Generative Adversarial Nets 분석과 적용사례

이준환, *유지상
광운대학교, *광운대학교
tarje3@kw.ac.kr, *jsyoo@kw.ac.kr

Generative Adversarial Nets Analysis and Applications

JunHwan Lee *Jisang Yoo
Kwangwoon University *Kwangwoon University

요 약

2014 년 Ian Goodfellow 가 발표한 한편의 논문은 머신러닝 분야에 새로운 방향을 제시하였다. Generative Adversarial Networks, 일명 GAN 이라 불리는 이 논문은 이전까지 딥러닝으로 하지못했던 새로운 것을 창조해내는 작업을 하는 첫번째 딥러닝 알고리즘이다. 이전까지는 딥러닝을 통해 영상에서 객체의 종류를 판단하는 Classification 문제나, 영상에서 특정 객체를 검출하여 위치를 찾는 Object detection, 영상 내 특정 객체만 분리해내는 Image segmentation 문제를 해결하고 있었다. GAN 의 등장으로, 다양한 방면에서 GAN 을 적용하여 기존에는 하지 못했던 새로운 분야에 딥러닝을 적용한 사례들이 등장하고 있다. 본 논문에서는 GAN 의 원리 분석과 GAN 을 응용하여 여러 분야에 적용한 사례들을 살펴보고자 한다.

1. 서론

최근 굉장히 주목을 받고있는 논문인 Generative Adversarial Networks(GAN)에 관한 이야기를 해보고자 한다[1]. GAN 은 Ian Goodfellow 가 2014 년에 제안한 논문으로 한국어로 번역하면 “대립쌍 구조를 사용하는 생성 모델” 이라고 할 수 있다. 서로 대립적인 관계에 있는 두개의 네트워크를 구성하여 서로 대립하는 과정에서 훈련 타겟을 생성하는 방법을 알도록 학습시키는 구조이다. 논문의 저자인 Ian Goodfellow 는 몬트리올대학교 졸업후 Google Brain 으로 자리를 옮겼고 최근에는 비영리 인공지능 연구소인 OpenAI 에서 연구를 계속하고있다. 뉴욕대학교 교수이자 Facebook 의 AI research 팀의 Director 인 Deep learing 의 대가로 꼽히는 Yann LeCun 이 인터뷰에서 최근 등장한 딥러닝 기술을 묻는 질문에 GAN 이야말로 제일 중요한 연구가 될 것이라고 답변을 하였다. 이 연구는 지금까지 기존의 머신러닝 방법으로 해결하지 못하고 오직 인간만이 해낼 수 있다고 생각했던 창조하는 문제를 해결할 수 있는 가능성을 가지고 있다. 본 논문에서는 GAN 의 구조 분석과 GAN 의 적용사례들을 살펴보고자 한다.

2. Minimax two-player game

논문의 제목에서 알 수 있듯이 GAN 은 두개의 구조로 구성되어있다. 먼저 Generator 라고 부르는 생성기와, Discriminator 라고 부르는 판별기 이다. 논문에서 저자는 이해하기 쉬운 예시로 지폐위조범과 경찰을 예시로 든다. 생성기를 지폐위조범으로, 판별기를 경찰로 비유한다. 생성기는 위조 지폐를 계속해서 만들어 내고 판별기는 경찰의 역할인 위조 지폐의 진위여부를 가린다. 이러한 경쟁구조속에서 계속해서 반복하다 보면 지폐를 위조하는 능력과, 위조된 지폐를 판별하는 능력이 모두 개선되고 결과적으로는 위조지폐를 구별할 수 없는 정도가 되어 구별할 확률이 1/2 이 된다는 것이 논문의 주된 내용이다 [1].

