**专利申请技术交底书**

|  |  |  |  |
| --- | --- | --- | --- |
| **一、专利名称** | **基于大模型的机器学习流水线自动化** | | |
| **二、专利的相关信息** | | | |
| 专利类型 | 发明专利　　 　实用新型专利 发明+实用新型 | | |
| 是否费减 | 是 否 | | |
| 申请人 |  | | |
| 发明人 |  | | |
| 第一发明人 | 姓名 | 身份证号 | 国籍 |
| 技术联系人 | 姓名 | 电话 | 电子邮箱 |
| **三、缩略语和关键术语定义**  **（请列出本专利中所出现的缩略的英文全称及中文定义（没有可不填写））** | | | |
| LLM： Large Language Model 大语言模型 ML: Machine Learning 机器学习  AutoML: Auto Machine Learning自动化机器学习 | | | |
| **四、专利背景（**与本专利相近的已有技术及其缺陷描述） | | | |
| 已有技术描述 | 写明已有技术中与本专利技术最接近的方案，有附图请结合附图描述 | | |
| **Data Interpreter 技术方案介绍**    Data Interpreter 是一种创新的基于大语言模型（LLM）的数据科学解决方案，旨在通过自动化手段调整和优化数据处理过程，以提升数据科学问题解决的效率和准确性。它集成了三项关键技术，推动了数据科学的技术进步，具体包括：   1. 动态规划与层次图结构： Data Interpreter 采用动态规划与层次图结构来实现数据的实时适应性。这一技术能够根据输入数据的特性和变化，智能调整数据处理流程，从而确保在面对复杂和不断变化的数据集时，能够实时地适应和优化解决方案。 2. 动态集成工具： 通过动态集成多种工具和技术，Data Interpreter 在执行过程中能够不断提升代码的执行效率和技术熟练度。这一过程不仅增强了其处理任务的能力，还丰富了模型所需的专业知识，确保在不同的数据科学任务中都能够提供精准且高效的解决方案。 3. 逻辑不一致性识别与反馈优化： Data Interpreter 能够在处理数据时及时识别逻辑不一致性，并通过经验反馈来不断优化其表现。每一次处理过程中的反馈信息都会被记录下来，形成一个不断积累的知识库，从而在后续任务中提升整体效率和准确度。   性能评估: Data Interpreter 在多种数据科学任务和现实世界应用场景中经过了广泛的评估，并且展现了显著的性能优势。与现有的开源基线相比，Data Interpreter 在机器学习任务中的表现有了显著提升，从传统基线的 0.86 提高至 0.95。此外，在 MATH 数据集上的表现提高了 26%，而在开放式任务中，性能提升更是达到了 112%。  **auto-sklearn 技术方案介绍**  auto-sklearn 是一个基于 scikit-learn 的自动化机器学习（AutoML）工具，它通过自动选择、优化和组合多个机器学习算法和预处理方法，简化了机器学习模型的开发过程。利用 auto-sklearn，用户可以在不需要深厚机器学习背景的情况下，快速地对数据进行建模并实现高效的预测任务。  主要特点   1. 自动化模型选择和优化： Auto-Sklearn 采用了贝叶斯优化算法（Bayesian Optimization），能够根据数据集的特性自动选择最合适的算法和参数。这使得它能够在有限的时间内，找到最优的机器学习模型，提升模型性能。 2. 模型组合（Ensemble）： Auto-Sklearn 通过生成多个候选模型，并将它们组合成一个强大的集成模型，从而进一步提高预测准确性。集成学习方法（Ensemble Learning）可以有效地减少单一模型的偏差和方差，提高模型的鲁棒性。 3. 自动特征工程： 除了自动选择最优模型外，Auto-Sklearn 还集成了多种自动化的特征处理和工程方法。它能够自动进行特征选择、特征缩放、缺失值处理等数据预处理任务，从而减少了手动调试的时间。 4. 灵活的扩展性和兼容性： 作为基于 scikit-learn 的工具，Auto-Sklearn 完全兼容 scikit-learn 中的所有算法，并能够与其他机器学习库和框架结合使用。其良好的扩展性使得用户可以根据实际需求进行定制。 5. 高效的计算资源管理： 在大规模数据集的训练过程中，Auto-Sklearn 通过并行化处理和资源调度，有效利用多核计算资源，加快模型的训练速度，同时保持较高的模型质量。 | | |
| 已有技术的问题及其缺陷描述 | ①客观评价，现有技术方案的缺点是相对于本专利的优点来说的，本专利不能解决的缺点不必写；  ②不能单纯讲缺陷，要结合产生缺陷的原因来描述。 | | |
