Risposte esami vecchi

Algoritmi genetici

Un **algoritmo genetico** è un algoritmo ispirato al principio della selezione naturale ed evoluzione biologica.

Può essere utilizzato per risolvere problemi di ottimizzazione abbastanza generali (non sono richieste proprietà particolarmente stringenti).

L'aggettivo *genetico* è dovuto al fatto che gli algoritmi genetici attuano dei meccanismi concettualmente simili a quelli dei processi biochimici scoperti nella genetica.

In sintesi, un algoritmo genetico consente di valutare diverse soluzioni di partenza, come se fossero diversi individui di una popolazione, e ricombinandole (analogamente alla riproduzione biologica sessuata) ed introducendo mutazioni casuali, produrre nuove soluzioni (nuovi individui).

Il procedimento si ripete in questo modo fino ad aver raggiunto una soluzione soddisfacente o un numero massimo di iterazioni, nel tentativo di convergere verso soluzioni ottime.

Uno degli svantaggi degli algoritmi genetici è che data la natura casuale dell'algoritmo, non è possibile sapere a priori se convergerà ad una soluzione ottima o se avverrà una convergenza prematura a una soluzione non-ottima.

Se si otterrà un risultato soddisfacente, non è detto che si capisca perchè abbia funzionato.

A livello pratico per implementare un algoritmo genetico dobbiamo introdurre alcune definizioni:

- Cromosoma: è una delle soluzioni del problema considerato, tipicamente è rappresentato da una stringa di bit.
- Popolazione: è un insieme di soluzioni (cromosomi) del problema considerato.
- Gene: una parte del cromosoma. Tipicamente, alcuni "bit" del cromosoma compongono un gene.
- Fitness: Valutazione associata ad una soluzione (cromosoma), ottenuta attraverso una funzione di fitness apposita o attraverso una simulazione.
- Crossover: generazione di una nuova soluzione mescolando le soluzioni esistenti,
- Mutazione: Alterazione (piccola) casuale di una soluzione

Il crossover permette di esplorare aree anche distanti dello spazio di ricerca.

La mutazione permette di esplorare aree vicine alla soluzione originale nello spazio di ricerca.

È possibile migliorare l'algoritmo introducendo ulteriori meccanismi, alcuni basati, ad esempio, sulla scelta dei genitori:

Uno di questi è l'elitismo, che copia gli individui migliori della generazione precedente senza applicarvi mutazioni o crossover, permettendo di evitare la perdita di individui con alto fitness a causa della casualità.

Un'altra tecnica è rappresentata dallo scaling, per cui la probabilità di essere selezionato non è proporzionale alla fitness, ma ad una funzione correlata.

Un sistema dinamico che evolve verso un attrattore elabora informazione. Si discuta facendo riferimento anche ad alcuni casi studiati.

(La risposta che segue ignora completamente il fatto che nella domanda c'è scritto "elabora informazione". è troppo generico imo è non so cosa voglia che io scriva quindi facciamo che va bene così)

Un **sistema dinamico** è un sistema che evolve nel tempo seguendo delle regole.

È composto da uno spazio delle fasi (o degli stati) in cui avviene l'evoluzione del sistema, e dalla legge che descrive questa evoluzione. Un esempio di sistema dinamico è il sistema solare, dove i corpi (pianeti) orbitano attorno al sole, ma sono soggetti anche gli uni alle forze gravitazionali degli altri.

I sistemi dinamici lineari sono sistemi la cui evoluzione è governata da equazioni lineari. Nella maggior parte dei casi, i sistemi dinamici lineari non sono sufficientemente sofisticati per descrivere dei fenomeni reali (sul lungo periodo).

Nei sistemi dinamici è possibile individuare dei punti fissi. Ad esempio, in un sistema dinamico che modella due popolazioni in un ambiente, è possibile trovare un punto fisso nello stato (0, 0) inteso come 0 individui vivi della popolazione numero 1 e 0 individui vivi della popolazione numero 2 (entrambe le popolazioni sono estinte).

Un punto fisso è un punto in cui il sistema è in "equilibrio", se non perturbato non si muove da quello stato.

Esistono punti fissi stabili e instabili. Un punto fisso è stabile se, data una piccola perturbazione allo stato, il sistema dinamico torna sul punto fisso (esempio: pallina in una valle).

Un punto fisso è instabile se data una piccola perturbazione allo stato, il sistema dinamico non torna più in quel punto fisso (esempio:pallina su un cucuzzolo).

I sistemi dinamici possono contenere degli attrattori, cioè dei sottoinsiemi dello spazio delle fasi che attirano molte configurazioni iniziali diverse. L'insieme delle configurazioni iniziali che vanno a finire eventualmente in un certo attrattore è chiamato bacino di attrazione.

