2023.11.7周报

本周任务完成情况

1、科研学习：
了解mappo算法与coppo算法。

2、项目进度：
在超算下进行训练,尚未完成训练任务。

下述主要讲解mappo算法的简单训练流程：

代码总体流程
1）环境设置，设置智能体个数、动作空间维度、观测空间维度
2）初始化环境，将obs输入到actor网络生成action，将cent_obs输入到critic网络生成values
3）计算折扣奖励
4）开始训练，从buffer中抽样数据，计算actor的loss、critic的loss
5）保存模型，计算average episode rewards

其中，十分重要的是ppo_update函数的更新方式：

1）从buffer中抽样建立sample
2）将抽样的数据传递给rMAPPOPolicy.py中的evaluate_actions函数，得到values, action_log_probs, dist_entropy
3）计算actor的loss
4）计算critic的loss

```
    def ppo_update(self, sample, update_actor=True):
        """
        Update actor and critic networks.
        :param sample: (Tuple) contains data batch with which to update networks.
        :update_actor: (bool) whether to update actor network.

        :return value_loss: (torch.Tensor) value function loss.
        :return critic_grad_norm: (torch.Tensor) gradient norm from critic up9date.
        ;return policy_loss: (torch.Tensor) actor(policy) loss value.
        :return dist_entropy: (torch.Tensor) action entropies.
        :return actor_grad_norm: (torch.Tensor) gradient norm from actor update.
        :return imp_weights: (torch.Tensor) importance sampling weights.
        """
        share_obs_batch, obs_batch, rnn_states_batch, rnn_states_critic_batch,
actions_batch, \
        value_preds_batch, return_batch, masks_batch, active_masks_batch,
old_action_log_probs_batch, \
        adv_targ, available_actions_batch = sample #然后从buffer中采样数据，把线程、智能体的纬度全部降掉

        old_action_log_probs_batch =
check(old_action_log_probs_batch).to(**self.tpdv)
        adv_targ = check(adv_targ).to(**self.tpdv)
        value_preds_batch = check(value_preds_batch).to(**self.tpdv)
```

```
        return_batch = check(return_batch).to(**self.tpdv)
        active_masks_batch = check(active_masks_batch).to(**self.tpdv)

        # Reshape to do in a single forward pass for all steps
        values, action_log_probs, dist_entropy =
self.policy.evaluate_actions(share_obs_batch,

 obs_batch,

 rnn_states_batch,

 rnn_states_critic_batch,

 actions_batch,

 masks_batch,

 available_actions_batch,

 active_masks_batch)
        # actor update 计算actor的loss
        imp_weights = torch.exp(action_log_probs - old_action_log_probs_batch)

        surr1 = imp_weights * adv_targ
        surr2 = torch.clamp(imp_weights, 1.0 - self.clip_param, 1.0 +
self.clip_param) * adv_targ

        if self._use_policy_active_masks:
            policy_action_loss = (-torch.sum(torch.min(surr1, surr2),
                                             dim=-1,
                                             keepdim=True) *
active_masks_batch).sum() / active_masks_batch.sum()
        else:
            policy_action_loss = -torch.sum(torch.min(surr1, surr2), dim=-1,
keepdim=True).mean()

        policy_loss = policy_action_loss

        self.policy.actor_optimizer.zero_grad()

        if update_actor:
            (policy_loss - dist_entropy * self.entropy_coef).backward()

        if self._use_max_grad_norm:
            actor_grad_norm =
nn.utils.clip_grad_norm_(self.policy.actor.parameters(), self.max_grad_norm)
        else:
            actor_grad_norm = get_gard_norm(self.policy.actor.parameters())

        self.policy.actor_optimizer.step()

        # critic update 计算critic的loss
```

```
        value_loss = self.cal_value_loss(values, value_preds_batch, return_batch,
active_masks_batch)

        self.policy.critic_optimizer.zero_grad()

        (value_loss * self.value_loss_coef).backward()

        if self._use_max_grad_norm:
            critic_grad_norm =
nn.utils.clip_grad_norm_(self.policy.critic.parameters(), self.max_grad_norm)
        else:
            critic_grad_norm = get_gard_norm(self.policy.critic.parameters())

        self.policy.critic_optimizer.step()

        return value_loss, critic_grad_norm, policy_loss, dist_entropy,
actor_grad_norm, imp_weights
```

下周任务制定：

先考虑单智能体环境下的ppo原理，以便于更好的理解多智能体下延申出的各种ppo算法。