

UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

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

Relatore:  
**Dott. Michele Allegra**

Laureanda:  
**Angela Bortolato**

16 luglio 2024

# Neuroscienze e reti neurali



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Le reti neurali artificiali aiutano a comprendere i processi neurali del cervello umano.

Si ispirano all'architettura delle reti neuronali biologiche.

Lo studio dei metodi di addestramento può fornire intuizioni su come il cervello apprende e si adatta.



# Neuroscienze e reti neurali



Le reti neurali artificiali aiutano a comprendere i processi neurali del cervello umano.

Si ispirano all'architettura delle reti neuronali biologiche.

Lo studio dei metodi di addestramento può fornire intuizioni su come il cervello apprende e si adatta.



Uno degli obiettivi è sviluppare modelli che non solo riproducano il comportamento osservato dei sistemi neurali, ma che siano anche *biologicamente plausibili*.

# Indice



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

1. Modelli e metodi
2. Implementazione numerica
3. Risultati

# Reti neurali artificiali e ricorrenti

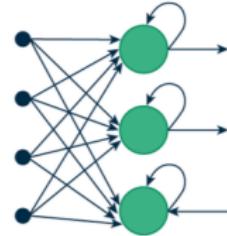
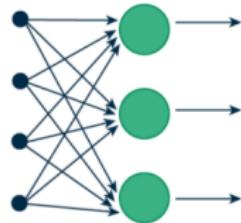


UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

**Rete neurale artificiale:** Modello costituito da molti “neuroni artificiali” interconnessi.

**Rete neurale feed-forward:** L'informazione fluisce attraverso strati con un meccanismo “feed-forward”.

**Rete neurale ricorrente:** Modello ad architettura ciclica, i nodi possono influenzare sé stessi attraverso cicli temporali.  
Utile per compiti che richiedono informazioni sul contesto (memoria, processi decisionali, controllo motorio).



# Metodi di addestramento



Un algoritmo biologicamente plausibile deve avere le seguenti caratteristiche: **Località** delle informazioni, **Assenza di supervisione o ricompensa** in gran parte del processo ed **Efficace** nell'allenare una RNN fortemente connessa e con una dinamica complessa.

# Metodi di addestramento



Un algoritmo biologicamente plausibile deve avere le seguenti caratteristiche: **Località** delle informazioni, **Assenza di supervisione o ricompensa** in gran parte del processo ed **Efficace** nell'allenare una RNN fortemente connessa e con una dinamica complessa.

I metodi tradizionali proposti in letteratura non soddisfano tali richieste. Recentemente, Miconi ha proposto un metodo che rispetta queste caratteristiche. Abbiamo cercato di riprodurne i risultati.

# Metodi di addestramento



Un algoritmo biologicamente plausibile deve avere le seguenti caratteristiche: **Località** delle informazioni, **Assenza di supervisione o ricompensa** in gran parte del processo ed **Efficace** nell'allenare una RNN fortemente connessa e con una dinamica complessa.

I metodi tradizionali proposti in letteratura non soddisfano tali richieste.

Recentemente, Miconi ha proposto un metodo che rispetta queste caratteristiche. Abbiamo cercato di riprodurne i risultati.

Data una rete, si indica con  $x_i$  il potenziale del neurone  $i$ -esimo,  $r_i = \tanh(x_i)$  la sua risposta e  $J$  la matrice di connettività.

# Apprendimento supervisionato



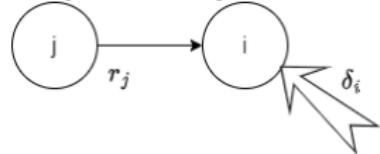
UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Minimizzazione della distanza tra l'output della rete e un segnale di riferimento tramite discesa del gradiente.

$\delta_i$  è un segnale continuo di errore che arriva dalla rete esterna e dipende dalla performance.

