

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Dipartimento
di Fisica
e Astronomia
Galileo Galilei

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Corso di Laurea in Fisica

Confronto di reti neurali ricorrenti addestrate con algoritmi standard e biologicamente plausibili

Relatore

Dott. Michele Allegra

Laureanda

Angela Bortolato

ANNO ACCADEMICO 2023-2024

Sommario

Nella letteratura neuroscientifica, lo studio delle dinamiche che regolano l'apprendimento attraverso metodi di intelligenza artificiale basati sulle reti neurali ha assunto sempre maggiore importanza. Nel caso di processi cognitivi che richiedono informazioni sul contesto, come memoria, processi decisionali e controllo motorio, le reti neurali ricorrenti (RNN) sono uno strumento versatile per le neuroscienze, in quanto forniscono un modello semplificato dei cluster di neuroni biologici. L'algoritmo standard con cui vengono allenate le RNN è la discesa del gradiente, implementato mediante *backpropagation*, un metodo efficiente ma non plausibile dal punto di vista biologico. Pertanto, sono stati proposti altri algoritmi che si basano su esplorazioni casuali e ricompense. In questa tesi, si allena una RNN a svolgere un semplice compito decisionale "delayed nonmatch-to-sample" con un metodo biologicamente plausibile; si analizzano gli effetti dell'addestramento sulla dinamica della rete risultante dal *training*.

Indice

1	Introduzione	1
2	Modelli e metodi	3
2.1	Reti neurali artificiali e ricorrenti	3
2.2	Metodi di addestramento	5
2.2.1	Apprendimento supervisionato	5
2.2.2	Apprendimento Hebbiano	6
2.2.3	<i>Node-perturbation</i>	6
2.2.4	Metodo Hebbiano-Esplorativo	7
2.2.5	Metodo di apprendimento proposto da Miconi	8
3	Struttura della rete neurale	11
3.1	Implementazione numerica	11
3.2	Delayed nonmatch-to-sample task	12
4	Risultati	15
5	Conclusioni	21
	Bibliografia	23

Capitolo 1

Introduzione

Negli ultimi decenni, le neuroscienze hanno compiuto significativi progressi nella comprensione dei meccanismi neurali sottostanti al funzionamento del cervello umano, ciò è stato reso possibile anche grazie ai contributi di metodi di intelligenza artificiale. Uno degli obiettivi principali di questo campo è sviluppare modelli computazionali che non solo riproducano il comportamento osservato dei sistemi neurali, ma che siano anche biologicamente plausibili. In particolare, lo studio dei metodi di addestramento per reti neurali artificiali ispirati ai processi biologici è diventato un argomento di grande interesse, poiché può fornire nuove intuizioni su come il cervello apprende e si adatta. Tra queste reti, le reti neurali ricorrenti (RNN) costituiscono una classe fondamentale per modellare sequenze temporali: presentano connessioni ricorrenti tra i neuroni, caratteristica essenziale della struttura della corteccia cerebrale, che permettono alla RNN di riprodurre le capacità cognitive, come la memoria, controllo motorio o processi decisionali [1] [2]. Inoltre, recenti studi [3] evidenziano che le rappresentazioni neurali sono altamente dinamiche, aspetto replicabile con modelli basati su RNN fortemente connesse che operano nel regime quasi caotico.

Nonostante i numerosi progressi nella progettazione di algoritmi di apprendimento supervisionato e non supervisionato, molti dei meccanismi di apprendimento che possono essere applicati alle reti neurali artificiali non rispecchiano i processi di apprendimento naturale osservati nei cervelli biologici. Questo crea una discrepanza significativa tra i modelli computazionali e i sistemi neurali reali. Una realizzazione biologica di un meccanismo di apprendimento deve rispettare la località delle informazioni necessarie alla sinapsi, la non simmetria delle connessioni e l'assenza di supervisione in gran parte del processo [4].

La letteratura esistente suggerisce diversi approcci per l'apprendimento, come l'apprendimento supervisionato basato sulla retropropagazione del segnale, nella versione del *backpropagation-through-time* (BPTT) [5], l'apprendimento per rinforzo [6], a cui si ispira il *node-perturbation* [7], e l'apprendimento Hebbiano modulato da ricompense, nella versione E-

H [8] [9]. Tuttavia, queste tecniche presentano limitazioni in termini di complessità computazionale e precisione biologica. La presente tesi si propone di esplorare un metodo di addestramento, proposto da Miconi [10], con l'obiettivo di combinare l'efficacia computazionale con la plausibilità biologica, offrendo una soluzione più vicina ai processi naturali di apprendimento: si basa sui principi dell'apprendimento Hebbiano modulato da un segnale di ricompensa sporadico e in ritardo rispetto agli stimoli, come in gran parte degli esempi di apprendimento animale.

L'obiettivo di questa ricerca è simulare la dinamica di una rete neurale ricorrente completamente connessa, addestrata con il metodo di apprendimento proposto da Miconi a svolgere un compito decisionale che dipende dal contesto (*"Delayed nonmatch-to-sample"*), e valutare le prestazioni di questo algoritmo in termini di efficacia, efficienza e plausibilità biologica.

Nel Capitolo 2, si presentano le caratteristiche principali delle reti neurali artificiali, in particolare delle reti neurali ricorrenti, con la rispettiva descrizione matematica della dinamica. Si descrivono poi i vari metodi di addestramento presenti in letteratura, incluso il metodo di Miconi preso in esame.

Il Capitolo 3 si occupa di descrivere la struttura della rete simulata in questa tesi: l'implementazione numerica della dinamica e dell'algoritmo di apprendimento; inoltre, viene illustrato il task *"Delayed nonmatch-to-sample"*: le principali caratteristiche e come viene codificato l'input e l'output.

Nel Capitolo 4, sono discussi i risultati delle simulazioni computazionali: gli effetti dell'addestramento sulle connessioni e sull'attività neuronale.

Capitolo 2

Modelli e metodi

2.1 Reti neurali artificiali e ricorrenti

Negli ultimi dieci anni è fiorito lo sviluppo di metodi di intelligenza artificiale basati sulle reti neurali in cui il computer impara a svolgere dei compiti a partire da un vasto insieme di esempi di allenamento. Le reti neurali artificiali offrono alla ricerca neuroscientifica un valido strumento per modellizzare in modo semplificato la dinamica dei processi cognitivi che avvengono all'interno del cervello.

Ispirata vagamente alla semplificazione di una rete neuronale biologica, una rete neurale è un modello computazionale che consiste in un gran numero di nodi di calcolo detti “neuroni” artificiali densamente interconnessi. Le moderne reti neurali artificiali sono organizzate in diversi strati di nodi chiamati “layer” (nel caso di reti multilayer si parla di “deep learning”) in cui l'informazione fluisce secondo un meccanismo “feed-forward”, ovvero i dati si muovono in una direzione sola: ciascun nodo è connesso ai nodi dello strato precedente, da cui riceve informazioni, e ai nodi dello strato successivo a cui invia informazioni.

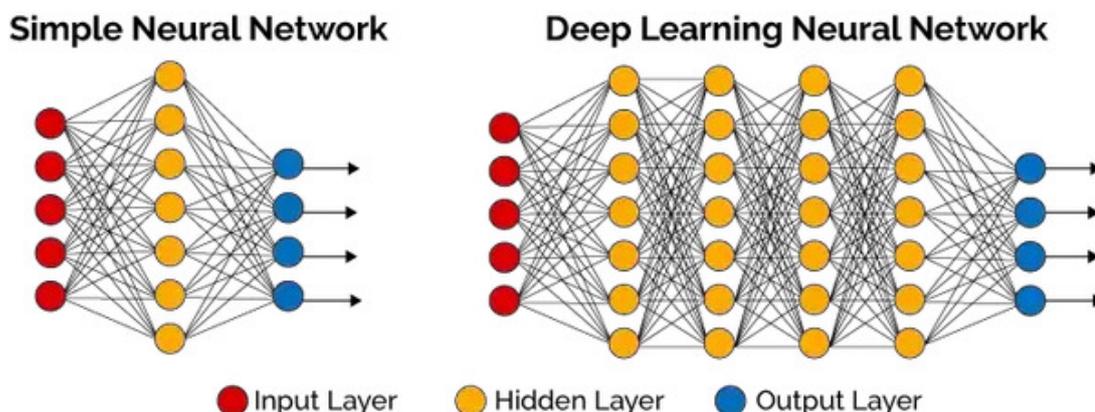


Figura 2.1: Reti neurali artificiali semplici e multilayer (*deep learning*)

Una particolare classe di reti neurali artificiali è costituita dalle reti neurali ricorrenti (RNN), le quali sono particolarmente efficaci per l'elaborazione di dati sequenziali, come serie temporali. La loro struttura considera nel processo di calcolo non solo l'input corrente, ma anche una memoria degli stati precedenti, rendendole ideali per compiti che richiedono informazioni sul contesto per fare previsioni o classificazioni accurate.

