELA BREABĂN

Agenda

Generative vs. discriminative models

GANs architecture

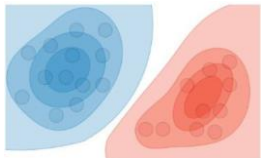
Training GANs, problems and solutions

Conditional and controllable generation

Evaluating GANs

Generative vs. discriminative models in Machine Learning

Generative models:



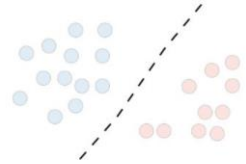
- model the process that generates the observable data (X)

- work by estimating P(X) or P(X/Y)
  - Maximum Likelihood Estimation (EM)
  - Bayes formula for computing P(Y/X)

Examples: Naïve Bayes, Gaussian mixture models, mostly unsupervised methods

They can generate data

Discriminative models:



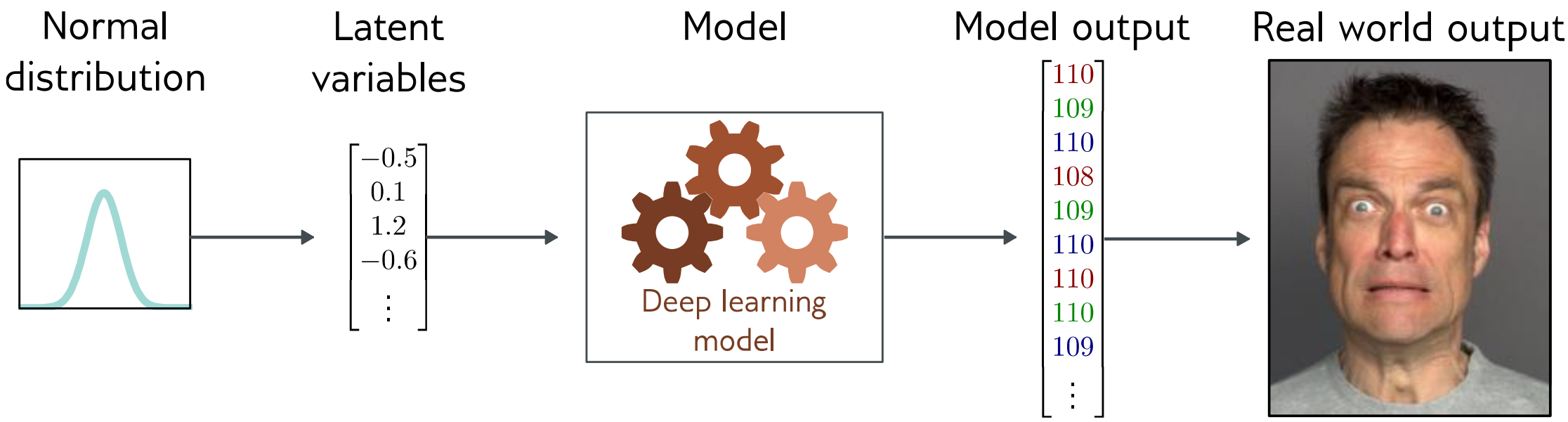
- do not try to understand the data generation process

- generally work by identifying rules that discriminate between classes/clusters or directly estimate P(Y/X)

Most ML algorithms are discriminative
  - Identify class boundaries: decision trees, kNN SVMs, logistic regression, NNs, …
  - Identify cluster representatives: k-Means,…

They can not generate data

Latent variable models



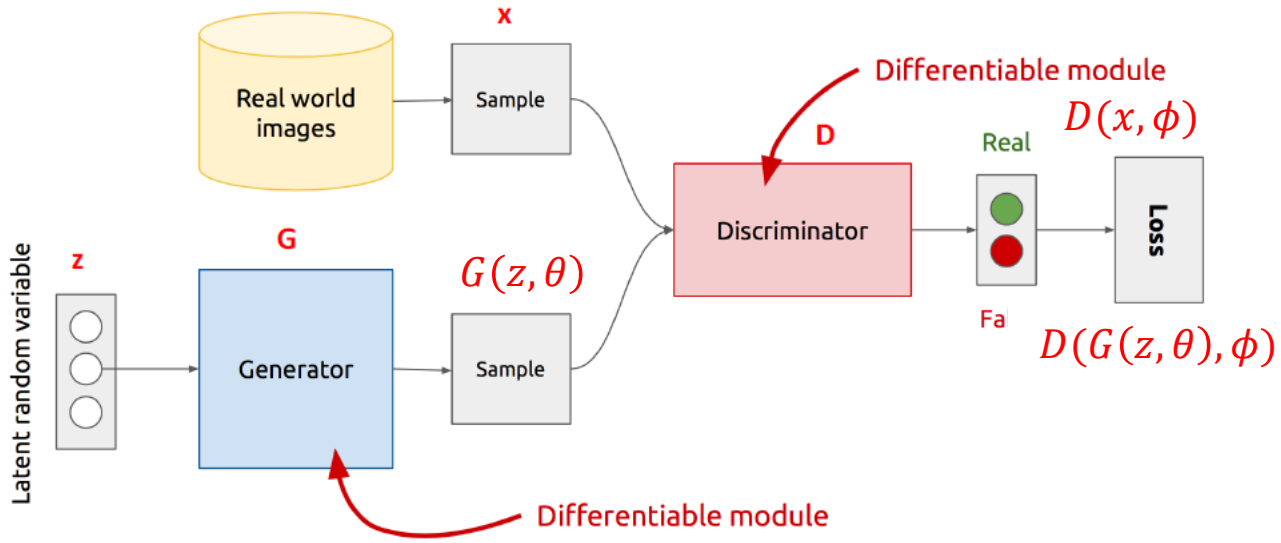
Latent variable models map a random “latent” variable to create a new data sample
In GANs, this is called the generator:

$$x_* = G(z, \theta)$$

Generative adversarial networks

- a generative network
  - Learns to produce realistic data

- a discriminative model
  - Learns to discriminate real data from fake/generated data



