Introduction

GAN의 주요한 문제점으로 학습의 불안정성(generator가 터무니 없는 이미지를 생성하는 등),

학습하는 과정을 시각화해서 볼 수 없다는 점이 존재했습니다.

- We propose and evaluate a set of constraints on the architectural topology of Convolutional GANs that make them stable to train in most settings. We name this class of architectures Deep Convolutional GANs (DCGAN)
- We use the trained discriminators for image classification tasks, showing competitive performance with other unsupervised algorithms.
- We visualize the filters learnt by GANs and empirically show that specific filters have learned to draw specific objects.

그래서 이 논문에서는

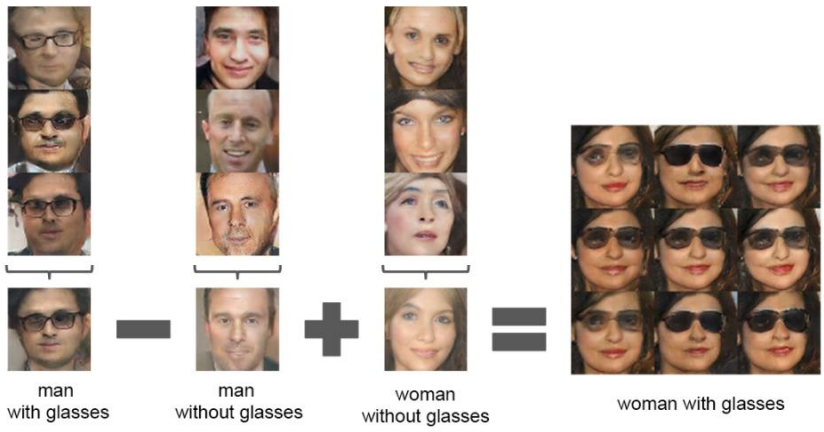
1. cnn을 이용해서 gan모델을 만들 때 안정적으로 학습이 될 수 있는 구조를 알아내고, 이를 "Deep Convolutional GANs (DCGAN)" 이라고 합니다.

2. 해당 모델로 학습한 discriminator는 이미지 분류를 다른 비지도 학습 모델들과 비슷한 수준으로 해낼 수 있었습니다.

3. 필터가 어떻게 특정 물체를 학습했는지 시각화해서 관찰할 수 있었습니다.

4. generator의 noise vector를 이용해서 semantic한 계산이 가능했습니다.

아래와 같은 작업이 가능하다는 것이죠.



APPROACH AND MODEL ARCHITECTURE

이전에는 CNN을 이용한 GAN모델이 성공적이지 않았습니다.

저자는 CNN모델에서 좋다고 증명된 방법 3가지를 적용시켰다고 합니다.

Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

1. 풀링 레이어대신 discriminator에서는 strided convolution으로, generator에서는 fractional-strided convolution 을 사용했습니다. Max pooling과 같은 결정론적 공간적 풀링 대신에 자연스럽게 공간적으로 다운샘플링할 수 있도록 한 것입니다.

2. batch normalization을 사용했습니다. 모든 레이어에서 배치정규화를 사용했더니 sample oscilliation과 모델이 불안정해졌기에 generator의 아웃풋 레이어와 discriminator의 인풋 레이어는 제외했습니다.

3. fully connected layer를 없앴습니다. 대신에 global average pooling을 사용했습니다.

4. ReLU함수를 generator의 output layer빼고 적용한다.

5. leaky ReLU함수를 discriminator의 모든 레이어에서 사용한다.

이렇게 적용시 더 빠르게 saturate 된다고 합니다.

