Deep (5): Generative models

Matthieu Cord LIP6 / SU

Generation

Writing Poems?

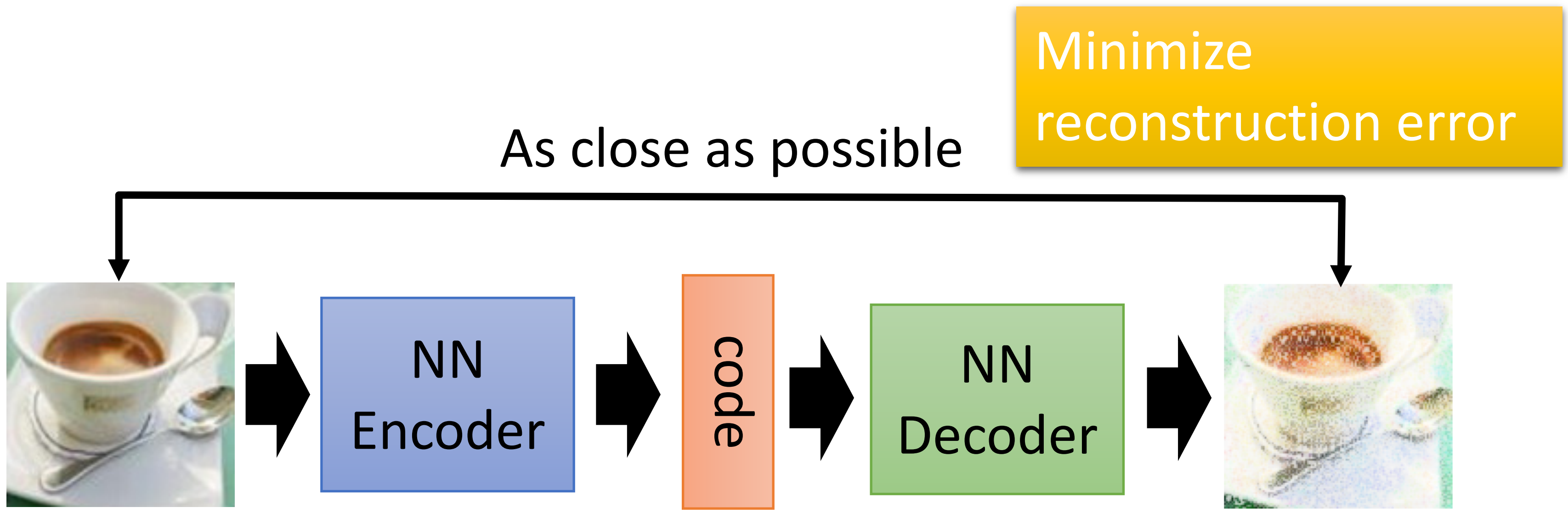
Drawing?

