



(12) 发明专利申请

(10) 申请公布号 CN 112154458 A

(43) 申请公布日 2020. 12. 29

(21) 申请号 201980032894.3

(22) 申请日 2019.05.20

(30) 优先权数据

62/673,747 2018.05.18 US

(85) PCT国际申请进入国家阶段日

2020.11.16

(86) PCT国际申请的申请数据

PCT/EP2019/062909 2019.05.20

(87) PCT国际申请的公布数据

W02019/219962 EN 2019.11.21

(71) 申请人 渊慧科技有限公司

地址 英国伦敦

(72) 发明人 W. 恰尔内茨基 S. 贾亚库马尔

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 金玉洁

(51) Int.Cl.

G06N 3/00 (2006.01)

G06N 3/04 (2006.01)

G06N 3/08 (2006.01)

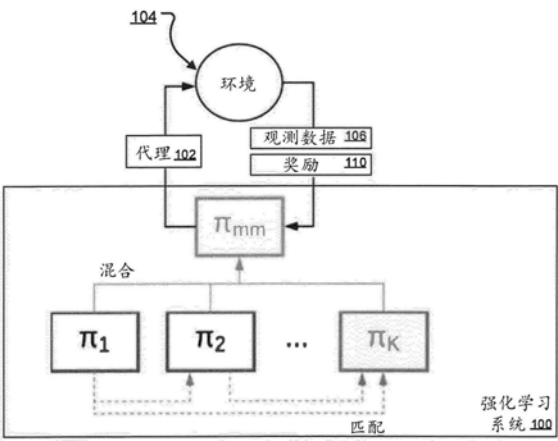
权利要求书2页 说明书13页 附图5页

(54) 发明名称

使用代理课程的强化学习

(57) 摘要

使用代理课程进行强化学习的方法、系统和装置,包括编码在计算机存储介质上的计算机程序。



1. 一种训练最终代理策略神经网络的方法,所述最终代理策略神经网络用于选择要由与环境交互的代理执行的动作以执行强化学习任务,所述方法包括:

维护指定多个候选代理策略神经网络的数据,其中所述多个候选代理策略神经网络包括所述最终代理策略神经网络,并且其中所述最终代理策略神经网络为所述代理定义比至少一个其他候选代理策略神经网络所定义的动作选择策略更复杂的动作选择策略;

初始化为所述候选代理策略神经网络中的每一个分配相应权重的混合数据;

联合训练所述多个候选代理策略神经网络以执行所述强化学习任务,包括,在多个训练迭代的每一个中:

获得包括对环境的观测数据的训练网络输入,

使用所述候选代理策略神经网络并且根据该训练迭代时的混合数据中的权重,使用所述训练网络输入生成组合的动作选择策略,以及

使用强化学习技术训练所述候选代理策略神经网络,以生成组合的动作选择策略,所述组合的动作选择策略导致在所述强化学习任务上的性能提高;以及

在训练期间,反复地调整所述混合数据中的权重,以支持更高性能的候选代理策略神经网络。

2. 根据权利要求1所述的方法,其中,联合训练所述多个候选代理策略神经网络以执行强化学习任务还包括:

训练所述候选代理策略神经网络以生成与由所述其他候选代理策略神经网络通过处理相同的训练网络输入生成的其他动作选择策略一致的动作选择策略。

3. 根据权利要求2所述的方法,其中,训练候选代理策略神经网络以生成一致的动作选择策略包括:

随着所述最终代理策略神经网络的权重增加,降低训练所述候选代理策略神经网络以生成一致的动作选择策略的影响。

4. 根据权利要求1-3中任一项所述的方法,其中,所述最终代理策略神经网络具有比至少一个其他候选代理策略神经网络更多的参数。

5. 根据权利要求1-4中任一项所述的方法,其中,与至少一个其他候选代理策略神经网络相比,所述最终代理策略神经网络生成为所述代理定义更大的动作空间的输出。

6. 根据权利要求1-5中任一项所述的方法,其中,使用所述候选代理策略神经网络并且根据该训练迭代时的混合数据中的权重,使用所述训练网络输入生成组合的动作选择策略,包括:

使用所述候选代理策略神经网络中的每一个来处理训练网络输入,以为每个候选代理策略神经网络生成相应的动作选择策略;以及

根据该训练迭代时的权重来组合所述动作选择策略,以生成组合的动作选择策略。

7. 根据权利要求1-6中任一项所述的方法,其中,联合训练所述多个候选代理策略神经网络以执行强化学习任务包括:

训练候选代理策略神经网络的组合的群体,并且其中反复地调整所述混合数据中的权重,以支持更高性能的候选代理策略神经网络,包括:

在训练期间,使用基于群体的训练技术、基于更高性能组合所使用的权重,调整较低性能组合所使用的混合数据中的权重。

8. 根据权利要求7所述的方法, 其中, 组合的性能基于在训练期间生成的所述组合的策略输出的质量。

9. 根据权利要求7所述的方法, 其中, 组合的性能仅基于由所述组合中的最终代理策略神经网络生成的策略输出的质量, 而不基于由所述组合中的其他代理策略神经网络生成的策略输出。

10. 存储指令的一个或多个计算机可读存储介质, 当由一个或多个计算机运行指令时, 所述指令使得一个或多个计算机执行权利要求1-9中任一项所述的方法的相应操作。

11. 一种包括一个或多个计算机和一个或多个存储指令的存储设备的系统, 当由一个或多个计算机运行指令时, 所述指令使得一个或多个计算机执行权利要求1-9中任一项所述的方法的相应操作。

使用代理课程的强化学习

技术领域

[0001] 本说明书涉及强化学习。

背景技术

[0002] 在强化学习系统中,代理(agent)通过执行由强化学习系统响应于接收表征环境的当前状态的观测数据(observation)而选择的动作来与环境交互。

[0003] 一些强化学习系统响应于接收给定的观测数据,根据神经网络的输出选择要由代理执行的动作。

[0004] 神经网络是采用非线性单元的一个或多个层来为接收到的输入预测输出的机器学习模型。一些神经网络是深度神经网络,其除了输出层之外,还包括一个或多个隐藏层。每个隐藏层的输出被用作该网络中下一个层(即下一个隐藏层或输出层)的输入。该网络的每个层根据相应参数集的当前值、从接收到的输入生成输出。

发明内容

[0005] 本说明书描述了一个或多个位置的一个或多个计算机上、被实现为计算机程序的系统,该系统训练最终动作策略神经网络,该神经网络用于选择要由与环境交互的强化学习代理执行的动作。

[0006] 该系统训练最终动作策略神经网络(即,在训练后将被用于控制强化学习代理的神经网络),作为候选代理策略神经网络集合的部分。最终动作策略神经网络一般定义该集合中的网络的任何一个的最复杂的策略,即,该集合中至少一个其他动作策略神经网络定义的动作选择策略不如最终动作策略神经网络定义的策略复杂。

[0007] 在训练的开始,系统初始化混合数据,该混合数据为该集合中的候选代理策略神经网络中的每一个分配相应的权重。

[0008] 然后,系统联合训练这些候选代理策略神经网络,以执行强化学习任务。特别地,在训练期间,系统使用组合的动作选择策略,该组合的动作选择策略是由该集合中的候选网络生成的单个动作选择策略的(根据混合数据中的权重的)组合。

