Generative Adversarial Networks

Du Ang

April 16, 2018

VISION@OUC

Overview

- Supervised Learning vs Unsupervised Learning
- Introduction to generative models
- Popular generative models in detail
  - Variational Autoencoders (VAE)
  - Generative Adversarial Networks (GAN)
- GAN research frontiers

Supervised Learning

vs

Unsupervised Learning

Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.

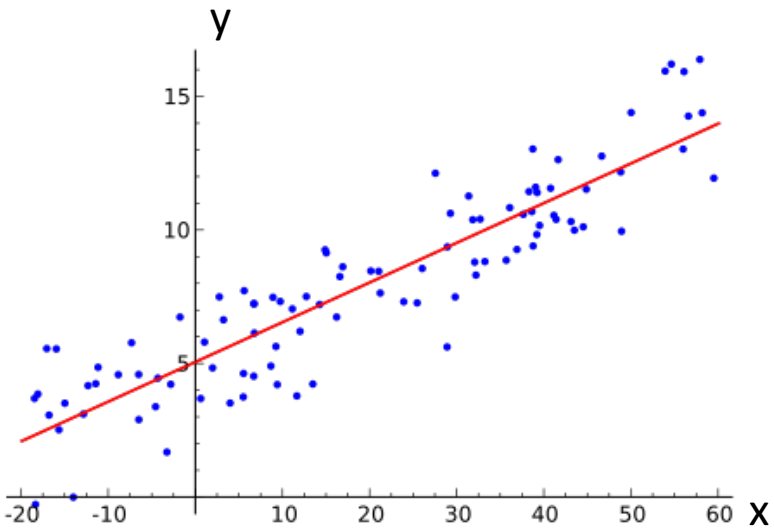
Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.



Linear Regression

Image from https://upload.wikimedia.org/wikipedia/commons/thumb/3/3a/Linear_regression.svg/438px-Linear_regression.svg.png

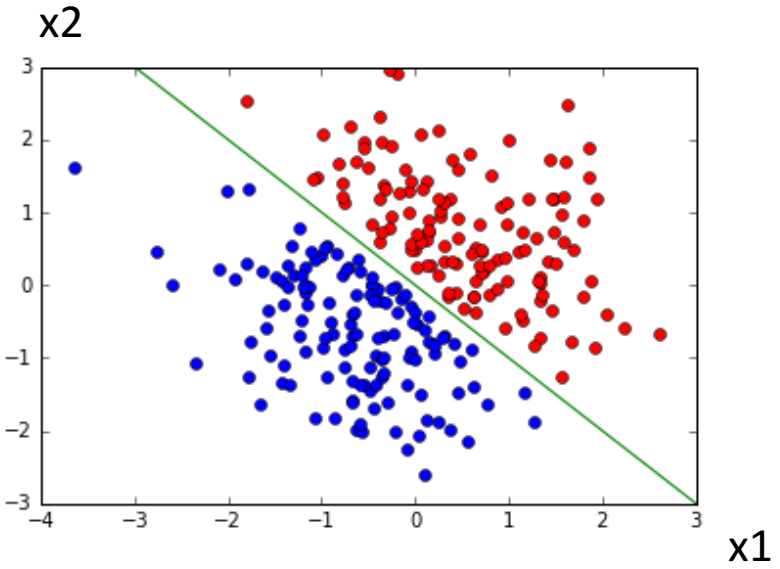
Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.



Logistic Regression

Image from https://images2015.cnblogs.com/blog/802399/201606/802399-20160613142313838-1220773185.png

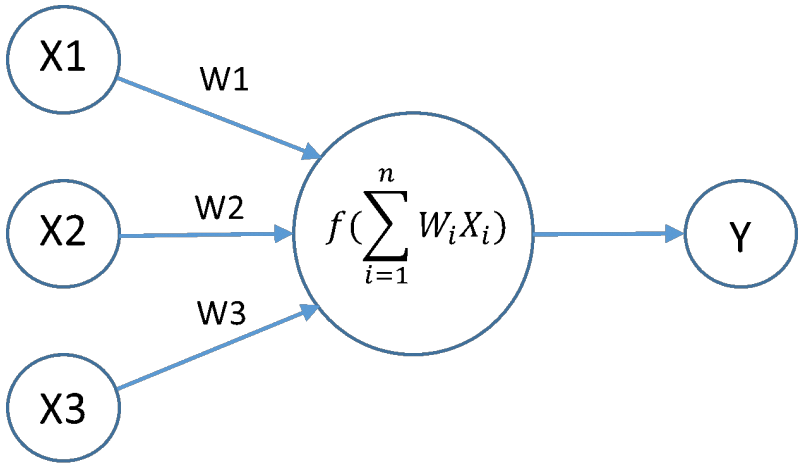
Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.



Logistic Regression

Image from https://i.stack.imgur.com/gzrsx.png

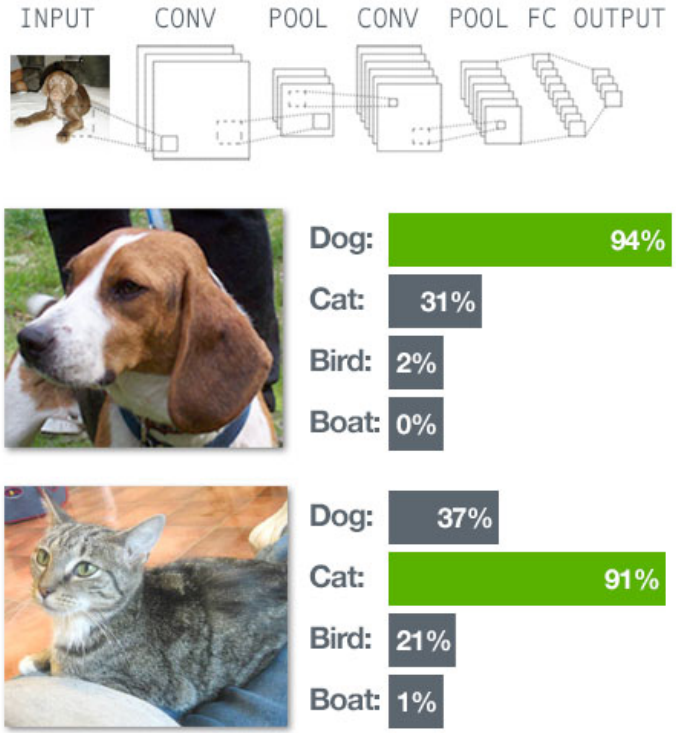
Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.



Classification

Image from https://www.pyimagesearch.com/wp-content/uploads/2016/05/deep_learning_example.jpg

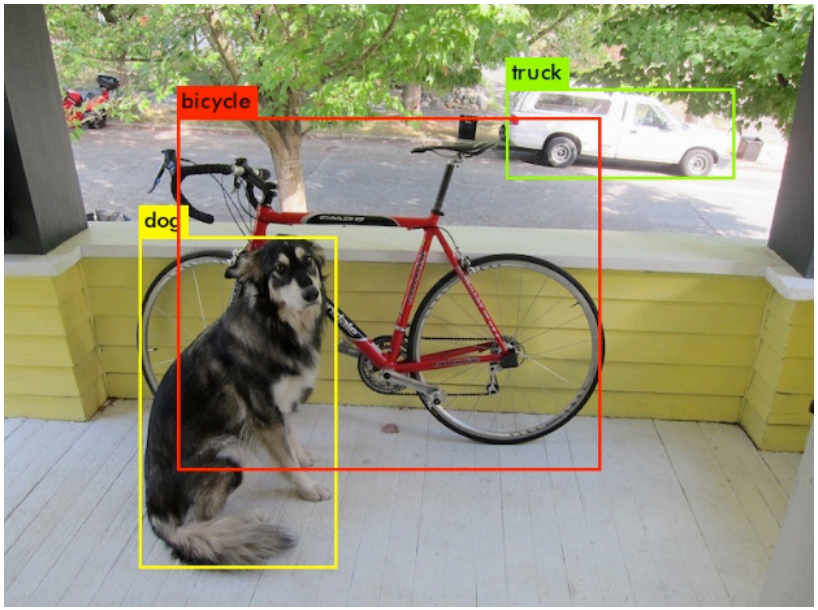
Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.



<object-class> <x> <y> <width> <height>

Object Detection

Image from https://pjreddie.com/darknet/yolo/

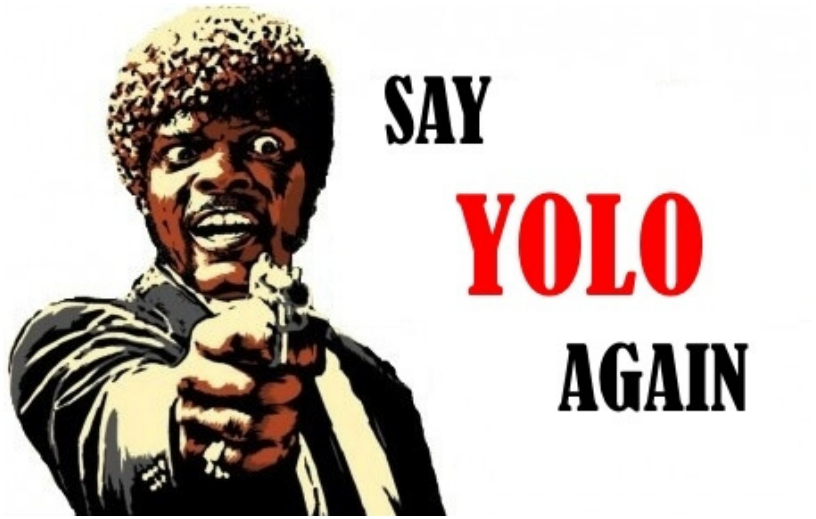
Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.



YOLO: Real-Time Object Detection

YOLOv3 is released!

Object Detection

Image from https://pjreddie.com/darknet/yolo/

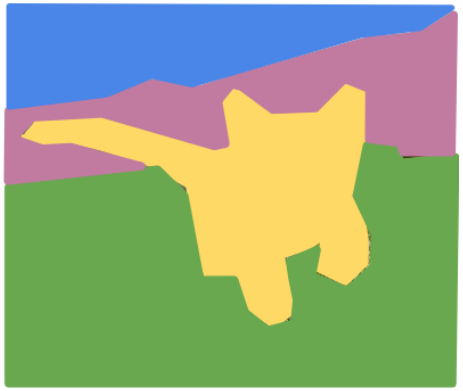
Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.



GRASS, CAT,
TREE, SKY

Semantic Segmentation

Image from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

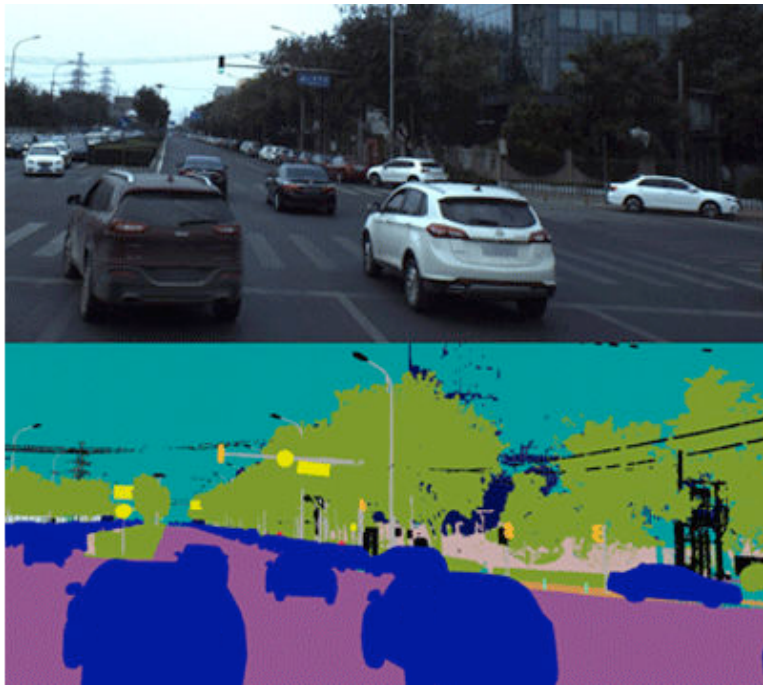
Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.



Semantic Segmentation

Image from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.

a larger jetliner sitting on top of an airport tarmac

Image Captioning

Image from https://cs.stanford.edu/people/karpathy/neuraltalk2/demo.html

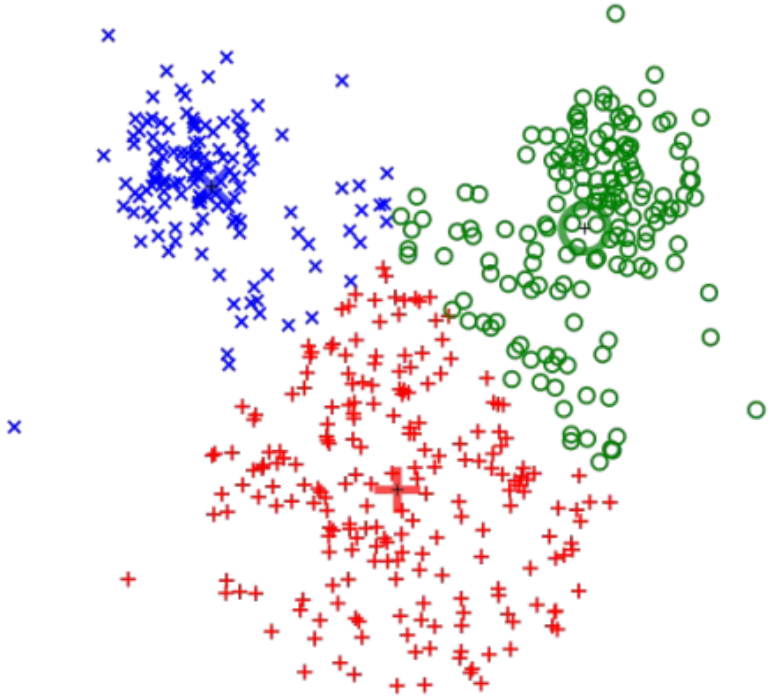
Supervised Learning vs Unsupervised Learning

Unsupervised Learning

Data: x
Just data, no labels!

Goal:
Learn some underlying hidden *structure* of the data

Examples: Clustering, dimension reduction, feature learning, density estimation, etc.



K-means clustering

Image from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Supervised Learning vs Unsupervised Learning

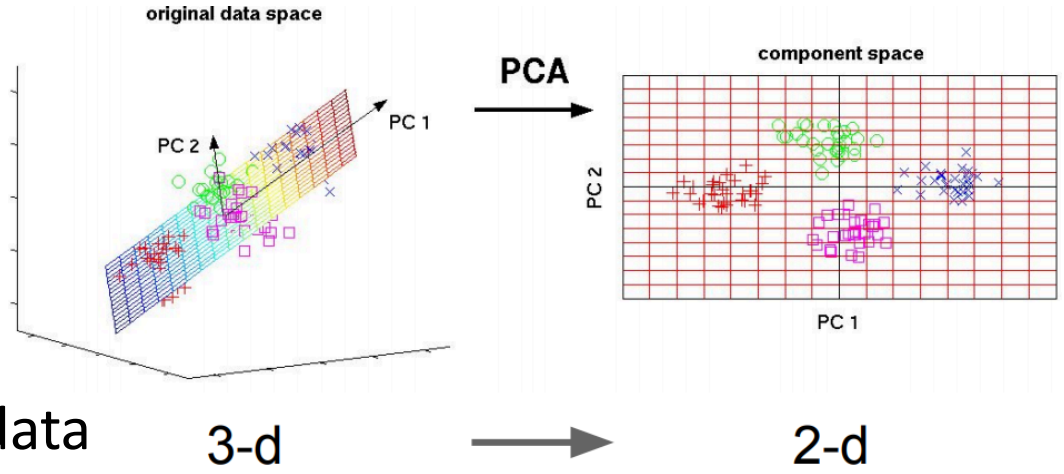
Unsupervised Learning

Data: x

Just data, no labels!

Goal:

Learn some underlying hidden *structure* of the data



Examples: Clustering, dimension reduction, feature learning, density estimation, etc.

minimize $\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - x_{approx}^{(i)}\right\|^2$

Principal Component Analysis (Dimensionality reduction)

Image from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Supervised Learning vs Unsupervised Learning

Unsupervised Learning

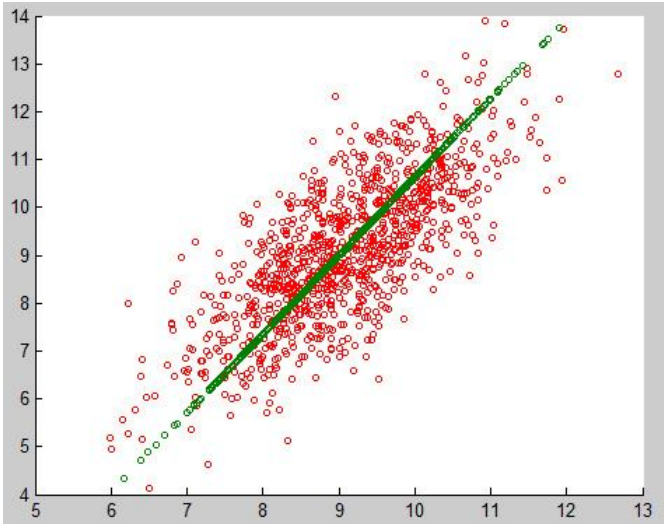
Data: x

Just data, no labels!

Goal:

Learn some underlying hidden *structure* of the data

Examples: Clustering, dimension reduction, feature learning, density estimation, etc.



2-d ⟶ 1-d

minimize $\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - x_{approx}^{(i)}\right\|^2$

Principal Component Analysis (Dimensionality reduction)

Image from https://wenku.baidu.com/view/d7db330e443610661ed9ad51f01dc281e43a5648.html

Supervised Learning vs Unsupervised Learning

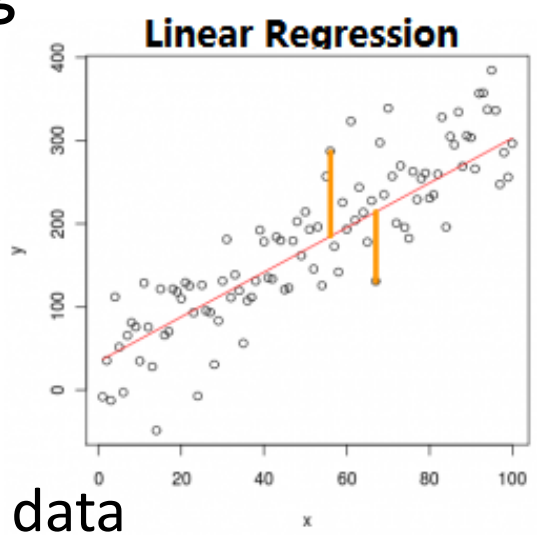
Unsupervised Learning

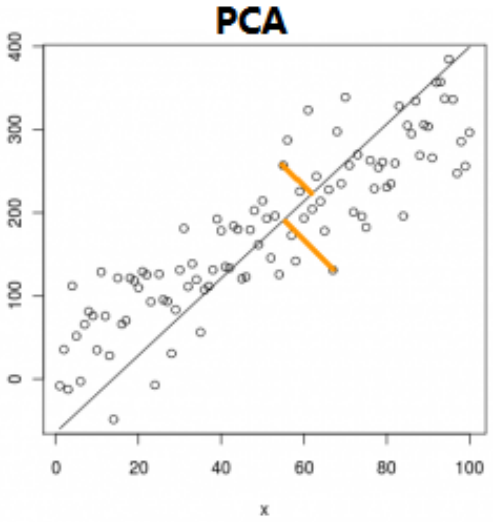
Data: x

Just data, no labels!

Goal:

Learn some underlying hidden *structure* of the data





Examples: Clustering, dimension reduction, feature learning, density estimation, etc.

minimize $\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - x_{approx}^{(i)}\right\|^2$

Principal Component Analysis (Dimensionality reduction)

Image from http://www.cnblogs.com/python27/p/MachineLearningWeek08.html

Supervised Learning vs Unsupervised Learning

Unsupervised Learning

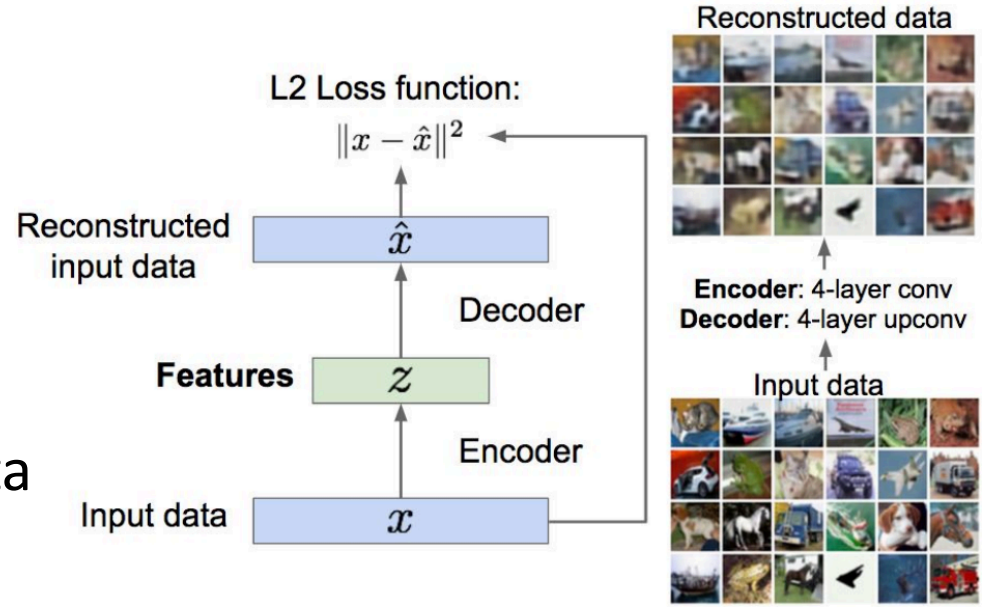
Data: x

Just data, no labels!

Goal:

Learn some underlying hidden *structure* of the data

Examples: Clustering, dimension reduction, feature learning, density estimation, etc.



Autoencoders (Feature learning)

Generative Models

Image from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Supervised Learning vs Unsupervised Learning

Unsupervised Learning

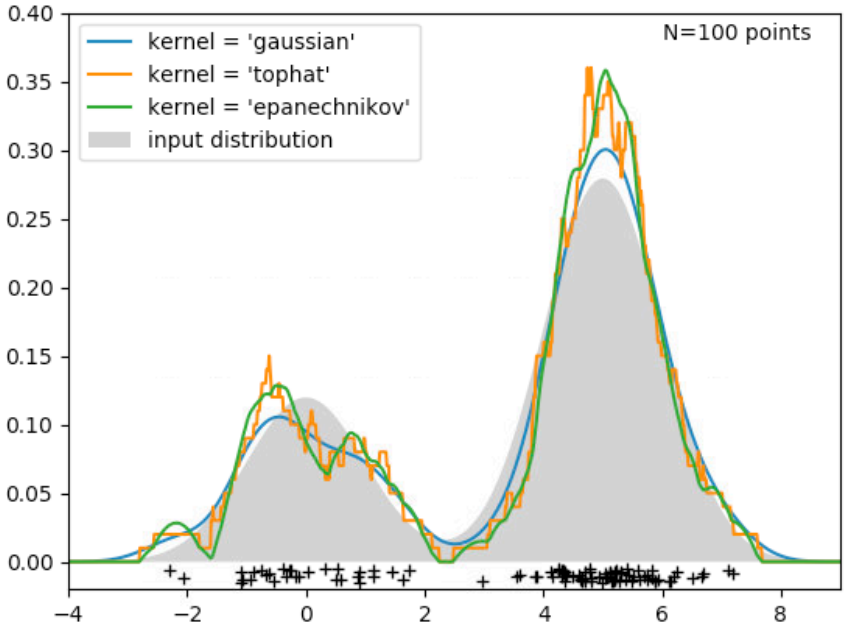
Data: x

Just data, no labels!

Goal:

Learn some underlying hidden *structure* of the data

Examples: Clustering, dimension reduction, feature learning, density estimation, etc.



Density estimation

Generative Models

Image from http://scikit-learn.org/stable/_images/sphx_glr_plot_kde_1d_003.png

Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.

Unsupervised Learning

Data: x
Just data, no labels!

Goal:
Learn some underlying hidden *structure* of the data

Examples: Clustering, dimension reduction, feature learning, density estimation, etc.

Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.

Unsupervised Learning

Data: x
Just data, no labels!

Training data is cheap

Goal:
Learn some underlying hidden *structure* of the data

Examples: Clustering, dimension reduction, feature learning, density estimation, etc.

Supervised Learning vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal:
Learn a function to map x -> y

Examples: Regression, classification, object detection, semantic segmentation, image captioning, etc.

Unsupervised Learning

Data: x
Just data, no labels!

