Deep Probabilistic Generative Models

1. Reminder

1.1 Karush-Kuhn-Tucker conditions

Unconstrained Optimization

let $\mathrm{F}: \mathbb{R}^n \to \mathbb{R}$ be a function

$$\min_{x \in \mathbb{R}^2} \mathrm{F}(x)$$

If F is **convex** and **differentiable** Then a necessary and sufficient condition for $\tilde{x} \in \mathbb{R}^n$ to be optimal is : $\nabla \mathrm{F}(\tilde{x}) = 0$

Convex set
$X \subseteq \mathbb{R}^n$ is a convex set iff:

$$\forall x_1, x_2 \in X, \forall t \in [0,1]: \quad t x_1 + (1-t) x_2 \in X$$

Convex fonction
Let X be a convex set and $F: X \to \mathbb{R}$ be a function
F is convex iff: $F(t x_1 + (1-t) x_2) \leqslant t F(x_1) + (1-t) F(x_2)$
$F(x)$ is convex $\Leftrightarrow -F(x)$ is concave

Convex fonction: 1st order condition
F is a convex function iff:

$$\forall x, y \in x: F(y) \geqslant F(x) + \nabla F(x)^{\top}(x-y)$$

Convex fonction: 2st order condition
$\underbrace{\nabla^2 F(x)}_{\text{Hessian}}$ is positive semi-definte matrix.

Property

- Affine function is convex and concave.
- $F(x) = \exp(x)$ is convex.
- $F(x) = \log(x)$ is concave.
- $F(x) = ax^2 + bx^2 + c$ if $a \geqslant 0 \Rightarrow convex$ if else $a \leqslant 0 \Rightarrow concave$.
- Let $F_1 ... F_n$ be a set of convex fonctions and $w_1 .. w_n \geqslant 0$. Then $F(x) = w_1 F_1(x) + ... + w_n F_n(x)$ is a convex Fonction

Constrained Optimization

$$\min_{x \in \mathbb{R}^2} \mathrm{F}(x)$$

subject to: $g_i(x) = 0 \quad \forall i \in [1....n]$ and $h_i(x) \leqslant 0 \quad \forall i \in [1....p]$

Necessary optimisation conditions (KKT conditions)
Let $\hat{x} \in \mathbb{R}^n, \hat{\lambda} \in \mathbb{R}^n$ and $\hat{\mu} \in \mathbb{R}^P$ be primal and dual variables. If $\hat{x}, \hat{\lambda}$ and $\hat{\mu}$ are oplimal, then:
Stationarity

$$\nabla f(\hat{x}) + \sum_{i=1}^{m} \hat{\lambda}_i \nabla g_i(\hat{x}) + \sum_{i=1}^{p} \hat{\mu}_i \nabla h_i(\hat{x}) = 0$$

Primal Feasibility

$$\begin{aligned} \forall i \in [1...n] \quad & g_i(\hat{x}) = 0 \\ \forall i \in [1...p] \quad & h_i(\hat{x}) \leqslant 0 \end{aligned}$$

Dual Feasibility

$$\forall i \in [1...p] \quad \hat{\mu}_i \geqslant 0$$

Complementary slackness

$$\forall i \in [1...p] \quad \hat{\mu}_i h_i(\hat{x}) = 0$$

Note
IF : F is convex, $\quad \forall i \quad g_i$ is affine, $\quad \forall i \quad h_i$ is convex

The same thing for the problem of maximization but:

$$\nabla f(\hat{x}) + \sum_{i=1}^{m} \hat{\lambda}_i \nabla g_i(\hat{x}) - \sum_{i=1}^{p} \hat{\mu}_i \nabla h_i(\hat{x}) = 0$$

Then KKT conditions are sufficient

1.2 Probability

1.2.1 CDF

$$F(x) \triangleq P(X \leq x) = \begin{cases} \sum_{u \leq x} p(u) & \text{, discrete} \\ \int_{-\infty}^{x} f(u) \mathrm{d}u & \text{, continuous} \end{cases} \tag{1}$$

