Глубокие генеративные модели

Е. Бурнаев,

Сколтех, Москва

Слайды были адаптированы из лекции Fei-Fei Li & Justin Johnson & Serena Yeung

План

- Кратко об обучении без учителя
- База: автоэнкодеры (AE)
- Генеративные модели
  - Вариационные автоэнкодеры (VAE)
  - Генеративно-состязательные сети (GANs)

Обучение с учителем и без учителя

Обучение с учителем

Данные: (x, y)
x - признаки, y - метки

Цель: Обучить функцию отображения x -> y

Примеры: задачи классификации, регрессии, распознавания объектов (детекции), семантической сегментации, составления подписей к изображениям, и т.д.

Кот

Классификация

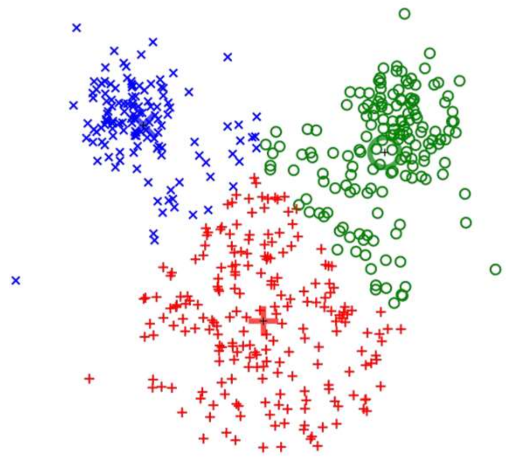
Обучение с учителем и без учителя

Обучение без учителя

Данные: x
Только признаки, без меток!

Цель: Найти некоторую скрытую структуру данных

Примеры: кластеризация, снижение размерности, извлечение признаков, оценка плотности и т.д.



Кластеризация методом k средних (K-means)

Генеративные Модели

Получают выборку для обучения, порождают новые примеры из того же распределения

Данные для обучения ~ $p_{data}(x)$

Сгенерированные примеры ~ $p_{model}(x)$

Хотим получить $p_{model}(x)$, похожее на $p_{data}(x)$

Генеративные Модели

Получают выборку для обучения, порождают новые примеры из того же распределения

Данные для обучения ~ $p_{data}(x)$

Сгенерированные примеры ~ $p_{model}(x)$

Хотим получить $p_{model}(x)$, похожее на $p_{data}(x)$

Рассмотрим оценку плотности — основную проблему в обучении без учителя

Некоторые подходы:

- Явная оценка плотности: явно определяем и вычисляем $p_{model}(x)$
- Неявная оценка плотности: обучить модель, умеющую сэмплировать из $p_{model}(x)$ без явного определения плотности

Почему Генеративные Модели?

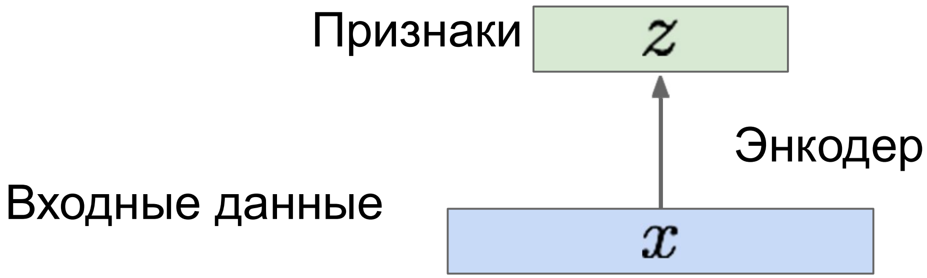
- Реалистичность при художественных работах, повышении разрешения изображения, цветокоррекции и т.д.

- Используются для моделирования временных рядов (а также в обучении с подкреплением!)
- Позволяют находить скрытые представления, которые могут быть использованы в качестве новых признаков

Skoltech
Skolkovo Institute of Science and Technology

Немного предыстории: Автоэнкодеры

Метод обучения без учителя для выучивания низкоразмерного представления данных без меток в пространстве признаков



Немного предыстории: Автоэнкодеры

Метод обучения без учителя для представления данных без меток в пространстве признаков меньшей размерности

z обычно меньше x
(снижение размерности)

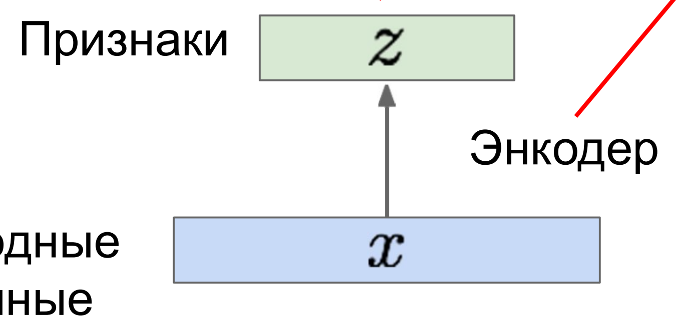
Q: Почему снижение размерности?

A: Хотим, чтобы функции фиксировали значимые факторы изменения данных

Первоначально: линейный слой + нелинейность (сигмоида)

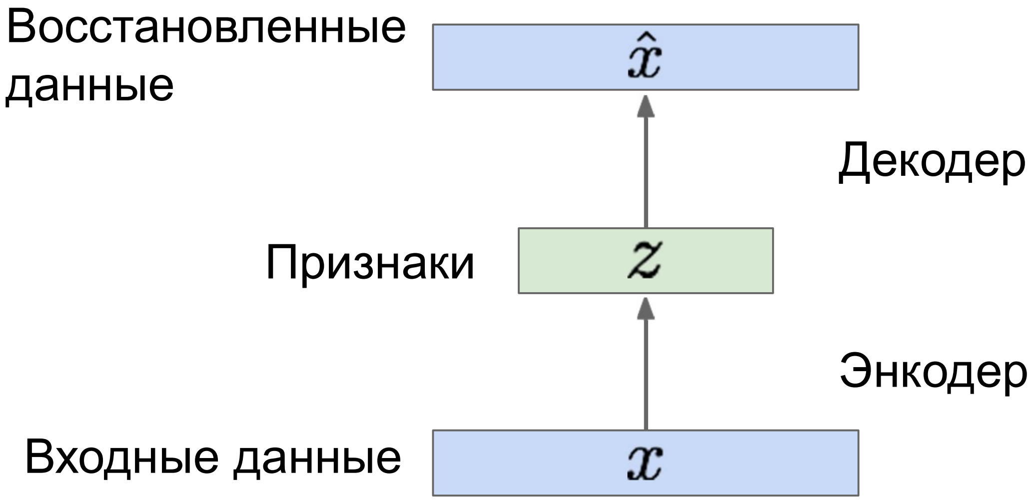
Позднее: полносвязные, более глубокие нейронные сети

