(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111160525 A
(43)申请公布日 2020.05.15

(21)申请号 201911299550.7

(22)申请日 2019.12.17

(71)申请人 天津大学
地址 300072 天津市南开区卫津路92号

(72)发明人 曲冠锦 吴华明

(74)专利代理机构 天津市三利专利商标代理有限公司 12107
代理人 张义

(51)Int.Cl.
G06N 3/04(2006.01)
G06N 3/08(2006.01)
H04L 29/08(2006.01)

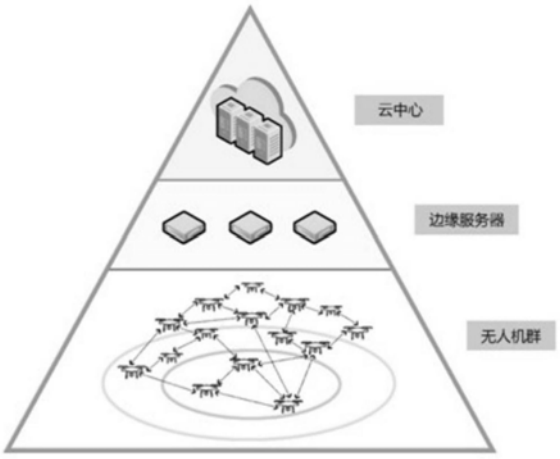
权利要求书1页 说明书6页 附图4页

(54)发明名称

一种边缘计算环境下基于无人机群的任务卸载智能决策方法

(57)摘要

本发明公开了一种边缘计算环境下基于无人机群的任务卸载智能决策方法,(1)采集环境信息;(2)进行元学习,若发现边缘服务器或云中心的环境发生变化将修改模型的初始参数;(3)进行检索机制与强化学习,其中检索机制负责检索之前是否存在相似任务,若存在,直接输出决策结果;若不存在,进行强化学习,强化学习负责训练和判定整个强化学习系统,其中用到的两大模块是网络冻结和经验回放,判定完后取值函数最大的动作为决策结果进行输出。本方案采用元学习模型可以快速适应环境,当决策系统的环境发生改变时,本方案可以快速调整并迅速给出合理结果。针对无人机群任务相似,本方案引入记忆功能,对相似的任务可以作出快速的决策。



CN 111160525 A

1.一种边缘计算环境下基于无人机群的任务卸载智能决策方法,其特征在于,包括如下步骤:

(1)采集环境信息;

(2)进行元学习,若发现边缘服务器或云中心的环境发生变化将修改模型的初始参数;

(3)进行检索机制与强化学习,其中检索机制负责检索之前是否存在相似任务,若存在,直接输出决策结果;若不存在,进行强化学习,强化学习负责训练和判定整个强化学习系统,其中用到的两大模块是网络冻结和经验回放,判定完后取值函数最大的动作为决策结果进行输出。

2.根据权利要求1所述的一种边缘计算环境下基于无人机群的任务卸载智能决策方法,其特征在于,

所述步骤(1)中采集环境信息包括采集无人机群的任务信息、无人机群的状态以及云中心的情况。

3.根据权利要求1或2所述的一种边缘计算环境下基于无人机群的任务卸载智能决策方法,其特征在于,

所述方法采用任务卸载智能决策模型,模型共分为两层,其中,内层是传统的卸载决策模型,负责接收工作流并训练、决策给出最终的卸载决定;外层是元学习部分,负责当决策系统的环境发生变化时,它可以调整内层系统中神经网络的参数,使得系统可以快速适应新环境,用很少的训练量就可以学习的很好。

4.根据权利要求3所述的一种边缘计算环境下基于无人机群的任务卸载智能决策方法,其特征在于,

模型采集任务数据和环境数据,然后外层模型判别环境是否发生变化,若发生环境变化将调整初始参数;之后输入到内层模型中,来检索是否存在相似任务,若存在相似任务则直接输出已有决策结果,否则将任务状态输入到内层模型的神经网络中进行计算,求出的结果通过网络冻结来求出损失函数进而更新网络参数;将值函数最大的动作设定为决策结果输入到记忆机制里,同时作为结果输出。

