

# Auto Encoder를 이용한 위치데이터의 차원 확장 GAN학습\*

전희준<sup>0</sup>, 송하윤

홍익대학교 컴퓨터공학과

yeonso16385@naver.com, hayoon@hongik.ac.kr

## Extending the Dimension of Human Mobility Data with Auto Encoder and GAN Training

Heejun Jeon<sup>0</sup>, Hayoon Song

Dept. of Computer Engineering, Hongik University

### 요 약

Generative Adversarial Network(GAN)은 훈련데이터의 확률분포를 학습하는 대표적인 생산적 모델(Generative Model)로 여러 분야에 활용되고 있다. 최근 모바일 장치를 이용한 포지셔닝 데이터의 대량 수집이 가능해지면서 GAN을 활용해 위치데이터(위도, 경도)를 생산하는 연구가 있었다. 하지만, 훈련 데이터가 위치데이터와 같이 복잡한 분포를 가지는 저차원 데이터인 경우 GAN의 학습이 불안정해진다는 문제점이 있다. 본 논문은 기본적인 Auto Encoder(AE)를 이용해 위치데이터의 차원을 확장시키는 방법을 제시한다. 실험을 통해, 해당 방법으로 차원이 늘어난 데이터를 GAN에 학습시킨다면 학습 안정에 효과가 있고, 의미있는 학습이 가능하다는 것을 확인하였다.

### 1. 서 론

Generative Adversarial Network(GAN) [1]은 생산적 모델(Generative Model)의 한 분야로, Generator와 Discriminator 간의 적대적인 학습을 통해 훈련 데이터의 확률분포를 배우는 모델이다. 다른 생산적 모델보다 더욱 다양한 분포를 표현할 수 있기 때문에 의료, 경제 등 여러 분야에서 활발한 연구가 진행되고 있다. 하지만, 적대적인 학습으로 인한 mode collapse현상과 gradient vanishing 등의 문제는 GAN의 학습을 어렵게 만든다. 특히, 데이터가 저차원 매니폴드(manifold)를 가지고, Discriminator가 갖는 매개변수의 수가 클수록 학습은 더욱 불안정해진다[2].

위치데이터 분야에서도 GAN의 연구가 진행되어 왔다. 그러나 위도, 경도 2차원으로 구성된 위치데이터의 경우 저차원 데이터이고, 분포가 복잡하기 때문에 용량이 작은 Discriminator로는 학습이 힘들다. 반면, 용량이 큰 Discriminator를 이용할 경우 학습은 불안정해진다 [2]. 따라서, 기존 연구는 [3] 위치데이터는 2차원 지도 위에 분포한다는 점을 이용해, 여러 위치데이터를 묶어 위치 시퀀스(sequence)를 구성한 후, 고차원 이미지 데이터로

변환해 학습을 진행하였다. 즉, 훈련데이터의 특수한 성질을 이용해 차원을 확장했다. 본 논문은 위치데이터를 고차원 데이터로 변환하는 방법으로 기본적인 Auto Encoder(AE) [4]를 이용한다. 실험을 통해, 이 방법으로 차원이 확장된 데이터를 GAN에 학습시킬 경우 보다 안정적인 학습이 가능하다는 것을 확인했다. 또한, 학습을 마친 GAN이 생성한 데이터들의 각 feature들 사이 관계를 시각화함으로써 의미있는 학습이 일어나는 것을 확인했다.

### 2. 관련 연구 및 배경

#### 2.1. Generative Adversarial Network(GAN)

GAN은 Generator와 Discriminator라는 두 네트워크로 구성되어 있다. Discriminator의 목적은 Generator의 목적은 Discriminator를 속여 자신이 만든 페이크데이터(fake data)를 리얼데이터(real data)로 판단하게 만드는 것이고 Discriminator의 목적은 두 데이터의 클래스를 제대로 분류하는 것이다. 즉, 두 네트워크는 서로 반대되는 방향으로 학습을 진행한다. GAN의 전체적인 구조는 그림 1과 같다. Generator는 random vector를 입력받아 페이크 데이터를 생성한다. Discriminator는 페이크 데이터와 리얼 데이터를 입력받아 각 데이터가 리얼 데이터로 추정되는 확률을 출력한다. 두 네트워크는 다음과 같은 손실함수  $V(G,D)$ 를 이용해 학습을 진행한다.

