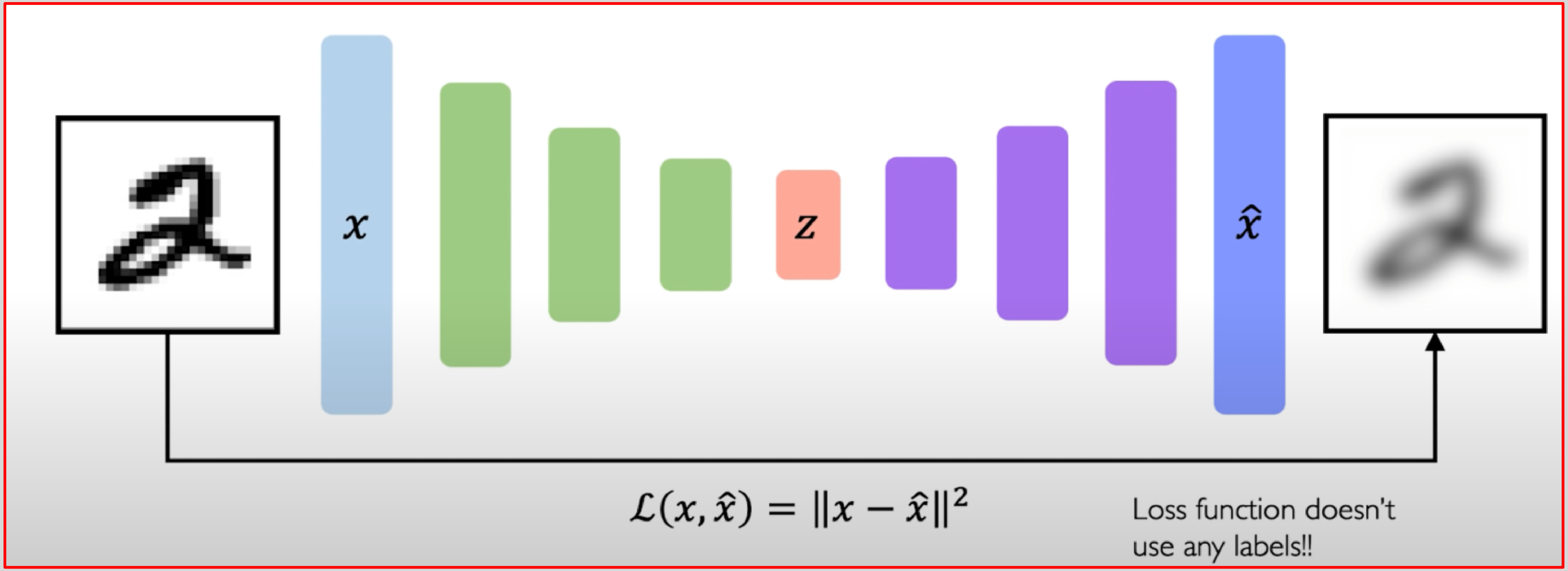
MODULE 10: INTRODUCTION TO GAN

BA713 - Machine Learning & AI

CONTENTS

DEEP AUTOENCODERS: BACKGROUND

- Train the model to learn features and reconstruct original data
- Autoencoding→ **Auto**matically **encoding** data

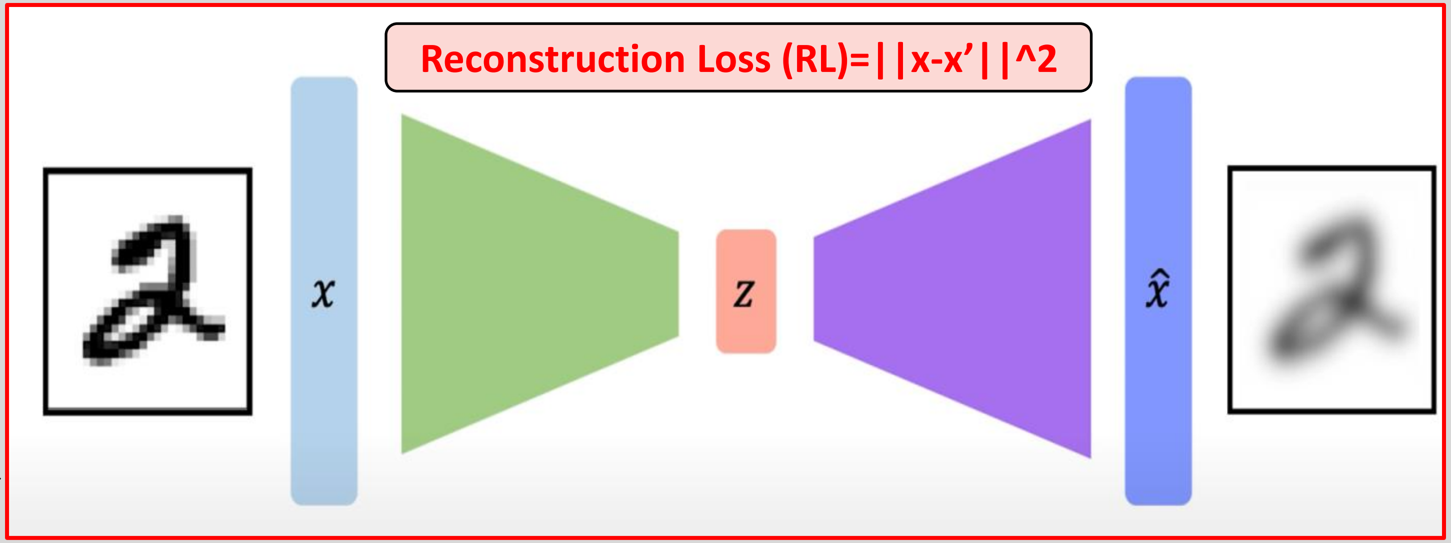


Source: introtodeeplearning.com

DAE VERSUS VAE

Accepts input, compresses it and recreates the original input

DAE



Assumes source data has underlying probability distribution (Gaussian). Generates new data from lower dimensional latent space

VAE

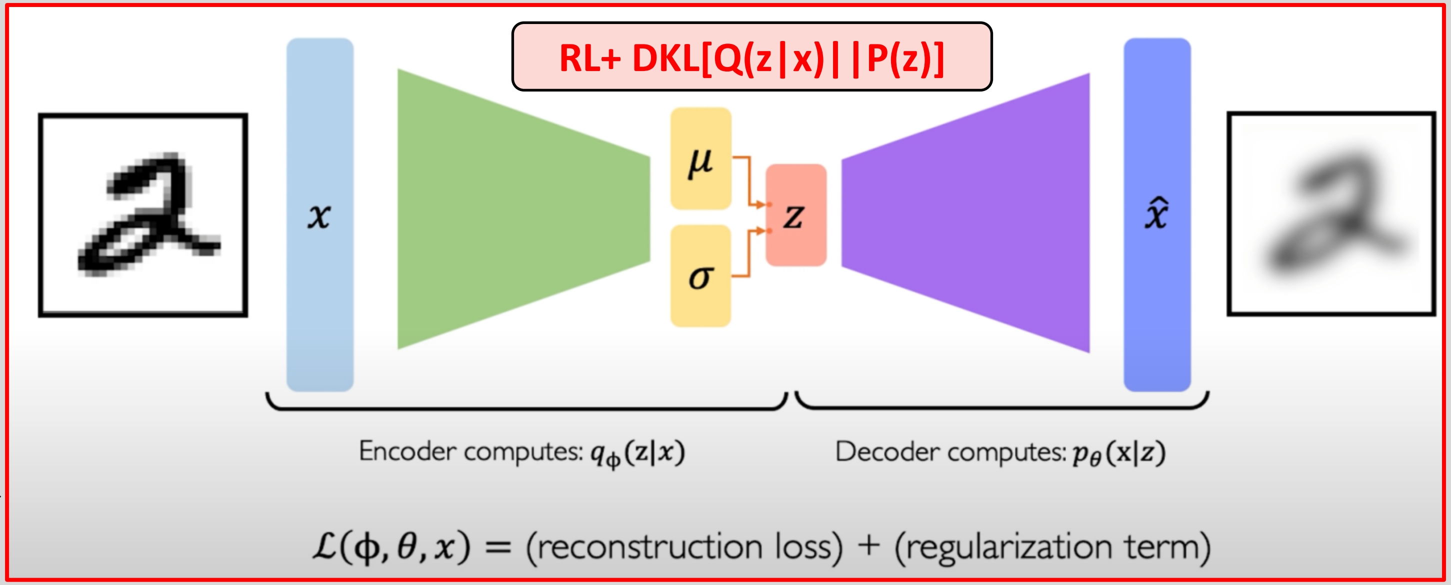


Image Source: introtodeeplearning.com

GENERATIVE ADVERSARIAL NETWORK (GAN)

