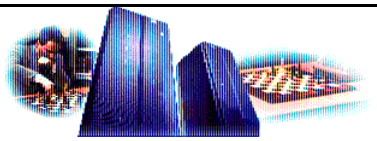
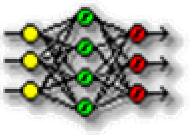




L'intelligenza biologica

Apprendimento con Rinforzo

Alberto Borghese
Università degli Studi di Milano
Laboratorio di Motion Analysis and Virtual Reality (MAVR)
Dipartimento di Scienze dell'Informazione
borghese@dsi.unimi.it

Sommario

Il neurone, modelli deterministici (L-system) e stocastici (frattali).
Reti Neurali.
RBF: reti neurali con neuroni a base radiale.
Mappe topologiche e clustering.
Apprendimento con Rinforzo (Reinforcement Learning).
- Che cos'è il Reinforcemente Learning?
- Modalità di apprendimento.
- Apprendimento su sistemi dinamici.

La corteccia

Evoluzione storica - I

•1943	Warren McCulloch (neurofisiologo) & Walter Pitts (matematico) •Modello di neurone elementare a soglia	
•1949	Donald Hebb •Teorie sull'apprendimento	albori
•1960	Widrow & Hoff •Delta rule; Adaline	
•1961	Steinbuck •Memorie associative	
•1961	Caianiello •Teoria statistica	
•1962	Rosenblatt •Perceptrone; perceptron learning rule	periodo "romantico"
•1969	Minsky & Papert •Problemi di apprendimento del perceptrone	

Evolzione storica - II

•1968	Anderson •Memorie associative	separazione del connessionismo dall'intelligenza artificiale simbolica
•1974	Kohonen •Memorie associative, mappe autoorganizzanti	
•1983	Barto, Sutton and Anderson •Reinforcement Learning	
•1983	Hinton e Sejnoswky •Unità stocastiche	
•1985	Amit •Spin glass	
•1985	Rumelhart, Hinton & Parker •Back propagation (perceptrone multi-layer)	
•1974	Werbos (economista) •Back propagation	
•1989	Kohonen •Memorie associative, mappe autoorganizzanti	"revival"
•1998	Vapnik •Teoria dell'apprendimento e Support Vector Machines per problemi di classificazione	

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 giusto/sbagliato successo/fallimento.

Questa è un'informazione qualitativa ➔ ***learning with a critic.***

L'informazione disponibile si chiama ***segnale di rinforzo.*** *Non dà alcuna informazione su come aggiornare 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.

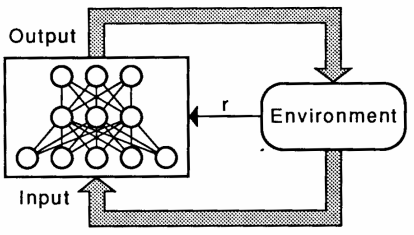
Formalizzazione

- Eseguire delle azioni sul mondo (Output)
- Osservare lo stato del mondo (Inut).

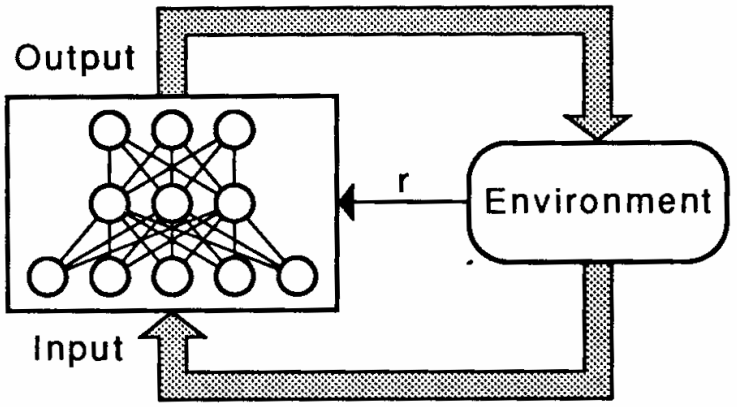
Riceve un'informazione **puntuale** sul successo (fallimento), r.

Imparare una politica di controllo (Output = f(Input)).

Come?



Reinforcement learning



Rete: Funzione non-lineare multi-input / multi-output.
Ambiente: scalare, r (reward / penalty or success / fail).

I tue tipi di rinforzo

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

Rinforzo puntuale "una-tantum" (condizionamento operante).

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

Il Condizionamento classico

Condizionamento classico. La risposta riflessa ad uno stimolo incondizionato viene evocata da uno stimolo condizionante.

Esperimenti di Pavlov. Campanello (stimolo condizionante), cibo (stimolo), risposta (salivazione).

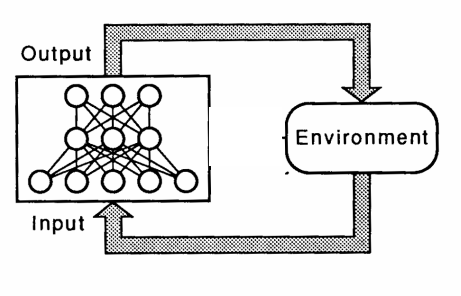
Stimolo-Risposta. Lo stimolo condizionante triggera una risposta condizionata.

Cf. Apprendimento Hebbiano.

