



А.Крюков

Машинное обучение в астрофизике

Лекция 6

Генеративные состязательные сети (GAN)



В предыдущих сериях

- Лекция 1. Что такое машинное обучение и сферы его применения в науке.
- Лекция 2. Основные понятия.
- Лекция 3. Метод обратного распространения ошибок.
- Лекция 4. Деревья решений.
- Лекция 5. Сверточные сети
- Лекция 6. Гамма астрономия и МО
- **Лекция 7. Генеративные нейронные сети**



Типы обучения

- Обучение с учителем.
 - Обучающий набор включает желаемый результат
- Обучение с подкреплением.
 - Учимся действовать в условиях оценочной обратной связи (поощрения)
- Обучение без учителя.
 - Поиск структуры данных
 - Данные обучения не включают желаемые результаты



Типы генеративных моделей

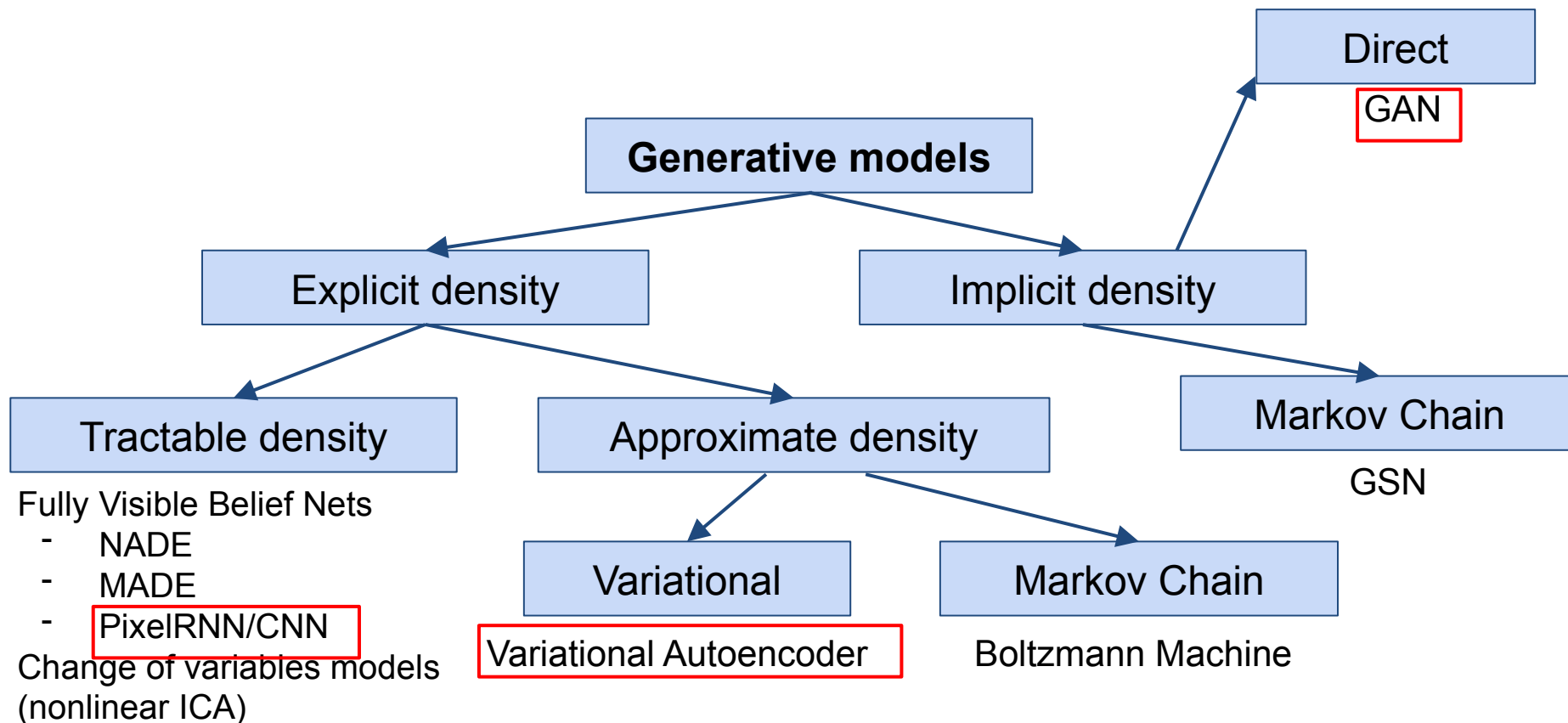


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.



