最优化与人工智能

修贤超

上海大学自动化系

https://xianchaoxiu.github.io

目录

什么是最优化

- 最优化问题一般可以描述为

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{X} \end{aligned}$$

  - “min” 指 minimize, “s.t.” 指 subject to
  - $x = (x_1, x_2, \cdots, x_n)^\top \in \mathbb{R}^n$ 是决策变量
  - $f: \mathbb{R}^n \to \mathbb{R}$ 是目标函数
  - $\mathcal{X} \subseteq \mathbb{R}^n$ 是约束集合或可行域

最优化问题的类型

- **线性规划** 目标函数和约束函数均为线性函数的问题
- **整数规划** 变量只能取整数的问题
- **非线性规划** 目标函数和约束函数中至少有一个为非线性函数的问题
- **二次规划** 目标函数是二次函数而约束函数是线性函数的问题
- **半定规划** 极小化关于半正定矩阵的线性函数的问题
- **稀疏优化** 最优解只有少量非零元素的问题
- **非光滑优化** 包含非光滑函数的问题
- **低秩矩阵优化** 最优解是低秩矩阵的问题
- **等等**

目录

最小二乘法

- 最小二乘问题的一般形式如下

$$\min_{x\in\mathbb{R}^n} \quad \sum_{i=1}^{m} r_i^2(x)$$

- 如果所有的 $r_i : \mathbb{R}^n \to \mathbb{R}$ 都是线性函数, 则称线性最小二乘问题
- 1801 年, 24 岁的高斯计算出小行星的运动轨道

线性最小二乘

- 线性最小二乘问题是回归分析中的一个基本模型, 它可以表示为

$$\min_{x\in\mathbb{R}^n} \quad \sum_{i=1}^{m}(a_i^\top x - b_i)^2$$

- 记 $A = [a_1, a_2, \cdots, a_m]^\top$, 上式可以等价地写成

$$\min_{x\in\mathbb{R}^n} \quad f(x) = \frac{1}{2}\|Ax - b\|^2$$

- $x \in \mathbb{R}^n$ 为其全局极小解当且仅当 x 满足

$$\nabla f(x) = A^\top(Ax - b) = 0$$

非线性最小二乘

- 给定数据集 $\{a_i \in \mathbb{R}^p, b_i \in \mathbb{R}^q, i = 1, 2, \cdots, m\}$, 插值是求一个映射 f, 使得

$$b_i = f(a_i), \quad i = 1, 2, \cdots, m$$

- 利用线性函数 $f(a) = Xa + y$ 逼近, 可以建立如下最小二乘问题

$$\min_{X \in \mathbb{R}^{q \times p}} \quad \sum_{i=1}^{m} \|Xa_i + y - b_i\|^2$$

- 设 $\{\phi_i(a)\}_{i=1}^n (n \leq m)$ 为插值空间的一组基, 数据插值可以写成

$$b_j = f(a_j) = \sum_{i=1}^{n} x_i \phi_i(a_j) \;\Rightarrow\; \min_{x \in \mathbb{R}^n} \quad \sum_{j=1}^{m} \| \sum_{i=1}^{n} x_i \phi_i(a_j) - b_j \|^2$$

非线性最小二乘

- 设非线性向量函数 $\phi_i(\theta) : \mathbb{R}^q \to \mathbb{R}^q$, 并构造如下复合函数

$$f(\theta) = \phi_n(X_n\phi_{n-1}(X_{n-1}\cdots\phi_1(X_1\theta + y_1)\cdots + y_{n-1}) + y_n)$$

- 常用的有 ReLU, 即

$$\phi_i(\theta) = (\mathrm{ReLU}(\theta_1), \mathrm{ReLU}(\theta_2), \cdots, \mathrm{ReLU}(\theta_q))^\top, \quad i = 1, 2, \cdots, n$$

