

조건부 딥 컨볼루션 생성적 적대 신경망(CDCGAN) 구현 및 실험 보고서

나노전자물리학과 20201914 박규민

April 15, 2025

1 서론

본 과제의 목표는 MNIST 숫자 이미지 데이터셋을 기반으로 조건부 딥 컨볼루션 생성적 적대 신경망 (Conditional Deep Convolutional Generative Adversarial Network, CDCGAN)을 구현하고, 클래스 라벨(숫자 0부터 9까지)에 조건을 부여한 이미지 생성 모델을 학습하는 것이다. CDCGAN은 기본 GAN 구조에 조건 정보를 결합하여, 생성자가 단순히 무작위 이미지를 생성하는 것이 아니라 특정 클래스에 해당하는 이미지를 생성할 수 있도록 설계된다.

모델의 학습이 완료된 후, Gradio 기반의 사용자 인터페이스를 활용하여 다양한 조건과 latent vector를 입력으로 주고, 생성 이미지의 품질과 변화를 관찰하였다. 본 보고서에서는 CDCGAN의 모델 설계 방식, 학습 과정, Gradio 기반의 실험 결과를 중심으로 과제 수행 내용을 정리하고자 한다.

2 모델 설계

본 과제에서 구현한 CDCGAN 모델은 조건 정보(숫자 클래스)를 활용하여, 해당 조건에 맞는 MNIST 숫자 이미지를 생성하는 구조로 설계되었다. 전체 네트워크는 생성기(Generator)와 판별기(Discriminator)로 구성되며, 두 네트워크 모두 class label 정보를 입력으로 받아 조건부 이미지 생성을 수행한다.

2.1 Generator 구조

생성기는 입력된 latent vector $z \in R^{nz}$ 와 조건 $c \in \{0, 1, \dots, 9\}$ 를 함께 입력으로 받아, 해당 조건에 맞는 28×28 크기의 흑백 이미지를 생성하는 역할을 한다. latent vector z 는 5차원에서 선형 변환을 통해 50차원으로 투영되며, 조건 정보 c 역시 50차원 임베딩 벡터로 변환된다. 이후 이 두 벡터를 concat하여 총 100차원 벡터로 구성된다.

이때, z 를 50차원으로 확장(projection)하는 이유는 조건 벡터와 동등한 표현력을 갖도록 하기 위함이며, 두 벡터의 정보량 균형을 맞추므로써 결합된 벡터가 효과적으로 학습될 수 있도록 한다. 즉, 조건 정보에 과도하게 의존하거나 반대로 무시되는 현상을 방지하고, 생성기 입력에서 각 요소의 영향력을 적절히 유지하기 위한 사전 조정 과정이다.

이 벡터는 선형 계층을 거쳐 $256 \times 7 \times 7$ 크기의 텐서로 투영되고, 이후 ConvTranspose2D 기반 업샘플링 연산을 통해 최종 이미지를 생성한다.

업샘플링 과정에서는 총 세 단계의 ConvTranspose2D 블록을 사용하며, 각 블록마다 Batch Normalization과 LeakyReLU 활성화 함수가 적용된다. 추가적으로, 중간 단계에 MaxPooling과 AvgPooling 연산을 삽입하여 feature map의 공간적 분포를 안정화하고, 생성 이미지의 표현 품질을 향상시켰다. 마지막에는 이미지 크기를 28×28 로 정규화하기 위해 CenterCrop 연산을 적용하였다.

2.2 Discriminator 구조

판별기는 입력 이미지 $x \in R^{1 \times 28 \times 28}$ 와 조건 $c \in \{0, 1, \dots, 9\}$ 를 함께 받아, 해당 이미지가 진짜(real)인지 생성된(fake) 이미지인지를 판별하는 역할을 수행한다. 조건 c 는 임베딩 레이어를 통해 784차원($=28 \times 28$) 벡터로 매핑되며 $(1 \times 28 \times 28)$ 텐서로 reshape되어 원본 이미지와 채널 차원에서 concat된다. 이를 통해 최종 입력은 $(2 \times 28 \times 28)$ 형태가 된다.

이후 Conv2D 기반의 다운샘플링 블록이 두 단계 적용되며, 각각 LeakyReLU 활성화 함수와 Dropout(0.3)을 포함한다. Flatten 연산 후, 최종적으로 1차원 출력으로 압축되며, Sigmoid 함수로 이진 분류 결과를 출력한다. Discriminator는 Generator보다 상대적으로 간단한 구조를 가지며, 판별 정확도 향상을 위해 dropout을 도입하였다.

