<탄생배경>

Reviewer : 김정학

이 논문의 목표는 데이터의 본질의 특징을 잘 잡아내어 다른 일반적인 혹은 간단한 분포로 바꾸어낼 수 있는 모델을 만드는 것이다.

x가 원데이터고 h가 우리가 해석가능한 분포 $p_H(\cdot)$을 가진다고 하자

$x \xrightarrow{f} h$ 같이 f는 $h=f(x)$로 x을 h로 mapping 시켜주는 함수다. $p_H(h) = \prod_d p_{H_d}(h_d)$ 로 나타낼 수 있다. 이는 분포를 factorize 할 수 있다는 뜻이다. (MLP와 비슷한 느낌이라고 생각하면 된다) 중요한 점은 f는 invertible 하고 h의 차원은 x와 같다. 'change of variable rule'을 쓰면

① ... $p_x(x) = p_H(f(x)) \left| \det \frac{df(x)}{dx} \right|$ 를 얻을 수 있다. 후에 보겠지만 f는 $|\det|$를 구하기 쉽고 쉽게 inverse 되어야한다. 그리고 이 논문의 핵심 idea는 x을 두개로 쪼개는 거다. (x_1, x_2)로 찢고

$$\begin{cases} y_1 = x_1 \\ y_2 = x_2 + m(x_1) \end{cases}$$ 로 나타내는 것이다. m은 아무런 함수여도 된다 (뉴럴 네이어도 된다) 이렇게 나타내면 역으로도 쉽게 쓸 수 있다. (아래 같이)

$$\begin{cases} x_1 = y_1 \\ x_2 = y_2 - m(y_1) \end{cases}$$

<content>

① 식을 log를 취하면 (이러는 이유는 최대우도를 얻기위해서..)

$\log(p_x(x)) = \log(p_H(f(x))) + \log\left(\left|\det\left(\frac{df(x)}{dx}\right)\right|\right)$

p_H는 우리가 지정한 사전확률이다. $p_H(h) = \prod_d p_{H_d}(h_d)$ 라면

$\log(p_x(x)) = \sum_{d=1}^{D} \left(\log(p_{H_d}(f_d(x))) + \log\left(\left|\det\left(\frac{df(x)}{dx}\right)\right|\right)\right)$ 로 나타낼 수 있다.

데이터를 대충히 축소해서 학습이 잘 일어날 수 있지만 자코비안 항 때문에

축소하는 것을 지양하고 고안로 지역을 떠나간다.

① triangular structure

det(·) 부분이 tractable하고 계산이 쉽도록 하는 종류를 찾아야 하는데 이 논문이 제안한 건 triangular matrics다. 왜냐하면 det(·)가 diagonal element를 곱하는 걸로 간단히 구할 수 있고 inverting 하는 것도 계산이 간단하다. (square matrix M = LU로 구할 수 있다. L은 lower triangular, R은 upper triangular)

② Coupling layer

이 논문에서 가장 기초가 되는 layer다.

$x \sim R^D$ 이고 이를 (x_{I_1}, x_{I_2})로 쪼갤 수 있다.

($I_1 \to 1 \sim d$ 까지 $I_2 \to d \sim D$ 까지를 나타냄.)

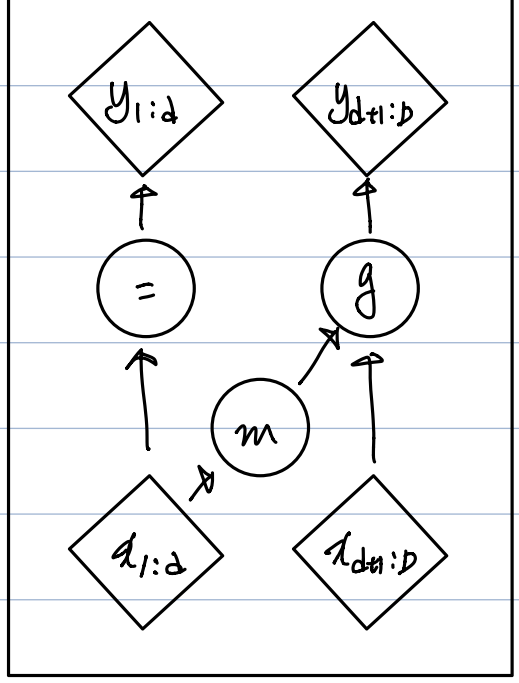
$$\begin{cases} y_{I_1} = x_{I_1} \\ y_{I_2} = g(x_{I_2}; m(x_{I_1})) \end{cases}$$ 로 나타낼 수 있다. g는 그냥 invertible 한 함수다.

($R^{D-d} \times m(R^d) \to R^{D-d}$를 만족하면 된다)

그러면 $\frac{dy}{dx} = \begin{bmatrix} I_d & 0 \\ \frac{dy_{I_2}}{dx_{I_1}} & \frac{dy_{I_2}}{dx_{I_2}} \end{bmatrix}$ 인 triangular matrix를 얻을 수 있다. 역으로 나타낼 수 있다.

$$\begin{cases} x_{I_1} = y_{I_1} \\ x_{I_2} = g^{-1}(y_{I_2}; m(y_{I_1})) \end{cases}$$

이 layer를 복잡하게 연결만하면 된다. '복잡하게'라는 게 인풋을 나누는 것과 각 layer마다 나누는 순간을 다르게 설정하면 그만이다. (계속 똑같이 나누면 x_{I_2}은 바뀌지 않기 때문)



그림으로 나타내면 왼쪽과 같다.

③ Allowing rescaling

각 layer에서 자코비안 항이 있다. 그 항상도 결국 자코비안 항이 필요할 것인데 이는 volume preserving의 특성을 가지고 있다. 문제 이를 깰 수 있는데 대각행렬 S를 추가하는 것이다. 이를 통해 어떤 차원은 가중치를 더 주고 덜 주고 할 수 있다. 또 식도 다음과 같이 바뀐다.

$$\log(p_X(x)) = \sum_{j=1}^{D} \left[\log(p_{H_j}(f_j(x))) + \log(|S_{jj}|) \right]$$