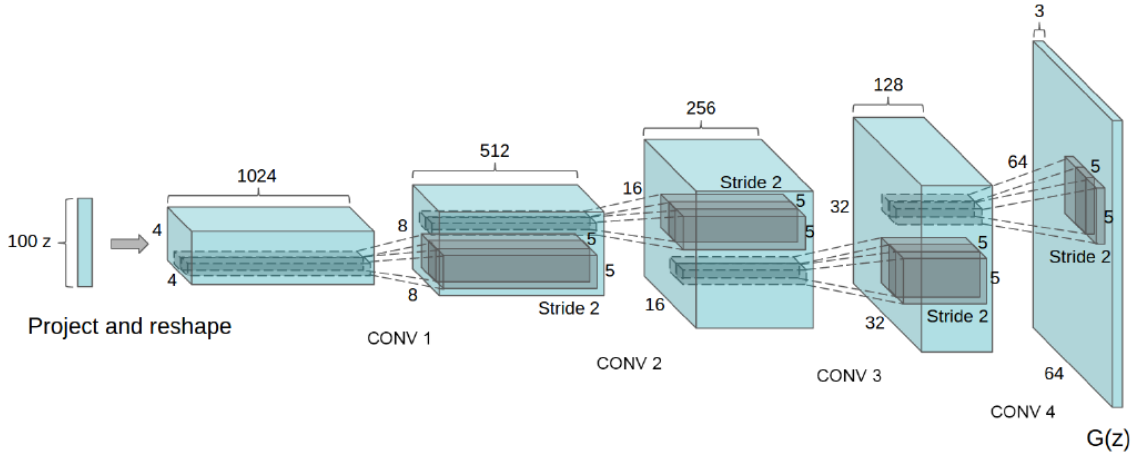
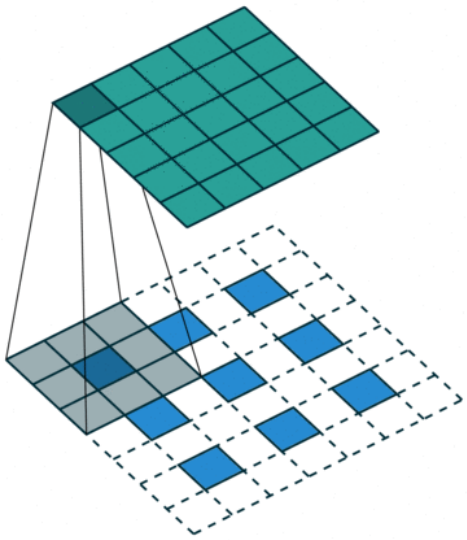


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

Generator는 위와 같은 구조를 가집니다. 100 차원을 가지는 z가 input으로 들어가고, 4개의 fractionally strided convolution을 통과하게 됩니다.(transpose convolution이라고도 합니다. 일반적인 컨볼루션연산과 다르게 연산 후 피쳐맵이 더 커집니다.) 그리고 최종적으로 64*64크기를 가지는 이미지가 완성됩니다.



fractionally strided convolution

CLASSIFYING CIFAR-10 USING GANS AS A FEATURE EXTRACTOR

비지도 학습이 잘됐는지 판단하는 좋은 방법은 feature extrctor로서 지도학습을 해서 성능을 보는

것입니다.

CIFAR-10데이터는 원래 K-means 기법이 정확도가 좋다고 알려져있는데 DCGAN은 이보다 더 적은 피쳐맵 크기를 가지면서더 좋은 성능을 냈습니다.

Table 1: CIFAR-10 classification results using our pre-trained model. Our DCGAN is not pre-trained on CIFAR-10, but on Imagenet-1k, and the features are used to classify CIFAR-10 images.

Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% (±0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% (±0.7%)	3200
View Invariant K-means	81.9%	72.6% (±0.7%)	6400
Exemplar CNN	84.3%	77.4% (±0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% (±0.4%)	512

WALKING IN THE LATENT SPACE

데이터를 단순히 기억해서 이미지를 생성하는지 판별하기위해서 latent space의 매니폴드가 중간 중간 끊기는 지점이 없이 자연스럽게 구성이 되었나 확인한 결과는 다음과 같았습니다.

Figure 4: Top rows: Interpolation between a series of 9 random points in Z show that the space learned has smooth transitions, with every image in the space plausibly looking like a bedroom. In the 6th row, you see a room without a window slowly transforming into a room with a giant window. In the 10th row, you see what appears to be a TV slowly being transformed into a window.

위의 두 그림에서 창문이 없다가 자연스럽게 생겨나는 것과, tv가 없다 자연스럽게 생겨나는 것을 볼 수 있습니다.

Vector arithmetic on face samples

vector("King") - vector("Man") + vector("Woman") 과 같은 연산이 z를 이용해서 할 수 있을 지 살펴봤다고 합니다.

