SAPIENZA
UNIVERSITÀ DI ROMA

FACOLTÀ DI INGEGNERIA DELL'INFORMAZIONE,
INFORMATICA E STATISTICA

CORSO DI LAUREA IN INFORMATICA

Relazione di Tirocinio

Analisi, sviluppo e test di un sistema di visione attiva artificiale

Laureando

Fabio Ticconi

Matricola 1165383

Responsabile interno

Prof. Andrea Sterbini

Responsabile esterno

Dott. Stefano Nolfi

Anno Accademico 2010/2011

Ad Anto

Ringraziamenti

Il mio percorso di laurea e il tirocinio al CNR hanno impiegato non solo molte delle mie energie, ma anche il tempo e i pensieri di tanti altri. Soprattutto quelli che mi hanno dovuto sopportare.

Un ringraziamento speciale va ad Anto, a cui questa tesi è dedicata: senza di te non sarei mai arrivato fin qui. È più importante sapersi vicini di mille parole e discorsi.

Un grazie di cuore va alla mia famiglia, nella quale mi sono sempre sentito appoggiato e supportato, emotivamente e finanziariamente, qualunque cosa decidessi di fare. Poter scegliere il proprio percorso (non solo universitario), senza vincoli difficilmente aggirabili e senza pressioni psicologiche di alcun tipo, e sapendo di essere apprezzati qualunque sia il risultato finale: questi sono valori di cui solo recentemente ho compreso l'importanza. Non tutti sono così fortunati.

Durante l'università si conoscono moltissimi studenti con i quali si condividono le gioie e i dolori dei periodi d'esame (e non solo): anche selezionando sarebbe difficile ringraziarli tutti. Così, chi mi legge si senta già ringraziato, e di cuore. Ma, senza nulla togliere agli altri, una menzione speciale va a Tommaso e ad Antonio. Difficilmente ho conosciuto ragazzi della cui sincerità e disponibilità fossi stato sicuro senza mai un dubbio, e questo pur vedendoci raramente. Spero di essere stato un amico altrettanto

valido per loro.

Anche al CNR, in questi mesi per me splendidi nonostante i contemporanei impegni universitari, ho conosciuto persone che mi hanno fatto sentire subito a mio agio, impedendo sul nascere qualsiasi timore e imbarazzo. Senza contare le mille conversazioni, che spesso mi hanno aiutato a rendere le mie idee più chiare e le mie aspirazioni più precise, e i continui consigli e suggerimenti sul progetto che portavo avanti. Ringrazio tutti voi, ma soprattutto i "più vicini", spazialmente ma non solo: Diana, Gianluca, Giuseppe, Onofrio, Tomassino. Menzione d'onore a Tomas che mi ha sopportato come vicino di scrivania.

Last but not least, e mai espressione fu più vera, un enorme grazie ad Andrea e Stefano, coloro i quali mi hanno seguito, guidato e consigliato con estrema competenza in questo progetto di tirocinio, correggendo la tesi a tempi di record e trattandomi con più gentilezza e rispetto di quanto avrei potuto immaginare.

Introduzione

L'occhio che si dice finestra dell'anima è la principale via, donde il comune senso può più copiosa e magnificamente considerare le infinite opere di natura, e l'orecchio è il secondo, il quale si fa nobile per le cose racconte, le quali ha veduto l'occhio.

Leonardo da Vinci

La visione è da sempre considerata un'attività fondamentale per la nostra vita, e il senso della vista, tra i cinque sensi canonici, è quello a cui gli esseri umani tendono a essere più legati. Non stupisce che la tecnologia di tutti i tempi abbia cercato di porre rimedio ai disturbi della visione (dalle lenti correttive alle moderne retine artificiali), come anche di "imitare" artificialmente l'esperienza visiva per poterla archiviare e riutilizzare al bisogno (dalle fotografie ai moderni sistemi di videosorveglianza).

Nel corso del XX secolo, e in particolare dagli anni '70, in seno all'informatica e all'ingegneria si è affermata la disciplina della **Visione Artificiale**, nella quale sono stati sviluppati un'enorme quantità di strumenti, teorie e tecniche per la cattura, il processamento e l'estrazione di informazioni da immagini statiche e video, per i più svariati scopi, tra i quali:

- Sistemi robotici autonomi
- Video sorveglianza
- Interazione uomo-macchina
- Ricerca in database di immagini

Una caratteristica che hanno in comune la maggior parte deisistemi di visione artificiale è l' **approccio passivo all'analisi delle immagini**. Dopo aver catturato un'immagine (o un fotogramma da un video), vengono applicate delle trasformazioni e manipolazioni che permettono di estrarne delle informazioni specifiche, che a loro volta vengono poi usate per costruire un modello esplicito del mondo che si vuole comprendere o con cui si vuole interagire.

Se questo è un approccio che si è dimostrato molto efficiente per l'analisi in *delayed time* di immagini statiche, lo stesso non si può dire per tutte quelle applicazioni che richiedono il processamento in *real time* di quantità enormi di fotogrammi, dal quale dipendono le successive azioni del sistema. Il caso tipico è quello della robotica autonoma, dove l'*overhead* introdotto dall'analisi passiva di ogni fotogramma risulta inaccettabile per la maggior parte degli scenari possibili: esplorazione di ambienti ostili o critici (dai territori di guerra allo spazio), imprevedibilità dell'ambiente, necessità di ridurre le possibilità di danneggiamento dell'agente (dato il costo dei sistemi robotici sofisticati).

A partire da queste considerazioni e da ricerche degli ultimi decenni sulla visione umana e animale, si è andata sviluppando una sotto-disciplina della visione artificiale chiamata **Visione Attiva**, la quale postula la necessità per un agente di esplorare attivamente l'ambiente (**decidendo** in modo autonomo dove orientare i propri sensori) per poter **comprendere** in maniera efficace ed efficiente una particolare scena visiva.

Obiettivo di questo Tirocinio Formativo è stato quello di utilizzare i metodi della **Robotica Evolutiva** (Nolfi e Floreano, 2000) per sviluppare e testare un sistema di

visione attiva artificiale, estendendo il software **Evorobot***[1].

L'intento è stato quello di costringere il sistema a sfruttare il movimento, e quindi a scegliere autonomamente quale porzione dell'immagine visualizzare, per risolvere compiti di categorizzazione che altrimenti, date le risorse volutamente limitate, sarebbe stato troppo difficile completare con successo.

Il tempo inoltre assume un ruolo importante: per il particolare tipo di sistema di controllo usato (che vedremo nei capitolo successivi), la porzione di immagine esperita al tempo t collabora al processo di categorizzazione insieme alla porzione di immagine esperita al tempo $t-1$. Questo permette al sistema di orientarsi gradualmente verso zone dell'immagine che migliorino le sue possibilità di categorizzare. Si può consultare Mirolli *et al.* (2010) per approfondimenti e una review sull'argomento.

Questa relazione riporta le basi teoriche, le scelte progettuali e i risultati degli esperimenti effettuati durante il tirocinio, secondo la seguente struttura:

Capitolo 1 - La Visione Artificiale

In questo capitolo, dopo una breve introduzione sull'importanza della percezione negli organismi viventi e nei sistemi artificiali, viene presentata la disciplina della Visione Artificiale, mettendo a confronto l'approccio classico con quello attivo. È anche presente un accenno al dibattito se sia necessaria una rappresentazione esplicita del mondo per capire e riprodurre la visione.

Capitolo 2 - Metodi e Tecniche della Robotica Evolutiva

Viene presentato l'approccio *embodied* e *situated* alla cognizione e in particolare all'intelligenza artificiale. Vengono brevemente descritti gli algoritmi genetici e le reti neurali artificiali. Vengono riportati alcuni esperimenti di robotica evolutiva sulla visione attiva.

[1] Il software è *open source* e liberamente scaricabile dal sito: http://laral.istc.cnr.it/evorobotstar

Capitolo 3 - Implementazione su Evorobot*

Viene introdotta la piattaforma utilizzata per le simulazioni, Evorobot*, e le modifiche effettuate per supportare la retina artificiale e il processo di categorizzazione.

Capitolo 4 - Risultati Sperimentali

Questo capitolo mostra i risultati ottenuti in alcuni setup sperimentali portati avanti durante il Tirocinio, utilizzando immagini di diverso tipo e dimensione.

Indice

Elenco delle figure

Capitolo 1

Visione Artificiale

Perceptual activity is exploratory, probing, searching; percepts do not simply fall onto sensors as rain falls onto ground. We do not just see, we look. And in the course [...] sometimes we even put on spectacles.

Ružena Bajcsy

1.1 L'importanza di percepire

I sensi rappresentano la nostra finestra sul mondo, che è enormemente più grande e dettagliato di quanto ci appaia. I fotorecettori nei nostri occhi reagiscono solo a una piccola porzione dell'intero spettro elettromagnetico, e lo stesso accade al nostro apparato uditivo con le onde sonore, così come siamo in grado di percepire con l'olfatto e con le papille gustative solo un numero molto limitato delle molecole che sono nell'aria e in ciò che mangiamo. La realtà che ci circonda è estremamente più ricca di quanto l'apparato sensoriale di cui l'evoluzione ci ha dotato, che agisce **da vero e proprio filtro**, ci faccia credere.

Il sistema sensoriale non è, tuttavia, solo questo. Difetti nelle parti anatomiche che permettono la visione o l'udito possono compromettere il corretto sviluppo neurale e cognitivo di un bambino, con effetti irreversibili senza un intervento tempestivo, i

cui tempi sono dettati da precise fasi dello crescita pre- e post-natale, accuratamente studiate negli animali e nell'uomo. Ad esempio, viene riscontrata un'alta incidenza di problemi psichiatrici, quali autismo e psicosi, in persone con sordocecità congenita, alcuni dei quali non sono in grado di sviluppare neanche una forma di comunicazione basata sul tatto se non inseriti per tempo in un percorso terapeutico ed educativo specializzato. Ma la percezione non è solo negli organi di senso: l'informazione sensoriale passa nel sistema nervoso periferico e infine in quello centrale, per venire processata da specifiche aree della corteccia cerebrale. In coloro che perdono la funzionalità di uno o più degli organi di senso vi è un vero e proprio adattamento della corteccia cerebrale, le cui aree ormai inutilizzate vengono gradualmente **riciclate** per la ricezione e l'analisi di altri stimoli sensoriali. (Berardi e Pizzorusso, 2006; Giovanelli, 1998; Dammeyer, 2011; Bruce, 2005; Bear *et al.*, 2007)

Risulta chiaro come la percezione, anche solo nella sua veste di **mezzo attraverso cui esperiamo il mondo**[1], sia di fondamentale importanza per la nostra vita. La percezione ricopre inoltre un ruolo primario nel comportamento, perchè ciò che esperiamo determina ciò che facciamo, direttamente o indirettamente.

Per via della sua centralità nell'esistenza umana, la percezione è stata oggetto di riflessioni filosofiche fin dagli antichi Greci, ed è stata studiata sistematicamente dalla Psicologia fin dai suoi albori, in particolare dalla scuola della **Gestalt** nella prima metà del '900. Quest'ultima sviluppò un **approccio fenomenologico** allo studio della percezione, per il quale dobbiamo prendere in considerazione non la **realtà oggettiva**, misurabile sperimentalmente, ma la nostra esperienza soggettiva del mondo, rimarcando la centralità delle elaborazioni che la nostra mente compie sulle informazioni sensoriali. I gestaltisti fecero proprio il motto "il tutto è più della somma delle sue parti" e portarono avanti, in modo preciso e sistematico, un grande numero di esperimenti

[1] In questo caso è più appropriato chiamarla **sensazione**, per distinguerla dallo stadio successivo, chiamato **percezione** in Psicologia, in cui avviene il processamento e l'organizzazione degli stimoli sensoriali

con le illusioni ottiche, principale evidenza della mancata corrispondenza tra realtà ed esperienza percettiva. (Canestrari e Godino, 2002; Mecacci, 2001)

Verso la fine degli anni '70, lo psicologo **J.J. Gibson**, dopo decenni di studi sperimentali sulla percezione e in contrasto con le teorie cognitivo-computazionali dell'epoca, elaborò un **approccio ecologico** per il quale la nostra percezione della realtà è strettamente legata ai nostri sensi e all'ambiente che ci circonda. Studiando la percezione dobbiamo tenere conto di dove un particolare organismo si sia evoluto e dell'ambiente in cui si muova, poiché la sua particolare esperienza percettiva è fortemente dipendente da entrambi. L'apparato sensoriale di un organismo fornisce tutte le informazioni necessarie a percepire, essendo **tarato** per cogliere le particolari invarianti strutturali dell'ambiente circostante. Per Gibson, il **movimento è necessario** all'osservatore, permettendogli di variare la porzione di mondo esperita tramite i sensi per poter meglio comprendere e agire sull'ambiente. (Gibson, 1979)

La percezione (in particolare quella visiva) diventa quindi non più un processo passivo e astratto, ma **attivo**, dinamico, per la cui comprensione bisogna tenere conto del **corpo** dell'organismo che vogliamo studiare e dell'**ambiente** che lo circonda e in cui si è evoluto. Questi due concetti vengono chiamati *embodiment* e *situatedness*, e sono alla base di alcune teorie moderne sulla cognizione (sia biologica che artificiale). (Pfeifer e Bongard, 2006)

1.1.1 La percezione in Robotica

La parola *robot* è stata usata per la prima volta in (Čapek, 1921), prendendo ispirazione dalla parola in lingua ceca *robota* che significa **lavoro forzato**. Essa indica quindi un essere usato come schiavo, capace di fare lavori meccanici e ripetitivi che l'uomo non può o non vuole più fare. La parola *robotics* è stata invece per la prima volta utilizzata in (Isaac AsMirolli M., 1941), ad indicare la scienza che si occupa della progettazione dei robot.

Pur se nata in un contesto fantascientifico, la robotica è divenuta molto presto una disciplina eterogenea, con campi di applicazione, tecniche e modelli anche molto differenti tra loro. Se ad esempio nell'immaginario comune il tipico robot è un umanoide (i cosiddetti **androidi**), attualmente il mercato più grande appartiene ai robot industriali, che semplificano il lavoro in fabbrica, automatizzandolo e velocizzandolo. Il tipico braccio robotico industriale non necessita di molta sensoristica, perché il suo comportamento è programmato totalmente o quasi, e l'ambiente in cui opera tende a essere costante o con poche variazioni[2].

Gran parte della robotica da esplorazione, che richiede sensori più sofisticati di quella industriale, è stata fino a qualche anno fa costituita da mezzi almeno in parte comandati a distanza[3]. I sensori più comuni per questo tipo di robot sono, tra gli altri, telecamere con una buona risoluzione, per permettere all'operatore remoto di comprendere l'ambiente, e sensori a infrarossi per rilevare la distanza da oggetti vicini.

Negli ultimi tempi l'interesse verso la **robotica autonoma** è drasticamente aumentato, e questo ha portato alla necessità di un cambio di paradigma nella progettazione di robot destinati ad agire senza controllo su un ambiente mutevole e sconosciuto. I sensori devono essere, come è per noi umani, la finestra sul mondo degli agenti artificiali, e devono poter essere gestiti velocemente ed efficacemente dal sistema di controllo per permettere una reazione opportuna a una situazione di pericolo o a mutate condizioni ambientali.

Una integrazione completa e veloce tra le informazioni catturate tramite i sensori e il successivo processamento (con risposta comportamentale) da parte del sistema di controllo si scontra bruscamente con le tecniche classiche per la progettazione di sistemi intelligenti, che tendono a richiedere una gran quantità di computazioni difficilmente

[2]Negli ultimi anni i robot industriali sono stati invece dotati di molti sensori quali telecamere, sensori di movimento, sensori di tatto e pressione, con l'obiettivo di ridurre danni accidentali agli operai che vi lavorano vicino: molti robot industriali sono in grado di fermarsi immediatamente in caso di contatto con un oggetto/corpo non previsto, o quando una persona si avvicina troppo al robot stesso.

[3]Tra i principali ricordiamo i mezzi da ricognizione aerea, sottomarina e spaziale.

sostenibile da un computer a bordo del robot[4].

