

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI INGEGNERIA E ARCHITETTURA
CORSO DI LAUREA TRIENNALE IN INGEGNERIA
BIOMEDICA

**Tatto artificiale:
studio ed implementazione di una rete neurale per la
localizzazione degli impatti**

Tesi in
Calcolatori Elettronici

**Relatore:
Prof. Nicola Testoni**

**Presentata da:
Melissa Negrini**

**Correlatori:
Prof. Tullio Salmon Cinotti
Cristiano Aguzzi**

**Sessione II
Anno Accademico 2018/2019**

Ai miei genitori e a mio fratello,

Indice

Elenco delle figure	4
Introduzione	5
1 Il sistema nervoso	7
1.1 Il sistema nervoso centrale	8
1.2 Il sistema nervoso periferico	10
1.3 Il neurone	13
1.4 Il sistema somatosensoriale	15
1.4.1 I recettori cutanei	16
2 Il sistema nervoso artificiale	19
2.1 Il neurone artificiale	21
2.1.1 Funzioni di attivazione	23
2.2 La rete neurale artificiale	24
3 Ambiente di sviluppo	31
3.1 L'apprendimento supervisionato	33
3.1.1 Le fasi di apprendimento	34
3.2 Deep learning	35
3.3 Campo di utilizzo	35
4 Riconoscimento di impatti	37
4.1 Il materiale	38
4.2 L'algoritmo di simulazione d'impatti	39
4.3 L'implementazione della rete neurale	40

5 Risultati sperimentali	49
Conclusioni	56
Appendice A	59
Ringraziamenti	71

Elenco delle figure

1.1	Diagramma completo di un neurone biologico.	13
1.2	Recettori cutanei	17
2.1	Schema del neurone artificiale	21
2.2	Rete neurale	25
4.1	La piastra di alluminio (a sinistra) ed uno dei sensori (a destra) simulati per gli esperimenti descritti in questa tesi	38
4.2	Piastra specifica	39
4.3	Neural Network Start	41
4.4	Schermata di selezione dei dati	43
4.5	Schermata di convalida e prova dei dati	44
4.6	Schermata per l'architettura della rete	45
4.7	Schermata dei risultati	47
4.8	Schermata di elaborazione dei dati	48
5.1	Outoput 1 SimPlateImpacts	50
5.2	Outoput 2 SimPlateImpacts	52
5.3	Outoput 3 SimPlateImpacts	53

Introduzione

Nel presente elaborato di tesi viene proposta una rete neurale in grado di stabilire la posizione di un impatto su una lastra piana. Tale rete neurale è stata implementata e testata in laboratorio con risultati positivi. Lo scopo della rete neurale è quello di emulare il sistema nervoso biologico che, attraverso il senso del tatto, è in grado di interagire con l'ambiente esterno, captando informazioni e trasmettendole al cervello, dove vengono elaborate per poi produrre una risposta adeguata. Quando veniamo punti da un ape, per esempio, sappiamo immediatamente dove siamo stati punti. Questo processo, che porta ad una risposta immediata nell'uomo, può essere insegnato a un calcolatore elettronico tramite l'utilizzo delle reti neurali.

Per riprodurre il sistema nervoso è stata usata una rete neurale; per emulare il funzionamento della cute una lastra piana di alluminio; per imitare il funzionamento dei recettori cutanei sono stati impiegati dei sensori piezoelettrici. Toccando la piastra i sensori ricevono delle onde descritte da parametri quali angolo d'incidenza, energia e tempo di ricezione. Questi dati vengono manipolati tramite il software MatlabTM e in seguito forniti alla rete neurale che li elabora.

Durante l'attività di tesi descritta in questo elaborato ho sia generato i dati necessari per l'addestramento, che progettato ed addestrato la

rete neurale, e verificato l'accuratezza del comportamento del sistema al termine dell'addestramento. Per ottimizzare il comportamento della rete ho utilizzato il *Pattern Recognition Toolbox* fornito da MatlabTM; è stato necessario sia modificare gli ingressi della rete, che sono stati fatti aumentare con gradualità così da fornire al sistema dati in volume sempre maggiore da cui apprendere, sia scegliere il numero dei neuroni utilizzati e la loro struttura gerarchica, per garantire che la rete sia in grado di generalizzare in modo adeguato. Il risultato ottenuto mostra come sia possibile identificare correttamente l'area di un contatto sulla lastra con soli 4 sensori e con un errore medio nettamente inferiore al 10%.

Capitolo 1

Il sistema nervoso

Il sistema nervoso, oltre alla coordinazione delle funzioni vitali e cognitive, consente ad un organismo di interagire con l'ambiente. Esso è costituito da un insieme di elementi che, dal punto di vista prettamente ingegneristico, formano una stratificazione di reti di comunicazione. È provvisto di apparati sensoriali, volti a rilevare gli eventi che si verificano nell'ambiente, di apparati d'integrazione, che elaborano i dati sensoriali e li depositano nella memoria, e di apparati motori, che promuovono i movimenti e le secrezioni ghiandolari. Le funzioni del sistema nervoso comprendono la ricezione sensoriale, l'elaborazione delle informazioni e il comportamento.

Tra i molteplici processi compresi nell'elaborazione delle informazioni, quelli che sono di maggior interesse per questa attività di tesi riguardano la trasformazione e la combinazione dei segnali, ovvero l'integrazione nervosa, e l'apprendimento. L'elaborazione delle informazioni, compresi l'apprendimento e la memoria dipendono dalle connessioni intercellulari presenti nei circuiti neurali, realizzate in ambito

biologico da un insieme di eventi elettrici e chimici. Il comportamento può pertanto essere definito come la risposta globale dell'organismo al suo ambiente e si può manifestare sotto forma di movimento o tramite una risposta autonoma.

Il sistema nervoso, costituito da complessi di cellule di cui una parte sono impiegate nella rete di comunicazione ed un'altra per la matrice di supporto, è normalmente suddiviso in 2 porzioni: il sistema nervoso centrale e il sistema nervoso periferico. Il primo sovrintende alle principali funzioni di controllo ed elaborazione, ed è contrapposto al secondo, che ha funzione trasmissiva di stimoli e risposte; il sistema nervoso periferico si può a sua volta dividere in sistema nervoso somatico e sistema nervoso autonomo. Il sistema nervoso somatico coinvolge i sensi e permette all'organismo di percepire suoni, odori, sapori, immagini, temperature e pressioni, mentre il sistema nervoso autonomo controlla le funzioni degli organi vitali.[2]

1.1 Il sistema nervoso centrale

Il sistema nervoso centrale è formato dall'encefalo, di cui il cervello è una parte, e dal midollo spinale. Ha il compito di percepire ed elaborare le informazioni ottenute dal sistema nervoso periferico oltre ad organizzare i riflessi e altre risposte comportamentali. Esso svolge anche funzioni cognitive, di apprendimento e di memorizzazione, ed è preposto alla programmazione e all'esecuzione dei movimenti volontari.

L'encefalo, racchiuso nella scatola cranica, è costituito dal cervello, dal tronco encefalico e dal cervelletto. Il tronco encefalico costituisce di fatto la prosecuzione craniale del midollo spinale, e possiede para-

gonabili circuiti che generano archi riflessi oltre a complesse funzioni integrative, in particolar modo legate al controllo delle funzioni vitali. Il cervelletto riveste il ruolo di centro di coordinazione motoria in base a molteplici informazioni ottenute dal sistema somatico.

Il cervello è il centro integrativo più alto e raffinato del sistema nervoso centrale, esso può essere suddiviso in due macro parti, il diencefalo ed il telencefalo; il primo, più piccolo ed interno, è avvolto superiormente e lateralmente dal secondo, più esterno, che include la corteccia cerebrale ed è estremamente sviluppato nella specie umana, se confrontato con altri mammiferi. La corteccia cerebrale umana gioca un ruolo centrale nei meccanismi o funzioni mentali cognitive complesse come pensiero, consapevolezza, memoria, attenzione, linguaggio. In essa i neuroni possono avere connessioni cortico-corticali con neuroni di altre aree della corteccia stessa, o connessioni cortico-sottocorticali con strutture più interne dell'encefalo. Schematicamente si può affermare che le cellule nervose di questa parte del cervello sono disposte in modo tale che il corpo del neuroni sia posto nella parte più esterna dell'encefalo (materia grigia), mentre nella parte più interna si trovano le fibre nervose che trasmettono e ricevono le informazioni da ogni parte del corpo (materia bianca).

Il midollo spinale può essere schematizzato come una sorta di lungo cilindro posto all'interno della colonna vertebrale. Esso ha una configurazione a specchio rispetto alla corteccia cerebrale: la materia grigia, e quindi i corpi dei neuroni, si trova nella parte più interna al midollo spinale mentre la materia bianca è situata all'esterno.

Sia il cervello che il midollo spinale hanno come unità funzionale il neurone, cellula dotata di prolungamenti avvolti da mielina; da en-

trambi partono nervi che portano le informazioni alle parti del corpo ed entrambi sono avvolti da una membrana protettiva che chiamiamo meningi.[2]

1.2 Il sistema nervoso periferico

Il sistema nervoso periferico è una vera e propria interfaccia tra il sistema nervoso centrale e l'ambiente, e permette alle informazioni di muoversi dal centro alla periferia e viceversa. Esso comprende componenti sensoriali (formate da recettori sensoriali e da neuroni afferenti primari) e componenti motorie (formate da motoneuroni somatici e autonomi).

I recettori sensoriali funzionano da trasduttori, capaci di rilevare gli stimoli¹ provenienti dall'ambiente e fornire all'organismo le informazioni che ne ricavano. L'effetto che lo stimolo ha sull'organismo è chiamato *risposta*, resa possibile dalla trasduzione sensoriale che a sua volta si verifica in seguito all'applicazione di energia meccanica, termica o chimica. Le informazioni ottenute dai recettori sensoriali sono trasmesse al sistema nervoso centrale tramite le fibre afferenti primarie sotto forma di scariche d'impulsi nervosi. I recettori sensoriali possono essere classificati in base al compito che svolgono (o meglio alle parti che mettono in comunicazione): esterocettori, intercettori e propriocettori (questi ultimi forniscono, ad esempio, informazioni sulla posizione del corpo nello spazio).

Molto importante nella fisiologia sensoriale è lo studio della relazione tra la localizzazione di uno stimolo e l'attivazione di un determinato

¹evento ambientale che attiva i recettori sensoriali

gruppo di neuroni. Il campo ricettivo di un neurone sensoriale è la regione periferica la cui stimolazione provoca una modificazione della scarica del neurone. I campi recettivi di neuroni del sistema sensoriale, ad esempio, sono piccole zone del corpo. Un recettore sensoriale può essere attivato solo dalla deformazione di una piccola regione cutanea detta campo ricettivo eccitatorio del recettore sensoriale.

I campi ricettivi dei neuroni sensoriali centrali hanno grandezza superiore a quella dei campi ricettivi dei recettori perché i neuroni sensoriali centrali ricevono informazioni dai vari recettori sensoriali ognuno dei quali ha un diverso campo recettivo. I campi ricettivi dei recettori sensoriali sono eccitatori mentre i neuroni sensoriali centrali possono essere anche inibitori.[2]

Codifica delle informazioni sensoriali

I neuroni sensoriali codificano gli stimoli. Nel processo di trasduzione sensoriale uno o più aspetti dello stimolo devono essere codificati in modo che possano essere interpretati dal sistema nervoso centrale. Alcuni degli aspetti che vengono codificati sono la localizzazione spaziale, l'intensità, la frequenza e la durata.

La localizzazione spaziale dello stimolo è segnalata dall'attivazione di una particolare popolazione di neuroni sensoriali i cui campi recettivi vengono interessati dallo stimolo. La capacità di discriminare come distinti due stimoli adiacenti può dipendere sia dall'eccitazione di popolazioni parzialmente separate di neuroni sia da interazioni inibitorie.

Si definisce *stimolo soglia* il minimo stimolo percepibile, ovvero lo

stimolo di intensità più bassa per il quale si ha una risposta. Uno stimolo è avvertito dal sistema nervoso centrale se provoca la comparsa di potenziali di recettore abbastanza ampi da attivare una o più fibre afferenti primarie. Il livello di sommazione spaziale e temporale è un parametro che definisce il numero di fibre afferenti primarie che devono essere eccitate affinché si verifichi l'attivazione del neurone sensoriale

La relazione tra l'intensità dello stimolo e la risposta può essere rappresentata come funzione stimolo-risposta:

$$risposta = costante \times n \times (stimolo - stimolo\ soglia)$$

I recettori di temperatura (termocettori) sono caratterizzati da $n = 1$, mentre i recettori del dolore (nocicettori) hanno $n \leq 1$.

