Generative Models

(Many figures adapted from Stanford CS231n, MIT 6.S191, and Illinois CS 498)

Outline

Introduction

Variational Autoencoders (VAEs)

- Autoencoders

Generative Adversarial Networks (GANs)

Summary

Introduction (1/3)

Supervised vs unsupervised learning

Supervised Learning

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

Goal: Learn function to map
$$x \rightarrow y$$

Examples: Classification, regression, object detection, semantic segmentation, etc.

Unsupervised Learning

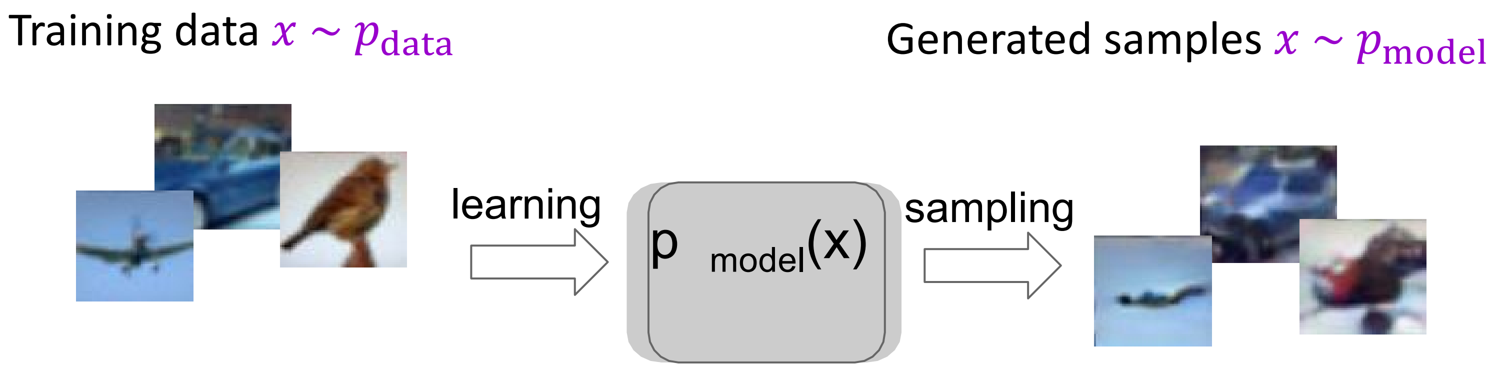
Data: x
x is data, no labels!

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

Examples: Clustering, feature or dimensionality reduction, etc.

Introduction (2/3)

- What are Generative Models?
  - Generative models are an Unsupervised Learning approach
  - Given training data, generate new samples from same distribution



We want to learn p_{model} that matches p_{data}

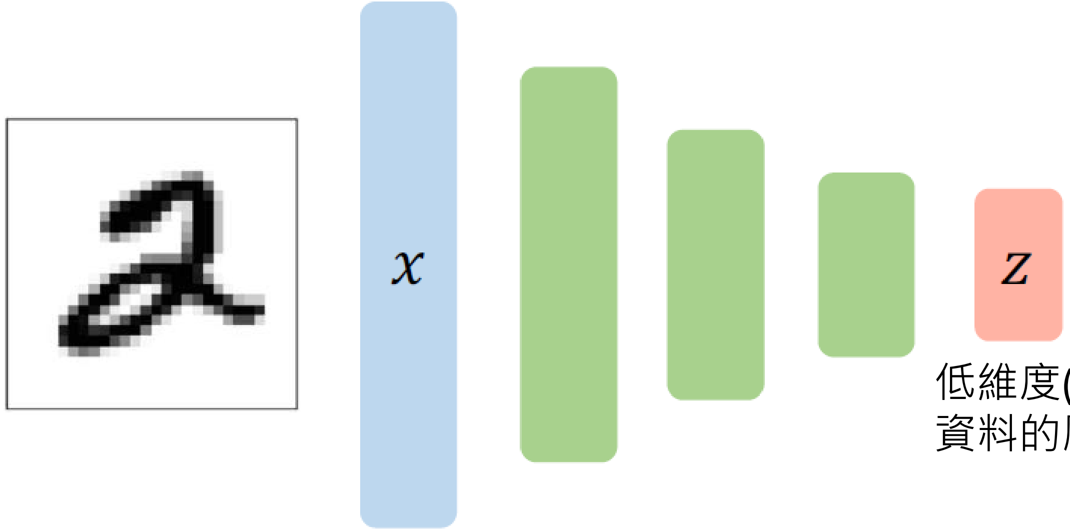
Introduction (3/3)

- Why generative models?
  - Debiasing
  - Outlier detection
  - and more…
- Introduce two most popular types of generative models today
  - Variational Autoencoders (VAEs)
  - Generative Adversarial Networks (GANs)

Autoencoders: Background (1/7)

我沒有告訴他怎麼編碼，只是要求他從高維度變成低維度

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



低維度(好像資料的壓縮)