Ещё позднее: ReLU CNN



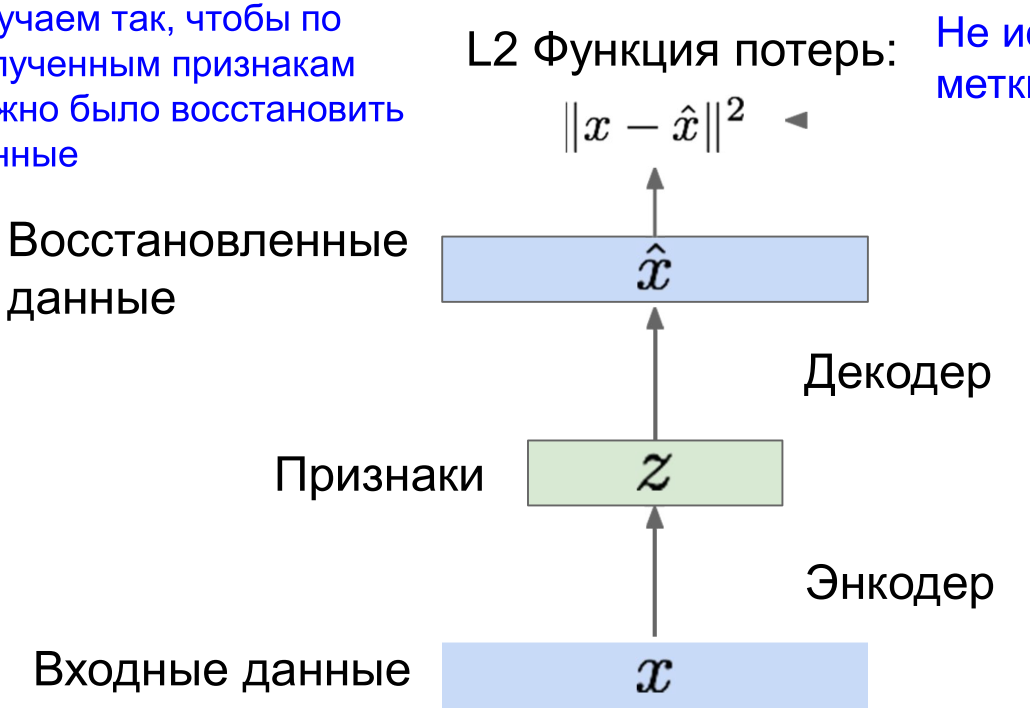
Немного предыстории: Автоэнкодеры

Как получить это представление?
Выучить признаки, по которым можно восстановить данные
“Autoencoding” - самокодирование



Немного предыстории: Автоэнкодеры

Обучаем так, чтобы по полученным признакам можно было восстановить данные



Не использует метки!

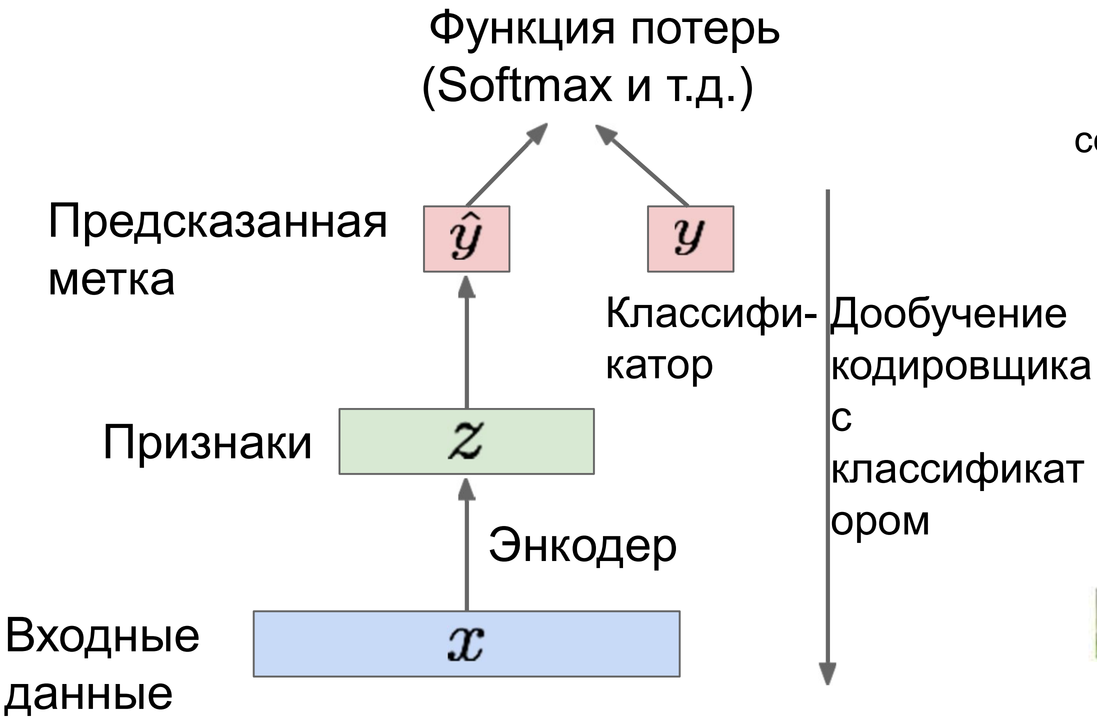
Восстановленные данные

Encoder: 4-слойная conv NN
Декоде: 4-слойная upconv NN

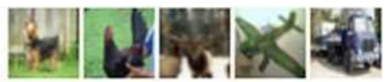
Входные данные

Немного предыстории: Автоэнкодеры

Энкодер можно использовать для инициализации модели, обучаемой с учителем



Дообучение под итоговую задачу (иногда на малых данных)



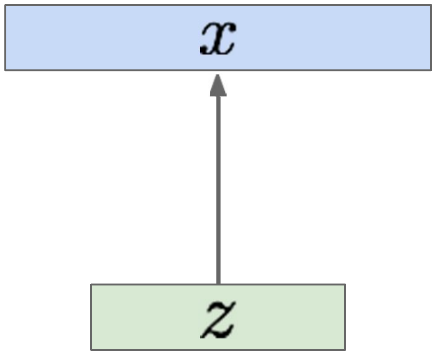
Вариационные Автоэнкодеры (VAE)

Вариационные Автоэнкодеры

Вероятностный взгляд на энкодеры — семплируем из модели для получения данных!
Предположим, данные для обучения $\{x^{(i)}\}_{i=1}^{N}$ получены из скрытого ненаблюдаемого (латентного) представления **z**

Выборка из настоящего распределения
$p_{\theta^*}(x \mid z^{(i)})$

Выборка из настоящего априорного распределения
$p_{\theta^*}(z)$



Интуитивно (помним тз автоэнкодеров!):
x - изображение, **z** — скрытые параметры для генерации **x:** атрибуты, ориентация и т.д.

