NET4AI：6G 支持 AI 即服务

李项[1]，张航[1]，彭程晖[2]，刘哲[2]，王飞[2]

[1] 渥太华无线先进系统能力中心

[2] 无线技术实验室

摘要

本文采用系统性的方法来解决不断涌现的隐私保护深度学习问题，不仅泛化了分割学习（Split Learning），还将分割学习与联邦学习（Federated Learning）相结合，形成两级学习框架。该框架继承了分割学习与联邦学习的优点，同时摒弃了二者的缺点，支持通过分割层选择来定制学习方法。此外，本文还提出一个名为NET4AI的6G系统架构，以支持两级学习框架（如学习方法定制），并提供k匿名和数据机密性保护。NET4AI并不仅仅局限于隐私保护深度学习，相反，它将打造成一种通用架构，支持6G时代任何面向计算的服务与人工智能（Artificial Intelligence，AI）应用。利用6G无处不在的边缘计算能力，NET4AI为基于网络的AI提供从部署到运营的端到端解决方案。

关键词

隐私，AI，6G，边缘计算

1 引言

人工智能（Artificial Intelligence，AI）有时也称为机器智能，是机器所展示的智能。AI不仅可以感知外部数据，还能采取行动实现目标。现在，商业分析以及决策过程中的很多挑战都需要依赖AI技术来解决。未来，AI应用会延伸到全球经济的每一个角落，影响社会的方方面面[1]。

AI技术大致分为两类：基于规则的AI和基于学习的AI。在基于规则的AI（如专家系统）中，人类知识通过编码转化成规则，用于处理解决问题所需的输入数据。受知识库所限，基于规则的AI无法解决未知问题。而基于学习的AI则能从历史数据中学习规则，并利用这些规则实现特定目标。当前，AI研发开始回暖，基于学习的AI是其核心，其中又以机器学习类方法最为流行。

学习模型是机器学习算法的核心，它定义了输入数据与输出规则之间的关系。典型的学习模型包括人工神经网络（Artificial Neural Network，ANN）、遗传算法、回归分析等等（此处就不一一列举了）。本文将重点阐述基于ANN的机器学习，特别是深度学习，此类学习可以从训练数据中提取特征，并识别出那些与目标问题匹配的特征。基于ANN的机器学习适用于相关性数据，在各式应用中都能看到它的身影[2]。此外，本文还会交替使用“学习模型”和“AI模型”这两个概念。

深度学习中的ANN在输入层和输出层之间存在多个隐藏层，所以通常又称为“深度神经网络”（Deep Neural Network，DNN）。图1中的DNN含有三个隐藏层。深度学习需要频繁的数据访问和密集的计算来训练DNN，为缩短训练时间，会采用分布式系统将耗时的计算、缓慢的I/O并行化[3]。

在机器学习蓬勃发展的同时，移动计算也迎来了一个全新的时代，对于众多用户和应用而言，智能手机、平板电脑等个人终端逐渐成为主要的计算平台。这些终端所访问的数据量空前巨大，其中很多是个人数据，甚至属于隐私数据。

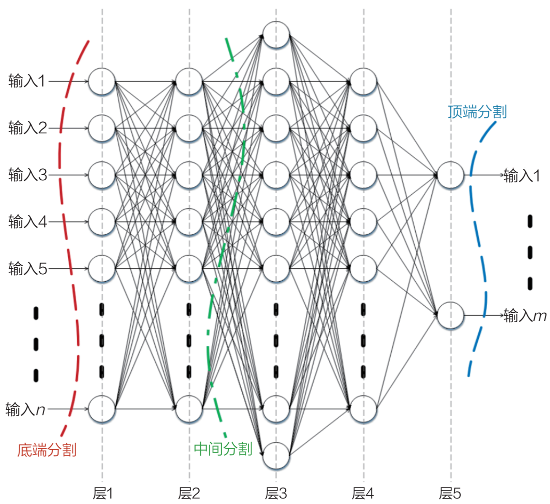

图 1 含有三个隐藏层（层 2 ~层 4）的 DNN

深度学习与移动计算的奇妙结合，产生了一种新的范式——基于分散数据的隐私保护深度学习[4]。收集、存储此类敏感数据会带来隐私风险，并需承担相应的隐私保护责任。差分隐私[5]近来已成为深度学习的重点研究对象，其目的是保护单个终端的训练数据，直至无法辨别。

训练后的学习模型可用于推理。模型推理可以在不同位置进行，取决于模型的分布。如果模型分布在客户端上，则在本地进行推理，本地推理可能会给客户端带来很大的计算工作量。如果模型分布在服务器上，客户端需要向服务器上传推理数据，存在信息泄漏的风险。研究[6]建议在客户端和服务器之间分割模型，通过这样的方式，客户端只需要向服务器发送中间结果，无需发送原始数据。由于服务器无法根据中间结果得出原始推理数据相关的信息，这种分割推理不仅能减少通信开销和时延，还能保护数据隐私。这与分割学习类似，详见1.1节。

在技术上，与模型训练中的前向传播步骤类似，模型推理由推理数据项而不是训练数据项触发，会从输出层返回分类结果，不会触发损失函数评估和反向传播。由于这种技术的相似性，本文将重点讨论模型训练，虽然我们的方案也适用于模型推理。下面简要回顾下隐私保护深度学习方面的一些工作。

1.1 业界研究进展

在深度学习方面，差分隐私方法包括：（1）在不影响统计性质的前提下为训练数据加噪，使训练后的模型仍能捕捉到原始数据集的特征[7]；（2）应用密码技术，只学习加密数据而不解密[8]。

业界提出一种替代方案，即客户端只转发随机信息，而不发送原始训练数据。在该替代方案中，以联邦学习[9]和分割学习[10]最为典型，二者都会训练深度学习模型（如DNN），但原始训练数据无需移出客户端（如上传到训练服务器）。

