

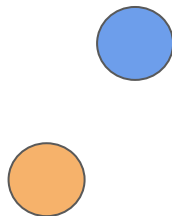
Semi-supervised Learning with Deep Generative Models

Кожевников Георгий

Semi-supervised learning

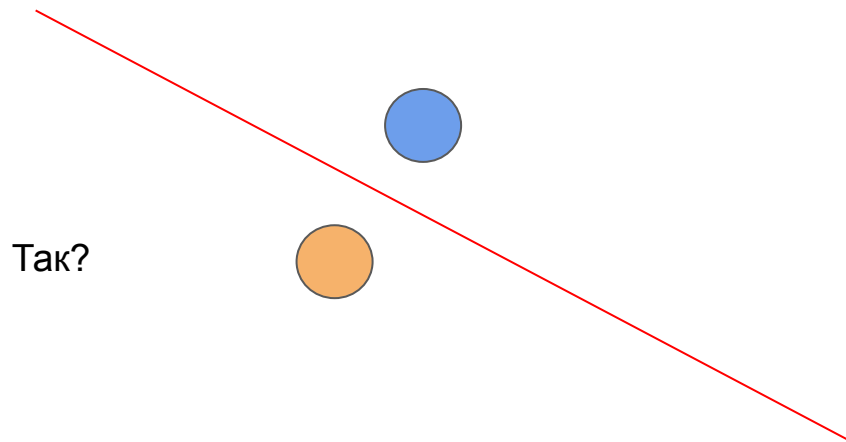
Обучение с использованием неразмеченных данных

Как бы вы провели
разделяющую прямую?



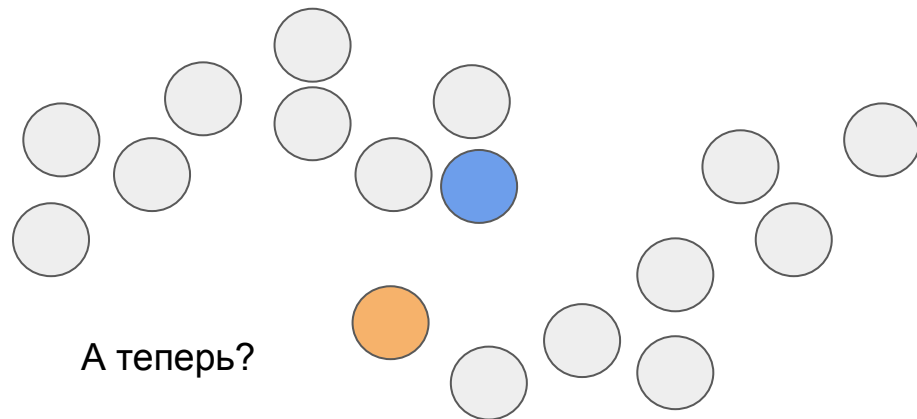
Semi-supervised learning

Обучение с использованием неразмеченных данных



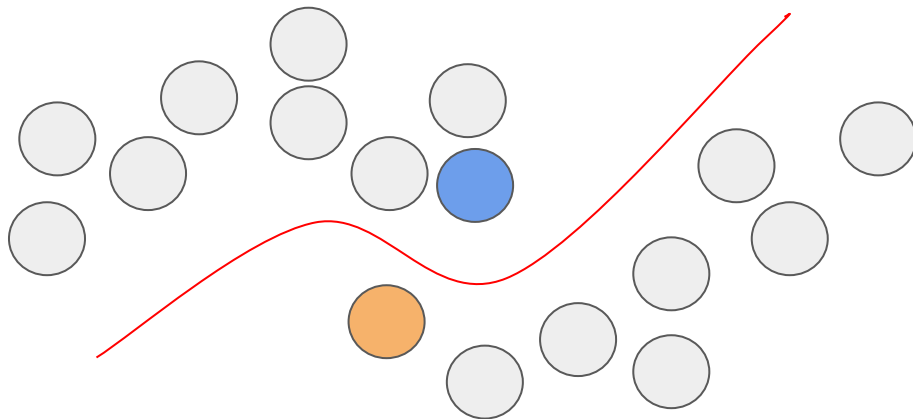
Semi-supervised learning

Обучение с использованием неразмеченных данных



Semi-supervised learning

Неразмеченные данные **имеют значение** и могут помогать в решении задачи.



