



**UNIVERSITÀ  
DI TORINO**

**Università degli Studi di Torino**

*Master's Degree Thesis*

**Title**

Multi-Task Learning in Neural Networks: A Teacher-Student Setup

**Supervisor**

Osella Matteo

**Co-supervisor**

Milanesio Federico

**Candidate**

Quaglia Eugenio

2024/2025



## Abstract

In questo lavoro di tesi si è analizzato in profondità il *Multi-Task Learning* (MTL), ovvero l'approccio in cui una rete neurale viene addestrata su più compiti simultaneamente, con l'obiettivo di migliorare la generalizzazione del task principale attraverso l'utilizzo di task ausiliari. A partire da intuizioni classiche (Caruana, 1996; Solla et al., 1996), è stato adottato un set-up *teacher-student* per studiare in modo controllato le dinamiche di apprendimento e i benefici teorici e pratici del MTL.

Il lavoro si è articolato in due direzioni principali: lo sviluppo di un modello matematico in grado di descrivere analiticamente l'evoluzione dell'errore di generalizzazione, e la validazione numerica tramite simulazioni estensive. In particolare, è stato dimostrato che esiste sempre un valore ottimale del parametro  $\beta$ , che pondera l'importanza relativa dei task ausiliari, e che questo minimo è robusto rispetto ai parametri di rete. È stato inoltre evidenziato che, in contesti in cui i task ausiliari non introducono rumore spurio, l'aumento del loro numero comporta un vantaggio crescente, con un miglioramento che tende asintoticamente all'ottimo teorico.

È stato poi approfondito l'effetto della correlazione tra le uscite dei task, mostrando — anche in casi di correlazione totale — un vantaggio non banale in termini di generalizzazione. Questo risultato, apparentemente controintuitivo, è stato confermato sia analiticamente sia attraverso simulazioni numeriche ed è in linea con osservazioni sperimentali della letteratura.

Infine, partendo dal formalismo di Goldt et al. (2019), è stato esteso un modello ODE per descrivere l'evoluzione dinamica dell'errore di generalizzazione in presenza di due uscite e correlazioni arbitrarie tra i task. Le espressioni analitiche ottenute sono risultate in ottimo accordo con i dati simulati, confermando la validità del modello e la sua utilità per comprendere i principi fondamentali del Multi-Task Learning.

# Indice

<b>1</b>	<b>Confronto tra Multi-Task Learning e Single-Task Learning</b>	<b>8</b>
1.1	Differenza tra multi task learning e single task learning . . . . .	8
1.2	Un esempio pratico: 1D-DOORS . . . . .	9
1.2.1	Il problema . . . . .	9
1.2.2	Risultati . . . . .	11
<b>2</b>	<b>Setup teacher-student e problema MTL</b>	<b>13</b>
2.1	Formalizzazione del problema di apprendimento multitask . . . . .	13
2.1.1	Setup del problema . . . . .	13
2.1.2	Architettura della rete teacher-student . . . . .	13
2.1.3	Funzione di perdita e ruolo del parametro $\beta$ . . . . .	14
2.1.4	Algoritmo di apprendimento e aggiornamento dei pesi . . . . .	14
2.2	Generazione e struttura del dataset sintetico . . . . .	15
2.2.1	Campionamento degli input . . . . .	15
2.2.2	Etichettatura tramite rete teacher . . . . .	15
2.2.3	Divisione del dataset . . . . .	16
2.2.4	Proprietà delle etichette . . . . .	16
2.2.5	Distribuzione delle etichette . . . . .	16
2.3	Metriche di valutazione delle performance . . . . .	16
2.3.1	Errore sul task principale . . . . .	16
2.3.2	Valutazione del multitask learning . . . . .	17
<b>3</b>	<b>Ruolo di <math>\beta</math>, del numero di task e della correlazione nel multitask learning</b>	<b>18</b>
3.1	Dipendenza dell' <i>Advantage Score</i> dal parametro $\beta$ . . . . .	18
3.2	Effetto del numero di task ausiliari sull' <i>Advantage Score</i> . . . . .	19
3.3	Impatto della correlazione tra uscite sull' <i>Advantage Score</i> . . . . .	20
<b>4</b>	<b>Comportamento dell'SGD in una rete shallow con architettura multi-output</b>	<b>23</b>
4.1	Definizione del problema . . . . .	23
4.1.1	Architettura della rete <i>Teacher</i> . . . . .	23
4.1.2	Architettura della rete <i>Student</i> . . . . .	23
4.1.3	Funzione di <i>Training</i> e <i>Generalization Error</i> . . . . .	24
4.2	Derivazione delle equazioni del moto e di $\epsilon_g$ . . . . .	24
4.2.1	Sviluppo di $\epsilon_g$ e delle metriche $Q, R, T$ . . . . .	24
4.2.2	Aggiornamenti discreti dei pesi . . . . .	25
4.2.3	Calcolo di $\frac{dR_{ki}}{dt}$ . . . . .	25
4.2.4	Calcolo di $\frac{dQ_{ki}}{dt}$ . . . . .	26
4.2.5	Sistema di ODE completo . . . . .	29
4.3	Espansione asintotica del sistema per $t \rightarrow \infty$ . . . . .	30
4.3.1	Semplificazione del sistema di ODE . . . . .	30
4.4	Considerazioni su $\langle \epsilon_g \rangle$ quando $K=2$ . . . . .	33
4.4.1	Calcolo esplicito del valore di $\beta_{\min}$ . . . . .	33