Condizionamento operante

Condizionamento operante (reinforcement learning).

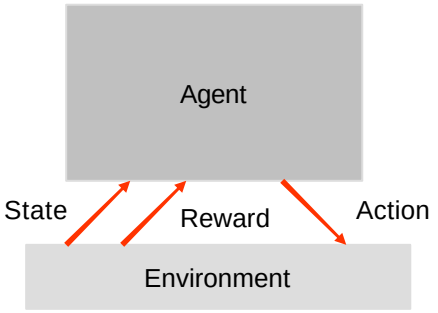
Interessa un comportamento. Una catena di input / output che può essere modificata agendo sul sistema. Il condizionamento arriva in un certo istante di tempo ed agisce a ritroso sul sistema di controllo.



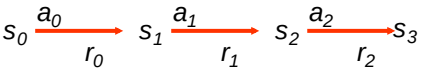
La Funzione Rinforzo

Viene ripetuto il ciclo:

- Eseguire delle azioni sul mondo {a}
- Osservare lo stato del mondo {s}.
- Osservare la ricompensa {r}.



Imparare una politica di controllo (a= f(s)) tale che viene massimizzata la ricompensa totale ("life reward":



$r_0 + \gamma r_1 + \gamma^2 r_2 \ldots$

Per ogni stato, con $0 < \gamma <$

Da dove vengono gli $\{r_i\}$?

NB: Unsupervised learning. Delayed reward

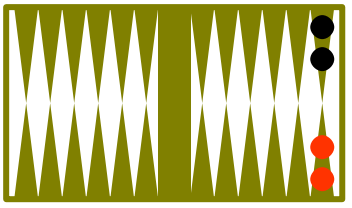
Back-gammon through RL (G. Tesauro, 1995)

Numero di situazioni:

- Configurazioni della scacchiera (10^{20})

Azioni:

- Mosse



Reward:

- +100 se vince
- - 100 se perde
- 0 per tutti gli altri stati

- Rete neurale allenata giocando 1,5 milioni di partite da sola.

Attualmente la macchina gioca a livello dei giocatori migliori.

Aspetti comuni dell'apprendimento

"Stimolo ad agire".
Stato. Input.
Risposta. Output.
"Stimolo". Reward / penalty

Variazione della relazione input/output (funzione di controllo) mediante ad esempio aggiornamento dei pesi sinaptici, se il controllo viene modellato con una rete neurale.

La variazione è attivata dallo stimolo condizionante. Come trasformare uno stimolo eterogeneo rispetto alla risposta in uno stimolo efficace?

Tipi di problemi di apprendimento

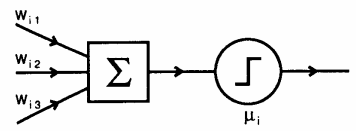
I. Ambiente deterministico, senza dinamica.
II. Ambiente stocastico, senza dinamica.
III. Ambiente deterministico e/o stocastico, con dinamica

I) Apprendimento con rinforzo di pattern di input/output

Nel caso più semplice, il segnale di rinforzo è disponibile per ogni coppia di segnali ingresso/uscita. Esiste cioè una trasformazione definita tra ingresso e uscita che la rete deve imparare.

Questa è simile alla situazione di apprendimento supervisionato. Rosenblatt **perceptron learning rule (neurone binario a soglia)**:

$$\Delta w_{ij} = h\Theta\left(1 - y_i^D y_i\right) y_i^D u_j$$



$\Theta(\bullet) \Rightarrow (1-y_i^D y_i) \Rightarrow y_i^D y_i$ decide solo se la correzione deve essere effettuata, può essere interpretato come yes/no.

I) Apprendimento con rinforzo di pattern di input/output – funzioni di attivazione non-lineari

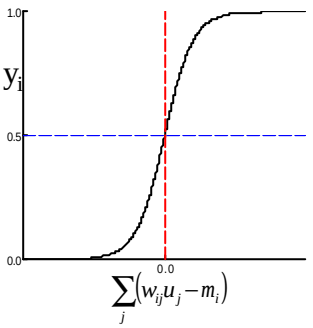
$$J = E(\mathbf{w}) = \frac{1}{2}\sum_p \left[\sum_i \left(y_{ip}^D - y_{ip}\right)^2 = \frac{1}{2}\sum_i \left(y_{ip}^D - \left(\sum_j w_{ij} u_{jp}\right) \right)^2 \right]$$

$$\Delta w_{ijp} = +h\left(y_i^D - y_i\right) y_i (1 - y_i) u_j$$

Possiamo supporre che le condizioni: $y_{ip} > y_{ip}^D$ e $y_{ip} < y_{ip}^D$ attivino l'apprendimento.

↓

$$\Delta w_{ijp} = \Theta\left(\left| y_i^D - y_i \right|\right) f(u_i, y_i)$$



II) Apprendimento con rinforzo in ambienti stocastici

Questo tipo è generalmente applicato ad ambienti stocastici. In questo caso una particolare coppia ingresso/uscita determina una certa ***probabilità*** che il rinforzo sia positivo. La probabilità è comunque fissata (stazionaria) per ogni coppia ingresso/uscita.

Esempio two-armed bandit problem.

Massimizzare il reward, minimizzando il rischio.

