



Artificial Neural Networks

[2500WETANN]

José Oramas



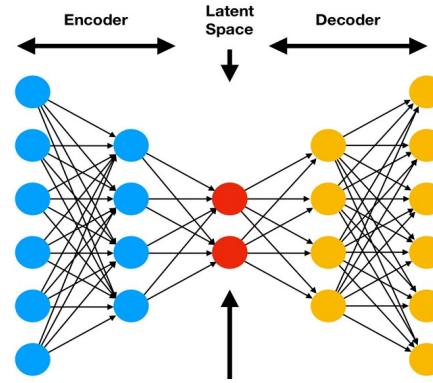
Deep Generative Networks

[Learning how to generate data]

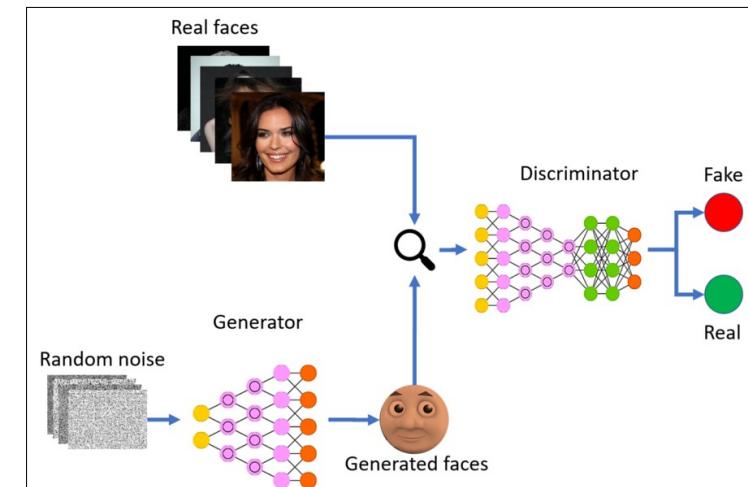
José Oramas

Today's Lecture : Deep Generative Networks

- [Variational] Autoencoders



- Generative Adversarial Networks (GANs)



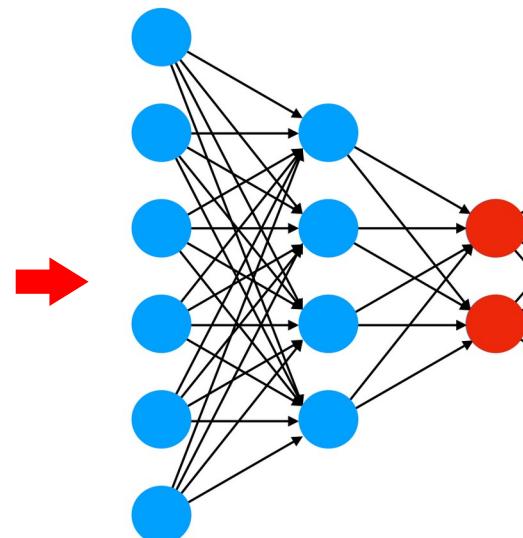
Autoencoders

[Reconstructing Inputs]