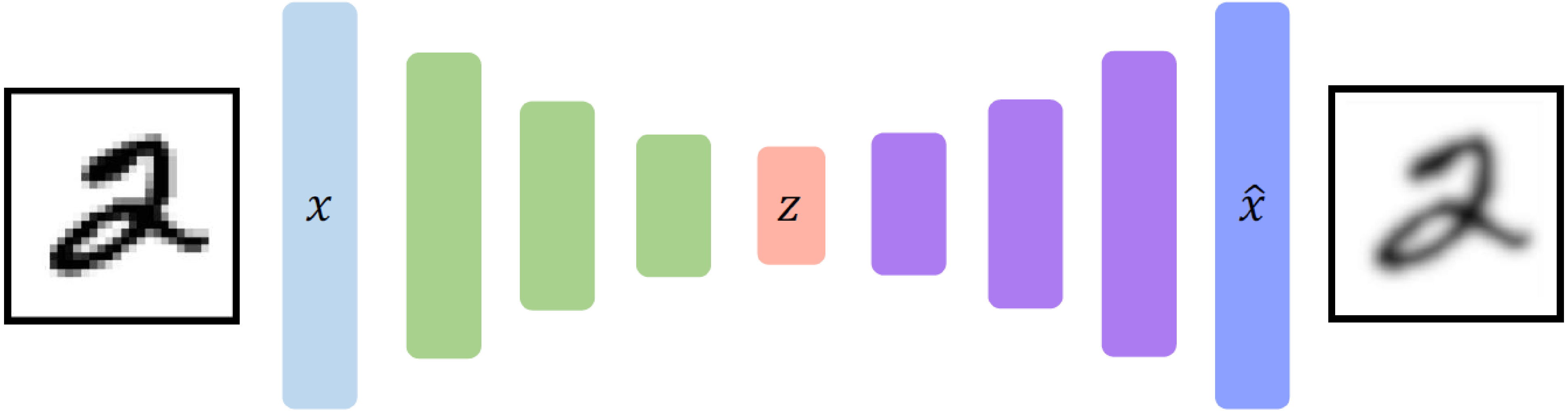
Why do we care about a low-dimensional z?

latent 隱藏、潛在

"Encoder" learns mapping from the data, x, to a low-dimensional latent space, z

Autoencoders: Background (2/7)

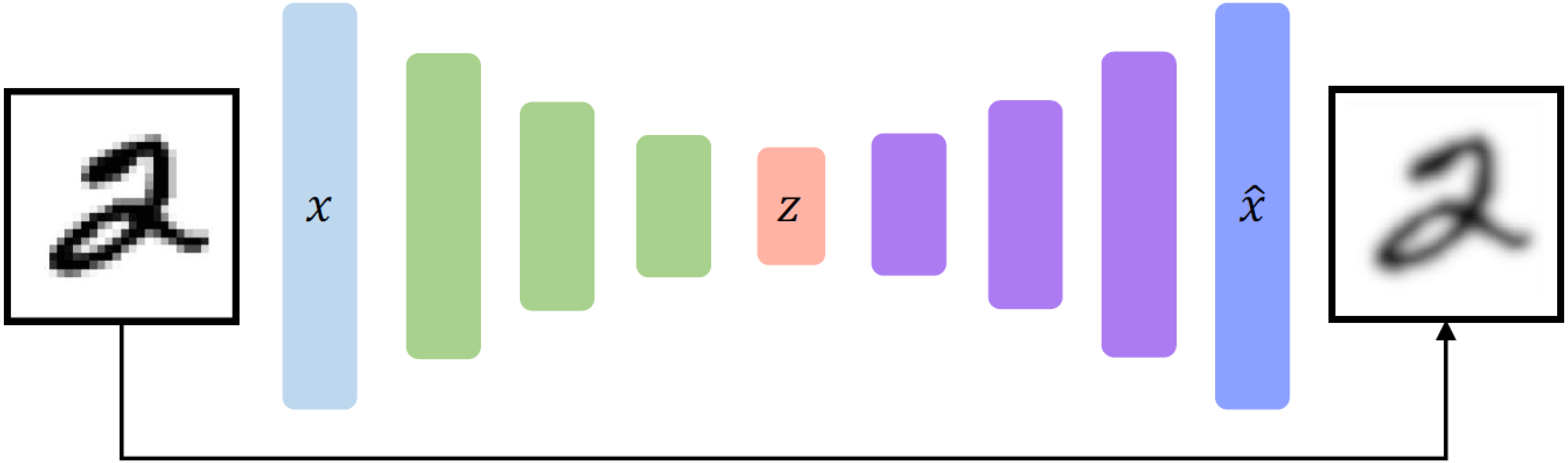
How can we learn this latent space?
Train the model to use these features to **reconstruct the original data**



"Decoder" learns mapping back from latent, z, to a reconstructed observation, $\hat{x}$

Autoencoders: Background (3/7)

How can we learn this latent space?
Train the model to use these features to **reconstruct the original data**

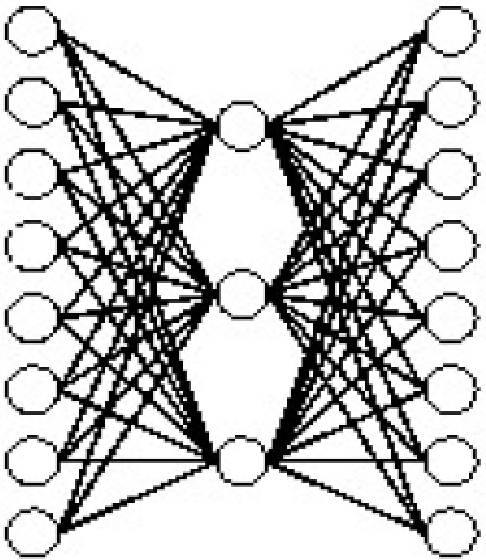


$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$$

Loss function doesn't use any labels!!

