

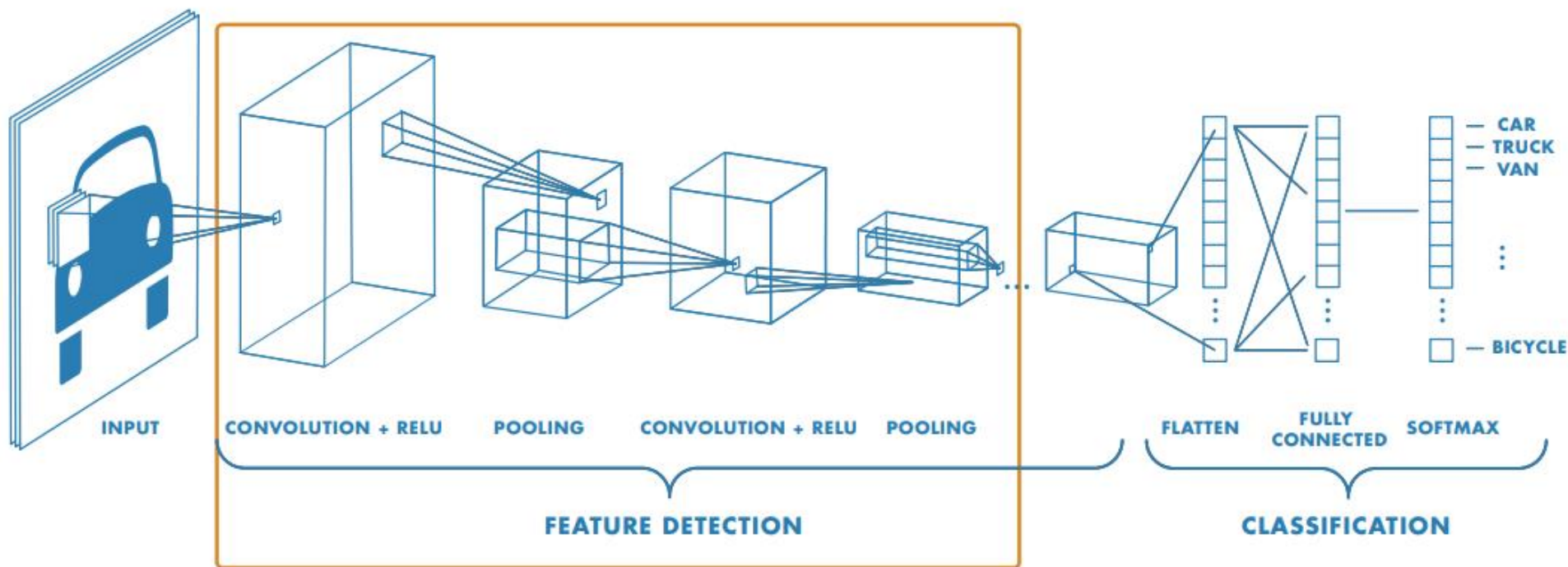
DL: Сверточные сети

Приложения

План

- Сиамские сети
- Автоэнкодер
- GAN

Передача обучения



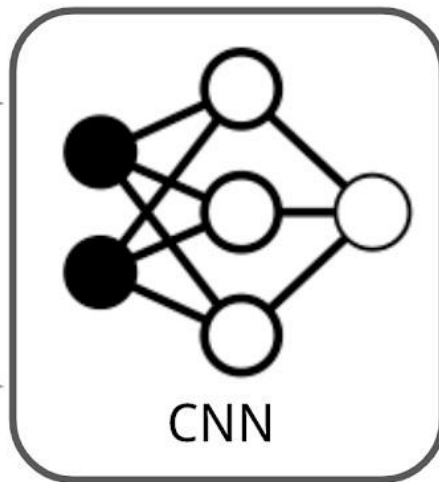
Сиамские сети



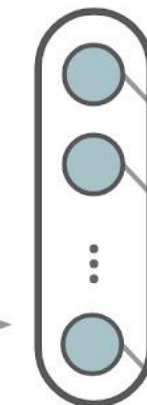
input 1



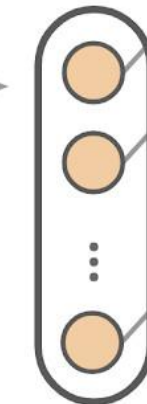
input 2



CNN



embed 1



