(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115049040 A
(43)申请公布日 2022.09.13

(21)申请号 202210209448.9

(22)申请日 2022.03.04

(30)优先权数据
17/195,451 2021.03.08 US

(71)申请人 辉达公司
地址 美国加利福尼亚州

(72)发明人 P·莫尔恰诺夫 尹洪旭
J·M·A·洛佩兹 X·董

(74)专利代理机构 北京市磐华律师事务所 11336
专利代理师 高伟

(51)Int.Cl.
G06N 3/04(2006.01)
G06N 3/08(2006.01)

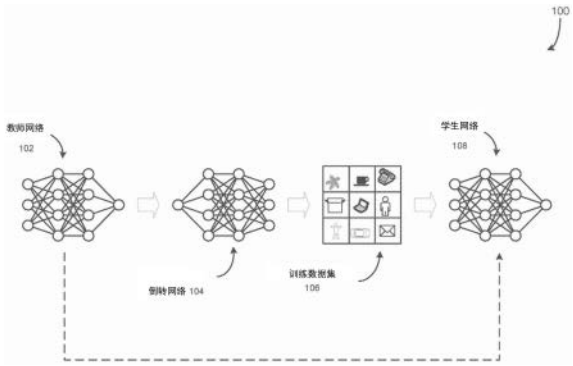
权利要求书2页 说明书112页 附图56页

(54)发明名称

神经网络训练技术

(57)摘要

公开了神经网络训练技术,具体公开了用于倒转神经网络的装置、系统和技术。在至少一个实施例中,一个或更多个神经网络层被倒转,并且在至少一个实施例中以相反的顺序被加载。



CN 115049040 A

1.一种处理器,包括:一个或更多个电路,用于使一个或更多个神经网络层以相反的顺序被加载到所述处理器中。

2.根据权利要求1所述的处理器,所述一个或更多个电路用于生成包括所述一个或更多个神经网络层的倒转神经网络,所述一个或更多个神经网络层通过将预训练的神经网络中的一个或更多个神经网络层进行倒转来生成。

3.根据权利要求1所述的处理器,其中所述一个或更多个神经网络层包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。

4.根据权利要求1所述的处理器,所述一个或更多个电路用于通过至少部分地基于预训练的神经网络的卷积层生成一个或更多个转置卷积层来对所述预训练的神经网络的一个或更多个卷积层进行倒转。

5.根据权利要求1所述的处理器,所述一个或更多个电路用于通过至少部分地基于预训练的神经网络的一个或更多个批归一化层的线性变换生成一个或更多个倒转批归一化层,来对所述预训练的神经网络的一个或更多个批归一化层进行倒转。

6.根据权利要求1所述的处理器,所述一个或更多个电路用于插入激活层。

7.根据权利要求1所述的处理器,其中包括所述一个或更多个神经网络层的倒转神经网络将输入的传真输出至预训练的神经网络。

8.根据权利要求1所述的处理器,所述一个或更多个电路用于至少部分地基于层一致性损失和重建损失来微调所述一个或更多个神经网络层。

9.一种系统,包括:

一个或更多个处理器,用于使一个或更多个神经网络层以相反的顺序被加载到所述一个或更多个处理器中。

10.根据权利要求9所述的系统,所述一个或更多个处理器用于生成包括所述一个或更多个神经网络层的倒转神经网络,所述一个或更多个神经网络层通过至少将预训练的神经网络中的一个或更多个神经网络层进行倒转来生成。

11.根据权利要求9所述的系统,其中所述一个或更多个神经网络层包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。

12.根据权利要求11所述的系统,其中所述倒转卷积层至少部分地基于预训练的神经网络的卷积层的转置来生成。

13.根据权利要求11所述的系统,其中所述倒转批归一化层至少部分地基于预训练的神经网络的批归一化层的线性变换来生成。

14.根据权利要求9所述的系统,其中激活层被添加到所述一个或更多个神经网络层中。

15.根据权利要求9所述的系统,其中包括所述一个或更多个神经网络层的倒转神经网络的输出包括对预训练的神经网络的输入的传真。

16.根据权利要求9所述的系统,所述一个或更多个处理器用于至少部分地基于层一致性损失和重建损失来微调所述一个或更多个神经网络层。

17.一种机器可读介质,具有存储在其上的一组指令,所述一组指令如果由一个或更多个处理器执行,则使所述一个或更多个处理器至少:

使一个或更多个神经网络层的输出以相反的顺序被计算。

18.根据权利要求17所述的机器可读介质,具有存储在其上的另一组指令,所述另一组指令如果由一个或更多个处理器执行,则使所述一个或更多个处理器至少:

生成包括所述一个或更多个神经网络层的倒转神经网络,所述一个或更多个神经网络层至少部分地基于预训练的神经网络的一个或更多个神经网络层来生成。

19.根据权利要求17所述的机器可读介质,其中所述一个或更多个神经网络层包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。

20.根据权利要求19所述的机器可读介质,其中所述倒转卷积层至少部分地基于预训练的神经网络的卷积层的转置来生成。

21.根据权利要求19所述的机器可读介质,其中所述倒转批归一化层至少部分地基于预训练的神经网络的批归一化层的线性变换来生成。

22.根据权利要求17所述的机器可读介质,具有存储在其上的另一组指令,所述另一组指令如果由一个或更多个处理器执行,则使所述一个或更多个处理器至少将激活层添加到所述一个或更多个神经网络层。

23.根据权利要求17所述的机器可读介质,其中包括所述一个或更多个神经网络层的倒转神经网络的输出包括对预训练的神经网络的输入的传真。

24.根据权利要求17所述的机器可读介质,具有存储在其上的另一组指令,所述另一组指令如果由一个或更多个处理器执行,则使所述一个或更多个处理器至少部分地基于层一致性损失和重建损失来至少微调所述一个或更多个神经网络层。

25.一种方法,包括:

至少部分地基于第一神经网络生成倒转神经网络;

使用所述倒转神经网络来生成训练数据;以及

使用所生成的训练数据来训练第二神经网络。

26.根据权利要求25所述的方法,其中所述倒转神经网络相对于所述第一神经网络具有反向计算流。

27.根据权利要求25所述的方法,其中所述倒转神经网络包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。

28.根据权利要求27所述的方法,其中所述倒转卷积层至少部分地基于预训练的神经网络的卷积层的转置来生成。

29.根据权利要求27所述的方法,其中所述倒转批归一化层至少部分地基于预训练的神经网络的批归一化层的线性变换来生成。

30.根据权利要求25所述的方法,还包括:将激活层添加到所述倒转神经网络。

31.根据权利要求25所述的方法,还包括:

通过向所述倒转神经网络输入logit或指示编码特征的数据中的至少一个来生成所述训练数据。

32.根据权利要求25所述的方法,还包括:

至少部分地基于层一致性损失和重建损失来微调所述倒转网络。

神经网络训练技术

技术领域

[0001] 至少一个实施例涉及用于使用倒转神经网络生成神经网络的训练数据的一个或更多个处理器。

背景技术

[0002] 使用预训练的神经网络来生成训练数据会使用大量的计算资源。可以改进使用预训练的神经网络生成训练数据的技术。

附图说明

[0003] 图1示出了根据至少一个实施例的使用倒转神经网络的知识蒸馏的示例；

[0004] 图2示出了根据至少一个实施例的教师神经网络和倒转神经网络的示例；

[0005] 图3示出了根据至少一个实施例的神经网络的块的倒转的示例；

[0006] 图4示出了根据至少一个实施例的倒转网络的输出的示例；

[0007] 图5示出了根据至少一个实施例的对倒转网络进行微调的示例；

[0008] 图6示出了根据至少一个实施例的在教师网络与学生网络之间的知识转移的示例过程；

[0009] 图7A示出了根据至少一个实施例的推理和/或训练逻辑；

[0010] 图7B示出了根据至少一个实施例的推理和/或训练逻辑；

[0011] 图8示出了根据至少一个实施例的神经网络的训练和部署；

[0012] 图9示出了根据至少一个实施例的示例数据中心系统；

[0013] 图10A示出了根据至少一个实施例的自主车辆的示例；

[0014] 图10B示出了根据至少一个实施例的图10A的自主车辆的相机位置和视野的示例；

[0015] 图10C是根据至少一个实施例的示出图10A的自主车辆的示例系统架构的框图；

[0016] 图10D是根据至少一个实施例的示出用于一个或更多个基于云的服务器与图10A的自主车辆之间进行通信的系统的图；

[0017] 图11是根据至少一个实施例的示出计算机系统的框图；

[0018] 图12是根据至少一个实施例的示出计算机系统的框图；

[0019] 图13示出了根据至少一个实施例的计算机系统；

[0020] 图14示出了根据至少一个实施例的计算机系统；

[0021] 图15A示出了根据至少一个实施例的计算机系统；

[0022] 图15B示出了根据至少一个实施例的计算机系统；

[0023] 图15C示出了根据至少一个实施例的计算机系统；

[0024] 图15D示出了根据至少一个实施例的计算机系统；

[0025] 图15E和图15F示出了根据至少一个实施例的共享编程模型；

[0026] 图16示出了根据至少一个实施例的示例性集成电路和相关的图形处理器。

[0027] 图17A和图17B示出了根据至少一个实施例的示例性集成电路和相关联的图形处

理器。

[0028] 图18A和图18B示出了根据至少一个实施例的附加的示例性图形处理器逻辑；

[0029] 图19示出了根据至少一个实施例的计算机系统；

[0030] 图20A示出了根据至少一个实施例的并行处理器；

[0031] 图20B示出了根据至少一个实施例的分区单元；

[0032] 图20C示出了根据至少一个实施例的处理集群；

[0033] 图20D示出了根据至少一个实施例的图形多处理器；

[0034] 图21示出了根据至少一个实施例的多图形处理单元(GPU)系统；

[0035] 图22示出了根据至少一个实施例的图形处理器；

[0036] 图23是根据至少一个实施例的示出用于处理器的处理器微架构的框图；

[0037] 图24示出了根据至少一个实施例的深度学习应用程序处理器；

[0038] 图25是根据至少一个实施例的示出了示例神经形态处理器的框图；

[0039] 图26示出了根据一个或更多个实施例的图形处理器的至少部分；

[0040] 图27示出了根据一个或更多个实施例的图形处理器的至少部分；

[0041] 图28示出了根据一个或更多个实施例的图形处理器的至少部分；

[0042] 图29是根据至少一个实施例的示出了图形处理器的图形处理引擎的框图；

[0043] 图30是根据至少一个实施例的示出了图形处理器核心的至少部分的框图；

[0044] 图31A和图31B示出了根据至少一个实施例的线程执行逻辑，其包括图形处理器核心的处理元件的阵列。

[0045] 图32示出了根据至少一个实施例的并行处理单元(“PPU”)；

[0046] 图33示出了根据至少一个实施例的通用处理集群(“GPC”)；

[0047] 图34示出了根据至少一个实施例的并行处理单元(“PPU”)的存储器分区单元；

[0048] 图35示出了根据至少一个实施例的流式多处理器。

[0049] 图36是根据至少一个实施例的高级计算管线的示例数据流图；

[0050] 图37是根据至少一个实施例的用于在高级计算管线中训练、适应、实例化和部署机器学习模型的示例系统的系统图；

[0051] 图38包括根据至少一个实施例的用于处理成像数据的高级计算管线的示例图示；

[0052] 图39A包括根据至少一个实施例的支持超声设备的虚拟仪器的示例数据流图；

[0053] 图39B包括根据至少一个实施例的支持CT扫描仪的虚拟仪器的示例数据流图；

[0054] 图40A示出了根据至少一个实施例的用于训练机器学习模型的过程的数据流图；以及

[0055] 图40B是根据至少一个实施例的利用预训练的注释模型来增强注释工具的客户端-服务器架构的示例图示。

具体实施方式

[0056] 图1示出了根据至少一个实施例的使用倒转(inverted)神经网络的知识蒸馏的示例。在至少一个实施例中，神经网络蒸馏的示例过程100包括从教师网络102到学生网络108的知识转移。在至少一个实施例中，为了执行该任务，从教师网络102生成倒转网络104，并使用倒转网络104来形成训练数据集106，训练数据集106包括从倒转网络104的输出生成的

图像。在至少一个实施例中，训练数据集106然后与教师数据集102一起用于训练学生网络108。

[0057] 在至少一个实施例中，知识蒸馏包括使知识从一个神经网络可转移到另一个神经网络的技术。在至少一个实施例中，它允许通过模仿(mimicking)预训练的模型(有时被称为教师网络)的输出logits来学习新模型(有时被称为学生网络)。在至少一个实施例中，知识蒸馏用于诸如修剪、量化、域适配、迁移学习、性能提升等之类的应用中。在至少一个实施例中，期望使得能够在无需访问与用于训练教师网络的那些图像具有相同分布的自然图像的情况下进行知识转移。

[0058] 在至少一个实施例中，分阶段执行从教师网络102到学生网络108的知识转移。在至少一个实施例中，第一阶段包括将教师网络102倒转以形成倒转网络104。

[0059] 在至少一个实施例中，第二阶段包括通过在宽域上通过倒转网络104传递输入来生成人工或合成训练数据的大型训练数据集106。在至少一个实施例中，该输出类似于最初用于训练教师网络102的训练数据。例如，在至少一个实施例中，训练数据集106包括用于训练教师网络102的图像的传真(facsimile)。

[0060] 在至少一个实施例中，第三阶段包括使用训练数据集106来将学习从教师神经网络102转移到学生神经网络108。在至少一个实施例中，通过使教师神经网络102和学生神经网络104基于来自训练数据集106的输入执行对应任务来完成该转移。在至少一个实施例中，基于指示学生网络108模仿教师网络102的程度的损失因子来训练学生神经网络108。

[0061] 在至少一个实施例中，这种知识蒸馏过程是通过电路和软件的任何合适的组合来执行的。在至少一个实施例中，这些组合包括本文关于各个附图描述的硬件和软件，诸如各个计算设备、并行处理单元等。

[0062] 在至少一个实施例中，倒转教师网络102以形成倒转网络104提高了示例过程100的计算效率。在至少一个实施例中，使用显式倒转框架来倒转预训练的教师模型(诸如教师网络102)。在至少一个实施例中，这允许利用经由倒转网络104的正向传递完成的数据合成以零样本学习方式对学生网络的后续训练。在至少一个实施例中，权重转置运算符用于教师网络102的直接倒转。在至少一个实施例中，在基于图像的情况下，这种直接倒转可以用于生成展现原始对象的各个特征(诸如其形状)的图像，并且可以有效地用于促进知识蒸馏。在至少一个实施例中，为了进一步改进倒转网络104，使用循环一致性损失来管理逐层特征匹配和批归一化统计，而不使用任何非合成图像。所得到的倒转网络104对于知识蒸馏是有效的，其中合成的数据具有改善的效率。

[0063] 图2示出了根据至少一个实施例的教师神经网络和倒转神经网络的示例。在至少一个实施例中，如示例200所描绘的，对教师神经网络202进行倒转以形成倒转神经网络204。在至少一个实施例中，这通过以下方式完成：单独倒转每个块来反转(reverse)预训练的教师网络202的总体计算流，并且然后以相反顺序堆叠这些倒转的块以构建倒转网络204。在至少一个实施例中，倒转网络204用于恢复或近似教师网络202的输入。

[0064] 在至少一个实施例中，如图2所示，教师网络202包括块1、块2、……块L 210-214，并且计算流源自从图像204导出的输入，通过块1、块2、……块L 210-214，至输出logits208a。

[0065] 在至少一个实施例中，倒转网络204包括倒转块1、倒转块2、……倒转块L 220-

224。在至少一个实施例中，通过来自教师网络202的对应块210-214的单独执行的倒转来生成这些块220-224。例如，在至少一个实施例中，通过倒转教师网络202的块1 210生成倒转块1 220，通过倒转块2 212生成倒转块1 222，以此类推。

[0066] 在至少一个实施例中，倒转网络204的计算流源自logits 208b，其被用作输入并且可对应于从教师网络202输出的logits 208a。在至少一个实施例中，计算流然后从倒转块L 224前进通过倒转块1 220，该倒转块1 220输出图像206。在至少一个实施例中，该输出图像是图像204的传真。在至少一个实施例中，通过倒转教师网络202引入了一些误差，并且因此输出图像206可能不是输入图像204的精确副本。

[0067] 在至少一个实施例中，倒转预训练的网络由于其是非线性的并且具有多对一映射而是具有挑战性的。例如，在至少一个实施例中，从高度精简的输出反转有效输入可能涉及显著的信息丢失。

[0068] 在至少一个实施例中，直接网络倒转包括利用已经学习的权重来识别倒转的可行路径，该可行路径在数学上重新制定计算流，以将输出投影回模型的输入。在至少一个实施例中，转置操作产生预训练模型中的各个计算块的有效倒转。在至少一个实施例中，通过反转这种块的流，创建新的倒转网络，其以简单的正向传递将输出映射回到输入。在至少一个实施例中，这种方法减轻了对从头开始训练辅助网络或随机变量的需要，并且由此降低了计算成本。

[0069] 在至少一个实施例中，使用循环一致性损失来微调倒转网络，以更好地促进目标。在至少一个实施例中，在没有对自然图像或其他训练数据的访问的情况下，如在无数据设置中，倒转网络被重新连接回到其目标模型，形成循环路径。在至少一个实施例中，假设理想地，倒转网络应当恢复其每嵌入的输入，同时合成图像或其他训练数据的另一传递应当达到该同一嵌入。在至少一个实施例中，基于这些假设，引入循环一致性损失，以惩罚沿该循环的网络响应中的失配，并相应地更新倒转网络。

[0070] 在至少一个实施例中，将倒转预训练的图像分类模型(老师)的任务公式化为F：$\mathbb{R}^n \to \mathbb{R}^k$，其中，n是输入空间的维度(例如，像素数)，并且k是所考虑的类别的数量。在无一般性损失的情况下，可以考虑具有以下形式的L块的前向网络：

[0071] $$\mathbf{y} = F(\boldsymbol{x}) = \{\phi_i\}_{i=1}^{L}(\boldsymbol{x}) = \phi_L\left(\phi_{L-1}\left(\ldots\phi_2\left(\phi_1(\boldsymbol{x})\right)\ldots\right)\right)。$$

[0072] 在至少一个实施例中，每个块$\Phi_1(\bullet)$表示其输入的标准三步变换：投影、归一化以及通过非线性函数的激活。在至少一个实施例中，投影选择可以在线性变换(有时被称为全连接层)或卷积变换(有时被称为卷积层)之间变化。在至少一个实施例中，批归一化用于归一化变换。在至少一个实施例中，对于非线性，考虑ReLU(x)＝max{x,0}形式的ReLU激活。在至少一个实施例中，用卷积层替换池化层。

[0073] 在至少一个实施例中，通过用最小附加训练通过显式倒转(inversion)模型F^{-1}倒转预训练模型$F(\bullet)$来形成倒转网络，使得：

[0074] $x \approx F^{-1}(y)$，以及$y \approx F(F^{-1}(y))$

[0075] 在至少一个实施例中，系统(诸如计算设备、处理器或并行处理单元)直接倒转预训练模型$F(\bullet)$的计算图。在至少一个实施例中，所述系统试图至少部分地基于追溯回到$F(\bullet)$的计算流上从相应的输出y恢复或近似输入x。在至少一个实施例中，通过$F(\bullet)$中的近

似反向运算获得结果反向计算图$F^{-1}(\bullet)$。然而，鉴于此任务是NP完成的，为这种任务利用预训练的$F(\bullet)$可能是有挑战性的，并且将期望避免使用真实数据来优化倒转模型，或避免对教师模型的权重应用诸如Lipschitz范数、输入-输出维数比或高斯分布之类的硬约束。在至少一个实施例中，避免使用这些约束，并且可使用现成的教师模型。

[0076] 在至少一个实施例中，系统通过每次一个地单独处理其计算块$\phi_l(x^{l-1})$来倒转网络。在至少一个实施例中，依次反转这些块导致合理的倒转结果。在至少一个实施例中，为了进一步改进该倒转过程，利用循环一致性损失。在至少一个实施例中，该循环一致性损失考虑块内信息流。

[0077] 在至少一个实施例中，系统通过转置执行直接倒转。在至少一个实施例中，系统执行模型的一个块中的每个层的直接倒转。在至少一个实施例中，不同的倒转过程被用于不同类型的层。

[0078] 图3示出了根据至少一个实施例的神经网络的块的倒转的示例。在至少一个实施例中，块是神经网络的层或一个或更多个层。在至少一个实施例中，块倒转的示例300涉及教师网络的块302，其包括一个或更多个卷积层310，按计算顺序随后是一个或更多个批归一化层312，并且随后是一个或更多个ReLU层314。

[0079] 在至少一个实施例中，通过倒转其各个层或各组层生成倒转块304。在至少一个实施例中，这些还可被称为块或子块。

[0080] 在至少一个实施例中，块304包括对应于教师块302中的一个或更多个卷积层310的一个或更多个转置卷积层320、对应于倒转批归一化层322的批归一化层312以及对应于ReLU层314的一个或更多个ReLU层324，其中倒转块304中的计算流从一个或更多个ReLU层324流向一个或更多个倒转批归一化层322，然后流向一个或更多个转置卷积层320。

[0081] 在至少一个实施例中，基于与教师块302中的一个或更多个层相关联的运算类型来执行那些层的转置。例如，在至少一个实施例中，针对线性层和全连接层、卷积层、批归一化层和激活层执行不同的转置运算。在至少一个实施例中，为了倒转教师模型中的块，系统忽略一个或更多个不可逆ReLU层324，并用反向批归一化和转置卷积来近似批归一化312和卷积层310的倒转，如下面更详细描述的。在至少一个实施例中，为了限制从这些上述倒转块传递的误差，一个或更多个ReLU层324被添加回到每个倒转块的开始。

[0082] 在至少一个实施例中，线性层或全连接层是可以经由矩阵-向量积实现的线性运算，如以下所示出的，其中通过使用X和Y来表示一个层或一个块的输入和输出来简化符号：

[0083] $Y=WX, \boldsymbol{W} \in \mathbb{R}^{n_l \times n_{l-1}}$

[0084] 其中，$\mathbf{X} \in \mathbb{R}^{n_{l-1}}$是向量化输入，并且$n_{l-1}$、$n_l$分别是输入和输出维度。在无一般性损失的情况下，这里省略偏置项b，因为它可以通过添加具有恒定输入的一个或更多个维度而被吸收到W中。

[0085] 在至少一个实施例中，使用线性代数，如果权重矩阵W是可逆的(W是正方形矩阵并且具有满秩)，则可以通过取权重矩阵的逆W^{-1}来倒转以上线性运算。然而，在至少一个实施例中，分类模型中的线性层通常具有小于一个扩展比$n_l/n_{l-1}<<1$。在至少一个实施例中，实际逆W^{-1}没有被很好地定义，并且近似值P被找到，使得$PW\approx I$，其中$\boldsymbol{I} \in \mathbb{R}^{n_{l-1} \times n_{l-1}}$被用作单位矩阵。如果W的列是正交向量，那么该矩阵称为半正交的，并且转置具有$W^TW\approx I$的属性。在至少一个实施例中，从提供正交性的正态分布提取初始化时的权重，即使在训练之后，大

多数向量也近似是正交的。

[0086] 在至少一个实施例中，卷积层被倒转。例如，在至少一个实施例中，将二维卷积运算符考虑为输入张量$\mathbf{X} \in \mathbb{R}^{C_{\text{in}} \times S_{\text{in}} \times S_{\text{in}}}$：

[0087] Y＝conv(W,X)，

[0088] 给定权重$\mathbf{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times K \times K}$和输出$\mathbf{Y} \in \mathbb{R}^{C_{\text{out}} \times S_{\text{out}} \times S_{\text{out}}}$，$C_{in}$、$C_{out}$作为输入和输出通道，并且K、$S_{in}$、$S_{out}$表示内核大小、输入和输出空间大小。在至少一个实施例中，该卷积层是对输入的另一线性运算，并且存在实现这种倒转的多种方式。在至少一个实施例中，使其与全连接层的前述公式一致，并且通过权重张量的aninput和托普利兹(Toeplitz)形式的向量化，将卷积视为矩阵-向量乘积：

[0089] VEC(Y)＝TOEP(W)VEC(X)。

[0090] 在至少一个实施例中，假设卷积层的输入和输出通道为一维。在至少一个实施例中，Toeplitz矩阵的大小则是S_{out}^2乘以S_{in}^2，并且Toeplitz矩阵的第i行对应于内核在卷积过程中的第i个窗口滑动位置，并且可以通过将内核展开成向量并且用零填充内核未覆盖区域来获得。在至少一个实施例中，这通过水平地串接用于输入通道的Toeplitz矩阵和垂直地串接用于输出通道的Toeplitz矩阵而扩展到多个输入和输出通道用例。

[0091] 在至少一个实施例中，用于Y和VEC(Y)的方程被视为在数学上等价，并且通过将TOEP(W)的转置取为下式来倒转卷积层：

[0092] $\text{VEC}(X) \approx \text{TOEP}(W)^{T}\text{VEC}(Y) = \text{TCONV}(W,Y)$

[0093] 在至少一个实施例中，Toeplitz矩阵TOEP(W)可以非常大(具有$C_{\text{out}}S_{\text{out}}^2 \times C_{\text{in}}S_{\text{in}}^2$的大小)，并且因此存储在存储器中是低效的或不切实际的。在至少一个实施例中，而是转置卷积运算符TCONV(•)用于倒转，以解决此类存储器使用问题。

[0094] 在至少一个实施例中，批归一化层被倒转。批归一化(“BN”)层专注于归一化深度神经网络的中间层中的激活。根据给定的通道，BN对所有激活应用该相同的归一化，减去一批中的样本上的均值激活，将中心的激活除以标准偏差(对于数值稳定性加∈)，并且然后应用通过γ、β参数化的仿射变换。BN层的定义是线性算子，其容易倒转。在至少一个实施例中，利用已经存储在目标模型中的BN统计。在至少一个实施例中，这些统计包括均值μ_r和方差σ_r^2的运行平均值。在至少一个实施例中，这些提供了原始激活的有效分布先验。在数学上，它们联合地作用于输入如下：

[0095] $$\mathbf{Y}_{i,:,:} = \frac{\mathbf{X}_{i,:,:} - \mu_{ri}}{\sqrt{\sigma_r^2{}_i + \epsilon}} \cdot \gamma_i + \beta_i$$

[0096] 在至少一个实施例中，根据此方程，则批归一化层可以经由线性变换被倒转，如下：

[0097] $$\mathbf{X}_{i,:,:}=\frac{\sqrt{\sigma_{r}^{2}{}_{i}+\epsilon}}{\gamma_{i}}\mathbf{Y}_{i,:,:}+\left(\mu_{r_i}-\frac{\beta_{i}\sqrt{\sigma_{r}^{2}{}_{i}+\epsilon}}{\gamma_{i}}\right)。$$

[0098] 在至少一个实施例中，在倒转网络中用适合于在倒转网络中使用的运算来替换激活函数。在至少一个实施例中，ReLU函数不可区分地过滤出所有负值，并且因而在没有额外信息的情况下不能被倒转。在至少一个实施例中，考虑教师模型中的第l块的输出，如果第l块中的最后一层是ReLU，则所有元素应当是非负的。在至少一个实施例中，在倒转期间，第l倒转块的输入是第l+1倒转块的输出。在至少一个实施例中，每个块的倒转都是近似值，并且引入可能导致第l倒转块的输入具有负值的一些倒转误差。在至少一个实施例中，为了限制倒转误差，在每个倒转块的开始处应用ReLU层。

[0099] 在至少一个实施例中，按块倒转教师网络中的运算符，然后将其组装成倒转网络。在至少一个实施例中，这种组装包括按照反向顺序以$F(\cdot)$串接倒转块。在至少一个实施例中，结果模型$F^{-1}(\cdot)$采用logit向量作为输入并且使用单个前向传递达到合成数据。在至少一个实施例中，在该倒转过程中不使用训练。在至少一个实施例中，刚才描述的通过转置的模型倒转不涉及显著的存储器和计算问题。在至少一个实施例中，通过转置的模型倒转允许使用预训练的“框外”模型(诸如MNIST)的倒转来生成图像。

[0100] 图4示出了根据至少一个实施例的倒转网络的输出的示例。在至少一个实施例中，在示例400中，对应于地面实况402的图像被用于训练教师网络以识别数值数字。在至少一个实施例中，该教师网络输出指示识别出的数字的logits。

[0101] 在至少一个实施例中，倒转网络用于基于logits生成图像。在至少一个实施例中，该倒转网络基于logits生成图像404。在至少一个实施例中，这些logits对应于来自教师网络的那些输出。例如，在至少一个实施例中，这些logits可以对应于包括数值数字的标识符的域。在至少一个实施例中，logits在给定域上被随机采样，并且作为输入被提供给倒转网络，以便生成各种图像。

[0102] 在至少一个实施例中，倒转网络用于基于特征来生成图像406。在至少一个实施例中，特征指嵌入或其他数据，其可以取自这个倒转网络的一个或更多个层中的任一个。在至少一个实施例中，从所采样特征生成的图像可以包括更大的多样性和改进的视觉特征，诸如具有不同长度和取向的线、圆和弧。在至少一个实施例中，代替倒转所有的教师模型，系统可以直接访问教师模型中的全连接层和倒转模型中的它们对应的倒转层的输出，并且将来自教师模型的特征输出直接传递到倒转模型的对应输入。在至少一个实施例中，这被描述为从特征倒转。在至少一个实施例中，从特征倒转的图像更加巧妙地反映地面实况图像的视觉特性，诸如形状、取向和笔触。

[0103] 在至少一个实施例中，通过将每个运算倒转而构建的倒转网络单独地累积导致最终输出降级的小误差。在至少一个实施例中，倒转块的校准可以用于减少这种误差传播。在至少一个实施例中，这通过使用逐层循环一致性优化倒转模型来完成。在至少一个实施例中，循环一致性损失用于微调经由直接倒转生成的倒转模型。在至少一个实施例中，为此，利用原始模型和倒转模型之间的直接每层对应。在至少一个实施例中，结果证明每个倒转块应该能够从原始模型产生对应块的预期输入。在至少一个实施例中，输入数据被馈送至

原始模型，并且存储所得到的激活，诸如块的输出以及输出logits。在至少一个实施例中，在完成此之后，这些所存储的激活被顺序地馈送到相应的倒转块，并且这些激活被用于将分布与预期的层输入进行匹配。

[0104] 图5示出了根据至少一个实施例的倒转网络的微调的示例500。在至少一个实施例中，为了优化块1，系统不仅最小化重建损失508，而且将倒转网络重新连接回到原始模型。在至少一个实施例中，如果倒转输入X′在感知上接近X，则它们之间的层一致性损失506应当很小。

[0105] 在至少一个实施例中，系统通过从倒转第一倒转块开始并一次优化随后的倒转块中的一个来优化倒转神经网络。在至少一个实施例中，一旦完成第l-1倒转块的优化，就将其视为固定的或最终的，并且开始后续第l倒转块的优化。在至少一个实施例中，该块优化过程以由图5组成的方式执行。在至少一个实施例中，给定来自数据分布的一批输入，系统在教师模型中存储目标(第l)块的输入$\{\phi_i\}_{i=1}^{l-1}(\mathbf{X})$和输出$\mathbf{Y}=\{\phi_i\}_{i=1}^{l}(\mathbf{X})$。在至少一个实施例中，为了优化倒转块l，将块l的输出传递到倒转块l并且获得块l的重建输入$\psi_l(Y)$，并且获得最终倒转数据$\mathbf{X}'=\{\psi_i\}_{i=l}^{1}(\mathbf{Y})$。在至少一个实施例中，原始块l的输入$\{\phi_i\}_{i=1}^{l-1}(\mathbf{X})$与从倒转块l重建输入$\psi_l(Y)$之间的$L_1$距离被用作测量逐元素重建质量的重建损失508，

[0106] $$\mathcal{L}_{\text{rec}}=\| \{\phi_i\}_{i=1}^{l-1}(\mathbf{X})-\psi_l(\mathbf{Y}) \|_1 \qquad (8)$$

[0107] 在至少一个实施例中，教师模型中的块被认为是多对一映射，并且可能不是完全可逆的，因为多个输入候选者可导致完全相同的输出。在至少一个实施例中，为了选择合理的重建输入，系统可以测量块输入和重建输入之间的语义距离或感知距离。

[0108] 在至少一个实施例中，使用预训练的教师模型来量化图像或特征嵌入之间的语义相似度或感知相似度。在至少一个实施例中，倒转的输入图像X′被传递回教师模型并获得块l的输入$\{\phi_i\}_{i=1}^{l-1}(\mathbf{X}')$和输出$\{\phi_i\}_{i=1}^{l}(\mathbf{X}')$。来自原始随机输入和倒转输入的块l的输入(和输出)之间的L_1距离用于将$\{\phi_i\}_{i=1}^{l-1}(\mathbf{X})$与$\psi_l(Y)$之间的语义相似度量化为

[0109] $$\mathcal{L}_{\text{cyc}}=\| \{\phi_i\}_{i=1}^{l-1}(\mathbf{X}')-\{\phi_i\}_{i=1}^{l-1}(\mathbf{X}) \|_1+\| \{\phi_i\}_{i=1}^{l}(\mathbf{X}')-\{\phi_i\}_{i=1}^{l}(\mathbf{X}) \|_1 。$$

[0110] 在至少一个实施例中，系统将重建损失和循环一致性损失组合在一起，并且以逐层的方式用目标函数来优化每个倒转块：

[0111] $$\mathcal{L}_{\text{inv}}=\alpha_{\text{cyc}}\cdot\mathcal{L}_{\text{cyc}}+\alpha_{\text{rec}}\cdot\mathcal{L}_{\text{rec}}。$$

[0112] 在这个方程中，α_{cyc}和α_{rec}是用于平衡损失项的超参数。在至少一个实施例中，该损失被称为倒转损失。在至少一个实施例中，此优化由来自数据集分布的随机输入图像驱动。在至少一个实施例中，为了避免对访问最初用于训练教师网络502的数据集的需要，使用深度倒转过程来获得单批合成图像作为提供关于原始分布的指导的代理，其中，深度倒转(与教师网络的直接倒转相反，如本文相对于各个实施例描述的)指的是最初由随机噪声组成的图像(或其他训练示例)通过教师网络被馈送并经过多次迭代被优化以形成在训练数据的原始分布中使用的图像的传真的过程。在至少一个实施例中，重复这个过程以生成一批

图像或训练示例，作为在原始训练分布中使用的内容的传真。

[0113] 在至少一个实施例中，通过深度倒转过程生成的单批数据足以支持倒转网络块的优化。在至少一个实施例中，在使用深度倒转生成该初始批之后，经由倒转网络504的前向传递生成倒转图像的新批。在至少一个实施例中，图像可用单个前向传递来生成。

[0114] 在至少一个实施例中，尽管从经优化的倒转模型生成的图像具有可接受的视觉质量，但它们不一定与教师模型的原始训练分布重叠。在至少一个实施例中，用这样的倒转数据进行知识蒸馏可以引起域偏移并且导致在所生成的训练数据的所得集合中次优的准确度。在至少一个实施例中，为了进一步改善所生成的数据的有用性，特征分布正则化被合并到倒转块的优化中。在至少一个实施例中，对于教师模型的块中的每个批归一化层，分布正则化尝试最小化计算的批均值和方差与存储的运行均值和方差之间的差。例如，在至少一个实施例中，当将倒转的输入图像传递回教师模型时，记录批均值$\mu(X')$和方差$\sigma^2(X')$，并且考虑那些与所存储的运行均值μ_r和方差σ_r^2之间的差：

[0115] $$\mathcal{L}_{\text{bn}} = \sum_{i=1}^{L} \left(\| \mu_{r_i} - \mu_i(\mathbf{X}') \|_2 + \| \sigma_{r_i}^2 - \sigma_i^2(\mathbf{X}') \|_2 \right)。$$

[0116] 在至少一个实施例中，用于优化倒转模型的总损失被公式化为：

[0117] $$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{inv}} + \alpha_{\text{bn}} \cdot \mathcal{L}_{\text{bn}}$$

[0118] 其中，α_{bn}是作为超参数的特征分布正则化项的系数。

[0119] 图6示出了根据至少一个实施例的在教师网络与学生网络之间的知识转移的示例过程。尽管示例过程700被描述为一系列操作，但应当理解的是，在实施例中，关于图7描述的操作可以各种方式改变，并且一些操作可以被省略、重新排序或与其他操作并行地执行，除非明确陈述或逻辑上暗示顺序，例如当来自一个操作的输入取决于另一操作的输出时。

[0120] 在至少一个实施例中，关于图7所描绘的操作由任何合适的系统执行，包括计算机、计算设备、处理器、图形处理单元、算术逻辑单元等中的一个或更多个。

[0121] 在602处，在至少一个实施例中，系统获得预训练的教师网络。在至少一个实施例中，预训练包括教导该教师网络执行人工智能任务，诸如推理任务。在至少一个实施例中，使用有时被称为“真实”训练数据的数据集或者有时被称为原始数据集的数据集来训练该教师网络。在至少一个实施例中，该数据集包括图像。在至少一个实施例中，训练该教师网络以对这些图像执行推理任务，诸如对象识别、对象分类、图像分割等。在至少一个实施例中，期望将此教师网络的知识转移到另一神经网络。在至少一个实施例中，这个其他神经网络被称为学生神经网络。

[0122] 在604处，在至少一个实施例中，系统计算教师网络块的倒转。在至少一个实施例中，块包括神经网络的一个或更多个层。在至少一个实施例中，根据关于图2-3所描述的实施例来执行这些块的倒转。

[0123] 在606处，在至少一个实施例中，通过以相反的顺序加载这些倒转块来生成倒转网络。在至少一个实施例中，相反的顺序是指计算顺序，使得倒转包括以与教师网络的计算方向相比相反的顺序加载这些块。例如，在至少一个实施例中，在较早的块之前加载和评估教师网络的较晚的块。

[0124] 在608处，在至少一个实施例中，此倒转网络被微调。在至少一个实施例中，以与关于图5所描述的实施例一致的方式执行微调。

[0125] 在610处，在至少一个实施例中，此倒转网络用于生成训练数据。在至少一个实施例中，将logits输入到该倒转网络，并且输出来自原始数据集的项的传真，其中传真(facsimile)是指原始项的近似。在至少一个实施例中，示例输出与关于图4所描绘的一致，但是将理解的是，这些示例旨在是说明性的，并且因此不应当以将潜在实施例限制为仅符合那些示例的那些实施例的方式来解释。

[0126] 在612处，在至少一个实施例中，此系统使用此教师网络和倒转网络来训练学生网络。在至少一个实施例中，使用知识蒸馏技术来将知识从该教师网络迁移到该学生网络。在至少一个实施例中，知识蒸馏包括利用此生成的训练数据来训练此学生网络，以模仿此教师网络的输出。

[0127] 在至少一个实施例中，示例系统包括过程。在至少一个实施例中，该处理器包括一个或更多个电路，用于使一个或更多个神经网络层以相反的顺序加载到该处理器，其中顺序是相对于预训练的教师网络的。在至少一个实施例中，该加载结合生成该教师网络的倒转版本来完成。在至少一个实施例中，该网络的一个或更多个层的块被倒转，例如，如关于图2-3所描述的，并且以相反的顺序加载。在至少一个实施例中，以相反的顺序评估这些块导致生成包括在训练教师网络时使用的训练数据的分布的传真的输出。

[0128] 在至少一个实施例中，倒转神经网络层包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。在至少一个实施例中，通过基于该预训练的神经网络的卷积层生成一个或更多个转置卷积层来完成该预训练的神经网络的卷积层的倒转。在至少一个实施例中，批归一化层的倒转是通过至少部分地基于一个或更多个批归一化层的线性变换生成一个或更多个倒转批归一化层来完成的。

[0129] 在至少一个实施例中，将一个或更多个激活层插入到倒转网络中。在至少一个实施例中，这些被插入在对应于预训练的网络中的ReLU层的点处。

[0130] 在至少一个实施例中，对倒转网络进行微调。在至少一个实施例中，至少部分地基于层一致性损失和重建损失来对倒转网络的一个或更多个神经网络层进行微调。

[0131] 在至少一个实施例中，从对应于相应的预训练网络的输出的域中提取倒转网络的输入。在至少一个实施例中，将该数据输入到倒转网络导致输出，该输出包括用于训练相应的预训练的教师网络的训练分布的传真。在至少一个实施例中，使用该预训练的教师网络要求该输出将学习迁移到学生网络。

[0132] 在至少一个实施例中，一种训练神经网络的方法包括：至少部分地基于第一神经网络生成倒转神经网络，使用此倒转神经网络生成训练数据，以及使用此生成的训练数据训练第二神经网络。在至少一个实施例中，该倒转神经网络具有相对于该原始的、未反转的神经网络的反转的计算流程。

[0133] 在至少一个实施例中，该倒转神经网络包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。在至少一个实施例中，至少部分地基于预训练神经网络的卷积层的转置来生成倒转卷积层。在至少一个实施例中，至少部分地基于预训练神经网络的批归一化层的线性变换来生成倒转批归一化层。在至少一个实施例中，一个或更多个激活层被添加到该倒转网络，以限制误差累积。

[0134] 在至少一个实施例中，该方法还包括：在倒转之后微调倒转网络。在至少一个实施例中，此微调至少部分地基于层一致性损失和重建损失。在至少一个实施例中，根据关于图

5描述的实施例实现微调。

[0135] 在至少一个实施例中，使用倒转神经网络来生成对应于在对应的预训练教师网络的训练分布中使用的训练数据的输出。在至少一个实施例中，通过将指示编码特征的logit或数据中的至少一个从该倒转网络的某个层输入到该倒转神经网络来生成训练数据。

[0136] 推理和训练逻辑

[0137] 图7A示出了用于执行与一个或更多个实施例相关联的推理和/或训练操作的推理和/或训练逻辑715。下面结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。

[0138] 在至少一个实施例中，推理和/或训练逻辑715可以包括但不限于代码和/或数据存储701，用于存储前向和/或输出权重和/或输入/输出数据，和/或在一个或更多个实施例的方面中配置被训练为和/或用于推理的神经网络的神经元或层的其他参数。在至少一个实施例中，训练逻辑715可以包括或耦合到用于存储图形代码或其他软件以控制时序和/或顺序的代码和/或数据存储701，其中权重和/或其他参数信息被加载以配置逻辑，包括整数和/或浮点单元(统称为算术逻辑单元(ALU))。在至少一个实施例中，代码(诸如图代码)基于该代码所对应的神经网络的架构将权重或其他参数信息加载到处理器ALU中。在至少一个实施例中，代码和/或数据存储701存储在使用一个或更多个实施例的方面训练和/或推理期间的输入/输出数据和/或权重参数的前向传播期间结合一个或更多个实施例训练或使用的神经网络的每个层的权重参数和/或输入/输出数据。在至少一个实施例中，代码和/或数据存储701的任何部分都可以包括在其他片上或片外数据存储内，包括处理器的L1、L2或L3高速缓存或系统存储器。

[0139] 在至少一个实施例中，代码和/或数据存储701的任何部分可以在一个或更多个处理器或其他硬件逻辑设备或电路的内部或外部。在至少一个实施例中，代码和/或数据存储701可以是高速缓存存储器、动态随机可寻址存储器(“DRAM”)、静态随机可寻址存储器(“SRAM”)、非易失性存储器(例如闪存)或其他存储。在至少一个实施例中，对代码和/或数据存储701是在处理器的内部还是外部的选择，例如，或者由DRAM、SRAM、闪存或某种其他存储类型组成，可以取决于存储片上或片外的可用存储空间，正在执行训练和/或推理功能的延迟要求，在神经网络的推理和/或训练中使用的数据的批大小或这些因素的某种组合。

[0140] 在至少一个实施例中，推理和/或训练逻辑715可以包括但不限于代码和/或数据存储705，以存储与在一个或更多个实施例的方面中被训练为和/或用于推理的神经网络的神经元或层相对应的反向和/或输出权重和/或输入/输出数据神经网络。在至少一个实施例中，在使用一个或更多个实施例的方面训练和/或推理期间，代码和/或数据存储705存储在输入/输出数据和/或权重参数的反向传播期间结合一个或更多个实施例训练或使用的神经网络的每个层的权重参数和/或输入/输出数据。在至少一个实施例中，训练逻辑715可以包括或耦合到用于存储图代码或其他软件以控制时序和/或顺序的代码和/或数据存储705，其中权重和/或其他参数信息被加载以配置逻辑，该逻辑包括整数和/或浮点单元(统称为算术逻辑单元(ALU))。

[0141] 在至少一个实施例中，代码(诸如图代码)使得基于该代码所对应的神经网络的架构将权重或其他参数信息加载到处理器ALU中。在至少一个实施例中，代码和/或数据存储705的任何部分可以与其他片上或片外数据存储一起包括，包括处理器的L1、L2或L3高速缓存或系统存储器。在至少一个实施例中，代码和/或数据存储705的任何部分可以在一个或

更多个处理器或其他硬件逻辑设备或电路上的内部或外部。在至少一个实施例中，代码和/或数据存储705可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如闪存)或其他存储。在至少一个实施例中，代码和/或数据存储705是在处理器的内部还是外部的选择，例如，是由DRAM、SRAM、闪存还是其他某种存储类型组成，取决于可用存储是片上还是片外，正在执行的训练和/或推理功能的延迟要求，在神经网络的推理和/或训练中使用的数据批大小或这些因素的某种组合。

[0142] 在至少一个实施例中，代码和/或数据存储701以及代码和/或数据存储705可以是分开的存储结构。在至少一个实施例中，代码和/或数据存储701以及代码和/或数据存储705可以是相同的存储结构。在至少一个实施例中，代码和/或数据存储701以及代码和/或数据存储705可以部分地被组合和部分分离。在至少一个实施例中，代码和/或数据存储701以及代码和/或数据存储705的任何部分可以与其他片上或片外数据存储包括在一起，包括处理器的L1、L2或L3高速缓存或系统存储器。

[0143] 在至少一个实施例中，推理和/或训练逻辑715可以包括但不限于一个或更多个算术逻辑单元(“ALU”)710(包括整数和/或浮点单元)，用于至少部分地基于训练和/或推理代码(例如，图代码)或由其指示来执行逻辑和/或数学运算，其结果可能会产生存储在激活存储720中的激活(例如，来自神经网络内部的层或神经元的输出值)，其是存储在代码和/或数据存储701和/或代码和/或数据存储705中的输入/输出和/或权重参数数据的函数。在至少一个实施例中，激活响应于执行指令或其他代码，由ALU710执行的线性代数和/或基于矩阵的数学生成在激活存储720中存储的激活，其中存储在代码和/或数据存储705中和/或代码和/或数据存储701中的权重值用作具有其他值的操作数，例如偏置值、梯度信息、动量值或其他参数或超参数，可以将任何或所有这些存储在代码和/或数据存储705或代码和/或数据存储701或其他片上或片外存储中。

[0144] 在至少一个实施例中，一个或更多个处理器或其他硬件逻辑设备或电路中包括一个或更多个ALU 710，而在另一实施例中，一个或更多个ALU 710可以在处理器或其他硬件逻辑设备或使用它们(例如协处理器)的电路外。在至少一个实施例中，可以将一个或更多个ALU 710包括在处理器的执行单元之内，或者以其他方式包括在由处理器的执行单元可访问的ALU组中，该处理器的执行单元可以在同一处理器内或者分布在不同类型的不同处理器之间(例如，中央处理单元、图形处理单元、固定功能单元等)。在至少一个实施例中，代码和/或数据存储701、代码和/或数据存储705以及激活存储720可以共享处理器或其他硬件逻辑设备或电路，而在另一实施例中，它们可以在不同的处理器或其他硬件逻辑设备或电路或相同和不同处理器或其他硬件逻辑设备或电路的某种组合中。在至少一个实施例中，激活存储720的任何部分可以与其他片上或片外数据存储包括在一起，包括处理器的L1、L2或L3高速缓存或系统存储器。此外，推理和/或训练代码可以与处理器或其他硬件逻辑或电路可访问的其他代码一起存储，并可以使用处理器的提取、解码、调度、执行、退出和/或其他逻辑电路来提取和/或处理。

[0145] 在至少一个实施例中，激活存储720可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如，闪存)或其他存储。在至少一个实施例中，激活存储720可以完全地或部分地在一个或更多个处理器或其他逻辑电路内部或外部。在至少一个实施例中，可以取决于片上或片外可用的存储，进行训练和/或推理功能的延迟要求，在推理和/或训练神经网络中

使用的数据的批量大小或这些因素的某种组合，选择激活存储720是处理器的内部还是外部，例如，或者包含DRAM、SRAM、闪存或其他存储类型。

[0146] 在至少一个实施例中，图7A中所示的推理和/或训练逻辑715可以与专用集成电路（“ASIC”）结合使用，例如来自Google的TensorFlow®处理单元、来自Graphcore™的推理处理单元（IPU）或来自Intel Corp的Nervana®（例如“Lake Crest”）处理器。在至少一个实施例中，图7A所示的推理和/或训练逻辑715可与中央处理单元（“CPU”）硬件，图形处理单元（“GPU”）硬件或其他硬件（例如现场可编程门阵列（“FPGA”））结合使用。

[0147] 图7B示出了根据至少一个实施例的推理和/或训练逻辑715。在至少一个实施例中，推理和/或训练逻辑715可以包括但不限于硬件逻辑，其中计算资源被专用或以其他方式唯一地连同对应于神经网络内的一层或更多层神经元的权重值或其他信息一起使用。在至少一个实施例中，图7B中所示的推理和/或训练逻辑715可以与专用集成电路（ASIC）结合使用，例如来自Google的TensorFlow®处理单元，来自Graphcore™的推理处理单元（IPU）或来自Intel Corp的Nervana®（例如“Lake Crest”）处理器。在至少一个实施例中，图7B中所示的推理和/或训练逻辑715可以与中央处理单元（CPU）硬件、图形处理单元（GPU）硬件或其他硬件（例如现场可编程门阵列（FPGA））结合使用。在至少一个实施例中，推理和/或训练逻辑715包括但不限于代码和/或数据存储701以及代码和/或数据存储705，其可以用于存储代码（例如，图代码）、权重值和/或其他信息，包括偏置值、梯度信息、动量值和/或其他参数或超参数信息。在图7B中所示的至少一个实施例中，代码和/或数据存储701以及代码和/或数据存储705中的每一个都分别与专用计算资源（例如计算硬件702和计算硬件706）相关联。在至少一个实施例中，计算硬件702和计算硬件706中的每一个包括一个或更多个ALU，这些ALU仅分别对存储在代码和/或数据存储701和代码和/或数据存储705中的信息执行数学函数（例如线性代数函数），执行函数的结果被存储在激活存储720中。

[0148] 在至少一个实施例中，代码和/或数据存储701和705以及相应的计算硬件702和706中的每一个分别对应于神经网络的不同层，使得从代码和/或数据存储701和计算硬件702的一个“存储/计算对701/702”得到的激活提供作为代码和/或数据存储705和计算硬件706的下一个“存储/计算对705/706”的输入，以便反映神经网络的概念组织。在至少一个实施例中，每个存储/计算对701/702和705/706可以对应于一个以上的神经网络层。在至少一个实施例中，在推理和/或训练逻辑715中可以包括在存储计算对701/702和705/706之后或与之并行的附加存储/计算对（未示出）。

[0149] 神经网络训练和部署

[0150] 图8示出了根据至少一个实施例的深度神经网络的训练和部署。在至少一个实施例中，使用训练数据集802来训练未经训练的神经网络806。在至少一个实施例中，训练框架804是PyTorch框架，而在其他实施例中，训练框架804是TensorFlow，Boost，Caffe，Microsoft Cognitive Toolkit/CNTK，MXNet，Chainer，Keras，Deeplearning4j或其他训练框架。在至少一个实施例中，训练框架804训练未经训练的神经网络806，并使它能够使用本文所述的处理资源来训练，以生成经训练的神经网络808。在至少一个实施例中，权重可以被随机选择或通过使用深度信念网络预训练。在至少一个实施例中，可以以有监督、部分有监督或无监督的方式执行训练。

[0151] 在至少一个实施例中，使用有监督学习来训练未经训练的神经网络806，其中训练数据集802包括与用于输入的期望输出配对的输入，或者其中训练数据集802包括具有已知输出的输入和神经网络806是手动分级的输出。在至少一个实施例中，以有监督的方式来训练未经训练的神经网络806，并且处理来自训练数据集802的输入，并将结果输出与一组期望或想要的输出进行比较。在至少一个实施例中，然后通过未经训练的神经网络806将误差传播回去。在至少一个实施例中，训练框架804调整控制未经训练的神经网络806的权重。在至少一个实施例中，训练框架804包括用于监视未经训练的神经网络806向模型(例如，经训练的神经网络808)收敛的程度的工具，适于基于输入数据(例如新数据集812)生成正确答案(例如结果814)的模型。在至少一个实施例中，训练框架804反复训练未经训练的神经网络806，同时调整权重以使用损失函数和调整算法(例如随机梯度下降)来改善未经训练的神经网络806的输出。在至少一个实施例中，训练框架804训练未经训练的神经网络806，直到未经训练的神经网络806达到期望的精度为止。在至少一个实施例中，然后可以部署经训练的神经网络808以实现任何数量的机器学习操作。

[0152] 在至少一个实施例中，使用无监督学习来训练未经训练的神经网络806，其中未经训练的神经网络806尝试使用未标记的数据来训练自己。在至少一个实施例中，无监督学习训练数据集802将包括输入数据，而没有任何关联的输出数据或“地面实况”数据。在至少一个实施例中，未经训练的神经网络806可以学习训练数据集802内的分组，并且可以确定各个输入如何与未经训练的数据集802相关。在至少一个实施例中，可以使用无监督训练来在经训练的神经网络808中生成自组织图，其能够执行对减少新数据集812的维度有用的操作。在至少一个实施例中，无监督训练也可以用于执行异常检测，这允许识别新数据集812中偏离新数据集812的正常模式的数据点。

[0153] 在至少一个实施例中，可以使用半监督学习，这是一种技术，其中在训练数据集802中包括标记数据和未标记数据的混合。在至少一个实施例中，训练框架804可以用于例如通过传递的学习技术来执行递增学习。在至少一个实施例中，递增学习使得经训练的神经网络808能够适应新数据集812，而不会忘记在初始训练期间注入到经训练的神经网络808内的知识。

[0154] 数据中心

[0155] 图9示出了可以使用至少一个实施例的示例数据中心900。在至少一个实施例中，数据中心900包括数据中心基础设施层910、框架层920、软件层930和应用程序层940。

[0156] 在至少一个实施例中，如图9所示，数据中心基础设施层910可以包括资源协调器912、分组计算资源914和节点计算资源(“节点C.R.”)916(1)-916(N)，其中“N”代表正整数(其可以是与其他图中使用的整数不同的整数“N”)。在至少一个实施例中，节点C.R.916(1)-916(N)可以包括但不限于任何数量的中央处理单元(“CPU”)或其他处理器(包括加速器、现场可编程门阵列(FPGA)、图形处理器等)，存储器存储设备918(1)-918(N)(例如动态只读存储器、固态硬盘或磁盘驱动器)，网络输入/输出(“NW I/O”)设备，网络交换机，虚拟机(“VM”)，电源模块和冷却模块等。在至少一个实施例中，节点C.R.916(1)-916(N)中的一个或更多个节点C.R.可以是具有一个或更多个上述计算资源的服务器。

[0157] 在至少一个实施例中，分组计算资源914可以包括容纳在一个或更多个机架内的节点C.R.的单独分组(未示出)，或者容纳在各个地理位置的数据中心内的许多机架(也未

示出)。在至少一个实施例中,分组的计算资源914内的节点C.R.的单独分组可以包括可以被配置或分配为支持一个或更多个工作负载的分组的计算、网络、存储器或存储资源。在至少一个实施例中,可以将包括CPU或处理器的几个节点C.R.分组在一个或更多个机架内,以提供计算资源来支持一个或更多个工作负载。在至少一个实施例中,一个或更多个机架还可以包括任何数量的电源模块、冷却模块和网络交换机,以任意组合。

[0158] 在至少一个实施例中,资源协调器912可以配置或以其他方式控制一个或更多个节点C.R.916(1)-916(N)和/或分组的计算资源914。在至少一个实施例中,资源协调器912可以包括用于数据中心900的软件设计基础结构("SDI")管理实体。在至少一个实施例中,资源协调器712可以包括硬件、软件或其某种组合。

[0159] 在至少一个实施例中,如图9所示,框架层920包括作业调度器922、配置管理器924、资源管理器926和分布式文件系统928。在至少一个实施例中,框架层920可以包括支持软件层930的软件932和/或应用程序层940的一个或更多个应用程序942的框架。在至少一个实施例中,软件932或应用程序942可以分别包括基于Web的服务软件或应用程序,例如由Amazon Web Services,Google Cloud和Microsoft Azure提供的服务或应用程序。在至少一个实施例中,框架层920可以是但不限于一种免费和开放源软件网络应用程序框架,例如可以利用分布式文件系统928来进行大范围数据处理(例如"大数据")的Apache Spark™(以下称为"Spark")。在至少一个实施例中,作业调度器932可以包括Spark驱动器,以促进对数据中心900的各个层所支持的工作负载进行调度。在至少一个实施例中,配置管理器924可以能够配置不同的层,例如软件层930和包括Spark和用于支持大规模数据处理的分布式文件系统928的框架层920。在至少一个实施例中,资源管理器926能够管理映射到或分配用于支持分布式文件系统928和作业调度器922的集群或分组计算资源。在至少一个实施例中,集群或分组计算资源可以包括数据中心基础设施层910上的分组计算资源914。在至少一个实施例中,资源管理器926可以与资源协调器912协调以管理这些映射的或分配的计算资源。

[0160] 在至少一个实施例中,包括在软件层930中的软件932可以包括由节点C.R.916(1)-916(N)的至少一部分,分组的计算资源914和/或框架层920的分布式文件系统928使用的软件。在至少一个实施例中,一种或更多种类型的软件可以包括但不限于Internet网页搜索软件、电子邮件病毒扫描软件、数据库软件和流视频内容软件。

[0161] 在至少一个实施例中,应用程序层940中包括的一个或更多个应用程序942可以包括由节点C.R.916(1)-916(N)的至少一部分、分组计算资源914和/或框架层920的分布式文件系统928使用的一种或更多种类型的应用程序。在至少一个实施例中,一种或更多种类型的应用程序可以包括但不限于任何数量的基因组学应用程序、认知计算、应用程序和机器学习应用程序,包括训练或推理软件,机器学习框架软件(例如PyTorch、TensorFlow、Caffe等)或其他与一个或更多个实施例结合使用的机器学习应用程序。

[0162] 在至少一个实施例中,配置管理器924、资源管理器926和资源协调器912中的任何一个可以基于以任何技术上可行的方式获取的任何数量和类型的数据来实现任何数量和类型的自我修改动作。在至少一个实施例中,自我修改动作可以减轻数据中心900的数据中心操作员做出可能不好的配置决定并且可以避免数据中心的未充分利用和/或执行差的部分。

[0163] 在至少一个实施例中，数据中心900可以包括工具、服务、软件或其他资源，以根据本文所述的一个或更多个实施例来训练一个或更多个机器学习模型或者使用一个或更多个机器学习模型来预测或推理信息。例如，在至少一个实施例中，可以通过使用上文关于数据中心900描述的软件和计算资源，根据神经网络架构通过计算权重参数来训练机器学习模型。在至少一个实施例中，通过使用通过本文所述的一种或更多种训练技术计算出的权重参数，可以使用上面与关于数据中心900所描述的资源，使用对应于一个或更多个神经网络的经训练的机器学习模型来推理或预测信息。

[0164] 在至少一个实施例中，数据中心可以使用CPU、专用集成电路(ASIC)、GPU、FPGA或其他硬件来使用上述资源来执行训练和/或推理。此外，上述的一个或更多个软件和/或硬件资源可以配置成一种服务，以允许用户训练或执行信息推理，例如图像识别、语音识别或其他人工智能服务。

[0165] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在系统图9中使用，用于至少部分地基于使用神经网络训练操作、神经网络函数和/或体系架构，或者本文所述的神经网络用例计算的权重参数推理或预测操作。

[0166] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0167] 自主车辆

[0168] 图10A示出了根据至少一个实施例的自动驾驶车辆1000的示例。在至少一个实施例中，自动驾驶车辆1000(在本文中可替代地称为“车辆1000”)可以是但不限于客运车辆，例如汽车、卡车、公共汽车和/或可容纳一个或更多个乘客的另一种类型的车辆。在至少一个实施例中，车辆1000可以是用于拖运货物的半牵引车-拖车。在至少一个实施例中，车辆1000可以是飞机、机器人车辆或其他类型的车辆。

[0169] 可以根据由美国运输部下属的国家公路交通安全管理局(“NHTSA”)和汽车工程师学会(“SAE”)“与用于道路机动车辆的驾驶自动化系统有关的术语(Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles)”(例如，于2018年6月15日发布的标准号J3016-201806，于2016年9月30日发布的标准号J3016-201609，以及该版本的以前和将来的版本此标准)定义的自动化级别来描述自动驾驶汽车。在至少一个实施例中，车辆1000可能能够根据自动驾驶级别的级别1至级别5中的一个或更多个来进行功能。例如，在至少一个实施例中，根据实施例，车辆1000可能能够进行条件自动化(级别3)、高度自动化(级别4)和/或全自动(级别5)。

[0170] 在至少一个实施例中，车辆1000可以包括但不限于组件，诸如底盘、车身、车轮(例如2、4、6、8、18等)、轮胎、车轴和车辆的其他组件。在至少一个实施例中，车辆1000可以包括但不限于推进系统1050，例如内燃机、混合动力装置、全电动发动机和/或另一种推进系统

类型。在至少一个实施例中，推进系统1050可以连接至车辆1000的传动系，其可以包括但不限于变速器，以使得能够对车辆1000进行推进。在至少一个实施例中，可以响应于从油门/加速器1052接收信号以控制推进系统1050。

[0171] 在至少一个实施例中，当推进系统1050正在运行时(例如，当车辆1000行驶时)，转向系统1054(其可以包括但不限于方向盘)用于使车辆1000转向(例如，沿着期望的路径或路线)。在至少一个实施例中，转向系统1054可以从转向致动器1056接收信号。在至少一个实施例中，方向盘对于全自动化(级别5)功能可以是可选的。在至少一个实施例中，制动传感器系统1046可以用于响应于从制动致动器1048和/或制动传感器接收到的信号来操作车辆制动器。

[0172] 在至少一个实施例中，控制器1036可以包括但不限于一个或更多个片上系统(“SoC”)(图10A中未示出)和/或图形处理单元(“GPU”)向车辆1000的一个或更多个组件和/或系统提供信号(例如，代表命令)。例如，在至少一个实施例中，控制器1036可以发送信号以通过制动致动器1048操作车辆制动，通过一个或更多个转向致动器1056操作转向系统1054，通过一个或更多个油门(throttle)/加速器1052操作推进系统1050。在至少一个实施例中，一个或更多个控制器1036可以包括一个或更多个机载(例如，集成)计算设备，其处理传感器信号并输出操作命令(例如，表示命令的信号)以实现自动驾驶和/或协助驾驶员驾驶车辆1000。在至少一个实施例中，一个或更多个控制器1036可以包括用于自动驾驶功能的第一控制器，用于功能安全功能的第二控制器，用于人工智能功能(例如计算机视觉)的第三控制器，用于信息娱乐功能的第四控制器，用于紧急情况下的冗余的第五控制器和/或其他控制器。在至少一个实施例中，单个控制器可以处理上述功能中的两个或更多个，两个或更多控制器可以处理单个功能和/或其任何组合。

[0173] 在至少一个实施例中，一个或更多个控制器1036响应于从一个或更多个传感器(例如，传感器输入)接收到的传感器数据，提供用于控制车辆1000的一个或更多个组件和/或系统的信号。在至少一个实施例中，传感器数据可以从传感器接收，传感器类型例如但不限于一个或更多个全球导航卫星系统(“GNSS”)传感器1058(例如，一个或更多个全球定位系统传感器)、一个或更多个RADAR传感器1060、一个或更多个超声波传感器1062、一个或更多个LIDAR传感器1064、一个或更多个惯性测量单元(IMU)传感器1066(例如，一个或更多个加速度计、一个或更多个陀螺仪、一个或更多个磁罗盘、一个或更多个磁力计等)、一个或更多个麦克风1096、一个或更多个立体声相机1068、一个或更多个广角相机1070(例如鱼眼相机)、一个或更多个红外相机1072、一个或更多个环绕相机1074(例如，360度相机)、远程相机(图10A中未示出)、中程相机(图10A中未示出)、一个或更多个速度传感器1044(例如，用于测量车辆1000的速度)、一个或更多个振动传感器1042、一个或更多个转向传感器1040、一个或更多个制动传感器(例如，作为制动传感器系统1046的一部分)和/或其他传感器类型接收。

[0174] 在至少一个实施例中，一个或更多个控制器1036可以从车辆1000的仪表板1032接收输入(例如，由输入数据表示)并通过人机接口(“HMI”)显示器1034、声音信号器、扬声器和/或车辆1000的其他组件提供输出(例如，由输出数据、显示数据等表示)。在至少一个实施例中，输出可包括信息，诸如车速、速度、时间、地图数据(例如，高清晰度地图(图10A中未显示)、位置数据(例如，车辆1000的位置，例如在地图上)、方向、其他车辆的位置(例如，占

用光栅)、关于对象的信息以及由一个或更多个控制器1036感知到的对象的状态等。例如,在至少一个实施例中,HMI显示器1034可以显示关于一个或更多个对象的存在的信息(例如,路牌、警告标志、交通信号灯变更等)和/或有关驾驶操作车辆已经、正在或将要制造的信息(例如,现在改变车道、在两英里内驶出34B出口等)。

[0175] 在至少一个实施例中,车辆1000还包括网络接口1024,其可以使用一个或更多个无线天线1026和/或一个或更多个调制解调器通过一个或更多个网络进行通信。例如,在至少一个实施例中,网络接口1024可能能够通过长期演进("LTE")、宽带码分多址("WCDMA")、通用移动电信系统("UMTS")、全球移动通信系统("GSM")、IMT-CDMA多载波("CDMA2000")网络等进行通信。在至少一个实施例中,一个或更多个无线天线1026还可以使用一个或更多个局域网(例如Bluetooth、Bluetooth Low Energy(LE)、Z-Wave、ZigBee等)和/或一个或更多个低功耗广域网(以下简称"LPWAN")(例如LoRaWAN、SigFox等协议),使环境中的对象(例如,车辆、移动设备)之间进行通信。

[0176] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中,推理和/或训练逻辑715可以在系统图10A中用于至少部分地基于使用神经网络训练操作\神经网络功能和/或架构或本文所述的神经网络用例计算出的权重参数来推理或预测操作。

[0177] 在至少一个实施例中,参照所述图,一个或更多个电路、处理器、计算系统、或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0178] 图10B示出了根据至少一个实施例的图10A的自动驾驶车辆1000的相机位置和视野的示例。在至少一个实施例中,相机和各自的视野是一个示例实施例,并且不旨在进行限制。例如,在至少一个实施例中,可以包括附加的和/或替代的相机和/或相机可以位于车辆1000上的不同位置。

[0179] 在至少一个实施例中,用于相机的相机类型可以包括但不限于可以适于与车辆1000的组件和/或系统一起使用的数字相机。在至少一个实施例中,一个或更多个相机可以以汽车安全完整性等级("ASIL")B和/或其他ASIL进行操作。在至少一个实施例中,根据实施例,相机类型可以具有任何图像捕获速率,例如60帧每秒(fps)、1220fps、240fps等。在至少一个实施例中,相机可以能够使用滚动快门、全局快门、另一种类型的快门或其组合。在至少一个实施例中,滤色器阵列可以包括红色透明透明("RCCC")滤色器阵列、红色透明透明蓝色("RCCB")滤色器阵列、红色蓝色绿色透明("RBGC")滤色器阵列、Foveon X3滤色器阵列、拜耳(Bayer)传感器("RGGB")滤色器阵列、单色传感器滤色器阵列和/或其他类型的滤色器阵列。在至少一个实施例中,可以使用透明像素相机,例如具有RCCC、RCCB和/或RBGC滤色器阵列的相机,以努力提高光敏性。

[0180] 在至少一个实施例中,一个或更多个相机可以用于执行先进驾驶员辅助系统("ADAS")功能(例如,作为冗余或故障安全设计的一部分)。例如,在至少一个实施例中,可

以安装多功能单声道相机以提供包括车道偏离警告、交通标志辅助和智能大灯控制的功能。在至少一个实施例中，一个或更多个相机(例如，所有相机)可以同时记录并提供图像数据(例如，视频)。

[0181] 在至少一个实施例中，可以将一个或更多个相机安装在安装组件中，例如定制设计的(三维(“3D”)打印的)组件，以便切出杂散光和来自在车辆1000内的反光(例如，仪表板的反射在挡风玻璃镜中反光)，其可能会干扰相机的图像数据捕获能力。关于后视镜安装组件，在至少一个实施例中，后视镜组件可以是3D打印定制的，使得相机安装板匹配后视镜的形状。在至少一个实施例中，一个或更多个相机可以被集成到后视镜中。在至少一个实施例中，对于侧视相机，一个或更多个相机也可以集成在舱室的每个角落的四个支柱内。

[0182] 在至少一个实施例中，具有包括车辆1000前面的环境的部分的视野的相机(例如，前向相机)可以用于环视，以及在一个或更多个控制器1036和/或控制SoC的帮助下帮助识别向前的路径和障碍物，从而提供对于生成占用网格和/或确定优选的车辆路径至关重要的信息。在至少一个实施例中，前向相机可以用于执行许多与LIDAR相似的ADAS功能，包括但不限于紧急制动、行人检测和避免碰撞。在至少一个实施例中，前向相机也可以用于ADAS功能和系统，包括但不限于车道偏离警告(“LDW”)、自动巡航控制(“ACC”)和/或其他功能(例如交通标志识别)。

[0183] 在至少一个实施例中，各种相机可以用于前向配置，包括例如包括CMOS(“互补金属氧化物半导体”)彩色成像器的单目相机平台。在至少一个实施例中，广角相机1070可以用于感知从外围进入的对象(例如，行人、过马路或自行车)。尽管在图10B中仅示出了一个广角相机1070，但是，在其他实施例中，车辆1000上可以有任何数量(包括零)的广角相机。在至少一个实施例中，任何数量的远程相机1098(例如，远程立体相机对)可用于基于深度的对象检测，尤其是对于尚未训练神经网络的对象。在至少一个实施例中，远程相机1098也可以用于对象检测和分类以及基本对象跟踪。

[0184] 在至少一个实施例中，任何数量的立体声相机1068也可以包括在前向配置中。在至少一个实施例中，一个或更多个立体声相机1068可以包括集成控制单元，该集成控制单元包括可缩放处理单元，该可缩放处理单元可以提供可编程逻辑(“FPGA”)和具有单个芯片上集成的控制器局域网(“CAN”)或以太网接口的多核心微处理器。在至少一个实施例中，这样的单元可以用于生成车辆1000的环境的3D地图，包括对图像中所有点的距离估计。在至少一个实施例中，一个或更多个立体相机1068可以包括但不限于紧凑型立体视觉传感器，其可以包括但不限于两个相机镜头(左右分别一个)和一个图像处理芯片，其可以测量从车辆1000到目标对象的距离并使用所生成的信息(例如，元数据)来激活自主紧急制动和车道偏离警告功能。在至少一个实施例中，除了本文所述的那些之外，还可以使用其他类型的立体相机1068。

[0185] 在至少一个实施例中，具有包括车辆1000侧面的环境的一部分的视野的相机(例如，侧视相机)可以用于环绕查看，从而提供用于创建和更新占据网格的信息，以及产生侧面碰撞警告。例如，在至少一个实施例中，环绕相机1074(例如，如图10B所示的四个环绕相机)可以定位在车辆1000上。在至少一个实施例中，一个或更多个环绕相机1074可以包括但不限于，任意数量和组合的广角相机、一个或更多个鱼目镜头、一个或更多个360度相机和/或类似相机。例如，在至少一个实施例中，四个鱼目镜头相机可以位于车辆1000的前、后和

侧面。在至少一个实施例中，车辆1000可以使用三个环绕相机1074(例如，左、右和后面)，并且可以利用一个或更多个其他相机(例如，前向相机)作为第四个环视相机。

[0186] 在至少一个实施例中，具有包括车辆1000后方的环境的一部分的视野的相机(例如，后视相机)可以用于停车辅助、环视、后方碰撞警告、以及创建和更新占用光栅。在至少一个实施例中，可以使用各种各样的相机，包括但不限于还适合作为一个或更多个前向相机的相机(例如，远程相机1098和/或一个或更多个中程相机1076、一个或更多个立体相机1068、一个或更多个红外相机1072等)，如本文所述。

[0187] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。结合图7A和/或图7B，在本文中提供了关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在图10B的系统中使用，用于至少部分地基于使用神经网络训练操作计算出的权重参数、神经网络函数和/或架构、或本文所述的神经网络用例来推理或预测操作。

[0188] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0189] 图10C示出了根据至少一个实施例的图10A的自动驾驶车辆1000的示例系统架构的框图。在至少一个实施例中，图10C中的车辆1000的一个或更多个组件、一个或更多个特征和一个或更多个系统中的每一个都示出为经由总线1002连接。在至少一个实施例中，总线1002可包括但不限于CAN数据接口(在本文中可替代地称为“CAN总线”)。在至少一个实施例中，CAN可以是车辆1000内部的网络，用于帮助控制车辆1000的各种特征和功能，例如制动器的致动、加速、制动、转向、雨刷等。在一个实施例中，总线1002可以配置成具有数十个甚至数百个节点，每个节点具有其自己的唯一识别符(例如，CAN ID)。在至少一个实施例中，可以读取总线1002以找到方向盘角度、地面速度、发动机每分钟转动次数(“RPM”)、按钮位置和/或其他车辆状态指示器。在至少一个实施例中，总线1002可以是符合ASIL B的CAN总线。

[0190] 在至少一个实施例中，除了CAN之外或来自CAN，可使用FlexRay和/或以太网(Ethernet)协议。在至少一个实施例中，可以有任意数量的成型总线1002，其可以包括但不限于零或更多的CAN总线，零或更多的FlexRay总线，零或更多的以太网总线，和/或零或更多的使用其他协议的其他类型的总线。在至少一个实施例中，两个或更多个总线可以用于执行不同的功能，和/或可以用于冗余。例如，第一总线可以用于碰撞避免功能，并且第二总线可以用于致动控制。在至少一个实施例中，总线1002的每个总线可以与车辆1000的任何组件通信，并且总线1002的两个或更多个可以与相应的组件通信。在至少一个实施例中，任何数量的片上系统(“SoC”)1004(例如，SoC 1004(A)和SoC 1004(B))中的每一个，一个或更多个控制器1036中的每一个和/或车辆内的每个计算机都可以访问相同的输入数据(例如，来自车辆1000的传感器的输入)，并且可以连接到公共总线，例如CAN总线。

[0191] 在至少一个实施例中，车辆1000可以包括一个或更多个控制器1036，诸如本文关

于图10A所描述的那些。在至少一个实施例中，控制器1036可以用于多种功能。在至少一个实施例中，控制器1036可以耦合到车辆1000的各种其他组件和系统中的任何一个，并且可以用于控制车辆1000、车辆1000的人工智能、车辆1000的信息娱乐和/或其他功能。

[0192] 在至少一个实施例中，车辆1000可以包括任何数量的SoC 1004。在至少一个实施例中，SoC 1004中的每一个可以包括但不限于中央处理单元(“一个或更多个CPU”)1006、图形处理单元(“一个或更多个GPU”)1008、一个或更多个处理器1010、一个或更多个高速缓存1012、一个或更多个加速器1014、一个或更多个数据存储1016和/或其他未显示的组件和特征。在至少一个实施例中，一个或更多个SoC 1004可以用于在各种平台和系统中控制车辆1000。例如，在至少一个实施例中，一个或更多个SoC 1004可以与高清晰度(“HD”)地图1022在系统(例如，车辆1000的系统)中组合，该高清晰度地图1022可以经由网络接口1024从一个或更多个服务器(图10C中未示出)获得地图刷新和/或更新。

