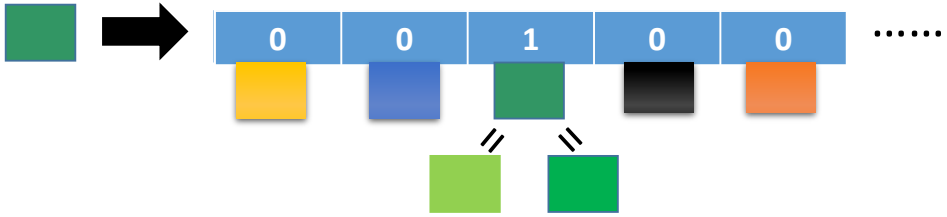
Unsupervised Learning: Generative Models

Component-by-component

Practicing Generation Models: Pokémon Creation

- Tips
  - Each pixel is represented by 3 numbers (corresponding to RGB)
  - Each pixel is represented by a 1-of-N encoding feature

R=50, G=150, B=100



Clustering the similar color → 167 colors in total

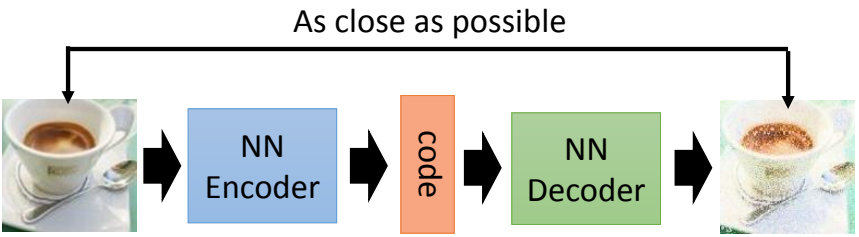
PixelRNN

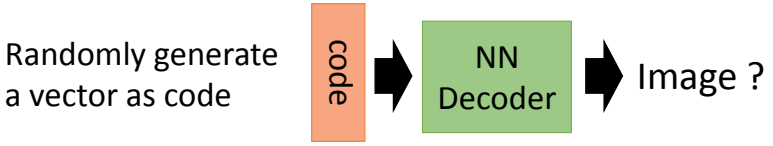
Ref: Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu, Pixel Recurrent Neural Networks, arXiv preprint, 2016

