Lecture 14 - Simulated Annealing & AI for COP

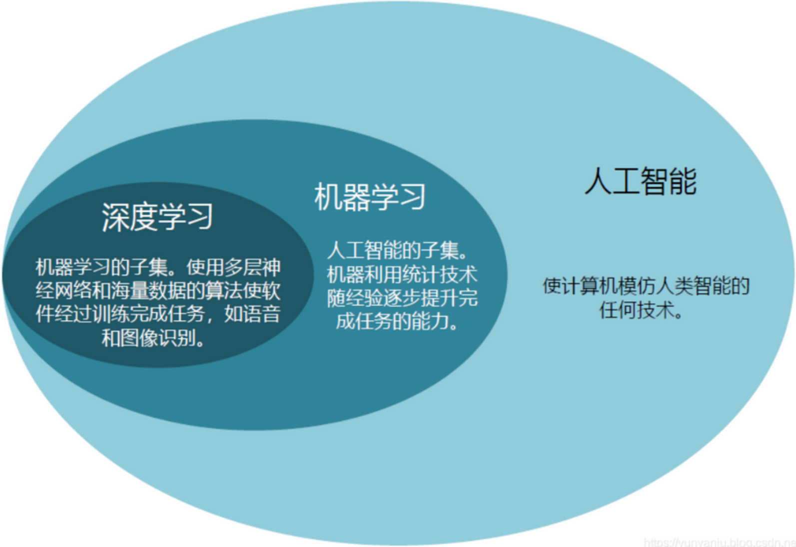
赵尉辰

南开大学 统计与数据科学学院

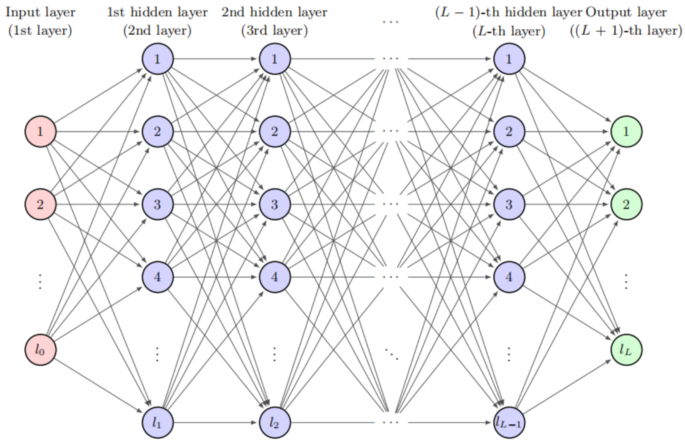
目录

目录

人工智能


深度学习
机器学习的子集。使用多层神经网络和海量数据的算法使软件经过训练完成任务，如语音和图像识别。
机器学习
人工智能的子集。机器利用统计技术随经验逐步提升完成任务的能力。
人工智能
使计算机模仿人类智能的任何技术。

深度神经网络



用矩阵形式表示:

$$H_{l+1} = \sigma(H_l W_l + b_l), \ l = 1, 2, \ldots, L$$

神经网络的优化

- 梯度下降法

$$\Theta^{(k+1)} = \Theta^{(k)} - \eta_k \frac{1}{N} \sum_{i=1}^{N} \nabla_{\Theta^{(k)}} \mathscr{L}(f(x_i, \Theta^{(k)}), y_i)$$

- 反向传播算法
  通过复合函数求导的链式法则，来计算每层参数的梯度。
- 随机梯度下降法&Mini-batch梯度下降法

$$\Theta^{(k+1)} = \Theta^{(k)} - \eta_k \nabla_{\Theta^{(k)}} \mathscr{L}(f(x_i, \Theta^{(k)}), y_i)$$

$$\Theta^{(k+1)} = \Theta^{(k)} - \eta_k \frac{1}{B} \sum_{i \in \mathcal{B}} \nabla_{\Theta^{(k)}} \mathscr{L}(f(x_i, \Theta^{(k)}), y_i)$$

其中$\mathcal{B} \subset \{1, 2, \ldots, N\}$，$B := |\mathcal{B}| < N$.

