Generative Adversarial Networks

Vikky Masih

Research Scholar, MFS-DS&AI, IITG

Generative Modelling

- The network models a distribution over samples. Unsupervised.
  - Quality measure: Sample from the distribution and compare with original.

Samples from Deep Boltzmann Machines,
(Salakhutdinov and Hinton, 2009)
[CIFAR10 Dataset]

Samples from Progressive GAN
(Karras et al., 2018)
[CelebFaces Attributes (CelebA) Dataset]

Training Data $\sim p_{data}(x)$
Generated Sample $\sim p_{model}(x)$

We want both distributions to be similar.

Deep Generative Model based on MLE

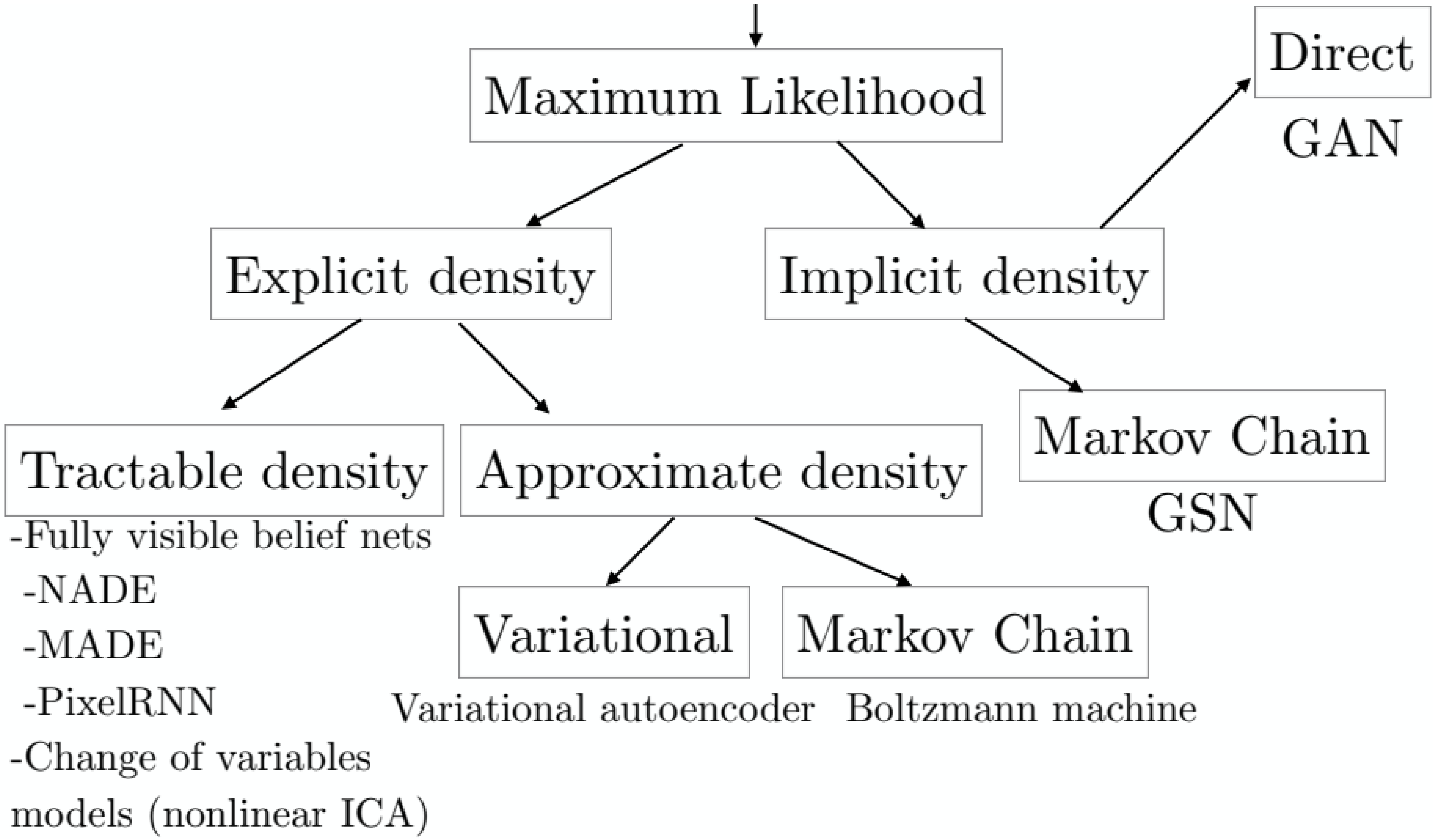


Image Source: Goodfellow, I. "NIPS 2016 tutorial: Generative adversarial networks. arXiv 2016." *arXiv preprint arXiv:1701.00160*.

