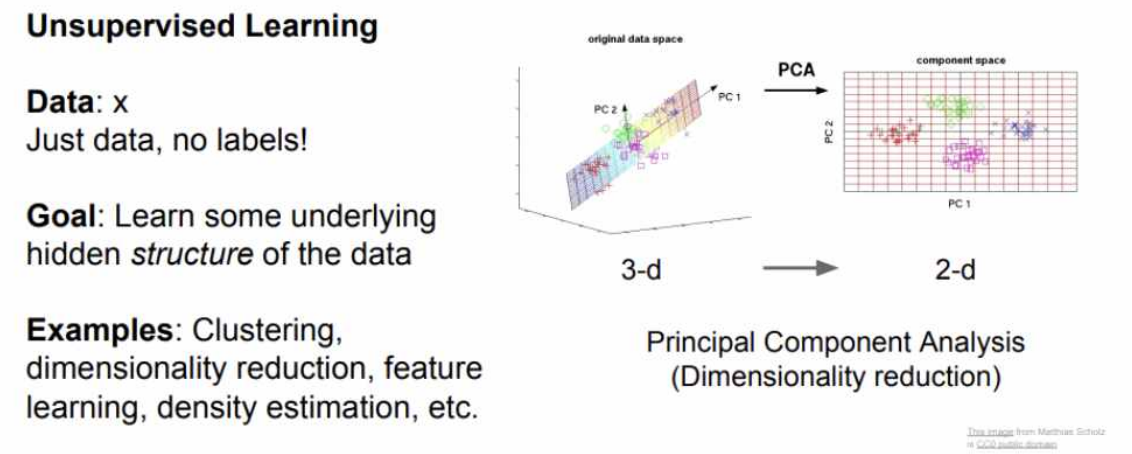
Lecture13: Generative Models

1. 비지도 학습(unsupervised learning)
지금까지는 다양한 지도학습(supervised learning) 문제를 다뤄왔다. 지도학습에는 데이터 X, 레이블 Y가 있었다. 지도 학습의 목적은 데이터 X를 레이블 Y에 매핑시키는 함수를 배우는 것이다. 지도 학습의 예로 classification, object detection, semantic segmentation, image captioning 등이 있었다.

반면에 비지도학습(unsupervised learning) 은 레이블 없이 학습 데이터만 가지고 데이터에 숨어있는 기본적인 구조를 학습시키는 것이다. 비지도학습의 예는 다음과 같다.

군집화(clustering)
군집화의 목표는 일정 metric을 가지고 유사한 데이터들끼리 묶어(group)주는 것이다.

차원 축소(dimensionality reduction)
차원 축소 문제에서는 학습 데이터가 가장 많이 퍼져있는 축을 찾아내는 것이다. 축을 곧 데이터에 숨어있는 구조의 일부분이라고 볼 수 있다. 이 방법은 데이터의 차원을 감소시키는데 사용할 수 있다.



위의 그림에 3차원 데이터가 있다. 이 데이터에서 두 개의 축을 찾아내 2차원으로 차원을 축소시킬 수 있다.(projection)

Autoencoder로 데이터의 feature representation을 학습
AE의 Loss는 입력 데이터를 얼마나 잘 재구성했는지를 나타내는데, 이를 이용해서 특징들을 학습시킬 수 있다. AE를 사용하면 레이블 없이도 feature representation을 학습시킬 수 있다.

