

用于生成神经网络模型的方法、装置、系统、设备、介质和程序产品

技术领域

本公开的实施例主要涉及人工智能领域。更具体地，本公开的实施例涉及用于生成神经网络模型的方法、装置、系统、设备、计算机可读存储介质以及计算机程序产品。

背景技术

近年来，联邦学习（Federated Learning）作为一种新兴的分布式机器学习技术，目前已经应用到很多场景，例如移动服务、医疗健康和物联网等。联邦学习能够充分的利用客户端处的数据和计算能力，使多方可以协作构建通用、更健壮的机器学习模型而不需要共享数据。在数据监管越来越严格的大环境下，联邦学习能够解决数据所有权、数据隐私、数据访问权等关键问题，具有极大的商业价值。

发明内容

鉴于以上，本公开的实施例提出了适用于联邦学习的生成神经网络模型的解决方案。

根据本公开的第一方面，提供了一种用于生成神经网络模型的方法，包括：第一设备向第二设备发送关于子网络模型的结构指示，子网络模型是通过调整超网络模型的结构来确定的；第一设备从第二设备接收子网络模型的参数，子网络模型的参数由第二设备基于指示和子网络模型而确定；以及第一设备基于所接收的子网络模型的参数来训练子网络模型。通过上述方式，提供了多个设备之间的高效的联邦学习方案，在提升模型精度的同时，还减少了联邦学习过程所需的通信成本、以及设备的计算成本。

在一些实施例中，方法还可以包括第一设备获取超网络模型的预配置参数，以及第一设备基于预配置参数，通过调整超网络模型的结构来确定子网络模型的结构。通过上述方式，可以在设备上生成个性化的子网络模型。

在一些实施例中，获取超网络模型的预配置参数可以包括：第一设备通过训练超网络模型来生成本地参数；第一设备向第二设备发送本地参数；以及第一设备从第二设备接收预配置参数，预配置参数由第二设备至少基于从第一设备接收的本地参数来确定。通过上述方式，能够基于设备的数据分布为超网络模型生成经过优化的预配置参数，从而有利于在较低计算成本下确定子网络模型的结构。

在一些实施例中，确定子网络模型的结构可以包括：将子网络模型初始化为具有预配置参数的超网络模型；以及通过执行以下操作至少一次，对子网络模型进行迭代更新：调整子网络模型的多个层的结构，以得到多个候选网络模型；基于多个候选网络模型的准确度，选择一个候选网络模型来更新子网络模型；并且如果子网络模型满足第一设备的约束，确定子网络模型的结构。通过上述方式，设备可以以准确度作为衡量指标来简化超网络模型，以得到满足自身资源限制和数据分布的个性化神经网络模型。

在一些实施例中，调整多个层的结构可以包括：删除与多个层中的一个层的一部分节点有关的参数，以得到多个候选网络模型中的一个。在一些实施例中，方法还包括：基于预定的数目比率，确定一部分节点。通过上述方式，可以去除对模型准确度影响较小的节点和参数，从而在简化模型结构的同时保障模型的准确度。

在一些实施例中，其中约束包括：子网络模型的计算量低于第一阈值，或者子网络模型的参数的数目低于第二阈值。通过上述方式，可以将设备的子网络模型的计算量和参数数目减少至符合相应的资源约束。

在一些实施例中，其中第一阈值和第二阈值均与第一设备的性能相关联。通过上述方式，根据设备性能来设置相应的资源约束。

在一些实施例中，指示具有掩码的形式，掩码指示子网络模型是否具有超网络模型的对应参数。通过上述方式，在联邦学习过程中，仅需要传输子网络模型特定的参数，从而减少在设备之间传输的通信成本。

在一些实施例中，根据第一方面的方法还可以包括：第一设备通过计算经训练的子网络模型的参数与从第二设备接收到的参数的差来确定参数的变化。通过上述方式，在联邦学习过程中，仅需要传输训练后发生变化的参数，从而减少设备之间的传输的通信成本。

根据本公开的第二方面，提供了一种用于生成神经网络模型的方法。该方法包括：第二设备从多个第一设备接收关于子网络模型的结构指示，子网络模型是通过调整超网络模型的结构来确定的；第二设备基于指示和超网络模型来确定子网络模型的参数；第二设备向多个第一设备发送子网络模型的参数，以用于多个第一设备分别训练子网络模型。通过上述方式，提供了多个设备之间的高效的联邦学习方案，在提升模型精度的同时，还减少了联邦学习过程所需的通信成本、以及设备的计算成本。

在一些实施例中，指示具有掩码的形式，掩码指示子网络模型是否具有超网络模型的对应参数。通过上述方式，在联邦学习过程中，仅需要传输个性化模型特定的参数，从而减少在设备之间传输的通信成本。

在一些实施例中，根据第二方面的方法还可以包括：第二设备从多个第一设备接收子网络模型在经过训练后的参数的变化。通过上述方式，在联邦学习过程中，仅需要传输训练后发生变化的参数，从而减少在服务器设备和客户端设备之间传输的通信成本。

在一些实施例中，根据第二方面的方法还可以包括：第二设备使用所接收的参数的变化来更新超网络模型。通过上述方式，在联邦学习过程中，第二设备处的超网络模型可以被迭代地更新，从而生成更符合设备第一设备的数据分布的超网络模型。

在一些实施例中，更新超网络模型可以包括：基于子网络模型的参数的更新权重来更新超网络模型，更新权重取决于具有参数的子网络模型的数目。通过上述方式，在联邦学习过程中，服务器设备的参数可以被加权更新，从而生成更符合客户端数据分布的超网络模型。

在一些实施例中，根据第二方面的方法还可以包括：第二设备确定超网络模型的预配置参数，以用于多个第一设备从超网络模型确定各自的子网络模型。通过上述方式，多个第一设备可以从具有相同预配置参数的超网络模型开始，使用本地数据来确定子网络模型的结构。

在一些实施例中，确定超网络模型的预配置参数可以包括：基于超网络模型被多个第一设备本地训练所确定的本地参数来确定超网络模型的预配置参数。通过上述方式，能够基于多个第一设备的数据分布为超网络模型生成经过优化的预配置参数，从而有利于第一设备在较低计算成本下生成个性化神经网络模型。

根据本公开的第三方面，提供了一种用于生成神经网络模型的装置，包括：发送单元，被配置为向第二设备发送关于子网络模型的结构指示，子网络模型是通过调整超网络模型的结构来确定的；接收单元，被配置为从第二设备接收子网络模型的参数，子网络模型的参

数由第二设备基于该指示和超网络模型而确定；以及训练单元，被配置为基于所接收的子网络模型的参数来训练子网络模型。发送单元还被配置为向第二设备发送经训练的子网络模型的参数，以用于第二设备更新超网络模型。通过上述方式，提供了多个设备之间的高效的联邦学习方案，在提升模型精度的同时，还减少了联邦学习过程所需的通信成本、以及设备的计算成本。

在一些实施例中，接收单元还可以被配置为获取超网络模型的预配置参数。在一些实施例中，根据第三方面的装置还包括模型确定单元，模型确定单元被配置为基于预配置参数，通过调整超网络模型的结构来确定子网络模型的结构。通过上述方式，可以在设备上生成个性化的子网络模型。

在一些实施例中，在根据第三方面的装置中，训练单元被配置为在本地训练超网络模型来确定超网络模型的本地参数。发送单元还被配置为向第二设备发送本地参数。接收单元还被配置为从第二设备接收预配置参数，预配置参数由第二设备至少基于从第一设备接收的本地参数来确定。通过上述方式，能够基于设备的数据分布为超网络模型生成经过优化的预配置参数，从而有利于设备在较低计算成本下确定子网络模型的结构。