$$\mathrm{ReLU}(t) = \begin{cases} t, & t \geq 0 \\ 0, & \text{其他} \end{cases}$$

- 更多 “未知的非线性” 可能在更大的函数空间中得到一个更好的逼近

$$\min_{\theta \in \mathbb{R}^q} \quad \frac{1}{2}\|f(\theta) - b\|^2$$

通用近似定理

- **通用近似定理**（Universal Approximation Theorem），1989
- 令 $\phi(\cdot)$ 是一个有界、单调递增的连续函数，$\mathcal{J}_D$ 是一个 D 维的单位超立方体 $[0,1]^D$，$C(\mathcal{J}_D)$ 是定义在 $\mathcal{J}_D$ 上的连续函数集合。对于任意给定的一个函数 $f \in C(\mathcal{J}_D)$，存在整数 M，和实数 $v_m, b_m \in \mathbb{R}$ 以及实数向量 $w_m \in \mathbb{R}^D$，可以定义函数

$$F(x) = \sum_{m=1}^{M} v_m \phi(w_m^\top x + b_m)$$

作为函数 f 的近似实现，即

$$|F(x) - f(x)| < \epsilon, \forall x \in \mathcal{J}_D$$

神经网络优化

- 给定训练集为 $D = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$，将每个样本 $x^{(n)}$ 输入给神经网络，得到网络输出为 $\hat{y}^{(n)}$，其在数据集 D 上的结构化风险函数为

$$R(W, b) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(y^{(n)}, \hat{y}^{(n)}) + \frac{\lambda}{2} \|W\|_F^2$$

- 梯度下降

$$W^{(l)} \leftarrow W^{(l)} - \alpha \frac{\partial R(W, b)}{\partial W^{(l)}}$$

$$b^{(l)} \leftarrow b^{(l)} - \alpha \frac{\partial R(W, b)}{\partial b^{(l)}}$$

- 根据链式法则设计反向传播算法

自动微分

- 自动微分是利用链式法则来自动计算一个复合函数的梯度

$$f(x;w,b)=\frac{1}{\exp(-(wx+b))+1}$$

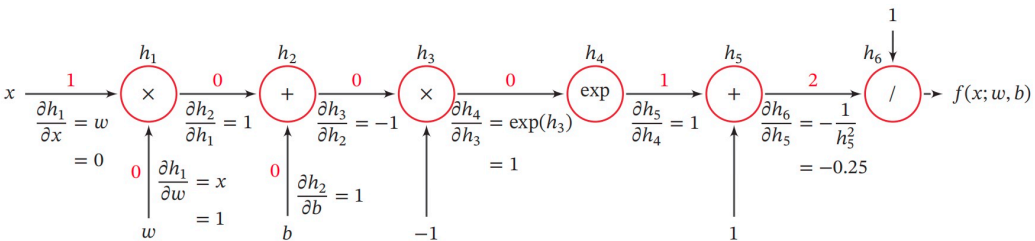
- 基本函数及其导数

函数	导数	
$h_1=x\times w$	$\frac{\partial h_1}{\partial w}=x$	$\frac{\partial h_1}{\partial x}=w$
$h_2=h_1+b$	$\frac{\partial h_2}{\partial h_1}=1$	$\frac{\partial h_2}{\partial b}=1$
$h_3=h_2\times -1$	$\frac{\partial h_3}{\partial h_2}=-1$	
$h_4=\exp(h_3)$	$\frac{\partial h_4}{\partial h_3}=\exp(h_3)$	
$h_5=h_4+1$	$\frac{\partial h_5}{\partial h_4}=1$	
$h_6=1/h_5$	$\frac{\partial h_6}{\partial h_5}=-\frac{1}{h_5^2}$	

自动微分

- 当 $x = 1,\ w = 0,\ b = 0$ 时，可以得到

$$\begin{aligned}\frac{\partial f(x;w,b)}{\partial w} &= \frac{\partial f(x;w,b)}{\partial h_6}\frac{\partial h_6}{\partial h_5}\frac{\partial h_5}{\partial h_4}\frac{\partial h_4}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial w} \\ &= 1 \times (-0.25) \times 1 \times 1 \times (-1) \times 1 \times 1 = 0.25\end{aligned}$$



