

CS236n : Lecture 2

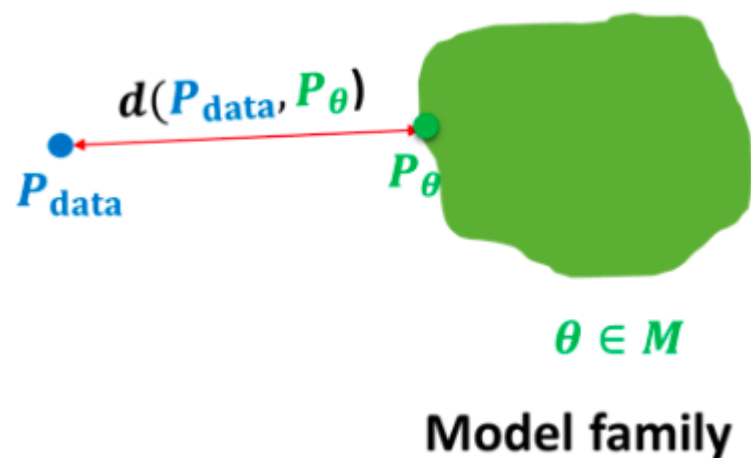
1. Learning Generative models

We are given a training set of examples, e.g., images of dogs



우리는 주어진 샘플로부터 $P(x)$ 를 모델링하게 되는데,
이렇게 잘 모델링된 $P(x)$ 는 다음 3가지에 이용이 될 수 있다.

- 생성
- 밀도 추정 (밀도 추정을 통해 $P(x)$ 값이 낮다면 해당 분포에는 잘 맞지 않는 샘플 → Anomaly Detection)
- 비지도 기반 표현 학습 (Unsupervised 기반 환경에서 이미지들간의 공통된 시각적 특징을 포착하게 해줌)



→ 파라미터로 조정가능한 확률분포 $P(x)$ 의 파라미터에 따른 모든 가능한 경우의 집합 중(Model Family), 가장 샘플 데이터와의 간극이 적은 (교안에서는 이 간극을 $d(p_{\text{data}}, p_{\theta})$ 로 표현. 이 값이 가장 작은) 확률 분포 $P(x)$ 의 파라미터를 찾아야함

2. Example of Joint Distribution

Discrete한 확률 변수들이 구성되는 공통 확률분포에서의 파라미터들은 각 사건(state)에 대한 확률인데 MNIST IMAGE인 간단한 시나리오만 하더라도 공통확률 분포를 모델링 하기 위해 엄청난 파라미터 수를 필요로함.



- Suppose X_1, \dots, X_n are binary (Bernoulli) random variables, i.e., $\text{Val}(X_i) = \{0, 1\} = \{\text{Black}, \text{White}\}$.
- How many possible images (states)?

$$\underbrace{2 \times 2 \times \dots \times 2}_{n \text{ times}} = 2^n$$

- Sampling from $p(x_1, \dots, x_n)$ generates an image
- How many parameters to specify the joint distribution $p(x_1, \dots, x_n)$ over n binary pixels?

$$2^n - 1$$

각 픽셀에 대해 가지는 값을 X_i 라는 확률 변수로 두면 $\text{Val}(X_i) = \{0, 1\} = \{\text{흰}, \text{검}\}$

2^{n-1} 개

(2^n 개의 state가 발생할 수 있고, 각 state가 발생할 확률의 총합은 1이라는 것을 고려하면 2^{n-1} 개의 확률 값이 파라미터임)

공통 확률분포를 정의하기 위한 Parameter가.... 너무 많아 $\pi\pi$

줄일 수 있는 방법이 없을까?

→ 확률 변수간의 관계에 대한 가정을 하자!!!