PixelCNN определяют функцию плотности, оптимизируют вероятность обучающих данных:

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

VAE определяет функцию плотности со скрытыми параметрами z .

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

Невозможно оптимизировать напрямую, вместо этого можно получить и оптимизировать нижнюю границу вероятности



PixelCNN определяют функцию плотности, оптимизируют вероятность обучающих данных:

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

VAE определяют трудноразрешимую функцию плотности со скрытым z .

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

Невозможно оптимизировать напрямую, вместо этого получить и оптимизировать нижнюю границу вероятности

Что, если мы откажемся от явного моделирования плотности и просто захотим сэмплировать?

GAN: не работает с какой-либо явной функцией плотности!



Generative Adversarial Networks (GANs)

- GANs использует совокупность следующих двух главных идей:
 1. Learning to Sample
 - (High-level) Connection to Inverse Transform Sampling
 2. Adversarial Training



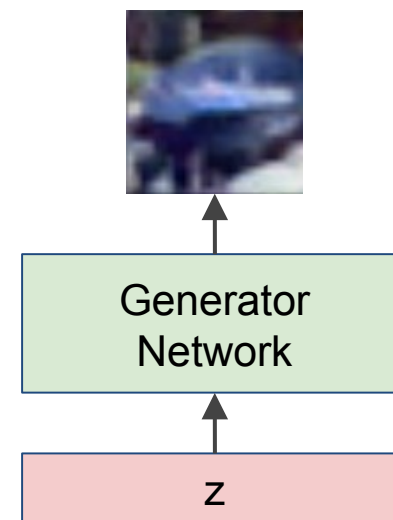
Generative Adversarial Networks

- Проблема: хотите получить выборку из сложного многомерного обучающего распределения. Нет прямого способа сделать это!
- Решение: Мспользовать образцы из простого набора, например, случайный шум. Найдем преобразование в распределение обучающей выборки.

Q: What can we use to represent this complex transformation?

A: A neural network!

