

Лекция 14

Генеративные модели

Машинное обучение
Сергей Муравьёв

04.12.2020

План лекции

- Задача генерации новых объектов
- PixelCNN и PixelRNN
- Вариационные автокодировщики
- Генеративно-состязательные модели
- Интересные идеи в GAN-ах

- В презентации используются материалы:
 - F.F. Li et al.' курс "Convolutional Neural Networks for Visual Recognition"
 - А.С. Артамонов "Image and video analysis"
- Слайды доступны: shorturl.at/ltVZ3
- Видео доступны: shorturl.at/hjyAX

План лекции

- Задача генерации новых объектов
- PixelCNN и PixelRNN
- Вариационные автокодировщики
- Генеративно-состязательные модели
- Интересные идеи в GAN-ах

Генерация новых объектов

Какой это тип задачи?

- С учителем
- Без учителя
- Частичное с учителем
- С подкреплением

Определение препятствий для создания новых объектов

Какой это тип задачи?

- Это зависит от того, как мы определяем новизну объекта и как мы измеряем степень принадлежности объекта к желаемому множеству
- Ничто не мешает создать набор эмпирических правил, при помощи которых мы будем создавать новые объекты
- Тогда не нужно никакого машинного обучения

В контексте машинного обучения

Мы хотим, чтобы

- сгенерированное изображение было изображением
- было изображением из предметной области:
 - все детали изображения соответствовали бы предметной области
 - взаимодействие между деталями было бы реалистично
 - ...

Препятствие в контексте машинного обучения

- Учитывая заданный образец, учёт новых объектов маловероятен, потому что мы никогда не видели их в образце.

Препятствие в контексте машинного обучения

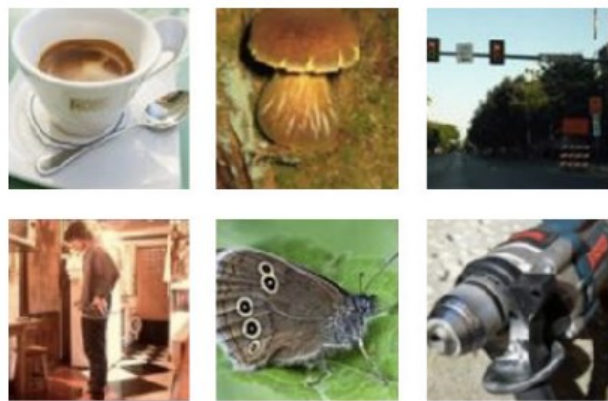
- Учитывая заданный образец, учёт новых объектов маловероятен, потому что мы никогда не видели их в образце.
- Мы хотим создать объект, который правдоподобен в отношении некоторой скрытой структуры объектов.

Задача генерации — задача обучения без учителя

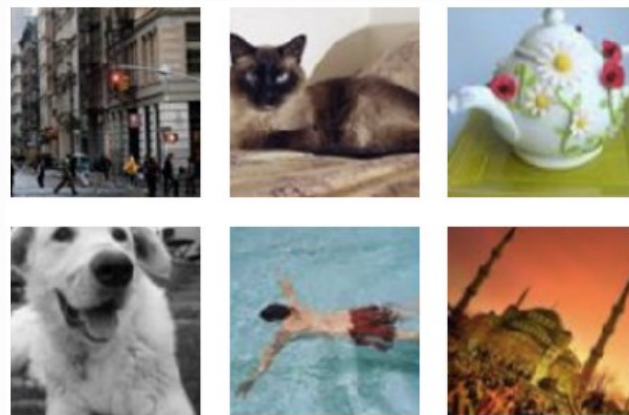
- Учитывая заданный образец, учёт новых объектов маловероятен, потому что мы никогда не видели их в образце.
- Мы хотим создать объект, который правдоподобен в отношении некоторой скрытой структуры объектов.
- Это задача обучения без учителя! Нам нужно изучить **скрытую структуру** объектов, изучить **распределение по ним**, а затем просто выбрать новый объект из этого распределения.

Задача генерации

По заданной выборке требуется сгенерировать новые образцы из того же распределения



Обучающая выборка $p_{\text{data}}(x)$



Сгенерированная
выборка $p_{\text{model}}(x)$

$p_{\text{model}}(x)$ должно быть похоже на $p_{\text{data}}(x)$

Как измерить сходство распределений?

Дивергенция Кульбака-Лейблера:

Дана выборка P , распределенная согласно p ,
и Q , распределенная согласно q ,

$$D_{\text{KL}}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} d(x) .$$

Второе название — **относительная энтропия** P по отношению к Q .

Важно, что она несимметрична, поэтому не является расстоянием

Сложности

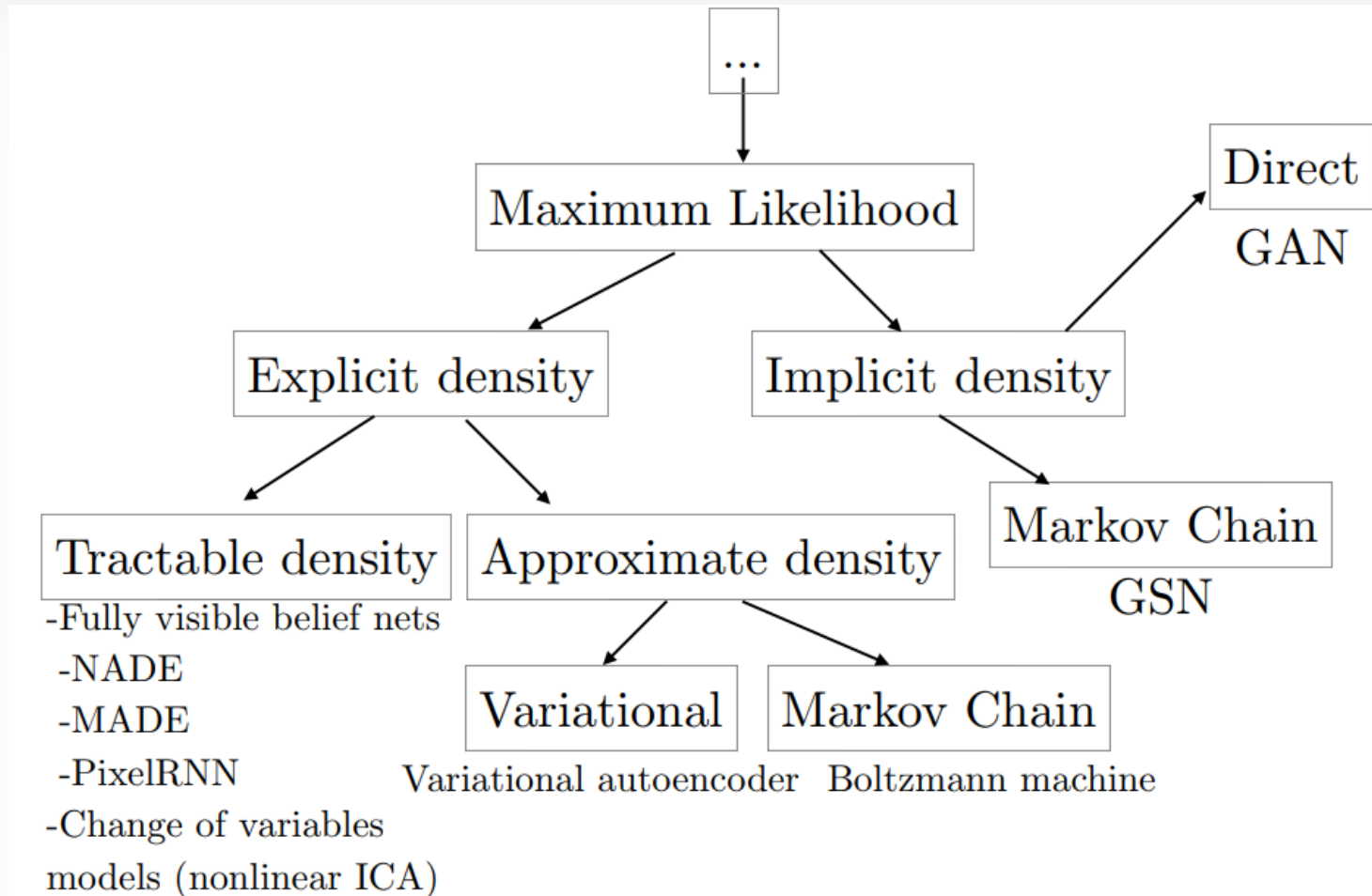
Оценка плотности — ключевая проблема в обучении без учителя.

Два основных способа:

- Непосредственная оценка плотности: непосредственное определение и решение для $p_{\text{model}}(x)$
- Неявная оценка плотности: модель, которая может выполнять выборку из $p_{\text{model}}(x)$ без ее непосредственного определения

Чем выше размерность пространства, тем сложнее восстанавливать многомерные распределения

Таксономия генеративных моделей



План лекции

- Задача генерации новых объектов
- PixelCNN и PixelRNN
- Вариационные автокодировщики
- Генеративно-состязательные модели
- Интересные идеи в GAN-ах

Глубокая сеть доверия

Явная модель плотности

Используйте цепное правило, чтобы разложить вероятность изображения x на произведение одномерных распределений:

$$p(x) = \prod_{i=1}^n p(x_i | x_1, \dots, x_{i-1})$$

probability of objects (images) conditional probability of observing features (pixels)

После этого мы увеличиваем правдоподобие наблюдаемых данных.

