(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2024 年 3 月 28 日 (28.03.2024)

WIPO | PCT

(10) 国际公布号
WO 2024/061052 A1

(51) 国际专利分类号:
G06F 18/214 (2023.01)

(21) 国际申请号: PCT/CN2023/118186

(22) 国际申请日: 2023 年 9 月 12 日 (12.09.2023)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202211145581.9 2022年9月20日 (20.09.2022) CN

(71) 申请人: 华为技术有限公司 (**HUAWEI TECHNOLOGIES CO., LTD.**) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: 杨渊 (**YANG, Yuan**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 金修浪 (**JIN, Xiulang**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 张澍坤 (**ZHANG, Shukun**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 王泽 (**WANG, Ze**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(74) 代理人: 深圳市深佳知识产权代理事务所（普通合伙） (**SHENPAT INTELLECTUAL PROPERTY AGENCY**); 中国广东省深圳市罗湖区南湖街道春风路庐山大厦B座18C2、18D、18E、18E2, Guangdong 518001 (CN)。

WO 2024/061052 A1

(54) **Title:** MODEL PROCESSING METHOD AND DEVICE, AND MODEL-BASED DATA PROCESSING METHOD AND DEVICE

(54) 发明名称: 模型的处理方法、基于模型的数据处理方法及相关装置

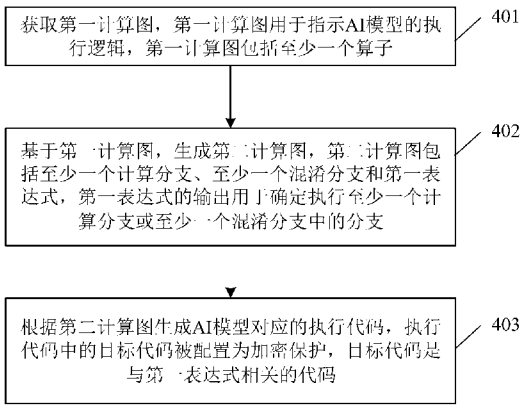


图 4

401 Obtain a first computational graph, the first computational graph being used for indicating an execution logic of an AI model, and the first computational graph comprising at least one operator

402 Generate a second computational graph on the basis of the first computational graph, wherein the second computational graph comprises at least one computation branch, at least one obfuscation branch, and a first expression, and an output of the first expression is used for determining to execute a branch in the at least one computation branch or the at least one obfuscation branch

403 According to the second computational graph, generate execution codes corresponding to the AI model, a target code in the execution codes being configured to be subjected to encryption protection, and the target code being a code related to the first expression

(57) **Abstract:** A model processing method, applied to the technical field of artificial intelligence (AI). In the method, on the basis of original computational logics of an AI model, new obfuscation and computational nodes are added, the execution relationship between original computational nodes and obfuscation nodes is determined by means of expressions, and a correct computational node can be executed only when an output of the expressions is correct. In this way, the execution sequence and the dependency relationship of operators in an original model can be fuzzified by means of the newly added obfuscation nodes, and model structure scrambling

[见续页]

(81) 指定国(除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明，要求每一种可提供的地区保护): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:
— 包括国际检索报告(条约第21条(3))。

is realized; and protection of the normal execution process of the whole AI model can be achieved simply by performing encryption protection on the newly added expressions, such that the additional performance overhead caused by model protection is reduced, and the popularization and application of the AI model are facilitated.

(57) 摘要: 一种模型的处理方法，应用于人工智能(Artificial Intelligence，AI)技术领域。在该方法中，在AI模型原有的计算逻辑的基础上，增加新的混淆计算节点，并且通过表达式来确定原有计算节点和混淆节点之间的执行关系，只有在表达式的输出正确时才能够执行到正确的计算节点。这样一来，通过新增的混淆节点能够模糊原模型中算子的执行顺序和依赖关系，实现模型结构加扰，并且只需要对新增的表达式进行加密保护，则能够实现对整个AI模型正常执行流程的保护，降低了模型保护所带来的额外性能开销，有利于AI模型的推广应用。

模型的处理方法、基于模型的数据处理方法及相关装置

本申请要求于 2022 年 9 月 20 日提交中国专利局、申请号为 202211145581.9、发明名称为“模型的处理方法、基于模型的数据处理方法及相关装置”的中国专利申请的优先权，其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（Artificial Intelligence，AI）技术领域，尤其涉及一种模型的处理方法及相关装置。

背景技术

随着 AI 理论和硬件算力的不断突破，AI 技术迎来了飞速发展。在计算机视觉、自然语言处理、语音识别等领域，AI 系统已经实现大规模部署，越来越多的厂商提供 AI 服务。一般来说，AI 服务提供商在本地完成模型训练和调优后，将 AI 模型部署到第三方平台(如终端设备、边缘设备和云服务器)上来提供推理服务。由于 AI 模型的设计和训练需要投入大量时间、数据和算力，因此如何防止 AI 模型在传输、存储以及运行等环节被窃取，已经成为 AI 服务提供商最为关心的问题。

为解决 AI 模型容易被窃取的问题，目前业界提出了一些模型机密性保护方案。例如，基于加解密算法的模型保护方案是采用加密算法对传输和存储过程中的 AI 模型进行加密，并且在执行 AI 模型的推理前将 AI 模型解密到内存中。这种方案虽然可以保护 AI 模型的机密性，但是每次执行 AI 模型的推理前都需要对整个 AI 模型进行解密，计算开销巨大，会导致 AI 模型的推理时延被大大地延长，严重地限制了 AI 模型的应用。

因此，如何实现高效的模型机密性保护成为亟待解决的问题。

发明内容

本申请提供了一种模型的处理方法，能够实现对整个 AI 模型正常执行流程的保护，且降低模型保护所带来的额外性能开销，有利于 AI 模型的推广应用。

本申请第一方面提供一种模型的处理方法，应用于服务器或终端设备等物理设备或虚拟设备上。以该方法应用于服务器为例，该方法包括：服务器通过解析 AI 模型的模型文件获取第一计算图，该第一计算图用于指示 AI 模型的执行逻辑，且第一计算图包括至少一个算子。即，第一计算图可以通过指示至少一个算子之间的依赖关系的方式来实现指示 AI 模型的执行逻辑。AI 模型的执行逻辑可以是指有序地执行该至少一个算子。

然后，基于第一计算图，服务器生成第二计算图，第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，第一表达式的输出用于指示执行至少一个计算分支中的分支或至少一个混淆分支中的分支。并且，第一计算图中的至少一个算子包含于第二计算图中的至少一个计算分支中，且至少一个计算分支与至少一个混淆分支具有不同的计算逻辑。在第一表达式的输出正确时，执行第二计算图中的至少一个计算分支，从而保证实际计算逻辑与原 AI 模型的计算逻辑相同；在第一表达式的输出错误时，执行第二计算图中的至少一个混淆分支，使得实际计算逻辑与原 AI 模型的计算逻辑不同，进而实现 AI 模型的保护。其中，混淆分支是一个具有算子的分支，能够将输入混淆分支的数据通过混淆分支中的算子进行计算得到输出数据。并且，由于混淆分支的计算逻辑与计算分支的计算逻辑不同，即混淆分支中的算子与计算分支中的算子不同，因此对于相同的输入数据，混淆分支和计算分支能够得到不同的输出数据。

其次，服务器根据第二计算图生成 AI 模型对应的执行代码，该执行代码中的目标代码被配置为加密保护，目标代码是与第一表达式相关的代码。由于在第二计算图中，第一表达式的输出能够影响第二计算图中的计算分支与混淆分支之间的执行情况，因此通过对第一表达式的目标代码进行加密保护，则能够实现对第二计算图的计算逻辑进行保护。简单来说，即便攻击者通过窃取 AI 模型对应的执行代码实现了 AI 模型的盗取，由于第一表达式对应的目标代码受到加密保护，因此攻击者无法获取到第一表达式的计算逻辑，从而无法得到正确的输出值，进而无法获取到 AI 模型正确的执行逻辑。

在 AI 模型原有的计算逻辑的基础上，增加与 AI 模型中的计算分支并列的混淆分支，并且通过表达式来确定计算分支和混淆分支之间的执行关系，只有在表达式的输出正确时才能够执行到正确的计算分支。这样一来，通过新增的混淆分支能够模糊原模型中显式的算子执行顺序和依赖关系，实现模型结构加扰，并且只需要对新增的表达式进行加密保护，则能够实现对整个 AI 模型正常执行流程的保护，降低了模型保护所带来的额外性能开销，有利于 AI 模型的推广应用。

在一种可能的实现方式中，第二计算图仅包括一个计算分支，且第一表达式的输出用于指示执行计算分支以及至少一个混淆分支中的一个分支。其中，第二计算图中的计算分支包括了第一计算图中的所有算子以及所有算子之间的依赖关系，因此通过执行第二计算图中的一个计算分支，则能够实现第一计算图中的计算逻辑。并且，第一表达式可以是分别与一个计算分支以及至少一个混淆分支连接，用于指示执行计算分支以及至少一个混淆分支中的任意一个分支。只有在第一表达式的输出为预设目标值时，才会执行第二计算图中的计算分支；否则，在第一表达式的输出不为预设目标值时，会执行第二计算图中的至少一个混淆分支中的一个分支。

其中，第一表达式与计算分支以及至少一个混淆分支连接可以是指在代码中定义了基于第一表达式的输出值来跳转到计算分支以及至少一个混淆分支。因此，在执行第一表达式后，能够跳转至执行计算分支或至少一个混淆分支。

本方案中，通过在第二计算图中设置一个或多个与计算分支并列的混淆分支，能够有效地模糊原模型中显式的算子执行顺序和依赖关系，实现对模型结构进行加扰，提高模型的机密性保护。

在一种可能的实现方式中，第二计算图包括多个计算分支，且第二计算图还包括第二表达式。其中，第一表达式的输出用于指示执行多个计算分支和至少一个混淆分支中的一个分支；第二表达式与多个计算分支和至少一个混淆分支连接，第二表达式用于指示第一表达式的循环执行次数，且第一表达式的输入与上一次执行的分支相关。

其中，第二表达式与计算分支以及至少一个混淆分支连接可以是指在代码中定义了执行多个计算分支以及至少一个混淆分支中的任意一个分支之后，均会跳转至执行第二表达式。因此，在执行任意一个计算分支或混淆分支之后，能够跳转至第二表达式，从而第二表达式指示是否继续循环执行第一表达式。

也就是说，第二计算图中的多个计算分支和至少一个混淆分支并列，且第一表达式用于控制需要执行的分支，第二表达式用于控制第一表达式的循环执行次数。因此，通过第二表达式和第一表达式的配合，能够实现依次执行相应的计算分支；且，通过并列的至少一个混淆分支，能够有效地模糊原模型中显式的算子执行顺序和依赖关系，实现对模型结构进行加扰，提高模型的机密性保护。

在一种可能的实现方式中，第二表达式的输入与上一次执行的分支相关，第二表达式的输出用于指示是否循环执行第一表达式。

在执行第二计算图的过程中，基于输入值，运行第一表达式，得到第一表达式的输出值；然后，基于第一表达式的输出值确定执行的分支。在执行一个分支后，执行第二表达式，以确定是否继续循环执行第一表达式。在确定继续循环第一表达式的情况下，基于执行分支所得到的输出值，继续执行第一表达式，以确定下一个需要执行的分支。通过循环执行上述的步骤，直至基于第二表达式的输出值确定终止循环第一表达式，从而实现依次执行多个计算分支，使得第二计算图的计算逻辑与第一计算图的计算逻辑相同。

在一种可能的实现方式中，第一表达式的初始输入包括第一数值，第一数值被配置为加密保护。例如，第一数值被配置为运行于可信执行环境中；或者，第一数值被配置为采用加密算法进行加密。

也就是说，第一表达式的初始输入可以是固定的，只有第一表达式的初始输入为第一数值时，第一表达式才能够输出正确值，从而使得第二计算图的计算逻辑与第一计算图的计算逻辑相同。此外，通过将第一数值配置为加密保护，可以实现对第一表达式的输出进行保护，攻击者即便获取到了整个第二计算图，也会因为无法获取到第一表达式正确的输入值而无法获取到第二计算图正确的计算逻辑，从而实现模型的机密性保护。

在一种可能的实现方式中，目标代码被配置为运行于可信执行环境中，和/或所述目标代码被配置为进行代码混淆。当模型使用者的终端设备需要运行 AI 模型时，终端设备则可以将目标代码加载至可信

执行环境中运行，从而基于可信执行环境来实现对目标代码的加密保护。另外，在目标代码为被配置为进行代码混淆的情况下，目标代码会被转换为功能相同但难以阅读和理解的代码，从而实现对目标代码的保护，且不影响目标代码的正常执行。

在一种可能的实现方式中，目标代码被配置为采用加密算法加密保护。当模型使用者的终端设备需要运行 AI 模型时，终端设备需要先采用解密算法对目标代码进行解密，才能够得到解密后的代码，从而基于解密后的代码获取第一表达式。示例性地，加密算法例如可以为高级加密标准（Advanced Encryption Standard，AES）算法、数据加密标准（Data Encryption Standard，DES）算法、国际数据加密算法（International Data Encryption Algorithm，IDEA）以及 RSA 算法。

在一种可能的实现方式中，该方法还包括：服务器获取第三计算图，第三计算图用于指示 AI 模型的执行逻辑，第三计算图包括第一算子。其中，第三计算图与上述的第一计算图可以是 AI 模型中不同的两个计算图；第三计算图也可以是第一计算图的一个子计算图，即第三计算图所包括的第一算子属于第一计算图中的多个算子中的一个。

基于第三计算图，服务器生成第四计算图，第四计算图包括第二算子和第三表达式，第二算子是对第一算子的权重参数修改后得到的，第三表达式的输入包括第二算子的输出，且第三表达式的输出与第一算子在采用与第二算子相同的输入时的输出相同。简单来说，在第二算子的权重参数是对第一算子的权重参数进行修改得到的情况下，基于相同的输入，第二算子的输出与第一算子的输出必然是不相同的；因此，通过引入一个第三表达式，来将第二算子的输出转换为与第一算子的输出相同的值。

最后，服务器根据第二计算图和第四计算图生成 AI 模型对应的执行代码。

本方案中，通过对 AI 模型中算子的权重参数进行加扰，能够有效地混淆 AI 模型中显式的各个算子，有效地避免攻击者通过盗用 AI 模型中算子的方式来实现对 AI 模型的盗用，保护了 AI 模型的机密性。

在一种可能的实现方式中，第二算子的权重参数是基于第二数值对第一算子的权重参数修改后得到的，第三表达式的输入包括第二数值，且第二数值被配置为加密保护。

其中，在第三表达式的输入不正确的情况下，第三表达式无法将第二算子的输出转换为第一算子的输出。因此，通过对第一数值进行加密保护，则能够实现对第四计算图中的计算逻辑进行保护，从而实现对整个 AI 模型正常执行流程的保护，并降低了模型保护所带来的额外性能开销。

本申请第二方面提供一种基于模型的数据处理方法，应用于部署有 AI 模型的设备，例如服务器或终端设备等物理设备或虚拟设备。以该方法应用于终端设备为例，该方法包括：基于 AI 模型的执行代码，终端设备获取第二计算图，第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，第一表达式的输出用于指示执行至少一个计算分支中的分支或至少一个混淆分支中的分支，至少一个计算分支包括至少一个 AI 模型的算子，至少一个计算分支与至少一个混淆分支具有不同的计算逻辑，所述执行代码中的目标代码被配置为加密保护，所述目标代码是与所述第一表达式相关的代码；终端设备获取 AI 模型的输入数据；终端设备基于第二计算图对输入数据进行处理，得到 AI 模型的输出数据。

在一种可能的实现方式中，第二计算图仅包括一个计算分支；第一表达式的输出用于指示执行计算分支以及至少一个混淆分支中的一个分支。

在一种可能的实现方式中，第二计算图包括多个计算分支，且第二计算图还包括第二表达式；第一表达式的输出用于指示执行多个计算分支和至少一个混淆分支中的一个分支；第二表达式与多个计算分支和至少一个混淆分支连接，第二表达式用于指示第一表达式的循环执行次数。

在一种可能的实现方式中，第二表达式的输入与上一次执行的分支相关，第二表达式的输出用于指示是否循环执行第一表达式。

在一种可能的实现方式中，第一表达式的初始输入包括第一数值，第一数值被配置为加密保护。

在一种可能的实现方式中，目标代码被配置为运行于可信执行环境中，和/或所述目标代码被配置为进行代码混淆。

在一种可能的实现方式中，目标代码被配置为采用加密算法加密保护。终端设备可以对执行代码中的目标代码进行解密，得到解密后的代码；并且，终端设备执行解密后的代码，得到第一表达式。

在一种可能的实现方式中，该方法还包括：基于 AI 模型的执行代码，获取第四计算图，第四计算图包括第二算子和第三表达式，第二算子是对 AI 模型中第一算子的权重参数修改后得到的，第三表达式的输入包括第二算子的输出，且第三表达式的输出与第一算子在采用与第二算子相同的输入时的输出相同；基于第二计算图对输入数据进行处理，包括：基于第二计算图和第四计算图对输入数据进行处理。

