

Week17 논문 리뷰

Image-to-Image Translation with Conditional Adversarial Networks (Pix2pix)

이미지 처리, 그래픽 및 비전의 많은 문제들이 input image를 output image로 변환하는 것과 관련이 있다. 이러한 문제는 설정이 항상 동일하더라도 응용 프로그램 별 알고리즘으로 각각 처리되는 경우가 많다. cGAN(조건부 적대적 네트워크)은 다양한 문제에서 잘 작동하는 일반적인 방법이다.

** Edge를 Photo로 매핑하기 위해 Conditional GAN을 학습한다. Unconditional GAN과 다르게 generator, discriminator 모두 edge map을 살핀다.

cGAN을 이용해 한 유형의 이미지를 다른 유형의 이미지로 변환할 수 있는 framework를 제시한다. GAN을 사용하기 때문에 L1, L2 loss를 사용할 때보다 더 선명한 이미지를 얻을 수 있다. 또한 L1 loss를 사용했기 때문에 pixel간의 관계를 고려한 structured loss 개념도 적용되었다.

이미지의 빈 영역을 그럴싸하게 채우려면 대상 영상의 semantic을 정확하게 이해해야한다. 그러나 L1/L2 loss를 이용하는 것만으로는 복원 영상이 blur해지는 것을 피할 수 없다. 이 문제를 해결하기 위해 Adversarial Loss를 사용한다.

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

Bring-Old-Photos-Back-to-Life

손상된 사진을 자동 수정하는 Microsoft의 최신 사진 복원 프로젝트가 있다.

특히 스크래치 감지, 얼굴 개선 및 기타 기술을 활용하여 PyTorch에서 딥 러닝 구현을 통해 복잡한 성능 저하를 겪고 있는 오래된 사진을 복원한다.

연구논문에 따르면: "우리는 오래된 사진과 깨끗한 사진을 각각 두 개의 latent spaces로 변형시키기 위해 두 개의 Variational Autoencoders(VAE)를 훈련합니다. 그리고 이 두 latent spaces 사이의 변환은 합성 쌍 데이터(synthetic paired data)를 통해 학습됩니다. 이 변환은 콤팩트한 latent space에서 도메인 갭이 닫혀 있으므로 실제 사진으로 잘 일반화됩니다. 또한 우리는 오래된 사진에 혼합된 여러 가지 문제를 해결하기 위해 스크래치 및 먼지 얼룩과 같은 구조적 결함을 타겟팅하는 부분적 비로컬 블록과 글로벌 브랜치를 설계합니다. 또

한, 노이즈와 흐릿한 것들과 같은 구조화되지 않은 결함을 타겟팅하는 하는 로컬 브랜치도 설계합니다.”

아래 데모에서 알 수 있듯이 이 모델은 확실히 기존의 최첨단 방법을 능가합니다.

<https://github.com/microsoft/Bringing-Old-Photos-Back-to-Life>

Denoising Diffusion Probabilistic Model

생성 모형은 기본적으로 어떤 도메인(이미지, 자연어, 음성 등)의 복잡한 확률 분포를 모형화하는 것을 목적으로 한다. 여기서는 이미지 생성 모형에 대한 것을 다룬다. GAN은 2014년 이안 굿펠로가 제안한 framework로 생성자와 판별자의 minmax(adversarial) training을 통해 realistic한 이미지를 생성하는 생성자를 학습한다. 장점은 실제 사진과 유사한 일관성이 있는 이미지를 생성할 수 있다는 것이고, 모델 구조의 선택이 자유롭다는 것이다. 단점은 model의 coverage가 좁아서, 사실적인 이미지를 생성하기는 하지만 다양한 이미지를 생성하지는 못 한다는 것이다. 훈련 과정이 불안정하기도 하다.

Likelihood Based Model

flow model은 딥러닝 모델에 대한 역변환을 통해 단순한 분포(Normal distribution)에서 시작하여 복잡한 분포를 모델링하는 방법이다. 확률론에서 자주 언급되는 change of variable theorem에 기반한 방법이고, log likelihood objective로 학습이 이루어진다.

Diffusion Model

이는 원본 이미지에 gaussian noise를 순차적으로 추가하며 완전한 random noise로 만들어주는 과정(diffusion process)으로부터 이것의 역변환(inverse process)을 학습하고, 이 학습된 역변환을 사용하여 random noise로부터 이미지를 생성하는 메커니즘을 따른다.

DDPM

이미지에 노이즈를 순차적으로 추가하는 diffusion process의 역변환을 학습하는 것이 목적이다.

- diffusion process
원본 이미지 x_0 에 대해 순차적으로 gaussian Markov chain을 적용하는 것이고 그 횟수가 충분히 많아지면 x_T 는 완전한 random gaussian을 따르게 된다.
- Inverse process
Inverse process는 완전한 gaussian을 따르는 x_T 로부터 시작하여, 이것을 다시 x_0 로 되돌리는 과정이다. 이 과정을 학습하는 것이 ddpm의 목적이고 이는 gaussian Markov chain의 모수로 parameterize된다.

- Training Objective (variational bound)

이 논문에서는 negative log likelihood of x_0 의 variational bound(upper bound)를 최소화하는 방식으로 최적화를 진행하는데, 수식 전개는 다음과 같다.

$$\begin{aligned}
\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] &\leq \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] = \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \\
&= \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \\
&= \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \\
&= \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right] \\
&= \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \cdot \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right] \\
&= \mathbb{E}_q \left[-\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \\
&= \mathbb{E}_q \left[D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T)) + \sum_{t \geq 1} D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right]
\end{aligned}$$

세 번째 식에서 네 번째 식으로 넘어가는 부분이 중요하다고 한다.

1. Forward Process: L_T 이 부분은 학습 가능한 parameter가 없음.
2. Reverse Process: $L_{1:T-1}$ 이 부분을 최적화함으로써 최적의 값을 얻을 수 있음.

Training / Sampling 알고리즘 수도코드는 다음과 같다.

Algorithm 1 Training

```
1: repeat  
2:  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$   
3:  $t \sim \text{Uniform}(\{1, \dots, T\})$   
4:  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   
5: Take gradient descent step on  
    $\nabla_{\theta} \|\epsilon - \epsilon_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2$   
6: until converged
```

Algorithm 2 Sampling

```
1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   
2: for  $t = T, \dots, 1$  do  
3:  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$   
4:  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$   
5: end for  
6: return  $\mathbf{x}_0$ 
```