GAN 구조를 좀더 상세하게 보면, Generator model G 는 우리가 갖고 있는 데이터 x 의 분포를 알아내려고 한다. 만약 G 가 정확한 데이터의 분포를 모사해낼 수 있다면, 거기서 뽑은 샘플은 데이터와 구별할 수 없다. 그리고 Discriminator model D 는 현재 자신이 보고있는 샘플이 학습데이터에서 온 진짜인지, 혹은 G 로부터 만들어진 가짜인지를 구별하여 0 부터 1 사이의 값으로 출력한다.

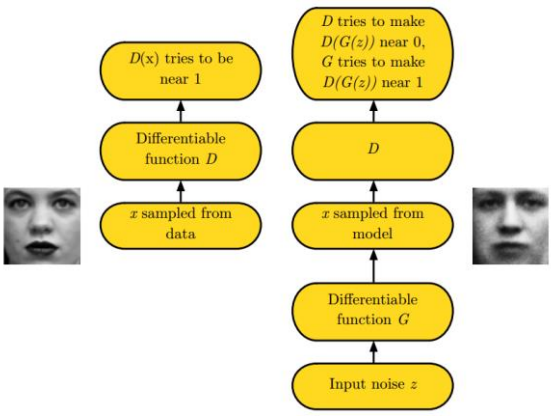


그림 1. Generator 와 Discriminator 의 학습과정[2]

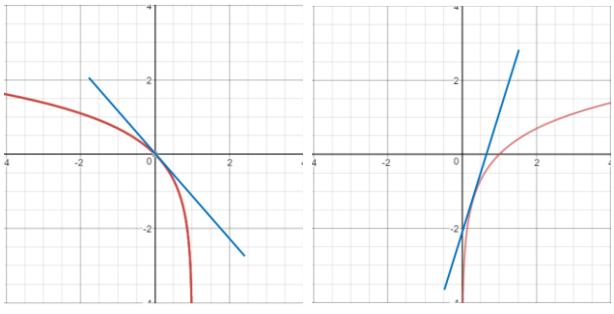
위 그림을 보고 설명하자면 D 의 입장에서는 데이터로부터 뽑은 샘플 x 는 D(x)=1 이 되고, G 에 임의의 noise distribution 으로부터 뽑은 input z 넣고 만들어진 샘플에 대해서는 D(G(z))=0 이 되도록 노력한다. 즉, D 는 실수할 확률을 낮추기(mini) 위해 노력하고 반대로 G 는 D 가 실수할 확률을 높이기(max)위해 노력하는데, 따라서 둘을 같이 놓고 보면 “minimax two-player game”이라 할 수 있다. 논문에서는 G 와 D 를 Multi-layer Perceptron(MLP)을 사용하고 랜덤 노이즈 z 를 사용한다. 하지만 꼭 신경망으로 만들 필요가 없으며 어떤 구조이든 서로의 역할을 잘 해줄 수 있다면 상관이 없다.

논문과 같이 G 와 D 모두 MLP 구조를 사용할 때 학습하는 방법을 식으로 표현하면 아래와 같이 V(G, D)에 대한 minimax problem 을 푸는것과 같아진다.

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (1)$$

위 수식에서 먼저 판별기 D(x)는 입력데이터가 실제데이터일 때 D(x)=1 이 되고 이때 첫번째 텀에서 log 값이 사라지게 되고 G(z)가 만들어낸 데이터라면 D(G(z))=0 이므로 두번째 텀 역시 0 으로 사라진다. 이때가 D 의 입장에서 V 가 최대값이다. G 의 입장에서 V 는 첫번째 텀과 독립적이고 V 를 최소화 시키려면 D(G(z))=1 을 만족시켜야 한다. 이는 G 가 가짜데이터가 아닌 진짜와 같은 실제 데이터샘플을 만들어낼 때를 의미한다.