[0193] 在至少一个实施例中，一个或更多个CPU 1006可以包括CPU集群或CPU复合体(在本文中可替代地称为“CCPLEX”)。在至少一个实施例中，一个或更多个CPU 1006可以包括多个核心和/或二级(“L2”)高速缓存。例如，在至少一个实施例中，一个或更多个CPU 1006可以在相互耦合的多处理器配置中包括八个核心。在至少一个实施例中，一个或更多CPU 1006可以包括四个双核心集群，其中每个集群具有专用的L2高速缓存(例如，2MB L2高速缓存)。在至少一个实施例中，一个或更多CPU 1006(例如，CCPLEX)可以配置成支持同时的集群操作，使得一个或更多CPU 1006的集群的任何组合在任何给定的时间都可以是活跃的。

[0194] 在至少一个实施例中，一个或更多个CPU 1006可以实现电源管理功能，这些功能包括但不限于以下特征中的一个或更多个：空闲时可以自动对各个硬件模块进行时钟门控以节省动态功率；当该核心由于执行等待中断(“WFI”)/事件等待(“WFE”)指令而未主动执行指令时，可以对每个核心时钟进行门控；每个核心都可以独立供电；当所有核心都被时钟门控或功率门控时，每个核心集群可以被独立地时钟门控；以及/或当所有核心都被功率门控时，每个核心集群可以被独立地功率门控。在至少一个实施例中，一个或更多CPU 1006可以进一步实现用于管理功率状态的增强算法，其中指定了允许的功率状态和预期的唤醒时间，并且硬件/微码确定了针对核心、集群和CCPLEX输入的最佳功率状态。在至少一个实施例中，处理核心可以在软件中支持简化的功率状态输入序列，其中工作被分担给微码。

[0195] 在至少一个实施例中，一个或更多个GPU 1008可以包括集成的GPU(在本文中或者称为“iGPU”)。在至少一个实施例中，一个或更多个GPU 1008可以是可编程的，并且对于并行工作负载可以是有效的。在至少一个实施例中，一个或更多个GPU 1008可以使用增强的张量指令集。在至少一个实施例中，一个或更多个GPU 1008可以包括一个或更多个流式微处理器，其中每个流式微处理器可以包括一级(“L1”)高速缓存(例如，具有至少96KB的存储容量的L1高速缓存)，以及两个或更多个流式微处理器可以共享L2高速缓存(例如，具有512KB存储容量的L2高速缓存)。在至少一个实施例中，一个或更多个GPU 1008可以包括至少八个流式微处理器。在至少一个实施例中，一个或更多个GPU 1008可以使用计算应用程序编程接口(API)。在至少一个实施例中，一个或更多GPU 1008可以使用一个或更多个并行计算平台和/或编程模型(例如，NVIDIA的CUDA模型)。

[0196] 在至少一个实施例中，一个或更多个GPU 1008可以经功耗优化以在汽车和嵌入式用例中获得最佳性能。例如，在一个实施例中，可以在鳍式场效应晶体管(“FinFET”)电路上

制造一个或更多个GPU 1008。在至少一个实施例中，每个流式微处理器可以包含多个划分为多个块的混合精度处理核心，。例如但不限于，可以将64个PF32核心和32个PF64核心划分为四个处理块。在至少一个实施例中，可以为每个处理块分配16个FP32核心、8个FP64核心、16个INT32核心、两个用于深度学习矩阵算术的混合精度NVIDIA张量核心、零级(“L0”)指令缓存、线程束调度器、分派单元和/或64KB寄存器文件。在至少一个实施例中，流式微处理器可以包括独立的并行整数和浮点数据路径来提供混合了计算和寻址运算的工作量的有效执行。在至少一个实施例中，流式微处理器可以包括独立的线程调度能力，以实现更细粒度的同步和并行线程之间的协作。在至少一个实施例中，流式微处理器可以包括组合的L1数据高速缓存和共享存储器单元，以便在简化编程的同时提高性能。

[0197] 在至少一个实施例中，一个或更多个GPU 1008可以包括高带宽存储器(“HBM”)和/或16GB HBM2存储器子系统，以在一些示例中提供约900GB/秒的峰值存储带宽。在至少一个实施例中，除了或替代于HBM存储器，可以使用同步图形随机存取存储器(“SGRAM”)，例如图形双倍数据速率类型的五同步随机存取存储器(“GDDR5”)。

[0198] 在至少一个实施例中，一个或更多个GPU 1008可以包括统一存储器技术。在至少一个实施例中，地址转换服务(“ATS”)支持可以用于允许一个或更多个GPU 1008直接访问一个或更多个CPU 1006页表。在至少一个实施例中，当一个或更多个GPU 1008中的GPU的一个存储器管理单元(“MMU”)经历未命中时，可以将地址转换请求发送到一个或更多个CPU 1006。作为响应，在至少一个实施例中，一个或更多个CPU 1006中的2 CPU可以在其页面表中查找地址的虚拟-物理的映射并将转换传送回一个或更多个GPU 1008。在至少一个实施例中，统一存储器技术可以允许单个统一虚拟地址空间用于一个或更多个CPU 1006和一个或更多个GPU 1008两者的存储器，从而简化了一个或更多个GPU 1008的编程以及将应用程序移植到一个或更多个GPU 1008。

[0199] 在至少一个实施例中，一个或更多个GPU 1008可以包括任意数量的访问计数器，其可以跟踪一个或更多个GPU 1008对其他处理器的存储器的访问频率。在至少一个实施例中，一个或更多个访问计数器可以帮助确保将存储器页移动到最频繁访问页面的处理器的物理存储器中，从而提高处理器之间共享的存储器范围的效率。

[0200] 在至少一个实施例中，一个或更多个SoC 1004可以包括任何数量的高速缓存1012，包括本文所述的那些。例如，在至少一个实施例中，一个或更多个高速缓存1012可以包括可用于一个或更多个CPU 1006和一个或更多个GPU 1008(例如，连接到CPU 1006和GPU 1008)的三级(“L3”)高速缓存。在至少一个实施例中，一个或更多个高速缓存1012可以包括回写式高速缓存，该回写式高速缓存可以例如通过使用高速缓存相干协议(例如，MEI、MESI、MSI等)来跟踪线的状态。在至少一个实施例中，尽管可以使用较小的高速缓存大小，根据实施例，L3高速缓存可以包括4MB存储器或更多。

[0201] 在至少一个实施例中，一个或更多个SoC 1004可以包括一个或更多个加速器1014(例如，硬件加速器、软件加速器或其组合)。在至少一个实施例中，一个或更多个SoC 1004可以包括硬件加速集群，其可以包括优化的硬件加速器和/或大的片上存储器。在至少一个实施例中，大的片上存储器(例如4MB的SRAM)可以使硬件加速集群能够加速神经网络和其他计算。在至少一个实施例中，硬件加速集群可以用于补充一个或更多个GPU 1008并且卸载一个或更多个GPU 1008的一些任务(例如，释放一个或更多个GPU 1008的更多周期以执

行其他任务)。在至少一个实施例中,一个或更多个加速器1014可以用于足够稳定以经得起加速检验的目标工作负载(例如,感知、卷积神经网络(“CNN”)、递归神经网络(“RNN”)等)。在至少一个实施例中,CNN可以包括基于区域或区域卷积神经网络(“RCNN”)和快速RCNN(例如,如用于对象检测)或其他类型的CNN。

[0202] 在至少一个实施例中,一个或更多个加速器1014(例如,硬件加速集群)可以包括一个或更多个深度学习加速器(“DLA”)。在至少一个实施例中,一个或更多个DLA可以包括但不限于一个或更多个Tensor处理单元(“TPU”),其可以配置成每秒提供额外的10万亿次操作用于深度学习应用程序和推理。在至少一个实施例中,TPU可以是配置成并被优化用于执行图像处理功能(例如,用于CNN、RCNN等)的加速器。在至少一个实施例中,可以针对神经网络类型和浮点运算以及推理的特定集合进一步优化一个或更多个DLA。在至少一个实施例中,一个或更多个DLA的设计可以提供比典型的通用GPU更高的每毫米性能,并且通常大大超过CPU的性能。在至少一个实施例中,一个或更多个TPU可执行若干功能,包括支持例如INT8、INT16和FP16数据类型以用于特征和权重的单实例卷积功能以及后处理器功能的。在至少一个实施例中,一个或更多个DLA可以针对各种功能中的任何功能,在处理或未处理的数据上快速且有效地执行神经网络,尤其是CNN,包括例如但不限于:用于使用来自相机传感器的数据进行对象识别和检测的CNN;用于使用来自相机传感器的数据进行距离估算的CNN;用于使用来自麦克风的数据进行紧急车辆检测以及识别和检测的CNN;用于使用来自相机传感器的数据进行人脸识别和车主识别的CNN;以及/或用于安全和/或安全相关事件的CNN。

[0203] 在至少一个实施例中,DLA可以执行一个或更多个GPU 1008的任何功能,并且通过使用推理加速器,例如,设计者可以将一个或更多个DLA或一个或更多个GPU 1008作为目标用于任何功能。例如,在至少一个实施例中,设计者可以将CNN的处理和浮点运算集中在一个或更多个DLA上,并将其他功能留给一个或更多个GPU 1008和/或一个或更多个加速器1014。

[0204] 在至少一个实施例中,一个或更多个加速器1014可以包括可编程视觉加速器(“PVA”),其在本文中可以可替代地称为计算机视觉加速器。在至少一个实施例中,一个或更多个PVA可以设计和配置为加速用于高级驾驶员辅助系统(“ADAS”)1038、自动驾驶、增强现实(“AR”)应用程序和/或虚拟现实(“VR”)应用程序的计算机视觉算法。在至少一个实施例中,一个或更多个PVA可以在性能和灵活性之间取得平衡。例如,在至少一个实施例中,一个或更多个PVA中的每一个可以包括例如但不限于任何数量的精简指令集计算机(“RISC”)核心、直接存储器访问(“DMA”)和/或任意数量的向量处理器。

[0205] 在至少一个实施例中,RISC核心可以与图像传感器(例如,本文描述的任意相机的图像传感器)、图像信号处理器等交互。在至少一个实施例中,每个RISC核心可以包括任意数量的存储器。在至少一个实施例中,根据实施例,RISC核心可以使用多种协议中的任意一种。在至少一个实施例中,RISC核心可以执行实时操作系统(“RTOS”)。在至少一个实施例中,可以使用一个或更多个集成电路设备、专用集成电路(“ASIC”)和/或存储设备来实现RISC核心。例如,在至少一个实施例中,RISC核心可以包括指令高速缓存和/或紧密耦合的RAM。

[0206] 在至少一个实施例中,DMA可以使PVA的组件能够独立于一个或更多个CPU 1006访

问系统存储器。在至少一个实施例中,DMA可以支持用于向PVA提供优化的任何数量的特征,包括但不限于,支持多维寻址和/或循环寻址。在至少一个实施例中,DMA可以支持多达六个或更多个寻址的维度,其可以包括但不限于块宽度、块高度、块深度、水平块步进、垂直块步进和/或深度步进。

[0207] 在至少一个实施例中,向量处理器可以是可编程处理器,其可以设计为有效且灵活地执行用于计算机视觉算法并提供信号处理能力的编程。在至少一个实施例中,PVA可以包括PVA核心和两个向量处理子系统分区。在至少一个实施例中,PVA核心可以包括处理器子系统、DMA引擎(例如,两个DMA引擎)和/或其他外围设备。在至少一个实施例中,向量处理子系统可以用作PVA的主要处理引擎,并且可以包括向量处理单元(“VPU”)、指令高速缓存和/或向量存储器(例如“VMEM”)。在至少一个实施例中,VPU核心可以包括数字信号处理器,例如,单指令多数据(“SIMD”)、超长指令字(“VLIW”)数字信号处理器。在至少一个实施例中,SIMD和VLIW的组合可以提高吞吐量和速度。

[0208] 在至少一个实施例中,每个向量处理器可以包括指令高速缓存并且可以耦合到专用存储器。结果,在至少一个实施例中,每个向量处理器可以配置为独立于其他向量处理器执行。在至少一个实施例中,特定PVA中包括的向量处理器可以配置为采用数据并行性。例如,在至少一个实施例中,单个PVA中包括的多个向量处理器可以执行通用计算机视觉算法,除了在图像的不同区域上之外。在至少一个实施例中,包括在特定PVA中的向量处理器可以在一个图像上同时执行不同的计算机视觉算法,或者甚至在序列图像或部分图像上执行不同的算法。在至少一个实施例中,除其他外,在硬件加速集群中可以包括任何数量的PVA,并且在每个PVA中可以包括任何数量的向量处理器。在至少一个实施例中,PVA可以包括附加的纠错码(“ECC”)存储器,以增强整体系统安全性。

[0209] 在至少一个实施例中,一个或更多个加速器1014可以包括片上计算机视觉网络和静态随机存取存储器(“SRAM”),用于为一个或更多个加速器1014提供高带宽,低延迟SRAM。在至少一个实施例中,片上存储器可以包括至少4MB SRAM,其包括例如但不限于八个现场可配置的内存块,PVA和DLA均可以对其进行访问。在至少一个实施例中,每对存储块可以包括高级外围总线(“APB”)接口、配置电路、控制器和多路复用器。在至少一个实施例中,可以使用任何类型的存储器。在至少一个实施例中,PVA和DLA可以经由为PVA和DLA提供对存储器的高速访问的主干网来访问存储器。在至少一个实施例中,主干网可以包括片上计算机视觉网络,其将PVA和DLA互连到存储器(例如,使用APB)。

[0210] 在至少一个实施例中,片上计算机视觉网络可以包括接口,该接口在传输任何控制信号/地址/数据之前确定PVA和DLA均提供就绪和有效信号。在至少一个实施例中,接口可以提供用于发送控制信号/地址/数据的单独的相位和单独的信道,以及用于连续数据传输的突发型通信。在至少一个实施例中,尽管可以使用其他标准和协议,但是接口可以符合国际标准化组织(“ISO”)26262或国际电工委员会(“IEC”)61508标准。

[0211] 在至少一个实施例中,一个或更多个SoC 1004可以包括实时视线追踪硬件加速器。在至少一个实施例中,实时视线追踪硬件加速器可以用于快速且有效地确定对象的位置和范围(例如,在世界模型内),以生成实时可视化模拟,以用于RADAR信号解释,用于声音传播合成和/或分析,用于SONAR系统的模拟,用于一般的波传播模拟,与用于定位和/或其他功能的LIDAR数据进行比较,和/或用于其他用途。

[0212] 在至少一个实施例中，一个或更多个加速器1014具有用于自动驾驶的广泛用途。在至少一个实施例中，PVA可以用于ADAS和自动驾驶汽车中的关键处理阶段。在至少一个实施例中，在低功耗和低延迟下PVA的能力与需要可预测的处理的算法域良好匹配。换句话说，PVA在半密集或密集的常规计算中表现出色，即使在小型数据集上也是如此，这些数据集可能需要具有低延迟和低功耗的可预测的运行时间。在至少一个实施例中，诸如在车辆1000中PVA可能被设计为运行经典的计算机视觉算法，因为它们可以在对象检测和整数数学运算方面是有效的。

[0213] 例如，根据技术的至少一个实施例，PVA被用于执行计算机立体视觉。在至少一个实施例中，可以在一些示例中使用基于半全局匹配的算法，尽管这并不意味着限制性。在至少一个实施例中，用于3-5级自动驾驶的应用程序在运行中使用动态的估计/立体匹配(例如，从运动中恢复结构、行人识别、车道检测等)。在至少一个实施例中，PVA可以对来自两个单目相机的输入执行计算机立体视觉功能。

[0214] 在至少一个实施例中，PVA可以用于执行密集的光流。例如，在至少一个实施例中，PVA可以处理原始RADAR数据(例如，使用4D快速傅立叶变换)以提供处理后的RADAR数据。在至少一个实施例中，例如，通过处理原始飞行时间数据以提供处理后的飞行时间数据，将PVA用于飞行时间深度处理。

[0215] 在至少一个实施例中，DLA可用于运行任何类型的网络以增强控制和驾驶安全性，包括例如但不限于神经网络，其输出用于每个对象检测的置信度。在至少一个实施例中，可以将置信度表示或解释为概率，或者表示为提供每个检测相对于其他检测的相对“权重”。在至少一个实施例中，置信度测量结果使系统能够做出进一步的决定，即关于哪些检测应当被认为是真正的阳性检测而不是假阳性检测。在至少一个实施例中，系统可以为置信度设置阈值，并且仅将超过阈值的检测视为真阳性检测。在使用自动紧急制动(“AEB”)系统的实施例中，假阳性检测将导致车辆自动执行紧急制动，这显然是不希望的。在至少一个实施例中，高度自信的检测可以被认为是AEB的触发。在至少一个实施例中，DLA可以运行用于回归置信度值的神经网络。在至少一个实施例中，神经网络可以将参数的至少一些子集作为其输入，例如包围盒尺寸，获得的地平面估计(例如，从另一子系统)，与从神经网络和/或其他传感器(例如，一个或更多个LIDAR传感器1064或一个或更多个RADAR传感器1060)等获得的对象的车辆1000方向、距离、3D位置估计相关的一个或更多个IMU传感器1066的输出。

[0216] 在至少一个实施例中，一个或更多个SoC 1004可以包括一个或更多个数据存储装置1016(例如，存储器)。在至少一个实施例中，一个或更多个数据存储1016可以是一个或更多个SoC 1004的片上存储器，其可以存储要在一个或更多个GPU 1008和/或DLA上执行的神经网络。在至少一个实施例中，一个或更多个数据存储1016可以具有足够大的容量以存储神经网络的多个实例以用于冗余和安全。在至少一个实施例中，一个或更多个数据存储1016可以包括L2或L3高速缓存。

[0217] 在至少一个实施例中，一个或更多个SoC 1004可以包括任何数量的处理器1010(例如，嵌入式处理器)。在至少一个实施例中，一个或更多个处理器1010可以包括启动和电源管理处理器，该启动和电源管理处理器可以是专用处理器和子系统，以处理启动电源和管理功能以及相关的安全实施。在至少一个实施例中，启动和电源管理处理器可以是一个或更多个SoC 1004启动序列的一部分，并且可以提供运行时电源管理服务。在至少一个实

施例中，启动功率和管理处理器可以提供时钟和电压编程，辅助系统低功率状态转换，一个或更多个SoC 1004热和温度传感器管理和/或一个或更多个SoC 1004功率状态管理。在至少一个实施例中，每个温度传感器可以实现为其输出频率与温度成比例的环形振荡器，并且一个或更多个SoC 1004可以使用环形振荡器来检测一个或更多个CPU 1006，一个或更多个GPU1008和/或一个或更多个加速器1014的温度。在至少一个实施例中，如果确定温度超过阈值，则启动和电源管理处理器可以进入温度故障例程，并将一个或更多个SoC 1004置于较低功耗状态和/或将车辆1000置于司机的安全停车图案(例如，使车辆1000安全停车)。

[0218] 在至少一个实施例中，一个或更多个处理器1010可以还包括一组嵌入式处理器，其可以用作音频处理引擎，所述音频处理引擎可以是音频子系统，其能够通过多个接口以及广泛且灵活范围的音频I/O接口为硬件提供对多通道音频的完全硬件支持。在至少一个实施例中，音频处理引擎是专用处理器核心，其具有带专用RAM的数字信号处理器。

[0219] 在至少一个实施例中，一个或更多个处理器1010可以还包括始终在线的处理器引擎，该引擎可以提供必要的硬件特征以支持低功率传感器管理和唤醒用例。在至少一个实施例中，始终在线的处理器引擎上的处理器可以包括但不限于处理器核心、紧密耦合的RAM、支持外围设备(例如，定时器和中断控制器)、各种I/O控制器外围设备以及路由逻辑。

[0220] 在至少一个实施例中，一个或更多个处理器1010可以还包括安全集群引擎，该安全集群引擎包括但不限于用于处理汽车应用程序的安全管理的专用处理器子系统。在至少一个实施例中，安全集群引擎可以包括但不限于两个或更多个处理器核心、紧密耦合的RAM、支持外围设备(例如，定时器、中断控制器等)和/或路由逻辑。在安全模式下，在至少一个实施例中，两个或更多个核心可以以锁步模式操作，并且可以用作具有用以检测其操作之间的任何差异的比较逻辑的单个核心。在至少一个实施例中，一个或更多个处理器1010可以还包括实时相机引擎，该实时相机引擎可以包括但不限于用于处理实时相机管理的专用处理器子系统。在至少一个实施例中，一个或更多个处理器1010可以还包括高动态范围信号处理器，该高动态范围信号处理器可以包括但不限于图像信号处理器，该图像信号处理器是作为相机处理管线的一部分的硬件引擎。

[0221] 在至少一个实施例中，一个或更多个处理器1010可以包括视频图像合成器，该视频图像合成器可以是处理块(例如，在微处理器上实现)，该处理块实现视频回放应用程序产生最终的视频所需要的视频后处理功能，以产生用于播放器窗口的最终图像。在至少一个实施例中，视频图像合成器可以在一个或更多个广角相机1070、一个或更多个环绕相机1074和/或一个或更多个舱内监控相机传感器上执行透镜畸变校正。在至少一个实施例中，优选地，由在SoC 1004的另一实例上运行的神经网络来监控舱室内监控相机传感器，该神经网络被配置为识别舱室事件并相应地做出响应。在至少一个实施例中，舱室内系统可以执行但不限于唇读以激活蜂窝服务和拨打电话、指示电子邮件、改变车辆的目的地、激活或改变车辆的信息娱乐系统和设置、或者提供语音激活的网上冲浪。在至少一个实施例中，当车辆以自主模式运行时，某些功能对于驾驶员是可用的，否则将其禁用。

[0222] 在至少一个实施例中，视频图像合成器可以包括用于同时空间和时间降噪的增强的时间降噪。例如，在至少一个实施例中，在运动发生在视频中的情况下，降噪适当地对空间信息加权，从而减小由相邻帧提供的信息的权重。在至少一个实施例中，在图像或图像的一部分不包括运动的情况下，由视频图像合成器执行的时间降噪可以使用来自先前图像的

信息来降低当前图像中的噪声。

[0223] 在至少一个实施例中，视频图像合成器还可以配置为对输入的立体透镜帧执行立体校正。在至少一个实施例中，当使用操作系统桌面时，视频图像合成器还可以用于用户接口合成，并且不需要一个或更多个GPU 1008来连续渲染新表面。在至少一个实施例中，当对一个或更多个GPU 1008供电并使其活跃地进行3D渲染时，视频图像合成器可以被用于卸载一个或更多个GPU 1008以改善性能和响应性。

[0224] 在至少一个实施例中，SoC 1004中的一个或更多个SoC可以还包括用于从相机接收视频和输入的移动工业处理器接口(“MIPI”)相机串行接口、高速接口和/或可用于相机和相关像素输入功能的视频输入块。在至少一个实施例中，一个或更多个SoC 1004可以还包括输入/输出控制器，该输入/输出控制器可以由软件控制并且可以被用于接收未提交给特定角色的I/O信号。

[0225] 在至少一个实施例中，SoC 1004中的一个或更多个SoC可以还包括广泛的外围接口，以使得能够与外围设备、音频编码器/解码器(“编解码器”)，电源管理和/或其他设备通信。在至少一个实施例中，一个或更多个SoC 1004可用于处理来自(例如，通过千兆位多媒体串行链路和以太网通道连接)相机、传感器(例如，一个或更多个LIDAR传感器1064，一个或更多个RADAR传感器1060等，其可以通过以太网通道连接)的数据，来自总线1002的数据(例如，车辆1000的速度、方向盘位置等)，来自一个或更多个GNSS传感器1058的数据(例如，通过以太网总线或CAN总线连接)等。在至少一个实施例中，SoC 1004中的一个或更多个SoC可以还包括专用高性能海量存储控制器，其可以包括它们自己的DMA引擎，并且可以用于使一个或更多个CPU 1006摆脱常规数据管理任务。

[0226] 在至少一个实施例中，一个或更多个SoC 1004可以是具有灵活架构的端到端平台，其跨越自动化级别3-5级，从而提供利用并有效使用计算机视觉和ADAS技术来实现多样性和冗余的综合的功能安全架构，其提供了可提供灵活、可靠的驾驶软件堆栈以及深度学习工具的平台。在至少一个实施例中，一个或更多个SoC 1004可以比常规系统更快、更可靠，并且甚至在能量效率和空间效率上也更高。例如，在至少一个实施例中，一个或更多个加速器1014当与一个或更多个CPU 1006、一个或更多个GPU 1008以及一个或更多个数据存储装置1016结合时，可以提供用于3-5级自动驾驶车辆的快速、有效的平台。

[0227] 在至少一个实施例中，计算机视觉算法可以在CPU上执行，CPU可以使用高级编程语言(例如C)配置为在多种视觉数据上执行多种处理算法。然而，在至少一个实施例中，CPU通常不能满足许多计算机视觉应用程序的性能要求，例如与执行时间和功耗有关的性能要求。在至少一个实施例中，许多CPU不能实时执行复杂的对象检测算法，该算法被用于车载ADAS应用程序和实际3-5级自动驾驶车辆中。

[0228] 本文所述的实施例允许同时和/或序列地执行多个神经网络，并且允许将结果结合在一起以实现3-5级自动驾驶功能。例如，在至少一个实施例中，在DLA或离散GPU(例如，一个或更多个GPU 1020)上执行的CNN可包括文本和单词识别，从而允许超级计算机读取和理解交通标志，包括神经网络尚未被专门训练的标志。在至少一个实施例中，DLA还可包括神经网络，该神经网络能够识别、解释并提供符号的语义理解，并将该语义理解传递给在CPU Complex上运行的路径规划模块。

[0229] 在至少一个实施例中，对于3、4或5级的驱动，可以同时运行多个神经网络。例如，

在至少一个实施例中，由“警告标志声明：闪烁的灯指示结冰状况(Caution:flashing lights indicate icy conditions)”连通电灯一起组成的警告标志可以由多个神经网络独立地或共同地解释。在至少一个实施例中，可以通过第一部署的神经网络(例如，已经训练的神经网络)将该警告标志本身识别为交通标志，可以通过第二部署的神经网络来解释文本“闪烁的灯指示结冰状况(flashing lights indicate icy conditions)”，其通知车辆的路径规划软件(最好在CPU Complex上执行)：当检测到闪烁的灯光时，就会存在结冰状况。在至少一个实施例中，可以通过在多个帧上操作第三部署的神经网络来识别闪烁的灯，向车辆的路径规划软件通知存在(或不存在)闪烁的灯。在至少一个实施例中，所有三个神经网络可以同时运行，例如在DLA内和/或在一个或更多个GPU 1008上。

[0230] 在至少一个实施例中，用于面部识别和车辆所有者识别的CNN可以使用来自相机传感器的数据来识别授权驾驶员和/或车辆1000的所有者的存在。在至少一个实施例中，当所有者接近驾驶员门并打开灯时，常开传感器处理器引擎可用于解锁车辆，并且，在安全模式下，当所有者离开该车辆时，可用于禁用该车辆。以此方式，一个或更多个SoC 1004提供防止盗窃和/或劫车的保障。

[0231] 在至少一个实施例中，用于紧急车辆检测和识别的CNN可以使用来自麦克风1096的数据来检测和识别紧急车辆警报器。在至少一个实施例中，一个或更多个SoC 1004使用CNN来对环境和城市声音进行分类，以及对视觉数据进行分类。在至少一个实施例中，训练在DLA上运行的CNN以识别紧急车辆的相对接近速度(例如，通过使用多普勒效应)。在至少一个实施例中，还可以训练CNN来识别针对车辆正在运行的区域的紧急车辆，如一个或更多个GNSS传感器1058所识别。在至少一个实施例中，当在欧洲运行时，CNN将寻求检测欧洲警报器，而在北美洲时，CNN将寻求仅识别北美警报器。在至少一个实施例中，一旦检测到紧急车辆，就可以在一个或更多个超声波传感器1062的辅助下使用控制程序来执行紧急车辆安全例程、减速车辆、将车辆驶至路边、停车、和/或使车辆闲置，直到紧急车辆通过。

[0232] 在至少一个实施例中，车辆1000可以包括一个或更多个CPU 1018(例如，一个或更多个离散CPU或一个或更多个dCPU)，其可以经由高速互连(例如PCIe)耦合到一个或更多个SoC 1004。在至少一个实施例中，一个或更多个CPU 1018可以包括X86处理器，例如一个或更多个CPU 1018可用于执行各种功能中的任何功能，例如包括在ADAS传感器和一个或更多个SoC 1004之间潜在的仲裁不一致的结果，和/或一个或更多个监控控制器1036的状态和健康和/或片上信息系统(“信息SoC”)1030。

[0233] 在至少一个实施例中，车辆1000可以包括一个或更多个GPU 1020(例如，一个或更多个离散GPU或一个或更多个dGPU)，其可以经由高速互连(例如NVIDIA的NVLINK通道)耦合到一个或更多个SoC 1004。在至少一个实施例中，一个或更多个GPU 1020可以提供附加的人工智能功能，例如通过执行冗余和/或不同的神经网络，并且可以至少部分地基于来自车辆1000的传感器的输入(例如，传感器数据)来用于训练和/或更新神经网络。

[0234] 在至少一个实施例中，车辆1000可以还包括网络接口1024，其可以包括但不限于一个或更多个无线天线1026(例如，用于不同通信协议的一个或更多个无线天线，诸如蜂窝天线、蓝牙天线等)。在至少一个实施例中，网络接口1024可以用于使能通过互联网云服务(例如，采用服务器和/或其他网络设备)与其他车辆和/或计算设备(例如乘客的客户端设备)的无线连接。在至少一个实施例中，为了与其他车辆通信，可以在车辆1000和另一车辆

之间建立直接链路和/或可以建立间接链路(例如,通过网络和互联网)。在至少一个实施例中,可以使用车辆到车辆的通信链路来提供直接链路。在至少一个实施例中,车辆到车辆的通信链路可以向车辆1000提供关于车辆1000附近的车辆的信息(例如,车辆1000前面、侧面和/或后面的车辆)。在至少一个实施例中,该前述功能可以是车辆1000的协作自适应巡航控制功能的一部分。

[0235] 在至少一个实施例中,网络接口1024可以包括SoC,其提供调制和解调功能并使一个或更多个控制器1036能够通过无线网络进行通信。在至少一个实施例中,网络接口1024可以包括射频前端,用于从基带到射频的上转换以及从射频到基带的下转换。在至少一个实施例中,可以以任何技术上可行的方式执行频率转换。例如,可以通过公知的过程和/或使用超外差过程来执行频率转换。在至少一个实施例中,射频前端功能可以由单独的芯片提供。在至少一个实施例中,网络接口可以包括用于通过LTE、WCDMA、UMTS、GSM、CDMA2000、蓝牙、蓝牙LE、Wi-Fi、Z-Wave、ZigBee、LoRaWAN和/或其他无线协议进行通信的无线功能。

[0236] 在至少一个实施例中,车辆1000可以还包括一个或更多个数据存储1028,其可以包括但不限于片外(例如,一个或更多个SoC 1004)存储。在至少一个实施例中,一个或更多个数据存储1028可以包括但不限于一个或更多个存储元件,包括RAM、SRAM、动态随机存取存储器(“DRAM”)、视频随机存取存储器(“VRAM”)、闪存、硬盘和/或其他组件和/或可以存储至少一位数据的设备。

[0237] 在至少一个实施例中,车辆1000可以还包括一个或更多个GNSS传感器1058(例如,GPS和/或辅助GPS传感器),以辅助地图绘制、感知、占用光栅生成和/或路径规划功能。在至少一个实施例中,可以使用任何数量的GNSS传感器1058,包括例如但不限于使用具有以太网的USB连接器连接到串行接口(例如RS-232)桥的GPS。

[0238] 在至少一个实施例中,车辆1000可以还包括一个或更多个RADAR传感器1060。在至少一个实施例中,一个或更多个RADAR传感器1060可以由车辆1000用于远程车辆检测,即使在黑暗和/或恶劣天气条件下。在至少一个实施例中,RADAR功能安全等级可以是ASIL B。在至少一个实施例中,一个或更多个RADAR传感器1060可以使用CAN总线和/或总线1002(例如,以传输由一个或更多个RADAR传感器1060生成的数据)来进行控制和访问对象跟踪数据,在某些示例中可以访问以太网通道以访问原始数据。在至少一个实施例中,可以使用各种各样的RADAR传感器类型。例如但不限于,RADAR传感器1060中的一个或更多个传感器可适合于前、后和侧面RADAR使用。在至少一个实施例中,一个或更多个RADAR传感器1060是脉冲多普勒RADAR传感器。

[0239] 在至少一个实施例中,一个或更多个RADAR传感器1060可以包括不同的配置,例如具有窄视野的远程、具有宽事业的近程、近程侧面覆盖等。在至少一个实施例中,远程RADAR可以用于自适应巡航控制功能。在至少一个实施例中,远程RADAR系统可以提供通过两次或更多次独立扫描(例如在250m范围内)实现的宽广的视野。在至少一个实施例中,一个或更多个RADAR传感器1060可以帮助在静态对象和运动对象之间区分,并且可以被ADAS系统1038用于紧急制动辅助和向前碰撞警告。在至少一个实施例中,包括在远程RADAR系统中的一个或更多个传感器1060可以包括但不限于具有多个(例如六个或更多个)固定RADAR天线以及高速CAN和FlexRay接口的单基地多模式RADAR。在至少一个实施例中,具有六个天线、中央四个天线可以创建聚焦的波束图,该波束图设计为以较高的速度记录车辆1000的周围

环境,而相邻车道的交通干扰最小。在至少一个实施例中,其他两个天线可以扩大视野,从而可以快速检测进入或离开车道的车辆1000。

[0240] 在至少一个实施例中,作为示例,中程RADAR系统可包括例如高达160m(前)或80m(后)的范围,以及高达42度(前)或150度(后)的视野。在至少一个实施例中,短程RADAR系统可以包括但不限于设计成安装在后保险杠的两端的任意数量的RADAR传感器1060。当安装在后保险杠的两端时,在至少一个实施例中,RADAR传感器系统可以产生两个光束,该两个光束不断地监测车辆后部方向和附近的盲点。在至少一个实施例中,短程RADAR系统可以在ADAS系统1038中用于盲点检测和/或车道改变辅助。

[0241] 在至少一个实施例中,车辆1000可以还包括一个或更多个超声传感器1062。在至少一个实施例中,可以定位在车辆1000的前、后和/或侧面位置的一个或更多个超声传感器1062可以用于停车辅助和/或创建和更新占用光栅。在至少一个实施例中,可以使用各种各样的超声传感器1062,并且可以将不同的超声传感器1062用于不同的检测范围(例如2.5m、4m)。在至少一个实施例中,超声传感器1062可以在ASIL B的功能安全等级下操作。

[0242] 在至少一个实施例中,车辆1000可以包括一个或更多个LIDAR传感器1064。在至少一个实施例中,一个或更多个LIDAR传感器1064可以用于对象和行人检测、紧急制动、避免碰撞和/或其他功能。在至少一个实施例中,一个或更多个LIDAR传感器1064可以在功能安全等级ASIL B操作。在至少一个实施例中,车辆1000可以包括可以使用以太网通道的多个(例如,两个、四个、六个等)LIDAR传感器1064(例如,将数据提供给千兆以太网交换机)。

[0243] 在至少一个实施例中,一个或更多个LIDAR传感器1064可能能够提供针对360度视野的对象及其距离的列表。在至少一个实施例中,市售的一个或更多个LIDAR传感器1064例如可以具有大约100m的广告范围,具有2cm-3cm的精度,并且支持100Mbps的以太网连接。在至少一个实施例中,可以使用一个或更多个非突出的LIDAR传感器。在这样的实施例中,一个或更多个LIDAR传感器1064可以包括可以嵌入到车辆1000的前、后、侧面和/或拐角位置中的小型设备。在至少一个实施例中,一个或更多个LIDAR传感器1064,在这样的实施例中,即使对于低反射率的对象,也可以提供高达120度的水平视野和35度的垂直视野,并且具有200m的范围。在至少一个实施例中,可将前向一个或更多个LIDAR传感器1064配置为用于45度至105度之间的水平视野。

[0244] 在至少一个实施例中,也可以使用LIDAR技术(诸如3D闪光LIDAR)。在至少一个实施例中,3D闪光LIDAR使用激光闪光作为传输源,以照亮车辆1000周围大约200m。在至少一个实施例中,闪光LIDAR单元包括但不限于接收器,该接收器记录激光脉冲传播时间和每个像素上的反射光,该像素又对应于从车辆1000到对象的范围。在至少一个实施例中,闪光LIDAR可以允许利用每个激光闪光来生成周围环境的高度准确且无失真的图像。在至少一个实施例中,可以部署四个闪光LIDAR传感器,在车辆1000的每一侧部署一个传感器。在至少一个实施例中,3D闪光LIDAR系统包括但不限于除了风扇(例如非扫描LIDAR设备)以外没有移动部件的固态3D视线阵列LIDAR相机。在至少一个实施例中,闪光LIDAR设备可以每帧使用5纳秒的I类(人眼安全)激光脉冲,并且可以捕获反射激光,作为3D测距点云和共同登记的强度数据。

[0245] 在至少一个实施例中,车辆1000还可包括一个或更多个IMU传感器1066。在至少一个实施例中,一个或更多个IMU传感器1066可位于车辆1000的后轴中心。在至少一个实施例

中，一个或更多个IMU传感器1066可以包括，例如但不限于，一个或更多个加速度计、一个或更多个磁力计、一个或更多个陀螺仪、一个磁罗盘、多个磁罗盘和/或其他传感器类型。在至少一个实施例中，例如在六轴应用程序中，一个或更多个IMU传感器1066可以包括但不限于加速度计和陀螺仪。在至少一个实施例中，例如在九轴应用程序中，一个或更多个IMU传感器1066可以包括但不限于加速度计、陀螺仪和磁力计。

[0246] 在至少一个实施例中，一个或更多个IMU传感器1066可以实现为结合了微机电系统(“MEMS”)惯性传感器，高灵敏度GPS接收器和先进的卡尔曼滤波算法的微型高性能GPS辅助惯性导航系统(“GPS/INS”)，以提供位置、速度和姿态的估算；在至少一个实施例中，一个或更多个IMU传感器1066可使车辆1000估算航向而无需来自磁传感器通过直接观测和关联从GPS到一个或更多个IMU传感器1066的速度变化来实现的输入。在至少一个实施例中，一个或更多个IMU传感器1066和一个或更多个GNSS传感器1058可以组合在单个集成单元中。

[0247] 在至少一个实施例中，车辆1000可以包括放置在车辆1000内和/或周围的一个或更多个麦克风1096。在至少一个实施例中，此外，一个或更多个麦克风1096可以用于紧急车辆检测和识别。

[0248] 在至少一个实施例中，车辆1000可以还包括任何数量的相机类型，包括一个或更多个立体相机1068、一个或更多个广角相机1070、一个或更多个红外相机1072、一个或更多个环绕相机1074、一个或更多个远程相机1098、一个或更多个中程相机1076和/或其他相机类型。在至少一个实施例中，相机可用于捕获车辆1000的整个外围周围的图像数据。在至少一个实施例中，所使用的相机的类型取决于车辆1000。在至少一个实施例中，相机类型的任何组合可以是用于在车辆1000周围提供必要覆盖范围。在至少一个实施例中，部署的相机的数量可以根据实施例而不同。例如，在至少一个实施例中，车辆1000可以包括六个相机、七个相机、十个相机、十二个相机或其他数量的相机。在至少一个实施例中，相机可以作为示例但不限于支持千兆位多媒体串行链路(“GMSL”)和/或千兆位以太网通信。在至少一个实施例中，本文先前参照图10A和图10B可以更详细地描述了每个相机。

[0249] 在至少一个实施例中，车辆1000可以还包括一个或更多个振动传感器1042。在至少一个实施例中，一个或更多个振动传感器1042可以测量车辆1000的部件(例如，轴)的振动。例如，在至少一个实施例中，振动的变化可以指示路面的变化。在至少一个实施例中，当使用两个或更多个振动传感器1042时，振动之间的差异可以用于确定路面的摩擦或打滑(例如，当在动力驱动轴和自由旋转轴之间存在振动差异时)。

[0250] 在至少一个实施例中，车辆1000可以包括ADAS系统1038。在至少一个实施例中，ADAS系统1038可以包括但不限于SoC。在至少一个实施例中，ADAS系统1038可以包括但不限于任何数量的自主/自适应/自动巡航控制(“ACC”)系统、协作自适应巡航控制(“CACC”)系统、前撞警告(“FCW”)系统、自动紧急制动(“AEB”)系统、车道偏离警告(“LDW”)系统、车道保持辅助(“LKA”)系统、盲区警告(“BSW”)系统、后方交叉交通警告(“RCTW”)系统、碰撞警告(“CW”)系统、车道对中(“LC”)系统和/或其他系统、特征和/或功能及其组合。

[0251] 在至少一个实施例中，ACC系统可以使用一个或更多个RADAR传感器1060、一个或更多个LIDAR传感器1064和/或任何数量的相机。在至少一个实施例中，ACC系统可以包括纵向ACC系统和/或横向ACC系统。在至少一个实施例中，纵向ACC系统监控并控制到紧邻车辆1000的另一车辆的距离，并自动调节车辆1000的速度以保持与前方车辆的安全距离。在至

少一个实施例中，横向ACC系统执行距离保持，并在需要时建议车辆1000改变车道。在至少一个实施例中，横向ACC与其他ADAS应用程序有关，例如LC和CW。

[0252] 在至少一个实施例中，CACC系统使用来自其他车辆的信息，该信息可以经由网络接口1024和/或一个或更多个无线天线1026从其他车辆接收经由无线链路或者间接经由网络连接(例如，经由互联网)接收。在至少一个实施例中，直接链路可以由车辆到车辆(“V2V”)的通信链路提供，而间接链路可以由基础设施到车辆(“I2V”)的通信链路提供。通常，V2V通信提供关于紧接在前的车辆(例如，紧接在车辆1000之前并与之在同一车道上的车辆)的信息，而I2V通信提供关于更前方交通的信息。在至少一个实施例中，CACC系统可以包括I2V和V2V信息源之一或两者。在至少一个实施例中，在给定车辆1000之前的车辆的信息的情况下，CACC系统可以更可靠，并且具有改善交通流的平滑度并减少道路拥堵的潜力。

[0253] 在至少一个实施例中，FCW系统被设计成警告驾驶员危险，以便该驾驶员可以采取纠正措施。在至少一个实施例中，FCW系统使用前向相机和/或一个或更多个RADAR传感器1060，其耦合至专用处理器、DSP、FPGA和/或ASIC，其电耦合至提供驾驶员反馈，例如显示器、扬声器和/或振动组件。在至少一个实施例中，FCW系统可以提供警告，例如以声音、视觉警告，振动和/或快速制动脉冲的形式。

[0254] 在至少一个实施例中，AEB系统检测到与另一车辆或其他对象的即将发生的向前碰撞，并且如果驾驶员在指定的时间或距离参数内未采取纠正措施，则可以自动施加制动。在至少一个实施例中，AEB系统可以使用耦合到专用处理器、DSP、FPGA和/或ASIC的一个或更多个前向相机和/或一个或更多个RADAR传感器1060。在至少一个实施例中，当AEB系统检测到危险时，它通常首先警告驾驶员采取纠正措施以避免碰撞，并且，如果该驾驶员没有采取纠正措施，则该AEB系统可以自动施加制动器以试图防止或至少减轻预测碰撞的影响。在至少一个实施例中，AEB系统可以包括诸如动态制动器支持和/或即将发生碰撞的制动的技术。

[0255] 在至少一个实施例中，当车辆1000越过车道标记时，LDW系统提供视觉、听觉和/或触觉警告，例如方向盘或座椅振动，以警告驾驶员。在至少一个实施例中，当驾驶员诸如通过激活转向信号灯指示有意的车道偏离时，LDW系统不活跃。在至少一个实施例中，LDW系统可以使用耦合到专用处理器、DSP、FPGA和/或ASIC的面向正面的相机，其被电耦合以提供诸如显示器、扬声器和/或振动组件之类的驾驶员反馈。在至少一个实施例中，LKA系统是LDW系统的一种变型。在至少一个实施例中，如果车辆1000开始离开车道，则LKA系统提供转向输入或制动以校正车辆1000。

[0256] 在至少一个实施例中，BSW系统检测并警告汽车盲区中的车辆驾驶员。在至少一个实施例中，BSW系统可以提供视觉、听觉和/或触觉警报，以指示合并或改变车道是不安全的。在至少一个实施例中，当驾驶员使用转向灯时，BSW系统可以提供附加警告。在至少一个实施例中，BSW系统可以使用耦合到专用处理器、DSP、FPGA和/或ASIC的一个或更多个朝后侧的相机和/或一个或更多个RADAR传感器1060，其电耦合到驾驶员反馈，例如显示器、扬声器和/或振动组件。

[0257] 在至少一个实施例中，当在车辆1000倒车时在后相机范围之外检测到对象时，RCTW系统可以提供视觉、听觉和/或触觉通知。在至少一个实施例中，RCTW系统包括AEB系统，以确保应用程序车辆制动器以避免碰撞。在至少一个实施例中，RCTW系统可以使用一个

或更多个面向后方的RADAR传感器1060，其耦合到专用处理器、DSP、FPGA和/或ASIC，其被电耦合以提供诸如显示器、扬声器和/或振动组件之类的驾驶员反馈。

[0258] 在至少一个实施例中，常规的ADAS系统可能易于产生误报结果，这可能使驾驶员烦恼和分散注意力，但通常不是灾难性的，因为常规的ADAS系统会警告驾驶员并允许该驾驶员决定安全状况是否真正存在并采取相应动作。在至少一个实施例中，在结果冲突的情况下，车辆1000本身决定是否听从主计算机或副计算机(例如，控制器1036的第一控制器或第二控制器)的结果。例如，在至少一个实施例中，ADAS系统1038可以是用于将感知信息提供给备份计算机合理性模块的备用和/或辅助计算机。在至少一个实施例中，备用计算机合理性监控器可以在硬件组件上运行冗余的各种软件，以检测感知和动态驾驶任务中的故障。在至少一个实施例中，可以将来自ADAS系统1038的输出提供给监控MCU。在至少一个实施例中，如果来自主计算机的输出和来自辅助计算机的输出冲突，则监督MCU决定如何协调冲突以确保安全操作。

[0259] 在至少一个实施例中，主计算机可以配置为向监督MCU提供置信度分数，以指示该主计算机对所选结果的置信度。在至少一个实施例中，如果该置信度得分超过阈值，则该监督MCU可以遵循该主计算机的指示，而不管该辅助计算机是否提供冲突或不一致的结果。在至少一个实施例中，在置信度得分不满足阈值的情况下，并且在主计算机和辅助计算机指示不同的结果(例如，冲突)的情况下，监督MCU可以在计算机之间仲裁以确定适当的结果。

[0260] 在至少一个实施例中，监督MCU可以配置为运行神经网络，该神经网络被训练和配置为至少部分地基于来自主计算机的输出和来自辅助计算机的输出来确定该辅助计算机提供错误警报的条件。在至少一个实施例中，监督MCU中的神经网络可以学习何时可以信任辅助计算机的输出，以及何时不能信任。例如，在至少一个实施例中，当该辅助计算机是基于RADAR的FCW系统时，该监督MCU中的神经网络可以学习FCW系统何时识别实际上不是危险的金属对象，例如会触发警报的排水格栅或井盖。在至少一个实施例中，当辅助计算机是基于相机的LDW系统时，当存在骑自行车的人或行人并且实际上车道偏离是最安全的操作时，监督MCU中的神经网络可以学会覆盖LDW。在至少一个实施例中，监督MCU可以包括适合于运行具有相关联的存储器的神经网络的DLA或GPU中的至少一个。在至少一个实施例中，监督MCU可以包括和/或被包括为一个或更多个SoC 1004的组件。

[0261] 在至少一个实施例中，ADAS系统1038可以包括使用传统的计算机视觉规则执行ADAS功能的辅助计算机。在至少一个实施例中，该辅助计算机可以使用经典计算机视觉规则(如果-则)，并且监督MCU中的神经网络的存在可以提高可靠性、安全性和性能。例如，在至少一个实施例中，多样化的实现方式和有意的非同一性使得整个系统更加容错，尤其是对于由软件(或软件-硬件接口)功能引起的故障。例如，在至少一个实施例中，如果在主计算机上运行的软件中存在软件漏洞或错误，并且在辅助计算机上运行的不相同的软件代码提供了一致的总体结果，则监督MCU可以更有把握地认为总体结果是正确，并且该主计算机上的软件或硬件中的漏洞不会导致重大错误。

[0262] 在至少一个实施例中，可以将ADAS系统1038的输出输入到主计算机的感知模块和/或主计算机的动态驾驶任务模块中。例如，在至少一个实施例中，如果ADAS系统1038由于正前方的对象而指示向前碰撞警告，则感知块可以在识别对象时使用该信息。在至少一个实施例中，如本文所述，辅助计算机可以具有其自己的神经网络，该神经网络经过训练从

而降低了误报的风险。

[0263] 在至少一个实施例中，车辆1000可以还包括信息娱乐SoC 1030(例如，车载信息娱乐系统(IVI))。尽管被示出和描述为SoC，但是在至少一个实施例中，信息娱乐系统SoC 1030可以不是SoC，并且可以包括但不限于两个或更多个分立组件。在至少一个实施例中，信息娱乐SoC 1030可以包括但不限于硬件和软件的组合，其可以用于提供音频(例如，音乐、个人数字助理、导航指令、新闻、广播等)、视频(例如，电视、电影、流媒体等)、电话(例如，免提通话)、网络连接(例如，LTE、WiFi等)和/或信息服务(例如，导航系统、后停车辅助、无线电数据系统、与车辆相关的信息，例如燃油水平、总覆盖距离、制动燃油水平、油位、车门打开/关闭、空气滤清器信息等)到车辆1000。例如，信息娱乐SoC 1030可以包括收音机、磁盘播放器、导航系统、视频播放器、USB和蓝牙连接、汽车、车载娱乐系统、WiFi、方向盘音频控制、免提语音控制、抬头显示器(“HUD”)、HMI显示器1034、远程信息处理设备、控制面板(例如，用于控制各种组件、特征和/或系统和/或与之交互)和/或其他组件。在至少一个实施例中，信息娱乐SoC 1030可以进一步用于向车辆1000的用户提供信息(例如，视觉和/或听觉的)，诸如来自ADAS系统1038的信息、自动驾驶信息(诸如计划的车辆操纵)、轨迹、周围环境信息(例如，交叉路口信息、车辆信息、道路信息等)和/或其他信息。

[0264] 在至少一个实施例中，信息娱乐SoC 1030可以包括任何数量和类型的GPU功能。在至少一个实施例中，信息娱乐SoC 1030可以通过总线1002与车辆1000的其他设备、系统和/或组件通信。在至少一个实施例中，信息娱乐SoC 1030可以是耦合到监控MCU，使得信息娱乐系统的GPU可以在主控制器1036(例如，车辆1000的主计算机和/或备用计算机)发生故障的情况下执行一些自动驾驶功能。在至少一个实施例中，信息娱乐SoC 1030可以使车辆1000进入司机到安全停止模式，如本文所述。

[0265] 在至少一个实施例中，车辆1000可以还包括仪表板1032(例如，数字仪表板、电子仪表板、数字仪表操纵板等)。在至少一个实施例中，仪表板1032可以包括但不限于控制器和/或超级计算机(例如，离散控制器或超级计算机)。在至少一个实施例中，仪表板1032可以包括但不限于一组仪表的任何数量和组合，例如车速表、燃料水平、油压、转速表、里程表、转弯指示器、换档位置指示器、一个或更多个安全带警告灯、一个或更多个驻车制动警告灯、一个或更多个发动机故障灯、辅助约束系统(例如安全气囊)信息、照明控件、安全系统控件、导航信息等。在某些示例中，信息可能是在信息娱乐SoC 1030和仪表板1032之间显示和/或共享。在至少一个实施例中，仪表板1032可以被包括作为信息娱乐SoC 1030的一部分，反之亦然。

[0266] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在系统图10C中用于至少部分地基于使用神经网络训练操作\神经网络功能和/或架构或本文所述的神经网络用例计算出的权重参数来推理或预测操作。

[0267] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在

至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0268] 图10D是根据至少一个实施例的在基于云的服务器与图10A的自动驾驶车辆1000之间进行通信的系统1076的图。在至少一个实施例中，系统1076可以包括但不限于一个或更多个服务器1078、一个或更多个网络1090以及任何数量和类型的车辆，包括车辆1000。在至少一个实施例中，一个或更多个服务器1078可以包括但不限于，多个GPU 1084(A)-1084(H)(在本文中统称为GPU 1084)、PCIe交换机1082(A)-1082(D)(在本文中统称为PCIe交换机1082)，和/或CPU 1080(A)-1080(B)(在本文中统称为CPU 1080)、GPU 1084、CPU 1080和PCIe交换机1082可以与高速连接线互连，例如但不限于，由NVIDIA开发的NVLink接口1088和/或PCIe连接1086。在至少一个实施例中，GPU 1084通过NVLink和/或NVSwitchSoC连接，GPU 1084和PCIe交换机1082通过PCIe互连连接。尽管示出了八个GPU 1084、两个CPU 1080和四个PCIe交换机1082，但这并不旨在进行限制。在至少一个实施例中，一个或更多个服务器1078中的每一个可以包括但不限于任意数量的GPU 1084、CPU 1080和/或PCIe交换机1082的任何组合。例如，在至少一个实施例中，一个或更多个服务器1078可各自包括八个、十六个、三十二个和/或更多个GPU 1084。

[0269] 在至少一个实施例中，一个或更多个服务器1078可以通过一个或更多个网络1090并从车辆接收表示图像的图像数据，该图像示出了意外的或改变的道路状况，例如最近开始的道路工程。在至少一个实施例中，一个或更多个服务器1078可以通过一个或更多个网络1090并且向车辆传输经更新的等神经网络1092，和/或地图信息1094，包括但不限于关于交通和道路状况的信息。在至少一个实施例中，对地图信息1094的更新可以包括但不限于对HD地图1022的更新，例如关于建筑工地、坑洼、便道、洪水和/或其他障碍物的信息。在至少一个实施例中，神经网络1092和/或地图信息1094可能是由从环境中的任何数量的车辆接收的数据中表示的新训练和/或经验产生的，和/或至少基于在数据中心执行的训练(例如，使用一个或更多个服务器1078和/或其他服务器)。

[0270] 在至少一个实施例中，一个或更多个服务器1078可以用于至少部分地基于训练数据来训练机器学习模型(例如，神经网络)。在至少一个实施例中，训练数据可以由车辆产生，和/或可以在模拟中产生(例如，使用游戏引擎)。在至少一个实施例中，标记任何数量的训练数据(例如，在相关的神经网络受益于监督学习的情况下)和/或经历其他预处理。在至少一个实施例中，没有对任何数量的训练数据进行标记和/或预处理(例如，在相关联的神经网络不需要监督学习的情况下)。在至少一个实施例中，一旦机器学习模型被训练，机器学习模型就可以被车辆使用(例如，通过一个或更多个网络1090传输到车辆，和/或机器学习模型可以被一个或更多个服务器1078使用以远程监控车辆。

[0271] 在至少一个实施例中，一个或更多个服务器1078可以从车辆接收数据并且将数据应用程序于最新的实时神经网络以用于实时智能推理。在至少一个实施例中，一个或更多个服务器1078可以包括由一个或更多个GPU 1084供电的深度学习超级计算机和/或专用AI计算机，例如由NVIDIA开发的DGX和DGX Station机器。然而，在至少一个实施例中，一个或更多个服务器1078可以包括使用CPU供电的数据中心的深度学习基础设施。

[0272] 在至少一个实施例中，一个或更多个服务器1078的深度学习基础结构可能能够进行快速、实时的推理，并且可以使用该能力来评估和验证车辆1000中处理器、软件和/或相

关硬件的健康。例如，在至少一个实施例中，深度学习基础设施可以从车辆1000接收周期性更新，例如车辆1000在该图像序列中所定位的图像序列和/或对象(例如，通过计算机视觉和/或其他机器学习对象分类技术)。在至少一个实施例中，深度学习基础设施可以运行其自己的神经网络以识别对象并将它们与车辆1000所识别的对象进行比较，并且，如果结果不匹配和深度学习基础设施断定车辆1000中的AI正在发生故障，则一个或更多个服务器1078可以将信号发送到车辆1000，以指示车辆1000的故障安全计算机采取控制、通知乘客并完成安全停车操作。

[0273] 在至少一个实施例中，一个或更多个服务器1078可以包括一个或更多个GPU 1084和一个或更多个可编程推理加速器(例如NVIDIA的TensorRT 3设备)。在至少一个实施例中，GPU驱动的服务器和推理加速的组合可以使实时响应成为可能。在至少一个实施例中，例如在性能不太关键的情况下，可以将由CPU、FPGA和其他处理器驱动的服务器用于推理。在至少一个实施例中，硬件结构715用于执行一个或更多个实施例。本文结合图7A和/或图7B提供关于硬件结构715的细节。

[0274] 计算机系统

[0275] 图11是示出根据至少一个实施例示例性计算机系统的框图，该示例性计算机系统可以是具有互连的设备和组件的系统，片上系统(SOC)或它们的某种形成有处理器的组合，该处理器可以包括执行单元以执行指令。在至少一个实施例中，根据本公开，例如本文所述的实施例，计算机系统1100可以包括但不限于组件，例如处理器1102，其执行单元包括逻辑以执行用于过程数据的算法。在至少一个实施例中，计算机系统1100可以包括处理器，例如可从加利福尼亚圣塔克拉拉的英特尔公司(Intel Corporation of Santa Clara, California)获得的PENTIUM®处理器家族、Xeon™、Itanium®、XScale™和/或StrongARM™，Intel® Core™或Intel ®Nervana™微处理器，尽管也可以使用其他系统(包括具有其他微处理器的PC、工程工作站、机顶盒等)。在至少一个实施例中，计算机系统1100可以执行可从华盛顿州雷蒙德市的微软公司(Microsoft Corporation of Redmond,Wash.)获得的WINDOWS操作系统版本，尽管其他操作系统(例如UNIX和Linux)、嵌入式软件和/或图形用户界面也可以使用。

[0276] 实施例可以用在其他设备中，例如手持设备和嵌入式应用程序。手持设备的一些示例包括蜂窝电话、互联网协议(Internet Protocol)设备、数码相机、个人数字助理(“PDA”)和手持PC。在至少一个实施例中，嵌入式应用程序可以包括微控制器、数字信号处理器(“DSP”)、片上系统、网络计算机(“NetPC”)、机顶盒、网络集线器、广域网(“WAN”)交换机，或根据至少一个实施例可以执行一个或更多个指令的任何其他系统。

[0277] 在至少一个实施例中，计算机系统1100可包括但不限于处理器1102，该处理器1102可包括但不限于一个或更多个执行单元1108，以根据本文描述的技术执行机器学习模型训练和/或推理。在至少一个实施例中，计算机系统1100是单处理器台式机或服务器系统，但是在另一实施例中，计算机系统1100可以是多处理器系统。在至少一个实施例中，处理器1102可以包括但不限于复杂指令集计算机(“CISC”)微处理器、精简指令集计算(“RISC”)微处理器、超长指令字(“VLIW”)微处理器、实现指令集组合的处理器，或任何其他处理器设备，例如数字信号处理器。在至少一个实施例中，处理器1102可以耦合到处理器总线1110，该处理器总线1110可以在处理器1102与计算机系统1100中的其他组件之间传输数

据信号。

[0278] 在至少一个实施例中，处理器1102可以包括但不限于1级(“L1”)内部高速缓存存储器(“cache”)1104。在至少一个实施例中，处理器1102可以具有单个内部高速缓存或多级内部缓存。在至少一个实施例中，高速缓存存储器可以驻留在处理器1102的外部。根据特定的实现和需求，其他实施例也可以包括内部和外部高速缓存的组合。在至少一个实施例中，寄存器文件1106可以在各种寄存器中存储不同类型的数据，包括但不限于整数寄存器、浮点寄存器、状态寄存器和指令指针寄存器。

[0279] 在至少一个实施例中，包括但不限于执行整数和浮点运算的逻辑的执行单元1108，其也位于处理器1102中。在至少一个实施例中，处理器1102还可以包括微码(“ucode”)只读存储器(“ROM”)，用于存储某些宏指令的微代码。在至少一个实施例中，执行单元1108可以包括用于处理封装指令集1109的逻辑。在至少一个实施例中，通过将封装指令集1109包括在通用处理器的指令集中，以及要执行指令的相关电路，可以使用处理器1102中的封装数据来执行许多多媒体应用程序使用的操作。在至少一个实施例中，可以通过使用处理器的数据总线的全宽度来在封装的数据上执行操作来加速和更有效地执行许多多媒体应用程序，这可能不需要在该处理器的数据总线上传输较小的数据单元来一次执行一个数据元素的一个或更多个操作。

[0280] 在至少一个实施例中，执行单元1108也可以用在微控制器、嵌入式处理器、图形设备、DSP和其他类型的逻辑电路中。在至少一个实施例中，计算机系统1100可以包括但不限于存储器1120。在至少一个实施例中，存储器1120可以为动态随机存取存储器(“DRAM”)设备、静态随机存取存储器(“SRAM”)设备、闪存设备或另一个存储设备。在至少一个实施例中，存储器1120可以存储由处理器1102可以执行的由数据信号表示的指令1119和/或数据1121。

[0281] 在至少一个实施例中，系统逻辑芯片可以耦合到处理器总线1110和存储器1120。在至少一个实施例中，系统逻辑芯片可以包括但不限于存储器控制器集线器(“MCH”)1116，并且处理器1102可以经由处理器总线1110与MCH 1116通信。在至少一个实施例中，MCH 1116可以提供到存储器1120的高带宽存储器路径1118以用于指令和数据存储以及用于图形命令、数据和纹理的存储。在至少一个实施例中，MCH 1116可以在处理器1102、存储器1120和计算机系统1100中的其他组件之间启动数据信号，并且在处理器总线1110、存储器1120和系统I/O接口1122之间桥接数据信号。在至少一个实施例中，系统逻辑芯片可以提供用于耦合到图形控制器的图形端口。在至少一个实施例中，MCH 1116可以通过高带宽存储器路径1118耦合到存储器1120，并且图形/视频卡1112可以通过加速图形端口(Accelerated Graphics Port)(“AGP”)互连1114耦合到MCH 1116。

[0282] 在至少一个实施例中，计算机系统1100可以使用系统I/O接口1122作为专有集线器接口总线来将MCH 1116耦合到I/O控制器集线器(“ICH”)1130。在至少一个实施例中，ICH 1130可以通过本地I/O总线提供与某些I/O设备的直接连接。在至少一个实施例中，本地I/O总线可以包括但不限于用于将外围设备连接到存储器1120、芯片组和处理器1102的高速I/O总线。示例可以包括但不限于音频控制器1129、固件集线器(“Flash BIOS”)1128、无线收发器1126、数据存储1124、包含用户输入和键盘接口的传统I/O控制器1123、串行扩展端口1127(例如通用串行总线(USB)端口)和网络控制器1134。在至少一个实施例中，数据存储

1124可以包括硬盘驱动器、软盘驱动器、CD-ROM设备、闪存设备或其他大容量存储设备。

[0283] 在至少一个实施例中，图11示出了包括互连的硬件设备或“芯片”的系统，而在其他实施例中，图11可以示出SoC。在至少一个实施例中，图11中示出的设备可以与专有互连、标准化互连(例如，PCIe)或其某种组合互连。在至少一个实施例中，计算机系统1100的一个或更多个组件使用计算快速链路(CXL)互连来互连。

[0284] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在图11的系统中使用，用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0285] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0286] 图12是示出根据至少一个实施例的用于利用处理器1210的电子设备1200的框图。在至少一个实施例中，电子设备1200可以是，例如但不限于，笔记本电脑、塔式服务器、机架服务器、刀片服务器、膝上型计算机、台式机、平板电脑、移动设备、电话、嵌入式计算机或任何其他合适的电子设备。

[0287] 在至少一个实施例中，电子设备1200可以包括但不限于通信地耦合到任何合适数量或种类的组件、外围设备、模块或设备的处理器1210。在至少一个实施例中，处理器1210使用总线或接口耦合，诸如I^2C总线、系统管理总线(“SMBus”)、低引脚数(LPC)总线、串行外围接口(“SPI”)、高清音频(“HDA”)总线、串行高级技术附件(“SATA”)总线、通用串行总线(“USB”)(1、2、3版等)或通用异步接收器/发送器(“UART”)总线。在至少一个实施例中，图12示出了系统，该系统包括互连的硬件设备或“芯片”，而在其他实施例中，图12可以示出示例性SoC。在至少一个实施例中，图12中所示的设备可以与专有互连线、标准化互连(例如，PCIe)或其某种组合互连。在至少一个实施例中，图12的一个或更多个组件使用计算快速链路(CXL)互连线来互连。

[0288] 在至少一个实施例中，图12可以包括显示器1224、触摸屏1225、触摸板1230、近场通信单元(“NFC”)1245、传感器集线器1240、热传感器1246、快速芯片组(“EC”)1235、可信平台模块(“TPM”)1238、BIOS/固件/闪存(“BIOS，FW Flash”)1222、DSP1260、驱动器1220(例如固态磁盘(“SSD”)或硬盘驱动器(“HDD”))、无线局域网单元(“WLAN”)1250、蓝牙单元1252、无线广域网单元(“WWAN”)1256、全球定位系统(GPS)单元1255、相机(“USB 3.0相机”)1254(例如USB 3.0相机)和/或以例如LPDDR3标准实现的低功耗双倍数据速率(“LPDDR”)存储器单元(“LPDDR3”)1215。这些组件可以各自以任何合适的方式实现。

[0289] 在至少一个实施例中，其他组件可以通过本文所述的组件通信地耦合到处理器1210。在至少一个实施例中，加速度计1241、环境光传感器(“ALS”)1242、罗盘1243和陀螺仪1244可以可通信地耦合到传感器集线器1240。在至少一个实施例中，热传感器1239、风扇

1237、键盘1236和触摸板1230可以通信地耦合到EC 1235。在至少一个实施例中，扬声器1263、耳机1264和麦克风(“mic”)1265可以通信地耦合到音频单元(“音频编解码器和D类放大器”)1262，其又可以通信地耦合到DSP 1260。在至少一个实施例中，音频单元1262可以包括例如但不限于音频编码器/解码器(“编解码器”)和D类放大器。在至少一个实施例中，SIM卡(“SIM”)1257可以通信地耦合到WWAN单元1256。在至少一个实施例中，组件(诸如WLAN单元1250和蓝牙单元1252以及WWAN单元1256)可以被实现为下一代形式因素(NGFF)。

[0290] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在系统图15中使用，用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0291] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0292] 图13示出了根据至少一个实施例的计算机系统1300。在至少一个实施例中，计算机系统1300配置为实现贯穿本公开描述的各种过程和方法。

[0293] 在至少一个实施例中，计算机系统1300包括但不限于至少一个中央处理单元(“CPU”)1302，该中央处理单元(“CPU”)1302连接到使用任何合适协议实现的通信总线1310，诸如PCI(“外围设备互联”)、外围组件互连Express(“PCI-Express”)、AGP(“加速图形端口”)、超传输或任何其他总线或点对点通信协议。在至少一个实施例中，计算机系统1300包括但不限于主存储器1304和控制逻辑(例如，实现为硬件、软件或其组合)，并且数据可以采取随机存取存储器(“RAM”)的形式存储在主存储器1304中。在至少一个实施例中，网络接口子系统(“网络接口”)1322提供到其他计算设备和网络的接口，用于使用计算机系统1300接收数据并将数据传输到其他系统。

[0294] 在至少一个实施例中，计算机系统1300在至少一个实施例中包括但不限于输入设备1308、并行处理系统1312和显示设备1306，它们可以使用常规的阴极视线管(“CRT”)、液晶显示器(“LCD”)、发光二极管(“LED”)显示器、等离子显示器或其他合适的显示技术实现。在至少一个实施例中，从输入设备1308(诸如键盘、鼠标、触摸板、麦克风等)接收用户输入。在至少一个实施例中，本文所述模块中的每一个可以位于单个半导体平台上以形成处理系统。

[0295] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在系统图13中使用，以至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文所述的神经网络用例计算出的权重参数来进行推理或预测操作。

[0296] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术

被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0297] 图14示出了根据至少一个实施例的计算机系统1400。在至少一个实施例中，计算机系统1400包括但不限于计算机1410和USB盘1420。在至少一个实施例中，计算机1410可以包括但不限于任何数量和类型的处理器(未示出)和存储器(未示出)。在至少一个实施例中，计算机1410包括但不限于服务器、云实例、膝上型计算机和台式计算机。

[0298] 在至少一个实施例中，USB盘1420包括但不限于处理单元1430、USB接口1440和USB接口逻辑1450。在至少一个实施例中，处理单元1430可以是任何指令执行系统、装置或能够执行指令的设备。在至少一个实施例中，处理单元1430可以包括但不限于任何数量和类型的处理核心(未示出)。在至少一个实施例中，处理单元1430包括专用集成电路(“ASIC”)，该专用集成电路被优化为执行与机器学习相关联的任何数量和类型的操作。例如，在至少一个实施例中，处理单元1430是张量处理单元(“TPC”)，其被优化以执行机器学习推理操作。在至少一个实施例中，处理单元1430是视觉处理单元(“VPU”)，其被优化以执行机器视觉和机器学习推理操作。

[0299] 在至少一个实施例中，USB接口1440可以是任何类型的USB连接器或USB插座。例如，在至少一个实施例中，USB接口1440是用于数据和电源的USB 3.0Type-C插座。在至少一个实施例中，USB接口1440是USB 3.0Type-A连接器。在至少一个实施例中，USB接口逻辑1450可以包括使处理单元1430能够经由USB连接器1440与设备(例如计算机1410)相连接的任何数量和类型的逻辑。

[0300] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在系统图14中使用，至少部分地基于使用神经网络训练操作计算出的权重参数、神经网络函数和/或架构或本文所述的神经网络用例来推理或预测操作。

[0301] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0302] 图15A示出了示例性架构，其中多个GPU 1510(1)-1510(N)通过高速链路1540(1)-1540(N)(例如，总线/点对点互连等)通信地耦合到多个多核心处理器1505(1)-1505(M)。在至少一个实施例中，高速链路1540(1)-1540(N)支持4GB/s、30GB/s、80GB/s或更高的通信吞吐量。在至少一个实施例中，可以使用各种互连协议，包括但不限于PCIe 4.0或5.0以及NVLink 2.0。在各个图中，“N”和“M”表示正整数，其值可因图而异。

[0303] 此外，在一个实施例中，两个或更多个GPU 1510通过高速链路1529(1)-1529(2)互

连,该高速链路可以使用与用于高速链路1540(1)-1540(N)的协议/链路类似或不同的协议/链路来实现。类似地,两个或更多个多核心处理器1505可以通过高速链路1528连接,该高速链路可以是以20GB/s、30GB/s、120GB/s或更高的速度运行的对称多处理器(SMP)总线。可替代地,可以使用类似的协议/链路(例如,通过公共互连结构)来完成图15A中所示的各种系统组件之间的所有通信。

[0304] 在至少一个实施例中,每个多核心处理器1505分别经由存储器互连1526(1)-1526(M)通信地耦合到处理器存储器1501(1)-1501(M),并且每个GPU 1510(1)-1510(N)分别通过GPU存储器互连1550(1)-1550(N)通信地耦合到GPU存储器1520(1)-1520(N)。在至少一个实施例中,存储器互连1526和1550可以利用相似或不同的存储器访问技术。作为示例而非限制,处理器存储器1501(1)-1501(M)和GPU存储器1520可以是易失性存储器,诸如动态随机存取存储器(DRAM)(包括堆叠的DRAM)、图形DDR SDRAM(GDDR)(例如GDDR5、GDDR6),或高带宽存储器(HBM),和/或可以是非易失性存储器,例如3D XPoint或Nano-Ram。在至少一个实施例中,处理器存储器1501的某些部分可以是易失性存储器,而另一部分可以是非易失性存储器(例如,使用两级存储器(2LM)层次结构)。

[0305] 如本文所述,尽管各种多核心处理器1505和GPU 1510可以分别物理地耦合到特定存储器1501、1520,和/或可以实现统一存储器架构,其中虚拟系统地址空间(也称为“有效地址”空间)分布在各个物理存储器之间。例如,处理器存储器1501(1)-1501(M)可以各自包含64GB的系统存储器地址空间,并且GPU存储器1520(1)-1520(N)可以各自包含32GB的系统存储器地址空间,从而当M=2和N=4时,导致总计256GB的可寻址存储器大小。N和M也可能是其他值。

[0306] 图15B示出了根据一个示例性实施例的用于多核心处理器1507和图形加速模块1546之间互连的附加细节。在至少一个实施例中,图形加速模块1546可以包括集成在线路卡上的一个或更多个GPU芯片,该线路卡经由高速链路1540(例如,PCIe总线、NVLink等)耦合到处理器1507。在至少一个实施例中,图形加速模块1546可以选择性地集成在具有处理器1507的封装或芯片上。

[0307] 在至少一个实施例中,处理器1507包括多个核心1560A-1560D,每个核心都具有转换后备缓冲区(“TLB”)1561A-1561D和一个或更多个高速缓存1562A-1562D。在至少一个实施例中,核心1560A-1560D可以包括未示出的各种其他组件,用于执行指令和处理数据。在至少一个实施例中,高速缓存1562A-1562D可以包括级别1(L1)和级别2(L2)高速缓存。此外,一个或更多个共享高速缓存1556可以被包括在高速缓存1562A-1562D中,并且由各组核心1560A-1560D共享。例如,处理器1507的一个实施例包括24个核心,每个核心具有其自己的L1高速缓存,十二个共享的L2高速缓存,和十二个共享的L3高速缓存。在该实施例中,两个相邻核心共享一个或更多个L2和L3高速缓存。在至少一个实施例中,处理器1507和图形加速模块1546与系统存储器1514连接,该系统存储器1514可以包括图15A中的处理器存储器1501(1)-1501(M)。

[0308] 在至少一个实施例中,通过一致性总线1564经由核心间通信为存储在各个高速缓存1562A-1562D、1556和系统存储器1514中的数据和指令维护一致性。在至少一个实施例中,例如,每个高速缓存可以具有与其相关联的高速缓存一致性逻辑/电路,以响应于检测到对特定高速缓存行的读取或写入通过一致性总线1564进行通信。在至少一个实施例中,

通过一致性总线1564实现高速缓存监听协议，以监听(snoop)高速缓存访问。

[0309] 在至少一个实施例中，代理电路1525将图形加速模块1546通信地耦合到一致性总线1564，从而允许图形加速模块1546作为核心1560A-1560D的对等方参与高速缓存一致性协议。特别地，在至少一个实施例中，接口1535通过高速链路1540提供到代理电路1525的连接，并且接口1537将图形加速模块1546连接到高速链路1540。

[0310] 在至少一个实施例中，加速器集成电路1536代表图形加速模块的多个图形处理引擎1531(1)-1531(N)提供高速缓存管理、存储器访问、上下文管理和中断管理服务。在至少一个实施例中，图形处理引擎1531(1)-1531(N)可各自包括单独的图形处理单元(GPU)。在至少一个实施例中，图形处理引擎1531(1)-1531(N)选择性地可以包括GPU内的不同类型的图形处理引擎，诸如图形执行单元、媒体处理引擎(例如，视频编码器/解码器)、采样器和blit引擎。在至少一个实施例中，图形加速模块1546可以是具有多个图形处理引擎1531(1)-1531(N)的GPU，或者图形处理引擎1531(1)-1531(N)可以是集成在通用封装、线路卡或芯片上的各个GPU。

[0311] 在至少一个实施例中，加速器集成电路1536包括存储器管理单元(MMU)1539，用于执行各种存储器管理功能，例如虚拟到物理存储器转换(也称为有效到真实存储器转换)，还包括用于访问系统存储器1514的存储器访问协议。在至少一个实施例中，MMU 1539还可包括转换后备缓冲区(“TLB”)(未示出)，用于高速缓存虚拟/有效到物理/真实地址转换。在至少一个实施例中，高速缓存1538可以存储命令和数据，用于图形处理引擎1531(1)-1531(N)有效地访问。在至少一个实施例中，可能使用获取单元1544，将存储在高速缓存1538和图形存储器1533(1)-1533(M)中的数据与核心高速缓存1562A-1562D、1556和系统存储器1514保持一致。如前所述，可以经由代表高速缓存1538和图形存储器1533(1)-1533(M)的代理电路1525来完成该任务(例如，将与处理器高速缓存1562A-1562D、1556上的高速缓存行的修改/访问有关的更新发送到高速缓存1538，并从高速缓存1538接收更新)。

