

Глубинное обучение

Генеративные сети. Генеративно-состязательные сети. Вариационные
автокодировщики

Даниил Водолазский

ВШЭ

8 сентября 2021 г.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Содержание

1 Генеративные модели

2 Генеративно-состязательные сети (GANs)

Постановка задачи

Проблемы с обучением

Свёрточные ганы

Зоопарк из ганов

3 Вариационный автокодировщик

4 Что узнали

Определение

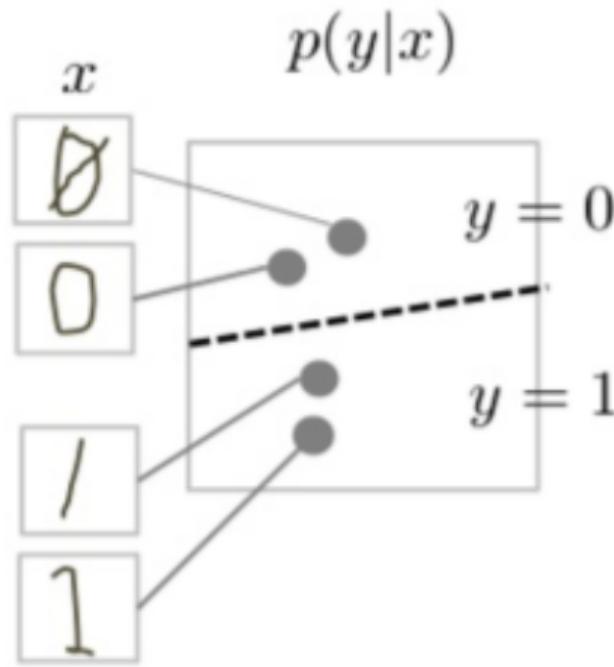
Порождающие модели (англ. **generative model**) — это класс моделей, которые обучаются совместное распределение данных $p(x, y)$.

дискриминативная модель (англ. **discriminative model**) обучает только условное распределение $p(y | x)$.

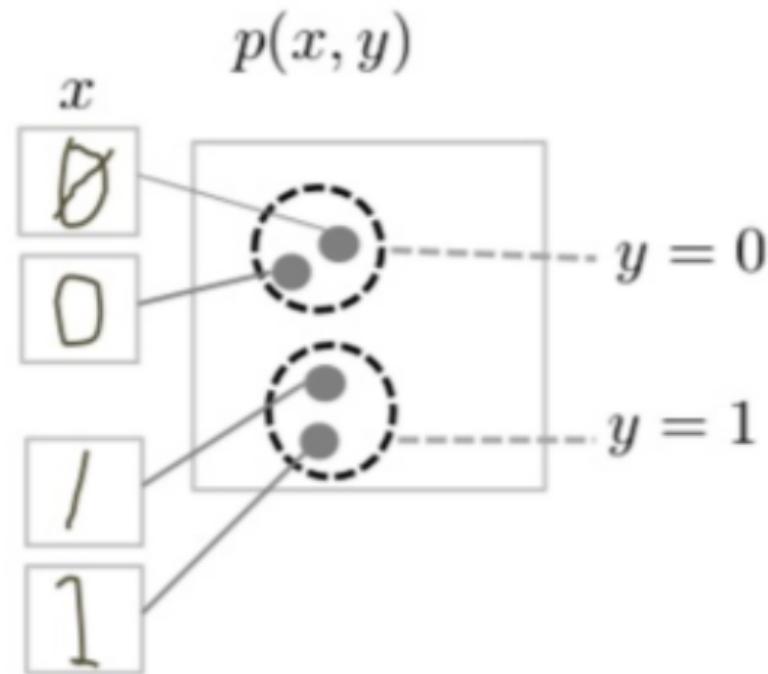
— <http://neerc.ifmo.ru/wiki/index.php?title=> —

Генеративные модели

- Discriminative Model



- Generative Model



Содержание

1 Генеративные модели

2 Генеративно-состязательные сети (GANs)

Постановка задачи

Проблемы с обучением

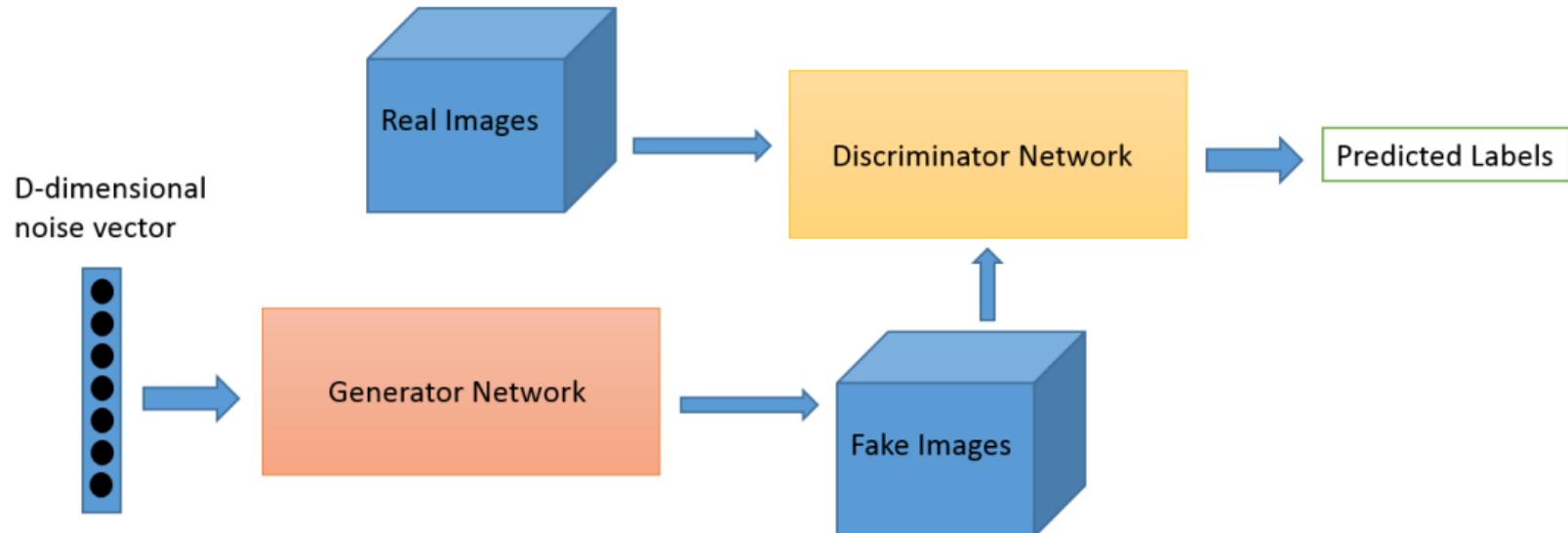
Свёрточные ганы

Зоопарк из ганов

3 Вариационный автокодировщик

4 Что узнали

Генеративно-состязательные сети (GANs)



