Advanced Machine Learning
Summer 2019

Part 17 – Generative Adversarial Networks
26.06.2019

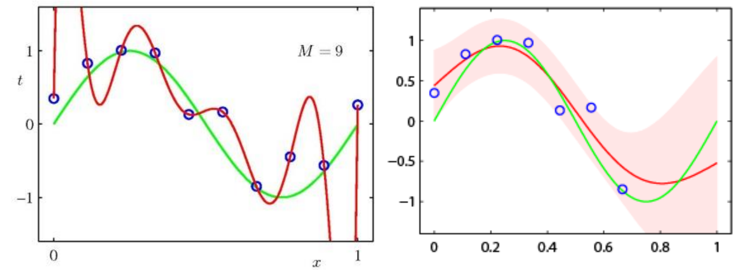
Prof. Dr. Bastian Leibe

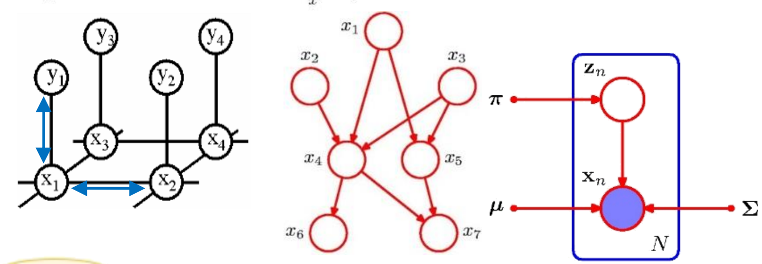
RWTH Aachen University, Computer Vision Group
http://www.vision.rwth-aachen.de

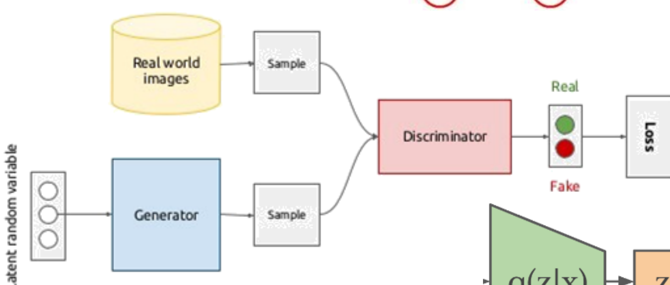
Course Outline

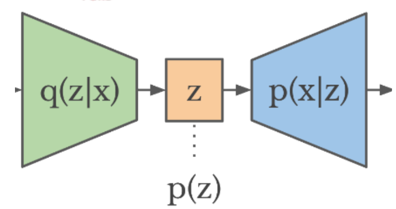
- Regression Techniques
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Kernels (Kernel Ridge Regression)
- Deep Reinforcement Learning
- Probabilistic Graphical Models
  - Bayesian Networks
  - Markov Random Fields
  - Inference (exact & approximate)
  - Latent Variable Models
- Deep Generative Models
  - Generative Adversarial Networks
  - Variational Autoencoders

$$f : \mathcal{X} \to \mathbb{R}$$









Topics of This Lecture

- Recap: Bayesian Mixture Models
- Generative Adversarial Networks (GANs)
  - Generative networks
  - GAN loss and training procedure
- Applications & Extensions
  - GANs for image generation
  - GANs for superresolution
  - Conditional GANs
- Problems of GANs
  - Problems during training
  - Conceptual problems
  - Extension: Wasserstein GANs

Recap: Bayesian Mixture Models

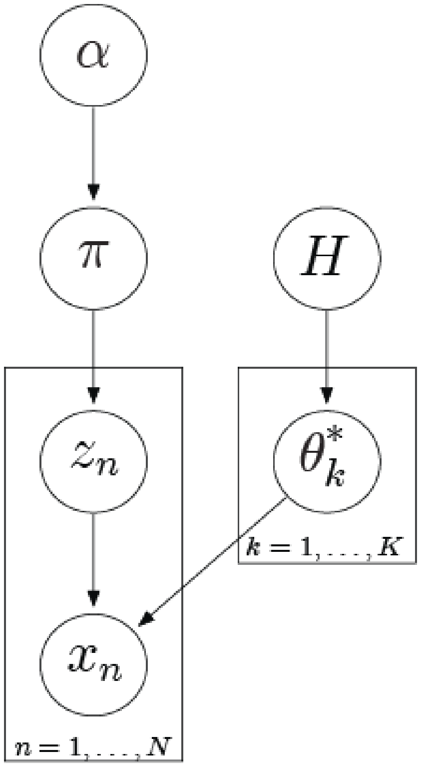
- Let's be Bayesian about mixture models
  - Place priors over our parameters
  - Again, introduce variable $\mathbf{z}_n$ as indicator which component data point $\mathbf{x}_n$ belongs to.

$$\begin{aligned} \mathbf{z}_n | \boldsymbol{\pi} &\sim \mathrm{Multinomial}(\boldsymbol{\pi}) \\ \mathbf{x}_n | \mathbf{z}_n = k, \boldsymbol{\mu}, \boldsymbol{\Sigma} &\sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k) \end{aligned}$$

  - Introduce conjugate priors over parameters

$$\begin{aligned} \boldsymbol{\pi} &\sim \mathrm{Dirichlet}(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}) \\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k &\sim H = \mathcal{N} - \mathcal{IW}(0, s, d, \phi) \end{aligned}$$

"Normal – Inverse Wishart"



Slide inspired by Yee Whye Teh

Recap: Bayesian Mixture Models

- Full Bayesian Treatment
  - Given a dataset, we are interested in the cluster assignments

$$p(\mathbf{Z}|\mathbf{X}) = \frac{p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})}{\sum_{\mathbf{Z}} p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})}$$

  where the likelihood is obtained by marginalizing over the parameters θ

$$\begin{aligned} p(\mathbf{X}|\mathbf{Z}) &= \int p(\mathbf{X}|\mathbf{Z},\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \\ &= \int \prod_{n=1}^{N}\prod_{k=1}^{K} p(\mathbf{x}_n|z_{nk},\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k|H)\mathrm{d}\boldsymbol{\theta} \end{aligned}$$

- The posterior over assignments is intractable!
  - Denominator requires summing over all possible partitions of the data into K groups!

$\Rightarrow$ Need efficient approximate inference methods to solve this...

Recap: Mixture Models with Dirichlet Priors

- Integrating out the mixing proportions π

$$p(\mathbf{z}|\alpha) = \int p(\mathbf{z}|\boldsymbol{\pi})p(\boldsymbol{\pi}|\alpha)\mathrm{d}\boldsymbol{\pi}$$
$$= \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} \prod_{k=1}^{K} \frac{\Gamma(N_k + \alpha/K)}{\Gamma(\alpha/K)}$$

- Conditional probabilities
  - Examine the conditional of $\mathbf{z}_n$ given all other variables $\mathbf{z}_{-n}$

$$p(z_{nk} = 1|\mathbf{z}_{-n}, \alpha) = \frac{p(z_{nk} = 1, \mathbf{z}_{-n}|\alpha)}{p(\mathbf{z}_{-n}|\alpha)}$$
$$= \frac{N_{-n,k} + \alpha/K}{N - 1 + \alpha} \qquad N_{-n,k} \stackrel{\text{def}}{=} \sum_{i=1, i\neq n}^{N} z_{ik}$$

$\Rightarrow$ The more populous a class is, the more likely it is to be joined!