DNN Classifiers

Learn to classify the input – Discriminative Model

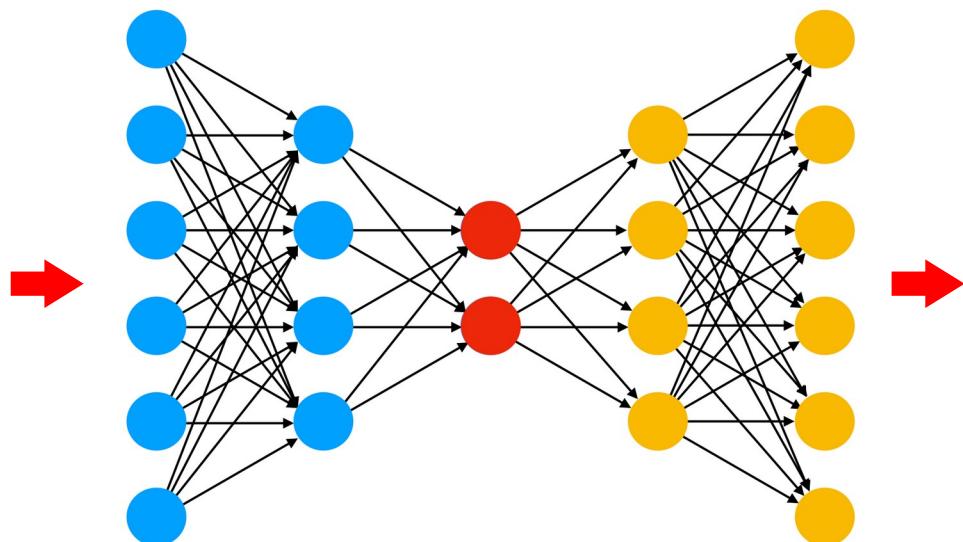
Original Input



Autoencoders

Learn to reconstruct the input

Original Input

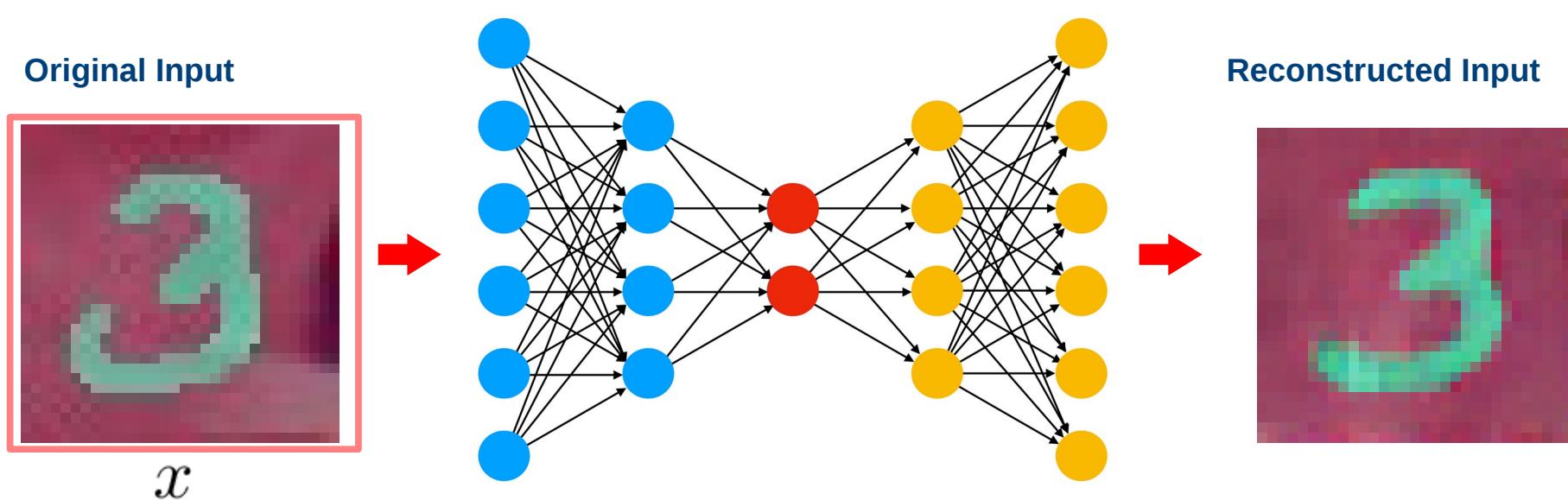


Reconstructed Input



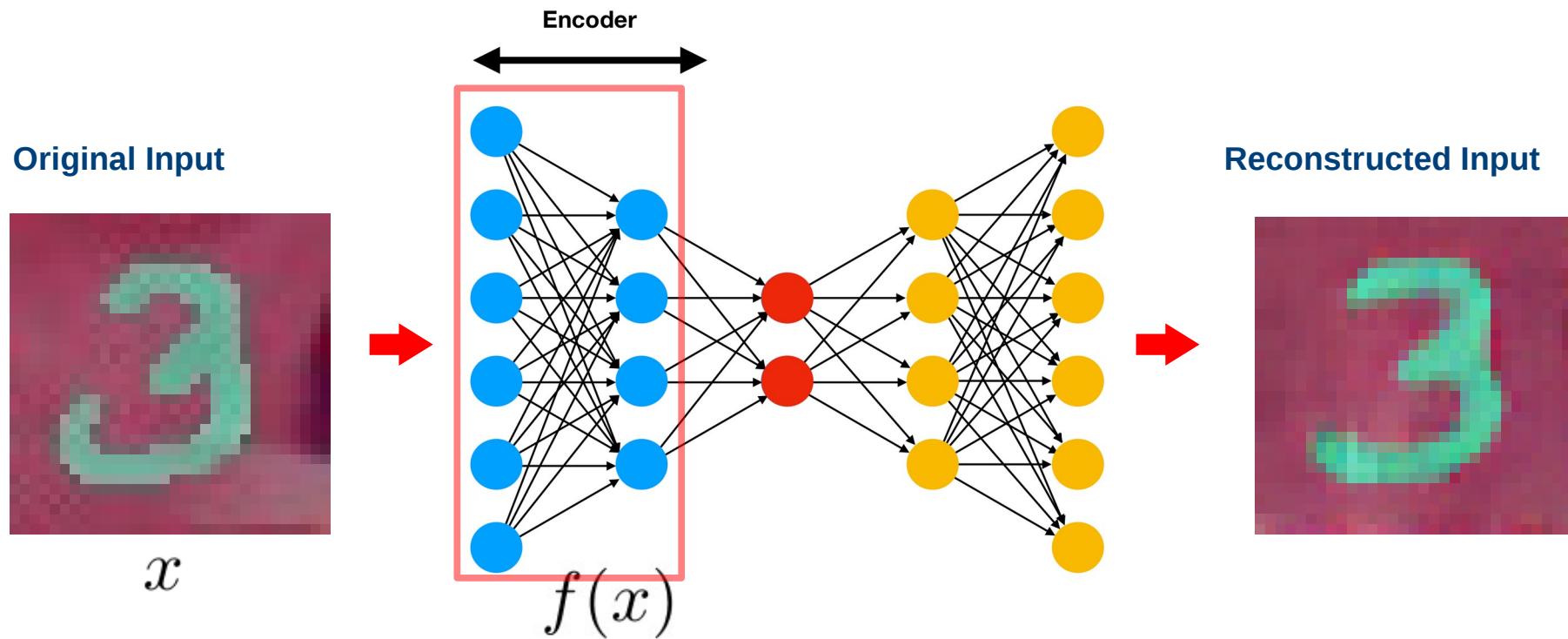
Autoencoders

Learn to reconstruct the input



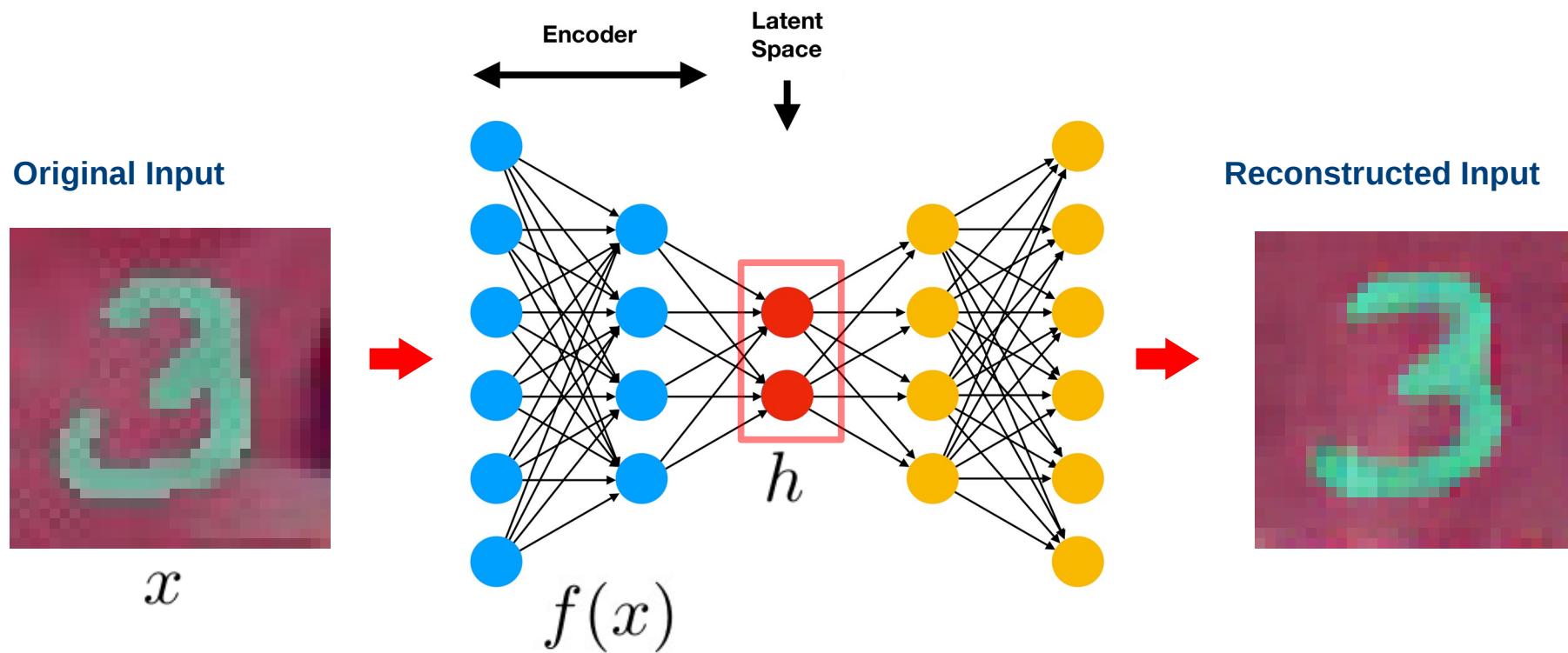
Autoencoders

Learn to reconstruct the input



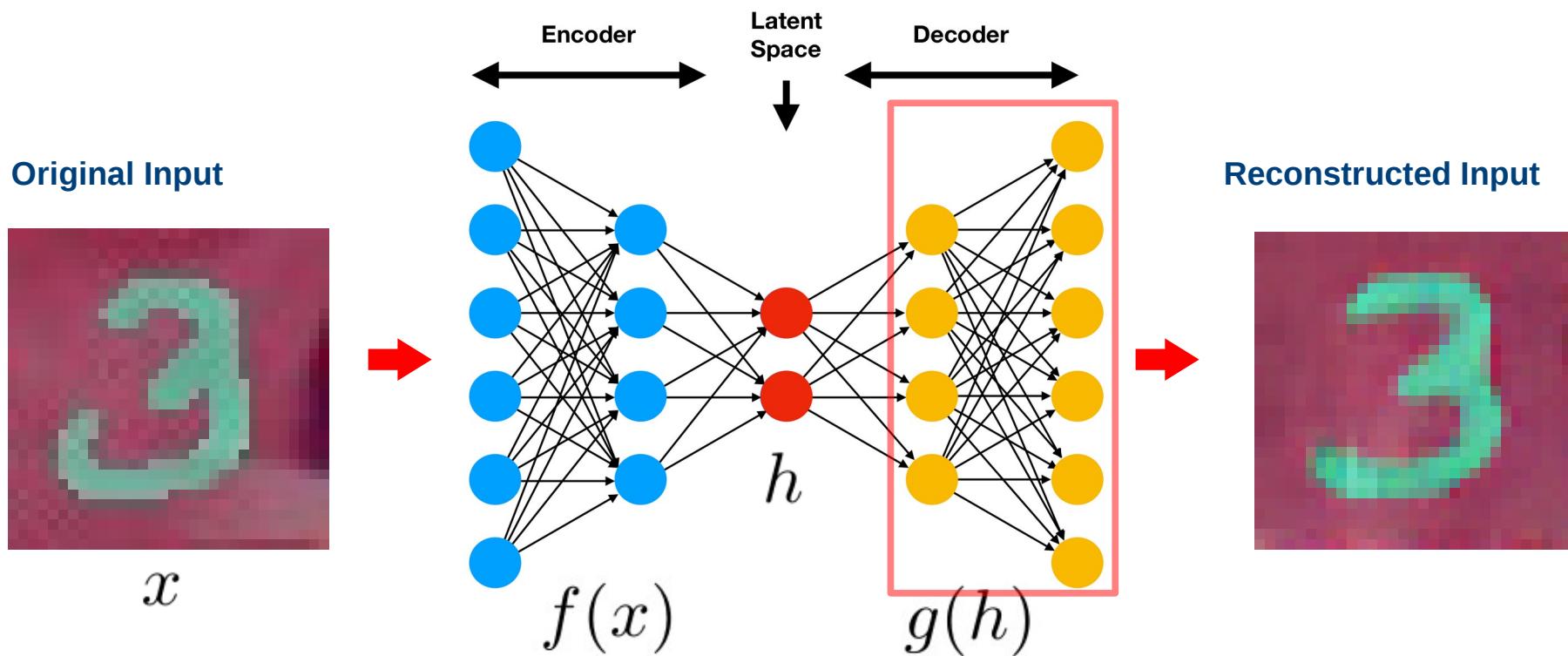
Autoencoders

Learn to reconstruct the input



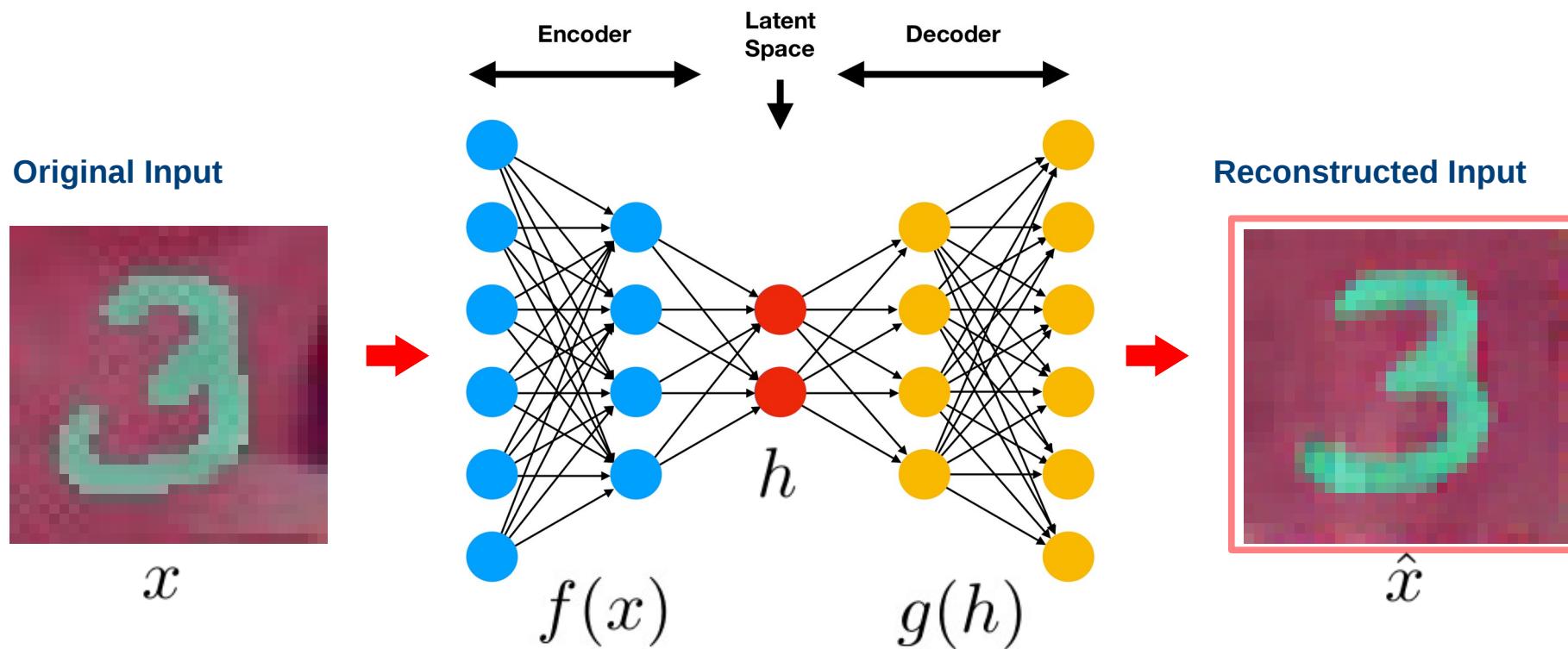
Autoencoders

Learn to reconstruct the input



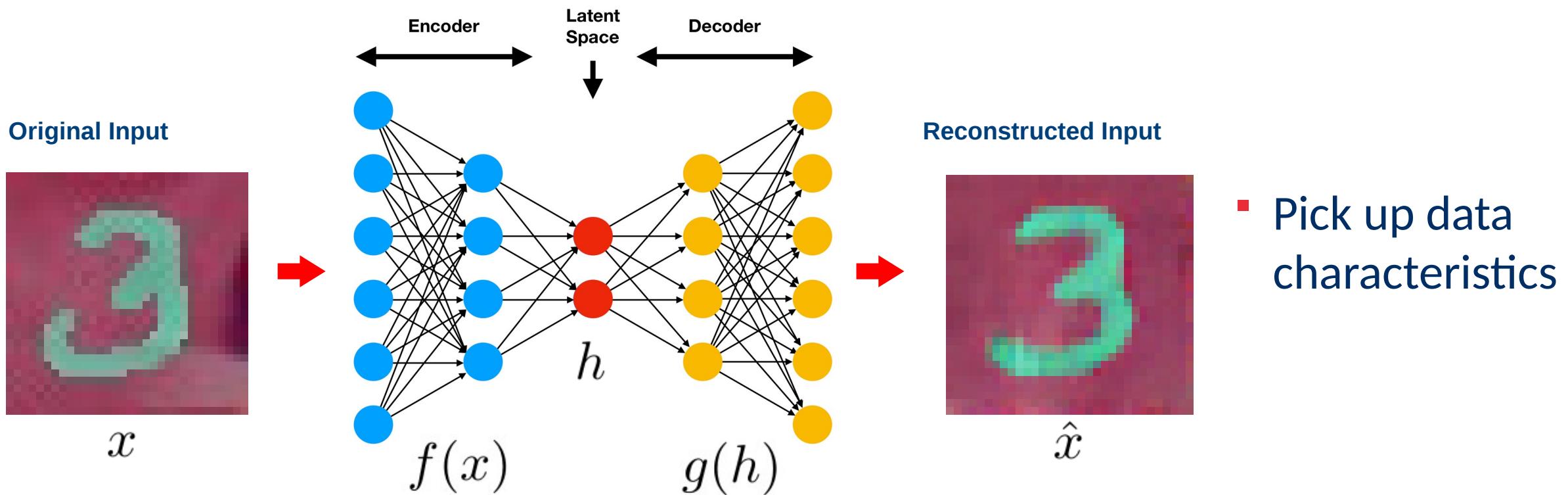
Autoencoders

Learn to reconstruct the input



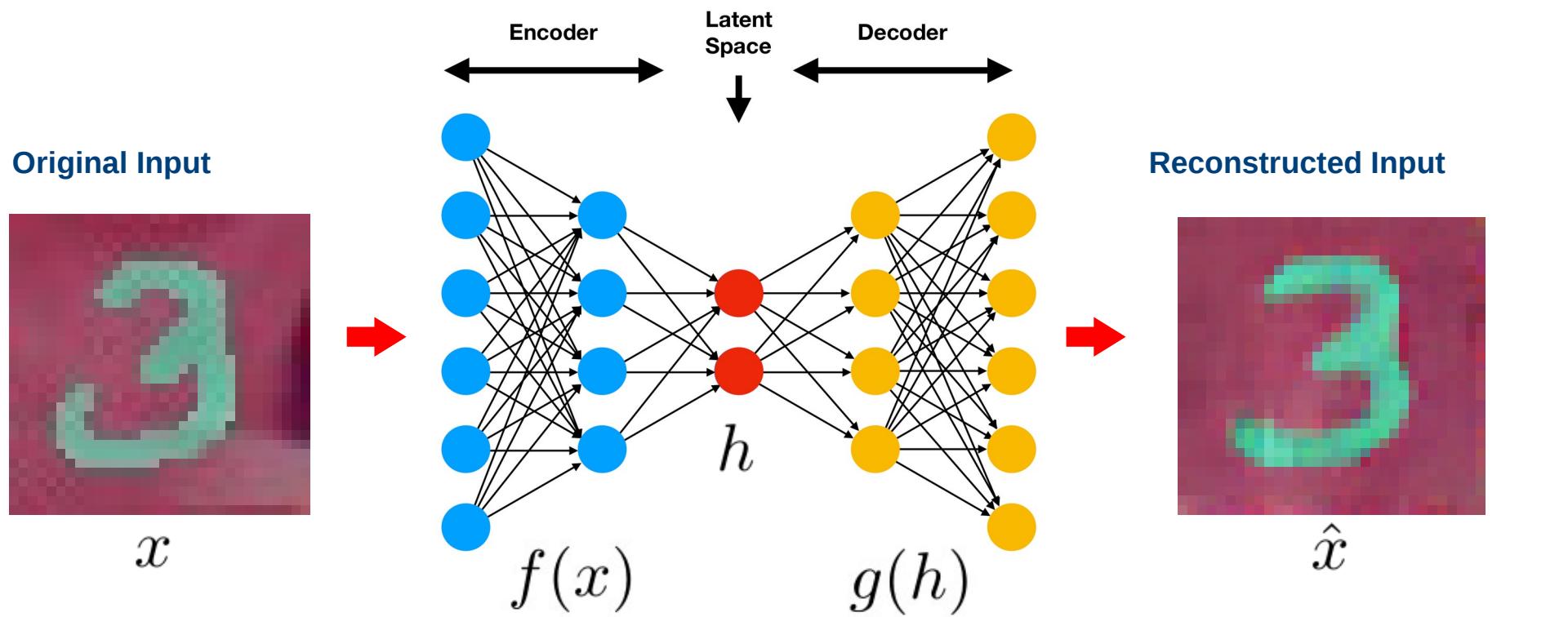
Autoencoders

Learn to reconstruct the input



Autoencoders – Training

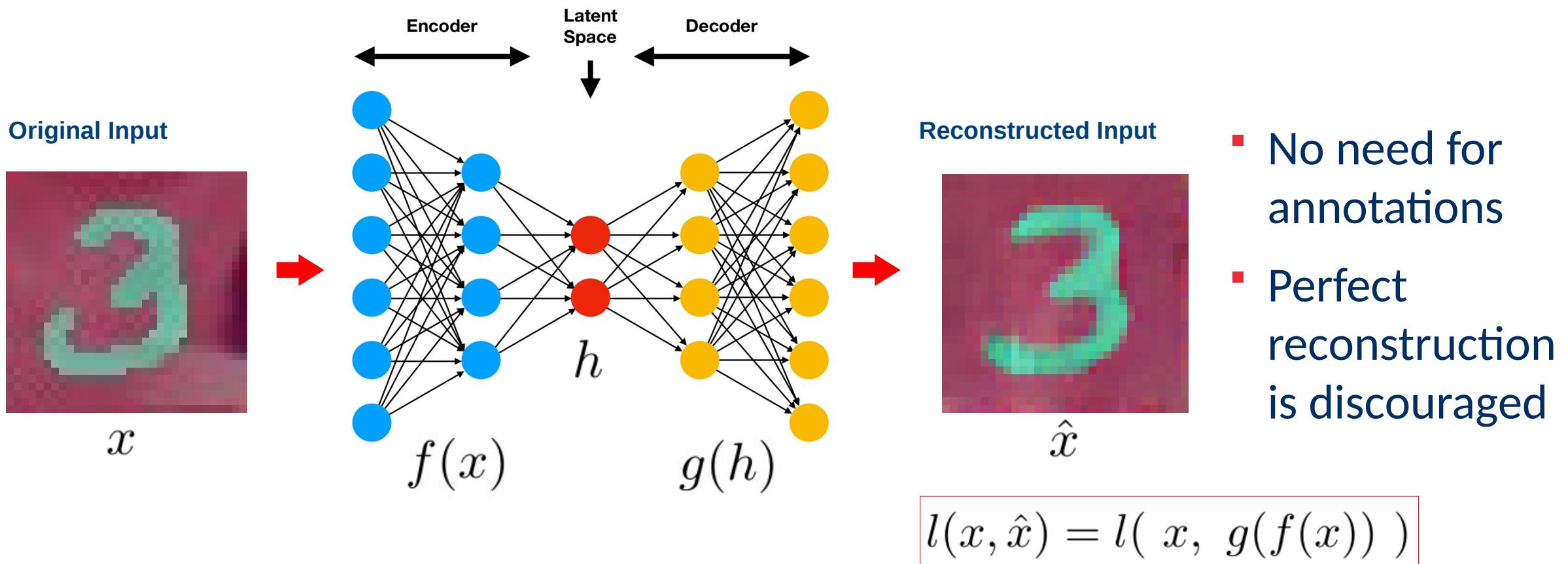
Measure reconstruction error



$$l(x, \hat{x}) = l(x, g(f(x)))$$

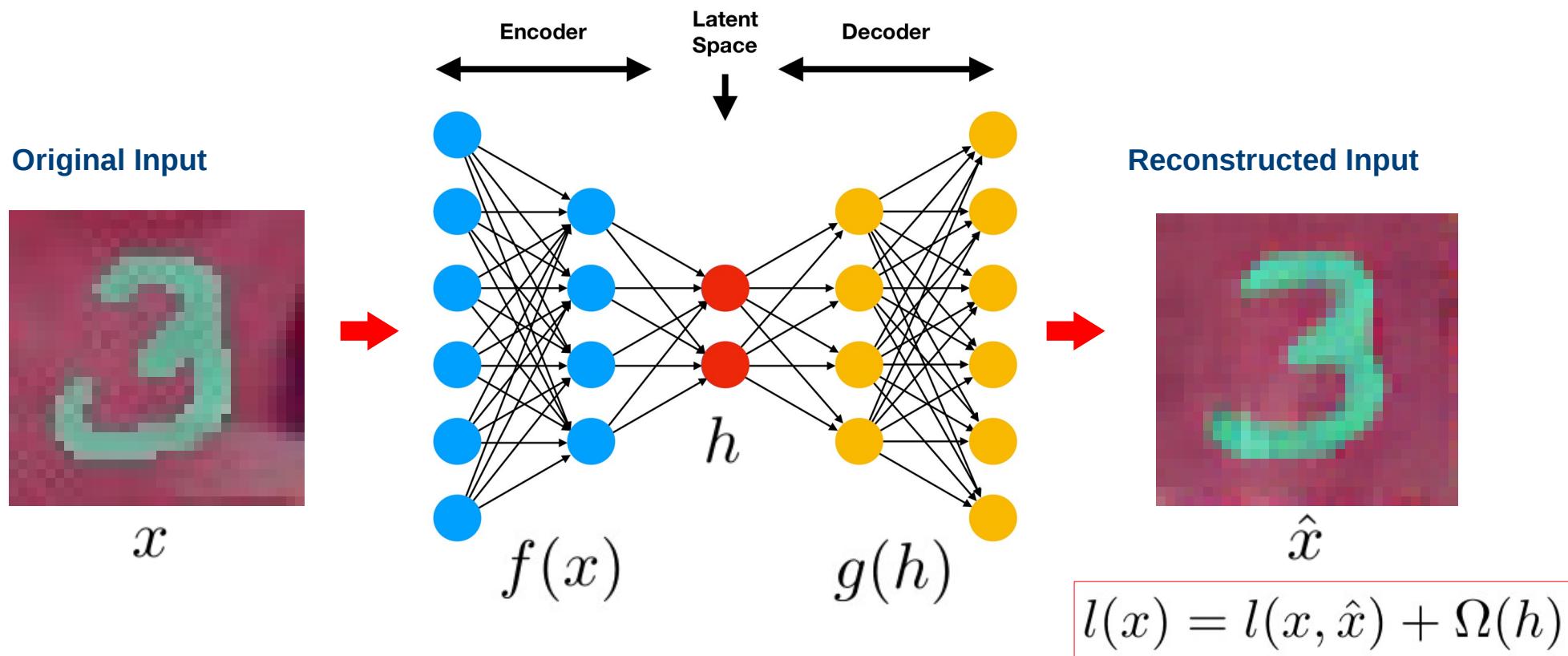
Autoencoders – Training

Measure reconstruction error



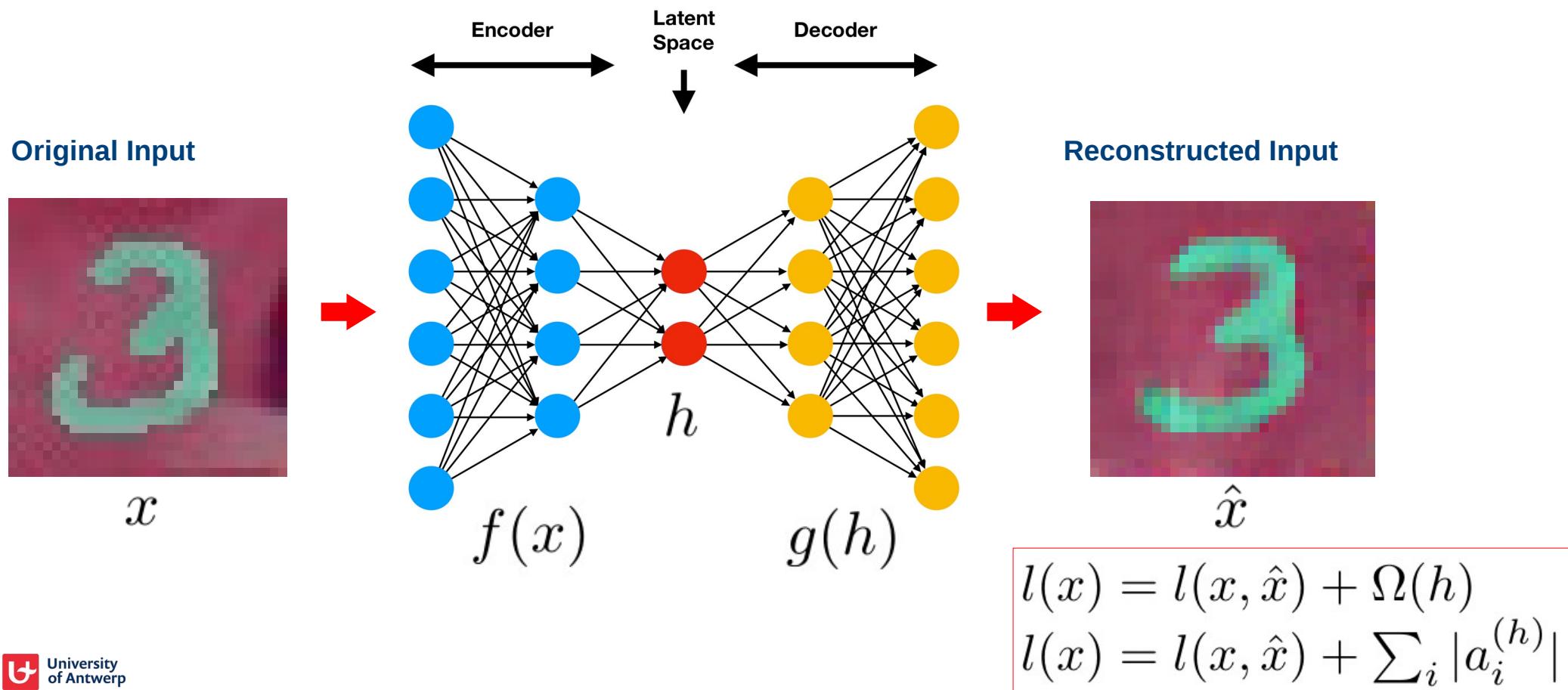
Sparse Autoencoders

Impose additional constraints on the latent space



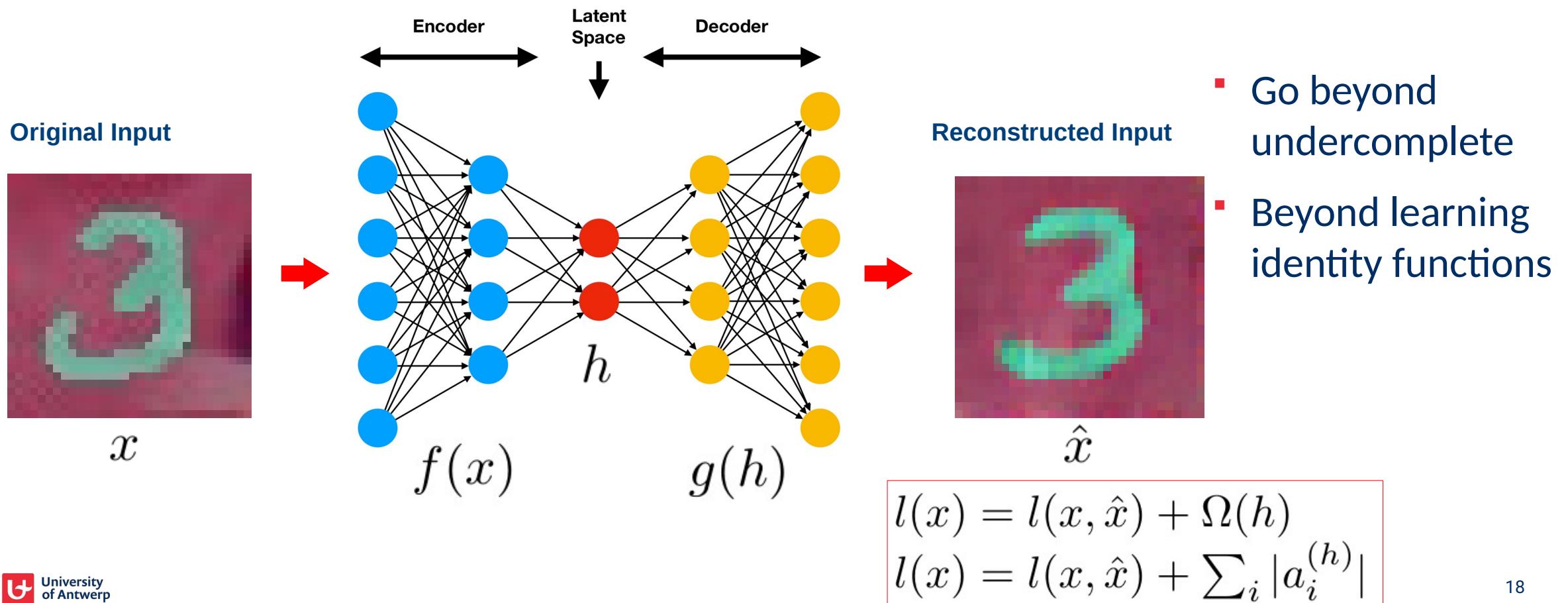
Sparse Autoencoders

Impose additional constraints on the latent space



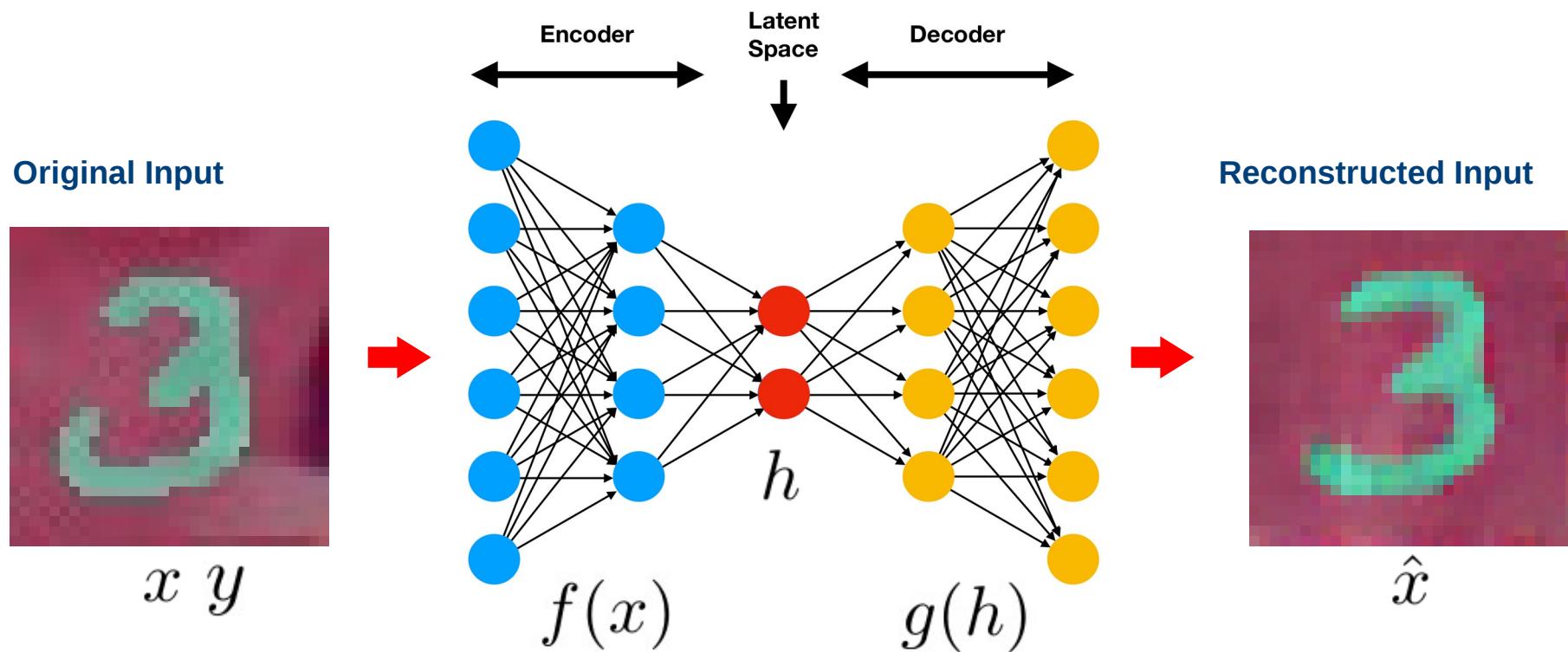
Sparse Autoencoders

Impose additional constraints on the latent space



Sparse Autoencoders

Impose additional constraints on the latent space



$$l(x) = l_{rec}(x, \hat{x}) + l_{class}(y, \hat{y})$$

Denoising Autoencoders (DAEs)

Obtaining a clean reconstruction from a noisy input

Original Input x



Denoising Autoencoders (DAEs)

Obtaining a clean reconstruction from a noisy input

Original Input x



Noise
Injection



Noisy Input x'



Denoising Autoencoders (DAEs)

Obtaining a clean reconstruction from a noisy input

Original Input x



Noise
Injection

Noisy Input x'



Encoder Latent Space

Decoder

$f(x)$

$g(h)$

Reconstructed Input



\hat{x}

Denoising Autoencoders (DAEs)

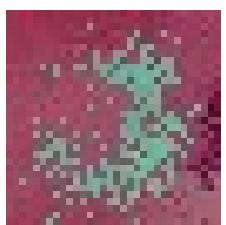
Obtaining a clean reconstruction from a noisy input

Original Input x



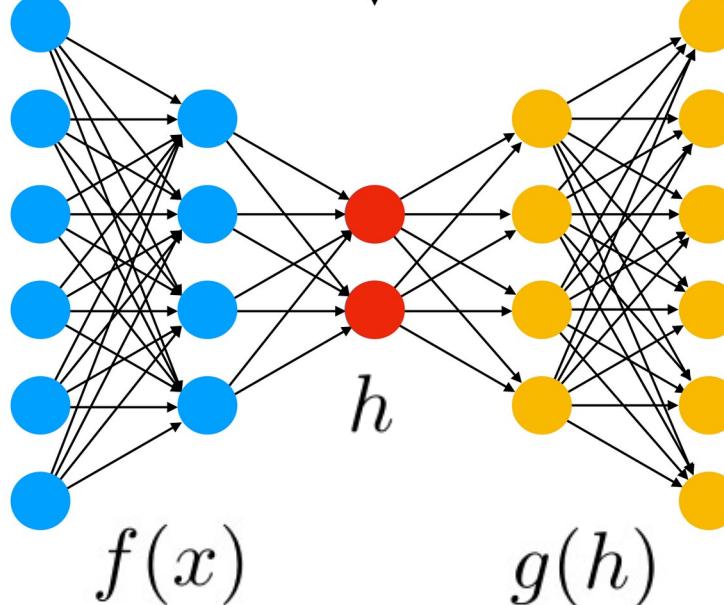
Noise
Injection

Noisy Input x'



Encoder Latent Space

Decoder



Reconstructed Input



\hat{x}

$$l(x, \hat{x}) = l(x, g(f(x')))$$

Denoising Autoencoders (DAEs)

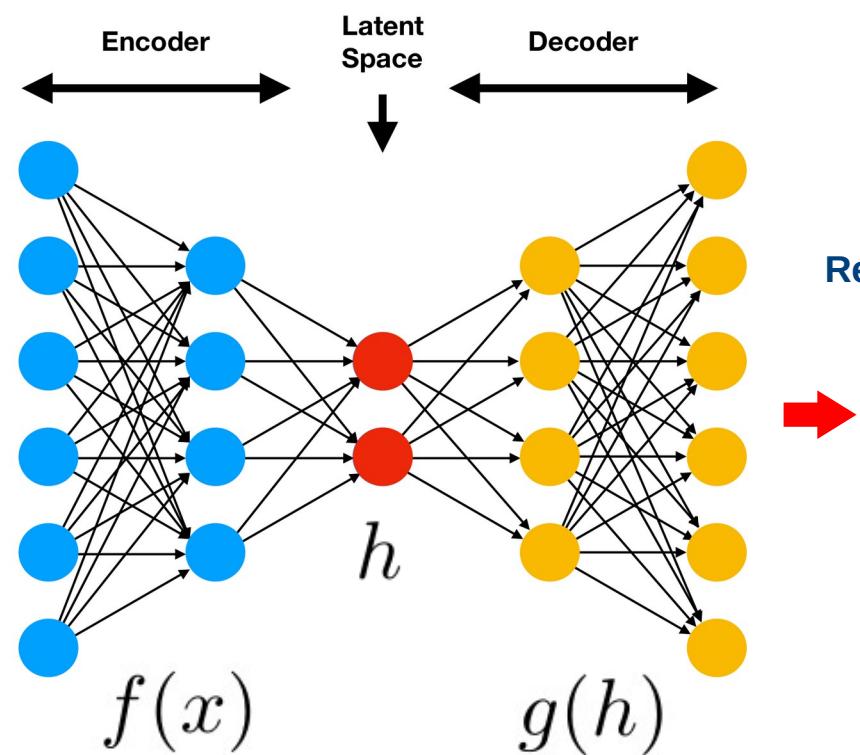
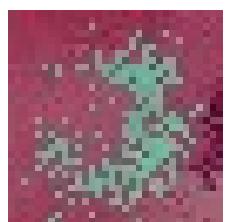
Obtaining a clean reconstruction from a noisy input

