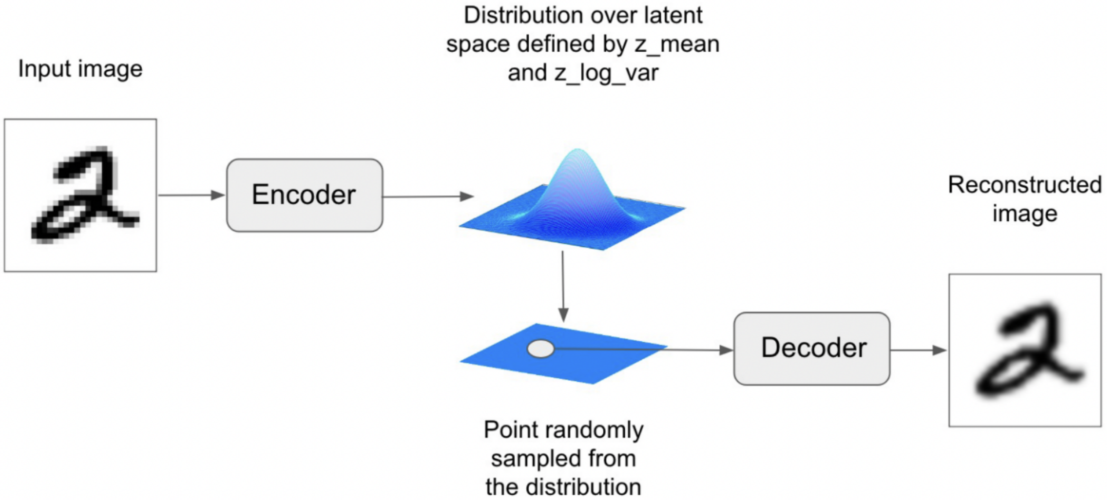
Вариационный автокодировщик

$$\sum_{i=1}^{\ell}\left(\mathbb{E}_{q(z|x_i)}\log p(x_i|z) - \mathrm{KL}\big(q(z|x)\parallel \mathcal{N}(0,1)\big)\right) \to \max$$

«функция потерь»
reconstruction likelihood

«регуляризатор»



$$\sum_{i=1}^{\ell}\left(\mathbb{E}_{q(z|x_i)}\log p(x_i|z) - \mathrm{KL}\big(q(z|x)\parallel \mathcal{N}(0,1)\big)\right) \to \max$$

- $q(z|x)$ — кодировщик (полносвязная или свёрточная нейросеть, z — возможный эмбеддинг для x)
- $p(x|z) = \mathrm{decoder}(z) + \varepsilon$ — декодировщик (полносвязная или свёрточная нейросеть)
- $\mathbb{E}_{q(z|x_i)}\log p(x_i|z)$ — как бы закодировали x_i, сгенерировали все возможные представления и посчитали среднюю ошибку реконструкции

l - все эмбединги для картинки
q - вероятность получить кодировщиком эмбеддинг z из xi
p - вероятность получить декодировщиком xi из эмбединга z

GAN

- Мы договорились, что X – класс 1, а $\hat{X} = G(Z)$ – класс 0.

- Тогда функция потерь имеет вид:

$$L(\boldsymbol{D}, \boldsymbol{G}) = -\frac{1}{n}\sum_{x_i \in X} \log \boldsymbol{D}(x_i) - \frac{1}{n}\sum_{z_i \in Z} \log\big(1 - \boldsymbol{D}(\boldsymbol{G}(z_i))\big)$$

- Обучение дискриминатора и генератора:

$$\max_{\boldsymbol{G}} \min_{\boldsymbol{D}} L(\boldsymbol{D}, \boldsymbol{G})$$

Если дискриминатор идеально разделяет объекты генератора и реальные:

Затухание градиентов

- В результате получаем:

$$L(\boldsymbol{D}, \boldsymbol{G}) = -\frac{1}{n}\sum_{x_i \in X} \log \mathbf{1} - \frac{1}{n}\sum_{z_i \in Z} \log(1 - \mathbf{0}) = 0 = \text{const}$$

- $\nabla L(\boldsymbol{D}, \boldsymbol{G}) = 0$ =>
- Дискриминатор и генератор больше **не обучаются**!