La caratteristica distintiva delle RNN è la loro architettura ciclica, in cui, oltre alla normale struttura a layer, i nodi della rete possono influenzare sé stessi attraverso cicli temporali. Questo si traduce in una rete che possiede una memoria interna, che può teoricamente conservare informazioni su eventi passati arbitrariamente lontani nella sequenza di input. Ogni unità di una RNN riceve un input dall'esterno e un input dallo stato precedente della rete, calcolando così una nuova risposta del nodo per ogni passo temporale.

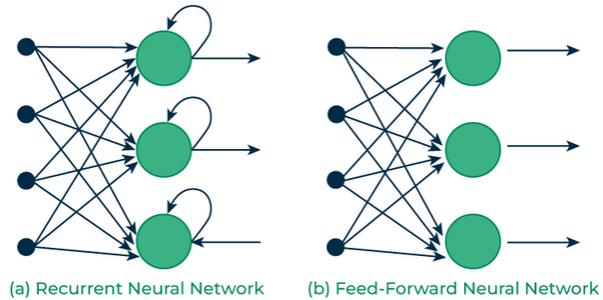


Figura 2.2: Differenti strutture di una rete neurale ricorrente (RNN) e una rete neurale “feed-forward”.

Questa dinamica permette alle RNN di modellare sequenze di dati in cui il contesto storico è cruciale. Per questa motivazione le RNN sono impiegate nello studio dei meccanismi computazionali con cui i cluster di neuroni biologici svolgono diverse funzioni come memoria, processi decisionali e controllo motorio.

Recenti studi evidenziano che le rappresentazioni neurali sono altamente dinamiche, codificano molti aspetti di compiti, stimoli e comandi nell'attività congiunta di popolazioni di neuroni interconnessi (invece dell'attivazione statica di singoli neuroni). Modelli basati su RNN che operano nel regime quasi caotico descrivono bene simili dinamiche.

La dinamica della rete neurale ricorrente presa in esame in questa tesi è descritta matematicamente dalle equazioni classiche delle RNN a tempo continuo completamente connesse formate da N neuroni:

$$\tau \frac{dx_i}{dt} = -x_i(t) + \sum_{j=1}^N J_{i,j} r_j(t) + \sum_{k=1}^M B_{i,k} u_k(t) \quad (2.1)$$

$$r_i(t) = \tanh(x_i(t)) \quad (2.2)$$

dove τ è il tempo di rilassamento dell'attivazione neurale, x_i è il potenziale (o eccitazione) del neurone i -esimo, r_i è la sua attività (o risposta), $J_{i,j}$ è il peso della connessione tra i neuroni j ed i , $u_k(t)$ è il valore corrente di ciascuno degli M input esterni e $B_{i,k}$ è il peso della connessione dall'input esterno k al neurone i .

Il segnale di input avviene attraverso i canali u_k , mentre il segnale di output è letto dall'attività

di un neurone della rete scelto casualmente.

2.2 Metodi di addestramento

Per determinare le matrici dei pesi J e B , esistono diversi metodi e tecniche di addestramento. Tuttavia, i metodi solitamente usati per il training di questi modelli ricorrenti non sono plausibili dal punto di vista biologico.

In [4] si descrive come algoritmo biologicamente plausibile un modello che rispetti le seguenti caratteristiche:

- tutti i cambi di peso sono calcolati da informazioni locali alla sinapsi, ovvero l'attività corrente dell'unità pre-sinaptica e l'attività corrente o salvata nella memoria a breve termine dell'unità post-sinaptica;
- calcoli complessi come rinormalizzazioni, prodotti interni e divisioni sono evitati;
- le connessioni tra le unità sono asimmetriche;
- gran parte dell'apprendimento avviene in maniera non supervisionata.

2.2.1 Apprendimento supervisionato

La tecnica più comunemente utilizzata consiste nell'apprendimento supervisionato, in cui la distanza tra l'output della rete e un segnale output di riferimento viene minimizzata tramite un algoritmo di discesa del gradiente.

Questo tipo di algoritmo richiede un segnale di supervisione costantemente aggiornato e questo entra in contrasto con la maggioranza dei task comportamentali, in cui le informazioni di riscontro sono date saltuariamente e in ritardo rispetto agli stimoli che le hanno causate.

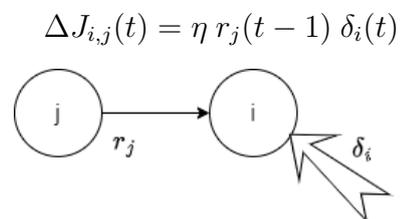


Figura 2.3: Schema apprendimento supervisionato; r_j è la risposta del neurone j-esimo, η è una costante di apprendimento e δ_i è un segnale continuo di errore che arriva dalla rete esterna.

2.2.2 Apprendimento Hebbiano

Una forma di apprendimento più vicina alla realtà biologica è l'apprendimento Hebbiano modulato dalla ricompensa.

Durante il processo di calcolo ogni connessione (sinapsi artificiale) accumula un potenziale cambio di peso caratteristico dell'apprendimento Hebbiano classico, ovvero le attività in input e output della sinapsi vengono moltiplicate in ogni istante e tale prodotto viene accumulato lungo tutto il tempo del processo. Infine, questi potenziali cambi di peso sono moltiplicati con un segnale ricompensa basato sulla performance della rete, andando a determinare le effettive variazioni delle matrici dei pesi.

Questo tipo di apprendimento si ispira all'effetto della dopamina sulla plasticità cerebrale ed è stato testato con successo in reti feed-forward e debolmente connesse; tuttavia, questo metodo di apprendimento non è applicabile a reti ricorrenti fortemente connesse dotate di una dinamica ricca e complessa come quella presa in esame.

2.2.3 Node-perturbation

Un metodo in grado di allenare tali reti ricorrenti è il metodo *node-perturbation*:

è un metodo di apprendimento per rinforzo che richiede segnali di ricompensa sporadici e ritardati e si ispira al meccanismo intuitivo dell'algoritmo REINFORCE [6], in cui si applicano perturbazioni esplorative casuali alle attività neurali, e si modificano i pesi delle connessioni in modo che gli output futuri siano più simili a quello corrente (se l'output perturbato ha indotto una buona ricompensa), o meno simili (se, viceversa, la ricompensa è stata scarsa). Per fare ciò, si calcolano dapprima cambi di peso potenziali (moltiplicando l'input non perturbato con la perturbazione), che vengono moltiplicati con un segnale di ricompensa (positiva o negativa) per determinare i cambi di peso effettivi.

$$E_{i,j} = \sum_t r_j(t-1)x_i(t)$$

$$\Delta J_{i,j} = \eta E_{i,j} R$$

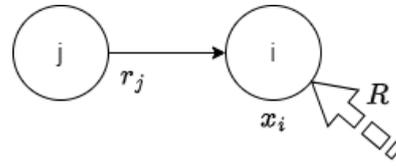


Figura 2.4: Schema apprendimento Hebbiano; $E_{i,j}$ è la matrice dei potenziali cambi di peso, r_j è la risposta del neurone j -esimo pre-sinaptico, x_i è il potenziale del neurone i -esimo (post sinaptico), R è un segnale di ricompensa sporadico.

$$E_{i,j} = \sum_t r_j(t-1) n_i(t)$$

$$\Delta J_{i,j} = \eta E_{i,j} R$$

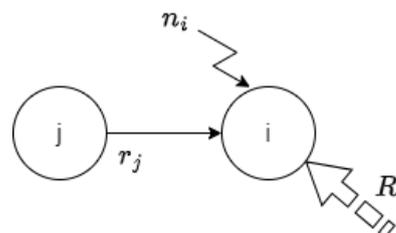


Figura 2.5: Schema apprendimento node-perturbation; n_i è la perturbazione esplorativa in input.

Questo metodo è stato applicato con buon esito a reti feed-forward e reti ricorrenti caotiche. Il node perturbation è un modello non Hebbiano, in quanto moltiplica due tipi di input (input non perturbato e perturbazione), anziché un input e un output, e richiede che le sinapsi abbiano la capacità di discriminare tra essi. Pertanto, è dubbio che reti neurali biologiche possano davvero implementare il node perturbation, ben poco affine ai modelli noti di plasticità cerebrale, tutti basati su informazioni localmente accessibili alla sinapsi (le attività pre- e post-sinaptiche).

