FEDERICO FAGGIN

LE RETI NEURONALI*

Vorrei subito mettere a fuoco il problema che stiamo cercando di risolvere. Si tratta di creare delle macchine intelligenti e autonome, cioè delle macchine in grado di ricevere e interpretare dei segnali sensoriali dal mondo reale e produrre un comportamento intelligente, in tempo reale.

Questo problema rappresenta in realtà un insieme di problemi molto difficili, che hanno messo in evidenza le limitazioni del computer tradizionale.

È proprio nel tentativo di trovare risposta a questi quesiti che abbiamo rivolto l'attenzione al sistema nervoso animale e alle reti neuronali artificiali.

Per chiarire che cosa sono le «reti neuronali» e mettere in evidenza la differenza fondamentale con il computer tradizionale, è necessario che prima vi parli del computer.

Il computer è un oggetto recente. Nato circa quarantacinque anni fa, il suo compito originale era quello di fare calcoli numerici molto più rapidamente dell'uomo dotato di carta e matita. Nel corso del suo sviluppo avvenne la scoperta del «programma in memoria» cioè dell'idea di trattare il programma come un qualsiasi altro dato numerico immagazzinato nella memoria: cambia solo l'interpretazione che si dà a quei dati.

Prima di questa idea il programma era realizzato con una serie di connessioni fisiche, dove ciascun filo corrispondeva all'istradamento da dare ai dati che dovevano passare da un organo all'altro del calcolatore. La struttura fisica del calcolatore doveva quindi esser cambiata a seconda del problema che si voleva risolvere: cosa che richiedeva parecchio tempo e pazienza da parte di personale specializzato.

La nuova scoperta porta ad organizzare il calcolatore con una unità centrale di calcolo ed una unità di memoria capace di immagazzinare una certa quantità di dati.

Possiamo immaginare la memoria come un alveare, composto di tante celle uguali. Ogni cella può essere libera od occupata, e

* Conferenza tenuta nell'Auditorium Canneti di Vicenza il 27 settembre 1989.

questa è l'unica cosa che ci interessi di questa cella: il suo «stato».

Se è libero, diciamo che lo stato è «zero». Se è occupata, diciamo che lo stato è «uno». Questo è il «bit» di informazione: la rappresentazione più elementare possibile, che richiede l'abilità di distinguere solo due stati.

Ad ogni cella è associato un numero che la identifica, un indirizzo, e ci sono circuiti che fanno parte della memoria che sono in grado di interrogare lo stato di ciascuna cella – cioè di leggere la memoria – nonché di alterare lo stato di occupazione di ciascuna cella, cioè di scrivere nella memoria.

Possiamo utilizzare un certo numero di questi stati binari, di questi *bit*, per rappresentare qualsiasi tipo di informazione deterministica: un numero, un comando, un indirizzo in memoria: in breve, un qualsiasi *simbolo*. Sta nell'unità centrale del computer il compito di distinguere il tipo di rappresentazione di ciascun gruppo di *bit* e di operare su di essi nel modo opportuno e rigorosamente preordinato.

Gli ordini all'unità centrale li dà il programma. Il programma è completamente indistinguibile dai dati; l'unica differenza è che l'unità centrale sa dove il programma inizia. Da quel punto in poi è compito del programma di mantenere chiara la distinzione, e di evitare che l'unità centrale cerchi istruzioni nel posto sbagliato.

Ogni tanto questo sbaglio succede e il computer dà letteralmente *i numeri*!

L'idea di programma in memoria è estremamente semplice, ma è un'idea potente che scatena e generalizza il calcolo elettronico digitale, rendendo il computer uno strumento di estrema utilità e versatilità.

Con l'idea di programma, il calcolatore si trasforma da macchina per il calcolo sui numeri a computer, cioè ad uno strumento per la manipolazione dei simboli. È interessante notare come questo fatto venne capito solo a metà degli anni cinquanta, cioè dieci anni dopo lo sviluppo iniziale. Si aprì così la strada ad applicazioni nuove che estesero moltissimo l'utilità del computer, trascendendo lo scopo originale che ne aveva motivato lo sviluppo.