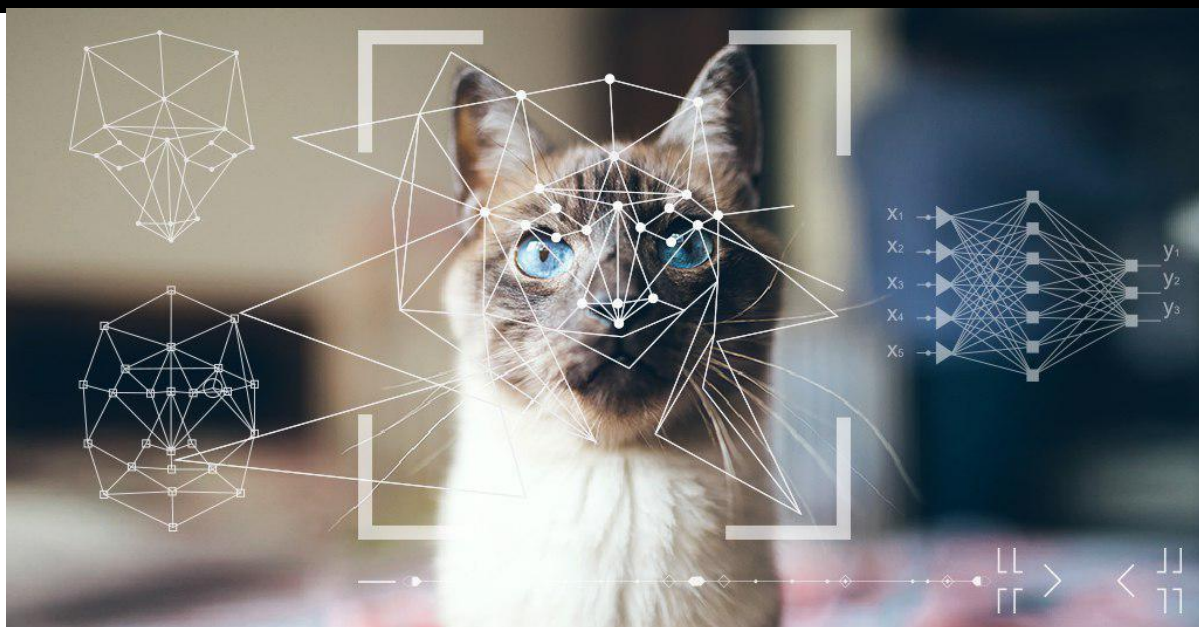
embed 2



distance

probability
of input 1 & 2 are
in the same class

Сиамские сети

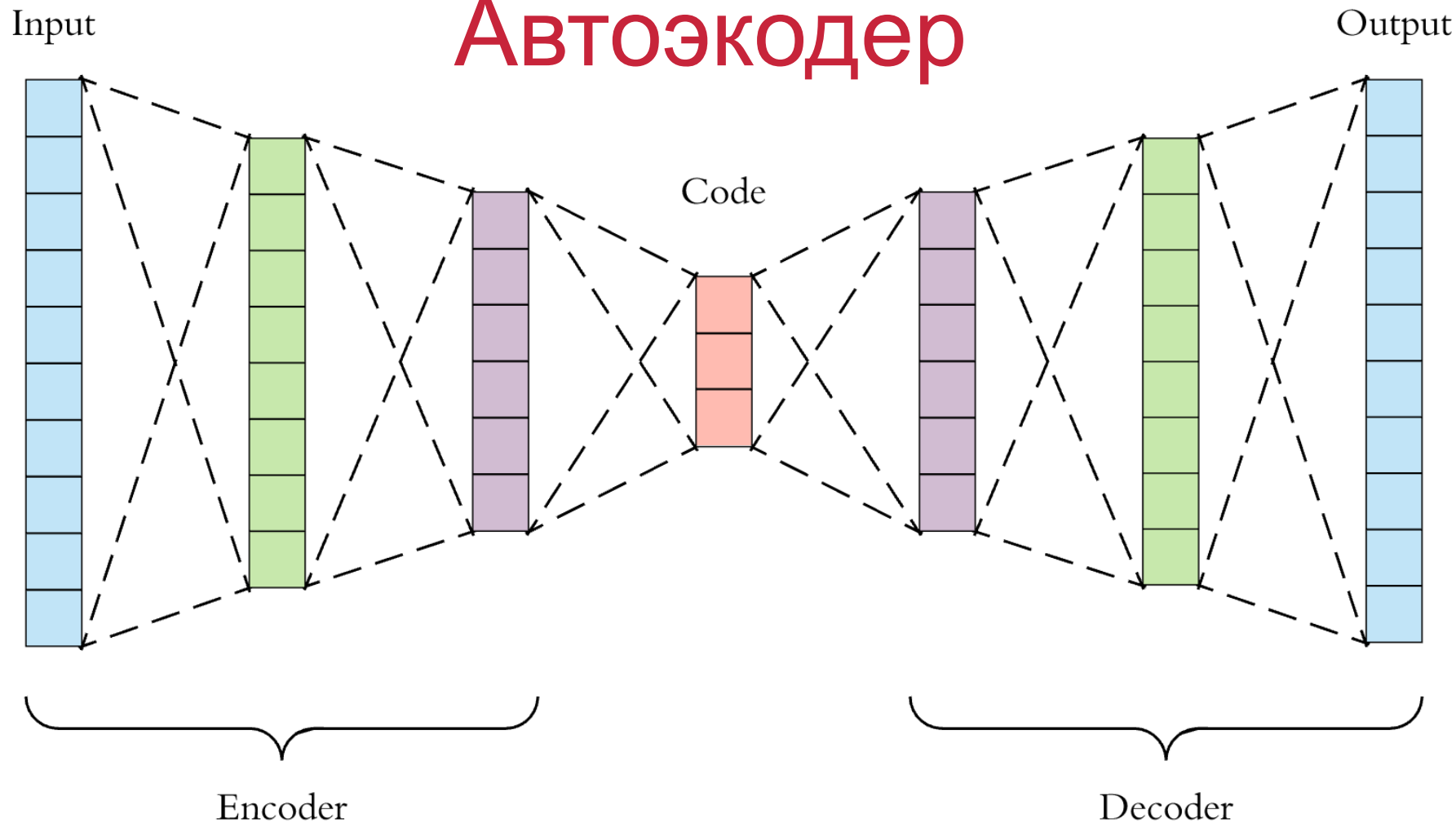


$$L = \max(\|F(A) - F(P)\|^2 - \|F(A) - F(N)\|^2 + \alpha, 0)$$

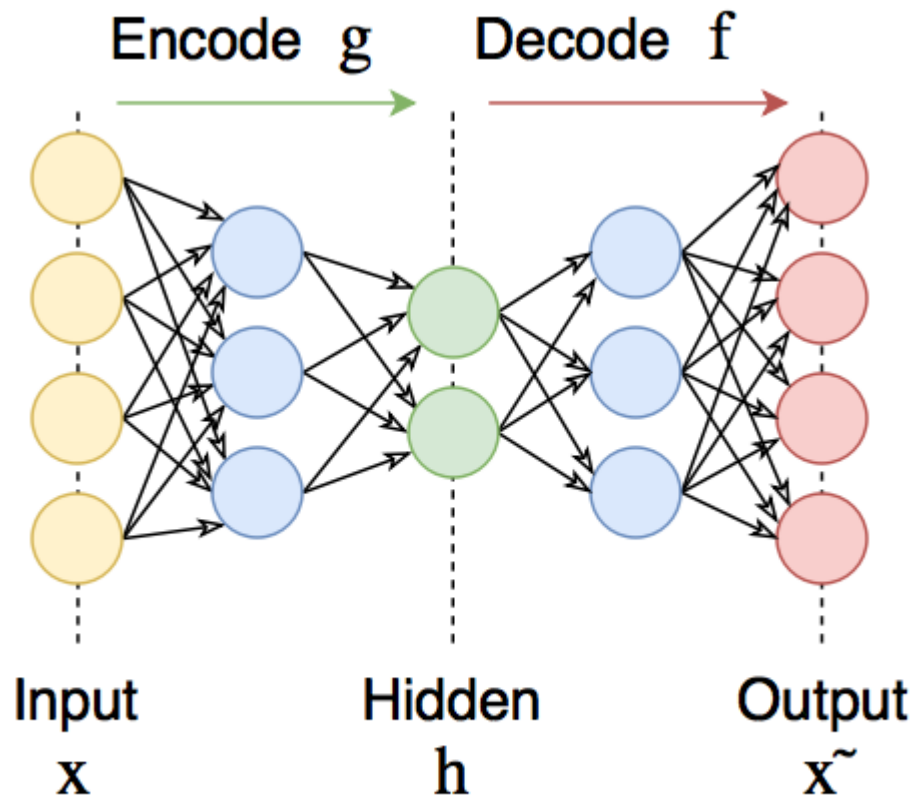
$$L = YD^2 + (1 - Y)\max(\alpha - D, 0)^2$$

