академия
больших
данных

mail.ru
group

Генеративно-состязательные нейронные сети. Введение

Фёдор Киташов

Программист-исследователь в команде компьютерного зрения

План лекции

- Мотивация и цели это лекции
  - Какие задачи решают генеративные сети
  - Эволюция генеративных сетей за последние 6 лет
- Статья “Generative Adversarial Nets” by I. Goodfellow
  - Архитектура: генератор и дискриминатор
  - Функция потерь и обучение
  - Проблемы сходимости и их решения.
  - Оптимальные дискриминатор и генератор

План лекции

- Проблемы обучения ганов
- Mode Collapse
- Wasserstein GAN
  - Earth Mover’s Distance
  - Непрерывность по Липшицу
  - Примеры работы

На следующей лекции:

- Современные подходы к генерации изображений
  - Progressive growing of GANs
  - BigGAN
  - AdaIN слои
  - StyleGAN и StyleGAN2
- Conditional GANs
- Метрики для сравнения результатов генерации

Прогресс


2014

GAN


2015

DCGAN


2016

CGAN


2017

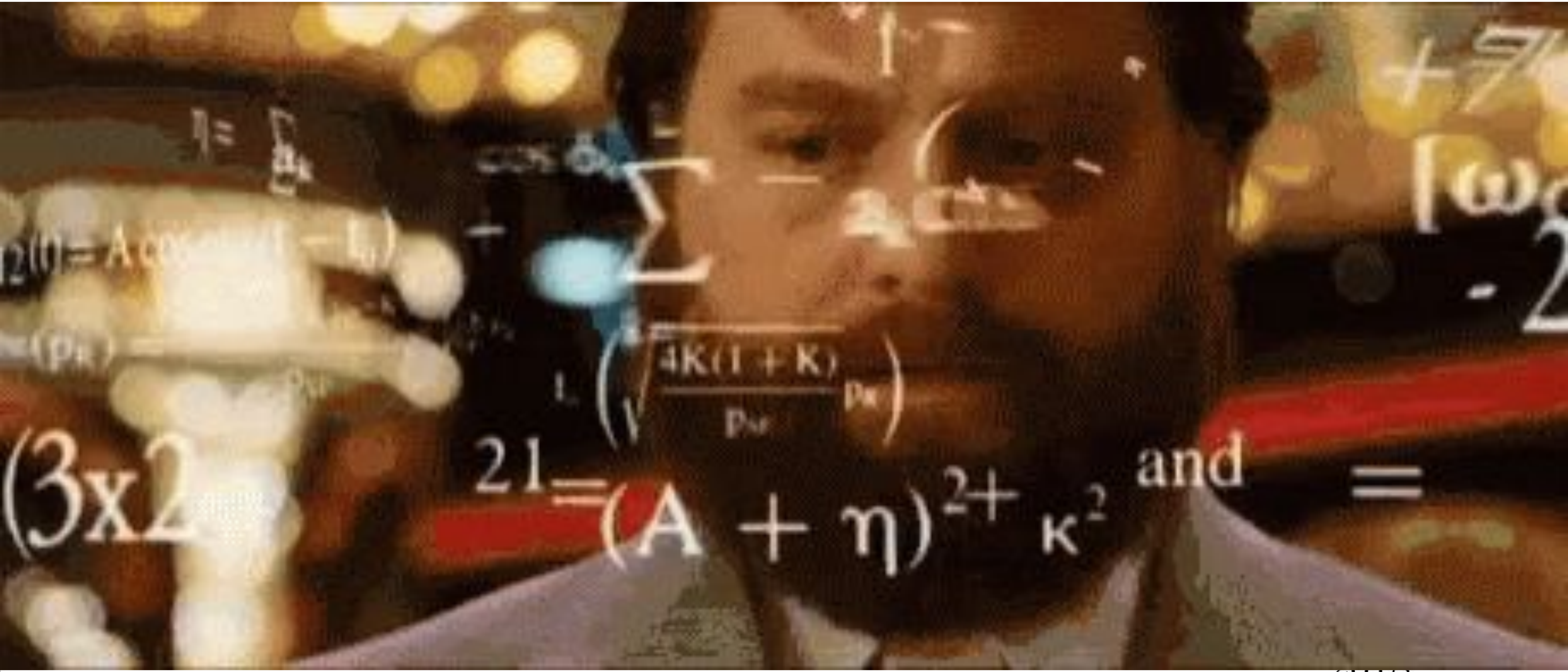
PGGAN


2018

StyleGAN

https://arxiv.org/abs/1406.2661 https://arxiv.org/abs/1511.06434 https://arxiv.org/abs/1606.07536 https://arxiv.org/abs/1710.10196 https://arxiv.org/abs/1812.04948

Прогресс



https://arxiv.org/abs/1406.2661 https://arxiv.org/abs/1511.06434 https://arxiv.org/abs/1606.07536 https://arxiv.org/abs/1710.10196 https://arxiv.org/abs/1812.04948

Generative Adversarial Nets

- Статья “Generative Adversarial Nets” by I. Goodfellow
  - Архитектура: генератор и дискриминатор
  - Функция потерь и обучение
  - Оптимальные дискриминатор и генератор
  - Проблемы сходимости и их решения.

https://arxiv.org/pdf/1406.2661.pdf

Архитектура: генератор и дискриминатор

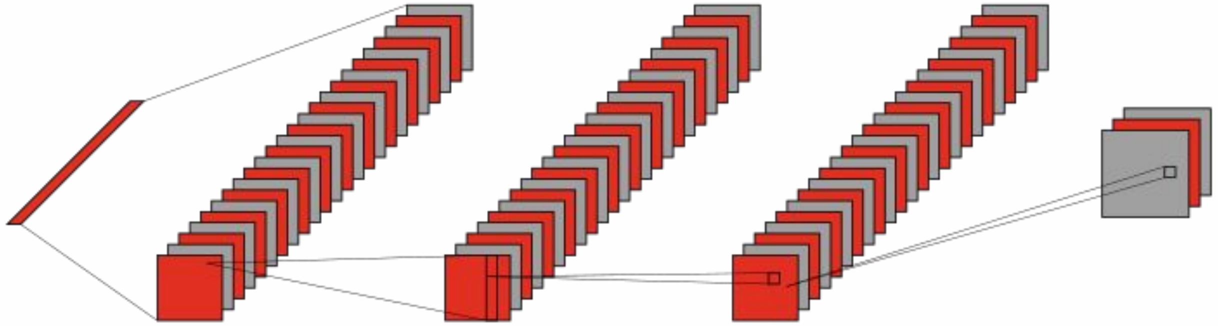
- Генератор:
  - нейронная сеть
  - генерирует изображения из случайного шума

Архитектура: генератор и дискриминатор

- ❖ Генератор:
  - ➢ нейронная сеть
  - ➢ генерирует изображения из случайного шума
- ❖ Дискриминатор:
  - ➢ нейронная сеть
  - ➢ учится как бинарный классификатор
  - ➢ учится отличать настоящие изображения от сгенерированных

Архитектура: генератор и дискриминатор

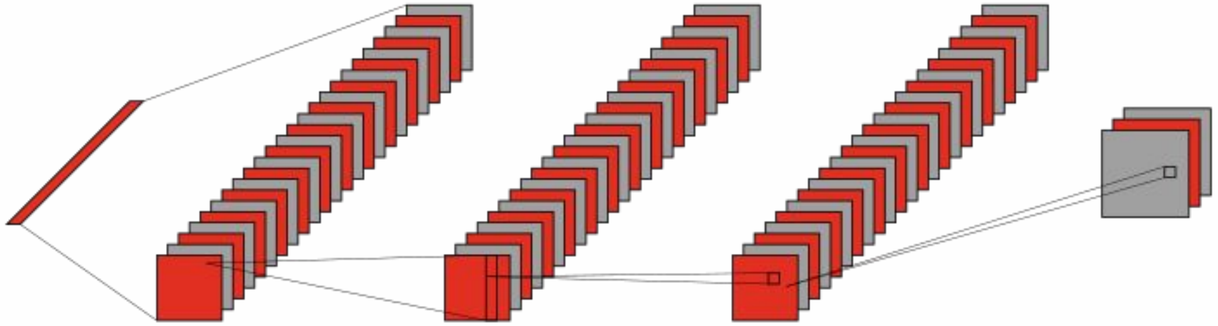
- Генератор:
  - на вход: вектор случайных чисел из равномерного или нормального распределения
  - на выход: изображение

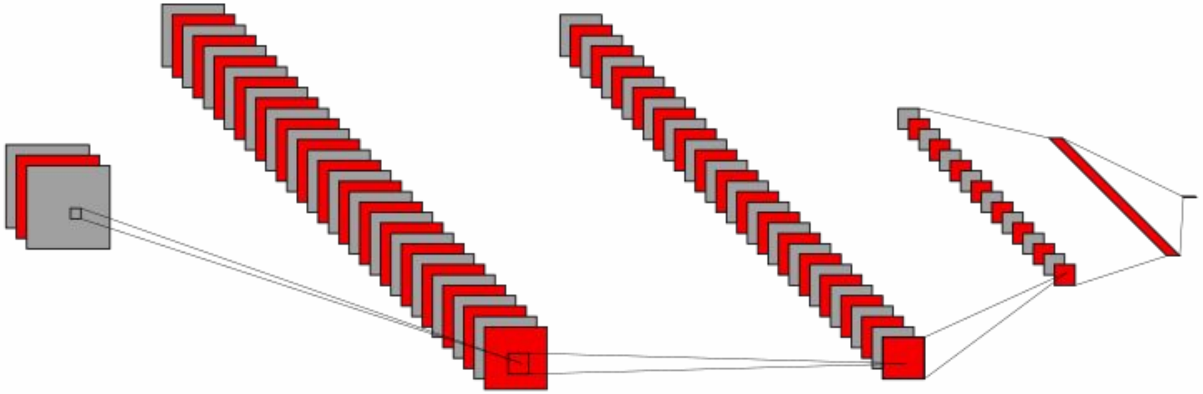


https://github.com/goodfeli/adversarial/blob/1720041f51efcb92df61d1638f1a74aea2ae74c2/__init__.py#L199

Архитектура: генератор и дискриминатор

- Генератор:
  - на вход: вектор случайных чисел из равномерного или нормального распределения
  - на выход: изображение
- Дискриминатор:
  - на вход: изображение
  - на выход: вероятность того, что изображение настоящее

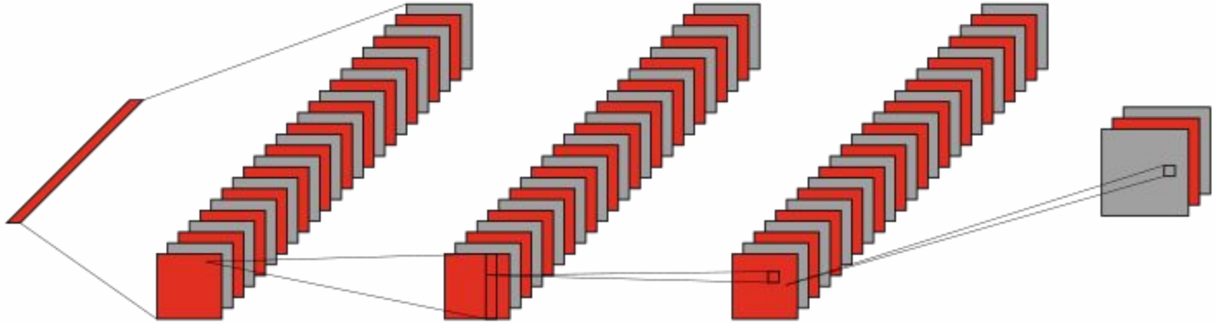




https://arxiv.org/pdf/1406.2661.pdf

Архитектура: генератор и дискриминатор

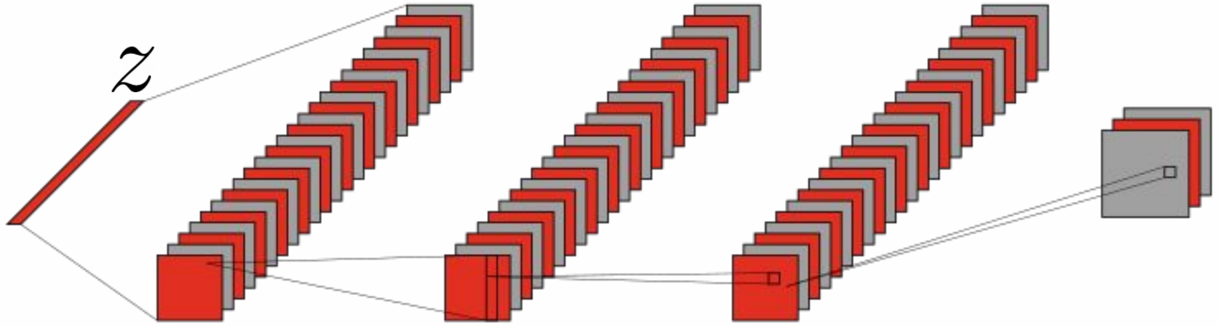
- ❖ Генератор (дифференцируемая функция G)



https://arxiv.org/pdf/1406.2661.pdf

Архитектура: генератор и дискриминатор

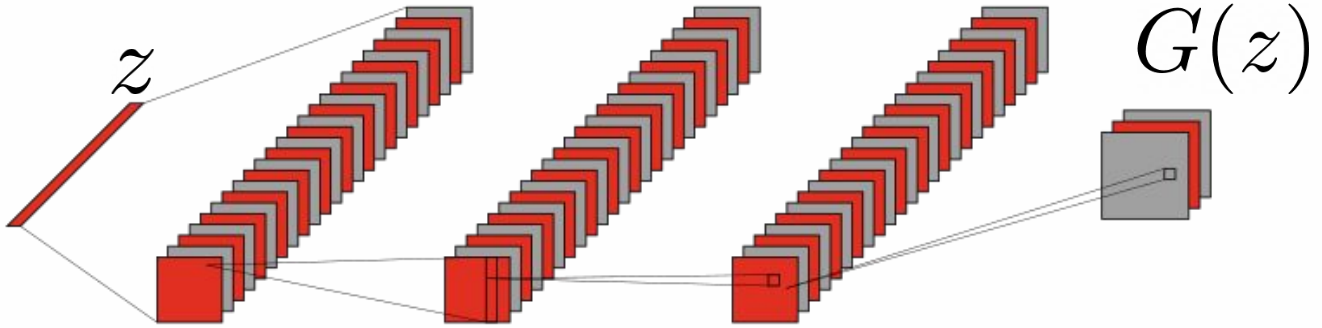
- Генератор (дифференцируемая функция G)
  - на вход: вектор шума ***z***



https://arxiv.org/pdf/1406.2661.pdf

Архитектура: генератор и дискриминатор

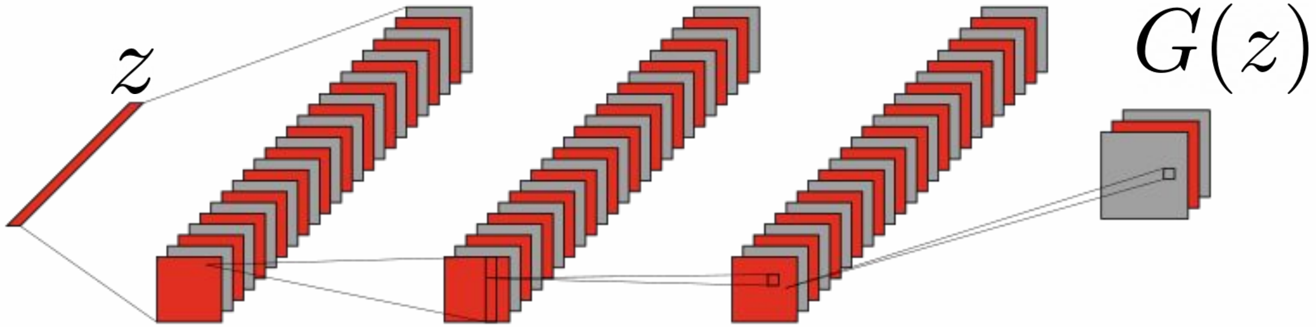
- Генератор (дифференцируемая функция G)
  - на вход: вектор шума ***z***
  - на выход: изображение ***G(z)***



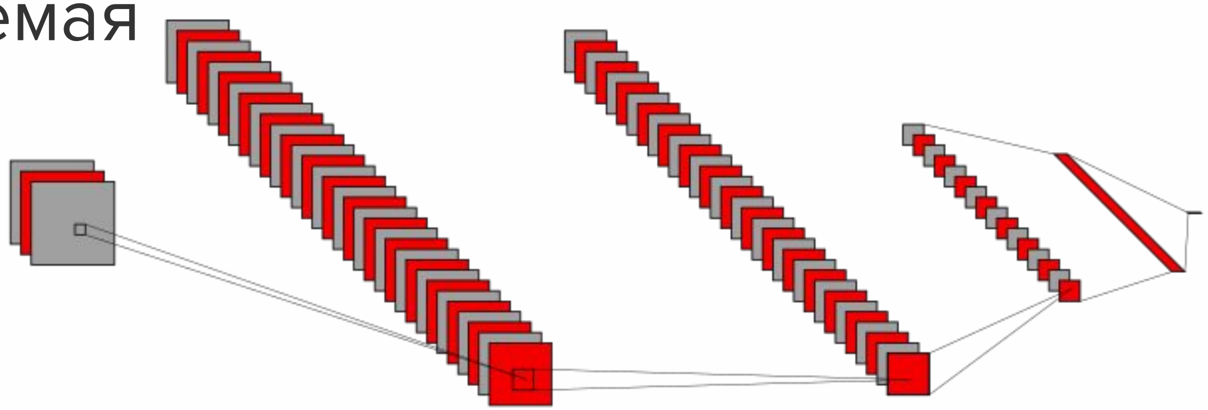
https://arxiv.org/pdf/1406.2661.pdf

Архитектура: генератор и дискриминатор

- Генератор (дифференцируемая функция G)
  - на вход: вектор шума ***z***
  - на выход: изображение ***G(z)***



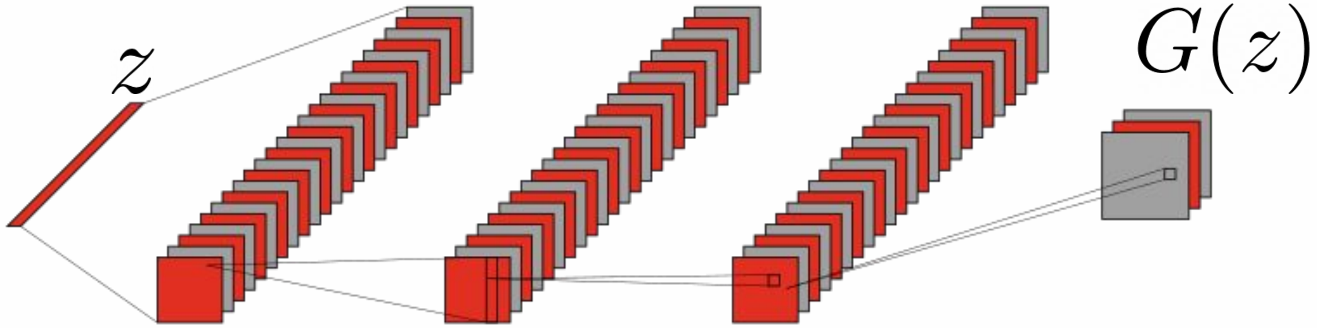
- Дискриминатор (дифференцируемая функция D)



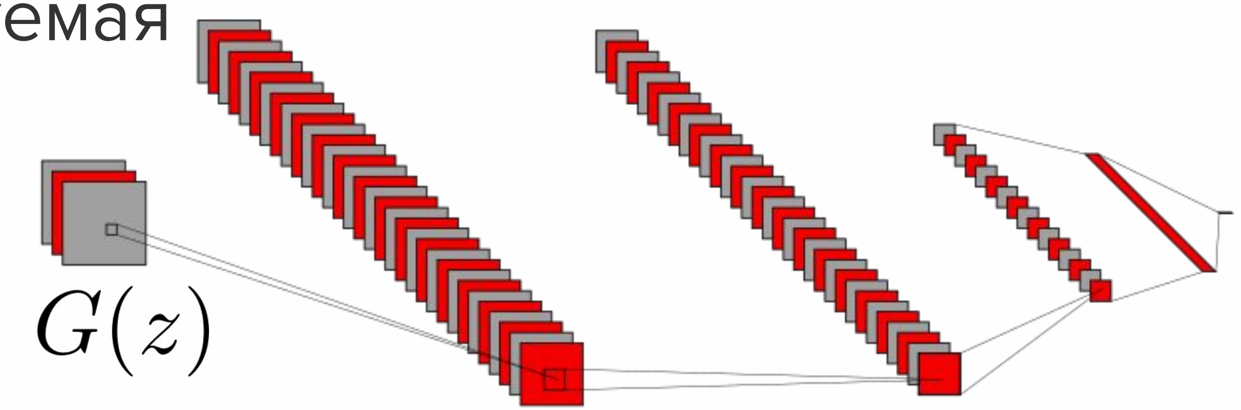
https://arxiv.org/pdf/1406.2661.pdf

Архитектура: генератор и дискриминатор

- Генератор (дифференцируемая функция G)
  - ➢ на вход: вектор шума ***z***
  - ➢ на выход: изображение ***G(z)***



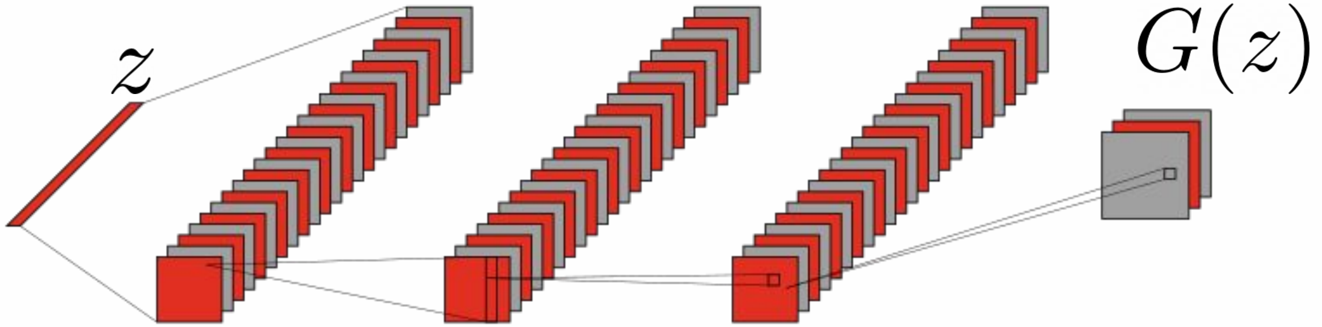
- Дискриминатор (дифференцируемая функция D)
  - ➢ на вход: изображение ***G(z)***



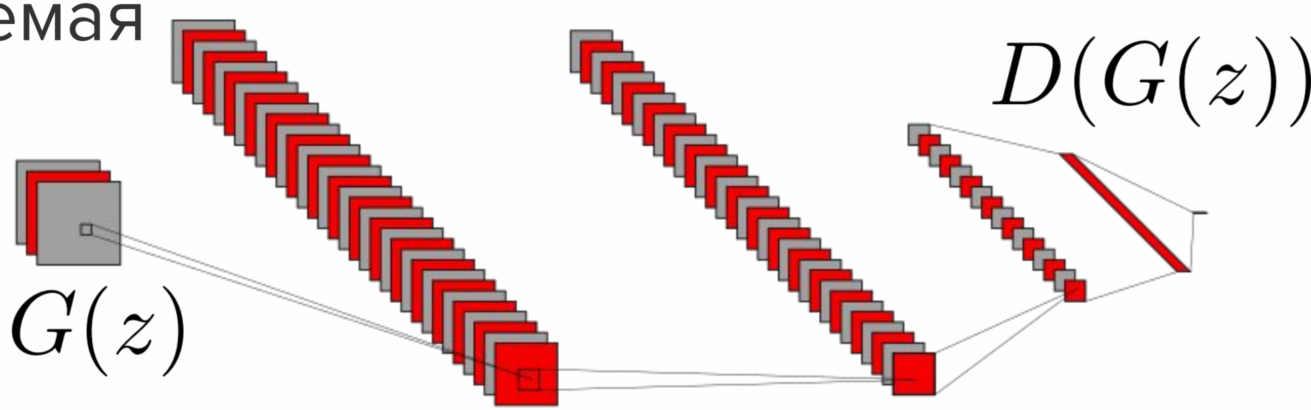
https://arxiv.org/pdf/1406.2661.pdf

Архитектура: генератор и дискриминатор

- Генератор (дифференцируемая функция G)
  - на вход: вектор шума ***z***
  - на выход: изображение ***G(z)***



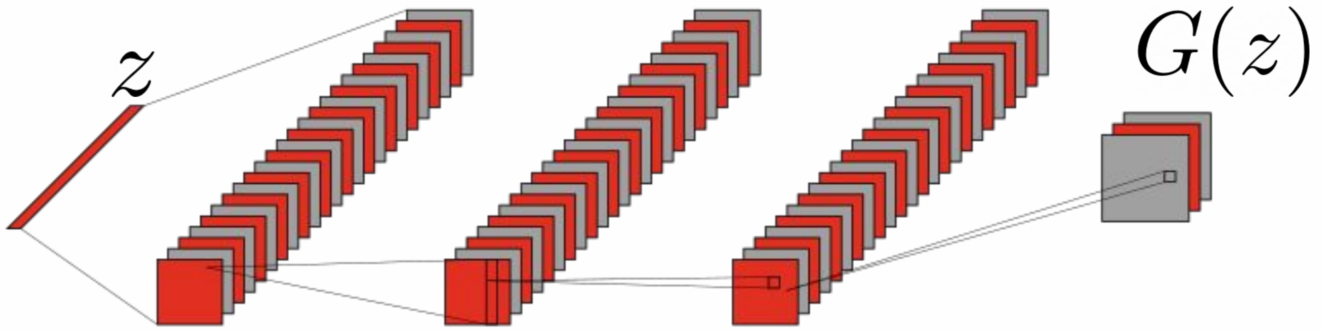
- Дискриминатор (дифференцируемая функция D)
  - на вход: изображение ***G(z)***
  - на выход: число ***D(G(z))***



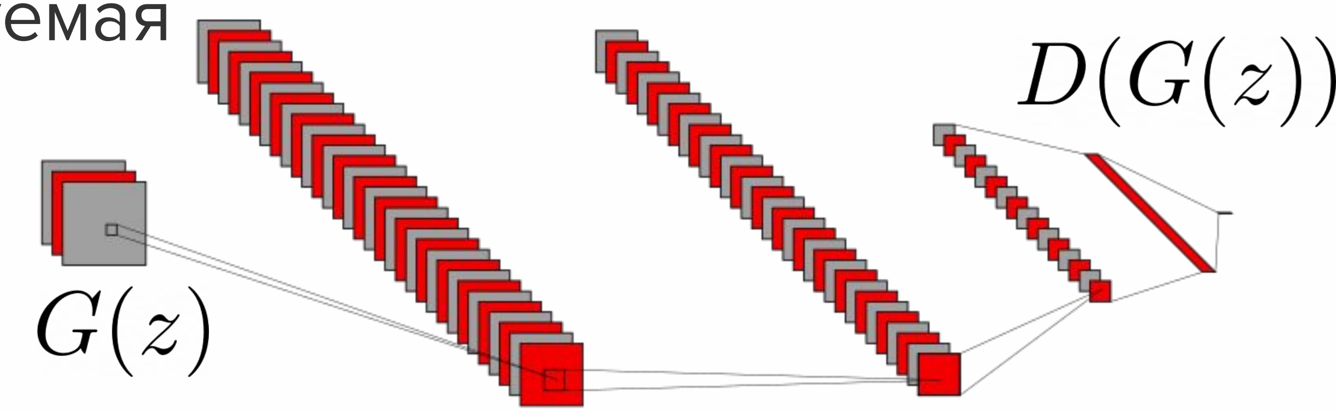
https://arxiv.org/pdf/1406.2661.pdf

Архитектура: генератор и дискриминатор

- Генератор (дифференцируемая функция G)
  - на вход: вектор шума ***z***
  - на выход: изображение ***G(z)***



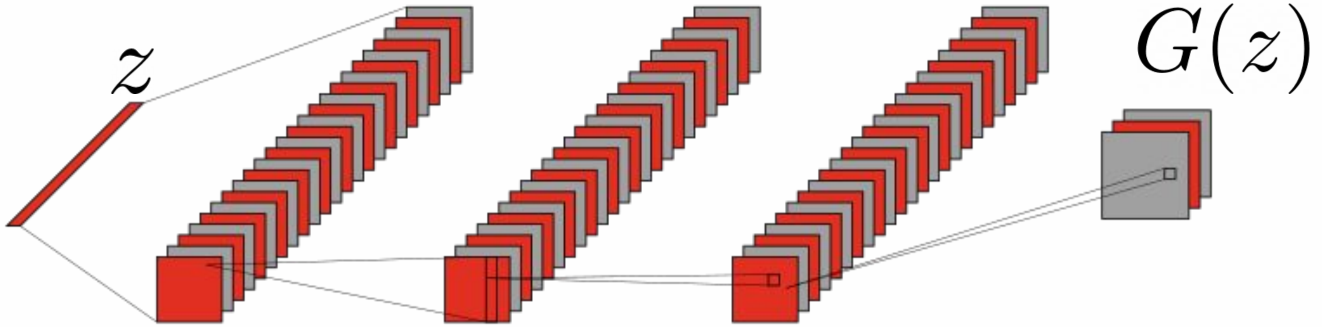
- Дискриминатор (дифференцируемая функция D)
  - на вход: изображение ***G(z)***
  - на выход: число ***D(G(z))***
  - Дискриминатор D пытается сделать ***D(G(z)) = 0***



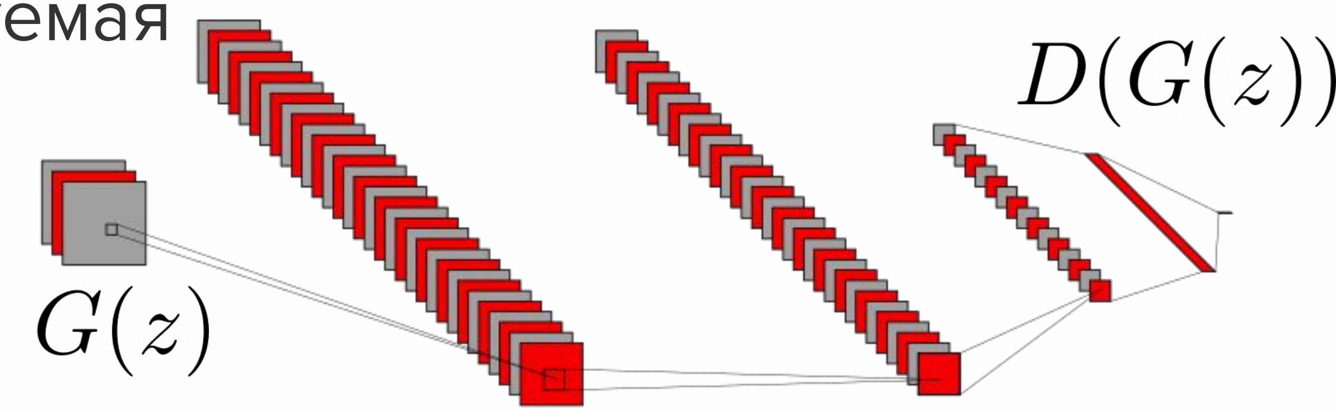
https://arxiv.org/pdf/1406.2661.pdf

Архитектура: генератор и дискриминатор

- Генератор (дифференцируемая функция G)
  - на вход: вектор шума ***z***
  - на выход: изображение ***G(z)***
  - Генератор G пытается сделать ***D(G(z)) = 1***



- Дискриминатор (дифференцируемая функция D)
  - на вход: изображение ***G(z)***
  - на выход: число ***D(G(z))***
  - Дискриминатор D пытается сделать ***D(G(z)) = 0***



https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\min_G \max_D V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$

- Учим дискриминатор правильно отличать примеры из датасета и из G(z)

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\min_G \boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$

- Учим дискриминатор правильно отличать примеры из датасета и из G(z)
- Дискриминатор максимизирует выражение V(D, G)

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\min_G \boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}}$$

- Учим дискриминатор правильно отличать примеры из датасета и из ***G(z)***
- Дискриминатор максимизирует выражение ***V(D, G)***
- Учим генератор обманывать дискриминатор. Генератор минимизирует ***log(1 - D(G(z))***

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\boxed{\min_G}\boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}}$$

- Учим дискриминатор правильно отличать примеры из датасета и из ***G(z)***
- Дискриминатор максимизирует выражение ***V(D, G)***
- Учим генератор обманывать дискриминатор. Генератор минимизирует ***log(1 - D(G(z))***

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\boxed{\min_G}\,\boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}}$$

- Учим дискриминатор правильно отличать примеры из датасета и из ***G(z)***
- Дискриминатор максимизирует выражение ***V(D, G)***
- Учим генератор обманывать дискриминатор. Генератор минимизирует ***log(1 - D(G(z))***
- Чего добивается дискриминатор: $\max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log 1] + \mathbb{E}_{z\sim p_z(z)}[\log(1-0)] = 0 + 0 = 0$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\boxed{\min_G}\boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}}$$

- Учим дискриминатор правильно отличать примеры из датасета и из ***G(z)***
- Дискриминатор максимизирует выражение ***V(D, G)***
- Учим генератор обманывать дискриминатор. Генератор минимизирует ***log(1 - D(G(z))***
- Чего добивается дискриминатор: $\max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log 1] + \mathbb{E}_{z\sim p_z(z)}[\log(1-0)] = 0 + 0 = 0$
- Чего добивается генератор?

$$\min_G V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log 1] + \mathbb{E}_{z\sim p_z(z)}[\log(1-1)] = 0 - \infty = -\infty$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение : minimax game