Original Input x



Noise Injection

Noisy Input x'



Reconstructed Input



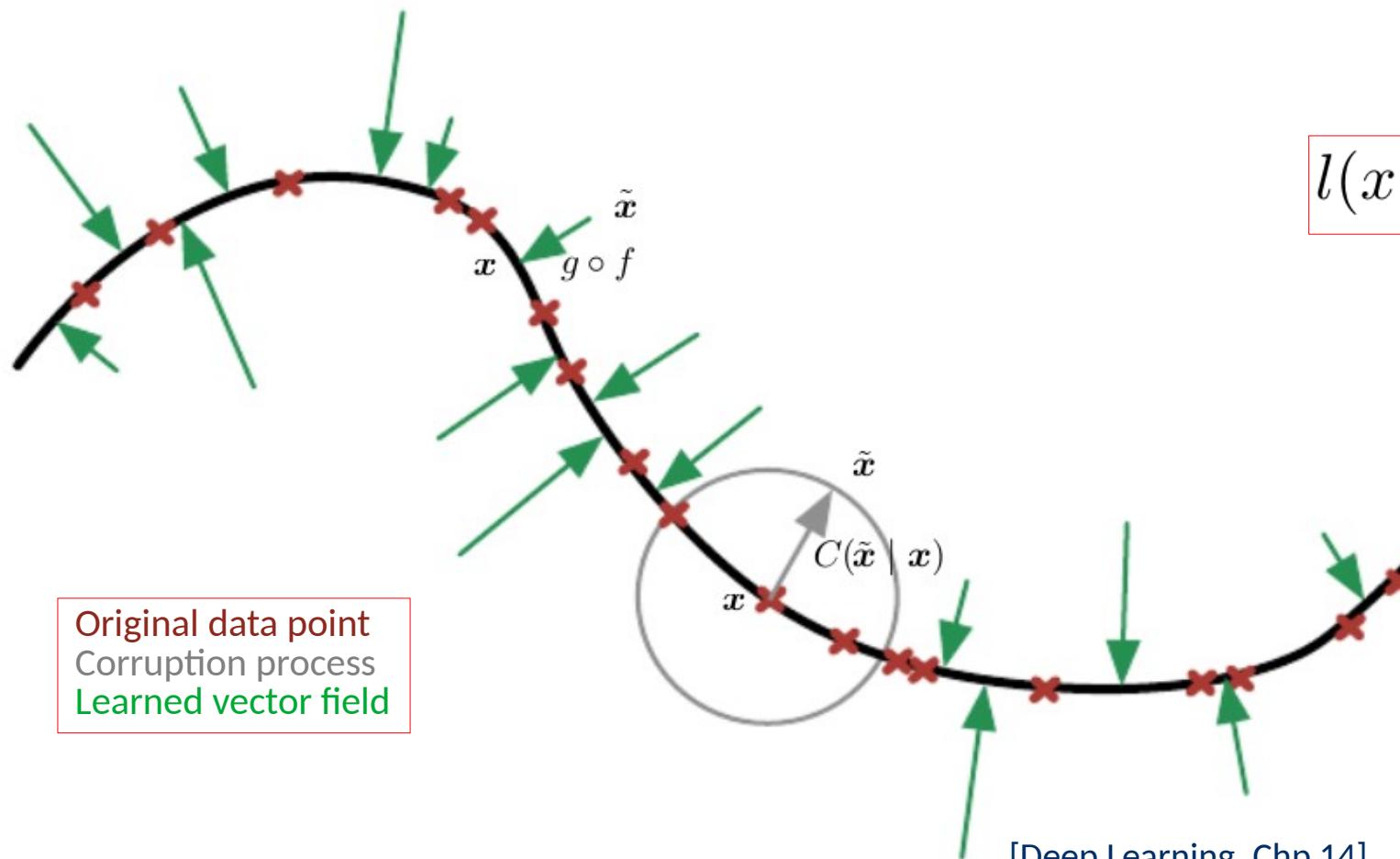
\hat{x}

- Learn how to undo the corruption

$$l(x, \hat{x}) = l(x, g(f(x')))$$

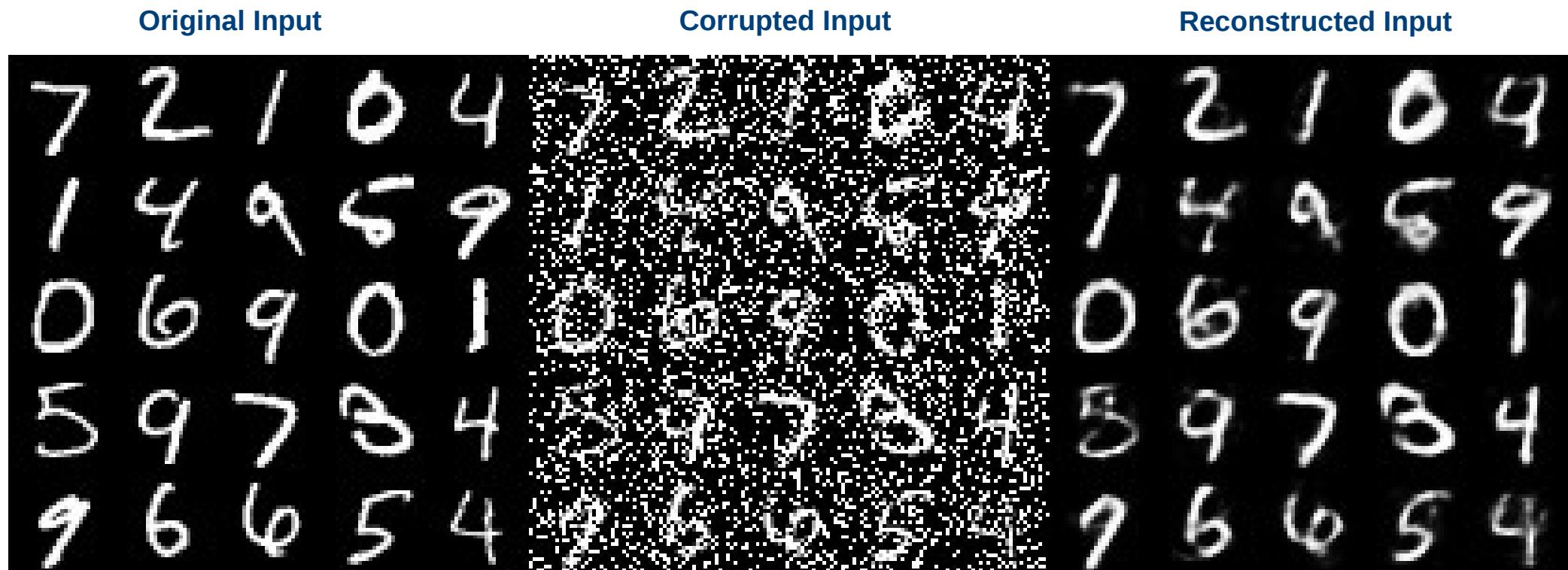
Denoising Autoencoders (DAEs)

Obtaining a clean reconstruction from a noisy input



Denoising Autoencoders (DAEs)

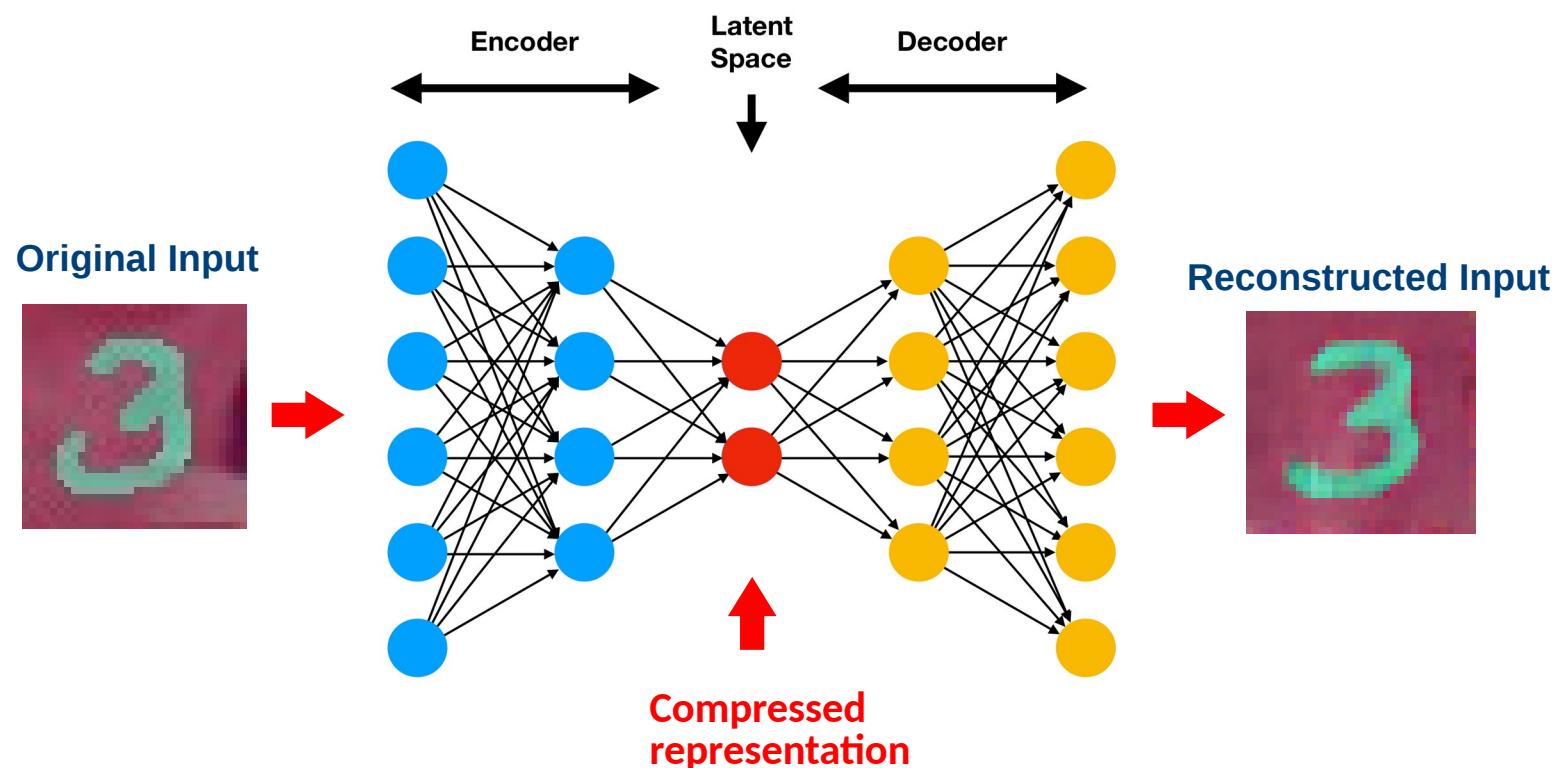
Obtaining a clean reconstruction from a noisy input



[opendeep.org]

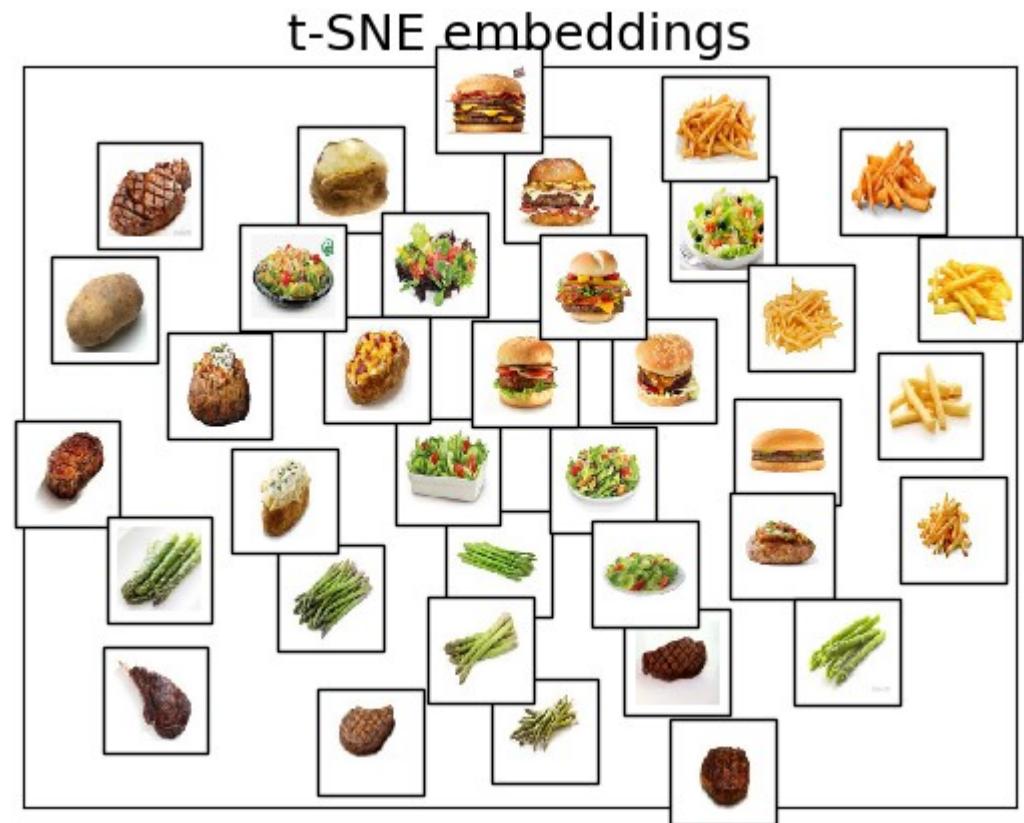
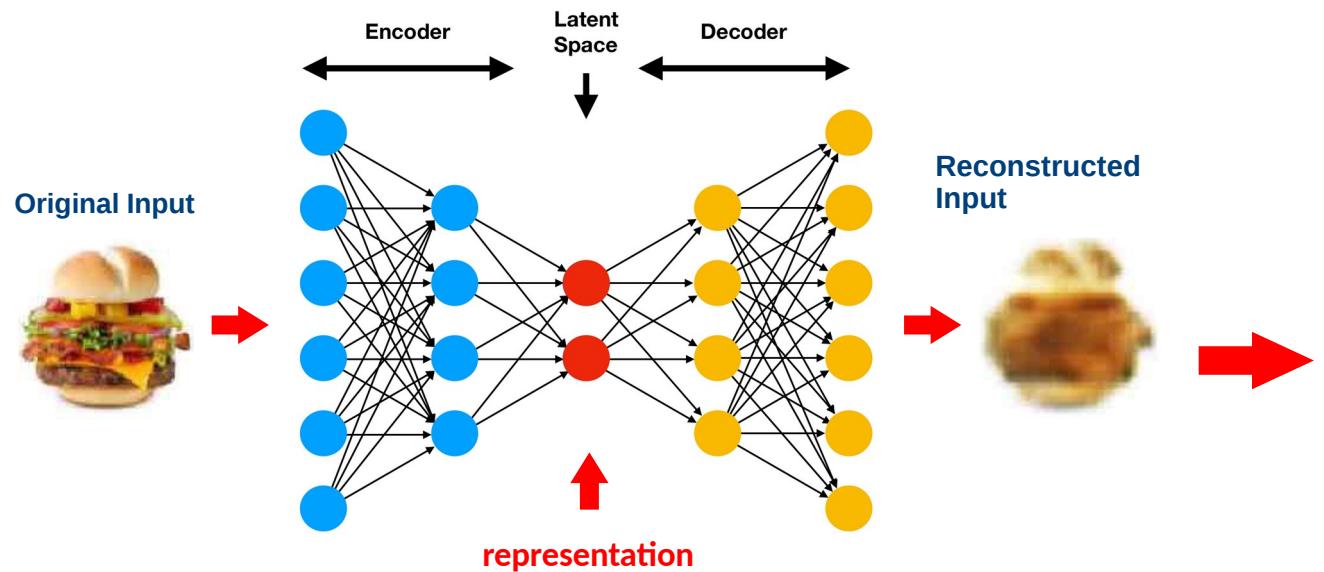
Autoencoders – Applications

Compression & Retrieval



Autoencoders – Applications

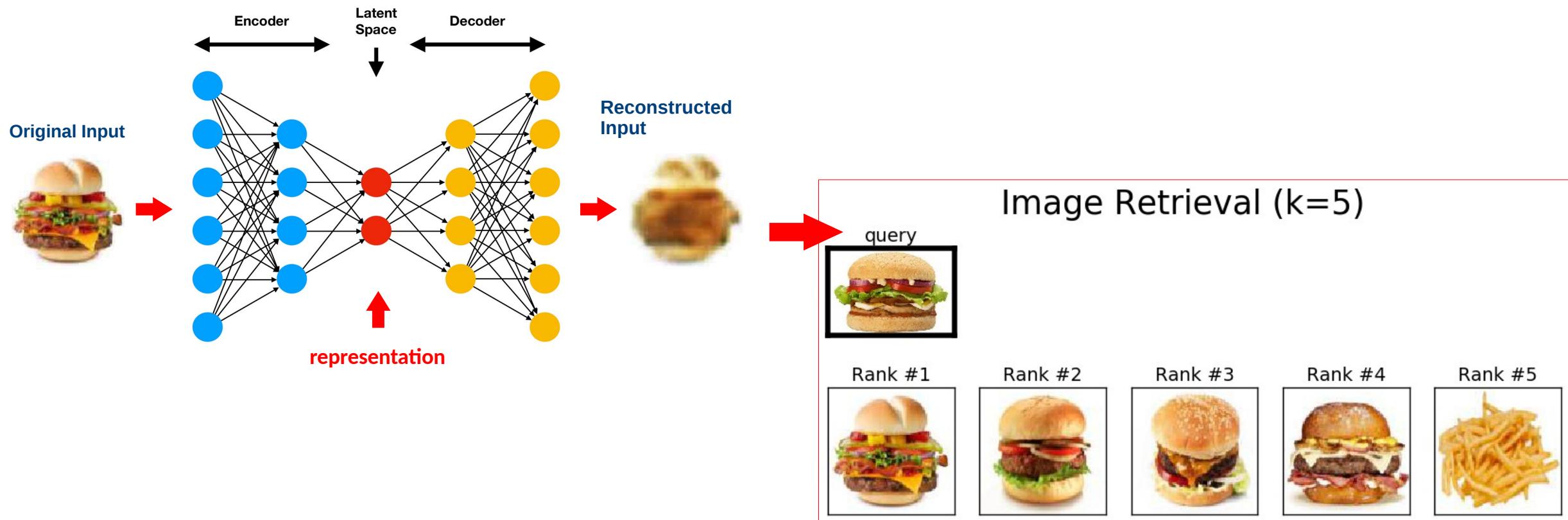
Compression & Retrieval



[A. Wong 2017]

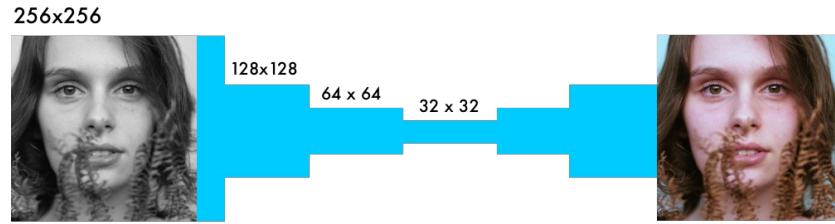
Autoencoders – Applications

Compression & Retrieval



Autoencoders – Applications

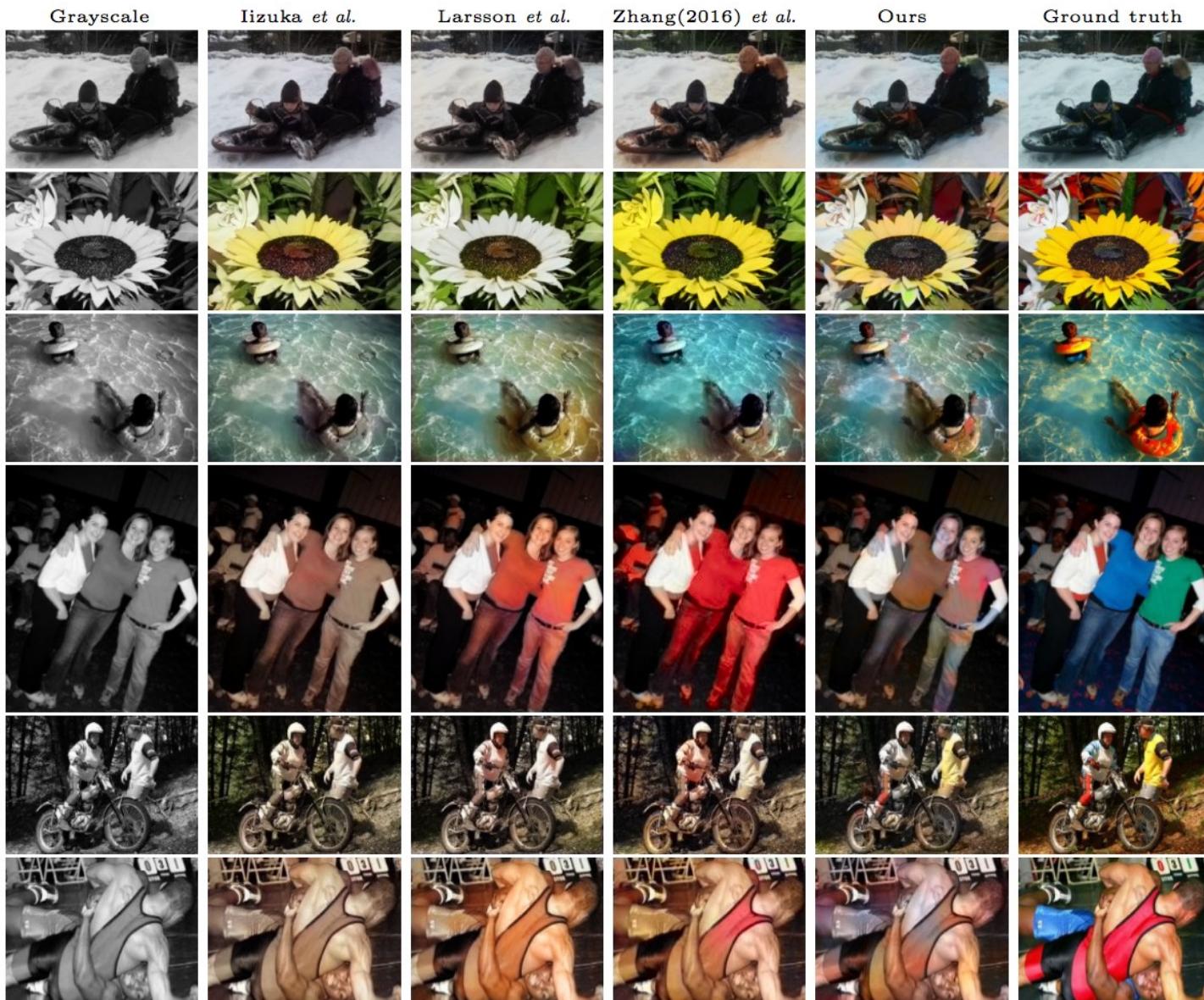
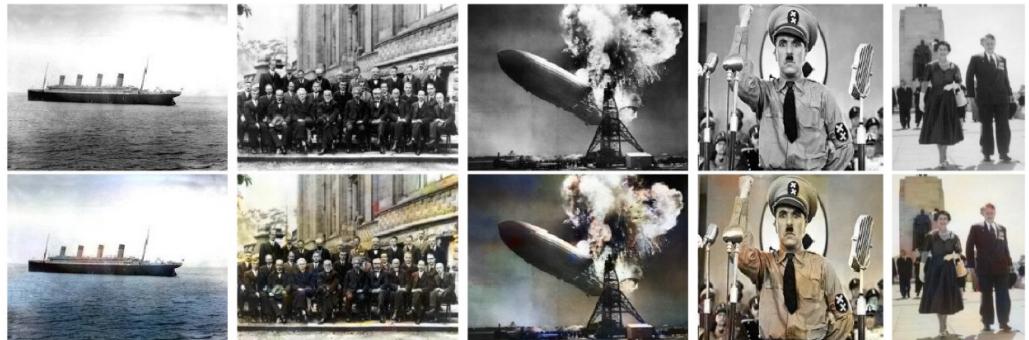
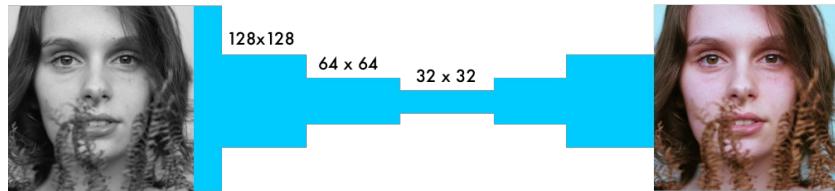
Image Colorization