$Y = 1$ для изображений одного класса, $Y = 0$ для изображений разного класса

АВТОЭКОДЕР



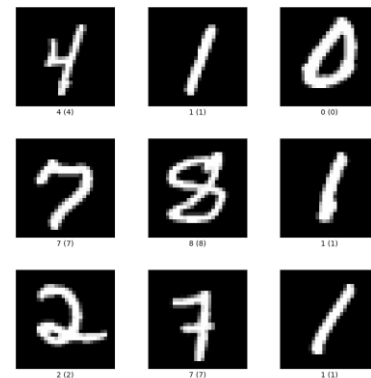
Автоэкодер



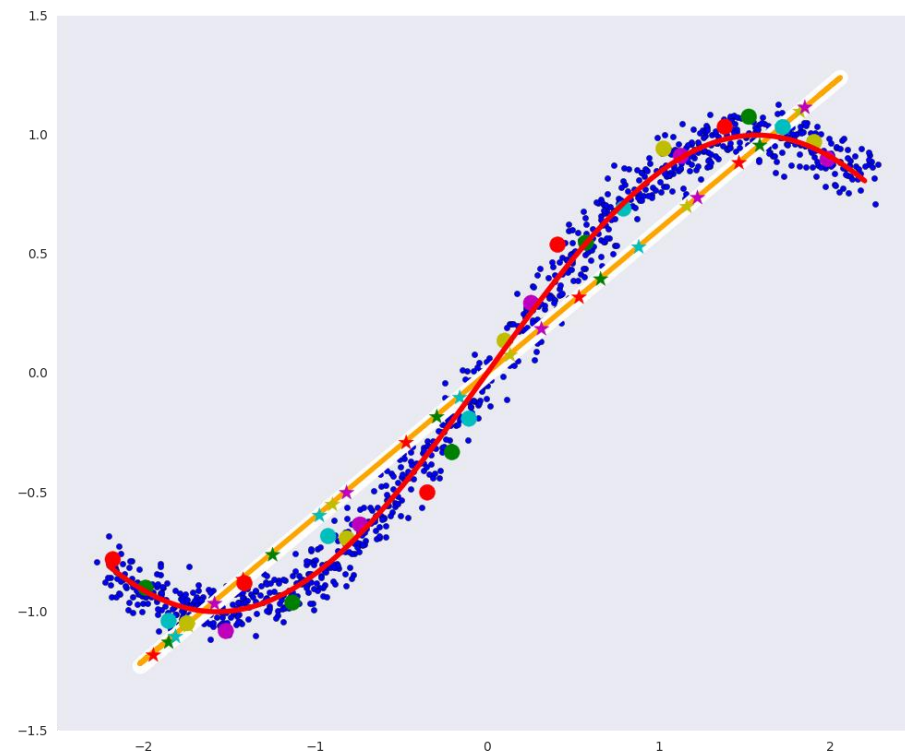
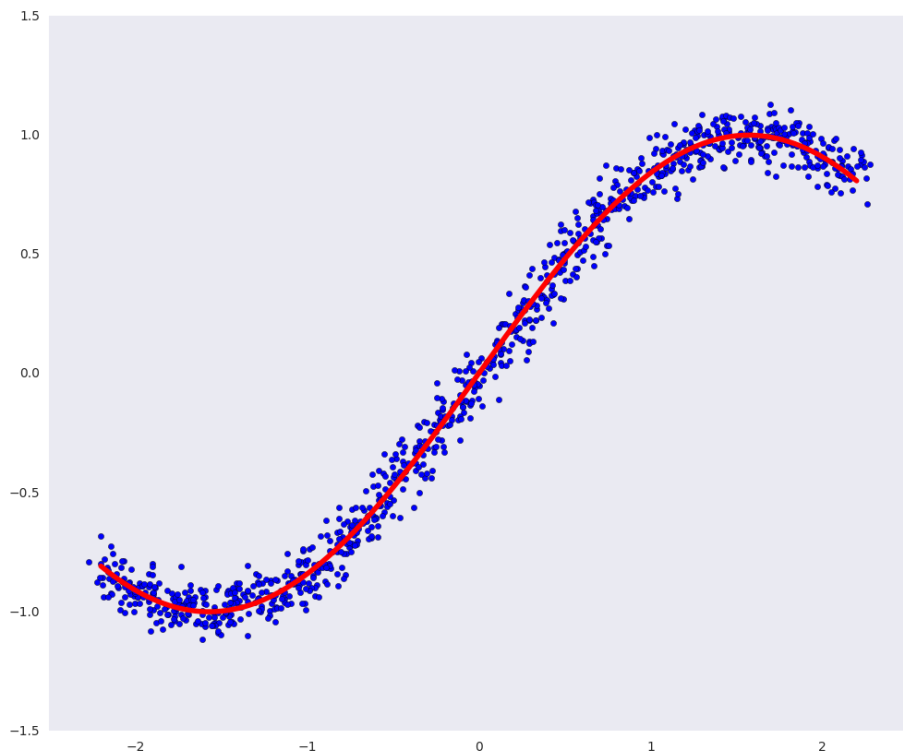
$$h = g(x)$$

$$\tilde{x} = f(h)$$

$$E = \|x - g(f(x))\|$$

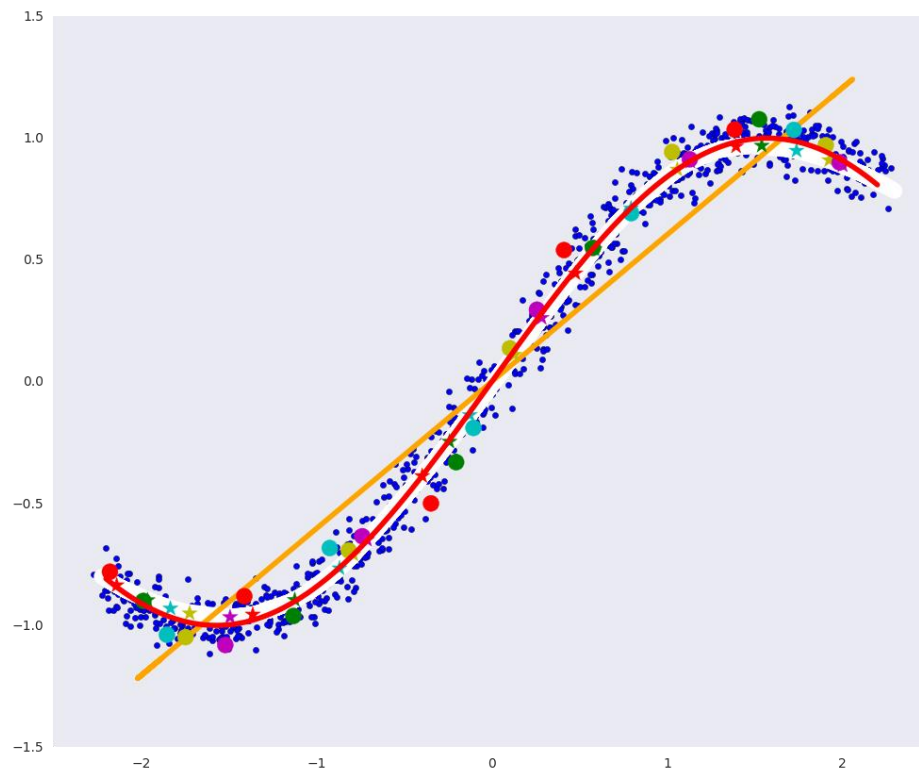
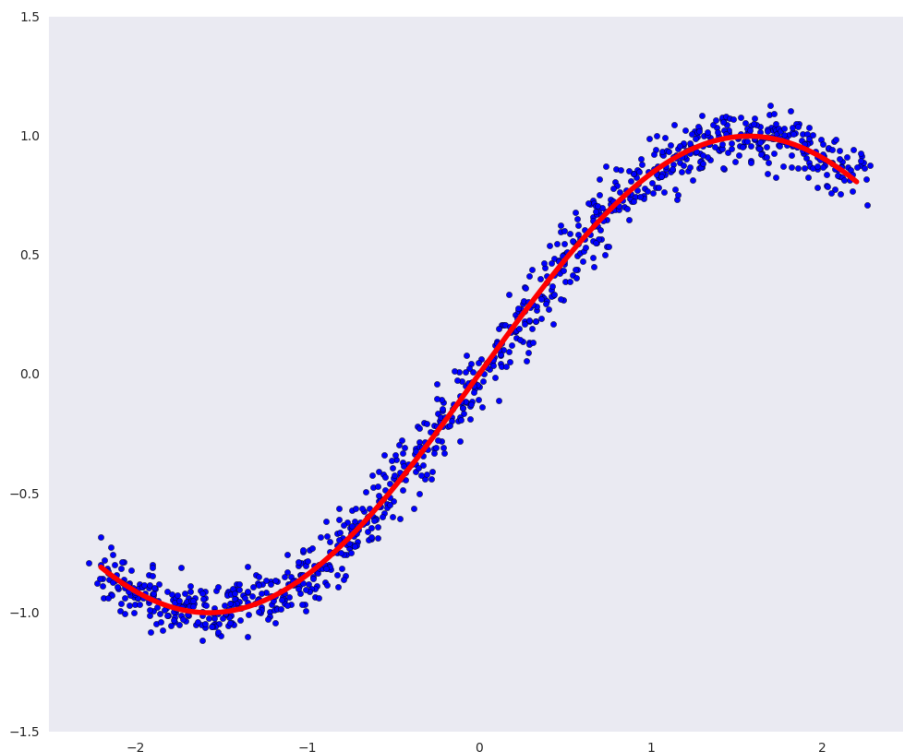