Slide adapted from Zoubin Gharamani

Recap: Infinite Dirichlet Mixture Models

- Conditional probabilities: Finite K

$$p(z_{nk} = 1 | \mathbf{z}_{-n}, \alpha) = \frac{N_{-n,k} + \alpha/K}{N - 1 + \alpha}, \qquad N_{-n,k} \stackrel{\text{def}}{=} \sum_{i=1, i \neq n}^{N} z_{ik}$$

- Conditional probabilities: Infinite K
  - Taking the limit as $K \rightarrow \infty$ yields the conditionals

$$p(z_{nk} = 1 | \mathbf{z}_{-n}, \alpha) = \begin{cases} \frac{N_{-n,k}}{N-1+\alpha} & \text{if } k \text{ represented} \\ \frac{\alpha}{N-1+\alpha} & \text{if all } k \text{ not represented} \end{cases}$$

  - Left-over mass $\alpha \Rightarrow$ countably infinite number of indicator settings

Slide adapted from Zoubin Gharamani

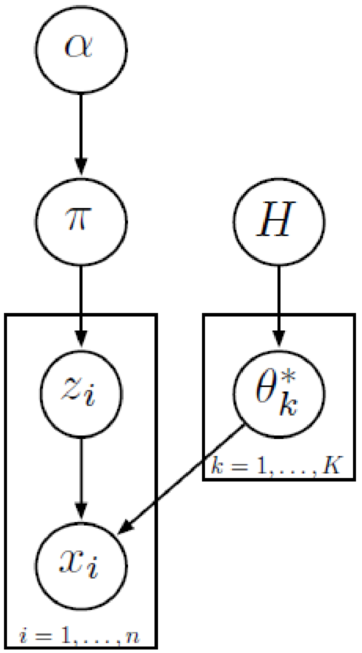
Recap: Gibbs Sampling for Finite Mixtures

- We need approximate inference here
  - Gibbs Sampling: Conditionals are simple to compute

$$p(\mathbf{z}_n = k|\text{others}) \propto \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\boldsymbol{\pi} \mid \mathbf{z} \sim \mathrm{Dir}(N_1 + \alpha/K, \ldots, N_K + \alpha/K)$$

$$\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k|\text{others} \sim \mathcal{N} - \mathcal{IW}(v', s', d', \phi')$$



- However, this will be rather inefficient…
  - In each iteration, algorithm can only change the assignment for individual data points.
  - There are often groups of data points that are associated with high probability to the same component. $\Rightarrow$ Unlikely that group is moved.
  - Better performance by collapsed Gibbs sampling which integrates out the parameters π, μ, $\boldsymbol{\Sigma}$.

Slide adapted from Yee Whye Teh

Image source: Yee Whye Teh

Recap: Collapsed Finite Bayesian Mixture

- More efficient algorithm
  - Conjugate priors allow analytic integration of some parameters
  - Resulting sampler operates on reduced space of cluster assignments (implicitly considers all possible cluster shapes)

- Procedure
  - The model implies the factorization

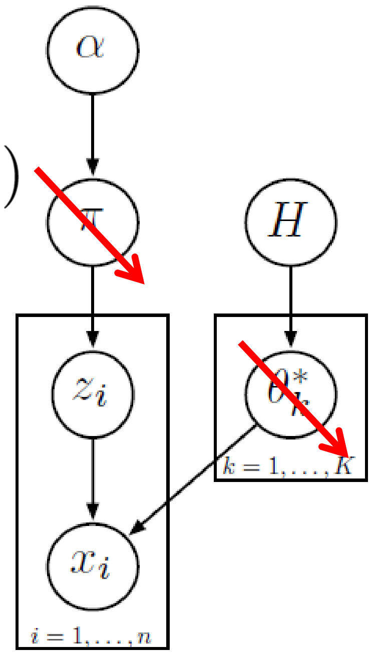
$$p(\mathbf{z}_n|\mathbf{z}_{-n}, \mathbf{x}, \alpha, H) \propto p(\mathbf{z}_n|\mathbf{z}_{-n}, \alpha)p(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_{-n}, H)$$

  - Derive

$$p(\mathbf{z}|\alpha) = \int p(\mathbf{z}|\boldsymbol{\pi})p(\boldsymbol{\pi}|\alpha)\mathrm{d}\boldsymbol{\pi}$$

$$p(\mathbf{x}_n|\mathbf{z}_n, H) = \int \sum_{k=1}^{K} z_{nk} p(\mathbf{x}_n|\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k|H)\mathrm{d}\boldsymbol{\theta}$$

$\Rightarrow$ Conjugate prior, Normal - Inverse Wishart



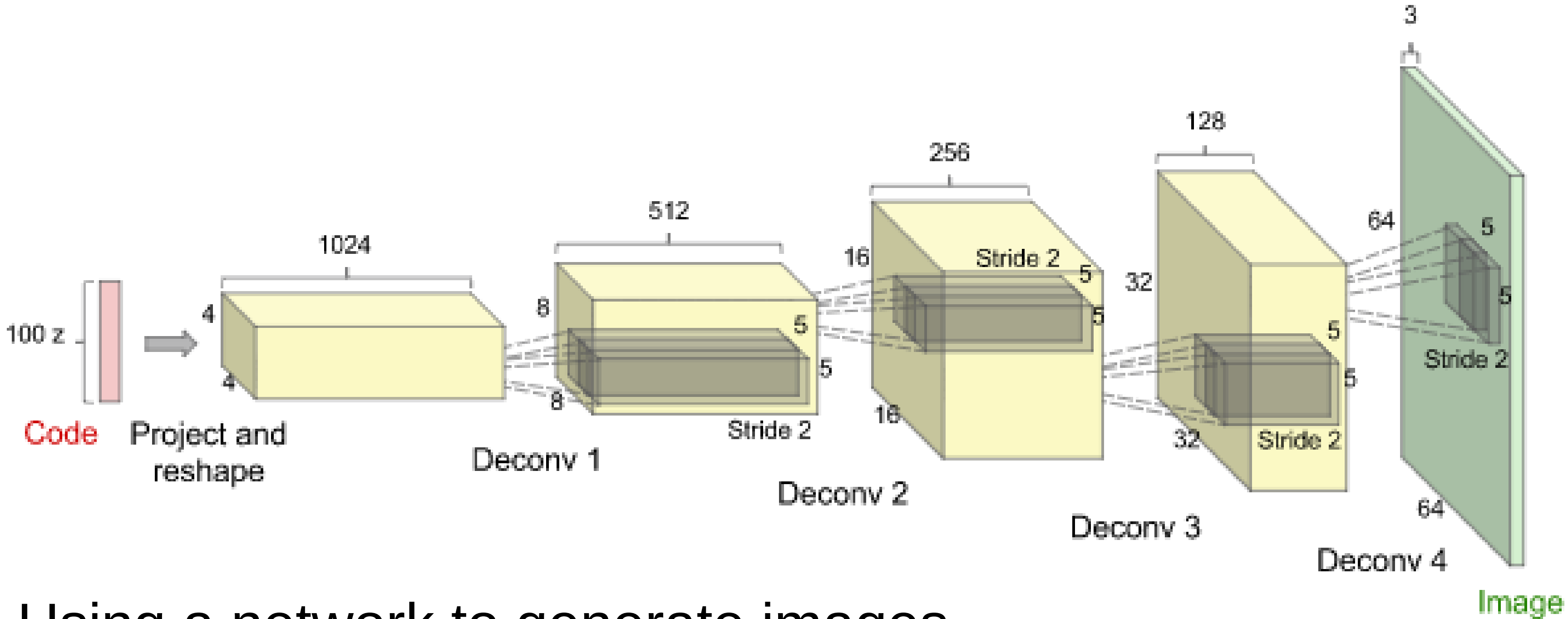
Slide adapted from Erik Sudderth

Image source: Yee Whye Teh

Topics of This Lecture

- Recap: Bayesian Mixture Models
- Generative Adversarial Networks (GANs)
  - Generative networks
  - GAN loss and training procedure
- Applications & Extensions
  - GANs for image generation
  - GANs for superresolution
  - Conditional GANs
- Problems of GANs
  - Problems during training
  - Conceptual problems
  - Extension: Wasserstein GANs