2.2.4 Metodo Hebbiano-Esplorativo

Il lavoro di Legenstein, Hoerzer e Maass [8], [9] mostra come, sotto certe condizioni, l'algoritmo node-perturbation venga implementato con aspetti tipici del metodo Hebbiano (viene infatti chiamato metodo Hebbiano-Esplorativo "E-H") diventando biologicamente più plausibile. Si basa sul principio che, se le perturbazioni sono sufficientemente veloci e intense, allora esse si riflettono in fluttuazioni rapide dell'output del neurone (identificabili sottraendo al segnale di output una sua media mobile a breve termine).

Queste fluttuazioni, che sostituiscono le perturbazioni esplorative dell'approccio node-perturbation, vanno moltiplicate per l'attività presinaptica nel computo della "eligibility trace" (viene chiamato così il potenziale cambio di peso), che a sua volta viene modulata da un segnale di ricompensa. In questo modo si è modificato l'algoritmo di node-perturbation in una versione Hebbiana, in cui le informazioni coinvolte nel calcolo sono "localmente accessibili" alla sinapsi artificiale.

Questo metodo è in grado di allenare con buon fine reti neurali ricorrenti caotiche a svolgere task complessi. Tuttavia, questo sistema richiede un segnale di ricompensa istantaneo e continuamente aggiornato, richiesta non compatibile con la maggior parte dei compiti cognitivi in cui i premi sono forniti solo in modo sporadico e ritardato per indirizzare il processo di apprendimento.

Nel metodo E-H è necessario un segnale continuo di ricompensa per rimediare agli effetti di rilassamento: infatti sottrarre una media mobile dall'attività neuronale corrente isola la perturbazione esterna nell'istante in cui è appena comparsa, ma successivamente la perturbazione rientra nel computo della media mobile e la differenza tra attività istantanea e media mobile diventa negativa portando all'effetto di rilassamento. La somma di tutti i termini negativi di

$$\Delta J_{i,j}(t) = \eta r_j(t-1) (x_i(t) - \bar{x}_i) (R(t) - \bar{R})$$

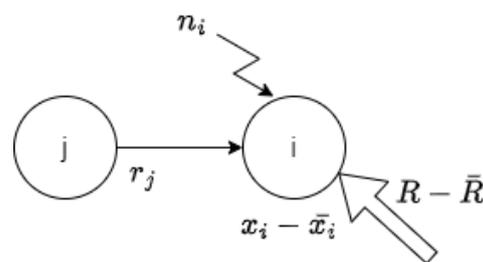


Figura 2.6: Schema apprendimento "E-H"; \bar{x}_i è la media a breve termine salvata nella memoria del neurone, R è il segnale continuo di ricompensa.

rilassamento nell'eligibility trace va ad annullare l'effetto della perturbazione completamente o parzialmente. Questo effetto rende l'algoritmo E-H incapace di apprendere con ricompense

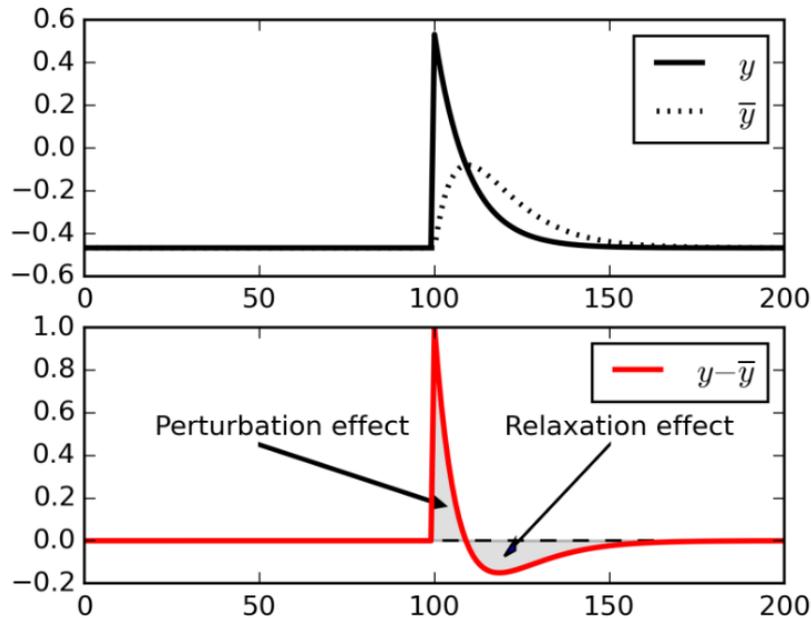


Figura 2.7: Visualizzazione effetto di rilassamento, da [10]: l'area positiva dell'effetto perturbativo coincide con l'area negativa propria dell'effetto di rilassamento.

saltuarie. Si risolve il problema usando un segnale di ricompensa continuamente aggiornato, che è a sua volta filtrato sottraendovi la sua media mobile; quando si presenta l'effetto negativo di rilassamento contemporaneamente sia sull'attività neuronale che sulla ricompensa, il loro prodotto dà un valore positivo portando ad un rafforzamento ulteriore dei termini correlati alla perturbazione.

2.2.5 Metodo di apprendimento proposto da Miconi

In [10], Miconi propone un nuovo metodo di apprendimento Hebbiano modulato da premi (*reward-modulated*). Contrariamente ai precedenti, il metodo di Miconi è in grado di allenare reti ricorrenti per compiti con premi in ritardo rispetto agli stimoli e forniti solo al termine di ciascun ciclo come nella maggior parte degli schemi di apprendimento animale. Si tratta di un metodo Hebbiano che usa solo informazioni localmente accessibili alla sinapsi e non necessita un segnale di ricompensa istantaneo. Si applica a diversi compiti che richiedono decisioni flessibili a seconda del contesto, mantenimento della memoria, coordinazione tra multipli output. Studiando la rappresentazione neurale delle proprietà rilevanti del compito, si nota che le reti allenate presentano casi di selettività mista (ciascun neurone contribuisce a codificare più di una proprietà), similmente a quanto già osservato nella corteccia prefrontale

dei primati. Si conclude che le reti ricorrenti allenate con un metodo di apprendimento Hebbiano reward-modulated offrono un modello plausibile dell'apprendimento in sistemi neurali biologici.

Il metodo proposto da Miconi è la realizzazione biologicamente più plausibile dell'approccio node-perturbation: come in tale metodo, si applicano perturbazioni casuali alle risposte neurali e si modificano i pesi nell'ottica di produrre risposte future più simili a quella indotta dalla perturbazione se l'output perturbato ha portato a un risultato migliore dell'aspettativa, altrimenti si modificano i pesi in modo da produrre risposte differenti da quelle indotte dalla perturbazione.

$$E_{i,j} = \sum_t S(r_j(t-1)(x_i(t) - \bar{x}_i))$$

$$\Delta J_{i,j} = \eta E_{i,j} (R - \bar{R})$$

Come il metodo Hebbiano-Esplorativo (E-H), il nuovo metodo prende le perturbazioni esplorative dall'output sottraendo un media mobile dalla risposta neurale istantanea.

A differenza del metodo E-H, però, il metodo di Miconi non richiede un continuo segnale di ricompensa, ma è in grado di apprendere da premi saltuari e ritardato al termine di ciascun ciclo. Ciò è possibile grazie all'amplificazione sopralineare dei cambi proposti dei pesi sinaptici.

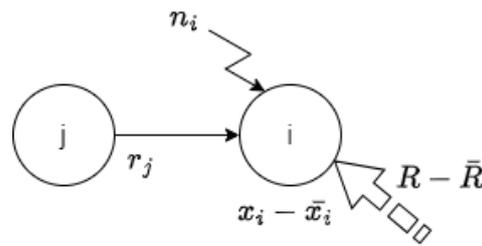


Figura 2.8: Schema apprendimento proposto da Miconi in [10]; $S(x)$ è una funzione sopralineare, R è il segnale di ricompensa sporadico.