Глубокая сеть доверия

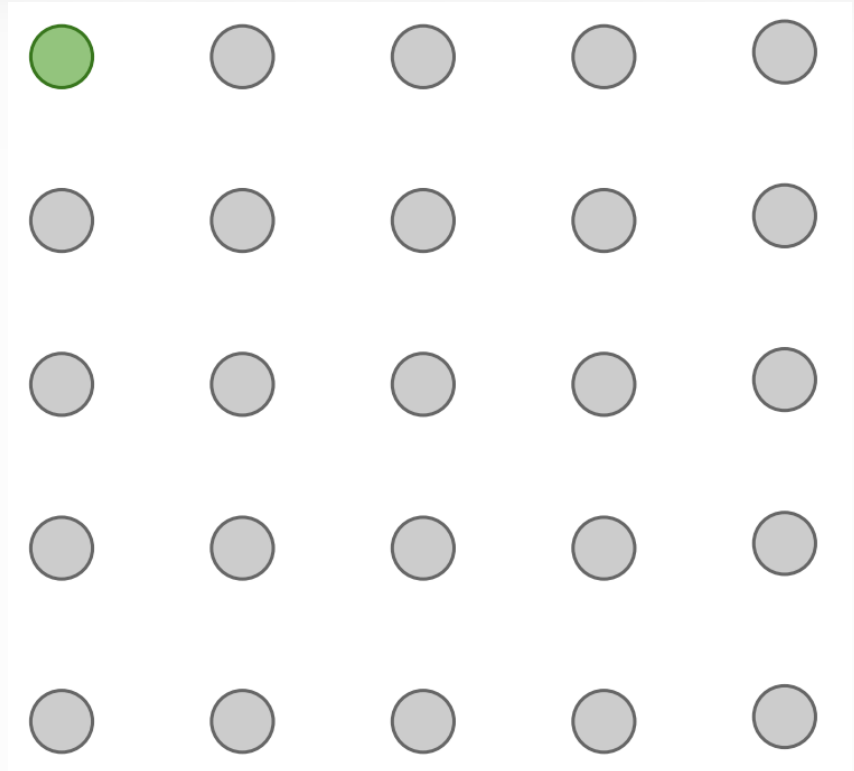
Мы можем выразить вероятности с помощью нейронных сетей

Для этого нам нужны определенные вероятности «предыдущих» пикселей.

PixelRNN

Генерация пикселей
изображения, начиная
с левого угла

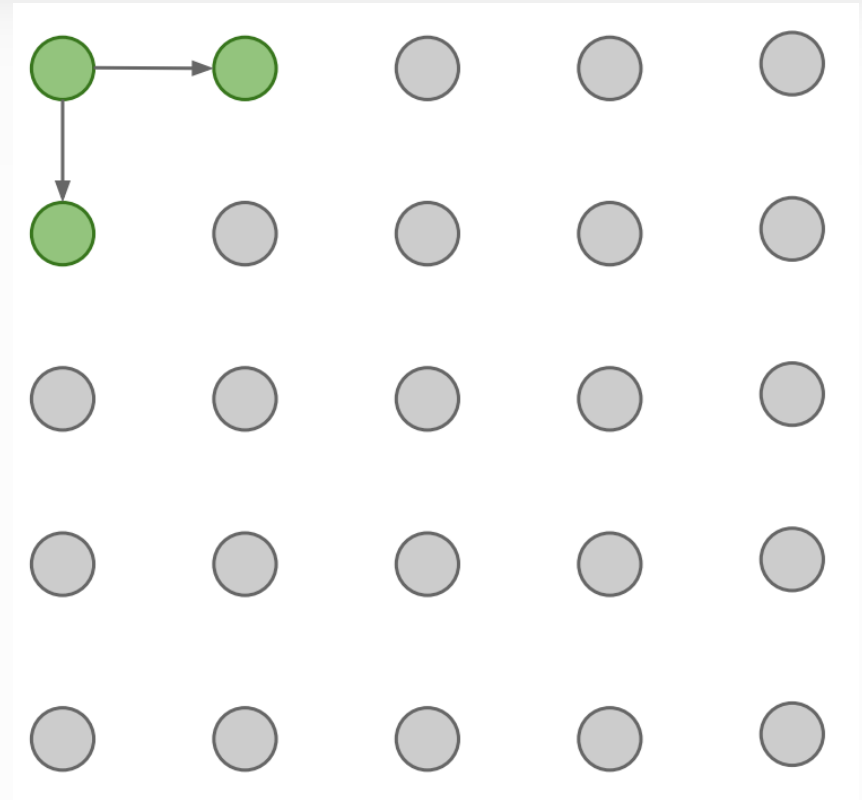
Зависимость от
предыдущих пикселей
моделируется с
помощью RNN (LSTM).



PixelRNN

Генерация пикселей изображения, начиная с левого угла

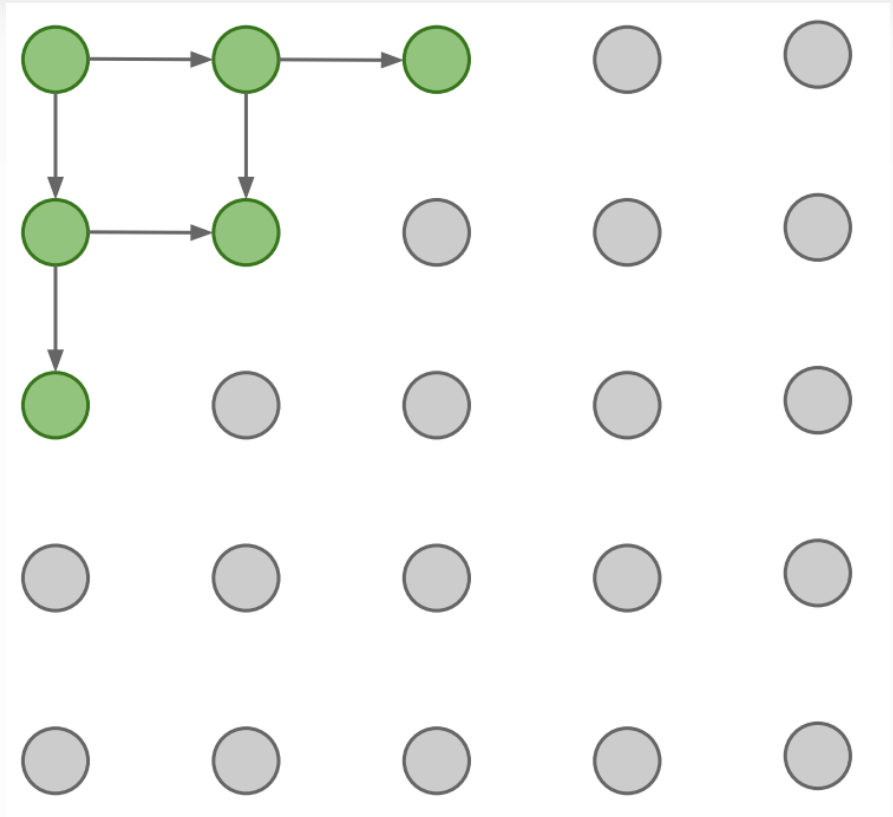
Зависимость от предыдущих пикселей моделируется с помощью RNN (LSTM).



PixelRNN

Генерация пикселей изображения, начиная с левого угла

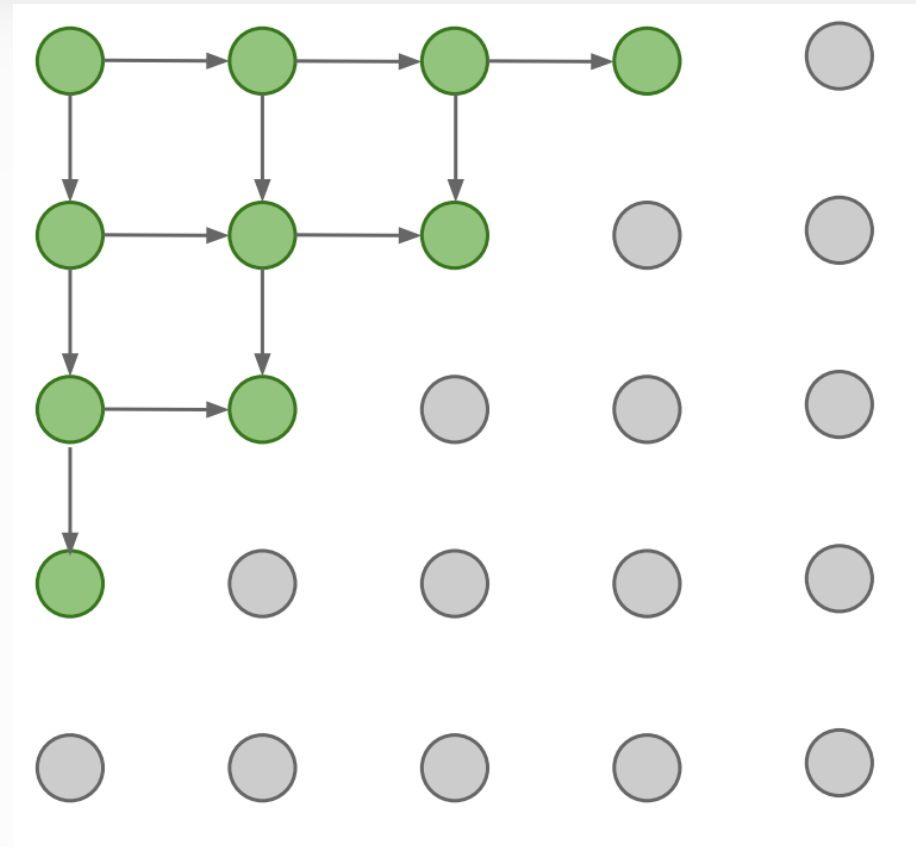
Зависимость от предыдущих пикселей моделируется с помощью RNN (LSTM).



PixelRNN

Генерация пикселей
изображения, начиная
с левого угла

Зависимость от
предыдущих пикселей
моделируется с
помощью RNN (LSTM).