一种边缘计算环境下基于无人机群的任务卸载智能决策方法

技术领域

[0001] 本发明涉及无人机群任务卸载决策技术领域，尤其涉及一种边缘计算环境下基于无人机群的任务卸载智能决策方法。

背景技术

[0002] 近年来，随着5G与无人机技术的日渐成熟，无人机群被越来越广泛的应用。通过与移动通信的结合，凭借其不受空间影响、响应能力强等特点，无人机群可以提供图像采集、信息传输等多项应用。然而，有限的计算能力和无人机缓存大小阻碍了其移动应用程序的应用，并导致大量的计算处理时间。此外，在无人机上进行任务计算会使得移动设备能耗增加，电池寿命缩短，降低无人机的使用时间。

[0003] 移动边缘计算是一种新兴的计算范式，可通过靠近无人机群的边缘服务器，将无人机群与云计算中心进行连接，形成“无人机-边缘-云”计算环境下这一任务计算卸载模式，如图1所示。其中，云中心负责为执行移动应用程序提供弹性和按需的计算资源，边缘服务器负责决策哪些无人机群的计算任务需要卸载以及提供有限量的计算资源。这样无人机群所产生的计算任务先经边缘服务器进行任务卸载决策，再确定由哪一部分来进行计算处理。从而可以有效缓解无人机群计算能力差的缺陷。

[0004] 无人机群在“无人机-边缘-云”计算环境下的任务卸载决策：即当无人机产生任务需求时，它可将计算任务迁移到边缘服务器或云服务器来进行计算。目前的卸载决策主要分为传统系统和智能系统。其中，传统系统多采用一些启发式算法，对于复杂问题往往无法求解，且计算量大，需要消耗大量计算资源；相反，最近兴起的智能算法可以有效的解决该类问题，通过引入深层神经网络可以使决策系统自动学习到合理可行的决策方案，但是仍存在学习速度慢，可移植性差等问题。

发明内容

[0005] 为了解决上述问题，本申请的目的即是提供一种边缘计算环境下一种基于无人机群的任务卸载智能决策方法。

[0006] 为实现本发明的目的，本发明提供了一种边缘计算环境下基于无人机群的任务卸载智能决策方法，包括如下步骤：

[0007] (1)采集环境信息；

[0008] (2)进行元学习，若发现边缘服务器或云中心的环境发生变化将修改模型的初始参数；

[0009] (3)进行检索机制与强化学习，其中检索机制负责检索之前是否存在相似任务，若存在，直接输出决策结果；若不存在，进行强化学习，强化学习负责训练和判定整个强化学习系统，其中用到的两大模块是网络冻结和经验回放，判定完后取值函数最大的动作为决策结果进行输出。

[0010] 其中，

[0011] 所述步骤(1)中采集环境信息包括采集无人机群的任务信息、无人机群的状态以及云中心的情况。

[0012] 其中，

[0013] 所述方法采用任务卸载智能决策模型，模型共分为两层，其中，内层是传统的卸载决策模型，负责接收工作流并训练、决策给出最终的卸载决定；外层是元学习部分，负责当决策系统的环境发生变化时，它可以调整内层系统中神经网络的参数，使得系统可以快速适应新环境，用很少的训练量就可以学习的很好。

[0014] 其中，

[0015] 模型采集任务数据和环境数据，然后外层模型判别环境是否发生变化，若发生环境变化将调整初始参数；之后输入到内层模型中，来检索是否存在相似任务，若存在相似任务则直接输出已有决策结果，否则将任务状态输入到内层模型的神经网络中进行计算，求出的结果通过网络冻结来求出损失函数进而更新网络参数；将值函数最大的动作设定为决策结果输入到记忆机制里，同时作为结果输出。

[0016] 与现有技术相比，本发明的有益效果为，

[0017] 针对传统方法计算量大，面对复杂问题无法求解的问题，本方案采用了智能算法，在强化学习的模型上加入了深度网络，可以针对具有相关性的复杂工作流进行卸载决策。

[0018] 针对目前已有的智能算法的训练速度慢，可移植性差的缺点，本文引入了元学习的算法，引入了外层模型，观察模型所在环境是否发生变化，若发生变化则调整模型的初始参数，减去了模型初始参数从随机数开始学习的过程。使得模型可以快速适应新的环境，从而增强了模型的适应能力。

[0019] 针对无人机群环境下任务往往重复或相似的情况，本文引入了记忆功能，模型在收到任务信息后会检查过往任务中是否存在相似任务，若存在则直接输出当时的决策结果而不必进入神经网络，从而使得模型可以针对相似的任务迅速给出决策，减少了系统的计算量，增加了模型的决策速度。