[0009] 在训练期间,系统反复地调整混合数据中的权重,以支持(favor)更高性能的候选代理策略神经网络,例如通过向它们赋以更大的权重。

[0010] 因为该集合中的不同网络定义了具有不同复杂度水平的动作选择策略,并且因为不同网络之间的权重在整个训练期间被调整,所以代理策略神经网络集合也被称为代理的“课程(curriculum)”。

[0011] 组合的动作选择策略可用于选择要由代理执行的动作。然而,强化学习可以按策略(on-policy)执行,也可以不按策略(off-policy)执行。也就是说,训练候选代理策略神经网络可以在线执行,或者根据存储在回放(replay)存储器中的先前生成的(使用候选代理策略神经网络参数的较旧版本生成的)训练数据执行。

[0012] 如后面更详细地描述的,本文的“更复杂”一般涉及训练的复杂度。因此,更复杂的

动作选择策略可以指比另一个动作选择策略(例如,另一个候选代理策略神经网络的动作选择策略)花费更长的时间来训练(即,需要更多的训练步骤),以实现相同的性能,例如,在多个操作片段(episode)上的平均奖励(reward),或者对于不同的超参数设置(例如,学习速率、目标函数权重、小批次(mini-batch)大小等)不太鲁棒(robust)。在一些实现方式中,更复杂的动作选择策略可以对应于更复杂的架构,例如更深或更大(具有更多单元和/或参数)的代理策略神经网络,或者具有更多不同类型的层,例如包括循环(recurrent)层。在一些实现方式中,更复杂的动作选择策略可以对应于在更大的动作空间(即,在学习执行相同的任务的同时,还具有更多可从中选择的动作)中操作的策略。

[0013] 在一些实现方式中,通过处理相同的训练网络输入,候选代理策略神经网络被训练以生成与由其他候选代理策略神经网络生成的其他动作选择策略一致(aligned)的动作选择策略。例如,强化学习损失可以包括一致策略的成本,诸如取决于策略之间的差别(例如取决于强化学习的类型,对策略输出分布之间的差别的测量)的成本。

[0014] 随着最终代理策略神经网络的权重增加,系统可以降低训练候选代理策略神经网络以生成一致的动作选择策略的影响。也就是说,系统可以从使用多个候选代理策略神经网络逐渐切换到使用最终代理策略神经网络,并且在极限情况下,可以仅依赖最终代理策略神经网络来选择动作。这可以通过随着训练的进行、调整分配给混合更新的权重来实现。

[0015] 在实现方式中,生成组合的动作选择策略可以包括使用候选代理策略神经网络中的每一个来处理训练网络输入,以为每个候选代理策略神经网络生成相应的动作选择策略(输出),并且根据该训练迭代时的权重组合动作选择策略组合,以生成组合的动作选择策略。

[0016] 原则上,权重可以被手动调整,或通过使用适当的退火(anneal)策略调整。然而,在一些实现方式中,候选代理策略神经网络的组合的群体(population)被训练。然后,在训练期间,可以通过基于更高性能组合所使用的权重来调整较低性能组合所使用的权重,来调整权重。例如,如下所述,可以使用基于群体的训练技术,使得性能不佳的组合(如由组合的动作选择策略的性能度量所测量的)复制更强组合的神经网络参数,并且执行对它们的超参数的局部修改,使得性能不佳的组合被用于探索超参数空间。可以使用取决于训练期间生成的组合策略输出的质量的任何方便的性能度量,例如k个片段上的奖励。

[0017] 可以实现本说明书中描述的主题的特定实施例,以便实现以下优点中的一个或多个。

[0018] 通过在如本说明书所述的训练中对代理使用课程(即,通过如本说明书所述的调整权重),复杂代理可以学习(即,可以训练复杂代理策略选择神经网络),以使用比传统方法更少的计算资源和更少的训练时间来执行强化学习任务。特别地,通过利用课程中不太复杂的代理的知识,更复杂的代理可以在强化学习任务中快速地实现高性能,即,比复杂代理在特定任务上以独立的方式被训练快得多。事实上,在一些情况下,通过使用代理课程,复杂代理可以在任务上快速地实现高性能,即使该代理在以独立方式训练时无法从头开始学习任务。换句话说,更复杂的代理可以从更简单的代理所找到的解决方案中自举(bootstrap),以学习该代理原本不可能学习的任务,或者以比原本需要的更少的训练迭代来学习任务。此外,通过如本说明书中所述的分布所述训练和所述权重的调整,相对于仅训练单个最终代理,总训练时间可以减少,即使多个代理被联合训练也是如此。

[0019] 说明书描述的主题的一个或多个实施例的细节在附图和以下描述中阐述。根据说明书、附图和权利要求,本主题的其他特征、方面和优点将变得清楚。

附图说明

[0020] 图1示出了示例强化学习系统。

[0021] 图2A-图2C是示出了各种候选代理策略神经网络的示例的图。

[0022] 图3是用于训练候选代理策略神经网络集合的示例过程的流程图。

[0023] 图4是用于执行训练迭代的示例过程的流程图。

[0024] 不同附图中相同的附图标记和名称表示相同的元件。

具体实施方式

[0025] 图1示出了示例强化学习系统100。强化学习系统100是在一个或多个位置的一个或多个计算机上被实现为计算机程序的系统的示例,在该系统中,可以实现下面描述的系统、组件和技术。

[0026] 强化学习系统100通过强化学习训练代理策略神经网络,用于控制代理102在与环境104交互时执行强化学习任务。

[0027] 特别地,在训练期间的每个时间步长处,强化学习系统100接收表征环境104的当前状态的数据。表征环境的状态的数据在本说明书中将被称为观测数据106。响应于观测数据,系统100选择要由代理102执行的动作,并使代理102执行所选的动作。一旦代理102已经执行了所选的动作,环境104就转换到新的状态,并且系统100接收奖励110。

[0028] 一般地,奖励110是数值。奖励110可以指示代理102是否已经完成了任务,或者代理102朝向完成任务的进展。例如,如果任务规定代理102应该通过环境导航到目标位置,则一旦代理到达该目标位置,每个时间步长处的奖励可以具有正值,否则为零值。作为另一示例,如果任务规定代理应该探索环境,则如果代理在时间步长处导航到先前未探索的位置,则该时间步长处的奖励可以具有正值,否则为零值。

[0029] 在一些实现方式中,环境是真实世界环境,并且代理是与真实世界环境交互的机械代理,例如,在环境中导航的机器人或自动或半自动陆地、空中或海上交通工具。

[0030] 在这些实现方式中,观测数据可以包括,例如图像、对象位置数据和传感器数据中的一个或多个,以作为代理在其与环境交互时捕获观测数据,例如来自图像、距离或位置传感器或来自致动器的传感器数据。

[0031] 例如,在机器人的情况下,观测数据可以包括表征机器人当前状态的数据,例如,以下中的一个或多个:关节位置、关节速度、关节力、扭矩或加速度,例如重力补偿的扭矩反馈,以及机器人保持物品的整体或相对姿势。