在联邦学习[9]中，每个客户端仅使用各自的数据集训练本地模型，然后将模型参数更新到训练服务器上，由训练服务器维护一个全局模型（特别是全局模型参数）。训练服务器汇总各个客户端上报的更新，以此调整全局模型，然后将全局模型参数返回给客户端。收到参数后，客户端再更新本地模型并继续训练。该过程不断重复，直至全局模型收敛。联邦学习可以视为随机梯度下降（Stochastic Gradient Descent）[11]的广义实现，其特点是批处理大小和参与客户端都很灵活。

在分割学习[10]中，分割层将DNN分成两个分离组件。下层组件包括输入层，在客户端侧运行，而其余部分作为上层组件，在服务器侧运行。理论上，可以自由分割，只要能产生

两个分离分区即可。但实际上，分割通常发生在DNN的两层之间（如图1的层2和层3）。因此，这两个组件可以视为两个级联的学习模型——客户端侧模型和服务器侧模型——客户端模型的输出就是服务器侧模型的输入。客户端（如终端）以迭代的方式向训练服务器发送中间结果（如客户端模型的输出），并从服务器接收相应的梯度，按顺序与训练服务器交互，使用本地数据训练DNN。当一台客户端完成训练时，该客户端将最新的模型参数发给下一台客户端，下一台客户端再使用自己的数据集继续训练。循环往复，直至所有客户端都完成训练，然后再根据需要启动新一轮的训练。

联邦学习将单点同时训练的一个个本地模型汇聚成一个全局模型。本地模型仅包含本地数据，而本地数据由于非独立同分布（Independent and Identically Distributed，IID）的原因，联邦学习收敛缓慢。而分割学习直接使用所有本地数据集来训练全局模型，因此可以实现快速收敛——这里说的是时长，训练轮数可能并非如此。但是，由于分割学习本质上属于顺序学习，需要客户端之间保持同步。关于联邦学习和分割学习的详细比较，请参见[4]。对于联邦学习和分割学习，客户端都不会向训练服务器发送原始训练数据，二者均能提供差分隐私。

研究[12]表明，内部对手对学习模型了如指掌，可以利用模型收敛的渐进过程，构建与训练数据极为相似的信息。在联邦学习中，信息可能会泄露给恶意客户端，但并不会侵犯差分隐私。相比之下，分割学习则不存在该问题，因为没有哪个客户端完全了解其深度学习模型。但是，如果学习过程仅涉及少量客户端，仅在这些客户端的本地数据中验证信息相似性，则可能会出现严重的信息泄漏。因此需要保证参与客户端的最小数量k（k为系统参数），称为k匿名，通过这种方式进一步保护隐私[13]。

[14]提出了一种安全聚合协议，该协议秉承“秘密共享”的理念，在终端与服务器之间运行，助力联邦学习实现k匿名。考虑到k匿名依赖服务器的参与，该协议可能无法缓解终端隐私问题，特别是在终端不信任服务器的情况下。用户之所以担忧隐私，是因为担心权力过度集中：服务器不仅是学习的主要实体（它拥有学习模型并监督整个学习过程），而且还知道哪些终端对学习做出了贡献。针对这个问题，本文提出一个系统性的方法来缓解用户的担忧。

此方法与[15]中的“共享机器学习系统”有一定的相关性。共享机器学习系统利用专用硬件提供一个可信的执行环境，以保护数据安全和隐私。但是，这种系统并不适合电信网络等大型公共系统。

1.2 我们的研究成果

在训练数据源为移动终端设备的场景下，基于分散数据的隐私保护深度学习与电信网络紧密耦合。6G无线系统不仅能开通连接，更重要的是实现了连接智能，即分布式学习与推理，这要求6G无线系统原生支持这种新的AI计算范式。

本文首先通过扩展分割层定义来泛化分割学习[10]，将联邦学习[9]、集中学习（Centralized Learning）视为分割学习的两种特殊情况。然后将分割学习和联邦学习结合起来，打造出一个两级学习框架。该框架继承了分割学习与联邦学习的优点，同时摒弃了二者的缺点。我们建议为AI模型选择合适的分割，从而在两级框架的底层定制学习方法。分割层的选择需考虑诸多因素，以更好地平衡终端、服务器和网络的学习开销。正是由于这一优化，本地学习（客户端侧）、集中学习（服务器侧）、分割学习（客户端侧以及服务器侧）得以同时出现在两级学习框架的底层。

我们还提出一个名为NET4AI的6G无线系统架构。该架构采用面向服务的设计，支持两级学习框架，并提供“学习方法定制”这一增值服务。借助NET4AI架构，系统可以为基于网络的AI应用提供从部署到运营的端到端支持。在运营阶段，系统协调AI应用和用户设备（如UE）的AI计算模块进行匿名通信，完成AI计算（如模型训练、模型推理）。系统还可以进一步强制k匿名，保护用户隐私，并应用代理重加密，确保数据机密性。

后文信息组织大体如下：第2节讲解一些重要概念与假设，第3节介绍两级学习框架和NET4AI架构，第4节中分析NET4AI面临的种种挑战，最后在第5节给出结论。

2 术语与假设

本节提出一些有关网络基础设施的假设，并介绍“无线计算节点”（Radio Computing Node）这个概念——“无线计算节点”指的是具有边缘计算能力的无线接入网（Radio Access Network，RAN）节点。此外，还涉及一些AI计算相关的概念，包括作业（Job）、任务（Task）、例程（Routine）等，用于解释AI计算服务的组网逻辑。

2.1 无处不在的边缘云

算法、数据、算力是AI创新的主要驱动力。按照以前的惯例，AI算力通常由集中式的云平台提供，数据需推送到中心云进行处理。在大数据和AI时代，这种做法会引发数据隐私、时延、效率等一系列问题，由此诞生“让计算走进数据”这一完全相反的范式转变。

边缘计算（Edge Computing）将计算和存储下沉到网络边缘附近，从而离最终用户和数据更近，这样可以缩短传输时延、降低带宽消耗。随着边缘计算技术日益成熟，边缘计算

能力预计会以边缘云的形式在网络中广泛部署，比如与RAN节点或基站合设。具有边缘计算能力的RAN节点或基站，就是本文所说的“无线计算节点”。

由于无线网络基础设施同时提供计算资源（如CPU周期、内存、存储、I/O访问）和通信资源（包括无线资源和传输资源），可以联合优化分布式多样化基础设施资源的利用率，为最终用户提供极致的端到端性能。而基础设施资源管理则由网络中的一个或多个实体（如资源管理器）来执行。

