DL   L 16 : 04.07.2022

Cost f^{n} of VAE :

$$\min_{\underline{\theta}} L(\underline{\theta}) = \frac{1}{N} \sum_{n=1}^{N} l(\underline{x}(n); \underline{\theta}),$$

$$l(\underline{x}; \underline{\theta}) = l_{rec}(\underline{x}; \underline{\theta}_E; \underline{\theta}_D) + l_{KL}(\underline{x}; \underline{\theta}_E)$$

2^{nd} trick to implement :

AE : deterministic encoding

encoder o/p = value of $\underline{z}$ for a given $\underline{x}$ (one-to one mapping)

VAE : stochastic encoding

encoder o/p $\neq$ ~~but~~ $\underline{z}$, rather mean $\underline{\mu}(\underline{x})$ and covariance $\underline{\underline{C}}(\underline{x})$ of

$\underline{z} \sim N(\underline{\mu}, \underline{\underline{C}})$ to draw random samples of $\underline{z}$ for generative purpose (~~one~~ many to many mapping)

$\Rightarrow$ generative similar to $\hat{x}$

9-16

Forward pass is good but in backprop. algo. there is an issue as there is no differentiable f^{n} for the random sample generator.

3rd trick : Reparametrization

slide 9-17

⋮

9-18

⋮

9.4 Generative Adversarial Network (GAN)

- Generative model, very powerful
- Different than VAE
- "coolest thing in DL"

9-19

Generator $\widehat{=}$ decoder

9-20

Intuitions for training:

D: a) Max $D(\underline{x})$ for real $\underline{x} \sim p_{Data}$

b) min $D(\underline{x})$ for fake $\underline{x} = G(\underline{z}),\ \underline{z} \sim p_{noise}$

G: c) max $D(\underline{x})$ for fake $\underline{z} \sim p_{noise}$, $\underline{x} = G(\underline{z})$

(b), (c) are opposite → adversarial training ≙ enemy

Training:

$$\min_{\underline{\theta}_G} \max_{\underline{\theta}_D} L(\underline{\theta}_G, \underline{\theta}_D) \;\hat{=}\; \underbrace{\text{minimax optimization}}_{\text{see DPR}}$$

$$L(\underline{\theta}_G, \underline{\theta}_D) = E_{\underline{x} \sim p_{Data}} \ln D(\underline{x}; \underline{\theta}_D) \cancel{+ E_{\underline{z} \sim p_{noise}} \ln(1 - D(G(\underline{z}; \underline{\theta}_G);}$$

a) ↑

D

G

independent of θ_G

$$+ E_{\underline{z} \sim p_{noise}} \ln \left(\underbrace{1 - D(G(\underline{z}; \underline{\theta}_G); \underline{\theta}_D)} \right)$$

(b) ↑ ↓

(c) ↓ ↑

$\cancel{G}$

Why so much complicated?

ch 4.6 : Categorical CE for binary classification

min − Categorical CE

or max $y \ln f(\underline{x};\underline{\theta}) + (1-y)\ln[1-f(\underline{x};\underline{\theta})]$

class label real : $y=1$ fake $y=0$

Training in Practice :

- Fix one parameter and optimize the other } Repetitive.
- fix the optimized and optimize the other.
- $E(\ )$ replaced by $\frac{1}{B}\sum_{n=1}^{B}(\ )$ of one minibatch.
- Alternatively : $\max_{\underline{\theta}_D} L$ for a fixed $\underline{\theta}_G$

  and then $\min_{\underline{\theta}_G} L$ for a fixed $\underline{\theta}_D$

9 - 21

9 - 22

9-23
⋮
9-24
⋮
9-25

Applications of GAN: general realistic fake samples

- data augmentation - to generate artificial data
- data generation without privacy concern.

Further developments of GANs:

Limitations of GAN:

noise $\underline{z} \xrightarrow{G} \hat{\underline{x}}$

no control about $\hat{\underline{x}}$, no label : which digit to get?

A. Sol^{n} Conditional GAN (CGAN)

controlled generation

9-26

B Further execution of C GAN :

image - to - image translation

signal to signal "

condition on an image / signal instead of label

given image $\xrightarrow{G}$ translated image

9 - 27

Application : all kinds of translation

$$\underline{y} \longrightarrow \boxed{\qquad\qquad} \longrightarrow \hat{\underline{x}}, \text{ close to } \underline{x}$$

much like (linear) filter, but

- any non-linear mapping
- mapping is not human designed, rather learnt automatically from data

$\hat{=}$ general framework of non-linear signal processing without any signal processing

9 - 28

+ great, realistic fake samples

-