LEC9 → VAE

1. Generative Model

Motivation: Density Estimation

Given $\mathcal{D} = \{(x_i)\}_{i=1}^{N}$, $x_i \sim p^*$ (GT)

Goal: find (estimate) $\hat{p} \approx p^*$ (parametric model)

$\Longrightarrow$ **Generative Model**:

Given $\mathcal{D} = \{(x_i)\}_{i=1}^{N}$ $x_i \sim p^*$ (GT)

Goal sample $x^{(new)} \sim p^*$ approximately

[Toy Example] GMM

① $z \sim$ Multi-bernoulli $(K) \in [K]$

② $x \sim$ Gaussian (μ_z, Σ_z)

$x \sim p^* \Rightarrow$ MLE to estimate $\begin{cases} \alpha_k \\ \begin{cases} \mu_k \\ \Sigma_k \end{cases} \end{cases}$

$\boxed{\text{EM Framework for hidden model}}$

2. Limitation of AE

① hidden space has no structure (not smooth w.r.t distribution of data points)

↓

we should give some structure to hidden space

⇓

example $\boxed{z \sim \mathcal{N}(0, I_K)}$

3. Latent Variable Model (GMM is one example)

$$P_\theta(x) = \int_z P_\theta(x \mid z)\, P_\theta(z)\, dz$$

$z \rightarrow$ Hidden (Latent) Variable

$$\begin{cases} P_\theta(z) : \text{latent variable prior} \\ P_\theta(x \mid z) : \text{conditional distribution} \end{cases}$$

Generating Process: model $\begin{cases} P_\theta(z) \\ P_\theta(x \mid z) \end{cases} \rightarrow$ Learn from $\mathcal{D}$

$\rightarrow$ Given $x^{(new)}$

$\rightarrow$ calculate $P_\theta(z \mid x^{(new)})$ from Bayes

$\rightarrow$ $z^{(new)} \sim \boxed{P_\theta(z \mid x^{(new)})}$ $\rightarrow$ intractable might be

$\rightarrow$ $x' \sim P_\theta(x \mid z^{(new)})$

approximate with $q_\phi(\cdot)$
(Variational Inference)

4. VAE setting

$$\begin{cases} ① \; z \sim \text{Gaussian}(\underline{0}, I_K) \\ ② \; x \mid z \sim \text{Gaussian}(\mu_{dec}(z;\theta),\; 1\,\cancel{\sigma^2_{dec}}(z;\theta)\, I_D) \end{cases}$$

(also can use Bernoulli model) $\rightarrow$ $\boxed{\text{CE}}$

$\rightarrow$ $\boxed{\text{Square Loss}}$

Note: (X, Z) is not Gaussian any more

$\Rightarrow$ untractable $z \mid x$!

Therefore, approximate $z \mid x$ via Varational Inference

$$\boxed{z \mid x \sim \text{Gaussian}(f_\phi(x) ; g_\phi(x)^2 I_K)}$$ Approximately

$\Rightarrow$ $z \mid x = f_\phi(x) + g_\phi^2(x) \circ \varepsilon$ Here $\varepsilon \sim \mathcal{N}(0, I_K)$

$\boxed{\text{reparameterize trick}}$

Question : How to learn parameters ?

① $\xrightarrow{\text{MLE}}$ $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \log P(X|\theta) \Rightarrow \boxed{\text{log-likelihood } \mathcal{L}(\theta)}$

$$= \underset{\theta}{\operatorname{arg\,max}} \sum_{i=1}^{n} \log \int_z P(x_i|z) P(z)\, dz$$

(untractable)

Decomposition $L(\theta)$:

$$\mathcal{L}(\theta) = \log P(X|\theta)$$

$$= \mathbb{E}_{z\sim q}[\log P(x|\theta)]$$

$$= \mathbb{E}_{z\sim q_\phi}\left[\log \frac{P(x,z|\theta)}{P(z|X,\theta)}\right]$$

$$= \mathbb{E}_{z\sim q_\phi}\left[\log \frac{P(x,z|\theta)}{q_\phi(z)}\right] - \mathbb{E}_{z\sim q_\phi}\left[\log \frac{P(z|X,\theta)}{q_\phi(z)}\right]$$

$$= \underline{\text{ELBO}} + D_{KL}(q_\phi(\cdot) \,|\, P(\cdot|X;\theta))$$

$$\text{ELBO} := \mathbb{E}_{z\sim q_\phi}\left[\log \frac{P(x,z|\theta)}{q_\phi(z)}\right]$$

② $\xrightarrow{\text{Maximize ELBO}}$ $(\hat{\theta}, \hat{\phi}) = \underset{\theta,\phi}{\operatorname{arg\,max}} \text{ELBO}(\theta, \phi)$

$$= \underset{\theta,\phi}{\operatorname{argmax}} \mathbb{E}_{z\sim q_\phi(\cdot)}\left[\log \frac{P(X,z|\theta)}{q_\phi(z)}\right]$$

$$= \underset{\theta,\phi}{\operatorname{argmax}} \mathbb{E}_{z\sim q_\phi(\cdot)}[\log P(X|z;\theta)]$$

$$+ \mathbb{E}_{z\sim q_\phi(\cdot)}\left[\frac{P(z|\theta)}{q_\phi(z)}\right]$$

$$= \underset{\theta,\phi}{\operatorname{arg\,max}} \mathbb{E}_{z\sim q_\phi(\cdot)}[\log P(X|z;\theta)] - D_{KL}(q_\phi(\cdot) \,\|\, P(\cdot|\theta))$$

$$= \underset{\theta,\phi}{\arg\min} \; \underbrace{- \mathbb{E}_{z \sim q_\phi(\cdot|x)} \left[\tfrac{1}{2} \| x - f(z) \|_2^2 \right]}_{\substack{\text{(decoder)} \\ \text{reconstruction loss}}} \rightarrow \boxed{\text{require sample}}$$

$$+ \underbrace{D_{KL} \left(q_\phi(\cdot|x) \,\|\, p(\cdot|\theta) \right)}_{\substack{\text{(encoder)} \\ \text{regularization term}}} \rightarrow \boxed{\text{closed form}}$$

