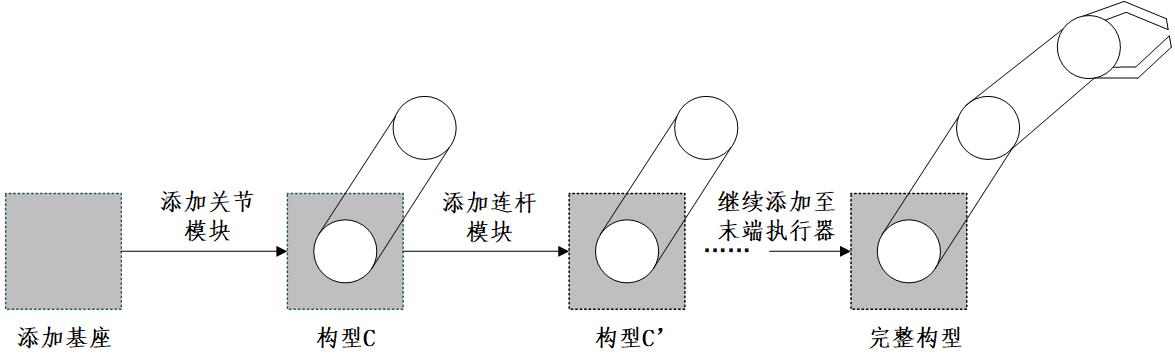
### 算法设计

串联机械臂重构过程如图所示。在该过程中，首先添加基座模块，通过向基座模块上添加新的关节模块形成构型C，通过向构型C中继续添加构型库中的模块形成新构型C’，并且只以末端执行器模块结束的构型定义为完整构型。



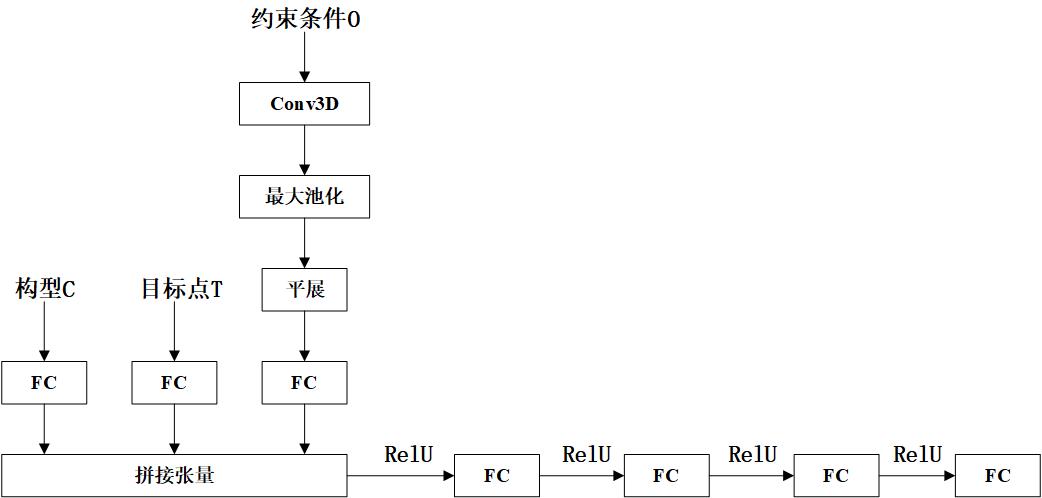
模块化串联机械臂的构型设计问题可以用有限马尔可夫决策过程来表述，串联机械臂在每个时刻，强化学习智能体选择的动作是模块的索引值，将对应的模块添加到构型C中。状态中变化的部分仅是构型C，因此下一个时刻的状态仅取决于上一个时刻状态和添加模块的动作。此时产生新的状态，包含新的构型C'和一个来自环境的奖励。动作空间A由机器人模块库定义，状态空间S由构型集合定义。

累计回报公式为，r为回报函数，γ为折扣系数⽤来计算累积回报R，其中折扣系数，为策略函数。算法的核心使用强化学习来估计最优状态-动作值函数，其能够通过贝尔曼最优方程给出定义，



本文旨在寻找机械臂在不同任务下的最佳构型，使其能够准确到达指定任务点。为实现此目标，需要以具体任务驱动构型设计和优化，并通过目标函数评估构型优劣，包括最少模块数、末端可达性和最轻质量等指标。同时，设置约束条件，要求机器人在操作过程中不能与障碍物或自身发生碰撞。