Training data is cheap

Goal:
Learn some underlying hidden *structure* of the data

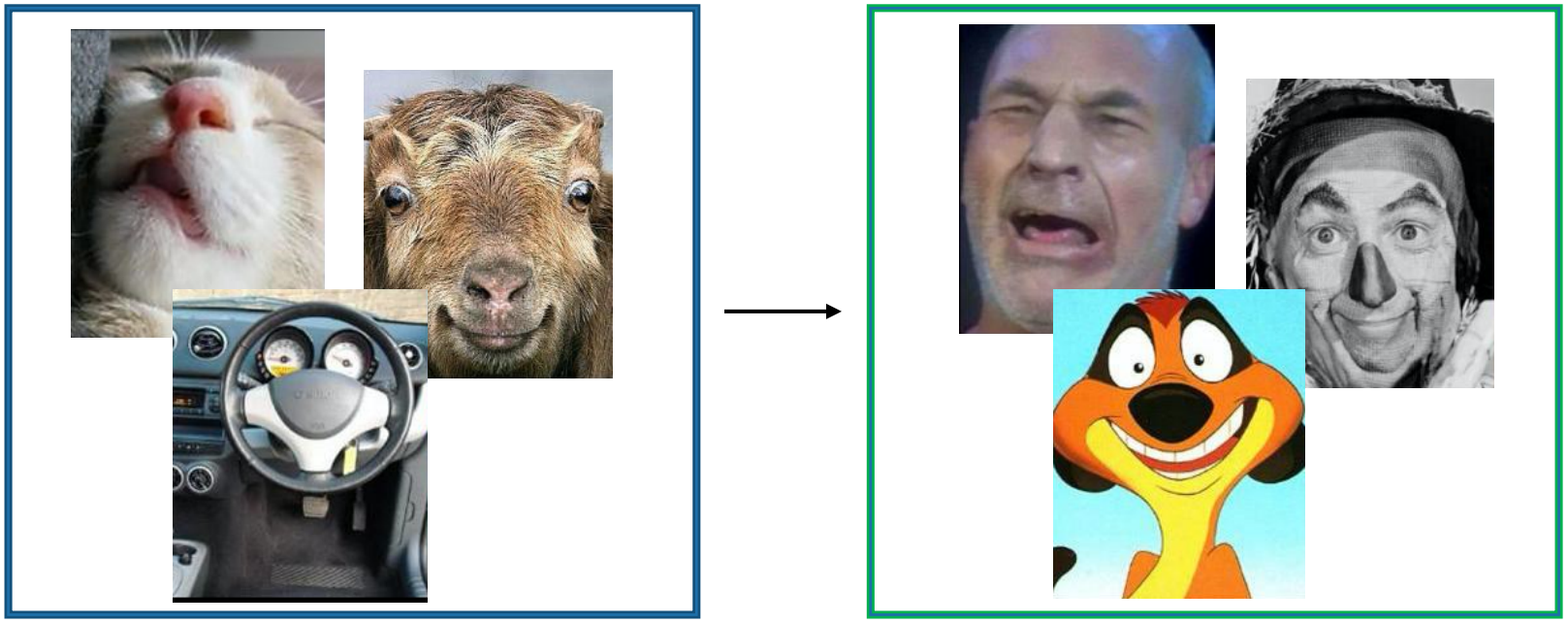
Examples: Clustering, dimension reduction, feature learning, density estimation, etc.

Generative Models

What are generative models?

Generative Models

Takes a training set, consisting of samples drawn from a distribution p_{data}, and learns to represent an estimate of that distribution with p_{model}.

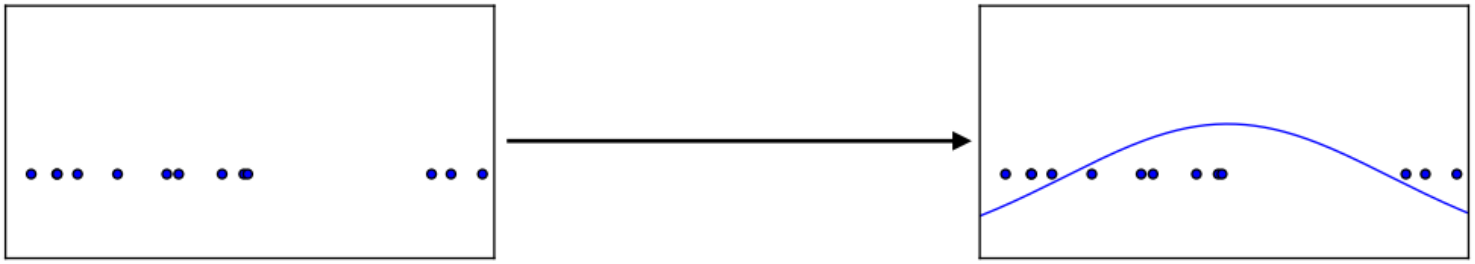


p_{data} p_{model}

Images from https://sites.google.com/view/totally-looks-like-dataset

Generative Models

Density estimation



$p_{data}(x)$ $p_{model}(x)$

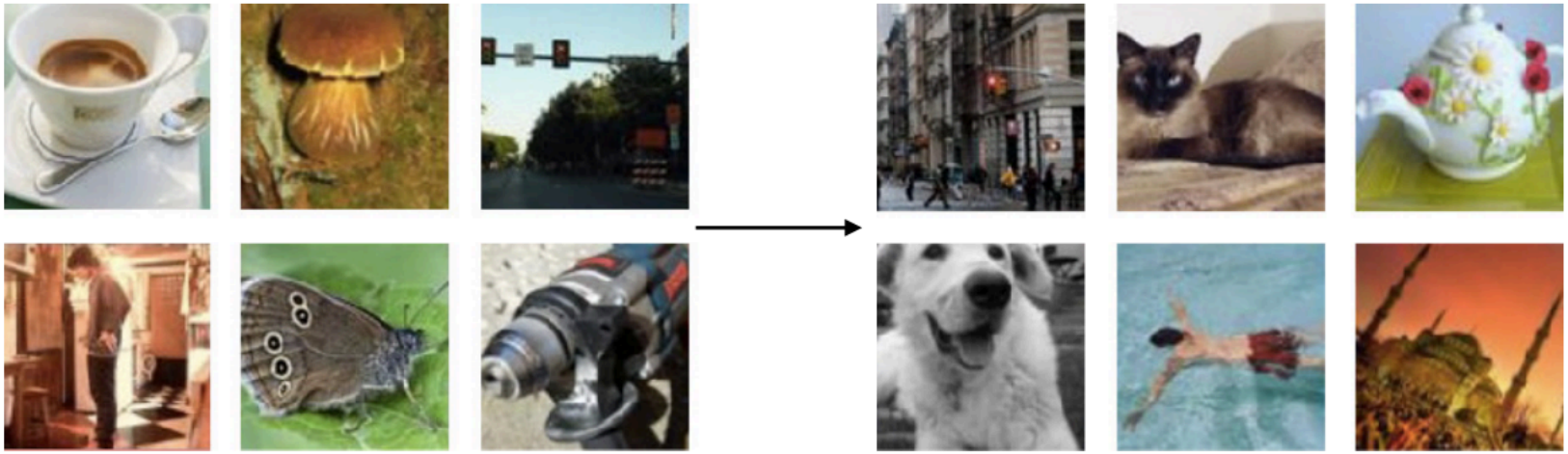
Here, the model estimates $p_{model}(x)$ explicitly, such as Gaussian distribution.

Images from http://www.iangoodfellow.com/slides/2016-12-04-NIPS.pdf

Generative Models

Sample generation

Given training data, generate new samples from same distribution.



Training data
$p_{data}(x)$

Model samples
$p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

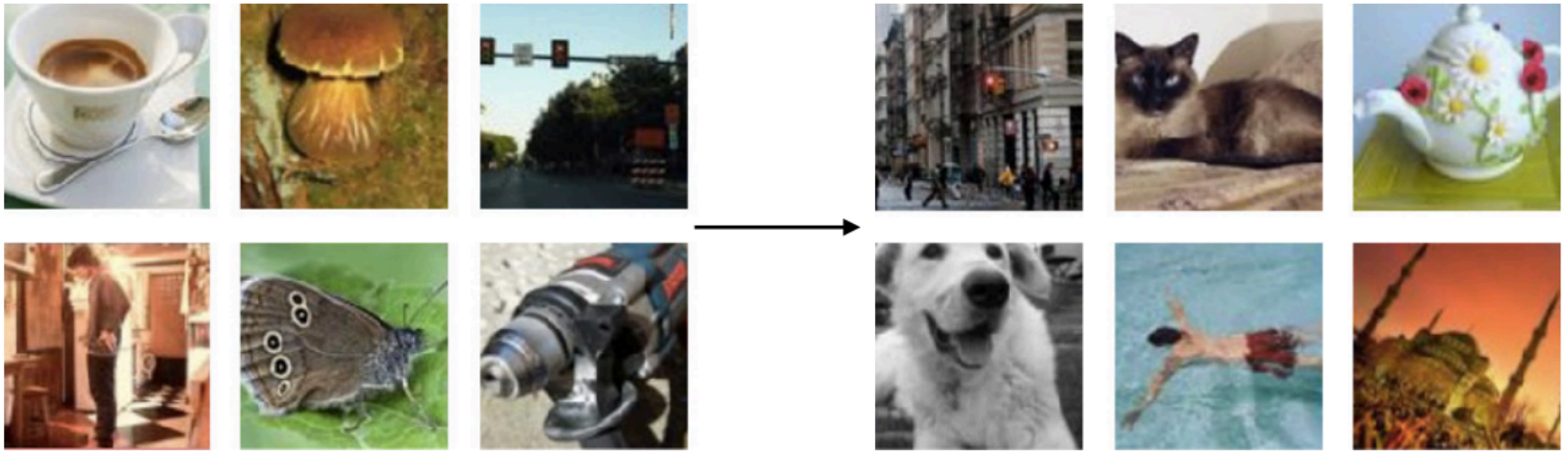
Images from http://www.iangoodfellow.com/slides/2016-12-04-NIPS.pdf

Generative Models

What we mainly talk in this class

Sample generation

Given training data, generate new samples from same distribution.



Training data
$p_{data}(x)$

Model samples
$p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Images from http://www.iangoodfellow.com/slides/2016-12-04-NIPS.pdf

Why study generative models?

Why study generative models?

Have you ever seen these stars?

Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation” in ICLR, 2018.

Why study generative models?

Have you ever seen these stars?

Of course not … because they are all … FAKE!

Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation” in ICLR, 2018.

Why study generative models?

Realistic Image Generation

Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation” in ICLR, 2018.

Why study generative models?

Anime Illustration Generation

Real

Fake

Images from https://github.com/tdrussell/IllustrationGAN

Why study generative models?

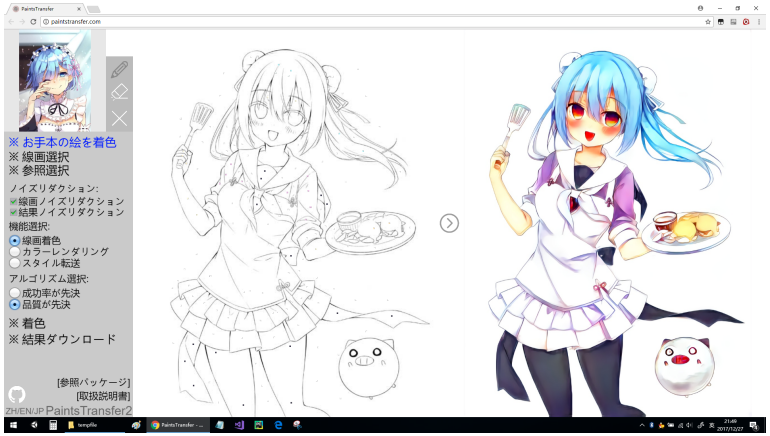
Colorize Pictures

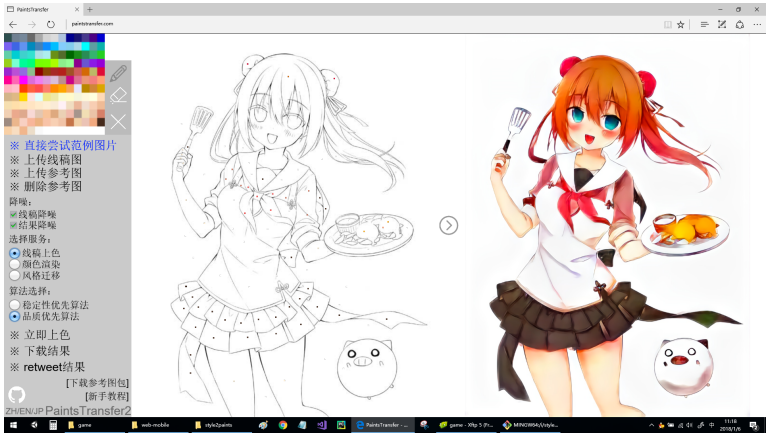
Images from https://github.com/lllyasviel/style2paints

Zhang et al., “Colorful Image Colorization” in ECCV, 2016

Why study generative models?

Colorize Pictures



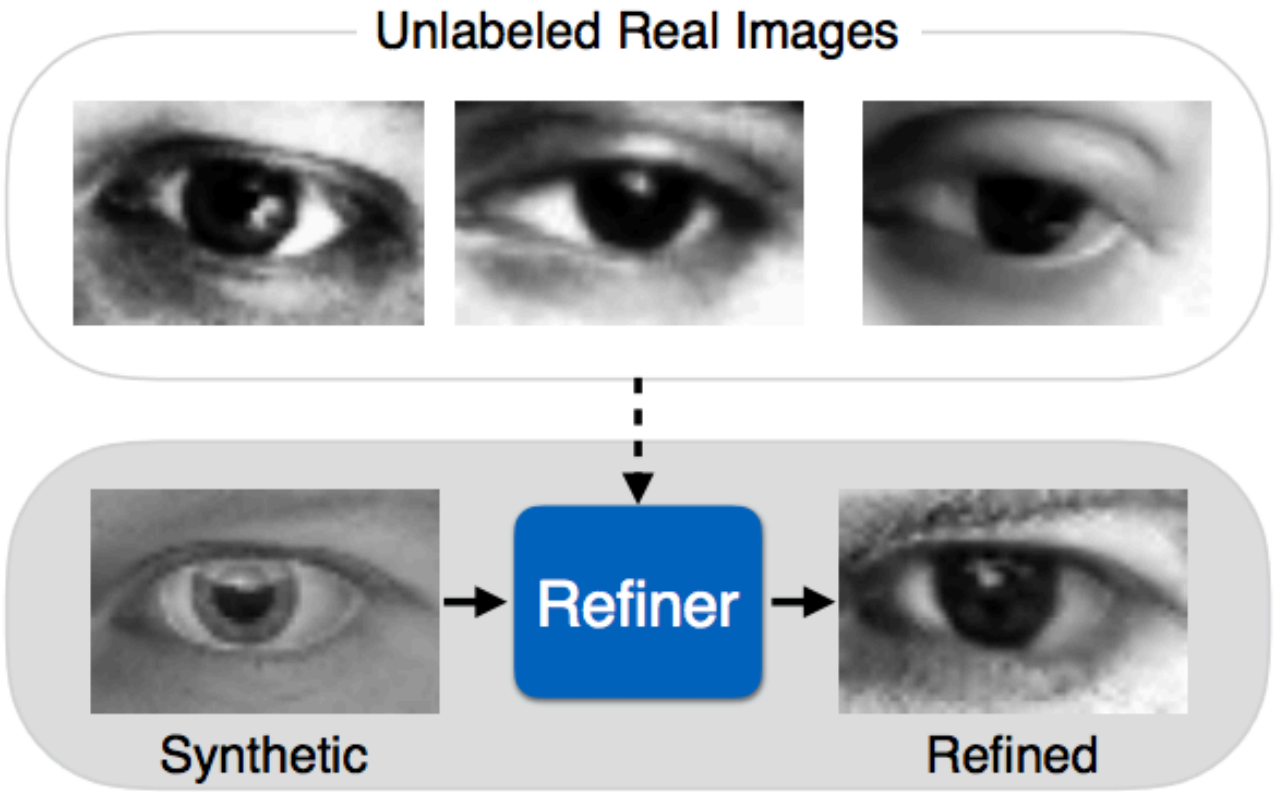


Images from https://github.com/lllyasviel/style2paints

Zhang et al., “Real-Time User-Guided Image Colorization with Learned Deep Priors” in In ACM Transactions on Graphics (SIGGRAPH), 2017

Why study generative models?

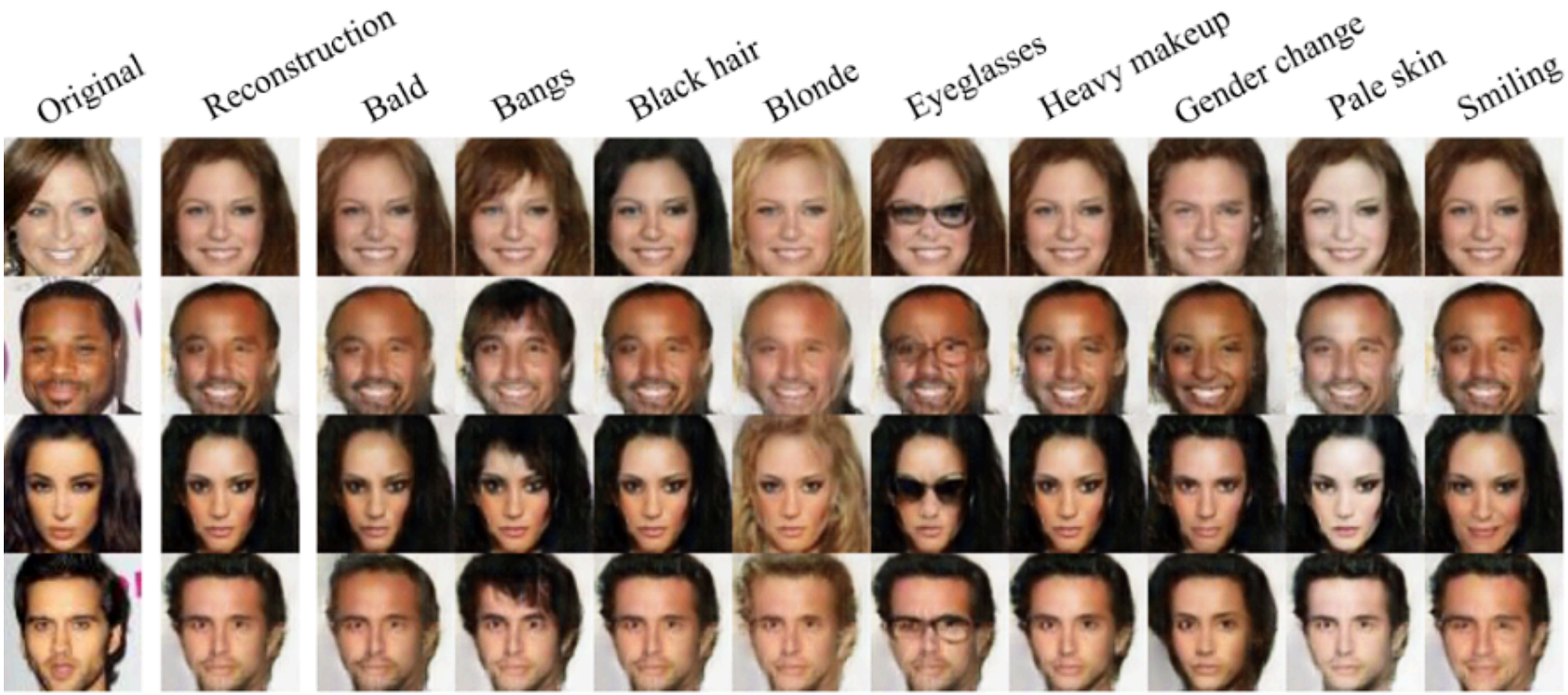
Synthetic Data Generation



Shrivastava et al., “Learning from Simulated and Unsupervised Images through Adversarial Training” in CVPR, 2017.

Why study generative models?

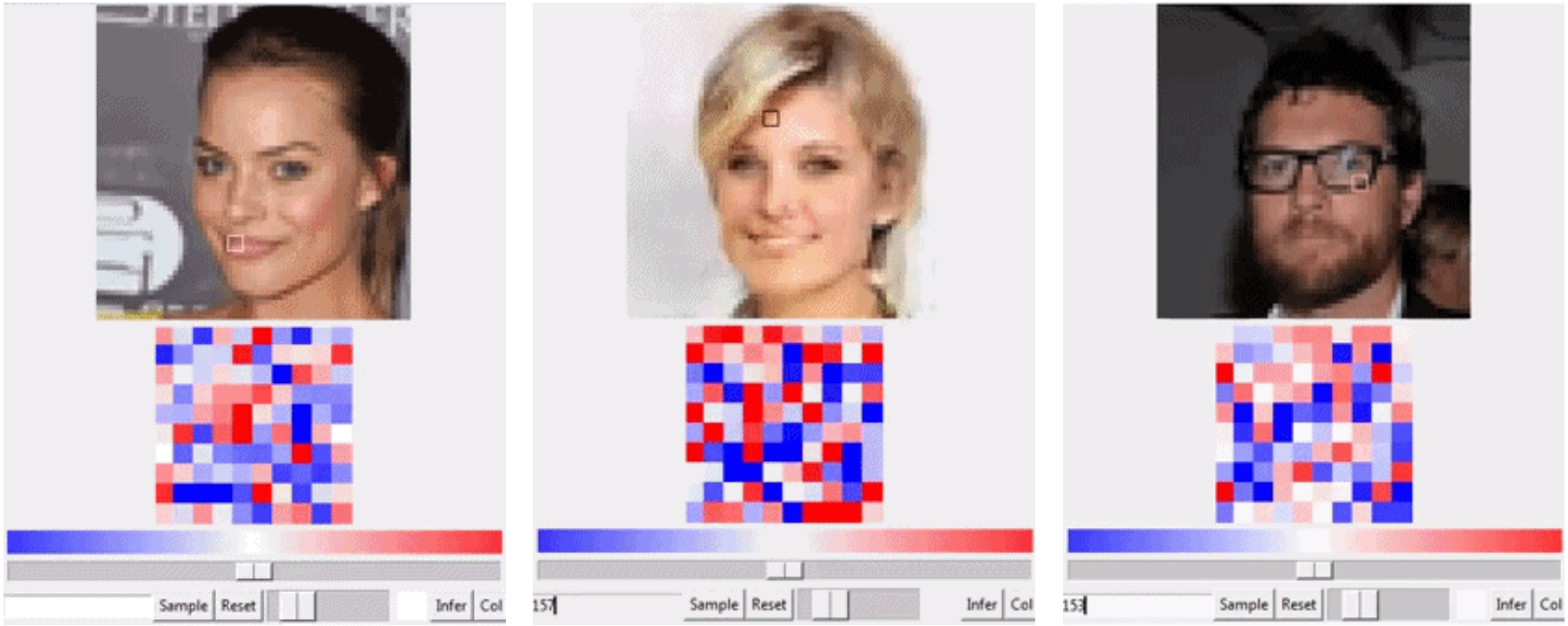
Image Editing



Perarnau et al., “Invertible Conditional GANs for image editing” in NIPS Workshop, 2016

Why study generative models?

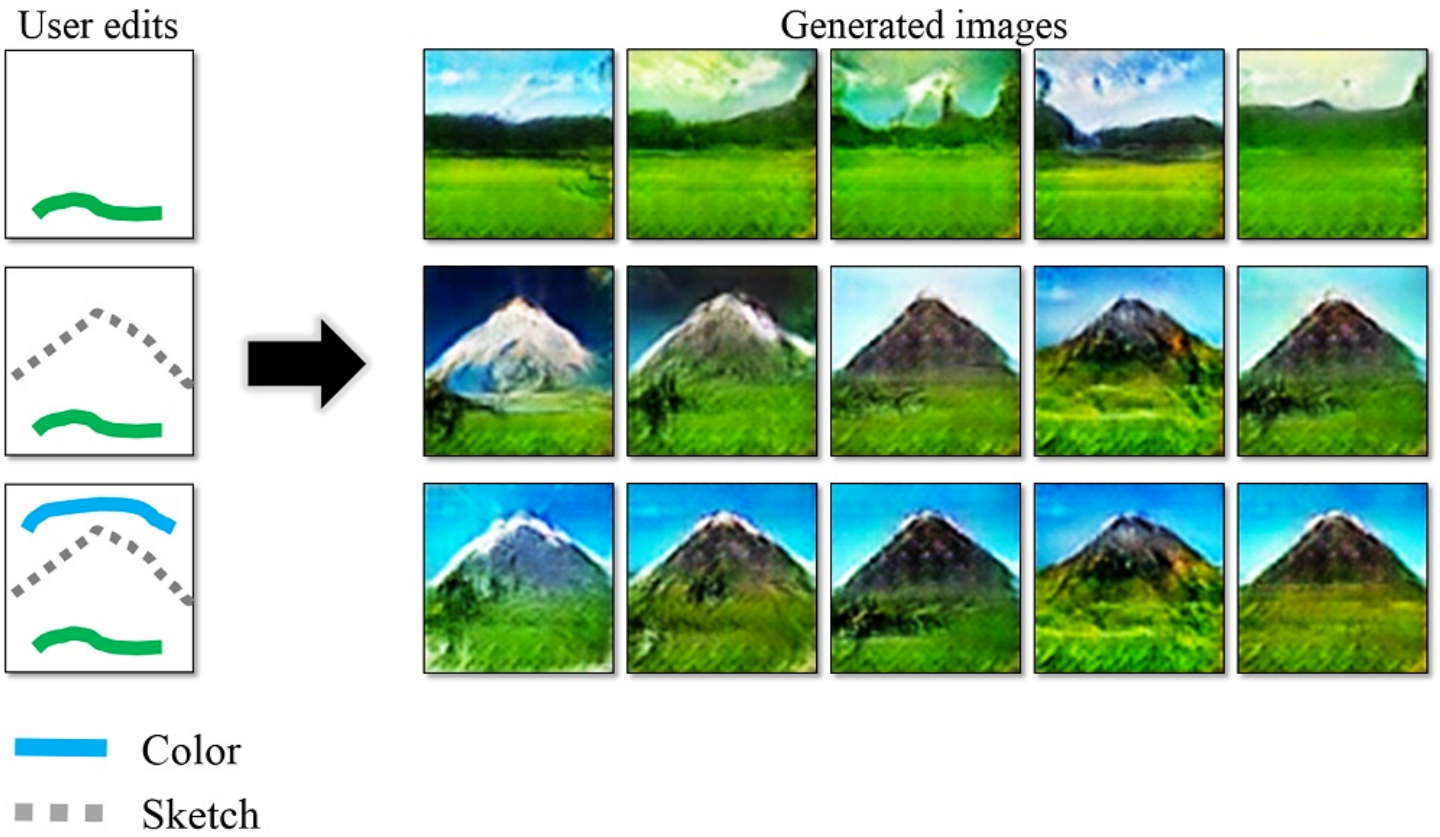
Neural Photo Editor



Brock et al., “Neural Photo Editing with Introspective Adversarial Networks” in arXiv, 2016

Why study generative models?