Автоэкодер: manifold learning



<https://habr.com/ru/post/331382/>

Автоэкодер: manifold learning

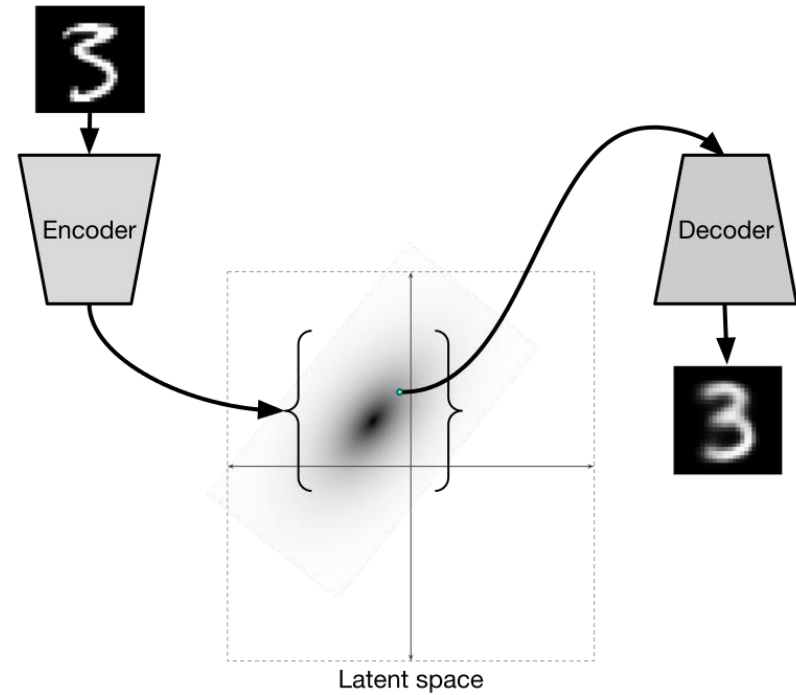
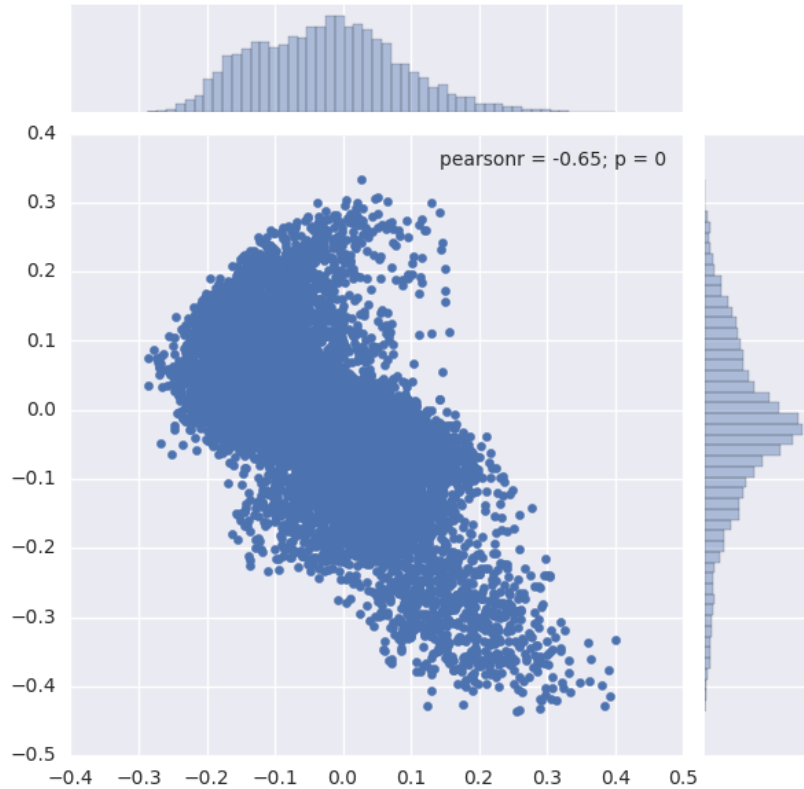


<https://habr.com/ru/post/331382/>

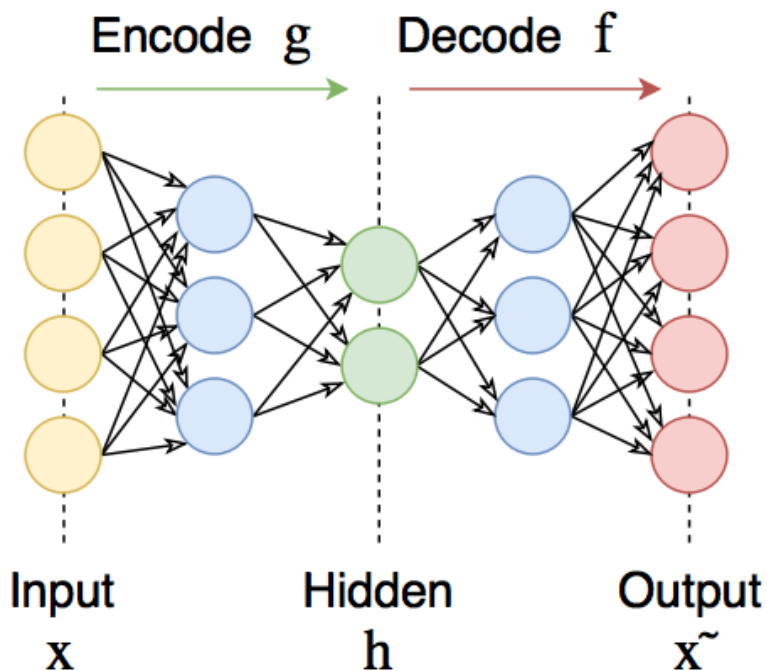
Автоэкодер: manifold learning



VAE: Вариационный автоэнкодер



VAE: Вариационный автоэнкодер



$$P(X) = \int_H P(X|h)P(h)dh \quad P(X|h) = f(h) + \varepsilon$$

$$P(X; \theta) = \int_H P(X|h; \theta)P(h)dh \quad P(X|h; \theta) = f(h; \theta) + \varepsilon$$

$$P(X|h; \theta) = N(X|f(h; \theta), \sigma^2 I)$$

VAE: Вариационный автоэнкодер

$$P(X) = \int_H P(X|h)P(h)dh \qquad P(X|h; \theta) = N(X|f(h; \theta), \sigma^2 I)$$

Выберем подмножество $H' \in H$ из которого мы получаем множество X

Введем распределение $Q(H|X)$ которое даст нам те $H \sim Q$, которые привели к X

$$KL[Q(H|X)||P(H, X)] = E_{H \sim Q}[\log Q(H|X) - \log P(H|X)]$$

$$KL[Q(H|X)||P(H, X)] = E_{H \sim Q}[\log Q(H|X) - \log P(X|H) - \log P(H)] + \log P(X)$$

$$KL[Q(H|X)||P(H, X)] = KL[Q(H|X)||P(H)] - E_{H \sim Q}[\log P(X|H)] + \log P(X)$$

$$\log P(X) - KL[Q(H|X)||P(H, X)] = E_{H \sim Q}[\log P(X|H)] - KL[Q(H|X)||P(H)]$$

