State of Art in AI

Generative Deep Learning

Rajagopal.motivate@gmail.com
https://sites.google.com/view/aiforeveryone/

Generative Deep Learning

Networks that generate

The future of AI

1. Deep Learning

- “Biggest leap since invention of computers”
- Output of AI: Scalar values (numbers)

Output = 2

2. Generative Deep Learning

- GAN “the most interesting idea in the last 10 years in ML.”
- Output of AI: Vectors , Images

Output =

3. AI that creates another AI

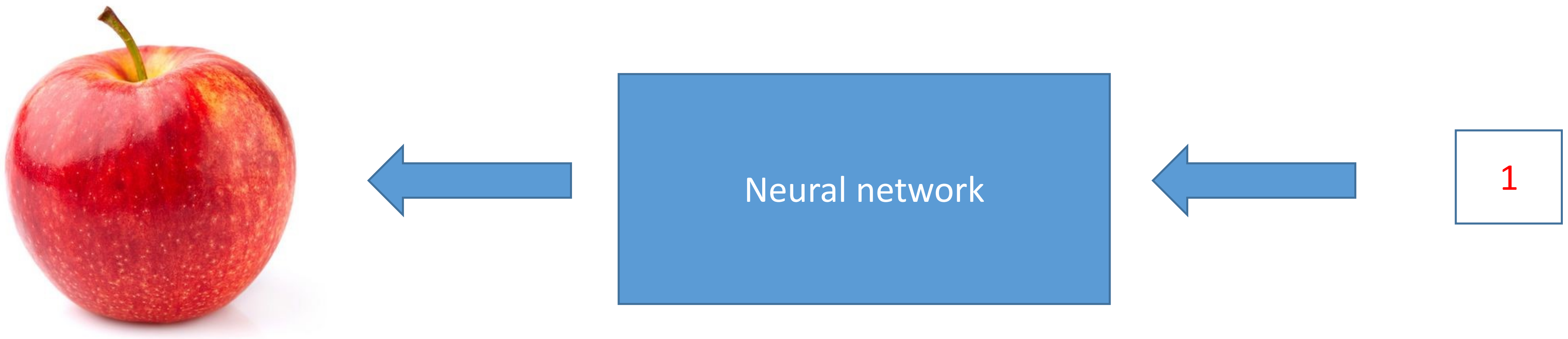
- Progressive, AutoML
- Output of AI: Neural netwoks

Output =

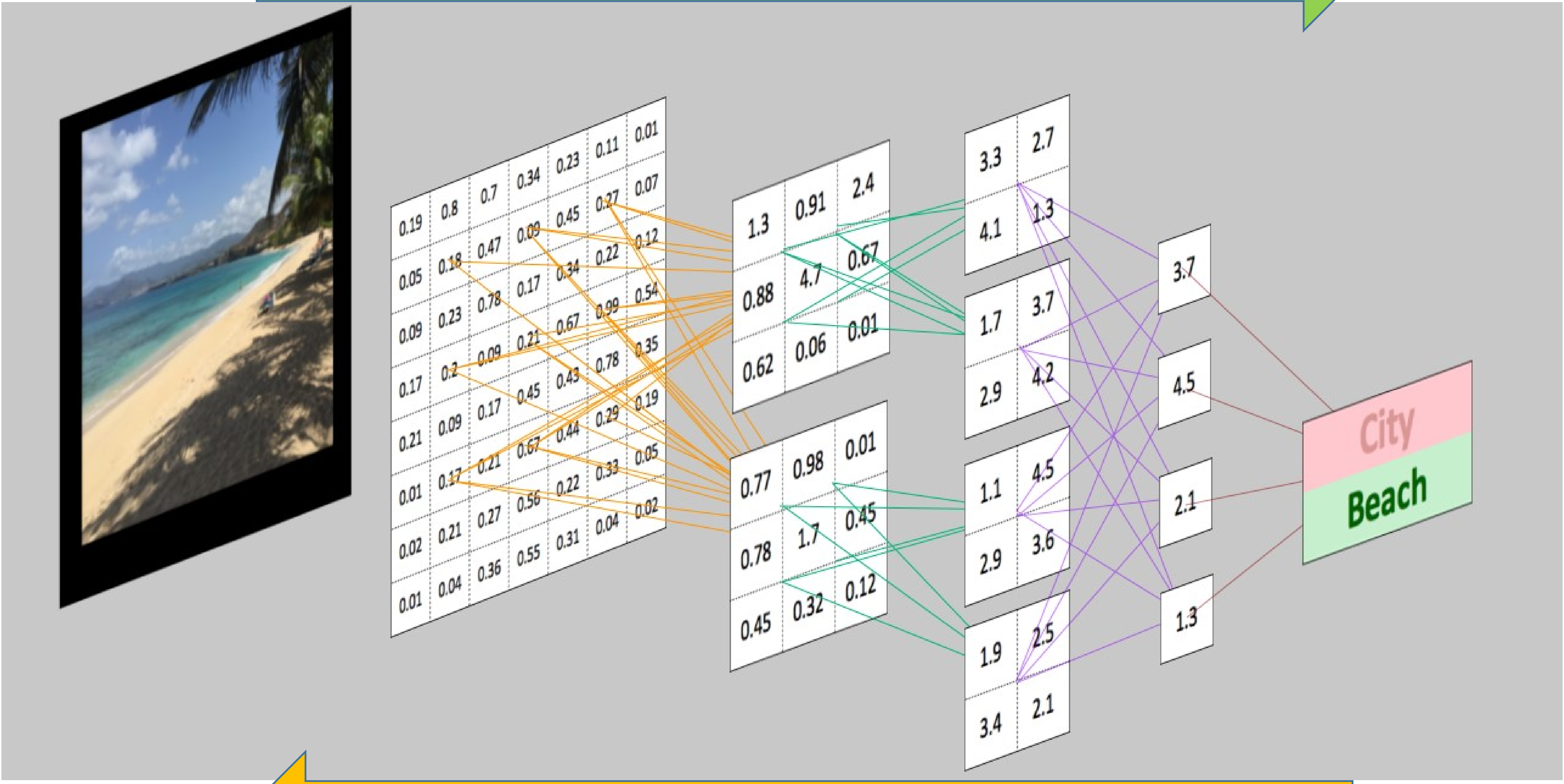
A neural network learns the patterns about apple and oranges

Acknowledgments & Credits are mentioned to inspirational resources presented last slides. For more like this, https://sites.google.com/view/AIforEveryone

So how can we use the learning made by a neural network?



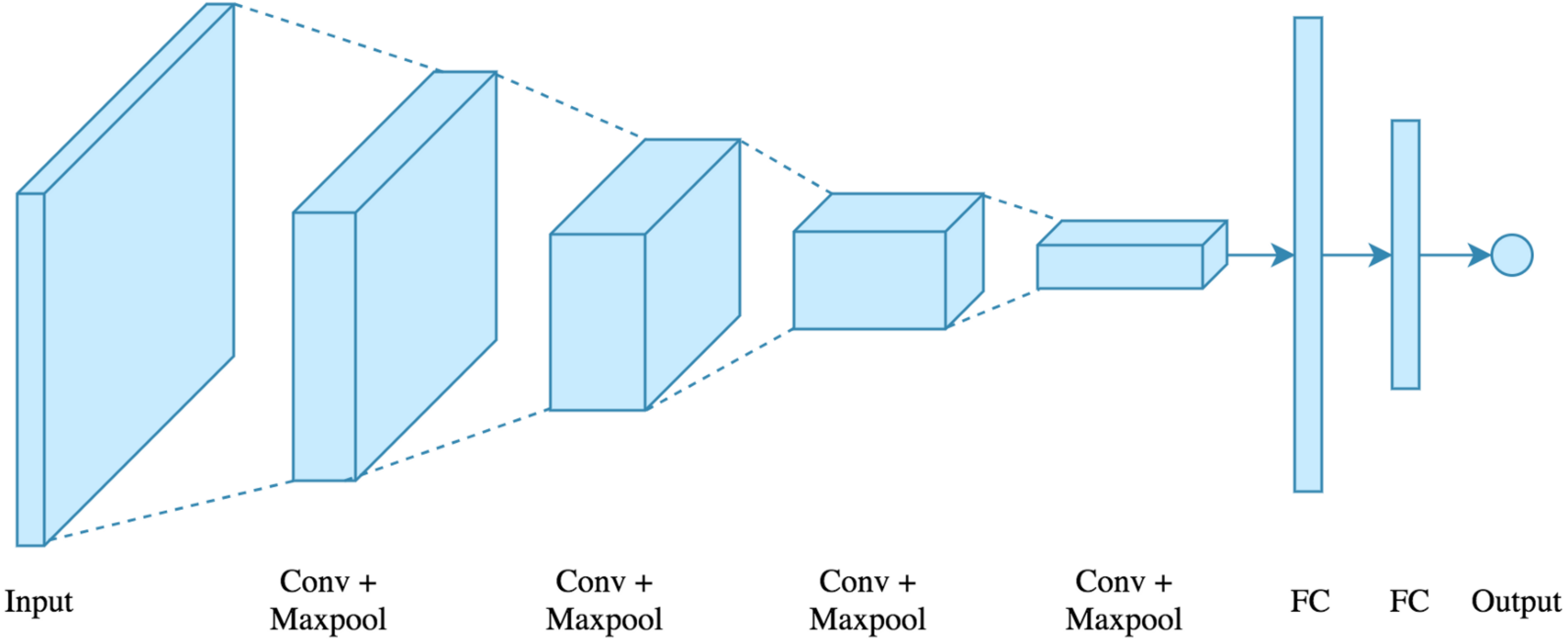
Each layer is a way to represent the image