<https://arxiv.org/abs/1406.2661>

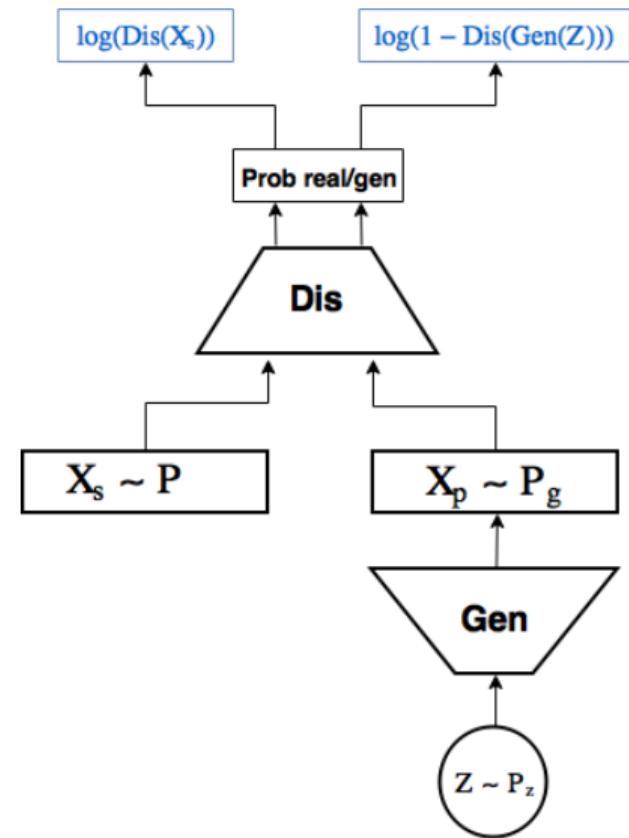
Генеративно-состязательные сети (GANs)

- Две нейронных сетки.
- **Генератор** сэмплирует из какого-то распределения вектора и пытается переработать их в картинки.
- **Дискриминатор** получает на вход сгенерированные картинки и настоящие и пытается отличить реальные данные от подделки.

Генеративно-состязательные сети (GANs): пример из жизни



Схема



Обучение модели

- Обучение происходит поэтапно.
- Сначала несколько шагов дискриминатор учится различать правду от лжи, минимизируется

$$L_D = -y_i \cdot \log \hat{p}_i - (1 - y_i) \cdot \log(1 - \hat{p}_i) \rightarrow \min_D$$

Обучение модели

- Обучение происходит поэтапно.
- Сначала несколько шагов дискриминатор учится различать правду от лжи, минимизируется

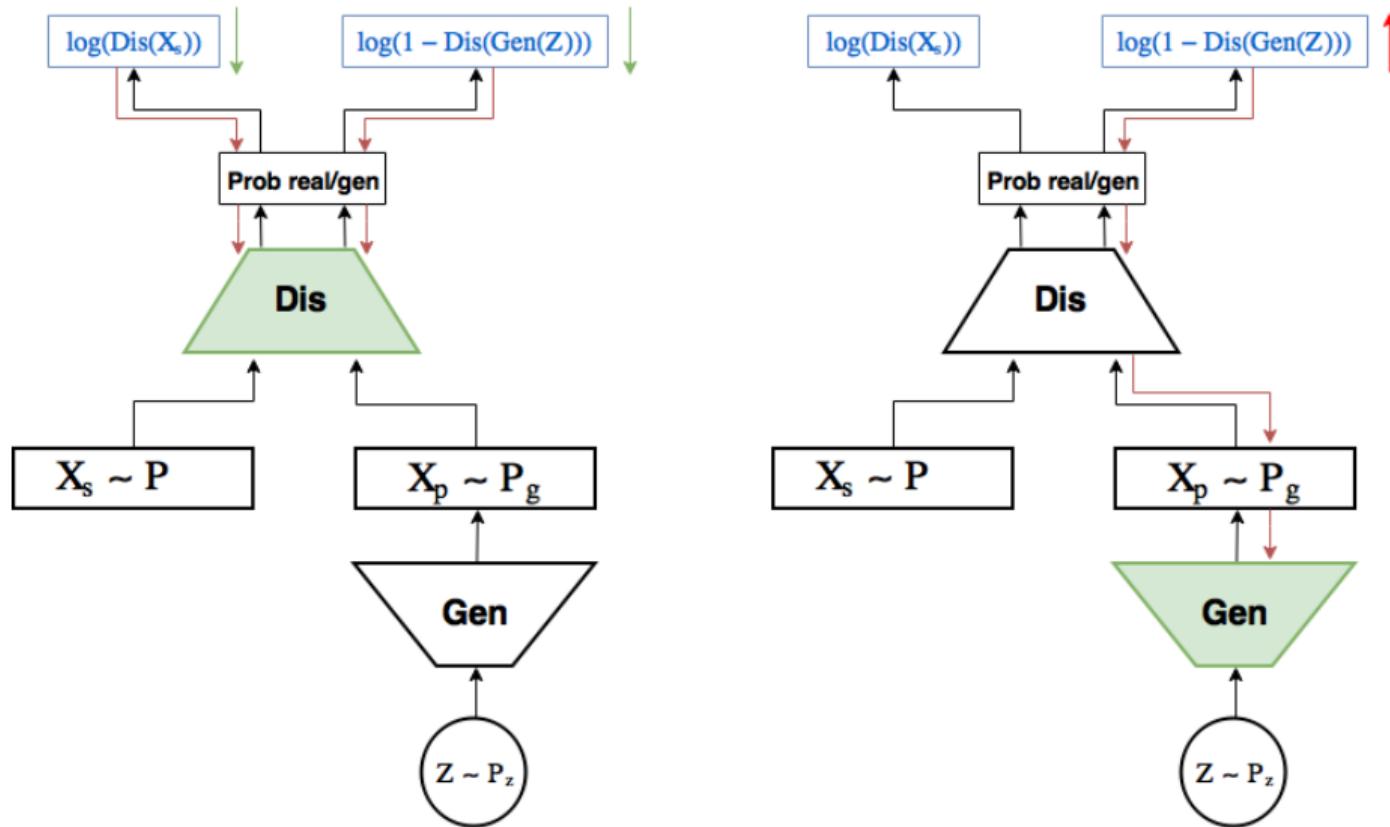
$$L_D = -y_i \cdot \log \hat{p}_i - (1 - y_i) \cdot \log(1 - \hat{p}_i) \rightarrow \min_D$$

- После происходит один шаг обучения генератора, он пытается заставить дискриминатор ошибаться и максимизирует значение ошибки на ложных картинках:

$$L_G = -\log(1 - \hat{p}_i) \rightarrow \max_G .$$

- Цель обучения — достичь равновесия между генератором и дискриминатором.

Обучение сеток



Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k , is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

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

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[\log D \left(\mathbf{x}^{(i)} \right) + \log \left(1 - D \left(G \left(\mathbf{z}^{(i)} \right) \right) \right) \right].$$

end for

- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Update the generator by descending its stochastic gradient:

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

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Задача

- Если совсем-совсем заниматься формализмом, мы решаем задачу:

$$L_{GAN}(G, D) = \min_G \max_D \mathbb{E}_x[\log D(x)] + \mathbb{E}_z[\log(1 - D(G(z)))].$$

Задача

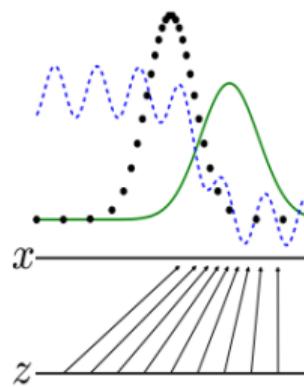
- Если совсем-совсем заниматься формализмом, мы решаем задачу:

$$L_{GAN}(G, D) = \min_G \max_D \mathbb{E}_x[\log D(x)] + \mathbb{E}_z[\log(1 - D(G(z)))].$$

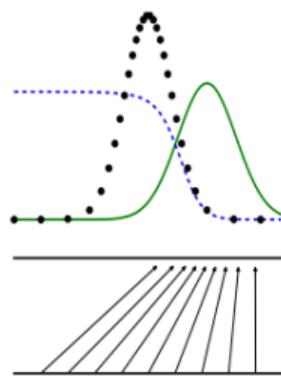
- При заданном генераторе оптимальный дискриминатор прогнозирует

$$D^*(x) = \frac{P(x)}{P_g(x) + P(x)}.$$

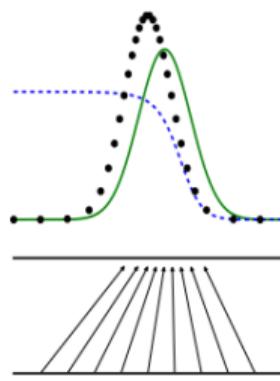
Оптимальное решение задачи



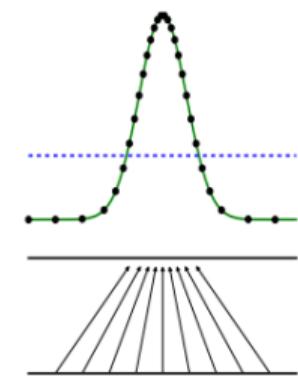
(a)