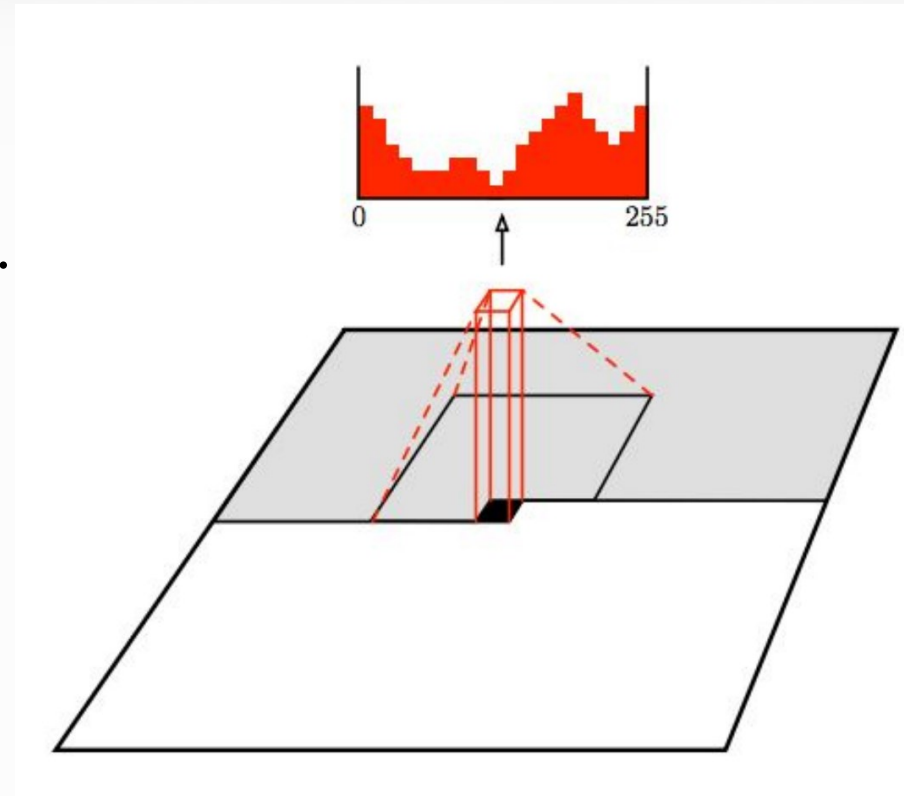
PixelCNN

Вычисление RNN идет слишком медленно.

Вместо использования RNN мы можем использовать CNN.

По-прежнему генерируем пиксели изображения, смотрящие из угла

Зависимость от предыдущих пикселей моделируется с использованием CNN по контекстной области.



PixelCNN

Обучается быстрее PixelRNN.

Генерация должна продолжаться последовательно, что замедляет ее.

Анализ Pixel*NN

Преимущества:

- Может явно вычислить правдоподобие $p(x)$
- Явное правдоподобие обучающих данных дает хорошую метрику оценки
- Хорошие образцы

Недостатки:

- Последовательная генерация медленная

Повышение производительности PixelCNN

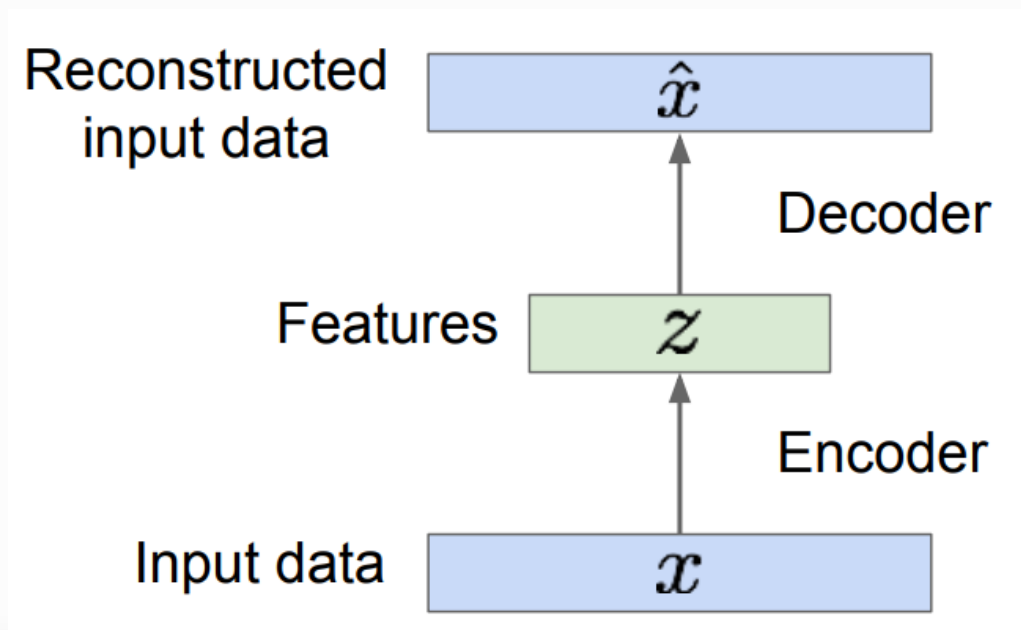
- Закрытые сверточные слои (Gated convolutional layers)
- Укороченные соединения (Short-cut)
- Дискретные логистические потери
- «Мультимасштаб»
- Трюки в процессе обучения

План лекции

- Задача генерации новых объектов
- PixelCNN и PixelRNN
- Вариационные автокодировщики
- Генеративно-состязательные модели
- Интересные идеи в GAN-ах

Автокодировщики (напоминание)

Автокодировщик (autoencoder) — глубокая нейронная сеть, способная строить низкоразмерные представления данных за счет нелинейной трансформации.



Основная идея

Вместо того, чтобы использовать некоторые предположения о том, как должна выглядеть структура вероятностной модели, мы определяем неразрешимую функцию плотности с некоторой скрытой переменной z :

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

Скрытые переменные

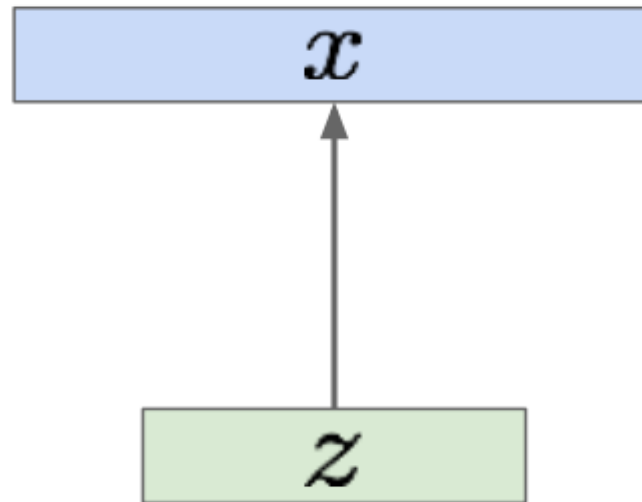
Предположим, что данные обучения генерируются в зависимости от некоторого латентного z , устроенного достаточно простым образом

Sample from
true conditional

$$p_{\theta^*}(x \mid z^{(i)})$$

Sample from
true prior

$$p_{\theta^*}(z)$$



Представление $p(x|z)$

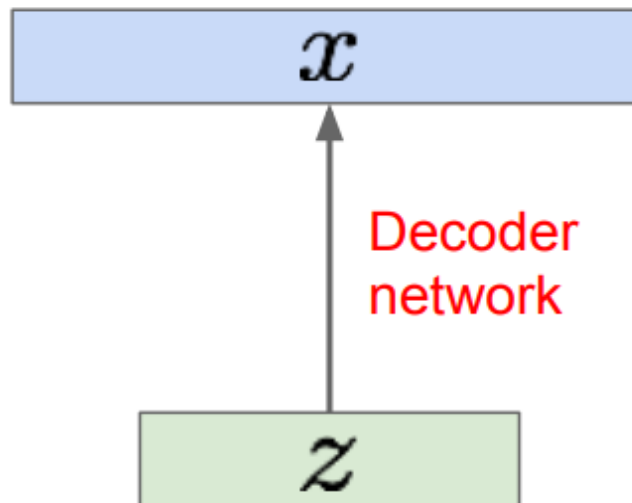
Условное $p(x | z)$ является сложным, восстановим его при помощи нейронной сети

Sample from
true conditional

$$p_{\theta^*}(x | z^{(i)})$$

Sample from
true prior

$$p_{\theta^*}(z)$$



Параметры обучения

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

$$\theta^* = \arg \min_{\theta} -\log \int p_{\theta}(x|z)p_{\theta}(z)dz$$

В чём состоит проблема?

Параметры обучения

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

$$\theta^* = \arg \min_{\theta} -\log \int p_{\theta}(x|z)p_{\theta}(z)dz$$

$\int p_{\theta}(x|z)p_{\theta}(z)dz$ трудно разрешим!

Параметры обучения

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

$$\theta^* = \arg \max_{\theta} \log \int p_{\theta}(x|z)p_{\theta}(z)dz$$

$\int p_{\theta}(x|z)p_{\theta}(z)dz$ трудно разрешим!

Вероятность апостериорных данных также нельзя выразить!

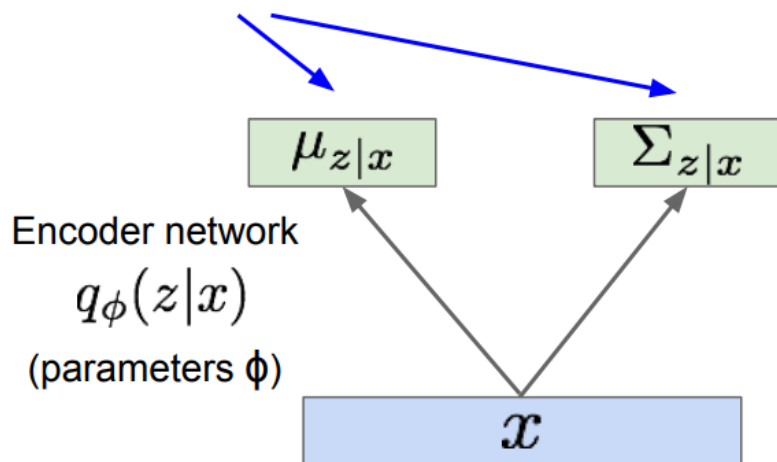
Сеть кодировщика

Идея: добавить сеть
кодировщика $q_\phi(z|x)$,
аппроксимирующую $p_\phi(z|x)$

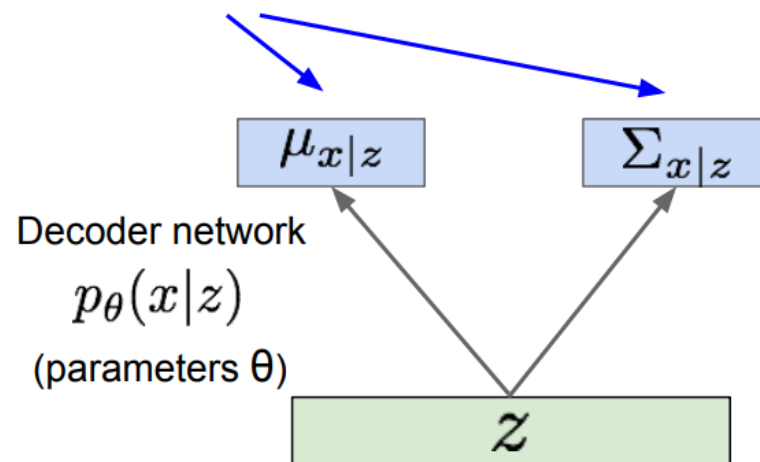
Кодировщик и декодировщик

Кодировщик и декодировщик являются вероятностными, оба предполагают гиперпараметры распределения (скажем, по Гауссу)

