大模型应用

汇报人：明楷

目录

大模型应用概览

- 泛科技行业
  - 自然语言处理（NLP）
    - 在NLP领域，大模型的应用使得文本生成、情感分析、机器翻译等任务变得更加高效和精准。
  - 计算机视觉（CV）
    - 大模型在计算机视觉中的应用，尤其是在图像识别、物体检测和面部识别等任务中表现卓越。Vision Transformer（ViT）等大模型的引入使得深度学习能够处理更复杂的视觉数据，推动自动驾驶、安防监控以及医疗影像分析等领域的进步。

大模型应用概览

❊泛科技行业

❋语音识别与合成

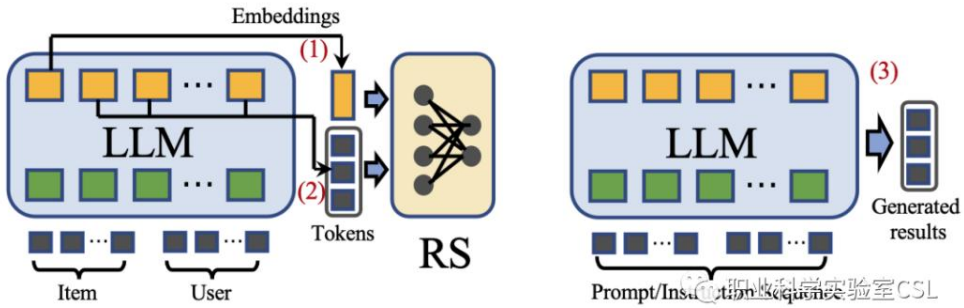
❁大模型在语音识别与合成中的应用也极为广泛，改变了人们与机器的互动方式。通过深度学习模型，像Google Assistant、Apple Siri这样的语音助手可以精准理解用户指令，并通过自然的语音合成与用户进行对话。此外，自动化客服系统的语音识别技术已经能够有效处理大量客户咨询，减少人工干预，提高效率。

大模型应用概览

❊泛科技行业

❊推荐系统

❁大模型在推荐系统中的应用大大提升了个性化推荐的准确性。通过分析用户的历史行为和偏好，推荐系统能够为用户推荐最相关的产品或内容。Amazon、Netflix、YouTube等平台都广泛使用基于大模型的推荐算法，为用户提供量身定制的购物清单、影视内容和视频推荐。



目录

大模型应用概览

❊政府服务

❊智能公共服务

❁大模型在智能公共服务中的应用为市民提供了更加便捷和高效的服务。例如，许多城市的智能政务平台已经通过大模型支持的聊天机器人，帮助民众快速解答有关政策法规、社会福利、税务申报等方面的问题。政府机关通过自然语言处理技术，能够理解并自动回应市民的咨询，从而减少人工干预，提升服务效率。

大模型应用概览

❊政府服务

❊北京市12345市民服务热线的智能客服系统

- ❁为了提高服务效率和质量，北京市12345市民服务热线引入了基于大模型的智能客服系统。这个系统利用自然语言处理（NLP）技术，可以理解市民的咨询意图，并自动提供准确的答案。
  - ❁数据训练：通过收集和整理大量的市民咨询记录，训练出一个能够理解多种咨询场景的大模型。
  - ❁系统集成：将这个大模型集成到12345市民服务热线的在线平台和移动应用中，使其能够作为一个聊天机器人与市民进行交互。
  - ❁功能实现：市民可以通过文字输入他们的问题，智能客服系统会实时解析问题，并从知识库中检索相关信息，提供即时的回答。

大模型应用概览

❊政府服务

❋数据分析与决策支持

❁大模型可以帮助政府通过海量数据的分析，提供精准的决策支持。利用大数据技术和机器学习算法，政府能够更好地预测经济趋势、社会发展方向以及应急事件的发生，为制定更为科学和合理的政策提供依据。特别是在社会保障、环境保护、公共健康等领域，大模型能够帮助政府深入分析民生数据，发现潜在问题，提升社会治理能力。

大模型应用概览

❊政府服务

❊数据分析与决策支持例子-上海市交通拥堵分析与预测系统

❁上海市交通委员会引入了一个基于大模型的交通拥堵分析与预测系统。这个系统结合了大数据分析技术和机器学习算法，能够处理和分析海量的交通数据。

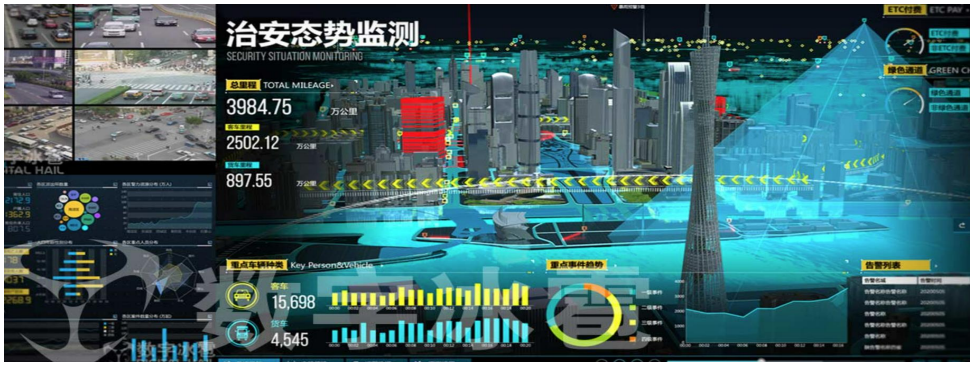
- ❁数据收集：系统收集了来自交通摄像头、GPS设备、移动应用、社交媒体等多种来源的实时交通数据。
- ❁模型训练：利用历史交通数据，训练机器学习模型，使其能够识别交通流量模式和拥堵趋势。
- ❁实时分析：系统实时分析交通数据，使用大模型预测特定时间段和地区的交通拥堵情况，并提供决策支持。

大模型应用概览

❊政府服务

❊公共安全与治安管理

❁在公共安全和治安管理方面，大模型的应用能够通过智能监控和数据分析预测潜在的安全风险和违法犯罪活动。通过视频监控、社交媒体分析等数据源，大模型能够帮助警方识别犯罪活动的规律，预警危险行为，并优化资源配置。例如，在大规模公共事件如大型集会、体育赛事等期间，政府可以通过大模型分析实时监控数据，迅速做出应对决策。