**조건부 분포와 공통확률분포, 주변 확률 개념 : <https://excelsior-cjh.tistory.com/193>

3. Structure Through Independence

- If X_1, \dots, X_n are independent, then

$$p(x_1, \dots, x_n) = p(x_1)p(x_2) \cdots p(x_n)$$

- 확률 변수들이 서로 독립이다 -> 2^{n-1} 에서 n 개로 줄일 수 있음(여전히 각 확률변수가 0 또는 1의 값만 가질 수 있다고 가정했을 때) 그러나 이 가정은 말이 안됨. 이미지의 각 픽셀들이 서로 연관성이 없다는 것을 가정하게 되므로...



두가지 Rule을 집고 넘어가자!

- ① **Chain rule** Let S_1, \dots, S_n be events, $p(S_i) > 0$.

$$p(S_1 \cap S_2 \cap \dots \cap S_n) = p(S_1)p(S_2 | S_1) \dots p(S_n | S_1 \cap \dots \cap S_{n-1})$$

- ② **Bayes' rule** Let S_1, S_2 be events, $p(S_1) > 0$ and $p(S_2) > 0$.

$$p(S_1 | S_2) = \frac{p(S_1 \cap S_2)}{p(S_2)} = \frac{p(S_2 | S_1)p(S_1)}{p(S_2)}$$

Chain_Rule : <https://chatgpt.com/share/68888482-0874-8008-a0d2-972c0127f130>

- Using Chain Rule Autoregressive Model에 사용되는 방식

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

각각의 확률 변수가 가질 수 있는 값이 {0,1} 밖에 없을 때의 상황임

- How many parameters? $1 + 2 + \dots + 2^{n-1} = 2^n - 1$
 - $p(x_1)$ requires 1 parameter
 - $p(x_2 | x_1 = 0)$ requires 1 parameter, $p(x_2 | x_1 = 1)$ requires 1 parameter
Total 2 parameters.

- Chain rule에서는 확률변수 간의 어떤 가정을 내린 것은 아님 (그냥 joint distribution을 conditional distribution들의 곱으로 나타낸 것임). 원래 joint parameter에서 가지는 총 state의 수로 파라미터 개수를 가져감.
- 다만 **특별한 가정**을 추가하여 더 단순화 할 수 있음 → 파라미터 수 감소

4. Conditional Independence를 가정(특별한 가정)해서 파라미터 수를 줄이자!

- $2^n - 1$ is still exponential, chain rule does not buy us anything.
- Now suppose $X_{i+1} \perp X_1, \dots, X_{i-1} | X_i$, then

$$\begin{aligned} p(x_1, \dots, x_n) &= p(x_1)p(x_2 | x_1)p(x_3 | \cancel{x_1}, x_2) \dots p(x_n | \cancel{x_1, \dots, x_{i-1}}, x_{i-1}) \\ &= p(x_1)p(x_2 | x_1)p(x_3 | x_2) \dots p(x_n | x_{n-1}) \end{aligned}$$

- How many parameters? $2n - 1$. Exponential reduction!

→ 조건부 독립 : 어떤 확률 변수가 주어졌을 때의 독립성을 표현. 위에서 X_i 가 주어졌을 때, X_{i+1} 은 $X_1 \sim X_{i-1}$ 까지의 확률 변수와 독립이다.

5. Bayesian network

공통 확률분포를 표현하려는 목표를 가지는 네트워크. 그러나 Bayesian Network는 사전 지식을 반영하여 확률 변수 간의 조건부 독립성(어떤 변수 X_i 는 자신의 **부모 노드(Parents)**에만 조건부로 의존하고, 부모가 주어지면 나머지 변수들과는 조건부 독립이라고 가정합니다.)을 도입함으로써 이 공통 확률을 구성하는 파라미터의 수를 줄인다. 또한 조건부 독립성을 가정한 확률변수간의 관계를 DAG로 표현한 것이 BN.