Mean and (diagonal) covariance of $\mathbf{z} | \mathbf{x}$

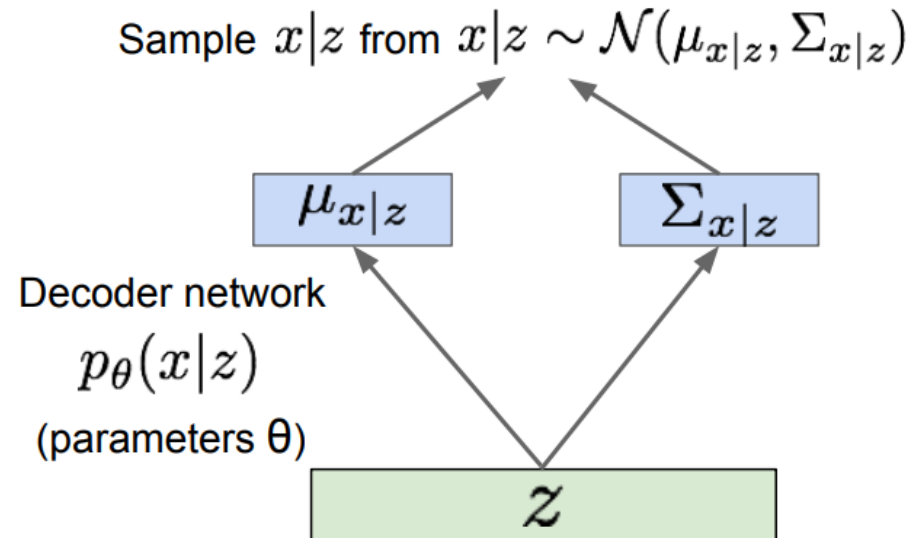
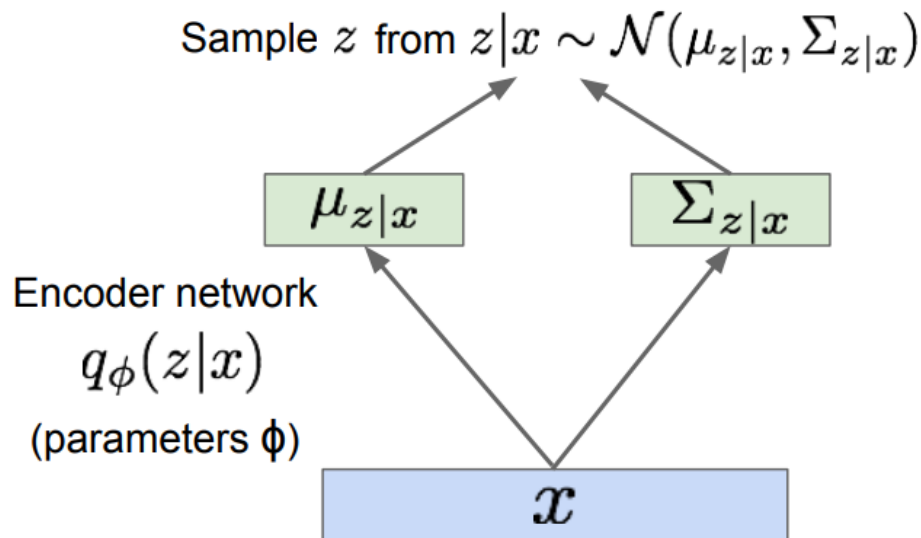


Mean and (diagonal) covariance of $\mathbf{x} | \mathbf{z}$



Сэмплирование с кодировщиком и декодировщиком

Мы можем выбрать z с помощью кодировщика и $x \mid z$ с помощью декодировщика.



Возвращаясь к правдоподобию

$$\begin{aligned}\theta^* &= \arg \max_{\theta} \log p_{\theta}(x) \\ \log p_{\theta}(x^{(i)}) &= \mathbb{E}_{z \sim q_{\phi}(z|x^{(i)})} [\log p_{\theta}(x^{(i)})] = \\ &= \mathbb{E}_z \left[\log \frac{p_{\theta}(x^{(i)}|z)p_{\theta}(z)}{p_{\theta}(z|x^{(i)})} \right] = \\ &= \mathbb{E}_z \left[\log \frac{p_{\theta}(x^{(i)}|z)p_{\theta}(z)}{p_{\theta}(z|x^{(i)})} \frac{q_{\phi}(z|x^{(i)})}{q_{\phi}(z|x^{(i)})} \right] = \\ &= \mathbb{E}_z [\log p_{\theta}(x^{(i)}|z)] - \mathbb{E}_z \left[\log \frac{q_{\phi}(z|x^{(i)})}{p_{\theta}(z)} \right] + \mathbb{E}_z \left[\log \frac{q_{\phi}(z|x^{(i)})}{p_{\theta}(z|x^{(i)})} \right] = \\ &= \mathbb{E}_z [\log p_{\theta}(x^{(i)}|z)] - D_{KL} \left(q_{\phi}(z|x^{(i)}) || p_{\theta}(z) \right) \\ &\quad + D_{KL} \left(q_{\phi}(z|x^{(i)}) || p_{\theta}(z|x^{(i)}) \right)\end{aligned}$$

Нижние границы

$$\mathbb{E}_z [\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)}) || p_\theta(z)) + D_{KL}(q_\phi(z|x^{(i)}) || p_\theta(z|x^{(i)}))$$

↑
Decoder network gives $p_\theta(x|z)$, can compute estimate of this term through sampling.

↑
This KL term (between Gaussians for encoder and z prior) has nice closed-form solution!

↑
 $p_\theta(z|x)$ intractable (saw earlier), can't compute this KL term :(But we know KL divergence always ≥ 0 .

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

Tractable lower bound which we can take gradient of and optimize! ($p_\theta(x|z)$ differentiable, KL term differentiable)

Обучение VAE

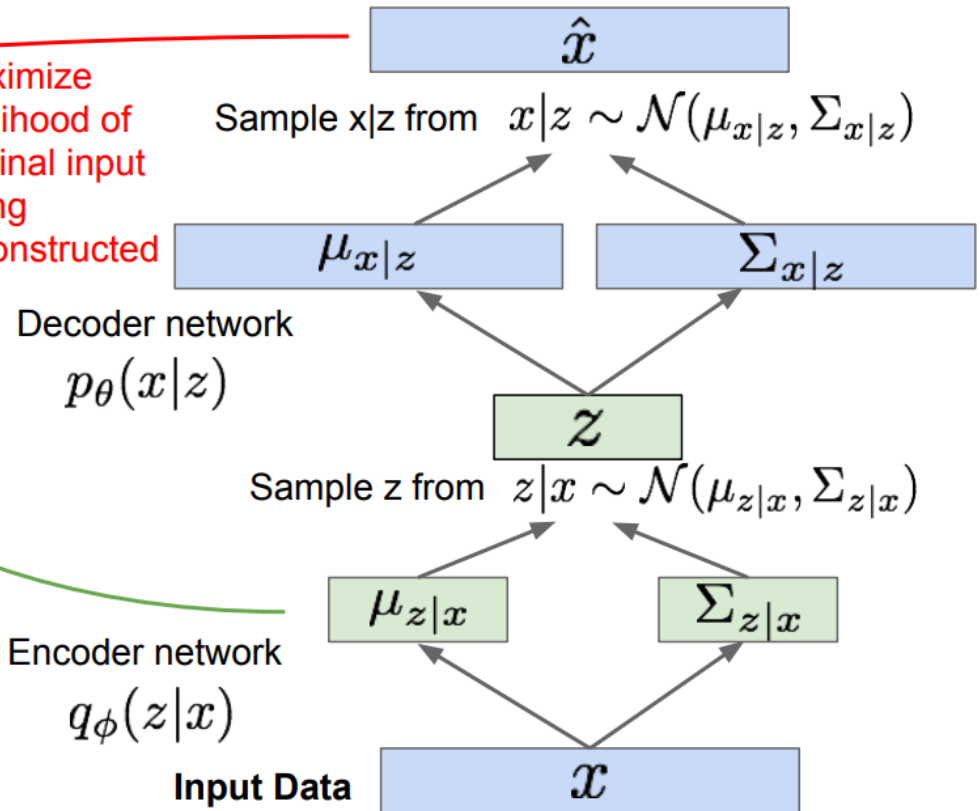
Putting it all together: maximizing the likelihood lower bound