[0032] 在机器人或其他机械代理或者交通工具的情况下,观测数据可以类似地包括位置、线速度或角速度、力、扭矩或加速度、以及代理的一个或多个部分的整体或相对姿势中的一个或多个。观测数据可以被定义为一维、二维或三维,并且可以是绝对和/或相对观测数据。

[0033] 例如,观测数据还可以包括所感测到的电子信号,诸如马达电流或温度信号;和/或例如来自相机或LIDAR传感器的图像或视频数据,例如来自代理的传感器的数据或来自

与环境中的代理分开放置的传感器的数据。

[0034] 在这些实现方式中,动作可以是控制机器人的控制输入,例如机器人关节的扭矩或更高级别的控制命令;或者是控制自动或半自动的陆地、空中、海上交通工具的控制输入,例如对交通工具的控制表面或其他控制元件的扭矩或更高级别的控制命令。

[0035] 换句话说,例如,动作可以包括机器人的一个或多个关节或另一机械代理的部分的位置、速度或力/扭矩/加速度数据。附加地或可替代地,动作数据可以包括电子控制数据,诸如马达控制数据,或者更一般地,用于控制环境中的一个或多个电子设备的数据,对电子设备的控制对观测到的环境状态有影响。例如,在自动或半自动陆地或空中或海上交通工具的情况下,动作可以包括控制导航(例如转向)以及运动(例如交通工具的制动和/或加速)的动作。

[0036] 在一些实现方式中,环境是模拟环境,并且代理被实现为与模拟环境交互的一个或多个计算机。

[0037] 模拟环境可以是运动模拟环境,例如驾驶模拟或飞行模拟,并且代理可以是在运动模拟中进行导航的模拟交通工具。在这些实现方式中,动作可以是控制模拟用户或模拟交通工具的控制输入。

[0038] 在另一示例中,模拟环境可以是视频游戏,并且代理可以是玩视频游戏的模拟用户。一般,在模拟环境的情况下,观测数据可以包括先前描述的观测数据或观测数据类型中的一个或多个的模拟版本,并且动作可以包括先前描述的动作或动作类型中的一个或多个的模拟版本。

[0039] 在电子代理的情况下,观测数据可以包括来自监控工厂或服务设施的部分的一个或多个传感器的数据,诸如电流、电压、功率、温度和其他传感器和/或表示的电子和/或机械设备部件的功能的电子信号。

[0040] 在一些其他应用中,代理可以例如在数据中心、电/水配给系统中、或者在制造厂或服务设施中,控制包括设备部件的真实环境中的动作。观测数据可以与工厂或设施的操作有关。例如,观测数据可以包括对设备的电或水使用的观测数据,或对发电或配电控制的观测数据,或者对资源使用或废物产生的观测数据。动作可以包括在工厂/设施的设备部件上控制或强加操作条件的动作,和/或导致工厂/设施的操作中的设置改变的动作,例如调整或打开/关闭工厂/设施的组件。

[0041] 系统100训练最终动作策略神经网络(即在训练后将用于控制强化学习代理的神经网络),作为候选代理策略神经网络集合的部分。在图1的示例中,该集合中的神经网络被表示为 π_1 至 π_K ,其中 π_K 表示最终动作策略神经网络。

[0042] 一般地,集合中的每个动作策略神经网络接收包括观测数据的网络输入,并生成定义动作选择策略的网络输出,该动作选择策略用于响应于观测数据选择要由代理执行的动作。

[0043] 在一些实现方式中,网络输出定义了可能动作集中的动作的似然分布。例如,网络输出可以包括可能动作集中每个动作的相应数字似然值。作为另一示例,网络输出可以包括定义参数概率分布的参数的相应数值(例如,正态分布的均值和标准差)。在这个示例中,可能动作集可以是连续的集合(例如,实数的连续范围)。在这些实现方式中的一些中,系统100通过基于似然性分布从可能动作集中对动作进行采样来选择要由代理执行的动作。

[0044] 在一些实现方式中,网络输出从可能动作集中识别动作。例如,如果代理是机器人代理,网络输出可以识别要施加到代理关节的扭矩。在这些实现方式中的一些中,系统100选择由网络输出识别的动作作为要由代理执行的动作,或者向所识别的动作添加噪声,并且选择噪声动作(noisy action)作为要执行的动作。

[0045] 在一些实现方式中,网络输入包括观测数据和来自可能动作集的给定动作两者,并且网络输出是,如果代理响应于观测数据执行给定动作、对系统将接收到的返回的估计。返回是指当代理在多个时间步长中与环境交互时,对系统接收到的奖励的累积测量。例如,返回可以指系统接收到的长期时间折扣(time-discounted)奖励。在这些实现方式中的一些中,系统100可以选择具有最高返回的动作作为要执行的动作,或者可以应用 ϵ -贪婪(epsilon-greedy)动作选择策略。

[0046] 尽管策略神经网络都接收相同类型的网络输入并生成相同类型的网络输出,但最终策略神经网络一般是集合中最复杂的神经网络。换句话说,最终代理策略神经网络为代理定义动作选择策略,该动作选择策略比由至少一个其他候选代理策略神经网络定义的动作选择策略更复杂。

[0047] 如本说明书中所使用的,动作选择策略的复杂度是指训练复杂度,即从头开始训练神经网络、以使代理使用由神经网络生成的动作选择策略来执行强化学习任务有多困难。出于各种原因,对于给定的强化学习任务,一个神经网络可能比另一个神经网络更复杂。

[0048] 例如,一个神经网络可以生成比其他网络为代理定义更大的动作空间的输出。特别地,集合中的其他候选网络可以被约束以生成策略,该策略将被选择的非零似然仅分配给可以由代理执行的有限数量的可能动作,而最终代理策略神经网络的输出不这样被约束。

[0049] 作为另一示例,一个神经网络可以具有比另一个神经网络更复杂的神经网络架构。例如,最终代理策略神经网络可能具有比其他网络多得多的参数。作为另一示例,最终代理策略神经网络可以包括其他网络中不包括的某些类型的层。作为特定的示例,最终代理策略神经网络可以包括一般难以训练到收敛(例如循环神经网络层)并且在其他候选神经网络中不存在的层。

[0050] 作为另一示例,强化学习任务可以是多个不同的个体任务的组合,并且一个神经网络可以是不同的个体任务中的每一个生成相应输出的多任务神经网络,而其他的神经网络只为一个个体任务生成输出。

[0051] 图2A-图2C是示出了具有不同复杂程度的各种候选代理策略神经网络的示例的图。

[0052] 在图2A的示例中,系统正在使用动作-评价(actor-critic)强化学习技术来训练候选神经网络。因此,组合的输出包括组合的策略输出 π_{mm} 和组合的值输出 v_{mm} 两者。

[0053] 组合的值输出为接收到的观测数据“obs”所表征的环境的当前状态分配值。特别地,该值是如果根据当前策略从当前状态的环境开始选择动作、对系统将接收到的预期返回的估计。

[0054] 组合的策略输出定义了代理响应于观测数据而执行的操作。例如,组合的策略输出可以是要由代理执行的可能动作集的概率分布,并且系统可以通过从概率分布中采样来