Output: Sample from training distribution

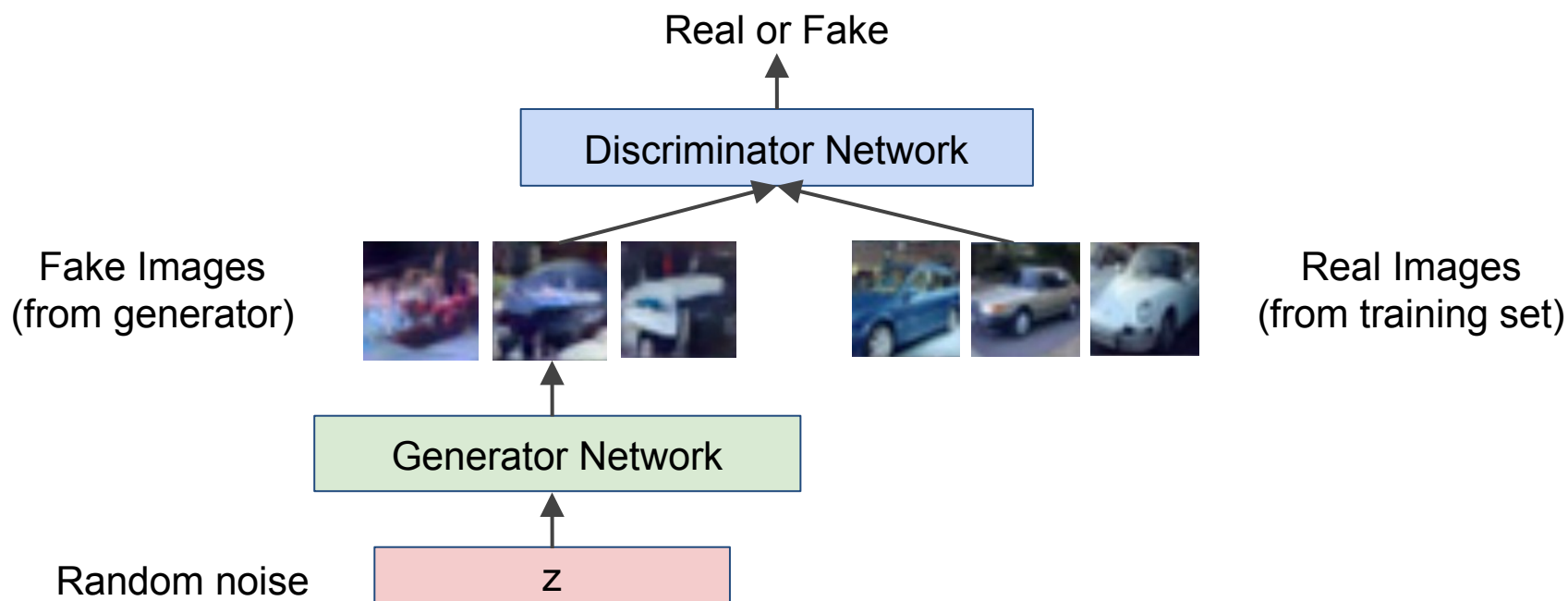


Input: Random noise



Обучение GAN — это игра двух игроков

- Генератор: попытайтесь обмануть дискриминатор, создав реалистичные изображения.
- Дискриминатор: попробуйте отличить настоящие изображения от поддельных.





Обучение GAN — это игра двух игроков

- Генератор: попытайтесь обмануть дискриминатор, создав реалистичные изображения.
- Дискриминатор: попробуйте отличить настоящие изображения от поддельных.

Train jointly in **minimax game**

Minimax objective function:

Discriminator outputs likelihood in (0,1) of real image

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data } x} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data } G(z)}) \right]$$



Обучение GAN — это игра двух игроков

- Генератор: попытайтесь обмануть дискриминатор, создав реалистичные изображения.
- Дискриминатор: попробуйте отличить настоящие изображения от поддельных.

Train jointly in **minimax game**

Minimax objective function:

Discriminator outputs likelihood in (0,1) of real image

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \underbrace{\log D_{\theta_d}(x)}_{\text{Discriminator output for real data } x} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data } G(z)}) \right]$$

- Discriminator (θ_d) wants to **maximize objective** such that $D(x)$ is close to 1 (real) and $D(G(z))$ is close to 0 (fake)
-
- Generator (θ_g) wants to **minimize objective** such that $D(G(z))$ is close to 1 (discriminator is fooled into thinking generated $G(z)$ is real)



Обучение GAN — это игра двух игроков

Minimax objective function:

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

Alternate between:

1. **Gradient ascent** on discriminator

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

2. **Gradient descent** on generator

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



Обучение GAN — это игра двух игроков

Minimax objective function:

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

Alternate between:

1. **Gradient ascent** on discriminator

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

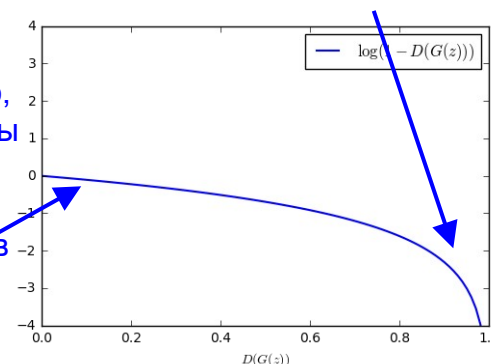
Градиентный сигнал преобладает в области, где образец уже хороший

2. **Gradient descent** on generator

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

На практике оптимизация этой цели генератора не работает!