Infatti, si nota che le fluttuazioni derivanti dalle perturbazioni sono grandi, mentre quelle dovute all'effetto di rilassamento sono di piccola entità; quindi, se applichiamo una funzione sopralineare ai cambi proposti, i grandi termini che contengono la perturbazione isolata vengono amplificati, invece i termini ridotti dell'effetto di rilassamento vengono soppressi. Questa soluzione si ispira ai metodi Hebbiani a soglia [11], in cui si innesca il meccanismo di neuroplasticità solo quando il prodotto Hebbiano (*eligibility trace*) raggiunge una certa soglia. Una funzione sopralineare amplifica più dolcemente le fluttuazioni, ma con un effetto molto simile in quanto ignora le piccole correlazioni tra input e output accidentali e mantiene amplificandole le fluttuazioni intense, probabili portatrici di informazioni.

Il metodo proposto da Miconi può essere schematizzato come di seguito:

1. Si inizializzano le matrici dei pesi J e B casualmente secondo quanto richiesto dal tipo di task che si vuole allenare;

2. Si inizializzano i potenziali dei neuroni x_i casualmente e si calcola la risposta r_i di ciascun neurone;
3. Si propaga l'equazione differenziale che descrive la dinamica della rete neurale ricorrente;
4. Si aggiunge al potenziale di ciascun neurone una perturbazione casuale e indipendente;
5. Si accumula per ciascuna connessione sinaptica ad ogni passo temporale l'eligibility trace $E_{i,j}$ (il potenziale cambio di peso), con un incremento calcolato come prodotto tra la risposta $r_j(t-1)$ del neurone mittente e la fluttuazione dell'eccitazione del neurone ricevente $x_i(t) - \bar{x}_i$, a cui si applica una funzione sopralineare ($\Delta E_{i,j}(t) = S(r_j(t-1)(x_i(t) - \bar{x}_i)$).
6. Al termine del ciclo sperimentale (di seguito indicato con "trial"):
 - 6.1. viene emessa una ricompensa R a seconda del tipo di compito cognitivo, e si calcola un "reward prediction error" sottraendo ad R \bar{R} , ovvero la ricompensa attesa calcolata come media delle ricompense ottenute nei trial precedenti dello stesso tipo;
 - 6.2. si calcola l'effettivo cambio di pesi ΔJ modulando l'eligibility trace con il reward prediction error; per stabilizzare l'addestramento si limitano le modifiche dei pesi con una soglia;
7. Si azzerà l'eligibility trace e si ricomincia un nuovo trial partendo dal punto 2 fino al completamento del processo di addestramento.

Capitolo 3

Struttura della rete neurale

3.1 Implementazione numerica

Come già visto nella sezione 2.1, la dinamica della rete è descritta dalle equazioni 2.1 e 2.2.

Per prima cosa si inizializzano tutte le grandezze descritte:

- J con distribuzione normale a media 0 e varianza g^2/N ;
- $B_{k,i}$ fissati e presi da distribuzione uniforme nell'intervallo $[-1;1]$;
- x_i con distribuzione uniforme nell'intervallo $[-0.1;0.1]$;

Si usano i valori costanti indicati nell'articolo: $N = 200$, $M = 2$, $\tau=30\text{ms}$, $g = 1.5$ (regime early chaotic). Durante tutto il tempo del trial, 4 neuroni scelti casualmente (per semplicità si scelgono i primi 4) mantengono un valore costante di attivazione $x = 1$. Non c'è un output separato, ma la risposta della rete coincide con la risposta di un neurone scelto casualmente tra i neuroni non fissati (per semplicità si sceglie l' N -esimo); quest'ultimo non dev'essere influenzato dagli input, dunque le connessioni $B_{N,k} = 0$.

Si propaga l'equazione differenziale in (2.1) con il metodo di Eulero implicito, applicando un'opportuna discretizzazione temporale $dt=1\text{ms}$:

$$x_i(t+1) = \frac{1}{1+dt/\tau}x_i(t) + \frac{dt}{dt+\tau} \cdot \left(\sum_{j=1}^N J_{i,j}r_j(t) + \sum_{k=1}^M B_{i,k}u_k(t) \right) \quad (3.1)$$

Ad ogni passo temporale, sia durante l'addestramento che nella fase di verifica, si aggiungono ai potenziali dei neuroni x_i le perturbazioni esplorative: tali perturbazioni n_i sono prese da una distribuzione uniforme in $[-0.5, 0.5]$ e applicate a ciascun neurone casualmente e indipendentemente con un rate di 1Hz.

Seguendo lo schema del metodo di addestramento proposto da Miconi (sezione 2.2.5), si calcola il prodotto Hebbiano

$$\Delta E_{i,j}(t) = [r_j(t-1) \cdot (x_i(t) - \bar{x}_i)]^3 \quad (3.2)$$

dove si è usata la funzione sopralineare cubica $S(x) = x^3$; \bar{x}_i rappresenta una media mobile a breve termine del potenziale di ciascun neurone aggiornata con la formula (con $\beta=0.05$):

$$\bar{x}_i(t) = \beta \bar{x}_i(t-1) + (1 - \beta) x_i(t) \quad (3.3)$$

Durante tutto il tempo del trial di addestramento, per ciascuna sinapsi, che collega il neurone j -esimo all' i -esimo, gli incrementi $\Delta E_{i,j}(t)$ si accumulano ad ogni time step nell'eligibility trace $E_{i,j}$.

Al termine del trial, si stima l'errore ϵ come valore assoluto della differenza tra l'output atteso e quello restituito dalla rete, e la ricompensa (*reward*) è $R = -\epsilon$. Per ogni tipologia di trial, si tiene traccia di una media mobile che rappresenta la ricompensa attesa in assenza della perturbazione e al k -esimo trial viene aggiornata (con $\alpha=0.5$):

$$\bar{R}(k) = \alpha \bar{R}(k-1) + (1 - \alpha) R(k) \quad (3.4)$$

Si può determinare quindi l'effettivo cambio della matrice dei pesi con parametro di apprendimento (*learning rate*) $\eta = 0.1$, che viene moltiplicato per $-\bar{R}$ durante il training per modularne l'intensità:

$$\Delta J = -\eta (R(k) - \bar{R}(k)) E (-\bar{R}(k)) \quad (3.5)$$

Per stabilizzare l'addestramento, impostiamo una soglia $\theta = 10^{-3}$ per i ΔJ : questi, se il cambio di pesi supera la soglia (in valore assoluto), vengono troncati al valore θ .

Una volta modificati i pesi delle connessioni tra i neuroni, si possono ri-inizializzare gli x_i , azzerare E e iterare il procedimento per n trial, con $n = 10000$.

3.2 Delayed nonmatch-to-sample task

La RNN viene allenata a svolgere un compito di "delayed nonmatch-to-sample", ovvero la rete deve indicare se due segnali, uno in ritardo rispetto all'altro, combaciano.

Questo task presenta delle caratteristiche d'interesse per la ricerca neuroscientifica: in primo luogo, è probabilmente il più semplice tra i compiti decisionali che dipendono dal contesto (percependo il secondo stimolo, la rete deve produrre una risposta che dipende dal contesto stabilito dal primo stimolo). Altro aspetto peculiare è il fatto che l'intervallo tra i due input è molto più

grande del tempo di rilassamento dell'attività neuronale, pertanto la rete deve mantenere in memoria l'entità del primo stimolo prima che il secondo compaia. Infine, per risolvere il compito, alcuni neuroni della rete devono avere una forma di “selettività mista non-lineare” (*nonlinear mixed selectivity*), un tratto caratteristico dell'attività neuronale nella corteccia prefrontale [12]. La *nonlinear mixed selectivity* consiste nella codifica congiunta di diversi caratteri degli stimoli; si sono osservate delle rappresentazioni di questo tipo nelle regioni corticali dei sistemi sensoriali primari, dei processi decisionali e del movimento. In queste aree cerebrali, le informazioni sono quasi sempre interpretate non linearmente, al posto di rappresentazioni separate o con semplice combinazione lineare additiva, generalmente chiamate “selettività pura” (*pure selectivity*) [13].

A scopo esemplificativo consideriamo un semplice task in cui vengono sottoposti due stimoli α o β ad un neurone, si ha:

Pure selectivity (rappresentazione separata), se neurone risponde +1 quando il primo stimolo è α . La risposta del neurone si può scrivere con l'equazione $r = \delta(\text{stim1} = \alpha)$.

Linear mixed selectivity (rappresentazione con combinazione lineare), se il neurone risponde +1 ogni volta in cui compare uno stimolo α . La risposta neuronale è descritta da $r = \delta(\text{stim1} = \alpha) + \delta(\text{stim2} = \alpha)$.

Nonlinear mixed selectivity (rappresentazione non-lineare), se il neurone risponde +1 quando entrambi gli stimoli sono α . La risposta è descritta da $r = \delta(\text{stim1} = \alpha) \cdot \delta(\text{stim2} = \alpha)$.