在一种可能的实现方式中，第二算子的权重参数是基于第二数值对第一算子的权重参数修改后得到的，第三表达式的输入包括第二数值，且第二数值被配置为加密保护。

本申请第三方面提供一种模型的处理装置，包括：获取模块，用于获取第一计算图，所述第一计算图用于指示人工智能 AI 模型的执行逻辑，所述第一计算图包括至少一个算子；处理模块，用于基于所述第一计算图，生成第二计算图，所述第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，所述第一表达式的输出用于指示执行所述至少一个计算分支中的分支或所述至少一个混淆分支中的分支，所述至少一个算子包含于所述至少一个计算分支中，所述至少一个计算分支与所述至少一个混淆分支具有不同的计算逻辑；处理模块，用于根据所述第二计算图生成所述 AI 模型对应的执行代码，所述执行代码中与所述第一表达式相关的目标代码被配置为加密保护。

在一种可能的实现方式中，所述第二计算图仅包括一个计算分支；所述第一表达式的输出用于指示执行所述计算分支以及所述至少一个混淆分支中的一个分支。

在一种可能的实现方式中，所述第二计算图包括多个计算分支，且所述第二计算图还包括第二表达式；所述第一表达式的输出用于指示执行所述多个计算分支和所述至少一个混淆分支中的一个分支；所述第二表达式与所述多个计算分支和所述至少一个混淆分支连接，所述第二表达式用于指示所述第一表达式的循环执行次数，且所述第一表达式的输入与上一次执行的分支相关。

在一种可能的实现方式中，所述第二表达式的输入与上一次执行的分支相关，所述第二表达式的输出用于指示是否循环执行所述第一表达式。

在一种可能的实现方式中，所述第一表达式的初始输入包括第一数值，所述第一数值被配置为加密保护。

在一种可能的实现方式中，所述目标代码被配置为运行于可信执行环境中，和/或所述目标代码被配置为进行代码混淆。

在一种可能的实现方式中，所述目标代码被配置为采用加密算法加密保护。

在一种可能的实现方式中，所述获取模块，还用于获取第三计算图，所述第三计算图用于指示所述 AI 模型的执行逻辑，所述第三计算图包括第一算子；所述处理模块，还用于：基于所述第三计算图，生成第四计算图，所述第四计算图包括第二算子和第三表达式，所述第二算子是对所述第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二算子的输出，且所述第三表达式的输出与所述第一算子在采用与所述第二算子相同的输入时的输出相同； 根据所述第二计算图和所述第四计算图生成所述 AI 模型对应的执行代码。

在一种可能的实现方式中，所述第二算子的权重参数是基于第二数值对所述第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二数值，且所述第二数值被配置为加密保护。

本申请第四方面提供一种基于模型的数据处理装置，包括：获取模块，用于基于 AI 模型的执行代码，获取第二计算图，所述第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，所述第一表达式的输出用于指示执行所述至少一个计算分支中的分支或所述至少一个混淆分支中的分支，所述至少一个计算分支包括至少一个所述 AI 模型的算子，所述至少一个计算分支与所述至少一个混淆分支具有不同的计算逻辑，所述执行代码中与所述第一表达式相关的目标代码被配置为加密保护；所述获取模块，还用于获取所述 AI 模型的输入数据；所述处理模块，还用于基于所述第二计算图对所述输入数据进行处理，得到所述 AI 模型的输出数据。

在一种可能的实现方式中，所述第二计算图仅包括一个计算分支；所述第一表达式的输出用于指示执行所述计算分支以及所述至少一个混淆分支中的一个分支。

在一种可能的实现方式中，所述第二计算图包括多个计算分支，且所述第二计算图还包括第二表达式；所述第一表达式的输出用于指示执行所述多个计算分支和所述至少一个混淆分支中的一个分支；所述第二表达式与所述多个计算分支和所述至少一个混淆分支连接，所述第二表达式用于指示所述第一表达式的循环执行次数，且所述第一表达式的输入与上一次执行的分支相关。

在一种可能的实现方式中，所述第二表达式的输入与上一次执行的分支相关，所述第二表达式的输出用于指示是否循环执行所述第一表达式。

在一种可能的实现方式中，所述第一表达式的初始输入包括第一数值，所述第一数值被配置为加密保护。

在一种可能的实现方式中，所述目标代码被配置为运行于可信执行环境中，和/或所述目标代码被配置为进行代码混淆。

在一种可能的实现方式中，所述目标代码被配置为采用加密算法加密保护；所述处理模块，还用于对所述执行代码中的所述目标代码进行解密，得到解密后的代码；执行所述解密后的代码，得到所述第一表达式。

在一种可能的实现方式中，所述获取模块，还用于基于所述 AI 模型的执行代码，获取第四计算图，所述第四计算图包括第二算子和第三表达式，所述第二算子是对所述 AI 模型中第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二算子的输出，且所述第三表达式的输出与所述第一算子在采用与所述第二算子相同的输入时的输出相同；所述处理模块，还用于基于所述第二计算图和所述第四计算图对所述输入数据进行处理。

在一种可能的实现方式中，所述第二算子的权重参数是基于第二数值对所述第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二数值，且所述第二数值被配置为加密保护。

本申请第五方面提供一种电子设备，该电子设备包括：存储器和处理器；所述存储器存储有代码，所述处理器被配置为执行所述代码，当所述代码被执行时，所述电子设备执行如第一方面中的任意一种实现方式的方法。

本申请第六方面提供一种电子设备，该电子设备包括：存储器和处理器；所述存储器存储有代码，所述处理器被配置为执行所述代码，当所述代码被执行时，所述电子设备执行如第二方面中的任意一种实现方式的方法。

本申请第七方面提供一种 AI 系统，该 AI 系统包括：如第三方面任一实现方式所述的模型的处理装置以及如第四方面任一实现方式所述的基于模型的数据处理装置。

本申请第八方面提供一种计算机可读存储介质，该计算机可读存储介质中存储有计算机程序，当其在计算机上运行时，使得计算机执行如第一方面或第二方面中的任意一种实现方式的方法。

本申请第九方面提供一种计算机程序产品，当其在计算机上运行时，使得计算机执行如第一方面或第二方面中的任意一种实现方式的方法。

本申请第十方面提供一种芯片，包括一个或多个处理器。处理器中的部分或全部用于读取并执行存储器中存储的计算机程序，以执行上述第一方面或第二方面中的任意一种实现方式中的方法。

可选地，该芯片该包括存储器，该存储器与该处理器通过电路或电线与存储器连接。可选地，该芯片还包括通信接口，处理器与该通信接口连接。通信接口用于接收需要处理的数据和/或信息，处理器从该通信接口获取该数据和/或信息，并对该数据和/或信息进行处理，并通过该通信接口输出处理结果。该通信接口可以是输入输出接口。本申请提供的方法可以由一个芯片实现，也可以由多个芯片协同实现。

其中，第二方面至第十方面中任一种设计方式所带来的技术效果可参见上述第一方面中不同实现方式所带来的技术效果，在此不再赘述。

附图说明

图 1 为本申请实施例提供的一种模型文件明文部署的示意图；

图 2 为本申请实施例提供的一种模型文件加密部署的示意图；

图 3 为本申请实施例提供的一种模型的处理方法的应用场景示意图；

图 4 为本申请实施例提供的一种模型的处理方法的流程示意图；

图 5A 为本申请实施例提供的一种基于第一计算图生成第二计算图的结构示意图；

图 5B 为本申请实施例提供的一种基于第一计算图生成第二计算图的另一结构示意图；

图 6A 为本申请实施例提供的一种基于第一计算图生成第二计算图的另一结构示意图；

图 6B 为本申请实施例提供的一种第二计算图的运行示意图；

图 7A 为本申请实施例提供的一种基于第一计算图生成第二计算图的另一结构示意图；

图 7B 为本申请实施例提供的一种执行第二计算图的示意图；

图 8 为本申请实施例提供的一种基于第一计算图生成第二计算图的另一结构示意图；

图 9 为本申请实施例提供的一种基于第三计算图得到第四计算图的示意图；

图 10 为本申请实施例提供的一种基于模型的数据处理方法的流程示意图；

图 11 为本申请实施例提供的一种处理模型以及基于模型处理数据的流程示意图；

图 12 为本申请实施例提供的一种模型的处理装置的结构示意图；

图 13 为本申请实施例提供的一种基于模型的数据处理装置的结构示意图；

图 14 为本申请实施例提供的执行设备的一种结构示意图；

图 15 为本申请实施例提供的芯片的一种结构示意图；

图 16 为本申请实施例提供的一种计算机可读存储介质的结构示意图。

具体实施方式

为了使本申请的目的、技术方案及优点更加清楚明白，下面结合附图，对本申请的实施例进行描述。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。本领域普通技术人员可知，随着新应用场景的出现，本申请实施例提供的技术方案对于类似的技术问题，同样适用。

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”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描述在适当情况下可以互换，以便使实施例能够以除了在本申请图示或描述的内容以外的顺序实施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或模块的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或模块，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模块。在本申请中出现的对步骤进行的命名或者编号，并不意味着必须按照命名或者编号所指示的时间/逻辑先后顺序执行方法流程中的步骤，已经命名或者编号的流程步骤可以根据要实现的技术目的变更执行顺序，只要能达到相同或者相类似的技术效果即可。本申请中所出现的单元的划分，是一种逻辑上的划分，实际应用中实现时可以有另外的划分方式，例如多个单元可以结合成或集成在另一个系统中，或一些特征可以忽略，或不执行，另外，所显示的或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，单元之间的间接耦合或通信连接可以是电性或其他类似的形式，本申请中均不作限定。并且，作为分离部件说明的单元或子单元可以是也可以不是物理上的分离，可以是也可以不是物理单元，或者可以分布到多个电路单元中，可以根据实际的需要选择其中的部分或全部单元来实现本申请方案的目的。

为了便于理解，以下先介绍本申请实施例所涉及的技术术语。

（1）AI

AI 是利用数字计算机或者数字计算机控制的机器模拟、延伸和扩展人的智能，感知环境、获取知识并使用知识获得最佳结果的理论、方法、技术及应用系统。换句话说，AI 是计算机科学的一个综合技术，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。AI 也就是研究各种智能机器的设计原理与实现方法，使机器具有感知、推理与决策的功能。

（2）AI 模型

AI 模型是 AI 技术中的一种用于处理特定任务的技术手段。一般来说，AI 模型可以为由神经单元组成的神经网络。通常，AI 模型是在模型所有者的设备或平台 (如：服务器、虚拟机 (virtual machine，

VM)或容器(container)中进行训练得到的，训练好的 AI 模型会以模型文件的形式存储。在模型使用者的设备(如：终端设备、服务器或边缘设备、VM 或容器等)需要使用该 AI 模型时，可以是模型使用者的设备主动加载该 AI 模型的模型文件；也可以是模型所有者的设备主动向模型使用者的设备发送 AI 模型的模型文件，以使得模型使用者的设备能够加载并执行该 AI 模型的模型文件。

其中，服务器是一种物理机。VM 或容器都可以是在物理机的硬件资源上采用虚拟化的方式划分出来的虚拟化的设备。

终端设备(也可以称为用户设备(user equipment，UE))是一种具有无线收发功能的设备，可以部署在陆地上，包括室内或室外、手持或车载；也可以部署在水面上(如轮船 等)；还可以部署在空中 (例如飞机、气球和卫星上等)。示例性地，终端设备例如可以是智能手机 (mobile phone)、平板电脑(pad)、带无线收发功能的电脑、物联网设备、虚拟现实(virtual reality，VR)终端、增强现实(augmented reality，AR)终端、工业控制(industrial control)中的无线终端、无人驾驶(self driving)中的无线终端、远程医疗(remote medical)中的无线终端、智能电网(smart grid)中的无线终端、运输安全(transportation safety)中的无线终端、智慧城市(smart city)中的无线终端、智慧家庭(smart home)中的无线终端等。

（3）计算图

计算图是用图论语言表示数学函数的一种方式，即将计算过程图形化表示出来。一般来说，计算图被定义为一个有向图，由节点和边所构成。在计算图中，输入值和计算函数都以节点的形式出现，而节点的输出项之间的关系则以有向线段（即节点间的边）表示。

（4）表达式

表达式，是由数字、算符、数字分组符号（括号）、自由变量和约束变量等以能求得数值的有意义排列方法所得的组合。简单来说，表达式是由操作数和运算符组成的式子，是一个具有完整意义的计算机指令。例如，表达式可以为(x+6)*3*COS(1)/2*8+7。

（5）不透明谓词

不透明谓词是一个表达式。不透明谓词的值在执行到某处时，对于不透明谓词的编写者而言必然是已知的，但是编译器或者静态分析器则无法推断出这个值，只能在运行不透明谓词时才能确定不透明谓词的值。

（6）可信执行环境（Trusted Execution Environment，TEE）

可信执行环境是指通过软硬件方法在中央处理器中构建一个安全区域，保证其内部加载的程序和数据在机密性和完整性上得到保护。简单来说，可信执行环境是中央处理器内的一个安全区域，中央处理器用于确保可信执行环境中代码和数据的机密性和完整性都得到保护，即运行在可信执行环境中的代码和数据，是保密且不可篡改的。

（7）代码混淆(obfuscated code)

代码混淆亦称花指令，是将计算机程序的代码，转换成一种功能上等价，但是难于阅读和理解的形式的行为。代码混淆可以用于程序源代码，也可以用于程序编译而成的中间代码。执行代码混淆的程序被称作代码混淆器。

简单来说，代码混淆可以是指将代码中的各种元素（例如变量，函数，类）的名字改写成无意义的名字。比如，将代码中的元素改写成单个字母，或是简短的无意义字母组合，甚至改写成“__”这样的符号，使得阅读的人无法根据名字猜测其用途。代码混淆还可以是重写代码中的部分逻辑，将其变成功能上等价，但是更难理解的形式。比如，将 for 循环改写成 while 循环，将循环改写成递归，精简中间变量等等。代码混淆还可以是打乱代码的格式。比如删除空格，将多行代码挤到一行中，或者将一行代码断成多行等等。

目前，AI 服务提供商一般是在本地的模型训练环境中完成 AI 模型的训练和调优，得到 AI 模型对应的模型文件，然后通过模型文件的形式将 AI 模型部署至模型使用者的模型部署环境中（例如终端设备）。当 AI 模型的模型文件以明文形式部署于模型部署环境中时，容易受到不法分子的窃取。示例性地，请参阅图 1，图 1 为本申请实施例提供的一种模型文件明文部署的示意图。如图 1 所示，AI 服务提供商

在模型训练环境中训练得到 AI 模型，并生成 AI 模型的模型文件。然后，AI 服务提供商将明文的模型文件部署至模型部署环境中，以使得模型部署环境中的 AI 计算框架能够通过导入模型文件来实现执行 AI 模型。然而，由于模型文件是以明文形式部署在模型部署环境中，因此不法分子能够通过攻击模型部署环境从而直接从模型部署环境中窃取得到模型文件，进而得到窃取的模型。

为解决 AI 模型容易被窃取的问题，目前业界提出了一些模型机密性保护方案。示例性地，请参阅图 2，图 2 为本申请实施例提供的一种模型文件加密部署的示意图。如图 2 所示，AI 服务提供商在模型训练环境中训练得到 AI 模型，并生成 AI 模型的模型文件。然后，AI 服务提供商采用加密算法对模型文件进行加密，并将加密后的模型文件部署至模型使用者的模型部署环境中。当模型使用者需要使用 AI 模型时，通过应用程序触发模型文件的解密，并将模型文件解密至内存中，从而在内存中执行 AI 模型。

图 2 所示例的这种方案虽然可以保护 AI 模型的机密性，但是每次执行 AI 模型的推理前都需要对整个 AI 模型进行解密，计算开销巨大，会导致 AI 模型的推理时延被大大地延长，严重地限制了 AI 模型的应用。

基于此，本申请实施例提供了一种模型的处理方法，在 AI 模型原有的计算逻辑的基础上，增加与 AI 模型中的计算分支并列的混淆分支，并且通过表达式来确定计算分支和混淆分支之间的执行关系，只有在表达式的输出正确时才能够执行到正确的计算分支。这样一来，通过新增的混淆分支能够模糊原模型中显式的算子执行顺序和依赖关系，实现模型结构加扰，并且只需要对新增的表达式进行加密保护，则能够实现对整个 AI 模型正常执行流程的保护，降低了模型保护所带来的额外性能开销，有利于 AI 模型的推广应用。

请参阅图 3，图 3 为本申请实施例提供的一种模型的处理方法的应用场景示意图。如图 3 所示，在模型的混淆阶段，通过对原模型文件进行解析，获取到原模型文件对应的计算图。然后，通过本申请实施例所提供的模型的处理方法对获取得到的计算图进行处理，并且基于处理后的计算图生成混淆态模型文件。其中，混淆态模型文件可以部署于模型使用者环境中。这样，在模型的推理阶段，AI 应用程序能够获取到混淆态模型文件以及推理数据，并实现混淆模型的加载。通过执行本申请实施例提供的基于模型的数据处理方法，能够实现基于混淆模型执行推理，得到推理结果。

具体地，本申请实施例提供的模型的处理方法以及基于模型的数据处理方法均可以应用于电子设备或虚拟化设备上，该电子设备例如为上述的服务器以及终端设备；该虚拟化设备例如为上述的虚拟机和容器。

为了便于理解，以下将依次从模型的处理阶段和模型的使用阶段来介绍本申请实施例所提供的方法。并且，为了便于叙述，以下将以模型的处理阶段在服务器上执行，且模型的使用阶段在终端设备上执行为例，对本申请实施例提供的方法进行介绍。在实际应用中，模型的处理阶段并不限定于在服务器上执行，模型的使用阶段也不限定于在终端设备上执行。