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

❖ Обучение проходит в два шага

https://arxiv.org/pdf/1406.2661.pdf

Обучение : minimax game

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- Обучение проходит в два шага
- 1 шаг. Обучение дискриминатора:
  - сэмплируем настоящие картинки, считаем ошибку
  - генерируем сэмплы, считаем ошибку
  - обновляем веса дискриминатора

https://arxiv.org/pdf/1406.2661.pdf

Обучение : minimax game

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

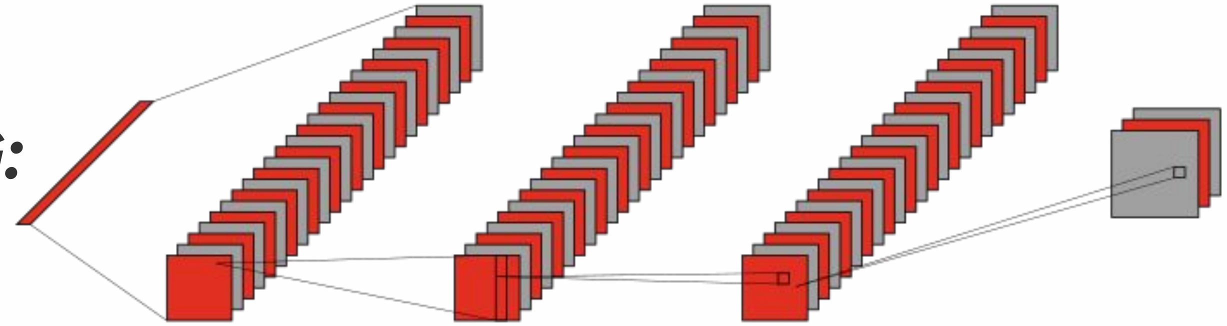
- Обучение проходит в два шага
- 1 шаг. Обучение дискриминатора:
  - сэмплируем настоящие картинки, считаем ошибку
  - генерируем сэмплы, считаем ошибку
  - обновляем веса дискриминатора
- 2 шаг. Обучение генератора:
  - генерируем сэмплы
  - считаем ошибку дискриминатора
  - пробрасываем градиенты в генератор
  - обновляем веса генератора

https://arxiv.org/pdf/1406.2661.pdf

Обучение генератора: minimax game

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

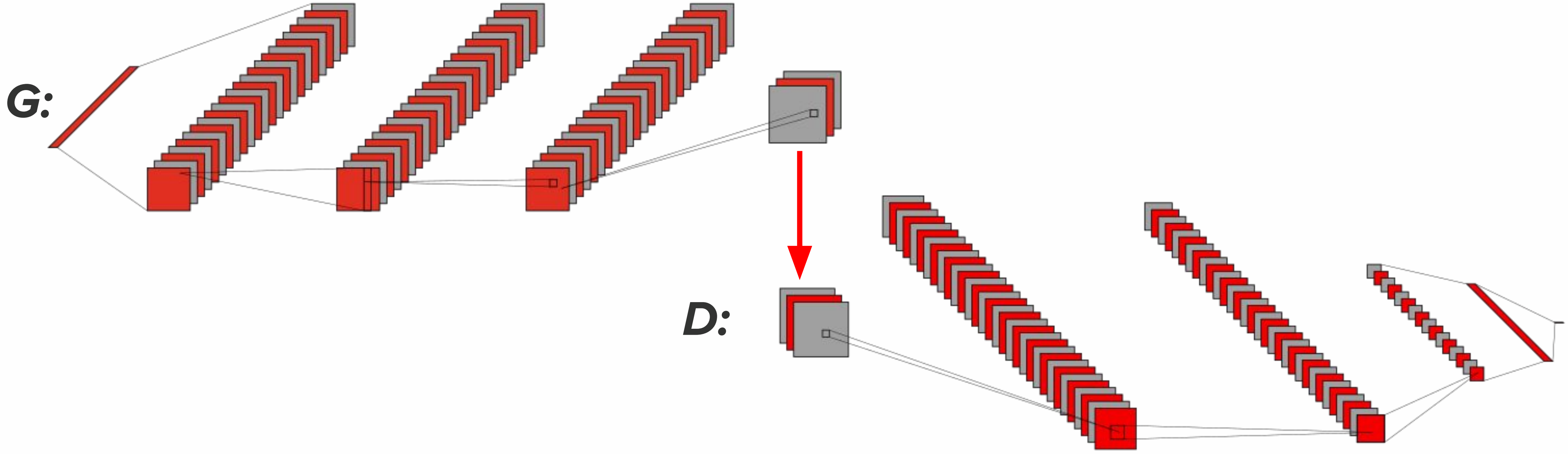
G:



https://arxiv.org/pdf/1406.2661.pdf

Обучение генератора: minimax game

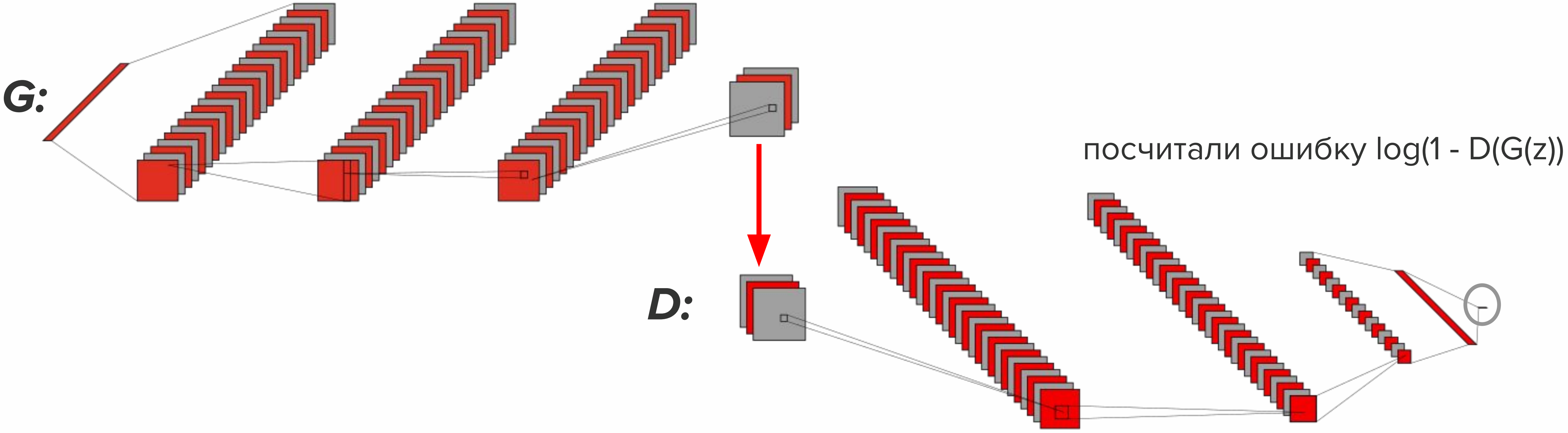
$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



https://arxiv.org/pdf/1406.2661.pdf

Обучение генератора: minimax game

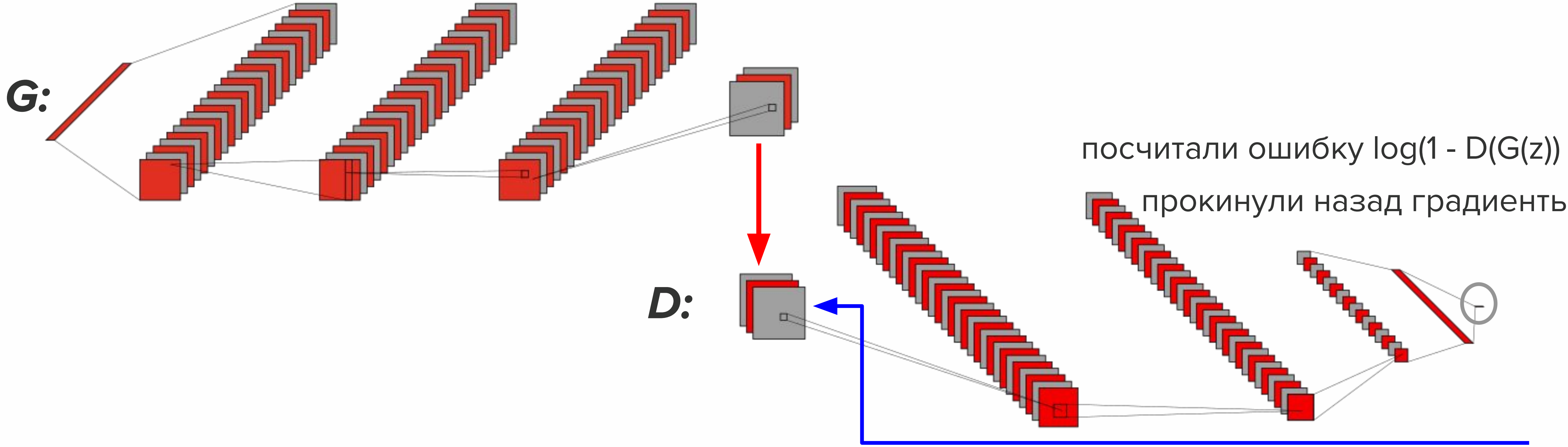
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



https://arxiv.org/pdf/1406.2661.pdf

Обучение генератора: minimax game

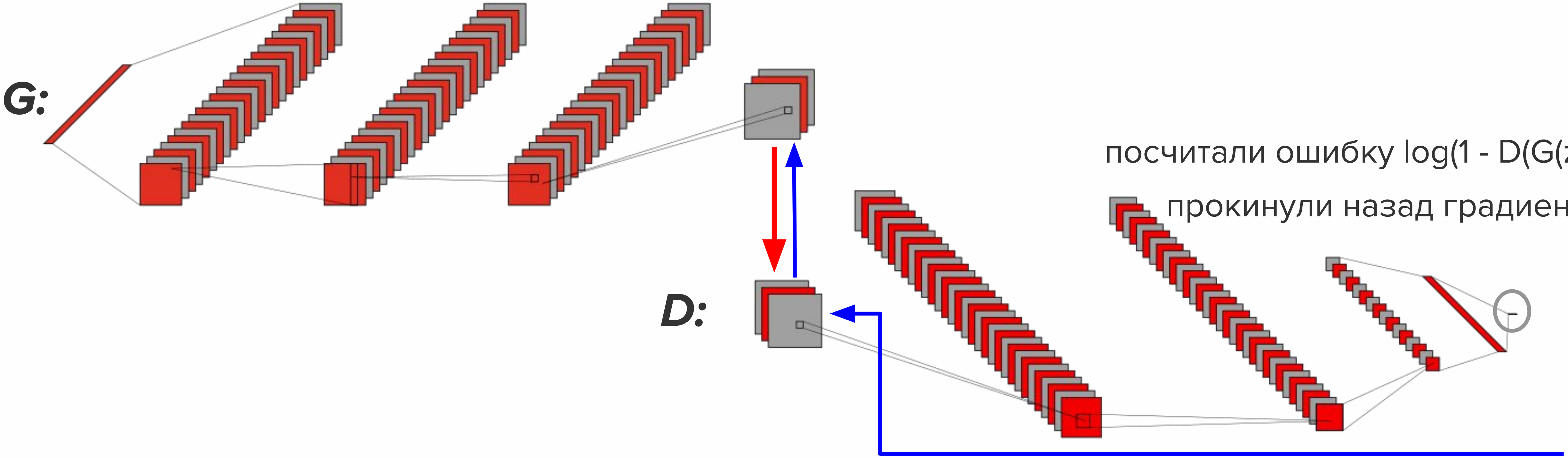
$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



https://arxiv.org/pdf/1406.2661.pdf

Обучение генератора: minimax game

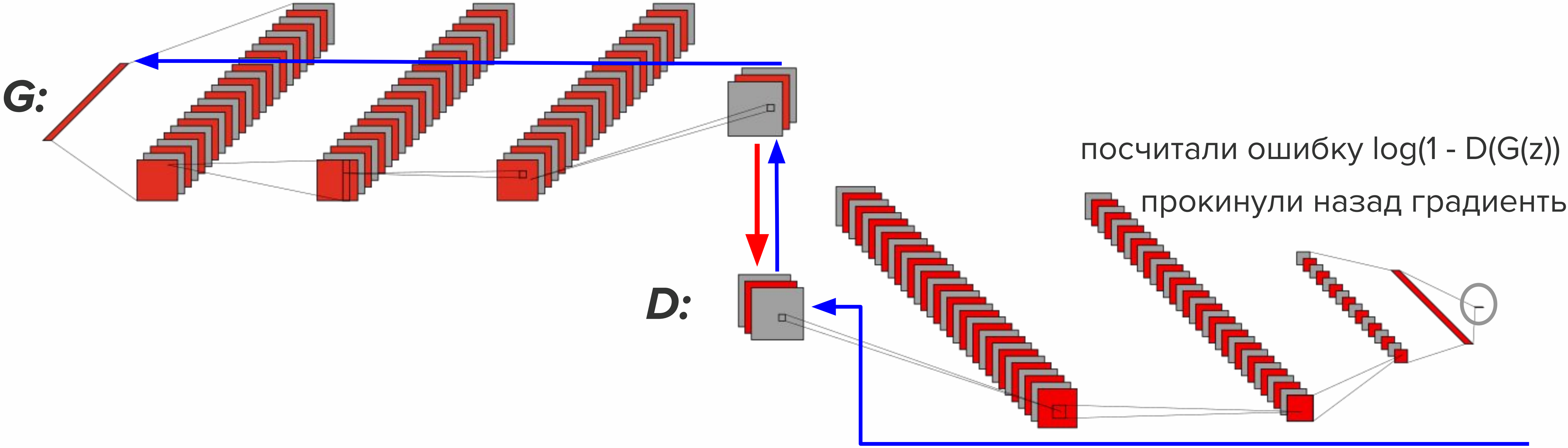
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



https://arxiv.org/pdf/1406.2661.pdf

Обучение генератора: minimax game

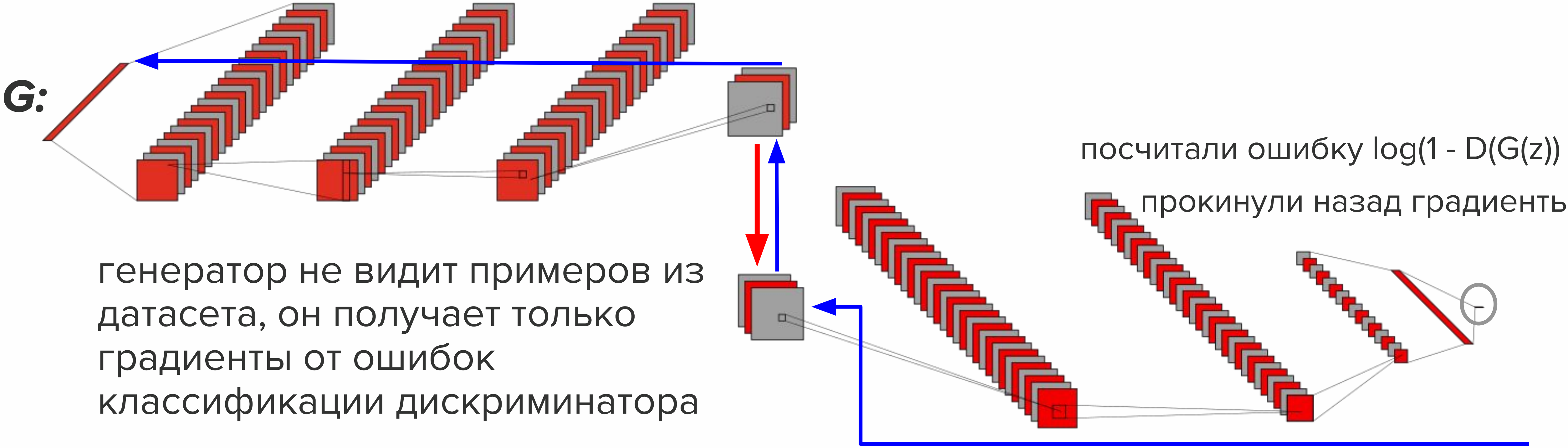
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



https://arxiv.org/pdf/1406.2661.pdf

Обучение генератора: minimax game

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



генератор не видит примеров из датасета, он получает только градиенты от ошибок классификации дискриминатора

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\boxed{\min_G}\boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}}$$

- Учим дискриминатор правильно отличать примеры из датасета и из ***G(z)***
- Дискриминатор максимизирует выражение ***V(D, G)***
- Учим генератор обманывать дискриминатор. Генератор минимизирует ***log(1 - D(G(z))***

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\boxed{\min_G} \boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}}$$

- Учим дискриминатор правильно отличать примеры из датасета и из ***G(z)***
- Дискриминатор максимизирует выражение ***V(D, G)***
- Учим генератор обманывать дискриминатор. Генератор минимизирует ***log(1 - D(G(z))***
- ***Есть проблема.***

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Затухание градиентов

- Генератор G пытается сделать ***D(G(z)) = 1***
- Дискриминатор D пытается сделать ***D(G(z)) = 0***

$$\boxed{\min_G}\boxed{\max_D} V(D,G) = \boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}}$$

- Учим дискриминатор правильно отличать примеры из датасета и из ***G(z)***
- Дискриминатор максимизирует выражение ***V(D, G)***
- Учим генератор обманывать дискриминатор. Генератор минимизирует ***log(1 - D(G(z))***
- ***Есть проблема:* если D уже хорошо обучился, а G всё еще генерирует плохие картинки, слагаемое log (1 - D(G(z)) начинает давать очень маленькие градиенты.**

https://arxiv.org/pdf/1406.2661.pdf

Логистическая регрессия. Повторение

❖ Логлосс: $$-\sum_{i=1}^{\ell}\left([y_i = +1]\log b(x_i) + [y_i = -1]\log(1 - b(x_i))\right) \to \min$$

https://github.com/esokolov/ml-course-hse/blob/master/2016-fall/lecture-notes/lecture05-linclass.pdf https://arxiv.org/pdf/1406.2661.pdf

Логистическая регрессия. Повторение

- Логлосс: $$-\sum_{i=1}^{\ell}\left([y_i = +1]\log b(x_i) + [y_i = -1]\log(1 - b(x_i))\right) \to \min$$
- Если вместо b(x_i) поставить сигмоиду от <w, x>, то получается: $$\sum_{i=1}^{\ell}\log\left(1 + \exp(-y_i\langle w, x_i\rangle)\right).$$

https://github.com/esokolov/ml-course-hse/blob/master/2016-fall/lecture-notes/lecture05-linclass.pdf https://arxiv.org/pdf/1406.2661.pdf

Логистическая регрессия. Повторение

- Логлосс: $$-\sum_{i=1}^{\ell}\left([y_i = +1]\log b(x_i) + [y_i = -1]\log(1 - b(x_i))\right) \to \min$$
- Если вместо b(x_i) поставить сигмоиду от <w, x>, то получается: $$\sum_{i=1}^{\ell}\log\left(1 + \exp(-y_i\langle w, x_i\rangle)\right).$$

$$b(x) = \sigma(\langle w, x\rangle)$$

$$\sigma(z) = \frac{1}{1+\exp(-z)}$$

https://github.com/esokolov/ml-course-hse/blob/master/2016-fall/lecture-notes/lecture05-linclass.pdf https://arxiv.org/pdf/1406.2661.pdf

Логистическая регрессия. Повторение

- ❖ Логлосс: $$-\sum_{i=1}^{\ell}\left([y_i = +1]\log b(x_i) + [y_i = -1]\log(1 - b(x_i))\right) \to \min$$
- ❖ Если вместо b(x_i) поставить сигмоиду от <w, x>, то получается: $$\sum_{i=1}^{\ell}\log\left(1 + \exp(-y_i\langle w, x_i\rangle)\right).$$

https://github.com/esokolov/ml-course-hse/blob/master/2016-fall/lecture-notes/lecture05-linclass.pdf https://arxiv.org/pdf/1406.2661.pdf

Логистическая регрессия. Повторение

❖ Логлосс: $$-\sum_{i=1}^{\ell}\left([y_i = +1]\log b(x_i) + [y_i = -1]\log(1 - b(x_i))\right) \to \min$$
❖ Если вместо b(x_i) поставить сигмоиду от <w, x>, то получается: $$\sum_{i=1}^{\ell}\log\left(1 + \exp(-y_i\langle w, x_i\rangle)\right).$$
❖ Возможны два варианта:
  ➢ либо классификатор делает ошибку, и мы получаем большой градиент
  ➢ либо классификатор идеально работает, и градиент практически нулевой

https://github.com/esokolov/ml-course-hse/blob/master/2016-fall/lecture-notes/lecture05-linclass.pdf https://arxiv.org/pdf/1406.2661.pdf

Логистическая регрессия. Повторение

- Логлосс: $$-\sum_{i=1}^{\ell}\left([y_i=+1]\log b(x_i)+[y_i=-1]\log(1-b(x_i))\right)\to\min$$
- Если вместо b(x_i) поставить сигмоиду от <w, x>, то получается: $$\sum_{i=1}^{\ell}\log\left(1+\exp(-y_i\langle w,x_i\rangle)\right).$$
- Возможны два варианта:
  - либо классификатор делает ошибку, и мы получаем большой градиент
  - либо классификатор идеально работает, и градиент практически нулевой
- Сравните формулу логлосса с формулой лосса

$$\min_G\max_D V(D,G)=\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]+\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]$$

https://github.com/esokolov/ml-course-hse/blob/master/2016-fall/lecture-notes/lecture05-linclass.pdf https://arxiv.org/pdf/1406.2661.pdf

Логистическая регрессия. Повторение

$$-\sum_{i=1}^{\ell}\left([y_i = +1]\log b(x_i) + [y_i = -1]\log(1 - b(x_i))\right) \to \min$$

- Логлосс:
- Если вместо b(x_i) поставить сигмоиду от <w, x>, то получается: $\sum_{i=1}^{\ell} \log\left(1 + \exp(-y_i\langle w, x_i\rangle)\right).$
- Возможны два варианта:
  - либо классификатор делает ошибку, и мы получаем большой градиент
  - либо классификатор идеально работает, и градиент практически нулевой
- Сравните формулу логлосса с формулой лосса дискриминатора:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- В нашей minimax-игре для генератора мы получаем плохое свойство: каждый раз, когда Генератор G сильно проигрывает дискриминатору, мы получаем очень маленький градиент

https://github.com/esokolov/ml-course-hse/blob/master/2016-fall/lecture-notes/lecture05-linclass.pdf https://arxiv.org/pdf/1406.2661.pdf

Логистическая регрессия. Повторение

❖ Логлосс: $$-\sum_{i=1}^{\ell}\left([y_i = +1]\log b(x_i) + [y_i = -1]\log(1 - b(x_i))\right) \to \min$$

❖ Если вместо b(x_i) поставить сигмоиду от <w, x>, то получается: $$\sum_{i=1}^{\ell} \log\left(1 + \exp(-y_i\langle w, x_i\rangle)\right).$$

❖ Возможны два варианта:
  ➢ либо классификатор делает ошибку, и мы получаем большой градиент
  ➢ либо классификатор идеально работает, и градиент практически нулевой

❖ Сравните формулу логлосса с формулой лосса дискриминатора:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\boxed{[\log(1 - D(G(\boldsymbol{z})))]}$$

❖ В нашей minimax-игре для генератора мы получаем плохое свойство: каждый раз, когда Генератор G сильно проигрывает дискриминатору, мы получаем очень маленький градиент

https://github.com/esokolov/ml-course-hse/blob/master/2016-fall/lecture-notes/lecture05-linclass.pdf https://arxiv.org/pdf/1406.2661.pdf

Логистическая регрессия. Повторение

$$-\sum_{i=1}^{\ell}\left([y_i = +1]\log b(x_i) + [y_i = -1]\log(1 - b(x_i))\right) \to \min$$

- Логлосс:
- Если вместо b(x_i) поставить сигмоиду от <w, x>, то получается:

$$\sum_{i=1}^{\ell}\log\left(1 + \exp(-y_i\langle w, x_i\rangle)\right).$$

- Возможны два варианта:
  - либо классификатор делает ошибку, и мы получаем большой градиент
  - либо классификатор идеально работает, и градиент практически нулевой
- Сравните формулу логлосса с формулой лосса дискриминатора:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

https://github.com/esokolov/ml-course-hse/blob/master/2016-fall/lecture-notes/lecture05-linclass.pdf https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Решение проблемы

Было:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Решение проблемы

Было:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Стало:

D: $$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

G: $$\max_G \mathbb{E}_{z\sim p_z(z)} \log(D(G(z)))$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Решение проблемы

Было:

$$\boxed{\min_G}\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \boxed{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$

Стало:

D: $$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

G: $$\max_G \boxed{\mathbb{E}_{z\sim p_z(z)} \log(D(G(z)))}$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Решение проблемы

Было:

$$\boxed{\min_G} \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \boxed{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$

Стало:

_D:_ $$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

_G:_ $$\max_G \boxed{\mathbb{E}_{z\sim p_z(z)} \log(D(G(z)))}$$

вместо того, чтобы минимизировать лог-вероятность правильного ответа, мы заставляем генератор максимизировать вероятность неправильно
ответа

https://youtu.be/HGYYEUSm-0Q?t=2727 https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Решение проблемы

Было:

$$\boxed{\min_G} \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \boxed{\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$

Стало:

D: $$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

G: $$\max_G \boxed{\mathbb{E}_{z \sim p_z(z)} \log(D(G(z))}$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

Воспользовались определением мат. ожидания непрерывной случайной величины:

3 Expected value of a continuous random variable

Definition: Let X be a continuous random variable with range $[a,b]$ and probability density function $f(x)$. The expected value of X is defined by

$$E(X) = \int_a^b x f(x)\,dx.$$

https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading6a.pdf https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D, G) = \int_x p_{\text{data}}(x) \log(D(x))dx + \int_z p_z(z) \log(1 - D(G(z)))dz$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

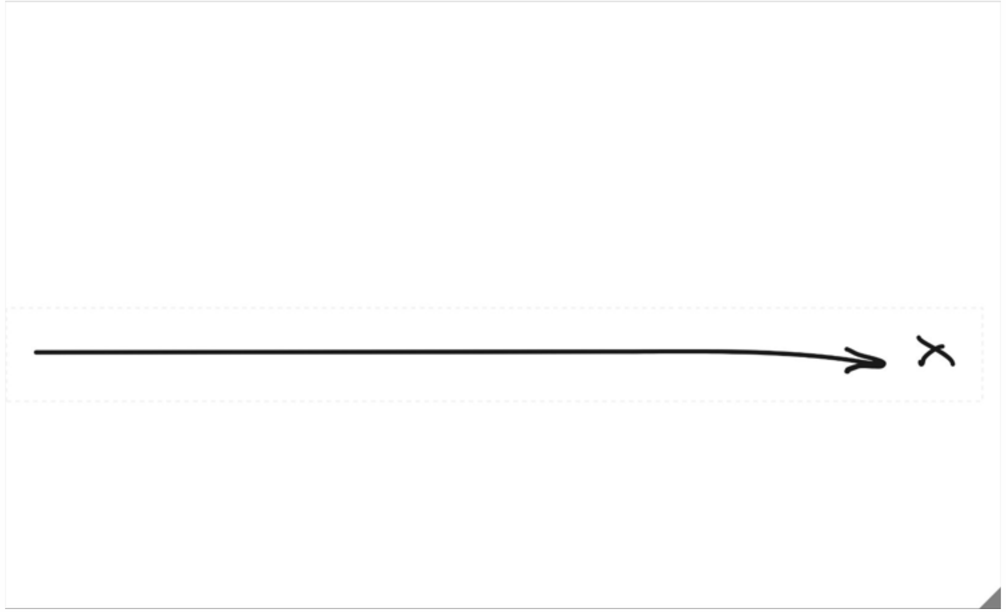
$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x) \log(D(x))dx + \int_z p_z(z) \log(1 - D(G(z)))dz$$

