Generative Models 1

- **Generation (sampling)**

  확률분포 $p(x)$와 유사한 어떤 새로운 x_{new}를 만들어서, 유사 이미지 만들어냄.

  - $x_{new} \sim p(x)$
- **Density estimation (anomaly detection)**

  어떤 이미지 x가 맞는지 아닌지 판별

  $p(x)$ 높으면 맞음 / 낮으면 아님

- Discriminator 모델 포함, 어떤 이미지가 특정 레이블에 속하는지 아닌지 판단
  - implicit model(암시적 모델): VAE, GAN처럼 생성만 하는 모델 (Generation만 함)
  - explicit model(명시적 모델): 확률값까지도 얻어낼 수 있는 모델 (Discriminator 포함)
- **Unsupervised representation learning (feature learning)**

  이미지의 공통적 특징(feature) 학습

- **Basic Discrete Distributions**
  - **Bernoulli distribution**(베르누이 분포): (biased) coin flip (1,0)

    $D = \{\text{Heads, Tails}\}$

    앞면 나올 확률: $P(X = \text{Heads}) = p$, 뒷면 나올 확률: $P(X = \text{tails}) = 1 - p$

    $X \sim \text{Ber}(p)$로 표현

    파라미터 1개 (p 하나)

  - **Categorical distribution**(카테고리 분포): (biased) m-sided dice

    $D = \{1, \ldots, m\}$

    주사위 던져 i가 나올 확률: $P(Y = i) = p_i$, $\sum_{i=1}^{m} p_i = 1$

    $Y \sim \text{Cat}(p_1, \ldots, p_m)$로 표현

    파라미터 $m - 1$개

- RGB 픽셀 하나, Joint distribution(결합분포) 모델링

  $(r,\ g,\ b) \sim p(R,\ G,\ B)$

  가능한 색상 종류(경우의 수): $256 \times 256 \times 256$개

  필요한 파라미터의 개수: $255 \times 255 \times 255$개

  => 파라미터 숫자 많음

- n개의 binary pixels(1개의 binary image) $X_1, \ldots, X_n$

  가능한 경우의 수:$2 \times \ldots \times 2 = 2^n$개

  필요한 파라미터의 개수: $2^n - 1$개

  => 파라미터 많이 필요

- **Structure Through Independence**

  모든 픽셀 $X_1, \ldots, X_n$가 각각 **독립적이라고 가정(independace assumption)**

  $p(x_!, .., x_n) = p(x_1)p(x_2)...p(x_n)$

가능한 경우의 수: 2^n개

$p(x_1, ..., x_n)$을 구하기 위해 필요한 파라미터 개수: n개

똑같이 2^n개의 경우의 수 나타내지만, 파라미터 n개만 필요

- **Conditional Independence**

  기존 Fully Dependant 모델링과 Independent 모델링의 중간점으로 타협

  - **Chain rule**

    $p(x_1, ... x_n) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)...p(x_n|x_1, ..., x_{n-1})$

  - **Bayes' rule**

    $p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(y|x)p(x)}{p(y)}$

  - **Conditional independence**

    If $x \perp y|z$, then $p(x|y, z) = p(x|z)$

    - Markov assumption 적용 시, 획기적으로 감소

- **parameters**

  - **Chain rule 이용 시,**

    $p(x_1, ... x_n) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)...p(x_n|x_1, ..., x_{n-1})$