L'intensità dello stimolo può essere codificata dal numero dei recettori che vengono attivati da quello stimolo: uno stimolo che abbia intensità che raggiunge la soglia di percezione può attivare una o poco più fibre afferenti primarie, uno stimolo più intenso attiverà un numero maggiore di recettori. I neuroni centrali che ricevono afferenze da una particolare classe di recettori sensoriali saranno attivati più intensamente qualora venga attivato un numero maggiore di afferenze. Stimoli di intensità diversa possono attivare gruppi diversi di recettori sensoriali: ad esempio, uno stimolo meccanico lieve sulla cute stimola l'attivazione dei meccanocettori, ma se l'intensità aumenta notevolmente potrebbero attivarsi anche i nocicettori.

La frequenza dello stimolo può essere codificata dagli intervalli esistenti tra le scariche dei neuroni sensoriali.

La durata degli stimoli viene codificata in modo diverso per recettori ad andamento lento o veloce. Nel primo caso la durata corrisponde

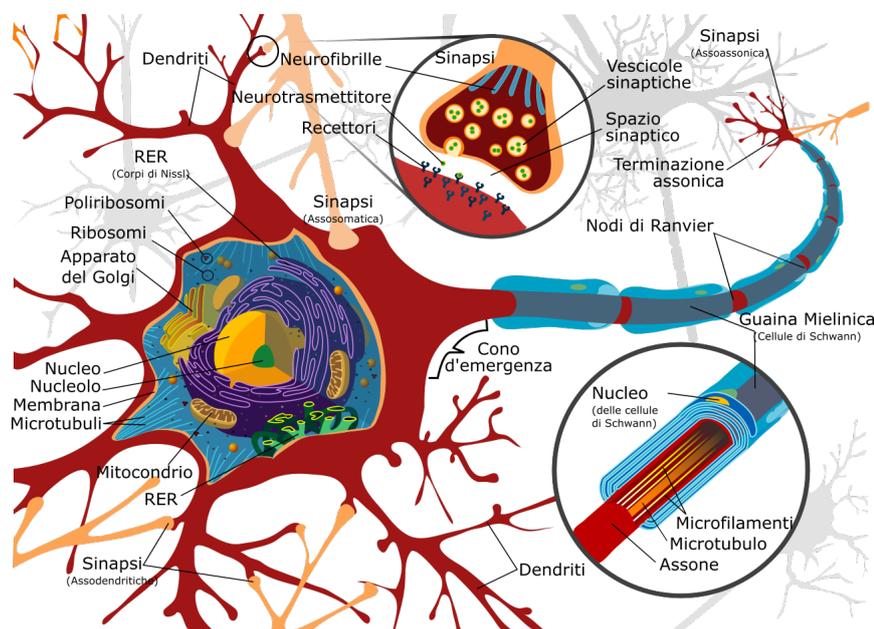


Figura 1.1: Diagramma completo di un neurone biologico.

alla durata dell'aumentata frequenza di scarica, mentre nel secondo caso l'applicazione e la rimozione dello stimolo sono segnalati da brevi scariche di potenziali d'azione.[2]

1.3 Il neurone

I tre componenti fondamentali del sistema nervoso sono i neuroni principali, i neuroni intrinseci e le fibre nervose. I neuroni sono le unità funzionali del sistema nervoso, sono le cellule che hanno il compito di veicolare le informazioni, sotto forma di impulsi nervosi, dagli organi al cervello e viceversa.[3]

Come mostrato in Figura 1.1, il neurone ha una superficie ricettiva formata da un corpo cellulare (chiamato soma), da una o più ramifi-

cazioni dette dendriti e una struttura tubolare più lunga detta assone alla cui estremità vi sono le sinapsi attraverso le quali le fibre nervose trasportano i segnali da un neurone all'altro. Un neurone ha un buon grado di convergenza e divergenza: i segnali provenienti da, potenzialmente, migliaia di altri neuroni vengono captati dai dendriti che li convergono verso il soma del neurone il cui assone a sua volta li diverge fino a migliaia di terminali (sinapsi) che vanno a toccare diversi altri neuroni. Qui l'impulso nervoso stimola il rilascio dei neurotrasmettitori, sostanze chimiche presenti nelle sinapsi, che vengono captati e riconosciuti dai recettori del neurone ricevente. Le sinapsi convertono un segnale elettrico in un segnale chimico e viceversa, ma sono a senso unico: alcune svolgono la prima trasformazione, altre la seconda. Vi sono due tipi di sinapsi: la maggioranza sono eccitatorie, mentre quelle inibitorie sono molte meno ma posizionate in punti strategici e risulteranno tanto più efficaci quanto si trovano in prossimità del soma.

Il neurone emette un segnale lungo il suo assone quando la differenza di potenziale tra l'interno e l'esterno della membrana raggiunge un certo livello di soglia. Mentre a riposo è possibile misurare un potenziale di membrana di circa -70 mV, la ricezione di un segnale causa una rapida depolarizzazione della membrana verso valori positivi: tale fenomeno è in grado di provocare l'emissione di una scarica lungo l'assone, cui segue una fase di ripolarizzazione per tornare ai valori negativi maggiori di quelli a riposo. Questo ciclo di depolarizzazione-ripolarizzazione è dovuto al flusso di diverse correnti di ioni attraverso la membrana del soma. La risposta del neurone è influenzata dalle proprietà della membrana, dai tipi di correnti ioniche e dalla presenza di neurotrasmettitori e neuromodulatori nei pressi del soma.

I neuroni controllano cosa che facciamo, dai semplici movimenti involontari fino alle funzioni cognitive superiori. La ricezione e la trasmissione delle informazioni sono permesse dalla proprietà dei neuroni chiamata eccitabilità che si manifesta con eventi chimici quali potenziali d'azione, potenziali di recettore e potenziali sintetici.

1.4 Il sistema somatosensoriale

Una via sensoriale può essere vista come un insieme di più neuroni messi in serie; convenzionalmente si possono dividere in neuroni di primo, secondo, terzo e quarto ordine.

I neuroni del primo ordine sono i neuroni sensoriali primari, la cui estremità periferica costituisce il recettore sensoriale; perciò questo neurone risponde agli stimoli e trasmette al sistema nervoso centrale informazioni codificate. Il corpo di questi neuroni si trova nei gangli delle radici dorsali o in quelli dei nervi cranici.

I neuroni del secondo ordine sono generalmente nel midollo spinale o nel tronco cerebrale, essi ricevono le informazioni dai neuroni del primo ordine e le trasmettono al talamo². Queste informazioni possono subire una prima elaborazione grazie ai circuiti locali di questi neuroni.

I neuroni del terzo ordine sono situati nei nuclei sensoriali del talamo. Anche qui l'informazione può essere rielaborata prima di arrivare alla corteccia cerebrale.

²Sorta di "sorvegliante" che concorre a decidere quali sono le emozioni e i pensieri consentiti. Il talamo valuta il grado di coerenza delle informazioni rispetto a quanto appreso in passato ed è perciò responsabile della creazione dei modelli, valori e obiettivi che regolano il nostro comportamento

I neuroni del quarto ordine sono situati nell'appropriata area di ricezione sensoriale primaria della corteccia cerebrale. Essi, insieme a neuroni di ordine superiore, rielaborano ulteriormente l'informazione proviene dai neuroni di ordine inferiore.

Il sistema somatoviscerale comprende le unità sensoriali i cui apparati di ricezioni sono localizzati nella cute, nei muscoli, nelle articolazioni e nei visceri. Le informazioni derivanti da questi recettori raggiungono il sistema nervoso centrale attraverso i neuroni afferenti primari. Ogni assone si biforca in un processo periferico, che ha le caratteristiche strutturali dell'assone e termina perifericamente come recettore sensoriale, ed in un processo centrale, che è anch'esso un assone ed entra nel midollo spinale o nel tronco cerebrale.

1.4.1 I recettori cutanei

Questo tipo di recettori, illustrati in Figura 1.2, possono essere suddivisi in base al tipo di stimolo al quale rispondono. I principali sono i meccanocettori, i termocettori, i nocicettori e i chemocettori.

I meccanocettori rispondono a stimoli meccanici che deformano la cute o che vengono mossi lentamente (come nello strofinamento). Possono essere a rapido o lento andamento. I primi sono rappresentati dai recettori dei follicoli piliferi delle regioni cutanee dove sono presenti peli, dai corpuscoli di Meissner della cute glabra e dai corpuscoli di Bacini del tessuto sottocutaneo. I meccanocettori ad andamento lento sono rappresentati dalle terminazioni delle cellule di Merkel (campi recettivi puntiformi) e dai corpuscoli di Ruffini (attivabili da uno stira-

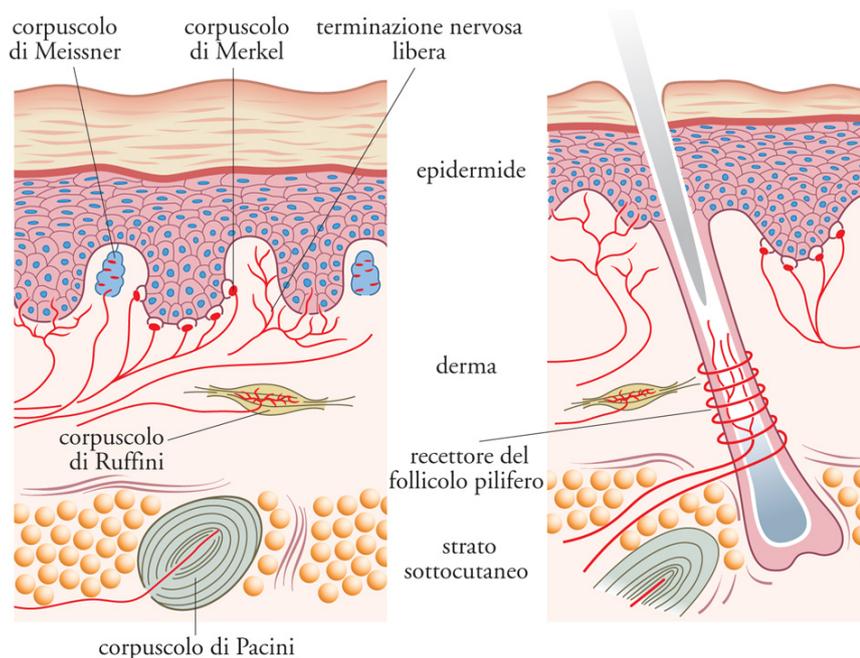
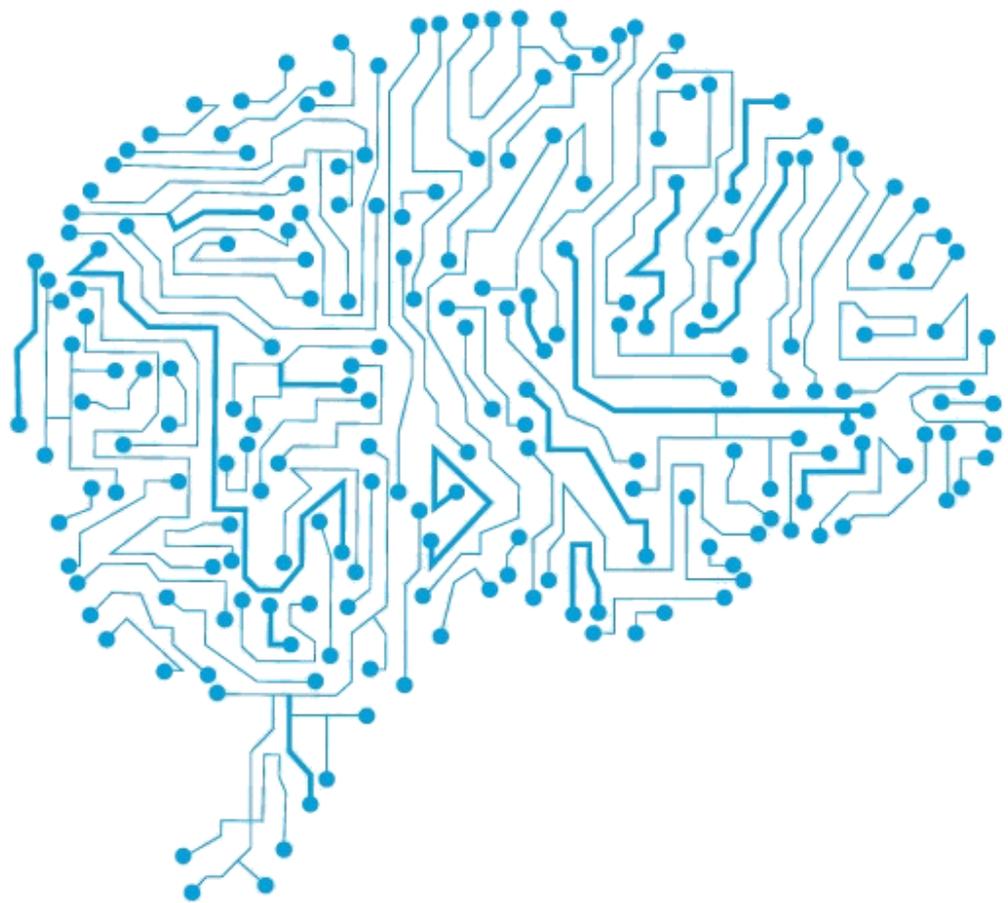


Figura 1.2: Recettori cutanei

mento di una regione cutanea posta a una certa distanza dai terminali del recettore).

