



Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского
Институт информационных технологий, математики и механики

Образовательный курс
«Современные методы и технологии
глубокого обучения в компьютерном зрении»

Генерация синтетических данных с использованием генеративных состязательных сетей

При поддержке компании Intel

Кустикова Валентина

Содержание

- ❑ Цель лекции
- ❑ Понятие генеративной модели
- ❑ Отличие дискриминативных и генеративных моделей
- ❑ Генеративные состязательные сети
 - Общая схема модели
 - Постановка задачи обучения
 - Алгоритм обучения модели
- ❑ Классификация генеративных состязательных сетей
- ❑ Примеры приложений генеративных состязательных сетей
- ❑ Заключение



Цель лекции

- **Цель** – изучить общую схему построения генеративных состязательных сетей и алгоритм их обучения, рассмотреть классификацию генеративных состязательных сетей и примеры их практического использования



ПОНЯТИЕ ГЕНЕРАТИВНОЙ МОДЕЛИ



Генеративная модель

- **Генеративная модель** описывает общие правила генерации набора данных в терминах вероятностной модели. Сэмплирование данных из построенной вероятностной модели позволяет генерировать новые данные



Схема генеративного моделирования (1)

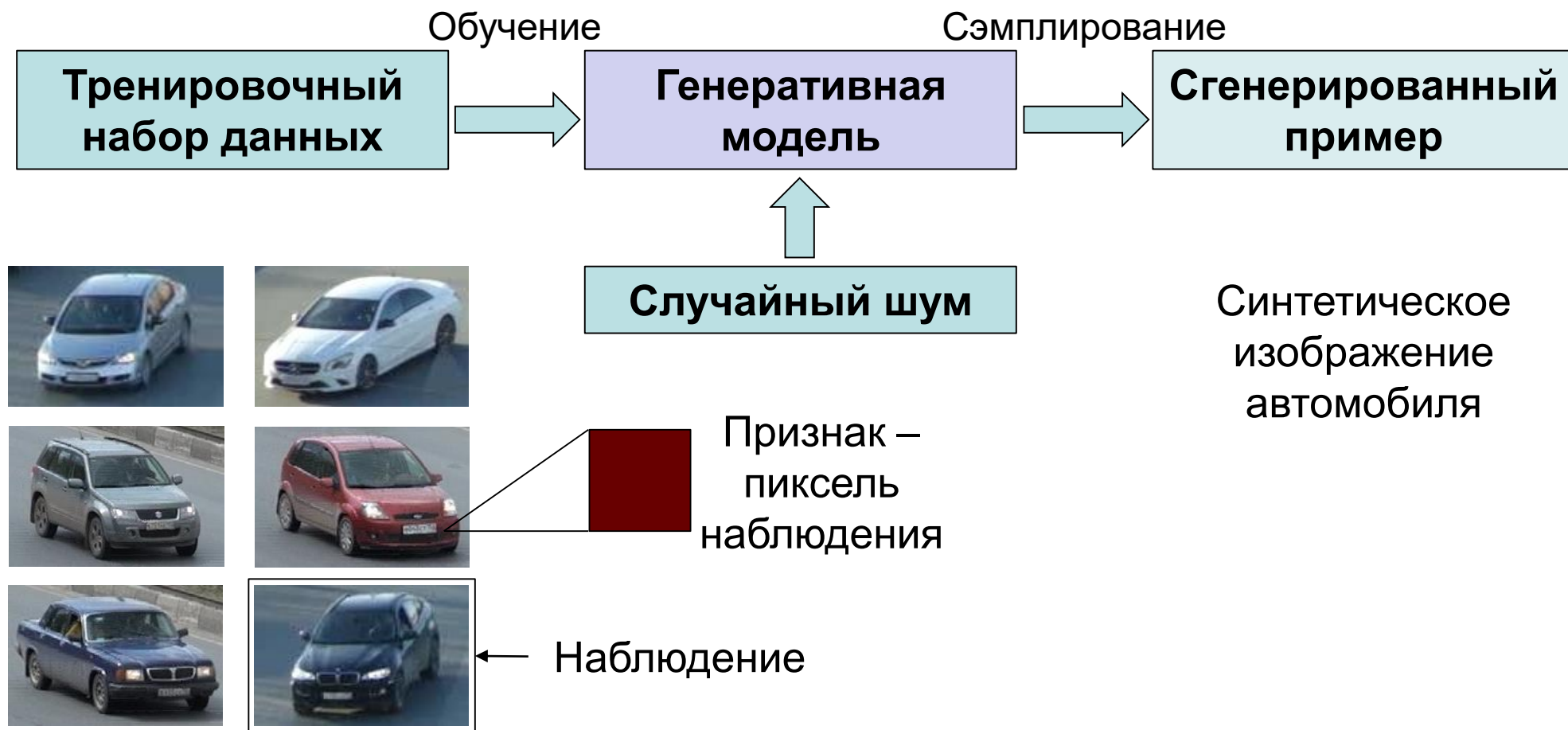


Схема генеративного моделирования (2)

- ❑ Предполагается, что имеется набор данных, содержащий множество примеров сущностей, которые требуется генерировать, – **тренировочный набор данных** (training dataset)
- ❑ **Наблюдение** (observation) – отдельный элемент тренировочного набора данных
- ❑ В задаче генерации изображений каждое наблюдение состоит из множества признаков, как правило, **признак** – значение интенсивности отдельного пикселя
- ❑ Модель обучается генерировать **данные** (изображения) по тем же правилам, что построен тренировочный набор данных
- ❑ В процессе генерации каждому пикселю назначается некоторое значение интенсивности

Схема генеративного моделирования (3)

- ❑ Генеративная модель должна быть вероятностной
- ❑ Если модель в процессе сэмплирования, например, в качестве значения интенсивности пикселя выдает среднее значение по тренировочной выборке, то модель не является генеративной, поскольку обеспечивает всегда одинаковый результат
- ❑ Модель должна включать элемент случайности, отражающий индивидуальные особенности сгенерированного примера
- ❑ Таким образом, **должно существовать вероятностное распределение, которое объясняет, почему некоторые наблюдения (изображения) похожи в тренировочном наборе, а другие – нет**
- ❑ **Цель – построить модель, наилучшим образом имитирующую это распределение**



ОТЛИЧИЕ ДИСКРИМИНАТИВНЫХ И ГЕНЕРАТИВНЫХ МОДЕЛЕЙ



Схема дискриминативного моделирования (1)

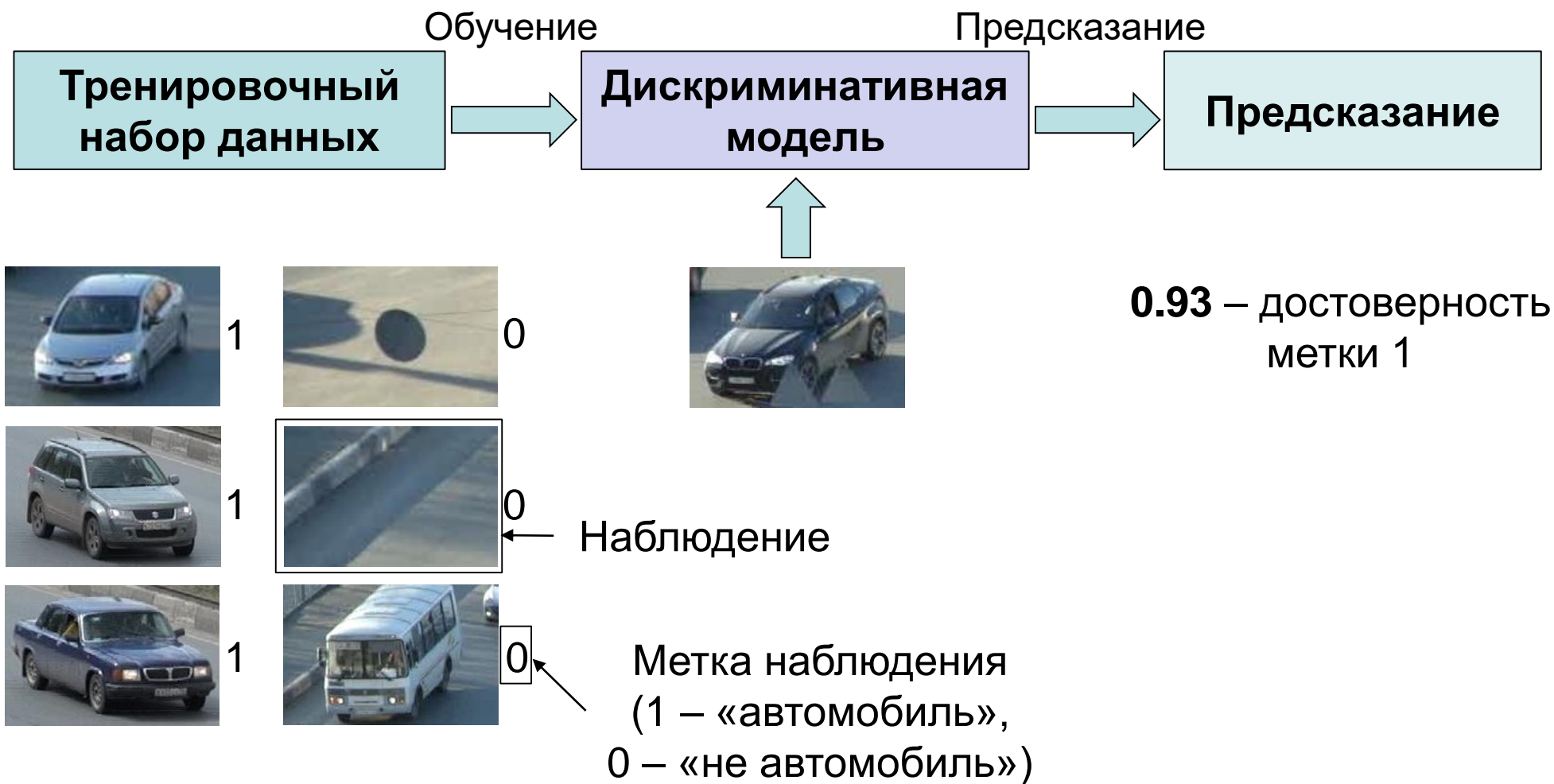


Схема дискриминативного моделирования (2)

- ❑ Предполагается, что имеется **тренировочный набор данных** (training dataset)
- ❑ **Наблюдение** (observation) – отдельный элемент тренировочного набора данных
- ❑ Каждое наблюдение имеет **метку** (label)
- ❑ **Цель – построить функцию на основании тренировочного набора данных, которая наилучшим образом отображает наблюдения на множество меток**



