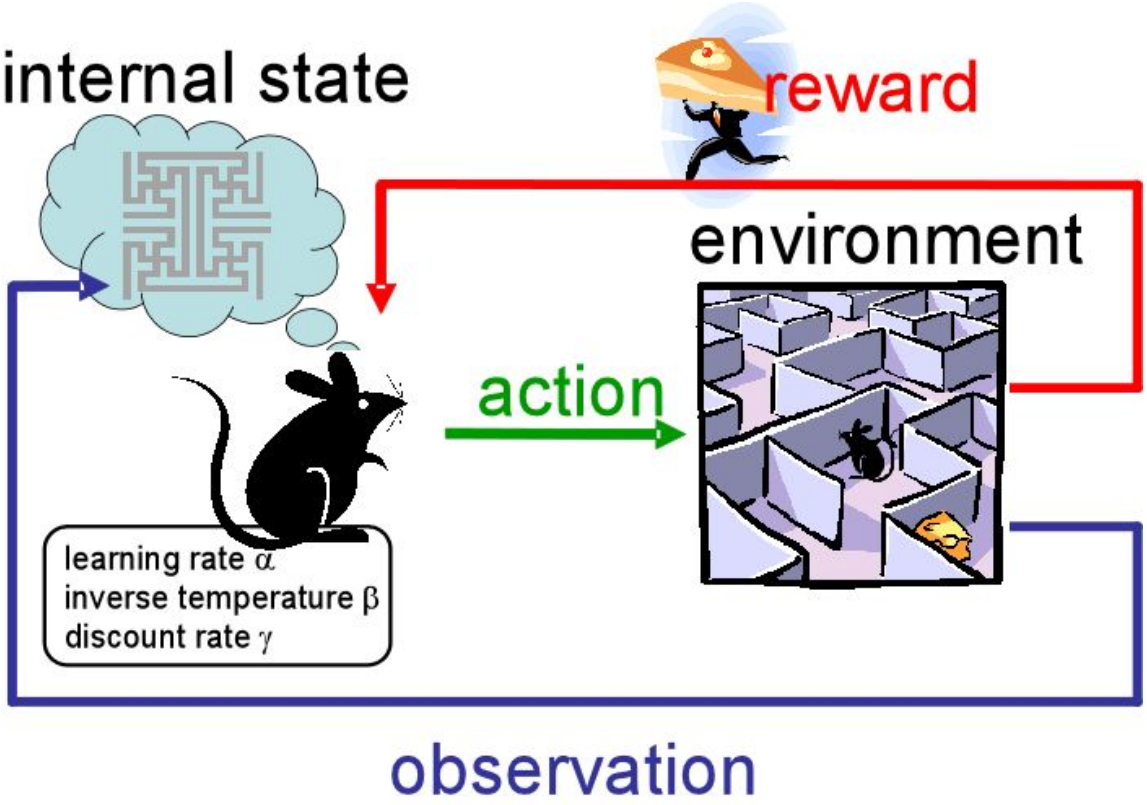
A3C 平行度對學習效率的影響

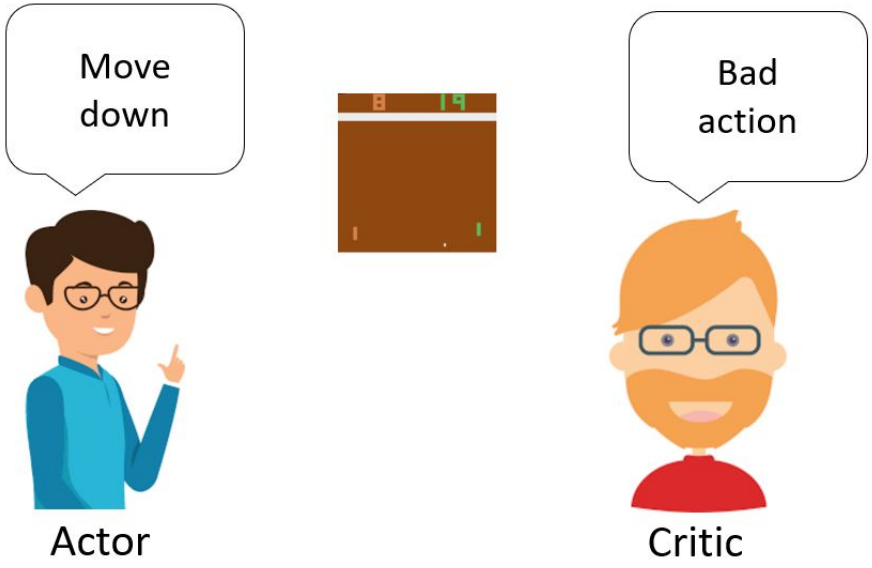
Group 36 周聖諺 嚴中璟

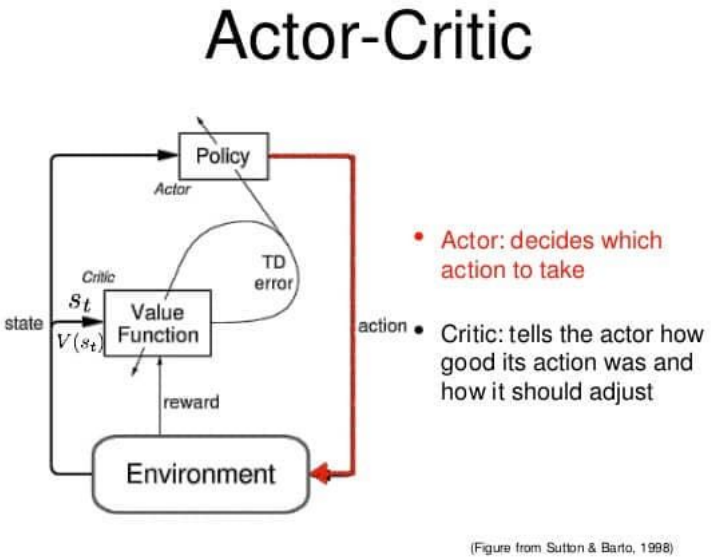