大模型应用概览

❊政府服务

❋公共安全与治安管理例子-深圳市公共安全智能监控系统

- ❁该系统利用大模型和人工智能技术，整合了视频监控、社交媒体分析、物联网设备等多种数据源，以实现对城市安全的全面监控和分析。
  - ❁数据整合：系统整合了全市的视频监控数据、交通流量数据、社交媒体信息、气象数据等，形成了一个庞大的数据池。
  - ❁模型训练：通过机器学习算法，训练大模型识别异常行为模式，如可疑人员聚集、交通异常等。
  - ❁实时监控与分析：系统实时分析监控数据，使用大模型预测潜在的安全风险和犯罪活动。
  - ❁预警与响应：当系统识别出潜在风险时，会自动向警方发出预警，并提供相应的应对建议。

❊政府服务

❊政策透明化

❁大模型还能够帮助政府提升政策透明度和公众参与度。通过人工智能技术，政府可以更高效地与民众进行互动，并向公众提供实时的政策解读和反馈。大模型可以帮助政府自动生成政策文档、公告、报告等，并通过智能平台向公众传达。此外，民众意见的采集和分析也能通过大模型实现，增强政府政策制定的公众参与感。

大模型应用概览

❊政府服务

❊政策透明化例子-新加坡政府的"智慧城市"政策透明化

- ❁他们开发了一个名为"SingPass"的智能平台，并结合自然语言处理（NLP）和机器学习技术，来优化政府与市民的互动。
  - ❁自动生成政策文档：利用大模型自动生成政策文档、公告和报告，通过智能平台实时发布给公众。
  - ❁智能政策解读：大语言模型提供实时政策解读，市民可通过自然语言提问，系统自动生成简明的解读答案。
  - ❁民众意见收集与分析：通过AI技术自动收集并分析民众的反馈，帮助政府更好地了解公众态度并改进政策。
  - ❁智能问答系统：智能问答平台基于NLP技术，支持市民与政府之间的互动，快速解答政策相关问题。

目录

大模型应用概览

❊金融行业

❊风险管理与信贷评估

❁在金融行业，风险管理和信贷评估是至关重要的环节。大模型通过分析历史数据、市场趋势和客户行为，能够帮助金融机构更加精准地评估贷款申请者的信用风险，并预测可能的违约行为。

❁应用案例

- ❁信用评分： 例如，许多银行和金融科技公司如Ant Group使用大模型来替代传统的信贷评估模型，通过分析客户的交易记录、社交网络和其他非结构化数据，提供更为精准的信用评分。
- ❁风险预警： 通过大数据分析，金融机构能够实时监控客户的财务状况，提前预警潜在的违约风险，为风险管理提供有力支持。

大模型应用概览

❊金融行业

❊风险管理与信贷评估例子-Ant Group“芝麻信用”评分系统

- ❁其推出的“芝麻信用”评分系统是一个典型的应用大模型的信贷评估平台，基于大数据和人工智能，提供个人和小微企业的信用评估。
  - ❁多样化数据源：通过收集用户的消费行为、社交网络互动、第三方数据等多种数据源，全面评估信用风险。
  - ❁机器学习模型：利用大模型和机器学习算法，从历史数据中提取模式，自动进行信用评分和风险预测。
  - ❁实时信用评估：系统根据实时交易数据、用户行为变化，动态调整信用评分，提供准确的信贷决策支持。
  - ❁违约预测与风险预警：通过分析客户的行为模式和市场趋势，预测潜在的违约风险，提前向金融机构发出预警，降低信贷损失。

大模型应用概览

❊金融行业

❊金融市场预测与投资决策

❁大模型还被广泛应用于金融市场预测和投资决策中。通过分析历史的市场数据、新闻报道、社交媒体数据等信息，深度学习模型可以识别出潜在的市场趋势和价格波动，为投资者提供科学的决策依据。

大模型应用概览

❊金融行业

❊金融市场预测与投资决策例子-高盛的“Quantitative Investment Strategies”

❁高盛（Goldman Sachs）是一家全球领先的投资银行和金融服务公司，其“量化投资策略（Quantitative Investment Strategies）”团队利用大数据和深度学习模型来分析市场数据、新闻报道、社交媒体动态等，为客户提供市场预测和投资决策支持。

大模型应用概览

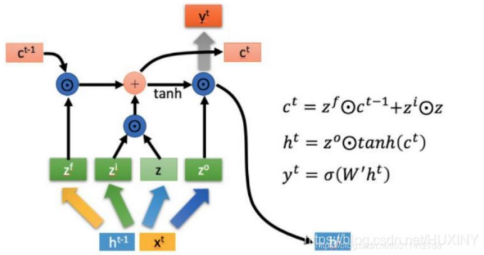
❊金融行业

- ❊金融市场预测与投资决策例子-高盛的“Quantitative Investment Strategies”
  - ❁实现原理与过程
  - ❁数据采集与多源融合：
    - ❁历史市场数据：包括股票价格、交易量、波动率等，帮助模型识别历史市场走势和价格波动规律。
    - ❁新闻与社交媒体数据：通过NLP技术和情感分析，分析金融新闻、社交媒体（如Twitter、Reddit）上的讨论，了解市场情绪对价格波动的影响。
    - ❁经济指标与宏观数据：结合GDP增长率、失业率、利率等宏观经济数据，为投资决策提供大环境分析。

大模型应用概览

❊金融行业

❊金融市场预测与投资决策例子-高盛的“Quantitative Investment Strategies”



❁实现原理与过程

❁深度学习与模型训练

- ❁深度神经网络（DNN）：高盛的量化团队使用深度神经网络模型（如LSTM、Transformer等）来处理和分析时间序列数据（如股市历史价格、交易量等）和非结构化数据（如新闻文本、社交媒体帖子等）。
- ❁情感分析：通过情感分析算法（如基于BERT的情感分析模型），从新闻报道和社交媒体中的文本数据中提取情绪信息，识别出积极或消极情绪的趋势，这些情绪变化可能对市场走势产生影响。

大模型应用概览

❊金融行业

❊金融市场预测与投资决策例子-高盛的“Quantitative Investment Strategies”