Interactive Image generation



Zhu et al., “Generative Visual Manipulation on the Natural Image Manifold” in ECCV, 2016.

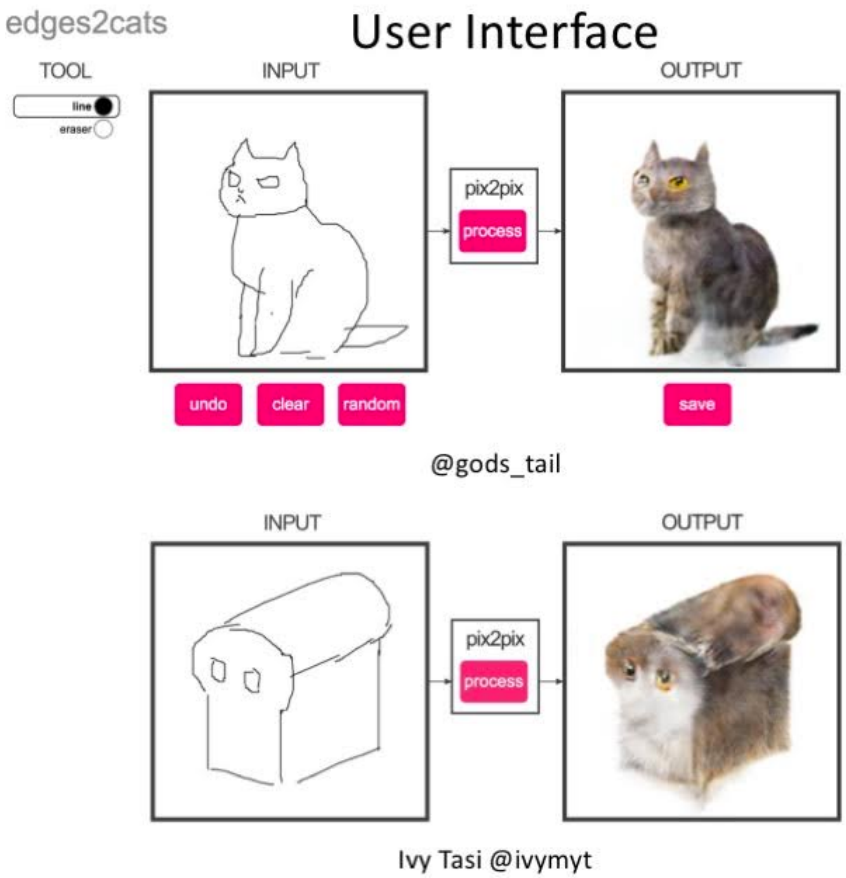
Why study generative models?

Interactive Image generation

Zhu et al., “Generative Visual Manipulation on the Natural Image Manifold” in ECCV, 2016.

Why study generative models?

Edge to Cats



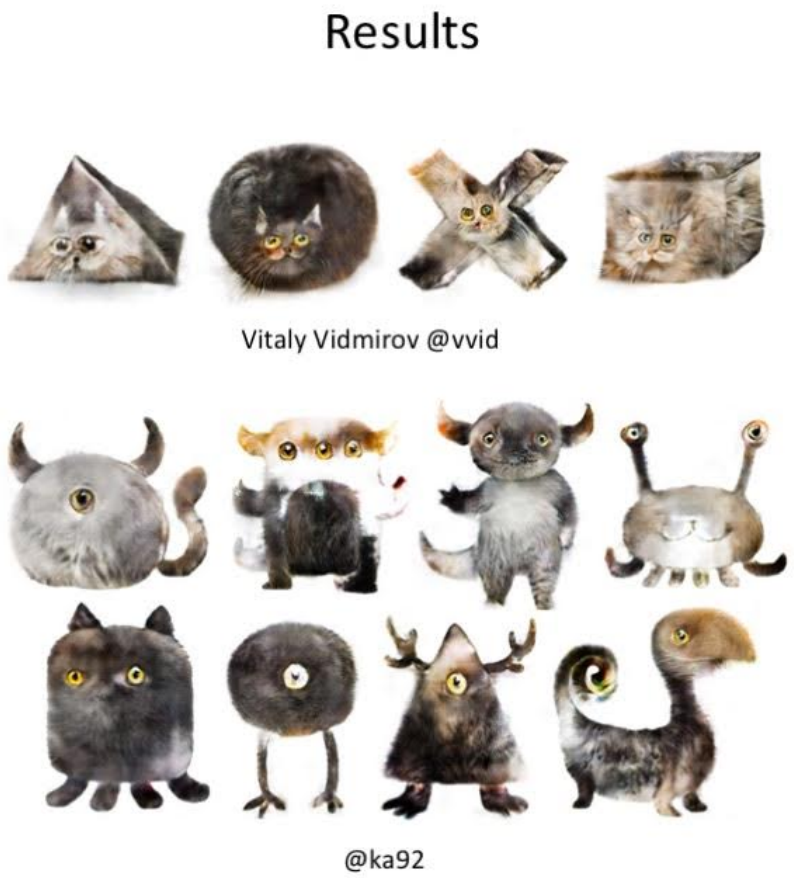
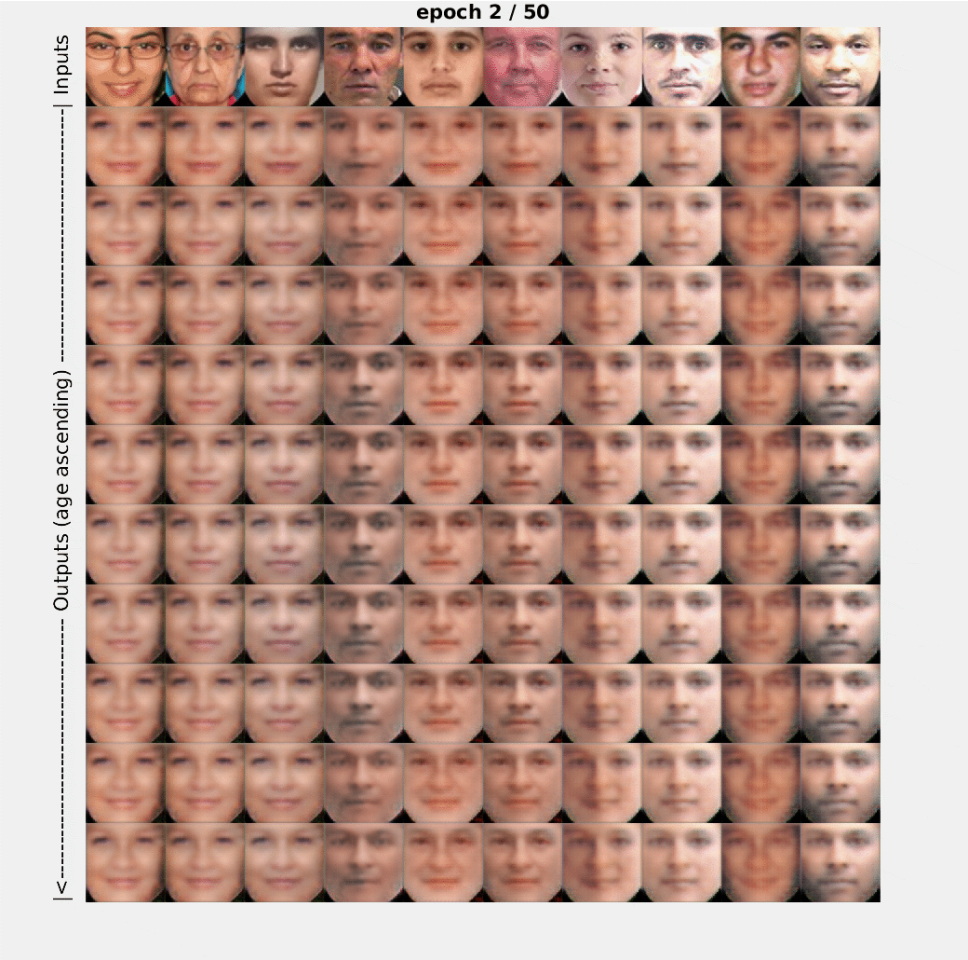


Image from https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

Why study generative models?

Age Progression/Regression



Zhang et al., “Age Progression/Regression by Conditional Adversarial Autoencoder” in CVPR, 2017.

Why study generative models?

Text to Image

A bird with a red belly and breast, green wings and blue head and neck

generative models →

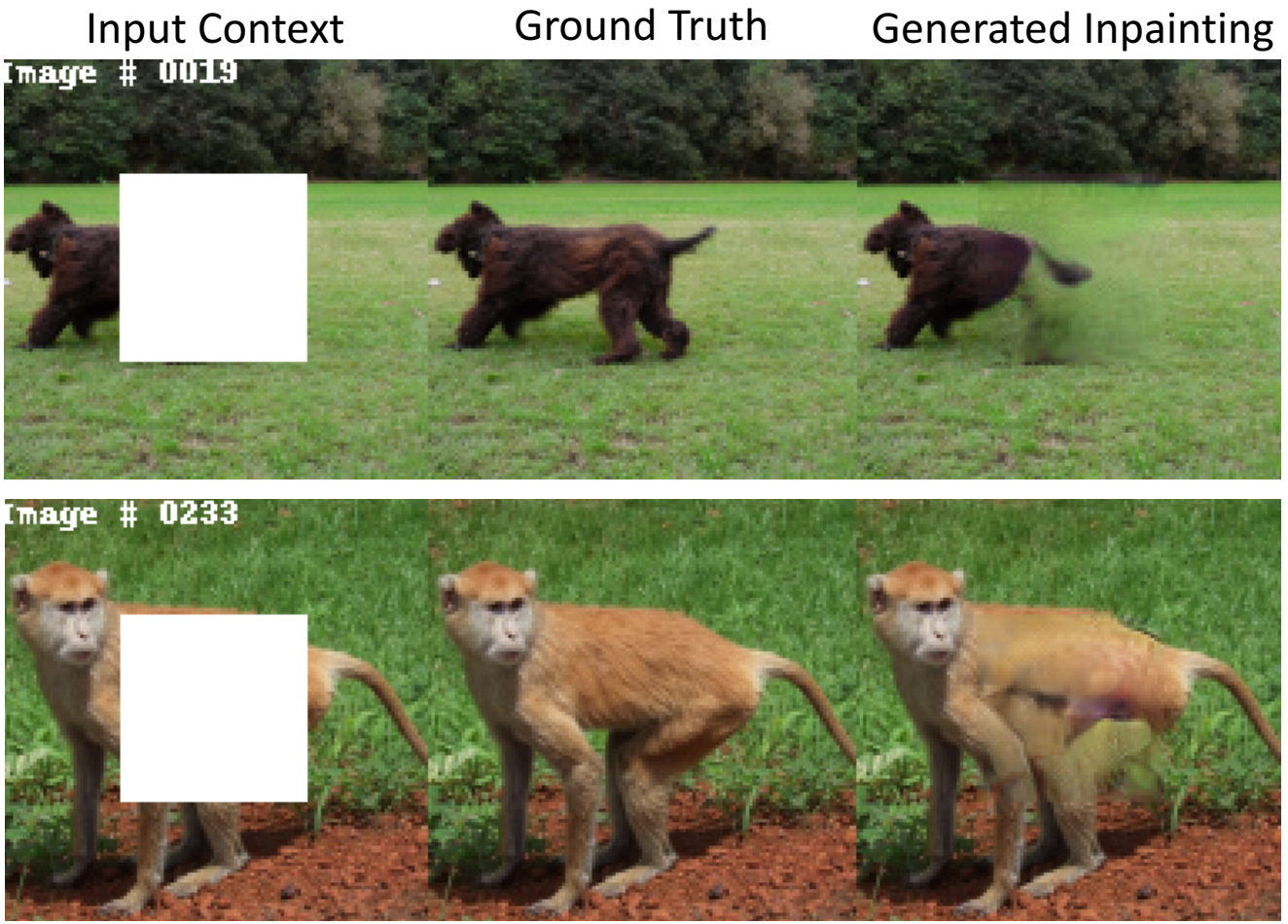
A white bird with a long orange beak and a dark spot over its head

generative models →

Shuang et al., “DA-GAN: Instance-level Image Translation by Deep Attention Generative Adversarial Networks” in CVPR, 2018.

Why study generative models?

Image Inpainting



Pathak et al., “Context Encoders: Feature Learning by Inpain” in CVPR, 2016

Why study generative models?

Super-Resolution

generative models

Ledig et al., “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network” in CVPR, 2017

Why study generative models?

Image Deraining

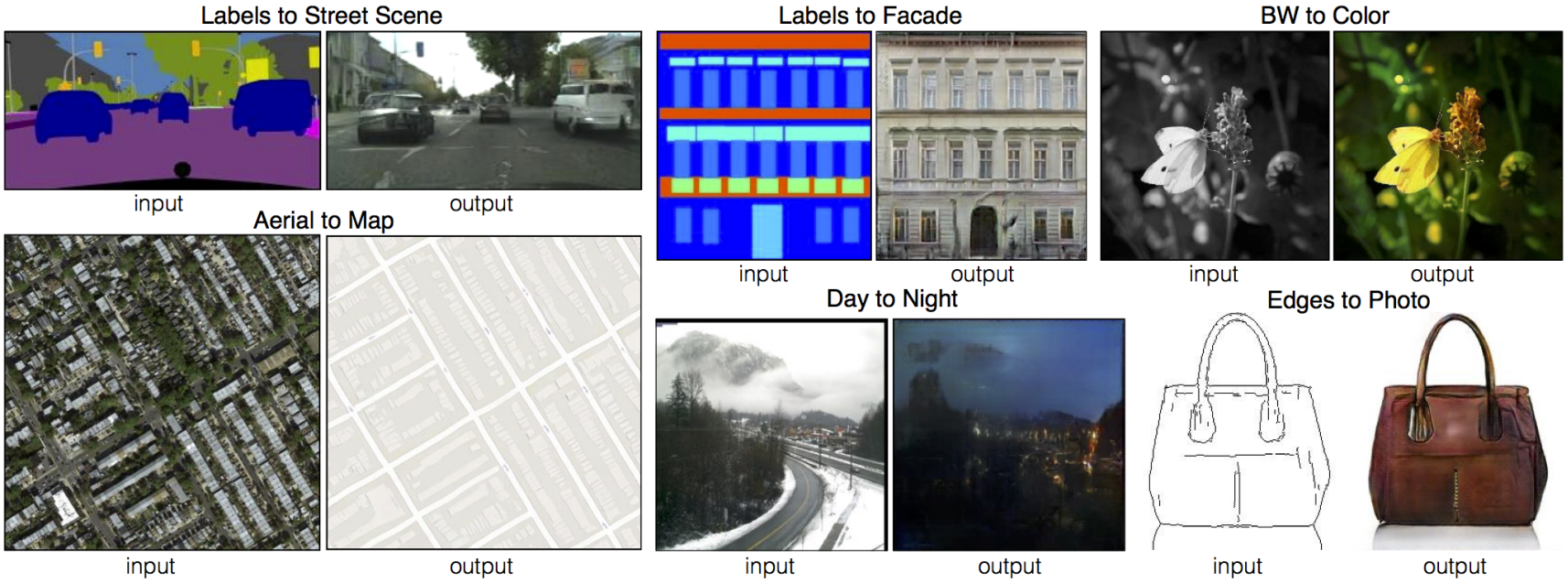
before deraining

after deraining

Zhang et al., “Image De-raining Using a Conditional Generative Adversarial Network” in arXiv, 2017

Why study generative models?

Domain-transfer



Isola et al., "Image-to-Image Translation with Conditional Adversarial Networks" in CVPR, 2017

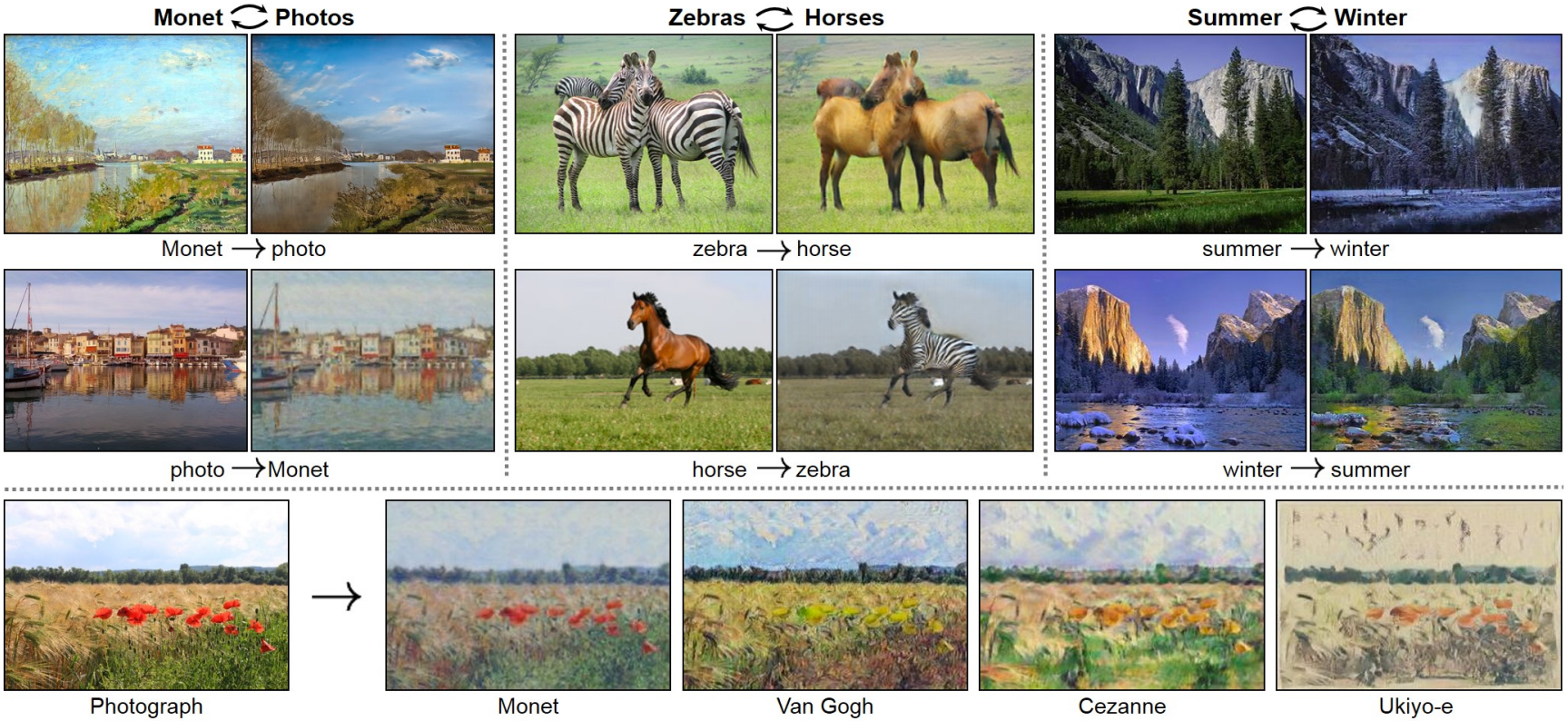
Why study generative models?

Domain-transfer

Zhu et al., “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss” in ICCV, 2017

Why study generative models?

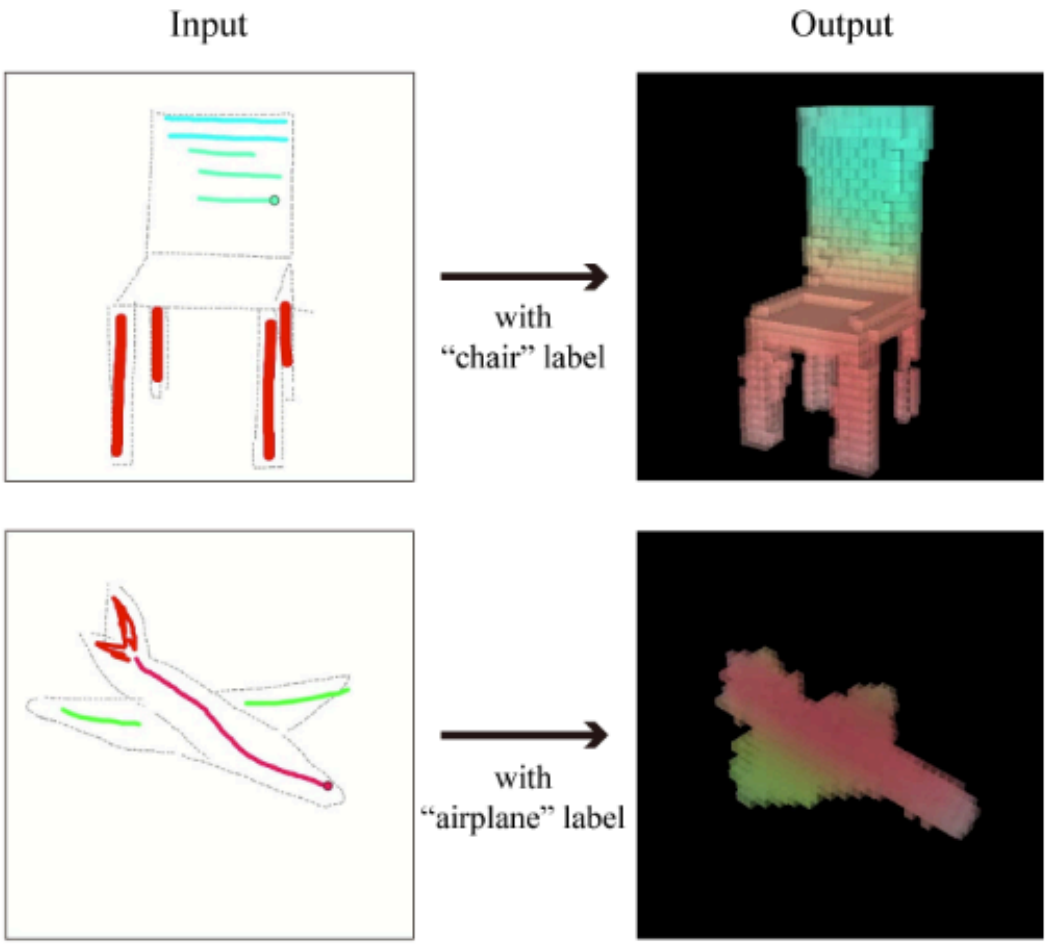
Domain-transfer



Zhu et al., "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss" in ICCV, 2017

Why study generative models?

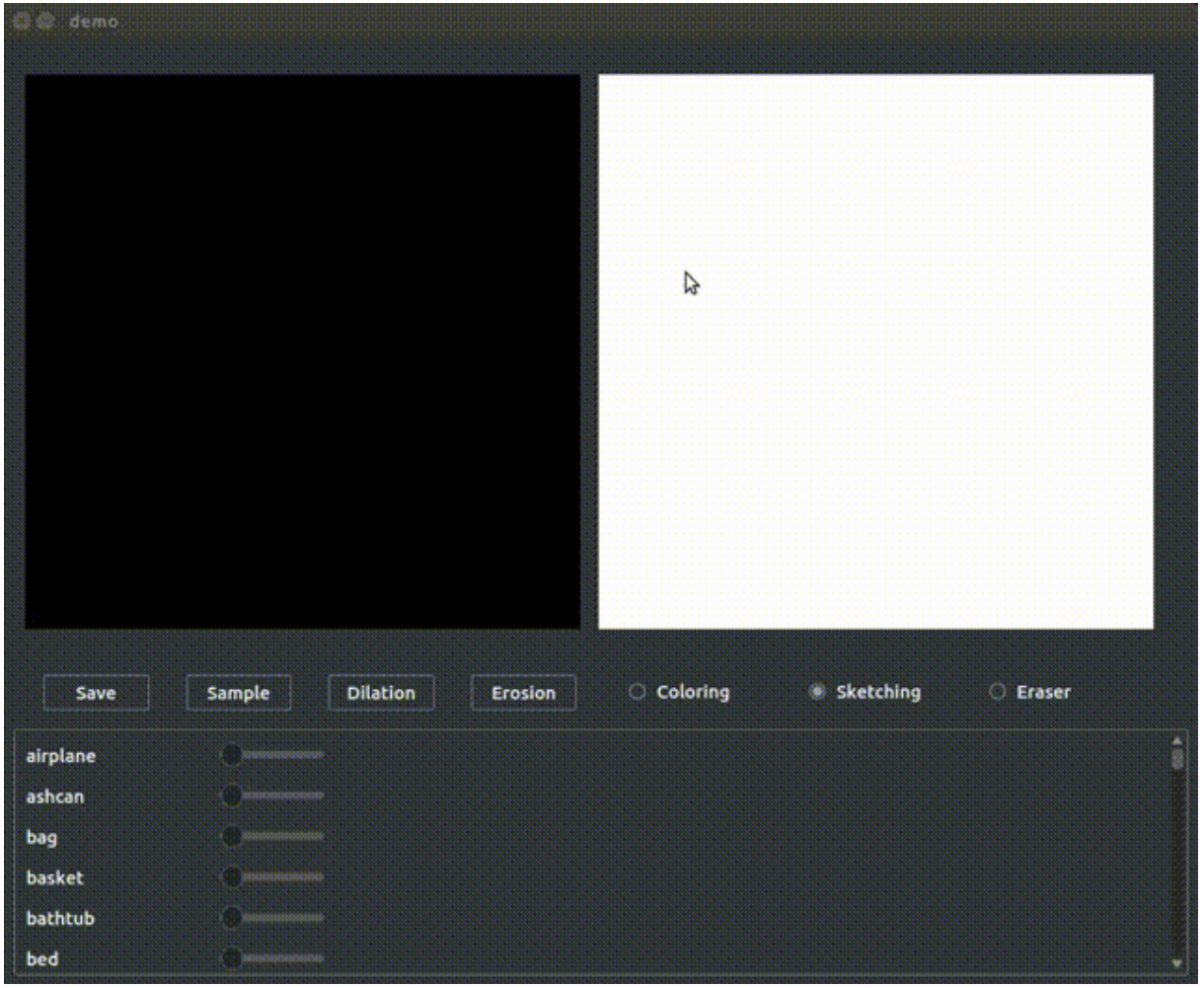
3D Object Generation



https://github.com/maxorange/pix2vox

Why study generative models?