Esistono vari tipi di attrattori, alcuni presenti solo nei sistemi non lineari: punti, periodi (ripetizioni periodiche degli stessi valori),centri (orbite nello spazio delle fasi), cicli limite (esempio del pendolo con molla), ma anche attrattori "strani", con struttura geometrica complessa, spesso frattale (esempio: attrattore di Lorenz)..

Presentare il modello delle reti neurali a strati (percettroni e MLP) e descriverne le proprietà.

Si tratta di un modello di rete neurale artificiale che mappa insiemi di dati in ingresso con insieme di dati in uscita.

Operativamente dobbiamo immaginare strati multipli di neuroni connessi in maniera analoga ad un grafo diretto completo, in cui ogni strato è connesso completamente con il successivo.

Per permetterne l'apprendimento, si utilizza la retropropagazione del gradiente (back-propagation).

Significa che: dando dei dati in ingresso, vogliamo ottenere un determinato risultato in uscita per calcolare quindi il gradiente della funzione di perdita (o di costo). Durante il training avviene un costante processo di aggiustamento dei pesi dei collegamenti tra i nostri neuroni (weights) al fine di ottenere il risultato auspicato.

Poiché durante la retropropagazione i gradienti ai vari livelli vengono moltiplicati tramite la regola della catena, il prodotto di n numeri, in un insieme come (0,1), può decrescere esponenzialmente rispetto alla profondità n della rete. Paradossalmente, un valore troppo elevato può portare all'esplosione del gradiente.

Ad oggi il problema della perdita di gradiente è stato in parte risolto dalla potenza dall'elevata potenza di calcolo degli attuali strumenti computazionali e grazie alla transazione da CPU a GPGPU. Inoltre si è scoperto che un altro metodo utile per contrastare questo tipo di fenomeno è quello di allenare ogni strato in maniera indipendente e poi assemblarli tutti insieme.

Si presentino gli agenti e i sistemi multiagente, e come netlogo ne utilizzi i paradigmi.

Un agente è definito come un'unità di elaborazione che può essere (o meno) in grado di percepire e interagire con il mondo che la circonda, e possiede una certa autonomia nello svolgere delle azioni, le quali sono almeno in parte legate all'esperienza personale dell'agente. L'agente possiede un set di regole e delle variabili interne (lo stato dell'agente) che possono sia essere costanti sia variare durante la sua vita.

Un'agente può quindi muoversi e interagire con l'ambiente, interagire con altri agenti, possedere delle tendenze, avere un obiettivo ed elaborare strategie per portarlo a termine, offrire servizi e riprodursi.

Quando si parla di agente, lo si può fare in due modi: il paradigma debole illustra la sua autonomia decisionale, la sua reattività (reagisce a stimoli esterni), la sua proattività (prende l'iniziativa per portare a termine il suo compito) e le sue abilità sociali (comunicazione con altri agenti), mentre il paradigma forte attribuisce all'agente caratteristiche umane. Di un agente è inoltre possibile valutare caratteristiche come la benevolenza (l'agente compie azioni corrette dal punto di vista "etico") e la razionalità (le azioni dell'agente sono concordi con il suo obiettivo).

Esistono numerosi modi per classificare gli agenti: per come prendono decisioni, per la natura dell'obiettivo e infine per come percepiscono il mondo che li circonda.

In base al loro processo decisionale, gli agenti vengono divisi in:

- Logici: agenti il cui processo decisionale è affidato ad una serie di elaborazioni di tipo logico
- Reattivi: l'azione successiva è generalmente stabilita da un evento che accade (ad esempio, agente reattivo è il roomba, perché ha una logica del tipo: ho sbattuto contro a un muro, come reagisco? Mi giro un po' e continuo a pulire)
- BDI: gli agenti BDI prendono decisioni sulla base di strutture dati che sono in essi contenuti, che rappresentano il Belief, il Desire e l'Intention
- Layered: gli agenti layered sono costruiti a strati, con ciascuno strato che percepisce in modo diverso il mondo e che reagisce quindi in modi differenti

In base alla natura dell'obiettivo, l'agente si dice riflessivo se questo riflette lo stato interno dell'agente (ad esempio, se l'agente deve spostare l'oggetto A, ma ha fame, l'obiettivo diventa il trovare cibo) mentre si dice teleonomico se ha un obiettivo esplicito.

In base al modo in cui l'agente percepisce e rappresenta il mondo, si dice che l'agente è cognitivo se ha una rappresentazione esplicita del mondo (e questo gli consente di elaborare strategie), mentre si dice reattivo se ha una rappresentazione subsimbolica del mondo (ad esempio mediante regole).