附图说明

[0020] 图1为无人机群在无人机-边缘-云环境下的任务卸载示意图；

[0021] 图2为本申请模型总体框架示意图；

[0022] 图3为本申请内层模型流程图；

[0023] 图4为本申请模型的流程图；

[0024] 图5为本申请模型逻辑框图。

具体实施方式

[0025] 以下结合附图和具体实施例对本发明作进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0026] 需要注意的是，这里所使用的术语仅是为了描述具体实施方式，而非意图限制根据本申请的示例性实施方式。如在这里所使用的，除非上下文另外明确指出，否则单数形式也意图包括复数形式，此外，还应当理解的是，当在本说明书中使用属于“包含”和/或“包括”时，其指明存在特征、步骤、操作、部件或者模块、组件和/或它们的组合。

[0027] 需要说明的是，在不冲突的情况下，本申请中的实施例及实施例中的特征可以相互组合。

[0028] 本申请提出的任务卸载智能决策模型总体框架如图2所示。

[0029] 从总体上看，该模型共分为两层，其中，内层是传统的卸载决策模型，负责接收工作流并训练、决策给出最终的卸载决定(这期间外层不参与)；外层是元学习部分，负责当决策系统的环境发生变化时(如边缘服务器的性能或者带宽改变)时，它可以调整内层系统中神经网络的参数，使得系统可以快速适应新环境，用很少的训练量就可以学习的很好。

[0030] 其中：

[0031] 1.内层模型

[0032] 内层模型主要负责对于输入的任务信息进行判断，并通过计算各种动作空间中的最大值函数来给出决策，同时内层模型里面嵌入记忆机制可以帮助训练模型和检索相似任务。任务信息进入决策系统后，先经过记忆机制，检索是否有相似任务，若有则直接输出当时的决策结果，减少计算量。没有的话则进入神经网络，进行Q学习(Q-Learning)，得出的判别结果按最高价值进行选择，选取完后输出结果，同时将任务与其结论存到记忆机制里，以便进行训练和以后的检索。

[0033] 1.1参数假设

[0034] a)状态s：对环境的描述，在本方案里，用工作流中的任务量v和各任务间的数据通信量e表示，决策模型将根据状态s来求出决策。

[0035] b)动作a：对决策模型可选择的决策的描述，在本方案中动作有三种：任务不卸载(即在本地执行)、任务卸载到边缘进行计算、任务卸载到云端进行计算。

[0036] c)策略$\pi(a|s)$：是模型根据环境状态s来决定下一步的动作a的函数。

[0037] d)状态转移概率$p(s'|s,a)$：模型根据当前状态s做出一个动作a之后，环境在下一个时刻转变为状态s′的概率。

[0038] e)及时奖励$r(s,a,s')$：及时奖励是一个标量函数，即模型根据当前状态s做出动作a之后，环境会反馈给模型一个奖励，这个奖励也经常和下一个时刻的状态s′有关。

[0039] f)神经网络NN：本模型是通过神经网络来进行动作空间值函数的计算。其中NN_{env}表示目标神经网络，其参数实时更新。NN_{target}表示的是冻结神经网络，它的结构与NN_{env}一致，但是其参数是间断更新，负责消除任务间的关联性。

[0040] g)任务记录Φ[S,a,R,S′]：当模型决策完毕后要将任务记录存至记忆单元，其中S表示原状态，a表示所选择的动作，R表示所获得的及时奖励，S′表示执行动作后的状态。

[0041] h)状态值$V_{\pi}(S)$：表示从状态s开始，执行策略π得到的期望总汇报：

[0042] $$V^{\pi}(S) = E_{\tau\sim p(\tau)}[\sum_{t=0}^{T-1}\gamma^{t}r_{t+1}|\tau_{s_0} = s] \qquad (1)$$

[0043] 由于在本模型中，我们处理的是马尔科夫过程，所以$V^{\pi}(S)$可展开成贝尔曼方程：

[0044] $$V^{\pi}(S) = E_{a\sim\pi(a|s)}E_{s'\sim p(s'|s,a)}[r(s,a,s') + \gamma V^{\pi}(s')] \qquad (2)$$