深度学习方法的优势

- 数据驱动的高效表示；
- 端到端的能力；
- 泛化能力；
- 大规模并行计算(GPU)。

AI for Science

AI for Science[1]

- 生命科学：蛋白质结构预测、蛋白质结合点位预测、蛋白质设计...
- 化学材料：分子表示、新材料发现、化学分子设计、分子动力学模拟...
- 物理：量子自旋系统基态表示...
- 医学：药物设计...
- 数学：偏微分方程数值解、组合优化求解...

[1] Zhang X, Wang L, Helwig J, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems[J]. arXiv preprint arXiv:2307.08423, 2023.

AI for COP[2]

组合优化问题(Combinatorial Optimization Problem, COP)是非常重要的问题。
传统求解方法(精确算法、近似算法、启发式算法...)存在以下局限性:

- 计算复杂度高：组合优化问题通常是NP难的，传统精确算法在问题规模较大时计算时间呈指数级增长，难以实际应用。
- 算法设计依赖经验：人工设计算法对领域知识依赖性强，且难以泛化到不同问题实例。
- 适应性不足：实际场景中约束条件或目标函数可能随时间变化，传统方法需要反复重新求解，效率低下。

[2] Guo T D, Li A Q, Han C Y. Machine learning method for combinatorial optimization problems (in Chinese). Sci Sin Math, 2025, 55: 1–30, doi: 10.1360/SSM-2024-0180

目录

模拟退火算法

回顾Boltzmann分布/Gibbs测度，在一个由 N 个粒子组成的系统中，当系统处于温度 T 时，粒子在能量为 E_i 的状态下的概率 P_i 由以下公式给出：

$$P_i = \frac{e^{-\beta E_i}}{Z}$$

其中$\beta = \frac{1}{k_B T}$, $Z \triangleq \sum_j e^{-\beta E_j}$.

考察$T \to \infty$时，对于每个状态$i \in \mathcal{S}$,

$$e^{-\beta E_i} = e^0 = 1$$

所以高温状态下，所有可能出现的状态是等概率的

$$\lim_{T \to \infty} P_i = \frac{1}{|\mathcal{S}|}$$

模拟退火算法

考察退火过程，$T \to 0$时，

$$
\begin{aligned}
\lim_{T\to 0} P_i &= \lim_{T\to 0} \frac{1}{Z}\exp\left(-\frac{E_i}{kT}\right) \\
&= \lim_{T\to 0} \frac{\exp\left(-\frac{E_i - E_{\min}}{kT}\right)}{\sum_{i\in\mathcal{S}} \exp\left(-\frac{E_i - E_{\min}}{kT}\right)} \\
&= \lim_{T\to 0} \frac{\exp\left(-\frac{E_i - E_{\min}}{kT}\right)}{\sum_{i\in\mathcal{S}_{\min}} \exp\left(-\frac{E_i - E_{\min}}{kT}\right) + \sum_{i\notin\mathcal{S}_{\min}} \exp\left(-\frac{E_i - E_{\min}}{kT}\right)} \\
&= \begin{cases} \frac{1}{|\mathcal{S}_{\min}|}, & i \in \mathcal{S}_{\min} \\ 0, & i \notin \mathcal{S}_{\min} \end{cases}
\end{aligned}
$$

其中$E_{\min} = \min_{i\in\mathcal{S}} E_i$, $\quad \mathcal{S}_{\min} = \{i : E_i = E_{\min}\}$. 这表明，随着温度降低，粒子会倾向于能量最低的状态中去。

模拟退火算法

模拟退火(Simulated Annealing)是一种优化算法，它的基本想法是：把优化的目标函数 f 当作是物理中的能量函数，然后通过退火，即$T \to 0$，求得 f 的全局最小值。

那么优化问题转化成了：如何采样Boltzmann分布

模拟退火算法能保证收敛到全局最小值[3].