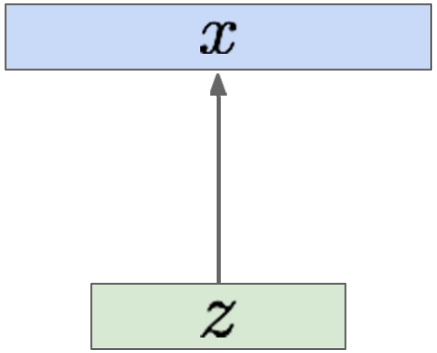
Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры

Хотим определить настоящие параметры θ^* этой генеративной модели

Выборка из настоящего распределения $p_{\theta^*}(x \mid z^{(i)})$



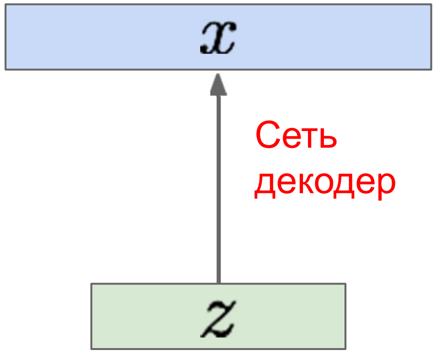
Выборка из настоящего априорного распределения $p_{\theta^*}(z)$

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные Автоэнкодеры

Выборка из настоящего распределения $p_{\theta^*}(x \mid z^{(i)})$

Выборка из настоящего априорного распределения $p_{\theta^*}(z)$



Хотим определить настоящие параметры θ^* этой генеративной модели

Как представить эту модель?

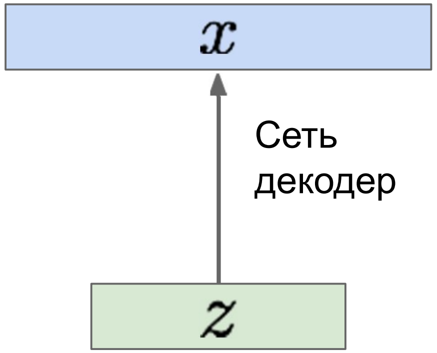
Выберем простое априорное распределение p(z), например, гауссовское.
Распределение p(x|z) сложное (генерирует изображения) => приближаем нейросетью

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные Автоэнкодеры

Выборка из настоящего распределения $p_{\theta^*}(x \mid z^{(i)})$

Выборка из настоящего априорного распределения



Хотим определить настоящие параметры θ^* этой генеративной модели

Как обучать модель?

Обучаем параметры, максимизируя функцию правдоподобия на обучающей выборке

$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$

Q: В чём проблема?

A: Плотность не вычислима!

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные Автоэнкодеры: невычислимость

Правдоподобие данных: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные Автоэнкодеры: невычислимость

Правдоподобие данных: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Простое априорное
гауссовское
распределение

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры: невычислимость

Правдоподобие данных: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Нейросеть-декодер

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры: невычислимость

X

Правдоподобие данных: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Невозможно вычислить p(x|z) для каждого z!

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные Автоэнкодеры: невычислимость

Правдоподобие данных: $p_\theta(x) = \int \overset{\times}{p_\theta(z)} \, p_\theta(x|z) dz$

Апостериорная плотность также невычислима: $p_\theta(z|x) = p_\theta(x|z)p_\theta(z)/p_\theta(x)$

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры: невычислимость

Правдоподобие данных: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Апостериорная плотность также невычислима: $p_\theta(z|x) = p_\theta(x|z) p_\theta(z) / p_\theta(x)$

Невычислимое
правдоподобие данных

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные Автоэнкодеры: невычислимость

Правдоподобие данных: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Апостериорная плотность также невычислима: $p_\theta(z|x) = p_\theta(x|z) p_\theta(z) / p_\theta(x)$

Решение: добавим к декодеру, аппроксимирующему $p_\theta(x|z)$, энкодер $q_\phi(z|x)$, аппроксимирующий $p_\theta(z|x)$

Это позволяет получить вычислимую нижнюю границу правдоподобия данных, которую можно оптимизировать

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

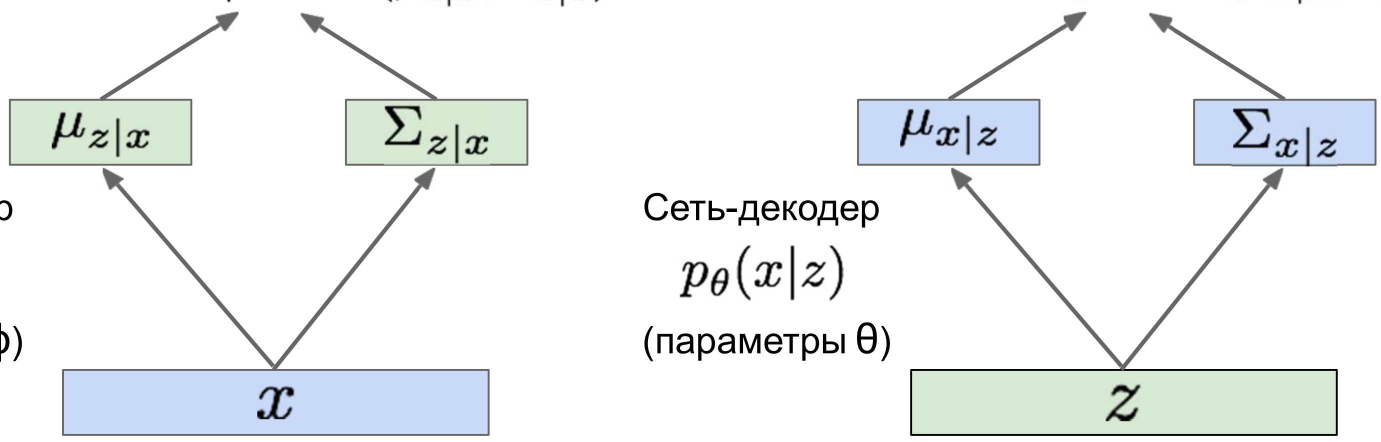
Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры

Поскольку моделируется вероятностная генерация данных, энкодер и декодер - вероятностные

семплируем z из $z|x \sim \mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$

семплируем x|z из $x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$



Сеть-энкодер

$q_\phi(z|x)$

(параметры ϕ)

Сеть-декодер

$p_\theta(x|z)$

(параметры θ)

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные Автоэнкодеры

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z \sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

Берём ожидание по z (используем сеть-энкодер), это пригодится нам позже

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Bayes' Rule)}$$

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad (\text{Multiply by constant})
\end{aligned}$$

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Bayes' Rule)} \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \qquad \text{(Multiply by constant)} \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Logarithms)}
\end{aligned}
$$

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z)\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Bayes' Rule)}\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \qquad \text{(Multiply by constant)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \qquad \text{(Logarithms)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))
\end{aligned}$$

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))$$

Взятие мат.ожидания по z (используя сеть-энкодер) позволяет выразить часть в терминах KL-дивергенции

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))$$

Сеть-декодер даёт $p_\theta(x|z)$, может вычислить оценку через семплирование. (Семплирование дифференцируемо через репараметрический трюк.)

Эту KL-дивергенцию (между Гауссианами для энкодера и априорного распределения на z) можно выписать в явном виде

$p_\theta(z|x)$ невычислима (см. ранее), не можем вычислить дивергенцию :(но знаем, что она всегда >= 0.