Inserendo nello stesso ambiente più agenti, si crea un sistema multi-agente, all'interno del quale gli agenti possono interagire tra di loro, più o meno direttamente. Le ragioni per le quali vengono usati sistemi del genere sono molteplici: può, ad esempio, essere un modo per incrementare le performance di un software attraverso parallelismo (ad esempio, se un agente da solo legge 10 righe al secondo, mettendone di più in parallelo il numero di linee lette al secondo aumenta), dal momento che un agente è limitato (può nel suo arco vitale fare solo un numero limitato di operazioni), oppure può essere l'unico modo per approcciare sistemi che sono intrinsecamente distribuiti.

In generale, per maneggiare sistemi complessi, si rivela molto più facile pensare in termini di collettività piuttosto che singoli individui, in quanto allo stesso modo è impostata la natura umana (un singolo umano, come un agente, è limitato).

Gli impieghi di sistemi multi-agente sono variegati, anche se l'applicazione più interessante è in ambito simulativo: è possibile, creando agenti sufficientemente vicini al corrispettivo reale, simulare interi sistemi naturali, e valutarne l'evoluzione e studiarne le caratteristiche significative. Questa applicazione rappresenta una valida alternativa alle simulazioni di tipo analitico.

Nel nostro percorso di studio, per interagire con gli agenti, abbiamo impiegato il software NetLogo. NetLogo fornisce all'utente un valido supporto grafico che permette di collocare, mediante un dialetto del CommonLisp, in un mondo virtuale agenti, e di gestirne il comportamento. NetLogo gestisce 4 tipologie di agenti, ciascuna con un uso specifico:

- Turtle: i turtle sono gli agenti che si muovono all'interno del mondo, e che possono ad esempio rappresentare delle formiche, o delle termiti.
- Patches: il tassellamento del mondo (che di default ha topologia toroidale, ma può anche essere configurato diversamente) di netlogo è composto da questi agenti, che costituiscono l'ambiente percepito dai turtle. Ogni patch è identificata da una coppia di coordinate.
- Link: un link è un collegamento tra più agenti.
- Observer: l'utente, l'osservatore del mondo.

NetLogo permette di creare dei "breed" personalizzati degli agenti Turtle e Patches, ed è possibile fornir loro delle variabili private. Con netlogo, usando poco codice, è possibile ricreare complessi sistemi naturali simulati, in particolare provvisti di swarm intelligence.

Si presentino i sistemi multiagente, in particolare presentando i meccanismi peculiari della swarm intelligence.

Un sistema multi agente è composto da un insieme di agenti che interagiscono tra di loro e con l'ambiente in cui si trovano.

Un agente è un'entità caratterizzata dal fatto di essere almeno parzialmente autonoma nel cercare di raggiungere il suo scopo.

Nei sistemi multiagente, è possibile osservare l'emergere della "swarm intelligence", traducibile come "intelligenza di sciame".

Le caratteristiche della swarm intelligence sono generalmente che:

- ogni individuo che compone lo sciame di agenti è tipicamente poco sofisticato e ha capacità limitate
- ogni individuo del sistema non conosce lo stato globale del sistema;
- non è presente un coordinatore / leader.

Gli agenti possono comunicare tra di loro in modo diretto oppure attraverso la stigmergia.

La stigmergia è una forma di comunicazione indiretta che avviene alterando lo stato dell'ambiente in modo tale da influenzare il comportamento degli altri individui, poichè l'ambiente stesso è uno stimolo per gli altri individui.

Nel nostro percorso abbiamo esaminato vari esempi di sistemi multi agente.

Il sistema delle termiti, dove è presente un ambiente con una certa densità di cibo, e una moltitudine di agenti "termiti" che devono accumulare il cibo in gruppetti (invece che mantenerlo sparso come nella situazione iniziale).

In questo sistema non emerge una swarm intelligence, il compito di raggruppare il cibo è svolto con successo anche se introduciamo una sola termite solitaria nel sistema.

Una altra osservazione che ci suggerisce che non c'è una swarm intelligence è che se raddoppio il numero di termiti (esempio da 50 a 100) il tempo impiegato per raggiungere un certo stato in media dimezza.

Con una vera swarm intelligence di solito il beneficio di raddoppiare gli agenti dovrebbe essere maggiore del raddoppiamento della performance.

Un sistema in cui invece abbiamo osservato l'emergere della swarm intelligence è quello delle formiche che devono portare il cibo da dei cumuli al proprio nido, locato al centro dell'ambiente.

L'ambiente in questo modello ha un ruolo attivo fondamentale: deve diffondere e fare evaporare le tracce che gli agenti (le formiche) possono lasciarvi sopra.

In questo sistema, se mettiamo una sola formica osserviamo che non è in grado di portare a termine il compito in maniera efficiente.

Aggiungendo un numero significativo di formiche il compito viene portato a termine molto più velocemente.

Altri esempi di swarm intelligence che abbiamo visto sono: gli stormi di storni, i banchi di sardine, le lucciole.

Si illustrino le proprietà dei sistemi a classificatori.