2.2 计算相关概念

AI计算服务可以是应用层服务，也可以是优化网络管理或运营的网络服务。

AI计算服务包含一个或多个计算模块，称为“业务功能”。AI计算服务与明确定义的计算逻辑相关联，根据输入数据下发计算结果。计算逻辑包括每一个业务功能的算法实现以及业务功能之间的交互，后者可以通过作业、任务和例程指定，这点后文会有提及。

AI计算服务包含一个或多个独立的作业，每个作业专注于一个特定的计算目标，并向服务的消费者公开。执行可以在作业级别触发，触发条件包括用户请求、特定事件或特定状况。当作业处于执行状态时，服务消费者可以通过提供输入数据、接收计算结果等方式来参与或贡献。

作业由一个或多个相互依赖的任务组成，在作业执行期间，任务会根据任务间的依赖关系依次执行。如果任务A依赖任务B，那么就必须先执行B，再执行A。每个任务都与一个业务功能链（Service Function Chain）相关联，每个业务功能链由一个或多个例程组成，每个例程对应业务功能链中的两个相邻的业务功能（例程服务器功能和例程客户端功能）。在每个例程内，例程客户端功能可以是终端，也就是说该功能的计算逻辑可以运行在终端上。但是，例程服务器功能不可以是终端。

在任务执行过程中，其例程沿业务功能链依次执行。例程一旦执行，就会触发对应的客户端功能和服务器功能，二者相互通信。图2展示了服务、作业、任务、例程和业务功能之间的关系，箭头表示依赖关系。这里我们以AI计算服务或应用为例进行讲解。在AI计算服务中，模型训练、模型推理可以是两个独立的作业。（模型）训练作业包括一个模型训练任务及其依赖的模型验证任务，模型训练任务可与包含三个功能的业务功能链关联（图3），任务中的两个例程分别对应3.1节介绍的两级学习框架中的底层学习和顶层学习。

业务功能（无论是例程客户端功能还是例程服务器功能），如果不表示终端，则视为“具体业务功能”；而表示终端的业务功能则视为“抽象业务功能”。如果网络部署了AI计算服务，该服务的具体业务功能会在边缘云等网络位置实例化，然后业务功能的实例运行在所处网络位置的应用服务器（Application Server，AS）上。在给定的例程中，例程客户端功能的实例称为“例程客户端”，例程服务器功能的实例称为“例程服务器”。如果例程客户端功能表示终端，这些终端就是“例程客户端”。

3 AI应用原生支持

本节将讨论如何为6G系统中的AI应用提供原生支持，特别是在隐私保护深度学习方面。本节将描述一个支持学习方法定制的两级学习框架，以及面向服务的系统架构，为AI计算服务提供端到端的解决方案。本节同时也会探讨无线计算节点（同时提供无线与计算资源的RAN节点）的协议设计，并通过案例证明方案的有效性。

3.1 学习方法定制

首先，通过扩展分割层定义来泛化分割学习[10]，使联邦学习[9]、集中学习成为分割学习的两种特殊情况。在联邦学习中，所有终端都对AI模型了如指掌，使用本地数据集训练AI模型。联邦学习可以看成是应用了顶端分割的分割学习，分割层位于输出层上方。而集中学习要求终端向服务器发送原始训练数据，学习是纯服务器侧的行为，因此，集中学习可以看成

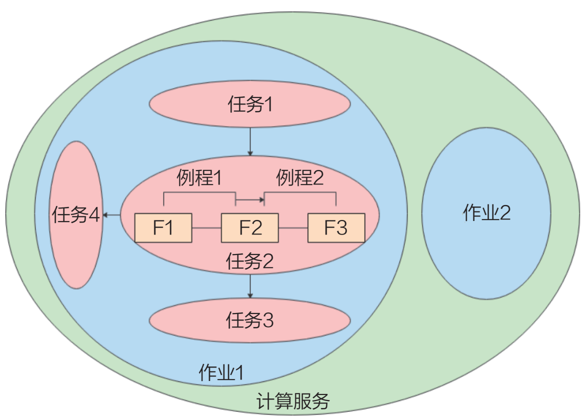

图 2 服务、作业、任务、例程和业务功能（F）

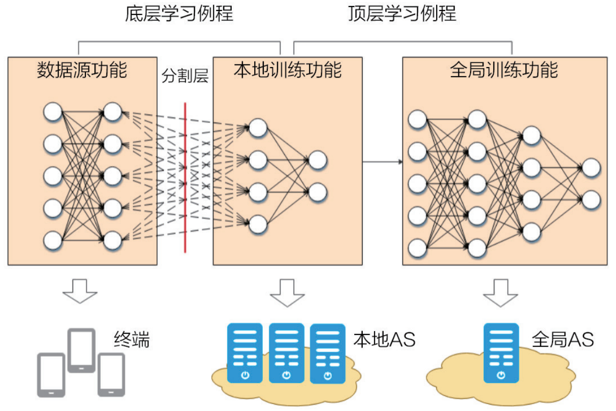

图 3 两级学习框架

是应用了底端分割的分割学习，分割层位于输入层下方。传统的分割学习[9]对应的是用中间分割来划分的AI模型。底端分割、中间分割和顶端分割如图1所示。

然后，联邦学习和泛化的分割学习可以结合起来，组成一个通用的两级学习框架。如图3所示，该框架可以用包含两个例程的任务来表示，一个底层学习例程、一个顶层学习例程。泛化的分割学习可用在框架的底层（底层学习例程），而联邦学习则用在顶层（顶层学习例程）。底层学习例程在两个业务功能之间运行：数据源功能和本地训练功能。数据源功能表示终端，本地训练功能在本地AS上实例化，以便训练本地AI模型。例如，这些本地AS可能设在无线计算节点上。顶层学习例程在本地训练功能和全局训练功能之间运行，后者可以在全局AS上实例化。为了更好地服务多个本地AS（无线计算节点），全局AS所处的边缘云可能距RAN相对较远。在这个两级学习框架中，学习方法定制是通过底层的分割层选择来实现的。