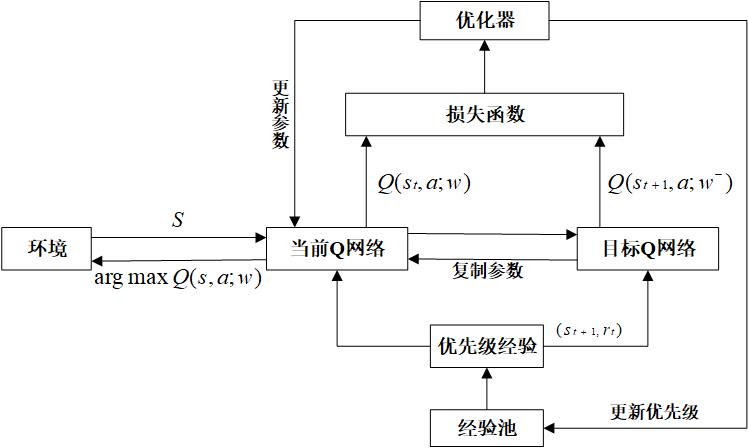
在训练之前需定义神经网络的输入，神经网络的输入即影响构型评估Q值的元素。以构型C、工作空间 环境中障碍物O为神经网络的输入，为了简化环境中障碍物的描述，将空间进行网格划分，其中每一个顶点的数值（0或1）表示是否存在障碍物。定义为障碍物矩阵，障碍物定义为立方体边长为。图（）描述了神经网络的结构。



#### 3.1带PER的DDQN算法

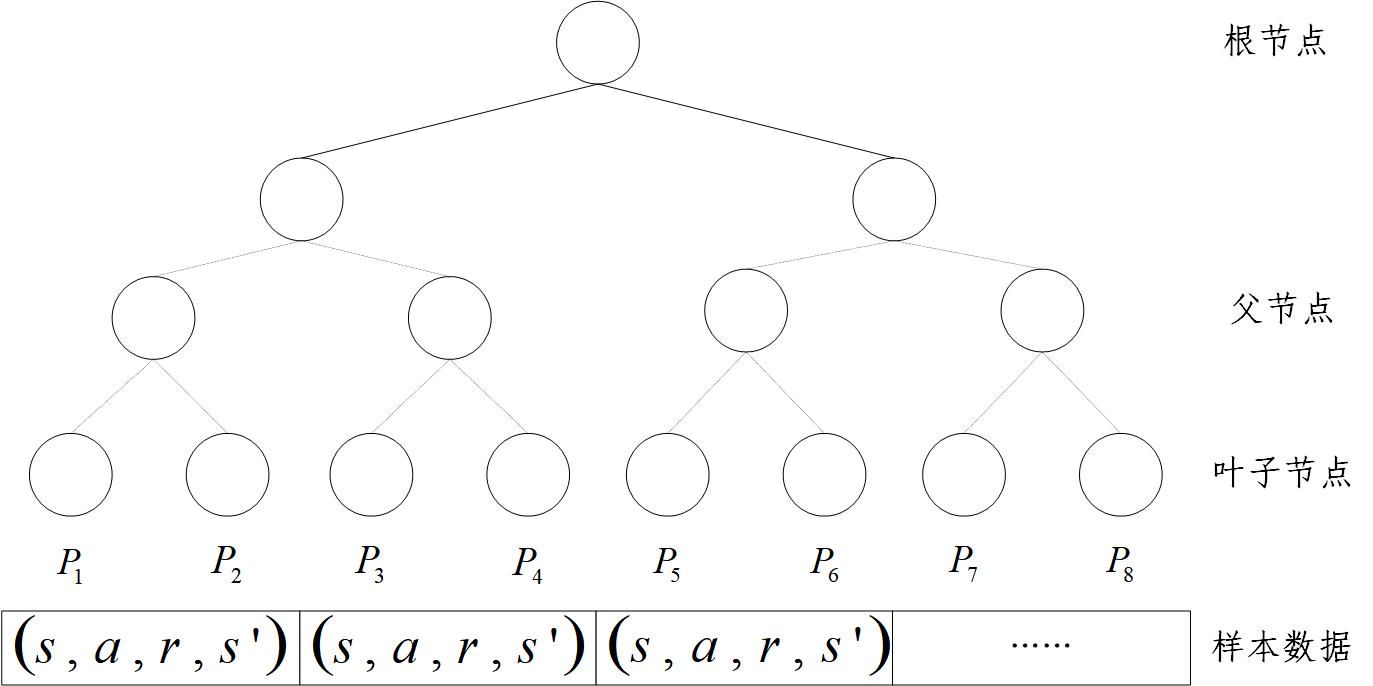
选择最优构型进行模块化机械臂重构是一个多任务问题，需要同时解决末端可达性、模块运动学、最少模块数以及最轻质量等问题。只有在综合考虑这些因素后，才能确定适用于特定应用的最佳模块化机械臂构型。本文采用了一种基于Double DQN的模块化机械臂重构算法来解决上述问题。Double DQN算法通过将在线网络和目标网络分离，目标Q值的计算与动作选择解耦，通过使用目网络来计算当前状态下的最优动作，有效减少了过度估计的问题。同时利用优先经验重放（PER）减少学习所需的经验量。在训练集中，迷你批次被赋予优先级权重，这有助于减少训练过程中的迭代次数和模型的训练时间。该模型如图（）所示。

