大语言模型的版本管理方法、装置、设备及介质

本申请要求在2025年02月21日提交中国专利局、申请号为202510193916.1的中国专利申请的优先权，以上申请的全部内容通过引用结合在本申请中。

技术领域

本申请涉及人工智能技术领域，例如涉及一种大语言模型的版本管理方法、装置、设备及介质。

背景技术

随着大语言模型在各行各业的广泛渗透，汽车金融领域对其应用的需求日益增长且愈发精细化。

大模型微调作为一种高效、经济的适配手段，通过利用基础模型并结合特定领域数据进行针对性训练，已成为该领域普及的应用模式。然而，在汽车金融这一特殊场景下，大模型微调仍面临诸多挑战，尤其是因时间节点以及功能等特性变化导致的版本需求差异，不仅提升了版本管理的复杂性，还使得版本回滚操作变得尤为棘手。

发明内容

本申请提供了一种大语言模型的版本管理方法、装置、设备及介质，能够大幅简化了版本管理的复杂度，提高了针对特定错误进行快速且精准回滚的能力。

根据本申请的一方面，提供了一种大语言模型的版本管理方法，包括：

获取目标租户内大语言模型训练任务对应的版本信息和元数据；其中，版本信息包括时间戳信息和地域信息；所述元数据包括性能指标数据；

根据所述时间戳信息、所述地域信息以及所述性能指标数据确定大语言模型的版本回滚策略；

根据所述版本回滚策略对所述大语言模型执行版本回滚操作，以完成所述大语言模型的版本回滚。

根据本申请的另一方面，提供了一种大语言模型的版本管理装置，包括：

数据获取模块，设置为获取目标租户内大语言模型训练任务对应的版本信息和元数据；其中，版本信息包括时间戳信息和地域信息；所述元数据包括性能指标数据；

策略确定模块，设置为根据所述时间戳信息、所述地域信息以及所述性能指标数据确定大语言模型的版本回滚策略；

回滚操作执行模块，设置为根据所述版本回滚策略对所述大语言模型执行版本回滚操作，以完成所述大语言模型的版本回滚。

根据本申请的另一方面，提供了一种电子设备，所述电子设备包括：

至少一个处理器；以及

与所述至少一个处理器通信连接的存储器；其中，

所述存储器存储有可被所述至少一个处理器执行的计算机程序，所述计算机程序被所述至少一个处理器执行，以使所述至少一个处理器能够执行本申请任一实施例所述的大语言模型的版本管理方法。

根据本申请的另一方面，提供了一种计算机可读存储介质，所述计算机可读存储介质存储有计算机指令，所述计算机指令设置为使处理器执行时实现本申请任一实施例所述的大语言模型的版本管理方法。

本申请实施例的技术方案，通过获取目标租户内大语言模型训练任务对应的版本信息和元数据；其中，版本信息包括时间戳信息和地域信息；所述元数据包括性能指标数据；根据所述时间戳信息、所述地域信息以及所述性能指标数据确定大语言模型的版本回滚策略；根据所述版本回滚策略对所述大语言模型执行版本回滚操作，以完成所述大语言模型的版本回滚。本技术方案，可以根据大语言模型的性能数据与时间戳信息确定对应的版本回滚策略，从而能够大幅简化了版本管理的复杂度，提高了针对特定错误进行快速且精准回滚的能力。

附图说明

图 1 是根据本申请实施例一提供的一种大语言模型的版本管理方法的流程

图；

图 2 是根据本申请实施例二提供的一种大语言模型的版本管理方法的流程图；

图 3 是根据本申请实施例三提供的一种大语言模型的版本管理装置的结构示意图；

图 4 是根据本申请实施例四提供的一种电子设备的结构示意图。

具体实施方式

为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分的实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都应当属于本申请保护的范围。

需要说明的是，本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”以及“目标”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换，以便这里描述的本申请的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外，术语“包括”和“具有”及其任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

实施例一

图 1 是根据本申请实施例一提供的一种大语言模型的版本管理方法的流程图，本实施例可适用于对大语言模型进行版本管理的情况，该方法可以由一种大语言模型的版本管理装置来执行，该一种大语言模型的版本管理装置可以采用硬件和/或软件的形式实现，该一种大语言模型的版本管理装置可配置于具有数据处理能力的电子设备中。如图 1 所示，该方法包括：