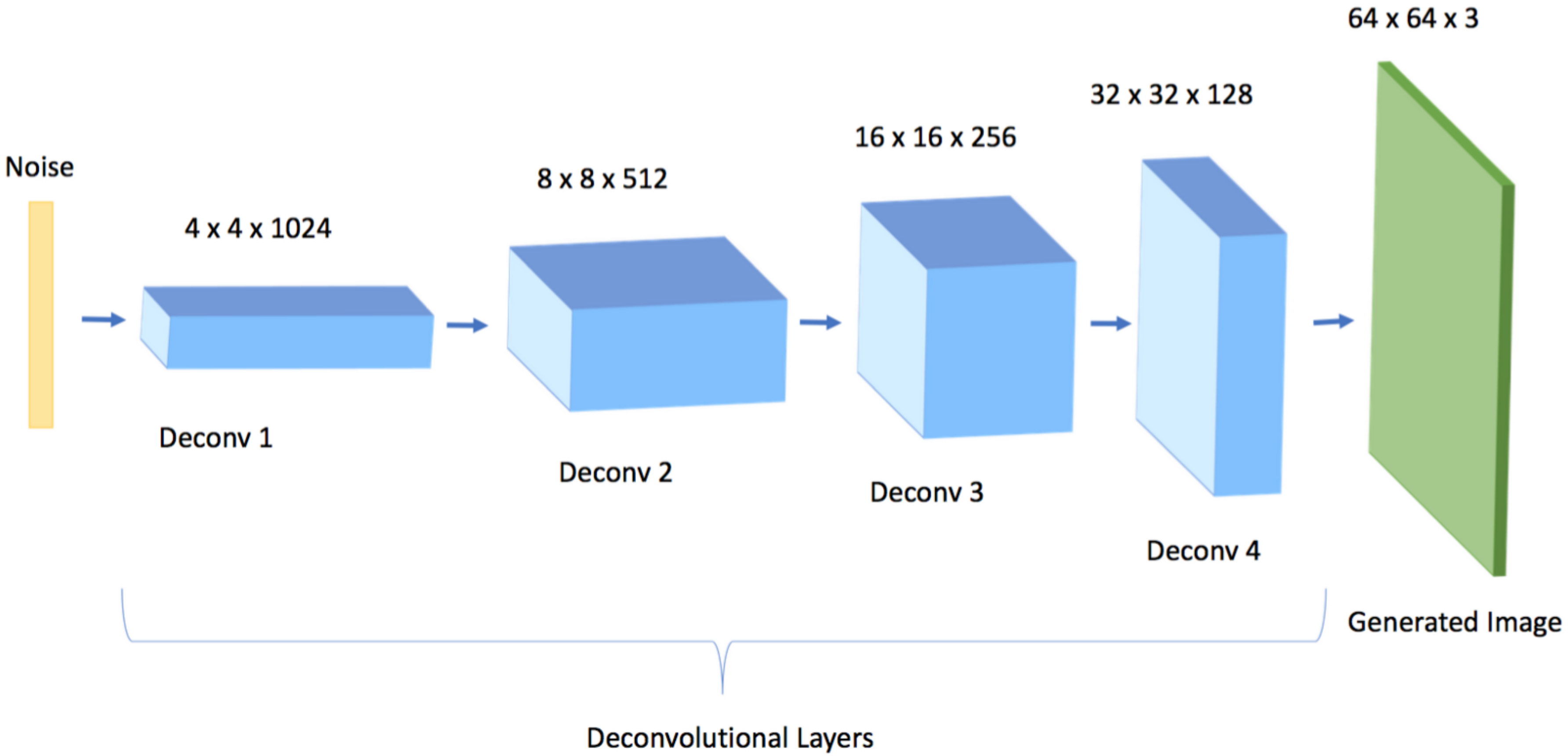
Is it possible to represent an integer as array ?

Acknowledgments & Credits are mentioned to inspirational resources presented last slides. For more like this, https://sites.google.com/view/AIforEveryone

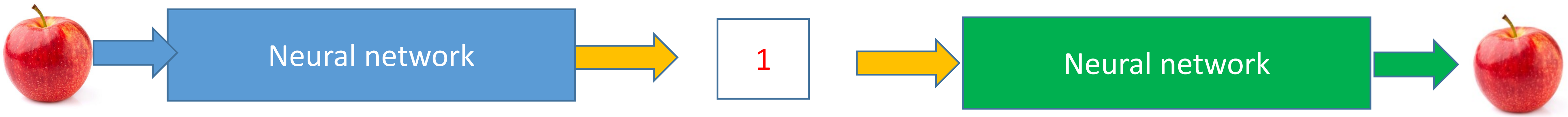
A image can be transformed to a integer
(representation learning)

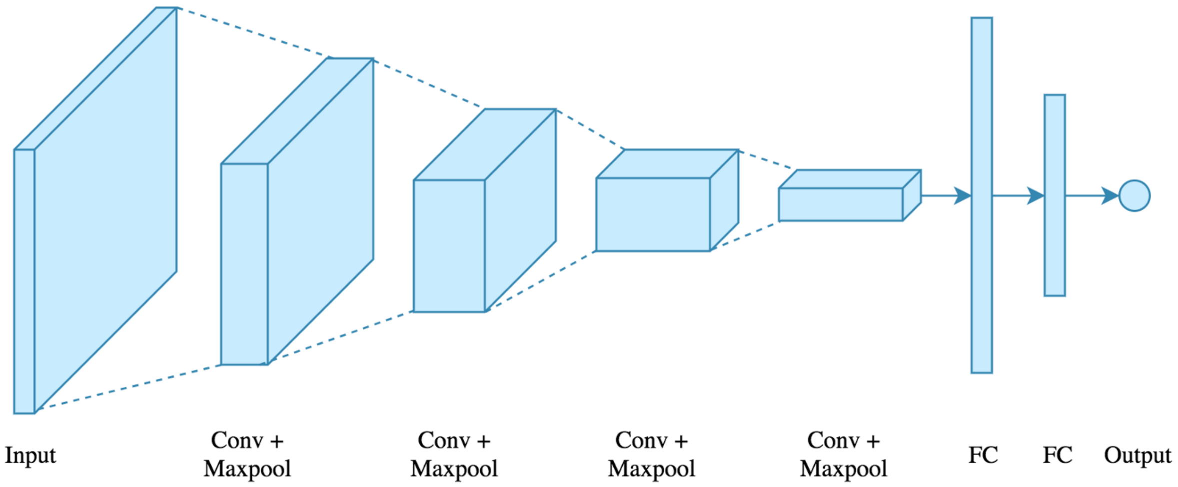


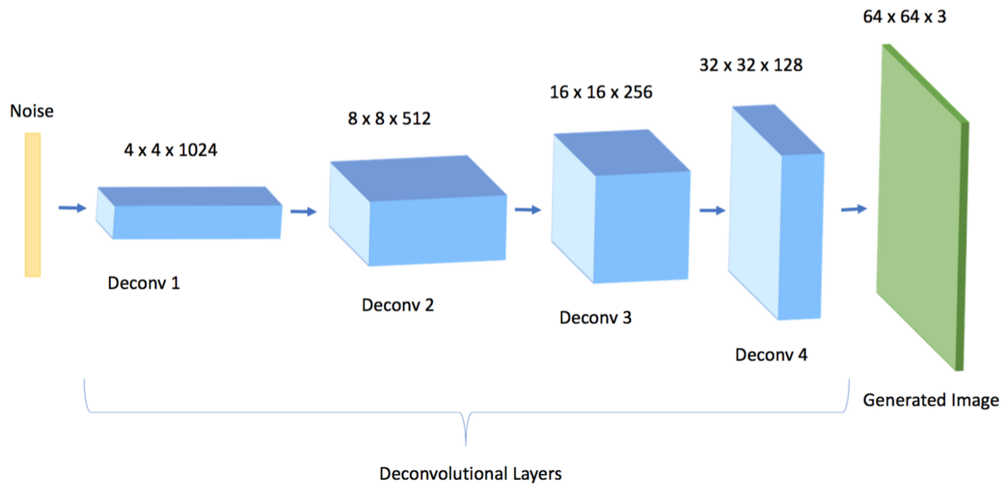
A integer can be transformed to a image (representation learning)

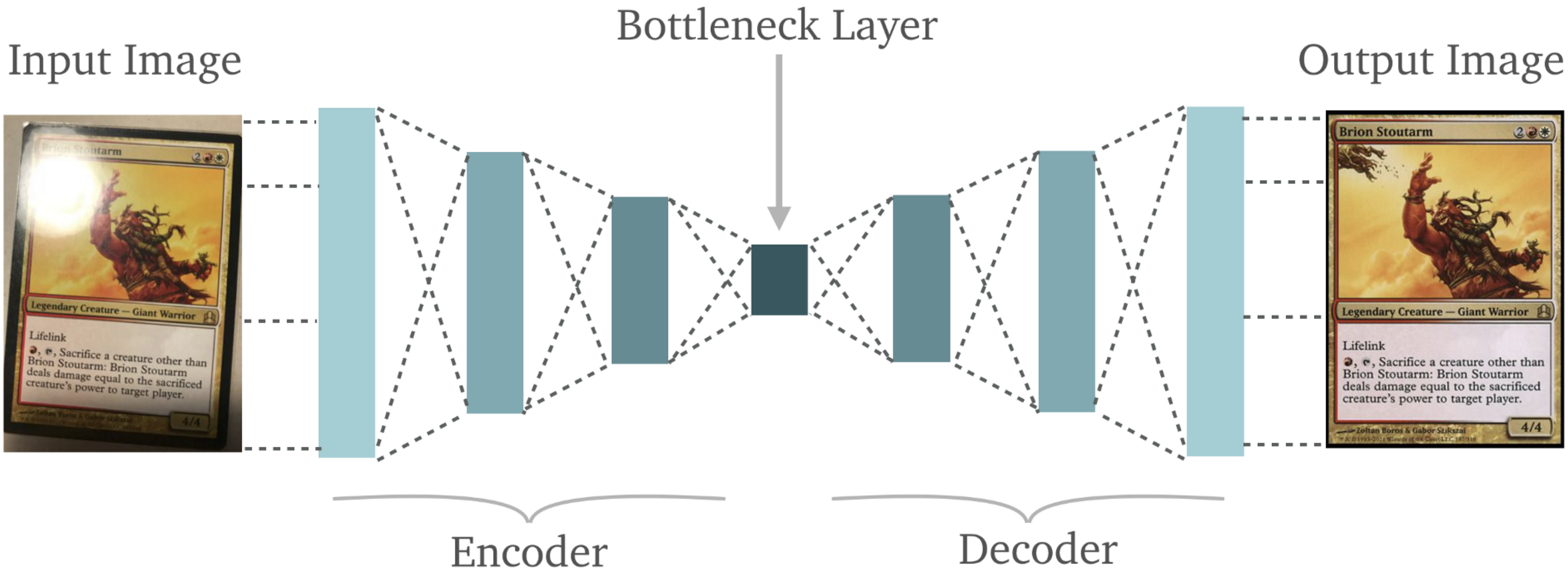


A image can be transformed to a integer
(representation learning)



