Generative Adversarial Nets

NIPS 2014, I. Goodfellow et. al.

Presentation by

GIST College Physics Concentration Hanse Kim

Basic Information

Paper : Generational Adversarial Nets

Author(s) : Ian J. Goodfellow, et. al. 7

Journal : Advances in Neural Information Processing Systems 27 (NIPS 2014)

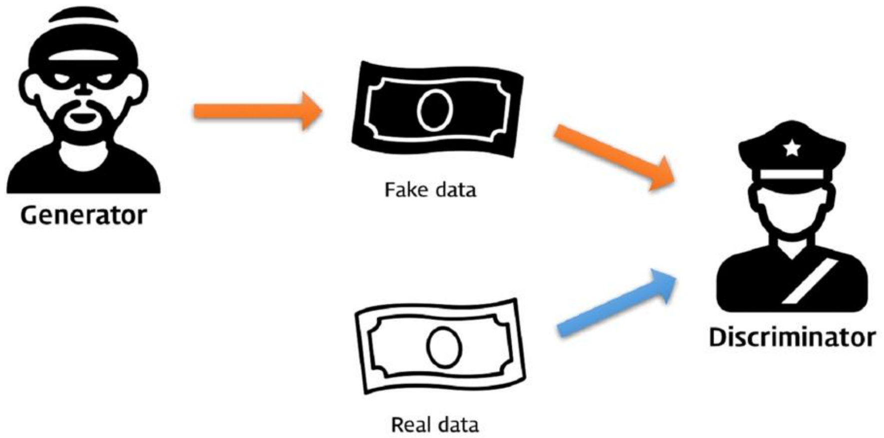
Citations : 26219 (as of 2021/01/10)

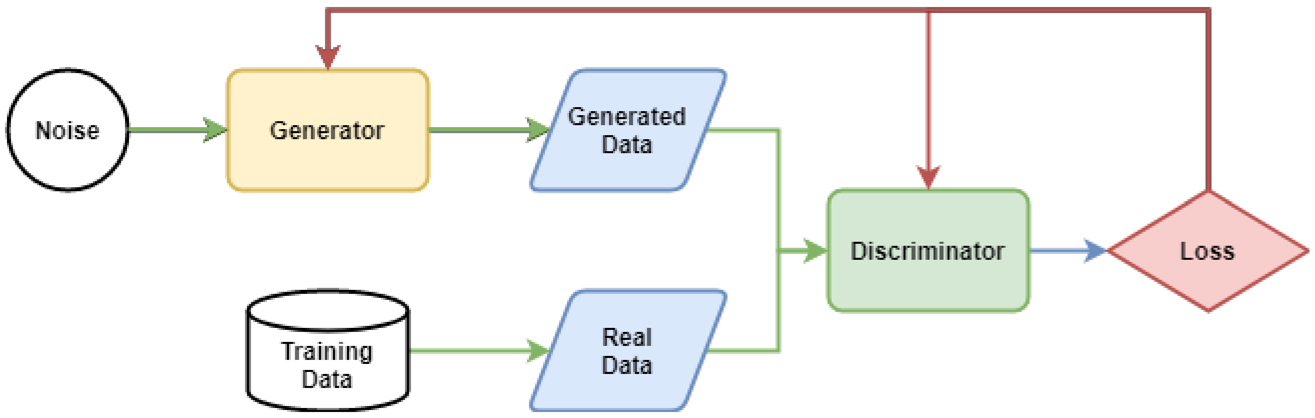
Theoretic Proposition/Results : Section 3-4

Experimental Analysis : Section 5-6

Introduction

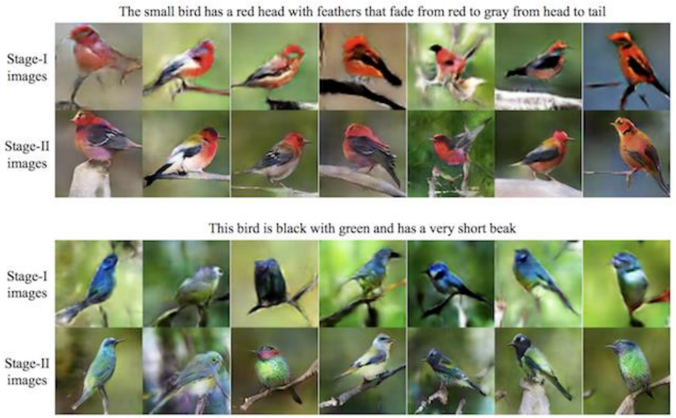
- GAN : Generative Adversarial Network
  - Simultaneous training of generative model and discriminative model
  - Generative model : generates data from training data
  - Discriminative model : adversary to generative model, tries to distinguish training data from generated data
  - Minimax two-player game; analogy of counterfeiter