S110、获取目标租户内大语言模型训练任务对应的版本信息和元数据。

其中，版本信息包括时间戳信息和地域信息；元数据包括性能指标数据。时间戳信息可以包括每个训练任务所对应版本的训练开始时间、结束时间以及部署时间。版本信息通常指的是大语言模型训练任务所使用的模型版本信息。元数据可以是指大语言模型训练任务的数据，具体可以包括训练数据、算法参数、性能指标等数据。地域信息可以是指大语言模型使用的地理区域或市场信息。可以理解的，本实施例中对于不同地区的业务要求存在差异，因此不同地区所对应的大语言模型也存在不同。大语言模型是指使用大量文本数据训练的深度学习模型，使得该模型可以生成自然语言文本或理解语言文本的含义。本实施例中的大语言模型是指在多租户服务环境中，为租户提供定制化自然语言处理服务的先进模型。

其中，目标租户可以任务是训练平台上的任一租户。本实施例中对每个组合进行进一步细化租户建的资源分配管理以及权限管理，以实现资源的高效利用和租户间的资源隔离。本实施例对训练平台的各个租户设计了一套增强的资源隔离方式。

关于资源隔离部分，本实施例中在利用 Kubernetes（k8s）强大的容器编排与资源管理能力基础上，进行了针对性的优化与创新。具体的，本实施例中针对训练平台的租户数据库的建立与库级别的隔离。首先，新租户通过训练平台的注册系统完成信息录入后，平台会自动触发数据库创建流程，为每个新租户生成一个独立的数据库实例。这些数据库实例在物理和逻辑层面均实现了严格的库级别隔离，确保租户数据的安全。接着，引入了基于角色的访问控制（RBAC）机制，以进一步细化租户间的权限管理。根据租户的业务需求和职责划分，定义了管理员、开发者、测试者等多种角色，并为每个角色分配了明确的权限边界。这些权限边界通过 k8s 的 RBAC 机制得以实施，即通过定义 Role 和 RoleBinding 来实现精细的权限管理。在实施细节上，通过确保角色定义细致到具体的 API 资源、操作类型和范围，以实现权限的精确控制。同时，我们还定期审计权限使用情况，及时发现并纠正任何异常或滥用权限的行为，确保租户间的操作互不干扰。在资源分配与管理方面，利用 k8s 的 Custom Resource Definition（CRD）功能，定义了自定义的资源类型“TenantResource”。这一创新举措使得我们能够根据租户的具体需求，为每种“TenantResource”配置专门的资源配额，包括 CPU、内存、存储等关键资源。在实施细节上，“TenantResource”

的定义中包含了租户标识、资源类型、资源配额等关键信息，以确保资源分配的准确性和合理性。同时，还根据租户的业务规模、增长趋势和峰值需求来动态调整资源配额的配置，以满足租户不断变化的业务需求。此外，还制定了合适的调度策略，综合考虑资源的可用性、租户的优先级以及亲和性/反亲和性规则，以实现资源的高效利用和租户间的隔离。最后，在网络安全层面，同样给予了高度重视。通过选择支持虚拟网络隔离的k8s网络插件(如Calico或Cilium)，为每个租户创建了一个独立的虚拟网络。这些虚拟网络在逻辑上相互隔离，有效防止了租户间的网络攻击和数据泄露风险。在实施细节上，根据租户的业务需求和网络拓扑结构来精心规划虚拟网络的创建过程，并确保网络策略的配置细致到具体的网络端口、协议和IP地址范围。同时，还定期监控网络流量和异常行为，及时发现并处理任何潜在的网络安全威胁，确保租户的网络环境安全稳定。

本实施中每个租户所对应的物理资源不同，大语言模型会部署在各自租户对应的物理资源上，本实施例中可以利用k8s的Annotations功能，为每个训练任务记录了详细的版本信息和元数据，包括训练数据、算法参数、性能指标等。具体的，在在利用k8s的Annotations记录版本信息和元数据时，增加每个版本的时间戳，包括训练开始时间、结束时间以及部署时间，这有助于追踪模型在不同时间段的表现，并为基于时间的回滚策略提供数据支持。为每个版本添加地域标签，表示该模型适用的地理区域或市场，这有助于在需要针对特定地区进行模型回滚时快速定位。因此，本实施例中在对不同的租户进行版本管理处理时，可以对应获取目标租户的大语言模型的每个训练任务所对应的详细的版本信息训练数据、算法参数、性能指标等数据。