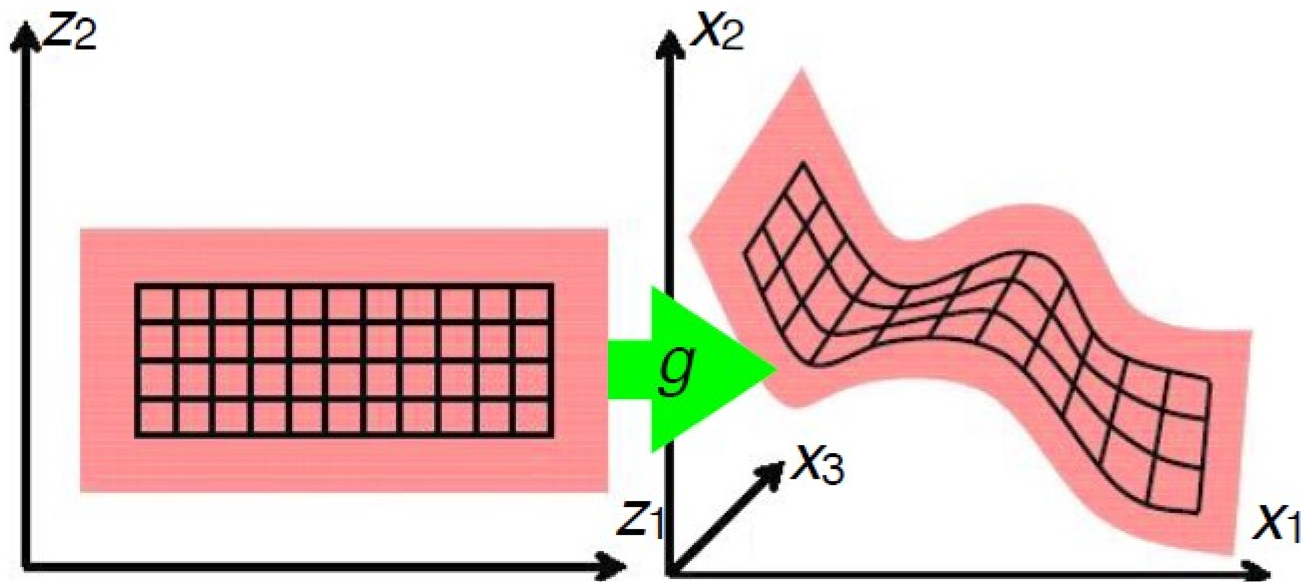


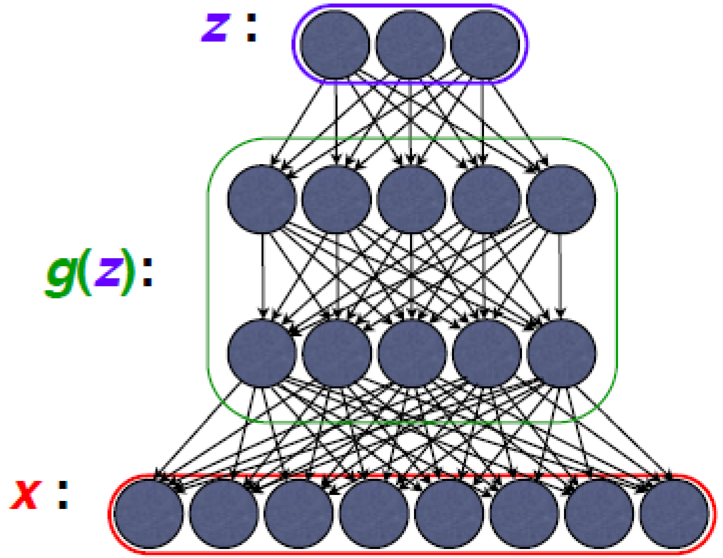


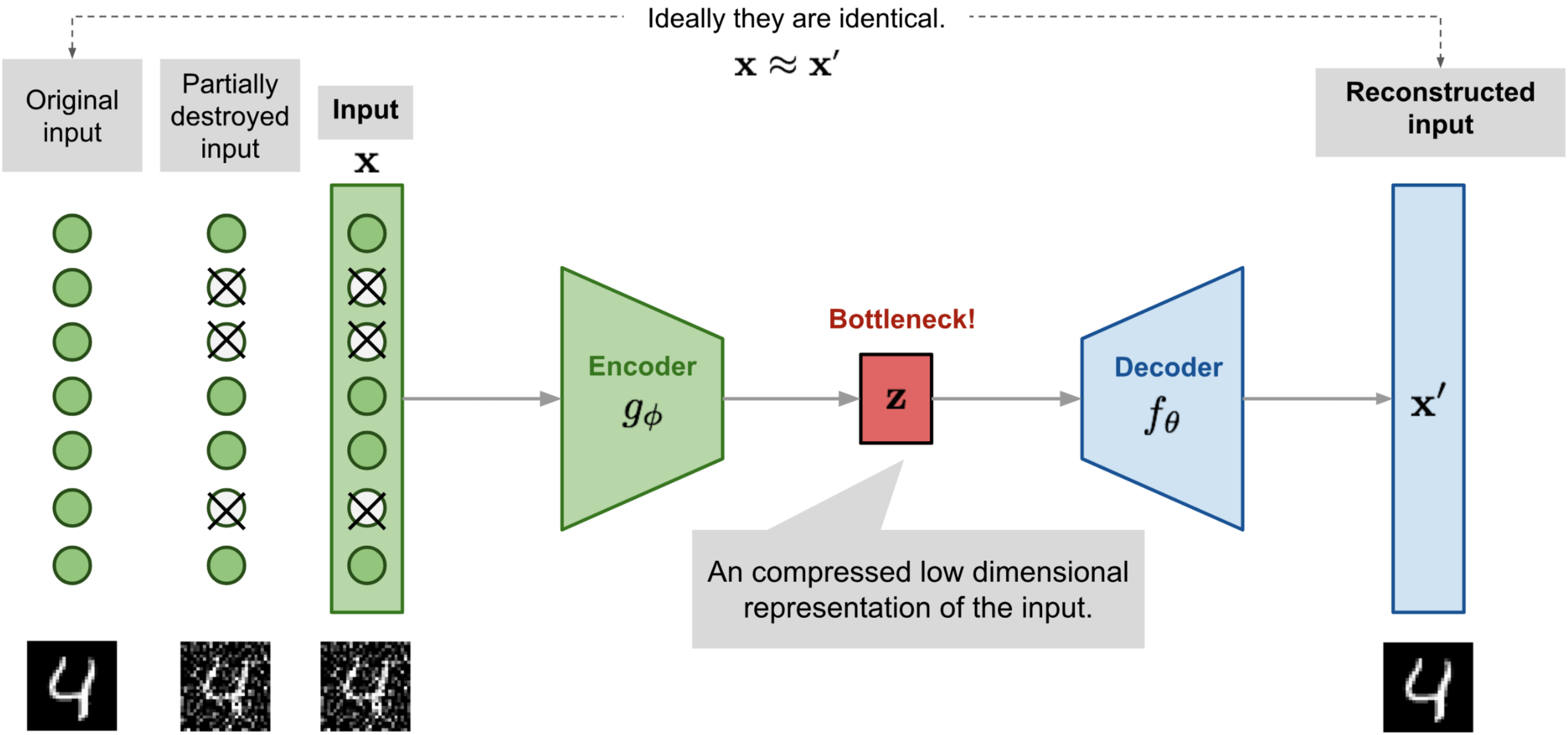

Bottleneck Layer
Input Image
Output Image
Brion Stoutarm
Legendary Creature — Giant Warrior
Lifelink
Sacrifice a creature other than Brion Stoutarm: Brion Stoutarm deals damage equal to the sacrificed creature's power to target player.
4/4
Encoder
Decoder

Latent variable models

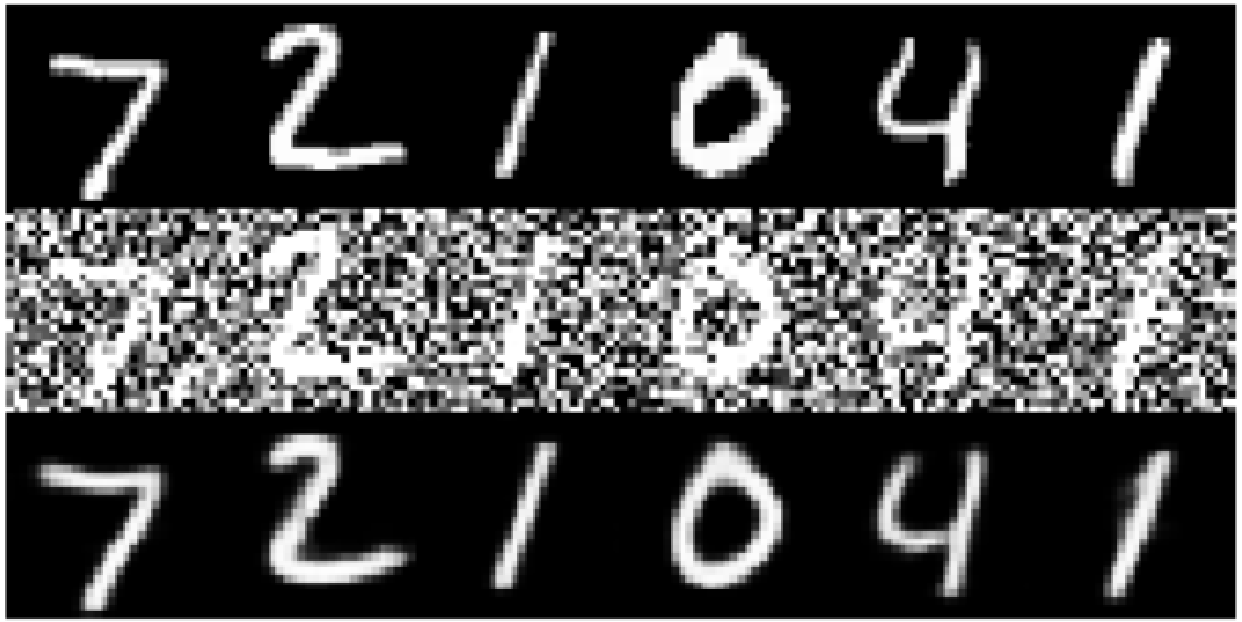
- Remember representation learning
- Transform by moving in latent space







Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Original input
Partially destroyed input
Input
$\mathbf{x}$
Reconstructed input
Encoder
g_ϕ
Bottleneck!
$\mathbf{z}$
Decoder
f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.