Когда образец, вероятно, является поддельным, вы хотите извлечь из него уроки, чтобы улучшить генератор. Но градиент в этой области относительно пологий!





Обучение GAN — это игра двух игроков

Minimax objective function:

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

Alternate between:

1. **Gradient ascent** on discriminator

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

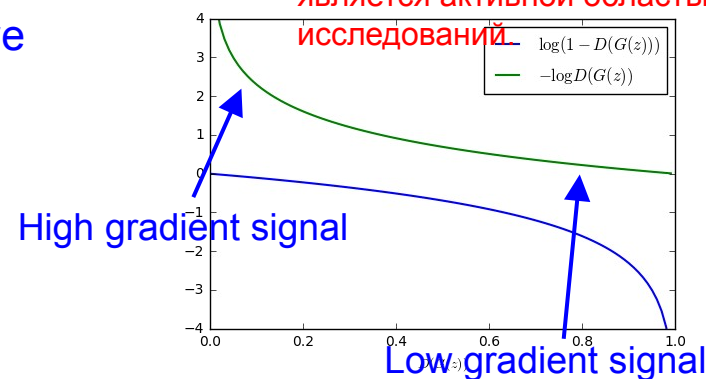
2. **Instead: Gradient ascent** on generator, **different objective**

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

Вместо того, чтобы сводить к минимуму вероятность того, что дискриминатор окажется правильным, теперь максимизируйте вероятность того, что дискриминатор окажется неправильным.

Та же цель обмануть дискриминатор, но теперь более высокий градиент сигнала для плохих образцов => работает намного лучше! Стандарт на практике.

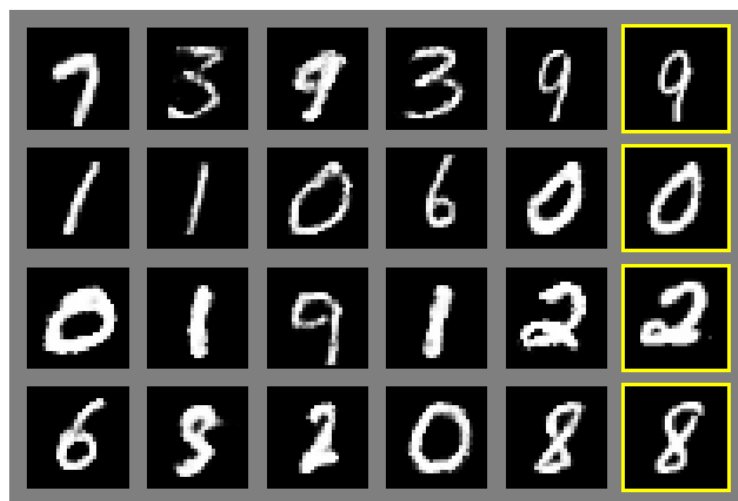
Кроме того: совместное обучение двух сетей является сложной задачей, может быть нестабильной. Выбор целей с лучшими ландшафтами потерь помогает обучению и является активной областью исследований.