3D Object Generation

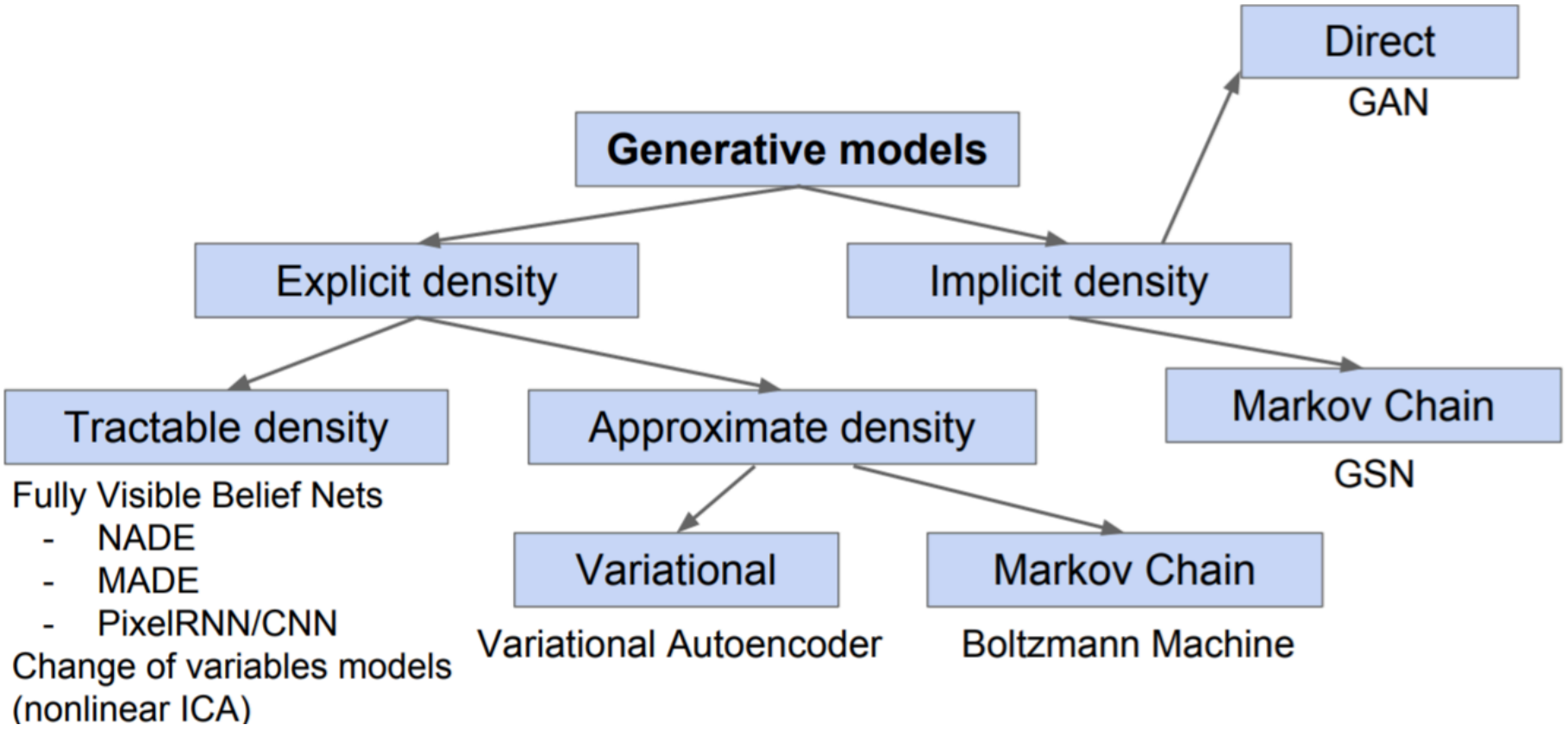


https://github.com/maxorange/pix2vox

So, why study generative models?

Because it is cool.

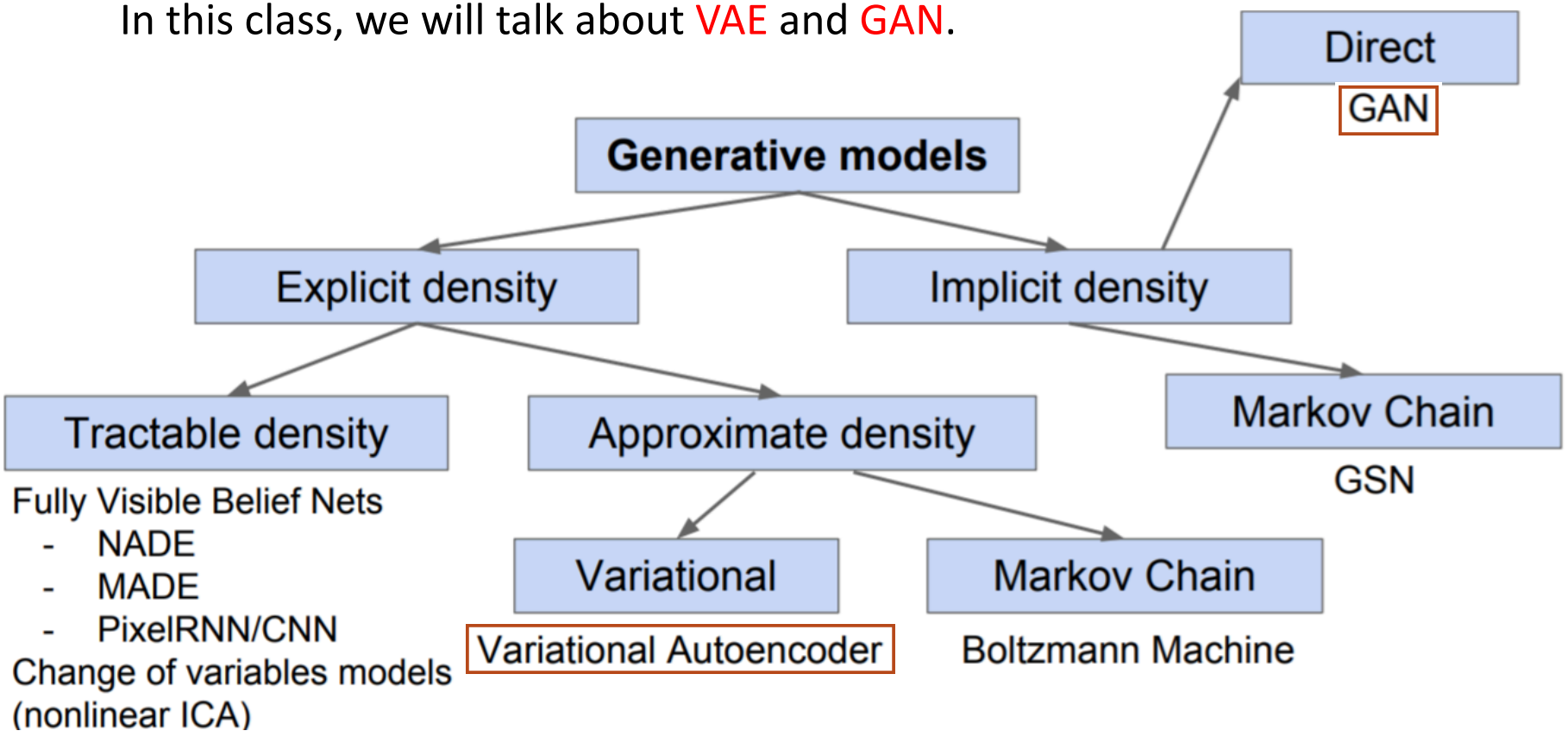
Taxonomy of Generative Models



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Taxonomy of Generative Models

In this class, we will talk about VAE and GAN.

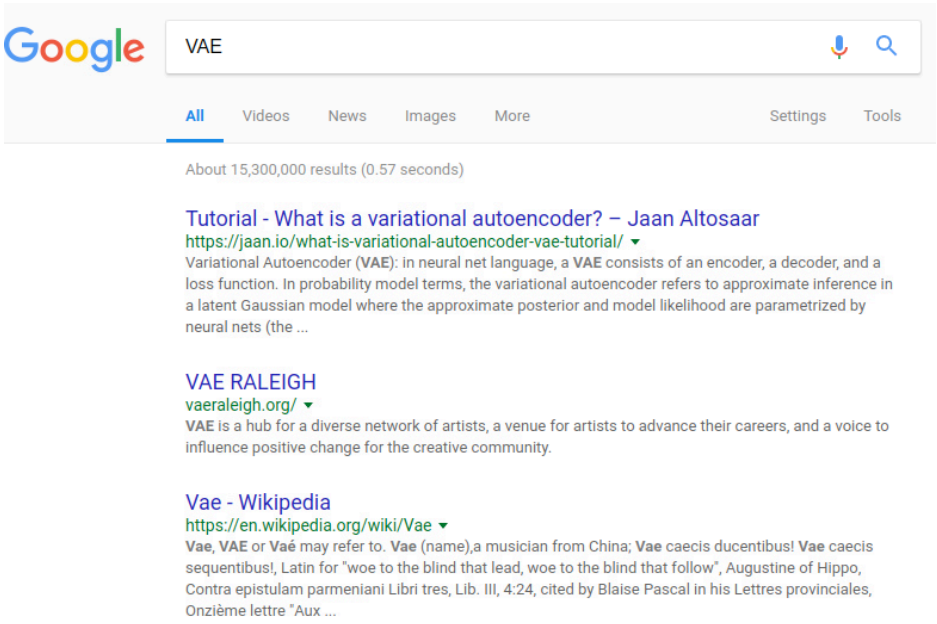


http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Variational Autoencoders (VAE)

VAE?

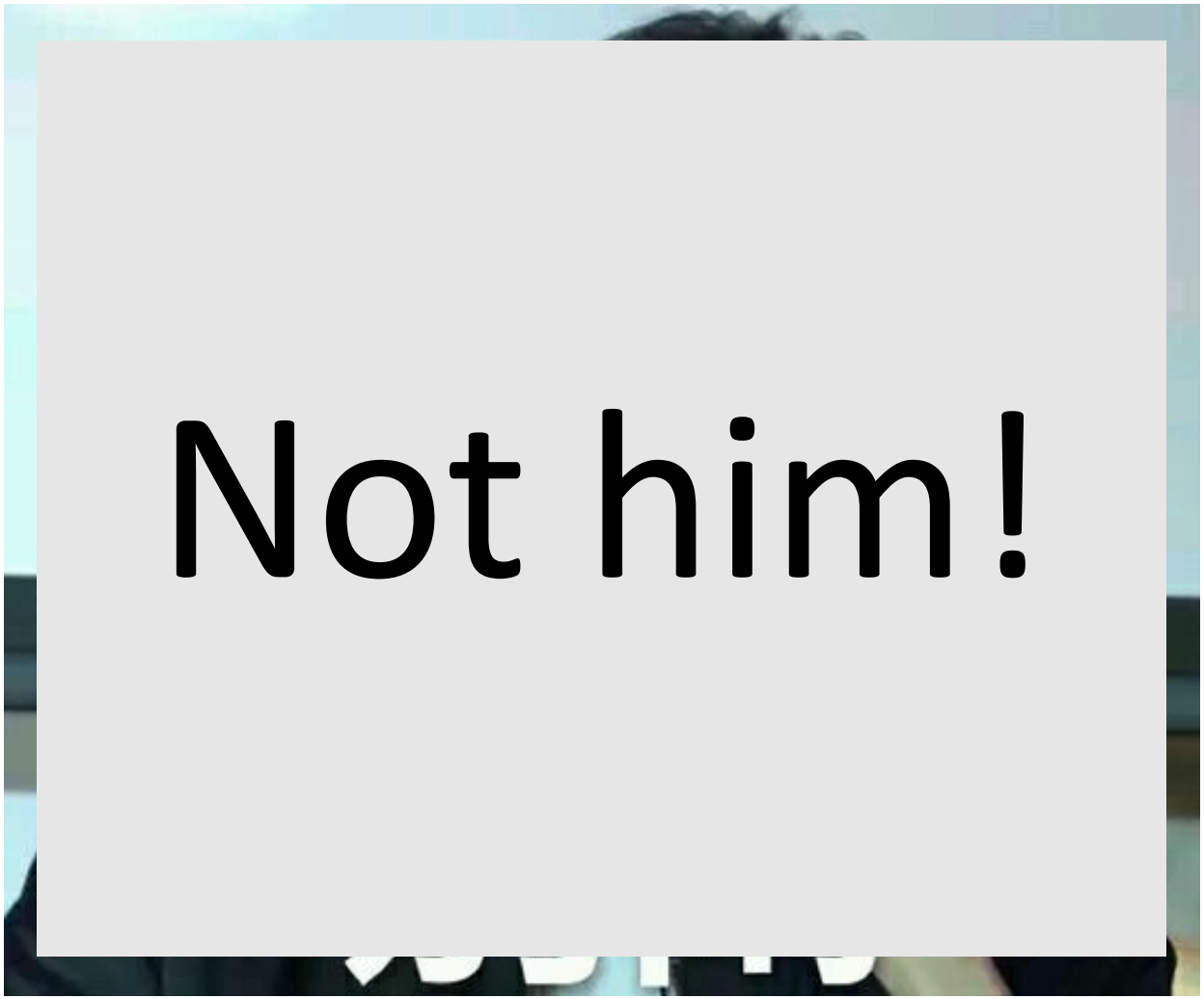

Baidu 百度
VAE
百度一下
网页 新闻 贴吧 知道 音乐 图片 视频 地图 文库 更多»
百度为您找到相关结果约13,900,000个
搜索工具
您可以仅查看：英文结果
vae_百度百科
职业：歌手
生日：1986年5月14日
个人信息：180cm/60kg/金牛座/A型
代表作品：最佳歌手、雅俗共赏、玫瑰花的葬礼、城府、叹...
简介：许嵩（Vae），1986年5月14日生于安徽省合肥市，中国...
早年经历 演艺经历 主要作品 获奖记录 人物评价
baike.baidu.com/
为您推荐：许嵩vae的由来 许嵩的女朋友徐佳颖 许嵩在海蝶的地位
许嵩vae的由来_百度知道
1个回答 - 提问时间: 2014年08月07日
最佳答案: 首先,我要强调一下哦。许嵩是 Vae 不是 VAE 也是不 vae
话说Vae这个单词已经深入我们的心中了,我自己英语很差,单词...
https://zhidao.baidu.com/quest...
vae是什么意思? 23个回答 2011-02-09
vae有首歌的歌词是 我叫VAE 5个回答 2013-12-14
VAE乳液707和705的区别是什么啊? 6个回答 2008-12-29
更多知道相关问题>>


Google
VAE
All Videos News Images More Settings Tools
About 15,300,000 results (0.57 seconds)
Tutorial - What is a variational autoencoder? – Jaan Altosaar
https://jaan.io/what-is-variational-autoencoder-vae-tutorial/
Variational Autoencoder (VAE): in neural net language, a VAE consists of an encoder, a decoder, and a loss function. In probability model terms, the variational autoencoder refers to approximate inference in a latent Gaussian model where the approximate posterior and model likelihood are parametrized by neural nets (the ...
VAE RALEIGH
vaeraleigh.org/
VAE is a hub for a diverse network of artists, a venue for artists to advance their careers, and a voice to influence positive change for the creative community.
Vae - Wikipedia
https://en.wikipedia.org/wiki/Vae
Vae, VAE or Vaé may refer to. Vae (name),a musician from China; Vae caecis ducentibus! Vae caecis sequentibus!, Latin for "woe to the blind that lead, woe to the blind that follow", Augustine of Hippo, Contra epistulam parmeniani Libri tres, Lib. III, 4:24, cited by Blaise Pascal in his Lettres provinciales, Onzième lettre "Aux ...

VAE?

http://tieba.baidu.com/p/4556713664

VAE?



http://tieba.baidu.com/p/4556713664

VAE

Variational Autoencoders

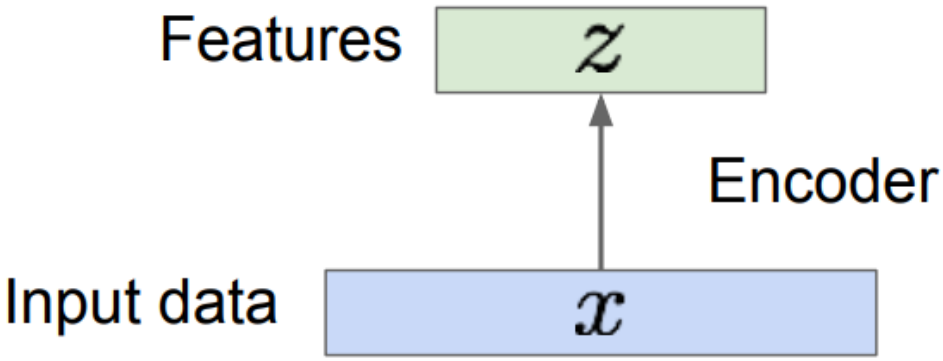
VAE

Variational Autoencoders

Background: Autoencoders

What is an Autoencoder?

Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data



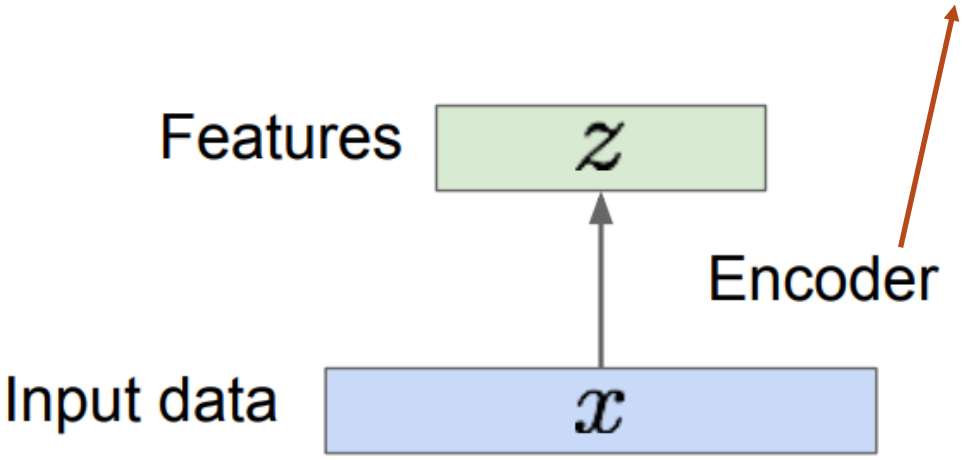
http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Background: Autoencoders

What is an Autoencoder?

Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

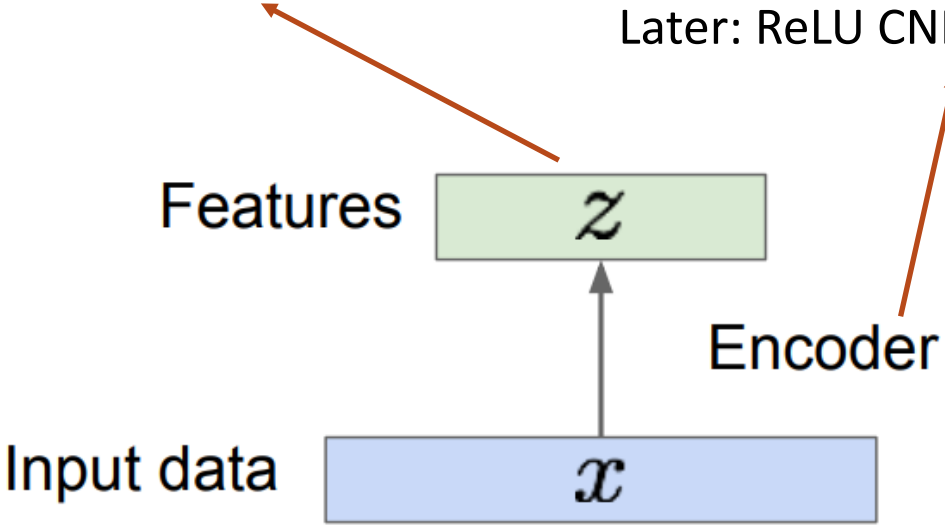
Background: Autoencoders

What is an Autoencoder?

Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

z usually smaller than x
(dimensionality reduction)

Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Background: Autoencoders

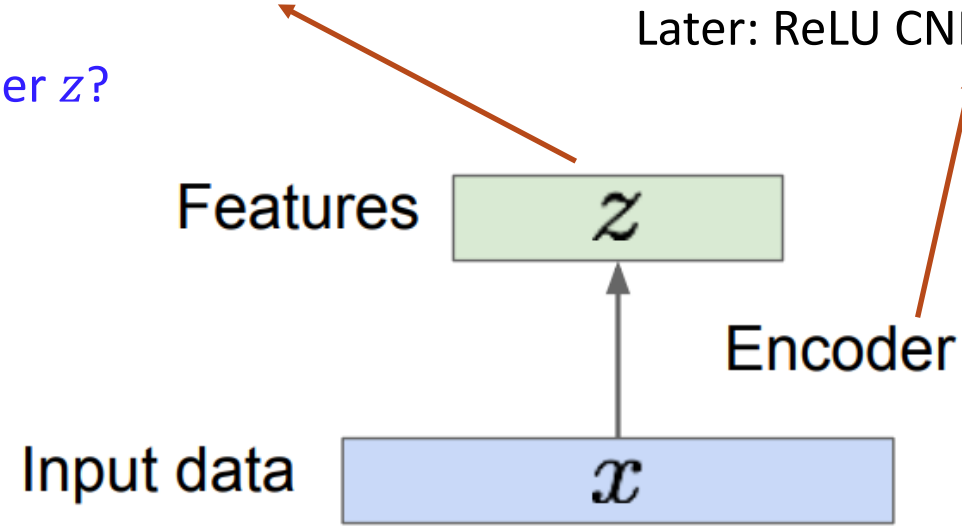
What is an Autoencoder?

Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

z usually smaller than x
(dimensionality reduction)

Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN

Q: Why smaller z?



Background: Autoencoders

What is an Autoencoder?

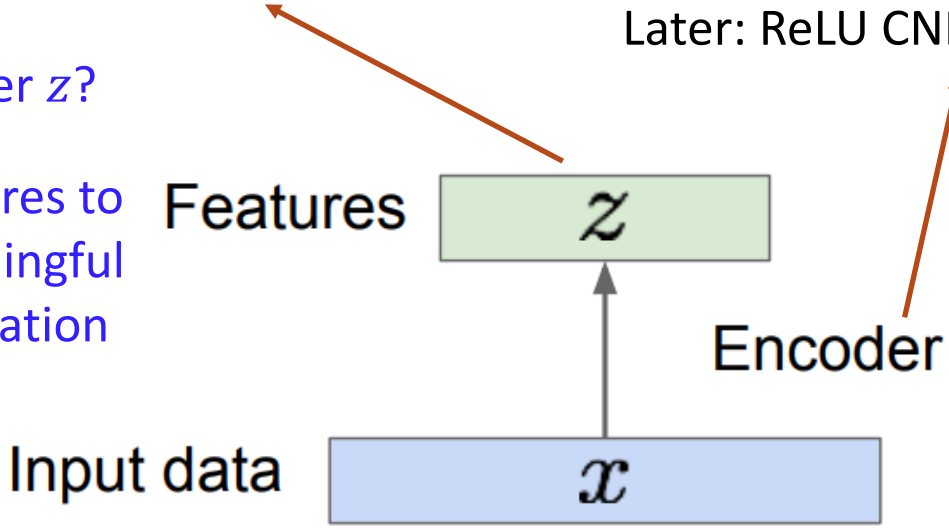
Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data

z usually smaller than x
(dimensionality reduction)

Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN

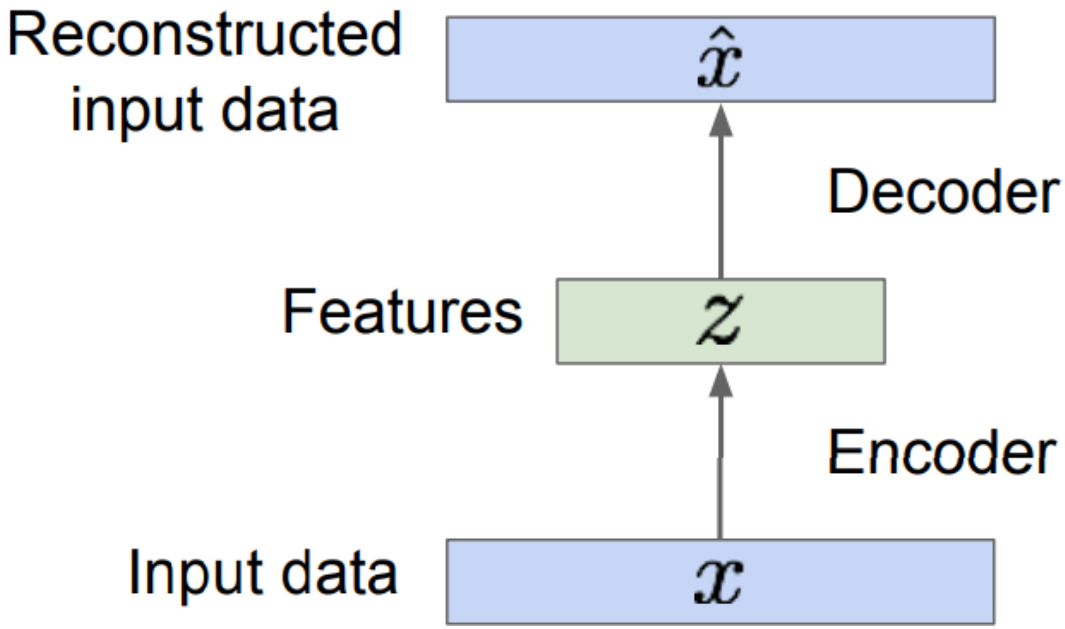
Q: Why smaller z?