PixelRNN – Explicit Tractable Density

Explicit density model

Use chain rule to decompose likelihood of an image x into product of 1-d distributions:

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, ..., x_{i-1})$$

Likelihood of image x

Probability of i'th pixel value given all previous pixels

Then maximize likelihood of training data

Image Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

PixelRNN – Explicit Tractable Density

Generate image pixels starting from corner

Dependency on previous pixels modeled using an RNN (LSTM)

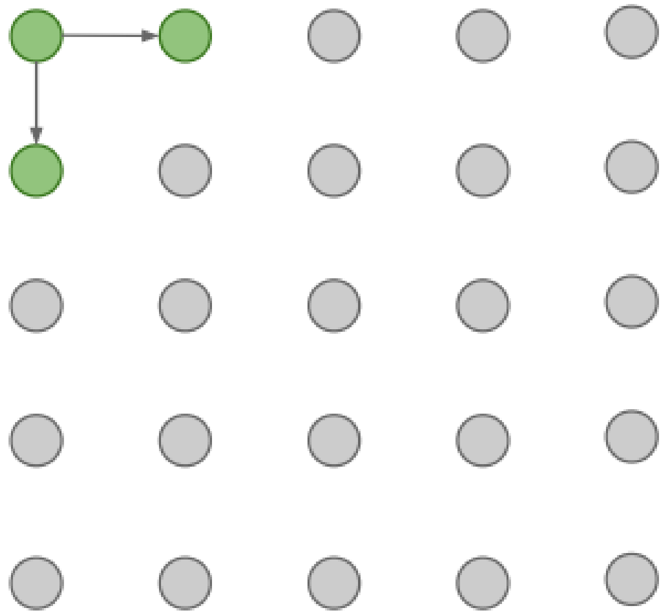
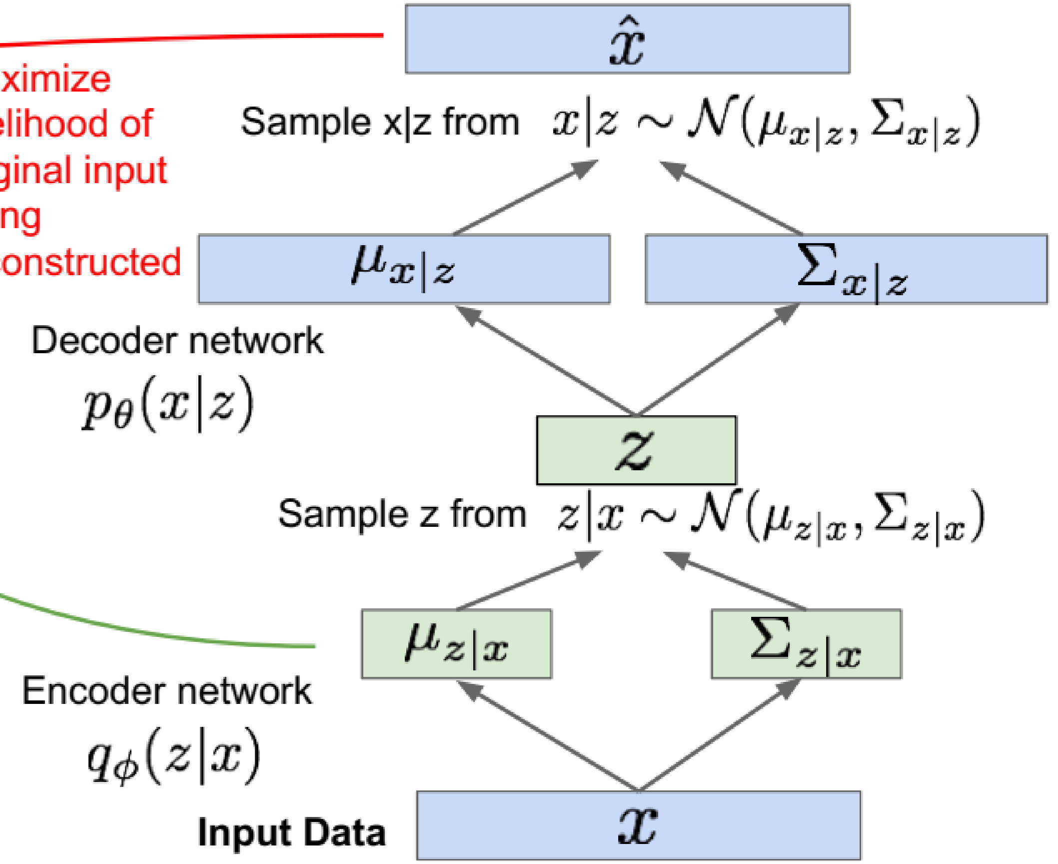