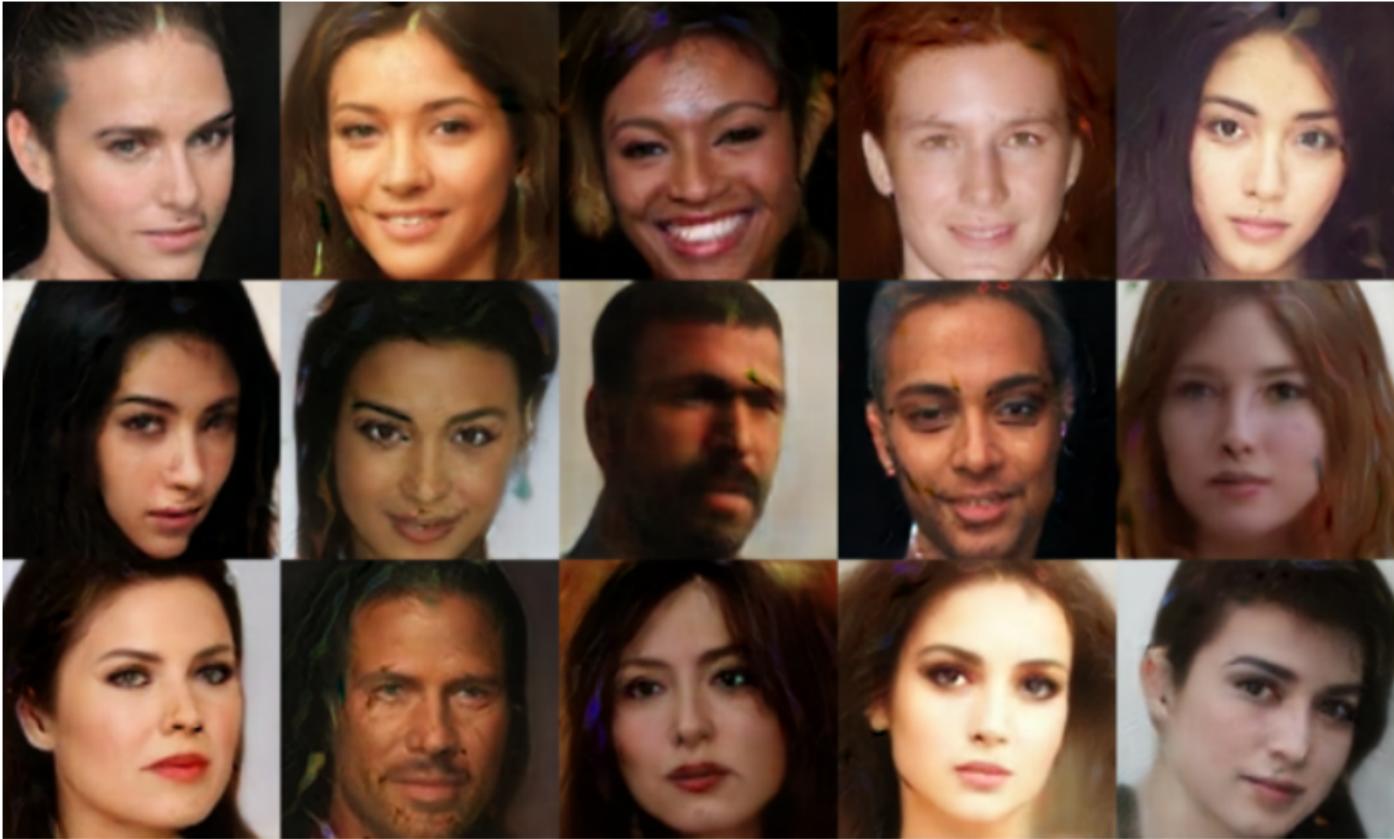
(b)



(c)



(d)



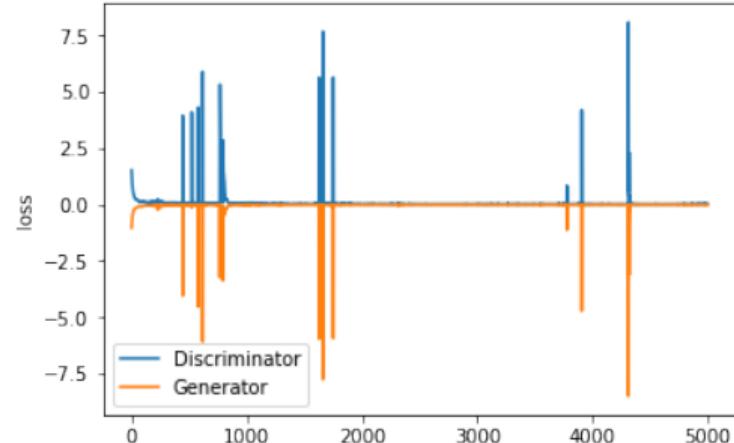
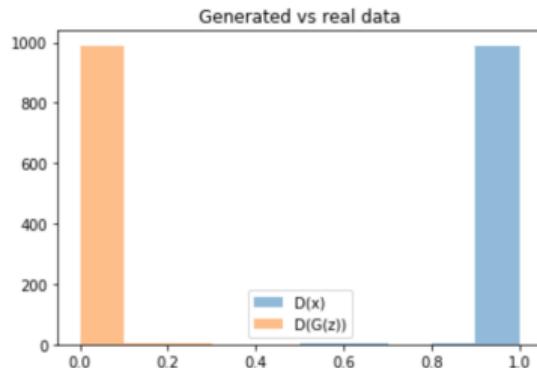
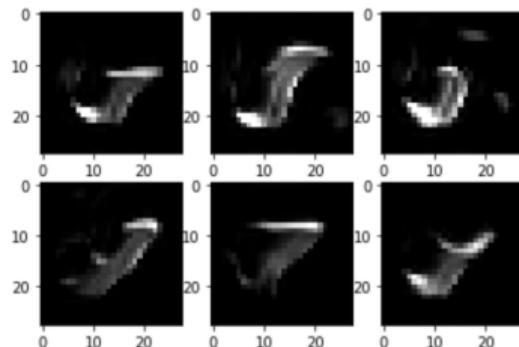
<https://thispersondoesnotexist.com>