Semi-supervised learning

Применения

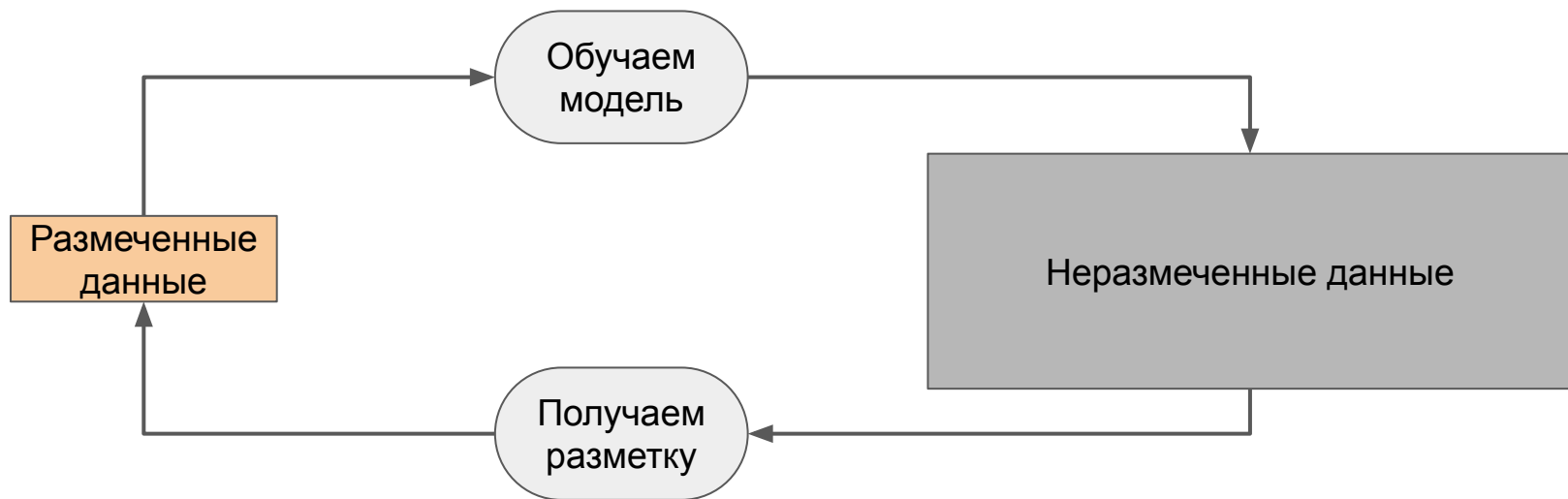
Задачи, в которых сложно добыть разметку:

- Разметка речи
- Сегментация изображений, облак точек
- Разметка, требующая эксперта (медицинские данные)
- Разметка тегов слов

Semi-supervised learning

Как использовать неразмеченные данные в обучении?

Self-training



Semi-supervised learning

Как использовать неразмеченные данные в обучении?

Self-training



Легко



Можно использовать уже готовые сильные модели



Широко применим



Ошибки сами себя подкрепляют

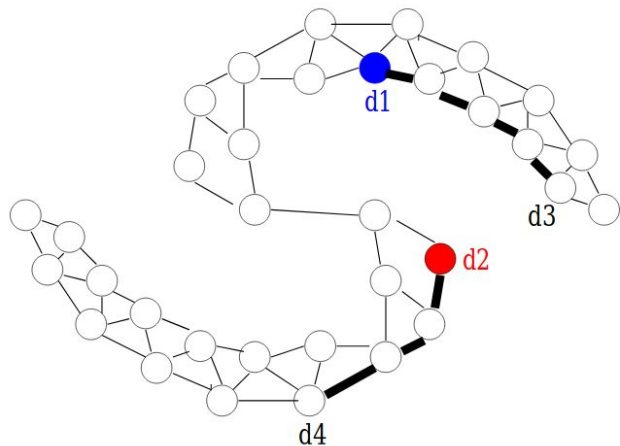


Сложно формализовать

Semi-supervised learning

Как использовать неразмеченные данные в обучении?

Graph-based



Строим граф, вершины - точки, рёбра - функции от расстояния между ними.

С помощью графа размечаем новые точки, похожие на существующие размеченные

Semi-supervised learning

Как использовать неразмеченные данные в обучении?