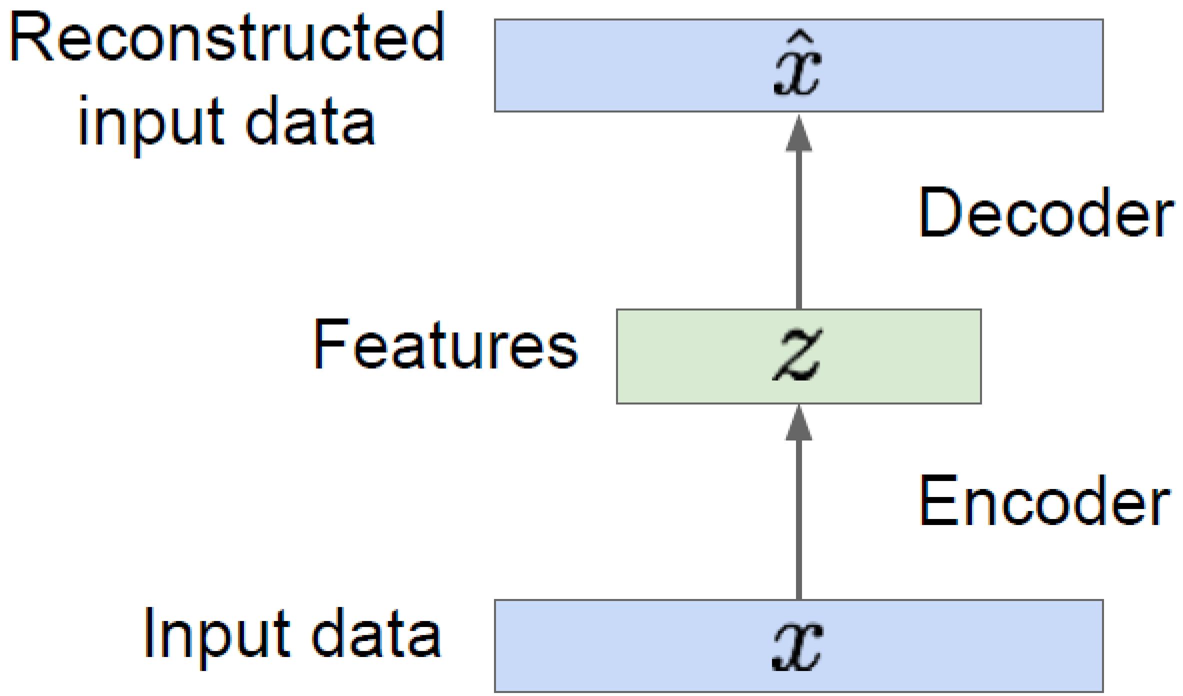


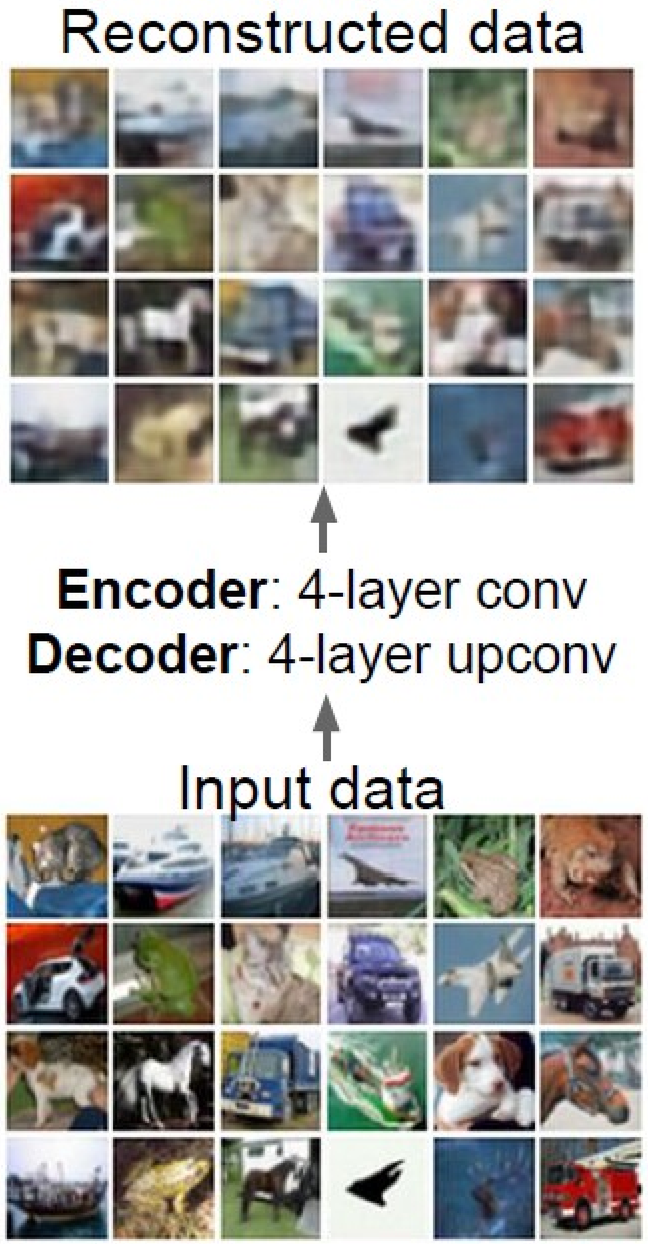
Auto encoders

How to learn this feature representation?