Per questo motivo, le moderne teorie sulla cognizione vengono sviluppate parallelamente da studiosi di agenti biologici e artificiali: i concetti di embodiment e situatedness costituiscono infatti un eccellente *framework* per lo sviluppo di robot autonomi capaci di agire prontamente in risposta a stimoli non prevedibili a priori. (Pfeifer e Bongard, 2006; Nolfi, 2009)

1.2 La visione artificiale

In questa tesi si è preso in considerazione solo uno tra i vari tipi di percezione: quella visiva. La visione artificiale, dagli anni '60 a oggi, ha fatto molti passi in avanti, cambiando spesso paradigma e sfruttando molti dei risultati della Psicologia e della Neurofisiologia. **Nikos Drakos** ha delineato brevemente il percorso compiuto dalla visione artificiale negli ultimi decenni[5]:

1965 Visione come tecniche di riconoscimento:
clustering, classificazione

1975 Visione come comprensione delle immagini (AI):
segmentazione, rappresentazione della conoscenza

1980 Visione come ricostruzione:
forma da X, *world modeling*

1985 Visione attiva:
profondità cinetica, *visual servoing*

1990 Visione come processo:
integrazione visiva, *continous operation*, controllo

[4]Per questo motivo, una parte delle computazioni è a volte effettuata remotamente, cosa non sempre possibile e comunque soggetta ai tipici problemi della comunicazione wireless: disturbi, interruzioni, spionaggio.

[5]©1993, 1994, Nikos Drakos, Computer Based Learning Unit, University of Leeds.

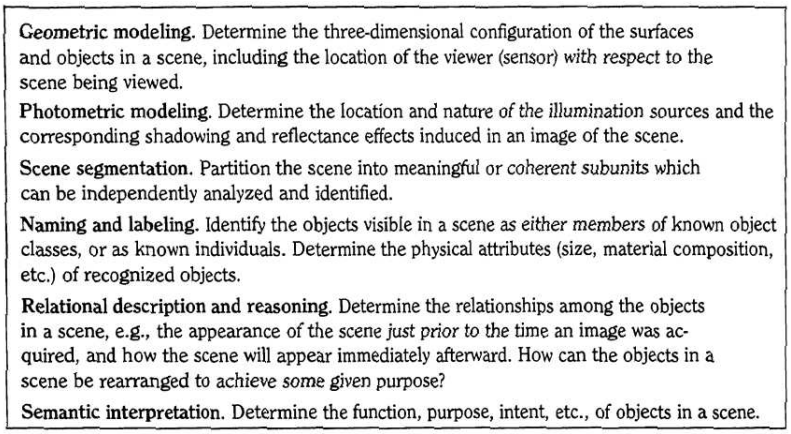

Figura 1.1: Requisiti funzionali per un generico sistema di Visione Artificiale

Fischler delinea quelli che lui chiama **requisiti funzionali** per un sistema di visione artificiale, che riportiamo in figura 1.1.

Il modello predominante in questa disciplina è il cosiddetto **paradigma signals-to-symbols**, che prende ispirazione dal proverbio: "se sei un martello, ogni cosa ti sembra essere un chiodo", che Fischler ripropone come: "se sei un computer digitale, allora ogni cosa ti sembra essere un numero o un simbolo". Abbiamo infatti bisogno, per sviluppare un sistema di visione artificiale, di una parte *hardware* che raccolga le informazioni ambientale (talvolta modificandole parzialmente), ma soprattutto che trasformi queste informazioni in qualcosa di comprensibile per un computer, sul quale sarà installato un *software* capace di darci le risposte che cerchiamo su una determinata scena visiva. (Fischler e Firschein, 1987)

Generalmente il software è costituito da una pila di algoritmi eseguiti in sequenza, per fare analisi di basso, medio e alto livello, ottenendo una rappresentazione esplicita di ciò che ci interessa. Questo processo è rappresentato schematicamente nella figura 1.2.

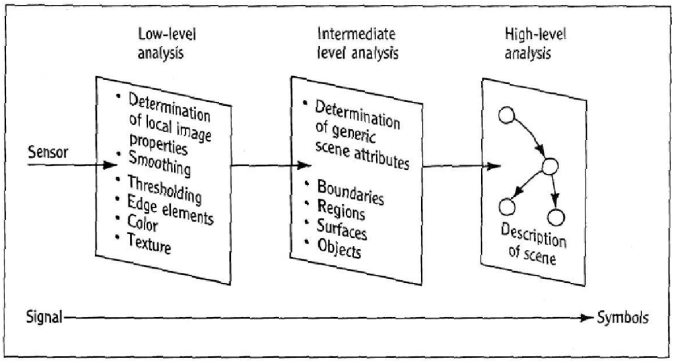

Figura 1.2: Il paradigma Signals-to-Symbols per la Visione Artificiale

1.2.1 La visione attiva

Nel primo articolo (Aloimonos *et al.*, 1988) in cui viene utilizzato il termine **visione attiva**, è sottolineata l'importanza del movimento dell'osservatore nel ridurre la complessità di alcuni tipi di analisi visiva:

> *We prove that an active observer can solve basic vision problems in a much more efficient way than a passive one. Problems that are ill-posed and nonlinear for a passive observer become well-posed and linear for an active observer. In particular, the problems of shape from shading and depth computation, shape from contour, shape from texture, and structure from motion are shown to be much easier for an active observer than for a passive one.*

Poco tempo dopo Bajcsy in (Bajcsy, 1988), che può essere considerato l'articolo fondante della **visione attiva artificiale**, pone l'accento sul fatto che il suo approccio ha come obiettivo lo studio dei modelli e delle strategie di controllo finalizzate alla percezione, piuttosto che quello di diventare un paradigma scientifico. Secondo l'autrice, un sistema di visione attiva sviluppato modellando i sensori, gli oggetti, l'ambiente circostante e le loro interazioni, risulta essere più efficiente nel risolvere un compito di navigazione, o di riconoscimento, rispetto a un sistema passivo.

Nello stesso articolo l'autrice propone un modello generale per un sistema di visione attiva per il riconoscimento di forme tridimensionali, composto da varie fasi, elencate di seguito dal livello più basso a quello più alto:

1. Controllo del dispositivo fisico, con l'obiettivo di ottenere una prima immagine della scena con un'adeguata messa a fuoco
2. Controllo del sistema binoculare (hardware e software), al fine di acquisire informazioni sulla profondità ed ottenerne una mappa
3. Controllo della mappa di profondità, finalizzata all'estrazione di parti di superfici
4. Riconoscimento degli oggetti
5. Interpretazione della scena

In questo modello si assume che tra uno stadio e l'altro vi siano *feedback*, che permettano lo scambio di dati in entrambi le direzioni.

Si tratta, come la stessa autrice sottolinea, di uno tra i possibili modelli generali per la visione attiva, ma dal momento della pubblicazione del suo articolo c'‘e stato un progressivo incremento di interesse verso questo campo tra ricercatori di varie discipline.

Pochi anni più tardi un altro ricercatore (Ballard, 1991) pubblica un articolo fondamentale in questo campo. Ballard utilizza il nome **visione animata** per non confondere la percezione attiva con l'*active sensing*[6], cosa sottolineata anche da Bajcsy nell'articolo già citato.

Ballard concentra il suo studio sulla riduzione della complessità di un sistema visivo con controllo dei movimenti dell'occhio, sottolineando come, in questo modo, sia necessaria una risoluzione della camera molto più bassa per ottenere risultati ottimi.

[6]Si parla di active sensing quando un sistema è dotato di sensori che agiscono sull'ambiente per ottenere informazioni. Un esempio tipico sono i sensori di distanza a laser, o i sonar. Non ha nulla a che vedere con la percezione attiva.

Per l'autore, la visione animata è il miglior paradigma qualora si voglia implementare un sistema di visione artificiale *real time*, che risponda prontamente agli stimoli visivi.

Viene inoltre fatto un paragone con studi sulle neuroscienze della visione, mostrando come l'occhio umano sia dotato di un'area molto piccola del campo visivo ad altissima risoluzione, chiamata **fovea**. Allontanandosi dalla fovea la risoluzione diminuisce esponenzialmente. L'occhio umano (e della maggior parte degli animali) si muove continuamente con le cosiddette **saccadi**, direzionando la propria fovea verso l'ambiente circostante, con *pattern* non ancora del tutto compresi.

Table 1
A comparison of the computational features of fixed camera vision and animate vision.

Fixed camera vision	Animate vision
Local constraints that relate physical parameters to photometric parameters are underdetermined.	Local constraints are sufficient.
Minimalist constraints such as smoothness used to regularize the solution.	Maximalist constraints such as specific behavioral assumptions used to obtain solution
Algorithm requires parallel iterations over the retinally indexed array.	Algorithm is local and has a constant time solution.
Frame of reference is camera-centered (egocentric).	Frame of reference is fixation point centered (exocentric).

Figura 1.3: I vantaggi della visione animata rispetto ai sistemi classici.

Si mostra inoltre come, negli studi recenti sulla visione animale, vi sia un **forte legame tra percezione e azione**: l'informazione visiva a bassa risoluzione catturata dalla periferia della retina viene utilizzata per rispondere velocemente e **senza consapevolezza** a particolari stimoli visivi. Ballard introduce quindi nel campo della visione artificiale i primi elementi dell'approccio sensomotorio alla cognizione, ricollegandosi agli studi di neuroscienze e psicologia della percezione post-gibsoniani.

La consapevolezza che noi percepiamo il mondo stabile e uniforme nonostante la visione sia un processo irregolare, costituito da migliaia di saccadi anche per piccoli task visivi, dove solo una piccola porzione del mondo alla volta è ad alta risoluzione, ci costringe a porre la questione: come è possibile che questo accada?

La risposta di Ballard è che il sistema visivo genera l'illusione di una stabilità tridimensionale grazie al fatto che è in grado di eseguire comportamenti in modo molto rapido. La nostra percezione del mondo dipenderebbe quindi più dalla possibilità di esplorare qualsiasi scena visiva molto velocemente ed efficacemente piuttosto che da una rappresentazione molto fedele della realtà nella nostra mente.

Riportiamo in figura 1.3 una caratterizzazione del modello proposto da Ballard. I sistemi di visione animata possono:

1. far uso di ricerca fisica, ad esempio muovendo la camera più vicino agli oggetti, cambiando il focus o il punto di vista;

2. fare movimenti (approssimativamente) conosciuti della camera, in modo automatico, diminuendo il costo computazionale rispetto all'analisi di una singola immagine;

3. usare un sistema di coordinate esocentrico, ad esempio fissato sull'oggetto da analizzare invece che sull'osservatore;

4. usare algoritmi relativi (o qualitativi), cosa non possibile con un sistema di coordinate egocentrico;

5. segmentare (in modo pre-categoriale) aree di interesse nell'immagine, grazie al controllo della foveazione;

6. sfruttare il contesto ambientale, aiutati dal fatto che un sistema di coordinate centrato sull'oggetto è invariante rispetto al movimento dell'osservatore, diversamente da un sistema centrato sull'osservatore;

7. usare in modo nativo molti algoritmi di apprendimento. In particolare, sono perfetti per i sistemi di apprendimento che usano *indexical reference*, e in particolar modo per gli algoritmi di apprendimento supervisionato.

L'autore afferma nelle conclusioni che, sebbene una rappresentazione esplicita del mondo sembri importante per un buon sistema visivo, essa sia all'atto pratico inutile.

> *Animate systems that rapidly change their coupling with the real world place a great premium on maintaining elaborate representations of the world. However, it may be the case that memorizing such representations is unnecessary, since they can be rapidly and incrementally computed on demand.*

1.3 L'immagine del mondo nella testa

Negli ultimi decenni le teorie a favore di una cognizione non rappresentazionalista si sono moltiplicate, partendo dai concetti di *embodiment*, *situatedness* e dal **ciclo percezione-azione**, come accennato precedentemente. Verso l'inizio del nuovo millennio **O'Regan e Noë** hanno proposto un *framework* sulla visione e sulla coscienza visiva (O'Regan e Noë, 2001), costruito sulla base (supportata, a detta degli autori, da molti studi sperimentali) dell'idea che **non abbiamo bisogno di alcuna rappresentazione interna della realtà**: il mondo in cui viviamo viene usato come una memoria esterna[7].

L'idea contro cui vanno gli autori è che siano proprio le rappresentazioni interne, attivate da eventi reali, a dare vita a quella che possiamo chiamare l'esperienza soggettiva del vedere (o del sentire in generale). Poiché non vi sono rappresentazioni interne, argomentano O'Regan e Noë, l'esperienza visiva deve nascere da qualcos'altro: dalla nostra *abilità* nello sfruttare ed integrare l'informazione visiva (che dipende sia dalla struttura del mondo che del nostro apparato sensoriale), o, nelle parole degli autori, dall'esercizio della padronanza delle appropriate **contingenze sensomotorie**.

Il concetto di contingenza sensomotoria si spiega facilmente con un esempio. Consideriamo un sistema automatico di guida per missili dotato di telecamera. Quando un missile si muove nell'aria per raggiungere un bersaglio, l'informazione visiva può cambiare in molti (predicibili) modi. Quando il missile si avvicina al bersaglio, questo

[7] Dall'articolo sopracitato: *The outside world serves as its own, external, representation.*

diventa più grande, viceversa diventa più piccolo. Quando il bersaglio vira bruscamente, scompare dal campo visivo della telecamera del missile, e così via. Ora, se il missile è in grado di gestire queste variazioni[8] raggiungendo con successo il suo scopo, se cioè "conosce" le leggi che governano ciò che succede quando si esperisce ciò che si può esperire quando si insegue un aereo, diremo che ha **padronanza delle contingenze sensomotorie** legate all'inseguimento di aerei.

Nel loro approccio, l'esperienza sensoriale cosciente è una modalità motoria, dipende da ciò che facciamo, e da quanto siamo "bravi" a farlo. Questa teoria ha sollevato un'ampia discussione tra chi la condivide (del tutto o parzialmente) e chi la respinge, confutandola. Molti di coloro i quali la condividono, tuttavia, pongono dubbi sulla realisticità di un cervello senza rappresentazioni interne. In particolare si sottolinea come di rappresentazioni, con studi neurofisiologici, ne siano state trovate molte, mentre le cosiddette **rappresentazione figurative** (per le quali il mondo esterno è memorizzato nel nostro cervello come una sorta di video, un insieme di fotogrammi recuperabili all'occorrenza) sarebbero quelle contro cui dovrebbero andare gli autori e che non sembrerebbero esistere realmente.

Nonostante O'Regan e Noë abbiano attaccato principalmente la cognizione rappresentazionalista, la maggiore critica che è stata rivolta a questa teoria è di essere contraddittoria sull'argomento della coscienza visiva, e di non dare, di fatto, una soluzione soddisfacente al cosiddetto **hard-problem** della coscienza, posto pochi anni prima da **Chalmers** con questa frase (Chalmers, 1995, 1997):

> *Why does the feeling which accompanies awareness of sensory information exist at all?*

Il problema ruotava (e ruota tutt'ora) intorno alla peculiarità dell'esperienza cosciente. Perché esiste? A cosa serve realmente? E cosa ancora più importante, siamo

[8] Dovute sia alle caratteristiche dell'ambiente (*space-related*) sia del proprio apparato sensoriale (*apparatus-related*)

sicuri che il cervello è anche condizione sufficiente, oltreché necessaria, allo sviluppo di una mente cosciente?

Gli autori non escludono che vi sia un'esperienza cosciente, sottolineando però come l'hard-problem, se applicato alla coscienza visiva, sia un falso problema, un'illusione: non cè nulla come i *visual qualia*[9] o come l'*intenzionalità*[10], sottolineano O'Regan e Noë.

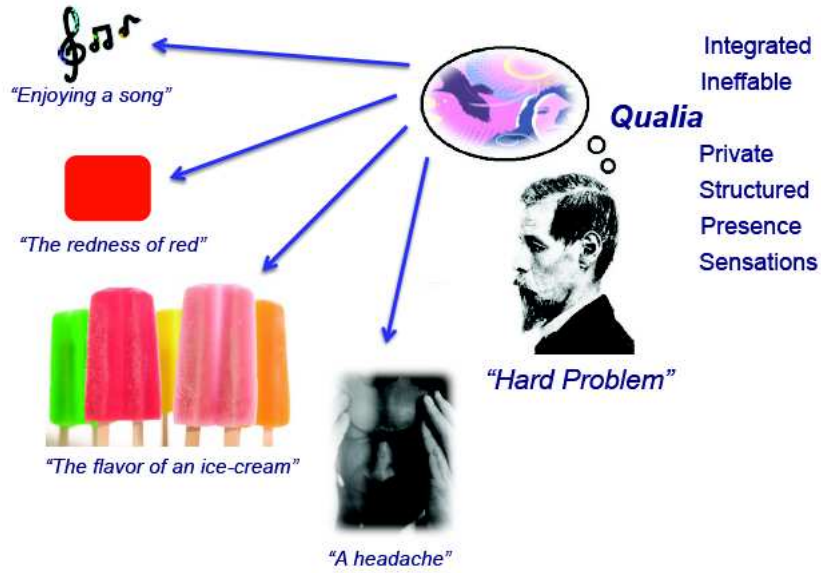

Figura 1.4: Rappresentazione figurativa dei qualia.