1.2.2 Mean μ and variance σ^2

$$\mathbb{E}[X] \triangleq \begin{cases} \sum_{x \in \mathcal{X}} x p(x) & \text{, discrete} \\ \int_{\mathcal{X}} x p(x) \mathrm{d}x & \text{, continuous} \end{cases} \tag{2}$$

$$var[X] = \mathbb{E}[(X-\mu)^2] \tag{3}$$
$$= \mathbb{E}[X^2] - \mu^2 \tag{4}$$

1.2.3 product rule

:

$$p(X, Y) = P(X|Y) P(Y) \tag{5}$$

1.2.4 Bayes rule

$$\begin{aligned} p(Y=y|X=x) &= \frac{p(X=x, Y=y)}{p(X=x)} \\ &= \frac{p(X=x|Y=y) p(Y=y)}{\sum_{y'} p(X=x|Y=y') p(Y=y')} \end{aligned} \tag{6}$$

$$P(w_i \mid D) = \frac{P(D \mid w_i) P(w_i)}{\sum_{j=1}^{N} P(D \mid w_j) P(w_j)} \tag{7}$$

With $P(w_i)$ the prior probability, $P(w_i \mid D)$ the posterior probability, $P(D \mid w_i)$ the likelihood, $\sum_{j=1}^{N} P(D \mid w_j) P(w_j)$ the marginal likelihood or "model evidence".

1.2.5 Gaussian (normal) distribution

Table 1: *Summary of Gaussian distribution.*

Written as	$f(x)$	$\mathbb{E}[X]$	mode	var$[X]$
$X \sim \mathcal{N}(\mu, \sigma^2)$	$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$	μ	μ	σ^2

1.2.6 Covariance and correlation

$$\begin{aligned} \operatorname{cov}[X, Y] &\triangleq \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \end{aligned} \tag{8}$$

$$\text{corr}[X, Y] \triangleq \frac{\text{Cov}[X, Y]}{\sqrt{\text{var}[X], \text{var}[Y]}} \tag{9}$$

1.2.7 Central limit theorem

Given N random variables $X_1, X_2, \cdots, X_N$, each variable is **independent and identically distributed**, and each has the same mean μ and variance σ^2, then

$$\frac{\sum_{i=1}^{n} X_i - N\mu}{\sqrt{N}\sigma} \sim \mathcal{N}(0,1) \tag{10}$$

this can also be written as

$$\frac{\bar{X} - \mu}{\sigma / \sqrt{N}} \sim \mathcal{N}(0,1) \quad \text{, where } \bar{X} \triangleq \frac{1}{N} \sum_{i=1}^{n} X_i \tag{11}$$

2. Generative modeling

A model is generative if it places a joint distribution over all observed dimensions of the data.
Consider a supervised learning task with features X and labels Y:

- Generative models want to learn P(X, Y).
- Discriminative models want to learn P(Y | X).

2.1 How to design a rich family of probability distributions?

Three basic recipes for using a flexible function $f_\theta()$:

1. Apply a richly parameterized transformation to a simple random variable.

$$Z \sim \mathcal{N}(0, \mathbf{I}) \qquad X = f_\theta(Z)$$

2. Use a rich mixing distribution for a simple parametric family.

$$Z \sim \mathcal{N}(0, \mathbf{I}) \qquad X = \mathcal{N}(f_\theta(Z), \Sigma)$$

3. Specify a complicated distribution via its log density:

$$X \sim \frac{1}{\mathcal{Z}_\theta} \exp\{f_\theta(x)\} \quad \mathcal{Z}_\theta = \int \exp\{f_\theta(x)\}\, dx$$

2.1.1 Recipe 1: Transform a simple random variable

Construct a family of densities $g_\theta(x)$ on R^K with parameters θ.