논문에서 한가지 실용적인 팁을 언급하는데, 실제 학습을 할 때 V(D, G)를 식(1)을 그대로 사용하지 않고 수정하여 사용한다. 기존의 V(D, G)식에서 Log(1-D(G(z))) 부분을 G 에 대해 minimize 하는 대신 log(D(G(z)))를 maximize 하도록 G 를 학습시킨다. 이는 저자가 이론적인 동기가 아닌 실용적인 측면에서 적용을 하게 되었다고 언급하고 있다. 식을 수정하여 학습 시키는 이유는 학습 초기에 학습속도가 너무 느리기때문이다.



y=log(1-x) y=log(x)

그림 2. 식의 변형에 따른 gradient 비교

그림 2 에서 y=log(1-x) 그래프보다 y=log(x) 일 때 0 과 1 사이에서 gradient 가 더 높다. 학습 초기에는 G 가 실제 데이터와 차이가 많이 나는 영상을 생성하기 때문에 D 가 너무 쉽게 이를 실제 데이터와 구별하게 되고 따라서 log(1-D(G(z))) 값이 saturation 되어 gradient 가 아주 작은 값을 가지기 때문에 학습이 매우 느리다.

3. Extensions of GAN

앞장에서 GAN 의 이론적 원리와 어떻게 동작하는지를 살펴보았다. 이번장에서는 기존 알고리즘의 문제점을 해결하거나 향상된 새로운 GAN 모델과, GAN 을 사용한 다양한 방면에서의 적용사례들을 살펴보고자 한다. 개인적으로 가장 중요하고 의미 있는 모델은 Deep Convolutional GAN(DCGAN)[3] 이라고 생각한다. 기존 GAN 에서 사용하였던 Multi-Layer Perceptron(MLP)을 convolutional neural network 로 대체한 것으로서 영상에 특화되어 가장 좋은 성능을 보이는 구조이다. 2015 년에 DCGAN 논문이 나온 이후 영상을 생성하는 모델에서는 대부분의 논문이 DCGAN 을 기반으로 하고있다. DCGAN 의 구조가 갖는 의미를 정리해보자면 대부분의 상황에서 언제나 안정적으로 학습이 되는 장점이 있고 간단한 벡터 산술 연산을 통해 semantic sample generation 이 가능하다는 점 그리고 특정 filter 들이 이미지의 특정 물체를 학습했다는 것을 보여줄 수 있다는 점들이 있다. 먼저 기존의 안정적인 학습이 어려웠던 문제는 DCGAN 에서 굉장히 많은 횟수의 학습을 시도하여 기법 최적화를 통해서 어느정도 해결했다. DCGAN 모델의 몇가지 특징을 살펴보면 먼저 pooling layer 를 사용하지않고 1 보다 큰 stride 를 주어서 convolution 함으로서 영상의 크기를 줄이고 늘린다. 그리고 Batch Normalization 을 사용하고 fully connected layer 를 삭제한다. Generator 의 활성 함수는 ReLU(Rectifier Linear Unit)를 사용하되 최종 레이어에는 Tanh 함수를 사용한다. Discriminator 의 활성 함수는 모든 레이어에서 LeakyReLU(Leaky Rectifier Linear Unit)를 사용한다. 이러한 특징들은 모두 저자가 체계적인 계획과 반복적 학습을 통해 정한것으로서 정답은 아니지만 좋은 성능과 안정적인 학습을 보여준다[3].

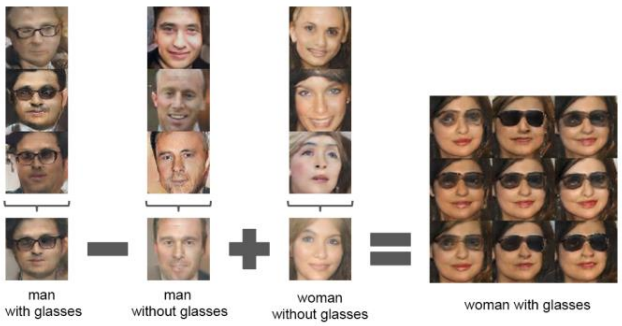


그림 3. 벡터 산술 연산을 통한 semantic sample generation 의 예

그림 3 은 앞서 언급한 간단한 벡터 산술 연산을 통한 semantic sample generation 의 예시를 보여준다. Generator 가 영상을 생성하기 위해 입력으로 필요한 latent code 라고 불리는 벡터를 더하고 빼는 연산을 통해 새로운 영상을 손쉽게 만들어 낼 수 있다. 벡터를 하나만 사용해서는 가시적인 효과를 보기

힘들어 3 개의 벡터를 평균을 취해 위와 같은 영상을 얻었다고 한다.

