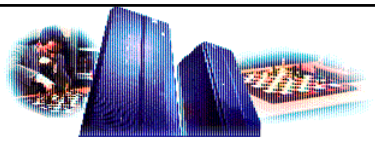
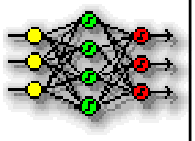




Sistemi Intelligenti
Learning: l'apprendimento degli agenti

Alberto Borghese

Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
Dipartimento di Scienze dell'Informazione
borghese@dsi.unimi.it

Riassunto

- Gli agenti
- Il Reinforcement Learning
- Gli elementi del RL
- Un esempio: tris

Why agents are important?

Agente (software): essere software che svolge servizi per conto di un altro programma, solitamente in modo automatico ed invisibile. Tali software vengono anche detti agenti intelligenti

"They are seen as a natural metaphor for conceptualising and building a wide range of complex computer systems (the world contains many passive objects, but it also contains very many *active* components as well);

They cut across a wide range of different technology and application areas, including telecoms, human-computer interfaces, distributed systems, WEB and so on;

They are seen as a natural development in the search for ever-more powerful abstractions with which to build computer systems."

Agente

- Può scegliere un'azione sull'ambiente tra un insieme continuo o discreto.

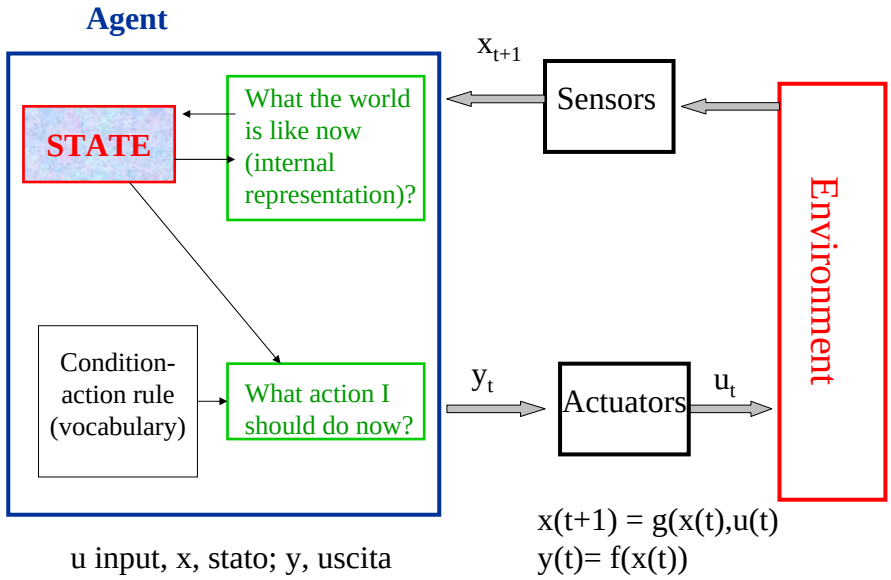
- L'azione dipende dalla situazione. La situazione è riassunta nello **stato** del sistema.

- L'agente monitora continuamente l'ambiente (input) e modifica continuamente lo stato.

- La scelta dell'azione è non banale e richiede un certo grado di "intelligenza".

- L'agente ha una memoria "intelligente".

Schematic diagram of an agent



u input, x, stato; y, uscita

$x(t+1) = g(x(t),u(t)$
$y(t)= f(x(t))$

L'agente

- Inizialmente l'attenzione era concentrata sulla progettazione dei sistemi di "controllo". Valutazione, sintesi…
- L'intelligenza artificiale e la "computational intelligence" hanno consentito di spostare l'attenzione sull'apprendimento delle strategie di controllo e più in generale di comportamento.
- Macchine dotate di meccanismi (algoritmi, SW), per apprendere.

I vari tipi di apprendimento

$x(t+1) = f[x(t),u(t)]$ Ambiente
$y(t) = \mathbf{g}[x(t)]$ Agente

Supervisionato (learning with a teacher). Viene specificato per ogni pattern di input, il pattern desiderato in output.