Real World

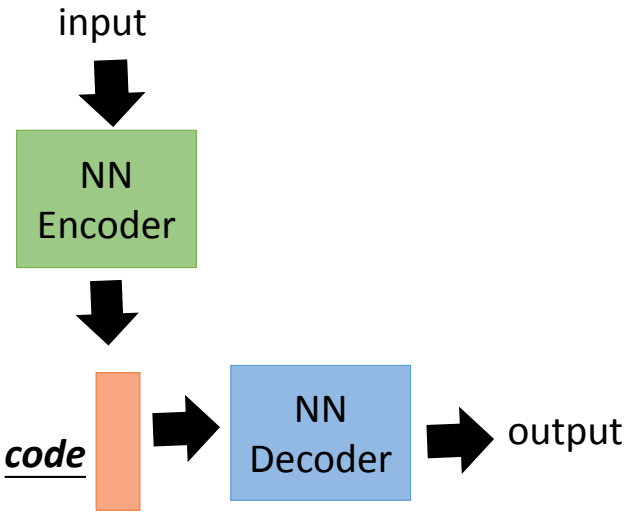
Autoencoder

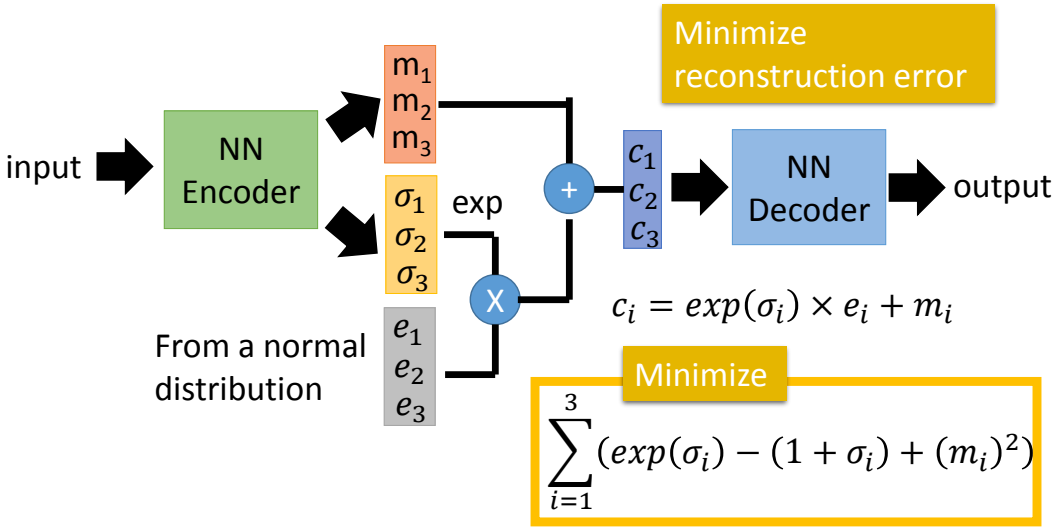
Auto-encoder





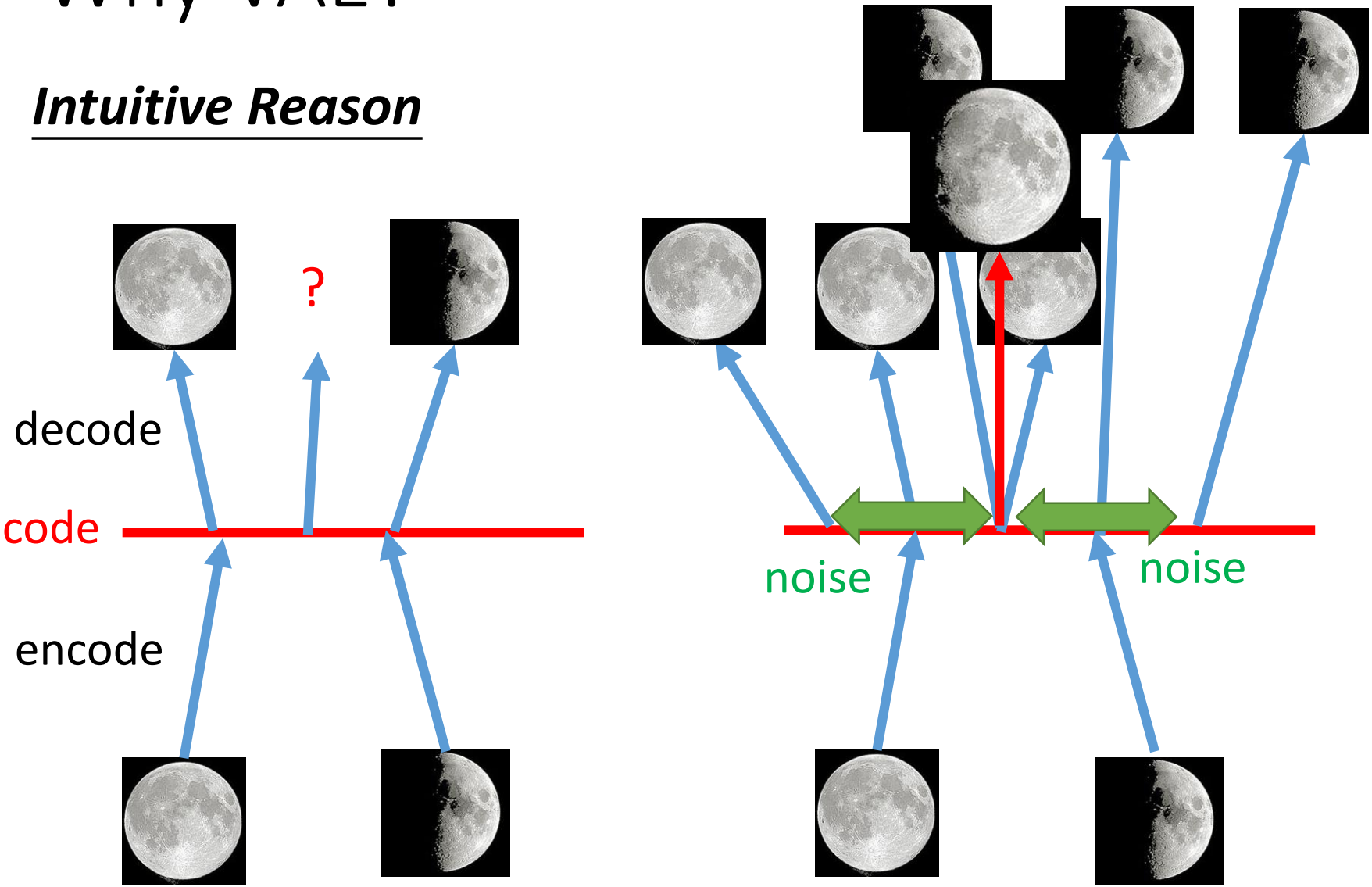
VAE





Why VAE?