Image Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

Variational Autoencoders – Explicit Intractable

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

Maximize likelihood of original input being reconstructed



Make approximate posterior distribution close to prior

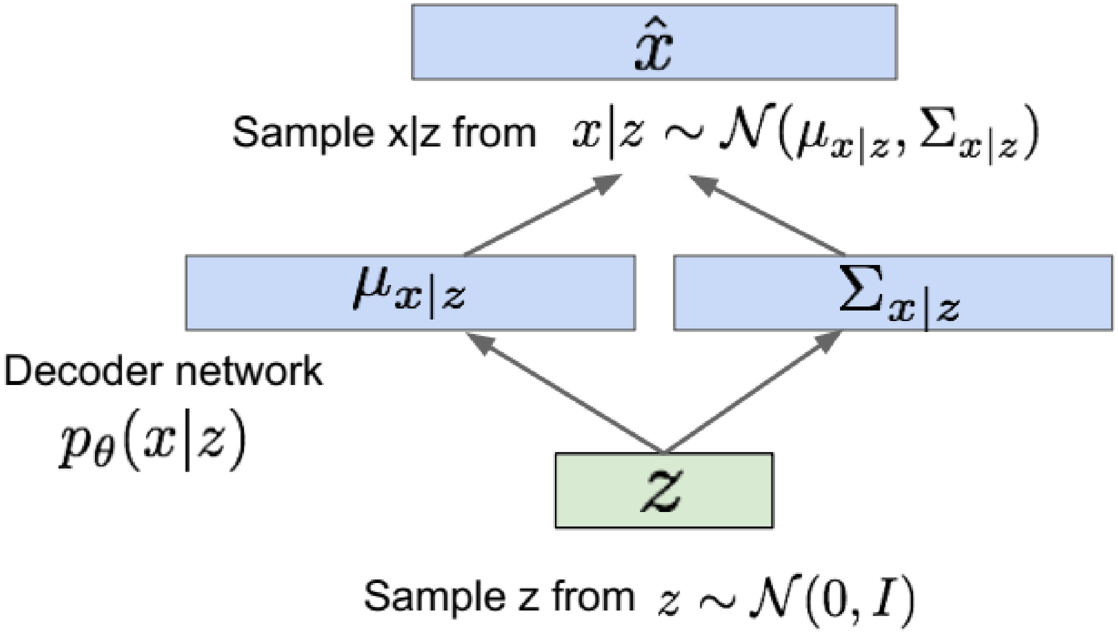
$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$

$$\log p_\theta(x^{(i)}) = \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))$$

Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

Variational Autoencoders – Sampling

Use decoder network. Now sample z from prior!



Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Data manifold for 2-d **z**

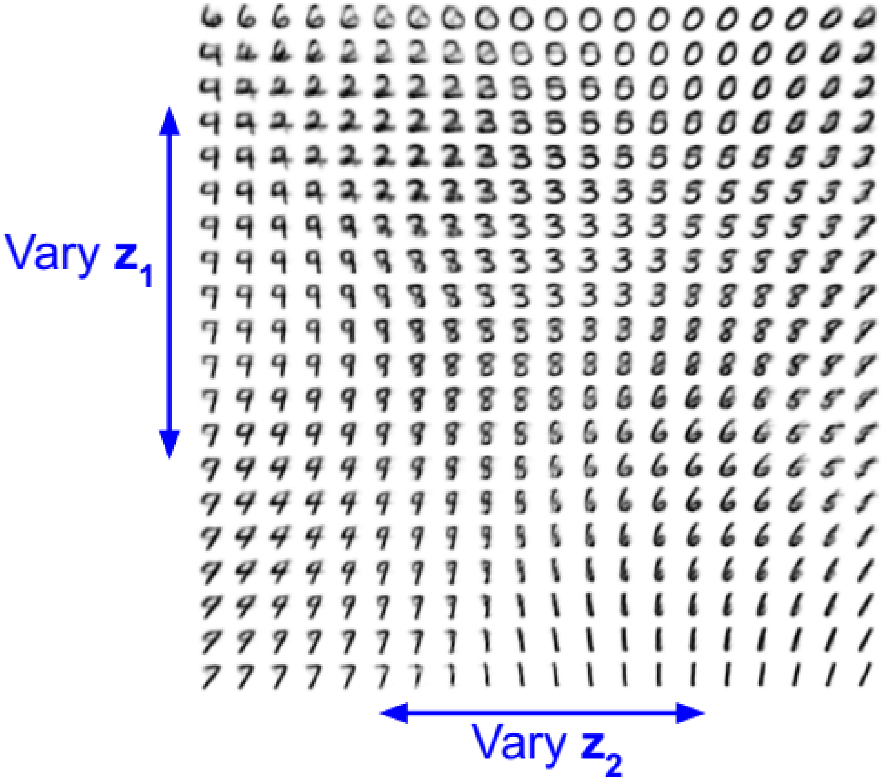


Image Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

PixelCNN and VAE

PixelCNNs define tractable density function, optimize likelihood of training data:

$$p_\theta(x) = \prod_{i=1}^{n} p_\theta(x_i|x_1, ..., x_{i-1})$$

VAEs define intractable density function with latent **z**:

$$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$$

Cannot optimize directly, derive and optimize lower bound on likelihood instead

Image Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

Generative Adversarial Networks

- Does not work with explicit density function.
- Game theoretic approach: Zero-sum game.
  - Learn to generate samples from the training data distribution.
  - Sample from a simple distribution; Learn to transform it into a data sample.

