

머신러닝을 활용한 자동 채색 시스템 알고리즘 비교 분석¹⁾

이송은¹, 이지연², 김나현³, 김진환⁴

동국대학교 정보통신공학과

e-mail: ¹juliegogo200@naver.com, ²cjstkdmdm@naver.com,

³nellykim577@naver.com, ⁴wlsghks3109@naver.com

Comparison Analysis on Automatic Coloring System Algorithm Using Machine Learning

Lee song eun, Lee Ji Yeon, Kim Na Heon, Kim Jin Hwan
Department of Information and Communication Engineering, Dongguk
University

요 약

현재 머신러닝(Machine Learning) 기술은 기존의 머신러닝과 조합 및 변형 되어 조금 더 발전 된 형태로 연구되어지고 있다. 따라서 수많은 알고리즘이 개발되고 있는 시점이다. 본 연구는 최근 좋은 결과로 관심을 받고있는 GAN(Generative Adversarial Net)을 중심으로 IT기술의 머신러닝과 그림을 조합하여 자동채색을 목적으로 GAN 알고리즘을 비교하고 분석하고자 한다. GAN 알고리즘들 가운데서 ‘Conditional GAN’과 ‘Wasserstein GAN’을 사용하여 자동채색을 적용시켰고, 가장 부합한 알고리즘을 찾고 성능을 비교하여 어떠한 알고리즘이 ‘자동채색’ 목적에 더 부합한지 비교하고 판단 한다.

1. 서론

현재 IT기술의 발달로 급격하게 새로운 기술들이 생겨나고 발전하고 있다. 그 중 알파고의 영향으로 많은 IT기술의 중심이 머신러닝으로 대두되고 있다. 이러한 관심과 함께 새로운 알고리즘이 개발되고 있다. 다양한 알고리즘은 각각의 장단점을 가지고 있고, 어떤 주제에 활용할지에 따라 그 성능에도 차이가 있다. 본 논문에서는 자동 채색 시스템이라는 주제를 바탕으로 한다. 자동 채색 시스템을 Generative Adversarial Nets의 두 종류인 Conditional GAN과 Wasserstein GAN에 적용시켜, 결과를 비교해보고, 다양한 각도에서 성능을 비교 및 분석 해본다.

본 논문의 구성은 다음과 같다. 2장에서는 GAN과 이어서 파생된 알고리즘인 Conditional GAN과 Wasserstein GAN의 원리를 설명하고, 각 알고리즘의 구성 설명을 한다. 이어서 3장에서는 Convolutional Neural Network의 원리를 설명한다. 4장에서는 이 두 알고리즘을 사용하여 얻게 된 결과를 비교하고 5장에서는 결과에 대한 이유를 분석하여 자동 채색이라는 목적에 가장 알맞은 알고리즘을 선택한다. 마지막으로 6장에서는 결론을 맺는다.

2. GAN 알고리즘

1) 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음. 과제번호 (2016-0-00017)

2.1 GAN(Generative Adversarial Net)

GAN은 지도학습인 CNN(Convolution Neural Net)과는 달리 비지도 학습 방식을 따른다. 라벨링을 통해 데이터를 분류하여 학습하는 방식과 다르게 비지도 학습방식은 데이터 각각이 어떤 의미가 있는지 모른다. 하지만 데이터들 간의 상관 관계를 통해 군집화(Clustering) 또는 분포 추정(Underlying Probability Density Estimation)과정을 거쳐 학습하게 된다. GAN(Generative Adversarial Nets)은 크게 두 부분으로 나누어져 있다. GAN은 데이터를 만들어 내는 Generator와 이렇게 해서 만들어진 데이터를 분석하는 Discriminator가 서로 대립하면서 서로의 성능을 점차 개선해 나가는 부분으로 구성되어 있다. Generator는 입력 데이터의 분포를 알아내려는 목적이 있으며 만약 Generator가 정확히 데이터 분포를 구현할 수 있다면 Generator가 내놓은 샘플은 완벽히 데이터와 구별할 수 없다. 한편 Discriminator는 현재 자기가 분석하는 샘플이 학습용 데이터인 것인지 Generator로 부터 만들어 낸 샘플인지 구별하여 각각의 경우에 대한 확률을 추정한다.