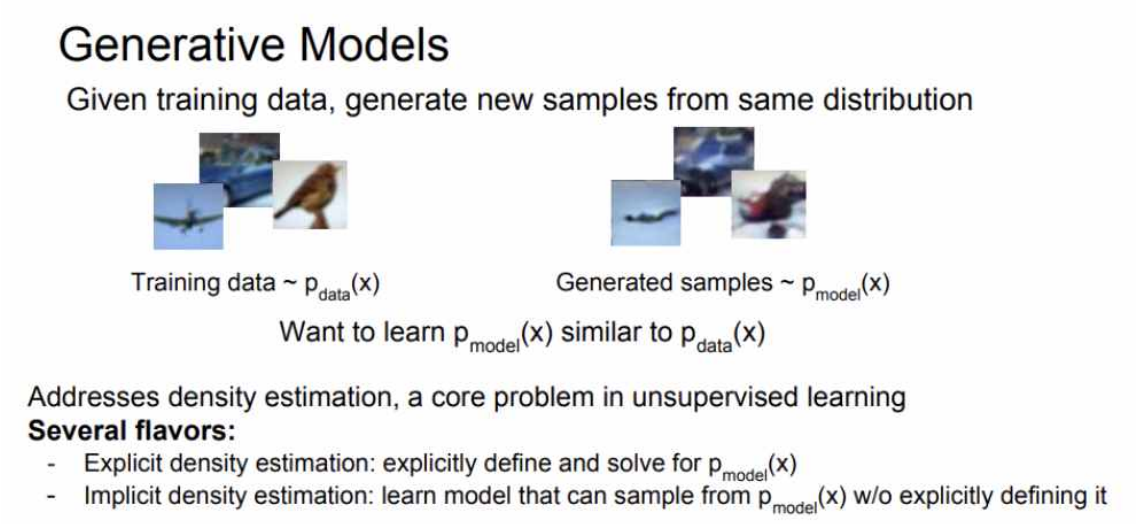
분포 추정(density estimation)
이는 데이터가 가진 기본적인(underlyting) 분포를 추정하는 방법이다.

- 지도/비지도 학습의 차이점을 요약하면, 지도학습의 경우에는 레이블을 통해 X에서 Y로 가능 함수 매핑을 학습한다. 비지도 학습의 경우에는 레이블이 없고 대신에 데이터의 숨겨진 구조를 학습한다. 군집화(grouping), 변화의 중심 축(axis of variation), 혹은 데이터의 밀도추정 등이 있다. 비지도 학습이 인기 있는 이유는 레이블이 필요 없어서 데이터를 아주 많이 모을 수 있다는 점과, 데이터에 대한 비용도 아주 적게 든다는 점 때문이다.

2. Generative models(생성 모델)

1) 생성모델의 개념

생성 모델은 비지도 학습의 일종으로, 생성 모델의 목적은 동일한 분포에서 새로운 샘플들을 생성해 내는 것이다.



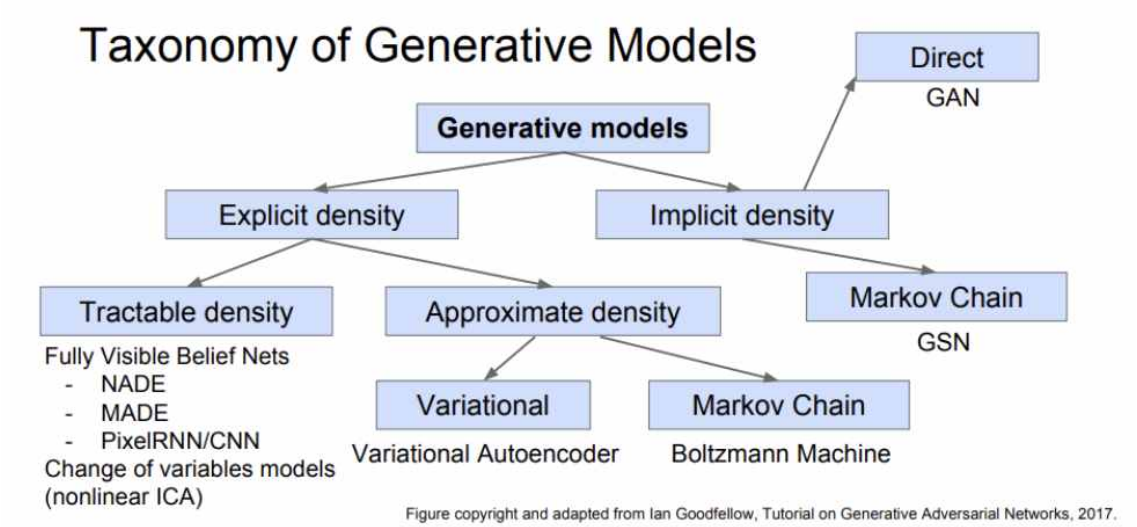
왼쪽 그림엔 분포 p_data로부터 나온 학습데이터가 있다. 우리가 원하는 것은 오른쪽의 p_model이 p_data와 같은 데이터를 생성하도록 학습시키는 것이다. 생성 모델에서는 "분포 추정" 다뤄야 한다. 즉 학습데이터의 근본이 되는 분포를 추정해야 한다. 분포 추정에는 몇 가지 전략이 있다.