在一些实施例中，模型确定单元还被配置为将子网络模型初始化为具有预配置参数的超网络模型。模型确定单元还被配置为通过执行以下操作至少一次，对子网络模型进行迭代更新：调整子网络模型的多个层的结构，以得到多个候选网络模型；基于多个候选网络模型的准确度，选择一个候选网络模型来更新子网络模型；并且如果子网络模型满足第一设备的约束，停止迭代更新。通过上述方式，设备可以以准确度作为衡量指标来简化超网络模型，以得到满足自身资源限制和数据分布的个性化神经网络模型。

在一些实施例中，模型确定单元还被配置为：删除与多个层中的一个层的一部分节点有关的参数，以得到多个候选网络模型中的一个。在一些实施例中，模型确定单元被配置为基于预定的数目比率，确定一部分节点。通过上述方式，可以去除对模型准确度影响较小的节点和参数，从而在简化模型结构的同时保障模型的准确度。

在一些实施例中，子网络模型的计算量低于第一阈值，或者子网络模型的参数的数目低于第二阈值。通过上述方式，可以将设备的子网络模型的计算量和参数数目减少至符合相应的资源约束。

在一些实施例中，第一阈值和第二阈值均与第一设备的性能相关联。通过上述方式，根据设备的性能大小来设置相应的资源约束。

在一些实施例中，指示具有掩码的形式，掩码指示子网络模型是否具有超网络模型的对参数。通过上述方式，在联邦学习过程中，仅需要传输子网络模型特定的参数，从而减少在服务器设备和客户端设备之间传输的通信成本。

在一些实施例中，发送单元还可以被配置为向第二设备发送经训练的子网络模型的参数的变化，以用于第二设备更新超网络模型，其中参数的变化可以被确定为经训练的子网络模型的参数与从第二设备接收到的参数的差。通过上述方式，在联邦学习过程中，仅需要传输训练后发生变化的参数，从而减少设备之间的传输的通信成本。

根据本公开的第四方面，提供了一种用于生成神经网络模型的装置，包括：接收单元，被配置为从多个第一设备接收子网络模型的结构指示，子网络模型是通过调整超网络模型的结构来确定的；子网络模型参数确定单元，被配置为基于指示和超网络模型来确定子网络

模型的参数；发送单元，被配置为向多个第一设备发送子网络模型的参数，以用于多个第一设备分别训练子网络模型；以及超网络更新单元，其中接收单元还被配置为从多个第一设备接收经训练的子网络模型的参数；并且超网络更新单元被配置为使用所接收的参数来更新超网络模型。通过上述方式，提供了多个设备之间的高效的联邦学习方案，在提升模型精度的同时，还减少了联邦学习过程所需的通信成本、以及设备的计算成本。

在一些实施例中，指示具有掩码的形式，掩码指示子网络模型是否具有超网络模型的对应参数。通过上述方式，在联邦学习过程中，仅需要传输个性化模型特定的参数，从而减少在设备之间传输的通信成本。

在一些实施例中，接收单元还可以被配置为从多个第一设备接收子网络模型在经过训练后的参数的变化。通过上述方式，在联邦学习过程中，仅需要传输训练后发生变化的参数，从而减少在服务器设备和客户端设备之间传输的通信成本。

在一些实施例中，超网络更新单元还可以被配置为：使用所接收的参数的变化来更新超网络模型。通过上述方式，在联邦学习过程中，第二设备处的超网络模型可以被迭代地更新，从而生成更符合设备第一设备的数据分布的超网络模型。

在一些实施例中，超网络更新单元还可以被配置为基于子网络模型的参数的更新权重来更新超网络模型，更新权重取决于具有参数的子网络模型的数目。通过上述方式，通过上述方式，在联邦学习过程中，服务器设备的参数可以被加权更新，从而生成更符合客户端数据分布的超网络模型。

在一些实施例中，超网络更新单元还可以被配置为确定超网络模型的预配置参数，以用于多个第一设备从超网络模型确定各自的子网络模型。通过上述方式，多个第一设备可以从具有相同预配置参数的超网络模型开始，使用本地数据来确定子网络模型的结构。

在一些实施例中，超网络更新单元还可以被配置为基于超网络模型被多个第一设备本地训练所确定的本地参数来确定预配置参数。通过上述方式，能够基于多个第一设备的数据分布为超网络模型生成经过优化的预配置参数，从而有利于第一设备在较低计算成本下生成个性化神经网络模型。

根据本公开的第五方面，提供了一种用于生成神经网络模型的系统，包括：第一设备，被配置为执行根据本公开的第一方面的方法；以及第二设备，被配置为执行根据本公开的第二方面的方法。

在本公开的第六方面，提供了一种电子设备，包括：至少一个计算单元；至少一个存储器，至少一个存储器被耦合到至少一个计算单元并且存储用于由至少一个计算单元执行的指令，指令当由至少一个计算单元执行时，使得设备实现第一方面或者第二方面中的任意一种实现方式中的方法。

在本公开的第七方面，提供了一种计算机可读存储介质，其上存储有计算机程序，其中计算机程序被处理器执行实现第一方面或者第二方面中的任意一种实现方式中的方法。

在本公开的第八方面，提供一种计算机程序产品，包括计算机可执行指令，当指令在被处理器执行时实现第一方面或者第二方面中的任意一种实现方式中的方法的部分或全部步骤。

可以理解地，上述提供的第五方法的系统、第六方面的电子设备、第七方面的计算机存储介质或者第八方面的计算机程序产品均用于执行第一方面和/或第二方面所提供的方法。因此，关于第一方面和/或第二方面的解释或者说明同样适用于第五方面、第六方面、第七方面

和第八方面。此外，第五方面、第六方面、第七方面和第八方面所能达到的有益效果可参考对应方法中的有益效果，此处不再赘述。

附图说明

结合附图并参考以下详细说明，本公开各实施例的上述和其他特征、优点及方面将变得更加明显。在附图中，相同或相似的附图标注表示相同或相似的元素，其中：

图 1 示出了根据本公开的实施例的联邦学习系统的架构图；

图 2 示出了根据本公开的实施例的用于联邦训练的通信的示意图；

图 3 示出了根据本公开的一些实施例的用于生成神经网络模型的过程的流程图；

图 4 示出了根据本公开的一些实施例的用于生成神经网络模型的另一方法的流程图；

图 5 示出了根据本公开的实施例的用于确定子网络模型的结构通信的示意图；

图 6 示出了根据本公开的一些实施例的用于初始化超网络的过程的流程图；

图 7 示出了示出了根据本公开的实施例的用于确定子网络模型的结构的过程的流程图；

图 8 示出了根据本公开的实施例的对超网络剪枝的示意图；

图 9A 至 9D 分别示出了根据本公开的一些实施例的测试结果；

图 10 示出了根据本公开的一些实施例的用于生成神经网络模型的装置的示意性框图；

图 11 示出了根据本公开的一些实施例的用于生成神经网络模型的另一装置的示意性框图；以及

图 12 示出了能够实施本公开的多个实施例的计算设备的框图。

具体实施方式

下面将参照附图更详细地描述本公开的实施例。虽然附图中显示了本公开的某些实施例，然而应当理解的是，本公开可以通过各种形式来实现，而且不应该被解释为限于这里阐述的实施例，相反提供这些实施例是为了更加透彻和完整地理解本公开。应当理解的是，本公开的附图及实施例仅用于示例性作用，并非用于限制本公开的保护范围。

