Relazione Dino Chrome IA

Come implementare un algoritmo genetico

Emanuele Riccardi 05121 07254

Luigi Emanuele Sica 05121 09540

Università di Salerno

Professore - F. Palomba

`https://github.com/RCCMNL/ProgettoFIA`

Contents

1 Introduzione

Lo scopo di questo progetto è quello di creare una IA capace di "giocare" al gioco Dino Chrome. Il gioco Dino Chrome, conosciuto anche come T-Rex Game e Dino Runner, è un gioco disponibile sul browser Google Chrome, in assenza di connessione Internet. Sviluppato nel 2014 da Sebastian Gabriel, il gioco ha lo scopo di evitare degli ostacoli , tra cui cactus e pterodattili mentre il dinosauro corre ,con l'avanzare del game, sempre più veloce verso di essi.

2 Algoritmi genetici

2.1 Un pò di contesto

Il famoso scienziato Charles Darwin ne l'Origine della Specie (1859) ha introdotto i fondamenti della teoria dell'evoluzione. In essa, organismi della medesima specie evolvono tramite un processo che prende il nome di selezione naturale.
Questa teoria è riassumibile in 4 punti cardine:

- **Variazione:** gli individui in una popolazione differiscono nel patrimonio genetico (genotipo), che implica molte variazioni nelle loro caratteristiche fisiche (fenotipo).
- **Ereditarietà:** gli individui si riproducono e trasmettono parte del loro materiale genetico alla loro prole.
- **Selezione:** (and Adaptation): alcuni individui possiedono tratti ereditati che gli permettono di sopravvivere più a lungo in un ambiente e /o produrre ancora più prole. Di conseguenza, questi tratti tenderanno ad essere predominanti nell'intera popolazione.
- **Tempo:** a lungo andare la selezione può comportare la nascita di nuove specie (speciazione)

John Holland introduce il concetto di genetic plans nel libro Adaptation in Natural and Artificial Systems, nel 1975, oggi i GAs sono usati per la loro estrema praticità e flessibilità.

Algoritmo Genetico: procedura ad alto livello (metaeuristica) ispirata dalla genetica per definire un algoritmo di ottimizzazione capace di esplorare in maniera efficiente lo spazio di ricerca. Un GA fa evolvere iterativamente una popolazione di individui (soluzioni candidate), producendo di volta in volta nuove generazioni di individui migliorati, rispetto ad una cosiddetta misura di fitness, finché uno o più criteri di arresto non sono soddisfatti. La generazioni di nuovi individui avviene per mezzo di tre operatori di ricerca, quali selezione, crossover e mutazione

3 Elementi di un algoritmo genetico

3.1 I cromosomi

Sono composti dall'insieme dei valori
W1 = peso velocità, **W2** = peso distanza, **W3** = peso larghezza.
X1 = velocità di gioco, **X2** = distanza dell'ostacolo, **X3** = larghezza dell'ostacolo.
Problema: K = W1 * X1 + W2 * X2 + W3 * X3

Qualsiasi valore casuale di <W1, W2, W3>è una possibile soluzione, quindi il termine <W1, W2, W3> può rappresentare un cromosoma.

3.2 La generazione

Una generazione comprende più cromosomi, con il P numero di cromosomi che si riferisce alla dimensione della popolazione. negli algoritmi genetici questo valore potrebbe cambiare con ogni generazione, ma generalmente viene mantenuto costante durante tutto il processo come abbiamo fatto noi dove manteniamo il numero di 30 dinosauri costante per ogni generazione.

3.3 La logica di selezione

Una parte elementare dell'evoluzione in natura è la selezione naturale, che si basa sulla "forma fisica" di un individuo per adattarsi al mondo che cambia o ai suoi dintorni. Allo stesso modo, nel nostro algoritmo, creiamo il concetto di fitness (andando a prendere sempre i migliori due individui, quelli che sopravvivono più a lungo)in modo da poter confrontare due cromosomi e selezionare quello migliore da trasmettere alla generazione successiva.

3.4 Il crossover

Ora che abbiamo selezionato i cromosomi "adatti" per l'accoppiamento, dobbiamo decidere come modellare il processo di accoppiamento per formare una soluzione migliore che abbia le buone proprietà di entrambi i suoi genitori.Per fare ciò seleziono tramite la funzione Math.random un punto randomico del cromosoma dei genitori e scambio i valori successivi al punto di crossover.