[0312] 在至少一个实施例中，一组寄存器1545存储由图形处理引擎1531(1)-1531(N)执行的线程的上下文数据，并且上下文管理电路1548管理线程上下文。例如，上下文管理电路1548可以执行保存和恢复操作，以在上下文切换期间保存和恢复各个线程的上下文(例如，其中保存第一线程并且存储第二线程，以便可以由图形处理引擎执行第二线程)。例如，上下文管理电路1548在上下文切换时，可以将当前寄存器值存储到存储器中的(例如，由上下文指针标识的)指定区域。然后，当返回上下文时可以恢复寄存器值。在至少一个实施例中，中断管理电路1547接收并处理从系统设备接收的中断。

[0313] 在至少一个实施例中，MMU 1539将来自图形处理引擎1531的虚拟/有效地址转换为系统存储器1514中的真实/物理地址。在至少一个实施例中，加速器集成电路1536支持多个(例如，4、8、16)图形加速器模块1546和/或其他加速器设备。在至少一个实施例中，图形加速器模块1546可专用于在处理器1507上执行的单个应用程序，或者可以在多个应用程序之间共享。在至少一个实施例中，呈现了虚拟化的图形执行环境，其中图形处理引擎1531(1)-1531(N)的资源与多个应用程序或虚拟机(VM)共享。在至少一个实施例中，可基于处理要求和与VM和/或应用程序相关联的优先级，将资源细分为“切片”，其被分配给不同的VM和/或应用程序。

[0314] 在至少一个实施例中，加速器集成电路1536作为图形加速模块1546的系统的桥来

执行,并提供地址转换和系统存储器高速缓存服务。另外,在至少一个实施例中,加速器集成电路1536可以为主机处理器提供虚拟化设施,以管理图形处理引擎1531(1)-1531(N)的虚拟化、中断和存储器管理。

[0315] 在至少一个实施例中,由于图形处理引擎1531(1)-1531(N)的硬件资源被明确地映射到主机处理器1507看到的真实地址空间,因此任何主机处理器都可以使用有效地址值直接寻址这些资源。在至少一个实施例中,加速器集成电路1536的一个功能是物理分离图形处理引擎1531(1)-1531(N),使得它们在系统看来为独立的单元。

[0316] 在至少一个实施例中,一个或更多个图形存储器1533(1)-1533(M)分别耦合到每个图形处理引擎1531(1)-1531(N),并且N=M。在至少一个实施例中,图形存储器1533(1)-1533(M)存储指令和数据,所述指令和数据由每个图形处理引擎1531(1)-1531(N)处理。在至少一个实施例中,图形存储器1533(1)-1533(M)可以是易失性存储器,例如DRAM(包括堆叠的DRAM)、GDDR存储器(例如,GDDR5,GDDR6)或HBM,和/或可以是非易失性存储器,例如3D XPoint或Nano-Ram。

[0317] 在至少一个实施例中,为了减少高速链路1540上的数据流量,可以使用偏置技术以确保存储在图形存储器1533(1)-1533(M)中的数据是图形处理引擎1531(1)-1531(N)最常使用的,并且最好核心1560A-1560D不使用(至少不经常使用)的数据。类似地,在至少一个实施例中,偏置机制试图将核心(并且优选地不是图形处理引擎1531(-1)-1531(N))需要的数据保持在高速缓存1562A-1562D、1556和系统存储器1514中。

[0318] 图15C示出了另一个示例性实施例,其中加速器集成电路1536被集成在处理器1507内。在该实施例中,图形处理引擎1531(1)-1531(N)经由接口1537和接口1535(同样可以是任何形式的总线或接口协议)通过高速链路1540直接与加速器集成电路1536通信。在至少一个实施例中,加速器集成电路1536可以执行与关于图15B描述的操作类似的操作,但是由于它紧密靠近一致性总线1564和高速缓存1562A-1562D、1556,可能具有更高的吞吐量。在至少一个实施例,加速器集成电路支持不同的编程模型,包括专用进程编程模型(无图形加速模块虚拟化)和共享编程模型(具有虚拟化),所述编程模型可以包括由加速器集成电路1536控制的编程模型和由图形加速模块1546控制的编程模型。

[0319] 在至少一个实施例中,图形处理引擎1531(1)-1531(N)专用于单个操作系统下的单个应用程序或进程。在至少一个实施例中,单个应用程序可以将其他应用程序请求汇聚(funnel)到图形处理引擎1531(1)-1531(N),从而在VM/分区内提供虚拟化。

[0320] 在至少一个实施例中,图形处理引擎1531(1)-1531(N)可以被多个VM/应用程序分区共享。在至少一个实施例中,共享模型可以使用系统管理程序来虚拟化图形处理引擎1531(1)-1531(N),以允许每个操作系统进行访问。在至少一个实施例中,对于没有管理程序的单分区系统,操作系统拥有图形处理引擎1531(1)-1531(N)。在至少一个实施例中,操作系统可以虚拟化图形处理引擎1531(1)-1531(N),以提供对每个进程或应用程序的访问。

[0321] 在至少一个实施例中,图形加速模块1546或个体图形处理引擎1531(1)-1531(N)使用进程句柄来选择进程元素。在至少一个实施例中,进程元素被存储在系统存储器1514中,并且可使用本文所述的有效地址到真实地址转换技术来寻址。在至少一个实施例中,进程句柄可以是特定于实现方式的值,其在向图形处理引擎1531(1)-1531(N)注册其上下文时提供给主机进程(即,调用系统软件以将进程元素添加到进程元素链接列表)。在至少一

个实施例中,进程句柄的较低16位可以是进程元素在进程元素链接列表中的偏移量。

[0322] 图15D示出了示例性加速器集成切片1590。在至少一个实施例中,“切片”包括加速器集成电路1536的处理资源的指定部分。在至少一个实施例中,应用程序是系统存储器1514中的有效地址空间1582,其存储进程元素1583。在至少一个实施例中,响应于来自在处理器1507上执行的应用程序1580的GPU调用1581,存储进程元素1583。在至少一个实施例中,进程元素1583包含相应的应用程序1580的进程状态。在一个实施例中,包含在进程元素1583中的工作描述符(WD)1584可以是由应用程序请求的单个作业,或者可以包含指向作业队列的指针。在至少一个实施例中,WD 1584是指向应用程序的有效地址空间1582中的作业请求队列的指针。

[0323] 在至少一个实施例中,图形加速模块1546和/或各个图形处理引擎1531(1)-1531(N)可以由系统中所有进程或进程子集共享。在至少一个实施例中,可以包括用于设置进程状态并将WD 1584发送到图形加速模块1546以在虚拟化环境中开始作业的基础设施。

[0324] 在至少一个实施例中,专用进程编程模型是特定于实现方式的。在至少一个实施例中,在该模型中,单个进程拥有图形加速模块1546或个体图形处理引擎1531。在至少一个实施例中,当图形加速模块1546由单个进程拥有时,管理程序初始化用于所拥有的分区的加速器集成电路,当指派了图形加速模块1546时,操作系统初始化用于所拥有的进程的加速器集成电路1536。

[0325] 在至少一个实施例中,在操作中,加速器集成切片1590中的WD获取单元1591获取下一个WD 1584,其包括要由图形加速模块1546的一个或更多个图形处理引擎完成的工作的指示。在至少一个实施例中,来自WD 1584的数据可以存储在寄存器1545中,并由MMU 1539、中断管理电路1547和/或上下文管理电路1548使用,如图所示。例如,MMU 1539的一个实施例包括用于访问OS虚拟地址空间1585内的段/页表1586的段/页漫游电路。在至少一个实施例中,中断管理电路1547可以处理从图形加速模块1546接收的中断事件1592。在至少一个实施例中,当执行图形操作时,由图形处理引擎1531(1)-1531(N)生成的有效地址1593被MMU 1539转换为真实地址。

[0326] 在至少一个实施例中,为每个图形处理引擎1531(1)-1531(N)和/或图形加速模块1546复制寄存器1545,并且所述寄存器1545可以由管理程序或操作系统初始化。在至少一个实施例中,这些复制的寄存器中的每一个可以被包括在加速器集成切片1590中。可以由管理程序初始化的示例性寄存器在表1中示出。

[0327]

表1–管理程序初始化的寄存器	
寄存器#	描述
1	切片控制寄存器
2	实地址（RA）调度的进程区域指针
3	权限屏蔽覆写寄存器
4	中断向量表条目偏移量
5	中断向量表条目限制
6	状态寄存器
7	逻辑分区ID
8	实地址（RA）管理程序加速器利用率记录指针
9	存储描述寄存器

[0328] 表2中示出了可由操作系统初始化的示例性寄存器。

[0329]

表2–操作系统初始化的寄存器	
寄存器#	描述
1	进程和线程标识
2	有效地址（EA）上下文保存/恢复指针
3	虚拟地址（VA）加速器利用率记录指针
4	虚拟地址（VA）存储段表指针
5	权限屏蔽
6	工作描述符

[0330] 在至少一个实施例中，每个WD 1584特定于特定的图形加速模块1546和/或图形处理引擎1531(1)-1531(N)。在至少一个实施例中，它包含图形处理引擎1531(1)-1531(N)完成工作所需的所有信息，或者它可以是指向存储器位置的指针，在该存储器位置应用程序已经设置了要完成的工作的命令队列。

[0331] 图15E示出了共享模型的一个示例性实施例的附加细节。该实施例包括管理程序真实地址空间1598，其中存储了进程元素列表1599。在至少一个实施例中，可经由管理程序1596来访问管理程序实地址空间1598，所述管理程序1596虚拟化用于操作系统1595的图形加速模块引擎。

[0332] 在至少一个实施例中，共享编程模型允许来自系统中全部分区或分区子集的全部进程或进程子集使用图形加速模块1546。在至少一个实施例中，存在两种编程模型，其中图形加速模块1546由多个进程和分区共享，即，时间切片共享和图形定向共享。

[0333] 在至少一个实施例中，在该模型中，系统管理程序1596拥有图形加速模块1546，并使其功能可用于所有操作系统1595。在至少一个实施例中，对于图形加速模块1546通过系统管理程序1596支持虚拟化，图形加速模块1546可以遵守某些要求，例如(1)应用程序的作业请求必须是自主的(即，不需要在作业之间保持状态)，或者图形加速模块1546必须提供上下文保存和恢复机制，(2)图形加速模块1546保证应用程序的作业请求在指定的时间量内完成，包括任何转换错误，或者图形加速模块1546提供了抢占作业处理的能力，并且(3)在有向共享编程模型中进行操作时，必须确保图形加速模块1546进程之间的公平性。

[0334] 在至少一个实施例中，需要应用程序1580使用图形加速模块类型、工作描述符(WD)、权限屏蔽寄存器(AMR)值和上下文保存/恢复区域指针(CSRP)进行操作系统1595系统调用。在至少一个实施例中，图形加速模块类型描述了用于系统调用的目标加速函数。在至少一个实施例中，图形加速模块类型可以是系统特定的值。在至少一个实施例中，WD是专门为图形加速模块1546格式化的，并且可以采用图形加速模块1546命令、指向用户定义的结构的有效地址指针、指向命令队列的有效地址指针的形式，或描述要由图形加速模块1546完成的工作的任何其他数据结构。

[0335] 在至少一个实施例中，AMR值是用于当前进程的AMR状态。在至少一个实施例中，传递给操作系统的值与设置AMR的应用程序类似。在至少一个实施例中，如果加速器集成电路1536(未示出)和图形加速模块1546的实现不支持用户权限屏蔽覆写寄存器(UAMOR)，则在管理程序调用中传递AMR之前，操作系统可以将当前UAMOR值应用于AMR值。在至少一个实施例中，管理程序1596可以在将AMR放入进程元素1583中之前选择性地应用当前权限屏蔽覆写寄存器(AMOR)值。在至少一个实施例中，CSRP是寄存器1545中的一个，所述寄存器包含应用程序的有效地址空间1582中的区域的有效地址，供图形加速模块1546保存和恢复上下文状态。在至少一个实施例中，如果不需要在作业之间保存状态或者当作业被抢占时，则该指针是可选的。在至少一个实施例中，上下文保存/恢复区域可以是固定的系统存储器。

[0336] 在接收到系统调用时，操作系统1595可以验证应用程序1580已经注册并且被授予使用图形加速模块1546的权限。然后，在至少一个实施例中，操作系统1595使用表3中所示的信息来调用管理程序1596。

[0337]

表3–操作系统到管理程序的调用参数

参数#	描述
1	工作描述符（WD）
2	权限屏蔽寄存器（AMR）值（可能被屏蔽）
3	有效地址（EA）上下文保存/恢复区域指针（CSRP）
4	进程ID（PID）和可选线程ID（TID）
5	虚拟地址（VA）加速器利用率记录指针（AURP）
6	存储段表指针的虚拟地址（SSTP）

[0338]

7	逻辑中断服务号（LISN）

[0339] 在至少一个实施例中，在接收到管理程序调用时，管理程序1596验证操作系统1595已注册并被授予使用图形加速模块1546的权限。然后，在至少一个实施例中，管理程序1596将进程元素1583放入相应的图形加速模块1546类型的进程元素链接列表中。在至少一个实施例中，进程元素可以包括表4中所示的信息。

[0340]

表4–进程元素信息

元素#	描述
1	工作描述符（WD）
2	权限屏蔽寄存器（AMR）值（可能被屏蔽）
3	有效地址（EA）上下文保存/恢复区域指针（CSRP）
4	进程ID（PID）和可选线程ID（TID）
5	虚拟地址（VA）加速器利用率记录指针（AURP）
6	存储段表指针的虚拟地址（SSTP）
7	逻辑中断服务号（LISN）
8	从管理程序调用参数派生的中断向量表
9	状态寄存器（SR）值
10	逻辑分区ID（LPID）
11	真实地址（RA）管理程序加速器利用率记录指针
12	存储描述符寄存器（SDR）

[0341] 在至少一个实施例中，管理程序初始化多个加速器集成切片1590寄存器1545。

[0342] 如图15F所示，在至少一个实施例中，使用统一存储器，所述统一存储器可经由用于访问物理处理器存储器1501(1)-1501(N)和GPU存储器1520(1)-1520(N)的公共虚拟存储器地址空间来寻址。在该实现方式中，在GPU 1510(1)-1510(N)上执行的操作利用相同的虚拟/有效存储器地址空间来访问处理器存储器1501(1)-1501(M)，反之亦然，从而简化了可编程性。在至少一个实施例中，虚拟/有效地址空间的第一部分被分配给处理器存储器1501(1)，第二部分被分配给第二处理器存储器1501(N)，第三部分被分配给GPU存储器1520(1)，以此类推。在至少一个实施例中，整个虚拟/有效存储器空间(有时称为有效地址空间)由此分布在处理器存储器1501和GPU存储器1520的每一个中，从而允许任何处理器或GPU采用映射到任何物理存储器的虚拟地址访问该存储器。

[0343] 在至少一个实施例中，一个或更多个MMU 1539A-1539E内的偏置/一致性管理电路1594A-1594E确保一个或更多个主机处理器(例如，1505)与GPU 1510的高速缓存之间的高速缓存一致性，并实现指示应在其中存储某些类型的数据的物理存储器的偏置技术。在至少一个实施例中，虽然在图15F中示出了偏置/一致性管理电路1594A-1594E的多个实例，但可以在一个或更多个主机处理器1505的MMU内和/或在加速器集成电路1536内实现偏置/一致性电路。

[0344] 一个实施例允许将GPU存储器1520映射为系统存储器的一部分，并使用共享虚拟存储器(SVM)技术进行访问，但不会遭受与完整系统高速缓存一致性相关的性能缺陷。在至少一个实施例中，将GPU存储器1520作为系统存储器来访问而无需繁重的高速缓存一致性开销的能力为GPU卸载提供了有利的操作环境。在至少一个实施例中，该布置允许主机处理器1505的软件设置操作数并访问计算结果，而没有传统的I/O DMA数据拷贝的开销。在至少一个实施例中，这样的传统拷贝包括驱动程序调用、中断和存储器映射I/O(MMIO)访问，相对于简单的存储器访问而言，这些访问效率均较低。在至少一个实施例中，在没有高速缓存一致性开销的情况下访问GPU存储器1520的能力对于卸载的计算的执行时间可能是关键的。在至少一个实施例中，例如，在具有大量流式写入存储器流量的情况下，高速缓存一致性开销可以显著降低GPU 1510所看到的有效写入带宽。在至少一个实施例中，操作数设置的效率、结果访问的效率和GPU计算的效率可能会在确定GPU卸载的有效性方面发挥作用。

[0345] 在至少一个实施例中，GPU偏置和主机处理器偏置的选择由偏置跟踪器数据结构驱动。在至少一个实施例中，例如，可以使用偏置表，所述偏置表可以是页面粒度结构(例如，以存储器页面的粒度来控制)，该页面粒度结构包括每个GPU附加的存储器页面1或2位。在至少一个实施例中，在GPU 1510中具有或不具有偏置高速缓存(例如，用于高速缓存偏置表的频繁/最近使用的条目)的情况下，可以在一个或更多个GPU存储器1520的被盗存储器范围中实现偏置表。替代地，在至少一个实施例中，可以在GPU内维护整个偏置表。

[0346] 在至少一个实施例中，在实际访问GPU存储器之前，访问与对GPU附加存储器1520的每次访问相关联的偏置表条目，从而引起以下操作。在至少一个实施例中，来自GPU 1510的在GPU偏置中找到其页面的本地请求被直接转发到对应的GPU存储器1520。在至少一个实施例中，来自GPU的在主机偏置中找到其页面的本地请求被转发至处理器1505(例如，通过本文所述的高速链路)。在至少一个实施例中，来自处理器1505的在主机处理器偏置中找到所请求页面的请求完成了与正常存储器读取类似的请求。替代地，可以将指向GPU偏置页面的请求转发到GPU 1510。在至少一个实施例中，如果GPU当前不使用页面，则GPU可随后将页面迁移到主机处理器偏置。在至少一个实施例中，页面的偏置状态可以通过基于软件的机制、基于硬件辅助的软件的机制、或者在有限的情况下通过纯粹基于硬件的机制来改变。

[0347] 在至少一个实施例中，一种用于改变偏置状态的机制采用API调用(例如OpenCL)，所述API调用随后调用GPU的设备驱动程序，所述设备驱动程序随后发送消息(或使命令描述符入队)到GPU，引导GPU改变偏置状态，并在某些迁移中在主机中执行高速缓存刷新操作。在至少一个实施例中，高速缓存刷新操作用于从主机处理器1505偏置到GPU偏置的迁移，但是不用于相反的迁移。

[0348] 在至少一个实施例中，高速缓存一致性是通过暂时渲染主机处理器1505无法高速缓存的GPU偏置页面来维护的。在至少一个实施例中，为了访问这些页面，处理器1505可以请求来自GPU 1510的访问，GPU 1510可以或可以不立即授予访问权限。因此，在至少一个实施例中，为了减少处理器1505和GPU 1510之间的通信，确保GPU偏置页面是GPU所需的页面而不是主机处理器1505所需的页面是有益的，反之亦然。

[0349] 一个或更多个硬件结构715用于执行一个或更多个实施例。在本文中可以结合图7A和/或图7B提供关于一个或更多个硬件结构715的细节。

[0350] 图16示出了根据本文所述的各个实施例的示例性集成电路和相关联的图形处理

器,其可以使用一个或更多个IP核心来制造。除了图示之外,在至少一个实施例中可以包括其他逻辑和电路,包括附加的图形处理器/核心、外围接口控制器或通用处理器核心。

[0351] 图16是示出根据至少一个实施例的可使用一个或更多个IP核心制造的芯片集成电路1600上的示例性系统的框图。在至少一个实施例中,集成电路1600包括一个或更多个应用程序处理器1605(例如,CPU)、至少一个图形处理器1610,并且可以另外包括图像处理器1615和/或视频处理器1620,其中任意一个可能是模块化IP核心。在至少一个实施例中,集成电路1600包括外围或总线逻辑,其包括USB控制器1625、UART控制器1630、SPI/SDIO控制器1635和$I^2$2S/$I^2$2C控制器1640。在至少一个实施例中,集成电路1600可以包括显示设备1645耦合到高清多媒体接口(HDMI)控制器1650和移动工业处理器接口(MIPI)显示接口1655中的一个或更多个。在至少一个实施例中,存储可以由闪存子系统1660提供,包括闪存和闪存控制器。在至少一个实施例中,可以经由存储器控制器1665提供存储器接口以用于访问SDRAM或SRAM存储器设备。在至少一个实施例中,一些集成电路还包括嵌入式安全引擎1670。

[0352] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中,推理和/或训练逻辑715可以在集成电路1600中用于至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文描述的神经网络用例计算的权重参数来推理或预测操作。

[0353] 在至少一个实施例中,一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成一个倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0354] 图17A和图17B示出了根据本文所述的各个实施例的示例性集成电路和相关联的图形处理器,其可以使用一个或更多个IP核心来制造。除了图示之外,在至少一个实施例中可以包括其他逻辑和电路,包括附加的图形处理器/核心、外围接口控制器或通用处理器核心。

[0355] 图17A和图17B是示出根据本文描述的实施例的在SoC内使用的示例性图形处理器的框图。图17A示出了根据至少一个实施例的片上系统集成电路的示例性图形处理器1710,片上系统集成电路可以使用一个或更多个IP核心来制造。图17B示出了根据至少一个实施例的片上系统集成电路的另外示例性图形处理器1740,片上系统集成电路可以使用一个或更多个IP核心来制造。在至少一个实施例中,图17A的图形处理器1710是低功耗图形处理器核心。在至少一个实施例中,图17B的图形处理器1740是更高性能的图形处理器核心。在至少一个实施例中,每个图形处理器1710、1740可以是图16的图形处理器1610的变体。

[0356] 在至少一个实施例中,图形处理器1710包括顶点处理器1705和一个或更多个片段处理器1715A-1715N(例如1715A、1715B、1715C、1715D至1715N-1和1715N)。在至少一个实施例中,图形处理器1710可以经由单独的逻辑来执行不同的着色器程序,使得顶点处理器1705被优化以执行针对顶点着色器程序的操作,而一个或更多个片段处理器1715A-1715N

执行片段(例如,像素)着色操作用于片段或像素或着色器程序。在至少一个实施例中,顶点处理器1705执行3D图形管线的顶点处理阶段并生成图元和顶点数据。在至少一个实施例中,一个或更多个片段处理器1715A-1715N使用由顶点处理器1705生成的图元和顶点数据来生成在显示设备上显示的帧缓冲区。在至少一个实施例中,一个或更多个片段处理器1715A-1715N被优化以执行如在OpenGL API中所提供的片段着色器程序,其可以用于执行与在Direct 3D API中所提供的像素着色器程序类似的操作。

[0357] 在至少一个实施例中,图形处理器1710附加地包括一个或更多个存储器管理单元(MMU)1720A-1720B、一个或更多个高速缓存1725A-1725B和一个或更多个电路互连1730A-1730B。在至少一个实施例中,一个或更多个MMU 1720A-1720B提供用于图形处理器1710的虚拟到物理地址的映射,包括用于顶点处理器1705和/或片段处理器1715A-1715N,其可以引用存储在存储器中的顶点或图像/纹理数据,除了存储在一个或更多个高速缓存1725A-1725B中的顶点或图像/纹理数据之外。在至少一个实施例中,一个或更多个MMU 1720A-1720B可以与系统内的其他MMU同步,包括与图16的一个或更多个应用程序处理器1605、图像处理器1615和/或视频处理器1620相关联的一个或更多个MMU,使得每个处理器1605-1620可以参与共享或统一的虚拟存储器系统。在至少一个实施例中,一个或更多个电路互连1730A-1730B使图形处理器1710能够经由SoC的内部总线或经由直接连接与SoC内的其他IP核心相连接。

[0358] 在至少一个实施例中,图形处理器1740包括一个或更多个着色器核心1755A-1755N(例如,1755A、1755B、1755C、1755D、1755E、1755F到1755N-1和1755N),如图17B所示,其提供了统一的着色器核心架构,其中单个核心或类型或核心可以执行所有类型的可编程着色器代码,包括用于实现顶点着色器、片段着色器和/或计算着色器的着色器程序代码。在至少一个实施例中,多个着色器核心可以变化。在至少一个实施例中,图形处理器1740包括核心间任务管理器1745,其充当线程分派器以将执行线程分派给一个或更多个着色器核心1755A-1755N和分块单元1758,以加速基于图块渲染的分块操作,其中在图像空间中细分了场景的渲染操作,例如,以利用场景内的局部空间一致性或优化内部缓存的使用。

[0359] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中,推理和/或训练逻辑715可以在集成电路图17A和/或图17B中用于至少部分地基于使用神经网络训练操作、神经网络函数或架构,或本文所述的神经网络用例计算的权重参数来进行推理或预测操作。

[0360] 在至少一个实施例中,一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0361] 图18A-18B示出了根据本文描述的实施例的附加示例性图形处理器逻辑。在至少一个实施例中,图18A示出了可以包括在图16的图形处理器1610内的图形核心1800,并且在至少一个实施例中,其可以是如图17B所示的统一着色器核心1755A-1755N。图18B示出了在

至少一个实施例中的适用于在多芯片模块上部署的高度并行的通用图形处理单元(“GPGPU”)1830。

[0362] 在至少一个实施例中,图形核心1800包括共享指令高速缓存1802、纹理单元1818和高速缓存/共享存储器1820,它们对于图形核心1800内的执行资源是通用的。在至少一个实施例中,图形核心1800可包括多个切片1801A-1801N或每个核心的分区,并且图形处理器可包括图形核心1800的多个实例。在至少一个实施例中,切片1801A-1801N可包括支持逻辑,所述逻辑包括本地指令高速缓存1804A-1804N、线程调度器1806A-1806N、线程分派器1808A-1808N和一组寄存器1810A-1810N。在至少一个实施例中,切片1801A-1801N可以包括一组附加功能单元(AFU 1812A-1812N)、浮点单元(FPU 1814A-1814N)、整数算术逻辑单元(ALU 1816A-1816N)、地址计算单元(ACU 1813A-1813N)、双精度浮点单元(DPFPU 1815A-1815N)和矩阵处理单元(MPU 1817A-1817N)。

[0363] 在至少一个实施例中,FPU 1814A-1814N可以执行单精度(32位)和半精度(16位)浮点运算,而DPFPU 1815A-1815N则执行双精度(64位)浮点运算点操作。在至少一个实施例中,ALU 1816A-1816N可以以8位、16位和32位精度执行可变精度整数运算,并且可以配置为混合精度运算。在至少一个实施例中,MPU 1817A-1817N还可被配置用于混合精度矩阵运算,包括半精度浮点运算和8位整数运算。在至少一个实施例中,MPU 1817-1817N可以执行各种矩阵运算以加速机器学习应用程序框架,包括使得能够支持加速的通用矩阵到矩阵乘法(GEMM)。在至少一个实施例中,AFU 1812A-1812N可以执行浮点数或整数单元不支持的附加逻辑运算,包括三角运算(例如,正弦,余弦等)。

[0364] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。这里结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中,推理和/或训练逻辑715可在图形核心1800中使用,用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0365] 在至少一个实施例中,一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0366] 图18B示出了在至少一个实施例中的通用处理单元(GPGPU)1830,其可以被配置为使得高度并行的计算操作能够由一组图形处理单元来执行。在至少一个实施例中,GPGPU 1830可以直接链接到GPGPU 1830的其他实例,以创建多GPU集群以提高用于深度神经网络的训练速度。在至少一个实施例中,GPGPU 1830包括主机接口1832,以实现与主机处理器的连接。在至少一个实施例中,主机接口1832是PCI Express接口。在至少一个实施例中,主机接口1832可以是厂商专用的通信接口或通信结构。在至少一个实施例中,GPGPU 1830接收主机处理器的命令,并使用全局调度器1834,以将与那些命令相关联的执行线程分配给一组计算集群1836A-1836H。在至少一个实施例中,计算群集1836A-1836H共享高速缓存存储器1838。在至少一个实施例中,高速缓存存储器1838可以用作计算群集1836A-1836H内的高

速缓存存储器的更高级别的高速缓存。

[0367] 在至少一个实施例中，GPGPU 1830包括存储器1844A-1844B，所述存储器1844A-1844B经由一组存储器控制器1842A-1842B与计算集群1836A-1836H耦合。在至少一个实施例中，存储器1844A-1844B可以包括各种类型的存储器设备，包括动态随机存取存储器(DRAM)或图形随机存取存储器，例如同步图形随机存取存储器(SGRAM)，其包括图形双倍数据速率(GDDR)存储器。

[0368] 在至少一个实施例中，计算集群1836A-1836H每个都包括一组图形核心，例如图18A的图形核心1800，所述图形核心可以包括多种类型的整数和浮点逻辑单元，所述逻辑单元可以在计算机各种精度范围上执行计算操作，包括适用于机器学习计算的精度。例如，在至少一个实施例中，每个计算集群1836A-1836H中的浮点单元的至少一个子集可以被配置为执行16位或32位浮点运算，而浮点单元的不同子集可以配置为执行64位浮点运算。

[0369] 在至少一个实施例中，GPGPU 1830的多个实例可以被配置为用作计算集群。在至少一个实施例中，计算集群1836A-1836H用于同步和数据交换的通信在实施例之间变化。在至少一个实施例中，GPGPU 1830的多个实例通过主机接口1832进行通信。在至少一个实施例中，GPGPU 1830包括I/O集线器1839，所述集线器将GPGPU 1830与GPU链路1840耦合，使得能够直接连接到GPGPU 1830的其他实例。在至少一个实施例中，GPU链路1840耦合到专用GPU到GPU桥，所述桥使得GPGP 1830的多个实例之间能够通信和同步。在至少一个实施例中，GPU链路1840与高速互连耦合，以向其他GPGPU或并行处理器发送和接收数据。在至少一个实施例中，GPGPU 1830的多个实例位于单独的数据处理系统中，并通过可通过主机接口1832访问的网络设备进行通信。在至少一个实施例中，GPU链路1840可被配置为使得能够连接到主机除主机接口1832之外或作为其替代的处理器。

[0370] 在至少一个实施例中，GPGPU 1830可以被配置为训练神经网络。在至少一个实施例中，可以在推理平台内使用GPGPU 1830。在至少一个实施例中，在其中使用GPGPU 1830进行推理的情况下，相对于使用GPGPU 1830训练神经网络时，GPGPU 1830可以包括更少的计算集群1836A-1836H。在至少一个实施例中，与存储器1844A-1844B相关联的存储器技术可以在推理和训练配置之间有所不同，其中更高带宽的存储器技术专用于训练配置。在至少一个实施例中，GPGPU 1830的推理配置可以支持推理特定指令。例如，在至少一个实施例中，推理配置可以提供对一个或更多个8位整数点积指令的支持，该指令可以在部署的神经网络的推理操作期间使用。

[0371] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在GPGPU 1830中使用，用于至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0372] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述

图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0373] 图19示出了根据至少一个实施例的计算机系统1900的框图。在至少一个实施例中，计算机系统1900包括具有一个或更多个处理器1902的处理子系统1901和系统存储器1904，所述系统存储器1904经由可包括存储器集线器1905的互连路径通信。在至少一个实施例中，存储器集线器1905可以是芯片组部件内的单独部件，或者可以集成在一个或更多个处理器1902内。在至少一个实施例中，存储器集线器1905通过通信链路1906与I/O子系统1911耦合。在一个实施例中，I/O子系统1911包括I/O集线器1907，所述I/O集线器可以使计算机系统1900能够接收来自一个或更多个输入设备1908的输入。在至少一个实施例中，I/O集线器1907可以使显示控制器向一个或更多个显示设备1910A提供输出，所述显示控制器可以包括在一个或更多个处理器1902中。在至少一个实施例中，与I/O集线器1907耦合的一个或更多个显示设备1910A可以包括本地，内部或嵌入式显示设备。

[0374] 在至少一个实施例中，处理子系统1901包括经由总线或其他通信链路1913耦合到存储器集线器1905的一个或更多个并行处理器1912中。在至少一个实施例中，通信链路1913可以使用任何一种许多基于标准的通信链路技术或协议，例如但不限于PCI Express，或者可以是特定于供应商的通信接口或通信结构。在至少一个实施例中，一个或更多个并行处理器1912形成计算集中的并行或矢量处理系统，所述系统可以包括大量处理核心和/或处理集群，例如多集成核心(MIC)处理器。在至少一个实施例中，一个或更多个并行处理器1912形成图形处理子系统，所述图形处理子系统可以将像素输出到经由I/O集线器1907耦合的一个或更多个显示设备1910A之一。在至少一个实施例中，并行处理器1912还可以包括显示控制器和显示接口(未示出)，以使得能够直接连接到一个或更多个显示设备1910B。

[0375] 在至少一个实施例中，系统存储单元1914可以连接到I/O集线器1907，以提供用于计算机系统1900的存储机制。在至少一个实施例中，I/O交换机1916可以用于提供一个接口机制，以实现I/O集线器1907与其他组件之间的连接，例如可以集成到平台中的网络适配器1918和/或无线网络适配器1919，以及可以通过一个或更多个附加设备1920添加的各种其他设备。在至少一个实施例中，网络适配器1918可以是以太网适配器或另一有线网络适配器。在至少一个实施例中，无线网络适配器1919可以包括Wi-Fi、蓝牙、近场通信(NFC)中的一个或更多个，或包括一个或更多个无线电设备的其他网络设备。

[0376] 在至少一个实施例中，计算机系统1900可以包括未明确示出的其他组件，所述其他组件包括USB或其他端口连接、光学存储驱动器、视频捕获设备等，所述其他组件也可以连接到I/O集线器1907。在至少一个实施例中，可以使用任何合适的协议(例如基于PCI(外围组件互连)的协议(例如PCI-Express)或其他总线或点对点通信接口和/或协议)来实现互连图19中各个组件的通信路径，例如NV-Link高速互连或互连协议。

[0377] 在至少一个实施例中，一个或更多个并行处理器1912包括为图形和视频处理而优化的电路，所述电路包括例如视频输出电路，并构成图形处理单元(GPU)。在至少一个实施例中，并行处理器1912包括为通用处理而优化的电路。在至少一个实施例中，计算机系统1900的组件可以与单个集成电路上的一个或更多个其他系统元件集成。例如，在至少一个实施例中，并行处理器1912、存储器集线器1905、处理器1902和I/O集线器1907，可以被集成到片上系统(SoC)集成电路中。在至少一个实施例中，计算机系统1900的组件可以被集成到单个封装中，以形成系统级封装(SIP)配置。在至少一个实施例中，计算机系统1900的组件

的至少一部分可以被集成到多芯片模块(MCM)中,所述多芯片模块可以与其他多芯片模块互连到模块化计算机系统中。

[0378] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中,推理和/或训练逻辑715可以在图19的系统1900中使用,用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0379] 在至少一个实施例中,一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0380] 处理器

[0381] 图20A示出了根据至少一个实施例的并行处理器2000。在至少一个实施例中,并行处理器2000的各种组件可以使用一个或更多个集成电路设备来实现,例如可编程处理器、专用集成电路(ASIC)或现场可编程门阵列(FPGA)。在至少一个实施例中,所示的并行处理器2000是根据示例性实施例的图19所示的一个或更多个并行处理器1912的变体。

[0382] 在至少一个实施例中,并行处理器2000包括并行处理单元2002。在至少一个实施例中,并行处理单元2002包括I/O单元2004,其使得能够与其他设备进行通信,包括并行处理单元2002的其他实例。在至少一个实施例中,I/O单元2004可以直接连接到其他设备。在至少一个实施例中,I/O单元2004通过使用集线器或交换机接口(例如,存储器集线器2005)与其他设备连接。在至少一个实施例中,存储器集线器2005与I/O单元2004之间的连接形成通信链路2013。在至少一个实施例中,I/O单元2004与主机接口2006和存储器交叉开关2016连接,其中主机接口2006接收用于执行处理操作的命令,而存储器交叉开关2016接收用于执行存储器操作的命令。

[0383] 在至少一个实施例中,当主机接口2006经由I/O单元2004接收命令缓冲区时,主机接口2006可以引导工作操作以执行那些命令到前端2008。在至少一个实施例中,前端2008与调度器2010耦合,调度器2010配置成将命令或其他工作项分配给处理集群阵列2012。在至少一个实施例中,调度器2010确保在将任务分配给处理集群阵列2012之前,处理集群阵列2012被正确地配置并且处于有效状态。在至少一个实施例中,调度器2010通过在微控制器上执行的固件逻辑来实现。在至少一个实施例中,微控制器实现的调度器2010可配置成以粗粒度和细粒度执行复杂的调度和工作分配操作,从而实现对在处理阵列2012上执行的线程的快速抢占和上下文切换。在至少一个实施例中,主机软件可以证明用于通过多个图形处理路径之一在处理阵列2012上进行调度的工作负载。在至少一个实施例中,工作负载然后可以由包括调度器2010的微控制器内的调度器2010逻辑在处理阵列2012上自动分配。

[0384] 在至少一个实施例中,处理集群阵列2012可以包括多达“N”个处理集群(例如,集群2014A、集群2014B到集群2014N),其中“N”代表一个正整数(可以是与其他图中使用的整数“N”不同的整数)。在至少一个实施例中,处理集群阵列2012的每个集群2014A-2014N可以

执行大量并发线程。在至少一个实施例中，调度器2010可以使用各种调度和/或工作分配算法将工作分配给处理集群阵列2012的集群2014A-2014N，其可以根据每种程序或计算类型产生的工作负载而变化。在至少一个实施例中，调度可以由调度器2010动态地处理，或者可以在配置为由处理集群阵列2012执行的程序逻辑的编译期间部分地由编译器逻辑来辅助。在至少一个实施例中，可将处理集群阵列2012的不同的集群2014A-2014N分配用于处理不同类型的程序或用于执行不同类型的计算。

[0385] 在至少一个实施例中，处理集群阵列2012可以配置成执行各种类型的并行处理操作。在至少一个实施例中，处理集群阵列2012配置成执行通用并行计算操作。例如，在至少一个实施例中，处理集群阵列2012可以包括执行处理任务的逻辑，该处理任务包括对视频和/或音频数据的过滤，执行建模操作，包括物理操作以及执行数据转换。

[0386] 在至少一个实施例中，处理集群阵列2012配置成执行并行图形处理操作。在至少一个实施例中，处理集群阵列2012可以包括附加逻辑以支持这种图形处理操作的执行，包括但不限于执行纹理操作的纹理采样逻辑，以及镶嵌逻辑和其他顶点处理逻辑。在至少一个实施例中，处理集群阵列2012可以配置成执行与图形处理有关的着色器程序，例如但不限于顶点着色器、曲面细分着色器、几何着色器和像素着色器。在至少一个实施例中，并行处理单元2002可以经由I/O单元2004从系统存储器传送数据以进行处理。在至少一个实施例中，在处理期间，可以在处理期间将传送的数据存储到片上存储器(例如，并行处理器存储器2022)，然后将其写回到系统存储器。

[0387] 在至少一个实施例中，当并行处理单元2002用于执行图形处理时，调度器2010可以配置成将处理工作负载划分为近似相等大小的任务，以更好地将图形处理操作分配给处理集群阵列2012的多个集群2014A-2014N。在至少一个实施例中，处理集群阵列2012的部分可以配置成执行不同类型的处理。例如，在至少一个实施例中，第一部分可以配置成执行顶点着色和拓扑生成，第二部分可以配置成执行镶嵌和几何着色，并且第三部分可以配置成执行像素着色或其他屏幕空间操作，以生成用于显示的渲染图像。在至少一个实施例中，可以将由集群2014A-2014N中的一个或更多个产生的中间数据存储在缓冲区中，以允许在集群2014A-2014N之间传输中间数据以进行进一步处理。

[0388] 在至少一个实施例中，处理集群阵列2012可以经由调度器2010接收要执行的处理任务，该调度器2010从前端2008接收定义处理任务的命令。在至少一个实施例中，处理任务可以包括要被处理的数据的索引，例如，表面(补丁)数据、原始数据、顶点数据和/或像素数据，以及状态参数和定义如何处理数据的命令(例如，要执行什么程序)。在至少一个实施例中，调度器2010可以配置成获取与任务相对应的索引，或者可以从前端2008接收索引。在至少一个实施例中，前端2008可以配置成确保在启动由传入命令缓冲区(例如，批缓冲区(batch-buffer)、推送缓冲区等)指定的工作负载之前，处理集群阵列2012配置成有效状态。

[0389] 在至少一个实施例中，并行处理单元2002的一个或更多个实例中的每一个可以与并行处理器存储器2022耦合。在至少一个实施例中，可以经由存储器交叉开关2016访问并行处理器存储器2022，所述存储器交叉开关2016可以接收来自处理集群阵列2012以及I/O单元2004的存储器请求。在至少一个实施例中，存储器交叉开关2016可以经由存储器接口2018访问并行处理器存储器2022。在至少一个实施例中，存储器接口2018可以包括多个分

区单元(例如,分区单元2020A、分区单元2020B到分区单元2020N),其可各自耦合至并行处理器存储器2022的一部分(例如,存储器单元)。在至少一个实施例中,多个分区单元2020A-2020N为配置为等于存储器单元的数量,使得第一分区单元2020A具有对应的第一存储器单元2024A,第二分区单元2020B具有对应的存储器单元2024B,第N分区单元2020N具有对应的第N存储器单元2024N。在至少一个实施例中,分区单元2020A-2020N的数量可以不等于存储器单元的数量。

[0390] 在至少一个实施例中,存储器单元2024A-2024N可以包括各种类型的存储器设备,包括动态随机存取存储器(DRAM)或图形随机存取存储器,例如同步图形随机存取存储器(SGRAM),包括图形双倍数据速率(GDDR)存储器。在至少一个实施例中,存储器单元2024A-2024N还可包括3D堆叠存储器,包括但不限于高带宽存储器(HBM)。在至少一个实施例中,可以跨存储器单元2024A-2024N来存储诸如帧缓冲区或纹理映射的渲染目标,从而允许分区单元2020A-2020N并行地写入每个渲染目标的部分,以有效地使用并行处理器存储器2022的可用带宽。在至少一个实施例中,可以排除并行处理器存储器2022的本地实例,以有利于利用系统存储器与本地高速缓存存储器结合的统一存储器设计。

[0391] 在至少一个实施例中,处理集群阵列2012的集群2014A-2014N中的任何一个都可以处理将被写入并行处理器存储器2022内的任何存储器单元2024A-2024N中的数据。在至少一个实施例中,存储器交叉开关2016可以配置为将每个集群2014A-2014N的输出传输到任何分区单元2020A-2020N或另一个集群2014A-2014N,集群2014A-2014N可以对输出执行其他处理操作。在至少一个实施例中,每个集群2014A-2014N可以通过存储器交叉开关2016与存储器接口2018通信,以从各种外部存储设备读取或写入各种外部存储设备。在至少一个实施例中,存储器交叉开关2016具有到存储器接口2018的连接以与I/O单元2004通信,以及到并行处理器存储器2022的本地实例的连接,从而使不同处理集群2014A-2014N内的处理单元与系统存储器或不是并行处理单元2002本地的其他存储器进行通信。在至少一个实施例中,存储器交叉开关2016可以使用虚拟通道来分离集群2014A-2014N和分区单元2020A-2020N之间的业务流。

[0392] 在至少一个实施例中,可以在单个插入卡上提供并行处理单元2002的多个实例,或者可以将多个插入卡互连。在至少一个实施例中,并行处理单元2002的不同实例可以配置成相互操作,即使不同实例具有不同数量的处理核心,不同数量的本地并行处理器存储器和/或其他配置差异。例如,在至少一个实施例中,并行处理单元2002的一些实例可以包括相对于其他实例而言更高精度的浮点单元。在至少一个实施例中,结合并行处理单元2002或并行处理器2000的一个或更多个实例的系统可以以各种配置和形式因素来实现,包括但不限于台式机、膝上型计算机或手持式个人计算机、服务器、工作站、游戏机和/或嵌入式系统。

[0393] 图20B是根据至少一个实施例的分区单元2020的框图。在至少一个实施例中,分区单元2020是图20A的分区单元2020A-2020N之一的实例。在至少一个实施例中,分区单元2020包括L2高速缓存2021、帧缓冲区接口2025和ROP 2026(光栅操作单元)。在至少一个实施例中,L2高速缓存2021是读/写高速缓存,其配置成执行从存储器交叉开关2016和ROP 2026接收的加载和存储操作。在至少一个实施例中,L2高速缓存2021将读取未命中和紧急回写请求输出到帧缓冲区接口2025以进行处理。在至少一个实施例中,还可以经由帧缓冲

区接口2025将更新发送到帧缓冲区以进行处理。在至少一个实施例中，帧缓冲区接口2025与并行处理器存储器中的存储器单元(诸如图20A的存储器单元2024A-2024N(例如，在并行处理器存储器2022内))之一相互作用。

[0394] 在至少一个实施例中，ROP 2026是一种处理单元，其执行光栅操作，诸如模版、z测试、混合等。在至少一个实施例中，ROP 2026然后输出存储在图形存储器中的处理后的图形数据。在至少一个实施例中，ROP 2026包括压缩逻辑以压缩被写入存储器的深度或颜色数据并解压缩从存储器读取的深度或颜色数据。在至少一个实施例中，压缩逻辑可以是利用多种压缩算法中的一种或更多种的无损压缩逻辑。在至少一个实施例中，ROP 2026执行的压缩的类型可以基于要压缩的数据的统计特性而变化。例如，在至少一个实施例中，基于每图块基础上的深度和颜色数据执行增量颜色压缩。

[0395] 在至少一个实施例中，ROP 2026包括在每个处理集群内(例如，图20A的集群2014A-2014N)，而不是在分区单元2020内。在至少一个实施例中，通过存储器交叉开关2016而不是像素片段数据传输对像素数据的读取和写入请求。在至少一个实施例中，经处理的图形数据可以在显示设备上(诸如图19的一个或更多个显示设备1910之一)显示，由处理器1902路由以供进一步处理，或者由图20A的并行处理器2000内的处理实体之一路由以供进一步处理。

[0396] 图20C是根据至少一个实施例的并行处理单元内的处理集群2014的框图。在至少一个实施例中，处理集群是图20A的处理集群2014A-2014N之一的实例。在至少一个实施例中，处理集群2014可以配置成并行执行许多线程，其中“线程”是指在特定的一组输入数据上执行的特定程序的实例。在至少一个实施例中，单指令多数据(SIMD)指令发布技术用于支持大量线程的并行执行而无需提供多个独立的指令单元。在至少一个实施例中，使用单指令多线程(SIMT)技术来支持并行执行大量一般同步的线程，这使用了公共指令单元，该公共指令单元配置成向每个处理集群内的一组处理引擎发出指令。

[0397] 在至少一个实施例中，可以通过将处理任务分配给SIMT并行处理器的管线管理器2032来控制处理集群2014的操作。在至少一个实施例中，管线管理器2032从图20A的调度器2010接收指令，通过图形多处理器2034和/或纹理单元2036管理这些指令的执行。在至少一个实施例中，图形多处理器2034是SIMT并行处理器的示例性实例。然而，在至少一个实施例中，处理集群2014内可以包括不同架构的各种类型的SIMT并行处理器。在至少一个实施例中，在处理集群2014内可以包括图形多处理器2034的一个或更多个实例。在至少一个实施例中，图形多处理器2034可以处理数据，并且数据交叉开关2040可以用于将处理后的数据分发到多个可能的目的(包括其他着色器单元)地之一。在至少一个实施例中，管线管理器2032可以通过指定要经由数据交叉开关2040分配的处理后的数据的目的地来促进处理后的数据的分配。

[0398] 在至少一个实施例中，处理集群2014内的每个图形多处理器2034可以包括相同的一组功能执行逻辑(例如，算术逻辑单元、加载存储单元等)。在至少一个实施例中，可以以管线方式配置功能执行逻辑，其中可以在先前的指令完成之前发出新的指令。在至少一个实施例中，功能执行逻辑支持多种操作，包括整数和浮点算术、比较操作、布尔运算、移位和各种代数函数的计算。在至少一个实施例中，可以利用相同的功能单元硬件来执行不同的操作，并且可以存在功能单元的任何组合。

[0399] 在至少一个实施例中，传送到处理集群2014的指令构成线程。在至少一个实施例中，跨一组并行处理引擎执行的一组线程是线程组。在至少一个实施例中，线程组在不同的输入数据上执行通用程序。在至少一个实施例中，线程组内的每个线程可被分配给图形多处理器2034内的不同处理引擎。在至少一个实施例中，线程组可包括比图形多处理器2034内的多个处理引擎更少的线程。在至少一个实施例中，当线程组包括的线程数少于处理引擎的数量时，一个或更多个处理引擎在正在处理该线程组的循环期间可能是空闲的。在至少一个实施例中，线程组还可以包括比图形多处理器2034内的多个处理引擎更多的线程。在至少一个实施例中，当线程组包括比图形多处理器2034内的处理引擎的数量更多的线程时，可以在连续的时钟周期内执行处理。在至少一个实施例中，可以在图形多处理器2034上同时执行多个线程组。

[0400] 在至少一个实施例中，图形多处理器2034包括内部高速缓存存储器，以执行加载和存储操作。在至少一个实施例中，图形多处理器2034可以放弃内部高速缓存并使用处理集群2014内的高速缓存存储器(例如，L1高速缓存2048)。在至少一个实施例中，每个图形多处理器2034还可以访问分区单元(例如，图20A的分区单元2020A-2020N)内的L2高速缓存，这些分区单元在所有处理集群2014之间共享并且可以用于在线程之间传输数据。在至少一个实施例中，图形多处理器2034还可以访问片外全局存储器，其可以包括本地并行处理器存储器和/或系统存储器中的一个或更多个。在至少一个实施例中，并行处理单元2002外部的任何存储器都可以用作全局存储器。在至少一个实施例中，处理集群2014包括图形多处理器2034的多个实例，它们可以共享可以存储在L1高速缓存2048中的公共指令和数据。

[0401] 在至少一个实施例中，每个处理集群2014可以包括配置成将虚拟地址映射为物理地址的存储器管理单元(“MMU”)2045。在至少一个实施例中，MMU 2045的一个或更多个实例可以驻留在图20A的存储器接口2018内。在至少一个实施例中，MMU 2045包括一组页表条目(PTE)，其用于将虚拟地址映射到图块的物理地址以及可选地映射到高速缓存行索引。在至少一个实施例中，MMU 2045可以包括地址转换后备缓冲区(TLB)或可以驻留在图形多处理器2034或L1高速缓存2048或处理集群2014内的高速缓存。在至少一个实施例中，处理物理地址以分配表面数据访问局部性，以便在分区单元之间进行有效的请求交织。在至少一个实施例中，高速缓存行索引可以用于确定对高速缓存线的请求是命中还是未命中。

[0402] 在至少一个实施例中，可以配置处理集群2014，使得每个图形多处理器2034耦合到纹理单元2036，以执行纹理映射操作，所述操作确定纹理样本位置、读取纹理数据以及过滤纹理数据。在至少一个实施例中，根据需要从内部纹理L1高速缓存(未示出)或从图形多处理器2034内的L1高速缓存中读取纹理数据，并从L2高速缓存、本地并行处理器存储器或系统存储器中获取纹理数据。在至少一个实施例中，每个图形多处理器2034将处理后的任务输出到数据交叉开关2040，以将处理后的任务提供给另一处理集群2014以进行进一步处理或将处理后的任务存储在L2高速缓存、本地并行处理器存储器、或经由存储器交叉开关2016的系统存储器中。在至少一个实施例中，preROP 2042(光栅前操作单元)配置成从图形多处理器2034接收数据，将数据引导至ROP单元，该ROP单元可以与本文所述的分区单元(例如，图20A的分区单元2020A-2020N)一起定位。在至少一个实施例中，PreROP 2042单元可以执行用于颜色混合的优化、组织像素颜色数据以及执行地址转换。

[0403] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练

操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在图形处理集群2014中用于至少部分地基于使用本文描述的神经网络训练操作、神经网络函数和/或架构或神经网络用例计算的权重参数来进行推理或预测操作。

[0404] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成一个倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0405] 图20D示出了根据至少一个实施例的图形多处理器2034。在至少一个实施例中，图形多处理器2034与处理集群2014的管线管理器2032耦合。在至少一个实施例中，图形多处理器2034具有执行管线，该执行管线包括但不限于指令高速缓存2052、指令单元2054、地址映射单元2056、寄存器文件2058、一个或更多个通用图形处理单元(GPGPU)核心2062和一个或更多个加载/存储单元2066。在至少一个实施例中，GPGPU核心2062和加载/存储单元2066与高速缓存存储器2072和共享存储器2070通过存储器和高速缓存互连2068耦合。

[0406] 在至少一个实施例中，指令高速缓存2052从管线管理器2032接收要执行的指令流。在至少一个实施例中，将指令高速缓存在指令高速缓存2052中并将其分派以供指令单元2054执行。在一个实施例中，指令单元2054可以分派指令作为线程组(例如，线程束)，将线程组的每个线程分配给GPGPU核心2062内的不同执行单元。在至少一个实施例中，指令可以通过在统一地址空间内指定地址来访问任何本地、共享或全局地址空间。在至少一个实施例中，地址映射单元2056可以用于将统一地址空间中的地址转换成可以由加载/存储单元2066访问的不同的存储器地址。

[0407] 在至少一个实施例中，寄存器文件2058为图形多处理器2034的功能单元提供了一组寄存器。在至少一个实施例中，寄存器文件2058为连接到图形多处理器2034的功能单元(例如，GPGPU核心2062、加载/存储单元2066)的数据路径的操作数提供了临时存储。在至少一个实施例中，在每个功能单元之间划分寄存器文件2058，使得为每个功能单元分配寄存器文件2058的专用部分。在至少一个实施例中，寄存器文件2058在图形多处理器2034正在执行的不同线程束之间划分。

[0408] 在至少一个实施例中，GPGPU核心2062可以各自包括用于执行图形多处理器2034的指令的浮点单元(FPU)和/或整数算术逻辑单元(ALU)。在至少一个实施例中，GPGPU核心2062在架构上可以相似或架构可能有所不同。在至少一个实施例中，GPGPU核心2062的第一部分包括单精度FPU和整数ALU，而GPGPU核心的第二部分包括双精度FPU。在至少一个实施例中，FPU可以实现用于浮点算法的IEEE 754-2008标准或启用可变精度浮点算法。在至少一个实施例中，图形多处理器2034可以另外包括一个或更多个固定功能或特殊功能单元，以执行特定功能，诸如复制矩形或像素混合操作。在至少一个实施例中，GPGPU核心2062中的一个或更多个也可以包括固定或特殊功能逻辑。

[0409] 在至少一个实施例中，GPGPU核心2062包括能够对多组数据执行单个指令的SIMD逻辑。在一个实施例中，GPGPU核心2062可以物理地执行SIMD4、SIMD8和SIMD16指令，并且在

逻辑上执行SIMD1、SIMD2和SIMD32指令。在至少一个实施例中，用于GPGPU核心的SIMD指令可以在编译时由着色器编译器生成，或者在执行针对单程序多数据(SPMD)或SIMT架构编写和编译的程序时自动生成。在至少一个实施例中，可以通过单个SIMD指令来执行为SIMT执行模型配置的程序的多个线程。例如，在至少一个实施例中，可以通过单个SIMD8逻辑单元并行执行执行相同或相似操作的八个SIMT线程。

[0410] 在至少一个实施例中，存储器和高速缓存互连2068是将图形多处理器2034的每个功能单元连接到寄存器文件2058和共享存储器2070的互连网络。在至少一个实施例中，存储器和高速缓存互连2068是交叉开关互连，其允许加载/存储单元2066在共享存储器2070和寄存器文件2058之间实现加载和存储操作。在至少一个实施例中，寄存器文件2058可以以与GPGPU核心2062相同的频率操作，从而在GPGPU核心2062和寄存器文件2058之间进行数据传输的延迟非常低。在至少一个实施例中，共享存储器2070可以用于启用在图形多处理器2034内的功能单元上执行的线程之间的通信。在至少一个实施例中，高速缓存存储器2072可以用作例如数据高速缓存，以高速缓存在功能单元和纹理单元2036之间通信的纹理数据。在至少一个实施例中，共享存储器2070也可以用作程序管理的高速缓存。在至少一个实施例中，除了存储在高速缓存存储器2072中的自动高速缓存的数据之外，在GPGPU核心2062上执行的线程还可以以编程方式将数据存储在共享存储器中。

[0411] 在至少一个实施例中，如本文所述的并行处理器或GPGPU通信地耦合到主机/处理器核心，以加速图形操作、机器学习操作、图案分析操作以及各种通用GPU(GPGPU)功能。在至少一个实施例中，GPU可以通过总线或其他互连(例如，诸如PCIe或NVLink的高速互连)通信地耦合到主机处理器/核心。在至少一个实施例中，GPU可以与核心集成在封装或芯片上，并通过内部处理器总线/互连(即，封装或芯片的内部)通信地耦合到核心。在至少一个实施例中，不管GPU连接的方式如何，处理器核心可以以工作描述符中包含的命令/指令序列的形式向该GPU分配工作。在至少一个实施例中，该GPU然后使用专用电路/逻辑来有效地处理这些命令/指令。

[0412] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。下面结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在图形多处理器2034中用于至少部分地基于使用本文描述的神经网络训练操作、神经网络功能和/或架构或神经网络用例计算的权重参数来进行推理或预测操作。

[0413] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0414] 图21示出了根据至少一个实施例的多GPU计算系统2100。在至少一个实施例中，多GPU计算系统2100可以包括经由主机接口交换机2104耦合到多个通用图形处理单元(GPGPU)2106A-D的处理器2102。在至少一个实施例中，主机接口交换机2104是将处理器2102耦合到PCI Express总线的PCI Express交换机设备，处理器2102可以通过PCI

Express总线与GPGPU 2106A-D通信。在至少一个实施例中,GPGPU 2106A-D可以经由一组高速P2P GPU到GPU链路2116互连。在至少一个实施例中,GPU到GPU链路2116经由专用GPU链路连接到GPGPU 2106A-D中的每一个。在至少一个实施例中,P2P GPU链路2116使得能够在每个GPGPU 2106A-D之间进行直接通信,而无需通过处理器2102所连接的主机接口总线2104进行通信。在至少一个实施例中,在GPU到GPU业务定向到P2P GPU链路2116的情况下,主机接口总线2104保持可用于系统存储器访问或例如经由一个或更多个网络设备与多GPU计算系统2100的其他实例进行通信。虽然在至少一个实施例中,GPGPU 2106A-D经由主机接口交换机2104连接到处理器2102,但是在至少一个实施例中,处理器2102包括对P2P GPU链路2116的直接支持,并且可以直接连接到GPGPU 2106A-D。

[0415] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中,推理和/或训练逻辑715可以在多GPU计算系统2100中使用,用于至少部分地基于使用本文描述的神经网络训练操作、神经网络函数和/或架构或神经网络用例计算的权重参数来进行推理或预测操作。

[0416] 在至少一个实施例中,参照所述图,一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0417] 图22是根据至少一个实施例的图形处理器2200的框图。在至少一个实施例中,图形处理器2200包括环形互连2202、管线前端2204、媒体引擎2237和图形核心2280A-2280N。在至少一个实施例中,环形互连2202将图形处理器2200耦合到其他处理单元,所述处理单元包括其他图形处理器或一个或更多个通用处理器核心。在至少一个实施例中,图形处理器2200是集成在多核心处理系统内的许多处理器之一。

[0418] 在至少一个实施例中,图形处理器2200经由环形互连2202接收多批命令。在至少一个实施例中,输入的命令由管线前端2204中的命令流转化器(streamer)2203解释。在至少一个实施例中,图形处理器2200包括可扩展执行逻辑,用于经由图形核心2280A-2280N执行3D几何处理和媒体处理。在至少一个实施例中,对于3D几何处理命令,命令流转化器2203将命令提供给几何管线2236。在至少一个实施例中,对于至少一些媒体处理命令,命令流转化器2203将命令提供给视频前端2234,该视频前端与媒体引擎2237耦合。在至少一个实施例中,媒体引擎2237包括用于视频和图像后处理的视频质量引擎(VQE)2230,以及用于提供硬件加速的媒体数据编码和解码的多格式编码/解码(MFX)2233引擎。在至少一个实施例中,几何管线2236和媒体引擎2237各自生成用于由至少一个图形核心2280提供的线程执行资源的执行线程。

[0419] 在至少一个实施例中,图形处理器2200包括具有(featuring)图形核心2280A-2280N(其可以是模块化的并且有时被称为核心切片)的可扩展线程执行资源,每个图形核心具有多个子核心2250A-2250N,2260A-2260N(有时称为核心子切片)。在至少一个实施例中,图形处理器2200可以具有任意数量的图形核心2280A。在至少一个实施例中,图形处理

器2200包括具有至少第一子核心2250A和第二子核心2260A的图形核心2280A。在至少一个实施例中，图形处理器2200是具有单个子核心(例如2250A)的低功率处理器。在至少一个实施例中，图形处理器2200包括多个图形核心2280A-2280N，每个图形核心包括一组第一子核心2250A-2250N和一组第二子核心2260A-2260N。在至少一个实施例中，第一子核心2250A-2250N中的每个子核心至少包括第一组执行单元2252A-2252N和媒体/纹理采样器2254A-2254N。在至少一个实施例中，第二子核心2260A-2260N中的每个子核心至少包括第二组执行单元2262A-2262N和采样器2264A-2264N。在至少一个实施例中，每个子核心2250A-2250N，2260A-2260N共享一组共享资源2270A-2270N。在至少一个实施例中，共享资源包括共享高速缓存存储器和像素操作逻辑。

[0420] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715可以在图形处理器2200中用于至少部分地基于使用本文描述的神经网络训练操作、神经网络功能和/或架构或神经网络用例计算的权重参数来进行推理或预测操作。

[0421] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0422] 图23是根据至少一个实施例的说明用于处理器2300的微架构的框图，该处理器2300可以包括用于执行指令的逻辑电路。在至少一个实施例中，处理器2300可以执行指令，包括x86指令、ARM指令、用于专用集成电路(ASIC)的专用指令等。在至少一个实施例中，处理器2300可以包括用于存储封装数据的寄存器，例如作为加利福尼亚州圣克拉拉市英特尔公司采用MMX技术启用的微处理器中的64位宽MMX™寄存器。在至少一个实施例中，整数和浮点数形式可用的MMX寄存器可以与封装的数据元素一起运行，所述封装的数据元素伴随单指令多数据(“SIMD”)和流式SIMD扩展(“SSE”)指令。在至少一个实施例中，与SSE2、SSE3、SSE4、AVX或更高版本(一般称为“SSEx”)技术有关的128位宽XMM寄存器可以保存此类封装数据操作数。在至少一个实施例中，处理器2300可以执行指令以加速机器学习或深度学习算法、训练或推理。

[0423] 在至少一个实施例中，处理器2300包括有序前端(“前端”)2301，以提取要执行的指令并准备稍后在处理器管线中使用的指令。在至少一个实施例中，前端2301可以包括几个单元。在至少一个实施例中，指令预取器2326从存储器中获取指令并将指令提供给指令解码器2328，指令解码器2328又对指令进行解码或解释。例如，在至少一个实施例中，指令解码器2328将接收到的指令解码为机器可执行的所谓的“微指令”或“微操作”(也称为“微操作”或“微指令”)的一个或更多个操作。在至少一个实施例中，指令解码器2328将指令解析为操作码以及相应的数据和控制字段，其可以由微架构用来使用以根据至少一个实施例来执行操作。在至少一个实施例中，跟踪高速缓存2330可以将解码的微指令组装成微指令队列2334中的程序排序的序列或追踪以供执行。在至少一个实施例中，当追踪高速缓存

2330遇到复杂指令时，微码ROM 2332提供完成操作所需的微指令。

[0424] 在至少一个实施例中，可以将一些指令转换成单个微操作，而另一些指令则需要几个微操作来完成全部操作。在至少一个实施例中，如果需要多于四个的微指令来完成一条指令，则指令解码器2328可以访问微码ROM 2332以执行该指令。在至少一个实施例中，可以将指令解码为少量的微指令以在指令解码器2328处进行处理。在至少一个实施例中，如果需要多个微指令完成该操作，则可以将指令存储在微码ROM 2332中。在至少一个实施例中，追踪高速缓存器2330参考入口点可编程逻辑阵列(“PLA”)以确定正确的微指令指针，用于根据至少一个实施例从微码ROM 2332读取微码序列以完成一个或更多个指令。在至少一个实施例中，在微码ROM 2332完成对指令的微操作排序之后，机器的前端2301可以恢复从追踪高速缓存2330获取微操作。

[0425] 在至少一个实施例中，乱序执行引擎(“乱序引擎”)2303可以准备用于执行的指令。在至少一个实施例中，乱序执行逻辑具有多个缓冲区，以使指令流平滑并重新排序，以在指令沿管线下降并被调度执行时优化性能。在至少一个实施例中，乱序执行引擎2303包括但不限于分配器/寄存器重命名器2340、存储器微指令队列2342、整数/浮点微指令队列2344、存储器调度器2346、快速调度器2302、慢速/通用浮点调度器(“慢速/通用FP调度器”)2304和简单浮点调度器(“简单FP调度器”)2306。在至少一个实施例中，快速调度器2302、慢速/通用浮点调度器2304和简单浮点调度器2306也统称为“微指令调度器2302、2304、2306”。在至少一个实施例中，分配器/寄存器重命名器2340分配每个微指令按序列执行所需要的机器缓冲区和资源。在至少一个实施例中，分配器/寄存器重命名器2340将逻辑寄存器重命名为寄存器文件中的条目。在至少一个实施例中，分配器/寄存器重命名器2340还为两个微指令队列之一中的每个微指令分配条目，存储器微指令队列2342用于存储器操作和整数/浮点微指令队列2344用于非存储器操作，在存储器调度器2346和微指令调度器2302、2304、2306的前面。在至少一个实施例中，微指令调度器2302、2304、2306基于它们的从属输入寄存器操作数源的就绪性和需要完成的执行资源微指令的可用性来确定何时准备好执行微指令。至少一个实施例的快速调度器2302可以在主时钟周期的每个一半上调度，而慢速/通用浮点调度器2304和简单浮点调度器2306可以在每个主处理器时钟周期调度一次。在至少一个实施例中，微指令调度器2302、2304、2306对调度端口进行仲裁，以调度用于执行的微指令。

[0426] 在至少一个实施例中，执行块2311包括但不限于整数寄存器文件/支路网络2308、浮点寄存器文件/支路网络(“FP寄存器文件/支路网络”)2310、地址生成单元(“AGU”)2312和2314、快速算术逻辑单元(“快速ALU”)2316和2318、慢速算术逻辑单元(“慢速ALU”)2320、浮点ALU(“FP”)2322和浮点移动单元(“FP移动”)2324。在至少一个实施例中，整数寄存器文件/支路网络2308和浮点寄存器文件/旁路网络2310在本文中也称为“寄存器文件2308、2310”。在至少一个实施例中，AGU 2312和2314、快速ALU 2316和2318、慢速ALU 2320、浮点ALU 2322和浮点移动单元2324在本文中也称为“执行单元2312、2314、2316、2318、2320、2322和2324”。在至少一个实施例中，执行块2311可以包括但不限于任意数量(包括零)和类型的寄存器文件、支路网络、地址生成单元和执行单元(以任何组合)。

[0427] 在至少一个实施例中，寄存器网络2308、2310可以布置在微指令调度器2302、2304、2306与执行单元2312、2314、2316、2318、2320、2322和2324之间。在至少一个实施例

中，整数寄存器文件/支路网络2308执行整数运算。在至少一个实施例中，浮点寄存器文件/支路网络2310执行浮点操作。在至少一个实施例中，寄存器网络2308、2310中的每一个可以包括但不限于支路网络，该支路网络可以绕过或转发尚未写入寄存器文件中的刚刚完成的结果到新的从属对象。在至少一个实施例中，寄存器网络2308、2310可以彼此通信数据。在至少一个实施例中，整数寄存器文件/支路网络2308可以包括但不限于两个单独的寄存器文件、一个寄存器文件用于低阶32位数据，第二寄存器文件用于高阶32位数据。在至少一个实施例中，浮点寄存器文件/支路网络2310可以包括但不限于128位宽的条目，因为浮点指令通常具有宽度为64至128位的操作数。

[0428] 在至少一个实施例中，执行单元2312、2314、2316、2318、2320、2322、2324可以执行指令。在至少一个实施例中，寄存器网络2308、2310存储微指令需要执行的整数和浮点数据操作数值。在至少一个实施例中，处理器2300可以包括但不限于任何数量的执行单元2312、2314、2316、2318、2320、2322、2324及其组合。在至少一个实施例中，浮点ALU 2322和浮点移动单元2324，可以执行浮点、MMX、SIMD、AVX和SSE或其他操作，包括专门的机器学习指令。在至少一个实施例中，浮点ALU 2322可以包括但不限于64位乘64位浮点除法器，以执行除法、平方根和余数微操作。在至少一个实施例中，可以用浮点硬件来处理涉及浮点值的指令。在至少一个实施例中，可以将ALU操作传递给快速ALU 2316、2318。在至少一个实施例中，快速ALU 2316、2318可以以半个时钟周期的有效延迟执行快速操作。在至少一个实施例中，大多数复杂的整数运算进入慢速ALU 2320，因为慢速ALU 2320可以包括但不限于用于长延迟类型操作的整数执行硬件，例如乘法器、移位、标志逻辑和分支处理。在至少一个实施例中，存储器加载/存储操作可以由AGU 2312、2314执行。在至少一个实施例中，快速ALU 2316、快速ALU 2318和慢速ALU 2320可以对64位数据操作数执行整数运算。在至少一个实施例中，可以实现快速ALU 2316、快速ALU 2318和慢速ALU 2320以支持包括十六、三十二、128、256等的各种数据位大小。在至少一个实施例中，浮点ALU 2322和浮点移动单元2324可以实现为支持具有各种宽度的位的一定范围的操作数，例如可以结合SIMD和多媒体指令对128位宽封装数据操作数进行操作。

[0429] 在至少一个实施例中，微指令调度器2302、2304、2306在父加载完成执行之前调度从属操作。在至少一个实施例中，由于可以在处理器2300中推测性地调度和执行微指令，处理器2300还可以包括用于处理存储器未命中的逻辑。在至少一个实施例中，如果数据高速缓存中的数据加载未命中，则可能存在在管线中正在运行的从属操作，其使调度器暂时没有正确的数据。在至少一个实施例中，一种重放机制追踪踪并重新执行使用不正确数据的指令。在至少一个实施例中，可能需要重放从属操作并且可以允许完成独立操作。在至少一个实施例中，处理器的至少一个实施例的调度器和重放机制也可以设计为捕获用于文本串比较操作的指令序列。

[0430] 在至少一个实施例中，“寄存器”可以指代可以用作识别操作数的指令的一部分的机载处理器存储位置。在至少一个实施例中，寄存器可以是那些可以从处理器外部使用的寄存器(从程序员的角度来看)。在至少一个实施例中，寄存器可能不限于特定类型的电路。相反，在至少一个实施例中，寄存器可以存储数据、提供数据并执行本文描述的功能。在至少一个实施例中，本文描述的寄存器可以通过处理器内的电路使用多种不同技术来实现，例如专用物理寄存器、使用寄存器重命名动态分配的物理寄存器、专用和动态分配的物理

寄存器的组合等。在至少一个实施例中，整数寄存器存储32位整数数据。至少一个实施例的寄存器文件还包含八个用于封装数据的多媒体SIMD寄存器。

[0431] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，可以将推理和/或训练逻辑715的部分或全部并入执行块2311以及示出或未示出的其他存储器或寄存器。例如，在至少一个实施例中，本文描述的训练和/或推理技术可以使用执行块2311中示出的一个或更多个ALU。此外，权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中，该寄存器和/或寄存器配置执行块2311的ALU以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0432] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0433] 图24示出了根据至少一个实施例的深度学习应用程序处理器2400。在至少一个实施例中，深度学习应用程序处理器2400使用指令，如果由深度学习应用程序处理器2400执行，则指令使深度学习应用程序处理器2400执行贯穿本公开描述的一些或全部过程和技术。在至少一个实施例中，深度学习应用程序处理器2400是专用集成电路(ASIC)。在至少一个实施例中，应用程序处理器2400执行矩阵乘法运算或者“硬连线”到硬件中，作为执行一个或更多个指令或两者的结果。在至少一个实施例中，深度学习应用程序处理器2400包括但不限于处理集群2410(1)-2410(12)、芯片间链路(“ICL”)2420(1)-2420(12)、芯片间控制器(“ICC”)2430(1)-2430(2)、第二代高带宽存储器(“HBM2”)2440(1)-2440(4)、存储器控制器(“Mem Ctrlr”)2442(1)-2442(4)、高带宽存储器物理层(“HBM PHY”)2444(1)-2444(4)、管理控制器中央处理单元(“管理控制器CPU”)2450、串行外围设备接口、内部集成电路和通用输入/输出块(“SPI、I2C、GPIO”)2460，外围组件互连快速控制器和直接存储器访问块(“PCIe控制器和DMA”)2470、以及十六通道外围组件互连快速端口(“PCI Express x 16”)2480。

[0434] 在至少一个实施例中，处理集群2410可以执行深度学习操作，包括基于一种或更多种训练技术计算的权重参数的推理或预测操作，包括本文所述的那些技术。在至少一个实施例中，每个处理集群2410可以包括但不限于任何数量和类型的处理器。在至少一个实施例中，深度学习应用程序处理器2400可以包括任何数量和类型的处理集群2400。在至少一个实施例中，芯片间链路2420是双向的。在至少一个实施例中，芯片间链路2420和芯片间控制器2430使多个深度学习应用程序处理器2400能够交换信息，包括从执行一个或更多个神经网络中体现的一种或更多种机器学习算法而产生的激活信息。在至少一个实施例中，深度学习应用程序处理器2400可以包括任意数量(包括零)和类型的ICL 2420和ICC 2430。

[0435] 在至少一个实施例中，HBM2 2440提供总共32GB的存储器。在至少一个实施例中，HBM2 2440(i)与存储器控制器2442(i)和HBM PHY 2444(i)都相关联，其中“i”是任意整数。在至少一个实施例中，任何数量的HBM2 2440可以提供任何类型和总量的高带宽存储器，并

且可以与任何数量(包括零)和类型的存储器控制器2442和HBM PHY 2444相关联。在至少一个实施例中,可以用任何数量和类型的块替换SPI、I2C、GPIO 2460、PCIe控制器和DMA 2470和/或PCIe2480,以任何技术上可行的方式实现任何数量和类型的通信标准。

[0436] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中,深度学习应用程序处理器用于训练机器学习模型(例如神经网络),以预测或推理提供给深度学习应用程序处理器2400的信息。在至少一个实施例中,深度学习应用程序处理器2400用于基于已经由另一处理器或系统或由深度学习应用程序处理器2400训练的经训练的机器学习模型(例如,神经网络)来推理或预测信息。在至少一个实施例中,处理器2400可以用于执行本文所述的一个或更多个神经网络用例。

[0437] 在至少一个实施例中,一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0438] 图25是根据至少一个实施例的神经形态处理器2500的框图。在至少一个实施例中,神经形态处理器2500可以从神经形态处理器2500外部的源接收一个或更多个输入。在至少一个实施例中,这些输入可以被传输到神经形态处理器2500内的一个或更多个神经元2502。在至少一个实施例中,可以使用包括一个或更多个算术逻辑单元(ALU)的电路或逻辑来实现神经元2502及其组件。在至少一个实施例中,神经形态处理器2500可以包括但不限于成千上万个神经元2502的实例,但是可以使用任何合适数量的神经元2502。在至少一个实施例中,神经元2502的每个实例可以包括神经元输入2504和神经元输出2506。在至少一个实施例中,神经元2502可以生成可以传输到神经元2502的其他实例的输入的输出。在至少一个实施例中,神经元输入2504和神经元输出2506可以经由突触2508互连。

[0439] 在至少一个实施例中,神经元2502和突触2508可以互连,使得神经形态处理器2500操作以处理或分析由神经形态处理器2500接收的信息。在至少一个实施例中,当通过神经元输入2504接收到的输入超过阈值时,神经元2502可以发送输出脉冲(或“触发”或“峰值”)。在至少一个实施例中,神经元2502可以对在神经元输入2504处接收到的信号进行求和或积分。例如,在至少一个实施例中,神经元2502可以实现为有泄漏的积分-触发神经元,其中如果求和(称为“膜电位”)超过阈值,则神经元2502可以使用诸如sigmoid或阈值函数的传递函数来产生输出(或“触发”)。在至少一个实施例中,泄漏的积分-触发神经元可以将在神经元输入2504处接收到的信号求和成膜电位,并且可以应用程序衰减因子(或泄漏)以减小膜电位。在至少一个实施例中,如果在神经元输入2504处接收到足够快以超过阈值的多个输入信号(即,在膜电势衰减得太低而不能触发之前),则泄漏的积分-触发神经元可能会触发。在至少一个实施例中,神经元2502可以使用接收输入、将输入积分到膜电位、并衰减膜电位的电路或逻辑来实现。在至少一个实施例中,可以对输入求平均,或者可以使用任何其他合适的传递函数。此外,在至少一个实施例中,神经元2502可以包括但不限于当将传递函数应用程序于神经元输入2504的结果超过阈值时在神经元输出2506处产生输出尖峰