4.4.2	Caso $a=b$ . . . . .	34
4.4.3	Caso $a \neq b$ . . . . .	35
4.4.4	Scaling di $\epsilon_g$ . . . . .	37
4.5	Soluzione generale e correlazione . . . . .	40
<b>5</b>	<b>Conclusioni</b>	<b>41</b>
<b>A</b>	<b>Forme esplicite degli integrali sui campi locali</b>	<b>43</b>

# Elenco delle figure

1.1	Addestramento con backpropagation a singolo task (STL) su quattro task che condividono gli stessi input. . . . .	9
1.2	Addestramento con backpropagation multitask (MTL) su quattro task che condividono gli stessi input. . . . .	10
1.3	Esempi di porte singole e doppie nel dominio 1D-DOORS. . . . .	11
2.1	Grafico della funzione di attivazione $g(x) = \text{erf}(\frac{x}{\sqrt{2}})$ . . . . .	14
2.2	Schematizzazione del setup teacher student . . . . .	15
2.3	Distribuzione delle labels . . . . .	16
3.1	Andamento del vantaggio ( <i>Advantage</i> ) medio in funzione del parametro $\beta$ . . . . .	19
3.2	Andamento del vantaggio rispetto al Single Task Learning (STL) in funzione del parametro $\beta$ , per diversi numeri $n$ di task ausiliarie. . . . .	20
3.3	Andamento dell'advantage score in funzione della correlazione . . . . .	22
4.1	Heatmap degli overlap $R_{i,k}$ e $Q_{i,k}$ e $v_k$ e $u_k$ . . . . .	31
4.2	Scaling di $\epsilon_g$ in funzione di $a$ . . . . .	35
4.3	Scaling di $\epsilon_g$ in funzione di $r$ . . . . .	37
4.4	Scaling di $\epsilon_g$ in funzione di $\eta$ . . . . .	38
4.5	Scaling di $\epsilon_g$ in funzione di $K$ . . . . .	39

## Introduzione

L'apprendimento è un processo fondamentale attraverso il quale gli esseri viventi acquisiscono conoscenze, competenze e comportamenti in risposta agli stimoli provenienti dall'ambiente che li circonda. Nel mondo reale, caratterizzato da complessità, variabilità e incertezza, la capacità di apprendere in modo efficiente ed adattivo, consente agli individui di affrontare nuove sfide, risolvere problemi e migliorare continuamente le proprie capacità.