2.3 조건 주입 방식

Generator와 Discriminator 모두에서 조건 정보 c 는 네트워크에 명시적으로 주입된다. Generator에서는 latent vector z 를 선형 변환하여 50차원으로 확장한 후, 50차원으로 임베딩된 조건 벡터와 벡터 차원에서 결합하였다. 반면 Discriminator에서는 조건 c 를 28×28 크기의 텐서로 임베딩한 후, 입력 이미지와 채널 차원에서 concat하여 $(2 \times 28 \times 28)$ 형태로 입력하였다. 이러한 조건 주입 방식은 class label이 명확히 반영된 이미지 생성을 가능하게 하며, 모델이 각 조건에 해당하는 시각적 특성을 학습할 수 있도록 돕는다.

3 학습 설정 및 진행

본 과제에서는 PyTorch Lightning 프레임워크를 기반으로 CDCGAN 모델의 학습을 수행하였다. 학습 데이터는 torchvision 패키지의 MNIST 숫자 이미지 데이터셋을 사용하였으며, ToTensor()와 Normalize(mean=0.5, std=0.5)를 적용하여 $[-1, 1]$ 범위로 정규화하였다. 배치 크기는 100으로 설정하였다.

학습은 Generator와 Discriminator를 각각 Adam Optimizer로 훈련시키는 방식으로 구성되었으며, 주요 하이퍼파라미터는 아래와 같다.

- Learning rate: 0.0002
- β_1 : 0.5
- Latent vector dimension: 5
- Epochs: 3
- Loss: Binary Cross Entropy (BCE)

모델의 학습은 조건 라벨 c 와 랜덤 노이즈 z 를 입력으로 받아 Generator가 이미지를 생성하고, Discriminator가 진짜/가짜를 조건과 함께 판단하는 방식으로 이루어진다. Generator와 Discriminator는 manual optimization 방식으로 번갈아 학습되며, 학습이 완료된 후 모델 가중치는 cdcgan.ckpt 파일로 저장되었다.

전체 학습 파라미터 수는 약 2.4M이며, 이 중 Generator가 약 2.2M, Discriminator가 약 147K를 차지한다. 학습된 모델은 Gradio 인터페이스를 통해 사용자 입력에 따라 숫자 이미지를 생성하는 데 활용되었다.

4 결과 분석

4.1 Gradio를 통한 이미지 생성

학습된 CDCGAN 모델은 PyTorch Lightning을 통해 훈련된 후, Gradio 기반의 웹 인터페이스를 통해 사용자와의 상호작용이 가능하도록 구성되었다. 사용자는 조건 라벨 c 를 직접 지정하고, latent vector의 다섯 가지 차원 $z_0 \sim z_4$ 값을 슬라이더로 조절할 수 있다. 이러한 조작을 통해 조건에 맞는 다양한 숫자 이미지 생성이 실시간으로 이루어진다.

4.2 생성 이미지 예시

그림 1은 모든 latent vector 값이 0으로 설정된 상태($z_0 = z_1 = z_2 = z_3 = z_4 = 0$)에서 각 조건 라벨 $c = 0, 1, \dots, 9$ 에 따라 생성된 숫자 이미지들을 보여준다.

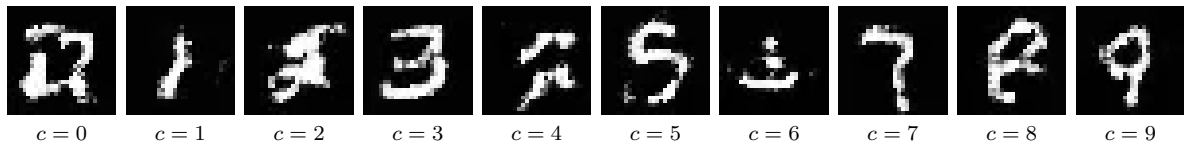


Figure 1: latent vector $z = \vec{0}$ 일 때, 조건 $c = 0 \sim 9$ 에 따라 생성된 이미지

위 결과를 보면, 모든 latent vector를 0으로 고정하더라도 각 class label에 따라 서로 다른 형태의 숫자가 생성됨을 확인할 수 있다. 이는 Generator가 조건 정보를 충분히 반영하여 class 특성을 학습했음을 보여준다.