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \qquad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \qquad (\text{Multiply by constant}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \qquad (\text{Logarithms}) \\
&= \boxed{\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)}} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\boxed{\geq 0}}
\end{aligned}$$

Вычислимая нижняя граница, у которой можно брать градиент и оптимизировать! ($p_\theta(x|z)$ дифференцируема, KL-дивергенция дифференцируема)

Skoltech
Skolkovo Institute of Science and Technology

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z) p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z) p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad (\text{Multiply by constant}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Logarithms}) \\
&= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}
\end{aligned}
$$

$$\boxed{\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)}, \theta, \phi)}$$

Вероятностная нижняя граница (“ELBO”)

$$\boxed{\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, \theta, \phi)}$$

Обучение: максимизируем нижнюю границу

Вариационные Автоэнкодеры

Теперь, вооружившись энкодером и декодером, определим логарифм правдоподобия данных:

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

$$= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}$$

Восстанавливаем входные данные

Делаем приближённое апостериорное распределение близким к априорному

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)},\theta,\phi)$$

Вероятностная нижняя граница (“ELBO”)

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)},\theta,\phi)$$

Обучение: максимизируем нижнюю границу

Skoltech
Skolkovo Institute of Science and Technology

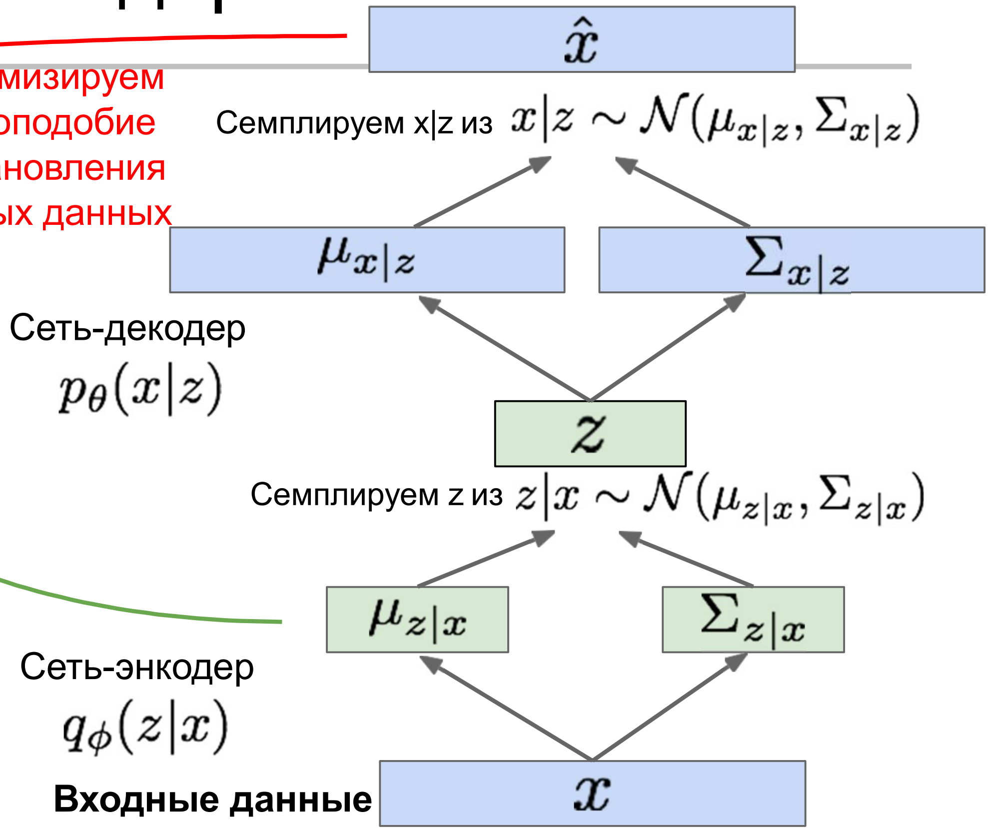
Вариационные автоэнкодеры

Итого: максимизируем нижнюю оценку правдоподобия

Максимизируем правдоподобие восстановления входных данных

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

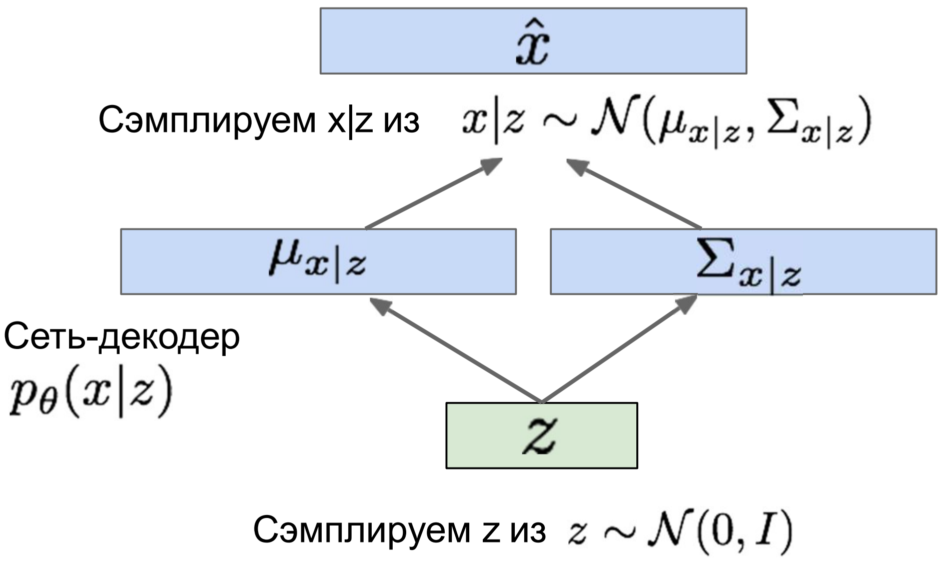
Делаем приближенное апостериорное распределение близким к априорному



Для каждого минибатча: вычисляем прямой проход, затем — обратный!

Вариационные автоэнкодеры: порождение данных!

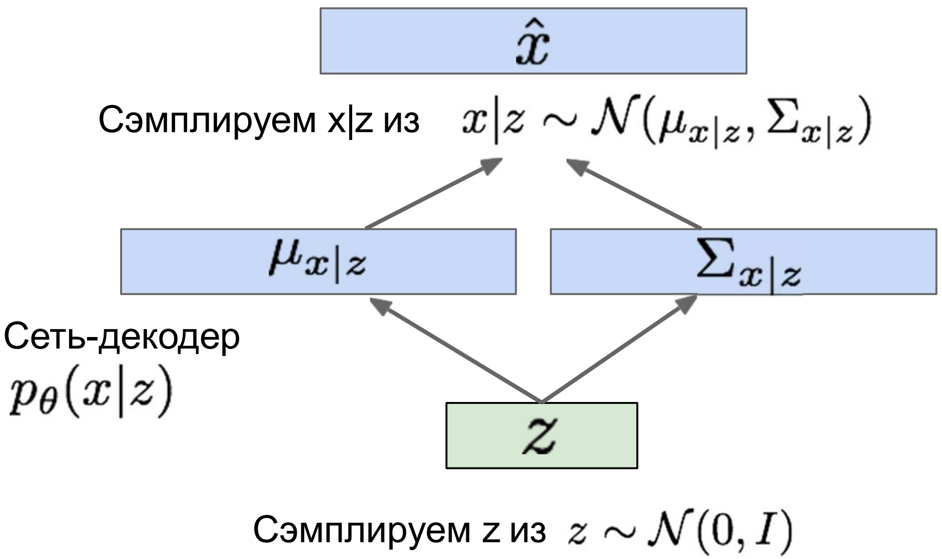
Используем сеть-декодер. Выбираем z из априорного распределения!