생성 모델 p_model의 분포가 어떨지 명시적(explicitly) 으로 정의해 주는 방법
간접적(implicit)인 방법

- 왜 생성모델(generative models)이 중요할까?

생성모델은 비지도 학습에서 아주 중요한 문제이다. 데이터 분포로부터, 아주 사실적인(realistic) 샘플들을 생성해낼 수만 있다면 이를 이용하여 아주 많은 것들을 할 수 있다. 생성모델을 이미지에 적용하면 super resolution이나 colorization과 같은 테스크에 적용할 수 있다.

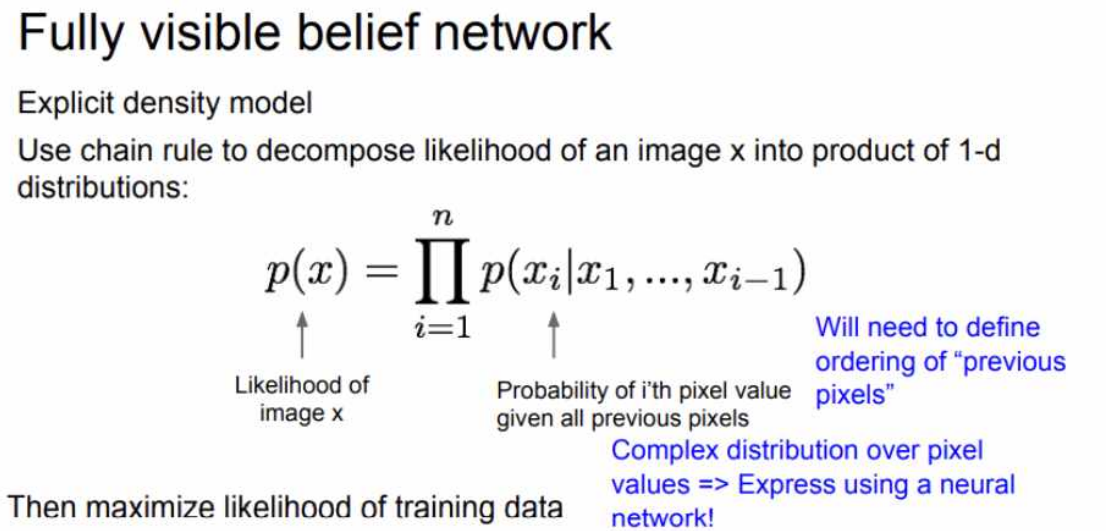
2) 생성모델의 종류



다양한 생성모델 중에서도, 현재 연구가 아주 활발하게 이루어지고 있는 세가지 모델만 알아볼 것이다. pixelRNN/CNN, variational autoencoders(VAE), generative adversarial networks(GAN)

3. pixelRNN/CNN

pixelRNN/CNN은 fully visible brief networks의 일종이다.



이미지 x에 대한 likelihood인 p(x)를 모델링한다. chain rule로 likelihood를 나타내는 p(x)를 1차원 분포들간의 곱의 형태로 분해(decompose)한다. 이렇게 분해하면, 픽셀 x_i에 대해서 각각 p(x_i | conditions)를 정의할 수 있다.

여기서 모델을 학습시키려면 학습 데이터의 likelihood를 최대화시키면 되는데, 픽셀 값에 대한 분포를 보면 아주 복잡하다.

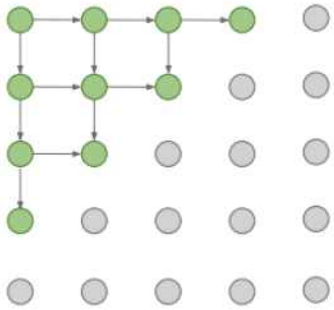
- 이미지 내의 각 픽셀들의 분포를 어떻게 알아낼까? 일단 복잡한 함수를 표현하기 위해서 신경망을 이용할 것이다.