Graph-based



Хорошо показывает себя при хорошем подборе графа



Всё формально



Нужно подбирать граф

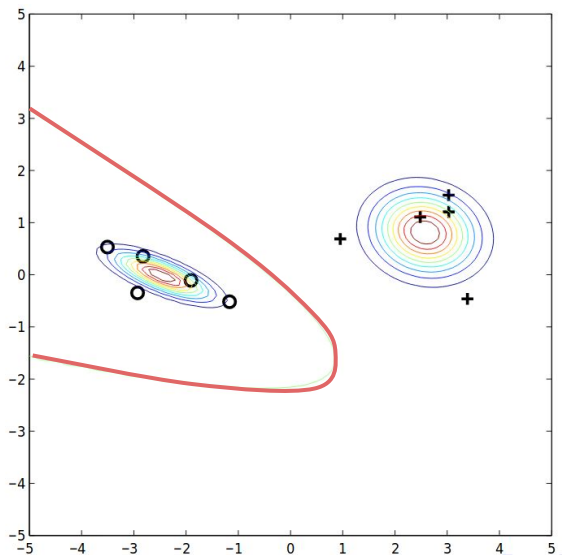
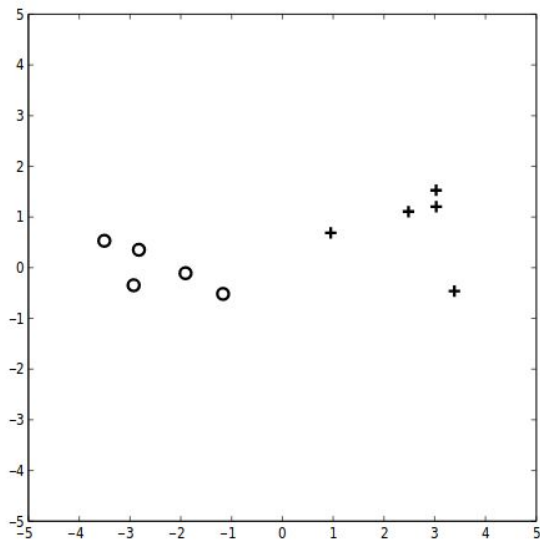


Если граф плохой, то вычислительная эффективность плохая

Semi-supervised learning

Как использовать неразмеченные данные в обучении?

Generative models

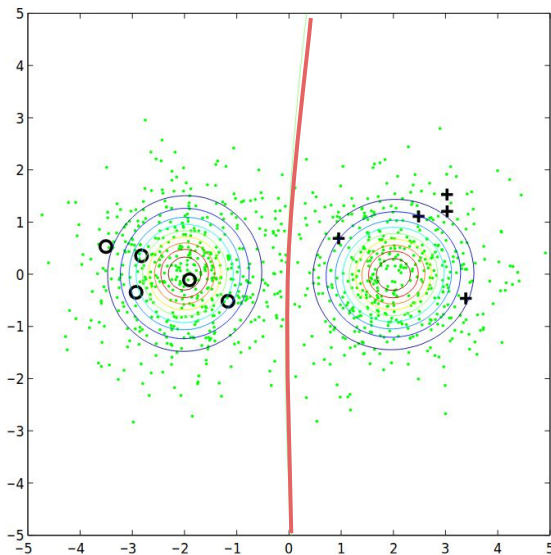
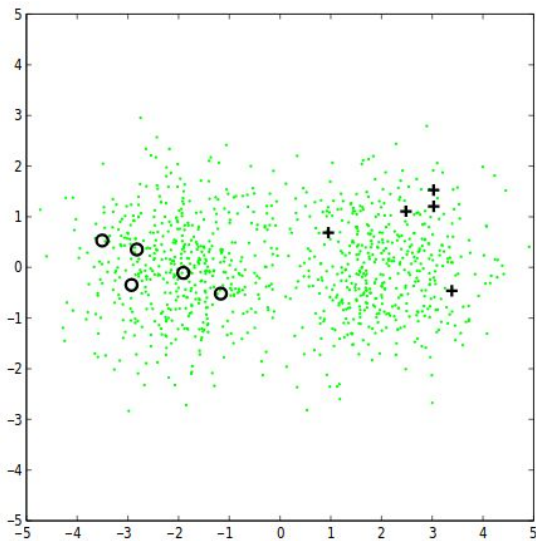


$$p(X_l, Y_l | \theta)$$

Semi-supervised learning

Как использовать неразмеченные данные в обучении?