Generative Networks

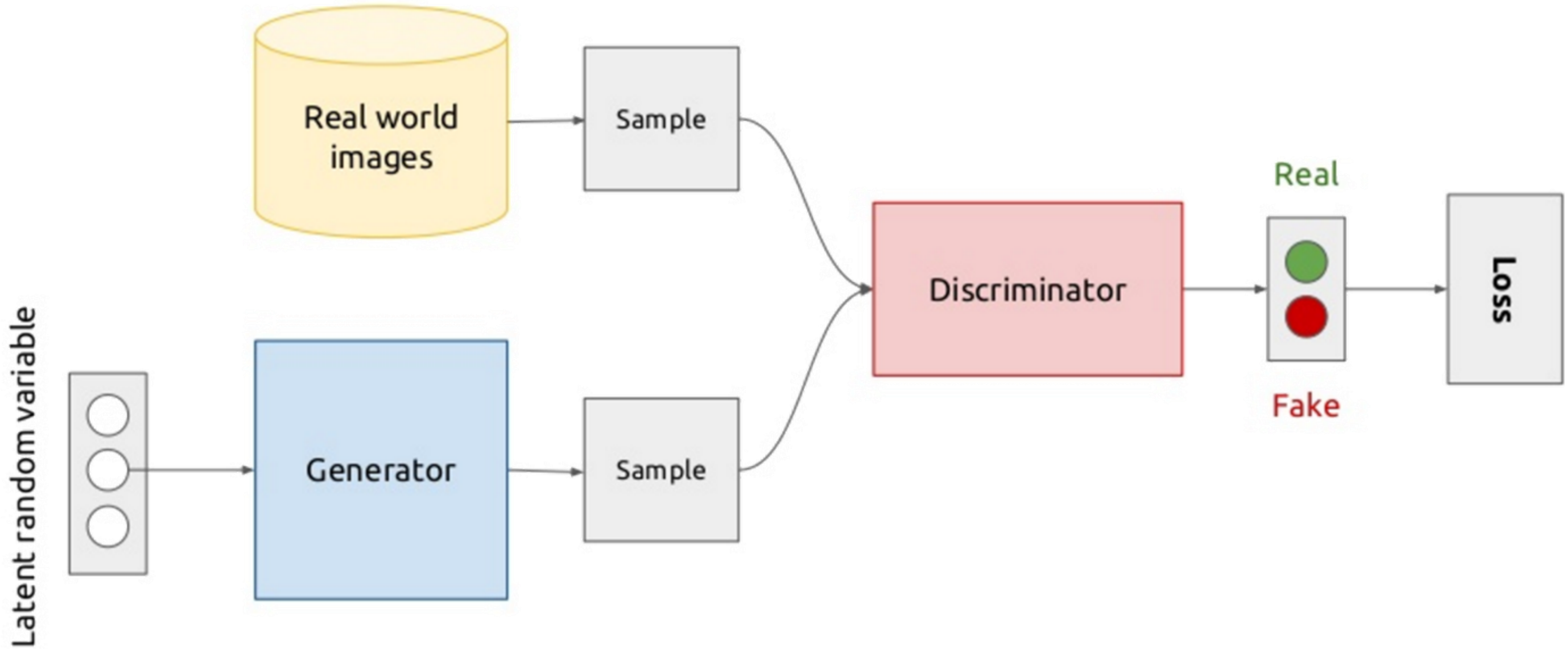


- Using a network to generate images
  - Sampling from noise distribution
  - Sequence of upsampling layers to generate an output image
  - *How can we train such a model to produce the desired output?*

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

Generative Adversarial Networks (GANs)

- Conceptual view

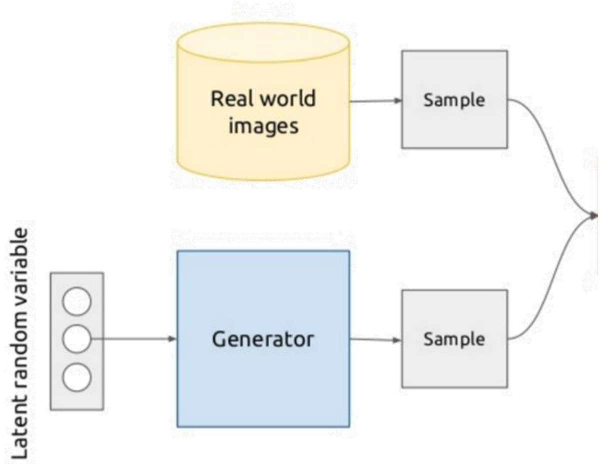


- Main idea
  - Simultaneously train an image generator G and a discriminator D.
  - Interpreted as a two-player game

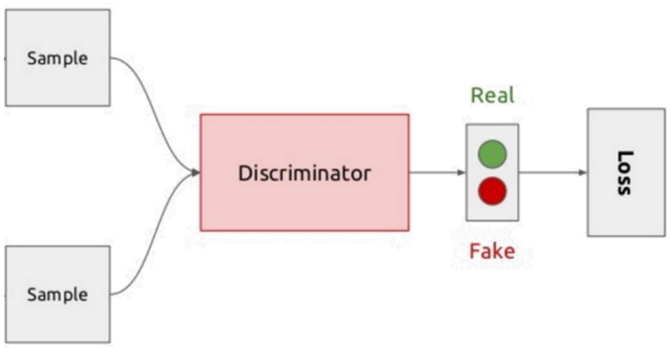
Image credit: Kevin McGuiness

Two-Player Game

- Generator
  – Tries to draw samples from $p(\boldsymbol{x})$.
  – Analogy: *counterfeiter*



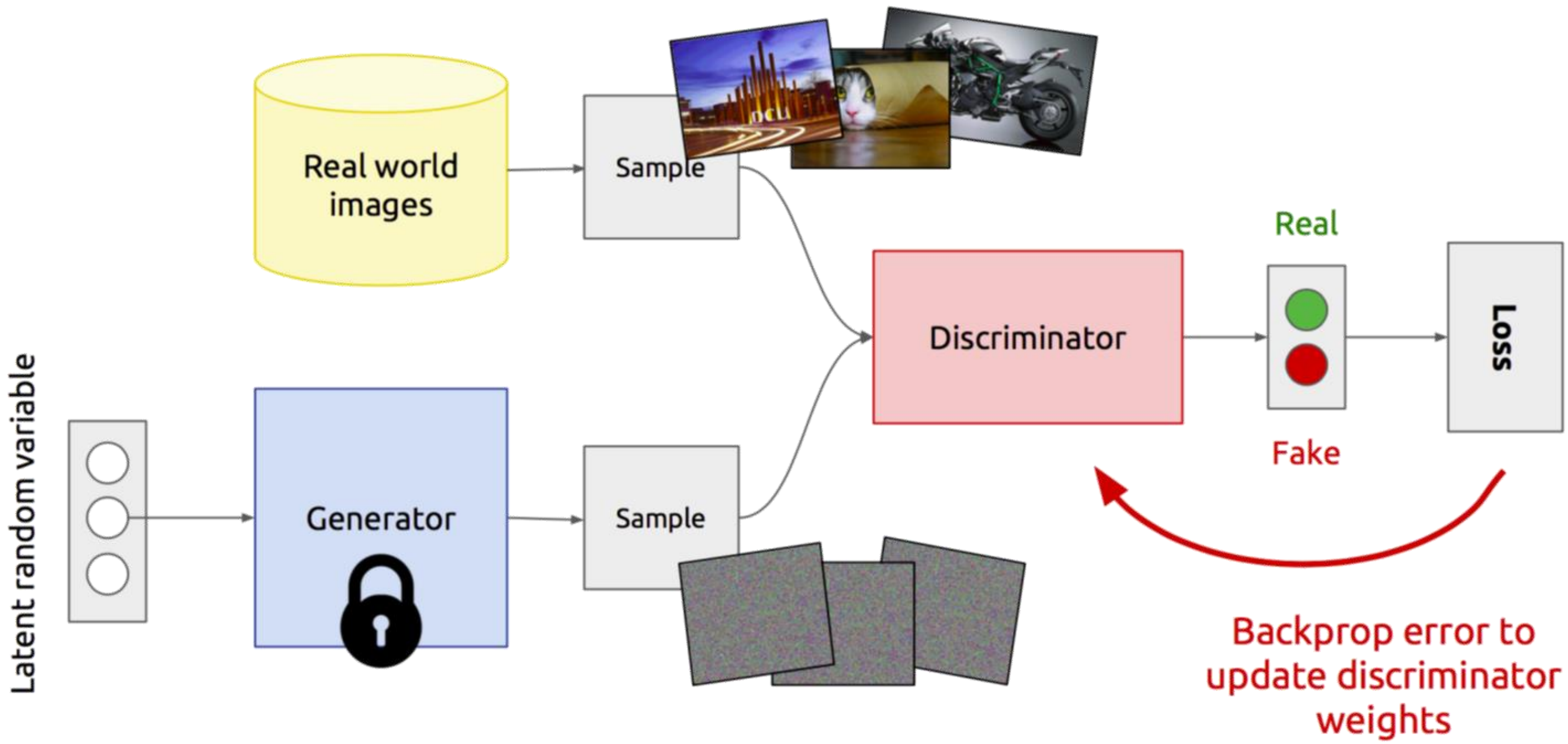
- Discriminator
  – Tries to determine whether the sample came from the generator or the data distribution.
  – Analogy: *police investigator*



- Both generator and discriminator are deep networks
  – We can train them with backprop.

