Generative Adversarial Text to Image Synthesis(01-04)

논문 : https://arxiv.org/abs/1605.05396

발표 일자 : 2022-01-04(화)

발표자 : @김하연

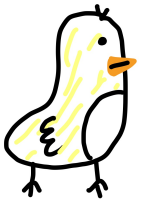
Abstract

현재 이미지에 대한 GAN은 이미 잘 연구되어 있다. 이에 이 논문은 text를 이미지 모델로 발전시키는데 관심을 가진다(즉, char → pixel) : text description으로부터 이미지 생성

Introduction

: 인간이 적은 묘사 → 이미지 픽셀

ex) this small bird has a short, pointy orange beak and white belly



natural language는 물체를 묘사하는데 있어서 **일반적이고 flexible**한 interface를 제공한다. (따라서, text description이 discriminator 성능을 높여줄 것이다.)

이 논문에서는, **Caltech-UCSD에서 zero-shot visual recognition**의 방법을 응용하였다.

▼ **Zero Shot**

- **제로샷 학습**은 학습된 데이터를 분류하고 정리해 카테고리를 형성한 뒤 그 카테고리의 의미를 이해하기 때문에 각 카테고리에 대한 의미적 이해를 바탕으로 이전에 경험하지 못한 새로운 예제를 이해하고 분석가능(전이학습에서 발전된 것)
- **전이학습 :** 데이터 간의 관계와 공통점을 이용해 학습(데이터가 많이 없어도 유용)

두 가지 문제 : → 딥러닝으로 이를 한번에 푸는 것이 목표!

1) 중요한 visual detail을 가지는 text feature로 학습을 해야한다.(natural language representation)

2) 이러한 feature로 사람들이 속을만큼 매력적이게 합성해야한다.(image synthesis)

가장 큰 문제 : ouput의 분포가 **multimodal**(그럴법한 image가 많음)

⇒ 반대(image→text)의 경우, chain rule을 이용하여 순차적으로 decompose가능

⇒ But, 우리(text→image)의 경우, 한 번에 진행해야 하기때문에 어렵다.

Main Contribution : simple하고 effective한 GAN구조와 training 전략을 만들어서 text를 통해 image를 잘 생성하는 것(데이터로 Caltech-UCSE Birds, Oxford-102 Flowers, MS COCO dataset 사용)

Method

Text feature에 대해 **DC-GAN**을 이용, Generator Network G와 discriminator network D는 feed-forward inference를 수행

4.1 Network Architecture

$G : \mathbb{R}^Z \times \mathbb{R}^T \rightarrow \mathbb{R}^D$, $D : \mathbb{R}^D \times \mathbb{R}^T \rightarrow \{0, 1\}$

(T : text의 dimension, D : image의 dimension, Z : generator network G에서 noise input의 dimension)

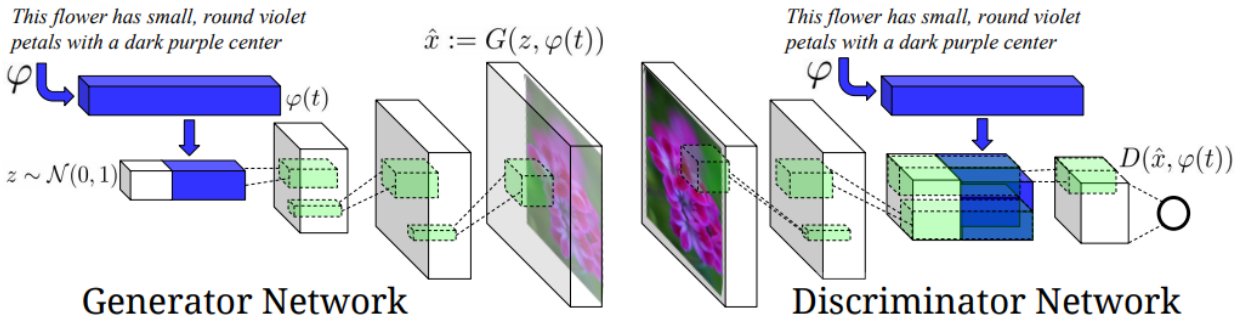


Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

1) generator G : noise prior z로부터 샘플을 추출하고 text encoder φ를 이용하여 text query t를 encode한다.(φ : 128 fully connected layer + leaky-ReLU → noise vector z에 concatenated → 이후, normal deconvolution network에 진행시켜서 합성된 이미지 x를 얻는다.)

text encoder is projected to a lower-dimensions

$$\hat{x} \leftarrow G(z, \varphi(t))$$

합성된 image $\hat{x}$는 다음과 같이 만들어 진다.(query text와 noise sample을 input으로!) : feed forward

2) Discriminator D : stride 2 conv with spatial BN + leaky ReLU

description embedding의 차원과 같아질때까지 반복한다. → 4 X 4 차원이 될 때 텍스트 벡터를 복사해서 D로 부터 final score를 구한다.

4.2 Matching-aware discriminator (GAN-CLS)

Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0, 1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

s_r : real, s_w : fake, s_f : fake, $\frac{\partial \mathcal{L}_D}{\partial D}$: D의 목적함수의 gradient

GAN을 학습하는 가장 직관적인 방법 : (text, image) pair를 real / fake로 구분하기(naive) → but, 진짜 이미지가 text embedding context와 일치하는지 확인할 수 없다.