다만 일부 숫자(예: $c = 4, c = 6, c = 8$ 등)는 형태가 불분명하거나 붕괴된 경향이 있으며, 이는 latent vector의 다양성이 부족하거나 학습 epoch 수가 충분하지 않아서 생긴 현상일 수 있다. 일반적으로 z 가 0에 가까운 값일 때는 평균적인 숫자 형태가 나타나지만, CDCGAN의 경우 class 조건의 비중이 커서 latent vector가 고정되어도 생성 결과가 제법 다양하게 나타날 수 있다.

이 실험은 latent vector의 영향 없이 label 정보만으로도 이미지 생성이 가능함을 보여주며, 향후에는 다양한 z 값을 조절하여 생성 이미지의 스타일 변화를 추가적으로 분석할 수 있다.

4.3 생성 이미지 분석

그림 2은 조건 라벨 $c = 3$ 을 고정한 상태에서, latent vector z 의 값을 변화시켰을 때 생성되는 이미지의 변화를 나타낸다.

CDCGAN 모델을 통해 조건(label)에 맞는 숫자 이미지를 생성하는 실험을 수행한 결과, 주어진 label에 따라 전형적인 숫자 형태가 잘 생성되었으나, latent vector를 변화시켜도 생성 이미지가 거의 달라지지 않는 현상이 관찰되었다. 이는 모델이 학습 과정에서 latent vector보다 label 정보에 훨씬 강하게 의존하게 되었으며, 결과적으로 z 의 영향력이 사실상 무시되고 있다는 것을 의미한다.

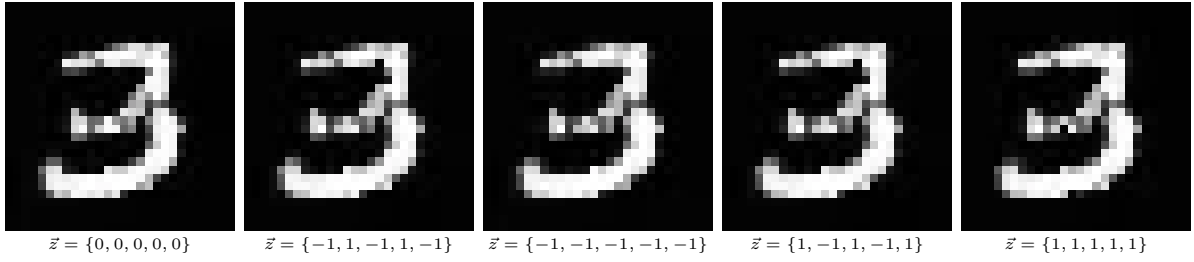


Figure 2: 조건 $c = 3$ 고정 하에 다양한 latent vector z 조합에 따라 생성된 숫자 이미지

이를 방지하기 위해 Generator에서는 latent vector z 와 label 임베딩 벡터가 동일한 차원(50)으로 조정되도록 설계하였다. 구체적으로는 5차원의 z 를 선형 계층을 통해 50차원으로 확장하고, 50차원으로 임베딩된 label 벡터와 concat하여 label 정보와 z 정보가 동등한 비중으로 결합되도록 하였다. 이는 Generator가 학습 과정에서 두 정보를 균형 있게 고려하게 만들기 위한 코드적 장치였다. 그러나 이러한 구조적 설계에도 불구하고, 최종 결과에서는 여전히 label 정보가 주도적인 역할을 하고, z 의 다양성이 충분히 반영되지 않는 한계가 있었다.

이는 CVAE에서의 결과와 뚜렷한 대조를 이룬다. CVAE의 경우 latent 공간의 연속적인 이동에 따라 이미지의 굵기, 기울기, 윤곽 등이 자연스럽게 달라졌고, label이 동일하더라도 다양한 시각적 표현이 가능하였다. 반면 CDCGAN에서는 동일한 label이면 z 를 극단적으로 바꾸더라도 거의 동일한 이미지가 생성되었으며, 이는 Generator가 label 정보에만 의존한 채 학습되었거나, latent vector의 차원이나 표현력이 부족했을 가능성이 있다.