Train such that features can be used to reconstruct original da

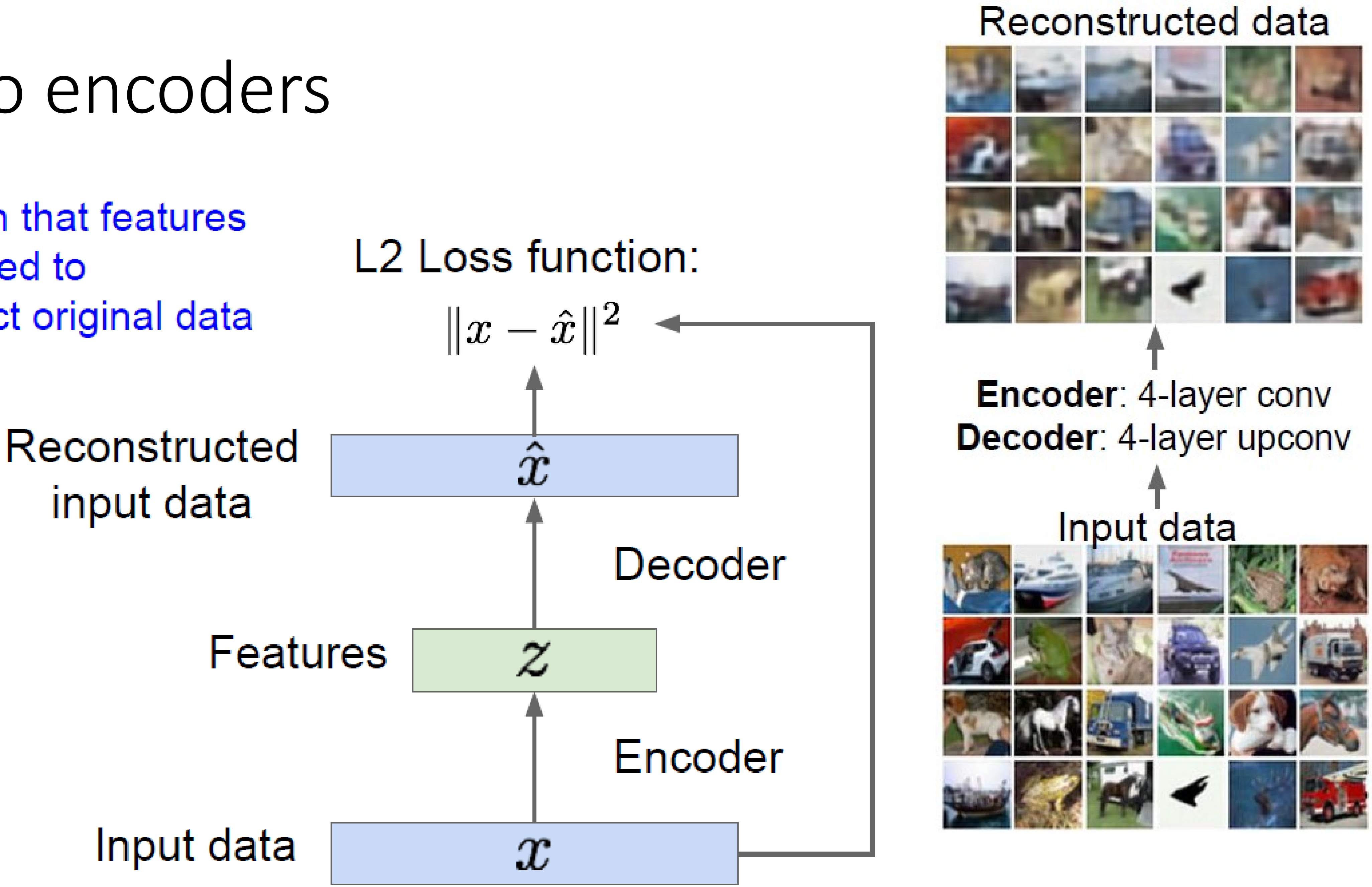
“Autoencoding” - encoding itself





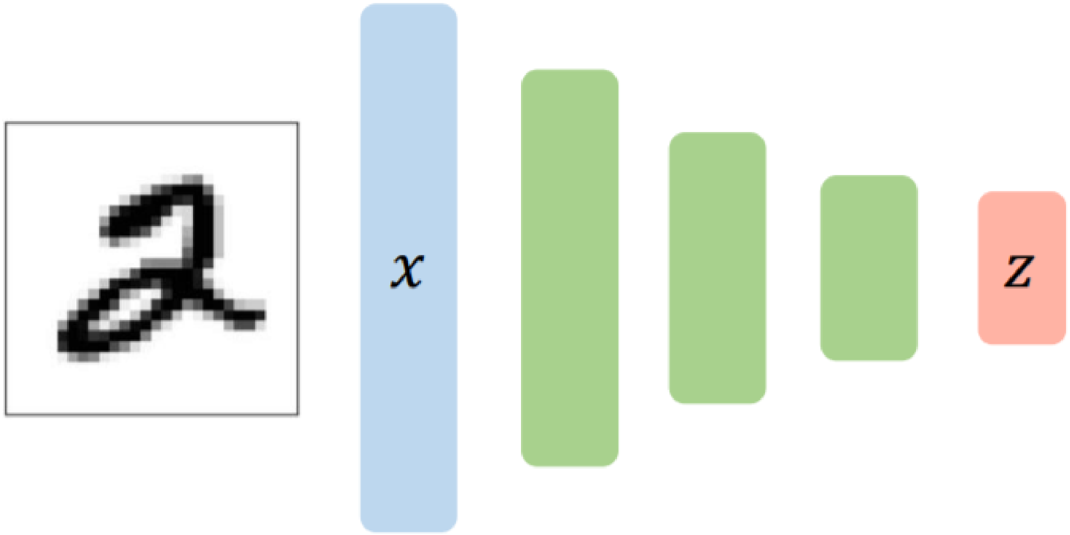
Auto encoders

Train such that features can be used to reconstruct original data



Autoencoders: background

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

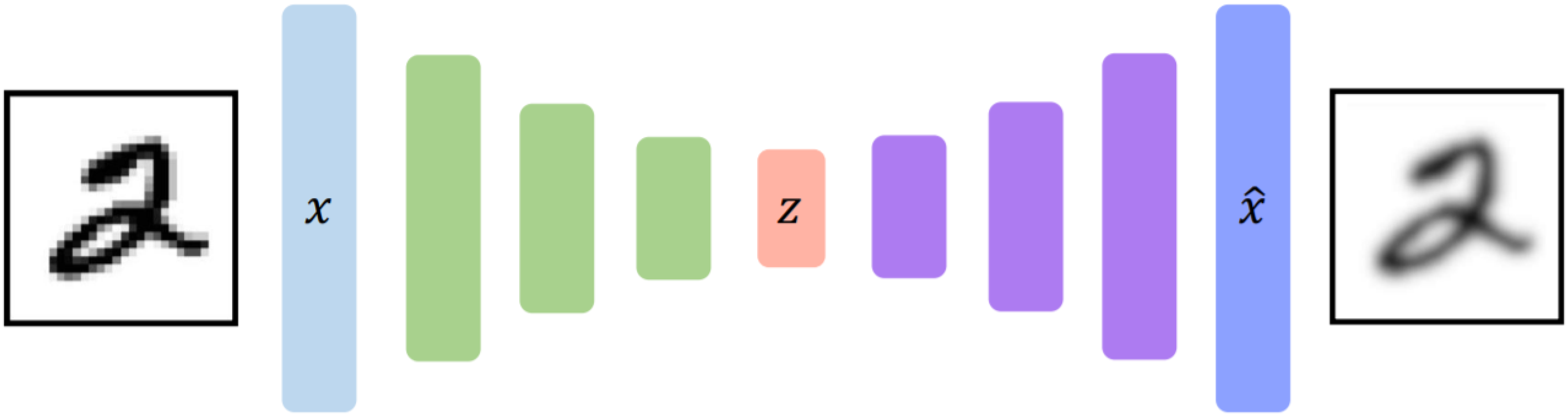


Why do we care about a low-dimensional z?

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

Autoencoders: background

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



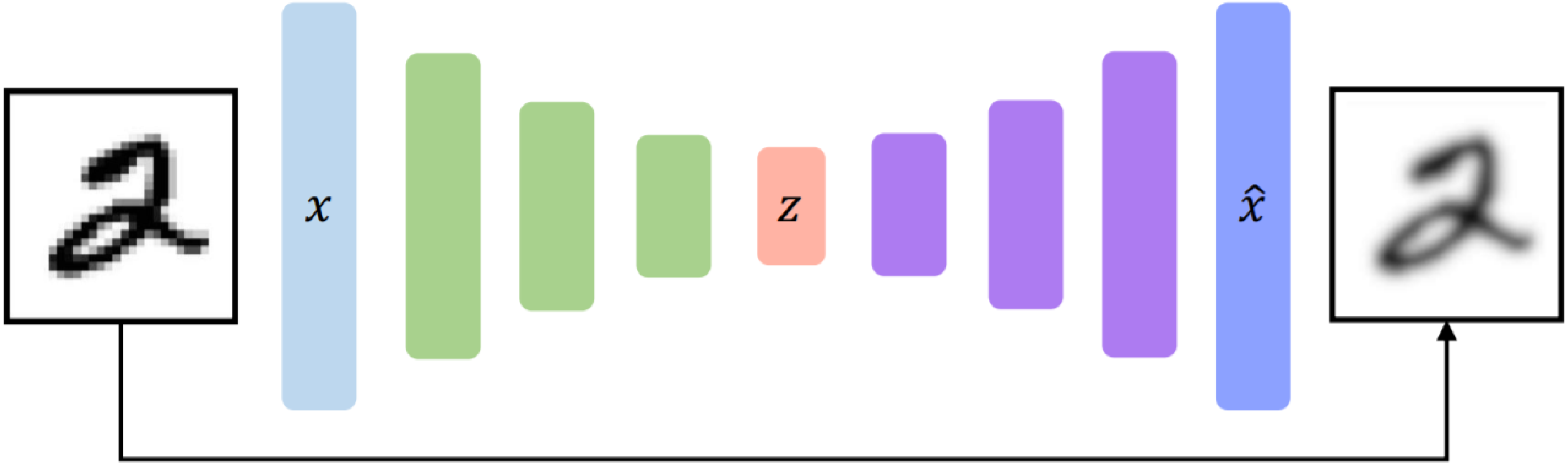
"Decoder" learns mapping back from latent, z, to a reconstructed observation, $\hat{x}$

Acknowledgments & Credits are mentioned to inspirational resources presented last slides. For more like this, https://sites.google.com/view/AIforEveryone

Autoencoders: background

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!!

Bottleneck hidden layer forces network to learn a compressed latent representation

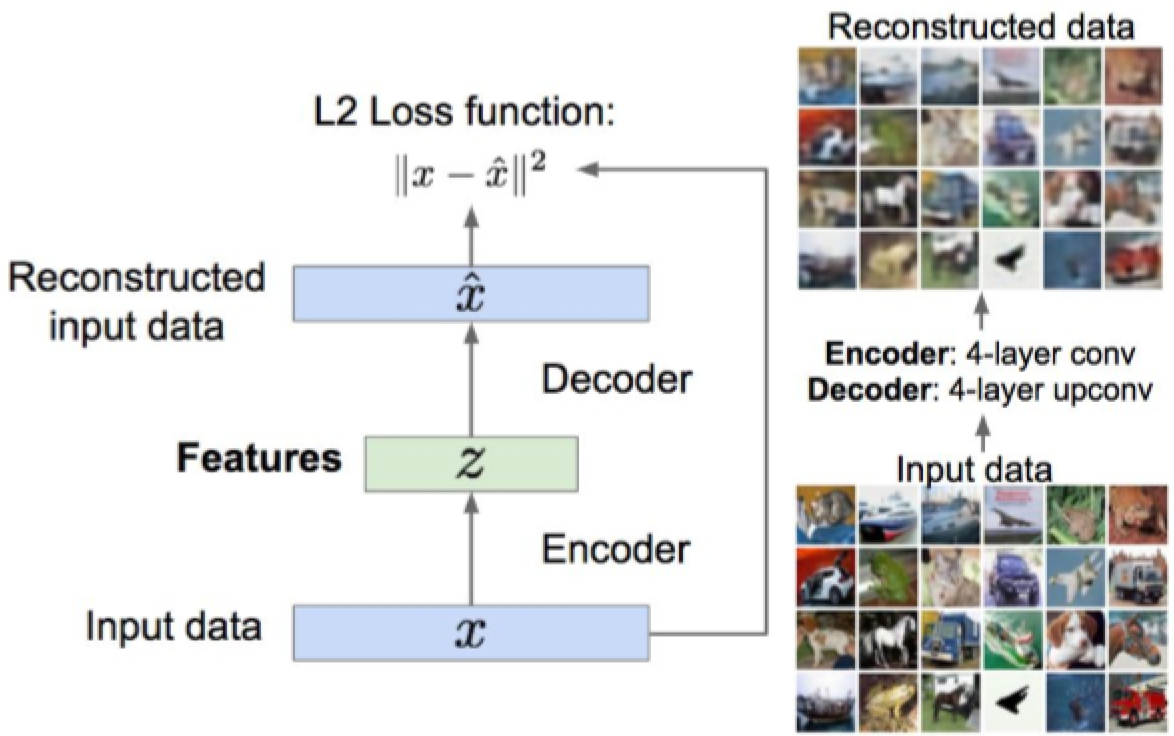
Reconstruction loss forces the latent representation to capture (or encode) as much "information" about the data as possible

Unsupervised Learning

Data: x
Just data, no labels!