选择动作。

[0055] 具体地,图2A示出了两种架构210和220,其中架构220比架构210更复杂,即架构220在强化学习任务上更难从头开始训练。架构210和220可以是包括在候选代理策略神经网络集合中的两个代理策略神经网络中的架构。尽管在图2A的示例中,这是该集合中仅有的两个神经网络,但是实际上,该集合还可以包括图2A中未示出的其他候选代理策略神经网络。

[0056] 在图2A的示例中,架构210和220两者都接收观测数据(obs)并处理该观测数据以生成相应的策略输出 π_1 和 π_2 。两种架构都包括卷积编码器神经网络,其后是一个或多个长短期记忆(long short-term memory,LSTM)层。事实上,在一些实现方式中,架构210和220的这些部分是被共享的,即,参数值被约束为在两种架构之间相同。

[0057] 然而,架构210包括线性层,其后是掩码(masking)操作,该掩码操作将被分配给集合中可能动作的子集的概率设置为零。因此,由架构210生成的策略输出可以将被选择的非零似然仅分配给可以由代理执行的有限数量的可能动作。另一方面,架构220包括线性层,后面没有掩码操作,因此架构220生成的策略输出可以将被选择的非零似然分配给可以由代理执行的可能动作中的任何一个。因此,由架构220生成的策略输出为代理定义了更大的动作空间。尽管图2A仅示出了架构220的线性层生成值输出,但是实际上,架构210的线性层也可以生成值输出,其与架构220的值输出相混合(组合),以生成组合的值输出。

[0058] 图2B示出了两种架构230和240。架构230包括卷积神经网络编码器,其后是一个或多个线性层,然后是生成策略输出和值输出的最终线性层。然而,架构240包括相同的卷积编码器,但是其后包括一个或多个LSTM层,随后是生成策略输出和值输出的最终线性层。因此,架构240是循环的(recurrent),而架构230不是。相对于架构230,这增加了架构240的训练复杂度,即,因为循环层比前馈线性层更难训练。

[0059] 图2C示出了三种架构250、260和270。在图2C的示例中,强化学习任务包括两个个体任务i和j。架构250只为任务i生成输出,且架构270只为任务j生成输出。另一方面,架构260为任务i和任务j两者都生成输出。因此,尽管架构250、260和270在参数数量方面是相似的并且包括相同类型的神经网络层,但是架构270训练起来更复杂,因为它必须在任务i和任务j两者上训练,而其他的架构仅在单个任务上训练。

[0060] 回到对图1的描述,在训练的开始,系统100初始化混合数据,该混合数据为集合中的候选代理策略神经网络中的每一个分配相应的权重。一般地,混合数据最初分配给集合中最不复杂的神经网络的权重比分配给集合中最复杂的神经网络的权重高得多。作为特定的示例,系统可以最初将为1的权重(或接近1的值)分配给集合中最不复杂的神经网络,同时将为0的权重(或接近0的值)分配给集合中的每个其他神经网络。

[0061] 系统100然后联合训练候选代理策略神经网络来执行强化学习任务。特别地,在训练期间,系统使用组合的动作选择策略 π_{mm} 来选择要由代理102执行的动作,该组合的动作选择策略 π_{mm} 是由集合中的候选网络生成的个体动作选择策略(根据混合数据中的权重)的组合。

[0062] 具体地,本说明书将描述系统为通过计算由策略神经网络生成的个体动作选择策略的加权和(即,根据混合数据中的权重进行加权)来组合动作选择策略。在替代实现方式中,系统可以根据混合数据中的权重对策略神经网络采样,然后使用由采样的策略网络生

成的输出作为组合的动作选择策略。

[0063] 在训练期间,系统100使用强化学习反复地调整代理策略神经网络的参数值。

[0064] 具体地,系统100通过强化学习来调整代理策略神经网络的参数值,使得作为组合(“混合”)由策略网络生成的个体动作选择策略的结果而生成的组合的动作选择策略在强化学习任务上表现出提高的性能。

[0065] 此外,在训练期间,系统100还训练候选代理策略神经网络,以生成与由其他候选代理策略神经网络通过处理相同的训练网络输入生成的其他动作选择策略一致的动作选择策略。这被称为“匹配”。

[0066] 此外,系统100反复地调整混合数据中的权重,以逐渐支持更复杂的代理策略神经网络,包括最终代理策略神经网络。

[0067] 因为权重最初支持最不复杂的网络,并且最不复杂的网络可以快速提高它们在强化学习任务上的性能,所以更复杂的代理策略神经网络可以最初(通过训练期间的匹配更新)从由更简单的网络找到的解决方案中自举,以帮助更复杂的网络学习任务。然而,尽管不太复杂的网络可以容易且快速地确定任务的一些解决方案,但是由于不太复杂的网络的有限容量,例如,由于不太复杂的网络的有限的动作空间、有限的架构容量等,解决方案一般是受限的。

[0068] 随着训练的进行,通过增加被分配给更复杂的网络的权重,更复杂的网络找到更好的解决方案,因为组合的策略输出变得更不依赖于由简单网络所找到的简单解决方案。

[0069] 训练后,集合中的其他候选网络可以被丢弃,并且最终策略神经网络可以被用于控制代理。可替代地,系统可以向另一个系统提供最终策略神经网络的参数的最终经训练的值,以用于控制代理。

[0070] 图3是用于训练候选策略神经网络的示例过程300的流程图。为了方便,过程300将被描述为由位于一个或多个位置的一个或多个计算机的系统来执行。例如,经过适当编程的强化学习系统(例如图1的强化学习系统100)可以执行过程300。

[0071] 系统初始化混合数据(步骤302)。特别地,如上所述,系统初始化混合数据,以向不太复杂的策略网络分配比更复杂的策略网络更高的权重。

[0072] 系统根据混合数据,训练集合中的动作策略神经网络(步骤304)。特别地,系统执行一个或多个训练迭代来更新集合中策略网络的参数值。在训练期间,系统更新策略网络的参数值,以(1)生成组合的动作选择策略,该组合的动作选择策略导致在强化学习任务上的性能提高,以及(2)生成与由其他候选代理策略神经网络通过处理相同的训练网络输入生成的其他动作选择策略一致的动作选择策略。下面将参考图4更详细地描述执行对训练策略神经网络的迭代。

[0073] 系统调整混合数据中的权重(步骤306)。

[0074] 在一些实现方式中,系统使用预定的退火调度来调整权重,以增加被分配给更复杂的策略网络的权重。例如,退火调度可以规定随着训练的进行、被分配给更复杂的策略网络的权重线性增加,而被分配给不太复杂的策略网络的权重线性减少。

[0075] 在其他实现方式中,系统采用基于群体的训练技术来更新混合数据中的权重。在这种技术中,系统并行地训练候选代理策略神经网络集合的群体,即,训练多个不同的、完全一样的候选代理策略神经网络集合。在该训练期间,系统使用基于群体的训练技术、基于