模拟退火算法可以求解组合优化问题。

[3] Holley R A, Kusuoka S, Stroock D W. Asymptotics of the spectral gap with applications to the theory of simulated annealing[J]. J. Funct. Anal, 1989, 83(2): 333-347.

目录

Quadratic Unconstrained Binary Optimization (QUBO)

定义 1 (Quadratic Unconstrained Binary Optimization)

QUBO 问题定义为

$$\min x^T Qx$$
$$s.t. \quad x \in S$$

其中S是一个二元*(binary)*离散空间$\{0,1\}^n$或者$\{-1,1\}^n$，$Q \in \mathbb{R}^{n\times n}$为一个$n$维方阵。

QUBO是一类非常广泛的问题，能够涵盖很多优化问题[4]：二次指派问题、背包问题、最大团问题、最大独立集问题、最大割问题、图染色问题、集合划分问题、约束满足问题、……

[4] Kochenberger G, Hao J K, Glover F, et al. The unconstrained binary quadratic programming problem: a survey[J]. Journal of combinatorial optimization, 2014, 28: 58-81.

QUBO 与 Ising 模型[5]

设有n个粒子，每个粒子的自旋(spin) $\sigma_i = -1$或者1，那么n个粒子的自旋组成一种构型/状态(configuration) $\sigma = (\sigma_i)$，记构型空间 $S := \{+1, -1\}^n$。

定义 2 (Ising模型)

构型空间S上定义如下的能量函数/哈密顿量*(energy function/Hamiltonian)*：

$$E(\sigma) = -\sum_{ij\text{相邻}} J_{ij}\sigma_i\sigma_j - h\sum_i \sigma_i$$

称为*Ising*模型，其中J_{ij}是相互作用强度，h是外场强度，$-\sum_{ij\text{相邻}} J_{ij}\sigma_i\sigma_j$表示相邻格点对上自旋相互作用对能量的贡献，$-h\sum_i \sigma_i$表示外场带来的势能。

Ising 模型的能量函数$E(\sigma)$和 QUBO 的优化目标$f(x) = x^T Qx$是一致的。

[5] Lucas A. Ising formulations of many NP problems[J]. Frontiers in physics, 2014, 2: 5.

QUBO 与 Ising 模型

回顾退火方法，当统计物理系统处于平衡时，粒子分布服从Boltzmann-Gibbs分布

$$p(\sigma, \beta) = \frac{e^{-\beta E(\sigma)}}{Z}$$

而系统平衡时会倾向于处于能量最低的状态，所以我们从中采样构型σ，大概率会使得能量函数$E(\sigma)$处于最小值，那么根据 QUBO 与 Ising 模型的关系，此时的σ大概率为 QUBO 问题的解。

直接从

$$p(x) = \lim_{\beta \to \infty} p(x, \beta) = \lim_{\beta \to \infty} \frac{e^{-\beta f(x)}}{Z} = \lim_{\beta \to \infty} \frac{e^{-\beta x^T Q x}}{Z}$$

离散采样得到x即可求解QUBO问题。[6]

[6] Sun H, Goshvadi K, Nova A, et al. Revisiting sampling for combinatorial optimization[C]//International Conference on Machine Learning. PMLR, 2023: 32859-32874.

基于退火求解QUBO

基于退火方法求解 QUBO 已经转化为从 $p(x,\beta)=\frac{\mathrm{e}^{-\beta x^T Qx}}{Z}$ 中采样的问题。按是否使用深度学习可以分成两类方法:

- 经典方法直接采样，例如MCMC (online method)[7];
- 首先训练一个深度学习模型 $q_\theta(x,\beta)$ 来近似 $p(x,\beta)$，然后用模型$q_\theta(x,\beta)$输出样本 (offline training + online inference)

我们接下来关注第二类方法。

[7] Sun H, Goshvadi K, Nova A, et al. Revisiting sampling for combinatorial optimization[C]//International Conference on Machine Learning. PMLR, 2023: 32859-32874.