Il dibattito è ancora aperto, ed è vivo soprattutto negli ambienti accademici della Filosofia e della Psicologia: un ulteriore approfondimento va oltre gli scopi di questa relazione.

[9]Per visual qualia si intendono gli aspetti soggettivi dell'esperienza visiva cosciente, quali ad esempio la "sensazione di vedere rosso".

[10]Lo psicologo Brentano, a inizio novecento, introdusse il concetto di intenzionalità come l'idea che la coscienza (e quindi i fenomeni psichici diversamente dai fenomeni fisici) sia sempre diretta verso un oggetto: credere, desiderare, hanno sempre un cosiddetto "oggetto intenzionale", il creduto, il desiderato.

Capitolo 2

Metodi e Tecniche della Robotica Evolutiva

2.1 Intelligenza Artificiale

Nonostante nei millenni di storia umana siano documentati numerosi tentativi di costruzione di artefatti simili all'uomo, ripresi anche in chiave mitologica, e molti filosofi e pensatori dall'antica Grecia in poi abbiano provato a definire e formalizzare l'atto di pensare e l'uso della ragione, solo a metà del secolo scorso possiamo parlare di vera e propria nascita dell'Intelligenza Artificiale come disciplina.

Come il nome lascia intuire, questa disciplina ha come obiettivo la costruzione di artefatti che siano intelligenti. Di entrambi i concetti (artefatto e intelligenza), ma in particolare del secondo, sono state date numerose definizioni, spesso contrastanti, e ad oggi nessuna può essere considerata definitiva.

Russell e Norvig (2005) descrivono quattro diverse definizioni di intelligenza artificiale, fatte proprie da alcuni gruppi di ricercatori, tra loro piuttosto distinti e talvolta in aspra contrapposizione.

Occorre, secondo gli autori, prima di tutto distinguere tra chi vede l'IA come il

tentativo di costruire qualcosa di simile agli esseri umani, e chi invece ha come modello la **razionalità**, come è stata studiata in filosofia fin dai suoi albori. La questione è semplice: indubbiamente gli esseri umani possono essere razionali, possono far uso della ragione in modo sistematico e formale per risolvere problemi del mondo reale o per riflettere su questioni meno tangibili. È però altrettanto vero che gli esseri umani possono comportarsi in modo del tutto irrazionale, possono agire emotivamente, hanno riflessi automatici e reazioni inconsce: tutte cose che non possono rientrare nella sfera della razionalità.

Vi è un'ulteriore divisione: in entrambi i casi, l'intelligenza può essere riscontrata nel **comportamento intelligente** oppure va indagata a livello più profondo, nei **meccanismi del pensiero**.

Riassumendo, per gli autori l'IA ha almeno quattro obiettivi diversi, portati avanti normalmente da distinti gruppi di ricerca:

Agire come esseri umani Noi amiamo, facciamo amicizia, ci scambiamo pettegolezzi, condividiamo paure e gioie. Abbiamo scopi, desideri, ambizioni. Un artefatto potrà dirsi intelligente solo quando sarà capace di fare tutto questo, e in particolare, sulla scia del **Test di Turing**, quando sarà talmente integrato con gli esseri umani che essi non potranno distinguerlo da uno di loro.

Pensare come esseri umani Cosa rende l'uomo intelligente? La sua mente. Studiando la mente (con introspezione e indagine psicologica sperimentale), si può provare a trovare un modello della cognizione umana, e renderlo abbastanza solido e generale da poterlo implementare in una macchina.

Pensare razionalmente Aristotele credeva che la logica da lui definita fosse un buon modello delle leggi del pensiero umano. Oggi l'obiettivo della cosiddetta **tradizione logicista** dell'IA è quello di trovare sistemi logici formali che permettano

di descrivere il mondo e di creare sistemi intelligenti che deducano una soluzione corretta e possibilmente ottima (la più razionale) a partire da qualsiasi problema.

Agire razionalmente Togliamo dagli esseri umani le emozioni (e in generale ciò che è comunemente considerato irrazionale) e otteniamo un agente capace di trovare soluzioni ragionevolmente corrette e attinenti ai problemi che incontra, di imparare dai propri errori e di aumentare le proprie conoscenze. Questo approccio mira a identificare i principi generali dell'agire razionalmente, implementandoli in macchine quali robot e computer. Bisogna sottolineare che, sebbene questa sia la categoria più comune, la maggior parte della ricerca si focalizza sulla soluzione di singoli problemi (giocare a scacchi, categorizzare un insieme di oggetti) piuttosto che sul costruire un agente completo.

Sono molte le sotto-discipline che tentano di risolvere i numerosi problemi di ciascuno di questi quattro gruppi. Ad esempio l'apprendimento automatico utilizza principalmente metodi statistici per creare programmi che generalizzino proprietà in grandi insiemi di dati, e che siano successivamente in grado, di fronte ad input nuovi e sconosciuti, di emettere un output appropriato basandosi sui *pattern* appresi precedentemente. Sistemi molto sofisticati di apprendimento automatico sono stati usati per tentare (senza successo) di passare il test di Turing, in una competizione che si svolge ogni anno[1].

Vi sono poi i sistemi di controllo dei robot, che a seconda dell'approccio del progettista possono usare sistemi logico-deduttivi, statistici, bio-ispirati o basati sul comportamento (*behaviour-based*).

Fino agli anni '80, la ricerca in robotica e IA era in mano prevalentemente all'ingegneria e alla logica. La Psicologia Cognitiva elaborava modelli algoritmici di funzioni mentali, basati talvolta su ricerche in IA, che a sua prendeva ispirazione dai lavori dei cognitivisti. Questo approccio si rivelò fallimentare in quanto, nonostante la tecnologia

[1]Loebner prize: http://www.loebner.net/Prizef/loebner-prize.html

migliorasse costantemente, non si riusciva a progettare un agente che fosse in grado di effettuare decentemente ciò che è nella quotidianità di ogni essere umano. L'informatica guadagnò moltissimo da questi decenni di ricerca: algoritmi efficienti per il *path-finding* e per la categorizzazione di file di testo in gruppi affini, sistemi robotici per l'industria meno costosi e più precisi, e così via. Ma non si andò più vicino alla creazione di un agente veramente intelligente di quanto non si fosse a inizio secolo.

Tra gli anni '80 e '90 tornarono alla ribalta le reti neurali, utilizzate dagli psicologici connessionisti. Il connessionismo, sull'onda delle ricerche neurofisiologiche precedenti, vedeva la mente come una rete piuttosto che come un programma, e per rappresentare i suoi modelli di funzione cognitive (ad esempio apprendimento e categorizzazione) usava reti neurali artificiali, molto migliorate rispetto al percettrone presentato a inizio '900.

Contemporaneamente si andarono sviluppando approcci alla robotica differenti rispetto a quello ingegneristico, prima fra tutti la *Behaviour-based Robotics* di Rodney Brooks. Piuttosto che programmare completamente un sistema di controllo per robot che raggiunga un certo obiettivo, si possono trovare dei "sotto-comportamenti" molto più semplici da realizzare ed eseguire, e dotando il robot di un sistema per coordinare e scegliere in che ordine eseguire i sotto-comportamenti si può assistere all'emergere di una sequenza di azioni che risolva il compito richiesto. Ciò è particolarmente utile quando il comportamento da realizzare è molto complesso e non ne abbiamo un modello preciso.

Dagli anni '90 in poi cominciarono a formarsi gruppi di ricercatori convinti della necessità di estendere la visione connessionista: si argomentava, sulla base anche di già citati studi di psicologia ed ecologia, che senza un corpo e senza un ambiente in cui un agente possa crescere ed evolvere insieme ad altri agenti, la cognizione non si potrebbe sviluppare. L'idea di studiare la cognizione facendo evolvere in modo automatico dei robot, piuttosto che facendoli progettare e costruire da esseri umani, portò alla nascita della disciplina conosciuta come **Robotica Evolutiva**.

Tutti questi tentativi di soluzione al problema della creazione di una mente artificiale sono ancora portati avanti e sviluppati da molti gruppi di ricerca, sebbene sia da notare che l'approccio matematico-ingegneristico rimane ancora oggi il piùtilizzato (specialmente nell'industria).

2.1.1 Algoritmi Genetici

Gli algoritmi genetici (AG) sono una classe di algoritmi di ottimizzazione (più propriamente detti algoritmi di ricerca) che prende ispirazione dall'evoluzione biologica (in prevalenza, ma non necessariamente, dalla teoria dell'evoluzione darwiniana). Per spiegarne il funzionamento faremo riferimento ad alcuni algoritmi di ricerca sviluppati prima degli AG (Russell e Norvig, 2005).

Supponiamo di avere un problema per cui non si abbia una soluzione analitica. Si ipotizza una soluzione, e la si testa. Se si procede cambiando piccole parti della nostra soluzione, una alla volta, potremmo ottenere risultati migliori o peggiori. Teniamo i cambiamenti che portano a una soluzione migliore e scartiamo quelli che peggiorano le *performance*. Stiamo cercando una soluzione tramite una tecnica chiamata *hill climbing* (figura 2.1), perché si vedono le possibili soluzioni come punti di un grafico, dove sull'asse delle ascisse c'è la combinazione delle componenti di una soluzione (per due dimensioni si usa un asse per ogni componente) e sull'asse delle ordinate le performance di quella particolare soluzione, e dove il "picco" più alto è la soluzione ottima. Chiamiamo "passi" lo spostamento da una soluzione candidata ad un'altra.

Questo algoritmo è piuttosto efficiente ma cade facilmente preda dei cosiddetti *local maxima*, quei punti dai quali ogni soluzione in un dato intorno ha performance peggiori della soluzione corrente, che però non è la migliore in assoluto.

Per evitare di incappare in massimi locali sono stati sviluppati vari algoritmi, tra i quali lo *stochastic hill-climbing*, che sceglie il passo da effettuare casualmente (sempre, comunque, tra i "vicini" della soluzione corrente) e accetta la nuova soluzione nel caso

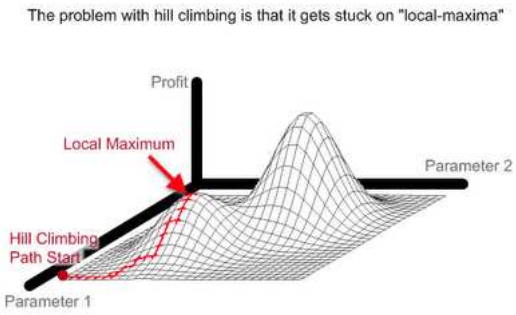

Figura 2.1: Il problema dei *local maxima* con l'hill-climbing.

essa sia migliore della precedente.

Un sistema più sofisticato è il cosiddetto *simulated annealing*, che prende ispirazione dal processo, utilizzato in metallurgia, di riscaldamento e lento raffreddamento di un metallo. Durante il riscaldamento gli atomi guadagnano energia e si muovono liberamente, mentre il raffreddamento controllato permette al materiale di impostarsi su una configurazione più stabile e con meno difetti rispetto a prima.

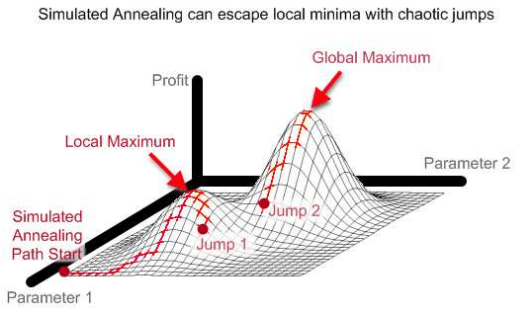

Figura 2.2: Il simulated annealing può risolvere il problema dei local maxima

Nei termini del nostro discorso, definiamo un numero fisso di iterazioni e una temperatura massima, e creiamo una soluzione iniziale. Per ogni iterazione prendiamo una soluzione candidata tra i "vicini" della soluzione corrente. Se è migliore di quest'ultima, la prendiamo. Se è peggiore, c'è comunque una probabilità di prenderla. Questa probabilità è tanto più alta quanto più elevata è la "temperatura" attuale del sistema (che viene decrementata secondo uno schema), ed anche quanto più simile è la performance

della soluzione candidata rispetto alla precedente.

Col progredire delle iterazioni, quindi, il simulated annealing tende a essere simile all'hill climbing, ma con un numero di iterazioni abbastanza grande evita di bloccarsi su minimi locali.

Se invece creiamo un insieme iniziale di soluzioni casuali, le testiamo sulla base di una funzione che, presa in input la soluzione, dia un valore numerico per indicarne la performance, e scartiamo quelle peggiori; se alle migliori applichiamo degli operatori quali ad esempio la **mutazione** (un cambiamento casuale di una piccola porzione della soluzione, applicato con un probabilità molto bassa) e l'**incrocio** (due o più soluzioni vengono combinate tra loro in vari modi), e tramite questi operatori generiamo delle nuove soluzioni (che chiameremo **individui**), andandole a sostituire agli individui peggiori che abbiamo precedentemente scartato; se, infine, ripetiamo questa procedura per un numero di **generazioni** sufficiente a trovare un individuo abbastanza performante, allora abbiamo appena costruito un algoritmo genetico, che in sostanza è un po' come mandare in parallelo tanti *simulated annealer*, e ad ogni generazione combinare in vari modi le soluzioni migliori di ciascuno[2].

La funzione di test si chiama **funzione di fitness**, e l'individuo viene rappresentato tramite un **cromosoma**, i cui **geni** possono codificare caratteristiche di ogni tipo (morfologiche o comportamentali).

Se la funzione di fitness è progettata bene e il numero di generazioni è sufficiente, gli AG possono trovare soluzioni estremamente efficienti per un ampio spettro di problemi. Possono ad esempio massimizzare o minimizzare funzioni complesse, risolvere problemi ingegneristici in ambito robotico, aerospaziale o meccanico, trovare velocemente soluzioni sub-ottimali a problemi NP-completi[3] e così via.

[2] Alternativamente, si può vedere il simulated annealing come un algoritmo genetico la cui popolazione è composta da un solo individuo, e in cui la perturbazione (nell'AG chiamata mutazione) ha una probabilità fissa di accadere ad ogni generazione

[3] Ad esempio per il problema del commesso viaggiatore si ottengono soluzioni *near-optimal*

2.1.2 Reti Neurali

Le reti neurali artificiali (RN) sono una rappresentazione astratta del nostro sistema nervoso, che contiene una collezione di neuroni i quali comunicano fra loro mediante connessioni dette assoni.

Il primo modello di neurone artificiale fu proposto nel 1943 da **McCulloch** e **Pitts** nei termini di un modello computazionale dell'attività nervosa. A questo modello sono seguiti altri proposti da **John Von Neumann**, **Marvin Minsky**, **Frank Rosenblatt** (il cosiddetto Percettrone) e molti altri.

Esistono molti tipi diversi di neuroni nel nostro sistema nervoso, ma per comodità descriveremo il cosiddetto **neurone prototipico**, dal quale le cellule nervose reali differiscono solo marginalmente.

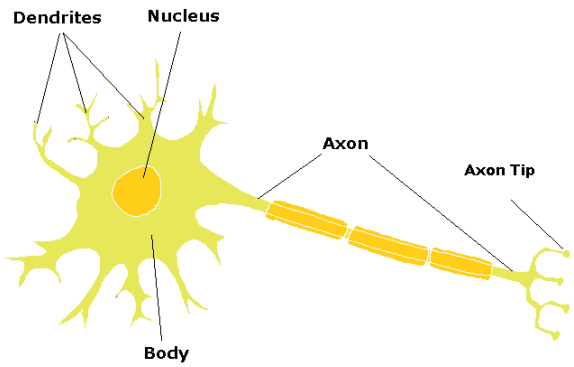

Figura 2.3: Il neurone prototipico

Si nota un'ampia arborizzazione, costituita dai molti **dendridi** e dal singolo **assone**. L'assone si connette ai dendriti di altre cellule nervose, inviandogli segnali attraverso giunzioni dette **sinapsi**. Normalmente un neurone riceve molti segnali in input, ma ne invia pochi in output.

L'informazione ricevuta dalle sinapsi sui dendridi è di tipo elettrico o chimico, e può avere sia carattere eccitatorio che inibitorio. Più segnali che entrano contempo-

raneamente nel neurone si sommano, e se l'informazione eccitatoria è predominante il neurone si attiva e invia lui stesso un segnale attraverso l'assone.