- Use backpropagation to train the two networks
- Alternate the training: train the discriminator, than the generator, again the discriminator,…

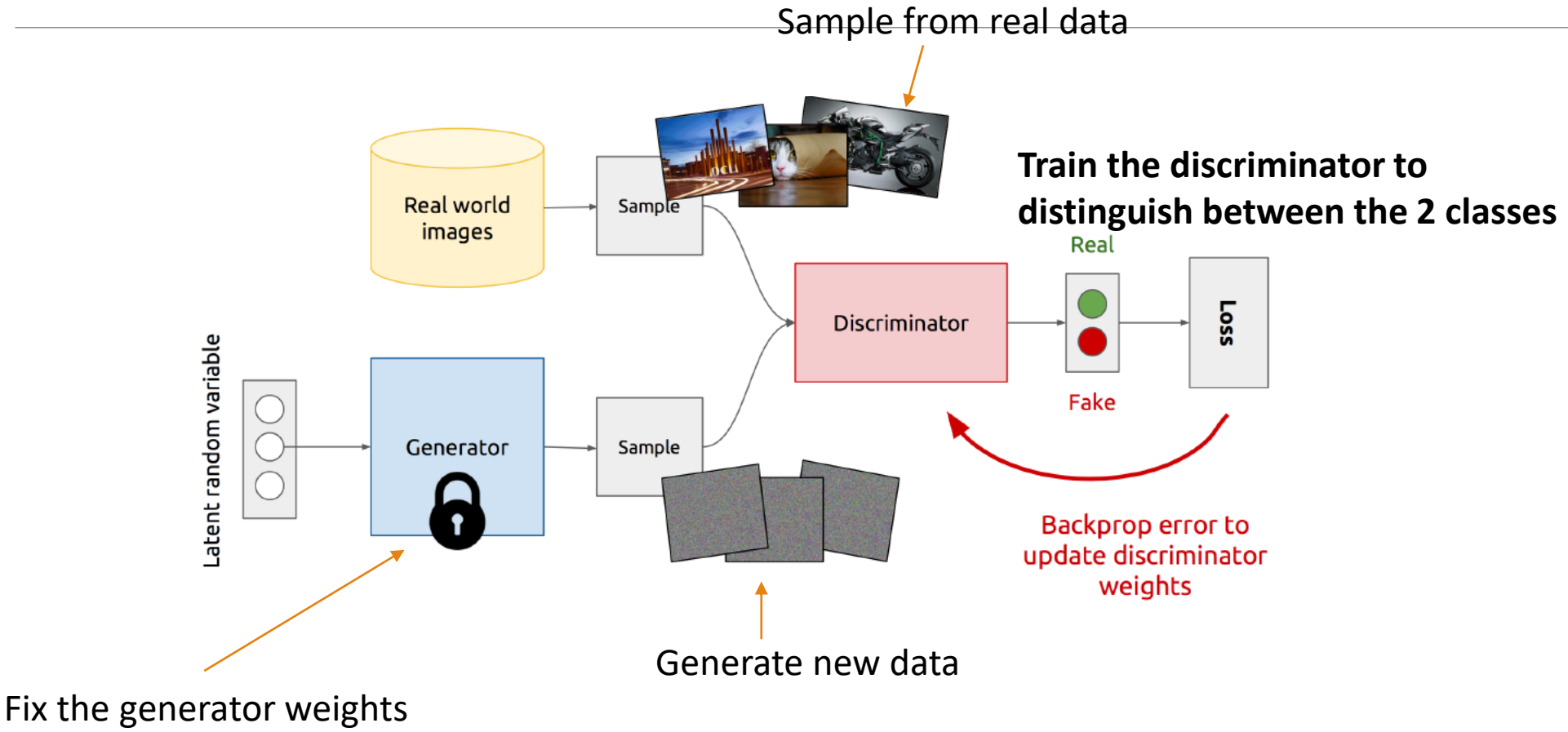
https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

GANs results over time

Karras, Tero, et al. "Analyzing and improving the image quality of stylegan." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

Have a look at https://www.thispersondoesnotexist.com/

Training the discriminator



Discriminator loss

Solves a binary classification problem

- Binary cross-entropy:

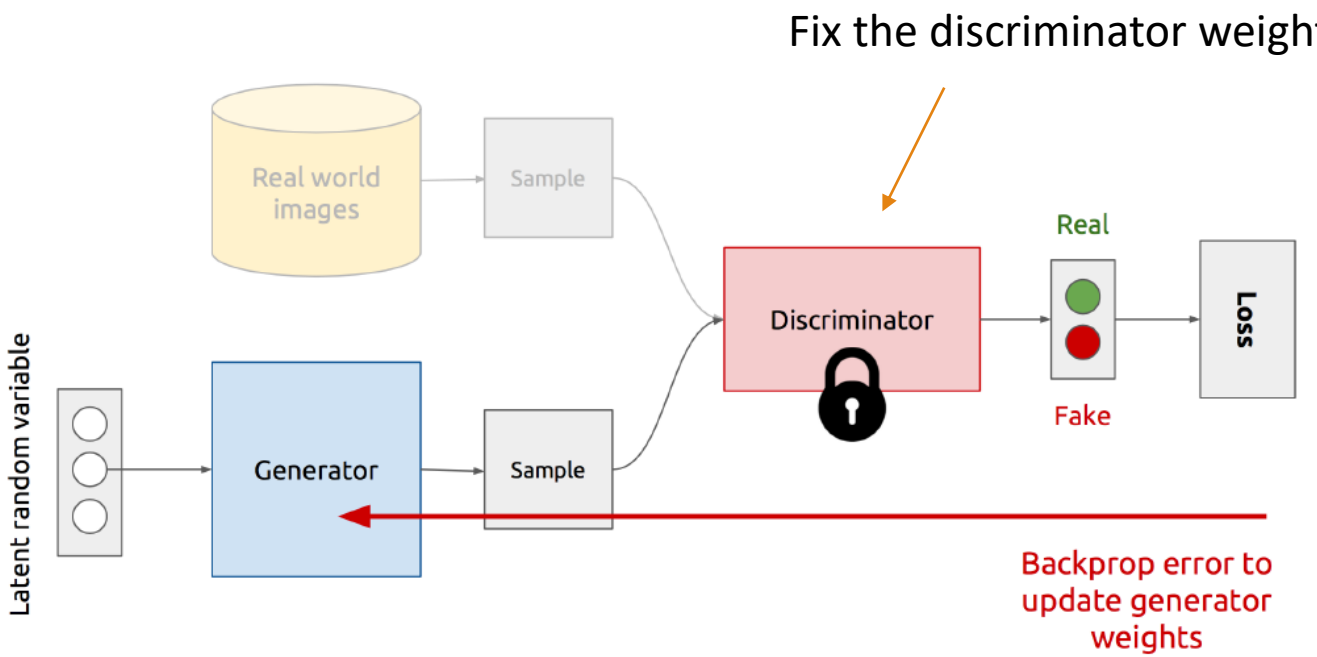
$$-\frac{1}{n}\sum_{i=1}^{n}[y_i \cdot \log(\mathrm{p}(\mathrm{y_i})) + (1-y_i)\cdot \log(1-\mathrm{p}(\mathrm{y_i}))]$$

$$y_i = \begin{cases} 1\ for\ real\ data \\ 0\ for\ gen.data \end{cases}$$

- For n real instances x_i and m generated instances $G(z_j, \theta)$, it translates to:

$$-\frac{1}{n}\sum_{i=1}^{n}\log(\mathrm{D}(\mathrm{x_i}, \phi)) - \frac{1}{m}\sum_{j=1}^{m}\log(1-\mathrm{D}\left(G\left(z_j, \theta\right)\right))$$

Training the generator



Generator loss

Aims at misleading the discriminator when classifying the generated data -> the discriminator should assign high probabilities

- For n generated instances $G(z_i, \theta)$:

$$+ \frac{1}{n} \sum_{i=1}^{n} \log(1 - \mathrm{D}\big(G(z_\mathrm{i}, \theta)\big)$$

Training a GAN

for #num_iterations

for k steps

Sample $\{z_1, z_2, \dots, z_m\}$ from the latent space and generate $G(z_i, \theta), i = 1..m$

Sample m examples $\{x_1, x_2, \dots, x_m\}$ from the data set of real instances

Update the discriminator by descending gradient:

$$\nabla_\phi - \frac{1}{m}\sum_{i=1}^{m} \log(D(x_i, \phi)) - \frac{1}{m}\sum_{j=1}^{m} \log(1 - D\left(G\left(z_j, \theta\right)\right))$$

end for

Sample $\{z_1, z_2, \dots, z_m\}$ from the latent space and generate $G(z_i, \theta), i = 1..m$

Update the discriminator by descending gradient:

$$\nabla_\theta + \frac{1}{n}\sum_{i=1}^{n} \log(1 - D\left(G(z_i, \theta)\right)$$

End for

A minimax game

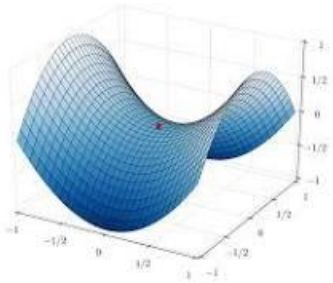
$$\hat{\phi}\hat{\theta} = \underset{\theta}{\text{argmax}}[\underset{\phi}{\text{argmin}}[-\frac{1}{n}\sum_{i=1}^{n}\log(\mathrm{D}(\mathrm{x_i},\phi)) - \frac{1}{m}\sum_{j=1}^{m}\log(1-\mathrm{D}\left(G\left(z_j,\theta\right)\right))]$$

$$\hat{\phi}\hat{\theta} = \underset{\theta}{\text{argmin}}[\underset{\phi}{\text{argmax}}[\frac{1}{n}\sum_{i=1}^{n}\log(\mathrm{D}(\mathrm{x_i},\phi)) + \frac{1}{m}\sum_{j=1}^{m}\log(1-\mathrm{D}\left(G\left(z_j,\theta\right)\right))]$$

- The generator tries to find new ways to fool the discriminator
- The discriminator searches for new ways to distinguish generated samples from real examples.