请参阅图 4，图 4 为本申请实施例提供的一种模型的处理方法的流程示意图。如图 4 所示，该模型的处理方法包括以下的步骤 401-403。

步骤 401，获取第一计算图，第一计算图用于指示 AI 模型的执行逻辑，第一计算图包括至少一个算子。

本实施例中，服务器可以通过解析 AI 模型的模型文件来获取第一计算图。其中，第一计算图可以是指示 AI 模型的部分执行逻辑，第一计算图也可以是指示整个 AI 模型的全部执行逻辑。

一般来说，AI 模型中会包括多个计算单元，如：卷积单元、池化单元或加法单元等各种类型用于执行相应计算的单元，每个计算单元可以称为一个算子。AI 模型的执行逻辑也就是每个计算单元执行的先后顺序以及各个计算单元之间的依赖关系。因此，在采用计算图来表示 AI 模型中的计算单元以及计算单元之间的输入或输出关系的情况下，上述的第一计算图能够指示 AI 模型的执行逻辑。

可选的，第一计算图中可以是包括 AI 模型中的所有算子或部分算子。并且，第一计算图中的算子的类型可以有一种或多种，同一种类型的算子也可以有一个或多个。算子的类型通常表征算子的计算属

性，如：卷积类型、池化类型、加法类型、批标准化类型或线性修正类型等。其中，卷积类型的算子指的是用于做卷积运算的算子，池化类型的算子指的是用于做池化运算的算子，加法类型的算子指的是用于做加法运算的算子，批标准化类型的算子指的是用于做批标准化的算子，线性修正类型的算子指的是用于做线性修正的算子。此外，在第一计算图中，每个算子都会有一个唯一的标识或唯一的名称，如：卷积算子 1、卷积算子 2、池化算子 1 或加法算子 1 等。

可选的，在服务器执行本申请实施例提供的方法之前，用户可以是对 AI 模型中需要进行混淆保护的算子进行指定。例如，用户指定 AI 模型中的某一个算子或多个算子需要进行混淆保护；或者，用户指定 AI 模型中的某一类或多类算子是需要进行混淆保护的。这样，服务器在解析 AI 模型的模型文件时，能够根据用户所指定的需要进行混淆保护的算子，确定第一计算图，该第一计算图中包括需要进行混淆保护的算子。

步骤 402，基于第一计算图，生成第二计算图，第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，第一表达式的输出用于指示执行至少一个计算分支或至少一个混淆分支中的分支，至少一个算子包含于至少一个计算分支中，至少一个计算分支与至少一个混淆分支具有不同的计算逻辑。

本实施例中，在第二计算图中，第一表达式可以是与至少一个计算分支以及至少一个混淆分支连接。并且，在执行第二计算图的过程中，先执行第一表达式，并根据第一表达式的输出选择执行至少一个计算分支或至少一个混淆分支中的分支。只有在第一表达式的输出是预设目标值的情况下，才执行上述的至少一个计算分支；在第一表达式的输出不是预设目标值的情况下，则执行上述的至少一个混淆分支。其中，预设目标值可以是包括一个或多个值。

具体地，第一计算图中的至少一个算子包含于第二计算图的至少一个计算分支中。因此，在第一表达式的输出为预设目标值时，执行第二计算图中的至少一个计算分支，从而保证实际计算逻辑与原 AI 模型的计算逻辑相同；在第一表达式的输出不为预设目标值时，执行第二计算图中的至少一个混淆分支，使得实际计算逻辑与原 AI 模型的计算逻辑不同，进而实现 AI 模型的保护。

步骤 403，根据第二计算图生成 AI 模型对应的执行代码，执行代码中的目标代码被配置为加密保护，目标代码是与第一表达式相关的代码。

在得到第二计算图后，服务器可以根据第二计算图的计算逻辑生成 AI 模型对应的执行代码，以便于将 AI 模型部署至其他的设备上。其中，在服务器所生成的执行代码中，与第一表达式相关的目标代码被配置为加密保护，而执行代码中其他部分的代码则可以是不被配置为加密保护。

由于在第二计算图中，第一表达式的输出能够影响第二计算图中的计算分支与混淆分支之间的执行情况，因此通过对第一表达式的目标代码进行加密保护，则能够实现对第二计算图的计算逻辑进行保护。简单来说，即便攻击者通过窃取 AI 模型对应的执行代码实现了 AI 模型的盗取，由于第一表达式对应的目标代码受到加密保护，因此攻击者无法获取到第一表达式的计算逻辑，从而无法得到正确的输出值，进而无法获取到 AI 模型正确的执行逻辑。

本实施例中，在 AI 模型原有的计算逻辑的基础上，增加与 AI 模型中的计算分支并列的混淆分支，并且通过表达式来确定计算分支和混淆分支之间的执行关系，只有在表达式的输出正确时才能够执行到正确的计算分支。这样一来，通过新增的混淆分支能够模糊原模型中显式的算子执行顺序和依赖关系，实现模型结构加扰，并且只需要对新增的表达式进行加密保护，则能够实现对整个 AI 模型正常执行流程的保护，降低了模型保护所带来的额外性能开销，有利于 AI 模型的推广应用。

可选的，将与第一表达式相关的目标代码配置为加密保护的方式有多种。

在一种可能的实现方式中，与第一表达式相关的目标代码被配置为运行于可信执行环境中。当模型使用者的终端设备需要运行 AI 模型时，终端设备则将目标代码加载至可信执行环境中运行，从而基于可信执行环境来实现对目标代码的加密保护。

可以理解的是，由于可信执行环境是一种基于软硬件方法在中央处理器中所构建的一个安全区域，因此在目标代码被配置为运行于可信执行环境的情况下，需要模型使用者的终端设备本身支持可信执行环境，即对需要加载 AI 模型的终端设备具有一定的硬件要求。

在另一种可能的实现方式中，目标代码被配置为进行代码混淆。这样，在目标代码为被配置为进行

代码混淆的情况下，目标代码会被转换为功能相同但难以阅读和理解的代码，从而实现对目标代码的保护，且不影响目标代码的正常执行。

在另一种可能的实现方式中，目标代码被配置为采用加密算法加密保护。其中，加密算法例如可以为高级加密标准（Advanced Encryption Standard，AES）算法、数据加密标准（Data Encryption Standard，DES）算法、国际数据加密算法（International Data Encryption Algorithm，IDEA）以及 RSA 算法，本实施例并不对加密算法的类型做具体限定。当模型使用者的终端设备需要运行 AI 模型时，终端设备需要先采用解密算法对目标代码进行解密，才能够得到解密后的代码，从而基于解密后的代码获取第一表达式。

在一些示例中，服务器也可以是采用上述的多种实现方式同时对与第一表达式相关的目标代码进行加密保护。简单来说，目标代码被配置为运行于可信执行环境中，且目标代码同时被配置为采用加密算法加密保护。这样，在模型使用者的终端设备需要运行 AI 模型时，终端设备可以是在可信执行环境中对目标代码进行解密，得到解密后的代码，从而基于解密后的代码获取第一表达式；或者，终端设备可以是对目标代码进行解密，并将解密得到的代码在可信执行环境中运行，从而获取到第一表达式。

此外，除了对与第一表达式相关的目标代码进行加密保护之外，还可以对第一表达式本身进行保护。示例性地，第一表达式可以为不透明谓词，服务器采用不透明谓词来作为第一表达式。这样，在获取到第一表达式对应的目标代码时，无法通过目标代码推断出第一表达式的值，而是只有在运行目标代码的过程中才能够获取到第一表达式的值，从而实现第一表达式的隐藏保护。

以上介绍了通过生成新的计算图来对 AI 模型进行加密保护的过程，为便于理解，以下将详细介绍基于 AI 模型原有的计算图生成新的计算图的具体过程。

本实施例中，服务器基于第一计算图生成第二计算图的方式有多种。

实现方式 1，服务器生成具有一个计算分支、至少一个混淆分支和第一表达式的第二计算图。

其中，第二计算图中的计算分支包括了第一计算图中的至少一个算子，第一表达式的输出用于指示执行计算分支以及至少一个混淆分支中的一个分支。

简单来说，在实现方式 1 中，第二计算图中的一个计算分支包括了第一计算图中的所有算子以及所有算子之间的依赖关系，通过执行第二计算图中的一个计算分支，则能够实现第一计算图中的计算逻辑。并且，第一表达式可以是分别与一个计算分支以及至少一个混淆分支连接，用于指示执行计算分支以及至少一个混淆分支中的任意一个分支。只有在第一表达式的输出正确时，才会执行第二计算图中的计算分支；否则，在第一表达式的输出错误时，会执行第二计算图中的至少一个混淆分支中的一个分支。

示例性地，可以参阅图 5A，图 5A 为本申请实施例提供的一种基于第一计算图生成第二计算图的结构示意图。如图 5A 所示，第一计算图中包括依次连接的三个算子，分别为卷积算子、批归一化算子和线性修正算子。其中，卷积算子的输入为第一计算图的输入，批归一化算子的输入为卷积算子的输出，线性修正算子的输入为批归一化算子的输出，线性修正算子的输出为第一计算图的输出。在基于第一计算图所生成的第二计算图中，依次连接的卷积算子、批归一化算子和线性修正算子成为第二计算图中的一个计算分支。第二计算图中还包括一个与计算分支并列的混淆分支，且计算分支和混淆分支均与第一表达式连接。在第二计算图中，根据第一表达式输出的值，来确定所执行的分支。具体地，在第一表达式输出的值为预设目标值的情况下，则执行计算分支，从而实现如第一计算图所示的计算逻辑，使得第二计算图的输出与第一计算图的输出相同；在第一表达式输出的值不为预设目标值的情况下，则执行混淆分支，从而实现与第一计算图中不同的计算逻辑，使得第二计算图的输出与第一计算图的输出不同。

示例性地，可以参阅图 5B，图 5B 为本申请实施例提供的一种基于第一计算图生成第二计算图的另一结构示意图。如图 5B 所示，图 5B 中的第一计算图与图 5A 中所示的第一计算图相同；并且，图 5B 中第二计算图的计算分支与图 5A 中第一计算图的计算分支相同，区别在于图 5B 中的第二计算图包括多个混淆分支，即混淆分支 1、混淆分支 2…混淆分支 N。在图 5B 所示的第二计算图中，当第一表达式的输出为正确值时，执行计算分支；当第一表达式的输出为错误值 1 时，则执行混淆分支 1；当第一表达式的输出为错误值 2 时，则执行混淆分支 2；当第一表达式的输出为错误值 N 时，则执行混淆分支 N。

本方案中，通过在第二计算图中设置一个或多个与计算分支并列的混淆分支，能够有效地模糊原模型中显式的算子执行顺序和依赖关系，实现对模型结构进行加扰，提高模型的机密性保护。

实现方式 2，服务器生成具有多个计算分支、至少一个混淆分支和第一表达式的第二计算图。

其中，第一表达式与多个计算分支以及至少一个混淆分支连接，用于指示有序地执行有序地执行多个计算分支以及至少一个混淆分支中的多个分支。当第一表达式输出的值为预设目标值时，第一表达式则用于指示有序地执行多个计算分支；当第一表达式输出的值不为预设目标值时，第一表达式则用于指示有序地执行其他多个分支或者是乱序地执行多个计算分支。总的来说，只有在第一表达式的输出为预设目标值时，第二计算图的计算逻辑才与第一计算图的计算逻辑相同；在第一表达式的输出不为预设目标值时，第二计算图的计算逻辑则不与第一计算图的计算逻辑相同。

示例性地，请参阅图 6A，图 6A 为本申请实施例提供的一种基于第一计算图生成第二计算图的另一结构示意图。如图 6A 所示，图 6A 中的第一计算图与图 5A 中所示的第一计算图相同。并且，图 6A 中的第二计算图包括并列的多个计算分支和多个混淆分支，该多个计算分支中的每个计算分支包括一个算子。具体地，在图 6A 的第二计算图中，从左往右依次排列有六个分支，其中第一个分支为包括卷积算子的计算分支，第二个分支为混淆分支 1，第三个分支为包括批归一化算子的计算分支，第四个分支为混淆分支 2，第五个分支为包括线性修正算子的计算分支，第六个分支为混淆分支 3。此外，上述的六个分支还连接有一个判断式，该判断式用于判定已执行的分支数量；且该判断式与第一表达式连接，用于根据第一表达式的输出值确定每次执行完一个分支后下一个继续执行的分支。

请参阅图 6B，图 6B 为本申请实施例提供的一种第二计算图的运行示意图。如图 6B 所述，第二计算图的执行步骤共包括 7 个。步骤 1，运行第一表达式，并基于第一表达式的输出值确定执行包括卷积算子的计算分支。步骤 2，执行卷积算子，并基于判断式判断是否已执行三个分支。步骤 3，判断式基于第一表达式的输出值，确定需执行的第二个分支为包括批归一化算子的计算分支。步骤 4，执行批归一化算子，并基于判断式判断是否已执行三个分支。步骤 5，判断式基于第一表达式的输出值，确定需执行的第三个分支为包括线性修正算子的计算分支。步骤 6，执行线性修正算子，并基于判断式判断是否已执行三个分支。步骤 7，判断式将执行线性修正算子所得到的结果作为输出值输出。

由图 6B 可知，基于图 6A 中所示的第二计算图能够实现有序地执行第一计算图中所示的多个算子，从而使得第二计算图的计算逻辑与第一计算图的计算逻辑相同。

此外，在图 6A 的第一表达式的输出值为错误值时，第二计算图的计算逻辑则不为依次执行第一个分支、第三个分支以及第五个分支，从而使得第二计算图的计算逻辑与第一计算图的计算逻辑不同。例如，在第一表达式的输出值为错误值 1 时，第二计算图的计算逻辑可以为依次执行第二个分支、第四个分支以及第六个分支。又例如，在第一表达式的输出值为错误值 2 时，第二计算图的计算逻辑可以为依次执行第二个分支、第三个分支以及第四个分支。

实现方式 3，服务器生成具有多个计算分支、至少一个混淆分支、第一表达式和第二表达式的第二计算图。

其中，第一表达式的输出用于指示执行多个计算分支和至少一个混淆分支中的一个分支。第二表达式与多个计算分支和至少一个混淆分支连接，且第二表达式用于指示第一表达式的循环执行次数。并且，第一表达式的输入与上一次执行的分支相关。

具体地，第二表达式的输入与上一次执行的分支相关，第二表达式的输出用于指示是否循环执行第一表达式。

在执行第二计算图的过程中，基于输入值，运行第一表达式，得到第一表达式的输出值；然后，基于第一表达式的输出值确定执行的分支。在执行分支后，执行第二表达式，以确定是否继续循环执行第一表达式。在确定继续循环第一表达式的情况下，基于执行分支所得到的输出值，继续执行第一表达式，以确定下一个需要执行的分支。通过循环执行上述的步骤，直至基于第二表达式的输出值确定终止循环第一表达式，从而实现依次执行多个计算分支，使得第二计算图的计算逻辑与第一计算图的计算逻辑相同。

示例性地，请参阅图 7A，图 7A 为本申请实施例提供的一种基于第一计算图生成第二计算图的另一

结构示意图。如图 7A 所示，图 7A 中的第一计算图与图 5A 中所示的第一计算图相同。并且，图 7A 中的第二计算图包括并列的多个计算分支和多个混淆分支，该多个计算分支中的每个计算分支包括一个算子。此外，对于多个计算分支和多个混淆分支中的每一个分支，分支内还具有计算表达式（即 P1、P2、P3、P4、P5、P6 等表达式），这些计算表达式用于给变量 next 赋值。此外，第一表达式包括两个输入，一个输入是作为后续所执行的分支的输入，另一个输入则为变量 next，第一表达式用于基于变量 next 的值得到相应的输出。第二表达式则基于前一个所执行的分支内给变量 next 所赋予的值，确定是否继续第一表达式。

示例性地，请参阅图 7B，图 7B 为本申请实施例提供的一种执行第二计算图的示意图。如图 7B 所示，第一表达式为一个选择（switch）算子，用于基于变量 next 的值选择相应的分支。其中，第一表达式的初始输入为 next=P0，基于该初始输入，第一表达式选择执行包括卷积算子的分支。

在执行包括卷积算子的分支时，该分支同时将变量 next 的值赋为表达式 P1 的输出值。其中表达式 P1-表达式 P4 值均小于表达式 P5 的值，表达式 P6 的值大于表达式 P5 的值。

在执行完毕包括卷积算子的分支后，执行第二表达式，判断 next 的值是否大于等于表达式 P5 的输出值，并基于当前 next 的值小于表达式 P5 的输出值，选择循环执行第一表达式。

在第二次执行第一表达式时，第一表达式的输入包括执行卷积算子后的输出，以及 next=P1；基于 next=P1，第一表达式选择执行包括批归一化算子的分支，并将执行执行卷积算子后的输出作为批归一化算子的输入。

在执行包括批归一化算子的分支时，该分支同时将变量 next 的值赋为表达式 P3 的输出值。

在执行完毕包括批归一化算子的分支后，执行第二表达式，判断 next 的值是否大于等于表达式 P5 的输出值，并基于当前 next 的值小于表达式 P5 的输出值，选择循环执行第一表达式。

在第三次执行第一表达式时，第一表达式的输入包括执行批归一化算子后的输出，以及 next=P3；基于 next=P3，第一表达式选择执行包括线性修正算子的分支，并将执行执行批归一化算子后的输出作为线性修正算子的输入。