Nell'essere umano, l'apprendimento è guidato da meccanismi cognitivi e neurobiologici che permettono di integrare esperienze, generalizzare da esempi passati e trasferire abilità acquisite in contesti differenti. I processi di memoria, attenzione e rinforzo cooperano per ottimizzare le rappresentazioni interne e adattare i comportamenti futuri. Forse è proprio la somiglianza tra i numerosi compiti che impariamo a permetterci di apprendere così tanto con così poca esperienza. Si impara a giocare a tennis in un mondo che ci chiede di imparare molte altre cose. Si impara anche a camminare, a correre, a saltare, a fare esercizio, ad afferrare, a lanciare, a colpire, a riconoscere oggetti, a prevedere traiettorie, a riposare, a parlare, a leggere, a studiare, a esercitarsi, ecc. Questi compiti non sono identici — correre nel tennis è diverso dal correre in pista — ma sono correlati. Forse le somiglianze tra le migliaia di compiti che impariamo sono ciò che ci permette di apprendere ognuno di essi, incluso il tennis, con così pochi dati di addestramento.

Il Multi-Task Learning (MTL) si pone come naturale estensione di questi principi: anziché addestrare un singolo modello per ciascun compito in isolamento, l'MTL propone di apprendere simultaneamente più compiti correlati, condividendo una rappresentazione comune. In questo modo, le informazioni derivate da un task fungono da vincoli di regolarizzazione per gli altri, migliorando la generalizzazione complessiva, come sostenuto da Rich Caruana (1997), le informazioni acquisite da un task aiutano l'apprendimento degli altri. In un contesto artificiale, addestrare un modello da zero su un singolo compito complesso — ad esempio riconoscere oggetti in immagini ad alta risoluzione — richiede enormi quantità di dati. Tuttavia, se lo stesso modello impara simultaneamente a riconoscere contorni, forme, texture, ombre, testi, orientamento, dimensioni e distanza, la conoscenza condivisa migliora la robustezza e la generalizzazione.

Lo scopo di questo lavoro è quello di analizzare in profondità il paradigma del *Multi-Task Learning* (MTL), con particolare attenzione alla comprensione teorica delle sue dinamiche e dei benefici che può apportare all'apprendimento supervisionato. L'analisi si concentra su vari aspetti fondamentali: in primo luogo, lo studio del ruolo del parametro di ponderazione  $\beta$ , che regola il bilanciamento tra il task principale e quelli ausiliari, con l'obiettivo di individuarne un valore ottimale in grado di minimizzare l'errore di generalizzazione. In secondo luogo, si indaga l'influenza del numero di task ausiliari in scenari controllati, valutando in che misura l'aggiunta di ulteriori compiti possa migliorare le performance di apprendimento. Viene inoltre affrontato il problema della correlazione tra i target dei vari task, analizzando se e come l'informazione condivisa possa risultare ridondante oppure vantaggiosa. Per supportare queste analisi, viene costruito un modello matematico che descrive l'evoluzione temporale dell'errore di generalizzazione, basato su un sistema di equazioni differenziali ispirato a risultati della teoria statistica dell'apprendimento. Infine, i risultati teorici vengono confrontati con simulazioni numeriche per verificare la validità delle assunzioni adottate e l'aderenza dei modelli analitici al comportamento empirico osservato. Nel complesso, il lavoro si propone di fornire una base teorica solida per l'interpretazione dei meccanismi del MTL, contribuendo alla definizione di condizioni in cui questa tecnica può essere applicata in modo efficace e giustificato.

# Capitolo 1

## Confronto tra Multi-Task Learning e Single-Task Learning

### 1.1 Differenza tra multi task learning e single task learning

In questo lavoro si utilizza il termine *task* per indicare il compito che deve essere appreso a partire da un insieme di esempi di addestramento, detti *training set*.

Ciascun esempio è associato a una *label*  $y_n$  e rappresentato da un vettore contenente  $k$  *feature*:

$$X_n = (X_{1,n}, X_{2,n}, X_{3,n}, \dots, X_{k,n});$$

L'intero dataset di addestramento si scrive come:

$$TrainingSet = \{(y_n, X_n), n = 1, \dots, N\}.$$

Quando si parla di *supervised learning*, si intendono modelli in cui, per ogni esempio  $X_n$ , è nota la rispettiva *label*  $y_n$ . L'obiettivo è apprendere una funzione in grado di predire accuratamente i valori di  $y$  per nuovi input  $X$ , cioè sul *test set*.