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

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



Autoencoders
(Feature learning)

Generative Adversarial Networks (GANs)

Supervised vs Unsupervised Learning

Supervised Learning

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

Goal: Learn a *function* to map x -> y

Unsupervised Learning

Training data is cheap

Data: x
Just data, no labels!

Holy grail: Solve unsupervised learning => understand structure of visual world

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

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!

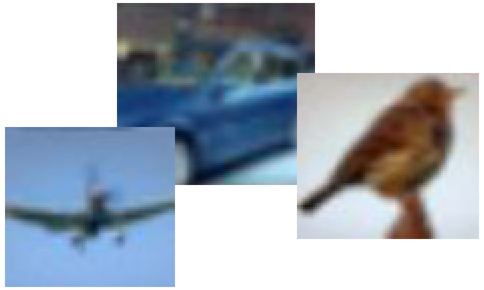
Goal: Learn some *hidden* or *underlying structure* of the data

Examples: Clustering, feature or dimensionality reduction, etc.

Acknowledgments & Credits are mentioned to inspirational resources presented last slides. For more like this, https://sites.google.com/view/AIforEveryone

Generative Models

Given training data, generate new samples from same distribution



Training data ~ $p_{data}(x)$

Generated samples ~ $p_{model}(x)$

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

Sample Generation

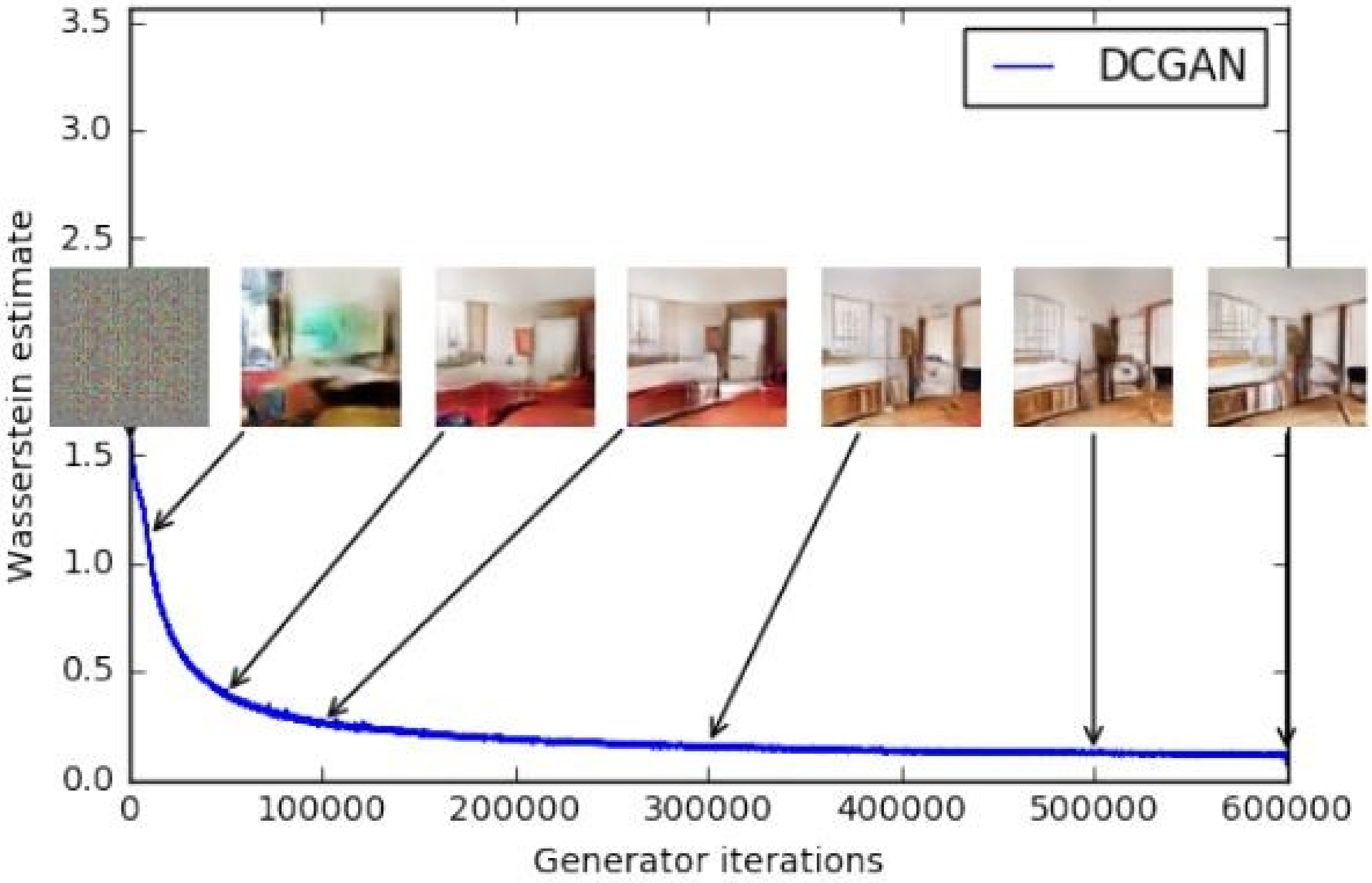