更高性能的组合(群体)所使用的权重,周期性地调整较低性能的组合(群体)所使用的混合数据中的权重。

[0076] 换句话说,系统并行地训练集合的群体,这些集合周期性地相互查询,以检查它们相对于其他集合的性能如何。性能差的集合复制更强的集合的权重(神经网络参数),并且性能差的集合采用超参数,这些超参数是对更强的集合的超参数的局部修改。这样,性能差的集合被用于探索超参数空间。

[0077] 在Jaderberg、Max、Dalibard、Valentin、Osindero、Simon、Czarnecki、Wojciech M.、Donahue、Jeff、Razavi、Ali、Vinyals、Oriol、Green、Tim、Dunning、Iain、Simonyan、Karen、Fernando、Chrisantha和Kavukcuoglu、Koray的“Population based training of neural networks”,CoRR,2017一文中,更详细地描述了使用基于群体的训练来进行训练和用于使用基于群体的训练复制参数并探索超参数(包括混合权重)的技术。

[0078] 为了评估给定的策略网络集合的表现,系统可以(i)基于由该集合在训练期间生成的组合的策略输出的质量,或者(ii)仅基于由该集合中的最终代理策略神经网络生成的策略输出的质量、而不基于由该集合中的其他代理策略神经网络生成的策略输出来评估性能。作为示例,评估函数可以测量(i)当使用组合的策略来控制时、对任务的最后k个片段的奖励,或者(ii)如果仅使用最终策略来控制代理、对任务的最后k个片段的奖励。当模型被认为具有从简单的模型切换到更复杂的模型的明显好处(就性能而言)时,使用(i)评估性能可以获得良好的结果。当不知道是否会是一种情况时,使用(ii)评估性能可能比使用(i)来评估可以获得更好的结果。

[0079] 对于混合数据中权重的探索函数,系统可以随机添加或减去固定值(在0和1之间截断(truncating))。

[0080] 因此,使用基于群体的训练,一旦切换到更复杂的训练有显著的好处,切换将作为开发(exploitation)/探索过程的部分自动地发生。

[0081] 系统可以重复执行步骤304和306,以更新神经网络的参数,并调整混合数据中的权重,直到满足某些标准,例如,已经执行了一定次数的训练迭代,或者最终网络的性能满足某些标准。

[0082] 图4是用于执行训练迭代的示例过程400的流程图。为了方便,过程400将被描述为由位于一个或多个位置的一个或多个计算机的系统来执行。例如,适当编程的强化学习系统(例如图1的强化学习系统100)可以执行过程400。

[0083] 当系统使用基于群体的训练技术时,系统可以对群体中的每个候选集合并行地执行过程400。

[0084] 系统确定对策略神经网络的参数的当前值的强化学习更新(步骤402)。

[0085] 系统可以使用适合策略网络被配置为生成的网络输出的种类的任何强化技术来确定强化学习更新。

[0086] 特别地,强化学习技术可以是按策略的技术,也可以是不按策略的技术。

[0087] 当技术是按策略的技术时,系统通过根据策略网络的参数的当前值控制代理来生成训练数据,即,通过使用根据当前值生成的组合的策略输出来控制代理,然后在训练数据上训练神经网络。

[0088] 更详细地,为了生成训练数据,系统可以重复地使代理在环境中动作,直到生成了

阈值量的训练数据。为了使代理在环境中动作,系统接收观测数据,并且使用候选代理策略神经网络中的每一个、使用每个策略来处理包括该观测数据的网络输入,以为每个候选代理策略神经网络生成相应的动作选择策略。然后,系统根据该训练迭代时的混合数据中的权重来组合动作选择策略,以生成组合的动作选择策略,即,通过计算动作选择策略的加权和,然后根据组合的动作选择策略来选择要由代理执行的动作。

[0089] 为了训练神经网络,系统计算强化学习损失函数 $\mathcal{L}_{RL}$ 的梯度,该损失函数适于策略网络被配置为生成的网络输出的种类并且促进组合的策略在强化学习任务上表现出提高的性能。按策略的强化学习的强化学习损失函数的示例包括SARSA损失函数和按策略的动作-评价损失函数。特别地,作为计算梯度的部分,系统通过组合的策略输出反向传播到集合中的各个神经网络,以便计算对网络参数的更新。

[0090] 当技术是不按策略的技术时,系统将环境中的动作分离(decouple),以从在训练数据上的训练中生成训练数据。

[0091] 具体地,系统通过使代理在如上所述的环境中动作来生成训练数据,然后将训练数据存储在重放存储器中。

[0092] 然后,系统从重放存储器中采样训练数据,并使用采样的训练数据来训练神经网络。因此,在任何给定训练迭代中使用的训练数据可能已经使用与给定训练迭代的当前值不同的参数值生成了。尽管如此,训练数据是通过使用组合的控制策略控制代理而生成的。

[0093] 为了训练神经网络,系统计算不按策略的强化学习损失函数的梯度,该损失函数适于策略网络被配置为生成的网络输出的种类并且促进组合的策略在强化学习任务上表现出提高的性能。当计算梯度时,系统使用组合的策略并根据混合数据中的当前权重来计算作为强化损失函数的输入的策略。不按策略的强化学习的强化学习损失函数的示例包括Q学习(Q-learning)损失函数和不按策略的动作-评价损失函数。特别地,作为计算梯度的部分,系统通过组合的策略输出反向传播到集合中的各个神经网络,以便对计算网络参数的更新。

[0094] 系统确定对策略神经网络的参数的当前值的匹配更新(步骤404)。一般地,匹配更新使由集合中的策略网络生成的动作选择策略彼此一致。在一些实现方式中,随着最终代理策略神经网络的权重增加,即,随着训练的进行,系统降低训练候选代理策略神经网络以生成一致的动作选择策略的影响。

[0095] 特别地,系统获得观测数据集,其是在与环境交互期间接收的,即,是作为使用组合的动作选择策略控制代理的结果而接收的。接收的观测数据可以与在计算强化学习更新中所使用的观测数据相同,或者可以是不同的观测数据集。例如,当强化学习技术是按策略的技术时,观测数据可以与所生成的训练数据中的观测数据相同。作为另一示例,当强化学习技术是不按策略的技术时,系统可以从仅存储最近遇到的观测数据的存储器缓冲器中获得观测数据集,即,而不是从存储在更长时期上遇到的观测数据的重放存储器获得观测数据集。

[0096] 然后,系统通过确定匹配成本函数的梯度来计算匹配更新,该匹配成本函数测量由集合中的策略网络生成的策略输出的差。特别地,匹配成本函数满足:

$$[0097] \quad \mathcal{L}_{mm}(\theta) = \sum_{i,j=1}^K D(\pi_i(\cdot|\cdot, \theta_i), \pi_j(\cdot|\cdot, \theta_j), i, j, \alpha)$$

[0098] 其中, K 是集合中网络的总数, 并且 D 是函数, 该函数用于测量在给定 (i) 两个策略网络 θ_i 和 θ_j 的参数的当前值以及 (ii) 混合数据中的当前权重 α 的情况下, 策略网络 π_i 和 π_j 为观测数据集生成的策略输出之间的差。

[0099] 作为特殊的示例, 集合中策略网络 π_1 和 π_2 之间的函数 D 可以满足:

$$[0100] \quad \frac{1-\alpha}{|S|} \sum_{s \in S} \sum_{t=1}^{|s|} D_{KL}(\pi_1(\cdot|s_t) \parallel \pi_2(\cdot|s_t)),$$