I termocettori segnalano al sistema nervoso centrale la temperatura della pelle. Vi sono i termocettori sensibili al freddo e quelli sensibili al caldo; entrambi hanno un andamento lento. Questi recettori sono dotati di attività spontanea. Ci sono fasce di temperatura in cui entrambi i tipi sono attivi (ad esempio a 35 gradi) mentre nelle altre fasce di temperatura, quando sono attivi i termocettori sensibili al caldo quelli sensibili al freddo sono silenti, e viceversa. I recettori per il caldo si “spengono” anche quando la temperatura supera un limite oltre il quale è considerata nociva. In questo stato essi non sono più in grado di trasmettere al sistema nervoso centrale alcuna informazione.

I nocicettori rispondo a stimoli che possono produrre un danno.



Capitolo 2

Il sistema nervoso artificiale

Il modo in cui i programmi tradizionali elaborano l'informazione è radicalmente diverso dal modo in cui funzionano i sistemi biologici primari. Questo è il principale ostacolo per cui i computer non possono compiere azioni che per noi sono banali.

Un computer è composto da un processore (che esegue in successione delle istruzioni/operazioni) e da tre tipi di memoria: una contiene le informazioni che servono per svolgere le operazioni, una temporanea da cui vengono letti i dati necessari e depositati i risultati dei calcoli effettuati e una permanente in cui i dati rimangono registrati.

Un programma è formato da un insieme di istruzioni/operazioni ordinate in modo tale che vengano eseguite dal computer seguendo regole e conoscenze predefinite.

I sistemi di elaborazione seriale sono molto efficienti nel risolvere problemi che noi essere umani troviamo complicati mentre sono parti-

colarmente inefficienti quando si tratta di azioni tratte dalla semplice vita quotidiana. I primi sono problemi risolvibili tramite l'utilizzo di regole e formule; sono quindi problemi analitici e sono appunto per questo affrontati in modo più rapido da un calcolatore rispetto ad un uomo. I secondi invece sono problemi che non seguono un percorso predefinito per arrivare a una soluzione e perciò risulteranno irrisolvibili per un computer ma molto semplici per un uomo.

Si arriva quindi alla conclusione che il funzionamento del sistema nervoso è radicalmente diverso da quello di un sistema di elaborazione seriale dell'informazione. Le principali differenze riguardano i seguenti aspetti:

- L'elaborazione dell'informazione nei sistemi nervosi avviene in parallelo, mentre nei calcolatori tradizionali ogni dato viene elaborato individualmente e in successione. Il fatto che i neuroni lavorino in parallelo spiega la maggior velocità del cervello a risolvere problemi che richiedono l'elaborazione contemporanea di più dati, ad esempio il riconoscimento visivo di oggetti.
- L'elaborazione nei sistemi nervosi è distribuita su molti elementi, ovvero vi sono più neuroni che svolgono la stessa operazione.
- Ogni dato della memoria dei calcolatori viene identificato da un indirizzo usato poi per recuperare quell'informazione, invece gli esseri umani accedono alle loro informazioni in base al contenuto, pertanto possiamo accedere a un ricordo grazie a un profumo, una voce, etc. . .
- I sistemi nervosi imparano autonomamente mentre un calcolatore

deve essere programmato. L'apprendimento è associato alla forza con cui un neurone è legato agli altri: tanto è più forte la sinapsi tanto maggiore sarà l'effetto del segnale che passa sul neurone ricevente.

In conclusione, i computer al momento non possono essere considerati intelligenti nel senso umano del termine, sebbene siano in grado già ora di risolvere problemi complessi con una opportuna programmazione. Allo scopo di colmare questa lacuna, le reti neurali artificiali si propongono come sistemi di elaborazione dell'informazione appositamente pensati per i computer ma il cui comportamento prende ispirazione dai sistemi nervosi biologici.

2.1 Il neurone artificiale

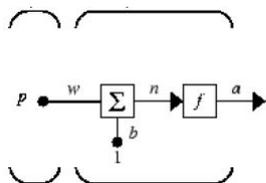


Figura 2.1: Schema del neurone artificiale

I neuroni artificiali sono le unità di elaborazione delle reti neurali artificiali e sono caratterizzati da un insieme di sinapsi che corrispondono ai terminali di altri neuroni, da una soglia e da una funzione di attivazione. Questi neuroni sono chiamati anche unità, nodi o processori. Le unità che ricevono informazioni dall'ambiente sono dette unità d'ingresso (input); quelle che emettono risposte nell'ambiente sono dette unità d'uscita (output). Oltre agli input e agli output possono esse-

re presenti nodi che comunicano tra di loro all'interno della rete: tali nodi questi sono chiamati *hidden* ovvero nascosti all'ambiente perché appunto non si interfacciano con l'ambiente nè in entrata nè in uscita. Nella maggior parte dei modelli vengono ignorati i ritardi di trasmissione dovuti alla lunghezza e alle caratteristiche delle fibre nervose, di conseguenza l'effetto di un segnale x_j sul neurone postsinaptico è dato dal prodotto tra x_j e il peso attribuito alla sinapsi (w_{ij}).

Come i neuroni nel sistema nervoso, i neuroni artificiali svolgono ognuno un'operazione molto semplice che consiste nel diventare attivi se la quantità totale di segnale che riceve supera la soglia di attivazione (q) propria dell'unità. Se un'unità si attiva emette un segnale che viene trasmesso lungo i canali di comunicazione fino alle altre unità a cui essa è connessa. Ciascun punto di connessione (che rappresenta la sinapsi del sistema nervoso biologico) agisce come un filtro che trasforma il segnale ricevuto in inibitorio o eccitatorio aumentandone o diminuendone l'intensità a seconda delle proprie caratteristiche individuali.

L'input netto, o potenziale di attivazione A_i di un neurone i -esimo è dato da

$$A_i = \sum_{j=1}^N w_{ij}x_j - \theta_i \quad (2.1)$$

dove θ_i rappresenta il valore della soglia.

La risposta del neurone y_i viene calcolata sottoponendo il potenziale di attivazione all'azione di una delle numerose funzione di attivazione Φ a disposizione del progettista della rete, alcune delle quali verranno

descritta in seguito

$$y_i = \Phi(A_i) = \Phi \left(\sum_{j=1}^N w_{ij} x_j - \theta_i \right) \quad (2.2)$$

Le sinapsi artificiali hanno il compito di pesare l'intensità dei segnali trasmessi, e vengono anche definite pesi (w_{ij}). Essi possono essere positivi o negativi e possono essere modificati durante la fase di apprendimento. Poiché una rete neurale è composta di uno o più neuroni, ciascuno dei quali riceve una o più connessioni sintetiche, è vantaggioso analizzare il sistema in notazione vettoriale. Il potenziale di attivazione di un neurone è una funzione lineare dei segnali d'ingresso quindi il potenziale di un intero strato di neuroni si può scrivere come segue

$$A = Wx \quad (2.3)$$

dove x è il vettore dei segnali d'ingresso e W è la matrice di connessioni sinaptiche in cui le righe corrispondono ai neuroni riceventi e le colonne ai segnali d'ingresso.

2.1.1 Funzioni di attivazione

La funzione di attivazione determina il tipo di risposta che un neurone è in grado di emettere. Nella formulazione di McCulloch e Pitts la risposta del neurone è data da una funzione gradino (la soglia (θ) viene mantenuta fuori dal calcolo del potenziale di attivazione)

$$\Phi(A) = \begin{cases} 1, & \text{se } A > \theta \\ 0, & \text{altrimenti} \end{cases} \quad (2.4)$$

In modo alternativo l'output può essere bipolare

$$\Phi(A) = \begin{cases} 1, & \text{se } A > \theta \\ -1, & \text{altrimenti} \end{cases} \quad (2.5)$$

In ogni caso il neurone può essere in soli due stati (attivo o inattivo) e trasmettere quindi un bit di informazione. Maggior informazione può essere trasmessa se si usa una funzione continua lineare $\Phi(A) = kA$ che può essere forzata a lavorare dentro un determinato intervallo che contenga l'attivazione del neurone. La proprietà di trasmettere una gradazione di segnali di intensità tale da risultare utile per la risposta dei neuroni riceventi è paragonabile alla frequenza di scarica dei neuroni biologici. Un esempio di funzione continua non lineare è la funzione sigmoide

$$\Phi(A) = \frac{1}{1 + e^{-kA}} \quad (2.6)$$

in cui k è una costante che controlla l'inclinazione della curva (quanto $k \rightarrow \infty$ la funzione sigmoide approssima una funzione gradino). Nella maggior parte dei modelli tutti i neuroni della rete, tranne i recettori, utilizzano la stessa funzione di attivazione per calcolare il proprio segnale di uscita.

È possibile dimostrare che un insieme di questi neuroni artificiali connessi in rete può svolgere qualsiasi tipo di calcolo aritmetico e funzione logica.

2.2 La rete neurale artificiale

Quando un vettore input (stimolo) viene applicato ai neuroni d'ingresso della rete neurale i segnali viaggiano in parallelo lungo le connessioni

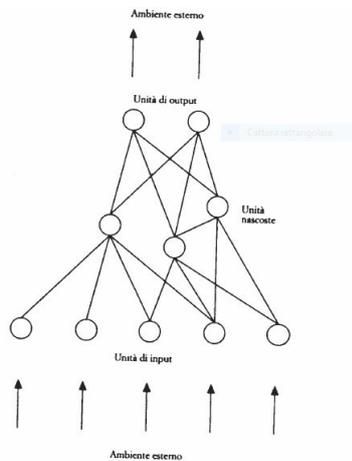


Figura 2.2: Rete neurale

attraverso i nodi interni, se ce ne sono, fino ad arrivare all'uscita e produrre quindi la risposta della rete neurale. Nella formulazione più semplice, ciascun nodo elabora solo l'informazione locale, non conosce lo scopo globale dell'elaborazione e non possiede memoria. La risposta e il comportamento della rete dipendono in modo intrinseco dalla sua architettura e dai valore delle sinapsi artificiali.

L'architettura di una rete neurale è caratterizzata dalla distinzione tra neuroni di ingresso e neuroni di uscita, dal numero di strati di sinapsi (o neuroni) e dalla presenza di connessioni di retroazione.

Nelle reti etero-associative i nodi d'ingresso che ricevono input dall'ambiente esterno sono distinti dai nomi di uscita che forniscono la risposta della rete. Queste reti imparano ad associare coppie di vettori diversi: il vettore di input e il vettore di risposta. Questo tipo di rete viene usata per compiti di classificazione ed estrazione di informazioni rilevate dal segnale d'ingresso. In alcuni casi un singolo strato di sinapsi non è sufficiente per apprendere un'associazione desiderata

tra pattern di ingresso e pattern di uscita: in questi casi è necessario impiegare reti multi strato che possiedano neuroni interni e più di uno strato di sinapsi. La risposta di una rete del genere si ottiene calcolando l'attivazione di uno strato di neuroni alla volta procedendo gradualmente dai nodi interni verso i nodi di uscita.

Il termine *feedforward* viene usato per indicare le reti in cui ciascun nodo riceve connessioni solo dagli strati inferiori. Queste reti emettono una risposta per ogni pattern di ingresso ma non riescono a cogliere l'eventuale struttura temporale dell'informazione d'ingresso o ad esibire dinamiche temporali endogene. Un modo per dotare le reti etero-associative di caratteristiche temporali consiste nell'aggiungere connessioni di retroazione, ovvero connessioni ricorrenti che permettono la ricezione di connessioni da neuroni dello stesso strato o da neuroni di strati superiori. In questo caso l'output di un nodo avrà un'equazione del tipo

$$y_i(t) = \Phi \left(\sum_{j=1}^N w_{ij}x_j(t) + c \sum_{l=1}^M r_{il}q_l(t-1) - \theta_i \right) \quad (2.7)$$

dove $q_l(t-1)$ sono le attivazioni dei nodi dello stesso strato o degli strati superiori all'istante precedente e r_{il} sono le connessioni sintetiche ricorrenti. Quando un nodo è connesso a tutti i nodi della rete si dice completamente connesso. I pesi delle sinapsi di una rete neurale sono i parametri che devono essere stimati durante l'apprendimento dei pattern di input.

