# 3 REINFORCEMENT LEARNING

In questo terzo capitolo verranno discussi i concetti teorici più rilevanti relativi al *Reinforcement Learning* (RL). Verranno forniti gli elementi di base necessari per comprendere l'operato esposto successivamente nel capitolo 6.

## 3.1 INTRODUZIONE

Il *Reinforcement Learning*, come anticipato nel capitolo 2, è una tecnica di *Machine Learning* nella quale un agente apprende come comportarsi all'interno di un ambiente a lui sconosciuto, eseguendo azioni e osservando i risultati ottenuti.

Negli ultimi anni la ricerca in questo settore ha compiuto notevoli sviluppi, principalmente nel settore dei giochi e videogiochi: in (Mnih et al., 2015) vengono migliorate notevolmente le prestazioni ottenute da altri algoritmi sui giochi della console ATARI 2600, mentre in (Silver et al., 2017), nel gioco da tavolo *Go*, l'algoritmo *Alpha Go Zero* batte con un punteggio di 100 a 0 l'algoritmo *Alpha Go*, il quale a sua volta aveva battuto il campione mondiale.

Tuttavia, lo sviluppo di questa tecnica si sta espandendo ad altre aree di interesse come la robotica, i sistemi di raccomandazione e gradualmente anche nella finanza.

Basandosi su due fonti autorevoli come (Sutton et al., 2018) e il corso dell'università di Londra *UCL* tenuto dal Professor *David Silver* (*Corso Reinforcement Learning*), di seguito verrà definito il concetto di *Reinforcement Learning* e verranno illustrati alcuni aspetti utili alla sua comprensione.

### 3.1.1 Definizione

Come affermato da Sutton, il *Reinforcement Learning* consiste nell'apprendere come comportarsi, nello specifico come collegare certe situazioni a determinate azioni, al fine di massimizzare la *reward* (ricompensa che si può ottenere) (Sutton et al., 2018).

L'azione da scegliere non viene comunicata dal sistema; al contrario, si devono effettuare diverse valutazioni per individuare in autono-

mia quali sono le azioni che generano la maggiore *reward* (*trial-and-error*). Inoltre, generalmente le azioni scelte possono influire nel lungo periodo sui *reward* futuri.

Queste due caratteristiche, ovvero la ricerca delle azioni attraverso tecniche di *trial-and-error* e l'effetto dei *reward* nel lungo periodo, costituiscono gli elementi distintivi del *Reinforcement Learning* (RL)

Il *Reinforcement Learning* (RL) è formato da due componenti che interagiscono fra loro:

- Un **Agente** interagisce con un *Environment* (ambiente) effettuando *azioni* $A_t$, le quali vengono scelte basandosi sulla valutazione degli *stati* $S_t$ e delle *reward* $R_t$ ricevute dall'ambiente in precedenza.
- Utilizzando un *modello*, che può essere più o meno noto, e una volta ricevuta un'azione dall'*Agente*, un **Environment** invia come *feedback* un nuovo stato adoperando la *Transition-Probability* e una *reward* attraverso una *Reward Function*, entrambi elementi del modello.

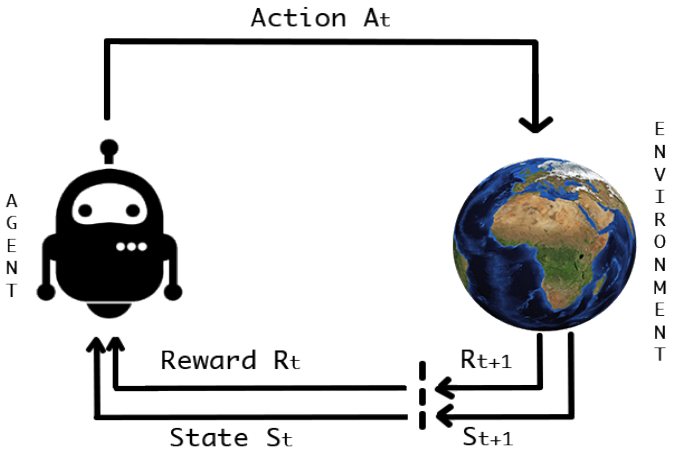


**Figura 3.1:** Interazione Agente e Ambiente nel *Reinforcement Learning*

**Funzionamento**
Al tempo t un agente riceve dall'ambiente una rappresentazione dello stato $S_t \in S$ e una *reward* $R_t \in R \subset \mathbb{R}$ e, basandosi su queste informazioni, sceglie un'azione $A_t \in A(S_t)$ da compiere. Al tempo successivo $t+1$, l'ambiente analizza l'azione ricevuta e invia all'agente come *feedback* una *reward* $R_{t+1}$ e un nuovo stato $S_{t+1}$.

L'interazione tra l'agente e l'ambiente implica una sequenza di azioni, *reward* e stati osservati nel tempo t = 1, 2, ..., T, che prende il nome di episodio e viene definita come segue:

$$\text{Episodio} = S_1, A_1, R_2, S_2, A_2, ..., S_T \tag{3.1}$$

### 3.1.2 Modello

Il modello utilizzato dall'ambiente permette di definire la *Reward Function* e le *State-Transition Probabilities*. Occorre chiarire che, poiché il modello può essere noto o sconosciuto, è possibile utilizzare due approcci diversi per risolvere un problema di *Reinforcement Learning*:

- nel primo approccio, chiamato *Model-Based*, viene utilizzata l'esperienza acquisita per costruire un modello che tenga traccia delle transizioni (ossia quando passare da uno stato $S_t$ a uno stato $S_{t+1}$), e delle *rewards* associate alle transizioni possibili. Una volta completo, il modello permette di scegliere l'azione ottimale.
- nel secondo approccio, chiamato *Model-Free*, l'esperienza acquisita tramite *trial-and-error* non viene utilizzata per la costruzione di un modello, bensì per apprendere alcune funzioni che permettono di individuare direttamente quale azione scegliere.

### 3.1.3 State-Transition Probabilities

Le *State-Transition Probabilities* sono componenti del modello che permettono di rappresentare la probabilità di passare dallo stato $s$ a uno stato successivo $s'$ compiendo un'azione $a$.

$$\mathcal{P}^a_{ss'} = p(s'|s,a) = \mathbb{P}[S_{t+1} = s'|S_t = s,\ A_t = a] \tag{3.2}$$

### 3.1.4 Reward Function

La funzione di *reward* valuta la qualità di un'azione in uno stato, attribuendo un valore numerico chiamato *reward* immediata.

$$\mathcal{R}^a_s = r(s,a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] \tag{3.3}$$