VAE: Вариационный автоэнкодер

$$P(X) = \int_H P(X|h)P(h)dh$$

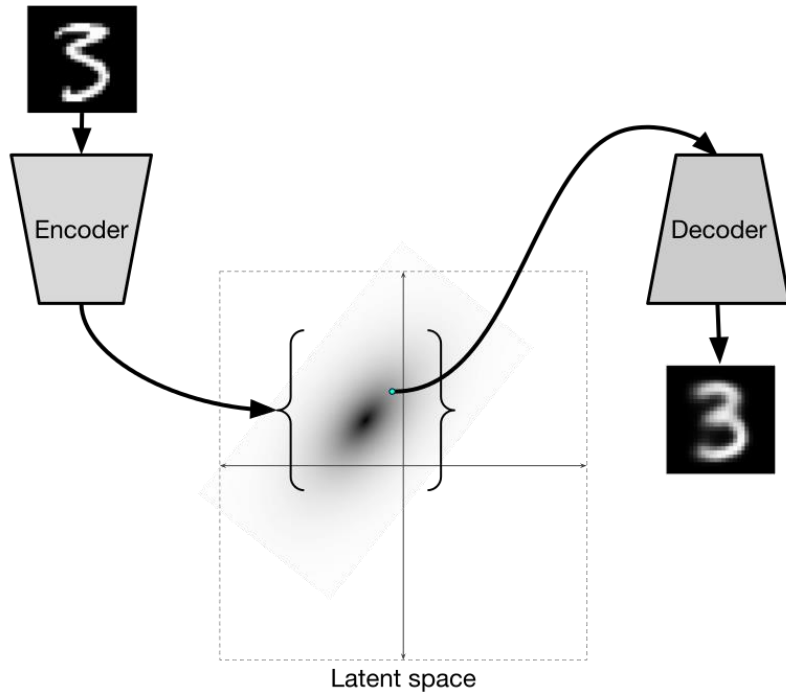
$$\log P(X) - KL[Q(H|X)||P(H,X)] = E_{H \sim Q}[\log P(X|H)] - KL[Q(H|X)||P(H)]$$

$$\log P(X|\theta_2) - KL[Q(H|X; \theta_1)||P(H,X; \theta_2)] = E_{H \sim Q}[\log P(X|H; \theta_2)] - KL[Q(H|X; \theta_1)||N(0, I)]$$

$$Q(H|X; \theta_1) - ?$$

VAE: Вариационный автоэнкодер

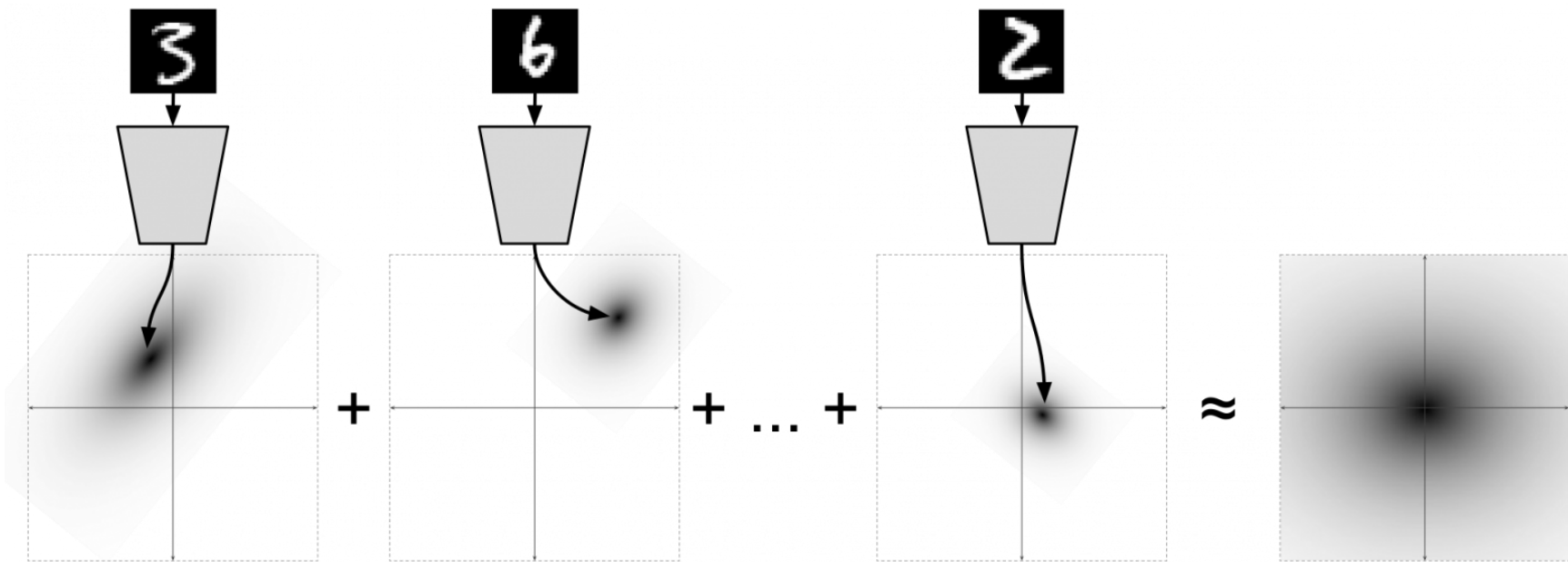
$$Q(H|X; \theta_1) = N(\mu(X; \theta_1), \Sigma(X; \theta_1))$$