Il *task principale* è il compito di riferimento su cui si valuta la qualità dell'apprendimento del modello, ovvero quello per cui si misurano le performance finali. I *task ausiliari*, invece, sono obiettivi aggiuntivi introdotti durante l'addestramento con lo scopo di facilitare o migliorare l'apprendimento del task principale. Pur non essendo rilevanti per la valutazione finale, questi task secondari aiutano la rete a imparare rappresentazioni interne più generali e stabili.

Per esempio, supponiamo di voler addestrare una rete neurale per riconoscere l'animale raffigurato in una fotografia (ad esempio: gatto, cane, cavallo). Questo costituisce il *task principale*, ovvero il compito su cui si valuta la performance del modello e che rappresenta l'obiettivo finale dell'addestramento. Durante la fase di training, è possibile introdurre anche uno o più *task ausiliari*, che hanno lo scopo di facilitare l'apprendimento della rete. Ad esempio, si può chiedere al modello di predire la razza dell'animale (es. bulldog, siamese, pastore tedesco) oppure la sua posizione all'interno dell'immagine. Questi compiti ausiliari aiutano la rete a imparare rappresentazioni interne più significative e generalizzabili, migliorando indirettamente le prestazioni sul task principale.

Per comprendere meglio la differenza tra Multi Task Learning (MTL) e Single Task Learning (STL) consideriamo due configurazioni diverse. Nella prima, illustrata nella figura 1.1, si utilizzano quattro reti neurali distinte, ciascuna con otto ingressi identici e un solo output. In questa configurazione, ogni rete viene addestrata separatamente tramite backpropagation. Poiché non esiste alcuna connessione tra le reti, ciò che una rete apprende non ha alcun impatto sulle altre. Questo tipo di addestramento indipendente è noto come *Single Task Learning* (STL).

La seconda configurazione, mostrata nella Figura 1.2, impiega un'unica rete con gli stessi otto input, ma dotata di quattro output. Questi output condividono un livello nascosto comune completamente connesso. Durante l'addestramento, il *backpropagation* viene applicato simultaneamente a tutti gli output.

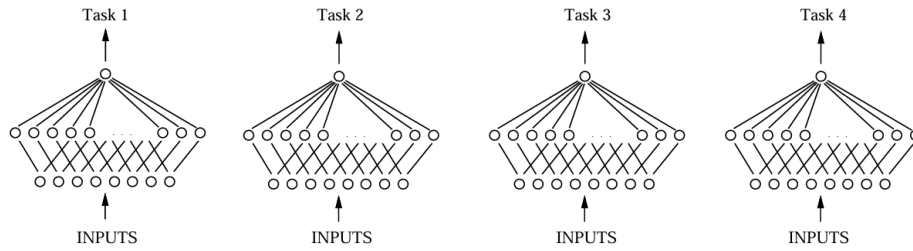


Figura 1.1: Addestramento con backpropagation a singolo task (STL) su quattro task che condividono gli stessi input.

Grazie alla condivisione del livello nascosto, le rappresentazioni interne utili a un task possono essere riutilizzate anche dagli altri. È proprio questa condivisione della rappresentazione, durante l'addestramento parallelo, che costituisce il principio fondante del *Multitask Learning* (MTL).

Il multitask learning, dunque, non si identifica con un unico algoritmo, ma rappresenta un insieme di strategie, tecniche e criteri di apprendimento. Si tratta di una modalità di trasferimento induttivo che privilegia l'apprendimento congiunto di più funzioni obiettivo, facendo leva su una struttura condivisa in grado di generalizzare meglio. Nei modelli a backpropagation, l'MTL consente che i neuroni del livello nascosto sviluppino feature utili a più tasks contemporaneamente, contribuendo a una maggiore efficienza e profondità dell'apprendimento. Inoltre, questa architettura permette che alcune unità si specializzino in modo selettivo per certi tasks, mentre gli altri possono di fatto ignorarle riducendo i pesi associati. Questo meccanismo favorisce sia la cooperazione che la specializzazione tra tasks, rendendo l'apprendimento più flessibile e robusto.