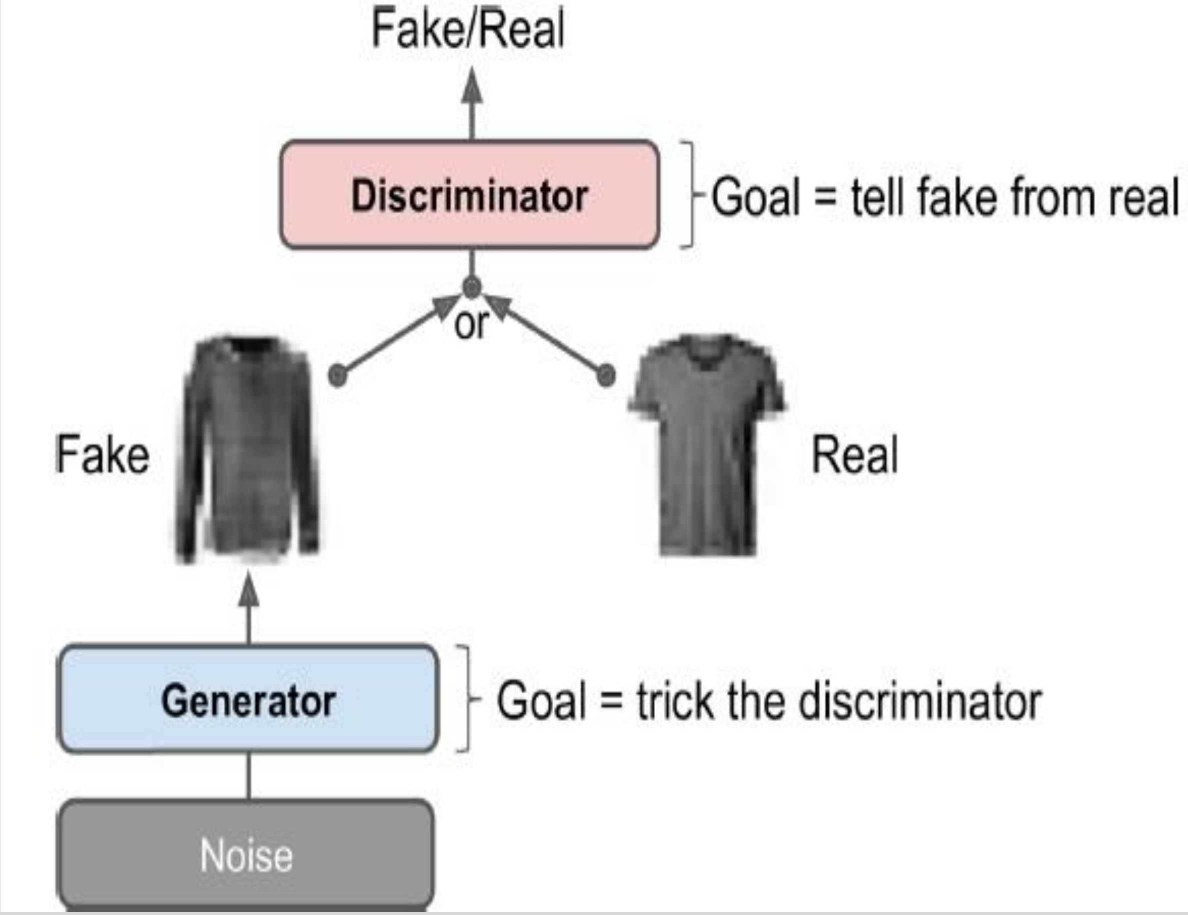
- 2014, **Ian Goodfellow et al.**
- Composed of two neural networks-Generator (**G**) and Discriminator (**D**)
- Neural Networks compete against each other

Generator (G):

- takes input from a random distribution (Gaussian/Normal) and outputs some data
- Random distributions→ similar to latent in VAE
- offers same functionality as decoder in VAE
- Tries to produce inputs (say images) similar to real images to trick the discriminator

Discriminator (D):

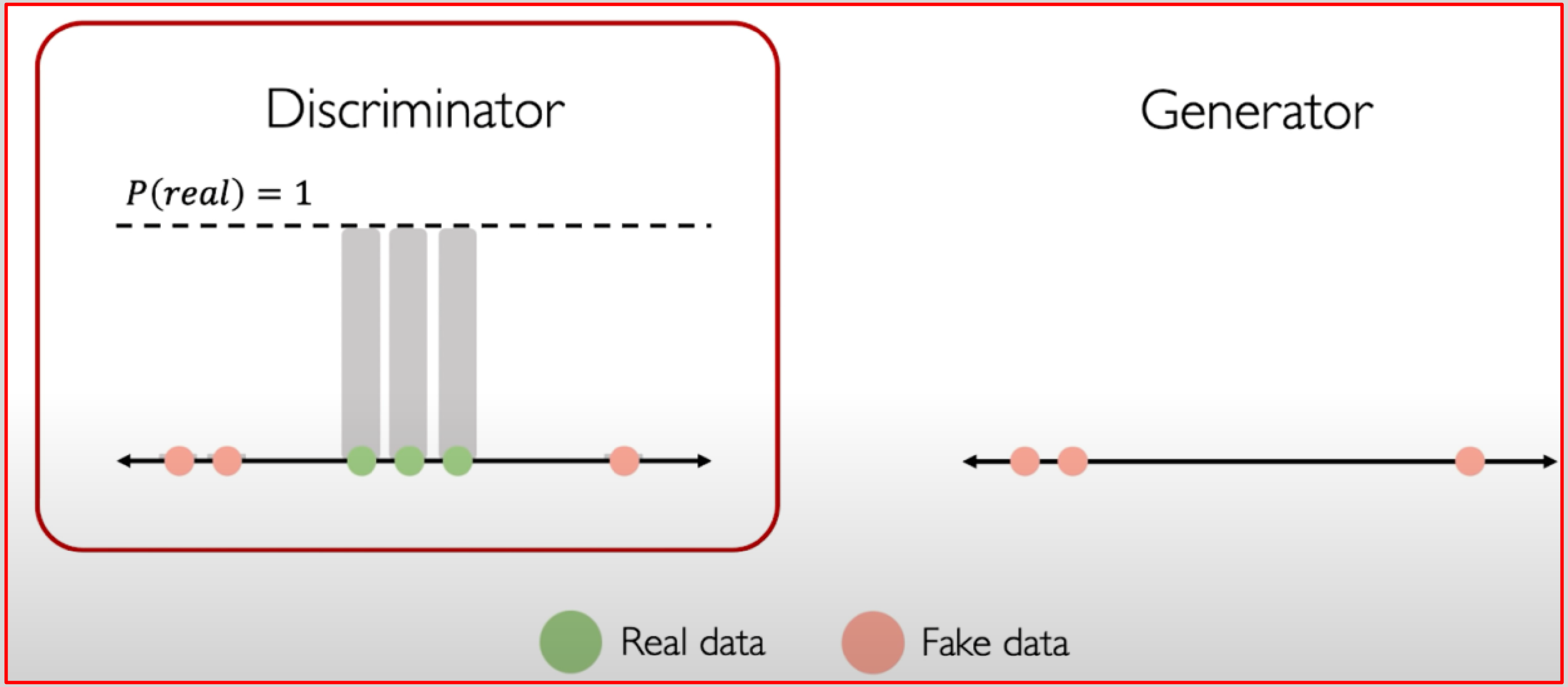
- Takes fake data (say image) from generator or real image from training data to guess if the input image is real or fake.



Source: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

INSIGHT BEHIND GANs

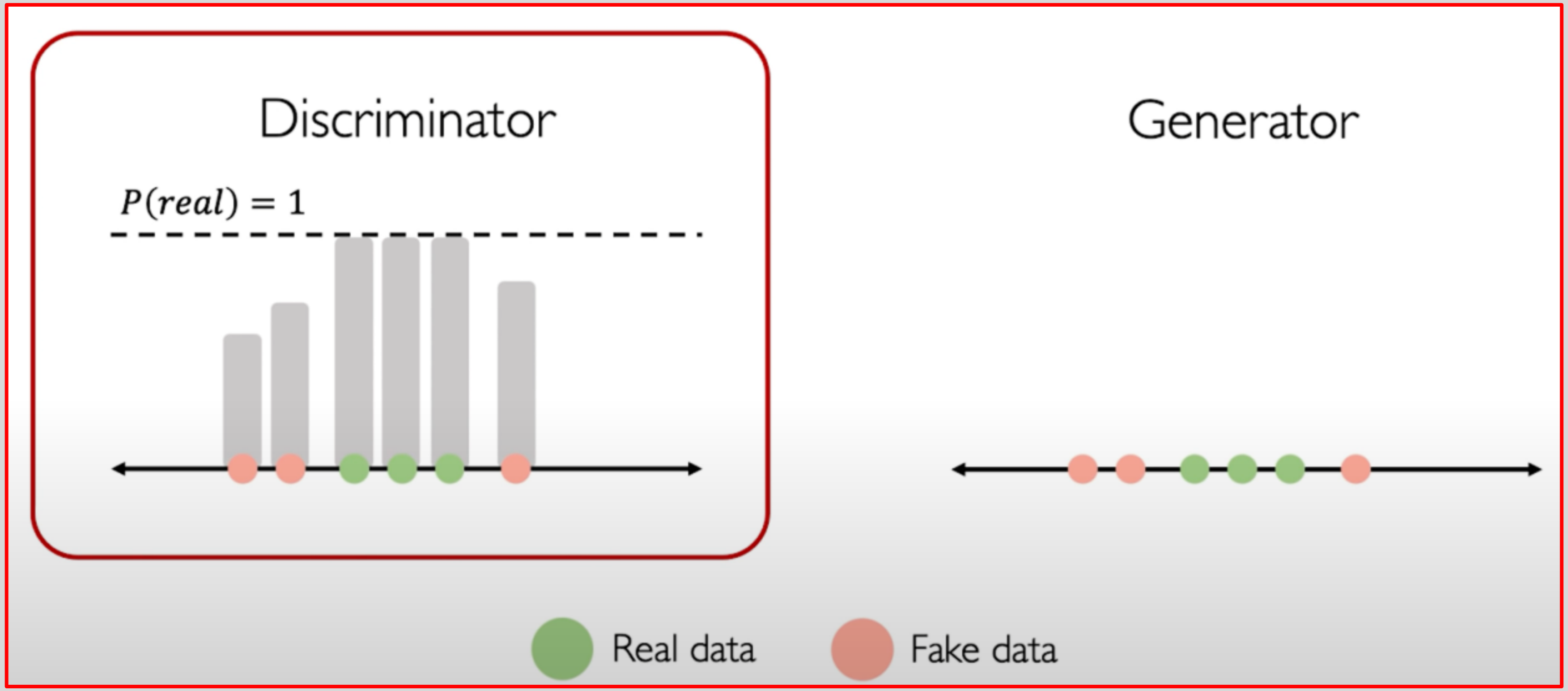
- Generator starts by creating an imitation of data from noise (e.g., Gaussian Noise)
- Discriminator looks at real data as well as data synthesized by the Generator and tries to identity real from fake



