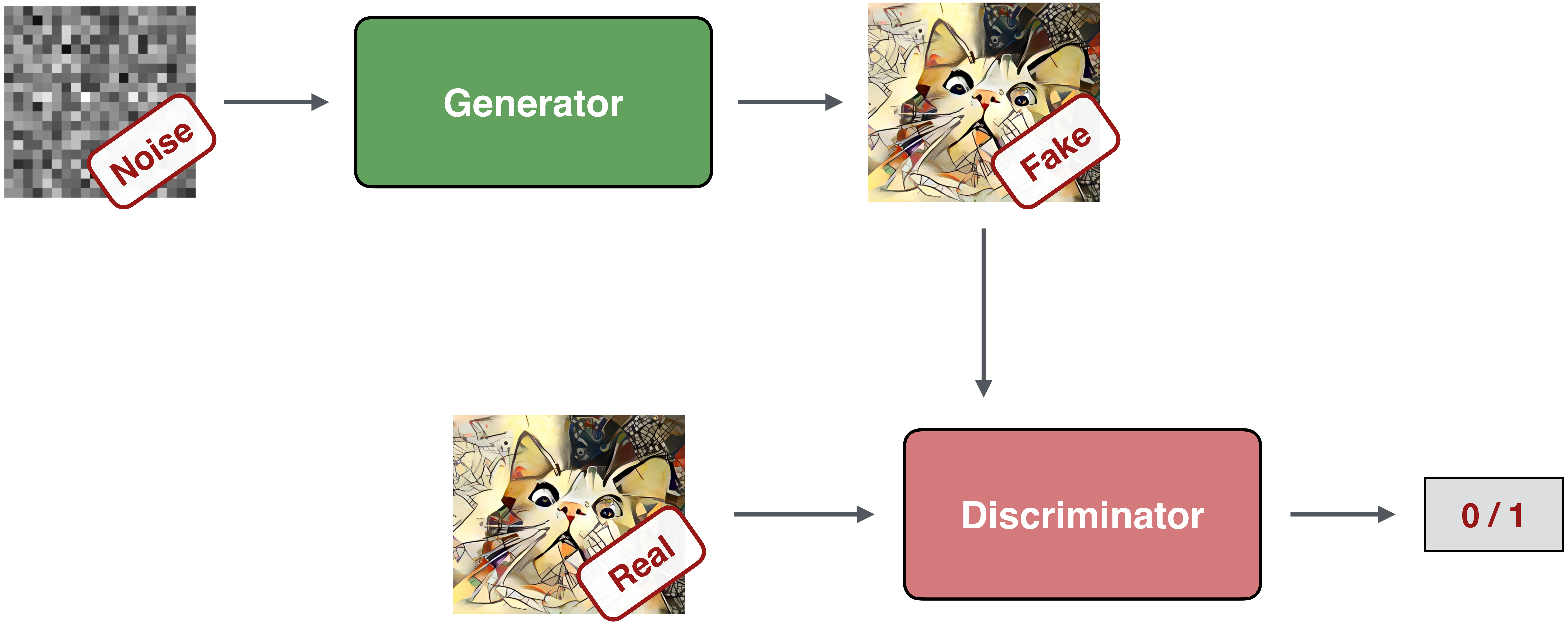
Tutorial 7: DCGAN

Practical Deep Learning for Science
30 May, 2023

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

- Nilotpal

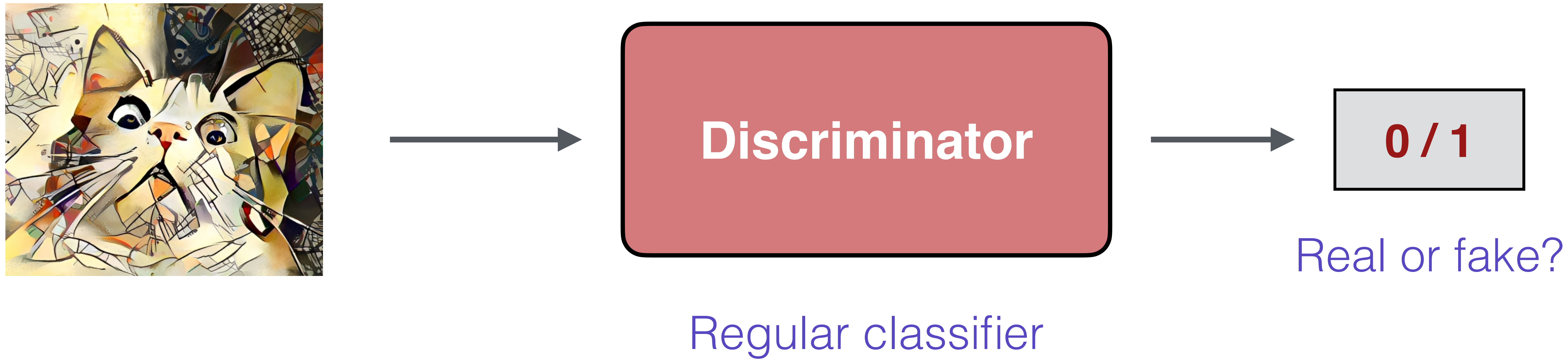
GAN



Regular classifier

Loss? It’s a bit tricky…

Discriminator is a regular binary classifier