Stochastic learning automata.

Trade-off tra exploration ed exploitation.

III) Apprendimento con rinforzo del comportamento di sistemi dinamici

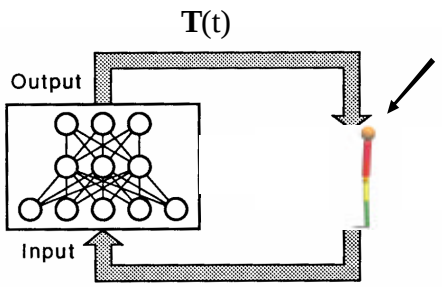
Nel caso più generale l'ambiente stesso è governato da leggi dinamiche molto complesse. Sia il segnale di rinforzo che lo stato atttuale (input al controllore) dipendono dalla storia passata delle uscite della rete.

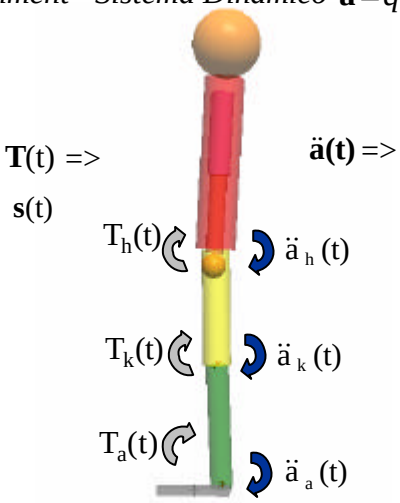
L'applicazione più classica è quella del gioco, dove l'ambiente rappresenta l'altro giocatore o gli altri giocatori. Se si considera per esempio il gioco degli scacchi, il segnale di rinforzo (vittoria o sconfitta) è inviato alle rete solo dopo un numero elevato di mosse. Applicazioni simili sono state sviluppate anche in psicologia dinamica.

Più recentemente un numero sempre crescente di applicazioni sono state sviluppate nell'ambito del controllo di sistemi complessi in ambienti non noti.

Apprendimento del controllo della postura di un robot umanoide.

"Environment" Sistema Dinamico $\ddot{\mathbf{a}} = q(\mathbf{T}, \mathbf{a})$



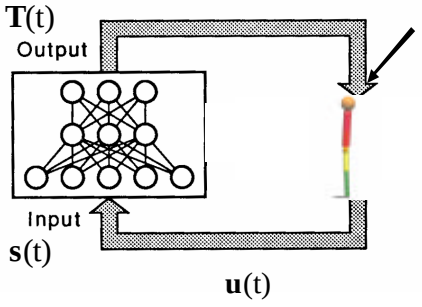


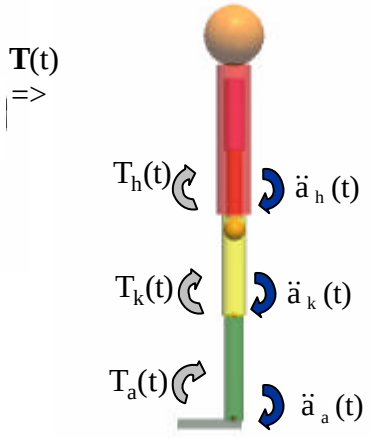
Da $\ddot{\mathbf{a}}(t)$ tramite integrazione ottengo: $\dot{\mathbf{a}}(t)$ e $\mathbf{a}(t)$

Considero lo stato, $\mathbf{s}(t) = [\dot{\mathbf{a}}(t); \mathbf{a}(t)]$ costituito da posizione e velocità dei segmenti.

Comportamento iniziale (I)

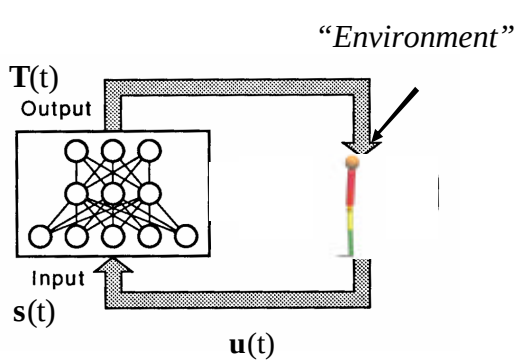
"Environment"

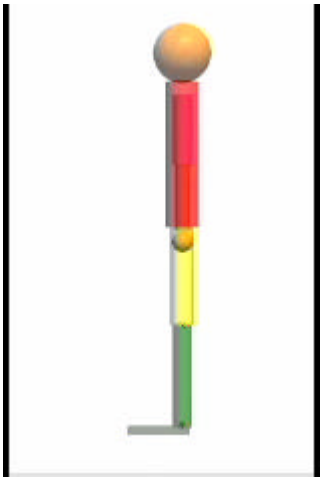




$\ddot{\mathbf{a}}(t) \rightarrow \mathbf{s}(t) = [\dot{\mathbf{a}}(t); \mathbf{a}(t)]$

Comportamento iniziale (II)





$\ddot{a}(t) \rightarrow s(t) = [\dot{a}(t); a(t)]$

Credit Assignement

Temporal credit assignement. In che istante la rete ha sbagliato?

Structural credit assignement. Quale unità della rete ha sbagliato?