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

Make approximate posterior distribution close to prior

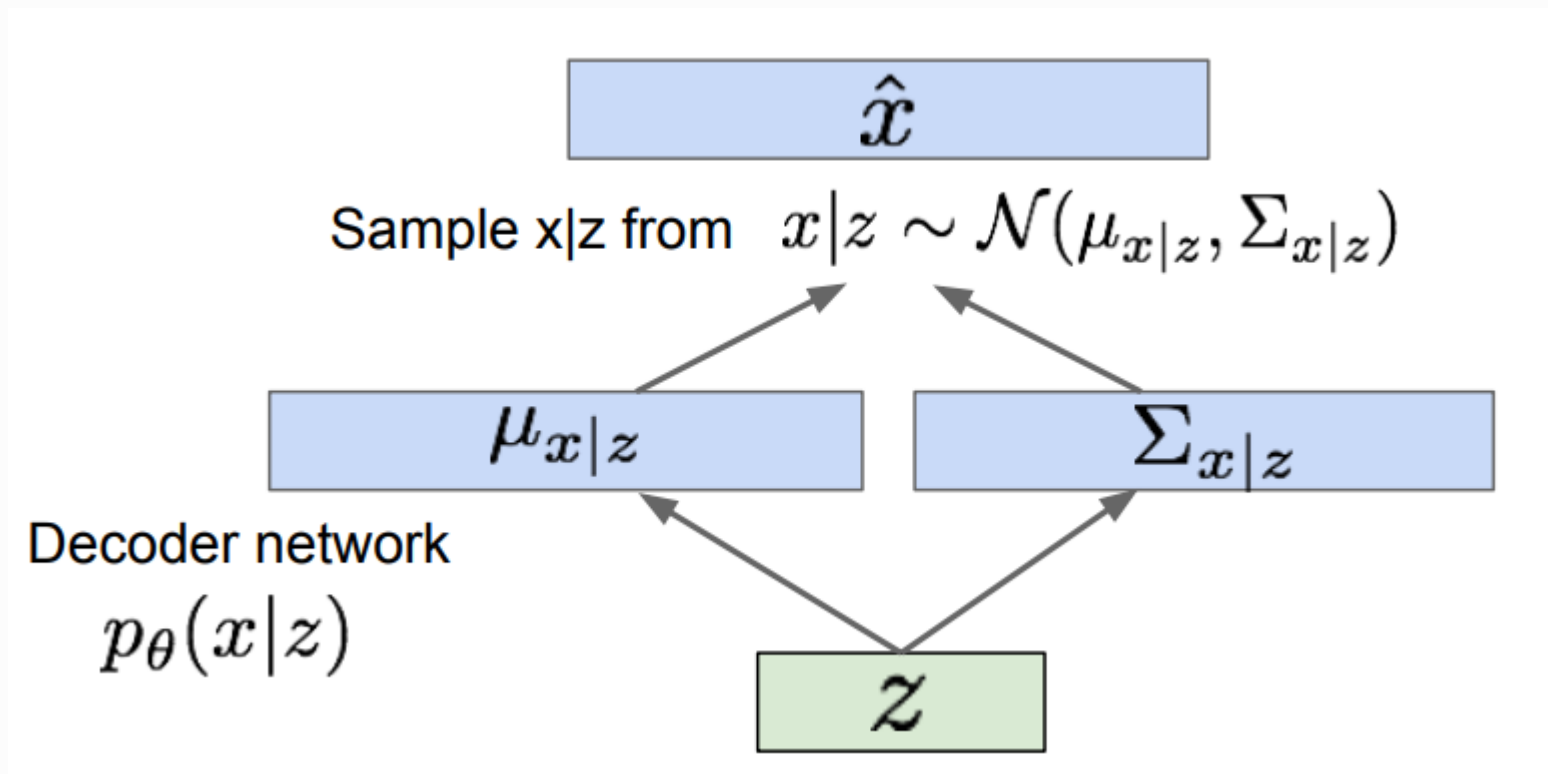
For every minibatch of input data: compute this forward pass, and then backprop!

Maximize likelihood of original input being reconstructed



Создание данных с помощью VAE

Простой пример с декодером



Анализ VAE

Преимущества:

- Принципиальный подход к генеративным моделям
- Позволяет сделать вывод $q(z | x)$, может быть полезным представлением функции для других задач

Недостатки:

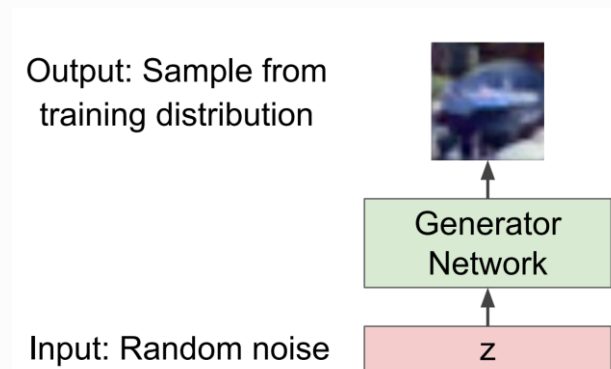
- Максимизация нижней границы вероятности - не такая хорошая оценка, как PixelRNN / PixelCNN.
- Образцы более размытые и более низкого качества по сравнению с современными (GAN)

План лекции

- Задача генерации новых объектов
- PixelCNN и PixelRNN
- Вариационные автокодировщики
- **Генеративно-состязательные модели**
- Интересные идеи в GAN-ах

Основные идеи

- Не будем пытаться непосредственно представить распределения
- Будем обучать преобразование в объекты из случайного шума
- Будем использовать теоретико-игровой подход

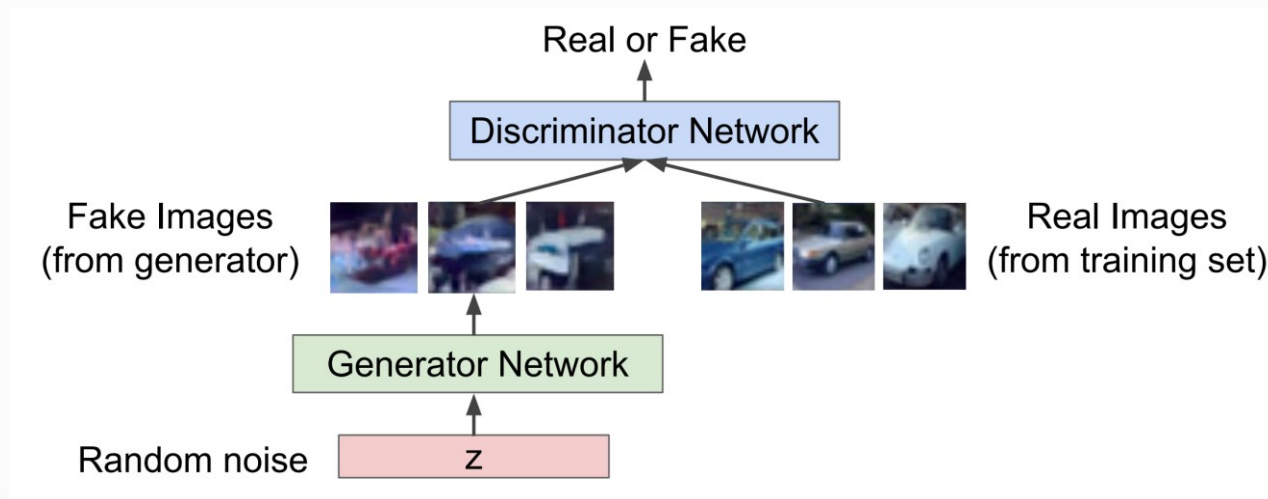


Генератор vs Дискриминатор

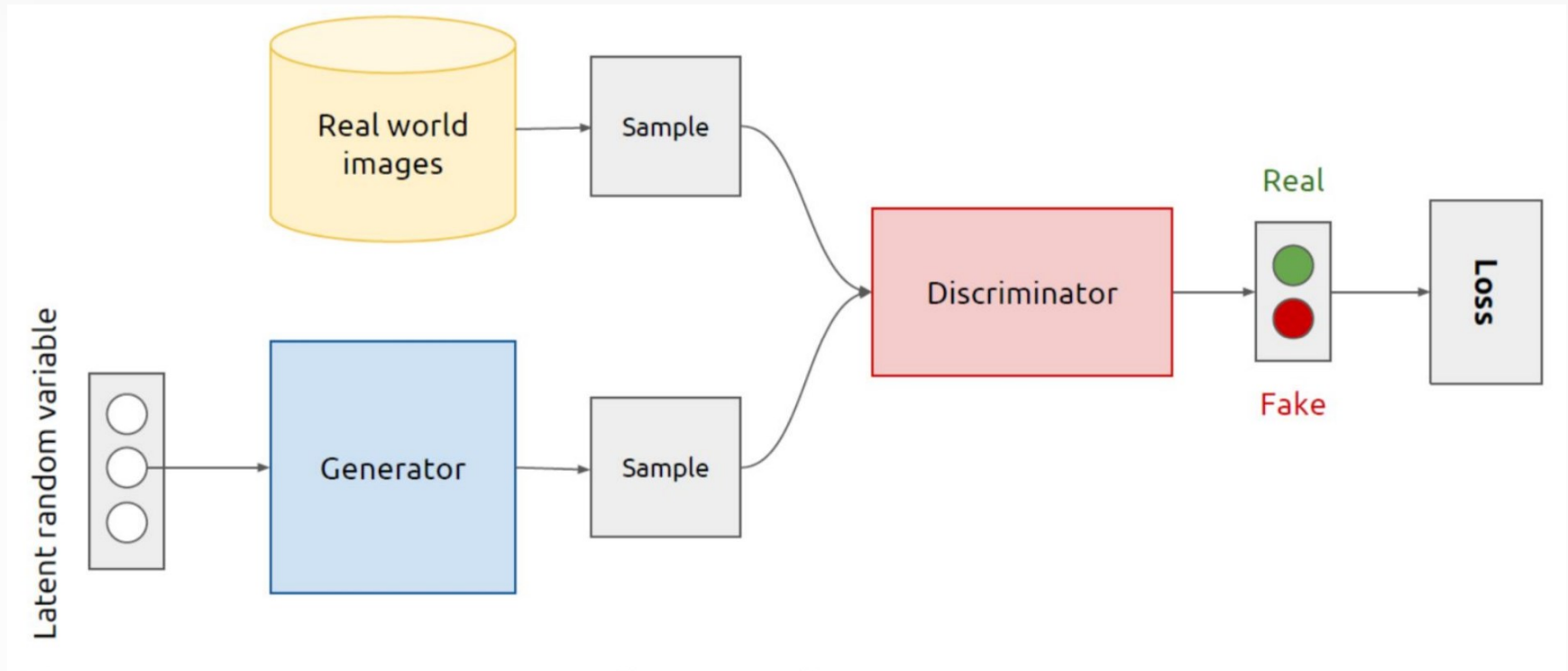
Генератор — сеть, которая будет генерировать объекты, пытаясь «обмануть» дискриминатор

Дискриминатор — сеть, которая будет отличать сгенерированные объекты от настоящих

Обучение будет построено как противостояние двух этих «игроков»