Generative models

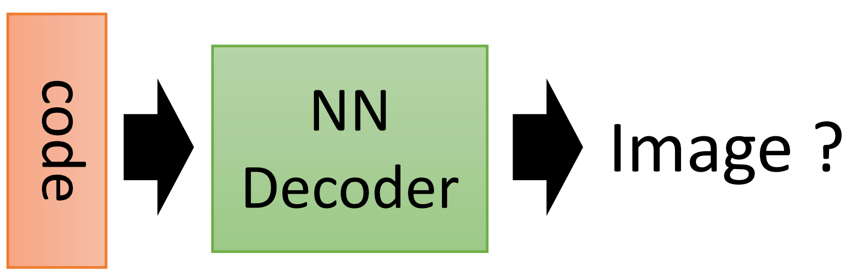
Outline

1. Preview: Auto-Encoders, VAE
2. Generative models
3. GAN architectures for image generation

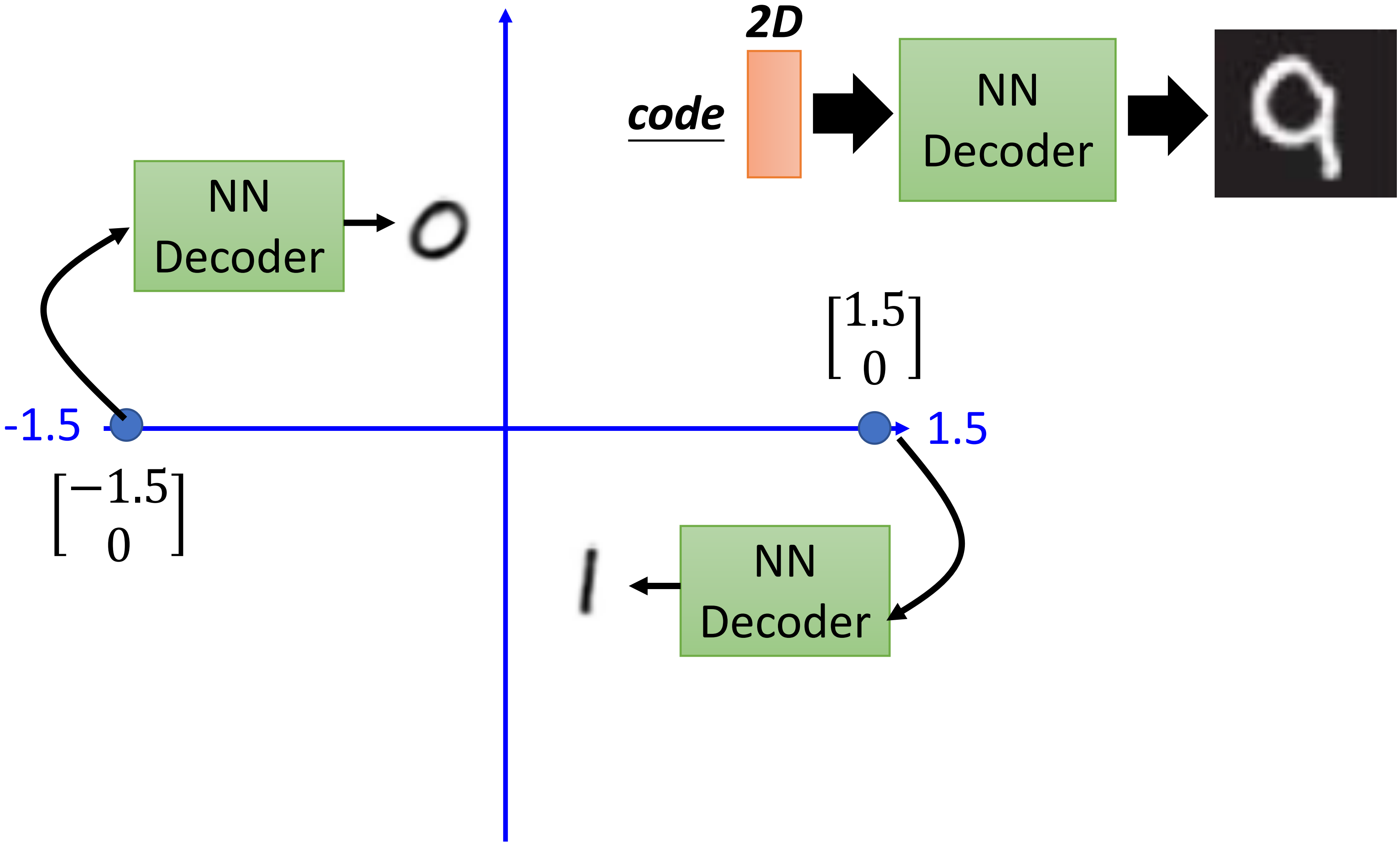
Review: Auto-encoder



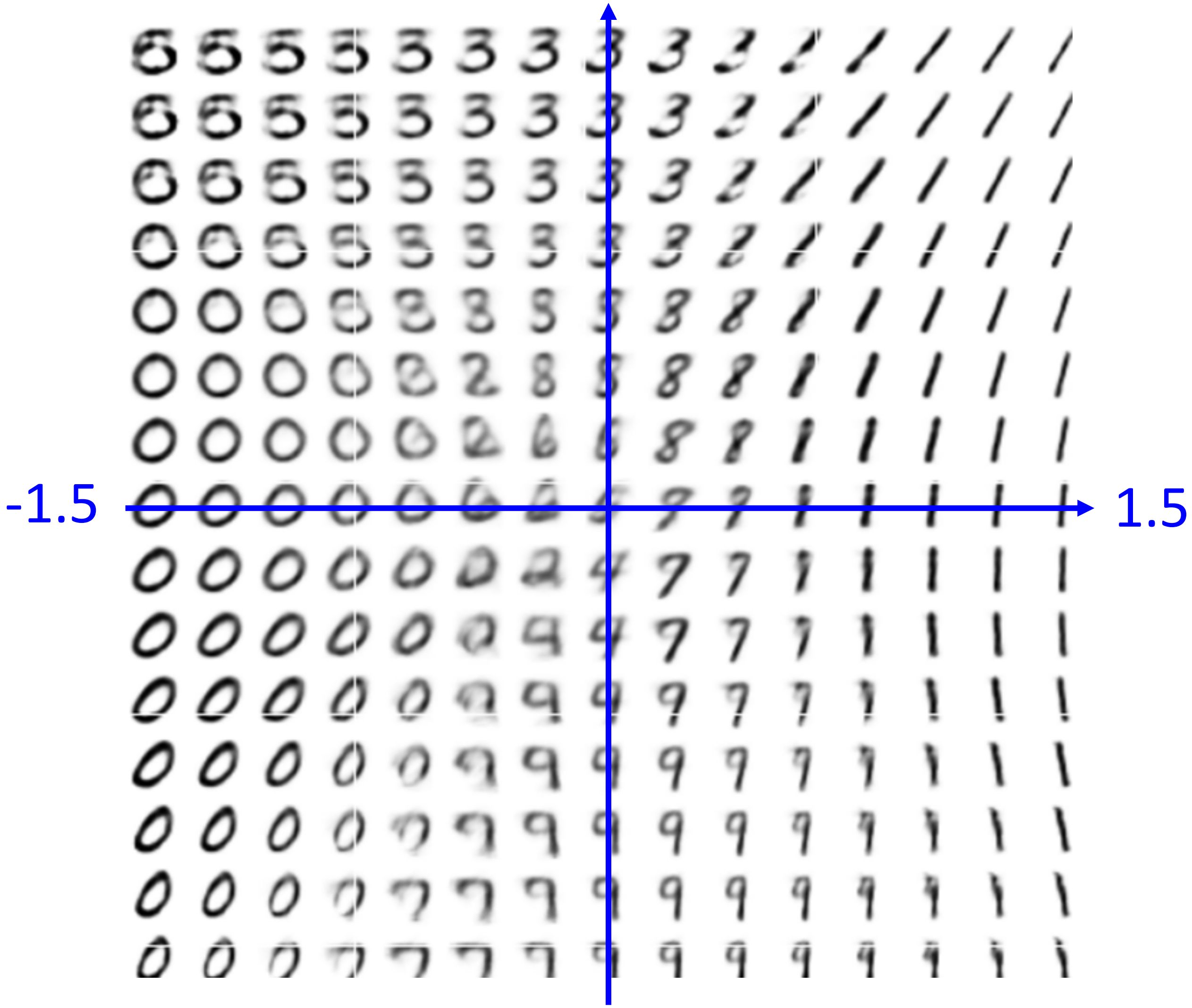
Randomly generate
a vector as code



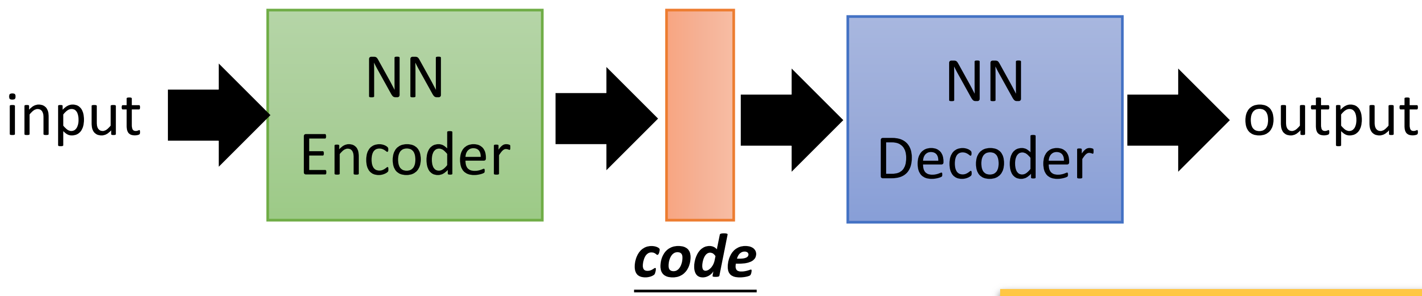
Review: Auto-encoder



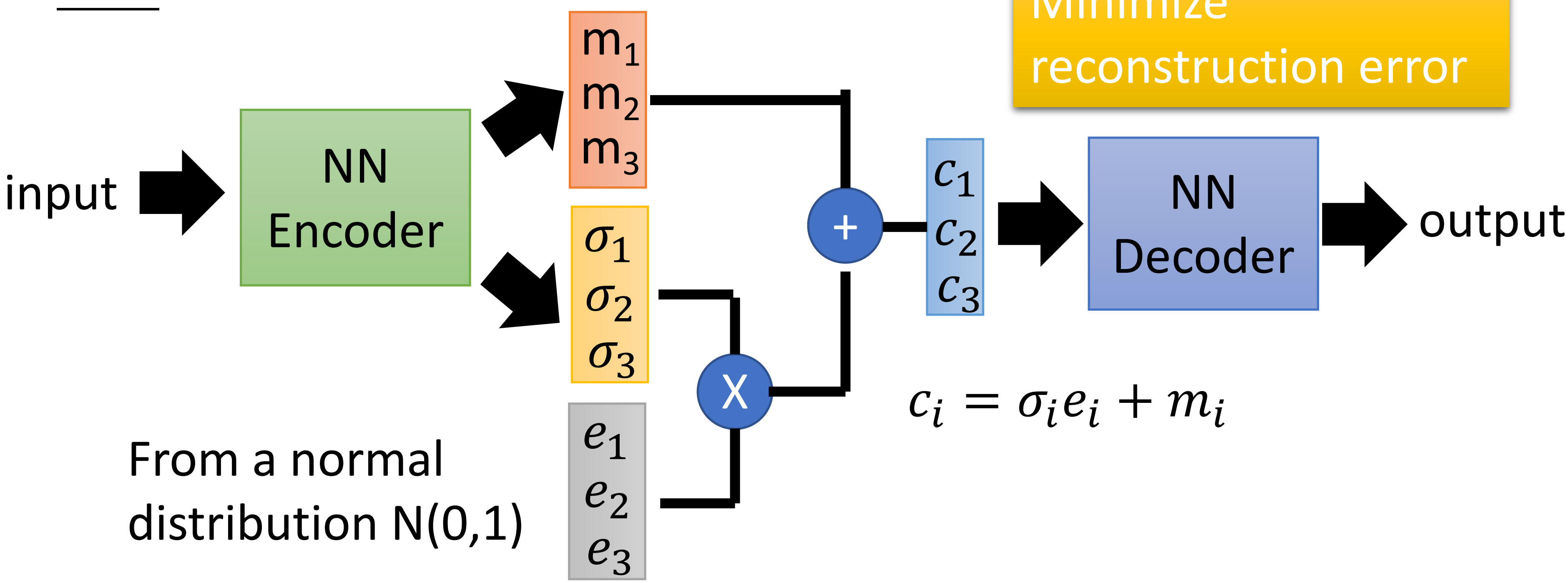
Review: Auto-encoder



Auto-encoder