VAE: Вариационный автоэнкодер

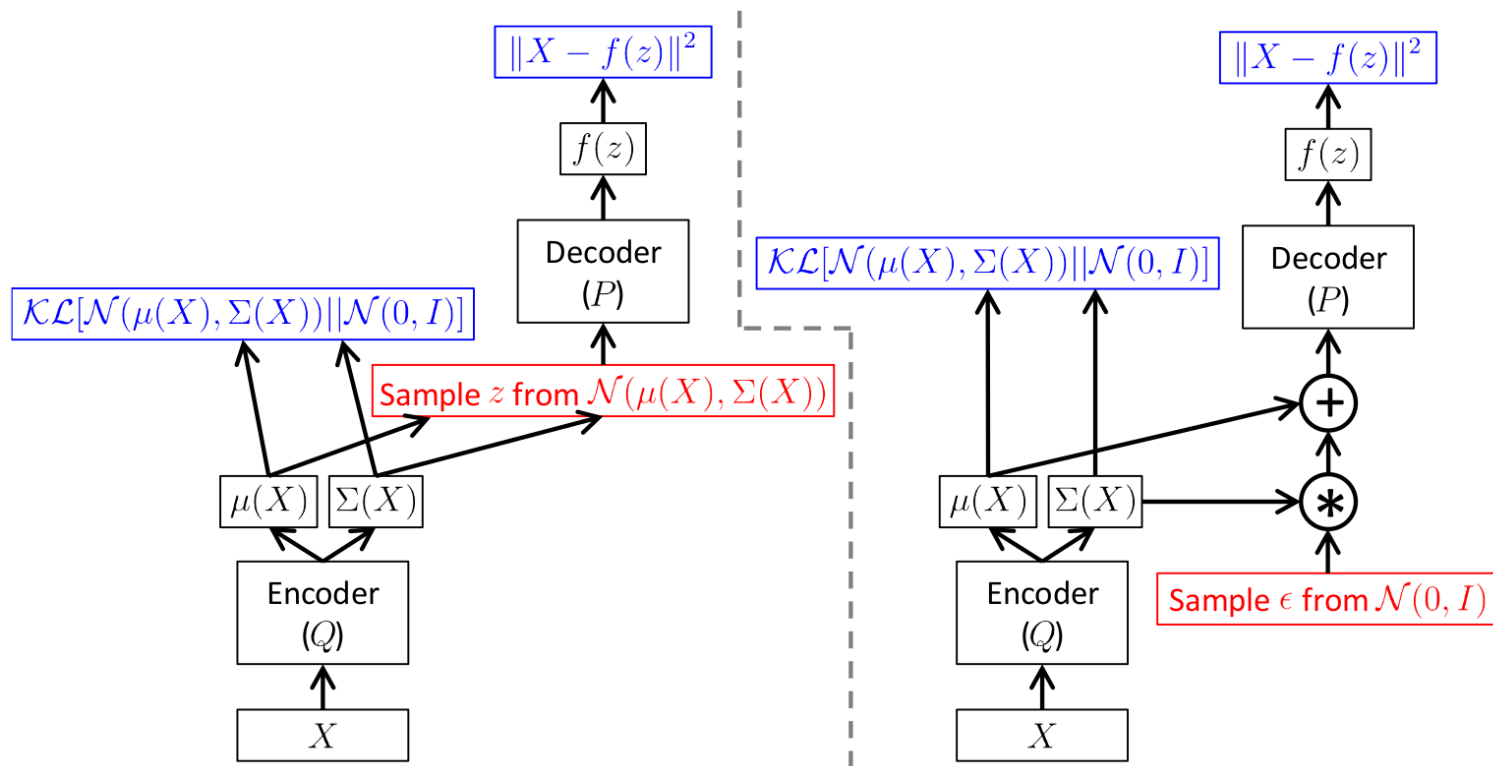
$$P(H|X) = N(\mu(X), \Sigma(X))$$

$$P(H) = N(0, I)$$

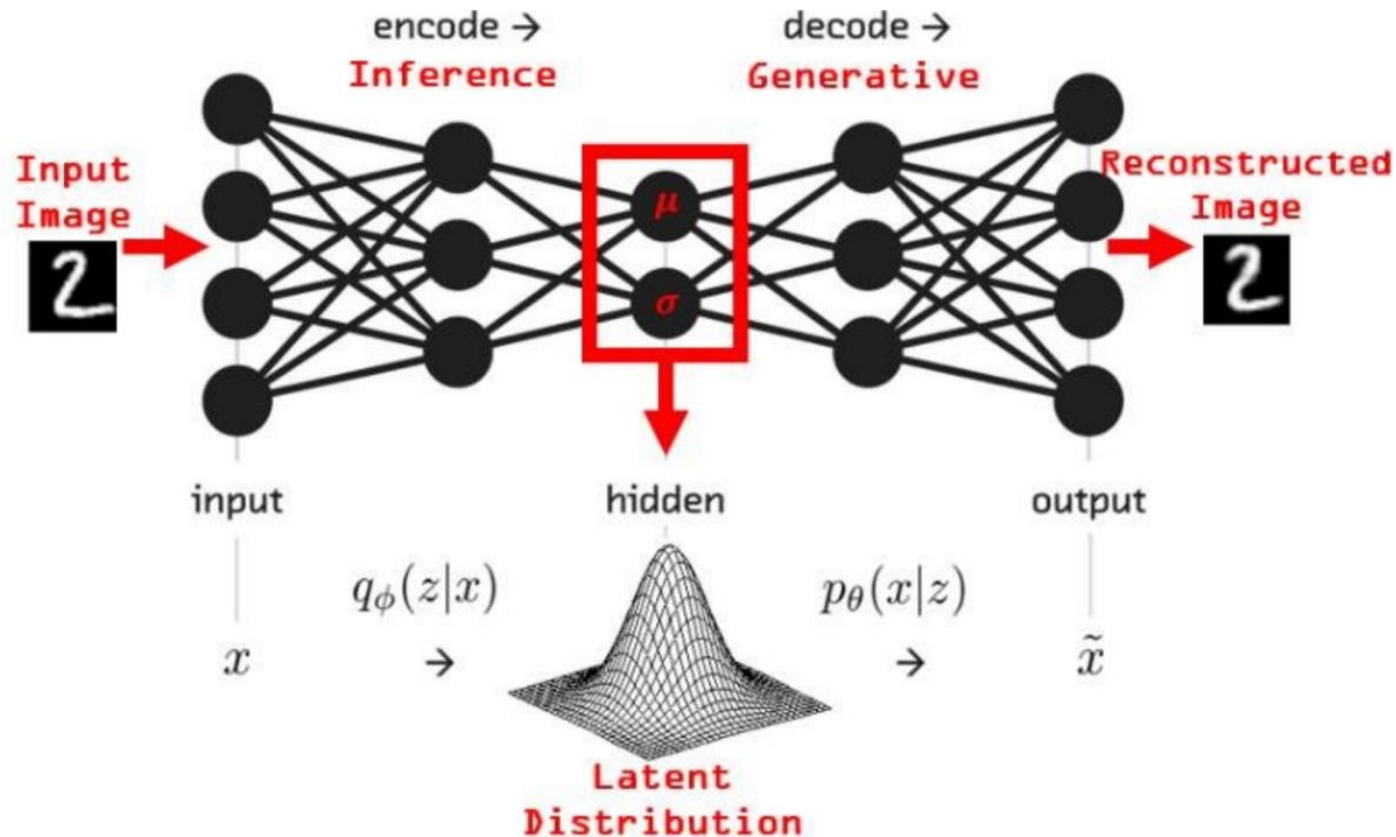


$$KL[Q(H|X; \theta_1) || N(0, I)] = 0.5(\text{tr}(\Sigma(X)) + \mu(X)^T \mu(X) - k - \log \det \Sigma(X))$$

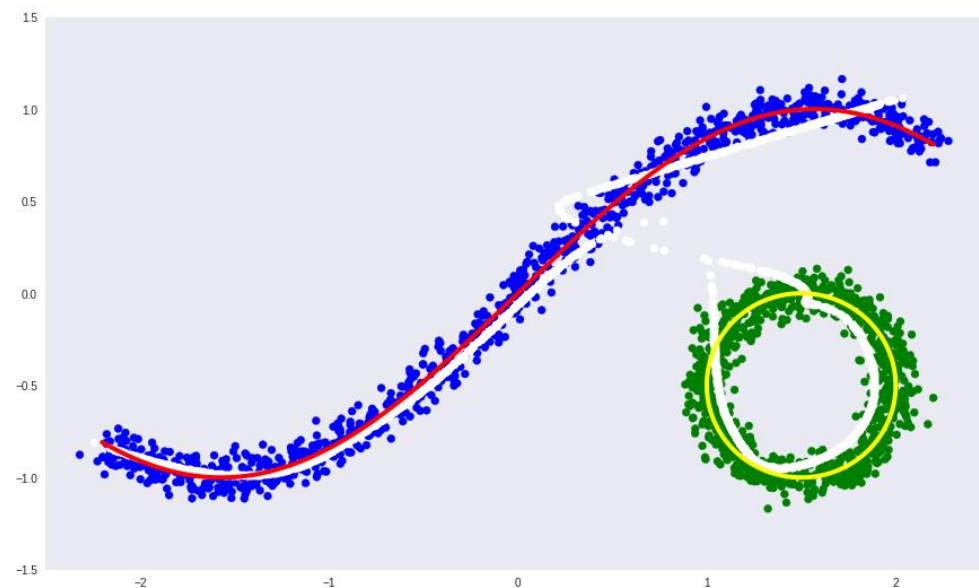
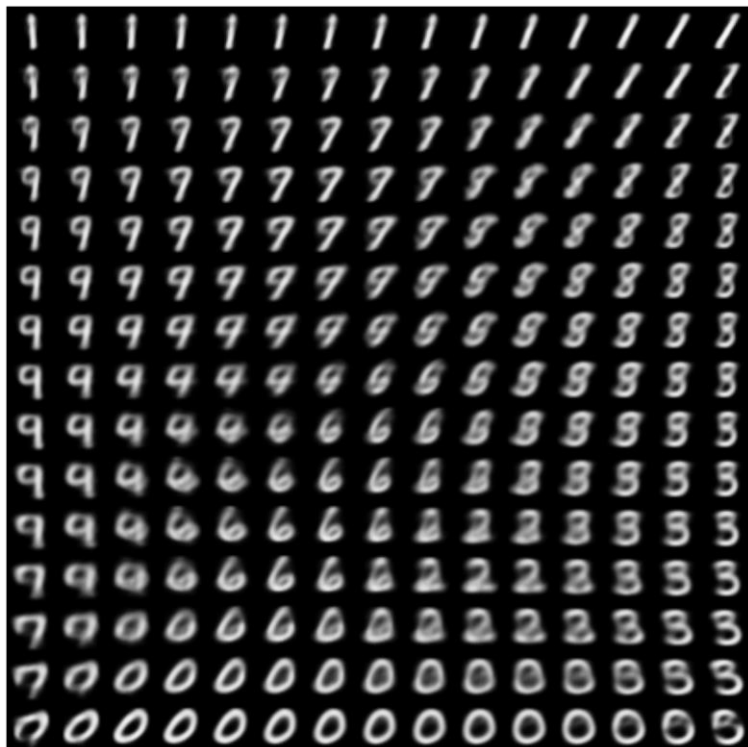
VAE: Вариационный автоэнкодер