Non-supervisionato (learning without a teacher). Estrazione di similitudine statistiche tra pattern di input. Clustering. Mappe neurali.

Apprendimento con rinforzo (reinforcement learning, learning with a critic). L'ambiente fornisce un'informazione puntuale, di tipo qualitativo, ad esempio success or fail.

I - Apprendimento supervisionato

$\mathbf{y}(t) = g[w; \mathbf{u}(t)]$

$u \in R^n \quad y \in R^m$

$\min_{\{w\}} J(.) \qquad J = \left\| Y^D - g(W;U) \right\|$

Y^D è l'uscita desiderata nota.

- Si tratta di un problema di minimizzazione di una cifra di merito, J(.), nello spazio dei parametri W, che caratterizzano l'agente.
- Appredimento supervisionato si applica a problemi di *classificazione* o di *regressione*.

Soluzione iterativa: Obbiettivo: se esiste una soluzione, trovare ΔW in modo iterativo tale che l'insieme dei pesi W^{nuovo} ottenuto come:

$$W^{nuovo} = W^{vecchio} + \Delta W$$

dia luogo a un errore sulle uscite di norma minore che con $W^{vecchio}$

- **Capacità di generalizzazione**
- Modello globale vs insieme di modelli locali.

Esempio Apprendimento Supervisionato

Task di classificazione
Uscita booleana

Riconoscimento della scrittura (progetto su un problema ridotto)

1. Determinare il tipo di “training example” adatti al problema.

2. Definire un training set rappresentativo. Training set = coppie di input / output (ruolo del supervisore umano).

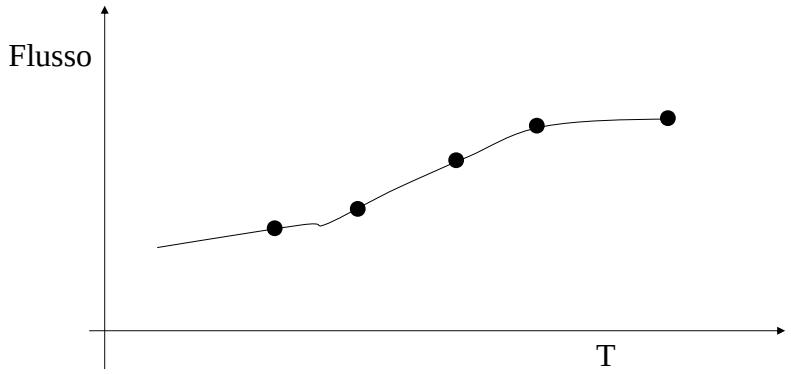
3. Preprocessing. Determinare quali feature è opportuno estrarre dai pattern in ingresso perchè costituiscano l’input per l’agente.

4. Determinare i dati utilizzati per l’apprendimento (quali -> rappresentatività; e quanti -> tanti ma non troppi….).

5. Determinare le modalità di apprendimento: empirical risk minimization, structural risk minimization….

6. Determinare come valutare l’apprendimento (e.g. through cross-validation).

Apprendimento supervisionato



Controllo della portata di un condizionatore in funzione della temperatura. “Imparo” una funzione continua a partire da alcuni campioni.

Clustering

- Raggruppamento degli input in classi omogenee tra loro.
  - Raggruppamento per colore
  - Raggruppamento per forme
  - Raggruppamento per tipi
  - …..

Apprendimento non-supervisionato: Clustering & Classificazione

Descrizione numerica dell'oggetto:
altezza, colore, forma, posizione, ….