Le reti auto-associative possiedono un unico strato di unità interamente connesse tra di loro. Ogni unità riceve input dall'ambiente e dalle altre unità, la sua risposta quindi può variare nel tempo anche se

il segnale che riceve è costante. La risposta di questa rete viene calcolata ripetendo per molte volte il calcolo dell'attivazione dei suoi nodi. Questo tipo di rete viene utilizzata per apprendere, immagazzinare e ricostruire i pattern. Esse associano l'una all'altra parti diverse di uno stesso pattern. L'apprendimento avviene presentando ciascun pattern a tutte le unità della rete e modificando i valori di tutte le connessioni sinaptiche. Un volta che il pattern è stato memorizzato è sufficiente presentarne solo una parte e calcolare in modo ciclico l'output di ciascun nodo

$$y_i(t) = \Phi(I_i + \sum_{j \neq i}^N w_{ij} y_j(t-1)) \quad (2.8)$$

Dopo alcune interazioni del calcolo dell'attivazione la rete ricostruisce la versione originale del pattern memorizzato. Questo passaggio corrisponde a uno stato di equilibrio in cui le attivazioni dei nodi della rete cessano di variare.

Una rete neurale impara a dare le risposte corrette a ciascuno stimolo d'ingresso modificando i valori delle proprie connessioni sinaptiche in base a delle regole di apprendimento. Solitamente per "insegnare" alla rete è necessario presentare numerose volte lo stesso esperimento. Le regole di apprendimento di un modello neurale sono le condizioni locali e le modalità in cui le sinapsi si modificano a prescindere dal tipo di compito per cui la rete verrà utilizzata.

Alcune caratteristiche importanti delle reti neurale sono la robustezza, la flessibilità, la generalizzazione e il recupero in base al contenuto.

Le funzioni di attivazione sono scelte in base a quello che si vuole ottenere in uscita. L'adozione di una particolare funzione ha conse-

guenze che influenzano le prestazioni e l'applicabilità di un modello neurale. Le codifiche bipolari si sono rivelate più vantaggiose rispetto a quelle binarie perché riescono a risolvere problemi che le codifiche binarie non riescono a svolgere. Esse hanno inoltre il vantaggio di porre uguale a 0 i componenti mancanti (di input incompleti) mentre nelle codifiche binarie lo 0 ha un valore non neutro. È presente un altro tipo di codifica, la cosiddetta codifica locale, scartata a priori per le reti neurali artificiali a causa dei numerosi difetti che presenta, quali la fragilità e la necessità di un numero elevato di unità per svolgere il problema. Vi sono poi vari tipi di codifica distribuita. Esse non sono in grado di rappresentare separatamente oggetti distinti che vengono presentati simultaneamente alla rete, al contrario del sistema nervoso biologico che riesce in questo intento.

Lavorando con dati reali ci si rende conto che essi non sono omogenei in quanto ad energia: tale fatto può compromettere il funzionamento della rete neurale. Stimando l'energia di ogni segnale di ingresso è possibile riportare questi dati su di una stessa scala, ovvero è possibile normalizzarli. La normalizzazione consiste nel fare in modo che l'energia di ciascun vettore definita come $\|x\|_2 = \sqrt{x^T x} = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$ sia uguale a 1. Questo è facilmente ottenibile tramite la seguente sostituzione

$$x_j \leftarrow \frac{x_j}{\sqrt{\sum_{i=1}^N x_i^2}} \quad (2.9)$$

La normalizzazione rende più facili le operazioni riguardanti la distanza tra i vettori di addestramento e i vettori sinaptici, base del funzionamento degli algoritmi neurali. Qualsiasi rete neurale ben strutturata

può apprendere un determinato compito se i pattern di input sono stati elaborati e rappresentati in modo opportuno.

Capitolo 3

Ambiente di sviluppo

Quando si studiano le reti neurali artificiali lo scopo è quello di renderle sempre più simili alla rete neurale biologica. Esse infatti combinano diversi strati di processi non lineari che lavorano in parallelo così come fa il sistema nervoso. Una caratteristica fondamentale è l'apprendimento, ovvero la capacità della rete neurale artificiale di riuscire a captare uno stimolo, elaborarlo e infine produrre una risposta ad esso (così come fa il nostro sistema nervoso).

La risposta di una rete neurale dipende dai collegamenti sinaptici tra i nodi che la compongono. Come il sistema nervoso biologico impara dall'esperienza così la rete neurale impara variando i valori delle sinapsi attraverso la ripetizione di una serie di avvenimenti. Vi sono due tipi di apprendimento: l'*apprendimento supervisionato* nel quale i valori delle sinapsi vengono modificati usando una misura di errore tra la risposta fornita dalla rete neurale e la risposta desiderata per ogni vettore di input e l'*apprendimento per auto-organizzazione* nel quale sono imposte alcune regole di plasticità sinaptica che danno luogo

a una graduale auto-organizzazione della rete durante l'esposizione ai pattern di input. Queste due distinzioni non sono nette: il raggiungimento di un dato obiettivo può richiedere l'uso simultaneo di entrambi i metodi.

Tutti i modi di apprendimento hanno delle basi comuni. Inizialmente i valori iniziali dei pesi sinaptici della rete vengono assegnati in modo casuale entro un piccolo campo di variazione. Successivamente l'apprendimento prevede la ripetizione di una serie di vettori detti pattern di apprendimento: nell'apprendimento supervisionato il pattern è costituito da una coppia di vettori, quello d'ingresso e quello della risposta desiderata, nell'apprendimento auto-organizzato è presente solo il vettore degli input. La nuova configurazione di valori sinaptici dopo un ciclo di addestramento $w_{ij}(t)$ è calcolata sommando la modifica ottenuta $\Delta w_{ij}(t)$ (il calcolo di questo valore è quello che interessa agli algoritmi di apprendimento) e la configurazione sinaptica precedente $w_{ij}(t - 1)$. L'apprendimento procede in modo ricorsivo e graduale così che ciò che è stato appena appreso non cancelli o modifichi quello che già si sapeva. La velocità di apprendimento è regolata da una costante detta tasso di apprendimento η il quale controlla la modifica sui valori sinaptici.

L'equazione generale dell'apprendimento sarà quindi

$$w_{ij}(t) = w_{ij}(t - 1) + \eta \Delta w_{ij}(t) \quad (3.1)$$

dove η è strettamente compreso tra 0 e 1 in quanto andando oltre all'unità si potrebbe riscontrare instabilità. Al termine della fase di apprendimento i valori sinaptici vengono congelati ed è possibile studiare la rete su dei vettori di test: vengono quindi presentati dei nuovi

pattern d'ingresso e viene calcolata l'attivazione dei nodi della rete senza che i pesi sinaptici vengano modificati.

Il tempo necessario per considerare terminato l'apprendimento di una rete neurale dipende da molti fattori quali, ad esempio, il metodo usato o le finalità di utilizzo della rete.

3.1 L'apprendimento supervisionato

L'apprendimento supervisionato presuppone la presenza di un "insegnante" estraneo alla rete neurale che fornisce l'informazione utilizzata per la correzione della risposta della rete neurale: tale informazione può essere la risposta desiderata dalla rete o una valutazione della bontà della risposta della rete.

Data una rete neurale con N nodi d'ingresso e P coppie di addestramento, ogni coppia composta da un vettore d'ingresso bipolare x_p e da un target t_p anch'esso bipolare, l'output della rete per ciascun pattern d'ingresso è dato da

$$y = \begin{cases} 1, & \text{se } \sum_{i=0}^N w_i x_i > 0 \\ -1, & \text{altrimenti} \end{cases} \quad (3.2)$$

Una volta calcolato, il valore di y viene poi confrontato con t_p per quello stesso pattern d'ingresso: se $y = t_p$ i valori delle sinapsi non variano, se invece si verifica una discrepanza tra i due valori allora i pesi sinaptici vengono modificati in base alla risposta corretta $\Delta w_i(t) = \eta x_i(t)$ e addizionata al valore precedente $w_i(t) = w_i(t-1) + \Delta w_i(t)$.

I perceptron¹ sono una particolare categoria di reti neurali in grado di modificare il peso delle sinapsi. Essi hanno due importanti caratteristiche: in primo luogo l'apprendimento è graduale e richiede la presentazione ripetuta di tutte le coppie di addestramento, in secondo luogo, se esiste un vettore che soddisfa la trasformazione richiesta da input ad output, la regola di apprendimento riuscirà a trovarlo in un numero finito di cicli di apprendimento. La condizione necessaria affinché sia presente almeno una soluzione al problema dell'addestramento è che i pattern siano linearmente separabili.

3.1.1 Le fasi di apprendimento

La rete neurale che è stata implementata e che verrà descritta nel dettaglio nei capitoli successivi lavora secondo tre fasi: training, validation e testing. La prima fase, *training*, è quella dedicata all'apprendimento della rete; qui vengono studiati il 70% dei dati forniti.

La fase successiva, *validation*, è quella di convalida; in questa fase vengono usati la metà dei dati rimanenti. Eseguire la convalida a intervalli regolari aiuta a determinare se la rete sta andando in overfitting.

Infine c'è la fase di analisi, *testing*, nella quale la rete utilizza gli ultimi dati rimasti per verificare l'apprendimento.

¹[4]: Entità con un ingresso, un'uscita e una regola di apprendimento basata sulla minimizzazione dell'errore. L'apprendimento o la memorizzazione vengono ottenute per retroazione adattando i pesi numerici fintantoché l'uscita non sia resa pari a quella desiderata. Le capacità computazionali di un singolo perceptrone non sono però elevate e le prestazioni ottenibili dipendono sia dall'insieme degli ingressi sia dalla funzione da implementare.

3.2 Deep learning

Il deep learning, letteralmente "apprendimento profondo", è una branca dell'apprendimento automatico che mira ad insegnare ad un elaboratore a risolvere problemi analoghi a quelli affrontati dagli esseri umani. Il calcolatore studia i dati che gli vengono forniti utilizzando algoritmi che sfruttano metodi matematici per manipolare l'informazione senza dover usare un modello con un'equazione predefinita.

Il software Matlab[1] permette di creare reti neurali da zero oppure modificare reti preesistenti. È possibile usare una rete preaddestrata per la classificazione, il trasferimento dell'apprendimento, cioè riutilizzare una rete preaddestrata su un nuovo gruppo di dati, o per l'estrazione delle caratteristiche, ovvero usare una rete preaddestrata come supporto di partenza per la costruzione di una nuova rete.

3.3 Campo di utilizzo

Gli ambiti di utilizzo delle reti neurali sono molto ampi.

Per citare alcuni esempi, in ambito informatico le reti neurali sono state impiegate per la compressione/decompressione dei dati, per l'eliminazione del rumore, per il riconoscimento dei segnali sonar, per realizzare macchina da scrivere fonetiche e per il riconoscimento dei caratteri. Nei sistemi di controllo le reti neurali hanno trovato ampio utilizzo nel campo dei robot e dei veicoli intelligenti. Anche nelle analisi finanziarie sono state impiegate le reti neurali per vari compiti quali la valutazione dei rischi sugli investimenti e prestiti o l'analisi dell'andamento della borsa. In medicina le reti neurali hanno aiutato nella

diagnosi e prognosi delle patologie, potendo elaborare immagini quali raggi X, ad esempio, e riuscendo a riconoscere la presenza o meno di patologie. In neuroscienza il rapporto con le reti neurali prevede uno scambio reciproco di informazioni: le conoscenze neurofisiologiche offrono ispirazione e spunto per la costruzione di modelli artificiali le cui proprietà computazionali emergenti saranno studiate dalla neuroscienza per capire meglio il funzionamento del sistema nervoso. In psicologia i modelli neurali artificiali sono visti come alternativa ai modelli seriali nella comprensione della microstruttura dei processi cognitivi.

Per quanto riguarda il legame tra reti neurali e computer è da evidenziare il fatto che su ogni tipo di computer è possibile simulare una rete neurale. Se il computer possiede un'architettura di tipo seriale la fase di attivazione della rete neurale viene discretizzata e gli stati dei nodi vengono calcolati in modo seriale; se la rete possiede più di uno strato, per passare a quello superiore bisogna prima conoscere tutti i nodi di quello inferiore.

Capitolo 4

Riconoscimento di impatti

In questo capitolo verrà illustrato il procedimento che è stato utilizzato per la realizzazione di una rete neurale utile per la classificazione di impatti su una lastra metallica; verrà introdotto e descritto il materiale utilizzato durante l'esperienza in laboratorio. L'algoritmo *SimPlateImpacts* in appendice, listing A.1, permette di generare gli input per la rete neurale conformemente al setup sperimentale, e da la possibilità di considerare casi sia ideali che non ideali.

Per la trattazione di questa tesi ci siamo limitati al caso ideale: le rette rappresentative degli angoli captati da ogni sensore si intersecano tutte e quattro perfettamente nel punto dell'impatto. Allontanandosi dal caso ideale può capitare che il punto di intersezione sia discostato dal punto dell'impatto oppure che le rette non si intersechino tutte e quattro ma solo alcune di esse.