Intuitive Reason



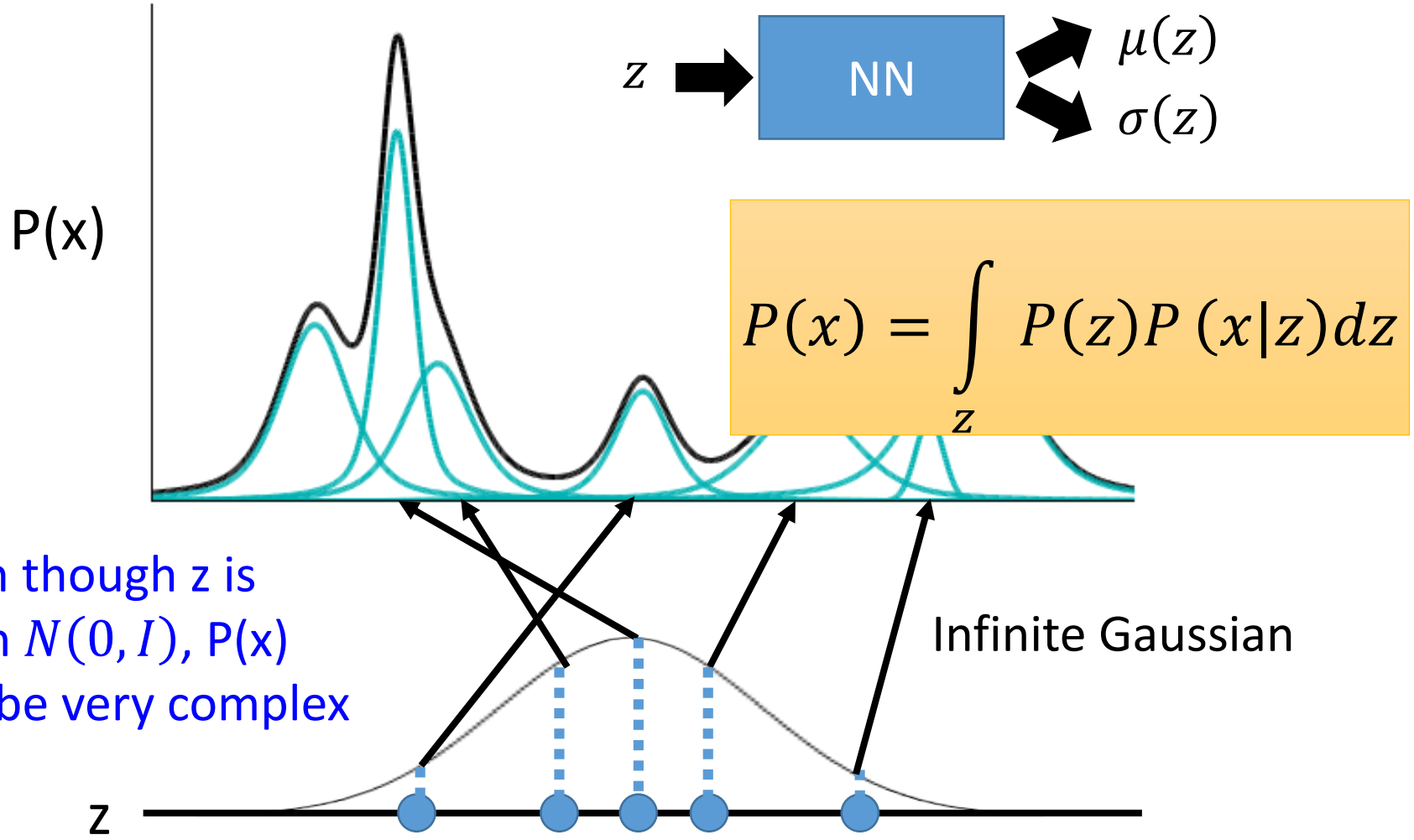
VAE

$z \sim N(0, I)$

$x|z \sim N\big(\mu(z), \sigma(z)\big)$

z is a vector from normal distribution

Each dimension of z represents an attribute



Maximizing Likelihood

$$P(x) = \int_z P(z)P(x|z)dz$$

$$L = \sum_x logP(x)$$

$$logP(x) = \int_z q(z|x)logP(x)dz$$

P(z) is normal distribution

$$x|z \sim N\big(\mu(z), \sigma(z)\big)$$

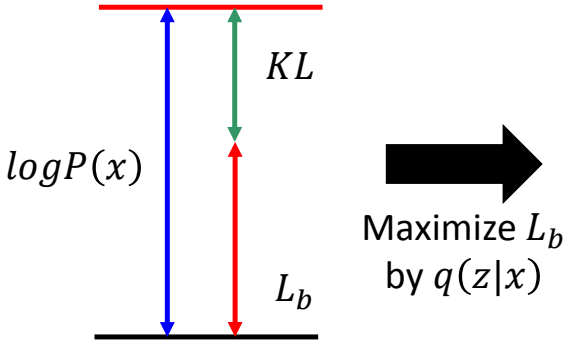
$\mu(z), \sigma(z)$ is going to be estimated

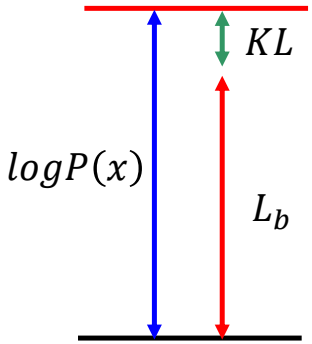
Maximizing the likelihood of the observed x