Discriminator

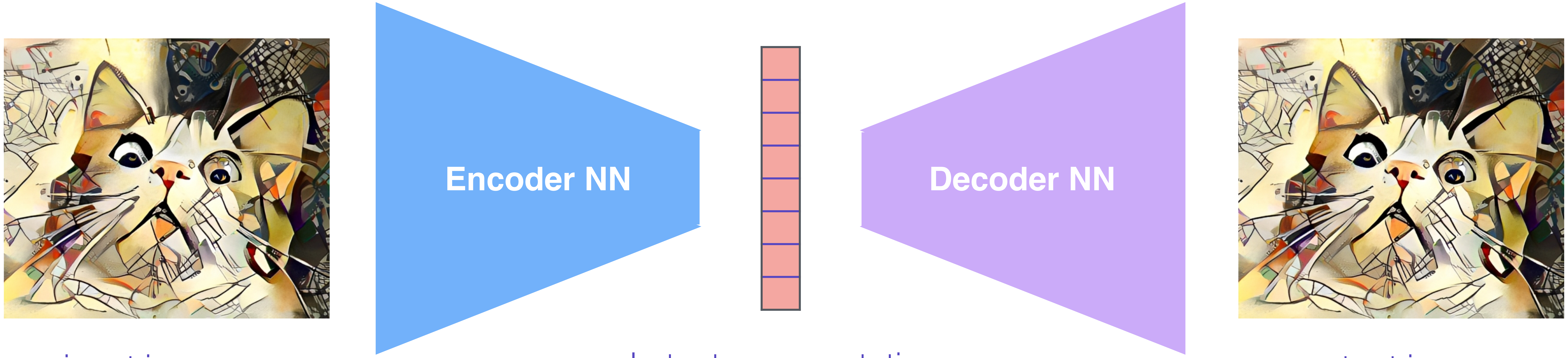
0 / 1

Real or fake?

Regular classifier

Binary Cross entropy loss

Generator loss?



input image
(x)

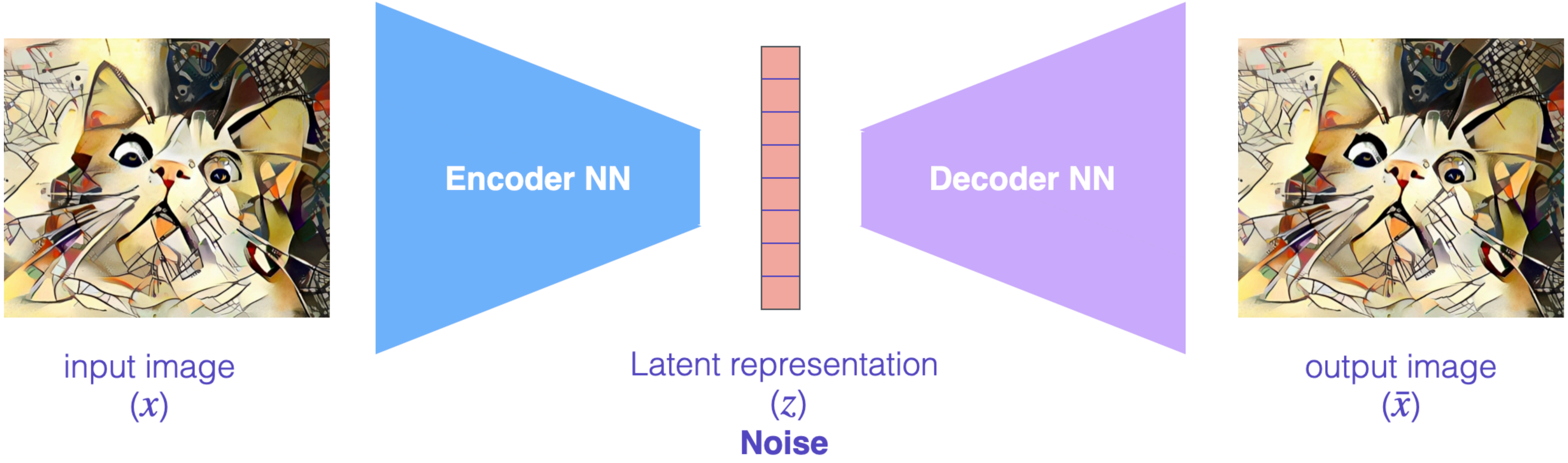
Latent representation
(z)
Noise

output image
$(\bar{x})$

AutoEncoder

Loss (input image, output image

Generator loss?