- ❁实现原理与过程
- ❁市场趋势预测与价格波动建模：
  - ❁趋势识别：利用深度学习模型自动从历史数据中识别出市场的潜在趋势（如上涨或下跌的市场周期）。模型能够根据不同的市场条件调整其预测策略。
  - ❁波动性预测：通过分析历史波动性和市场情绪变化，模型能够预测未来的价格波动，提供短期和长期的市场走势预判。

大模型应用概览

❊金融行业

❊金融市场预测与投资决策例子-高盛的“Quantitative Investment Strategies”

- ❁实现原理与过程
- ❁投资决策与风险管理：
  - ❁资产配置优化：基于大模型的预测结果，高盛的投资决策团队能够为客户提供量化的资产配置建议，帮助其优化投资组合，减少潜在的投资风险。
  - ❁自动化交易：高盛还利用大模型实现算法交易，根据市场预测信号自动执行交易，捕捉价格波动中的盈利机会。

大模型应用概览

金融行业

智能客户服务与财务咨询

- 大模型在客户服务领域的应用极大地提升了金融机构的服务质量和效率。许多银行和保险公司通过大模型开发智能客服系统，能够24小时为客户提供咨询服务。这些智能客服不仅能够回答常见问题，还能够根据客户的历史行为和需求提供个性化的财务咨询和产品推荐。
- 应用案例
  - 智能客服： 例如，花旗银行和摩根大通等国际金融机构已经通过大模型部署智能聊天机器人，为客户提供贷款咨询、账户查询、产品推荐等服务。
  - 个性化理财顾问：大模型能够分析客户的财务状况、投资目标等，提供定制化的理财建议和投资组合，帮助客户实现财富增值。

大模型应用概览

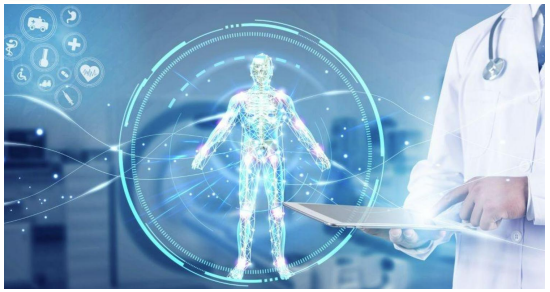
❊金融行业

❊智能客户服务与财务咨询例子-摩根大通的智能财务顾问—“You Invest”平台

- ❁摩根大通通过其智能客户服务平台“You Invest”为个人客户提供个性化的理财建议、投资组合管理和市场分析。
  - ❁通过分析客户的财务数据和行为，评估客户的财务状况、风险偏好和投资目标。
  - ❁智能客服系统通过NLP技术理解客户问题，自动提供财务解答和产品推荐。
  - ❁结合机器学习算法，根据客户数据和市场趋势预测最佳投资策略。
  - ❁根据客户需求和风险承受能力，提供定制化的理财建议和投资组合，帮助实现财富增值。

目录

大模型应用概览



❊医疗

❊疾病诊断与影像分析

❁大模型在医学影像分析中的应用非常广泛，特别是在影像数据处理和疾病诊断方面。通过训练深度学习模型，医疗系统可以自动分析CT扫描、X光片、MRI图像等医学影像，准确识别病灶辅助医生进行诊断。例如CNN已被用于检测肺结核、乳腺癌等疾病，极大提高了诊断的准确性和速度。

❁应用案例

❁肺部疾病诊断： 深度学习模型通过分析胸部X光和CT影像，能够检测出早期肺癌和肺结核等疾病，帮助医生及时做出治疗决策。

大模型应用概览

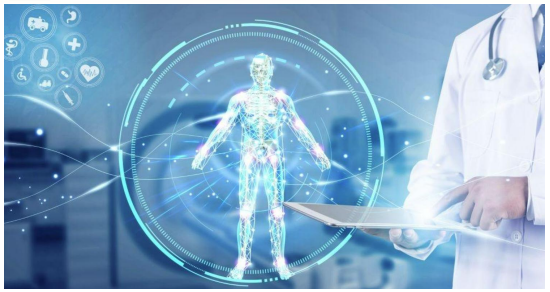
❊医疗

❊疾病诊断与影像分析例子-谷歌健康团队的肺癌检测系统

❁谷歌健康团队（Google Health）开发了一种基于深度学习的肺癌检测系统，该系统通过分析胸部CT扫描影像，自动识别肺癌的早期迹象。

- ❁数据收集与标注：通过收集大量的胸部CT扫描影像数据，并由专家医生进行标注，确保模型能学习病灶区域与正常组织的区别。
- ❁深度学习模型训练：采用卷积神经网络（CNN）对影像数据进行训练，自动提取和学习肺部疾病的影像特征。
- ❁模型优化与验证：通过与传统诊断方法对比，优化模型的准确性，并确保其在不同病例和影像中的泛化能力。
- ❁早期疾病检测：通过分析CT影像，系统能够自动识别肺癌的早期迹象，辅助医生及时做出诊断和治疗决策。

大模型应用概览



❊医疗

❋个性化治疗与精准医疗

❁大模型通过分析患者的基因组数据、电子健康记录和临床表现，能够为每个患者量身定制个性化的治疗方案，推动精准医疗的发展。大模型不仅可以在肿瘤治疗、药物选择等方面提供科学依据，还能够预测患者对不同治疗方法的反应，为临床医生提供决策支持。

❁应用案例

❁肿瘤治疗优化： 基于患者的基因数据，深度学习模型能够预测某种特定治疗对肿瘤患者的效果，帮助医生制定个性化的治疗方案。

大模型应用概览

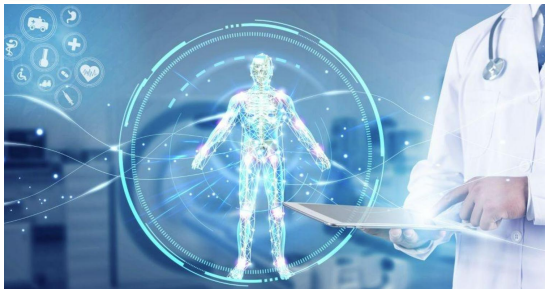
❊医疗

❊个性化治疗与精准医疗例子-IBM Watson for Oncology（IBM沃森肿瘤学）

❁该系统通过分析患者的基因组数据、临床数据、医学文献以及治疗历史，为肿瘤患者提供个性化的治疗建议，帮助医生做出科学的决策。

- ❁数据整合与分析：收集并整合患者的基因组数据、电子健康记录（EHR）和医学文献，全面了解患者的病史、基因特征和最新的临床研究成果。
- ❁深度学习与NLP模型训练：利用深度学习和自然语言处理（NLP）技术，对患者数据进行分析，从中提取有价值的治疗信息，并与全球的医学文献对比，生成个性化治疗建议。
- ❁个性化治疗方案推荐：根据患者的基因组特征和临床数据，预测治疗效果并为医生提供科学依据，帮助制定最适合患者的个性化肿瘤治疗方案。