What is A3C ?

Reinforce Learning



Advantage Actor Critic(A2C)

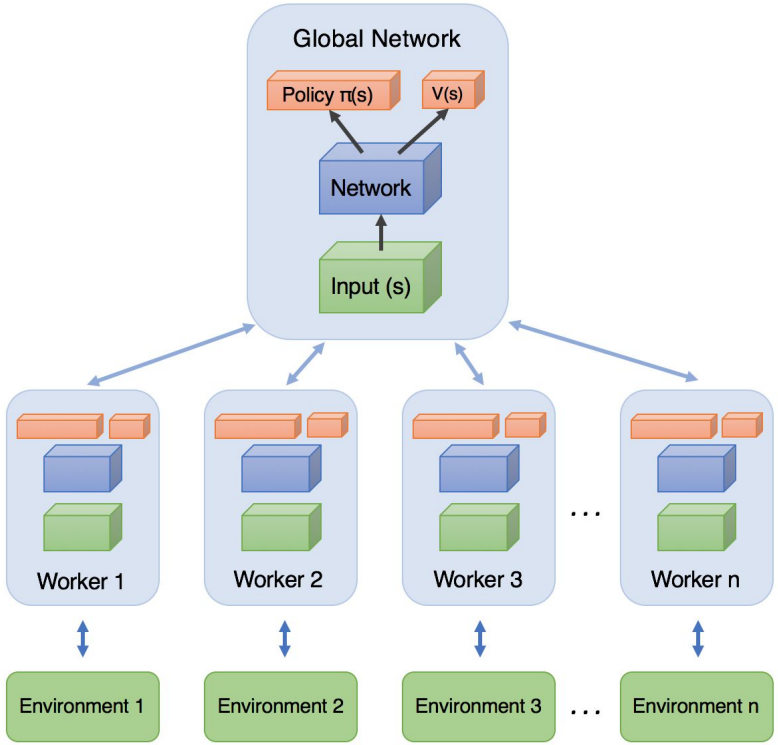




Asynchronous Advantage Actor Critic(A3C)