e any distribution

$$L_b = \int_z q(z|x)log\left(\frac{P(x|z)P(z)}{q(z|x)}\right)dz$$

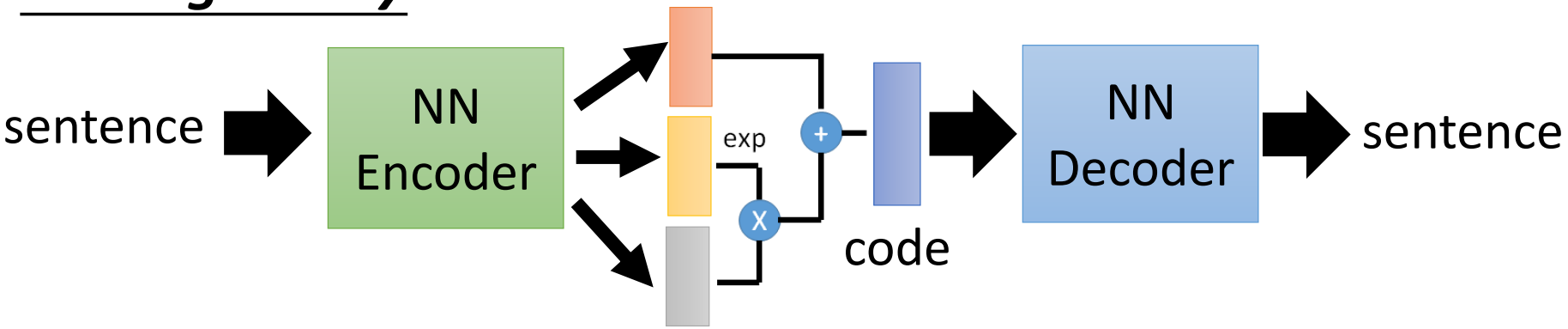
Find $P(x|z)$ and $q(z|x)$ maximizing L_b



