UNIVERSITÀ DI TRENTO
Dipartimento di Ingegneria e Scienza dell'Informazione

Backpropagation-based reinforcement learning for decision trees

Supervisore

- Prof. IACCA Giovanni

Co-supervisore

- Dott. CUSTODE Leonardo Lucio

Laureando

- SCEVAROLI Simone

Contesto generale

Contesto

- Reinforcement Learning
- Sviluppo di un metodo di evoluzione dei decision trees che combina Grammatical Evolution e un algoritmo creato per reti neurali (***backpropagation***)
- Principali vantaggi e svantaggi, comparato al solo Grammatical Evolution

Algoritmi e modelli usati

Decision Tree e Grammatical Evolution (def.)

Modello:

- Decision Tree

Algoritmo principale:

- Grammatical Evolution, un algoritmo genetico basato sulla mappatura genotipo-fenotipo[1]

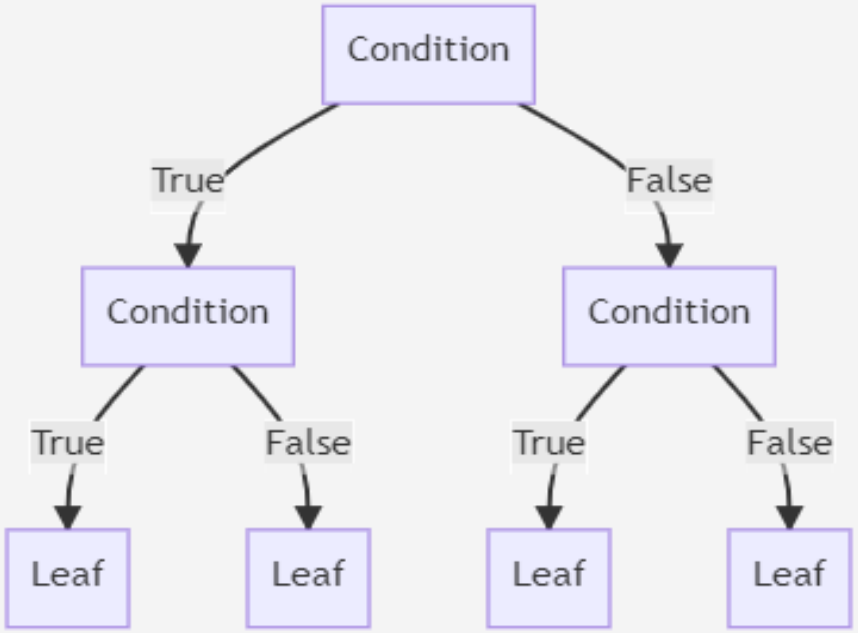


Figura_1: Esempio di decision tree

Proximal Policy Optimization (def.)

Proximal Policy Optimization (abbr. ***PPO***) è un algoritmo per il Reinforcement Learning che si basa sulla discesa del gradiente.[2]

L'algoritmo alterna:

- Sample dei dati
- Ottimizzazione funzione «surrogata» tramite ***ascesa stocastica del gradiente***

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right] \quad \text{dove} \quad r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

Figura_2: Clipped surrogate objective function

Processo evolutivo

Overview del processo evolutivo

PPO è stato applicato ogni «tot» generazioni su un albero scelto dalla popolazione tra:

- il migliore fino a quella generazione (***exploit***)
- uno scelto randomicamente (***explore***)

PPO agisce direttamente sul fenotipo

→ gli alberi ottimizzati da PPO non sono stati reinseriti nella popolazione

Test e risultati

CartPole-v1

Questa task consiste nel tenere un'asta in equilibrio attaccata ad un carrello che può muoversi a destra o a sinistra.[3]