如果底层学习（例程）选择了中间分割，两级学习框架会兼具联邦学习与分割学习的优势。底层学习使用的是所有终端数据集的全集，相对单终端的数据集来说，全集的非IID特征明显弱化。因此，在本地AS上训练的本地AI模型会比联邦学习更准确——在联邦学习中，每个终端只使用自身的数据集训练模型。一般来说，本地模型精度越高，全局模型收敛越快。由于终端不完全了解AI模型，训练数据中的信息不会像[13]所述的泄露给对手终端。

如果底层选择了顶端分割，本地AI模型则由单个终端来训练，本地训练功能在收到本地AI模型参数后会进行聚合，再发给全局训练功能。在这种情况下，框架仅包含联邦学习，有信息泄漏的风险。如果底层学习应用了底端分割，框架则不提供差分隐私。出于隐私保护考虑，一般不建议使用顶端分割和底端分割，但也有一些例外情况，比如下文提到的几个因素。

在底层学习中，如果多个终端与本地训练功能的同一个实例相关联，它们应该分配同一个分割层，这样才能在学习期间表现出和本地训练功能实例一致的行为。在此约束下，综合考虑模型结构、终端状态（如算力、能级）、服务器条件（如AS加载）、终端位置、本地训练功能的部署位置、网络条件（如带宽、拥塞等）等因素，底层学习可以应用混合分割，同时优化终端性能、服务器性能以及网络性能。应用混合分割时，可以为不同的终端组选择不同的分割层，每个组关联一个不同的本地训练功能实例，如图4所示。

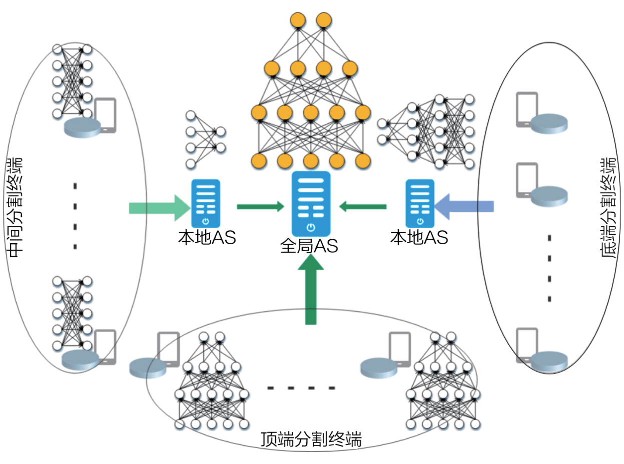

图 4 模型训练应用的混合分割

3.2 NET4AI 系统架构

围绕作业、任务、例程等概念，我们提出6G无线系统架构NET4AI（见图5），以支持上文所述的两级学习框架。采用NET4AI架构的6G无线系统称为NET4AI系统，此类系统可以为客户提供NET4AI服务，并为AI计算服务提供端到端的支持。NET4AI架构假设业务功能可以部署或运行在终端或边缘云上。

NET4AI架构包含控制平面（Control Plane）和计算平面（Compute Plane）。控制平面除了提供传统的终端管理功能外，还会管理AI计算服务，并控制这些服务的作业执行和任务执行。控制平面有许多实体，包括业务管理器、编排器、资源管理器、接入管理器、作业管理器、任务管理器等。在具体实现中，有些控制平面实体可能会合并，例如，将作业管理器合并到业务管理器中；再如，控制平面既可以同时涵盖RAN网络和核心网段，也可以只涵盖RAN网络或核心网段。计算平面控制AI计算服务例程的执行，支持业务功能之间进行数据通信。计算平面可能包含例程管理器以及转发子平面（转发子平面也可简单称为“转发平面”，Forwarding Plane）。转发平面对应3GPP 5G系统架构[17]中的用户平面（User Plane），它包含转发平面功能（Forwarding Plane Function，FPF）和RAN节点（准确地说，是RAN节点的用

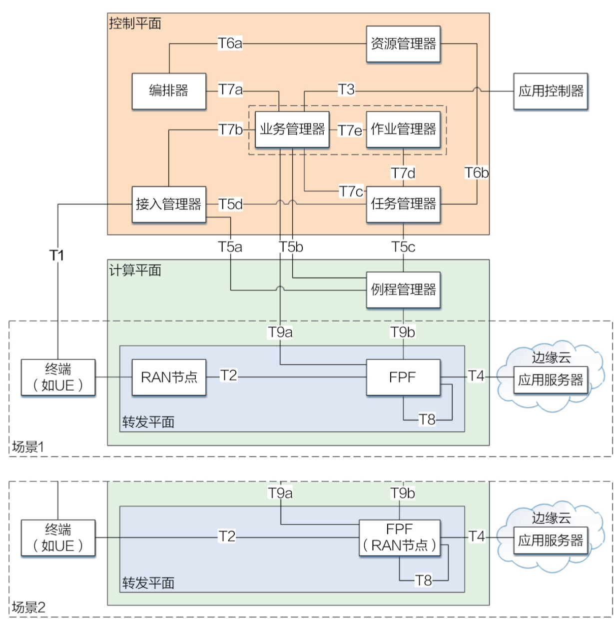

图 5 NET4AI 架构图

户平面部分），而FPF可以与RAN节点集成。

图5在各实体通信接口之间建立了链路，支持以服务为粒度来定义或创建链路。这里需要重点介绍下T2接口，当FPF独立于RAN节点时（图5场景1），T2接口连接RAN节点，映射到无线承载。当FPF与RAN节点集成时（图5场景2），RAN节点实现FPF的功能，此时，T2接口连接到终端设备，由无线承载实现。无论哪种情况，多个终端都可以共享无线承载，如3.3节介绍的计算无线承载（Computing Radio Bearer，CRB）。在场景2中，如果RAN节点与边缘云集成（例如，RAN节点用作无线计算节点），那么RAN节点内部需要支持T4接口。