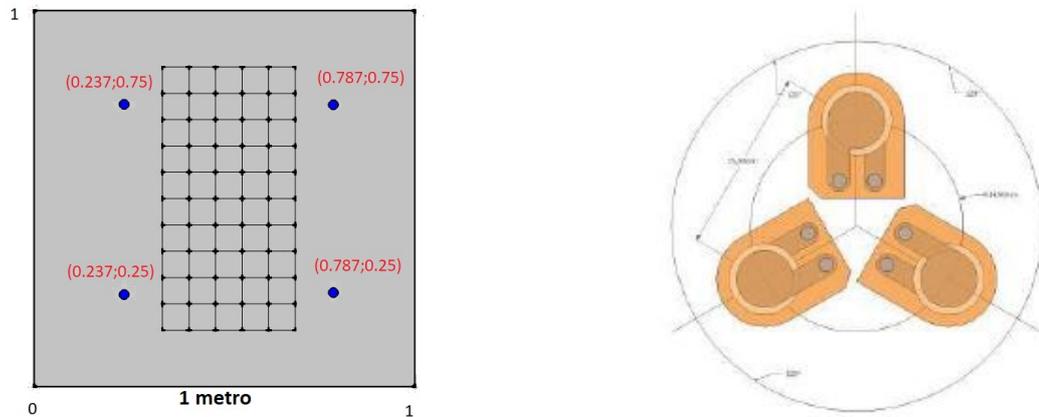


Figura 4.1: La piastra di alluminio (a sinistra) ed uno dei sensori (a destra) simulati per gli esperimenti descritti in questa tesi

4.1 Il materiale

Per gli esperimenti descritti in questa tesi si è fatto uso della simulazione di una lastra piana di alluminio di forma quadrata con lato pari ad 1 m illustrata in Figura 4.1 a sinistra. Su di essa sono stati posizionati quattro nodi sensori [5]: essi sono posti ai vertici di un quadrato di lato pari a 50 cm ed ogni sensore è composto da una tripletta di singoli elementi piezoelettrici posti a 120° l'uno rispetto all'altro come mostrato in Figura 4.1 a destra. L'area delimitata dai sensori è una porzione dell'area totale della lastra. Per l'addestramento della rete si è fatto uso del *Neural Network Toolbox*, insieme di strumenti realizzati in ambiente MatlabTM che verrà descritto di seguito, in grado di creare o modificare reti in grado di apprendere.

4.2 L'algoritmo di simulazione d'impatti

L'algoritmo SimPlateImpacts (vedi appendice) serve per simulare degli impatti su una lastra piana.

All'interno di questo programma come prima cosa è stata definita la geometria della lastra, l'ubicazione dei sensori, il numero e il luogo degli impatti.

Ai fini dell'esperienza descritta in questa tesi, dell'area totale della lastra è stata considerata solo la parte delimitata dal perimetro dei sensori. Nella Figura 4.2 è stata evidenziata in giallo la parte di griglia d'interesse. Gli impatti, nel caso ideale, sono localizzati su ciascun vertice dei quadretti gialli mostrati in figura.

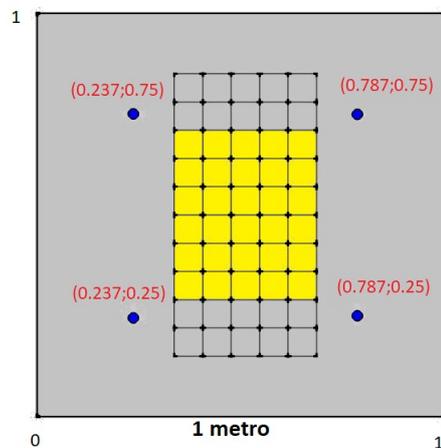


Figura 4.2: Piastra specifica

Il software SimPlateImpacts per ciascuno di questi impatti simulati produce in uscita l'angolo di incidenza relativo a ogni sensore (a), l'energia dell'impatto rilevata da ogni sensore (e) e il tempo di ricezione dell'impulso (t). Considerando che l'area evidenziata è quella

all'interno dei sensori, gli angoli in uscita avranno valori compresi in un intervallo di 90° .

4.3 L'implementazione della rete neurale

L'implementazione della rete neurale di cui mi sono occupata mira a classificare gli impatti sulla lastra per poterli poi elaborare con lo scopo di trasmettere all'interlocutore in quale punto della lastra è avvenuto l'impatto. In particolare è stata implementata una rete neurale di tipo feed-forward a un solo strato nascosto. Questo è esattamente quello che fa il sistema nervoso biologico; quando la cute viene sollecitata il cervello elabora l'informazione e sa in quale punto è avvenuto il contatto e con che intensità.

Per addestrare la rete neurale servono esempi da cui essa impara come rispondere; più il numero di esempi è alto più la rete avrà una risposta verosimile alla realtà.

Tramite l'applicazione MatlabTM *Pattern Recognition app* è possibile elaborare e modificare rete neurali pre-strutturate. Tramite il comando *nstart* nella Command Window si apre la pagina iniziale mostrata in Figura 4.3 e selezionando l'applicazione desiderata si può iniziare a modellare la rete a proprio piacimento.

La realizzazione di una rete neurale prosegue tramite 4 step:

1. **Select Data:** dalla schermata mostrata in Figura 4.4 è possibile selezionare inputs e targets della nostra rete e la modalità di lettura degli esempi per righe o per colonne .

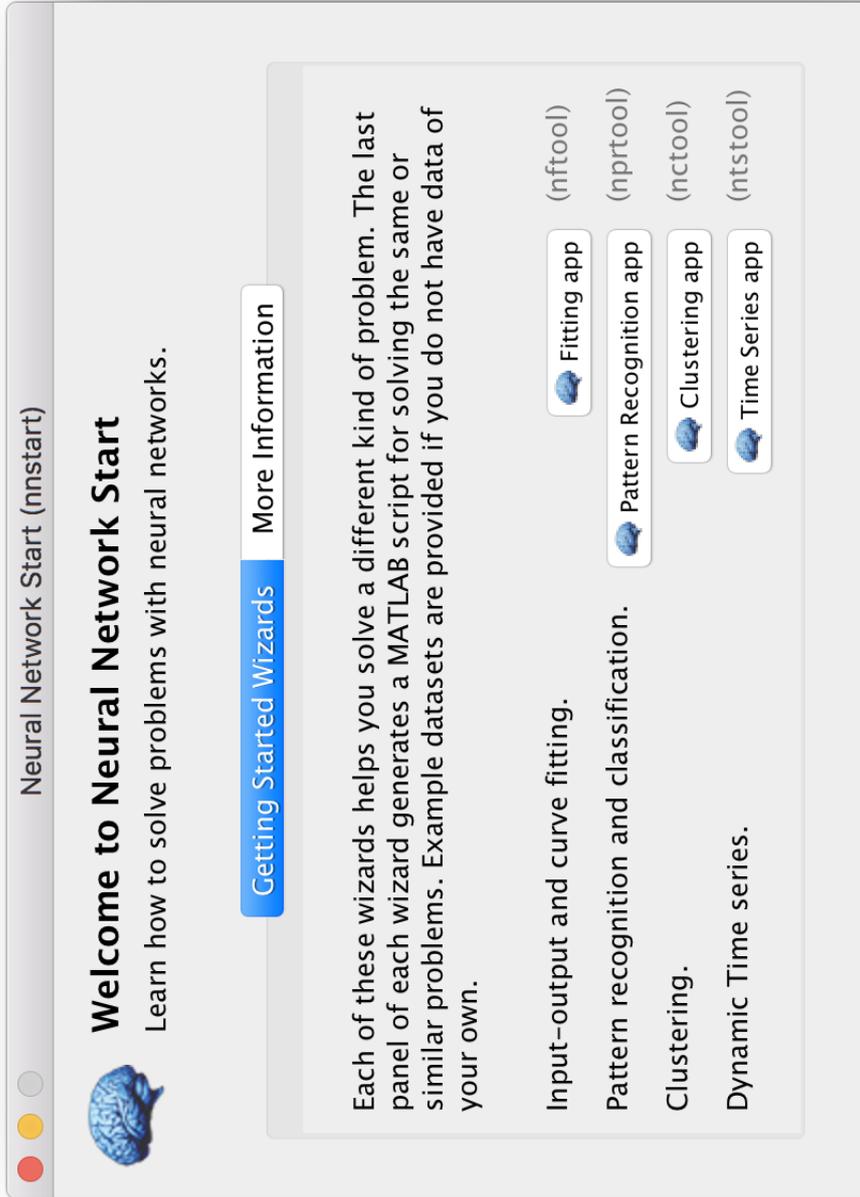


Figura 4.3: Neural Network Start

Gli ingressi vengono prelevati dal programma Matlab che è stato implementato a partire da SimPlateImpacts, l'uscita desiderata invece deriva dal programma Matlab *my_out* riportato in appendice, listing A.2. L'input della rete neurale può essere un valore singolo o un vettore, in questo caso l'input è rappresentato dagli angoli, captati dai sensori sulla piastra, relativi a ogni singolo impatto. Facendo riferimento al codice ?? i dati di input vengono generati da SimPlateImpacts e assegnati alla matrice a .

Anche l'output della rete neurale (target) è un vettore bidimensionale (Z) A con lo stesso numero di righe della matrice di input formato da 0 e 1 dove ogni 1 localizza un impatto.

2. **Validation and test data:** la rete considera il 70% dei campioni totali per la parte di training, ma è possibile scegliere le percentuali da dedicare alle fasi di *validation* e *testing*. Come si vede in Figura 4.5 per questo esperimento sono state scelte, e mantenute, valori del 15% per le fasi di Validation e Testing, in modo da avere lo stesso quantitativo di dati per le due fasi.
3. **Network architecture:** dalla schermata visibile in Figura 4.6 si può scegliere il numero di neuroni da impiegare negli strati nascosti della rete neurale con un minimo obbligatorio di 1. Come mostrato nella figura in basso la struttura della rete neurale è costituita da due strati uno dei quali nascosto. Questo pannello di controllo sarà utile per modificare i parametri della rete neurale durante la valutazione dei dati.
4. **Train network:** una volta impostati tutti i parametri attraverso