基于退火求解QUBO的深度学习方法

训练$q_\theta(x,\beta)$ 近似 $p(x,\beta)$有两个关键问题：

- 如何构建损失函数？

  在机器学习中一个常见的度量两个分布之间距离的损失函数是KL散度

  $$D_{KL}(p(x,\beta)||q_\theta(x,\beta)) \triangleq \int p(x,\beta)\log\frac{p(x,\beta)}{q_\theta(x,\beta)}dx \approx \frac{1}{N}\sum_{n=1}^{N}\log\frac{p(X_n,\beta)}{q_\theta(X_n,\beta)}$$

  其中$X_n \sim p(x,\beta)$，但是我们要处理的是采样问题，没有$p(x,\beta)$的样本。

- 如何建模$q_\theta(x,\beta)$？

  这对应了不同的模型。

损失函数

Inverse KL 散度

$$D_{KL}(q_\theta(x,\beta)||p(x,\beta)) \triangleq \int q_\theta(x,\beta)\log\frac{q_\theta(x,\beta)}{p(x,\beta)}dx$$
$$= \int q_\theta(x,\beta)(\log q_\theta(x,\beta) + \beta f(x) + \log Z)dx$$
$$:= \beta(F_q(\theta,\beta) - F)$$

其中

$$F_q(\theta,\beta) = \frac{1}{\beta}\int q_\theta(x,\beta)(\log q_\theta(x,\beta) + \beta f(x)), \quad F = -\frac{1}{\beta}\log Z.$$

那么我们的优化目标转化为

$$\min_\theta D_{KL}(q_\theta(x,\beta)||p(x,\beta)) \Leftrightarrow \min_\theta F_q(\theta,\beta).$$

平均场近似[8]

建模$q_\theta(x,\beta)$的一个平凡的方法是用平均场近似(Mean-Field Approximation)：

$$q_\theta(x,\beta)=\prod_i q_\theta(x_i,\beta)$$

其中x_i是高维向量x的第i个分量，独立地优化 $q_\theta(x_i,\beta)$即可。

这种建模方法比较简单，难以近似复杂的分布$p(x,\beta)$。

[8]Shen Z S, Pan F, Wang Y, et al. Free-energy machine for combinatorial optimization[J]. Nature Computational Science, 2025: 1-11.

自回归建模[9]

自回归建模 (Autoregressive Modelling) 方法：

$$q_\theta(x,\beta) = \prod_i q_\theta(x_i | x_1, ..., x_{i-1}; \beta)$$

运用策略梯度 (policy gradient) 方法训练

$$\beta \nabla_\theta F_q(\theta,\beta) = \mathbb{E}_{x\sim q_\theta(x,\beta)} \left(\nabla \log q_\theta(x,\beta)(\log q_\theta(x,\beta) + \beta f(x))\right)$$

这种建模方法一旦解的前几个分量 $x_1, x_2, \ldots$ 有错误，便只能将错就错下去。

[9] Sanokowski S, Berghammer W, Hochreiter S, et al. Variational annealing on graphs for combinatorial optimization[J]. Advances in Neural Information Processing Systems, 2023, 36: 63907-63930.

扩散模型建模[10]

关于KL散度，一个基本的观察是:

$$D_{KL}\left(q_\theta(x) \| p(x)\right) \leq D_{KL}\left(q_\theta(x, z) \| p(x, z)\right)$$

其中

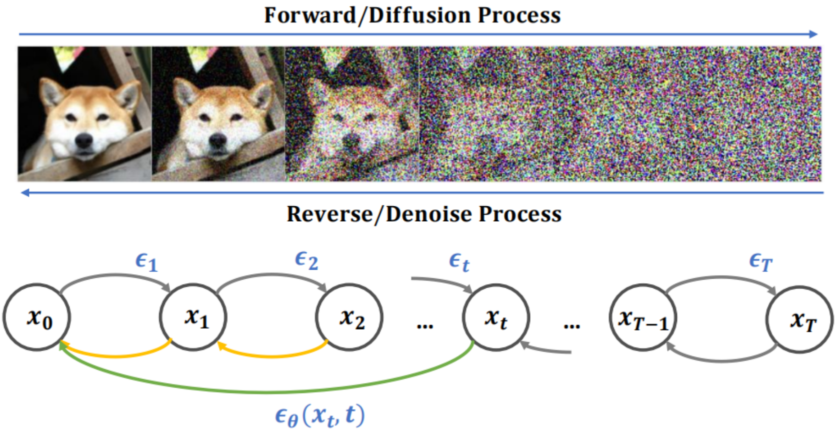
$$q_\theta(x, z) = q_\theta(x \mid z) q(z), \quad p(x, z) = p(x \mid z) q(z)$$