在执行完毕包括线性修正算子的分支后，执行第二表达式，判断 next 的值是否大于等于表达式 P5 的输出值，并基于当前 next 的值等于表达式 P5 的输出值，选择不再循环执行第一表达式，从而输出线性修正算子的输出值。

总的来说，在图 7B 所示的第二计算图中，通过变量 next、第一表达式和第二表达式的配合，能够实现有序地执行多个计算分支，使得第二计算图的计算逻辑与第一计算图的计算逻辑相同。

在上述的实现方式 1 和实现方式 2 中，第一表达式的初始输入可以包括第一数值，且第一数值被配置为加密保护。也就是说，第一表达式的初始输入可以是固定的，只有第一表达式的初始输入为第一数值时，第一表达式才能够输出正确值，从而使得第二计算图的计算逻辑与第一计算图的计算逻辑相同。此外，通过将第一数值配置为加密保护，可以实现对第一表达式的输出进行保护，攻击者即便获取到了整个第二计算图，也会因为无法获取到第一表达式正确的输入值而无法获取到第二计算图正确的计算逻辑，从而实现模型的机密性保护。

此外，第一表达式的初始输入也可以是包括计算表达式的输出，该计算表达式的输出是对第一数值进行处理得到的。同样地，只有在计算表达式的输入为第一数值时，第一表达式才能够输出正确值。

示例性地，以实现方式 1 为例，以下将结合附图介绍第一表达式的输入还包括第一数值时的第二计算图。请参阅图 8，图 8 为本申请实施例提供的一种基于第一计算图生成第二计算图的另一结构示意图。如图 8 所示，图 8 中的第一计算图与图 5A 中所示的第一计算图相同。图 8 中的第二计算图包括计算分支和混淆分支，其中第一表达式包括两个输入，一个输入用于作为后续计算分支或混淆分支的输入，另一个输入则作为第一表达式本身的输入。其中，另一个输入可以为第一数值或者计算表达式基于第一数值所得到的输出值。只有在第一表达式本身的输入正确的情况下，第一表达式的输入为第一数值或计算表达式的输入为第一数值时，第一表达式的输出才为预设目标值，即第二计算图才会执行到计算分支；否则，在第一表达式的输入不为第一数值或计算表达式的输入不为第一数值时，第一表达式的输出不为预设目标值，第二计算图会执行混淆分支，从而使得第二计算图的实际计算逻辑与第一计算图的计算逻

辑不同。

以上介绍了通过生成第二计算图的方式对AI模型结构进行加扰，从而实现对AI模型的机密性保护。以下将介绍从另外一个角度对AI模型的机密性进行保护的方式。

本实施例中，服务器可以是对AI模型中算子的权重参数进行加扰混淆，避免攻击者通过盗用AI模型中各个算子来实现对AI模型的盗用。

示例性地，服务器可以获取第三计算图，第三计算图用于指示AI模型的执行逻辑，且第三计算图包括第一算子。其中，第三计算图与上述的第一计算图可以是AI模型中不同的两个计算图；第三计算图也可以是第一计算图的一个子计算图，即第三计算图所包括的第一算子属于第一计算图中的多个算子中的一个。

基于第三计算图，生成第四计算图，第四计算图包括第二算子和第三表达式，第二算子是对第一算子的权重参数修改后得到的，第三表达式的输入包括第二算子的输出，且第三表达式的输出与第一算子在采用与第二算子相同的输入时的输出相同。也就是说，服务器对第三计算图中的第一算子的权重参数进行了修改，得到了第四计算图中的第二算子。并且，服务器还在第二算子后插入了第三表达式，以使得第二算子和第三表达式结合后的输出值能够与第一算子的输出值相同，即维持第四计算图的计算结果与第三计算图的计算结果不变。

简单来说，在第二算子的权重参数是对第一算子的权重参数进行修改得到的情况下，基于相同的输入，第二算子的输出与第一算子的输出必然是不相同的；因此，通过引入一个第三表达式，来将第二算子的输出转换为与第一算子的输出相同的值。

在得到第四计算图后，服务器可以是根据第二计算图和第四计算图生成AI模型对应的执行代码。

可选的，第二算子的权重参数是基于第二数值对第一算子的权重参数修改后得到的，第三表达式的输入包括第二数值，且第二数值被配置为加密保护。即，只有在第三表达式的输入为第二数值的情况下，第三表达式才能够将第二算子的输出转换为与第一算子的输出相同；否则，在第三表达式的输入不为第二数值的情况下，第三表达式无法将第二算子的输出转换为与第一算子的输出相同的值。

其中，第二数值被配置为加密保护的方式可以是与上述实施例中所介绍的第一表达式被配置加密保护的方式相同，即第二数值被配置为被配置为采用加密算法加密和/或运行于可信执行环境，具体请参考上述的实施例，在此不再赘述。

示例性地，请参阅图9，图9为本申请实施例提供的一种基于第三计算图得到第四计算图的示意图。如图9所示，第三计算图中包括卷积算子1，该卷积算子1的计算公式为y=x*w+b；其中，y为输出值，x为输入值，w和b为权重参数。基于第三计算图所得到的第四计算图中，包括卷积算子2和第三表达式。其中，卷积算子2是对卷积算子1中的权重参数w进行修改后得到的，卷积算子2的计算公式为y=x*w'+b；其中， w'=w+r，或者w'=w+S（r），r为第一数值，S（r）为经过计算表达式对r进行处理后的值。此外，第三表达式的输入包括卷积算子2的输出以及第一数值或第一数值经过计算表达式后的输出。当卷积算子1与卷积算子2的输入相同的情况下，第三表达式的输出与卷积算子1的输出相同。也就是说，假设第三表达式为D，对于任意输入x，均满足以下的等式：D（r，x*w'+b）= x*w+b。

由上述的示例可以看出，在输入第三表达式的第一数值不正确的情况下，第三表达式无法将第二算子的输出转换为第一算子的输出，因此通过对第一数值进行加密保护，则能够实现对第四计算图中的计算逻辑进行保护，从而实现对整个AI模型正常执行流程的保护，并降低了模型保护所带来的额外性能开销。

以上介绍了本申请实施例提供的一种模型的处理方法，以下将介绍在得到混淆态的模型之后，基于模型对数据进行处理的方法。

请参阅图10，图10为本申请实施例提供的一种基于模型的数据处理方法的流程示意图。如图10所示，该基于模型的数据处理方法包括以下的步骤1001-1003。

步骤1001，基于AI模型的执行代码，获取第二计算图，第二计算图包括至少一个计算分支、至少

一个混淆分支和第一表达式，第一表达式的输出用于指示执行至少一个计算分支或至少一个混淆分支中的分支，至少一个计算分支包括至少一个 AI 模型的算子，至少一个计算分支与至少一个混淆分支具有不同的计算逻辑，执行代码中与第一表达式相关的目标代码被配置为加密保护。

本实施例中，AI 模型的执行代码是基于上述图 4 对应的实施例中所述的模型的处理方法得到的。因此，终端设备通过解析 AI 模型的执行代码，能够得到第二计算图。其中，本实施例中所述的第二计算图与图 4 对应的实施例中所述的第二计算图类似，具体请参考上述图 4 对应的实施例，在此不再赘述。

步骤 1002，获取 AI 模型的输入数据。

其中，AI 模型的输入数据即为待处理的数据。例如，假设 AI 模型为图像处理模型（例如图像分类模型或图像分割模型）时，AI 模型的输入数据可以为图像数据；假设 AI 模型为语音处理模型（例如语音识别模型）时，AI 模型的输入数据可以为语音数据。总之，AI 模型的输入数据可以是根据 AI 模型的实际类型来确定，本实施例并不限定 AI 模型的输入数据的类型。

步骤 1003，基于第二计算图对输入数据进行处理，得到 AI 模型的输出数据。

在得到第二计算图的情况下，终端设备可以是基于第二计算图对输入数据进行处理，从而得到 AI 模型的输出数据。在第二计算图为 AI 模型中的部分计算图的情况下，终端设备还可以是基于第二计算图以及其他的计算图来对输入数据进行处理，得到 AI 模型的输出数据。

在一种可能的实现方式中，第二计算图仅包括一个计算分支；第一表达式的输出用于指示执行计算分支以及至少一个混淆分支中的一个分支。

在一种可能的实现方式中，第二计算图包括多个计算分支，且第二计算图还包括第二表达式；第一表达式的输出用于指示执行多个计算分支和至少一个混淆分支中的一个分支；第二表达式与多个计算分支和至少一个混淆分支连接，第二表达式用于指示第一表达式的循环执行次数，且第一表达式的输入与上一次执行的分支相关。

在一种可能的实现方式中，第二表达式的输入与上一次执行的分支相关，第二表达式的输出用于指示是否循环执行第一表达式。

在一种可能的实现方式中，第一表达式的初始输入包括第一数值，第一数值被配置为加密保护。

在一种可能的实现方式中，目标代码被配置为运行于可信执行环境中；基于 AI 模型的执行代码，获取第二计算图，包括：终端设备在可信执行环境中运行目标代码，以获取第二计算图中的第一表达式。

在一种可能的实现方式中，目标代码被配置为采用加密算法加密保护。终端设备可以对执行代码中的目标代码进行解密，得到解密后的代码；并且，终端设备执行解密后的代码，得到第一表达式。

在一种可能的实现方式中，该方法还包括：基于 AI 模型的执行代码，获取第四计算图，第四计算图包括第二算子和第三表达式，第二算子是对 AI 模型中第一算子的权重参数修改后得到的，第三表达式的输入包括第二算子的输出，且第三表达式的输出与第一算子在采用与第二算子相同的输入时的输出相同；基于第二计算图对输入数据进行处理，包括：基于第二计算图和第四计算图对输入数据进行处理。

在一种可能的实现方式中，第二算子的权重参数是基于第二数值对第一算子的权重参数修改后得到的，第三表达式的输入包括第二数值，且第二数值被配置为加密保护。

为了便于理解，以下将结合具体例子对本申请实施例提供的模型的处理方法以及基于模型的数据处理方法进行介绍。

请参阅图 11，图 11 为本申请实施例提供的一种处理模型以及基于模型处理数据的流程示意图。

在模型的混淆阶段，通过对原模型文件进行解析，获取到原模型文件对应的计算图。然后，通过本申请实施例所提供的模型的处理方法对获取得到的计算图进行控制流结构混淆、加密数据结构混淆和/或基于加密数据的模型权重混淆，得到混淆态模型文件。

其中，控制流结构混淆是指通过添加混淆分支的方式，模糊计算图中显式算子执行顺序和依赖关系，达到隐藏模型真实计算逻辑的效果。其中，添加混淆分支的方式如上述图 5A 至图 7B 对应实施例所述，服务器生成一个与待保护计算子图对应的虚假子图，并结合用户自定义的不透明谓词表达式（即上述的第一表达式），建立一个 switch 算子结构，switch 算子结构的一个分支是待保护计算子图，另一个分支

是虚假子图或表达式，以此模糊算子之间的依赖关系。又或者，对待保护计算子图生成一定数量的虚假子图和用于判断分支选择的计算表达式，使用 switch 算子，在每个 switch 的分支中分别插入待保护计算子图和虚假子图，通过计算表达式和用户自定义的不透明谓词控制分支的执行顺序，从而隐藏模型算子的执行顺序。

加密数据结构混淆是指基于加密的数据对模型结构进行混淆，具体如图 8 对应的实施例所述。具体地，服务器对待保护计算子图生成对应的虚假子图和计算表达式。通过引入外部生成的随机数，将随机数当作输入条件传入计算表达式中，从而判断控制流结构的执行分支。在推理时只有传入正确的随机数，混淆后的模型才能输出正确的结果，从而防止模型被盗用。

基于加密数据的模型权重混淆是指对于每个需要保护的权重，引用外部生成的随机噪声对权重进行加扰，且每个权重加扰时可以使用不同的随机噪声。为保证权重加扰不会影响模型执行结果的正确性，需在权重加扰过的算子后加入新建子图（即上述实施例所述的第三表达式），将随机噪声时和权重加扰过的算子的结果传入该新建子图。新建子图可以复原被保护算子未加扰的输出，从而满足权重加扰不影响模型执行的结果准确性。

在模型的混淆阶段，由于在模型混淆的过程中使用到了随机数，因此还可以对随机数进行加密，生成元数据文件。

模型混淆完毕后，将混淆态模型文件以及相应的加密后的元数据文件部署到终端设备上。

在模型推理阶段，终端设备加载并解析混淆态模型文件，以获取 AI 模型的计算图。然后，终端设备对元数据文件进行解密来获取随机数集合，解密过程可以在可信执行环境中执行。在解密得到随机数集合后，终端设备根据 AI 应用程序输入的推理数据以及解密后的随机数，遍历执行任务序列中的每个计算单元，直接进行混淆态模型推理，得到推理结果。

以上介绍了本申请实施例提供的方法，为便于理解，以下将介绍用于执行上述实施例所述的方法的装置。

请参阅图 12，图 12 为本申请实施例提供的一种模型的处理装置的结构示意图。如图 12 所示，该模型的处理装置包括：获取模块 1201，用于获取第一计算图，所述第一计算图用于指示 AI 模型的执行逻辑，所述第一计算图包括至少一个算子；处理模块 1202，用于基于所述第一计算图，生成第二计算图，所述第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，所述第一表达式的输出用于指示执行所述至少一个计算分支中的分支或所述至少一个混淆分支中的分支，所述至少一个算子包含于所述至少一个计算分支中，所述至少一个计算分支与所述至少一个混淆分支具有不同的计算逻辑；处理模块 1202，用于根据所述第二计算图生成所述 AI 模型对应的执行代码，所述执行代码中与所述第一表达式相关的目标代码被配置为加密保护。

在一种可能的实现方式中，所述第二计算图仅包括一个计算分支；所述第一表达式的输出用于指示执行所述计算分支以及所述至少一个混淆分支中的一个分支。

在一种可能的实现方式中，所述第二计算图包括多个计算分支，且所述第二计算图还包括第二表达式；所述第一表达式的输出用于指示执行所述多个计算分支和所述至少一个混淆分支中的一个分支；所述第二表达式与所述多个计算分支和所述至少一个混淆分支连接，所述第二表达式用于指示所述第一表达式的循环执行次数，且所述第一表达式的输入与上一次执行的分支相关。

在一种可能的实现方式中，所述第二表达式的输入与上一次执行的分支相关，所述第二表达式的输出用于指示是否循环执行所述第一表达式。

在一种可能的实现方式中，所述第一表达式的初始输入包括第一数值，所述第一数值被配置为加密保护。

在一种可能的实现方式中，所述目标代码被配置为运行于可信执行环境中。

在一种可能的实现方式中，所述目标代码被配置为采用加密算法加密保护。

在一种可能的实现方式中，所述获取模块 1201，还用于获取第三计算图，所述第三计算图用于指示所述 AI 模型的执行逻辑，所述第三计算图包括第一算子；所述处理模块 1202，还用于：基于所述第三

计算图，生成第四计算图，所述第四计算图包括第二算子和第三表达式，所述第二算子是对所述第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二算子的输出，且所述第三表达式的输出与所述第一算子在采用与所述第二算子相同的输入时的输出相同； 根据所述第二计算图和所述第四计算图生成所述 AI 模型对应的执行代码。

在一种可能的实现方式中，所述第二算子的权重参数是基于第二数值对所述第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二数值，且所述第二数值被配置为加密保护。

请参阅图 13，图 13 为本申请实施例提供的一种基于模型的数据处理装置的结构示意图。如图 13 所示，该基于模型的数据处理装置包括：获取模块 1301，用于基于 AI 模型的执行代码，获取第二计算图，所述第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，所述第一表达式的输出用于指示执行所述至少一个计算分支中的分支或所述至少一个混淆分支中的分支，所述至少一个计算分支包括至少一个所述 AI 模型的算子，所述至少一个计算分支与所述至少一个混淆分支具有不同的计算逻辑，所述执行代码中与所述第一表达式相关的目标代码被配置为加密保护；所述获取模块 1301，还用于获取所述 AI 模型的输入数据；所述处理模块 1302，还用于基于所述第二计算图对所述输入数据进行处理，得到所述 AI 模型的输出数据。

在一种可能的实现方式中，所述第二计算图仅包括一个计算分支；所述第一表达式的输出用于指示执行所述计算分支以及所述至少一个混淆分支中的一个分支。

在一种可能的实现方式中，所述第二计算图包括多个计算分支，且所述第二计算图还包括第二表达式；所述第一表达式的输出用于指示执行所述多个计算分支和所述至少一个混淆分支中的一个分支；所述第二表达式与所述多个计算分支和所述至少一个混淆分支连接，所述第二表达式用于指示所述第一表达式的循环执行次数，且所述第一表达式的输入与上一次执行的分支相关。

在一种可能的实现方式中，所述第二表达式的输入与上一次执行的分支相关，所述第二表达式的输出用于指示是否循环执行所述第一表达式。

在一种可能的实现方式中，所述第一表达式的初始输入包括第一数值，所述第一数值被配置为加密保护。

在一种可能的实现方式中，所述目标代码被配置为运行于可信执行环境中，和/或所述目标代码被配置为进行代码混淆。

在一种可能的实现方式中，所述目标代码被配置为采用加密算法加密保护；所述处理模块 1302，还用于对所述执行代码中的所述目标代码进行解密，得到解密后的代码；执行所述解密后的代码，得到所述第一表达式。