Non locale e supervisionato: richiede un segnale costante di errore proveniente dall'esterno della rete.

$$\Delta J_{ij}(t) = \eta \ r_j(t-1) \ \delta_i(t)$$



# Apprendimento Hebbiano

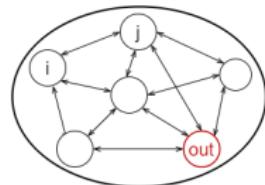


UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Siano  $r_{out}$  la risposta del neurone di output e  $r^*$  la risposta attesa, la **ricompensa** è data da:

$$R(t) = 1 - \epsilon(t) \quad \text{con } \epsilon(t) = |r_{out}(t) - r^*| \quad \text{se istantanea}$$

$$R = 1 - \epsilon \quad \text{con } \epsilon = \sum_t |r_{out}(t) - r^*| \quad \text{se sporadica}$$

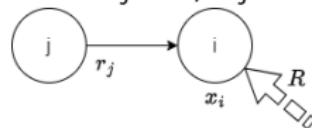


Accumulo di potenziali cambi di peso secondo l'apprendimento Hebbiano classico, modulato al termine di ciascun ciclo da un segnale di ricompensa sporadico.

Non efficace nell'allenare reti completamente connesse con dinamica ricca.

$$E_{ij} = \sum_t r_j(t-1)x_i(t)$$

$$\Delta J_{ij} = \eta E_{ij} R$$



# Node-Perturbation

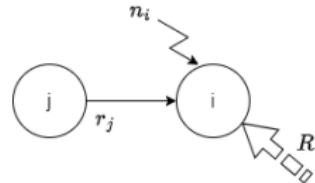


Applicazione di perturbazioni casuali alle attività neurali e, alla fine del trial, modifica dei pesi delle connessioni in base alla ricompensa sporadica ricevuta.

$n_i$  è la perturbazione esplorativa in input.

Non locale: richiede alla rete di discriminare tra l'input della sinapsi e l'input esterno della perturbazione.

$$E_{ij} = \sum_t r_j(t-1) n_i(t)$$
$$\Delta J_{ij} = \eta E_{ij} R$$



# Metodo Hebbiano-Esplorativo

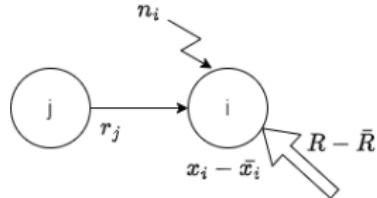


UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Perturbazioni rapide e intense si riflettono in fluttuazioni dell'output neurale; queste, moltiplicate per l'attività presinaptica, formano la "eligibility trace", modulata da un continuo segnale di ricompensa istantaneo.

Supervisionato: richiede un segnale di ricompensa istantaneo e continuamente aggiornato.

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



# Metodo Miconi



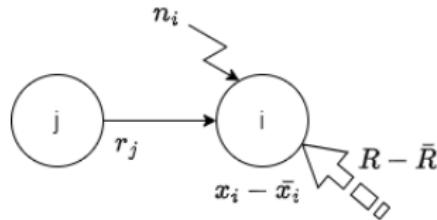
UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Thomas Miconi: *Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks.* In: *Elife* 6 (2017), e20899

- Come nel Node-Perturbation, applica perturbazioni casuali alle attività neurali e modifica i pesi per migliorare le risposte future.
- Come nel metodo E-H, ricava le perturbazioni esplorative dall'output sottraendo un media mobile dalla risposta neurale istantanea.
- Rispetto al Metodo E-H non richiede un continuo segnale di ricompensa, ma può apprendere da premi ritardati e sporadici.
  - $S(x) = x^3$  è una funzione sopralineare, rende l'algoritmo più efficace

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

$$\Delta J_{ij} = \eta E_{ij} (R - \bar{R}) \bar{R}$$



# Indice



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

1. Modelli e metodi
2. Implementazione numerica
3. Risultati

# Dinamica della rete



Equazione classica delle RNN a tempo continuo completamente connesse formate da  $N$  neuroni:

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

$$r_i(t) = \tanh(x_i(t)) \quad (2)$$

Il segnale di input avviene attraverso i canali  $u_k$ , il segnale di output è letto dall'attività di un neurone della rete scelto casualmente.

