

# Нейросетевой синтез текстур с трендами

Будалян Я. С.

Научный руководитель к.т.н., доц. Грачев Е. А.

2017 г.

## Введение

Задача состоит в синтезе изображений среды, которые будут содержать в себе тренд, т.е. изменение некоторой статистической характеристики. Такими трендами могут быть, например, изменение интенсивности появления частиц среды вдоль изображения, или изменение пористости среды.

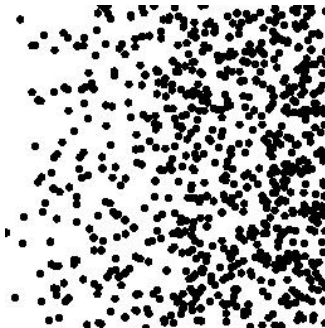


Рис.: Пример текстуры с трендом интенсивности частиц

## Математическая постановка

С математической точки зрения, задача сводится к синтезу случайного изображения  $X'$  (и построению соответствующей процедуры синтеза), имеющего распределение, близкое к желаемому:

$$P_{X'} \approx P_X,$$

где  $P_X$  - распределение изображений с трендами, удовлетворяющих следующим ограничениям (для упрощения задачи):

- Это монохромные изображения  $256 \times 256$  пикселей
- Изменяющимся свойством является интенсивность появления частиц  $\lambda$
- Тренд является линейным и направлен вдоль оси  $x$ :  $\lambda = \lambda_0 + kx$

## Подходы к решению задачи

Существует несколько подходов к решению задач подобного рода:

- 'Классический' статистический подход
- Первый нейросетевой подход
- Генеративные состязательные сети (GAN)

## 'Классический' статистический подход

- Вводится параметризованное семейство распределений вероятности  $P_{\theta}(x)$
- Параметры  $\theta$  находятся из обучающей выборки:

$$\mathcal{L}_{\theta}(D) = \prod_{x \in D} P_{\theta}(x)$$

$$\theta^* = \arg \max_{\theta} \mathcal{L}_{\theta}(D)$$

- Сгенерировать семпл из  $P_{\theta^*}$

Этот подход приводит к проблемам:

- Пространство параметров  $\theta$  может быть огромной размерности
- Или же известной параметрической модели распределения может вообще не существовать

Простой пример - генерирование человеческих лиц, похожих на реальные:  
параметрической модели для такой задачи не существует.

## Первый нейросетевой подход

- Вводится параметризованное семейство распределений вероятности  $P_{\theta}(x)$ 
  - ▶ Вводятся скрытые переменные  $V$  и функция(нейросеть) для получения  $x$  из  $V$  (фактически, классификация, развернутая в другую сторону)
- Определяются параметры распределения (т.е. обучение нейросети)
- Генерируются семплы из  $P_{\theta^*}$

Этот подход возможен, однако на практике трудноосуществим.

Изначальная задача: найти процедуру генерирования  $X'$  так, чтобы  $P_{X'} \approx P_X$ .  
 Переформулируем:

$$\rho(P_{X'}, P_X) \longrightarrow \min_{P_{X'}}$$

- Введем некоторые скрытые переменные с фиксированным распределением, например

$$V \sim U^n[-1, 1]$$

- и параметризованную процедуру генерации:

$$X' = g_\theta(V)$$

Переформулируем:

$$\rho(P_{X'}, P_X) \longrightarrow \min_{P_{X'}}$$

$$\rho(g_\theta(V), P_X) \longrightarrow \min_{g_\theta(V)}$$

$$\rho(g_\theta(V), P_X) \longrightarrow \min_{\theta}$$

Остается вопрос: что использовать в качестве метрики похожести двух распределений  $\rho$ , где одно из распределений задано обучающей выборкой.

- В качестве метрики статистической похожести можно использовать loss-функцию обученного классификатора:

$$\rho(P_{X'}, P_X) \longrightarrow \min \Leftrightarrow \mathcal{L} \longrightarrow \max,$$

где  $\mathcal{L}$  - функция потерь обученного классификатора.



- Введем две нейросети:
  - $d_\zeta(x)$  - классификатор для измерения расстояния, **дискриминатор**
  - $g_\theta(x)$  - сеть, трансформирующая  $V$  в  $x$ , **генератор**
- Введем loss-функцию дискриминатора(например, кросс-энтропия):

$$\begin{aligned}
 L(X, X') &= \frac{1}{2} \mathbb{E}_{x \sim X} l(d_\zeta(x), 1) + \frac{1}{2} \mathbb{E}_{x' \sim X'} l(d_\zeta(x'), 0) = \\
 &= -\frac{1}{2} (\mathbb{E}_{x \sim X} \log d_\zeta(x) + \mathbb{E}_{x' \sim X'} \log(1 - d_\zeta(x'))) = \\
 &= -\frac{1}{2} (\mathbb{E}_{x \sim X} \log d_\zeta(x) + \mathbb{E}_{v \sim V} \log(1 - d_\zeta(g_\theta(v)))) = \\
 &= L(\zeta, \theta)
 \end{aligned}$$

- Loss-функция обученного классификатора:

$$L^*(\theta) = \min_{\zeta} L(\zeta, \theta)$$

- Соответственно,

$$\min_{\zeta} L(\zeta, \theta) \longrightarrow \max_{\theta}$$

$$\theta^* = \arg \max_{\theta} \left[ \min_{\zeta} L(\zeta, \theta) \right]$$

- Определим оптимальный дискриминатор:

$$d_{\theta}^* = d_{\zeta^*(\theta)}$$

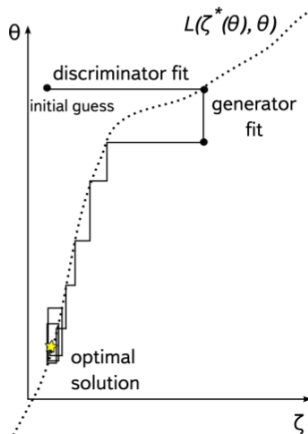
$$\zeta^*(\theta) = \arg \min_{\zeta} L(\zeta, \theta)$$

# Обучение GAN

В итоге, процесс обучения принимает следующий вид:

- Обучаем дискриминатор при 'замороженном' генераторе
- Обучаем генератор при 'замороженном' дискриминаторе
- Повторяем много раз

Схематически процесс обучения изображен на иллюстрации справа



# Критерий качества

## Результаты, графики

## Графики-2

# Выводы