2.2 cGAN(Conditional GAN)이란?

cGAN(Conditional Generative adversarial net)은 데이터 군집화에 따른 분류 알고리즘이다. cGAN은 Generator와 Discriminator로 구성되어있고, Generator는 노이즈가 추가된 데이터를 학습하고, 새로운 결과값을 생성하게 된다. Discriminator또한 진짜 데이터를 학습하고 Generator가

생성한 결과값을 실제 결과값과의 유사도를 비교하게 된다. cGAN은 기존의 GAN과 다르게 클래스 레이블을 조건으로 하여, Generator와 Discriminator 모두 이를 받아들여지게 된다. 이렇게 받아들인 클래스 레이블은 추가 입력 레이어와 합쳐지게 된다.

2.3 wGAN(Wasserstein GAN)이란?

wGAN은 데이터 확률분포의 수렴값에 따른 데이터 분류 알고리즘이다. wGAN에서는 Discriminator 역할을 Critic으로 데이터 확률 분포 및 분류 알고리즘을 구현한다. Critic 모델에서는 수렴값에 대한 정확한 측정 기준을 제시함으로써 학습 진행 상황을 유의미한 값으로 측정할 수 있으며 output의 성능을 높일 수 있다.

wGAN은 데이터의 확률분포 경계값을 계산한다. 이때 경계값에 극한값을 가지는 특정분포에 대해서 상한값과 하한값을 지정해 준다. 이를 지정해 줌으로써 모든 특정분포 집합을 상한값과 하한값으로 구할 수 있기 때문에 분포집합을 유의미한 값으로 측정 가능하다.

서로 다른 두 특정분포에 대한 가중치는 최적화 경로로 계산된다. (가중치의 감소) 각분포에 대한 특정 위치는 지정된 데이터 확률분포에 따른 상한값과 하한값의 평균값인 샘플로 설정했다. 이때 최적화 알고리즘으로 RMSProp을 사용하여 학습률을 분산시켜서 가중치의 수량에 대한 평균값을 구한다.

경로에 대한 가중치를 축소하게 되면 sample의 충돌문제를 개선하여서 cGAN보다 더 안정적인 학습이 가능해진다. 다만, 다양한 데이터 값들이 존재하기 때문에 최적화 경로로 출력물을 개선하려면 상당한 시간이 걸린다. 일반적으로 Discriminator보다 Critic은 더 많은 학습을 수반한다.

3. CNN(Convolutional Neural Network)

영상에서 특정위치에 있는 픽셀들은 그 주변에 있는 일부 픽셀들과 correlation이 높고 거리가 멀어질수록 그 영향은 감소하게 된다. 이러한 특징을 이용하여 CNN에서는 이미지를 해석하여 인식알고리즘을 사용할 경우 이미지 전체 픽셀을 같은 중요도로 처리하는 대신 특정 범위를 한정하여 처리하는 과정에서 효성성을 증가 시킨다. 또한 Local 정보를 활용하여 공간적으로 인접한 신호들에 대한 상관관계를 비선형 필터를 적용하여 추출하는 과정 거친다. 이때 필터를 여러 개 적용하게 되면 다양한 local 특징을 추출해 낼 수 있다.

Convolution layer 에서는 convolution feature를 추출하는 layer로 이미지에서 의미 있는 특징들을 추출한다. 이미지라는 데이터 타입의 특성상 많은 데이터를 요구하고 처리하게 되는데 이러한 feature를 줄이기 위해 subsampling 하는 과정을 거친다. 다음 convolution layer와 pooling layer에서 나온 feature들을 이용하여 분류하는 layer를 통해 마지막으로 학습하는 과정을 겪는다.

CNN 처리 과정은 단순하게 분류기로 구성 된 것이 아니라 특징을 추출하는 단계가 포함되어 있기 때문에, 원본 이미지에 대해 직접 작업이 가능하며, 기존 알고리즘과 달리 별도의 전처리 단계를 필요로 하지 않는다. 특징 추출과 topology invariance를 얻기 위해 필터와 sub-sampling을 거치며 이 과정을 반복적으로 수행하여 local feature로부터 global feature를 얻어 낸다.