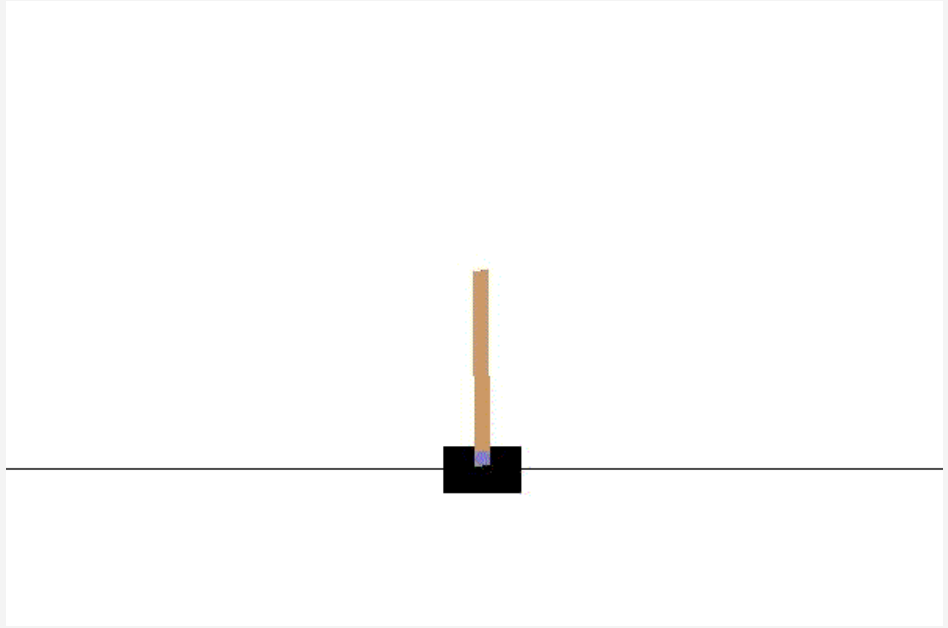
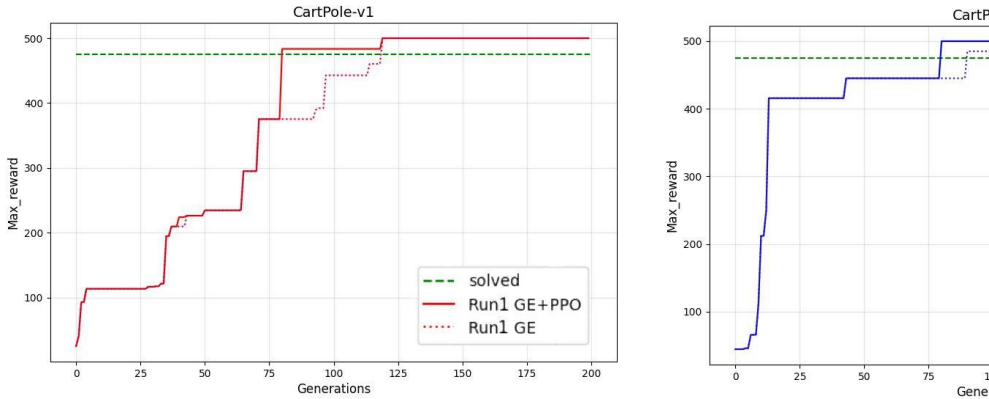


Figura_3: Render di CartPole-v1

CartPole-v1 (risultati)

- Algoritmo GE+PPO ***più performante*** rispetto al solo Grammatical Evolution
- In più di una run gli individui inviati a PPO hanno poi ottenuto l'ottimo
- Significativa riduzione per alcune run del numero di generazioni necessarie per ottenere l'ottimo

CartPole-v1 (risultati) (cont.)



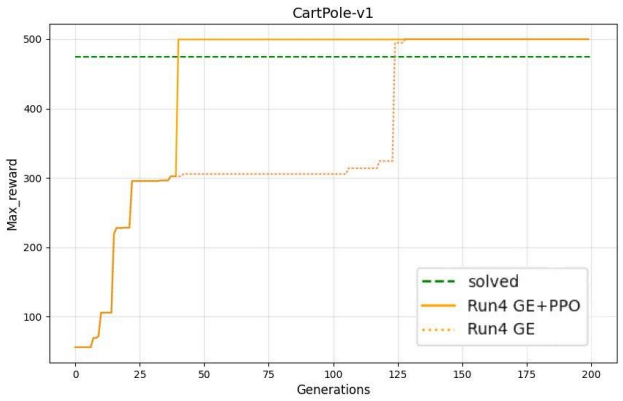


Figura_4: Focus su 3 run.

La linea continua indica l'uso di GE+PPO, mentre la linea tratteggiata indica l'uso del solo GE.