的比较器电路或逻辑。在至少一个实施例中，一旦神经元2502触发，它可以通过例如将膜电位复位为0或另一合适的默认值来忽略先前接收的输入信息。在至少一个实施例中，一旦膜电位被重置为0，则神经元2502可以在合适的时间段(或修复期)之后恢复正常操作。

[0440] 在至少一个实施例中，神经元2502可以通过突触2508互连。在至少一个实施例中，突触2508可以操作以将从第一神经元2502的输出的信号传输到第二神经元2502的输入。在至少一个实施例中，神经元2502可以在一个以上的突触2508实例上传输信息。在至少一个实施例中，神经元输出2506的一个或更多个实例可以通过突触2508的实例连接到同一神经元2502中神经元输入2504的实例。在至少一个实施例中，相对于突触2508的那个实例，神经元2502的实例产生要在突触2508的实例上传输的输出可以被称为“突触前神经元”。在至少一个实施例中，相对于突触2508的实例，神经元2502的实例接收通过突触2508的实例传输的输入可以被称为“突触后神经元”。在至少一个实施例中，关于突触2508的各种实例，因为神经元2502的实例可以接收来自一个或更多个突触2508实例的输入，并且还可以通过一个或更多个突触2508实例传输输出，因此神经元2502的单个实例可以既是“突触前神经元”又是“突触后神经元”。

[0441] 在至少一个实施例中，神经元2502可以被组织成一层或更多层。在至少一个实施例中，神经元2502的每个实例可以具有一个神经元输出2506，该神经元输出2506可以通过一个或更多个突触2508扇出到一个或更多个神经元输入2504。在至少一个实施例中，第一层2510中的神经元2502的神经元输出2506可以连接到第二层2512中的神经元2502的神经元输入2504。在至少一个实施例中，层2510可以被称为“前馈层”。在至少一个实施例中，在第一层2510的实例中神经元2502的每个实例可以扇出到第二层2512中的神经元2502的每个实例。在至少一个实施例中，第一层2510可以被称为“完全连接的前馈层”。在至少一个实施例中，在第二层2512的每个实例中的神经元2502的每个实例扇出到少于在第三层2514中的神经元2502的所有实例。在至少一个实施例中，第二层2512可以被称为“稀疏连接的前馈层”。在至少一个实施例中，第二层2512中的神经元2502可以扇出到多个其他层中的神经元2502，也包括扇出到第二层2512中的神经元2502。在至少一个实施例中，第二层2512可以被称为“循环层”。在至少一个实施例中，神经形态处理器2500可以包括但不限于循环层和前馈层的任何合适的组合，包括但不限于稀疏连接的前馈层和完全连接的前馈层。

[0442] 在至少一个实施例中，神经形态处理器2500可以包括但不限于可重新配置的互连架构或专用硬连线互连，以将突触2508连接到神经元2502。在至少一个实施例中，神经形态处理器2500可以包括但不限于电路或逻辑，其根据神经网络拓扑结构和神经元扇入/扇出，允许根据需要将突触分配给不同神经元2502。例如，在至少一个实施例中，可以使用互连结构(诸如片上网络)或通过专用连接将突触2508连接到神经元2502。在至少一个实施例中，可以使用电路或逻辑来实现突触互连及其组件。

[0443] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他装置或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0444] 图26示出了根据至少一个实施例的处理系统。在至少一个实施例中，系统2600包括一个或更多个处理器2602和一个或更多个图形处理器2608，并且可以是单处理器台式机系统、多处理器工作站系统或具有大量处理器2602或处理器核心2607的服务器系统。在至少一个实施例中，系统2600是结合在片上系统(SoC)集成电路内的处理平台，以在移动、手持或嵌入式设备使用。

[0445] 在至少一个实施例中，系统2600可以包括或结合在基于服务器的游戏平台中，包括游戏和媒体控制台的游戏控制台、移动游戏控制台、手持游戏控制台或在线游戏控制台。在至少一个实施例中，系统2600是移动电话、智能电话、平板计算设备或移动互联网设备。在至少一个实施例中，处理系统2600还可包括与可穿戴设备耦合或集成在可穿戴设备中，例如智能手表可穿戴设备、智能眼镜设备、增强现实设备或虚拟现实设备。在至少一个实施例中，处理系统2600是电视或机顶盒设备，其具有一个或更多个处理器2602以及由一个或更多个图形处理器2608生成的图形界面。

[0446] 在至少一个实施例中，一个或更多个处理器2602每个包括一个或更多个处理器核心2607，以处理指令，该指令在被执行时执行针对系统和用户软件的操作。在至少一个实施例中，一个或更多个处理器核心2607中的每一个被配置为处理特定指令序列2609。在至少一个实施例中，指令序列2609可以促进复杂指令集计算(CISC)、精简指令集计算(RISC)，或通过超长指令字(VLIW)进行计算。在至少一个实施例中，处理器核心2607可以各自处理不同的指令序列2609，该指令序列可以包括有助于仿真其他指令序列的指令。在至少一个实施例中，处理器核心2607还可以包括其他处理设备，例如数字信号处理器(DSP)。

[0447] 在至少一个实施例中，处理器2602包括高速缓存存储器2604。在至少一个实施例中，处理器2602可以具有单个内部高速缓存或多个级别的内部高速缓存。在至少一个实施例中，高速缓存存储器在处理器2602的各个组件之间共享。在至少一个实施例中，处理器2602还使用外部高速缓存(例如，三级(L3)高速缓存或最后一级高速缓存(LLC))(未示出)，可以使用已知的高速缓存一致性技术在处理器核心2607之间共享该外部高速缓存。在至少一个实施例中，处理器2602中另外包括寄存器文件2606，处理器可以包括用于存储不同类型的数据的不同类型的寄存器(例如，整数寄存器、浮点寄存器、状态寄存器和指令指针寄存器)。在至少一个实施例中，寄存器文件2606可以包括通用寄存器或其他寄存器。

[0448] 在至少一个实施例中，一个或更多个处理器2602与一个或更多个接口总线2610耦合，以在处理器2602与系统2600中的其他组件之间传输通信信号，例如地址、数据或控制信号。在至少一个实施例中，接口总线2610在一个实施例中可以是处理器总线，例如直接媒体接口(DMI)总线的版本。在至少一个实施例中，接口总线2610不限于DMI总线，并且可以包括一个或更多个外围组件互连总线(例如，PCI，PCI Express)、存储器总线或其他类型的接口总线。在至少一个实施例中，处理器2602包括集成存储器控制器2616和平台控制器集线器2630。在至少一个实施例中，存储器控制器2616促进存储器设备与处理系统2600的其他组件之间的通信，而平台控制器集线器(PCH)2630通过本地I/O总线提供到输入/输出(I/O)设备的连接。

[0449] 在至少一个实施例中，存储器设备2620可以是动态随机存取存储器(DRAM)设备、静态随机存取存储器(SRAM)设备、闪存设备、相变存储设备或具有适当的性能以用作处理器存储器。在至少一个实施例中，存储设备2620可以用作处理系统2600的系统存储器，以存

储数据2622和指令2621，以在一个或更多个处理器2602执行应用程序或过程时使用。在至少一个实施例中，存储器控制器2616还与可选的外部图形处理器2612耦合，其可以与处理器2602中的一个或更多个图形处理器2608通信以执行图形和媒体操作。在至少一个实施例中，显示设备2611可以连接至处理器2602。在至少一个实施例中，显示设备2611可以包括内部显示设备中的一个或更多个，例如在移动电子设备或膝上型设备或通过显示器接口(例如显示端口(DisplayPort)等)连接的外部显示设备中。在至少一个实施例中，显示设备2611可以包括头戴式显示器(HMD)，诸如用于虚拟现实(VR)应用或增强现实(AR)应用中的立体显示设备。

[0450] 在至少一个实施例中，平台控制器集线器2630使外围设备能够通过高速I/O总线连接到存储设备2620和处理器2602。在至少一个实施例中，I/O外围设备包括但不限于音频控制器2646、网络控制器2634、固件接口2628、无线收发器2626、触摸传感器2625、数据存储设备2624(例如，硬盘驱动器、闪存等)。在至少一个实施例中，数据存储设备2624可以经由存储接口(例如，SATA)或经由外围总线来连接，诸如外围组件互连总线(例如，PCI、PCIe)。在至少一个实施例中，触摸传感器2625可以包括触摸屏传感器、压力传感器或指纹传感器。在至少一个实施例中，无线收发器2626可以是Wi-Fi收发器、蓝牙收发器或移动网络收发器，诸如3G、4G或长期演进(LTE)收发器。在至少一个实施例中，固件接口2628使能与系统固件的通信，并且可以是例如统一可扩展固件接口(UEFI)。在至少一个实施例中，网络控制器2634可以启用到有线网络的网络连接。在至少一个实施例中，高性能网络控制器(未示出)与接口总线2610耦合。在至少一个实施例中，音频控制器2646是多通道高清晰度音频控制器。在至少一个实施例中，处理系统2600包括可选的传统(legacy)I/O控制器2640，用于将传统(例如，个人系统2(PS/2))设备耦合到系统2600。在至少一个实施例中，平台控制器集线器2630还可以连接到一个或更多个通用串行总线(USB)控制器2642，该控制器连接输入设备，诸如键盘和鼠标2643组合、相机2644或其他USB输入设备。

[0451] 在至少一个实施例中，存储器控制器2616和平台控制器集线器2630的实例可以集成到离散的外部图形处理器中，例如外部图形处理器2612。在至少一个实施例中，平台控制器集线器2630和/或存储器控制器2616可以在一个或更多个处理器2602的外部。例如，在至少一个实施例中，系统2600可以包括外部存储器控制器2616和平台控制器集线器2630，其可以配置成在与处理器2602通信的系统芯片组中的存储器控制器集线器和外围控制器集线器。

[0452] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，部分或全部推理和/或训练逻辑715可以结合到图形处理器2608中。例如，在至少一个实施例中，本文描述的训练和/或推理技术可以使用一个或更多个ALU，所述ALU体现在3D管线中。此外，在至少一个实施例中，本文描述的推理和/或训练操作可以使用除图7A或图7B所示的逻辑之外的逻辑来完成。在至少一个实施例中，权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中，其配置图形处理器2608的ALU，以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0453] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更

多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0454] 图27是根据至少一个实施例的具有一个或更多个处理器核心2702A-2702N、集成存储器控制器2714和集成图形处理器2708的处理器2700的框图。在至少一个实施例中，处理器2700可以包含附加核心，多达并包括以虚线框表示的附加核心2702N。在至少一个实施例中，每个处理器核心2702A-2702N包括一个或更多个内部高速缓存单元2704A-2704N。在至少一个实施例中，每个处理器核心还可以访问一个或更多个共享高速缓存单元2706。

[0455] 在至少一个实施例中，内部高速缓存单元2704A-2704N和共享高速缓存单元2706表示处理器2700内的高速缓存存储器层次结构。在至少一个实施例中，高速缓存存储器单元2704A-2704N可以包括每个处理器核心内的至少一级指令和数据高速缓存以及共享中级高速缓存中的一级或更多级缓存，例如2级(L2)、3级(L3)、4级(L4)或其他级别的高速缓存，其中将外部存储器之前的最高级别的高速缓存归类为LLC。在至少一个实施例中，高速缓存一致性逻辑维持各种高速缓存单元2706和2704A-2704N之间的一致性。

[0456] 在至少一个实施例中，处理器2700还可包括一组一个或更多个总线控制器单元2716和系统代理核心2710。在至少一个实施例中，一个或更多个总线控制器单元2716管理一组外围总线，例如一个或更多个PCI或PCIe总线。在至少一个实施例中，系统代理核心2710为各种处理器组件提供管理功能。在至少一个实施例中，系统代理核心2710包括一个或更多个集成存储器控制器2714，以管理对各种外部存储器设备(未示出)的访问。

[0457] 在至少一个实施例中，一个或更多个处理器核心2702A-2702N包括对多线程同时进行的支持。在至少一个实施例中，系统代理核心2710包括用于在多线程处理期间协调和操作核心2702A-2702N的组件。在至少一个实施例中，系统代理核心2710可以另外包括电源控制单元(PCU)，该电源控制单元包括用于调节处理器核心2702A-2702N和图形处理器2708的一个或更多个电源状态的逻辑和组件。

[0458] 在至少一个实施例中，处理器2700还包括用于执行图处理操作的图形处理器2708。在至少一个实施例中，图形处理器2708与共享高速缓存单元2706和包括一个或更多个集成存储器控制器2714的系统代理核心2710耦合。在至少一个实施例中，系统代理核心2710还包括用于驱动图形处理器输出到一个或更多个耦合的显示器的显示器控制器2711。在至少一个实施例中，显示器控制器2711也可以是经由至少一个互连与图形处理器2708耦合的独立模块，或者可以集成在图形处理器2708内。

[0459] 在至少一个实施例中，基于环的互连单元2712用于耦合处理器2700的内部组件。在至少一个实施例中，可以使用替代性互连单元，例如点对点互连、交换互连或其他技术。在至少一个实施例中，图形处理器2708经由I/O链路2713与环形互连2712耦合。

[0460] 在至少一个实施例中，I/O链路2713代表多种I/O互连中的至少一种，包括促进各种处理器组件与高性能嵌入式存储器模块2718(例如eDRAM模块)之间的通信的封装I/O互连。在至少一个实施例中，处理器核心2702A-2702N和图形处理器2708中的每一个使用嵌入式存储器模块2718作为共享的最后一级高速缓存。

[0461] 在至少一个实施例中，处理器核心2702A-2702N是执行公共指令集架构的同质核

心。在至少一个实施例中，处理器核心2702A-2702N在指令集架构(ISA)方面是异构的，其中一个或更多个处理器核心2702A-2702N执行公共指令集，而一个或更多个其他处理器核心2702A-2702N执行公共指令集的子集或不同指令集。在至少一个实施例中，就微架构而言，处理器核心2702A-2702N是异构的，其中具有相对较高功耗的一个或更多个核心与具有较低功耗的一个或更多个功率核心耦合。在至少一个实施例中，处理器2700可以在一个或更多个芯片上实现或被实现为SoC集成电路。

[0462] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，部分或全部推理和/或训练逻辑715可以结合到图形处理器2708中。例如，在至少一个实施例中，本文描述的训练和/或推理技术可以使用一个或更多个ALU，所述ALU体现在图27中的3D管线、图形核心2702、共享功能逻辑，或其他逻辑中。此外，在至少一个实施例中，本文描述的推理和/或训练操作可以使用除图7A或图7B所示的逻辑之外的逻辑来完成。在至少一个实施例中，权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中，其配置处理器2700的ALU以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0463] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0464] 图28是图形处理器2800的框图，该图形处理器可以是分立的图形处理单元，或者可以是与多个处理核心集成的图形处理器。在至少一个实施例中，图形处理器2800经由存储器映射的I/O接口与图形处理器2800上的寄存器以及放置在存储器中的命令进行通信。在至少一个实施例中，图形处理器2800包括用于访问存储器的存储器接口2814。在至少一个实施例中，存储器接口2814是到本地存储器、一个或更多个内部高速缓存、一个或更多个共享的外部高速缓存和/或到系统存储器的接口。

[0465] 在至少一个实施例中，图形处理器2800还包括用于将显示输出数据驱动到显示设备2820的显示控制器2802。在至少一个实施例中，显示控制器2802包括用于显示设备2820的一个或更多个覆盖平面的硬件以及多层视频或用户接口元素的组合。在至少一个实施例中，显示设备2820可以是内部或外部显示设备。在至少一个实施例中，显示设备2820是头戴式显示设备，例如虚拟现实(VR)显示设备或增强现实(AR)显示设备。在至少一个实施例中，图形处理器2800包括视频编解码器引擎2806，以将媒体编码、解码或转码为一种或更多种媒体编码格式，从一种或更多种媒体编码格式编码、解码或转码，或在一种或更多种媒体编码格式之间进行编码、解码或转码，所述媒体编码格式包括但不限于运动图像专家组(MPEG)格式(例如MPEG-2)，高级视频编码(AVC)格式(例如H.264/MPEG-4AVC，以及美国电影电视工程师协会(SMPTE)421M/VC-1)和联合图像专家组(JPEG)格式(例如JPEG)和Motion JPEG(MJPEG)格式。

[0466] 在至少一个实施例中，图形处理器2800包括块图像传送(BLIT)引擎2804，以执行

二维(2D)光栅化器操作,包括例如位边界块传送。但是,在至少一个实施例中,使用图形处理引擎(GPE)2810的一个或更多个组件来执行2D图形操作。在至少一个实施例中,GPE 2810是用于执行图形操作(包括三维(3D)图形操作和媒体操作)的计算引擎。

[0467] 在至少一个实施例中,GPE 2810包括用于执行3D操作的3D管线2812,例如使用对3D图元形状(例如,矩形、三角形等)进行操作的处理功能来渲染三维图像和场景。在至少一个实施例中,3D管线2812包括执行各种任务和/或产生到3D/媒体子系统2815的执行线程的可编程和固定功能元素。虽然3D管线2812可用于执行媒体操作,但是在至少一个实施例中,GPE 2810还包括媒体管线2816,其用于执行媒体操作,诸如视频后处理和图像增强。

[0468] 在至少一个实施例中,媒体管线2816包括固定功能或可编程逻辑单元,用于执行一种或更多种专门的媒体操作,例如视频解码加速,视频去隔行和视频编码加速,代替或代表视频编解码器引擎2806。在至少一个实施例中,媒体管线2816还包括线程产生单元,用于产生线程以在3D/媒体子系统2815上执行。在至少一个实施例中,产生的线程在3D/媒体子系统2815中包含的一个或更多个图形执行单元上执行媒体操作的计算。

[0469] 在至少一个实施例中,3D/媒体子系统2815包括用于执行3D管线2812和媒体管线2816产生的线程的逻辑。在至少一个实施例中,3D管线2812和媒体管线2816将线程执行请求发送到3D/媒体子系统2815,其包括用于仲裁各种请求并将其分派给可用线程执行资源的线程分派逻辑。在至少一个实施例中,执行资源包括用于处理3D和媒体线程的图形执行单元的阵列。在至少一个实施例中,3D/媒体子系统2815包括用于线程指令和数据的一个或更多个内部高速缓存。在至少一个实施例中,子系统2815还包括共享存储器,其包括寄存器和可寻址存储器,以在线程之间共享数据并存储输出数据。

[0470] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中,可以将推理和/或训练逻辑715的部分或全部合并到处理器2800中。例如,在至少一个实施例中,本文描述的训练和/或推理技术可以使用3D管线2812中包含的一个或更多个ALU。此外,在至少一个实施例中,本文描述的推理和/或训练操作可以使用除图7A或图7B所示的逻辑以外的逻辑来完成。在至少一个实施例中,权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中,其配置图形处理器2800的ALU以执行一种或更多种机器学习算法、神经网络架构、用例或本文介绍的训练技术。

[0471] 在至少一个实施例中,参照所述图,一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0472] 图29是根据至少一个实施例的图形处理器的图形处理引擎2910的框图。在至少一个实施例中,图形处理引擎(GPE)2910是图28中所示的GPE 2810的版本。在至少一个实施例中,媒体管线2916是可选的,并且可以不显式地包括在GPE 2910中。在至少一个实施例中,单独的媒体和/或图像处理器耦合到GPE 2910。

[0473] 在至少一个实施例中,GPE 2910耦合到或包括命令流转化器2903,其向3D管线

2912和/或媒体管线2916提供命令流。在至少一个实施例中，命令流转化器2903耦合到存储器，所述存储器可以是系统存储器，也可以是内部高速缓存存储器和共享高速缓存存储器中的一个或更多个。在至少一个实施例中，命令流转化器2903从存储器接收命令，并且将命令发送到3D管线2912和/或媒体管线2916。在至少一个实施例中，命令是从环形缓冲区中获取的指令、基元或微操作，该环形缓冲区存储用于3D管线2912和媒体管线2916的命令。在至少一个实施例中，环形缓冲区还可以包括存储各批多个命令的批命令缓冲区。在至少一个实施例中，用于3D管线2912的命令还可以包括对存储在存储器中的数据的引用，例如但不限于用于3D管线2912的顶点和几何数据和/或用于媒体管线2916的图像数据和存储器对象。在至少一个实施例中，3D管线2912和媒体管线2916通过执行操作或通过将一个或更多个执行线程分派到图形核心阵列2914，来处理命令和数据。在至少一个实施例中，图形核心阵列2914包括一个或更多个图形核心块(例如，一个或更多个图形核心2915A、一个或更多个图形核心2915B)，每个块包括一个或更多个图形核心。在至少一个实施例中，每个图形核心包括一组图形执行资源，所述图形执行资源包括通用和图形特定的执行逻辑，用于执行图形和计算操作，以及固定功能纹理处理和/或机器学习和人工智能加速逻辑，包括图7A和图7B中的推理和/或训练逻辑715。

[0474] 在至少一个实施例中，3D管线2912包括固定功能和可编程逻辑，用于通过处理指令并将执行线程分派到图形核心阵列2914，来处理一个或更多个着色器程序，例如顶点着色器、几何着色器、像素着色器、片段着色器、计算着色器或其他着色器程序。在至少一个实施例中，图形核心阵列2914提供统一的执行资源块，所述执行资源块用于处理着色器程序。在至少一个实施例中，在图形核心阵列2914的图形核心2915A-2915B内的多用途执行逻辑(例如，执行单元)包括对各种3D API着色器语言的支持，并且可以执行与多个着色器关联的多个同时执行线程。

[0475] 在至少一个实施例中，图形核心阵列2914还包括执行逻辑，用于执行媒体功能，诸如视频和/或图像处理。在至少一个实施例中，除了图形处理操作之外，执行单元还包括可编程以执行并行通用计算操作的通用逻辑。

[0476] 在至少一个实施例中，输出数据可以将数据输出到统一返回缓冲区(URB)2918中的存储器，所述输出数据由在图形核心阵列2914上执行的线程生成。在至少一个实施例中，URB 2918可以存储多个线程的数据。在至少一个实施例中，URB 2918可以用于在图形核心阵列2914上执行的不同线程之间发送数据。在至少一个实施例中，URB 2918还可用于图形核心阵列2914上的线程与共享功能逻辑2920内的固定功能逻辑之间的同步。

[0477] 在至少一个实施例中，图形核心阵列2914是可缩放的，使得图形核心阵列2914包括可变数量的图形核心，每个图形核心具有基于GPE 2910的目标功率和性能水平的可变数量的执行单元。在至少一个实施例中，执行资源是动态可伸缩的，使得执行资源可以根据需要被启用或禁用。

[0478] 在至少一个实施例中，图形核心阵列2914耦合到共享功能逻辑2920，该共享功能逻辑包括在图形核心阵列2914中的图形核心之间共享的多个资源。在至少一个实施例中，由共享功能逻辑2920执行的共享功能体现在向图形核心阵列2914提供专门的补充功能的硬件逻辑单元中。在至少一个实施例中，共享功能逻辑2920包括但不限于采样器单元2921、数学单元2922和线程间通信(ITC)逻辑2923。在至少一个实施例中，一个或更多个高速缓存

2925被包含在或耦合到共享功能逻辑2920中。

[0479] 在至少一个实施例中，如果对专用功能的需求不足以包含在图形核心阵列2914中，则使用共享功能。在至少一个实施例中，专用功能的单个实例在共享功能逻辑2920中使用，并且在图形核心阵列2914内的其他执行资源之间共享。在至少一个实施例中，特定共享功能可以包括在图形核心阵列2914内的共享功能逻辑2926内，所述特定共享功能在图形核心阵列2914广泛使用的共享功能逻辑2920内。在至少一个实施例中，图形核心阵列2914内的共享功能逻辑2926可包括共享功能逻辑2920内的一些或全部逻辑。在至少一个实施例中，共享功能逻辑2920内的所有逻辑元件可在图形核心阵列2914的共享功能逻辑2926内复制。在至少一个实施例中，排除共享功能逻辑2920，以支持图形核心阵列2914内的共享功能逻辑2926。

[0480] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，部分或全部推理和/或训练逻辑715可以结合到图形处理器2600中。例如，在至少一个实施例中，本文描述的训练和/或推理技术可以使用一个或更多个ALU，所述ALU体现在3D管线2912、图形核心2915、共享功能逻辑2926、共享功能逻辑2920或图29中的其他逻辑中。此外，在至少一个实施例中，本文描述的推理和/或训练操作可以使用除图7A或图7B所示的逻辑之外的逻辑来完成。在至少一个实施例中，权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中，其配置图形处理器2910的ALU，以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0481] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他装置或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0482] 图30是根据本文所述的至少一个实施例的图形处理器核心3000的硬件逻辑的框图。在至少一个实施例中，图形处理器核心3000被包括在图形核心阵列内。在至少一个实施例中，图形处理器核心3000(有时称为核心切片)可以是模块化图形处理器内的一个或更多个图形核心。在至少一个实施例中，图形处理器核心3000是一个图形核心切片的示例，并且本文所述的图形处理器可以基于目标功率和性能包络线包括多个图形核心切片。在至少一个实施例中，每个图形核心3000可以包括与多个子核心3001A-3001F耦合的固定功能块3030，也称为子切片，其包括通用和固定功能逻辑的模块块。

[0483] 在至少一个实施例中，固定功能块3030包括几何和固定功能管线3036，例如，在较低性能和/或较低功率的图形处理器实施方式中，该几何和固定功能管线3036可以由图形处理器3000中的所有子核心共享。在至少一个实施例中，几何和固定功能管线3036包括3D固定功能管线、视频前端单元，线程产生器和线程分派器以及管理统一返回缓冲区的统一返回缓冲区管理器。

[0484] 在固定的至少一个实施例中，固定功能块3030还包括图形SoC接口3037、图形微控制器3038和媒体管线3039。在至少一个实施例中，图形SoC接口3037提供了图形核心3000以

及片上集成电路系统中的其他处理器核心之间的接口。在至少一个实施例中，图形微控制器3038是可编程子处理器，其可配置为管理图形处理器3000的各种功能，包括线程分派、调度和抢占。在至少一个实施例中，媒体管线3039包括有助于对包括图像和视频数据的多媒体数据进行解码、编码、预处理和/或后处理的逻辑。在至少一个实施例中，媒体管线3039经由对子核心3001-3001F内的计算或采样逻辑的请求来实现媒体操作。

[0485] 在至少一个实施例中，SoC接口3037使图形核心3000能够与通用应用程序处理器核心(例如，CPU)和/或SoC内的其他组件通信，包括存储器层次结构元素，诸如共享的最后一级高速缓存、系统RAM和/或嵌入式片上或封装DRAM。在至少一个实施例中，SoC接口3037还可以使得能够与SoC内的固定功能设备(例如，相机成像管线)进行通信，并且使得能够使用和/或实现可以在图形核心3000和SoC内部的CPU之间共享的全局存储器原子。在至少一个实施例中，图形SoC接口3037还可以实现用于图形处理器核心3000的电源管理控制，并且启用图形处理器核心3000的时钟域与SoC内的其他时钟域之间的接口。在至少一个实施例中，SoC接口3037使得能够从命令流转化器和全局线程分派器接收命令缓冲区，其配置为向图形处理器内的一个或更多个图形核心中的每一个提供命令和指令。在至少一个实施例中，当要执行媒体操作时，可以将命令和指令分派给媒体管线3039，或者当要执行图形处理操作时，可以将其分配给几何形状和固定功能管线(例如，几何形状和固定功能管线3036，和/或几何形状和固定功能管线3014)。

[0486] 在至少一个实施例中，图形微控制器3038可以配置为对图形核心3000执行各种调度和管理任务。在至少一个实施例中，图形微控制器3038可以在子核心3001A-3001F中的执行单元(EU)阵列3002A-3002F、3004A-3004F内的各种图形并行引擎上执行图形和/或计算工作负载调度。在至少一个实施例中，在包括图形核心3000的SoC的CPU核心上执行的主机软件可以提交多个图形处理器路径之一的工作负载，其调用适当的图形引擎上的调度操作。在至少一个实施例中，调度操作包括确定接下来要运行哪个工作负载、将工作负载提交给命令流转化器、抢先在引擎上运行的现有工作负载、监控工作负载的进度以及在工作负载完成时通知主机软件。在至少一个实施例中，图形微控制器3038还可以促进图形核心3000的低功率或空闲状态，从而为图形核心3000提供在图形核心3000内独立于操作系统和/或系统上的图形驱动程序软件的跨低功率状态转换的保存和恢复寄存器的能力。

[0487] 在至少一个实施例中，图形核心3000可以具有比所示的子核心3001A-3001F多或少达N个模块化子核心。对于每组N个子核心，在至少一个实施例中，图形核心3000还可以包括共享功能逻辑3010、共享和/或高速缓存存储器3012、几何/固定功能管线3014以及附加的固定功能逻辑3016以加速各种图形和计算处理操作。在至少一个实施例中，共享功能逻辑3010可以包括可由图形核心3000内的每个N个子核心共享的逻辑单元(例如，采样器、数学和/或线程间通信逻辑)。在至少一个实施例中，共享和/或高速缓存存储器3012可以是图形核心3000内的N个子核心3001A-3001F的最后一级高速缓存，并且还可以用作可由多个子核心访问的共享存储器。在至少一个实施例中，可以包括几何/固定功能管线3014来代替固定功能块3030内的几何/固定功能管线3036，并且可以包括相似的逻辑单元。

[0488] 在至少一个实施例中，图形核心3000包括附加的固定功能逻辑3016，其可以包括供图形核心3000使用的各种固定功能加速逻辑。在至少一个实施例中，附加的固定功能逻辑3016包括用于仅位置着色中使用的附加的几何管线。在仅位置着色中，存在至少两个几

何管线，而在几何和固定功能管线3014、3036内的完整几何管线和剔除管线中，其是可以包括在附加的固定功能逻辑3016中的附加几何管线。在至少一个实施例中，剔除管线是完整几何管线的修整版。在至少一个实施例中，完整管线和剔除管线可以执行应用程序的不同实例，每个实例具有单独的环境。在至少一个实施例中，仅位置着色可以隐藏被丢弃的三角形的长剔除运行，从而在某些情况下可以更早地完成着色。例如，在至少一个实施例中，附加固定功能逻辑3016中的剔除管线逻辑可以与主应用程序并行执行位置着色器，并且通常比完整管线更快地生成关键结果，因为剔除管线获取并遮蔽顶点的位置属性，无需执行光栅化和将像素渲染到帧缓冲区。在至少一个实施例中，剔除管线可以使用生成的临界结果来计算所有三角形的可见性信息，而与这些三角形是否被剔除无关。在至少一个实施例中，完整管线(在这种情况下可以称为重播管线)可以消耗可见性信息来跳过剔除的三角形以仅遮盖最终传递到光栅化阶段的可见三角形。

[0489] 在至少一个实施例中，附加的固定功能逻辑3016还可包括机器学习加速逻辑，例如固定功能矩阵乘法逻辑，用于实现包括用于机器学习训练或推理的优化。

[0490] 在至少一个实施例中，在每个图形子核心3001A-3001F内包括一组执行资源，其可用于响应于图形管线、媒体管线或着色器程序的请求来执行图形、媒体和计算操作。在至少一个实施例中，图形子核心3001A-3001F包括多个EU阵列3002A-3002F、3004A-3004F，线程分派和线程间通信(TD/IC)逻辑3003A-3003F，3D(例如，纹理)采样器3005A-3005F，媒体采样器3006A-3006F，着色器处理器3007A-3007F和共享本地存储器(SLM)3008A-3008F。在至少一个实施例中，EU阵列3002A-3002F、3004A-3004F每个都包含多个执行单元，这些执行单元是通用图形处理单元，能够为图形、媒体或计算操作提供服务，执行浮点和整数/定点逻辑运算，包括图形、媒体或计算着色器程序。在至少一个实施例中，TD/IC逻辑3003A-3003F为子核心内的执行单元执行本地线程分派和线程控制操作，并促进在子核心的执行单元上执行的线程之间的通信。在至少一个实施例中，3D采样器3005A-3005F可以将与纹理或其他3D图形相关的数据读取到存储器中。在至少一个实施例中，3D采样器可以基于与给定纹理相关联的配置的采样状态和纹理格式来不同地读取纹理数据。在至少一个实施例中，媒体采样器3006A-3006F可以基于与媒体数据相关联的类型和格式来执行类似的读取操作。在至少一个实施例中，每个图形子核心3001A-3001F可以可替代地包括统一的3D和媒体采样器。在至少一个实施例中，在每个子核心3001A-3001F内的执行单元上执行的线程可以利用每个子核心内的共享本地存储器3008A-3008F，以使在线程组内执行的线程能够使用片上存储器的公共池来执行。

[0491] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715的部分或全部可以被合并到图形处理器3010中。例如，在至少一个实施例中，本文描述的训练和/或推理技术可以使用在3D管线、图形微控制器3038、几何和固定功能管线3014和3036或图30中的其他逻辑中体现的一个或更多个ALU。此外，在至少一个实施例中，本文描述的推理和/或训练操作可以使用除图7A或图7B所示的逻辑以外的逻辑来完成。在至少一个实施例中，权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中，其配置图形处理器3000的ALU以执行一种或更多种本文介绍的机器学习算法、神经网络架构、用例或训练技术。

[0492] 在至少一个实施例中，一个或更多个电路、处理器、计算系统或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0493] 图31A-31B示出了根据至少一个实施例的包括图形处理器核心的处理元件的阵列的线程执行逻辑3100。图31A示出了至少一个实施例，其中使用了线程执行逻辑3100。图31B示出了根据至少一个实施例的图形执行单元3108的示例性内部细节。

[0494] 如图31A中所示，在至少一个实施例中，线程执行逻辑3100包括着色器处理器3102、线程分派器3104、指令高速缓存3106、包括多个执行单元3107A-3107N和3108A-3108N的可缩放执行单元阵列、采样器3110、数据高速缓存3112和数据端口3114。在至少一个实施例中，可缩放执行单元阵列可以例如基于工作负载的计算要求，通过启用或禁用一个或更多个执行单元(例如，执行单元3108A-N或3107A-N中的任意一个)来动态缩放。在至少一个实施例中，可缩放执行单元通过链路到每个执行单元的互连结构互连。在至少一个实施例中，线程执行逻辑3100包括通过指令高速缓存3106、数据端口3114、采样器3110和执行单元3107或3108中的一个或更多个到存储器(诸如系统存储器或高速缓存存储器)的一个或更多个连接。在至少一个实施例中，每个执行单元(例如3107A)是独立的可编程通用计算单元，其能够执行多个同时的硬件线程，同时针对每个线程并行处理多个数据元素。在至少一个实施例中，执行单元3107和/或3108的阵列可缩放以包括任意数量的单独执行单元。

[0495] 在至少一个实施例中，执行单元3107和/或3108主要用于执行着色器程序。在至少一个实施例中，着色器处理器3102可以处理各种着色器程序并经由线程分派器3104来分派与着色器程序相关联的执行线程。在至少一个实施例中，线程分派器3104包括用于仲裁来自图形和媒体管线的线程初始化庆祝以及在执行单元3107和/或3108中的一个或更多个执行单元上实例化请求的线程的逻辑。例如，在至少一个实施例中，几何管线可以将顶点、镶嵌或几何着色器分派到线程执行逻辑以进行处理。在至少一个实施例中，线程分派器3104还可以处理来自执行着色器程序的运行时线程产生请求。

[0496] 在至少一个实施例中，执行单元3107和/或3108支持一种指令集，该指令集包括对许多标准3D图形着色器指令的本机支持，从而使图形库(例如Direct 3D和OpenGL)中的着色器程序只需最少的转换即可执行。在至少一个实施例中，执行单元支持顶点和几何处理(例如，顶点程序、几何程序、和/或顶点着色器)、像素处理(例如，像素着色器、片段着色器)和通用处理(例如，计算和媒体着色器)。在至少一个实施例中，每个执行单元3107和/或3108包括一个或更多个算术逻辑单元(ALU)，能够执行多发出单指令多数据(SIMD)，并且多线程操作实现了高效的执行环境尽管有更高的延迟存储器访问。在至少一个实施例中，每个执行单元内的每个硬件线程具有专用的高带宽寄存器文件和相关的独立线程状态。在至少一个实施例中，执行是每个时钟到管线的多次发出，管线能够进行整数、单精度和双精度浮点运算、SIMD分支功能、逻辑运算、先验运算和其他其他运算。在至少一个实施例中，在等待来自存储器或共享功能之一的数据时，执行单元3107和/或3108内的依赖性逻辑使等待线程休眠直到返回了所请求的数据。在至少一个实施例中，当等待线程正在休眠时，硬件资

源可以专用于处理其他线程。例如,在至少一个实施例中,在与顶点着色器操作相关联的延迟期间,执行单元可以对像素着色器、片段着色器或另一类型的着色器程序(包括不同的顶点着色器)执行操作。

[0497] 在至少一个实施例中,执行单元3107和/或3108中的每一个执行单元在数据元素的阵列上进行操作。在至少一个实施例中,多个数据元素是"执行大小"或指令的通道数。在至少一个实施例中,执行通道是用于指令内的数据元素访问、屏蔽和流控制的执行的逻辑单元。在至少一个实施例中,多个通道可以独立于用于特定图形处理器的多个物理算术逻辑单元(ALU)或浮点单元(FPU)。在至少一个实施例中,执行单元3107和/或3108支持整数和浮点数据类型。

[0498] 在至少一个实施例中,执行单元指令集包括SIMD指令。在至少一个实施例中,各种数据元素可以作为封装数据类型存储在寄存器中,并且执行单元将基于那些元素的数据大小来处理各种元素。例如,在至少一个实施例中,当对256位宽的向量进行操作时,将向量的256位存储在寄存器中,并且执行单元对向量进行操作,作为四个单独的64位封装数据元素(四字(QW)大小数据元素)、八个单独的32位封装数据元素(双字(DW)大小数据元素)、十六个单独的16位封装数据元素(单词(W)大小数据元素)或三十二个单独的8位数据元素(字节(B)大小的数据元素)。然而,在至少一个实施例中,不同的向量宽度和寄存器大小是可能的。

[0499] 在至少一个实施例中,一个或更多个执行单元可以被组合成具有执行对于融合EU的线程控制逻辑(3111A-3111N)的融合执行单元3109A-3109N,例如将执行单元3107A与执行单元3108A融合为融合执行单元3109A中。在至少一个实施例中,可以将多个EU合并成一个EU组。在至少一个实施例中,融合EU组中的EU的数量可以配置为执行单独的SIMD硬件线程,融合的EU组中的EU的数量可能根据各个实施例而变化。在至少一个实施例中,每个EU可以执行各种SIMD宽度,包括但不限于SIMD8、SIMD16和SIMD32。在至少一个实施例中,每个融合图形执行单元3109A-3109N包括至少两个执行单元。例如,在至少一个实施例中,融合执行单元3109A包括第一EU 3107A、第二EU 3108A以及第一EU 3107A和第二EU 3108A共有的线程控制逻辑3111A。在至少一个实施例中,线程控制逻辑3111A控制在融合图形执行单元3109A上执行的线程,从而允许融合执行单元3109A-3109N内的每个EU使用公共指令指针寄存器来执行。

[0500] 在至少一个实施例中,一个或更多个内部指令高速缓存(例如3106)被包括在线程执行逻辑3100中以高速缓存用于执行单元的线程指令。在至少一个实施例中,包括一个或更多个数据高速缓存(例如3112)以在线程执行期间高速缓存线程数据。在至少一个实施例中,包括采样器3110以提供用于3D操作的纹理采样和用于媒体操作的媒体采样。在至少一个实施例中,采样器3110包括专门的纹理或媒体采样功能,以在将采样数据提供给执行单元之前在采样过程中处理纹理或媒体数据。

[0501] 在执行期间,在至少一个实施例中,图形和媒体管线通过线程产生和分派逻辑将线程发起请求发送到线程执行逻辑3100。在至少一个实施例中,一旦一组几何对象已经被处理并光栅化成像素数据,则在着色器处理器3102内的像素处理器逻辑(例如,像素着色器逻辑、片段着色器逻辑等)被调用以进一步计算输出信息并且导致将结果写入输出表面(例如,颜色缓冲区、深度缓冲区、模板缓冲区等)。在至少一个实施例中,像素着色器或片段着

色器计算要在光栅化对象上插值的各种顶点属性的值。在至少一个实施例中，着色器处理器3102内的像素处理器逻辑然后执行应用程序接口(API)提供的像素或片段着色器程序。在至少一个实施例中，为了执行着色器程序，着色器处理器3102经由线程分派器3104将线程分派到执行单元(例如3108A)。在至少一个实施例中，着色器处理器3102使用采样器3110中的纹理采样逻辑来访问存储在存储器中的纹理贴图中的纹理数据。在至少一个实施例中，对纹理数据和输入几何数据的算术运算为每个几何片段计算像素颜色数据，或者丢弃一个或更多个像素以进行进一步处理。

[0502] 在至少一个实施例中，数据端口3114提供了一种用于线程执行逻辑3100的存储器访问机制，以将处理后的数据输出到存储器以在图形处理器输出管线上进行进一步处理。在至少一个实施例中，数据端口3114包括或耦合到一个或更多个高速缓存存储器(例如，数据高速缓存3112)以高速缓存数据以便经由数据端口进行存储器访问。

[0503] 如图31B所示，在至少一个实施例中，图形执行单元3108可以包括指令获取单元3137、通用寄存器文件阵列(GRF)3124、架构寄存器文件阵列(ARF)3126、线程仲裁器3122、发送单元3130、分支单元3132、一组SIMD浮点单元(FPU)3134，以及在至少一个实施例中，一组专用整数SIMD ALU 3135。GRF 3124和ARF 3126包括一组与可以在图形执行单元3108中活跃的每个同时硬件线程相关联的通用寄存器文件和架构寄存器文件。在至少一个实施例中，在ARF 3126中维护每个线程架构状态，而在线程执行期间使用的数据存储在GRF 3124中。在至少一个实施例中，每个线程的执行状态，包括每个线程的指令指针，可以被保存在ARF 3126中的线程专用寄存器中。

[0504] 在至少一个实施例中，图形执行单元3108具有一种架构，该架构是同时多线程(SMT)和细粒度交错多线程(IMT)的组合。在至少一个实施例中，架构具有模块化配置，该模块化配置可以在设计时基于同时线程的目标数量和每个执行单元的寄存器数量来进行微调，其中执行单元资源在用于执行多个同时线程的逻辑上分配。

[0505] 在至少一个实施例中，图形执行单元3108可以共同发布多个指令，每个指令可以是不同的指令。在至少一个实施例中，图形执行单元线程3108的线程仲裁器3122可以将指令分派到发送单元3130、分支单元3132或SIMD FPU 3134之一以供执行。在至少一个实施例中，每个执行线程可以访问GRF 3124中的128个通用寄存器，其中每个寄存器可以存储32个字节，可以作为32位数据元素的SIMD 8元素向量进行访问。在至少一个实施例中，每个执行单元线程可以访问GRF 3124中的4KB，尽管实施例不限于此，并且在其他实施例中可以提供更多或更少的寄存器资源。在至少一个实施例中，尽管每个执行单元的线程数量也可以根据实施例而变化，但是最多可以同时执行七个线程。在其中七个线程可以访问4KB的至少一个实施例中，GRF 3124可以存储总共28KB。在至少一个实施例中，灵活的寻址模式可以允许将寄存器一起寻址以有效地建立更宽的寄存器或表示跨步的矩形块数据结构。

[0506] 在至少一个实施例中，经由由消息传递发送单元3130执行的“发送”指令来调度存储器操作、采样器操作和其他更长延迟的系统通信。在至少一个实施例中，将分支指令分派到分支单元3132促进SIMD发散和最终收敛。

[0507] 在至少一个实施例中，图形执行单元3108包括一个或更多个SIMD浮点单元(FPU)3134，以执行浮点操作。在至少一个实施例中，一个或更多个FPU 3134还支持整数计算。在至少一个实施例中，一个或更多个FPU 3134可以SIMD执行多达M个32位浮点(或整数)运算，

或者SIMD执行多达2M个16位整数或16位浮点运算。在至少一个实施例中，至少一个FPU提供扩展的数学能力以支持高吞吐量的先验数学函数和双精度64位浮点。在至少一个实施例中，还存在一组8位整数SIMD ALU 3135，并且可以被专门优化以执行与机器学习计算相关的操作。

[0508] 在至少一个实施例中，可以在图形子核心分组(例如，子切片)中实例化图形执行单元3108的多个实例的阵列。在至少一个实施例中，执行单元3108可以跨多个执行通道执行指令。在至少一个实施例中，在图形执行单元3108上执行的每个线程在不同的通道上执行。

[0509] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。下面结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，推理和/或训练逻辑715的部分或全部可以被结合到线程执行逻辑3100中。此外，在至少一个实施例中，可以使用除了图7A或图7B中所示的逻辑之外的逻辑来完成在此描述的推理和/或训练操作。在至少一个实施例中，权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中，其配置线程执行逻辑3100的ALU以执行一种或更多种机器学习算法、神经网络架构、用例或本文介绍的训练技术。

[0510] 在至少一个实施例中，一个或更多个电路、处理器、计算系统或其他设备或技术被适配成用于参考所述图生成一个倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0511] 图32示出了根据至少一个实施例的并行处理单元(“PPU”)3200。在至少一个实施例中，PPU 3200配置有机器可读代码，该机器可读代码如果由PPU 3200执行，则使得PPU 3200执行贯穿本公开描述的一些或全部过程和技术。在至少一个实施例中，PPU 3200是在一个或更多个集成电路设备上实现的多线程处理器，并且利用多线程作为被设计为处理在多个线程上并行执行的计算机可读指令(也称为机器可读指令或简单的指令)的延迟隐藏技术。在至少一个实施例中，线程是指执行线程，并且是被配置为由PPU 3200执行的一组指令的实例。在至少一个实施例中，PPU 3200是图形处理单元(“GPU”)，图形处理单元配置为实现用于处理三维(“3D”)图形数据的图形渲染管线，以便生成用于在显示设备(诸如液晶显示器(“LCD”)设备)上显示的二维(“2D”)图像数据。在至少一个实施例中，PPU 3200用于执行计算，诸如线性代数运算和机器学习运算。图32仅出于说明性目的示出了示例并行处理器，并且应被解释为在本公开的范围内设想的处理器架构的非限制性示例，并且可以采用任何适当的处理器来对其进行补充和/或替代。

[0512] 在至少一个实施例中，一个或更多个PPU 3200配置成加速高性能计算(“HPC”)、数据中心和机器学习应用程序。在至少一个实施例中，PPU 3200配置成加速深度学习系统和应用程序，包括以下非限制性示例：自动驾驶汽车平台、深度学习、高精度语音、图像、文本识别系统、智能视频分析、分子模拟、药物发现、疾病诊断、天气预报、大数据分析、天文学、分子动力学模拟、财务建模、机器人技术、工厂自动化、实时语言翻译、在线搜索优化以及个性化用户推荐等。

[0513] 在至少一个实施例中，PPU 3200包括但不限于输入/输出(“I/O”)单元3206、前端单元3210、调度器单元3212、工作分配单元3214、集线器3216、交叉开关(“Xbar”)3220、一个或更多个通用处理集群(“GPC”)3218和一个或更多个分区单元(“存储器分区单元”)3222。在至少一个实施例中，PPU 3200通过一个或更多个高速GPU互连(“GPU互连”)3208连接到主机处理器或其他PPU 3200。在至少一个实施例中，PPU 3200通过系统总线3202连接到主机处理器或其他外围设备。在一实施例中，PPU 3200连接到包括一个或更多个存储器设备(“存储器”)3204的本地存储器。在至少一个实施例中，存储器设备3204包括但不限于一个或更多个动态随机存取存储器(“DRAM”)设备。在至少一个实施例中，一个或更多个DRAM设备配置和/或可配置为高带宽存储器(“HBM”)子系统，并且在每个设备内堆叠有多个DRAM管芯。

[0514] 在至少一个实施例中，高速GPU互连3208可以指代系统使用其来进行缩放的基于线的多通道通信链路，并包括与一个或更多个中央处理单元结合的一个或更多个PPU 3200(“CPU”)，支持PPU 3200和CPU之间的缓存相干以及CPU主控。在至少一个实施例中，高速GPU互连3208通过集线器3216将数据和/或命令传输到PPU 3200的其他单元，例如一个或更多个复制引擎、视频编码器、视频解码器、电源管理单元和/或在图32中可能未明确示出的其他组件。

[0515] 在至少一个实施例中，I/O单元3206配置为通过系统总线3202从主机处理器(图32中未示出)发送和接收通信(例如，命令、数据)。在至少一个实施例中，I/O单元3206直接通过系统总线3202或通过一个或更多个中间设备(例如存储器桥)与主机处理器通信。在至少一个实施例中，I/O单元3206可以经由系统总线3202与一个或更多个其他处理器(例如一个或更多个PPU 3200)通信。在至少一个实施例中，I/O单元3206实现外围组件互连Express(“PCIe”)接口，用于通过PCIe总线进行通信。在至少一个实施例中，I/O单元3206实现用于与外部设备通信的接口。

[0516] 在至少一个实施例中，I/O单元3206对经由系统总线3202接收的分组进行解码。在至少一个实施例中，至少一些分组表示被配置为使PPU 3200执行各种操作的命令。在至少一个实施例中，I/O单元3206如命令所指定的那样将解码的命令发送到PPU 3200的各种其他单元。在至少一个实施例中，命令被发送到前端单元3210和/或被发送到集线器3216或PPU 3200的其他单元，例如一个或更多个复制引擎、视频编码器、视频解码器、电源管理单元等(图32中未明确示出)。在至少一个实施例中，I/O单元3206配置为在PPU 3200的各种逻辑单元之间路由通信。

[0517] 在至少一个实施例中，由主机处理器执行的程序在缓冲区中对命令流进行编码，该缓冲区将工作负载提供给PPU 3200以进行处理。在至少一个实施例中，工作负载包括指令和要由那些指令处理的数据。在至少一个实施例中，缓冲区是可由主机处理器和PPU 3200两者访问(例如，读/写)的存储器中的区域—主机接口单元可以配置为访问经由I/O单元3206通过系统总线3202传输的存储器请求连接到系统总线3202的系统存储器中的缓冲区。在至少一个实施例中，主机处理器将命令流写入缓冲区，然后将指示命令流开始的指针发送给PPU 3200，使得前端单元3210接收指向一个或更多个命令流指针并管理一个或更多个命令流，从命令流中读取命令并将命令转发到PPU 3200的各个单元。

[0518] 在至少一个实施例中，前端单元3210耦合到调度器单元3212，该调度器单元3212

配置各种GPC 3218以处理由一个或更多个命令流定义的任务。在至少一个实施例中，调度器单元3212配置为跟踪与调度器单元3212管理的各种任务有关的状态信息，其中状态信息可以指示任务被分配给哪个GPC 3218，任务是活跃的还是非活跃的，与任务相关联的优先级等等。在至少一个实施例中，调度器单元3212管理在一个或更多个GPC 3218上执行的多个任务。

[0519] 在至少一个实施例中，调度器单元3212耦合到工作分配单元3214，该工作分配单元3214配置为分派任务以在GPC 3218上执行。在至少一个实施例中，工作分配单元3214跟踪从调度器单元3212接收到的多个调度任务并且工作分配单元3214管理每个GPC 3218的待处理任务池和活跃任务池。在至少一个实施例中，待处理任务池包括多个时隙(例如32个时隙)，这些时隙包含分配给要由特定的GPC 3218处理的任务；活跃任务池可包括用于由GPC 3218主动处理的任务的多个时隙(例如4个时隙)，以使随着GPC 3218中的一个完成任务的执行，该任务将从GPC 3218的活动任务池中逐出，并且从待处理任务池中选择另一个任务，并安排其在GPC 3218上执行。在至少一个实施例中，如果活跃任务在GPC 3218上处于空闲状态，例如在等待数据依赖性解决时，则活跃任务从GPC 3218中驱逐并返回到待处理任务池，同时选择了待处理任务池中的另一个任务并调度在GPC 3218上执行。

[0520] 在至少一个实施例中，工作分配单元3214经由XBar 3220与一个或更多个GPC 3218通信。在至少一个实施例中，XBar 3220是互连网络，其将PPU 3200的许多单元耦合到PPU 3200的其他单元，并且可以配置为将工作分配单元3214耦合到特定的GPC 3218。在至少一个实施例中，一个或更多个PPU 3200的其他单元也可以通过集线器3216连接到XBar 3220。

[0521] 在至少一个实施例中，任务由调度器单元3212管理，并由工作分配单元3214分配给GPC 3218之一。在至少一个实施例中，GPC 3218配置为处理任务并产生结果。在至少一个实施例中，结果可以由GPC 3218中的其他任务消耗，通过XBar 3220路由到不同的GPC 3218或存储在存储器3204中。在至少一个实施例中，结果可以通过分区单元3222写到存储器3204中，其实现了用于向存储器3204写入数据或从存储器3204读取数据的存储器接口。在至少一个实施例中，结果可以经由高速GPU互连3208传输到另一PPU 3204或CPU。在至少一个实施例中，PPU 3200包括但不限于U个分区单元3222，其等于耦合到PPU 3200的分离且不同的存储器设备3204的数量，本文结合图34更详细地描述。

[0522] 在至少一个实施例中，主机处理器执行驱动器核心，该驱动程序核心实现应用程序编程接口(API)，该应用程序编程接口使在主机处理器上执行的一个或更多个应用程序能够调度操作以在PPU 3200上执行。在一个实施例中，多个计算应用程序由PPU 3200同时执行，并且PPU 3200为多个计算应用程序提供隔离、服务质量(“QoS”)和独立的地址空间。在至少一个实施例中，应用程序生成指令(例如，以API调用的形式)，该指令使驱动器核心生成一个或更多个任务以供PPU 3200执行，并且驱动器核心将任务输出至由PPU 3200处理的一个或更多个流。在至少一个实施例中，每个任务包括一个或更多个相关线程组，其可以被称为线程束(warp)。在至少一个实施例中，线程束包括可以并行执行的多个相关线程(例如32个线程)。在至少一个实施例中，协作线程可以指代多个线程，包括用于执行任务并且通过共享存储器交换数据的指令，结合图34根据至少一个实施例更详细地描述了线程和协作线程。

[0523] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，深度学习应用程序处理器用于训练机器学习模型(诸如神经网络)，以预测或推理提供给PPU 3200的信息。在至少一个实施例中，深度学习应用程序处理器3200用于基于已由另一处理器或系统或PPU 3200训练过的训练过的机器学习模型(例如，神经网络)推理或预测信息。在至少一个实施例中，PPU 3200可用于执行本文所述的一个或更多个神经网络用例。

[0524] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0525] 图33示出了根据至少一个实施例的通用处理集群(“GPC”)3300。在至少一个实施例中，GPC 3300是图32的GPC 3218。在至少一个实施例中，每个GPC 3300包括但不限于用于处理任务的多个硬件单元，并且每个GPC 3300包括但不限于管线管理器3302、预光栅操作单元(“preROP”)3304、光栅引擎3308、工作分配交叉开关(“WDX”)3316、存储器管理单元(“MMU”)3318、一个或更多个数据处理集群(“DPC”)3306，以及部件的任何合适组合。

[0526] 在至少一个实施例中，GPC 3300的操作由管线管理器3302控制。在至少一个实施例中，管线管理器3302管理一个或更多个DPC 3306的配置，以处理分配给GPC 3300的任务。在至少一个实施例中，管线管理器3302配置一个或更多个DPC 3306中的至少一个以实现图形渲染管线的至少一部分。在至少一个实施例中，DPC 3306配置为在可编程流式多处理器(“SM”)3314上执行顶点着色器程序。在至少一个实施例中，管线管理器3302配置为将从工作分配单元接收的数据包路由到GPC 3300内的适当逻辑单元，以及在至少一个实施例中，可以将一些数据包路由到preROP 3304和/或光栅引擎3308中的固定功能硬件单元，而可以将其他数据包路由到DPC 3306以由原始引擎3312或SM 3314进行处理。在至少一个实施例中，管线管理器3302配置DPC 3306中的至少一个以实现神经网络模型和/或计算管线。

[0527] 在至少一个实施例中，preROP单元3304配置为在至少一个实施例中将由光栅引擎3308和DPC 3306生成的数据路由到分区单元3222中的光栅操作(“ROP”)单元，上面结合图32更详细地描述。在至少一个实施例中，preROP单元3304配置为执行用于颜色混合的优化、组织像素数据、执行地址转换等等。在至少一个实施例中，光栅引擎3308包括但不限于配置为执行各种光栅操作的多个固定功能硬件单元，并且在至少一个实施例中，光栅引擎3308包括但不限于设置引擎、粗光栅引擎、剔除引擎、裁剪引擎、精细光栅引擎、图块聚合引擎及其任意合适的组合。在至少一个实施例中，设置引擎接收变换后的顶点并生成与由顶点定义的几何图元相关联的平面方程；平面方程式被传送到粗光栅引擎以生成基本图元的覆盖信息(例如，图块的x、y覆盖范围掩码)；粗光栅引擎的输出将传输到剔除引擎，在剔除引擎中与z测试失败的图元相关联的片段将被剔除，并传输到剪切引擎，在剪切引擎中剪切位于视锥范围之外的片段。在至少一个实施例中，将经过裁剪和剔除的片段传递给精细光栅引擎，以基于设置引擎生成的平面方程式生成像素片段的属性。在至少一个实施例中，光栅引擎3308的输出包括将由任何适当的实体(例如，由在DPC 3306内实现的片段着色器)处理的

片段。

[0528] 在至少一个实施例中，包括在GPC 3300中的每个DPC 3306包括但不限于M管线控制器(“MPC”)3310；图元引擎3312；一个或更多个SM 3314；及其任何合适的组合。在至少一个实施例中，MPC 3310控制DPC3306的操作，将从管线管理器3302接收的分组路由到DPC 3306中的适当单元。在至少一个实施例中，将与顶点相关联的分组路由到图元引擎3312，图元引擎3312配置为从存储器中获取与顶点关联的顶点属性；相反，可以将与着色器程序相关联的数据包发送到SM 3314。

[0529] 在至少一个实施例中，SM 3314包括但不限于可编程流式处理器，其配置为处理由多个线程表示的任务。在至少一个实施例中，SM 3314是多线程的并且配置为同时执行来自特定线程组的多个线程(例如32个线程)，并且实现单指令、多数据(“SIMD”)架构，其中将一组线程(例如，线程束)中的每个线程配置为基于相同的指令集来处理不同的数据集。在至少一个实施例中，线程组中的所有线程执行通用指令集。在至少一个实施例中，SM 3314实施单指令、多线程(“SIMT”)架构，其中一组线程中的每个线程配置为基于通用指令集来处理不同的数据集，但是其中线程组中的各个线程允许在执行期间发散。在至少一个实施例中，为每个线程束维护程序计数器、调用栈和执行状态，从而当线程束中的线程发散时，实现线程束和线程束内的串行执行之间的并发性。在另一个实施例中，为每个单独的线程维护程序计数器、调用栈和执行状态，从而使得在线程束内和线程束之间的所有线程之间具有相等的并发性。在至少一个实施例中，为每个单独的线程维持执行状态，并且可以收敛并并行地执行执行通用指令的线程以提高效率。本文更详细地描述SM 3314的至少一个实施例。

[0530] 在至少一个实施例中，MMU 3318在GPC 3300和存储器分区单元(例如，图32的分区单元3222)之间提供接口，并且MMU 3318提供虚拟地址到物理地址的转换、存储器保护以及存储器请求的仲裁。在至少一个实施例中，MMU 3318提供一个或更多个转换后备缓冲区(“TLB”)，用于执行虚拟地址到存储器中的物理地址的转换。

[0531] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，深度学习应用程序处理器用于训练机器学习模型(诸如神经网络)，以预测或推理提供给GPC 3300的信息。在至少一个实施例中，GPC 3300用于基于已由另一处理器或系统或GPC 3300训练过的机器学习模型(例如，神经网络)推理或预测信息。在至少一个实施例中，GPC 3300可用于执行本文所述的一个或更多个神经网络用例。

[0532] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0533] 图34示出了根据至少一个实施例的并行处理单元(“PPU”)的存储器分区单元3400。在至少一个实施例中，存储器分区单元3400包括但不限于光栅操作(“ROP”)单元3402；二级(“L2”)高速缓存3404；存储器接口3406；及其任何合适的组合。在至少一个实施

例中，存储器接口3406耦合到存储器。在至少一个实施例中，存储器接口3406可以实现32、64、128、1024位数据总线，或者类似的实现方式用于高速数据传输。在至少一个实施例中，PPU包括U个存储器接口3406，其中U是正整数，每对分区单元3400一个存储器接口3406，其中每对分区单元3400连接到对应的存储器设备。例如，在至少一个实施例中，PPU可以连接至多达Y个存储器设备，例如高带宽存储器堆栈或图形双数据速率版本5同步动态随机存取存储器(“GDDR5 SDRAM”)。

[0534] 在至少一个实施例中，存储器接口3406实现高带宽存储器第二代(“HBM2”)存储器接口，并且Y等于U的一半。在至少一个实施例中，HBM2存储器堆栈与PPU一起位于物理封装上，与传统的GDDR5 SDRAM系统相比，可提供大量功率并节省面积。在至少一个实施例中，每个HBM2堆栈包括但不限于四个存储器管芯，且Y＝4，每个HBM2堆栈包括每个管芯两个128位通道，用于总共8个通道和1024位的数据总线宽度。在至少一个实施例中，存储器支持单错误校正双错误检测(“SECDED”)错误校正码(“ECC”)以保护数据。在至少一个实施例中，ECC可以为对数据损坏敏感的计算应用程序提供更高的可靠性。

[0535] 在至少一个实施例中，PPU实现了多级存储器层次结构。在至少一个实施例中，存储器分区单元3400支持统一存储器以为中央处理单元(“CPU”)和PPU存储器提供单个统一虚拟地址空间，从而实现虚拟存储器系统之间的数据共享。在至少一个实施例中，追踪PPU对位于其他处理器上的存储器的访问频率，以确保将存储器页面移动到更频繁地访问页面的PPU的物理存储器。在至少一个实施例中，高速GPU互连3208支持地址转换服务，其允许PPU直接访问CPU的页表，并通过PPU提供对CPU存储器的完全访问。

[0536] 在至少一个实施例中，复制引擎在多个PPU之间或PPU与CPU之间传输数据。在至少一个实施例中，复制引擎可以为未被映射到页表中的地址生成页面错误，并且存储器分区单元3400然后为页面错误提供服务，将地址映射到页表中，之后复制引擎执行传输。在至少一个实施例中，为多个处理器之间的多个复制引擎操作固定(即不可分页)存储器，从而实质上减少了可用存储器。在至少一个实施例中，在硬件页面故障的情况下，可以将地址传递给复制引擎，而无需考虑是否驻留存储器页，并且复制过程是透明的。

[0537] 根据至少一个实施例，来自图32的存储器3204或其他系统存储器的数据由存储器分区单元3400获取，并将其存储在L2高速缓存3404中，L2高速缓存3404位于芯片上并且在各种GPC之间共享。在至少一个实施例中，每个存储器分区单元3400包括但不限于与对应的存储器设备相关联的L2高速缓存的至少一部分。在至少一个实施例中，在GPC内的各个单元中实现较低级别的高速缓存。在至少一个实施例中，图33的每个SM 3314可以实现一级(“L1”)高速缓存，其中L1高速缓存是专用于特定SM 3314的私有存储器，并且从L2高速缓存3404中获取数据并将其存储在每个L1高速缓存中，用于在SM 3314的功能单元中进行处理。在至少一个实施例中，L2高速缓存3404耦合到存储器接口3406和图32所示的XBar3220。

[0538] 在至少一个实施例中，ROP单元3402执行与像素颜色有关的图形光栅操作，诸如颜色压缩、像素混合等。在至少一个实施例中，ROP单元3402结合光栅引擎3308实施深度测试，从光栅引擎3308的剔除引擎接收与像素片段相关联的样本位置的深度。在至少一个实施例中，针对在与片段关联的样本位置的深度缓冲区中的相应深度测试深度。在至少一个实施例中，如果该片段通过了针对该样本位置的该深度测试，则ROP单元3402更新深度缓冲区，并将该深度测试的结果发送给光栅引擎3308。将意识到，分区单元3400的数量可以不同于

GPC的数量,因此,可以在至少一个实施例中将每个ROP单元3402耦合到每个GPC。在至少一个实施例中,ROP单元3402追踪从不同GPC接收到的分组,并且确定ROP单元3402生成的结果是否要通过XBar 3220路由到。

[0539] 图35示出了根据至少一个实施例的流式多处理器(“SM”)3500。在至少一个实施例中,SM 3500是图33的SM。在至少一个实施例中,SM 3500包括但不限于指令高速缓存3502;一个或更多个调度器单元3504;寄存器文件3508;一个或更多个处理核心(“核心”)3510;一个或更多个特殊功能单元(“SFU”)3512;一个或更多个加载/存储单元(“LSU”)3514;互连网络3516;共享存储器/一级(“L1”)高速缓存3518;和/或其任何合适的组合。

[0540] 在至少一个实施例中,工作分配单元调度任务以在并行处理单元(“PPU”)的通用处理集群(“GPC”)上执行,并且每个任务被分配给GPC内部的特定数据处理集群(“DPC”),并且如果任务与着色器程序相关联,则将该任务分配给SM 3500之一。在至少一个实施例中,调度器单元3504从工作分配单元接收任务并管理分配给SM 3500的一个或更多个线程块的指令调度。在至少一个实施例中,调度器单元3504调度线程块以作为并行线程的线程束来执行,其中,每个线程块被分配至少一个线程束。在至少一个实施例中,每个线程束执行线程。在至少一个实施例中,调度器单元3504管理多个不同的线程块,将线程束分配给不同的线程块,然后在每个时钟周期内将来自多个不同的协作组的指令分派给各种功能单元(例如,处理核心3510、SFU 3512和LSU 3514)。

[0541] 在至少一个实施例中,协作组可以指用于组织通信线程组的编程模型,其允许开发者表达线程正在通信的粒度,从而能够表达更丰富、更有效的并行分解。在至少一个实施例中,协作启动API支持线程块之间的同步以执行并行算法。在至少一个实施例中,常规编程模型的应用程序提供了用于同步协作线程的单一、简单的构造:跨线程块的所有线程的屏障(例如,syncthreads()函数)。但是,在至少一个实施例中,程序员可以在小于线程块粒度的情形下来定义线程组,并在所定义的组内进行同步,以实现更高的性能、设计灵活性以及以集合组范围功能接口的形式实现软件重用。在至少一个实施例中,协作组使程序员能够以子块(即,小到单个线程)和多块粒度明确定义线程组,并执行集合操作,例如对协作组中的线程进行同步。在至少一个实施例中,该编程模型支持跨软件边界的干净组合,从而库和实用程序功能可以在其本地环境中安全地同步,而不必进行关于收敛的假设。在至少一个实施例中,协作组图元使协作并行的新图案成为可能,包括但不限于生产者-消费者并行,机会主义并行以及整个线程块网格上的全局同步。

[0542] 在至少一个实施例中,调度单元3506配置为将指令发送到功能单元中的一个或更多个,并且调度器单元3504并包括但不限于两个调度单元3506,该两个调度单元3506使得来自共同线程束的两个不同指令能够在每个时钟周期被调度。在至少一个实施例中,每个调度器单元3504包括单个调度单元3506或附加调度单元3506。

[0543] 在至少一个实施例中,每个SM 3500在至少一个实施例中包括但不限于寄存器文件3508,该寄存器文件3508为SM 3500的功能单元提供了一组寄存器。在至少一个实施例中,寄存器文件3508在每个功能单元之间划分,从而为每个功能单元分配寄存器文件3508的专用部分。在至少一个实施例中,寄存器文件3508在由SM 3500执行的不同线程束之间划分,并且寄存器文件3508为连接到功能单元的数据路径的操作数提供临时存储。在至少一个实施例中,每个SM 3500包括但不限于多个L个处理核心3510,其中L是正整数。在至少一

个实施例中,SM 3500包括但不限于大量(例如128个或更多)不同的处理核心3510。在至少一个实施例中,每个处理核心3510包括但不限于全管线、单精度、双精度和/或混合精度处理单元,其包括但不限于浮点算术逻辑单元和整数算术逻辑单元。在至少一个实施例中,浮点算术逻辑单元实现用于浮点算术的IEEE 754-2008标准。在至少一个实施例中,处理核心3510包括但不限于64个单精度(32位)浮点核心、64个整数核心、32个双精度(64位)浮点核心和8个张量核心。

[0544] 根据至少一个实施例,张量核心配置为执行矩阵运算。在至少一个实施例中,一个或更多个张量核心包括在处理核心3510中。在至少一个实施例中,张量核心配置为执行深度学习矩阵算术,例如用于神经网络训练和推理的卷积运算。在至少一个实施例中,每个张量核心在4×4矩阵上操作并且执行矩阵乘法和累加运算D=A×B+C,其中A、B、C和D是4×4矩阵。

[0545] 在至少一个实施例中,矩阵乘法输入A和B是16位浮点矩阵,并且累加矩阵C和D是16位浮点或32位浮点矩阵。在至少一个实施例中,张量核心对16位浮点输入数据进行32位浮点累加运算。在至少一个实施例中,16位浮点乘法使用64个运算,并得到全精度乘积,然后使用32位浮点加法与其他中间乘积累加起来,以进行4x4x4矩阵乘法。在至少一个实施例中,张量核心用于执行由这些较小元件构成的更大的二维或更高维度的矩阵运算。在至少一个实施例中,API(诸如CUDA 9C++API)公开专门的矩阵加载、矩阵乘法和累加以及矩阵存储操作,以有效地使用来自CUDA-C++程序的张量核心。在至少一个实施例中,在CUDA级别,线程束级别接口假定跨越所有32个线程束线程的16×16大小的矩阵。

[0546] 在至少一个实施例中,每个SM 3500包括但不限于执行特殊功能(例如,属性评估、倒数平方根等)的M个SFU 3512。在至少一个实施例中,SFU 3512包括但不限于配置为遍历分层树数据结构的树遍历单元。在至少一个实施例中,SFU 3512包括但不限于配置为执行纹理映射过滤操作的纹理单元。在至少一个实施例中,纹理单元配置为从存储器中加载纹理映射(例如,纹理像素的2D阵列)和采样纹理映射,以产生采样的纹理值以供由SM 3500执行的着色器程序使用。在至少一个实施例中,将纹理映射存储在共享存储器/L1高速缓存3518中。在至少一个实施例中,根据至少一个实施例,纹理单元使用mip映射(mip-maps)(例如,细节级别不同的纹理映射)来实现纹理操作(诸如过滤操作)。在至少一个实施例中,每个SM 3500包括但不限于两个纹理单元。

[0547] 在至少一个实施例中,每个SM 3500包括但不限于实现共享存储器/L1高速缓存3518与寄存器文件3508之间的加载和存储操作的N个LSU 3514。在至少一个实施例中,互连网络3516将每个功能单元连接到寄存器文件3508,并且LSU 3514连接到寄存器文件3508和共享存储器/L1高速缓存3518。在至少一个实施例中,互连网络3516是交叉开关,其可以配置为将任何功能单元连接到寄存器文件3508中的任何寄存器,并且将LSU 3514连接到寄存器文件3508和共享存储器/L1高速缓存3518中的存储器位置。

[0548] 在至少一个实施例中,共享存储器/L1高速缓存3518是片上存储器的阵列,其在至少一个实施例中允许SM 3500与图元引擎之间以及SM 3500中的线程之间的数据存储和通信。在至少一个实施例中,共享存储器/L1高速缓存3518包括但不限于128KB的存储容量,并且位于从SM 3500到分区单元的路径中。在至少一个实施例中,共享存储器/L1高速缓存3518在至少一个实施例中用于高速缓存读取和写入。在至少一个实施例中,共享存储器/L1

高速缓存3518、L2高速缓存和存储器中的一个或更多个是后备存储。

[0549] 在至少一个实施例中，将数据高速缓存和共享存储器功能组合到单个存储器块中，为两种类型的存储器访问提供了改进的性能。在至少一个实施例中，容量由不使用共享存储器的程序使用或将其用作高速缓存，例如如果共享存储器配置为使用一半容量，并且纹理和加载/存储操作可以使用剩余容量。根据至少一个实施例，在共享存储器/L1高速缓存3518内的集成使共享存储器/L1高速缓存3518能够用作用于流传输数据的高吞吐量管线，同时提供对频繁重用的数据的高带宽和低延迟访问。在至少一个实施例中，当配置用于通用并行计算时，与图形处理相比，可以使用更简单的配置。在至少一个实施例中，绕过固定功能图形处理单元，从而创建了更加简单的编程模型。在至少一个实施例中，在通用并行计算配置中，工作分配单元直接将线程的块分配和分布给DPC。在至少一个实施例中，块中的线程执行通用程序，在计算中使用唯一的线程ID以确保每个线程生成唯一的结果，使用SM 3500执行程序并执行计算，使用共享存储器/L1高速缓存3518在线程之间进行通信，以及使用LSU 3514通过共享存储器/L1高速缓存3518和存储器分区单元来读写全局存储器。在至少一个实施例中，当被配置用于通用并行计算时，SM 3500向调度器单元3504写入可以用来在DPC上启动新工作的命令。

[0550] 在至少一个实施例中，PPU被包括在台式计算机、膝上型计算机、平板电脑、服务器、超级计算机、智能电话(例如，无线、手持设备)、个人数字助理(“PDA”)、数码相机、车辆、头戴式显示器、手持式电子设备等中或与之耦合。在至少一个实施例中，PPU被实现在单个半导体衬底上。在至少一个实施例中，PPU与一个或更多个其他设备(例如附加的PPU、存储器、精简指令集计算机(“RISC”)CPU，一个或更多个存储器管理单元(“MMU”)、数模转换器(“DAC”)等)一起被包括在片上系统(“SoC”)中。

[0551] 在至少一个实施例中，PPU可以被包括在包括一个或更多个存储设备的图形卡上。在至少一个实施例中，该图形卡可以配置为与台式计算机主板上的PCIe插槽相连接。在至少一个实施例中，该PPU可以是包括在主板的芯片组中的集成图形处理单元(“iGPU”)。

[0552] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。在至少一个实施例中，深度学习应用程序处理器用于训练机器学习模型(诸如神经网络)，以预测或推理提供给SM 3500的信息。在至少一个实施例中，SM 3500用于基于已由另一处理器或系统或由SM 3500训练过的机器学习模型(例如，神经网络)推理或预测信息。在至少一个实施例中，SM 3500可用于执行一个或更多个本文所述的神经网络用例。

[0553] 在至少一个实施例中，一个或更多个电路、处理器、计算系统或其他设备或技术被适配成用于参考所述图生成一个倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0554] 公开了实施例，其涉及用于高级计算的虚拟化计算平台，诸如医疗应用程序中的图像推理和图像处理。实施例可可以包括但不限于射线照相、磁共振成像(MRI)、核医学、超声、超声检查、弹性成像、光声成像、断层扫描、超声心动图、功能近红外光谱和磁粒子成像，

或其组合。在至少一个实施例中，本文所述的虚拟化计算平台和相关过程可以附加地或替代地用于但不限于法医科学分析、地下探测和成像(例如，石油勘探、考古学、古生物学等)、地形学、海洋学、地质学、骨学、气象学、智能区域或目标跟踪和监测、传感器数据处理(如雷达、声呐、激光雷达等)和/或基因组学和基因测序。

[0555] 参考图36，图36是根据至少一个实施例的用于生成和部署图像处理和推理管线的过程3600的示例数据流图。在至少一个实施例中，过程3600可以被部署用于成像设备、处理设备、基因组学设备、基因测序设备、放射设备和/或一个或更多个设施3602处的其他设备类型，该设施例如医疗设施、医院、医疗机构、诊所、研究或诊断实验室等。在至少一个实施例中，过程3600可以被部署为对测序数据进行基因组分析和推理。可以使用本文所述的系统和过程来执行基因组分析的示例，所述基因组分析的实例包括但不限于识别变体、突变检测和基因表达量化。