Autoencoders – Applications

Image Colorization

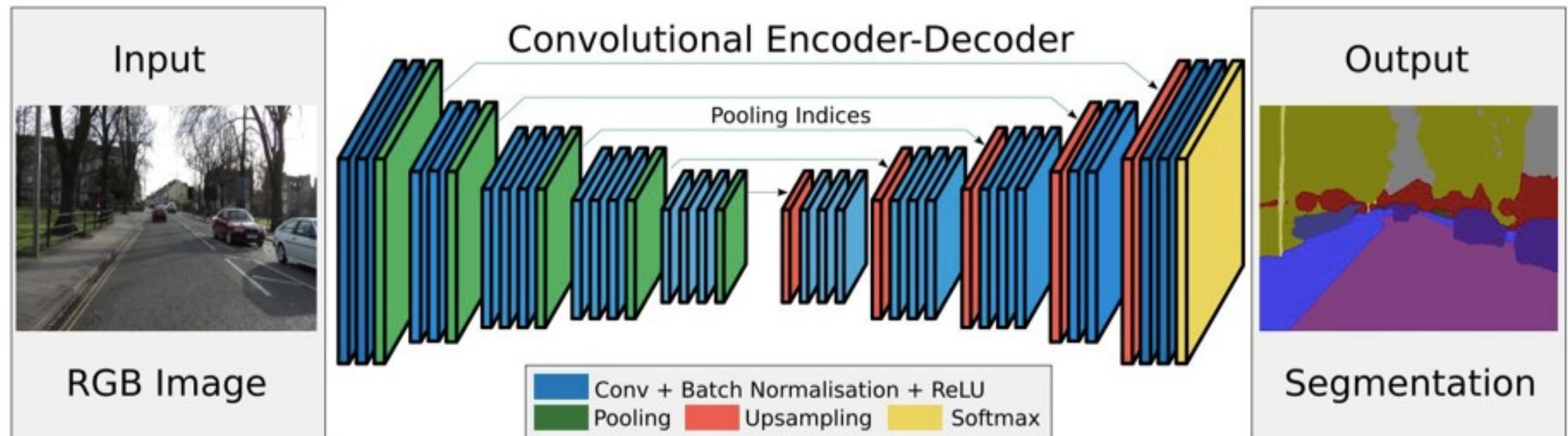
256x256



[Baldassarre, 2017]

Autoencoders – Applications

[Semantic] Segmentation



Autoencoders – Applications

Noise Reduction

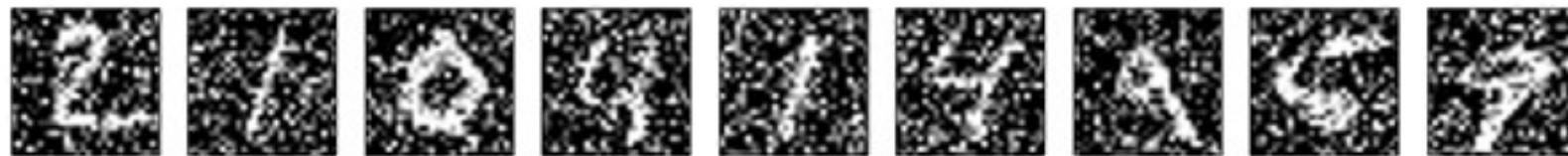
Noise
Removal



Autoencoders – Applications

Noise Reduction

Noise Removal



Watermark Removal

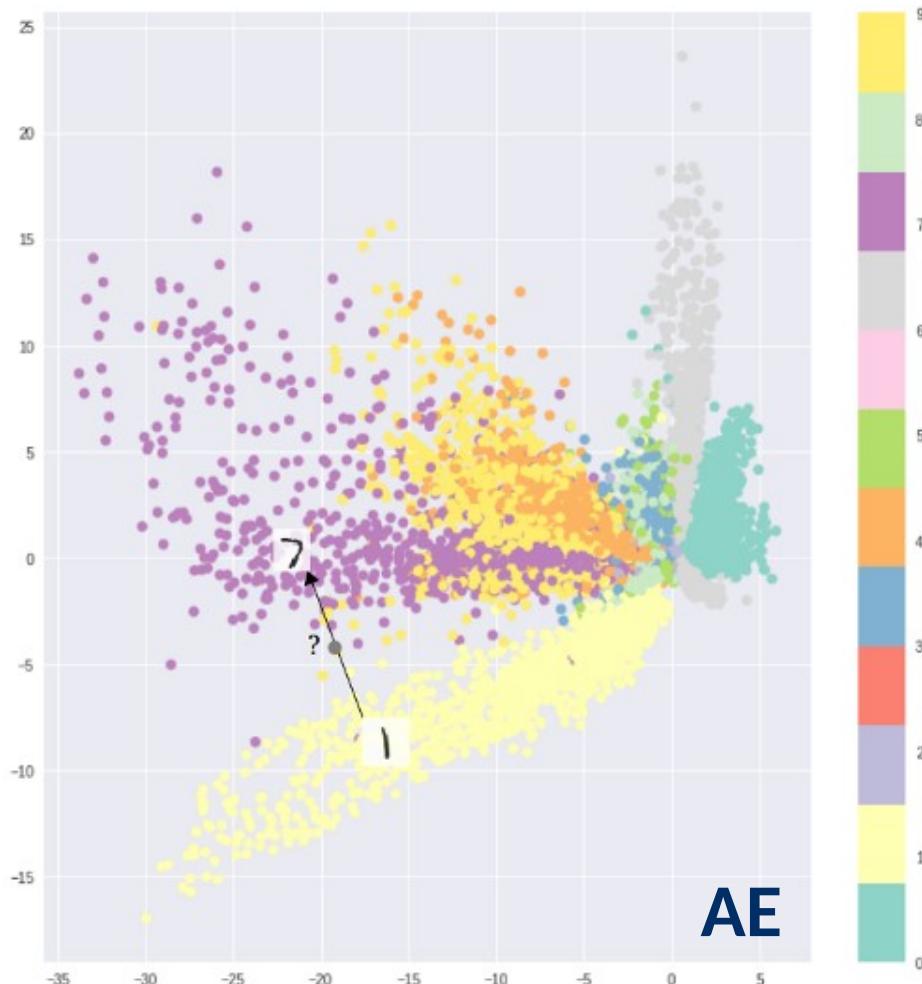


Variational Autoencoders

[autoencoders with continuous latent spaces]

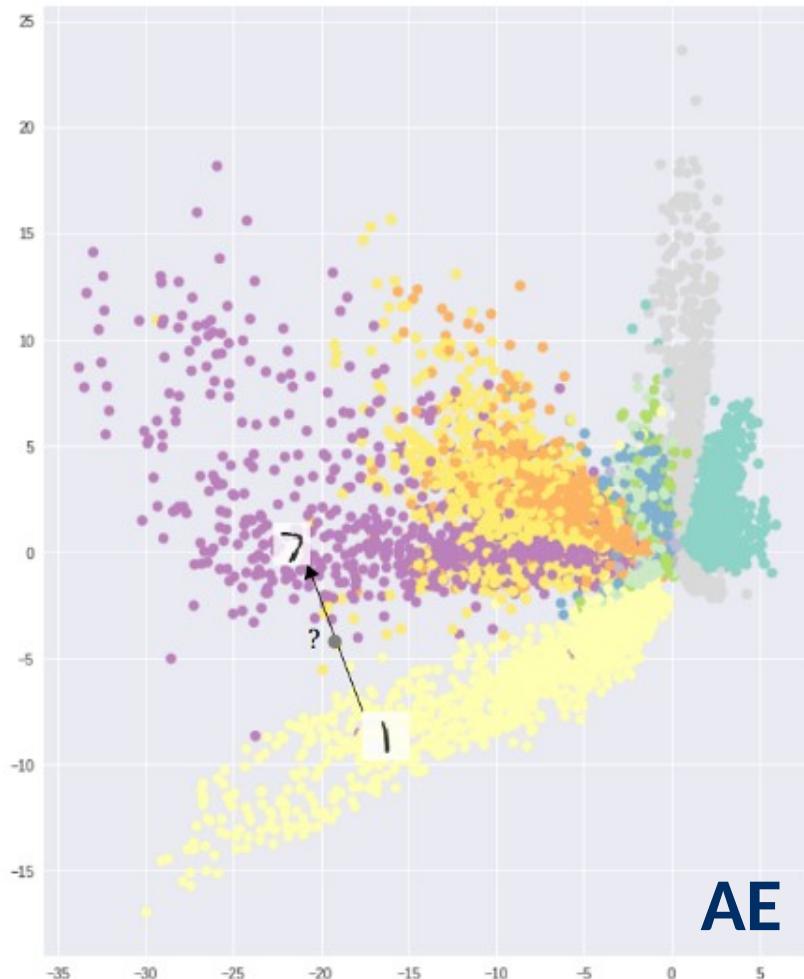
Autoencoders VS. Variational Autoencoders

Latent representations of MNIST digits

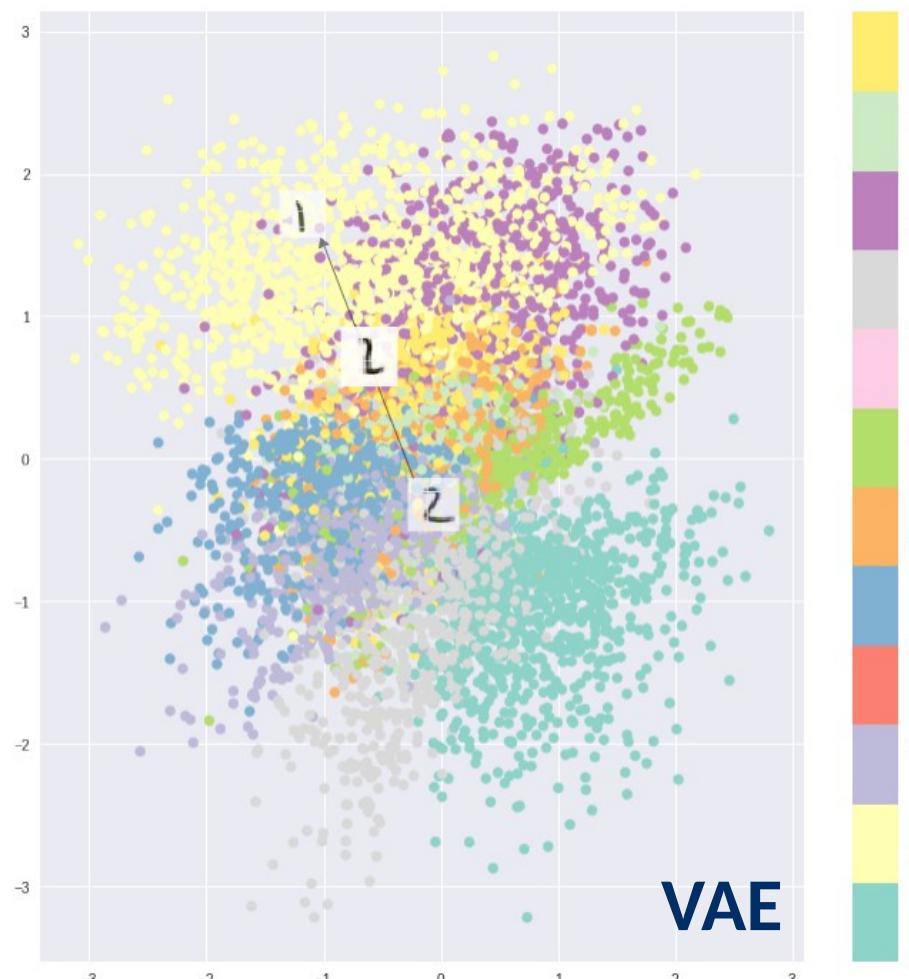


Autoencoders VS. Variational Autoencoders

Latent representations of MNIST digits



AE



VAE



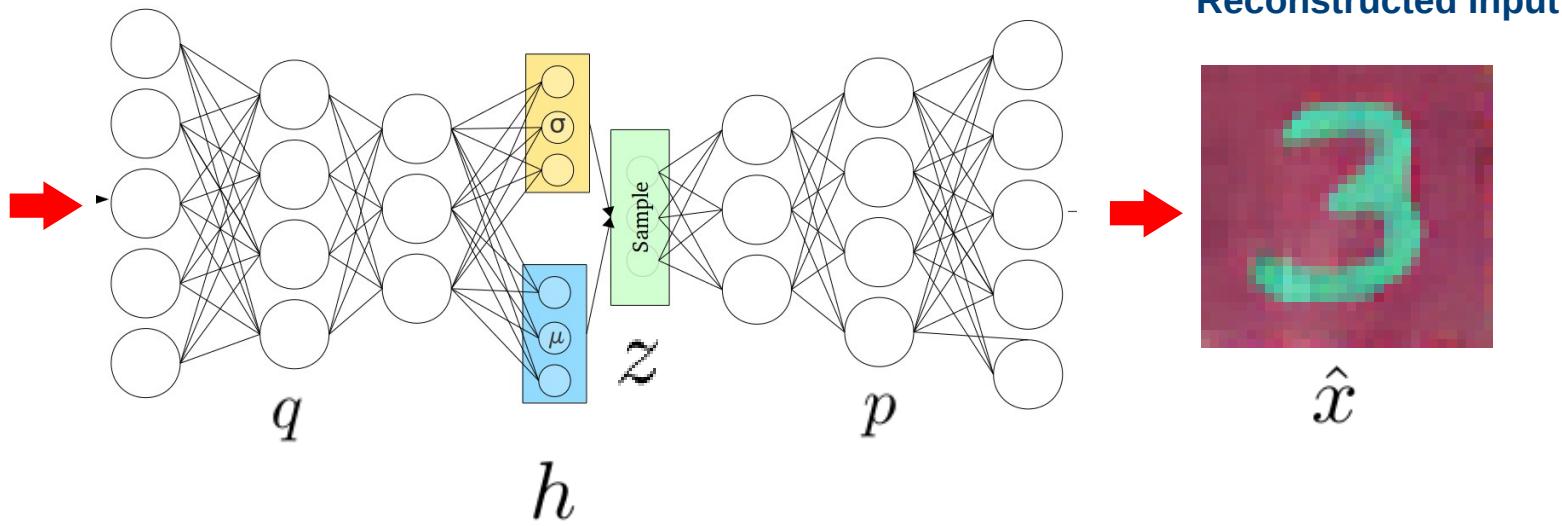
Variational Autoencoders

Mapping inputs to a distributions

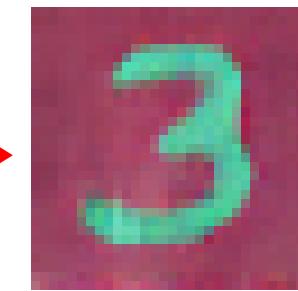
Original Input



x



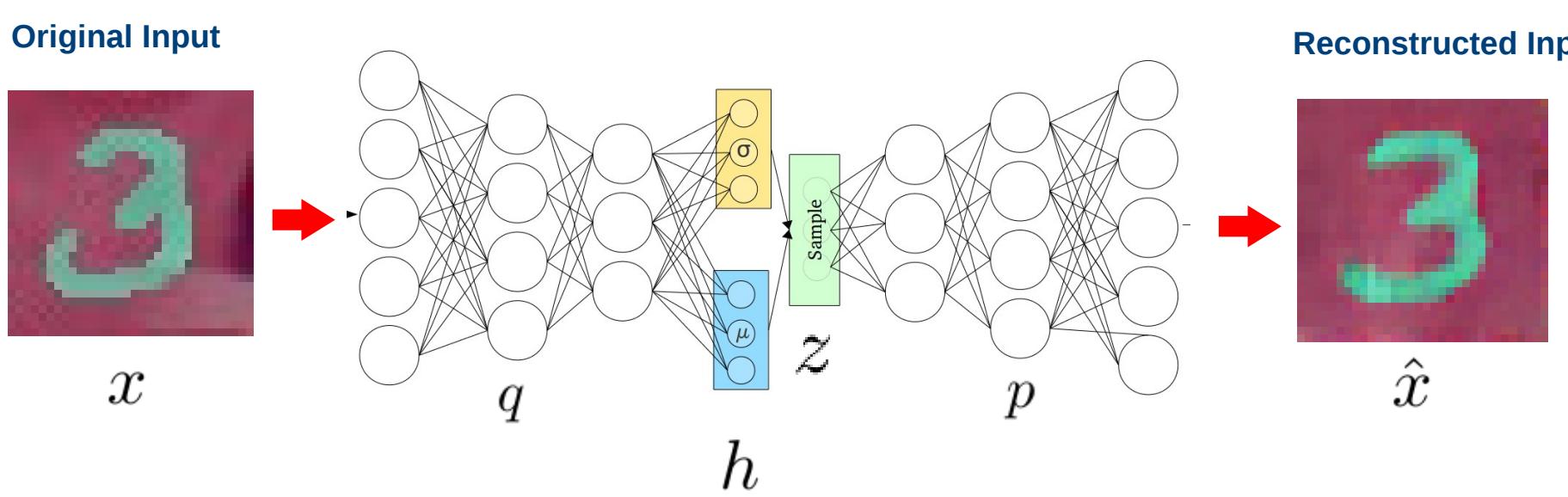
Reconstructed Input



\hat{x}

Variational Autoencoders

Mapping inputs to a distributions



- Bottleneck now includes components μ and σ .
- Generate variations of the input from a continuous latent space
- Decoder feeds from sample from distribution (μ, σ) .