Отличие дискриминативного и генеративного моделирования (1)

□ *Разные цели моделирования*

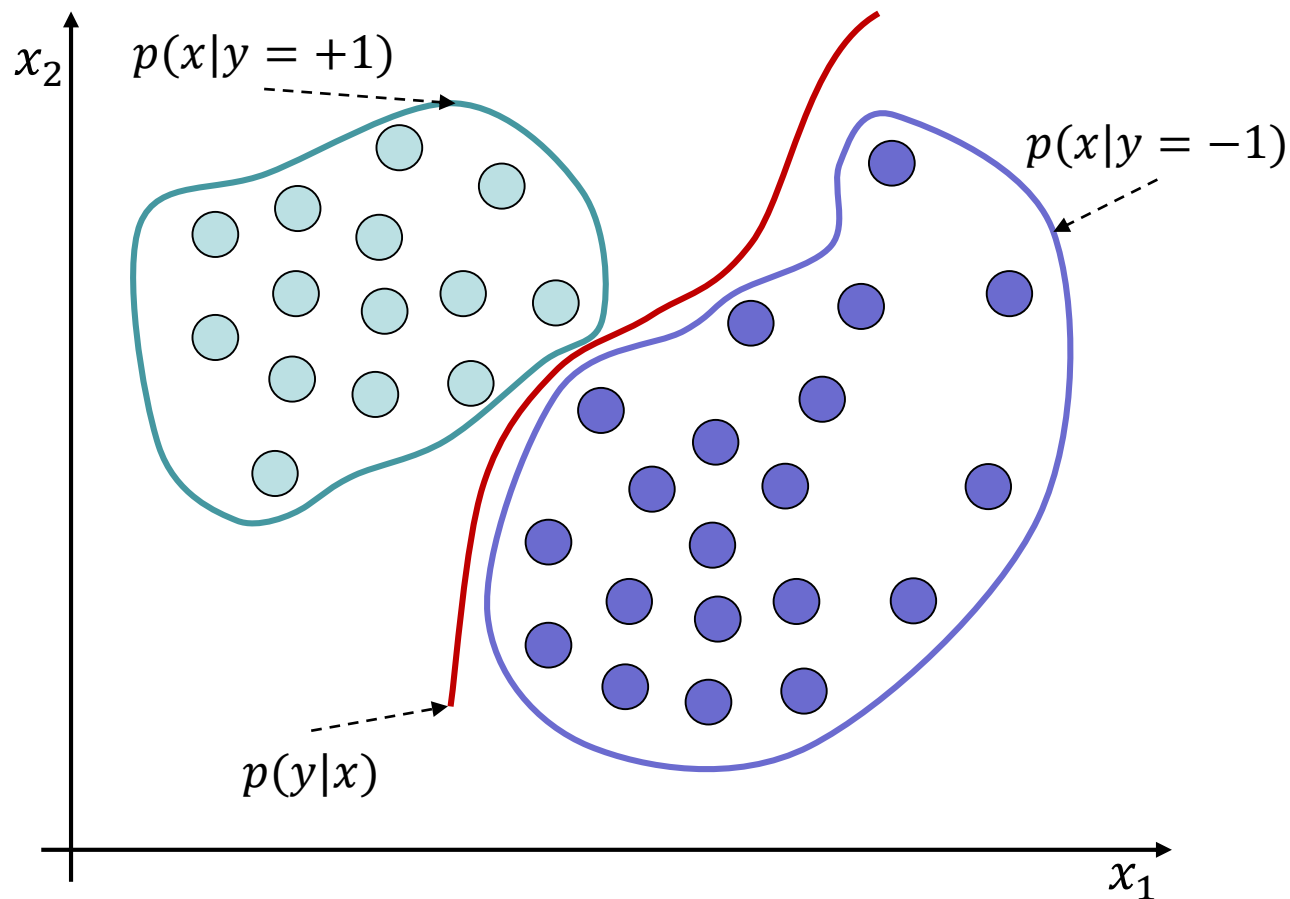
- **Генеративные модели** оценивают $p(x, y)$ – совместное распределение вероятностей (оценивают границы классов)
 - Если набор данных не размечен (решается задача обучения без учителя), то модели оценивают $p(x)$ – вероятность увидеть наблюдение x
 - Если тренировочный набор размечен, то генеративная модель может оценивать условную вероятность $p(x|y)$ (вычисляется по правилу Байеса) – вероятность увидеть наблюдение x при условии, что известна его принадлежность классу y
- **Дискриминативные модели** оценивают $p(y|x)$ – вероятность метки y при условии, что на входе имеется наблюдение x (оценивают границу между классами)

* Ng A.Y., Jordan M.I. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naïve Bayes // Advances in Neural Information Processing Systems. – 2002. – [<http://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf>].



Отличие дискриминативного и генеративного моделирования (2)

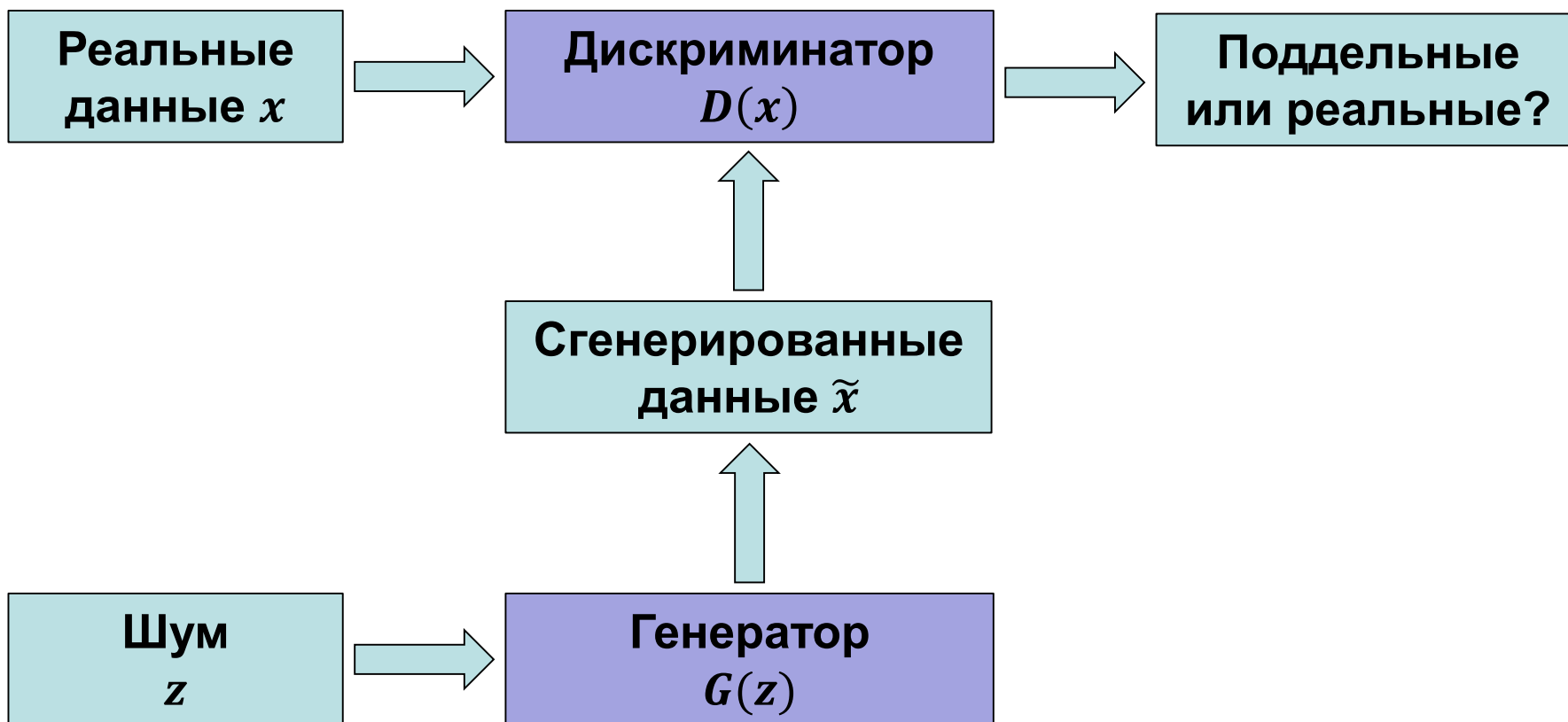
- Графическая интерпретация на примере двумерного наблюдения и двух классов объектов:



ГЕНЕРАТИВНЫЕ СОСТЯЗАТЕЛЬНЫЕ СЕТИ



Общая схема модели (1)



Общая схема модели (2)

- Генеративная состязательная сеть состоит из двух нейронных сетей:
 - **Генератор** – сеть, которая генерирует образцы. Цель генератора – научиться «обманывать» дискриминатор
 - **Дискриминатор** – сеть, которая пытается отличить реальные наблюдения от сгенерированных образцов. Цель дискриминатора – научиться наилучшим образом отличать «обман»
- Пример – генерация фотографий реалистичных лиц:
 - Вход генератора – многомерный случайный шум, выход генератора (вход дискриминатора) – сгенерированное RGB-изображение лица
 - Выход дискриминатора – достоверность того, что RGB-изображение – настоящее лицо (число от 0 до 1)



Постановка задачи обучения (1)

□ Обозначения:

- X – множество наблюдений из распределения p_{data}
- Z – пространство латентных факторов из распределения p_Z (например, множество случайных векторов из равномерного распределения на отрезке $[0; 1]$)

□ **Генератор**

- Отображение $G: Z \rightarrow \mathbb{R}^n$ с параметрами θ
- Цель G – сгенерировать образец, максимально похожий на наблюдения из распределения p_{data}

□ **Дискриминатор**

- Отображение $D: \mathbb{R}^n \rightarrow [0; 1]$ с параметрами γ
- Цель D – выдавать максимальную оценку на наблюдениях из X и минимальную оценку на образцах, сгенерированных G



Постановка задачи обучения (2)

- Обозначения:
 - θ и γ – параметры нейронных сетей, соответствующих генератору и дискриминатору
 - p_{gen} – распределение образцов, порождаемое генератором
- **Задача** – построить распределение p_{gen} , которое наилучшим образом описывает распределение p_{data}



Постановка задачи обучения (3)

- **Функция ошибки:**

$$E_{x \sim p_{data}} [\log D(x)] + E_{\tilde{x} \sim p_{gen}} [\log(1 - D(\tilde{x}))],$$