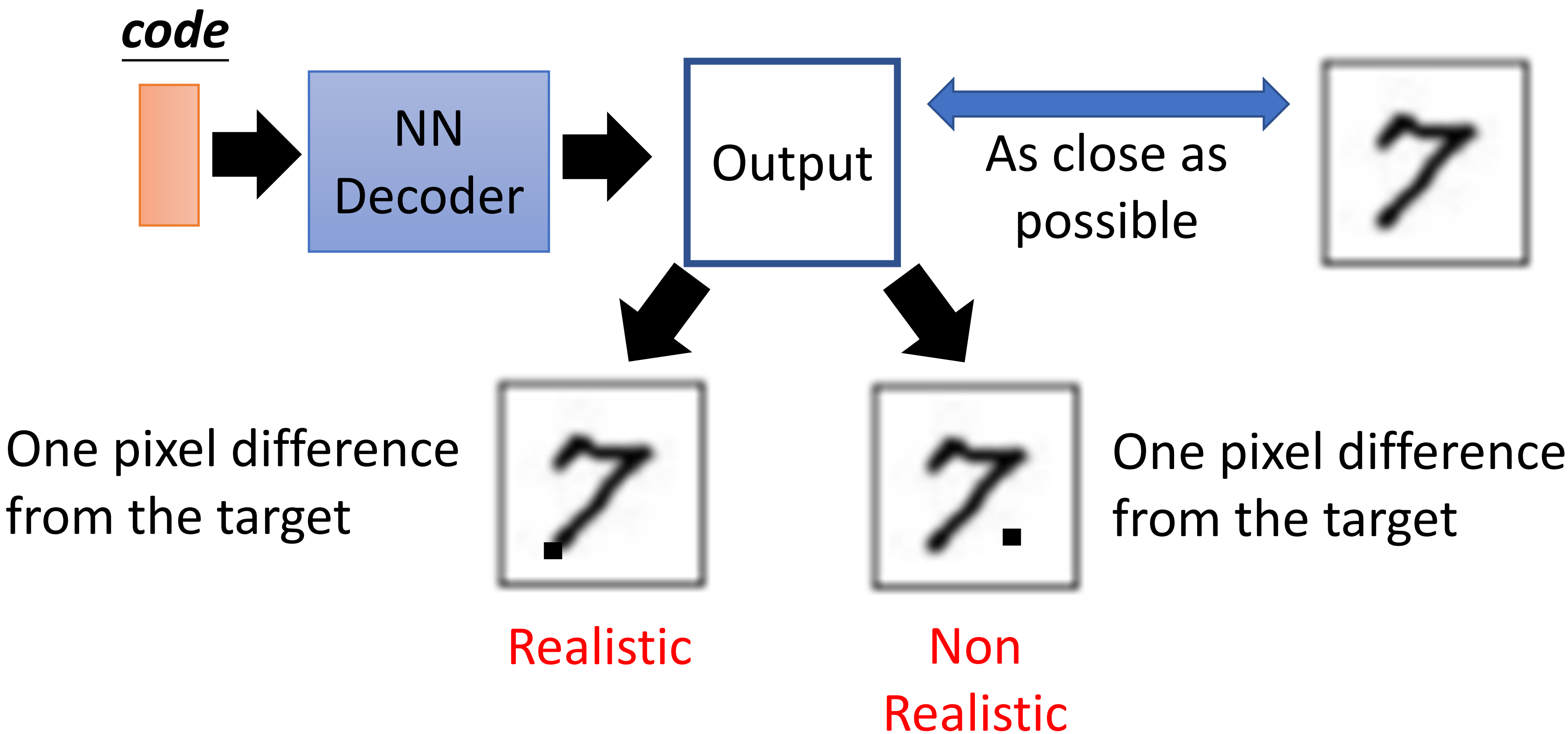
VAE



Auto-Encoding Variational Bayes, https://arxiv.org/abs/1312.6114

Problems of AE/VAE

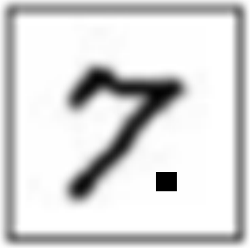
- It does not really try to simulate real images



Problems of AE/VAE

GAN to tackle this pb:

Realistic



Non Realistic

GAN: generative adversarial networks

Game scenario:

Player1, Generator, produces samples

Player2, – Its adversary **Discriminator**, attemps to distinguish real samples from fake generated ones (produced by P1) !

Generative models

Outline

1. Preview: Auto-Encoders, VAE
2. **Generative models**

   Details in course:

   AE notation and optimization

   O: Objective for Gen optimization framework

   01: Maximum Likelihood

   02: MMD Max Mean Discrepancy

   O3: GAN framework, optimization objective function, Algo
3. GAN architectures for image generation

GAN model

GAN – Generative Adversarial Nets
Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.