$$\min_G \max_D V(G,D) =$$

\* This research was supported by the WEF Global Partnership Program for the Fourth Industrial Revolution through KAIST KPC4IR funded by the Ministry of Science and ICT.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (NRF-2019R1F1A1056123)

$$E_{x \sim P_{data}(x)} [\log D(x)] + E_{z \sim P(z)} [\log (1 - D(G(z)))] \quad (1)$$

G와 D는 Generator와 Discriminator이고,  $P_{data}(x)$ 와  $P(z)$

는 리얼데이터와 random vector의 확률분포를 의미한다. 이론적으로, (1)의 결과 페이크 데이터의 확률분포와 리얼데이터의 확률분포 사이 거리로 측정되는 Jensen Shannon Divergence(JSD)가 0이 되면서 두 분포는 같아

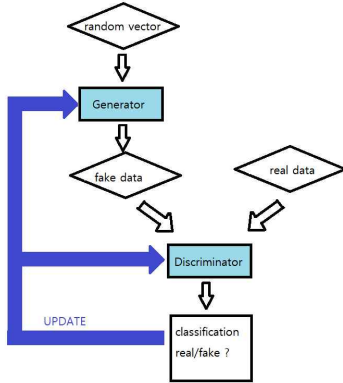


그림1. 전체적인 GAN의 구조.

진다. 하지만, GAN은 실제 학습과정에서 mode collapse, gradient vanishing 등 여러 문제점이 발생한다. 이를 해결하기 위해, WGAN [5]은 두 확률분포 사이 거리로 JSD 대신 Wasserstein distance를 이용해 손실함수를 새롭게 정의했다. WGAN의 손실함수는 (2)와 같다.

$$\min_G \max_f V(G, f) = E_{x \sim P_{data}(x)} [f(x)] - E_{z \sim P(z)} [f(G(z))] \quad (2)$$

$$\text{subject to: } |f(x_1) - f(x_2)| \leq |x_1 - x_2| \quad (3)$$

(2)의 결과로 페이크데이터, 리얼데이터의 확률분포 사이의 Wasserstein Distance가 0이 되고 두 분포는 같아진다. 여기서  $f(x)$ 는 1-Lipschitz 조건이라고도 불리는 (3)을 만족해야 한다. 이를 위해 WGAN-GP [6]는 (2)에 gradient penalty항 (4)를 추가하는 방법을 제안했다.

$$E_{x \sim P_z} [\lambda (\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2], \quad \hat{x} = \epsilon x + (1 - \epsilon) \tilde{x} \quad (4)$$

여기서  $x, \tilde{x}$ 는 리얼데이터와 페이크데이터를 의미하며,  $\epsilon \sim U[0, 1]$ 는 확률변수,  $\lambda$ 는 하이퍼파라미터이다.

한편, [2]는 실험을 통해 훈련데이터가 저차원 매니폴드를 갖고 Discriminator의 용량이 크다면 WGAN을 이용하더라도 GAN 학습이 수렴 되지 않고, 불안정하다는 것을 보였다. 또한, 손실함수에 gradient penalty항 (5)를 추가한다면 몇가지 조건 하에 GAN 학습이 안정적으로 수렴한다는 것을 증명했다. 저자는 이를 R1 정규화(regularization)라고 정의했다.

$$R1 = \lambda E_{x \sim P_{data}} [\|\nabla D(x)\|^2] \quad (5)$$

## 2.2 Auto Endoer(AE)

AE는 두 네트워크 Encoder와 Decoder로 구성된 모델이로, 훈련데이터를 Encoder에 입력하고, Encoder의 출력

(latent code)을 다시 Decoder에 입력하는 구조이다. 학습 목표는 Encoder의 입력, 즉 훈련데이터와 Decoder의 출력이 같도록 만들어, latent code가 훈련데이터의 중요한 정보를 학습하게 만드는 것이다. 주로 훈련데이터의 차원을 압축하기 위해 사용하지만, 본 논문에서는 차원을 확장하기 위해 사용했다.