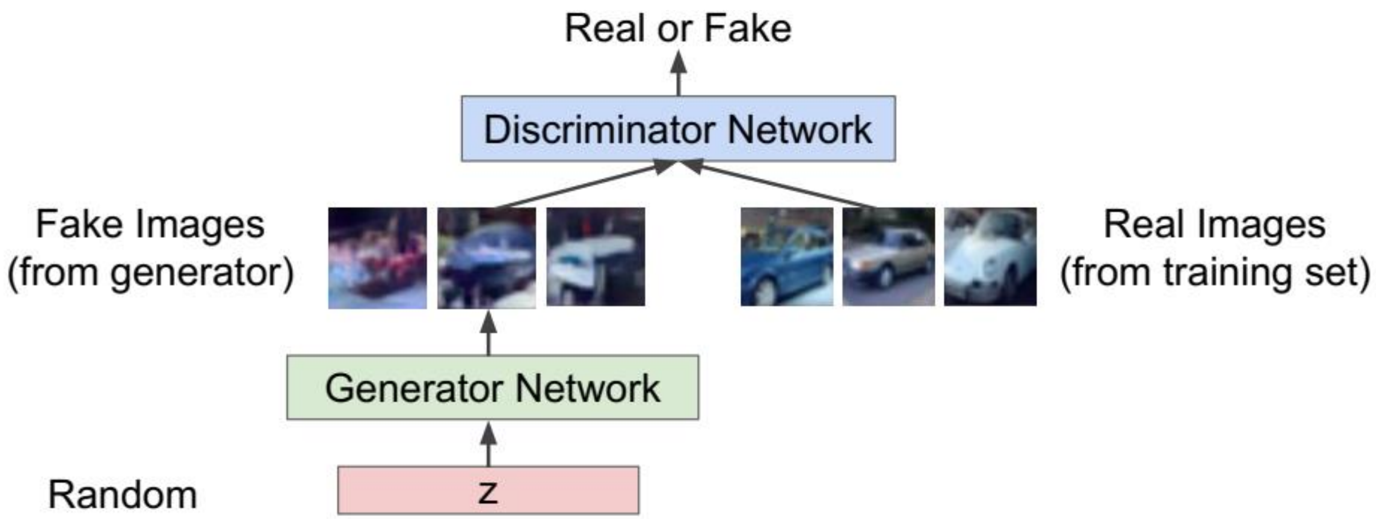


Image Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

Generative Adversarial Networks

Discriminator outputs likelihood in (0,1) of real image

Minimax objective function:

$$\min_{\theta_g}\max_{\theta_d}\left[\mathbb{E}_{x\sim p_{data}}\log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z\sim p(z)}\log\left(1-\underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}\right)\right]$$

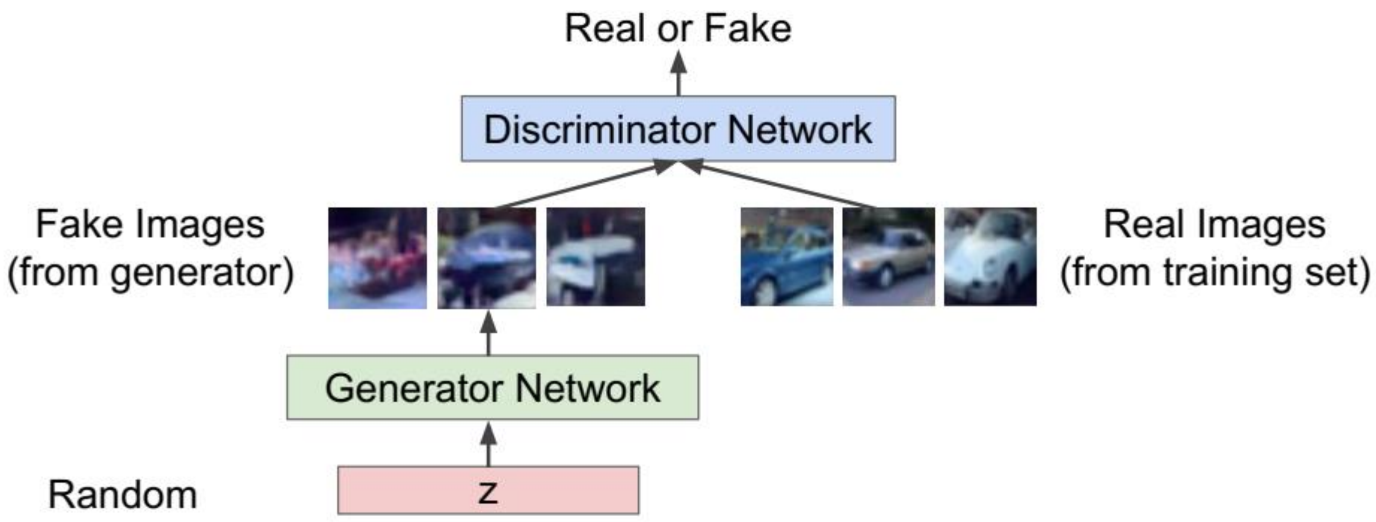


Image Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

Generative Adversarial Networks

Discriminator outputs likelihood in (0,1) of real image

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

Discriminator (θ_d) wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)

Generator (θ_g) wants to **minimize objective** such that D(G(z)) is close to 1 (discriminator is fooled into thinking generated G(z) is real)