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 $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
   • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
   • Update the discriminator by ascending its stochastic gradient:

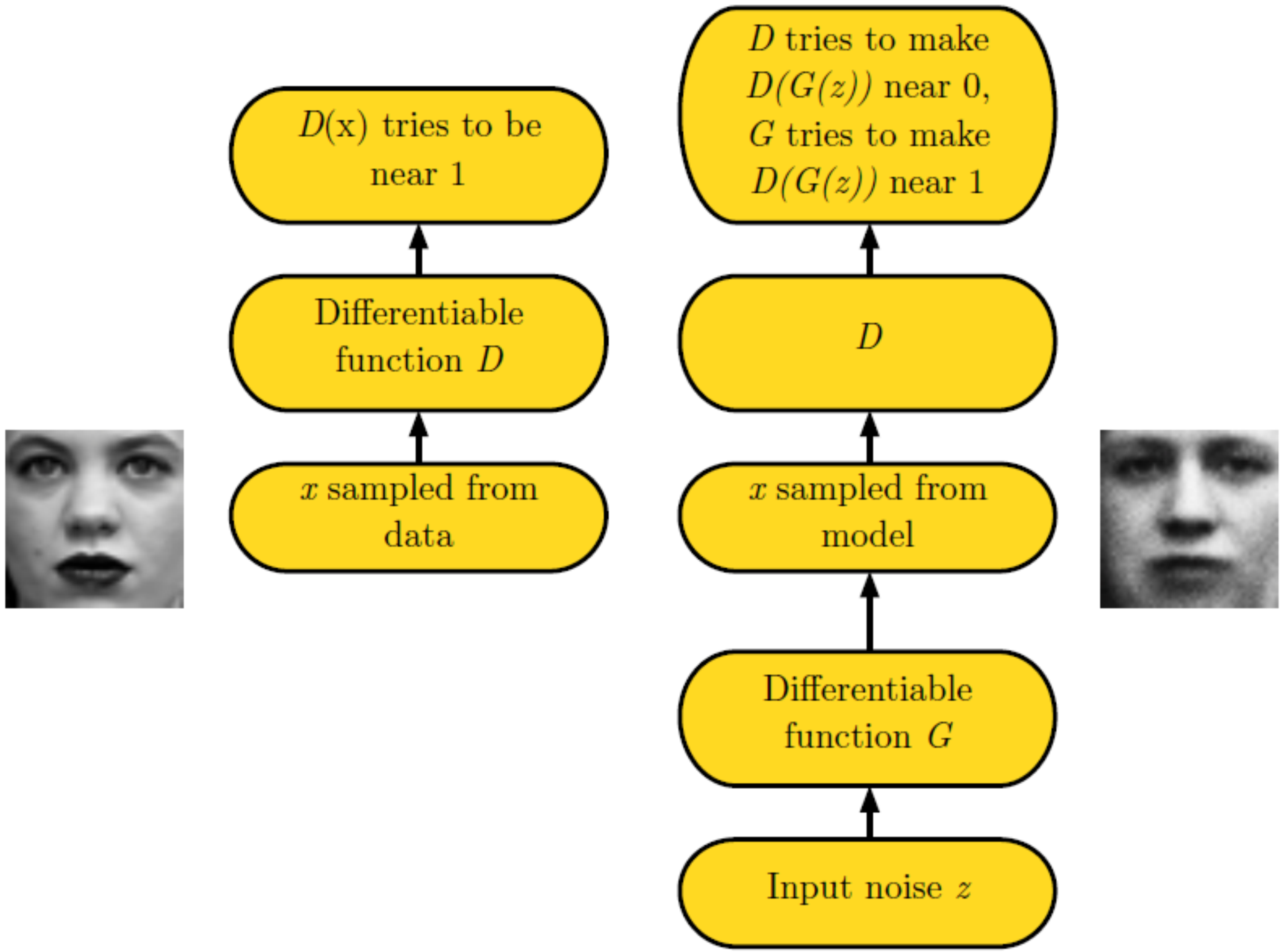
$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by descending its stochastic gradient:

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

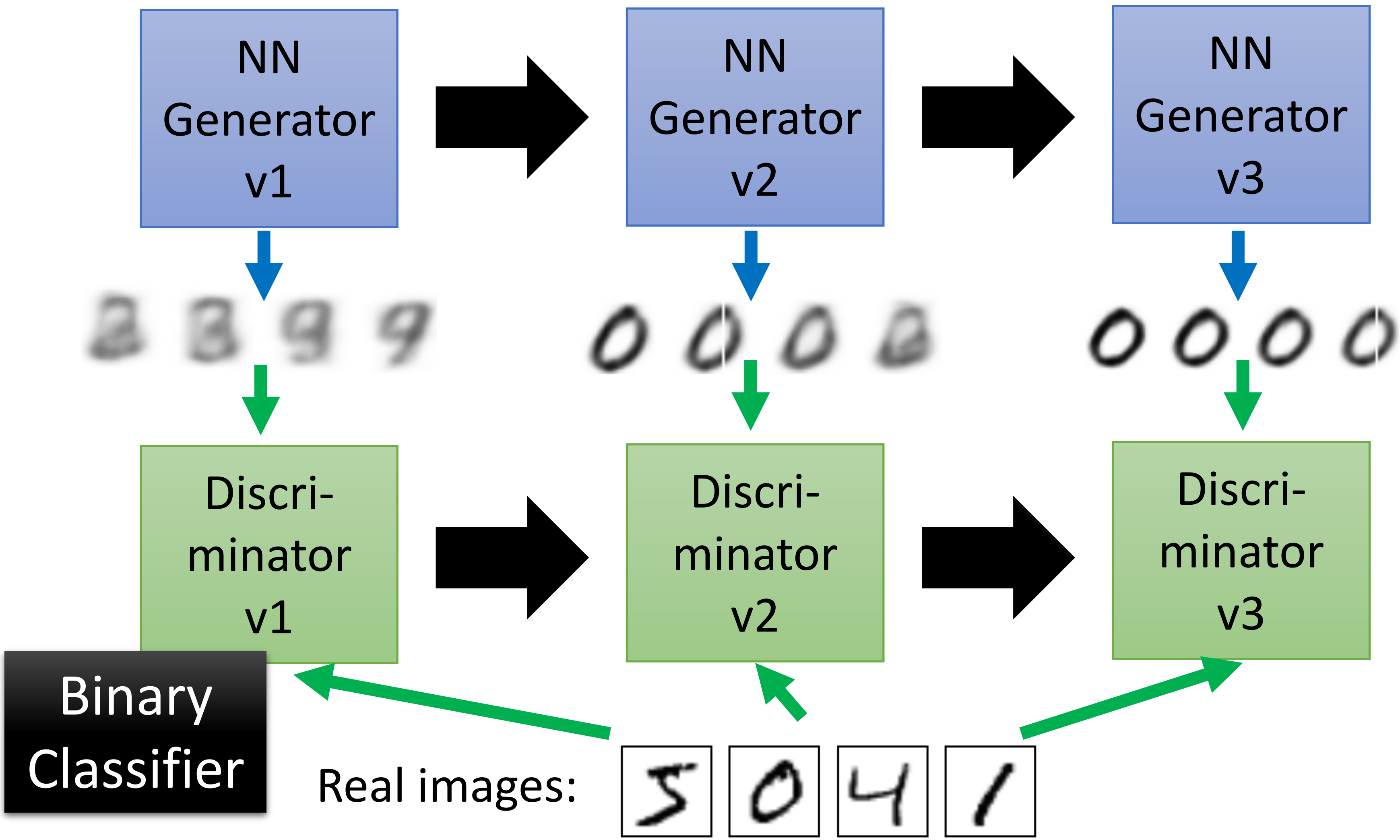
end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Adversarial Nets Framework