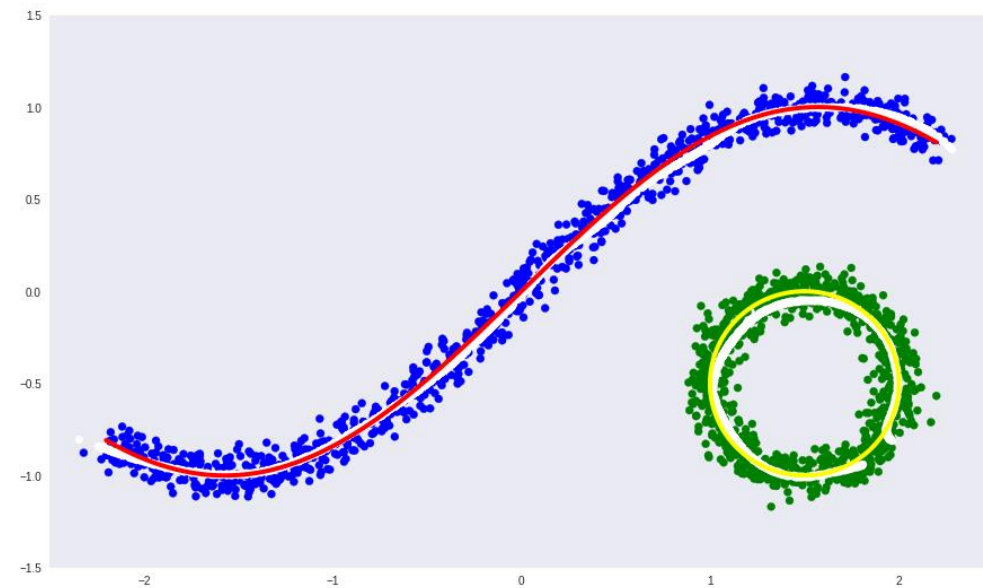
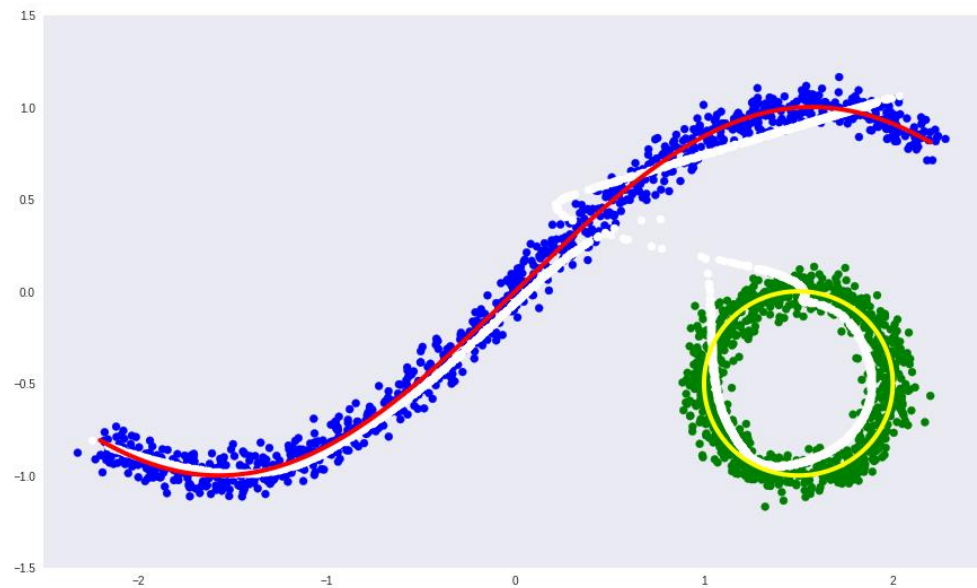
VAE: Вариационный автоэнкодер



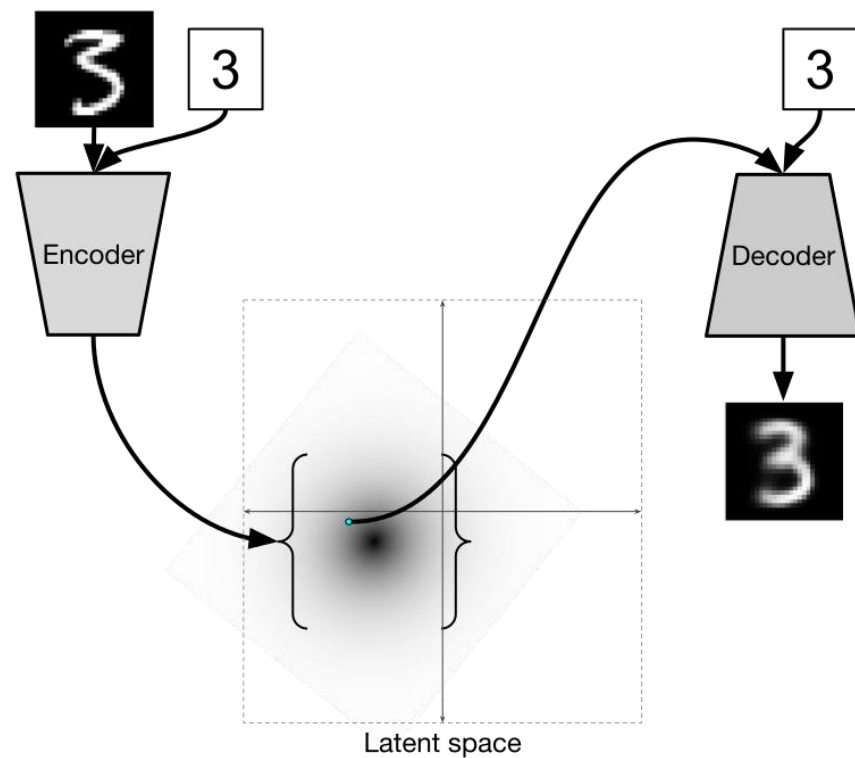
VAE: недостатки



Conditional VAE

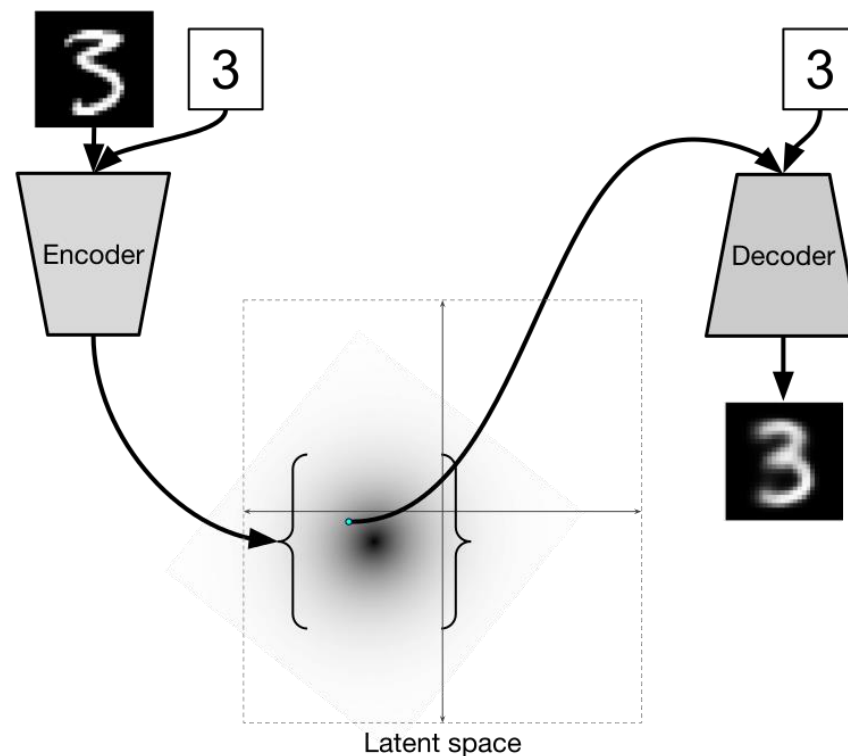


Conditional VAE



$$\log P(X|Y; \theta_2) - \text{KL}[Q(H|X, Y; \theta_1) || P(H|X, Y; \theta_2)] = E_{H \sim Q}[\log P(X|H, Y; \theta_2)] - \text{KL}[Q(H|X, Y; \theta_1) || N(0, I)]$$