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

# Task Delayed nonmatch-to-sample



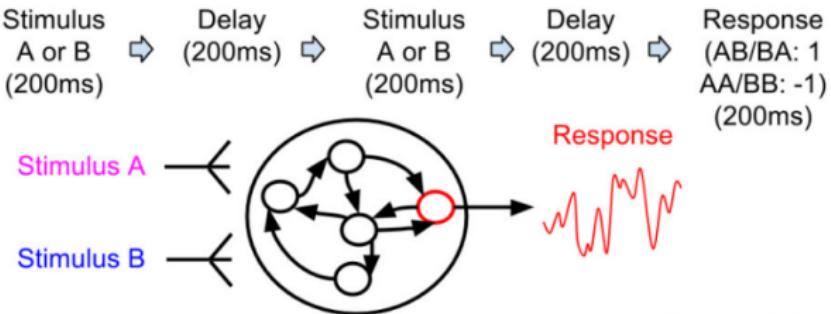
*La rete deve indicare se due segnali, uno in ritardo rispetto all'altro, combaciano.*

- Il più semplice tra i compiti decisionali che dipendono dal contesto.
- L'intervallo tra i due input è molto più grande del tempo di rilassamento, la rete deve mantenere una memoria.

# Task Delayed nonmatch-to-sample

*La rete deve indicare se due segnali, uno in ritardo rispetto all'altro, combaciano.*

- Il più semplice tra i compiti decisionali che dipendono dal contesto.
- L'intervallo tra i due input è molto più grande del tempo di rilassamento, la rete deve mantenere una memoria.



L'identità dello stimolo è determinata da quale canale d'ingresso si attiva: per l'input A  $u_1=1$  e  $u_2=0$ , mentre per l'input B  $u_1=0$  e  $u_2=1$ .

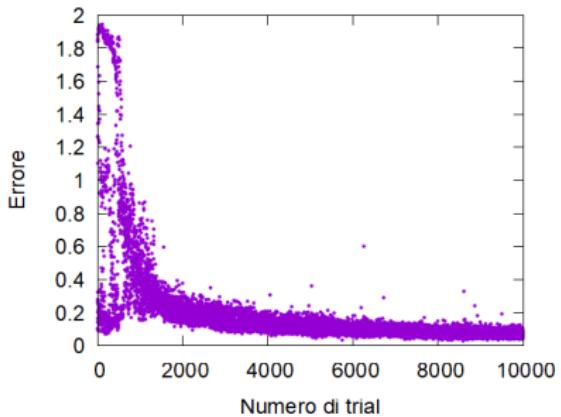
# Indice



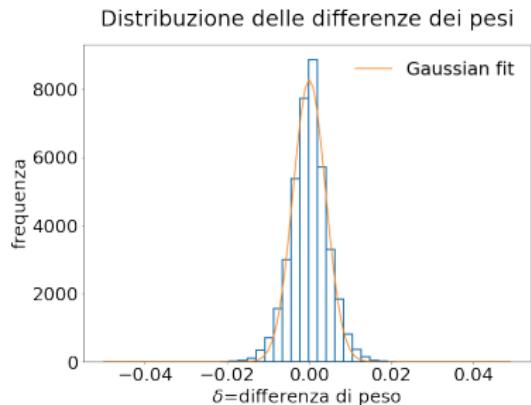
UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

1. Modelli e metodi
2. Implementazione numerica
3. Risultati

# Addestramento

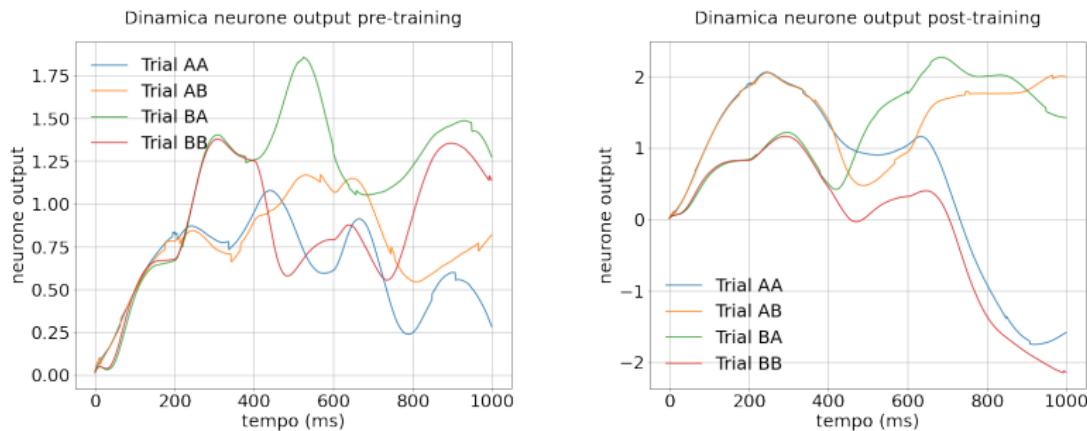


Dopo circa 2000 trial, l'errore presenta una chiara convergenza.