## 1.2 Un esempio pratico: 1D-DOORS

### 1.2.1 Il problema

Nella sezione seguente riportiamo un esempio tratto dal lavoro di tesi di Rich Caruana (1996), con l'obiettivo di mostrare al lettore come i principi finora esposti possano essere applicati anche a dataset reali. L'esempio considerato è quello del dataset *1D-DOORS*, in cui i compiti principali consistono nell'individuazione delle maniglie e nel riconoscimento del tipo di porta (singola o doppia), a partire da immagini acquisite da una videocamera a colori montata su un robot.

Sono state raccolte circa un migliaio di immagini mentre un robot si muoveva in modo quasi casuale al quinto piano del Wean Hall presso la Carnegie Mellon University. Dalle 1000 immagini, sono state selezionate le 402 in cui era visibile una maniglia. Per la maggior parte degli ingressi esistono due o più immagini. Le immagini sono state raggruppate in base alla porta rappresentata. Due terzi dei gruppi sono stati utilizzati per l'addestramento, mentre il restante terzo è stato destinato al test. Il campionamento è stato effettuato a livello di ingressi e non di singole immagini, in modo tale che il set di test contenesse porte che la rete non aveva mai visto durante la fase di addestramento. Questo processo ha prodotto un training set composto da circa 270 immagini.

Il problema è stato semplificato utilizzando strisce orizzontali delle immagini, una per il canale verde e una per il canale blu. Ogni striscia è larga 30 pixel (ottenuta tramite smoothing gaussiano dell'immagine originale larga 150 pixel) ed è estratta all'altezza verticale corrispondente alla maniglia della porta.

Sono stati utilizzati dieci compiti:

- posizione orizzontale della maniglia
- posizione orizzontale del centro della porta
- posizione orizzontale dello stipite sinistro
- larghezza dello stipite sinistro

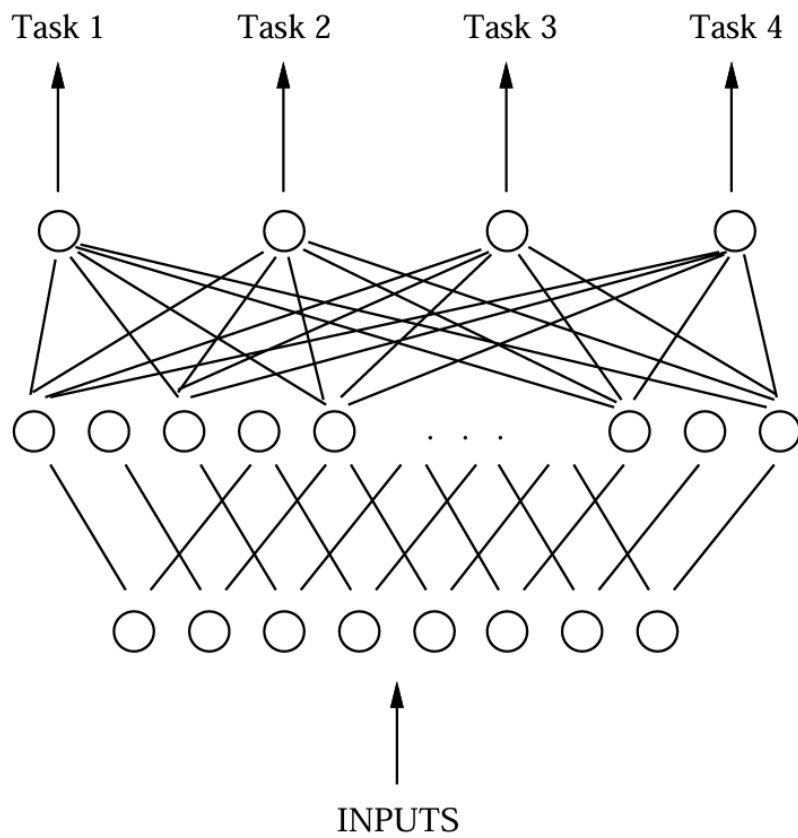


Figura 1.2: Addestramento con backpropagation multitask (MTL) su quattro task che condividono gli stessi input.