Source: introtodeeplearning.com

INSIGHT BEHIND GANs

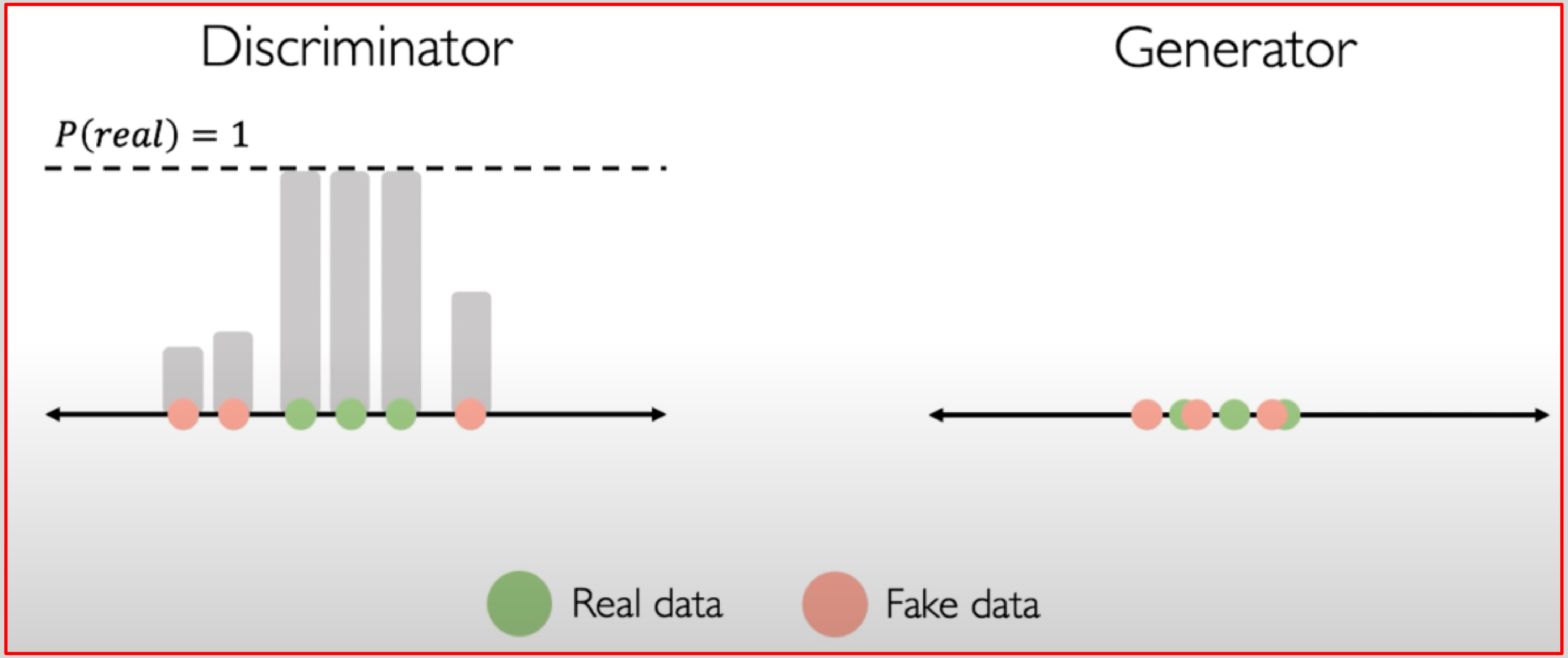
- If the Discriminator is successful in identifying real from fake, the Generator tries to improve its imitation of the data by synthesizing fake data points very close or similar to real data points



Source: introtodeeplearning.com

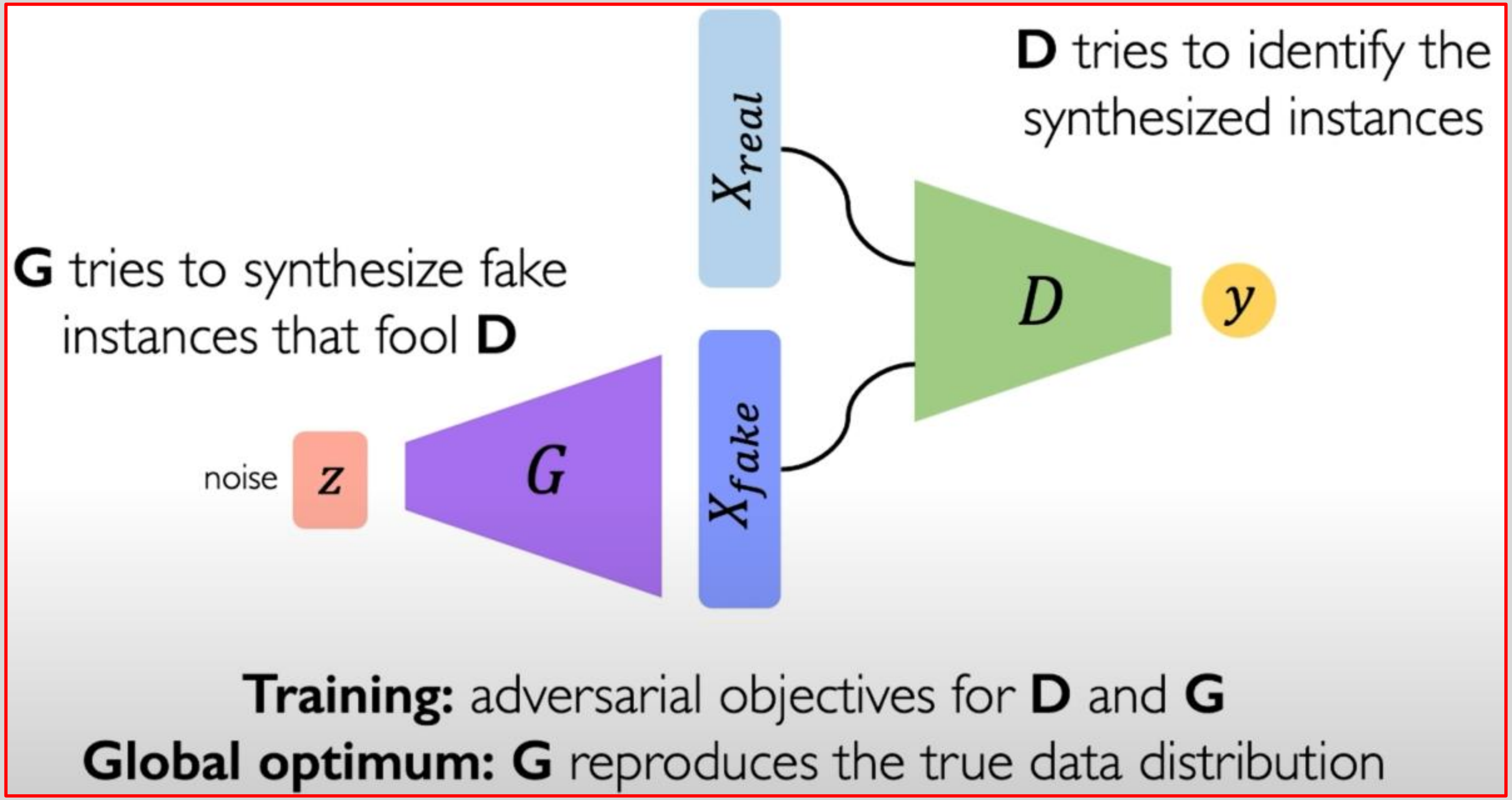
INSIGHT BEHIND GANs

- The Discriminator constantly tries to identify real from fake and with gradual training it learns
- The Generator on the other hand tries to trick the Discriminator until convergence



Source: introtodeeplearning.com

WORKING OF A GAN MODEL



Source: introtodeeplearning.com

GAN TRAINING

Phase 1 (Discriminator Training):