[0045] 该方程表示当前状态的值函数可以通过下个状态的值函数来计算。

[0046] i)状态-动作值函数$Q^{\pi}(s,a)$：表示初始状态为s并进行动作a，然后执行策略π得到的期望总汇报：

[0047] $$Q^{\pi}(s,a) = E_{s'\sim p(s'|s,a)}[r(s,a,s') + \gamma V^{\pi}(s')] \qquad (3)$$

[0048] 1.2内层模型的建立

[0049] 内层模型中有关强化学习的部分，本方案采用Q学习与神经网络结合的深度强化学习方法。

[0050] Q学习部分：Q学习算法是一个异策略的时序差分学习算法。在Q学习中，Q函数的估计方法为：

[0051] $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_a Q(s',a') - Q(s,a))$ (4)

[0052] 相当于让Q(s,a)直接去估计最优状态值函数Q*(s,a)。

[0053] 神经网络部分：神经网络部分则采用两个结构相同、参数不同的神经网络，其中一个是冻结目标网络，即在一个时间段内固定目标中的参数，来提高模型的稳定性。

[0054] 除了参数更新算法，本方案还建立了一个记忆机制，即当任务决策完后，会把任务的状态、选择结果储存在记忆机制内，这个措施有两个作用：一是方便检索，当有新任务输入后，将检索有无与其相近的任务，若有的话直接输出当时的决策结果，提高决策速度；二是作为经验回放，通过构建一个经验池来去除数据的相关性，避免模型陷入局部最优。训练时，随机从经验池中抽取样布来进行训练。这样，也可以就打破了和相邻训练样本的相似性。提高训练速度。避免模型陷入局部最优。

[0055] 如图3为内层模型流程图，无人机群将任务输入内层模型，模型首先通过检索机制检索以往有无相似任务，若有的话直接输出当时的结果。否则进入决策模型，在决策模型中将任务的状态输入神经网络NN_{env}、NN_{target}，得出各个动作的值函数，选取最大值函数的动作a_{max}进行输出，与此同时通过奖励函数和网络冻结来更新神经网络中的参数。得到输出动作后，将该任务的状态和动作输入记忆机制来进行经验回放的学习和检索功能，当在经验回放中得到新的动作时，将会替换原有动作以保证决策的合理性。

[0056] 2.外层模型的建立

[0057] 外层模型主要为了在环境发生变化时可以及时更新内层模型中的初始参数以保证模型可以快速适应新的环境。

[0058] 在本方案中，我们假设环境变化后任务集为Γ其中第i个任务表示为Γ_i，内层网络中的神经网络采用常规的梯度下降来更新参数，则外层模型对于内层模型的初始参数更新算法为：

[0059] $$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\Gamma_i \sim p(\Gamma)} \mathcal{L}_{\Gamma_i}(f_{\theta'_i}) \qquad (5)$$

[0060] 在下面给出卸载决策模型训练算法：

算法 1：卸载决策模型训练算法：

输入：环境集Γ，状态空间 S，动作空间 A，折扣率 r，学习率 a

1. **for** i:n in Γ：
2. $\theta \leftarrow \theta - \beta\nabla_{\theta}\sum_{\Gamma_i \sim p(\Gamma)} \mathcal{L}_{\Gamma_i}(f_{\theta'_i})$更新参数
3. **Endif**
4. **repeat**
5. 初始化起始状态 s
6. **repeat**
7. 在状态 s,选择动作 a
8. 执行动作 a，观察环境，得到及时奖励 r 和新的状态 s'；
9. 将 s, a, r, s'放入记忆机制中：
10. $$y=\begin{cases} rr, & ss' \text{为终止状态,} \\ rr + \gamma \max_{a'} Q_{\hat{\theta}}(ss', a'), & \text{否则} \end{cases}$$
11. 以$(Y - Q_{\theta}(ss, aa))^2$为损失函数训练 Q 网络。
12. $s \leftarrow s'$;
13. 每隔 C 步，$\hat{\theta} \leftarrow \theta$;
14. **until** s 为终止状态
15. **until**$\forall s, a, Q_{\theta}(s, a)$收敛

输出 Q 网络

[0061]