Image sources: www.bundesbank.de, weclipart.com, Kevin McGuiness

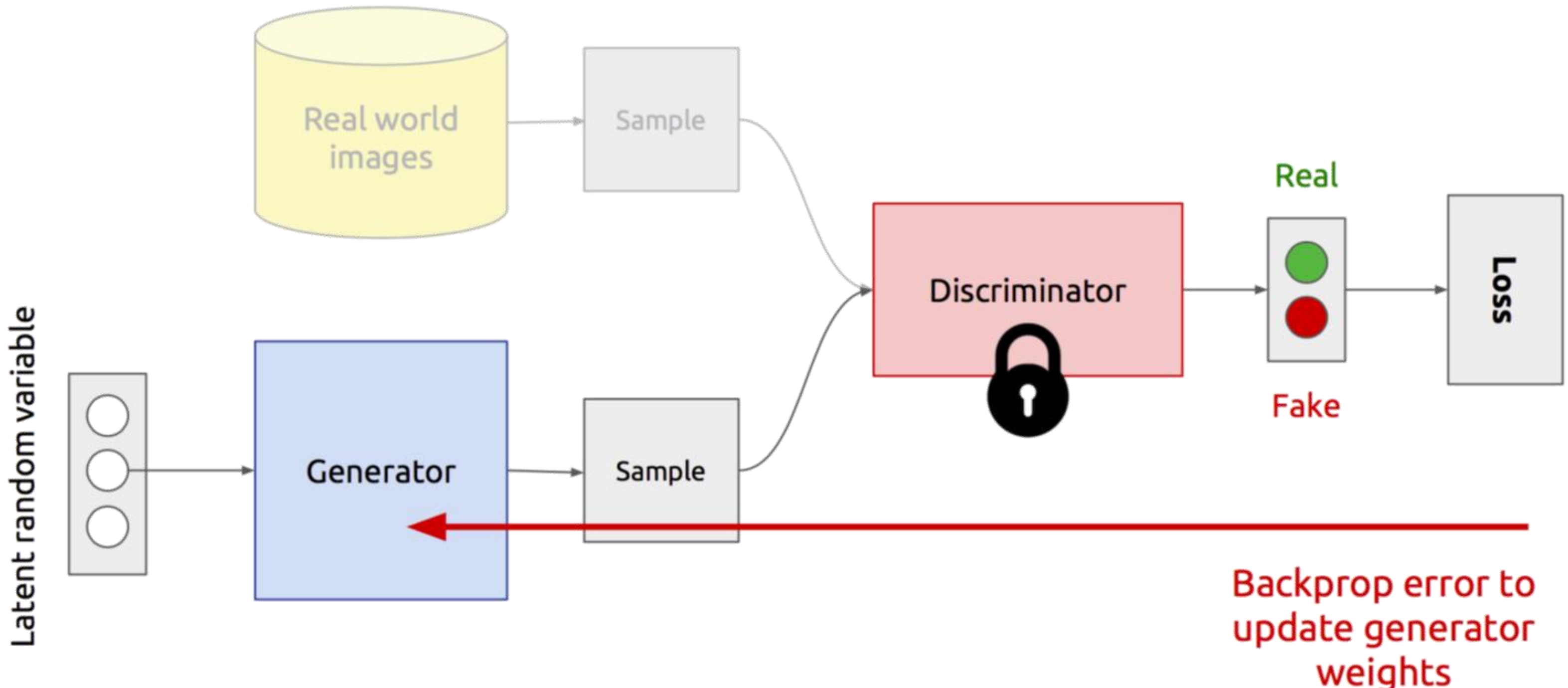
Training the Discriminator



- Procedure
  - Fix generator weights
  - Train discriminator to distinguish between real and generated images

Image credit: Kevin McGuiness

Training the Generator



- Procedure
  - Fix discriminator weights
  - Sample from generator
  - Backprop through discriminator to update generator weights

Image credit: Kevin McGuiness

Formalizing This Procedure

- This corresponds to a two-player minimax game:

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

- Explanation
  - Train D to maximize the probability of assigning the correct label to both training examples and samples from G.
  - Simultaneously train G to minimize $\log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$.

- The Nash equilibrium of this game is achieved at
  - $p_g(\boldsymbol{x}) = p_{data}(\boldsymbol{x}) \quad \forall \boldsymbol{x}$
  - $D(\boldsymbol{x}) = \frac{1}{2} \quad \forall \boldsymbol{x}$

GAN Algorithm

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

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

Discriminator updates

end for

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

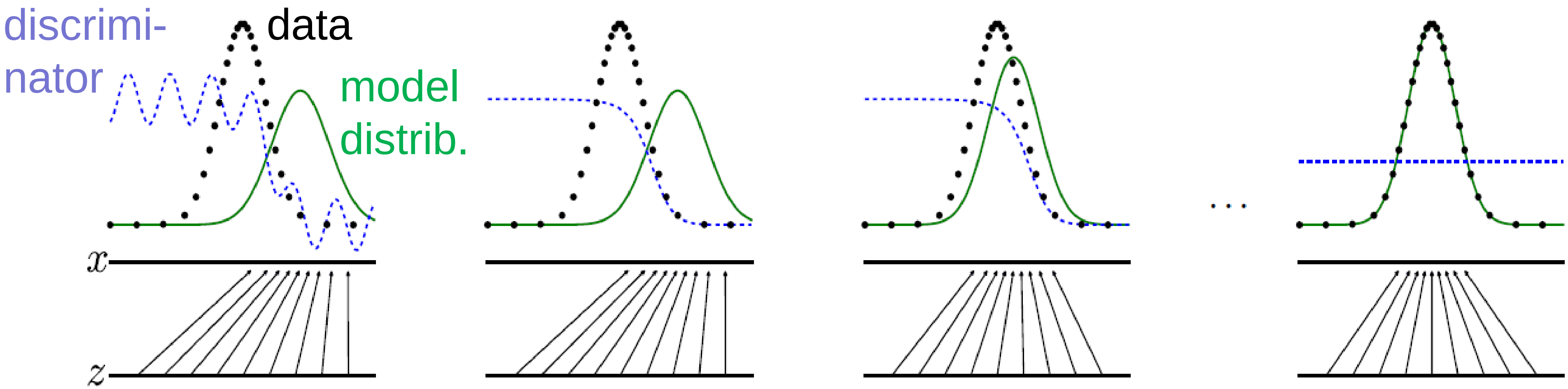
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