Con la nozione di programma nascono il *software*, i linguaggi di programmazione e l'arte di programmare.

Il cuore di un programma è una descrizione precisa e dettagliata di come risolvere un particolare problema: un *algoritmo*, una formula che prescrive ciascun passo.

Per molti anni sembrò che il computer fosse onnipotente, cioè che potesse risolvere ogni problema: bastava solo aver sufficiente memoria e velocità.

Sorse così la scienza dell'intelligenza artificiale, che si propose di far fare al computer dei compiti tipicamente umani, come tradurre da

una lingua ad un'altra oppure interpretare e descrivere una scena.

I mattoni da costruzione del computer furono all'inizio reperiti un po' dappertutto: valvole termoioniche e componenti passivi – resistenze, condensatori e induttanze – per i circuiti logici dell'unità centrale, toroidi di materiale magnetico per la memoria, e così via. Fu solo con il *transistore*, tuttavia, che fu possibile costruire i primi computer commerciali (affidabili anche se molto costosi) nella prima metà degli anni cinquanta.

Con lo sviluppo del circuito integrato, avvenuto agli inizi degli anni sessanta, la tecnologia dei semiconduttori cominciò a sostituire le altre tecnologie eterogenee usate nella costruzione del computer.

Dieci anni dopo, la tecnologia dei semiconduttori avanzò al punto che l'intera unità centrale del computer potè essere integrata in un solo *chip*. Nacque così il *microprocessore*.

Con il microprocessore il computer diventa un pezzetto di silicio che può venir applicato a basso costo, dovunque occorra un briciolo di intelligenza.

Il computer lascia così il laboratorio e il centro di calcolo e si mescola tra di noi; diventa un oggetto di uso comune.

Ma il computer non può far tutto, nemmeno tutto quello che richiede soltanto calcoli.

Il dubbio che il computer sia limitato cominciò a venirci proprio dallo studio dell'intelligenza artificiale. Certe operazioni fondamentali come il riconoscimento di forme, detto *pattern recognition* in inglese, e la memoria associativa che animali e uomini fanno senza sforzo apparente, sono difficilissime da realizzare con il computer.

In quei pochi casi semplici dove riusciamo a farlo, la potenza di calcolo richiesta è enorme.

In altre parole, benché il principio del computer (come fu dimostrato da Turing) sia un principio molto generale e permetta di risolvere qualsiasi problema logico, le risorse di memorie e soprattutto di tempo richiesto per la soluzione possono essere proibitive.

Quindi, in pratica, certi problemi non sono risolvibili in tempo utile. E non è nemmeno questione di aspettare che computers più potenti vengano sviluppati: c'è una differenza di molti ordini di grandezza tra ciò che è necessario e ciò che è ragionevole aspettarsi dalla futura evoluzione del computer nei prossimi dieci anni.

Per questa classe di problemi bisogna cambiare sistema e il cervello animale è l'unico esempio che abbiamo di un sistema capace di risolvere questo tipo di problemi.

È ormai molto chiaro che il principio di funzionamento del cervello è molto diverso da quello usato dal computer.

A questo punto è opportuno fare un esempio dettagliato. Introdur-

rò così la memoria associativa e farò vedere come tale funzione metta in difficoltà il computer.

Come ho già accennato prima, la memoria del computer, che si chiama *RAM (Random Access Memory)*, consiste in tante celle. Ogni cella ha un indirizzo e in ogni cella si trovano dei dati. In questo caso, invece di un solo *bit* immaginiamo che ogni cella contenga un gruppo di *bits*.

Se sappiamo l'indirizzo è molto facile accedere ai dati, oppure cambiare i dati contenuti in quella cella, perché abbiamo creato una corrispondenza precisa tra l'indirizzo e i dati. Ma cosa succede se non sappiamo l'indirizzo, oppure se sappiamo l'indirizzo solo approssimativamente? Siamo nei guai perché non conosciamo i dati e quindi anche se cercassimo dovunque non serve. Quei dati sono persi per sempre.

Questo metodo di memoria funziona in un computer perché c'è un programma – estensione dell'uomo che lo ha fatto – che tiene conto di tutto.