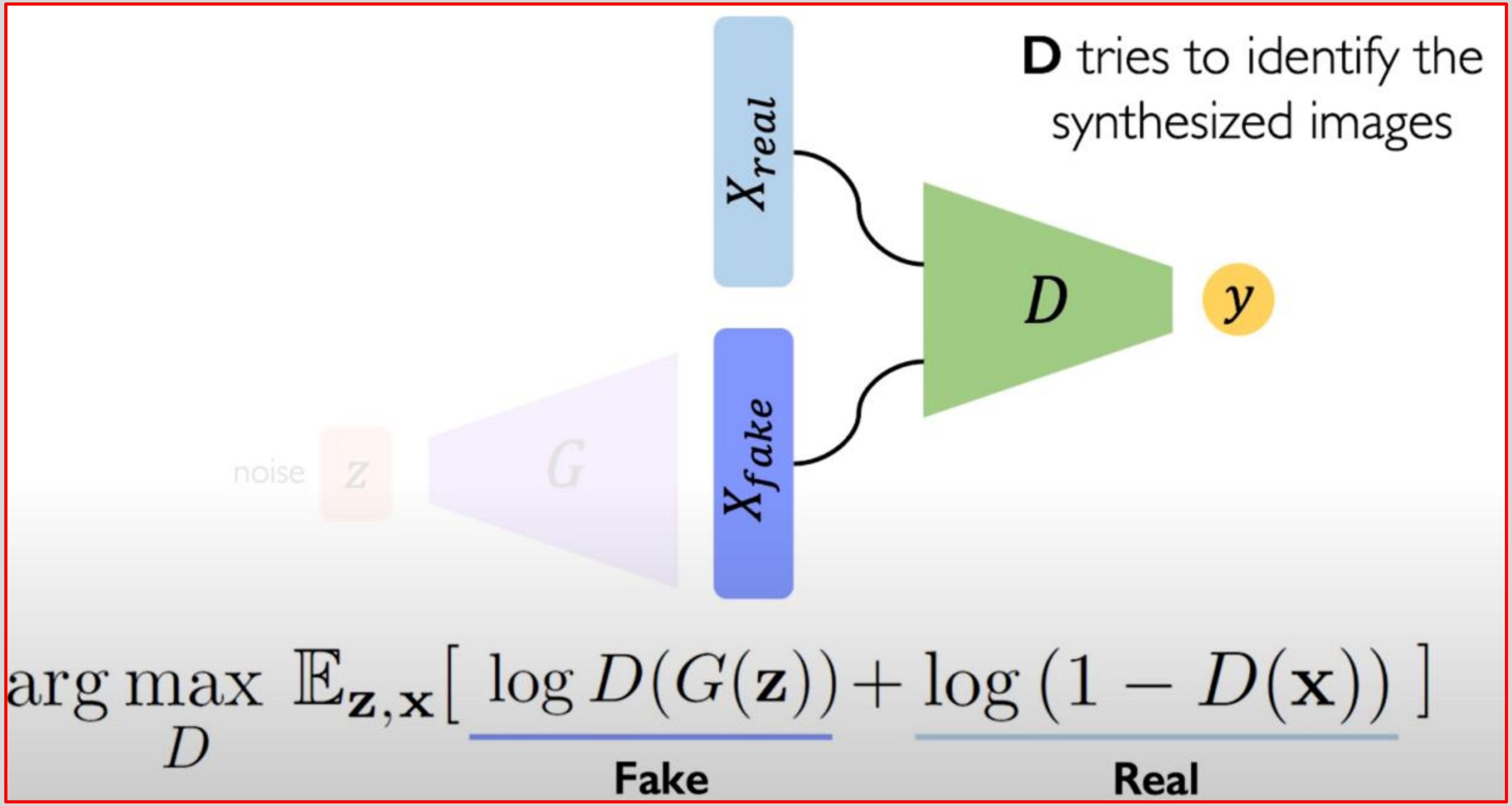
- First train the Discriminator on real and fake images (produced by the Generator)
- Labels for real are set to 1 and for fake are set to 0
- Training for one step
- Backpropagation only for Discriminator→ adjusting loss→ binary crossentropy

Phase 2 (Generator Training):

- Train Generator to produce fake images
- Discriminator identifies real from fake images
- Generator labels the fake images generated as 1 to deceive the Discriminator
- Weights of Discriminator are frozen during this step
- Updates through backpropagation occurs only for the Generator

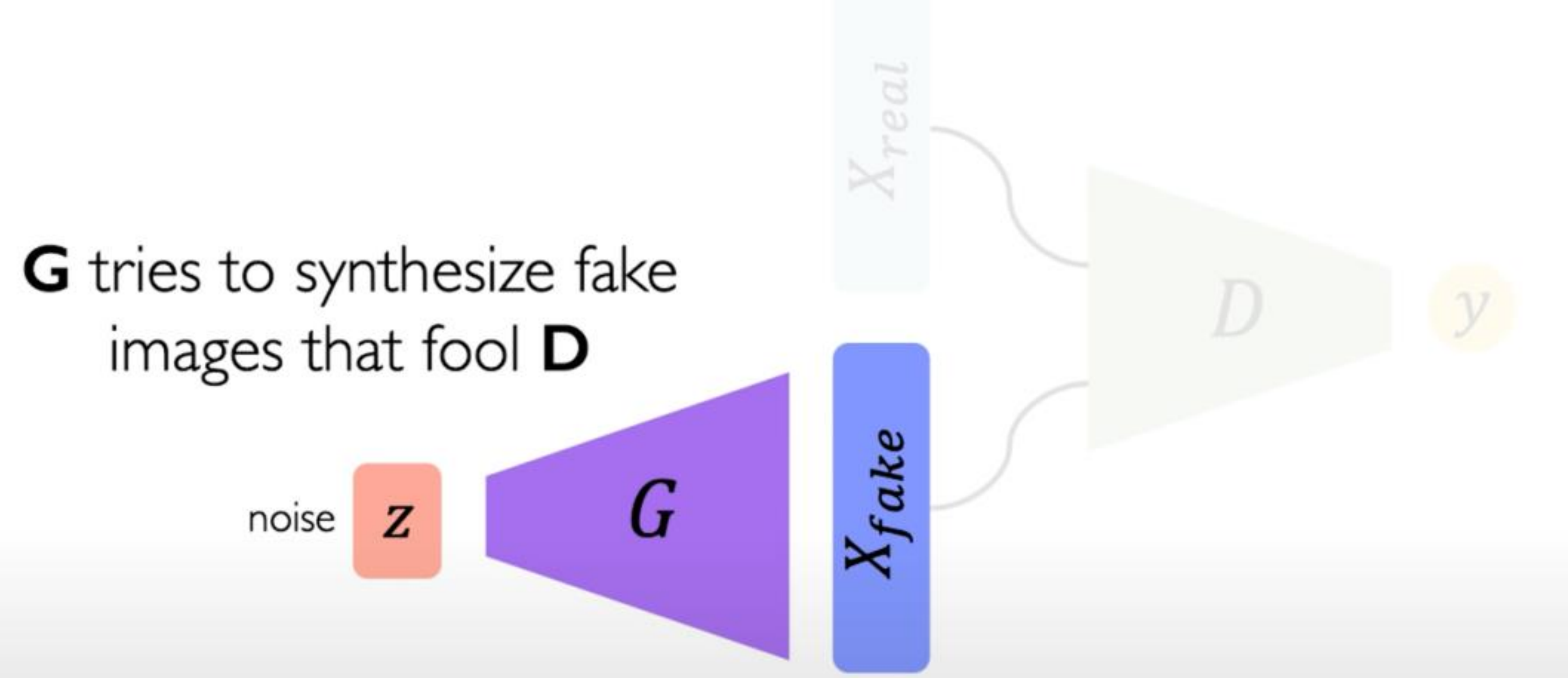
Source: introtodeeplearning.com

GAN LOSS FUNCTION: DISCRIMINATOR



Source: introtodeeplearning.com

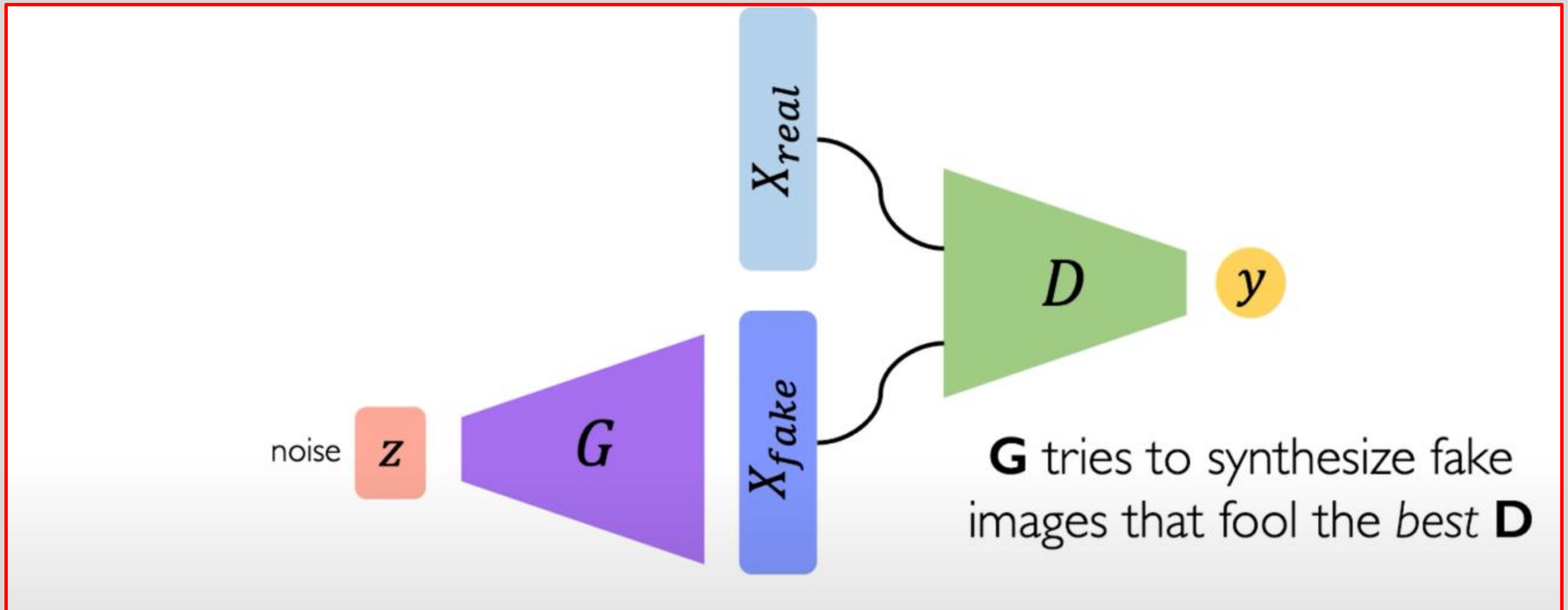
GAN LOSS FUNCTION: GENERATOR



$$\arg\min_{G} \; \mathbb{E}_{\mathbf{z},\mathbf{x}}\left[\, \log D(G(\mathbf{z})) + \log\left(1 - D(\mathbf{x})\right) \,\right]$$