大模型应用概览



❊医疗

❊临床决策支持与智能辅助

❁大模型为医生提供了强大的临床决策支持能力，能够根据患者的症状、检查结果以及过往病历数据，帮助医生做出更准确的诊断和治疗决策。例如，通过将自然语言处理模型应用于电子健康记录，大模型可以从患者的历史病历中提取关键信息，辅助医生诊断。

❁应用案例

❁智能诊断助手： 例如，IBM Watson for Health利用大模型帮助医生分析病人的数据，生成可能的疾病列表，并提出治疗方案建议。

大模型应用概览

❊医疗

❊临床决策支持与智能辅助例子-Google Health的“AI诊断助手”

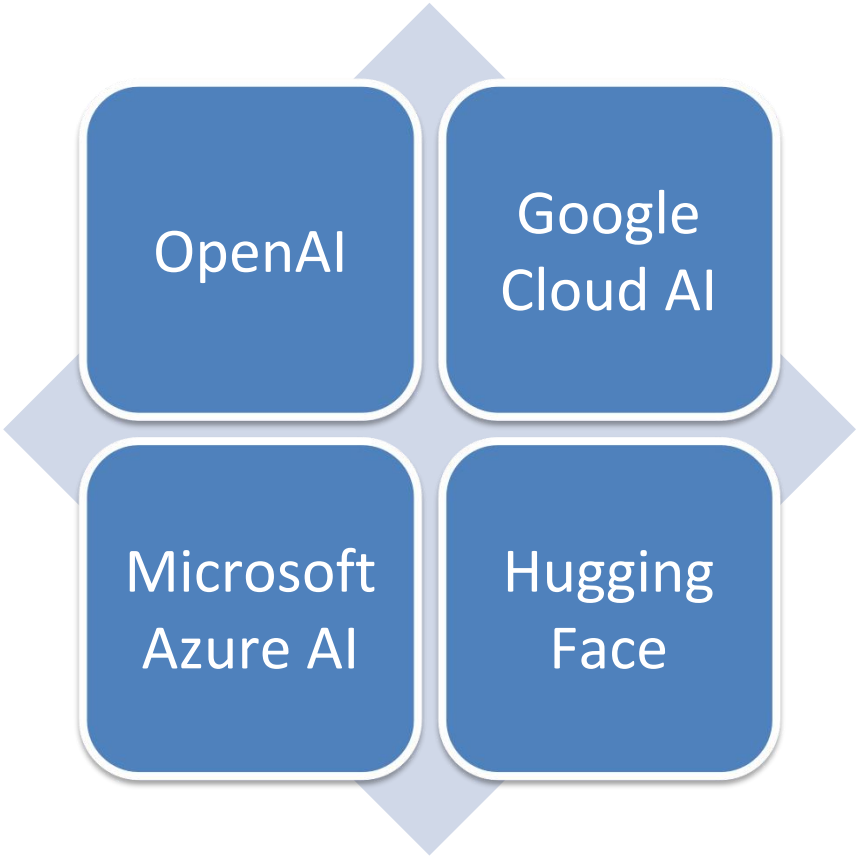
❁Google Health开发了一款智能诊断助手，结合了自然语言处理（NLP）和深度学习技术，能够从患者的电子健康记录（EHR）中提取关键数据，辅助医生做出更精确的诊断。

- ❁EHR数据整合与NLP处理：整合患者的症状、病史和实验室检查结果，利用自然语言处理（NLP）技术从电子健康记录中提取关键信息。
- ❁深度学习模型训练与疾病预测：通过深度学习技术训练模型，基于患者数据预测可能的疾病，并生成疾病候选列表。
- ❁临床决策支持与治疗建议：结合医疗知识库和治疗指南，为医生提供个性化的治疗方案建议，辅助诊断和决策过程。

目录

大模型平台简介

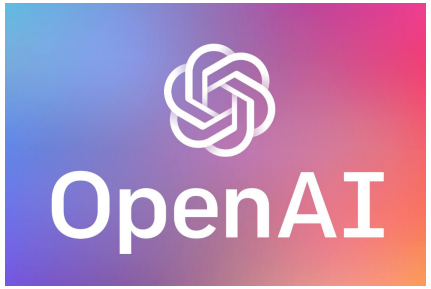
※主要大模型平台



大模型平台简介

主要大模型平台

OpenAI



- OpenAI是全球领先的人工智能研究机构，推出了以GPT系列为代表的大模型，支持自然语言处理、代码生成和多模态任务。
- 特点
  - 强大的自然语言处理能力，可用于聊天机器人、内容生成等领域。
  - 提供开放的API接口，支持开发者快速接入其大模型功能。
  - 最新的GPT-4引入了多模态支持，增强了图文理解与生成能力。

大模型平台简介

主要大模型平台

OpenAI

模型与技术

OpenAI平台提供了多个强大的AI模型，涵盖了各种任务和应用场景：

- GPT系列（包括GPT-3和GPT-4）：这是一个基于变换器（Transformer）架构的大型语言模型，能够理解和生成自然语言文本。GPT模型在多个领域（如对话生成、文本总结、情感分析等）具有强大的表现力。
- DALL·E：专注于图像生成的模型，可以根据文本描述生成高质量的图像。例如，用户只需要输入描述如“一个戴着太阳镜的海滩上的猫”，DALL·E就能根据描述生成相应的图像。