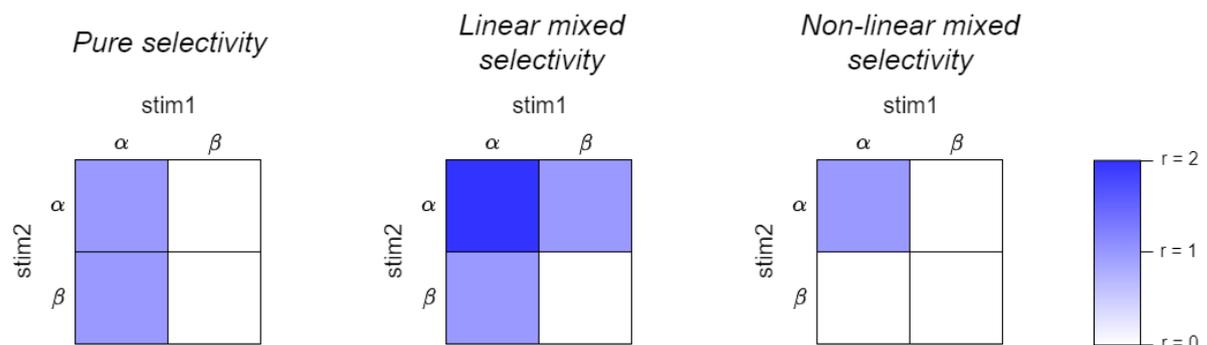


Figura 3.1: Diverse tipologie di selettività, semplice esempio con la risposta di un solo neurone.

In ogni trial del task “delayed nonmatch-to-sample”, si sottopongono alla rete due brevi input successivi, tra cui intercorre del tempo; ogni input può assumere due valori: A o B. Si usano due diversi canali d'ingresso u_1 e u_2 per specificare l'identità dello stimolo, determinata da quale canale si attiva: per l'input A $u_1=1$ e $u_2=0$, mentre per l'input B $u_1=0$ e $u_2=1$. Si ricorda che ogni canale d'ingresso u_k è connesso alla rete (escluso il neurone di output) con il rispettivo set di pesi indipendente, registrato in $B_{i,k}$.

L'obiettivo del task è determinare se i due input sono identici (AA o BB) e nel caso restituire l'output -1; oppure se gli input sono diversi (AB o BA) emettere l'output +1. Ogni combinazione di input costituisce una diversa tipologia di trial.

Ogni trial si svolge nel seguente modo: per 200ms si sottopone alla rete il primo stimolo, successivamente si aspettano 200ms e poi si sottopone il secondo stimolo per altri 200ms. Al di fuori delle fasi in cui gli stimoli sono presentati, i canali d'ingresso u_k sono fissati a 0.

Il trial prosegue per altri 400ms: dopo 200ms di attesa, gli ultimi 200ms costituiscono la fase di risposta, durante la quale l'attività del neurone scelto arbitrariamente come output costituisce la risposta complessiva della rete. Pertanto, il trial ha una durata totale di 1000ms.

L'errore globale per ciascun trial è la differenza assoluta tra la media lungo gli ultimi 200ms della risposta della rete (che è l'attività del neurone designato come output) e la risposta attesa (+1 o -1 a seconda della tipologia di trial).

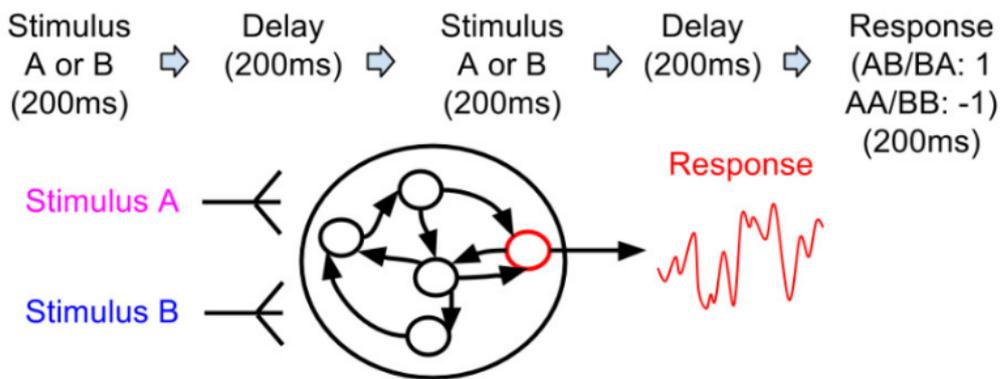


Figura 3.2: Descrizione del task “delayed nonmatch-to-sample”, da [10].

Capitolo 4

Risultati

In questo capitolo si presentano i risultati della rete addestrata come descritto nella sezione 3.1.

Durante l'addestramento si tiene traccia dell'errore di ciascun trial, rappresentato nei grafici in figura 4.1. Si osserva che, dopo circa 2000 trial, l'errore presenta una chiara convergenza e, al termine del *training*, si assesta attorno al valore $\epsilon \approx 0.1$.

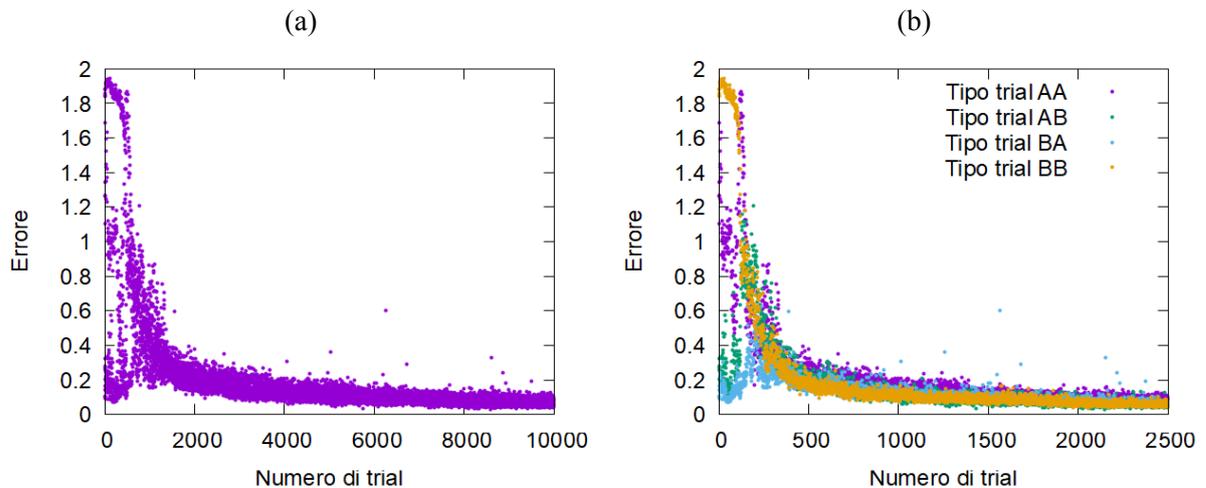


Figura 4.1: Andamento dell'errore nel corso dell'addestramento: (a) complessivo; (b) suddiviso per tipologia di trial.

Per analizzare gli effetti dell'addestramento, si studiano in primo luogo come variano i pesi. In figura 4.2 si rappresentano le matrici di connessione J , dove il colore indica il peso di ciascuna connessione, e la distribuzione dei pesi $J_{i,j}$ all'inizio, metà e fine dell'addestramento. Si nota come la distribuzione dei pesi subisca variazioni minime dall'inizializzazione di J con distribuzione apparentemente normale. Infatti, gli effetti dell'addestramento si distribuiscono su 40000 connessioni, portando a piccole variazioni sul valore dei singoli pesi, ma che complessivamente rendono la matrice efficacemente allenata a svolgere il compito "delayed nonmatch-to-sample".