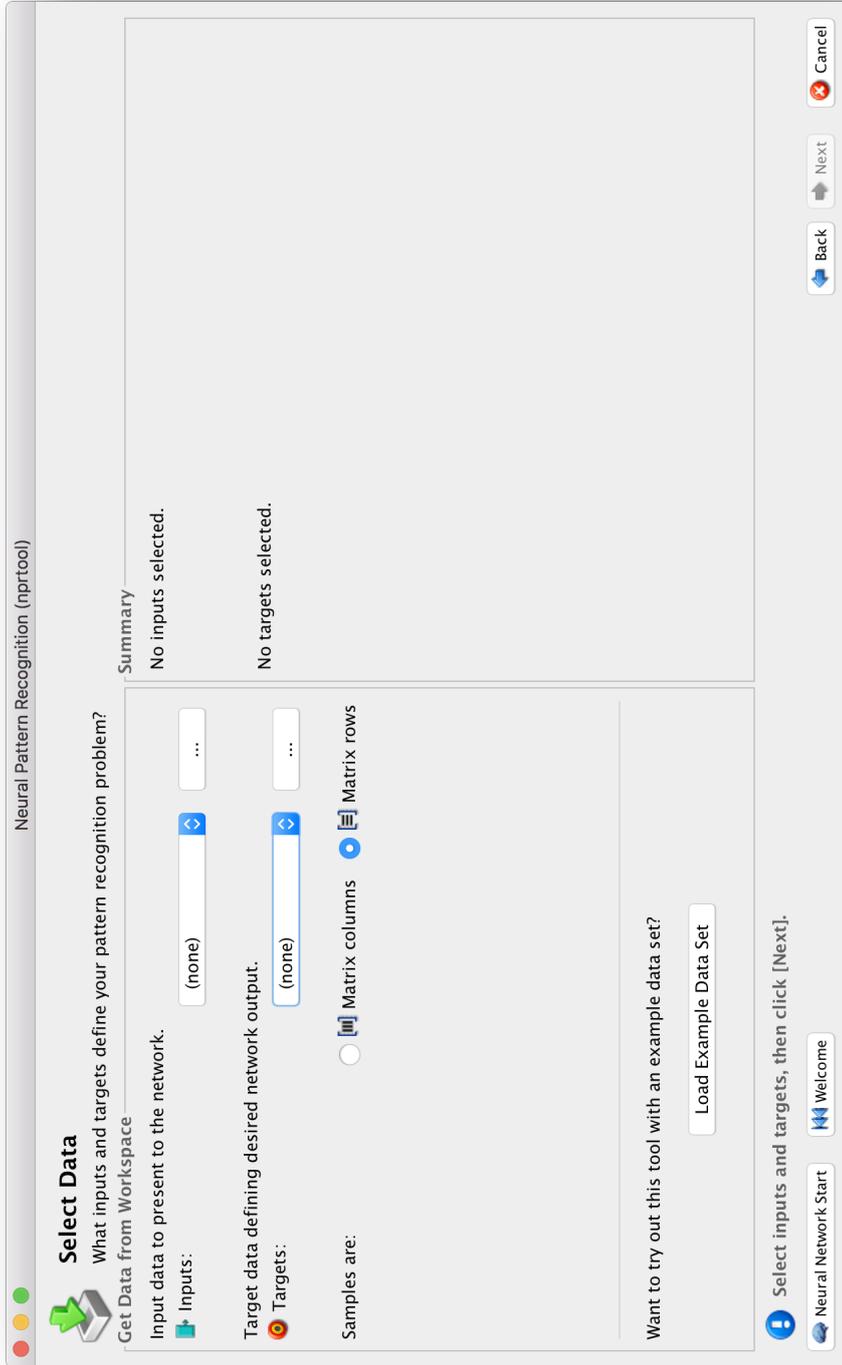


Figura 4.4: Schermata di selezione dei dati

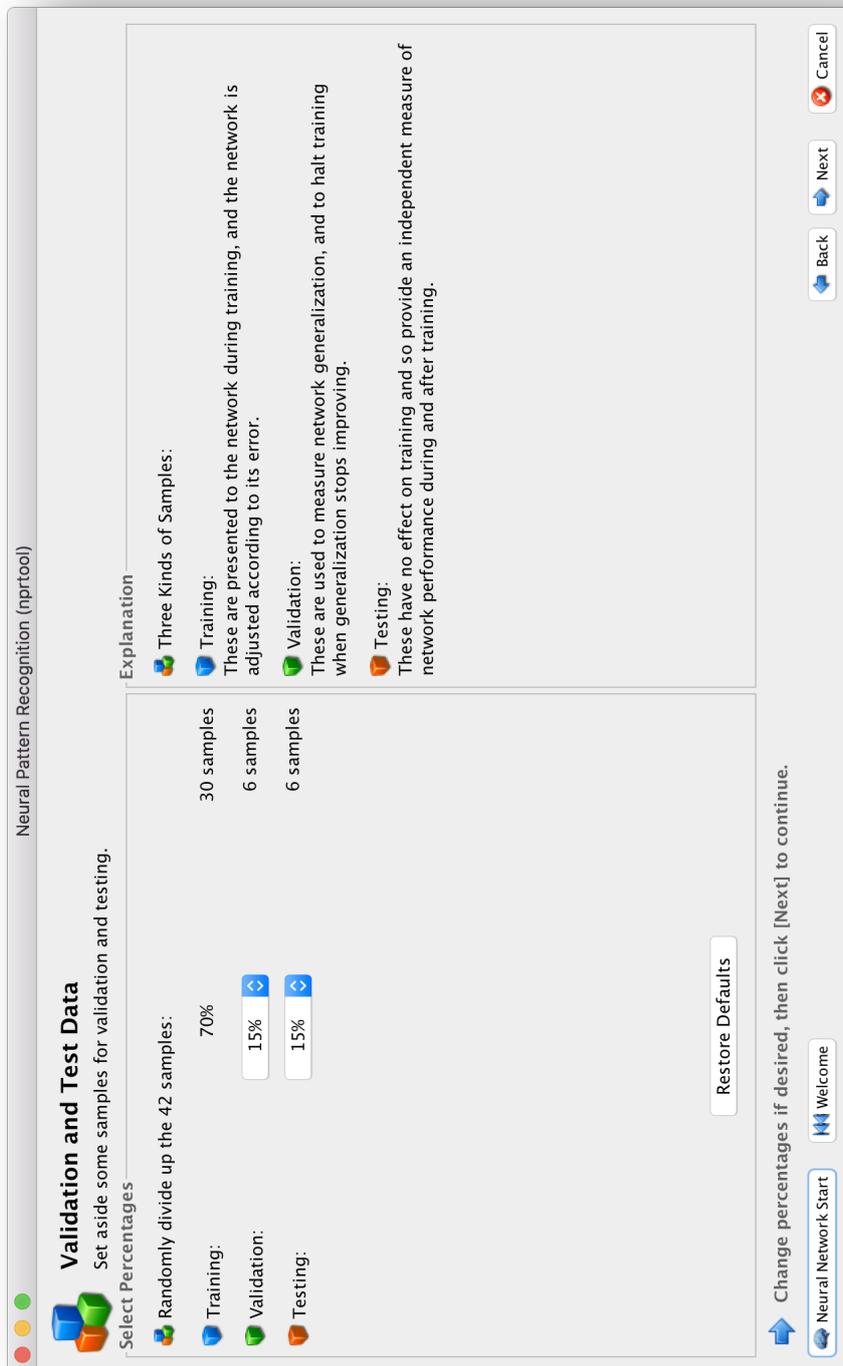


Figura 4.5: Schermata di convalida e prova dei dati

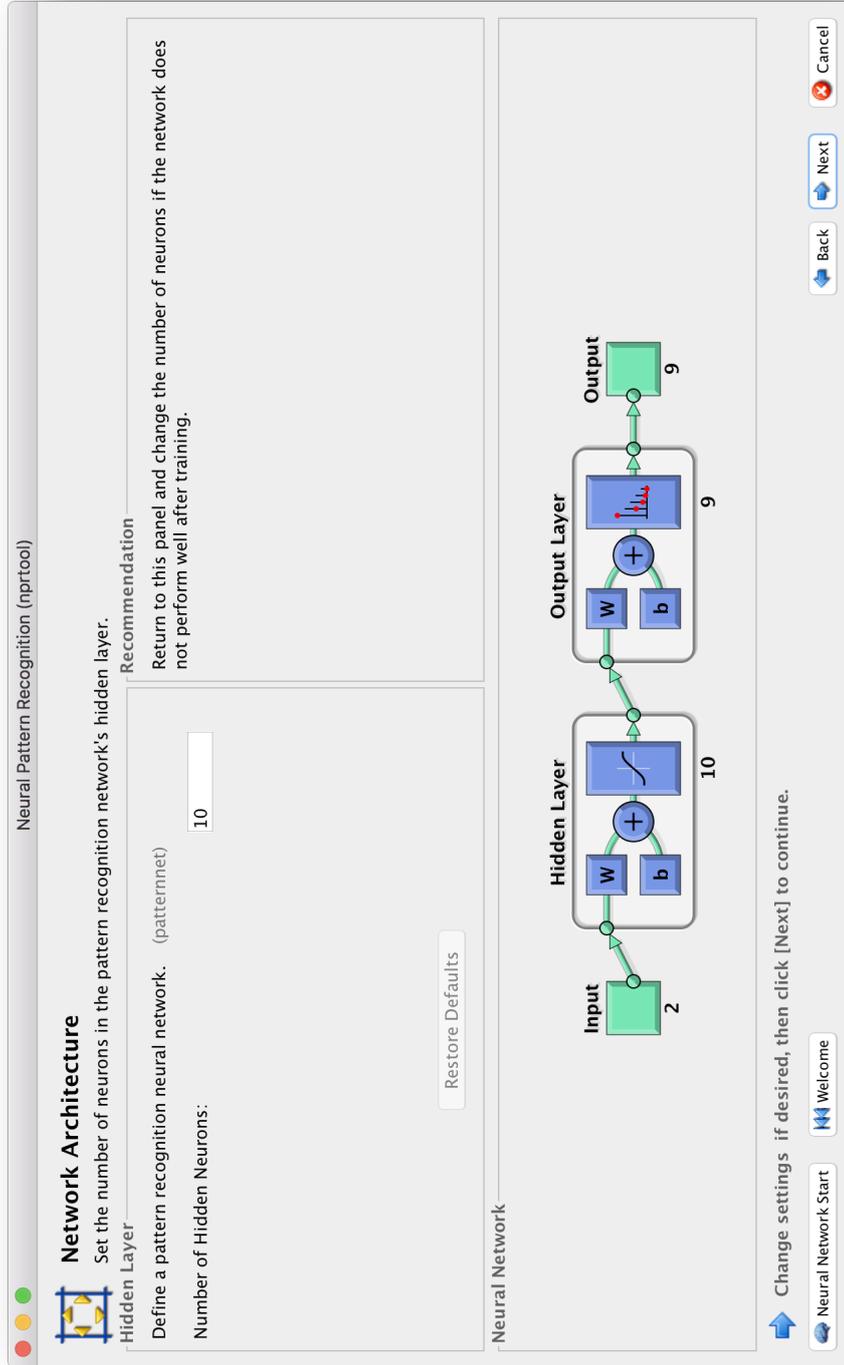


Figura 4.6: Schermata per l'architettura della rete

il tasto *train* nella schermata mostrata in Figura 4.7 è possibile fare iniziare lo studio dei dati da parte della rete. In base alla quantità di dati da elaborare la rete impiegherà più o meno tempo, una volta finito l'apprendimento verrà mostrata la schermata mostrata in Figura 4.8 che visualizza il riepilogo della rete neurale implementata: in alto si può vedere l'architettura, sotto i dati relativi all'algoritmo poi il processo di elaborazione e altri dati. Tornando alla schermata in Figura 4.7 dopo il termine dell'elaborazione dei dati compariranno gli errori relativi a ogni fase di sviluppo della rete. Il mio obiettivo era quello di portare il valore dell'errore a un numero accettabile.