Generator updates

end for

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

Intuition

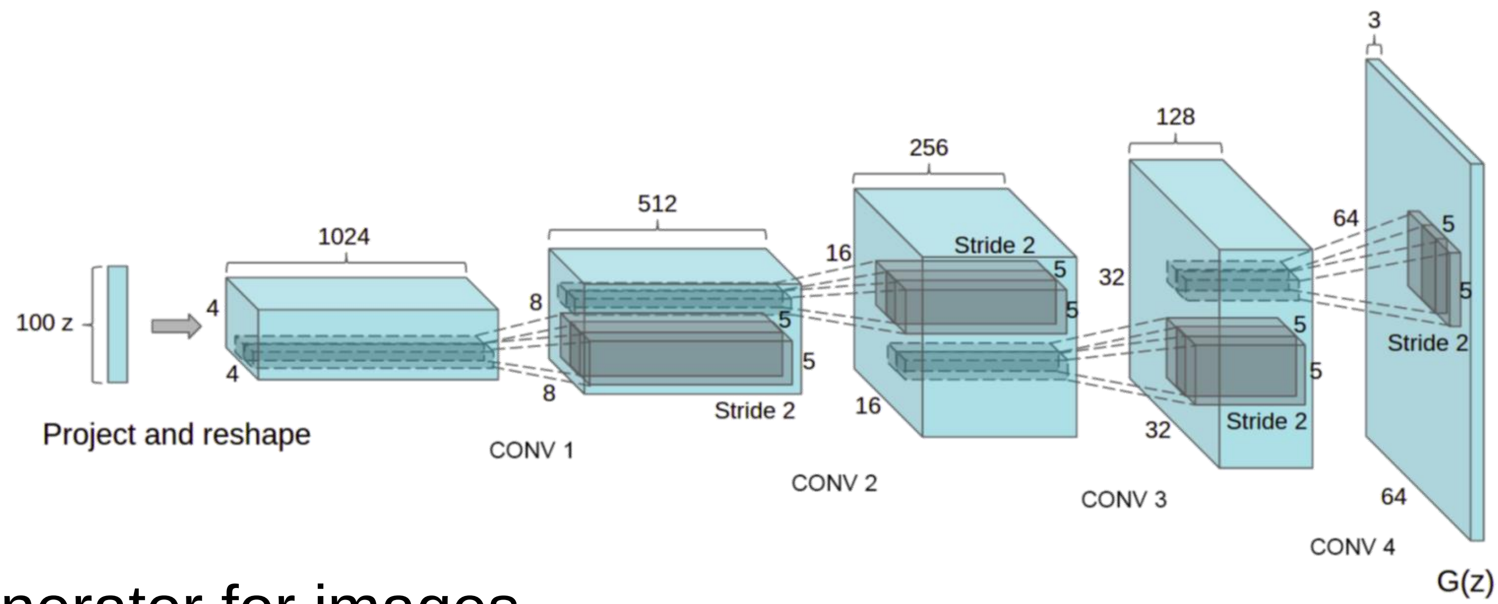


- Behavior near convergence
  - In the inner loop, D is trained to discriminate samples from data.
  - Gradient of D guides G to flow to regions that are more likely to be classified as data.
  - After several steps of training, G and D will reach a point at which they cannot further improve, because $p_g = p_{data}$.
  - Now, the discriminator is unable to differentiate between the two distributions, i.e., $D(\boldsymbol{x}) = 0.5$.

Topics of This Lecture

- Recap: Bayesian Mixture Models
- Generative Adversarial Networks (GANs)
  – Generative networks
  – GAN loss and training procedure
- Applications & Extensions
  – GANs for image generation
  – GANs for superresolution
  – Conditional GANs
- Problems of GANs
  – Problems during training
  – Conceptual problems
  – Extension: Wasserstein GANs

Example: Deep Convolutional GAN (DCGAN)



- Generator for images
  - Remove fully-connected layers
  - Upsampling with fractional strided convolutions
  - Batch normalization after each layer (important!)
  - Use ReLu in generator for hidden layers, tanh for output layer
  - Use Leaky ReLu in the discriminator for all layers

Image credit: Alec Radford

Example Application: Image Generation

- Generating bedroom images
  - Each sample is generated from a sampled random number

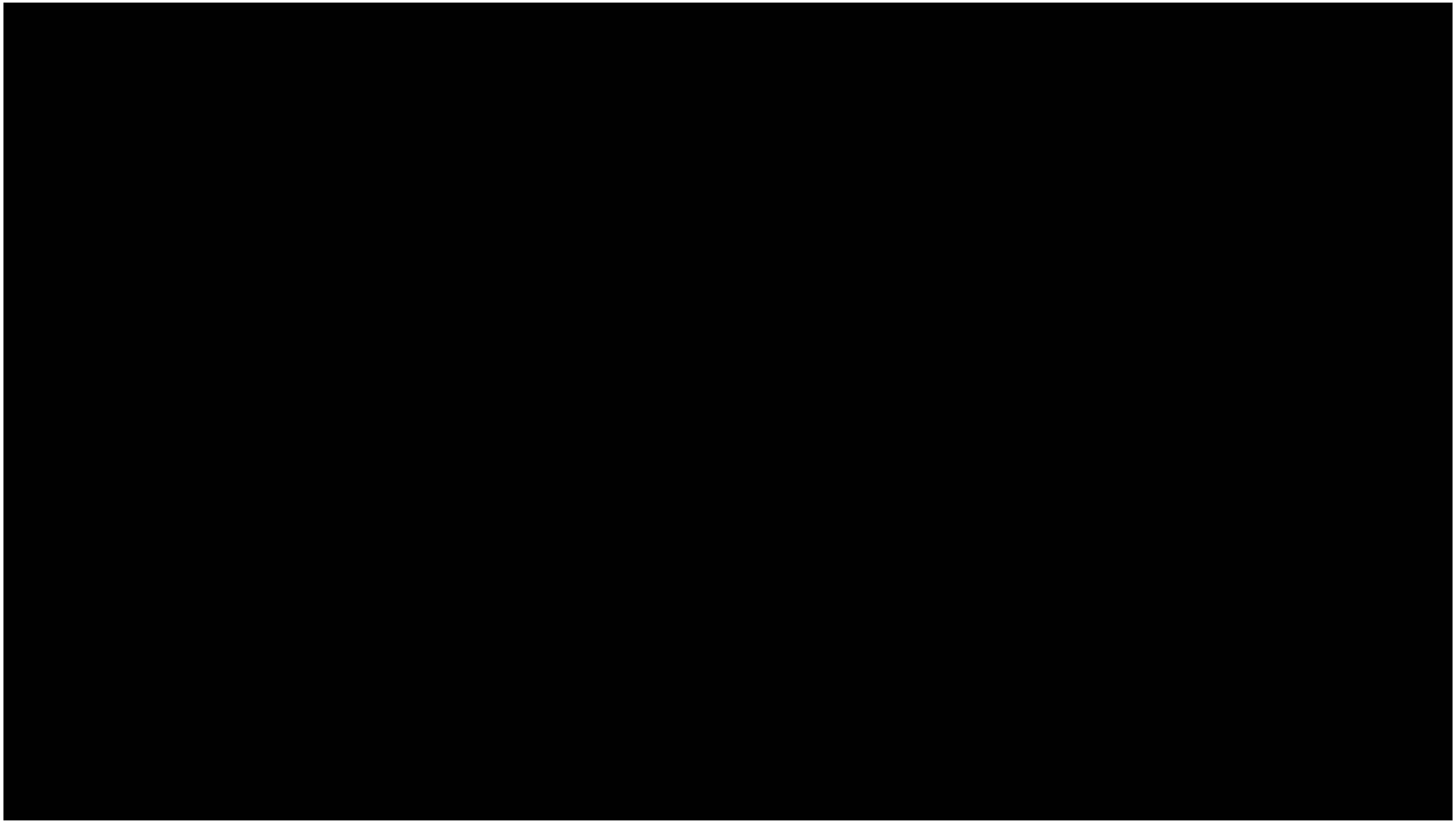
Visual Computing Institute | RWTH AACHEN UNIVERSITY

Image credit: Alec Radford et al.