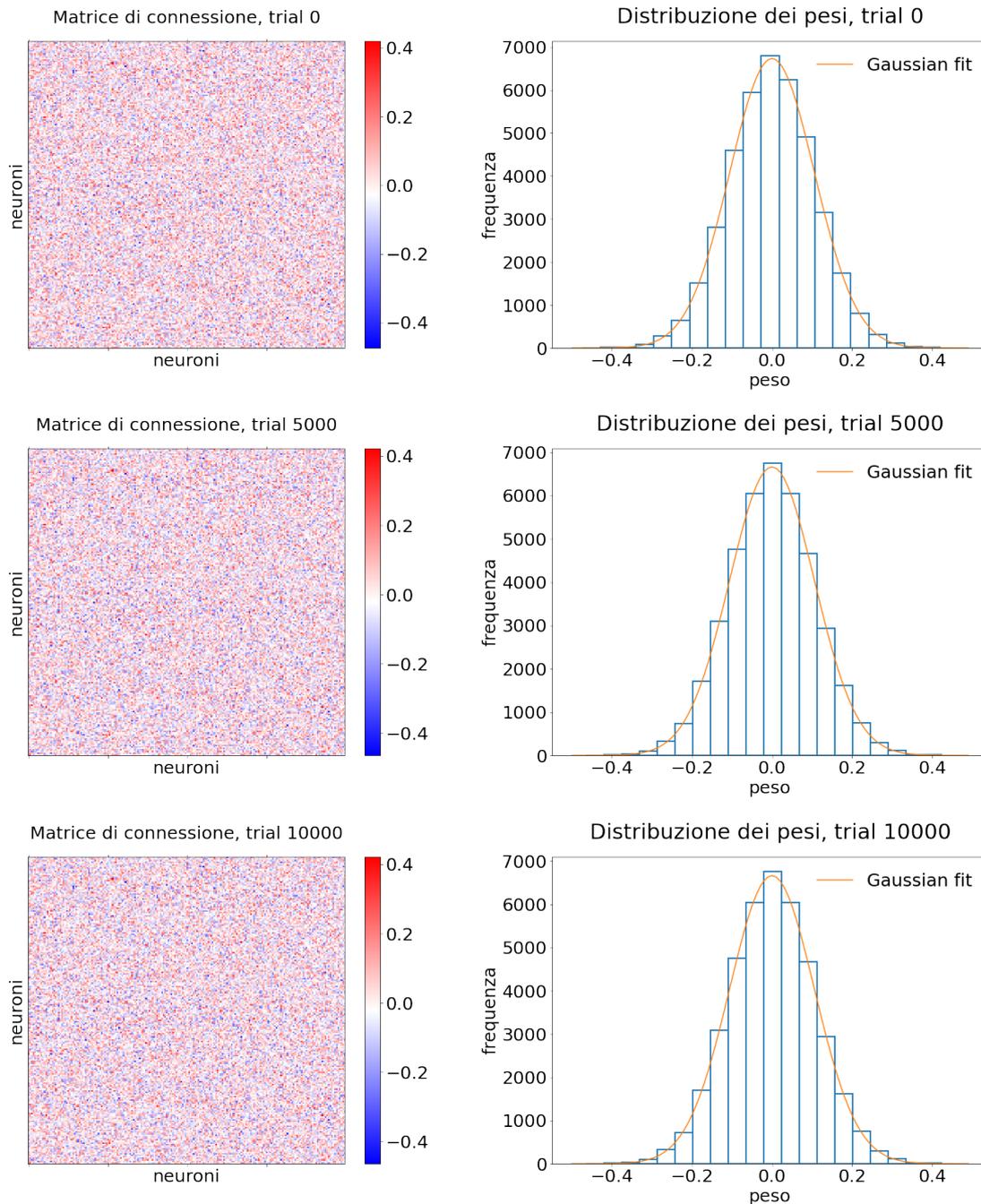


Figura 4.2: Matrici dei pesi e istogrammi distribuzioni dei pesi all'inizio, metà e fine dell'addestramento.

Si testa l'ipotesi nulla che la matrice dei pesi addestrata (trial 10000) abbia distribuzione normale con media e varianza non note con il test di Lilliefors [14]: si ottiene un $p\text{-value} = 0.39$, per cui l'ipotesi di distribuzione normale non può essere rifiutata.

Si valuta la differenza tra i pesi iniziali e al termine dell'addestramento. In figura 4.3, nella

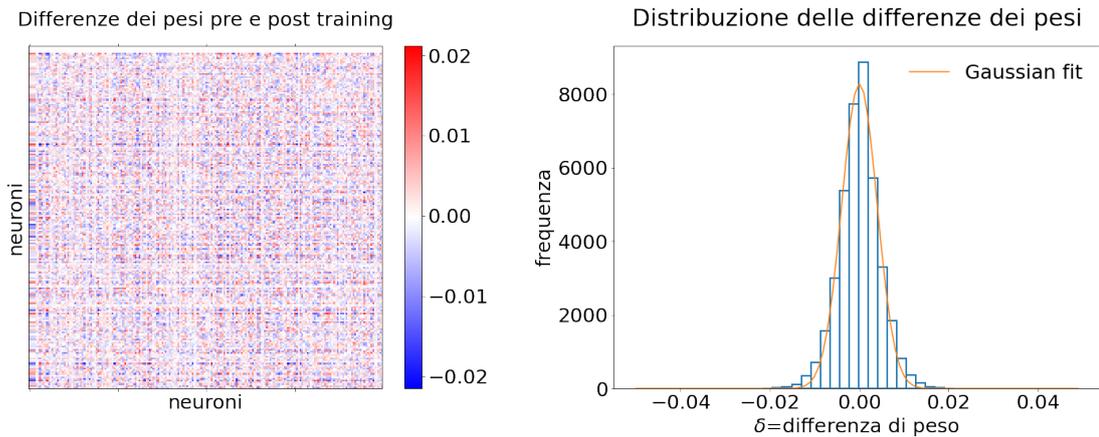


Figura 4.3: Matrice e istogramma distribuzione della differenza dei pesi pre e post *training*.

rappresentazione della matrice delle differenze di peso, si nota che le connessioni con i primi 4 neuroni (fascia bianca in alto) non variano, in quanto sono i *fixed neurons* (come descritto nella sezione 3.1). Dall'istogramma della distribuzione delle differenze di peso si evince che anche le differenze sembrano essere distribuite secondo una normale, i cui estremi raggiungono in assoluto $\delta = 0.02$. Si calcola per ogni connessione la variazione relativa di peso: $\left| \frac{\Delta J_{i,j}}{J_{i,j \text{ ini}}} \right|$; mediando su tutte le connessioni i, j si ottiene una variazione media relativa del 38%. Si ripete il test di Lilliefors sulla distribuzione delle differenze tra i pesi, ottenendo un $p\text{-value} = 0.001$ che induce a rifiutare l'ipotesi che le differenze di pesi siano distribuite secondo una normale.

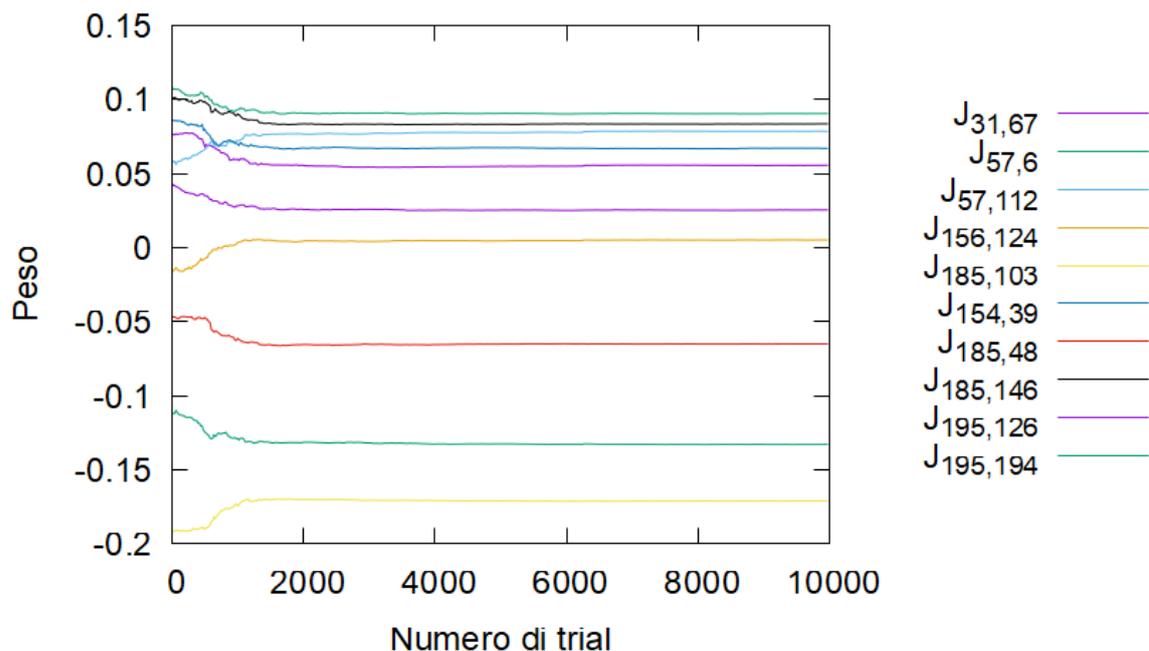


Figura 4.4: Variazione dei 10 pesi che cambiano maggiormente.