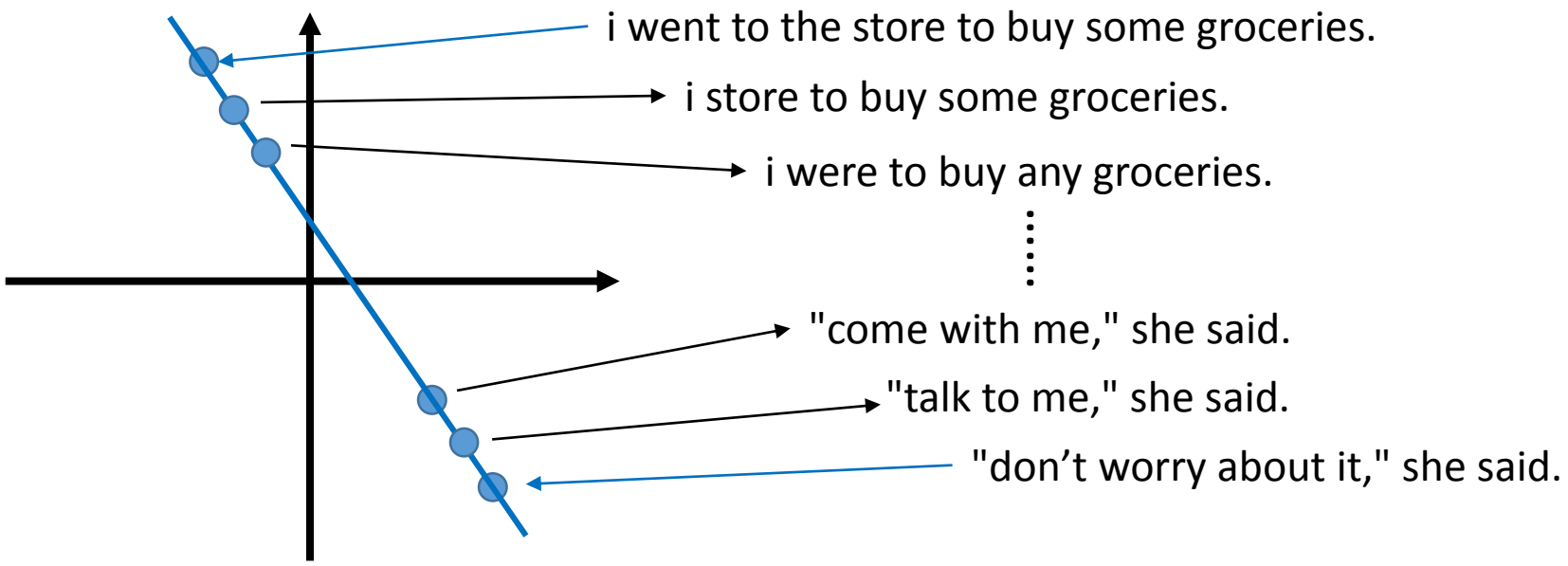


$q(z|x)$ will be an approximation of $p(z|x)$ in the end

Writing Poetry



Code Space

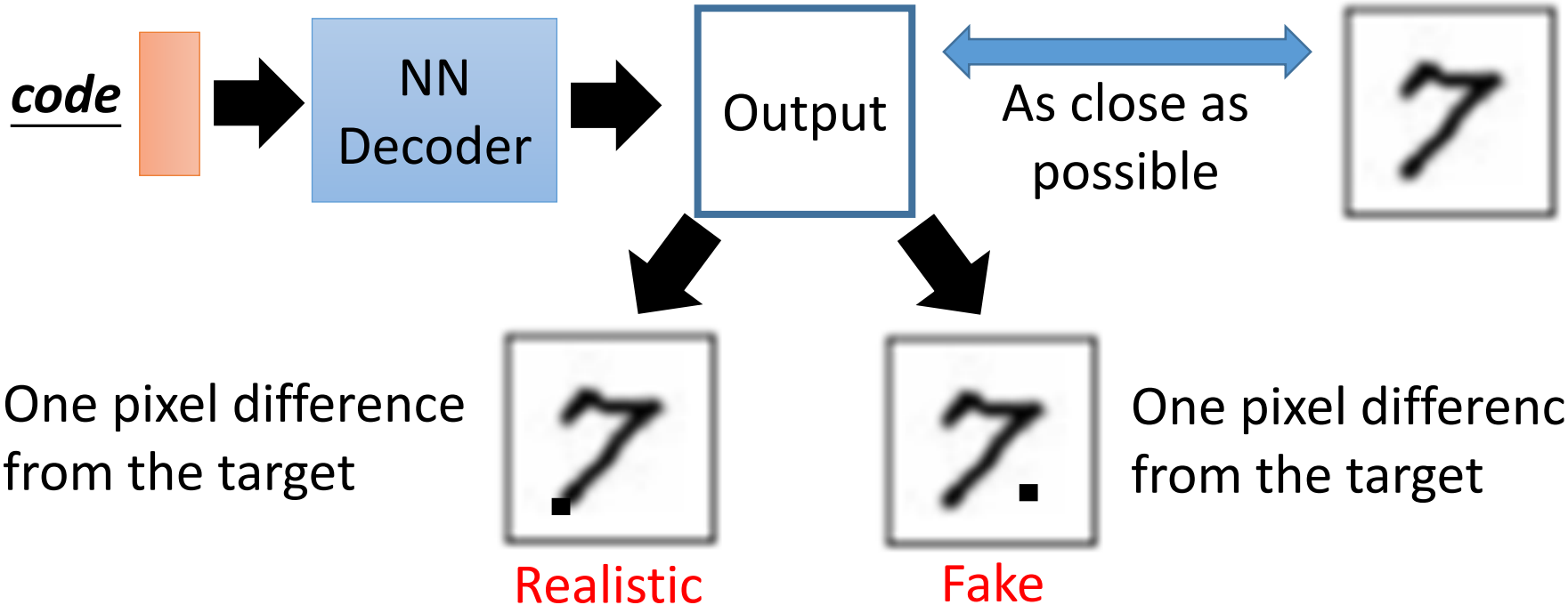


Ref: http://www.wired.co.uk/article/google-artificial-intelligence-poetry

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, Samy Bengio, Generating Sentences from a Continuous Space, arXiv prepring, 2015

Problems of VAE

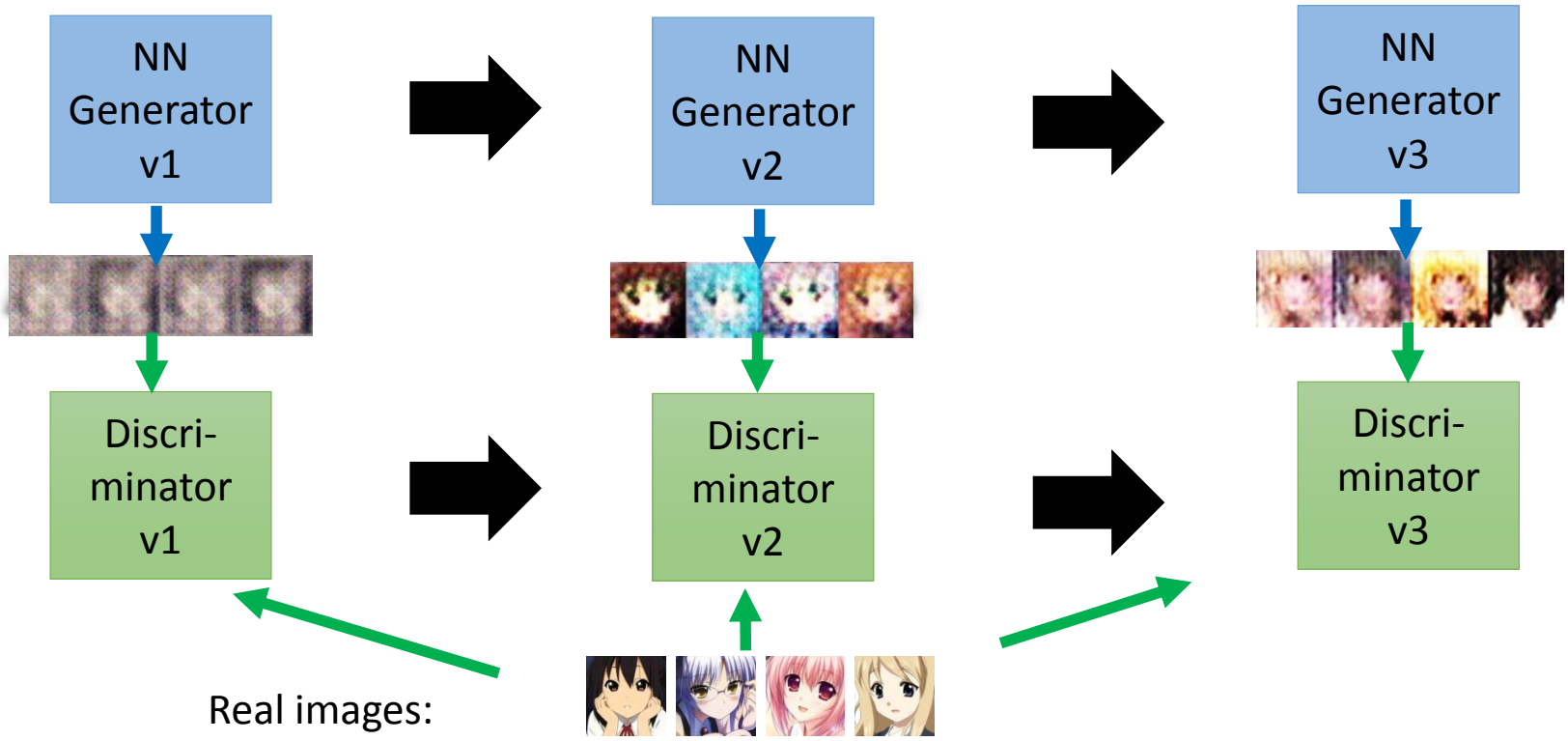
- It does not really try to simulate real images



