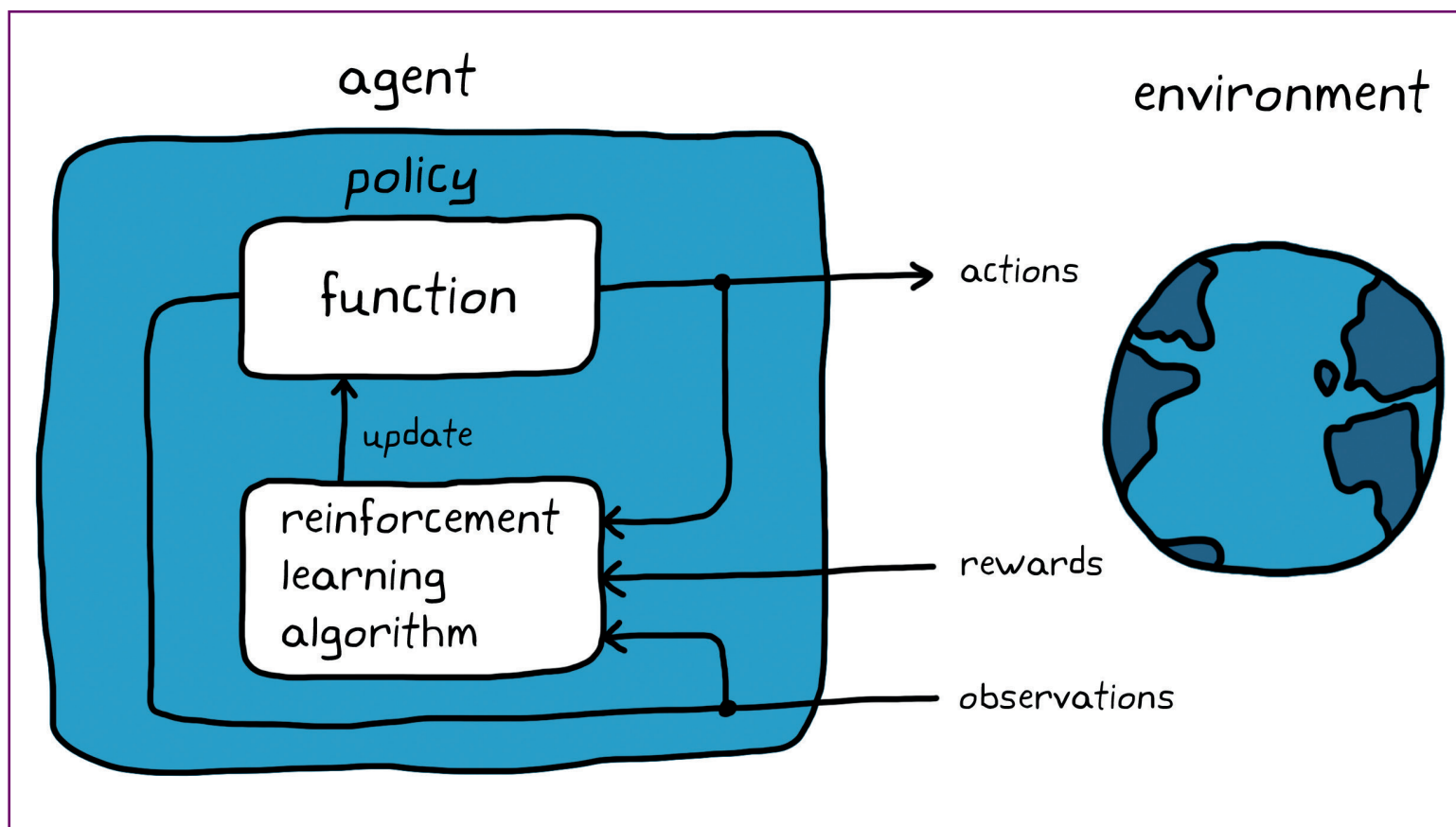


Tre cose da sapere sul Reinforcement Learning



Il Reinforcement Learning sta assumendo un ruolo sempre più importante per risolvere problemi decisionali complessi. Ma non tutti ne conoscono ancora le potenzialità. Il miglior modo per approcciare questa tecnologia è comprendere di cosa si tratta, quale sono le condizioni in cui può essere utilizzato e, infine, qual è il flusso di lavoro da seguire.