Discriminator는 G가 그럴듯한 이미지를 생성하지 못하기 때문에 conditioning information을 무시한다.(따라서, 전제조건은 G가 그럴듯한 이미지를 생성하고, D는 G로 부터 sample들이 conditioning constrain를 충족하는지 평가해야한다.)

- Naive GAN :

  두 개의 input(real image with matching text, synthetic image with arbitary text) → 두 가지 에러(unrealistic image, realistic image but mismatch conditioning information)

4.3 Learning with manifold interpolation(GAN-INT)

> Deep networks have been shown to learn representations in which interpolations between embedding pairs tend to be near the data manifold

DN은 embedding 쌍 사이 보간이 data manifold 근처에 있는 representation을 학습한다.

→ 이 아이디어에서 착안하여, 트레이닝 셋 caption사이를 보간함으로써 추가적인 텍스트를 생성할 수 있다.(추가적인 cost X)

$$\mathbb{E}_{t_1,t_2\sim p_{data}}[\log(1-D(G(z,\beta t_1+(1-\beta)t_2)))] \quad (5)$$

β : text embedding(사람이 이해할 수 있는 text → vector) t1과 t2사이를 보간(0.5일때 잘 되는..)

→ D가 잘 작동한다면, G는 training points 사이의 **data manifold를 잘 채울 수 있다**.
(training data는 유한개인데, discrete하여 text embedding사이 gap이 발생한다. 이때, D가 잘 작동한다면, 이 gap을 채운다)

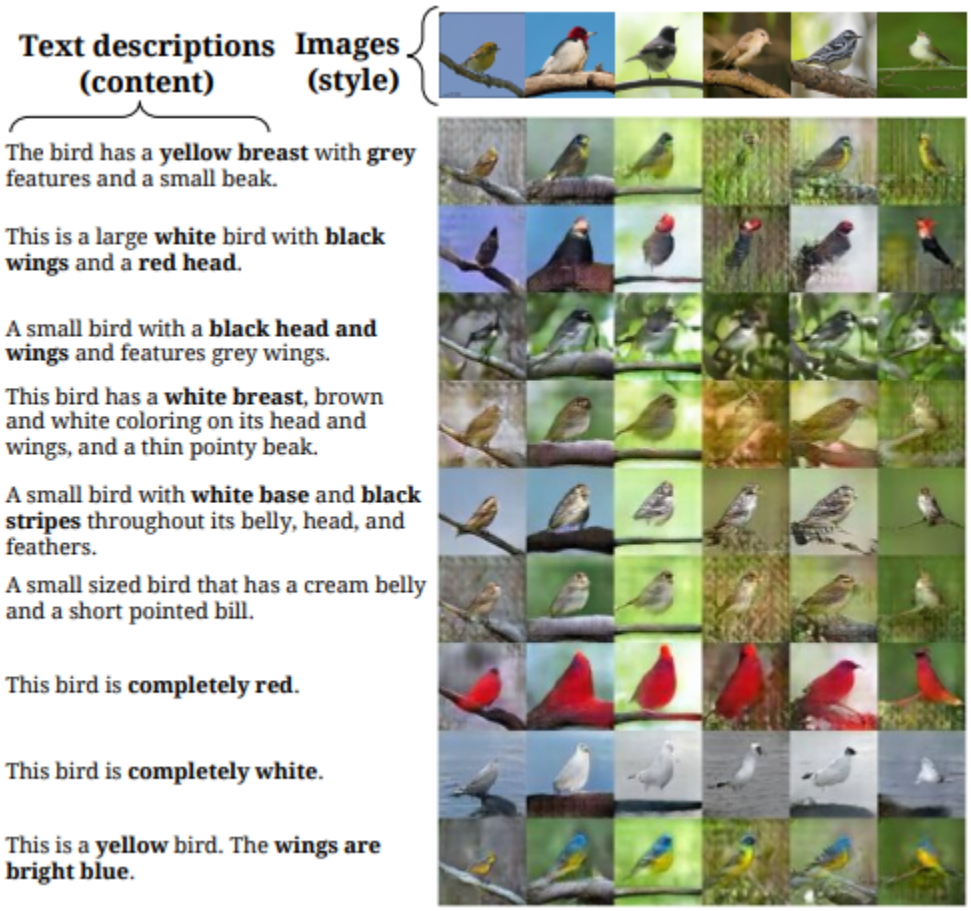
4.4 Inverting the generator for style transfer

- text encoding φ이 image content를 담아낸다면, noise sample z는 style factor를 담아내야 한다.
- $\hat{x} \leftarrow G(z,\varphi(t))$ 이 식으로 생성된 이미지 $\hat{x}$가 다시 noise z로 돌아가게 학습을 해야한다.
- S(style encoder)

$$\mathcal{L}_{style} = \mathbb{E}_{t,z\sim\mathcal{N}(0,1)}||z-S(G(z,\varphi(t)))||_2^2 \quad (6)$$

- style transfer(s = predicted style) : $s \leftarrow S(x), \hat{x} \leftarrow G(s,\varphi(t))$

Result


Text descriptions (content)
Images (style)
The bird has a yellow breast with grey features and a small beak.
This is a large white bird with black wings and a red head.
A small bird with a black head and wings and features grey wings.
This bird has a white breast, brown and white coloring on its head and wings, and a thin pointy beak.
A small bird with white base and black stripes throughout its belly, head, and feathers.
A small sized bird that has a cream belly and a short pointed bill.
This bird is completely red.
This bird is completely white.
This is a yellow bird. The wings are bright blue.