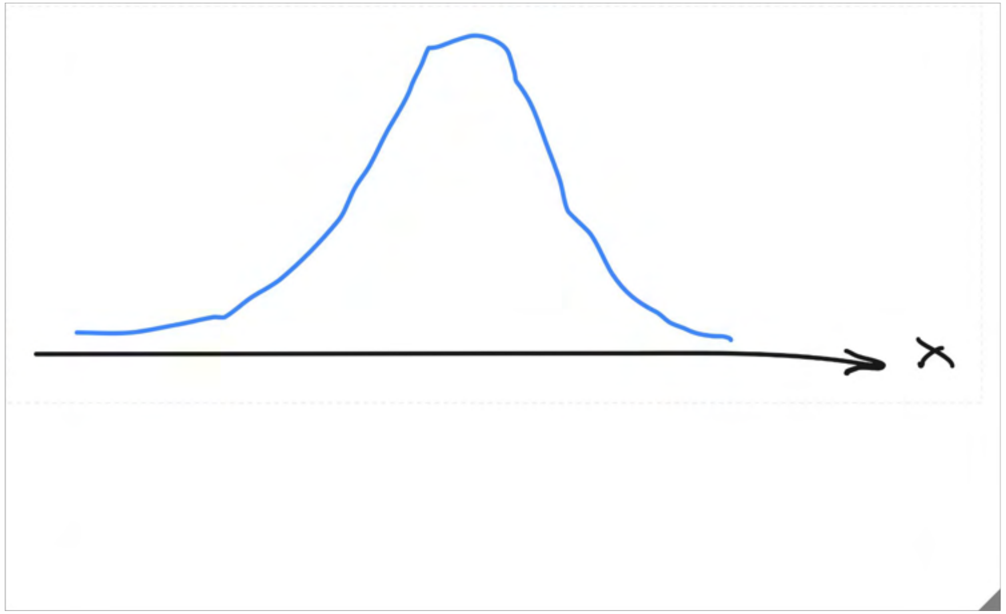


https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

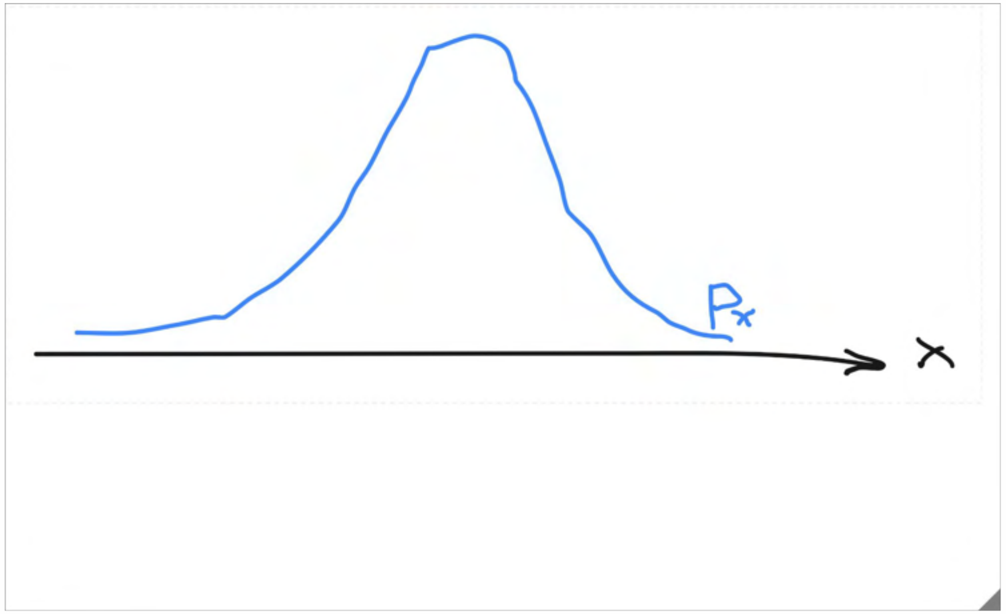


https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

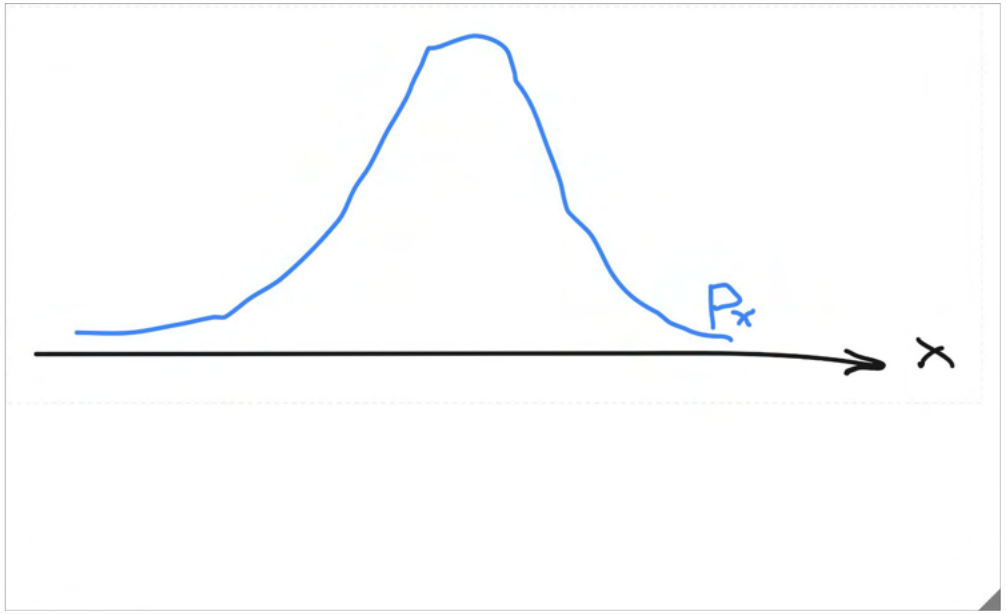


https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

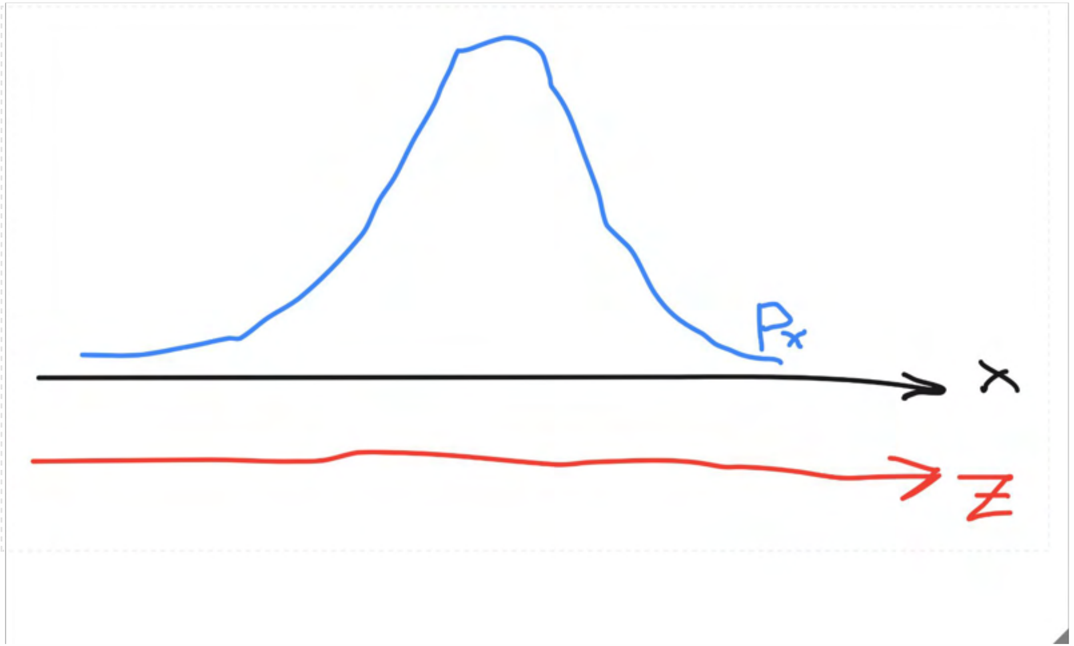


https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

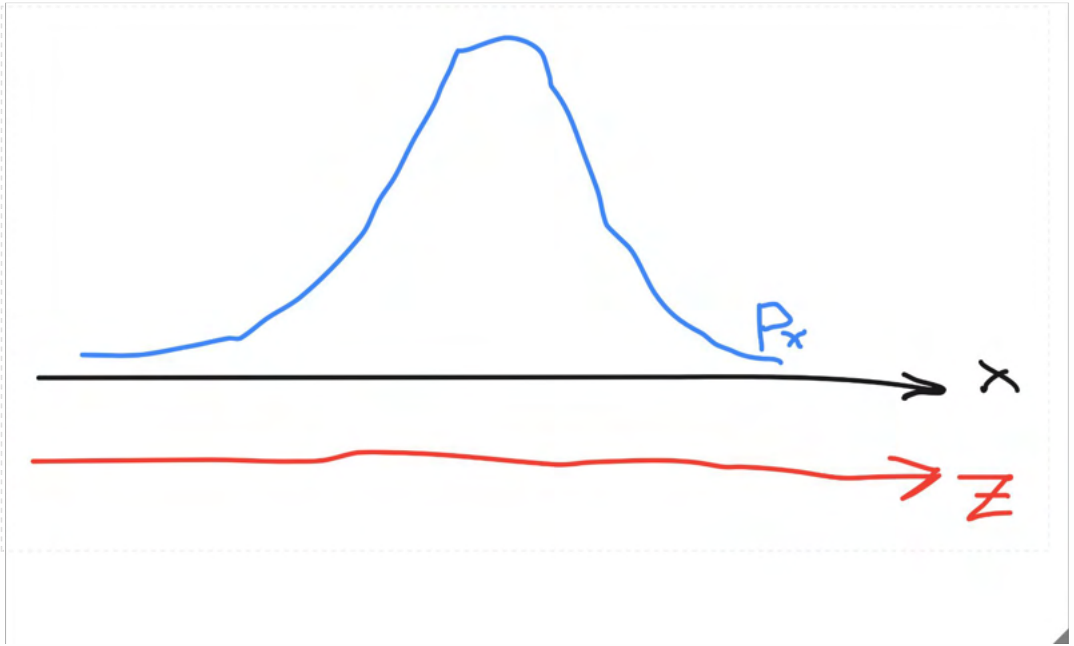


https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

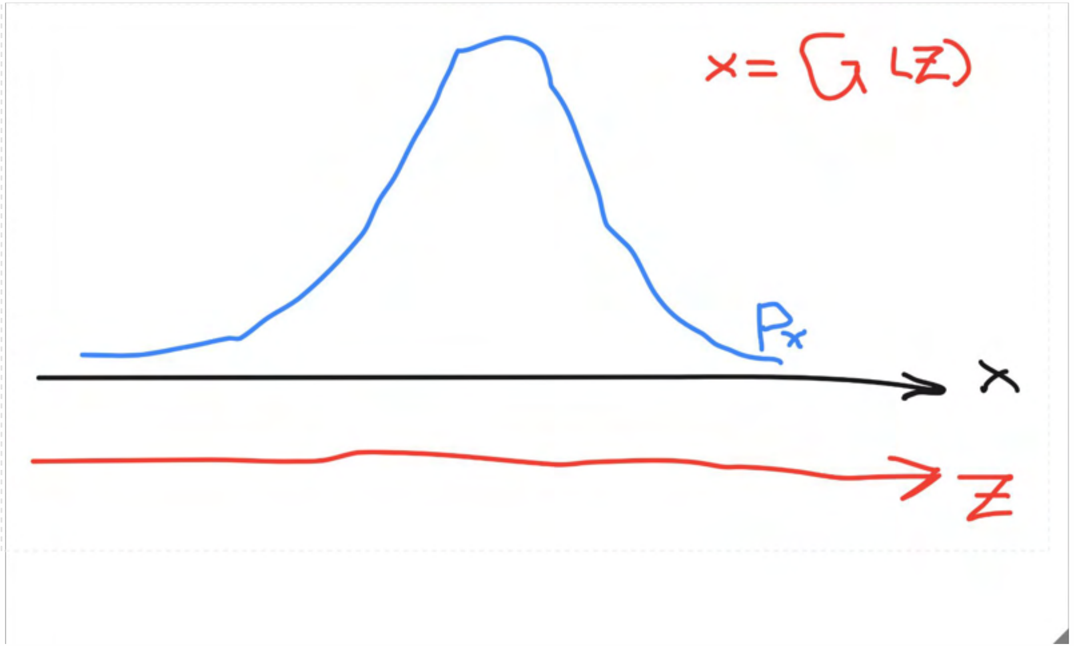


https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

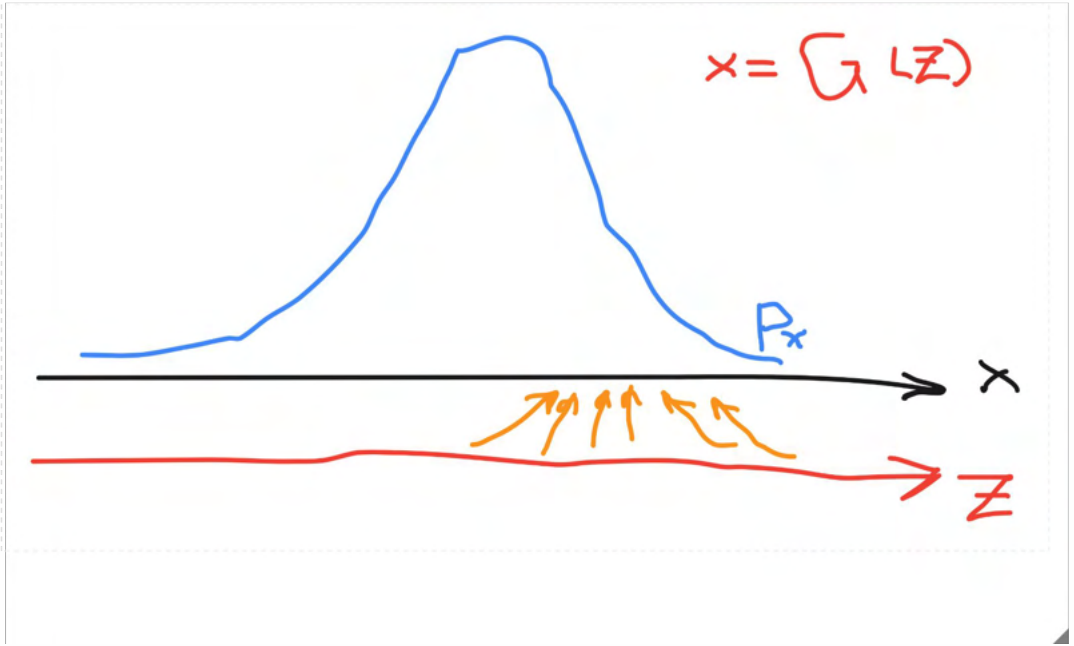


https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

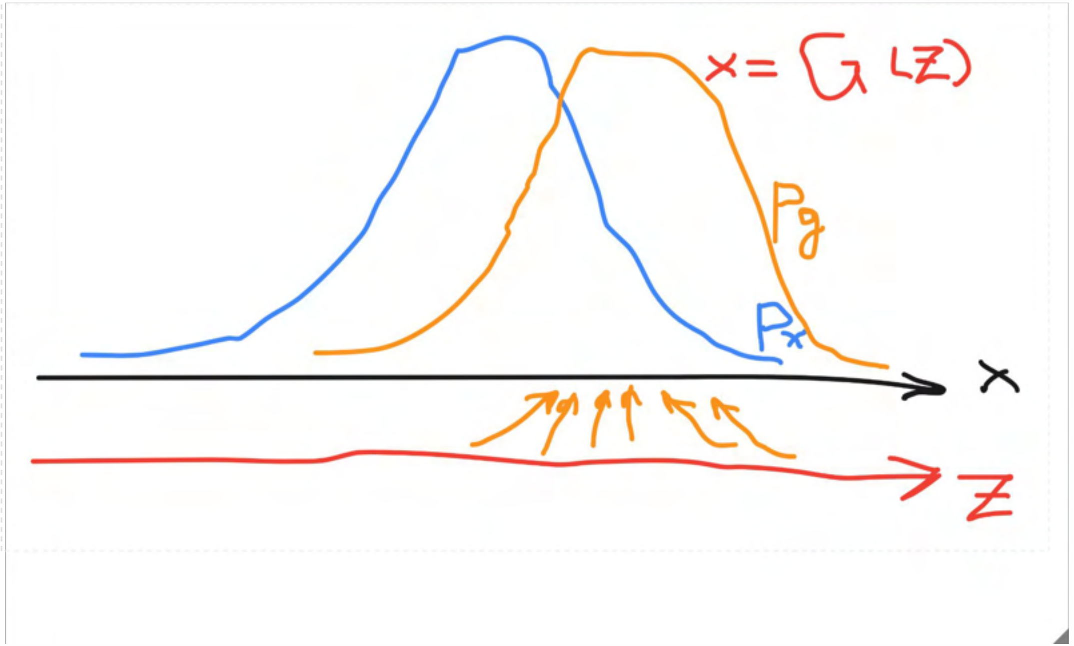


https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1-D(G(z)))dz$$

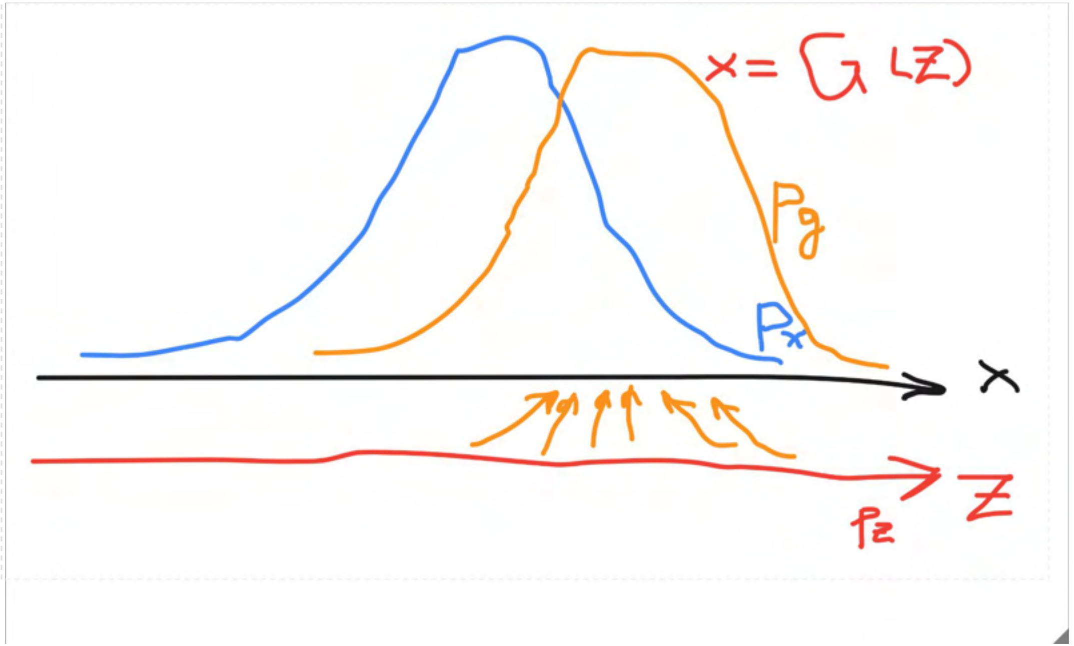


https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$



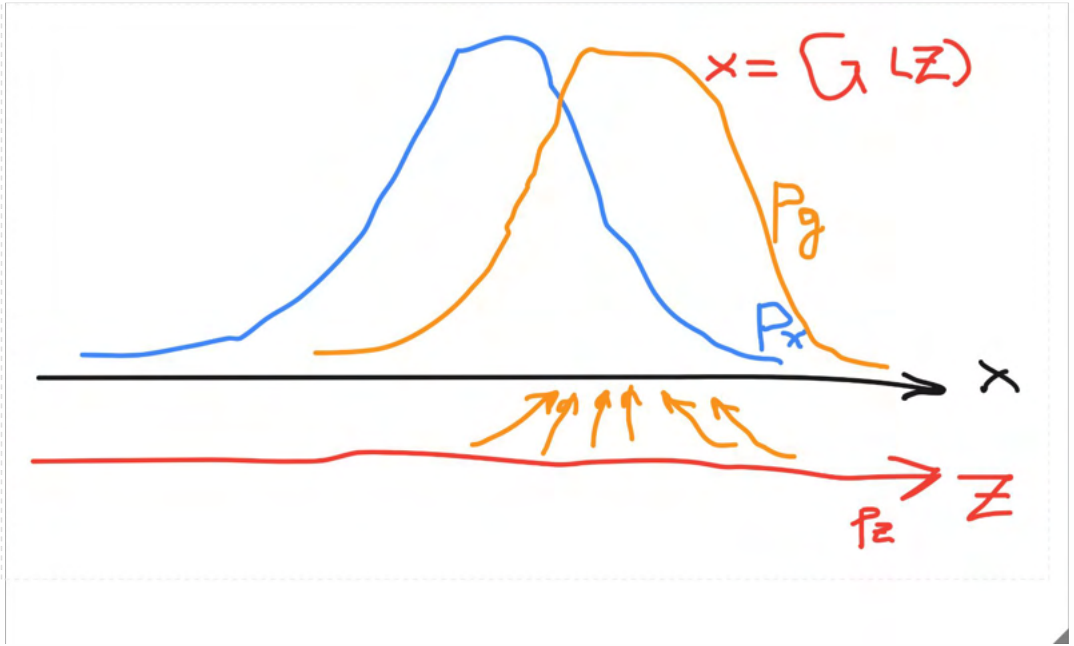
https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_x p_g(x)\log(1 - D(x))dx$$



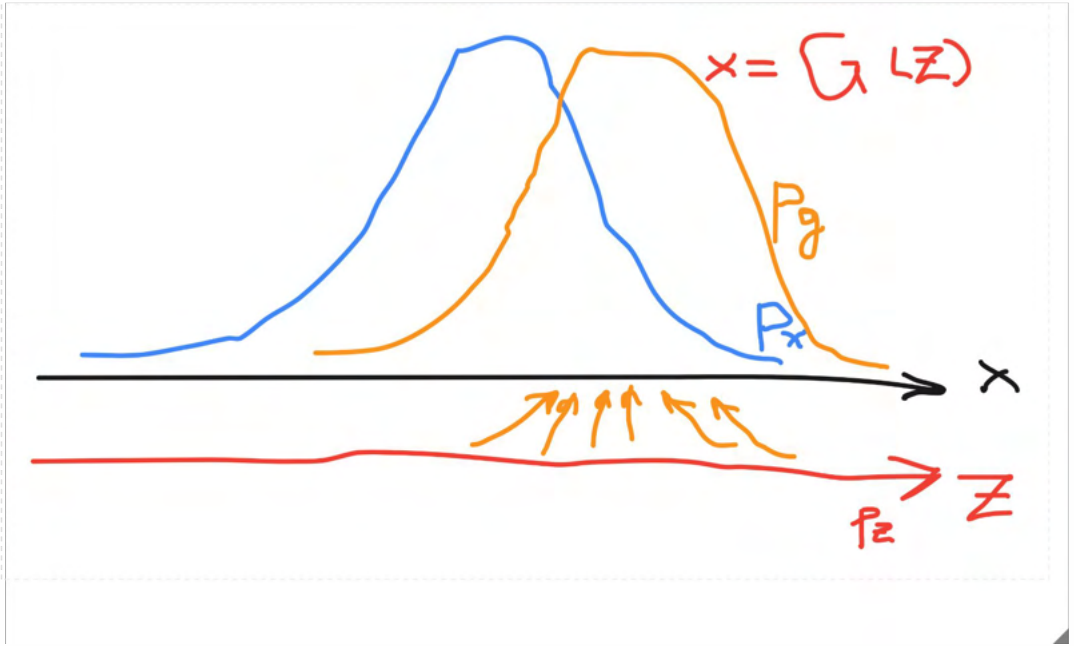
https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \boxed{\int_x p_g(x)}\log(1 - \boxed{D(x)})dx$$



https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \boxed{\int_x p_g(x)}\log(1 - \boxed{D(x))})dx$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x)) + p_g(x)\log(1 - D(x))dx$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x)) + p_g(x)\log(1-D(x))dx$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x)) + p_g(x)\log(1-D(x))dx$$

$$\frac{\partial V(D,G)}{\partial D(x)} =$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x)) + p_g(x)\log(1-D(x))dx$$

$$\frac{\partial V(D,G)}{\partial D(x)} = \frac{\partial[p_{data}(x)\log D(x) + p_{gen}(x)\log(1-D(x))]}{\partial D(x)} =$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x)) + p_g(x)\log(1-D(x))dx$$

$$\frac{\partial V(D,G)}{\partial D(x)} = \frac{\partial[p_{data}(x)\log D(x) + p_{gen}(x)\log(1-D(x))]}{\partial D(x)} =$$

$$\frac{p_{data}(x)}{D(x)} - \frac{p_{gen}(x)}{1-D(x)} = 0$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x)) + p_g(x)\log(1-D(x))dx$$

$$\frac{\partial V(D,G)}{\partial D(x)} = \frac{\partial[p_{data}(x)\log D(x) + p_{gen}(x)\log(1-D(x))]}{\partial D(x)} =$$

$$\frac{p_{data}(x)}{D(x)} - \frac{p_{gen}(x)}{1-D(x)} = 0$$

$$p_{data}(x)(1-D(x)) = p_{gen}(x)D(x)$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x)) + p_g(x)\log(1-D(x))dx$$

$$\frac{\partial V(D,G)}{\partial D(x)} = \frac{\partial[p_{data}(x)\log D(x) + p_{gen}(x)\log(1-D(x))]}{\partial D(x)} =$$

$$\frac{p_{data}(x)}{D(x)} - \frac{p_{gen}(x)}{1-D(x)} = 0$$

$$p_{data}(x)(1-D(x)) = p_{gen}(x)D(x)$$

$$D(x) = \frac{p_{data}(x)}{p_{data}(x)+p_{gen}(x)}$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

- Если дискриминатор идеально емкий (может описать любую функцию), он в каждой точке пространства будет предсказывать отношение плотности истинного вероятностного распределения к сумме плотностей истинного и генерируемого распределений.

$$D(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)}$$

https://jonathan-hui.medium.com/proof-gan-optimal-point-658116a236fb

https://stats.stackexchange.com/questions/264717/why-is-the-optimal-discriminator-d-gx-fracp-textdataxp-textd

Обучение: minimax game. Оптимальный D

- Если дискриминатор идеально емкий (может описать любую функцию), он в каждой точке пространства будет предсказывать отношение плотности истинного вероятностного распределения к сумме плотности истинного и генерируемого распределений.

$$D(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)} = \frac{1}{2}$$

это происходит, когда $p_{data} = p_{gen}$

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x)) + p_g(x)\log(1-D(x))dx$$

Чтобы найти максимум, приравняем производную подынтегрального выражения к нулю:

$$\frac{\partial V(D,G)}{\partial D(x)} = 0$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D, G) = \int_x p_{\text{data}}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx$$

Чтобы найти максимум, приравняем производную подынтегрального выражения к нулю:

$$\frac{\partial V(D, G)}{\partial D(x)} = 0$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D, G) = \int_x p_{\text{data}}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx$$

Чтобы найти максимум, приравняем производную подынтегрального выражения к нулю:

$$\frac{\partial V(D, G)}{\partial D(x)} = 0$$

Переименуем переменные:

$$D(x) = y$$

$$p_{\text{data}}(x) = a$$

$$p_{\text{gen}}(x) = b$$

$$y = a \log(y) + b \log(1 - y)$$

$$y' = \frac{a}{y} - \frac{b}{1 - y}$$

$$\frac{a}{y^*} = \frac{b}{1 - y^*}$$

$$\frac{1 - y^*}{y^*} = \frac{b}{a}$$

$$\frac{1}{y^*} = \frac{a + b}{a}$$

$$y^* = \frac{a}{a + b}$$

Find optimal y^* by setting $y' = 0$.

https://jonathan-hui.medium.com/proof-gan-optimal-point-658116a236fb

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\max_D V(D,G) = \int_x p_{\text{data}}(x)\log(D(x))dx + \int_z p_z(z)\log(1 - D(G(z)))dz$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\frac{\partial V(D,G)}{\partial D(x)} = 0$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\frac{\partial V(D,G)}{\partial D(x)} = \frac{\partial [p_{data}(x)\log D(x) + p_{gen}(x)\log(1 - D(x))]}{\partial D(x)} =$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\frac{\partial V(D,G)}{\partial D(x)} = \frac{\partial[p_{data}(x)\log D(x) + p_{gen}(x)\log(1 - D(x))]}{\partial D(x)} =$$

$$\frac{p_{data}(x)}{D(x)} - \frac{p_{gen}(x)}{1 - D(x)} = 0$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\frac{\partial V(D,G)}{\partial D(x)} = \frac{\partial[p_{data}(x)\log D(x) + p_{gen}(x)\log(1 - D(x))]}{\partial D(x)} =$$

$$\frac{p_{data}(x)}{D(x)} - \frac{p_{gen}(x)}{1 - D(x)} = 0$$

$$p_{data}(x)(1 - D(x)) = p_{gen}(x)D(x)$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

$$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

$$\frac{\partial V(D,G)}{\partial D(x)} = \frac{\partial [p_{data}(x)\log D(x) + p_{gen}(x)\log(1-D(x))]}{\partial D(x)} =$$

$$\frac{p_{data}(x)}{D(x)} - \frac{p_{gen}(x)}{1-D(x)} = 0$$

$$p_{data}(x)(1-D(x)) = p_{gen}(x)D(x)$$

$$D(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)}$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game. Оптимальный D

- Если дискриминатор идеально емкий (может описать любую функцию), он в каждой точке пространства будет предсказывать отношение плотности истинного вероятностного распределения к сумме плотностей истинного и генерируемого распределений.

$$D(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)}$$

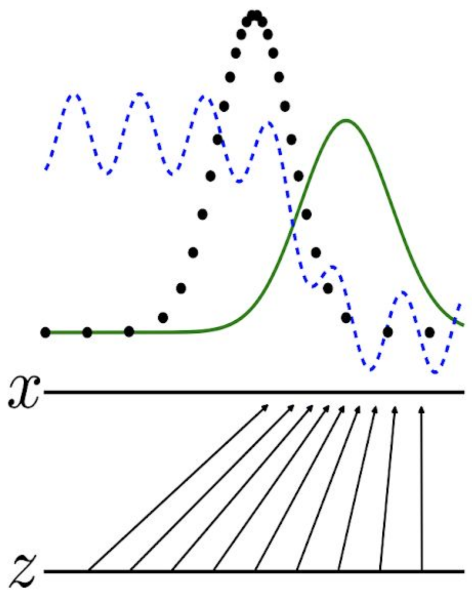
Обучение: minimax game. Оптимальный D

- Если дискриминатор идеально емкий (может описать любую функцию), он в каждой точке пространства будет предсказывать отношение плотности истинного вероятностного распределения к сумме плотностей истинного и генерируемого распределений.

$$D(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)} = \frac{1}{2}$$

это происходит, когда $p_{data} = p_{gen}$

Обучение: minimax game.



(a)

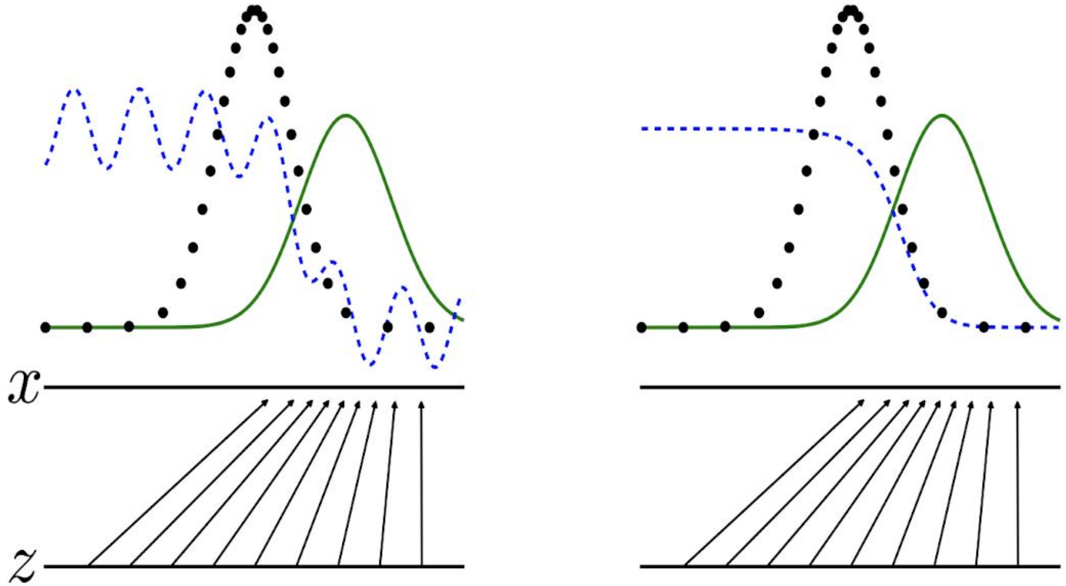
распределение данных (целевое распределение)

распределение ответов дискриминатора (дискриминативное распределение)

распределение ответов генератора (генерируемое распределение)

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game.



(a) (b)

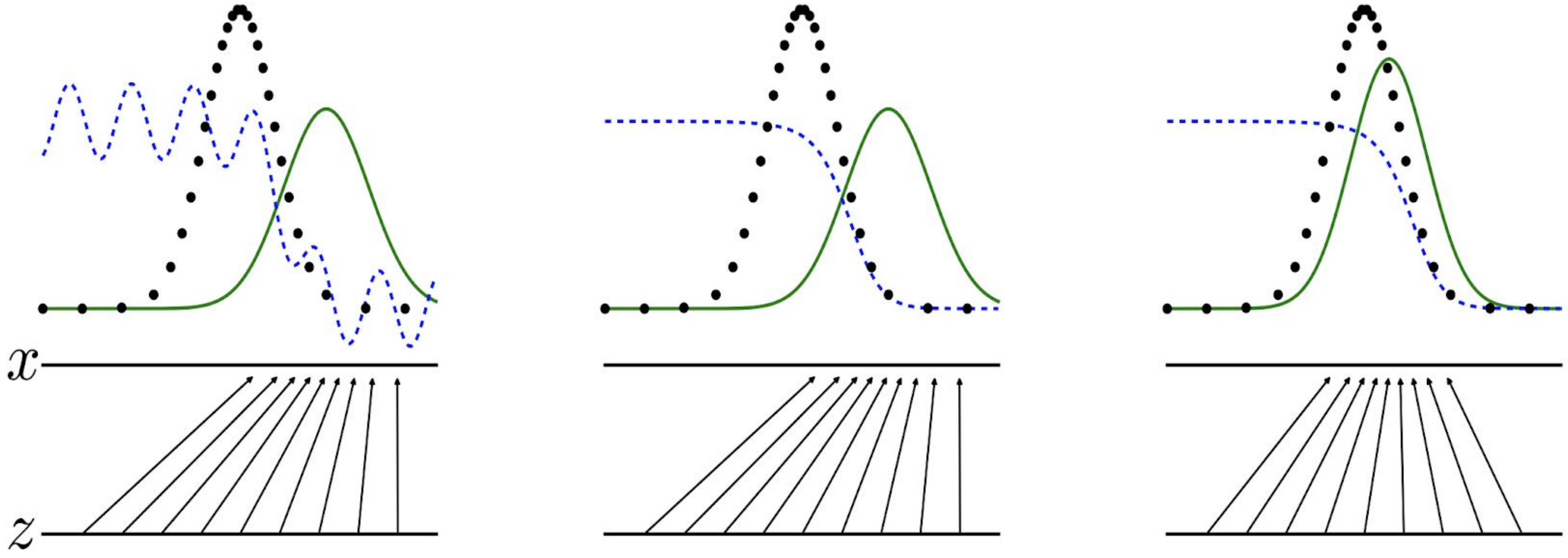
распределение данных (целевое распределение)

распределение ответов дискриминатора (дискриминативное распределение)

распределение ответов генератора (генерируемое распределение)

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game.



(a) (b) (c)

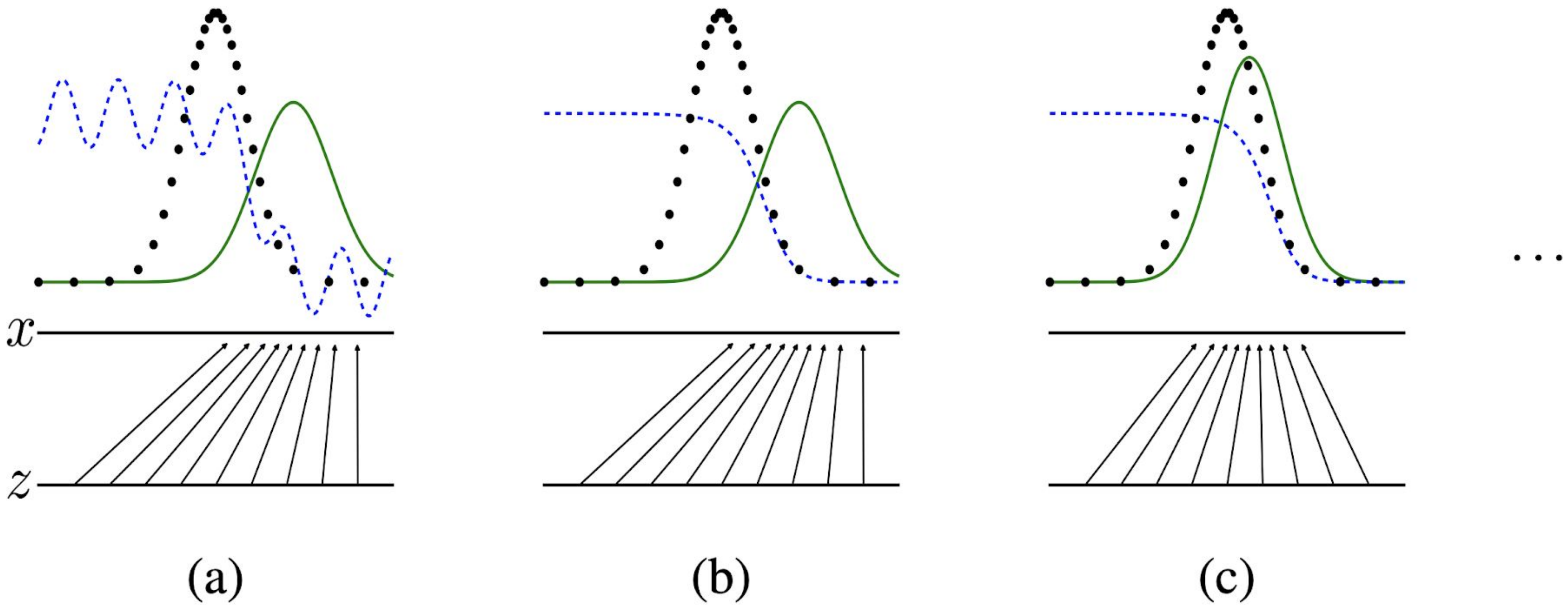
распределение данных (целевое распределение)

распределение ответов дискриминатора (дискриминативное распределение)

распределение ответов генератора (генерируемое распределение)

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game.



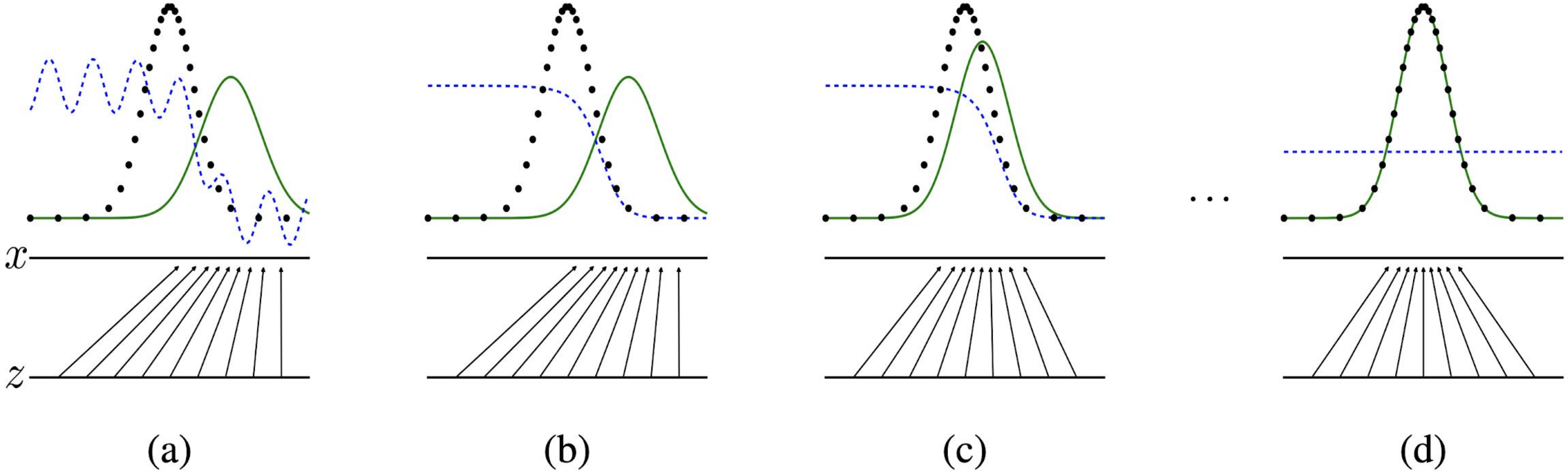
распределение данных (целевое распределение)

распределение ответов дискриминатора (дискриминативное распределение)

распределение ответов генератора (генерируемое распределение)

https://arxiv.org/pdf/1406.2661.pdf

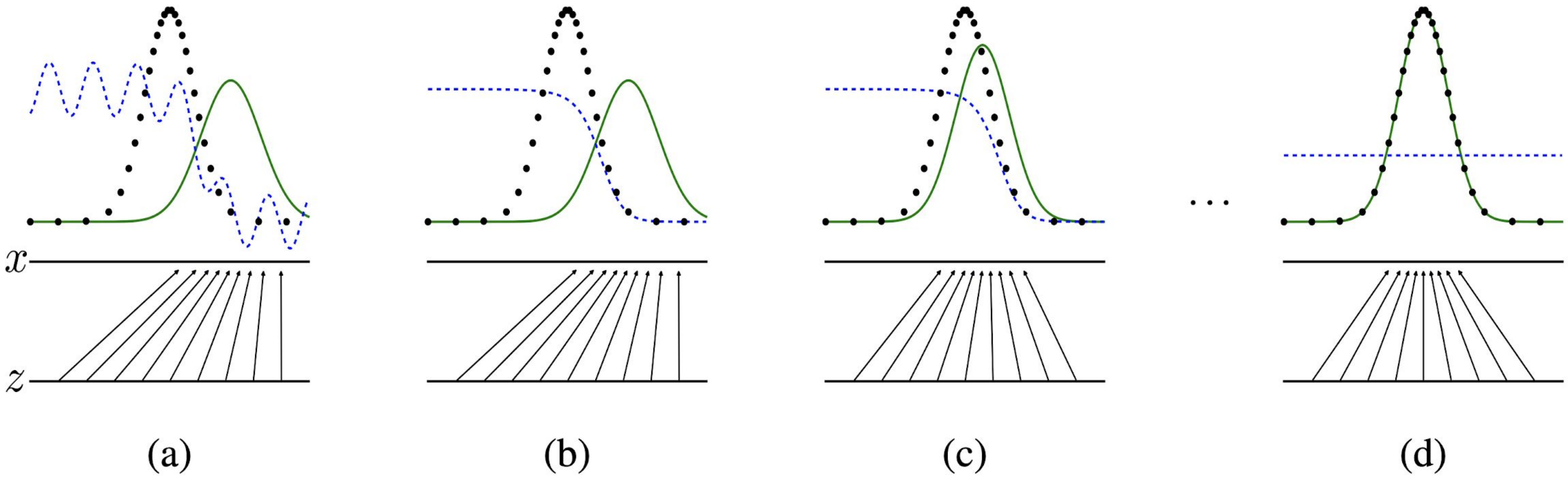
Обучение: minimax game.



$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}$$

https://arxiv.org/pdf/1406.2661.pdf

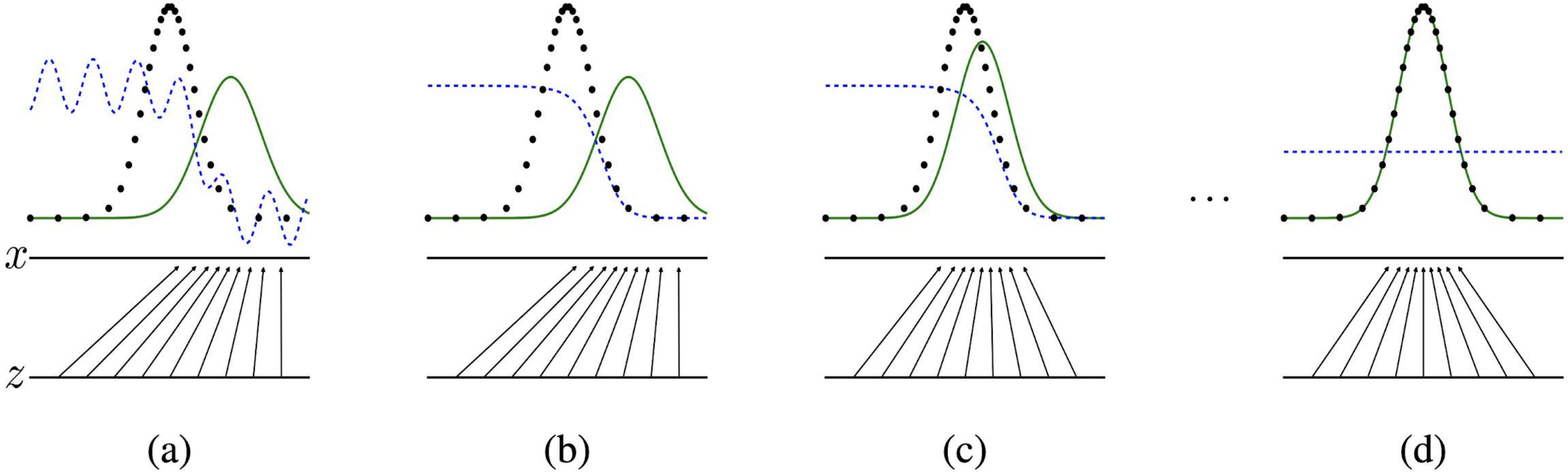
Обучение: minimax game.



$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)} = \frac{1}{2}$$

https://arxiv.org/pdf/1406.2661.pdf

Обучение: minimax game.