在本公开的实施例的描述中，术语“包括”及其类似用语应当理解为开放性包含，即“包括但不限于”。术语“基于”应当理解为“至少部分地基于”。术语“一个实施例”或“该实施例”应当理解为“至少一个实施例”。术语“第一”、“第二”等等可以指代不同的或相同的对象。下文还可能包括其他明确的和隐含的定义。

如本文所使用的，“神经网络”能够处理输入并且提供相应输出，其通常包括输入层和输出层以及在输入层与输出层之间的一个或多个隐藏层。在深度学习应用中使用的神经网络通常包括许多隐藏层，从而延长网络的深度。神经网络的各个层按顺序相连，从而前一层的输出被提供作为后一层的输入，其中输入层接收神经网络的输入，而输出层的输出作为神经网络的最终输出。在本文中，术语“神经网络”、“网络”、“神经网络模型”和“模型”可替换地使用。

如本文所使用的，“联邦学习”是一种机器学习算法，通常由一个服务器设备和多个客户端设备组成，通过传输参数等非敏感信息，利用多个客户端数据来训练机器学习模型，以达到保护隐私的目的。

如本文所使用的，“超网络”（也称为超网络模型或父网络）是指在联邦学习环境下由服

务器和多个客户端所共享的神经网络结构。“子网络”（也称为子网络模型或个性化网络）是指联邦学习环境下由各个客户端单独维护、修改、训练和推理的神经网络模型。在本文，子网络模型和子网络可以互换使用，超网络模型和超网络可以互换使用。

子网络模型可以通过剪枝超网络而得到。这里，剪枝是指将网络模型中的部分参数和相应计算操作删除，而剩余部分的计算过程保持不变。

联邦学习是一种分布式机器学习技术，服务器设备和多个客户端设备各自在本地进行模型训练，而无需收集数据。近几年随着用户隐私保护需求的日趋加强，联邦学习也受到越来越多的关注。

然而，传统的联邦学习需要由专家设计一个全局共享的模型架构，可能并不能达到最优性能。近期有一些工作使用神经网络架构搜索（Neural Architecture Search, NAS）方法自动搜索更优的模型架构，但是这些方法都只是搜索出一个全局共享的模型，并没有考虑端侧个性化问题。在实际场景中，各个客户端设备处的数据分布可能是不同的（例如，不同用户拍摄的图片），也可能有不同的资源约束（例如，不同客户端设备具有不同的计算能力）。因此，如果把同一个模型部署在所有客户端，会导致模型性能下降，影响使用体验。另一方面，不同客户端的数据分布是不均匀的，导致一个全局模型无法在多个客户端同时达到最优性能。

针对上述问题以及其他可能存在的问题，本公开的实施例提供了一种基于联邦学习的个性化网络架构搜索框架和训练方法。首先，在联邦学习系统中通过服务器-客户端交互的方式搜索个性化模型架构，为具有差异化计算机能力的客户端提供个性化的模型架构，以满足资源约束（例如，模型大小、浮点运算数 FLOPs、推理速度等）。然后，针对不同的客户端网络架构进行有效的联邦训练。相比于仅本地训练，本公开的实施例在提升模型精度的同时，还减少了联邦学习过程所需的通信成本、以及设备的计算成本。

以下参考附图来描述本公开的各种示例实施例。

示例环境

本公开的实施例提出了一种面向资源受限设备的个性化网络结构搜索框架（Architecture of Personalization Federated Learning, APFL）。针对不同设备的特定资源需求和本地数据分布，该框架能为其定制模型架构。此外，该框架提了一种通信友好的子网络联邦训练策略，用以高效地完成设备上的本地机器学习任务。

图 1 示出了根据本公开的一些实施例的实现上述框架的联邦学习系统 100 的架构图。系统 200 包括多个第一设备 110-1、110-2...110-K（统称为第一设备 110）和第二设备 120。在本文中，第一设备可以指代客户端设备或简称为客户端，其可以是例如智能手机、台式计算机、笔记本电脑、平板电脑等具有有限计算资源（例如，处理器计算能力、内存、存储空间）的终端设备。第二设备 120 可以指代被实现在云计算环境中的分布式或集中式的服务器设备或服务器集群，其通常具有相比于终端设备更高的计算资源或性能。

多个第一设备 110 和第二设备 120 可以通过各种有线或无线方式彼此通信和传输数据。根据本公开的实施例，第一设备 110 和第二设备 120 可以以协作方式来构建和训练针对例如图像识别、语音处理、自然语音处理的各种应用的神经网络模型，即联邦学习。

如图所示，第一设备 110 包括子网络 111、本地数据 112 和控制单元 113。控制单元 113 可以虚拟或物理的处理单元，其能够例如使用本地数据 112 来训练子网络 111，使用本地数据

112 来调整子网络 111 的结构,还可以使得第一设备 110 能够与第二设备 120 通信,例如从第二设备接收子网络的参数或向服务器传输子网络参数或其他信息。第二设备 120 可以包括超网络 121 和控制单元 123。类似地,控制单元 123 也可以是虚拟或物理的处理单元,其能够维护超网络 121,例如,通过训练或者基于子网络 111 的参数来更新超网络 121 的参数,还可以使得第二设备 120 能够与第一设备 110 中的一部分或全部设备通信,例如,向第一设备 110 发送超网络 123 的参数、针对不同第一设备 110 的子网络 111 的参数,以及从第一设备 110 接收子网络 111 的参数。在本文,子网络模型和子网络可以互换使用,超网络模型和超网络可以互换使用。

超网络 123 可以在第一设备 110 和第二设备 120 之间共享,使得第一设备 110 可以剪枝超网络来确定第一设备 110 的子网络 111。所确定的子网络 111 可以满足不同第一设备 110 的资源约束和数据分布。

根据本公开的实施例,提供了一种由多个第一设备 110 和第二设备来实现的针对超网络 121 和子网络 111 的联邦训练方法。以下参照图 2 至 8 详细描述。

联邦训练

根据本公开的实施例,多个第一设备 110 和第二设备 120 彼此传输各自神经网络模型的参数及其他信息,由此实现联邦训练。

图 2 示出了根据本公开的实施例的用于联邦训练的通信 200 的示意图。如图所示,多个第一设备 110 向第二设备 120 发送 202 关于其子网络模型 111 的结构的指示。根据本公开的实施例,子网络模型 111 的结构是基于超网络 123 确定的,结构的指示可以是掩码的形式,例如维度与超网络模型维度相同的 0-1 向量。分量的值为 1,表示子网络模型的相应参数被保留,分量的值为 0,表示相应参数被从超网络 123 删除。

第二设备 120 根据接收到的指示来计算 204 针对每个第一设备 110 的子网络模型 111 的。例如,第二设备 120 可以将超网络模型 121 的参数与接收到的掩码进行分量相乘,并且截取非零分量,以得到子网络模型 111 的参数。然后第二设备将得到的子网络模型 111 的参数传输 206 给对应的第一设备。

接下来,每个第一设备 110 基于接收到的参数,使用本地数据 112 来训练 208 各种的子网络模型 111,并且在将训练后的子网络模型 111 的参数 210 传输给第二设备 120。所传输的参数可以参数值也可以是参数的变化,即更新量。