在一种可能的实现方式中，所述获取模块 1301，还用于基于所述 AI 模型的执行代码，获取第四计算图，所述第四计算图包括第二算子和第三表达式，所述第二算子是对所述 AI 模型中第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二算子的输出，且所述第三表达式的输出与所述第一算子在采用与所述第二算子相同的输入时的输出相同；所述处理模块 1302，还用于基于所述第二计算图和所述第四计算图对所述输入数据进行处理。

在一种可能的实现方式中，所述第二算子的权重参数是基于第二数值对所述第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二数值，且所述第二数值被配置为加密保护。

接下来介绍本申请实施例提供的一种执行设备，请参阅图 14，图 14 为本申请实施例提供的执行设备的一种结构示意图，执行设备 1400 具体可以表现为手机、平板、笔记本电脑、智能穿戴设备、服务器等，此处不做限定。具体的，执行设备 1400 包括：接收器 1401、发射器 1402、处理器 1403 和存储器 1404 (其中执行设备 1400 中的处理器 1403 的数量可以一个或多个，图 14 中以一个处理器为例)，其中，处理器 1403 可以包括应用处理器 14031 和通信处理器 14032。在本申请的一些实施例中，接收器 1401、发射器 1402、处理器 1403 和存储器 1404 可通过总线或其它方式连接。

存储器 1404 可以包括只读存储器和随机存取存储器，并向处理器 1403 提供指令和数据。存储器 1404

的一部分还可以包括非易失性随机存取存储器（non-volatile random access memory，NVRAM）。存储器1404存储有处理器和操作指令、可执行模块或者数据结构，或者它们的子集，或者它们的扩展集，其中，操作指令可包括各种操作指令，用于实现各种操作。

处理器1403控制执行设备的操作。具体的应用中，执行设备的各个组件通过总线系统耦合在一起，其中总线系统除包括数据总线之外，还可以包括电源总线、控制总线和状态信号总线等。但是为了清楚说明起见，在图中将各种总线都称为总线系统。

上述本申请实施例揭示的方法可以应用于处理器1403中，或者由处理器1403实现。处理器1403可以是一种集成电路芯片，具有信号的处理能力。在实现过程中，上述方法的各步骤可以通过处理器1403中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器1403可以是通用处理器、数字信号处理器（digital signal processing，DSP）、微处理器或微控制器，还可进一步包括专用集成电路（application specific integrated circuit，ASIC）、现场可编程门阵列（field-programmable gate array，FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。该处理器1403可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成，或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器，闪存、只读存储器，可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器1404，处理器1403读取存储器1404中的信息，结合其硬件完成上述方法的步骤。

接收器1401可用于接收输入的数字或字符信息，以及产生与执行设备的相关设置以及功能控制有关的信号输入。发射器1402可用于通过第一接口输出数字或字符信息；发射器1402还可用于通过第一接口向磁盘组发送指令，以修改磁盘组中的数据；发射器1402还可以包括显示屏等显示设备。

本申请实施例中，在一种情况下，处理器1403，用于执行图4或图10对应实施例中的方法。

本申请实施例提供的电子设备具体可以为芯片，芯片包括：处理单元和通信单元，所述处理单元例如可以是处理器，所述通信单元例如可以是输入/输出接口、管脚或电路等。该处理单元可执行存储单元存储的计算机执行指令，以使执行设备内的芯片执行上述实施例描述的模型超参数的选择方法，或者，以使训练设备内的芯片执行上述实施例描述的模型超参数的选择方法。可选地，所述存储单元为所述芯片内的存储单元，如寄存器、缓存等，所述存储单元还可以是所述无线接入设备端内的位于所述芯片外部的存储单元，如只读存储器（read-only memory，ROM)或可存储静态信息和指令的其他类型的静态存储设备，随机存取存储器（random access memory，RAM)等。

具体的，请参阅图15，图15为本申请实施例提供的芯片的一种结构示意图，所述芯片可以表现为神经网络处理器NPU 1500，NPU 1500作为协处理器挂载到主CPU（Host CPU）上，由Host CPU分配任务。NPU的核心部分为运算电路1503，通过控制器1504控制运算电路1503提取存储器中的矩阵数据并进行乘法运算。

在一些实现中，运算电路1503内部包括多个处理单元（Process Engine, PE）。在一些实现中，运算电路1503是二维脉动阵列。运算电路1503还可以是一维脉动阵列或者能够执行例如乘法和加法这样的数学运算的其它电子线路。在一些实现中，运算电路1503是通用的矩阵处理器。

举例来说，假设有输入矩阵A，权重矩阵B，输出矩阵C。运算电路从权重存储器1502中取矩阵B相应的数据，并缓存在运算电路中每一个PE上。运算电路从输入存储器1501中取矩阵A数据与矩阵B进行矩阵运算，得到的矩阵的部分结果或最终结果，保存在累加器（accumulator）1508中。

统一存储器1506用于存放输入数据以及输出数据。权重数据直接通过存储单元访问控制器（Direct Memory Access Controller，DMAC）1505，DMAC被搬运到权重存储器1502中。输入数据也通过DMAC被搬运到统一存储器1506中。

BIU为Bus Interface Unit 即，总线接口单元1515，用于AXI总线与DMAC和取指存储器（Instruction Fetch Buffer，IFB）1509的交互。

总线接口单元1515（Bus Interface Unit，简称BIU），用于取指存储器1509从外部存储器获取指令，

还用于存储单元访问控制器 1505 从外部存储器获取输入矩阵 A 或者权重矩阵 B 的原数据。

DMAC 主要用于将外部存储器 DDR 中的输入数据搬运到统一存储器 1506 或将权重数据搬运到权重存储器 1502 中或将输入数据数据搬运到输入存储器 1501 中。

向量计算单元 1507 包括多个运算处理单元，在需要的情况下，对运算电路 1503 的输出做进一步处理，如向量乘，向量加，指数运算，对数运算，大小比较等等。主要用于神经网络中非卷积/全连接层网络计算，如 Batch Normalization(批归一化)，像素级求和，对特征平面进行上采样等。

在一些实现中，向量计算单元 1507 能将经处理的输出的向量存储到统一存储器 1506。例如，向量计算单元 1507 可以将线性函数；或，非线性函数应用到运算电路 1503 的输出，例如对卷积层提取的特征平面进行线性插值，再例如累加值的向量，用以生成激活值。在一些实现中，向量计算单元 1507 生成归一化的值、像素级求和的值，或二者均有。在一些实现中，处理过的输出的向量能够用作到运算电路 1503 的激活输入，例如用于在神经网络中的后续层中的使用。

控制器 1504 连接的取指存储器（instruction fetch buffer）1509，用于存储控制器 1504 使用的指令；

统一存储器 1506，输入存储器 1501，权重存储器 1502 以及取指存储器 1509 均为 On-Chip 存储器。外部存储器私有于该 NPU 硬件架构。

其中，上述任一处提到的处理器，可以是一个通用中央处理器，微处理器，ASIC，或一个或多个用于控制上述程序执行的集成电路。

可以参阅图 16，图 16 为本申请实施例提供的一种计算机可读存储介质的结构示意图。本申请还提供了一种计算机可读存储介质，在一些实施例中，上述图 4 或图 10 所公开的方法可以实施为以机器可读格式被编码在计算机可读存储介质上或者被编码在其它非瞬时性介质或者制品上的计算机程序指令。

图 16 示意性地示出根据这里展示的至少一些实施例而布置的示例计算机可读存储介质的概念性局部视图，示例计算机可读存储介质包括用于在计算设备上执行计算机进程的计算机程序。

在一个实施例中，计算机可读存储介质 1600 是使用信号承载介质 1601 来提供的。信号承载介质 1601 可以包括一个或多个程序指令 1602，其当被一个或多个处理器运行时可以提供以上针对图 4 或图 10 描述的功能或者部分功能。此外，图 16 中的程序指令 1602 也描述示例指令。

在一些示例中，信号承载介质 1601 可以包含计算机可读介质 1603,诸如但不限于，硬盘驱动器、紧密盘(CD)、数字视频光盘(DVD)、数字磁带、存储器、ROM 或 RAM 等等。

在一些实施方式中，信号承载介质 1601 可以包含计算机可记录介质 1604，诸如但不限于，存储器、读/写(R/W)CD、R/W DVD、等等。在一些实施方式中，信号承载介质 1601 可以包含通信介质 1605,诸如但不限于，数字和/或模拟通信介质(例如，光纤电缆、波导、有线通信链路、无线通信链路、等等)。因此，例如，信号承载介质 1601 可以由无线形式的通信介质 1605(例如，遵守 IEEE 802.16 标准或者其它传输协议的无线通信介质)来传达。

一个或多个程序指令 1602 可以是，例如，计算机可执行指令或者逻辑实施指令。在 一些示例中，计算设备的计算设备可以被配置为，响应于通过计算机可读介质 1603、计算机可记录介质 1604、和/或通信介质 1605 中的一个或多个传达到计算设备的程序指令 1602，提供各种操作、功能、或者动作。

另外需说明的是，以上所描述的装置实施例仅仅是示意性的，其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。另外，本申请提供的装置实施例附图中，模块之间的连接关系表示它们之间具有通信连接，具体可以实现为一条或多条通信总线或信号线。

通过以上的实施方式的描述，所属领域的技术人员可以清楚地了解到本申请可借助软件加必需的通用硬件的方式来实现，当然也可以通过专用硬件包括专用集成电路、专用 CPU、专用存储器、专用元器件等来实现。一般情况下，凡由计算机程序完成的功能都可以很容易地用相应的硬件来实现，而且，用来实现同一功能的具体硬件结构也可以是多种多样的，例如模拟电路、数字电路或专用电路等。但是，对本申请而言更多情况下软件程序实现是更佳的实施方式。基于这样的理解，本申请的技术方案本质上

或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在可读取的存储介质中，如计算机的软盘、U 盘、移动硬盘、ROM、RAM、磁碟或者光盘等，包括若干指令用以使得一台计算机设备（可以是个人计算机，训练设备，或者网络设备等）执行本申请各个实施例所述的方法。

在上述实施例中，可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时，可以全部或部分地以计算机程序产品的形式实现。

所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机程序指令时，全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一计算机可读存储介质传输，例如，所述计算机指令可以从一个网站站点、计算机、训练设备或数据中心通过有线（例如同轴电缆、光纤、数字用户线（DSL））或无线（例如红外、无线、微波等）方式向另一个网站站点、计算机、训练设备或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存储的任何可用介质或者是包含一个或多个可用介质集成的训练设备、数据中心等数据存储设备。所述可用介质可以是磁性介质，（例如，软盘、硬盘、磁带）、光介质（例如，DVD）、或者半导体介质（例如固态硬盘（Solid State Disk，SSD)）等。

权 利 要 求

1.一种模型的处理方法，其特征在于，包括：

获取第一计算图，所述第一计算图用于指示人工智能 AI 模型的执行逻辑，所述第一计算图包括至少一个算子；

基于所述第一计算图，生成第二计算图，所述第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，所述第一表达式的输出用于指示执行所述至少一个计算分支中的分支或所述至少一个混淆分支中的分支，所述至少一个算子包含于所述至少一个计算分支中，所述至少一个计算分支与所述至少一个混淆分支具有不同的计算逻辑；

根据所述第二计算图生成所述 AI 模型对应的执行代码，所述执行代码中的目标代码被配置为加密保护，所述目标代码是与所述第一表达式相关的代码。

2.根据权利要求 1 所述的方法，其特征在于，所述第二计算图仅包括一个计算分支；

所述第一表达式的输出用于指示执行所述计算分支以及所述至少一个混淆分支中的一个分支。

3.根据权利要求 1 所述的方法，其特征在于，所述第二计算图包括多个计算分支，且所述第二计算图还包括第二表达式；

所述第一表达式的输出用于指示执行所述多个计算分支和所述至少一个混淆分支中的一个分支；

所述第二表达式与所述多个计算分支和所述至少一个混淆分支连接，所述第二表达式用于指示所述第一表达式的循环执行次数。

4.根据权利要求 3 所述的方法，其特征在于，所述第二表达式的输入与上一次执行的分支相关，所述第二表达式的输出用于确定是否循环执行所述第一表达式。

5.根据权利要求 1-4 任意一项所述的方法，其特征在于，所述第一表达式的初始输入包括第一数值，所述第一数值被配置为加密保护。

6.根据权利要求 1-5 任意一项所述的方法，其特征在于，所述目标代码被配置为运行于可信执行环境中，和/或所述目标代码被配置为进行代码混淆。

7.根据权利要求 1-6 任意一项所述的方法，其特征在于，所述目标代码被配置为采用加密算法加密保护。

8.根据权利要求 1-7 任意一项所述的方法，其特征在于，所述方法还包括：

获取第三计算图，所述第三计算图用于指示所述 AI 模型的执行逻辑，所述第三计算图包括第一算子；

基于所述第三计算图，生成第四计算图，所述第四计算图包括第二算子和第三表达式，所述第二算子是对所述第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二算子的输出，且所述第三表达式的输出与所述第一算子在采用与所述第二算子相同的输入时的输出相同；

所述根据所述第二计算图生成所述 AI 模型对应的执行代码，包括：

根据所述第二计算图和所述第四计算图生成所述 AI 模型对应的执行代码。

9.根据权利要求 8 所述的方法，其特征在于，所述第二算子的权重参数是基于第二数值对所述第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二数值，且所述第二数值被配置为加密保护。

10.一种基于模型的数据处理方法，其特征在于，包括：

基于 AI 模型的执行代码，获取第二计算图，所述第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，所述第一表达式的输出用于指示执行所述至少一个计算分支中的分支或所述至少一个混淆分支中的分支，所述至少一个计算分支包括至少一个所述 AI 模型的算子，所述至少一个计算分支与所述至少一个混淆分支具有不同的计算逻辑，所述执行代码中的目标代码被配置为加密保护，所述目标代码是与所述第一表达式相关的代码；

获取所述 AI 模型的输入数据；

基于所述第二计算图对所述输入数据进行处理，得到所述 AI 模型的输出数据。

11.根据权利要求 10 所述的方法，其特征在于，所述第二计算图仅包括一个计算分支；

所述第一表达式的输出用于指示执行所述计算分支以及所述至少一个混淆分支中的一个分支。

12.根据权利要求 10 所述的方法，其特征在于，所述第二计算图包括多个计算分支，且所述第二计算图还包括第二表达式；

所述第一表达式的输出用于指示执行所述多个计算分支和所述至少一个混淆分支中的一个分支；

所述第二表达式与所述多个计算分支和所述至少一个混淆分支连接，所述第二表达式用于指示所述第一表达式的循环执行次数。

13.根据权利要求 12 所述的方法，其特征在于，所述第二表达式的输入与上一次执行的分支相关，所述第二表达式的输出用于确定是否循环执行所述第一表达式。

14.根据权利要求 10-13 任意一项所述的方法，其特征在于，所述第一表达式的初始输入包括第一数值，所述第一数值被配置为加密保护。

15.根据权利要求 10-14 任意一项所述的方法，其特征在于，所述目标代码被配置为运行于可信执行环境中，和/或所述目标代码被配置为进行代码混淆。

16.根据权利要求 10-15 任意一项所述的方法，其特征在于，所述目标代码被配置为采用加密算法加密保护；

所述方法还包括：

对所述执行代码中的所述目标代码进行解密，得到解密后的代码；

执行所述解密后的代码，得到所述第一表达式。

17.根据权利要求 10-16 任意一项所述的方法，其特征在于，所述方法还包括：

基于所述 AI 模型的执行代码，获取第四计算图，所述第四计算图包括第二算子和第三表达式，所述第二算子是对所述 AI 模型中第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二算子的输出，且所述第三表达式的输出与所述第一算子在采用与所述第二算子相同的输入时的输出相同；

所述基于所述第二计算图对所述输入数据进行处理，包括：

基于所述第二计算图和所述第四计算图对所述输入数据进行处理。

18.根据权利要求 17 所述的方法，其特征在于，所述第二算子的权重参数是基于第二数值对所述第一算子的权重参数修改后得到的，所述第三表达式的输入包括所述第二数值，且所述第二数值被配置为加密保护。

19.一种模型的处理装置，其特征在于，包括存储器和处理器；所述存储器存储有代码，所述处理器被配置为执行所述代码，当所述代码被执行时，所述装置执行如权利要求 1 至 9 任一项所述的方法。

20.一种基于模型的数据处理装置，其特征在于，包括存储器和处理器；所述存储器存储有代码，所述处理器被配置为执行所述代码，当所述代码被执行时，所述装置执行如权利要求 10 至 18 任一项所述的方法。