где $E_{\tilde{x} \sim p_{gen}} [\log(1 - D(\tilde{x}))] = E_{z \sim p_z} [\log(1 - D(G(z)))]$

- **Задача обучения** генеративной состязательной сети сводится к решению задачи оптимизации (минимаксная игра)

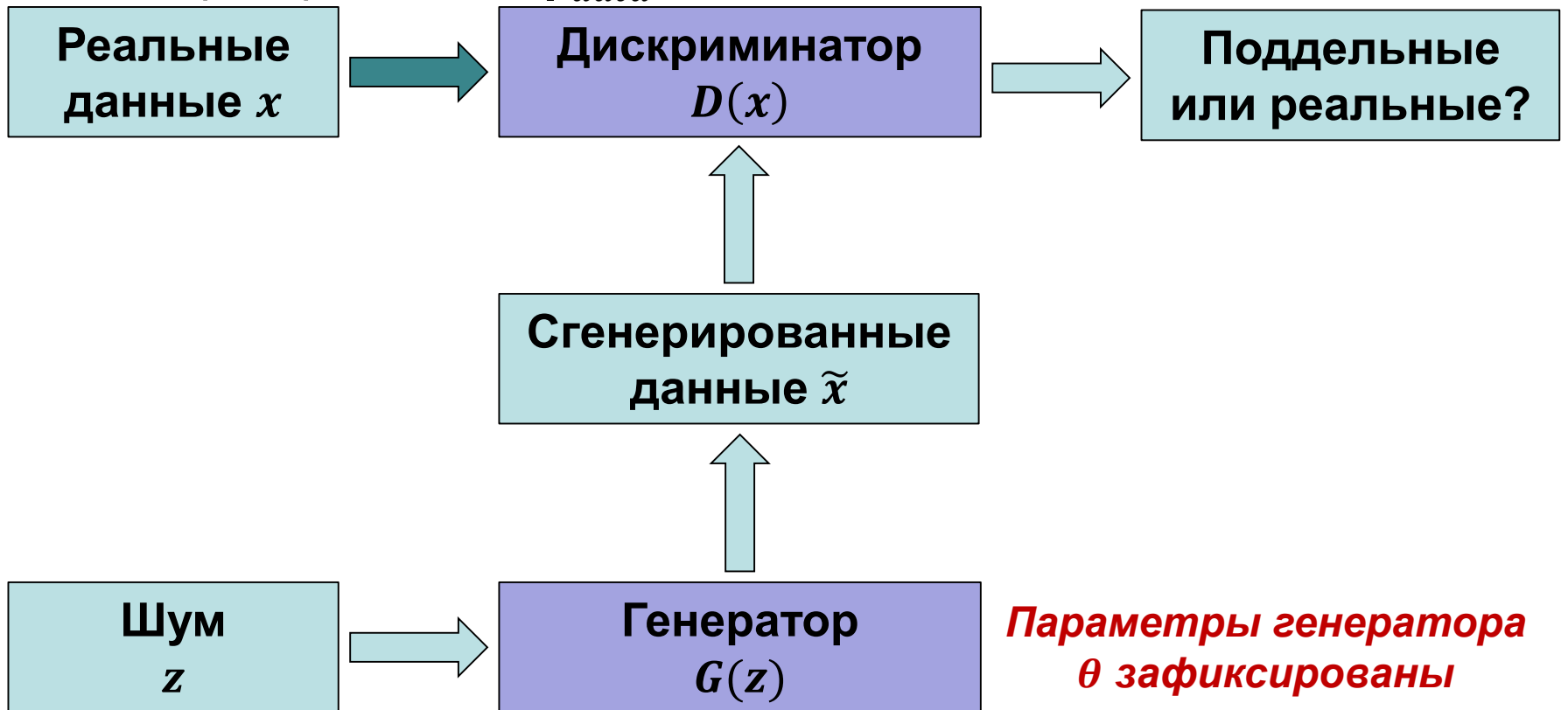
$$\min_G \max_D E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p_z} [\log(1 - D(G(z)))]$$

- **Примечание:** обоснование сходимости распределения p_{gen} к p_{data} описано в исходной статье*, где предложены модели

* Goodfellow I.J., et al. Generative Adversarial Nets. – 2014. – [\[https://arxiv.org/pdf/1406.2661.pdf\]](https://arxiv.org/pdf/1406.2661.pdf).

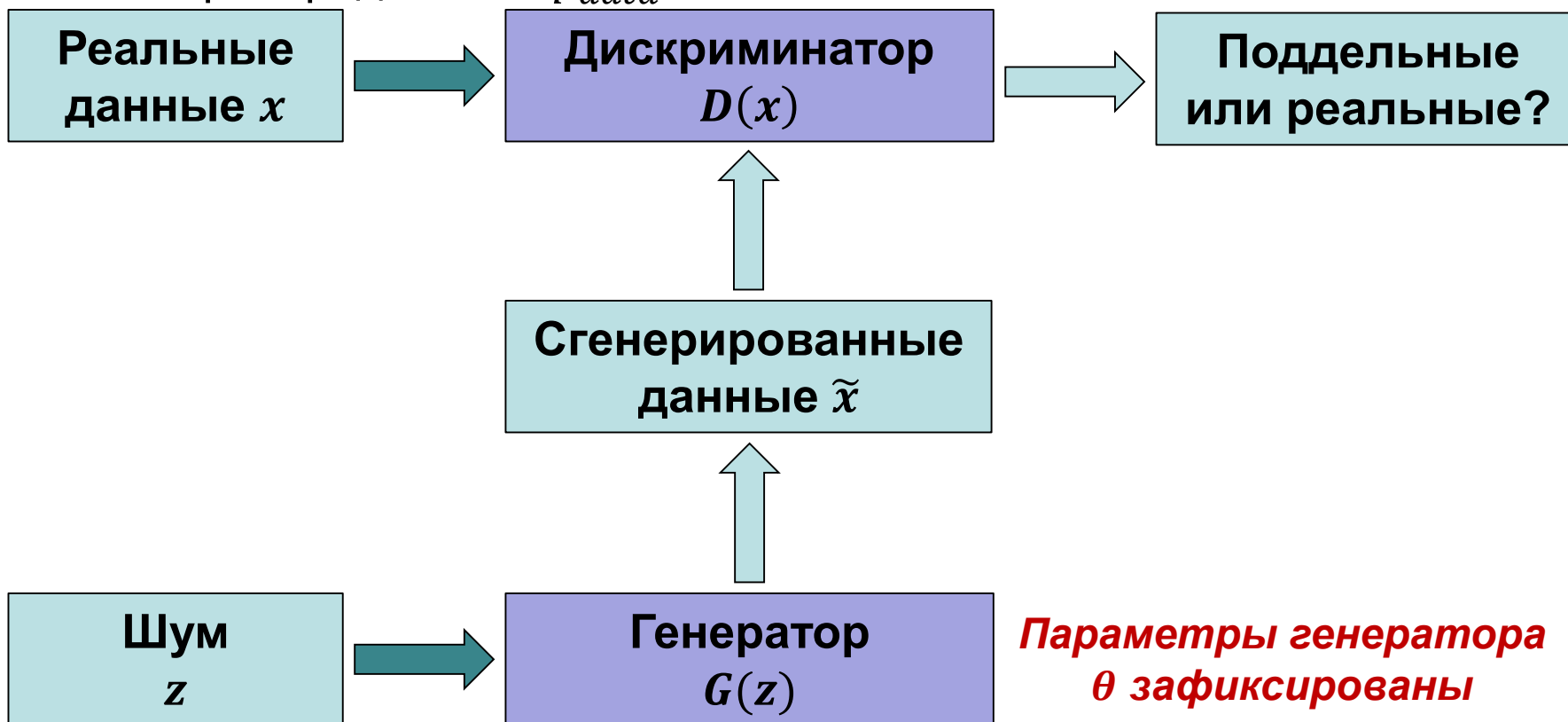
Алгоритм обучения. Этап 1 (1)

1. Сэмплировать $\{x_1, x_2, \dots, x_m\}$
из распределения p_{data}



Алгоритм обучения. Этап 1 (2)

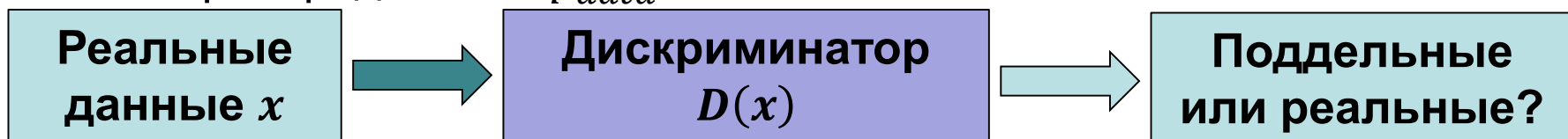
1. Сэмплировать $\{x_1, x_2, \dots, x_m\}$
из распределения p_{data}



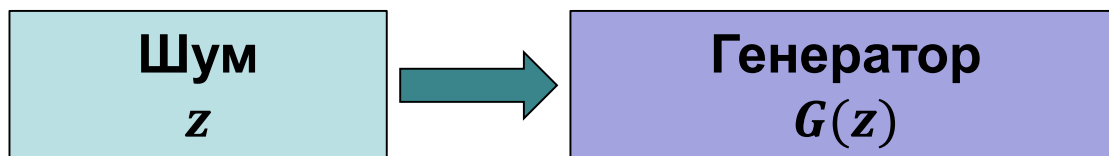
2. Сэмплировать $\{z_1, z_2, \dots, z_m\}$
из распределения p_z

Алгоритм обучения. Этап 1 (3)

1. Сэмплировать $\{x_1, x_2, \dots, x_m\}$
из распределения p_{data}



3. Обновить параметры дискриминатора γ (метод обратного распространения ошибки)