<https://arxiv.org/pdf/1812.04948.pdf>

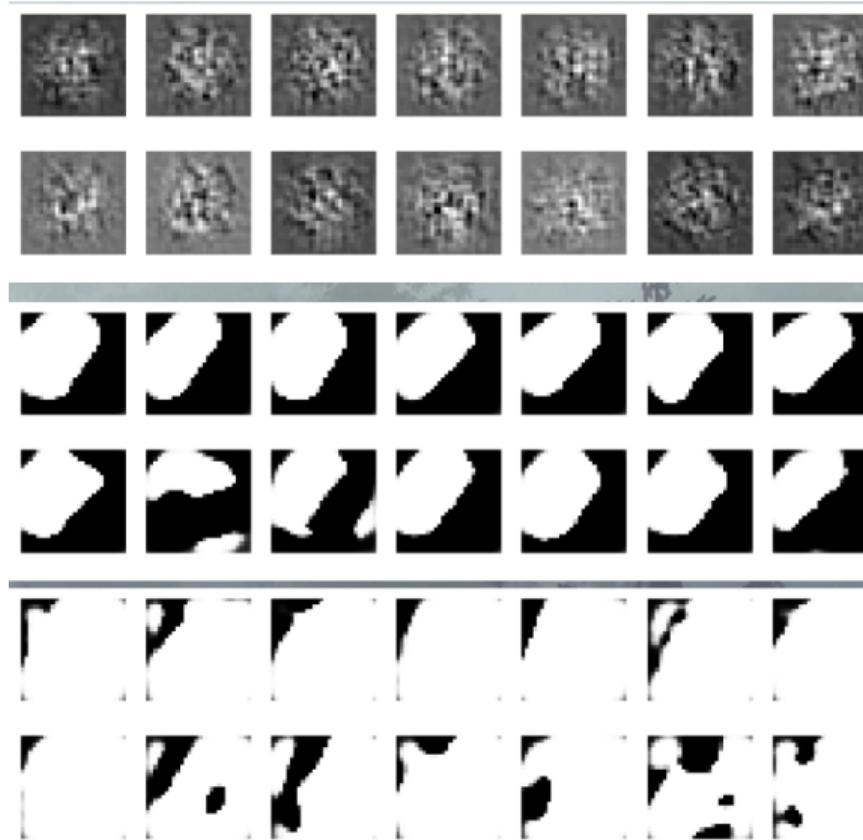


<https://thiscatdoesnotexist.com>

Кейс 1



Кейс 2





Soumith Chintala @soumithchintala

Following

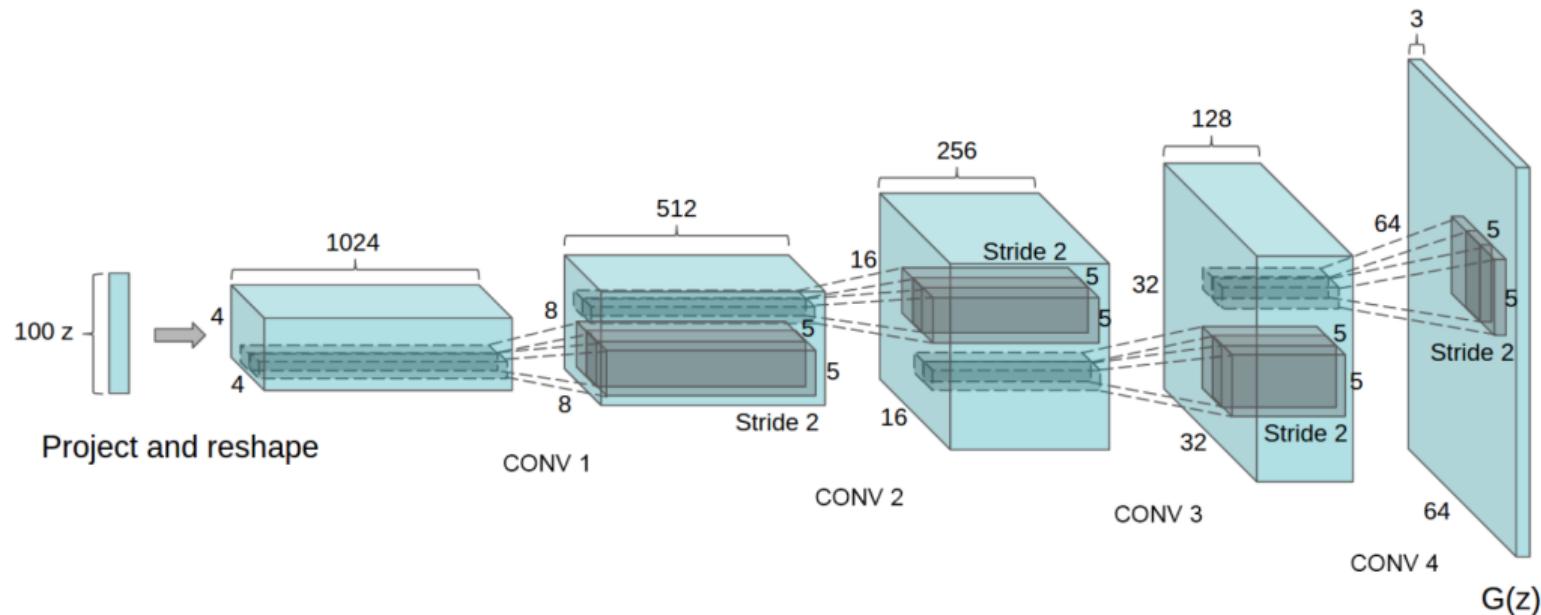
How to Train a GAN? I'm as clueless as you.
Best I'll do is summarize the tricks, hacks
and untold dark rituals that we've all used so
far!



10:08 AM - 3 Dec 2016



Deep Convolutional GAN



<https://arxiv.org/pdf/1511.06434.pdf>

Советы для хороших хозяшек [1]

- Нормализуйте входы на $[-1, 1]$ и на последний слой генератора добавляйте Tanh.
- Сэмплируйте из нормального распределения, а не из равномерного.
- Используйте разные BatchNorm в обеих сетках.
- Можно попробовать использовать разный BatchNorm для фейковых и реальных данных.
- Брать полносвязные слои — не очень хорошая мысль, свёрточные — хорошая.

Советы для хороших хозяшек [2]

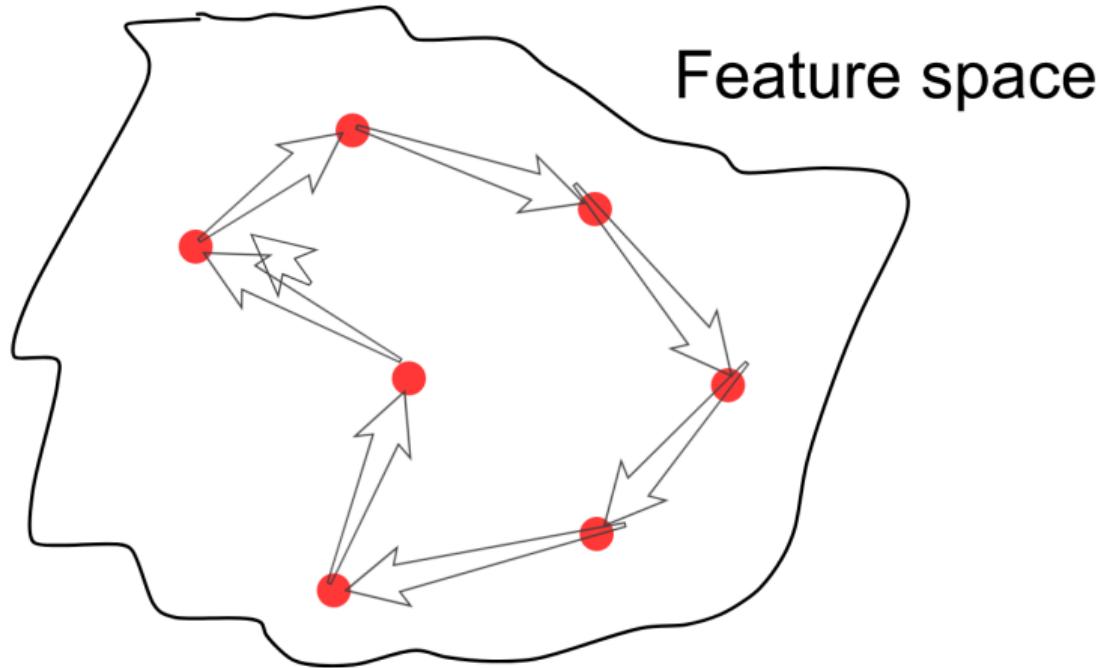
- Лучше избегать пулинга и делать stride внутри свёрток.
- Если градиенты разряжены, GAN ведёт себя стабильнее, ReLU и её модификации — хорошая идея.
- Зашумляйте реальные данные, чтобы не возникало переобучения.
- Если в самом начале дискриминатор сильно вырвется вперёд, обучение может заглохнуть.