Autoencoders: Background (4/7)

Example 1: A 8 x 3 x 8 network was trained to learn the identity function



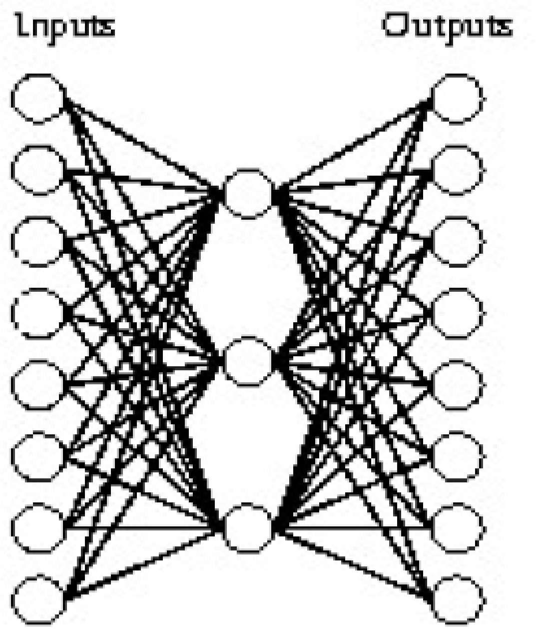
A target function:

Input		Output
10000000	→	10000000
01000000	→	01000000
00100000	→	00100000
00010000	→	00010000
00001000	→	00001000
00000100	→	00000100
00000010	→	00000010
00000001	→	00000001

Can this be learned??

Autoencoders: Background (5/7)

Example 1: A 8 x 3 x 8 network was trained to learn the identity function



Input		Hidden Values				Output
10000000	→	.89	.04	.08	→	10000000
01000000	→	.15	.99	.99	→	01000000
00100000	→	.01	.97	.27	→	00100000
00010000	→	.99	.97	.71	→	00010000
00001000	→	.03	.05	.02	→	00001000
00000100	→	.01	.11	.88	→	00000100
00000010	→	.80	.01	.98	→	00000010
00000001	→	.60	.94	.01	→	00000001

Notice that if the encoded values are rounded to zero or one, the result is the standard binary encoding for eight distinct values.

⇒ Eight input features reduce to three, and the low dimension features can reconstruct the original data.

Autoencoders: Background (6/7)

Example 2: MNIST dataset

Dimensionality of latent space → reconstruction quality

Autoencoding is a form of compression!

Smaller latent space will force a larger training bottleneck

2D latent space | 5D latent space | Ground Truth

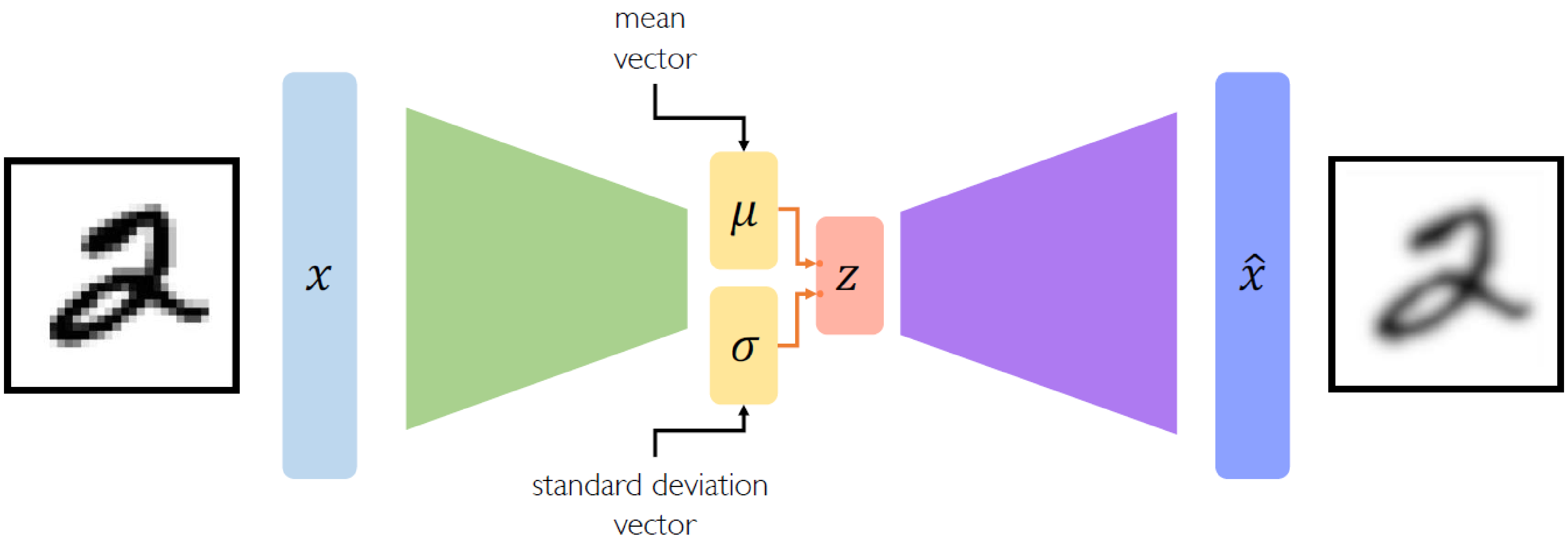
Autoencoders: Background (7/7)