Обучение GAN-ов



Минимаксная игра

Можно обучать совместно в постановке минимаксной игры

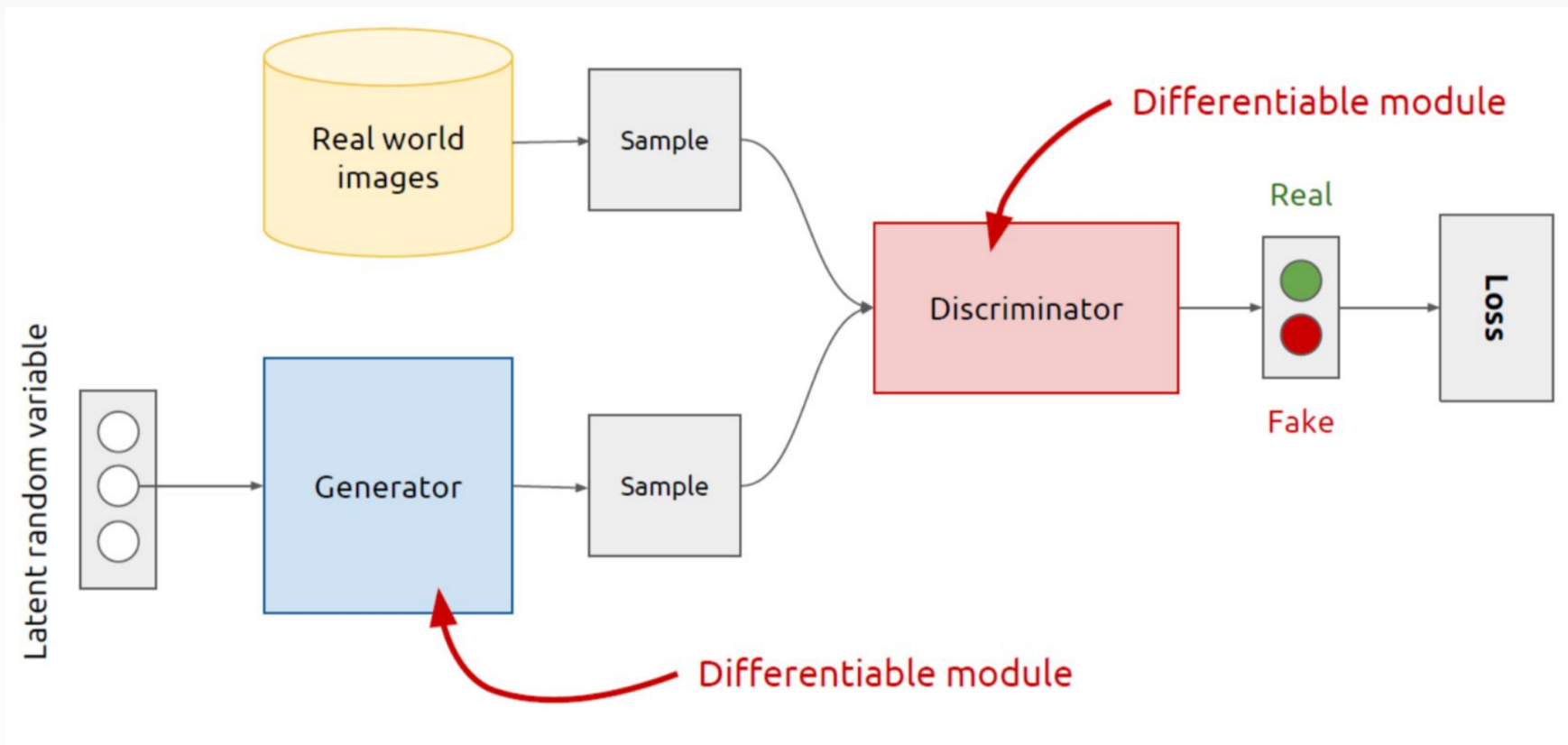
Минимаксная целевая функция:

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

где D_{θ_d} — дискриминатор с параметрами θ_d
(пытается максимизировать целевую функцию так, и сделать так, чтобы $D(x)$ был близок к 1 (настоящий) и $D(G(z))$ был близок к 0 (сгенерированный))

и G_{θ_g} — генератор с параметрами θ_g
(пытается минимизировать целевую функцию и сделать $D(G(z))$ близким к 1)

Обучение GAN-ов обратным распространением ошибки



Поочередное обучение

Поочередно:

1. Градиентный подъем по дискриминатору

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

2. Градиентный спуск по генератору

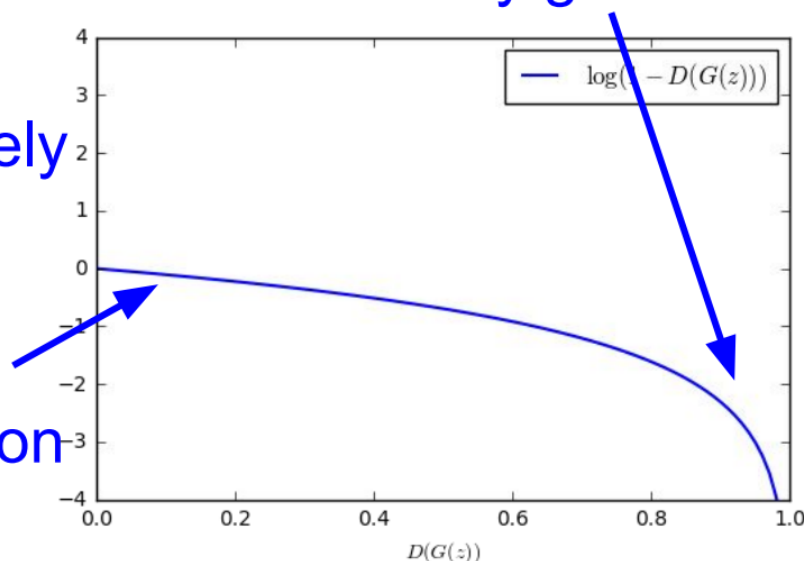
$$\min_{\theta_g} E_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

Поочередное обучение

На практике это не работает, потому что они смотрят в одном направлении.

When sample is likely fake, want to learn from it to improve generator. But gradient in this region is relatively flat!

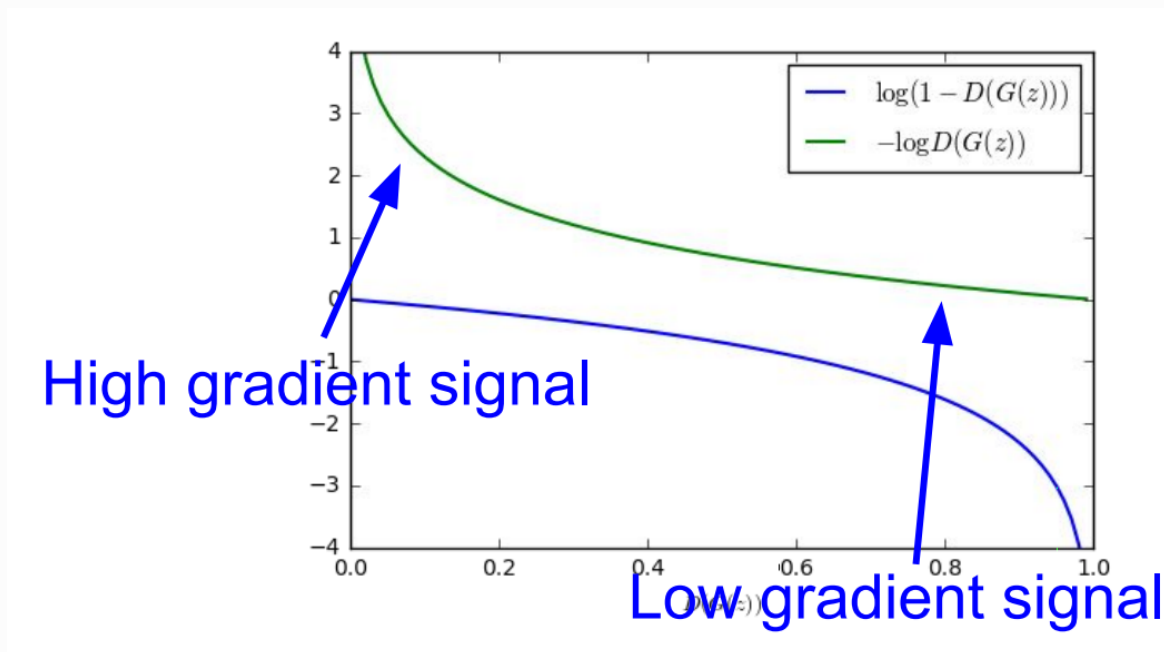
Gradient signal dominated by region where sample is already good



