**기수 :** 13 **이름 :** 한연주

|  |
| --- |
| A blue and white logo  Description automatically generatedA blue and white logo  Description automatically generated**25-1 DSL 정규 세션**  **Generative Model 과제** |
|  |
| * 본 과제는 학회 정규 세션 「Generative Model」의 내용을 다루며, 개념의 적용과 실제 활용 사례에 대한 이해를 돕기 위해 기획되었습니다. 해당 과제는 평가를 위한 것이 아니므로, 주어진 힌트(💡)를 적극 활용하시고 학회원 간 토론 및 Slack 질의응답을 적극 활용하여 해결해주십시오. 단, 답안 표절이나 LLM의 남용은 금지합니다. * 서술형 문제는 ✍️, 코딩 문제는 💻으로 표기가 되어 있습니다. 각 문제에서 요구하는 방법에 맞게 해결하며, 서술형 문제들은 따로 작성하시어 .pdf파일로, 코딩 문제들은 주어진 .ipynb 파일에 답안을 작성하여 제출해 주십시오. (**해당 과제의 경우 학기 중 진행되는 점을 고려, 이론 문제 (문제 1) 풀이는 개인 선택이니 참고바랍니다!**) * **4/17 (목) 23시 59분까지** Github에 .pdf 파일과 .ipynb 파일을 압축하여 **하나의 .zip 파일로 묶어 제출**해 주십시오. 반드시 **data나 사용한 모델 관련한 파일들을 제외**한 두 파일 만을 압축해 제출해 주십시오!  Github에 제출하는 방법을 모른다면 학술부장 혹은 과제 질의응답을 위한 오픈채팅방을 적극적으로 활용해 주십시오. |

|  |  |
| --- | --- |
| **문제 0** | Session Review |

**0** ✍️**:** Intro to GM / AE&VAE / GAN / Diffusion 세션에서 배운 내용을 자유롭게 요약 및 정리하세요.

|  |
| --- |
| 💡 어떠한 형태로든 요약해도 괜찮습니다. 다만 과제의 가장 큰 목적은 세션 내용의 복습이고, 제출하신 과제를 다 살펴보지 못하더라도 해당 부분은 반드시 확인할 예정이므로 어느 정도 노력을 들여 정리해주시길 바랍니다 :) |

**1. Intro to GM**

- Generative Models (GM): 목표는 데이터의 확률 분포를 학습하여 샘플링하거나 데이터 생성, 밀도 추정, 표현 학습을 수행하는 것.

- Discriminative Models: 데이터에 맞는 레이블을 예측하는 모델로, 𝑝(𝑦|𝑥)를 학습.

- Generative vs Discriminative: Generative는 데이터 𝑥의 확률 분포를 학습하고, Discriminative는 𝑝(𝑦|𝑥) 관계를 학습

- Autoregressive Models: 𝑝(𝑥)의 확률을 명시적으로 정의하고 계산하는 모델 (예: PixelRNN, PixelCNN)

- Explicit vs Implicit Models: Explicit은 분포를 명시적으로 정의하고 계산하는 반면, Implicit은 샘플링 방식으로 학습.

- Maximum Likelihood Estimation (MLE): 데이터와 모델의 분포 간 차이를 최소화하는 방법.

- Generative Model Taxonomy: 다양한 유형의 GM(Autoregressive, VAE, GAN 등) 분류

**2. GAN**

- GAN (Generative Adversarial Network): 생성자(Generator)와 판별자(Discriminator) 간의 경쟁을 통해 데이터 분포를 학습하여 가짜 데이터를 생성.

- Likelihood-Free Learning: GAN은 기존의 확률분포를 최대화하는 방식과 달리, 생성된 데이터와 실제 데이터를 비교하며 학습.

- Minimax Game: Generator와 Discriminator는 서로 최적화하며, Generator는 가짜 데이터를 생성하고, Discriminator는 실제와 가짜를 구별.

- Challenges: Mode Collapse, Vanishing Gradients, Unstable Optimization 등의 문제 발생.

- Evaluation Metrics: Inception Score(IS), Frechet Inception Distance(FID), Kernel Inception Distance(KID) 등을 사용하여 GAN 성능 평가.

- DCGAN, WGAN: DCGAN은 convolutional layers로 GAN 구조를 개선한 모델이고, WGAN은 Wasserstein distance를 사용하여 기울기 소실 문제를 해결.

**3. Diffusion**

- Diffusion Models: 이미지 생성에서 Noise를 점진적으로 추가하는 Forward Process와 이를 역으로 복원하는 Reverse Process로 구성.