Теорема о замене переменных

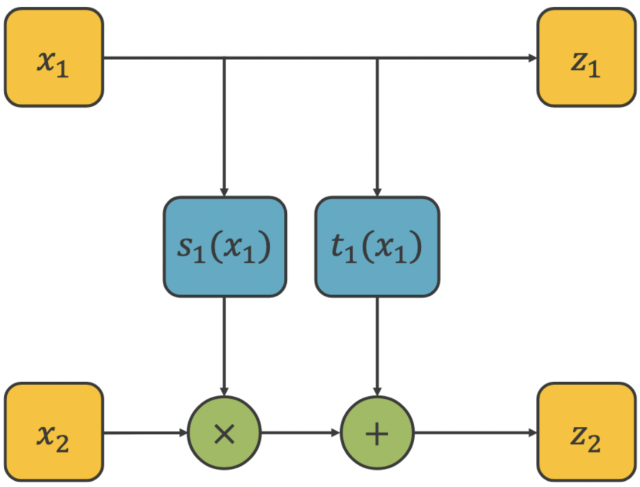
Пусть даны $\boldsymbol{p_z}(z)$ и $\mathrm{z} = \boldsymbol{f}(x)$, тогда $p_x(x)$ находим так:

$$p_x(x_i) = \boldsymbol{p_z}(\boldsymbol{f}(x_i)) \left| \det \frac{\partial \boldsymbol{f}(x_i)}{\partial x_i} \right|,$$

где матрица первых производных определяется так:

$$\frac{\partial \boldsymbol{f}(x_i)}{\partial x_i} = \begin{pmatrix} \frac{\partial \boldsymbol{f}(x_i)_1}{\partial x_{i1}} & \cdots & \frac{\partial \boldsymbol{f}(x_i)_1}{\partial x_{in}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{f}(x_i)_m}{\partial x_{i1}} & \cdots & \frac{\partial \boldsymbol{f}(x_i)_m}{\partial x_{in}} \end{pmatrix}.$$

Real-NVP



Функция

$$z = \boldsymbol{f}(x) = \begin{cases} z_{1:d} = x_{1:d} \\ z_{d+1:D} = x_{d+1:D} \odot \exp\big(\boldsymbol{s}(x_{1:d})\big) + \boldsymbol{t}(x_{1:d}) \end{cases}$$

где:

- $z_{1:d}$ - первые d компонент вектора z;
- $\boldsymbol{s}(x_{1:d})$ и $\boldsymbol{t}(x_{1:d})$ – **нейронные сети** с d входами и $D - d$ выходами;
- $\odot$ - поэлементное умножение.

Якобиан

- Матрица первых производных:

$$\frac{\partial \boldsymbol{f}(x)}{\partial x} = \begin{pmatrix} \mathbb{I}_d & 0 \\ \frac{\partial z_{1:d}}{\partial x_{1:d}} & diag\,(\exp(\boldsymbol{s}(x_{1:d}))) \end{pmatrix}$$

- Значение Якобиана:

$$\left|\det \frac{\partial \boldsymbol{f}(x)}{\partial x}\right| = \exp(\sum_{j=d+1}^{D} \boldsymbol{s}(x_{1:d})_j)$$

Обратная функция

$$x = \boldsymbol{f}^{-1}(z) = \begin{cases} x_{1:d} = z_{1:d} \\ x_{d+1:D} = (z_{d+1:D} - \boldsymbol{t}(x_{1:d})) \odot \exp\big(-\boldsymbol{s}(x_{1:d})\big) \end{cases}$$