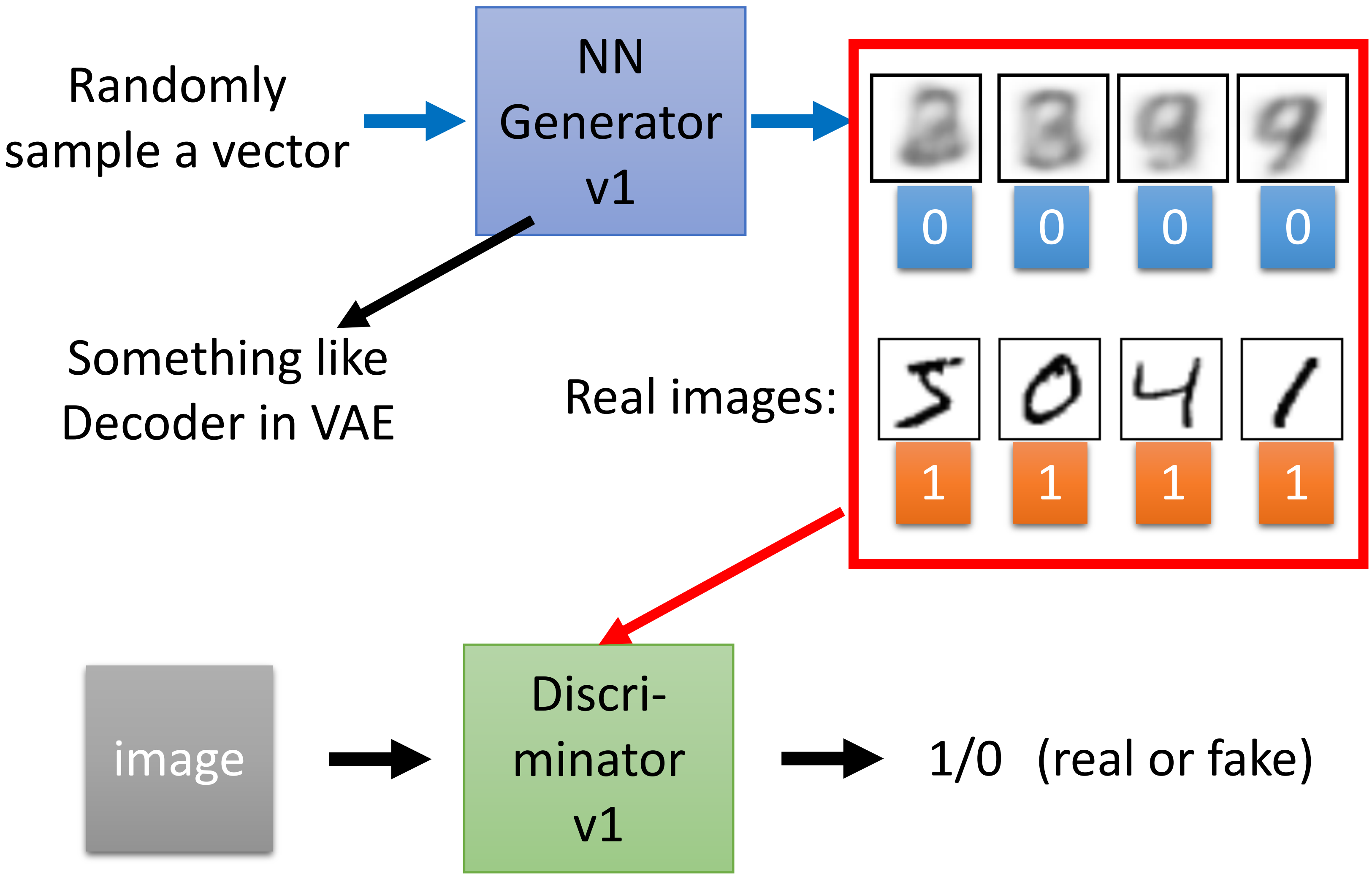


(Goodfellow 2016)