[0556] 在至少一个实施例中，过程3600可以在训练系统3604和/或部署系统3606内执行。在至少一个实施例中，训练系统3604可以用于执行机器学习模型(例如，神经网络、对象检测算法、计算机视觉算法等)的训练、部署和实现，以用于部署系统3606。在至少一个实施例中，部署系统3606可以被配置为在分布式计算环境中卸载处理和计算资源，以减少设施3602的基础设施需求。在至少一个实施例中，部署系统3606可以提供流水线平台用于选择、定制和实现虚拟仪器，以在设施3602处与成像设备(如MRI、CT扫描、X射线、超声波等)或测序设备一起使用。在至少一个实施例中，虚拟仪器可以包括用于针对由成像设备、测序设备、放射线设备和/或其他设备类型生成的成像数据执行一个或更多个处理操作的软件定义的应用程序。在至少一个实施例中，管线中的一个或更多个应用程序在应用程序执行期间，可以使用或调用部署系统3606的服务(例如，推理、可视化、计算、AI等)。

[0557] 在至少一个实施例中，在高级处理和推理管线中使用的一些应用程序可以使用机器学习模型或其他AI来执行一个或更多个处理步骤。在至少一个实施例中，可以使用在设施3602处生成(并存储在设施3602处的一个或更多个图片存档和通信系统(PACS)服务器上)的数据3608(例如成像数据)在设施3602处训练机器学习模型，可以使用来自另一个或更多个设施(例如，不同的医院、实验室、诊所等)的成像或测序数据3608来训练机器学习模型，或其组合。在至少一个实施例中，训练系统3604可以用于提供应用程序、服务和/或其他资源，以生成用于部署系统3606的工作的、可部署的机器学习模型。

[0558] 在至少一个实施例中，模型注册表3624可以由对象存储支持，该对象存储可以支持版本控制和对象元数据。在至少一个实施例中，可以从云平台内通过例如云存储(例如，图37的云3726)兼容的应用程序编程接口(API)来访问对象存储。在至少一个实施例中，模型注册表3624内的机器学习模型可以由与API交互的系统的开发者或合作伙伴上传、列出、修改或删除。在至少一个实施例中，API可以提供对方法的访问，所述方法允许具有适当凭证的用户将模型与应用程序相关联，使得模型可以作为应用程序的容器化实例化的执行的一部分来执行。

[0559] 在至少一个实施例中，训练管线3704(图37)可以包括以下情形：其中设施3602正在训练他们自己的机器学习模型，或者具有需要优化或更新的现有机器学习模型。在至少一个实施例中，可以接收由成像设备、测序设备和/或其他类型设备生成的成像数据3608。在至少一个实施例中，一旦接收到成像数据3608，AI辅助注释3610就可以用于帮助生成与

成像数据3608相对应的注释，以用作机器学习模型的地面实况数据。在至少一个实施例中，AI辅助注释3610可以包括一个或更多个机器学习模型(例如，卷积神经网络(CNN))，可以对该机器学习模型进行训练，以生成对应于某些类型的成像数据3608(例如，来自某些设备)的注释，和/或成像数据3608中某些类型的异常。在至少一个实施例中，然后AI辅助注释3610可以被直接使用，或者可以使用注释工具(例如，由研究人员、临床医生、医生、科学家等)进行调整或微调，以生成地面实况数据。在至少一个实施例中，在一些示例中，标记的临床数据3612(例如，由临床医生、医生、科学家、技术人员等提供的注释)可以用作训练机器学习模型的地面实况数据。在至少一个实施例中，AI辅助注释3610、标记的临床数据3612或其组合可以用作训练机器学习模型的地面实况数据。在至少一个实施例中，经训练的机器学习模型可以被称为输出模型3616，并且可以由部署系统3606使用，如本文所述。

[0560] 在至少一个实施例中，训练管线3704(图37)可以包括以下情形：其中设施3602需要机器学习模型，以用于执行用于部署系统3606中的一个或更多个应用程序的一个或更多个处理任务，但是设施3602当前可能没有这种机器学习模型(或者可能没有为此目的而优化的、高效的或有效的模型)。在至少一个实施例中，可以从模型注册表3624中选择现有的机器学习模型。在至少一个实施例中，模型注册表3624可以包括机器学习模型，其被训练为对成像数据执行各种不同的推理任务。在至少一个实施例中，可以在来自不同的设施(例如，位于远处的设施)而不是设施3602的成像数据上训练模型注册表3624中的机器学习模型。在至少一个实施例中，机器学习模型可能已经在来自一个位置、两个位置或任意数量的位置的成像数据上进行训练。在至少一个实施例中，当在来自特定位置的成像数据上进行训练时，可以在该位置处进行训练，或者至少以保护成像数据的机密性或限制成像数据从场外传递的方式进行训练(例如，遵守HIPAA法规、隐私法规等)。在至少一个实施例中，一旦在一个位置处训练了模型或部分地训练了模型，则可以将机器学习模型添加到模型注册表3624。在至少一个实施例中，然后可以在任意数量的其他设施处对机器学习模型进行重新训练或更新，经重新训练或更新的模型可以在模型注册表3624中使用。在至少一个实施例中，然后可以从模型注册表3624中选择机器学习模型(并称为输出模型3616)，并且可以在部署系统3606中，以执行用于部署系统的一个或更多个应用程序的一个或更多个处理任务。

[0561] 在至少一个实施例中，训练管线3704(图37)可用于包括设施3602的场景中，所述设施需要机器学习模型，以用于执行用于部署系统3606中的一个或更多个应用程序的一个或更多个处理任务，但是设施3602当前可能没有这样的机器学习模型(或者可能没有优化的、高效的或有效的模型)。在至少一个实施例中，由于用于训练机器学习模型的训练数据的种群差异、遗传变异、稳健性，训练数据异常的多样性，和/或训练数据的其他问题，从模型注册表3624中选择的机器学习模型可能不会针对在设施3602处生成的成像数据3608进行微调或优化。在至少一个实施例中，AI辅助注释3610可以用于帮助生成与成像数据3608相对应的注释，以用作训练或更新机器学习模型的地面实况数据。在至少一个实施例中，标记的临床数据3612(例如，由临床医生、医生、科学家等提供的注释)可以用作训练机器学习模型的地面实况数据。在至少一个实施例中，重新训练或更新机器学习模型可以称为模型训练3614。在至少一个实施例中，模型训练3614(例如AI辅助注释3610，标记的临床数据3612或其组合)可以用作重新训练或更新机器学习模型的地面实况数据。

[0562] 在至少一个实施例中，部署系统3606可以包括软件3618、服务3620、硬件3622和/或其他组件、特征和功能。在至少一个实施例中，部署系统3606可以包括软件“栈”，以使软件3618可以构建在服务3620的顶部上，并且可以使用服务3620来执行一些或全部处理任务，并且服务3620和软件3618可以构建在硬件3622的顶部上，并使用硬件3622来执行部署系统3606的处理、存储和/或其他计算任务。

[0563] 在至少一个实施例中，软件3618可以包括任意数量的不同容器，其中每个容器可以执行应用程序的实例化。在至少一个实施例中，每个应用程序可以在高级处理和推理管线中执行一个或更多个处理任务(例如，推理、对象检测、特征检测、分割、图像增强、校准等)。在至少一个实施例中，对于每种类型的成像设备(例如CT、MRI、X射线、超声、超声检查、超声心动图等)、测序设备、放射学设备、基因组学设备等，可能有任意数量的容器，其可以对由设备生成的成像数据3608(或其他数据类型，例如本文所述的数据类型)执行数据处理任务。在至少一个实施例中，除了接收和配置成像数据以供每个容器使用和/或在通过管线处理后由设施3602使用的容器以外，还可以基于对处理成像数据3608想要的或所需的不同容器的选择来定义高级处理和推理管线(例如，以将输出转换回可用的数据类型，诸如医学数字成像和通信(DICOM)数据、放射信息系统(RIS)数据、临床信息系统(CIS)数据、远程过程调用(RPC)数据、基本上符合表示状态传输(REST)接口的数据、基本上符合基于文件接口的数据、和/或原始数据，以便在设施3602进行存储和显示)。在至少一个实施例中，软件3618内的容器组合(例如，其构成管线)可以被称为虚拟仪器(如本文中更详细地描述的)，并且虚拟仪器可以利用服务3620和硬件3622来执行容器中实例化的应用程序的部分或全部处理任务。

[0564] 在至少一个实施例中，数据处理管线可以响应于推理请求(例如，来自部署系统3606的用户的请求，例如临床医生，医生，放射科医生等)接收DICOM、RIS、CIS、符合REST、RPC、原始、和/或其他格式的输入数据(例如，成像数据3608)。在至少一个实施例中，输入数据可以代表由一个或更多个成像设备、测序设备、放射设备、基因组设备和/或其他设备类型生成的一个或更多个图像、视频和/或其他数据表示。在至少一个实施例中，数据可以作为数据处理管线的一部分进行预处理，以准备数据用于由一个或更多个应用程序处理。在至少一个实施例中，可以在管线的一个或更多个推理任务或其他处理任务的输出上执行后处理，以准备下一个应用程序的输出数据和/或准备输出数据，以供用户传输和/或使用(例如作为对推理请求的响应)。在至少一个实施例中，推理任务可以由一个或更多个机器学习模型执行，例如经训练或部署的神经网络，所述模型可以包括训练系统3604的输出模型3616。

[0565] 在至少一个实施例中，数据处理管线的任务可以封装在容器中，每个容器表示能够引用机器学习模型的应用程序和虚拟化计算环境的离散、全功能实例化。在至少一个实施例中，容器或应用程序可被发布到容器注册表(本文更详细地描述)的私有(例如，有限访问)区域中，并且经训练或部署的模型可存储在模型注册表3624中，并与一个或更多个应用程序相关联。在至少一个实施例中，应用程序的图像(例如，容器图像)可在容器注册表中使用，并且一旦用户从容器注册表中选择图像以用于在管线中部署，则该图像可用于生成用于应用程序的实例化的容器，以供用户的系统使用。

[0566] 在至少一个实施例中，开发者(例如，软件开发者、临床医生、医生等)可以开发、发

布和存储应用程序(例如,作为容器),用于对所提供的数据执行图像处理和/或推理。在至少一个实施例中,可以使用与系统相关联的软件开发工具包(SDK)来执行开发、发布和/或存储(例如,以确保开发的应用程序和/或容器符合系统或与系统兼容)。在至少一个实施例中,所开发的应用程序可以使用SDK在本地测试(例如,在第一设施处,对来自第一设施的数据进行测试),所述SDK作为系统(例如图37中的系统3700)可以支持至少某些服务3620。在至少一个实施例中,由于DICOM对象可能包含一到数百个图像或其他数据类型,并且由于数据的变化,因此开发者可负责管理(例如,设置构造,用于将预处理构建到应用程序中等)传入的DICOM数据的提取和准备。在至少一个实施例中,一旦通过系统3700的验证(例如,为了准确性、安全性、患者隐私等),应用程序便会在容器注册表中可用,以供用户(例如,医院、诊所、实验室、医疗保健提供者等)选择和/或实现,以对用户的设施(例如第二设施)处的数据执行一个或更多个处理任务。

[0567] 在至少一个实施例中,开发者随后可以通过网络共享应用程序或容器,以供系统(例如,图37的系统3700)的用户访问和使用。在至少一个实施例中,可以将完成并经过验证的应用程序或容器存储在容器注册表中,并且可以将相关的机器学习模型存储在模型注册表3624中。在至少一个实施例中,请求实体(例如,医疗机构的用户)(其提供推理或图像处理请求)可以浏览容器注册表和/或模型注册表3624,以获得应用程序、容器、数据集、机器学习模型等,选择所需的元素组合以包含在数据处理管线中,并提交图像处理请求。在至少一个实施例中,请求可以包括执行请求所必需的输入数据(以及在一些示例中与患者相关的数据),和/或可以包括对在处理请求时要执行的应用程序和/或机器学习模型的选择。在至少一个实施例中,然后可以将请求传递到部署系统3606的一个或更多个组件(例如,云),以执行数据处理管线的处理。在至少一个实施例中,由部署系统3606进行的处理可以包括引用从容器注册表和/或模型注册表3624中选择的元素(例如,应用程序、容器、模型等)。在至少一个实施例中,一旦通过管线生成结果,结果可返回给用户以供参考(例如,用于在本地、本地工作站或终端上执行的查看应用程序套件中进行查看)。在至少一个实施例中,放射科医生可以从数据处理管线接收结果,所述数据处理管线包括任意数量的应用程序和/或容器,其中结果可以包括X射线、CT扫描、MRI等中的异常检测。

[0568] 在至少一个实施例中,为了帮助处理或执行管线中的应用程序或容器,可以利用服务3620。在至少一个实施例中,服务3620可以包括计算服务、人工智能(AI)服务、可视化服务和/或其他服务类型。在至少一个实施例中,服务3620可以提供软件3618中的一个或更多个应用程序所共有的功能,因此可以将功能抽象为可以被应用程序调用或利用的服务。在至少一个实施例中,由服务3620提供的功能可以动态且更有效地运行,同时还可以通过允许应用程序并行地处理数据(例如,使用图37中的并行计算平台3730)来很好地缩放。在至少一个实施例中,不是要求共享服务3620提供的相同功能的每个应用程序都必须具有服务3620的相应实例,而是可以在各种应用程序之间和之中共享服务3620。在至少一个实施例中,作为非限制性示例,服务可包括可用于执行检测或分割任务的推理服务器或引擎。在至少一个实施例中,可以包括模型训练服务,其可以提供机器学习模型训练和/或重新训练能力。在至少一个实施例中,可以还包括数据增强服务,其可以提供GPU加速的数据(例如,DICOM、RIS、CIS、符合REST、RPC、原始等)提取、调整大小、缩放和/或其他增强。在至少一个实施例中,可以使用可视化服务,其可以添加图像渲染效果(例如光线跟踪、光栅化、去噪、

锐化等),以向二维(2D)和/或三维(3D)模型添加真实感。在至少一个实施例中,可以包括虚拟仪器服务,其为虚拟仪器的管线内的其他应用程序提供波束赋形、分割、推理、成像和/或支持。

[0569] 在至少一个实施例中,在服务3620包括AI服务(例如,推理服务)的情况下,作为应用程序执行的一部分,可以通过调用(例如,作为API调用)推理服务(例如,推理服务器),以执行一个或更多个机器学习模型或其处理,来执行与用于异常检测(例如,肿瘤、生长异常、瘢痕形成等)的应用程序相关联的一个或更多个机器学习模型。在至少一个实施例中,在另一应用程序包括用于分割任务的一个或更多个机器学习模型的情况下,应用程序可以调用推理服务来执行机器学习模型,以用于执行与分割任务相关联的一个或更多个处理操作。在至少一个实施例中,实现高级处理和推理管线的软件3618,其包括分割应用程序和异常检测应用程序,可以被流水线化,因为每个应用程序可以调用相同的推理服务来执行一个或更多个推理任务。

[0570] 在至少一个实施例中,硬件3622可包括GPU、CPU、图形卡、AI/深度学习系统(例如,AI超级计算机,诸如NVIDIA的DGX超级计算机系统)、云平台或其组合。在至少一个实施例中,可以使用不同类型的硬件3622,以为部署系统3606中的软件3618和服务3620提供高效的、专门构建的支持。在至少一个实施例中,可以实现使用GPU处理来在AI/深度学习系统内、云系统中、和/或部署系统3606的其他处理组件中进行本地处理(例如,在设施3602处),以提高图像处理、图像重建、分割、MRI检查、中风或心脏病发作检测(例如实时地)、渲染的图像质量等的效率、准确性和效能。在至少一个实施例中,设施可包括成像设备、基因组设备、测序设备,和/或本地的其他设备类型,其可以利用GPU生成代表受试者解剖结构的成像数据。

[0571] 在至少一个实施例中,作为非限制性示例,关于深度学习、机器学习和/或高性能计算,可以针对GPU处理优化软件3618和/或服务3620。在至少一个实施例中,部署系统3606和/或训练系统3604的计算环境中的至少一些可以在具有GPU优化的软件(例如,NVIDIA DGX系统的硬件和软件组合)的数据中心、一个或更多个超级计算机或高性能计算机系统中执行。在至少一个实施例中,数据中心可以符合HIPAA的规定,使得关于患者数据的隐私性安全地处理成像数据和/或其他患者数据的接收、处理和传输。在至少一个实施例中,如本文所述,硬件3622可包括任意数量的GPU,所述GPU可被调用以并行执行数据处理。在至少一个实施例中,云平台还可包括用于深度学习任务的GPU优化执行、机器学习任务或其他计算任务的GPU处理。在至少一个实施例中,可以使用AI/深度学习超级计算机和/或GPU优化的软件(例如,如在NVIDIA的DGX系统上提供的)作为硬件抽象和缩放平台,来执行云平台(例如,NVIDIA的NGC)。在至少一个实施例中,云平台可以在多个GPU上集成应用程序容器集群系统或协调系统(例如,KUBERNETES),以实现无缝缩放和负载均衡。

[0572] 在至少一个实施例中,一个或更多个电路、处理器、计算系统或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0573] 图37是根据至少一个实施例的用于生成和部署成像部署管线的示例系统3700的系统图。在至少一个实施例中，系统3700可以用于实现图36的过程3600和/或其他过程，包括高级处理和推理管线。在至少一个实施例中，系统3700可以包括训练系统3604和部署系统3606。在至少一个实施例中，可以使用软件3618、服务3620和/或硬件3622，来实现训练系统3604和部署系统3606，如本文所述。

[0574] 在至少一个实施例中，系统3700(例如，训练系统3604和/或部署系统3606)可以在云计算环境中(例如，使用云3726)实现。在至少一个实施例中，系统3700可以在本地实现(关于医疗服务设施)，或者作为云计算资源和本地计算资源的组合来实现。在至少一个实施例中，在实现云计算的实施例中，患者数据可以与系统3700的一个或更多个组件分离，或者未由系统3700的一个或更多个组件处理，这将导致处理不符合HIPAA和/或其他数据处理和隐私法规或法律。在至少一个实施例中，可以通过制定安全措施或协议，将对云3726中的API的访问权限限制为授权用户。在至少一个实施例中，安全协议可以包括网络令牌，其可以由认证(例如，AuthN、AuthZ、Gluecon等)服务签名，并且可以携带适当的授权。在至少一个实施例中，虚拟仪器的API(本文中描述)或系统3700的其他实例可以被限制为已被审核或授权用于交互的一组公共IP。

[0575] 在至少一个实施例中，系统3700的各个组件可以使用多种不同网络类型中的任何一种在彼此之间进行通信，所述不同网络类型包括但不限于经由有线和/或无线通信协议的局域网(LAN)和/或广域网(WAN)。在至少一个实施例中，系统3700的设施和组件之间的通信(例如，用于发送推理请求、用于接收推理请求的结果等)可以通过一个或更多个数据总线、无线数据协议(Wi-Fi)、有线数据协议(例如以太网)等进行传送。

[0576] 在至少一个实施例中，类似于本文关于图36所描述的，训练系统3604可以执行训练管线3704。在至少一个实施例中，其中部署系统3606将在部署管线3710中使用一个或更多个机器学习模型，训练管线3704可用于训练或重新训练一个或更多个(例如，预训练的)模型，和/或实现一个或更多个预训练模型3706(例如，无需重新训练或更新)。在至少一个实施例中，作为训练管线3704的结果，可以生成输出模型3616。在至少一个实施例中，训练管线3704可以包括任意数量的处理步骤，诸如但不限于成像数据(或其他输入数据)的转换或适配(例如，使用DICOM适配器3702A将DICOM图像转换为适合于由各自的机器学习模型处理的另一种格式，例如Neuroimaging信息技术倡议(NIfTI)格式)、AI辅助注释3610、成像数据3608的标记或注释(用于生成标记的临床数据3612)、从模型注册表中选择模型、模型训练3614、训练、重新训练或更新模型，和/或其他处理步骤。在至少一个实施例中，对于由部署系统3606使用的不同的机器学习模型，可以使用不同的训练管线3704。在至少一个实施例中，类似于关于图36描述的第一示例的训练管线3704可用于第一机器学习模型，类似于关于图36描述的第二示例的训练管线3704可用于第二机器学习模型，类似于关于图36描述的第三示例的训练管线3704可用于第三机器学习模型。在至少一个实施例中，可以根据每个相应机器学习模型的要求来使用训练系统3604内任务的任何组合。在至少一个实施例中，一个或更多个机器学习模型可能已经被训练并准备好用于部署，因此训练系统3604可能不会对机器学习模型进行任何处理，并且一个或更多个机器学习模型可以由部署系统3606来实现。

[0577] 在至少一个实施例中，根据实现或实施例，输出模型3616和/或预训练模型3706可

包括任何类型的机器学习模型。在至少一个实施例中并且不限于此，系统3700使用的机器学习模型可以包括使用线性回归、逻辑回归、决策树、支持向量机(SVM)、朴素贝叶斯、k-最近邻(Knn)、k均值聚类、随机森林、降维算法、梯度提升算法、神经网络(例如，自动编码器、卷积、递归、感知器、长/短期记忆(LSTM)、Hopfield、Boltzmann、深层信念、反卷积、生成对抗、液体状态机等)，和/或其他类型的机器学习模型。

[0578] 在至少一个实施例中，训练管线3704可以包括AI辅助注释，如本文关于至少图40B更详细描述的。在至少一个实施例中，可以通过任何数量的技术来生成标记的临床数据3612(例如，传统注释)。在至少一个实施例中，在一些示例中可以在绘图程序(例如，注释程序)、计算机辅助设计(CAD)程序、标记程序、适用于生成地面实况的注释或标签的另一类型的应用程序中、和/或可以手绘，生成标签或其他注释。在至少一个实施例中，地面实况数据可以被合成产生(例如，从计算机模型或渲染生成)、真实产生(例如，从真实世界数据设计和生成)、机器自动产生(例如，使用特征分析和学习从数据中提取特征，然后生成标签)、人工注释(例如，标记器或注释专家，定义标签的位置)和/或其组合。在至少一个实施例中，对于成像数据3608(或机器学习模型使用的其他数据类型)的每个实例，可以存在由训练系统3604生成的相应的地面实况数据。在至少一个实施例中，可以作为部署管线3710的一部分执行AI辅助注释；补充或代替训练管线3704中包括的AI辅助注释。在至少一个实施例中，系统3700可以包括多层平台，所述多层平台可以包括诊断应用程序(或其他应用程序类型)的软件层(例如软件3618)，其可以执行一个或更多个医学成像和诊断功能。在至少一个实施例中，系统3700可以通信地耦合到(例如，经由加密链路)一个或更多个设施的PACS服务器网络。在至少一个实施例中，系统3700可被配置为从PACS服务器(例如，经由DICOM适配器3702或诸如RIS、CIS、符合REST、RPC、原始等的另一数据类型适配器)访问和引用数据(例如，DICOM数据、RIS数据、原始数据、CIS数据、符合REST的数据、RPC、原始数据等)，以执行操作，例如训练机器学习模型、部署机器学习模型、图像处理、推理和/或其他操作。

[0579] 在至少一个实施例中，软件层可以被实现为安全的、加密的和/或经认证的API，通过所述API可以从外部环境(例如，设施3602)援引(invoke)(例如，调用(call))应用程序或容器。在至少一个实施例中，应用程序随后可以调用或执行一个或更多个服务3620，以执行与各自的应用程序相关联的计算、AI或可视化任务，并且软件3618和/或服务3620可以利用硬件3622以有效和高效的方式执行处理任务。

[0580] 在至少一个实施例中，部署系统3606可以执行部署管线3710。在至少一个实施例中，部署管线3710可以包括任意数量的应用程序，所述应用程序可以是顺序的、非顺序的，或者以其他方式应用于成像数据(和/或其他数据类型)-包括AI辅助注释，所述成像数据由成像设备、测序设备、基因组学设备等生成，如上所述。在至少一个实施例中，如本文所述，用于个体设备的部署管线3710可以被称为用于设备的虚拟仪器(例如，虚拟超声仪器、虚拟CT扫描仪器、虚拟测序仪器等)。在至少一个实施例中，对于单个设备，可以存在不止一个部署管线3710，这取决于从设备生成的数据所期望的信息。在至少一个实施例中，在期望从MRI机器检测到异常的情况下，可以存在第一部署管线3710，并且在期望从MRI机器的输出进行图像增强的情况下，可以存在第二部署管线3710。

[0581] 在至少一个实施例中，可用于部署管线3710的应用程序可包括可用于对成像数据或来自设备的其他数据执行处理任务的任何应用程序。在至少一个实施例中，不同的应用

程序可负责图像增强、分割、重建、异常检测、对象检测、特征检测、治疗规划、剂量测定、波束规划(或其他辐射治疗程序)和/或其他分析、图像处理或推理任务。在至少一个实施例中,部署系统3606可以为每个应用程序定义构造,使得部署系统3606(例如,医疗设施、实验室、诊所等)的用户可以理解构造并将应用程序适配为在其各自的设施内实现。在至少一个实施例中,可以选择用于图像重建的应用程序,以包括在部署管线3710中,但是由成像设备生成的数据类型可以与在应用程序内使用的数据类型不同。在至少一个实施例中,可以在部署管线3710内使用DICOM适配器3702B(和/或DICOM读取器)或另一数据类型的适配器或读取器(例如,RIS、CIS、符合REST、RPC、原始等),以将数据转换为可由部署系统3606内的应用程序使用。在至少一个实施例中,对DICOM、RIS、CIS、符合REST、RPC、原始和/或其他数据类型库的访问可以被累积和预处理,包括解码、提取和/或对数据执行任何卷积、颜色校正、锐化、gamma和/或其他增强。在至少一个实施例中,DICOM、RIS、CIS、符合REST、RPC和/或原始数据可以是无序的,并且可以执行预传递以组织数据或对收集的数据排序。在至少一个实施例中,由于各种应用程序可以共享公共图像操作,因此在一些实施例中,可以使用数据增强库(例如,作为服务3620之一)来加速这些操作。在至少一个实施例中,为了避免依赖于CPU处理的传统处理方法的瓶颈,并行计算平台3730可用于这些处理任务的GPU加速。

[0582] 在至少一个实施例中,图像重建应用程序可包括处理任务,该处理任务包括使用机器学习模型。在至少一个实施例中,用户可能希望使用他们自己的机器学习模型,或者从模型注册表3624中选择机器学习模型。在至少一个实施例中,用户可以实现他们自己的机器学习模型或选择机器学习模型,以包含在执行处理任务的应用程序中。在至少一个实施例中,应用程序可以是可选择的和可定制的,并且通过定义应用程序的构造,针对特定用户的应用程序的部署和实现被呈现为更加无缝的用户体验。在至少一个实施例中,通过利用系统3700的其他特征(例如服务3620和硬件3622),部署管线3710可以更加用户友好,提供更容易的集成,并且产生更准确、高效和及时的结果。

[0583] 在至少一个实施例中,部署系统3606可以包括用户接口3714(例如,图形用户接口、Web接口等),所述用户接口可以被用于选择要包括在部署管线3710中的应用程序、布置应用程序、修改或改变应用程序或其参数或构造、在设置和/或部署期间使用部署管线3710以及与其交互,和/或以其他方式与部署系统3606交互。在至少一个实施例中,尽管没有关于训练系统3604示出,但是用户接口3714(或不同的用户接口)可用于选择在部署系统3606中使用的模型、用于选择用于在训练系统3604中训练或重新训练的模型,和/或用于以其他方式与训练系统3604交互。

[0584] 在至少一个实施例中,除了应用程序协调系统3728之外,还可以使用管线管理器3712来管理部署管线3710的应用程序或容器与服务3620和/或硬件3622之间的交互。在至少一个实施例中,管线管理器3712可以被配置为促进从应用程序到应用程序、从应用程序到服务3620,和/或从应用程序或服务到硬件3622的交互。在至少一个实施例中,尽管示出为包括在软件3618中,这并不旨在进行限制,并且在一些示例中(例如,如图38所示),管线管理器3712可以被包括在服务3620中。在至少一个实施例中,应用程序协调系统3728(例如,Kubernetes、DOCKER等)可以包括容器协调系统,其可以将应用程序分组到容器中,作为用于协调、管理、缩放和部署的逻辑单元。在至少一个实施例中,通过将来自部署管线3710的应用程序(例如,重建应用程序、分割应用程序等)与各个容器相关联,每个应用程序可以

在自包含的环境(例如,在内核级)中执行,以提高速度和效率。

[0585] 在至少一个实施例中,每个应用程序和/或容器(或其映像)可以被单独开发、修改和部署(例如,第一用户或开发者可以开发、修改和部署第一应用程序,第二用户或开发者可以开发、修改和部署与第一用户或开发者分开的第二应用程序),这可以允许专注并关注单个应用程序和/或容器的任务,而不受另一个应用程序或容器的任务的阻碍。在至少一个实施例中,管线管理器3712和应用程序协调系统3728可以辅助不同容器或应用程序之间的通信和协作。在至少一个实施例中,只要每个容器或应用程序的预期输入和/或输出是系统已知的(例如,基于应用程序或容器的构造),则应用程序协调系统3728和/或管线管理器3712可以促进每个应用程序或容器之间和之中的通信以及资源的共享。在至少一个实施例中,由于部署管线3710中的一个或更多个应用程序或容器可以共享相同的服务和资源,因此应用程序协调系统3728可以在各个应用程序或容器之间和之中进行协调、负载均衡,并确定服务或资源的共享。在至少一个实施例中,调度器可用于跟踪应用程序或容器的资源需求、这些资源的当前使用或计划使用,以及资源可用性。因此,在至少一个实施例中,考虑到系统的需求和可用性,调度器可以将资源分配给不同的应用程序,并在应用程序之间和之中分配资源。在一些示例中,调度器(和/或应用程序协调系统3728的其他组件)可以基于施加在系统上的约束(例如,用户约束)来确定资源可用性和分布,例如服务质量(QoS)、对数据输出的迫切需求(例如,以确定是执行实时处理还是延迟处理)等。

[0586] 在至少一个实施例中,由部署系统3606中的应用程序或容器利用并由其共享的服务3620,可以包括计算服务3716、AI服务3718、可视化服务3720和/或其他服务类型。在至少一个实施例中,应用程序可以调用(例如,执行)一个或更多个服务3620,以执行针对应用程序的处理操作。在至少一个实施例中,应用程序可以利用计算服务3716来执行超级计算或其他高性能计算(HPC)任务。在至少一个实施例中,可以利用一个或更多个计算服务3716来执行并行处理(例如,使用并行计算平台3730),以通过一个或更多个应用程序和/或单个应用程序的一个或更多个任务基本上同时地处理数据。在至少一个实施例中,并行计算平台3730(例如,NVIDIA的CUDA)可以在GPU(GPGPU)(例如,GPU 3722)上实现通用计算。在至少一个实施例中,并行计算平台3730的软件层可以提供对GPU的虚拟指令集和并行计算元素的访问,以执行计算内核。在至少一个实施例中,并行计算平台3730可以包括存储器,并且在一些实施例中,可以在多个容器之间和之中,和/或在单个容器内的不同处理任务之间和之中共享存储器。在至少一个实施例中,可以为多个容器和/或容器内的多个进程生成进程间通信(IPC)调用,以使用来自并行计算平台3730的共享存储器段的相同数据(例如,其中一应用程序或多个应用程序的多个不同阶段正在处理相同的信息)。在至少一个实施例中,不是复制数据并将数据移动到存储器中的不同位置(例如,读/写操作),而是可以将存储器相同位置中的相同数据用于任何数量的处理任务(例如,在同一时间、不同时间等)。在至少一个实施例中,由于作为处理的结果数据被用于生成新数据,因此数据的新位置的该信息可以在各个应用程序之间存储和共享。在至少一个实施例中,数据的位置以及经更新或修改的数据的位置可以是如何理解容器中的有效负载的定义的一部分。

[0587] 在至少一个实施例中,可以利用AI服务3718来执行推理服务,该推理服务用于执行与应用程序相关联的机器学习模型(例如,任务为执行应用程序的一个或更多个处理任务)。在至少一个实施例中,AI服务3718可以利用AI系统3724来执行机器学习模型(例如,诸

如CNN之类的神经网络)以用于分割、重建、对象检测、特征检测、分类和/或其他推理任务。在至少一个实施例中,部署管线3710的应用程序可以使用来自训练系统3604的一个或更多个输出模型3616和/或应用程序的其他模型,来对成像数据(例如,DICOM数据、RIS数据、CIS数据、符合REST的数据、RPC数据、原始数据等)执行推理。在至少一个实施例中,使用应用程序协调系统3728(例如,调度器)进行推理的两个或更多个示例可以是可用的。在至少一个实施例中,第一类别可以包括高优先级/低延时路径,其可以实现更高服务水平协议,例如用于在紧急情况下对紧急请求执行推理,或者在诊断过程中用于放射科医生。在至少一个实施例中,第二类别可以包括标准优先级路径,其可用于可能不紧急的请求或者可以在稍后的时间执行分析的情况。在至少一个实施例中,应用程序协调系统3728可以基于优先级路径来分配资源(例如,服务3620和/或硬件3622),以用于AI服务3718的不同推理任务。

[0588] 在至少一个实施例中,共享存储器可以被安装到系统3700中的AI服务3718。在至少一个实施例中,共享存储器可以操作为高速缓存(或其他存储设备类型),并且可以用于处理来自应用程序的推理请求。在至少一个实施例中,当提交推理请求时,部署系统3606的一组API实例可以接收请求,并且可以选择一个或更多个实例(例如,为了最佳拟合、为了负载均衡等)来处理请求。在至少一个实施例中,为了处理请求,可以将请求输入到数据库中,如果尚未在高速缓存中,则可以从模型注册表3624定位机器学习模型,验证步骤可以确保将适当的机器学习模型加载到高速缓存中(例如,共享存储),和/或可以将模型的副本保存到高速缓存中。在至少一个实施例中,如果应用程序尚未运行或没有足够的应用程序的实例,则可使用调度器(例如,管线管理器3712的调度器)来启动在请求中引用的应用程序。在至少一个实施例中,如果尚未启动推理服务器来执行模型,则可以启动推理服务器。在至少一个实施例中,每个模型可以启动任意数量的推理服务器。在至少一个实施例中,在将推理服务器聚类的拉(pull)模型中,每当负载均衡有利时,就可以将模型高速缓存。在至少一个实施例中,推理服务器可以静态加载到相应的分布式服务器中。

[0589] 在至少一个实施例中,可以使用在容器中运行的推理服务器来执行推理。在至少一个实施例中,推理服务器的实例可以与模型(并且可选地与模型的多个版本)相关联。在至少一个实施例中,如果在接收到对模型执行推理的请求时推理服务器的实例不存在,则可以加载新实例。在至少一个实施例中,当启动推理服务器时,可以将模型传递到推理服务器,使得可以使用相同的容器来服务不同的模型,只要推理服务器作为不同的实例运行即可。

[0590] 在至少一个实施例中,在应用程序执行期间,可以接收对给定应用程序的推理请求,并且可以加载(如果尚未加载的话)容器(例如,托管推理服务器的实例),以及可以调用启动程序。在至少一个实施例中,容器中的预处理逻辑可以(例如,使用CPU和/或GPU)对传入的数据进行加载、解码和/或执行任何附加的预处理。在至少一个实施例中,一旦数据准备好进行推理,容器就可以根据需要对数据进行推理。在至少一个实施例中,这可以包括对一个图像(例如,手部X射线)的单个推理调用,或可要求对数百个图像(例如,胸部CT)进行推理。在至少一个实施例中,应用程序可在完成之前总结结果,其可以包括但不限于单个置信度得分、像素级分割、体素级分割、生成可视化或生成文本以总结结果。在至少一个实施例中,可以为不同的模型或应用程序分配不同的优先级。例如,一些模型可具有实时(TAT小于1分钟)优先级,而其他模型可具有较低的优先级(例如,TAT小于10分钟)。在至少一个实

施例中,模型执行时间可以从请求机构或实体进行测量,并且可以包括合作网络遍历时间以及推理服务的执行时间。

[0591] 在至少一个实施例中,请求在服务3620和推理应用程序之间的传送可以隐藏在软件开发工具包(SDK)后面,并且可以通过队列提供鲁棒的传输。在至少一个实施例中,将通过API将请求放置在队列中,以用于个体应用程序/租户ID组合,并且SDK将从队列中拉取请求并将请求提供给应用程序。在至少一个实施例中,在SDK将从中拾取队列的环境中,可以提供队列的名称。在至少一个实施例中,通过队列的异步通信可能有用,因为它可以允许应用程序的任何实例在其可用时拾取工作。在至少一个实施例中,可以通过队列将结果传送回去,以确保没有数据丢失。在至少一个实施例中,队列还可以提供对工作进行分割的能力,因为最高优先级的工作可以进入与应用程序的大多数实例连接的队列,而最低优先级的工作可以进入与单个实例连接的队列,所述实例按照接收到的顺序处理任务。在至少一个实施例中,应用程序可以在GPU加速的实例上运行,所述实例在云3726中生成,并且推理服务可以在GPU上执行推理。

[0592] 在至少一个实施例中,可以利用可视化服务3720来生成用于查看应用程序和/或部署管线3710输出的可视化。在至少一个实施例中,可视化服务3720可以利用GPU 3722来生成可视化。在至少一个实施例中,可视化服务3720可以实现诸如光线追踪之类的渲染效果,以生成更高质量的可视化。在至少一个实施例中,可视化可以包括但不限于2D图像渲染、3D体渲染、3D体重建、2D层析切片、虚拟现实显示、增强现实显示等。在至少一个实施例中,可以使用虚拟化环境来生成虚拟交互显示或环境(例如,虚拟环境),以供系统用户(例如,医生、护士、放射科医生等)进行交互。在至少一个实施例中,可视化服务3720可以包括内部可视化器、电影和/或其他渲染或图像处理能力或功能(例如,光线追踪、光栅化、内部光学器件等)。

[0593] 在至少一个实施例中,硬件3622可以包括GPU 3722、AI系统3724、云3726和/或用于执行训练系统3604和/或部署系统3606的任何其他硬件。在至少一个实施例中,GPU 3722(例如,NVIDIA的TESLA和/或QUADRO GPU)可包括可用于执行计算服务3716、AI服务3718、可视化服务3720、其他服务和/或软件3618的任何特征或功能的处理任务的任意数量的GPU。例如,对于AI服务3718,GPU 3722可用于对成像数据(或机器学习模型使用的其他数据类型)执行预处理,对机器学习模型的输出执行后处理和/或执行推理(例如以执行机器学习模型)。在至少一个实施例中,云3726、AI系统3724和/或系统3700的其他组件可以使用GPU 3722。在至少一个实施例中,云3726可以包括用于深度学习任务的GPU优化的平台。在至少一个实施例中,AI系统3724可以使用GPU,并且可以使用一个或更多个AI系统3724来执行云3726(或者任务为深度学习或推理的至少部分)。同样,尽管硬件3622被示出为离散组件,但这并不意图是限制,并且硬件3622的任何组件可以与硬件3622的任何其他组件组合,或由硬件3622的任何其他组件利用。

[0594] 在至少一个实施例中,AI系统3724可包括专门构建的计算系统(例如,超级计算机或HPC),该计算系统配置用于推理、深度学习、机器学习和/或其他人工智能任务。在至少一个实施例中,除了CPU、RAM、存储器和/或其他组件、特征或功能之外,AI系统3724(例如,NVIDIA的DGX)还可以包括可以使用多个GPU 3722来执行分GPU优化的软件(例如,软件栈)。在至少一个实施例中,可以在云3726中(例如,在数据中心中)实现一个或更多个AI系统

3724,以执行系统3700的一些或全部基于AI的处理任务。

[0595] 在至少一个实施例中,云3726可以包括GPU加速的基础设施(例如,NVIDIA的NGC),其可以提供用于执行系统3700的处理任务的GPU优化的平台。在至少一个实施例中,云3726可以包括AI系统3724,其用于执行系统3700的一个或更多个基于AI的任务(例如,作为硬件抽象和缩放平台)。在至少一个实施例中,云3726可以与利用多个GPU的应用程序协调系统3728集成,以实现应用程序和服务3620之间和之中的无缝缩放和负载均衡。在至少一个实施例中,如本文所述,云3726可以负责执行系统3700的至少一些服务3620,包括计算服务3716、AI服务3718和/或可视化服务3720。在至少一个实施例中,云3726可以执行大小批的推理(例如,执行NVIDIA的TENSOR RT),提供加速的并行计算API和平台3730(例如,NVIDIA的CUDA),执行应用程序协调系统3728(例如,KUBERNETES),提供图形渲染API和平台(例如,用于光线跟踪,2D图形、3D图形和/或其他渲染技术以产生更高质量的电影效果),和/或可以为系统3700提供其他功能。

[0596] 在至少一个实施例中,为了保护患者的机密性(例如,在非现场使用患者数据或记录的情况下),云3726可以包括注册表-例如深度学习容器注册表。在至少一个实施例中,注册表可以存储用于实例化应用程序的容器,所述应用程序可以对患者数据执行预处理、后处理或其他处理任务。在至少一个实施例中,云3726可接收数据,所述数据包括患者数据以及容器中传感器数据,仅对那些容器中的传感器数据执行所请求的处理,然后将结果输出和/或可视化转发给适当的各方和/或设备(例如用于可视化或诊断的本地医疗设备),而无需提取、存储或以其他方式访问患者数据。在至少一个实施例中,根据HIPAA和/或其他数据规定来保留患者数据的机密性。

[0597] 在至少一个实施例中,参照所述图,一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中,这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中,倒转神经网络用于生成训练数据的分布,其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中,使用所述训练数据的分布来训练学生网络。在至少一个实施例中,所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0598] 图38包括根据至少一个实施例的用于处理成像数据的部署管线3710A的示例说明。在至少一个实施例中,系统3700(特别是部署系统3606),可以用于将部署管线3710A定制、更新和/或集成到一个或更多个生产环境中。在至少一个实施例中,图38的部署管线3710A包括部署管线3710A的非限制性示例,其可以由设施(例如,在医院、诊所、实验室、研究环境等)处的特定用户(或用户团队)自定义。在至少一个实施例中,为了定义用于CT扫描仪3802的部署管线3710A,用户可以例如从容器注册表中选择一个或更多个应用程序,所述应用程序执行关于由CT扫描仪3802生成的成像数据的特定功能或任务。在至少一个实施例中,应用程序可以作为容器而应用到部署管线3710A,所述容器可以利用系统3700的服务3620和/或硬件3622。此外,部署管线3710A可以包括附加处理任务或应用程序,所述附加处理任务或应用程序可以被实现以准备供应用程序使用的数据(例如DICOM适配器3702B和DICOM读取器3806可在部署管线3710A中使用,以准备供CT重建3808、器官分割3810等使用的数据)。在至少一个实施例中,可以定制或选择部署管线3710A,以用于一致的部署、一次使用,或另一频率或间隔使用。在至少一个实施例中,用户可能希望在特定间隔内具有针对

几个受试者的CT重建3808和器官分割3810，并且因此可以在该时间段内部署管线3710A。在至少一个实施例中，用户可以针对来自系统3700的每个请求选择用户想要针对该请求对该数据执行处理的应用程序。在至少一个实施例中，可以以任何间隔来调整部署管线3710A，并且由于系统3700内的容器结构的适应性和可缩放性，这可以是无缝的过程。

[0599] 在至少一个实施例中，图38的部署管线3710A可以包括生成患者或受试者的成像数据的CT扫描仪3802。在至少一个实施例中，来自CT扫描仪3802的成像数据可以存储在与容纳CT扫描仪3802的设施相关联的PACS服务器3804上。在至少一个实施例中，PACS服务器3804可以包括软件和/或硬件组件，所述软件和/或硬件组件可以与设施处的成像模态(例如，CT扫描仪3802)直接接口。在至少一个实施例中，DICOM适配器3702B可以允许使用DICOM协议发送和接收DICOM对象。在至少一个实施例中，DICOM适配器3702B可以帮助准备或配置来自PACS服务器3804的DICOM数据，以供部署管线3710A使用。在至少一个实施例中，一旦通过DICOM适配器3702B处理了DICOM数据，管线管理器3712就可以将数据路由到部署管线3710A。在至少一个实施例中，DICOM读取器3806可以从DICOM数据(例如，原始正弦图数据，如可视化3816A中所示)提取图像文件和任何相关联的元数据。在至少一个实施例中，所提取的工作文件可以被存储在高速缓存中，以被部署管线3710A中的其他应用程序更快地处理。在至少一个实施例中，一旦DICOM读取器3806完成了提取和/或存储数据，就可以将完成信号传送到管线管理器3712。在至少一个实施例中，管线管理器3712随后可以发起或调用部署管线3710A中的一个或更多个其他应用程序或容器。

[0600] 在至少一个实施例中，一旦数据(例如，原始正弦图数据)可用于由CT重建3808应用程序处理，就可以执行CT重建3808应用程序和/或容器。在至少一个实施例中，CT重建3808可以从高速缓存中读取原始正弦图数据，从原始正弦图数据中重建图像文件(例如，如可视化3816B所示)，并且将所得图像文件存储在高速缓存中。在至少一个实施例中，在重建完成时，可以向管线管理器3712发送重建任务完成的信号。在至少一个实施例中，一旦重建完成，并且重建的图像文件可以被存储在高速缓存(或其他存储设备)中，则器官分割3810应用程序和/或容器可以由管线管理器3712触发。在至少一个实施例中，器官分割3810应用程序和/或容器可以从高速缓存中读取图像文件，将图像文件归一化或转换为适合推理的格式(例如，将图像文件转换为机器学习模型的输入分辨率)，并对归一化的图像运行推理。在至少一个实施例中，为了对归一化的图像运行推理，器官分割3810应用程序和/或容器可以依赖服务3620，管线管理器3712和/或应用程序协调系统3728可以通过器官分割3810应用程序和/或容器来促进服务3620的使用。在至少一个实施例中，例如，器官分割3810应用程序和/或容器可以利用AI服务3718对归一化的图像执行推理，并且AI服务3718可以利用硬件3622(例如AI系统3724)来执行AI服务3718。在至少一个实施例中，推理结果可以是掩码文件(例如，如可视化3816C所示)，所述掩码文件可以存储在高速缓存(或其他存储设备)中。

[0601] 在至少一个实施例中，一旦处理DICOM数据和/或从DICOM数据提取的数据的应用程序已经完成处理，就可以为管线管理器3712生成信号。在至少一个实施例中，管线管理器3712随后可执行DICOM写入器3812，以从高速缓存(或其他存储设备)读取结果，将结果打包成DICOM格式(例如，作为DICOM输出3814)，以供设施处生成请求的用户使用。在至少一个实施例中，DICOM输出3814随后可以被发送到DICOM适配器3702B，以准备DICOM输出3814，以存

储在PACS服务器3804上(例如,以供设施处的DICOM查看器查看)。在至少一个实施例中,响应于对重建和分割的请求,可视化3816B和3816C可被生成并可供用户用于诊断、研究和/或其他目的。

[0602] 尽管在部署管线3710A中图示为连续应用程序,但在至少一个实施例中,可以并行处理CT重建3808和器官分割3810应用程序。在至少一个实施例中,其中应用程序彼此不具有依赖性,并且数据可用于每个应用程序(例如,在DICOM读取器3806提取数据之后),应用程序可在同一时间、基本上在同一时间或有一些重叠地执行。在至少一个实施例中,在两个或更多个应用程序需要类似服务3620的情况下,系统3700的调度器可用于负载均衡以及在各个应用程序之间和之中分配计算或处理资源。在至少一个实施例中,在一些实施例中,并行计算平台3730可用于对应用程序执行并行处理,以减少部署管线3710A的运行时间以提供实时结果。

[0603] 在至少一个实施例中并参考图39A-39B,部署系统3606可以实现为一个或更多个虚拟仪器,以使用成像设备(例如,CT扫描仪、X射线机、MRI机等)、测序设备、基因组学设备和/或其他设备类型来执行不同的功能,例如图像处理、分割、增强、AI、可视化和推理。在至少一个实施例中,系统3700可以允许创建和提供虚拟仪器,所述虚拟仪器可以包括软件定义的部署管线3710,该软件定义的部署管线3710可以接收由设备生成的原始/未经处理的输入数据并输出经处理/重建的数据。在至少一个实施例中,表示虚拟仪器的部署管线3710(例如3710A和3710B)可以在管线中实现智能(诸如通过利用机器学习模型),以向系统提供容器化的推理支持。在至少一个实施例中,虚拟仪器可以执行任何数量的容器,每个容器包括应用程序的实例。在至少一个实施例中,例如在想要实时处理的情况下,表示虚拟仪器的部署管线3710可以是静态的(例如,可以设置容器和/或应用程序),而在其他示例中,可以从应用程序或资源池中(例如,在容器注册表中)选择用于虚拟仪器的容器和/或应用程序(例如,基于每个请求)。

[0604] 在至少一个实施例中,系统3700可以作为一个或更多个虚拟仪器在设施处的例如计算系统中被本地实例化或执行,该计算机系统部署在放射机器、成像设备和/或设施处的另一设备类型旁边或与之通信。然而,在至少一个实施例中,可以在设备本身的计算系统(例如,与成像设备集成在一起的计算系统)中,在本地数据中心(例如,本地部署的数据中心)中和/或云环境中(例如,在云3726中)实例化或执行本地安装。在至少一个实施例中,在一些示例中,可以由超级计算机或其他HPC系统实例化作为虚拟仪器操作的部署系统3606。在至少一个实施例中,本地安装可以允许用于实时处理的高带宽用途(例如,通过更高吞吐量的本地通信接口,例如以太网上的RF)。在至少一个实施例中,在虚拟仪器支持超声设备或其他成像模态的情况下,实时或近实时处理可能特别有用,在该超声设备或其他成像模态中,期望或需要即时可视化以进行准确的诊断和分析。在至少一个实施例中,当本地需求超过本地容量或能力时,云计算架构可能够动态地突发到云计算服务提供商或其他计算集群。在至少一个实施例中,如在本文中关于训练系统3604所描述的,云架构在被实现时可被调整用于训练神经网络或其他机器学习模型。在至少一个实施例中,在训练管线就位的情况下,机器学习模型可以在处理来自其支持的设备的附加数据时被不断地学习和改进。在至少一个实施例中,可以使用附加数据、新数据、现有机器学习模型和/或新的或更新的机器学习模型来持续改进虚拟仪器。

[0605] 在至少一个实施例中，计算系统可以包括本文所述的硬件3622中的部分或全部，并且硬件3622可以以多种方式中的任一种来分布，包括：在设备内，作为耦合到设备并位于其附近的计算设备的一部分，在设施处的本地数据中心中和/或在云3726中。在至少一个实施例中，由于部署系统3606和相关联的应用程序或容器是在软件中创建的(例如，作为应用程序的离散容器化实例化)，因此可以根据需要修改或定制虚拟仪器的行为、操作和配置以及由虚拟仪器生成的输出，无需更改或改变虚拟仪器支持的设备的原始输出。

[0606] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0607] 图39A包括根据至少一个实施例的支持超声设备的虚拟仪器的示例数据流图。在至少一个实施例中，部署管线3710B可以利用系统3700的一个或更多个服务3620。在至少一个实施例中，部署管线3710B和服务3620可以利用本地或云3726中的系统的硬件3622。在一个实施例中，尽管未示出，但是可以通过管线管理器3712、应用程序协调系统3728和/或并行计算平台3730来促进过程3900。

[0608] 在至少一个实施例中，过程3900可以包括从超声设备3902接收成像数据。在至少一个实施例中，成像数据可以DICOM格式(或其他格式，例如RIS、CIS、符合REST、RPC、原始等)存储在PACS服务器上，也可以由系统3700接收以通过部署管线3710进行处理，所述部署管线3710被选择或定制为超声设备3902的虚拟仪器(例如，虚拟超声)。在至少一个实施例中，可以直接从成像设备(例如，超声设备3902)接收成像数据，并由虚拟仪器对其进行处理。在至少一个实施例中，通信地耦合在成像设备和虚拟仪器之间的换能器或其他信号转换器可以将由成像设备生成的信号数据转换成可以由虚拟仪器处理的图像数据。在至少一个实施例中，原始数据和/或图像数据可应用于DICOM读取器3806，以提取数据，以供部署管线3710B的应用程序或容器使用。在至少一个实施例中，DICOM读取器3806可以利用数据扩充库3914(例如，NVIDIA的DALI)作为服务3620(例如，作为计算服务3716之一)，用于提取、调整大小、重新缩放和/或以其他方式准备数据，以供应用程序或容器使用。

[0609] 在至少一个实施例中，一旦准备好数据，就可以执行重建3906应用程序和/或容器，以将来自超声设备3902的数据重建为图像文件。在至少一个实施例中，在重建3906之后或与重建3906同时，可以执行检测3908应用程序和/或容器，以用于异常检测、对象检测、特征检测和/或与数据有关的其他检测任务。在至少一个实施例中，可以在检测3908期间使用在重建3906期间生成的图像文件以识别异常、对象、特征等。在至少一个实施例中，检测3908应用程序可以利用推理引擎3916(例如，作为AI服务3718之一)，来对数据执行推理以生成检测。在至少一个实施例中，检测3908应用程序可以执行或调用一个或更多个机器学习模型(例如，来自训练系统3604)。

[0610] 在至少一个实施例中，一旦重建3906和/或检测3908完成，则从这些应用程序和/或容器输出的数据可用于生成在工作站或显示终端上显示的可视化3910，例如可视化3912(例如，灰度输出)。在至少一个实施例中，可视化可以允许技术人员或其他用户将关于超声

设备3902的部署管线3710B的结果可视化。在至少一个实施例中，可以通过利用系统3700的渲染组件3918(例如，可视化服务3720之一)来执行可视化3910。在至少一个实施例中，渲染组件3918可以执行2D、OpenGL或光线追踪服务以生成可视化3912。

[0611] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0612] 图39B包括根据至少一个实施例的支持CT扫描仪的虚拟仪器的示例数据流图。在至少一个实施例中，部署管线3710C可以利用系统3700的一个或更多个服务3620。在至少一个实施例中，部署管线3710C和服务3620可在本地或在云3726中利用系统的硬件3622。在至少一个实施例中，尽管未示出，但是管线管理器3712、应用程序协调系统3728和/或并行计算平台3730可以促进过程3920。

[0613] 在至少一个实施例中，过程3920可以包括CT扫描仪3922生成可以由DICOM读取器3806接收的原始数据(例如，在处理等之后，直接经由PACS服务器3804接收)。在至少一个实施例中，虚拟CT(由部署管线3710C实例化)可以包括第一实时管线，用于监视患者(例如，患者运动检测AI 3926)和/或用于调整或优化CT扫描仪3922的曝光(例如，使用曝光控制AI 3924)。在至少一个实施例中，一个或更多个应用程序(例如，3924和3926)可以利用服务3620，例如AI服务3718。在至少一个实施例中，曝光控制AI 3924应用程序(或容器)和/或患者运动检测AI 3926应用程序(或容器)的输出，可以用作对CT扫描仪3922和/或技术人员的反馈，以调整曝光(或CT扫描仪3922的其他设置)和/或通知患者减少运动。

[0614] 在至少一个实施例中，部署管线3710C可以包括用于分析由CT扫描仪3922生成的数据的非实时管线。在至少一个实施例中，第二管线可以包括CT重建3808应用程序和/或容器、粗略检测AI 3928应用程序和/或容器、精细检测AI 3932应用程序和/或容器(例如，其中通过粗略检测AI 3928检测某些结果)、可视化3930应用程序和/或容器、以及DICOM写入器3812(和/或其他数据类型编写器，例如RIS、CIS、符合REST、RPC、原始文件等)应用程序和/或容器。在至少一个实施例中，由CT扫描仪3922生成的原始数据可以传递通过部署管线3710C的管线(被实例化为虚拟CT仪器)以生成结果。在至少一个实施例中，来自DICOM写入器3812的结果可被发送以供显示，和/或可被存储在PACS服务器3804上以供技术人员、从业者或其他用户稍后检索、分析或显示。

[0615] 在至少一个实施例中，参照所述图，一个或更多个电路、处理器、计算系统或其他设备或技术被适配用于生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0616] 图40A示出了根据至少一个实施例的用于训练、重新训练或更新机器学习模型的过程4000的数据流图。在至少一个实施例中，可以使用作为非限制性示例的图37的系统

3700来执行过程4000。在至少一个实施例中，过程4000可以利用系统3700的服务3620和/或硬件3622，如本文所述。

[0617] 在至少一个实施例中，由过程4000生成的精炼模型4012可以由部署系统3606针对部署管线3710中的一个或更多个容器化的应用程序执行。

[0618] 在至少一个实施例中，模型训练3614可包括使用新的训练数据(例如，新的输入数据(诸如客户数据集4006)，和/或与输入数据相关联的新的地面实况数据)重新训练或更新初始模型4004(例如，预训练模型)。在至少一个实施例中，为了重新训练或更新初始模型4004，可以重置或删除初始模型4004的输出或损失层，和/或用更新的或新的输出或损失层代替。在至少一个实施例中，初始模型4004可以具有从先前的训练中保留下来的先前精细调整的参数(例如，权重和/或偏差)，因此训练或重新训练3614可能不需要花费与从头开始训练模型一样长的时间或不需要那么多的处理。在至少一个实施例中，在模型训练3614期间，通过重置或替换初始模型4004的输出或损失层，在新的客户数据集4006(例如图36的图像数据3608)上生成预测时，可以基于与输出或损失层的精度相关联的损失计算，更新和重新调整新数据集的参数。

[0619] 在至少一个实施例中，可以将经预训练的模型3706存储在数据存储或注册表中(例如，图36的模型注册表3624)。在至少一个实施例中，经预训练的模型3706可能已经至少部分地在除了执行过程4000的设施之外的一个或更多个设施处被训练。在至少一个实施例中，为了保护患者、受试者或不同设施的客户的隐私和权利，经预训练的模型3706可能已经使用本地生成的客户或患者数据在本地进行了训练。在至少一个实施例中，可以使用云3726和/或其他硬件3622来训练经预训练的模型3706，但是机密的、受隐私保护的患者数据可以不被传送到云3726的任何组件(或其他非本地硬件)、由其使用或由其访问。在至少一个实施例中，如果使用来自不止一个设施的患者数据来训练经预训练的模型3706，则在来自另一设施的患者或客户数据上进行训练之前，经预训练的模型3706可能已经针对每个设施进行了单独训练。在至少一个实施例中，例如在客户或患者数据已发布隐私问题(例如，通过放弃，用于实验用途等)，或者其中客户或患者数据包括在公共数据集中的情况下，来自任意数量的设施的客户或患者数据可以用于在本地和/或外部训练经预训练的模型3706，例如在数据中心中或其他云计算基础设施中。

[0620] 在至少一个实施例中，在选择应用程序以在部署管线3710中使用时，用户还可以选择用于特定应用程序的机器学习模型。在至少一个实施例中，用户可能没有模型以使用，因此用户可以选择要与应用程序一起使用的经预训练的模型3706。在至少一个实施例中，经预训练的模型3706可能没有被优化用于在用户设施的客户数据集4006上生成准确的结果(例如，基于患者多样性、人口统计、所使用的医学成像设备的类型等)。在至少一个实施例中，在将经预训练的模型3706部署到部署管线3710中以与一个或更多个应用程序一起使用之前，经预训练的模型3706可以被更新、重新训练和/或微调，以用于在各个设施处使用。

[0621] 在至少一个实施例中，用户可以选择要更新、重新训练和/或微调的经预训练的模型3706，并且经预训练的模型3706可以称为过程4000中训练系统3604的初始模型4004。在至少一个实施例中，客户数据集4006(例如，成像数据、基因组数据、测序数据或由设施处的设备生成的其他数据类型)可用于对初始模型4004执行模型训练3614(其可包括但不限于传递学习)，以生成精炼模型4012。在至少一个实施例中，可以由训练系统3604生成与客户

数据集4006相对应的地面实况数据。在至少一个实施例中，可以至少部分地由临床医生、科学家、医生、从业者在设施处生成地面实况数据(例如，如图36中的标记的临床数据3612)。

[0622] 在至少一个实施例中，在一些示例中可以使用AI辅助注释3610来生成地面实况数据。在至少一个实施例中，AI辅助注释3610(例如，使用AI辅助注释SDK实现)可以利用机器学习模型(例如，神经网络)来生成用于客户数据集的建议或预测的地面实况数据。在至少一个实施例中，用户4010可以在计算设备4008上的用户界面(图形用户界面(GUI))内使用注释工具。

[0623] 在至少一个实施例中，用户4010可以经由计算设备4008与GUI交互，以编辑或微调注释或自动注释。在至少一个实施例中，多边形编辑特征可以用于将多边形的顶点移动到更精确或微调的位置。

[0624] 在至少一个实施例中，一旦客户数据集4006具有相关联的地面实况数据，则地面实况数据(例如，来自AI辅助注释、手动标记等)可以在模型训练3614期间用于生成精炼模型4012。在至少一个实施例中，客户数据集4006可以被应用到初始模型4004任意次数，并且地面实况数据可以用于更新初始模型4004的参数，直到对于精炼模型4012达到可接受的精度水平为止。在至少一个实施例中，一旦生成精炼模型4012，就可以在设施处的一个或更多个部署管线3710内部署精炼模型4012，以用于执行关于医学成像数据的一个或更多个处理任务。

[0625] 在至少一个实施例中，可以将精炼模型4012上传到模型注册表3624中的经预训练的模型3706，以由另一个设施选择。在至少一个实施例中，他的过程可以在任意数量的设施处完成，使得可以在新数据集上对精炼模型4012进一步精炼任意次数，以生成更通用的模型。

[0626] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0627] 图40B是根据至少一个实施例的用于利用经预训练的注释模型来增强注释工具的客户端-服务器架构4032的示例图示。在至少一个实施例中，可以基于客户端-服务器架构4032来实例化AI辅助注释工具4036。在至少一个实施例中，成像应用程序中的注释工具4036可以帮助放射线医生，例如识别器官和异常。在至少一个实施例中，成像应用程序可以包括软件工具，作为非限制性示例，所述软件工具帮助用户4010识别原始图像4034中(例如，在3D MRI或CT扫描中)的特定感兴趣器官上的几个极值点，并接收特定器官的所有2D切片的自动注释结果。在至少一个实施例中，结果可以作为训练数据4038存储在数据存储中，并且用作(例如但不限于)用于训练的地面实况数据。在至少一个实施例中，当计算设备4008发送用于AI辅助注释3610的极值点时，例如，深度学习模型可以接收该数据作为输入并返回分割器官或异常的推理结果。在至少一个实施例中，预实例化的注释工具(例如图40B中的AI辅助注释工具4036B)可以通过对服务器(诸如注释助手服务器4040)进行API调用(例如API调用4044)来增强，注释助手服务器4040可包括存储在例如注释模型注册表中

的一组经预训练的模型4042。在至少一个实施例中，注释模型注册表可以存储经预训练的模型4042(例如，机器学习模型，诸如深度学习模型)，其被预训练以对特定器官或异常执行AI辅助注释。在至少一个实施例中，可以通过使用训练管线4004来进一步更新这些模型。在至少一个实施例中，随着添加新的标记的临床数据3612，可以随时间改进预安装的注释工具。

[0628] 推理和/或训练逻辑715用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图7A和/或图7B提供关于推理和/或训练逻辑715的细节。

[0629] 在至少一个实施例中，一个或更多个电路、处理器、计算系统、或其他设备或技术被适配成用于参考所述图生成倒转神经网络。在至少一个实施例中，这包括倒转一个或更多个神经网络层并按相反的顺序加载一个或更多个神经网络层。在至少一个实施例中，倒转神经网络用于生成训练数据的分布，其是用于训练教师网络的训练数据的分布的传真。在至少一个实施例中，使用所述训练数据的分布来训练学生网络。在至少一个实施例中，所述图的实施例结合了在此关于之前的图1-6所描述的这些和其他各种技术和实施例。

[0630] 鉴于以下条款，可以描述本公开的至少一个实施例：

[0631] 1.一种处理器，包括：一个或更多个电路，用于使一个或更多个神经网络层以相反的顺序被加载到所述处理器中。

[0632] 2.根据条款1的处理器，一个或更多个电路用于生成包括一个或更多个神经网络层的倒转神经网络，一个或更多个神经网络层通过对预训练神经网络中的一个或更多个神经网络层进行倒转生成。

[0633] 3.根据条款1或2所述的处理器，其中一个或更多个神经网络层包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。

[0634] 4.根据条款1-3中任一项所述的处理器，所述一个或更多个电路用于通过至少部分地基于预训练神经网络的卷积层生成一个或更多个转置卷积层来对所述预训练神经网络的一个或更多个卷积层进行倒转。

[0635] 5.根据条款1-4中任一项所述的处理器，一个或更多个电路用于通过至少部分地基于预训练神经网络的一个或更多个批归一化层的线性变换生成一个或更多个倒转批归一化层来对预训练神经网络的一个或更多个批归一化层进行倒转。

[0636] 6.根据条款1-5中任一项所述的处理器，所述一个或更多个电路用于插入激活层。

[0637] 7.根据条款1-6中任一项所述的处理器，其中包括一个或更多个神经网络层的倒转神经网络将输入的传真输出至预训练神经网络。

[0638] 8.根据条款7所述的处理器，所述一个或更多个电路用于使用所述输入的传真来训练神经网络。

[0639] 9.根据条款1-8中任一项所述的处理器，所述一个或更多个电路用于至少部分地基于层一致性损失和重建损失来微调一个或更多个神经网络层。

[0640] 10.一种系统，包括：

[0641] 一个或更多个处理器，用于使一个或更多个神经网络层以相反的顺序加载到所述一个或更多个处理器中。

[0642] 11.根据条款10所述的系统，一个或更多个处理器用于生成包括一个或更多个神经网络层的倒转神经网络，一个或更多个神经网络层通过至少对预训练神经网络中的一个

或更多个神经网络层进行倒转来生成。

[0643] 12.根据条款10或11所述的系统，其中一个或更多个神经网络层包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。

[0644] 13.根据条款12所述的系统，其中所述倒转卷积层至少部分地基于预训练神经网络的卷积层的转置来生成。

[0645] 14.根据条款12或13中任一项所述的系统，其中所述倒转批归一化层至少部分地基于预训练神经网络的批归一化层的线性变换来生成。

[0646] 15.根据条款10-14中任一项所述的系统，其中将激活层添加至一个或更多个神经网络层。

[0647] 16.根据条款10-15中任一项所述的系统，其中包括一个或更多个神经网络层的倒转神经网络的输出包括对预训练神经网络的输入的传真。

[0648] 17.根据条款10-16中任一项所述的系统，一个或更多个处理器用于至少部分地基于层一致性损失和重建损失来微调一个或更多个神经网络层。

[0649] 18.一种机器可读介质，具有存储在其上的一组指令，在由一个或更多个处理器执行所述一组指令时，使所述一个或更多个处理器至少：

[0650] 使一个或更多个神经网络层的输出以相反的顺序被计算。

[0651] 19.根据条款18所述的机器可读介质，具有存储在其上的另一组指令，在由一个或更多个处理器执行该另一组指令时，使一个或更多个处理器至少：

[0652] 生成包括所述一个或更多个神经网络层的倒转神经网络，所述一个或更多个神经网络层至少部分地基于预训练神经网络的一个或更多个神经网络层生成。

[0653] 20.根据条款18或19所述的机器可读介质，其中一个或更多个神经网络层包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。

[0654] 21.根据条款20所述的机器可读介质，其中所述倒转卷积层至少部分地基于预训练神经网络的卷积层的转置来生成。

[0655] 22.根据条款20或21所述的机器可读介质，其中所述倒转批归一化层至少部分地基于预训练神经网络的批归一化层的线性变换来生成。

[0656] 23.根据条款18-22中任一项所述的机器可读介质，具有存储在其上的另一组指令，在由一个或更多个处理器执行该另一组指令时，使一个或更多个处理器至少将激活层添加至一个或更多个神经网络层。

[0657] 24.根据条款18-23中任一项所述的机器可读介质，其中包括一个或更多个神经网络层的倒转神经网络的输出包括对预训练神经网络的输入的传真。

[0658] 25.根据条款18-24中任一项所述的机器可读介质，具有存储在其上的另一组指令，在由一个或更多个处理器执行该另一组指令时，使一个或更多个处理器至少部分地基于层一致性损失和重建损失来至少微调一个或更多个神经网络层。

[0659] 26.一种方法，包括：

[0660] 至少部分地基于第一神经网络生成倒转神经网络；

[0661] 使用所述倒转神经网络来生成训练数据；以及

[0662] 使用所生成的训练数据来训练第二神经网络。

[0663] 27.根据条款26所述的方法，其中倒转神经网络相对于第一神经网络具有反向计

算流。

[0664] 28.根据条款26或27所述的方法,其中倒转神经网络包括倒转全连接层和倒转卷积层或倒转批归一化层中的至少一个。

[0665] 29.根据条款28所述的方法,其中所述倒转卷积层至少部分地基于预训练神经网络的卷积层的转置来生成。

[0666] 30.根据条款28或29所述的方法,其中所述倒转批归一化层至少部分地基于预训练神经网络的批归一化层的线性变换来生成。

[0667] 31.根据条款26-30中任一项所述的方法,还包括:

[0668] 将激活层添加到所述倒转神经网络。

[0669] 32.根据条款26-31中任一项所述的方法,还包括:

[0670] 通过向所述倒转神经网络输入logit或指示编码特征的数据中的至少一者来生成所述训练数据。

[0671] 33.根据条款26-32中任一项所述的方法,还包括:

[0672] 至少部分地基于层一致性损失和重建损失来微调所述倒转网络。

[0673] 在至少一个实施例中,单个半导体平台可以指唯一的单一基于半导体的集成电路或芯片。在至少一个实施例中,可以使用具有增加的连接性的多芯片模块,其模拟芯片上的操作,并且相对于利用传统的中央处理单元(“CPU”)和总线实现方式进行了实质性的改进。在至少一个实施例中,根据用户的需求,各种模块也可以分开放置或以半导体平台的各种组合放置。

[0674] 在至少一个实施例中,返回参考图13,机器可读的可执行代码或计算机控制逻辑算法形式的计算机程序被存储在主存储器1304和/或辅助存储中。根据至少一个实施例,如果由一个或更多个处理器执行,则计算机程序使系统1300能够执行各种功能。在至少一个实施例中,存储器1304、存储和/或任何其他存储是计算机可读介质的可能示例。在至少一个实施例中,辅助存储可以指代任何合适的存储设备或系统,例如硬盘驱动器和/或可移除存储驱动器,其代表软盘驱动器、磁带驱动器、光盘驱动器、数字多功能盘(“DVD”)驱动器、记录设备、通用串行总线(“USB”)闪存等。在至少一个实施例中,各个先前附图的架构和/或功能是在CPU 1302;并行处理系统1312;能够具有两个CPU 1302的至少部分能力的集成电路;并行处理系统1312;芯片组(例如,设计成作为执行相关功能的单元工作并出售的一组集成电路等);和/或集成电路的任何适当组合的环境中实现的。

[0675] 在至少一个实施例中,各个先前附图的架构和/或功能在通用计算机系统、电路板系统、专用于娱乐目的的游戏控制台系统、专用系统等的环境中实现。在至少一个实施例中,计算机系统1300可以采取台式计算机、膝上型计算机、平板电脑、服务器、超级计算机、智能电话(例如,无线、手持设备)、个人数字助理(“PDA”)、数码相机、车辆、头戴式显示器、手持式电子设备、移动电话设备、电视、工作站、游戏机、嵌入式系统和/或任何其他类型的逻辑的形式。

[0676] 在至少一个实施例中,并行处理系统1312包括但不限于多个并行处理单元(“PPU”)1314和相关联的存储器1316。在至少一个实施例中,PPU 1314经由互连1318和交换机1320或多路复用器连接到主机处理器或其他外围设备。在至少一个实施例中,并行处理系统1312在可并行化的PPU 1314上分配计算任务,例如,作为跨多个图形处理单元(“GPU”)

线程块的计算任务分布的一部分。在至少一个实施例中，在PPU 1314中的一些或全部之间共享和访问存储器(例如，用于读取和/或写入访问)，尽管这种共享存储器可能引发相对于使用本地存储器和驻留在PPU1314上的寄存器的性能损失。在至少一个实施例中，通过使用命令(诸如__syncthreads())来同步PPU 1314的操作，其中块中的所有线程(例如，跨多个PPU 1314执行)在进行之前到达某个代码执行点。

[0677] 其他变型在本公开的精神内。因此，尽管公开的技术易于进行各种修改和替代构造，但是某些示出的其实施例在附图中示出并且已经在上面进行了详细描述。然而，应理解，无意将公开内容限制为所公开的一种或更多种特定形式，而是相反，其意图是涵盖落入如所附权利要求书所定义的本公开内容的精神和范围内的所有修改、替代构造和等同物。

[0678] 除非另有说明或显然与上下文矛盾，否则在描述所公开的实施例的上下文中(特别是在所附权利要求的上下文中)，术语“一”和“一个”和“该”以及类似指代的使用应被解释为涵盖单数和复数，而不是作为术语的定义。除非另有说明，否则术语“包括”、“具有”、“包含”和“含有”应被解释为开放式术语(意味着“包括但不限于”)。术语“连接”(在未经修改时指的是物理连接)应解释为部分或全部包含在内、附接到或连接在一起，即使有某些介入。除非本文另外指出，否则本文中对数值范围的引用仅旨在用作分别指代落入该范围内的每个单独值的简写方法，并且每个单独值都被并入说明书中，就如同其在本文中被单独叙述一样。在至少一个实施例中，除非另外指出或与上下文矛盾，否则术语“集”(例如“项目集”)或“子集”的使用应解释为包括一个或更多个成员的非空集合。此外，除非另外指出或与上下文矛盾，否则术语相应集的“子集”不一定表示对应集的适当子集，而是子集和对应集可以相等。

[0679] 除非以其他方式明确指出或与上下文明显矛盾，否则诸如“A,B和C中的至少一个”或“A,B与C中的至少一个”形式的短语之类的连接语在上下文中理解为通常用来表示项目、条款等，其可以是A或B或C，也可以是A和B和C集的任何非空子集。例如，在具有三个成员的集的说明性示例中，连接短语“A,B和C中的至少一个”和“A,B与C中的至少一个”是指以下任意集:{A},{B},{C},{A,B},{A,C},{B,C},{A,B,C}。因此，这种连接语言通常不旨在暗示某些实施例要求存在A中的至少一个，B中的至少一个和C中的至少一个。另外，除非另有说明或与上下文矛盾，否则术语“多个”表示复数的状态(例如，“多个项目”表示多个项目)。在至少一个实施例中，多个项目中项目的数量至少为两个，但如果明确指示或通过上下文指示，则可以更多。此外，除非另有说明或从上下文中可以清楚得知，否则短语“基于”是指“至少部分地基于”而不是“仅基于”。

[0680] 除非本文另外指出或与上下文明显矛盾，否则本文描述的过程的操作可以任何合适的顺序执行。在至少一个实施例中，诸如本文所述的那些过程(或其变形和/或其组合)之类的过程在配置有可执行指令的一个或更多个计算机系统的控制下执行，并且被实现为代码(例如，可执行指令，一个或更多个计算机程序或一个或更多个应用程序)，该代码通过硬件或其组合在一个或更多个处理器上共同执行。在至少一个实施例中，代码以例如计算机程序的形式存储在计算机可读存储介质上，该计算机程序包括可由一个或更多个处理器执行的多个指令。在至少一个实施例中，计算机可读存储介质是非暂时性计算机可读存储介质，其排除了暂时性信号(例如，传播的瞬态电或电磁传输)，但包括非暂时性数据存储电路(例如，缓冲区、高速缓存和队列)。在至少一个实施例中，代码(例如，可执行代码或源代码)

被存储在其上存储有可执行指令的一组一个或更多个非暂时性计算机可读存储介质(或用于存储可执行指令的其他存储器)上,该可执行指令在由计算机系统的一个或更多个处理器执行时(即,作为被执行的结果),使得计算机系统执行本文所述的操作。在至少一个实施例中,一组非暂时性计算机可读存储介质包括多个非暂时性计算机可读存储介质,并且多个非暂时性计算机可读存储介质中的个体非暂时性存储介质中的一个或更多个缺少全部代码,而是多个非暂时性计算机可读存储介质共同存储全部代码。在至少一个实施例中,可执行指令被执行,以使得不同的指令由不同的处理器执行,例如,非暂时性计算机可读存储介质存储指令,并且主中央处理单元("CPU")执行一些指令,而图形处理单元("GPU")执行其他指令。在至少一个实施例中,计算机系统的不同组件具有单独的处理器,并且不同的处理器执行指令的不同子集。