21.一种 AI 系统，其特征在于，包括：如权利要求 19 所述的模型的处理装置以及如权利要求 20 所述的基于模型的数据处理装置。

22.一种计算机存储介质，其特征在于，所述计算机存储介质存储有指令，所述指令在由计算机执行时使得所述计算机实施权利要求 1 至 18 任意一项所述的方法。

23.一种计算机程序产品，其特征在于，所述计算机程序产品存储有指令，所述指令在由计算机执行时使得所述计算机实施权利要求 1 至 18 任意一项所述的方法。

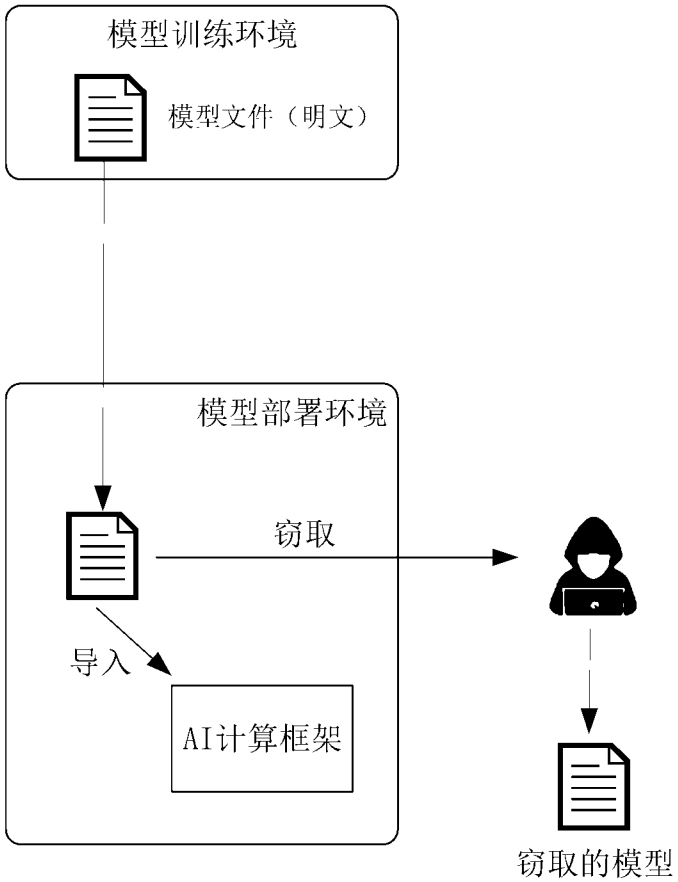


图 1

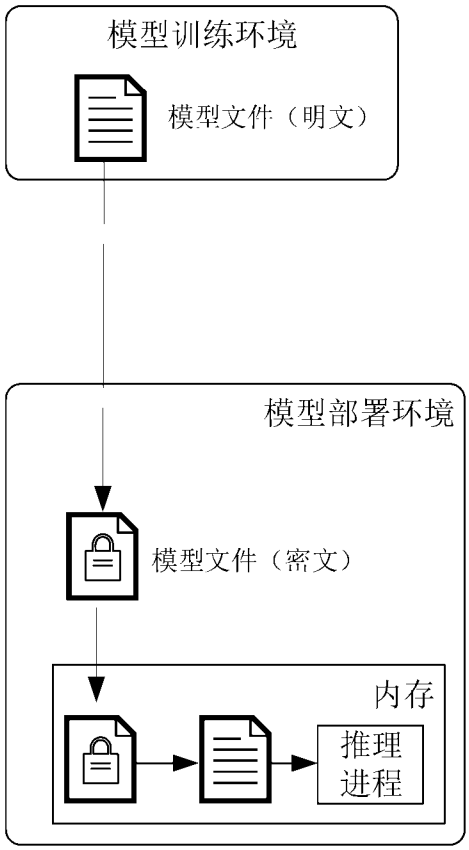


图 2

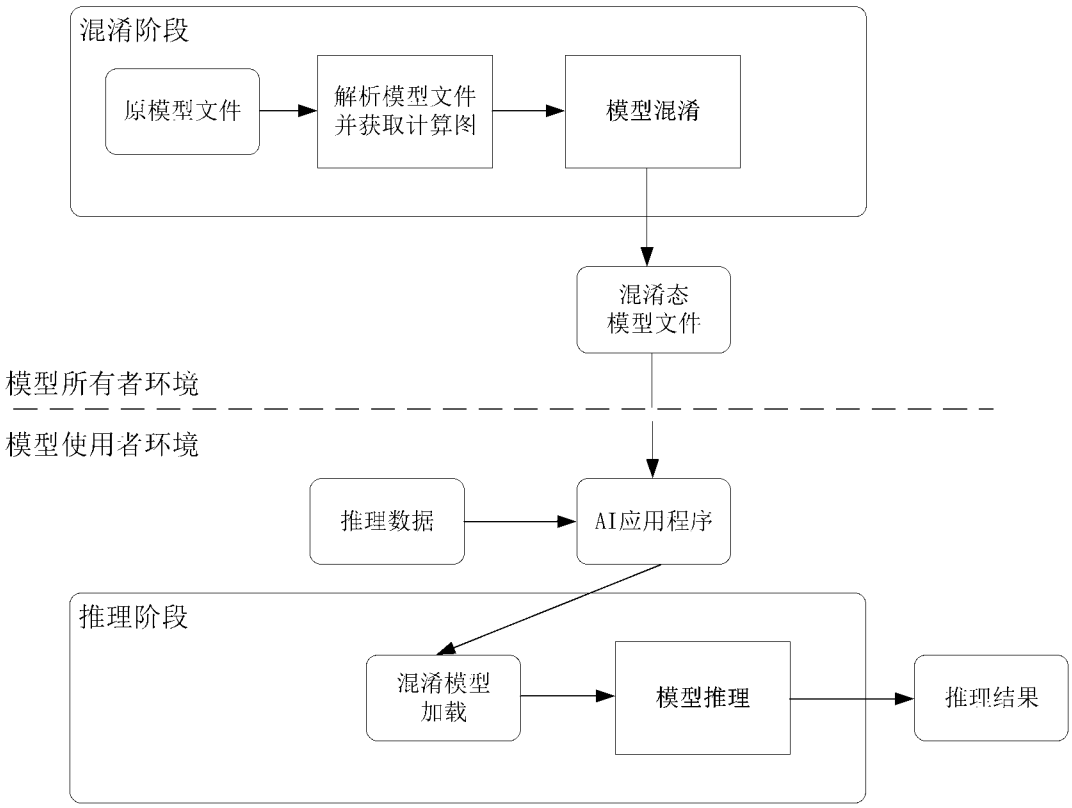

混淆阶段
原模型文件
解析模型文件
并获取计算图
模型混淆
混淆态
模型文件
模型所有者环境
模型使用者环境
推理数据
AI应用程序
推理阶段
混淆模型
加载
模型推理
推理结果

图 3

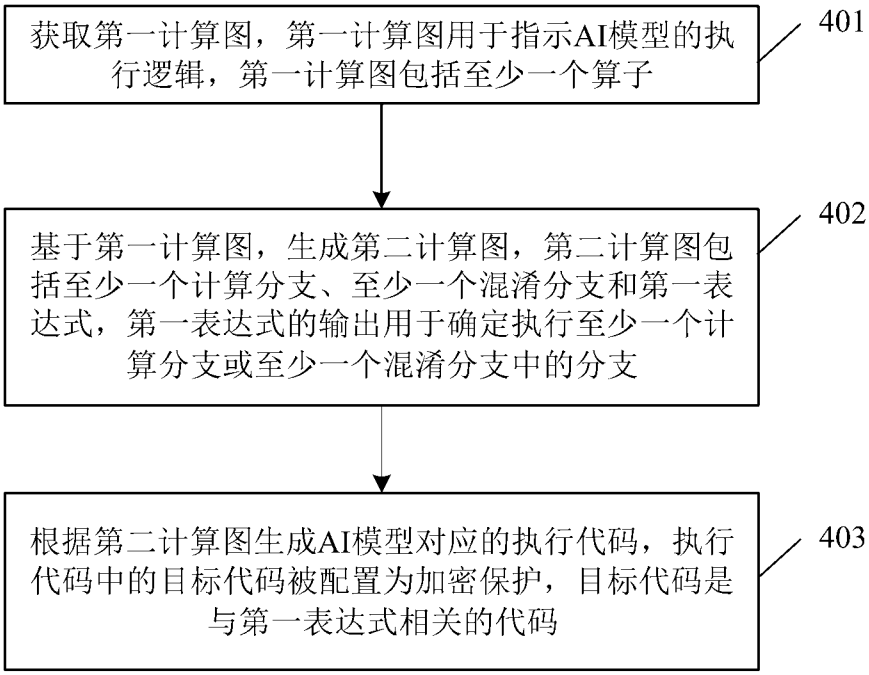

获取第一计算图，第一计算图用于指示AI模型的执行逻辑，第一计算图包括至少一个算子
401
基于第一计算图，生成第二计算图，第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，第一表达式的输出用于确定执行至少一个计算分支或至少一个混淆分支中的分支
402
根据第二计算图生成AI模型对应的执行代码，执行代码中的目标代码被配置为加密保护，目标代码是与第一表达式相关的代码
403

图 4

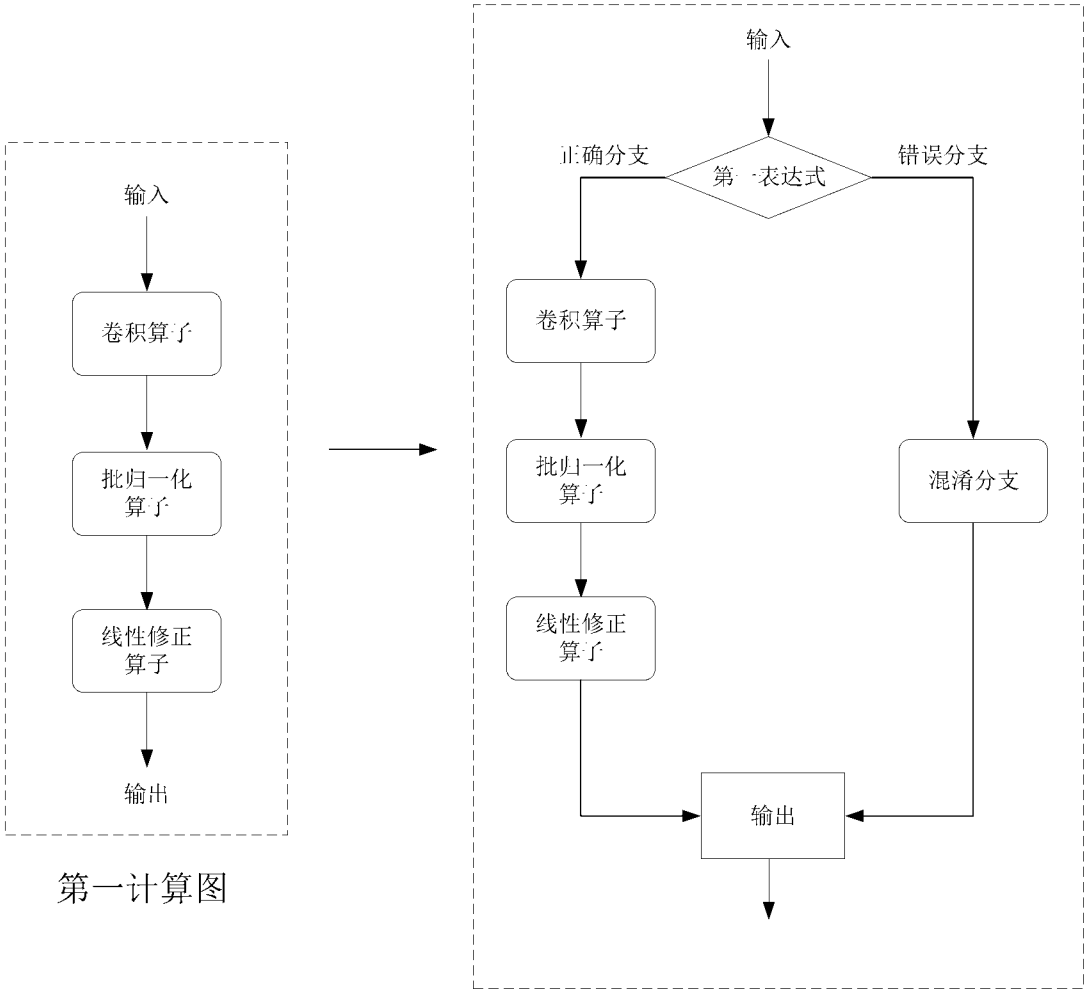

输入
卷积算子
批归一化算子
线性修正算子
输出
第一计算图
输入
正确分支
第一表达式
错误分支
卷积算子
批归一化算子
线性修正算子
混淆分支
输出
第二计算图

图 5A

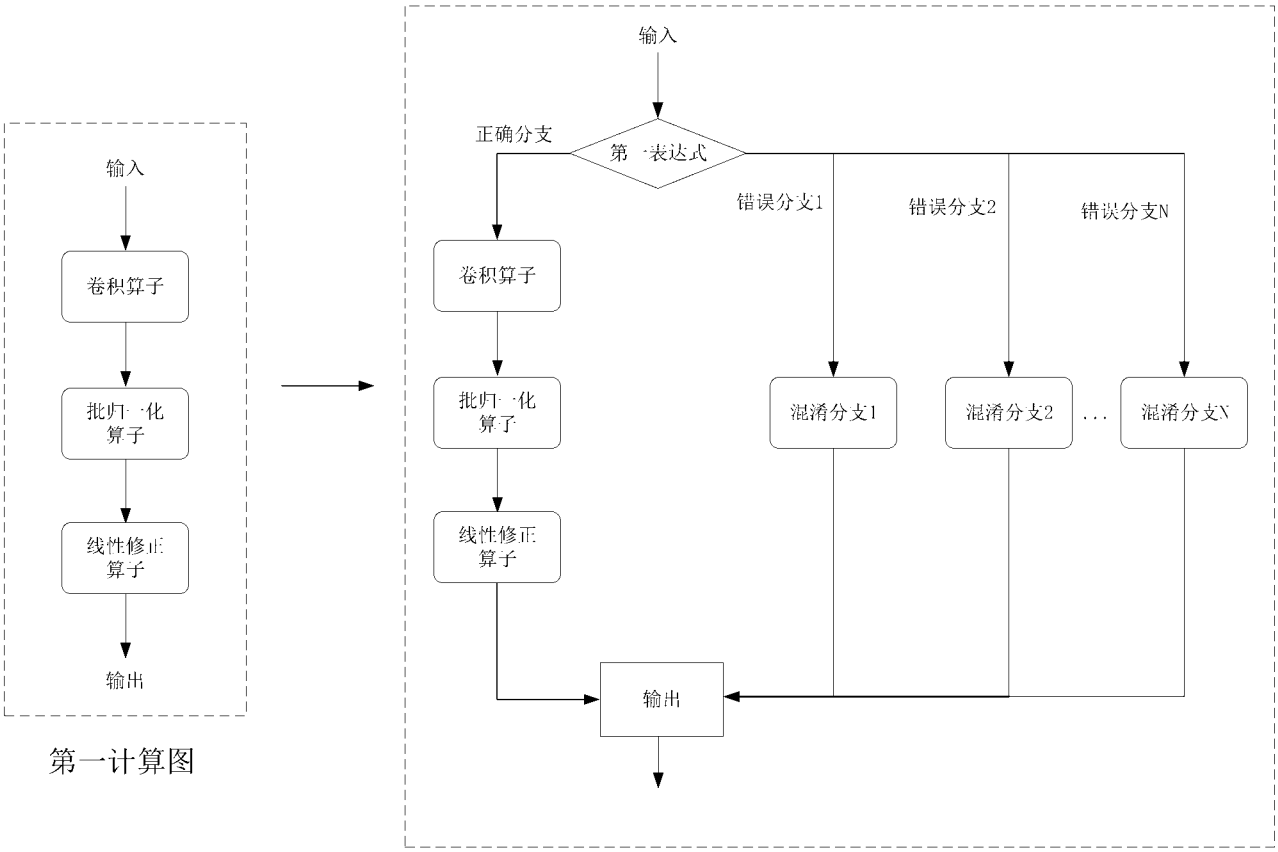

输入
卷积算子
批归一化算子
线性修正算子
输出
第一计算图
输入
正确分支
第一表达式
错误分支1
错误分支2
错误分支N
卷积算子
批归一化算子
线性修正算子
混淆分支1
混淆分支2
...
混淆分支N
输出
第二计算图