就把A2C平行化而已

- 每個Worker有獨立的Model和Environment
- Worker回傳gradient給global netowrk 更新
- 把更新Worker的model更新成最新的Global Network



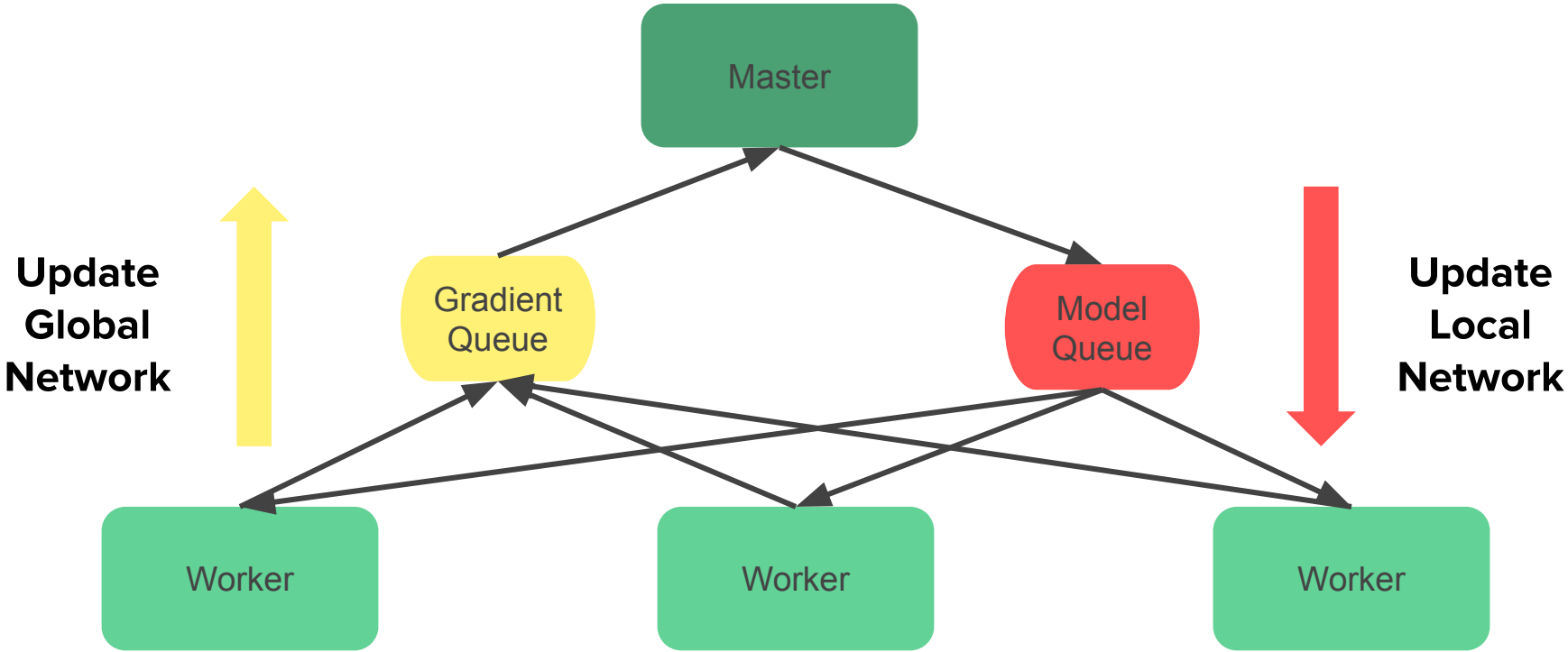
A3C 的問題

原Paper(Asynchronous Methods for Deep Reinforcement Learning)看似很美好，但卻沒有提到A3C的一些問題

- 越多Worker, Model Converge很不穩定，訓練相同的episode下model反而比較弱(Bad Learning Efficiency)
- 但太少Worker平行化加速效果就不好(Bad Perormance)

Tradeoff Between Performance and Learning Efficiency!!