Encoder NN

Decoder NN

input image
(x)

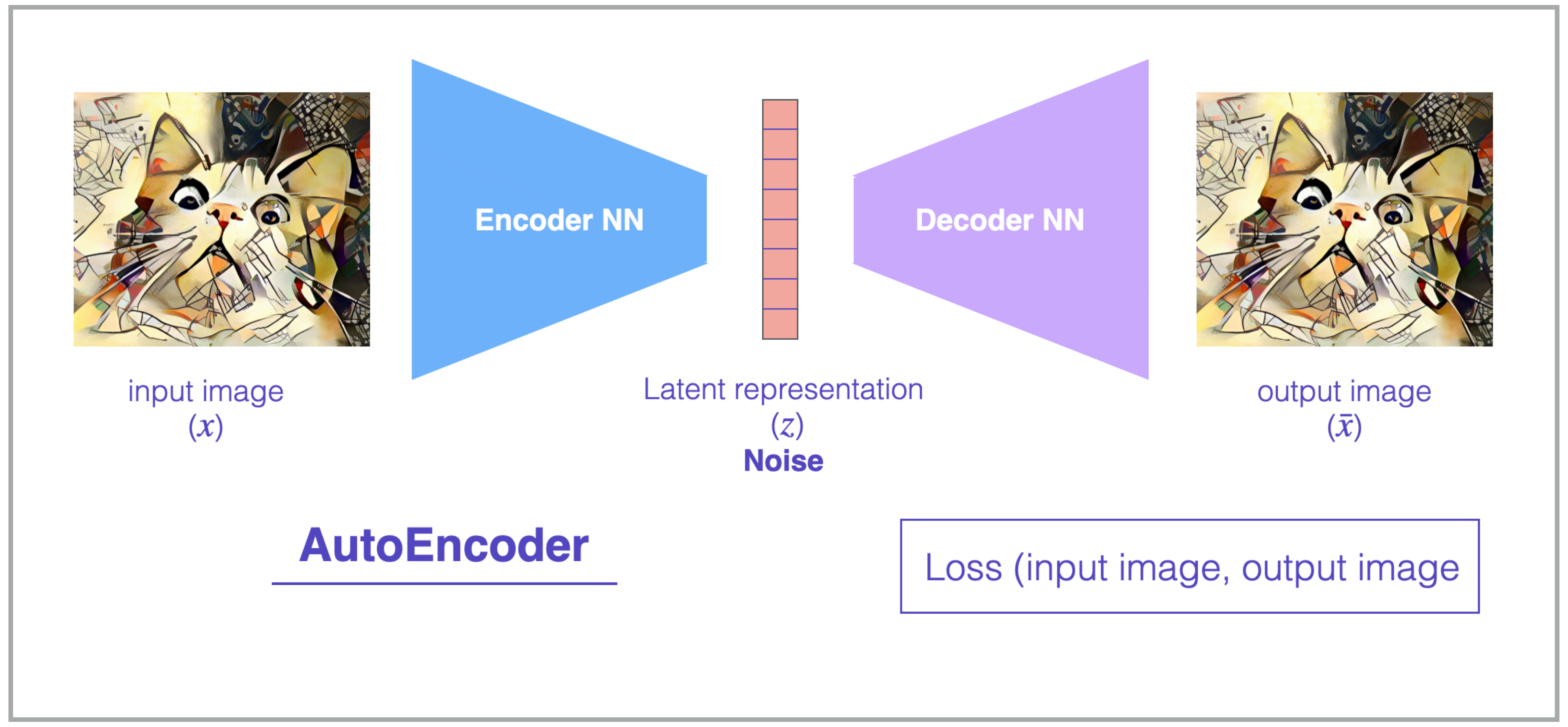
Latent representation
(z)
Noise

output image
$(\bar{x})$

AutoEncoder

Loss (input image, output image

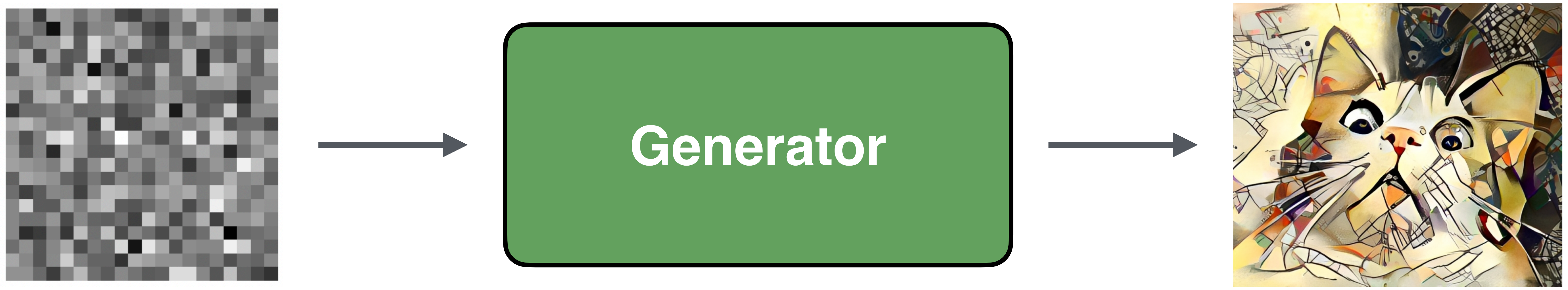
Generator loss?



AutoEncoder

Loss (input image, output image

No input image
Can't compute loss like in AE



Noise

Generator

Output image

Adversary

- Generator and Discriminator have exact opposite goal
- Will use the discriminator loss again
  - But we will flip the label. *Label the generated images as 1 (real)*
  - Want to increase the likelihood of the fake image being classified as real