실제로 실험에서는 $z = [-1, -1, -1, -1, -1]$ 부터 $z = [1, 1, 1, 1, 1]$ 까지 슬라이더를 통해 변경하였으나, 이미지 형태나 스타일에서 거의 차이가 발생하지 않았다. 이처럼 z 값이 출력 이미지에 거의 영향을 주지 않는 현상은 모델이 다양한 스타일을 생성하지 못하고 있다는 점에서 생성 모델로서의 표현력이 제한됨을 시사한다.

또한 CVAE와 비교했을 때, 학습 속도 측면에서도 큰 차이가 있었다. CVAE는 비교적 적은 epoch 수로도 빠르게 수렴하였으며, latent 공간 구조가 명확히 시각화되는 특성을 보였다. 반면 CDCGAN은 학습 자체에 더 많은 시간과 리소스를 요구하였고, Generator와 Discriminator의 균형을 맞추는 과정도 까다로웠다. 특히 PyTorch Lightning 기반의 manual optimization 방식에서는 Generator와 Discriminator를 번갈아 학습시키기 위한 제어가 필요한 반면, CVAE는 end-to-end 구조로 상대적으로 구현과 학습이 간단하였다.

결과적으로 CDCGAN은 label 조건에 따라 숫자 구조를 정확히 반영하는 데는 성공하였지만, latent vector의 의미를 충분히 반영하지 못했다는 한계를 가진다. 이는 향후 z 의 정보를 효과적으로 활용할 수 있도록 구조를 개선하거나, z 에 대한 보조 손실을 추가하는 방식으로 극복할 수 있을 것이다. 또한 latent dimension을 확장하거나, label과 z 를 보다 균형 있게 결합하는 방식으로 Generator의 다양성 학습을 유도할 필요가 있다.

5 결론 및 소감

이번 과제를 통해 조건부 딥 컨볼루션 생성적 적대 신경망(CDCGAN)을 직접 구현하고, MNIST 숫자 이미지에 class label을 조건으로 주어 다양한 이미지를 생성해보는 실험을 진행하였다. Generator와 Discriminator는 컨볼루션 신경망 구조를 기반으로 구성하였고, label 정보를 embedding하여 latent vector와 함께 사용함으로써 조건에 맞는 이미지를 생성하는 구조를 설계했다.

실험 결과, 모델은 각 class label에 따라 해당 숫자를 비교적 잘 생성해냈으며, latent vector를 모두 0으로 고정했을 때에도 label에 따라 서로 다른 숫자가 생성되는 것을 확인할 수 있었다. 하지만 latent vector 값을 변화시켜도 이미지가 크게 달라지지 않는 현상이 나타났고, 이는 모델이 label 정보에 너무 의존하면서 latent vector의 영향을 잘 활용하지 못했기 때문이라고 생각된다.

이러한 부분은 추후 학습 epoch을 늘리거나 latent vector의 차원을 조정하는 방식으로 개선할 수 있을 것 같다. 또한 Generator 구조를 조금 더 복잡하게 설계하거나, z 의 정보를 더 강하게 반영할 수 있도록 구조를 조정해보는 것도 흥미로운 시도일 것 같다.

Gradio 인터페이스를 활용해 직접 label과 latent vector를 조절해가며 이미지를 생성해본 과정은 특히 흥미로웠다. 단순히 코드만 작성하는 것이 아니라, 직접 시각적으로 결과를 확인하면서 모델이 어떻게 작동하는지를 몸으로 느낄 수 있는 경험이었다.

전반적으로 이번 과제를 통해 조건부 생성 모델의 작동 원리와 구현 과정을 잘 이해할 수 있었고, 생성 모델이 어떻게 데이터를 학습하고 표현하는지를 보다 실감나게 배울 수 있었다. 앞으로 GAN이나 다른 생성 모델에 대해서도 더 깊이 공부하고, 다양한 응용 사례에도 도전해보고 싶다는 생각이 들었다.

이번 분석을 통해 단순한 label-to-image 매핑을 넘어서, 진정한 조건부 생성 모델이 갖추어야 할 다양성과 표현력에 대해 다시금 생각해볼 수 있었다. 특히 동일한 조건에서 다양한 스타일의 출력을 생성하는 것이 얼마나 중요한 요소인지, 그리고 그것을 구현하기 위한 구조적 고민이 왜 필요한지를 실제 구현을 통해 체감할 수 있었던 의미 있는 경험이었다.