MADDPG Algorithm

AAIS in PKU　陈伟杰 1901111420

May 17, 2020

1 Problem Setting

已知足球环境 google football，采用 Simple115 的向量表示状态，行动空间采取默认的 19 维行动空间。运用 MADDPG 方法对 academy_3_vs_1_with_keeper 问题进行求解，同时引入 soft update 技术使得训练过程更稳定。

2 Experiment Result

实验中参数的设置可以查看 data.py，Actor 和 Critic 函数都采用 3 层全链接网络，Actor 网络最后引入 argmax 函数使其成为动作的确定输出，并且均采用 Adam 算法来优化损失函数。

对于 academy_3_vs_1_with_keeper，由于训练 reward 过于稀疏，因此对记录的记忆（memory）根据 reward 进行划分。此外，由于整条轨迹仅有最后一帧有 reward，为了使得记忆符合现实，因此将 reward 赋予整条轨迹，然后根据 reward 存储到不同的记忆里，共分为正值记忆（positive memory），负值记忆（negative memory）和零值记忆（zero memory）。其中负值记忆和零值记忆的存储长度限定为 200 个转移（transition），正值记忆的存储长度为 2000 个转移（transition）。

reward 采用滑动平均的处理方法使得图线更直观（即 $\hat{r}[i] = \frac{1}{k}\sum_{j=i}^{i+k} r[j]$，$k = 100$）

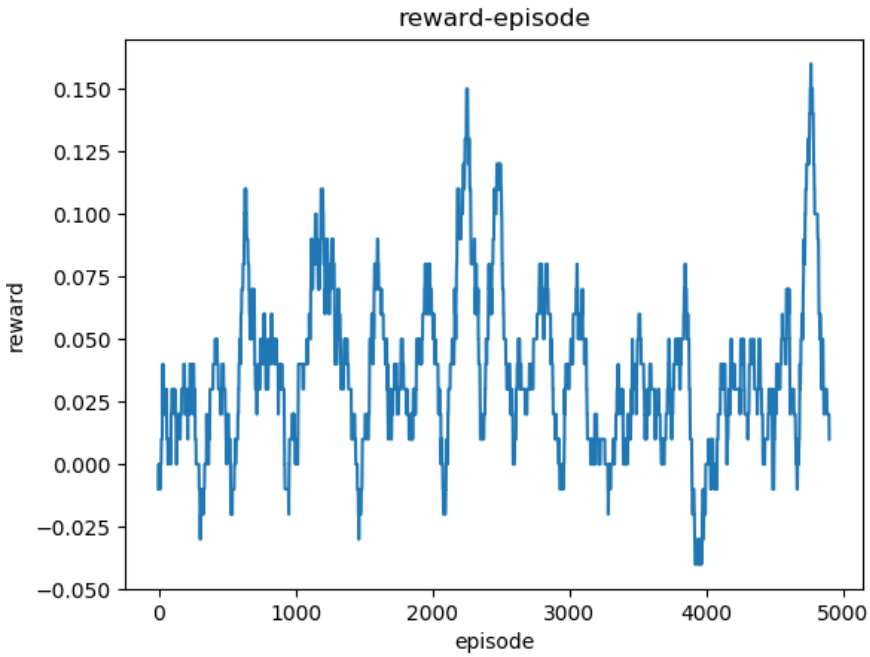

Figure 1: reward 与迭代次数的关系图

实验过程中发现模型训练的随机性极大，既有较高进球率 (约 15%)，也有进球率极低的情况 (不到 1%)。从图线可以看出，随着 episode 的增加，滑动平均的 reward 波动上升，但由于训练资源有限，上升情况并不明显。

3 README

代码包括 data.py，model.py 和 train.py 三个程序，需要 google football 环境。直接运行 train.py 可以得到上述结果

- 电脑：MacBook Pro(15-inch, 2019)
- 处理器：2.6 GHz 6-Core Intel Core i7
- 内存：16 GB 2400 MHz DDR4
- 操作系统：macOS Catalina version 10.15.1
- 语言：python3.6
- 实验环境：docker+ubuntu18.04