S120、根据时间戳信息、地域信息以及性能指标数据确定大语言模型的版本回滚策略。

其中，版本回滚策略可以认为是在大语言模型需要进行回滚操作时所确定的最合适的模型版本。

本实施例中可以根据时间戳信息、地域信息以及性能指标数据分析大语言模型的各个版本下的模型性能数据，可以选择模型性能最优的或者问题最少的模型版本作为回滚目标；若存在多个候选版本，也可以通过进一步的测试或者

性能评估等方式确定出最佳回滚版本，以得到大语言模型的版本回滚策略。

在本实施例中，当需要对大语言模型进行回滚操作时，系统不仅会根据时间戳信息、地域信息以及性能指标数据定位到最合适的回滚版本，还会自动分析错误原因，并生成一份回滚建议报告，以便于提供给用户参考。此外，本实施例中还引入了版本链的概念，允许用户在回滚时选择特定的版本链进行回滚，从而避免了错误数据或算法对后续版本的影响。在特殊业务场景下，如信用评估模型更新时，还设计了基于时间窗口的回滚策略，确保在不影响业务连续性的前提下，完成模型的平稳回滚。

S130、根据版本回滚策略对大语言模型执行版本回滚操作，以完成大语言模型的版本回滚。

本实施例中可以在确定出的版本回滚策略中的最佳模型版本后，先停止大语言模型当前版本的大语言模型服务，以避免数据冲突或不一致，然后基于对大语言模型执行版本回滚操作，将大语言模型恢复到指定的最佳模型版本，并验证回滚后的模型文件和数据是否完整、正确，从而完成对大语言模型的平稳回滚，以确保在不影响业务连续性的情况下，保证了大语言模型的模型性能。

本申请实施例的技术方案，通过获取目标租户内大语言模型训练任务对应的版本信息和元数据；其中，版本信息包括时间戳信息和地域信息；元数据包括性能指标数据；根据时间戳信息、地域信息以及性能指标数据确定大语言模型的版本回滚策略；根据版本回滚策略对大语言模型执行版本回滚操作，以完成大语言模型的版本回滚。本技术方案，可以根据大语言模型的性能数据与时间戳信息确定对应的版本回滚策略，从而能够大幅简化了版本管理的复杂度，提高了针对特定错误进行快速且精准回滚的能力。

实施例二

图 2 是根据本申请实施例二提供的一种大语言模型的版本管理方法的流程图，本实施例以上述实施例为基础进行优化。具体优化为：根据时间戳信息、地域信息以及性能指标数据确定大语言模型的版本回滚策略，包括：根据时间戳信息和性能指标数据确定大语言模型中各版本的目标性能数据；基于各版本的目标性能数据和/或地域信息确定大语言模型的版本回滚策略。如图 2 所示，

该方法包括：

S210、获取目标租户内大语言模型训练任务对应的版本信息和元数据。

其中，版本信息包括时间戳信息和地域信息；元数据包括性能指标数据。

S220、根据时间戳信息和性能指标数据确定大语言模型中各版本的目标性能数据。

其中，目标性能数据可以认为是大语言模型在各个版本对应的指定性能数据，示例性的，目标性能数据可以是业务连续性以及信用评估准确性等性能数据，本实施例中可以根据用户的实际需求设定对应的目标性能数据。

本实施例中可以通过性能指标数据进行评估大语言模型的在不同版本下的信用评估结果，并结合时间戳信息确定在时间窗口内不同信用评估结果的变化，从而确定回滚到不同版本对模型性能数据的影响结果。

在本实施例中，可选的，根据时间戳信息和性能指标数据确定大语言模型中各版本的目标性能数据，包括：根据性能指标数据评估大语言模型在各版本的信用评估影响结果；根据信用评估影响结果和时间戳信息确定大语言模型的目标性能数据。