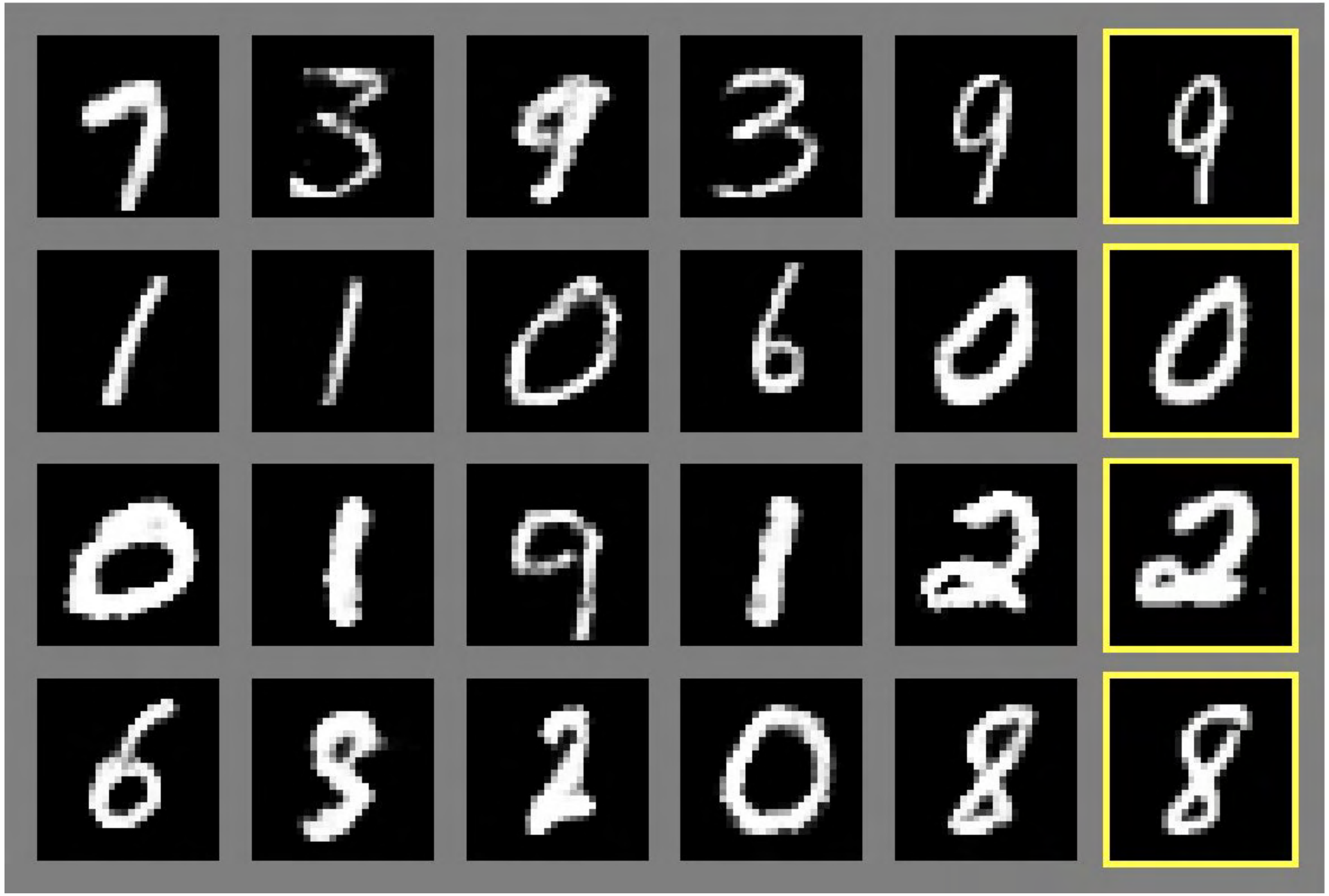


$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)} = \frac{1}{2}$$

Оценка этого отношения с помощью обучения с учителем - главный механизм аппроксимации, который помогает генеративным сетям работать

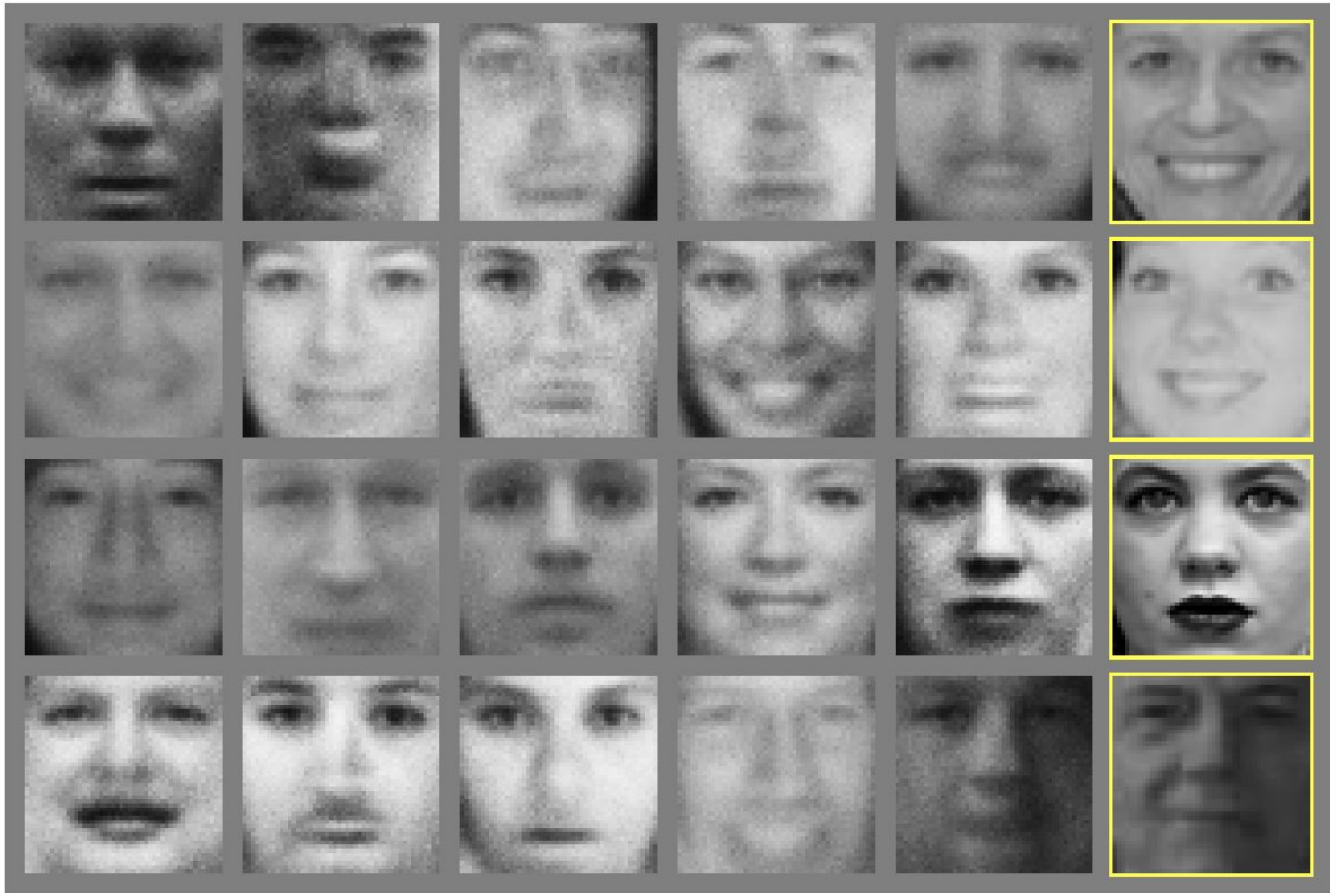
https://arxiv.org/pdf/1406.2661.pdf

Оригинальная статья. Примеры работы



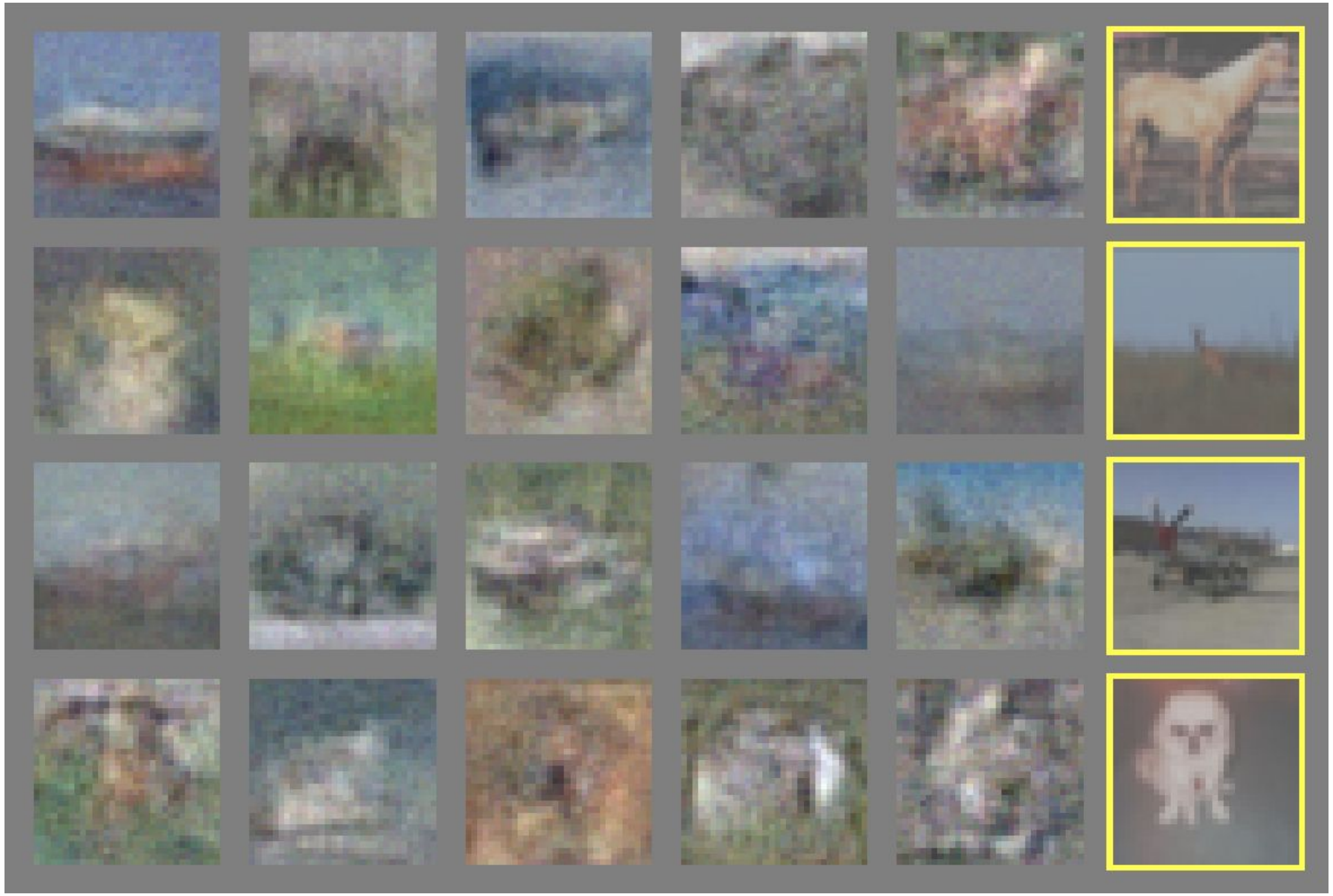
https://arxiv.org/pdf/1406.2661.pdf

Оригинальная статья. Примеры работы



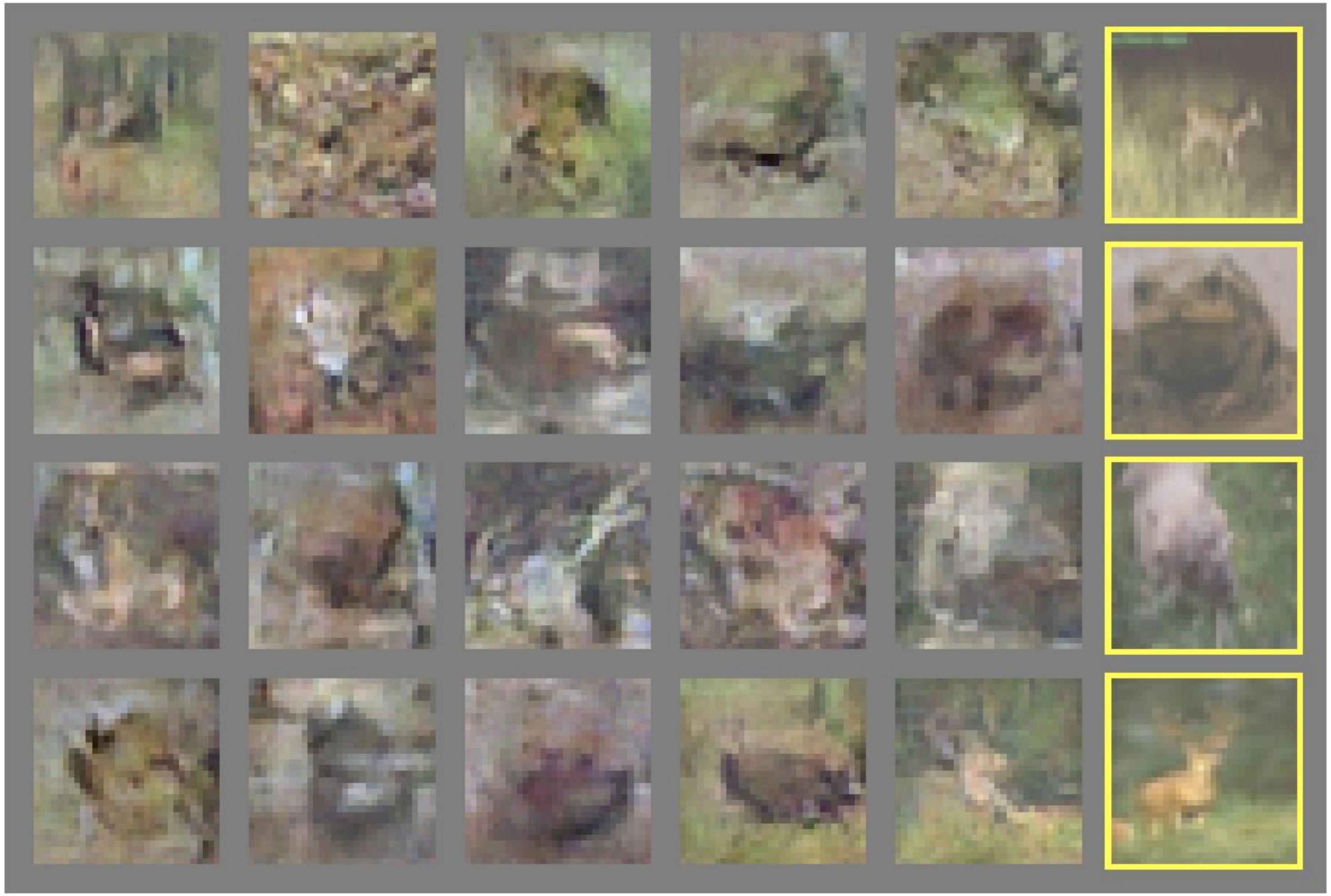
https://arxiv.org/pdf/1406.2661.pdf

Оригинальная статья. Примеры работы



https://arxiv.org/pdf/1406.2661.pdf

Оригинальная статья. Примеры работы



https://arxiv.org/pdf/1406.2661.pdf

DCGAN. Усовершенствование оригинальной статьи

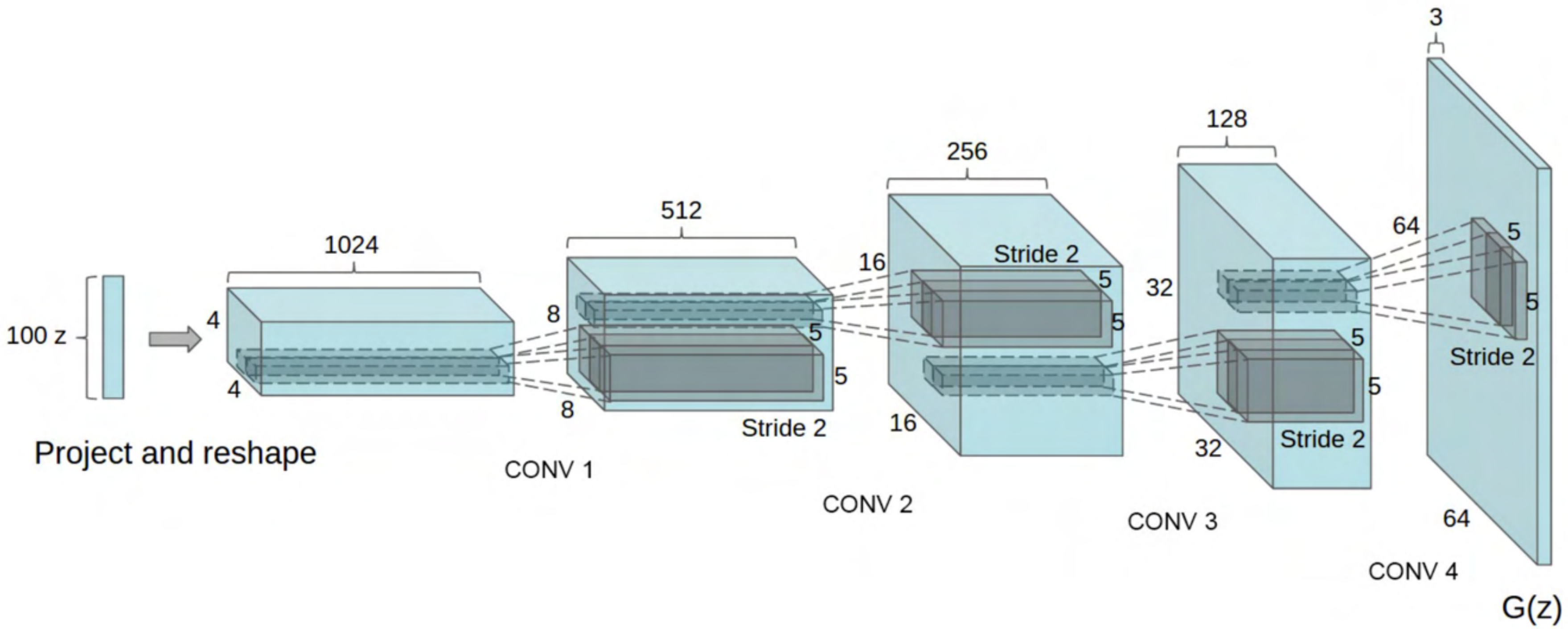


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

https://arxiv.org/abs/1511.06434

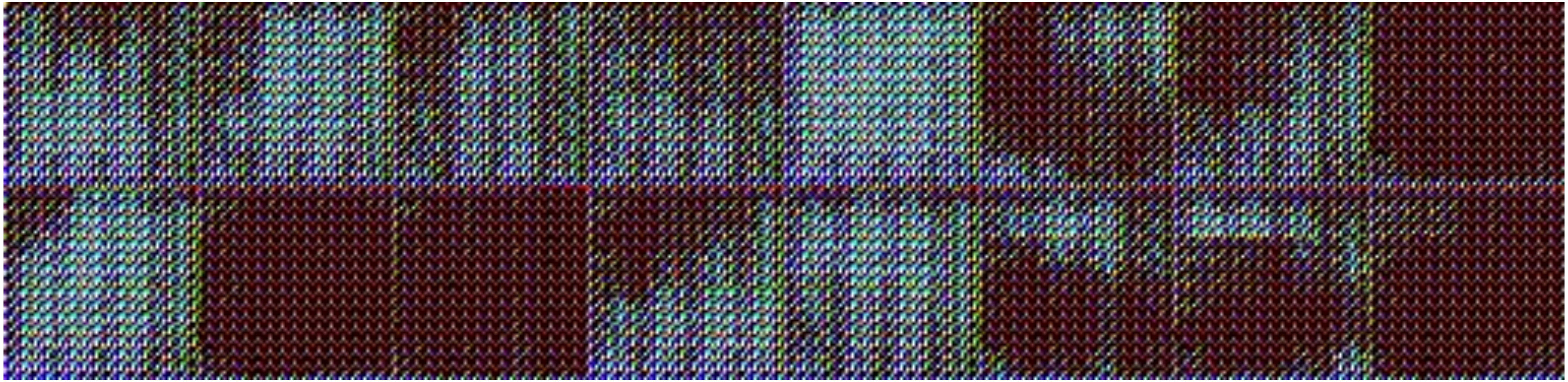
DCGAN. Результаты работа на датасете LSUN