Per ogni generazione sono stati valutati 200 individui su 15 episodi ciascuno.

Il plot rappresenta ad ogni generazione il «best so far».

CartPole-v1 (risultati) (cont.)

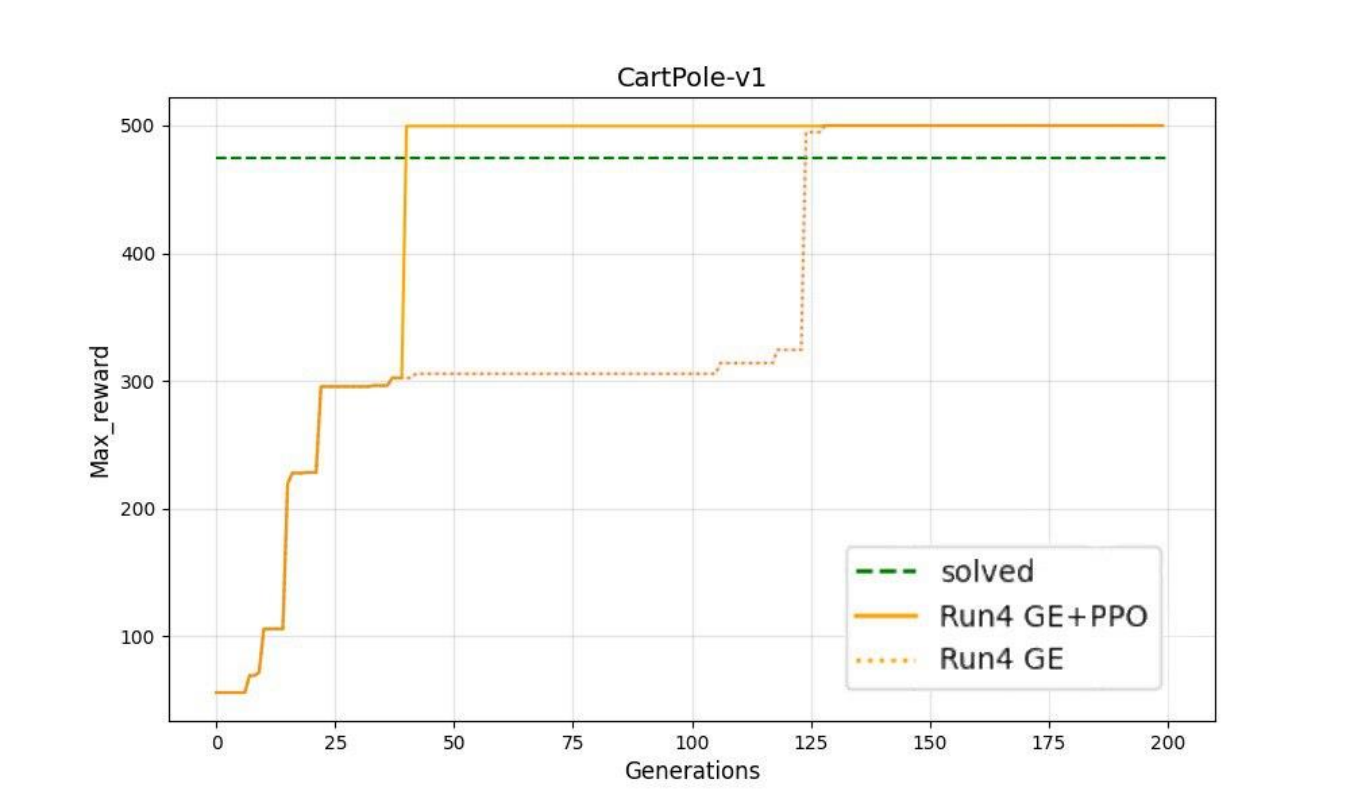


Figura_5:
La linea continua indica l'uso di GE+PPO, mentre la linea tratteggiata indica l'uso del solo GE.

Per ogni generazione sono stati valutati 200 individui su 15 episodi ciascuno.

Il plot rappresenta ad ogni generazione il «best so far».

CartPole-v1 (risultati) (cont.)