其中，信用评估影响结果可以是指不同版本所对应信用评估结果的预期变化的结果。本实施例中信用评估影响结果可以通过性能指标数据和机器学习算法预测模型确定出不同版本的信用评估影响结果。本实施例中可以将各个版本对应的性能指标数据和机器学习算法得到各个版本的信用评估影响结果，然后结合时间戳信息分析在各个时间段内大语言模型的信用评估影响结果变化情况，然后通过根据时间序列分析和版本对比的结果，从而确定大语言模型每个版本的目标性能数据。

本实施例中通过这样的设置，可以结合时间戳信息和性能指标信息分析各个版本的大语言模型的目标性能数据，以便于通过分析对比确定大语言模型的最佳回滚版本。

在本实施例中，可选的，性能指标数据包括准确指标数据、召回指标数据以及调和指标数据；根据性能指标数据评估大语言模型在各版本的信用评估影响结果，包括：将各个版本所对应的准确指标数据、召回指标数据以及调和指

标数据输入至信用评估预测模型，得到各个版本的信用评估影响结果。

其中，准确指标数据可以认为是大语言模型的准确率数据，可以用于评估模型处理结果正确率的指标。召回指标数据可以认为是大语言模型的召回率数据，又可以称为称查全率，是模型正确识别为正类的实例（真正类）占所有实际正类实例的比例。调和指标数据可以是指F1分数，其表示为准确率与召回率的调和平均数，通过调和指标数据兼顾了两个指标，以降低单一指标所带来的误导性。

本实施例中在元数据中包含信用评估的关键性能指标数据，如准确率、召回率以及F1分数等指标数据，以及这些指标随时间的变化趋势以便于评估大语言模型模型在信用评估任务中的性能，并为回滚决策提供依据。

本实施例中可以将各个版本所对应的准确率、召回率以及F1分数等指标数据输入至训练好的信用评估预测模型中，以得到各个版本对应的信用评估影响结果。

进一步的，本实施例中对于信用评估预测模型，设计一套基于时间窗口的回滚策略。例如，在业务高峰期前或特定时间段内，系统自动评估当前大语言模型的性能，并根据预设的阈值决定是否进行回滚。在进行回滚操作时，可以选择最近一个在时间窗口内表现稳定且信用评估性能良好的版本进行部署。

本实施例中通过这样的设置，可以利用历史数据和机器学习算法预测回滚到不同版本后信用评估结果的预期变化，帮助做出更明智的回滚决策，提高了回滚操作的可靠性。

S230、基于各版本的目标性能数据和/或地域信息确定大语言模型的版本回滚策略。

本实施例中在版本链中，不仅记录版本的线性关系，还标注每个版本的地域适用性和目标性能数据，可以依据各版本的目标性能数据，或者地域信息，或者目标性能数据和地域信息进行确定大语言模型的版本回滚策略。本实施例中版本回滚策略可以包括回滚的模型版本和具体的回滚方式。

具体的，本实施例中可以在回滚操作时可以选择对特定地域或信用评估性能影响最小的版本链进行回滚的操作。在依据地域信息确定版本回滚策略的过

程中，如果错误或性能下降仅影响某个特定地域，系统建议仅对该地域的模型进行回滚，以保持其他地域模型的稳定性和准确性。

在本实施例中，可选的，基于各版本的目标性能数据和/或地域信息确定大语言模型的版本回滚策略，包括：基于各版本的目标性能数据和/或地域信息生成大语言模型的回滚建议报告；根据回滚建议报告确定大语言模型的版本回滚策略。

其中，回滚建议报告可以包括多个回滚版本、回滚方式以及风险评估组成。本实施例中可以基于目标性能数据和地域信息，选择一个或多个历史版本作为回滚目标，然后根据实际需求确定回滚的具体方式，如直接回滚到指定版本、基于特定版本的微调等。直接回滚可能涉及版本控制系统的操作，而微调则需要在选定版本的基础上进行调整和优化。然后评估各个回滚版本执行时可能带来的风险，如性能下降、数据丢失等。这些风险应通过充分的测试来识别和缓解。本实施例中通过结合回滚目标、回滚方式以及风险评估等这三方面的内容生成大语言模型的回滚建议报告，然后根据回滚建议报告结合用户的实际需求确定出最适合的回滚版本和回滚方式作为版本回滚策略。