SPAZIO DEI CAMPIONI / DELLE FEATURES

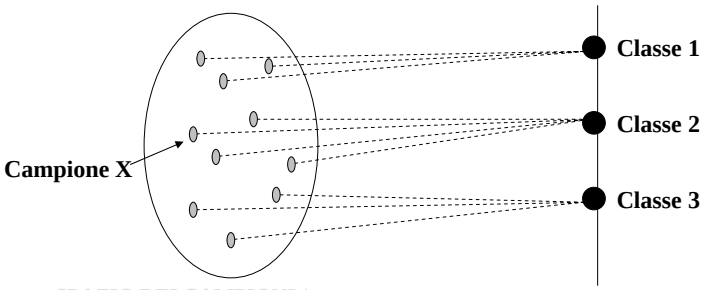
Classificatore

Classi omogene dell'oggetto:
(classe A, classe B, …)

SPAZIO DELLE CLASSI

Classificazione

Un'interpretazione geometrica:
Mappatura dello spazio dei campioni nello spazio delle classi.



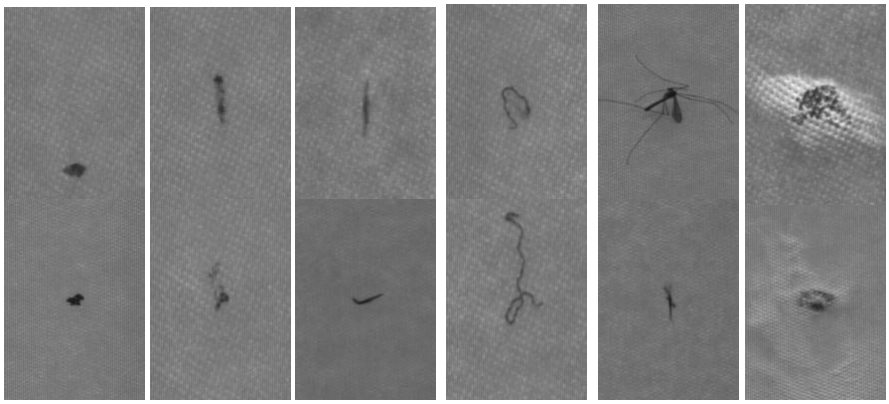
SPAZIO DEI CAMPIONI / DELLE FEATURES (CARATTERISTICHE)

SPAZIO DELLE CLASSI

Che differenza c'è rispetto al clustering?

Cos'è un concetto?

Riconoscimento difetti in linee di produzione
(finanziato da Electronic Systems: 2006-2007)



regolari | irregolari | allungati | fili | insetti | macchie su denso

Anche: pieghe, arruffati

Difetti – Classificazione real-time e apprendimento.
Feature extraction + SVM

Esempio di clustering

Clustering -> Indicizzazione

Tesi: ricerca immagini su WEB.

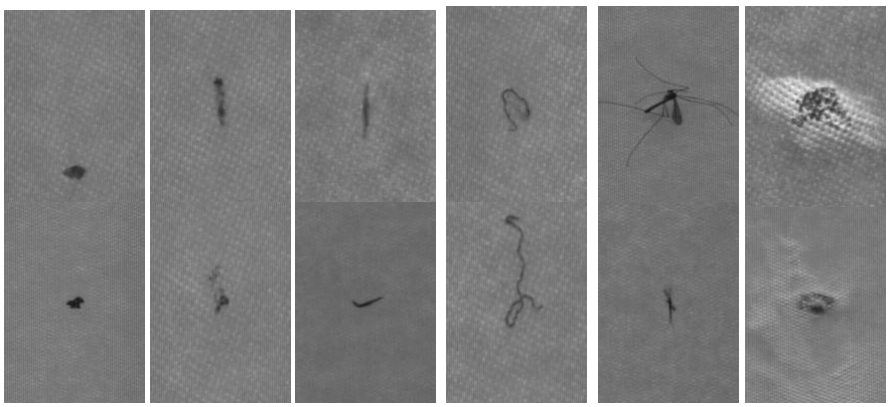
A cosa serve il clustering?

- Compressione dati (telecomunicazioni, immagini, …);
- Segmentazione (bio)immagini;
- Controllo robot;
- Indicizzazione

A cosa serve la classificazione?