Ma in un cervello non c'è il programma e tanto meno un programmatore; il cervello deve arrangiarsi da solo. Così, nel cervello si accede ai dati usando altri dati. Nel momento in cui si vuole immagazzinare l'informazione, si crea una corrispondenza tra due dati: a X corrisponde Y.

Nel futuro, conoscendo X o anche una parte di X, si può avere accesso a Y: e questa è la *memoria associativa*.

La cosa importante e essenziale è che non occorre conoscere X in maniera assolutamente precisa, ma basta anche sapere un X che si avvicini all'originale, per aver accesso a Y: ed è qui che sta la differenza fondamentale tra i due tipi di memoria.

Possiamo visualizzare una memoria associativa come costituita da due unità. La prima unità contiene un numero di celle, e ciascuna di queste celle contiene il valore di X, i dati di ingresso, nel momento in cui vogliamo creare l'associazione.

Non è importante in quale cella mettiamo i dati, purché la cella sia libera.

La seconda unità è costituita da una memoria tradizionale, una RAM. L'indirizzo della cella dove scriviamo X è disponibile come uscita ed è collegato alla RAM. La cella, nella RAM così indirizzata, viene usata per scrivere Y.

È chiaro che la prima unità della memoria associativa fa esattamente l'operazione inversa di una RAM. Si forniscono dei dati all'ingresso e la memoria produce un indirizzo in uscita, mentre nella RAM ad un indirizzo di ingresso corrispondono dei dati in uscita. Questo tipo di memoria si chiama CAM, le iniziali di *Content Addressable Memory*.

L'operazione interessante succede quando vogliamo avere accesso

all'informazione. In questo caso, all'ingresso della CAM forniamo dei dati che sono vicini, ma non necessariamente identici ai dati immagazzinati nella memoria.

Bisogna quindi calcolare la distanza fra i dati di ingresso e i dati immagazzinati in ciascuna delle celle della CAM, decidere quale delle celle ha i dati più vicini ai dati di ingresso – e questo implica una matrice opportuna – e finalmente fornire, all'uscita, l'indirizzo della cella così individuata.

Questo è esattamente l'indirizzo che individua la cella nella RAM dove si trova Y. E il gioco è fatto.

È chiaro che la CAM non può essere solo una memoria, ma deve anche fare dei calcoli. Infatti la CAM è un *pattern recognizer*, un riconoscitore di forma elementare.

Per avere una idea di quanti calcoli siano necessari, supponiamo che la CAM abbia quattromila celle e che ciascuna cella contenga un dato. Il dato in questo caso è una sequenza ordinata di 64 numeri. Si dice un vettore con 64 componenti reali.

Se vogliamo calcolare la distanza euclidea tra un vettore di ingresso e il vettore contenuto in una di queste celle, dobbiamo fare 64 sottrazioni, 64 moltiplicazioni, 64 addizioni e poi estrarre la radice quadrata. E siccome dobbiamo fare questo per ciascuna delle celle della CAM, dobbiamo fare quattromila volte le operazioni accennate. Se poi vogliamo fare tutto questo in un micro secondo – un milionesimo di secondo – dobbiamo eseguire l'equivalente di circa 180 miliardi (dieci alla nove) di operazioni per ogni secondo!

Questa velocità è di cento volte superiore alla capacità di calcolo del computer più potente che abbiamo.

Se questa è una delle operazioni fondamentali, è chiaro che l'*hardware* deve cambiare: ed una delle prime cose da fare è di organizzare il sistema in maniera diversa. Invece di avere una sola unità centrale di calcolo e memoria separata – come nel computer – e muovere i dati dalla memoria alla unità centrale e viceversa, bisogna operare direttamente, in parallelo, là dove ci sono i dati.

Se si opera sequenzialmente, si perde troppo tempo a muovere i dati avanti e indietro tra unità centrale e memoria, e a fare una operazione per volta. Bisogna quindi, idealmente, avere un processore elementare associato ad ogni dato.

Parliamo quindi di parallelismo massiccio, cioè di sistemi contenenti da qualche milione di processori (fra qualche anno) ad alcuni miliardi di processori fra dieci anni.