此外，在本实施例中生成回滚建议报告时，系统不仅分析错误原因，还结合时间窗口内的信用评估指标变化，评估回滚到不同版本对业务连续性和信用评估准确性的影响。

本实施例中通过这样的设置，根据各个版本的目标性能数据以及地域数据确定最适合大语言模型的版本回滚策略，能够确保大语言模型在不同任务和地域上的稳定性和性能。

S240、根据版本回滚策略对大语言模型执行版本回滚操作，以完成大语言模型的版本回滚。

在本实施例中，可选的，根据版本回滚策略对大语言模型执行版本回滚操作，以完成大语言模型的版本回滚，包括：基于版本回滚策略在大语言模型的第一回滚范围内进行版本回滚操作，以获得版本回滚后的模型性能数据；基于版本回滚后的模型性能数据对大语言模型继续执行版本回滚操作，以完成大语言模型的版本回滚。

其中，第一回滚范围可以是大语言模型所部署环境中的小范围内。本实施例中第一回滚范围可以根据实际需求进行设置。

本实施例中对于大规模部署的大语言模型，可以采用逐步回滚方式对大语言模型进行回滚操作，可以先在小范围内进行回滚测试，确保回滚后的模型性能稳定且符合预期，再逐步扩大回滚范围，从而完成大语言模型的版本回滚。示例性的，大规模部署的大语言模型可以认为是当前部署的大模型已经是大规模的应用了，比方说线上的服务器部署了几十台的服务都在用大语言模型。

本实施例中可以根据确定的版本回滚策略在大语言模型的预先设定的第一回滚范围内进行版本回滚操作，得到第一回滚范围内的进行版本回滚后的性能数据，根据第一回滚范围内的大语言模型在回滚后的模型性能数据确定是否满足预期效果，再逐步扩大回滚范围继续对大语言模型执行版本回滚操作，从而逐步实现大语言模型的版本回滚。

本实施例中通过这样的设置，可以对于大规模部署的大语言模型采用逐步回滚方式进行回滚操作，进一步保证了大语言模型的性能和回滚效果。

在本实施例中，可选的，基于版本回滚后的模型性能数据对大语言模型继续执行版本回滚操作，以完成大语言模型的版本回滚，包括：若版本回滚后的模型性能数据符合设定条件，则逐步向第二回滚范围内继续执行版本回滚操作，直至在大语言模型的全部范围内完成版本回滚操作。

其中，第二回滚范围大于第一回滚范围。设定条件可以是对模型性能数据设定的，可以是预先设定好的。本实施例中的设定条件可以是模型性能数据满足模型的预期数据，具体预期数据可以根据实际大语言模型进行配置。

本实施例中对于大规模部署的模型，采用逐步回滚策略，先在第一回滚范围内进行回滚测试，确保版本回滚后的模型性能数据符合设定提交，即版本回滚后的模型性能稳定且符合预期，再逐步扩大回滚范围，从而逐步对大语言模型的第二回滚范围内继续执行版本回滚操作，直至在大语言的全部范围内完成版本回滚操作；若版本回滚后的模型性能数据不符合设定条件，则可以停止对大语言模型进行版本回滚操作，可以依据模型性能进行调整版本回滚策略。

本实施例中在对大语言模型的回滚过程中，实时监控模型的性能和信用评

估指标，及时发现并处理任何异常情况。同时，收集用户反馈，对回滚效果进行持续评估和优化。

本实施例中通过这样的设置，可以对大规模部署的大语言模型采用逐步回滚方式，保证了大语言模型的回滚的灵活性和可靠性，也提高了大语言模型性能的稳定性。

本实施例中通过结合汽车金融的时间性、地域性和信用评估等属性，对版本回滚流程进行了进一步的完善和优化，不仅提高了回滚的灵活性和准确性，还确保了业务连续性和信用评估的准确性。

本申请实施例的技术方案，通过获取目标租户内大语言模型训练任务对应的版本信息和元数据；其中，版本信息包括时间戳信息和地域信息；元数据包括性能指标数据；根据时间戳信息和性能指标数据确定大语言模型中各版本的目标性能数据；基于各版本的目标性能数据和/或地域信息确定大语言模型的版本回滚策略。根据版本回滚策略对大语言模型执行版本回滚操作，以完成大语言模型的版本回滚。本技术方案，可以根据大语言模型的性能数据与时间戳信息确定对应的版本回滚策略，从而能够大幅简化了版本管理的复杂度，提高了针对特定错误进行快速且精准回滚的能力。