A: Want features to capture meaningful factors of variation in data



Background: Autoencoders

How to learn the feature representation?



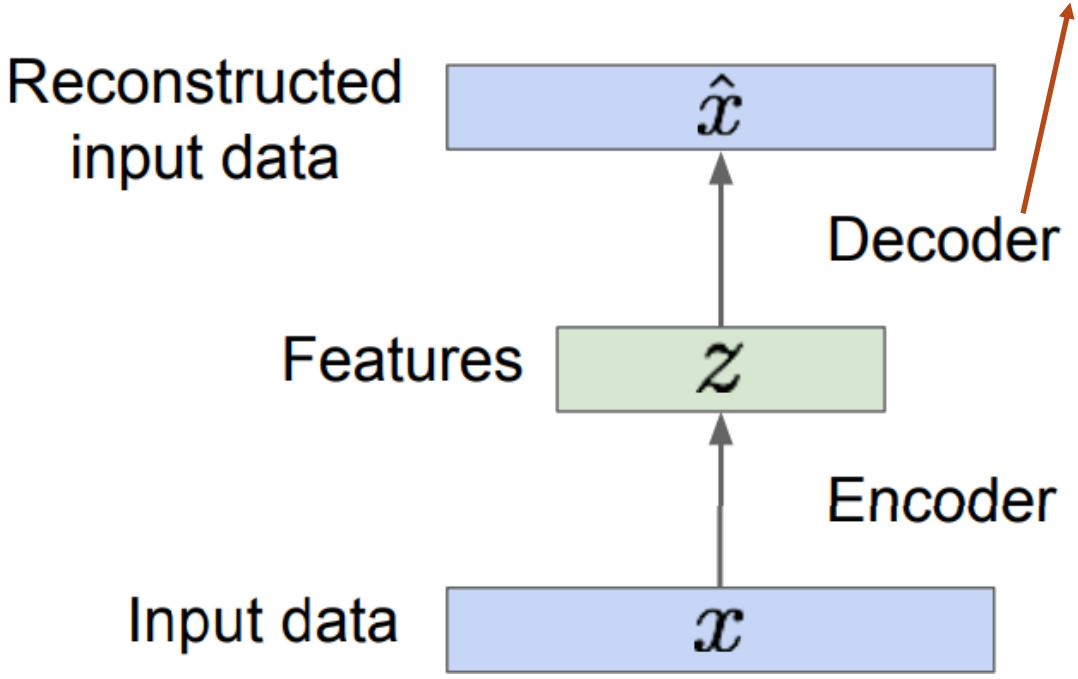
http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Background: Autoencoders

How to learn the feature representation?

Train such that features can be used to reconstruct original data

Originally: Linear + nonlinearity (sigmoid)
Later: Deep, fully-connected
Later: ReLU CNN (upconv)

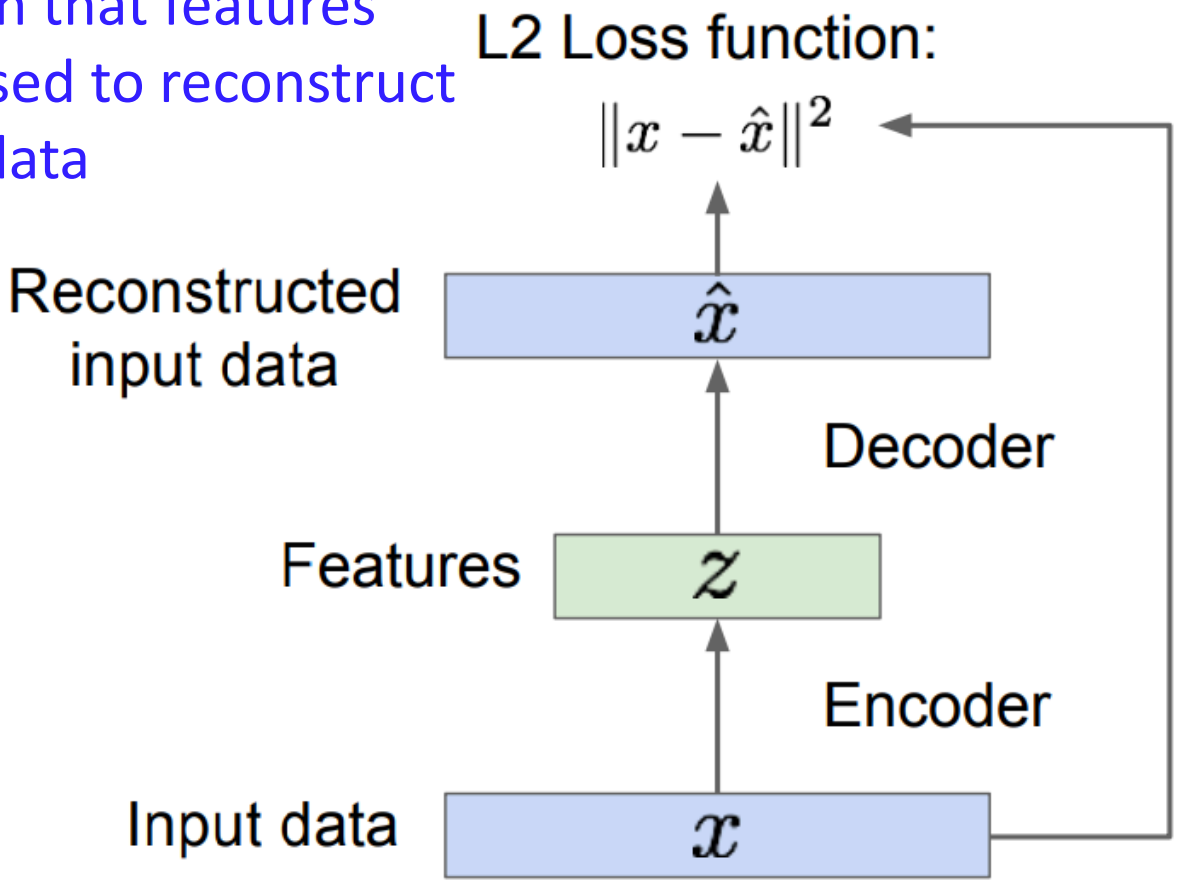


http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Background: Autoencoders

How to learn the feature representation?

Train such that features can be used to reconstruct original data



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Background: Autoencoders

A typical Autoencoder

"Autoencoding" – encoding itself

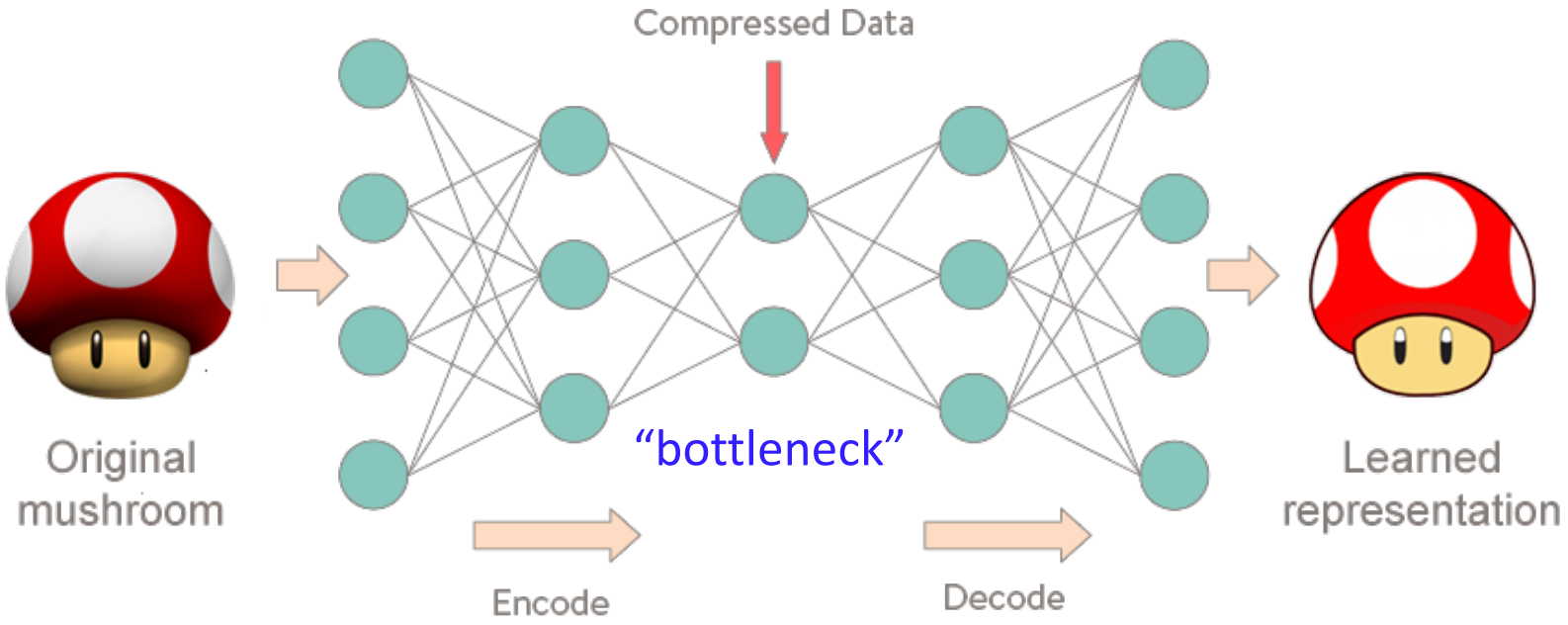


Image from http://curiousily.com/data-science/2017/02/02/what-to-do-when-data-is-missing-part-2.html

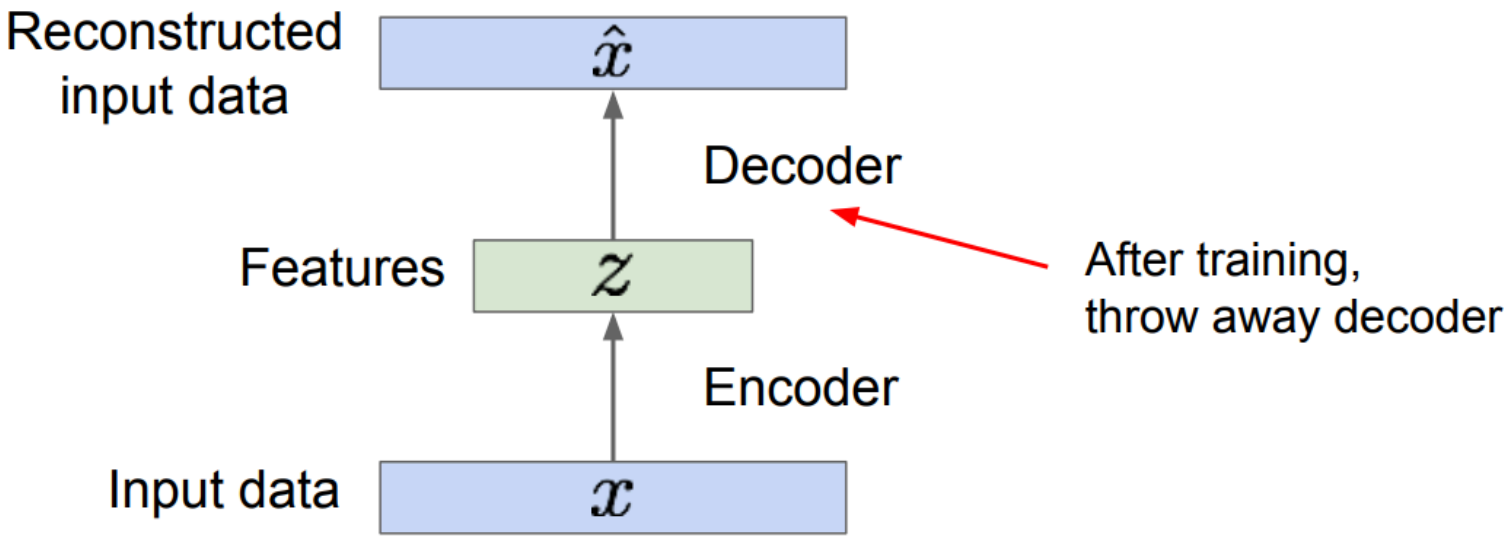
Background: Autoencoders

ConvNetJS demo

AE

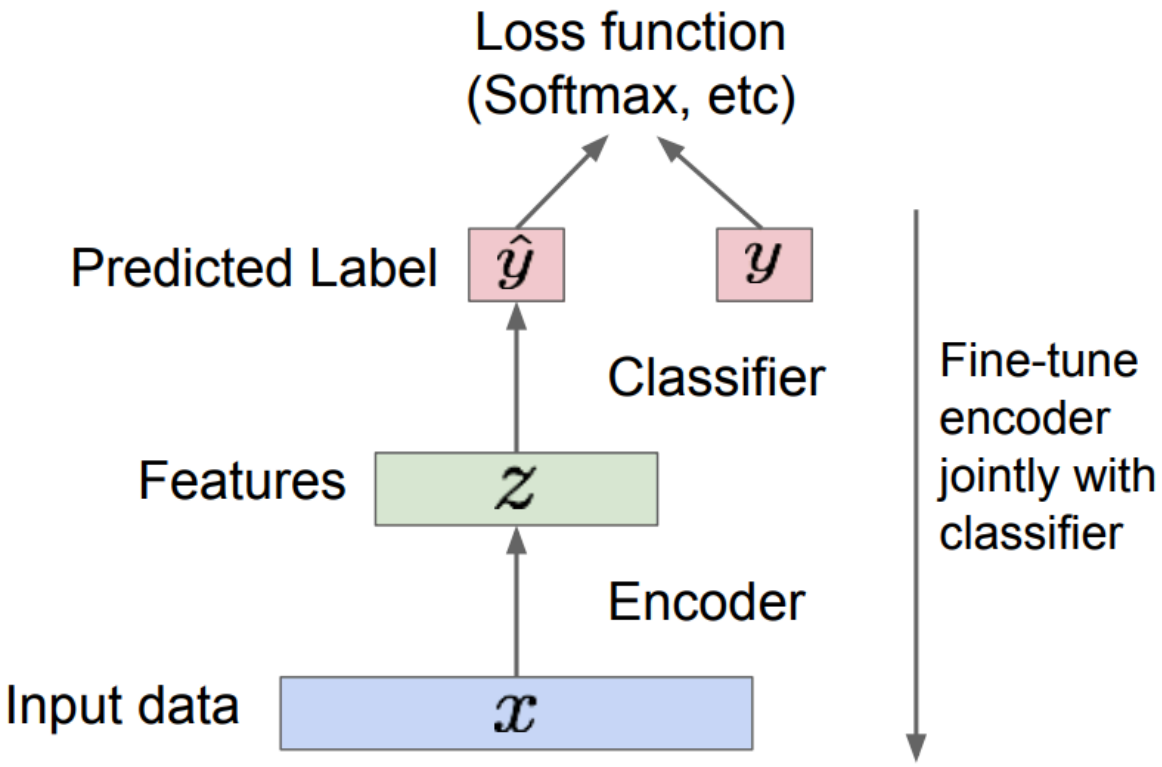
http://blog.otoro.net/2016/04/01/generating-large-images-from-latent-vectors/

Background: Autoencoders



Background: Autoencoders

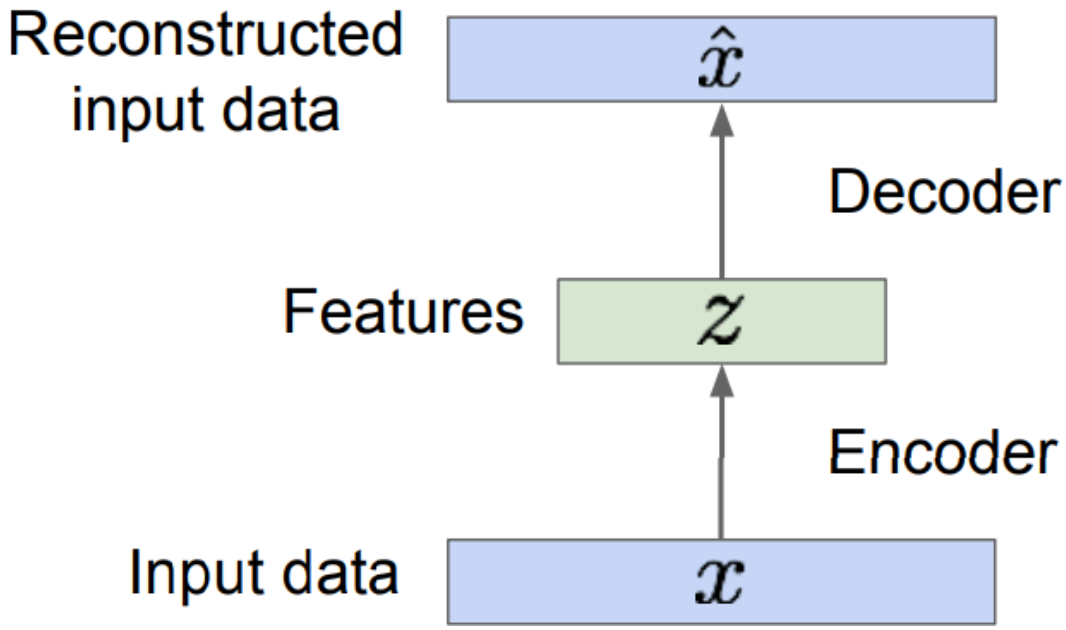
Encoder can be used to initialize a supervised model



Background: Autoencoders

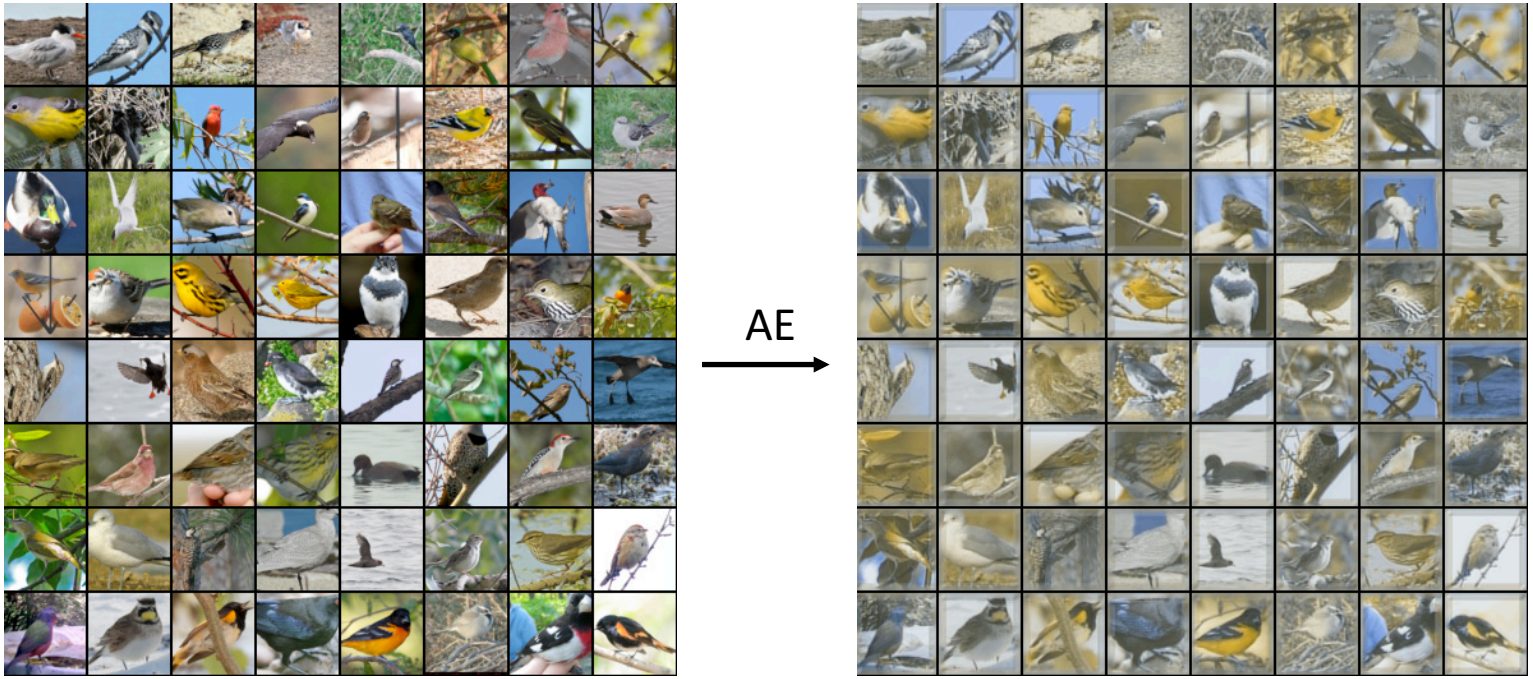
Story so far

- Autoencoders can reconstruct data, and can learn features to initialize a supervised model.
- Features capture factors of variation in training data.



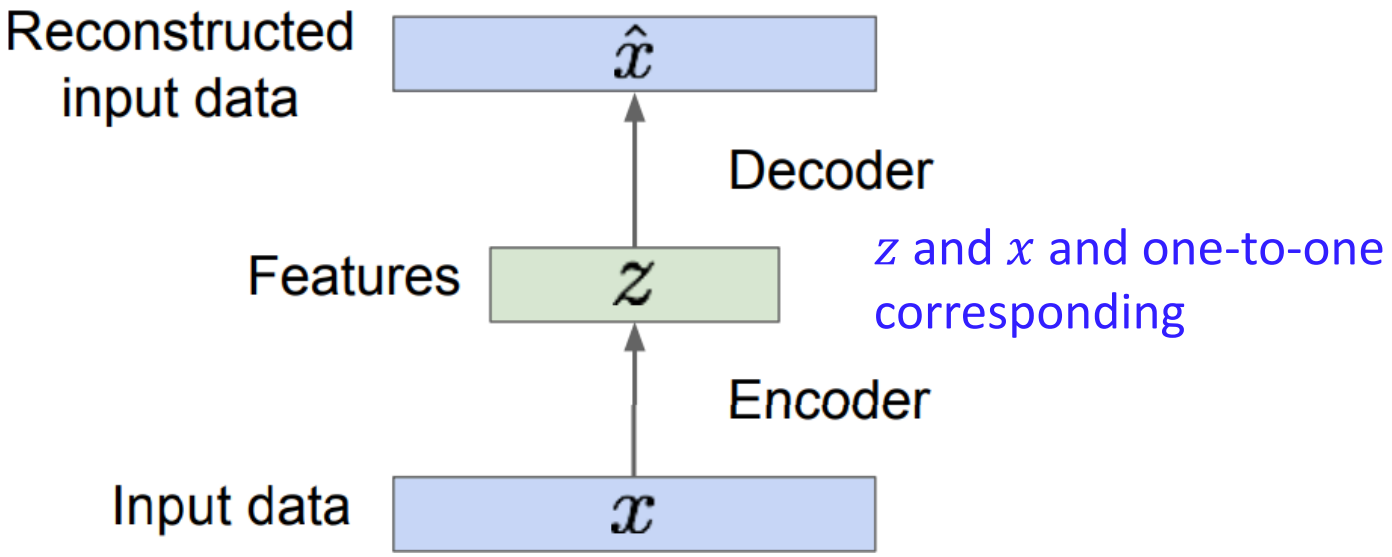
Background: Autoencoders

Can we generate new images from an autoencoder?