[0062] 任务卸载智能决策模型的流程图如图4所示。

[0063] 整个模型的流程为：

[0064] 先采集环境信息，包括无人机群的任务信息、无人机群的状态以及云中心的情况等。然后进行元学习，若发现边缘服务器或云中心的环境发生变化将修改模型的初始参数。元学习之后将进行检索机制与强化学习。其中检索机制负责检索之前是否存在相似任务，有的话直接输出决策结果。强化学习负责训练和判定整个强化学习系统，其中用到的两大模块是网络冻结和经验回放。判定完后取值函数最大的动作为决策结果进行输出。

[0065] 任务卸载智能决策模型的逻辑框图的内容如图5所示。首先，模型采集任务数据和环境数据，然后外层模型判别环境是否发生变化，若发生环境变化将调整初始参数。之后输入到内层模型中，来检索是否存在相似任务，若存在相似任务则直接输出已有决策结果，否则将任务状态输入到内层模型的神经网络中进行计算，求出的结果通过网络冻结来求出损失函数进而更新网络参数。将值函数最大的动作设定为决策结果输入到记忆机制里，同时作为结果输出。

[0066] 针对传统方法计算量大，面对复杂问题无法求解的问题，本方案采用了智能算法，在强化学习的模型上加入了深度网络，可以针对具有相关性的复杂工作流进行卸载决策。

[0067] 针对目前已有的智能算法的训练速度慢，可移植性差的缺点，本文引入了元学习的算法，引入了外层模型，观察模型所在环境是否发生变化，若发生变化则调整模型的初始参数，减去了模型初始参数从随机数开始学习的过程。使得模型可以快速适应新的环境，从

而增强了模型的适应能力。

[0068] 针对无人机群环境下任务往往重复或相似的情况，本文引入了记忆功能，模型在收到任务信息后会检查过往任务中是否存在相似任务，若存在则直接输出当时的决策结果而不必进入神经网络，从而使得模型可以针对相似的任务迅速给出决策，减少了系统的计算量，增加了模型的决策速度。

[0069] 以上所述仅是本发明的优选实施方式，应当指出的是，对于本技术领域的普通技术人员来说，在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

