ML_AS7 111652042 顏友君 (^_^)

1. Based on the paper "A Connection Between Score Matching and Denoising Autoencoders"

<u>Explain the concept of score matching.</u>

Score matching (SM) is a method for estimating the parameters of unnormalized probability density models. It was introduced by Hyvärinen (2005) and discussed in Vincent (2010) as an alternative to maximum likelihood (MLE) estimation when the partition function $Z(\theta)$ is intractable.

The main idea is to make the model's score function, which is the gradient of the log probability with respect to the data, similar to the score of the true data distribution.

The probability density model $p(x;\theta)$ with parameters θ can be written as

$$p(x;\theta) = \frac{1}{Z(\theta)} e^{-E(x;\theta)},$$

where E is the Energy function.

We called "score" the gradient of the log density with respect to the data vector x:

$$\psi(x;\theta) = \frac{\partial \log p(x;\theta)}{\partial x}.$$

Score matching minimizes the expected squared difference between this model score (i.e. $\psi(x;\theta)$) and true data score ($\frac{\partial \log g(x)}{\partial x}$, $g(x)$: Unknown true pdf, for $x \in \mathbb{R}^d$), i.e.

$$J_{ESM}(\theta) = \mathbb{E}_{g(x)}\left[\frac{1}{2}\left\|\psi(x;\theta) - \frac{\partial \log g(x)}{\partial x}\right\|^2\right].$$

In other words, it trains the model so that the way its log-density changes with respect to x matches how the real data distribution through its gradients instead of matching actual probability values.

Since the true distribution $g(x)$ is unknown, Hyvärinen also derived an equivalent version called "implicit score matching", which avoids computing $g(x)$ directly and depends only on the model and observed samples.

In short, SM helps the model learn the same "direction of change" in probability density that the real data would have, without requiring the normalization constant.

Describe how it is used in score-based (diffusion) generative models.

In score-based or diffusion generative models, SM trains a model to learn the gradient of the data log-density instead of the normalized probability.

Since the original method needs to compute the Hessian trace of the model's score function, which is expensive, scalable variants such as DSM (Vincent, 2010) were developed.

Denoising Score Matching (DSM):

DSM connects SM with Denoising Autoencoders (DAEs). It learns to reconstruct a clean input x from a noisy version $\tilde{x}$ which is obtained by adding Gaussian noise.

Its objective function is

$$J_{DAE_\sigma}(\theta) = \mathbb{E}_{q_\sigma(x,\tilde{x})}\left[\| W^T \text{sigmoid}(W\tilde{x}+b)+c-x\|^2\right]$$, where $\theta = \{W, b, c\}$ are optimzed.

Vincent showed that minimizing this loss equals minimizing the DSM objective.

$$J_{DSM_{q_\sigma}}(\theta) = \mathbb{E}_{q_\sigma(x,\tilde{x})}\left[\frac{1}{2}\| \psi(\tilde{x};\theta) - \frac{\partial \log q_\sigma(\tilde{x}|x)}{\partial \tilde{x}}\|^2\right]$$, where $\psi(\tilde{x};\theta)$ is the model's score function.

This avoids second derivatives (Hessian trace) and lets the model learn the correct score at different noise levels.

In diffusion models, a neural network $S_\theta(x,\sigma)$ trained with DSM learn to predict the gradient $\nabla_x \log q_\sigma(x)$ for all σ, which enables sampling by reversing the diffusion process from noise to data.

2. Unanswered question:

Vincent 的這篇論文都是在討論單一雜訊水平 σ (noise level),

但現代的 Diffusion Models 使用的是包含多個雜訊水平的雜訊排程 (noise schedule),

這個單一 σ 到多個 σ 的過程是怎麼演變的，或是有哪些功能進化了？