Variational Autoencoders

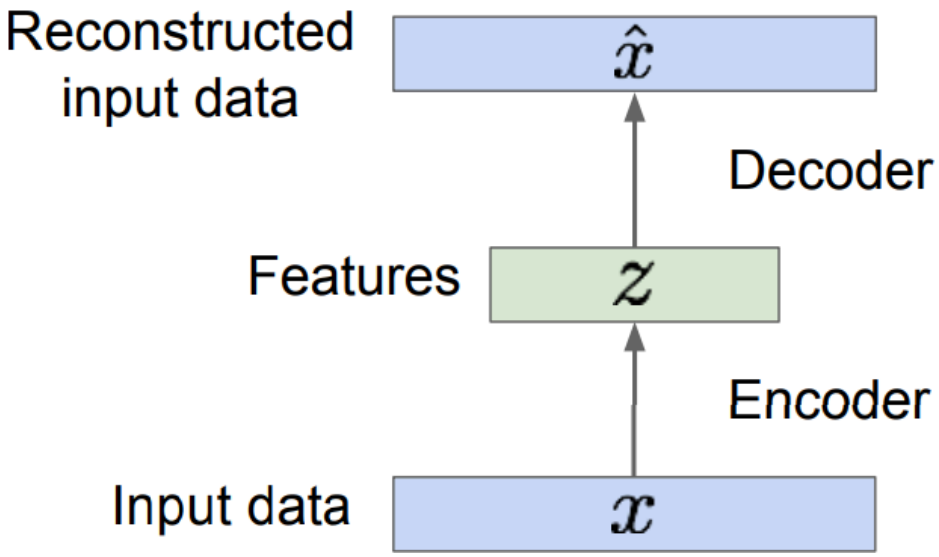
Intuition



We don't know how to create latent vectors other than encoding them from images. It is so limited!

Variational Autoencoders

Intuition

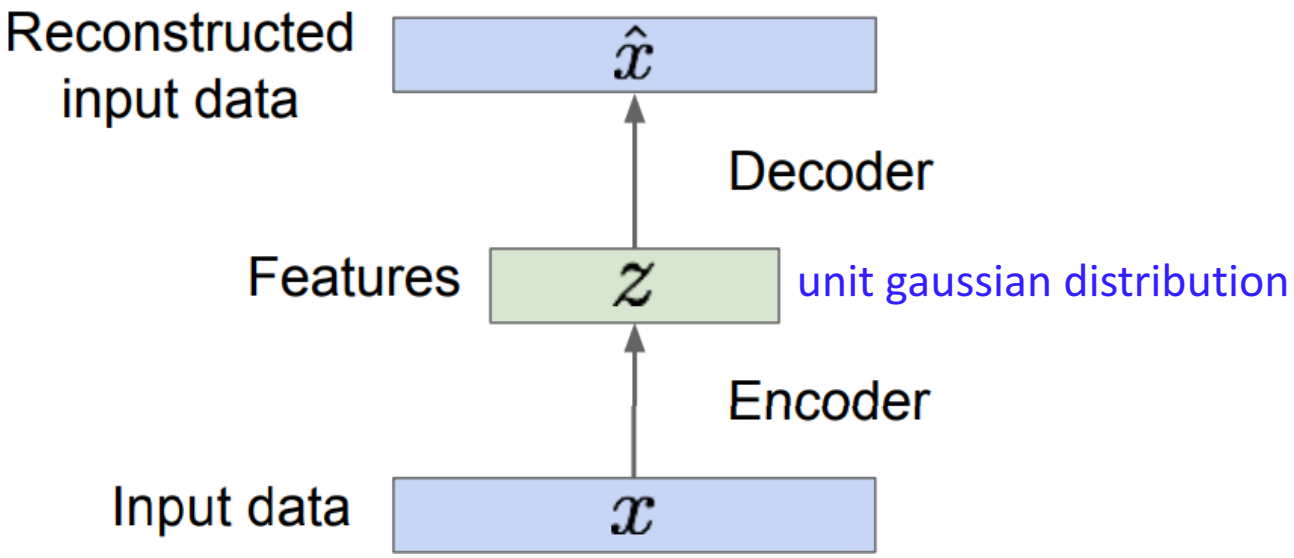


It will be better if we can sample z from some prior distribution $p(z)$, e.g., Gaussian distribution.

p.s. Why Gaussian distribution? Because it is simple and reasonable.

Variational Autoencoders

Intuition

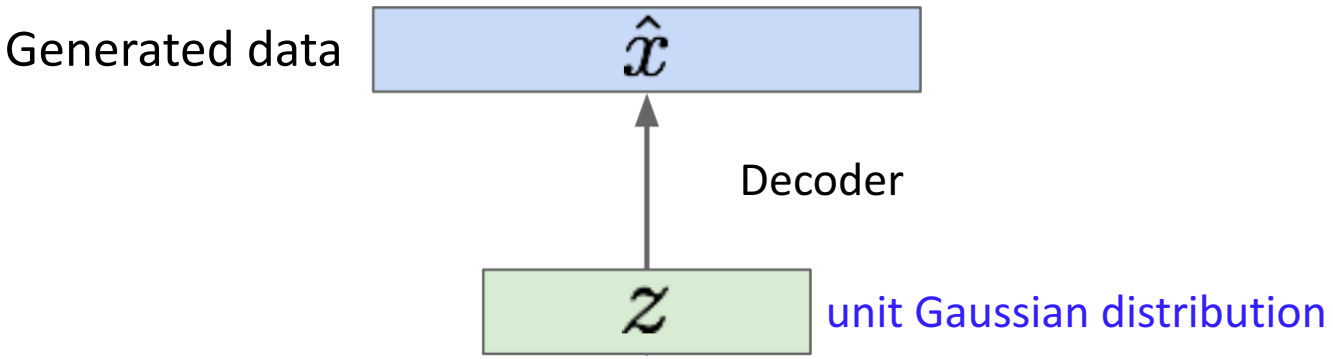


Add a constraint on the encoding network, that forces it to generate latent vectors that roughly follow a unit gaussian distribution.

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

Variational Autoencoders

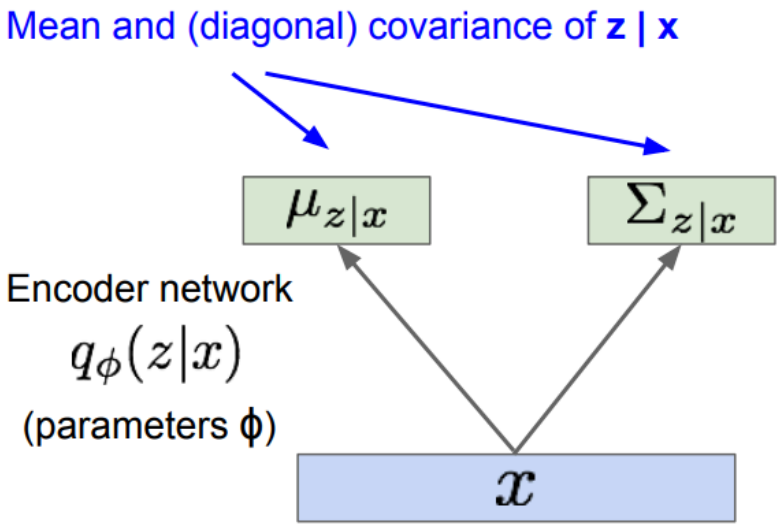
Intuition

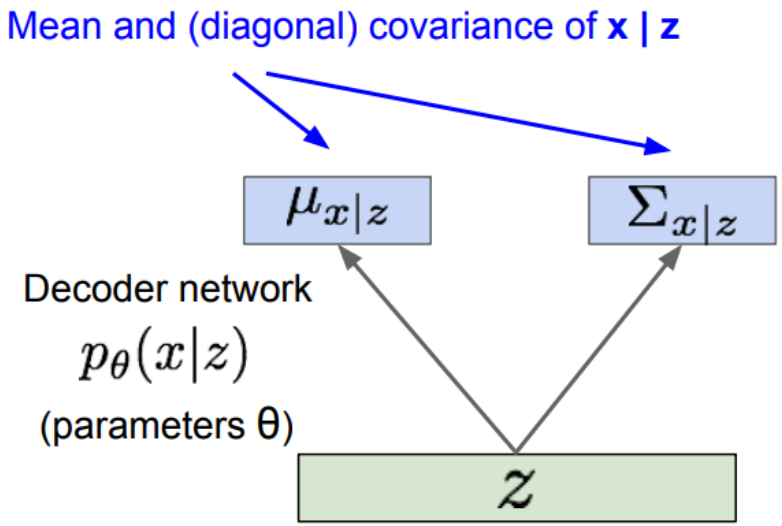


All we need to do is sample a latent vector from the unit Gaussian and pass it into the decoder. Then we will get the generated new data.

Kingma and Welling, “Auto-Encoding Variational Bayes” in ICLR, 2014

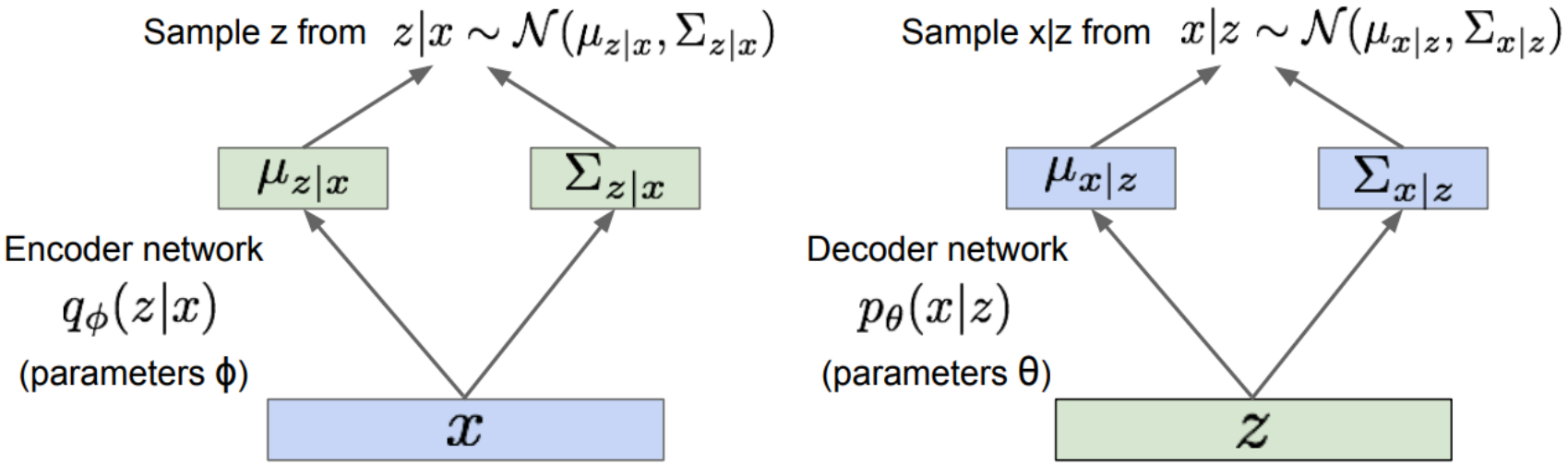
Variational Autoencoders





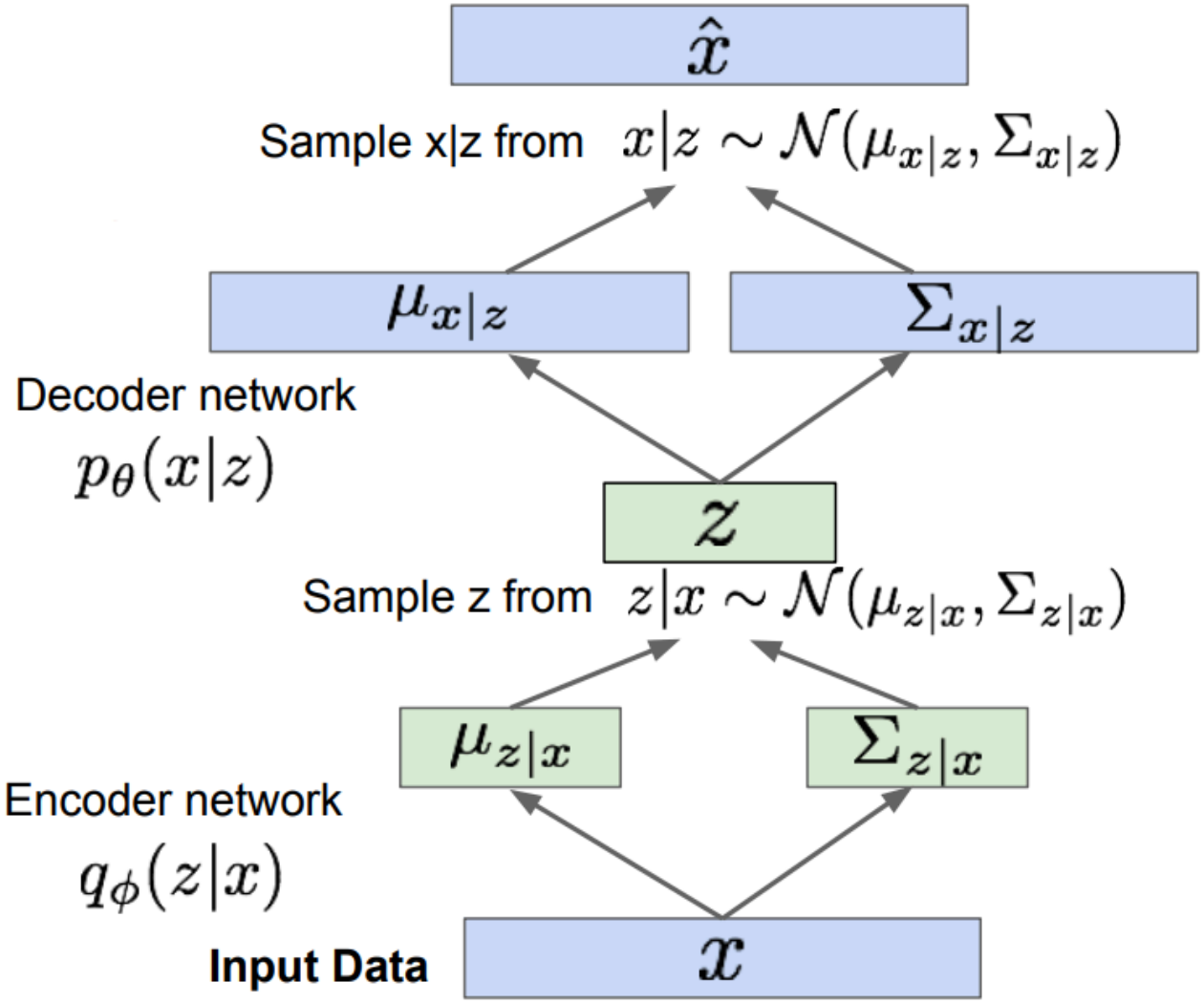
Kingma and Welling, “Auto-Encoding Variational Bayes” in ICLR, 2014

Variational Autoencoders



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

Variational Autoencoders



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

Variational Autoencoders

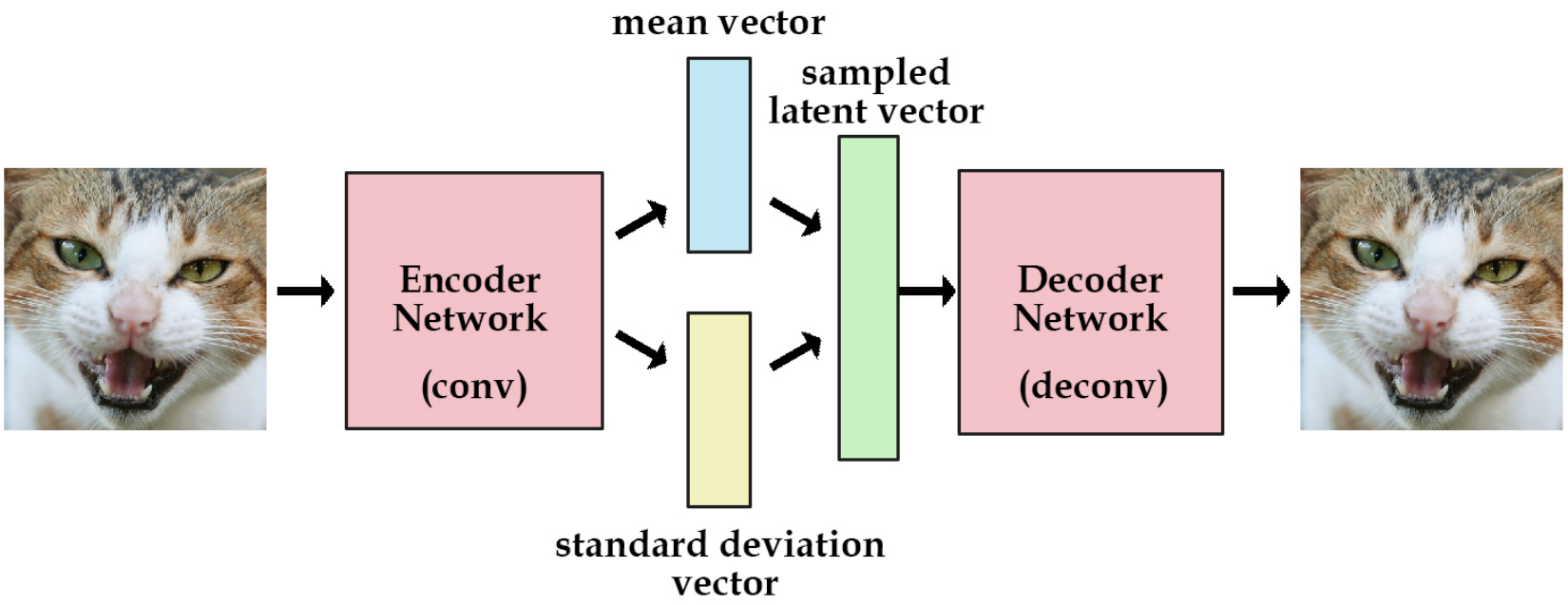


Image from http://kvfrans.com/variational-autoencoders-explained/

Variational Autoencoders

Data likelihood

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

$$= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}$$

Tractable lower bound which we can take gradient of and optimize! ($p_\theta(x|z)$ differentiable, KL term differentiable)

intractable

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

Variational Autoencoders

KL divergence (Kullback–Leibler divergence)

$$D_{KL}(p(x)||q(x)) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

Measure the distance between distribution $p(x)$ and $q(x)$

Variational Autoencoders

KL divergence (Kullback–Leibler divergence)

$$D_{KL}(p(x)||q(x)) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

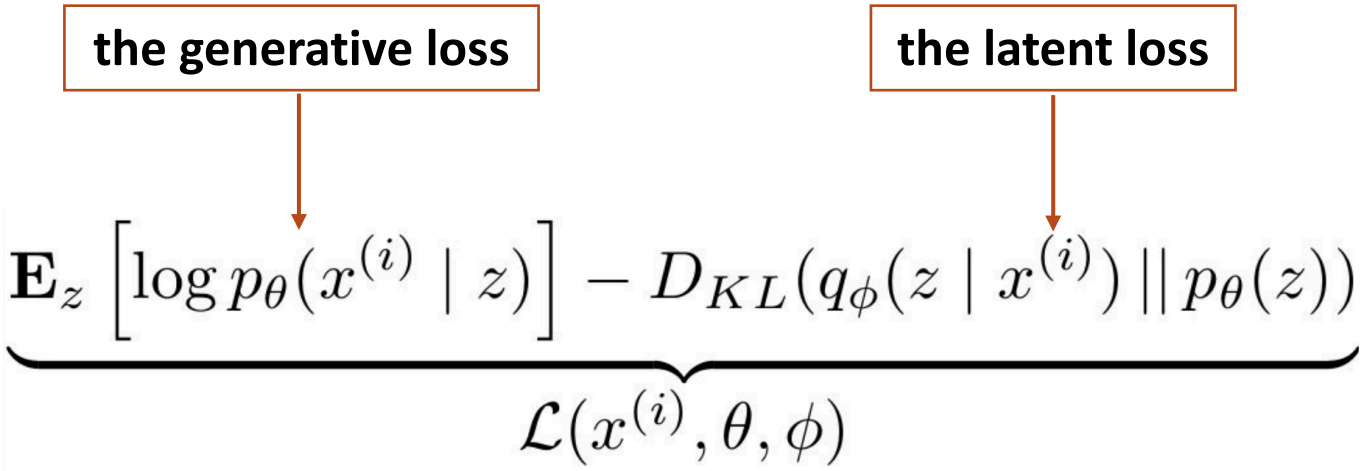
Measure the distance between distribution $p(x)$ and $q(x)$

If $p(x)$ and $q(x)$ are the same distribution, then $D_{KL}(p(x)||q(x)) = 0$

For any $p(x)$ and $q(x)$, $D_{KL}(p(x)||q(x)) \geq 0$

Variational Autoencoders

Explicit approximate density

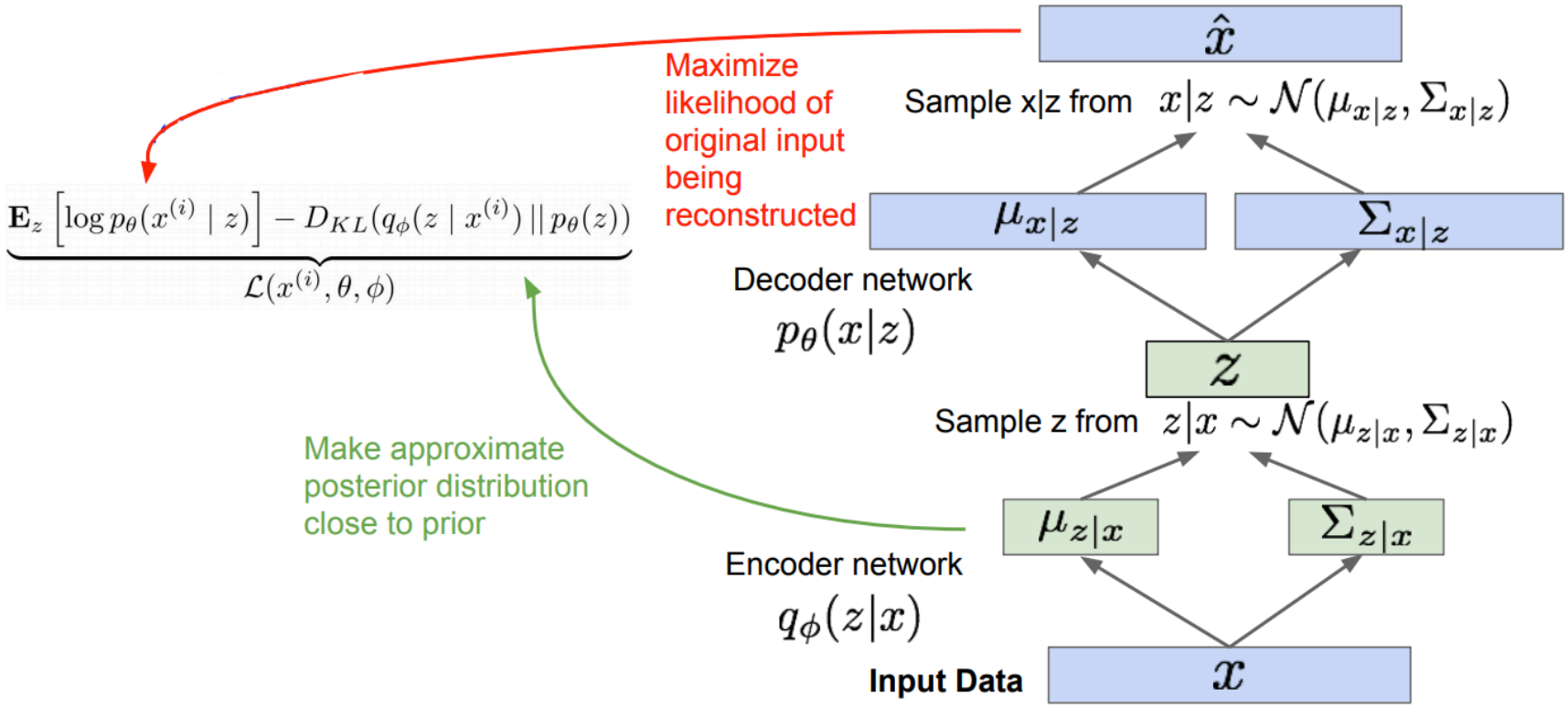


The generative loss: a mean squared error that measures how accurately the network reconstructed the images.

The latent loss: the KL divergence that measures how closely the latent variables match a unit Gaussian distribution.

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

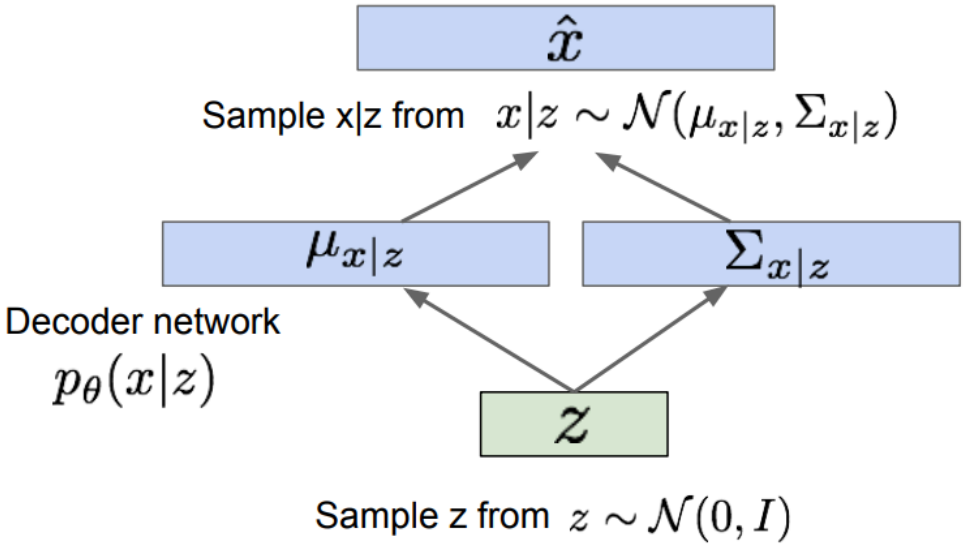
Variational Autoencoders

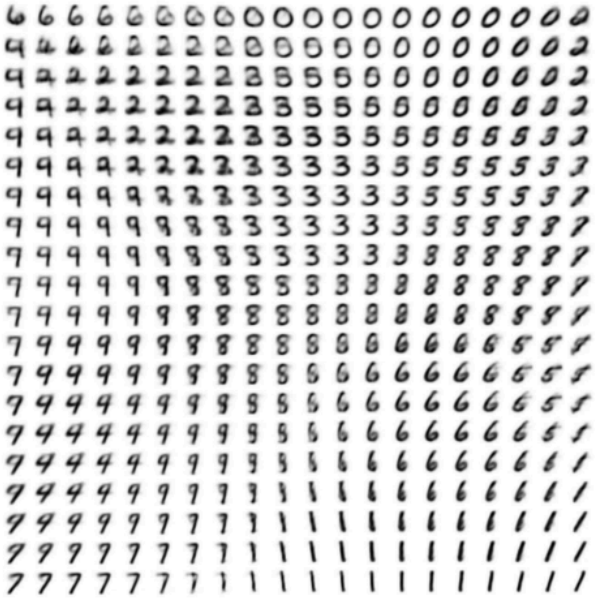


Kingma and Welling, "Auto-Encoding Variational Bayes" in ICLR, 2014

Variational Autoencoders

Use decoder network. Now sample z from prior!