MIN MAX OBJECTIVE FUNCTION OF GAN

- Ultimately the Discriminator will try its best to learn and identify fake and real and G tries to fool this best D

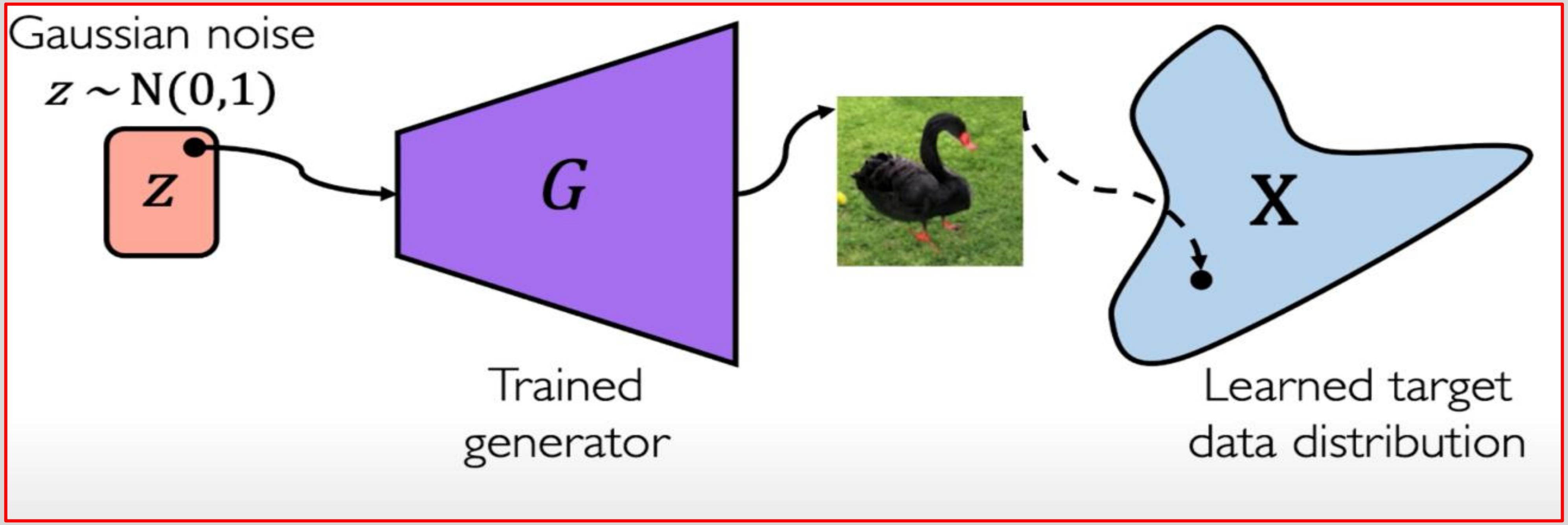


$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{z},\mathbf{x}}\left[\ \log D(G(\mathbf{z})) + \log\left(1 - D(\mathbf{x})\right)\ \right]$$

Source: introtodeeplearning.com

SYNTHESIZING NEW DATA INSTANCES USING TRAINED GAN

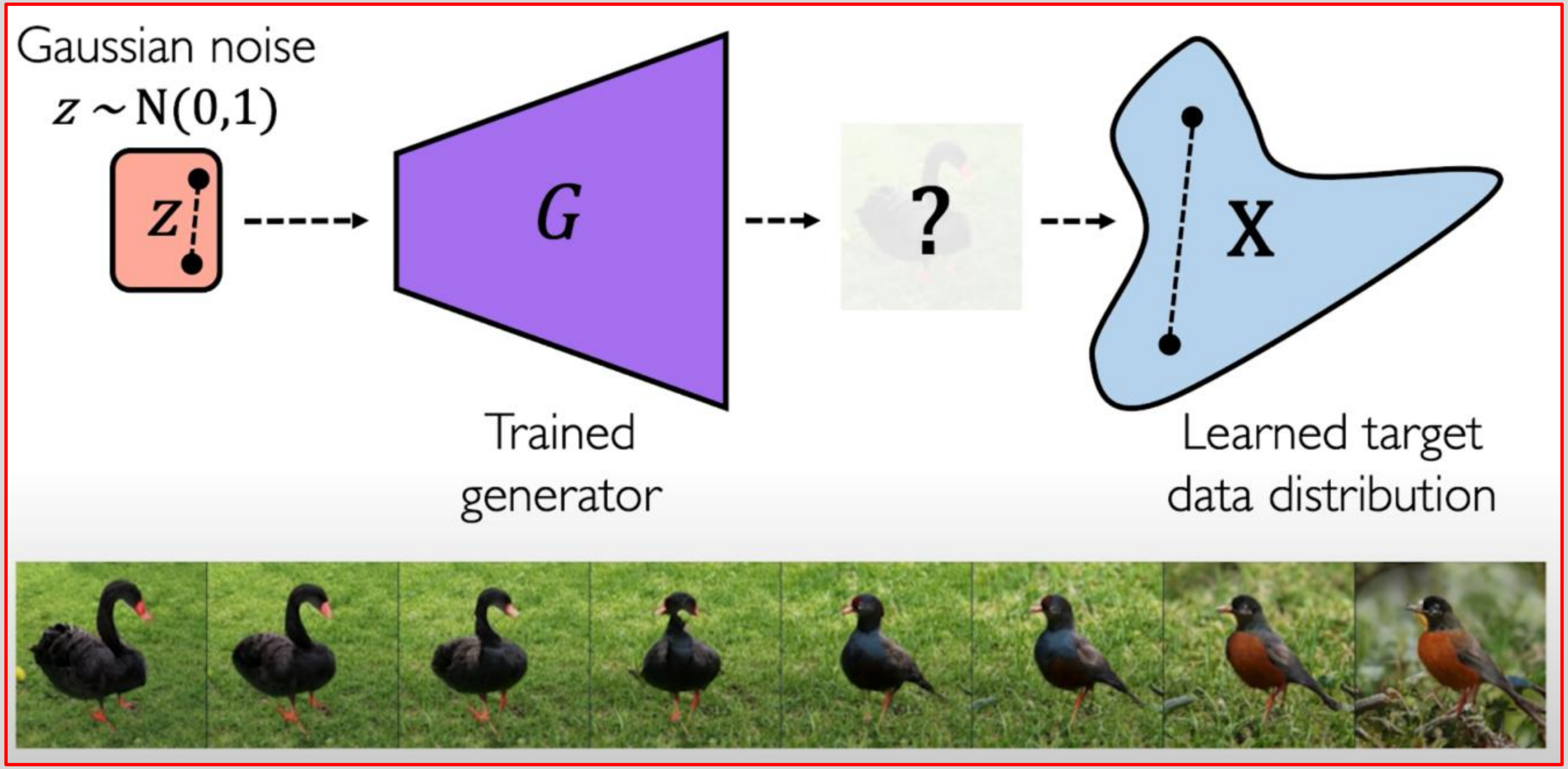
- During Adversarial training, G learns the mapping from a Gaussian Distribution (noise) to target data distribution
- The pre-trained generator model can now be used to create new data instances that you have never seen before
- One point from a latent distribution will result in a particular output in the target data space



Source: introtodeeplearning.com

SYNTHESIZING NEW DATA INSTANCES USING TRAINED GAN

- Traverse or interpolate through Gaussian Noise to synthesize an interpolation in target distribution



Source: introtodeeplearning.com

COMMON GAN PROBLEMS: FAILURE MODES

- A number of problems→ active research areas

1) Vanishing Gradient Problem

- D too good, G does not have enough information to learn, no progress
- **Solutions:** using modified minmax loss and Wasserstein loss