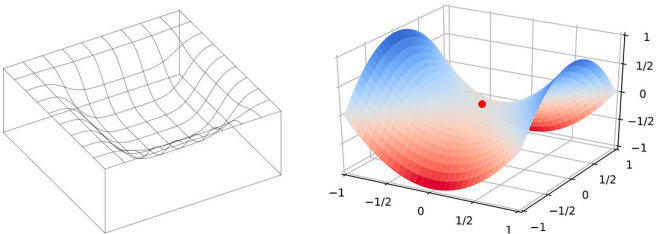
优化难点

- 深度学习的三个步骤

- 难点
  - 结构差异大
  - 非凸优化问题
  - 梯度消失（爆炸）问题

高维变量的非凸优化

- 高维空间中非凸优化的难点不在于逃离局部最优点，而是如何逃离鞍点



- 通过在梯度方向上引入随机性，可以有效地逃离鞍点
- 常用方法: sgd, asgd, adagrad, rmsprop, adadelta, adam, adamax
  https://github.com/pytorch/pytorch/tree/master/torch/optim

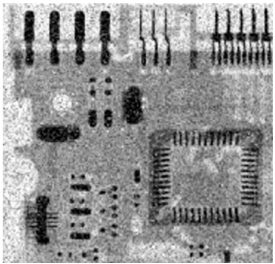
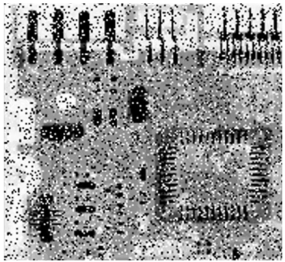
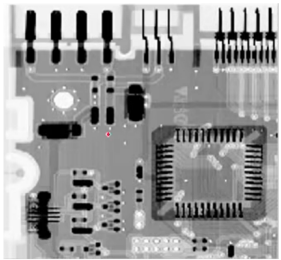
目录

- 如何进行图像去噪？考虑全变差模型

$$\min_x \quad \frac{1}{2}\|x-y\|^2+\gamma\|x\|_{\mathrm{TV}}$$

其中 $\|x\|_{\mathrm{TV}}$ 可取

$$\sqrt{(x_{i+1}-x_i)^2+(x_{i+n}-x_i)^2} \quad \text{或} \quad |x_{i+1}-x_i|+|x_{i+n}-x_i|$$

一般形式

- 复合优化问题

$$\min_{x} \quad f(x) + \gamma g(x)$$

  - $f(x)$ 是数据拟合项, $g(x)$ 是正则项
  - $\gamma \geq 0$ 是正则化参数

- 引入 $x = y$, 得到

$$\min_{x,y} \quad f(y) + \gamma g(x) \quad \text{s.t.} \quad x = y$$

- 增广拉格朗日函数为

$$\mathcal{L}_{\alpha}(x, y, u) = f(y) + \gamma g(x) + u^{\top}(x - y) + \frac{1}{2\alpha}\|x - y\|^2$$

一般形式

- 令 $\sigma^2 = \alpha\gamma$, 交替方向乘子算法（ADMM）

$$
\begin{aligned}
x^{k+1} &= \arg\min_x \{\sigma^2 g(x) + \frac{1}{2}\|x - (y^k - u^k)\|^2\} \\
y^{k+1} &= \arg\min_y \{\alpha f(y) + \frac{1}{2}\|y - (x^{k+1} + u^k)\|^2\} \\
u^{k+1} &= u^k + x^{k+1} - y^{k+1}
\end{aligned}
$$

- 具体形式为

$$
\begin{aligned}
x^{k+1} &= \mathrm{prox}_{\sigma^2 g}(y^k - u^k) \quad \Rightarrow \quad \text{图像去噪} \\
y^{k+1} &= \mathrm{prox}_{\alpha f}(x^{k+1} + u^k) \quad \Rightarrow \quad \text{与观测图像保持一致} \\
u^{k+1} &= u^k + x^{k+1} - y^{k+1}
\end{aligned}
$$