Solution: Nash equilibrium – the saddle point

- G will sample from the same distribution as X
- D(x)=0.5 for every image, real or generated

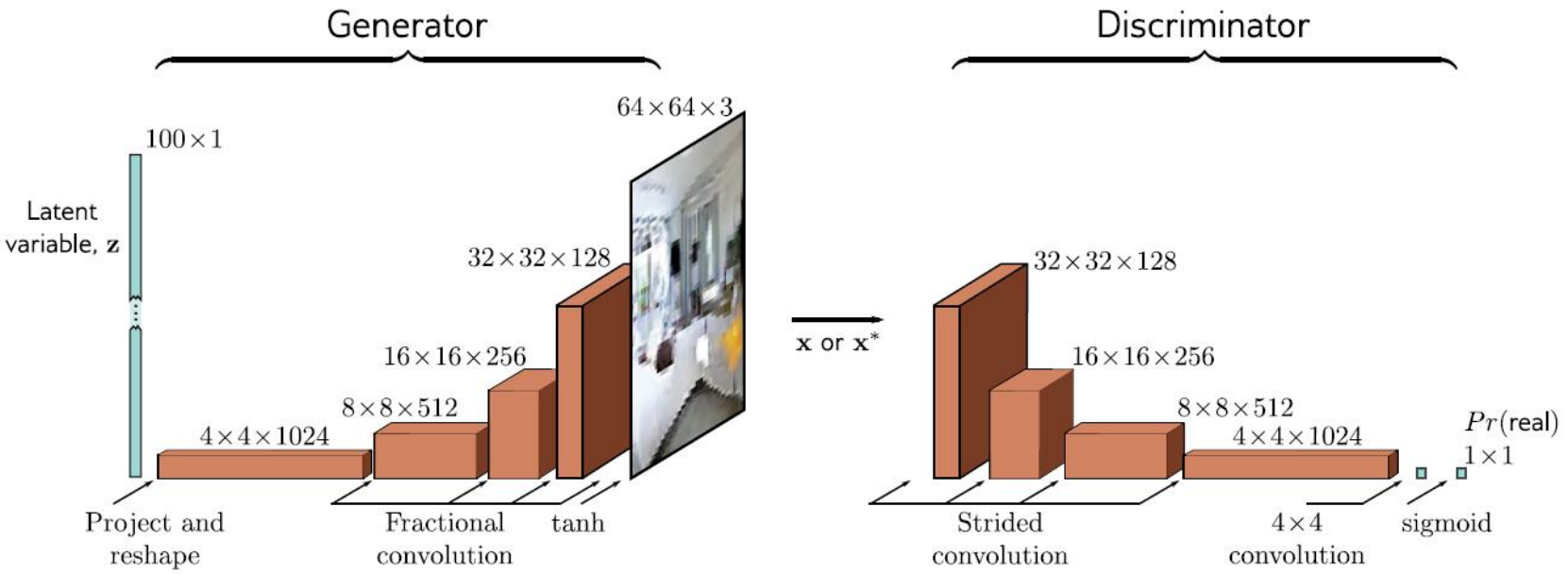


Generated images

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks." *NeurIPS* (2014).

DCGAN

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations*.



Replace FC hidden layers with Convolutions

Eliminate pooling layers and use

- downsampling convolutions for the Discriminator
- transposed convolutions for the Generator