PixelRNN *[van der Oord et al. 2016]*

Generate image pixels starting from corner

Dependency on previous pixels modeled using an RNN (LSTM)

Drawback: sequential generation is slow!



이 모델은 화살표 방향으로 순차적으로 픽셀을 생성해낸다. 이러한 방향성을 기반으로 한 픽셀들간의 종속성을 RNN(LSTM 모델)을 이용하여 모델링한다. 이런 방식으로 대각선 아래 방향으로 계속 내려가면서 픽셀들을 생성한다.
단, 한 가지 단점이 있는데, 순차적인 생성 방식으로 아주 느리다는 점이다.

PixelCNN *[van der Oord et al. 2016]*

Still generate image pixels starting from corner

Dependency on previous pixels now modeled using a CNN over context region

Training is faster than PixelRNN
(can parallelize convolutions since context region values known from training images)

Generation must still proceed sequentially
=> still slow

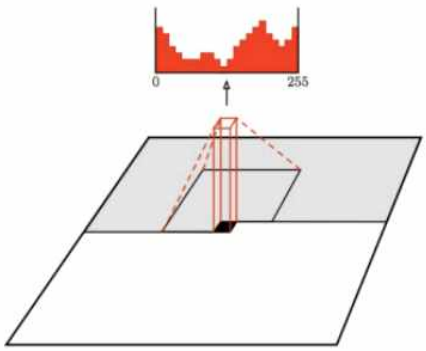
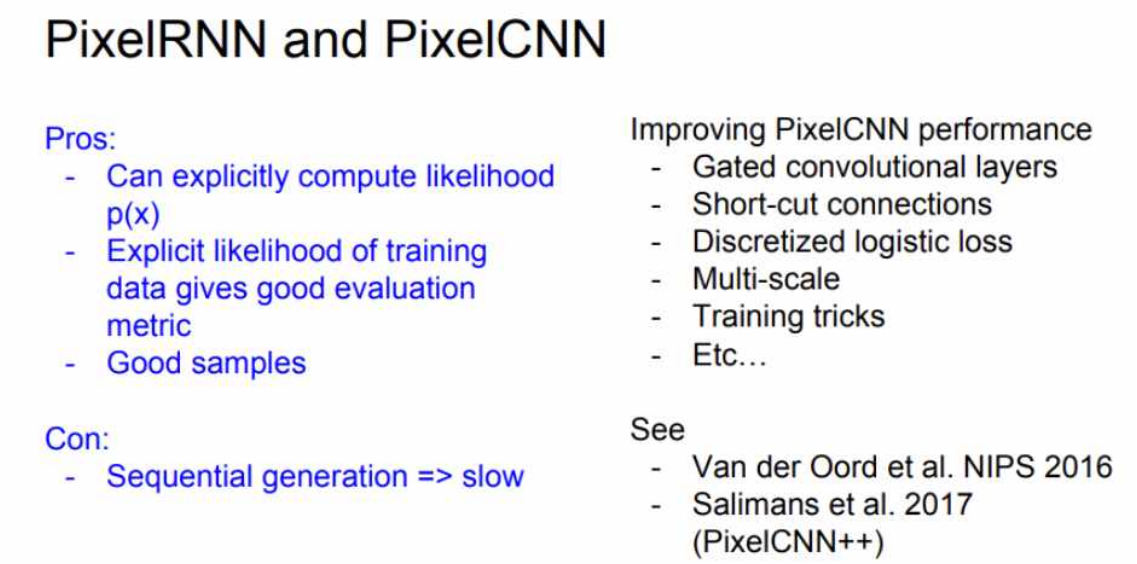


Figure copyright van der Oord et al., 2016. Reproduced with permission.