3.5 La mutazione

Un altro aspetto importante dell'evoluzione è una mutazione, in cui alcuni individui sviluppano proprietà casualmente . Qui, introduciamo anche cambiamenti casuali in alcune soluzioni per imitare la mutazione. Per ogni elemento del cromosoma mutuo un parametro tra il peso del valore velocità, lunghezza , distanza e bias con un valore randomico scelto tramite la funzione Math.random

Dopo questo processo di generazione, selezione, crossover e mutazione, raggiungiamo la generazione successiva.

4 Implementazione

4.1 GeneticModel.js

formazione(cromosoma) // forma il nostro cromosoma chiamando le funzioni di crossOver, seleziona,mutazione.
fit(cromosoma) //chiama la funzione formazione
seleziona(cromosoma) // Prendo i due migliori individui(due genitori) per generare i figli.
crossOver(genitori, cromosoma) // seleziono un punto randomico da dove poi scambio i vari valori.
mutazione(cromosoma) // Per ogni elemento del cromosoma mutuo un parametro tra velocità,lunghezza e distanza e bias con un valore randomico.

4.2 Genetic.js

function setup() // lanciata appena il file html viene caricato. Inizializza tutto nel gioco e avvia il gioco.
function gestioneReset(dinosauro) // se siamo al primo avvio genero i valori casualmente chiamata a RandomModel(), altrimenti con la geneticModel.fit addestriamo il nostro patrimonio genetico salvandolo.
function gestioneRunning(dino, state) // prende in input array dinosauri e il loro stato, attraverso la chiamata a predizioneSingola restituisce la predizione dell'azione che ogni dinosauro deve compiere.
function gestioneCrash(dino)

4.3 RandomModel.js

predizione(insiemeValori) // in base ai valori velocità,larghezza,distanza e bias prevede se correre o saltare.
train() // Addestra il modello per un dato insieme di input
getCromosoma() //restituisce un valore randomico
setCromosoma(cromosoma) //restituisce un valore randomico
function random() //restituisce un valore randomico

4.4 Passi dell'algoritmo

Reiterazione

Dopo diverse esecuzioni è risultato evidente che l'algoritmo performi meglio se eseguito con un maggiore numero di individui e in un certo lasso di tempo. Si è scelto di eseguire l'algoritmo N volte e scegliere la combinazione numero di individui e valore della randomize che restituisce un buon risultato in tempi accettabili.

Valore della randomize

Dopo vari test inoltre abbiamo deciso di settare il valore costate da sottrarre al risultato della funzione Math.random() a 0.5 in quanto valori maggiori portavano a miglioramenti troppo lenti mentre valori minori rendevano algoritmo ad apprendere troppo velocemente sin dalle prime generazioni.

Size della popolazione

Abbiamo testato 4 dimensioni di popolazione:
La popolazione da **10 individui** ha avuto risultati scarsi in quanto solo dopo più di dieci generazioni riesce a superare il punteggio di 1000.
La popolazione da **30 individui** è risultata un buon compromesso dando risultati soddisfacenti.
La popolazione da **75 individui** raggiunge velocemente l'obiettivo dei mille punti
La popolazione da **100 individui** raggiunge velocemente 1000 punti ma non differenzia molto dai precedenti casi di test che lo fanno salvando molti meno individui

Popolazione	gen1	gen2	gen3	gen4	gen5	gen6	gen7	gen8	gen9	gen10
100	109	92	158	183	503	668	1000+	1000+	1000+	1000+
75	73	56	102	172	873	1000+	1000+	1000+	1000+	1000+
30	65	95	68	295	473	800	1000+	1000+	1000+	1000+
10	32	95	82	61	139	160	218	270	404	804

Abbiamo deciso quindi di usare un popolazione di 30 individui, la quale pur memorizzando meno individui in memoria riesce comunque ad ottenere risultati equiparabili a quelli di popolazioni più grandi. Inoltre per avere un evoluzione in tempi accettabili abbiamo preferito usare (Math.random()-0.5) * 2; nella funzione Random della classe RandomModel.

5 Considerazioni finali

Ai fini del nostro progetto possiamo quindi ritenerci soddisfatti del risultato ottenuto in quanto anche se non perfetto (qualche generazione tende a bloccarsi in minimi e massimi globali) il dinosauro riesce comunque ad apprendere dalle generazioni precedenti migliorando di volta in volta il suo score.