[0681] 因此,在至少一个实施例中,计算机系统被配置为实现单独地或共同地执行本文所述的过程的操作的一个或更多个服务,并且这样的计算机系统被配置有使能实施操作的适用的硬件和/或软件。此外,实现本公开的至少一个实施例的计算机系统是单个设备,并且在另一实施例中是分布式计算机系统,其包括以不同方式操作的多个设备,使得分布式计算机系统执行本文所述的操作,并且使得单个设备不执行所有操作。

[0682] 本文提供的任何和所有示例或示例性语言(例如,"诸如")的使用仅旨在更好地阐明本公开的实施例,并且不对公开的范围构成限制,除非另有要求。说明书中的任何语言都不应被解释为表示任何未要求保护的要素对于实践公开内容是必不可少的。

[0683] 本文引用的所有参考文献,包括出版物、专利申请和专利,均通过引用并入本文,其程度就如同每个参考文献被单独且具体地指示为以引用的方式并入本文并且其全部内容在本文中阐述一样。

[0684] 在说明书和权利要求中,可以使用术语"耦合"和"连接"以及它们的派生词。应当理解,这些术语可能不旨在作为彼此的同义词。相反,在特定示例中,"连接"或"耦合"可用于指示两个或更多个元件彼此直接或间接物理或电接触。"耦合"也可能意味着两个或更多个元素彼此不直接接触,但仍彼此协作或交互。

[0685] 除非另有明确说明,否则可以理解,在整个说明书中,诸如"处理"、"计算"、"计算"、"确定"等之类的术语,是指计算机或计算系统或类似的电子计算设备的动作和/或过程,其将计算系统的寄存器和/或存储器中表示为物理量(例如电子)的数据处理和/或转换为类似表示为计算系统的存储器、寄存器或其他此类信息存储、传输或显示设备中的物理量的其他数据。

[0686] 以类似的方式,术语"处理器"可以指处理来自寄存器和/或存储器的电子数据并将该电子数据转换成可以存储在寄存器和/或存储器中的其他电子数据的任何设备或存储器的一部分。作为非限制性示例,"处理器"可以是CPU或GPU。"计算平台"可以包括一个或更多个处理器。如本文所使用的,"软件"进程可以包括例如随时间执行工作的软件和/或硬件实体,诸如任务、线程和智能代理。同样,每个过程可以指代多个过程,以连续地或间歇地顺序地或并行地执行指令。在至少一个实施例中,术语"系统"和"方法"在本文中可以互换使用,只要系统可以体现一种或更多种方法,并且方法可以被认为是系统。

[0687] 在本文件中,可以参考获得、获取、接收或将模拟或数字数据输入子系统、计算机系统或计算机实现的机器中。在至少一个实施例中,可以通过多种方式来完成获得、获取、

接收或输入模拟和数字数据的过程，例如通过接收作为函数调用或对应用程序编程接口的调用的参数的数据。在至少一个实施例中，可以通过经由串行或并行接口传输数据来完成获得、获取、接收或输入模拟或数字数据的过程。在至少一个实施例中，可以通过经由计算机网络将数据从提供实体传输到获取实体来完成获得、获取、接收或输入模拟或数字数据的过程。在至少一个实施例中，也可以参考提供、输出、传送、发送或呈现模拟或数字数据。在各种示例中，提供、输出、传送、发送或呈现模拟或数字数据的过程可以通过将数据作为函数调用的输入或输出参数、应用程序编程接口或进程间通信机制的参数进行传输来实现。

[0688] 尽管本文的描述阐述了所描述的技术的示例实现，但是其他架构可以用于实现所描述的功能，并且旨在落入本公开的范围内。此外，尽管出于描述的目的可以在上面定义了具体的职责分配，但是根据情况，可以以不同的方式分配和划分各种功能和职责。

[0689] 此外，尽管已经用特定于结构特征和/或方法动作的语言描述了主题，但是应当理解，所附权利要求书所要求保护的主题不必限于所描述的特定特征或动作。而是，公开了特定的特征和动作作为实现权利要求的示例性形式。

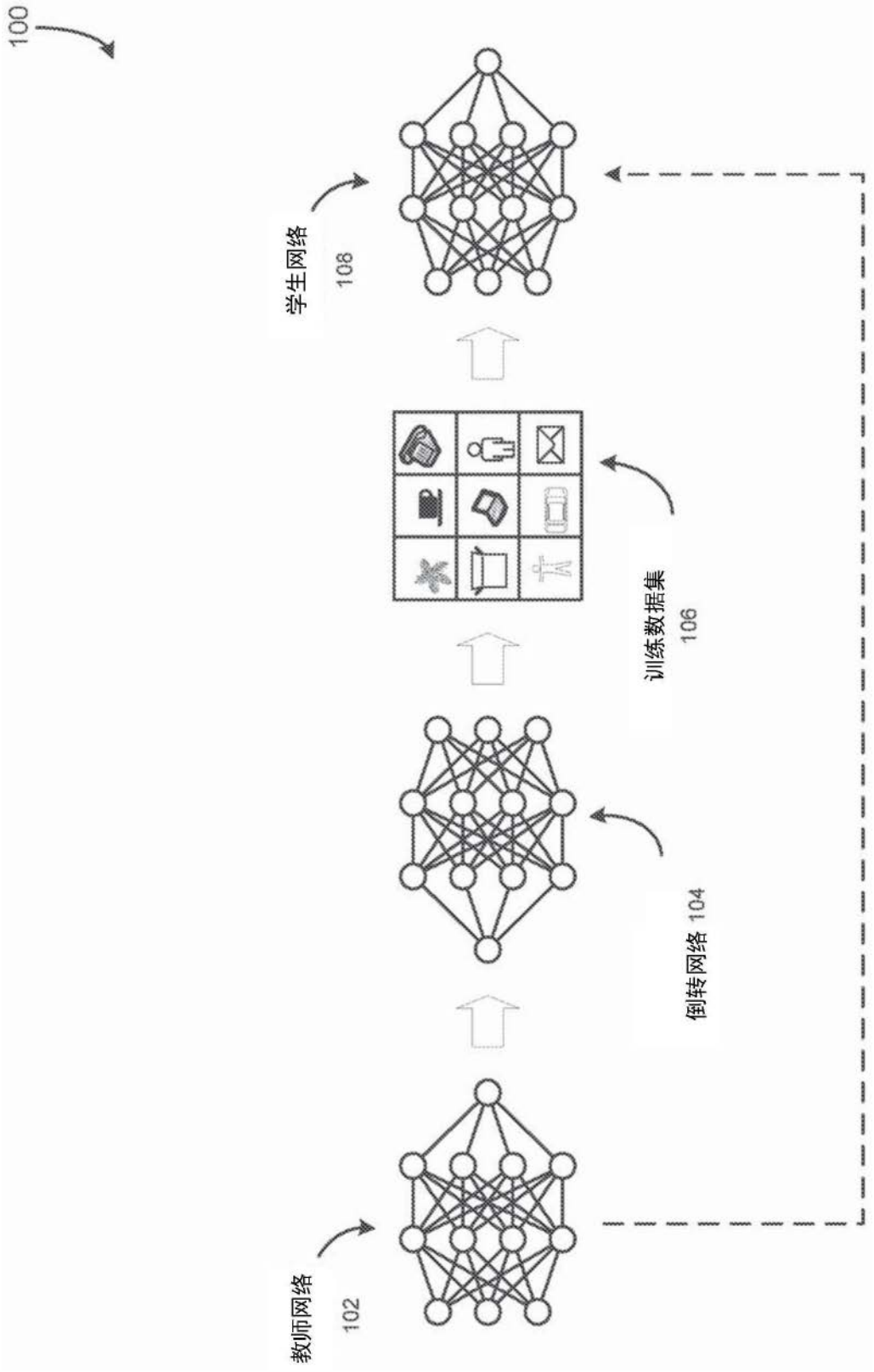


图1

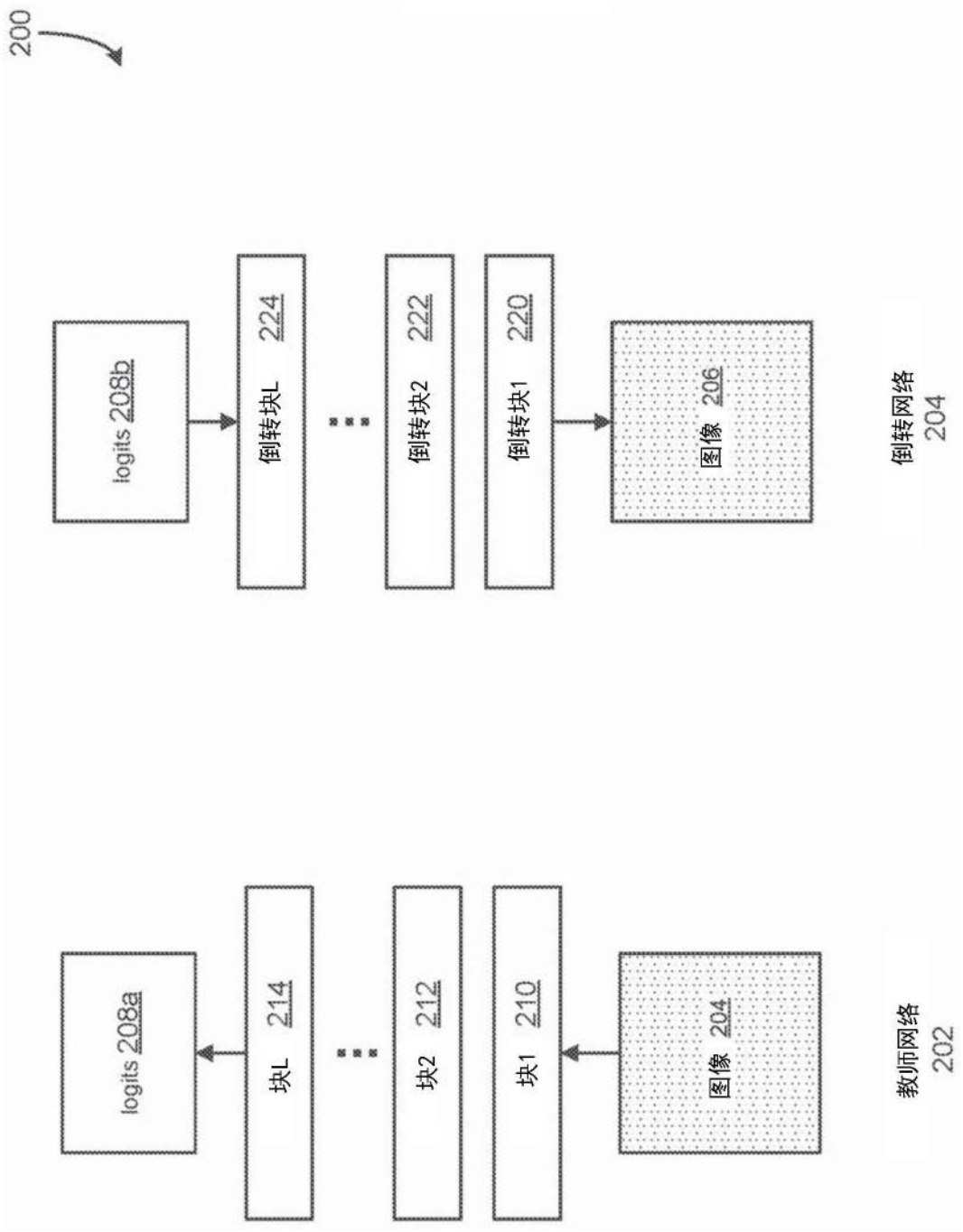

200
logits 208b
倒转块L 224
倒转块2 222
倒转块1 220
图像 206
倒转网络 204
logits 208a
块L 214
块2 212
块1 210
图像 204
教师网络 202

图2

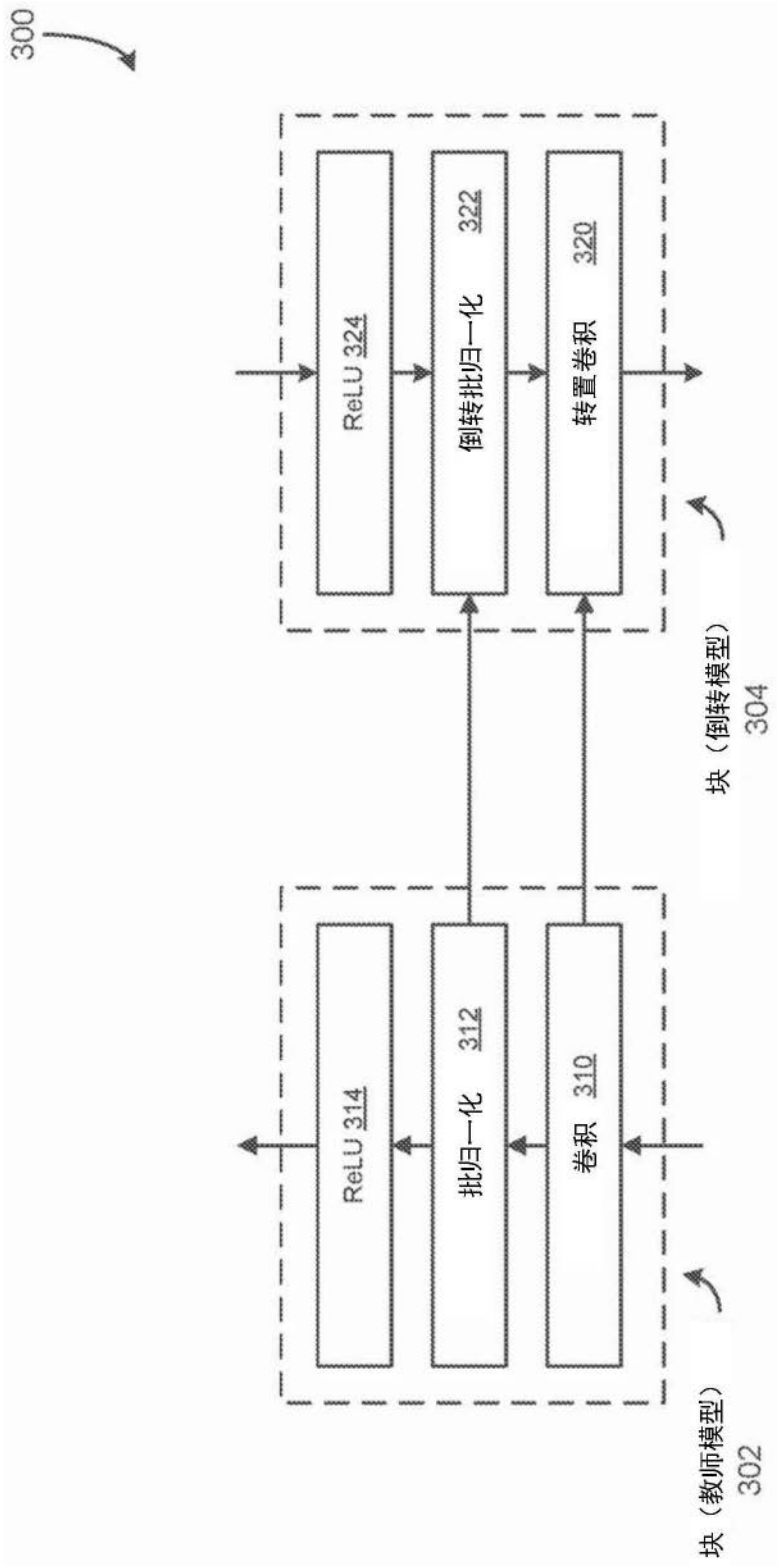

300
ReLU 324
倒转批归一化 322
转置卷积 320
块（倒转模型）304
ReLU 314
批归一化 312
卷积 310
块（教师模型）302

图3

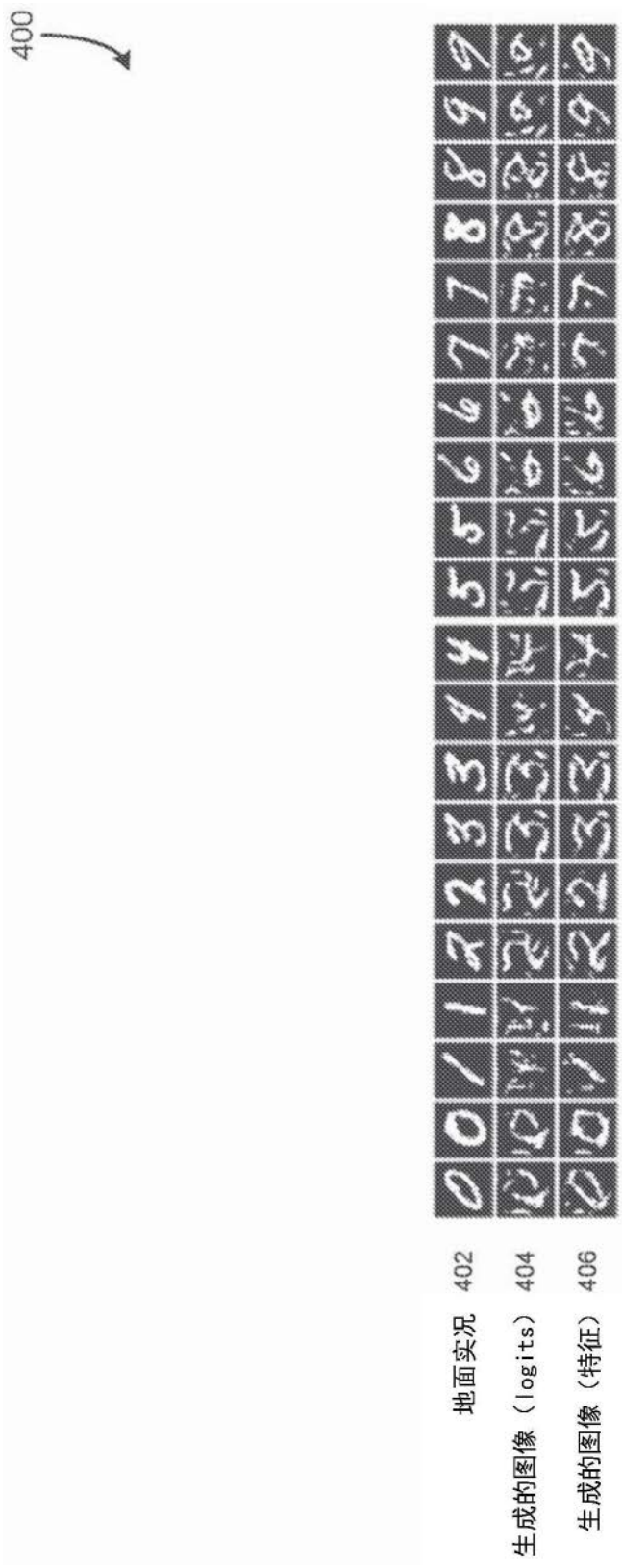


图4

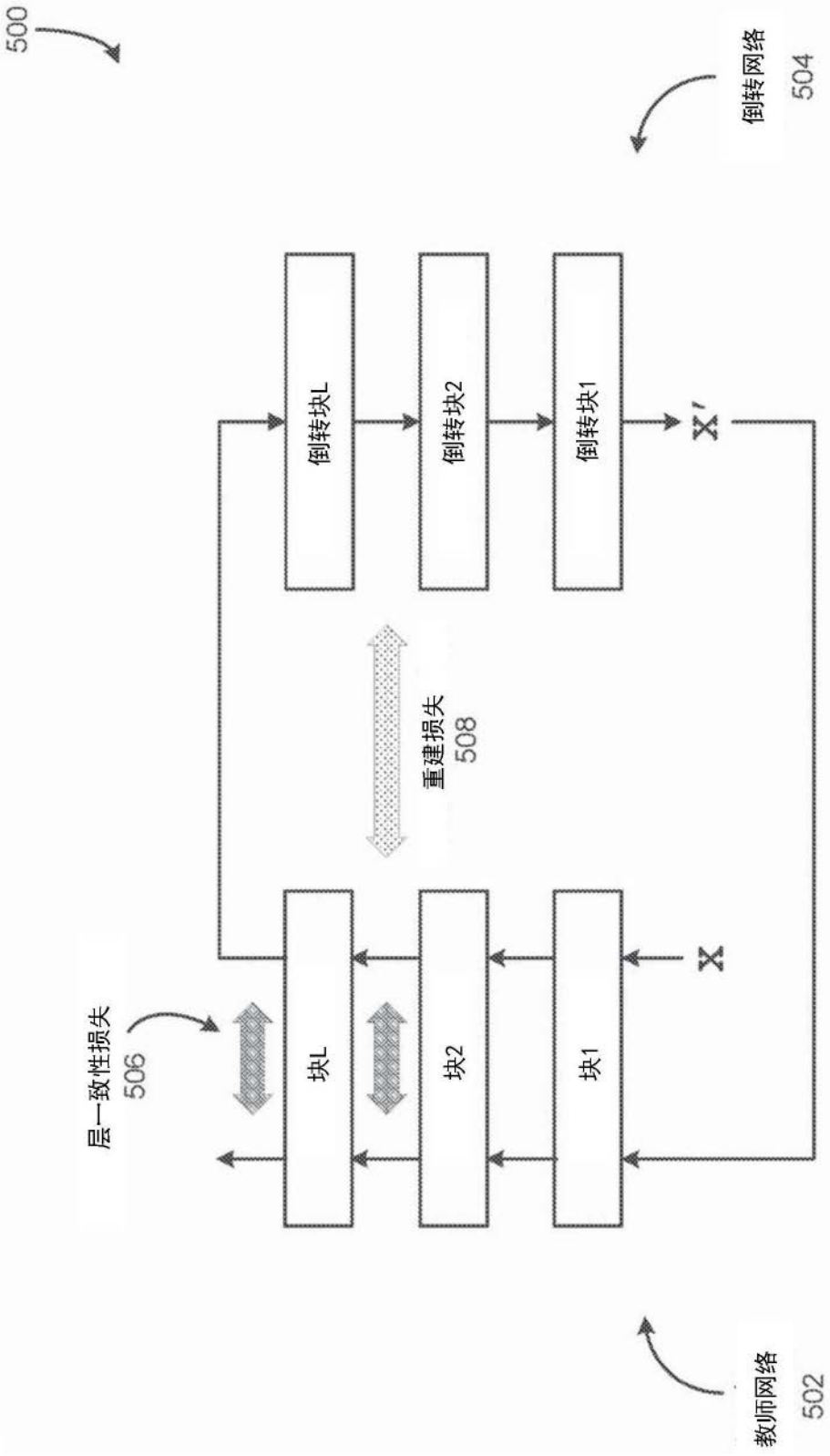

500
倒转网络
504
倒转块L
倒转块2
倒转块1
X′
重建损失
508
层一致性损失
506
块L
块2
块1
X
教师网络
502

图5

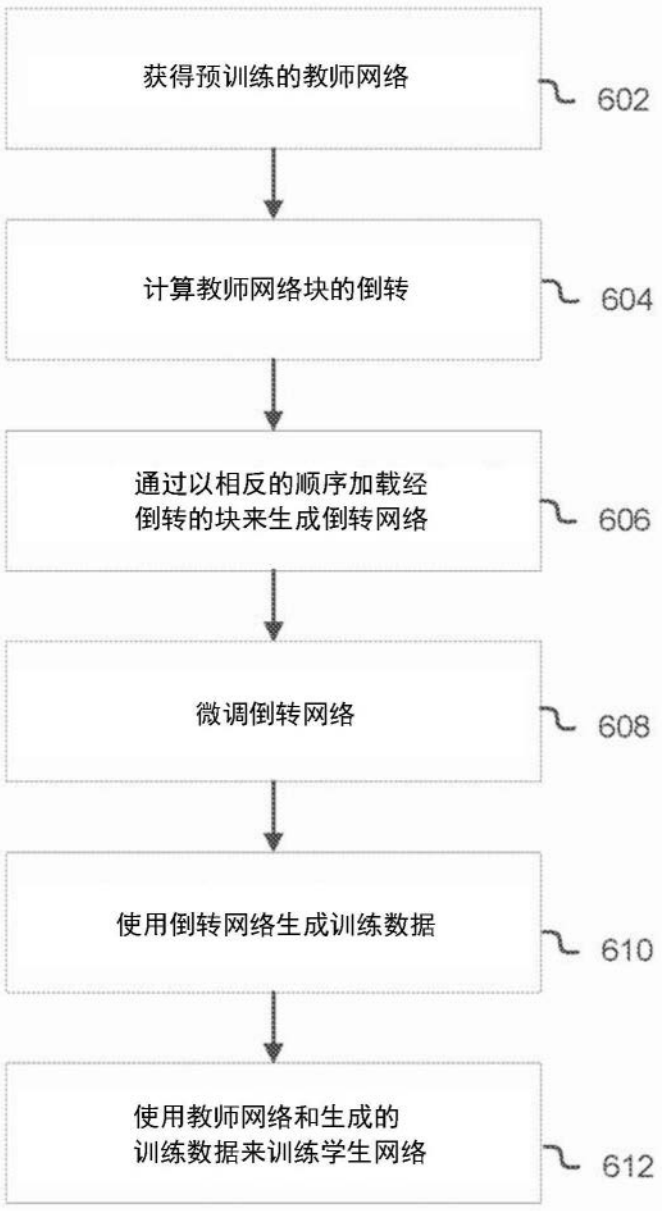

获得预训练的教师网络
602
计算教师网络块的倒转
604
通过以相反的顺序加载经
倒转的块来生成倒转网络
606
微调倒转网络
608
使用倒转网络生成训练数据
610
使用教师网络和生成的
训练数据来训练学生网络
612

图6

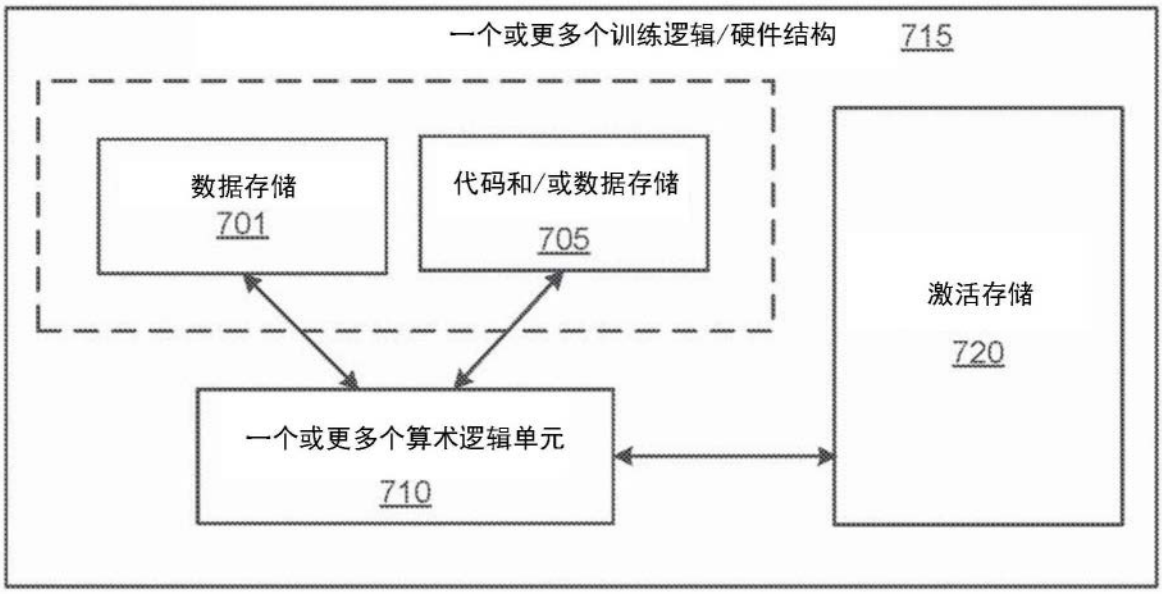

一个或更多个训练逻辑/硬件结构
715
数据存储
701
代码和/或数据存储
705
激活存储
720
一个或更多个算术逻辑单元
710

图7A

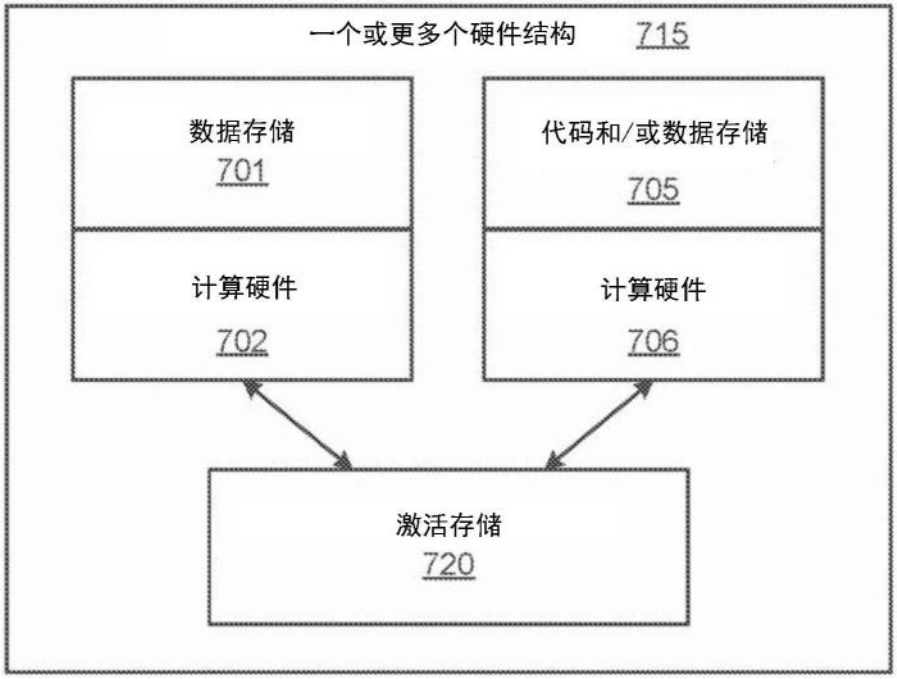

一个或更多个硬件结构
715
数据存储
701
代码和/或数据存储
705
计算硬件
702
计算硬件
706
激活存储
720

图7B

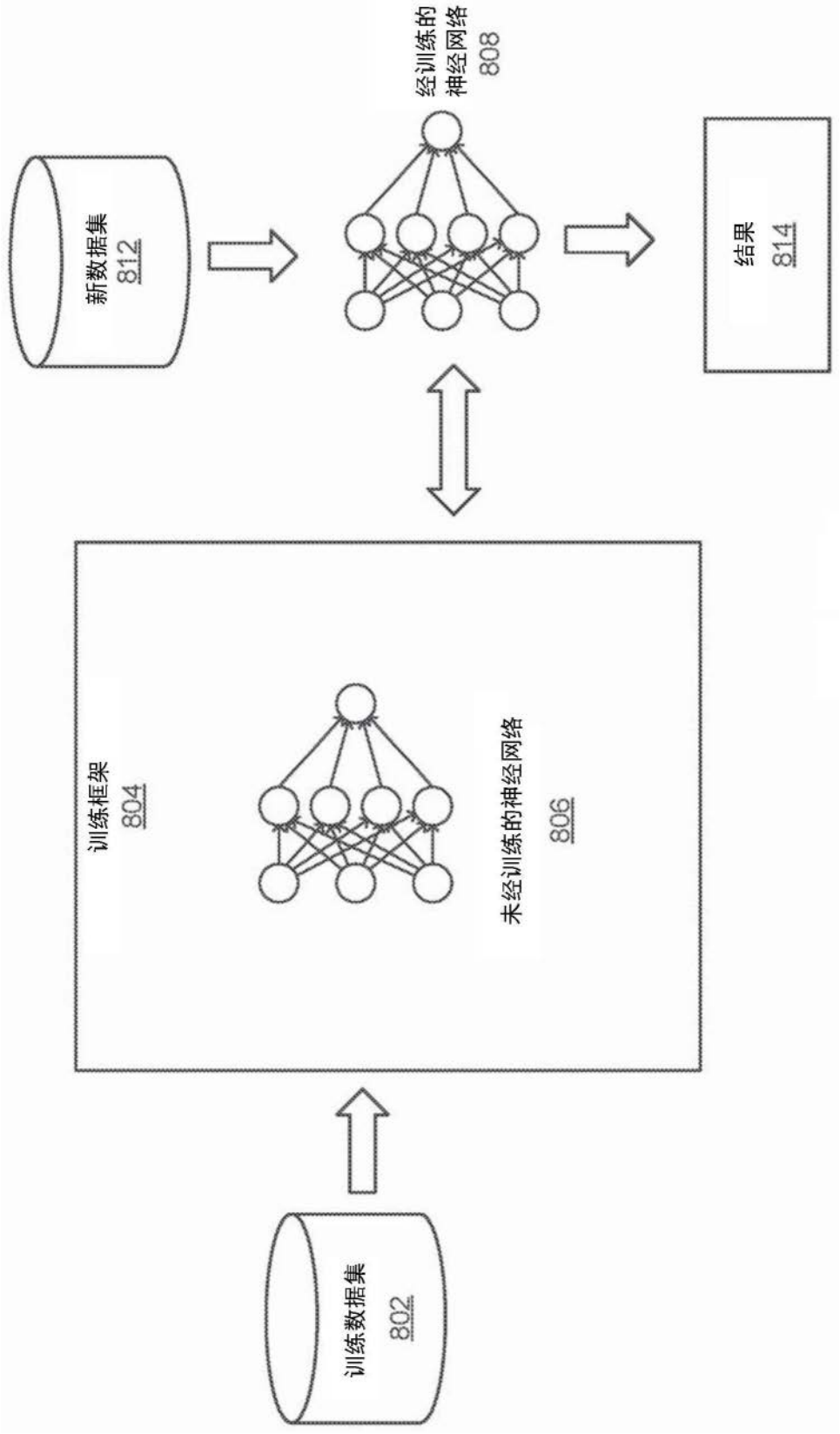

经训练的
神经网络
808
新数据集
812
结果
814
训练框架
804
未经训练的神经网络
806
训练数据集
802

图8

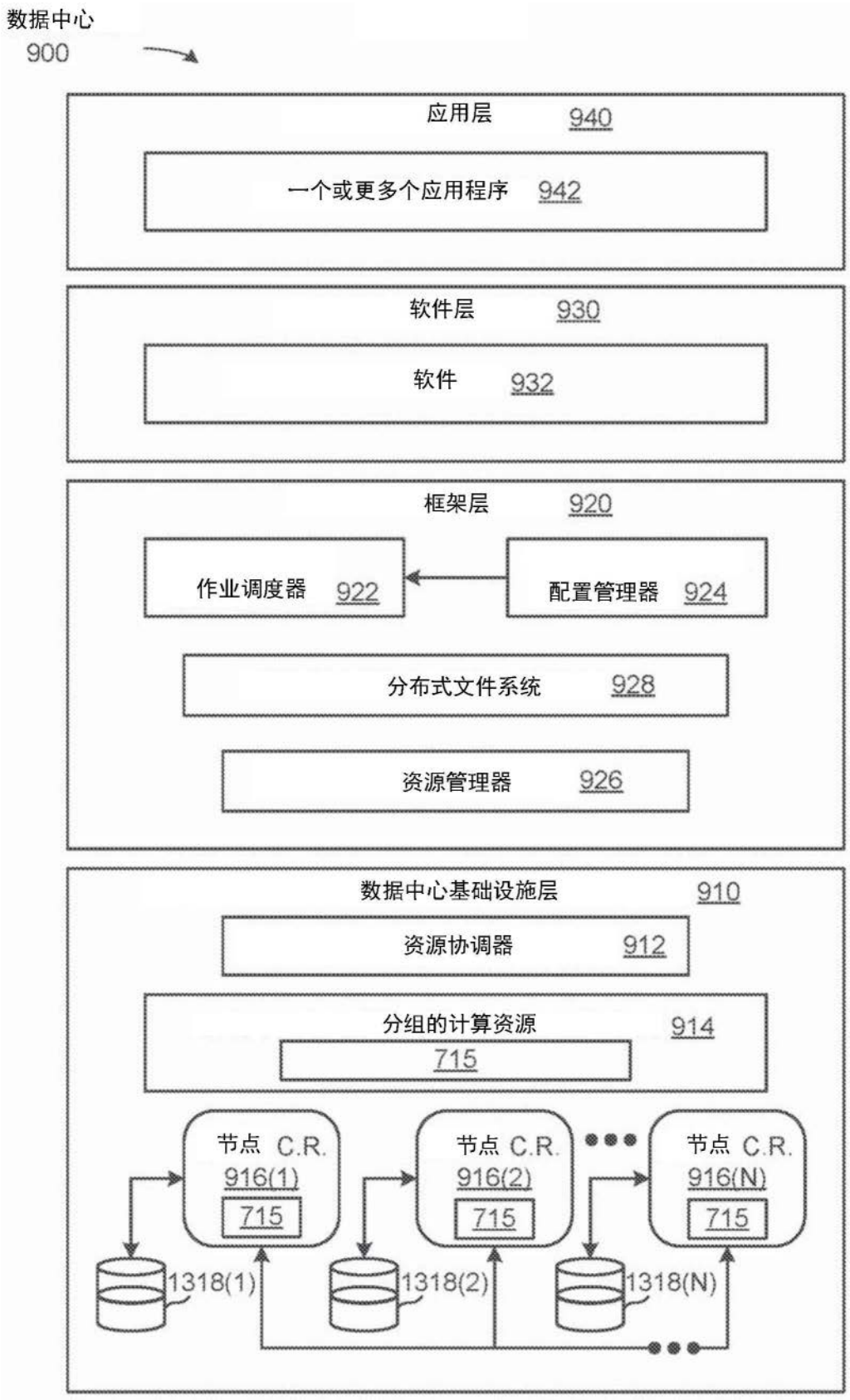

数据中心
900
应用层
940
一个或更多个应用程序
942
软件层
930
软件
932
框架层
920
作业调度器
922
配置管理器
924
分布式文件系统
928
资源管理器
926
数据中心基础设施层
910
资源协调器
912
分组的计算资源
914
715
节点 C.R.
916(1)
715
节点 C.R.
916(2)
715
节点 C.R.
916(N)
715
1318(1)
1318(2)
1318(N)

图9

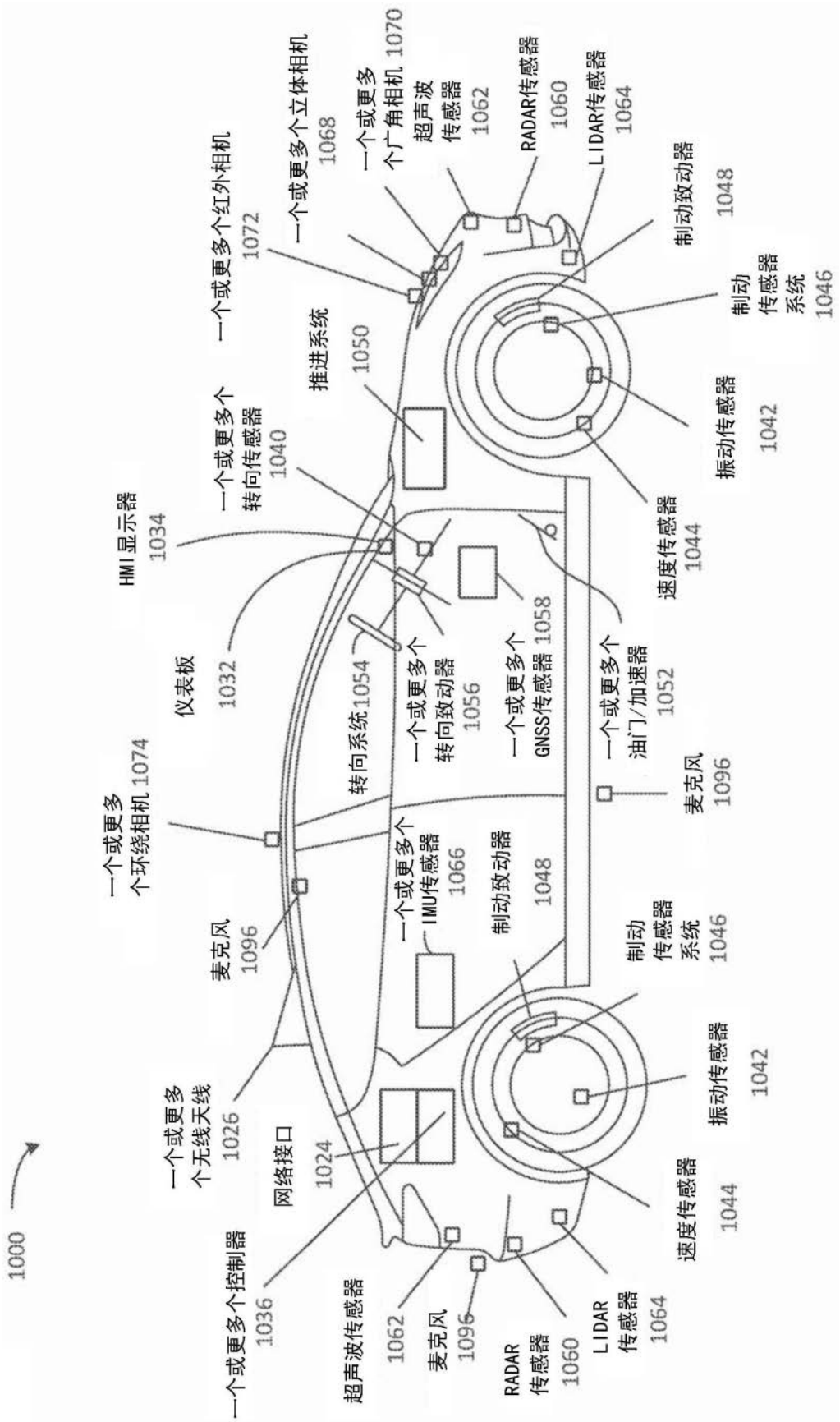

1000
一个或更多个控制器 1036
超声波传感器 1062
麦克风 1096
RADAR 传感器 1060
LIDAR 传感器 1064
速度传感器 1044
振动传感器 1042
制动传感器系统 1046
制动致动器 1048
一个或更多个 IMU传感器 1066
麦克风 1096
网络接口 1024
一个或更多个无线天线 1026
一个或更多个环绕相机 1074
转向系统 1054
一个或更多个转向致动器 1056
一个或更多个 GNSS传感器 1058
一个或更多个油门/加速器 1052
麦克风 1096
速度传感器 1044
振动传感器 1042
制动传感器系统 1046
制动致动器 1048
LIDAR传感器 1064
RADAR传感器 1060
一个或更多个广角相机 1070
超声波传感器 1062
一个或更多个立体相机 1068
一个或更多个红外相机 1072
推进系统 1050
一个或更多个转向传感器 1040
仪表板 1032
HMI 显示器 1034

图10A

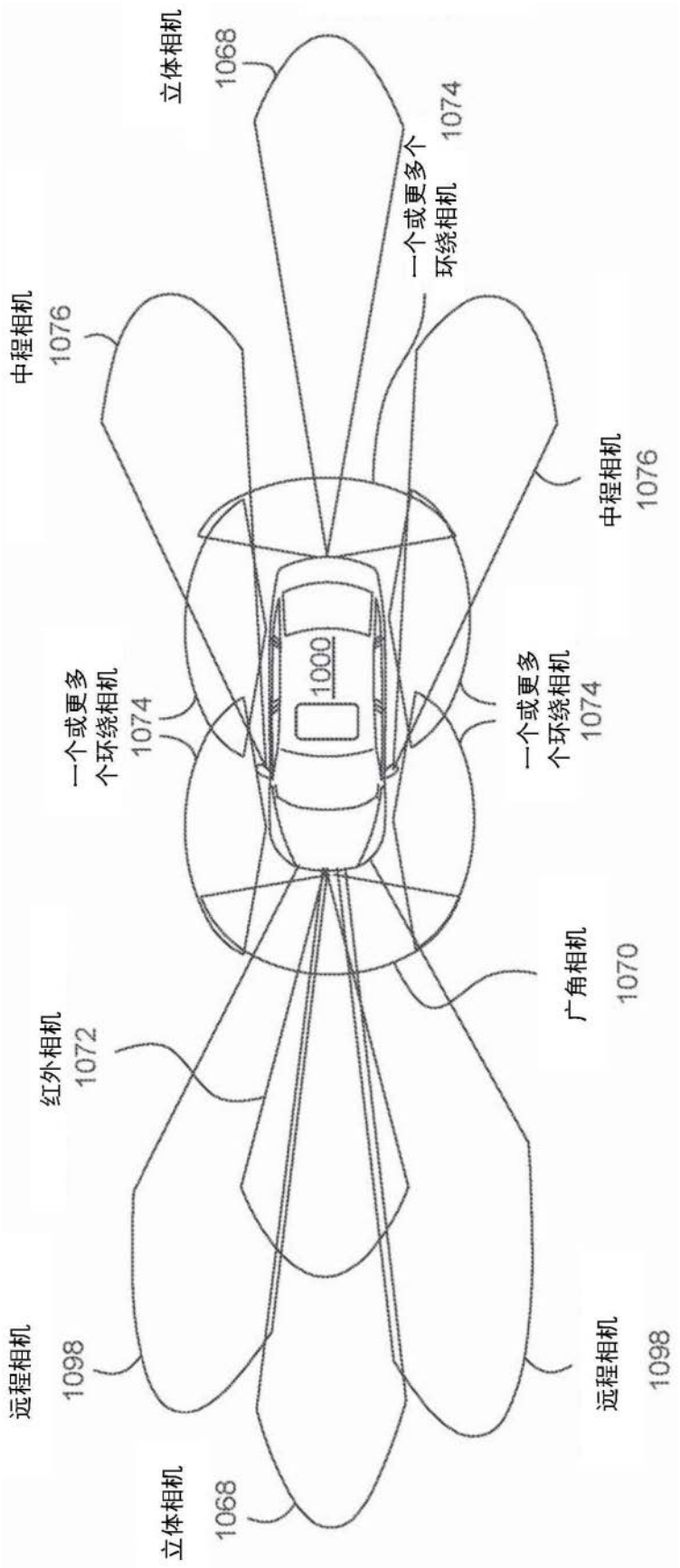

立体相机
1068
一个或更多个
环绕相机
1074
中程相机
1076
中程相机
1076
一个或更多
个环绕相机
1074
1000
一个或更多
个环绕相机
1074
广角相机
1070
红外相机
1072
远程相机
1098
远程相机
1098
立体相机
1068

图10B

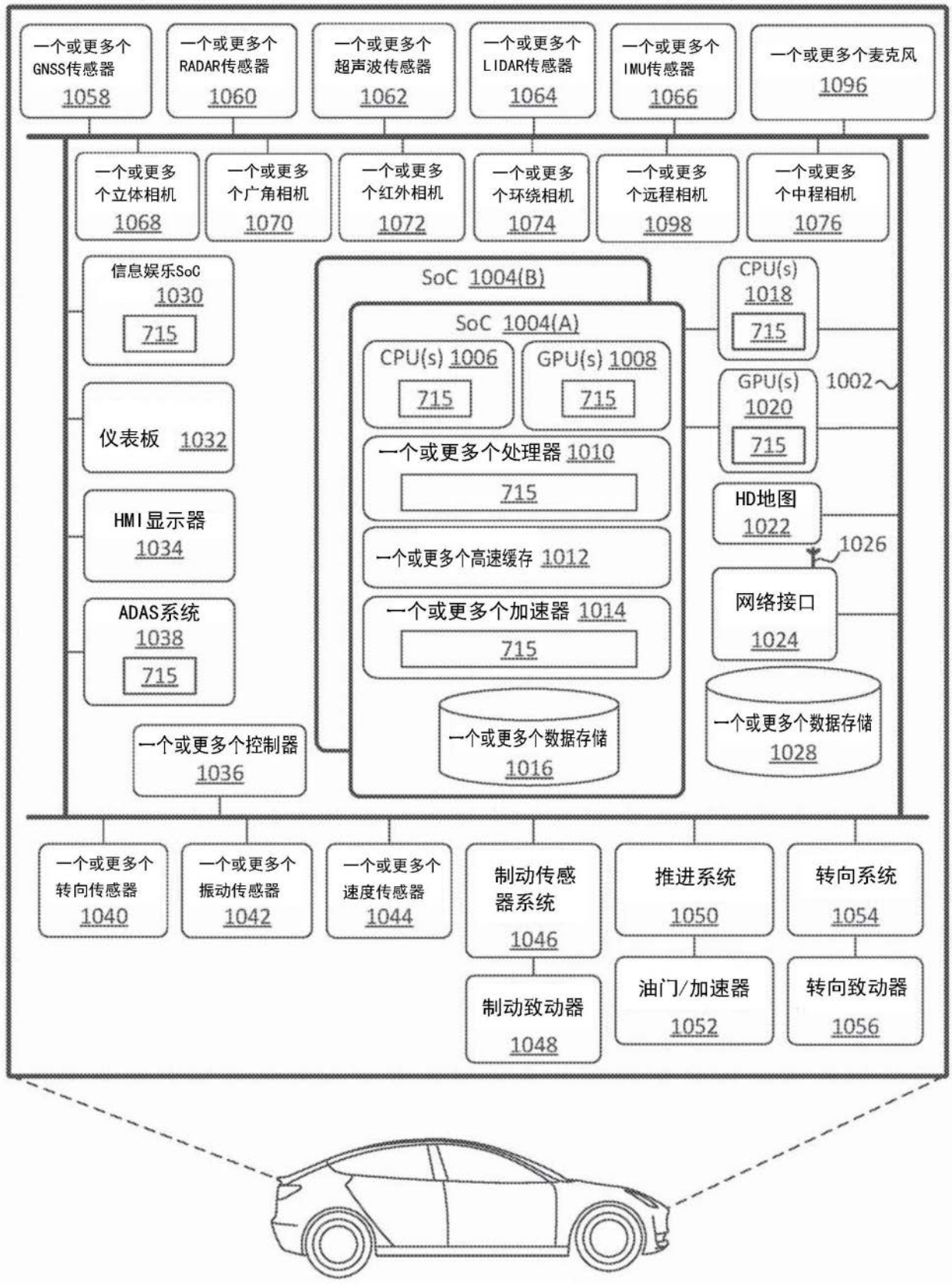

一个或更多个 GNSS传感器 1058
一个或更多个 RADAR传感器 1060
一个或更多个 超声波传感器 1062
一个或更多个 LIDAR传感器 1064
一个或更多个 IMU传感器 1066
一个或更多个麦克风 1096
一个或更多个立体相机 1068
一个或更多个广角相机 1070
一个或更多个红外相机 1072
一个或更多个环绕相机 1074
一个或更多个远程相机 1098
一个或更多个中程相机 1076
信息娱乐SoC 1030
715
仪表板 1032
HMI显示器 1034
ADAS系统 1038
715
一个或更多个控制器 1036
SoC 1004(B)
SoC 1004(A)
CPU(s) 1006
715
GPU(s) 1008
715
一个或更多个处理器 1010
715
一个或更多个高速缓存 1012
一个或更多个加速器 1014
715
一个或更多个数据存储 1016
CPU(s) 1018
715
GPU(s) 1020
715
1002
HD地图 1022
1026
网络接口 1024
一个或更多个数据存储 1028
一个或更多个 转向传感器 1040
一个或更多个 振动传感器 1042
一个或更多个 速度传感器 1044
制动传感器系统 1046
制动致动器 1048
推进系统 1050
油门/加速器 1052
转向系统 1054
转向致动器 1056

图10C

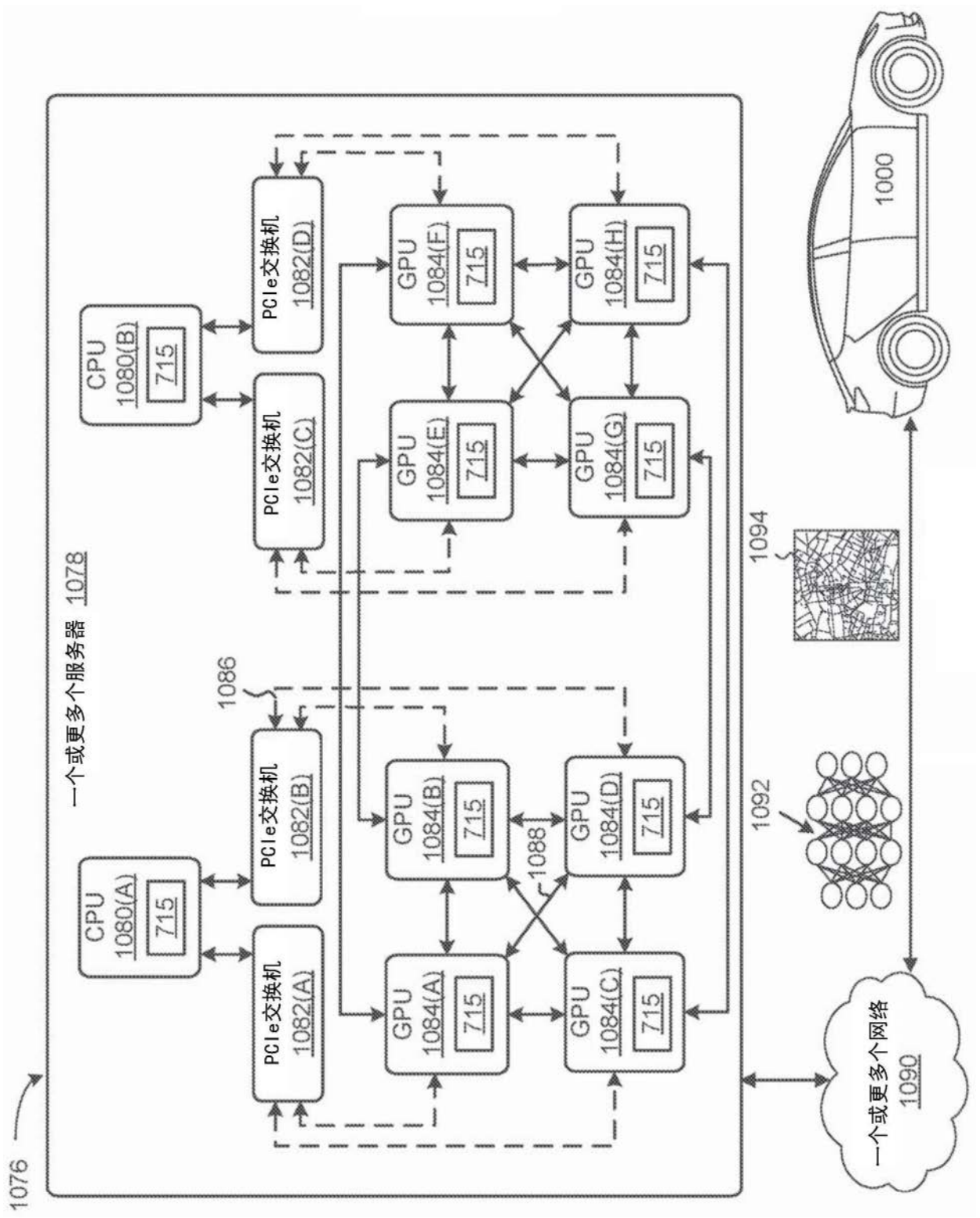

1076
一个或更多个服务器 1078
CPU 1080(A) Z15
CPU 1080(B) Z15
PCIe交换机 1082(A)
PCIe交换机 1082(B)
PCIe交换机 1082(C)
PCIe交换机 1082(D)
1086
1088
GPU 1084(A) Z15
GPU 1084(B) Z15
GPU 1084(C) Z15
GPU 1084(D) Z15
GPU 1084(E) Z15
GPU 1084(F) Z15
GPU 1084(G) Z15
GPU 1084(H) Z15
1000
1092
1094
一个或更多个网络 1090

图10D

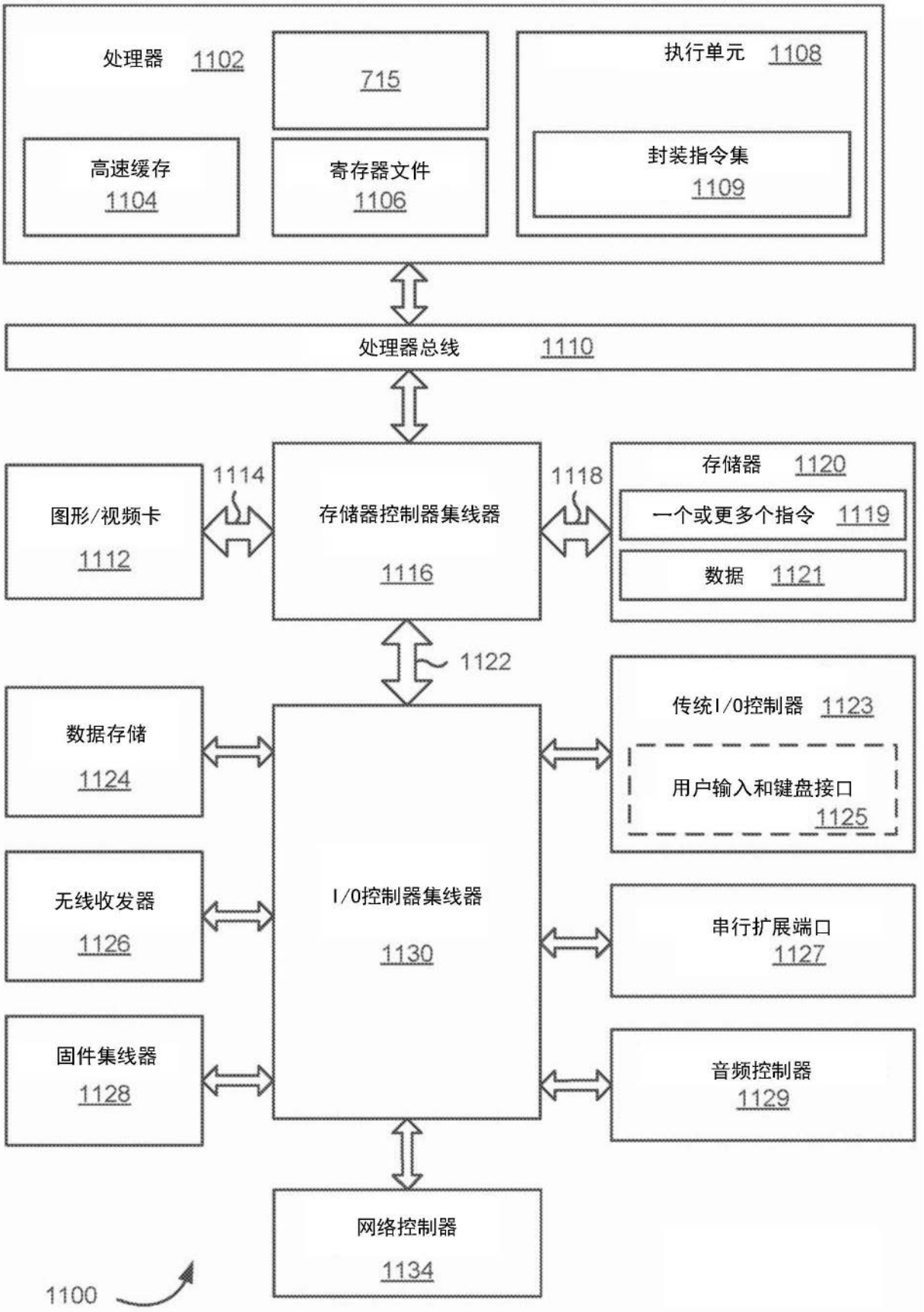

处理器 1102
715
执行单元 1108
高速缓存 1104
寄存器文件 1106
封装指令集 1109
处理器总线 1110
1114
1118
图形/视频卡 1112
存储器控制器集线器 1116
存储器 1120
一个或更多个指令 1119
数据 1121
1122
数据存储 1124
传统I/O控制器 1123
用户输入和键盘接口 1125
无线收发器 1126
I/O控制器集线器 1130
串行扩展端口 1127
固件集线器 1128
音频控制器 1129
网络控制器 1134
1100

图11

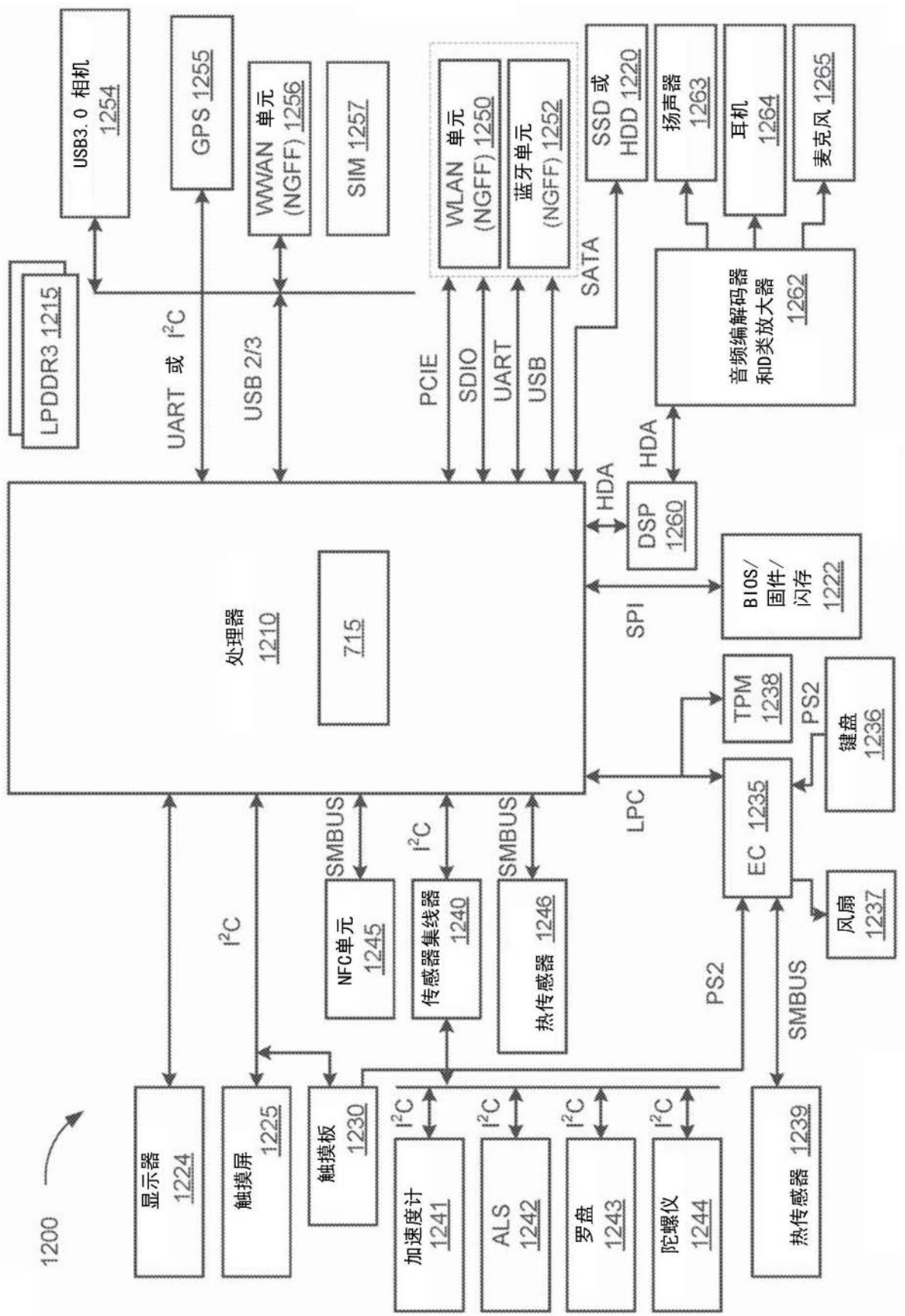

USB3.0 相机 1254
GPS 1255
WWAN 单元 (NGFF) 1256
SIM 1257
WLAN 单元 (NGFF) 1250
蓝牙单元 (NGFF) 1252
SSD 或 HDD 1220
扬声器 1263
耳机 1264
麦克风 1265
LPDDR3 1215
UART 或 I²C
USB 2/3
PCIE
SDIO
UART
USB
SATA
音频编解码器和D类放大器 1262
HDA
HDA
DSP 1260
处理器 1210
Z15
SPI
BIOS/固件/闪存 1222
TPM 1238
PS2
键盘 1236
LPC
EC 1235
风扇 1237
SMBUS
I²C
SMBUS
NFC单元 1245
传感器集线器 1240
热传感器 1246
I²C
PS2
SMBUS
显示器 1224
触摸屏 1225
触摸板 1230
I²C
I²C
I²C
I²C
加速度计 1241
ALS 1242
罗盘 1243
陀螺仪 1244
热传感器 1239
1200

图12

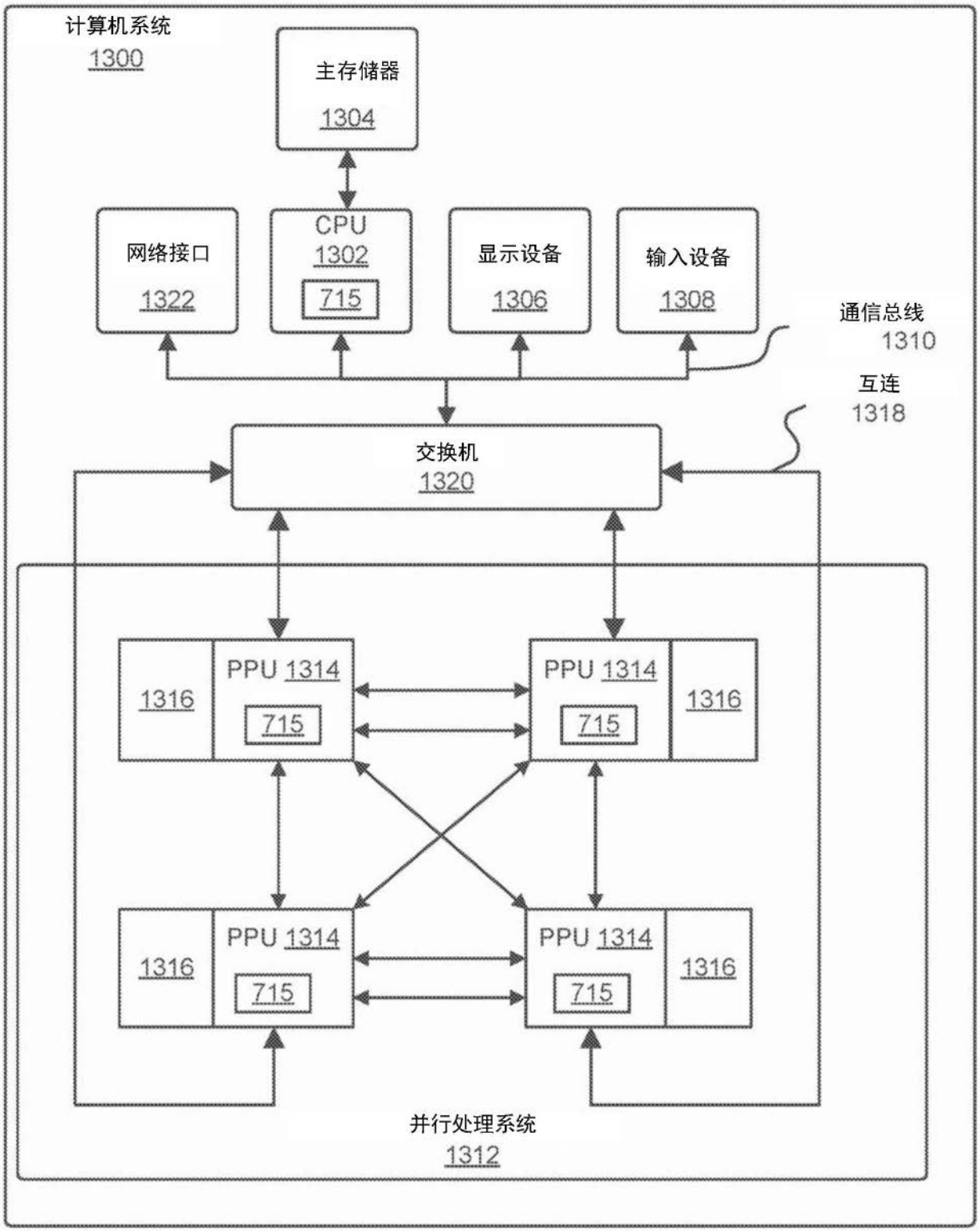

计算机系统
1300
主存储器
1304
网络接口
1322
CPU
1302
715
显示设备
1306
输入设备
1308
通信总线
1310
互连
1318
交换机
1320
1316
PPU 1314
715
PPU 1314
715
1316
1316
PPU 1314
715
PPU 1314
715
1316
并行处理系统
1312

图13

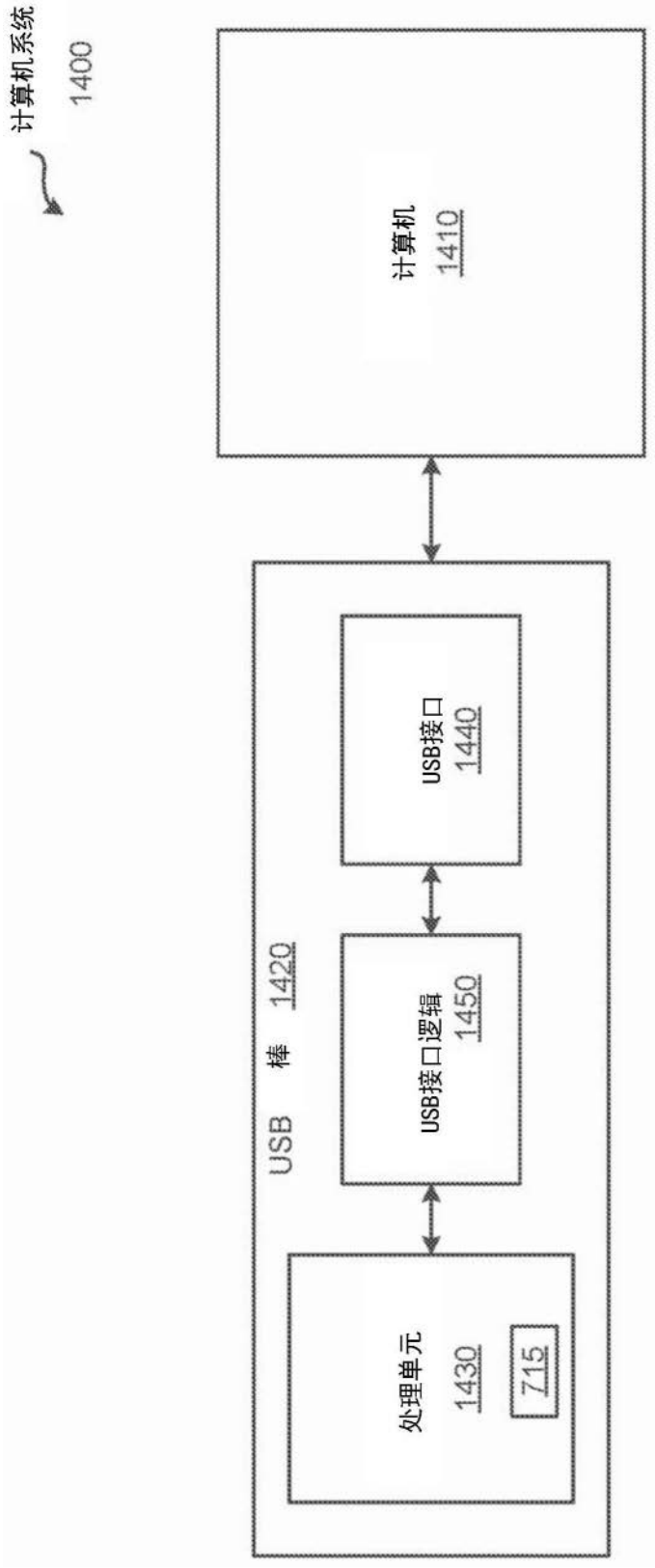

计算机系统
1400
计算机
1410
USB 棒 1420
USB接口
1440
USB接口逻辑
1450
处理单元
1430
Z15

图14

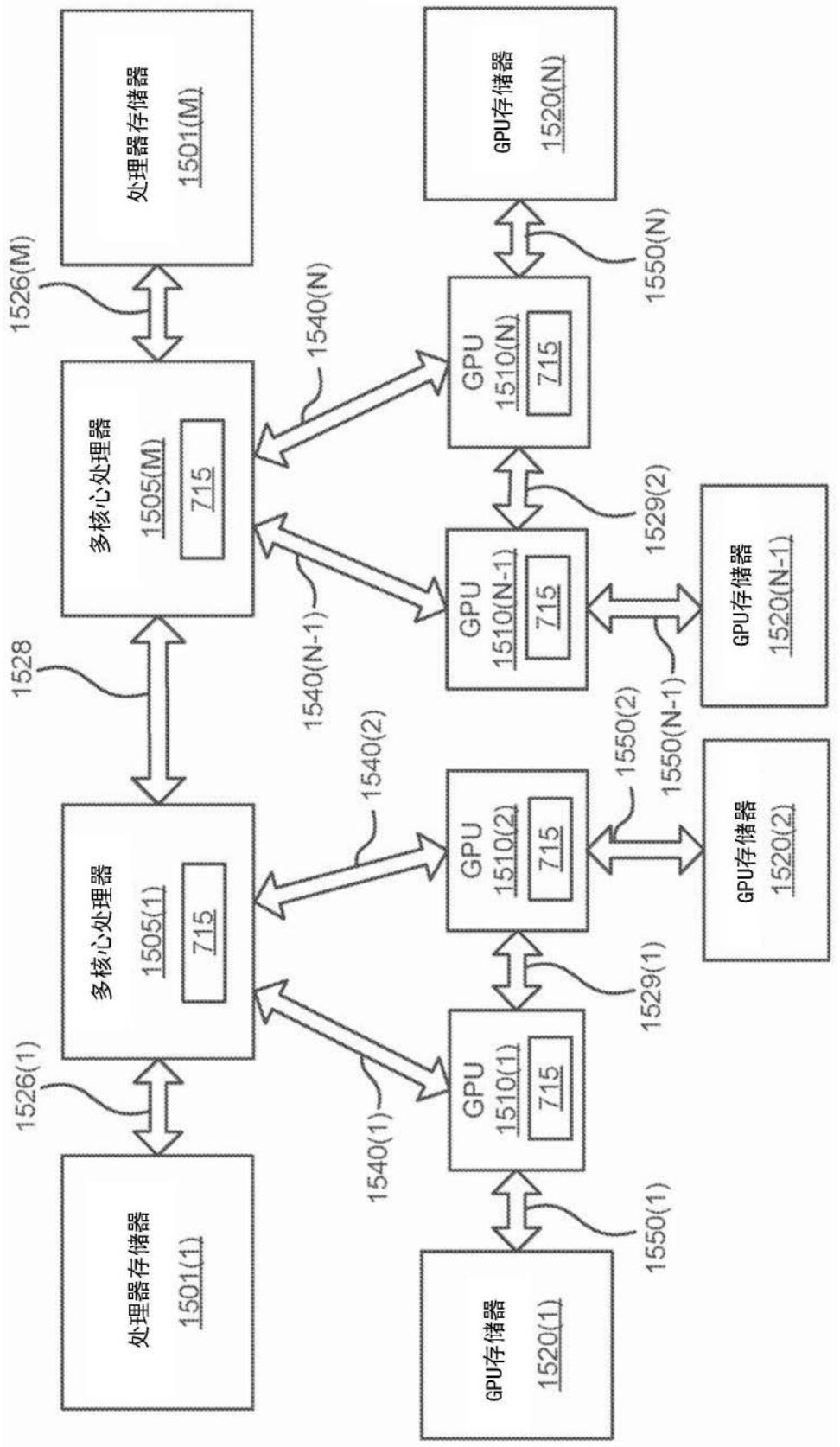

处理器存储器
1501(M)
GPU存储器
1520(N)
1526(M)
1550(N)
1540(N)
多核心处理器
1505(M)
Z15
GPU
1510(N)
Z15
1529(2)
1528
1540(N-1)
GPU
1510(N-1)
Z15
GPU存储器
1520(N-1)
1550(2)
1550(N-1)
1540(2)
多核心处理器
1505(1)
Z15
GPU
1510(2)
Z15
GPU存储器
1520(2)
1526(1)
1529(1)
1540(1)
GPU
1510(1)
Z15
处理器存储器
1501(1)
1550(1)
GPU存储器
1520(1)