Autoencoders Summary

- Autoencoder = Encoder + Decoder (編碼器+解碼器)
- Bottleneck hidden layer forces network to learn a compressed latent representation
- Reconstruction loss forces the latent representation to capture (or encode) as much "information" about the data as possible
- Training:
    - Inputs: original input X
    - Targets: original input X (X are Not Labels)
- Application: Dimensionality reduction

Variational Autoencoders (VAEs)

VAEs: key difference with traditional autoencoder



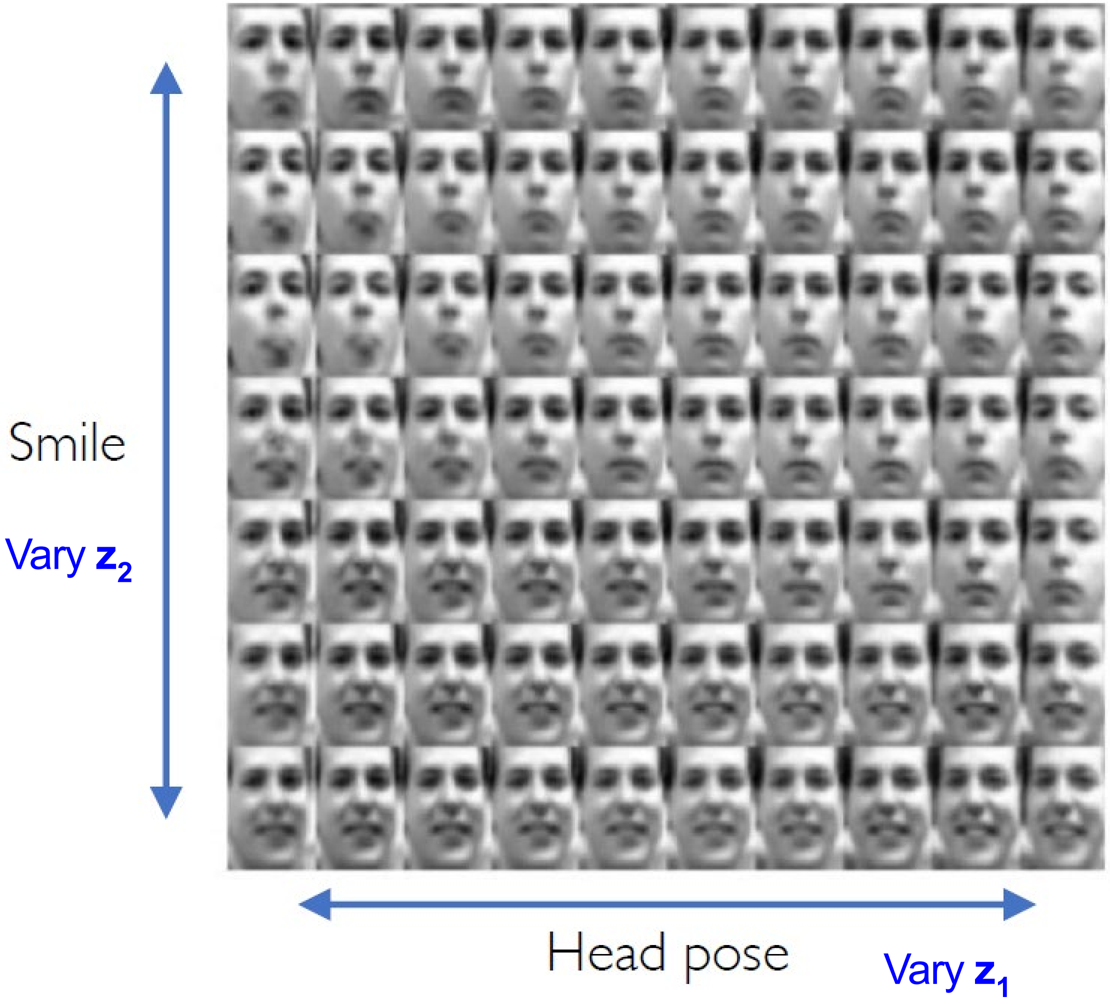
Variational autoencoders are a probabilistic twist on autoencoders!

Sample from the mean and standard dev. to compute latent sample

Variational Autoencoders (VAEs)

Variational Autoencoders:
Generating Data!

Different dimensions of **z** encode interpretable factors of variation

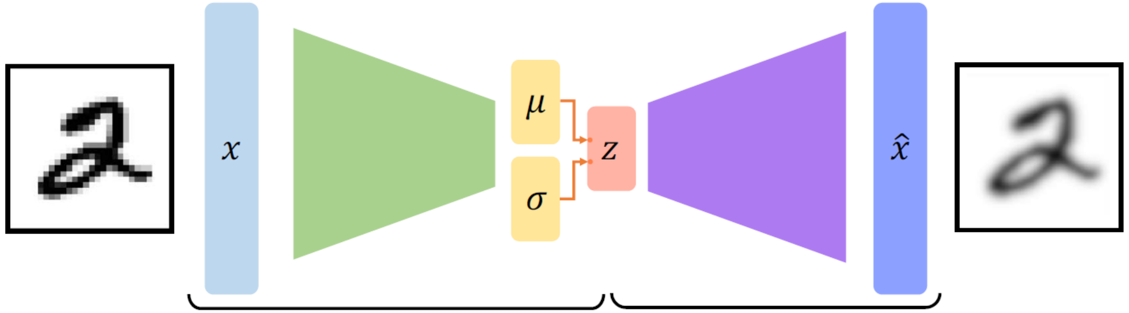


Variational Autoencoders (VAEs)