Il modello più semplice di rete neurale artificiale, biologicamente ispirato al neurone prototipico, è il percettrone in figura 2.4.

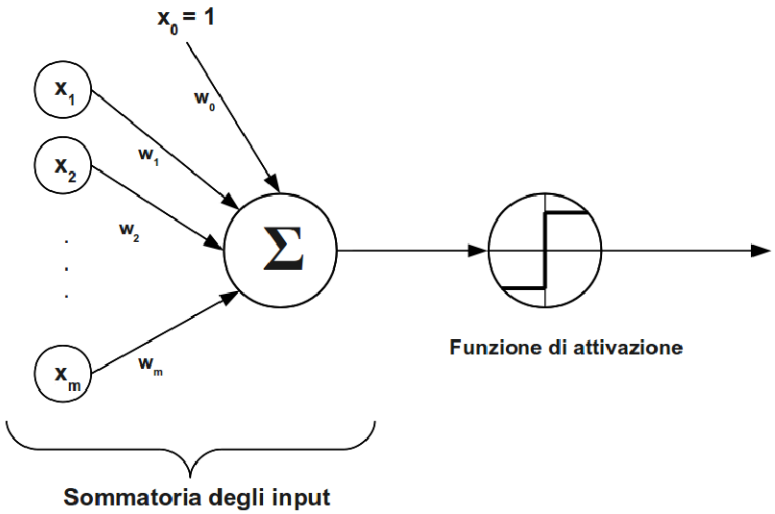

Figura 2.4: Il percettrone di Rosenblatt

Il percettrone prende in input un vettore di valori reali, $\vec{x} = (x_1, x_2, \cdots, x_m)$, ne calcola una combinazione lineare con i **pesi sinaptici** $w_1, \cdots, w_m$, e ritorna 1 se il risultato è maggiore di una certa **soglia**, altrimenti ritorna -1. In formule:

$$o(x_1, \cdots, x_m) = \begin{cases} 1 & w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_m x_m > 0 \\ -1 & altrimenti \end{cases}$$

Assunto che la quantità $(-w_0)$ è la soglia che la somma degli input pesati deve superare affinché il percettrone torni 1 (e risulti, quindi, **attivo**), possiamo considerare

un input fisso aggiuntivo $x_0 = 1$, chiamato *bias*, così da semplificare la notazione:

$$w_0 + w_1 x_1 + \cdots + w_m x_m = w_0 x_0 + w_1 x_1 + \cdots + w_m x_m = \sum_{i=0}^{m} w_i x_i = \vec{w} \cdot \vec{x}$$

La funzione $o(\vec{x})$ è chiamata **funzione di attivazione**, e nel caso del percettrone semplice è la cosiddetta funzione segno

$$o(\vec{x}) = sign(\vec{w} \cdot \vec{x})$$

dove

$$sign(x) = \begin{cases} 1 & x > 0 \\ -1 & altrimenti \end{cases}$$

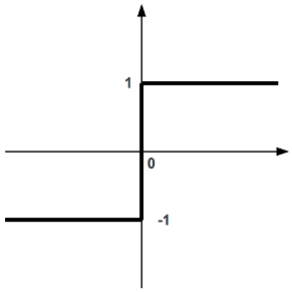

Figura 2.5: La funzione segno.

Si possono utilizzare altre funzioni di attivazione, preferibilmente non lineari (come la funzione sigmoide).

È possibile far apprendere una notevole quantità di funzioni a questo semplice modello, tramite un algoritmo di apprendimento che modifichi leggermente i pesi sinaptici utilizzando un insieme di dati selezionato dallo sperimentatore, chiamato *training set*. Quando il percettrone sarà addestrato su questo insieme, lo si potrà testare su altri input così da verificare le sue capacità di generalizzazione.

Ovviamente nei decenni sono stati sviluppati molti altri tipi di RN, con più strati (*layer*) di percettroni collegati tra loro e con connessioni ricorrenti, e algoritmi di apprendimento sempre più complessi. Un approccio molto efficiente è quello di evolvere una o più parti di una rete neurale tramite gli algoritmi genetici: i pesi sinaptici, il numero e il tipo di neuroni e la regola di apprendimento, possono tutti essere ottimizzati automaticamente.

2.2 La Robotica Evolutiva

La robotica evolutiva (da qui in avanti semplicemente RE), un campo relativamente recente dell'IA, utilizza un insieme di tecniche e modelli per progettare agenti (simulati o reali) che **agiscano in modo intelligente** secondo i principi dell'evoluzione darwiniana. Per farlo trae continua ispirazione dalle scienze naturali, facendo uso, oltre che della teoria dei sistemi dinamici e dell'ingegneria bio-mimetica, di modelli ispirati alla cognizione umana e animale (reti neurali) e all'evoluzione biologica (algoritmi genetici). L'accento è sulla reattività e sulle capacità di adattamento e apprendimento dell'agente, piuttosto che sull'indagine dei meccanismi profondi della cognizione.

Uno dei principi fondanti la RE è che, affinché un agente possa sviluppare un comportamento intelligente, debba essere inserito in un **ambiente** con cui deve poter interagire tramite il suo **corpo**. Il sistema di controllo dell'agente deve essere in grado di sfruttare le capacità del corpo in modo da ottenere i risultati migliori e aumentare la *fitness*,

Questi due concetti, chiamati *situatedness* ed *embodiment* e accennati precedentemente, sono in netto contrasto con l'IA classica, che tenta di riprodurre in modo logico-formale, e in generale algoritmico, le caratteristiche salienti dell'agire razionalmente (quali ad esempio giocare una partita a scacchi, o raggruppare una collezione di oggetti per una certa caratteristica).

Un tipico esperimento di RE è modellato sullo schema di un algoritmo genetico

(Nolfi e Floreano, 2000):

1. Viene creata casualmente una popolazione iniziale di cromosomi artificiali differenti, ciascuno che codifica il sistema di controllo (ed eventualmente anche aspetti morfologici) di un robot. Questa popolazione viene messa nell'ambiente (fisico o simulato).

2. Ogni robot agirà sulla base delle istruzioni codificate geneticamente e a seconda di ciò che incontrerà nell'ambiente. Alla fine del suo "ciclo vitale", detto in gergo *trial*, verrà dato un punteggio alla sua idoneità in quel particolare ambiente, secondo una funzione di fitness predefinita.

3. Gli individui con le performance migliori vengono fatti riprodurre, e i nuovi individui generati vanno a sostituire gli individui con le performance peggiori, e si potrà cominciare una nuova generazione.

Il numero di generazioni può essere fissato dallo sperimentatore oppure condizionato al raggiungimento di un certo valore di fitness da parte di almeno un individuo.

Facendo evolvere il sistema di controllo (ad esempio i pesi sinaptici della rete neurale che controlla il robot) o lo stesso corpo del robot, si possono ottenere brillanti ed efficienti soluzioni analiticamente troppo difficili (per uno sperimentatore umano) da trovare. Inoltre, sempre da un punto di vista ingegneristico, lo sperimentatore non deve preoccuparsi di dividere il comportamento dell'agente in semplici azioni atomiche, come nella robotica *behaviour-based*, ma deve solo preoccuparsi di organizzare un ambiente e una funzione di fitness adeguati per far emergere il comportamento desiderato[4]. Dopo aver avviato la simulazione deve solo aspettare un numero di generazioni sufficiente a far sviluppare un insieme minimo di comportamenti che risolvano il *task*.

Allo stesso tempo, aumentando la complessità dell'ambiente e dell'agente, si possono studiare fenomeni biologici ed etologici in un contesto controllato dallo sperimentatore,

[4]Ciò non toglie che la stessa funzione di fitness possa essere evoluta, visto che gli AG possono essere usati come generici ottimizzatori di funzioni

ad esempio testando ipotesi che, nel mondo reale, sono troppo ardue da verificare, come l'evoluzione degli agenti (e delle loro capacità adattive) che in natura richiede un tempo enorme anche per organismi molto semplici.

Negli ultimi anni molti hanno cercato di creare sistemi di visione attiva artificiale utilizzando le tecniche della RE, ad esempio (Floreano *et al.*, 2004) co-evolvendo la visione attiva e l'estrazione di *feature* dall'ambiente per tre tipi di compito: riconoscimento di forme, guida di automobili (simulate) e navigazione di robot nell'ambiente. I risultati mostrano che è possibile risolvere problemi complessi di visione artificiale utilizzando architetture molto semplici e a basso costo computazionale, a patto di evolvere l'agente nell'ambiente in cui dovrà operare, così che possa sviluppare l'insieme di abilità necessario al successo (quali il riconoscimento di parti dell'immagine e l'insieme di comportamenti atti a mantenere tali parti nel proprio campo visivo).

Capitolo 3

Implementazione su Evorobot*

3.1 Il software

Per realizzare il progetto di questo tirocinio è stato utilizzato il software Evorobot*, nato nel Laral[1] ad opera di Stefano Nolfi[2] e Onofrio Gigliotta[3].

Lo sviluppo di questo software è stato guidato dalla necessità di avere una piattaforma unica per l'implementazione di esperimenti di robotica evolutiva che gestisse tutte le fasi necessarie: la creazione dei robot e del loro ambiente simulato, la funzione di fitness con la quale valutarli, i parametri e il tipo di algoritmo genetico con cui evolverne il sistema di controllo, cioè la rete neurale. Lo stretto legame con la robotica fisica, oltre che simulata, è evidente ad esempio nella funzionalità di trasferimento del sistema di controllo (la rete neurale evoluta), tramite tecnologia *bluetooth*, dai robot simulati ai robot reali, nonché nell'ampio set di funzioni e parametri per gestire al meglio i robot *e-puck* e *Kepera*, molto usati in questo campo.

Come vedremo nella sezione successiva, **è stato necessario estendere Evorobot* (da qui ER) affinché supportasse un sistema di visione attiva**[4].

[1] Laboratorio di robotica autonoma e vita artificiale, dell'Istituto di Scienze e Tecnologie della Cognizione del CNR: http://laral.istc.cnr.it

[2] http://laral.istc.cnr.it/nolfi

[3] http://laral.istc.cnr.it/gigliotta

[4] Prima delle modifiche portate avanti in questo tirocinio, ER veniva usato quasi esclusivamente per

ER è sviluppato completamente in C/C++, è multipiattaforma e utilizza come librerie grafiche le Qt4. L'interfaccia grafica è molto intuitiva e semplice da utilizzare anche per chi non ha necessità di modificare il codice.

Grande valore infatti è stato dato alla possibilità di impostare gran parte del setup sperimentale tramite file di testo e un'interfaccia grafica completa, con la quale è possibile modificare i parametri del programma, la struttura della rete neurale, l'agente e l'ambiente in cui si muove, anche a tempo di esecuzione. La struttura lineare del codice permette di definire gli altri aspetti, quali ad esempio una nuova funzione di fitness o degli strumenti grafici specifici, con facilità.

Avviando l'evoluzione, compare un grafico con il valore di fitness del miglior individuo di ogni generazione, e con il valore medio di fitness di ogni generazione. L'evoluzione crea dei file contenenti il genoma di tutti gli individui, e altri file contenenti il genoma dei migliori individui. È possibile interrompere in ogni momento l'evoluzione: in tal caso i genomi della generazione corrente verranno salvati su file di testo, in modo che al successivo avvio l'evoluzione riparta da dove si era interrotta.

È possibile poi caricare il genoma di alcuni individui e testarlo, oppure testarli tutti. Questo crea dei file di statistiche che mostrano le performance del miglior individuo di ogni generazione. Durante il test di un singolo individuo si può vedere l'agente che si muove nell'ambiente e i valori numerici dei suoi neuroni, nonchè la rete neurale. Durante il test di tutti gli individui compare un grafico simile a quello dell'evoluzione.

3.1.1 Algoritmo genetico e rete neurale

Le parti più importanti di ER sono le funzioni per l'utilizzo di due tipi di algoritmo genetico, quello classico e quello steady-state, e del sistema di gestione per reti neurali artificiali, piuttosto flessibile, che permette (anche da interfaccia grafica) di impostare

simulazione di robot fisici (da soli o in gruppi anche ampi), evoluti per risolvere diversi tipi di compiti in ambienti più o meno ricchi. Rispetto a questo tipo di simulazioni, il supporto ad un occhio artificiale che esplori diverse immagini ha richiesto un certo numero di modifiche, analizzate nel dettaglio più avanti in questo capitolo.

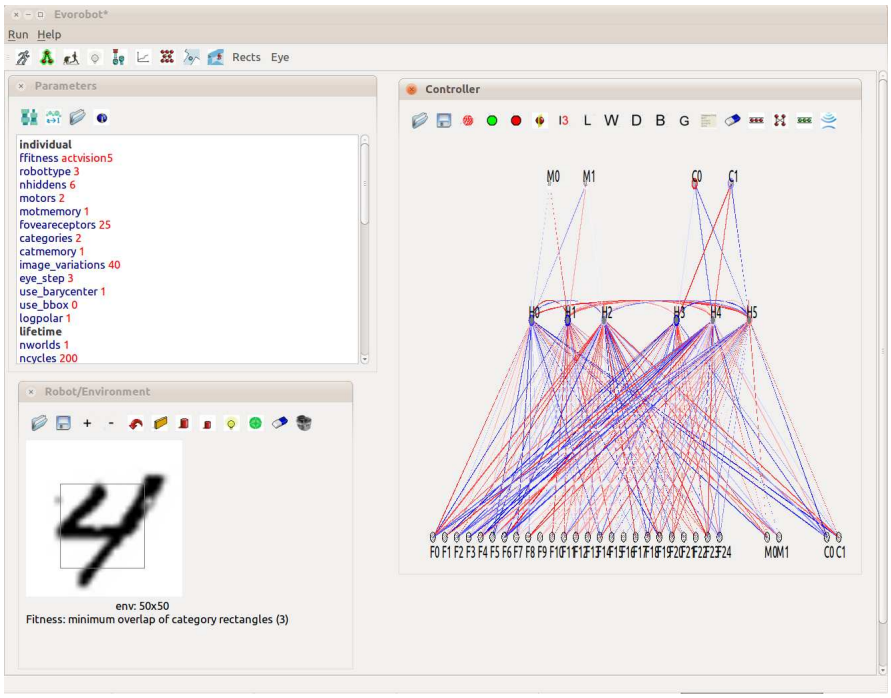

Figura 3.1: Tre *dialog* di modifica

neuroni di input, output e dello strato interno, ciascuno con bias, *delta* e *gain factor* (parametri che permettono di implementare modelli di reti neurali più potenti e complessi delle reti di percettroni).

Dell'algoritmo genetico classico, come descritto da Holland (1975), abbiamo già parlato nel capitolo 2. Delle molte versioni di AG che esistono, gli autori hanno inoltre deciso di implementare il cosiddetto AG *steady-state*.

3.2 Estensioni per la visione

Sebbene l'obiettivo degli autori di ER sia sempre stato quello di creare un framework il più generale possibile per esperimenti nell'ambito della robotica evolutiva (e non solo), mancava il supporto alla visione e in particolare a quella attiva. Per questo motivo, durante il tirocinio sono stati sviluppati una serie di strumenti che servissero a questo scopo, integrandosi perfettamente nello stile di ER.