a cura del Gruppo Meccatronica di ANIE Automazione

Il Reinforcement Learning è una branca del Machine Learning in grado di risolvere problemi decisionali complessi. Per capire il potenziale ruolo che questa tecnologia può assumere in un progetto, dobbiamo prima di tutto rispondere a tre domande fondamentali:

- ✓ Che cos'è il reinforcement learning?
- ✓ Quando si usa il reinforcement learning?
- ✓ Qual è il flusso di lavoro da seguire per risolvere un problema di reinforcement learning?

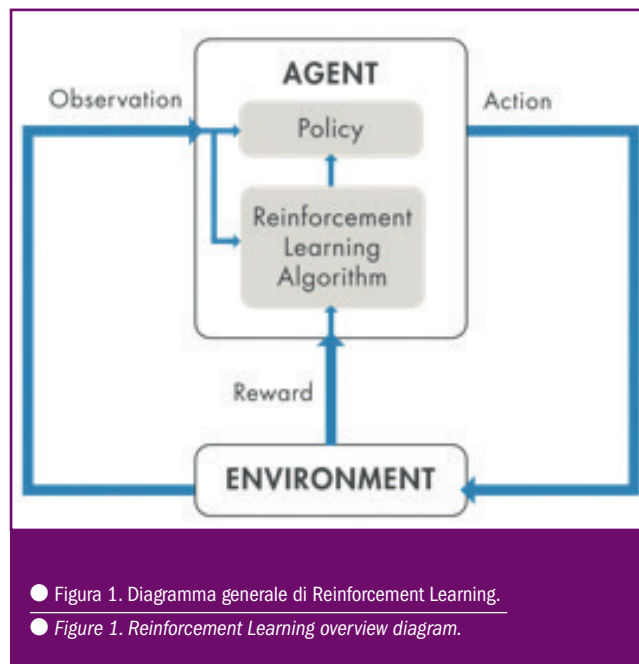
Che cosa si intende per reinforcement learning?

Il reinforcement learning aiuta un computer (agente) ad apprendere un comportamento tramite ripetute interazio-

ni di tipo “trial-and-error” (eseguite per tentativi ed errori) con un ambiente dinamico. Questo approccio consente all’agente di attuare una serie di decisioni in grado di massimizzare una metrica di ricompensa per l’attività, senza essere esplicitamente programmato per tale operazione e senza l’intervento dell’uomo.

Pensiamo di voler posteggiare l’auto con un sistema di parcheggio automatizzato. L’obiettivo di questa attività è che il computer del veicolo (l’agente) posteggi l’auto nell’apposito spazio.

La Figura 1 mostra una panoramica generale di un sistema di Reinforcement Learning. In questo scenario, tutto ciò che è al di fuori dell’agente, compresa la dinamica del veicolo, i veicoli circostanti, le condizioni atmosferiche, ecc., fa parte dell’ambiente. Durante l’addestramento, l’agente si serve delle letture provenienti dai sensori, come le telecamere e il GPS, e del lidar (osservazioni) per ottenere informazioni sullo stato dell’ambiente. Per apprendere una politica, ovvero per imparare a generare le azioni



● Figura 1. Diagramma generale di Reinforcement Learning.
● Figure 1. Reinforcement Learning overview diagram.

TECHNOLOGY

Three things you need to know about Reinforcement Learning

Reinforcement Learning is playing an increasingly important role in solving complex decision-making problems. But not everyone knows its potential yet. The best way to approach this technology is to understand what it is, under what conditions it can be used and, finally, what workflow should be followed.