VAEs Summary

- Reparameterization trick to train end-to-end
- Interpret hidden latent variables using perturbation

$\Rightarrow$ Generating new examples



Samples blurrier and lower quality compared to GANs

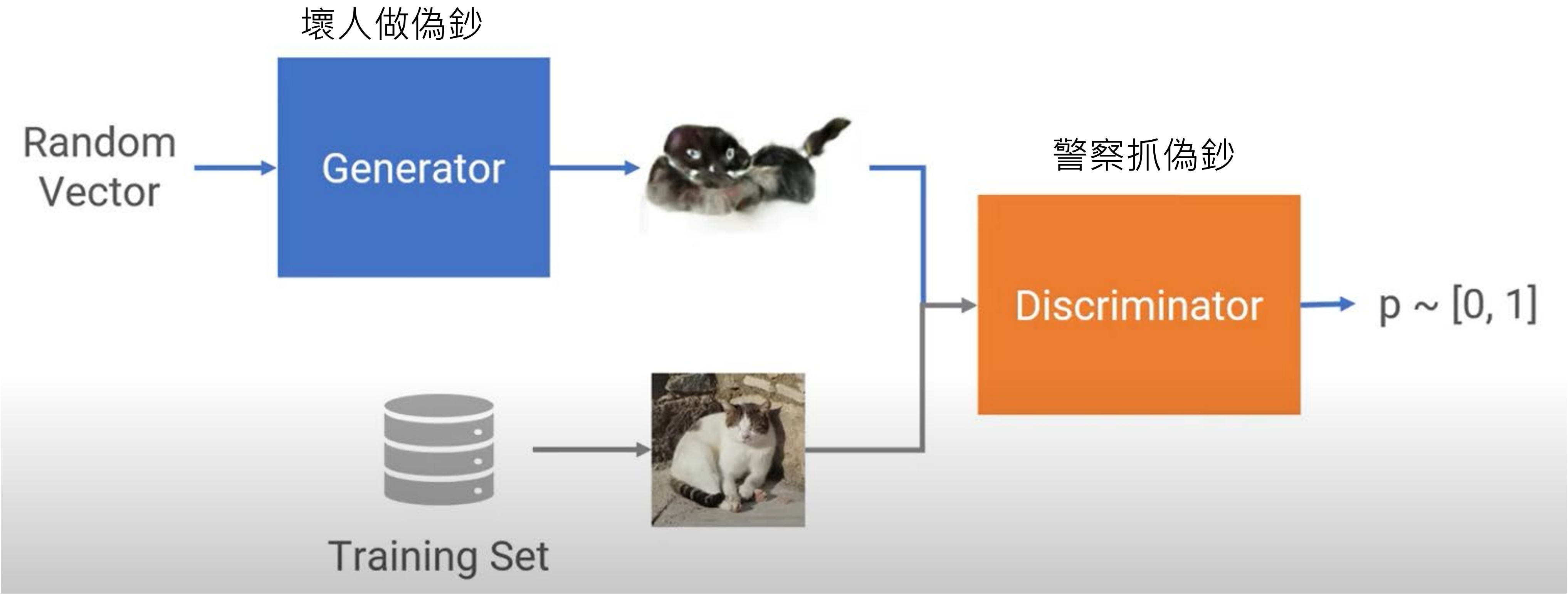
Generative Adversarial Networks (GANs) (1/2)

- GANs are an approach to generative modeling using deep learning methods.
- GANs consist of two neural networks that compete against each other during training.
  - The generator tries to generate realistic samples that have never been seen before.
  - The discriminator tries to identify whether its inputs are real or fake.

Generative Adversarial Networks (GANs) (2/2)

Train two networks with opposing objectives:

- **Generator:** tries to fool the discriminator by generating real-looking samples
- **Discriminator:** tries to distinguish between generated and real samples



GAN objective

- The discriminator $D(x)$ should output the probability that the sample x is real
  - That is, we want $D(x)$ to be close to 1 for real data and close to 0 for fake
- Expected conditional log likelihood for real and generated data:

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

We seed the generator with noise z
drawn from a simple distribution p

GAN objective

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

- The discriminator wants to correctly distinguish real and fake samples:

$$D^* = \arg\max_D V(G,D)$$

目標：最後讓警察覺得
真鈔的機率是0.5
假鈔的機率是0.5

- The generator wants to fool the discriminator:

$$G^* = \arg\min_G V(G,D)$$

- Train the generator and discriminator jointly in *a minimax game*

GAN Learning 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:

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

  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:

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

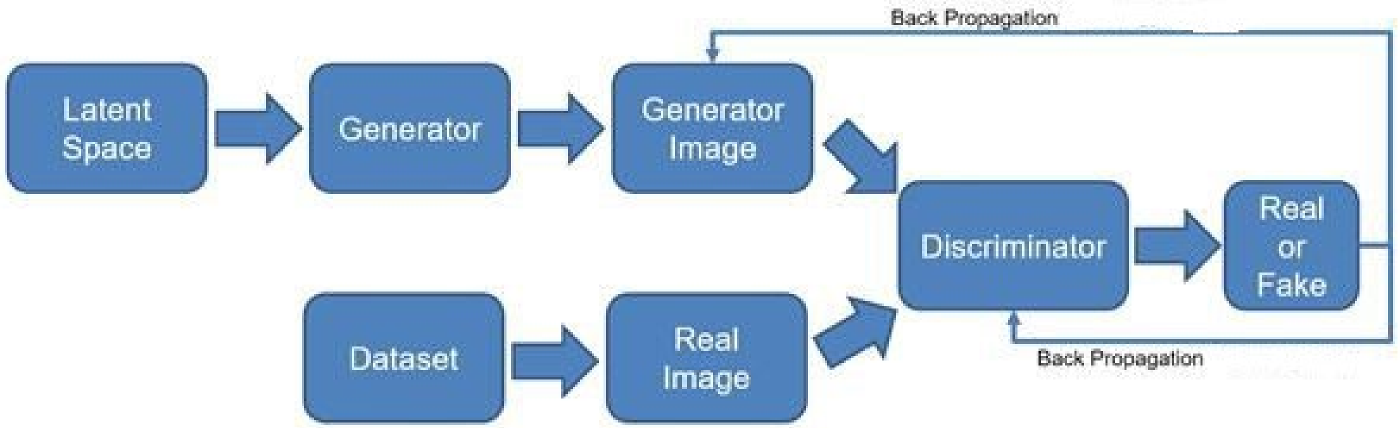
end for

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

Training of GAN

Repeat the 2 steps:

1. Update the discriminator network;
2. Update the generator network.



Phase 1: Update the Discriminator

Train a classifier

1. Generate a batch of fake samples by the generator;
2. Randomly sample a batch of real samples;
3. Inputs: $\mathbf{X} = [\text{real\_samples}, \text{fake\_samples}]$;
4. Targets: $\mathbf{y} = [\text{True}, \cdots, \text{True}, \text{False}, \cdots, \text{False}]$;
5. Update the discriminator network using $\mathbf{X}$ and $\mathbf{y}$.

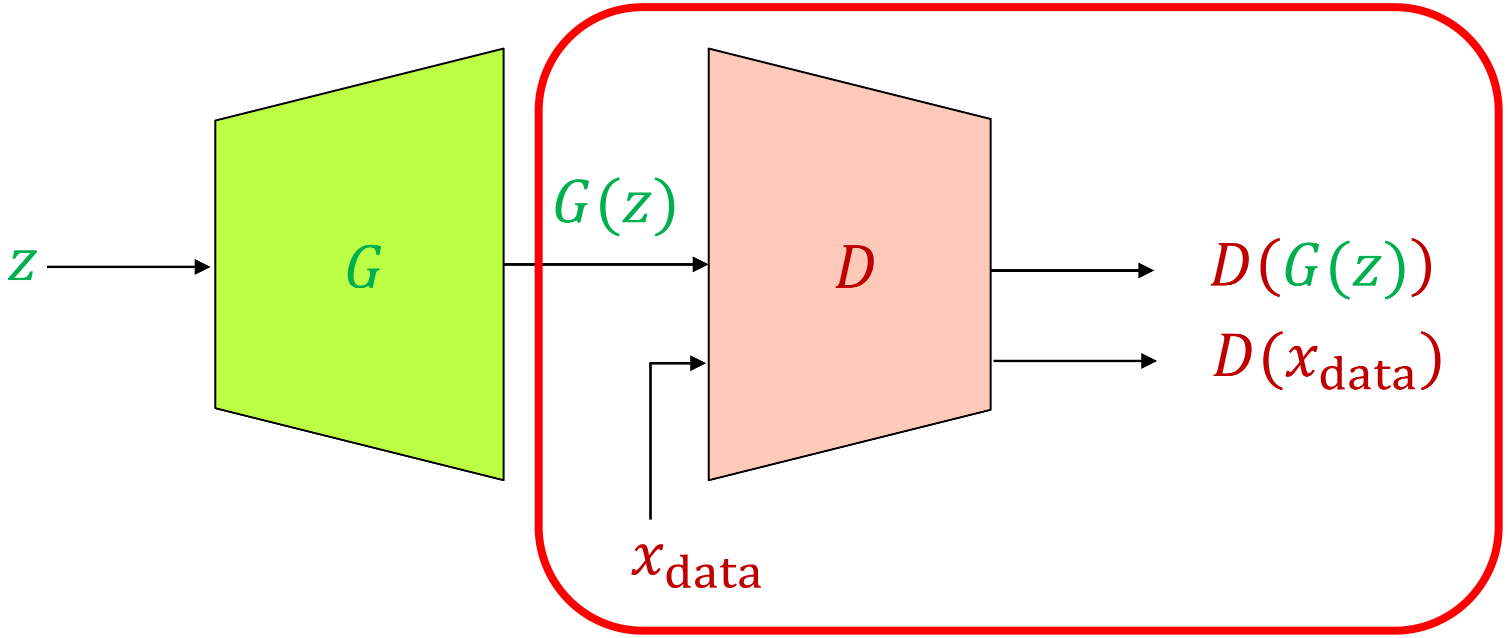
   (freeze generator's parameters)

Phase 2: Update the Generator

Connect the generator and discriminator (freeze discriminator's parameters).