4. 본 논문의 알고리즘

4.1 구성

실험을 위한 알고리즘은 다음과 같이 구성된다. 첫째, 비교를 위한 두개의 알고리즘은 공통적으로 GAN의 구성을 따르고 둘째, Generator의 세부적인 구성도는 <표 1>을 따른다. Discriminator는 Generator의 Layer 5까지의 구성을 한다. 이 때, activation function은 Generator와 Discriminator 모두 sigmoid를 사용한다.

Layer	Type	In - Channels	Out - Channels	Normalization	Activation
1	Conv.	256	128	BN	ReLU
2	Conv.	128	64	BN	ReLU
3	Conv.	64	32	BN	ReLU
4	Conv.	32	16	BN	ReLU
5	Conv.	16	8	BN	ReLU
6	Deconv.	8	16	BN	ReLU
7	Deconv.	16	32	BN	ReLU
8	Deconv.	32	64	BN	ReLU
9	Deconv.	64	128	BN	ReLU
10	Deconv.	128	256	BN	ReLU

<표 1> convolution 구성도 (Conv.: 2D convolution, Deconv. 2D convolution, BN: batch normalization. Layer 6, 4, 7과 Layer 3, 8과 Layer 2, 9와 Layer 1의 각각의 out channel은 residual connection을 적용한다.)

이를 바탕으로 구성된 wGAN과 cGAN 알고리즘을 구성하고, wGAN의 Critic은 optimizer를 RMSProp를 사용하고, cGAN의 Discriminator는 Adam Optimizer를 사용하여 학습 한다.

5. 결과

본 논문의 두가지 알고리즘 구성에 따라 각각 같은 데이터 셋을 이용하여 학습 시켰고 모델을 얻었다. 학습을 통해 얻은 모델에 그림1 (a) 이미지를 넣어 (b)와 (c)의 결과를 얻었다. CGAN의 결과인 원본과 비교하였을 때,



(그림 1) 채색 결과

색의 선명도는 적었지만 그림의 특징은 두드러지게 나타났다. 반면에 WGAN의 결과인 (c)의 경우, 테두리를 인식하지 못하였고 색의 번짐을 관찰할 수 있었다. 원본 이미지와 비교하였을 때, 색의 일치도가 떨어지는 것을 볼 수 있었다.

wGAN은 학습된 결과를 계속 확인하는 요소인 Critic 모델을 통해 안정적이고 안정된 결과를 얻게 되지만 이러한 과정에서 시간이 많이 필요하고 모델이 하나 더 추가되는 부분에서 Discriminator와의 연동 부에 있어서 변수들을 잘못 다룰 경우 학습이 안되는 경우가 빈번하게 발생할 수 있다. 상대적으로 cGAN은 wGAN에서 쓰이는 critic 모델이 없고, condition 설정은 critic 모델을 생성하는 것 보다는 시간이 절약되어 속도 면에서 학습시키기가 빠르고 쉽게 구현된다.

6 결론

본 연구를 위해 그림 채색을 학습시키는 과정에 있어서 cGAN의 학습 속도가 wGAN보다 뛰어났으며, 이는 Optimizer의 영향으로 있을 것이라 생각된다. wGAN의 참고 문헌에 따르면 RMSProp를 사용할 것을 권장하였다. Optimizer의 경우 Adam이 RMSProp보다 성능이 좋기 때문에 wGAN에도 Adam Optimizer를 적용시켜 보았지만 제대로 되지 않는 경우가 빈번하였고, 학습률과 momentum을 맞추는 것 또한 많은 시행을 거쳐야 했다.

그러므로 다양한 방면에서 고려하였을 때, 자동 색채 프

로그램에 적용시키는 것에 있어 적절한 것은 cGAN으로 판단된다.

참고문헌

- [1] Martin.A, Soumith.C, Leon.B, "Wasserstein GAN", Cornell University Library, arXiv:1701.07875v2, 2017.
- [2] Yagmur.G, Umut.G, Rob.van, Marcel.A.J, "Convolutional Sketch Inversion", Cornell University Library, arXiv:1606.03073v19, 2016.
- [3] Mehdi.M, Simon.O, "Conditional Generative Adversarial Nets", Cornell University Library, arXiv:1411.1784, 2014.
- [4] Kvfrans, "Outline Colorization through Tandem Adversarial Networks", Cornell University Library, arXiv:1704.08834v1, 2017.