Use Batch Normalization after each layer

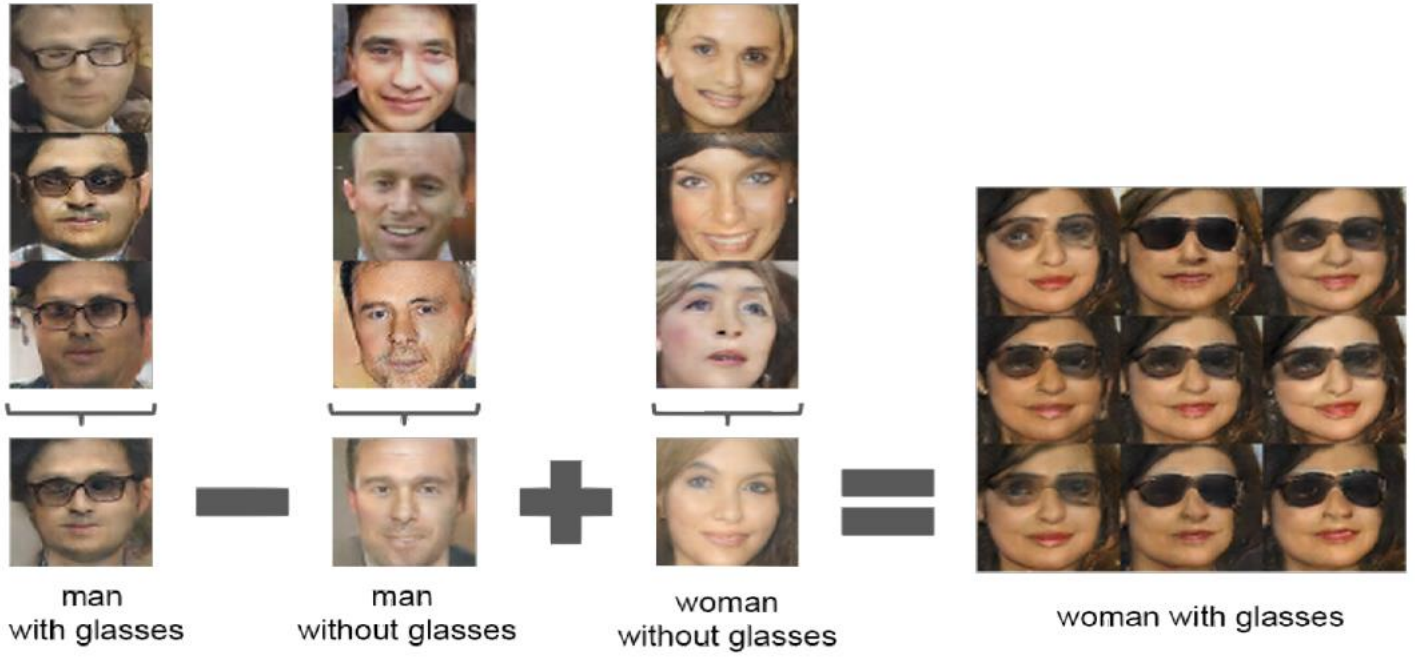
Inside Generator

- Use ReLU for hidden layers
- Use Tanh for the output layer

About transposed convolutions and artifacts:
https://distill.pub/2016/deconv-checkerboard/

DCGAN Results

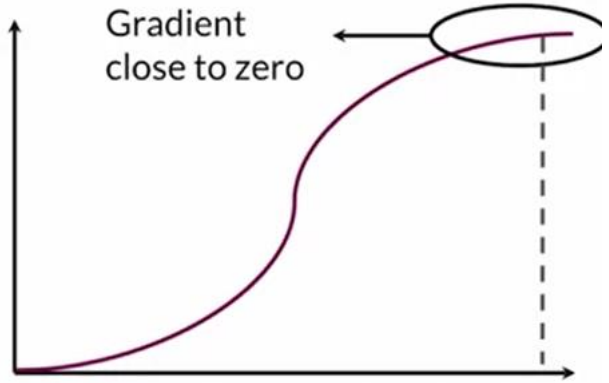
DCGAN
Latent vectors capture interesting patterns



Training GANs

- In an alternating way
- The two models should be at similar skill level

- What would be the feedback for the generator if the discriminator is superior?
  - The discriminator has an easier task
  - Gives less informative feedback for perfect answers – vanishing gradient



There is a very fine balance between the quality of the discriminator and the generator. If the discriminator becomes too good, the training updates of the generator are attenuated.

Problems in training GANs
Mode collapse

The generator produces good samples, but very few of them - gets stuck in local minima, producing very similar images

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. *International Conference on Machine Learning*, 214–223

Solutions to Mode collapse (1)

Let the Discriminator look at the entire batch instead of single examples

If there is lack of diversity, it will mark the examples as fake

-> Generator will be forced to produce diverse samples.

Extract features that capture diversity in the mini-batch

- For e.g. L2 norm of the difference between all pairs from the batch

- **Feed those features to the discriminator along with the image**
  - Thus, Discriminator will rely on those features for classification
  - Will force the Generator to match those feature values with the real data -> forces generated batch to have similar variation to real batch
  - Will generate diverse batches

Salimans, Tim, et al. "Improved techniques for training gans." *Advances in Neural Information Processing Systems*. 2016.

Solutions to Mode collapse (2)

It can be shown that the BCE loss for GANs is equivalent to minimizing a distance between probability distributions of real and fake images