大模型平台简介

主要大模型平台

OpenAI

模型与技术

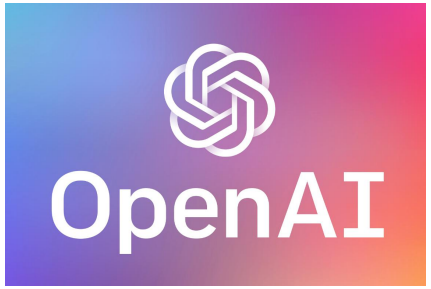
OpenAI平台提供了多个强大的AI模型，涵盖了各种任务和应用场景：

- Codex：这是一个生成代码的AI模型，能够根据自然语言指令生成编程代码。它是GitHub Copilot的核心技术，使开发者能够在编程过程中更高效地获得代码建议和自动完成。
- CLIP（Contrastive Language-Image Pre-training）：一个多模态模型，用于将文本和图像进行联合学习，能理解和关联图像与描述文本之间的关系，广泛应用于图像分类和搜索。
- Whisper：一个强大的语音识别模型，能够识别并转录语音到文本，支持多种语言，精度较高。

大模型平台简介

主要大模型平台

OpenAI



开放API与集成

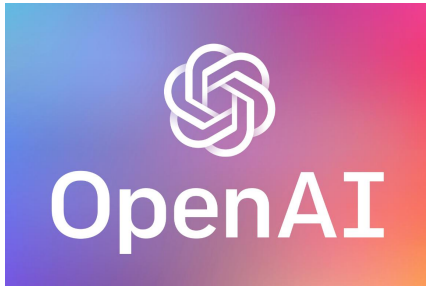
OpenAI平台提供了简便的API接口，开发者可以将其集成到各种应用中。通过这些API，用户可以访问OpenAI的各种模型：

- 文本生成与对话：利用GPT模型生成自然、流畅的对话和文本内容。例如，可以通过API构建智能客服、虚拟助手、内容创作工具等。
- 代码生成与编程支持：通过Codex模型，开发者可以将自然语言指令转化为编程代码，从而提高开发效率，特别适用于自动化编程任务、代码补全和修复等。

大模型平台简介

主要大模型平台

OpenAI



开放API与集成

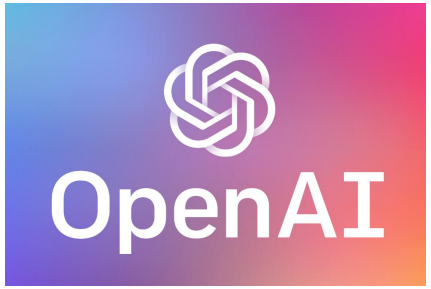
OpenAI平台提供了简便的API接口，开发者可以将其集成到各种应用中。通过这些API，用户可以访问OpenAI的各种模型：

- 图像生成：利用DALL·E模型，根据输入的文本描述生成图像，可以用于创意设计、广告制作等领域。
- 语音转录：通过Whisper模型，将音频文件中的语音内容转化为文本，广泛应用于会议记录、语音助手、字幕生成等场景。

大模型平台简介

主要大模型平台

OpenAI



定制化与训练

OpenAI平台允许用户根据自己的需求进行模型定制和微调。通过Fine-Tuning功能，用户可以在OpenAI的基础模型上进一步训练，适应特定领域或任务的要求，例如：

- 特定领域应用：为医疗、法律、金融等行业定制AI模型，使其能够理解行业专有术语和知识。
- 自定义对话：训练GPT模型，让其根据特定的对话风格和语境进行互动，增强与用户的个性化交流。
- 任务优化：通过Fine-Tuning使模型在特定任务（如情感分析、情节生成、推荐系统等）中表现更加优秀。

大模型平台简介

主要大模型平台

Google Cloud AI

- Google的AI平台依托于其强大的基础设施和开源工具（如TensorFlow），提供了一系列预训练大模型和AutoML服务。
- 特点
  - 支持大规模分布式计算，适合大模型的训练与部署。
  - 提供预训练模型（如BERT、T5）和自动化机器学习工具。
  - 在自然语言处理和计算机视觉领域具有突出优势。

大模型平台简介

- 主要大模型平台
  - Google Cloud AI
    - 核心功能
      - AI Platform Notebooks：提供托管的Jupyter笔记本环境，用于数据探索、模型开发和分析。
      - AI Platform Training：在Google Cloud的计算资源上训练机器学习模型，支持大规模分布式训练。
      - AI Platform Prediction：提供托管的预测服务，便于将训练好的模型部署并进行实时预测。

大模型平台简介

- 主要大模型平台
  - Google Cloud AI
    - Google Cloud AI APIs
      - Vision AI：可以识别图片中的物体、文字、情感等信息，支持图像分类、面部识别、物体检测等功能。
      - Speech-to-Text：将音频转换为文本，支持多种语言，广泛用于语音识别应用。
      - Text-to-Speech：将文本转换为自然语音，支持多个语言和声音样式。
      - Natural Language AI：提供自然语言处理服务，支持情感分析、实体识别、文本分类等功能。
      - Translation AI：提供实时翻译服务，支持多种语言之间的相互翻译。

大模型平台简介

主要大模型平台

Google Cloud AI

AutoML

Google的AutoML服务旨在为没有机器学习专业背景的开发者提供构建定制化AI模型的能力。

- AutoML Vision：自动构建图像分类模型，适用于自定义图片识别任务。
- AutoML Natural Language：用于文本分析，自动构建情感分析、实体识别等任务的模型。
- AutoML Translation：用于自动构建和优化翻译模型。
- AutoML Tables：用于表格数据的机器学习，帮助开发者进行结构化数据分析和预测建模。

大模型平台简介

主要大模型平台

Microsoft AI

- Microsoft AI整合了OpenAI的技术成果，并结合自身的认知服务，成为企业应用大模型的首选平台之一。
- 特点
  - 与OpenAI合作，支持GPT模型的企业级应用。
  - 提供一站式的AI开发工具，从数据预处理到模型部署全覆盖。
  - 强调数据安全和合规，适合大型企业使用。

大模型平台简介

主要大模型平台

Microsoft AI

Azure Cognitive Services

Azure Cognitive Services为开发者提供了数百种现成的AI功能，免去了开发和训练模型的复杂性。开发者可以通过简单的API调用，快速实现智能应用。

- Computer Vision：图像分析、文字识别、物体检测等。
- Face API：面部检测与识别，用于人脸验证与分析。
- Speech API：提供语音识别、语音合成、语音翻译等功能，支持多种语言。
- Text Analytics：进行情感分析、关键字提取、语言检测、实体识别等。
- Translator Text：提供实时文本翻译功能，支持多个语言对之间的翻译。
- Language Understanding (LUIS)：用于构建自然语言理解（NLU）模型，理解用户的意图和语境。