Generative Adversarial Nets

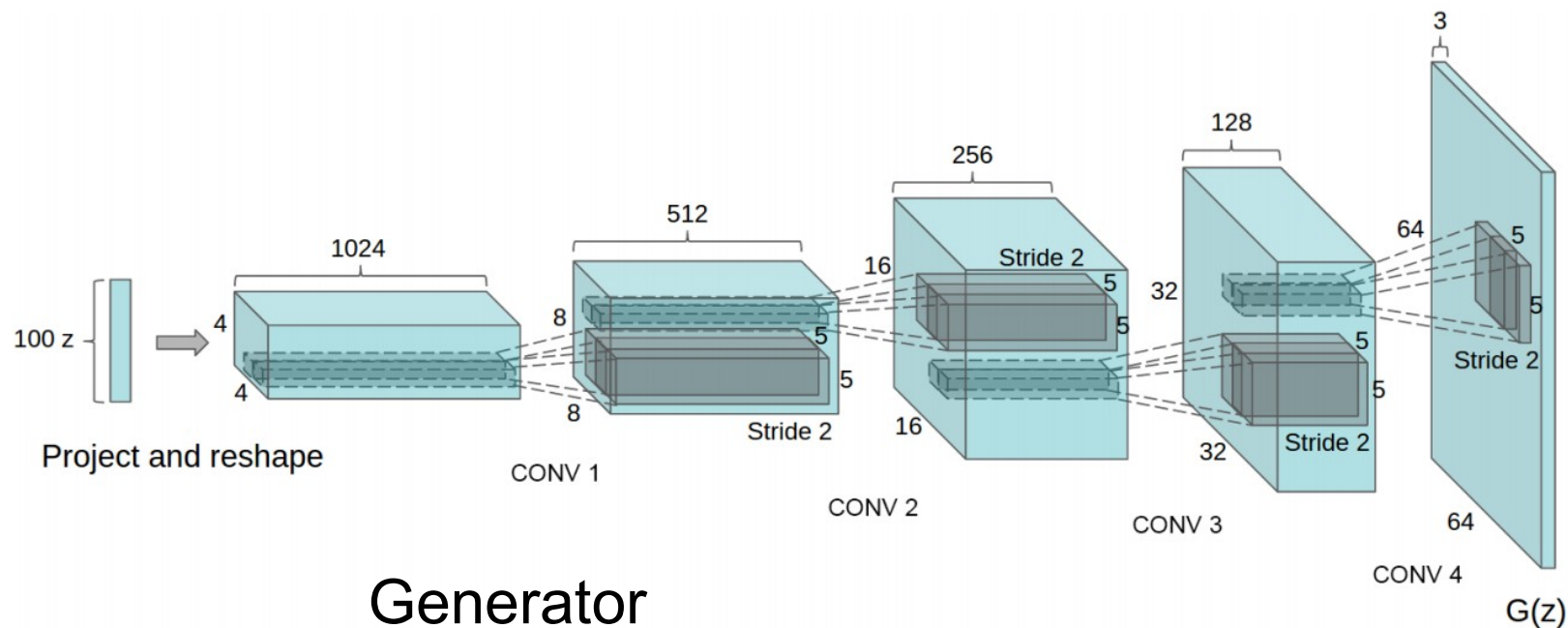
Generated samples



Nearest neighbor from training set

Figures copyright Ian Goodfellow et al., 2014. Reproduced with permission.

Generative Adversarial Nets: Convolutional Architectures



Generator

Radford et al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ICLR 2016



Результаты от года к году



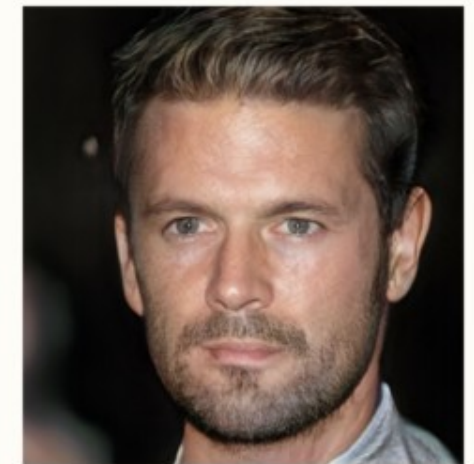
2014



2015



2016



2017



BigGAN



Large Scale GAN Training for High Fidelity Natural Image Synthesis
Andrew Brock, Jeff Donahue, Karen Simonyan <https://arxiv.org/abs/1809.11096>

