SVS - Assignment 1

Andrea Serafini, Alessandro Martignano, Angelo Tinti, Piero Sanchi

November 2022

1 Requisiti di sistema

L'assignment consiste nell'addestrare dei droni quadrirotore allo scopo che attraversino un corridoio tridimensionale contenente delle sfere di dimensione arbitraria.
Il drone dovrà arrivare sul fondo del percorso evitando di colpire le sfere, di uscire dai bordi del corridoio e di toccare il pavimento. La strada che dovrà percorrere il drone sull'asse X è di 60m, la larghezza (y) e l'altezza (z) del corridoio sono entrambe di 10 metri (da -10 a +10 per l'asse y, da 0 a +10 per l'asse z). Il risultato ottimo si ottiene nel caso in cui il drone riesca a generalizzare l'ambiente in cui si trova.

2 Architettura proposta

L'architettura proposta si basa su una soluzione che utilizza reinforcement learning con un meccanismo di reward che premia o punisce (tramite l'assegnazione di punti) il drone in base a dei comportamenti che esso assume durante le sue run, in modo che sia portato a massimizzare il suo punteggio finale. Per affrontare il problema dopo aver attentamente studiato il dominio applicativo abbiamo deciso di adottare degli accorgimenti per facilitare la computazione (soprattutto a livello di tempi) e l'apprendimento di un drone. Nella prossima sezione illustreremo brevemente le scelte intraprese a livello di environment.

2.1 Impostazioni dell'ambiente

Per agevolare il training ed il testing del modello abbiamo fatto ricorso a delle modifiche (in fase di training) all'ambiente, le riassumiamo brevemente:

- abbiamo optato per l'utilizzo di un singolo drone per ogni episodio dell'allenamento per alleggerire la computazione;
- lo spazio di osservazione del drone è stato ridimensionato da 12 a 42 elementi aggiungendo allo spazio di tipo KIN le distanze sui 3 assi delle 10 sfere più vicine;
- il numero di sfere adottato per i training è quello previsto dalla modalità "easy" (0.5).

2.2 Meccanismo di Reward

La funzione di reward è così definita

$$R = \begin{cases} (10/(70-x)) \cdot \varepsilon & \text{if } \Delta x > 10cm \\ 1 & \text{if else sphere surpassed} \\ -0.1 & \text{if else } \Delta x < -10cm \\ -1 & \text{if else drone out of bounds} \\ -1 & \text{if else collision} \end{cases}$$

In caso il drone stia avanzando, per ogni 10cm che percorre riceve una ricompensa proporzionale alla distanza percorsa lungo l'asse x e moltiplicata per un fattore ϵ così definito:

$$\varepsilon = 1 + 0.5 \cdot (1 - \left\| \begin{matrix} \Delta y \\ \Delta z \end{matrix} \right\|)$$

Il parametro ϵ smorza la reward in funzione di quanto il drone si allontana dal centro del piano delimitato dagli assi y e z. Questa distanza si ottiene sottraendo a 1 la norma del vettore di scostamento lungo questi 2 assi le cui componenti sono normalizzate tra 0 e 1. In questo modo, si premia il drone se avanza restando al centro di in un ipotetico "corridoio" che attraversa l'ambiente lungo l'asse x.
Il drone riceve una reward di 1 se sorpassa, lungo l'asse x, una nuova sfera.

Al contrario, se il drone retrocede di più di 10cm viene penalizzato di 0.1 Se invece il drone collide con una sfera o supera gli estremi dell'ambiente la penalità è di -1.

3 Analisi delle performance

Per monitorare l'andamento dei training ed evaluation oltre ai tempi impiegati per la computazione del training abbiamo fatto ricorso ai tool offerti da TensorBoard.

3.1 TensorBoard

Di seguito riporteremo i valori a cui abbiamo prestato maggiore attenzione durante i training del nostro modello:

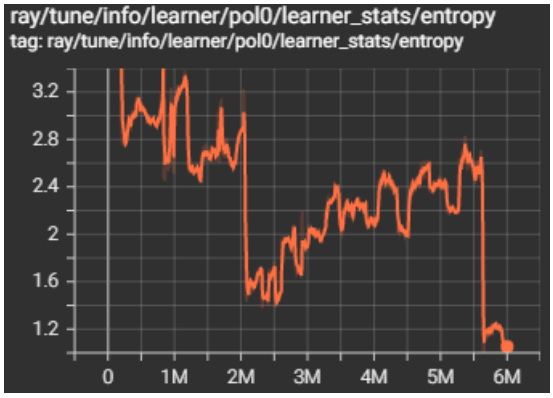

Figura 1: entropia

- **Entropy** $\rightarrow$ indica quanto randomicamente sono compiute le decisioni all'interno del modello. In un modello ben fatto dovrebbe decrescere durante il training. Come si può notare dalla Figura 1 nel nostro caso decresce affrontando dei picchi in cui il drone compie scelte randomiche.