大模型平台简介

主要大模型平台

Microsoft AI

Power BI + AI

Microsoft的Power BI是一个强大的商业分析工具，集成了AI功能，以帮助用户从数据中获取深刻洞察。借助Azure AI的能力，Power BI能够自动分析数据、生成报告、提供预测模型等。

- 智能数据分析：通过集成Azure AI，Power BI可以分析数据中的模式并自动生成报告。
- 自然语言查询：用户可以使用自然语言提问，Power BI会自动生成相关的报告和图表。
- 预测分析：通过集成机器学习模型，Power BI能够提供基于数据的趋势预测。

大模型平台简介

主要大模型平台

Microsoft AI

Azure Bot Services

Azure Bot Services是一个平台，帮助开发者构建智能聊天机器人。通过集成自然语言处理和AI模型，开发者可以创建与用户进行自然对话的智能助手。

- Bot Framework：一个开源框架，用于构建和连接智能聊天机器人。
- Language Understanding (LUIS)：帮助开发者通过语音和文字理解用户意图。
- QnA Maker：允许开发者轻松构建基于问答的聊天机器人，支持自动应答。

大模型平台简介

主要大模型平台

Hugging Face

- Hugging Face是专注于自然语言处理的开源平台，提供了丰富的大模型库（如BERT、RoBERTa、GPT等）和工具。
- 特点
  - 开放的Transformer模型库，开发者可以轻松找到各种任务的预训练模型。
  - 提供用户友好的API和集成工具。
  - 活跃的社区支持，开发者能够快速解决技术问题。

大模型平台简介

主要大模型平台

Hugging Face

Transformers 库

Transformers 是 Hugging Face 提供的一个开源库，专门用于处理自然语言处理任务。它包含了多种预训练的语言模型，适用于文本生成、文本分类、命名实体识别、问答、翻译等多种任务。

- 多种预训练模型：支持各种著名的预训练语言模型，包括 BERT、GPT、T5等。这些模型可以直接用于不同的NLP任务，且通常在大型数据集上经过精细调优，表现优异。
- 多种语言支持：支持多种语言，适合跨语言任务的研究和应用。
- 高效的模型加载和推理：通过 transformers 库，用户可以轻松加载和推理这些预训练模型，并对其进行微调（fine-tuning），以满足特定应用场景的需求。

大模型平台简介

主要大模型平台

Hugging Face

Datasets 库

Hugging Face 还提供了一个名为 Datasets 的库，旨在帮助研究人员和开发者轻松访问各种公共数据集。该库可以直接加载多个领域的标准数据集，用于训练、验证和测试模型。

- 丰富的数据集：包括文本、图像、语音等多种类型的开放数据集，如GLUE、SQuAD、MNLI、IMDB等。
- 高效的数据加载：能够快速加载和处理大规模数据集，且提供数据预处理、过滤等功能。
- 无缝集成：与Hugging Face的Transformers库、Trainer等其他工具高度集成，使得数据预处理和模型训练更加方便。

大模型平台简介

主要大模型平台

Hugging Face

Accelerate 库

Accelerate 是 Hugging Face 提供的一个用于简化分布式训练的库，旨在让开发者能够轻松地在单机、多个GPU或者分布式环境下进行高效的训练。它支持PyTorch和TensorFlow框架，尤其适合大规模训练和微调任务。

- 简化分布式训练：通过简化代码，开发者可以快速设置多设备训练环境，减少工程复杂度。
- 支持各种硬件：支持CPU、GPU以及TPU等硬件设备的无缝切换。
- 增强效率：帮助开发者优化资源使用，最大化训练效率，减少训练时间。

大模型平台简介

平台服务形式

API 接口服务

- 大模型平台通常提供易于集成的API接口，允许开发者将大模型的功能嵌入到现有系统或新应用中。通过API，用户无需关注模型训练和基础设施部署的复杂性，而是专注于应用开发和业务逻辑。
- 特点
  - 快速集成：用户可以通过RESTful API或SDK直接调用模型功能，轻松实现文本生成、图像识别、语音处理等任务。
  - 灵活性强：API服务通常提供可调节的模型参数如生成文本长度、温度控制等，满足不同业务需求。
  - 即开即用：无需进行复杂的模型训练，用户可以直接使用平台提供的预训练模型。

大模型平台简介

平台服务形式

模型托管与自定义服务

- 许多平台提供模型托管服务，允许用户直接使用预训练模型，同时支持用户上传自己的数据进行微调，从而获得针对特定场景优化的模型。此类服务适合那些需要高度定制化模型的企业或开发者。
- 特点
  - 预训练模型支持：平台托管的大模型覆盖多种任务（如自然语言处理、计算机视觉）。
  - 微调与定制：用户可以在已有模型的基础上，用自己的数据集进行微调，提升模型在特定领域的表现。
  - 弹性扩展： 支持按需配置计算资源，满足从小规模实验到大规模部署的需求。

大模型平台简介

平台服务形式

端到端解决方案

- 一些大模型平台还提供端到端的AI解决方案，将模型开发、训练、部署、管理等环节整合在一个闭环系统中。用户只需关注具体的业务需求，而无需处理复杂的技术细节。这种形式非常适合缺乏AI开发能力的企业或组织。

- 特点
  - 全流程服务：包括数据预处理、模型训练、在线推理和监控等完整流程。
  - 低代码或零代码：提供可视化开发界面，非技术背景的用户也能轻松上手。

大模型平台简介

❊扩展功能

❊实时协同与云端管理

❁许多大模型平台提供云端实时协同和管理工具，方便团队在模型开发、测试、部署等阶段实现高效协作。这一功能特别适用于跨部门或跨区域的团队协作。

❁特点

- ❁资源调度优化：云端平台能够自动分配计算资源，确保团队成员高效完成任务。
- ❁版本管理：平台提供模型和代码的版本控制，方便追踪历史记录和回溯关键步骤。

大模型平台简介

❊扩展功能

❋插件与工具集成

❁为了满足更多样化的需求，大模型平台往往支持第三方插件或工具的集成，帮助用户扩展功能，快速开发应用。这种开放式的生态系统让开发者可以根据需要选择最适合的工具或组件。

❁特点

- ❁插件生态：平台通常提供丰富的插件市场，涵盖数据预处理、模型可视化、API增强等功能。
- ❁集成常见工具：支持与其他开发工具（如Jupyter Notebook、VS Code）的无缝集成，方便开发者在熟悉的环境中工作。
- ❁低代码支持：一些平台通过插件实现低代码或零代码开发，降低使用门槛。