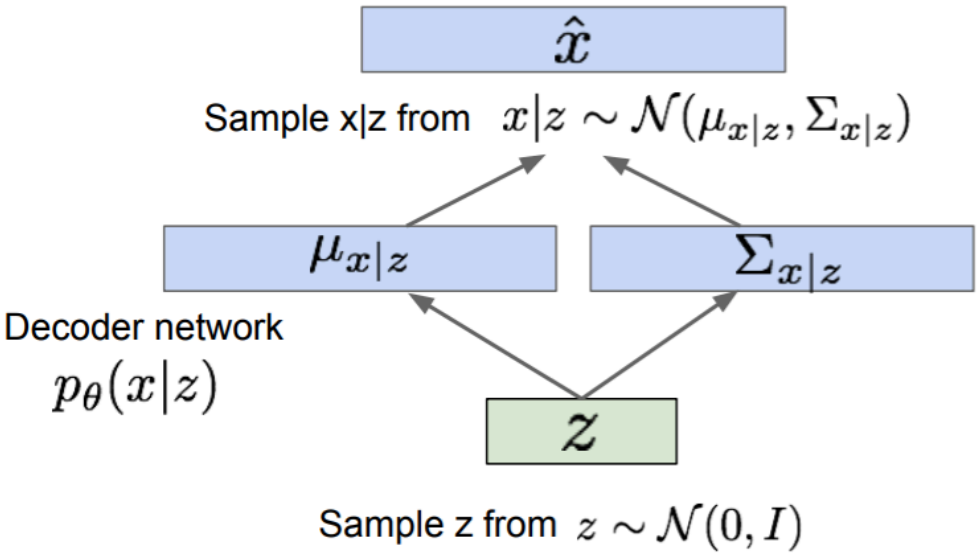




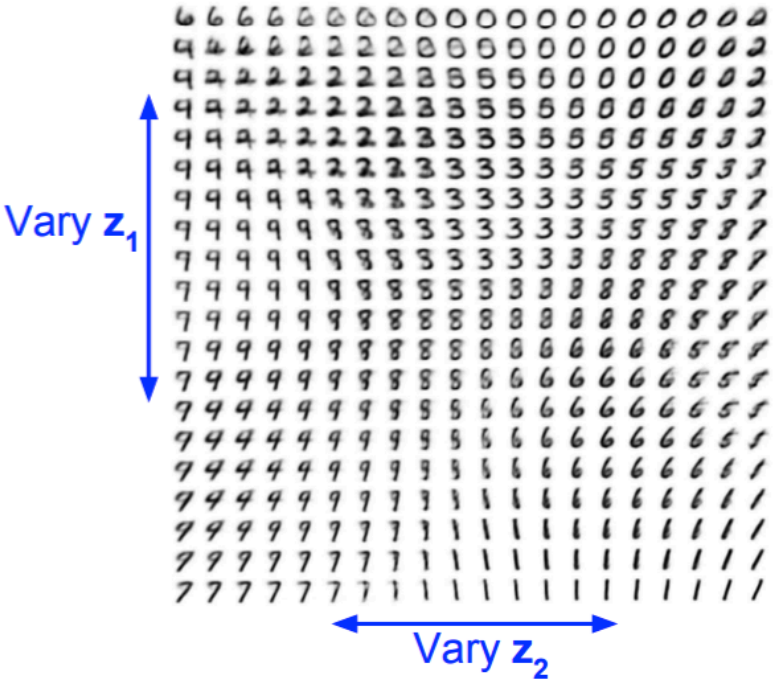
Kingma and Welling, "Auto-Encoding Variational Bayes" in ICLR, 2014

Variational Autoencoders

Use decoder network. Now sample z from prior!



Data manifold for 2-d **z**



Diagonal prior on z => independent latent variables

Different dimensions of z encode interpretable factors of variation

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

Variational Autoencoders

Diagonal prior on z
=> independent latent variables

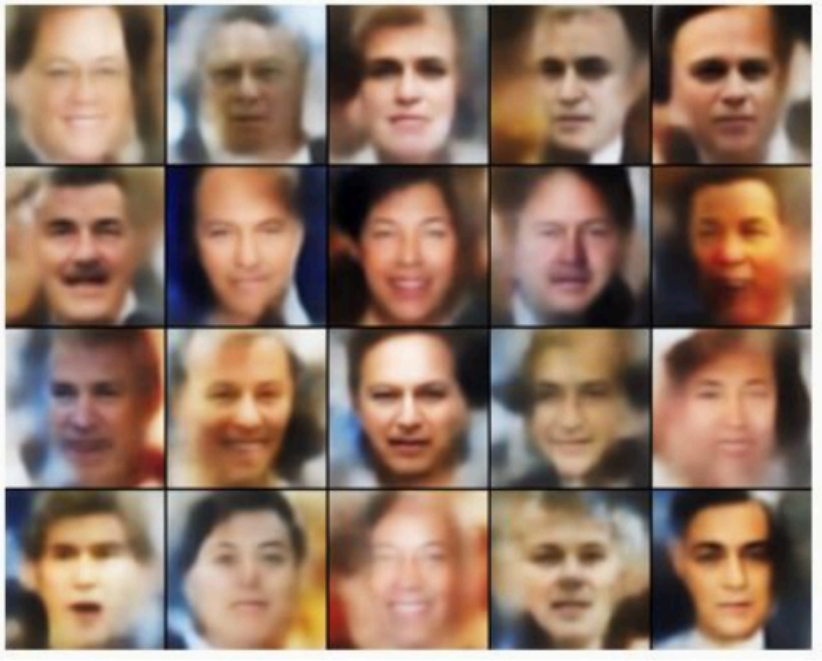
Different dimensions of z encode interpretable factors of variation

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

Variational Autoencoders

Generating data!

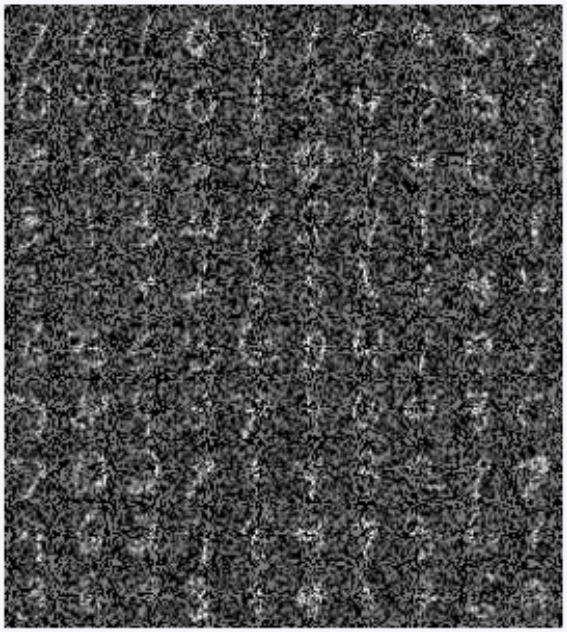
32x32 CIFAR-10



Labeled Faces in the Wild

Variational Autoencoders

Denoising

Original input image	Input image with noise	Restored image via VAE
		

https://github.com/hwalsuklee/tensorflow-mnist-VAE

Variational Autoencoders

Story so far

Defines an intractable density
=> derive and optimize a (vairational) lower bound

Probablisitic spin to traditional autoencoders
=> allows generating data

Variational Autoencoders

Story so far

Pros:

- Principled approach to generate models
- Allows inference of $q(z|x)$, can be useful feature representation for other tasks

Cons:

- Maximizes lower bound of likelihood: okay, but not as good evaluation as tractable density
- Samples blurrier and lower quality compared to state-of-the-art (GANs)

Active areas of research:

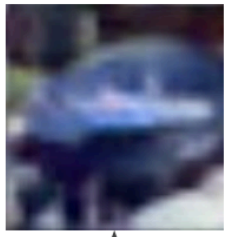
- More flexible approximations, e.g. richer approximate posterior instead of diagonal Gaussian
- Incorporating structure in latent variables

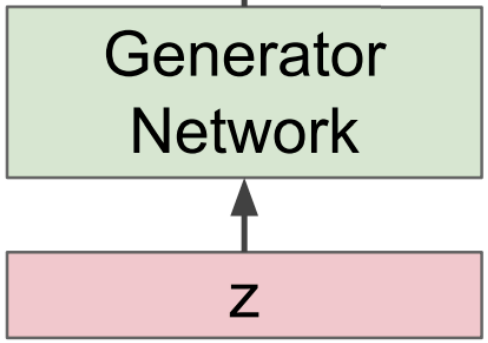
Generative Adversarial Networks (GAN)

Generative Adversarial Networks

Try a new way

Output: Sample from training distribution





Input: Random noise

Generative Adversarial Networks

Ian Goodfellow & his *Deep Learning* book

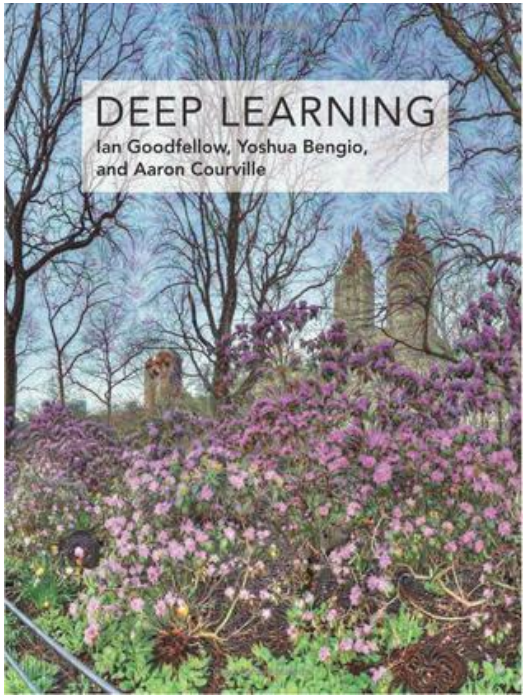
Ian Goodfellow

Image from http://www.sohu.com/a/224635709_473283

Generative Adversarial Networks

Ian Goodfellow & his book

Ian Goodfellow



Deep Learning book

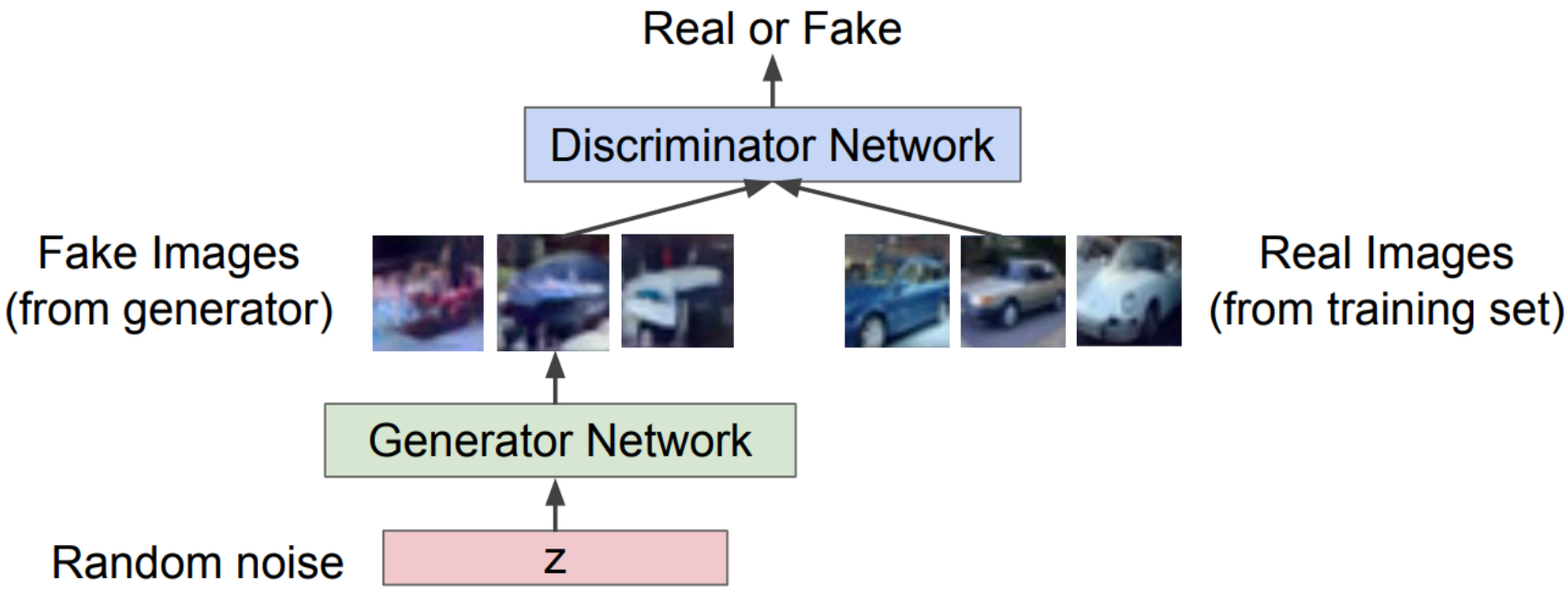
Image from http://www.sohu.com/a/224635709_473283

Generative Adversarial Networks

Architecture

Generator network: try to fool the discriminator by generating real-looking images

Discriminator network: try to distinguish between real and fake images



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Generative Adversarial Networks

Just like a counterfeiter-police game

Poster of film “Catch Me If You Can”, 2002

Generative Adversarial Networks

Just like a counterfeiter-police game

A counterfeiter-police game between two components:
a generator **G** and a discriminator **D**

G: counterfeiter, trying to fool police with fake currency

D: police, trying to detect the counterfeit currency

Competition drives both to improve, until counterfeits are *indistinguishable* from genuine currency

Generative Adversarial Networks

Training process: two-player game

Train jointly in minimax game

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Ian Goodfellow et al., “Generative Adversarial Nets” in NIPS, 2014

Generative Adversarial Networks

Training process: two-player game

Train jointly in minimax game

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

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Discriminator output for real data x

Discriminator output for generated fake data G(z)

Ian Goodfellow et al., "Generative Adversarial Nets" in NIPS, 2014

Generative Adversarial Networks

Training process: two-player game

Train jointly in minimax game

Minimax objective function:

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

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Discriminator output for real data x

Discriminator output for generated fake data G(z)

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)

Ian Goodfellow et al., "Generative Adversarial Nets" in NIPS, 2014

Generative Adversarial Networks

Simultaneous updates

- Both the generator and discriminator are trying to learn “moving targets”. Both networks are trained simultaneously.

- The discriminator needs to update based on how well the generator is doing. The generator is constantly updating to improve performance on the discriminator. These two need to be balanced correctly to achieve stable learning instead of chaos.

Generative Adversarial Networks

Stationary point - Nash equilibrium

There is a theoretical point in this game at which the game will be stable and both players will stop changing.

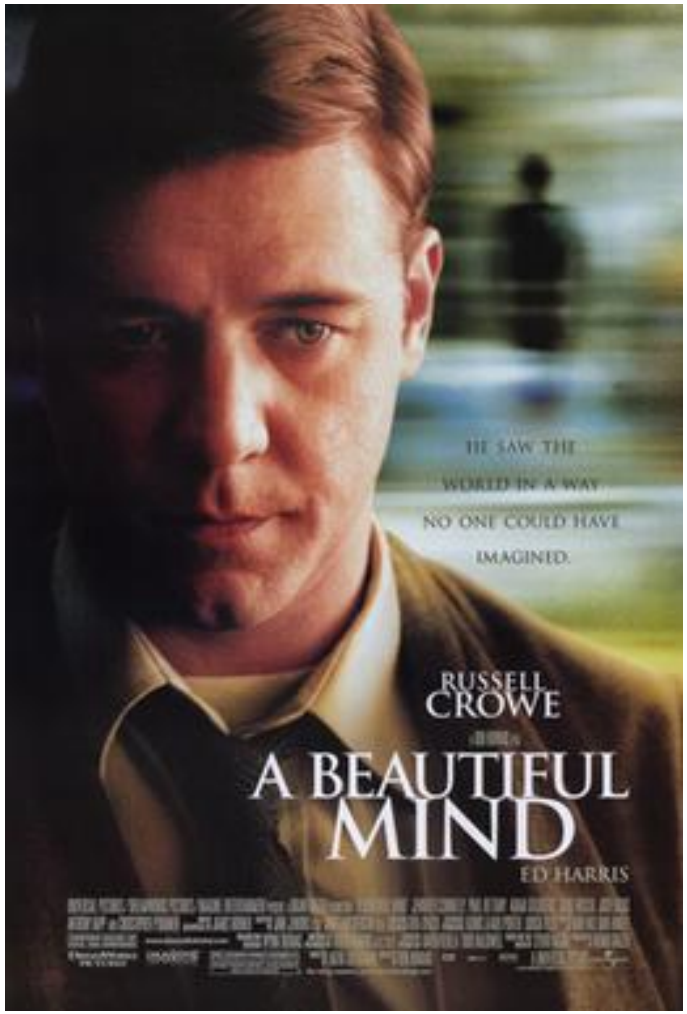
- $p_{model} = p_{data}$

  If the generated data exactly matches the distribution of the real data, the generator should output 0.5 for all points (argmax of loss function)

- $D_{\theta_d}(\mathrm{x}^*) = \mathrm{Constant}$

  If the discriminator is outputting a constant value for all inputs, then there is no gradient that should cause the generator to update