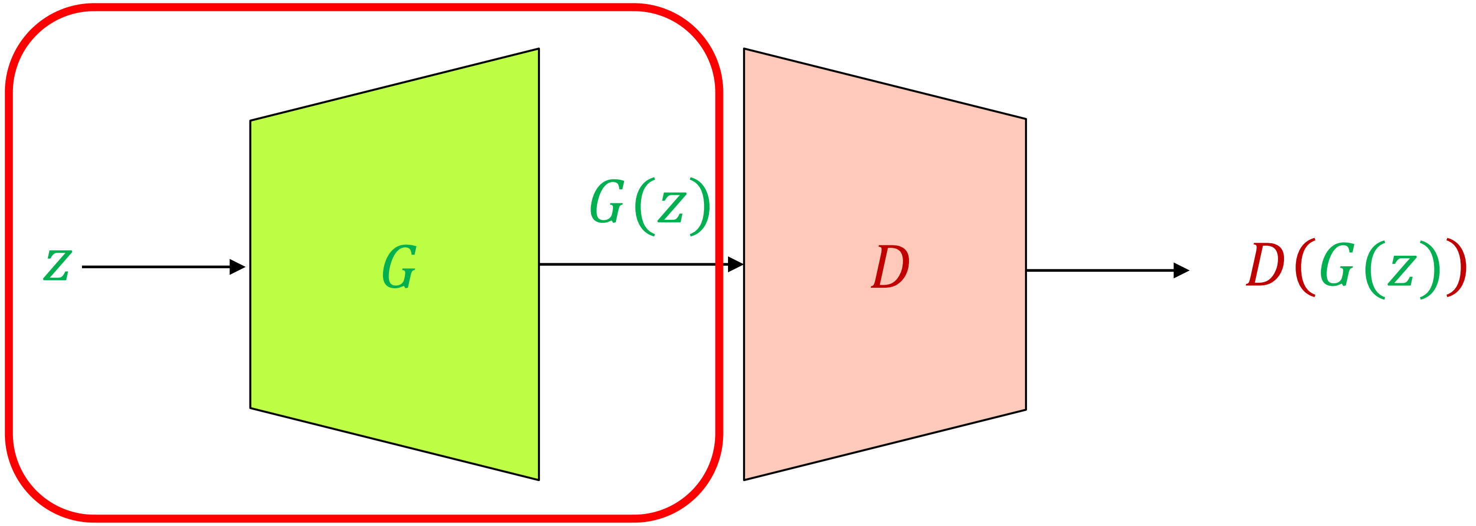
GAN: Conceptual picture

- Update discriminator
  - push $D(x_{\text{data}})$ close to 1 and $D(G(z))$ close to 0
  - freeze generator's parameters



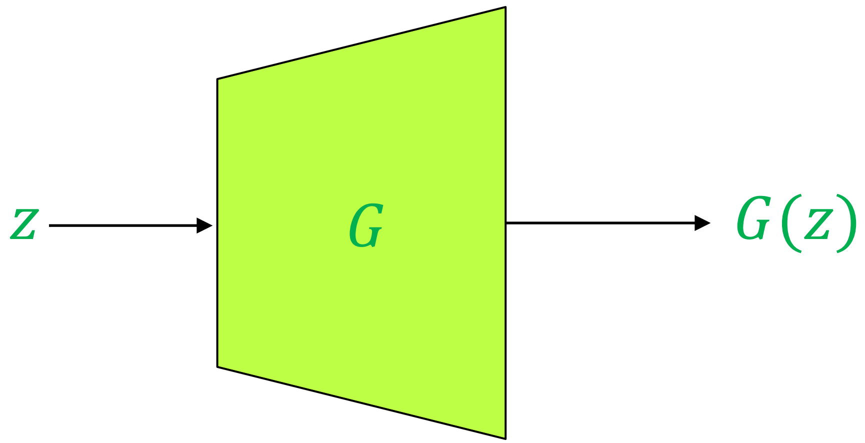
GAN: Conceptual picture

- Update generator: increase $D(G(z))$
  - Requires back-propagating through the composed generator-discriminator network
  - Freezes discriminator's parameter



GAN: Conceptual picture

- Test time – the discriminator is discarded



2017: Explosion of GANs

See also: https://github.com/soumith/ganhacks for tips and tricks for trainings 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

Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Smiling woman   Neutral woman   Neutral man