Riassunto

•Reinforcement learning. I pesi vengono modificati, rinforzando le soluzioni buone.

•Self-discovery of 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.

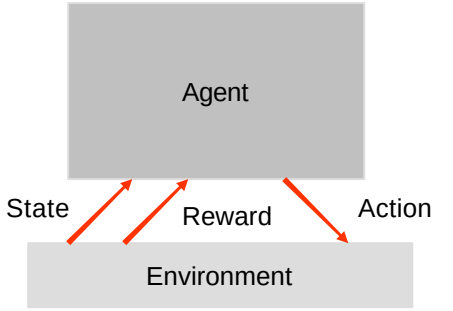
•Credit assignement.

•Come possiamo procedere in modo efficiente nello scoprire una strategia di successo? Esplorazione dello spazio dei pesi?

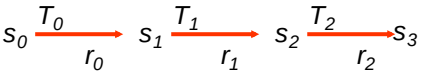
La Funzione Rinforzo

Viene ripetuto il ciclo:

- Eseguire delle azioni sul mondo {T}
- Osservare lo stato del mondo {s}.
- Osservare la ricompensa {r}.



Imparare una politica di controllo (a= f(s)) tale che viene massimizzata la ricompensa totale ("life reward":



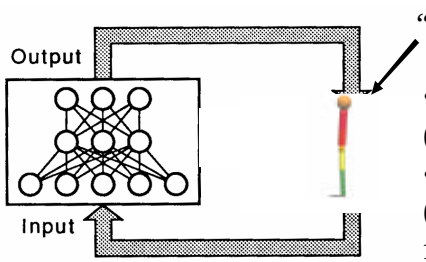
$r_0 + \gamma r_1 + \gamma^2 r_2 \ldots$

Per ogni stato, con $0 < \gamma < 1$

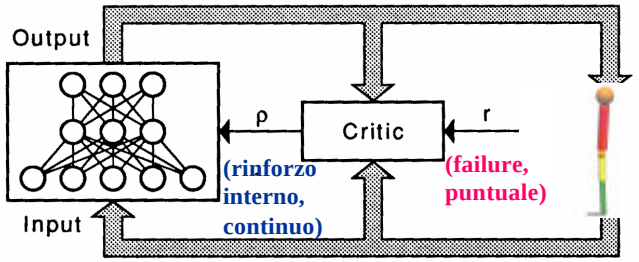
Da dove vengono gli $\{r_i\}$?

NB: Unsupervised learning. Delayed reward.

Reinforcement Learning



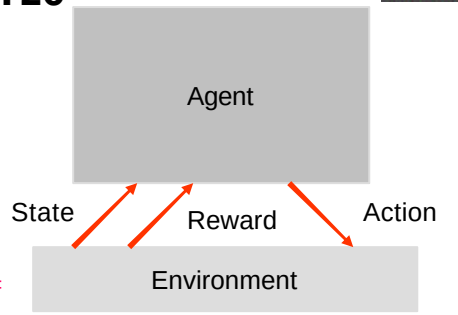
- r is the primary reinforcement (failure), scalare.
- r is the secondary reinforcement (derivato dal cost-to-go), scalare fornito con continuità nel tempo.



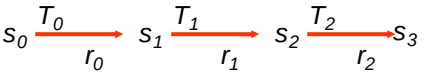
Lo schema dell'apprendimento con rinforzo

Viene ripetuto il ciclo:

- Eseguire delle azioni sul mondo {T}
- Osservare lo stato del mondo {s}.
- Osservare la ricompensa {r}.



Imparare una politica di controllo (T = f(s)) tale che viene massimizzata la ricompensa totale ("life reward")



Imparare una valutazione degli stati in funzione al loro "grado di rischio" o "grado di ricompensa" che promettono.

Come posso valutare la ricompensa a lungo termine?

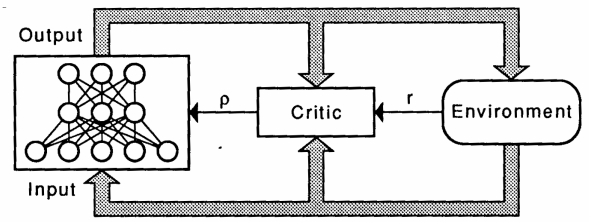
- Ho bisogno di una funzione che per ogni stato presente, in funzione della catena di ingressi (policy) che prevedo di scegliere in futuro, mi possa dire quanto mi costa, o quanto è vantaggiosa la policy di controllo utilizzata.
- E' una funzione che mi rappresenta la mappa di rischio.

Struttura della critica

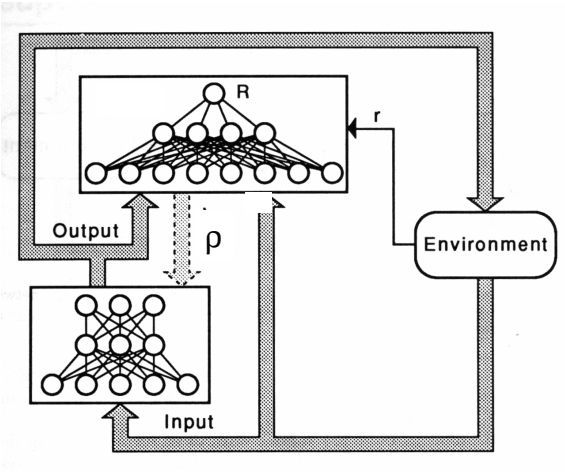
Per ogni istante t, la mappa di rischio, J(t) = J(**s**(t)), è una funzione dello **stato** definita a partire dalla sequenza di stati (e di Output).