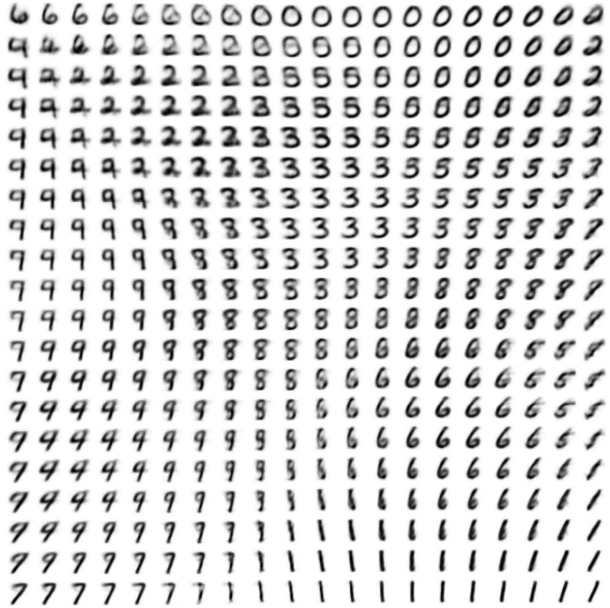


Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные автоэнкодеры: Порождение данных!

Используем сеть-декодер. Выбираем z из априорного распределения!

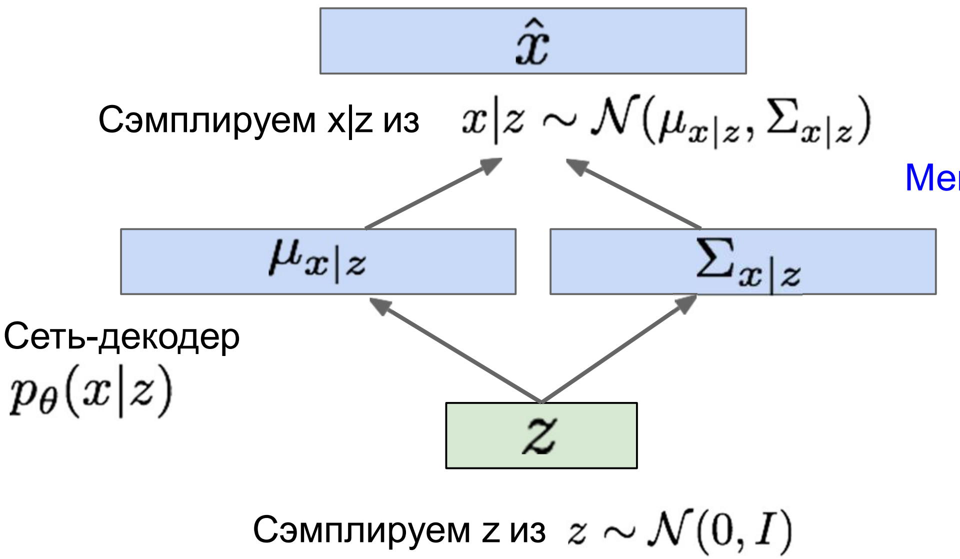


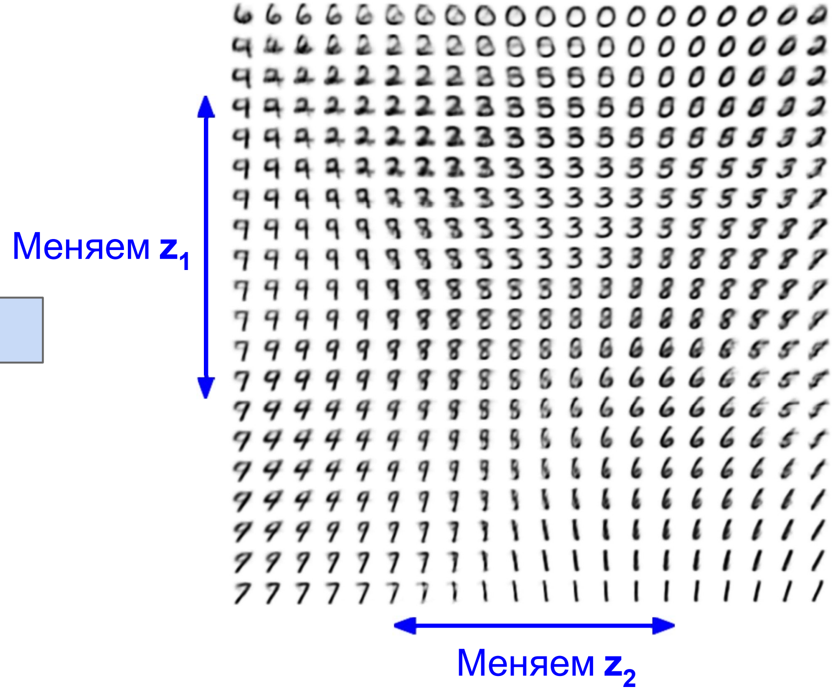


Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные автоэнкодеры: Порождение данных!

Используем сеть-декодер. Выбираем z из априорного распределения!





Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные автоэнкодеры: Порождение данных!

Диагональ **z**
=> независимые скрытые переменные

Разные размерности **z** кодируют интерпретируемые изменчивые признаки

Улыбчивость

Меняем $\mathbf{z}_1$

Меняем $\mathbf{z}_2$

Положение головы

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные автоэнкодеры: Порождение данных!

Диагональ **z**
=> независимые
скрытые
переменные

Разные размерности
z кодируют
интерпретируемые
изменчивые признаки

Улыбчивость

Меняем $\mathbf{z}_1$

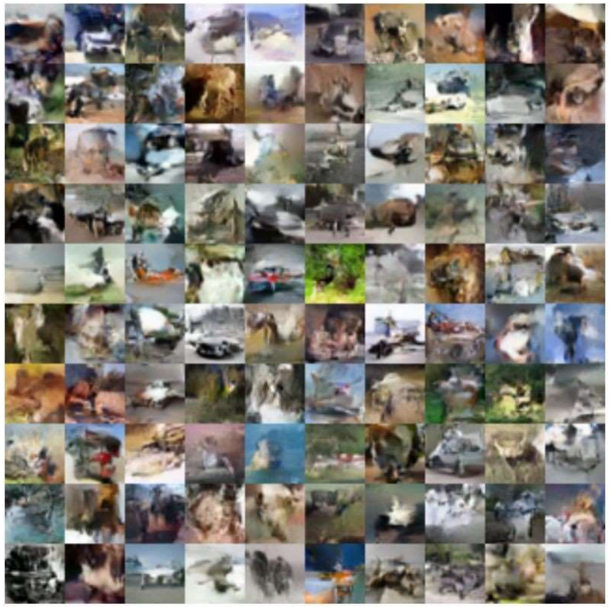
Меняем $\mathbf{z}_2$

Положение
головы

Также хорошее представление признаков
может быть вычислено при помощи $q_\phi(z|x)$!

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Вариационные автоэнкодеры: порождение данных!