通过授权的应用控制器，可以在NET4AI系统中注册AI计算服务，应用控制器归属服务提供商，负责管理AI计算服务。注册就是一个将计算服务实例化的过程：在一个或多个网络位置（如边缘云）将计算服务的具体业务功能实例化。注册完成后，NET4AI系统通过协同执行AI计算服务的例程、任务和作业，来控制AI计算服务的操作。下面详细介绍服务实例化和服务操作。

3.2.1 服务实例化

在NET4AI系统注册AI计算服务时，应用控制器向业务管理器发送该服务的业务功能、例程、任务和作业的描述信息，而后业务管理器与编排器交互，将AI计算服务实例化。在此过程中，由编排器来决定AI计算服务的部署，并选择合适的计算平面。

部署决策的内容包括：（1）具体业务功能实例的位置以及每个业务功能实例所需的资源，这些业务功能实例包括例程服务器，也可能包括AI计算服务的例程客户端；（2）每个例程的例程客户端和例程服务器之间的逻辑链路，通过逻辑链路，例程客户端和例程服务器可以在例程执行期间通过计算平面与对方通信。

例程客户端通过T2或T4接口连接到计算平面的FPF，例程服务器也通过T4接口连接到计算平面的FPF，这两个FPF可以是相同的实体，也可以是通过T8接口互连的不同实体。在计算平面，例程客户端、例程服务器会分配给同一个例程管理器。通过触发例程客户端、例程服务器之间的数据通信，例程管理器管理该例程的执行。

为实现部署决策，编排器需要利用资源管理器，并将计算平面选择结果通知给业务管理器，业务管理器再对计算平面进行相应的配置。

3.2.2 服务操作

作业——AI计算服务的一个重要组成部分——由任务构成，而任务则利用两级学习框架（详见3.1节）来训练AI模型。此类任务称为“模型训练任务”，涉及底层学习例程和顶层学习例程，如图3所示。模型训练任务涉及的业务功能有：数据源功能、本地训练功能、全局训练功能。数据源功能是底层学习例程的例程客户端功能，它表示终端，是训练数据的来源。本地训练功能是底层学习例程的例程服务器功能。接下来会详细介绍NET4AI系统是如何支持与作业相关的服务操作的。

· 控制平面行为

应用控制器向业务管理器发送作业指令，请求执行作业。收到指令后，业务管理器通知选定的作业管理器去执行作业。

如前所述，业务管理器和作业管理器在某些实现中可以合二为一。

在执行作业时，作业管理器为作业中的每个任务（包括模型训练任务）选择一个任务管理器，并触发任务管理器根据任务间的依赖关系去执行任务。执行任务时，任务管理器会识别出计算平面中的相关例程管理器，对应任务中的例程。任务管理器请求例程管理器按例程之间的依赖关系执行例程。

如果某例程的例程客户端功能（如数据源功能）表示终端，每个终端将作为例程客户端分配给例程管理器。这些终端需满足以下条件：允许接入AI计算服务、已注册且已达成共识（贡献作业）。终端可以注册到NET4AI系统，通过接入管理器提供自己的共识——接入管理器的功能类似于3GPP 5G系统架构中的接入和移动性管理功能（Access and Mobility Management Function，AMF）[17]。

在执行模型训练任务时，为定制AI模型的学习方法，相应的任务管理器会为底层学习例程选择分割层（详见3.1节）——选择分割层可以与上述例程客户端选择同步进行——并将分割层选择结果通知给例程管理器。

· 计算平面行为

NET4AI的计算平面深度参与了AI计算服务中每个例程的执行，负责协调例程客户端、例程服务器参与例程的执行，强制*k*匿名，并在例程客户端、例程服务器之间启用匿名数据通信。

针对每个例程，计算平面都会分配相应的例程客户端、例程服务器给例程管理器，每个例程服务器又可以与一个或多个例程客户端关联。在大多数情况下，这种服务器–客户端关联都是由任务管理器和/或例程管理器动态确定的。但如果例程客户端功能是一个具体业务功能（见3.2.1节），则由编排器在服务实例化期间确定。服务器–客户端关联在应用层不可知，应用层包括例程客户端、例程服务器、应用控制器。

在执行例程时，例程管理器会触发或邀请关联的例程客户端与例程服务器进行数据通信。这种通信是互匿名的，双方均不知道对方的身份。例程管理器首先通过转发平面邀请例程服务器（在这一步中，例程管理器会根据实际情况向例程服务器提供分割层信息），然后再通过接入管理器（如例程客户端本身就是终端时）或转发平面邀请例程客户端。

在通信过程中，数据流量通过转发平面的路由，在例程客户端与例程服务器之间传输。通信完成后，例程客户端或例程服务器会通知例程管理器，这样例程管理器就可以继续进行下一步操作，如邀请下一个例程客户端参与通信，或通知任务管理器例程执行已完成。

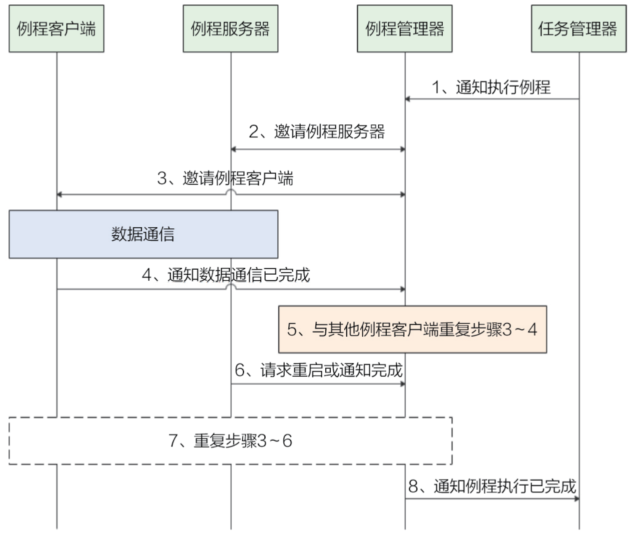

图 6 例程服务器的例程执行过程

图6展示了上述例程执行过程。在步骤6中，例程服务器可以请求重启数据通信，也可以发送“完成”通知。如果发送重启请求，例程管理器需要重复执行步骤3 ~ 6。等所有例程服务器都发送了“完成”通知，例程管理器通知任务管理器例程执行已完成（步骤8）。