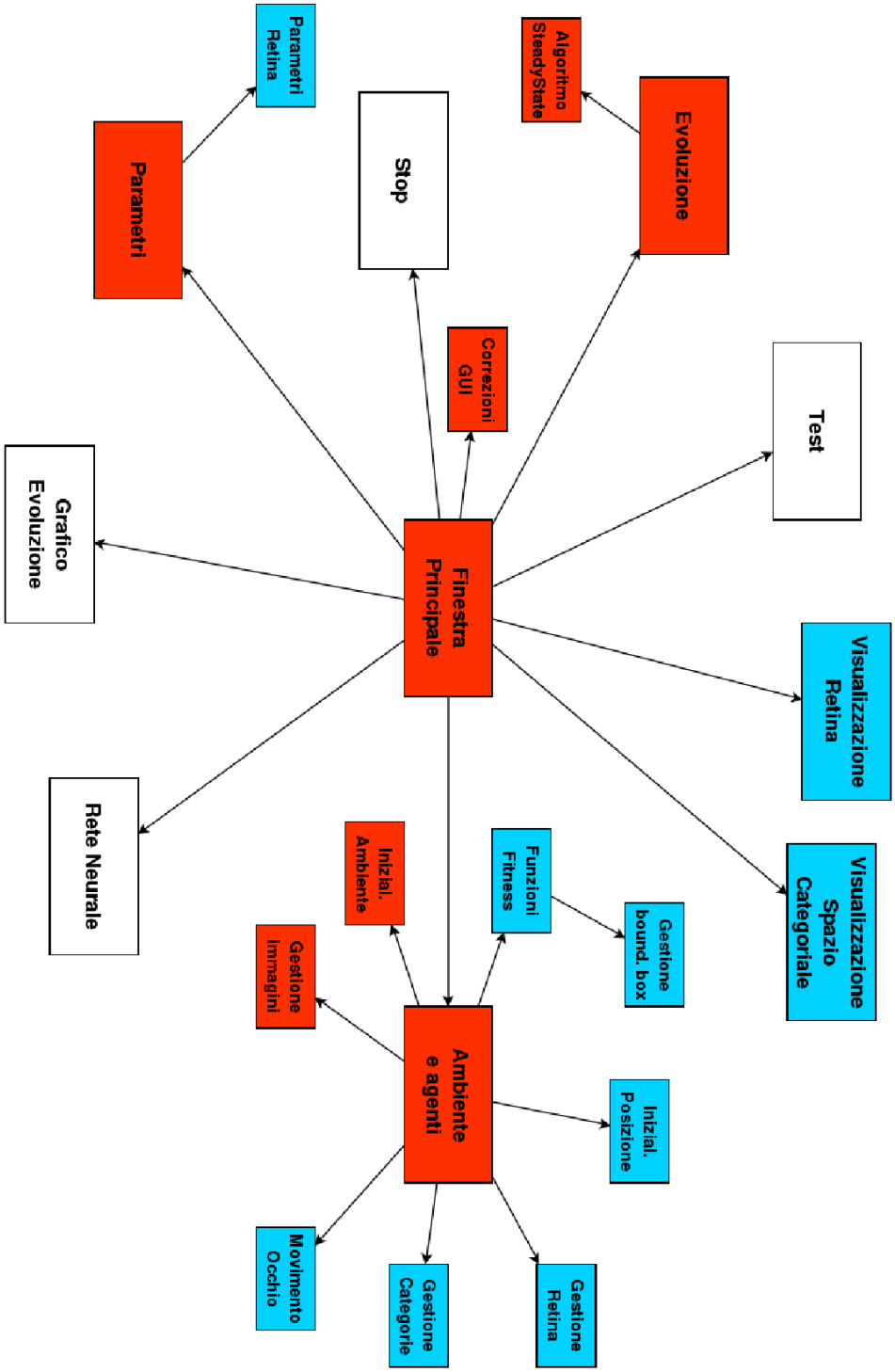

Figura 3.2: Visualizzazione grafica delle modifiche effettuate. I rettangoli grandi sono parte dell'interfaccia grafica, quelli piccoli rappresentano funzioni o gruppi di funzioni. Il colore celeste significa che quel modulo è stato scritto da zero, mentre rosso che è un'estensione di funzioni o classi già presenti.

Algorithm 1: Algoritmo genetico Steady-State

```
1  P = popolazione random iniziale di N individui
2  G = numero di generazioni
3  g = 0
4  while g < G do
5      foreach p ∈ P do
6          valutiamo p e salviamone la fitness
7          if g > 0 then
8              figlio = generaFiglioDi(p)
9              valutiamo figlio e salviamone la fitness
10             worst = prendiPeggiorIndividuo(P)
11             if fit(worst) < fit(figlio) then
12                 sostituiamo figlio a worst in P
13             end
14         end
15     end
16     g = g + 1
17 end
18 return P
```

In particolare è stato sviluppato il supporto a un occhio artificiale libero di muoversi su un piano bidimensionale, senza capacità di zoom e di inclinazione. Sono stati implementati molti parametri per poter variare, ad esempio, il tipo di occhio, il numero di neuroni e l'algoritmo genetico, e alcuni strumenti grafici per visualizzare gli input della retina artificiale e lo sviluppo delle corrette categorie.

Sono stati anche creati dei parametri per adattare il numero dei *trial* al particolare setup sperimentale, secondo questa formula:

$$n\_trial = trial\_per\_immagine * n\_immagini$$

dove il numero di trial per immagine è specificato come parametro nel file di configurazione e rappresenta quante volte una singola immagine fisica deve essere ripresentata

all'individuo[5], e il numero di immagini è anch'esso noto a priori.

Di seguito **sono elencate le modifiche e le aggiunte** ad ER fatte durante questo tirocinio.

3.2.1 Retina artificiale

L'agente è un occhio, come detto precedentemente, con limitate capacità sia di movimento che di risoluzione. L'intento è stato però quello di imitare, in modo estremamente stilizzato, una retina biologica. Per fare questo si è proceduto in modo incrementale.

È stato sviluppato per primo il supporto a una retina con un piccolo campo visivo, composta da 25 fotorecettori sistemati in un quadrato di 5x5 pixel, che si sovrappone all'immagine corrente. Ogni fotorecettore (numerati da F0 a F24, vedi figura 3.1) costituisce parte dello strato di input della rete neurale, ed assume un valore reale compreso tra 0 e 1 che corrisponde al valore di grigio nel pixel corrispondente.

Durante il test di un agente è possibile vedere la retina ingrandita, aggiornata in ogni istante di tempo. Questa retina ha chiaramente un campo visivo molto ridotto, problema risolvibile aumentando il numero di fotorecettori.

Avendo deciso di mantenere fisso a 25 il numero di fotorecettori totali, è stato quindi sviluppato il supporto a una retina con campo visivo più ampio e risoluzione più bassa; per fare questo si è introdotto il parametro di *zoom*. Se lo zoom è uguale a 1, abbiamo una retina 5x5 pixel, dove ogni fotorecettore prende il valore di grigio di un pixel. Se lo zoom è uguale a 2, la retina è un quadrato di 10x10 pixel, dove ogni fotorecettore prende il valore medio dei valori di grigio di un quadrato di 2x2 pixel, come mostrato in figura 3.4.

Più in generale la retina è un quadrato di (zoom x 5)x(zoom x 5) pixel, dove ogni fotorecettore prende il valore medio dei valori di grigio di un quadrato di (zoom x zoom)

[5]Supponendo di avere due categorie, "0" e "1", ciascuna con due variazioni, ad esempio immagini da "10x10" pixel e "20x20" pixel, e di ripresentare queste 4 immagini fisiche 2 volte ciascuna ad un individuo, avremo un numero totale di trial per individuo pari a 8.

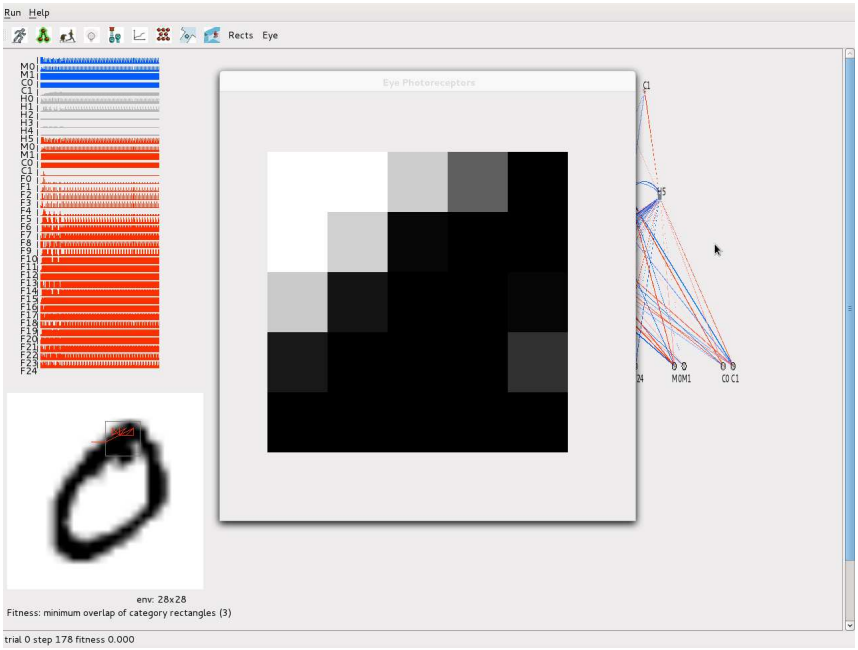

Figura 3.3: Il primo tipo di retina utilizzata, ingrandita durante il test.

pixel.

Infine, per avvicinarsi di più al modello biologico, è stato implementato il supporto a una retina di tipo *logpolare*, dove la zona centrale, come nella **fovea biologica**, ha una risoluzione maggiore, e andando in periferia la risoluzione diminuisce e il campo visivo aumenta esponenzialmente. Per motivi di efficienza computazionale è stato semplificato questo modello dividendo la retina in tre zone: la più centrale è un quadrato di 3x3 pixel, dove ciascun pixel corrisponde a un fotorecettore; la seconda zona è composta da otto quadrati di 3x3 pixel, dove ciascun quadrato corrisponde a un fotorecettore (media dei valori di grigio dei 9 pixel interni); e la terza zona è composta da otto quadrati da 9x9 pixel, dove ciascun quadrato corrisponde a un fotorecettore (media dei valori di grigio dei 18 pixel interni). Un esempio è visibile in figura 3.5.

Questa rappresentazione ha il vantaggio di avere un campo visivo di 27x27 pixel, dove con la rappresentazione precedente non sarebbe bastato uno zoom di 5 per ottenerlo, comunque perdendo molto in capacità di discriminazione.

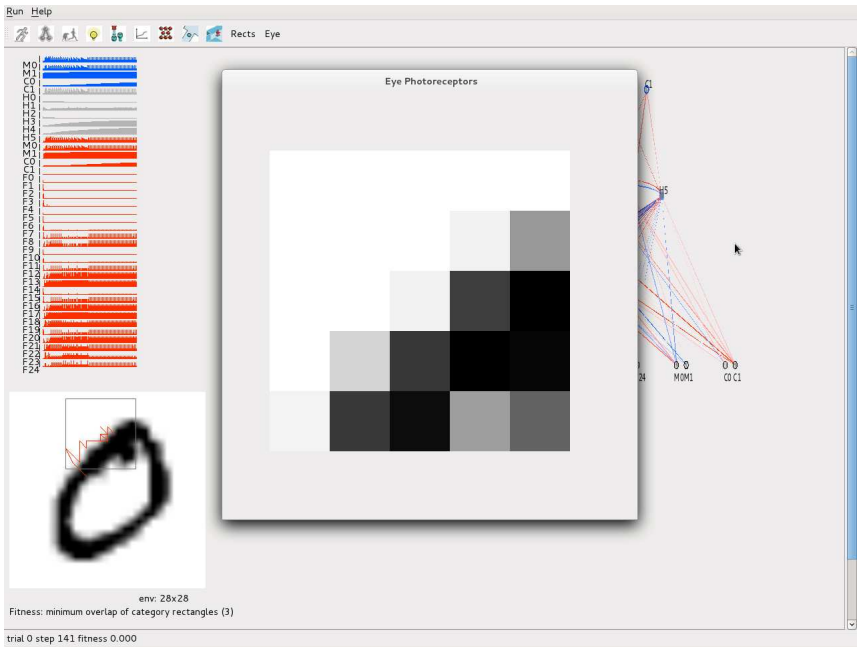

Figura 3.4: Il secondo tipo di retina, dove il parametro zoom è impostato a 2.

3.2.2 Categorizzazione e Fitness

Poiché l'obiettivo finale di ogni agente, sul quale viene valutato, è la sua capacità di categorizzare le immagini che gli vengono passate durante l'evoluzione (capacità che poi viene testata su un differente insieme di immagini), sono state sperimentate varie codifiche per supportare questo comportamento.

3.2.2.1 Bounding Box

La prima ad essere utilizzata verrà chiamata codifica delle "bounding box". Questa codifica è stata introdotta in (**?**) dove era necessario discriminare solo due forme. Il presente lavoro è stato il primo ad utilizzarla per un sistema di visione attiva.

Durante l'evoluzione un agente esplora ogni immagine che rappresenta una categoria tante volte quante sono le immagini per categoria moltiplicato i trial per immagine, come detto precedentemente. Alla fine di ogni trial, vengono salvati i valori di output dei due neuroni categoriali (C0 e C1, vedi figura 3.1). Questo può essere fatto per

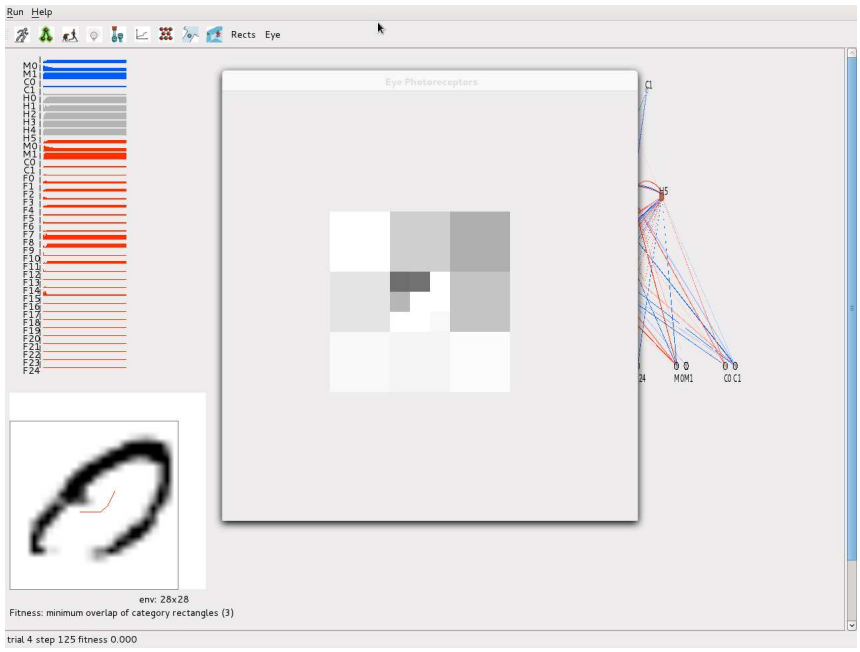

Figura 3.5: Una retina pseudo-logpolare.

l'ultimo istante di vita solamente, o per un numero arbitrario di essi. In ogni caso, ogni singola coppia (C0, C1) viene "segnata" come un punto su uno spazio cartesiano i cui assi hanno valore minimo 0 e valore massimo 1 ciascuno.

Risulta evidente che, una volta finiti i trial corrispondenti ad immagini di una particolare categoria, si può creare un rettangolo che contenga tutti i punti salvati. Questo rettangolo (bounding box) avrà diversi gradi di sovrapposizione con i rettangoli delle altre categorie. Tramite questa sovrapposizione è possibile definire una funzione di fitness degli individui.

Se, alla fine di tutti i trial per un individuo, abbiamo costruito una bounding box per ogni categoria, possiamo calcolare per ogni rettangolo l'area totale di intersezione con gli altri rettangoli.

Dividendo l'area di intersezione per l'area minima tra i due rettangoli intersecanti, otteniamo un numero che va da 0 (succede solo se l'intersezione è nulla) a 1 (vuol dire che il rettangolo più piccolo è completamente dentro l'altro), e che costituirà la nostra

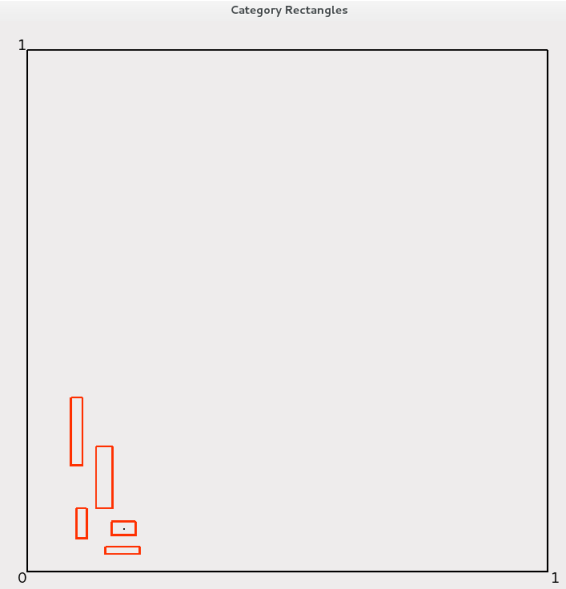

Figura 3.6: Esempio di categorizzazione “Bounding Box”

Algorithm 2: Funzione di fitness per la codifica “Bounding Box”

1 R = insieme delle bounding box
2 $i = 0$
3 **foreach** $r_1, r_2 \in R$ **do**
4 $\quad f_i = intersezione(r_1, r_2) / \min(area_{r_1}, area_{r_2})$
5 $\quad i = i + 1$
6 **end**
7 $Fitness = 1 - ((\sum_i f_i)/i)$
8 **return** Fitness

fitness intermedia.