图 5B

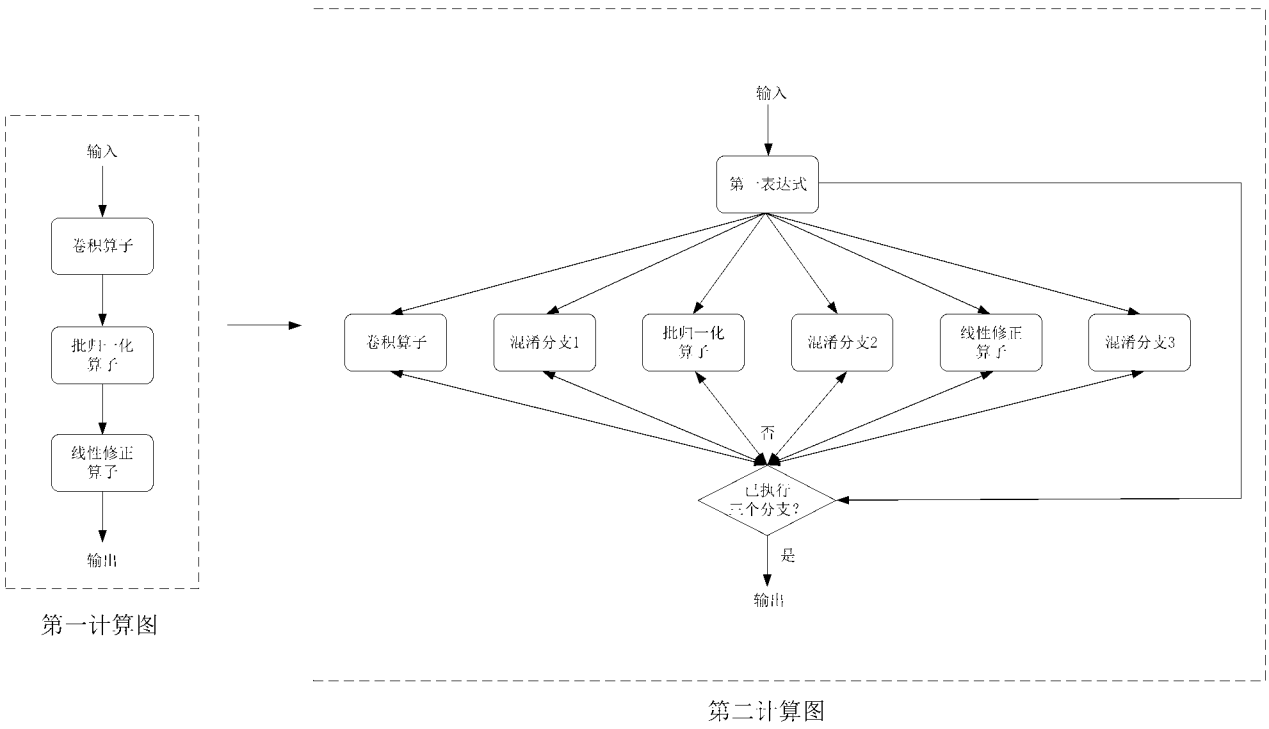
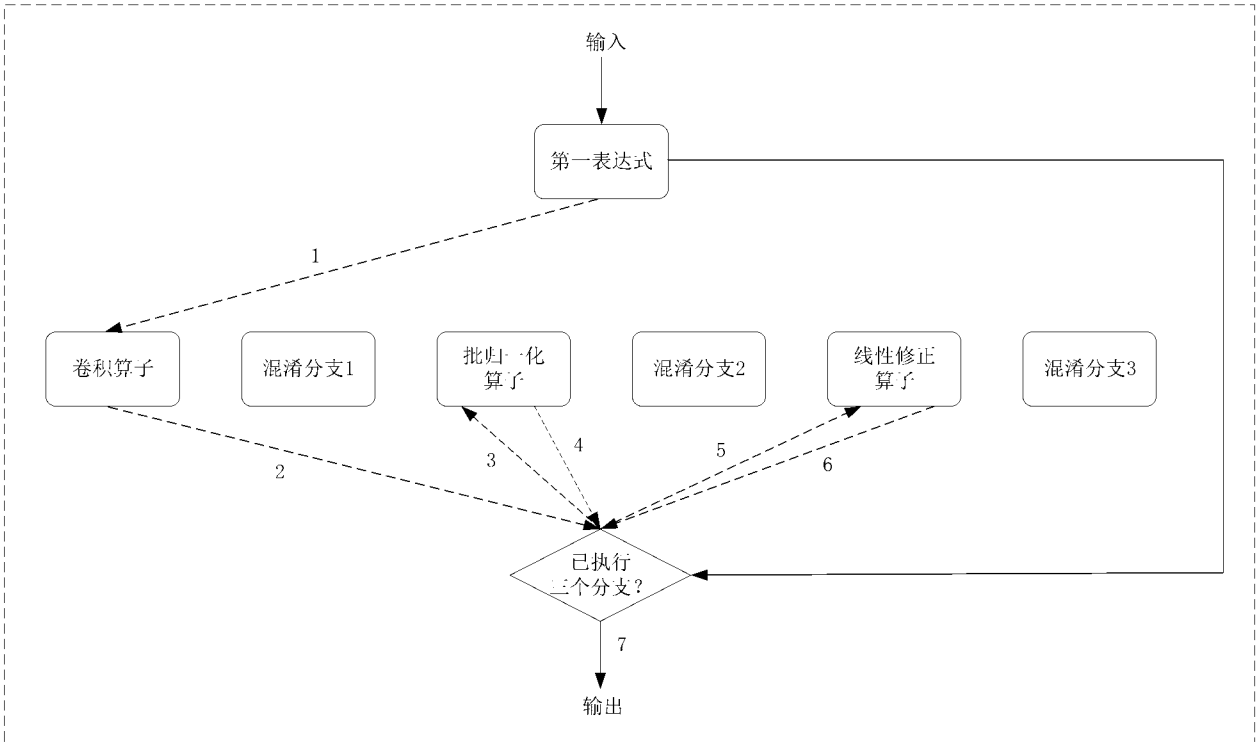
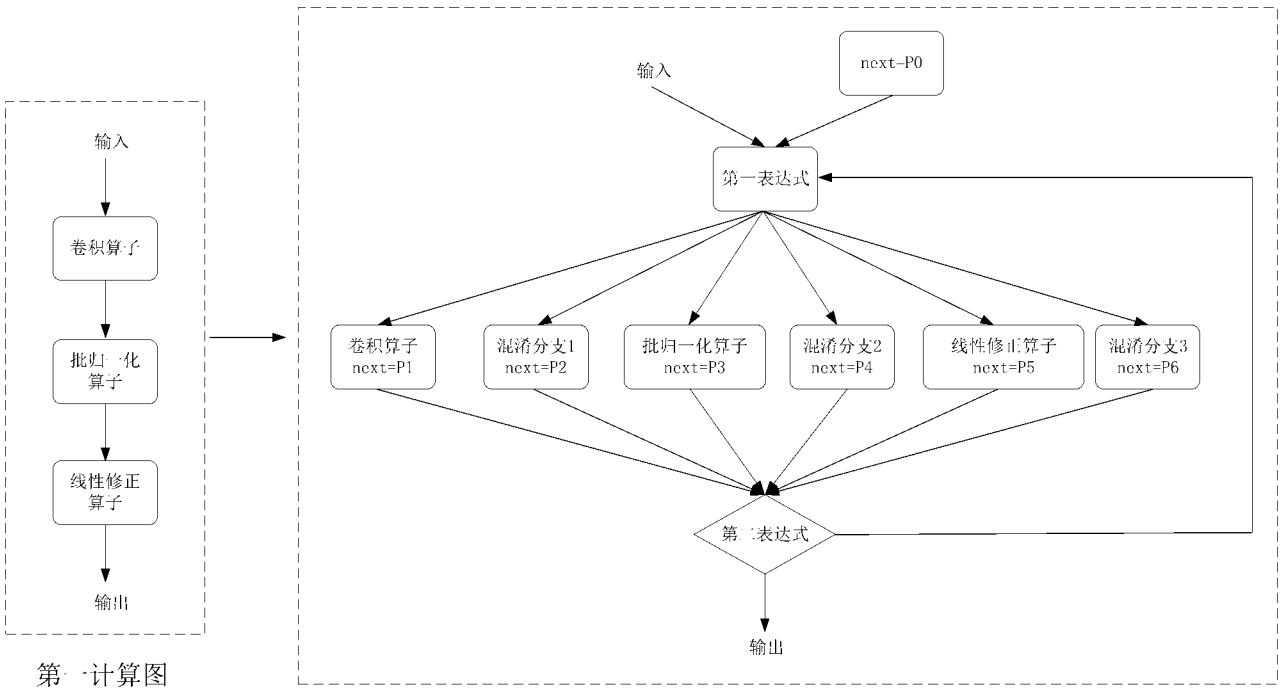

输入
卷积算子
批归一化算子
线性修正算子
输出
第一计算图
输入
第一表达式
卷积算子
混淆分支1
批归一化算子
混淆分支2
线性修正算子
混淆分支3
否
已执行三个分支?
是
输出
第二计算图

图 6A



第二计算图

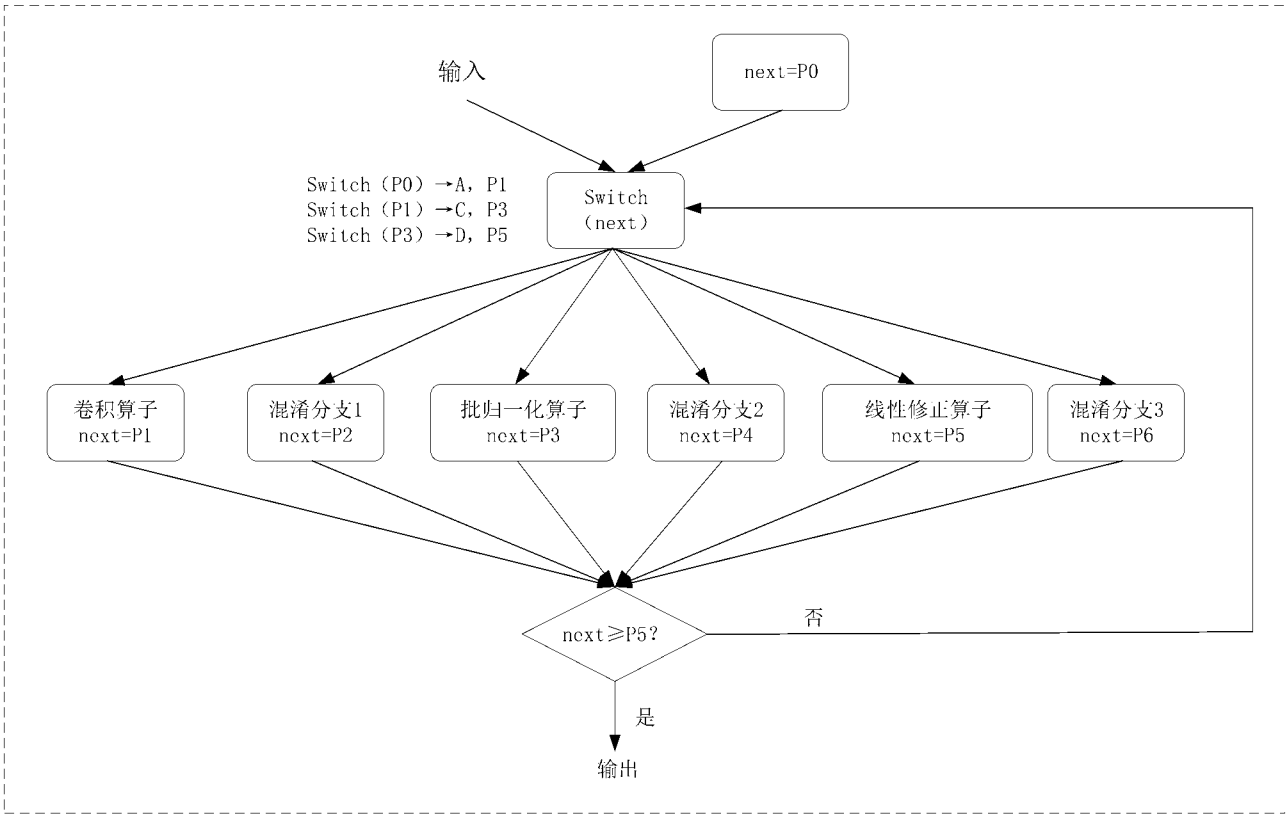
图 6B



第一计算图

第二计算图

图 7A


输入
next=P0
Switch（P0）→A，P1
Switch（P1）→C，P3
Switch（P3）→D，P5
Switch
（next）
卷积算子
next=P1
混淆分支1
next=P2
批归一化算子
next=P3
混淆分支2
next=P4
线性修正算子
next=P5
混淆分支3
next=P6
next≥P5?
否
是
输出

第二计算图

图 7B

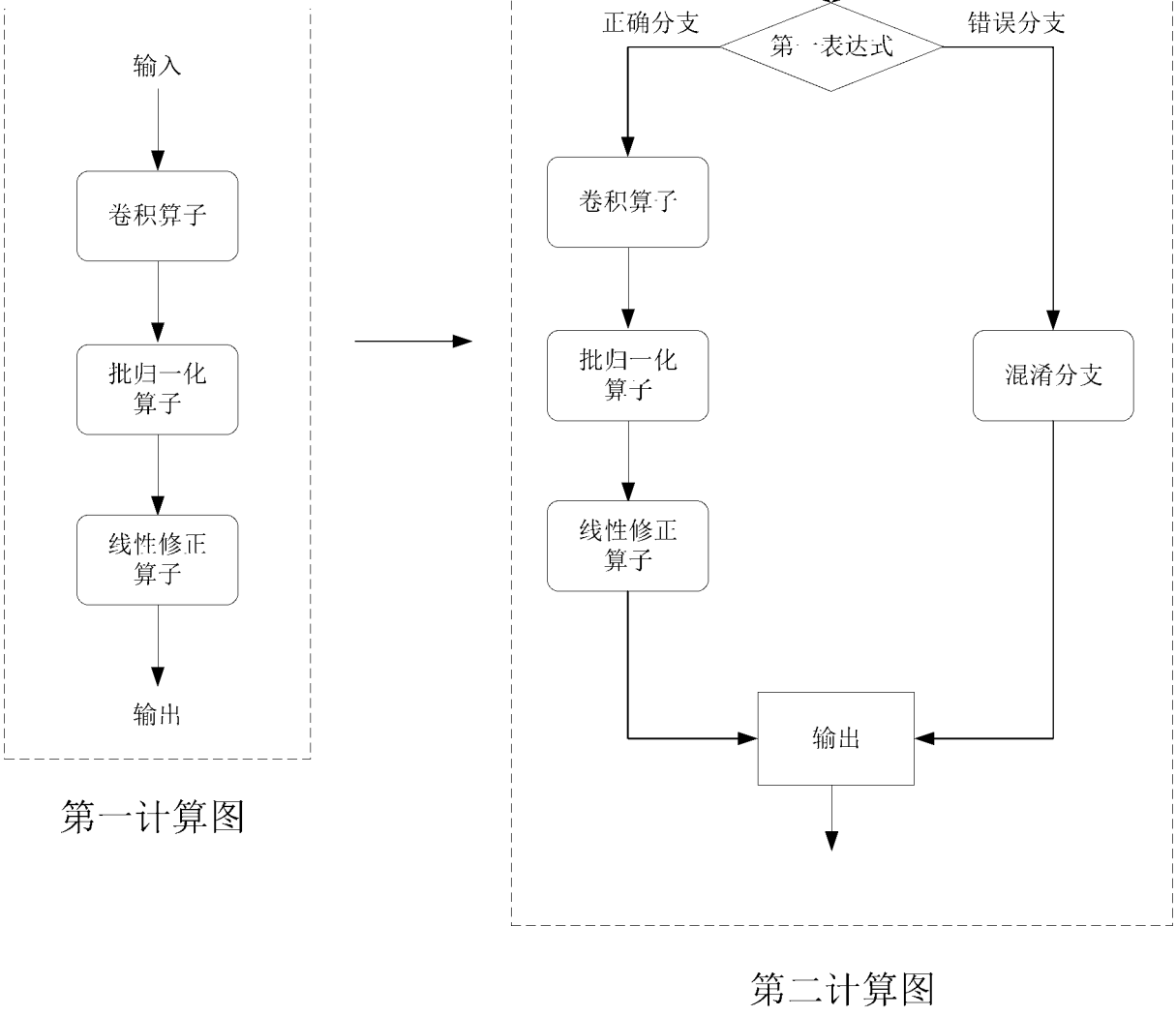

输入
卷积算子
批归一化算子
线性修正算子
输出
第一计算图
第一数值
输入
计算表达式
正确分支
第 表达式
错误分支
卷积算子
批归一化算子
线性修正算子
混淆分支
输出
第二计算图