我们可以通过优化上界来优化$D_{KL}\left(q_\theta(x) \| p(x)\right)$

扩散模型中加噪的过程可以对应到这里的z，这启发我们用扩散模型来建模$q_\theta(x, z)$

[10] Sanokowski S, Hochreiter S, Lehner S. A diffusion model framework for unsupervised neural combinatorial optimization[C]//Proceedings of the 41st International Conference on Machine Learning. 2024: 43346-43367.

Denoising Diffusion Probabilistic Models, DDPM[11]



[11] Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33, 2020

Forward Diffusion Process

- 一步加噪过程

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{(1-\alpha_t)}\boldsymbol{\epsilon}_{t-1}, \quad \text{where } \boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(0, \mathbf{I}).$$

- t步加噪过程

命题 1

条件分布$q(\boldsymbol{x}_t|\boldsymbol{x}_0)$为

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\mathbf{I}),$$

其中 $\overline{\alpha}_t = \prod_{i=1}^t \alpha_i$. 即 $\mathbf{x}_t = \sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon}_0$.

能够计算$q(\boldsymbol{x}_t|\boldsymbol{x}_0)$的好处在于给定$\boldsymbol{x}_0$，给一个$t$，可以直接得到$\boldsymbol{x}_t$.

Proof

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \underbrace{\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}}_{\mathbf{w}_1}.
\end{aligned}
$$

由于$\boldsymbol{\epsilon}_{t-2}$和$\boldsymbol{\epsilon}_{t-1}$都是标准高斯的，$\mathbf{w}_1$是均值为0的高斯，我们下面计算协方差

$$
\begin{aligned}
\mathbb{E}[\mathbf{w}_1\mathbf{w}_1^T] &= [(\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}})^2 + (\sqrt{1-\alpha_t})^2]\mathbf{I} \\
&= [\alpha_t(1-\alpha_{t-1}) + 1-\alpha_t]\mathbf{I} = [1-\alpha_t\alpha_{t-1}]\mathbf{I}.
\end{aligned}
$$

延用记号$\boldsymbol{\epsilon}_t$

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} \\
&= \sqrt{\alpha_t\alpha_{t-1}\alpha_{t-2}}\mathbf{x}_{t-3} + \sqrt{1-\alpha_t\alpha_{t-1}\alpha_{t-2}}\boldsymbol{\epsilon}_{t-3} \\
&= \cdots = \sqrt{\prod_{i=1}^{t}\alpha_i}\mathbf{x}_0 + \sqrt{1-\prod_{i=1}^{t}\alpha_i}\boldsymbol{\epsilon}_0.
\end{aligned}
$$

Reverse Denoising Process

我们希望用一个神经网络实现降噪过程，即

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx q(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

由Markov性，

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t)q(\mathbf{x}_t)}{q(\mathbf{x}_{t-1})} \xRightarrow{\text{condition on } \mathbf{x}_0} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}$$

在优化神经网络的过程中转化为[12][13]

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$$

[12] Luo C. Understanding diffusion models: A unified perspective[J]. arXiv preprint arXiv:2208.11970, 2022.

[13] Chan S H. Tutorial on Diffusion Models for Imaging and Vision[J]. arXiv preprint arXiv:2403.18103, 2024.