Reinforcement Learning is a type of machine learning that can help solve tough decision-making problems. But to be able to understand the potential role of this technology in a project, we need to answer three key questions:

- ✓ What is reinforcement learning?
- ✓ When do I use reinforcement learning?
- ✓ What workflow should I follow to solve my reinforcement learning problem?

What is Reinforcement Learning?

Reinforcement learning helps a computer agent learn a behavior through repeated trial-and-error interactions with a dynamic environment.

This approach enables the agent to make a series of decisions that maximize a reward

metric for the task without being explicitly programmed to do so and without human intervention.

Consider parking a vehicle using an automated parking system. The goal of this task is for the vehicle computer (agent) to park in the correct parking spot. Figure 1 shows the general overview of a reinforcement learning system. In this scenario, everything outside the agent, including the vehicle dynamics, surrounding vehicles, weather conditions, etc., is part of the environment.

During training, the agent uses readings from sensors such as cameras and GPS, and lidar (observations) to get information about the environment state. To learn a policy, i.e., how to generate the correct actions like steering and braking, from these observations, the

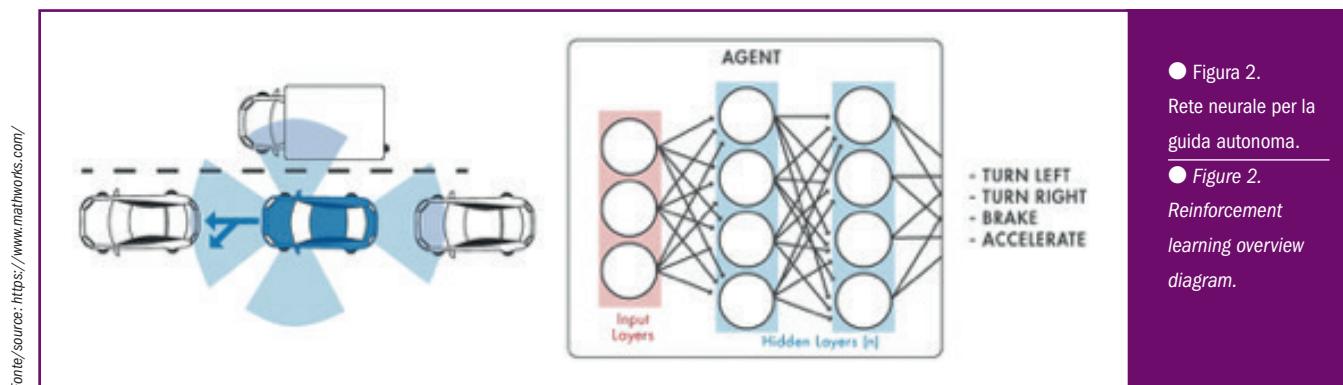
agent repeatedly tries to park the vehicle using a trial-and-error process, attempting to maximize a reward signal. The reward is used to evaluate the goodness of a trial and to guide the learning process.

Based on the collected observations, actions, and rewards, a training algorithm is responsible for tuning the agent’s policy. After training, the vehicle’s computer should be able to park using only the tuned policy and sensor readings.

When can Reinforcement Learning be used?

Several Reinforcement Learning training algorithms have been developed. Many of these are based on deep neural network (DNN) policies that allow the use of reinforcement learning in applications, such as automated driving (Figure 2), that are otherwise challenging to tackle with traditional algorithms. For example, traditional control design based on camera feedback





● Figura 2.
Rete neurale per la guida autonoma.
● Figure 2.
Reinforcement learning overview diagram.

corrette come la sterzata o la frenata, da queste osservazioni, l'agente tenta ripetutamente di posteggiare il veicolo usando un processo di tentativi ed errori definito "trial-and-error", cercando di massimizzare un segnale di ricompensa. La ricompensa viene utilizzata per valutare l'efficacia di un tentativo e per indirizzare il processo di apprendimento. In base alle osservazioni raccolte, alle azioni e alle ricompense, un algoritmo di addestramento procede alla regolazione della politica dell'agente. Dopo l'addestramento, il computer del veicolo dovrebbe essere in grado di effettuare il parcheggio servendosi esclusivamente della politica sottoposta a regolazione e delle letture dei sensori.