La seconda differenza fondamentale riguarda il modo di operare. Il computer richiede un programma, mentre il circuito neuronale impara da solo: basta dargli degli esempi.

Vorrei spiegare questa differenza in dettaglio. Consideriamo il problema di riconoscere una faccia, cioè di associare ad una immagine visiva di ingresso un nome, in uscita. Anche questo è un problema di memoria associativa; però qui le variazioni ammissibili nei dati di ingresso sono enormi.

Bisogna quindi, prima di usare il tipo di memoria associativa che ho discusso prima, ridurre di molto la variabilità nei dati. Occorre cioè estrarre quegli invarianti che possano descrivere in maniera compatta le caratteristiche salienti della faccia. Dobbiamo quindi operare una compressione sui dati.

Per fare questo bisogna trasformare successivamente l'immagine originale, creando una serie di immagini sempre più astratte, contenenti solo dati salienti. Qualcosa di simile a quello che fa un caricaturista, che con pochi tratti di penna riesce a comunicare una immagine riconoscibile.

Per esempio: per prima cosa, dobbiamo eliminare l'effetto dell'illuminazione sull'oggetto: una faccia è la stessa sia sotto il sole di mezzogiorno che al lume di candela.

Questa operazione, nell'occhio umano, è fatta nel primo strato di sensori che collegano i fotoricettori tra di loro: le cellule orizzontali.

Dobbiamo poi rimuovere l'effetto di scala e di rotazione: una faccia è la stessa sia vicina come lontana, e anche se è un po' inclinata.

È poi da eliminare l'effetto del movimento, l'effetto tridimensionale, l'espressione della faccia, e così via: e per fare questo dobbiamo eseguire una serie di operazioni simili a quelle descritte prima per la CAM, con enorme dispendio di calcolo.

Così, finalmente, avremo ridotto la variabilità dei dati al punto che possiamo immagazzinare in una memoria associativa questi tratti astratti: e benché questi tratti corrispondano ad *una sola* posizione della faccia, noi saremo in grado di riconoscere la stessa faccia in una varietà di posizioni espressioni e luce mai prima sperimentata.

Questa è l'abilità del cervello a *generalizzare*, cioè a creare una rappresentazione dei dati (a partire da pochi esempi particolari) che ci permette di riconoscere gli stessi oggetti, idee, *pattens*, in condizioni molto più generali.

Ho il sospetto che proprio questa abilità sia alla base anche della creatività. Queste cose, infatti, noi le facciamo senza rendercene conto: e ci sembrano talmente elementari che, all'inizio della ricerca sull'intelligenza artificiale, avevamo completamente sottovalutato il problema: il fatto che il computer non riesca a fare bene queste operazioni ci ha colto di sorpresa.

Per il computer, invece, è molto semplice riconoscere un oggetto quando questo è assolutamente identico ad un modello immagazzinato

in memoria. Per il computer torna facile imparare rigorosamente a memoria qualsiasi cosa; però basta cambiare anche un piccolissimo particolare, qualcosa che noi facciamo perfino fatica a notare, e il computer non riesce più a riconoscere lo stesso oggetto. Il computer non è in grado di «generalizzare».

Ecco quindi che i due sistemi sono diversi e complementari.

La terza differenza fondamentale è l'abilità del cervello di continuare a funzionare anche se un numero elevato dei suoi componenti sono guasti. Chiamiamo questa proprietà *fault tolerance*: tolleranza al guasto.

Il computer non è così: basta anche una minuscola imperfezione perché il tutto cessi di funzionare.

Oggi sappiamo fare dei computers che chiamiamo *fault-tolerant* ma in realtà si tratta di macchine che possono tollerare solo uno o due guasti prima di crollare.

Nel cervello più sono gli elementi che non funzionano e più si degrada il funzionamento della macchina: tutto questo in maniera graduale senza un salto netto tra l'operare ed il non operare.

Questa *gradualità* è una caratteristica che troviamo ovunque nel cervello.