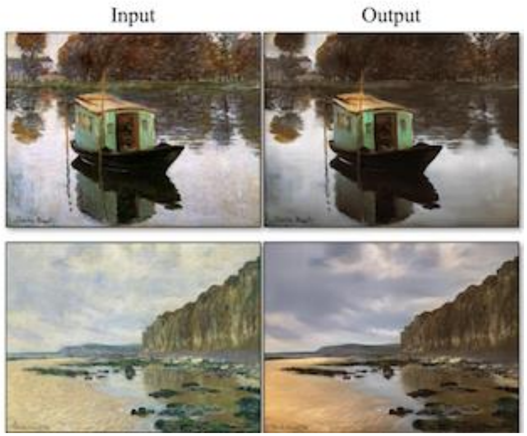


Introduction : Applications of GAN

- Both generative and discriminative models are useful
  - Generative model especially useful; little progress prior to development of GAN, especially for images and videos
  - Semi-supervised learning : “fill in” missing data
  - Reinforcement learning : simulating possible futures
  - Continuous improvements, application to non-ML fields expanded as well
- Image Generation/Recognition



Text to image



Painting to photo

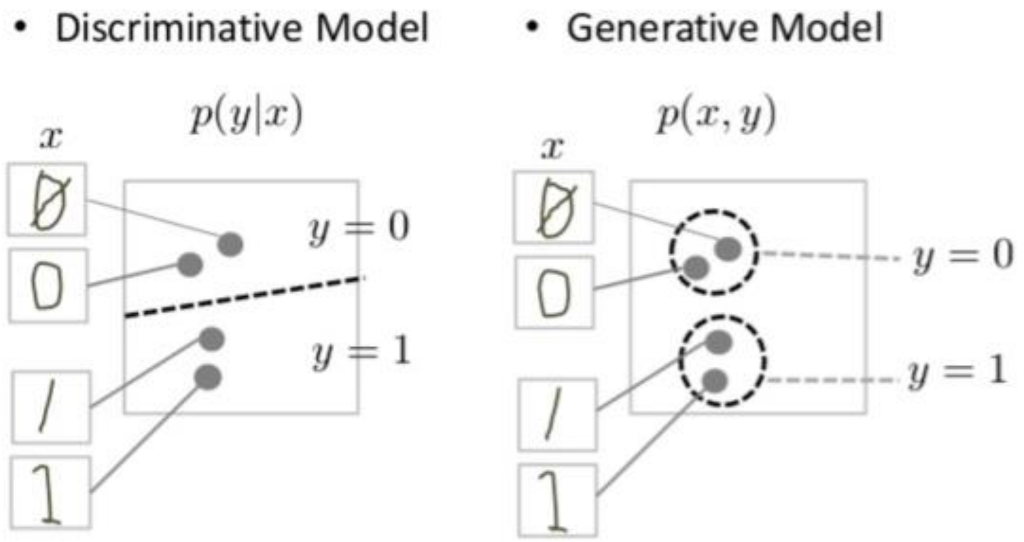
Frontal view generation

Introduction : Applications of GAN

- Natural Language
  - Generation of natural language : https://arxiv.org/abs/1705.10929
  - Text generation : https://arxiv.org/abs/1709.08624
- Art
  - Music generation : https://arxiv.org/abs/1709.06298
  - Painting : Generative imagine inpainting with contextual addition, Jiahui Yu et. al.
- Physics
  - Improvement of astrophotography : https://arxiv.org/abs/1702.00403
  - Prediction of distribution of dark matter : https://arxiv.org/abs/1706.02390
  - High energy particle collision : https://arxiv.org/abs/2012.06582
  https://arxiv.org/abs/1909.04451