Quindi, si identificano i 10 pesi che cambiano maggiormente dall'inizio alla fine del training e, in figura 4.4, si visualizza il loro andamento durante tutto il tempo di apprendimento. Nei primi 2000 trial i pesi hanno visibili cambiamenti (attorno alle differenze massime di $\delta = 0.02$), successivamente i loro valori si stabilizzano, in quanto l'algoritmo di addestramento ha raggiunto il punto di convergenza.

In conclusione, l'addestramento apporta minime correzioni ai valori dei singoli pesi, ma produce un effetto collettivo notevole, rendendo la rete in grado di svolgere efficacemente il task.

Si vuole ora analizzare come cambia la dinamica della rete neurale prima e dopo l'addestramento. Per ogni tipologia di trial, si prende in esame l'attivazione dei neuroni pre e post-training; per ridurre gli effetti casuali delle perturbazioni esplorative si media su 10 trial dello stesso tipo.

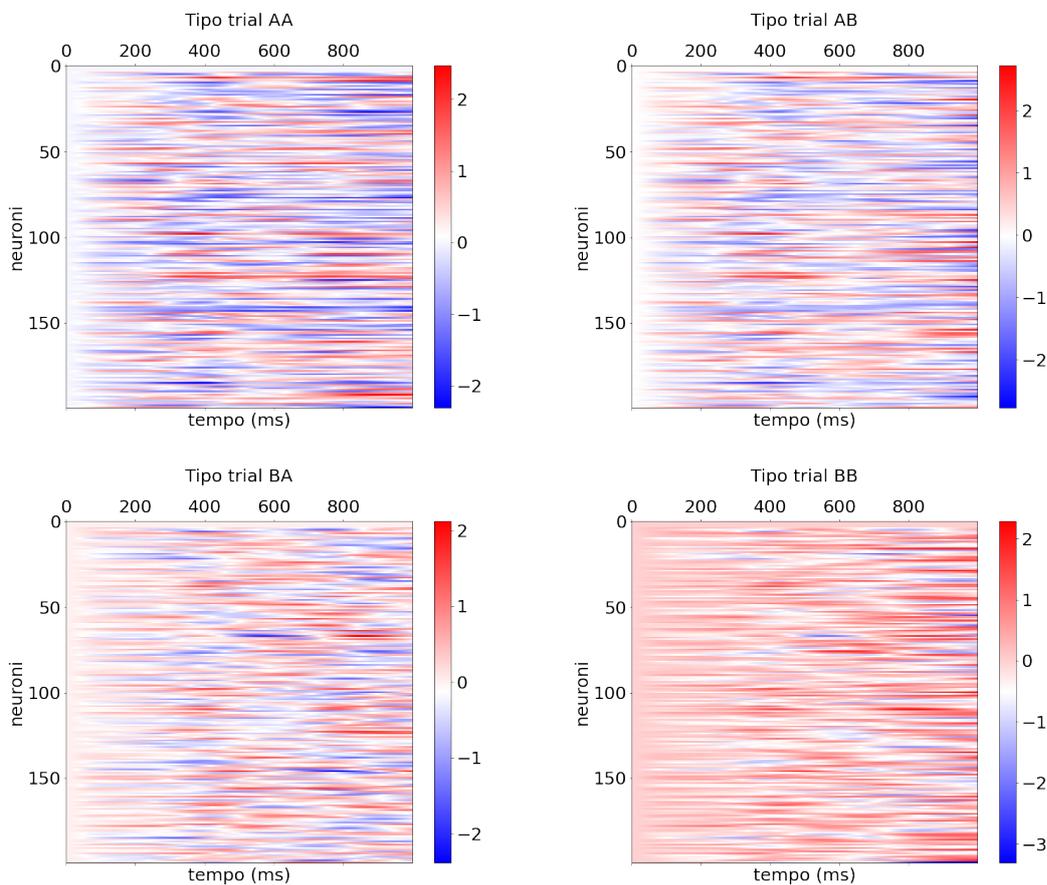


Figura 4.5: Differenza tra l'attivazione neuronale prima e dopo l'addestramento, mediata su 10 trial per ciascun tipo.

Le attività neuronali prima dell'addestramento presentano due fasce di attivazione intense negli intervalli di tempo da 0 a 200ms e da 400 a 600ms, i periodi in cui vengono presentati gli

stimoli esterni; subito dopo la stimolazione si presenta una fascia più chiara di rilassamento. Dopo l'addestramento le due fasce intense a 0-200ms e 400-600ms si ripresentano, inoltre si intensifica la fascia relativa alla fase di risposta (800-1000ms).

In figura 4.5, si rappresenta la differenza di attivazione dei singoli neuroni prima e dopo l'addestramento. Si può notare che la variazione di attività neuronale si concentra soprattutto nella fase finale del trial. Varia in particolare l'attività del neurone di output, rappresentata in figura 4.6: sia prima che dopo l'addestramento, la dinamica è ricca durante tutto il tempo del trial, provando che la rete è in grado di rappresentare e analizzare gli stimoli e produrre risposte flessibili in modo compatibile con le osservazioni nella corteccia prefrontale animale [15]. In particolare, si nota negli istanti in cui viene presentato un nuovo stimolo (0ms e 400ms) come l'attività codifica dinamicamente l'informazione; solo dopo l'addestramento si osserva (a circa 600ms) il passaggio da una rappresentazione specifica dello stimolo a una rappresentazione specifica della risposta.

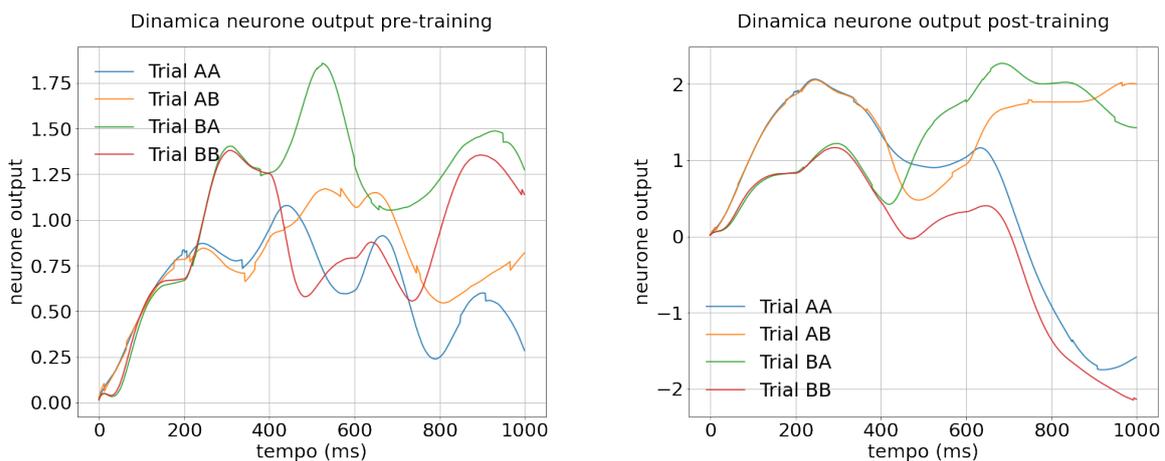


Figura 4.6: Attivazione neurone di output prima e dopo l'addestramento

Come ultima analisi, si verifica lo sviluppo della selettività mista durante l'addestramento. Si valuta la dipendenza dell'attivazione dei neuroni dagli stimoli in ingresso: per ogni neurone prima e dopo il training si interpola la curva:

$$x_i(t) = A_i u_1(t) + B_i u_2(t) + C_i u_1(t) \cdot u_2(t) + D_i \quad (4.1)$$

Si ottengono i coefficienti A , B , C e D dalle interpolazioni e si rappresentano le distribuzioni dei coefficienti prima e dopo l'addestramento per ciascun tipo di trial nella figura 4.7. Si nota in particolare che le code di distribuzione del coefficiente C si allungano dopo l'addestramento, ad indicare un aumento del termine relativo alla selettività mista non-lineare.