La somma di tutte le fitness intermedie divisa per il numero di queste ultime è un **valore da 0 a 1, dove 1 vuol dire che tutti i rettangoli coincidono, e 0 che nessun rettangolo tocca gli altri**. Con l’ultima istruzione dell’algoritmo si inverte questa rappresentazione, quindi **la fitness massima diventa 1**.

3.2.2.2 Nearest Neighbors

Un'altra codifica, che prende ispirazione dall'algoritmo di clustering omonimo, utilizza i singoli punti piuttosto che calcolare le bounding box per ogni categoria. In questo caso si è utilizzato un solo punto per ogni immagine fisica, cioè i valori di uscita dei neuroni C0 e C1 nell'ultimo istante di vita.

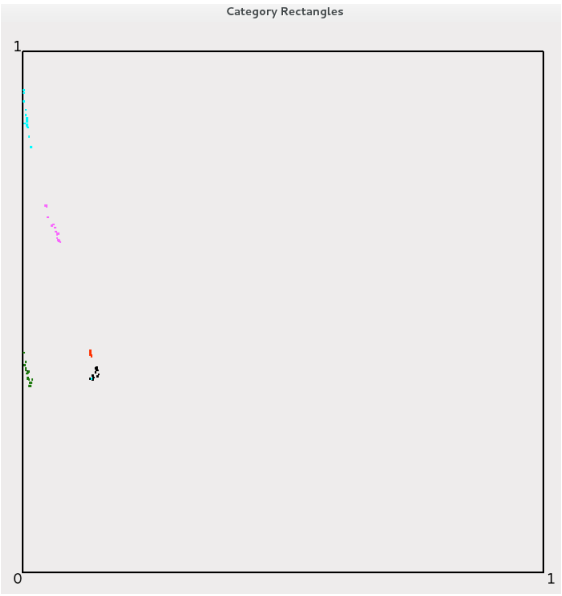

Figura 3.7: Esempio di categorizzazione "Nearest Neighbors"

Se i punti sono raggruppati per categoria in modo non ambiguo, l'individuo ha sviluppato una buona categorizzazione delle immagini presentategli. È facile sviluppare a questo punto una funzione di fitness adeguata.

Sapendo a priori quante immagini per ogni categoria vi sono (ad esempio N), diamo un valore di fitness tanto più alto quanto più gli N-1 vicini di ciascun punto sono della sua stessa categoria. In questo caso si otterrà un **valore di 1 solo nel caso in cui tutti i punti di una stessa categoria siano molto vicini e allo stesso tempo lontani dai punti delle altre categorie**. Si otterrà un **valore di 0 invece solo quando nessun punto avrà vicino anche solo un punto della stessa categoria**.

Algorithm 3: Funzione di fitness per la codifica Nearest Neighbors

```
1  P = insieme dei punti
2  N = numero di punti per categoria
3  i = 0
4  foreach p ∈ P do
5      calcola gli N-1 vicini di p
6      n = numero dei vicini con la stessa categoria
7      f_i = n/(N − 1)
8      i = i + 1
9  end
10 Fitness = (Σ_i f_i)/ |P|
11 return Fitness
```

3.2.3 Posizionamento iniziale

Molto importante è risultato essere il posizionamento iniziale dell'occhio sull'immagine. Poiché il posizionamento casuale (in un intorno del centro dell'immagine) creava talvolta dei problemi all'agente (ad esempio l'immagine veniva "persa" troppo presto), è stato implementato un posizionamento basato sul "baricentro" dell'immagine, dove peso minimo era dato al colore bianco e peso massimo al colore nero.

Una volta trovato il baricentro, l'occhio viene posizionato un intorno (molto piccolo) di quest'ultimo. Un agente evoluto con questo posizionamento tende a tenere le immagini nel suo campo visivo, senza perderle. In molti casi le performance sono anche migliorate in modo consistente.

Dove non altrimenti specificato, il posizionamento iniziale è tramite un piccolo intorno del baricentro.

Capitolo 4

Risultati Sperimentali

4.1 Obiettivi

L'obiettivo iniziale di questo tirocinio è stato quello di replicare un lavoro precedente, descritto in (Mirolli *et al.*, 2010).

Gli autori hanno implementato un sistema di visione attiva (una rete neurale evoluta con un algoritmo genetico classico) che si è dimostrato capace di categorizzare cinque lettere dell'alfabeto in scrittura corsiva, disegnate in scala di grigio, ciascuna con variazioni delle dimensioni.

Il sistema di categorizzazione è statico: cinque neuroni di output (i neuroni categoriali) sono liberi di attivarsi, e la fitness premia gli individui in cui il neurone categoriale più attivo sia quello corrispondente[1] alla categoria corretta, e nei quali i restanti neuroni categoriali abbiano un'attivazione il più possibile vicina allo zero.

Questa codifica non scala bene all'aumentare delle categorie, perché per ognuna di esse deve essere inserito un nuovo neurone categoriale nella rete, aumentando di molto il costo computazionale. Le due codifiche descritte nel capitolo 3 sono state indagate proprio nella loro capacità di adattarsi a setup con un numero di categorie variabili. Tuttavia, **per motivi di efficienza computazionale non è stato possi-**

[1] Tale corrispondenza 1-a-1 è definita a priori, per ogni esperimento.

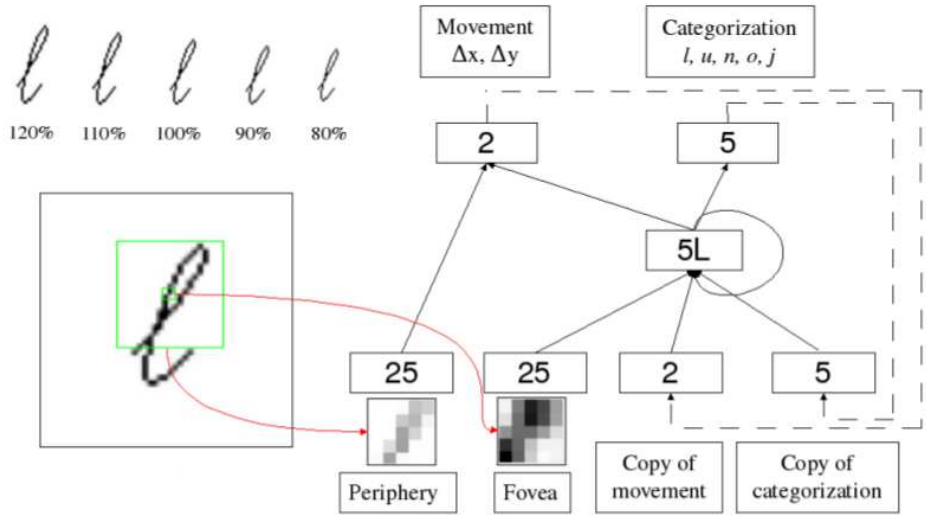

Figura 4.1: Rappresentazione del setup sperimentale, preso dall'articolo citato.

bile sperimentare un aumento consistente di categorie, preferendo concentrarsi sull'aumento del numero di variazioni per ogni categoria.

In generale, le performance ottenute nel lavoro citato sono molto alte. Questo risultato, per il setup con cinque lettere, come vedremo nelle sezioni successive è rimasto inalterato anche con le nuove codifiche.

4.2 Esperimenti effettuati

L'agente utilizzato è costituito da **un solo occhio che può esplorare liberamente delle immagini che gli vengono fornite in sequenza**. Il piano è bidimensionale e l'occhio si muove su di esso, in ogni istante di tempo, di una quantità casuale di pixel sull'asse delle ascisse e delle ordinate (vi è un "passo massimo" impostato a 12 pixel per asse).

Dove non altrimenti specificato, durante questo tirocinio sono state usate RN a tre strati, dove nello strato di input sono presenti i fotorecettori della retina e le copie

efferenti dei neuroni di output; lo strato di input è completamente connesso con lo strato interno, composto da sei *leaky neurons* (neuroni con delta) ciascuno dotato di bias; e lo strato interno è completamente connesso con lo strato di output, composto da due neuroni motori (M0 ed M1, corrispondenti alla variazione sull'asse x e y della posizione della retina) e da due neuroni categoriali (C0 e C1, che rappresentano un punto su uno spazio categoriale bidimensionale).

Molti esperimenti sono stati però effettuati anche con una rete leggermente diversa (cfr. figura 4.2), dove lo strato di input è sempre completamente connesso con lo strato interno; lo strato interno è separato in due gruppi da tre neuroni, dove ciascun gruppo è completamente connesso con se stesso e separato dall'altro gruppo; il gruppo di sinistra è completamente connesso con i due neuroni motori, mentre il gruppo di destra è completamente connesso con i neuroni categoriali. Nel seguito chiameremo "rete connessa" la rete dove non c'è separazione dello strato interno (cfr. figura 3.1), e "rete separata" l'altra.

Non viene utilizzata una regola di apprendimento, bensì **vengono evoluti tramite l'AG i pesi delle connessioni sinaptiche, i bias ed i delta**.

Il risultato atteso è che l'occhio sia capace di categorizzare le immagini passate durante l'evoluzione, e che generalizzi questa capacità con immagini che abbiano lo stesso contenuto semantico (che siano la stessa lettera o lo stesso numero, ad esempio), con delle variazioni morfologiche consistenti. Per avere un ampio insieme di immagini con stesso valore semantico ma anche molto differenti tra loro è stato usato, tra gli altri, il **MNIST Database**, una collezione di migliaia di numeri, da 0 a 9, scritti a mano.

È da notare che **ogni esperimento viene ripetuto dieci volte**, ciascuna con un *seed* differente per il generatore di numeri pseudo-casuali. Nelle sezioni successive si prende in considerazione solo la replicazione che contiene il miglior individuo per quel task.

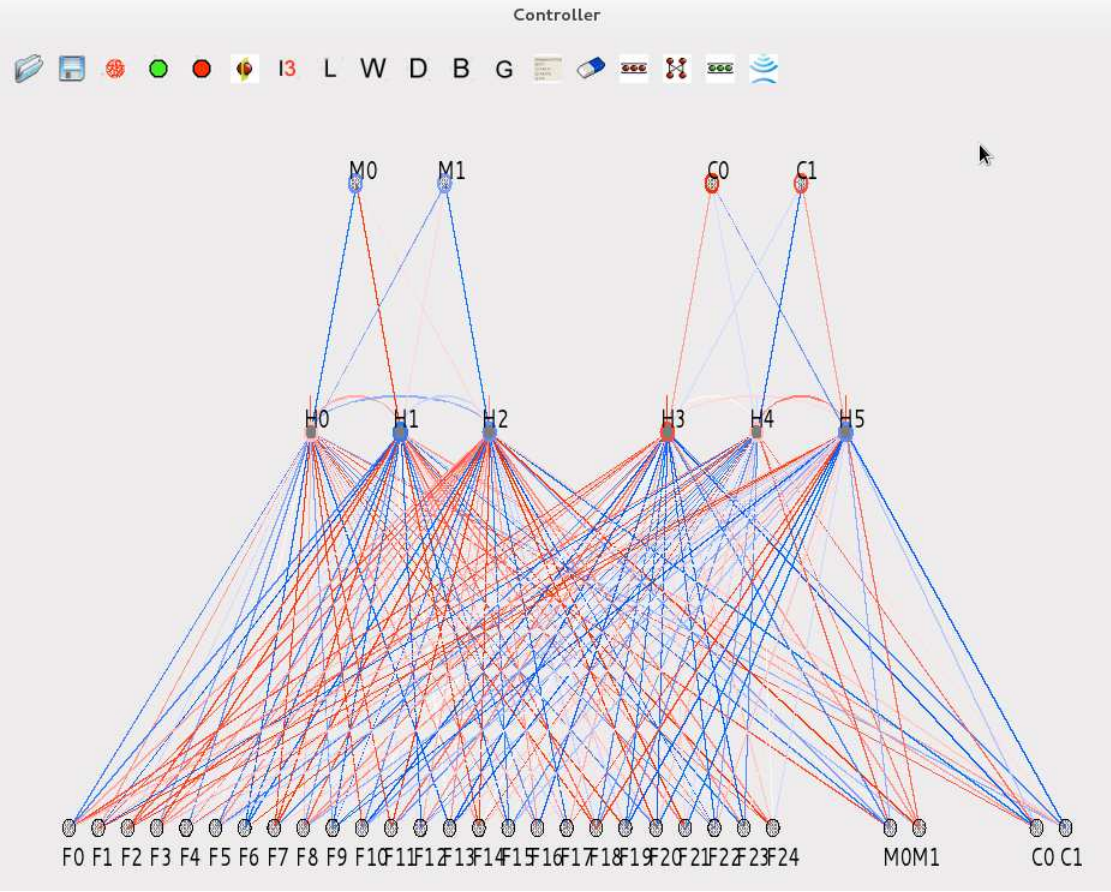

Figura 4.2: Lo strato interno è separato in due gruppi.

4.2.1 Lettere

4.2.1.1 Cinque lettere

Sono state utilizzate cinque lettere dell'alfabeto: l, u, n, o, j. Ciascuna di esse è disegnata in scrittura corsiva e in scala di grigi, inscritta in un'immagine con sfondo bianco di 100x100 pixel. Per ogni lettera vi sono cinque diverse variazioni di grandezza: 80%, 90%, 100%, 110% e 120%, dove ciò che varia è la lettera all'interno della "cornice" di 100x100 pixel. Ciascuna variazione viene ripresentata quattro volte all'agente.

Sono stati provati entrambi i tipi di retina: zoom e log-polare. In particolare lo "zoom" è stato impostato a tre e a cinque. Queste tre retine sono state provate con

entrambi gli algoritmi genetici, classico e steady-state. La prima codifica usata è stata la "Bounding Box" (cfr. capitolo 3 per i dettagli).

Figura 4.3: Le cinque immagini utilizzate

La colonna "FitEvo" rappresenta il test dell'individuo che, nell'ultima generazione dell'evoluzione, ha ottenuto il valore di fitness maggiore durante un test con le stesse immagini dell'evoluzione.

Le performance (come detto nel capitolo 3) sono sempre normalizzate tra 0 e 1, che rappresenta il successo completo.

Retina	Fitness	AG	RN	FitEvo
LogPolar	BoundingBox	Classico	Connessa	1.000
LogPolar	BoundingBox	Steady	Connessa	1.000
Zoom 3	BoundingBox	Classico	Connessa	1.000
Zoom 3	BoundingBox	Steady	Connessa	1.000
Zoom 5	BoundingBox	Classico	Connessa	1.000
Zoom 5	BoundingBox	Steady	Connessa	1.000

È stato poi fatto un tentativo con la codifica "Nearest Neighbors", entrambi gli AG classico e steady-state e la retina log-polare.

Retina	Fitness	AG	RN	FitEvo
LogPolar	NearestNeighbors	Classico	Connessa	1.000
LogPolar	NearestNeighbors	Steady	Connessa	0.999

Analisi Questo setup dimostra che il tipo di retina e le codifiche di categorizzazione usate sono in grado risolvere il compito richiesto in modo paragonabile a quanto descritto nell'articolo di confronto.

Figura 4.4: Boxplot di tutti gli individui delle ultime 50 generazioni di evoluzione per la configurazione: "LogPolar, NearestNeighbors, Classico, Connessa".

Un comportamento evidente degli individui testati è che essi tendono a cercare, qualunque sia il loro punto di partenza, una zona specifica di ciascuna lettera, anche al variare della grandezza dell'immagine. La categorizzazione quindi avviene quando l'agente riesce a trovare una porzione di immagine per ciascuna lettera che sia completamente visibile nel suo campo visivo e che sia "unica" rispetto alle altre immagi-

ni. Il numero esiguo di variazioni per ciascuna lettera probabilmente velocizza questo processo.

Come ben visibile in figura 4.4, per la migliore tra le configurazioni appena elencate, si hanno ottime prestazioni anche nelle generazioni precedenti l'ultima, con nessun picco drammaticamente negativo (fitness minima 0.8) e una mediana molto vicina al valore 1[2].

Tempo di esecuzione dell'evoluzione L'evoluzione di una configurazione di questo setup, con dieci replicazioni incluse, ha impiegato fino a tre giorni di tempo per concludersi.

4.2.1.2 Undici lettere

Sono state utilizzate undici lettere dell'alfabeto: l, u, n, o, j, r, s, w, h, y. Il setup è del tutto identico al precedente, ma si nota come la difficoltà del task sia aumentata in modo considerevole. Prima mostriamo i risultati con la codifica "Bounding Box".