通过采用带有优先经验回放的Double DQN算法，可以解决模块化机械臂重构中的多任务问题。本文利用该算法提高模块化机械臂重构的效果，并帮助实现更优的构型选择和性能提升。



#### 3.2基于SumTree的优先经验回放

##### 传统的经验回放机制阻断了训练样本的相关行，均匀采样的方式不利于算法的收敛。另外，传统的采样⽅法需要对所有样本排序，这⾮常消耗算⼒，为了更好地采样，本文采用基于SumTree的优先经验回放方法来进行求解。SumTree 是一种特殊的二叉树数据结构，如图所示。SumTree 中一共有四层节点结构，最顶部的节点称之为根节点，中间两行称之为父节点，最底层一行称之为叶子节点。所有经验样本的数据都是储存在叶子节点，叶子节点还会存储样本的优先级。



SumTree 采样主要是根据优先级来对样本进行训练，优先级取决于时序差分误差的大小，TD误差的值越大说明神经网络的反向传播作用越强，TD误差越大则优先级越高。经验的优先级及其选择的概率如下：





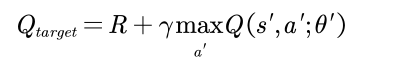
让抽样概率pt正比于的绝对值，为TD误差，为一个很小的数，防止概率等于0。是采样经验被选择的概率；是控制均匀采样和贪婪采样中采样偏好的超参数；且。当为0时，采样是均匀的；当它为1时，采样是贪婪的。

当采⽤优先回放的概率分布采样时，动作值函数的估计值是⼀个有偏估计。因为采样分布与动作值函数的分布是两个完全不同的分布，为了矫正这个偏差，因此需要乘以⼀个重要性采样系数



#### 3.3 基于DDQN的模块重构算法

DQN（Deep Q-Network）是一种主流且广泛应用的深度强化学习算法。它在推出的决策能力，能够高维连续状态和动作空间的问题。DQN使用两个具有相同结构但参数不同的神经网络。一个网络被用来实时更新参数，而另一个网络则用于更新目标Q值。目标 Q 值的更新公式为：



然而，普通的DQN算法是从目标Q网络中寻找各个动作最大的Q值，这样会大概率选择过估计的价值，进而导致的过估计问题。为了解决这个问题，Deepmind团队在2015年提出了DDQN，通过独立训练两个独立的Q网络来解决此问题。Double DQN引入了两个独立的神经网络（与DQN一样也是结构相同参数不同）：一个用于选择动作（动作选择网络），另一个用于评估所选择动作的价值（目标网络）。在计算目标Q值时，不再直接使用目标网络来选择下一个状态的最大动作值，而是使用动作选择网络来选择动作：



##### 将这个动作作为计算目标 Q值的动作，用目标网络完成最优动作的选择工作。Double DQN 中目标Q的计算如式（）所示：



##### 为了获得指定任务下的最优构型，采用Double DQN的模块化机械臂重构方法，通过输入当前构型的状态信息、约束信息和奖励信号，使用DDQN输出的Q值作评价构型优劣的指标。为评估机械臂构型的优劣，我们采用以下指标和函数：

1. 目标点可达性

本文将任务抽象为空间中的一系列位姿的集合，机器人末端执行器（end effector，ee）需要完成的任务可以通过工作空间目标（位姿）的集合来描述，其中表示末端执行器在空间中的坐标，表示末端执行器的方向。机械臂是从一组个模块中组合成的，模块的索引。选择一个模块序列C，为了评估构型是否能够达到给定目标，根据机器人的关节精度设置了位置误差和姿态误差，定义和表示给定目标下的逆运动学解对应正运动学的输出位姿。针对指定位姿下构型C的“可达性”函数由式给出定义：



1. 构型状态

当添加的模块是末端执行器时，智能体达到终止状态，环境返回的奖励按式中的“可达性”函数给出，而“可达性”评估只针对完整构型，对于在终止状态下未添加末端执行器的构型则返回-1的惩罚值（本文设置的最大模块数Nmax=15），