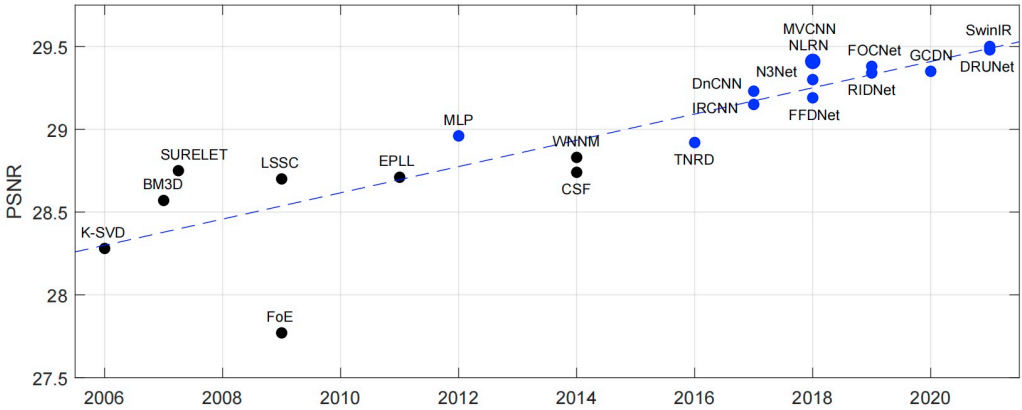
经典方法

- 如何选择 $g(x)$?

∼1970	Energy regularization	$\|\mathbf{x}\|_2^2$
1975–1985	Spatial smoothness	$\|\mathbf{L}\mathbf{x}\|_2^2$ or $\|\mathbf{D}_v\mathbf{x}\|_2^2 + \|\mathbf{D}_h\mathbf{x}\|_2^2$
1980–1985	Optimally learned transform	$\|\mathbf{T}\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{R}^{-1}\mathbf{x}$ where $\mathbf{T}/\mathbf{R}$ is learned via PCA
1980–1990	Weighted smoothness	$\|\mathbf{L}\mathbf{x}\|_{\mathbf{W}}^2$
1990–2000	Robust statistics	$\mathbf{1}^T\mu\{\mathbf{L}\mathbf{x}\}$ *e.g.,* Hubber-Markov
1992–2005	TV	$\int_{v\in\Omega}\|\nabla\mathbf{x}(v)\|dv$ or $\mathbf{1}^T\sqrt{\|\mathbf{D}_v\mathbf{x}\|^2 + \|\mathbf{D}_h\mathbf{x}\|^2}$
1987–2005	Other PDE-based options	$\int_{v\in\Omega} g\left[\nabla\mathbf{x}(v), \nabla^2\mathbf{x}(v)\right] dv$
2005-2009	Field-of-experts	$\sum_k \lambda_k \mathbf{1}^T\mu_k\{\mathbf{L}_k\mathbf{x}\}$
1993–2005	Wavelet sparsity	$\|\mathbf{W}\mathbf{x}\|_1$
2000–2010	Self-similarity	$\sum_k\sum_{j\in\Omega(k)} d\{\mathbf{R}_k\mathbf{x}, \mathbf{R}_j\mathbf{x}\}$
2002–2012	Sparsity methods	$\|\alpha\|_0\ s.t.\ \mathbf{x} = \mathbf{D}\alpha$
2010–2017	Low-rank assumption	$\sum_k \|\mathbf{X}_{\Omega(k)}\|_*$

性能对比

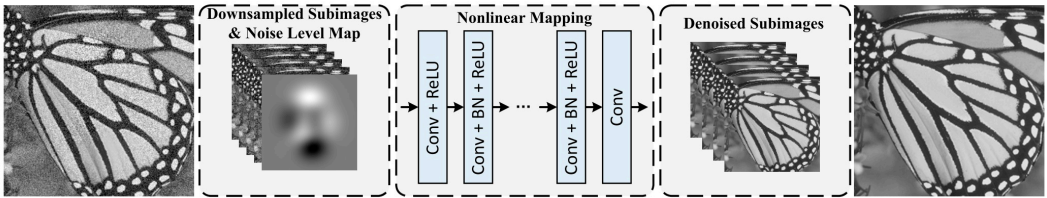
- 传统方法与深度学习方法的性能



代表性工作

- 能否利用成熟的去噪器?
  - 模型 + 数据, 优势互补
  - 当数据不足时, 无法进行端到端训练
- Toward a Fast and Flexible Solution for CNN-Based Image Denoising, 2018

$$\min_x \quad \frac{1}{2}\|x-y\|^2 + \gamma\Phi(x)$$



PnP-ADMM 算法

- 即插即用（Plug-and-Play, PnP）框架

$$x^{k+1} = \mathrm{prox}_{\sigma^2 g}(y^k - u^k) \quad \Rightarrow \quad x^{k+1} = H_\sigma(y^k - u^k)$$