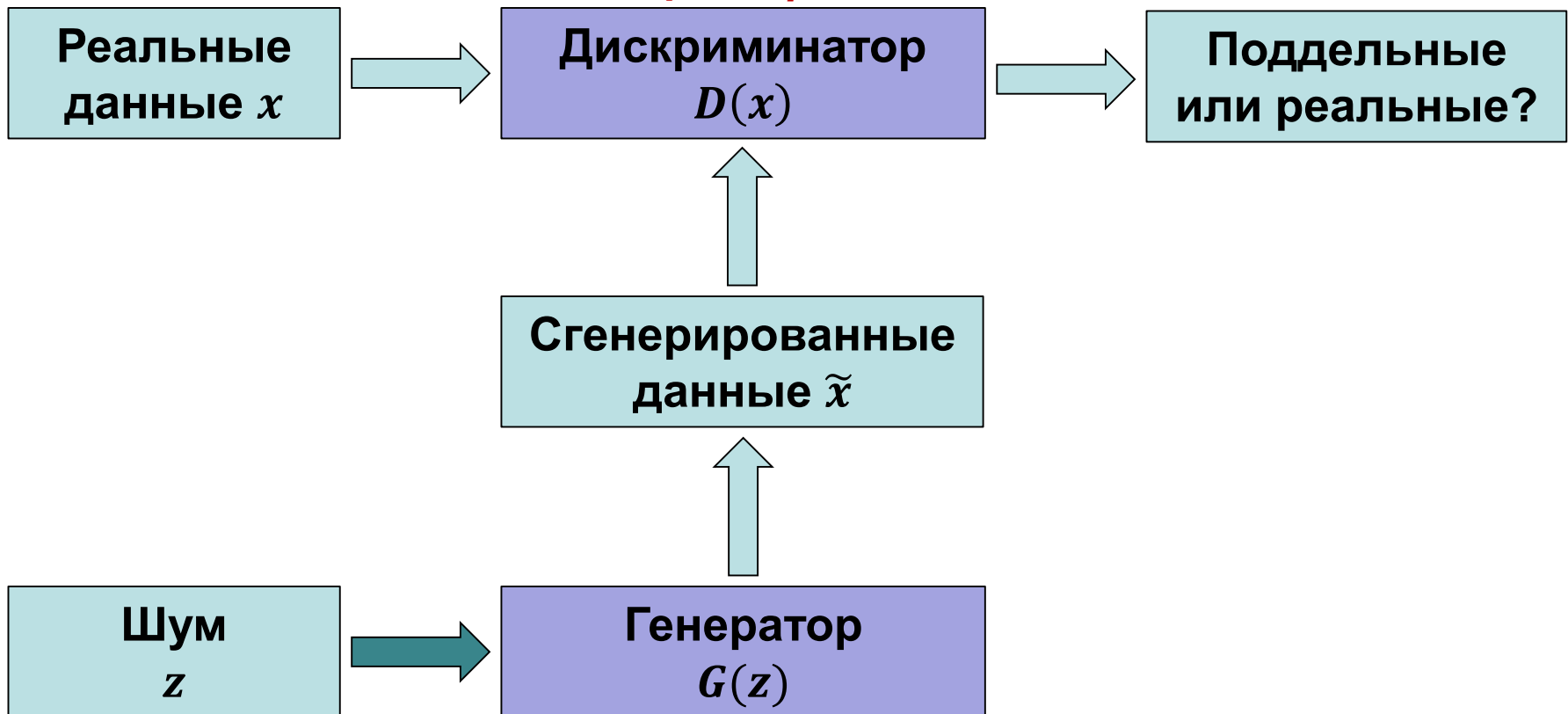


2. Сэмплировать $\{z_1, z_2, \dots, z_m\}$
из распределения p_z

Параметры генератора θ зафиксированы

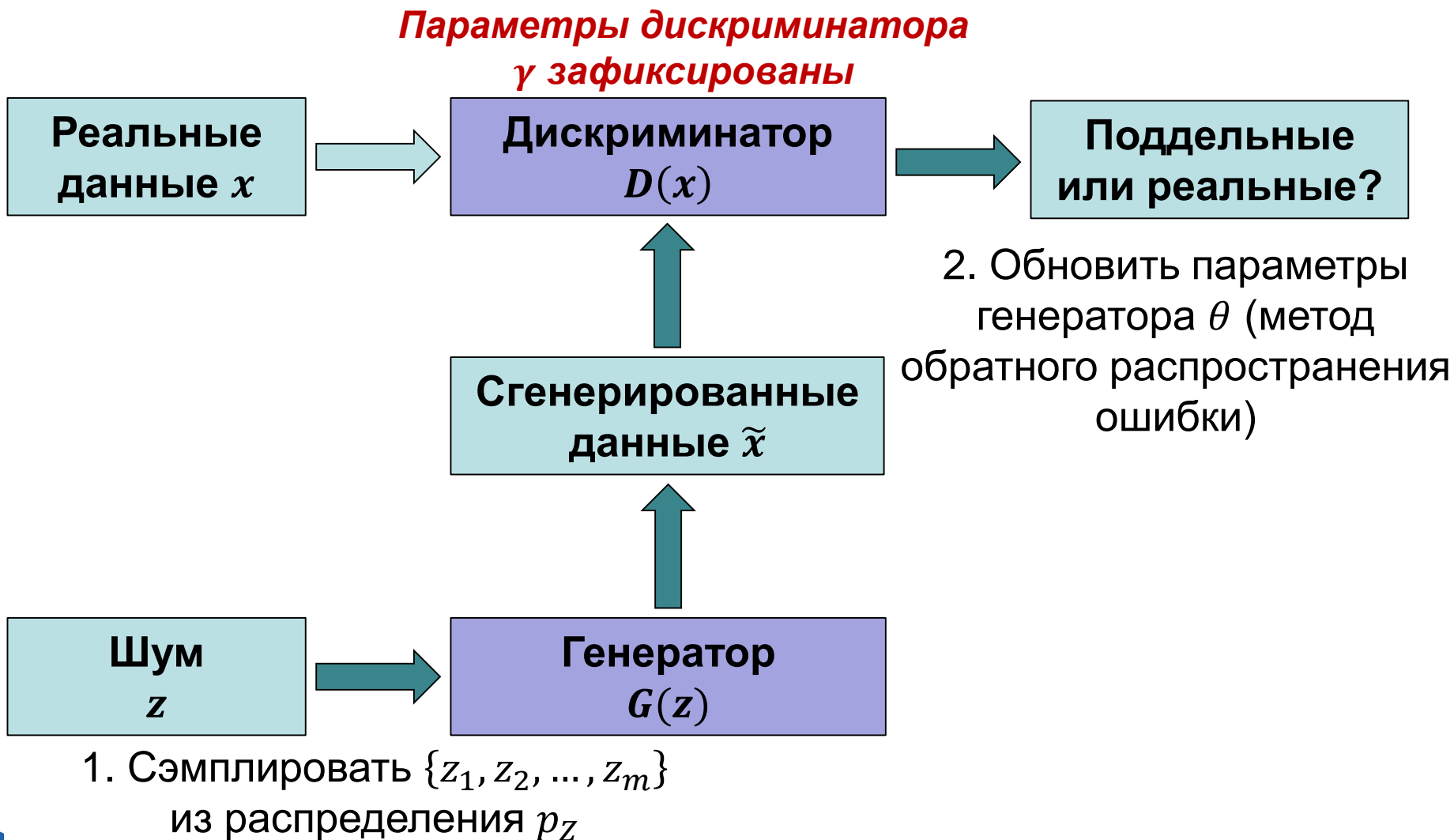
Алгоритм обучения. Этап 2 (1)

*Параметры дискриминатора
у зафиксированы*



1. Сэмплировать $\{z_1, z_2, \dots, z_m\}$
из распределения p_Z

Алгоритм обучения. Этап 2 (2)



Алгоритм обучения. Псевдокод

```
for i = 1..num_iteration do
```

```
  for j = 1..k do
```

```
    1.1. Сэмплировать  $\{x_1, \dots, x_m\}$  из распределения  $p_{data}$ 
```

```
    1.2. Сэмплировать  $\{z_1, \dots, z_m\}$  из распределения  $p_Z$ 
```

```
    1.3. Обновить параметры дискриминатора  $\gamma$ 
```

$$\Delta\gamma \leftarrow \nabla_{\gamma} \frac{1}{m} \sum_{t=1}^m [\log D(x_t)] + [\log(1 - D(G(z_t)))]$$

```
  end for
```

```
    2.1. Сэмплировать  $\{z_1, \dots, z_m\}$  из распределения  $p_Z$ 
```

```
    2.2. Обновить параметры генератора  $\theta$ 
```

$$\Delta\theta \leftarrow \nabla_{\theta} \frac{1}{m} \sum_{t=1}^m [\log(1 - D(G(z_t)))]$$

```
end for
```

Алгоритм обучения. Особенности реализации

- ❑ Количество итераций обновления параметров дискриминатора k и параметров генератора `num_iterations` являются параметрами алгоритма обучения
- ❑ При реализации обратного распространения ошибки в исходном методе обучения используется **стохастический градиентный спуск** (Stochastic Gradient Descend, SGD)



КЛАССИФИКАЦИЯ ГЕНЕРАТИВНЫХ СОСТЯЗАТЕЛЬНЫХ СЕТЕЙ



Классификация генеративных состязательных сетей (1)

- ❑ **Полносвязные генеративные состязательные сети** (Fully Connected GANs)
- ❑ **Условные генеративные состязательные сети** (Conditional GANs, CGAN)
- ❑ **Пирамида Лапласа из состязательных сетей** (Laplacian Pyramid of Adversarial Networks, LAPGAN)
- ❑ **Глубокие сверточные генеративные состязательные сети** (Deep Convolutional GANs, DCGAN)
- ❑ **Генеративные рекуррентные состязательные сети** (Generative Recurrent Adversarial Networks, GRAN)
- ❑ ...

* Alqahtani H., Kavakli-Thorne M., Kumar G. Applications of Generative Adversarial Networks (GANs): An Updated Review // Archives of Computational Methods in Engineering. – 2019.

Классификация генеративных состязательных сетей (2)

- ❑ **Генеративные состязательные сети, максимизирующие информацию** (Information Maximizing GANs, InfoGAN)
- ❑ **Двунаправленные генеративные состязательные сети** (Bidirectional GANs, BiGAN)
- ❑ За пределами лекции останутся следующие виды генеративных состязательных сетей:
 - Генеративные автокодировщики (Adversarial Autoencoders, AAE)
 - Гибриды вариационных автокодировщиков и генеративных состязательных сетей (Variational Autoencoder-GAN, VAE-GAN)
 - Некоторые другие специализированные модели

* Alqahtani H., Kavakli-Thorne M., Kumar G. Applications of Generative Adversarial Networks (GANs): An Updated Review // Archives of Computational Methods in Engineering. – 2019.

Fully Connected GANs

- ❑ **Полносвязные генеративные состязательные сети** (Fully Connected GANs) – модели, в которых генератор и дискриминатор имеют вид полносвязных глубоких нейронных сетей
- ❑ Впервые полносвязные генеративные состязательные сети использованы для генерации реалистичных изображений, аналогичных содержащимся в следующих наборах данных*:
 - MNIST [<http://yann.lecun.com/exdb/mnist>]
 - CIFAR-10 [<https://www.cs.toronto.edu/~kriz/cifar.html>]
 - Toronto Face Dataset (TFD)**

* Goodfellow I.J., et al. Generative Adversarial Nets // Advances in neural information processing systems. – 2014. – P. 2672-2680. – [<https://arxiv.org/pdf/1406.2661.pdf>].