## 3. 실험 및 결과

### 3.1 data set 및 전체적인 구조

실험을 위해, 한 개인이 직접 2014년 3월 1일부터 5월 31일(92일 기간)까지 sportstracker 모바일 어플을 이용해 수집한 위치데이터(위도, 경도) 중, 홍익대학교 근처로 측정된 좌표 약 30000개를 훈련데이터로 이용하였다. [2]에서 제시한 문제를 해결하기 위해, AE를 이용해 위도, 경도로 구성된 2차원의 위치데이터의 데이터의 차원을 확장한다. AE학습이 이상적으로 진행 된다면, Encoder의 출력(latent code)은 위치데이터의 중요한 정보를 가진 고차원 벡터이며, Decoder의 출력은 훈련데이터와 같다. 이후, latent code를 GAN의 훈련데이터로 사용한다. GAN학습이 이상적으로 진행된다면, Generator는 latent code의 확률분포를 배우게 되고, 이후 Generator가 생성한 페이크데이터를 Decoder에 입력하면 최종적으로 위도, 경도의 위치데이터가 출력된다.

### 3.2 AE구조 및 GAN구조

실험에서 AE는 간단한 Fully Connected를 이용했다. Encoder의 첫 번째 층은 입력이 2차원 위치데이터이고 출력은 4차원이다. 나머지 층을 지날수록 출력 차원은 2의 지수승(8,16,32)으로 확장되고 마지막 층의 출력은 32차원이다. Decoder는 Encoder와는 반대로 차원이 축소되고 대칭적인 구조를 가지고 있으며 마지막 층의 출력은 2차원이다. 해당 구조를 통해 학습한 결과, 전체 훈련데이터와 Decoder의 출력 사이 위도,경도 좌표별 오차는 평균 약 0.0001(1e-4)임을 확인했다. 하지만, 위치데이터의 경우 0.00001(1e-5)차이가 약 1m 차이를 의미하기 때문에, 오차를 더 줄이는 방법이 필요하다. 모델의 복잡성을 증가시킨다면 훨씬 더 좋은 결과를 얻을 것이라 기대한다.

GAN의 구조도 Fully Connected로 구현했다. Generator와 Discriminator 모두 은닉 층(hidden layer) 2개로 구성된다. Generator는 128차원의 random vector를 입력받고 이를 은닉층으로 전달한다. 은닉층은 feature간의 관계를 더 잘 표현하기 위해 이전 층의 출력을 다음 층의 출력과 접합(concatenate)시켜 순전파를 진행한다. 출력 층은 32차원 페이크데이터를 출력한다. Discriminator는 32차원의 데이터를 입력받아 은닉 층으로 전파한 뒤 출력층을 거쳐 1차원의 값을 반환한다.

### 3.3 차원확장과 GAN의 학습

먼저, 차원확장이 안정적인 GAN 학습에 영향을 주는 지 확인하기 위해 2차원 위치데이터 그대로 GAN을 학습시킨 경우와, 32차원으로 확장시킨 후의 경우를 비교했다. 학습은 WGAN-GP를 이용했다. 그림5 는 epoch마다

critic\_loss와 generator\_loss의 변화를 나타낸다. critic\_loss는 (2)의  $V(G,f)$ 와 관련된 값이며 두 확률분포 사이 Wasserstein distance를 추정할 값이다. generator\_loss는  $V(G,f)$ 의 뒤의 항이며 페이크데이터를 입력으로 받을 때 Discriminator의 값을 의미한다. 2차원의 경우 32차원 보다 critic\_loss, generator\_loss가 더 큰 범위 안에서 진동하는걸 볼 수있다. 이는 2차원 데이터의 학습이 더 불안정하다는 것과 차원 확장이 안정적인 학습에 도움이 된다는 것을 의미한다.

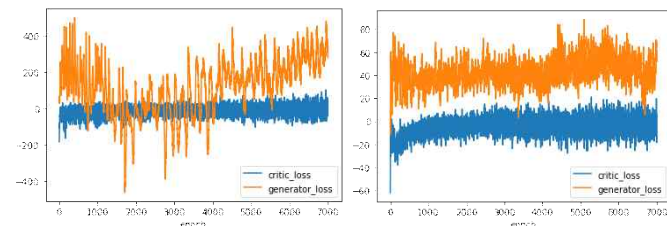


그림3. 2차원 데이터(왼쪽)와 32차원 데이터(오른쪽)의 GAN학습

다음으로, 차원 확장된 데이터를 이용한다면 GAN이 의미있는 학습을 하는지 확인하기 위해 실험을 진행했다. 이전 실험과 마찬가지로 32차원으로 확장시켰고 WGAN과 [2]에서 제시한 R1 정규화를 사용해 학습을 진행했다. 학습후 GAN이 생성한 32차원 페이크데이터의 feature 간의 관계를 시각화 했다. 또한, GAN의 훈련데이터의 feature간의 관계도 시각화 한 후, 두 결과를 비교했다. 이는 그림4 과 같다.