[0101] 其中, S 是观测数据集, s 是集合中观测数据的轨迹, $|s|$ 是轨迹中观测数据的数量, $|S|$ 是集合中观测数据的数量, D_{KL} 是 K-L 散度 (divergence), 并且符号 $(1-\alpha)$ 表示 1 减去混合数据中被分配给最终策略网络的权重。在这个示例中, 由于包含 $(1-\alpha)$ 项, 随着最终代理策略神经网络的权重增加, 系统降低训练候选代理策略神经网络生成一致的动作选择策略的影响。

[0102] 系统更新策略神经网络的参数的当前值 (步骤 406)。也就是说, 系统根据强化学习更新和匹配更新来确定最终更新, 然后将最终更新添加到参数的当前值。例如, 最终更新可以是强化学习更新和匹配更新的总和或加权和。等效地, 匹配成本函数 $\mathcal{L}_{mm}(\theta)$ 可以被添加到强化学习损失函数 $\mathcal{L}_{RL}$, 以形成用于训练的整体损失函数。

[0103] 系统可以继续重复过程 400, 直到满足用于更新混合数据中的权重的标准, 例如, 已经过去了一定量的时间, 已经执行了一定量的训练迭代, 或者直到最终策略网络在强化学习任务上达到可接受的准确度水平。

[0104] 本说明书结合系统和计算机程序组件使用术语“配置为”。对于由一个或多个计算机组成的系统来说, 被配置为执行特定的操作或动作意味着该系统已经在其上安装了软件、固件、硬件或它们的组合, 这些在操作中导致该系统执行操作或动作。对于被配置为执行特定操作或动作的一个或多个计算机程序来说, 意味着该一个或多个程序包括指令, 当该指令被数据处理设备装置时, 使得该装置执行操作或动作。

[0105] 在本说明书中所描述的主题和功能操作的实施例可以在数字电子电路中、在有形具体化的计算机软件或固件中、在计算机硬件 (包括在本说明书中所公开的结构和它们的结构等同物, 或者在它们中的一个或多个的组合中) 中实现。本说明书中所描述的主题的实施例可以被实现为一个或多个计算机程序, 即被编码在有形的非暂时性存储介质上的计算机程序指令的一个或多个模块, 用于由数据处理装置执行或控制数据处理装置的操作。计算机存储介质可以是机器可读存储设备、机器可读存储基底、随机或串行访问存储器设备, 或者它们中的一个或多个的组合。可替代地或附加地, 程序指令可以被编码在人工生成的传播信号 (例如, 机器生成的电、光或电磁信号) 上, 该信号被生成以编码信息, 用于到合适的接收器装置的传输, 以由数据处理装置执行。

[0106] 术语“数据处理装置”是指数据处理硬件, 并且包含用于处理数据的所有种类的装置、设备和机器, 包括例如可编程处理器、计算机或多个处理器或计算机。该装置还可以是,

或者还包括专用逻辑电路,例如,现场可编程门阵列(FPGA)或专用集成电路(ASIC)。除了硬件之外,该装置可以可选地包括为计算机程序创建执行环境的代码,例如,形成处理器固件、协议栈、数据库管理系统、操作系统或它们中的一个或多个的组的代码。

[0107] 计算机程序(也可以被称为或描述为程序、软件、软件应用、app、模块、软件模块、脚本或代码)可以用任何形式的编程语言(包括编译或解释语言、声明或过程语言)编写,并且计算机程序可以以任何形式部署,包括作为独立程序或作为模块、组件、子例程或适合在计算环境中使用的其他单元。程序可以,但不是必须,对应于文件系统中的文件。程序可以被存储在保存其他程序或数据的文件的部分(例如存储在标记语言文档中、存储在专用于所讨论的程序的单个文件中、或者存储在多个协同文件(例如存储一个或多个模块、子程序或代码部分的文件)中的一个或多个脚本)中。计算机程序可以被部署为在一个计算机上或者位于一个站点或分布在多个站点并通过数据通信网络互连的多个计算机上执行。

[0108] 在本说明书中,术语“数据库”被广泛用于指代任何数据的集合:数据不需要以任何特定的方式被结构化,或者根本不需要被结构化,并且数据可以被存储在一个或多个位置的存储设备上。因此,例如,索引数据库可以包括多个数据的集合,数据的集合中的每一个可以被不同地组织和访问。

[0109] 类似地,在本说明书中,术语“引擎”被广泛用于指代基于软件的系统、子系统或被编程为执行一个或多个特定功能的过程。一般地,引擎将被实现为安装在一个或多个位置的一个或多个计算机上的一个或多个软件模块或组件。在某些情况下,一个或多个计算机将专用于特定的引擎;在其他情况下,可以在相同的一个或多个计算机上安装并运行多个引擎。

[0110] 本说明书中所描述的过程和逻辑流程可以由一个或多个可编程计算机执行,该可编程计算机执行一个或多个计算机程序,以通过对输入数据进行操作并生成输出来执行功能。过程和逻辑流程也可以由专用逻辑电路(例如,FPGA或ASIC)来执行,或者通过专用逻辑电路和一个或多个编程的计算机的组合来执行。

[0111] 适于执行计算机程序的计算机可以基于通用或专用微处理器或两者,或者任何其他类型的中央处理单元。一般,中央处理单元将从只读存储器或随机访问存储器或者两者接收指令和数据。计算机的基本元件是用于执行(perform)或执行(execute)指令的中央处理单元以及一个或多个用于存储指令和数据的存储器设备。中央处理单元和存储器可以由专用逻辑电路补充,或者并入专用逻辑电路。一般,计算机还将包括或可操作地耦合到用于存储数据的一个或多个大容量存储设备(例如磁盘、磁光盘或光盘),以从其接收数据或向其传送数据,或两者兼有。然而,计算机不需要具有这样的设备。此外,计算机可以嵌入到另一个设备(例如移动电话、个人数字助理(PDA)、移动音频或视频播放器、游戏控制台、全球定位系统(GPS)接收器或便携式存储设备(例如通用串行总线(USB)闪存驱动器),仅举几个示例)中。

[0112] 适用于存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储器设备,包括例如半导体存储器设备(例如EPROM、EEPROM和闪存设备);磁盘(例如内部硬盘或可移动磁盘);磁光盘;以及CD ROM和DVD-ROM。

[0113] 为了提供与用户的交互,本说明书中所描述的主题的实施例可以在计算机上实现,该计算机具有用于向用户显示信息的显示设备(例如阴极射线管(CRT)或液晶显示器

(LCD) 监视器), 以及键盘和指示设备 (例如鼠标或轨迹球), 用户可以通过键盘和指示设备向计算机提供输入。也可以使用其他类型的设备来提供与用户的交互; 例如, 提供给用户的反馈可以是任何形式的感官反馈, 例如视觉反馈、听觉反馈或触觉反馈; 并且可以以任何形式接收来自用户的输入, 包括声音、语音或触觉输入。此外, 计算机可以通过向用户使用的设备发送文档和从用户使用的设备接收文档来与用户交互; 例如, 通过响应于从网络浏览器接收到的请求, 将网页发送到用户设备上的网络浏览器。此外, 计算机可以通过向个人设备 (例如, 运行消息应用的智能手机) 发送文本消息或其他形式的消息、并从用户接收响应消息来与用户交互。

