PPO × Family 第五讲技术问题 QA

Q0： 有没有第五节课内容的大白话总结?

A0:

小节	算法要点	代码和实践要点
POMDP 概述	• POMDP 的定义和特点 • 从数据编码角度应对（叠帧） • 从网络设计角度应对（完美价值网络） • 从优化方法角度应对（N-step）	/
PPO + LSTM	• 整体网络架构设计 • 隐状态维护更新问题 • 反应型和记忆型决策 • 时序展开如何优化	准备统一的轨迹数据 RL 中应用 LSTM 的三重境界
PPO + Transformer	• Transformer + RL 的优劣势 • LayerNorm 的位置之争 • 门控（Gate）调节机制	Transformer 的记忆模块实现 memory len 环境对比测试

Q1: PPO + gtrxl 算法中，传递给训练阶段的数据是不是应该是 $(T, B, *)$ 的数据，并且这个 T 是大于 1 的？而普通的 PPO 算法中，T 是等于 1 的，也就是传递给训练阶段的是 $(B, *)$ 的数据，这个 1 维数据的每个元素包含 $(S_t, a_t, r_t, done_t, S_t + 1)$ 等信息？在 DI-engine 中 PPO 的实现似乎也只考虑了数据是 $(B, *)$ 的情况，如果要处理带时序的 $(T, B, *)$ 的数据，应该怎么改动呢?

A1: 是的，带时间序列模型，数据就是 $(T, B, *)$ 这样的维度，不带时间序列模型，T=1，但是这种情况通常省略，所以就是 $(B, *)$ 这样的数据。DI-engine 默认实现的 PPO 只能处理 $(B, *)$ 的数据，如果要实现处理带时序的 $(T, B, *)$ 的数据，可对此处 RunningMeanStd 源代码 进行修改。原来对 $(B, *)$，计算其 std 时有以下处理:

```
class RunningMeanStd(object):
    ...
    def update(self, x):
        """
        Overview:
            Update mean, variable, and count
        Arguments:
            - ``x``: the batch, shape (*,)
```

```
        """
        batch_mean = np.mean(x, axis=0)
        batch_var = np.var(x, axis=0)
        batch_count = x.shape[0]
        ...
```

要处理 $(T, B, *)$ 的数据，则可以相应更改为：

```
class RunningMeanStd(object):
    ...
    def update(self, x):
        """
        Overview:
            Update mean, variable, and count
        Arguments:
            - ``x``: the batch, shape (*,)
        """
        batch_mean = np.mean(x, axis=x.shape[:2])
        batch_var = np.var(x, axis=x.shape[:2])
        batch_count = x.size

        ...
```

Q2: 理论题第一题中，梯度消失是指梯度变为 0 了吗?

A2: 梯度消失是一种梯度反向传播中常发生的现象，它不是指损失函数对某个模型系数的梯度变为 0，而是指梯度传导过程中，随着传导距离变长，远距离的梯度传导的数值趋向于 0 的现象。但近距离的梯度传导还是存在的，这一点可以从梯度计算的各个解析项中可以看到。在数值观测下，可以观察到 Loss 或目标函数在很小的数值范围内长时间震荡，观测某中间层的梯度基本为 0，停到更差的一个局部最优值，而无法收敛到我们需要的一个局部最优值（假如通过某些方法已经提前获知，比如对于一个具有解析解的问题）。

Q3: 理论题第一题中，梯度消失和梯度爆炸会导致 RNN 模型训练困难吗?

A3: 是的，梯度消失会使得较远距离传导的梯度衰减严重，从而模型的参数变更将更大程度上依赖时间序列上局部和近期的数据。梯度爆炸如果不加以控制，会使得梯度数值的量级越来越大，进而出现指数爆炸现象，使整个训练过程崩溃，不过因为在参数更新过程中常常采用了梯度截断的技巧，所以可以稍微缓解一些梯度爆炸现象的负面效果。

Q4: 理论题第二题中，belief state MDP 中，当前信念的大小，会如何影响更新后的信念？

A4: 当前的信念，可以使用已知的状态转移函数，可以估计计算下一个时刻的状态，并将该值作为贝叶斯定理中的先验（prior）。而观测函数，可以作为贝叶斯定理中的似然函数（likelihood），它衡量了不同状态下，观测到某个现象的可能性。贝叶斯定理计算获得的后验（posterior）就是更新后的信念，它是基于新的观测的基础上，对信念的重新估计。比如在扑克、军棋等非完全信息的场景中，每一次行动之后，产生的新的观测数据，都可以重新估计当前的各个状态的信念，比如当前对手的手牌、该棋子的可能的属性等等。

参考文献：

[1] Xu M, Quiroz M, Kohn R, et al. Variance reduction properties of the reparameterization trick[C]//The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, 2019: 2711-2720.

[2] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//International conference on machine learning. PMLR, 2018: 1861-1870.