

Cost fun of VAE :

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

$$\mathcal{L}(\underline{x}; \underline{\theta}) = \mathcal{L}_{\text{rec}}(\underline{x}; \underline{\theta}_E; \underline{\theta}_D) + \mathcal{L}_{\text{KL}}(\underline{x}; \underline{\theta}_E)$$

2nd 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 $\underline{\Sigma}(\underline{x})$ of

$\underline{z} \sim N(\underline{\mu}, \underline{\Sigma})$ 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 fun 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 $\hat{=}$ decoder

9-20

Intuitions for training:

D : a) Max $D(x)$ for real $x \sim P_{\text{Data}}$

b) min $D(x)$ for fake $x = G(z)_{z \sim P_{\text{Noise}}}$

G : c) max $D(x)$ for fake $z \sim P_{\text{Noise}}$, $x = G(z)$

(b), (c) are opposite \rightarrow adversarial training $\hat{=}$ enemy

Training:

$$\min_{\theta_G} \max_{\theta_D} L(\theta_G, \theta_D) \hat{=} \underbrace{\text{minimax optimization}}_{\text{see DPR}}$$

$$L(\theta_G, \theta_D) = E_{x \sim P_{\text{Data}}} \ln D(x; \theta_D) + E_{z \sim P_{\text{Noise}}} \ln (1 - D(G(z); \theta_G, \theta_D))$$

a) \uparrow
independent of θ_G

(b) \uparrow \downarrow
(c) \downarrow \uparrow

G

Why so much complicated?

ch 4.6 : Categorical CE for binary classification

min - Categorical CE

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

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:

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

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

A. Solⁿ Conditional GAN (CGAN)

controlled generation

9-26

B Further execution of CGAN:

image-to-image translation

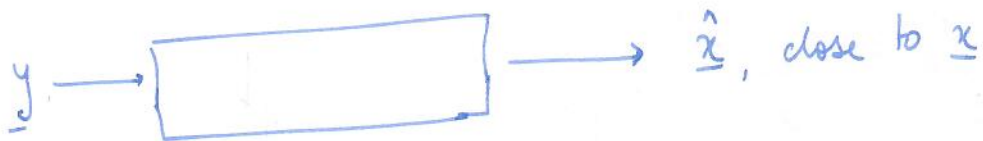
signal to signal

Condition on an image/signal instead of label

given image \xrightarrow{G} translated image

9-27

Application: all kinds of translation



much like (linear) filter, but

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

\equiv general framework of non-linear signal processing without any signal processing

9-28

+ great, realistic fake samples

-