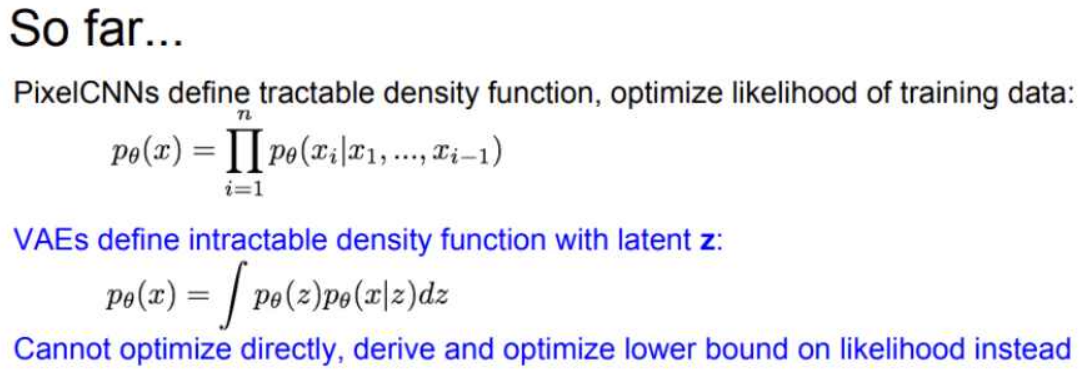
pixelCNN의 기본적인 문제 세팅 자체는 pixelRNN과 동일하다. 왼쪽 코너에서부터 새로운 이미지를 생성할 것이다. pixelRNN과 다른 점은 모든 종속성을 고려하여 모델링하는 RNN 대신에 pixelCNN은 CNN으로 모델링한다는 점이다.
그림의 회색 지역은 이미 생성된 픽셀들이고, 이들 중에서도 특정 영역만을 사용하여 다른 픽셀 값을 생성한다. PixelCNN에서는 각 픽셀에서 CNN을 수행한다. 픽셀을 생성하는 과정에서 각 픽셀 값은 정답 값(ground truth)을 가지고 있다. 이 정답값은 0-255 사이의 분류(classification) 문제를 풀기 위한 레이블이라고 볼 수 있다. 따라서 CNN에서는 출력 값을 가지고 softmax loss로 학습시킬 수 있다.



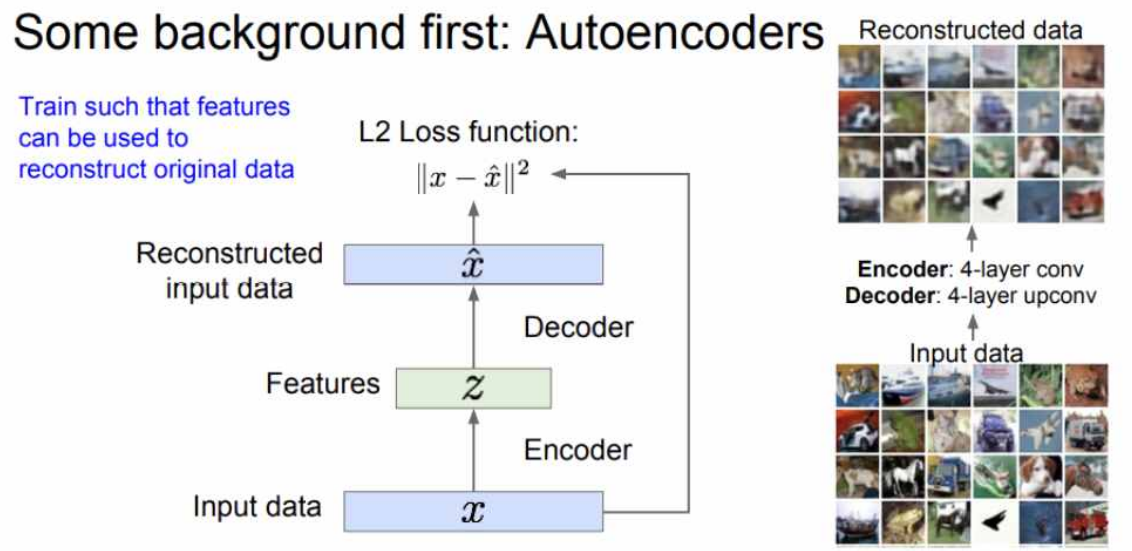
요약

pixelRNN/CNN은 likelihood p(x)를 명시적으로 계산하는 방법이다.
최적화시킬 수 있는 분포를 명시적으로 정의한다.
pixelRNN/CNN은 음성생성(audio generation)에도 사용될 수 있다.
"evaluation metic"이 존재한다는 장점이 있다.
생성 과정이 순차적이기 때문에 느리다는 단점이 있다.
pixelCNN을 사용하면 pixelRNN보다 학습이 더 빠르다. Train time에서는 (모든 픽셀들에 대해서) 학습 데이터의 likelihood를 최대화하는 것이기 때문이다.
하지만 새로운 이미지를 생성해야 하는 test time에서는 여전히 코너에서부터 시작해야 하고, 현재 픽셀을 생성하려면 이전 픽셀부터 순차적으로 처리해야 한다. 따라서 학습은 더 빨라졌어도 이미지를 생성하는데 걸리는 시간은 여전히 느리다.