- **Bayesian networks** given by (G, P) where P is specified as a set of local **conditional probability distributions** associated with G 's nodes

- 노드 : 확률 변수
- 노드 값 : 확률 변수에 대한 conditional distribution (given parents)
- 링크 : Direct influence from the parent to the child ($A \rightarrow B$: A가 B에 영향을 줌)
Indirect influence도 존재!

*DAG

- Directed (방향성 있음): 간선(edge)에 방향이 있어서, 노드 A에서 B로 가는 건 가능하지만 B에서 A로는 (동일 간선 기준) 갈 수 없음. →

조건부 의존성을 반영 ($A \rightarrow B$ 이면 A가 B에 영향을 주는 것임)

- Acyclic (순환 없음): 어떤 노드에서 출발해서 간선을 따라가다 보면 다시 자기 자신으로 돌아오는 경로가 없음. 즉, 사이클이 없음.

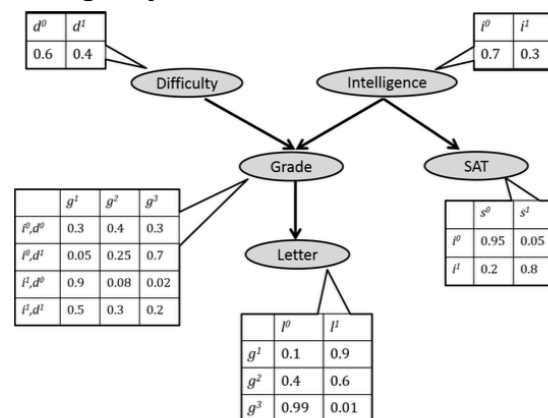
베이지안 네트워크에서의 Joint Probability Factorization

Defines a joint distribution:

$$p(x_1, \dots, x_n) = \prod_{i \in V} p(x_i \mid \mathbf{x}_{\text{Pa}(i)})$$

- ex) DAG

- Consider the following Bayesian network:



- What is its joint distribution?

$$p(x_1, \dots, x_n) = \prod_{i \in V} p(x_i \mid \mathbf{x}_{\text{Pa}(i)})$$

$$p(d, i, g, s, l) = p(d)p(i)p(g \mid i, d)p(s \mid i)p(l \mid g)$$

6. Bayesian Network에서의 샘플링 순서 (Topological Ordering)

Can sample from the joint by sampling from the CPDs according to the DAG ordering

Can identify some conditional independence properties by looking at graph properties

→ Bayesian Network로 Joint Probability를 정의했을 때, Joint Probability에 대한 Sampling은 DAG 구조 ordering에 따라 sampling 하면 됨.

→ Bayesian Network를 통해 Conditional Independence를 확인할 수 있음(Bayesian Ball Theory 참고)

7. Summary of Bayesian Network

→ 베이지안 네트워크도 역시 공통 확률분포를 표현하려는 목표를 가진다. 그러나 Bayesian Network는 사전 지식을 반영하여 확률 변수 간의 조건부 독립성(어떤 변수 X_i 는 자신의 **부모 노드(Parents)**에만 조건부로 의존하고, 부모가 주어지면 나머지 변수들과는 조건부 독립이라고 가정합니다.)을 도입함으로써 이 공통확률을 구성하는 파라미터의 수를 줄인다.