第二设备 120 使用接收到的经过第一设备 110 训练的子网络模型 111 的参数来更新超网络模型 121。例如,通过计算各个参数的平均值、加权平均或其他方式来更新超网络模型 121 的参数。根据本公开的实施例的这种联邦学习可以是迭代的,也就是说,在第二设备 120 更新超网络模型 121 之后,重复上述 204、206、208、210、212 的动作。此外,每次参与上述过程的第一设备 110 可以是变化的,例如,可以是当前在线的第一设备 110 的任何子集。

以下参照图 3 和图 4 分别描述第一设备 110 和第二设备 120 执行联邦学习以生成神经网络模型的过程,其中第二设备 120 维护超网络 $\hat{W}$,每个第一设备 110 的个性化网络 $\hat{W}_k$ 是 $\hat{W}$ 的一个子网络,并且各个子网络 $\hat{W}_1, \dots, \hat{W}_K$ 的架构可以是不同的。

图 3 示出了根据本公开的一些实施例的用于生成神经网络模型的过程 300 的流程图。过程 300 可以适于在 K 个第一设备 110 中每个设备 u_k ($k = 1, \dots, K$) 实现,更具体地,由控制单

元 113 来执行。

在框 310, 向第二设备发送子网络模型的结构指示, 子网络模型是通过调整超网络模型的结构来确定的。在一些实施例中, 指示可以掩码 (mask) 的形式, 指示子网络模型 111 是否具有超网络模型 121 的对应参数。例如, 第一设备 u_k 发送掩码 z_k 到第二设备 120。其中 z_k 是一个与超网络 121 $\hat{w}$ 维度相同的 0-1 向量, 代表子网络结构 $\hat{w}_k$, 即如果 z_k 里分量的值为 1, 则代表超网络 121 中相应的参数在 $\hat{w}_k$ 中被保留, 如果分量值为 0 则代表相应参数被删除。由此, 第二设备 120 可以获知第一设备 110 的子网络模型 111 的结构, 从而可以为每个第一设备确定子网模型 111 的参数。例如, 在子网络联邦训练开始时, 第二设备 120 的超网络 121 具有预配置参数, 即设置 $\hat{w}^{(0)} = \hat{w}$ 。

根据本公开的实施例, 子网络联邦训练可以是迭代执行的, 例如执行 T' 轮。在迭代 $t = 0, \dots, T' - 1$ 中, 可以从 K 个第一设备 110 中选择 K' ($K' \leq K$) 个第一设备来参与该轮迭代。对于被选择的第一设备, 执行如框 720 至 740 的动作。

在框 320, 从第二设备接收子网络模型的参数。根据本公开的实施例, 由第二设备 120 通过超网络模型 121 的当前参数和该子网络模型的掩码来计算子网络模型 111 的参数。例如, 计算 $\hat{w}^{(t)} \circ z_{t_k}$, 然后, 第二设备 120 可以截取非零分量 $\hat{w}_{t_k}^{(t)}$ 传输给第一设备 u_k 。这里 $\circ$ 代表向量分量相乘。由此, 第一设备 110 接收子网络模型 111 的参数。

然后, 在框 330, 基于所接收的参数来训练子网络模型。在一些实施例中, 第一设备 u_k 接收并给子网络模型 111 的参数赋值为 $\hat{w}_{t_k}^{(t)}$, 用本地训练数据 112 更新子网络参数得到经更新的参数 $\hat{w}_{t_k}^{(t)}$ 。例如, 可以用本地训练数据进行 S' 步梯度下降, 如以下等式 (1) 所示

$$\begin{aligned} \hat{w}_{t_k,0}^{(t)} &= \hat{w}_{t_k}^{(t)} \\ \hat{w}_{t_k,s+1}^{(t)} &= \hat{w}_{t_k,s}^{(t)} - \lambda' \nabla L(\hat{w}_{t_k,s}^{(t)}; D_{t_k,s}), s = 0, 1, \dots, S' - 1 \\ \hat{w}_{t_k}^{(t)} &= \hat{w}_{t_k,S'}^{(t)} \end{aligned} \quad (1)$$

其中 λ 是学习率, L 是交叉熵损失函数, $D_{t_k,s}$ 是训练集 D_{t_k} 的一个随机采样。

在框 340, 向第二设备发送经训练的子网络模型的参数。在一些实施例中, 第一设备 110 向第二设备 120 发送子网络模型 111 的参数可以包括计算参数的变化 $v_{t_k}^{(t)} = \hat{w}_{t_k}^{(t)} - \hat{w}_{t_k}^{(t)}$, 并且将计算出的 $v_{t_k}^{(t)}$ 传输给第二设备。备选地, 第二设备 120 也可以直接将经训练的子网络模型的参数发送给第二设备 120, 而不计算参数的变化。

相应地, 第二设备 120 从 K' 个客户端接收经过训练的子网络模型 111 的参数, 并且基于接收到的参数来更新超网络 121。如上, 第二设备 120 接收的参数可以是参数的变化或更新量, 也可以是更新后的子网络模型 121 的参数。

图 4 示出了根据本公开的实施例的用于生成神经网络模型的过程 400 的流程图。过程 400 可以适于在图 2 所示的第二设备 120 来实施, 更具体地, 由控制单元 123 来执行。

在框 410, 从多个第一设备 110 接收子网络模型 111 的结构指示, 子网络模型 111 是通过调整超网络模型 121 的结构来确定的。在一些实施例中, 指示具有掩码的形式, 掩码指示子网络模型是否具有超网络模型的对应参数。例如, 掩码可以是 0-1 向量, 如果该向量里的分量的值为 1, 则代表超网络中相应的参数在子网络模型中被保留, 如果分量值为 0 则代表相应参数被删除。

在框 420, 基于指示和超网络模型来确定子网络模型的参数。在一些实施例中, 第二设备 120 将掩码形式的指示和超网络 121 的参数做分量相乘, 并确定其中的非零向量, 作为子

网络模型 111 的参数。

在框 430, 向多个第一设备发送子网络模型的参数, 以用于多个第一设备分别训练子网络模型。由此, 第一设备 110 在接收到子网络模型 111 的参数后, 使用本地训练数据 112, 通过梯度下降法来训练本地的个性化子网络模型 111。如上所述, 在训练后, 第一设备 110 向第二设备 120 发送将经训练的子网络模型 111 的参数。

相应地, 在框 440, 从多个第一设备接收经训练的子网络模型的参数。在一些实施例中, 第二设备 120 接收的经训练的子网络模型的参数包括参数的变化, 参数的变化可以是参数在子网络模型训练前后的差。

在框 450, 使用所接收的参数来更新超网络模型。在一些实施例中, 基于子网络模型 111 的参数的更新权重来更新超网络模型 121, 更新权重取决于这些第一设备 110 的子网络模型中具有相应参数的子网络模型的数目。例如, 更新权重 Z_t 可以按照以下等式 (2) 设置:

$$Z_t = \text{Recip} \left(\sum_{k=1}^{K'} Z_{t_k} \right) \quad (2)$$

其中 $\text{Recip}(x)$ 对向量 x 中的每个分量取倒数, 如果分量值为 0 则置为 0。

然后, 第二设备 120 基于更新权重和接收到的子网络模型 111 的参数来更新超网络参数。按照如下等式 (3) 来更新超网络参数:

$$\hat{W}^{(t+1)} = \hat{W}^{(t)} + \sum_{k=1}^{K'} Z_t \circ v_{t_k}^{(t)} \quad (3)$$

其中 $\hat{W}^{(t+1)}$ 是本轮迭代后的超网络参数, $\hat{W}^{(t)}$ 是本地迭代开始时的超网络参数, Z_t 是更新权重, $v_{t_k}^{(t)}$ 是第一设备 110 处的参数的变化, $\circ$ 代表向量分量相乘。