Variazione media relativa dei pesi

$$\left\langle \left| \frac{\Delta J_{i,j}}{J_{i,j} \text{ ini}} \right| \right\rangle \sim 38\%$$



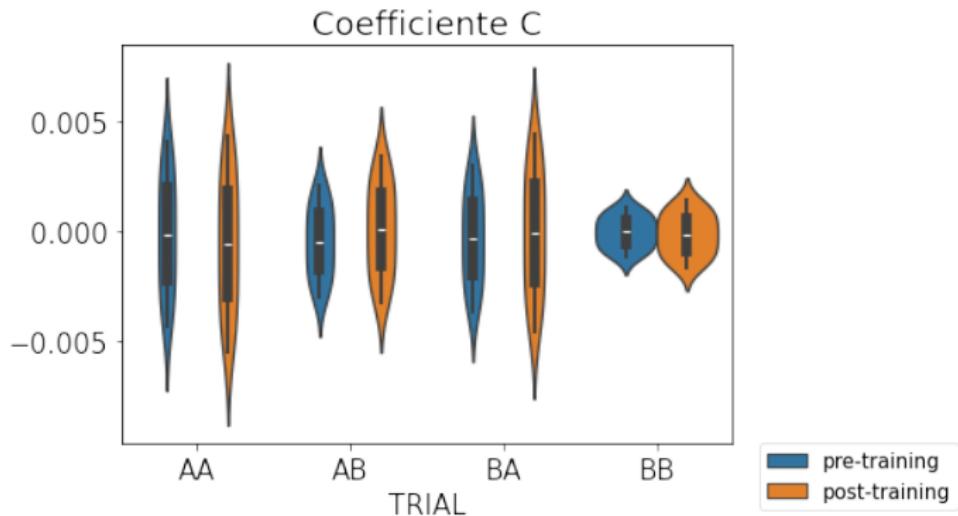
Prima dell'addestramento la risposta dipende dalla natura del primo stimolo.  
Dopo il training la rete è in grado di discriminare tra i 4 tipi di trial, passando da una rappresentazione specifica dello stimolo a una rappresentazione specifica della risposta.

# Selettività mista

La natura del task richiede che alcuni neuroni della rete abbiano una forma di “selettività mista non-lineare”.

Si valuta la dipendenza dell'attivazione dei neuroni dagli stimoli

$$x_i(t) = A_i u_1(t) + B_i u_2(t) + C_i u_1(t) \cdot u_2(t) + D_i \quad (3)$$



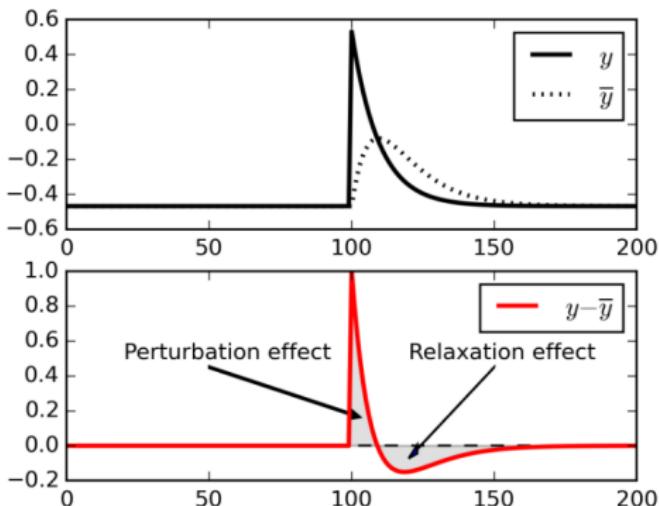
# Conclusioni



- 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.
- La distribuzione complessiva dei pesi è rimasta pressoché invariata, nonostante i singoli pesi abbiano subito variazioni significative, con un cambiamento medio del 38%.
- I profili di attivazione dei singoli neuroni hanno mostrato cambiamenti anche drastici.
- Si osserva il passaggio da una rappresentazione specifica dello stimolo a una rappresentazione specifica della risposta, in grado di discriminare i 4 tipi di trial.

