

Нейросетевой синтез текстур с трендами

Будалян Я. С.

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

2017 г.

Введение

Цель работы состоит в построении процедуры синтеза изображений среды, которые будут содержать в себе тренд. Под текстурой с трендом понимается изображение, в котором есть изменение некоторой статистической характеристики вдоль одного из направлений. Такими характеристиками, например, могут быть изменение интенсивности появления частиц среды или изменение пористости среды.

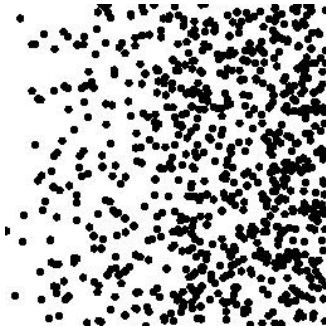


Рис.: Пример текстуры с трендом интенсивности частиц.

Рассмотрим многомерное пространство X , содержащее множество всех изображений x : $X = \{x\}$. Обучающая выборка изображений с трендами $D = \{x_i\}$ задает в этом пространстве вероятностное распределение P_X , такое, что точки, соответствующие изображениям из выборки, имеют высокую вероятность, а остальные - низкую. Тогда с математической точки зрения задача синтеза текстуры с трендом сводится к синтезу случайного изображения x' , принадлежащего распределению, близкому к задаваемому обучающей выборкой:

$$P_{X'} \approx P_X, \quad x' \sim X'$$

Математическая постановка

Для упрощения задачи, сузим множество изображений с трендами до множества изображений, удовлетворяющих следующим ограничениям:

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

Существующие подходы к синтезу

Есть несколько подходов к решению задач подобного рода:

- 'Классический' статистический подход
- Базовый нейросетевой подход
- Генеративные состязательные сети (Generative Adversarial Networks - GAN)

'Классический' статистический подход

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

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

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

- Генерируется объект(изображение) из P_{θ^*}

Этот подход приводит к проблемам:

- Пространство параметров θ может быть огромной размерности
- Известной параметрической модели распределения может вообще не существовать

Простой пример - синтез человеческих лиц: с помощью классического подхода эта задача не была решена с хорошим качеством.

Базовый 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)
- Генерируется объект(изображение) из P_{θ^*}

Этот подход возможен, однако на практике трудноосуществим или не приводит к хорошему качеству генерации.

Архитектура GAN была придумана в 2014 году специально для решения задачи генерации объектов из сложных распределений.

- Переформулируем изначальную задачу нахождения такой процедуры генерирования X' , чтобы $P_{X'} \approx P_X$:

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

- Введем параметризованную процедуру генерации:

$$X' = g_{\theta}(\cdot)$$

- Переформулируем:

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

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

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

Возникает вопрос: что использовать в качестве метрики похожести двух распределений ρ , где одно из распределений задано обучающей выборкой.

- В качестве такой метрики можно использовать функцию потерь обученного классификатора, потому что естественно предположить, что чем чаще ошибается обученный классификатор, тем больше одно распределение похоже на другое:

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

где L - функция потерь обученного классификатора.

- Введем две нейросети:

- ▶ $d_{\zeta}(x)$ - классификатор для измерения расстояния, **дискриминатор**
- ▶ $g_{\theta}(x)$ - сеть, трансформирующая шум в X' , **генератор**

Суть использования двух сетей состоит в том, что они обучаются совместно, конкурируя друг с другом: генератор пытается имитировать целевое распределение, а дискриминатор пытается классифицировать поступающие от генератора и из обучающей выборки изображения на 2 класса: реальные (из изначального распределения P_X) и ложные (из $P_{X'}$, т.е. произведенные генератором).

- Функция потерь 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

Таким образом, процесс обучения сети типа GAN принимает следующий вид:

- Обучаем дискриминатор при фиксированном генераторе
- Обучаем генератор при фиксированном дискриминаторе
- Повторяем до сходимости параметров обеих моделей

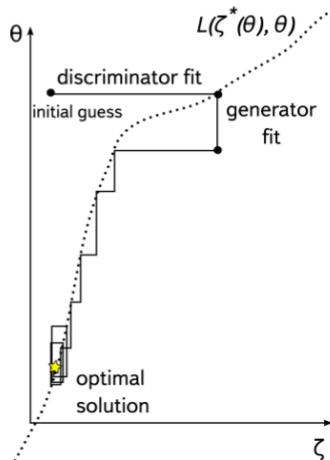


Рис.: Схематическое изображение процесса обучения GAN.

pix2pix GAN

Для решения задачи было попробовано применить модификацию GAN-сети под названием "pix2pix GAN". Ее отличие от схемы GAN, введенной ранее, состоит в том, что вместо шума на вход генератору приходят другие изображения, на которых он основывается при синтезе.

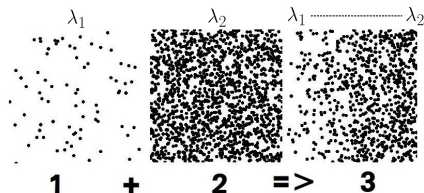
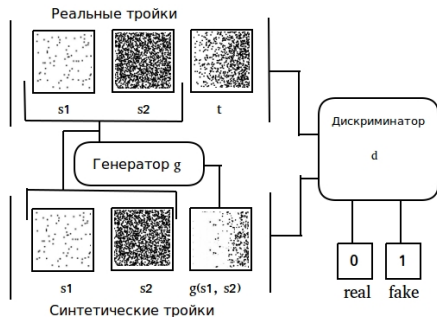


Рис.: Вход и желаемый выход нейросети-генератора.