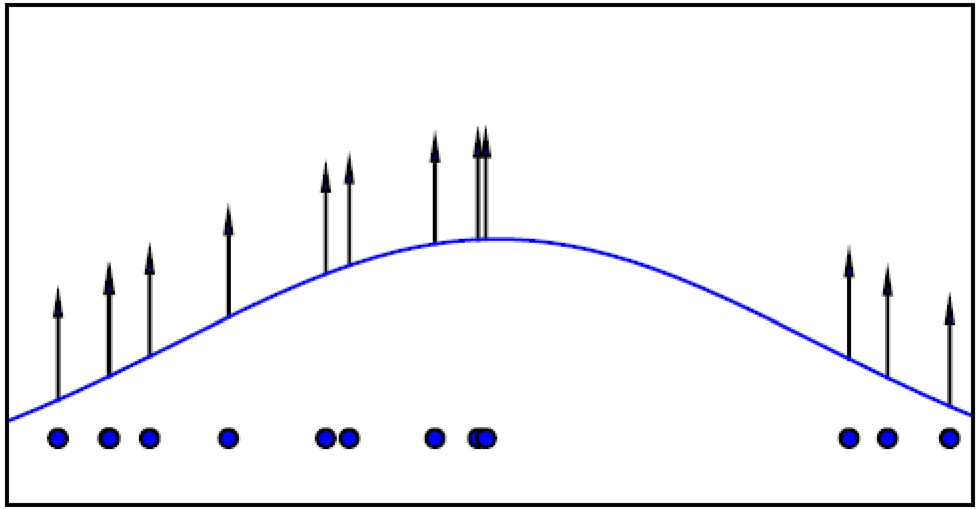
Theoretical Proposition : Background

- Discriminative Models : many successful cases prior to GAN
  - Conditional probability of data
  - Decision trees etc.
  - Labelling high-dimensional, rich sensory inputs; voice recognition etc.
- Generative Models
  - Joint distribution of data
  - 'Producing' data with the model
  - Maximum likelihood estimation : basis for most generative models



Theoretical Proposition : Background

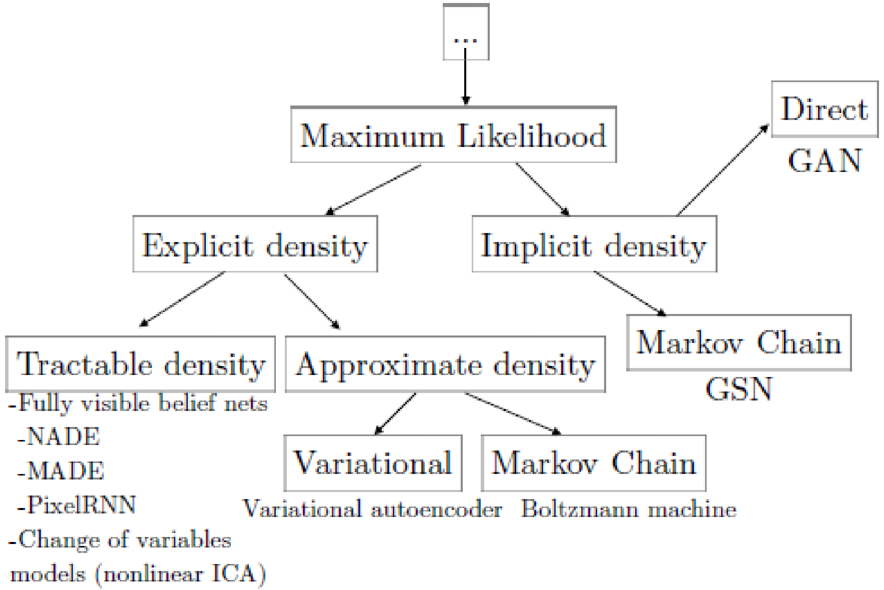
- Maximum Likelihood Estimation
  - Likelihood function : probability of an outcome interpreted as function of parameter
    $\mathcal{L}(\theta|x) = p_\theta(x) = P_\theta(X = x)$
  - Choose parameters to maximise likelihood of training (observed) data
  - log space used to simplify calculations
  - Likelihood function 'pushed up' at sample points of the dataset



$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

Theoretical Proposition : Related Research

- Explicit Density Model
  - Explicitly define density function $p_\theta(x)$
  - Maximising likelihood is straightforward
  - Difficulty in designing a model that captures complexity of data while being computationally feasible
- FVBN
  - Decomposition into products with chain rule
- Sampling approximations
  - VAE : sampling from ideal function, lower bound to likelihood
  - MCMC : sampling with Markov chain techniques