[0114] 用于实现机器学习模型的数据处理装置还可以包括, 例如用于处理机器学习训练或生产的公共和计算密集型部分 (即推理、工作负荷) 的专用硬件加速器单元。

[0115] 机器学习模型可以使用机器学习框架 (例如, TensorFlow 框架、微软认知工具包 (Microsoft Cognitive Toolkit) 框架、Apache Singa 框架或 Apache MXNet 框架) 来实现和部署。

[0116] 本说明书中所描述的主题的实施例可以在计算系统中实现, 该计算系统包括后端组件 (例如作为数据服务器), 或者包括中间件组件 (例如应用服务器), 或包括前端组件 (例如具有图形用户界面、web 浏览器或 app 的客户端计算机, 用户可以通过该图形用户界面、web 浏览器或 app 与本说明书中所描述的主题的实现进行交互), 或者一个或多个这样的后端、中间件或前端组件的任意组合。系统的组件可以通过任何形式或介质的数字数据通信 (例如通信网络) 来互连。通信网络的示例包括局域网 (LAN) 和广域网 (WAN), 例如互联网。

[0117] 计算系统可以包括客户端和服务器。客户端和服务器一般彼此远离, 并且通常通过通信网络进行交互。客户机和服务器的关系是通过运行在相应计算机上并且彼此具有客户机-服务器关系的计算机程序产生的。在一些实施例中, 服务器将数据 (例如, HTML 页面) 传输到用户设备, 例如, 为了向充当客户端的、与该设备交互的用户显示数据和从该用户接收用户输入。可以在服务器处从设备接收在用户设备处生成的数据, 例如用户交互的结果。

[0118] 尽管本说明书包含多个具体的实现方式细节, 但是这些不应被解释为对任何发明或所要求保护的范围的限制, 而是对特定发明的特定实施例的特定特征的描述。本说明书中在单独实施例的上下文中所描述的某些特征也可以在单个实施例中组合实现。相反, 在单个实施例的上下文中所描述的各种特征也可以在多个实施例中单独实现或者在任何合适的子组合中实现。此外, 尽管特征可以在上文中被描述为在某些组合中起作用, 并且甚至最初被如此要求保护的, 但是在一些情况下, 来自所要求保护的组合的一个或多个特征可以从该组合中被删除, 并且所要求保护的组合可以针对子组合或子组合的变型。

[0119] 类似地, 尽管在附图和所述的权利要求中以特定顺序描述了操作, 但是这不应该理解为要求以所示的特定顺序或顺序地执行这些操作, 或者要求执行所有示出的操作, 以获得期望的结果。在某些情况下, 多任务和并行处理可能是有利的。此外, 上述实施例中的各种系统模块和组件的分离不应该被理解为在所有实施例中都需要这种分离, 并且应该理解, 所描述的程序组件和系统一般可以被一起集成在单个软件产品中或者封装到多个软件产品中。

[0120] 已经描述了主题的特定实施例。其他实施例在所附权利要求的范围内。例如, 权利要求中所列举的动作可以以不同的顺序执行, 并且仍然获得期望的结果。作为一个示例, 附