| 一、Data Interpreter模块所存在的局限性  作为LLM代理系统的Data Interpreter未能充分发挥大语言模型的规划潜力。系统仅能机械地按照预设顺序执行任务列表，缺乏对机器学习任务进行智能化资源调配的能力。同时，系统在代码生成方面表现出较大的随机性和不确定性。即便经过多轮迭代优化，其生成的代码质量和任务配置方案仍难以达到理想效果，系统整体性能提升遇到瓶颈。这反映出当前技术方案在智能决策和自适应优化方面存在明显不足。  二、Auto-Sklearn框架的技术缺陷  Auto-Sklearn框架虽在固定流程的组件选择方面具有一定优势，但整体灵活性不足。具体表现在：其在模型选择、超参数优化以及集成学习设置等方面的自适应能力有待提高；在数据预处理环节，系统对非数值型数据的处理能力较弱，需要较多人工干预。这些限制显著降低了系统在处理多样化数据时的自动化程度和运行效率。  通过上述分析可见，现有技术方案在自动化水平、智能决策能力和环境适应性等关键方面均存在显著不足，难以满足当前智能计算领域对高效能、高适应性系统的迫切需求。这些技术局限正是本专利方案着力解决的核心问题。本专利提出的创新技术方案将在后续章节详细阐述。 | | |
| **五、本专利技术方案的详细阐述** | | | |
| **5.1本专利所要解决的技术问题**  针对现有技术方案的缺陷，说明本专利要解决的技术问题 | | | |
| 基于对前述技术缺陷的深入分析，本专利致力于解决以下关键技术问题：  一、增强系统的智能规划能力 针对现有Data Interpreter作为LLM代理系统在规划能力方面的不足，本专利创新性地引入蒙特卡洛树搜索（MCTS）算法来优化AutoML过程。该方案使代理系统能够智能地规划实验流程并迭代完善策略，从而实现对机器学习解决方案空间的高效探索。通过将MCTS与LLM的决策能力相结合，显著提升了系统在任务规划和资源调配方面的智能化水平。  二、提升代码生成的质量与稳定性 本专利采用创新的树形搜索结构来组织解决方案空间，实现了在每个技术阶段的智能化探索与方案生成。这种结构化的搜索机制不仅提高了生成代码的多样性，更通过实验反馈迭代机制持续优化解决方案，确保了代码质量的稳定性和可靠性。  三、增强系统灵活性与环境适应能力 通过建立动态化和多样化的策略体系，本专利使系统能够根据中间执行结果自适应地调整工作流程。这种智能化的自适应机制显著提升了系统应对不同场景和需求变化的能力，使其具备更强的环境适应性。  四、实现多层次的规划与优化机制 本专利创新性地结合了分阶段规划和迭代优化机制，在每个技术阶段都能探索并生成更优的解决方案。这种多层次的规划优化方法提供了更精细的控制粒度，显著提升了系统在复杂任务处理中的效能。  五、优化实验状态管理与资源利用 为提升系统运行效率并确保性能稳定性，本专利实现了细粒度的代码复用机制。通过系统性地缓存和管理每个实验配置的阶段代码，使框架在新配置与现有配置共享组件时能够高效复用已有代码。这一创新不仅优化了资源利用效率，还有效解决了LLM系统的非确定性问题，确保了性能表现的一致性和可预测性。 | | | |
| **5.2本专利提供的完整技术方案（请结合附图说明）** | | | |
| ①专利是关于结构的，请描述本专利所包含的各个元件，各元件之间的结构关系或者电路连接关系，描述专利技术的工作原理（有附图的请结合附图来描述）  ②专利是关于方法或流程的，请描述该方法或者流程所包括的所有步骤以及各步骤的详细情况，涉及软件开发过程的，以描述流程图及相关的设计说明为主（有流程图的请务必将流程图附上） | | | |
| 1. 系统组成： 本专利包括以下主要组件：   * ****大型语言模型（LLM）****：用于生成机器学习任务的洞察和代码。 * ****蒙特卡洛树搜索（MCTS）模块****：用于在树状结构的搜索空间中进行决策，以优化机器学习流程。 * ****实验执行器（Experiment Executor）****：根据LLM生成的配置执行机器学习实验，并反馈结果。  2. 工作原理：  * ****洞察提议（Insight Proposal）****：LLM根据问题描述和数据集信息生成一系列洞察，这些洞察构成 * 了搜索空间。洞察按照机器学习的不同阶段进行组织,如数据探索、数据预处理、特征工程和模型训练 * 等，形成树状结构。 * ****树状搜索空间构建****：MCTS模块将洞察组织成树状结构，每个节点代表一个潜在的解决方案路径。 * ****选择（Selection）****：使用修改版的UCT算法（UCT-DP）从搜索树中选择一个节点进行探索。 * ****扩展（Expansion）****：从选定节点生成子节点，代表不同的模型规范和训练策略。 * ****模拟（Simulation）****：实验执行器根据选定的配置执行完整的机器学习实验，并产生评估分数。 * ****反向传播（Backpropagation）****：将模拟结果的绩效分数反馈回搜索树，更新节点的值和访问次数。  3. 方法流程：  1. ****输入问题描述和数据集信息****：提供给LLM以生成搜索空间。 2. ****生成搜索空间****：LLM根据输入生成洞察，形成搜索空间。 3. ****初始化树结构****：使用洞察初始化MCTS的树结构。 4. ****执行多次模拟****：通过选择、扩展、模拟和反向传播步骤，进行多次模拟以探索不同的配置。 5. ****选择最佳解决方案****：根据模拟结果选择表现最佳的节点作为最终解决方案。 | | | |