Il mondo del computer è un mondo fatto di contrasti: aperto – chiuso; funziona – non funziona; appartiene ad una categoria – non appartiene ad una categoria, eccetera. Questo determinismo è appropriato al mondo artificiale dove oggi opera il computer ma nel mondo reale popolato da animali e uomini, le categorie non sono così nette, i concetti non sono così matematici, i bordi tra un insieme ed il suo complementare non sono come la lama di rasoio della logica *booleana*.

Per esempio, per noi è possibile considerare che una persona appartenga allo stesso tempo all'insieme dei belli come a quello dei brutti.

Per noi spesso il grado di appartenenza ad un insieme non è una funzione *booleana* (uno o zero, appartiene o non appartiene) ma piuttosto una funzione continua, che ammette tutti i valori in un intervallo che va dalla «certezza assoluta di appartenere» alla «certezza di non appartenere» passando per tutti gli stadi intermedi. Sicché, quando vogliamo che il computer ragioni come noi, lo possiamo fare ma facciamo fare al computer una cosa che non gli viene spontanea. Forziamo la sua natura, per così dire: ed il prezzo è che il computer diventa inefficiente e disadatto al problema.

Ecco perché bisogna cambiare il sistema.

Quasi quattro anni fa ho fondato una società con Carver Mead, un professore di Caltech, dedicata allo sviluppo di reti neuronali. La società si chiama Synaptics ed è situata a San José, nella Silicon Valley.

Insieme ad altri ingegneri e scienziati impiegati della società, stiamo sviluppando una nuova tecnologia per costruire macchine intelligenti e, in futuro, autonome. Questa tecnologia è ispirata ai principi di funzionamento del sistema nervoso biologico, e usa la stessa tecnologia di fabbricazione che è adoprata per i microprocessori. Con questi metodi, oggi possiamo integrare economicamente fino ad un milione di componenti in un solo *chip*, delle dimensioni di poco più di un centimetro quadrato.

Fra dieci anni sarà possibile fabbricare *chips* da trenta milioni di componenti; e se riusciremo a fare circuiti *fault tolerant* come è nostra intenzione, potremo avere un circuito grande come tutta una fetta di silicio. In questo caso il *chip* sarà gigantesco: venticinque centimetri di diametro e conterrà circa dieci miliardi di componenti.

Durante gli ultimi tre anni ci siamo concentrati a sviluppare una varietà di moduli per il calcolo analogico adattativo, da usare come mattoni per la costruzione di reti neuronali. Abbiamo già dei prototipi, e a questo punto si tratta di applicare questa tecnologia alla soluzione di alcuni problemi pratici esistenti.

Questo sarà il passo successivo che ci impegnerà per i prossimi due anni.

Oggi possiamo già integrare in un singolo *chip* centomila processori analogici elementari, ciascuno con il suo elemento dati.

Ciascuno di questi processori può fare una operazione elementare (per esempio il calcolo di un esponenziale e la somma del risultato con il processore vicino) in un microsecondo. Siccome tutti e centomila funzionano in parallelo, questo chip può fare l'equivalente di cento miliardi di operazioni al secondo (dieci alla undici).

È una scala da capogiro ma non è niente in confronto alla capacità di calcolo e alla complessità del cervello umano. Il cervello umano contiene infatti circa 10 alla 15 sinapsi – il processore elementare del sistema nervoso – distribuita su cento miliardi di neuroni. In media diecimila sinapsi per neurone.

Siccome occorrono da dieci a cento componenti elettronici per emulare la funzione di una sinapsi, ecco che il cervello è equivalente ad un sistema contenente da 10^{16} a 10^{17} componenti!

È un numero straordinario perché, pure assumendo di riuscire nell'anno duemila a fare la *Wafer Scale Integration*, cioè ad integrare, come ho accennato, 10 miliardi di componenti in tutta una fetta di silicio, occorrerebbero da un milione a dieci milioni di fette per ottenere la stessa complessità.

L'altro aspetto sorprendente del cervello è nell'uso parsimonioso di energia per adempiere il suo compito.

Il cervello consuma *grosso modo* 10 Watt di potenza, e fa l'equiva-

lente di almeno 10^{16} operazioni per secondo. Dove una operazione è intesa qui come il calcolo di una funzione non lineare.