Reverse Denoising Process

命题 2

条件分布$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$为一个高斯分布 $\mathcal{N}(\mathbf{x}_{t-1};\ \boldsymbol{\mu}_q(\mathbf{x}_t,\mathbf{x}_0),\boldsymbol{\Sigma}_q(t))$，其中

$$\mu_q(\mathbf{x}_t,\mathbf{x}_0)=\frac{(1-\overline{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\overline{\alpha}_t}\mathbf{x}_t+\frac{(1-\alpha_t)\sqrt{\overline{\alpha}_{t-1}}}{1-\overline{\alpha}_t}\mathbf{x}_0$$

$$\boldsymbol{\Sigma}_q(t)=\frac{(1-\alpha_t)(1-\sqrt{\overline{\alpha}_{t-1}})}{1-\overline{\alpha}_t}\mathbf{I}$$

$$\begin{aligned}
q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) &= \frac{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_0)q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\\
&= \frac{\mathcal{N}(\boldsymbol{x}_t;\sqrt{\alpha_t}\boldsymbol{x}_{t-1},(1-\alpha_t)\mathbf{I})\mathcal{N}(\boldsymbol{x}_{t-1};\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0,(1-\bar{\alpha}_{t-1})\mathbf{I})}{\mathcal{N}(\boldsymbol{x}_t;\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0,(1-\bar{\alpha}_t)\mathbf{I})}\\
&\propto \exp\left\{-\left[\frac{(\boldsymbol{x}_t-\sqrt{\alpha_t}\boldsymbol{x}_{t-1})^2}{2(1-\alpha_t)}+\frac{(\boldsymbol{x}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0)^2}{2(1-\bar{\alpha}_{t-1})}-\frac{(\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)^2}{2(1-\bar{\alpha}_t)}\right]\right\}
\end{aligned}$$

Reverse Denoising Process

注意到，给定加噪schedule，$\boldsymbol{\Sigma}_q(t)$是已知的，所以我们只需要参数化均值部分，即

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q(t))$$

两个高斯分布之间的KL散度[14]可以容易计算：

$$D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) = \frac{1}{2\boldsymbol{\Sigma}_q^2(t)}\left[\left\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\right\|_2^2\right]$$

注意到

$$\begin{aligned}
\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) &= \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{(1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}\boldsymbol{x}_0 \\
&= \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{(1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}} \\
&= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0
\end{aligned}$$

[14] $D_{KL}(p(x)||q(x)) \triangleq \int p(x)\log\frac{p(x)}{q(x)}dx$

Denoising Diffusion Probabilistic Models

考虑

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

我们要学习的目标其实是一个 Denoiser $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$。

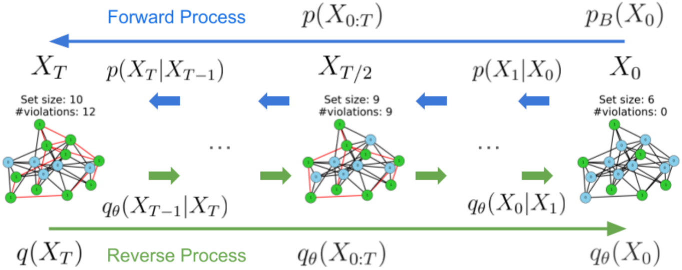
Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
until converged

Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

扩散模型建模



$$
\begin{aligned}
D_{KL}(q_\theta(X_{0:T})||p(X_{0:T})) = &-\sum_{t=1}^{T}\mathbb{E}_{X_{T:t}\sim q_\theta(X_{T:t})}\left[S(q_\theta(X_{t-1}|X_t))\right] \\
&-\sum_{t=1}^{T}\mathbb{E}_{X_{T:t-1}\sim q_\theta(X_{T:t-1})}\left[\log p(X_t|X_{t-1})\right] \\
&+\beta\mathbb{E}_{X_{T:0}\sim q_\theta(X_{T:0})}\left[f(X_0)\right]+C,
\end{aligned}
$$

其中$S(X)\triangleq -\int p(x)\log p(x)dx$是香农熵。

总结

- AI for COP 是一个正在被探索的领域，还有很多其他的方法;
- 在工业场景、管理科学中如何实际应用深度学习求解COP的方法;
- 欢迎同学找我讨论、课题研究。