Variational Autoencoders – Training

Desirable properties of the representation

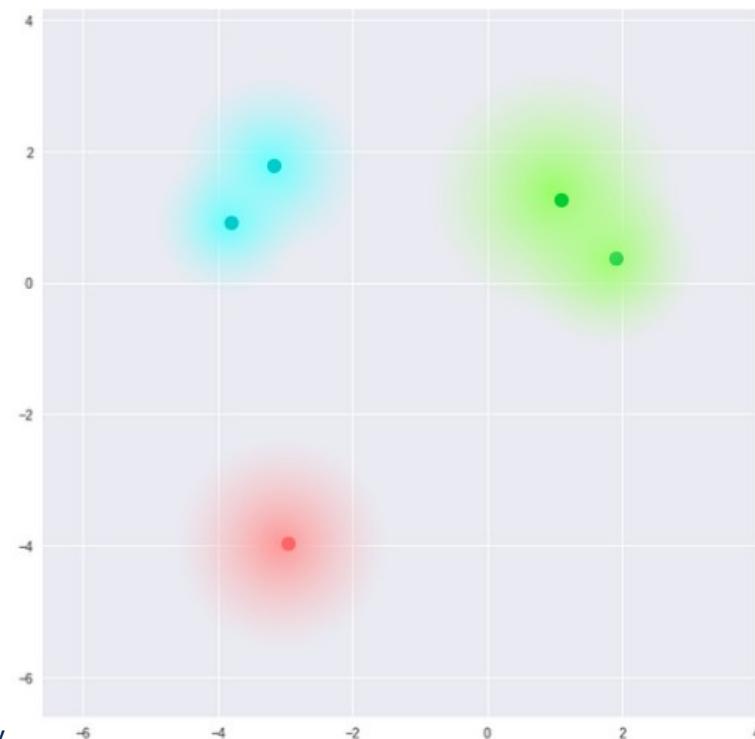
$$l(q) = \mathbb{E}_{z \sim q(z|x)} \log(p(x|z)) - D_{KL}((q(\mathbf{z}|x)||p(\mathbf{z}))$$

Variational Autoencoders – Training

Desirable properties of the representation

$$l(q) = \mathbb{E}_{z \sim q(z|x)} \log(p(x|z)) - D_{KL}((q(\mathbf{z}|x)||p(\mathbf{z}))$$

P1: Model local variations

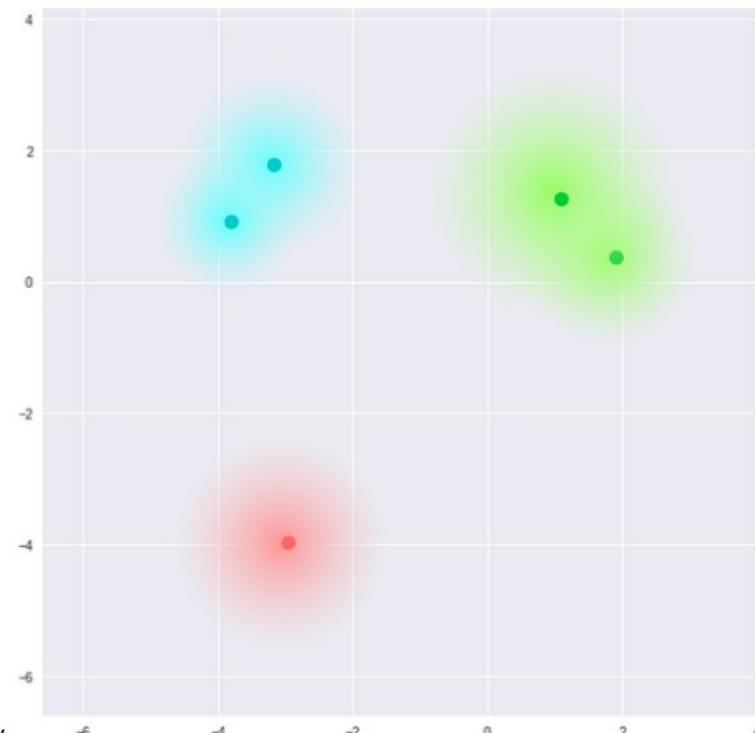


Variational Autoencoders – Training

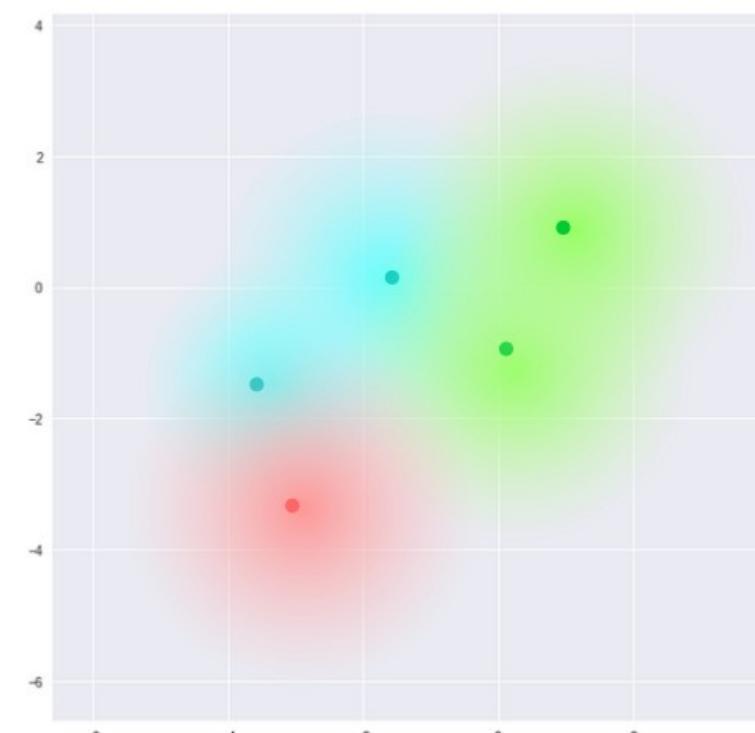
Desirable properties of the representation

$$l(q) = \mathbb{E}_{z \sim q(z|x)} \log(p(x|z)) - D_{KL}((q(\mathbf{z}|x)||p(\mathbf{z}))$$

P1: Model local variations



P2: Close encodings even when different

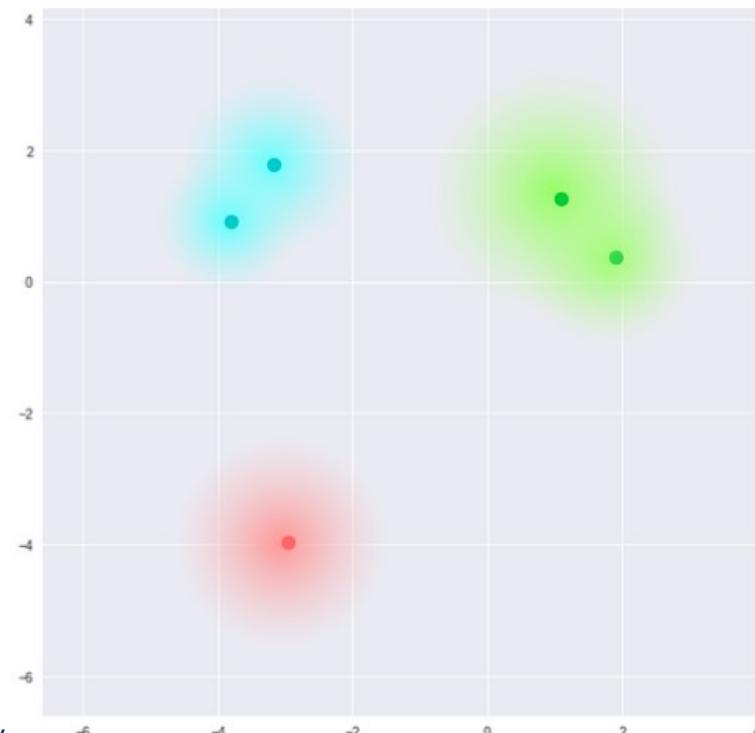


Variational Autoencoders – Training

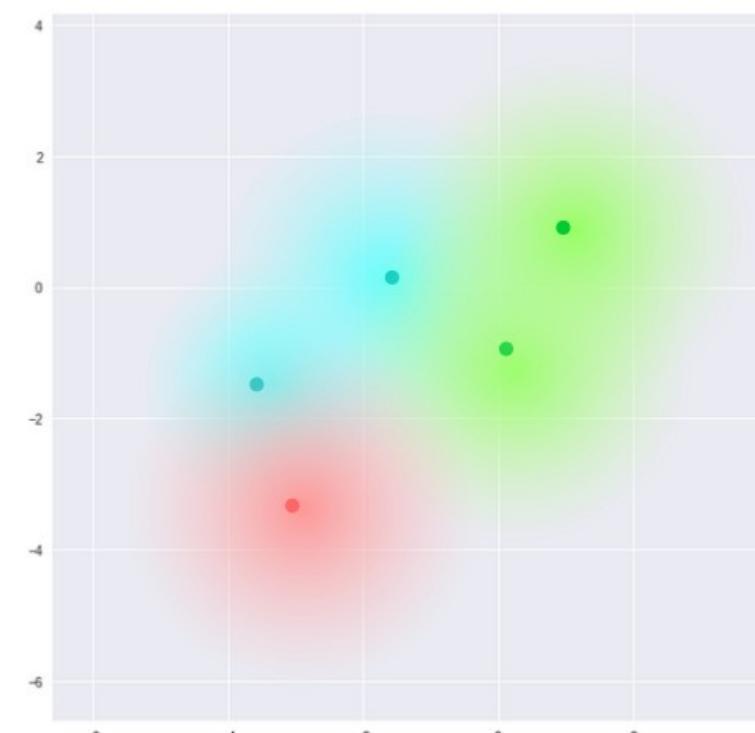
Desirable properties of the representation

$$l(q) = \mathbb{E}_{z \sim q(z|x)} \log(p(x|z)) - D_{KL}((q(\mathbf{z}|x)||p(\mathbf{z}))$$

P1: Model local variations



P2: Close encodings even when different

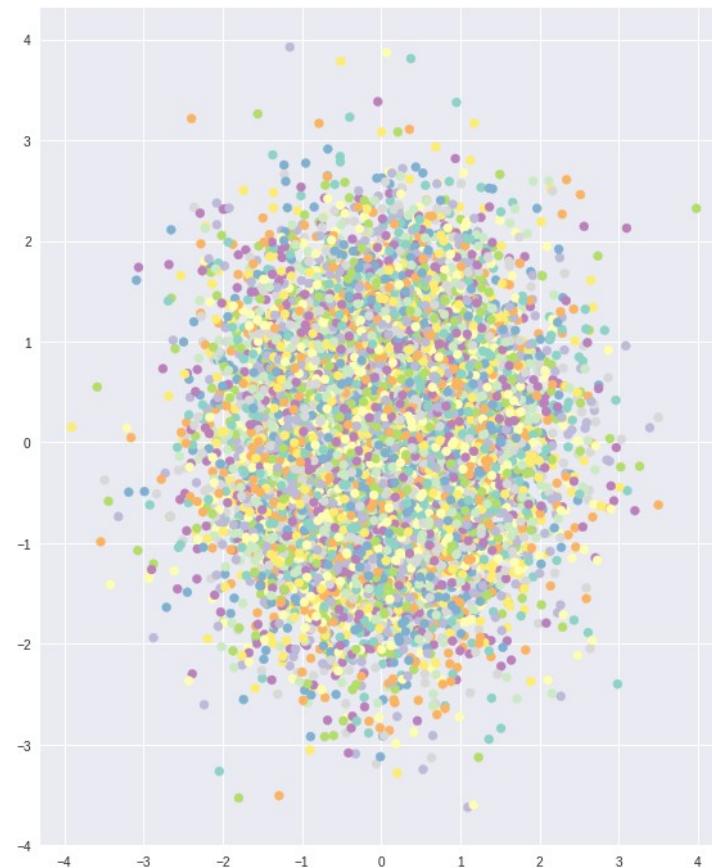


- P1 enabled by components (μ, σ)
- P2 enforced via the KL-divergence

Variational Autoencoders – Training

Desirable properties of the representation

$$l(q) = \mathbb{E}_{z \sim q(z|x)} \log(p(x|z)) - D_{KL}((q(\mathbf{z}|x)||p(\mathbf{z}))$$



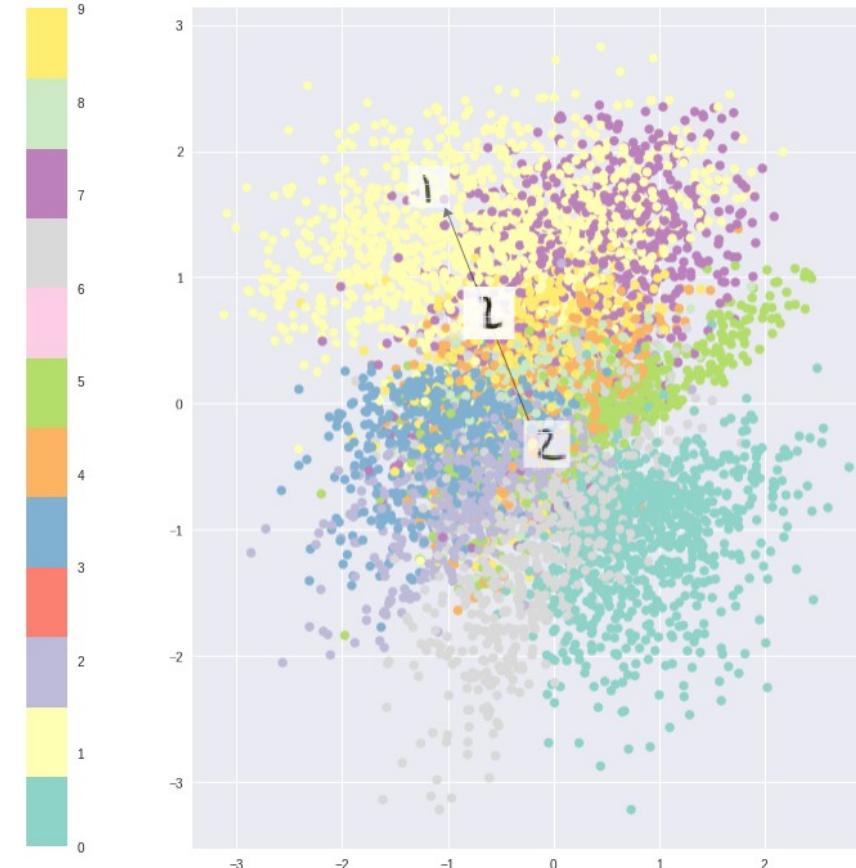
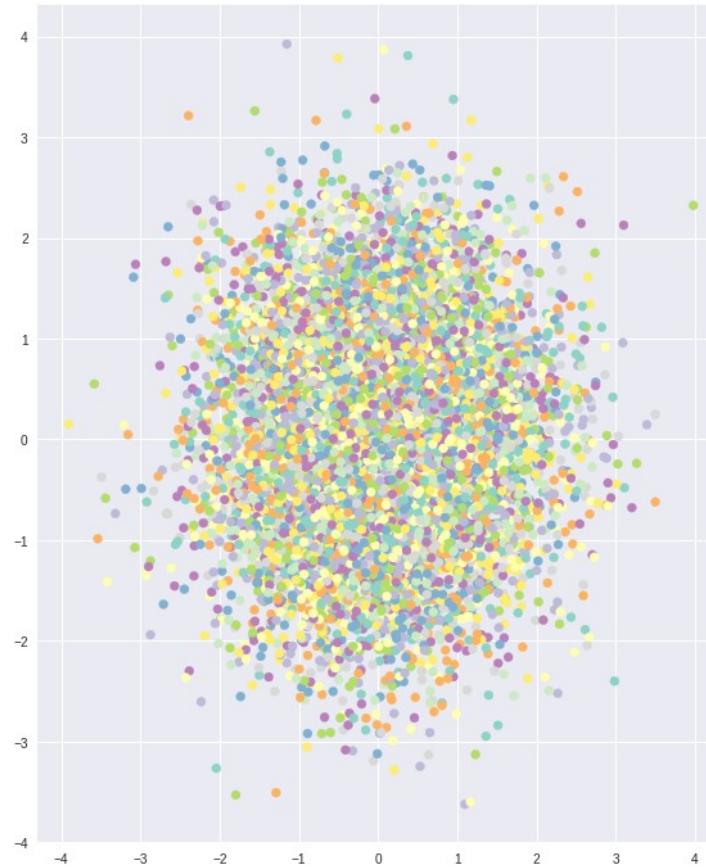
Optimizing only
the KL-divergence

Variational Autoencoders – Training

Desirable properties of the representation

$$l(q) = \mathbb{E}_{z \sim q(z|x)} \log(p(x|z)) - D_{KL}((q(\mathbf{z}|x)||p(\mathbf{z}))$$

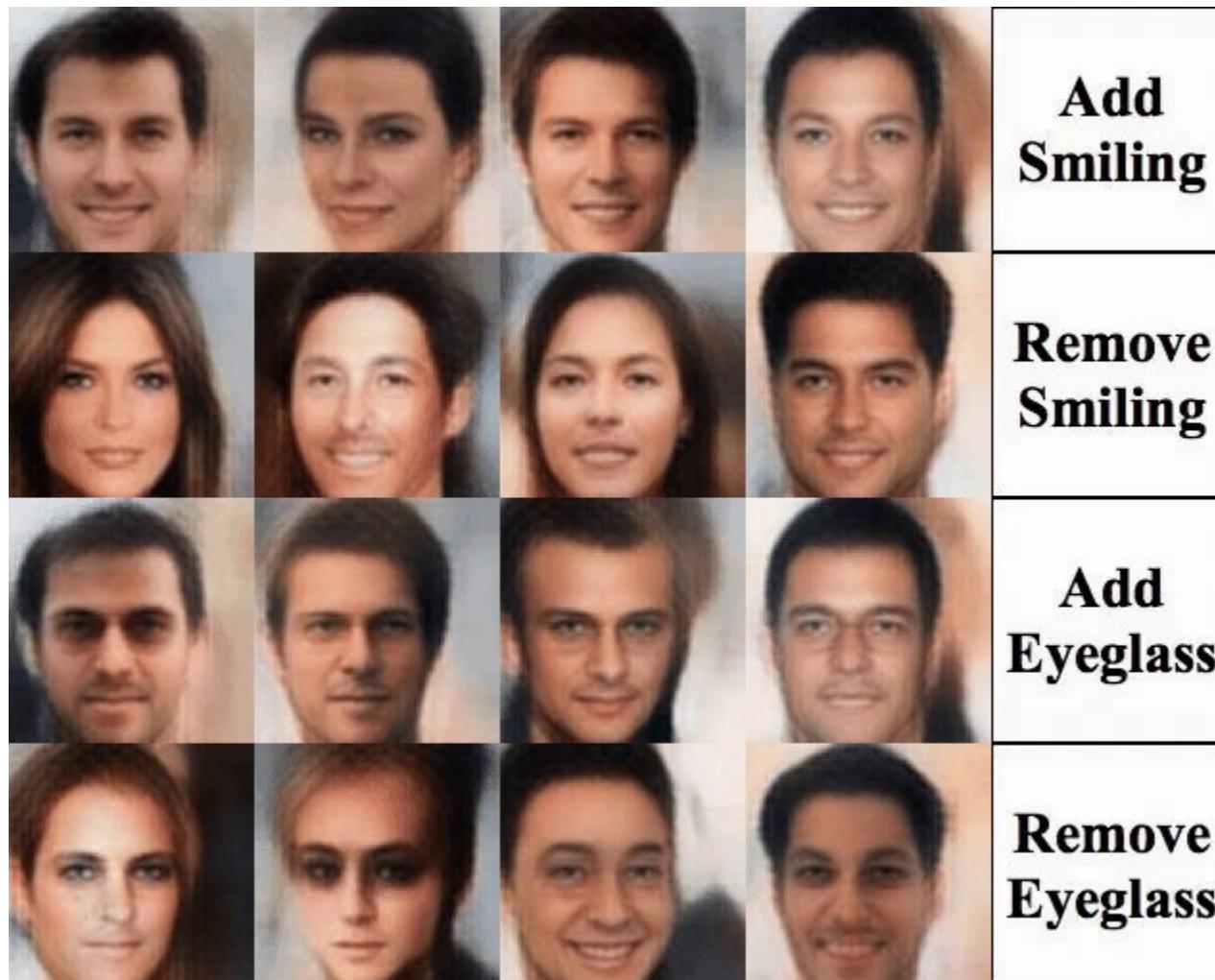
Optimizing only
the KL-divergence



Optimizing the
Reconstruction Loss
+ KL-divergence

Variational Autoencoders – Training

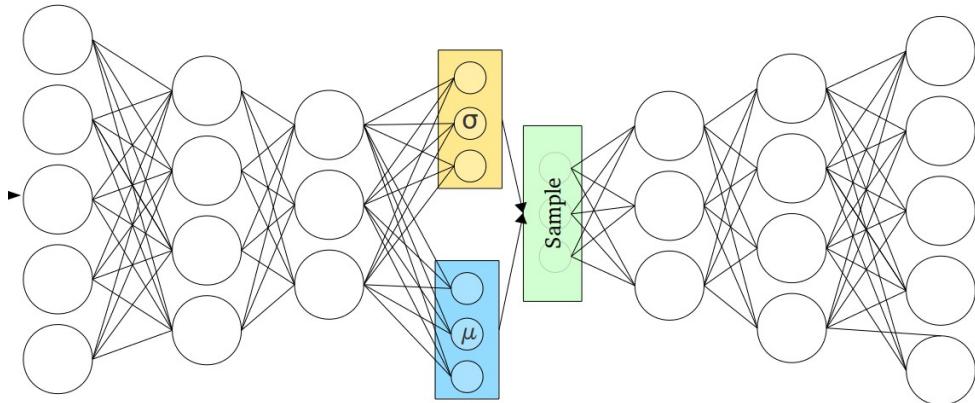
Interpolation accross the latent space



[Hou et al., 2017]

Variational Autoencoders – Training

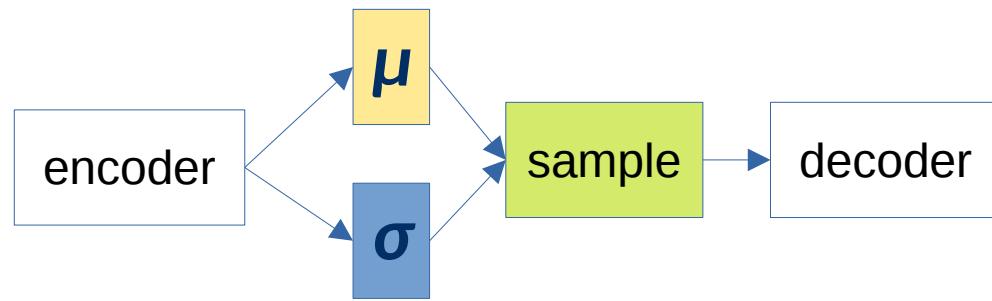
Difficulty → the sampling component is not differentiable