图15A

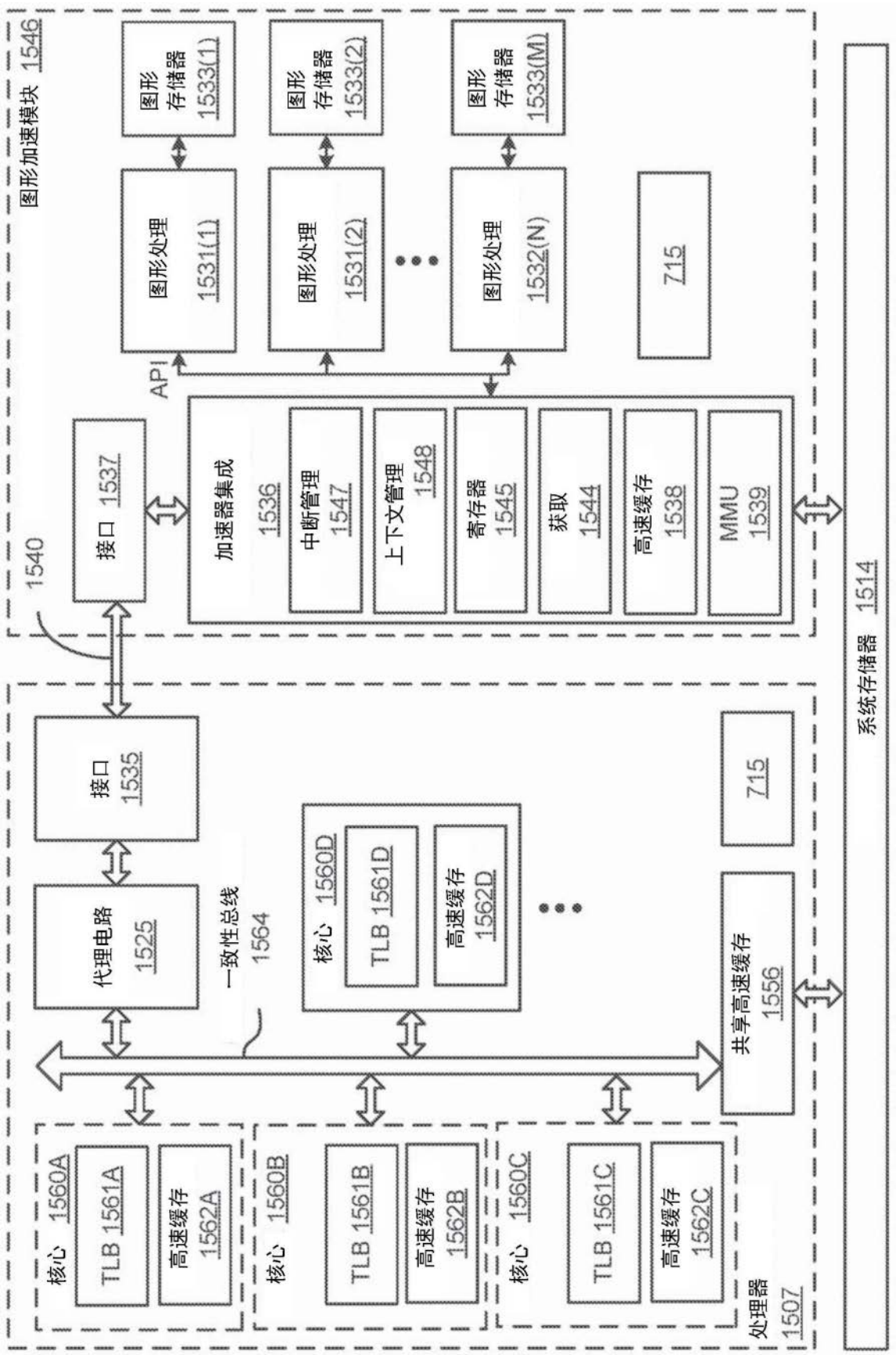

图形加速模块 1546
图形存储器 1533(1)
图形存储器 1533(2)
图形存储器 1533(M)
图形处理 1531(1)
图形处理 1531(2)
图形处理 1532(N)
Z15
API
1540
接口 1537
加速器集成 1536
中断管理 1547
上下文管理 1548
寄存器 1545
获取 1544
高速缓存 1538
MMU 1539
接口 1535
代理电路 1525
一致性总线 1564
核心 1560D
TLB 1561D
高速缓存 1562D
Z15
共享高速缓存 1556
系统存储器 1514
核心 1560A
TLB 1561A
高速缓存 1562A
核心 1560B
TLB 1561B
高速缓存 1562B
核心 1560C
TLB 1561C
高速缓存 1562C
处理器 1507

图15B

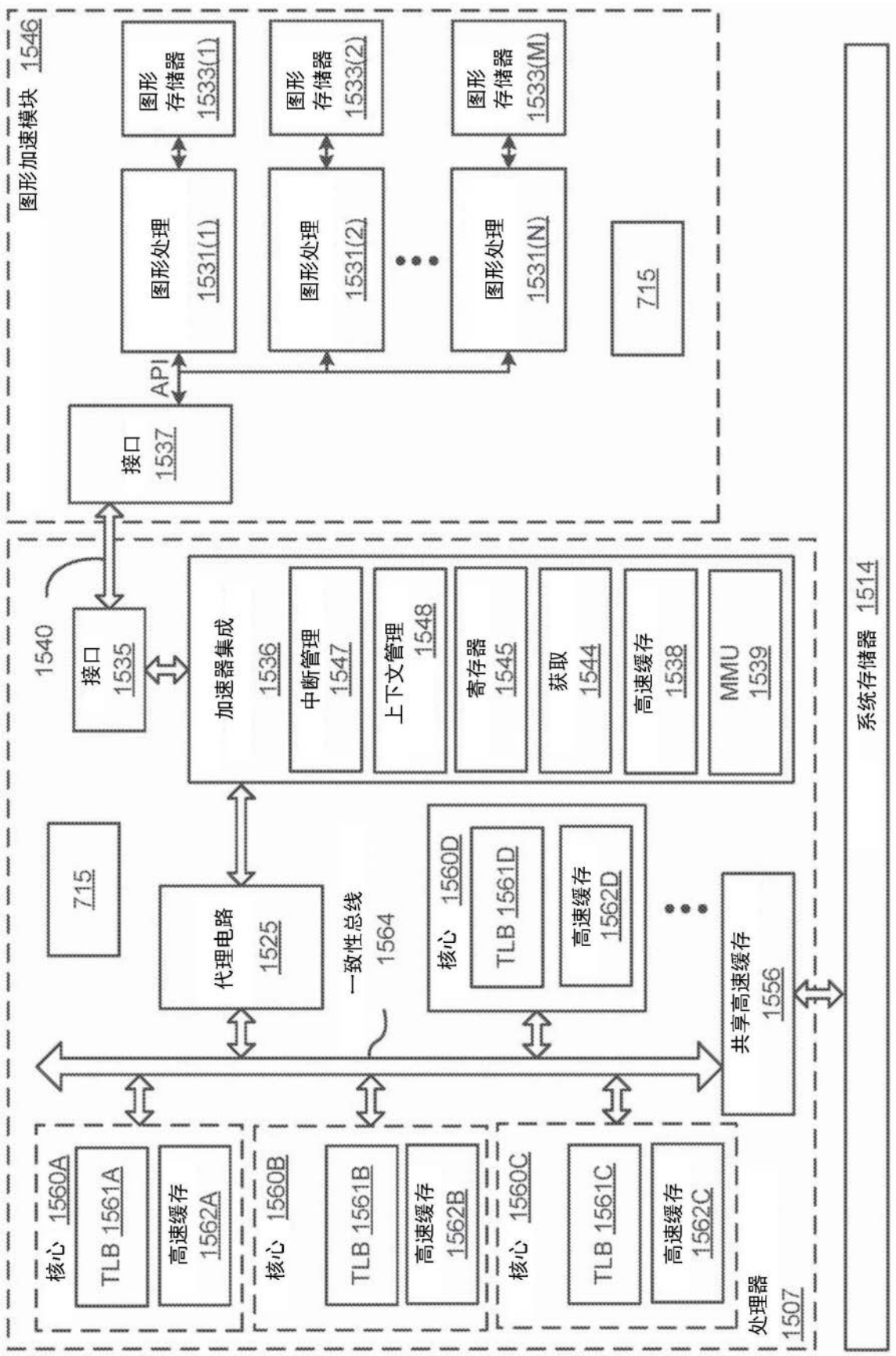

图形加速模块 1546
图形存储器 1533(1)
图形存储器 1533(2)
图形存储器 1533(M)
图形处理 1531(1)
图形处理 1531(2)
图形处理 1531(N)
715
API
接口 1537
1540
接口 1535
加速器集成 1536
中断管理 1547
上下文管理 1548
寄存器 1545
获取 1544
高速缓存 1538
MMU 1539
系统存储器 1514
715
代理电路 1525
一致性总线 1564
核心 1560D
TLB 1561D
高速缓存 1562D
共享高速缓存 1556
核心 1560A
TLB 1561A
高速缓存 1562A
核心 1560B
TLB 1561B
高速缓存 1562B
核心 1560C
TLB 1561C
高速缓存 1562C
处理器 1507

图15C

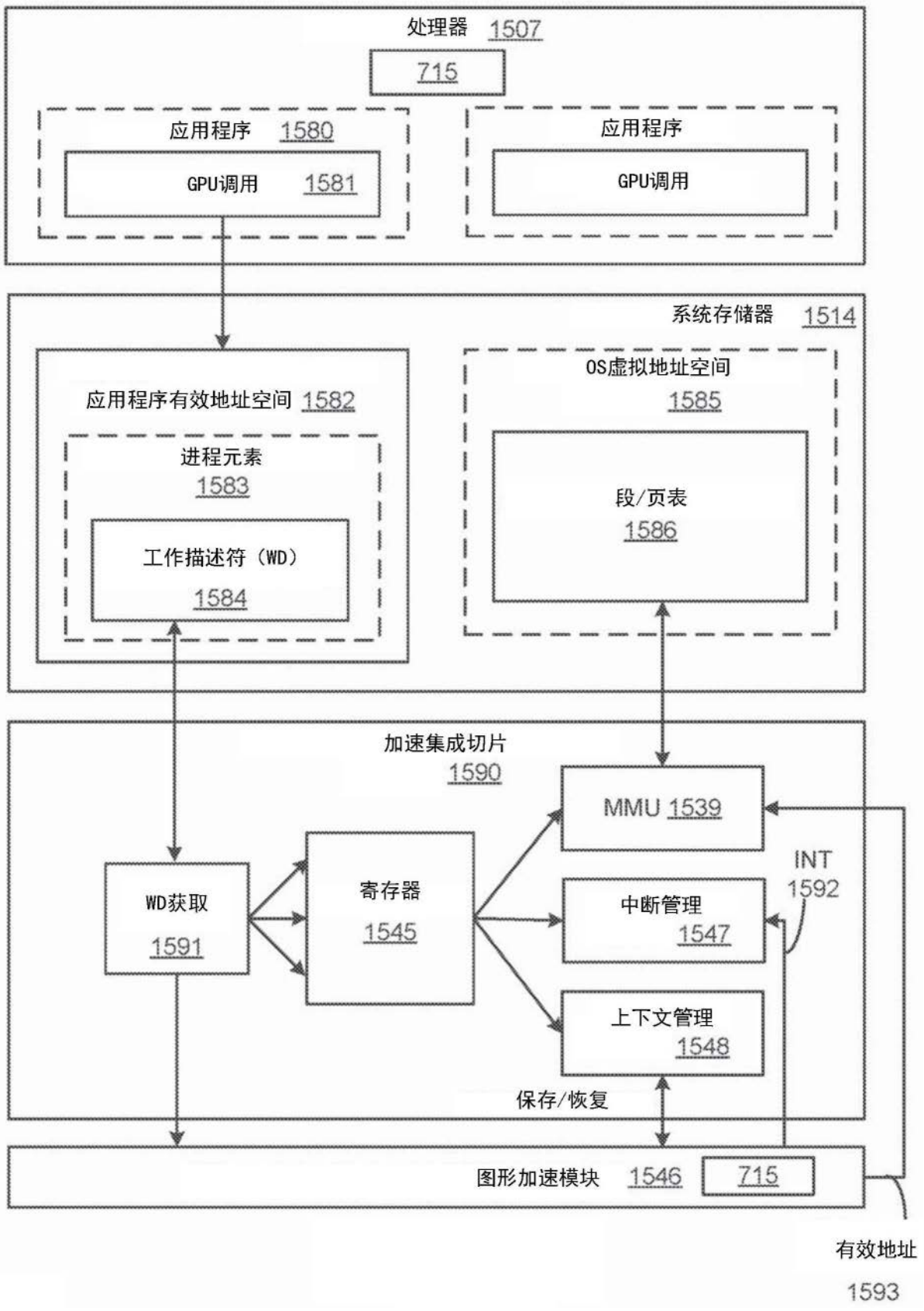

处理器 1507
715
应用程序 1580
GPU调用 1581
应用程序
GPU调用
系统存储器 1514
应用程序有效地址空间 1582
进程元素
1583
工作描述符（WD）
1584
OS虚拟地址空间
1585
段/页表
1586
加速集成切片
1590
MMU 1539
INT
1592
WD获取
1591
寄存器
1545
中断管理
1547
上下文管理
1548
保存/恢复
图形加速模块 1546
715
有效地址
1593

图15D

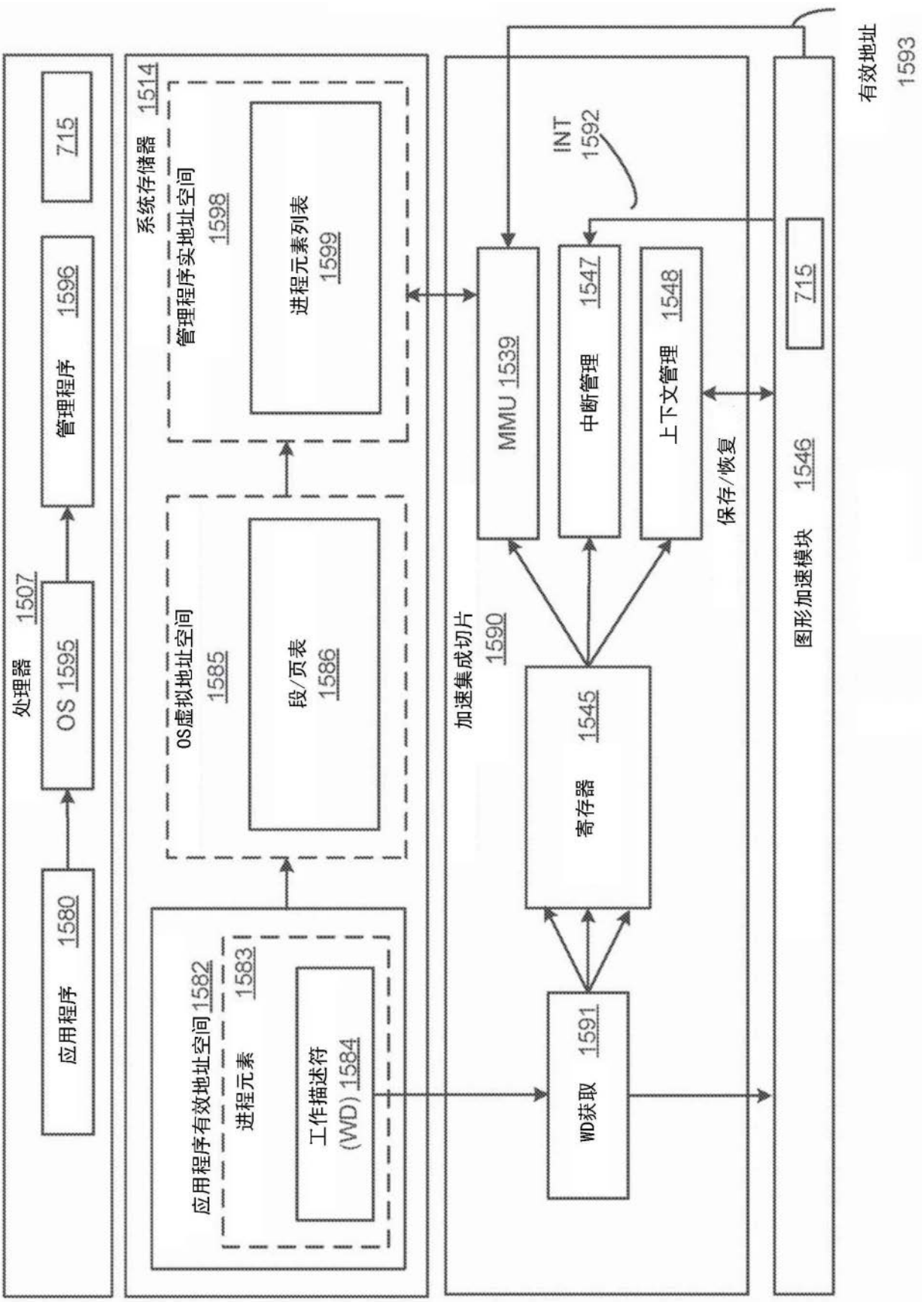

处理器 1507
应用程序 1580
OS 1595
管理程序 1596
715
系统存储器 1514
管理程序实地址空间 1598
进程元素列表 1599
OS虚拟地址空间 1585
段/页表 1586
应用程序有效地址空间 1582
进程元素 1583
工作描述符 (WD) 1584
加速集成切片 1590
MMU 1539
中断管理 1547
上下文管理 1548
INT 1592
寄存器 1545
WD获取 1591
保存/恢复
图形加速模块 1546
715
有效地址 1593

图15E

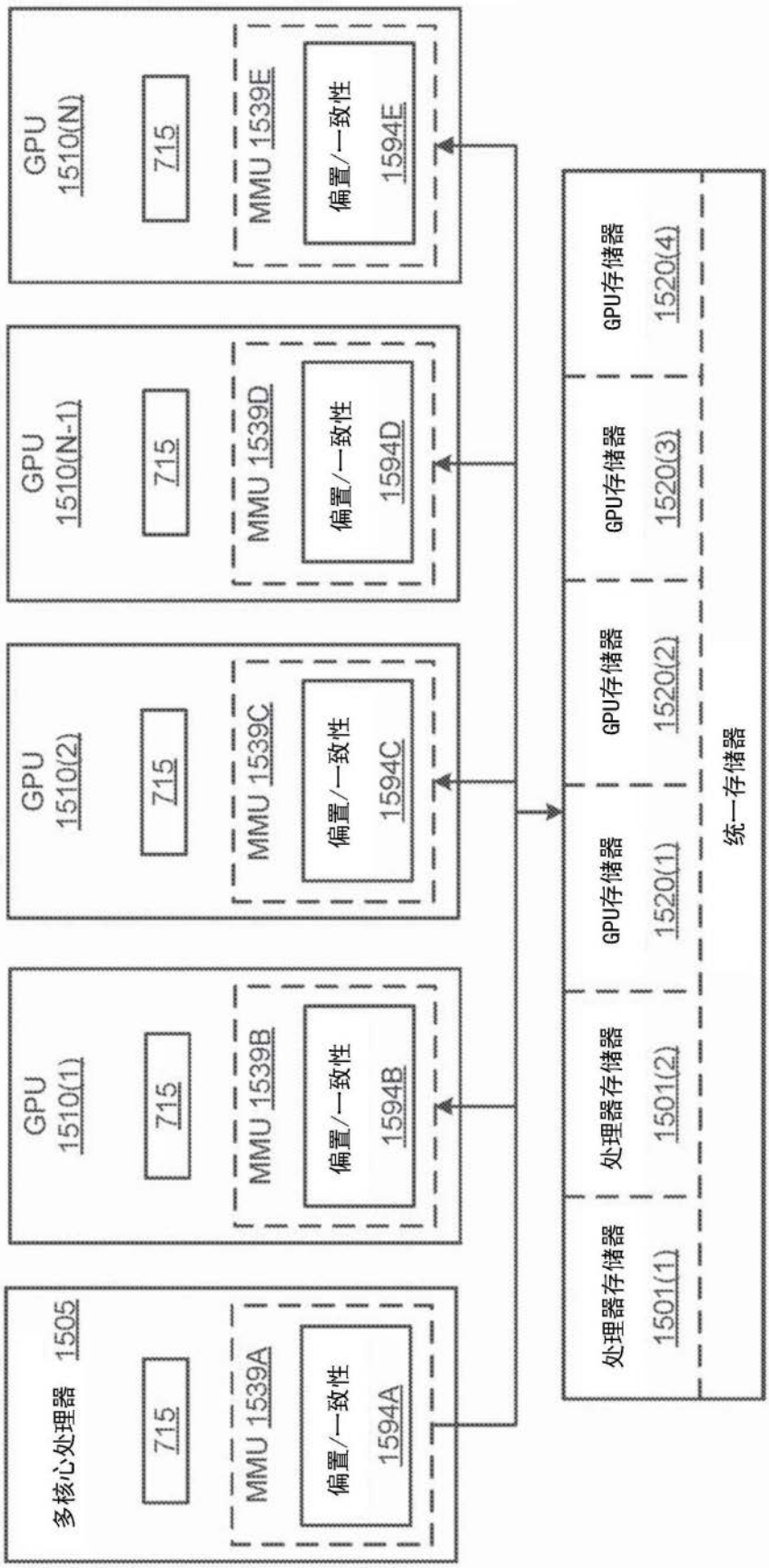

多核心处理器 1505
Z15
MMU 1539A
偏置/一致性 1594A
GPU 1510(1)
Z15
MMU 1539B
偏置/一致性 1594B
GPU 1510(2)
Z15
MMU 1539C
偏置/一致性 1594C
GPU 1510(N-1)
Z15
MMU 1539D
偏置/一致性 1594D
GPU 1510(N)
Z15
MMU 1539E
偏置/一致性 1594E
处理器存储器 1501(1)
处理器存储器 1501(2)
GPU存储器 1520(1)
GPU存储器 1520(2)
GPU存储器 1520(3)
GPU存储器 1520(4)
统一存储器

图15F

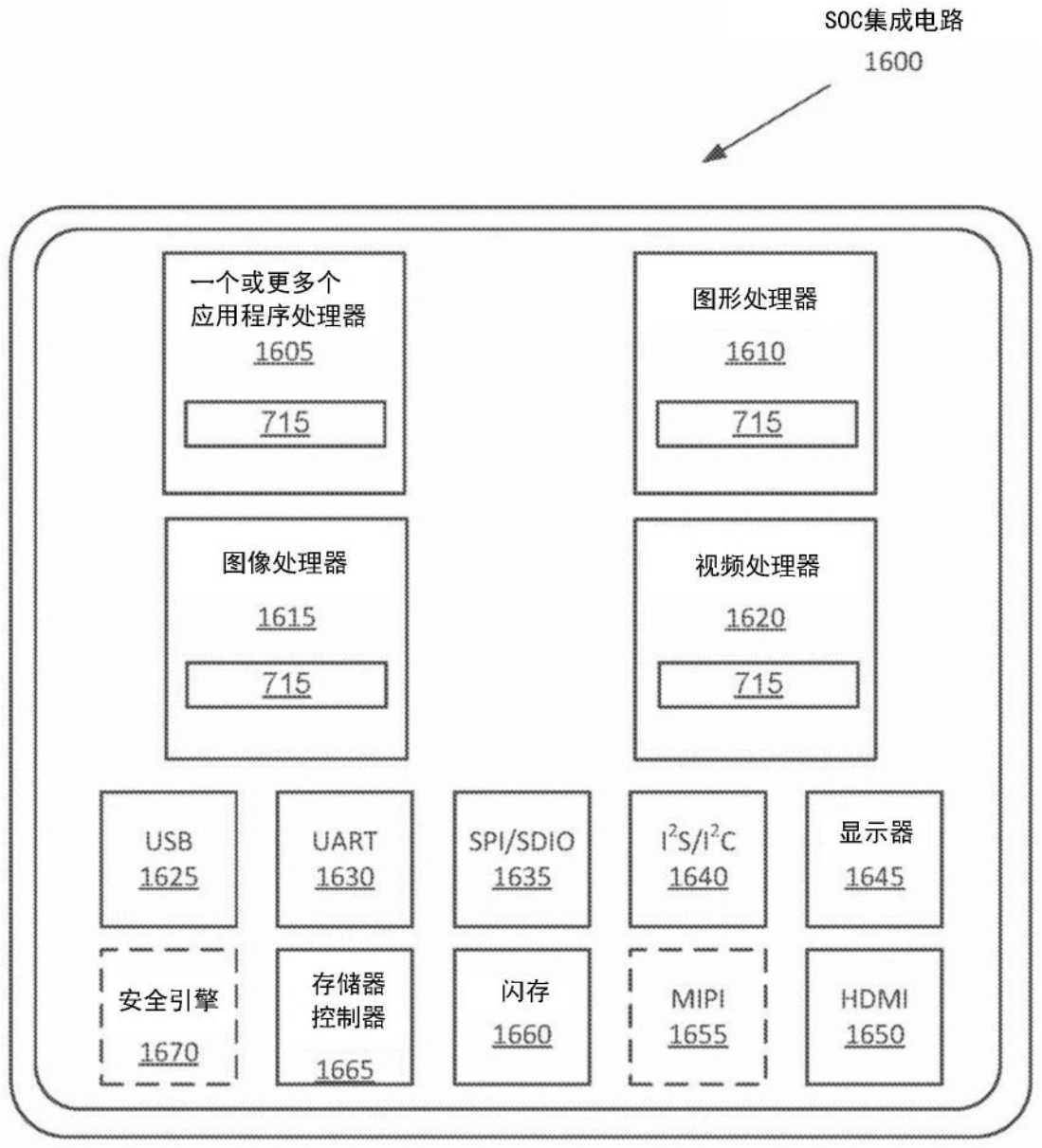

SOC集成电路
1600
一个或更多个
应用程序处理器
1605
715
图形处理器
1610
715
图像处理器
1615
715
视频处理器
1620
715
USB
1625
UART
1630
SPI/SDIO
1635
I²S/I²C
1640
显示器
1645
安全引擎
1670
存储器
控制器
1665
闪存
1660
MIPI
1655
HDMI
1650

图16

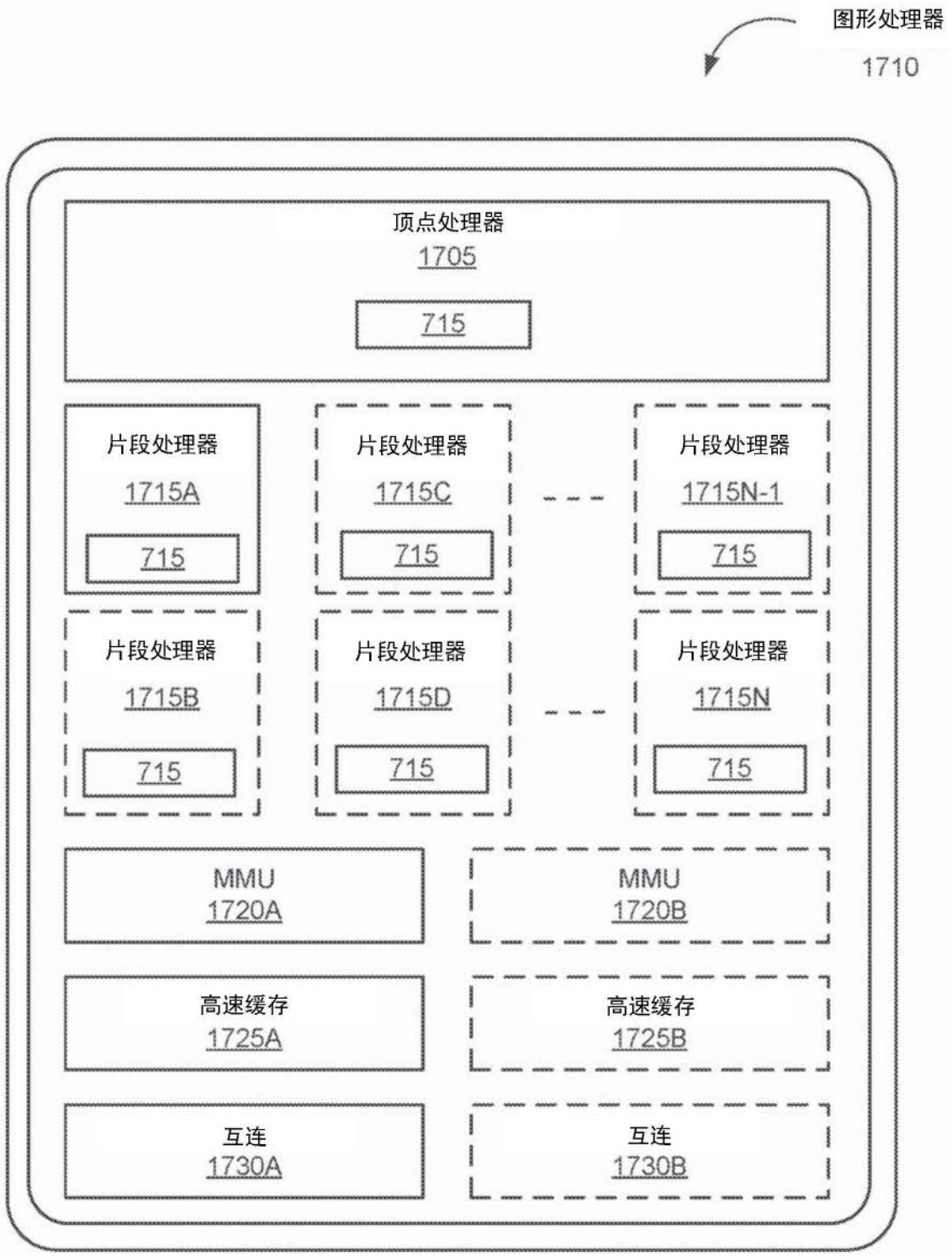

图形处理器
1710
顶点处理器
1705
715
片段处理器
1715A
715
片段处理器
1715C
715
片段处理器
1715N-1
715
片段处理器
1715B
715
片段处理器
1715D
715
片段处理器
1715N
715
MMU
1720A
MMU
1720B
高速缓存
1725A
高速缓存
1725B
互连
1730A
互连
1730B

图17A


图形处理器
1740
核心间任务管理器
（例如，线程分派器）
1745
着色器
核心
1755A
715
着色器
核心
1755C
715
着色器
核心
1755E
715
着色器
核心
1755N-1
715
着色器
核心
1755B
715
着色器
核心
1755D
715
着色器
核心
1755F
715
着色器
核心
1755N
715
分块单元 1758
MMU
1720A
MMU
1720B
高速缓存
1725A
高速缓存
1725B
互连
1730A
互连
1730B

图17B

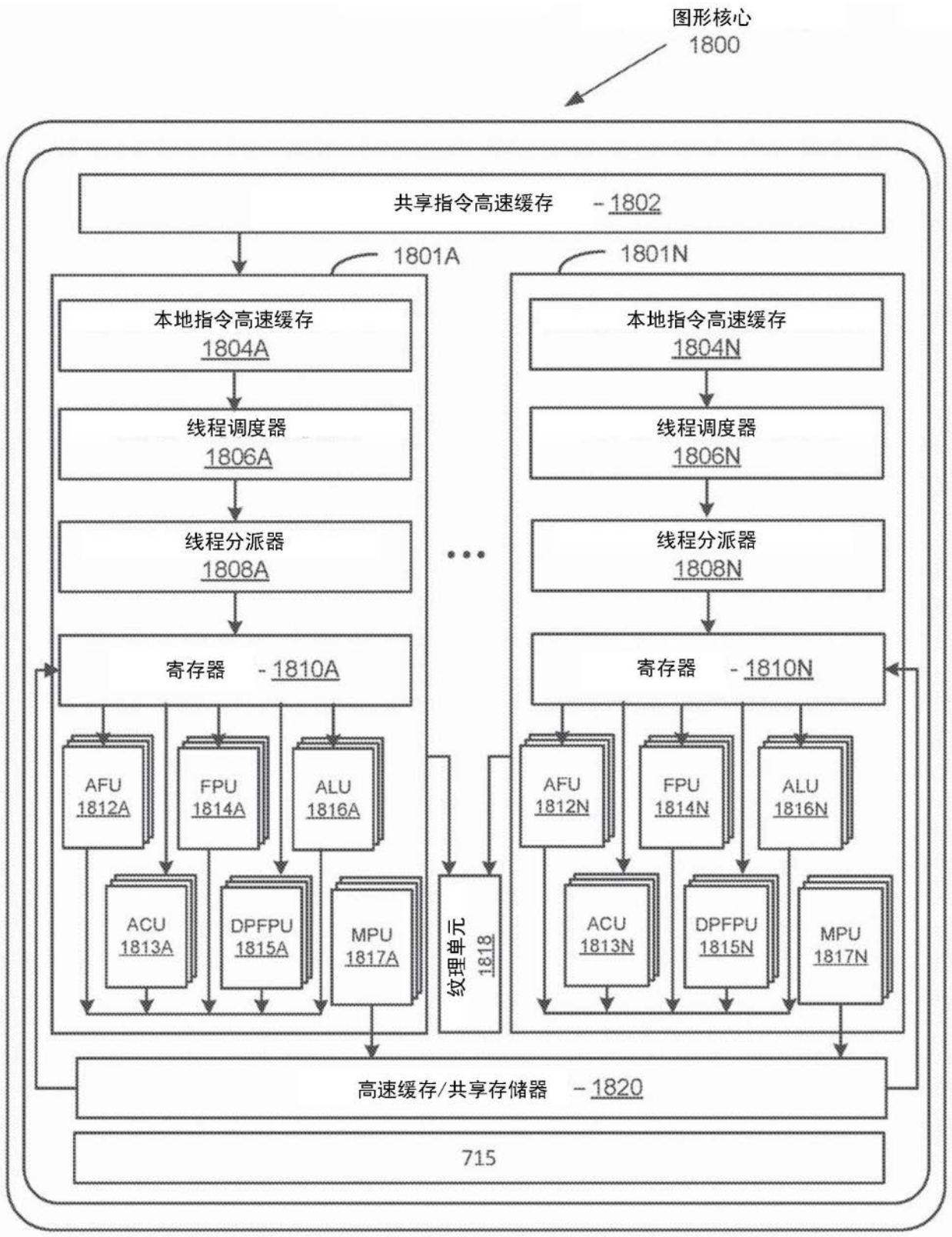

图形核心
1800
共享指令高速缓存 - 1802
1801A
1801N
本地指令高速缓存
1804A
本地指令高速缓存
1804N
线程调度器
1806A
线程调度器
1806N
线程分派器
1808A
线程分派器
1808N
寄存器 - 1810A
寄存器 - 1810N
AFU
1812A
FPU
1814A
ALU
1816A
AFU
1812N
FPU
1814N
ALU
1816N
ACU
1813A
DPFPU
1815A
MPU
1817A
纹理单元
1818
ACU
1813N
DPFPU
1815N
MPU
1817N
高速缓存/共享存储器 - 1820
715

图18A

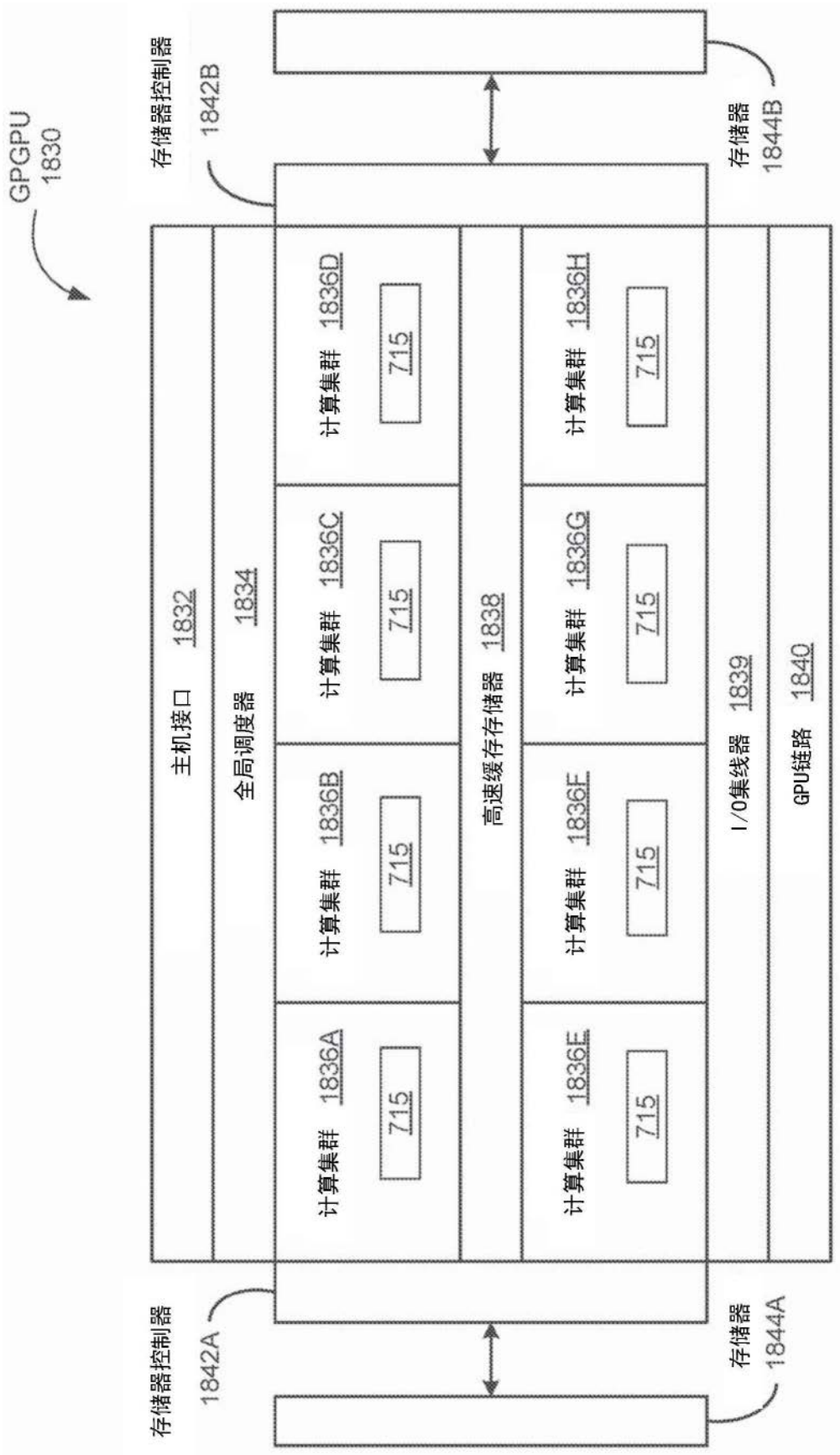

GPGPU
1830
存储器控制器
1842B
存储器
1844B
主机接口 1832
全局调度器 1834
计算集群 1836D
计算集群 1836C
计算集群 1836B
计算集群 1836A
计算集群 1836H
计算集群 1836G
计算集群 1836F
计算集群 1836E
715
高速缓存存储器 1838
I/O集线器 1839
GPU链路 1840
存储器控制器
1842A
存储器
1844A

图18B

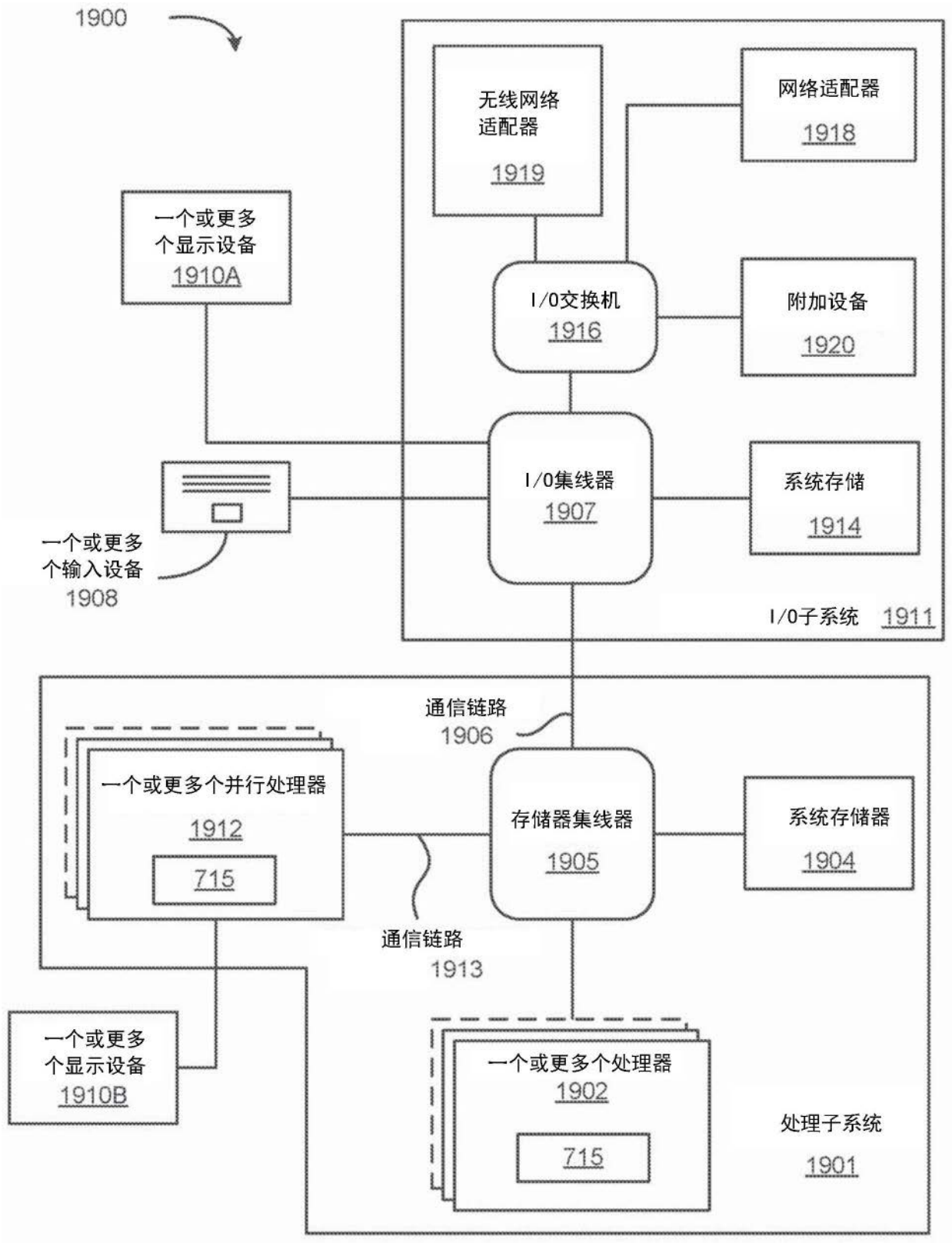

1900
无线网络
适配器
1919
网络适配器
1918
一个或更多
个显示设备
1910A
I/O交换机
1916
附加设备
1920
I/O集线器
1907
系统存储
1914
一个或更多
个输入设备
1908
I/O子系统 1911
通信链路
1906
一个或更多个并行处理器
1912
715
存储器集线器
1905
系统存储器
1904
通信链路
1913
一个或更多
个显示设备
1910B
一个或更多个处理器
1902
715
处理子系统
1901

图19

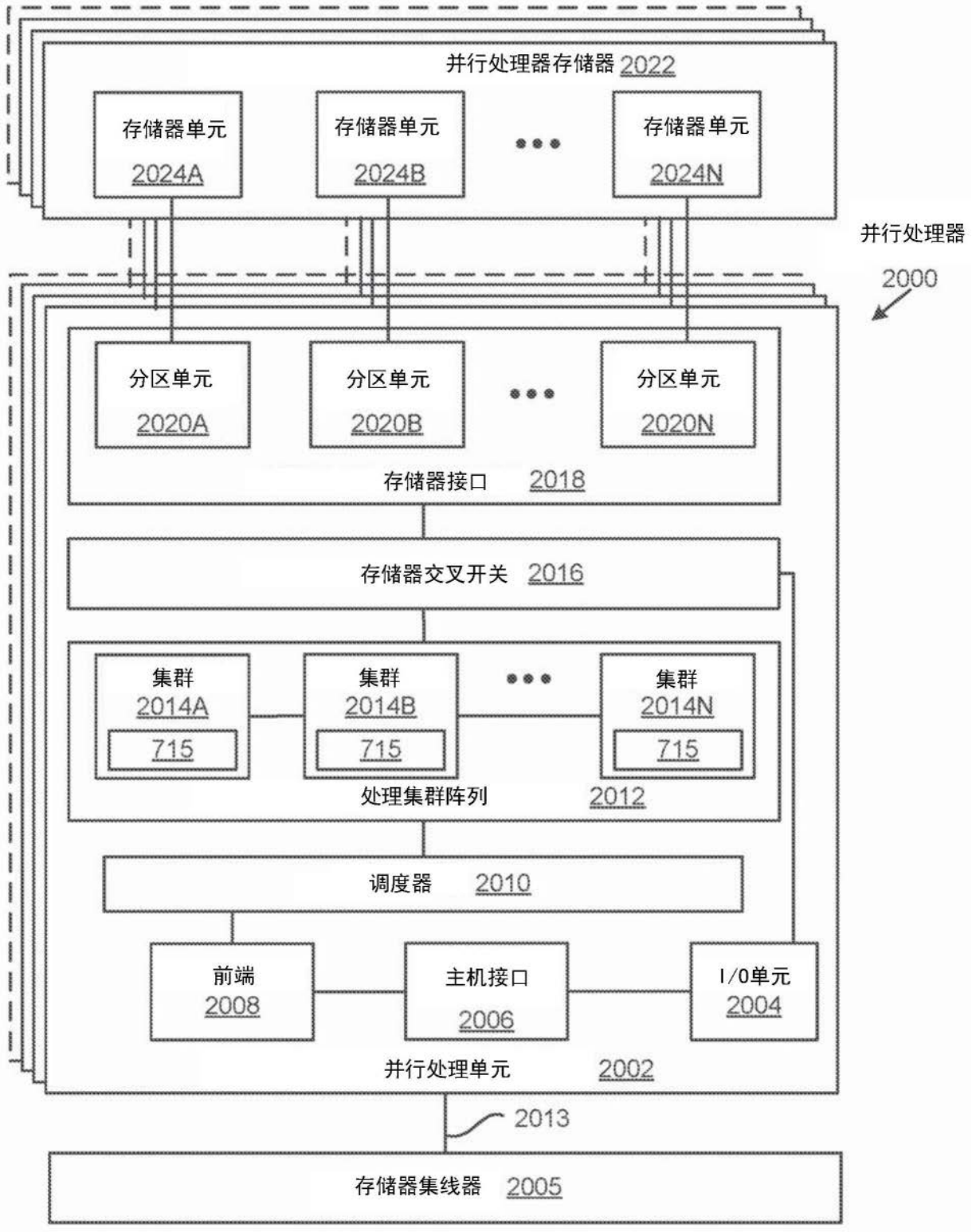

并行处理器存储器 2022
存储器单元 2024A
存储器单元 2024B
存储器单元 2024N
并行处理器 2000
分区单元 2020A
分区单元 2020B
分区单元 2020N
存储器接口 2018
存储器交叉开关 2016
集群 2014A
715
集群 2014B
715
集群 2014N
715
处理集群阵列 2012
调度器 2010
前端 2008
主机接口 2006
I/O单元 2004
并行处理单元 2002
2013
存储器集线器 2005

图20A

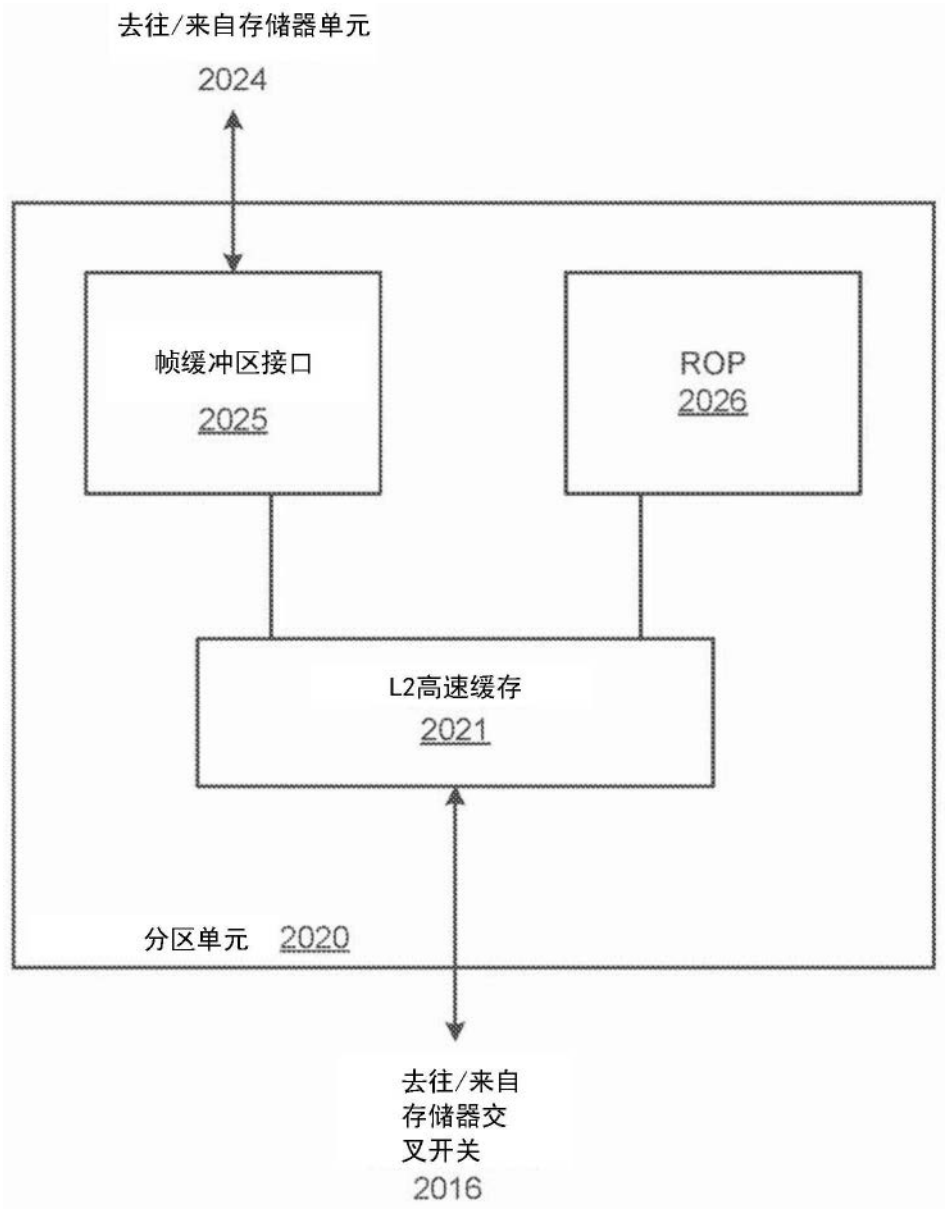

去往/来自存储器单元
2024
帧缓冲区接口
2025
ROP
2026
L2高速缓存
2021
分区单元 2020
去往/来自
存储器交
叉开关
2016

图20B

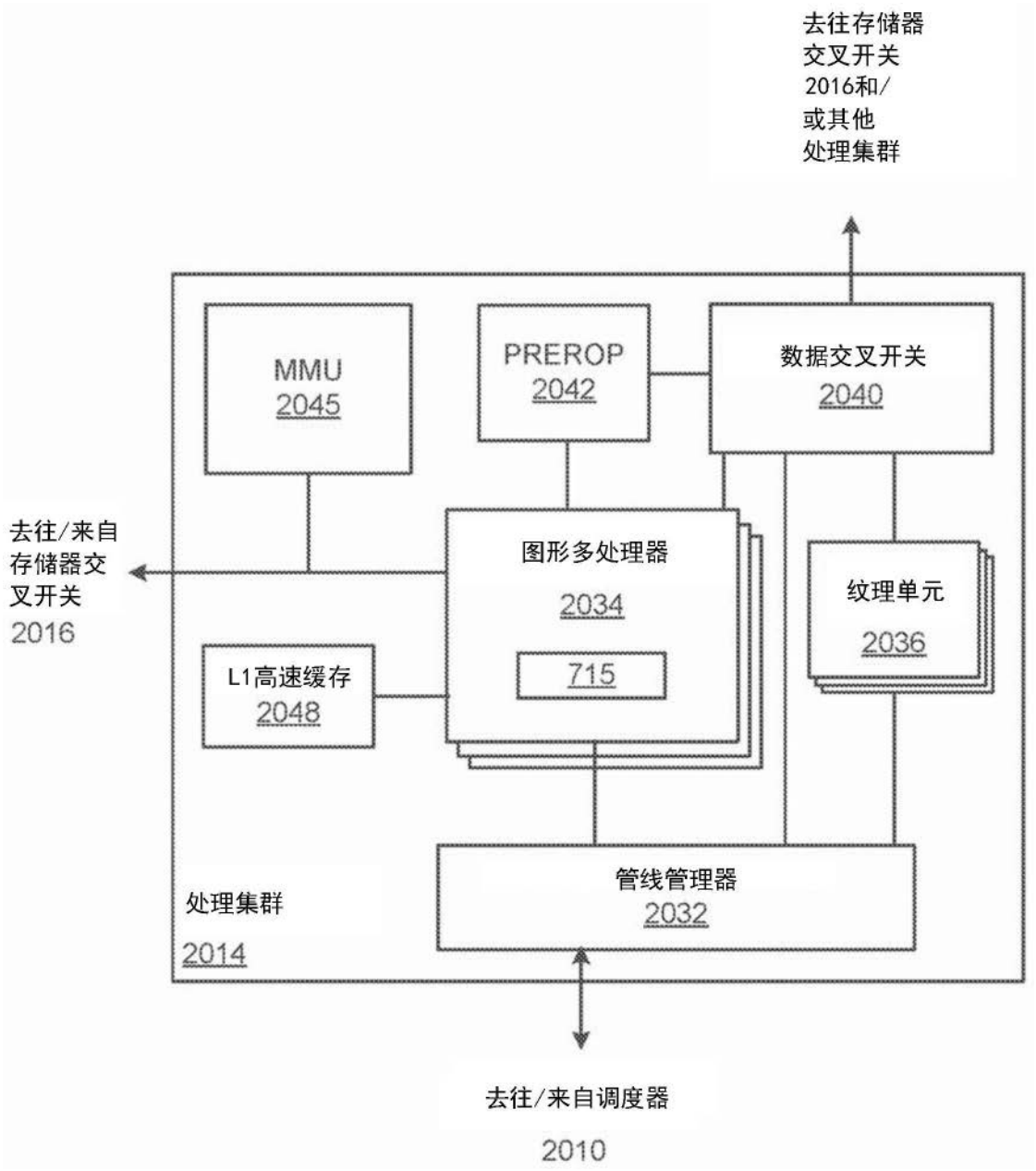

去往存储器
交叉开关
2016和/
或其他
处理集群
MMU
2045
PREROP
2042
数据交叉开关
2040
去往/来自
存储器交
叉开关
2016
图形多处理器
2034
715
纹理单元
2036
L1高速缓存
2048
管线管理器
2032
处理集群
2014
去往/来自调度器
2010

图20C

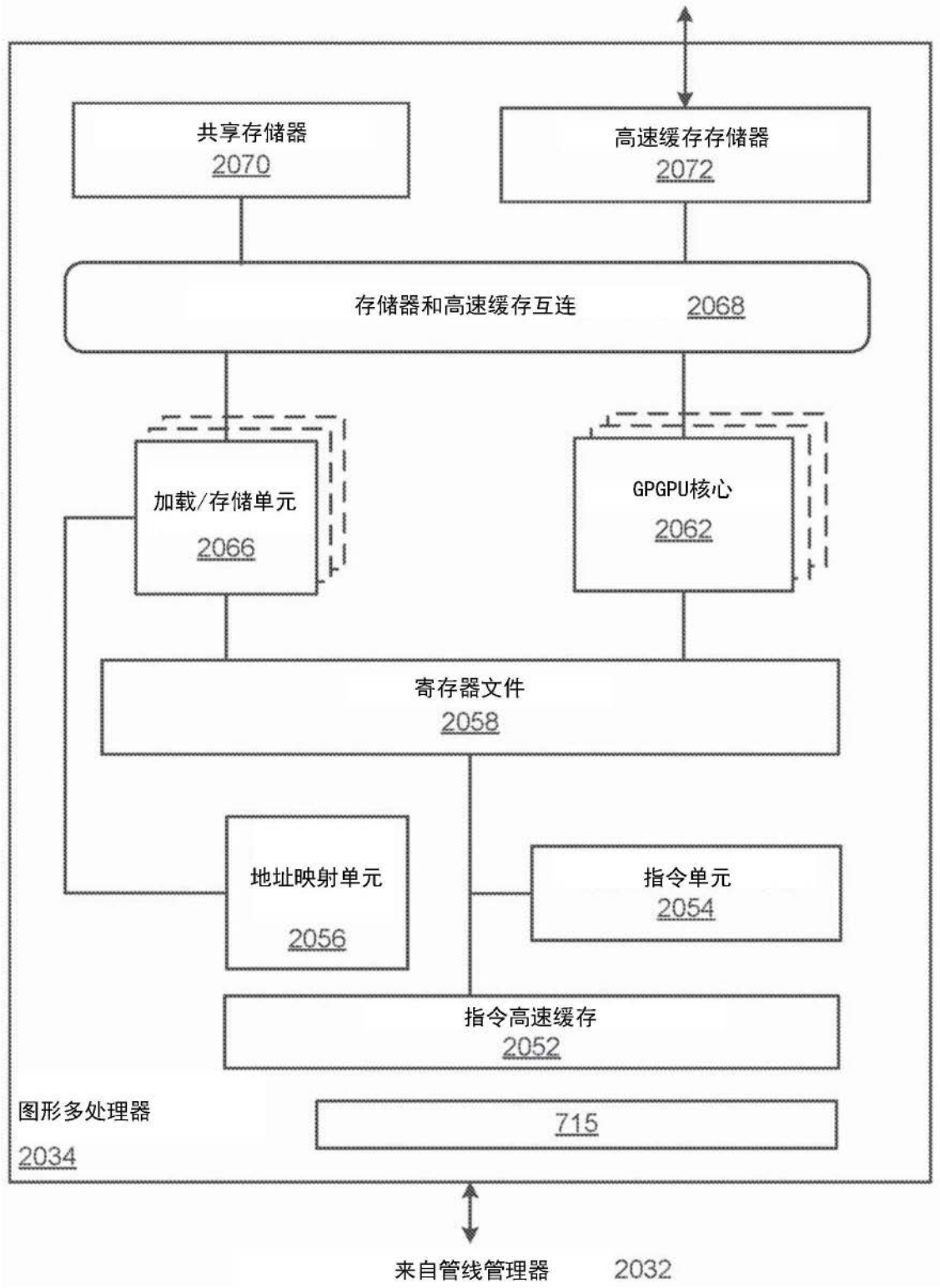

共享存储器
2070
高速缓存存储器
2072
存储器和高速缓存互连
2068
加载/存储单元
2066
GPGPU核心
2062
寄存器文件
2058
地址映射单元
2056
指令单元
2054
指令高速缓存
2052
图形多处理器
2034
715
来自管线管理器
2032

图20D

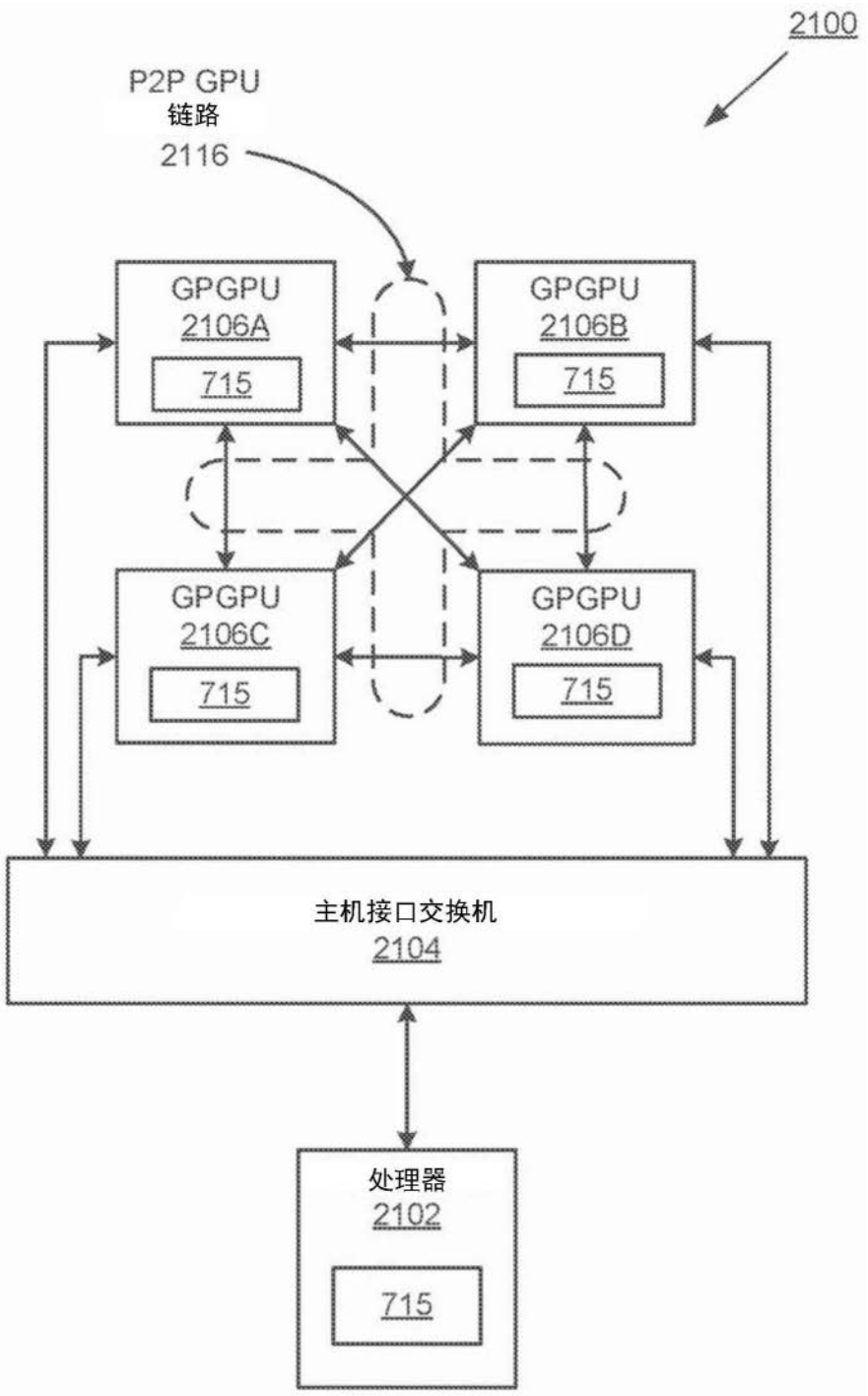

2100
P2P GPU
链路
2116
GPGPU
2106A
715
GPGPU
2106B
715
GPGPU
2106C
715
GPGPU
2106D
715
主机接口交换机
2104
处理器
2102
715

图21

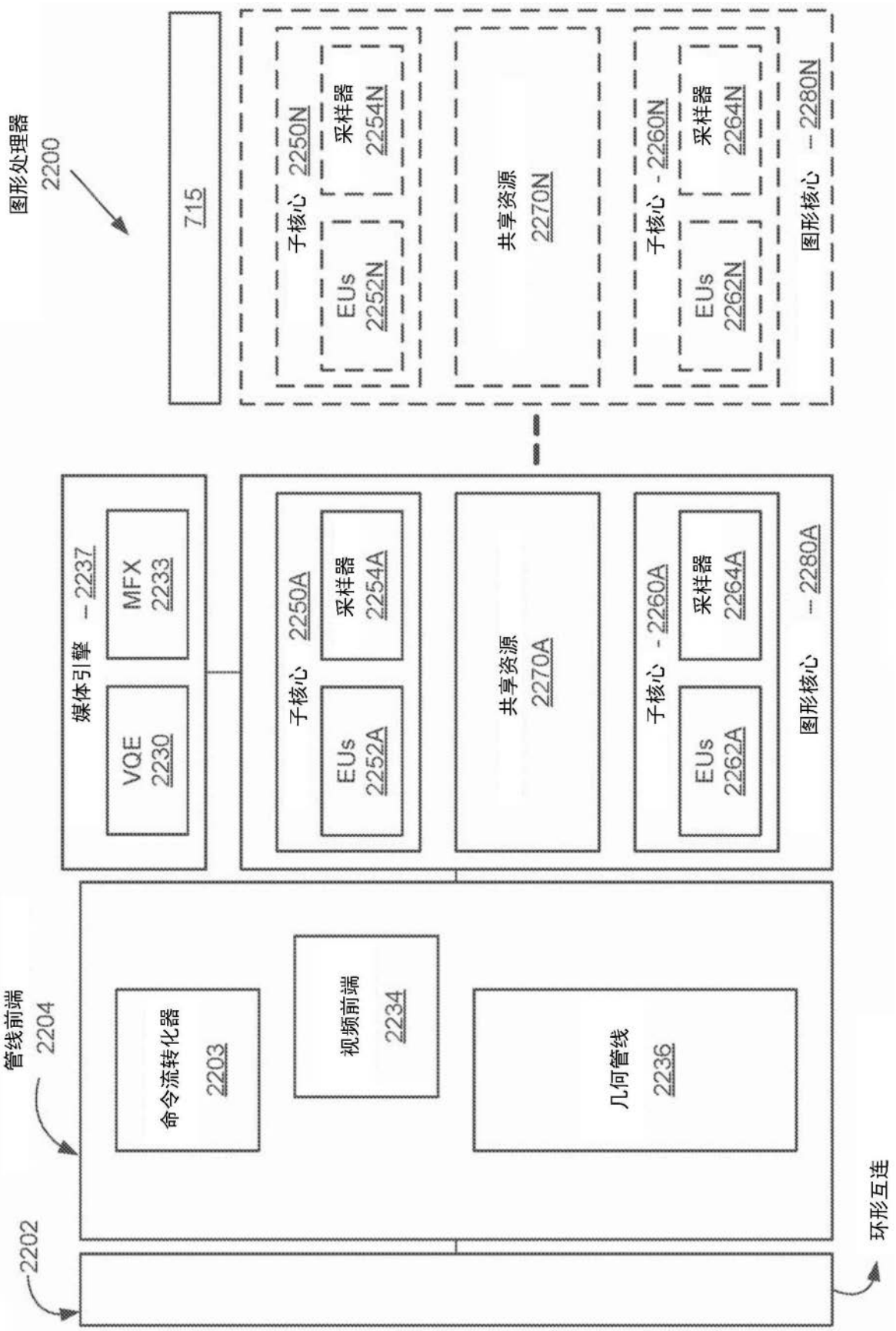

图形处理器
2200
715
子核心 2250N
EUs 2252N
采样器 2254N
共享资源 2270N
子核心 2260N
EUs 2262N
采样器 2264N
图形核心 2280N
媒体引擎 2237
VQE 2230
MFX 2233
子核心 2250A
EUs 2252A
采样器 2254A
共享资源 2270A
子核心 2260A
EUs 2262A
采样器 2264A
图形核心 2280A
管线前端 2204
命令流转化器 2203
视频前端 2234
几何管线 2236
2202
环形互连

图22

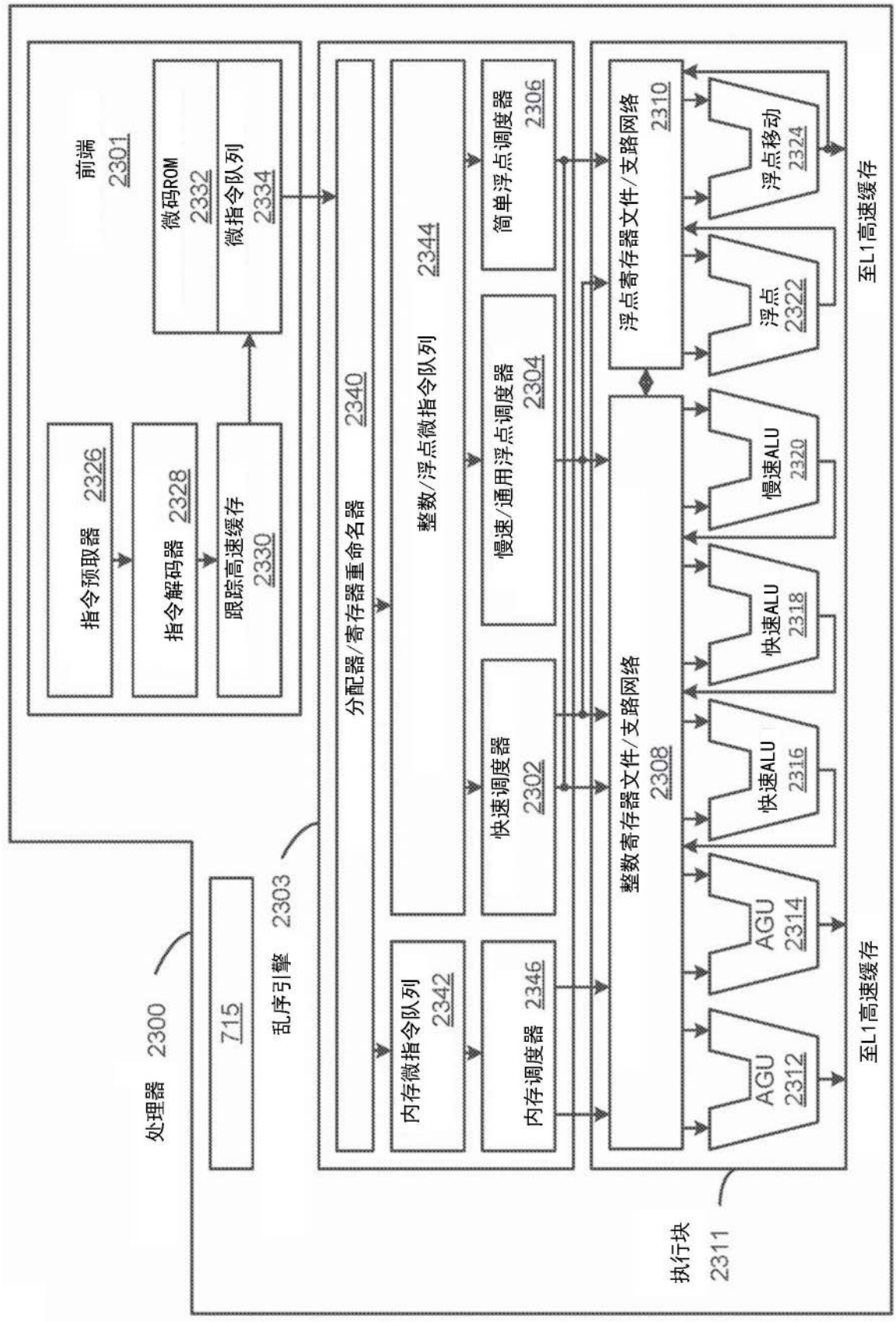

处理器 2300
715
乱序引擎 2303
前端 2301
指令预取器 2326
指令解码器 2328
跟踪高速缓存 2330
微码ROM 2332
微指令队列 2334
分配器/寄存器重命名器 2340
内存微指令队列 2342
内存调度器 2346
整数/浮点微指令队列 2344
快速调度器 2302
慢速/通用浮点调度器 2304
简单浮点调度器 2306
整数寄存器文件/支路网络 2308
浮点寄存器文件/支路网络 2310
AGU 2312
AGU 2314
快速ALU 2316
快速ALU 2318
慢速ALU 2320
浮点 2322
浮点移动 2324
执行块 2311
至L1高速缓存
至L1高速缓存

图23

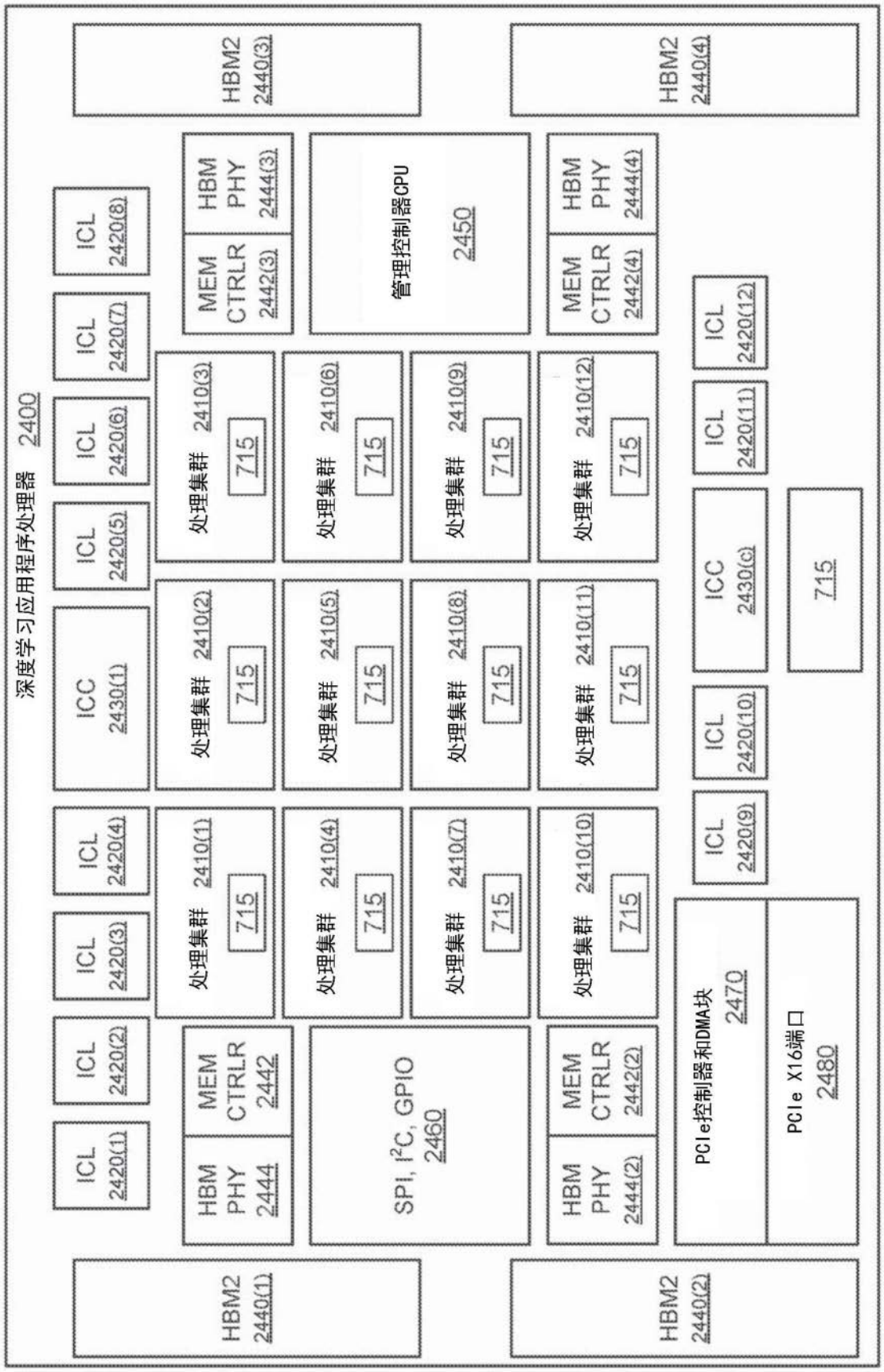

深度学习应用程序处理器 2400
HBM2 2440(1)
HBM2 2440(2)
HBM2 2440(3)
HBM2 2440(4)
HBM PHY 2444
MEM CTRLR 2442
HBM PHY 2444(2)
MEM CTRLR 2442(2)
HBM PHY 2444(3)
MEM CTRLR 2442(3)
HBM PHY 2444(4)
MEM CTRLR 2442(4)
SPI, I2C, GPIO 2460
PCIe控制器和DMA块 2470
PCIe X16端口 2480
管理控制器CPU 2450
ICC 2430(1)
ICC 2430(2)
ICL 2420(1)
ICL 2420(2)
ICL 2420(3)
ICL 2420(4)
ICL 2420(5)
ICL 2420(6)
ICL 2420(7)
ICL 2420(8)
ICL 2420(9)
ICL 2420(10)
ICL 2420(11)
ICL 2420(12)
处理集群 2410(1)
处理集群 2410(2)
处理集群 2410(3)
处理集群 2410(4)
处理集群 2410(5)
处理集群 2410(6)
处理集群 2410(7)
处理集群 2410(8)
处理集群 2410(9)
处理集群 2410(10)
处理集群 2410(11)
处理集群 2410(12)
715