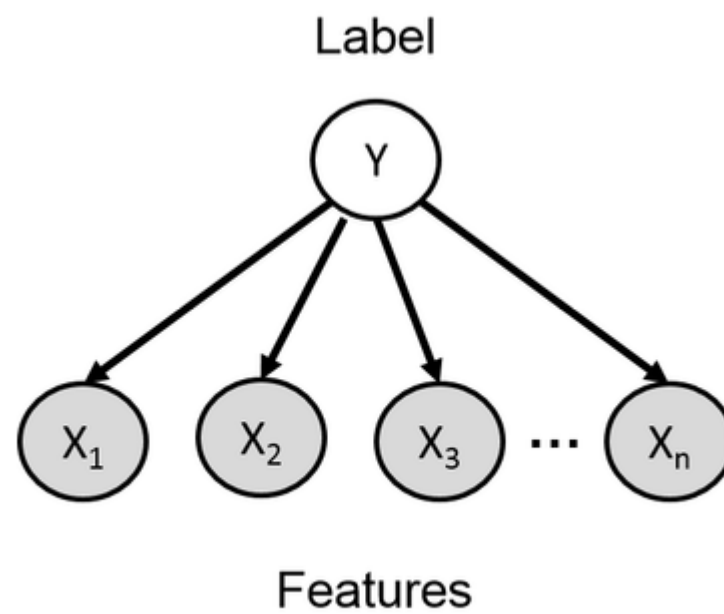
→ 베이지안 네트워크는 이 확률 변수들 간의 조건부 의존 관계가 DAG에 링크로 표현되어 있음. 이때 링크는 조건부 의존 관계 $A \rightarrow B$ 로의 영향을 나타냄.(=A가 B에 영향을 준다, A가 B의 직접적인 원인이 된다.)

→ 그 DAG의 parent→child 순서를 따라 차례대로 확률 변수를 생성하는 생성 모델이라고도 볼 수 있음.

→ 또한 이런 Bayesian Network의 DAG구조를 통해 우리는 확률 변수간의 조건부 독립성 또한 판단할 수 있습니다. (Bayesian Ball Theorem)

KAIST Supplementary Materials

8. Naive Bayes Model



→ Naive Bayes도 역시 공통 확률분포 $P(\text{features}, \text{Label})$ 을 모델링.

→ Naive Bayes에서 가정하는 확률변수간의 관계를 Bayesian Network로 표현하면 위와 같다.

두가지 특징

- Label이 각각의 Feature에 영향을 주는 구조 (화살표 $y \rightarrow x_1, y \rightarrow x_2, y \rightarrow x_3 \dots$)
- Conditional Independence, Label이 주어졌을 때, 각각의 Feature들은 서로 독립임.

그럼 Joint Probability는 다음과 같음

$$p(y, x_1, \dots, x_n) = p(y) \prod_{i=1}^n p(x_i | y)$$

Estimate parameters from training data. **Predict** with Bayes rule:

$$p(Y = 1 | x_1, \dots, x_n) = \frac{p(Y = 1) \prod_{i=1}^n p(x_i | Y = 1)}{\sum_{y \in \{0,1\}} p(Y = y) \prod_{i=1}^n p(x_i | Y = y)}$$

뒤에 설명하겠지만 Naive Bayes 방식은 Feature와 Label의 Joint Probability를 모델링하기에 학습데이터로부터 $P(Y)$ 와 $P(X|Y)$ 의 파라미터를 학습해야한다.

그후에 Bayes Rule을 통해 $P(Y|X)$ 를 $P(Y)$ 와 $P(X|Y)$ 로부터 예측한다 → 입력이 X 인데 Label 예측이 어떻게 돼?를 $P(x,y)$ 를 모델링하는 Generative Model에서도 수행가능함을 보여줌

9. Discriminative versus generative models

우선 내가 정리한 바는 아래와 같다.

✓ Generative Model (생성 모델)

- 목표: 데이터의 **joint probability** $P(X, Y)$ 를 모델링
- 방법: $P(Y|X)$ 를 구하려면 베이즈 정리를 통해 $P(X|Y)P(Y)$ 로 계산 가능

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

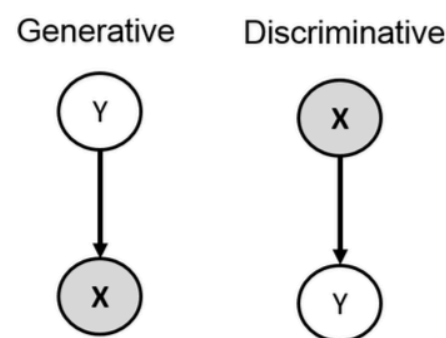
- 예시: Naive Bayes, Gaussian Mixture Model, Hidden Markov Model, Variational Autoencoder (VAE), Generative Adversarial Networks (GANs)*