Variational Autoencoders – Training

Difficulty → the sampling component is not differentiable

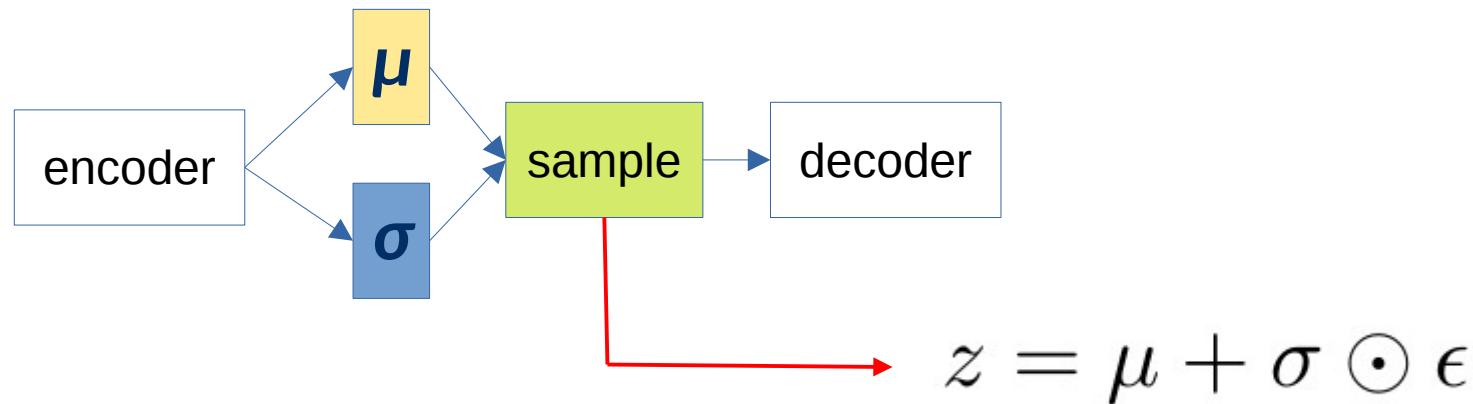
- Original



Variational Autoencoders – Training

Difficulty → the sampling component is not differentiable

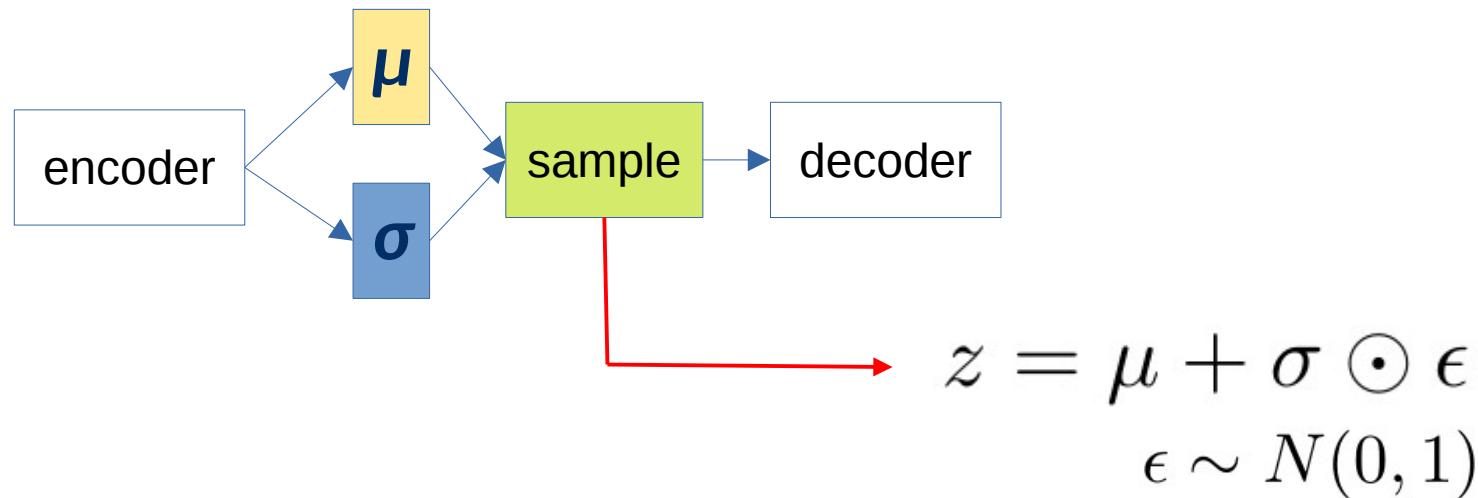
- Original



Variational Autoencoders – Training

Difficulty → the sampling component is not differentiable

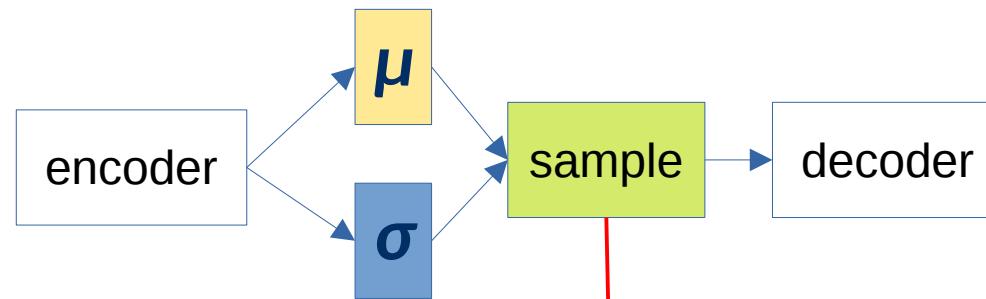
- Original



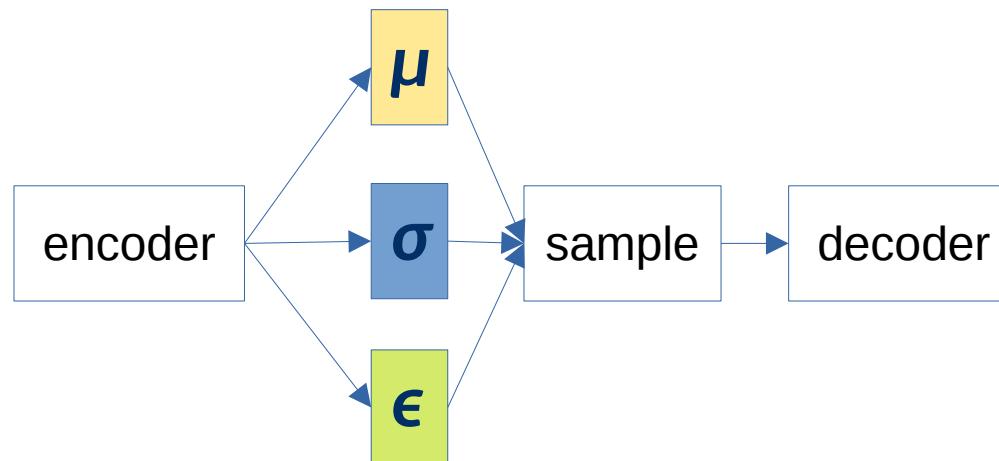
Variational Autoencoders – Training

Difficulty → the sampling component is not differentiable

- Original



- Reparametrized

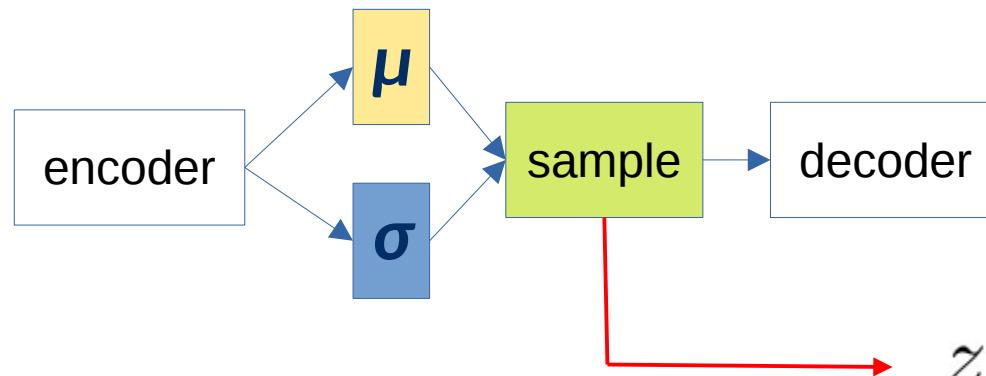


$$z = \mu + \sigma \odot \epsilon$$
$$\epsilon \sim N(0, 1)$$

Variational Autoencoders – Training

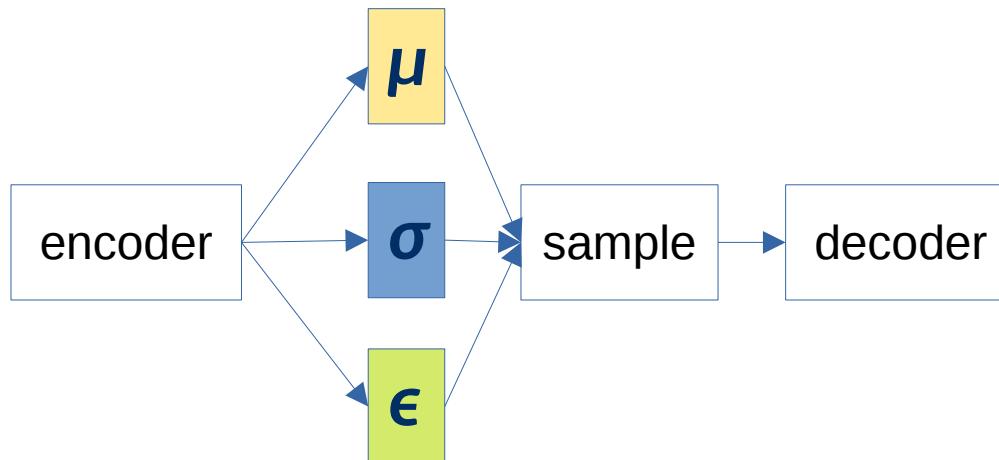
Difficulty → the sampling component is not differentiable

- Original



$$z = \mu + \sigma \odot \epsilon$$
$$\epsilon \sim N(0, 1)$$

- Reparametrized



- No backprop through ϵ
- ϵ remains stochastic.
- (μ, σ) are learnable

Generative Adversarial Networks

[*something-something-GAN*]

Generative Adversarial Networks (GANs)

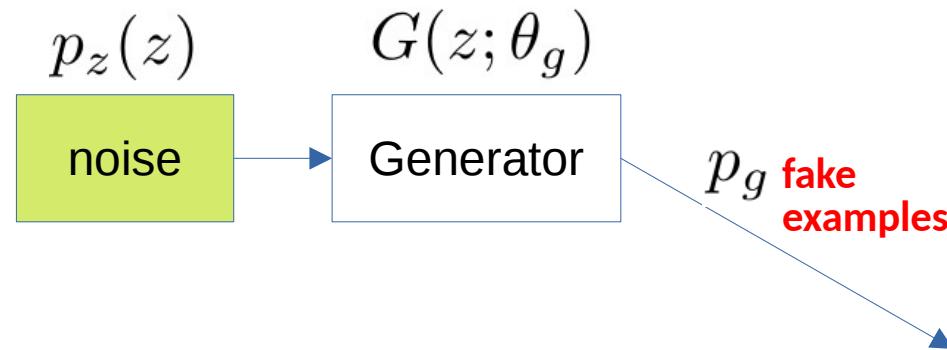
[Goodfellow et al., 2014]

Adversarial Training with Neural Networks

Generative Adversarial Networks (GANs)

[Goodfellow et al., 2014]

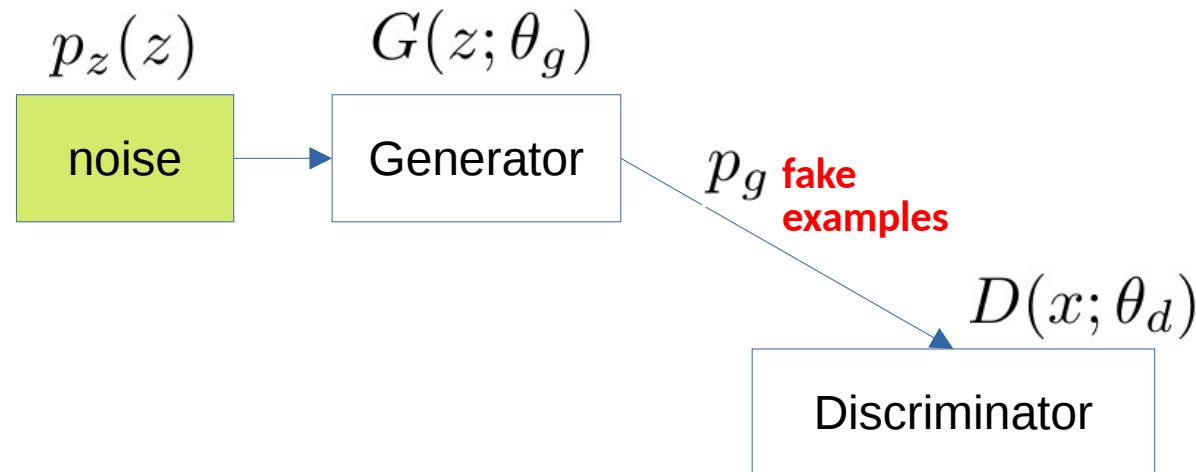
Adversarial Training with Neural Networks



Generative Adversarial Networks (GANs)

[Goodfellow et al., 2014]

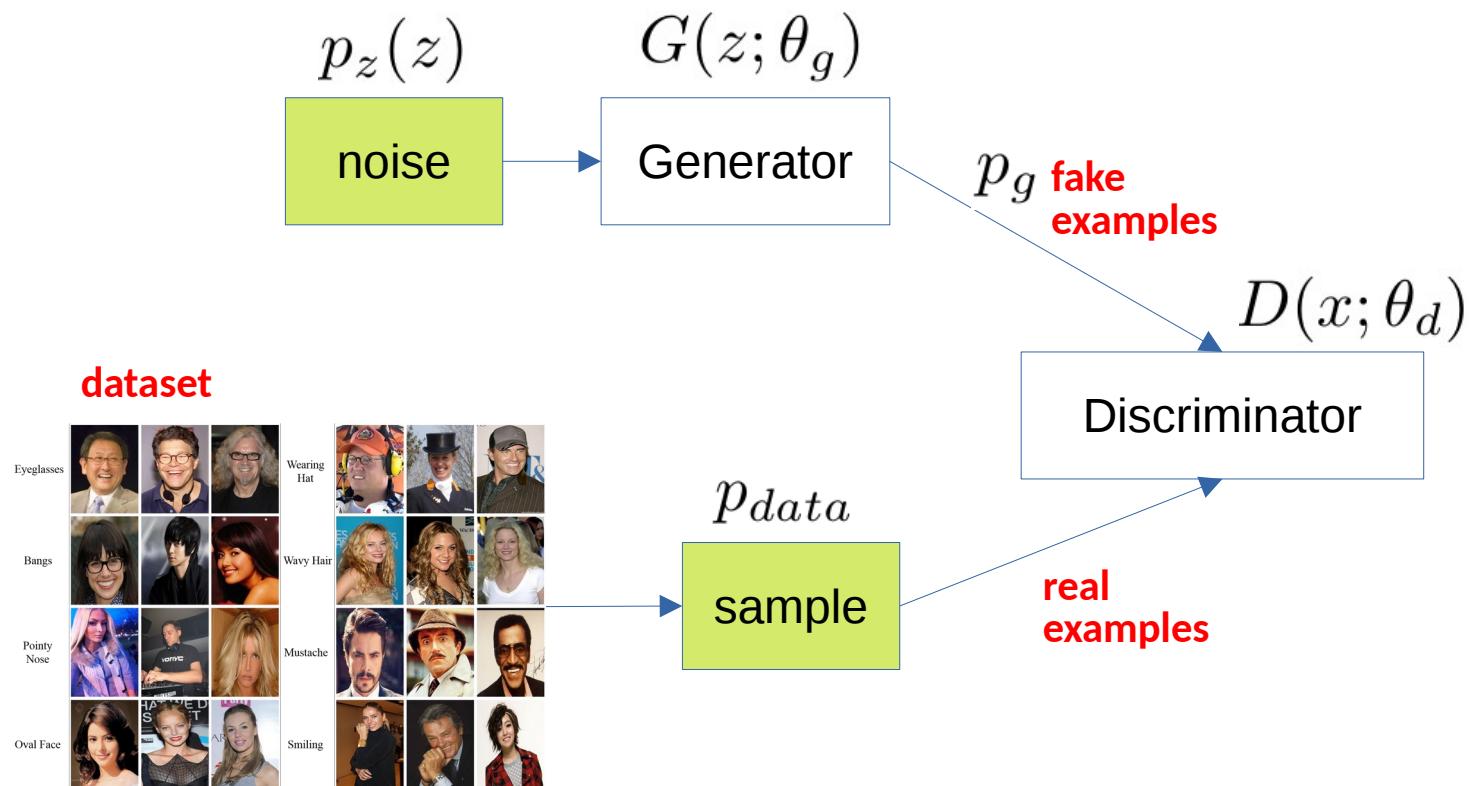
Adversarial Training with Neural Networks



Generative Adversarial Networks (GANs)

[Goodfellow et al., 2014]

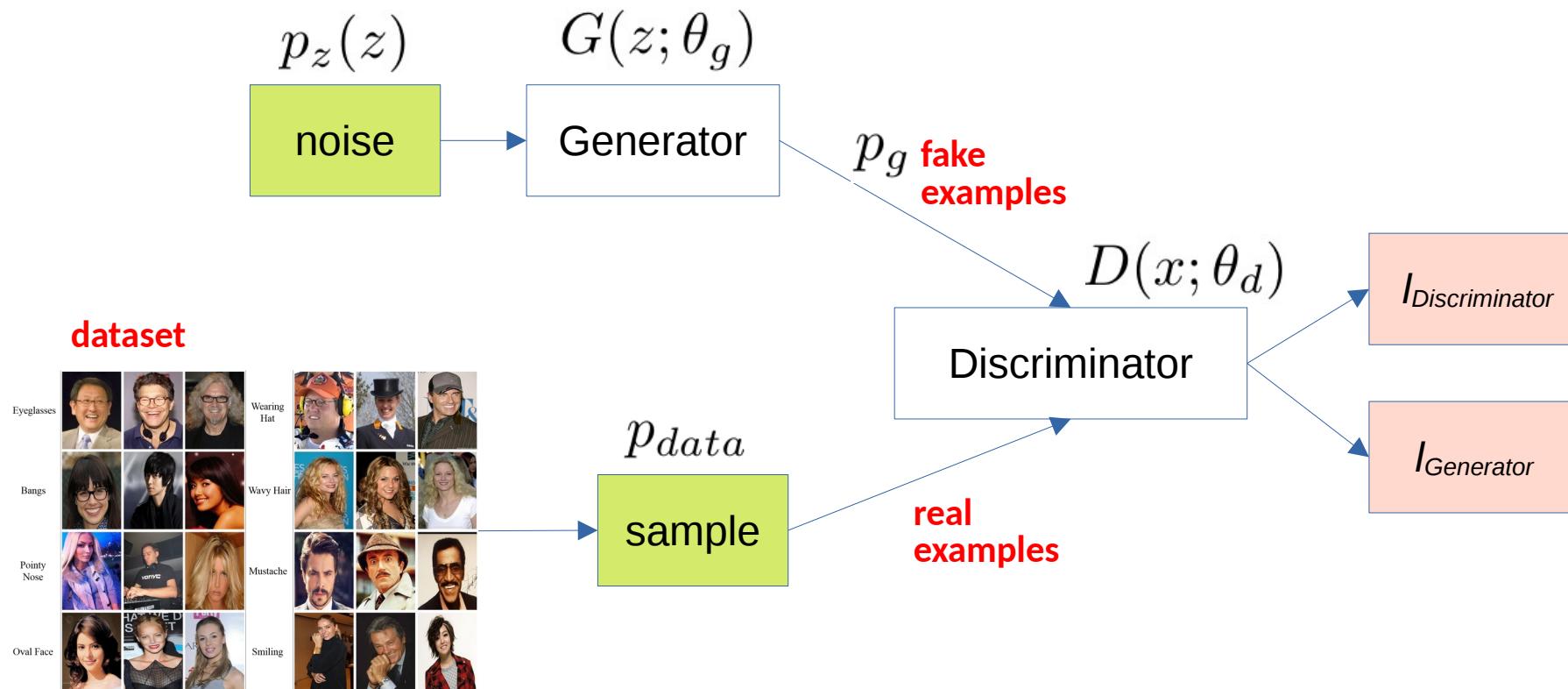
Adversarial Training with Neural Networks



Generative Adversarial Networks (GANs)

[Goodfellow et al., 2014]

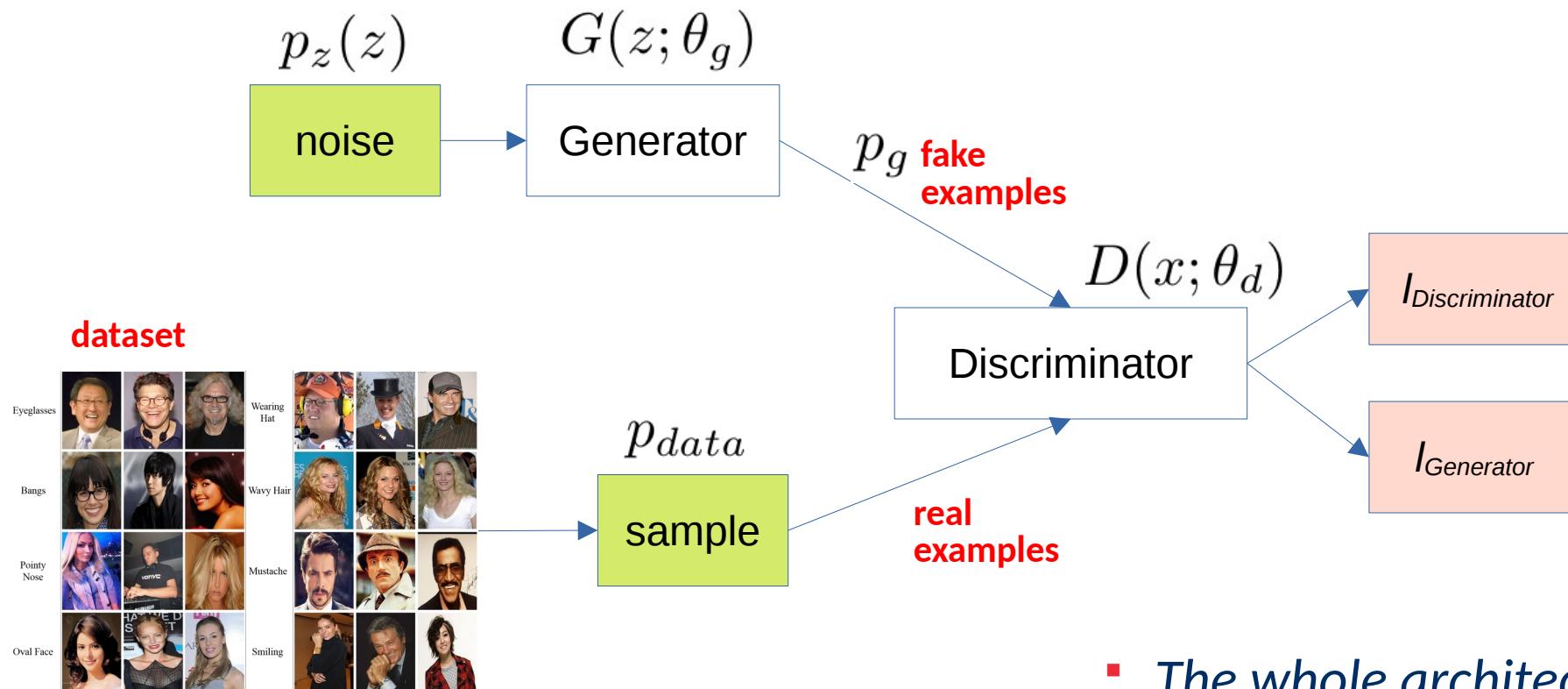
Adversarial Training with Neural Networks



Generative Adversarial Networks (GANs)

[Goodfellow et al., 2014]