2) Mode Collapse Problem

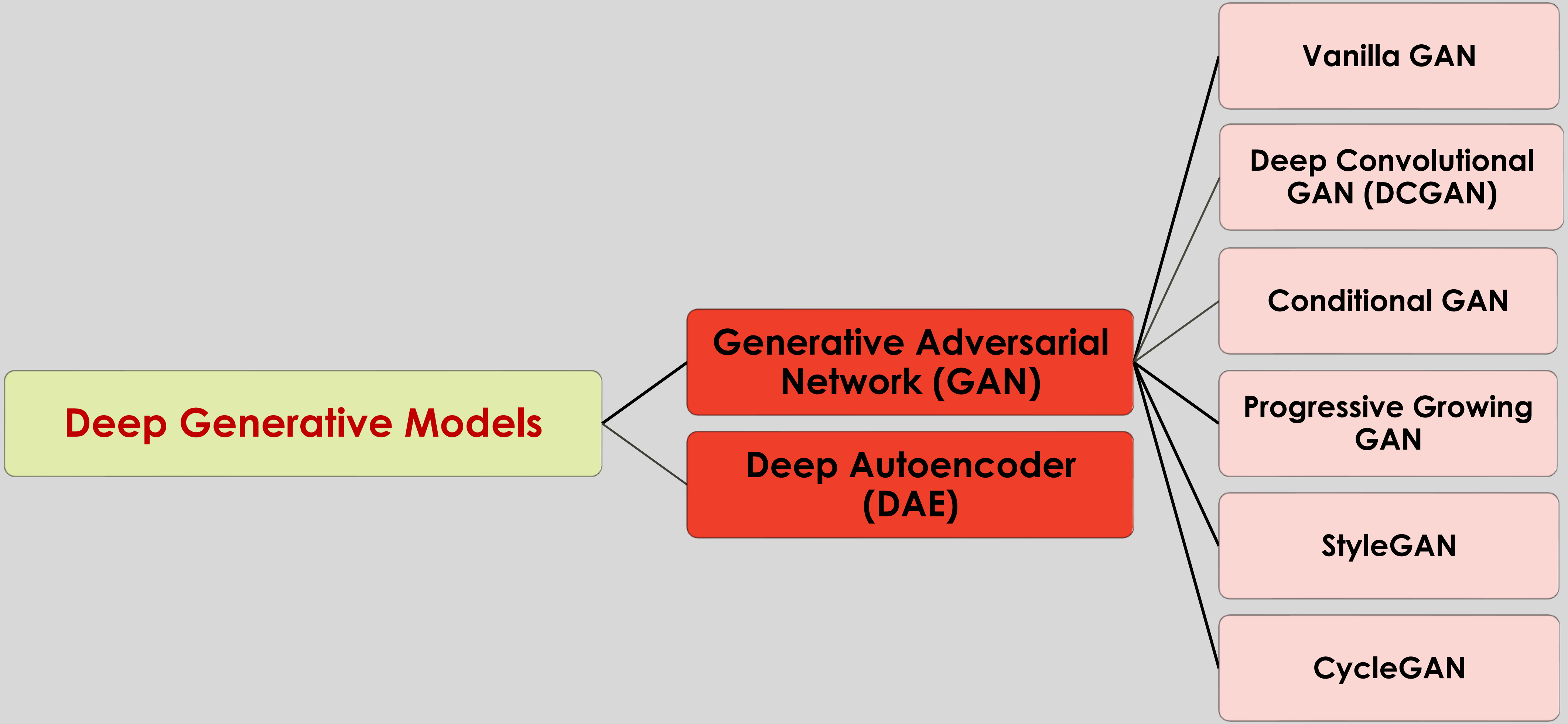
- G is always trying to find the one output that seems most plausible to the D
- D always rejects that output as fake→ stuck in local minima
- G overoptimizes for D and D never learns
- **Solutions:** Wasserstein loss and Unrolled GANs

3) Convergence Failure Problem

- GANs fail to converge, unstable training, complex architecture, min-max game
- **Solutions:** Regularization, add noise to D inputs, Penalizing D weights

Source: https://developers.google.com/machine-learning/gan/problems

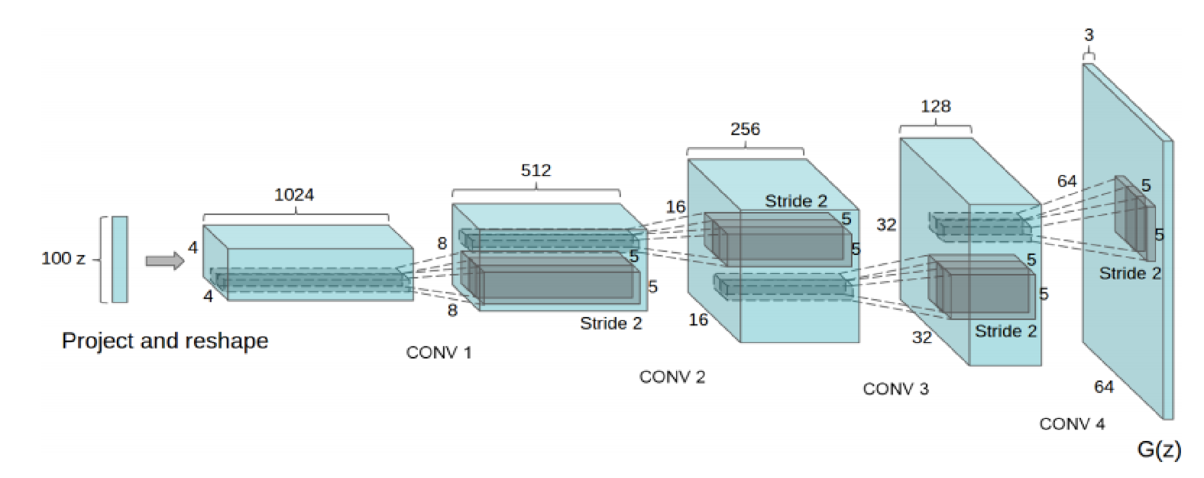
RECENT ADVANCES OF GAN



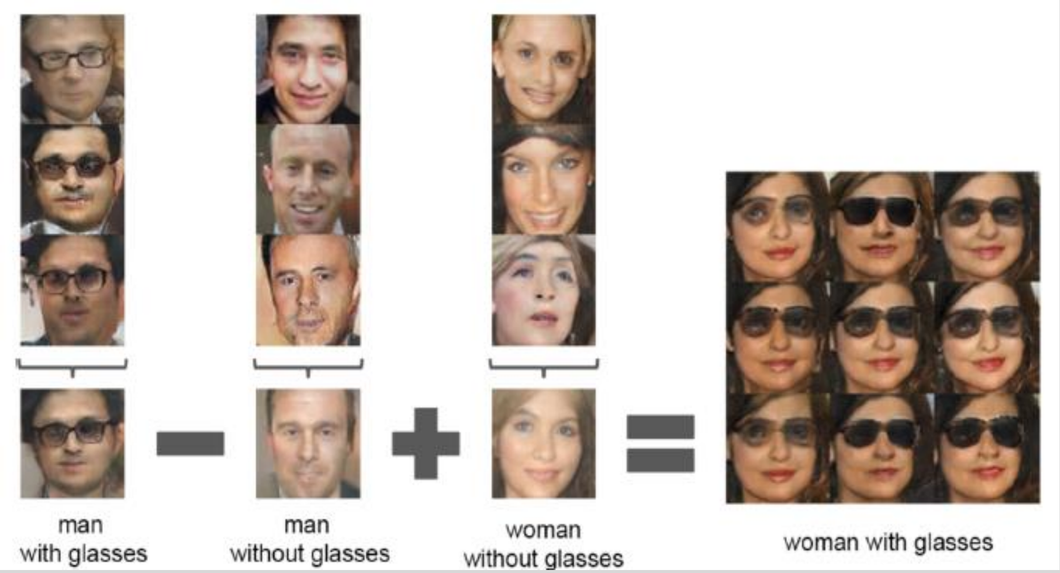
Source: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

DEEP CONVOLUTIONAL GAN (DCGAN)

- **Alec Radford et al.** in 2015
- Improve training of vanilla GAN
- Deeper convolutional nets for larger images
- Vector arithmetic for visual concepts