- PnP-ADMM 迭代格式

$$\begin{aligned} x^{k+1} &= H_\sigma(y^k - u^k) \\ y^{k+1} &= \mathrm{prox}_{\alpha f}(x^{k+1} + u^k) \\ u^{k+1} &= u^k + x^{k+1} - y^{k+1} \end{aligned}$$

- 称 PnP-ADMM 为不动点迭代, 如果 (x^*, u^*) 满足

$$\begin{aligned} x^* &= H_\sigma(x^* - u^*) \\ x^* &= \mathrm{prox}_{\alpha f}(x^* + u^*) \end{aligned}$$

PnP-ADMM 收敛性

- **假设** 对于所有 $x, y \in \mathbb{R}^d$, 存在 $\epsilon \geq 0$ 使得 $H_\sigma : \mathbb{R}^d \to \mathbb{R}^d$ 满足

$$\|(H_\sigma - I)(x) - (H_\sigma - I)(y)\| \leq \epsilon \|x - y\|$$

- **定理** 假设 f 是 μ-强凸函数, 那么 PnP-ADMM 算法收敛只需要

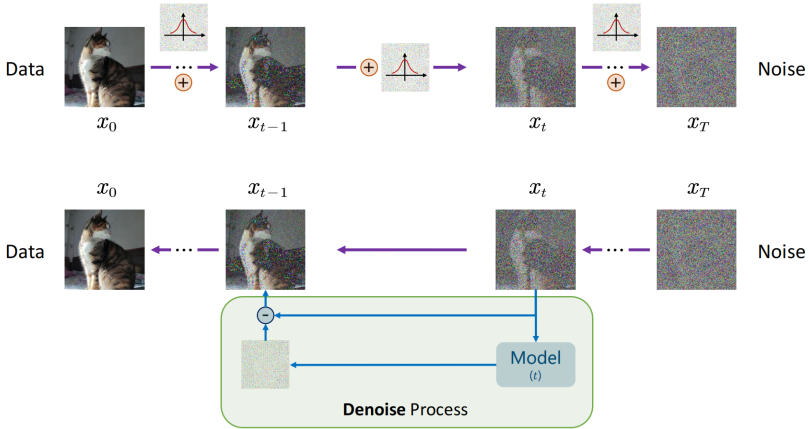
$$\frac{\epsilon}{(1 + \epsilon - 2\epsilon^2)\mu} < \alpha$$

- 注意到假设条件等价于限制残差 R 的利普希斯常数

Plug-and-play methods provably converge with properly trained denoisers
E Ryu, J Liu, S Wang, X Chen… - … on Machine Learning, 2019 - proceedings.mlr.press
Abstract Plug-and-play (PnP) is a non-convex framework that integrates modern denoising priors, such as BM3D or deep learning-based denoisers, into ADMM or other proximal

生成式方法

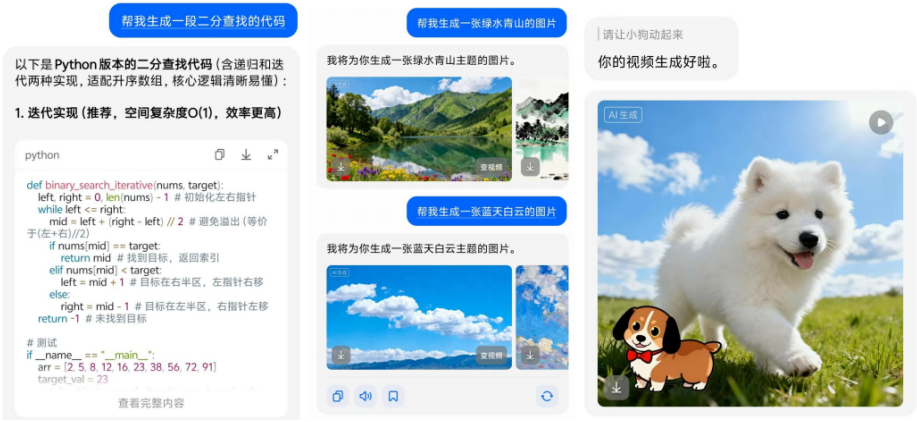
- 从数据中学习分布特征，并生成具有原始数据关键特征的全新样本，为最优化问题建模复杂数据分布提供了崭新的视角