Conditional VAE



<https://github.com/lyeoni/pytorch-mnist-VAE/blob/master/pytorch-mnist-VAE.ipynb>

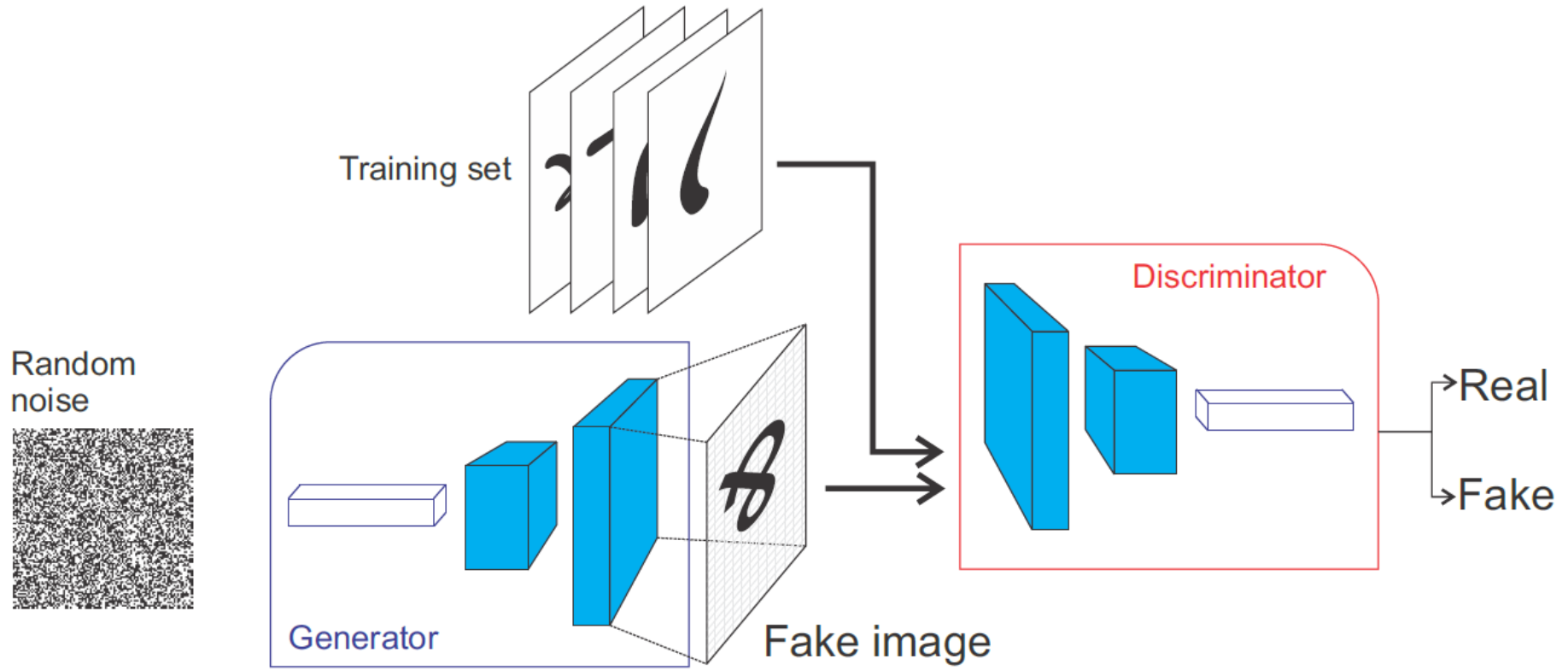
<https://github.com/lyeoni/pytorch-mnist-CVAE/blob/master/pytorch-mnist-CVAE.ipynb>

Conditional VAE: перенос стиля

- Обучаем CVAE на картинках с метками
- Кодировем стиль заданной картинки в H
- Меняем метки Y , создаем из закодированного H новые картинки

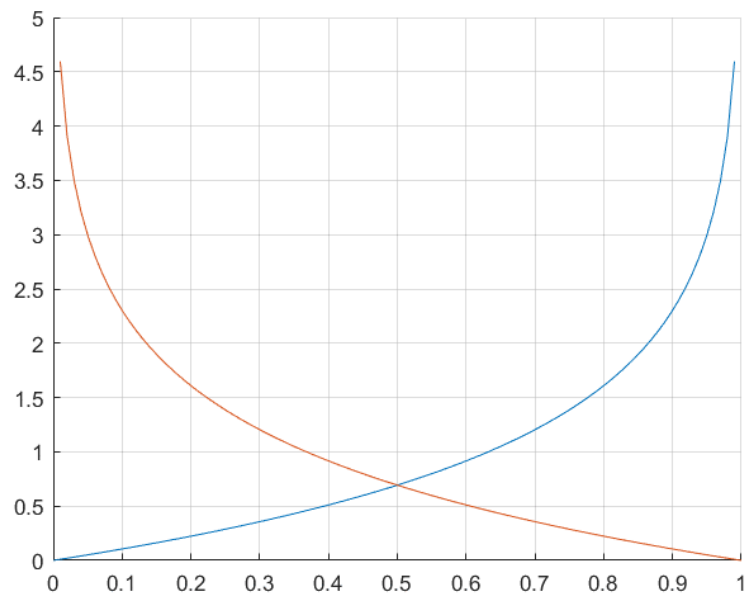


GAN

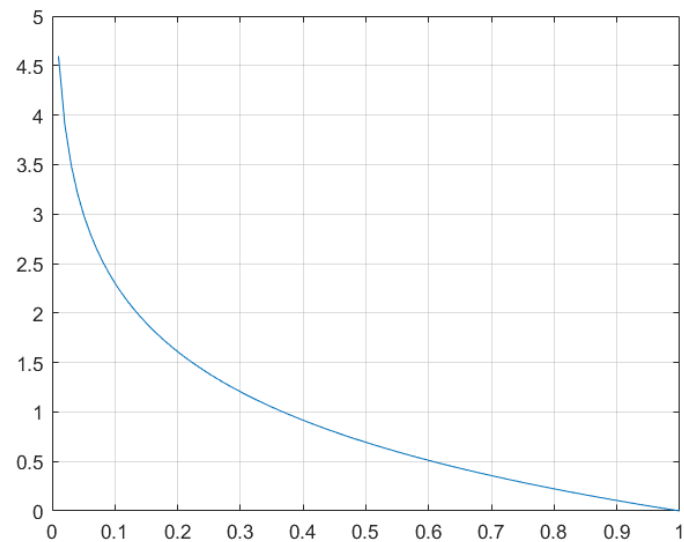


GAN

$$-E_x[\log(D(x))] - E_z[\log(1 - D(G(z)))]$$



$$-E_z[\log(D(G(z)))]$$



GAN+VAE

<https://github.com/csinva/gan-vae-pretrained-pytorch>

<https://github.com/rishabhd786/VAE-GAN-PYTORCH>