可以根据需要将上述联邦学习过程重复执行 T 次, 完成第二设备和若干个第一设备之间的联邦训练过程。

作为生成神经网络模型的结果, 针对每个客户端 u_k , $k=1, \dots, K$, 第二设备 120 基于第一设备 110 的结构或掩码来确定子网络模型 111 的训练结果, 具体方式如下: 第二设备 120 计算 $\hat{W}^{(T)} \circ Z_k$, 并截取非零分量 $\hat{W}_k^{(T)}$ 传输给特定第一设备 u_k , 该特定第一设备 u_k 接收并设置最终子网络参数为 $\tilde{W}_k = \hat{W}_k^{(T)}$ 。

子网络个性化

如上所述, 子网络模型 111 是由第一设备 110 通过调整超网络模型 123 的结构而确定, 并且所确定的子网络模型 111 能够满足各个第一设备 110 的资源约束和数据分布。本公开的实施例还提供了生成这种个性化子网络的方法, 以下结合图 5 至图 8 描述。

图 5 示出了根据本公开的实施例的用于确定子网络模型的结构通信 500 的示意图。根据本公开的实施例, 基于联邦学习方式来确定适合于第一设备 110 的个性化子网络模型。

首先, 第二设备 120 向多个第一设备 110 传输超网络 121 的参数。在一些实施例中, 该多个第一设备 110 可以是所有第一设备 110 的一部分, 优选是具有较高性能的第一设备。此时, 第一设备 110 只有共享的超网络, 第一设备 110 使用接收到的参数来训练 504 超网络。

然后, 第一设备 110 向第二设备 120 传输 506 经训练超网络的参数。第二设备 120 可以对这些参数进行汇总 508, 以更新超网络 121。经更新的超网络的参数可以作为预配置参数被传输 510 到第一设备 110。然后, 第一设备 120 可以将利用预配置参数来初始化本地的超网

络, 然后调整 (例如, 剪枝) 超网络的结构, 以确定满足自身资源约束和数据分布的子网络模型 111。

在一些实施例中, 第二设备 120 的超网络 123 可以被迭代地更新, 即, 第二设备 120 在更新超网络 123 后可以重复动作 502 至 508。并且, 每次迭代可以选择不同的第一设备 110。

以下结合图 6 至图 8 详细描述第一设备 110 和第二设备 120 执行联邦学习以确定个性化子网络 111 的过程。

图 6 示出了根据本公开的一些实施例的用于初始化超网络的过程 400 的流程图。过程 400 可以在第二设备 120 实施, 更具体地由控制单元 123 来执行。通过该过程, 可以确定超网络 121 的预配置参数, 以用于第一设备 110 后续生成个性化子网络模型 111。

在框 410, 选择超网络, 并生成初始参数。根据本公开的实施例, 第二设备 21 可以从神经网络模型库中选择一个神经网络模型来作为超网络 121, 例如 MobileNetV2。神经网络模型可以包括输入层、多个隐藏层以及输出层, 每个层包括多个节点、层与层之间的节点通过具有权重的边连接。权重可以被称为模型参数 w , 其可以通过训练而被学习。取决于网络模型中的层数或模型深度、各个层的节点数、以及边的数目, 神经网络模型具有相应的资源预算, 包括模型大小、浮点运算数、参数个数、推理速度等。在一些实施例中, 第二设备 120 可以随机生成超网络 121 的初始参数 $w^{(0)}$ 。

如图所示, 框 620 至框 650 可以被迭代地执行多次 (例如, T 轮, T 为正整数), 通过迭代 ($t = 0, 1, \dots, T-1$) 来更新超网络 121 的参数来确定超网络 121 的预配置参数。具体地, 在框 620, 向选择的多个第一设备 110 发送超网络参数, 以用于多个第二设备 120 在本地训练超网络。根据本公开的实施例, 第二设备 120 可以从在线的 K 个第一设备 110 中选择一部分第一设备, 例如, 可以从 K 个第一设备 110 中随机选择或优选性能较高的 K' 个客户端 $u_{t_1}, \dots, u_{t_{K'}}$, 第二设备 120 将当前的超网络参数 $w^{(t)}$ 发送给所选择的第一设备。然后, 每个被选择的第一设备 u_k 利用本地训练数据 112 得到超网络的本地参数 $w_k^{(t+1)}$ 。注意, 此时第一设备 110 还没有生成个性化子网络 111, 第一设备 110 使用本地训练数据 112 来训练在框 610 处确定的超网络。

在一些实施例中, 每个被选取的第一设备 u_k 利用本地训练数据 112, 通过 S 步梯度下降得到新的超网络参数 $w_k^{(t+1)}$, 如等式 (4) 所示。

$$\begin{aligned} w_{t_k,0}^{(t)} &= w^{(t)} \\ w_{t_k,s+1}^{(t)} &= w_{t_k,s}^{(t)} - \lambda \nabla L(w_{t_k,s}^{(t)}; D_{t_k,s}), s = 0, 1, \dots, S-1 \\ w_{t_k}^{(t+1)} &= w_{t_k,S}^{(t)} \end{aligned} \quad (4)$$

其中 λ 是学习率, L 是交叉熵损失函数, $D_{t_k,s}$ 是训练集 D_{t_k} 的一个随机采样。

然后, 第一设备 u_k 将本轮次迭代所生成的超网络参数 $w_{t_k}^{(t+1)}$ 传输到第二设备 120。

相应地, 在框 630, 从多个第一设备 110 接收经训练的超网络的本地参数 $w_{t_k}^{(t+1)}$ 。

然后, 框 640, 更新超网络的全局参数。根据本公开的实施例, 第二设备 120 根据接收到的客户端本地参数来更新超网络 121 的全局参数。例如, 第二设备 120 根据下等式 (5) 来更新全局参数:

$$w^{(t+1)} = \text{Update}(w_{t_1}^{(t+1)}, \dots, w_{t_{K'}}^{(t+1)}) \quad (5)$$

其中 $w^{(t+1)}$ 表示在 $t+1$ 次迭代后的超网络的全局参数 $w_{t_1}^{(t+1)}, \dots, w_{t_{K'}}^{(t+1)}$, 表示参与本次迭代的第一设备 110 生成的本地超网络参数。Update() 函数可以是预先设置的更新算法, 在一些实施例

中，基于训练数据量的加权平均来计算全局参数，如以下等式（6）所示

$$w^{(t+1)} = \sum_{k=1}^{K'} \frac{n_{t_k}}{n_t} w_{t_k}^{(t+1)} \quad (6)$$

其中 n_{t_k} 是客户端 u_k 训练数据集的大小， $n_t = \sum_{k=1}^{K'} n_{t_k}$ 。备选地，还可以计算参数的算术平均值作为超网络的全局参数。

在框 650，判断是否达到了 T 轮。如果没有达到 T 轮，可以返回框 620，进行下一轮迭代。如果达到 T 轮，则在框 660，将超网络 121 的当前全局参数确定超网络 121 的预配置参数，如等式（7）所示

$$\hat{w} = w^{(T)} \quad (7)$$

其中 $\hat{w}$ 是超网络 121 的预配置参数，其可以用于第一设备 110 生成个性化子网络 111。

在一些实施例中，第二设备 120 可以在迭代更新超网络 121 的全局参数时，根据第一设备 110 的性能级别来选择参与迭代的 K' 个第一设备 110，以加快超网络 121 的初始化。