Ne risulta che l'energia usata per operazione è di 10^{-15} *joule*.

Questa efficienza energetica va paragonata con l'efficienza del computer di oggi, che è di circa 10^{-6} joule/operazione, cioè un miliardo di volte meno efficiente del cervello.

La tecnologia che stiamo sviluppando alla «Synaptics» è molto più efficiente della tecnologia digitale e permette di raggiungere dei valori di 10^{-11} joule/op. al livello di chip e 10^{-10} joule-op. a livello di sistema: il che vuol dire che, per fare 10^{16} operazioni per secondo anche con la nostra tecnologia avanzata, ci vogliono un milione di Watt di potenza dissipata; e questo assumendo di riuscire a costruire un sistema di equivalente complessità, cosa che è totalmente impossibile da fare, almeno per i prossimi vent'anni.

Fra dieci anni, con la *Wafer Scale Integration*, sarà possibile ridurre il consumo di due ordini di grandezza: e, malgrado questo, occorreranno ancora 10000 watt di potenza per riuscire a fare la stessa quantità di calcoli del cervello umano.

Tutto quanto ho detto finora assume che la funzione fatta dal cervello sia quella emersa recentemente dallo studio della neurobiologia e neuro fisica. Non sarei sorpreso, tuttavia, se scoprissimo in futuro che il cervello è ancora più complesso del modello attuale e quindi i valori che ho dato sono ottimistici.

Che cosa possiamo aspettarci dunque nei prossimi dieci anni? Che cosa faremo con tutta questa potenza di calcolo?

Le prime applicazioni dei circuiti neuronali saranno, ovviamente, per far meglio – cioè a prestazioni superiori e a costi inferiori – le cose che già facciamo: e questo perché l'impiego di una nuova tecnologia è collegato strettamente a considerazioni economiche e di mercato; e poi anche perché ci vuole tempo per capire il potenziale di una nuova metodologia, e per figurarsi come applicare i nuovi dispositivi e le nuove tecniche nel mondo reale.

In un secondo tempo, cominceranno quelle applicazioni che già sono state individuate, ma che non sono fattibili tecnicamente o economicamente con la tecnologia attuale.

Infine, emergeranno applicazioni nuove, che con il passare del tempo si faranno sempre più numerose, e di cui oggi non abbiamo la più pallida idea.

La prima fascia di applicazioni è appena cominciata e risulterà in prodotti commerciali fra qualche anno. In questo caso si parla di sistemi esperti, di elaborazione di segnali per telefonia, di radar e sonar, di sistemi di controllo adattativi, di sistemi per il riconoscimento di caratteri e voce e così via.

La seconda fascia di applicazioni è anch'essa appena cominciata, ma risulterà in prodotti commerciali un po' più tardi, fra quattro-cinque anni, perché ci vorrà più tempo, sia per lo sviluppo del prodotto, sia per la introduzione di mercato. A questa categoria di applicazioni appartengono i sistemi di visione, sistemi per la compressione di immagini, «picture phone» (videotelefono), sistemi medicali diagnostici e di laboratorio, robotica, riconoscimento di scrittura, di firme, di impronte digitali, di facce, eccetera, eccetera...

La terza fascia inizierà fra qualche anno, dopo che saranno disponibili commercialmente i primi *chip*, e porterà alla introduzione nel mercato dei primi prodotti fra sei-sette anni.

Con i primi *chip* neuronali, la creatività naturale dell'uomo troverà un mezzo espressivo e produrrà cose che oggi sono impensabili.

Siccome le ho già definite «idee» oggi impensabili, non mi ci provo nemmeno a fare una previsione, perché, se queste idee le penso ora, appartengono di sicuro alla seconda categoria.

È proprio il marchio di idee potenti e rivoluzionarie quello di produrre conseguenze impensabili e impensate: e certamente il calcolo neuronale farà parte di quel piccolo numero di idee che trasformano irreversibilmente il mondo in cui viviamo.

Mi ritengo fortunato di far parte di quel gruppo di persone che intendono rendere tutto questo possibile.

FEDERICO FAGGIN