J(.) viene rappresentato da una funzione non-lineare, derivabile.

La critica impara una mappa di rischio per ogni stato, ed invia al controllore un segnale di rinforzo interno: ρ(t).



Da dove nasce la mappa di rischio?



•Deve essere appreso anch'esso.

•Deve trasformare (attraverso la mappa di rischio) lo scalare r puntuale, in un secondo salare ρ, fornito con continuità nel tempo.

•**Seconda rete neurale specializzata nell'apprendimento della mappa di rischio.**

Un'implementazione di RL (ACE/ASE)

A. Barto, R. Sutton and C:W. Anderson, Neuron-like Adaptive Elements That Can Solve Difficult Learning Control Problems, IEEE Trans. Systems, Man and Cybernetics, 1983.

ASE – Adaptive Search Element – Controllore.

ACE – Adaptive Critic Element – Critica.

Rappresentazione a box delle variabili di stato

Le variabili sono codificate a **box**.

Orientamento del polpaccio rispetto ad un asse verticale $\vartheta : 0, \pm 4, \pm 12, \pm 24$ deg
Velocità angolare del polpaccio $\dot{\jmath} : \pm 50, \pm\infty$ deg/s

Orientamento della coscia rispetto ad un asse verticale $w : 0, \pm 4, \pm 12, \pm 24$ deg
Velocità angolare della coscia $\dot{w} : \pm 50, \pm\infty$ deg/s

Orientamento del tronco rispetto ad un asse verticale $j : 0, \pm 4, \pm 12, \pm 24$ deg
Velocità angolare del tronco $\dot{j} : \pm 50, \pm\infty$ deg/s

Altra possibilità: fuzzy set. CMAC.

Modellazione del controllore con RL

Suppongo s(t) = 0 se il sistema non si trova in quel particolare stato, oppure s(t) = 1 viceversa.

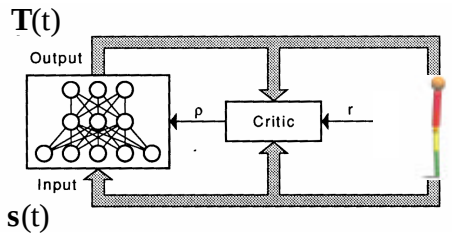
Il segnale di rinforzo esterno r = -1 nel momento della failure, altrimenti r = 0.

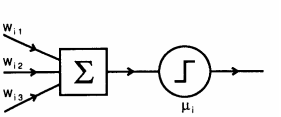
Considero che la critica mi fornisca uno scalare graduato che rappresenta il mio rinforzo interno o rischio.

Considero che il controllore fornisca uno scalare -1 o 1 per ciascuna delle variabili di controllo.

Struttura del controllore e della critica

$$T_i(t) = \Theta\left(\sum_i w_{ij}(t) s_i(t) + noise(t)\right)$$





Noise(t) – ha il ruolo di incoraggiare l'esplorazione dello spazio.x

$$p(t) = \left(\sum_i v_i(t) s_i(t)\right)$$ p(t) – mappa di rischio.

ρ(t) – rinforzo interno, scalare funzione di p(t), r(t), p(t-1).

Apprendimento nel controllore

L'eleggibilità

$$e_{ij}^{c}(t+1) = \delta e_{ij}^{c}(t) + (1-\delta)T_j(t)s_i(t) \qquad \delta < 1$$

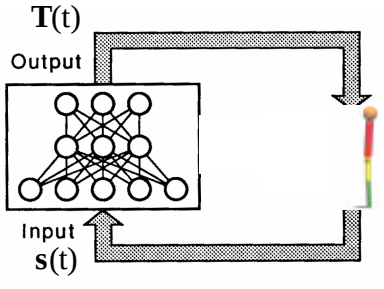
Se uno stato $s_i(t)$ non viene visitato ($s_i(t) = 0$), la sua eleggibilità decresce esponenzialmente.

Se uno stato $s_i(t)$ viene visitato di recente ($s_i(t) = 1$):
se $T_j(t)$ rimane dello stesso segno, la sua eleggibilità tende a $T_j * s_i$.
se $T_j(t)$ cambia spesso segno, la sua eleggibilità tende a 0.

La eleggibilità *aggiunge perciò la dimensione temporale al prodotto $T_j * s_i$: questo viene considerato valido solamente se si ripete nel tempo e se si ripete uguale (e.g. Torque positivo per valore dello stato negativo).*

Aggiornamento del controllore

$$T_j(t) = \Theta\left(\sum_i w_{ij}(t)s_i(t) + noise(t)\right)$$



$$\Delta w_{ij}^{c} = a\, r(t) e_{ij}(t)$$

$e_{ij}(t)$ – eleggibilità del peso ij.