在这个流程中，步骤5可以与步骤3、4同步执行，除非例程要求顺序通信。假如底层学习例程选择了中间分割，则可能需要顺序通信。在这种情况下，如果一台例程服务器关联了多个例程客户端，例程管理器一次只邀请一个客户端与例程服务器通信。在邀请例程客户端时，例程管理器向该客户端提供分割层信息。

在数据通信过程中，如果例程客户端本身就是终端，会对用户隐私产生极高的要求。为满足用户隐私要求，可以在计算平面中强制实施*k*匿名，其中*k*值取决于系统参数或应用控制器的要求。假设在服务实例化期间，例程管理器又分配到了$m-1$个例程服务器，那么，任务管理器将分配至少$m*k$个例程客户端给例程管理器，然后例程管理器将至少*k*个例程客户端关联到每个例程服务器。

数据通信过程中可能涉及数据保密性要求。鉴于通信双方互不可知，端到端加密在此场景并不适用。这就需要转发平面（其实是FPF）的代理重加密机制来保障数据机密性：来自发送方的数据是密文，转发平面不对密文进行解密，直接转发。

在转发前，会利用重加密密钥，重新加密数据，使接收方能够使用自己的私钥恢复原始数据（明文）。在步骤2或步骤3中，例程管理器会获取重加密密钥，并在数据通信开始之前配置到转发平面。

3.3 无线计算节点的计算平面协议

RAN节点的无线接口包括三个协议层：层一是物理层，层二是数据链路层，层三是网络层。NET4AI架构侧重于层二和层三的协议。

作为层二的逻辑信道，无线承载是层一、层三连接的桥梁，支撑用户或控制数据的传输。传统无线承载管理假设RAN节点是网络接入点，且计算发生在网络的另一端。而到了无线计算节点，通信和计算一体化，计算在网络内进行，传统的管理方法可能就无法满足需要或无法高效实施。因此，NET4AI计算平面在层二引入CRB，帮助无线计算节点区分计算平面以及用户平面的数据，从而在无线计算节点实现自环计算服务。CRB一端连接无线计算节点所服务的UE，另一端则连接无线计算节点上部署的一个或多个业务功能，从这个意义上讲，它提供了图5中的T2接口功能。深度边缘协议（Deep Edge

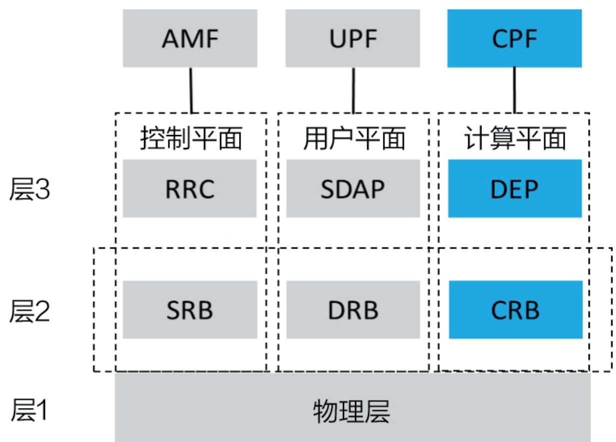

图 7 CRB 和 DEP 示意图

Protocol，DEP）是针对层三提出的一种新型极简协议，目的是帮助UE和业务功能高效地交换数据。之所以要引入DEP协议，一个最主要的原因是要在无线网络（如RAN）中部署业务功能。因此，与传统的云部署方式相比，数据传输协议可以进一步简化。

图7展示了CRB和DEP，以及5G的层二无线承载和层三协议。图中，AMF和UPF（User Plane Function）属于5G网络功能，在NET4AI架构中，它们分别提供了接入管理器和FPF的部分功能；而CPF（Compute Plane Function）是在无线计算节点中实现FPF功能的模块。这些无线承载和协议可以共用无线计算节点，支持不同的场景或需求。例如，通过数据无线承载（Data Radio Bearer，DRB）和业务数据适配协议（Service Data Adaptation Protocol，SDAP），UE可连接到无线计算节点以外的业务功能。

假设一个无线计算节点正在为UE提供服务，该节点部署了一个业务功能，当UE尝试接入此功能做计算时，无线计算节点就会为它分配一个CRB。通过这个CRB，UE就能连上业务功能。基本流程如下：UE通过无线计算节点向NET4AI系统的

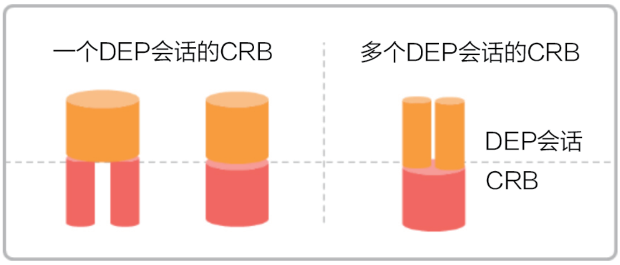

图 8 CRB 与 DEP 会话的映射关系

控制平面发送计算请求。在通知UE请求已受理的同时，控制平面也会通知无线计算节点为该UE建立DEP会话。然后，无线计算节点（控制面部分）将相应的DEP会话分配给UE，该会话会映射到一个已有的或新建的CRB。无线计算节点通过无线资源控制（Radio Resource Control，RRC）信令将CRB参数下发到UE，这样UE就可以开始与业务功能交换数据了。

如图8所示，每个DEP会话可以映射到一个或多个CRB，每个CRB也可以支持一个或多个DEP会话。不仅如此，每个CRB还可以配置特定的QoS能力，若一个DEP会话映射到多个CRB，而这些CRB的QoS能力都各不相同，DEP会话就可以通过映射不同的CRB支持多条QoS流。相应地，若只映射了一个CRB，则DEP会话只支持一条QoS流。