- Riconoscimento automatico;
- Pattern recognition;
- Ricostruzione superfici;
- …

Riconoscimento difetti in linee di produzione
(finanziato da Electronic Systems: 2006-2007)



regolari | irregolari | allungati | fili | insetti | macchie su denso

Anche: pieghe, arruffati

Difetti – Classificazione real-time e apprendimento.
Feature extraction + SVM

Riassunto

- Gli agenti
- Il Reinforcement Learning
- Gli elementi del RL
- Un esempio: tris

Reinforcement learning

Nell'apprendimento supervisionato, esiste un "teacher" che dice al sistema quale è l'uscita corretta (learning with a teacher). Non sempre e' possibile.

Spesso si ha a disposizione solamente un'informazione qualitativa (a volte binaria, giusto/sbagliato successo/fallimento), puntuale.

Questa è un'informazione qualitativa**.**

L'informazione disponibile si chiama ***segnale di rinforzo.*** *Non dà alcuna informazione su come aggiornare il comportamento dell'agente (e.g. i pesi). Non è possibile definire una funzione costo o un gradiente.*

Obbiettivo: creare degli agenti "intelligenti" che abbiano una "machinery" per apprendere dalla loro esperienza.

Reinforcement Learning: caratteristiche

- Apprendimento mediante interazione con l'**ambiente**. Un agente isolato non apprende.
- L'apprendimento è funzione del raggiungimento di uno o più **obbiettivi.**
- Non è necessariamente prevista una ricompensa ad ogni istante di tempo.
- Le azioni vengono valutate mediante la ricompensa a lungo termine ad esse associata (**delayed reward**). Il meccanismo di ricerca delle azioni migliori è imparentato con la ricerca euristica: **trial-and-error**.
- **L'agente sente l'input, modifica lo stato e genera un'azione che massimizza la ricompensa a lungo termine.**

Exploration vs Exploitation

Esplorazione (exploration) dello spazio delle azioni per scoprire le azioni migliori. Un agente che esplora solamente raramente troverà una buona soluzione.

Le azioni migliori vengono scelte ripetutamente (exploitation) perchè garantiscono ricompensa (reward). Se un agente non esplora nuove soluzioni potrebbe venire surclassato da nuovi agenti più dinamici.

Occorre non interrompere l'esplorazione.

Occorre un approccio statistico per valutare le bontà delle azioni.

Exploration ed exploitation vanno bilanciate. Come?

Dove agisce un agente?

- L'agente ha un comportamento goal-directed ma agisce in un ambiente incerto non noto a priori o parzialmente noto.
- Esempio: planning del movimento di un robot.
- Un agente impara interagendo con l'ambiente. Planning può essere sviluppato mentre si impara a conoscere l'ambiente (mediante le misure operate dall'agente stesso). La strategia è vicina al trial-and-error.

L'ambiente

Model of the environment. E' uno sviluppo relativamente recente. Da valutazione implicita dello svolgersi delle azioni future (trial-and-error) a valutazione esplicita mediante modello dell'ambiente della sequenza di azioni e stati futuri (planning) = dal sub-simblico al simbolico; emerging intelligence.

Incorporazione di AI:

- Planning (pianificazione delle azioni).
- Viene rinforzato il modulo di pianificazione dell'agente.

- Incorporazione della conoscenza dell'ambiente:
- Modellazione dell'ambiente (non noto o parzialmente noto).

Relazione con l'AI

- Gli agenti hanno dei goal da soddisfare. Approccio derivato dall'AI.
- Nell'apprendimento con rinforzo vengono utilizzati strumenti che derivano da aree diverse dall'AI:
  - Ricerca operativa.
  - Teoria del controllo.
  - Statistica.
- L'agente impara facendo. Deve selezionare i comportamente che ripetutamente risultano favorevoli a lungo termine.

Esempi

Un giocatore di scacchi. Per ogni mossa ha informazione sulle configurazioni di pezzi che può creare e sulle possibili contro-mosse dell'avversario.

Una gazzella in 6 ore impara ad alzarsi e correre a 40km/h.

Come fa un robot veramente autonomo ad imparare a muoversi in una stanza per uscirne? (cf. competizione Robocare@home).