Reparameterize trick :

$$\mathbb{E}_{z \sim q_\phi(\cdot|x)} [\cdot]$$

$$= \mathbb{E}_{u \sim N(0, I_k)} [\cdot]$$

$$\rightarrow \underline{\underline{z = f_\phi(x) + g_\phi^2(x) \circ u}}$$

③ We will have $\underline{q_\phi(\cdot|x) \approx p(z|x)} \rightarrow \boxed{\text{encoder part}}$

Reason : $\underline{\text{given } \theta}$, $\underline{P(x|\theta) \text{ is fixed}}$

$$\log P(x|\theta) = ELBO + D_{KL}(q(\cdot|x) \,\|\, P(\cdot|x;\theta))$$

since $\hat{\phi} = \underset{\phi}{\arg\max} \; ELBO$ given θ

$$= \underset{\phi}{\arg\min} \; D_{KL}(q_\phi(\cdot|x) \,\|\, P(\cdot|x;\theta))$$

$\Rightarrow$ we can expect $\underline{\underline{q_{\hat{\phi}}(z|x) \approx P(z|x;\theta)}}$

Lec 9

Re-cap:

- Unsupervised Learning
  - → PCA
  - → AE (Compression)
  - → Denoise AE
- Semi-supervised Learning
  - → Self-Learning
  - → Label Propagation

$$\mathcal{D} = \underbrace{\mathcal{D}_L}_{\boxed{\text{Small}}} \cup \underbrace{\mathcal{D}_u}_{\boxed{\text{Big}}}$$

Today:

1. Learning Across Tasks

↓

Assumption: Manifold Assumption

① Transfer Learning →
- Data Similar
- Task Very Different

we hope we can learn some features that are useful for different tasks

Approach
- fine-tune & pre-train
- warm-start

(w.r.t distribution)

② Domain Adaptation →
- Data Source Different
- Task Very Similar

Idea ⇒

Source Domain (Labelled)	→ train model
Target Domain (Unlabelled)	→ prediction

Approach
- a) Match distribution (via AE etc.)
- b) Common Representation Learning
- c) Label propagation

learn transformation f s.t. $f(\text{Target}) \approx \text{Source}$

$\boxed{\text{transport maps}}$

$\rightarrow$ Adversarial Training

2. VAE

① Probablistic Modelling

- a) Density Estimation
- b) Generative Model

a) Density Estimation $\quad \boxed{x \sim p^* \approx p_{\hat\theta}}$

$\boxed{\text{aim is } p_{\hat\theta}}$ density function

$\downarrow$

$\boxed{\text{difficult to sample}}$

$\downarrow$

Parametric density model $p_\theta(x)$

$\Downarrow$

MLE to estimate $\hat\theta$

b) Generative Model $\quad x \sim p^* \Rightarrow \text{approximately } \hat{x} \sim \hat{p}$

$\boxed{\text{aim is sample } \hat{x}}$

② GMM

$$\begin{cases} z \sim \text{Multi-Bernoulli}(\alpha) \\ x|z \sim \text{Gaussian}(\mu_z, \Sigma_z) \end{cases}$$

$$\boxed{p_\theta(x) = \sum_i \alpha_i \, p_{\text{Gaussian}}(x; \mu_i, \Sigma_i)}$$

⇒ Both Density Estimation & Generative Model

Learning is via MLE + EM Framework

$$\boxed{\Theta = \{ \{\hat{\alpha}_i\}_{i=1}^{k}, \{\hat{\mu}_i, \hat{\Sigma}_i\}_{i=1}^{k} \}}$$

Limitation :

a) Not complex enough w.r.t model capacity

b) Need to choose (# of clusters) → hyper-parameter

c) No prior knowledge (translation invariance of CNN)

③ Utilize AE to generate samples ?

↓

generate $\boxed{z}$ in latent space

$$\boxed{z} \xrightarrow{\text{decoder}} \boxed{dec(z)}$$

$\boxed{dec(z)}$ ⇝ sample we want to generate

④ To construct a " Generative Model "

we need { able to generate, model capacity

⇒ $\boxed{\text{latent variable model}}$

$$P_\theta(x) = \int_z P_\theta(x|z)\, P_\theta(z)\, dz$$

$$\begin{cases} p_\theta(z) \longrightarrow \text{prior dist} \\ p_\theta(x|z) \longrightarrow \text{generative dist} \end{cases}$$

$\boxed{\text{Generating Scheme}}$

1. learn $\underline{p_\theta(z)}$ & $\underline{p_\theta(x|z)}$
2. calculate $p_\theta(z|x)$ from Bayes Theorem
3. when we have x

(VI)

$\boxed{\text{intractable}}$

$\uparrow$

$\downarrow$

sample from $z \sim \boxed{p_\theta(z|x)}$

$\approx$

$q_\phi(z|x)$

$\downarrow$

sample from $\tilde{x} \sim p_\theta(x|z)$