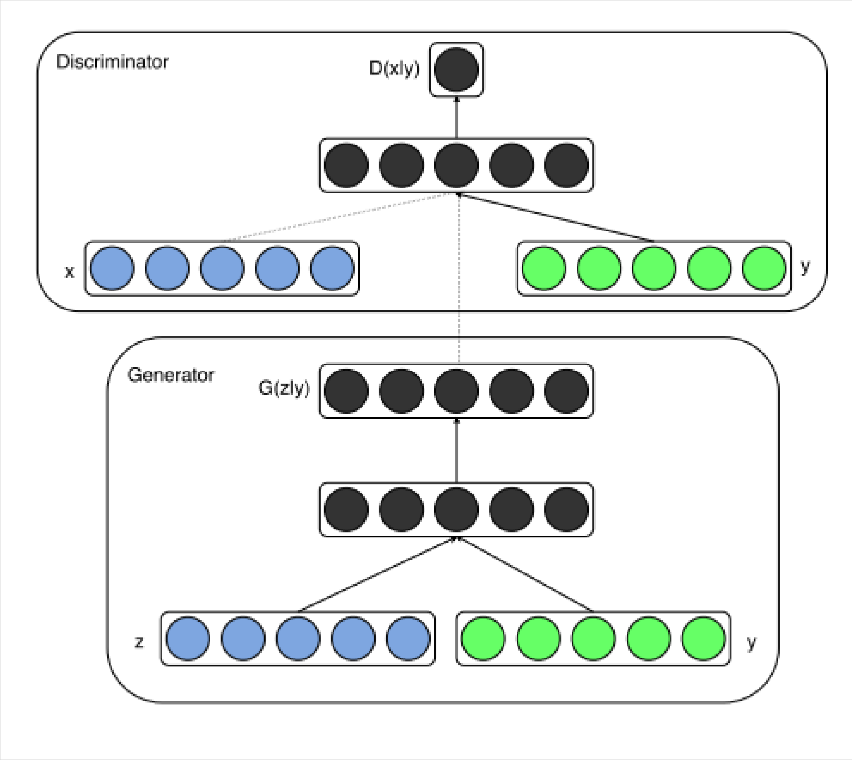
DCGAN Generator Architecture



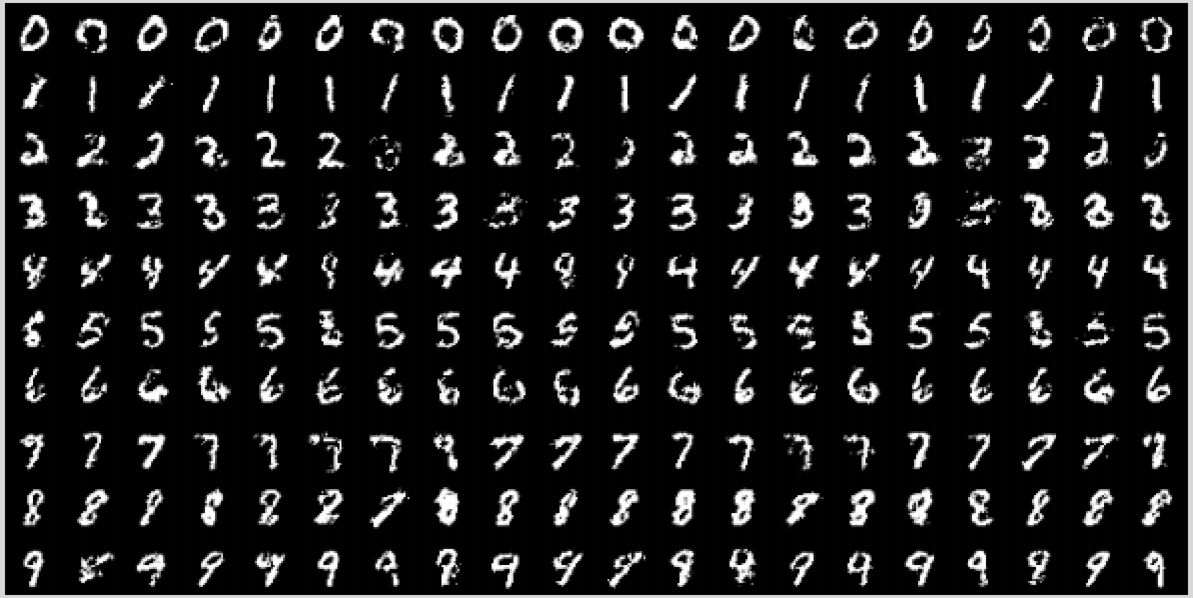
Source: DCGAN paper (https://arxiv.org/abs/1511.06434)

CONDITIONAL GAN

- conditional version of generative adversarial nets
- **Mirza et al.** in 2014
- feeding the data along with "y" to condition on to both the generator and discriminator



Conditional GAN Architecture

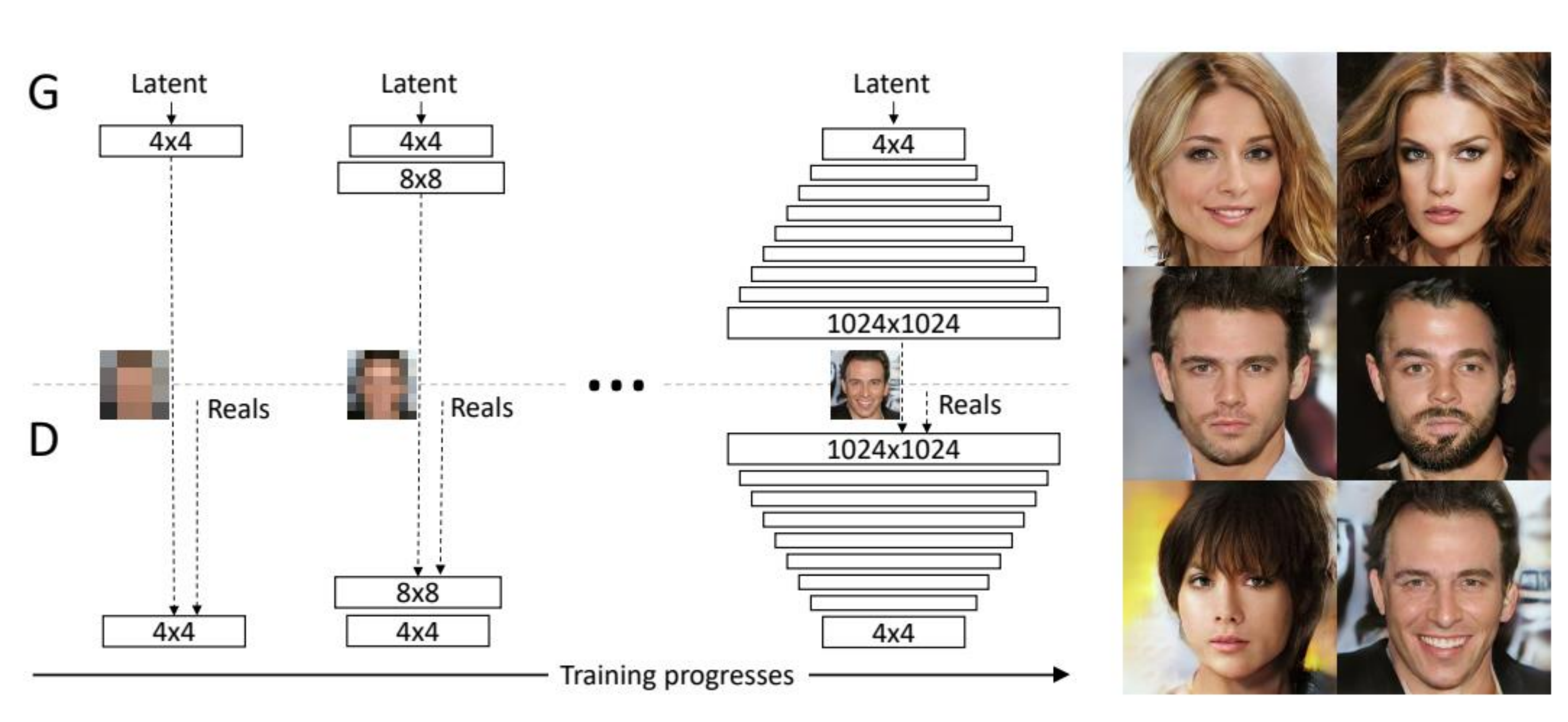


Generated MNIST digits, each row conditioned on one label

Source: Conditional GAN paper (https://arxiv.org/pdf/1411.1784.pdf)

PROGRESSIVE GROWING GANs

- Nvidia researchers **Tero Karras et al.**, 2018
- Greedy layerwise training
- generating small images at the beginning of training, then gradually adding convolutional layers to both the generator and the discriminator to produce larger and larger images
- (4 × 4, 8 × 8, 16 × 16, …, 512 × 512, 1,024 ×1,024)



Source: Progressive Growing GANs paper (https://arxiv.org/abs/1710.10196)

StyleGAN

- Nvidia researchers **Tero Karras et al.**, 2018

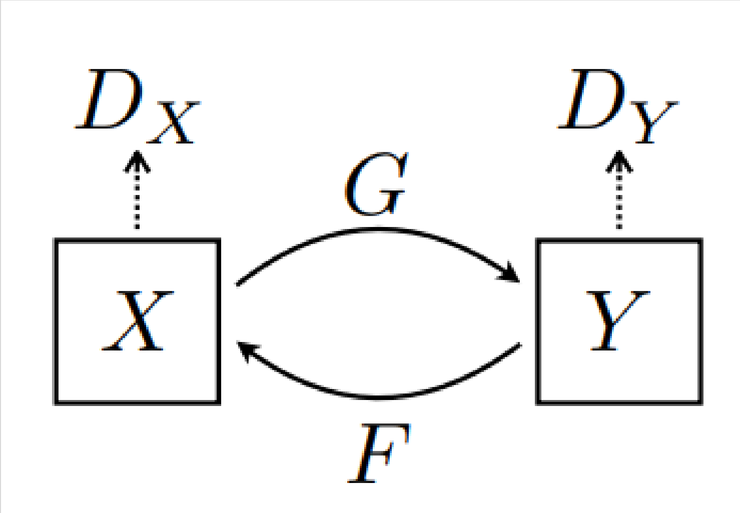
- Generator→ style transfer techniques to ensure that the generated images have the same local structure as the training images

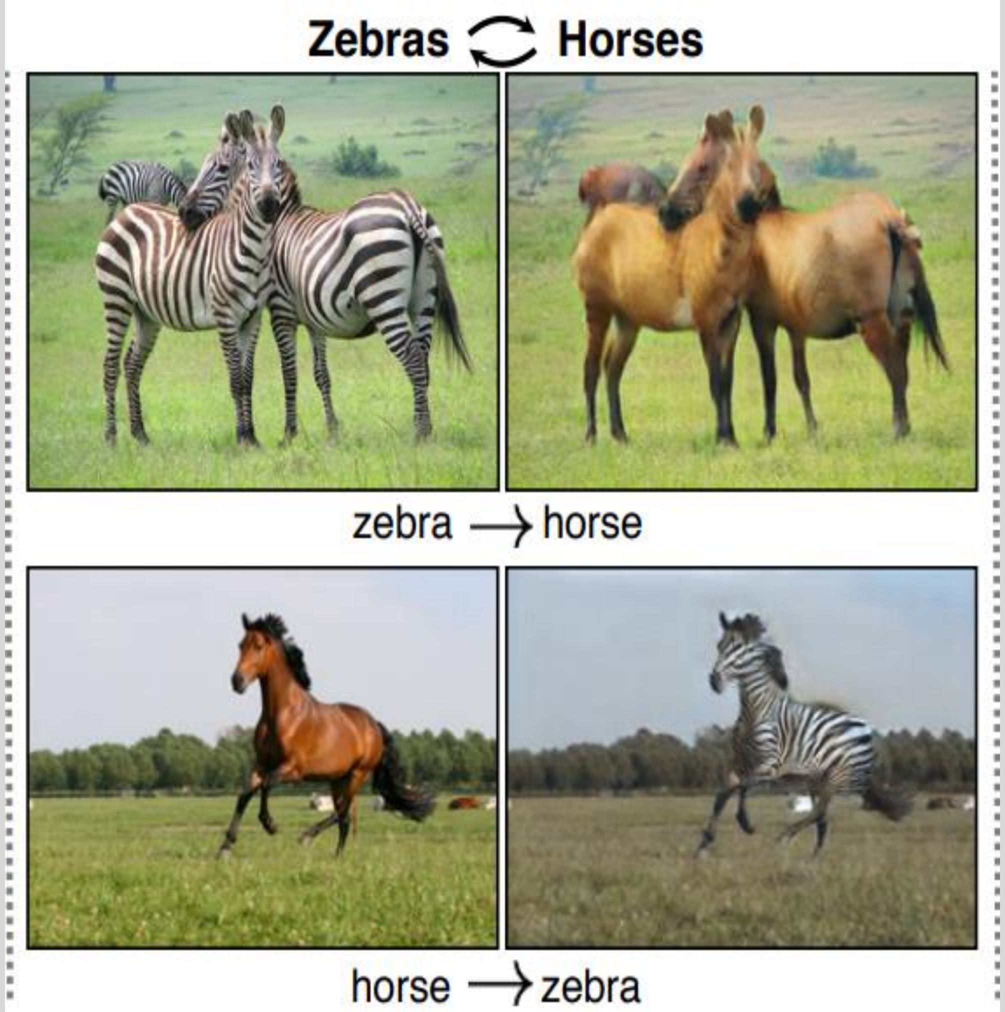
- improving the quality of the generated images