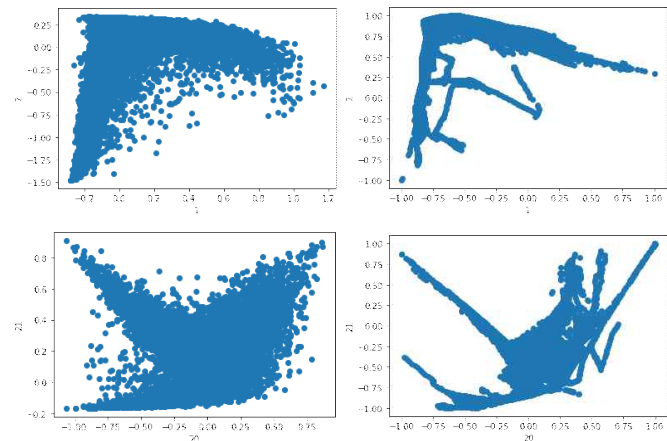


그림 4. 32차원 페이크데이터의 feature간의 관계(왼쪽)와 32차원 리얼데이터의 feature간의 관계(오른쪽): x,y축의 숫자 i는 데이터의 i번째 feature을 의미한다.

나머지 feature간의 관계도 페이크데이터의 것과 리얼데이터의 것이 비슷하다는 걸 확인할 수 있다. 하지만, 정밀하고 복잡한 관계는 잘 묘사하지 못한다. 학습을 마친 GAN이 생성한 20000개의 32차원 페이크데이터를 Decoder에 입력한 결과는 그림5의 왼쪽과 같다. 그림5의 오른쪽과 비슷한 모양이 나타나지만, 마찬가지로 정확한 묘사는 못하고 있다. 모델을 더 복잡하게 만들고 하이퍼파라미터를 잘 튜닝한다면 더욱 정밀한 결과를 얻으리라 기대한다.

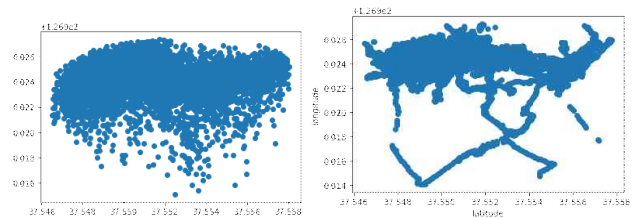


그림5. GAN이 생성한 32차원 페이크데이터 20000개를 Decoder에 입력한 결과(왼쪽), 원본데이터(위도,경도)를 AE에 입력했을 때 출력값(오른쪽)

#### 4. 결론

일반적으로 분포가 복잡한 저차원 데이터는 GAN에 학습시키기가 어렵다고 알려져있다. 본 논문은 AE를 이용해 고차원으로 변환하여 GAN을 학습하였고, 실험 결과 학습이 보다 안정되고, 의미있는 학습이 진행됨을 확인했다. 모델구조, 하이퍼파라미터 등을 잘 조절해 더 정밀한 결과를 얻을 수 있다면 저차원 데이터를 GAN에 학습시키는 일반적인 방법이 될 수 있으리라 기대한다.

#### 5. 참고문헌

- [1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David WardeFarley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc., 2014
- [2] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In International Conference on Machine Learning, pp. 3478–3487, 2018.
- [3] Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. A Non-Parametric Generative Model for Human Trajectories. In Proceedings of the 27<sup>th</sup> International Joint Conference on Artificial Intelligence IJCAI. pp. 3812–3817. 2018.
- [4] Kramer, Mark A. Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal. 37 (2): 233–243. 1991.
- [5] Martin Arjovsky, Soumith Chintala, and Leon Bottou. Wasserstein generative adversarial networks. In ICML, pp. 214–223, 2017.
- [6] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In Advances in Neural Information Processing Systems 30. pp. 5769–5779, 2017.