The evolution of generation



GAN - Discriminator



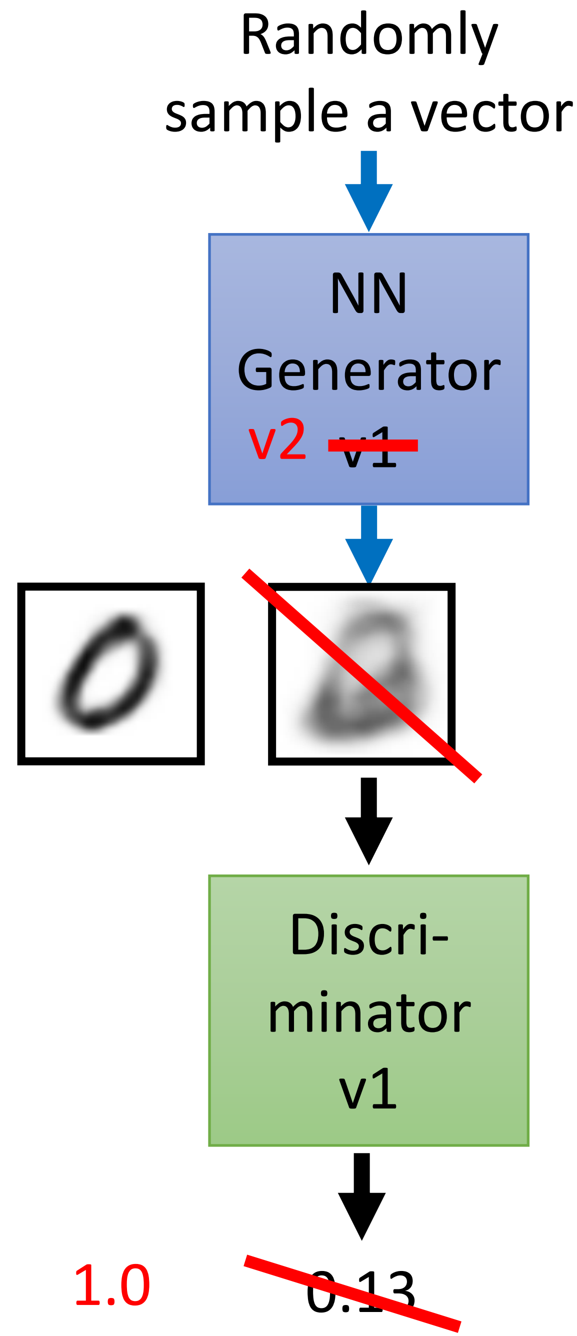
GAN - Generator

Updating the parameters of generator

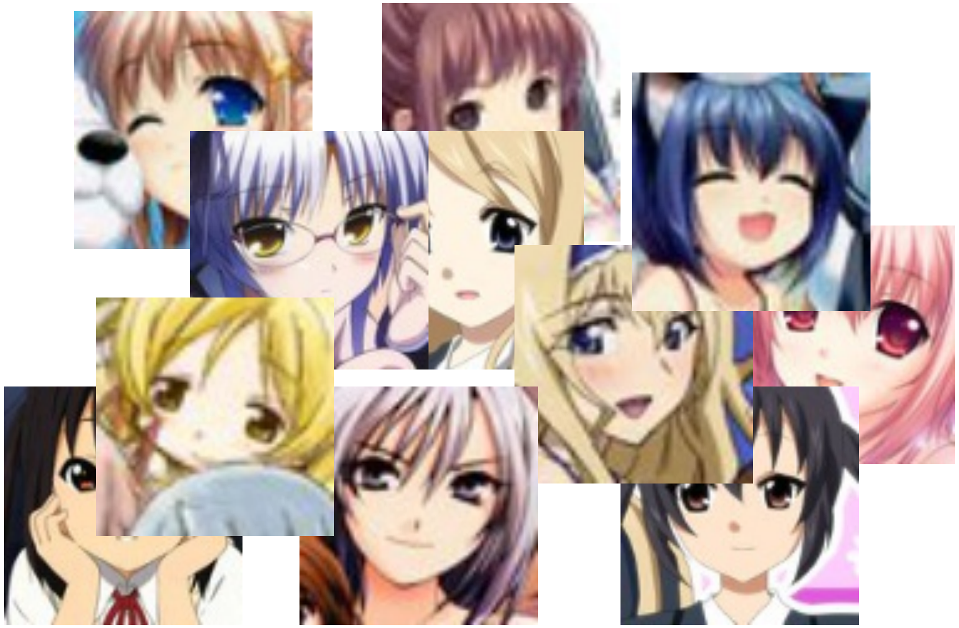
The output be classified as "real" (as close to 1 as possible)

Generator + Discriminator = a network

Using gradient descent to update the parameters in the generator, but fix the discriminator



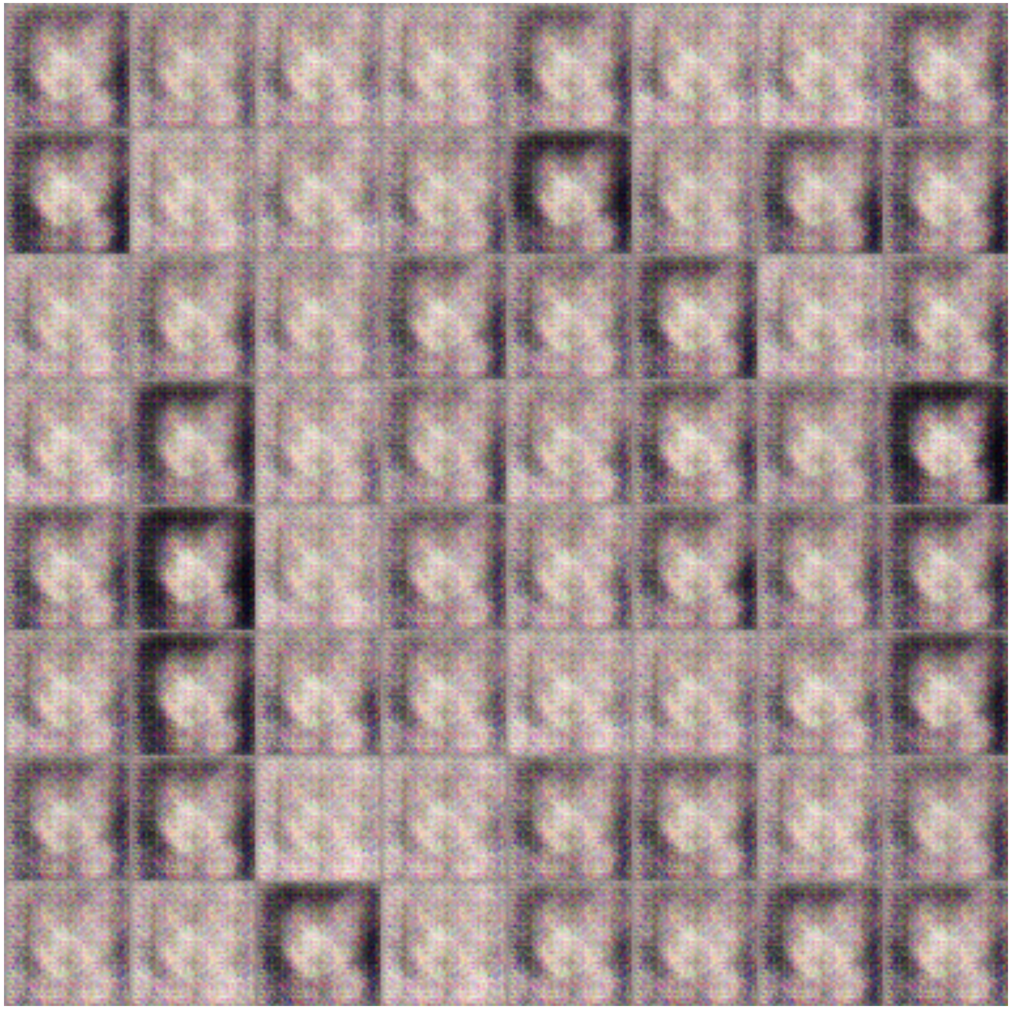
One example GAN



Source of images: https://zhuanlan.zhihu.com/p/24767059

DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

GAN



100 rounds

GAN

1000 rounds

GAN

2000 rounds

GAN

5000 rounds

GAN

10,000 rounds

GAN

20,000 rounds

GAN

50,000 rounds

Generative models

Outline

1. Auto-Encoders, VAE
2. GAN models
3. **GAN architectures for image generation**