https://arxiv.org/abs/1511.06434

DCGAN. Результаты работа на датасете LSUN

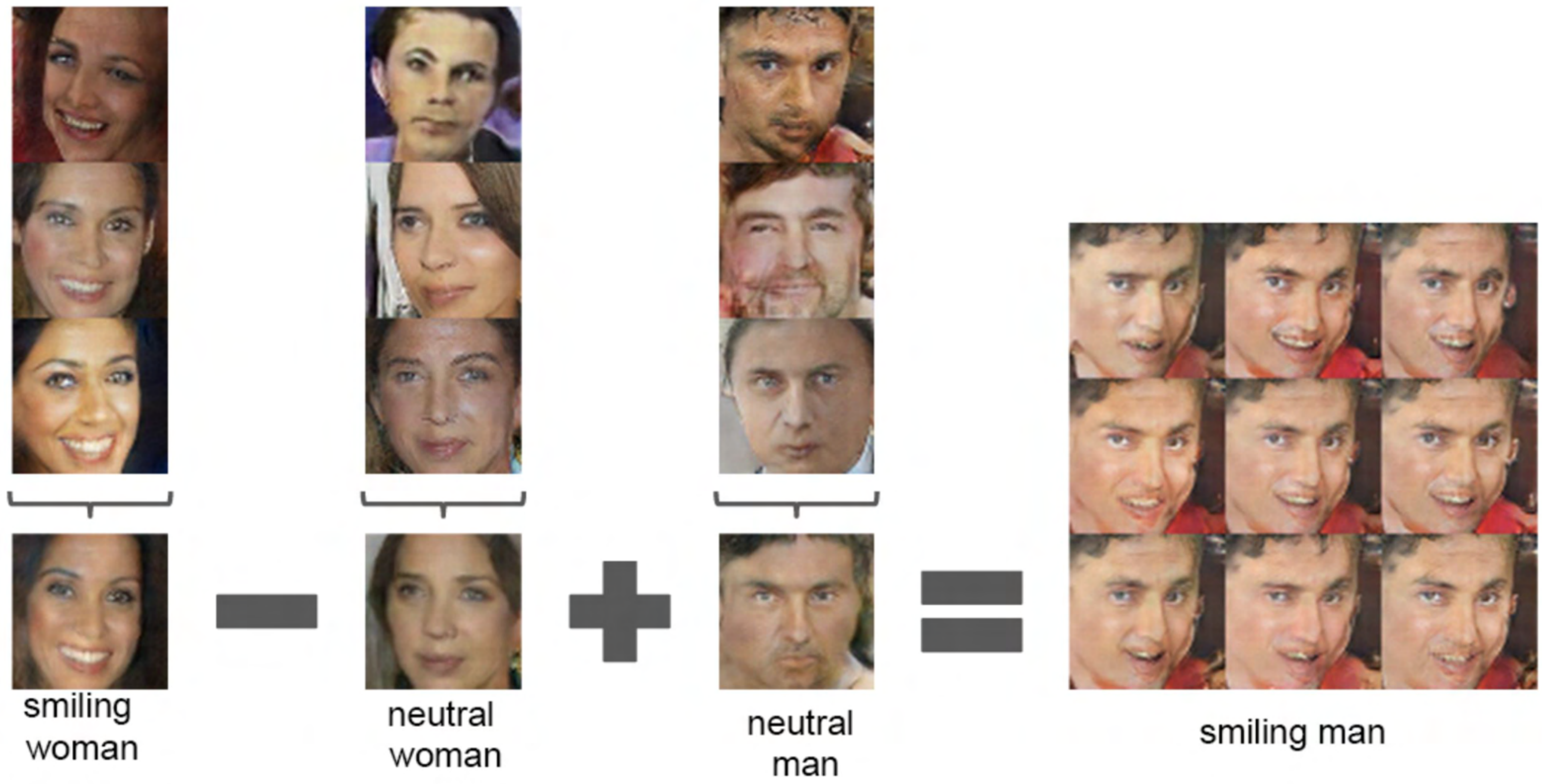
https://arxiv.org/abs/1511.06434

DCGAN. Результаты работа без BatchNorm



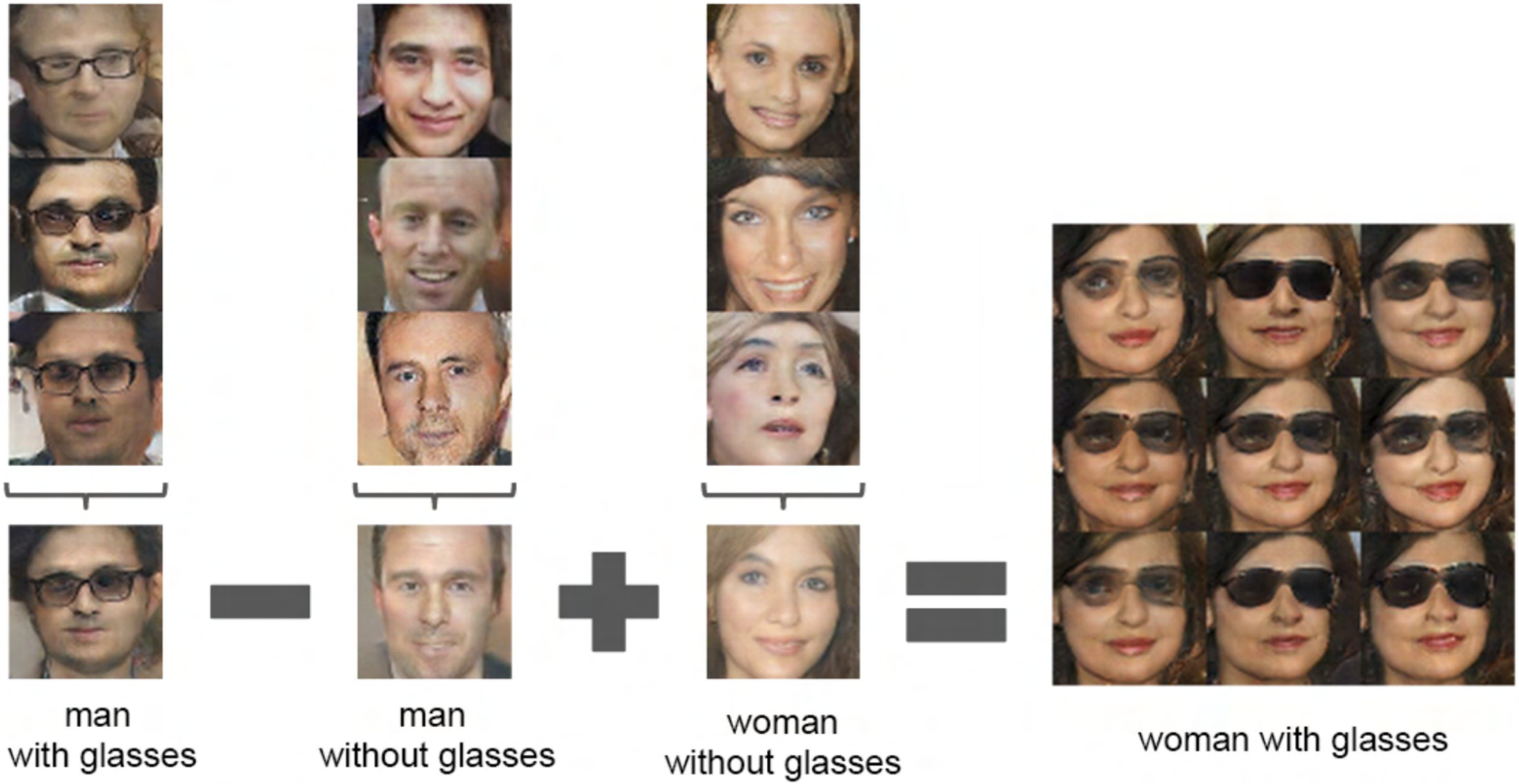
https://arxiv.org/abs/1511.06434

DCGAN. Векторная арифметика



https://arxiv.org/abs/1511.06434

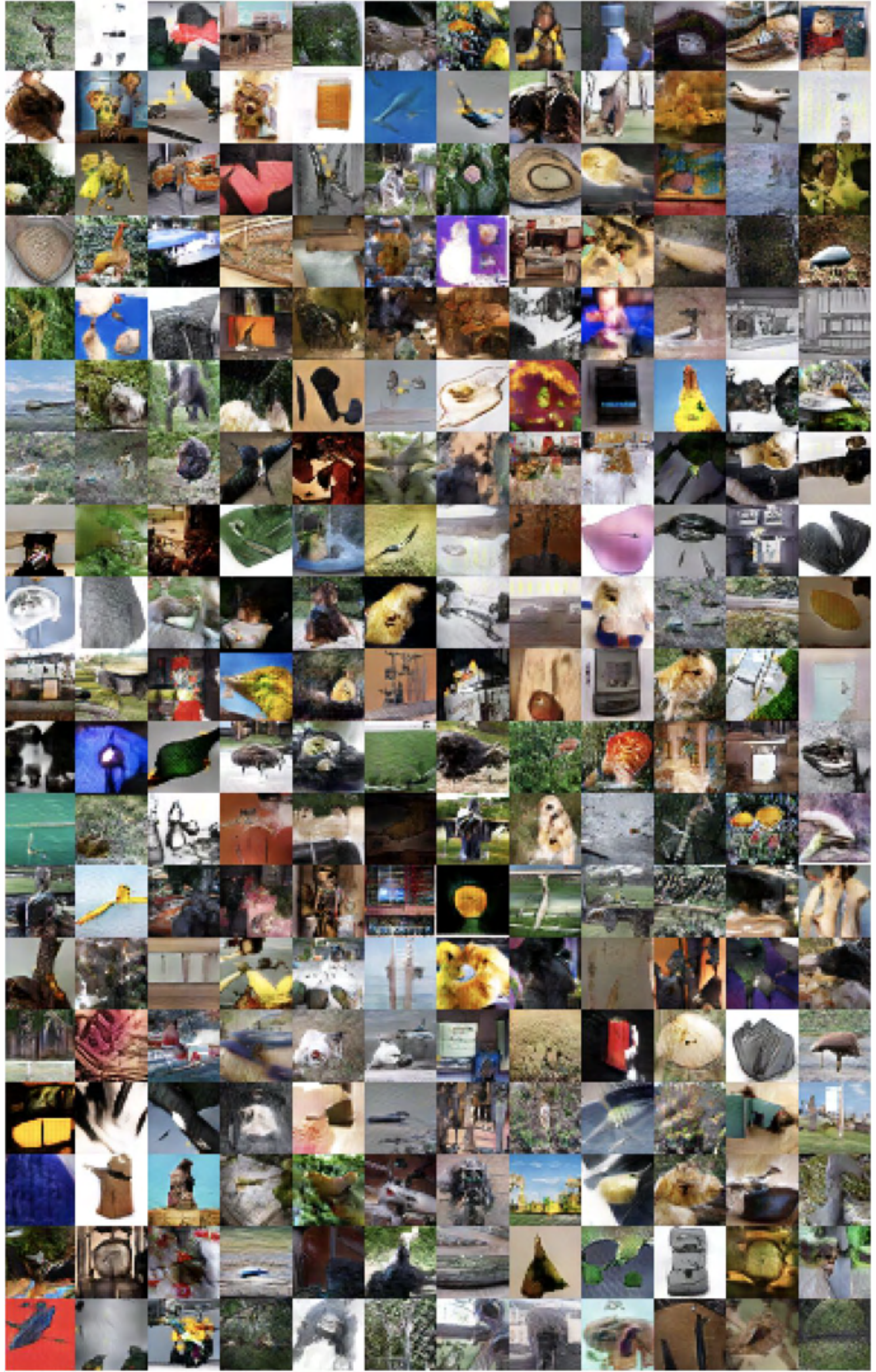
DCGAN. Векторная арифметика

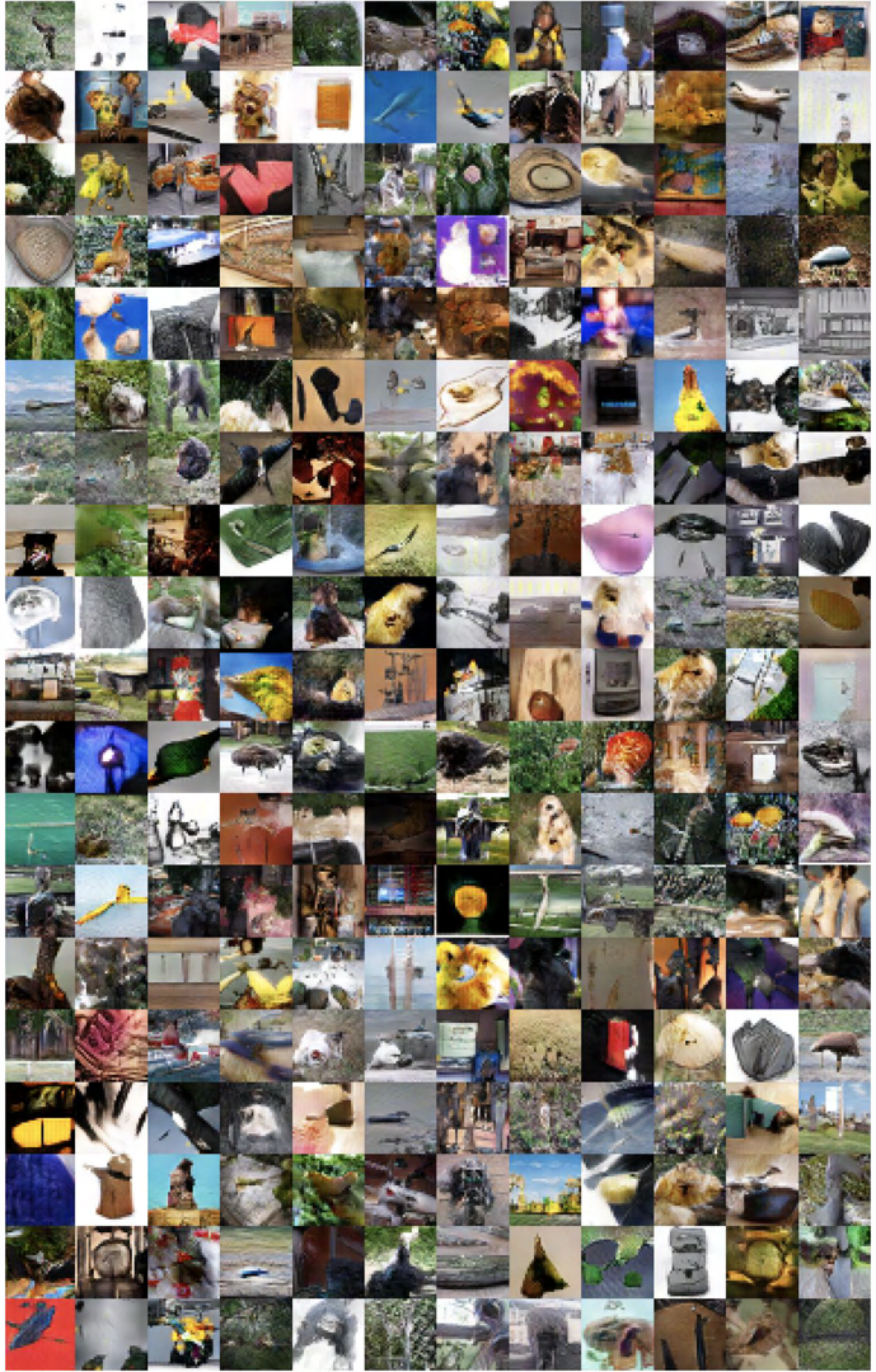


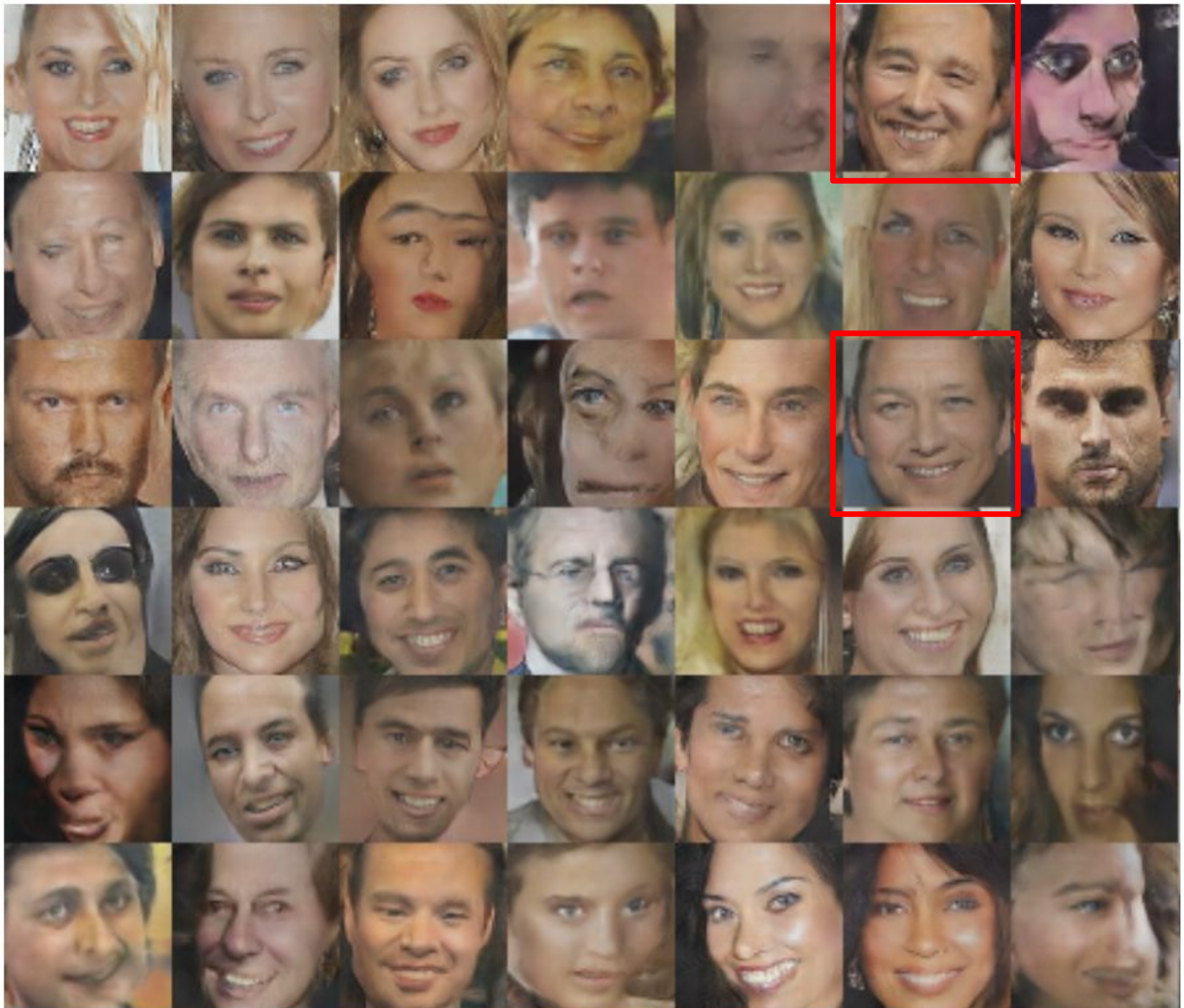
https://arxiv.org/abs/1511.06434

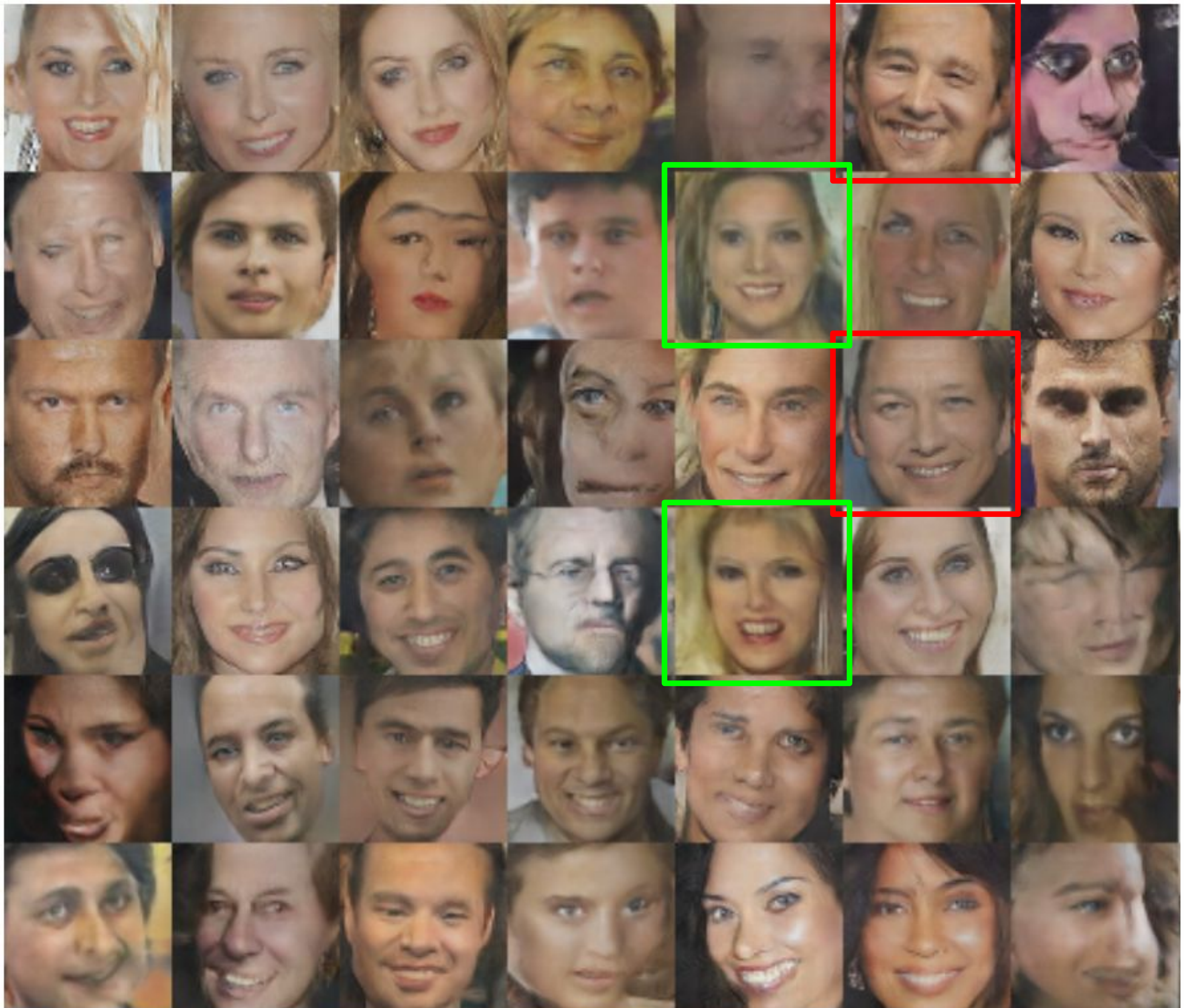
DCGAN. Пиксельная арифметика

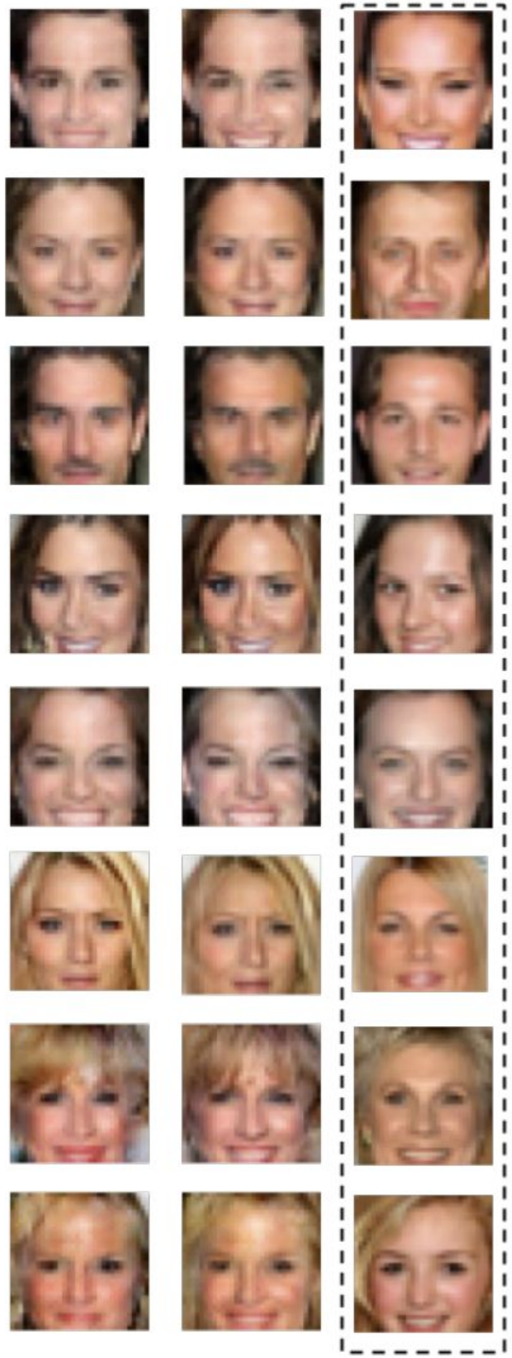
https://arxiv.org/abs/1511.06434









DCGAN

Published as a conference paper at ICLR 2018

DO GANS LEARN THE DISTRIBUTION? SOME THEORY AND EMPIRICS

Sanjeev Arora
Department of Computer Science
Princeton University
Princeton, NJ 08544, USA
arora@cs.princeton.edu

Andrej Risteski
Applied Mathematics Department and IDSS
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
risteski@mit.edu

Yi Zhang
Department of Computer Science
Princeton University
Princeton, NJ 08544, USA
y.zhang@cs.princeton.edu

https://arxiv.org/abs/1706.08224

Проблема идеального генератора

D: $$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

G: $$\max_G \mathbb{E}_{z\sim p_z(z)} \log(D(G(z))$$

https://arxiv.org/abs/1511.06434

Проблема идеального генератора

D: $$\max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

G: $$\max_G \mathbb{E}_{z\sim p_z(z)} \log(D(G(z)))$$

Если дискриминатор зафиксирован (не учится), то что нужно делать генератору, чтобы всегда обманывать дискриминатор?

$$G(z) = \operatorname*{argmax}_x D(x)$$

https://arxiv.org/abs/1511.06434

Проблема идеального генератора

- Если дискриминатор остановлен и не учится, оптимальный генератор будет для любого входа возвращать такую картинку, для которой дискриминатор возвращает максимальную вероятность.

$$G(z) = \operatorname*{argmax}_x D(x)$$

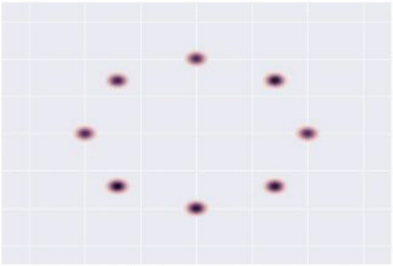
Проблема идеального генератора

- Если дискриминатор остановлен и не учится, оптимальный генератор будет для любого входа возвращать такую картинку, для которой дискриминатор возвращает максимальную вероятность.

$$G(z) = \underset{x}{\operatorname{argmax}}\, D(x)$$

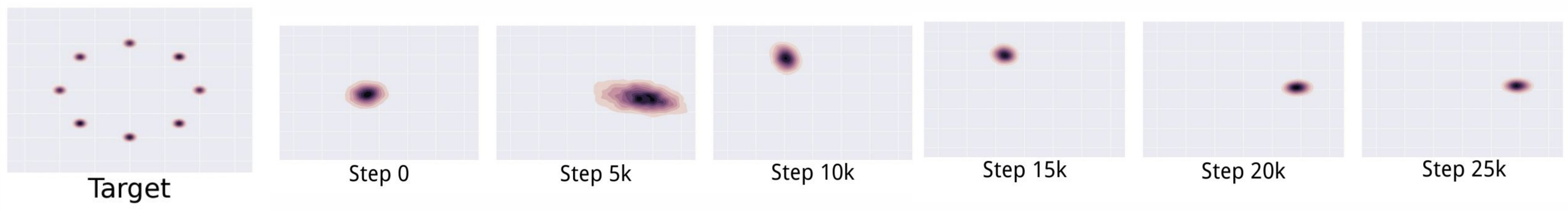
- Проблема в том, что мы никак не заставляет генератор вести себя по-другому, поэтому генератор может для какой-то части латентного пространства возвращать очень похожие выходы. Такая проблема называется **Mode Collapse**

Mode collapse



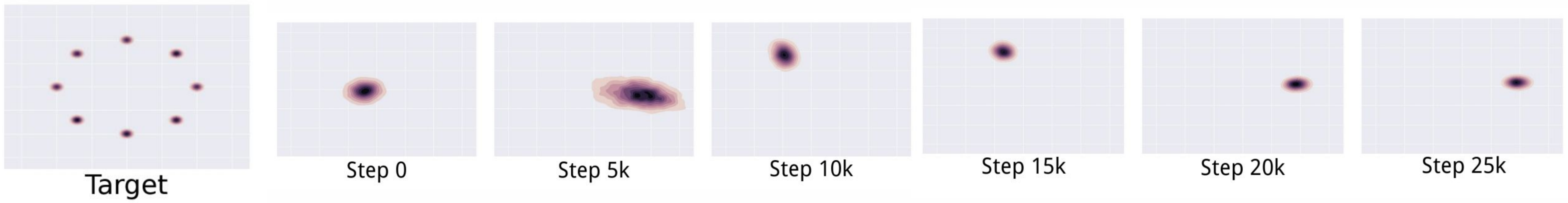
https://arxiv.org/abs/1611.02163

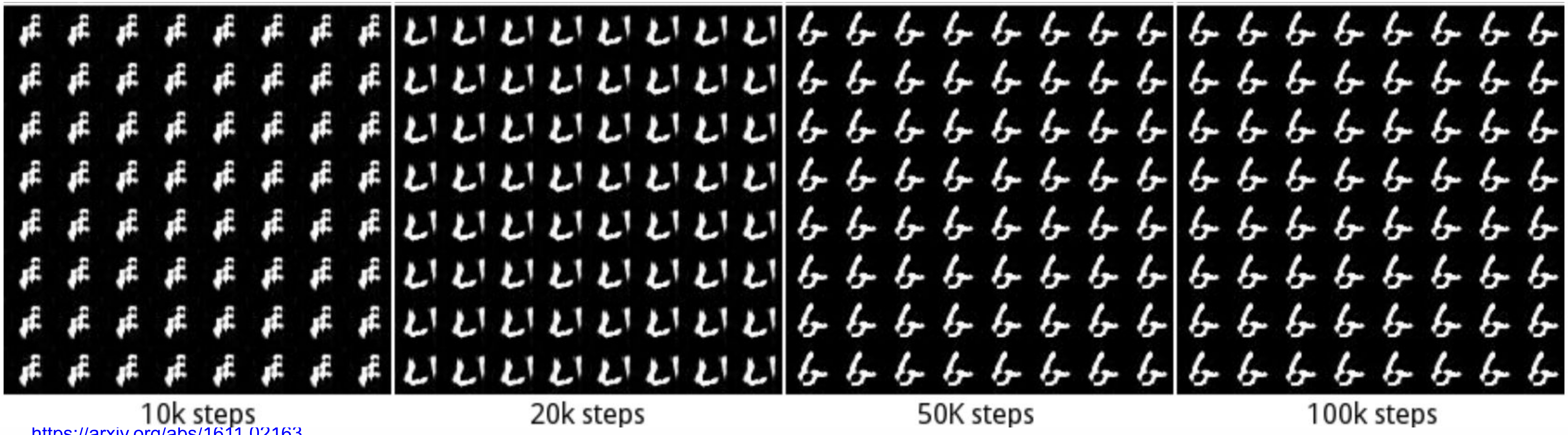
Mode collapse



https://arxiv.org/abs/1611.02163

Mode collapse





https://arxiv.org/abs/1611.02163

https://arxiv.org/abs/1912.04958

https://arxiv.org/
abs/1912.04958

Что мы уже прошли

- Оригинальная статья:
  - учим дискриминатор как бинарный классификатор
  - используем градиенты ошибки классификации для обучения дискриминатора и генератора
  - столкнулись с затуханием градиентов генератора
    - поменяли функционал ошибки генератора
  - узнали, что идеальный дискриминатор пытается предсказать отношение плотностей распределений
  - столкнулись с проблемой mode collapse. В 2020 она до сих пор не решена
    - причиной считается поведение генератора, который стремится для всего латентного пространства вернуть одинаковый выход

Что будет дальше

- Вспомним что такое KL и JS дивергенции
- Столкнемся с проблемами сходимости у сети из оригинальной статьи
- Предложим решение проблем

KL-дивергенция

- Несимметричный функционал, показывающий меру близости двух вероятностных распределений

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

KL-дивергенция

- Несимметричный функционал, показывающий меру близости двух вероятностных распределений

$$D_{KL}(P \parallel Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$$ для дискретных распределений

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

KL-дивергенция

❖ Несимметричный функционал, показывающий меру близости двух вероятностных распределений

$$D_{KL}(P \parallel Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$$

для дискретных распределений

$$D_{\mathrm{KL}}(P \parallel Q) = \int_X p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x$$

для абсолютно непрерывных распределений P и Q, где

p(x) и q(x) - плотности

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

KL-дивергенция

- Несимметричный функционал, показывающий меру близости двух вероятностных распределений

$$D_{KL}(P \parallel Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$$

для дискретных распределений

$$D_{\mathrm{KL}}(P \parallel Q) = \int_X p(x) \log \frac{p(x)}{q(x)}\, \mathrm{d}x$$

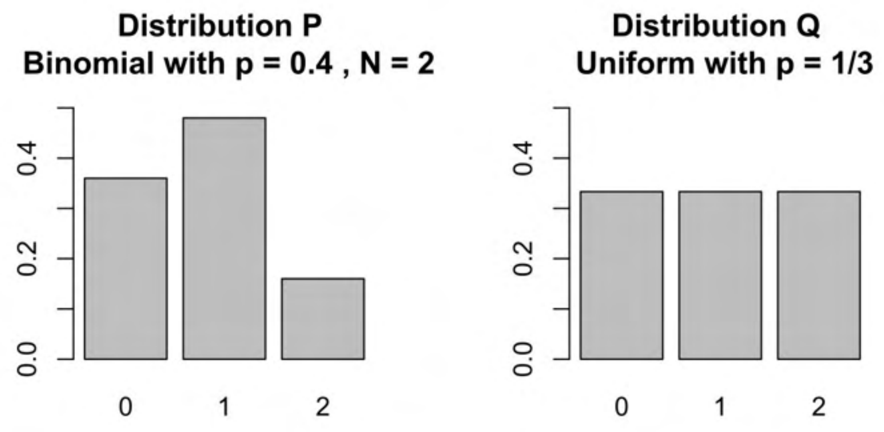
для абсолютно непрерывных распределений P и Q, где

p(x) и q(x) - плотности

$$D_{\mathrm{KL}}(P \parallel Q) \neq D_{\mathrm{KL}}(Q \parallel P)$$

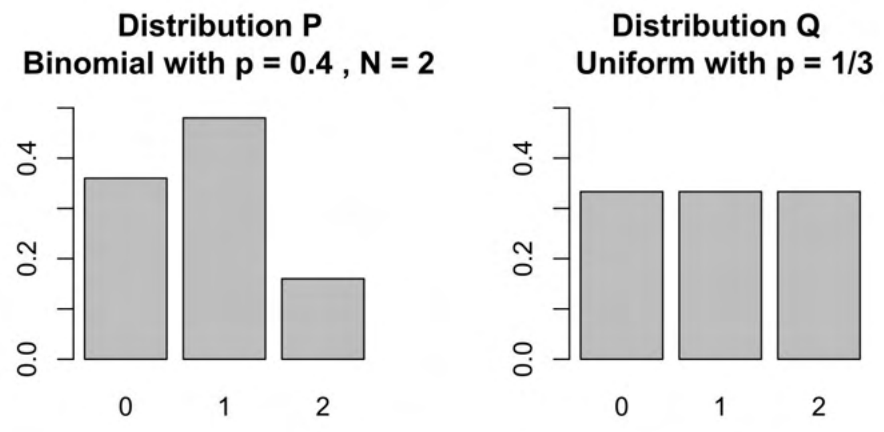
https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

KL-дивергенция. Дискретный случай



https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

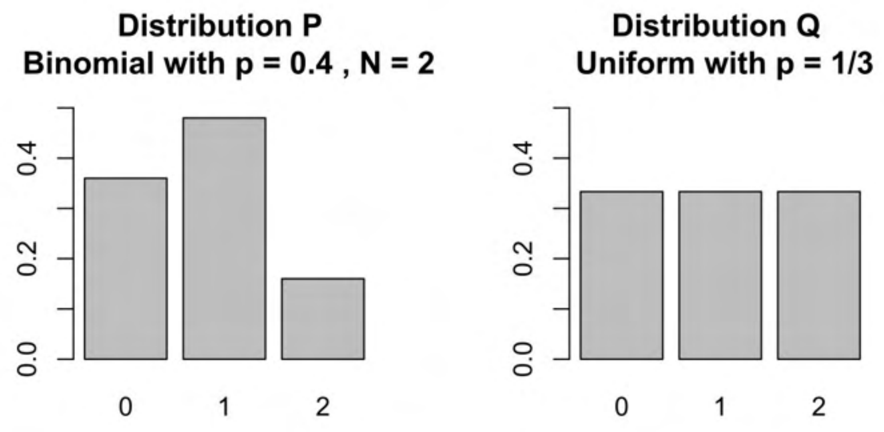
KL-дивергенция. Дискретный случай



x	1	2	3
P(x)	0.36	0.48	0.16
Q(x)	0.333	0.333	0.333

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Дискретный случай



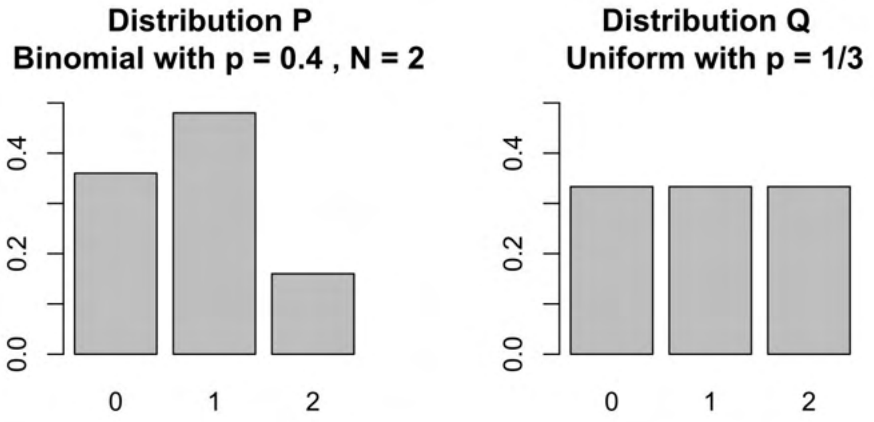
x	1	2	3
P(x)	0.36	0.48	0.16
Q(x)	0.333	0.333	0.333

$$D_{\mathrm{KL}}(P \parallel Q) =$$

$$D_{\mathrm{KL}}(Q \parallel P) =$$

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Дискретный случай



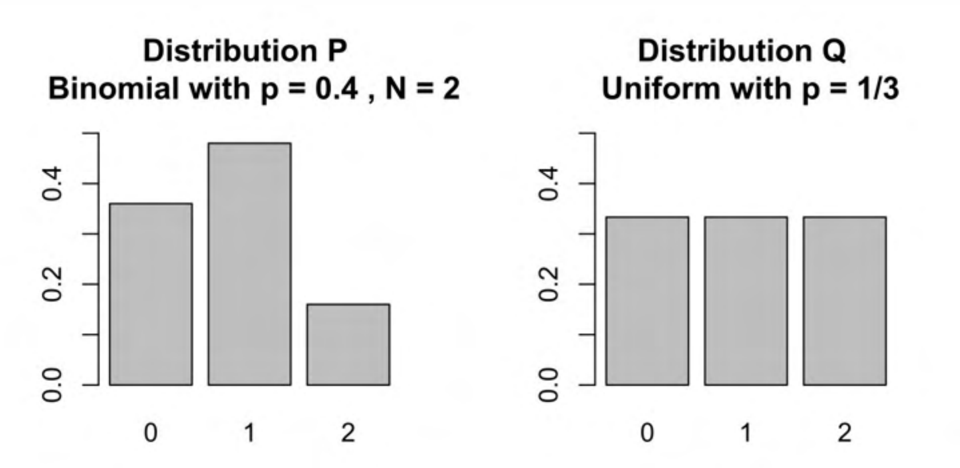
x	1	2	3
P(x)	0.36	0.48	0.16
Q(x)	0.333	0.333	0.333

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \ln\left(\frac{P(x)}{Q(x)}\right)$$

$$D_{\mathrm{KL}}(Q \parallel P) =$$

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Дискретный случай



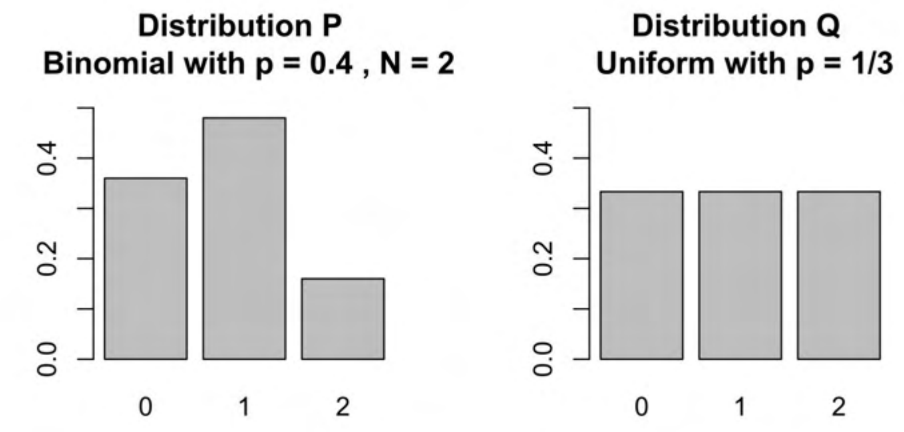
x	1	2	3
P(x)	0.36	0.48	0.16
Q(x)	0.333	0.333	0.333

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \ln\left(\frac{P(x)}{Q(x)}\right) = 0.36 \ln\left(\frac{0.36}{0.333}\right) + 0.48 \ln\left(\frac{0.48}{0.333}\right) + 0.16 \ln\left(\frac{0.16}{0.333}\right)$$

$$D_{\mathrm{KL}}(Q \parallel P) =$$

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Дискретный случай



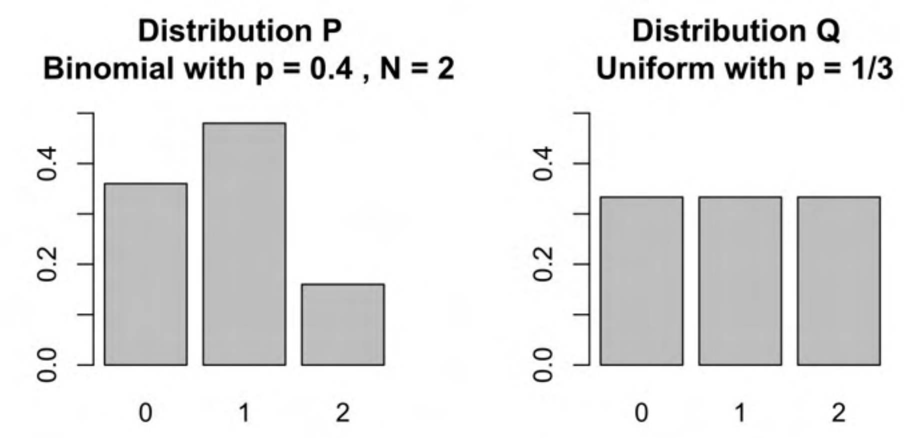
x	1	2	3
P(x)	0.36	0.48	0.16
Q(x)	0.333	0.333	0.333

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \ln\left(\frac{P(x)}{Q(x)}\right) = 0.36 \ln\left(\frac{0.36}{0.333}\right) + 0.48 \ln\left(\frac{0.48}{0.333}\right) + 0.16 \ln\left(\frac{0.16}{0.333}\right) = 0.085299$$

$$D_{\mathrm{KL}}(Q \parallel P) =$$

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Дискретный случай



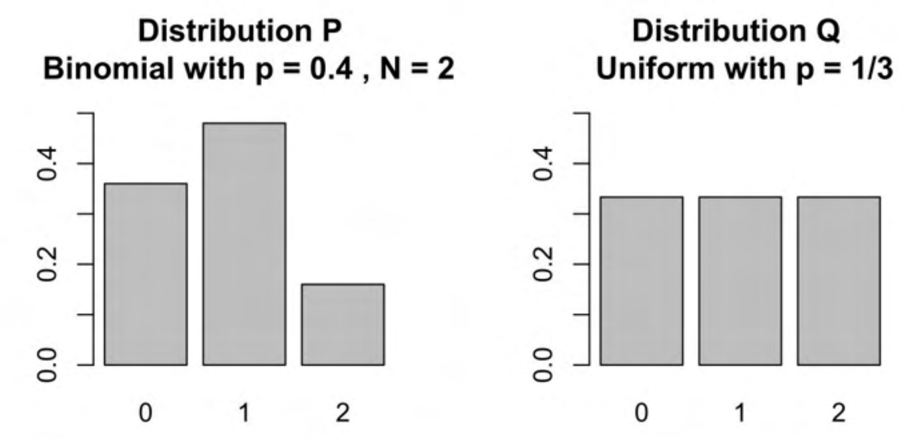
x	1	2	3
P(x)	0.36	0.48	0.16
Q(x)	0.333	0.333	0.333

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \ln\left(\frac{P(x)}{Q(x)}\right) = 0.36 \ln\left(\frac{0.36}{0.333}\right) + 0.48 \ln\left(\frac{0.48}{0.333}\right) + 0.16 \ln\left(\frac{0.16}{0.333}\right) = 0.085299$$

$$D_{\mathrm{KL}}(Q \parallel P) = \sum_{x \in \mathcal{X}} Q(x) \ln\left(\frac{Q(x)}{P(x)}\right)$$

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Дискретный случай



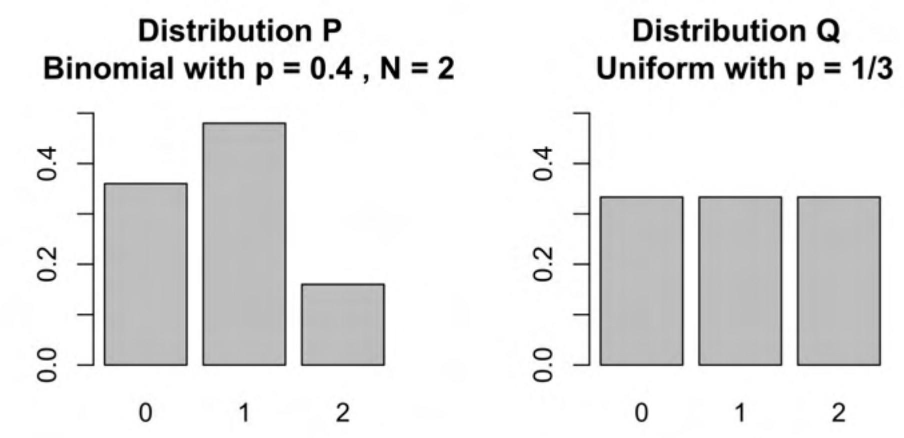
x	1	2	3
P(x)	0.36	0.48	0.16
Q(x)	0.333	0.333	0.333

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \ln\left(\frac{P(x)}{Q(x)}\right) = 0.36 \ln\left(\frac{0.36}{0.333}\right) + 0.48 \ln\left(\frac{0.48}{0.333}\right) + 0.16 \ln\left(\frac{0.16}{0.333}\right) = 0.085299$$

$$D_{\mathrm{KL}}(Q \parallel P) = \sum_{x \in \mathcal{X}} Q(x) \ln\left(\frac{Q(x)}{P(x)}\right) = 0.333 \ln\left(\frac{0.333}{0.36}\right) + 0.333 \ln\left(\frac{0.333}{0.48}\right) + 0.333 \ln\left(\frac{0.333}{0.16}\right)$$

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Дискретный случай



x	1	2	3
P(x)	0.36	0.48	0.16
Q(x)	0.333	0.333	0.333

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \ln\left(\frac{P(x)}{Q(x)}\right) = 0.36 \ln\left(\frac{0.36}{0.333}\right) + 0.48 \ln\left(\frac{0.48}{0.333}\right) + 0.16 \ln\left(\frac{0.16}{0.333}\right) = 0.085299$$

$$D_{\mathrm{KL}}(Q \parallel P) = \sum_{x \in \mathcal{X}} Q(x) \ln\left(\frac{Q(x)}{P(x)}\right) = 0.333 \ln\left(\frac{0.333}{0.36}\right) + 0.333 \ln\left(\frac{0.333}{0.48}\right) + 0.333 \ln\left(\frac{0.333}{0.16}\right) = 0.097455$$

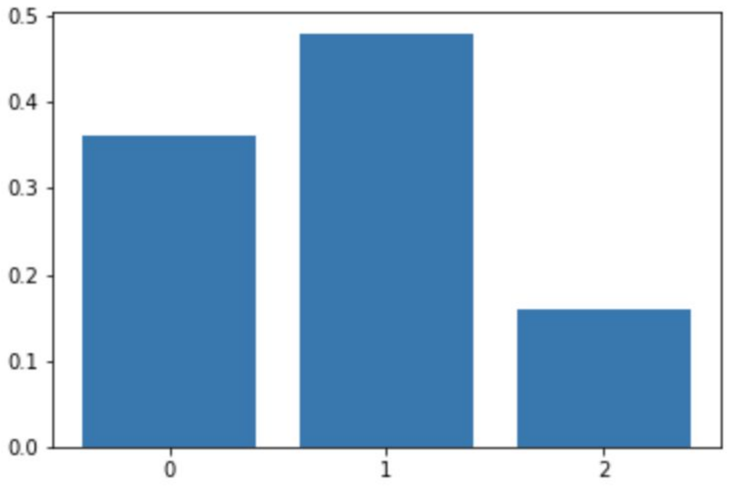
https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Код

Биномиальное распределение

```
rv = binom(2, 0.4)
x = np.arange(3)
binom_pmf = rv.pmf(x)
print ("Probability mass function values: {}".format(binom_pmf))
plt.bar(x, binom_pmf, width=0.8, tick_label=x)
plt.show()
```

```
Probability mass function values: [0.36 0.48 0.16]
```



https://arxiv.org/pdf/1406.2661.pdf

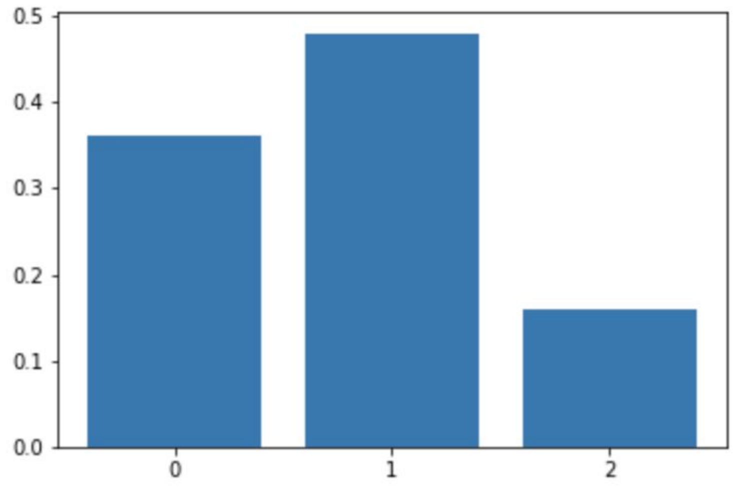
https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Код