實作 Naive A3C with Multiprocesses



Naive A3C 實作上的困難

- TF2 與 multiprocessing 相容性很差，網路上除了RAY，幾乎沒有人實作
- TF2 並沒有對單GPU給多個Model使用作優化
- 還有很多奇怪的Bugs...

實驗環境

CPU: Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz X2 (10C/20T)

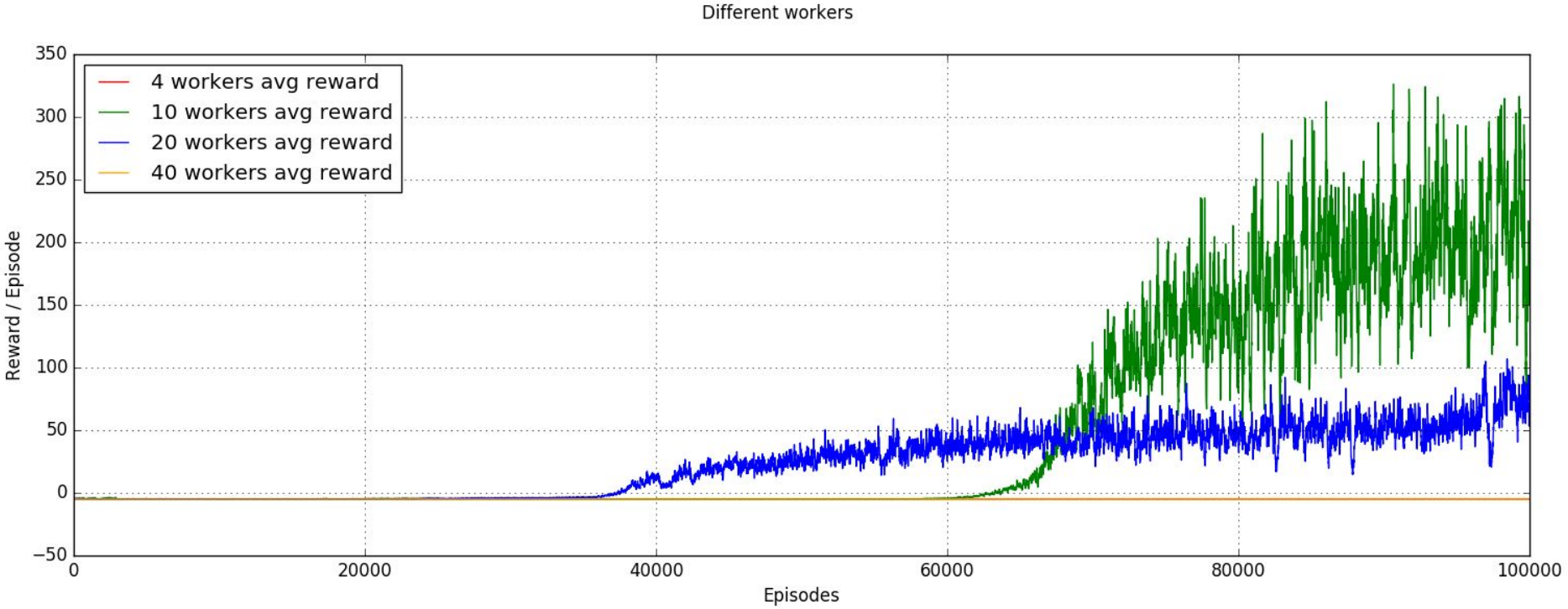
RAM: 128GB

GPU: NVIDIA Titan RTX 24GB X3

Model: 128 Unit Dense Layer