在完成超网络的全局初始化之后，第二设备 120 维护具有参数 $\hat{w}$ 的超网络 121。提供了一种个性化设备上的神经网络模型的方法，其能够通过剪枝来简化超网络 121 的结构，以生成个性化子网络 111，并且所生成的个性化子网络符合客户端的数据分布和资源约束。以下参照图 7 和图 8 来详细描述。

图 7 示出了根据本公开的实施例的用于确定子网络模型的结构的过程 700 的流程图。过程 700 可以适于在 K 个第一设备 110 中每个设备执行。过程 700 通过迭代地删除超网络 121 的节点和参数来简化超网络，同时还确保了所得到的子网络 111 的具有良好的准确度。

根据本公开的实施例，基于网络模型的准确度来剪枝超网络 121，例如，可以根据某种规则选取对准确率影响较小的参数。此外，还将资源约束当作一个硬性约束条件，每次剪枝操作需要删除至少一个参数以简化超网络。如上，超网络 121 可以由多个第一设备 110 和第二设备 120 之前通过上文参照图 6 描述的过程被初始化，因此可以能够达到较好的准确度，以用于被剪枝以生成个性化子网络模型 111。

在框 710，初始化子网络模型。根据本公开的实施例，第一设备 110 u_k 接收超网络 121 的预配置参数 $\hat{w}$ ，并且将本地子网络模型 111 初始化为具有该预配置参数的超网络，即， $\hat{w}_k = \hat{w}$ ，其中 $\hat{w}_k$ 表示客户端 k 处的子网络。也就是说，在开始时，第一设备的子网络模型 111 具有和第二设备 120 的超网络相同的结构和参数，并由此开始剪枝。本公开的实施例提供了一种逐层剪枝的方法，其迭代简化超网络的结构直到符合资源约束，同时可以确保所得子网络具有良好的准确度。

参照 MobileNetV2 来描述其被剪枝的过程，但是应理解，其他类型的神经网络模型适用于本公开的实施例提供的剪枝方法。

图 8 示出了根据本公开的实施例的对超网络剪枝的示意图。采用 MobileNetV2 作为超网络，使用通道（神经元）剪枝操作，即以通道为单位进行剪枝。网络 800 中某些部分无法进行剪枝，例如 depthwise 层、输入层、输出层等，而其中 34 层可以进行剪枝。假设可剪枝层为 $L_1, \dots, L_{34}$ ，剪枝层 L_i 包含 m_i 个通道 $C_1, \dots, C_{(m_i)}$ ，例如，第一设备 110 的本地数据可以为 $32 \times 32 \times 3$ 的图片格式，即输入层有 3 个通道，对应图片的 RGB 值。如图所示，实线箭头表示相邻层之间的计算连接，标记 X 表示被剪枝的通道（节点），虚线箭头表示相应被删除的连接，未被剪枝的通道计算过程保持不变。

在框 720，调整子网络模型的多个层的结构，得到多个候选神经网络模型。具体地，针

对多个可剪枝层，删除与子网络模型 111 的每个可剪枝层的一部分节点有关的参数，得到多个候选网络模型。在一些实施例中，针对每一层，可以基于预定的数目比率（例如，0.1、0.2、0.25 等）来确定一部分节点，并且删除与这些节点有关的参数。然后，可以使用第一设备的本地数据来确定所得到的经剪枝的候选神经网络模型的准确度。

例如，第一设备 u_k 对 $\hat{w}_k$ 进行剪枝，针对 MobileNetV2 的第 i 可剪枝层， $i = 1, \dots, 34$ ，执行如下操作：

1. 删除标号大于 $\lfloor 0.9m_i \rfloor$ 的通道，即保留通道 $C_1, \dots, C_{\lfloor 0.9m_i \rfloor}$ 。
2. 设删除的相应参数为 v_i 。
3. 计算网络 $\hat{w}_k \setminus v_i$ 在训练集 D_k 上的准确率，记为 A_i 。// 注意，计算准确率后 $\hat{w}_k$ 复原

然后，在框 730，基于多个候选网络模型的准确度来选择一个候选网络模型来更新子网络模型。在一些实施例中，可以确定被剪枝后准确率最高的可剪枝层 $i^* = \arg\max_i A_i$ 。由此，可以从子网络模型 $\hat{w}_k$ 中删除参数 v_{i^*} ，更新 $\hat{w}_k = \hat{w}_k \setminus v_{i^*}$ 。

在框 740，确定当前的子网络模型是否符合资源约束。在一些实施例中，可以将诸如浮点运算数（FLOPs）的计算量或模型的参数的数目为资源约束。例如，可以将第一设备 110 分类如下：(i) 高预算客户端，FLOPs 上限为 88M，参数量上限为 2.2M；(ii) 中预算客户端，FLOPs 上限为 59M，参数量上限为 1.3M；(iii) 低预算客户端，FLOPs 上限为 28M，参数量上限为 0.7M。在一些实施例中，可以考虑两种联邦学习设置：(1) 高性能配置，其中高、中、低预算客户端比例为 5:3:2；以及 (2) 低性能配置，其中高、中、低预算客户端比例为 2:3:5。由此兼顾此后的联邦训练的效率和准确度。

如果当前的子网络模型不符合资源约束，例如，子网络模型的计算量高于或等于某一阈值，或者子网络模型的参数的数目高于或等于另一阈值，或者上述二者均成立，则重复上述框 720 和 730，以进一步剪枝。在一些实施例中，上述关于计算量的阈值和关于参数的数目的阈值取决于第一设备 110 的性能。第一设备 110 的性能越高，则阈值可以被设置为更高，从而对子网络模型 111 进行较少的剪枝，这意味着可以在训练后实现更高的准确度。

如果当前的子网络模型 111 符合资源约束，即，子网络模型 111 的计算量低于第一阈值，或者子网络模型的参数的数目低于第二阈值，或者上述二者均成立，则在框 750，确定当前子网络模型的结构，作为第一设备 110 的用于联邦训练的个性化子网络。

值得注意的是，在上述过程中，子网络模型 111 的结构被迭代地调整，但是子网络模型 111 的参数值保持不变，即维持从第二设备 120 接收到的预配置参数 $\hat{w}$ （如果不被删除）。

以上描述了确定个性化子网络模型 111 的过程。所确定的子网络模型的结构可以通过例如掩码来指示，以用于联邦训练。

测试结果

图 9A、图 9B、图 9C 和图 9D 示出了根据本公开的一些实施例的测试结果。测试结果表明根据本公开的实施例的方案（FedCS）在分类任务上取得了优越性能，经过联邦学习和训练的个性化网络架构具有优势。

如图 9A 所示，对于高性能的客户端，本公开的方案 FebCS 相比于其他方法（FedUniform），在 IID 和 NIID 两类数据均有更高的准确率。对于低性能的客户端，优势更为明显。

如图 9B 所示，本公开的网络架构搜索框架（FedCS）所需的服务器-客户端之间的通信

量明显减少，这表明具有更快的收敛速度，节约了通信成本。

同时，搜索得到的模型在浮点运算数 (FLOPs) 约束相当的情况下，具有更小的参数量，因此节约客户端的存储空间，如下图 9C 所示。

另外，如图 9D 所示，相比于客户端单独训练策略，本公开的子网络联邦训练算法在准确率上有大幅提升，表明虽然各客户端子网络架构不同，但使用掩码策略通过超网络进行联邦训练，可以有效提升子网络性能。