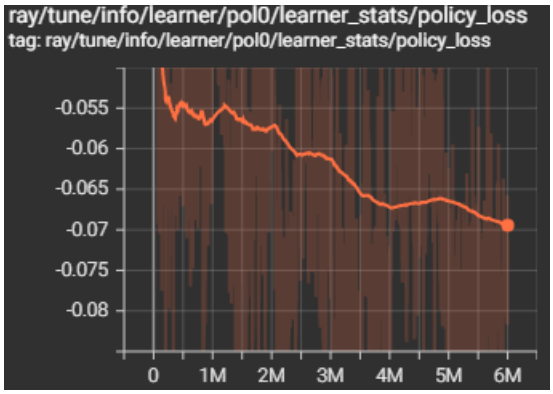

Figura 2: Policy loss

- **Policy loss** $\rightarrow$ indica quanto la policy (processo decisionale) viene modificato. La linea dovrebbe decrescere in un modello di successo.

Figura 3: Reward Media

- **Reward media** → la reward media tendenzialmente dovrebbe aumentare, da questa curva ci si può aspettare dei picchi di alti e bassi come possiamo anche notare dal nostro grafico, in cui comunque la tendenza è quella di migliorare la reward col procedere del training.

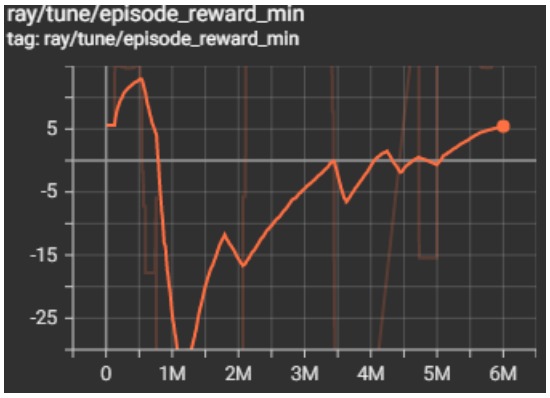

Figura 4: Reward minima

- **Reward minima** → anche la reward minima tendenzialmente dovrebbe aumentare con l'avanzamento del training, a segnalare un effettivo apprendimento da parte del modello nel minimizzare le perdite e guadagnare punti.

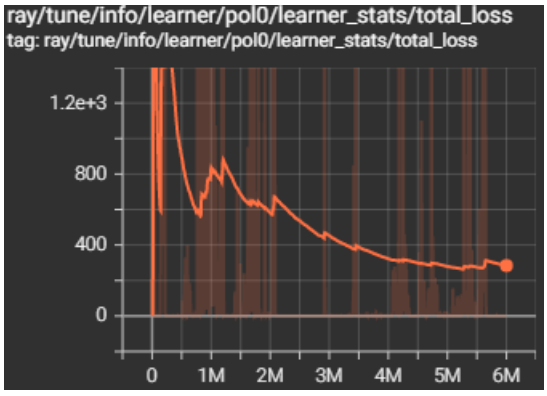

Figura 5: Total Loss

- **Total loss** $\rightarrow$ ci mostra quanto i modello sia capace di predirre il valore di ogni stato. Anche essa dovrebbe aumentare in una prima fase in cui l'agente sta imparando (le reward salgono) e poi decrescere progressivamente man mano che le reward si stabilizzano. Nel nostro caso il grafico rappresenta esattamente la situazione attesa.

4 Considerazioni finali

Sviluppare il sistema richiesto ha mostrato sin da subito delle complessità. Partendo dalla comprensione della libreria utilizzata al capire il corretto funzionamento delle numerosi componenti utilizzate per un progetto di reinforcement learning in un ambiente fisico simulato.

In Meccanismo di Reward è illustrata la funzione di reward utilizzata per allenare il modello, nonostante i numerosi test, non ha ancora portato al pieno raggiungimento del risultato richiesto nella consegna dell'elaborato. Infatti il drone durante le simulazioni può prediligere un comportamento errato, perché dal suo punto di vista è 'migliore' questa via.[1]

Forse degli addestramenti con un numero di iterazioni maggiori di quanto illustrato in TensorBoard avrebbe potuto portare ad un modello più adatto a quanto richiesto, viste però le tempistiche proposte, un numero alto di addestramenti non avrebbe permesso una frequenza adeguata di intervento sul codice.

In conclusione considerando questa nostra prima esperienza con tecnologie di questa tipologia, possiamo considerarci soddisfatti del risultato ottenuto in quanto, anche se non completamente, il modello tende a soddisfarre quanto richiesto dalla consegna dell'elaborato.

[1] Nonostante il meccanismo di reward penalizzi notevolmente questo comportamento.