Example Application: Image Generation

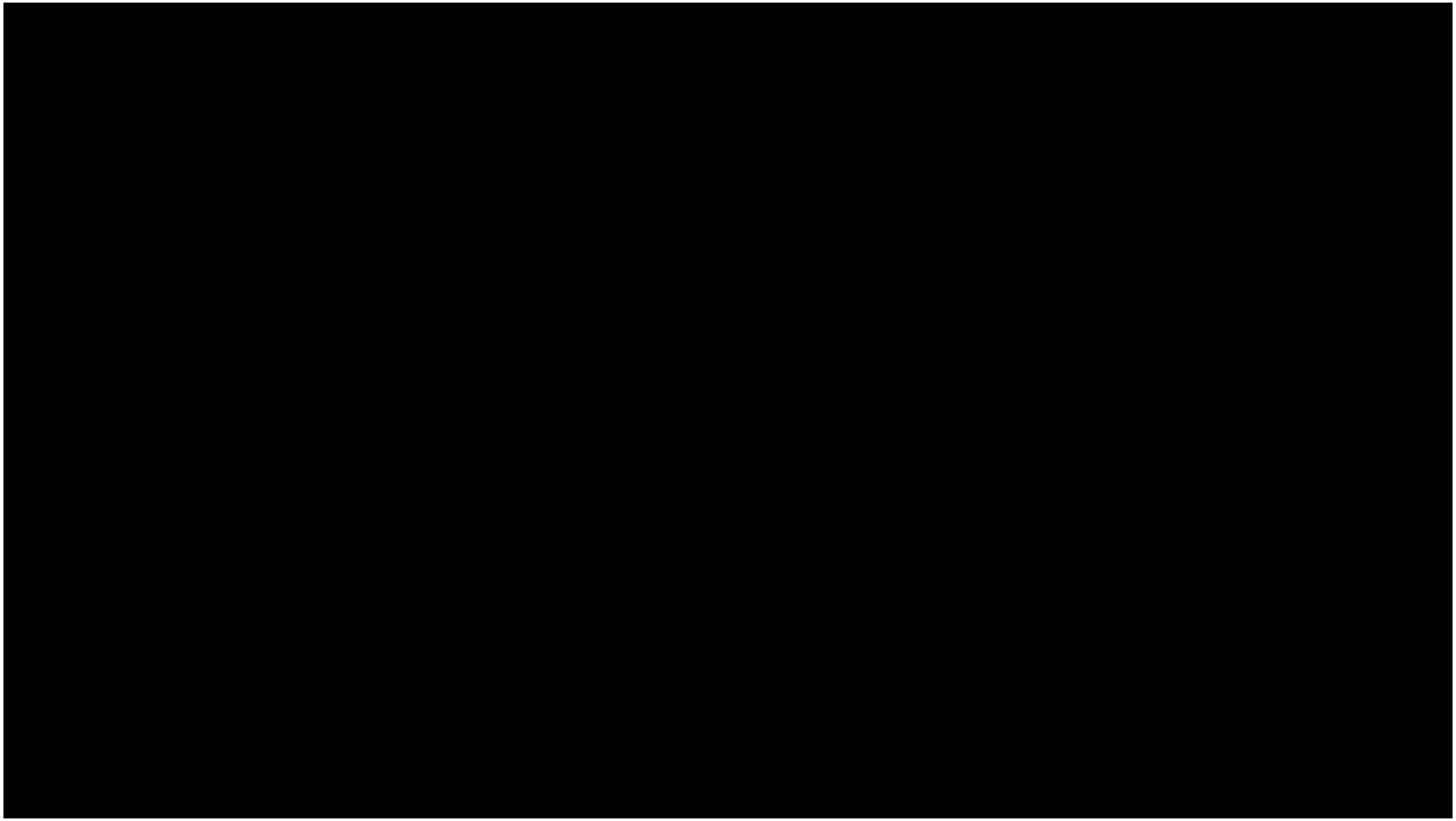
- Interpolating between the random points in latent space…

Interpolation between Face Images



Karras et al, “*Progressive growing of GANs for improved quality, stability, and variation*”, ICLR’18

Interpolation between Arbitrary Images



Karras et al, "*Progressive growing of GANs for improved quality, stability, and variation*", ICLR'18

Example Application: Super-Resolution (SRGAN)

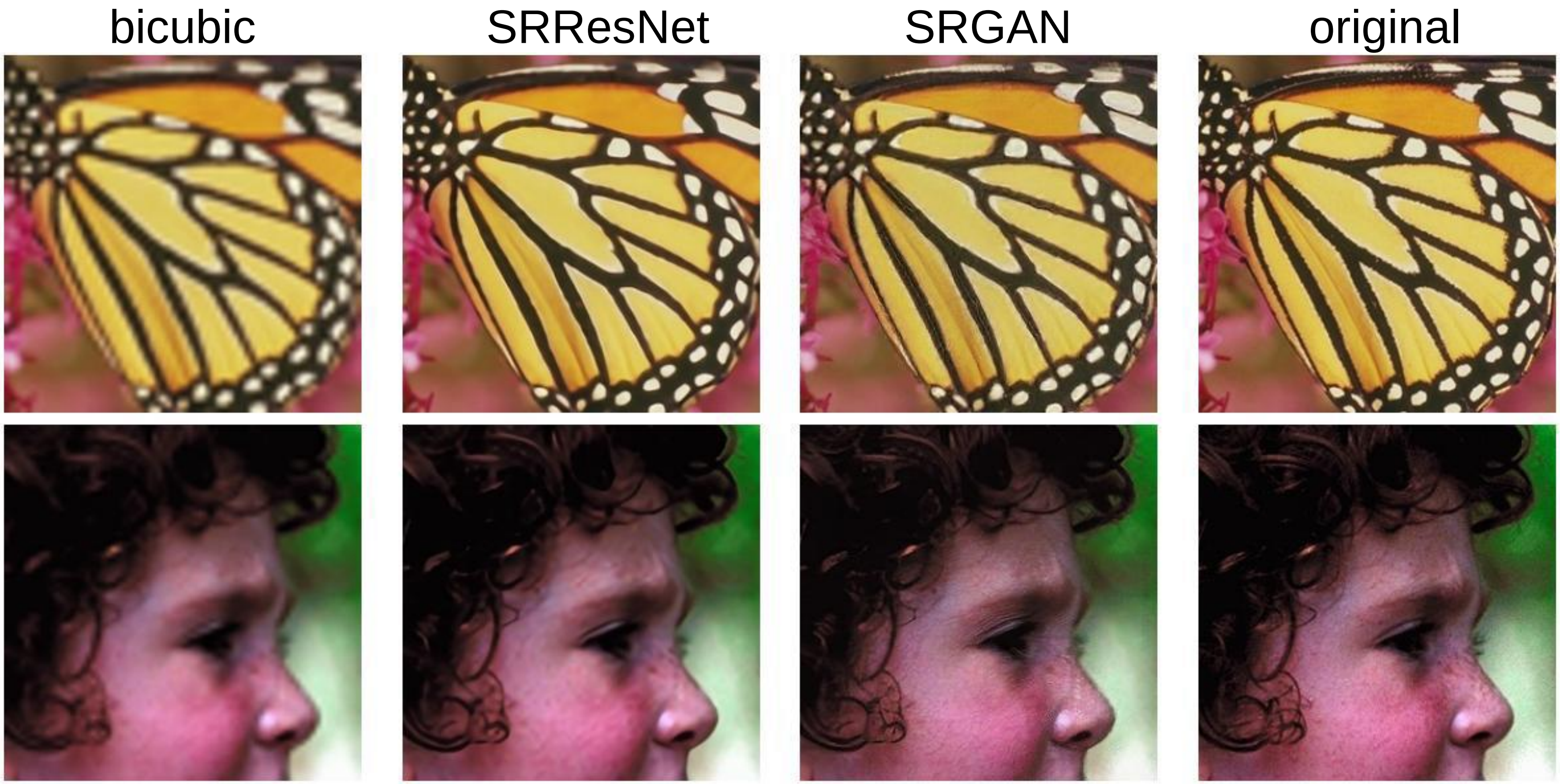
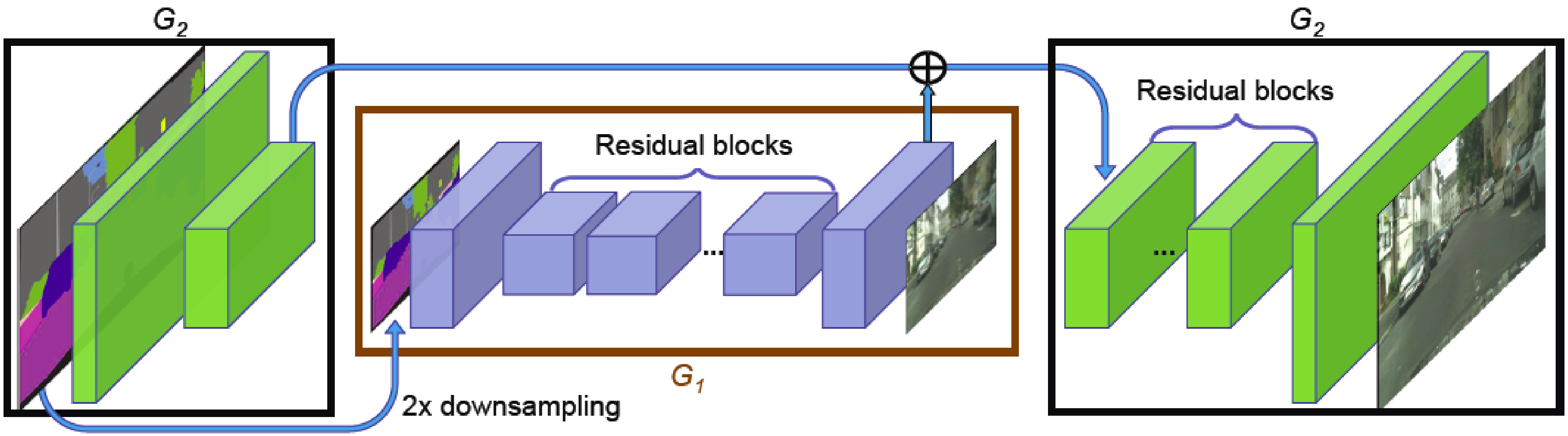


Image credit C. Ledig et al.