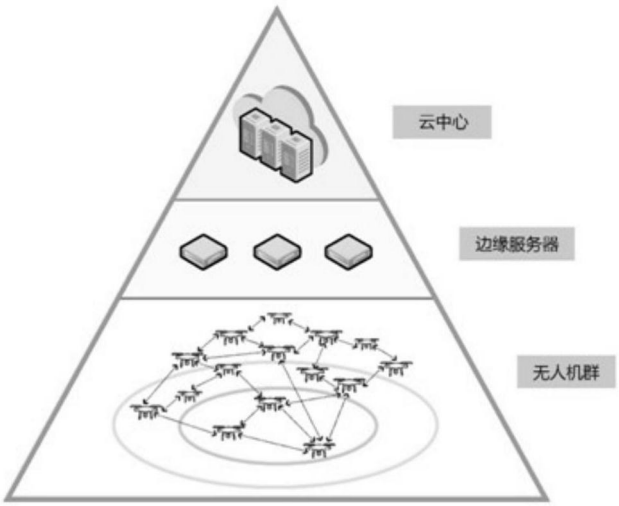


图1

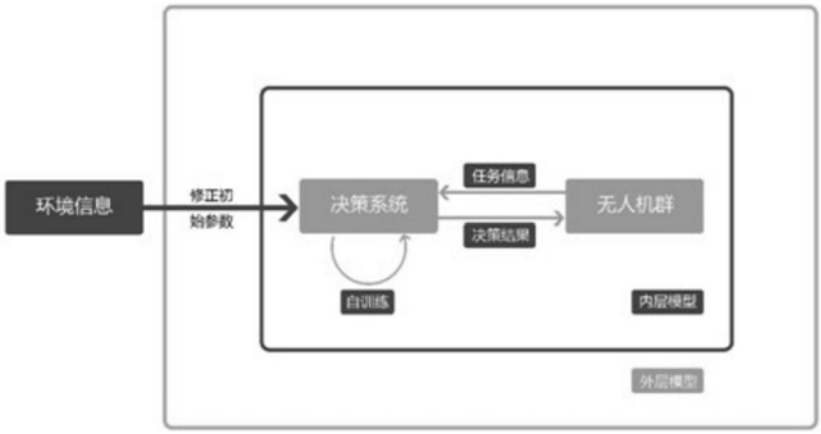


图2

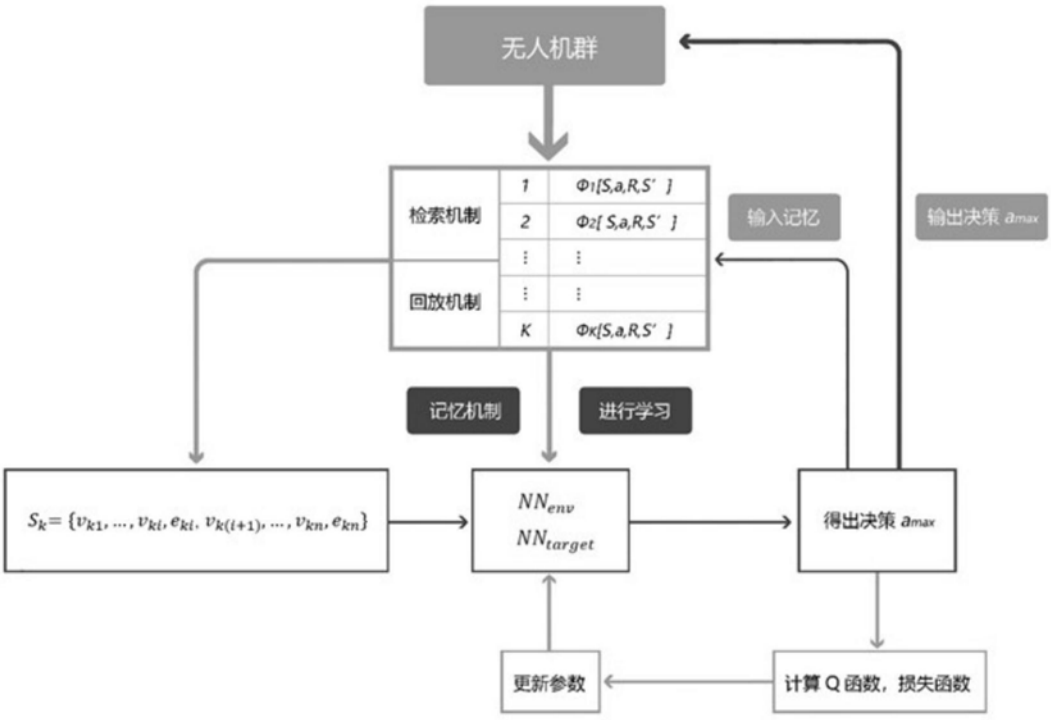

无人机群
检索机制
回放机制
1
2
K
Φ1[S,a,R,S']
Φ2[S,a,R,S']
ΦK[S,a,R,S']
输入记忆
输出决策 amax
记忆机制
进行学习
$S_k = \{v_{k1}, \cdots, v_{ki}, e_{ki}, v_{k(i+1)}, \cdots, v_{kn}, e_{kn}\}$
NN_{env}
NN_{target}
得出决策 amax
更新参数
计算 Q 函数，损失函数