| **5.3本专利技术方案带来的有益效果**  发明的效果，是与背景技术相比较的结果。应当清楚、有根据地说明发明与背景技术相比所具有的优点和（或者）积极效果。例如：产率、质量、精度、效率等的提高，能量、原材料的节省，环境污染的改善等等。 | | | |
| 1. 提高自动化水平：本专利通过整合LLM代理和MCTS，能够自动化地探索和优化机器学习管道，减少人工干预，使得非专家用户也能有效地设计可靠的机器学习解决方案。 2. 增强搜索效率：通过MCTS，本专利能够在庞大的决策空间中高效地导航，平衡探索（测试新策略）和利用（改进已知良好策略）之间的关系，从而更智能地选择下一个有前景的配置进行测试。 3. 提升解决方案质量：本专利通过迭代实验和策略优化，能够逐步改进解决方案，类似于人类专家基于持续反馈测试和改进策略的过程，从而发现基于实验反馈的最优路径。 4. 适应性和灵活性：本专利能够根据中间结果调整工作流程，允许在新信息出现时适应性地调整，这在处理独特数据集或特定任务要求时尤为重要。 5. 多样化的解决方案：本专利生成的解决方案具有多样性，因为它能够迭代地探索和改进整个机器学习管道，而不是仅限于单次尝试或固定的管道。 6. 广泛的适用性：虽然本专利是为机器学习设计的，但其核心方法可以适应广泛的顺序决策问题，只要这些问题可以表示为具有标量奖励的树结构。 7. 实验验证的有效性：在20个机器学习数据集上的广泛评估表明，本专利在各种机器学习任务中都能持续提供优越的性能，与传统的AutoML系统和现有的LLM-based方法相比，本专利在所有数据集上实现了65%到80%的胜率。 8. 成本效益：本专利实现了在保持低计算成本的同时找到准确的模型，这对于需要快速部署和更新模型的应用场景（如数据库系统）非常有用。 9. 可扩展性：本专利的设计允许它在不同的大型语言模型上表现一致，这表明它可以根据不同用户的偏好和可用性灵活地使用不同的LLM。 | | | |
| **六、针对第五部分的技术方案，是否还有别的替代方案同样能完成专利目的** | | | |
| 遗传算法(Genetic Algorithm)   1. 将机器学习管道的不同组件和超参数编码为一个"基因",多个基因组成一个"种群" 2. 通过选择、交叉和变异等遗传操作,优化种群中的基因,逐代搜索最优管道 3. 评估函数根据验证集性能为每个基因打分,高分基因有更大概率被选择和遗传 4. 经过多代进化,种群收敛到性能较优的机器学习管道   强化学习(Reinforcement Learning)   1. 将AutoML问题建模为一个序列决策过程,每个决策对应管道的一个组件或超参数 2. 通过与环境互动收集奖励反馈,训练一个RL模型来自动生成最优决策序列 3. 状态为当前的管道配置,动作为可选的组件或超参数,奖励为验证集性能 4. RL模型可选择Q-Learning、Policy Gradient或Actor-Critic等经典算法   贝叶斯优化(Bayesian Optimization)   1. 将搜索空间建模为一个高斯过程,预测性能与配置参数的函数关系 2. 引入acquisition function平衡探索新的配置和利用已知的好配置 3. 基于高斯过程的后验分布,选择下一个最有潜力的配置进行评估 4. 重复上述过程直到达到预算或性能阈值,输出性能最优的配置 | | | |
| **七、本专利的技术关键点和欲保护点是什么** | | | |
| · 一种基于MCTS和LLM的自动化机器学习系统及其工作方法,包括:   1. 通过LLM从问题描述生成搜索空间的方法 2. 在树状搜索空间中使用MCTS搜索最优机器学习管道的方法 3. 集成数据预处理、特征工程和模型训练等完整机器学习流程的自动化实验执行方法   · MCTS算法的改进,包括:   1. UCT-DP节点选择策略,优先探索深层节点的方法 2. 基于洞察池采样扩展MCTS节点,提升搜索效率的方法 3. 自适应调整探索-利用权重,动态引导搜索方向的方法   · 阶段级代码缓存和自动恢复技术,包括:   1. 缓存阶段级粒度的代码,最大化结果复用的方法 2. 自动恢复和重跑阶段代码,确保一致性的方法 3. 应对LLM不确定性,提升实验间性能稳定性的方法 | | | |
| **八、参考文献（如专利/标准/论文等）** | | | |
| <https://arxiv.org/abs/2410.17238>  <https://arxiv.org/abs/2402.18679>  <https://www.weco.ai/blog/technical-report>  <https://arxiv.org/abs/2007.04074> | | | |