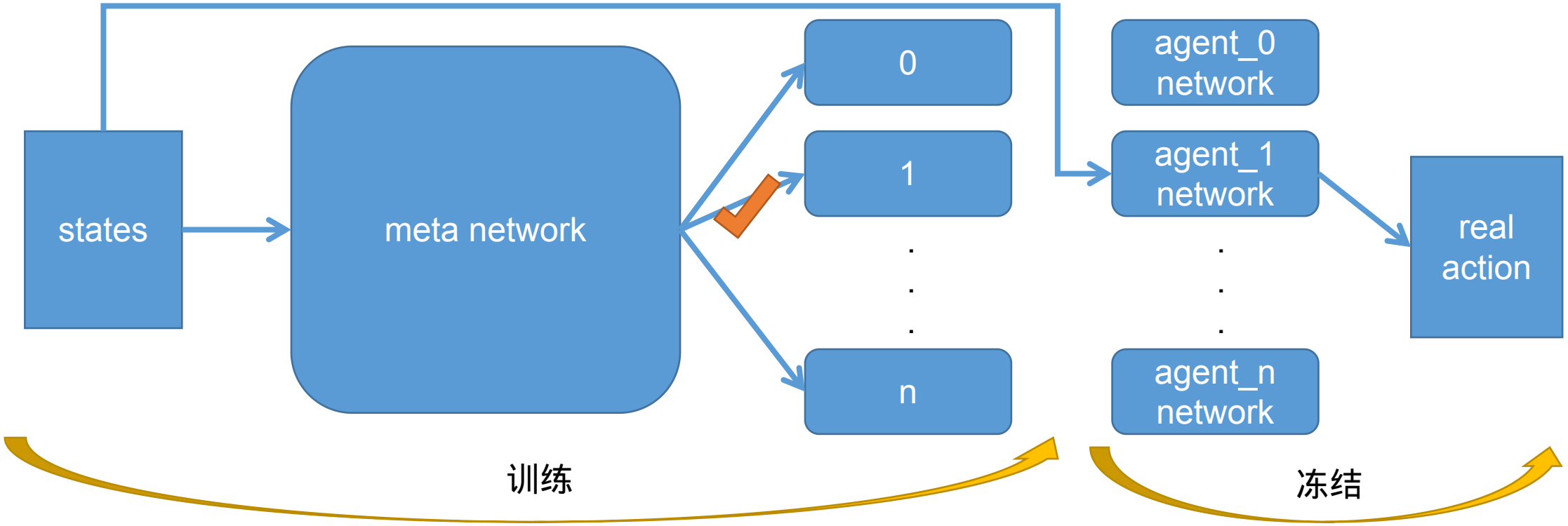
分治融合预实验

张麟睿

总体思路

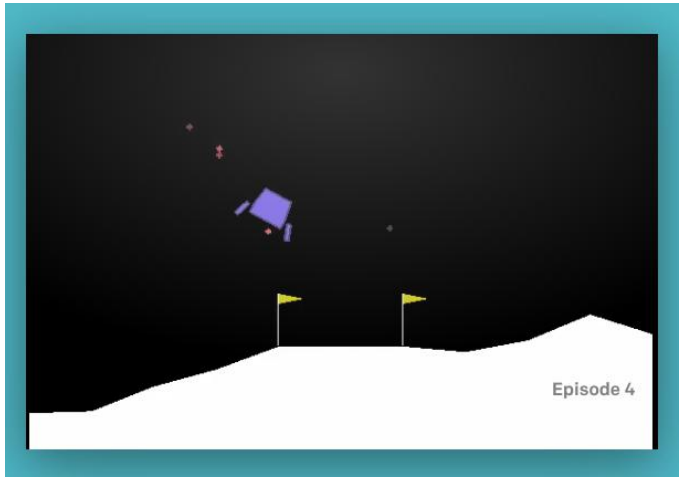
- 独立训练多个可以解决任务的agent，模拟其他家提供的不同策略
- 构建一个meta网络，负责选择agent。其中，meta网络的输入是和agent相仿的状态（此状态必须是所有agent需要状态的并集），输出是离散的，维度等于agent个数，实质是选择agent列表的index。
- 一旦输出index，并选择了对应agent，则真正执行的动作是：从状态中抽取该agent需要的state，并执行action = agent.select_action(state)得到动作，执行env.step(action)