图24

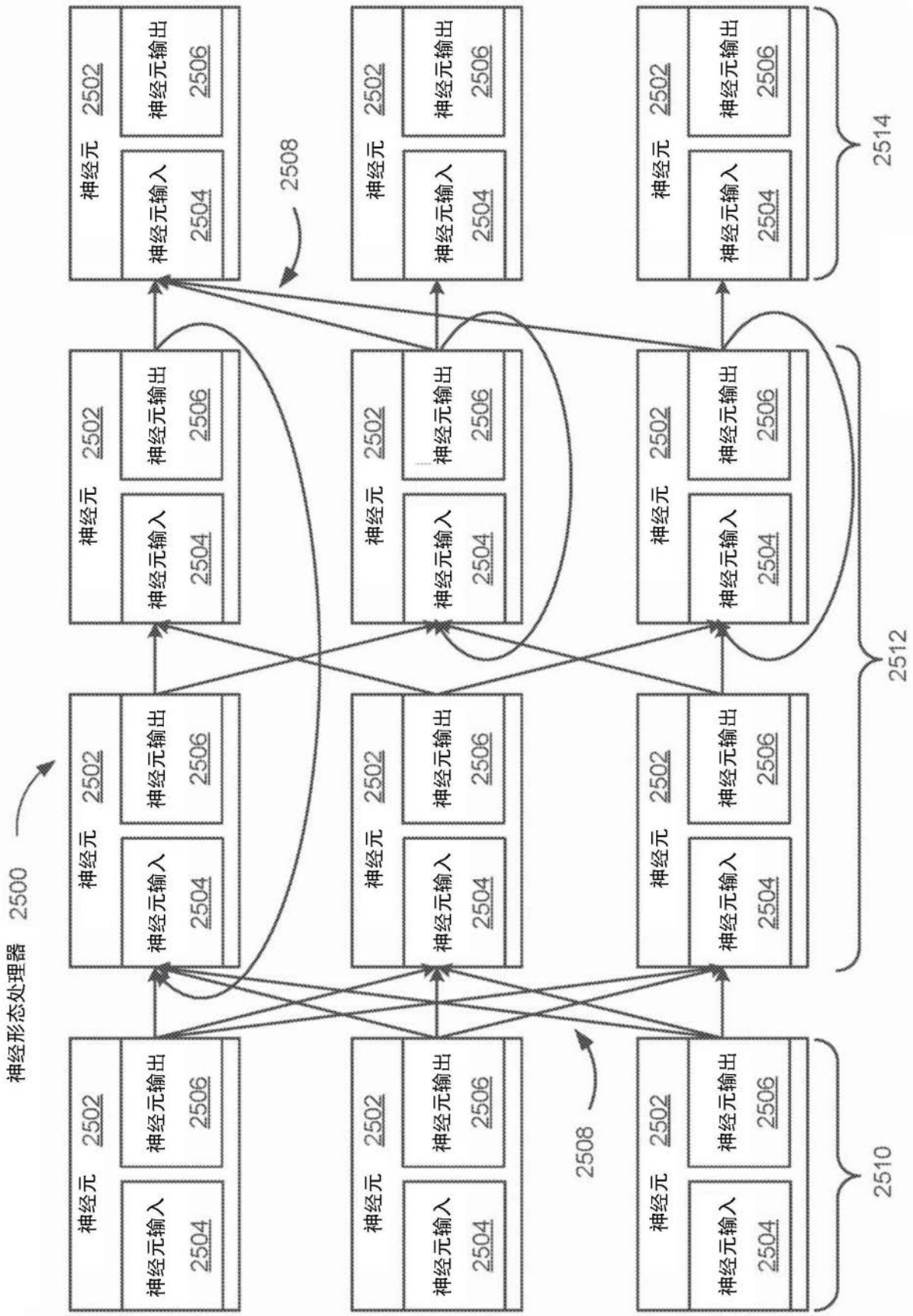

神经形态处理器 2500
神经元 2502
神经元输入 2504
神经元输出 2506
2508
2508
2510
2512
2514

图25

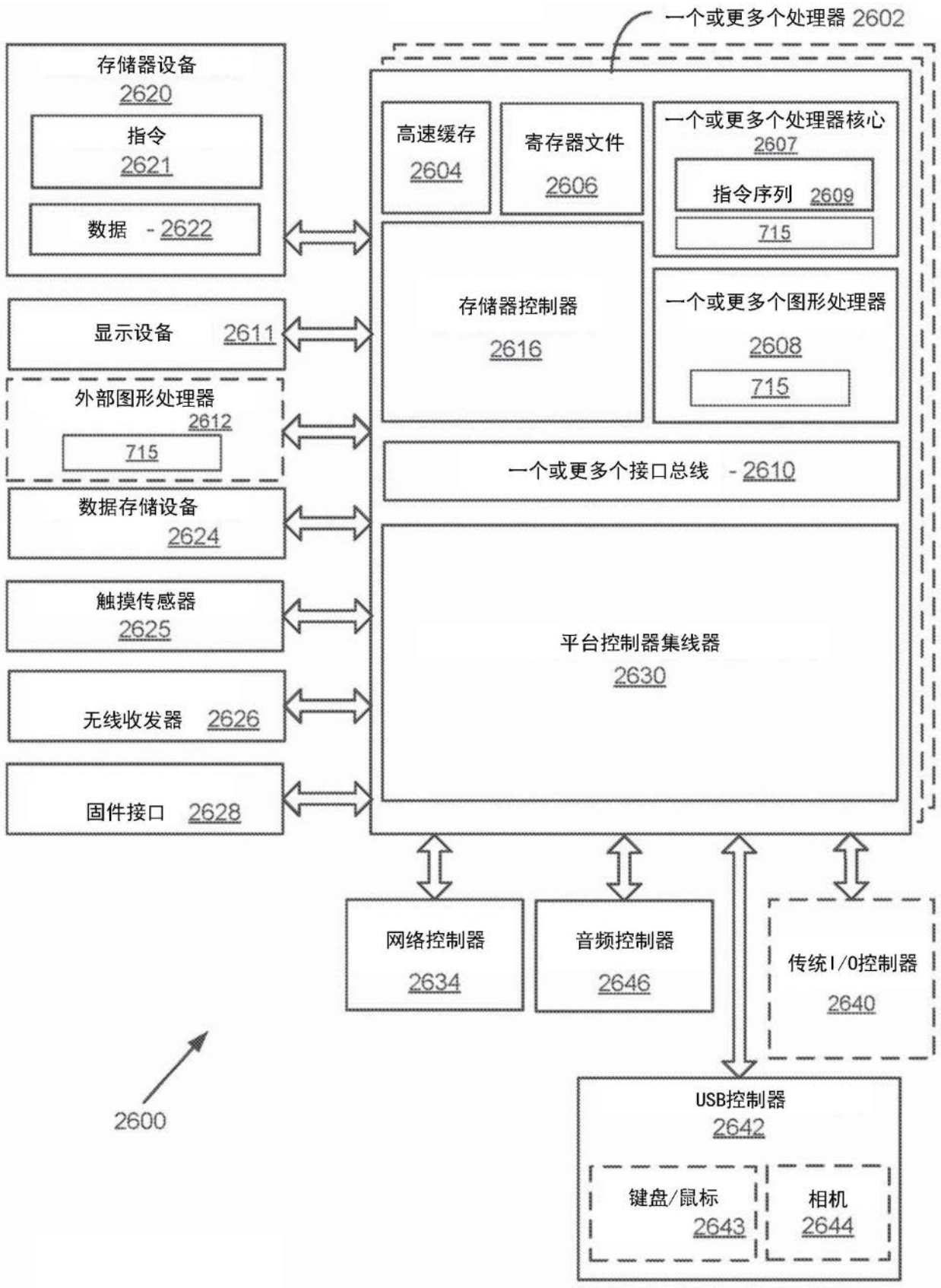

一个或更多个处理器 2602
存储器设备
2620
指令
2621
数据 - 2622
显示设备 2611
外部图形处理器
2612
715
数据存储设备
2624
触摸传感器
2625
无线收发器 2626
固件接口 2628
高速缓存
2604
寄存器文件
2606
一个或更多个处理器核心
2607
指令序列 2609
715
存储器控制器
2616
一个或更多个图形处理器
2608
715
一个或更多个接口总线 - 2610
平台控制器集线器
2630
网络控制器
2634
音频控制器
2646
传统I/O控制器
2640
USB控制器
2642
键盘/鼠标
2643
相机
2644
2600

图26

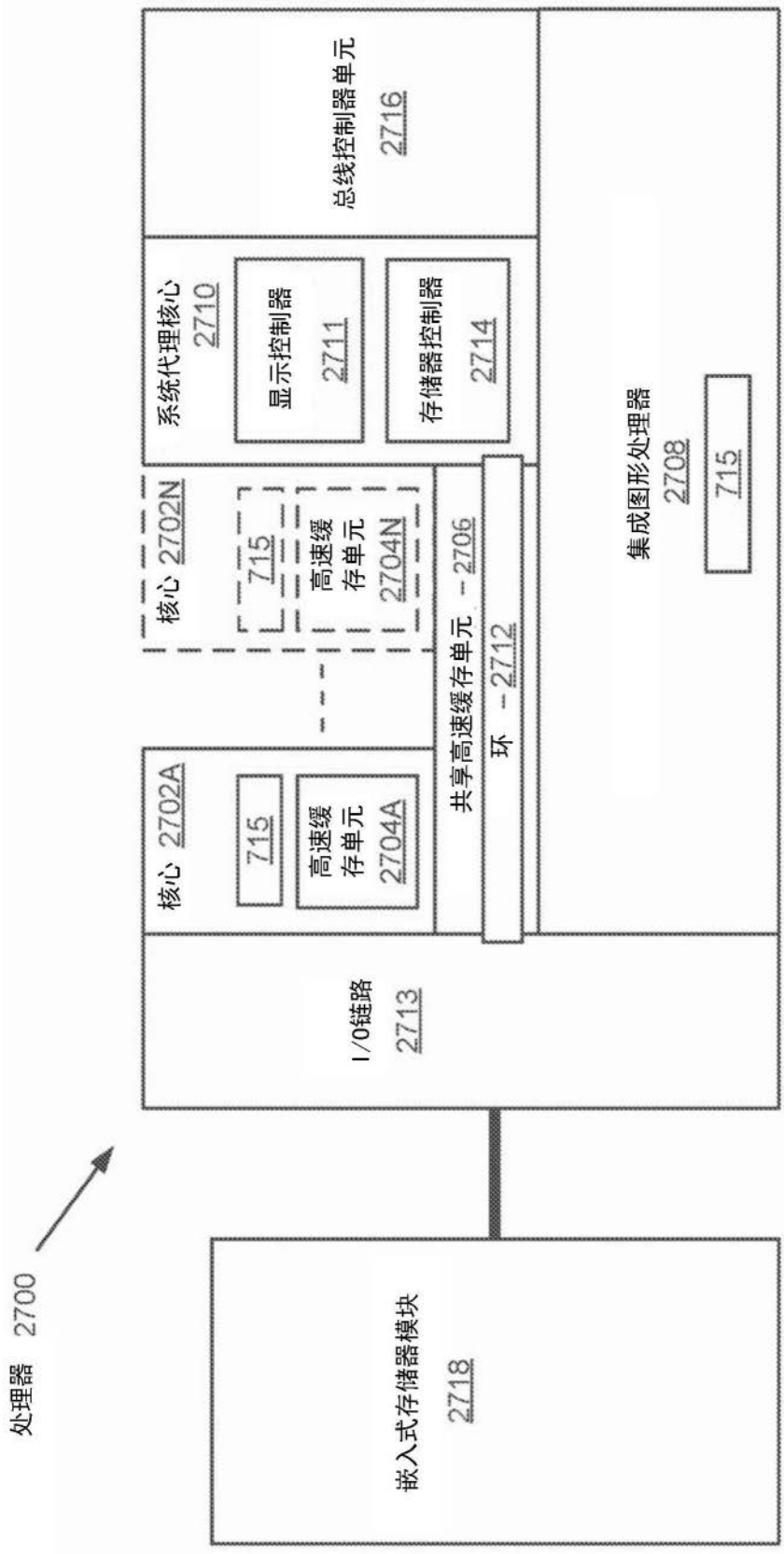

处理器 2700
嵌入式存储器模块 2718
I/O链路 2713
核心 2702A
715
高速缓存单元 2704A
核心 2702N
715
高速缓存单元 2704N
共享高速缓存单元 2706
环 2712
系统代理核心 2710
显示控制器 2711
存储器控制器 2714
总线控制器单元 2716
集成图形处理器 2708
715

图27

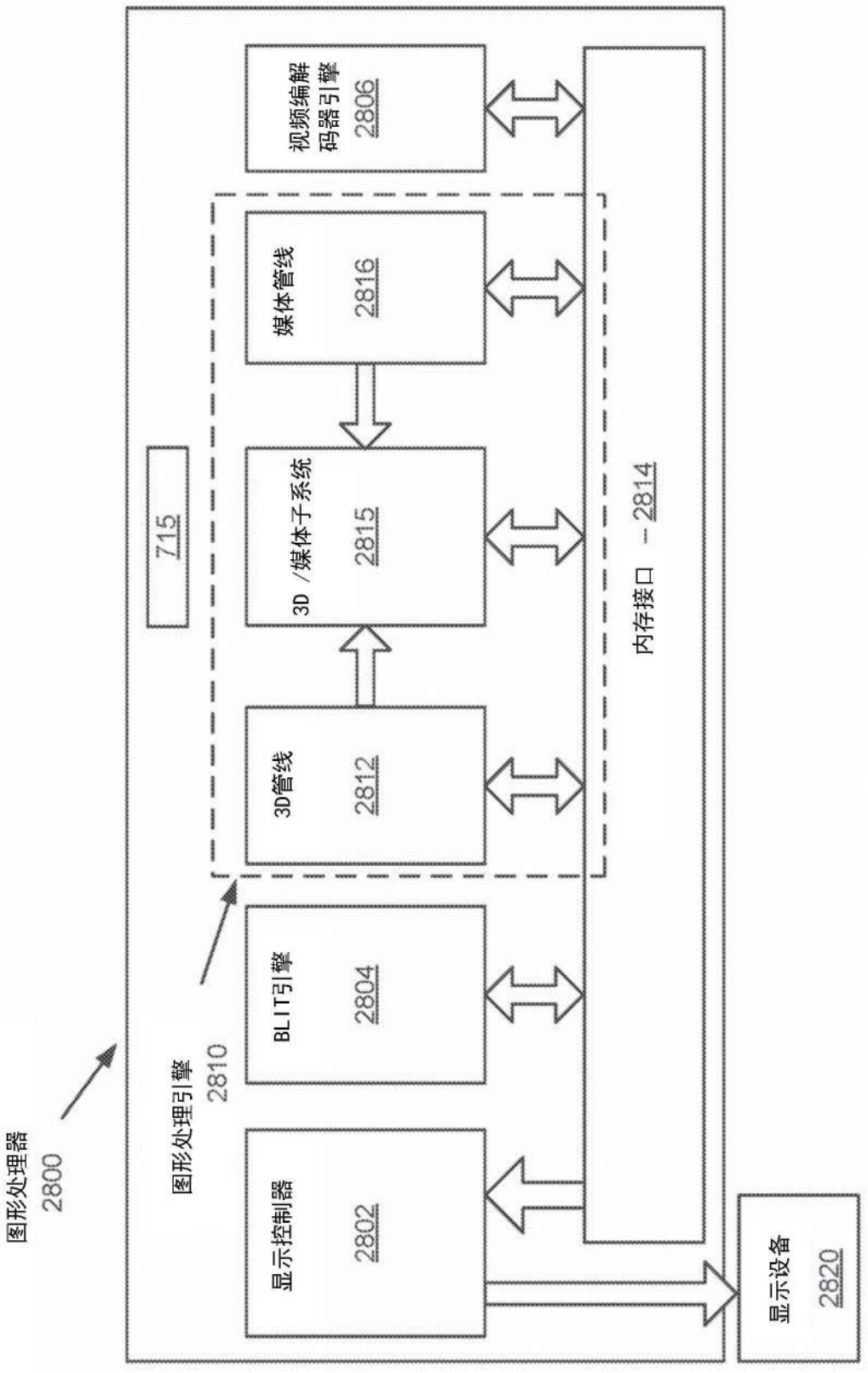

图形处理器
2800
图形处理引擎
2810
显示控制器
2802
BLIT引擎
2804
3D管线
2812
3D /媒体子系统
2815
媒体管线
2816
视频编解码器引擎
2806
715
内存接口
2814
显示设备
2820

图28

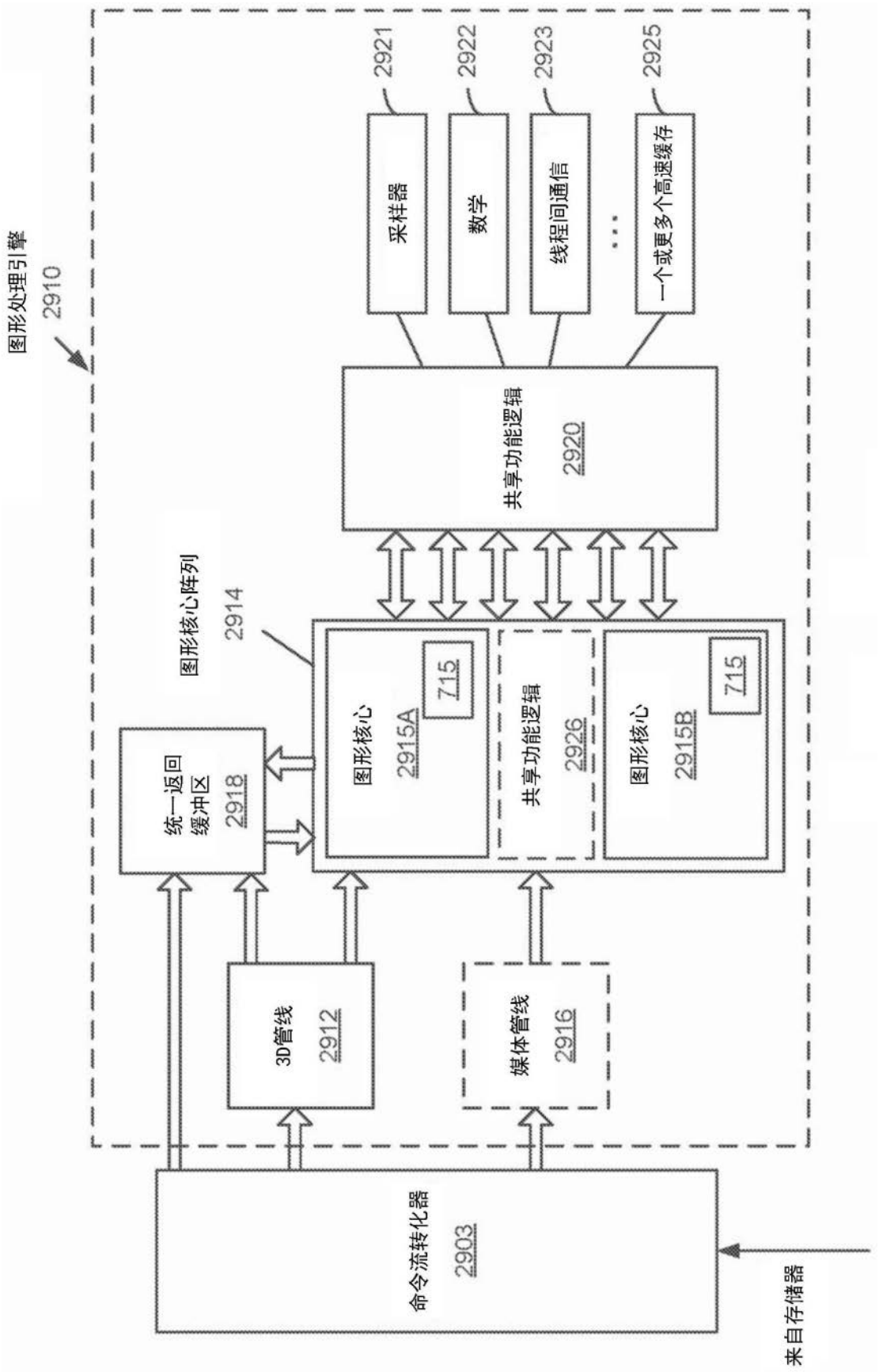

图形处理引擎
2910
采样器
2921
数学
2922
线程间通信
2923
...
一个或更多个高速缓存
2925
共享功能逻辑
2920
图形核心阵列
2914
图形核心
2915A
715
共享功能逻辑
2926
图形核心
2915B
715
统一返回
缓冲区
2918
3D管线
2912
媒体管线
2916
命令流转化器
2903
来自存储器

图29

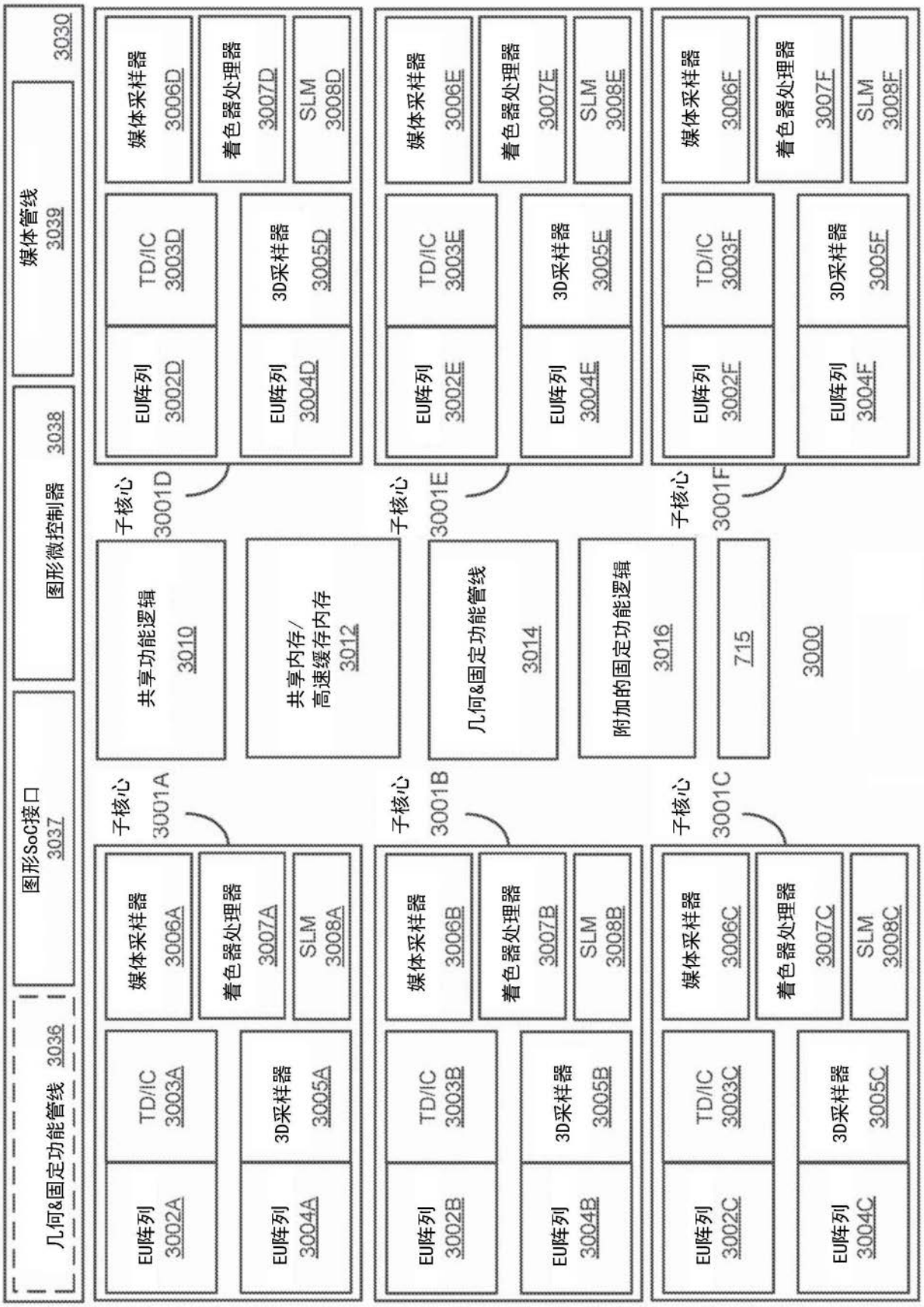

几何&固定功能管线 3036
图形SoC接口 3037
图形微控制器 3038
媒体管线 3039
3030
子核心 3001A
子核心 3001B
子核心 3001C
子核心 3001D
子核心 3001E
子核心 3001F
共享功能逻辑 3010
共享内存/高速缓存内存 3012
几何&固定功能管线 3014
附加的固定功能逻辑 3016
Z15
3000
媒体采样器 3006A
着色器处理器 3007A
SLM 3008A
TD/IC 3003A
3D采样器 3005A
EU阵列 3002A
EU阵列 3004A
媒体采样器 3006B
着色器处理器 3007B
SLM 3008B
TD/IC 3003B
3D采样器 3005B
EU阵列 3002B
EU阵列 3004B
媒体采样器 3006C
着色器处理器 3007C
SLM 3008C
TD/IC 3003C
3D采样器 3005C
EU阵列 3002C
EU阵列 3004C
媒体采样器 3006D
着色器处理器 3007D
SLM 3008D
TD/IC 3003D
3D采样器 3005D
EU阵列 3002D
EU阵列 3004D
媒体采样器 3006E
着色器处理器 3007E
SLM 3008E
TD/IC 3003E
3D采样器 3005E
EU阵列 3002E
EU阵列 3004E
媒体采样器 3006F
着色器处理器 3007F
SLM 3008F
TD/IC 3003F
3D采样器 3005F
EU阵列 3002F
EU阵列 3004F

图30

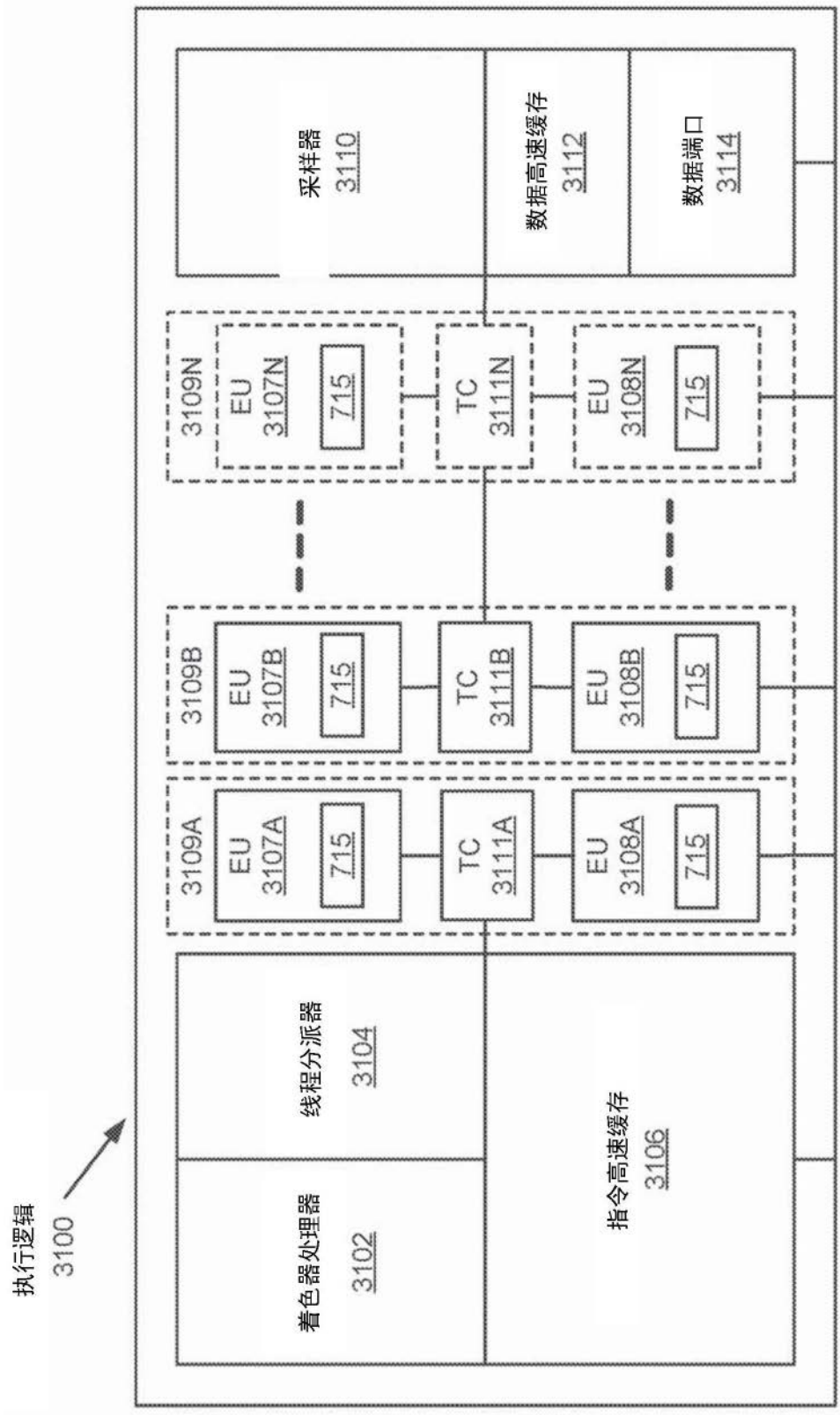

执行逻辑
3100
着色器处理器
3102
线程分派器
3104
指令高速缓存
3106
采样器
3110
数据高速缓存
3112
数据端口
3114
3109A
EU
3107A
Z15
TC
3111A
EU
3108A
Z15
3109B
EU
3107B
Z15
TC
3111B
EU
3108B
Z15
3109N
EU
3107N
Z15
TC
3111N
EU
3108N
Z15

图31A

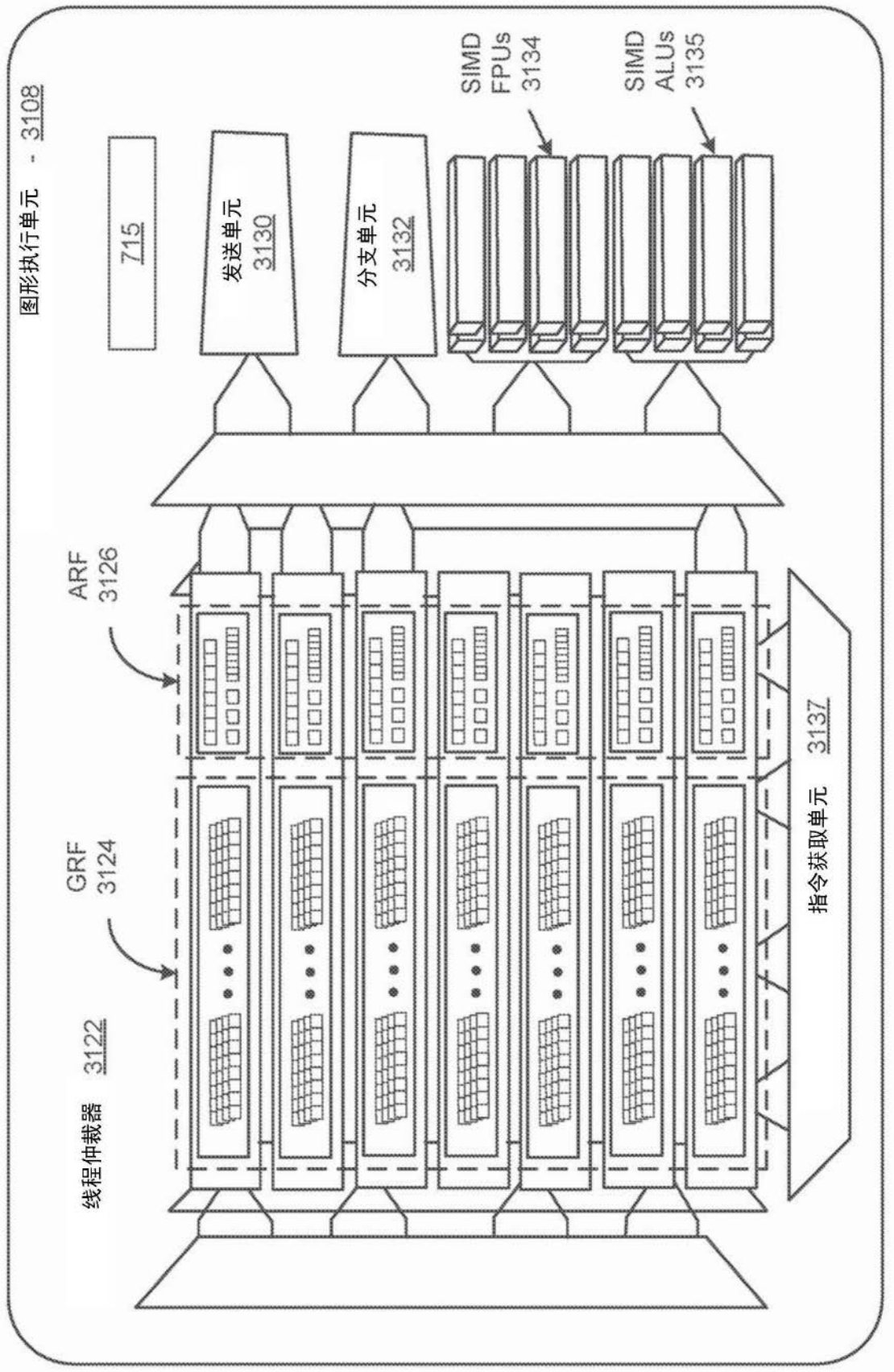

图形执行单元 - 3108
715
发送单元 3130
分支单元 3132
SIMD FPUs 3134
SIMD ALUs 3135
ARF 3126
GRF 3124
线程仲裁器 3122
指令获取单元 3137

图31B

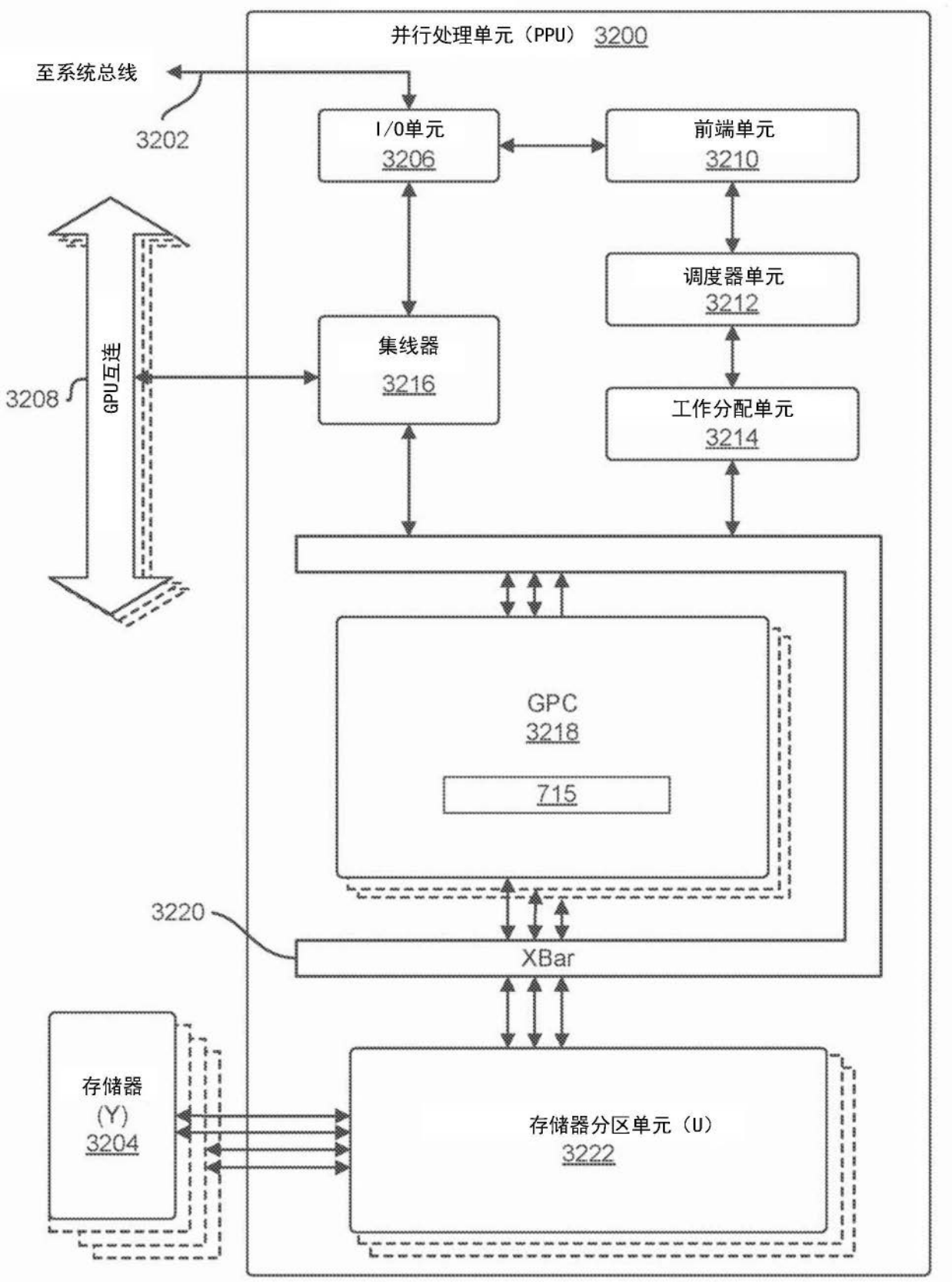

并行处理单元（PPU） 3200
至系统总线
3202
I/O单元
3206
前端单元
3210
调度器单元
3212
集线器
3216
工作分配单元
3214
3208
GPU互连
GPC
3218
715
3220
XBar
存储器
(Y)
3204
存储器分区单元（U）
3222

图32

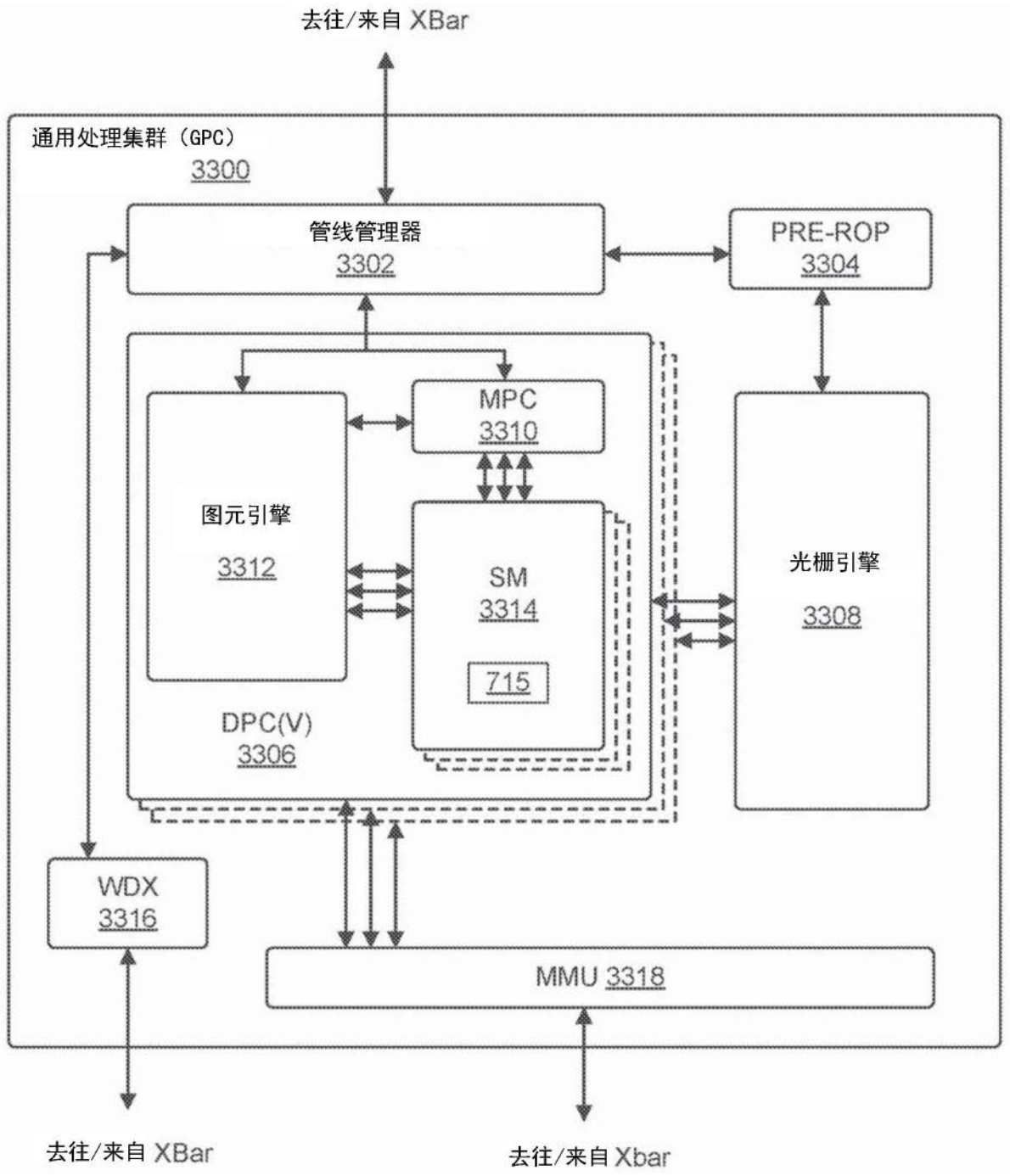

去往/来自 XBar
通用处理集群（GPC）
3300
管线管理器
3302
PRE-ROP
3304
MPC
3310
图元引擎
3312
SM
3314
715
光栅引擎
3308
DPC(V)
3306
WDX
3316
MMU 3318
去往/来自 XBar
去往/来自 Xbar

图33

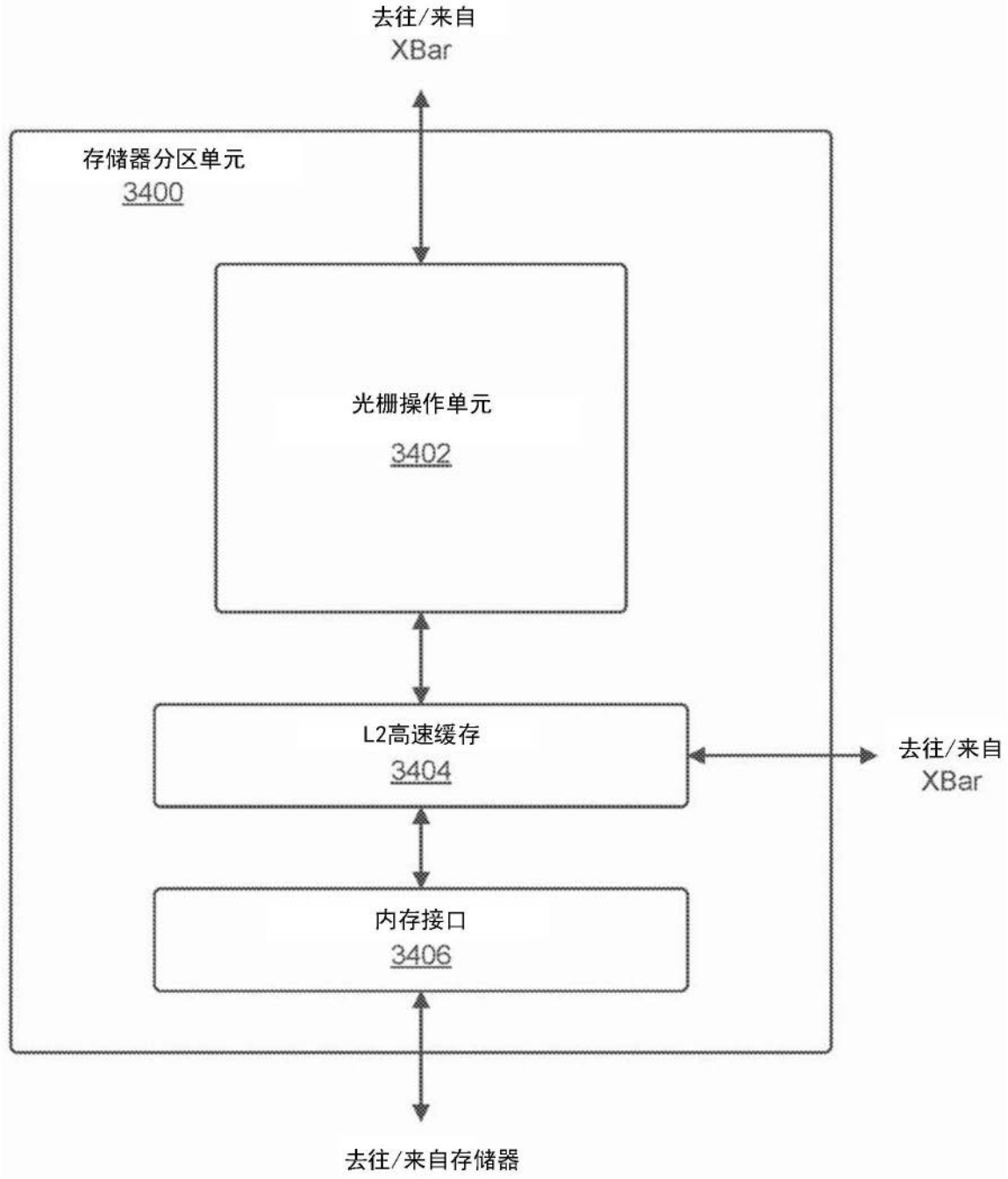

去往/来自
XBar
存储器分区单元
3400
光栅操作单元
3402
L2高速缓存
3404
去往/来自
XBar
内存接口
3406
去往/来自存储器

图34

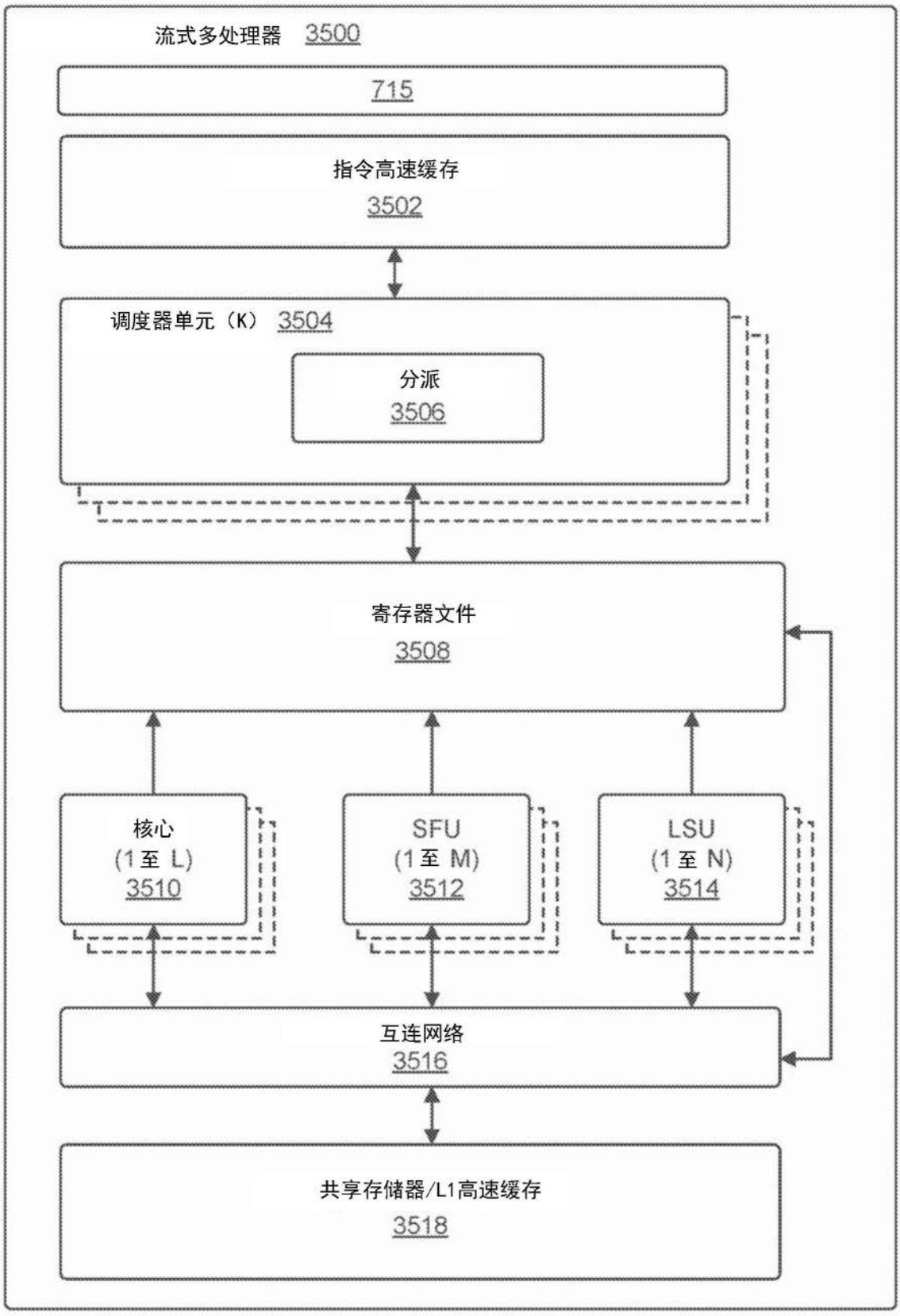

流式多处理器 3500
715
指令高速缓存
3502
调度器单元（K） 3504
分派
3506
寄存器文件
3508
核心
(1至 L)
3510
SFU
(1至 M)
3512
LSU
(1至 N)
3514
互连网络
3516
共享存储器/L1高速缓存
3518

图35

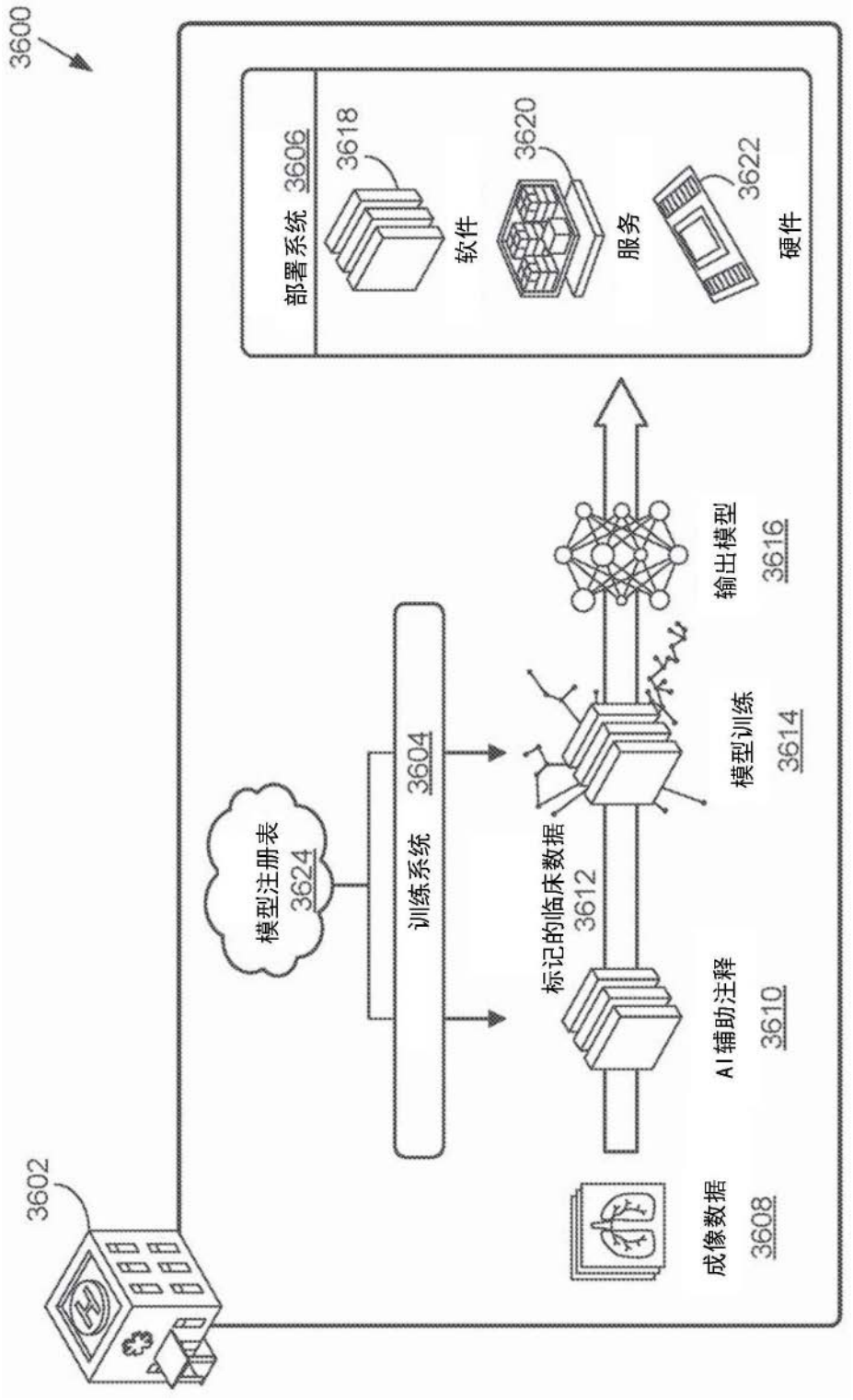

3600
部署系统 3606
3618
软件
3620
服务
3622
硬件
模型注册表
3624
训练系统 3604
标记的临床数据
3612
模型训练
3614
输出模型
3616
AI辅助注释
3610
成像数据
3608
3602

图36

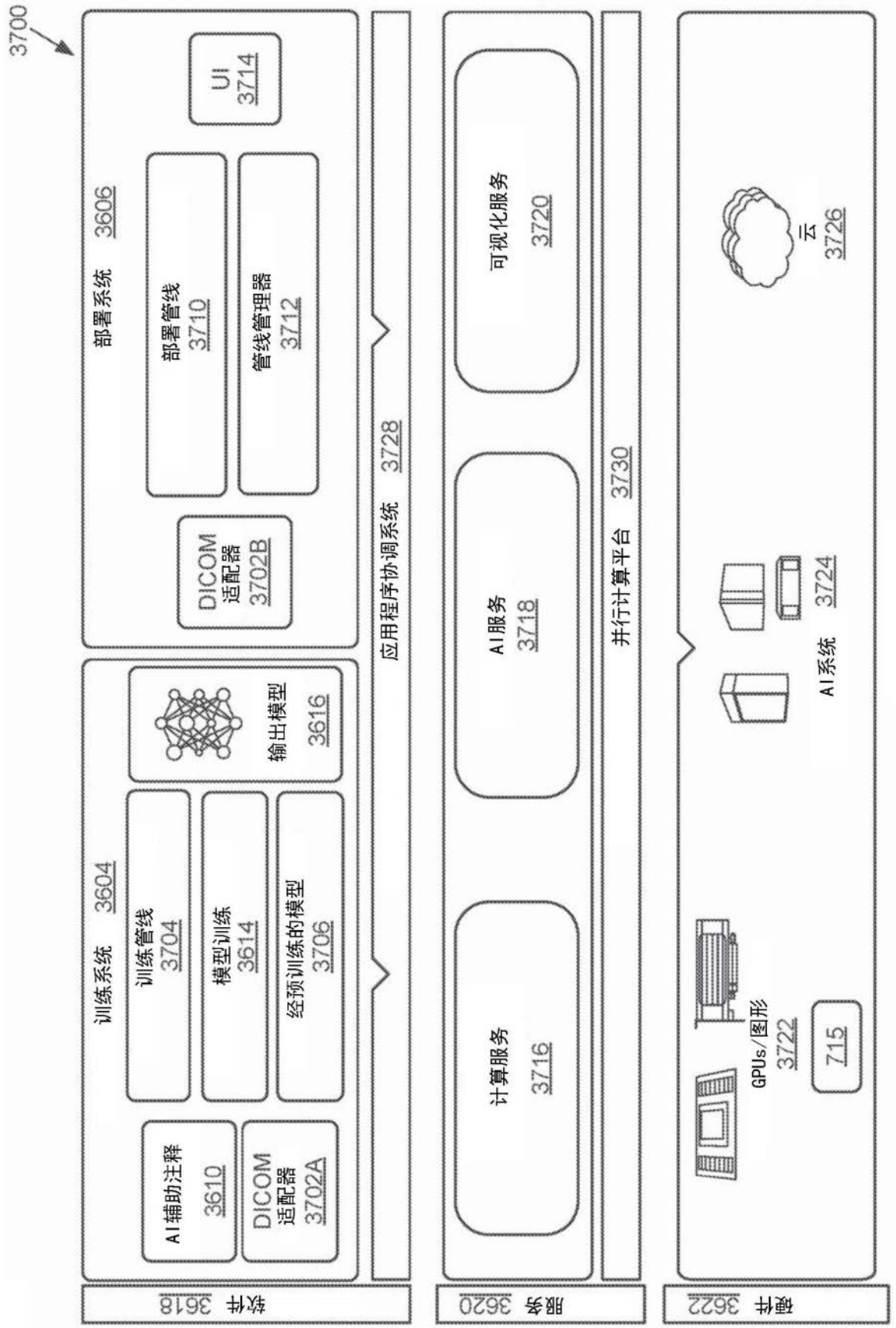

3700
部署系统 3606
部署管线 3710
管线管理器 3712
UI 3714
DICOM 适配器 3702B
应用程序协调系统 3728
训练系统 3604
训练管线 3704
模型训练 3614
经预训练的模型 3706
输出模型 3616
AI辅助注释 3610
DICOM 适配器 3702A
AI服务 3718
可视化服务 3720
计算服务 3716
并行计算平台 3730
云 3726
AI系统 3724
GPUs/图形 3722
Z15
软件 3618
服务 3620
硬件 3622

图37

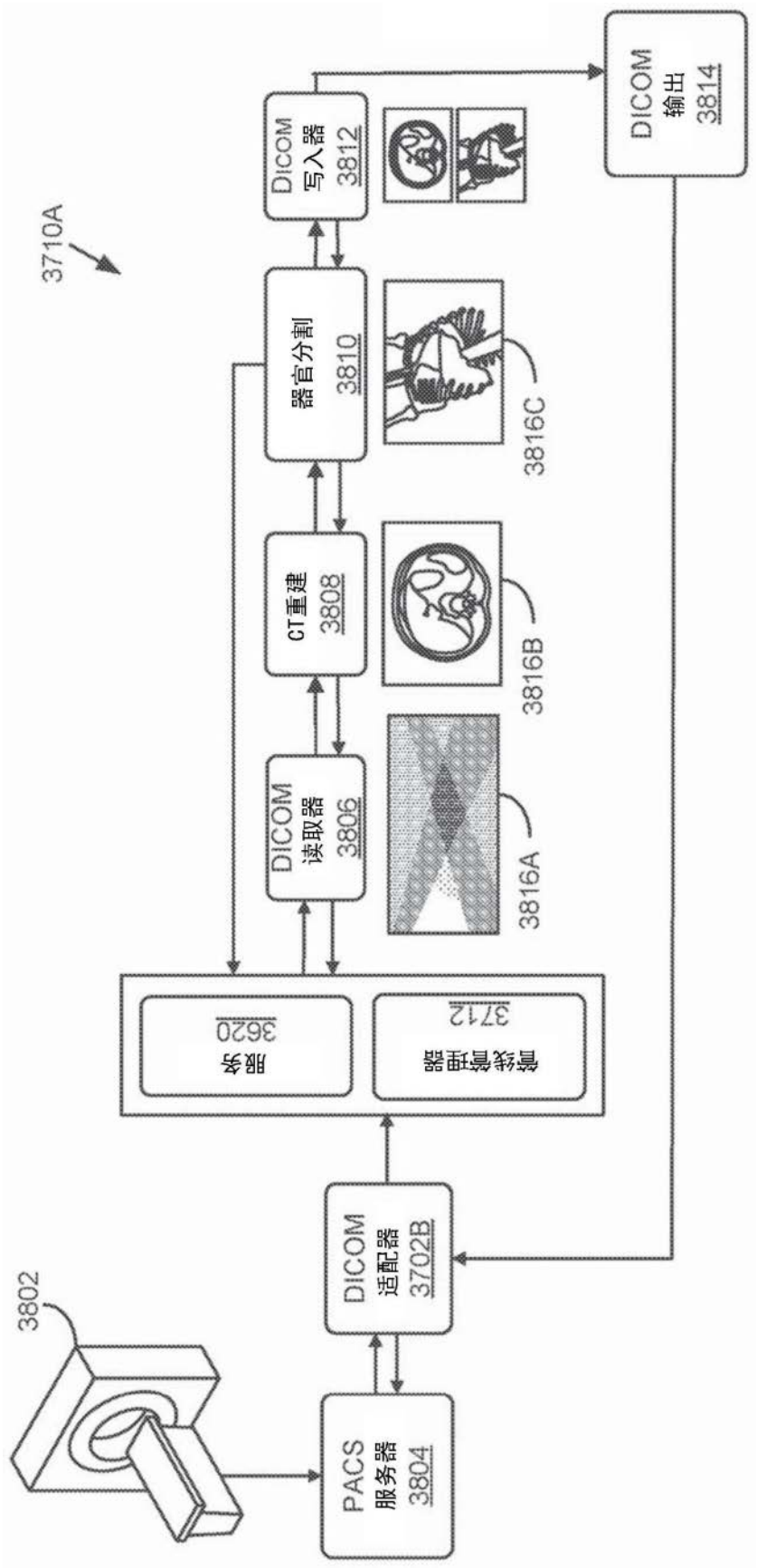

3710A
DICOM 写入器 3812
DICOM 输出 3814
器官分割 3810
CT重建 3808
DICOM 读取器 3806
3816A
3816B
3816C
服务 3620
管线管理器 3712
DICOM 适配器 3702B
PACS 服务器 3804
3802

图38

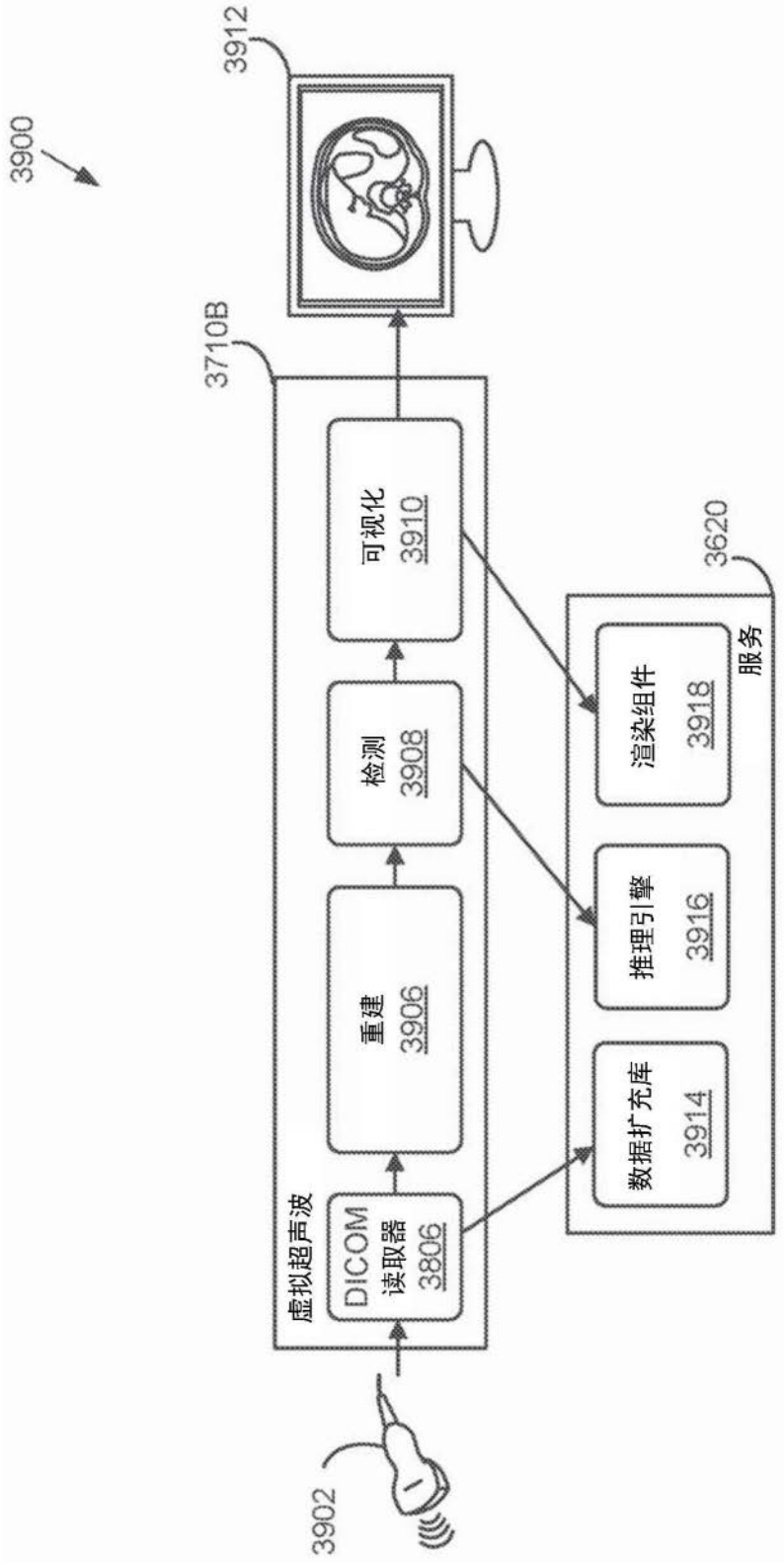

3900
3912
3710B
可视化
3910
检测
3908
重建
3906
DICOM
读取器
3806
虚拟超声波
3902
3620
渲染组件
3918
推理引擎
3916
数据扩充库
3914
服务

图39A

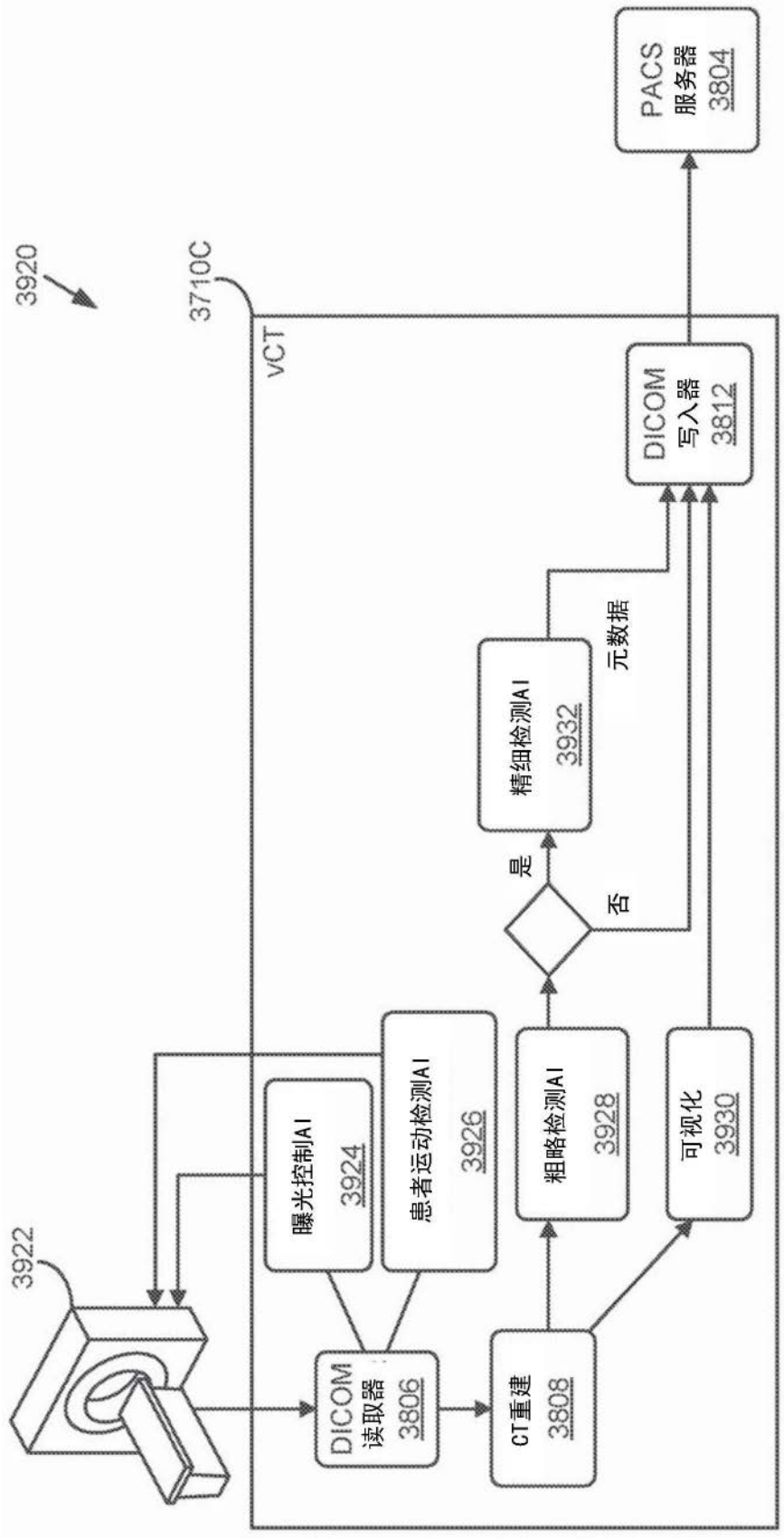

3920
3710C
vCT
PACS
服务器
3804
DICOM
写入器
3812
元数据
精细检测AI
3932
是
否
曝光控制AI
3924
患者运动检测AI
3926
粗略检测AI
3928
可视化
3930
3922
DICOM
读取器
3806
CT重建
3808

图39B

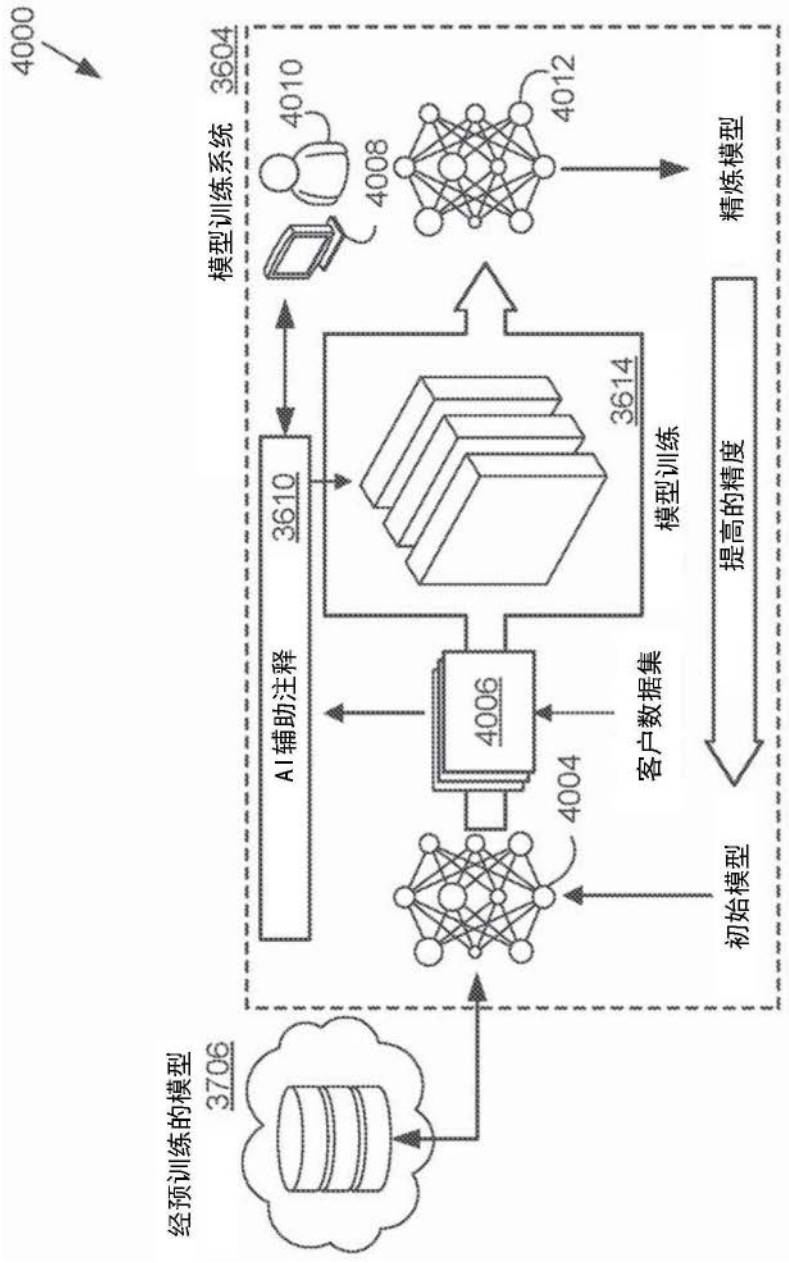

4000
模型训练系统
3604
4010
4008
4012
精炼模型
3614
模型训练
3610
AI辅助注释
提高的精度
4006
客户数据集
4004
初始模型
经预训练的模型
3706

图40A

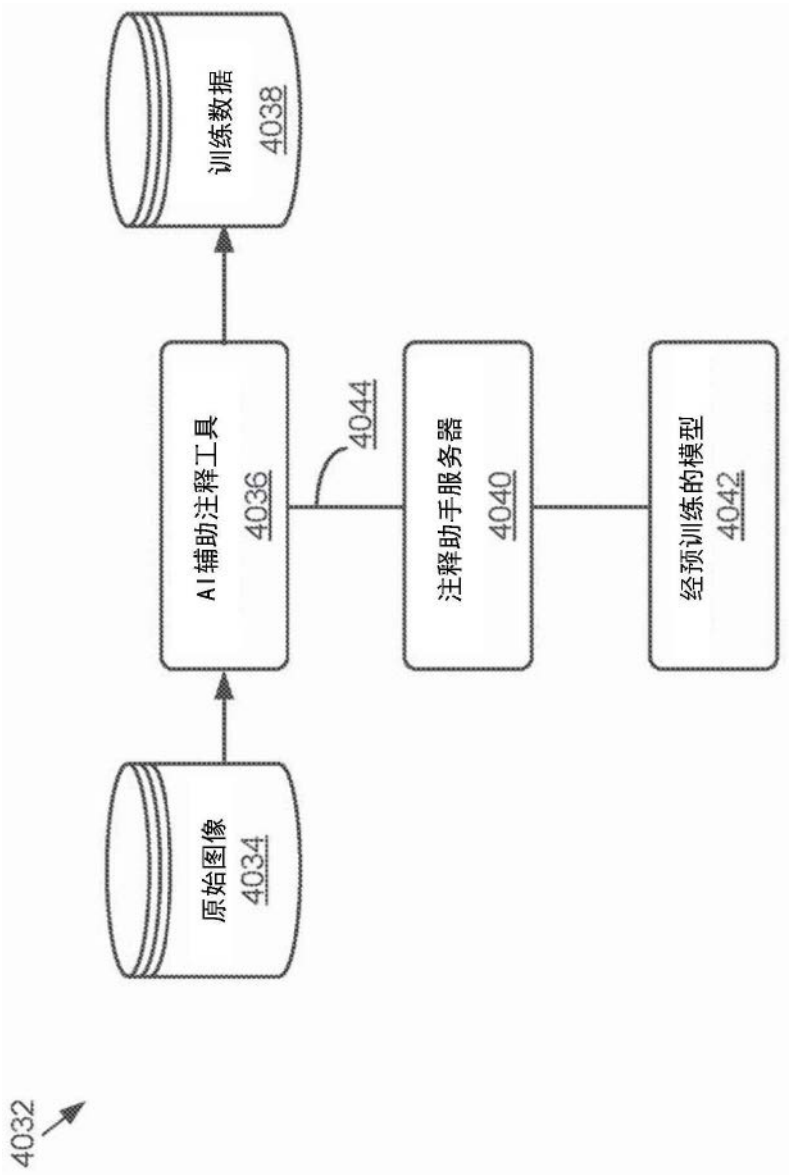

训练数据
4038
AI辅助注释工具
4036
4044
注释助手服务器
4040
经预训练的模型
4042
原始图像
4034
4032

图40B