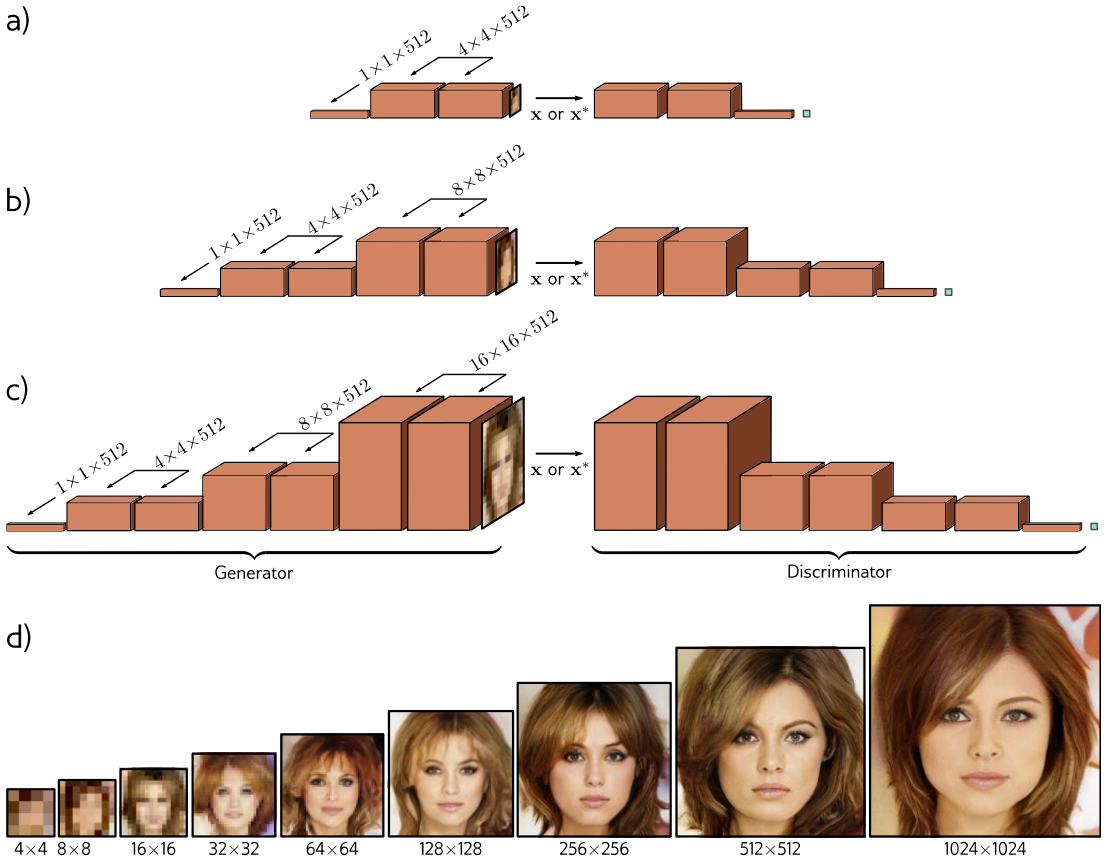
If the probability distributions are completely disjoint (the discriminator can perfectly separate the generated and real samples), this distance is infinite, and any small change to the generator will not decrease the loss.

Solution: replace the BCE loss with the **Wasserstein loss**:

- Quantity of work required to transport the probability mass from one distribution to create the other
- is well-defined even when the distributions are disjoint and decreases smoothly as they become closer to one another

$$\sum_j \mathrm{f}[\mathrm{g}[\mathbf{z}_j, \boldsymbol{\theta}], \boldsymbol{\phi}] - \sum_i \mathrm{f}[\mathbf{x}_i, \boldsymbol{\phi}]$$

Tricks for improving performance
Progressive growing



Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. *International Conference on Learning Representations*.

Results when interpolating the latent space

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. *International Conference on Learning Representations*.

Tricks for improving performance Truncation

Only sample random values of latent variables that are less than a threshold τ.

Reduces the variation in the samples but improves their quality

Brock, A., Donahue, J., & Simonyan, K. (2019). Large scale GAN training for high fidelity natural image synthesis. *International Conference on Learning Representations*.

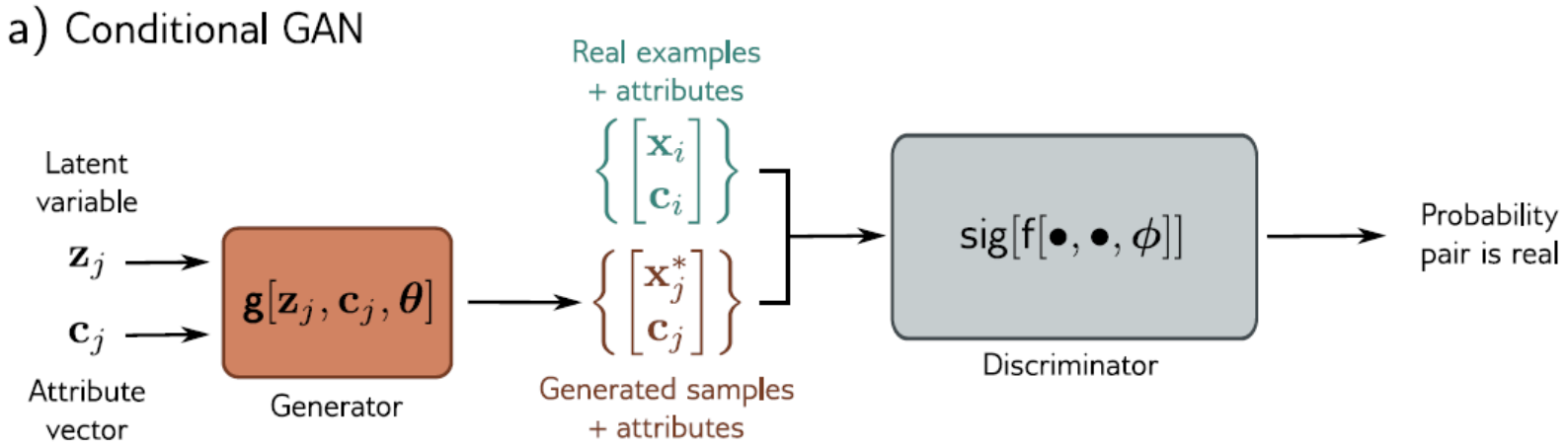
Conditional generation

Motivation:

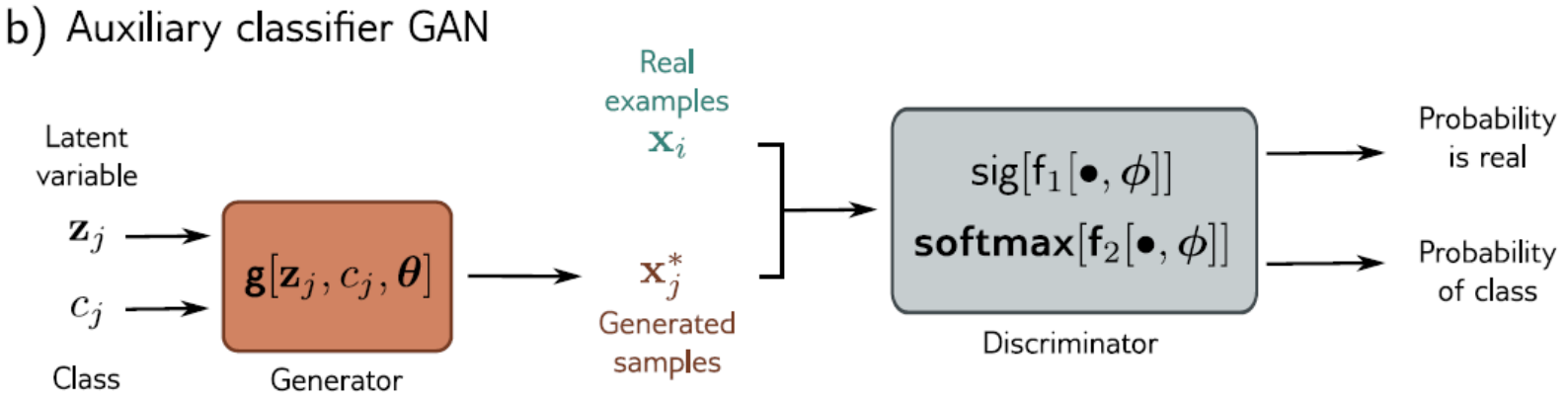
- Generate samples having certain properties (or class)
- Tackle the mode collapse problem

Needs labeled datasets

Conditional generation



Feeds the class to both the generator and discriminator
-how? different input shapes



Feeds the class to the generator
The discriminator predicts also the class

ACGAN Results

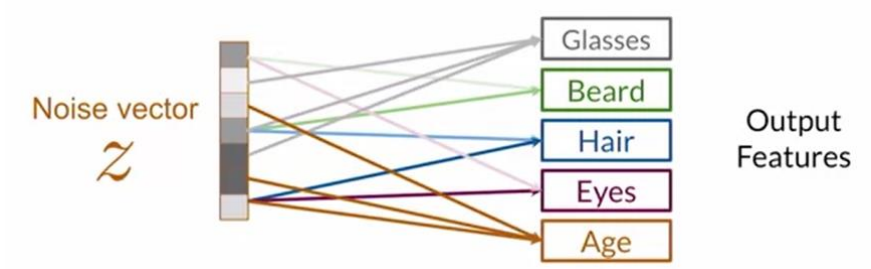
Odena, A., Olah, C., & Shlens, J. (2017). Conditional image synthesis with auxiliary classifier GANs. *International Conference on Machine Learning*, 2642–2651.

Controllable generation

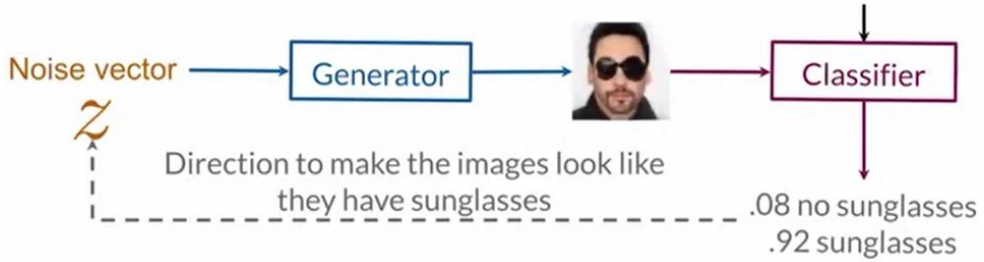
The latent space encodes image features: colors, textures, pose, objects

Idea: manipulate z to obtain images having desired features

Problems: entanglement in the latent space that makes impossible to control single output features (especially when the dimensionality is low)



Possible solution: use a pretrained classifier that distinguishes the images you want and backpropagate the classifier gradients to z in order to direct z to the type of images you want



Controlable generation InfoGAN

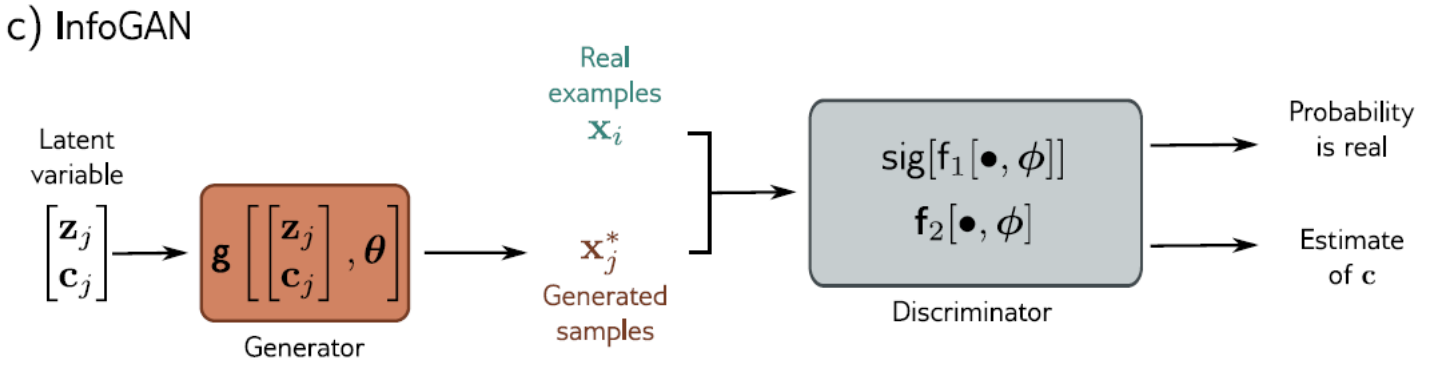
$$I(X;Y) = \sum_{x,y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right)$$