实施例三

图 3 是根据本申请实施例三提供的一种大语言模型的版本管理装置的结构示意图。如图 3 所示，该装置包括：

数据获取模块 310，设置为获取目标租户内大语言模型训练任务对应的版本信息和元数据；其中，版本信息包括时间戳信息和地域信息；元数据包括性能指标数据；

策略确定模块 320，设置为根据时间戳信息、地域信息以及性能指标数据确定大语言模型的版本回滚策略；

回滚操作执行模块 330，设置为根据版本回滚策略对大语言模型执行版本回滚操作，以完成大语言模型的版本回滚。

可选的，策略确定模块 320，包括：

目标性能数据确定单元，设置为根据时间戳信息和性能指标数据确定大语言模型中各版本的目标性能数据；

版本回滚策略确定单元，设置为基于各版本的目标性能数据和/或地域信息确定大语言模型的版本回滚策略。

可选的，目标性能数据确定单元，包括：

评估影响结果子单元，设置为根据性能指标数据评估大语言模型在各版本的信用评估影响结果；

性能数据确定子单元，设置为根据信用评估影响结果和时间戳信息确定大语言模型的目标性能数据。

可选的，性能指标数据包括准确指标、召回指标以及调和指标；

评估影响结果子单元，具体设置为：

将各个版本所对应的准确指标、召回指标以及调和指标输入至信用评估预测模型，得到各个版本的信用评估影响结果。

可选的，版本回滚策略确定单元，具体设置为：

基于各版本的目标性能数据和/或地域信息生成大语言模型的回滚建议报告；

根据回滚建议报告确定大语言模型的版本回滚策略。

可选的，回滚操作执行模块330，包括：

第一范围内回滚处理单元，设置为基于版本回滚策略在大语言模型的第一回滚范围内进行版本回滚操作，以获得版本回滚后的模型性能数据；

回滚操作继续执行单元，设置为基于版本回滚后的模型性能数据对大语言模型继续执行版本回滚操作，以完成大语言模型的版本回滚。

可选的，回滚操作继续执行单元，具体设置为：

若版本回滚后的模型性能数据符合设定条件，则逐步向第二回滚范围内继续执行版本回滚操作，直至在大语言模型的全部范围内完成版本回滚操作；其中，第二回滚范围大于第一回滚范围。

本申请实施例所提供的大语言模型的版本管理装置可执行本申请任意实施

例所提供的大语言模型的版本管理方法，具备执行方法相应的功能模块和有益效果。

实施例四

图 4 是根据本申请实施例四提供的一种电子设备的结构示意图。电子设备 10 旨在表示各种形式的数字计算机，诸如，膝上型计算机、台式计算机、工作台、个人数字助理、服务器、刀片式服务器、大型计算机、和其它适合的计算机。电子设备还可以表示各种形式的移动装置，诸如，个人数字处理、蜂窝电话、智能电话、可穿戴设备（如头盔、眼镜、手表等）和其它类似的计算装置。本文所示的部件、它们的连接和关系、以及它们的功能仅仅作为示例，并且不意在限制本文中描述的和/或者要求的本申请的实现。

如图 4 所示，电子设备 10 包括至少一个处理器 11，以及与至少一个处理器 11 通信连接的存储器，如只读存储器（ROM）12、随机访问存储器（RAM）13 等，其中，存储器存储有可被至少一个处理器执行的计算机程序，处理器 11 可以根据存储在只读存储器（ROM）12 中的计算机程序或者从存储单元 18 加载到随机访问存储器（RAM）13 中的计算机程序，来执行各种适当的动作和处理。在 RAM 13 中，还可存储电子设备 10 操作所需的各种程序和数据。处理器 11、ROM 12 以及 RAM 13 通过总线 14 彼此相连。输入/输出（I/O）接口 15 也连接至总线 14。

电子设备 10 中的多个部件连接至 I/O 接口 15，包括：输入单元 16，例如键盘、鼠标等；输出单元 17，例如各种类型的显示器、扬声器等；存储单元 18，例如磁盘、光盘等；以及通信单元 19，例如网卡、调制解调器、无线通信收发机等。通信单元 19 允许电子设备 10 通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