Adversarial Training with Neural Networks

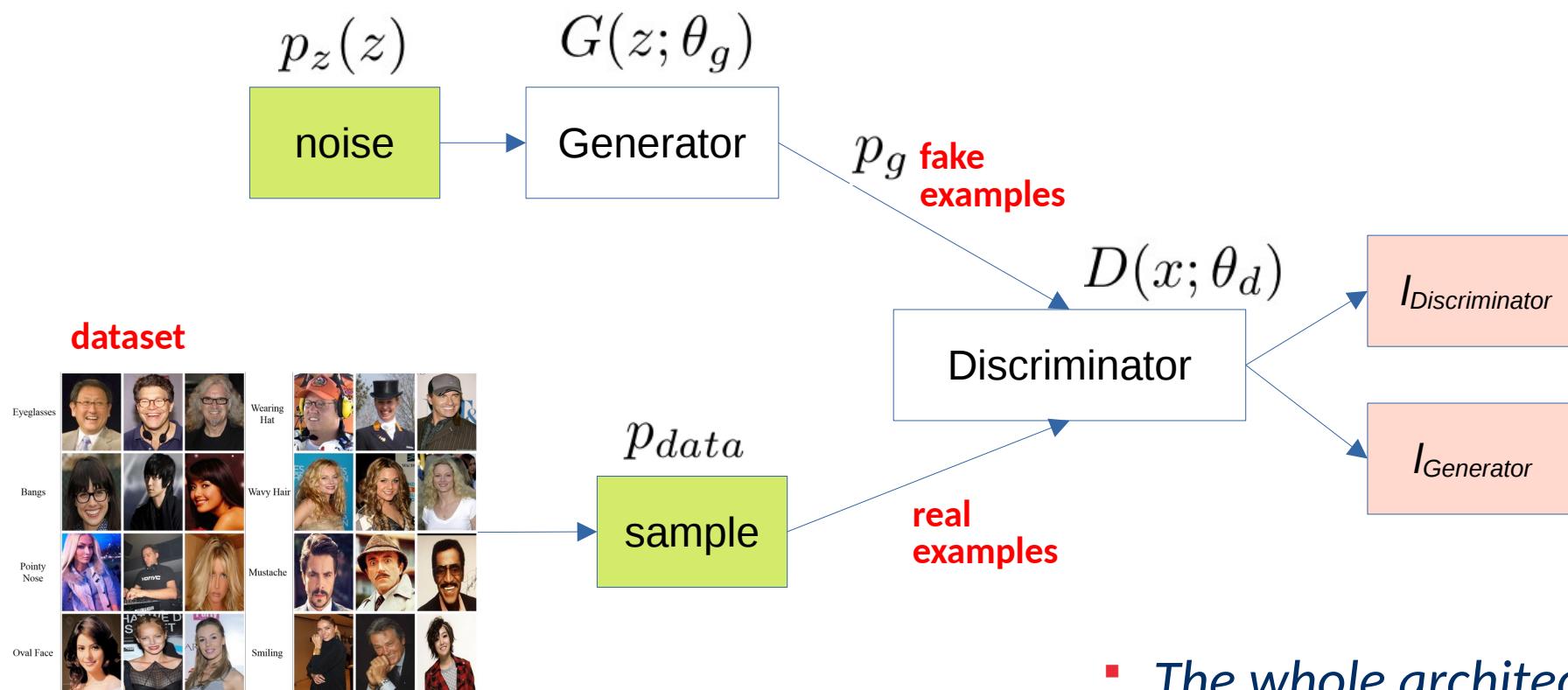


- *The whole architecture ensemble → GAN*

Generative Adversarial Networks (GANs)

[Goodfellow et al., 2014]

Adversarial Training with Neural Networks

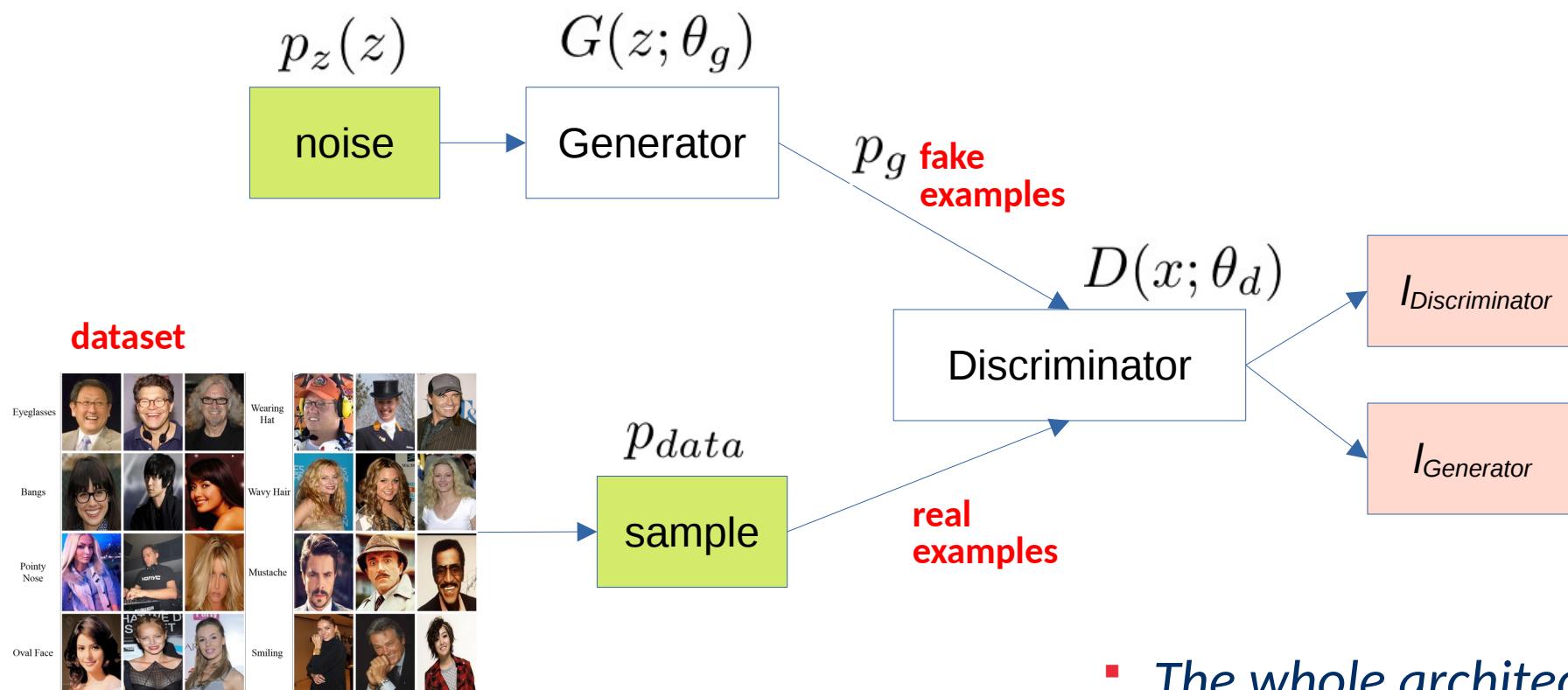


- The whole architecture ensemble → GAN

Generative Adversarial Networks (GANs)

[Goodfellow et al., 2014]

Adversarial Training with Neural Networks



- The whole architecture ensemble → GAN

Q: How would the Generator and Discriminator behave when reaching equilibrium?

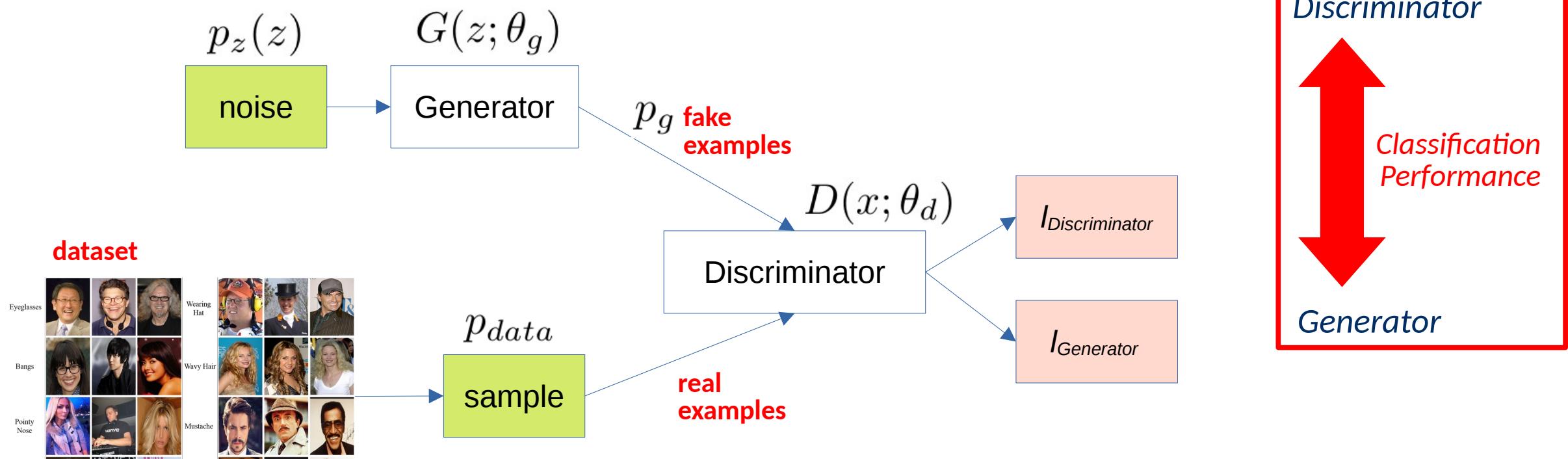
ok, but...
**How do we enforce the
competing behaviour?**



Generative Adversarial Networks – Training

[Goodfellow et al., 2014]

Adversarial Training with Neural Networks

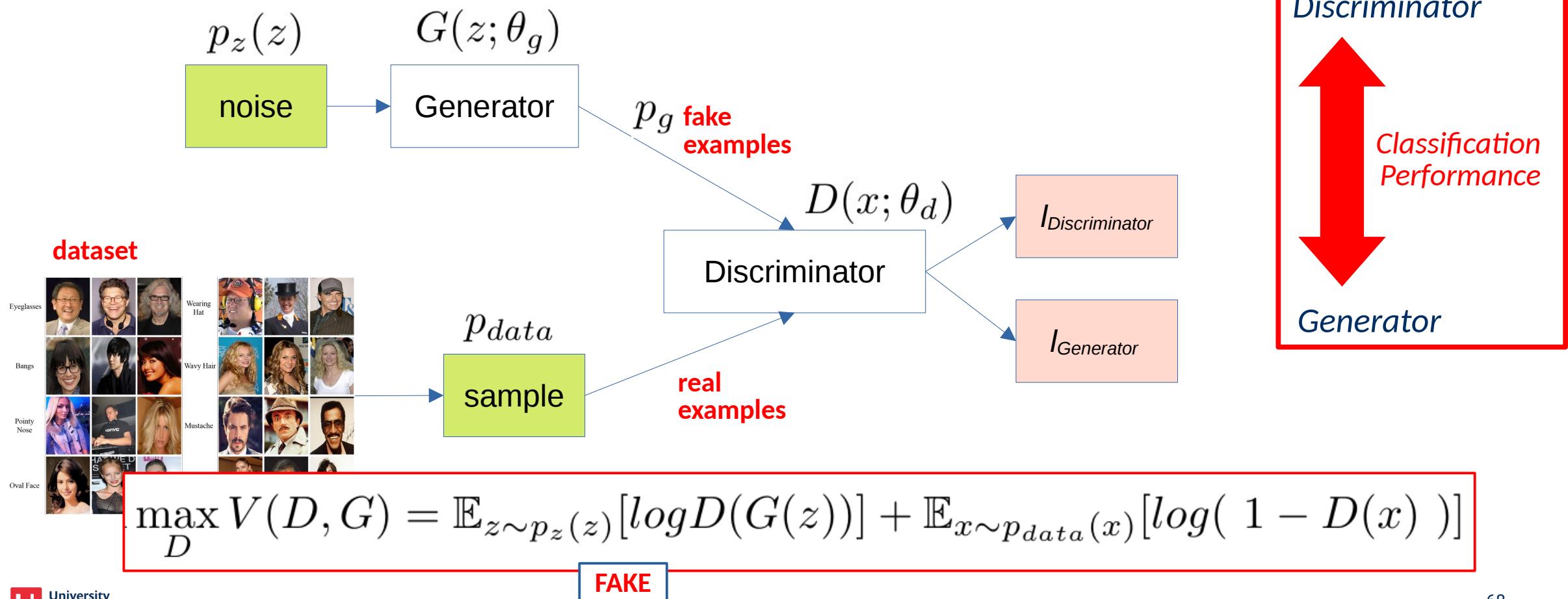


$$\min_G \max_D V(D, G) = \mathbb{E}_{z \sim p_z(z)} [\log D(G(z))] + \mathbb{E}_{x \sim p_{data}(x)} [\log(1 - D(x))]$$

Generative Adversarial Networks – Training

[Goodfellow et al., 2014]

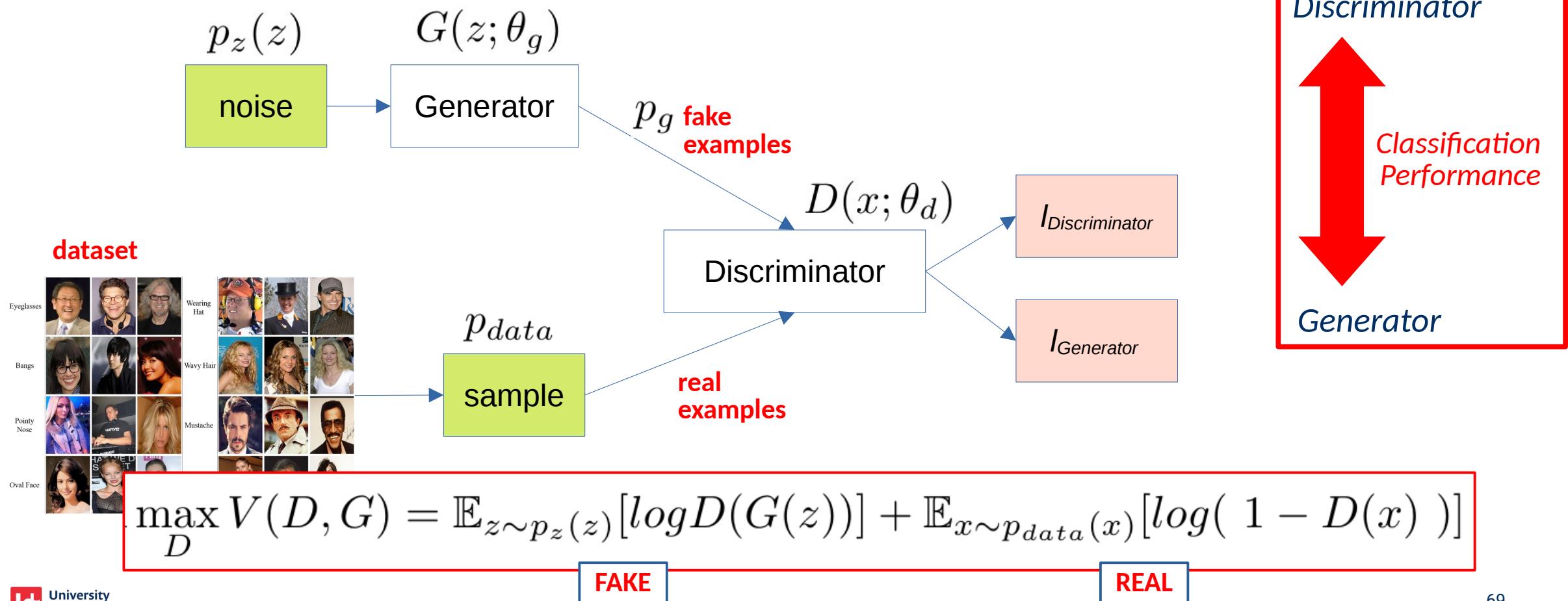
Adversarial Training with Neural Networks



Generative Adversarial Networks – Training

[Goodfellow et al., 2014]

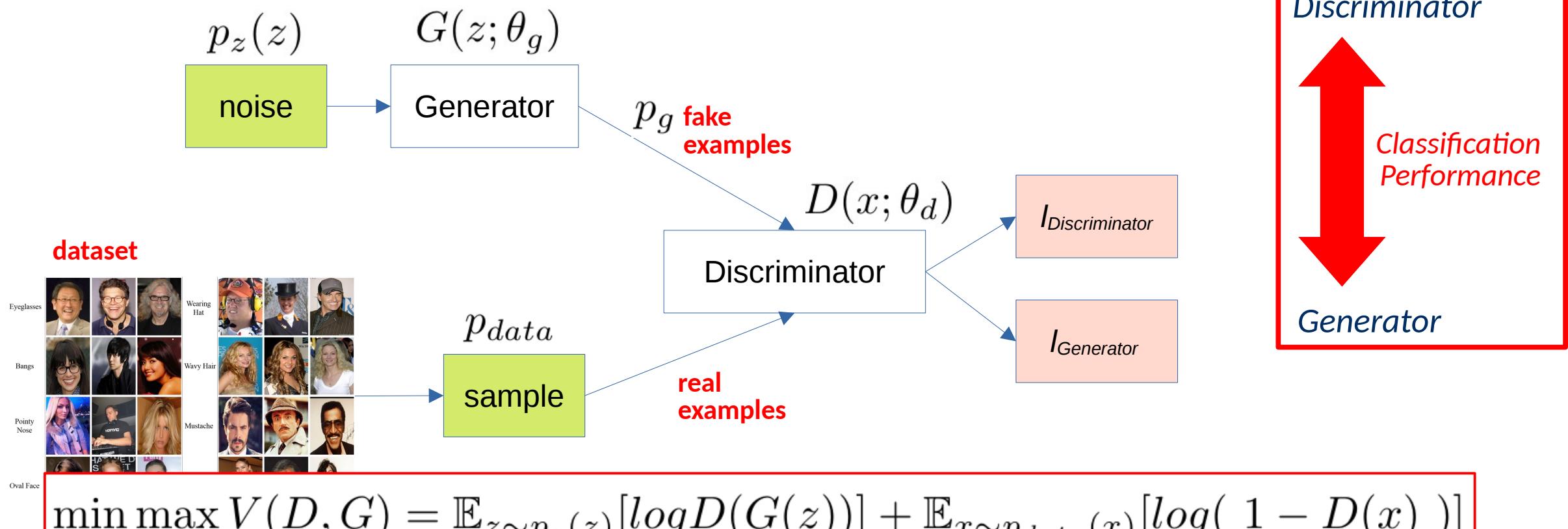
Adversarial Training with Neural Networks



Generative Adversarial Networks – Training

[Goodfellow et al., 2014]

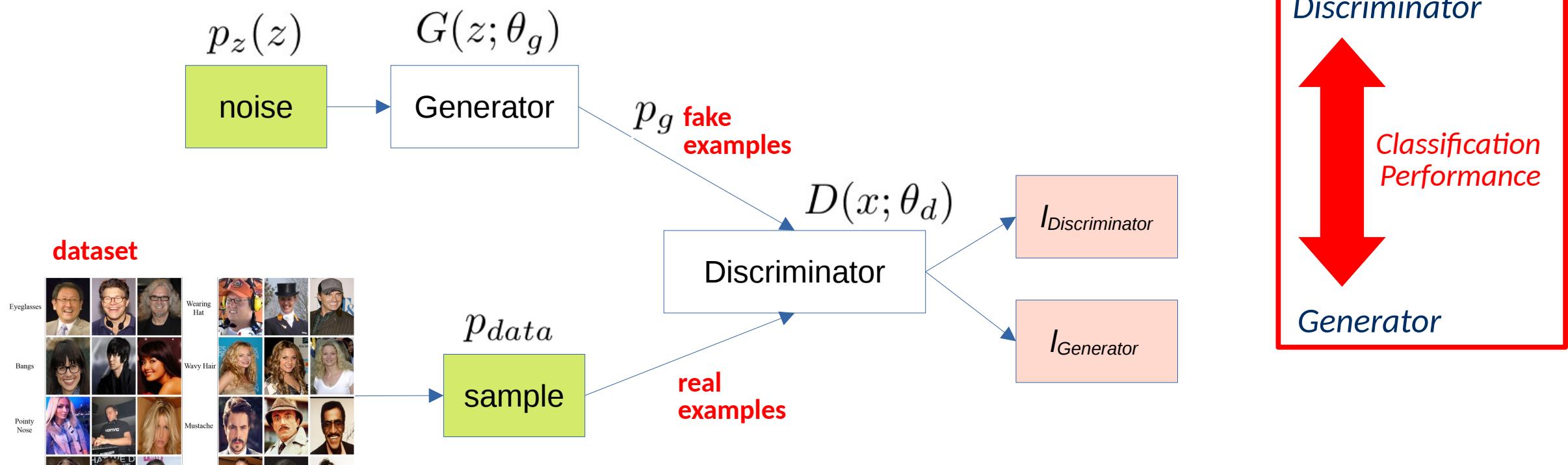
Adversarial Training with Neural Networks



Generative Adversarial Networks – Training

[Goodfellow et al., 2014]

Adversarial Training with Neural Networks



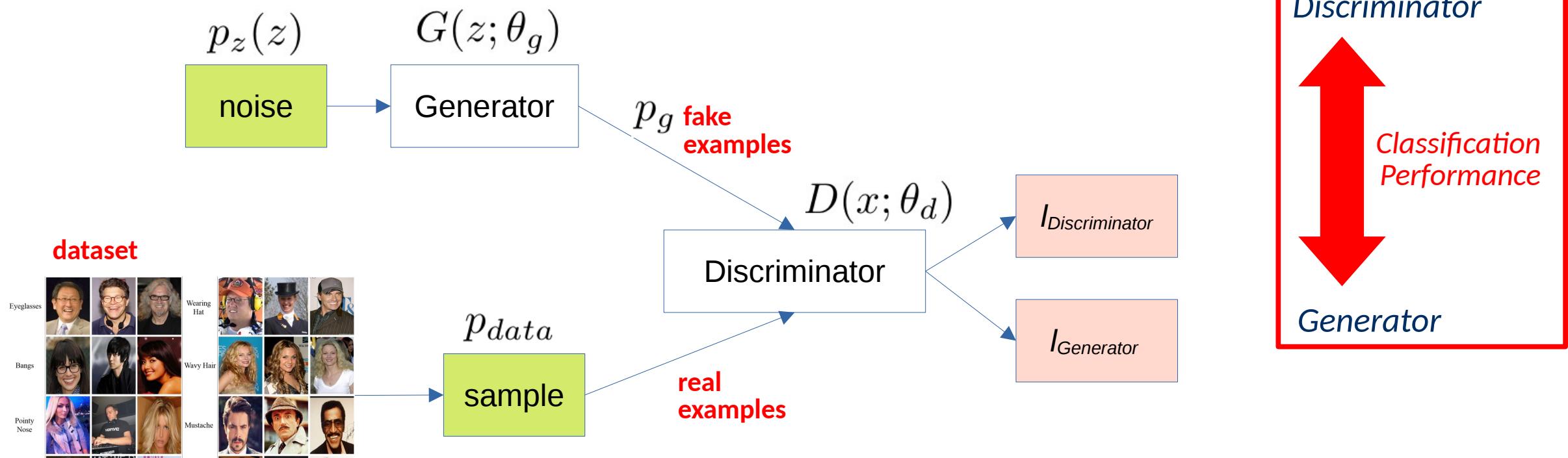
$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Alternatively Formulation

Generative Adversarial Networks – Training

[Goodfellow et al., 2014]

Adversarial Training with Neural Networks



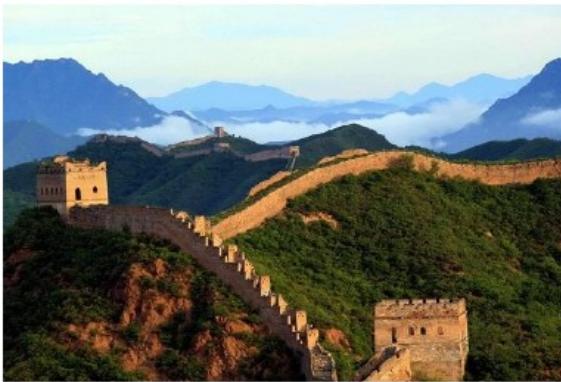
$$\min_G \max_D V(D, G) = \mathbb{E}_{z \sim p_z(z)} [\log D(G(z))] + \mathbb{E}_{x \sim p_{data}(x)} [\log(1 - D(x))]$$