3.4 案例研究

某医疗机构正在研制一款血压模型，这款基于DNN的AI模型会在特定的地理区域内，收集不同年龄层人群在一天中不同时间的血压情况。该医疗机构想向大众推广此模型，以便更好地训练模型。为此，他们在NET4AI系统中注册了自己的AI计算服务。

在注册AI计算服务的过程中，该医疗机构提供了AI模型的相关信息，包括层数、每层的神经元数、相邻层之间的链路数等。AI计算服务包含两个具体业务功能：本地训练功能和全局训练功能。本地训练功能支持AI模型的集中学习和分割学习，全局训练功能实现联邦学习的模型聚合逻辑。

AI计算服务包含模型构建作业，模型构建作业又包含模型

训练任务。如图3所示，模型训练任务包含底层学习例程和顶层学习例程。底层学习例程与本地训练功能以及代表终端的数据源功能关联，顶层学习例程与本地训练功能以及全局训练功能关联。此外，模型构建作业还可能包含模型验证任务，模型验证任务在模型训练任务完成后方可执行。限于篇幅，模型验证任务在此不作细述。

根据该医疗机构提供的信息，NET4AI系统在网络（如边缘云）中部署AI计算服务，包括一个全局训练功能实例和多个本地训练功能实例。每个实例都通过一个附着点连接NET4AI系统，该附着点可以是FPF，也可以是无线计算节点（如业务功能实例部署在无线计算节点上）。

待AI计算服务部署完成后，该医疗机构请求NET4AI系统执行模型构建作业。收到请求后，NET4AI系统开始执行模型构建作业，通知本地训练功能实例去执行底层学习例程。例程要求终端参与，因此在终端参与之前，不会执行作业。

连网后，NET4AI系统会将AI计算服务的模型构建作业通知终端。终端可以自愿参与模型构建作业，向NET4AI系统发送共识，并告知自身的状态（如算力、能级）和隐私要求。根据隐私要求，NET4AI系统就可以判断出终端是否愿意为模型构建作业提供原始数据。

根据终端反馈的信息以及其他信息（如网络条件），NET4AI系统为达成共识的终端分组。对应底层学习例程，NET4AI系统给每个终端组选择一个分割层，并将终端组与业务功能实例关联。该业务功能实例既可以是本地训练功能的实例，也可以是全局训练功能的实例。以图4为例，NET4AI系统为图中的三个组（每个椭圆代表一个组）分别选择底端分割、中间分割和顶端分割，前两个组分别关联到本地训练功能的两个实例，而第三个组则关联全局训练功能的实例。

NET4AI系统将分割层选择结果通知给每个组内的终端，并将终端连接到相应业务功能实例的附着点。分配到顶端分割或中间分割的终端可以从NET4AI系统（配置了分割层）获取本地训练功能，并在本地运行。

如果有足够多的可用终端（每组至少k个），NET4AI系统会邀请它们执行底层例程。然后，终端可以通过NET4AI系统将数据发送到相应的本地训练功能实例。在数据通信过程中，终端和本地训练功能实例互不可知。对于分配到底端分割的终端来说，它发送的数据包括血压读数、用户年龄等；对于分配到中间分割的终端来说，它发送的数据就是分割学习中所说的“中间结果”；而对于分配到顶端分割的终端来说，它发送的数据则是跟联邦学习一样的本地模型参数。

在终端关联的本地训练功能实例完成计算（即建立了本地AI模型）后，NET4AI系统就知道底层学习例程的执行已经完成，然后通知本地训练功能实例和全局训练功能实例去执行顶层学习例程。模型构建作业会以这种方式反复执行，直至全局训练功能实例通知NET4AI系统停止（例如，当全局模型收敛时）。

全局训练功能实例通过NET4AI系统向该医疗机构提供模型参数，据此，该医疗机构判断血压模型的训练已成功完成。

4 技术挑战

在“普惠智能”这一愿景的驱动下，6G网络需要考虑原生AI设计，而不只是在架构层面简单地叠加AI。因此6G网络不仅面临传统的连接问题，还需解决一些新的技术挑战，包括数据隐私、资源异构、节能等。无线边缘环境的复杂性也是尤为紧迫的问题，这种复杂性是无线连接的不稳定性、大规模分散性和边缘资源的异构性等种种因素共同作用的结果。下面列举几个颇具研究价值的难题：

· 能源效率

NET4AI可能需要支持6G网络中的大规模分布式训练。随着模型和参数的同步，数据通信越来越频繁，再加上日益增加的计算成本，能耗挑战愈加严峻。

考虑到训练过程，通常有两种方法可以减少通信开销。第一种是减少每轮通信交换的数据量，比如模型压缩法（如量化、稀疏化、知识蒸馏）[18-21]。第二种是减少通信轮数。例如，FedAvg在聚合前会执行多轮本地更新[9]。如何才能既兼顾AI训练性能，又降低通信开销，还有待进一步研究。

另一种提高通信效率的办法是优化通信网络拓扑结构设计[22]，例如，高性能计算（High-Performance Computing）中的Ring Allreduce方案可以降低通信带宽。然而，无线网络的拓扑没有物联网服务器那么灵活，实际上，是否能将这样成熟的高性能计算技术应用到无线网络，还需要大量的研究。

· 适应动态环境

分布式学习系统可能会考虑某些容错机制，如经典的拜占庭容错机制[23]。然而，要在无线网络运用这些机制是个不小的挑战。其中一个原因是，在带有连接的无线网络环境中，资源是动态变化的。比如，极端的流量突发可能会阻塞基站上的AI训练任务；无线连接不稳定也有可能导致终端突然掉线，从而阻塞任务。想要忽略这些故障节点[24-25]或是执行冗余备份[26]，则需要研究如何原生适应动态的无线环境。

· 异构资源调度

为了实现NET4AI，需要管理、调度6G网络中的海量异构资源，但要设计出高效的分布式调度框架和算法绝非易事。

为了正确且高效地将AI任务部署到无线网络中，首先必须对各种异构资源进行统一建模（包括算力、内存、存储及通信带宽），再定义资源调度的状态和动作空间，才能设计出分布式调度框架和算法。调度算法需要考虑一个NP问题：如何在海量异构资源之间高效地分配任务。此外，计算复杂度会随着扩容而大幅增加。