处理器 11 可以是各种具有处理和计算能力的通用和/或专用处理组件。处理器 11 的一些示例包括但不限于中央处理单元（CPU）、图形处理单元（GPU）、各种专用的人工智能（AI）计算芯片、各种运行机器学习模型算法的处理器、数字信号处理器（DSP）、以及任何适当的处理器、控制器、微控制器等。处理器 11 执行上文所描述的各个方法和处理，例如大语言模型的版本管理方法。

在一些实施例中，大语言模型的版本管理方法可被实现为计算机程序，其被有形地包含于计算机可读存储介质，例如存储单元 18。在一些实施例中，计算机程序的部分或者全部可以经由 ROM 12 和/或通信单元 19 而被载入和/或安装到电子设备 10 上。当计算机程序加载到 RAM 13 并由处理器 11 执行时，可以执行上文描述的大语言模型的版本管理方法的一个或多个步骤。备选地，在其他实施例中，处理器 11 可以通过其他任何适当的方式（例如，借助于固件）而被配置为执行大语言模型的版本管理方法。

本文中以上描述的系统和技术的各种实施方式可以在数字电子电路系统、集成电路系统、场可编程门阵列（FPGA）、专用集成电路（ASIC）、专用标准产品（ASSP）、芯片上系统的系统（SOC）、负载可编程逻辑设备（CPLD）、计算机硬件、固件、软件、和/或它们的组合中实现。这些各种实施方式可以包括：实施在一个或者多个计算机程序中，该一个或者多个计算机程序可在包括至少一个可编程处理器的可编程系统上执行和/或解释，该可编程处理器可以是专用或者通用可编程处理器，可以从存储系统、至少一个输入装置、和至少一个输出装置接收数据和指令，并且将数据和指令传输至该存储系统、该至少一个输入装置、和该至少一个输出装置。

用于实施本申请的方法的计算机程序可以采用一个或多个编程语言的任何组合来编写。这些计算机程序可以提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器，使得计算机程序当由处理器执行时使流程图和/或框图中所规定的功能/操作被实施。计算机程序可以完全在机器上执行、部分地在机器上执行，作为独立软件包部分地在机器上执行且部分地在远程机器上执行或完全在远程机器或服务器上执行。

在本申请的上下文中，计算机可读存储介质可以是有形的介质，其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的计算机程序。计算机可读存储介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备，或者上述内容的任何合适组合。备选地，计算机可读存储介质可以是机器可读信号介质。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、便携式计算机盘、硬盘、随机存取存储器（RAM）、只读存储器（ROM）、可擦除可编

程只读存储器（EPROM 或快闪存储器）、光纤、便捷式紧凑盘只读存储器（CD-ROM）、光学储存设备、磁储存设备、或上述内容的任何合适组合。

为了提供与用户的交互，可以在电子设备上实施此处描述的系统和技术，该电子设备具有：用于向用户显示信息的显示装置（例如，CRT（阴极射线管）或者 LCD（液晶显示器）监视器）；以及键盘和指向装置（例如，鼠标或者轨迹球），用户可以通过该键盘和该指向装置来将输入提供给电子设备。其它种类的装置还可以用于提供与用户的交互；例如，提供给用户的反馈可以是任何形式的传感反馈（例如，视觉反馈、听觉反馈、或者触觉反馈）；并且可以用任何形式（包括声输入、语音输入或者、触觉输入）来接收来自用户的输入。

可以将此处描述的系统和技术实施在包括后台部件的计算系统（例如，作为数据服务器）、或者包括中间件部件的计算系统（例如，应用服务器）、或者包括前端部件的计算系统（例如，具有图形用户界面或者网络浏览器的用户计算机，用户可以通过该图形用户界面或者该网络浏览器来与此处描述的系统和技术的实施方式交互）、或者包括这种后台部件、中间件部件、或者前端部件的任何组合的计算系统中。可以通过任何形式或者介质的数字数据通信（例如，通信网络）来将系统的部件相互连接。通信网络的示例包括：局域网（LAN）、广域网（WAN）、区块链网络和互联网。

计算系统可以包括客户端和服务器。客户端和服务器一般远离彼此并且通常通过通信网络进行交互。通过在相应的计算机上运行并且彼此具有客户端-服务器关系的计算机程序来产生客户端和服务器的关系。服务器可以是云服务器，又称为云计算服务器或云主机，是云计算服务体系中的一项主机产品，以解决了传统物理主机与 VPS 服务中，存在的管理难度大，业务扩展性弱的缺陷。