目录

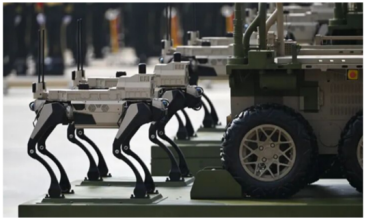
大模型具身智能

- 大模型是指具有大规模参数和复杂计算结构的机器学习模型，广泛应用于文本生成、图片生成和视频生成等场景



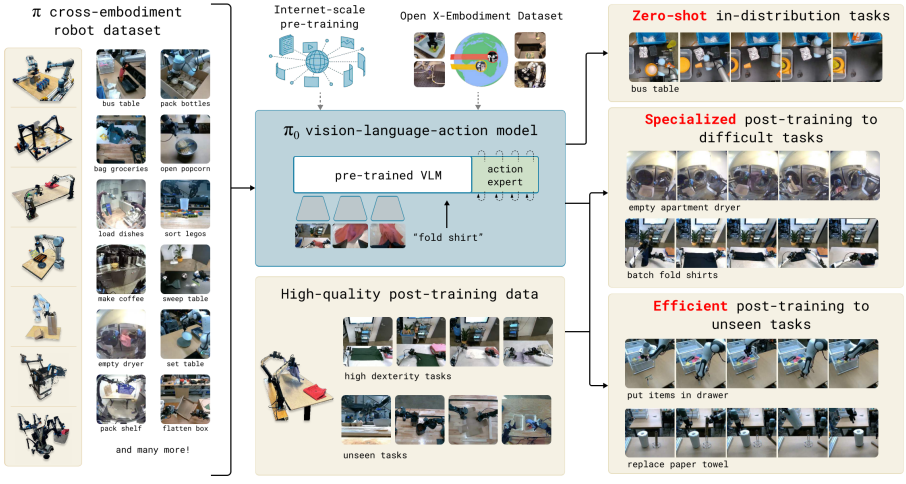
具身智能

- 具身智能强调智能体通过物理身体与环境的直接交互来产生认知和行动，其核心在于“感知-行动”闭环



具身智能

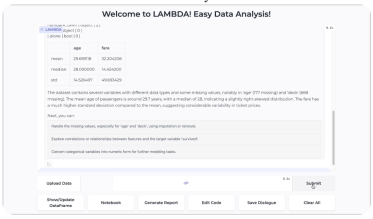
- https://www.physicalintelligence.company/blog/pi0



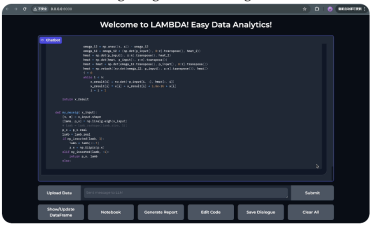
智能决策

- https://www.polyu.edu.hk/ama/cmfai/lambda.html

Data Analysis

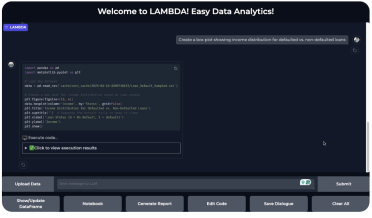


Integrating Human Intelligence



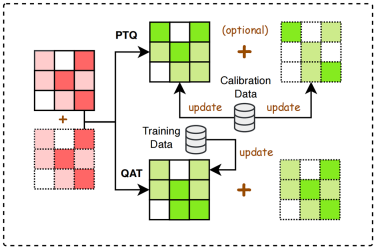
Data Science Education

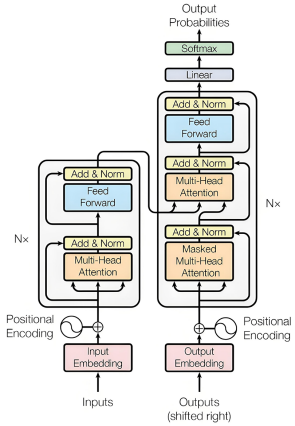
Loan Default Analysis



轻量化部署

- https://github.com/horseee/Awesome-Efficient-LLM





Q&A

Thank you!

感谢您的聆听和反馈