· 数据业务

为促进网络智能、数据共享，提高网络运营效率，6G网络可能会产生、处理、消耗大量数据（如感知数据）。这给6G数据业务带来了挑战：如何在确保数据安全的前提下充分挖掘数据价值，并在架构上原生遵守欧盟和欧洲经济区（European Economic Area，EEA）的通用数据保护条例（General Data Protection Regulation，GDPR）以及其他数据法规。

NET4AI需要确保用户对自己的个人数据有绝对的控制权，保证用户有权决定是否共享数据，利用数据变现，或将数据用于训练。单靠一些独立的数据保护方案，如*k*匿名、*l*多样性、*t*接近度[31]、差分隐私，可能还远远不够。6G面临的关键挑战是如何构建一套完善的、多方透明的架构级数据业务框架。

5 结语

本文提出了NET4AI这一6G系统架构，旨在支持面向未来的计算服务，特别是AI计算服务。NET4AI为AI计算服务提供从部署到运营的端到端支持，解决了隐私感知深度学习的新问题，并支持学习方法定制。通过强制*k*匿名，确保计算平面的数据机密性，提供强大的隐私保护。围绕无线计算节点（即具有边缘计算能力的RAN节点），本文还探讨了层二、层三的协议设计，引入CRB、DEP帮助无线计算节点实现高效通信、高效计算。在文章的最后，我们还识别了NET4AI所面临的技术挑战。总而言之，NET4AI架构仍处于起步阶段，关于例程、任务、作业和其他系统流程的执行，实际的实现细节（如移动性和连接管理）尚待开发。

参考文献

[1] R. Sagramsingh, "AI: A global survey," IEEE-USA, 2019.

[2] S. Dargan, M. Kumar, M.R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: A new paradigm," Archives of Computational Methods in Engineering, 2019.

[3] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J.S. Rellermeyer, "A survey on distributed machine learning," arXiv:1912.09789, 2019.

[4] P. Vepakomma, T. Swedish, R. Raskar, O. Gupta, and A. Dubey, "No peek: A survey

of private distributed deep learning," CoRR abs/1812.03288, 2018.

[5] C. Dwork, "Differential privacy," Proc. ICALP, pp. 1–12, 2006.

[6] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, and J. Mars L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," Proc. ASPLOS, 2017.

[7] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," Proc. TCC, pp. 265 – 284, 2006.

[8] M. Minelli, "Fully homomorphic encryption for machine learning," PhD thesis, PSL Research University, 2018.

[9] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," Proc. AISTATS, 2017.

[10] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," Journal of Network and Computer Applications, vol. 116, pp. 1 – 8, 2018.

[11] L. Bottou, F.E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," SIAM Review, vol. 60, no. 2, pp. 223 – 311, 2018.

[12] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," Proc. ACM CCS, pp. 603 – 618, 2017.

[13] V. Ayala-Rivera, P. McDonagh, T. Cerqueus, and L. Murphy, "A systematic comparison and evaluation of k-anonymization algorithms for practitioners," Trans. Data Privacy, vol 7, no. 3, pp. 337 – 370, 2014.

[14] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. Brendan McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," Proc. ACM CCS, pp. 1175 – 1191, 2017.

[15] Technical framework for a shared machine learning system, Recommendation ITU-T F.748.13.

[16] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," Int. J. Web and Grid Services, vol. 14, no. 4, pp.352–375, 2018.

[17] System architecture for the 5G System (5GS), 3GPP TS 23.501.

[18] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, "Federated learning: Strategies for improving communication efficiency," arXiv preprint arXiv:1610.05492, 2016b

[19] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Hierarchical quantized federated learning: Convergence analysis and system design," arXiv preprint arXiv:2103.14272, 2021.

[20] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, "Federated learning with compression: Unified analysis and sharp guarantees," in International Conference on Artificial Intelligence and Statistics, pp. 2350-2358, PMLR, 2021.

[21] Cheng Y, Wang D, Zhou P, *et al.*, "A survey of model compression and acceleration for deep neural networks," arXiv preprint arXiv:1710.09282, 2017.

[22] G. Neglia, C. Xu, D. Towsley, and G. Calbi, "Decentralized gradient methods: Does topology matter?" in Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics(S. Chiappa and R. Calandra, eds.), vol. 108 of Proceedings of Machine Learning Research, pp. 2348-2358, PMLR, 26-28 Aug 2020.

[23] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards

optimal statistical rates," in Proc. Int. Conf. Machine Learning, 2018, pp. 5650-5659.

[24] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, *et al*., "Towards federated learning at scale: System design," in Proc. Conf. Machine Learning and Systems, 2019

[25] T. Li, A. K. Sahu, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in Proc. Conf. Machine Learning and Systems, 2020.

[26] Z. Charles and D. Papailiopoulos, "Gradient coding using the stochastic block model," in Proc. Int. Symp. Information Theory, 2018, pp. 1998-2002. doi:10.1109/ISIT.2018.8437887

[27] Liu Y and Passino K M, "Swarm intelligence: Literature overview[J]," Department of electrical engineering, the Ohio State University, 2000.

[28] Xing Xu, Rongpeng Li, Zhifeng Zhao, and Honggang Zhang, "Stigmergic Independent Reinforcement Learning for Multi-Agent Collaboration," IEEE Trans. Neural Networks & Learning Systems (TNNLS), February 2021.

[29] W. Niu *et al*., "GRIM: A general, real-time deep learning inference framework for mobile devices based on fine-grained structured weight sparsity," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2021.3089687.

[30] Y. E. Sagduyu, Y. Shi, A. Fanous, and J. H. Li, "Wireless network inference and optimization: Algorithm design and implementation," in IEEE Transactions on Mobile Computing, vol. 16, no. 1, pp. 257-267, 1 Jan. 2017, doi: 10.1109/TMC.2016.2538233.

[31] Rajendran, Keerthana, Manoj Jayabalan, and Muhammad Ehsan Rana, "A study on k-anonymity, l-diversity, and t-closeness techniques," IJCSNS 17.12 (2017): 172.