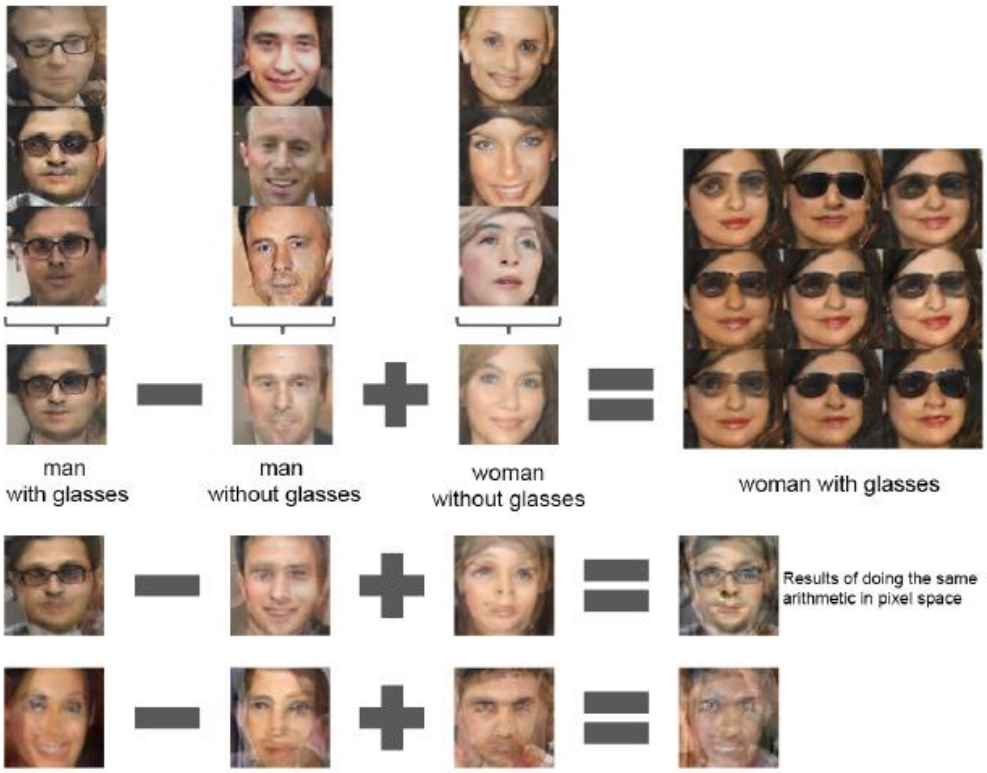


Figure 7: Vector arithmetic for visual concepts. For each column, the Z vectors of samples are averaged. Arithmetic was then performed on the mean vectors creating a new vector Y. The center sample on the right hand side is produce by feeding Y as input to the generator. To demonstrate the interpolation capabilities of the generator, uniform noise sampled with scale +-0.25 was added to Y to produce the 8 other samples. Applying arithmetic in the input space (bottom two examples) results in noisy overlap due to misalignment.

위의 사진에서 '안경 쓴 남자', '안경 안 쓴 남자', '안경 안 쓴 여자'를 그리게 하는 각각의 z의 평균치를 구해서 계산해서 나온 z를 generator에 넣어주면 위와 같은 결과를 볼 수 있습니다.

CONCLUSION AND FUTURE WORK

이 연구를 통해 GAN이 지도학습, 생성모델링을 위한 이미지 표현을 잘 학습할 수 있다는 것을 증명했습니다. 하지만 약간의 모델 불안정성이 있기 때문에 이를 보완하는 것이 필요하다고 합니다. 또한 이 모델을 비디오나 오디오에 적용하는 것도 흥미로울 것 같다고 합니다.

총정리

- **대부분의 상황에서 언제나 안정적으로 학습이 되는** Convolutional GAN 구조(DCGAN)를 제안하였다는 점
- 마치 word2vec과 같이 DCGAN으로 학습된 Generator가 **벡터 산술 연산이 가능**한 성질을 갖고 이것으로 semantic 수준에서의 sample generation을 해볼 수 있다는 점
- DCGAN이 학습한 filter들을 시각화하여 보여주고 **특정 filter들이 이미지의 특정 물체를 학습했다는 것**을 보여주었다는 점
- 이렇게 학습된 Discriminator가 다른 비지도 학습 알고리즘들과 비교하여 비등한 이미지 분류 성능을 보였다는 점

출처: http://jaejunyoo.blogspot.com/2017/02/deep-convolutional-gan-dcgan-1.html

위와 같은 점에서 큰 의의가 있는 논문이라고 생각됩니다.

참고:

https://angrypark.github.io/generative%20models/paper%20review/DCGAN-paper-reading/

https://jgrammer.tistory.com/entry/%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0-DCGAN-2016

https://angrypark.github.io/generative%20models/paper%20review/DCGAN-paper-reading/