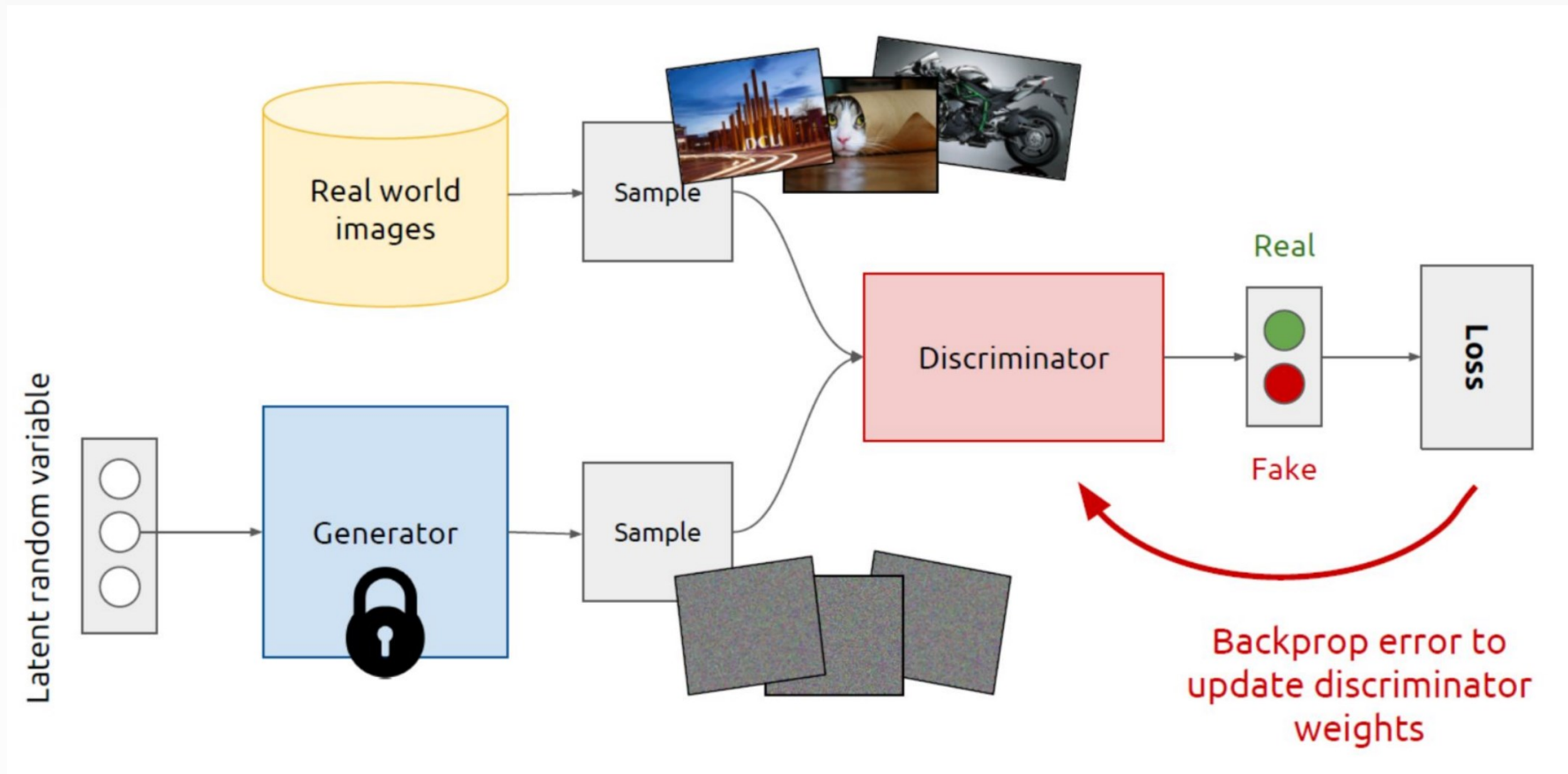
Изменение обучения генератора

Вместо этого попробуем **градиентный подъем** на генераторе

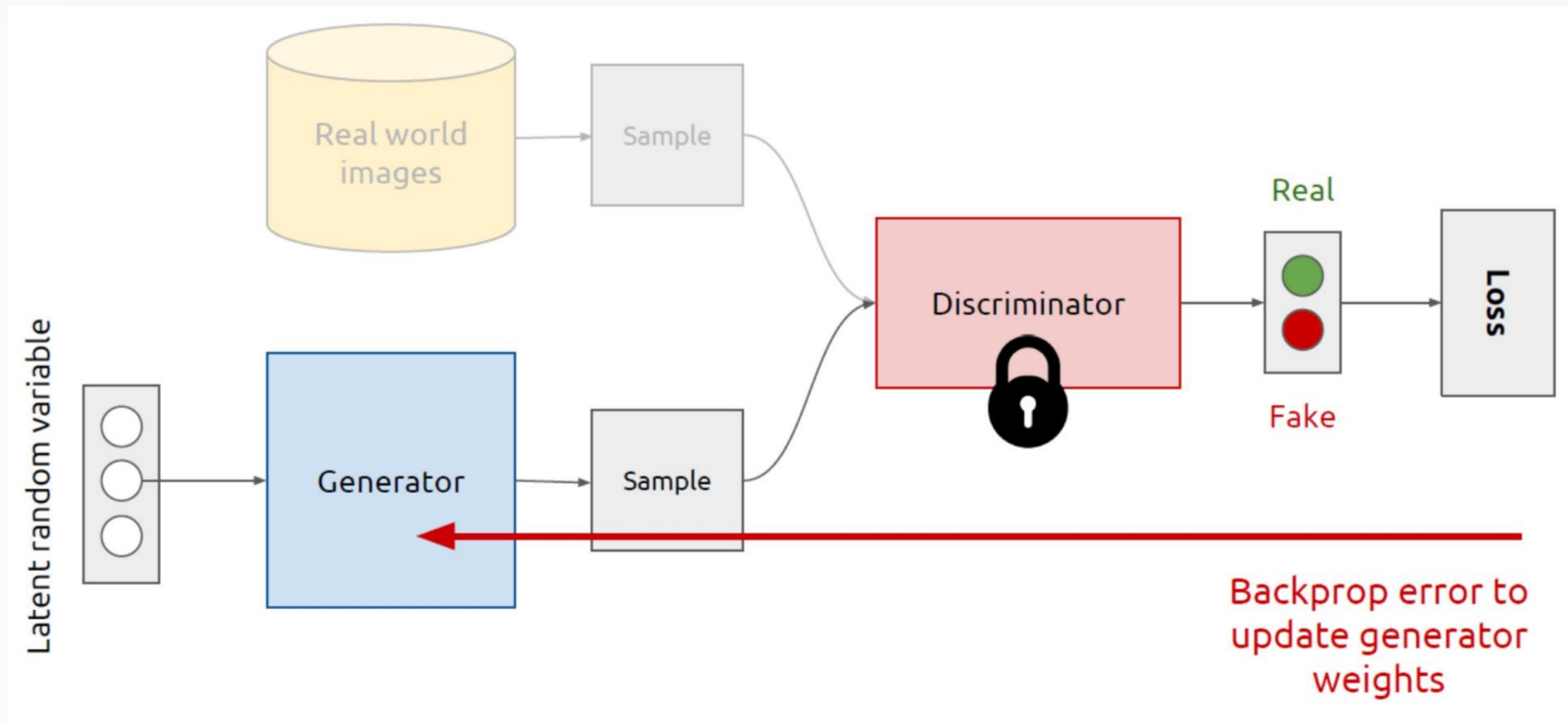
$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$