示例装置和设备

图 10 示出了根据本公开的一些实施例的用于生成神经网络模型的装置 1000 的示意性框图。装置 1000 可以实现在如图 1 所示的第一设备 110 中。

装置 1000 包括发送单元 1010，发送单元 1010 被配置为向第二设备发送关于子网络模型的结构指示，子网络模型是通过调整超网络模型的结构来确定的。装置 1000 还包括接收单元 1020，接收单元 1020 被配置为从第二设备接收子网络模型的参数，子网络模型的参数由第二设备基于该指示和超网络模型而确定。装置 1000 还包括训练单元 1030，训练单元 1030 被配置为基于所接收的子网络模型的参数来训练子网络模型。发送单元 1010 还被配置为向第二设备发送经训练的子网络模型的参数，以用于第二设备更新超网络模型。

在一些实施例中，接收单元 1020 还可以被配置为获取超网络模型的预配置参数。在一些实施例中，装置 1000 还包括模型确定单元 1040，模型确定单元 1040 被配置为基于预配置参数，通过调整超网络模型的结构来确定子网络模型的结构。

在一些实施例中，训练单元 1030 被配置为在本地训练超网络模型来确定超网络模型的本地参数。发送单元 1010 还被配置为向第二设备发送本地参数。接收单元 1020 还被配置为从第二设备接收预配置参数，预配置参数由第二设备至少基于从第一设备接收的本地参数来确定。

在一些实施例中，模型确定单元 1040 还被配置为将子网络模型初始化为具有预配置参数的超网络模型。模型确定单元 1040 还被配置为通过执行以下操作至少一次，对子网络模型进行迭代更新：调整子网络模型的多个层的结构，以得到多个候选网络模型；基于多个候选网络模型的准确度，选择一个候选网络模型来更新子网络模型；并且如果子网络模型满足第一设备的约束，停止迭代更新。

在一些实施例中，模型确定单元 1040 还被配置为：删除与多个层中的一个层的一部分节点有关的参数，以得到多个候选网络模型中的一个。在一些实施例中，模型确定单元 1040 被配置为基于预定的数目比率，确定一部分节点。通过上述方式，

在一些实施例中，子网络模型的计算量低于第一阈值，或者子网络模型的参数的数目低于第二阈值。

在一些实施例中，第一阈值和第二阈值均与第一设备的性能相关联。

在一些实施例中，指示具有掩码的形式，掩码指示子网络模型是否具有超网络模型的对应参数。

在一些实施例中，发送单元 1010 还可以被配置为向第二设备发送经训练的子网络模型的参数的变化，以用于第二设备更新超网络模型，其中参数的变化可以被确定为经训练的子网络模型的参数与从第二设备接收到的参数的差。

图 11 示出了根据本公开的一些实施例的用于生成神经网络模型的装置 1100 的示意性框图。装置 1100 可以实现在如图 1 所示的第二设备 120 中。

装置 1100 包括接收单元 1110，接收单元 1110 被配置为从多个第一设备接收子网络模型的结构指示，子网络模型是通过调整超网络模型的结构来确定的。装置 1100 还包括子网络模型参数确定单元 1120，子网络模型参数确定单元 1120 被配置为基于指示和超网络模型来确定子网络模型的参数。装置 1100 还包括发送单元 1130，发送单元 1130 被配置为向多个第一设备发送子网络模型的参数，以用于多个第一设备分别训练子网络模型。装置 1100 还包括超网络更新单元 1140，超网络更新单元 1140 被配置为使用通过接收单元 1110 从多个第一设备接收经训练的子网络模型的参数来更新超网络模型。

在一些实施例中，指示具有掩码的形式，掩码指示子网络模型是否具有超网络模型的对应参数。

在一些实施例中，接收单元 1110 还可以被配置为从多个第一设备接收子网络模型在经过训练后的参数的变化。通过上述方式，

在一些实施例中，超网络更新单元 1140 还可以被配置为：使用所接收的参数的变化来更新超网络模型。

在一些实施例中，超网络更新单元 1140 还可以被配置为基于子网络模型的参数的更新权重来更新超网络模型，更新权重取决于具有参数的子网络模型的数目。

在一些实施例中，超网络更新单元 1140 还可以被配置为确定超网络模型的预配置参数，以用于多个第一设备从超网络模型确定各自的子网络模型。

在一些实施例中，超网络更新单元 1140 还可以被配置为基于超网络模型被多个第一设备本地训练所确定的本地参数来确定预配置参数。通过上述方式，

图 12 示出了能够实施本公开的多个实施例的计算设备 1200 的框图。设备 1200 可以用于实现第一设备 110、第二设备 120、装置 1000 和 1100。如图所示，设备 1200 包括计算单元 1201，其可以根据存储在随机存取存储器（RAM）和/或只读存储器（ROM）1202 的计算机程序指令或者从存储单元 1207 加载到 RAM 和/或 ROM 1202 中的计算机程序指令，来执行各种适当的动作和处理。在 RAM 和/或 ROM 1202 中，还可存储设备 1200 操作所需的各种程序和数据。计算单元 1201 和 RAM 和/或 ROM 1202 通过总线 1203 彼此相连。输入/输出（I/O）接口 1204 也连接至总线 1203。

设备 1200 中的多个部件连接至 I/O 接口 1204，包括：输入单元 1205，例如键盘、鼠标等；输出单元 1206，例如各种类型的显示器、扬声器等；存储单元 1107，例如磁盘、光盘等；以及通信单元 1208，例如网卡、调制解调器、无线通信收发机等。通信单元 1208 允许设备 1200 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。

计算单元 1101 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。计算单元 1201 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计算单元、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。计算单元 1201 执行上文所描述的任何方法和处理。例如，在一些实施例中，上述过程可被实现为计算机软件程序，其被有形地包含于机器可读介质，例如存储单元 1207。在一些实施例中，计算机程序的部分或者全部可以经由 RAM 和/或 ROM 和/或通信单元 1208 而被载入和/或安装到设备 1200 上。当计算机程序加载到 RAM

和/或 ROM 并由计算单元 1201 执行时，可以执行上文描述的任何过程中的一个或多个步骤。备选地，在其他实施例中，计算单元 1201 可以通过其他任何适当的方式（例如，借助于固件）而被配置为执行上文所描述的任何方法和处理。

用于实施本公开的方法的程序代码可以采用一个或多个编程语言的任何组合来编写。这些程序代码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或控制器，使得程序代码当由处理器或控制器执行时使流程图和/或框图中所规定的功能/操作被实施。程序代码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服务器上执行。

在本公开的上下文中，机器可读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程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读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。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。

此外，虽然采用特定次序描绘了各操作，但是这应当理解为要求这样操作以所示出的特定次序或以顺序次序执行，或者要求所有图示的操作应被执行以取得期望的结果。在一定环境下，多任务和并行处理可能是有利的。同样地，虽然在上面论述中包含了若干具体实现细节，但是这些不应当被解释为对本公开的范围的限制。在单独的实施例的上下文中描述的某些特征还可以组合地实现在单个实现中。相反地，在单个实现的上下文中描述的各种特征也可以单独地或以任何合适的子组合的方式实现在多个实现中。

尽管已经采用特定于结构特征和/或方法逻辑动作的语言描述了本主题，但是应当理解所附权利要求书中所限定的主题未必局限于上面描述的特定特征或动作。相反，上面所描述的特定特征和动作仅仅是实现权利要求书的示例形式。