Come impostare i parametri di una raffineria (pressione petrolio, portata….) in tempo reale, in modo da ottenere il massimo rendimento o la massima qualità?

Caratteristiche degli esempi

Parole chiave:

- Interazione con l'ambiente. L'agente impara dalla propria esperienza.
- Obbiettivo dell'agente.
- Incertezza o conoscenza parziale dell'ambiente.

Osservazioni:

- Le azioni modificano lo stato (la situazione), cambiano le possibilità di scelta in futuro (delayed reward).
- L'effetto di un'azione non si può prevedere completamente.
- L'agente ha a disposizione una valutazione globale del suo comportamento. Deve sfruttare questa informazione per migliorare le sue scelte. Le scelte migliorano con l'esperienza.
- I problemi possono avere orizzonte temporale finito od infinito.

Riassunto

- Gli agenti
- Il Reinforcement Learning
- Gli elementi del RL
- Un esempio: tris

I tue tipi di rinforzo

L'agente deve scoprire quale azione (policy) fornisca la ricompensa massima provando le varie azioni (trial-and-error) sull'ambiente.

"Learning is an adaptive change of behavior and that is indeed the reason of its existence in animals and man (K. Lorentz, 1977).

Rinforzo puntuale istante per istante, azione per azione (condizionamento classico).

Rinforzo puntuale "una-tantum" (condizionamento operante), viene rinforzato una catena di azioni, un comportamento.

Il Condizionamento classico

L'agente deve imparare una (o più) trasformazione tra input e output. Queste trasformazioni forniscono un comportamento che l'ambiente premia.

Il segnale di rinforzo è sempre lo stesso per ogni coppia input – output.

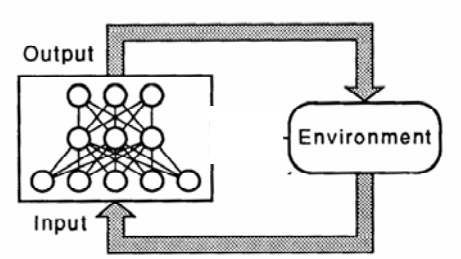
Esempio: risposte riflesse Pavolviane. Campanello (stimolo condizionante) prelude al cibo. Questo induce una risposta (salivazione). La risposta riflessa ad uno stimolo viene evocata da uno stimolo condizionante.

Stimolo-Risposta. Lo stimolo condizionante (campanello = input) induce la salivazione (uscita) in risposta al campanello.

Condizionamento operante

Reinforcement learning (operante).

Interessa un comportamento. Una sequenza di input / output che può essere modificata agendo sui parametri che definiscono il comportamento dell'agente. Il condizionamento arriva in un certo istante di tempo (spesso una-tantum) e deve valutare tutta la sequenza temporale di azioni, anche quelle precedenti nel tempo.



Comportamenti
=
Sequenza di azioni

Gli attori del RL

Policy. Descrive l'azione scelta dall'agente: mapping tra stato (input dall'ambiente) e azioni. Funzione di controllo. Le policy possono avere una componente stocastica. Viene utilizzato un modello adeguato del comportamento dell'agente (e.g. tabella, funzione continua parametrica…).

Reward function. Ricompensa **immediata**. Associata all'azione intrapresa in un certo stato. Può essere data al raggiungimento di un goal (esempio: successo / fallimento). E' uno scalare. Rinforzo primario.

Value function. "Cost-to-go". Ricompensa a **lungo termine**. Somma dei reward: costi associati alle azioni scelte istante per istante + costo associato allo stato finale. Orizzonte temporale ampio. Rinforzo secondario.

Ambiente. Può essere non noto o parzialmente noto. L'agente deve costruirsi una rappresentazione dell'ambiente.

- Quale delle due è più difficile da ottenere?
- L'agente agisce per massimizzare la funzione Value o Reward?

Proprietà del rinforzo

L'ambiente o l'interazione può essere complessa.

Il rinforzo può avvenire solo dopo una più o meno lunga sequenza di azioni (delayed reward).

E.g. agente = giocatore di scacchi.
ambiente = avversario.