- posizione orizzontale del bordo sinistro della porta
- classificazione tra porta singola o doppia
- larghezza totale della porta
- posizione orizzontale dello stipite destro
- larghezza dello stipite destro
- posizione orizzontale del bordo destro della porta



Figura 1.3: Esempi di porte singole e doppie nel dominio 1D-DOORS.

Essendo un dominio reale, i segnali di addestramento per questi compiti sono stati acquisiti manualmente. Un operatore ha utilizzato un mouse per cliccare sulle caratteristiche rilevanti in ciascuna immagine, sia nel set di addestramento sia in quello di test. Dal momento che era già necessario annotare manualmente le immagini per raccogliere i segnali relativi ai due compiti principali, ottenere anche quelli per i compiti ausiliari non ha comportato uno sforzo aggiuntivo significativo.

### 1.2.2 Risultati

Il confronto è stato effettuato esclusivamente sui due compiti principali: la localizzazione della maniglia e il riconoscimento del tipo di porta. Il metodo *Single Task Learning* (STL) è stato testato utilizzando reti con 6, 24 e 96 unità nascoste, mentre il *Multi Task Learning* (MTL) è stato testato su reti con 120 unità nascoste. I risultati di dieci prove per ciascun approccio (STL e MTL) sono riportati nella Tabella 1.1.

L'MTL mostra una capacità di generalizzazione superiore del 20–30% rispetto allo STL, anche nel confronto con la migliore tra le tre configurazioni STL. È importante osservare che i pattern di addestramento utilizzati per STL e MTL sono identici, fatta eccezione per il fatto che, nel caso MTL, ogni pattern include anche segnali di addestramento aggiuntivi. È proprio l'informazione contenuta in questi segnali supplementari a permettere al livello nascosto della rete di apprendere rappresentazioni interne più efficaci per l'individuazione delle maniglie e la classificazione dei tipi di porta.

Il dominio 1D-DOORS, invece, si basa su dati reali raccolti da una videocamera installata su un robot che si muoveva in un corridoio reale. Nonostante si sia cercato di mantenere alta la difficoltà del problema (ad esempio, evitando di mantenere il robot parallelo alle pareti, lasciando variare la distanza dalle porte e le condizioni di illuminazione, e acquisendo alcuni segnali di addestramento con un trackball su un portatile mentre si era a bordo di un autobus), si tratta comunque di un dominio costruito appositamente per dimostrare l'efficacia dell'MTL. La vera sfida sarà capire quanto bene possa funzionare l'MTL in un dominio reale che non sia stato progettato con questo scopo.

Task	STL (migliore)	MTL (120 HU)
Localizzazione maniglia	<u>0.081</u>	<b>0.062</b> (-23.5%)*
Tipo di porta	<u>0.086</u>	<b>0.059</b> (-31.4%)*

Tabella 1.1: Prestazioni di STL e MTL sui due compiti principali nel dominio 1D-DOORS. I valori sottolineati nelle colonne STL indicano le configurazioni migliori. Le differenze statisticamente significative al livello 0.05 o migliore sono indicate con un asterisco.

## Capitolo 2

# Setup teacher-student e problema MTL

### 2.1 Formalizzazione del problema di apprendimento multitask

#### 2.1.1 Setup del problema

Nel nostro approccio affrontiamo un problema di regressione supervisionata con struttura multi-task. L'obiettivo è apprendere una funzione del tipo

$$f : \mathbb{R}^N \rightarrow (y, \alpha),$$

dove  $y \in \mathbb{R}$  è il task principale e  $\alpha \in \mathbb{R}$  è il task ausiliario. L'apprendimento avviene su un dataset di  $P$  campioni:

$$\mathcal{D} = \{(x^\mu, y^\mu, \alpha^\mu)\}_{\mu=1}^P, \quad x^\mu \sim \mathcal{N}(0, I_N).$$