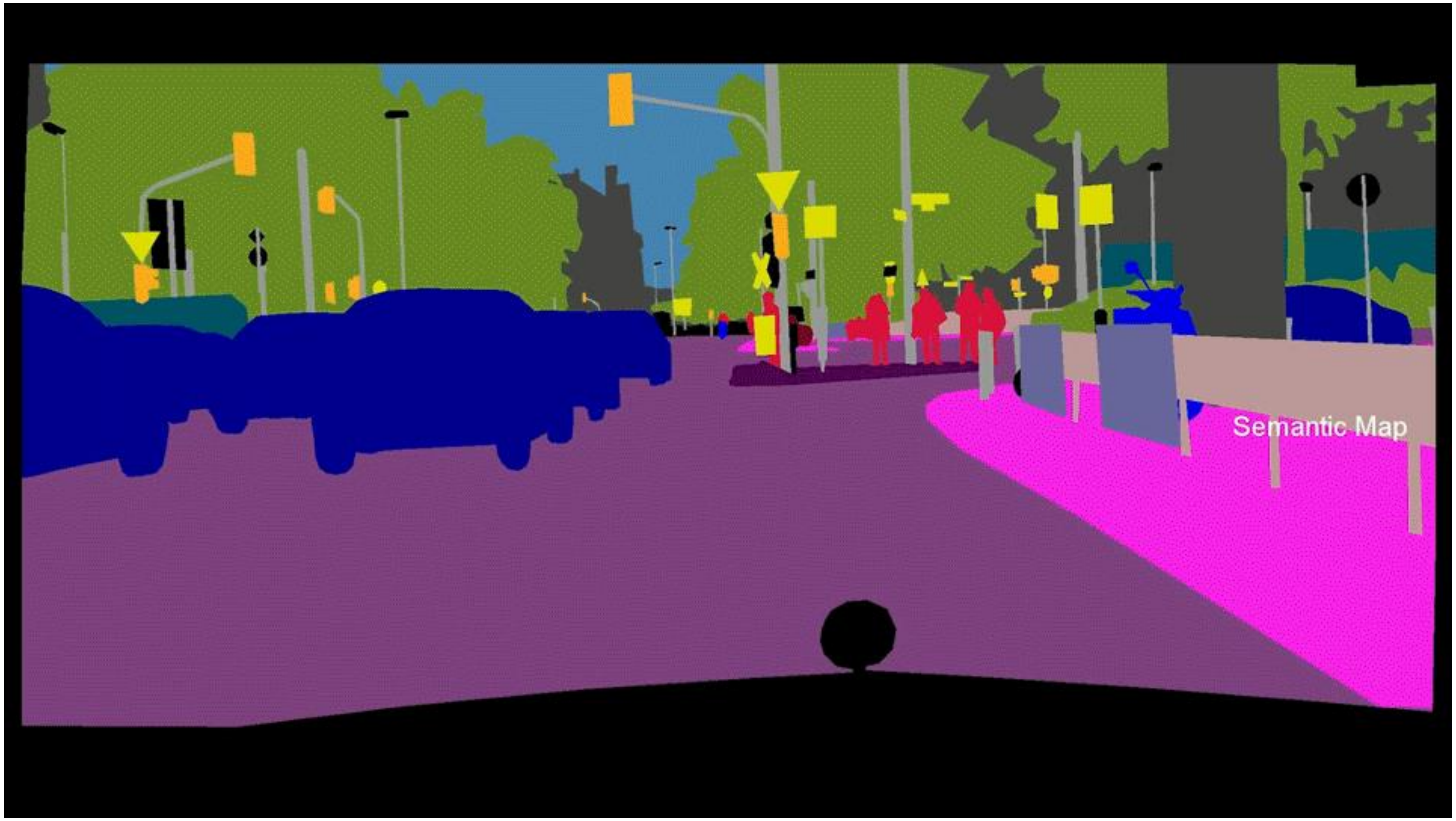
Extension: Conditional GANs



- Idea
  - Condition the latent space representation on an input image
  - Used to create the pix2pix network

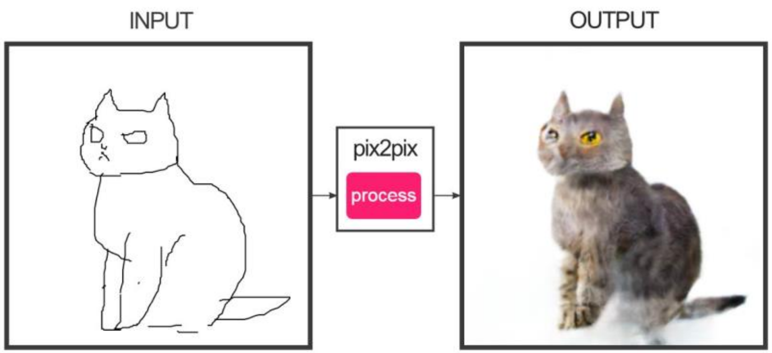
P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. CVPR 2017.

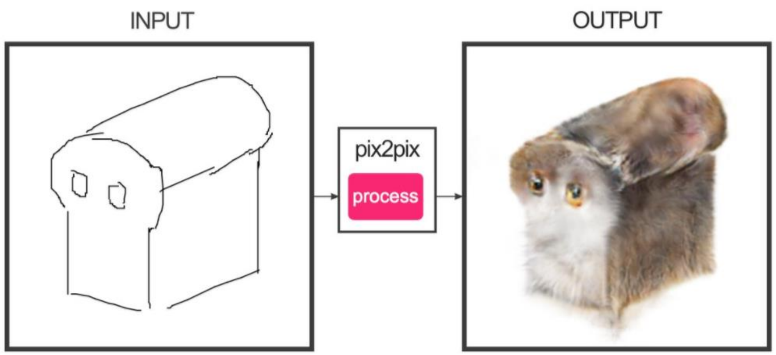
Extension: Conditional GANs



T-C. Wang, M-Y. Liu, J-Y. Zhu, A. Tao, J. Kautz, B. Catanzaro, High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, CVPR 2018.

Artist Project: edges2cats [Christopher Hesse]





Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

@ka92

Slide credit: Aljosa Efros

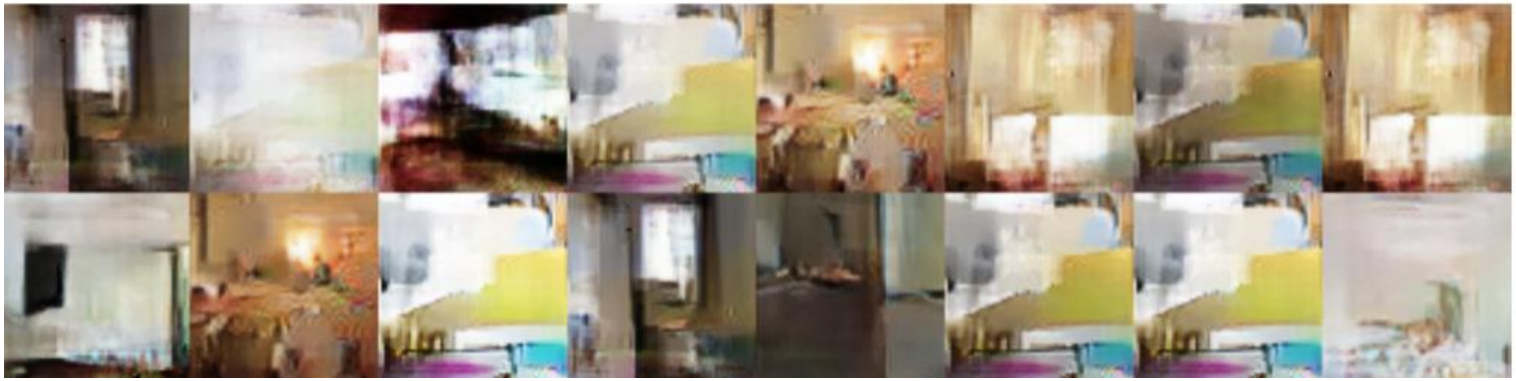
Topics of This Lecture

- Recap: Bayesian Mixture Models
- Generative Adversarial Networks (GANs)
  - Generative networks
  - GAN loss and training procedure
- Applications & Extensions
  - GANs for image generation
  - GANs for superresolution
  - Conditional GANs
- Problems of GANs
  - Problems during training
  - Conceptual problems
  - Extension: Wasserstein GANs