权利要求书

1. 一种用于生成神经网络模型的方法，包括：

所述第一设备向第二设备发送关于子网络模型的结构指示，所述子网络模型是通过调整超网络模型的结构来确定的；

所述第一设备从所述第二设备接收所述子网络模型的参数，所述子网络模型的参数由所述第二设备基于所述指示和所述超网络模型而确定；

所述第一设备基于所接收的所述子网络模型的参数来训练所述子网络模型；以及

所述第一设备向所述第二设备发送经训练的所述子网络模型的参数，以用于所述第二设备更新所述超网络模型。

2. 根据权利要求1所述的方法，还包括：

所述第一设备获取所述超网络模型的预配置参数；以及

所述第一设备基于所述预配置参数，通过调整所述超网络模型的结构来确定所述子网络模型的所述结构。

3. 根据权利要求2所述的方法，其中获取所述超网络模型的预配置参数包括

所述第一设备在本地训练所述超网络模型来确定所述超网络模型的本地参数；

所述第一设备向所述第二设备发送所述本地参数；以及

所述第一设备从所述第二设备接收所述预配置参数，所述预配置参数由所述第二设备至少基于从所述第一设备接收的所述本地参数来确定。

4. 根据权利要求2所述的方法，其中确定所述子网络模型的结构包括：

将所述子网络模型初始化为具有所述预配置参数的超网络模型；以及

通过执行以下操作至少一次，对所述子网络模型进行迭代更新

调整所述子网络模型的多个层的结构，以得到多个候选网络模型；

基于多个候选网络模型的准确度，选择一个候选网络模型来更新所述子网络模型；并且

如果所述子网络模型满足所述第一设备的约束，确定所述子网模型的结构。

5. 根据权利要求4所述的方法，调整所述多个层的结构包括：

删除与所述多个层中的一个层的一部分节点有关的参数，以得到所述多个候选网络模型中的一个。

6. 根据权利要求5所述的方法，还包括：

基于预定的数目比率，确定所述一部分节点。

7. 根据权利要求1所述的方法，其中所述约束包括：所述子网络模型的计算量低于第一阈值，或者所述子网络模型的参数的数目低于第二阈值。

8. 根据权利要求7所述的方法，其中所述第一阈值和所述第二阈值均与所述第一设备的性能相关联。

9. 根据权利要求1所述的方法，其中所述指示具有掩码的形式，所述掩码指示所述子网络模型是否具有所述超网络模型的对应参数。

10. 根据权利要求1所述的方法，发送经训练的所述子网络模型的参数包括：

通过计算经训练的所述子网络模型的参数与从所述第二设备接收到的参数的差来确定所述参数的变化；以及

向所述第二设备发送所述参数的变化。

11. 一种用于生成神经网络模型的方法，包括：

第二设备从多个第一设备接收关于子网络模型的结构指示，所述子网络模型是通过调整超网络模型的结构来确定的；

所述第二设备基于所述指示和所述超网络模型来确定所述子网络模型的参数；

所述第二设备向所述多个第一设备发送所述子网络模型的参数，以用于所述多个第一设备分别训练所述子网络模型；

所述第二设备从所述多个第一设备接收经训练的所述子网络模型的参数；以及

所述第二设备使用所接收的所述参数来更新所述超网络模型。

12. 根据权利要求 11 所述的方法，其中所述指示具有掩码的形式，所述掩码指示所述子网络模型是否具有所述超网络模型的对应参数。

13. 根据权利要求 11 所述的方法，其中接收经训练的所述子网络模型的参数还包括：

从所述多个第一设备接收所述子网络模型在经过训练后的参数的变化。

14. 根据权利要求 11 所述的方法，其中更新所述超网络模型包括：

基于所述子网络模型的参数的更新权重来更新所述超网络模型，所述更新权重取决于具有所述参数的所述子网络模型的数目。

15. 根据权利要求 11 所述的方法，还包括：

所述第二设备确定所述超网络模型的预配置参数，以用于所述多个第一设备从所述超网络模型确定各自的子网络模型。

16. 根据权利要求 15 所述的方法，其中确定所述预配置参数包括：

基于所述超网络模型被多个第一设备本地训练所确定的本地参数来确定所述预配置参数。

17. 一种用于生成神经网络模型的装置，包括：

发送单元，被配置为向第二设备发送关于所述子网络模型的结构指示，所述子网络模型是通过调整超网络模型的结构来确定的；

接收单元，被配置为从所述第二设备接收所述子网络模型的参数，所述子网络模型的参数由所述第二设备基于所述指示和所述超网络模型而确定；以及

训练单元，被配置为基于所接收的所述子网络模型的参数来训练所述子网络模型；

其中所述发送单元还被配置为向所述第二设备发送经训练的所述子网络模型的参数，以用于所述第二设备更新所述超网络模型。

18. 根据权利要求 17 所述的装置，其中所述接收单元还被配置为获取超网络模型的预配置参数，并且所述装置还包括：

所述模型确定单元，被配置为基于所述预配置参数，通过调整所述超网络模型的结构来确定所述子网络模型的所述结构。

19. 根据权利要求 18 所述的装置，其中所述模型确定单元还被配置为：

将所述子网络模型初始化为具有所述预配置参数的超网络模型；以及

通过执行以下操作至少一次，对所述子网络模型进行迭代更新

调整所述子网络模型的多个层的结构，以得到多个候选网络模型；

基于多个候选网络模型的准确度，选择一个候选网络模型来更新所述子网络模型；

并且

如果所述子网络模型满足所述第一设备的约束，确定所述子网模型的结构。

20. 一种用于生成神经网络模型的装置，包括：

接收单元，被配置为从多个第一设备接收关于子网络模型的结构指示，所述子网络模型是通过调整超网络模型的结构来确定的；

子网络模型参数确定单元，被配置为基于所述指示和所述超网络模型来确定所述子网络模型的参数；

发送单元，被配置为向所述多个第一设备发送所述子网络模型的参数，以用于所述多个第一设备分别训练所述子网络模型；以及

超网络更新单元，

其中所述接收单元还被配置为从所述多个第一设备接收经训练的所述子网络模型的参数；并且

所述超网络更新单元被配置为使用所接收的所述参数来更新所述超网络模型。

21. 根据权利要求 20 所述的装置，其中所述超网络更新单元还被配置为：

基于所述子网络模型的参数的更新权重来更新所述超网络模型，所述更新权重取决于具有所述参数的所述子网络模型的数目。

22. 根据权利要求 20 所述的装置，其中所述超网络更新单元还被配置为：

确定所述超网络模型的预配置参数，以用于所述多个第一设备从所述超网络模型确定各自的子网络模型。

23. 一种用于生成神经网络模型的系统，包括：

第一设备，被配置为执行如权利要求 1 至 10 中任一项所述的方法；以及

第二设备，被配置为执行如权利要求 11 至 16 中任一项所述的方法。

24. 一种电子设备，包括：

至少一个计算单元；

至少一个存储器，所述至少一个存储器被耦合到所述至少一个计算单元并且存储用于由所述至少一个计算单元执行的指令，所述指令当由所述至少一个计算单元执行时，使所述电子设备执行根据权利要求 1 至 10 中任一项所述的方法或根据权利要求 11 至 16 中任一项所述的方法。

25. 一种计算机可读存储介质，其上存储有一条或多条计算机指令，其中一条或多条计算机指令被处理器执行使所述处理器执行根据权利要求 1 至 10 中任一项所述的方法或者根据权利要求 11 至 16 中任一项所述的方法。

26. 一种计算机程序产品，包括机器可执行指令，所述机器可执行指令在由设备执行时使所述设备执行根据权利要求 1 至 10 中任一项所述的方法或者根据权利要求 11 至 16 中任一项所述的方法。