#### 2.1.2 Architettura della rete teacher-student

Le etichette  $y^\mu$  e  $\alpha^\mu$  sono generate da una rete neurale fissata (il *teacher*) con  $M$  neuroni nascosti e funzione di attivazione non lineare  $g : \mathbb{R} \rightarrow \mathbb{R}$ . Nel nostro caso,  $g(z) = \text{erf}(z/\sqrt{2})$ , riportata nella figura (2.1). La rete del teacher ha architettura a due layer:

$$\hat{y}(x) = \sum_{m=1}^M v_m^* g(w_m^* \cdot x), \quad \hat{\alpha}(x) = \sum_{m=1}^M u_m^* g(w_m^* \cdot x),$$

dove  $w_m^* \in \mathbb{R}^N$  sono i pesi della parte nascosta (condivisi tra le due uscite) e  $v_m^*, u_m^*$  sono pesi di lettura specifici per ciascun output. I pesi del teacher sono inizializzati casualmente da una distribuzione normale standard e restano fissi durante tutto l'addestramento dello studente.

La rete dello *studente* ha la stessa struttura architetturale, ma con  $K$  neuroni nascosti. Si può verificare che  $K > M$ , rendendo lo studente *over-parametrizzato* rispetto al modello generativo, ma nel nostro caso avremo sempre  $K = M$ . Le uscite dello studente sono:

$$y(x) = \sum_{k=1}^K v_k g(w_k \cdot x), \quad \alpha(x) = \sum_{k=1}^K u_k g(w_k \cdot x),$$

dove  $w_k \in \mathbb{R}^N$  e i pesi  $v_k, u_k$  sono appresi tramite stochastic gradient descent (SGD).

La geometria delle reti è illustrata nella figura (2.2)

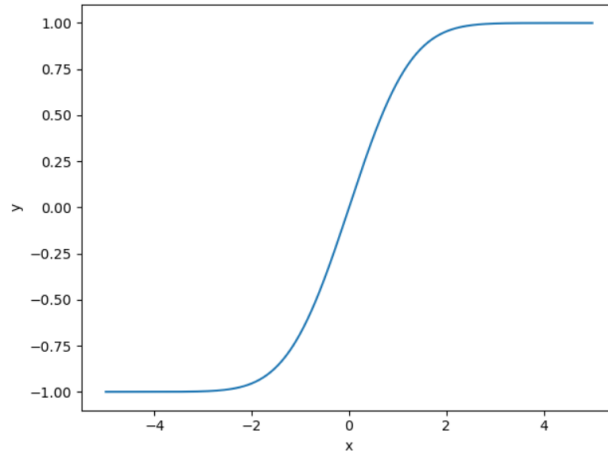


Figura 2.1: Grafico della funzione di attivazione  $g(x) = \text{erf}(\frac{x}{\sqrt{2}})$ .

### 2.1.3 Funzione di perdita e ruolo del parametro $\beta$

La funzione di perdita utilizzata durante l'addestramento sarà:

$$\mathcal{L}_{train}(\theta) = \mathcal{L}_{main}(\theta) + \beta \sum_{n=1}^{N_{aux}} \mathcal{L}_{aux}^{(n)}(\theta)$$

dove  $\beta > 0$  è un iperparametro che bilancia il contributo del task ausiliario, e  $\theta$  rappresenta l'insieme dei parametri della rete *student*. Le funzioni di perdita  $\mathcal{L}_{main}$  e  $\mathcal{L}_{aux}$  sono, nel nostro caso, entrambe *MSE*, ossia:

$$MSE(x_i, \hat{x}) = (x_i - \hat{x})^2$$

L'uso di neuroni condivisi consente alla rete di apprendere una rappresentazione comune dai dati, utile per entrambi i task. Il task ausiliario agisce da regolarizzatore, guidando l'apprendimento dello studente soprattutto nelle regioni dove il teacher fornisce maggiore confidenza (ovvero  $\alpha^\mu \approx 1$ ).