图 8

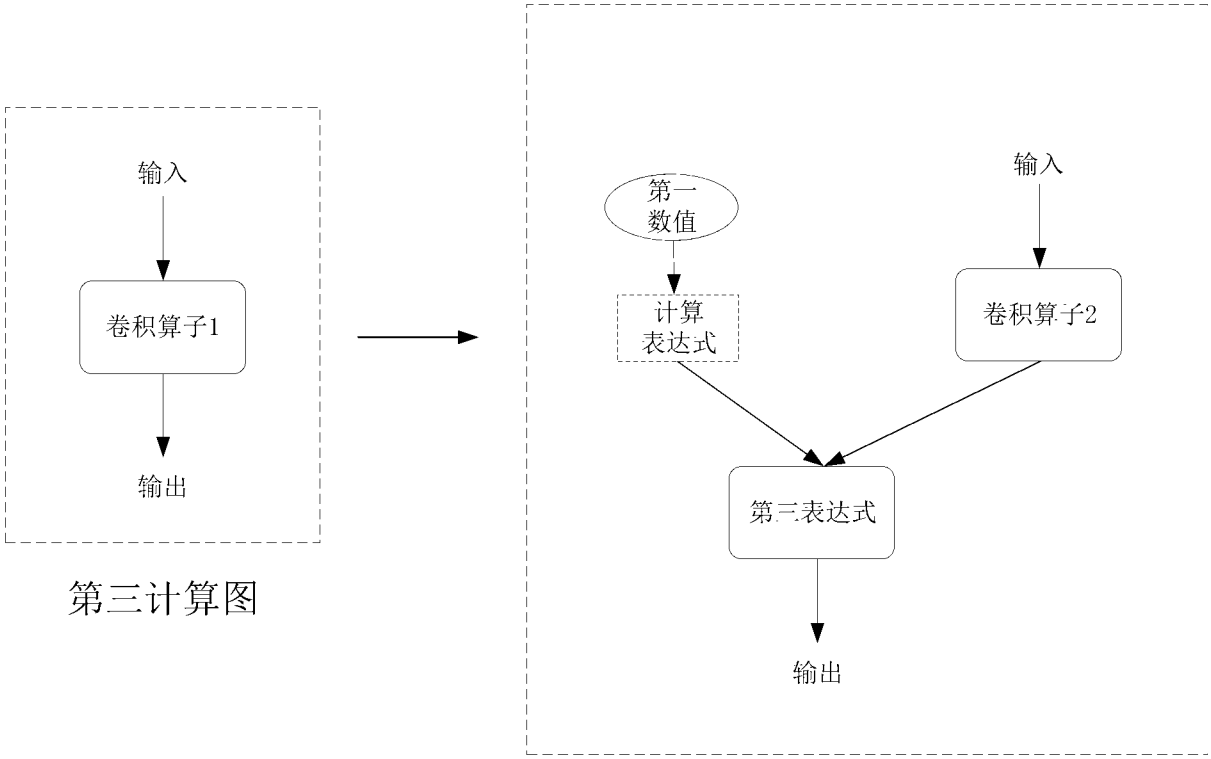

输入
卷积算子1
输出
第三计算图
第一数值
计算表达式
输入
卷积算子2
第三表达式
输出
第四计算图

图 9

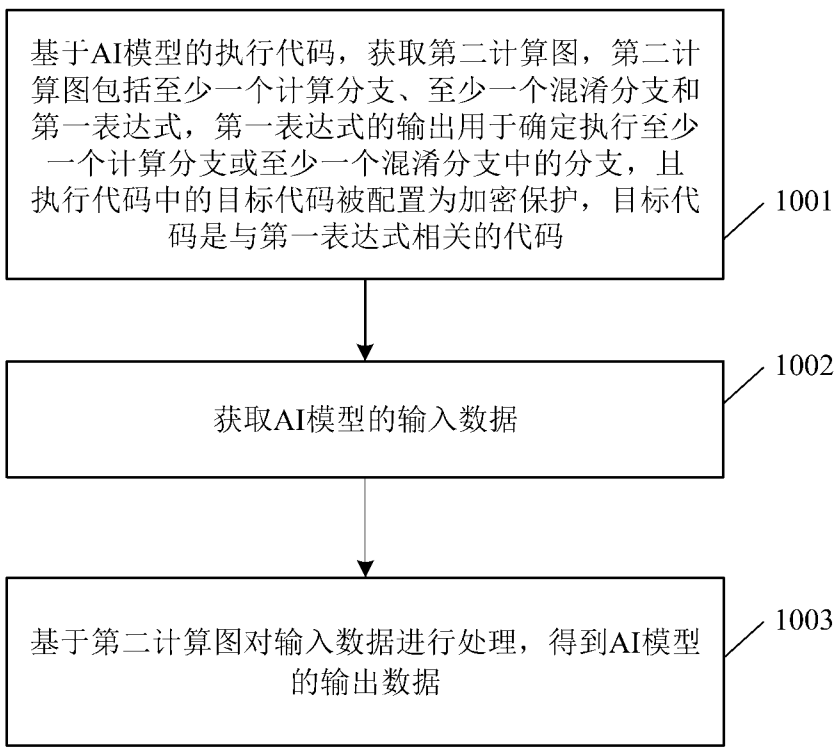

基于AI模型的执行代码，获取第二计算图，第二计算图包括至少一个计算分支、至少一个混淆分支和第一表达式，第一表达式的输出用于确定执行至少一个计算分支或至少一个混淆分支中的分支，且执行代码中的目标代码被配置为加密保护，目标代码是与第一表达式相关的代码
1001
获取AI模型的输入数据
1002
基于第二计算图对输入数据进行处理，得到AI模型的输出数据
1003

图 10

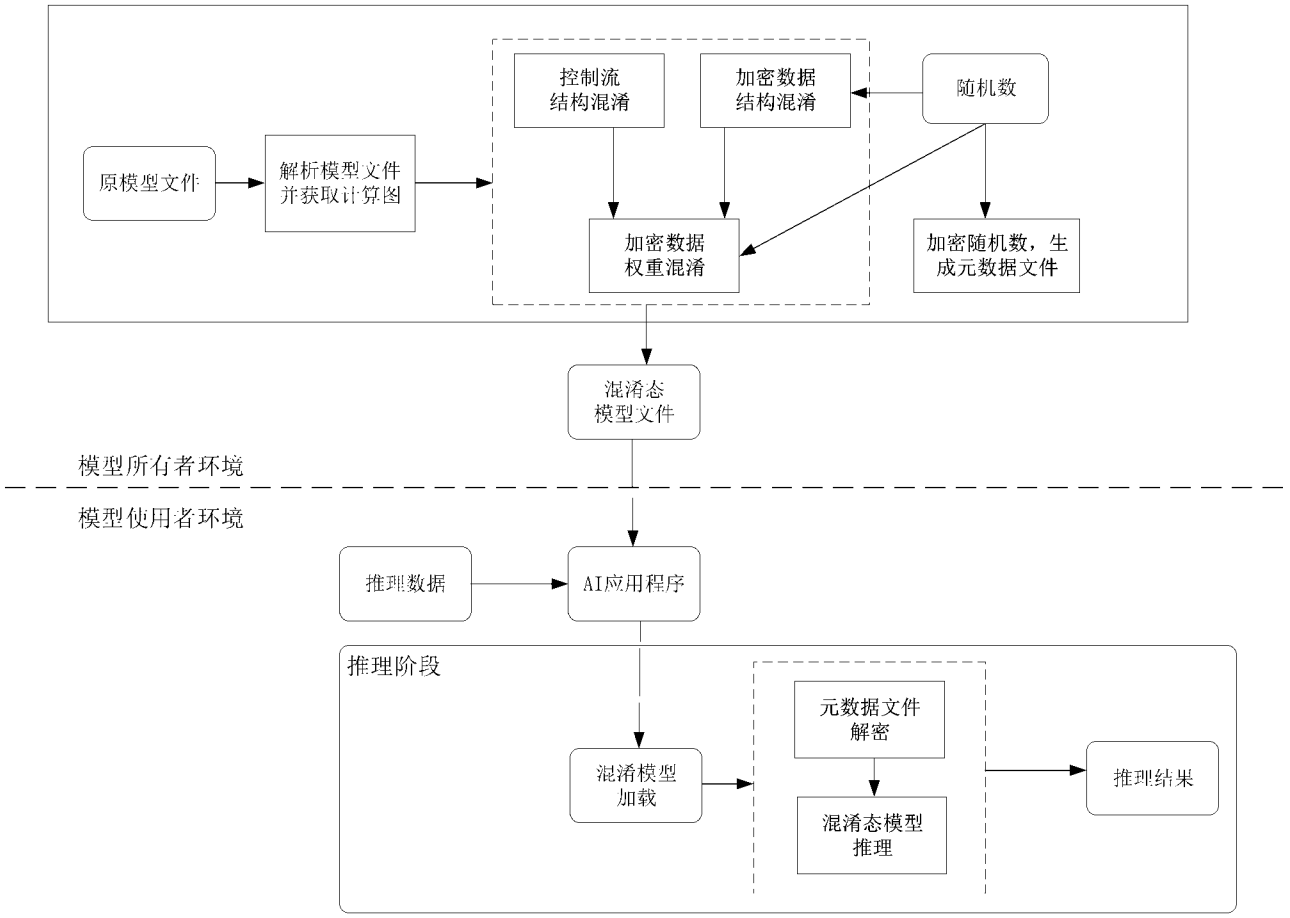

控制流
结构混淆
加密数据
结构混淆
随机数
原模型文件
解析模型文件
并获取计算图
加密数据
权重混淆
加密随机数，生
成元数据文件
混淆态
模型文件
模型所有者环境
模型使用者环境
推理数据
AI应用程序
推理阶段
元数据文件
解密
混淆模型
加载
混淆态模型
推理
推理结果

图 11

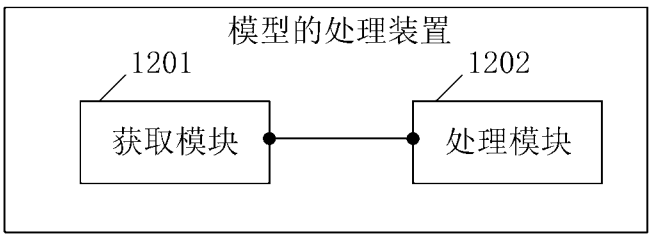

模型的处理装置
1201
1202
获取模块
处理模块

图 12

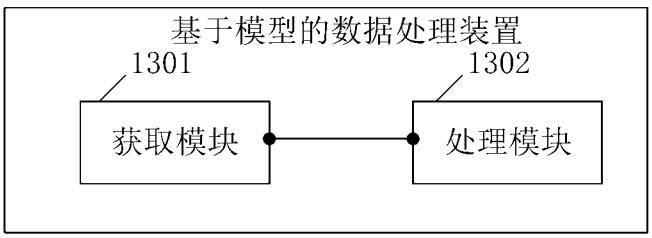

基于模型的数据处理装置
1301
1302
获取模块
处理模块

图 13

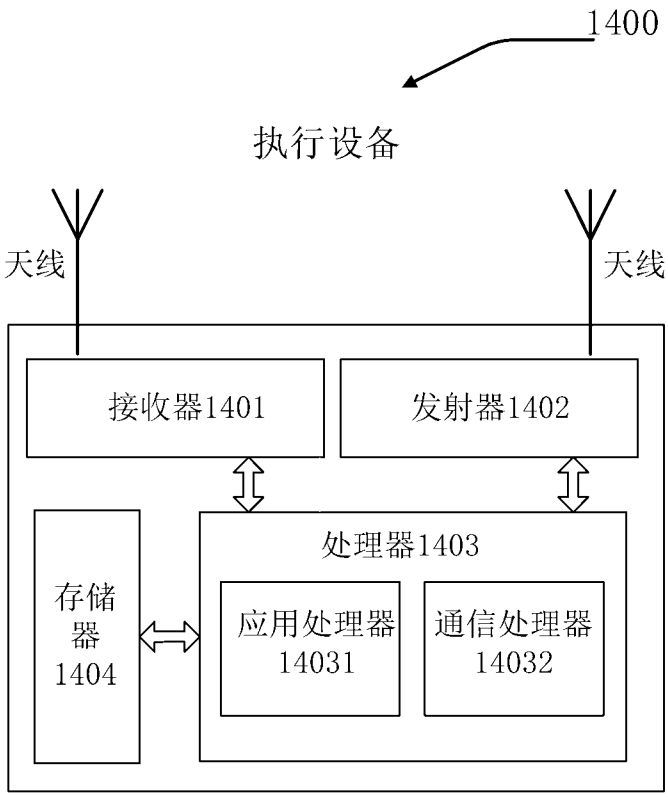

1400
执行设备
天线
天线
接收器1401
发射器1402
处理器1403
存储器1404
应用处理器14031
通信处理器14032

图 14

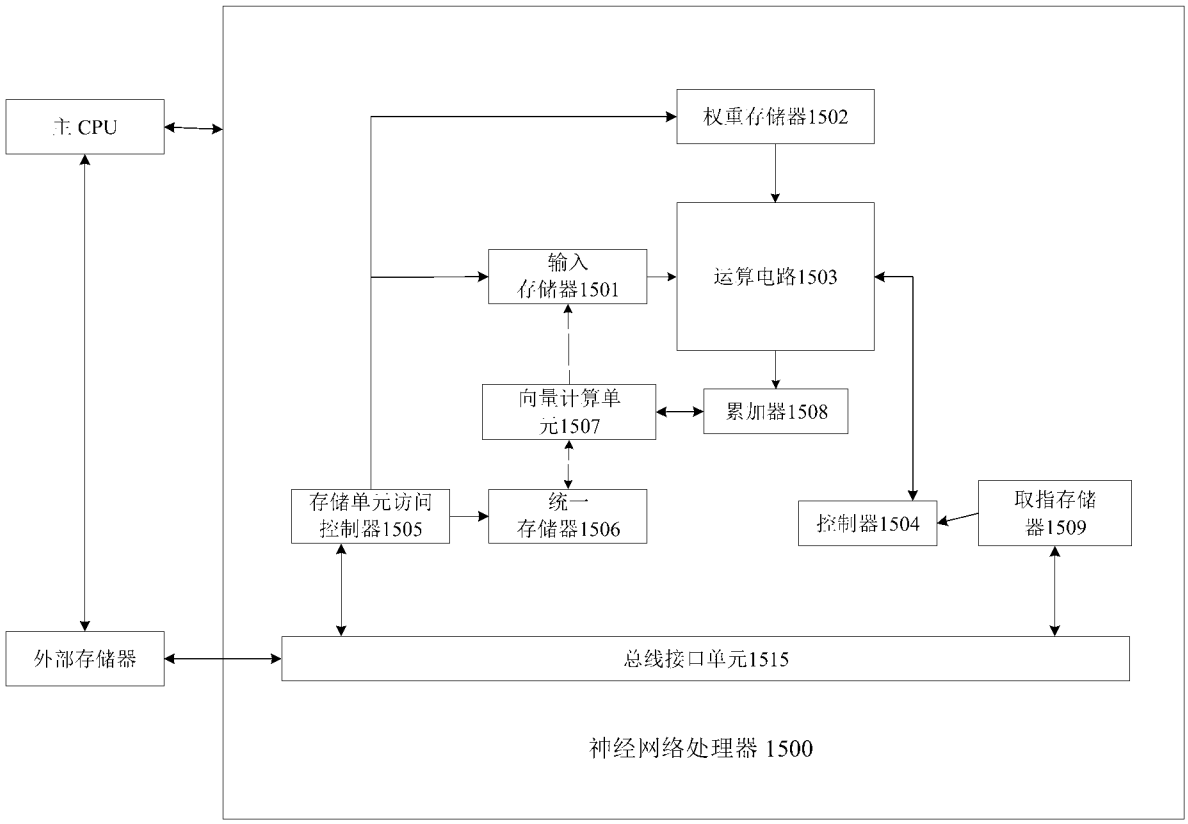

主 CPU
权重存储器1502
输入存储器1501
运算电路1503
向量计算单元1507
累加器1508
存储单元访问控制器1505
统一存储器1506
控制器1504
取指存储器1509
外部存储器
总线接口单元1515
神经网络处理器 1500

图 15


计算机可读存储介质 1600
信号承载介质 1601
程序指令 1602
计算机可读介质 1603
计算机可记录介质 1604
通信介质 1605

图 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2023/118186

A. CLASSIFICATION OF SUBJECT MATTER

G06F18/214(2023.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC:G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

VEN, CNABS, CNTXT, WOTXT, EPTXT, USTXT, CNKI, IEEE: 模型, 代码, 程序, 结构, 流程, 逻辑, 混淆, 干扰, 扰动, 增加, 添加, 分支, 选择, 判断, 表达式, 加密; AI, model, code, program, structure, flow, logic, obfuscation, disturb, add, branch, choose, expression, encrypt

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 115659169 A (HUAWEI TECHNOLOGIES CO., LTD.) 31 January 2023 (2023-01-31) claims 1-23	1-23
A	CN 114266336 A (HUAWEI TECHNOLOGIES CO., LTD.) 01 April 2022 (2022-04-01) description, paragraphs [0075]-[0144], and figures 5-6	1-23
A	CN 111177663 A (QINGDAO HAIER TECHNOLOGY CO., LTD.) 19 May 2020 (2020-05-19) entire document	1-23
A	CN 114282181 A (BEIJING DAJIA INTERNET INFORMATION TECHNOLOGY CO., LTD.) 05 April 2022 (2022-04-05) entire document	1-23
A	US 2013036473 A1 (APPLE INC.) 07 February 2013 (2013-02-07) entire document	1-23

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 October 2023	**23 October 2023**
Name and mailing address of the ISA/CN **China National Intellectual Property Administration (ISA/CN)** **China No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088**	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2023/118186

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 115659169 A	31 January 2023	None	
CN 114266336 A	01 April 2022	None	
CN 111177663 A	19 May 2020	None	
CN 114282181 A	05 April 2022	None	
US 2013036473 A1	07 February 2013	US 8751823 B2	10 June 2014

国际检索报告

国际申请号
PCT/CN2023/118186

A. 主题的分类

G06F18/214(2023.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

IPC:G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

VEN、CNABS、CNTXT、WOTXT、EPTXT、USTXT、CNKI、IEEE: 模型，代码，程序，结构，流程，逻辑，混淆，干扰，扰动，增加，添加，分支，选择，判断，表达式，加密；AI, model, code, program, structure, flow, logic, obfuscation, disturb, add, branch, choose, expression, encrypt

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
PX	CN 115659169 A (华为技术有限公司) 2023年1月31日 (2023 - 01 - 31) 权利要求1-23	1-23
A	CN 114266336 A (华为技术有限公司) 2022年4月1日 (2022 - 04 - 01) 说明书第[0075]-[0144]段，图5-6	1-23
A	CN 111177663 A (青岛海尔科技有限公司) 2020年5月19日 (2020 - 05 - 19) 全文	1-23
A	CN 114282181 A (北京达佳互联信息技术有限公司) 2022年4月5日 (2022 - 04 - 05) 全文	1-23
A	US 2013036473 A1 (APPLE INC.) 2013年2月7日 (2013 - 02 - 07) 全文	1-23

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“D” 申请人在国际申请中引证的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2023年10月16日	2023年10月23日
ISA/CN的名称和邮寄地址 中国国家知识产权局 中国北京市海淀区蓟门桥西土城路6号 100088	受权官员 张琳琳 电话号码 (+86) 010-53961404

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2023/118186

检索报告引用的专利文件	公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN 115659169 A	2023年1月31日	无	
CN 114266336 A	2022年4月1日	无	
CN 111177663 A	2020年5月19日	无	
CN 114282181 A	2022年4月5日	无	
US 2013036473 A1	2013年2月7日	US 8751823 B2	2014年6月10日