Source: StyleGAN paper (https://arxiv.org/pdf/1812.04948.pdf)

CycleGAN

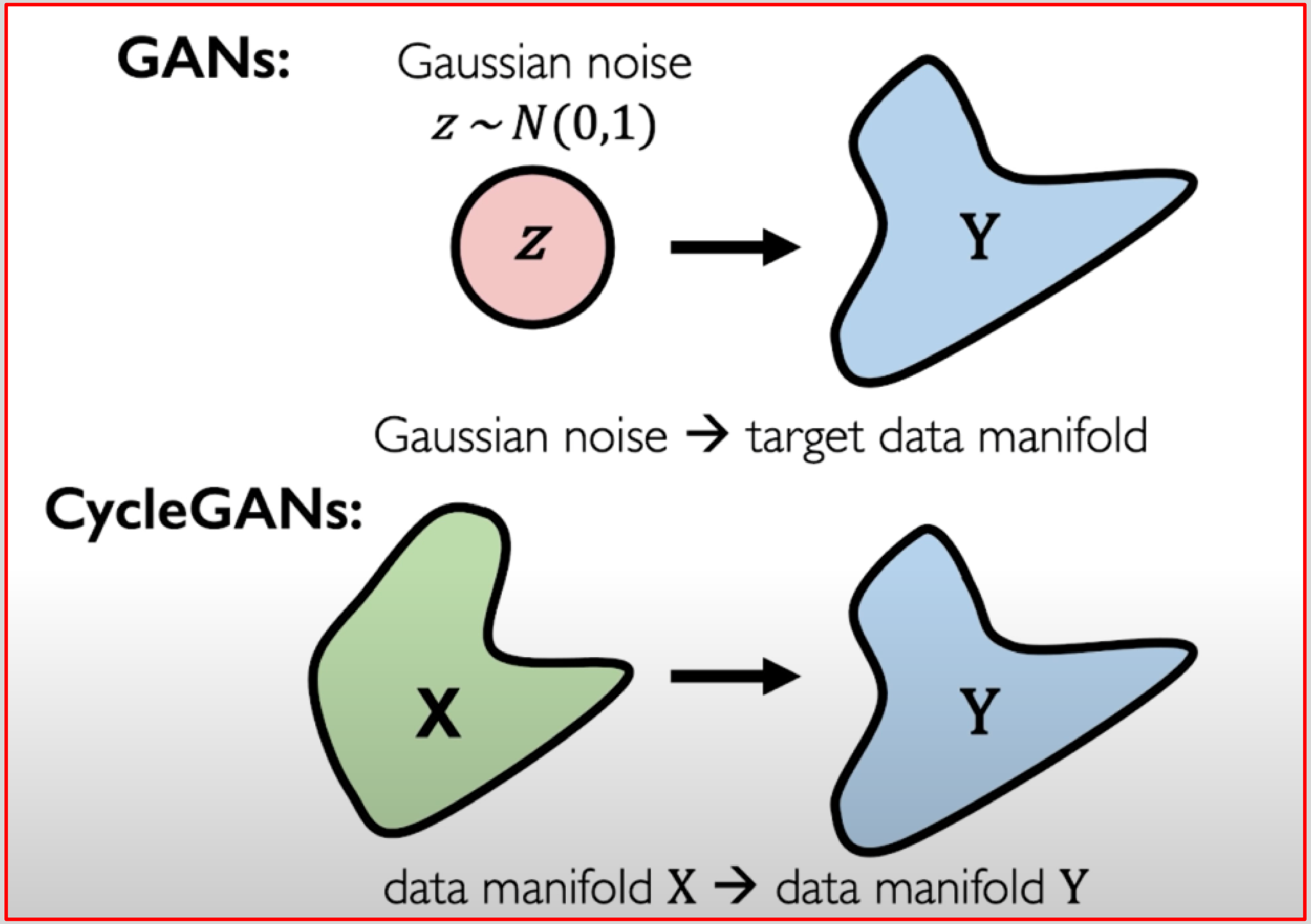
- **Zhu et al.**, 2017
- Unpaired Image-to-image translation
- Images from one domain learnt and translated to another domain without having any paired corresponding image in another domain
- Two Generators and two Discriminators





Source: CycleGAN paper (https://arxiv.org/pdf/1703.10593.pdf)

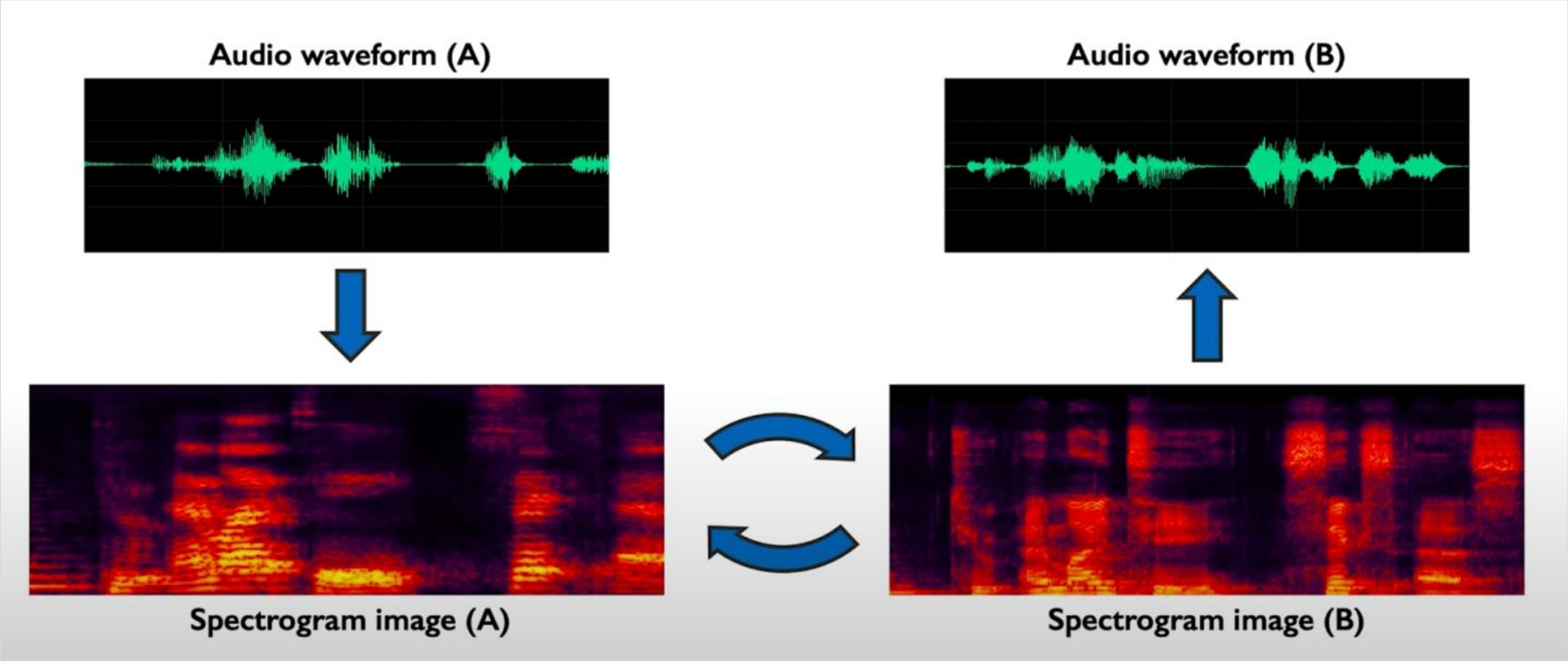
INSIGHT BEHIND CycleGANs



Source: introtodeeplearning.com

TRANSFORM SPEECH USING CycleGAN

- Synthesize speech in someone else's voice
- Audio recordings (A) & a set of audio recordings (B)
- Convert to image spectrogram format and then apply CycleGAN for translation
- For example: https://www.youtube.com/watch?v=l82PxsKHxYc



Source: introtodeeplearning.com

GAN VERSUS VAE

GAN

- Generator→ fool the Discriminator
- Discriminator→ classify fake and real data
- High quality images generated
- Higher diversity
- Difficult to train→ complex architecture→ unstable

VAE

- Reconstruct real data
- Blurry images based on minimizing reconstruction error
- Lower diversity
- Easier to train→ higher stability

GAN PROGRAMMING EXERCISE

THANKS!

Do you have any questions?