** Susskind J., Anderson A., Hinton G. E. The Toronto face dataset. Technical Report UTML TR 2010-001. – 2010.



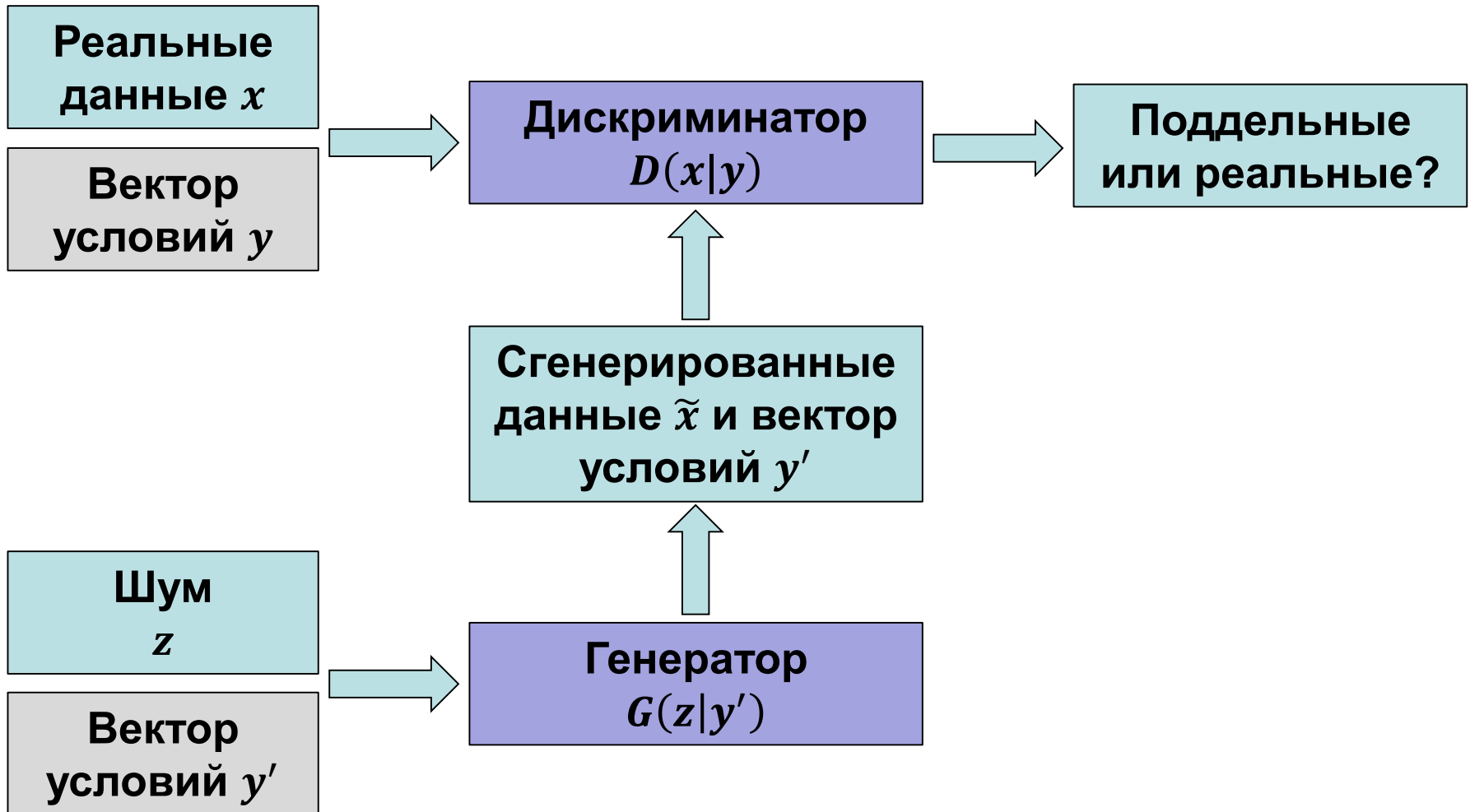
Conditional GANs (1)

- ❑ **Условные генеративные состязательные сети** (Conditional GANs, CGAN) – модели, позволяющие генерировать синтетические изображения, которые удовлетворяют некоторым условиям или обладают некоторыми свойствами (специфическими характеристиками)
- ❑ Генератор и дискриминатор получают дополнительную входную информацию
- ❑ В простейшем случае в качестве дополнительной информации используется класс изображения (вектор меток в представлении one-hot) или интересующие свойства

* Mirza M., Osindero S. Conditional generative adversarial nets. – 2014. – [<https://arxiv.org/pdf/1411.1784.pdf>].



Conditional GANs (2)



Conditional GANs (3)

□ *Генератор*

- Вход генератора перестает быть абсолютно случайным в результате добавления вектора условий в качестве входной информации
- Добавление вектора условий помогает генератору понять, каким образом лучше сгенерировать данные

□ *Дискриминатор*

- Дискриминатор принимает решение о реальности/поддельности данных, учитывая дополнительную информацию

□ *Общая схема алгоритма обучения аналогична ранее рассмотренной, разница состоит в функции ошибки, которая зависит от дополнительных векторов условий*

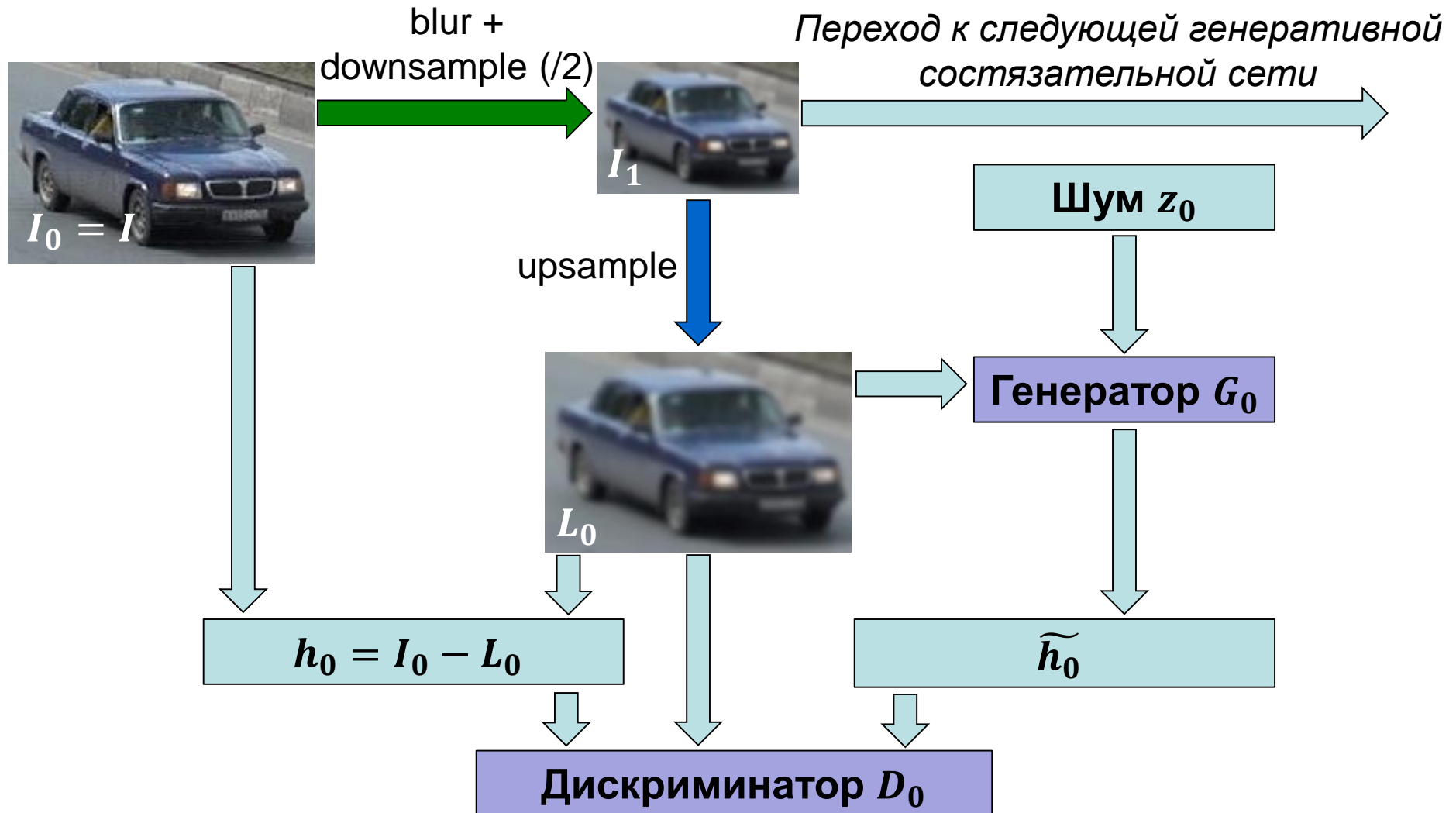


Laplacian Pyramid of Adversarial Networks (1)

- ❑ **Пирамида Лапласа из состязательных сетей** (Laplacian Pyramid of Adversarial Networks, LAPGAN) – модель, основанная на каскаде сверточных сетей, которые образуют пирамиду Лапласа для изображений разного масштаба
- ❑ Модель позволяет генерировать естественные изображения высокого разрешения за счет захвата особенностей на разных масштабах
- ❑ Пирамида Лапласа строится на базе пирамиды Гаусса с помощью операций повышающей (upsampling) и понижающей дискретизации (downsampling)
- ❑ Элемент пирамиды Лапласа задается разницей между соседними уровнями пирамиды Гаусса

* Denton E.L., et al. Deep generative image models using a Laplacian pyramid of adversarial networks // Advances in neural information processing systems. – 2015. – P. 1486-1494. – [<https://arxiv.org/pdf/1506.05751.pdf>].