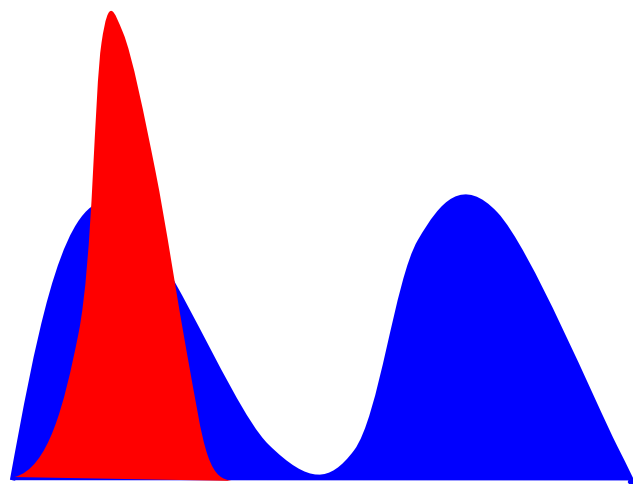


BigGAN



Коллапс мод

Generated
Distribution



Data Distribution

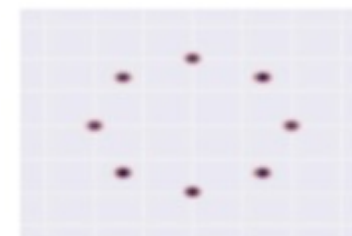


Иногда это трудно обнаружить, так как видно только то, что сгенерировано, а не то, что пропущено.



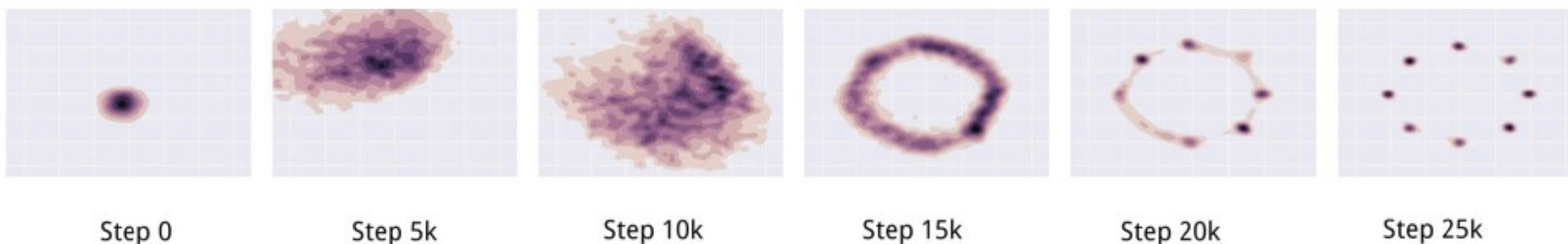
Коллапс мод: 8 гауссианов

P_{data}

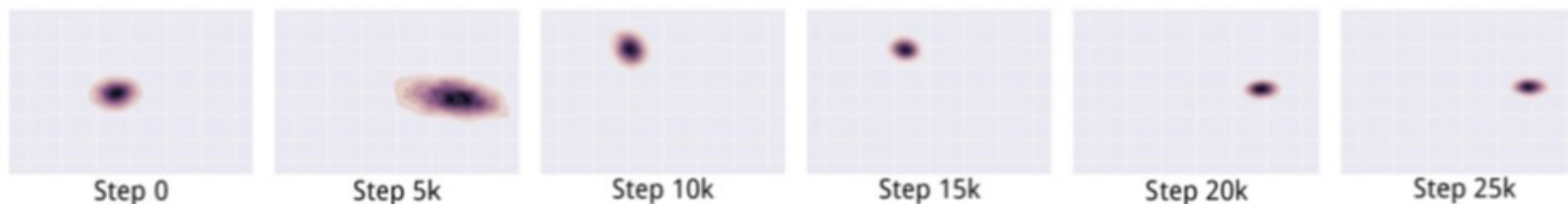


Target

Мы ожидаем



Вот что имеем в реальности ...





Другие GAN

- Conditional GANs
- Constrained GANs
- Wasserstein Distance GANs

Но об этом в следующий раз.



Материалы лекций:

https://theory.sinp.msu.ru/doku.php/ml_lectures