(*GAN은 explicit probability를 모델링하진 않지만 생성 모델로 분류됨)

✓ Discriminative Model (판별 모델)

- 목표: 조건부 확률 $P(Y|X)$ 또는 결정 경계(direct mapping)를 모델링
- 방법: X 가 주어졌을 때 Y 를 어떻게 잘 맞출지를 직접 학습
- 예시: Logistic Regression, Support Vector Machine, Decision Tree, Neural Networks (대부분), Conditional Random Fields (CRFs)

- ▶ Using chain rule $p(Y, \mathbf{X}) = p(\mathbf{X} | Y)p(Y) = p(Y | \mathbf{X})p(\mathbf{X})$.
Corresponding Bayesian networks:



각각의 방식에서 X, Y의 관계성의 Bayesian Network로의 표현과 이에 따른 Joint Distribution Factorization은 위와 같다.

However, suppose all we need for prediction is $p(Y | \mathbf{X})$

In the left model, we need to specify/learn *both* $p(Y)$ and $p(\mathbf{X} | Y)$, then compute $p(Y | \mathbf{X})$ via Bayes rule

In the right model, it suffices to estimate just the **conditional distribution** $p(Y | \mathbf{X})$

- We never need to model/learn/use $p(\mathbf{X})$!
- Called a **discriminative** model because it is only useful for discriminating Y 's label when given \mathbf{X}

일반적인 ML은 X가 주어졌을 때 Y가 뭔지를 알아내는 것이 Main Task이다.

→ $P(Y|X)$ 를 예측하는 것

이를 위해 GM에서는 $P(Y)$, $P(X|Y)$ 를 명시 혹은 학습해야하고, DM에서는 $P(Y|X)$ 만 명시 혹은 학습하면 된다.

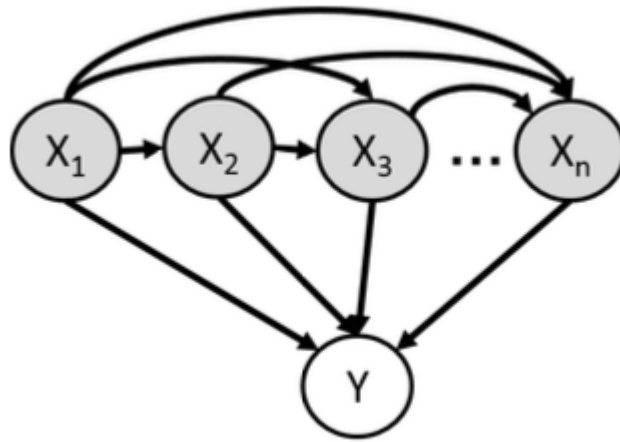
이로써, Discriminative Model에서는 Direct하게 $P(Y|X)$ 를 학습하고, Generative Model에서는 Bayes Rule을 이용해 $P(Y|X)$ 를 도출한다.

10. How To Train $P(Y|X)$ in DM? : Logistic Regression

$$p(Y = 1 | \mathbf{x}; \alpha) = f(\mathbf{x}, \alpha)$$

Logistic Regression은 다음과 같이 $P(Y=1|x)$ 를 α 를 파라미터로 가지는 Function으로 모델링하여 학습.

이외의 내용은 아는 것이니 생략.



→ Logistic Regression에서의 확률 변수 간의 관계를 나타내는 Bayesian Network

→ Naive Bayes와 같이 Given Y 일 때, Feature간의 독립성을 가정하지 않음, Naive Bayes와 달리 feature간의 관계를 고려한 의사결정을 할 수 있음.