Training Environment: Flappy Bird

實驗一 : 不同數量Worker訓練十萬次的平均得分



Naive A3C 實驗結果與觀察

結果

- 10 Worker最好，4, 20個Worker的學習效果變差，40 Worker則是爛掉

觀察

- 因Model很小，更新不大花時間
- 最吃時間的是Model和環境做互動的時候
- 一開始Model學的不好很快死掉，但越學越好，分數越高，每次玩得也越久

實驗比較

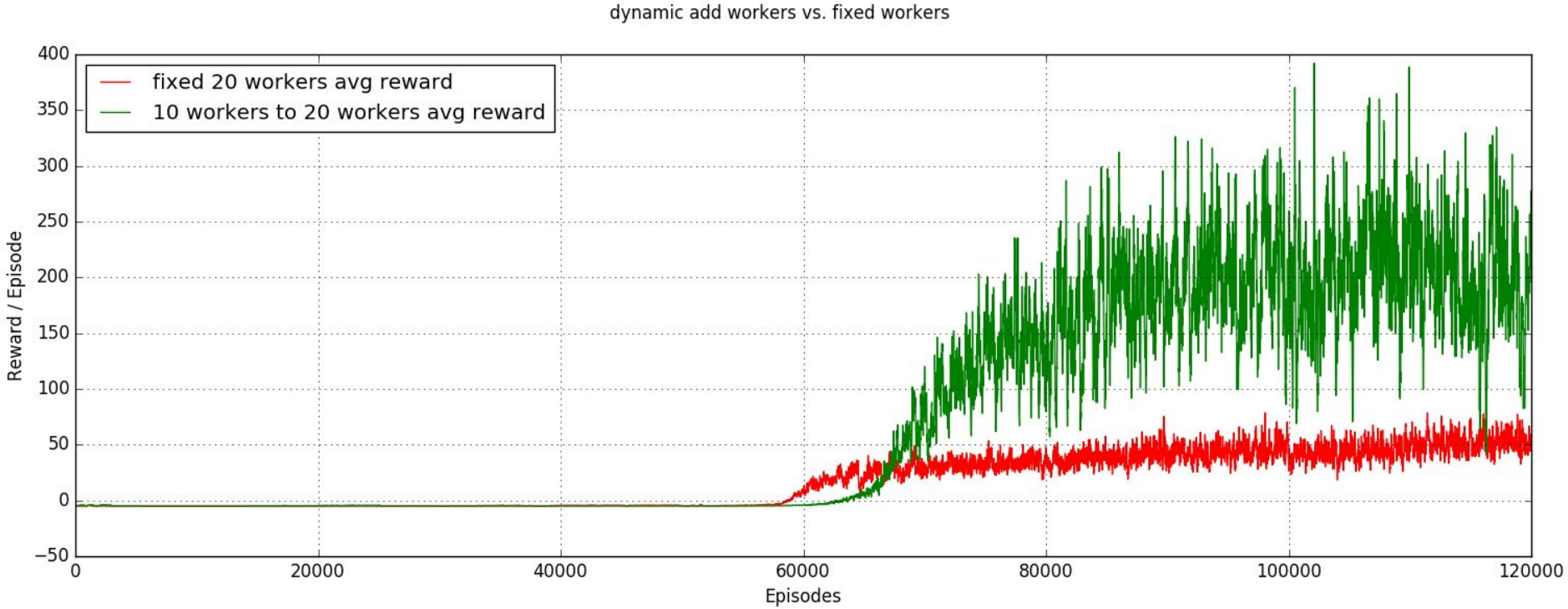
1. 動態增加worker
    a. 6 -> 12
    b. 10 -> 32