Ещё советы: <https://github.com/soumith/ganhacks>

Примеры драконов



Интерполяция



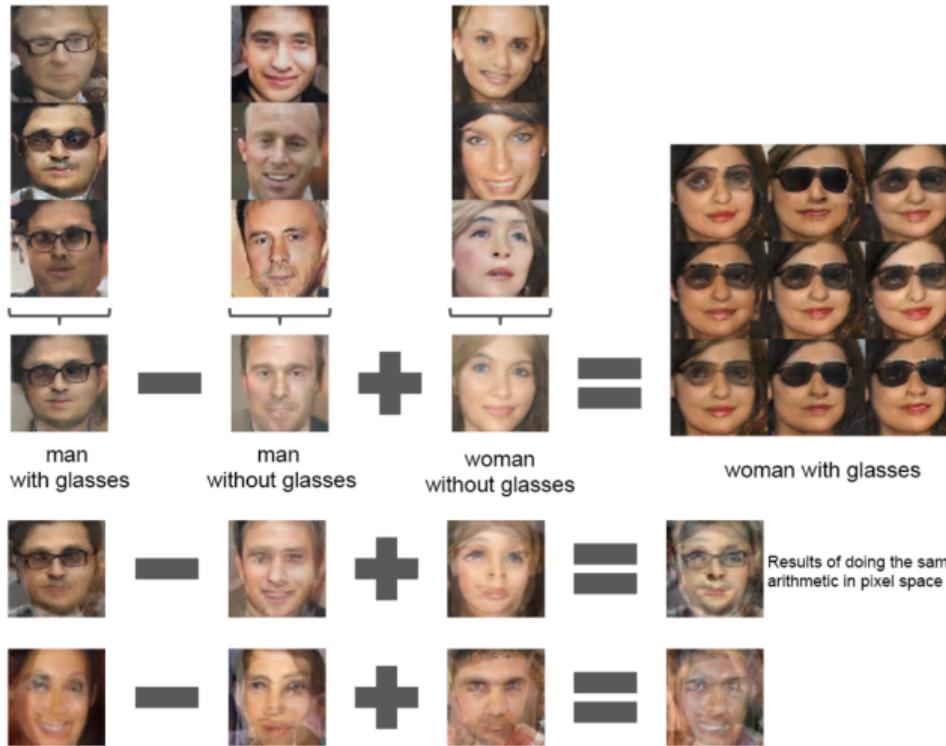
Интерполяция



Интерполяция



Арифметика



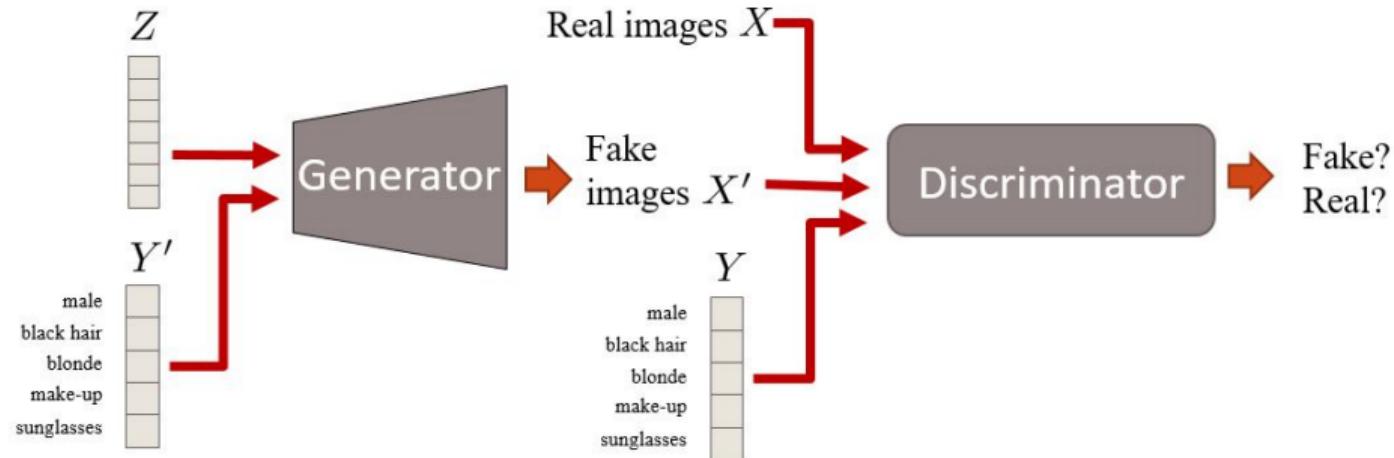
The GAN Zoo



Every week, new GAN papers are coming out and it's hard to keep track of them all, not to mention the incredibly creative ways in which researchers are naming these GANs! So, here's a list of what started as a fun activity compiling all named GANs!

<https://github.com/hindupuravinash/the-gan-zoo>

Conditional GAN



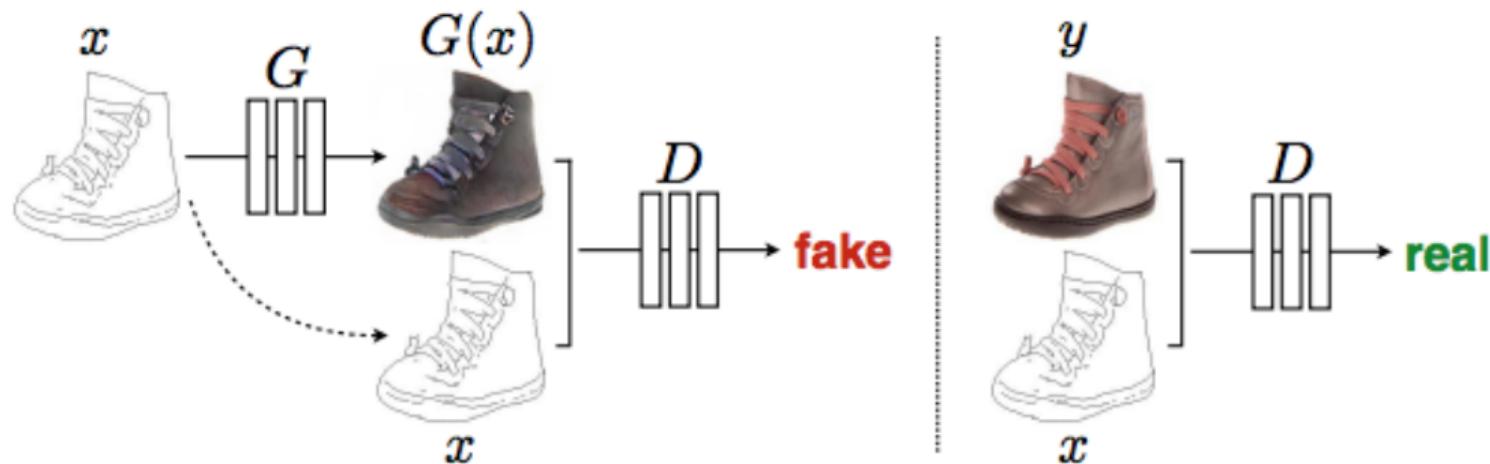
<https://arxiv.org/pdf/1411.1784.pdf>

Conditional GAN

- Использует информацию из меток, если они есть.
- Это позволяет генерировать не рандомные объекты, а конкретные.
- В качестве меток y можно пытаться использовать и разную другую информацию.
- Например, в случае лиц можно использовать цвет волос, наличие очков, эмоции на лице и т. п.
- Можно даже завести несколько разных меток.
-

$$L_{cGAN}(G, D) = \mathbb{E}_{y,x}[\log D(x, y)] + \mathbb{E}_{y,z}[\log(1 - D(y, G(z, y)))].$$