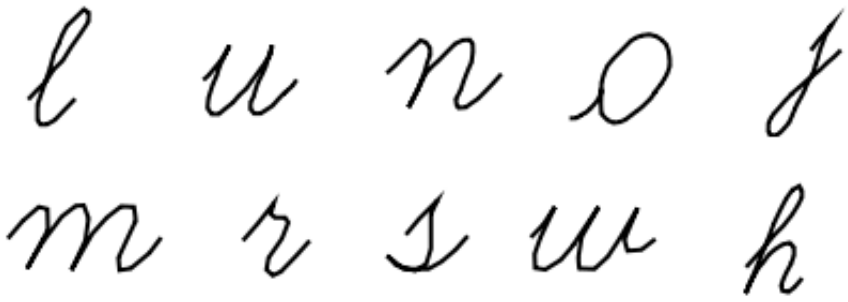

Figura 4.5: Le undici immagini utilizzate

[2]Il grafico mostra dei boxplot, dove sull'asse delle ascisse vi sono le generazioni (tutti i 100 individui di quella generazione) e su quello delle ordinate la fitness. Per ogni "boxplot", vi è un rettangolo il cui lato superiore corrisponde al primo quartile della distribuzione, e la base al terzo quartile. La linea interna al rettangolo è la mediana, mentre le due "braccia" che escono dal lato superiore e dalla base sono rispettivamente il valore massimo ed il valore minimo nella distribuzione.

Retina	Fitness	AG	RN	FitEvo
LogPolar	BoundingBox	Classico	Connessa	0.995
LogPolar	BoundingBox	Steady	Connessa	0.975
Zoom 3	BoundingBox	Classico	Connessa	0.975
Zoom 3	BoundingBox	Steady	Connessa	0.974
Zoom 5	BoundingBox	Classico	Connessa	1.000
Zoom 5	BoundingBox	Steady	Connessa	0.971

Questo invece è il tentativo con "Nearest Neighbors", come per il setup precedente.

Retina	Fitness	AG	RN	FitEvo
LogPolar	NearestNeighbors	Classico	Connessa	0.774
LogPolar	NearestNeighbors	Steady	Connessa	0.717

Analisi All'aumentare del numero di lettere si nota un lieve calo di performance, molto più accentuato con la codifica "Nearest Neighbors". Per la codifica "Bounding Box", dove le prestazioni sono quasi ottimali, l'agente testato **tende ad andare sempre nella stessa direzione**, in basso a sinistra di ciascuna immagine, dove probabilmente la differenza delle lettere è maggiore che altrove.

È interessante infatti notare che anche l'agente testato con la seconda codifica va sempre in una stessa direzione, ma questa volta **in alto a sinistra** di ciascuna immagine, ottenendo prestazioni molto minori. Il perché questa codifica, in questo particolare setup, porti l'agente a bloccarsi in una strategia sub-ottimale, non è stato indagato in questo lavoro.

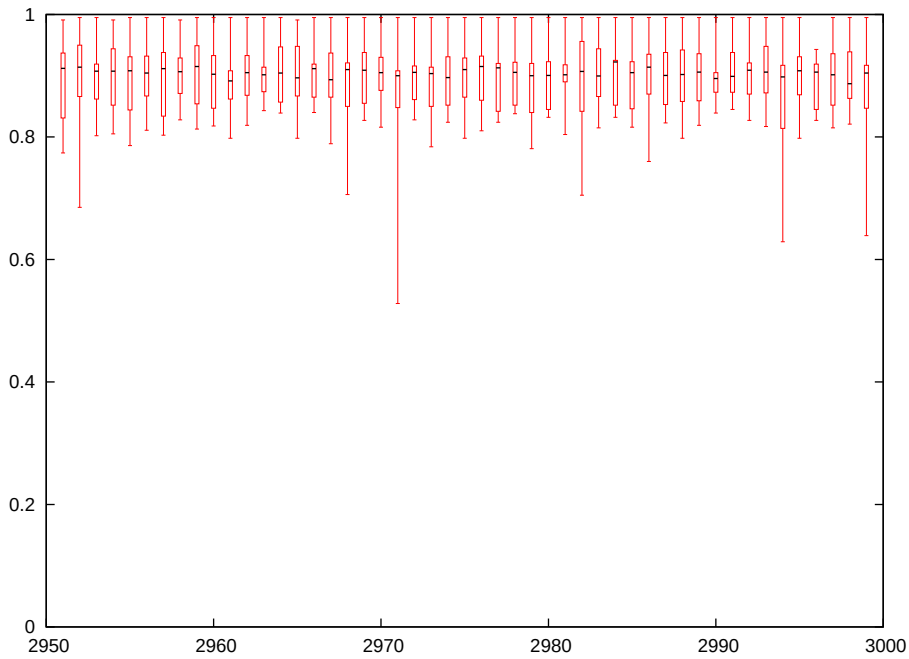

Figura 4.6: Boxplot di tutti gli individui delle ultime 50 generazioni di evoluzione per la configurazione: “LogPolar, BoundingBox, Classico, Connessa”.

Come ben visibile in figura 4.6, si hanno buone prestazioni anche nelle generazioni precedenti l’ultima: nonostante vi sia un picco negativo alla generazione 2971 si nota come vi sia sempre almeno un individuo molto vicino alla fitness 1.

Tempo di esecuzione dell’evoluzione L’evoluzione di una configurazione di questo setup, con dieci replicazioni incluse, ha impiegato fino a sette giorni di tempo per concludersi.

4.2.2 Numeri

4.2.2.1 Cinque numeri, dieci variazioni

Sono stati utilizzati cinque numeri, da 0 a 4, presi dal “MNIST Database”, ampiamente usato in letteratura per compiti di *machine learning*. Ciascun numero è disegnato in scrittura corsiva e in scala di grigi, inscritto in un’immagine con sfondo bianco di 28x28

pixel. Per ogni numero vi sono dieci variazioni della stessa grandezza: le immagini sono talvolta molto differenti tra loro, essendo campioni di scrittura corsiva di molte persone. Ciascuna variazione viene ripresentata tre volte all'agente.

Figura 4.7: I cinque numeri utilizzati nella sezione "Numeri"

Sono stati provati entrambi i tipi di retina: zoom e log-polare. In particolare lo "zoom" è stato impostato a due. Queste due retine sono state provate con entrambi gli algoritmi genetici, classico e steady-state. La codifica usata è stata la "Nearest Neighbors". Sono stati inoltre usati entrambi i tipi di rete neurale (cfr. sezione "Esperimenti effettuati").

La colonna "FitEvo" rappresenta il test dell'individuo che, nell'ultima generazione dell'evoluzione, ha ottenuto il valore di fitness maggiore durante un test con le stesse immagini dell'evoluzione. La colonna "FitTest" invece rappresenta la fitness dell'individuo migliore durante un test usando immagini differenti[3]. Questo vale per tutti i prossimi esperimenti.

[3]Le categorie sono le stesse, ma cambiando immagini si vede se l'agente è stato in grado di generalizzare le categorie apprese o se esse sono legate solamente a caratteristiche delle immagini fisiche che ha esperito durante l'evoluzione

Retina	Fitness	AG	RN	FitEvo	FitTest
LogPolar	NearestNeighbors	Classico	Connessa	0.828	0.594
LogPolar	NearestNeighbors	Classico	Sconnessa	0.854	0.562
LogPolar	NearestNeighbors	Steady	Connessa	0.780	0.598
LogPolar	NearestNeighbors	Steady	Sconnessa	0.799	0.547
Zoom 2	NearestNeighbors	Classico	Connessa	0.788	0.508
Zoom 2	NearestNeighbors	Classico	Sconnessa	0.741	0.535
Zoom 2	NearestNeighbors	Steady	Connessa	0.703	0.578
Zoom 2	NearestNeighbors	Steady	Sconnessa	0.795	0.620

È stata fatta anche una prova, per quattro delle precedenti configurazioni, aumentando la grandezza delle immagini da 28x28 pixel a 50x50 pixel. L'aumento di performance è lieve ma significativo, sia durante l'evoluzione che durante il test (dove è più marcato).

Retina	Fitness	AG	RN	FitEvo	FitTest
LogPolar	NearestNeighbors	Classico	Connessa	0.853	0.671
LogPolar	NearestNeighbors	Classico	Sconnessa	0.838	0.743
LogPolar	NearestNeighbors	Steady	Connessa	0.884	0.652
LogPolar	NearestNeighbors	Steady	Sconnessa	0.851	0.689

Analisi La figura 4.8 mostra come non vi siano grandi variazioni nelle ultime 50 generazioni: tutti gli individui, tranne qualche eccezione, sono "tarati" su una strategia

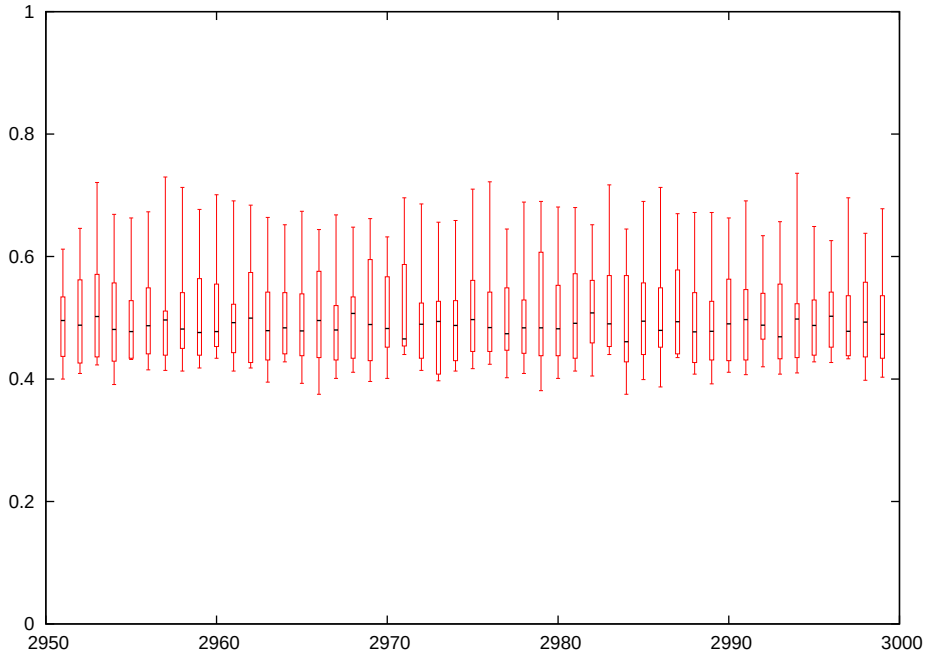

Figura 4.8: Boxplot di tutti gli individui delle ultime 50 generazioni di test per la configurazione: "LogPolar, NearestNeighbors, Classico, Sconnessa" con immagine grande.

sub-ottimale che li fa collocare quasi uniformemente nell'intervallo di fitness tra 0.4 e 0.6.

In entrambe le prove (immagini piccole ed immagini più grandi) l'agente tende a muoversi verso l'alto, con qualche eccezione (ad esempio con il numero due, dove talvolta si fissa a sinistra). Il calo di performance rispetto al task precedente è probabilmente da imputarsi, oltre al fatto che gli agenti si sono probabilmente bloccati su un massimo locale, alla grande diversità delle immagini utilizzate. Per esempio, quelle in figura 4.9 sono due immagini del numero "uno".

Tempo di esecuzione dell'evoluzione L'evoluzione di una configurazione di questo setup, con dieci replicazioni incluse, ha impiegato fino a sette giorni di tempo per concludersi.

Figura 4.9: Due esempi di numero uno, molto diversi tra loro.

4.2.2.2 Cinque numeri, trenta variazioni

Il setup è simile al precedente, con la differenza che sono state utilizzate trenta variazioni per ciascuno dei cinque numeri e che ciascuna variazione è presentata due sole volte all'agente.

Retina	Fitness	AG	RN	FitEvo	FitTest
LogPolar	NearestNeighbors	Classico	Connessa	0.687	0.605
LogPolar	NearestNeighbors	Classico	Sconnessa	0.762	0.658
LogPolar	NearestNeighbors	Steady	Connessa	0.761	0.644
LogPolar	NearestNeighbors	Steady	Sconnessa	0.721	0.616
Zoom 2	NearestNeighbors	Classico	Connessa	0.593	0.552
Zoom 2	NearestNeighbors	Classico	Sconnessa	0.664	0.545
Zoom 2	NearestNeighbors	Steady	Connessa	0.709	0.581
Zoom 2	NearestNeighbors	Steady	Sconnessa	0.652	0.591

È stata fatta anche una prova, per quattro delle precedenti configurazioni, aumentando la grandezza delle immagini da 28x28 pixel a 50x50 pixel. L'aumento di

performance è lieve ma significativo.

Retina	Fitness	AG	RN	FitEvo
LogPolar	NearestNeighbors	Classico	Connessa	0.727
LogPolar	NearestNeighbors	Classico	Sconnessa	0.805
LogPolar	NearestNeighbors	Steady	Connessa	0.781
LogPolar	NearestNeighbors	Steady	Sconnessa	0.734

Analisi In questo task l'agente tende (nella maggior parte dei casi) ad andare verso il basso a destra, e ciò potrebbe spiegare il motivo per cui le performance sono molto più basse del precedente, specialmente quando lo si testa con immagini differenti dall'evoluzione. Probabilmente il numero di generazioni (o di individui per generazione) non è abbastanza elevato da permettergli di trovare la strategia ottimale.

La figura 4.10 infatti ci mostra una situazione simile alla precedente ma con performance massime lievemente peggiori, e performance medie lievemente migliori: lo schiacciamento su un massimo locale, dal quale gli agenti non riescono a distaccarsi, è ancora più marcato, con meno oscillazioni (che siano negative o positive).

Ulteriori indagini con cinque categorie e un maggior numero di variazioni (abbinate a un maggior numero di generazioni) non è stato portato avanti in questo lavoro.

Tempo di esecuzione dell'evoluzione L'evoluzione di una configurazione di questo setup, con dieci replicazioni incluse, ha impiegato fino a undici giorni di tempo per concludersi.

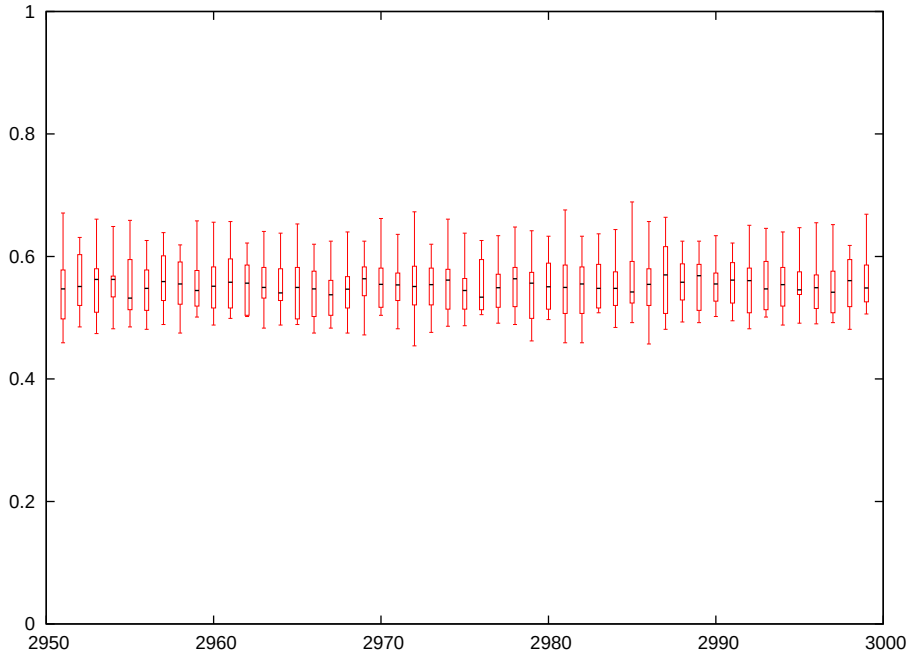

Figura 4.10: Boxplot di tutti gli individui delle ultime 50 generazioni di test per la configurazione: "LogPolar, NearestNeighbors, Classico, Sconnessa" con immagine grande.

4.2.3 Cinque numeri, 1-versus-4

In questo setup sperimentale si è deciso di evolvere cinque differenti reti neurali, ciascuna capace di discriminare rispetto agli altri uno dei cinque numeri dei due setup precedenti. Supponendo di voler categorizzare il numero zero rispetto ai numeri uno, due, tre e quattro, vengono usate 100 variazioni per il numero zero e 25 variazioni per ciascuno degli altri quattro numeri. Ciascuna variazione è presentata all'agente solo una volta.