Биномиальное распределение

```
rv = binom(2, 0.4)
x = np.arange(3)
binom_pmf = rv.pmf(x)
print ("Probability mass function values: {}".format(binom_pmf))
plt.bar(x, binom_pmf, width=0.8, tick_label=x)
plt.show()
```

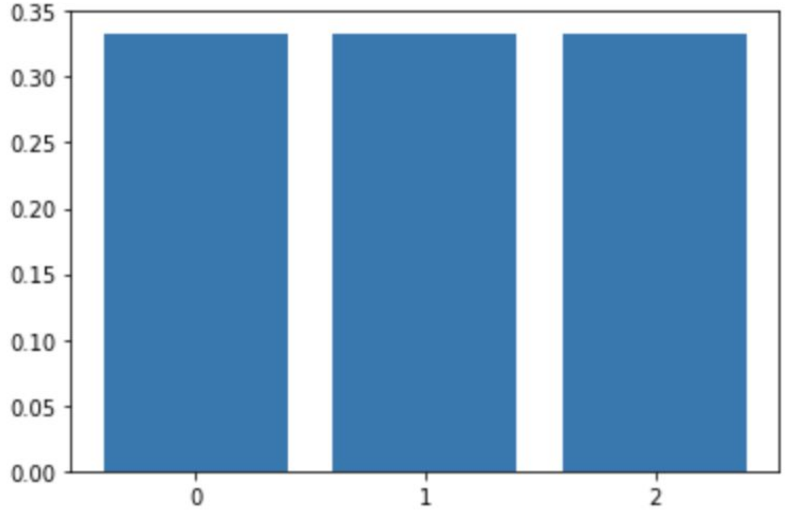
```
Probability mass function values: [0.36 0.48 0.16]
```



Равномерное распределение

```
uniform_pmf = 1 / len(x) * np.ones_like(x)
print ("Probability mass function values: {}".format(uniform_pmf))
plt.bar(x, uniform_pmf, width=0.8, tick_label=x)
plt.show()
```

```
Probability mass function values: [0.33333333 0.33333333 0.33333333]
```



https://arxiv.org/pdf/1406.2661.pdf

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Код

KL-дивергенция своими руками

```
# https://towardsdatascience.com/kl-divergence-python-example-b87069e4b810
def kl_divergence(p, q):
    return np.sum(np.where(p != 0, p * np.log(p / q), 0))
```

$D_{\mathrm{KL}}(P \parallel Q)$

```
print ("%.7f" % kl_divergence(binom_pmf, uniform_pmf))
```

```
0.0852996
```

$D_{\mathrm{KL}}(Q \parallel P)$

```
print ("%.6f" % kl_divergence(uniform_pmf, binom_pmf))
```

```
0.097455
```

https://arxiv.org/pdf/1406.2661.pdf

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

KL-дивергенция. Код

KL-дивергенция из scipy

```
# https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.kl_div.html
from scipy.special import kl_div
```

```
print ("%.7f" % np.sum(kl_div(binom_pmf, uniform_pmf)))
```

```
0.0852996
```

```
print ("%.6f" % np.sum(kl_div(uniform_pmf, binom_pmf)))
```

```
0.097455
```

https://arxiv.org/pdf/1406.2661.pdf

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

JS-дивергенция

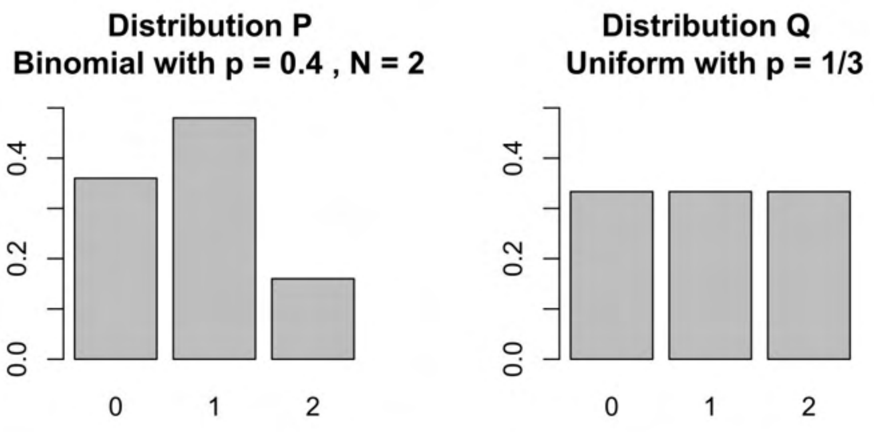
$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M)$$

JS-дивергенция

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M)$$

$$M = \frac{1}{2}(P + Q)$$

JS-дивергенция. Дискретный случай



x	1	2	3
P(x)	0.36	0.48	0.16
Q(x)	0.333	0.333	0.333
M(x)	0.347	0.407	0.247

$$D_{\mathrm{KL}}(P \parallel M) = \sum_{x \in \mathcal{X}} P(x) \ln\left(\frac{P(x)}{M(x)}\right) = 0.36 \ln\left(\frac{0.36}{0.347}\right) + 0.48 \ln\left(\frac{0.48}{0.407}\right) + 0.16 \ln\left(\frac{0.16}{0.247}\right) = 0.02295$$

$$D_{\mathrm{KL}}(Q \parallel M) = \sum_{x \in \mathcal{X}} Q(x) \ln\left(\frac{Q(x)}{M(x)}\right) = 0.333 \ln\left(\frac{0.333}{0.347}\right) + 0.333 \ln\left(\frac{0.333}{0.407}\right) + 0.333 \ln\left(\frac{0.333}{0.247}\right) = 0.01895$$

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M) = 0.02095$$

JS-дивергенция. Дискретный случай

JS-дивергенция своими руками

```
def js_divergence(p, q):
    m = (p + q) / 2
    return (kl_divergence(p, m) + kl_divergence(q, m)) / 2
```

```
print ("%.7f" % js_divergence(binom_pmf, uniform_pmf))
```

```
0.0224599
```

```
print ("%.7f" % js_divergence(uniform_pmf, binom_pmf))
```

```
0.0224599
```

https://github.com/BorisLestsov/MADE/blob/master/seminar10-gan/lecture_experiments.ipynb

JS-дивергенция. Дискретный случай

JS-дивергенция

```python
# https://scipy.github.io/devdocs/generated/scipy.spatial.distance.jensenshannon.html
from scipy.spatial.distance import jensenshannon
```

Compute the Jensen-Shannon distance (metric) between two 1-D probability arrays. This is the square root of the Jensen-Shannon divergence. The Jensen-Shannon distance between two probability vectors `p` and `q` is defined as, $\sqrt{\frac{D(p\|m)+D(q\|m)}{2}}$, where `m` is the pointwise mean of `p` and `q` and :math: `D` is the Kullback-Leibler divergence.

```python
print ("%.7f" % jensenshannon(binom_pmf, uniform_pmf) ** 2)
```

```
0.0224599
```

```python
print ("%.7f" % jensenshannon(uniform_pmf, binom_pmf) ** 2)
```

```
0.0224599
```

https://github.com/BorisLestsov/MADE/blob/master/seminar10-gan/lecture_experiments.ipynb

Оригинальная статья. Функционал потерь

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

https://arxiv.org/pdf/1406.2661.pdf https://medium.com/@jonathan_hui/proof-gan-optimal-point-658116a236fb

Оригинальная статья. Функция потерь

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

В оптимальной точке: $p_{data} = p_{gen}$, $D^*(x) = \dfrac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)} = \dfrac{1}{2}$

https://arxiv.org/pdf/1406.2661.pdf https://medium.com/@jonathan_hui/proof-gan-optimal-point-658116a236fb

Оригинальная статья. Функция потерь

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

В оптимальной точке: $p_{data} = p_{gen}$, $D^*(x) = \dfrac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)} = \dfrac{1}{2}$

$$\begin{aligned}\min_G V(D^*, G) &= \int_x \left(p_r(x)\log D^*(x) + p_g(x)\log(1 - D^*(x))\right)dx \\ &= \int_x \left(p_r(x)\log\frac{p_r(x)}{p_r(x)+p_g(x)} + p_g(x)\log\frac{p_g(x)}{p_r(x)+p_g(x)}\right)dx\end{aligned}$$

https://arxiv.org/pdf/1406.2661.pdf https://medium.com/@jonathan_hui/proof-gan-optimal-point-658116a236fb

Оригинальная статья. Функция потерь

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

В оптимальной точке: $p_{data} = p_{gen}$, $D^*(x) = \dfrac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)} = \dfrac{1}{2}$

$$\begin{aligned}\min_G V(D^*, G) &= \int_x \left(p_r(x)\log D^*(x) + p_g(x)\log(1 - D^*(x))\right)dx \\ &= \int_x \left(p_r(x)\log\frac{p_r(x)}{p_r(x)+p_g(x)} + p_g(x)\log\frac{p_g(x)}{p_r(x)+p_g(x)}\right)dx\end{aligned}$$

$$\begin{aligned}D_{JS}(p_r\|p_g) &= \frac{1}{2}D_{KL}(p_r||\frac{p_r+p_g}{2}) + \frac{1}{2}D_{KL}(p_g||\frac{p_r+p_g}{2}) \\ &= \frac{1}{2}\left(\int_x p_r(x)\log\frac{2p_r(x)}{p_r(x)+p_g(x)}dx\right) + \frac{1}{2}\left(\int_x p_g(x)\log\frac{2p_g(x)}{p_r(x)+p_g(x)}dx\right) \\ &= \frac{1}{2}\left(\log 2 + \int_x p_r(x)\log\frac{p_r(x)}{p_r(x)+p_g(x)}dx\right) + \\ &\quad \frac{1}{2}\left(\log 2 + \int_x p_g(x)\log\frac{p_g(x)}{p_r(x)+p_g(x)}dx\right) \\ &= \frac{1}{2}\left(\log 4 + \min_G V(D^*, G)\right)\end{aligned}$$

https://arxiv.org/pdf/1406.2661.pdf https://medium.com/@jonathan_hui/proof-gan-optimal-point-658116a236fb

Оригинальная статья. Функция потерь

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

В оптимальной точке: $p_{data} = p_{gen}$, $D^*(x) = \dfrac{p_{data}(x)}{p_{data}(x) + p_{gen}(x)} = \dfrac{1}{2}$

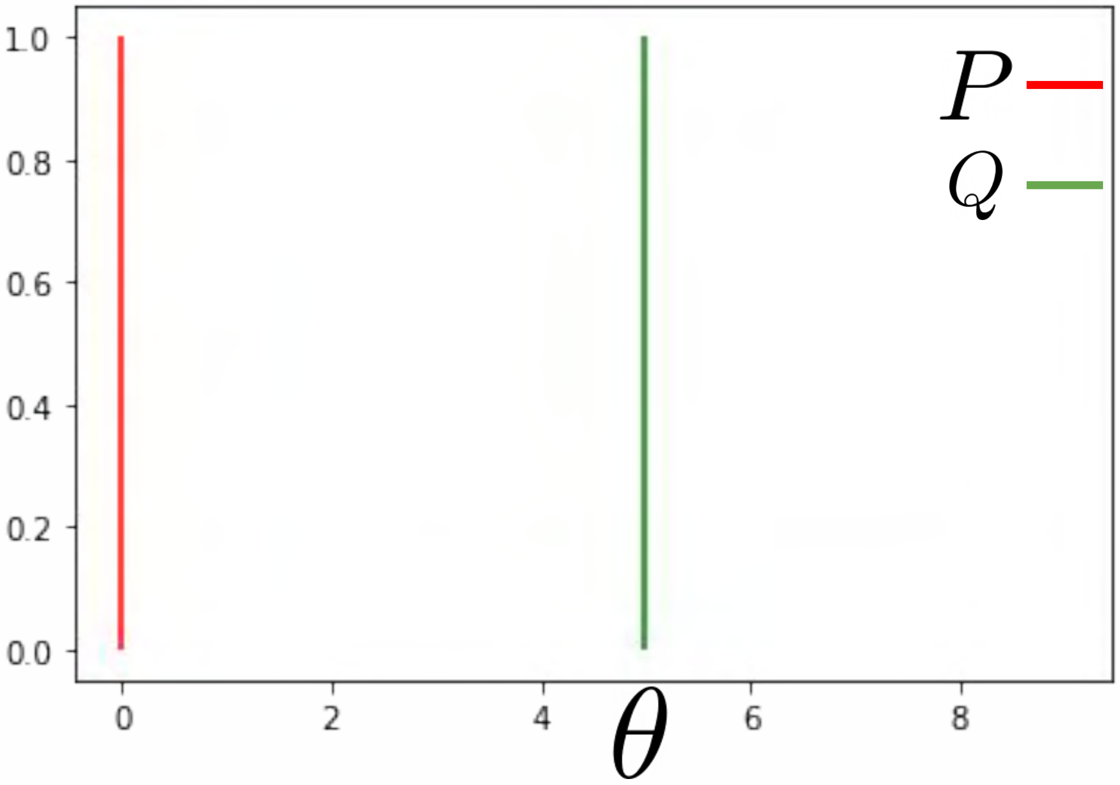
$$\begin{aligned}\min_G V(D^*, G) &= \int_x \left(p_r(x) \log D^*(x) + p_g(x) \log(1 - D^*(x)) \right) dx \\ &= \int_x \left(p_r(x) \log \frac{p_r(x)}{p_r(x) + p_g(x)} + p_g(x) \log \frac{p_g(x)}{p_r(x) + p_g(x)} \right) dx\end{aligned}$$

$$\begin{aligned}D_{JS}(p_r \| p_g) &= \frac{1}{2} D_{KL}(p_r || \frac{p_r + p_g}{2}) + \frac{1}{2} D_{KL}(p_g || \frac{p_r + p_g}{2}) \\ &= \frac{1}{2}\left(\int_x p_r(x) \log \frac{2p_r(x)}{p_r(x) + p_g(x)} dx \right) + \frac{1}{2}\left(\int_x p_g(x) \log \frac{2p_g(x)}{p_r(x) + p_g(x)} dx \right) \\ &= \frac{1}{2}\left(\log 2 + \int_x p_r(x) \log \frac{p_r(x)}{p_r(x) + p_g(x)} dx \right) + \\ &\quad \frac{1}{2}\left(\log 2 + \int_x p_g(x) \log \frac{p_g(x)}{p_r(x) + p_g(x)} dx \right) \\ &= \frac{1}{2}\left(\log 4 + \min_G V(D^*, G) \right)\end{aligned}$$

$$L(G, D) = -2\log(2) + 2D_{JS}(p_r \parallel p_g)$$

https://arxiv.org/pdf/1406.2661.pdf https://medium.com/@jonathan_hui/proof-gan-optimal-point-658116a236fb

Проблемы KL и JS дивергенций. Пример



$$\forall (x, y) \in P, x = 0 \text{ and } y \sim U(0,1)$$

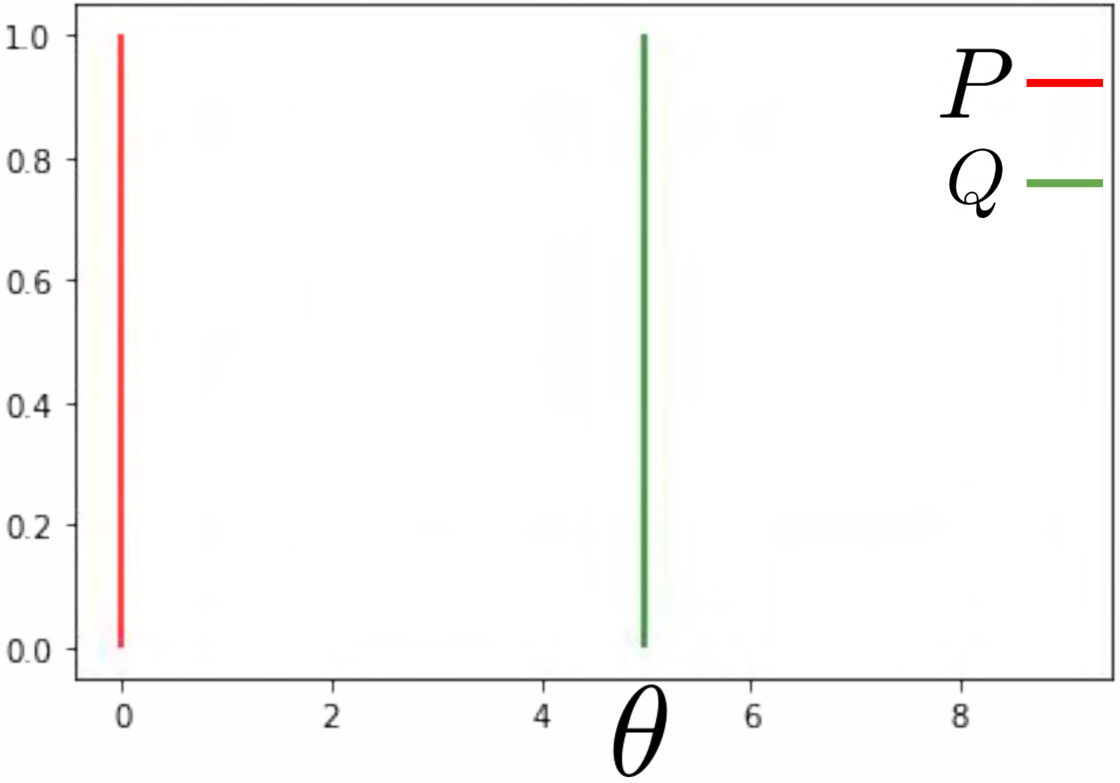
$$\forall (x, y) \in Q, x = \theta, 0 \leq \theta \leq \infty \text{ and } y \sim U(0,1)$$

$$D_{\mathrm{KL}}(P \parallel Q) =$$

$$\mathrm{JSD}(P \parallel Q) =$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример



$$\forall (x, y) \in P, x = 0 \text{ and } y \sim U(0,1)$$

$$\forall (x, y) \in Q, x = \theta, 0 \le \theta \le \infty \text{ and } y \sim U(0,1)$$

$$D_{KL}(P\|Q) = \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{KL}(Q\|P) = \sum_{x=\theta, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

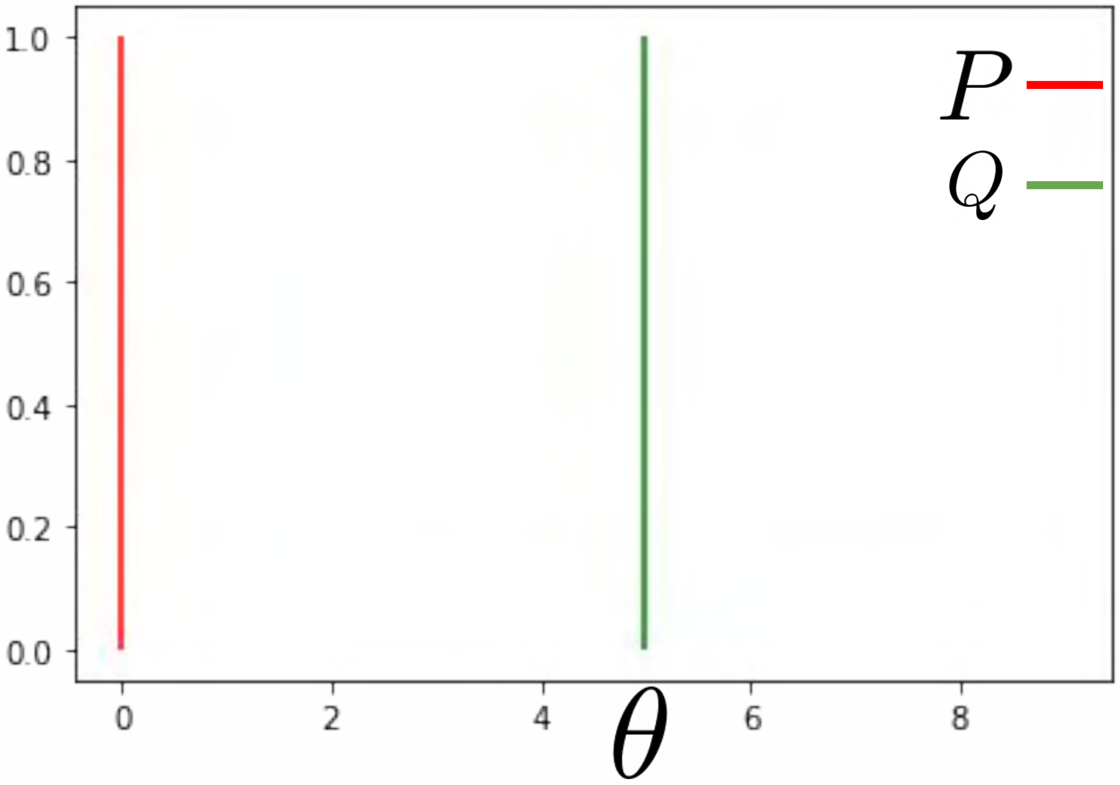
$$D_{\mathrm{KL}}(P \parallel Q) =$$

$$\mathrm{JSD}(P \parallel Q) =$$

https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html#kullbackleibler-and-jensenshannon-divergence

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример



$$\forall (x, y) \in P, x = 0 \text{ and } y \sim U(0,1)$$

$$\forall (x, y) \in Q, x = \theta, 0 \leq \theta \leq \infty \text{ and } y \sim U(0,1)$$

$$D_{KL}(P \| Q) = \sum_{x=0, y \sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{KL}(Q \| P) = \sum_{x=\theta, y \sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{JS}(P, Q) = \frac{1}{2}\left(\sum_{x=0, y \sim U(0,1)} 1 \cdot \log \frac{1}{1/2} + \sum_{x=0, y \sim U(0,1)} 1 \cdot \log \frac{1}{1/2}\right) = \log 2$$

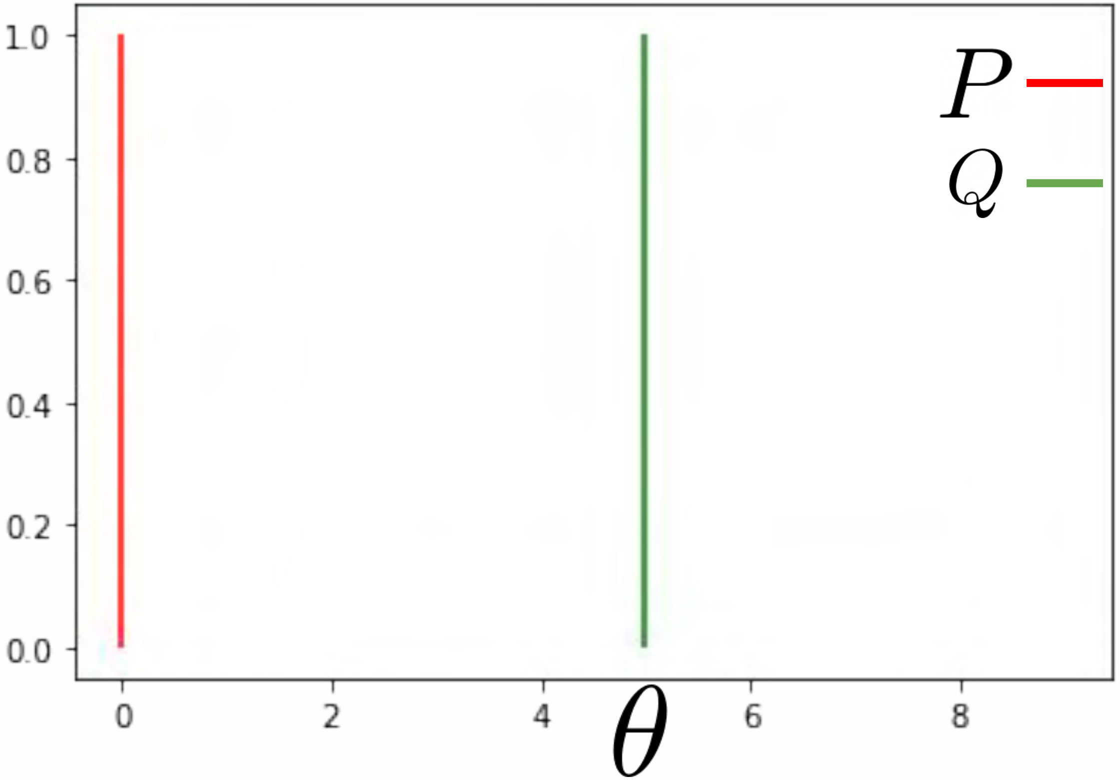
$$D_{\mathrm{KL}}(P \parallel Q) =$$

$$\mathrm{JSD}(P \parallel Q) =$$

https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html#kullbackleibler-and-jensenshannon-divergence

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример



$$\forall(x, y) \in P, x = 0 \text{ and } y \sim U(0, 1)$$

$$\forall(x, y) \in Q, x = \theta, 0 \leq \theta \leq \infty \text{ and } y \sim U(0, 1)$$

$$D_{KL}(P\|Q) = \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{KL}(Q\|P) = \sum_{x=\theta, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{JS}(P, Q) = \frac{1}{2}\left(\sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2} + \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2}\right) = \log 2$$

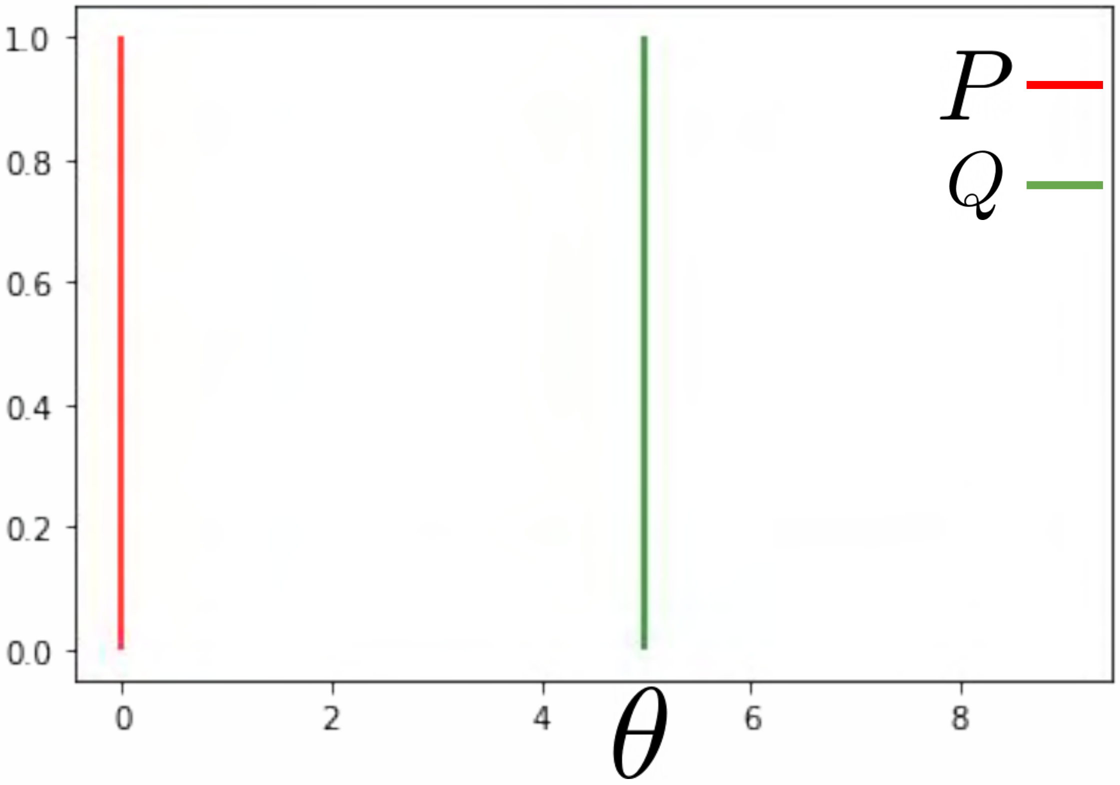
$$D_{\mathrm{KL}}(P \parallel Q) = \infty$$

$$\mathrm{JSD}(P \parallel Q) = \ln(2) = 0.69314$$

https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html#kullbackleibler-and-jensenshannon-divergence

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS



$$D_{\mathrm{KL}}(P \parallel Q) = \infty$$

$$\mathrm{JSD}(P \parallel Q) = \ln(2) = 0.693$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

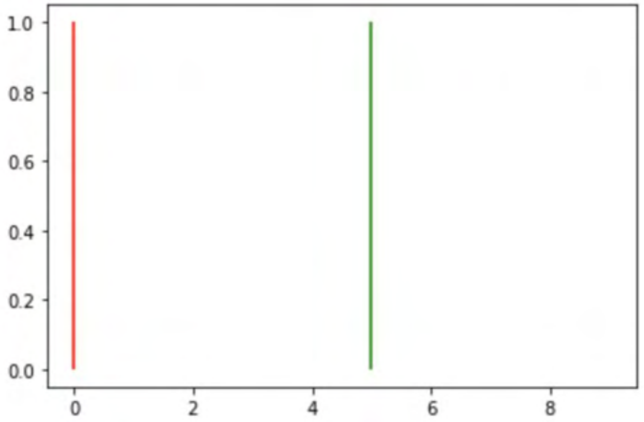
```
# https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.unit_impulse.h
from scipy.signal import unit_impulse
```

```
omega = 5
n_observations = 10
```

```
p = unit_impulse(n_observations)
q = unit_impulse(n_observations, idx=omega)
print ('p: {}'.format(p))
print ('q: {}'.format(q))
```

```
p: [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
q: [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
```

```
plt.vlines(np.arange(len(p)), 0, p, colors='r')
plt.vlines(np.arange(len(p)), 0, q, colors='g')
plt.show()
```



$D_{\mathrm{KL}}(P \parallel Q)$

```
np.sum(kl_div(p, q))
```

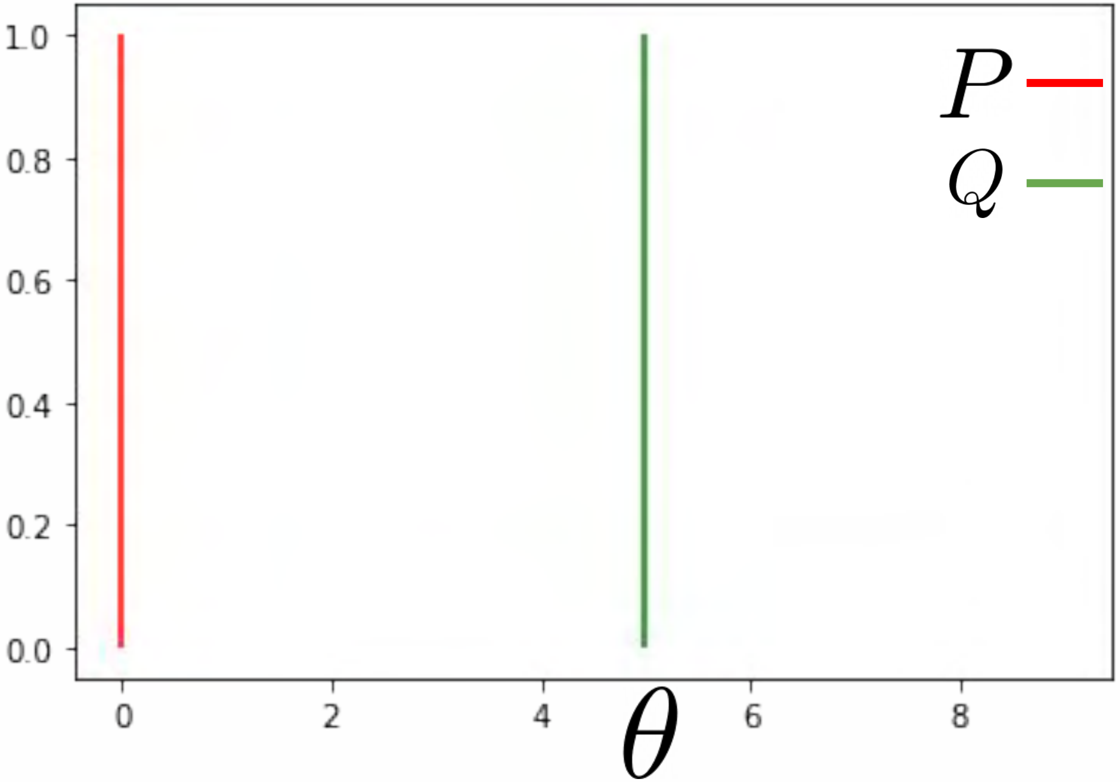
```
inf
```

$JSD(P \parallel Q)$

```
jensenshannon(p, q) ** 2 # это ln(2)
```

```
0.6931471805599452
```

Проблемы KL и JS дивергенций. Пример

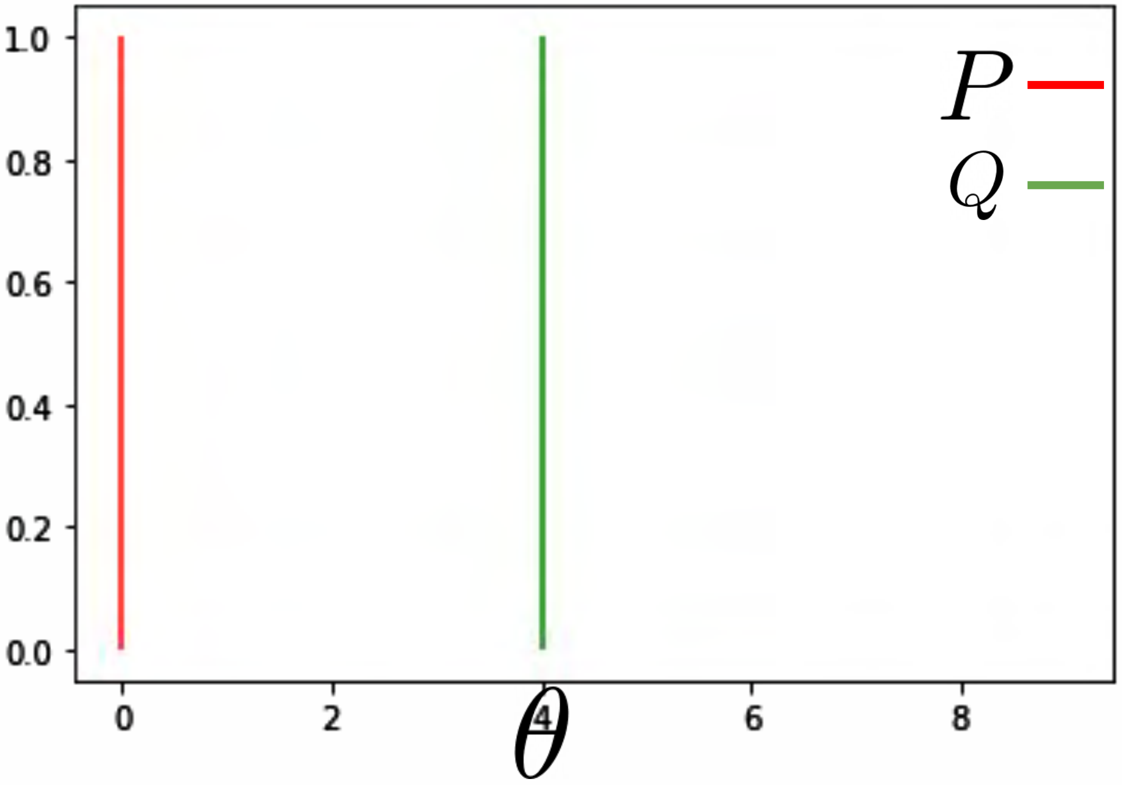


$$D_{\mathrm{KL}}(P \parallel Q) = \infty$$

$$\mathrm{JSD}(P \parallel Q) = \ln(2) = 0.69314$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример

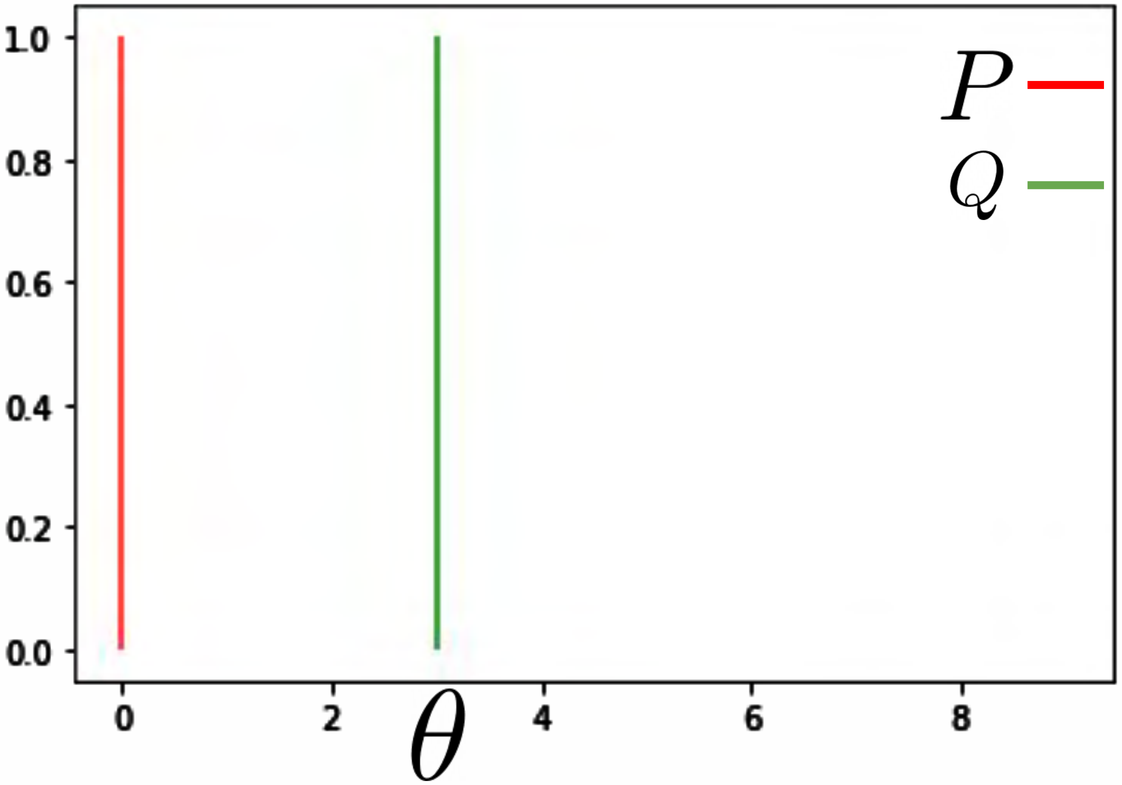


$$D_{\mathrm{KL}}(P \parallel Q) = \infty$$

$$\mathrm{JSD}(P \parallel Q) = \ln(2) = 0.69314$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример

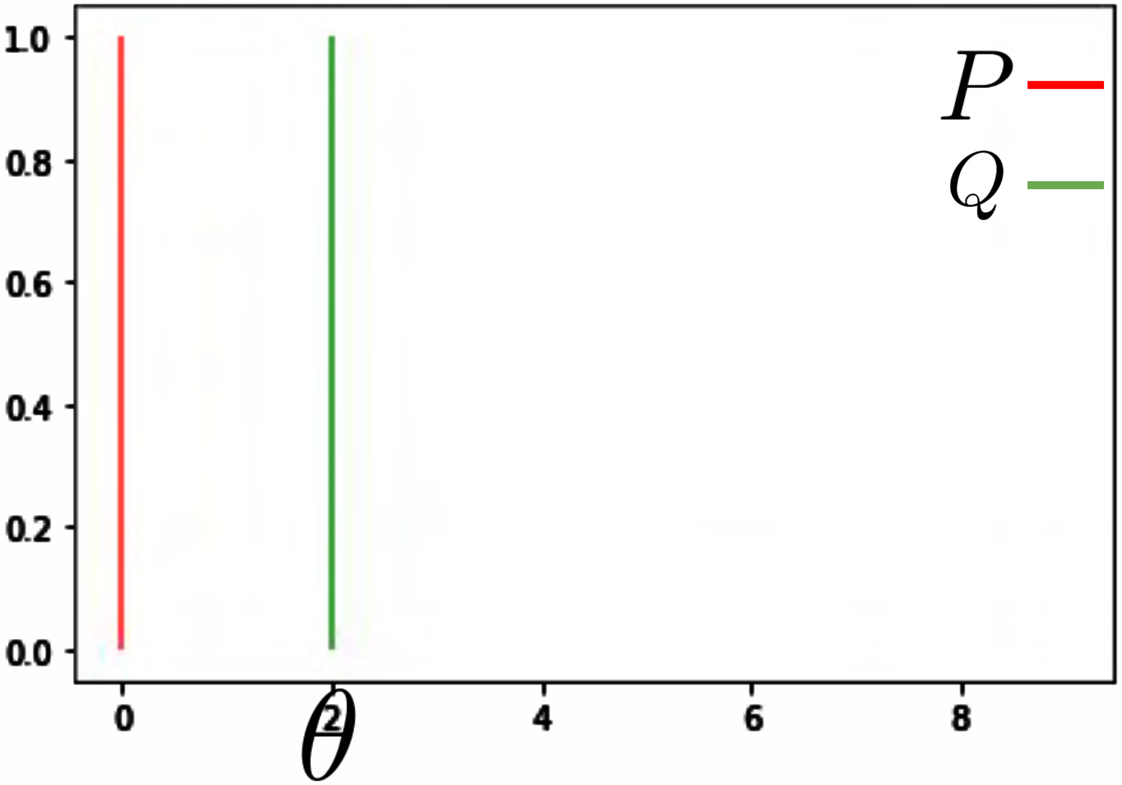


$$D_{\mathrm{KL}}(P \parallel Q) = \infty$$

$$\mathrm{JSD}(P \parallel Q) = \ln(2) = 0.69314$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример

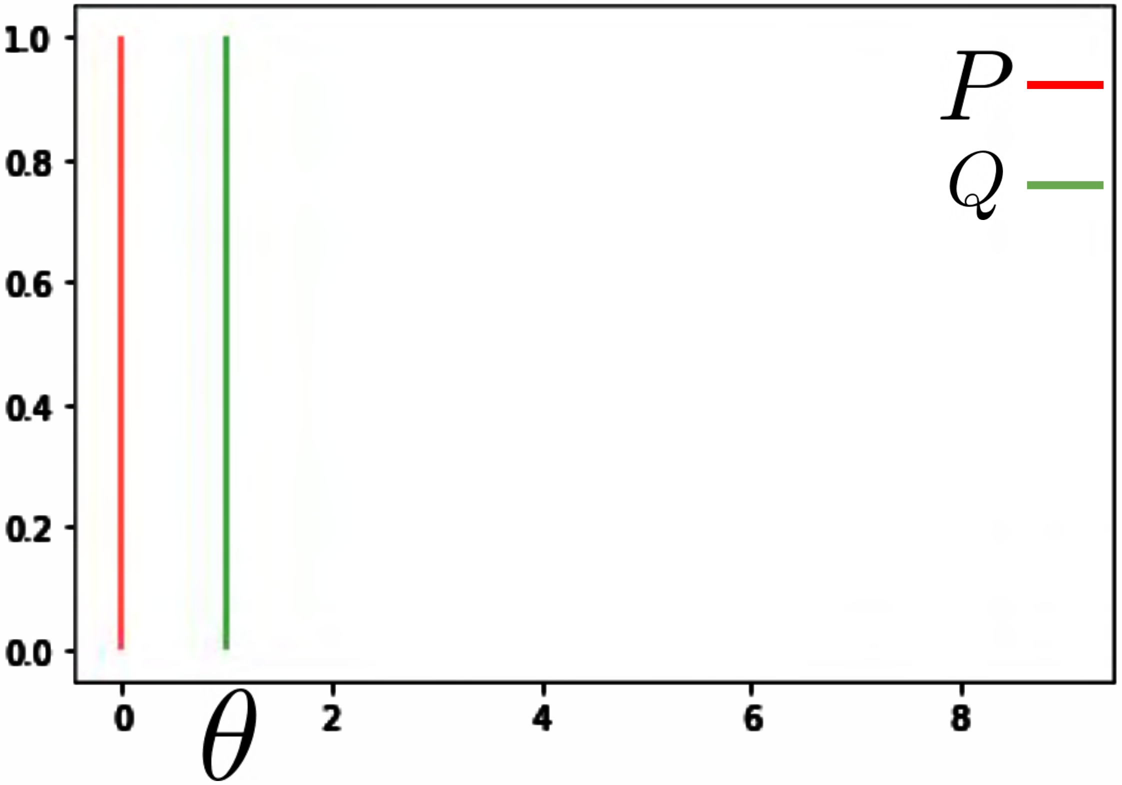


$$D_{\mathrm{KL}}(P \parallel Q) = \infty$$

$$\mathrm{JSD}(P \parallel Q) = \ln(2) = 0.69314$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример

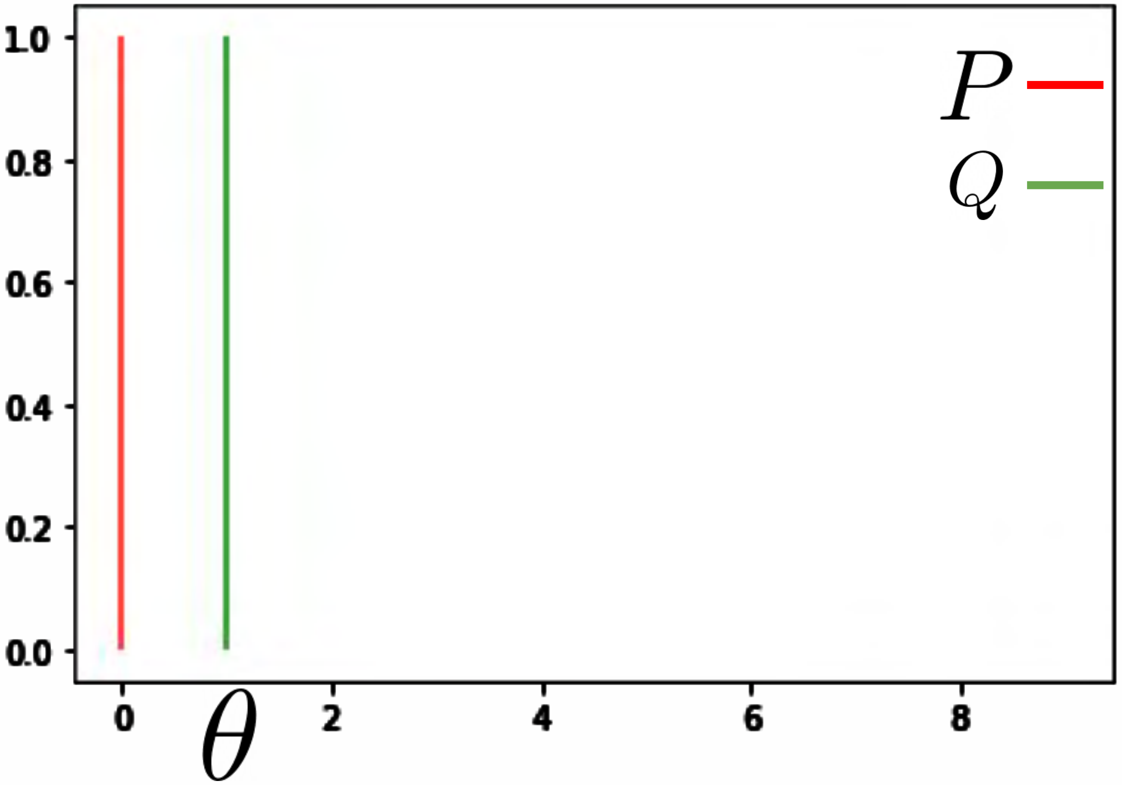


$$D_{\mathrm{KL}}(P \parallel Q) = \infty$$

$$\mathrm{JSD}(P \parallel Q) = \ln(2) = 0.69314$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример

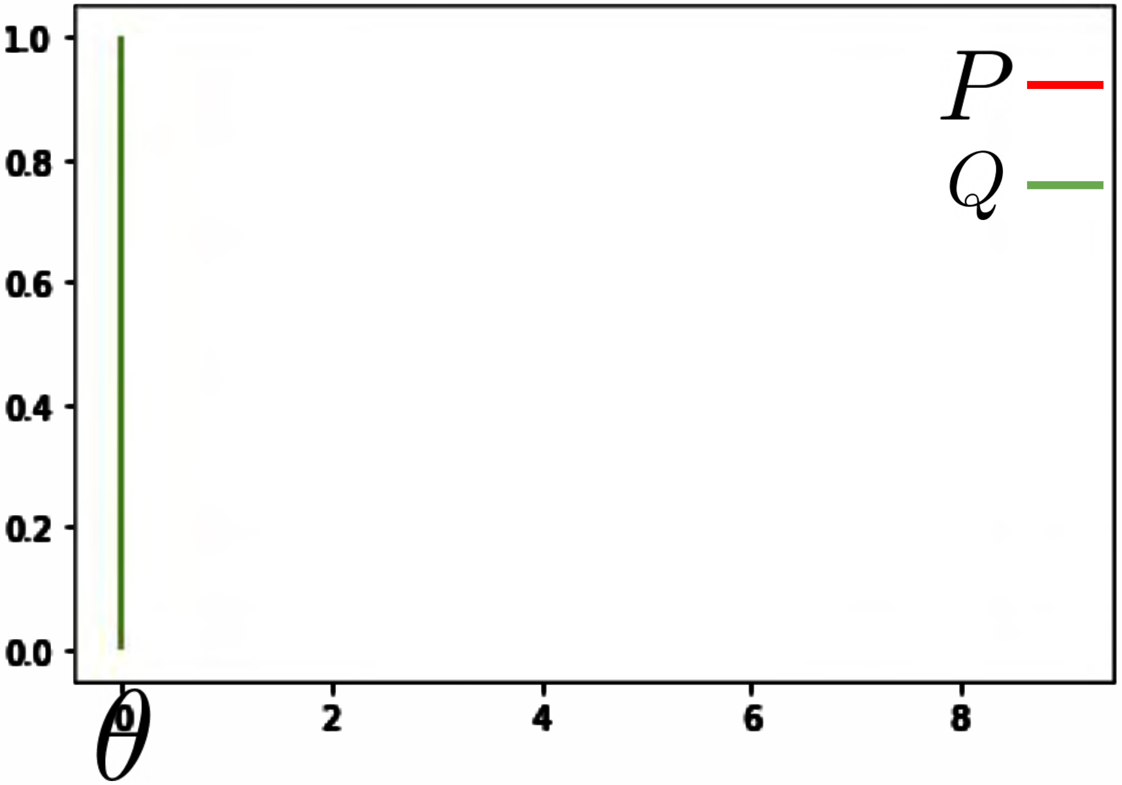


$$D_{\mathrm{KL}}(P \parallel Q) = \infty$$

$$\mathrm{JSD}(P \parallel Q) = \ln(2) = 0.69314$$

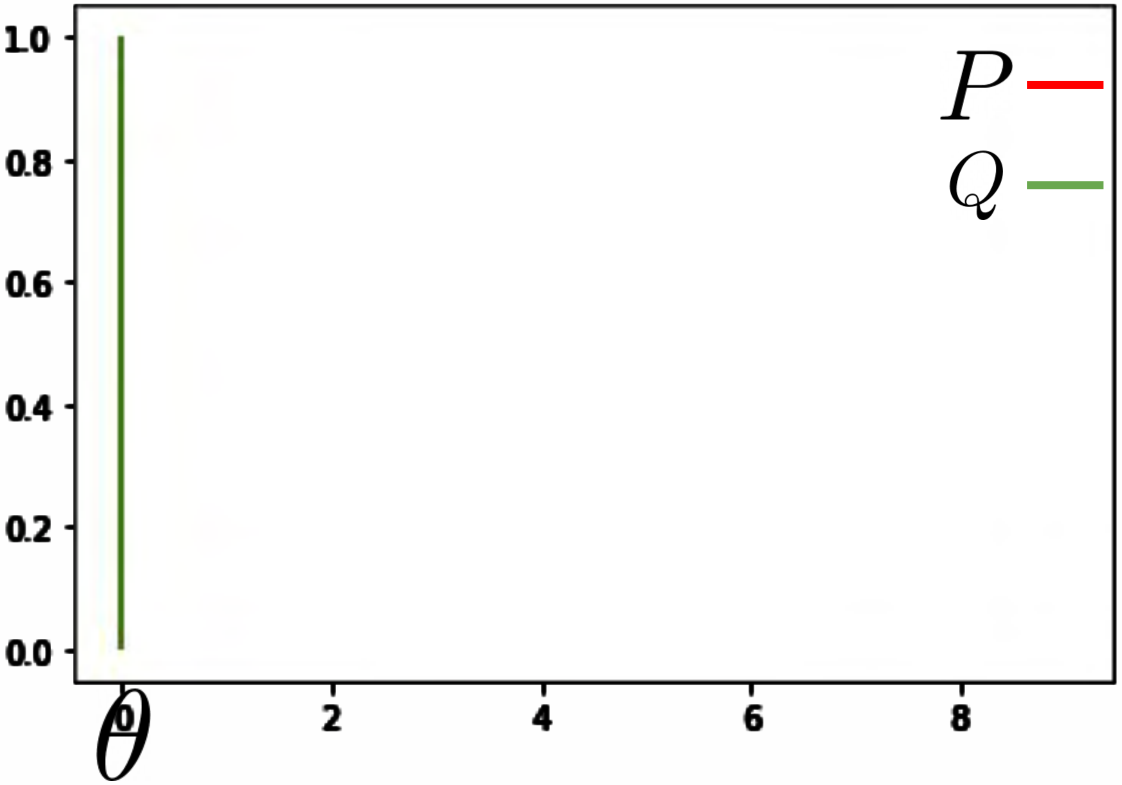
https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример



https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример



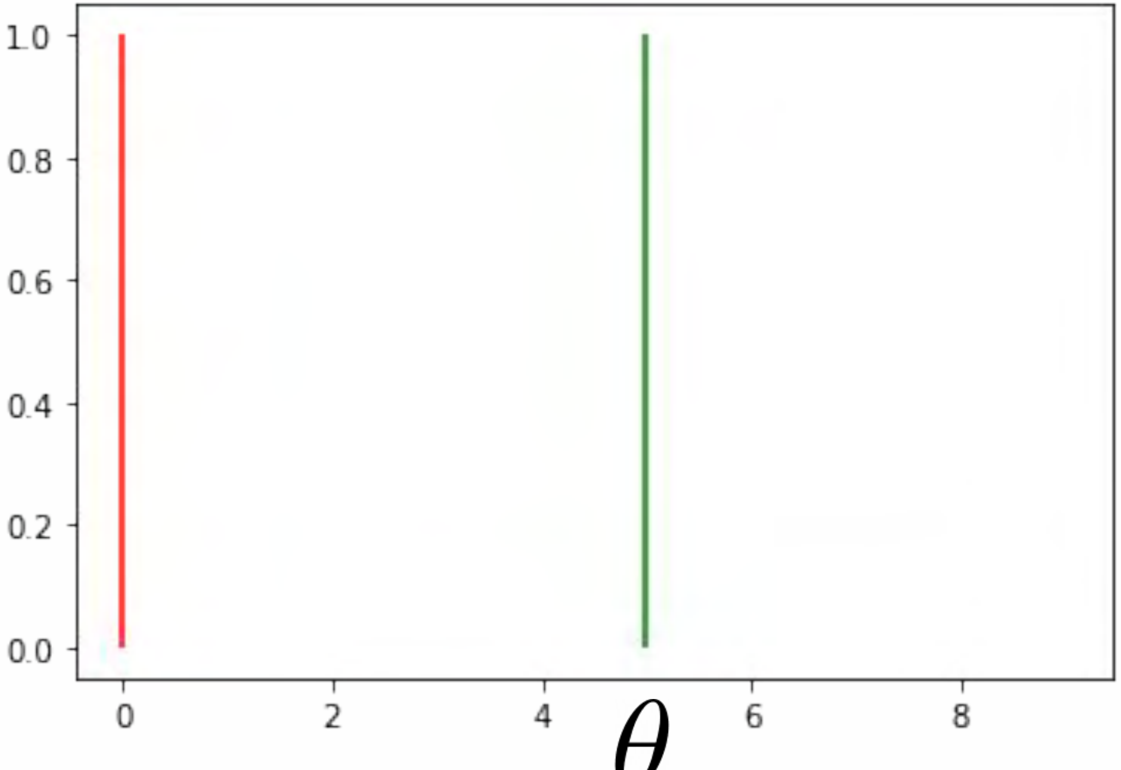
$$D_{\mathrm{KL}}(P \parallel Q) = 0$$

$$\mathrm{JSD}(P \parallel Q) = 0$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример

$$L(G, D) = -2\log(2) + 2D_{JS}(p_r \parallel p_g)$$



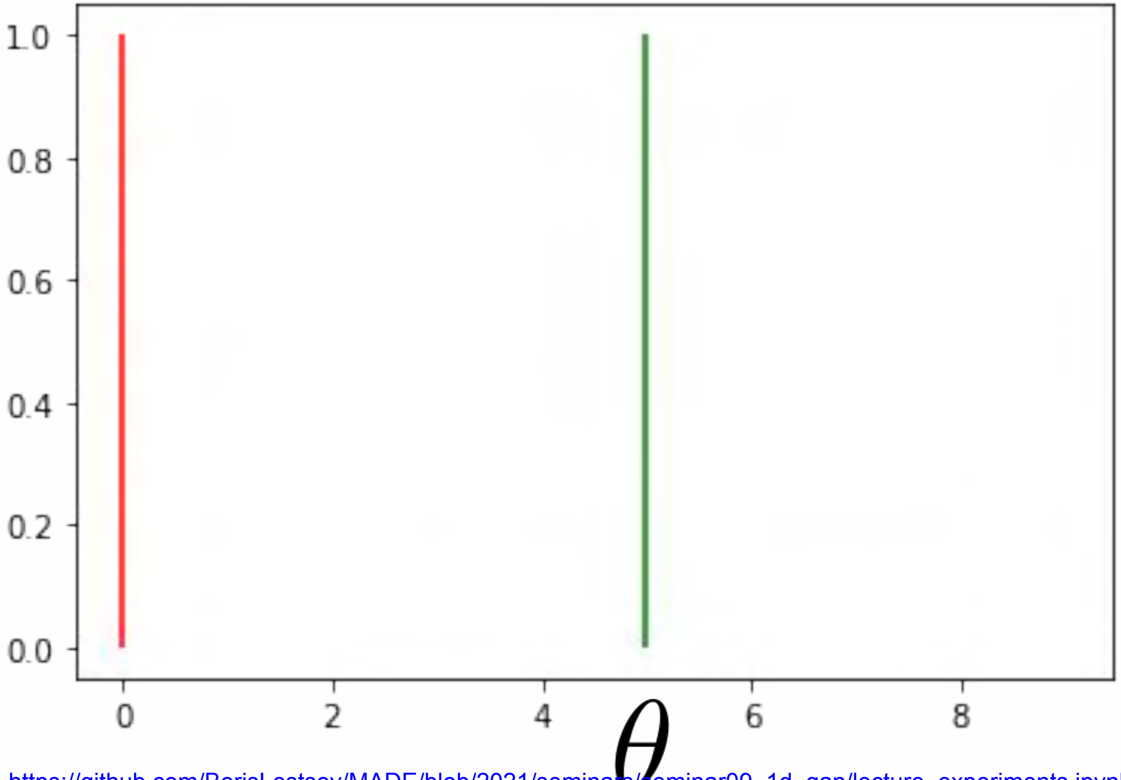
$$D_{\mathrm{KL}}(P \parallel Q) = \begin{cases} \infty & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

$$JSD(P \parallel Q) = \begin{cases} \ln(2) & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример

$$L(G, D) = -2\log(2) + 2D_{JS}(p_r \parallel p_g)$$



$$D_{\mathrm{KL}}(P \parallel Q) = \begin{cases} \infty & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

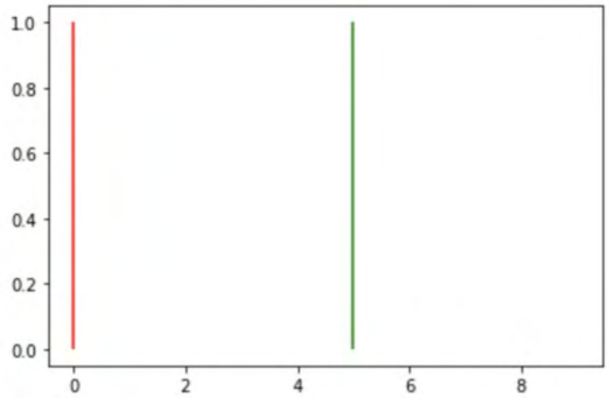
$$JSD(P \parallel Q) = \begin{cases} \ln(2) & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

$$W_1(P \parallel Q) = \theta$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

Проблемы KL и JS дивергенций. Пример

```
--------------------------------
KL div: inf,
JS div: 0.6931471805599452,
EMD: 5.0
```




```
================================
KL div: inf,
JS div: 0.6931471805599452,
EMD: 4.0
```

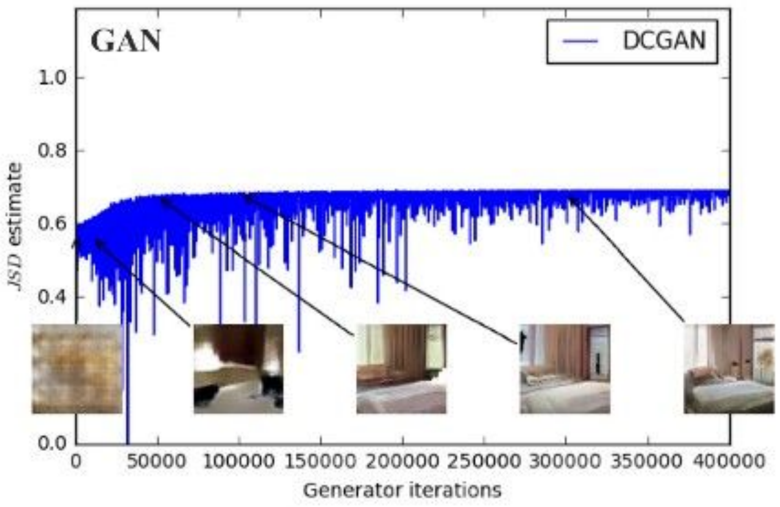
$$D_{\mathrm{KL}}(P \parallel Q) = \begin{cases} \infty & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

$$JSD(P \parallel Q) = \begin{cases} \ln(2) & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

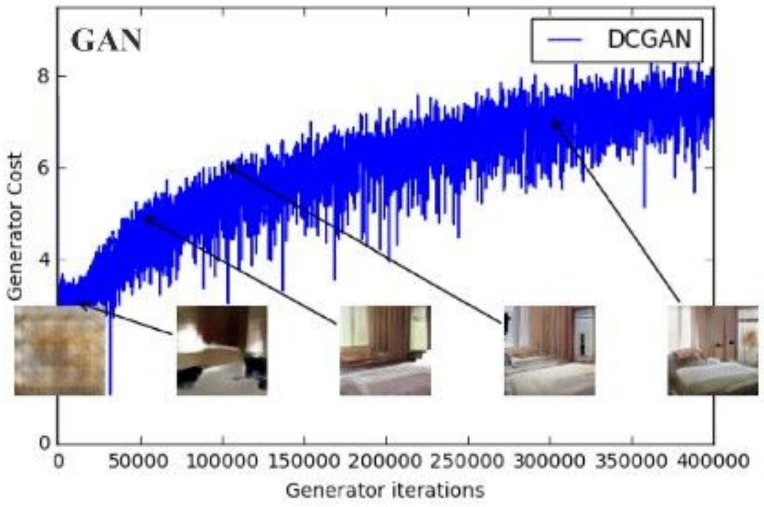
$$W_1(P \parallel Q) = \theta$$

https://github.com/BorisLestsov/MADE/blob/2021/seminars/seminar09_1d_gan/lecture_experiments.ipynb

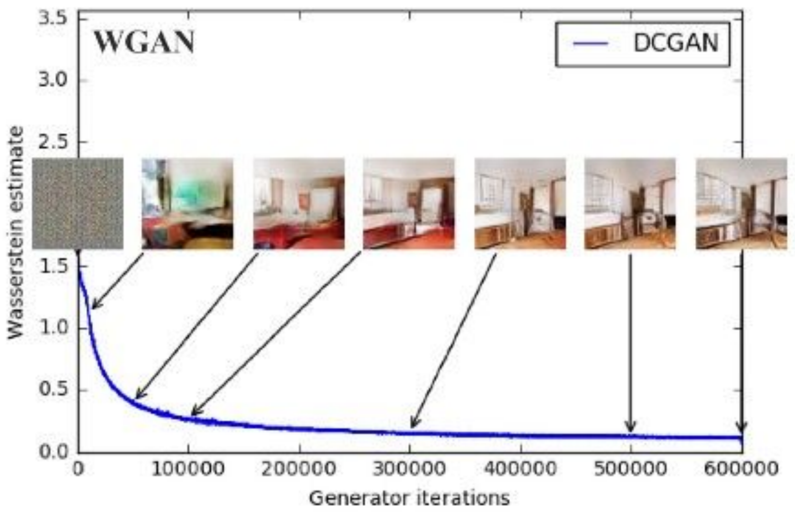
Wasserstein GAN. Качество зависит от лосса



$$\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)$$



$$\frac{1}{m}\sum_{i=1}^{m}-\log\left(D\left(G\left(z^{(i)}\right)\right)\right)$$



$$\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$$

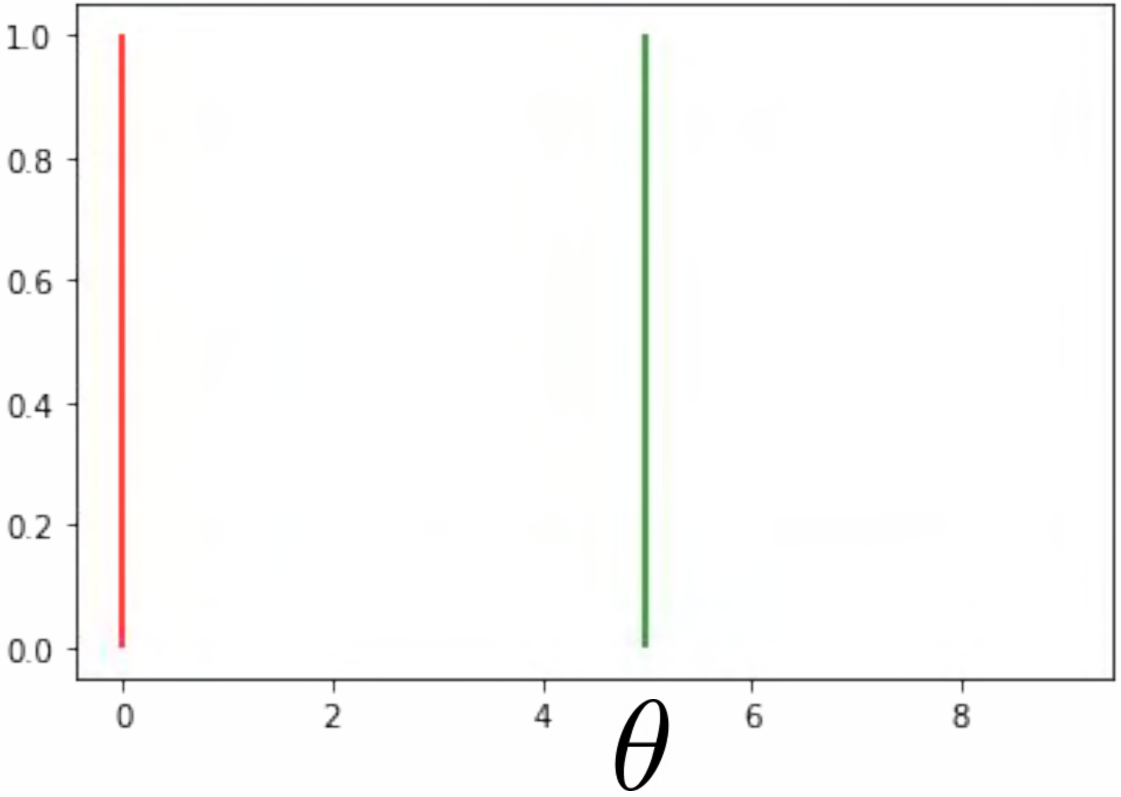
https://arxiv.org/abs/1701.07875 https://medium.com/@jonathan_hui/gan-what-is-wrong-with-the-gan-cost-function-6f594162ce01

Что мы уже прошли. Часть 2

- Оригинальная статья:
  - учим дискриминатор как бинарный классификатор
  - используем функционал ошибки классификации
    - увидели, что он может быть выведен как сумма JS-дивергенции и константы
    - увидели, что это приводит к проблемам со сходимостью и интерпретируемостью лосса

Проблемы KL и JS дивергенций. Пример

$$L(G, D) = -2\log(2) + 2D_{JS}(p_r \parallel p_g)$$



$$D_{\mathrm{KL}}(P \parallel Q) = \begin{cases} \infty & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

$$JSD(P \parallel Q) = \begin{cases} \ln(2) & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

$$W_1(P \parallel Q) = \theta$$

EMD: Earth mover’s distance. План

- Определение
- Дискретный пример
- Лосс с дистанцией Вассерштейна

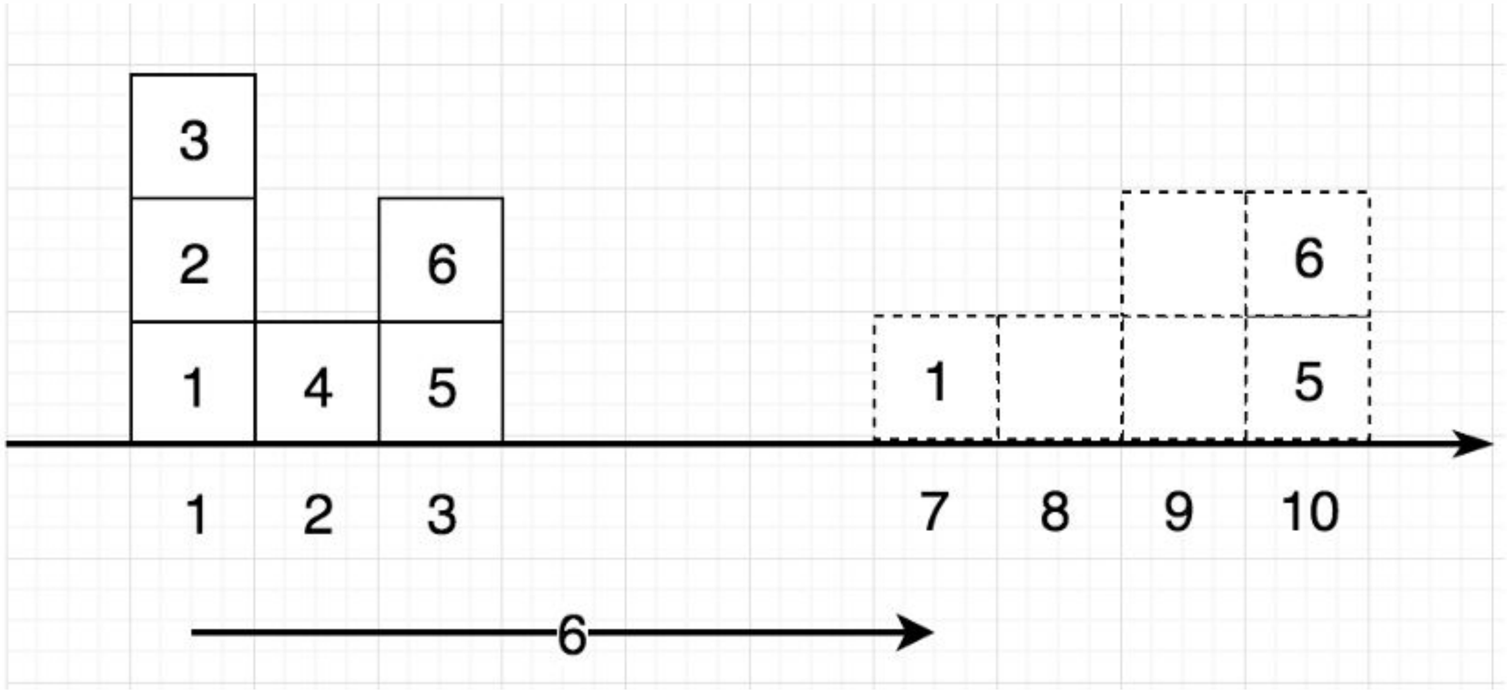
EMD: Earth mover's distance

- Метрика Вассерштейна (Wasserstein-1 distance, Earth Mover's Distance) - метрика, вычисляемая по формуле:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[\, \|x - y\| \,\right]$$