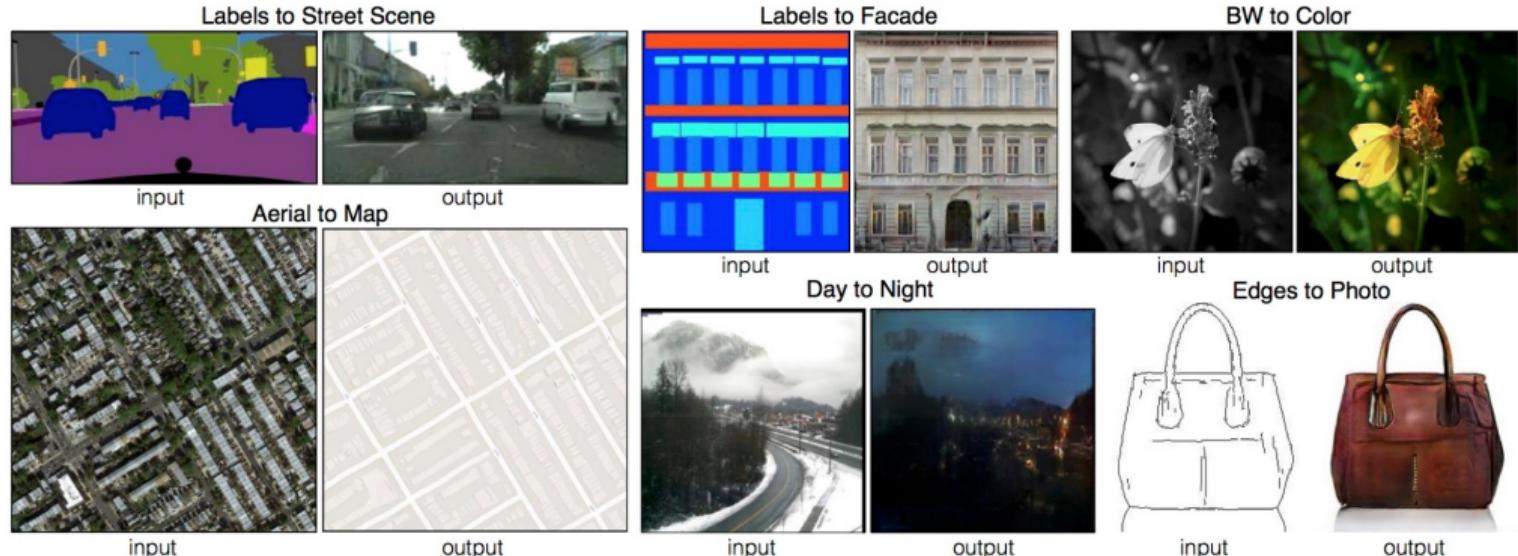
Pix2Pix GAN



<https://arxiv.org/pdf/1808.06601.pdf>

<https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>

Pix2Pix GAN

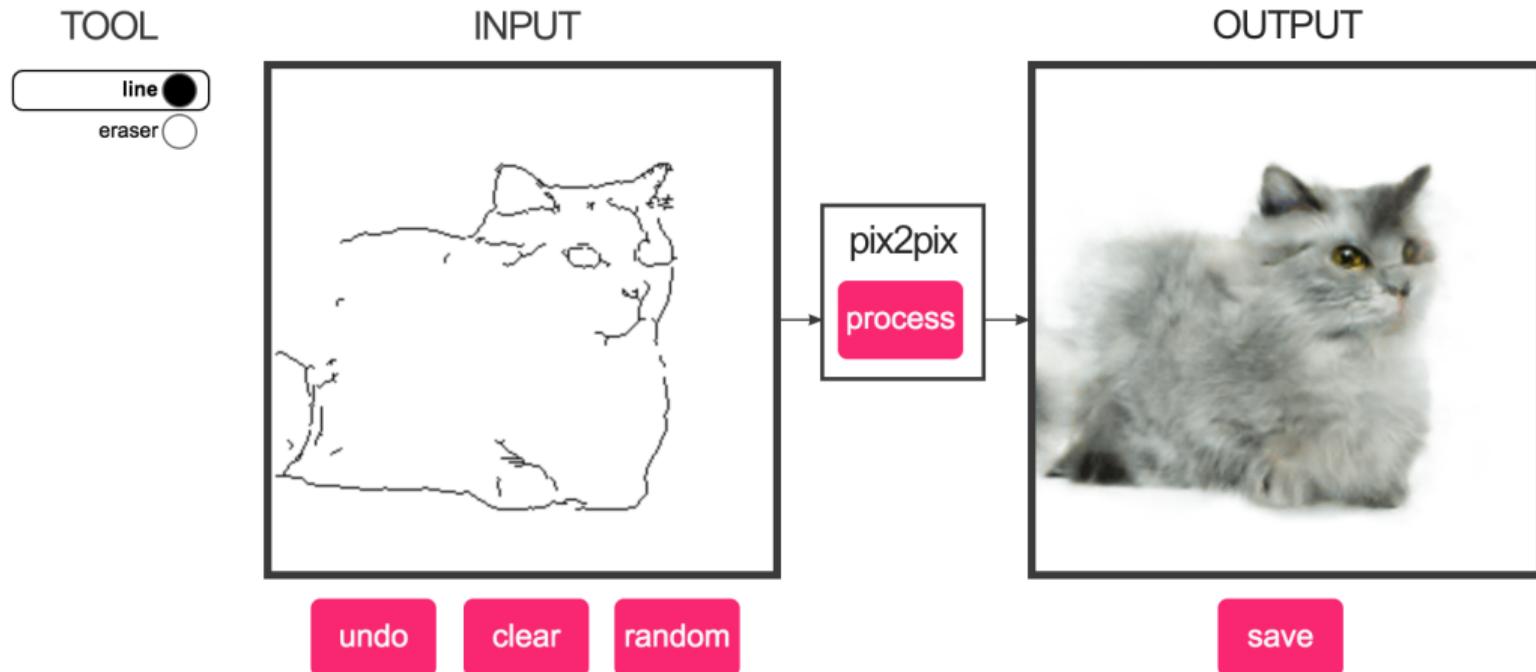


Example results on several image-to-image translation problems. In each case we use the same architecture and objective, simply training on different data.