32x32 CIFAR-10

Labeled Faces in the Wild

Skoltech
Skolkovo Institute of Science and Technology

Генеративно-состязательные сети (GAN)

Skoltech
Skolkovo Institute of Science and Technology

Итак...

VAE определили невычислимую плотность скрытого **z**:

$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$

Её нельзя оптимизировать напрямую, через производные или нижнюю границу правдоподобия

Что если мы откажемся от явного моделирования плотности и будем просто иметь возможность сэмплировать?

GANs: не работаем явно с плотностями!
Вместо этого рассмотрим теоретико-игровое приближение: учимся порождать выборки из обучающего распределения через игру двух игроков

Skoltech
Skolkovo Institute of Science and Technology

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

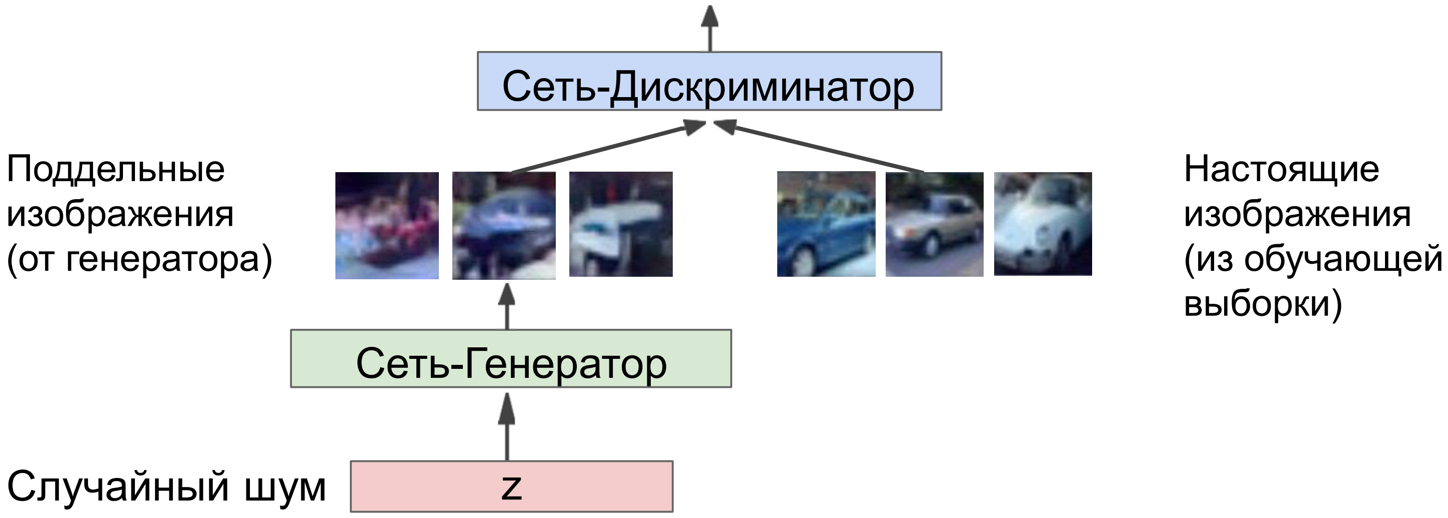
Обучаем GANы: игра двух игроков

Сеть-Генератор: пытается обмануть дискриминатор, генерируя реалистичные изображения
Сеть-Дискриминатор: пытается различать настоящие и поддельные изображения

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Обучаем GANы: игра двух игроков

Сеть-Генератор: пытается обмануть дискриминатор, генерируя реалистичные изображения
Сеть-Дискриминатор: пытается различать настоящие и поддельные изображения



Обучаем GANы: игра двух игроков

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Сеть-Генератор: пытается обмануть дискриминатор, генерируя реалистичные изображения
Сеть-Дискриминатор: пытается различать настоящие и поддельные изображения

Обучаются совместно в **минимаксной игре**

Целевая минимаксная функция:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Skoltech
Skolkovo Institute of Science and Technology

Обучаем GANы: игра двух игроков

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Сеть-Генератор: пытается обмануть дискриминатор, генерируя реалистичные изображения
Сеть-Дискриминатор: пытается различать настоящие и поддельные изображения

Обучаются совместно в **минимаксной игре**

Дискриминатор выдаёт правдоподобие настоящего изображения в интервале (0,1)

Целевая минимаксная функция:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x\sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Выход дискриминатора для настоящего изображения x}} + \mathbb{E}_{z\sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Выход дискриминатора Для сгенерированного изображения G(z)}}) \right]$$

Skoltech
Skolkovo Institute of Science and Technology

Обучаем GANы: игра двух игроков

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Сеть-Генератор: пытается обмануть дискриминатор, генерируя реалистичные изображения
Сеть-Дискриминатор: пытается различать настоящие и поддельные изображения

Обучаются совместно в **минимаксной игре**

Целевая минимаксная функция:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

- Дискриминатор (θ_d) **максимизирует целевую функцию** так, что D(x) близко к 1 (настоящее) и D(G(z)) близко к 0 (подделка)
- Генератор (θ_g) **минимизирует целевую функцию** так, что D(G(z)) близко к 1 (дискриминатор ошибается, думая, что подделка G(z) — настоящее изображение)

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Обучаем GANы: игра двух игроков

Минимаксная целевая функция:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Чередование:

1. **Градиентный подъём** дискриминатора

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Градиентный спуск** генератора

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

Skoltech
Skolkovo Institute of Science and Technology

Обучаем GANы: игра двух игроков

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Минимаксная целевая функция:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x\sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))\right]$$

Чередование:

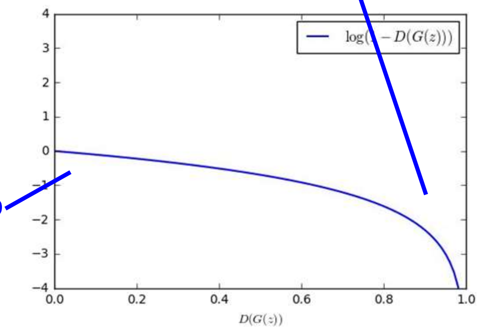
1. **Градиентный подъём** дискриминатора

$$\max_{\theta_d} \left[\mathbb{E}_{x\sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))\right]$$

2. **Градиентный спуск** генератора

$$\min_{\theta_g} \mathbb{E}_{z\sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

В градиенте есть регион с уже хорошими примерами



Когда пример больше похож на сгенерированный, фейковый, хотим с помощью него улучшить генератор. Но градиент в этом регионе относительно плоский

На практике оптимизация целевой функции генератора работает плохо!

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Обучаем GANы: игра двух игроков

Минимаксная целевая функция:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Чередование:

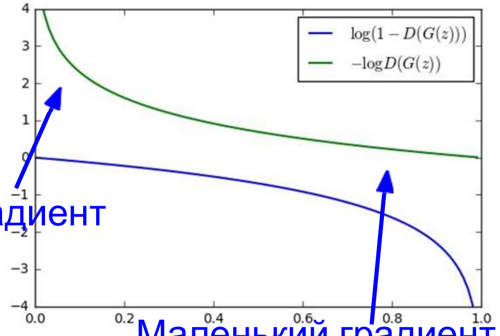
1. **Градиентный подъём** дискриминатора

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Вместо: Градиентный подъем** генератора, другая целевая функция

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

Вместо минимизации вероятности того, что дискриминатор будет работать правильно, теперь максимизируем вероятность того, что дискриминатор ошибается.
Та же цель обмануть дискриминатор, но теперь более высокий градиент сигнала для плохих образцов => работает намного лучше!