    - 파라미터 개수

      $p(x_1)$: 1개

      $p(x_2|x_!)$: 2개 => $p(x_2|x_1 = 0)$, $p(x_2|x_1 = 1)$

      $p(x_3|x_1, x_2)$: 4개

      total: $1 + 2 + 2^2 + ... + 2^{n-1} = 2^n - 1$개

      => fully dependant 모델과 같음

      n개의 independent binary pixel 가진 이미지의 파라미터 수로도 해석 가능

  - **Markov assumption** 적용 시, 획기적으로 감소

    X_{i+1}는 X_i에만 dependant, $X_1, ..., X_{i-1}$까지는 independent으로 가정

    $X_{i+1} \perp X_1, ..., X_{i-1}|X_i$

    $p(x_1, ..., x_n) = p(x_1)p(x_2|x_1)p(x_3|x_2)...p(x_n|x_{n-1})$

    - 파라미터 개수: $2n - 1$개

      => **파라미터 개수 지수 차원에서 끌어내릴 수 있음**

- **Auto-regressive Model**, AR모델, 자기회귀모델

  하나의 정보가 이전의 정보에 dependant한 것

  Markov assumption처럼 직전 정보에만 dependant한 것 & $x_1, ..., x_{i-1}$까지 모두 dependant한 것

  Markov chain 활용 => 이전의 n개(AR-n model) / 1개(AR-1 model) 고려

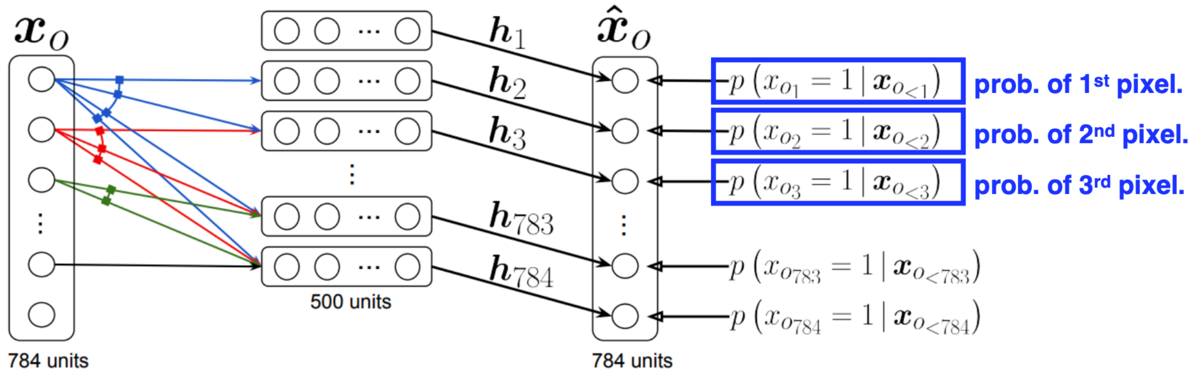
  chain rule로 joint distribution 나눔

  랜덤 변수들의 순서(order) 필요

  - 이미지(2차원)의 순서(1차원)을 어떻게 매기는가 => 성능 달라짐

- 어떤 식으로 conditional indepence 주는가 => 전체 모델의 structure 달라짐

- **NADE(Neural Autoregressive Density Estimator)**



$p(x_i|x_{1:i-1}) = \sigma(\alpha_i h_i)$ where $h_i = \sigma(W_{<i x_{1:i-1}} + c)$

i번째 픽셀을 $1, 2, ..., i$번째 픽셀까지에 dependent

매번 뉴럴넷 층으로 보냄 => 층의 사이즈 점차 커짐 => **weight 길이 가변적**

explicit model로, generation뿐만 아니라 **확률(density) 계산 가능**

$p(x_1, ..., x_{784}) = p(x_1)p(x_2|x_1)...p(x_{784}|x_{1:783})$

- joint dist 이용해 $p(x_i|x_{1:i-1})$ 계산, 전체 확률 알 수 있음

  => 해당 이미지 판별: **discriminator 역할 수행 가능**

continuous variables(임의의 연속변수) 모델링 경우, 마지막 layer에 gaussian mixture model(가우시안 혼합) 활용, 연속 dist 생성 가능

- **Pixel RNN**

AR 모델을 만들기 위해 RNN 활용

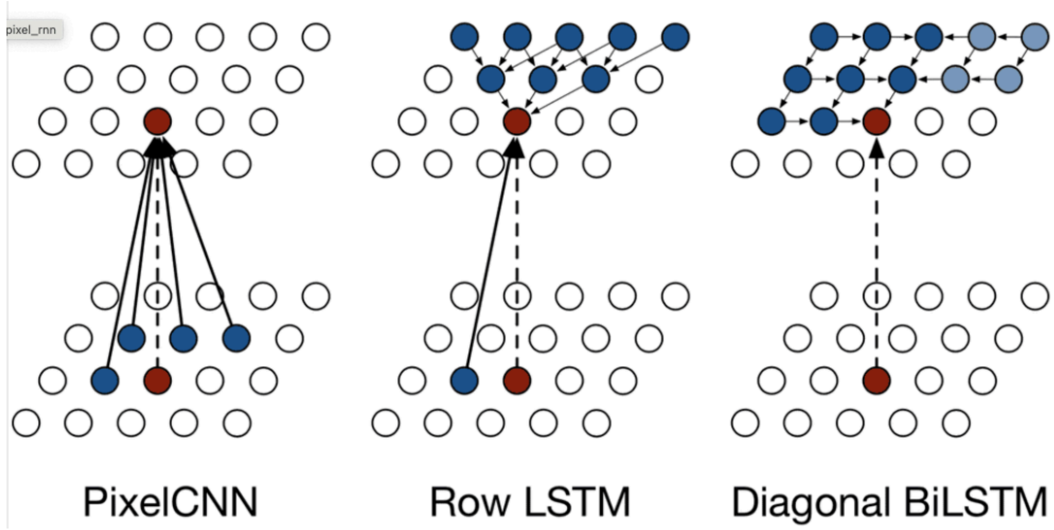
$p(x) = \prod_{i=1}^{n^2}$ (prob i번째 R) (prob i번째 G) (prob i번째 B)

$p(x) = \prod_{i=1}^{n^2} p(x_{i,R}|x_{<i})p(x_{i,G}|x_{<i}, x_{i,R})p(x_{i,B}|x_{<i}, x_{i,R}, x_{i,G})$

기존 모델: AR 모델을 **전열결계층(dense layer)** 가진 신경망

Pixel RNN: **RNN** 이용

- ordering 따른 종류



- **Row LSTM**

  i번째 픽셀을 만들 때, 위쪽 row와 바로 직전 픽셀 활용

- **Diagonal BiLSTM**

  i번째 픽셀을 만들 때 이전의 모든 픽셀 활용