Problemi collegati:
temporal credit assignement.
structural credit assignement.

L'apprendimento non è più da esempi, ma dall'osservazione del proprio comportamento nell'ambiente.

Riassunto

•Reinforcement learning. L'agente viene modificato, rinforzando le azioni che sono risultate buone a lungo termine. E' quindi una classe di algoritmi iterativi.

•Self-discovery of a successful strategy (it does not need to be optimal!). La strategia (di movimento, di gioco) non è data a-priori ma viene appresa attraverso **trial-and-error**.

•Credit assignement (temporal and structural).

•Come possiamo procedere in modo efficiente nello scoprire una strategia di successo? Cosa vuol dire modificare l'agente?

Riassunto

- Gli agenti
- Il Reinforcement Learning
- Gli elementi del RL
- Un esempio: tris (progetto)

Gli attori del RL

Policy. Descrive l'azione scelta dall'agente: mapping tra stato e azioni. Funzione di controllo.

Ambiente. Descrive tutto quello su cui agisce la policy. Generalmente è non noto o parzialmente noto e deve essere "scoperto". Non viene modellato esplicitamente, ma viene stimata la sua reazione alla policy mediante la value function.

Reward function. Ricompensa **immediata**. Associata all'azione intrapresa in un certo stato. Può essere data al raggiungimento di un goal (esempio: successo / fallimento). E' uno scalare associato allo stato dell'agente. Rinforzo primario.

Value function. "Cost-to-go". Ricompensa a **lungo termine**. Somma dei reward + costi associati alle azioni scelte istante per istante. Orizzonte temporale ampio. Rinforzo secondario. Ricompensa attesa.

Ciclo dell'agente (le tre fasi sono sequenziali):

1) Implemento una policy
2) Aggiorno la Value function
3) Aggiorno la policy.

Commenti

Con molti stati è impossibile esplorarli tutti ➔ Generalizzazione. Dalle funzioni combinatorie alle funzioni continue.

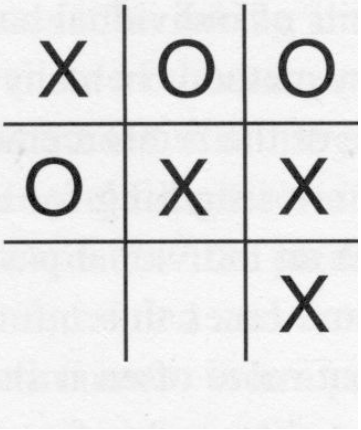
Può essere inserita della conoscenza a-priori sia sulla policy che sulla Value function.

Apprendimento della strategia per il gioco del tris

1) We can use classical game theory solution like minmax. This can work for the optimal opponent, not for "our" actual opponent.

2) We can use dynamic programming optimization, but we need a model of the opponent.

3) We can try a policy and see what happens for many games (evolutionary style, exhaustive search).



What can we do?

Come impostare il problema mediante RL?

State – configuration of ‘X’ and ‘O’

Value (of the state) – probability of winning associated to that state.

Which is the probability of a state in which we have 3 ‘X’ in a row (or column or diagonal)?
Which is the probability of a state in which we have 3 ‘0’ in a row (or column or diagonal)?

We set all the other states to 0.5.

Come decidere la mossa?

Supponiamo di essere in una configurazione non terminale.

Per ogni mossa valida, possiamo valutare il valore della nuova configurazione che si verrebbe a trovare. Come?

Possiamo occasionalmente scegliere delle mosse esploratorie. Quando non ha senso scegliere delle mosse esploratorie?

Dobbiamo perciò capire qual’è il valore delle diverse configurazioni della scacchiera.

Come stimare il valore di ogni configurazione?