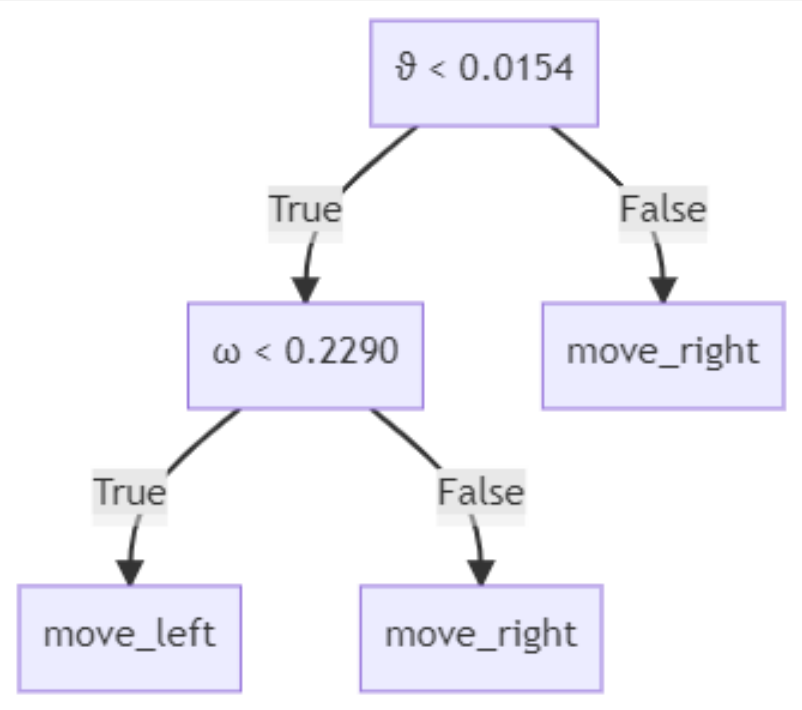


Figure_6: Miglior policy ottenuta da PPO (usato come supporto)

Dove:

- **ϑ** → angolo formato tra l'asta e l'asse y
- **ω** → velocità angolare dell'asta

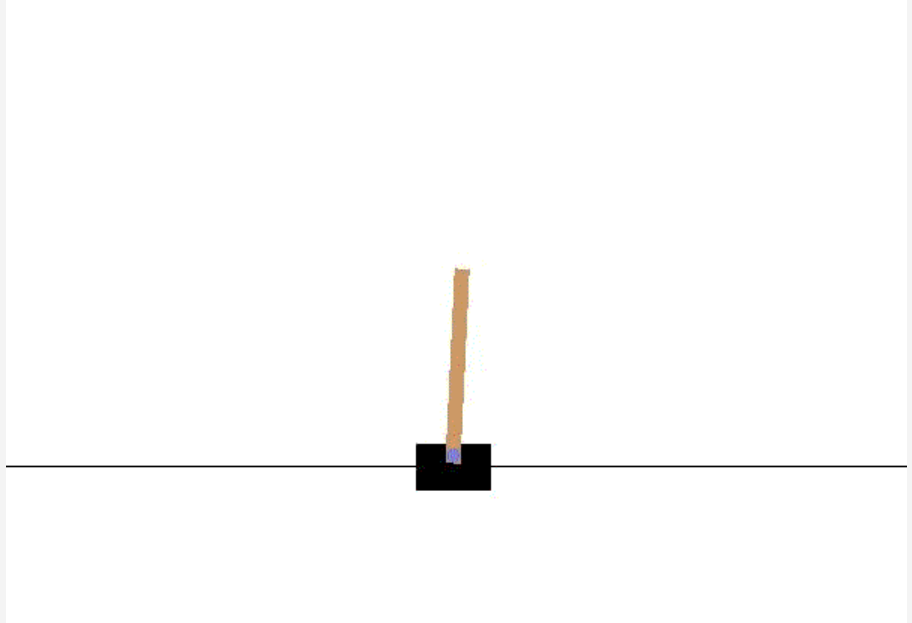


Figura_7: Render utilizzando la miglior policy ottenuta tramite PPO (usato come supporto). Reward totale: 500.0

Acrobot-v1

Il sistema è composto da due aste connesse linearmente fra loro a formare una catena, con un'estremità della catena fissata.