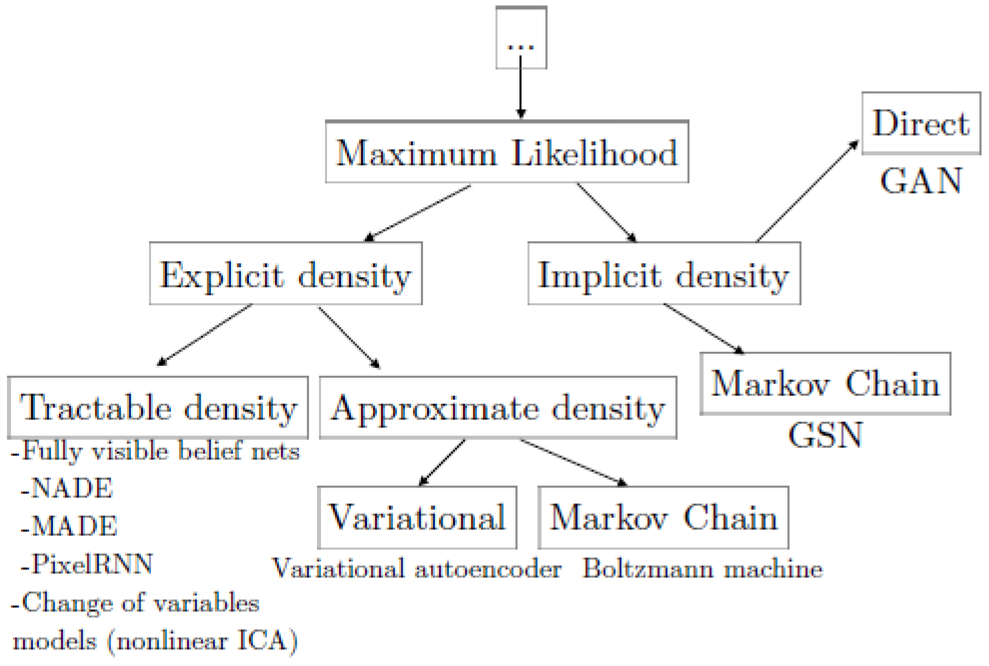


FVBN : $p_{\text{model}}(\boldsymbol{x}) = \prod_{i=1}^{n} p_{\text{model}}\left(x_i \mid x_1, \ldots, x_{i-1}\right)$

VAE : $\mathcal{L}(\boldsymbol{x};\boldsymbol{\theta}) \leq \log p_{\text{model}}(\boldsymbol{x};\boldsymbol{\theta})$

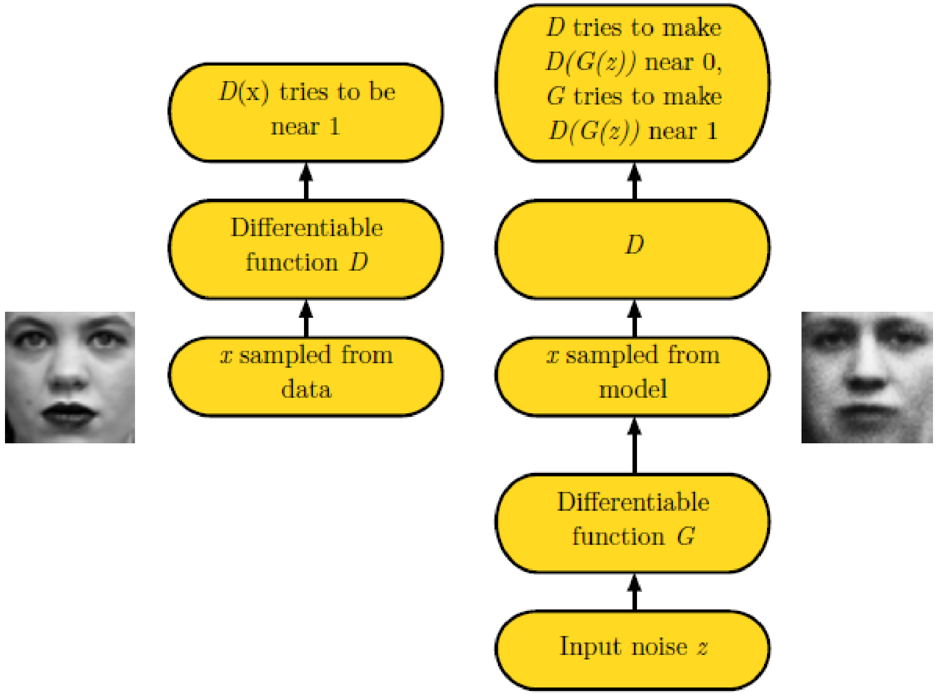
Theoretical Proposition : Related Research