<https://phillipi.github.io/pix2pix/>

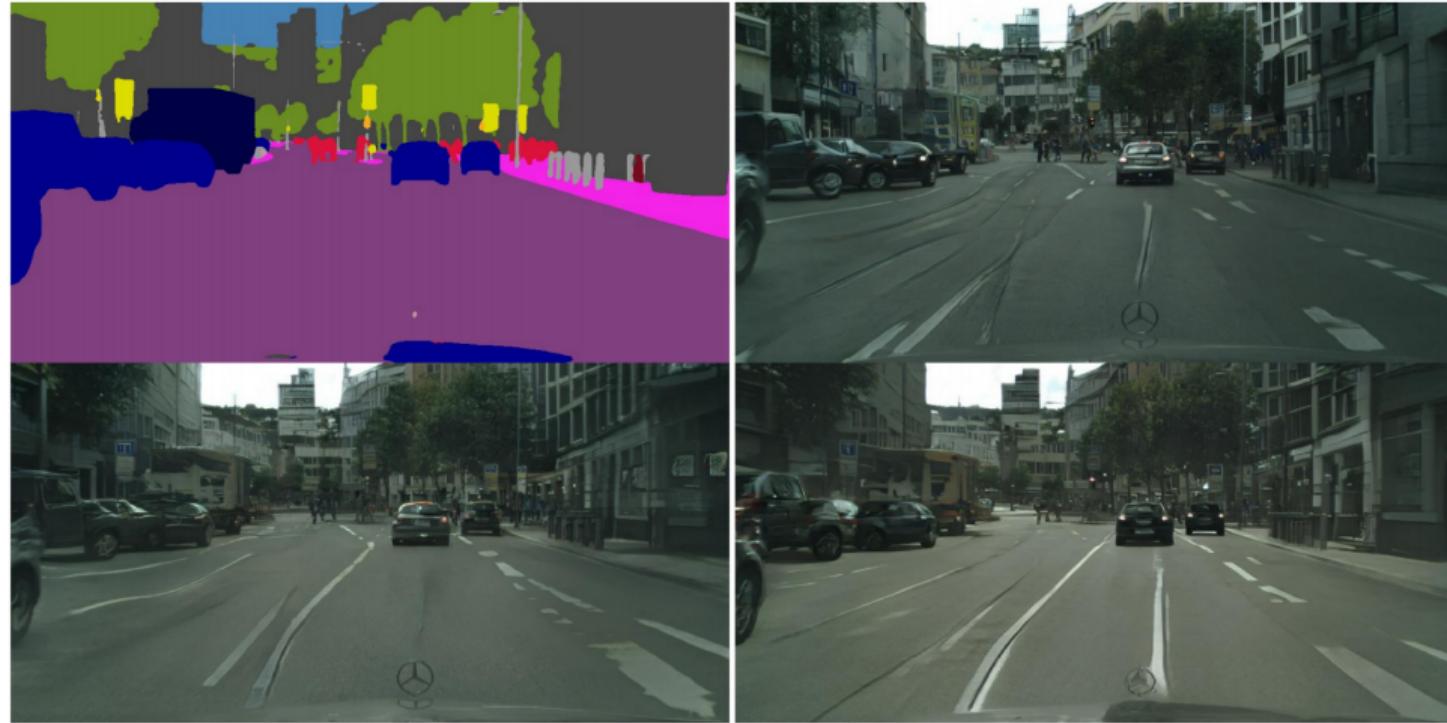
<https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>

Pix2Pix GAN



<https://affinelayer.com/pixsrv/>

Pix2Pix GAN



CycleGAN

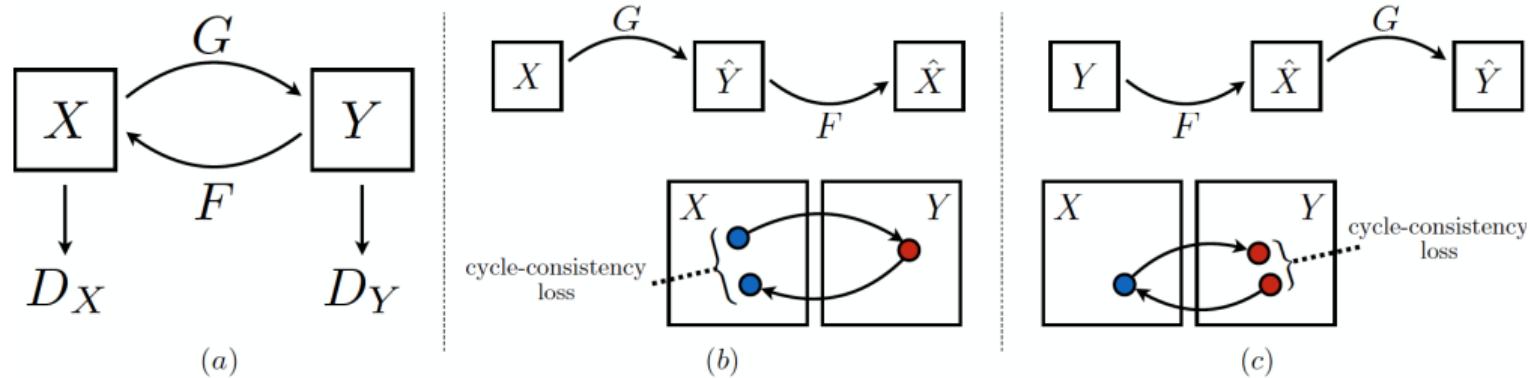


Figure 3: (a) Our model contains two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and associated adversarial discriminators D_Y and D_X . D_Y encourages G to translate X into outputs indistinguishable from domain Y , and vice versa for D_X , F , and X . To further regularize the mappings, we introduce two “cycle consistency losses” that capture the intuition that if we translate from one domain to the other and back again we should arrive where we started: (b) forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$

<https://arxiv.org/abs/1703.10593>

<https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>

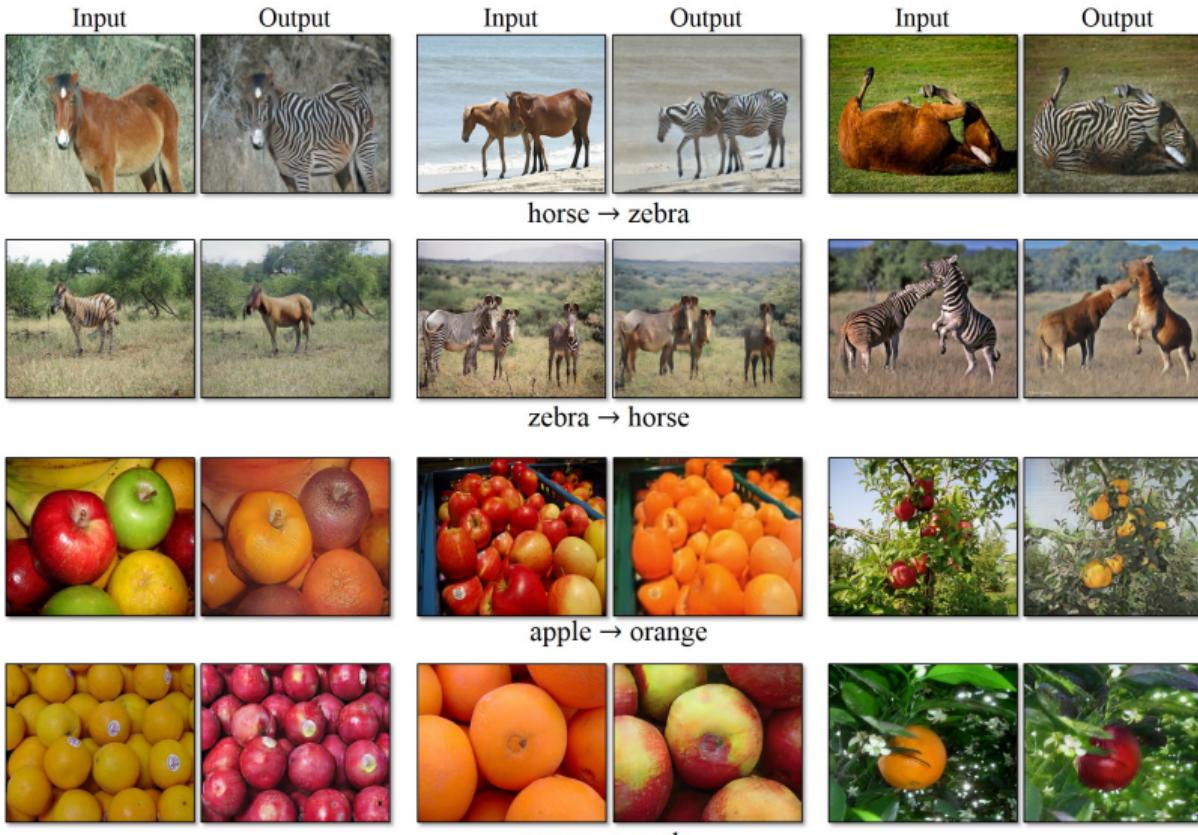
Целевая функция:

$$L_{CGAN}(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F),$$

где

$$L_{cyc}(G, F) = \mathbb{E}_x \|F(G(x)) - x\|_1 + \mathbb{E}_y \|G(F(y)) - y\|_1.$$

CycleGAN



- Отличный обзор разных статеек про GAN на хабре, от оригинальной статьи до продвинутых идей.
- Хорошая серия статей про ганы и автокодировщики.
- Генератор хороров от MIT.