Input samples

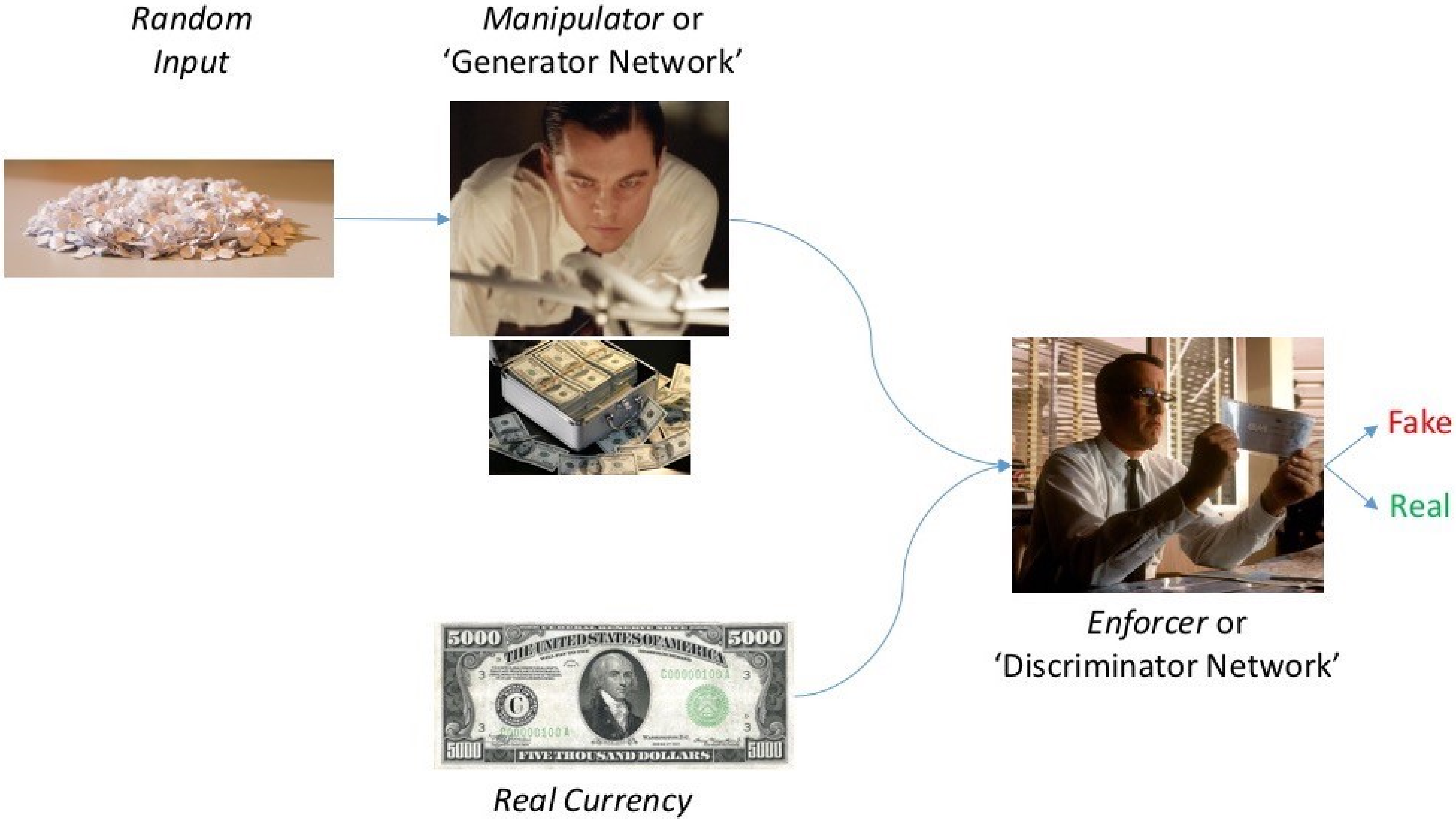
Training data $\sim P_{data}(x)$

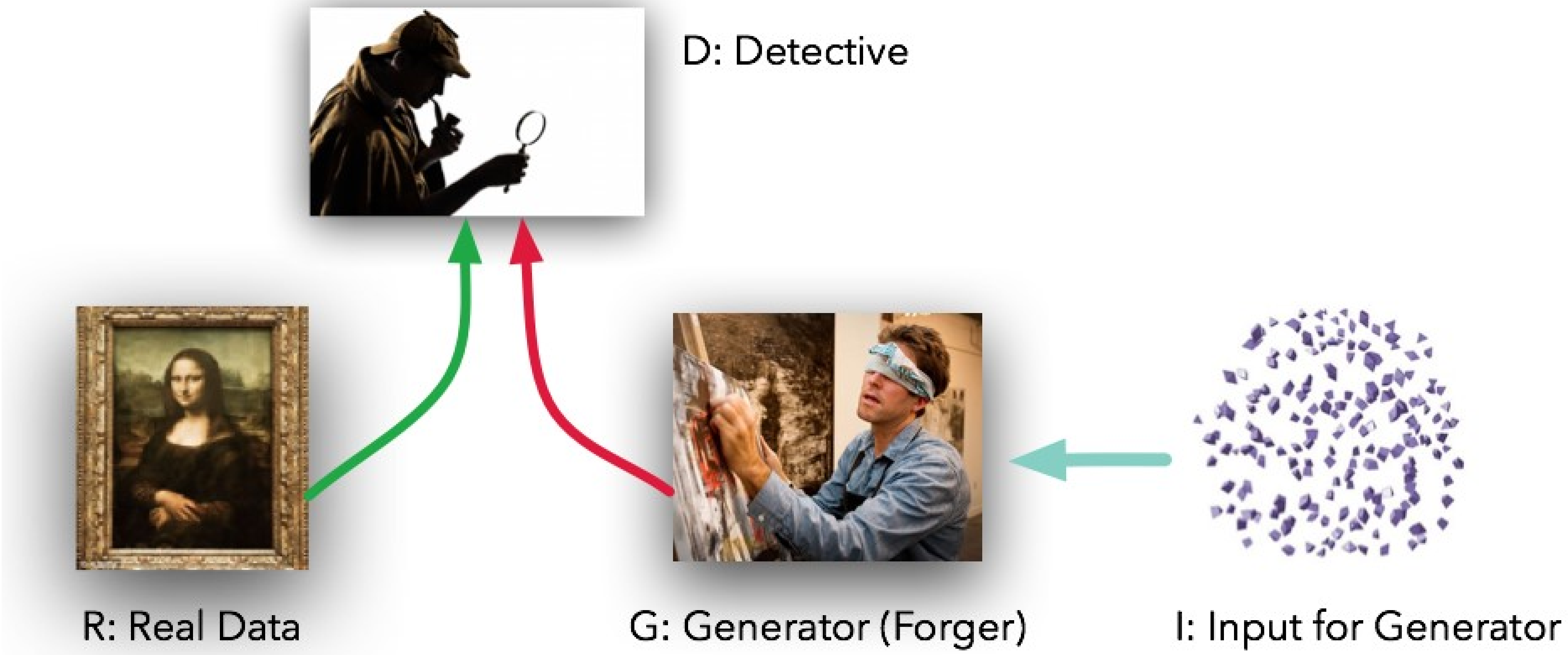
Generated samples

Generated $\sim P_{model}(x)$

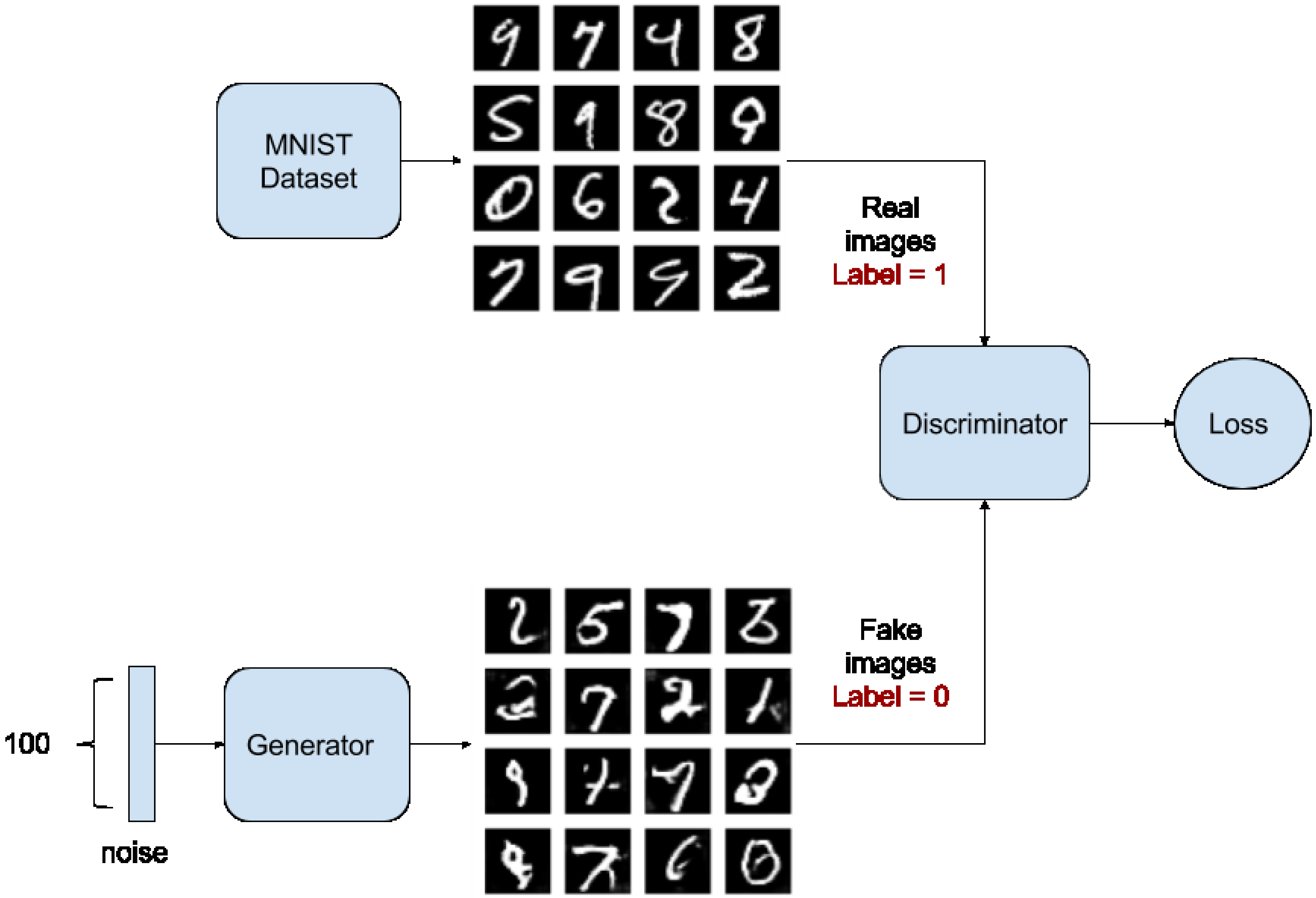
How can we learn $P_{model}(x)$ similar to $P_{data}(x)$?


DCGAN
Wasserstein estimate
3.5
3.0
2.5
1.5
1.0
0.5
0.0
0
100000
200000
300000
400000
500000
600000
Generator iterations


Random Input
Manipulator or 'Generator Network'
Enforcer or 'Discriminator Network'
Fake
Real
Real Currency



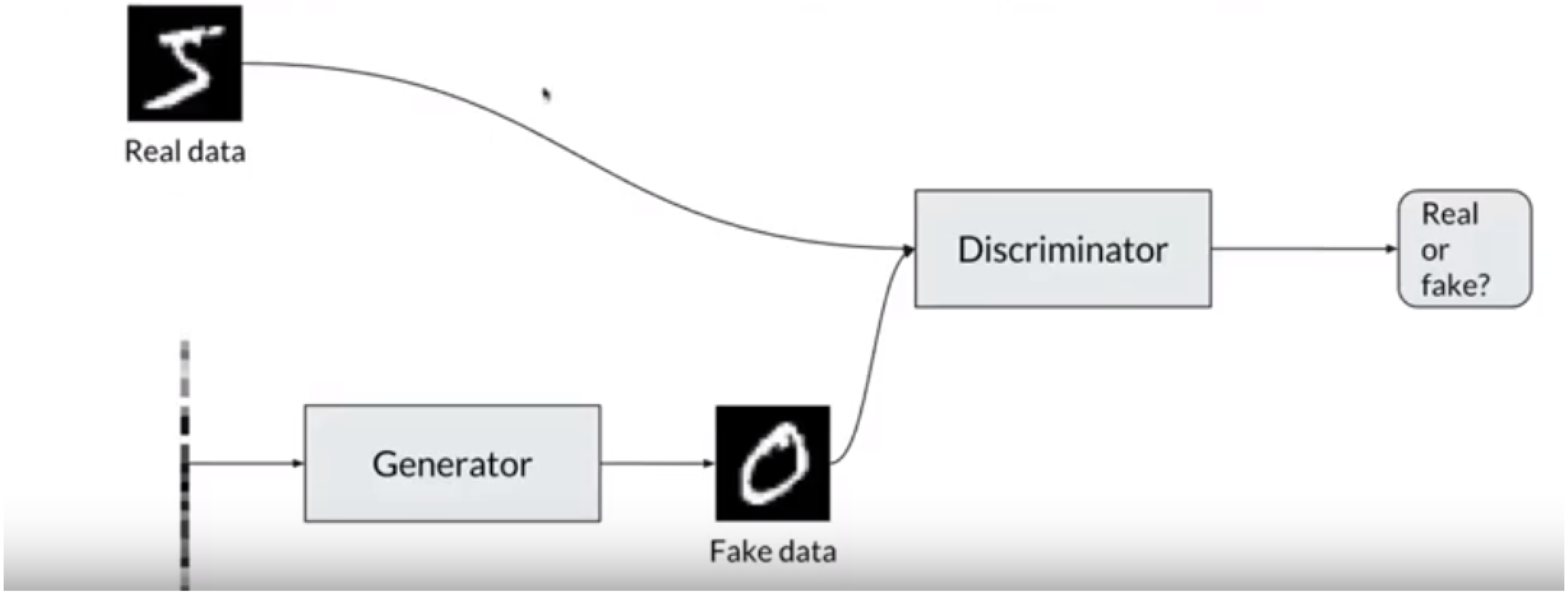
Acknowledgments & Credits are mentioned to inspirational resources presented last slides. For more like this, https://sites.google.com/view/AIforEveryone


MNIST
Dataset
Real
images
Label = 1
Discriminator
Loss
Fake
images
Label = 0
100
Generator
noise

Generative modeling

Goal: Take as input training samples from some distribution and learn a model that represents that distribution

Generative Adversarial Networks (GANs)



Generative Adversarial Networks

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

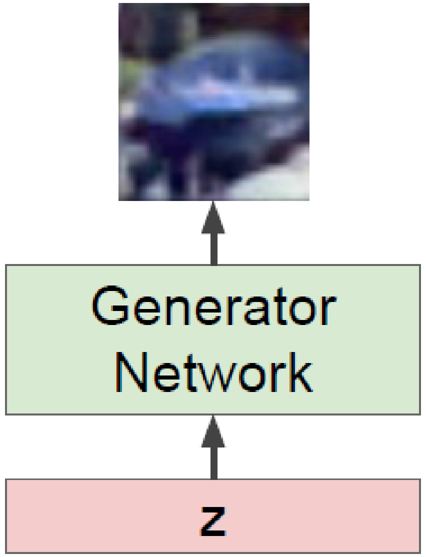
Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?