Nel caso del perceptrone era:

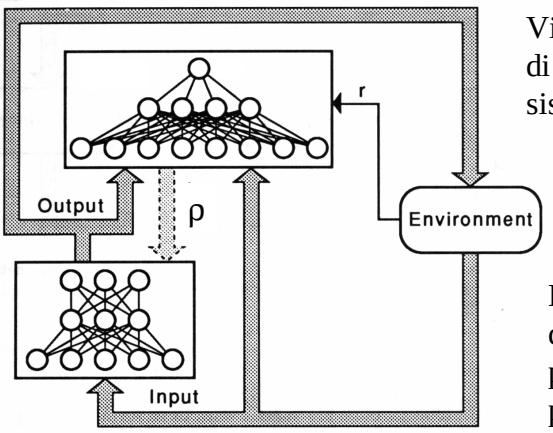
$$\Delta w_{ij} = h\Theta\left(1 - T_i^D T_i\right) T_i^D s_j$$

Il rinforzo, $\rho(t)$, decide l'intensità dell'aggiornamento dell'unità i al tempo t. NB Lo structural credit assignement è risolto dall'eleggibilità.

L'aggiornamento Hebbiano qui dipende dall'eleggibilità.

Apprendimento del rinforzo interno, r(t)

Due passi:



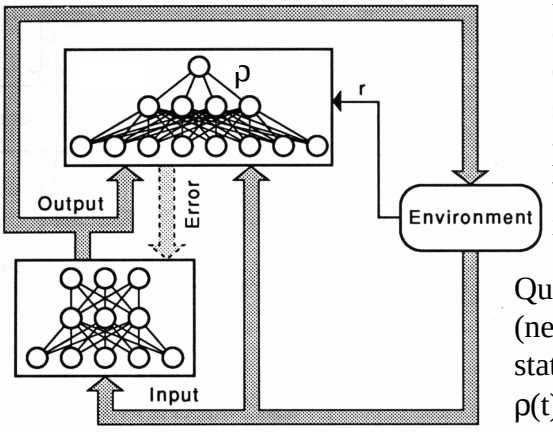
Viene calcolato per ogni istante di tempo, lo stato di rischio del sistema, p(t):

$$p(t) = \left(\sum_i v_i(t) s_i(t)\right)$$

Dallo stato di rischio attuale e dallo stato di rischio precedente (e dal rinforzo puntuale, r), determino il rinforzo interno, ρ(t).

Funzionamento del rinforzo interno

$$r(t) = r(t) + ?p(t) - p(t-1) \quad 0 < ? \leq 1$$



Fino a quando il controllore riesce a mantenere la postura eretta (nessun fallimento, r = 0), ρ(t) è **positivo**, quando il sistema passa da uno stato a più alto grado di rischio ad uno con un grado di rischio inferiore.

Quando arriva il reinforcement (negativo), r = -1. Non ci sono stati associati, per cui p(T) = 0. ρ(t) diventa **negativo**:

$$\rho(t) = -1 - p(t-1).$$

Apprendimento della mappa di rischio, p(t)

$$p(t) = \left(\sum_i v_i(t) s_i(t)\right) \qquad r(t) = r(t) + ?p(t) - p(t-1) \quad 0 < ? \leq 1$$

Eligibility di uno stato $s_i(t)$ dipende da quante volte lo stato è stato visitato nel passato. Uno stato sempre visitato avrà eligibility massima:

$$e_i^r(t+1) = l e_i^r(t) + (1-l) s_i(t)$$

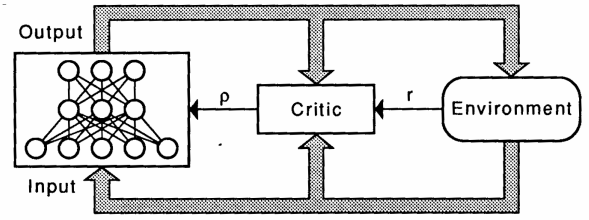
Aggiorno la mappa di rischio rinforzando quei pesi associati alle funzioni di rischio.

$$\Delta v_i = b\, r(t) e_i^r(t) \qquad \Delta w_{ij}^c = \alpha \rho(t) e_{ij}(t)$$

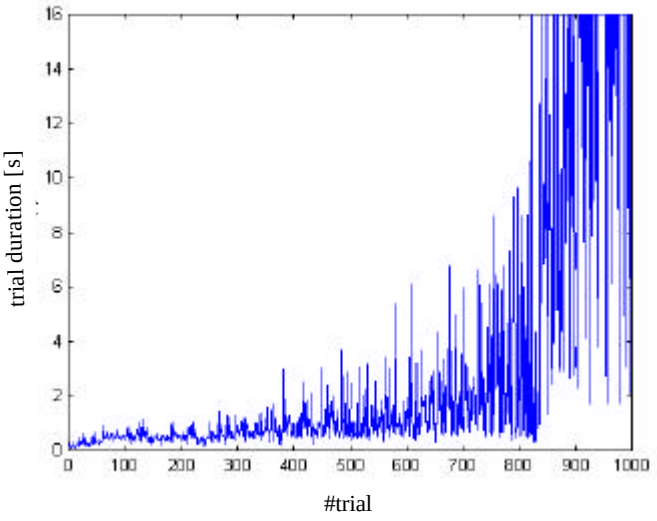
La critica

La critica deve valutare il funzionamento del controllore in un modo che sia: **appropriato** per l'obbiettivo del controllo e sufficientemente **informativo** perché il controllore apprenda.