John Nash

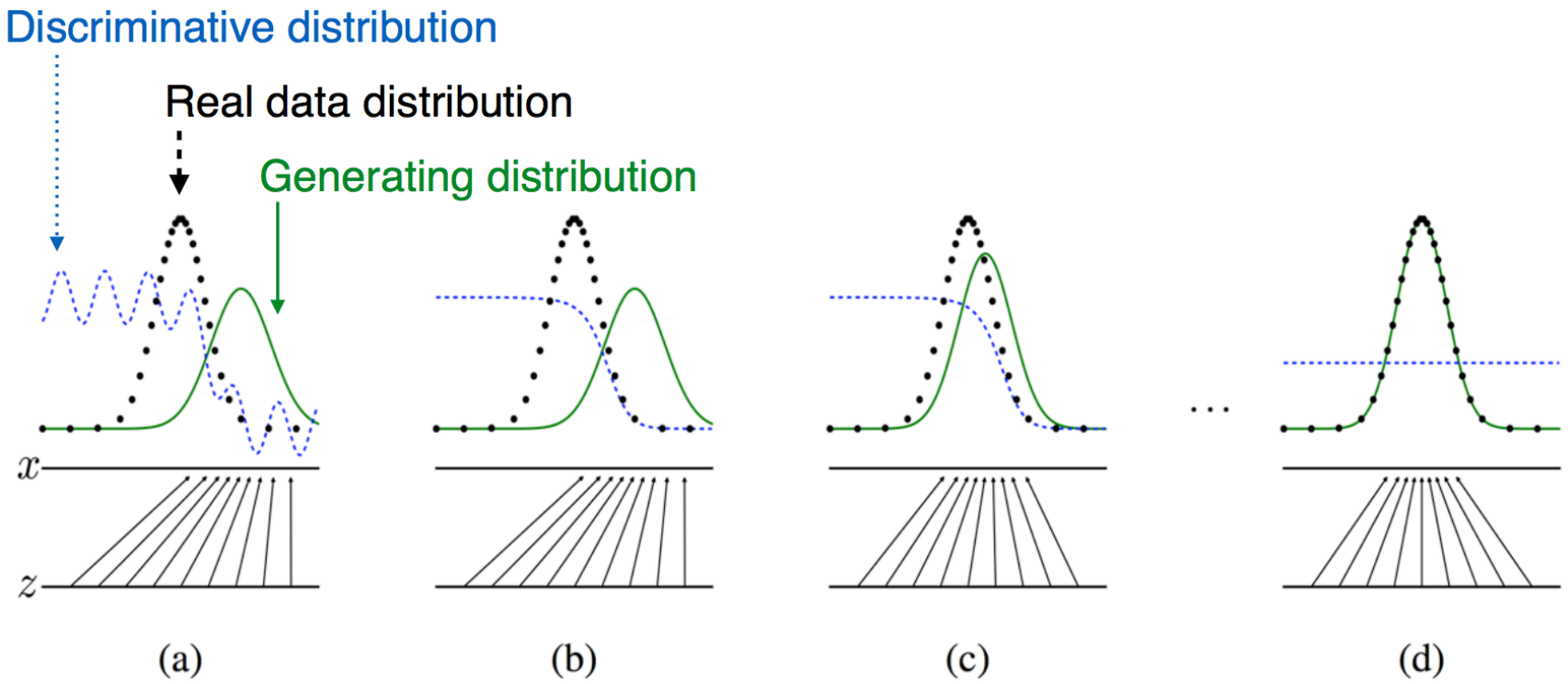


A Beautiful Mind

Poster of film “A Beautiful Mind”, 2001

Generative Adversarial Networks

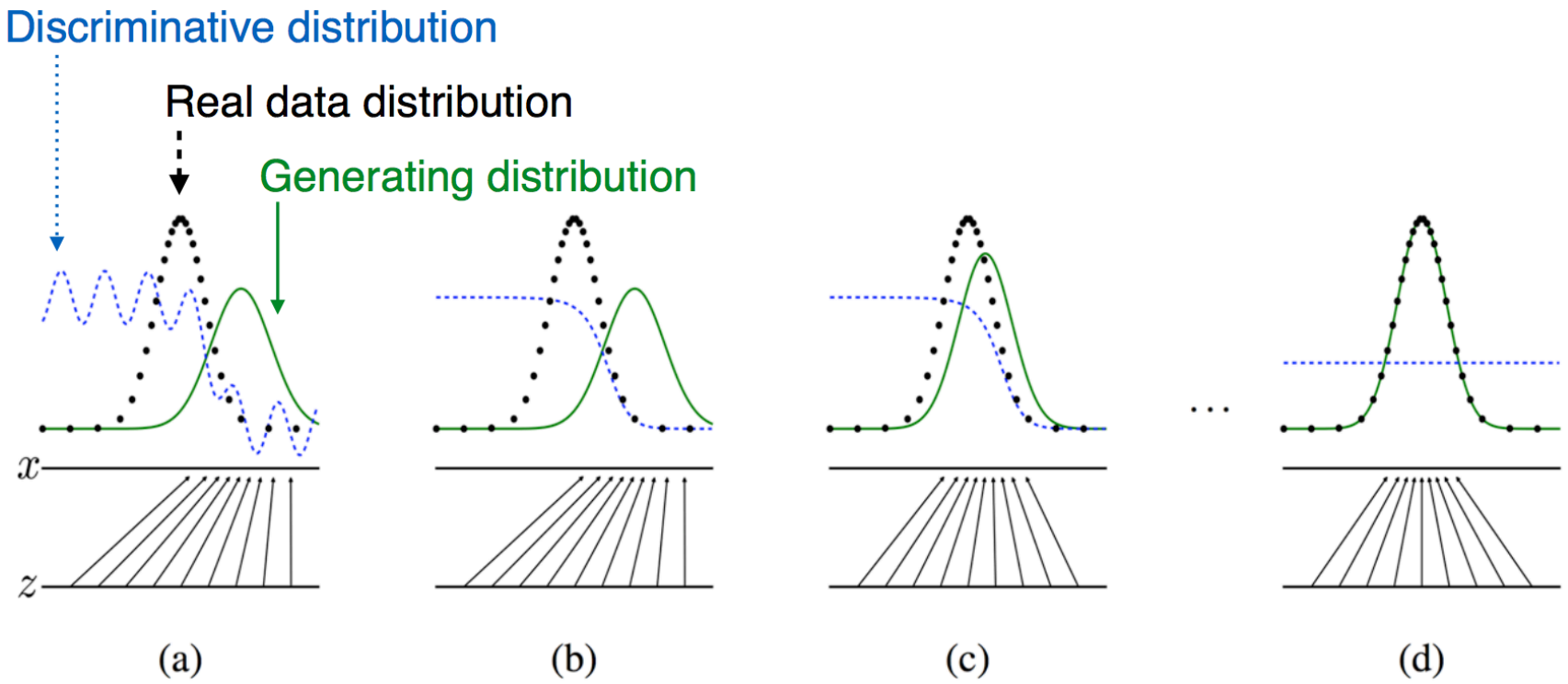
Optimization process



Ian Goodfellow et al., “Generative Adversarial Nets” in NIPS, 2014

Generative Adversarial Networks

Optimization process

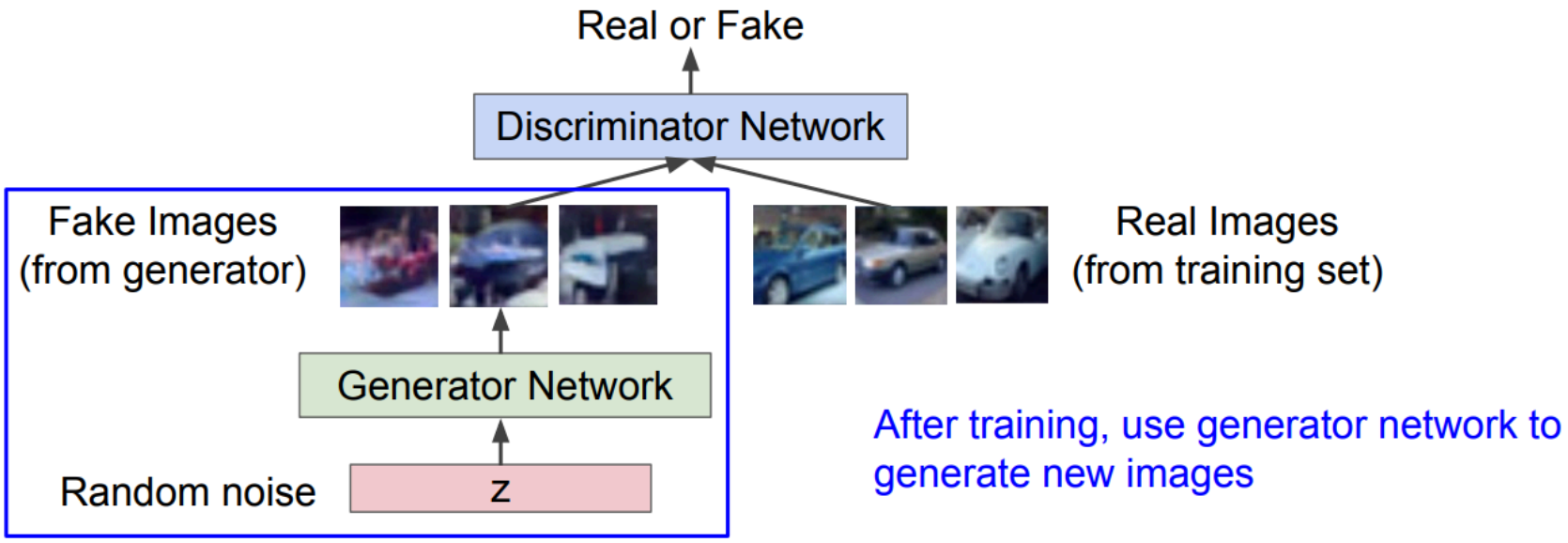


We rarely reach a completely stable point in practice due to practical issues

Ian Goodfellow et al., “Generative Adversarial Nets” in NIPS, 2014

Generative Adversarial Networks

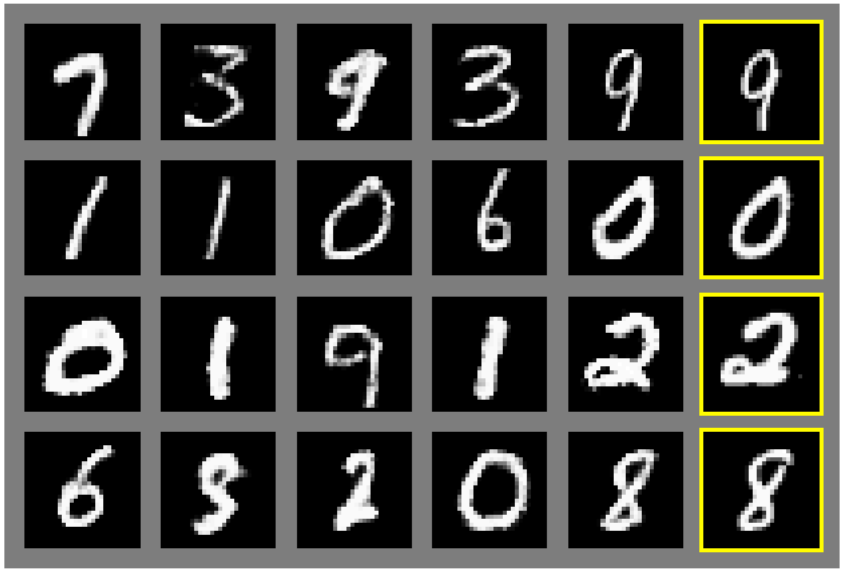
Generate new images



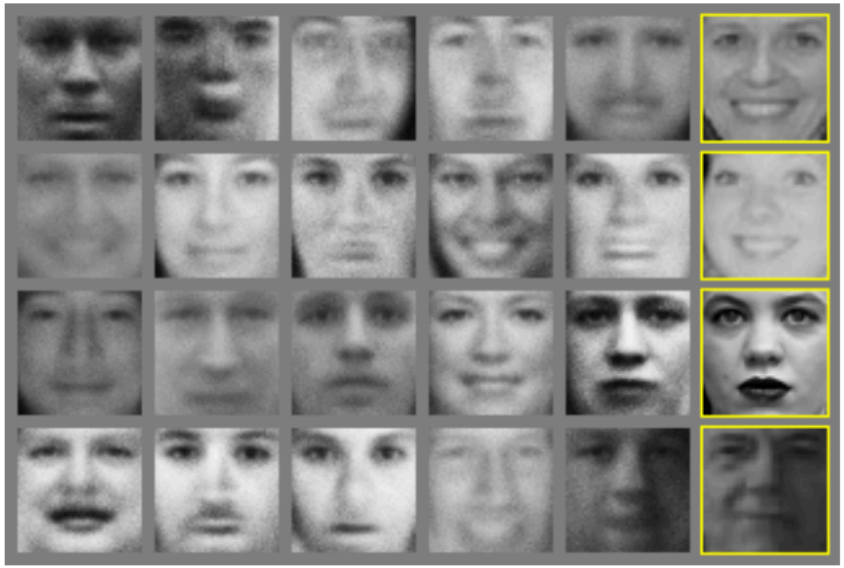
Ian Goodfellow et al., "Generative Adversarial Nets" in NIPS, 2014

Generative Adversarial Networks

Generated samples



MNIST



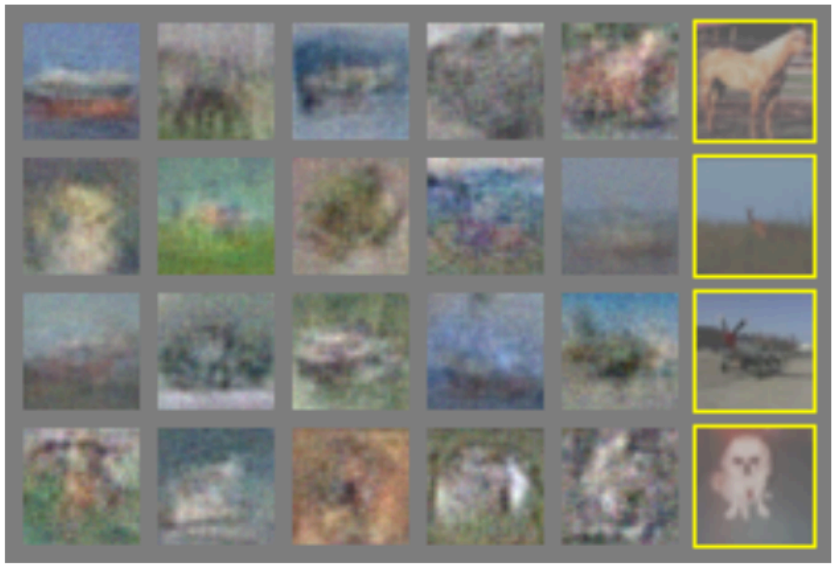
TFD

Nearest neighbor from training set

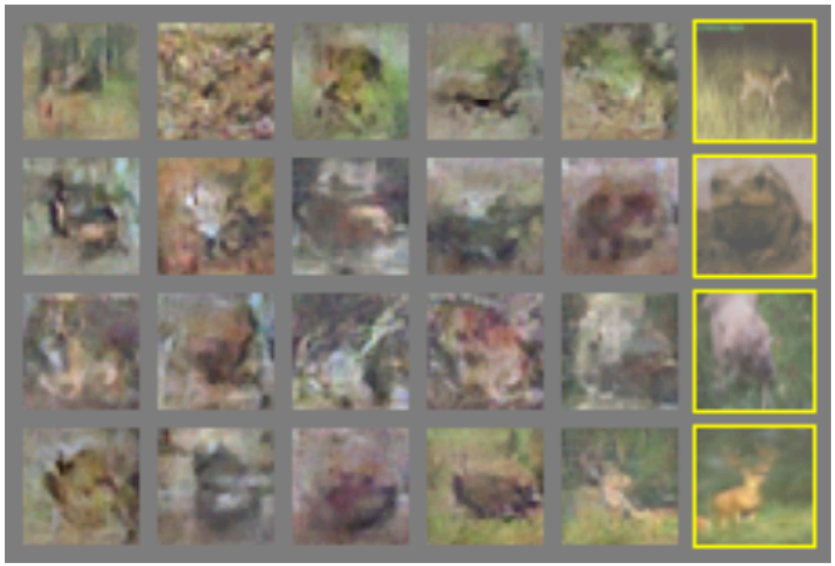
Ian Goodfellow et al., “Generative Adversarial Nets” in NIPS, 2014

Generative Adversarial Networks

Generated samples



CIFAR-10 (FC)



CIFAR-10 (CNN)

Nearest neighbor from training set

Ian Goodfellow et al., “Generative Adversarial Nets” in NIPS, 2014

Generative Adversarial Networks

What is the latent space like?

You can interpolate along the hidden space to produce smooth transitions of images.

Key differences between VAEs and GANs

- VAEs are more theoretically grounded than GANs. GANs are more based on what works.
- GANs traditionally only learn the decoder (generator) but there are variations that learn an encoder as well; there are some problems where you want both and some problems where just the decoder will suffice. VAEs learn an encoder/decoder pair.
- GAN decoder sees samples from prior $q(z)$, VAE decoder sees samples from model $p(z \mid x)$.

Most important difference between VAEs and GANs

"Distance":

- VAE objective for the decoder is some man-made objective function, like L2 distance between images
- GAN objective for the generator is some complicated objective function defined by a neural network

Most important difference between VAEs and GANs

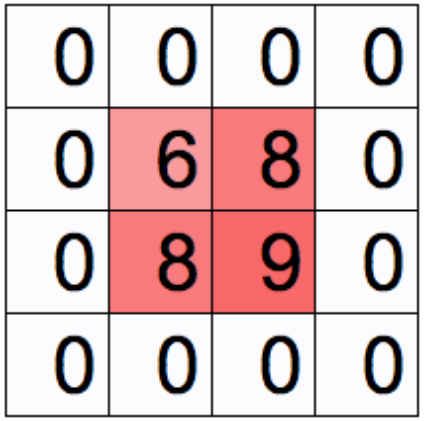
"Distance":

- VAE objective for the decoder is some man-made objective function, like L2 distance between images
- GAN objective for the generator is some complicated objective function defined by a neural network

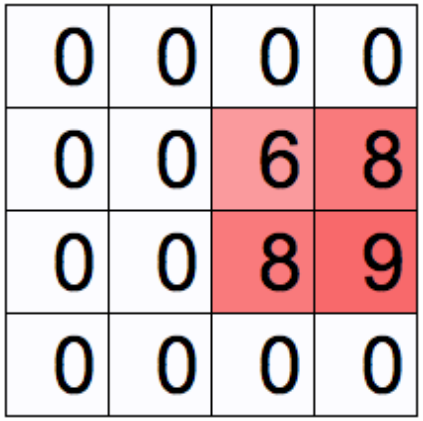
Use neural networks to define the metric!

Loss compared to VAEs

Consider when the decoder (generator) generate an image with some slight shift



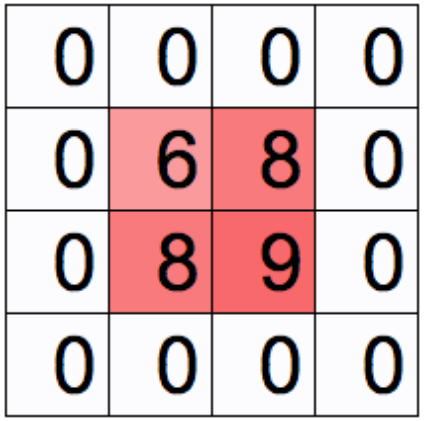
Original



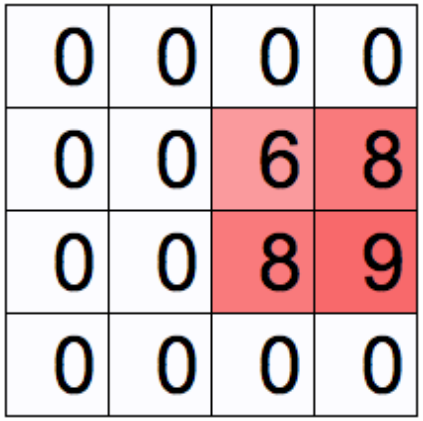
Generated

Loss compared to VAEs

For a VAE:



Original



Generated

$$L2\ distance = (6-0)^2 + (8-6)^2 + (0-8)^2 + (8-0)^2 + (9-8)^2 + (0-9)^2$$
$$= 250$$

Loss compared to VAEs

For a VAE:

0	0	0	0
0	6	8	0
0	8	9	0
0	0	0	0

Original

0	0	0	0
0	0	6	8
0	0	8	9
0	0	0	0

Generated

L2 = 250

$$L2\ distance = (6-0)^2 + (8-6)^2 + (0-8)^2 + (8-0)^2 + (9-8)^2 + (0-9)^2$$
$$= 250$$

Loss compared to VAEs

For a VAE:

0	0	0	0
0	6	8	0
0	8	9	0
0	0	0	0

Original

0	0	0	0
0	0	6	8
0	0	8	9
0	0	0	0

Generated
L2 = 250

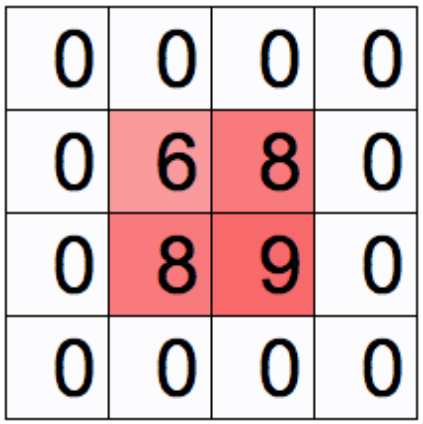
0	0	0	0
0	3	7	4
0	4	9	5
0	0	0	0

Generated

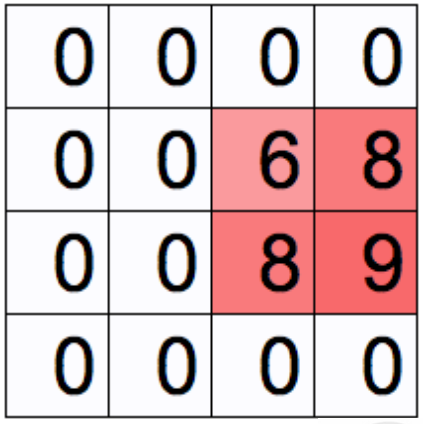
$$L2\ distance = (6-3)^2 + (8-7)^2 + (0-4)^2 + (8-4)^2 + (9-9)^2 + (0-5)^2$$
$$= 67.3$$

Loss compared to VAEs

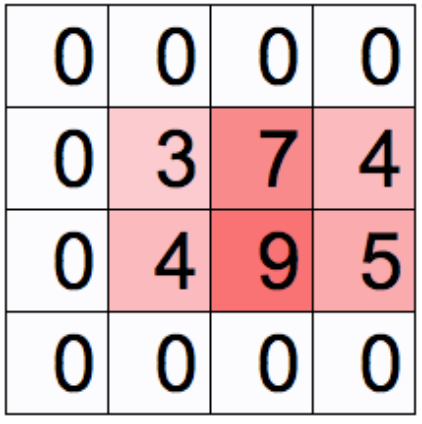
For a VAE:



Original



Generated
L2 = 250

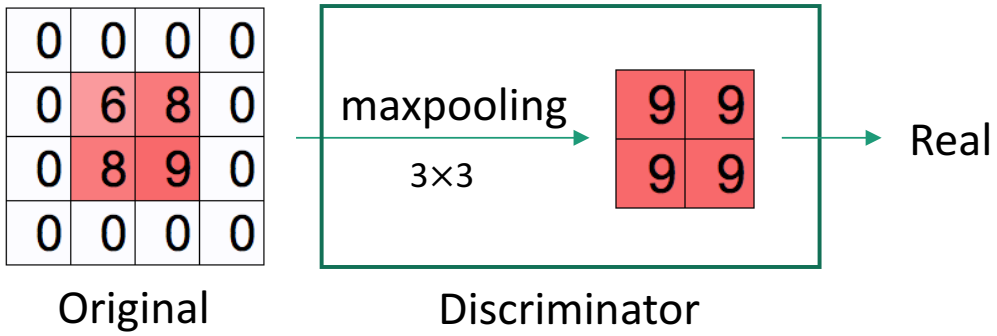


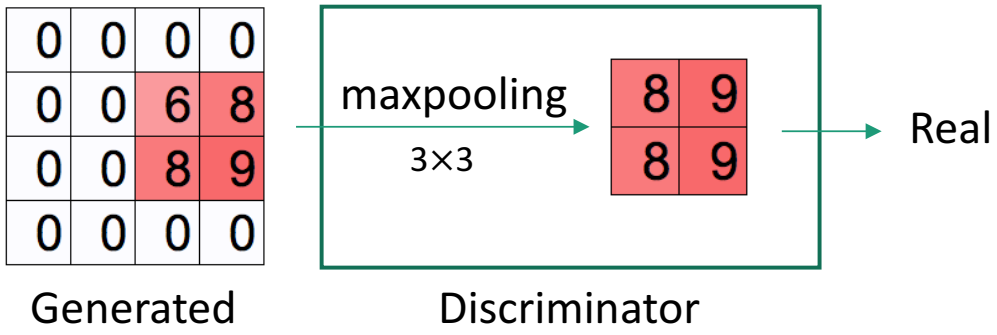
Generated
L2 = 67.3

$$L2\ distance = (6-3)^2 + (8-7)^2 + (0-4)^2 + (8-4)^2 + (9-9)^2 + (0-5)^2$$
$$= 67.3$$

Loss compared to VAEs

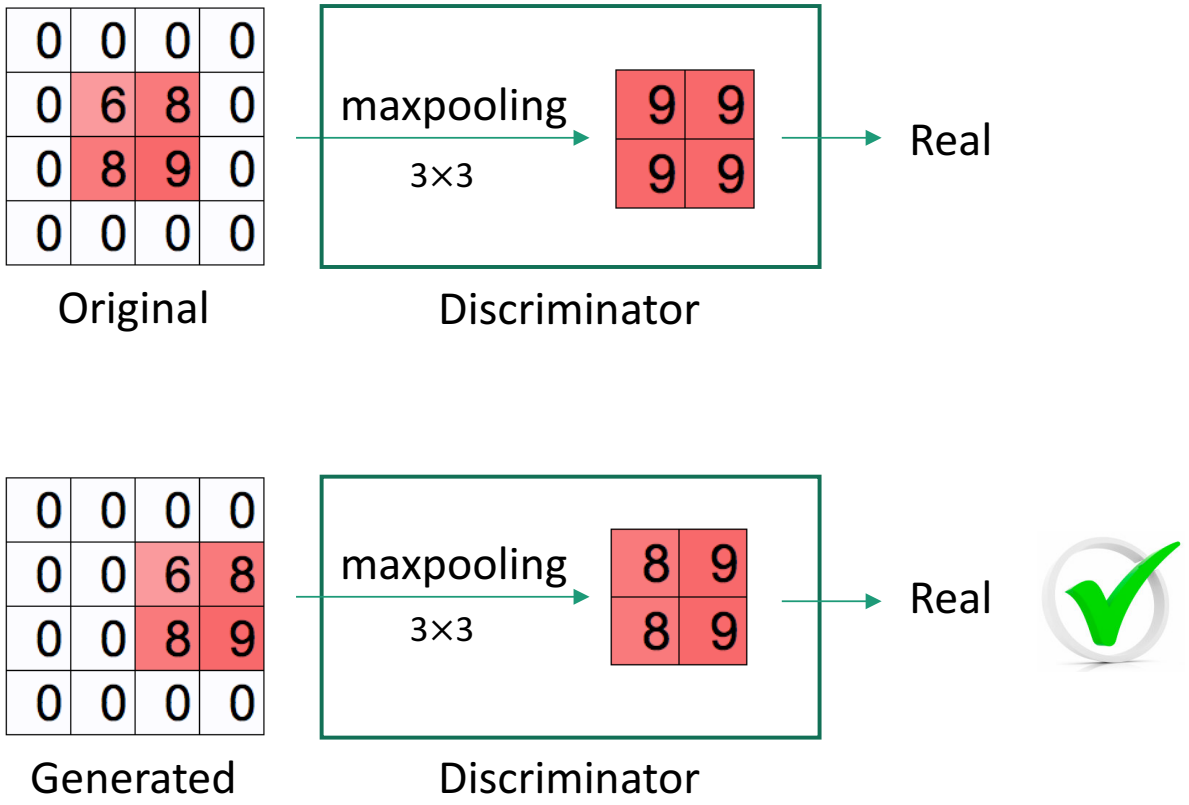
For a GAN:





Loss compared to VAEs

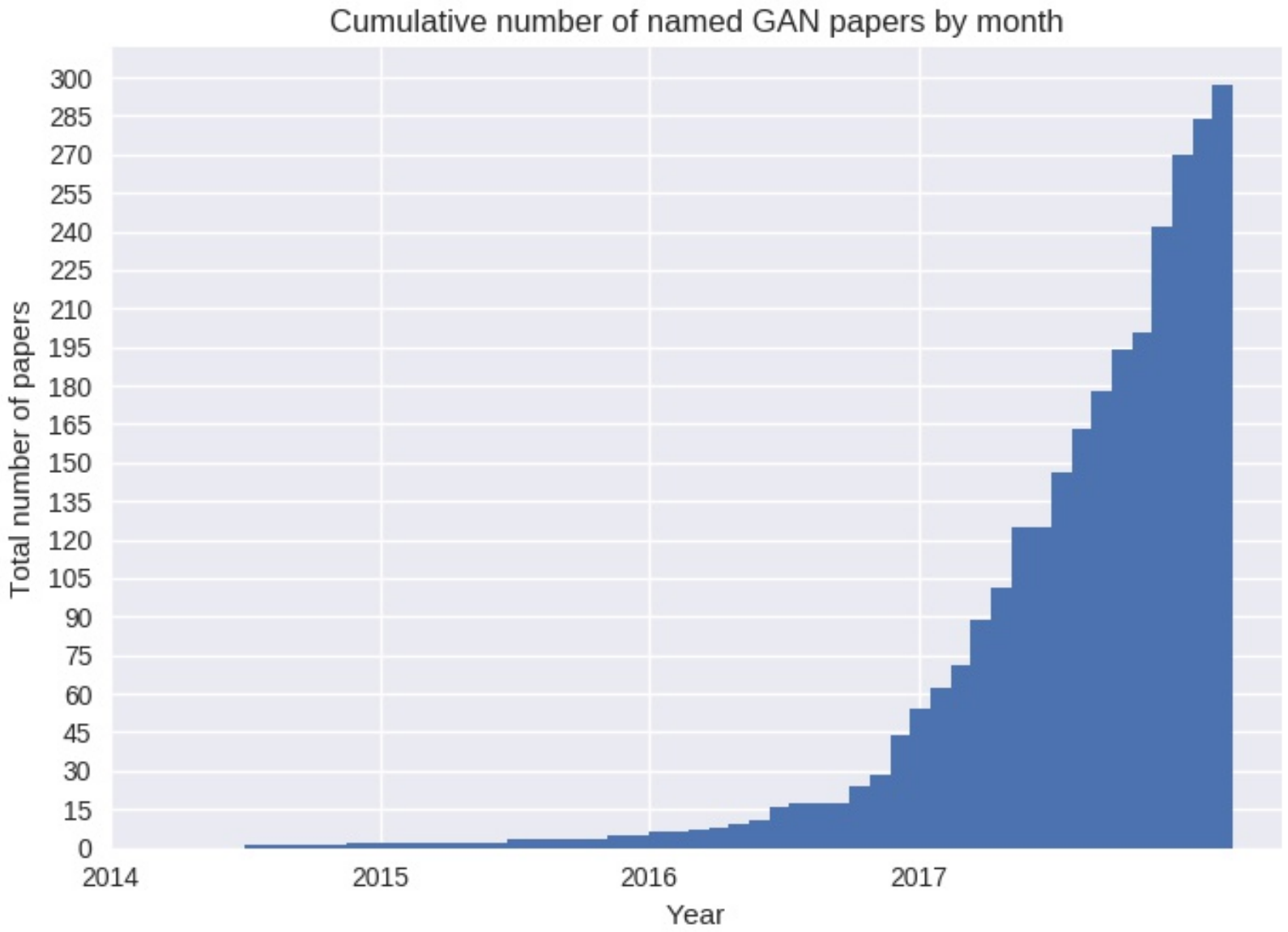
For a GAN:



Loss compared to VAEs

- In a VAE, the L2 distance makes models produce roughly average images (look blurry)
- The L2 distance is not a measure of how perceptually similar two things are.
- A neural network, with the right architecture, is arguably the definition of perceptual similarity (assuming our visual system is some sort of neural network).

Generative Adversarial Networks



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

2017: Year of GAN

To be continued …