Samples from the model

Smiling Man

Average Z vectors, do arithmetic

Generative Adversarial Nets: Interpretable Vector Math

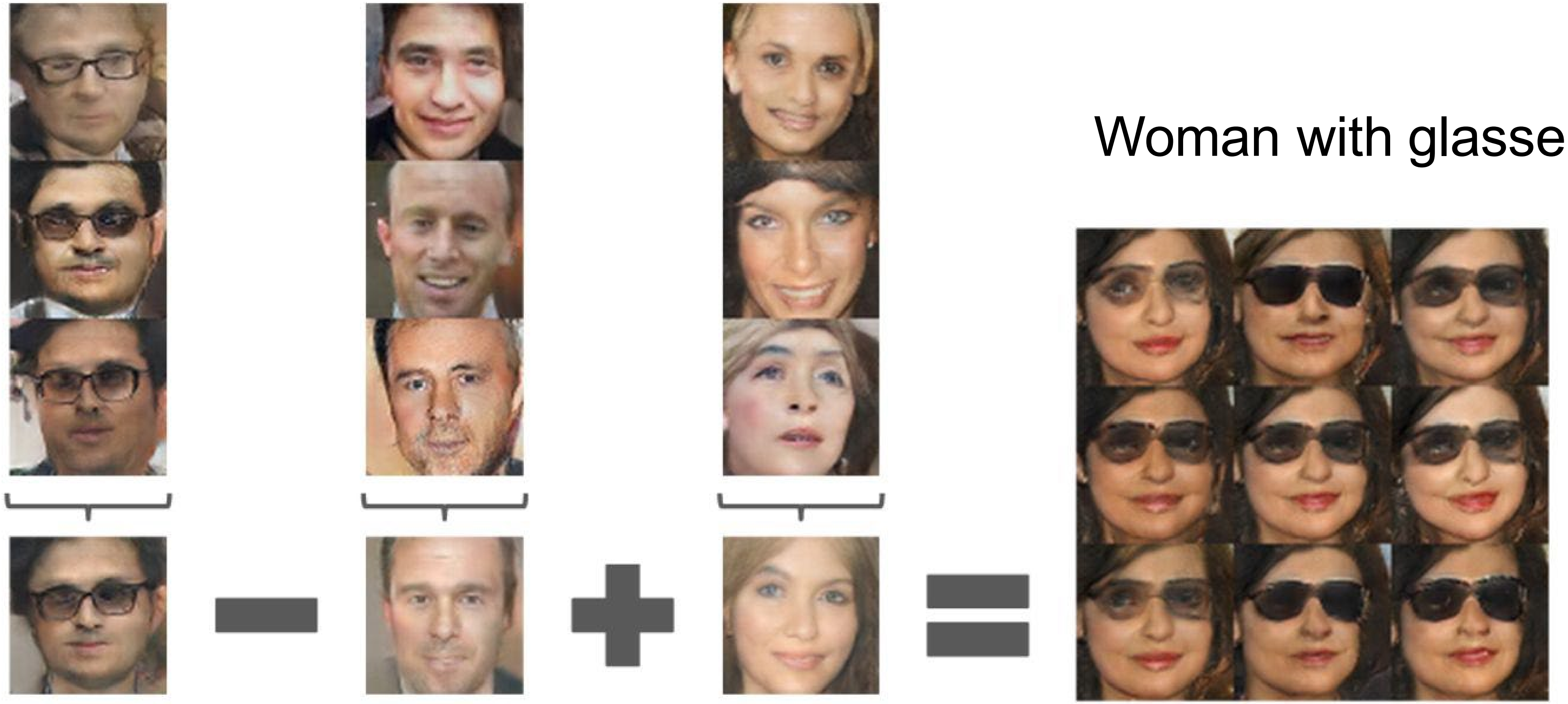
Glasses man

No glasses man

No glasses woman

Radford et al, ICLR 2016

Woman with glasses

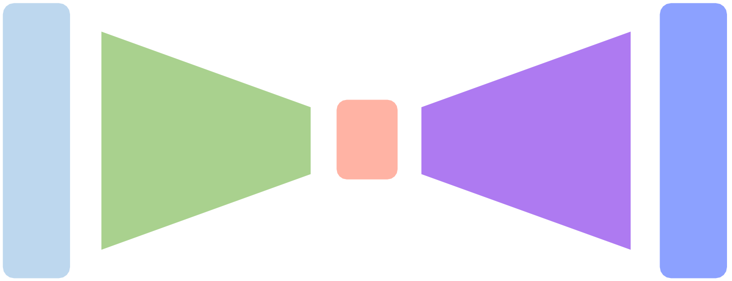


Which face is fake?

Generative Models: Summary

Autoencoders and Variational Autoencoders (VAEs)

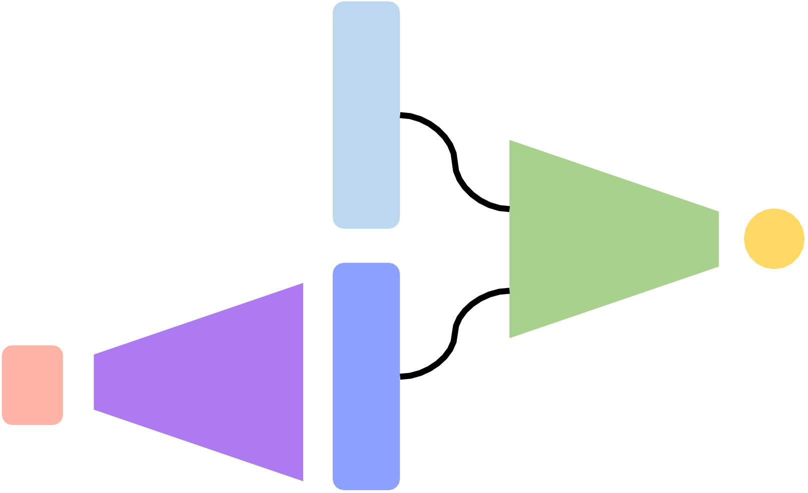
Learn lower-dimensional latent space and sample to generate input reconstructions



VAEs: Explicit density method

Generative Adversarial Networks (GANs)

Competing generator and discriminator networks



GANs: Implicit density method

Resources

Stanford CS231n: Convolutional Neural Networks for Visual Recognition

MIT 6.S191: Introduction to Deep Learning

Illinois CS 498: Introduction to Deep Learning