VAE may just memorize the existing images, instead of generating new images

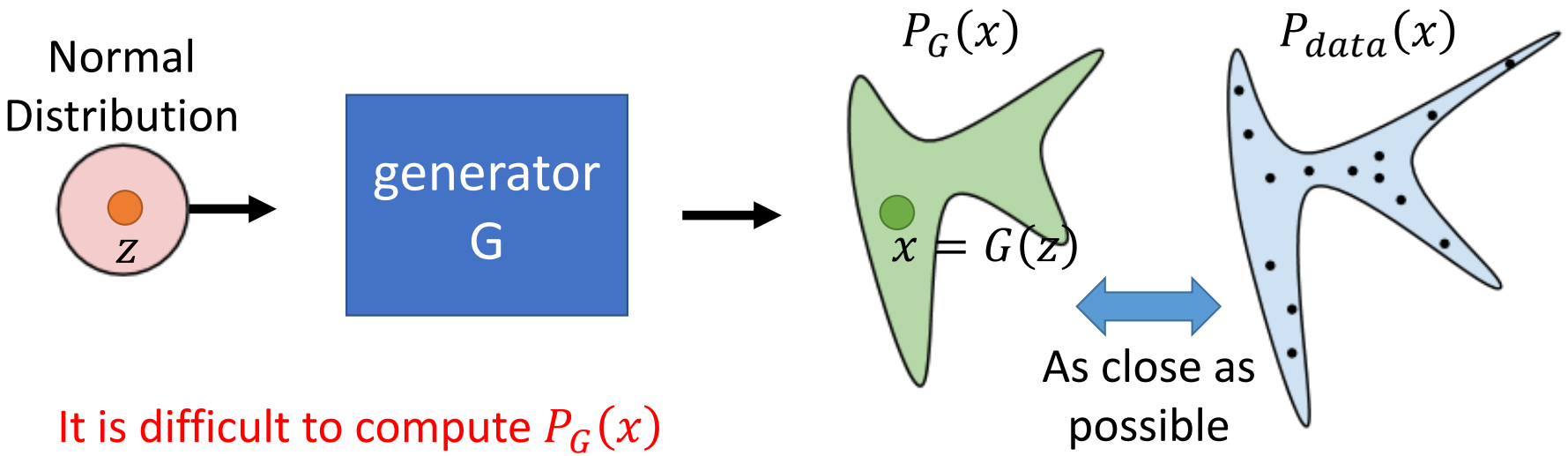
Generative Adversarial Network (GAN)