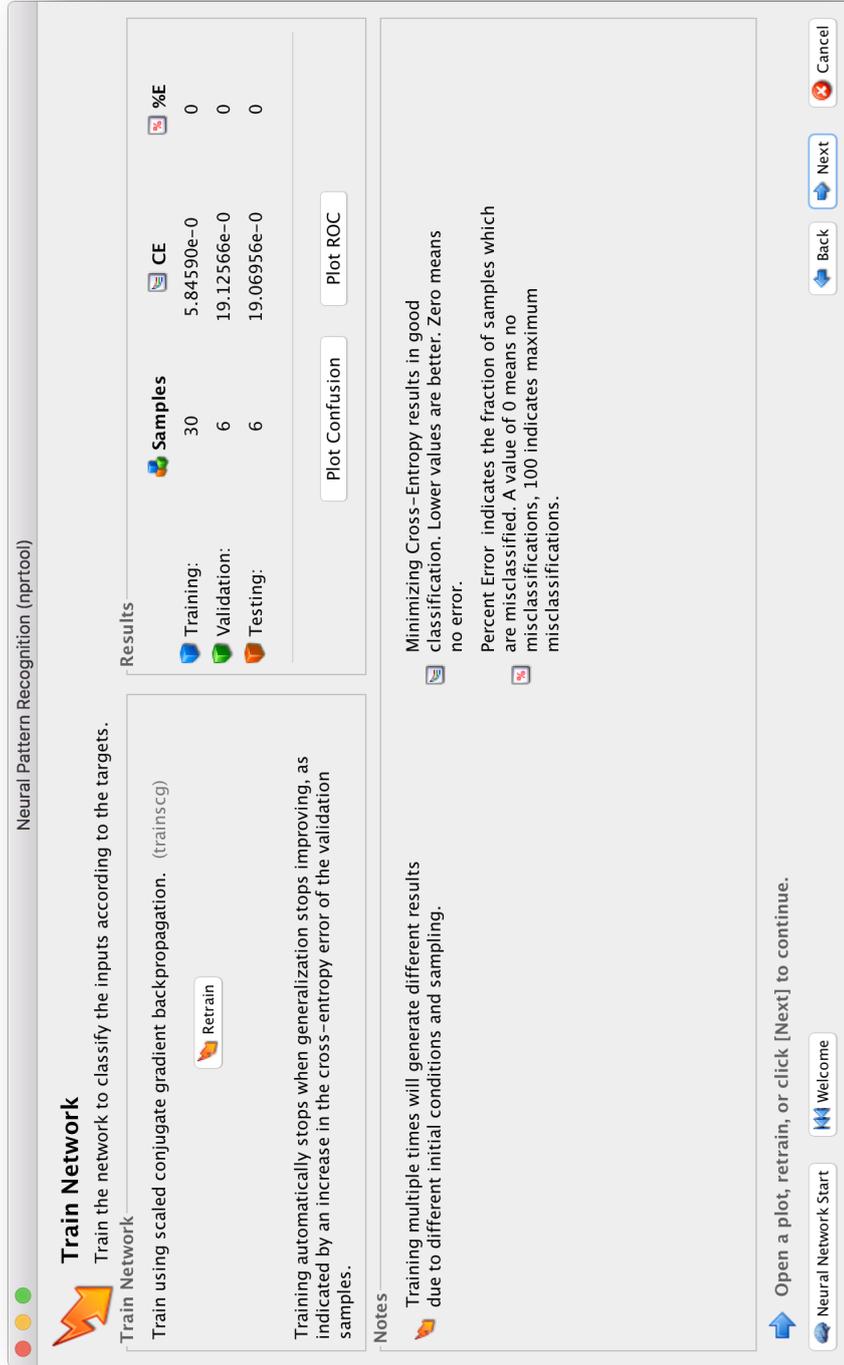


Figura 4.7: Schermata dei risultati

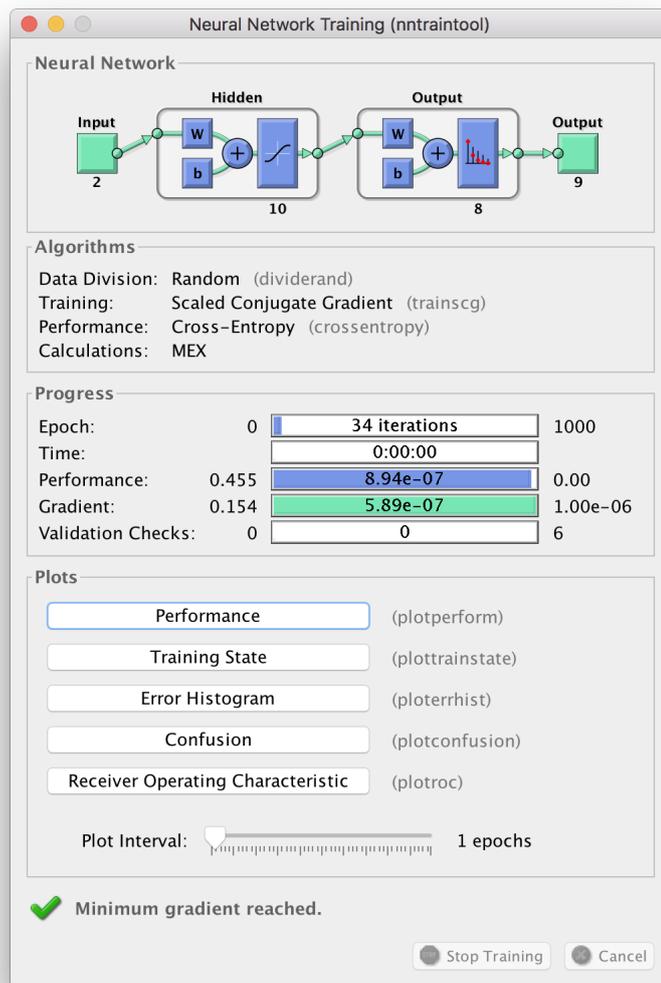


Figura 4.8: Schermata di elaborazione dei dati

Capitolo 5

Risultati sperimentali

In questo capitolo si andrà a descrivere in ogni sua fase l'esperienza concreta di laboratorio, verranno descritte le motivazioni e le osservazioni delle modifiche apportate tra un passaggio e il successivo fino ad arrivare alla rete neurale finale volta a classificare impatti su una lastra piana.

L'esperienza di laboratorio si è divisa in varie fasi: generazione dei dati, elaborazione dei dati e implementazione della rete neurale. Alcune prove erano state stabilite in partenza come la divisione dell'area in sotto aree attraverso l'uso di una griglia; altre invece sono state sviluppate in itinere come aumentare la densità della griglia e degli impatti. L'algoritmo iniziale su cui si è basato l'esperimento è quello descritto nel capitolo precedente 4.2 e riportato in appendice A. I parametri su cui abbiamo agito per migliorare la rete neurale sono stati: il numero degli impatti e la loro distribuzione all'interno dell'area e il numero delle sotto aree della griglia. Nelle prove che andrò a descrivere di seguito sono stati variati questi parametri e sono state fatte delle valutazioni

per analizzare gli effetti dei cambiamenti sulla rete neurale.

Oltre ai parametri in ingresso alla rete sono state apportate delle modifiche al numero di neuroni nello strato nascosto che la rete neurale utilizza per l'apprendimento facendoli variare all'interno di un range di valori che va da 10 a 500.

I casi analizzati partono da dati che rispecchiano il caso ideale; *SimPlateImpacts* genera angoli ideali ovvero quando le rette dei sensori si intersecano tutte e quattro esattamente sul punto dell'impatto.

Prova I

Nell'algoritmo *defines* l'area compresa tra i 4 sensori è stata suddivisa da una griglia 3x3 in 9 classi che nel codice *defines* in appendice A, listing A.3, sono riportate con il nome A_i . Gli impatti considerati 42 (corrispondenti ai vertici delle aree piccole che formano il rettangolo giallo mostrato in Figura 4.2). Di seguito viene riportata l'immagine visualizzata da *SimPlateImpacts* nel caso che sto descrivendo.

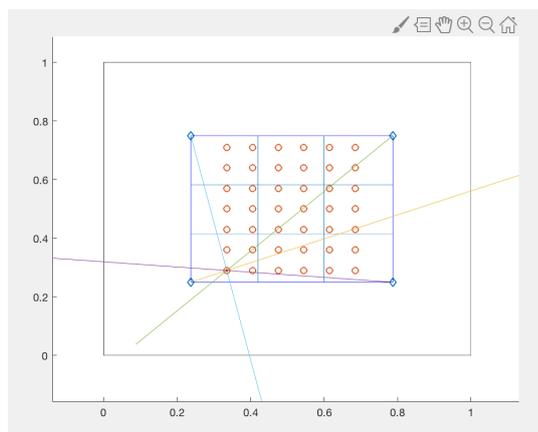


Figura 5.1: Outoput 1 SimPlateImpacts

L'uscita di *SimPlateImpacts* in questo caso fornisce una matrice di angoli (a) delle dimensioni 42x4 dove ogni riga della matrice riporta il valore dei 4 angoli captati dai sensori secondo la convenzione precisata al capitolo precedente. Nella figura 5.1 gli angoli sono rappresentati dalle linee che partono dai sensori e vanno a intersecarsi esattamente sul punto dell'impatto. Le coordinate degli impatti (iX) sono state utilizzate per il ciclo *for* all'interno dell'algoritmo *valuta* con lo scopo di valutare in quale delle 9 aree cadesse ogni singolo impatto.

Le aree della griglia sono associate a un vettore M di 9 elementi inizialmente tutti posti a 0. Lo script di post processing degli output inserisce nel vettore un 1 a seconda che l'impatto sia relativo alla classe di indice corrispondente.

Come ultimo passo è stato creato l'algoritmo *my_out*: esso crea la matrice di uscita Z di k righe formate dal vettore M corrispondente al k-esimo impatto. In questo modo le matrici *a* e Z hanno lo stesso numero di righe e posso essere poste rispettivamente come inputs e target nella rete neurale, dove vengono associate, appunto, per righe.

Una volta assegnati i valori che la rete neurale dovrà elaborare, è stato impostato il numero di neuroni a 10 utilizzando la schermata—, incrementandoli dopo ogni test fino ad arrivare ad utilizzarne 500.

Una volta impostati questi parametri la rete neurale è pronta a iniziare la così detta fase di apprendimento. Dopo aver studiato i dati la rete mostra come sono stati valutati, quanto tempo è stato impiegato e infine l'errore commesso durante le tre fasi (Training, Valuation e Testing).

Si è notato, come ci si aspettava, che aumentando il numero di neuroni

nello strato nascosto l'errore in uscita diminuisce.

Osservazioni: In questa prova abbiamo ottenuto un errore dello 0%. Questo risultato ottimale è stato possibile perchè gli impatti sono pochi e sono localizzati perfettamente dentro le aree, non ci sono impatti sui confini tra un'area e l'altra. Inoltre sempre per l'esiguo numero di dati la rete è molto veloce ad elaborare la risposta anche aumentando il numero di neuroni impiegati.

Prova II

Per raffinare la rete neurale abbiamo agito sul quantitativo di impatti (modificando il codice di *SimPlateImpacts*) passando da 42 a 396, mantenendo la divisione dell'area in 9 sottoaree.

In questo caso si è notato che alcuni degli impatti si trovano molto vicino ai confini tra le aree (come si può vedere in Figura 5.2 sotto).

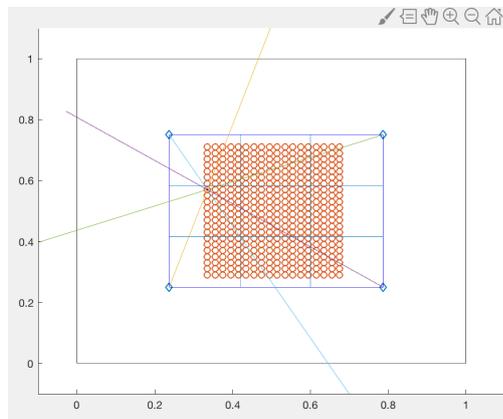


Figura 5.2: Outoput 2 SimPlateImpacts

Come conseguenza sono state apportate modifiche agli algoritmi *valuta*, *defines*, *my_out* per far rispettare a tutti i parametri le nuove dimensioni. L'input della rete neurale è la nuova matrice di angoli (a)

396x4 mentre la matrice $Z2$ è diventata il nuovo target.

Osservazioni: Il tempo di elaborazione della rete neurale non ha subito grosse modifiche così come l'errore in uscita non si è discostato di molto dallo 0%.

Prova III

Variando il numero di neuroni mantenendo la stessa divisione dell'area totale ci siamo resi conto che non si sono sviluppati grossi cambiamenti. Abbiamo deciso quindi di agire sulle aree passando a una griglia 5x5 a cui corrispondono aree circa quadrate di lato 10 cm.

Il primo test è stato fatto su il numero di impatti iniziale, ovvero 42. Si può vedere l'output di *SmPlateImpacs* nella figura di seguito:

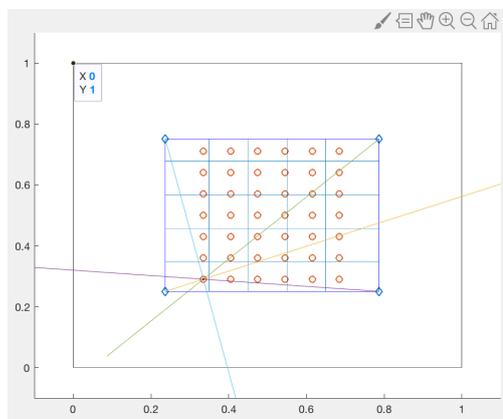


Figura 5.3: Outoput 3 SimPlateImpacts

Il risultato, intorno allo 0% è accettabile sempre per il fatto che gli impatti sono localizzati dentro le aree e non ci sono zone che creano dubbi di locazione.

Si è deciso quindi di mantenere questa nuova divisione di aree e di far

crece progressivamente il numero degli impatti; si è passati da 42 a 24249 impatti dentro il perimetro dei sensori. Ogni volta che variavano il numero degli impatti si sono svolte più prove variando il numero di neuroni da 10 a 500 per ogni caso.

Tabella 1

Numero impatti	Numero neuroni	Errore medio
567	10	13,5%
567	50	11%
567	100	6,5%
567	150	9%
567	200	8%
567	300	13,5%
567	500	38,5%
1008	10	7,5%
1008	50	7%
1008	100	8%
1008	150	7%
1008	200	6,5%
1008	300	7,5%
1008	500	10%
2268	10	8,5%
2268	50	7,5%
2268	100	7,5%
2268	150	8%
2268	200	8%
2268	300	10%
2268	500	23,5%

Tabella 1		
Numero impatti	Numero neuroni	Errore medio
3484	10	6%
3484	50	6%
3484	100	6%
3484	150	6,5%
3484	200	6,5%
3484	300	5,5%
3484	500	13,5%
8881	10	7%
8881	50	7%
8881	100	7%
8881	150	6,5%
8881	200	6,5%
8881	300	7%
8881	500	9,5%
24249	10	6,5%
24249	50	6%
24249	100	6%
24249	150	6%
24249	200	6%
24249	300	6%
24249	500	6,5%

Nella tabella sopra sono stati riportati alcuni dati: nella colonna di sinistra viene riportato il numero di impatti sull'area considerata, nella colonna al centro il numero di neuroni nello strato nascosto della rete neurale e nella colonna a destra l'errore medio delle 3 fasi.

Come si può notare nel caso con 567 impatti la curva di apprendimento ha un errore che varia prima in modo decrescente poi crescente. Nella prima metà la rete impara fino ad arrivare al suo massimo di apprendimento, corrispondente all'errore minimo, dopo di che la rete va in overfitting e l'errore cresce nuovamente.

Questa curva è sempre meno visibile, al netto dei test eseguiti, aumentando il numero degli impatti fino ad arrivare al caso i 24249 impatti in cui l'errore ha l'andamento di una retta.

Si è notato inoltre che ampliando il numero di dati la rete ha la possibilità di apprendere su più esempi e diventa sempre più affidabile ma questo comporta tempistiche di elaborazione più lunghe.

Osservazioni:La rete neurale produce risultati migliori se ha un numero elevato di esempi da cui studiare e apprendere. L'affidabilità della rete tuttavia va a discapito dei tempi di risposta.

Abbiamo notato infine che, in alcuni casi, aumentando troppo il numero di neuroni, con un numero elevato di inputs la rete va in overfitting passando da un errore basso a un errore molto alto perchè incapace di elaborare quel numero di dati (si veda la Tabella 1 nell'Appendice A nel caso di 2268 impatti).