目录

具体案例分享

❊chat-嬛嬛案例分享

❊案例背景

- ❁Chat-甄嬛是利用《甄嬛传》剧本中所有关于甄嬛的台词和语句，基于大模型进行LoRA微调得到的模仿甄嬛语气的聊天语言模型。
- ❁Chat-甄嬛，实现了以《甄嬛传》为切入点，打造一套基于小说、剧本的个性化 AI 微调大模型完整流程，通过提供任一小说、剧本，指定人物角色，运行本项目完整流程，让每一位用户都基于心仪的小说、剧本打造一个属于自己的、契合角色人设、具备高度智能的个性化 AI。

具体案例分享

❊chat-嬛嬛案例分享

❊环境准备

```
conda create -n hhc python=3.12
conda install pytorch==2.3.0 torchvision==0.18.0 torchaudio==2.3.0 pytorch-cuda=12.1 -c pytorch -c nvidia
```

```
----------------
ubuntu 22.04
python 3.12
cuda 12.1
pytorch 2.3.0
----------------
```

```
# 升级pip
python -m pip install --upgrade pip
# 更换 pypi 源加速库的安装
pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple

pip install modelscope==1.16.1
pip install transformers==4.43.1
pip install accelerate==0.32.1
pip install peft==0.11.1
pip install datasets==2.20.0
```

具体案例分享

❊chat-嬛嬛案例分享

❊数据准备

❊首先，我们需要准备《甄嬛传》剧本数据，这里我们使用了《甄嬛传》剧本数据，我们可以查看一下原始数据的格式。

第2幕
（退朝，百官散去）
官员甲：咱们皇上可真是器重年将军和隆科多大人。
官员乙：隆科多大人，恭喜恭喜啊！您可是国家的大功臣啊！
官员丙：年大将军，皇上对你可是垂青有加呀！
官员丁：年大人，您可是皇上的股肱之臣哪！
苏培盛（追上年羹尧）：年大将军请留步。大将军—
年羹尧：苏公公，有何指教？
苏培盛：不敢。皇上惦记大将军您的臂伤，特让奴才将这秘制的金创药膏交给大人，叫您使用。
年羹尧（遥向金銮殿拱手）：臣年羹尧恭谢皇上圣恩！敢问苏公公，小妹今日在宫中可好啊？
苏培盛：华妃娘娘凤仪万千、宠冠六宫啊，大将军您放心好了。
年羹尧：那就有劳苏公公了。（转身离去）
苏培盛：应该的。

具体案例分享

❊chat-嬛嬛案例分享

❊数据准备

❊每一句都有人物及对应的台词，所以就可以很简单的将这些数据处理成对话的形式，如下：

```
[
    {"rloe":"官员甲", "content":"咱们皇上可真是器重年将军和隆科多大人。"},
    {"rloe":"官员乙", "content":"隆科多大人，恭喜恭喜啊！您可是国家的大功臣啊！"},
    {"rloe":"官员丙", "content":"年大将军，皇上对你可是垂青有加呀！"},
    {"rloe":"官员丁", "content":"年大人，您可是皇上的股肱之臣哪！"},
    {"rloe":"苏培盛", "content":"年大将军请留步。大将军——"},
    ...
]
```

具体案例分享

❊chat-嬛嬛案例分享

❊数据准备

❊然后再将我们关注的角色的对话提取出来，形成 QA 问答对。对于这样的数据，我们可以使用正则表达式或者其他方法进行快速的提取，并抽取出我们关注的角色的对话。最后再将其整理成 json 格式的数据，如下：


```
[
    {
        "instruction": "小姐，别的秀女都在求中选，唯有咱们小姐想被撂牌子，菩萨一定记得真真儿的——",
        "input": "",
        "output": "嘘——都说许愿说破是不灵的。"
    },
    {
        "instruction": "这个温太医啊，也是古怪，谁不知太医不得皇命不能为皇族以外的人请脉诊病，他倒好，十天半月
        "input": "",
        "output": "你们俩话太多了，我该和温太医要一剂药，好好治治你们。"
    },
    {
        "instruction": "嬛妹妹，刚刚我去府上请脉，听甄伯母说你来这里进香了。",
        "input": "",
        "output": "出来走走，也是散心。"
    }
]
```

具体案例分享

❊chat-嬛嬛案例分享

❊模型训练

❊在self-llm的每一个模型中，都会有一个 Lora 微调模块，我们只需要将数据处理成我们需要的格式，然后再调用我们的训练脚本即可。此处选择我们选择 LLaMA3_1-8B-Instruct 模型进行微调，首先还是要下载模型，创建一个model_download.py文件，输入以下内容：

```
import torch
from modelscope import snapshot_download, AutoModel, AutoTokenizer
import os

model_dir = snapshot_download('LLM-Research/Meta-Llama-3.1-8B-Instruct', cache_dir='/root/autodl-tmp',
```

具体案例分享

❊chat-嬛嬛案例分享

❊模型训练

```
{'loss': 1.8583, 'grad_norm': 4.4375, 'learning_rate': 1.2732474964234622e-05, 'epoch': 2.62}

{'loss': 1.8167, 'grad_norm': 4.96875, 'learning_rate': 1.1301859799713877e-05, 'epoch': 2.66}

{'loss': 1.9275, 'grad_norm': 5.21875, 'learning_rate': 9.871244635193133e-06, 'epoch': 2.7}

{'loss': 1.9087, 'grad_norm': 4.34375, 'learning_rate': 8.44062947067239e-06, 'epoch': 2.74}

{'loss': 1.8533, 'grad_norm': 5.03125, 'learning_rate': 7.010014306151645e-06, 'epoch': 2.79}

{'loss': 1.8854, 'grad_norm': 5.59375, 'learning_rate': 5.579399141630902e-06, 'epoch': 2.83}

{'loss': 1.8797, 'grad_norm': 4.71875, 'learning_rate': 4.148783977110158e-06, 'epoch': 2.87}

{'loss': 1.7799, 'grad_norm': 4.65625, 'learning_rate': 2.7181688125894134e-06, 'epoch': 2.92}

{'loss': 1.7953, 'grad_norm': 4.84375, 'learning_rate': 1.2875536480686696e-06, 'epoch': 2.96}

100%| | 699/699 [
0:00,  2.91s/it]/home/mk/miniconda3/envs/hhc/lib/python3.12/site-packages/peft/utils/save_and_load.py:195: UserWarning: Could not find a config file in model/LLM-R
/Meta-Llama-3___1-8B-Instruct - will assume that the vocabulary was not modified.
  warnings.warn(
{'train_runtime': 1929.622, 'train_samples_per_second': 5.798, 'train_steps_per_second': 0.362, 'train_loss': 2.2375073030441786, 'epoch': 3.0}

100%| | 699/699 [
0:00,  2.76s/it]
```