1. 约束条件

由于受机械结构和关节驱动能力等因素的影响，机械臂各关节的关节转角是有一定约束范围，关节转角范围的约束定义为：

qmin ≤ qi ≤ qmax

其中：qi为可重构机械臂的逆运动学解;qmin和 qmax分别为关节转角的下限和上限。

1. 最优构型

评估机械臂构型C的最终目标是找到能够达到目标的构型最优构型C\*。同时，希望机器人具有较少的驱动模块和更轻的质量。由此，目标函数F由式给出定义：

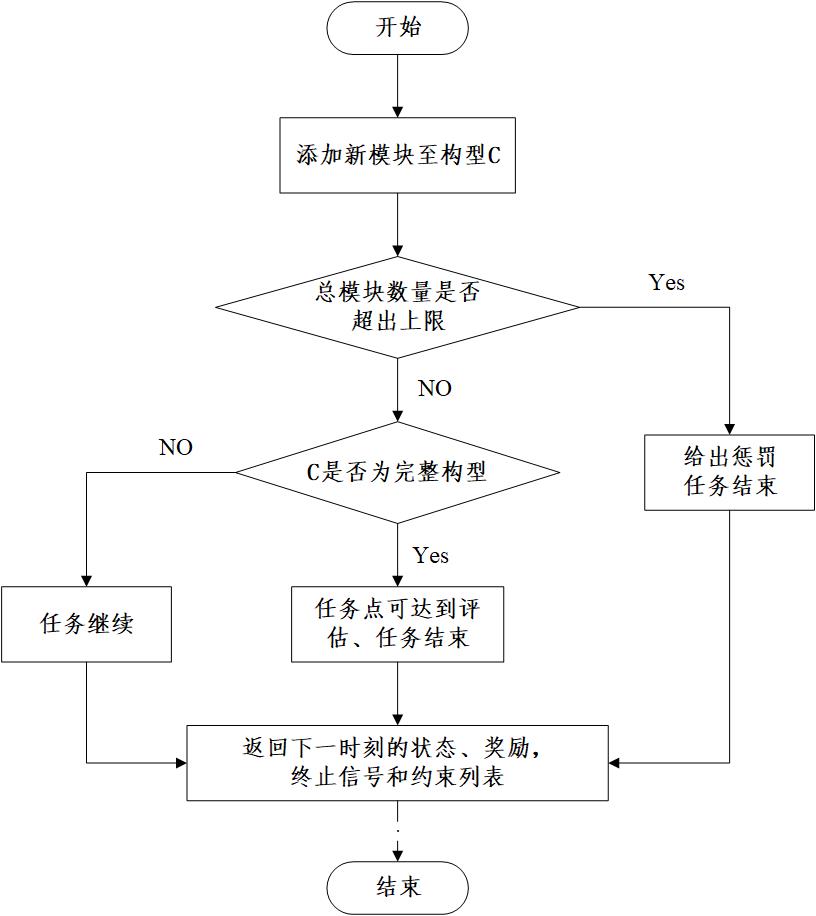


其中，表示给定机械臂构型C中驱动关节的数量，表示构型C的总质量，和分别是构型C的驱动模块数量和总质量的权重因子，以在多个目标之间进行权衡。最优构型C\*由式给出定义



最优构型设计流程图如图所示（）

将DDQN的输入样本通过神经网络前向传播，并将每个输出节点的值进行求和操作，前向传播求和即计算输出层所有节点的数值之和。求和操作可以帮助确定在给定状态下不同动作或策略的相对优劣程度。在每个迭代中，搜索算法会从搜索空间中提取具有最大数值的节点对应的策略作为当前状态下的最佳构型。如果此构型是一个完整的机器人，则进行逆运动学评估。否则，算法将继续扩展其子节点，通过DDQN计算其输出层节点数值之和，并将其添加到搜索空间中。这个过程不断循环，直到达到预设的最大搜索时长为止。



#### 3.4基于改进的DDQN模块化机械臂算法

本文提出了一种带有优先经验回放的DDQN模块化机械臂重构算法，并将SumTree作为经验回放池，该算法的伪代码如算法1所示：

|  |
| --- |
| 算法1：DDQN算法 |
| **输⼊：**状态(当前构型，目标位姿，约束条件)  **输出：**最优构型   1. :确定minibatch的⼤⼩，步⻓，回放周期，存储数据的总⼤⼩，常数； 2. :初始化回放记忆库; 3. :循环开始 4. :观测初试状态，选择动作:将添加新模块的动作作⽤于环境，观测环境返回的 5. :将数据存储到SumTree经验池中的叶子节点上，并根据样本的时序差分误差确定优先级； 6. :依次采集个样本; 7. :根据概率分布采样⼀个样本点; 8. :计算样本点的重要性权重; 9. :计算该样本点处的TD偏差 10. :更新该样本的优先级 11. :累积权重的改变量 12. :结束本样本的处理，采样下⼀个样本; 13. :采样并处理完个样本后更新权重值 14. :间歇性的复制新权重到目标网络中 15. :结束⼀次更新; 16. :根据新的策略选择下一个动作； 17. :利⽤新的动作作⽤于环境，得到新数据，进⼊新循环。 |