Slide Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

Generative Adversarial Networks

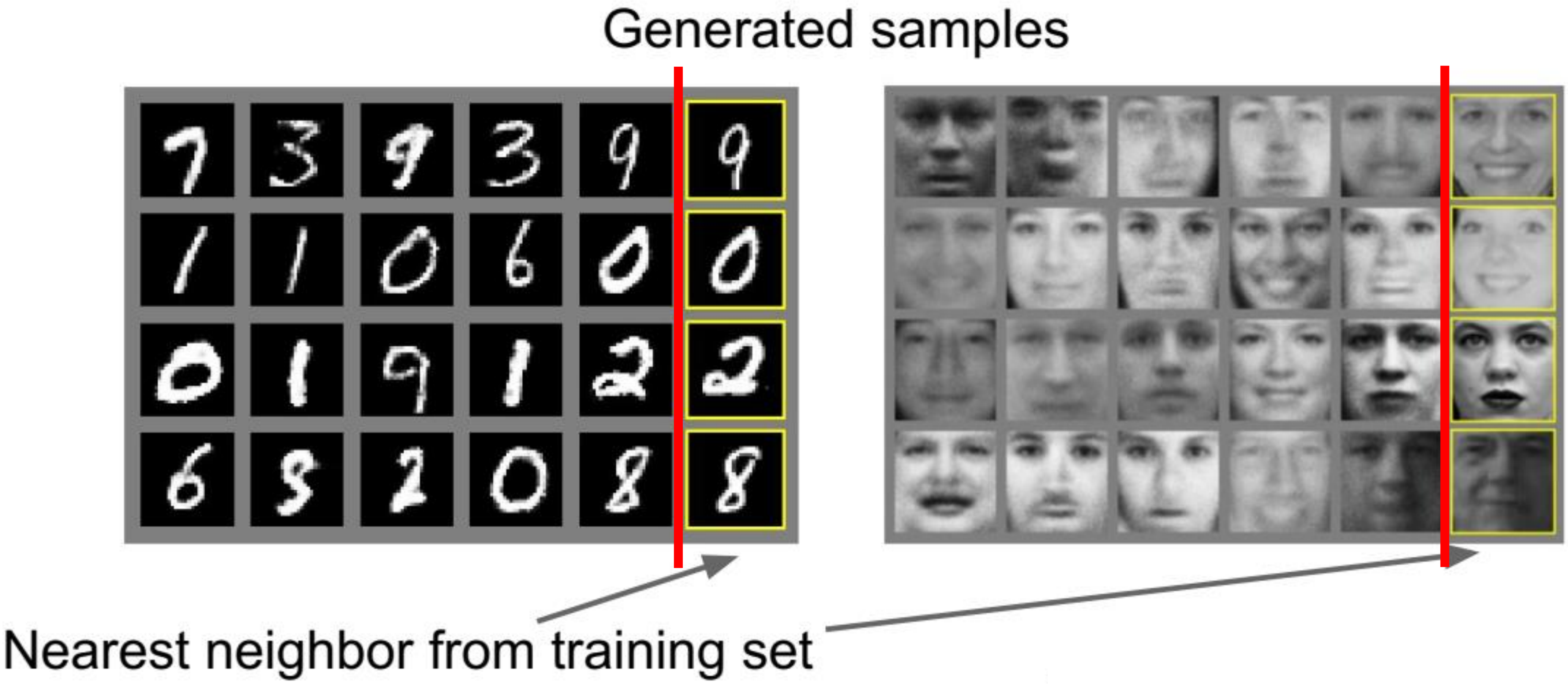


Image Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

GAN: Interpretable Vector Math

Slide Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

GAN

Don't work with an explicit density function
Take game-theoretic approach: learn to generate from training distribution through 2-player game

Pros:
- Beautiful, state-of-the-art samples!

Cons:
- Trickier / more unstable to train
- Can't solve inference queries such as p(x), p(z|x)

Active areas of research:
- Better loss functions, more stable training (Wasserstein GAN, LSGAN, many others)
- Conditional GANs, GANs for all kinds of applications

Slide Source: CS321n Lecture notes 2019 - Fei-Fei Li, Justin Johnson and Serena Yeung

References

- Goodfellow, I. "NIPS 2016 tutorial: Generative adversarial networks. arXiv 2016." arXiv preprint arXiv:1701.00160.
- Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." Advances in neural information processing systems 27 (2014).
- Li, Fei-Fei, Justin Johnson, and Serena Yeung. “Stanford University CS231n: Deep Learning for Computer Vision.” Accessed November 1, 2023. http://cs231n.stanford.edu/.