L'obbiettivo è quello di applicare una forza alla giuntura fra le due aste per far raggiungere l'estremità della catena oltre un'altezza di soglia.[3]

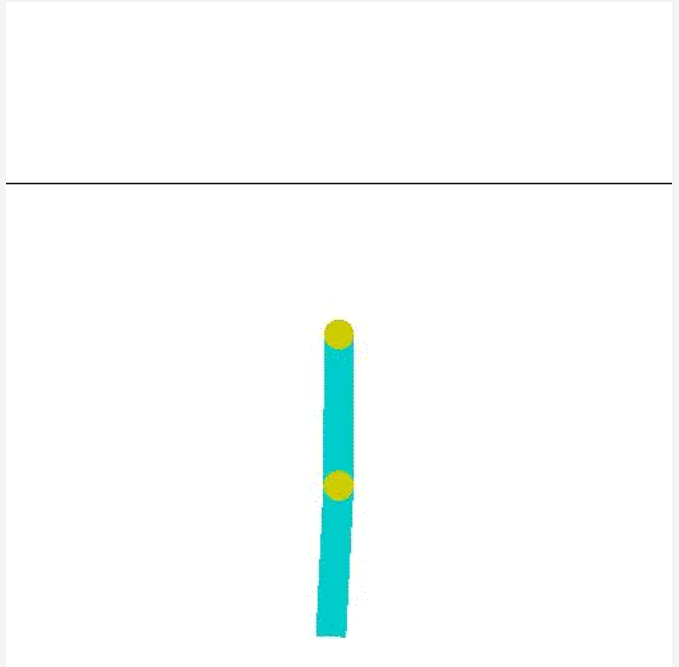


Figura_8: Render di Acrobot-v1

Acrobot-v1 (risultati)

- L'environment ha ***reward sparso***
- PPO risultato ***non utile*** per supportare il processo evolutivo
- Nessun albero passato a PPO durante il processo evolutivo risolve il task

Acrobot-v1 (risultati) (cont.)

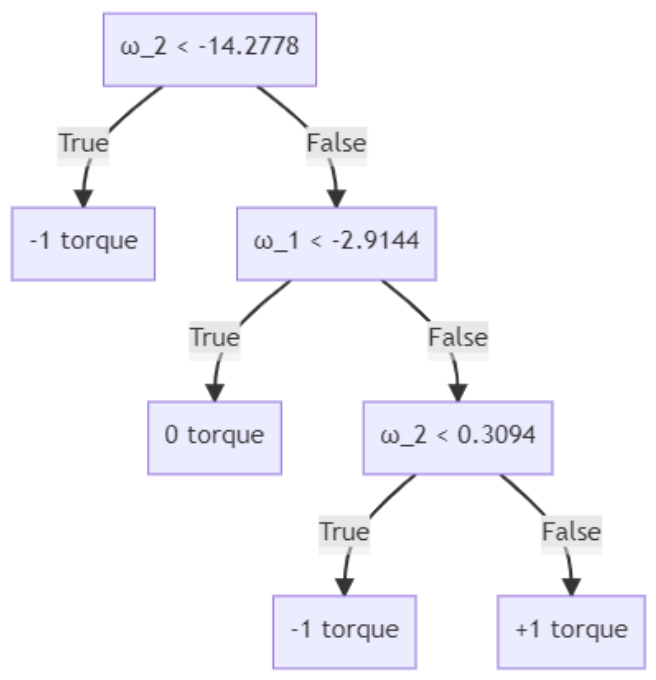


Figure_9 : Miglior policy ottenuta da PPO (usato come supporto)

Dove:

- **ω_1** e **ω_2** → velocità angolare della prima e della seconda asta

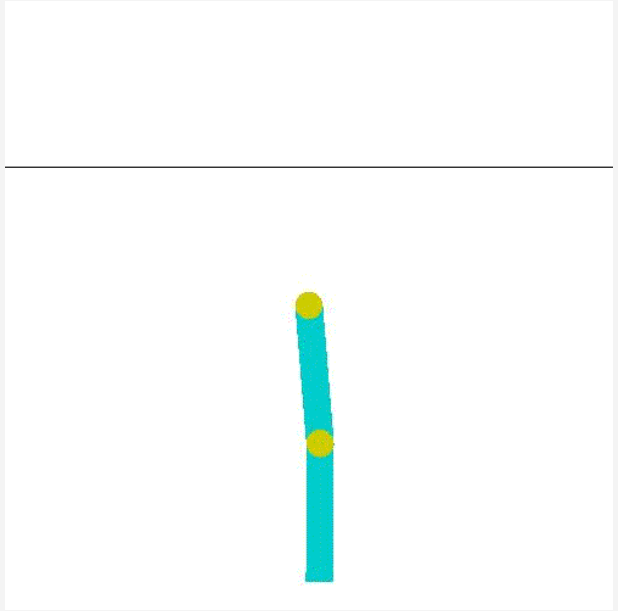


Figura_10: Render utilizzando la miglior policy ottenuta tramite PPO (usato come supporto). Reward totale: -112.0

MountainCar-v0

Questo problema consiste in una macchina posizionata stocasticamente ai piedi di una valle sinusoidale, con la possibilità di accelerare a destra o a sinistra.

Il goal è quello di raggiungere la cima della collina di destra.[3]

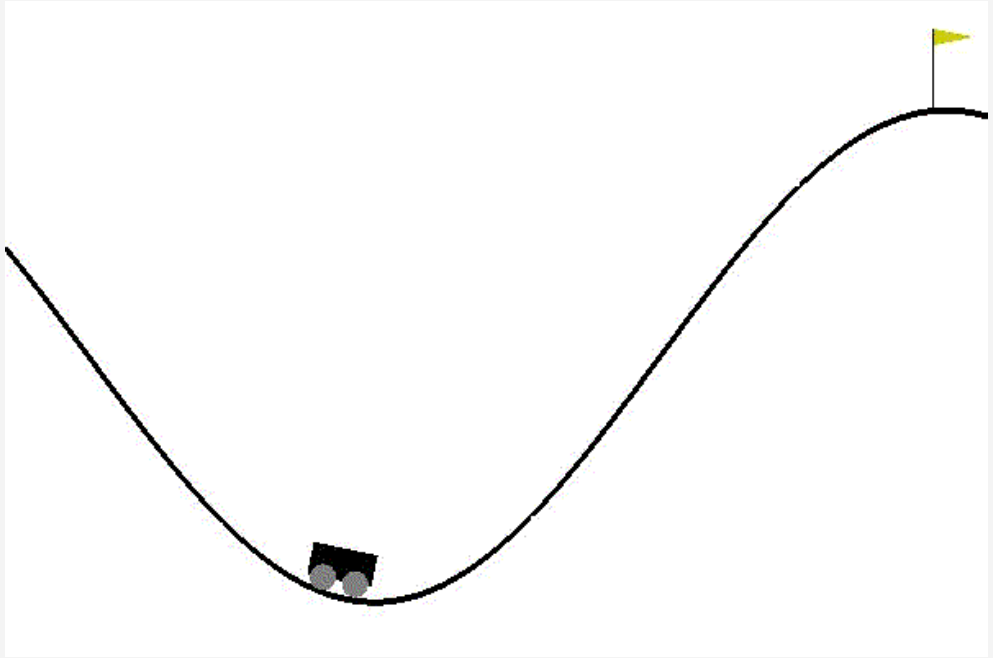


Figura_11: Render di MountainCar-v0

MountainCar-v0 (risultati)

- Task simile ad Acrobot-v1
- Aggiunta di un constraint per migliorare l'ottimizzazione degli alberi passati a PPO
- Alcuni alberi ottimizzati da PPO (usato come supporto) risolvono il task
- PPO usato come supporto ***ha aiutato lievemente*** il processo evolutivo, ma nel complesso l'algoritmo GE+PPO ***non*** è risultato ***migliore*** rispetto al solo GE

MountainCar-v0 (risultati) (cont.)