11. Limit Of Discriminative Model and Strength of Generative Model

→ 그러나 Discriminative Model은 X 를 구성하는 모든 요소를 가지고 있어야 $P(Y|X)$ 를 구할 수 있음. $P(Y|X)$ 에서 X GIVEN 이니까.

→ 반면 Generative Model은 X 의 일부 X evidence(partly unseen variable)만 가지고 있어도 Marginalization을 통해 $P(Y|X)$ 를 구할 수 있음.

12. Advance in Discriminative Model → Non-linearity via Neural Network

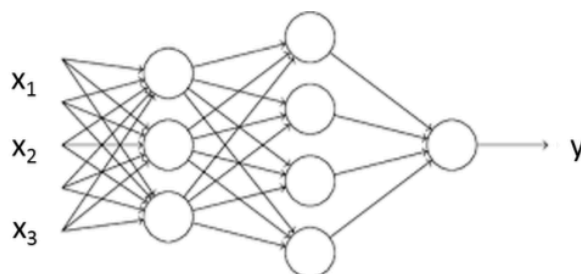
Logistic Regression은 non-linearity를 가정하지 않음(정확하게는 X 와 sigmoid 들어가기 전의 log-odds 간의 linearity를 가정하는 것임). $P(Y|X)$ 를 예측하는 $F(X; \alpha)$ 가 non-linearity를 가진다면 더욱 복잡한 관계를 capture할 수 있지 않을까?

→ 이 $F(X; \alpha)$ 를 Neural Network로 구현하자!

Non-linear dependence: let $\mathbf{h}(A, \mathbf{b}, \mathbf{x}) = f(A\mathbf{x} + \mathbf{b})$ be a non-linear transformation of the inputs (*features*).

$$p_{\text{Neural}}(Y = 1 | \mathbf{x}; \alpha, A, \mathbf{b}) = f(\alpha_0 + \sum_{i=1}^h \alpha_i h_i)$$

- More flexible
- More parameters: A, \mathbf{b}, α
- Can repeat multiple times to get a neural network



13. Comparison Between Bayesian Network & Neural Models

- Using Chain Rule

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2 | x_1)p(x_3 | x_1, x_2)p(x_4 | x_1, x_2, x_3)$$

Fully General

- 일반적인 Joint Probability를 Chain Rule을 통해 Factorization
- 확률 변수에 대한 가정은 없는 상태
- 파라미터 수가 매우 많음, 모든 state에 대한 확률값을 가지고 있어야함

Bayes Net

$$p(x_1, x_2, x_3, x_4) \approx p(x_1)p(x_2 | x_1)p(x_3 | \cancel{x_1}, x_2)p(x_4 | x_1, \cancel{x_2}, \cancel{x_3})$$