Generative models



Модель учит
совместное
распределение,
максимизируя его
правдоподобие

$$p(X_l, Y_l, X_u | \theta)$$

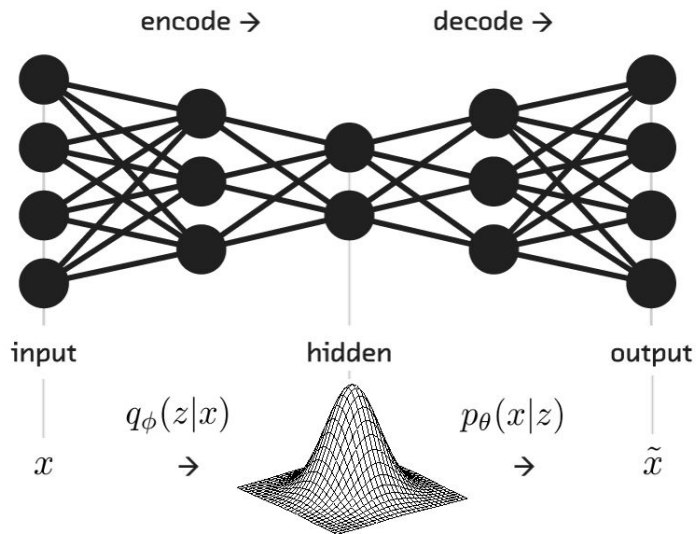
Semi-supervised learning

with Deep generative models

Статья предлагает две модели (M1 и M2), позволяющие использовать размеченные и неразмеченные данные для решения задач машинного обучения.

Semi-supervised learning

Deep generative models: пререквизит - VAE



Идея **автоэнкодера** - закодировать данные в скрытое представление, и восстановить их из него.

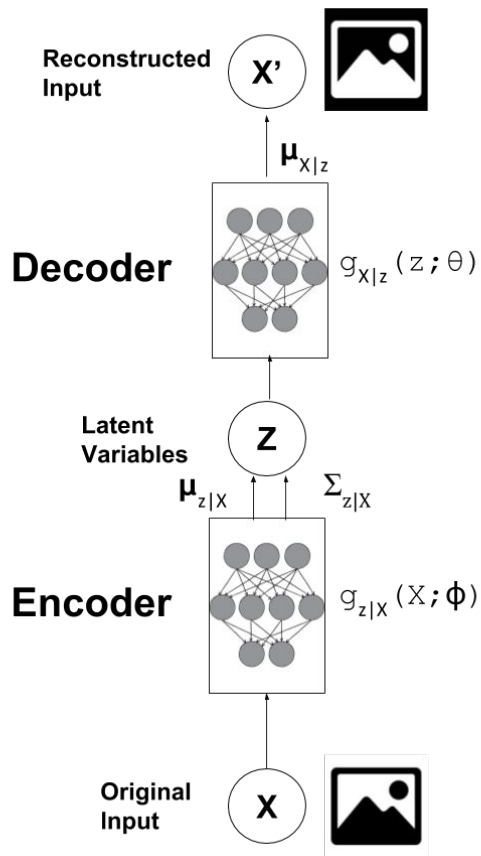
Отличие **вариационного автоэнкодера** - скрытое представление является распределением.

Semi-supervised learning

Deep generative models: пререквизит - VAE

Энкодер возвращает параметры скрытого распределения. Затем из него сэмплируются значения скрытых переменных, которые идут в декодер.

После обучения можем использовать энкодер для классификации, надев на него нужную голову.