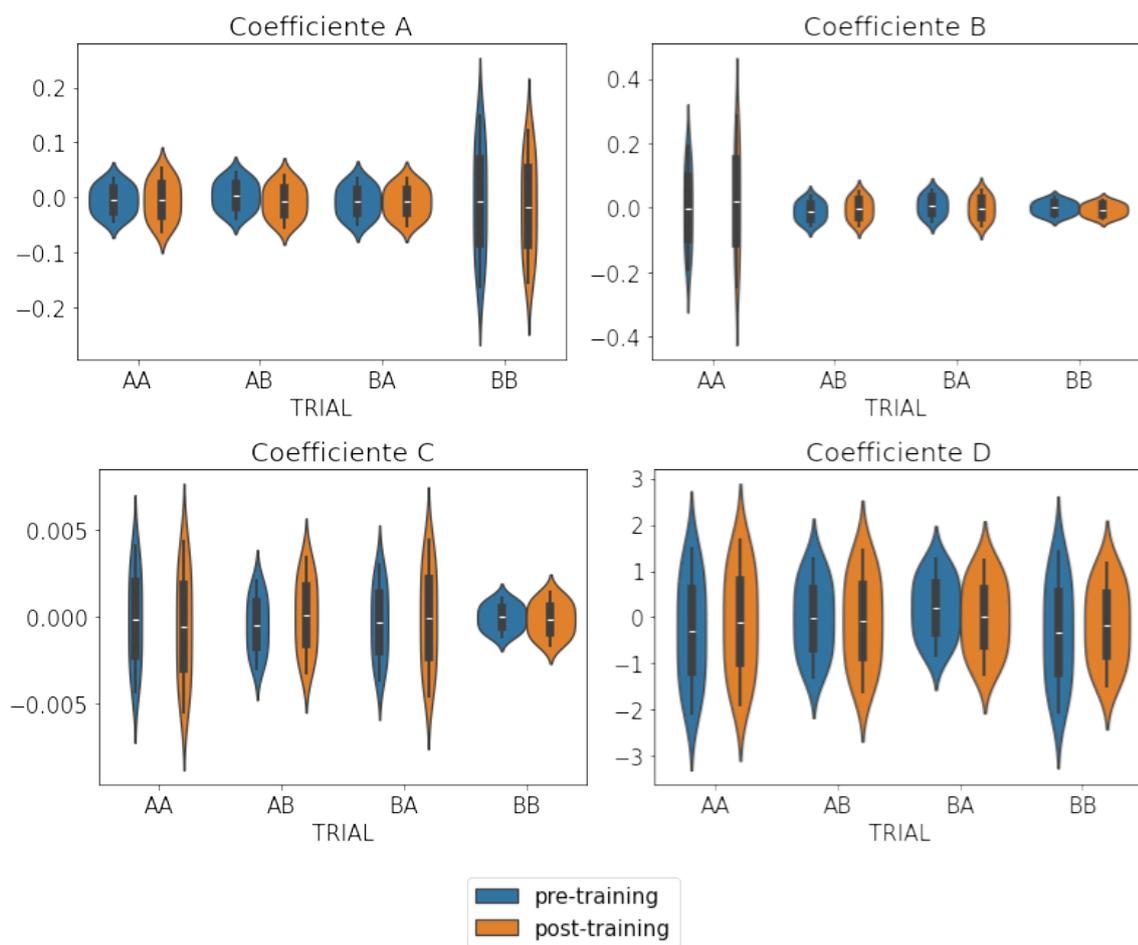


Figura 4.7: Distribuzioni coefficienti A, B, C e D; confronto prima e dopo l'addestramento.

Capitolo 5

Conclusioni

L'obiettivo di questa tesi era simulare la dinamica di una rete neurale ricorrente completamente connessa, addestrata con una regola di apprendimento locale (cioè, gli aggiornamenti dei pesi sinaptici dipendono esclusivamente da informazioni locali, inerenti ai neuroni pre- e post-sinaptico), la non simmetria delle connessioni e l'assenza di un segnale di supervisione durante la maggior parte del processo. A tal fine si è implementato l'algoritmo proposto da Miconi, addestrando una rete a svolgere il compito "Delayed nonmatch-to-sample", un semplice compito decisionale che dipende dal contesto, in cui è necessario che la rete abbia una memoria e una forma di "selettività mista non-lineare".

Nel nostro studio siamo riusciti a riprodurre efficacemente i risultati di Miconi: la regola di apprendimento proposta da Miconi si è rivelata funzionale per l'apprendimento del compito studiato, confermando la sua efficacia.

Dopo circa 1000 iterazioni, i pesi della matrice di connessione si sono stabilizzati, segnalando il raggiungimento di un equilibrio nella rete. La distribuzione complessiva dei pesi è rimasta pressoché invariata, nonostante i singoli pesi abbiano subito variazioni significative, con un cambiamento medio del 38%; ciò ha prodotto un effetto collettivo in grado di rendere la rete capace di svolgere il task.

Durante la dinamica della rete, i profili di attivazione dei singoli neuroni hanno mostrato cambiamenti anche drastici; in seguito all'addestramento, si osserva il passaggio da una rappresentazione specifica dello stimolo a una rappresentazione specifica della risposta. Inizialmente, la dinamica non era in grado di discriminare adeguatamente i quattro tipi di trial, ma dopo l'addestramento, la capacità di discriminazione è migliorata sensibilmente.

Infine, dopo il training, abbiamo osservato un aumento della selettività mista durante l'addestramento: usando un modello di regressione per quantificare la selettività di ciascun neurone, abbiamo riscontrato un allargamento nelle distribuzioni dei termini misti non lineari, corrispondenti alla selettività mista.

Bibliografia

- [1] O. Barak, «Recurrent neural networks as versatile tools of neuroscience research,» *Current opinion in neurobiology*, vol. 46, pp. 1–6, 2017.
- [2] T. C. Kietzmann, P. McClure e N. Kriegeskorte, «Deep neural networks in computational neuroscience,» *BioRxiv*, p. 133 504, 2017.
- [3] E. M. Meyers, D. J. Freedman, G. Kreiman, E. K. Miller e T. Poggio, «Dynamic population coding of category information in inferior temporal and prefrontal cortex,» *Journal of neurophysiology*, vol. 100, n. 3, pp. 1407–1419, 2008.
- [4] M. Tang, Y. Yang e Y. Amit, «Biologically Plausible Training Mechanisms for Self-Supervised Learning in Deep Networks,» *Frontiers in Computational Neuroscience*, vol. 16, 2022.
- [5] P. J. Werbos, «Backpropagation through time: what it does and how to do it,» *Proceedings of the IEEE*, vol. 78, n. 10, pp. 1550–1560, 1990.
- [6] R. J. Williams, «Simple statistical gradient-following algorithms for connectionist reinforcement learning,» *Machine learning*, vol. 8, pp. 229–256, 1992.
- [7] I. R. Fiete e H. S. Seung, «Gradient learning in spiking neural networks by dynamic perturbation of conductances,» *Physical review letters*, vol. 97, n. 4, p. 048 104, 2006.
- [8] R. Legenstein, S. M. Chase, A. B. Schwartz e W. Maass, «A reward-modulated hebbian learning rule can explain experimentally observed network reorganization in a brain control task,» *Journal of Neuroscience*, vol. 30, n. 25, pp. 8400–8410, 2010.
- [9] G. M. Hoerzer, R. Legenstein e W. Maass, «Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning,» *Cerebral cortex*, vol. 24, n. 3, pp. 677–690, 2014.
- [10] T. Miconi, «Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks,» *Elife*, vol. 6, e20899, 2017.
- [11] A. Soltoggio e J. J. Steil, «Solving the distal reward problem with rare correlations,» *Neural computation*, vol. 25, n. 4, pp. 940–978, 2013.

- [12] M. Rigotti, O. Barak, M. R. Warden et al., «The importance of mixed selectivity in complex cognitive tasks,» *Nature*, vol. 497, n. 7451, pp. 585–590, 2013.
- [13] W. J. Johnston, S. E. Palmer e D. J. Freedman, «Nonlinear mixed selectivity supports reliable neural computation,» *PLoS computational biology*, vol. 16, n. 2, e1007544, 2020.
- [14] H. W. Lilliefors, «On the Kolmogorov-Smirnov test for normality with mean and variance unknown,» *Journal of the American statistical Association*, vol. 62, n. 318, pp. 399–402, 1967.
- [15] M. G. Stokes, M. Kusunoki, N. Sigala, H. Nili, D. Gaffan e J. Duncan, «Dynamic coding for cognitive control in prefrontal cortex,» *Neuron*, vol. 78, n. 2, pp. 364–375, 2013.