L'interesse per la ricerca nel campo delle intelligenze artificiali è stato vivo sin dalla seconda metà del 1900, particolarmente foraggiato anche dalla fantascienza, che aveva inserito nell'immaginario collettivo dei più un'immagine di intelligenza artificiale universale, in grado di trattare qualsiasi tipologia di problema. A ravvivare ulteriormente le speranze, una piccola rivoluzione: i sistemi esperti, molto più flessibili dei normali programmi e dal potenziale molto, molto grande.

Un sistema esperto è composto, principalmente, da due componenti base: la base della conoscenza, che fornisce al sistema una rappresentazione simbolica del mondo basata su regole (inferenze), e il motore inferenziale, che si districa tra le inferenze per fornire una risposta. I primi prodotti di questo tipo, come ad esempio Mycin (sistema esperto per la diagnosi di malattie), riscossero un notevole successo, ravvivando la speranza di ottenere un'IA generalista in modo abbastanza facile. Questo ottimismo era, però, mal piazzato: Mycin, come gli altri sistemi esperti funzionanti, non era un'IA generalista: bensì era specialistica. Più si procedeva nella direzione dei sistemi a classificatori, più sorgevano problematiche: i sistemi esperti non sono robusti, non gestiscono bene contraddizioni e aggiornare la base della conoscenza è assai ostico. Se a questo si aggiungono i diversi addetti alle vendite che millantavano abilità straordinarie dei loro sistemi esperti, è facile capire come mai (nonostante i successi in ambito specializzato) a questo periodo sia seguito "l'inverno delle IA". Sistemi come Mycin, però, avevano mostrato come i S.E. fossero funzionanti, anche se non era possibile prendere scorciatoie.

Una sorta di evoluzione dei sistemi esperti è quella dei sistemi a classificatori: un sistema a classificatori può essere definito come un sistema esperto in grado di auto-apprendere.

Al S.a.C vengono fornite solamente regole base, e viene immerso in un mondo di cui sa poco o nulla: procedendo a tentativi, il sistema esperto riesce a far variare le proprie regole, e in tal modo riesce a diventare sempre più competente in quell'ambito. I sistemi a classificatori posseggono due dinamiche:

- La dinamica veloce, basata sull'algoritmo "bucket brigade", consente di retribuire i classificatori che si comportano correttamente
- La dinamica lenta, costituita da un GA con elitismo, permette di creare variazioni delle leggi con maggiore fitness

Un sistema a classificatori segue la seguente routine di funzionamento:

- Lettura dei messaggi di ingresso, provenienti da sensori o simili
- I messaggi vengono riconosciuti da alcuni classificatori: questi competono per impostare messaggi (e quindi possono anche causare l'attivazione di altri classificatori)
- I classificatori impostano i messaggi in uscita, coloro che sono stati attivati da terzi retribuiscono chi li ha attivati
- La lista messaggi viene aggiornata
- I messaggi in output vengono eseguiti
- I classificatori che hanno impostato il messaggio vengono retribuiti dall'ambiente (in modo positivo o negativo)

I messaggi sono delle stringhe binarie composte da 0 e 1, i classificatori sono composti da stringhe binarie 0, 1, # e le azioni ad essi legati sono stringhe 0, 1, * (pass through). Un valore significativo è la

specificità del classificatore, ovvero (n° valori diversi da # nel messaggio)/(L), dove L è la lunghezza di messaggi e classificatori.

Il bid, ovvero il valore che indica l'*importanza* del classificatore, è calcolato moltiplicando la forza (fitness) del classificatore per la specificità. Più è alto il bid, maggiore probabilità ha quel classificatore di impostare un messaggio di uscita. Come già accennato, il bucket brigade aumenta la fitness dei classificatori che attivano altri classificatori, e l'algoritmo genetico introduce delle variazioni nelle nuove generazioni di regole, cancellando ogni volta quelle più deboli. I sistemi a classificatori possono contenere meccanismi come il cover detector, che rileva quando un messaggio non viene accettato da alcun classificatore (e ne va a creare uno) e il cover effector, che crea un classificatore che invia un messaggio in uscita casuale valido (se necessario).

Le particolarità suggestive dei sistemi a classificatori sono che è presente la co-evoluzione (la variazione di un classificatore influenza la fitness di altri), ed è in grado di elaborare in parallelo. Particolarmente interessante è l'unione di una metafora di tipo economico a un concetto biologico, ma questo non basta per sopperire alle problematiche: è possibile la convergenza prematura, tutti gli individui hanno la stessa lunghezza, è impossibile prevedere lo stato finale e possiede una struttura estremamente piatta.

Si illustrino le proprietà del modello di Hopfield.

Una rete di Hopfield è un tipo di rete neurale artificiale, nota per essere il modello di rete che simula le capacità del cervello umano di ricordare le cose o di ricostruire le immagini distorte