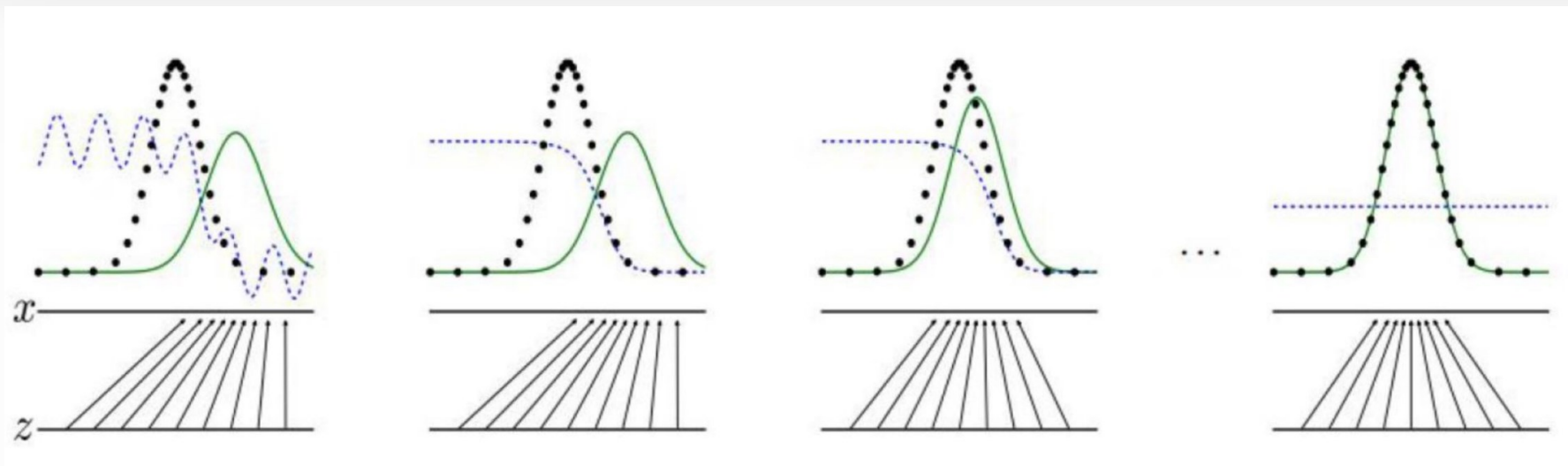
Шаг обучения дискриминатора



Шаг обучения дискриминатора



Изменение обучения генератора



Продолжаем «шаги» подъема до сходимости

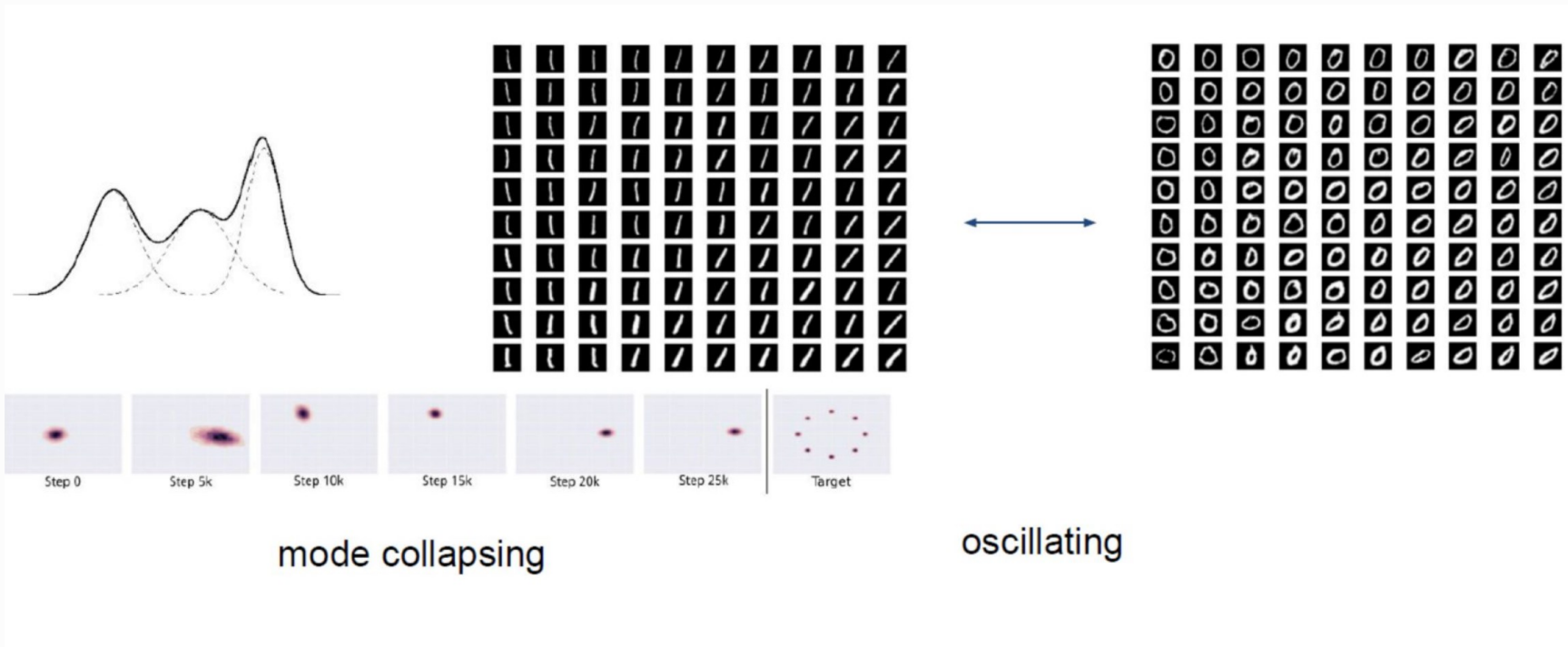
Однако мы можем не достичь сходимости

Недостатки GANs

- Нет гарантии равновесия
 - Схлопывание мод распределения (mode collapse)
 - Осцилляция
 - Нет индикатора, когда останавливаться

Mode collapse и осцилляции

Генератор воспроизводит только некоторые из мод распределения



План лекции

- Задача генерации новых объектов
- PixelCNN и PixelRNN
- Вариационные автокодировщики
- Генеративно-состязательные модели
- Интересные идеи в GAN-ах

Conditional GANs

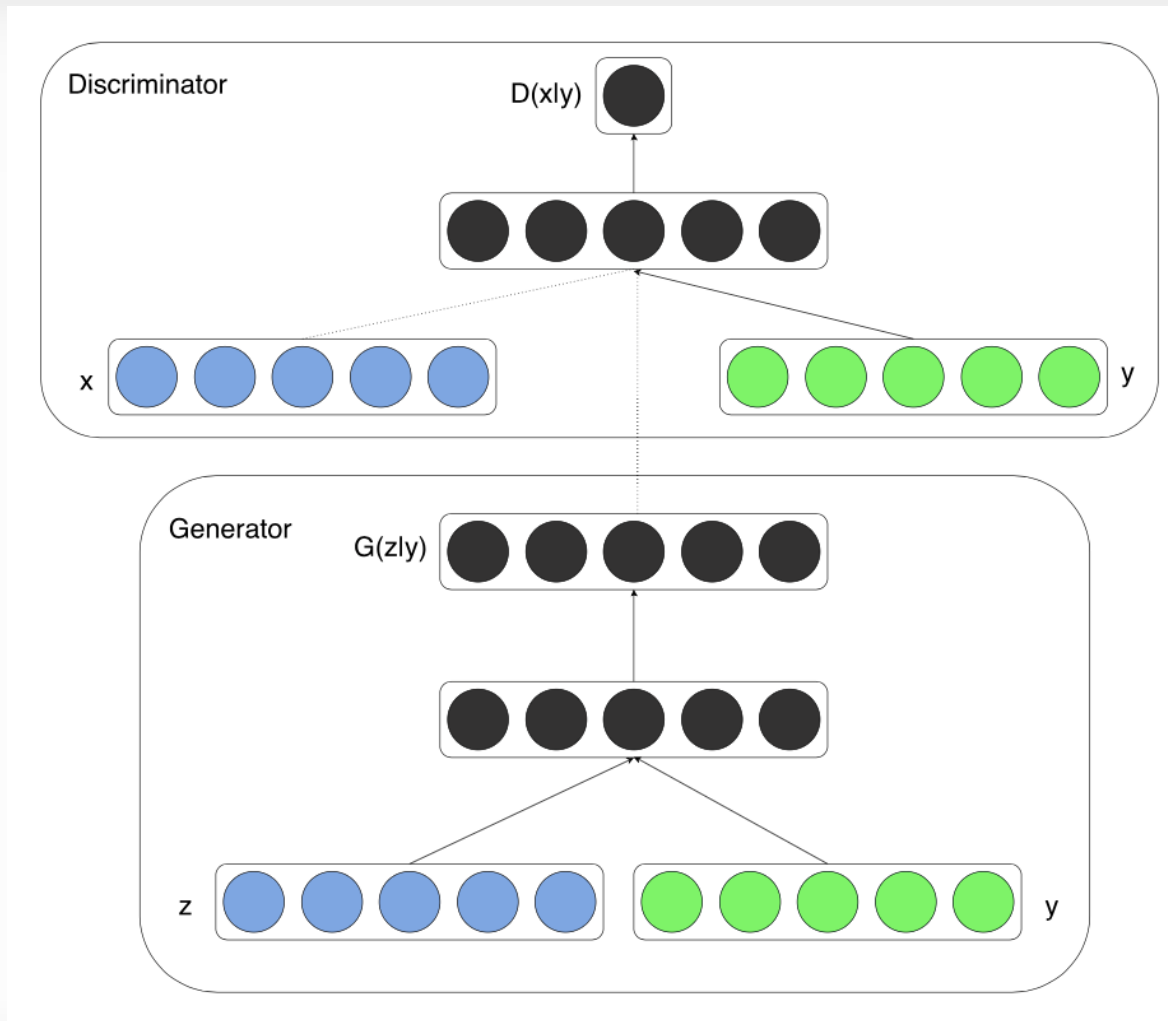
- Идея состоит в том, чтобы добавить несколько меток, чтобы дискриминатор мог работать как классификатор по отношению к некоторым меткам.
- В этом случае у нас разное распределение для каждого класса

Conditional GANs

- Тогда целевая функция выглядит так:

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

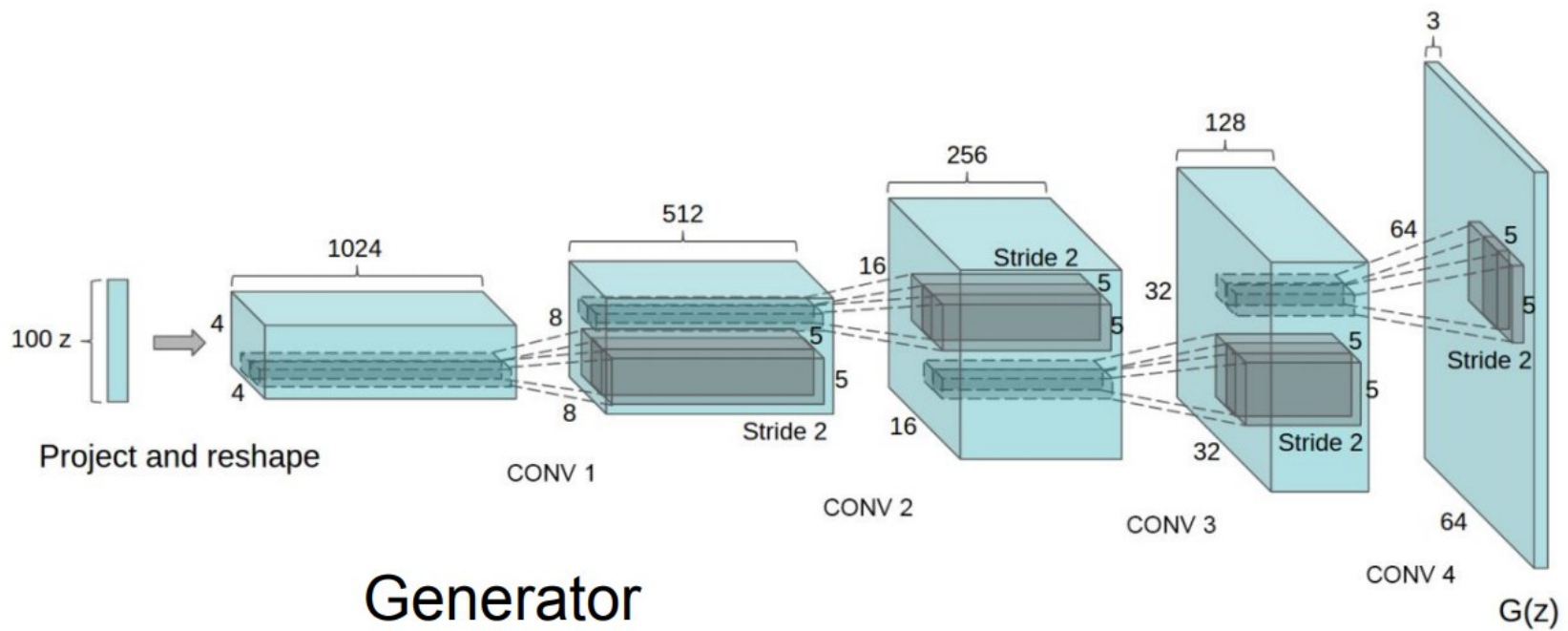
CGANs



DCGAN улучшение

- Нет пулинга (вместо него свертки)
- Батчнормализация в генераторе и дискриминаторе
- Нет полносвязных скрытых слоёв
- ReLU в генераторе на всех слоях и \tanh на выходе
- LeakyReLU в дискриминаторе на всех слоях

DCGAN



Задача оптимального перемещения

- KL не лучший способ измерения расстояний между распределениями
- Расстояние Вассерштайна:

$$W(p_{\text{data}}, p_{\text{gen}}) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_{\text{gen}})} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|]$$

где $\Pi(p_{\text{data}}, p_{\text{gen}})$ это множество совместных распределений над p_{data} и p_{gen}

- Его нельзя посчитать напрямую, но можно найти приближительное решение, которое поможет бороться с mode collapse

State-of-the-art on 12th Dec 2018