$V(s(t)) \leftarrow V(s(t)) + \alpha\,[V(s(t+1)) - V(s)]$

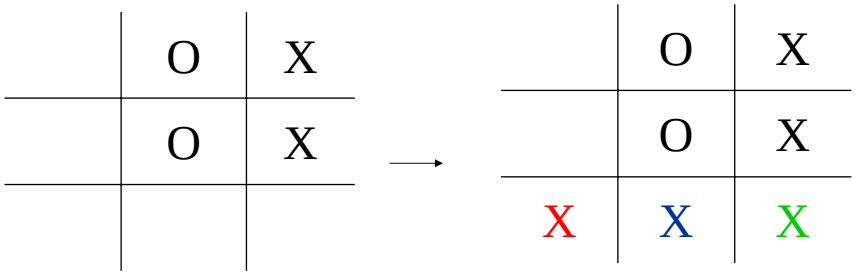
Tendo ad avvicinare il valore della mia configurazione al valore della configurazione successiva.

Esempio di *temporal difference learning*.

Diminuendo α con il numero di partite, la policy converge alla policy ottima per un avversario fissato (cioè che utilizzi sempre la strategia, ovvero sia la stessa distribuzione statistica di mosse).

Diminuendo α con il numero di partite, ma tenendolo > 0, la policy converge alla policy ottima anche per un avversario che cambi molto lentamente la sua strategia.

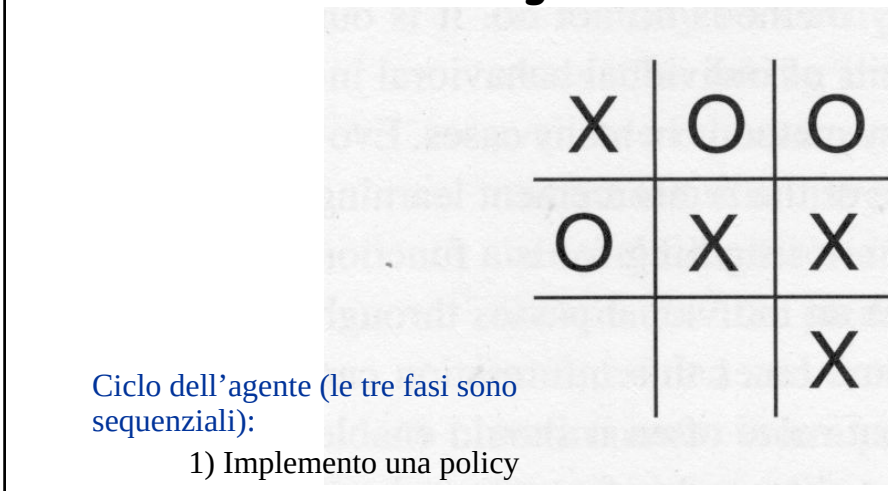
Esempio



X – scelta perdente -> dopo la mossa dell'avversario ho uno stato con value function = 0.
X – scelta neutrale -> dopo la mossa dell'avversario ho uno stato con value function intermedia (pareggio).
X – scelta vincente -> vado in una configurazione con value function = 1.

Cambio la policy e rivaluto la Value function.

Cosa fa l'agente?



Ciclo dell'agente (le tre fasi sono sequenziali):

1) Implemento una policy
2) Aggiorno la Value function
3) Aggiorno la policy.

Riflessioni su RL ed il gioco del tris

Supponete che l'agente dotato di RL giochi, invece che con un avveresario, contro sé stesso. Cosa pensate che succeda? Secondo voi imparerebbe una diversa strategia di gioco?

Molte posizioni del tris sembrano diverse ma sono in realtà la stessa per effetto delle simmetrie. Come si può modificare l'algoritmo di RL (definizione dello stato) per sfruttare le simmetrie? Come si può migliorare il meccanismo di apprendimento?

Supponiamo che l'avversario non sfrutti le simmetrie. In questo caso noi possiamo sfruttarle? E' vero che configurazioni della scacchiera equivalenti per simmetria devono avere la stessa funzione valore.

Potete pensare a modi per migliorare il gioco dell'agente? Potete pensare a metodi migliori (più veloci, più robusti….) perché un agente impari a giocare a tris?

Riassunto

- Gli agenti
- Il Reinforcement Learning
- Gli elementi del RL
- Un esempio: tris (progetto)