应该理解，可以使用上面所示的各种形式的流程，重新排序、增加或删除步骤。例如，本申请中记载的各步骤可以并行地执行也可以顺序地执行也可以不同的次序执行，只要能够实现本申请的技术方案所期望的结果，本文在此不进行限制。

1、一种大语言模型的版本管理方法，包括：

获取目标租户内大语言模型训练任务对应的版本信息和元数据；其中，版本信息包括时间戳信息和地域信息；所述元数据包括性能指标数据；

根据所述时间戳信息、所述地域信息以及所述性能指标数据确定大语言模型的版本回滚策略；

根据所述版本回滚策略对所述大语言模型执行版本回滚操作，以完成所述大语言模型的版本回滚。

2、根据权利要求 1 所述的方法，其中，根据所述时间戳信息、所述地域信息以及所述性能指标数据确定大语言模型的版本回滚策略，包括：

根据所述时间戳信息和所述性能指标数据确定所述大语言模型中各版本的目标性能数据；

基于所述各版本的目标性能数据和/或所述地域信息确定所述大语言模型的版本回滚策略。

3、根据权利要求 2 所述的方法，其中，根据所述时间戳信息和所述性能指标数据确定所述大语言模型中各版本的目标性能数据，包括：

根据所述性能指标数据评估所述大语言模型在各版本的信用评估影响结果；

根据所述信用评估影响结果和所述时间戳信息确定所述大语言模型的目标性能数据。

4、根据权利要求 3 所述的方法，其中，所述性能指标数据包括准确指标数据、召回指标数据以及调和指标数据；

根据所述性能指标数据评估所述大语言模型在各版本的信用评估影响结果，包括：

将各个版本所对应的所述准确指标数据、召回指标数据以及调和指标数据输入至信用评估预测模型，得到各个版本的信用评估影响结果。

5、根据权利要求 2 所述的方法，其中，基于所述各版本的目标性能数据和/或所述地域信息确定所述大语言模型的版本回滚策略，包括：

基于所述各版本的目标性能数据和/或所述地域信息生成大语言模型的回滚

建议报告；

根据所述回滚建议报告确定所述大语言模型的版本回滚策略。

6、根据权利要求 1 所述的方法，其中，根据所述版本回滚策略对所述大语言模型执行版本回滚操作，以完成所述大语言模型的版本回滚，包括：

基于所述版本回滚策略在大语言模型的第一回滚范围内进行版本回滚操作，以获得版本回滚后的模型性能数据；

基于所述版本回滚后的模型性能数据对所述大语言模型继续执行版本回滚操作，以完成所述大语言模型的版本回滚。

7、根据权利要求 6 所述的方法，其中，基于所述版本回滚后的模型性能数据对所述大语言模型继续执行版本回滚操作，以完成所述大语言模型的版本回滚，包括：

若所述版本回滚后的模型性能数据符合设定条件，则逐步向第二回滚范围内继续执行版本回滚操作，直至在所述大语言模型的全部范围内完成版本回滚操作；其中，所述第二回滚范围大于所述第一回滚范围。

8、一种大语言模型的版本管理装置，包括：

数据获取模块，设置为获取目标租户内大语言模型训练任务对应的版本信息和元数据；其中，版本信息包括时间戳信息和地域信息；所述元数据包括性能指标数据；

策略确定模块，设置为根据所述时间戳信息、所述地域信息以及所述性能指标数据确定大语言模型的版本回滚策略；

回滚操作执行模块，设置为根据所述版本回滚策略对所述大语言模型执行版本回滚操作，以完成所述大语言模型的版本回滚。

9、一种电子设备，所述电子设备包括：

至少一个处理器；以及

与所述至少一个处理器通信连接的存储器；其中，

所述存储器存储有可被所述至少一个处理器执行的计算机程序，所述计算机程序被所述至少一个处理器执行，以使所述至少一个处理器能够执行权利要

求 1-7 中任一项所述的大语言模型的版本管理方法。

10、一种计算机可读存储介质，所述计算机可读存储介质存储有计算机指令，所述计算机指令设置为使处理器执行时实现权利要求 1-7 中任一项所述的大语言模型的版本管理方法。