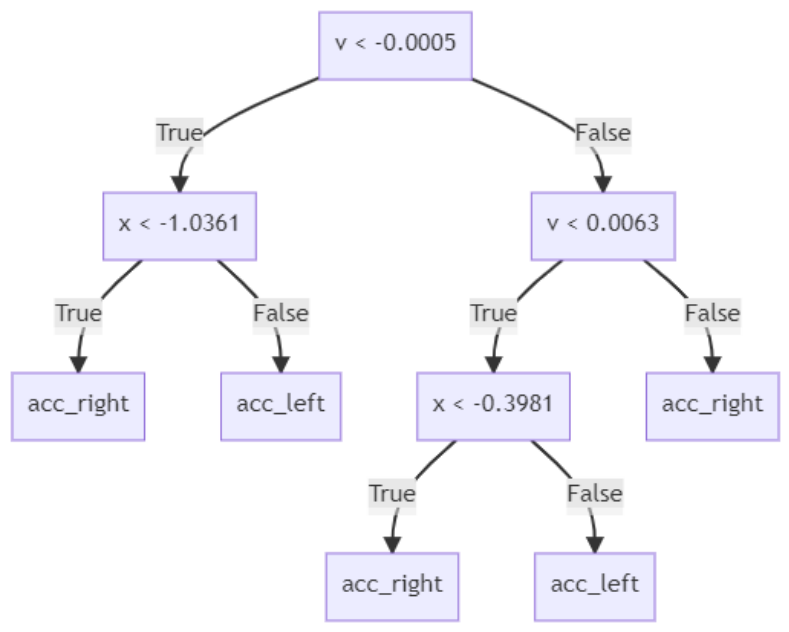


Figure_12 : Miglior policy ottenuta da PPO (usato come supporto)

Dove:

- **v** → velocità della macchina
- **x** → posizione rispetto all'asse x della macchina

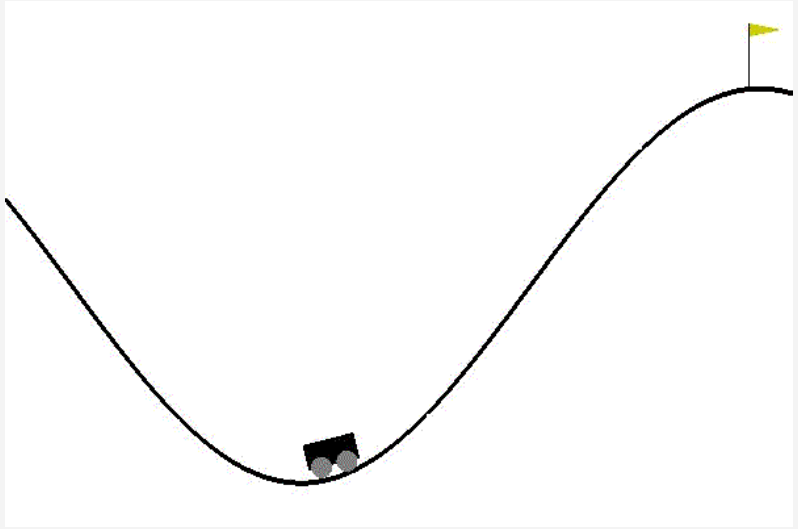


Figura_13: Render utilizzando la miglior policy ottenuta tramite PPO (usato come supporto). Reward totale: -103.0

Conclusioni

Conclusioni

Considerazioni finali

- PPO è un algoritmo ***valido*** a supportare l'evoluzione solo in environment ***con reward non sparso***
- Utile per velocizzare i tempi di evoluzione delle policy (***parallelizzabile***)

Spunti di miglioramento

- Tuning più accurato dei parametri di PPO
- Implementazione del metodo di conversione fenotipo-genotipo
- Trovare un benchmark più ampio per corroborare i risultati ottenuti
- Usare algoritmi di RL diversi da PPO per gestire il reward sparso[4]

Bibliografia

- [1] M. O'Neill and C. Ryan. Grammatical evolution. IEEE Transactions on Evolutionary Computation, 5(4):349–358, 2001.
- [2] John Schulman, FilipWolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. CoRR, abs/1707.06347, 2017.
- [3] Farama Foundation. Gym documentation. https://www.gymlibrary.ml/environments.
- [4] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel and Wojciech Zaremba. Hindsight Experience Replay. CoRR, abs/1707.01495, 2017.
- [5] Alexander Zai and Brandon Brown, Deep Reinforcement Learning in Action, March 2020

GRAZIE PER L'ATTENZIONE