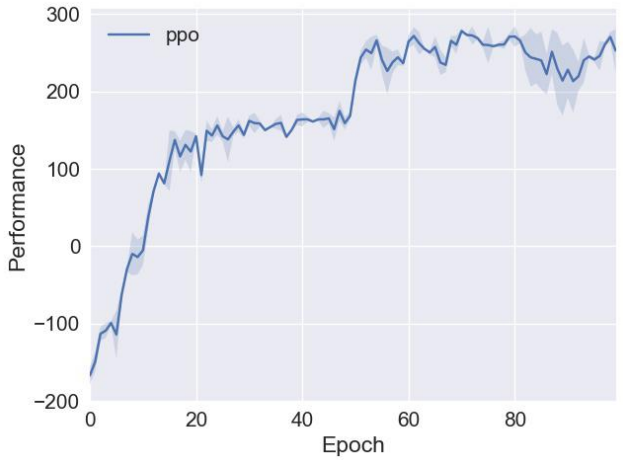


具体实现细节

- 任务选择

我选择了lunarlander环境（如下左图），该飞船有多个发动机控制，因此不同算法训练出来的控制策略可能会更不同。cartpole过于减单，担心不同策略基本上控制策略是一致的





正常来说，这个环境的学习曲线会收敛在300分左右（如上右图），我在每个agent达到200附近时中断训练，并保存此时的model

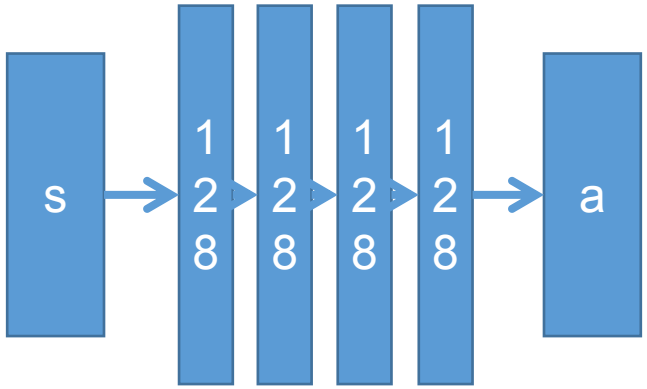
具体实现细节

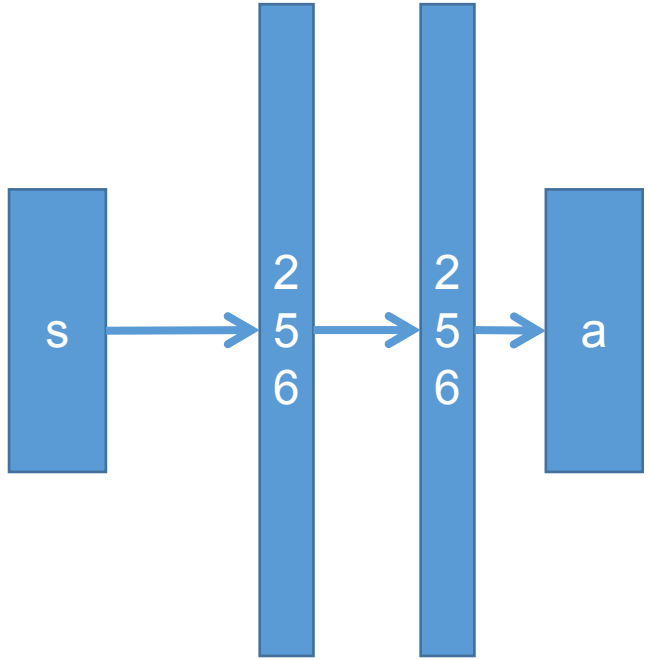
- agents设计

本次实验选择了2个agent的融合，他们都是用ppo训练，但使用不同的随机种子和网络结构

agent1使用2个隐藏层的MLP，每层神经元数256；

agent2使用4个隐藏层的MLP，每层神经元数128





具体实现细节

- 测试agents

1. 20轮agent1的平均测试表现192分，agent2是207分，多次测试发现，可能agent2的模型比agent1稍微好一点

2. 注意到DiffRate这个指标，代表在状态s下，两个agent选择不同动作的概率，大概在0.6-0.7左右，这说明目前这两个model是没有训练完全且风格不同的