È stata usata solamente la retina log-polare e la categorizzazione "Nearest Neighbors". L'unico tipo di AG usato è stato quello classico. Sono stati inoltre usati entrambi i tipi di rete neurale.

Numero	Retina	Fitness	AG	RN	FitEvo	FitTest
0	LogPolar	NearestNeighbors	Classico	Connessa	1.000	0.939
0	LogPolar	NearestNeighbors	Classico	Sconnessa	1.000	0.949
1	LogPolar	NearestNeighbors	Classico	Connessa	1.000	0.998
1	LogPolar	NearestNeighbors	Classico	Sconnessa	0.995	0.970
2	LogPolar	NearestNeighbors	Classico	Connessa	0.957	0.874
2	LogPolar	NearestNeighbors	Classico	Sconnessa	0.922	0.882
3	LogPolar	NearestNeighbors	Classico	Connessa	0.980	0.893
3	LogPolar	NearestNeighbors	Classico	Sconnessa	1.000	0.904
4	LogPolar	NearestNeighbors	Classico	Connessa	1.000	0.922
4	LogPolar	NearestNeighbors	Classico	Sconnessa	1.000	0.890

Analisi Il comportamento degli agenti in questo setup è molto diversificato. Mediamente, l'agente tende a seguire per brevi tratti le curve del tratto disegnato, qualora presenti (soprattutto nel caso dello zero), e per i restanti numeri tende ad andare verso l'alto, tranne nel numero due che viene esplorato più spesso andando a sinistra.

Le eccellenti performance di tutte le configurazioni fanno ipotizzare come il numero di categorie (molto basso) e di variazioni durante l'evoluzione (molto alto) possano influenzare notevolmente il risultato.

Ciononostante, è da notare che ciascuna "coppia" di configurazioni, in questo setup, **rappresenta una diversa rete neurale, specializzata per riconoscere un numero rispetto ad altri**. Questo significa che per avere un sistema capace di discriminare tutti e cinque i numeri sarebbero necessarie tutte e cinque le reti e un sistema di

valutazione e di disambiguazione del risultato[4]. Questo è chiaramente di enorme vantaggio rispetto ai setup precedenti, tuttavia ulteriori approfondimenti non sono stati fatti in questo tirocinio.

Tempo di esecuzione dell'evoluzione L'evoluzione di una configurazione di questo setup, con dieci replicazioni incluse, ha impiegato fino a dieci giorni di tempo per concludersi.

4.2.4 Facce

4.2.4.1 Due espressioni, donne

È stata effettuata anche una prova con due sole categorie applicate a dei volti femminili. Sono state usate trentotto immagini molto grandi (600x800 pixel) di differenti volti femminili, di cui metà rappresentano delle persone con espressione felice e sorridente, e l'altra metà le stesse persone con espressione triste o arrabbiata. Ogni immagine è stata ripresentata quattro volte all'agente.

È stata usata solamente la retina log-polare e la categorizzazione "Nearest Neighbors". I tipi di AG usati sono stati quello classico e steady-state. Sono stati inoltre usati entrambi i tipi di rete neurale.

Retina	Fitness	AG	RN	FitEvo	FitTest
LogPolar	NearestNeighbors	Classico	Connessa	1.000	0.831
LogPolar	NearestNeighbors	Classico	Sconnessa	0.984	0.750
LogPolar	NearestNeighbors	Steady	Connessa	1.000	0.719
LogPolar	NearestNeighbors	Steady	Sconnessa	1.000	0.811

[4]Potrebbe ad esempio accadere che due o più reti segnalino un'immagine di test come della "loro" categoria.

Figura 4.11: Le due espressioni facciali femminili: “triste” e felice”

Analisi La figura 4.12 mostra come, nonostante i migliori individui di ogni generazione tendano a distaccarsi dagli altri, questi ultimi sono concentrati in una zona a bassa fitness, poco più di 0.5.

L’agente tende ad esplorare le zone intorno agli occhi, agli zigomi e talvolta al naso e alla bocca, dove nella maggior parte delle immagini vi è una notevole differenza tra espressione felice ed espressione triste. Ad esempio nelle espressioni tristi gli occhi sono più sottili e hanno più ombre intorno, e nelle espressioni felici si vedono più spesso i denti delle persone fotografate, e si formano ombre sotto gli zigomi.

Come una rete neurale così semplice possa far uso di questi indizi, in alcuni casi assolutamente non ovvi, non è stato oggetto di analisi durante questo tirocinio.

Tempo di esecuzione dell’evoluzione L’evoluzione di una configurazione di questo setup, con dieci replicazioni incluse, ha impiegato fino a cinque giorni di tempo per concludersi.

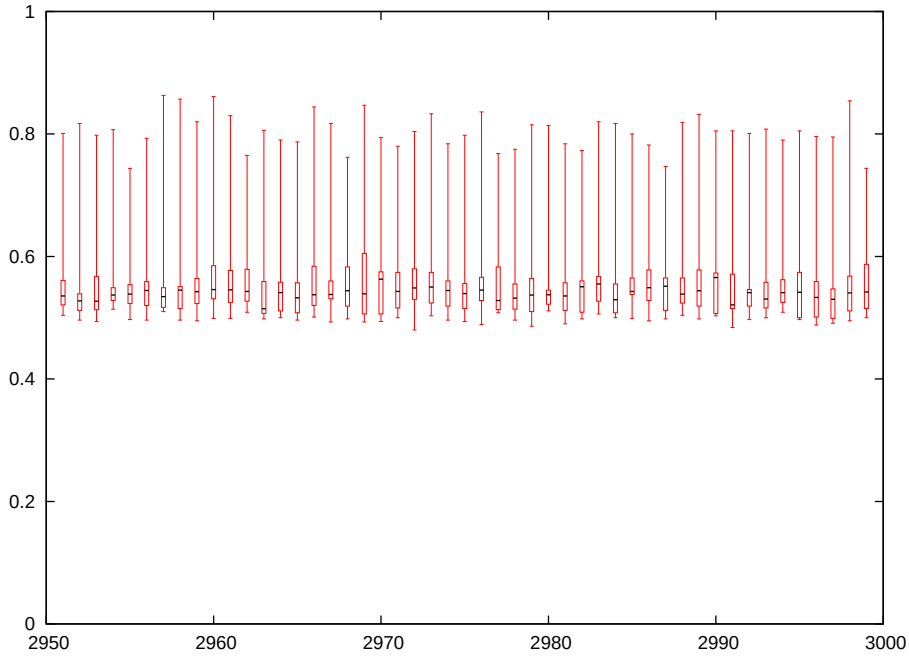

Figura 4.12: Boxplot di tutti gli individui delle ultime 50 generazioni di test per la configurazione: "LogPolar, NearestNeighbors, Classico, Connessa".

4.2.4.2 Due espressioni, donne e uomini

Questo setup è simile al precedente, con la differenza che le immagini sono state rimpicciolite (150x200 pixel), e ne sono state usate 10 femminili e 10 maschili, dove il task è di discriminare, indipendentemente dal sesso del soggetto fotografato, le due categorie: "felice" e "triste".

Retina	Fitness	AG	RN	FitEvo	FitTest
LogPolar	NearestNeighbors	Classico	Connessa	0.939	0.933
LogPolar	NearestNeighbors	Classico	Sconnessa	0.955	0.740
LogPolar	NearestNeighbors	Steady	Connessa	0.992	0.913
LogPolar	NearestNeighbors	Steady	Sconnessa	0.992	0.853

Figura 4.13: Due espressioni facciali maschili: “triste” e felice”

Analisi Il comportamento è molto simile al setup precedente, ma l’esplorazione è più ampia ed articolata (durante uno stesso trial l’agente può passare ad esempio dagli occhi alla bocca, mentre con immagini più grandi l’esplorazione durante un singolo trial tende ad essere limitata a una sola componente del viso[5]).

È importante notare come la figura 4.14 dimostri che il task è risultato essere, per gli agenti, molto più semplice del precedente. Si vede chiaramente che la maggior parte degli individui è situata in un intorno della fitness 0.8, con almeno un individuo a generazione che si avvicina o supera 0.9.

Tempo di esecuzione dell’evoluzione L’evoluzione di una configurazione di questo setup, con dieci replicazioni incluse, ha impiegato fino a tredici giorni di tempo per concludersi.

[5]Infatti il trial viene interrotto dopo un numero prefissato di cicli, che è stato lasciato uguale per le immagini grandi come per quelle piccole.

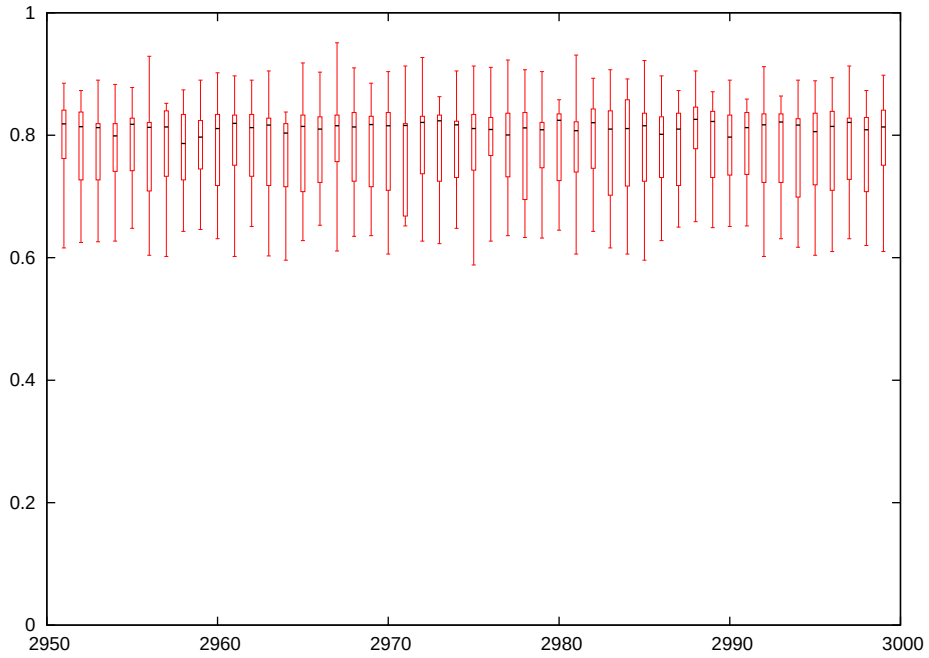

Figura 4.14: Boxplot di tutti gli individui delle ultime 50 generazioni per la configurazione: “LogPolar, NearestNeighbors, Classico, Connessa”.

Conclusioni e sviluppi futuri

Come detto precedentemente, obiettivo di questo lavoro è stato quello di proseguire un percorso iniziato in seno alla visione artificiale: indagare se e come la possibilità di muoversi, e quindi variare "autonomamente" la porzione visibile dell'immagine, potesse migliorare il processo di categorizzazione.

In particolare si è fatto questo nell'ambito della robotica evolutiva, che mira a progettare robot che abbiano il massimo dell'adattabilità usando il minimo delle risorse. La rete neurale utilizzata è stata necessariamente piccola, anche per il costo computazionale che comporta aumentare il numero di neuroni, ed il numero di generazioni limitato a poche migliaia. In alcuni casi **gli esperimenti hanno impiegato fino a diverse settimane per completarsi**: questo ha per forza di cose limitato sia il numero degli esperimenti[6]

Partendo dal presupposto che la **rete neurale dei robot doveva essere piccola e che il numero di neuroni non doveva variare** per tutti gli esperimenti, le performance ottenute sono state mediamente soddisfacenti, quando non ottime. Questo è il caso ad esempio dell'"1-versus-4", dove il miglior agente è stato perfettamente in grado di separare le due categorie (un numero contro gli altri quattro) nonostante il numero di immagini fosse molto più elevato dei task precedenti, e questa categorizzazione è risultata così generale da permettergli di **separare con solamente un lieve calo di performance 200 nuove immagini, talvolta molto differenti da quelle**

[6]Questo è il motivo per cui non tutte le combinazioni di *feature* sono state analizzate.

esperite durante l'evoluzione.

Molto soddisfacenti, nonostante le performance siano peggiori del task "1-versus-4", sono stati poi gli esperimenti con categorizzazione di volti. Con immagini di 600x800 pixel e un campo visivo di meno di 30x30 pixel, senza il movimento dell'occhio sarebbe stato impossibile discriminare tra le diverse espressioni. Per contro, con le immagini rimpicciolite a 150x200 pixel vi è una marcatura sfumatura che rende il compito ancora più complicato. Le foto dei volti non hanno espressioni standardizzate, al contrario sono tavolta esagerate o persino "errate", se si hanno in mente le tipiche espressioni di "felicità" e "tristezza". Il task era notevolmente complesso: le performance ottenute sono quindi particolarmente interessanti.

I tempi ristretti del tirocinio formativo non hanno permesso di esplorare tutte le possibilità inizialmente prese in considerazione. Possibili sviluppi futuri riguardano, ad esempio, un campo visivo più ampio e test in ambienti più complessi, nonché il test su robot fisici. Da sviluppare e migliorare le codifiche per la categorizzazione e le rispettive funzioni di fitness, affinché supportino un numero di categorie più ampio: come è chiaramente visibile nei setup "cinque numeri", quando le categorie sono più di due, all'aumentare del numero di variazioni diminuiscono velocemente le prestazioni.

Infine, vista l'eccellente capacità di separazione con solo due categorie e moltissime variazioni, da indagare ulteriormente è l'utilizzo di diverse reti neurali specializzate su particolari categorie o caratteristiche della scena visiva, evolute singolarmente ma che lavorino in parallelo una volta inserite in un ambiente di test, coordinate da un sistema centralizzato (non necessariamente una rete neurale).

Bibliografia

Aloimonos J.; Weiss I.; Bandyopadhyay A. (1988). Active vision. *International Journal of Computer Vision*, **1**, 333–356.

Bajcsy R. (1988). Active perception. *Proceedings of the IEEE*, **76**(8), 996–1005.

Ballard D. H. (1991). Animate vision. *Artificial Intelligence*, **48**(1), 57–86.

Bear M. F.; Connors B. W.; Paradiso M. A. (2007). *Neuroscienze. Esplorando il cervello.* Elsevier Masson.

Berardi N.; Pizzorusso T. (2006). *Psicobiologia dello sviluppo.* Editori Laterza.

Bruce S. M. (2005). The impact of congenital deafblindness on the struggle to symbolism. *International Journal of Disability, Development and Education*, **52**(3), 233–251.

Canestrari R.; Godino A. (2002). *Introduzione alla Psicologia Generale.* Bruno Mondadori.

Chalmers D. J. (1995). Facing up to the problem of consciousness. *Journal of Consciousness Studies*, pp. 200–219.

Chalmers D. J. (1997). *The Conscious Mind: in search of a fundamental theory.* Oxford University Press.

Dammeyer J. (2011). Mental and behavioral disorders among people with congenital deafblindness. *Research in Developmental Disabilities*, **32**(2), 571–575.

Fischler M. A.; Firschein O. (1987). *Intelligence: The Eye, the Brain and the Computer*. Addison-Wesley.

Floreano D.; Kato T.; Marocco D.; Sauser E. (2004). Coevolution of active vision and feature selection. *Biological cybernetics*, **90**(3), 218–28.

Gibson J. J. (1979). *The ecological approach to visual perception*. Lawrence Erlbaum Associates.

Giovanelli G. (1998). *Prenascere, nascere e rinascere*. Carocci editore.

Holland J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Oxford, England: U Michigan Press.

Isaac AsMirolli M., Ferrauto T. N. S. . C. t. e. a. i. a. a. v. s. C. S. . .-i. (1941). Liar! *Astounding Science-Fiction*.

Mecacci L. (2001). *Manuale di Psicologia Generale*. Giunti Editore.

Mirolli M.; Ferrauto T.; Nolfi S. (2010). Categorisation through evidence accumulation in an active vision system. *Connection Science*, **22**(4), 331–354.

Nolfi S. (2009). *Che cos'è la robotica autonoma*. Carocci.

Nolfi S.; Floreano D. (2000). *Evolutionary Robotics*. The MIT Press.

O'Regan J. K.; Noë A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, **24**, 939–973.

Pfeifer R.; Bongard J. C. (2006). *How the body shapes the way we think*. The MIT Press.

Russell S.; Norvig P. (2005). *Intelligenza Artificiale: un approccio moderno.* Pearson Education Italia.

Čapek K. (1921). *R.U.R.* Echo Library.