**游戏数值设定：**

1. 战场中天然矿石数量的最大值为500
2. 每支舰队的最大规模为1000艘飞船
3. 每支舰队所能携带矿石的最大数量为5000

**我们提取以下7个战场特征（构成21\*21\*7的张量）输入神经网络对Q值进行拟合：**

1. 所有位置的天然矿石数量（不包含舰队上的矿石）

2. 双方所有位置上舰队和船厂的规模（即包含船只的数量），分别用正数/负数表示我方/敌方舰队

3. 所有舰队当前的位置，分别用+1/-1表示我方/敌方舰队所在位置

4. 所有舰队所携带的矿石数量，不区分敌我双方

5. 所有船厂的位置以及该船厂回合最多造船数，分别用正数/负数表示我方/敌方船厂（该越大则说明船厂被控制了越久，对于玩家来说重要程度越高）

6. 敌方所有舰队下一步移动位置，用-1表示

7. 我方所有舰队下一步移动位置，用+1表示

**注：**对于特征1,2,4，采用将数值同时除以游戏数值设定最大值（这里为5000）的方法，将其映射到[0,1]区间或[-1,1]区间，以便于神经网络更好地提取特征进行学习。

**为简化模型，我们提取并总结出5套动作模式（平衡、攻击、采矿、扩张、防御），作为智能体选择的动作策略进行游戏对抗。**

**在以下五套动作策略中，为了避免每次都从相同的船厂开始动作指定，使用sample函数随机化船厂的顺序，并遍历所有船厂根据一系列条件对每个船厂的下一步动作进行指定：**

**平衡**(主要参考balanced.py)**：**

**评估位置(check\_location)：**计算特定位置附近区域（6\*6）的矿石总量，若该位置为己方船厂，则直接返回0。

**评估路径(check\_path)：**利用**math.pow(npv, steps) \* kore / (2 \* (dist\_a + dist\_b + 2))**公式来计算路径的预期收益。math.pow(npv, steps)用来计算未来收益折现到现在的因子，npv被设置为0.98，表示每一步的时间成本为2%，用于评估未来收益的当前价值，steps 是路径的长度；kore表示在路径上收集矿石的数量；2 \* (dist\_a + dist\_b + 2)表示路径的长度，dist\_a 和 dist\_b 是在每个方向上移动的步数。综上，在评估路径价值时综合考虑了收集资源的数量、时间折现及路径长度等多个因素。

——距离最近的敌方船厂规模小/距离近，若船厂船只充足，达到攻击舰队设定的阈值(100)，则**派出攻击舰队攻击距离最近的敌方船厂**；否则**在船厂中尽可能建造最多的船只**。

——剩余大量矿石且船厂造船能力强，若我方船只充足，数量足够建立船厂，则通过评估附近区域寻找最优的建厂位置，继而**派出部分舰队执行建立船厂任务**；否则**在原船厂中尽可能建造最多的船只**。

——船厂中船只数量多，通过模拟不同的飞行路线并评估路径，寻找预期价值最大的飞行计划，**派出21艘飞船形成舰队执行采矿任务**。

——若以上条件均不满足，则首先检查剩余的矿石数量是否足够船厂生成最多的飞船，如果满足，则先减去生成一艘飞船的成本，然后**再在船厂中尽可能建造最多的船只**。

注：在介绍另外四种动作策略前，为避免混淆，首先对动作与动作策略两个概念进行区分：动作是指单一船厂所执行的无需思考的单一指令，而策略是指玩家所控制的所有船厂经过思考评估后采取的一系列动作组合。例如，接招、化解、发功是中国传统武术中的基本动作，在战斗中单独使用往往收效甚微；但是，将这三个动作采用不同的策略组合在一起，往往能体现出四两拨千斤的惊人效果。

**我们小组在制定其他动作策略时，既考虑到了当前的战局状况，也考虑到了船厂自身的发育状态，从而避免其做出呆板、短视、机械化的决策。因此，我们设计的动作策略并非一一对应执行既定的动作，而采用了更加创新、灵活、多变的方式：首先，根据战局状况和船厂发育情况进行评估，设定最佳的战队规模与飞行计划；其次，设定执行四种动作的对应条件；最后，每种策略对应一个优先考虑的动作，在满足相应动作条件时，直接执行该动作；若无法满足，则依次考虑其他动作。**值得一提的是，在设定攻击策略时，考虑到攻击敌方船厂存在失败的风险，因此额外设定了多个严格的条件，以尽可能地提高攻击的成功率。以下为各个动作的详细介绍：

**评估飞行计划(check\_flight\_paths)：**通过迭代四个方向和搜索半径的组合(圆形，L型，长方形)，计算每个组合下的路径价值，计算飞船的平均收益，并依次来选择最佳的飞行路径和舰队规模**。**(没太看懂o(╥﹏╥)o)

**攻击动作(should\_attack)：**当同时满足以下条件——距离最近的敌方船厂规模小/距离近，若船厂船只数量达到攻击舰队设定的阈值(50)/剩余矿石足够再次造船，游戏时长足够(双方得到充分发育)/敌方船厂距离本方船厂过近形成威胁——则**派出攻击舰队攻击距离最近的敌方船厂;**若船只数量不足以形成攻击舰队，则**在原船厂中尽可能建造最多的船只**。

**采矿动作(should\_mine)：**当船厂内船只数量满足最佳舰队规模的要求时——则**根据之前评估的最佳飞行计划派出舰队执行采矿任务**。

**扩张动作(should\_build)：**当剩余矿石充足且船厂生产能力满足要求时——若当前船厂内船只数量足够建立新的船厂，则通过评估附近区域寻找最优的建厂位置，继而**派出部分舰队执行建立船厂任务**；否则**在原船厂中尽可能建造最多的船只**。

**防御动作(should\_defend)：**当本方船厂的附近区域出现敌方舰队，且敌方舰队船只数量充足，有击毁我方船厂的能力时——则**在原船厂中尽可能建造最多的船只**。

**备用策略：**若以上四种动作的触发条件均未满足，则采取以下两种备用策略：**在原船厂中尽可能建造最多的船只**；若我方仅剩一个船厂且舰队覆灭，若尚有剩余矿石，则**继续建造船只**，否则**将船厂中所有船只派出向北行驶**。

**模型建立：**我们小组使用create\_q\_model函数创建深度 Q 网络（DQN）模型，并用于强化学习中的游戏决策过程。函数返回一个 Keras 序贯模型，该模型包括输入层、卷积层、激活函数层、全连接层等，构成了一个卷积神经网络。输入层接受形状为 (21, 21, 7) 的张量数据，经过两个卷积层和激活函数层处理后，通过展平层将输出展平为一维张量，然后经过两个全连接层，最后一个全连接层的节点数为动作数5。