- From the discriminator perspective
  - Classify real images as real, fake images as fake
- From Generator's perspective
  - *Generate fake images that gets classified as real*

GAN recipe

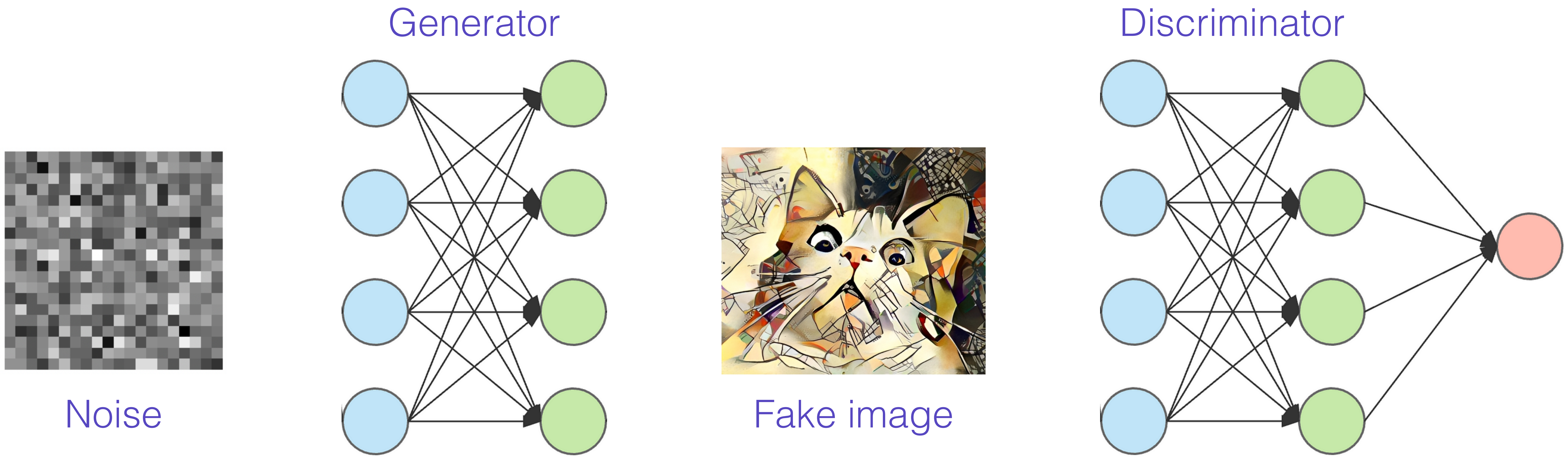
- Generate fake images
- Take real images
- Compute discriminator loss; update discriminator

- Label the fake images as real
- Compute discriminator loss; update generator

.detach()

During **discriminator training,** Torch will see **(Generator + Discriminator)** as **one big network**

Losses will get back propagated to generator as well



We need to detach the image from the generator with `fake_img.detach()`