What Can Possibly Go Wrong? Problems with GANs

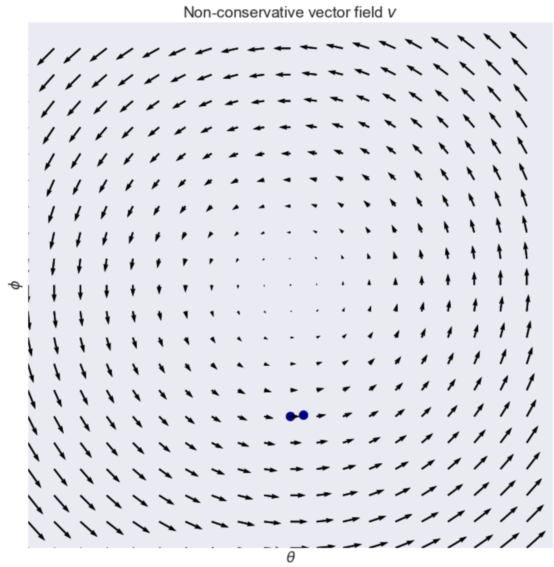
- Problem 1: Vanishing gradients
  - When the discriminator is perfect, the loss function falls to zero.
  - No gradient to update the loss during learning iterations.
  - Dilemma: Walking a fine line…
    - Discriminator behaves badly $\Rightarrow$ generator does not have accurate feedback
    - Discriminator does a great job $\Rightarrow$ gradient of the loss drops close to zero

What Can Possibly Go Wrong? Problems with GANs



- Problem 2: Mode collapse
  - Even though the generator might be able to trick the discriminator, it may fail to represent the complex real-world data distribution.
  - Training gets stuck in a small space with little variety

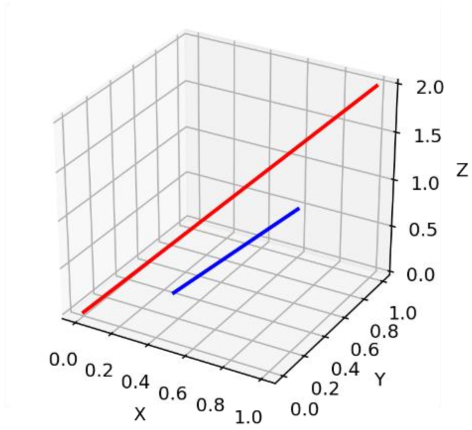
What Can Possibly Go Wrong? Problems with GANs



- Problem 3: Non-convergence
  - GANs involve two players
  - Each model updates its cost independently
  - This means we are performing simultaneous gradient descent
  - Problem: might not converge to Nash equilibrium

Image source: Ferenc Huszar, Lilian Weng

What Can Possibly Go Wrong? Problems with GANs



- Problem 4: Low-dimensional support
  - Both p_{data} and p_g lie on low-dimensional manifolds.
  - Those manifolds most likely do not intersect
  - The Jensen divergence implicitly optimized in GANs cannot deal with this well.
  - Wasserstein GANs fix this by introducing a different loss function.

Image source: Ferenc Huszar, Lilian Weng

Summary

- Advantages
  - GANs can be trained with backpropagation
  - Generated images are sharper than from VAEs
  - Robust to overfitting, since generator never sees the training data
  - Fast process: single forward pass generates a single sample
- Disadvantages
  - GANs are well known for being delicate and unstable
  - Problems with non-convergence
  - Problems with mode collapse
- Extensions
  - Wasserstein GANs fix several major problems in the GAN formulation
  - Energy-based GANs allow general loss functions

2017: Explosion of GANs

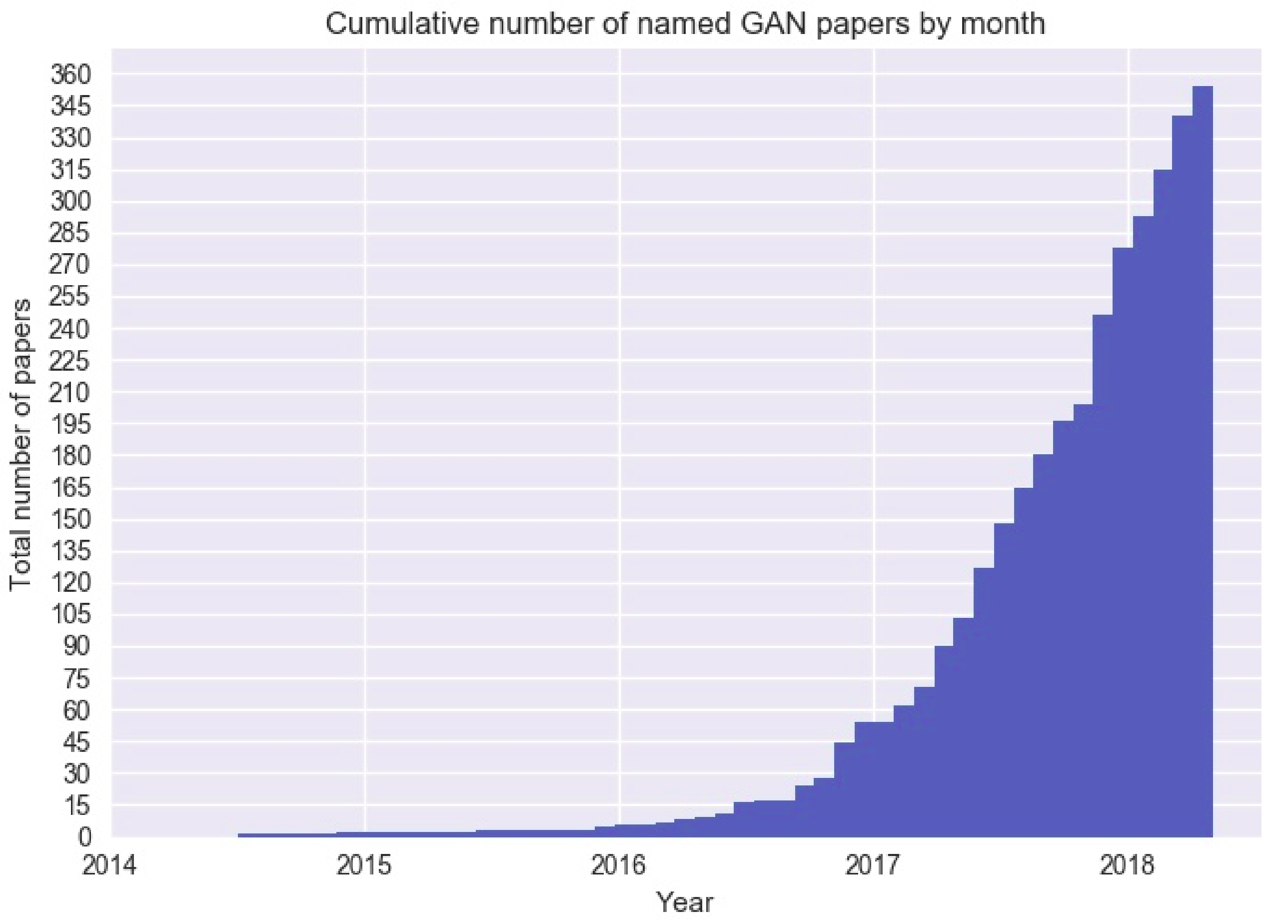
- “The GAN Zoo”

- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

https://github.com/hindupuravinash/the-gan-zoo

Slide credit: FeiFei Li

Interest in GANs Is Still Growing…



https://github.com/hindupuravinash/the-gan-zoo

References

- Original GAN
  - I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Nets, NIPS, 2014.
  - A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv 1511.06434, 2015.
- GAN Extensions
  - J. Zhao, M. Mathieu, Y. LeCun, Energy-based Generative Adversarial Networks, arXiv 1506.03365, 2015.
  - M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, arXiv 1701.07875, 2017.
- Evaluations
  - M. Lucic, K. Kurach, M. Michalski, S. Gelly, O. Bousquet, Are GANs Created Equal? A Large-Scale Study, arXiv 1711.10337, 2017.