Laplacian Pyramid of Adversarial Networks (2)

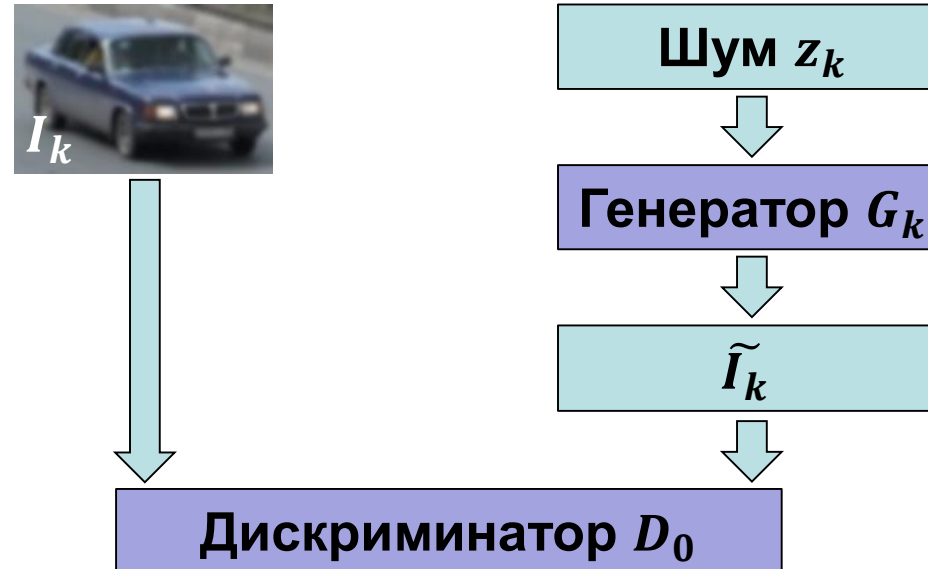


Laplacian Pyramid of Adversarial Networks (3)

- ❑ Первая состязательная сеть в каскаде в процессе обучения принимает на вход исходное изображение
- ❑ Изображение размывается и уменьшается вдвое средствами операции понижающей дискретизации
- ❑ Разрешение полученного изображения увеличивается до разрешения исходного изображения с использованием операции повышающей дискретизации
- ❑ **Генератор** принимает на вход увеличенное изображение и шум и пытается предсказать отличие увеличенного изображения относительно исходного
- ❑ **Дискриминатор** принимает решение, является ли полученное отличие реальным
- ❑ **Обучение модели – обучение условной генеративной состязательной сети**

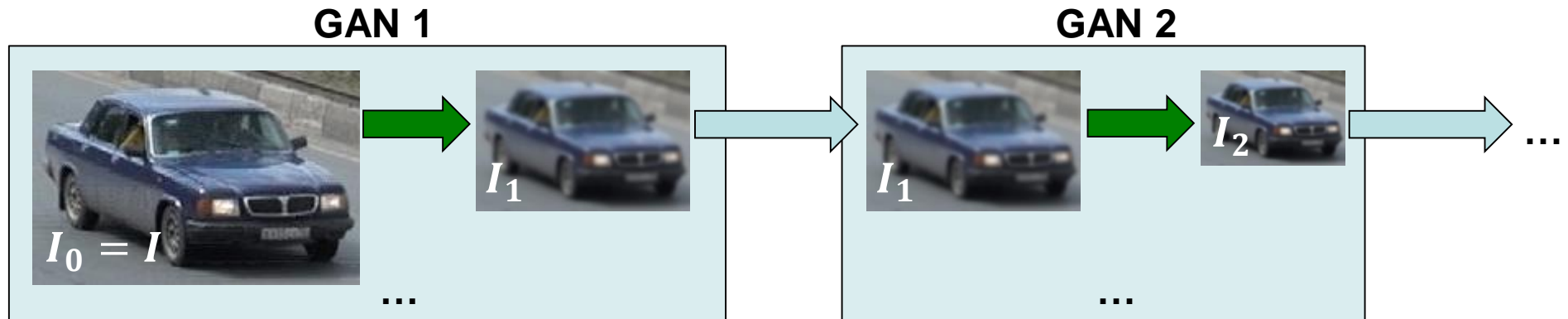
Laplacian Pyramid of Adversarial Networks (4)

- Схема построения каждой последующей генеративной состязательной сети в каскаде аналогична представленной, за исключением последней сети
- **Последняя сеть обеспечивает восстановление изображения на основании шума**, т.е. она не является условной генеративной состязательной сетью



Laplacian Pyramid of Adversarial Networks (5)

- При обучении очередной модели на вход подается изображение, полученное в результате размытия и уменьшения разрешения для предшествующей сети



- Обучение отдельных генеративных состязательных сетей может осуществляться независимо
- Использование обученного каскада сводится к обратному проходу по построенной последовательности моделей, на входе последней модели – изображение низкого разрешения

Deep Convolutional GANs (1)

- **Глубокие сверточные генеративные состязательные сети** (Deep Convolutional GANs, DCGAN) – модели, в которых генератор и дискриминатор представляют собой глубокие сверточные сети, обладающие следующими ограничениями:
 - Отсутствуют полносвязные слои
 - Дискриминатор вместо слоев пространственного объединения (pooling) содержит разреженные свертки (strided convolutions), а генератор – разреженные свертки с дробным шагом (fractional-strided convolutions)
 - В генераторе и дискриминаторе используется нормализация по пачке (batch normalization)
 - ...

* Radford A., Metz L., Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. – 2015. – [<https://arxiv.org/pdf/1511.06434.pdf>].



Deep Convolutional GANs (2)

- **Глубокие сверточные генеративные состязательные сети** (Deep Convolutional GANs, DCGAN) – модели, в которых генератор и дискриминатор представляют собой глубокие сверточные сети, обладающие следующими ограничениями:
 - На всех слоях генератора, кроме последнего, используется функция активации «положительная срезка» (ReLU). В исходной работе* используется гиперболический тангенс (Tanh)
 - На всех слоях дискриминатора используется «слабая положительная срезка» (Leaky ReLU)

* Radford A., Metz L., Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. – 2015. – [<https://arxiv.org/pdf/1511.06434.pdf>].



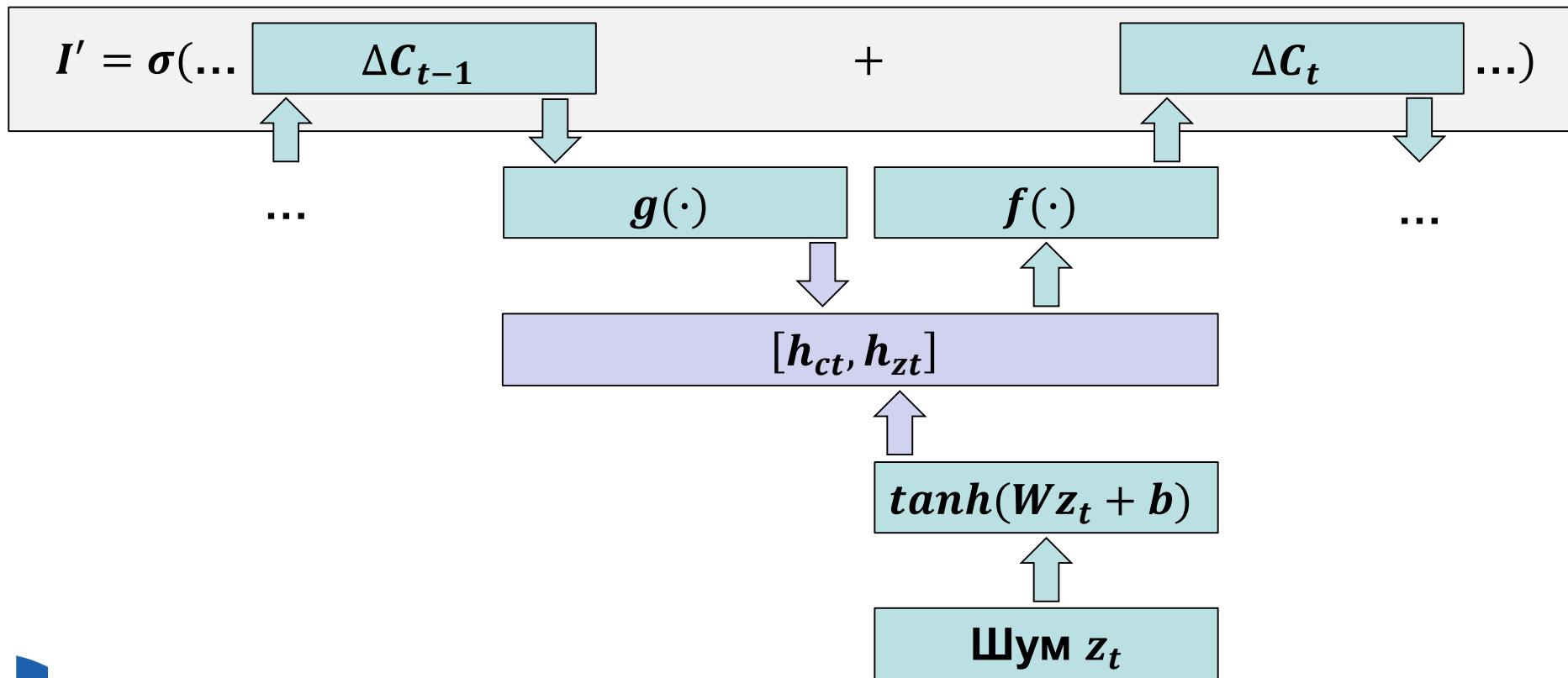
Generative Recurrent Adversarial Networks (1)

- ❑ **Генеративные рекуррентные состязательные сети** (Generative Recurrent Adversarial Networks, GRAN) – модели, в которых генератор содержит рекуррентную связь
- ❑ Генератор получает на вход последовательность примеров «шума» и создает последовательность изображений
- ❑ Очередное изображение построенной последовательности накапливает обновления, что приводит к получению финального образца
- ❑ Дискриминатор определяет, является ли полученное суммарное изображение реальным
- ❑ Далее приведена схема построения генератора

* Im D.J., Kim C.D., Jiang H., Memisevic R. Generating images with recurrent adversarial networks. – 2016. – [<https://arxiv.org/pdf/1602.05110.pdf>].