Applications

[*something-something-GAN*]

Generative Adversarial Networks – Applications

Style Transfer

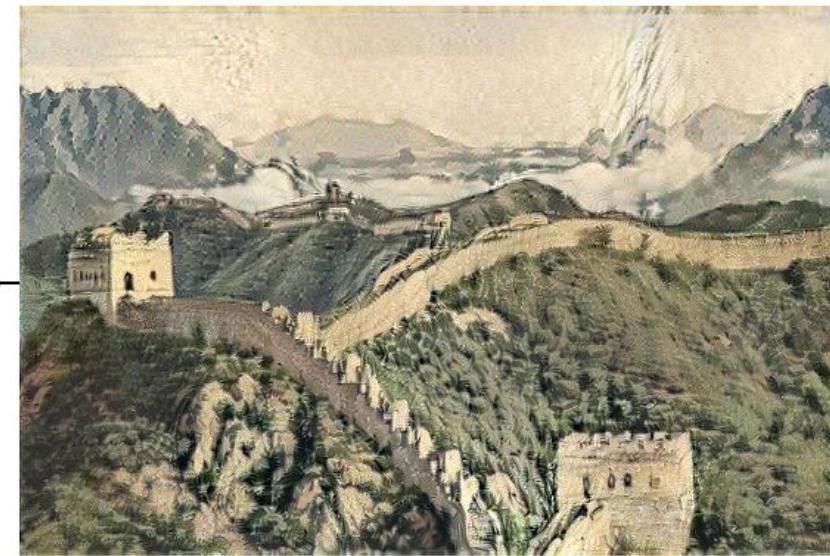


Input Content



Input Style

Neural Style Transfer



Output

[Jing et al., arXiv:1705.04058]

Generative Adversarial Networks – Applications

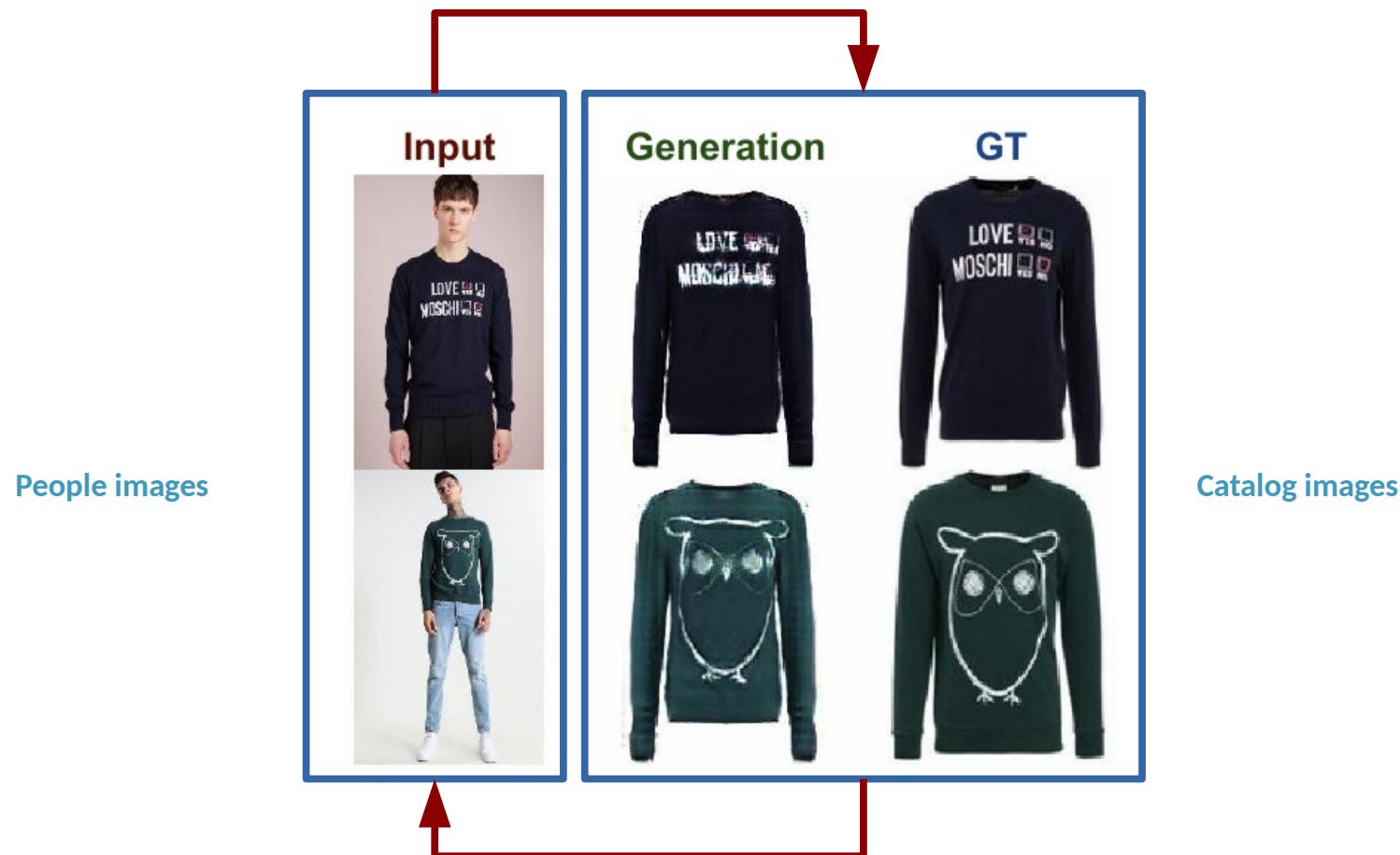
Style Transfer



[Zhu et al., ICCV'17]

Generative Adversarial Networks – Applications

Transferring shape, Preserving Style



Generative Adversarial Networks – Applications

Transferring shape, Preserving Style



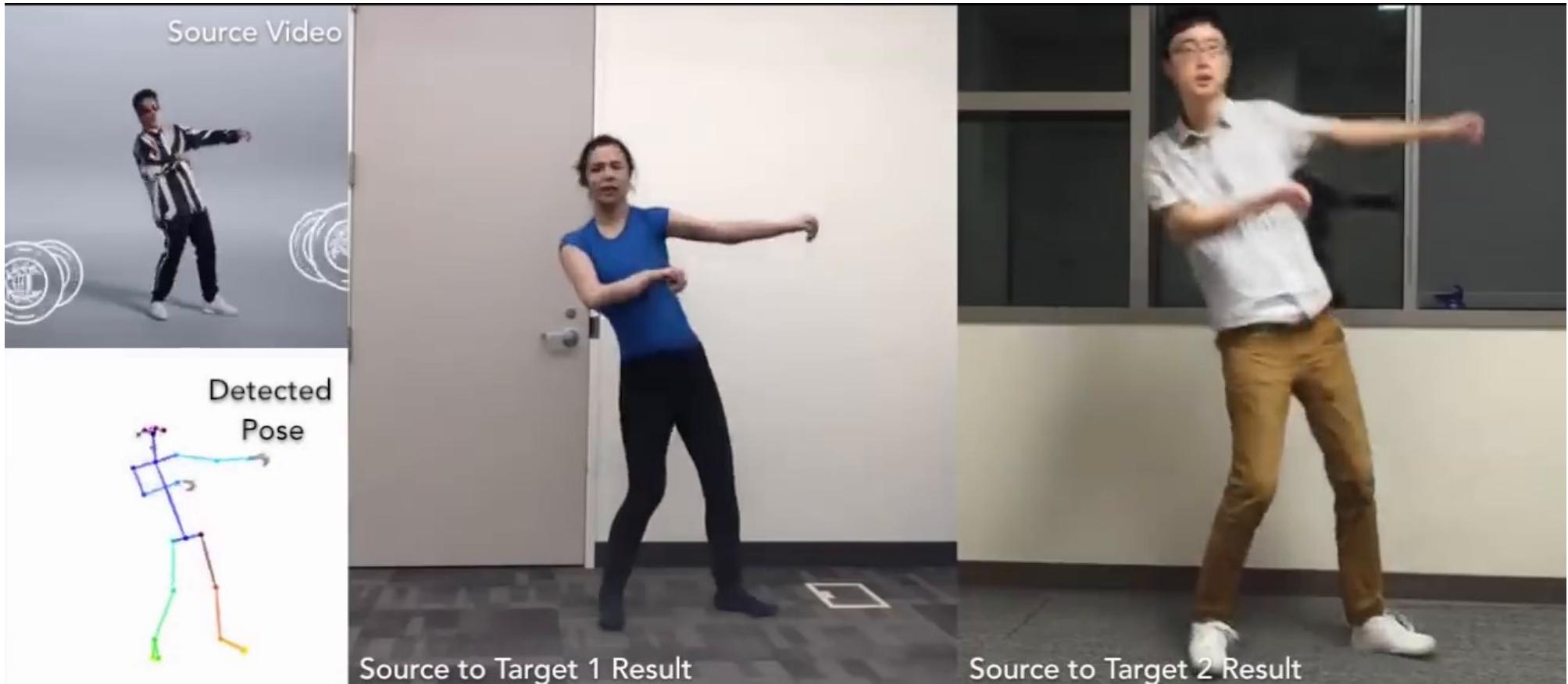
Generative Adversarial Networks – Applications

Transferring shape, Preserving Style



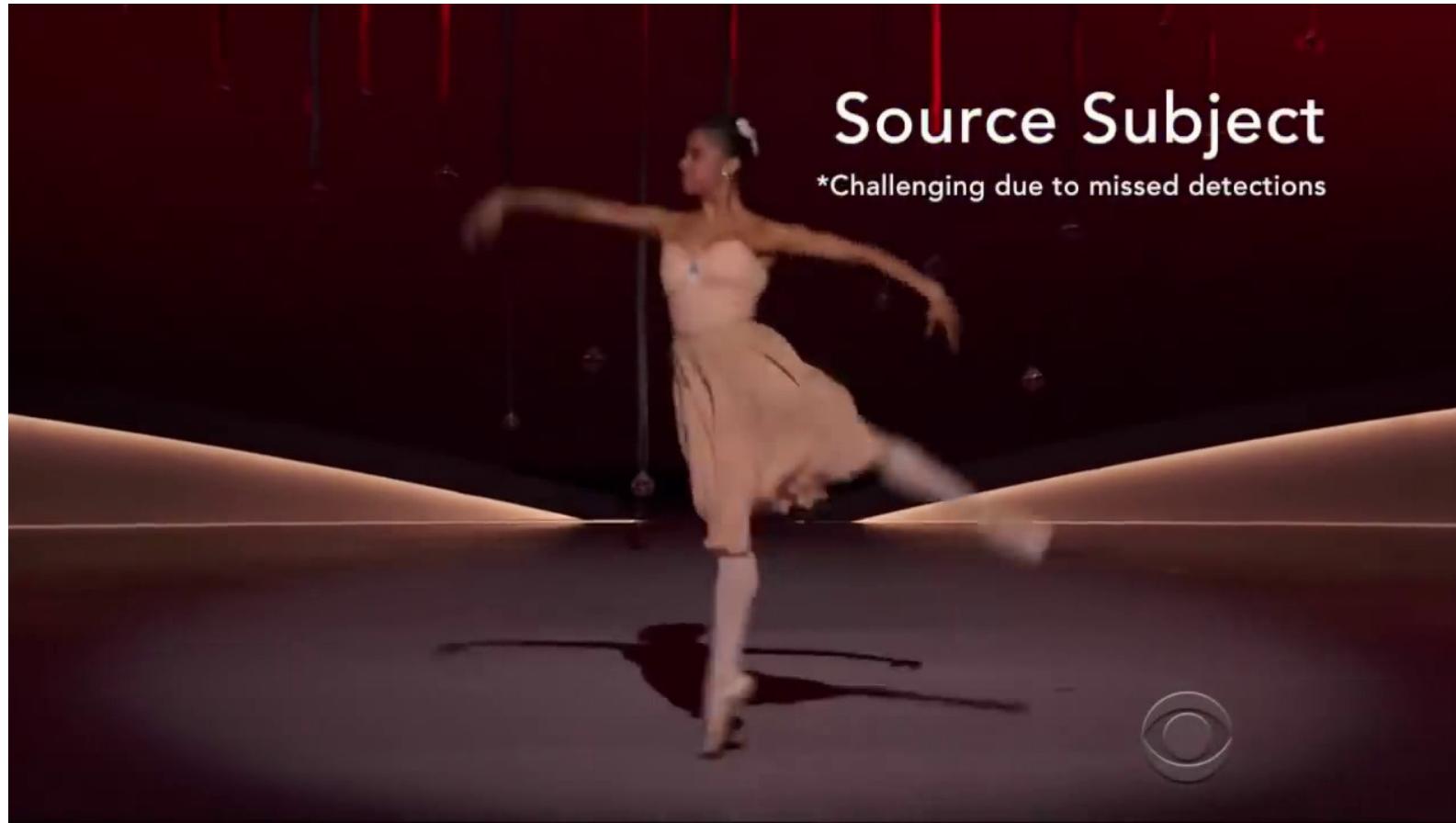
Generative Adversarial Networks – Applications

Transferring appearance to pose



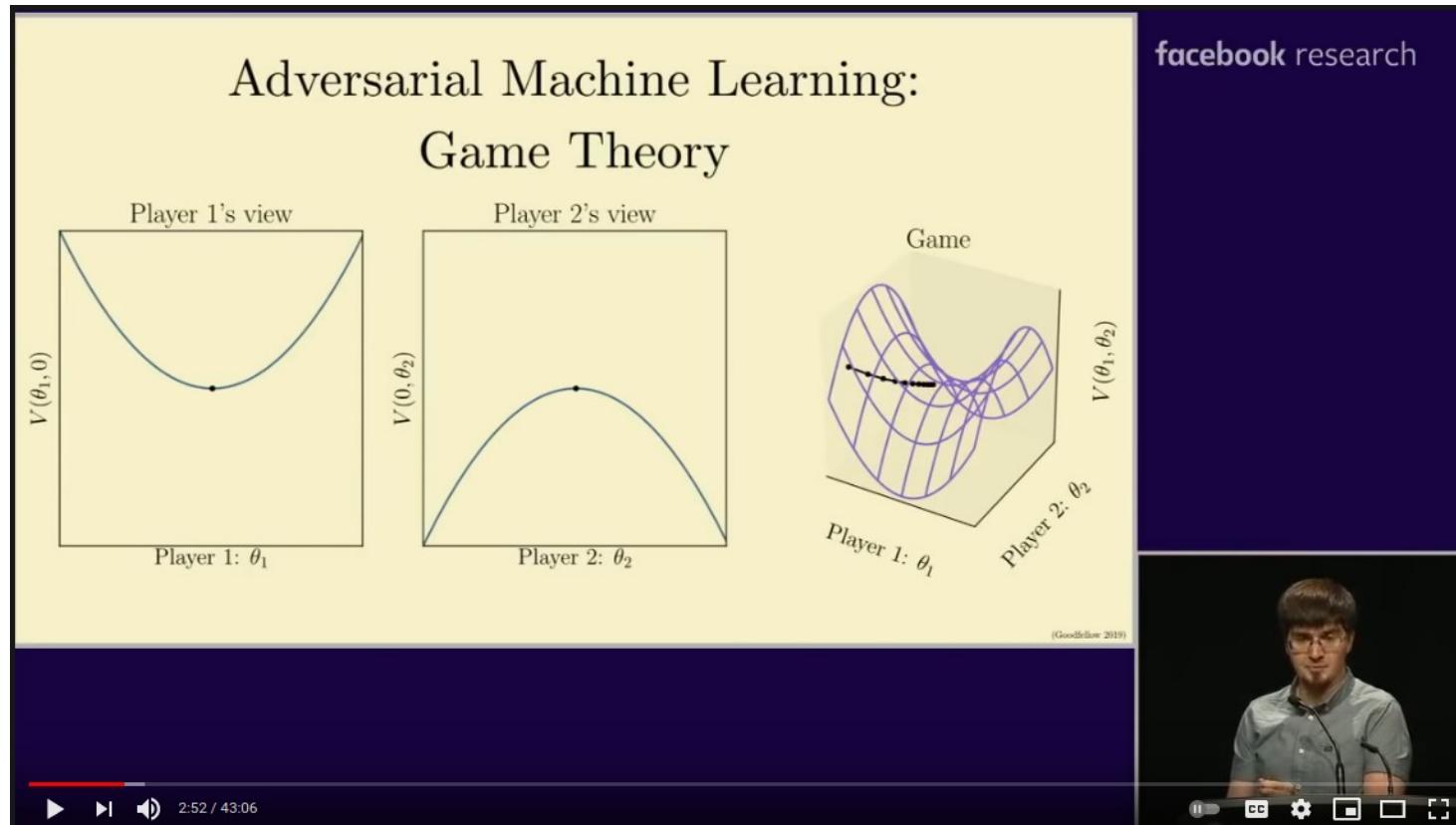
Generative Adversarial Networks – Applications

Transferring appearance to pose



Generative Adversarial Networks – Applications

For more: have a look at Ian Goodfellow's talk at ICLR'19



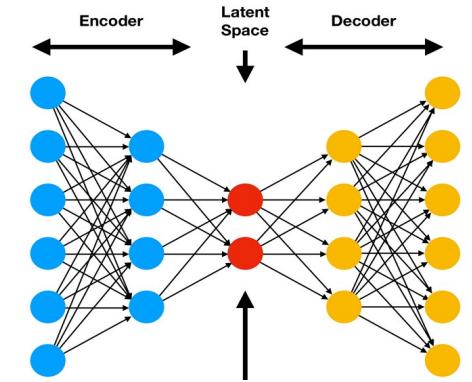
[Ian Goodfellow: Adversarial Machine Learning (ICLR 2019 invited talk)]

Summarizing

[Finally :D]

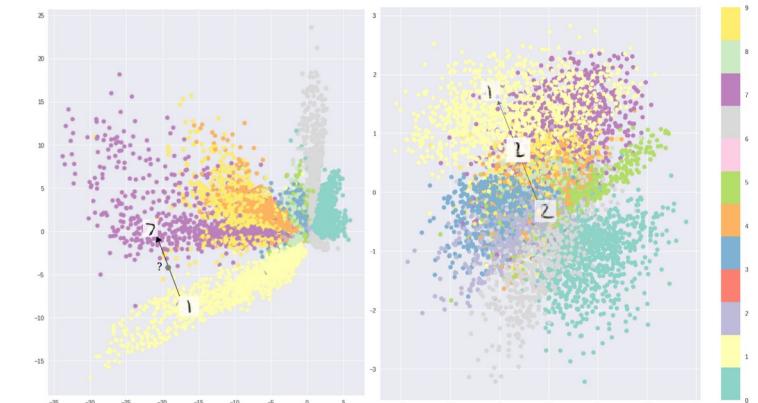
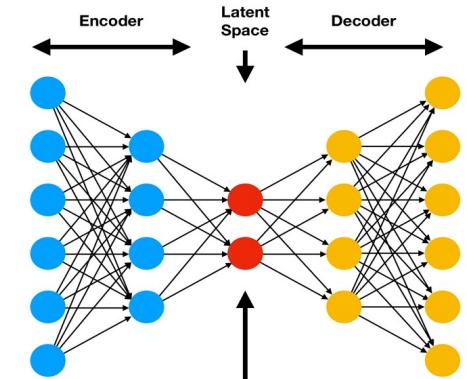
Summarizing

- Autoencoders map data points to compressed vectors
 - Does not require additional annotations
 - Learn how to preserve relevant properties from the data



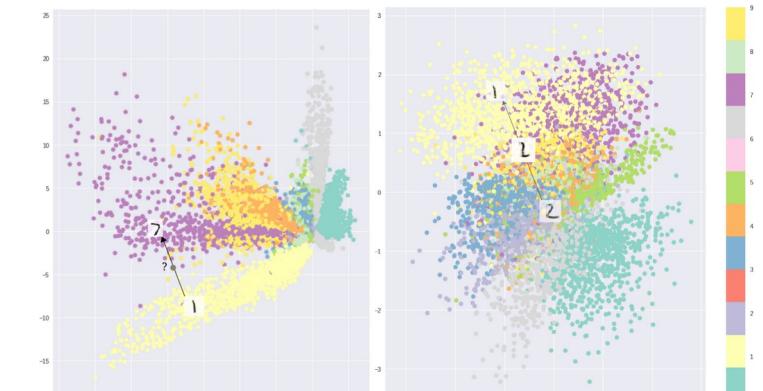
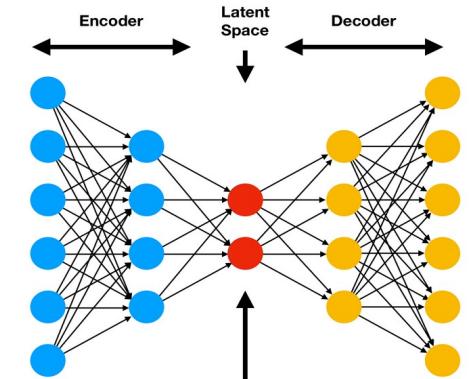
Summarizing

- Autoencoders map data points to compressed vectors
 - Does not require additional annotations
 - Learn how to preserve relevant properties from the data
- VAEs map data points to continuous spaces
 - Enables the generation of new examples
 - Enables for interpolation across data examples



Summarizing

- **Autoencoders map data points to compressed vectors**
 - Does not require additional annotations
 - Learn how to preserve relevant properties from the data
- **VAEs map data points to continuous spaces**
 - Enables the generation of new examples
 - Enables for interpolation across data examples
- **GANs: adversarial learning with DNNs**
 - Enables the generation of new examples
 - Training can become complex



Input



Output



Questions?

References

- Autoencoders - Deep Learning Book - Chapter 14.
<https://www.deeplearningbook.org/contents/autoencoders.html>
- Variational Autoencoders - Deep Learning Book - Chapter 20.10.3
https://www.deeplearningbook.org/contents/generative_models.html
- P. Vincent, H. Larochelle, Y. Bengio, P. Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders ICML 2008
- F. Baldassarre, D. González Morín, L. Rodés-Guirao. Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2. ArXiv:1712.03400
- I. Higgins, Loic Matthey, A. Pal, Christopher P. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, A. Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. ICLR 2017
- X Hou, L. Shen, K. Sun, G. Qiu. Deep Feature Consistent Variational Autoencoder. WACV 2017.

References

- I. Shafkat, Intuitively Understanding Variational Autoencoders
<https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf>
- A. Wong. Similar Image Retrieval using Autoencoders, 2017.
<https://towardsdatascience.com/find-similar-images-using-autoencoders-315f374029ea>
- I. J. Goodfellow*, J. Pouget-Abadie†, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio§. Generative Adversarial Nets, NeurIPS, 2014.
- J. Zhu, T. Park, P. Isola, A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. ICCV 2017. <https://junyanz.github.io/CycleGAN/>
- J. Brownlee. A Gentle Introduction to CycleGAN for Image Translation.
<https://machinelearningmastery.com/what-is-cyclegan>



Artificial Neural Networks

[2500WETANN]

José Oramas