Conclusioni

Nel presente elaborato è stato possibile realizzare una rete neurale per la classificazione di impatti partendo dalla conoscenza degli angoli di incidenza delle onde (causate dagli urti) sui sensori. Questa rete neurale è quindi in grado di emulare il lavoro svolto del tatto nel sistema somatosensoriale biologico.

Con l'aiuto di Matlab è stato possibile creare un algoritmo che simula una serie di impatti sulla lastra di alluminio. Questi impatti vengono ricevuti dai sensori con diverse tempistiche, a seconda della distanza dell'impatto dai sensori, e con determinate angolazioni ed energie. Per mezzo di una serie di algoritmi realizzati sempre nell'ambiente Matlab i dati ricevuti dai sensori sono stati raccolti, valutati ed elaborati in modo da ottenere una localizzazione di questi all'interno di un area prestabilita.

Dopo aver classificato gli impatti e aver creato la matrice di uscita delle aree, si è passati alla messa a punto della rete neurale vera e propria. I valori in ingresso della rete sono gli angoli generati da *SimPlateImpacts*, contenuti nella matrice \mathbf{a} , l'obiettivo è rappresentato dalla matrice \mathbf{Z} (ottenuta da *my_out* che mostra in quale area è avvenuto l'urto. Agendo su questi parametri sono state fatte diverse

prove che hanno portato al risultato finale. La rete neurale implementata si è rivelata molto affidabile, mostrando un errore pari al 6% in tutte le fasi di funzionamento della rete (training, validation e testing). Per arrivare a questo risultato si è rivelata sufficiente una rete neurale a due strati: uno nascosto e uno di output.

I risultati ottenuti con la rete neurale descritta in questo elaborato di tesi possono essere utilizzati in diversi ambienti. Nel campo delle protesi meccaniche potrebbe essere utile per arrivare a creare un arto artificiale in grado di trasmettere le sensazioni che derivano dal contatto con l'ambiente esterno come l'aria sulla pelle o il calore del sole. Questo porterebbe a protesi di ultima generazione e contribuirebbe a spingere le macchine ad assomigliare sempre più all'uomo.

Un altro ambiente d'interesse potrebbe essere quello industriale dove sarebbe possibile utilizzare l'algoritmo qui descritto per la localizzazione di danni su superfici anche molto grandi con il vantaggio di non dover controllare una vasta area passo per passo e guadagnando tempo e fatica.

Appendice A

In questa sezione sono riportati tutti i codici sviluppati in ambiente Matlab durante il lavoro di tesi descritto nell'elaborato.

Listing A.1: simPlateImpacts

```
function simPlateImpacts

defines;

% Prepare output
Ni = size(iX,1);
Ns = size(sX,1);
t = zeros(Ni,Ns);
a = zeros(Ni,Ns);
e = zeros(Ni,Ns);

% Simulate impacts
for idxI = 1:size(iX,1)
    I = iX(idxI,:);
    for idxS = 1:size(sX,1)
        S = sX(idxS,:);
```

```

% Livello 0: ideale
[t(idxI,idxS),a(idxI,idxS),e(idxI,idxS)] =
simulate_impact(I,S,idxI,[],0);

% Livello 1: realistico, senza riflessioni
[t(idxI,idxS),a(idxI,idxS),e(idxI,idxS)] =
simulate_impact(I,S,idxI,[],1);

% Livello 2: realistico, con riflessioni
[t(idxI,idxS),a(idxI,idxS),e(idxI,idxS)] =
simulate_impact(I,S,idxI,[],2,[],geo,1);
end
end

% Save simulation output
save impactsSim_Plate sX iX t a e

% Plot outputs
hf = figure;
s.new = false;
set(hf, 'UserData', s);
set(hf, 'KeyPressFcn', @KeyPressFcn);

%disegna gli assi
ha = axes('nextPlot','Add');
%perimetro della lastra
plot(ha,pX([end 1:end],1),pX([end 1:end],2),'k')
%sensori
plot(ha,sX(:,1),sX(:,2),'d')
%perimetro dei sensori
plot(ha,sX([end 1:end],1),sX([end 1:end],2),'b')
%rappresenta gli impatti

```

```

plot (ha, iX(:,1), iX(:,2), 'o')

%griglia interna ai sensori
y= linspace (0.25,0.75);
w= 0.42*ones (size (y));
line (w, y)
z= 0.6*ones (size (y));
line (z, y)

x=linspace (0.237,0.787);
k=0.416*ones (size (x));
line (x, k);
j=0.582*ones (size (x));
line (x, j);

idxI = 1;
doDraw = true;
hl = zeros (1, Ns+1);
while (true)
    drawnow

    if doDraw
        for idxS = 1:Ns
            if hl (idxS), delete (hl (idxS)); end
            x0 = sX (idxS, :);
            x1 = x0 + [cosd (a (idxI, idxS)+90)
                sind (a (idxI, idxS)+90)];
            hl (idxS) = plot (ha, [x0 (1) x1 (1)], [x0 (2)
                x1 (2)]);
        end
        if hl (idxS+1), delete (hl (idxS+1)); end
        hl (idxS+1) = plot (ha, iX (idxI, 1),

```

```

        iX(idxI,2),'.');
        doDraw = false;

        set(ha,'XLim',[-0.1 1.1],'YLim',[-0.1 1.1])
    end

    s = get(hf,'UserData');
    if(s.new)
        s.new = false;
        switch(double(s.key))
            case 27 % escape
                break
            case 28 % left arrow
                idxI = idxI-1;
                if idxI<1, idxI = Ni; end
                doDraw = true;
            case 29 % right arrow
                idxI = idxI+1;
                if idxI>Ni, idxI = 1; end
                doDraw = true;
        end
    end
    set(hf, 'UserData', s);
end

end

function KeyPressFcn(src,evt)
hf = ancestor(src,'figure');
s.key = get(hf,'CurrentCharacter');
s.mod = get(hf,'CurrentModifier');
s.new = true;

```

```
set (hf, 'UserData', s);  
end
```

Listing A.2: my_out

```
defines;  
%vettore rappresentativo delle aree della griglia  
M=zeros(1,9);  
  
%la matrice di uscita ha su ogni riga il vettore M  
corrispondente al  
%k-esimo impatto  
for k=1:42  
    Z(k, :)=M;  
end  
  
%seleziono la matrice delle coordinate  
load coordinate42  
%indice per scorrere le righe(i) e le colonne(j)  
della matrice coordinate  
i=0;  
j=0;  
%indice per la matrice finale Z  
k=1;  
%dimensione della matrice coordinate  
[R,C]=size(coordinate42);  
%ciclo per capire quale area viene colpita  
for i=1:R  
    x = coordinate42(i,1)  
    y = coordinate42(i,2)  
  
    M=valuta(x,y);  
    Z(k, :)=M;
```

```
        k= k +1;  
end
```

Listing A.3: defines

```
% Plate geometry [m]  
pX = [0 0;  
      1 0;  
      1 1;  
      0 1];  
  
geo(1).xy1 = pX(1,:); geo(1).xy2 = pX(2,:);  
geo(2).xy1 = pX(2,:); geo(2).xy2 = pX(3,:);  
geo(3).xy1 = pX(3,:); geo(3).xy2 = pX(4,:);  
geo(4).xy1 = pX(4,:); geo(4).xy2 = pX(1,:);  
  
% Sensor positions [m]  
sX = [0.237 0.25;  
      0.787 0.25;  
      0.787 0.75;  
      0.237 0.75];  
  
% Impact locations [m]  
x = 0.335 + (0:2:35)/100;  
% y = 0.15 + (0:3:70)/100;  
y = 0.15 + (14:2:56)/100;  
[X,Y] = meshgrid(x,y);  
iX = [X(:) Y(:)];  
  
% definisco i vertici che delimitano le aree  
(da s1 a s16)  
field1='x';
```

```

field2='y';
value1=[0.237,0.42,0.603,0.787];
value2=[0.75,0.584,0.418,0.25];

s=struct(field1,value1,field2,value2)

s1.x=0.237  s1.y=0.75  s2.x=0.42  s2.y=0.75  s3.x=0.603
s3.y=0.75  s4.x=0.787  s4.y=0.75  s5.x=0.237  s5.y=0.584
s6.x=0.42  s6.y=0.584  s7.x=0.603  s7.y=0.584  s8.x=0.787
s8.y=0.584  s9.x=0.237  s9.y=0.418  s10.x=0.42  s10.y=0.418
s11.x=0.603  s11.y=0.418  s12.x=0.787  s12.y=0.418  s13.x=0.237
s13.y=0.25  s14.x=0.42  s14.y=0.25  s15.x=0.603  s15.y=0.25
s16.x=0.787  s16.y=0.25

%definisco le aree della griglia
% vertice in alto a sinistra e vertice
in basso a destra
field1='up';
field2='down';
value1=[s1,s2,s3,s5,s6,s7,s9,s10,s11,];
value2=[s6,s7,s8,s10,s11,s12,s14,s15,s16];

A=struct(field1,value1,field2,value2)

A1.up=s1      A1.down=s6  A2.up=s2      A2.down=s7  A3.up=s3
A3.down=s8  A4.up=s5      A4.down=s10  A5.up=s6      A5.down=s11
A6.up=s7      A6.down=s12  A7.up=s9      A7.down=s14  A8.up=s10
A8.down=s15  A9.up=s11      A9.down=s16

save('punti_aree.mat','s1','s2','s3','s4','s5',
's6','s7','s8','s9','s10','s11','s12','s13','s14',
's15','s16');

```

Listing A.4: valuta

```
function [M] = valuta (x,y)
load('punti_aree.mat');

M=zeros(1,9);
if ((y<=s1.y) & (y>s5.y))
    if((x>=s1.x) & (x<s2.x))
        M(1,1)=1;
    elseif ((x>=s2.x) & (x<s3.x))
        M(1,2)=1;
    elseif ((x>=s3.x) & (x<=s4.x))
        M(1,3)=1;
    end
end

if ((y<=s5.y) & (y>s9.y))
    if((x>=s1.x) & (x<s2.x))
        M(1,4)=1;
    elseif ((x>=s2.x) & (x<s3.x))
        M(1,5)=1;
    elseif ((x>=s3.x) & (x<=s4.x))
        M(1,6)=1;
    end
end

if ((y<=s9.y) & (y>=s13.y))
    if((x>=s1.x) & (x<s2.x))
        M(1,7)=1;
    elseif ((x>=s2.x) & (x<s3.x))
        M(1,8)=1;
    elseif ((x>=s3.x) & (x<=s4.x))
```

```
        M(1,8)=1;  
    end  
end
```


Bibliografia

- [1] Mark Hudson Beale, Martin T Hagan, and Howard B Demuth. Neural network toolboxTM user's guide. *The Mathworks Inc*, 1992.
- [2] Robert M Berne, Matthew N Levy, and Enrique LM Ochoa. *Fisiologia*. Panamericana, 1987.
- [3] Dario Floreano and Claudio Mattiussi. Manuale sulle reti neurali. Technical report, Il mulino, 2002.
- [4] Raffaele Simone. *Treccani: dizionario della lingua italiana*. Istituto italiano della enciclopedia italiana; UTET, 2003.
- [5] Nicola Testoni, Luca De Marchi, and Alessandro Marzani. A stamp size, 40ma, 5 grams sensor node for impact detection and location. In *European Workshop on SHM*, 2016.

Ringraziamenti

Vorrei occupare questo spazio per ringraziare tutte le persone che mi hanno seguito e sostenuto in questo periodo e in tutti gli anni di università.

Per primo vorrei ringraziare il prof. Testoni, relatore di questa tesi, per avermi proposto questo progetto che mi ha affascinato dal primo momento, ma soprattutto per la sua infinita disponibilità e pazienza in questo periodo. Senza di lei non avrei mai scoperto il mio interesse per l'intelligenza artificiale!

Un grazie a Cristiano Aguzzi che mi ha seguito pazientemente nella progettazione.

Un grande ringraziamento va alla mia famiglia che mi ha sempre sostenuto e motivato durante il mio percorso. Mi rendo conto di come la fiducia che riponete ogni giorno in me sia fondamentale per non mollare mai.

Un grazie speciale a mio fratello che è sempre riuscito a tenermi il

morale alto e a spronarmi quando nei momenti in cui ne avevo più bisogno. Sei un esempio di vita per me.

Ringrazio Riccardo che mi ha supportato ma soprattutto sopportato più di ogni altro in questi anni. Mi hai trasmesso un'energia indescribibile, riesci a tirare fuori il meglio di me in ogni situazione e grazie a questo sono riuscita ad arrivare a questo enorme traguardo.

Ringrazio infine tutti i miei amici (ma proprio tutti!), hanno avuto un peso determinante nel completamento di questo percorso, condividendo con me momenti belli e brutti senza mai voltarmi le spalle. In particolare devo dire grazie a Francesca, compagna di studi dalle elementari, sostegno fondamentale durante questi anni di studio.