Better idea: disentaglement - add individual dimensions to independently and automatically (not apriori) capture key attributes of the image. **There is no need for class labels**

The GAN loss also maximizes the mutual information between c and generated data $\lambda I(c; G(z,c))$

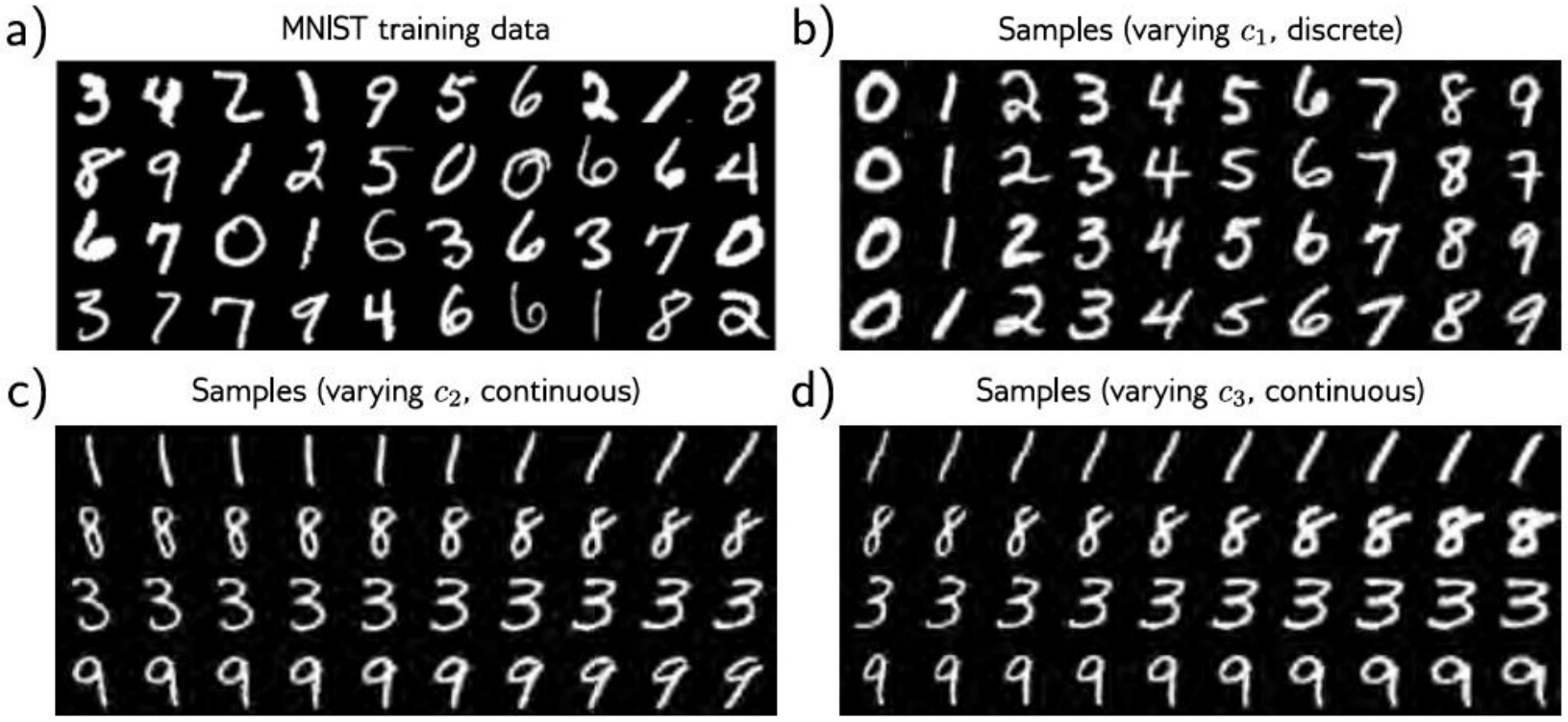
c vector captures the key variations in the image

- Discrete attributes (CE loss)
- Continuous attributes (MSE loss)



Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).

InfoGAN Results



Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).

How evaluate GANs?

It is difficult: no ground truth

Two perspectives:

Fidelity – quality or realism of the image

- Extract features/embeddings from images (both real and fake) using pretrained classifier CNNs
- Fit multivariate normal distributions
- Compute Frechet distance between real and fake datasets:

$$\|\mu_X - \mu_Y\|^2 + \mathrm{Tr}\left(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y}\right)$$

Diversity – variety in the generated image set

References

- Chapter 15 in *Prince, Simon JD. Understanding Deep Learning. MIT press, 2023*. https://udlbook.github.io/udlbook/