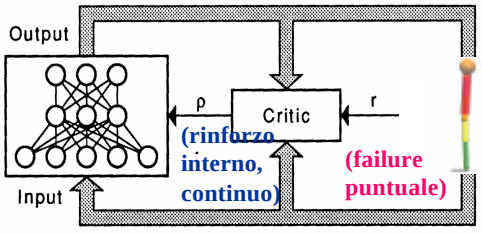
Determinare **come variare i pesi** del controllore in modo da migliorare le prestazioni, misurate dalla critica.

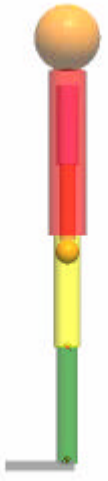


Curva di apprendimento



Apprendimento





La Stanza Cinese (J. Searle, 1980)

La persona (CPU).
Un libro di regole (Il programma).
Un pacco di fogli (la memoria).

Il calcolatore potrebbe dimostrare di essere intelligente al test di Turing, senza comprendere nulla.
Il signore nella stanza cinese riceve in ingresso dei simboli che manipola secondo regole a lui ignote e poi fornisce le risposte.

Lui non conosce il cinese!

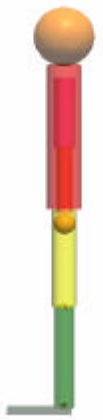
Riassunto sull'apprendimento con rinforzo

Necessita di una *critica*, che trasforma il segnale scalare di rinforzo (puntuale) in un segnale scalare temporale, $r(T) \rightarrow \rho(t)$.

La critica analizza le coppie input/output ed impara una mappa di rischio.

Utilizza questa mappa di rischio per fornire un segnale di rinforzo interno al controllore.

Il controllore aggiorna i pesi con un meccanismo Hebbiano, dove il prodotto ingresso/uscita viene valutato lungo la dimensione temporale.

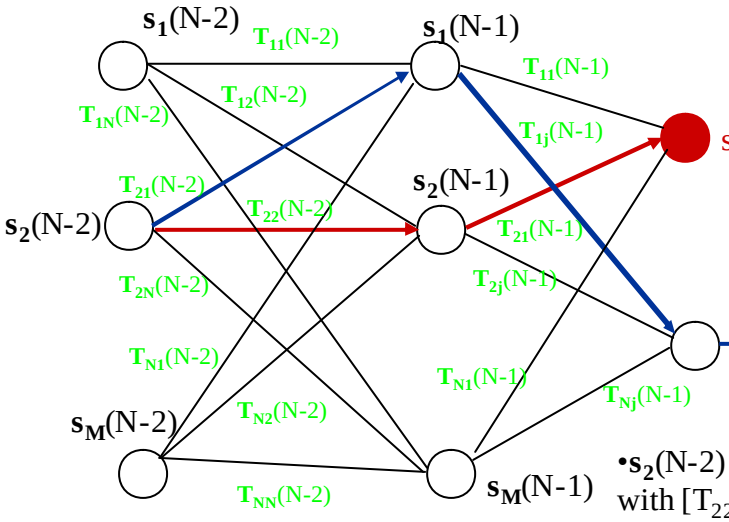


Traccia per ulteriori approfondimenti

Mappa di rischio e cost-to-go

- Ho bisogno di una funzione che per ogni stato presente, in funzione della catena di ingressi (policy) che prevedo di scegliere in futuro, mi possa dire quanto mi costa, o quanto è vantaggiosa la policy di controllo utilizzata.
- Questa funzione rappresenta la mappa di rischio, “cost-to-go”.

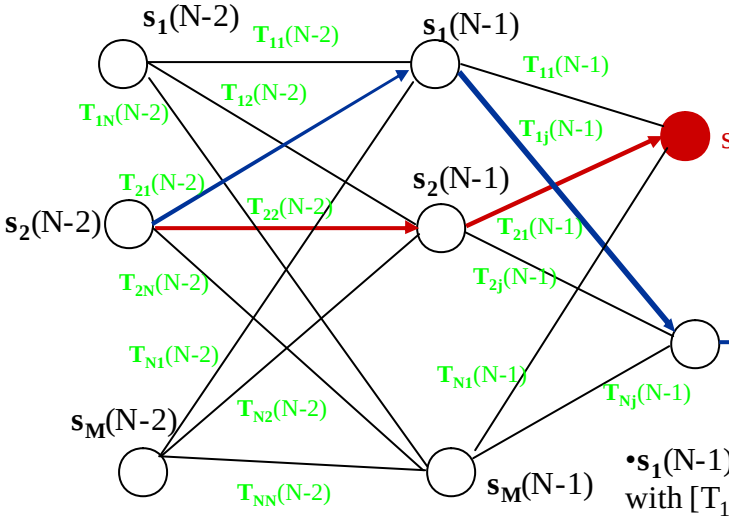
Il cost-to-go J(s(t))



Cost-to-go: $\sum_i T(N-i)$ + Cost of failure.

- s_2(N-2) high cost-to-go with [T_{22}(N-2); T_{21}(N-1)]
- s_2(N-2) low cost-to-go with [T_{21}(N-2); T_{1j}(N-1)]

Come si determina il cost-to-go?