Skoltech
Skolkovo Institute of Science and Technology

Обучаем GANы: игра двух игроков

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Минимаксная целевая функция:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Кроме того: раздельное обучение может быть трудным и нестабильным. Поиск хороших функций потерь — область активных исследований

Чередование:

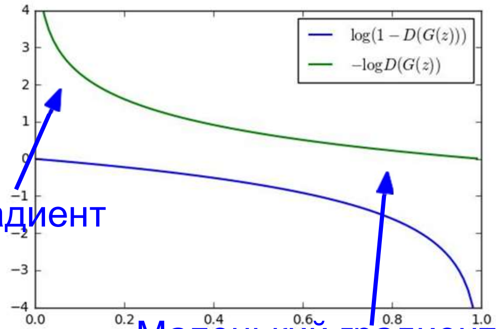
1. **Градиентный подъём** дискриминатора

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Вместо: Градиентный подъем** генератора, другая целевая функция

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

Вместо минимизации вероятности того, что дискриминатор будет работать правильно, теперь максимизируем вероятность того, что дискриминатор ошибается.
Та же цель обмануть дискриминатор, но теперь более высокий градиент сигнала для плохих образцов => работает намного лучше!



Большой градиент

Маленький градиент

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Обучаем GANы: игра двух игроков

Итого: алгоритм обучения GAN

for number of training iterations **do**

  for k steps **do**

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

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by ascending its stochastic gradient (improved objective):

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Обучаем GANы: игра двух игроков

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Итого: алгоритм обучения GAN

Использование k=1 более стабильно, k > 1 – не лучший выбор. Недавние работы (e.g. Wasserstein GAN) облегчили эту проблему, больше стабильности!

for number of training iterations **do**

  for $\boxed{k \text{ steps}}$ **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:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

  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 ascending its stochastic gradient (improved objective):

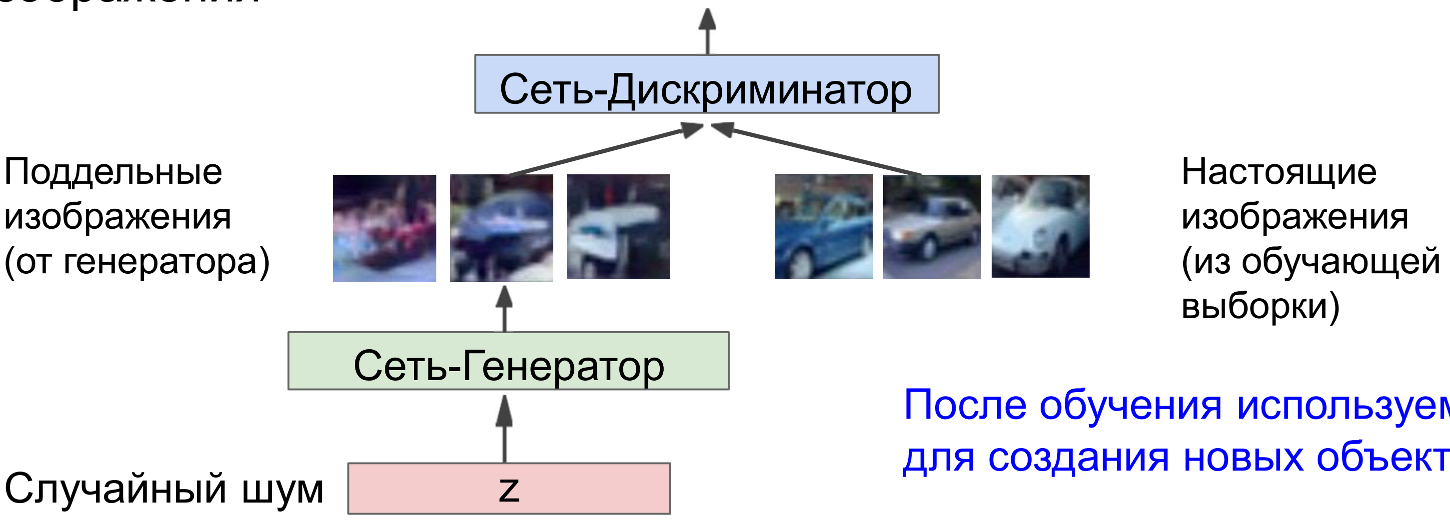
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Обучаем GANы: игра двух игроков

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Сеть-Дискриминатор: пытается различать настоящие и поддельные изображения
Сеть-Генератор: пытается обмануть дискриминатор, генерируя реалистичные изображения

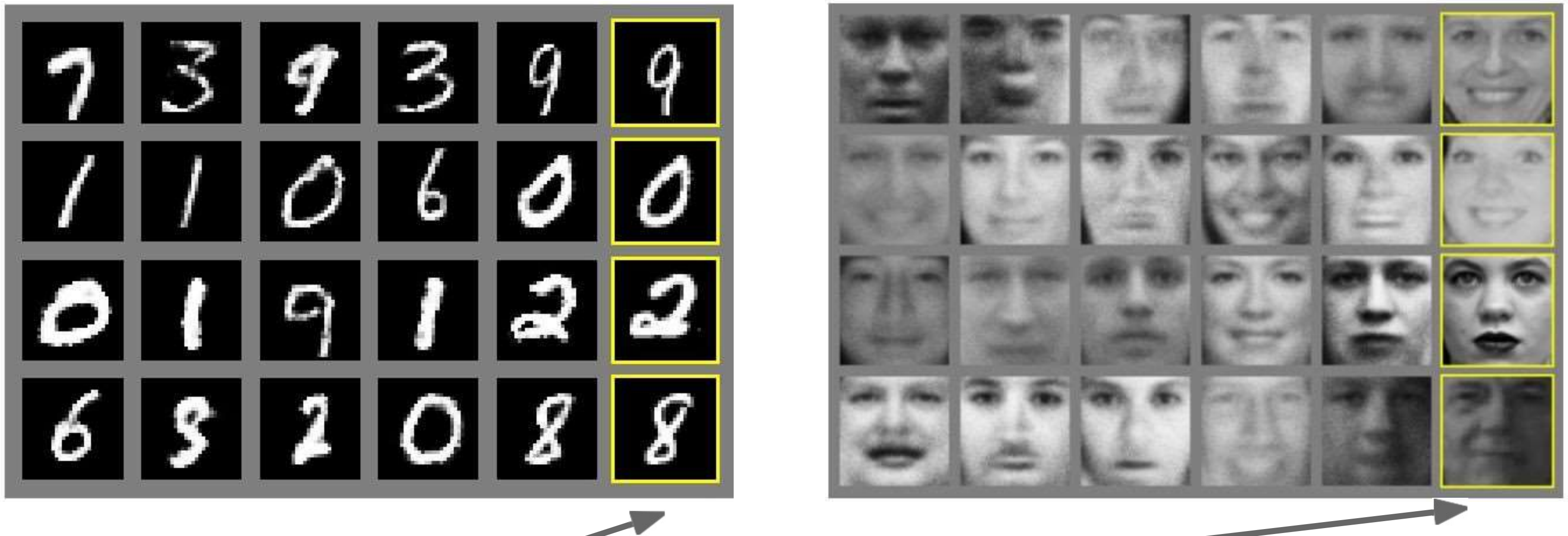


После обучения используем генератор для создания новых объектов

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Генеративные состязательные сети