The evolution of generation



Basic Idea of GAN

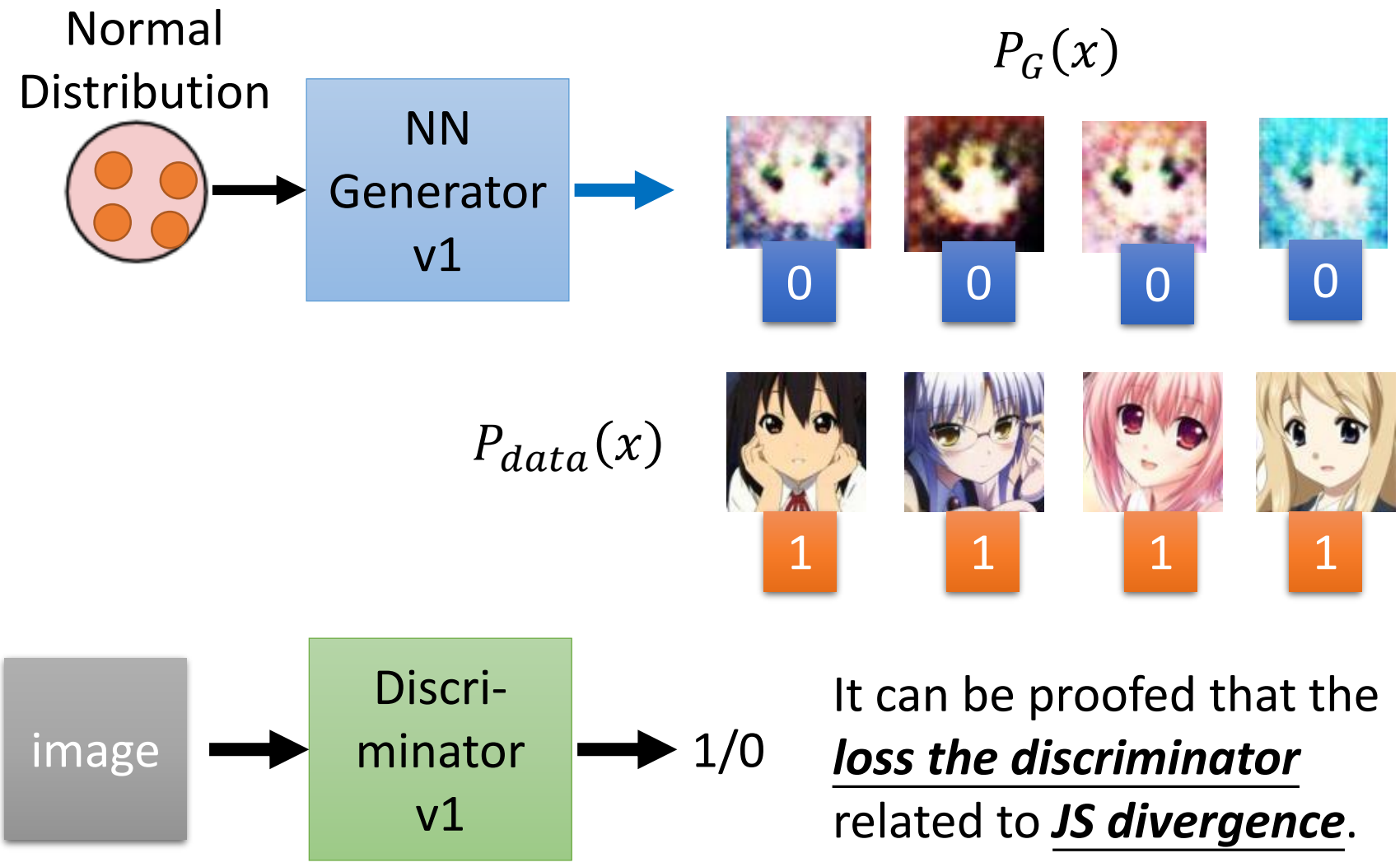
- A generator G is a network. The network defines a probability distribution.
- The data we want to generate has a distribution $P_{data}(x)$



It is difficult to compute $P_G(x)$

We do not know what the distribution looks like.

Basic Idea of GAN



It can be proofed that the ***loss the discriminator*** related to ***JS divergence***.

