CH20 - Deep Generative Models - I. Goodfellow

20.10.3 : Variational Autoencoders
20.10.4 : Generative Adversarial Networks

Variational Autoencoders

VAE = Directed model that uses learned approx inference & can be trained w/ gradient based methods

- VAE draws sample z from code dist $p_{model}(z)$ — goes through Autoencoder
- z run through gen network $g(z)$
- x sampled from $p_{model}(x ; g(z)) = p_M(x \mid z)$

During training, encoder uses $q(z \mid x)$ to obtain z

& $p_{model}(x \mid z)$ is the decoder

VAE is elegant, theo pleasing & simple

obtains excellent results among state of art generative modelling

main draw back is images tend to be blurry

One nice property of VAEs:
Simeltaneoasly train a parametric encoder in combination w/ generator. Forces the Model to learn a predictable coord sys that the encoder can capture

Makes it excellen manifold algo

20.10.4 Generative Adversarial Networks

GANS = Another generative model approach based on Differentiable generator networks

GANS based on theoretical scenario in which the generator must <u>compete against</u> an adversary

generator produces samples: $x = g(z; \theta^{(g)})$

Adversary/discriminator network:
Attempts to distinguish between samples drawn from the train data & samples from generator

Discrim gives probability value given by $d(x, \theta^{d})$ indic whether x is a real training sample

Zero-sum game to train

generator wants to learn to "trick" the discriminator

Convergence = ~~discrim~~ generator output is indistinguisable

↳ Disc is then discarded

→ generator is the trained model

Motivation for GANs
- requires neither Approx inference
- nor Approx of partition function

Unfort, GANs are Difficult to train when gen (G) & Pobvehe (D) are given by neural networks
& $\max_d v(g, d)$ is not convex

~~to~~ Non-converge causes GANs to underfit

Stabilization of GAN learning is a open problem

GANs perform well when model archicetire & hyperparas are carefully selected

Dropout is important for training