Quando è possibile usare il reinforcement learning?

Sono stati sviluppati diversi algoritmi di addestramento per il Reinforcement Learning. Molti di questi si basano sulle politiche delle reti neurali profonde (DNN) che consentono di usare il reinforcement learning in applicazioni quali la guida autonoma (Figura 2), difficilmente gestibili con gli algoritmi tradizionali. Ad esempio, la progettazione tradizionale dei sistemi di controllo basata sul feedback di fotocamere è piuttosto problematica in quanto

richiede una fase di pre-elaborazione molto lunga, ad esempio per l'estrazione delle feature, prima che i frame acquisiti possano essere effettivamente utilizzati in un ciclo di retroazione. Ciononostante, con le DNN la fase di estrazione delle feature entra a far parte della politica della rete neurale, consentendo così di giungere a soluzioni end-to-end.

D'altro canto, la problematica posta dagli algoritmi DNN, spesso costituiti da milioni di parametri, risiede proprio nella loro complessità, che non consente di spiegare le decisioni prese dalla rete e rende difficile stabilire delle garanzie di prestazioni formali. Un altro fattore da calcolare in relazione al Reinforcement Learning è che non è sottoposto al principio di sample efficiency. Per i progetti che richiedono un'esecuzione rapida con tempi di addestramento limitati, l'approccio del Reinforcement Learning, oneroso in termini di addestramento, pone ulteriori barriere. Ad esempio, il tempo di addestramento di questo approccio può andare da qualche minuto a svariati giorni, anche per applicazioni relativamente semplici. Infine, l'ampio numero di decisioni progettuali da prendere per definire un problema di Reinforcement Learning rende questa stessa operazione di definizione del problema

is challenging because there is a lot of preprocessing required, for example. for feature extraction, before the camera frames can be effectively used in a feedback loop. With DNNs however, the feature extraction step becomes part of the neural network policy, allowing for end-to-end-solutions. On the other hand, the challenge with DNN algorithms is their complexity - often consisting of millions of parameters - which makes it impossible to explain the decisions taken by the network and hard to establish formal performance guarantees. One additional factor to considering

reinforcement learning is that it is not sample efficient. For projects that require quick execution when limited training time is available, the training-intensive Reinforcement Learning approach presents additional barriers. For instance, training time for this approach can range from minutes to days even for relatively simple applications. Finally, the large number of design decisions that need to be made to set up a Reinforcement Learning problem makes correct problem set up tricky, often requiring multiple iterations to get right.

What workflow should you follow?

Figure 3 shows a general Reinforcement Learning workflow for training an agent (Figure 3).

✓ **Formulate Problem**- Define the task for the agent to learn, including how the agent interacts with the environment and any primary and secondary goals the agent must achieve.

✓ **Create Environment** - Define the environment within which the agent operates, including the interface between agent and environment and the environment dynamic model.

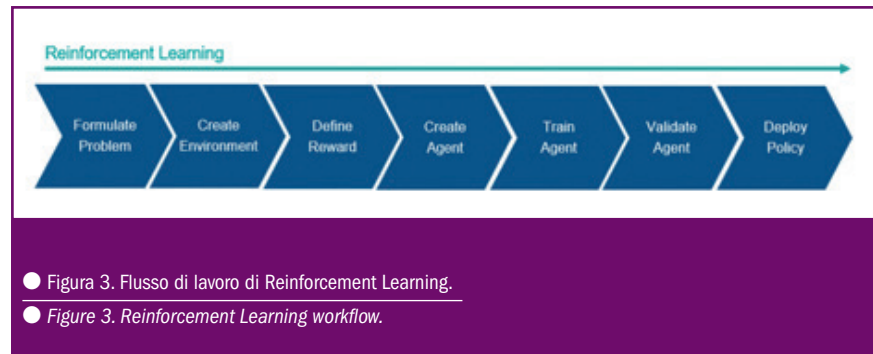
piuttosto complessa, in quanto spesso richiede iterazioni ripetute per arrivare alla soluzione corretta.