图3

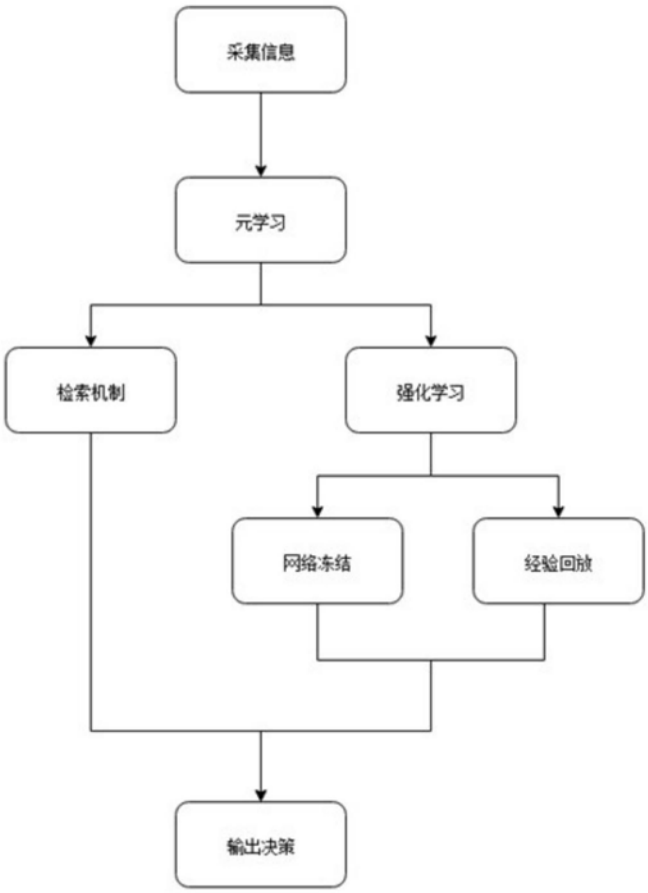


图4

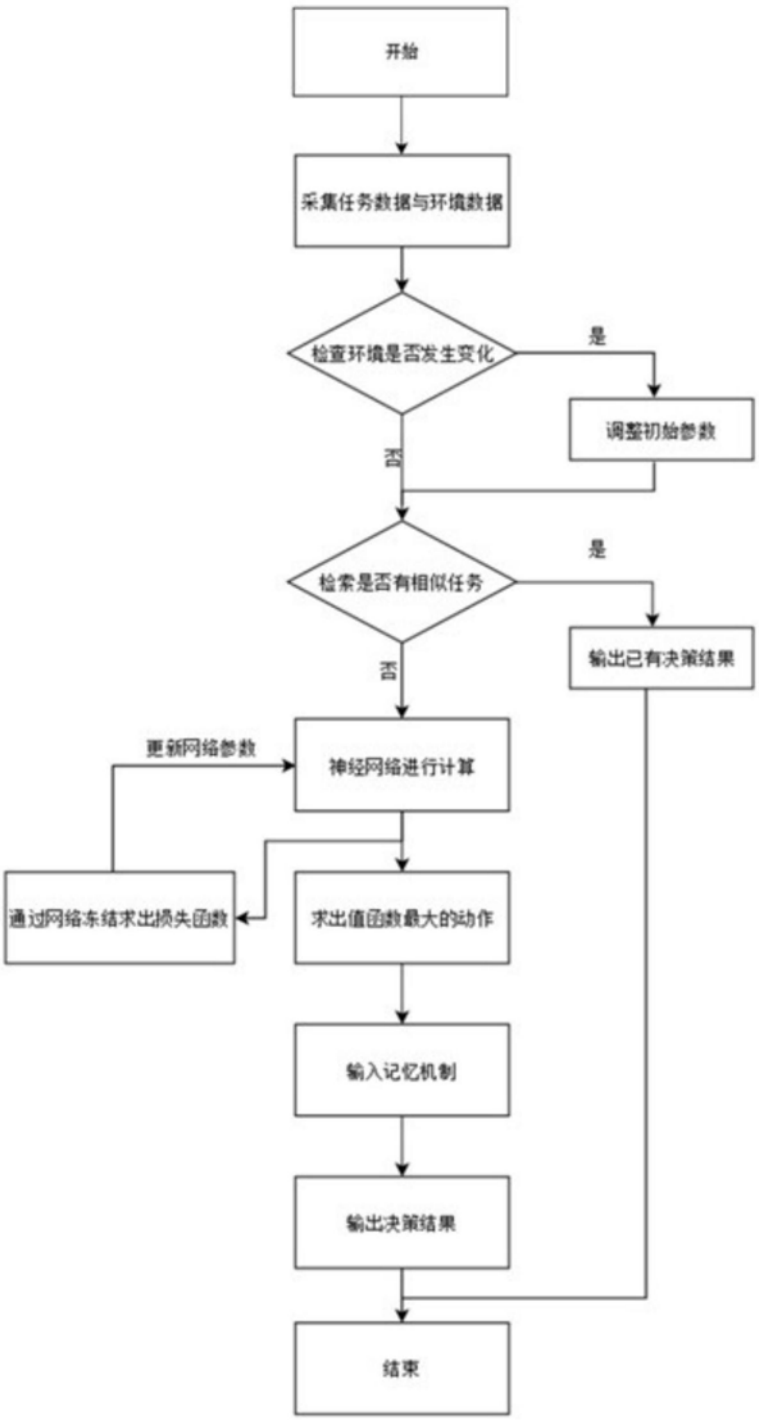

开始
采集任务数据与环境数据
检查环境是否发生变化
是
调整初始参数
否
检索是否有相似任务
是
输出已有决策结果
否
更新网络参数
神经网络进行计算
通过网络冻结求出损失函数
求出值函数最大的动作
输入记忆机制
输出决策结果
结束

图5