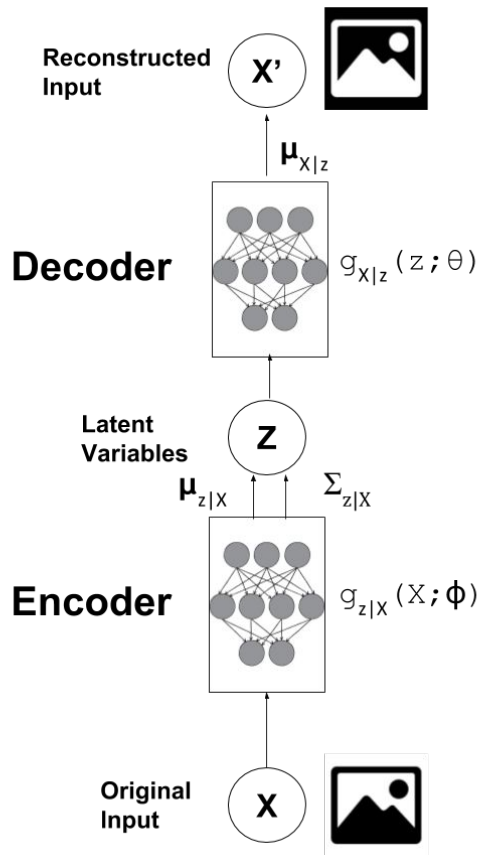
Semi-supervised learning

Deep generative models: пререквизит - VAE

Тут не нужны метки

Обучаемся и на размеченных и на
неразмеченных данных.

Это и есть модель M1



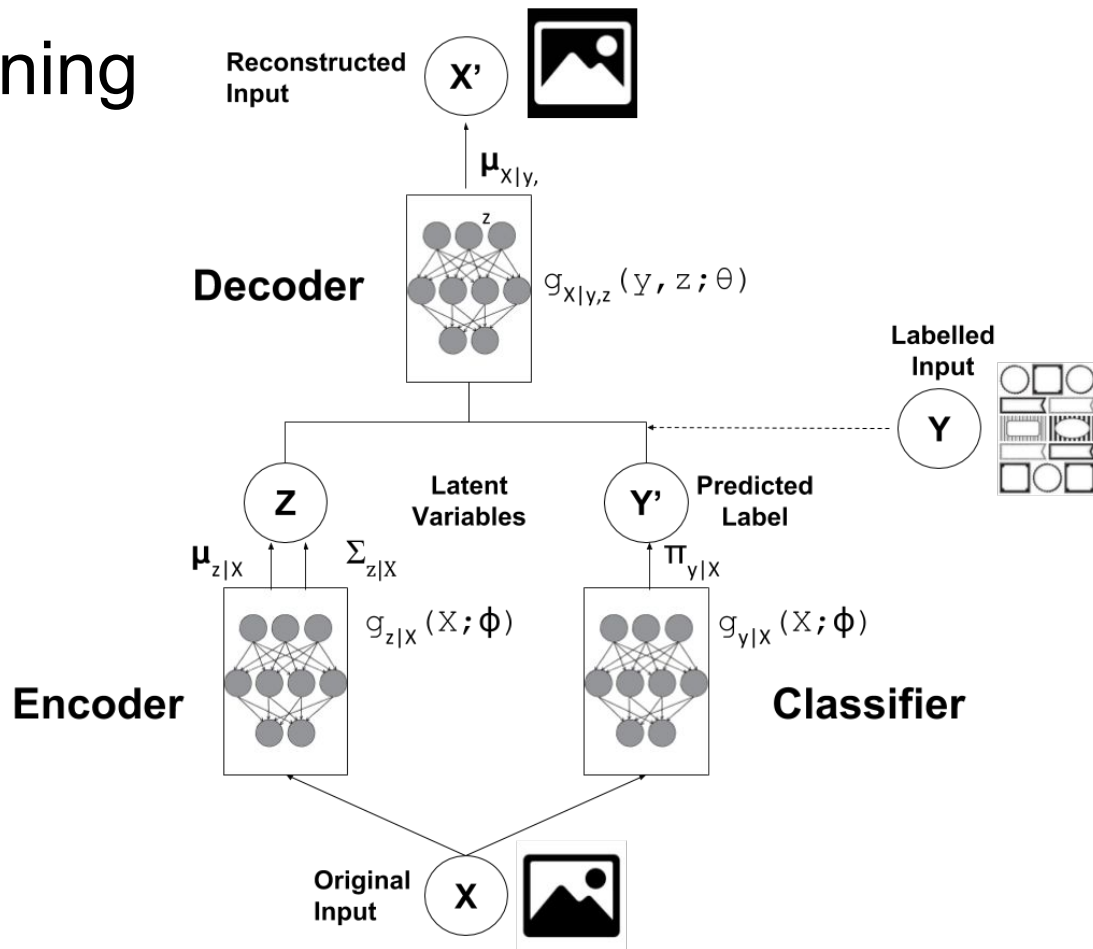
Semi-supervised learning

Deep generative models: M2

Идея - использовать метки классов как скрытые переменные.