GAN 의 가장 큰 문제점으로는 학습이 안정적으로 되지 않는 문제가 있는데 이를 개선하려는 연구도 많이 진행 되었다. 안정적인 학습을 방해하는 Mode Collapse 문제를 개선하기위한 Unrolled GAN[4], InfoGAN[5] 그리고 Gradient Vanishing 문제를 개선한 WGAN[6], WGAN-GP[7] 등이 있다.

GAN 을 다양한 분야에 적용한 연구 또한 많다. 영상의 해상도를 향상시키는 Super-Resolution(초해상화) 기법에 GAN 을 적용한 SRGAN[8], 그림 5 과 같이 입력 영상을 다른 도메인으로 변환하여 출력하는 Image-to-Image Translation 기법[9]에 GAN 을 적용한 CycleGAN[10], 텍스트를 영상화하는 Text-to-Image 기법에 GAN 을 적용한 Text to Image Synthesis[11], Text-to-Image 기법에 GAN 을 한단계 더 쌓아 해상도를 증가시킨 StackGAN[12], 영상에서 가려진 영역을 채우는 Inpainting 기법에 GAN 을 적용한 Context Encoders[13], 랜덤 잡음을 입력으로 받아서 한 마디 마다 멜로디 시퀀스를 생성하는 MidiNet[14] 등이 있다.

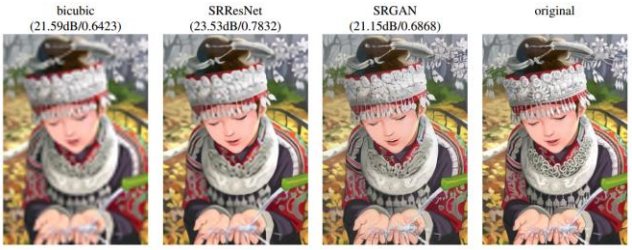


그림 4. Super-Resolution 기법을 적용한 초해상화의 예[8]

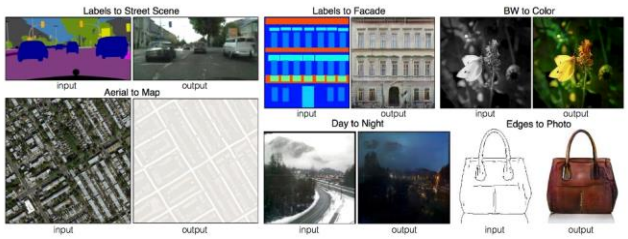


그림 5. 도메인을 변경하는 Image-to-Image Translation 의 예[9]

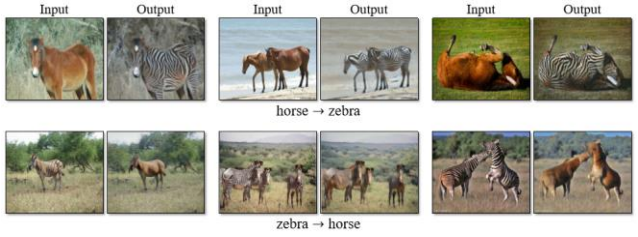


그림 6. 영상의 도메인을 변환하여 말을 얼룩말로, 얼룩말을 말로 변환하는 CycleGAN 의 예[10]

그림 7. 텍스트를 입력으로 받아 영상을 생성하고 GAN 을 한층 더 쌓아 해상도와 출력 영상의 품질을 향상시킨 StackGAN 의 예[12]