Порожденные изображения



Ближайший сосед из обучающей выборки

Skoltech
Skolkovo Institute of Science and Technology

GAN: сверточные архитектуры

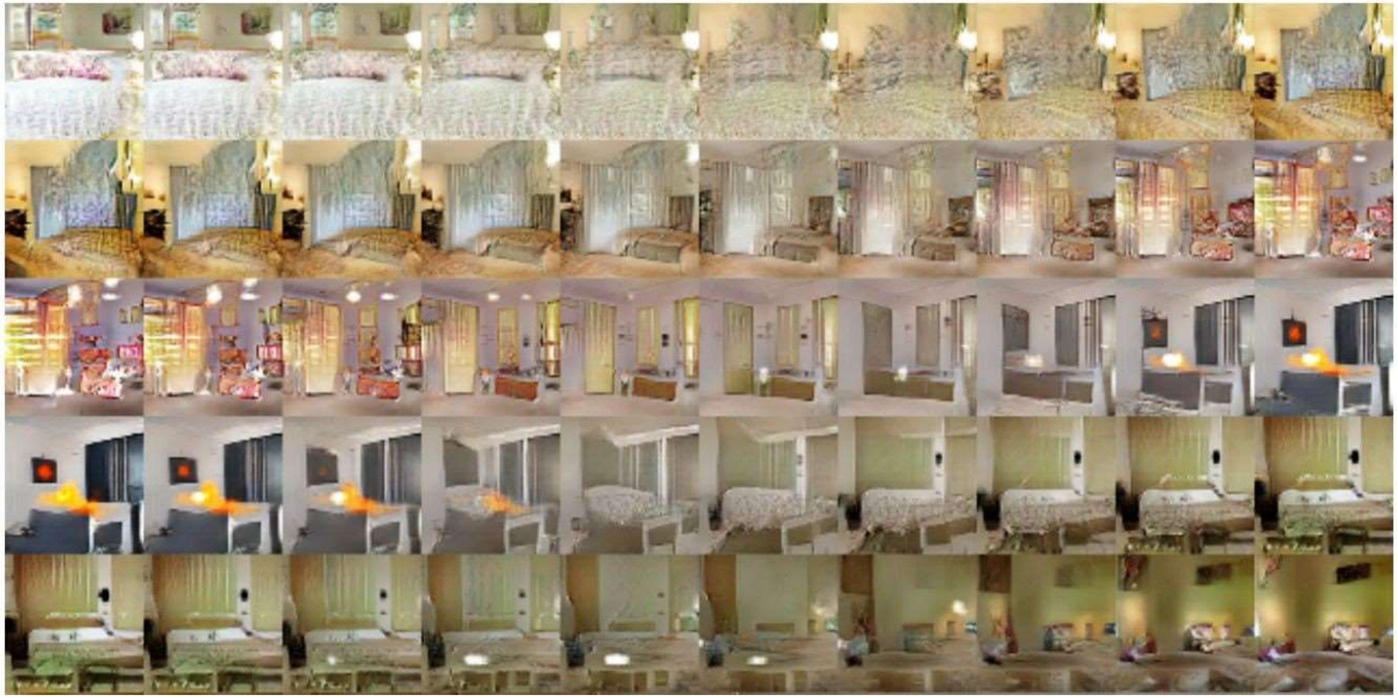
Созданные картинки выглядят впечатляюще!

Radford et al, ICLR 2016

Skoltech
Skolkovo Institute of Science and Technology

GAN: сверточные архитектуры

Интерполяция случайных точек скрытого пространства



Radford et al,
ICLR 2016

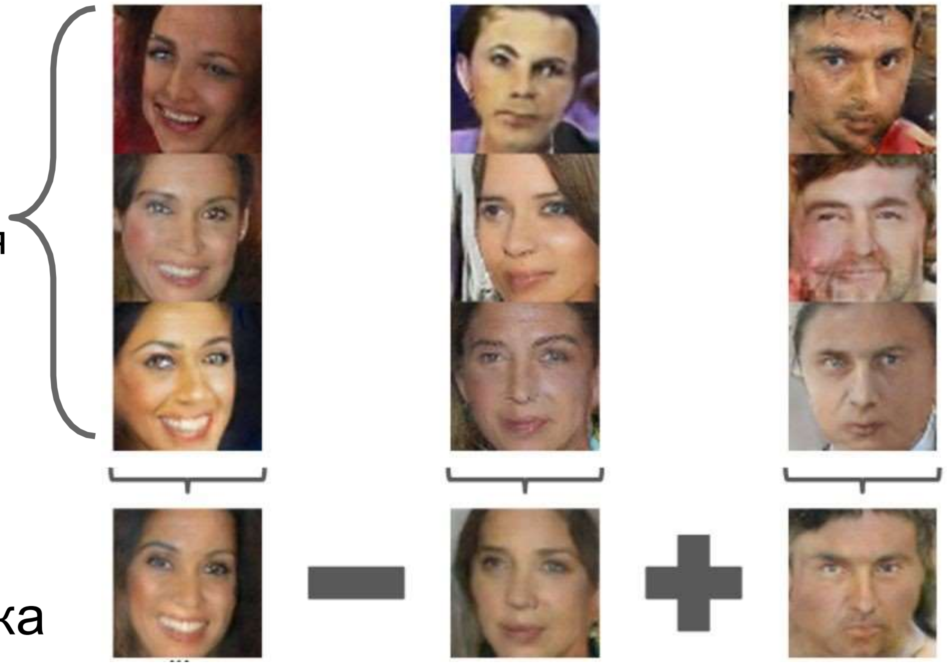
GAN: Математика Интерпретируемых Векторов

Radford et al, ICLR 2016

Улыбающаяся женщина | Спокойная женщина | Спокойный мужчина

Созданные изображения

Средний вектор Z, арифметика



Skoltech
Skolkovo Institute of Science and Technology

GAN: Математика Интерпретируемых Векторов

Radford et al, ICLR 2016

Skoltech
Skolkovo Institute of Science and Technology

GAN: Математика Интерпретируемых Векторов

Radford et al, ICLR 2016

GAN: Математика Интерпретируемых Векторов

Radford et al, ICLR 2016

Skoltech
Skolkovo Institute of Science and Technology

“Зоопарк GAN ”

- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

Skoltech
Skolkovo Institute of Science and Technology

Итог

Генеративные модели

- Вариационные Автоэнкодеры (VAEs)

Оптимизируйте вариационную нижнюю границу правдоподобия. Полезные латентные представления, явный вывод. Но текущий уровень моделей не самый лучший с точки зрения качества.

- Генеративные состязательные сети (GANs)

Игровой подход, отличные результаты! Но трудно и нестабильно обучаем.

- Состязательные Автоэнкодеры (AAEs)

Совмещаем VAE с состязательным обучением GAN