- Choose a simple continuous distribution on $\mathbb{R}^J$ with density $\pi(z)$.
- Parameterize a class of functions: $f_\theta : \mathbb{R}^J \to \mathbb{R}^K$.

$$g_\theta(x) = \pi\left(f_\theta^{-1}(x)\right) \left|\mathcal{J}\left[f_\theta^{-1}(x)\right]\right|$$

where $\mathcal{J}[]$ is the Jacobian matrix
Classic Example:

- Factor Analysis and Principal Component Analysis
- Independent Component Analysis (ICA)
- the decoder portion of an autoencoder
- generative adversarial network

2.1.2 Recipe 2: Mix a simple random variable

Construct a family of densities (or PMFs) $g_\theta(x)$ with parameters θ.

- Choose a family of simple distributions π_z , parameterized by z.
- The family π_z can be discrete, continuous, or both.
- Define a distribution $\Psi_\theta(z)$ on z with parameters θ
- Draw a z from $\Psi_\theta(z)$, then $x \sim \pi_z$.

Classic Example :

- Gaussian Mixture Model
- Latent Dirichlet Allocation
- Nonlinear Gaussian belief networks
- Variational autoencoder

2.1.3 Recipe 3: Specify a log density directly

Construct a family of densities (or PMFs) $g_\theta(x)$ with parameters θ.

- Parametrize any scalar function $f_\theta(x)$.
- Exponentiate and normalize:

$$g_\theta(x) = \frac{1}{\mathcal{Z}_\theta} \exp\{f_\theta(x)\}$$
$$\mathcal{Z}_\theta = \int \exp\{f_\theta(x)\}$$

- Typically requires Markov chain Monte Carlo to sample.

Markov chain Monte Carlo (MCMC):

- Random walk that converges to $g_\theta(x)$.
- Uses a stochastic operator $T\left(x \leftarrow x'\right)$.
- Ergodic and leave $g_\theta(x)$ invariant:

$$g_\theta(x) = \int g_\theta\left(x'\right) T\left(x \leftarrow x'\right) dx'$$

- Several common recipes:
  - Metropolis–Hastings
  - Gibbs sampling
  - Slice sampling
  - Hamiltonian Monte Carlo

Example :

- Ising Model
- Restricted Boltzmann Machine
- Deep Boltzmann Machines

3. The variational autoencoder

A standard autoencoder consists of an encoder and a decoder. Let the input data be X. The encoder produces the latent space vector z from X. Then the decoder tries to reconstruct the input data X from the latent vector z.

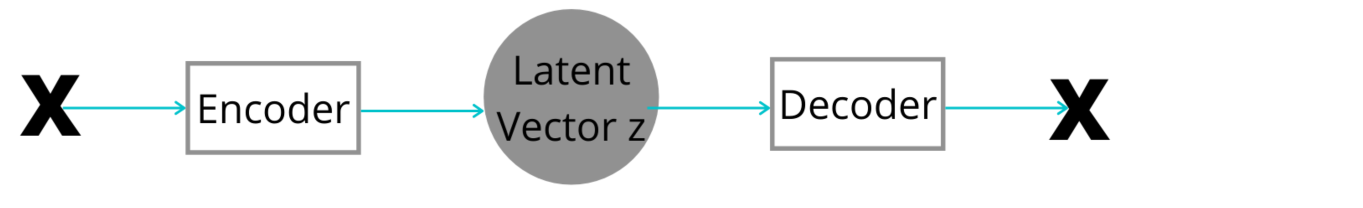

Figure 1: *The working of a simple deep learning autoencoder model.*

3.1 Basic VAE Generative Model

Spherical Gaussian latent variable:

$$Z \sim \mathcal{N}(0, \mathbf{I})$$

Transform with a neural network to parameterize another Gaussian:

$$x \mid z, \theta \sim \mathcal{N}\left(\boldsymbol{\mu}_\theta(z), \boldsymbol{\Sigma}_\theta(z)\right)$$

3.2 The Problem

In the case of an autoencoder, we have z as the latent vector. We sample $p_\theta(z)$ from z. Then we sample the reconstruction given z as $p_\theta(x|z)$. Here θ are the learned parameters.
We want to maximize the log-likelihood of the data. The marginal likelihood is composed of a sum over the marginal likelihoods of individual datapoints. That is,

$$\log p_\theta\left(x^{(1)}, \ldots, x^{(N)}\right) = \sum_{i=1}^{N} \log p_\theta\left(x^{(i)}\right)$$