Моделируем Z энкодером, как и в M1, а распределение Y - дополнительной сетью классификатором.

По сэмплам Z и Y пытаемся восстановить X .



Semi-supervised learning

Deep generative models: формально

скрытые переменные:

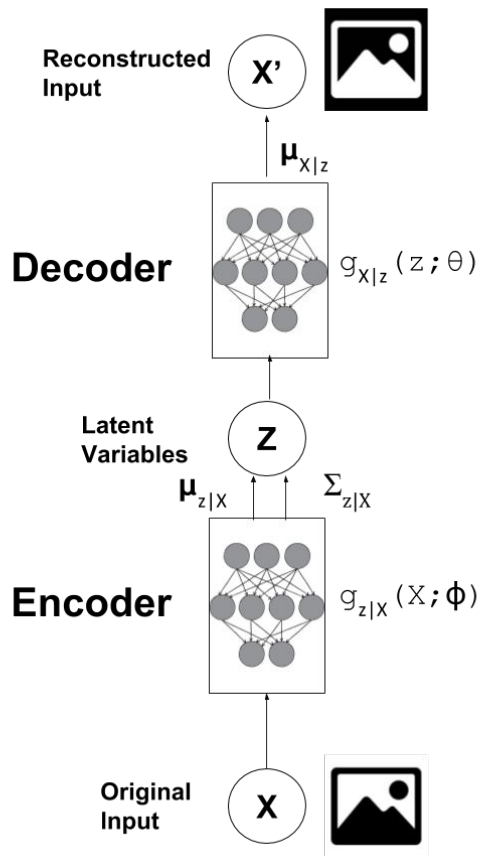
$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I});$$

декодер:

$$p_{\theta}(\mathbf{x}|\mathbf{z}) = f(\mathbf{x}; \mathbf{z}, \theta),$$

энкодер:

$$q_{\phi}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\phi}(\mathbf{x}), \text{diag}(\boldsymbol{\sigma}_{\phi}^2(\mathbf{x})))$$



Semi-supervised learning

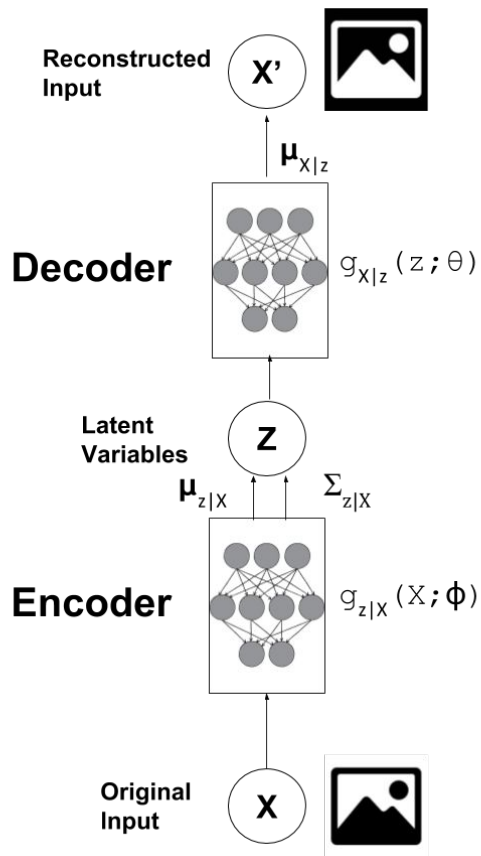
Deep generative models: формально

Нижняя оценка правдоподобия (ELBO):

$$\log p_{\theta}(\mathbf{x}) \geq \underbrace{\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})]}_{\text{reconstruction loss}} - \underbrace{KL[q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\theta}(\mathbf{z})]}_{\text{KL loss}} = -\mathcal{J}(\mathbf{x})$$

KL loss выводится аналитически, дифференцируем.