- Implicit Density Model
  - Train the model interacting only indirectly with p_{model}, by sampling from it
  - GSN : Utilise Markov chains to draw samples from p_{model}
  - Markov chains fail to scale to high dimensional spaces, increased computational costs
- GAN
  - GAN is also an implicit density mode
  - Sample generation only requiring single step, no direct sampling from training data for generator



Theoretical Proposition : Framework

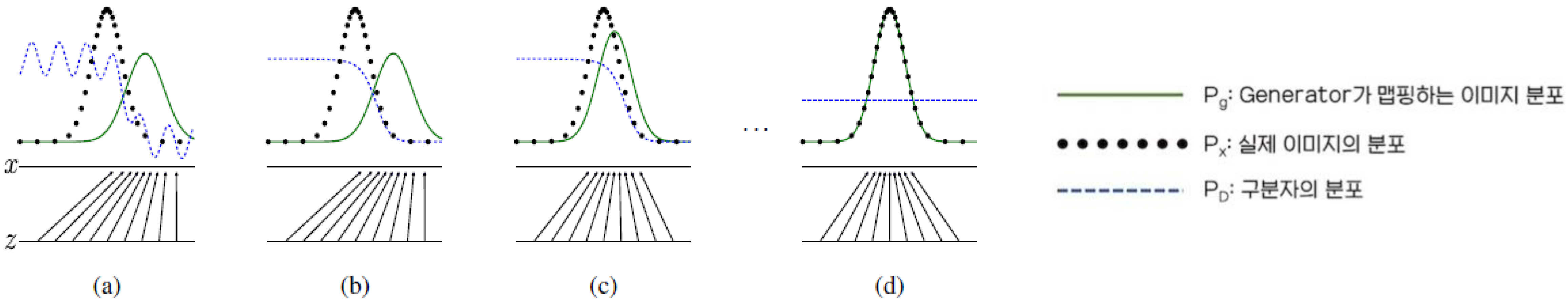
- Discriminative model D
  - Traditional supervised learning techniques
  - Goal : output $D(x)$ as near 1
- Generative model G
  - $p_z(z)$: prior over latent variables, noise
  - Goal : Capture distribution of data x, generate fake sample $G(z)$
- "Game" between both models
  - Loss functions : D wishes to minimise loss only by controlling $\theta^{(D)}$, vice versa for G
  - Minmax game between D, G
  - If both are equally competent, Nash equilibrium is at $D(x) = \frac{1}{2}$ for all x



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

Theoretical Proposition : Adversarial Nets

- Framework
  - Both G and D can be multilayer perceptrons, differentiable w.r.t. inputs x, z, and parameters
  - $G(z;\theta^{(g)})$: mapping to data space
  - $D(x;\theta^{(d)})$: mapping from data to scalar, probability that x came from the data rather than p_g
- Pedagogical Explanation of Training Process



Theoretical Results

- Theoretical Considerations
  - Wish to converge to a good estimator
  - Minimax game has $p_g = p_{data}$ as global optimum
  - Algorithm converges and optimises value function equation
  - Theoretic results in non-parametric situations

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
- Update the discriminator by ascending its stochastic gradient:

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

end for

- Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient:

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

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Theoretical Results

- Theoretical Considerations
  - Wish to converge to a good estimator
  - Minimax game has $p_g = p_{data}$ as global optimum
  - Algorithm converges and optimises value function equation
  - Theoretic results in non-parametric situations

Proposition 1. *For G fixed, the optimal discriminator D is* $D_G^*(\boldsymbol{x}) = \dfrac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$

Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-\log 4$.*

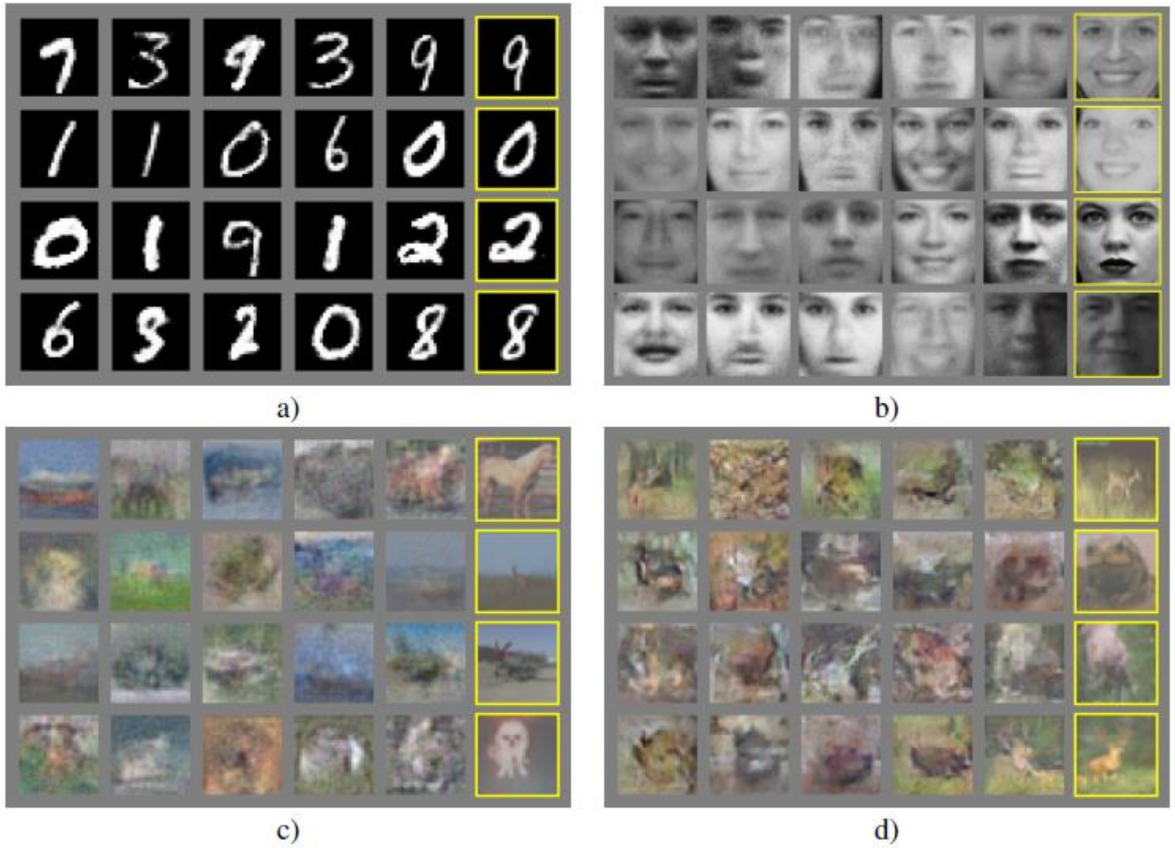
Proposition 2. *If G and D have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion*

$$\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))]$$

then p_g converges to p_{data}

Experimental Analysis

- a : MNIST, b : TFD, c, d : CIFAR-10
  - Images all generated samples from the model
  - Rightmost column shows nearest training example of neighbouring sample; model has not memorised training set



Conclusions

- Disadvantages
  - Finding equilibrium of a game is harder than optimising function
  - No explicit representation of generative model
  - Synchronisation between G, D important
- Advantages
  - Markov chains not required; high fidelity, low computation
  - Inference not used in learning
  - Generator not directly updated by training data
  - Only backpropagation used to obtain gradients
  - Wide variety of functions can be integrated into model

References

- Papers (which are not listed in previous slides)
  - NIPS 2016 Tutorial : Generative Adversarial Networks, Ian Goodfellow
- Websites
  - https://medium.com/thecyphy/gans-what-and-where-b377672283c5
  - https://dreamgonfly.github.io/blog/gan-explained/
  - http://jaejunyoo.blogspot.com/2017/01/generative-adversarial-nets-1.html
  - https://developers.google.com/machine-learning/gan
- Videos
  - https://www.youtube.com/watch?v=jB1DxJMUlxY