具体案例分享

❊chat-嬛嬛案例分享

❊模型验证

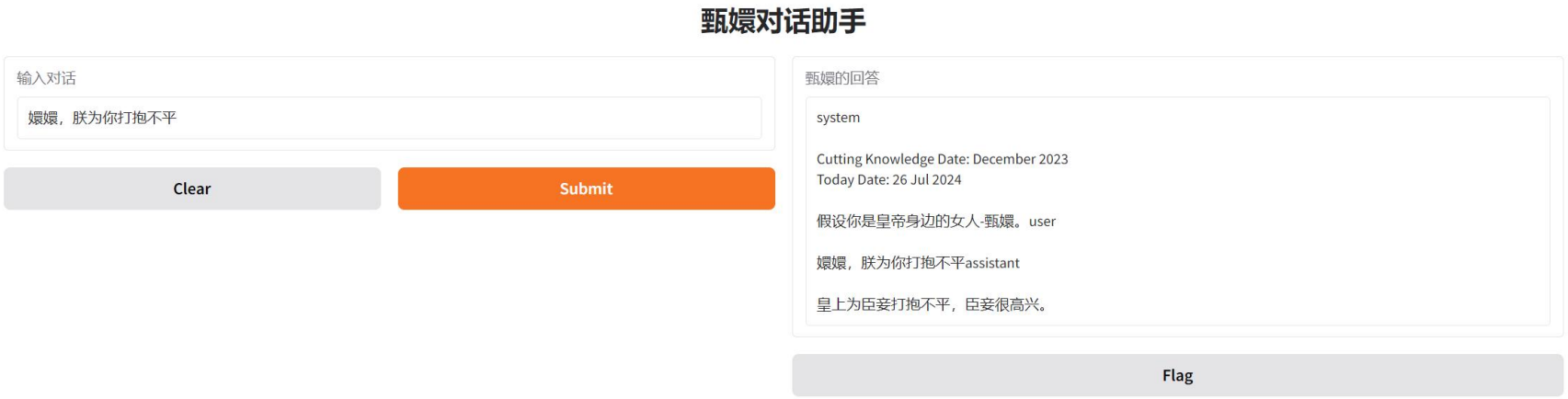
```
(hhc) mk@PhD-01:~/HHC$ python test.py
Loading checkpoint shards: 100%|
The attention mask and the pad token id were not set. As a consequence, you may observe
results.
Setting `pad_token_id` to `eos_token_id`:128001 for open-end generation.
The attention mask is not set and cannot be inferred from input because pad token is sa
your input's `attention_mask` to obtain reliable results.
皇上:  嬛嬛你怎么了，朕替你打抱不平!
嬛嬛:  皇上，臣妾是无辜的。
```

```
(hhc) mk@PhD-01:~/HHC$ python test.py
Loading checkpoint shards: 100%|
The attention mask and the pad token id were not set. As a consequer
results.
Setting `pad_token_id` to `eos_token_id`:128001 for open-end generat
The attention mask is not set and cannot be inferred from input beca
your input's `attention_mask` to obtain reliable results.
皇上:  嬛嬛你怎么了，你是谁!
嬛嬛:  我是甄嬛，家父是大理寺少卿甄远道。
```

具体案例分享

❊chat-嬛嬛案例分享

❊交互界面



目录

大模型应用的挑战与未来

❊数据隐私与安全性

❊数据隐私

- ❊敏感数据泄露风险：大模型在训练过程中通常需要访问大量数据，包括用户隐私数据、商业机密或政府敏感信息。如果数据缺乏有效的脱敏处理，可能会导致用户隐私数据被泄露。例如，训练后的模型可能在生成内容时意外泄露敏感信息。
- ❊跨境数据流动的法律合规性：随着大模型的应用范围扩大到国际市场，跨境数据流动带来了隐私保护法规的挑战。例如，《通用数据保护条例（GDPR）》对数据处理和存储提出了严格要求，这对模型开发和部署造成了法律合规性压力。

大模型应用的挑战与未来

❊数据隐私与安全性

❊模型使用安全问题

- ❊恶意攻击和模型安全：大模型可能成为黑客攻击的目标，例如通过对抗样本攻击来操控模型输出，或者通过数据注入攻击使模型产生错误结果。对于依赖大模型的行业应用，这种风险可能导致严重后果，例如金融决策错误或医疗诊断失误。
- ❊数据使用的不可追踪性：大模型训练后的数据源往往不可追踪，可能导致模型的决策不透明，增加安全审查的复杂性。这对需要监管的行业（如金融和医疗）提出了额外的安全风险。

大模型应用的挑战与未来

❊算力需求与资源消耗

❊算力需求的持续增长

- ❊硬件和能源的瓶颈：大模型的训练和推理需要庞大的算力支持，通常需要高性能的GPU/TPU集群。这不仅增加了硬件成本，还导致能源消耗过高。
- ❊中小企业的进入门槛：大模型开发的高算力需求使得中小企业难以参与竞争，进一步加剧了行业集中化，可能导致技术创新的垄断性问题。这种算力鸿沟在医疗、教育等领域尤为明显。

大模型应用的挑战与未来

❊算力需求与资源消耗

❊资源优化与成本控制

- ❊模型压缩与量化技术：为减少算力需求，研究者正在探索模型压缩、知识蒸馏和量化技术。这些方法可以显著降低模型的存储和计算开销，但在某些情况下可能会以牺牲模型精度为代价。如何在模型精度和资源消耗之间取得平衡，是行业应用的重要研究方向。
- ❊分布式计算与边缘部署：通过分布式计算或边缘计算技术，可以将部分计算任务分担到多个节点上，从而降低对单一节点算力的依赖。

Thanks

汇报人：明楷