Assumes conditional independencies

Simplifying the conditions by dropping

- Conditional Independence를 가정함으로써 joint probability modeling에서의 파라미터 수를 줄일 수 있음

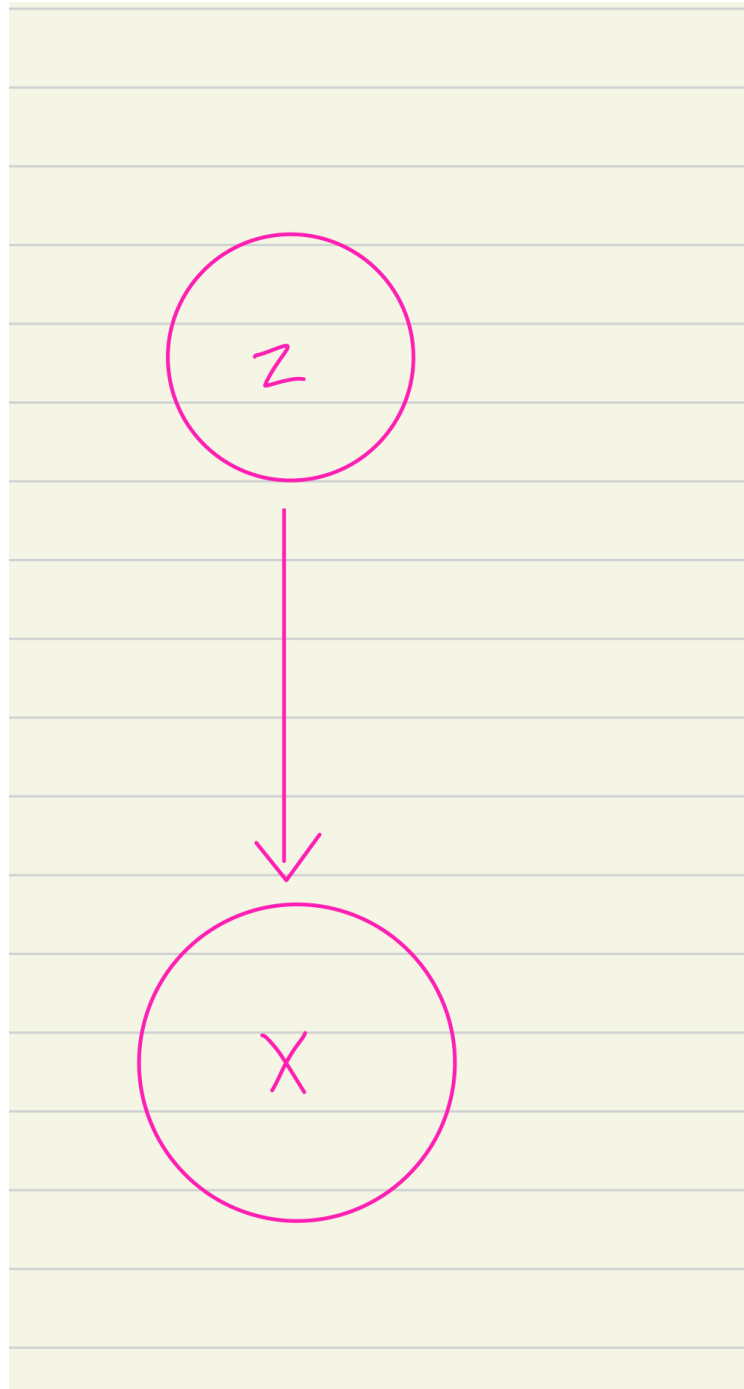
Neural Models

$$p(x_1, x_2, x_3, x_4) \approx p(x_1)p(x_2 | x_1)p_{\text{Neural}}(x_3 | x_1, x_2)p_{\text{Neural}}(x_4 | x_1, x_2, x_3)$$

- 맨 앞의 Chain Rule을 통한 Joint Distribution을 Neural Model을 통해 표현하겠다!
- Bayes Net과 달리 조건부 독립성을 명시적으로 가정하지 않음 (feature간의 관계를 반영)
- Neural Network가 확률 변수 간의 복잡한 의존 관계를 학습하게 됨
- 충분히 깊고 expressive한 네트워크라면 어떤 형태의 조건부 분포도 근사할 수 있음.

14. Generative Model with Latent Variable

잠재 변수 Z를 가지는 Generative Model에서 확률 변수 간의 조건부 의존 관계를 Bayes Network로 표현하면 아래와 같음



그럼 $P(z, x)$ 를 다음과 같이 Factorization 할 수 있음.

$$p_{Z,X}(z, x) = p_Z(z)p_{X|Z}(x | z)$$

Gaussian Mixture, GMM, VAE로 세가지 예시를 들었음

공통점)

- Joint Probability Factorization 방식, 확률 변수 간의 인과 관계(조건부 의존관계) : $z \rightarrow x$

차이점)

- Factorize된 각 분포를 명시 및 훈련하는 방식
- 이에 따라 구성하는 파라미터의 종류가 다름