图中所描述的过程不一定需要所示的特定顺序或顺序地来实现期望的结果。在一些情况中,多任务和并行处理可能是有利的。

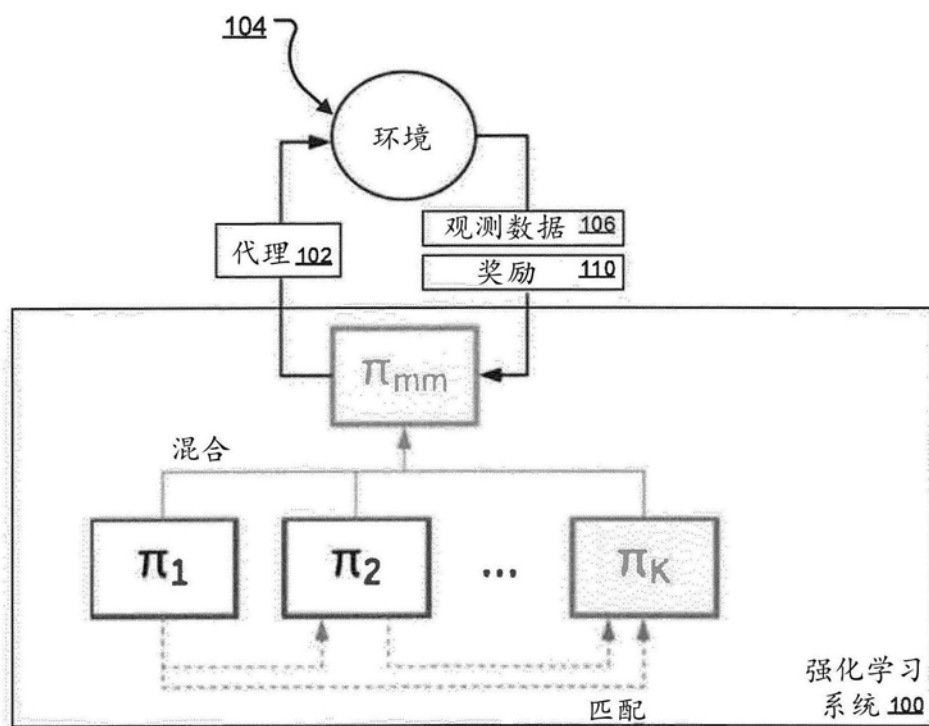


图1

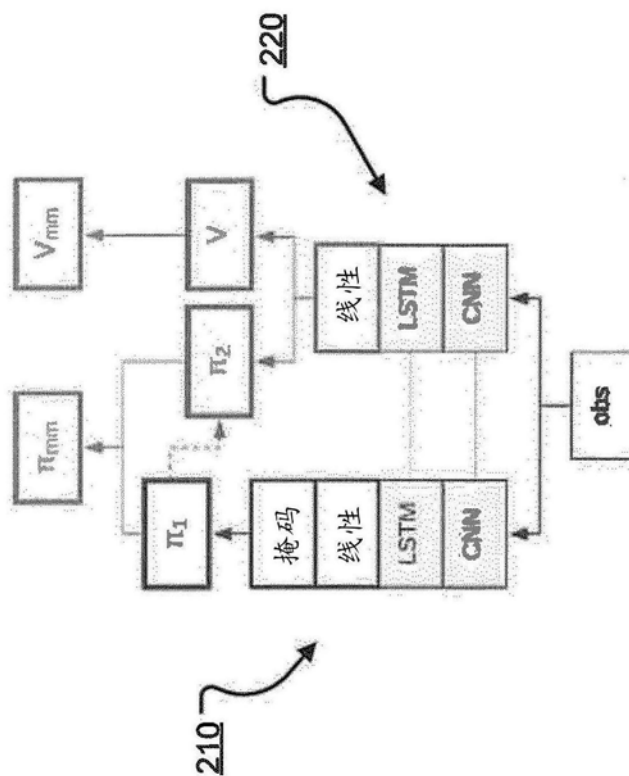


图2A

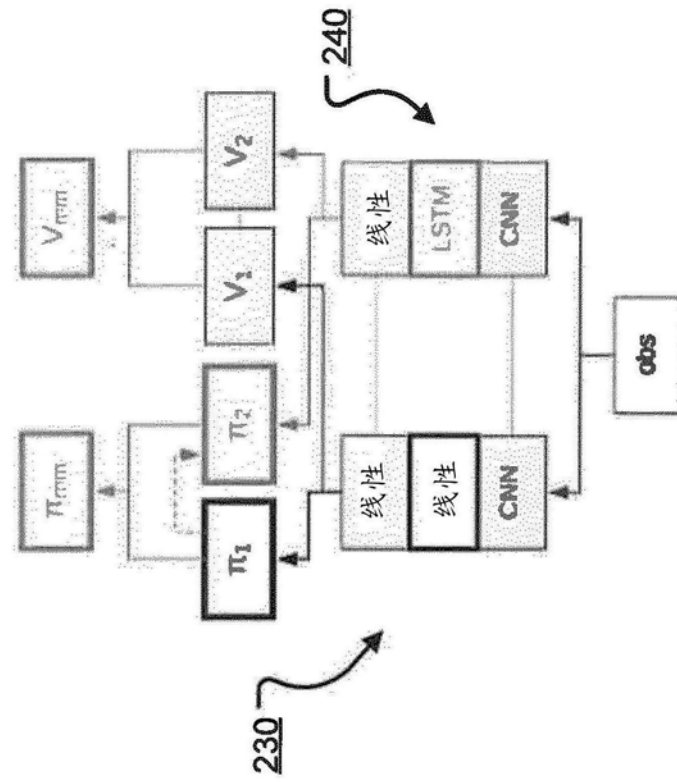


图2B

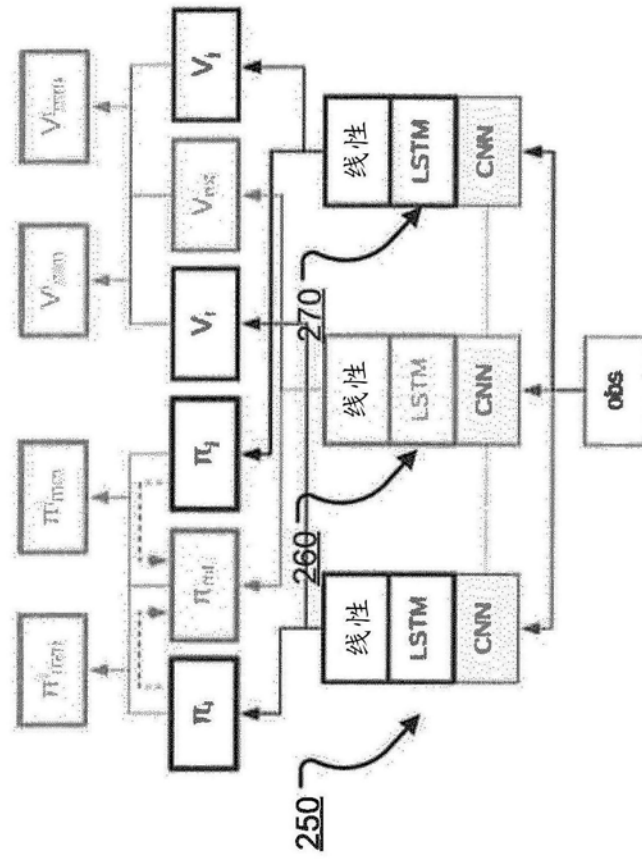


图2C

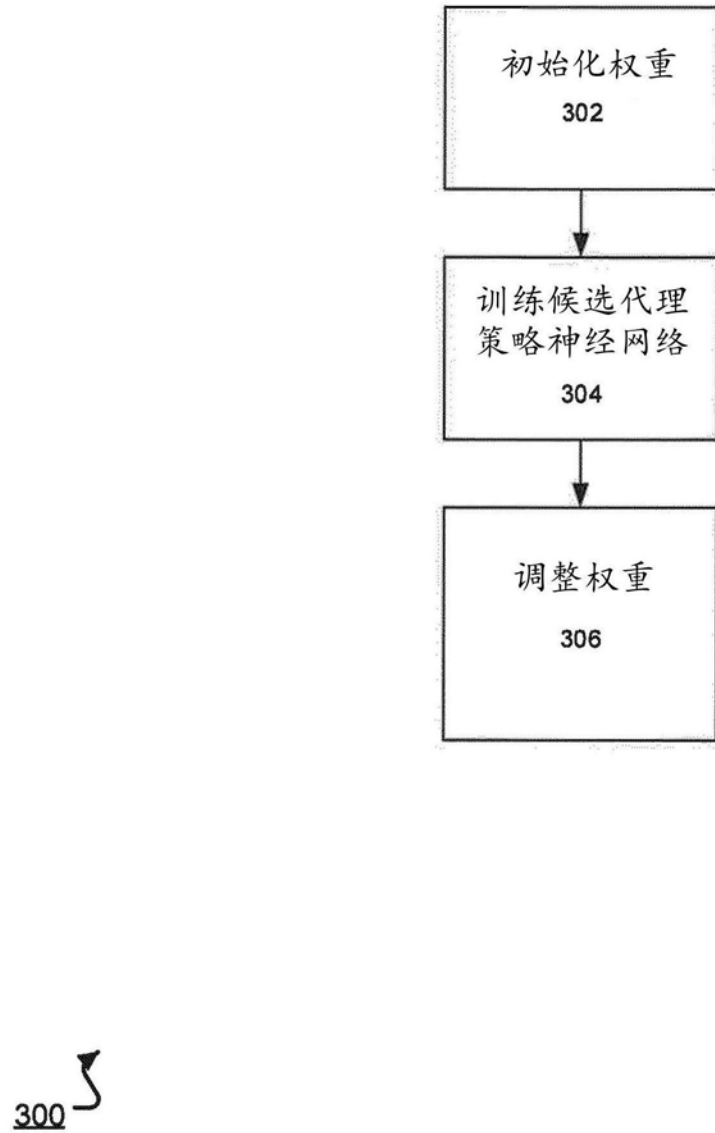
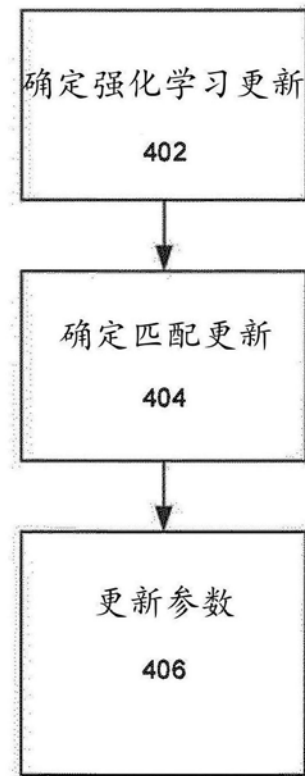


图3



400 }

图4