動態增加Worker的想法

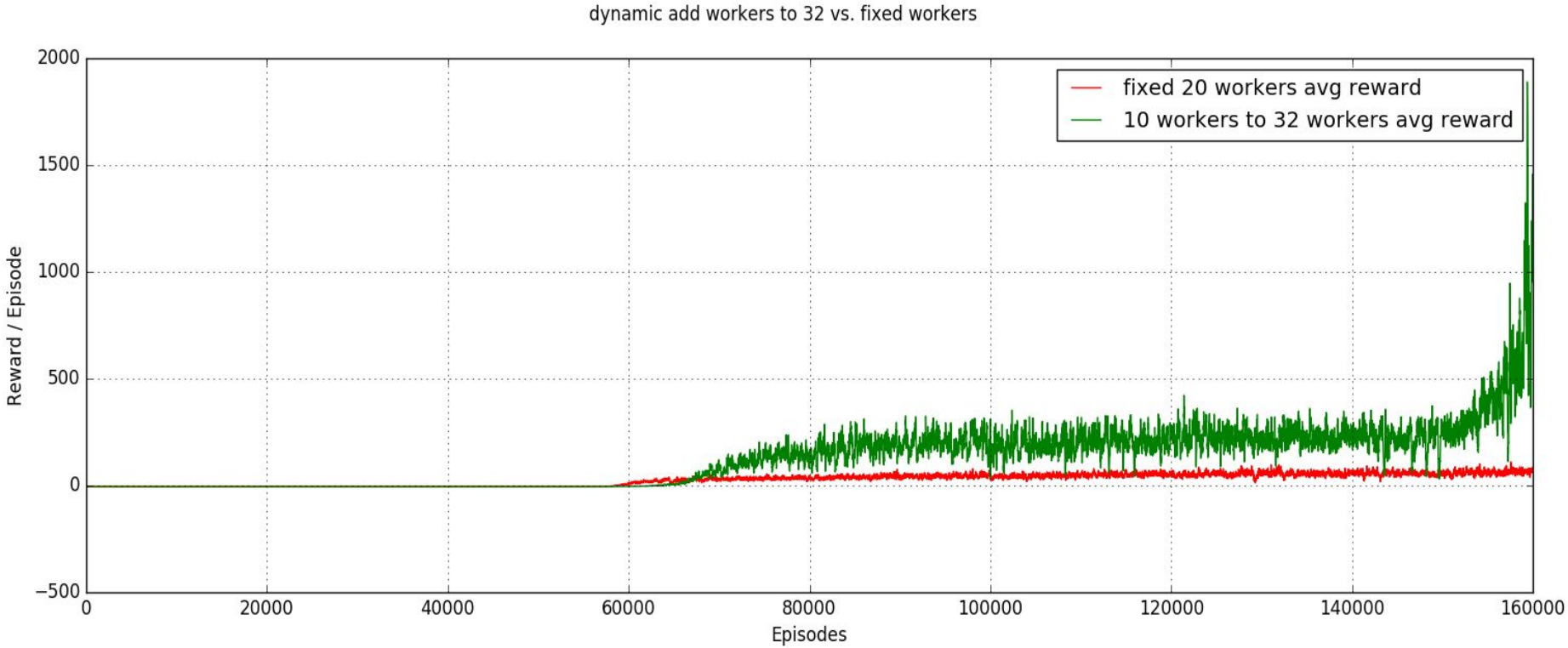
- Worker 太多導致 Master 負擔太重
  - 一開始先用少一點的 Worker
  - 當 Worker 與環境互動較久時 Master 的負擔變少，可再增加 Worker

- 測量 Worker 在每個 Epoch 平均所花時間
- 每次平均時間超過20秒時則增加1個 Worker
  - 平均 > 20秒時 + 1 個 Worker
  - 平均 > 40秒時 + 2個 Workers
  - 平均 > 60秒時 + 3個 Workers

動態增加 10➡20


dynamic add workers vs. fixed workers
fixed 20 workers avg reward
10 workers to 20 workers avg reward
Reward / Episode
Episodes
400
350
300
250
200
150
100
50
0
−50
0
20000
40000
60000
80000
100000
120000

動態增加 10➡32


dynamic add workers to 32 vs. fixed workers
fixed 20 workers avg reward
10 workers to 32 workers avg reward
Reward / Episode
Episodes
2000
1500
1000
500
0
−500
0
20000
40000
60000
80000
100000
120000
140000
160000

結論

- worker的數量變多，可大幅提升速度，但不容易穩定
- 用動態增加的方法來避免不穩定且達到加速的效果

Thanks For Listening