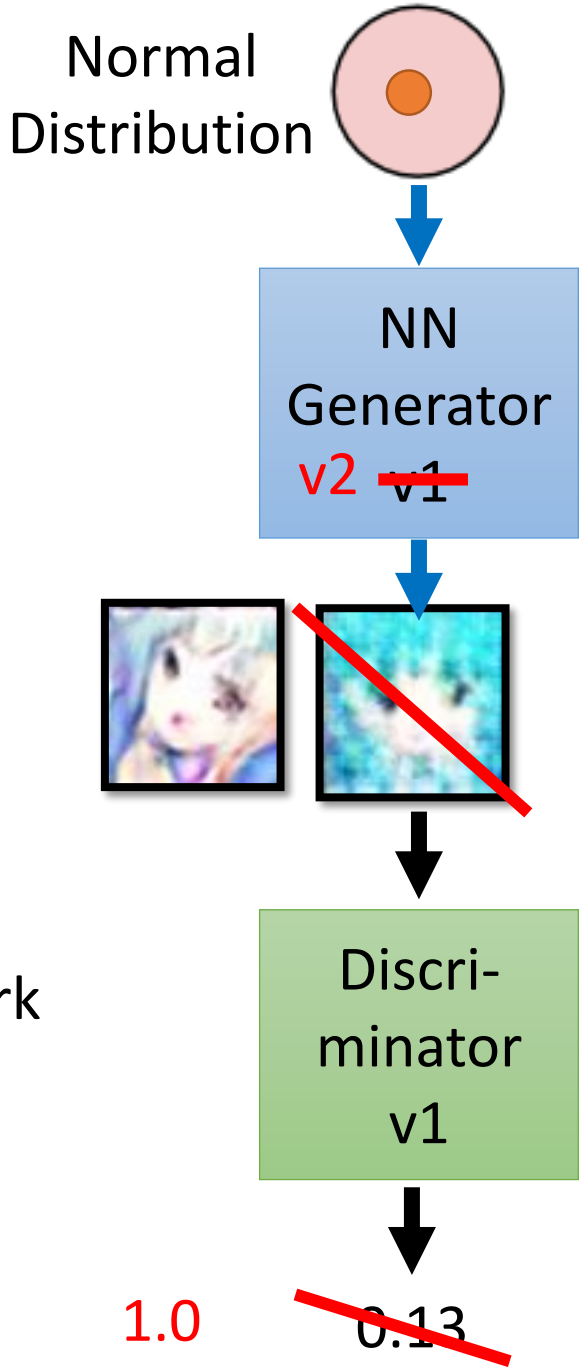
Basic Idea of GAN

- ***Next step***:
  - Updating the parameters of generator
  - To minimize the JS divergence

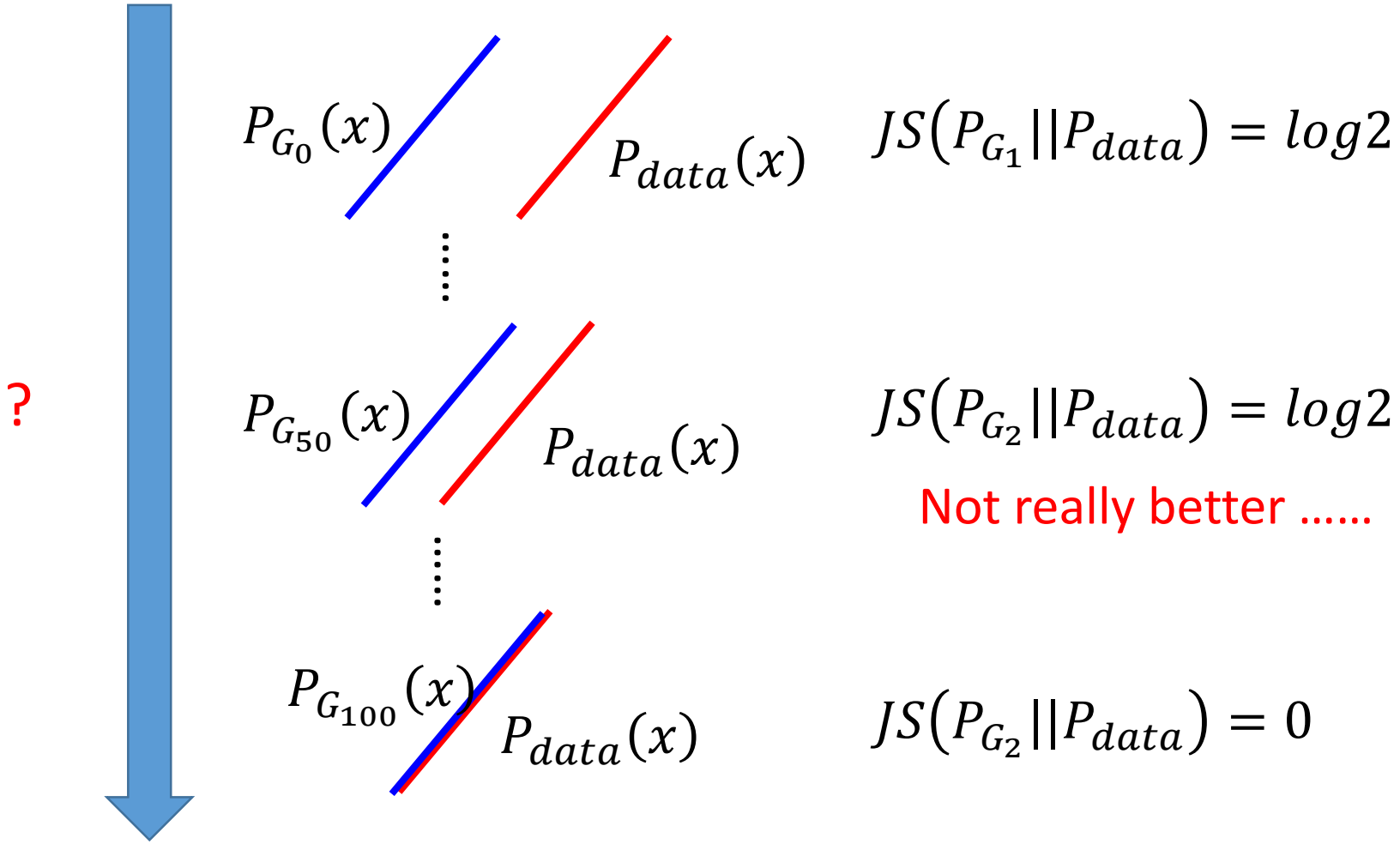
➡ The output be classified as "real" (as close to 1 as possible)

Generator + Discriminator = a network

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



Why GAN is hard to train?



WGAN

Using Wasserstein distance instead of JS divergence