Grazie per l'attenzione

# Effetto di rilassamento



Sottrarre una media mobile dall'attività neuronale isola temporaneamente una perturbazione esterna, ma nel tempo la perturbazione contribuisce alla media mobile.

# Algoritmo Miconi

1. Si inizializzano le matrici dei pesi  $J$  e  $B$  casualmente;
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;
4. Si aggiunge al potenziale di ciascun neurone una perturbazione casuale e indipendente;
5. Per ciascuna connessione ad ogni passo temporale si incrementa l'eligibility trace di  $\Delta E_{i,j}(t) = S(r_j(t-1)(x_i(t) - \bar{x}_i))$ ;
6. Al termine del "trial" viene emessa una ricompensa  $R$  e si calcola un "reward prediction error"  $R - \bar{R}$ .  
Si calcola l'effettivo cambio di pesi  $\Delta J_{i,j} = \eta E_{i,j}(R - \bar{R})\bar{R}$ ; per stabilizzare l'addestramento si limitano le modifiche dei pesi con una soglia;
7. Si azzera l'eligibility trace e si ricomincia dal punto 2 fino al completamento del processo di addestramento.

# Implementazione inizializzazione



- Si inizializza  $J$  con distribuzione normale a media 0 e varianza  $g^2/N$ ,  $B_{k,i}$  con distribuzione uniforme nell'intervallo [-1;1],  $x_i$  con distribuzione uniforme nell'intervallo [-0.1;0.1].
- Si fissano le costanti  $N = 200$ ,  $M = 2$ ,  $\tau = 30\text{ms}$ ,  $g = 1.5$  (regime early chaotic).
- I primi 4 neuroni mantengono un valore costante di attivazione  $x = 1$  durante tutto il trial.
- L'output della rete coincide con la risposta del neurone  $N$ -esimo; questo non è influenzato dagli input, dunque  $B_{N,k} = 0$ .
- Si aggiungono agli  $x_i$  le perturbazioni esplorative  $n_i$ , prese da una distribuzione uniforme in [-0.5, 0.5] e applicate a ciascun neurone casualmente e indipendentemente con un rate di 1Hz.

# Implementazione learning rule



- Nella learning rule si usa la funzione sopralineare cubica  $S(x) = x^3$ .
- La media mobile a breve termine  $\bar{x}_i$  si aggiorna con la formula (con  $\beta=0.05$ )

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

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

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

- L'effettivo cambio di pesi avviene con learning rate  $\eta = 0.1$ , viene moltiplicato per  $\bar{R}$  per modularne l'intensità.
- Per stabilizzare l'addestramento, impostiamo una soglia  $\theta = 10^{-3}$  per i  $\Delta J$ .

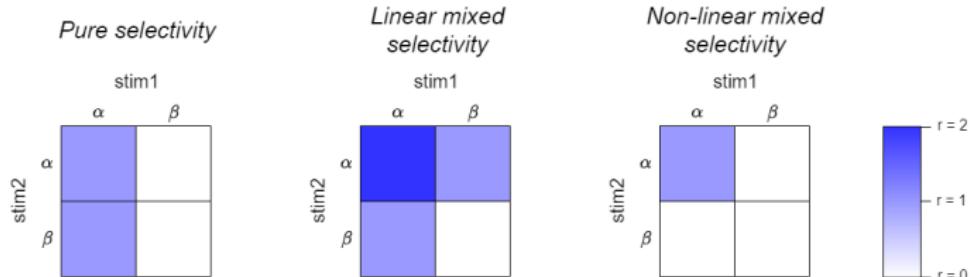
# Selettività mista

A scopo esemplificativo consideriamo un semplice task in cui vengono sottoposti due stimoli  $\alpha$  o  $\beta$  ad un neurone, si ha:

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

**Linear mixed selectivity:** il neurone risponde +1 ogni volta in cui compare uno stimolo  $\alpha$ . La risposta neuronale è descritta da  $r = \delta(\text{stim1} = \alpha) + \delta(\text{stim2} = \alpha)$

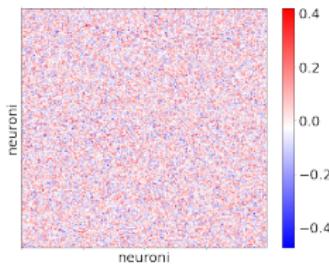
**Nonlinear mixed selectivity:** il neurone risponde +1 quando entrambi gli stimoli sono  $\alpha$ . La risposta è descritta da  $r = \delta(\text{stim1} = \alpha) \cdot \delta(\text{stim2} = \alpha)$