Рис.: Схематическое устройство сети pix2pix GAN.

Критерий качества

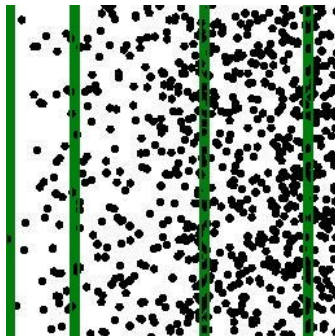
После обучения генератора, необходимо проверить, что сгенерированные им изображения действительно имеют искомые характеристики. Для этого нужно ввести специальную метрику, которая будет учитывать наличие в изображении тренда интенсивности частиц. Было решено использовать среднюю плотность черных пикселей в некотором окне, и проходить этим окном по изображению:

$$\xi_k = \frac{1}{Hw} \sum_{i=k}^{k+w} \sum_{j=0}^H \left| \frac{x(i,j) - 255}{255} \right|,$$

$$k = \overline{1, W - w + 1}$$

Построив график $\xi(k)$ можно увидеть, как меняется плотность пикселей и прослеживается ли тренд. В качестве метрики можно взять среднеквадратичную ошибку:

$$\xi = \frac{1}{W - w} \sum_{k=1}^{W-w+1} (\xi_k - \xi_{0k})^2,$$



где ξ_{0k} - ξ_k , усредненное по примерам из обучающей выборки.

Результаты

Было проведено обучение нейросети описанной архитектуры при различных гиперпараметрах. Обучающей выборкой был массив из 3500 троек изображений.

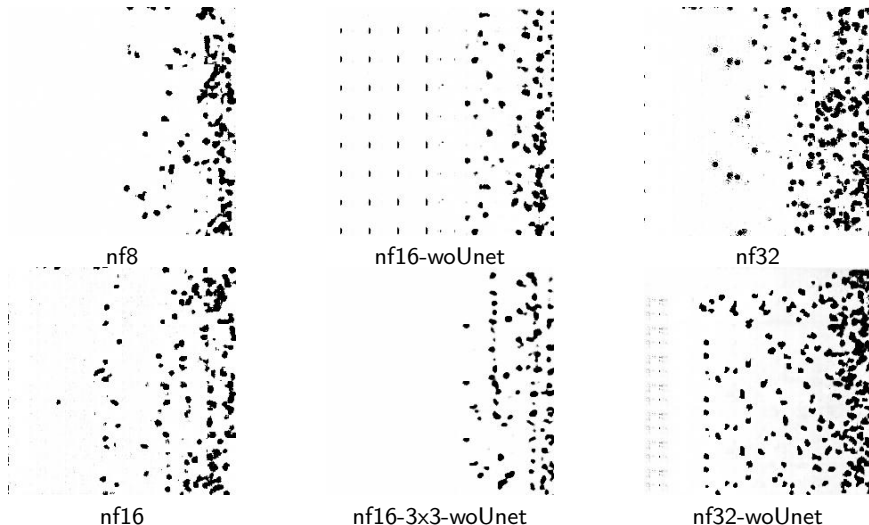
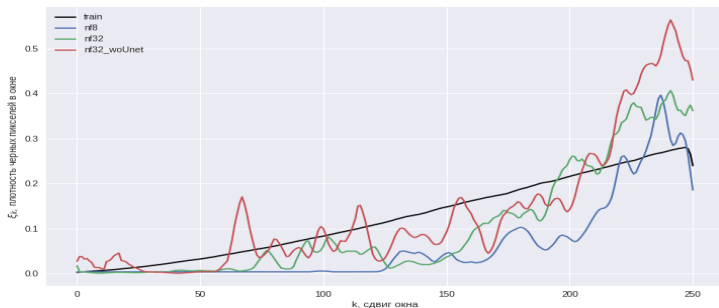


Рис.: Примеры сгенерированных текстур.

Результаты

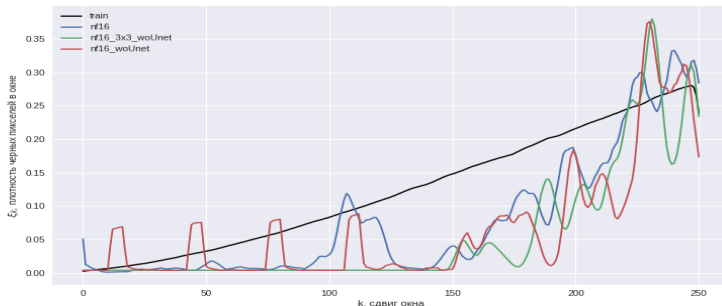
Для сгенерированных наборов текстур получились следующие результаты:



Сеть	Метрика
nf8	0.00825
nf32	0.00549
nf32-woUnet	0.00688

Таблица: Значения введенной метрики для разных сетей (меньше - лучше)

Результаты



Сеть	Метрика
nf16	0.00606
nf16-woUnet	0.00881
nf16-3x3-woUnet	0.01034

Таблица: Значения введенной метрики для разных сетей (меньше - лучше)

Заключение

В работе было:

- Исследовано применение архитектуры GAN для синтеза текстур с трендами
- Получены результаты синтеза при нескольких наборах гиперпараметров сети
- Проведено измерение качества генерации для каждого из наборов, используя введенную метрику

Полученные результаты показывают, что в принципе нейросеть способна уловить тренд и воспроизвести его, однако на данный момент качество генерации относительно невысоко. Необходимо провести дальнейшее исследование оптимальных гиперпараметров, и, возможно, увеличить объем обучающей выборки. Также можно провести аналогичные эксперименты с другими архитектурами генераторов.