A: A neural network!

Output: Sample from training distribution

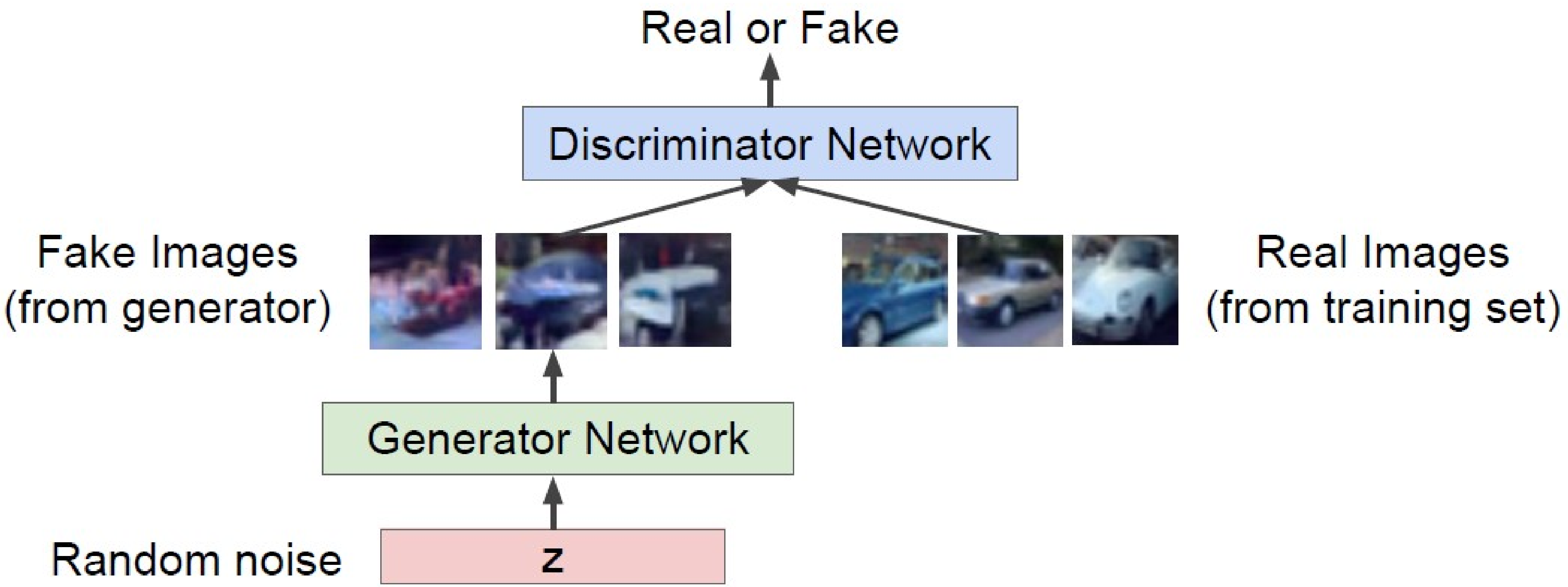


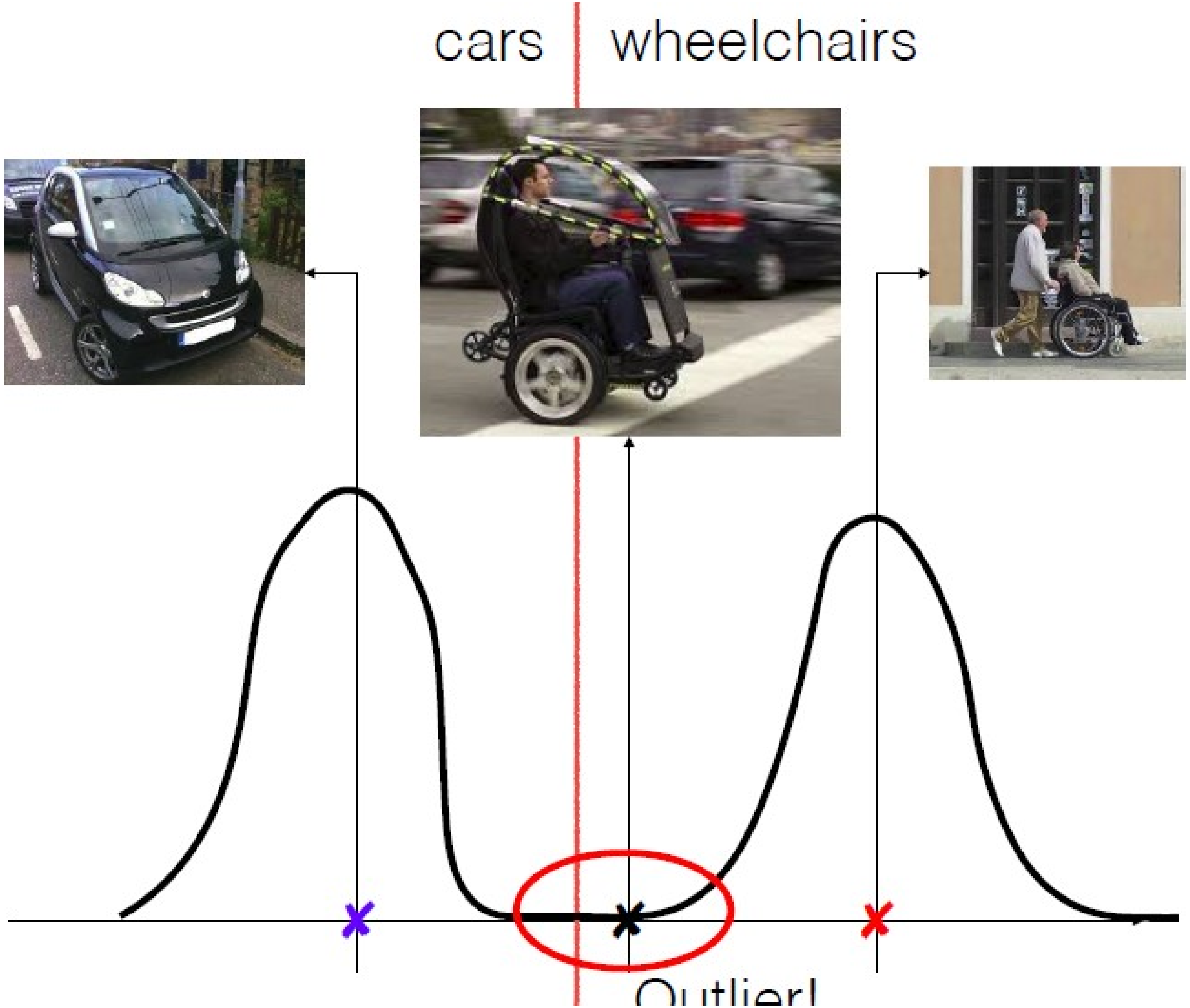
Input: Random noise

Training GANs: Two-player game

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

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

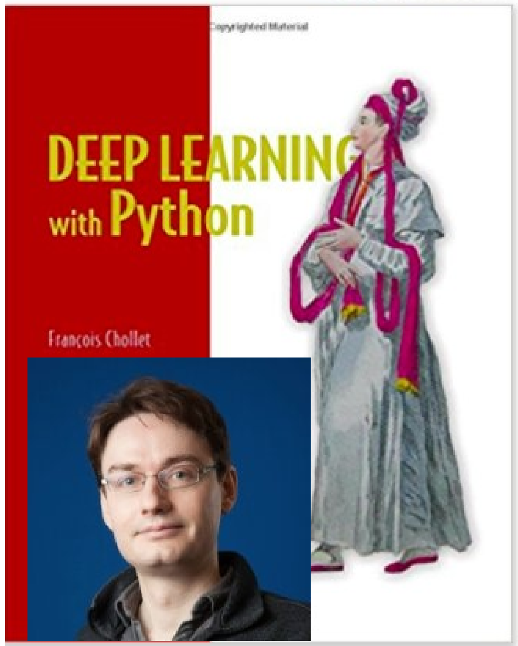




Acknowledgments & Credits are mentioned to inspirational resources presented last slides. For more like this, https://sites.google.com/view/AIforEveryone

Our inspiration

Friendly approaches :



1) **KERAS.io**

   François Chollet's
   Book on "Deep Learning with Python"

2) **Deeplearning.ai (Coursera.org)**

   Andrew Ng

3) **Fast.ai**

   Jeremy

Acknowledgments & Credits are mentioned to inspirational resources presented last slides. For more like this, https://sites.google.com/view/AIforEveryone

More inspiration

Excellent Resources

- **Stanford cs231 n**

http://cs231n.stanford.edu

- **MIT Deep Learning**

http://introtodeeplearning.com/
https://deeplearning.mit.edu

- **IIT Madras**

  my classes notes with Prof. Anurag (**Deep Learning)**