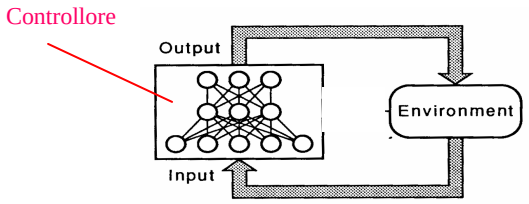


Partendo dallo stato finale, backwards.

- s_1(N-1) high cost-to-go with [T_{11}(N-1)]
- s_1(N-1) low cost-to-go with [T_{ij}(N-1)]

Osservazioni sul cost-to-go

- E se il task ha successo indefinitamente (problemi con orizzonte infinito)? Si può considerare un cost-to-go su una finestra temporale.
- Il cost-to-go è determinato perchè a partire da un certo stato, se non sopraggiungono eventi esterni (ambiente costante, controllore costante), l'evolversi della situazione è determinata.
- Anche se con una particolare sequenza di ingressi, il mio costo sarebbe minore, quella sequenza potrebbe non essere scelta dal controllore con la sua attuale configurazione dei pesi.



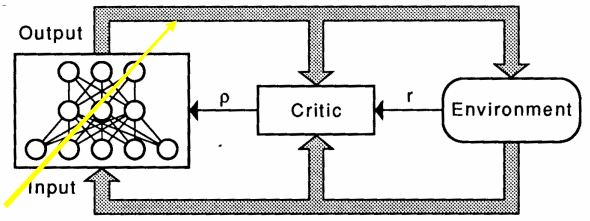
•$\mathbf{s_1}$(N-1) high cost-to-go with [T_{11}(N-1)]

•$\mathbf{s_1}$(N-1) low cost-to-go with [T_{ij}(N-1)]

Come si utilizza la critica

•Utilizziamo il cost-to-go in modo da forzare il controllore dallo stare alla larga dagli stati rischiosi.
•E' possibile quindi calcolare il gradiente $dJ/d\mathbf{s}(.)\Big|_t$ e determinare il nuovo stato: **s'**(t) = **s**(t) + d**s**(t) che migliora J(t): J(t)' = J(t) + dJ(t). (*J(.) è una funzione dello stato!*).
•Da d**s**(t) dobbiamo poi calcolare un d**T**(t) (inversione dell'environment)

•Possiamo quindi modificare i pesi del nostro controllore in modo tale che all'istante t, in modo che possiamo effetivamente ottenre **s'**(t).



Cost-to-go e ACE/ASE

$$r(t) = r(t) + \gamma p(t) - p(t-1) \quad 0 < \gamma \leq 1$$

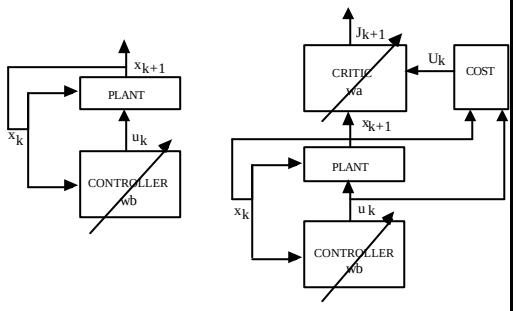
p(t), p(t-1) sono equivalenti ai cost-to-go.

L'apprendimento nell'ACE / ASE è Hebbiano. Esistono modelli più complessi di utilizzare il cost-to-go, p(t) per aggiornare i pesi del controllore.

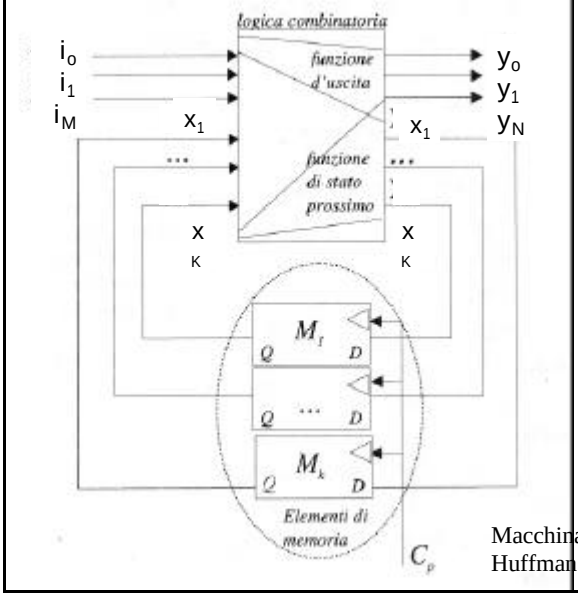
Approccio alternativo

Invece di considerare gli stati discretizzati $s_i(t) = 1$ se e solo se la variabile di ingresso corrispondente sta nell'intervallo i-esimo, si considerano variabili continue.

Viene definita una modalità per convertire la mappa di rischio in una variazione dei pesi del controllore, attraverso il calcolo esplicito del gradiente.



RL applicato agli automi a stati finiti (condizionamento operante)



Macchina di Huffman

Esempi di task per un agente:

Generazione di traiettorie, la correttezza può essere stabilita solamente alla fine del movimento.

Automi a Stati Finiti. Auto-apprendimento della funzione di transizione e di uscita.