Содержание

1 Генеративные модели

2 Генеративно-состязательные сети (GANs)

Постановка задачи

Проблемы с обучением

Свёрточные ганы

Зоопарк из ганов

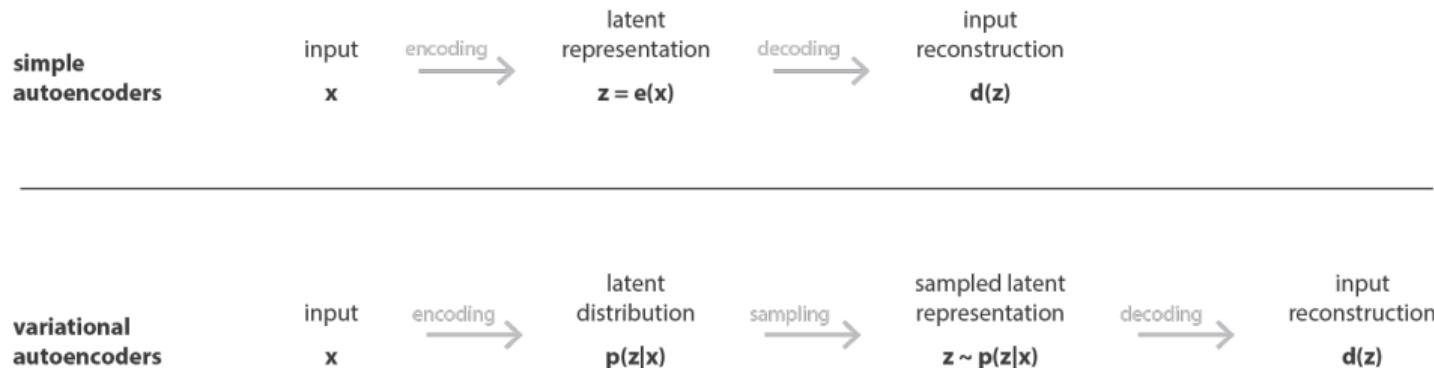
3 Вариационный автокодировщик

4 Что узнали

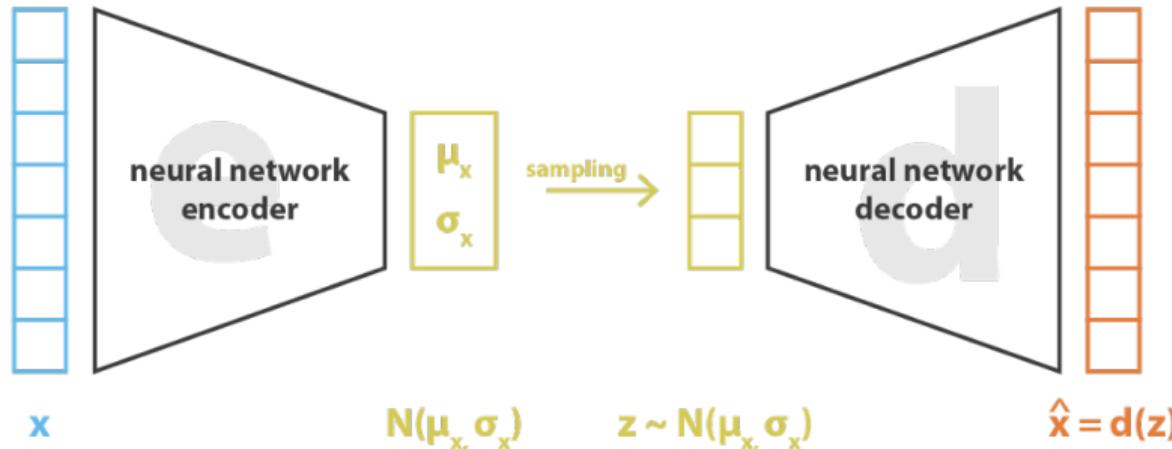
Вариационный автокодировщик

- Придумано в 2014 году.
- Продолжение идей разреженного автокодировщика, добавляем новых хотелок в наше признаковое пространство.
- По своей сути является генеративной моделью. Мы хотим найти удобное нам пространство, откуда удобно сэмплить точки для создания нового.
- Вместо того чтобы получать скрытое представление для объектов, энкодер учит распределение входных данных (чаще всего многомерное нормальное).
- Декодер, в свою очередь, учится восстанавливать данные из этого распределения.

Вариационный автокодировщик

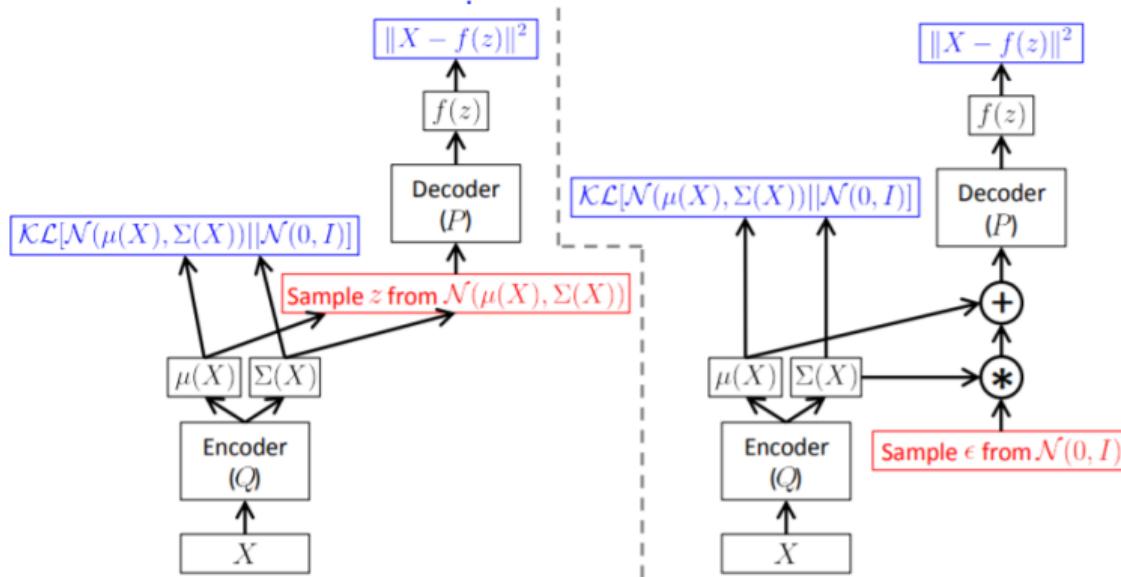


Вариационный автокодировщик



$$\text{loss} = \|x - \hat{x}\|^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)] = \|x - d(z)\|^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)]$$

Вариационный автокодировщик



Содержание

1 Генеративные модели

2 Генеративно-состязательные сети (GANs)

Постановка задачи

Проблемы с обучением

Свёрточные ганы

Зоопарк из ганов

3 Вариационный автокодировщик

4 Что узнали

Что узнали

- В чём отличие генеративных моделей от дискриминативных.
- Что такое генеративно-состязательные сети.
- Какие бывают разновидности ганов.
- Особенности вариационных автокодировщиков.