- DDPM (Denoising Diffusion Probabilistic Models): 데이터에 점진적으로 Gaussian Noise를 추가하고, 이를 역으로 복원하여 이미지를 생성.

- Noise Schedule: 각 시간 단계에서 Noise를 조정하여 학습.

- ELBO (Evidence Lower Bound): Variational Inference에서 Untractable한 최대화 문제를 근사하기 위해 사용.

- Classifier-Free Guidance (CFG): 별도의 Classifier 없이 조건을 주어 모델을 제어하는 방법.

- Latent Diffusion Models (LDM): 픽셀 공간과 잠재 공간을 분리하여 Diffusion 학습을 효율적으로 처리하는 방식.

**4. AE&VAE**

- AutoEncoder (AE): 입력 데이터를 압축하고, 복원하는 신경망 구조. Representation Learning에 사용.

- Variational AutoEncoder (VAE): AE의 확장으로, 잠재 변수 𝑧를 확률 분포로 모델링. ELBO를 사용해 파라미터를 최적화.

- Reparameterization Trick: VAE에서 잠재 변수를 샘플링할 수 있도록 하는 기법.

- Learning Objective: 데이터의 잠재 공간에서 표현을 학습하고, 이를 통해 새로운 데이터를 생성하거나 밀도 추정을 수행.

- Limitations: VAE는 확률 분포를 정확하게 모델링하는 데 어려움이 있음.

- Advanced Topics: CVAE(Conditional VAE) 및 AAE(Autoencoder-Adversarial Autoencoder) 등 확장된 모델.

|  |  |
| --- | --- |
| **문제 1** | ELBO for Conditional VAE (Optional) |
| <Introduction to Generative Model>과 <AE & VAE> 세션에서 우리는 VAE가 사용하는 중요한 원리 중 하나인 Evidence of Lower Bound, 줄여서 ELBO라고 부르는 형태를 살펴봤습니다. 우리가 원하는 데이터의 분포를 구하는 행위, 즉 데이터의 likelihood를 최대화하기 위한 행위는 ELBO를 최대화하는 방식으로 근사할 수 있다는 것입니다. 이러한 ELBO는 VAE에 라벨 정보를 추가로 활용하는 CVAE에도 동일하게 적용되는데요. 여기서는 그 내용과 관련된 두 가지 사실을 이론적으로 증명해보고자 합니다. | |

**1-1** ✍️**:** Conditional VAE의 variational lower bound를 유도하세요. 다시 말해,  
 텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

|  |
| --- |
| 💡 **y**라는 label 변수가 추가된 것뿐이지, 어쩌면 vanilla VAE와 증명 과정이 거의 비슷할 지 모릅니다!  💡 VAE의 증명 과정은 관련 세션들의 자료를 참고하세요. |

**1-2** ✍️**:** 두 분포 p와 q를 Gaussian Distribution이라 가정했을 때, 두 분포의 KL Divergence과 다음과 같음을 보이세요. (단, )  
텍스트, 폰트, 라인, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

|  |
| --- |
| 💡 해당 등식을 증명하는 과정에서 다음과 같은 trace trick이 필요하며, 기댓값 연산과 trace 연산은 교환 가능합니다. |

|  |  |
| --- | --- |
| **문제 2** | GAN and its variants |
| GAN 세션은 모두 재밌으셨나요? 이번 문제는 모두 구현 문제로, 이미 코드가 구현이 완료된 두 파일을 실행만 해보시면 되는 간단한!! 문제입니다. 직접 두 파일을 시행하며, GAN의 핵심적인 구조와 StyleGAN을 활용해 얼굴 이미지를 생성하는 과정을 살펴보도록 하죠 :D | |

**2-1** ✍️, 💻**:** GAN.ipynb 파일을 직접 실행해 보세요. 해당 결과와 관련 내용을 토대로, GAN에 비해 DCGAN의 성능이 상대적으로 향상된 이유를 간단히 설명하세요.

**DCGAN은 합성곱 층을 사용해 이미지의 특성을 잘 학습하므로, 더 선명한 이미지를 생성하고 더 잘 군집화한다. 반면, Vanilla GAN은 완전 연결층을 사용해 성능이 떨어진다.**

**2-2** 💻**:** StyleGAN.ipynb 파일을 실행하고 나온 결과 (웃는 얼굴, 무표정 얼굴 png 파일)를 여기에 첨부해주세요.

스크린샷, 사람이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

스크린샷, 인간의 얼굴, 체스 말이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.