4. Variational AutoEncoders(VAE)
1) AutoEncoder
지금까지 살펴본 pixelCNN은 "계산이 가능한 확률모델"을 기반으로 한다. 반면에, VAE의 경우에는 직접 계산이 불가능한(intractable) 확률 모델을 정의한다.



추가적인 잠재 변수(latent variable) z를 모델링할 것이다. VAE에서는 data likelihood p(x)가 적분의 형태를 띄고 있다. 하지만 이 식을 직접 최적화시킬 수는 없다. 대신에 likelihood(p(x))의 하안(lower bound)을 구해서(derive) 최적화시켜야만 한다.



AE는 레이블되지 않은 학습 데이터로부터 저차원의 feature representation을 학습하기 위한 비지도 학습 방법이다. 입력 데이터 x가 있고, 우리는 어떤 특징 "z"를 학습하길 원한다. encoder는 입력 데이터 x를 특징 z로 변환하는 매핑 함수의 역할을 한다. 일반적으로는 Neural network를 사용한다.
z는 x보다 작아서 AE를 통해 차원 축소의 효과를 기대할 수 있다. AE은 원본을 다시 복원(reconstruct)하는데 사용될 수 있는 특징들을 학습하는 방식을 취한다.

1. AE의 과정을 살펴보면, encoder는 x를 더 낮은 차원의 z로 매핑시킨다. z가 바로 encoder 네트워크의 출력이다.
2. 입력 데이터로부터 만들어진 특징 z는 두 번째 네트워크인 decoder에 사용된다. decoder의 출력은 입력 x과 동일한 차원이고, x와 유사하다. decoder는 기본적으로 encoder와 대칭적으로 동일한 구조를 지니고, CNN으로 구성된다.
CNN모델로 AE를 설계했을 때 encoder는 conv net으로 decoder는 upconv net으로 설계한다.
3. 복원된 이미지와 원본 이미지의 차이를 계산하기 위해서 L2 같은 loss 함수를 이용한다.

VAE를 요약하자면,

VAE는 autoencoders의 확률론적 변형 버전이다.
AE는 deterministic하게 x를 받아 z를 만들고 다시 x를 복원했다면 VAE는 데이터를 생성해내기 위해서 분포와 샘플링의 개념이 추가되었다. 그리고 계산할 수 없는(intractable) 분포를 다루기 위해서 하안(lower bound)를 계산했다. "variational"은 계산한 수 없는 형태를 계산

할 수 있도록 근사시키는 방법을 의미한다.(p(z given x) 를 계산못하니 q(z given x)로 근사)

VAE와 같은 접근방식의 이점은 생성 모델에 대한 원칙적 접근(principled approach) 방법이라는 점과 모델에서 q(z given x)를 추론한다는 점이다.
q(z given x)은 다른 테스크에서도 아주 유용한 feature representations이 될 수 있다.
likelihood의 하안(lower bound)을 계산한다는 점 때문에 pixelRNN/CNN 같이 직접 최적화하는 방법보다는 엄밀하지 않다.
GAN과 같은 다른 SOTA 생성모델에 비해서는 생성된 샘플이 블러하고(blurry), 퀄리티가 낮은 경향이 있다.

5. Generative Adversarial Networks (GAN)
GAN 요약

GANs은 특정 확률분포를 정의하지 않았다. (Implicit density)
two player game을 통해서 학습 데이터의 분포로부터 생성 모델을 학습시킨다.
GANs의 장점은 generator가 생성한 데이터의 퀄리티가 SOTA라는 것이다.
단점은 학습시키기 까다롭고 불안정하다는 것과 objective function을 직접적으로 최적화 하는게 아니라는 것이다.