Generative Recurrent Adversarial Networks (2)

- $\{z_t, t = \overline{1, T}\}, z_t \sim p(Z)$ – последовательность примеров шума
- $\Delta C_1, \Delta C_2, \dots, \Delta C_T$ – сгенерированная последовательность изображений



Generative Recurrent Adversarial Networks (3)

- ❑ $g(\cdot)$ – сеть, содержащая последовательность сверточных слоев и полносвязный слой, работает как кодировщик
- ❑ $f(\cdot)$ – обратная копия сети $g(\cdot)$ (полносвязный слой и последовательность разверточных слоев), работает как декодировщик
- ❑ h_{ct} – закодированное представление изображения, сгенерированного на шаге $t - 1$
- ❑ h_{zt} – гипотеза о необходимых обновлениях
- ❑ $[h_{ct}, h_{zt}]$ – конкатенация
- ❑ Выход генератора представляет собой сумму по всем сгенерированным изображениям
- ❑ Обучение реализуется посредством **метода обратного распространения ошибки с развертыванием во времени** (backpropagation through time)

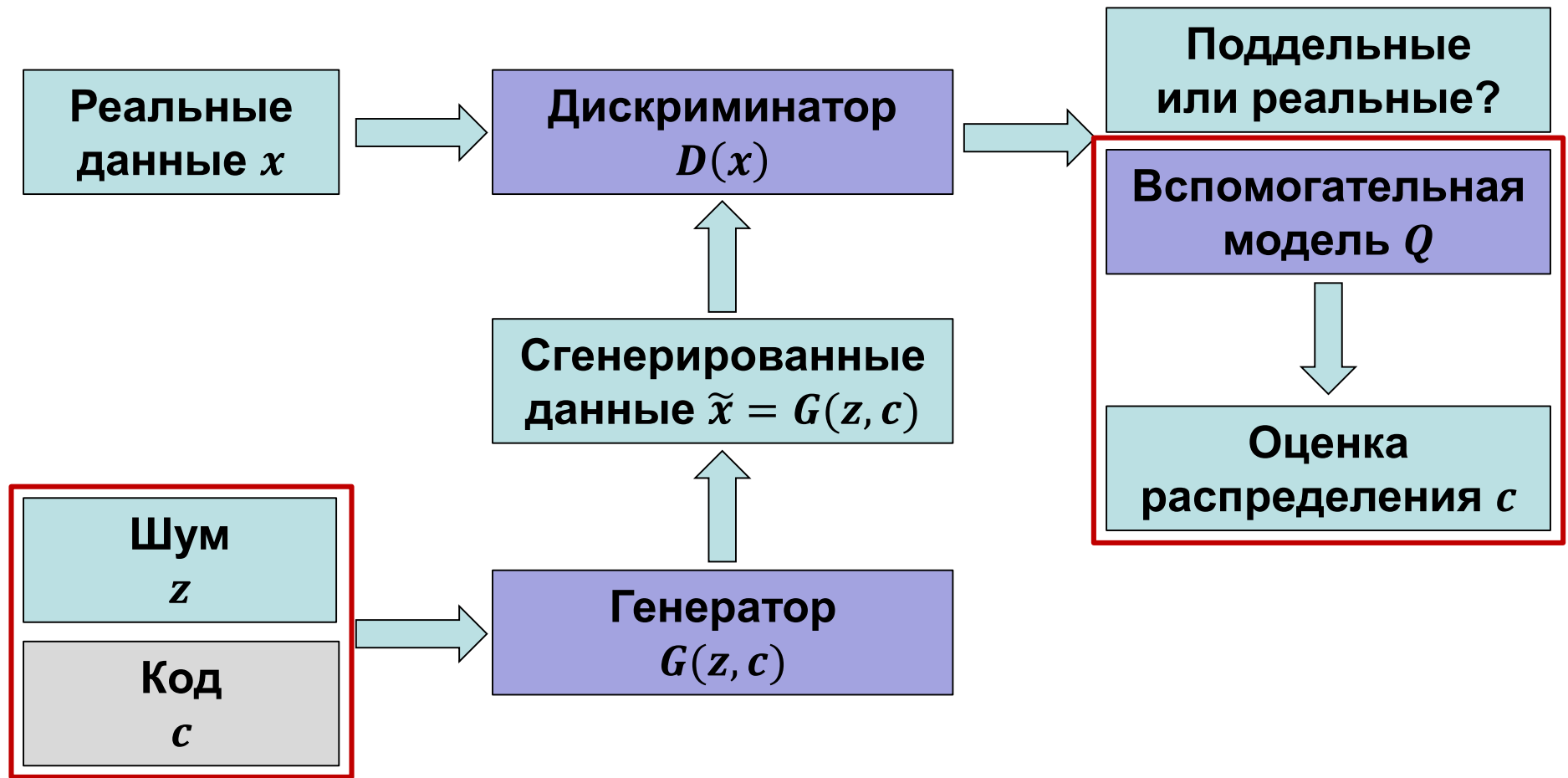


Information Maximizing GANs (1)

- ❑ **Генеративные состязательные сети, максимизирующие информацию** (Information Maximizing GANs, InfoGAN) – теоретико-информационное расширение генеративных моделей, которые способны строить распутанные признаковые представления без учителя
- ❑ Распутанное признаковое представление – набор признаков, которые явно представляют характерные особенности экземпляра данных и могут быть полезны для широко круга задач
- ❑ Цель моделей – построить значимые представления посредством максимизации взаимной информации между небольшим подмножеством переменных шума и наблюдений

* Chen X., et al. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets // Advances in Neural Information Processing Systems. – 2016. – P. 2172-2180. – [<https://arxiv.org/pdf/1606.03657.pdf>].

Information Maximizing GANs (2)



Information Maximizing GANs (3)

- Вектор шума раскладывается на две части:
 - z – источник шума
 - c – скрытый (latent) код, нацеленный на характерные семантические особенности распределения данных
- $G(z, c)$ – сгенерированные синтетические данные
- Между скрытым кодом и распределением генератора должна быть высокая взаимная информация $I(c; G(z, c))$
- Взаимная информация выражается как разница двух энтропий:
$$I(c; G(z, c)) = H(c) - H(c|G(z, c)) = H(G(z, c)) - H(G(z, c)|c)$$
- Функция ошибки содержит дополнительное слагаемое регуляризации:

$$E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p_z, c \sim p_c} \left[\log \left(1 - D(G(z, c)) \right) \right] - \lambda I(c; G(z, c))$$



Information Maximizing GANs (4)

- На практике вычисление взаимной информации $I(c; G(z, c))$ требует знания $P(c|\tilde{x})$
- Известно, что можно построить вспомогательное распределение $Q(c|\tilde{x})$, которое аппроксимирует $P(c|\tilde{x})$
- $Q(c|\tilde{x})$ моделируется параметризованной нейронной сетью
- Исходя из $Q(c|\tilde{x})$, можно получить нижнюю оценку для значения взаимной информации

$$\begin{aligned} I(c; G(z, c)) &= H(c) - H(c|G(z, c)) \\ &= E_{\tilde{x} \sim G(z, c)} \left[E_{c' \sim P(c|\tilde{x})} [\log P(c'|\tilde{x})] \right] + H(c) \\ &= E_{\tilde{x} \sim G(z, c)} \left[\underbrace{D_{KL}(P(\cdot|\tilde{x})||Q(\cdot|\tilde{x}))}_{\geq 0} + E_{c' \sim P(c|\tilde{x})} [\log Q(c'|\tilde{x})] \right] + H(c) \\ &\geq E_{\tilde{x} \sim G(z, c)} \left[E_{c' \sim P(c|\tilde{x})} [\log Q(c'|\tilde{x})] \right] + \underbrace{H(c)}_{const} \end{aligned}$$



Information Maximizing GANs (5)

- В исходной статье доказано следующее равенство:

$$E_{\tilde{x} \sim G(z,c)} \left[E_{c' \sim P(c|\tilde{x})} [\log Q(c'|\tilde{x})] \right] = E_{c \sim p_C, \tilde{x} \sim G(z,c)} [\log Q(c|\tilde{x})]$$

- Таким образом, вычисление слагаемого регуляризации сводится к следующей последовательности действий:
 - Сэмплирование скрытого кода c из распределения p_C
 - Сэмплирование шума z из распределения p_Z
 - Генерация примера данных $\tilde{x} = G(z, c)$
 - Вычисление вероятности $Q(c|\tilde{x} = G(z, c))$
- Финальная функция ошибки:

$$E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p_Z, c \sim p_C} \left[\log \left(1 - D(G(z, c)) \right) \right] - \lambda \left(E_{c \sim p_C, \tilde{x} \sim G(z,c)} [\log Q(c|\tilde{x})] + H(c) \right)$$

Information Maximizing GANs (6)

- В большинстве экспериментов нейронные сети D и Q имеют общие сверточные слои, и имеется один полносвязный слой для вывода параметров распределения $Q(c|x)$
- Скрытые коды могут быть как категориальными, так и непрерывными
 - Типичный категориальный код – класс, которому принадлежит генерируемый образец данных (например, цифра от 0 до 9 в задаче генерации рукописных цифр)
 - Типичный непрерывный код – значение параметра распределения, которому соответствует какой-либо признак генерируемых образцов данных (например, угол наклона генерируемой рукописной цифры)