Reconstruction loss считается с помощью reparametrization trick (location-scale transformation)

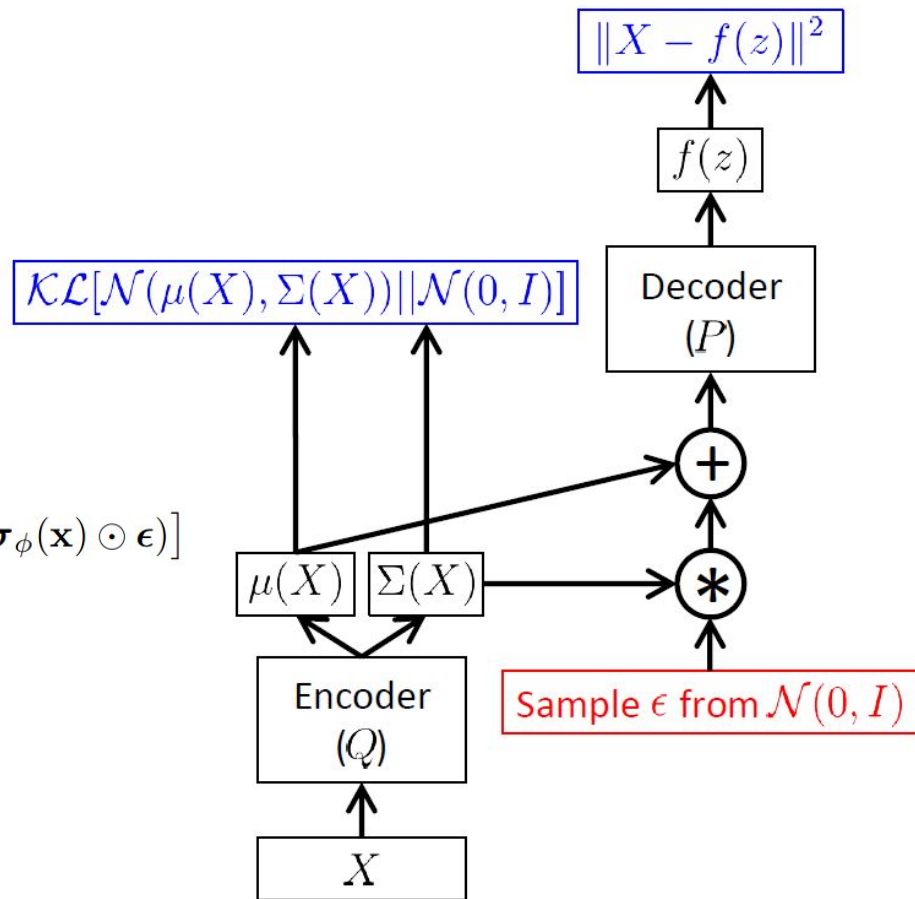


Semi-supervised learning

Deep generative models: формально

Reparametrization trick

$$\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] = \mathbb{E}_{\mathcal{N}(\epsilon|\mathbf{0},\mathbf{I})} [\log p_{\theta}(\mathbf{x}|\boldsymbol{\mu}_{\phi}(\mathbf{x}) + \boldsymbol{\sigma}_{\phi}(\mathbf{x}) \odot \epsilon)]$$



Semi-supervised learning

Deep generative models: формально

скрытые переменные:

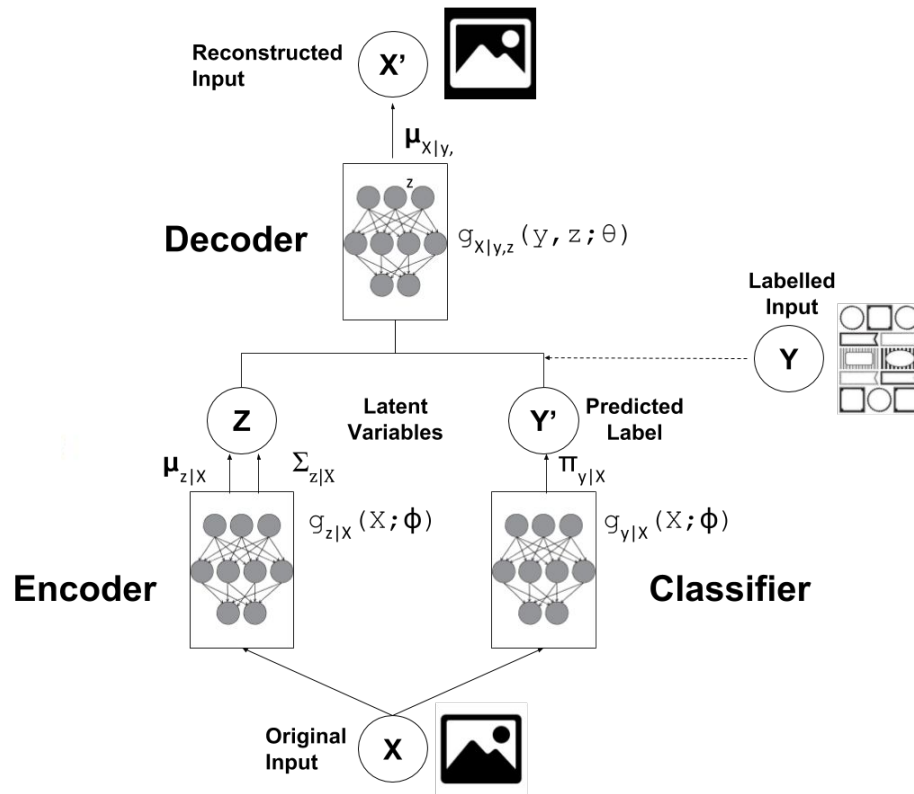
$$p(y) = \text{Cat}(y|\pi); \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I});$$

декодер:

$$p_{\theta}(\mathbf{x}|y, \mathbf{z}) = f(\mathbf{x}; y, \mathbf{z}, \theta).$$

энкодер:

$$q_{\phi}(\mathbf{z}|y, \mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\phi}(y, \mathbf{x}), \text{diag}(\boldsymbol{\sigma}_{\phi}^2(\mathbf{x}))); \quad q_{\phi}(y|\mathbf{x}) = \text{Cat}(y|\boldsymbol{\pi}_{\phi}(\mathbf{x}))$$



Semi-supervised learning

Deep generative models: формально

$$\log p_{\theta}(\mathbf{x}, y) \geq \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}, y)} [\log p_{\theta}(\mathbf{x}|y, \mathbf{z}) + \log p_{\theta}(y) + \log p(\mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x}, y)] = -\mathcal{L}(\mathbf{x}, y)$$

$$\begin{aligned} \log p_{\theta}(\mathbf{x}) &\geq \mathbb{E}_{q_{\phi}(y, \mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|y, \mathbf{z}) + \log p_{\theta}(y) + \log p(\mathbf{z}) - \log q_{\phi}(y, \mathbf{z}|\mathbf{x})] \\ &= \sum_y q_{\phi}(y|\mathbf{x}) (-\mathcal{L}(\mathbf{x}, y)) + \mathcal{H}(q_{\phi}(y|\mathbf{x})) = -\mathcal{U}(\mathbf{x}). \end{aligned}$$

$$\mathcal{J} = \sum_{(\mathbf{x}, y) \sim \tilde{p}_l} \mathcal{L}(\mathbf{x}, y) + \sum_{\mathbf{x} \sim \tilde{p}_u} \mathcal{U}(\mathbf{x}) \qquad \mathcal{J}^{\alpha} = \mathcal{J} + \alpha \cdot \mathbb{E}_{\tilde{p}_l(\mathbf{x}, y)} [-\log q_{\phi}(y|\mathbf{x})]$$

Semi-supervised learning

Deep generative models: результаты

N	NN	CNN	TSVM	CAE	MTC	AtlasRBF	M1+TSVM	M2	M1+M2
100	25.81	22.98	16.81	13.47	12.03	8.10 (± 0.95)	11.82 (± 0.25)	11.97 (± 1.71)	3.33 (± 0.14)
600	11.44	7.68	6.16	6.3	5.13	–	5.72 (± 0.049)	4.94 (± 0.13)	2.59 (± 0.05)
1000	10.7	6.45	5.38	4.77	3.64	3.68 (± 0.12)	4.24 (± 0.07)	3.60 (± 0.56)	2.40 (± 0.02)
3000	6.04	3.35	3.45	3.22	2.57	–	3.49 (± 0.04)	3.92 (± 0.63)	2.18 (± 0.04)



(a) Handwriting styles for MNIST obtained by fixing the class label and varying the 2D latent variable z



(b) MNIST analogies



(c) SVHN analogies