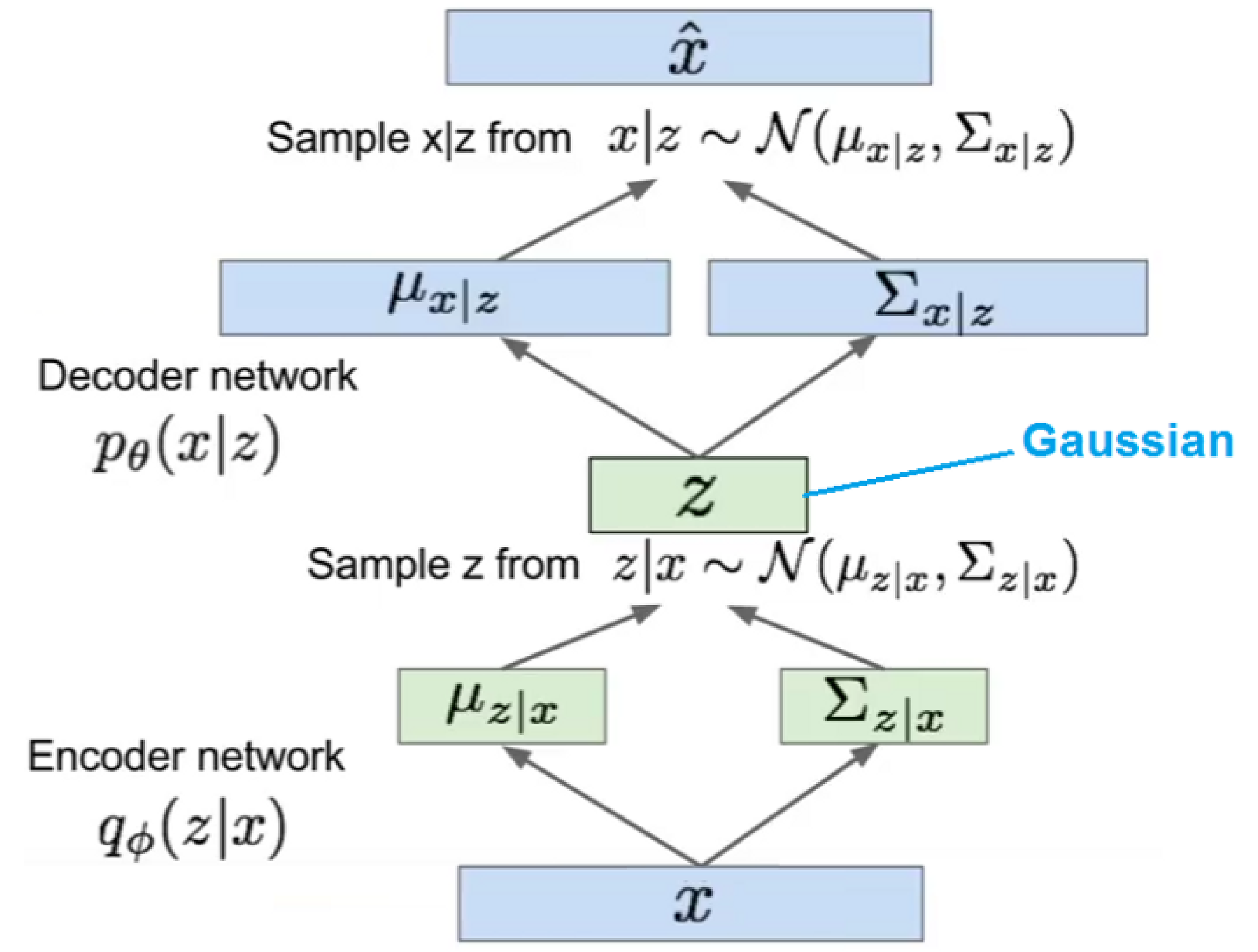

Figure 2: *VAE*