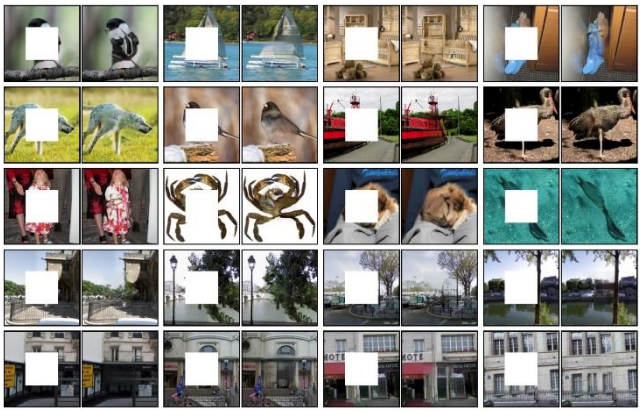


그림 8. 손상되거나 가려진 영역을 복원하는 Image Inpainting 의 예[13]

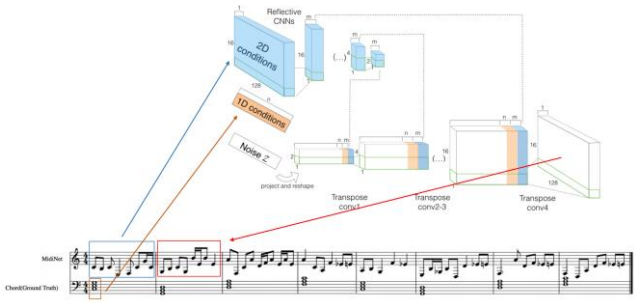


그림 9. 마디마다 멜로디 시퀀스를 생성하는 MidiNet 의 예[14]

4. 결론

위에서 GAN 을 적용한 다양한 사례를 알아보았다. 영상을 학습시켜 실제데이터와 유사한 것을 모방하여 만들어 내는 것으로 시작한 GAN 이 다양한 분야에 적용되고 응용되고 있다. GAN 에 관한 연구는 여기서 다 소개하지 못할 만큼 많고, 계속해서 다양한 분야에 활용하는 것이 시도 되고 있다. 현재 GAN 으로 생성한 이미지의 해상도가 낮고, 아직 완벽하지 못한 문제점이 있는데 더 많은 연구가 진행되어 이와 같은 문제점들을 해결한다면 더욱 다양한 분야에 쓰일 수 있을 것으로 기대되며, 아직 적용해보지 않은 새로운 분야에 GAN 을 적용함으로서 새로운 분야에 대한 확장도 기대해볼 수 있다.

ACKNOWLEDGEMENT

이 논문은 2016 년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2015-0-00258, 채널/객체 융합형 하이브리드 오디오 콘텐츠 제작 및 재생기술 개발)

5. 참고문헌

[1] Denton, E. L., Chintala, S., & Fergus, R. (2015). Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *Advances in neural information processing systems* (pp. 1486-1494).

[2] Goodfellow, I. (2016). NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv preprint arXiv:1701.00160*.

[3] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

[4] Metz, L., Poole, B., Pfau, D., & Sohl-Dickstein, J. (2016). Unrolled Generative Adversarial Networks. *arXiv preprint arXiv:1611.02163*.

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever: InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, 2016; [http://arxiv.org/abs/1606.03657 arXiv:1606.03657].

[6] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

[7] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved Training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*.

[8] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2016). Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*.

[9] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2016). Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*.

[10] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv preprint arXiv:1703.10593*.

[11] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016, May). Generative adversarial text to image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning* (Vol. 3).

[12] Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., & Metaxas, D. (2016). StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. *arXiv preprint arXiv:1612.03242*.

[13] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2536-2544).

[14] Yang, L. C., Chou, S. Y., & Yang, Y. H. (2017). MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation using 1D and 2D Conditions. *arXiv preprint arXiv:1703.10847*.