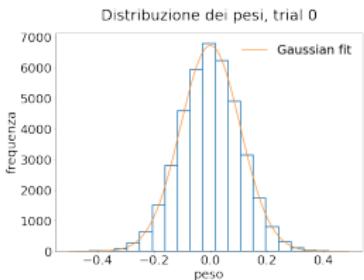
# Variazione dei pesi



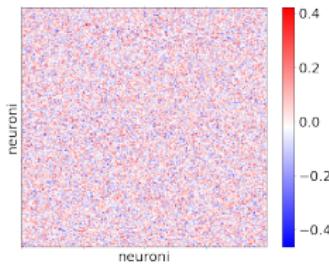
Matrice di connessione, trial 0



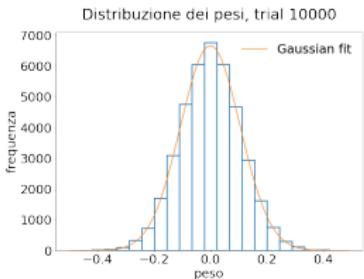
Distribuzione dei pesi, trial 0



Matrice di connessione, trial 10000

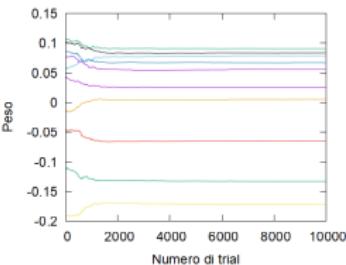


Distribuzione dei pesi, trial 10000



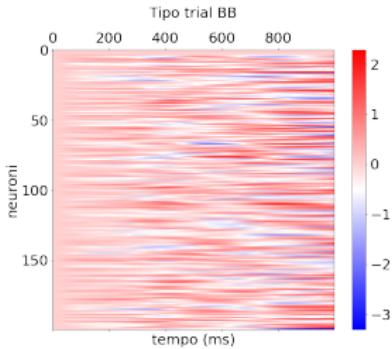
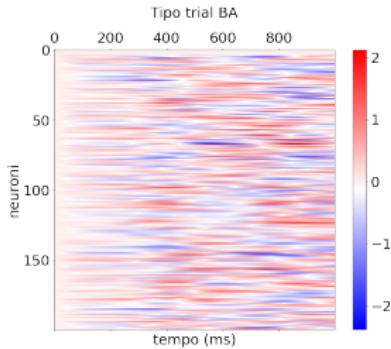
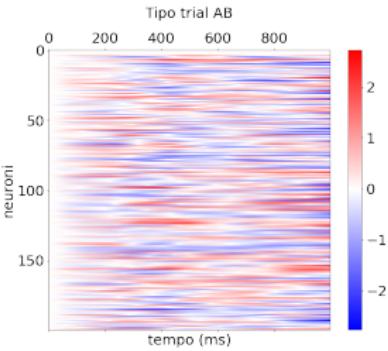
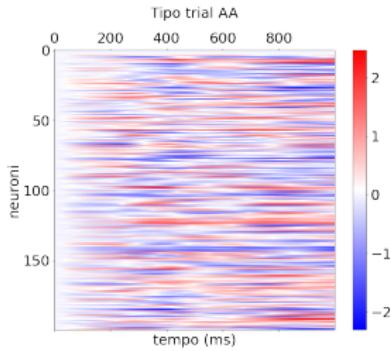
La distribuzione dei pesi resta pressoché invariata, gli effetti dell'addestramento si distribuiscono su 40000 connessioni  
→ effetto collettivo: la rete è in grado di svolgere efficacemente il task.

10 pesi che cambiano maggiormente



J<sub>31,67</sub>  
J<sub>57,6</sub>  
J<sub>57,112</sub>  
J<sub>156,124</sub>  
J<sub>185,103</sub>  
J<sub>154,39</sub>  
J<sub>185,48</sub>  
J<sub>185,146</sub>  
J<sub>195,126</sub>  
J<sub>195,194</sub>

# Variazione di attivazione



La variazione di attività neuronale si concentra soprattutto nella fase finale del trial.