AverageEpRet	192
StdEpRet	99.4
MaxEpRet	300
MinEpRet	-55.3
EpLen	338
DiffRate	0.692

AverageEpRet	207
StdEpRet	66.6
MaxEpRet	300
MinEpRet	116
EpLen	341
DiffRate	0.601

具体实现细节

- 测试使用未训练的meta网络的融合效果

现在我们测试使用没有训练过的meta网络选择agent执行动作的结果，

测试评分是187分，低于任何一个agent（多测试几次发现，有时候运气好会到220，但不会太好，而且方差很大）

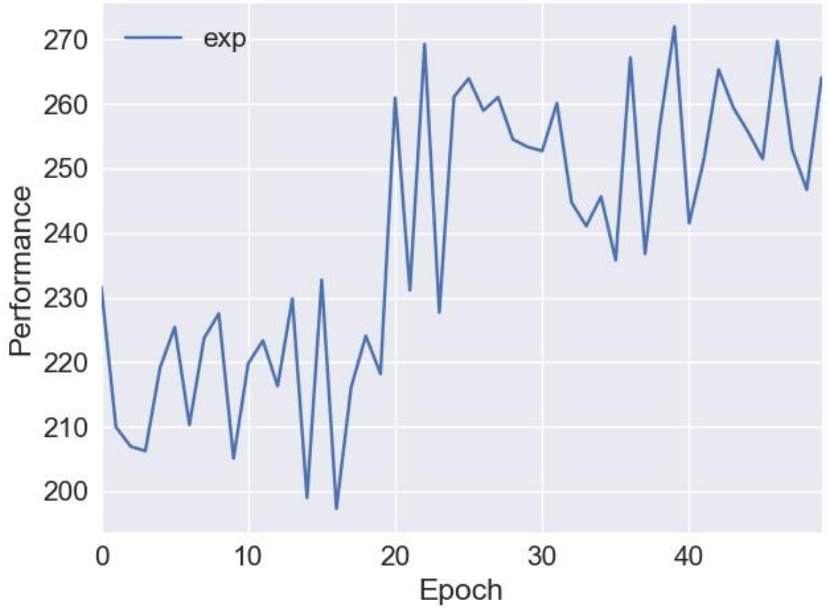
注意Action1Rate指标，代表选择agent1的概率，近似55开

AverageEpRet	187
StdEpRet	97.7
MaxEpRet	291
MinEpRet	-69
EpLen	326
DiffRate	0.67
Action1Rate	0.505

具体实现细节

- 训练meta网络的过程

我们依旧使用PPO，但这次只训练meta网络，所有agent全部冻结。
学习曲线如下，可以看到性能确实在提升



具体实现细节

- 测试使用训练过的meta网络的融合效果

最终测试表现268分，稳定超过agent1与agent2且方差较小，选择agent1的频率是0.147，符合我们认为agent2稍好于agent1的认知。

AverageEpRet	268
StdEpRet	17.8
MaxEpRet	294
MinEpRet	226
EpLen	253
Action1Rate	0.147

对照：agent1

AverageEpRet	192
StdEpRet	99.4
MaxEpRet	300
MinEpRet	-55.3
EpLen	338
DiffRate	0.692

对照：agent2

AverageEpRet	207
StdEpRet	66.6
MaxEpRet	300
MinEpRet	116
EpLen	341
DiffRate	0.601

实验结论

- 实验验证了在上文设定下算法的成功，这个方法看上去是有希望的
- 本实验没有使用固定多少帧只能使用一个策略，每个时刻都可以选择。目前看来，不需要固定。