

RL in HCI Competition 2 Group 9

M11215015 廖聖郝, M11152027 郭昱辰

Q1. How do you configure your training environment? (2%)

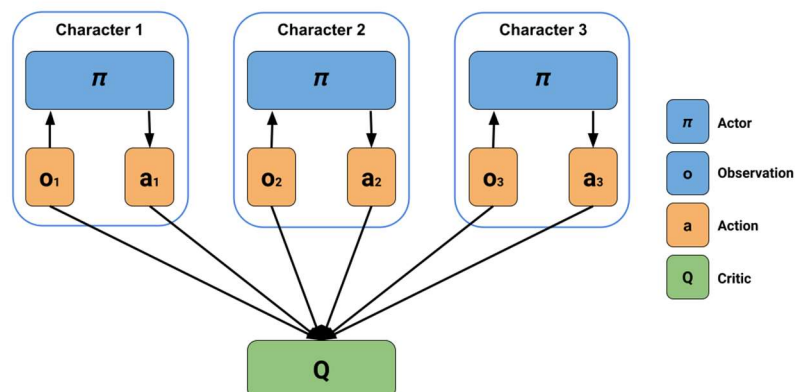
在訓練時使用Observation Space的336維向量(由11條Ray組成)輸入到神經網路，而在計算Reward時，則會使用到Additional Information中的player_info與ball_info，後續將詳細說明我們Reward Function的設計。

Q2. How do you design your RL model? (2%)

在本次的足球遊戲中，因為需要團隊合作，所以我們採用了Multi-agent reinforcement learning (MARL) 類型的演算法MA-POCA。

該演算法由Unity公司於2021年發表，此演算法的核心機制是透過一個中心的神經網路作為教練，為每個agent在隊伍中的表現打分數，這樣agent除了能夠參考自己的感測資訊，也能考慮到action對整個隊伍的影響，做出對團隊更有利的決定，進而達成更好的團隊合作。

除了上述的機制之外，MA-POCA加入了attention network，可以動態的處理不同數量的輸入，這對於會有人隨時加入或退出的線上遊戲來說，是非常適合的。



Q3. How do you design your reward? (2%)

- 跟同隊友碰撞扣分：我們認為與隊友的碰撞不但有可能會讓傳球方面產生失誤，兩個隊友位於同一個位置也不會利於防守，因此我設立了一個如果撞到隊友就會扣分的機制，避免上述的情況發生。
- 面向球加分：當球員踢球時，大部分時間是會面向球往前射門得分的，因此我決定鼓勵球員持續的去面對球所在的方向，認為這樣能增加尋找球並爭奪球權的機率。
- 持續踢球加分：跟上面的原因類似，我們希望球員能夠積極的去爭奪球，所以我們會鼓勵電腦持續的去踢到球，當連續踢到球時給予獎勵。
- 如果讓球靠近對手的球門會加分，太靠近自己的門會扣分：讓電腦學習去避免球到達他的球門附近，並積極的去讓球靠近對方的球門區域。
- 踢球踢進自己的門扣兩倍分數：訓練的過程中發現有自己踢進自己球門的情況，使用這個懲罰避免烏龍球的情況產生。

Q4. How was your agent' s learning process? (2%)

因為足球可以說是一種零和遊戲，所以在本次訓練中，重點不是讓reward越大越好，而是採用一種名叫ELO的評分系統，該系統的目的是評估出每位對手的"相對實力"。

從我們的訓練過程可以觀察到，在150萬step以下，ELO都不會有明顯的增長，數值處在1200~1230之間，接著ELO會快速成長，直到約350萬step時，ELO約1550，然後開始進入緩慢的增長期，通常我們在這時候就會中斷訓練，因為經過實測發現，訓練更久使ELO超過1600並不會讓agent更強，反而會產生反效果。

Q5. What special designs have you incorporated into your algorithm? (2%)

我認為我們reward裡面的設計很特別的其中之一就是面向球加分這點，因為在理論設計上或許有可能讓球員訓練成向後也能進球的結果，但我們仍這樣設計，希望能讓球員更擁有對於球的方向感。

除此之外，本來還有一個彼此之間傳球加分的設計，但因為發現在實際運作的情況下，其實並沒有太多相關時機，而且在訓練有更多值得去關注的要素，因此最後沒有採用。

另外，我們訓練出來的球員，其中一版會一人會往前搶球，另一人則後退守門，因為reward不同的設計而導致產生的不同行為相當有趣。

訓練與測試:

由於訓練與測試的套件版本使用有所差異，因此必須分成不同的環境去建立。

訓練:

依照src/training資料夾裡的readme.txt去建立anaconda環境

```
python train_ma-poca.py ./SoccerTwos.yaml --no-graphics --env=../Soccer-Two  
s_UnityBuild_list/final/Soccer-Twos.exe --run-id="final"
```

訓練完成後pytorch模型將儲存於testing/models/SoccerTwos_final.pth

測試:

依照src/testing資料夾裡的readme.txt去建立anaconda環境

測試方式1 (讓訓練好的模型自己對戰自己):

```
python -m soccer_twos.watch -m example_team_agent -p 22222
```

測試方式2 (可以指定不同的pytorch模型):

```
python testing.py <pytorch model file 1> <pytorch model file 2>
```

EX: python testing.py ./models/SoccerTwos_1.pth ./models/SoccerTwos_2.pth