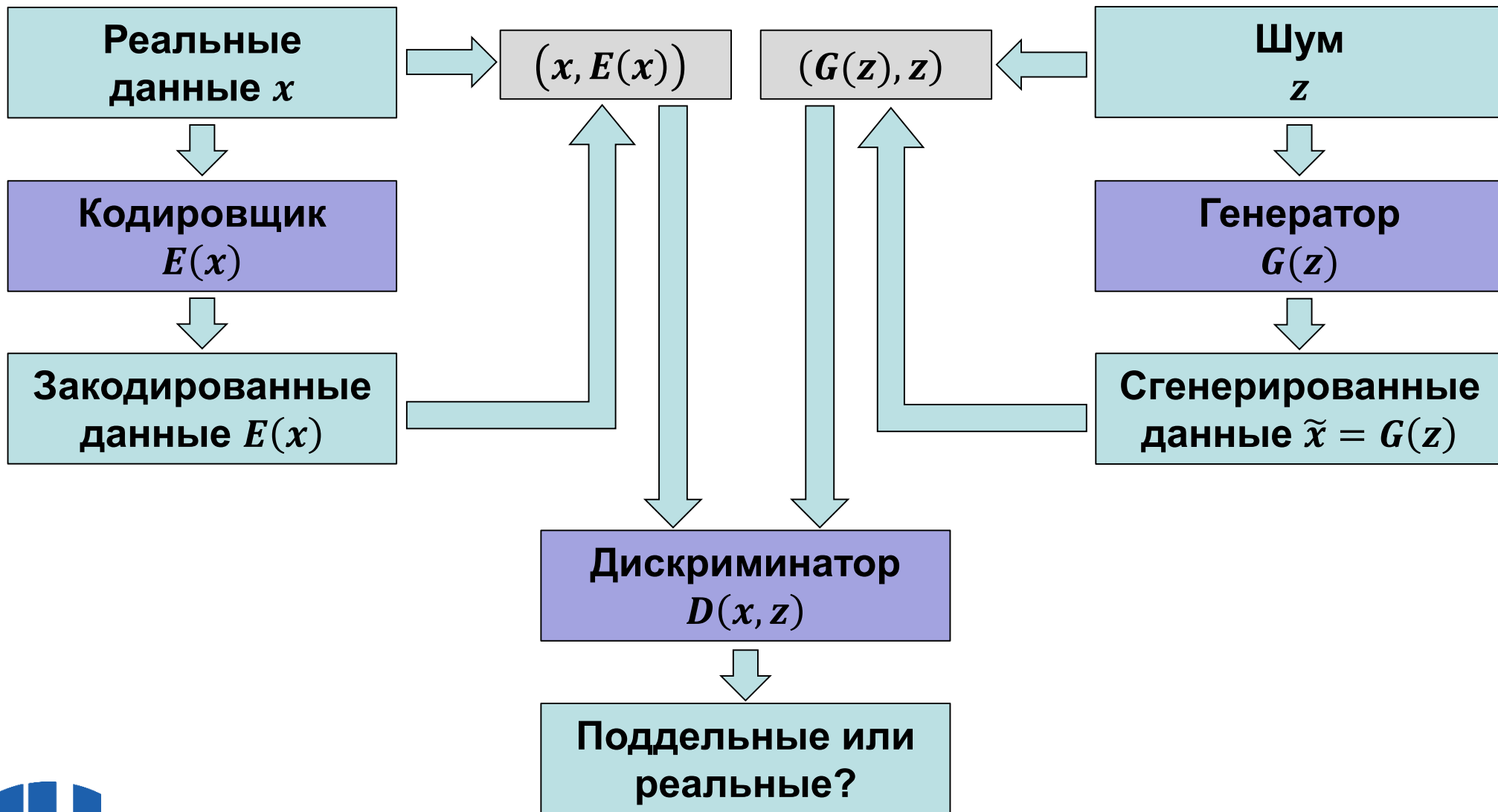


Bidirectional GANs (1)

- ❑ **Двунаправленные генеративные состязательные сети** (Bidirectional GANs, BiGAN) – модели, в которых дискриминатор для принятия решения о том, являются ли данные реальными, использует помимо самих данных их представление в латентном пространстве
- ❑ Входные данные кодируются в представление в латентном пространстве специально выделенным блоком – **кодировщиком**
- ❑ Дискриминатор получает на вход пару (пример данных, представление в латентном пространстве) в случае реальных данных и (сгенерированный пример, шум) при обработке выхода генератора

* Donahue J., Krahenbuhl P., Darrell T. Adversarial feature learning. – 2017. – [<https://arxiv.org/pdf/1605.09782.pdf>].

Bidirectional GANs (2)



Bidirectional GANs (3)

□ **Функция ошибки:**

$$\begin{aligned} & E_{x \sim p_{data}} [\log D(x, E(x))] + E_{z \sim p_Z} [\log(1 - D(G(z), z))] \\ &= E_{x \sim p_{data}} \left[E_{\tilde{z} \sim p_E(\cdot|x)} [D(x, \tilde{z})] \right] + E_{z \sim p_Z} \left[E_{\tilde{x} \sim p_G(\cdot|z)} [1 - D(\tilde{x}, z)] \right], \end{aligned}$$

где $p_E(\cdot | x)$ – распределение, моделируемое кодировщиком,
 $p_G(\cdot | z)$ – распределение, моделируемое генератором

- В двунаправленных генеративных состязательных сетях одновременно с генератором обучается кодировщик, который моделирует распределение $p_E(z|x) = \delta(z - E(x))$
- Дискриминатор обеспечивает оценку вероятности $p_D(y|x, z)$, где $y = 1$, если x – пример реальных данных, и $y = 0$, если x – сгенерированный образец



Другие виды генеративных состязательных сетей

- ❑ Множество генеративных состязательных сетей не ограничивается перечисленными в настоящей классификации
- ❑ Рассмотренные виды являются широко используемыми, на основании этих моделей разрабатываются различные модификации
- ❑ Наряду с этим, существуют специализированные генеративные состязательные сети, решающие узкие задачи
- ❑ Далее рассматриваются примеры задач и генеративных состязательных сетей



ПРИМЕРЫ ПРИЛОЖЕНИЙ ГЕНЕРАТИВНЫХ СОСТЯЗАТЕЛЬНЫХ СЕТЕЙ



Увеличение количества данных

- ❑ **Увеличение количества данных** (data augmentation) – генерация синтетических данных, похожих на данные в некоторой существующей выборке, но содержащих различные трансформации, с целью их последующего использования, как правило, для расширения тренировочного набора данных
- ❑ **CycleGAN** [<https://www.nature.com/articles/s41598-019-52737-x>] (2019) – задача сегментации данных компьютерного томографии
- ❑ **Data Augmentation GAN (DAGAN)** [<https://arxiv.org/pdf/1711.04340.pdf>] (2018) – увеличение количества данных за счет трансформаций
- ❑ **Balancing GAN (BAGAN)** [<https://arxiv.org/pdf/1803.09655.pdf>] (2018) – восстановление сбалансированности набора данных



Генерация изображений высокого качества

- ❑ ***Laplacian Pyramid of Adversarial Networks (LAPGAN)*** [<https://arxiv.org/pdf/1506.05751.pdf>] (2015) – повышение разрешения изображений
- ❑ ***Generative Adversarial What-Where Network (GAWWN)*** [<https://arxiv.org/pdf/1610.02454.pdf>] (2016) – генерация изображений с использованием дополнительного описания и информации о расположении объектов
- ❑ ***Generative Adversarial Network for image Super-Resolution (SRGAN)*** [<https://arxiv.org/pdf/1609.04802.pdf>] (2017) – повышение разрешения изображения
- ❑ ***Self-Attention Generative Adversarial Network (SAGAN)*** [<https://arxiv.org/pdf/1805.08318.pdf>] (2019) – моделирование зависимостей между отдельными частями изображения для генерации изображений высокого разрешения



Восстановление фрагментов изображений (1)

- **Восстановление фрагментов изображений** (image inpainting) – удаление нежелательных объектов на изображении или восстановление поврежденных частей старых фотографий

Реальное фото Входное фото Результат



* Yeh R.A., et al. Semantic Image Inpainting With Deep Generative Models // In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2017. – P. 5485-5493. – [\[http://openaccess.thecvf.com/content_cvpr_2017/html/Yeh_Semantic_Image_Inpainting_CVPR_2017_paper.html\]](http://openaccess.thecvf.com/content_cvpr_2017/html/Yeh_Semantic_Image_Inpainting_CVPR_2017_paper.html).

Восстановление фрагментов изображений (2)

- ❑ ***Semantic Image Inpainting with Deep Generative Models*** [http://openaccess.thecvf.com/content_cvpr_2017/html/Yeh_Semantic_Image_Inpainting_CVPR_2017_paper.html] (2017) – восстановление поврежденных фрагментов фотографий (лиц, автомобилей, естественных надписей)
- ❑ ***EdgeConnect*** [<https://arxiv.org/pdf/1901.00212.pdf>] (2019) – восстановление поврежденных фрагментов фотографий (лиц, естественных изображений)
- ❑ ***PEPSI*** [http://openaccess.thecvf.com/content_CVPR_2019/html/Sagong_PEPSI_Fast_Image_Inpainting_With_Parallel_Decoding_Network_CVPR_2019_paper.html] (2019) – снижение количества вычислений при восстановлении поврежденных фрагментов фотографий средствами генеративных состязательных сетей



Перенос стилей (1)

- **Перенос стилей** (style transfer) – перенос стиля одного изображения на другие, например, перенос стиля рисования картин художником на фотографии



* Zhu J.-Y., Park T., Isola P., Efros A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. – 2018. – [<https://arxiv.org/pdf/1703.10593.pdf>].

Перенос стилей (2)

- ❑ **CycleGAN** [<https://arxiv.org/pdf/1703.10593.pdf>] (2018) – преобразование одного изображения в другое (image-to-image translation) в отсутствии парных данных в тренировочной выборке, используемой в процессе обучения модели (перенос стиля художника на фотографию, перенос зимних эффектов на летнюю фотографию и т.п.)
- ❑ **StyleGAN** [<https://arxiv.org/pdf/1812.04948.pdf>] (2019) – генерация реалистичных лиц и перенос стиля (очки, длинные волосы и прочее) с одного лица на другое
- ❑ **BigGAN** [<https://arxiv.org/pdf/1809.11096.pdf>] (2019) – генерация реалистичных изображений естественного мира



Заключение

- ❑ Генеративные состязательные сети – глубокие модели, которые широко используются для генерации синтетических данных при решении различных задач
- ❑ Спектр задач охватывает как сферу развлечений, так и практически значимые области
- ❑ Генеративные состязательные сети активно развиваются. Появляются новые модификации этих сетей, которые содержат элементы других глубоких нейросетевых моделей



Основная литература

- ❑ Goodfellow I.J., et al. Generative Adversarial Nets // Advances in neural information processing systems. – 2014. – P. 2672-2680. – [<https://arxiv.org/pdf/1406.2661.pdf>].
- ❑ Goodfellow I.J. NIPS 2016 Tutorial: Generative Adversarial Networks. – 2016. – [<https://arxiv.org/pdf/1701.00160.pdf>].
- ❑ Alqahtani H., Kavakli-Thorne M., Kumar G. Applications of Generative Adversarial Networks (GANs): An Updated Review // Archives of Computational Methods in Engineering. – 2019.
- ❑ Foster D. Generative Deep Learning. – O'Reilly. – 2019.
- ❑ Sandfort V., Yan K., Pickhardt P.J., Summers R.M. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks // Scientific Reports. – 2019. – [<https://www.nature.com/articles/s41598-019-52737-x>].



Авторский коллектив

- ❑ **Турлапов Вадим Евгеньевич**
д.т.н., профессор кафедры МОСТ ИИТММ ННГУ
vadim.turlapov@itmm.unn.ru
- ❑ **Васильев Евгений Павлович**
преподаватель кафедры МОСТ ИИТММ ННГУ
evgeny.vasiliev@itmm.unn.ru
- ❑ **Гетманская Александра Александровна**
преподаватель кафедры МОСТ ИИТММ ННГУ
getmanskaya.alexandra@gmail.com
- ❑ **Кустикова Валентина Дмитриевна**
к.т.н., доцент кафедры МОСТ ИИТММ ННГУ
valentina.kustikova@itmm.unn.ru