Il flusso di lavoro da seguire per risolvere un problema di reinforcement learning

Quello mostrato in figura 3 è un flusso di lavoro generale di Reinforcement Learning per l'addestramento di un agente (Figura 3).

- ✓ Formulazione del problema: definire l'attività che l'agente deve apprendere, comprese le modalità con cui l'agente interagisce con l'ambiente e gli eventuali obiettivi primari e secondari che l'agente deve raggiungere.
- ✓ Creazione dell'ambiente: definire l'ambiente in cui l'agente opera, compresa l'interfaccia tra agente e ambiente e il modello dinamico di ambiente.
- ✓ Definizione della ricompensa: specificare il segnale di ricompensa che l'agente usa per misurare le sue performance rispetto agli obiettivi dell'attività e le modalità con cui questo segnale viene calcolato dall'ambiente.
- ✓ Creazione dell'agente: creare l'agente, compresa la definizione di una rappresentazione della politica e la configurazione dell'algoritmo di addestramento dell'agente.
- ✓ Addestramento dell'agente: addestrare la rappresentazione della politica dell'agente usando l'ambiente, la ricompensa e l'algoritmo di addestramento dell'agente definiti.
- ✓ Convalida dell'agente: valutare le performance dell'agente addestrato simulando contemporaneamente agente e ambiente.
- ✓ Distribuzione della politica: distribuire la rappresentazione della politica addestrata usando, per esempio, codice GPU generato.

L'addestramento di un agente tramite Reinforcement Learning è un processo iterativo. Le decisioni e i risultati



Fonte / source: <https://www.mathworks.com/>

delle fasi successive potrebbero richiedere il ritorno a una fase precedente del flusso di lavoro di addestramento. Ad esempio, se il processo di addestramento non converge su una politica accettabile entro un lasso di tempo ragionevole, potrebbe essere necessario aggiornare uno dei seguenti elementi prima di addestrare nuovamente l'agente:

- ✓ Impostazioni di addestramento
- ✓ Configurazione dell'algoritmo di addestramento
- ✓ Rappresentazione della politica
- ✓ Definizione del segnale di ricompensa
- ✓ Segnali di osservazione e azione
- ✓ Dinamica dell'ambiente

Ad oggi, gli strumenti dedicati al reinforcement learning possono essere utili ad ampliare e a implementare più velocemente i propri controllori e gli algoritmi di decision-making.

Indipendentemente dalla scelta dello strumento, per chi è interessato a introdurre la tecnologia del Reinforcement Learning nel proprio progetto ma non l'ha mai usata prima, il modo migliore per iniziare è rispondere a queste tre domande e capire se questo è l'approccio giusto. ●

- ✓ Define Reward - Specify the reward signal that the agent uses to measure its performance against the task goals and how this signal is calculated from the environment.
- ✓ Create Agent- Create the agent, which includes defining a policy representation and configuring the agent learning algorithm.
- ✓ Train Agent- Train the agent policy representation using the defined environment, reward, and agent learning algorithm.
- ✓ Validate Agent - Evaluate the performance of the trained agent by simulating the agent and environment together.
- ✓ Deploy Policy - Deploy the trained policy

representation using, for example, generated GPU code.
 Training an agent using Reinforcement Learning is an iterative process. Decisions and results in later stages may require returning to an earlier stage in the learning workflow. For example, if the training process does not converge to an acceptable policy within a reasonable amount of time, one of these may need updating before retraining the agent:

- ✓ Training settings
- ✓ Learning algorithm configuration
- ✓ Policy representation

- ✓ Reward signal definition
- ✓ Action and observation signals
- ✓ Environment dynamics

Today, tools focused on Reinforcement Learning can help you ramp up and implement controllers as well as decision-making algorithms faster. Regardless of the choice of tool, if you are interested in using reinforcement learning technology for your project but have not used it before, the right place to start is by answering these three questions to determine if it is the right approach for you. ●