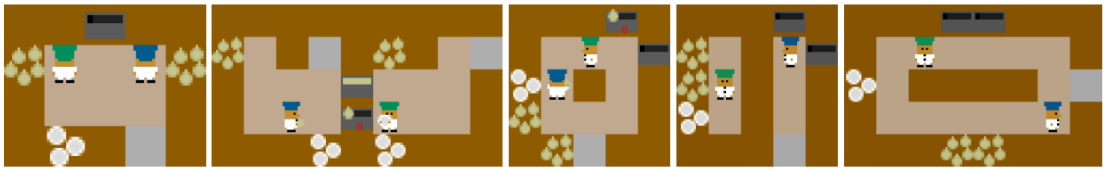
Overcooked 环境下 MAPPO 效果测试报告

一、 实验概述

MAPPO 是一种 on-policy 强化学习方法，在近期研究中表现出可和诸多 off-policy 方法媲美的效果。技术上， MAPPO 一般对所有智能体使用同一个 critic 网络，并共享 actor 网络的参数，实际上是 PPO 向多智能体情景的 CTDE 式推广。

Overcooked 作为一款合作性游戏，适合测试强化学习方法。On the Utility of Learning about Humans for Human-AI Coordination 一文研究了多种合作方式与训练方法的效果，并提供了接口完备的 Overcooked 环境。我们希望将 MAPPO 移植到 Overcooked 环境中，测试其在 self-play 合作方式下的性能。仿照该论文，我们也对 cramped room, asymmetric advantages, coordination ring, counter circuit 和 forced coordination 这五个场景进行训练。



二、实验细节

网络设置

由于 self-play，两个智能体的各自采取的动作由一套共享观测的 actor network 得到，并经过一个中心化的 critic network 反向传播进行训练。由于采用的是 Recurrent-MAPPO，我们需要以 RNN 作为网络主体，但也使用 MLP/CNN 来初步提取特征。理论上，只要状态编码足够理想，特征提取层数可以很少，这里我们只使用了一个线性层。而 RNN 的 recurrent layer 数目也只取了 1，即达到了最终的训练效果。

状态编码方式

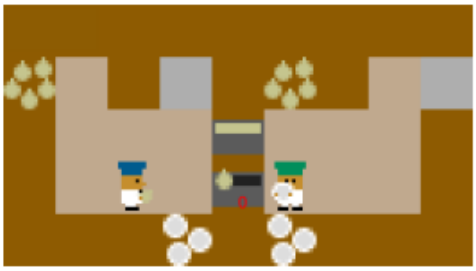
观测分为环境和 agent 状态两部分。环境包括每个 pot 里的洋葱数目、烹饪时间和各个 counter 上的物品类型。为了缩短长度，考虑到有效的输入应当是变量，我们不引入各目标点的位置信息。agent 状态包括节点的位置、朝向与所持物。我们对位置、所持物采用 one-hot 编码，对朝向使用 2bit 编码。由于本次实验的空间较小，可用一维编码+MLP 提取特征，之后输入到 RNN 中。

奖励设计

几种场景下的 reward 分配:

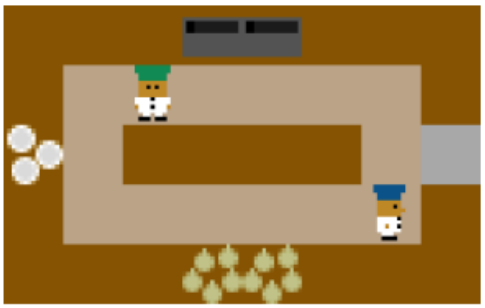
	Onion pickup	Placement in pot	Dish pickup	Soup Pickup	Delivery
Cramped Room	1	10	0	100	1000
Asymmetric Advantages	1	10	0	10	10
Coordination Ring	1	10	0	100	1000
Forced Coordination	0	1	0	10	100
Counter Circuit	1	10	0	100	1000

此次实验中我发现的一个简单有效的技巧是指数级奖励(exponential reward)设计。这个方法主要有两点好处。首先，在学习的前期，越靠后的动作越少被试到，因此一旦试到了就应该给充分大的 reward，加快训练速度。其次，本次实验的活动空间较小，因此只要 episode length 取得足够大（600），总有一定概率试出一两次后面的动作。利用 RNN 的有记忆性，若靠后的动作得到了满足，也就自然加固了前面步骤的训练结果，甚至可以减少前面 reward 的设置数目。通过此方法，我避免了引入 distance based reward，使得 reward 只和动作有关。并且经过测试，我发现甚至 dish pickup reward 也是可以去掉的。



不过这个方法也有一些问题，比如为了满足后面的动作，可能牺牲前面动作的质量，毕竟前面的 reward 太小了。如对 Asymmetric Advantages 而言，若简单沿用指数奖励，会导致只有一个 pot 得到使用。为此，我通过尝试加入了两个智能体获得 reward 的条件，即左侧智能体必须在有且仅有一个 ready pot 的条件下拿起 pot，而右侧智能体必须在已经有一个 cooking pot 的条件下加入 onion。同时，还需要控制两个 reward 相同，并减小 delivery reward，这样才使得结果较平稳地收敛于两个 pot 轮流工作的情况。

对 forced coordination 而言，引入 onion 或者 dish pickup reward 都是没有必要的，因为显然左侧 agent 很容易做出这些动作。因此我们的 reward 可以从 Placement in pot 开始设置。这里起重要作用的其实是状态编码里对 counter 上物品类型的标记，两个智能体在传递物品的时候需要这部分信息。



此外，对 agent 位置使用 one-hot 编码也是有必要的。在 counter circuit 场景里，起初尽管使用了 exponential reward，但仍然难以训得解。后来我将坐标输入换成了 one-hot 输入，即哪个位置上有 agent 哪个位置为 1 其他为 0，这样每个位置在网络中的贡献都是独立的。随后便快速训得了原论文中提到的隐藏合作方案。