Также часто формулируется как минимальное количество энергии, которое нужно потратить, чтобы придать одной куче земли форму второй кучи земли

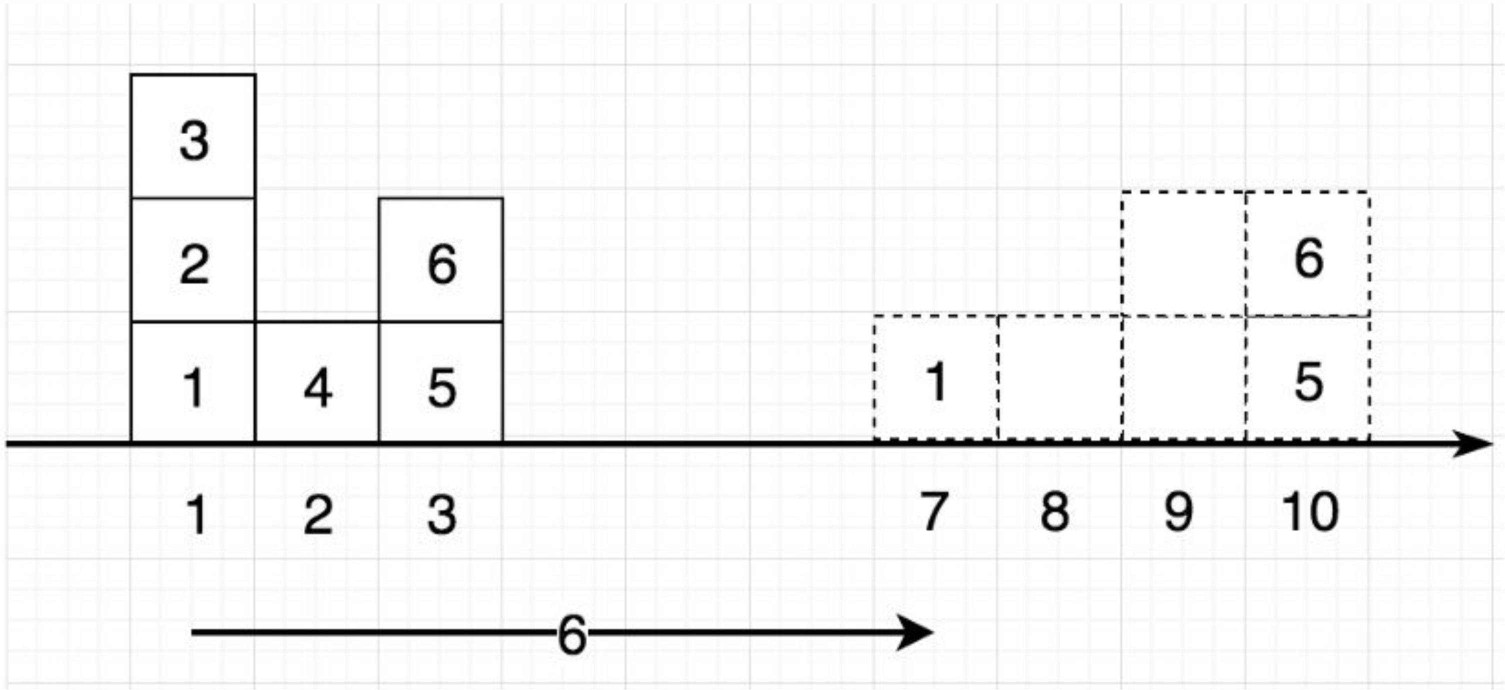
EMD: Earth mover’s distance. Пример



❖ Хотим поменять форму кучи из шести ящиков оптимальным способом

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

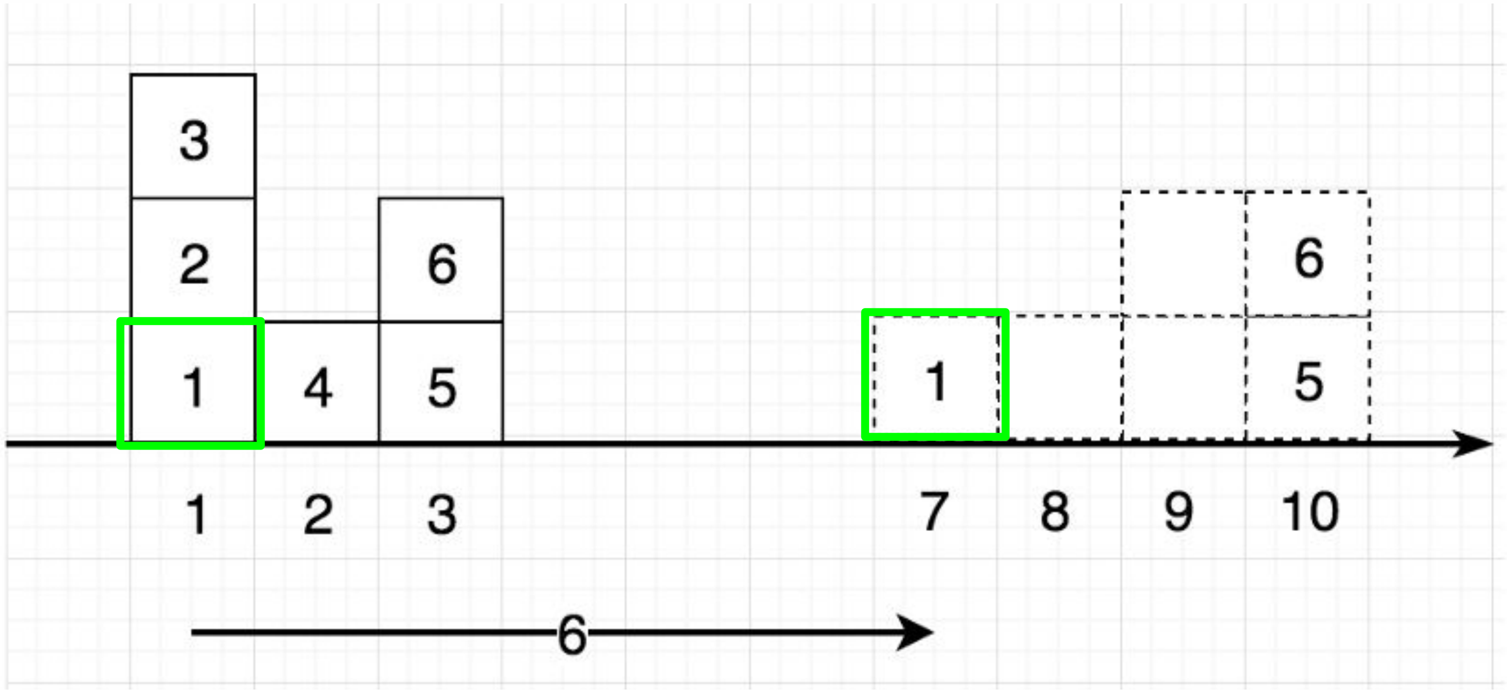
EMD: Earth mover’s distance. Пример



- Хотим поменять форму кучи из шести ящиков оптимальным способом
- Переносим бокс 1 из координаты 1 в координату 7. Стоимость 7 - 1 = 6

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

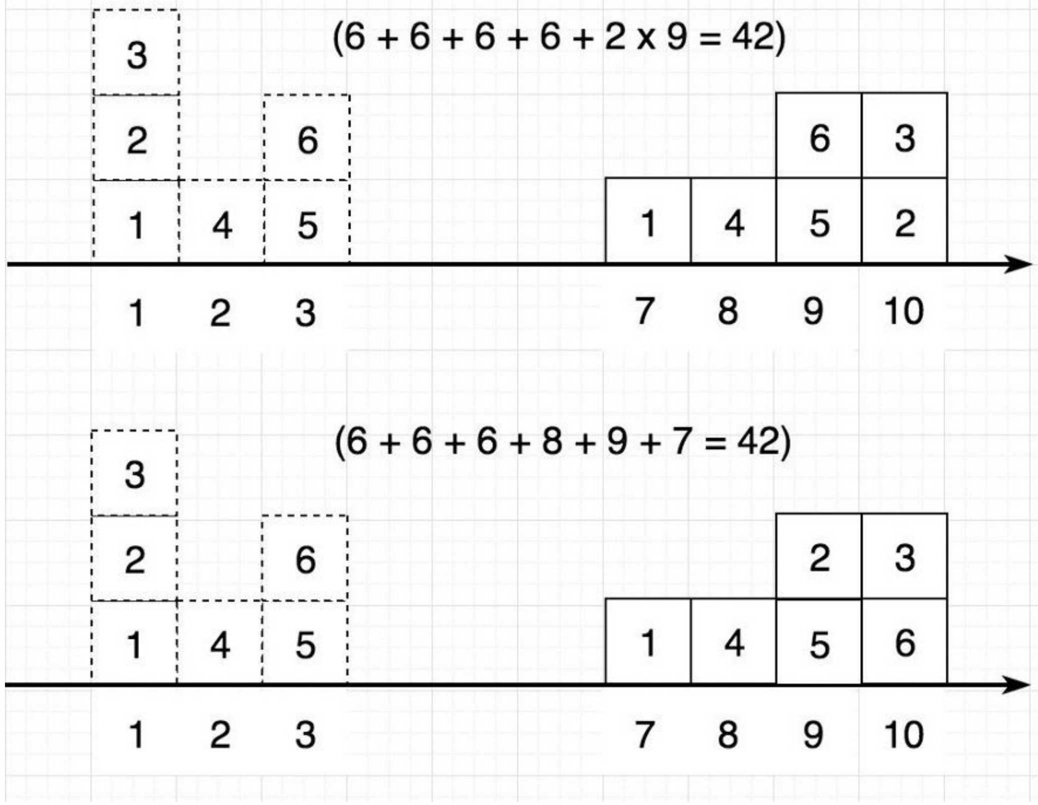
EMD: Earth mover’s distance. Пример



- Хотим поменять форму кучи из шести ящиков оптимальным способом
- Переносим бокс 1 из координаты 1 в координату 7. Стоимость 7 - 1 = 6

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

EMD: Earth mover’s distance. Пример



γ_1

	7	8	9	10
1	1	0	0	2
2	0	1	0	0
3	0	0	2	0

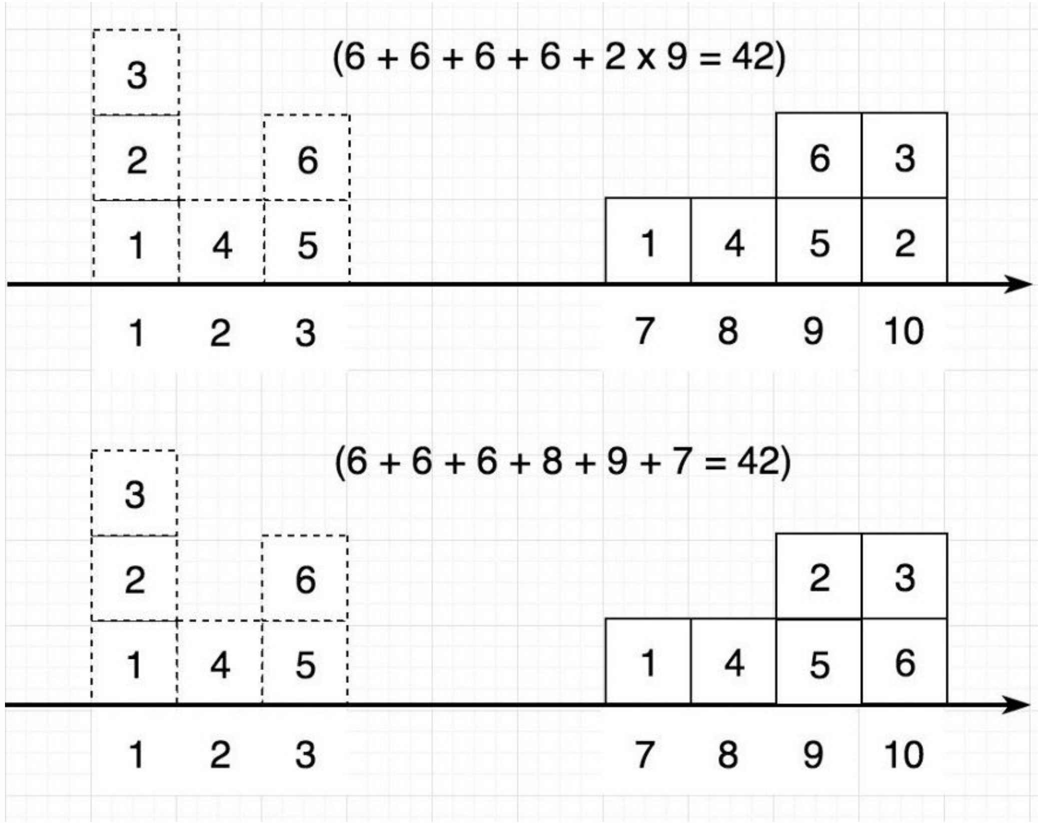
γ_2

	7	8	9	10
1	1	0	1	1
2	0	1	0	0
3	0	0	1	1

- Стратегий переноса может быть несколько.
- Обозначим γ_i i-ую стратегию переноса

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

EMD: Earth mover’s distance. Пример



γ_1	7	8	9	10
1	1	0	0	2
2	0	1	0	0
3	0	0	2	0

γ_2	7	8	9	10
1	1	0	1	1
2	0	1	0	0
3	0	0	1	1

- ❖ Стратегий переноса может быть несколько.
- ❖ Обозначим γ_i i-ую стратегию переноса
- ❖ Дистанция Вассерштейна - это стоимость самой дешевой стратегии переноса

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

EMD: Earth mover's distance.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

EMD: Earth mover's distance.

probability distribution of x & y
- before and after the move

a.k.a. min

expected moving cost of the plan γ

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

find one that has the lowest cost

all possible moving plans

distance between x & y

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

EMD: Earth mover's distance.

probability distribution of x & y
- before and after the move

a.k.a. min

expected moving cost of the plan γ

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

find one that has the lowest cost

all possible moving plans

distance between x & y

γ_1		7	8	9	10
	1	1	0	0	2
	2	0	1	0	0
	3	0	0	2	0

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

EMD: Earth mover's distance.

probability distribution of x & y
- before and after the move

a.k.a. min

expected moving cost of the plan γ

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \big[\, \|x - y\| \,\big]$$

find one that has the lowest cost

all possible moving plans

distance between x & y

γ_1

	7	8	9	10
1	1	0	0	2
2	0	1	0	0
3	0	0	2	0

γ_1

	7	8	9	10
1	$\frac{1}{6}$	0	0	$\frac{2}{6}$
2	0	$\frac{1}{6}$	0	0
3	0	0	$\frac{2}{6}$	0

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

EMD: Earth mover's distance.

probability distribution of x & y
- before and after the move

a.k.a. min

expected moving cost of the plan γ

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

find one that has the lowest cost

all possible moving plans

distance between x & y

γ_1

	7	8	9	10
1	1	0	0	2
2	0	1	0	0
3	0	0	2	0

γ_1

	7	8	9	10
1	$\frac{1}{6}$	0	0	$\frac{2}{6}$
2	0	$\frac{1}{6}$	0	0
3	0	0	$\frac{2}{6}$	0

$$\mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

$$= p(1, 7) \times 6 + p(1, 10) \times 9 + \ldots$$

$$= 1/6 \times 6 + 2/6 \times 9 + \ldots$$

https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

EMD: Earth mover's distance. Материалы

- https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html#kullbackleibler-and-jensenshannon-divergence
- https://vincentherrmann.github.io/blog/wasserstein/
- https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490
- https://medium.com/@jonathan_hui/gan-spectral-normalization-893b6a4e8f53

EMD: Earth mover’s distance.

- Проблема: EMD сложно считать

EMD: Earth mover’s distance.

- Проблема: EMD сложно считать
- Решение: можно переформулировать выражение EMD

EMD: Earth mover's distance.

- Проблема: EMD сложно считать
- Решение: можно переформулировать выражение EMD

Было:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

https://arxiv.org/pdf/1701.07875.pdf

EMD: Earth mover’s distance.

- Проблема: EMD сложно считать
- Решение: можно переформулировать выражение EMD

Было:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

Стало:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

Точная верхняя грань по всем функциям $f : \mathcal{X} \to \mathbb{R}$, непрерывных по Лишницу с константой меньшей или равной единице (Используется дуализм Канторовича-Рубинштейна)

https://arxiv.org/pdf/1701.07875.pdf

Идея: перейти от JS-дивергенции к W-1

Было: $$L(G, D) = -2\log(2) + 2D_{JS}(p_r \parallel p_g)$$

https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490

Идея: перейти от JS-дивергенции к W-1

Было: $$L(G,D) = -2\log(2) + 2D_{JS}(p_r \parallel p_g)$$

Стало: $$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \le 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490

Идея: перейти от JS-дивергенции к W-1

Было: $$L(G, D) = -2\log(2) + 2D_{JS}(p_r \parallel p_g)$$

Стало: $$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \le 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

	Discriminator/Critic	Generator
GAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$

https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490

Идея: перейти от JS-дивергенции к W-1

Было: $$L(G, D) = -2\log(2) + 2D_{JS}(p_r \parallel p_g)$$

Стало: $$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

	Discriminator/Critic	Generator
GAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$
WGAN	$\nabla_w \frac{1}{m} \sum_{i=1}^{m} \left[f\left(x^{(i)}\right) - f\left(G\left(z^{(i)}\right)\right)\right]$	$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f\left(G\left(z^{(i)}\right)\right)$

https://arxiv.org/abs/1701.07875

https://medium.com/@jonathan_hui/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490

https://www.alexirpan.com/2017/02/22/wasserstein-gan.html

Wasserstein GAN

- Решаемая задача:

WGAN $\nabla_w \frac{1}{m}\sum_{i=1}^{m}\left[f\left(x^{(i)}\right) - f\left(G\left(z^{(i)}\right)\right)\right]$ $\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f\left(G\left(z^{(i)}\right)\right)$

- Плюсы:
  - быстрее и стабильнее сходимость
  - меньше проблем с mode collapse
- Минусы:
  - Дискриминатор должен быть 1-Липшиц непрерывной функцией (этого сложно достичь)

https://arxiv.org/abs/1701.07875

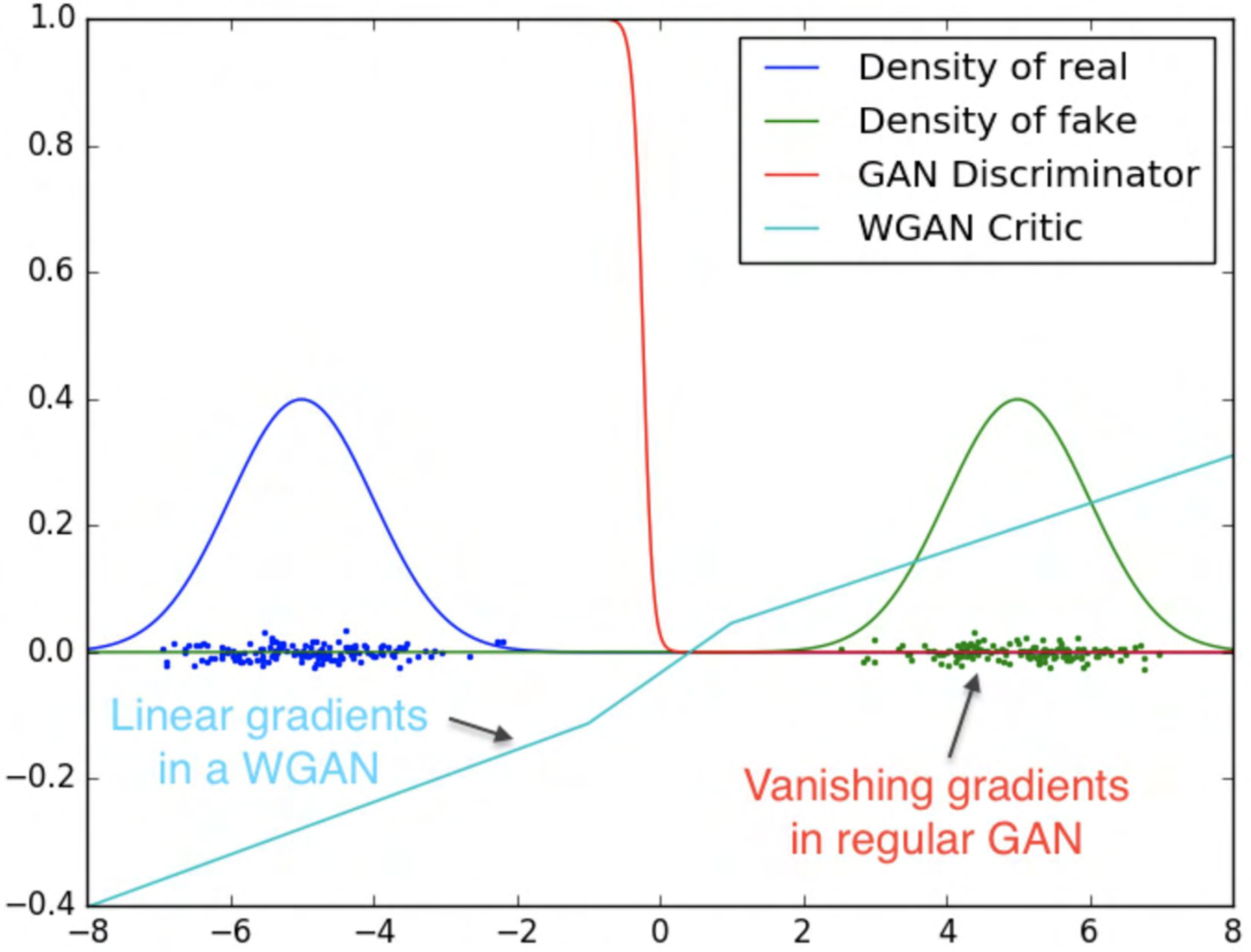


Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the discriminator of a minimax GAN saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

https://arxiv.c

Wasserstein GAN. Что почитать

- Оригинальная статья: https://arxiv.org/abs/1701.07875
- Обзор: https://www.alexirpan.com/2017/02/22/wasserstein-gan.html
- Разбор и объяснение деталей о Липшиц-непрерывности:

  https://vincentherrmann.github.io/blog/wasserstein/

Способы достижения Липшиц-непрерывности

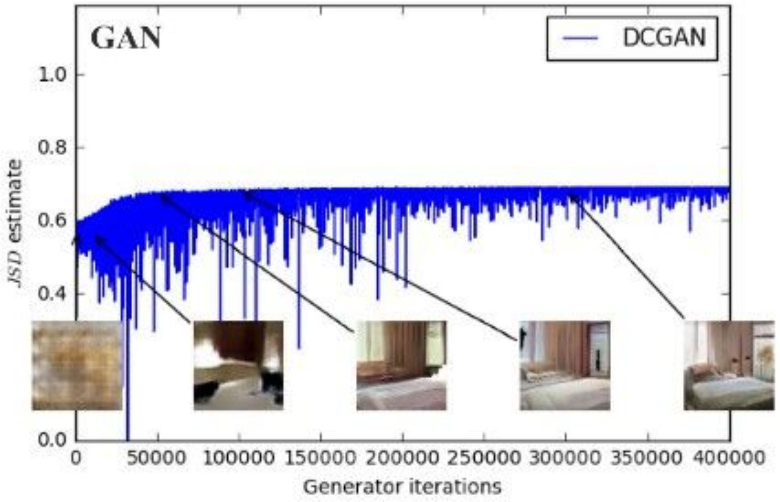
- Weight clipping - на каждой итерации обрезать веса до отрезка **[-c; c]**

$$w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$$
$$w \leftarrow \text{clip}(w, -c, c)$$

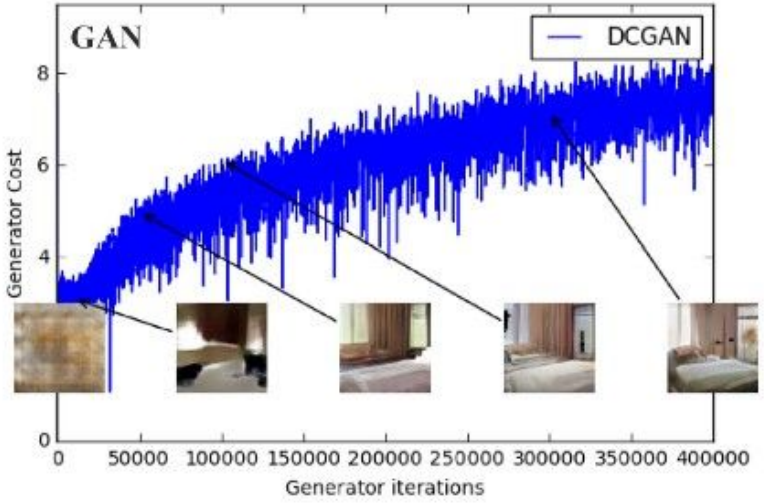
- Gradient Penalty (WGAN-GP) - на каждой итерации делим градиент на его норму, чтобы норма итогового градиента не превышала **1**
- Спектральная нормализация - делим каждый элемент матрицы весов на её спектальную норму
  - Спектральная норма - максимальное сингулярное число матрицы весов.

https://arxiv.org/abs/1701.07875 https://arxiv.org/pdf/1704.00028.pdf https://arxiv.org/abs/1802.05957

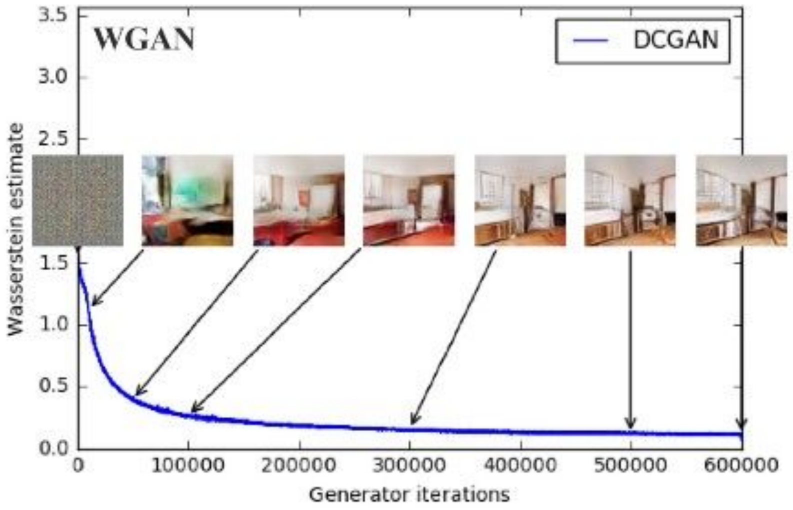
Wasserstein GAN. Качество зависит от лосса



$$\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(z^{(i)}\right)\right)\right)$$



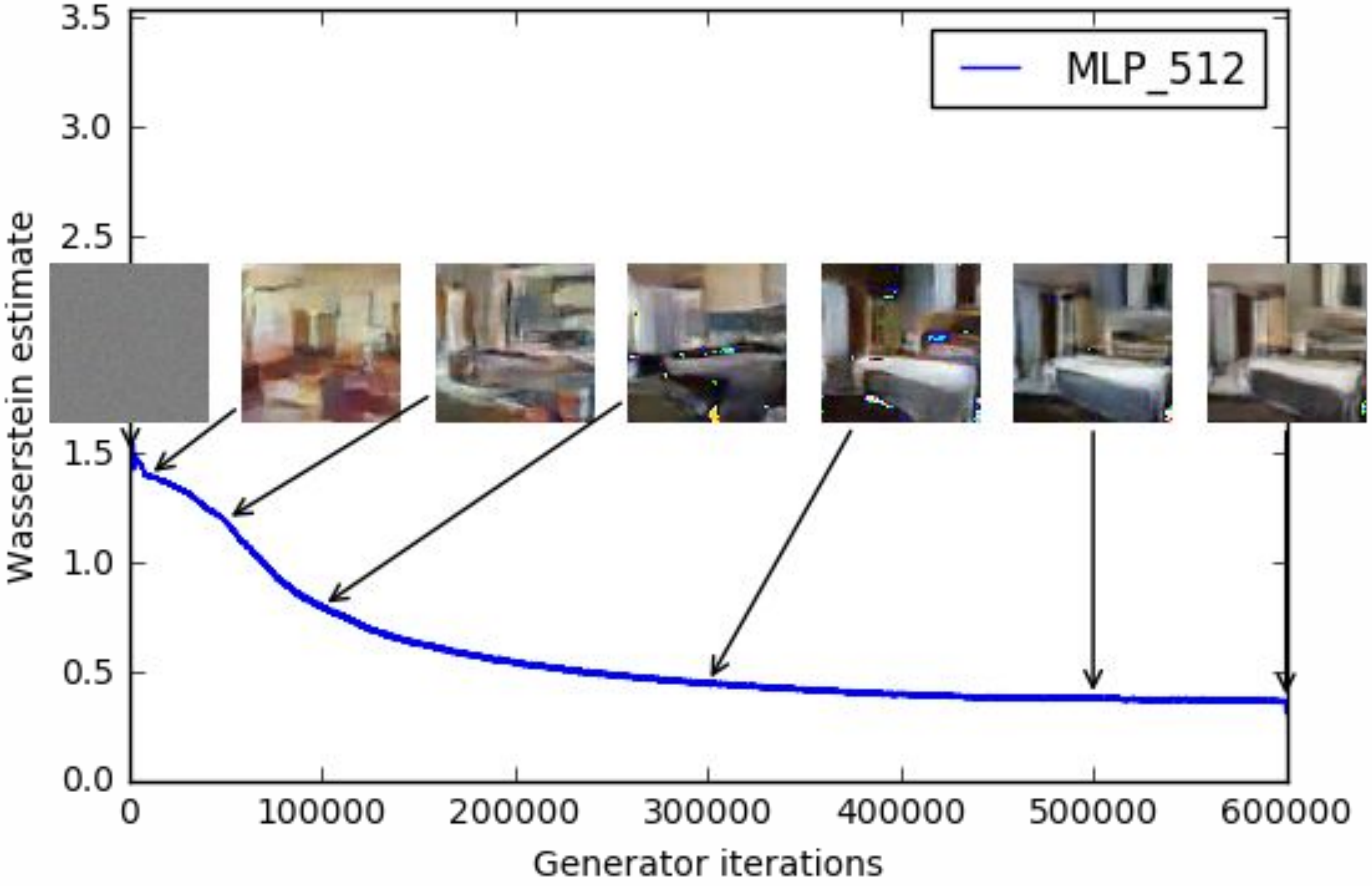
$$\frac{1}{m}\sum_{i=1}^{m}-\log\left(D\left(G\left(z^{(i)}\right)\right)\right)$$



$$\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$$

https://arxiv.org/abs/1701.07875

Wasserstein GAN. Качество зависит от лосса



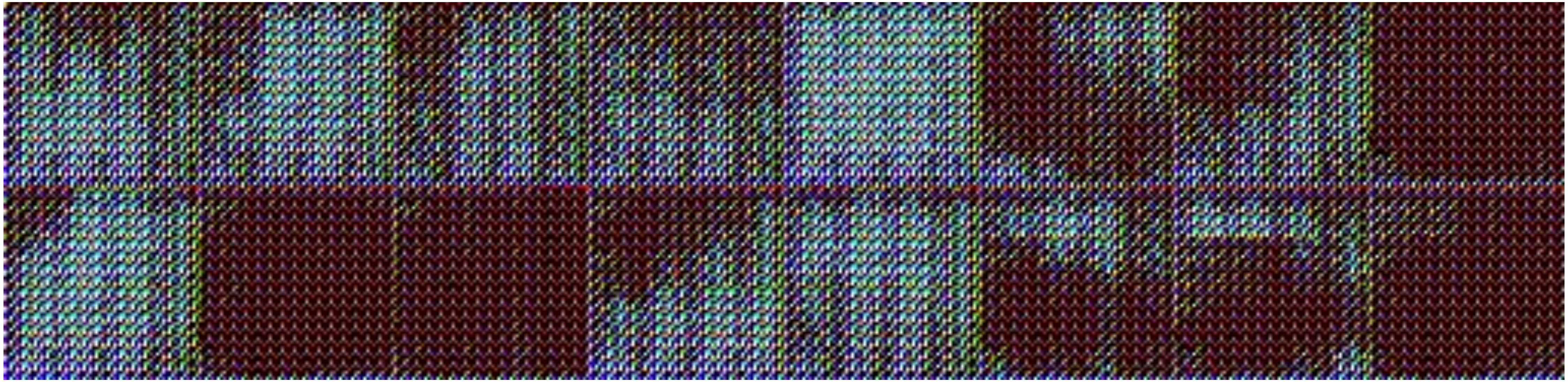
https://arxiv.org/abs/1701.07875

Wasserstein GAN. Сравнение с DCGAN

Верхняя картинка: WGAN с архитектурой DCGAN. Нижняя - DCGAN

https://arxiv.org/abs/1701.07875

Wasserstein GAN без BatchNorm



Верхняя картинка: WGAN с архитектурой DCGAN без BN. Нижняя - DCGAN без BN https://arxiv.org/abs/1701.07875

Wasserstein GAN

- Статья:
  - поднимает проблему использования JS-дивергеции в лоссе
    - предлагает замену: использование дистанции Вассерштейна
  - Считать дистанцию Вассерштейна в лоб сложно
    - авторы переформулировывают проблему
    - получили более стабильную сходимость
  - Теперь можем на каждой итерации обучения учить дискриминатор до сходимости с меньшей боязнью свалиться в mode collapse
    - значение лосса теперь коррелирует с качеством генерации
  - Дискриминатор теперь называется Критиком:
    - Критик возвращает не вероятность, а действительное число
    - Отличается от дискриминатора отсутствием сигмоиды в конце

Что мы уже прошли. Часть 1

- Оригинальная статья:
  - учим дискриминатор как бинарный классификатор
  - используем градиенты ошибки классификации для обучения дискриминатора и генератора
  - столкнулись с затуханием градиентов генератора
    - поменяли функционал потерь генератора
  - узнали, что идеальный дискриминатор пытается предсказать отношение плотностей распределений
  - столкнулись с проблемой mode collapse. В 2020 она до сих пор не решена
    - причиной считается поведение генератора, который стремится для всего латентного пространства вернуть одинаковый выход

Что мы уже прошли. Часть 2

- Оригинальная статья:
  - учим дискриминатор как бинарный классификатор
  - используем функционал ошибки классификации
    - увидели, что он может быть выведен как сумма JS-дивергенции и константы
    - увидели, что это приводит к проблемам со сходимостью и интерпретируемостью лосса

Что мы уже прошли. Часть 3

- Wasserstein GAN:
  - учим дискриминатор как бинарный классификатор
  - используем новый функционал ошибки классификации
    - дисперсия градиентов стала меньше
    - значение лосса стало интерпетируемо

На следующей лекции:

- Современные подходы к генерации изображений
  - Progressive growing of GANs
  - BigGAN
  - AdaIN слои
  - StyleGAN и StyleGAN2
- Conditional GANs
- Метрики для сравнения результатов генерации

Ссылки

- Generative Adversarial Nets. Goodfellow et al.
- NIPS 2016 Tutorial: Generative Adversarial Networks. I. Goodfellow
- Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. Radford et al.
- Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Ioffe et al.
- Unrolled Generative Adversarial Networks. Metz et al.
- https://youtu.be/T4UuL7U5asA
- https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html#kullbackleibler-and-jensenshannon-divergence
- https://vincentherrmann.github.io/blog/wasserstein/
- Progressive Growing of GANs: https://www.youtube.com/watch?v=t640zZzIRBY
- https://arxiv.org/abs/1710.10196
- https://towardsdatascience.com/explained-a-style-based-generator-architecture-for-gans-generating-and-tuning-realistic-6cb2be0f431
- https://arxiv.org/pdf/1706.08224.pdf
- Полезный и красивый семинар Саши Панчина из курса ШАДа по глубинному обучению: https://colab.research.google.com/github/yandexdataschool/Practical_DL/blob/spring20/seminar08-generative/simple_1d_gan_pytorch.ipynb