### 2.1.4 Algoritmo di apprendimento e aggiornamento dei pesi

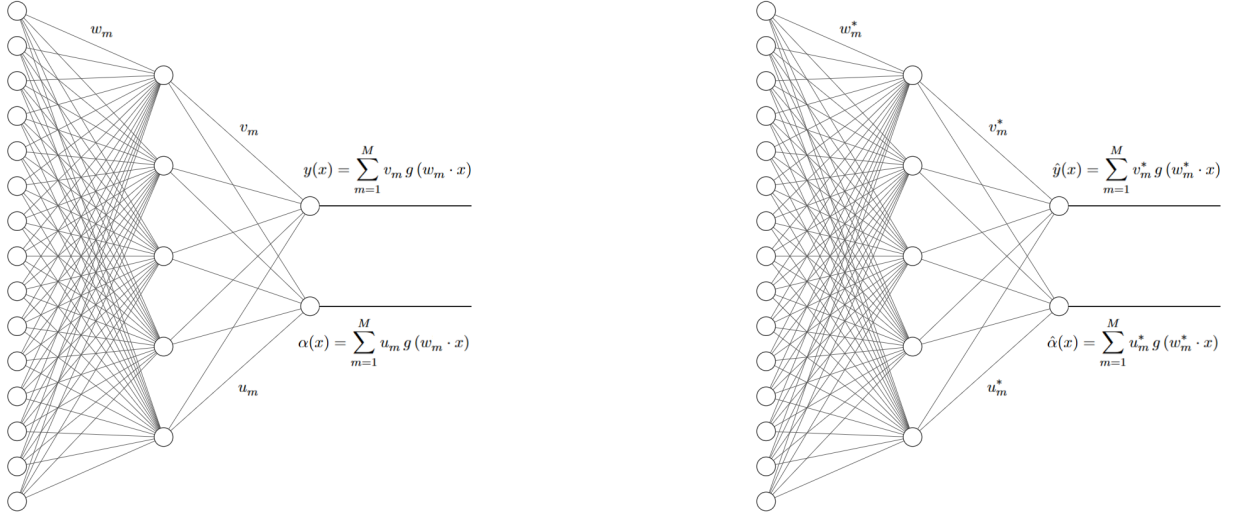
La rete student viene addestrata facendo *online learning*, ossia aggiornando i parametri dopo il passaggio di ogni singolo dato. Lo *Stochastic Gradient Descent* è una tecnica di apprendimento basata sulla discesa del gradiente che utilizza campioni casuali per aggiornare i parametri.

L'aggiornamento per ciascun peso  $w_j$  al passo  $t$  è:

$$w_j^{(t+1)} = w_j^{(t)} - \eta \frac{\partial \mathcal{L}(y^{(i)}, \hat{y}^{(i)})}{\partial w_j},$$

dove:

- $\eta$  è il *learning rate*;
- $\mathcal{L}(y, \hat{y})$  è la funzione di perdita (nel nostro caso MSE);
- $\hat{y}^{(i)}$  è la predizione del modello sul campione  $x^{(i)}$ ;
- la media sul mini-batch approssima il gradiente sull'intero dataset.



(a) Architettura della rete student

(b) Architettura della rete teacher

Figura 2.2: Schematizzazione del setup teacher student

Le regole di aggiornamento sono:

$$\begin{aligned}
 w_k^{t+1} &= w_k^t - \eta_w \frac{\partial \mathcal{L}^\mu}{\partial w_k}, \\
 v_k^{y, t+1} &= v_k^{y, t} - \eta_v \frac{\partial \mathcal{L}^\mu}{\partial v_k^y}, \\
 v_k^{\alpha, t+1} &= v_k^{\alpha, t} - \eta_v \frac{\partial \mathcal{L}^\mu}{\partial v_k^\alpha},
 \end{aligned}$$

dove  $\mathcal{L}^\mu$  è il valore della loss calcolato sul singolo campione. Le derivate coinvolgono  $g'(z) = \frac{2}{\sqrt{\pi}} e^{-z^2}$ , la derivata della funzione  $\text{erf}(z/\sqrt{2})$ .

## 2.2 Generazione e struttura del dataset sintetico

### 2.2.1 Campionamento degli input

I dati utilizzati nel presente lavoro sono stati generati sinteticamente. In primo luogo, le coordinate dei vettori di input  $\vec{x}_i \in \mathbb{R}^N$  sono state campionate da una distribuzione normale standard:

$$x_j \sim \mathcal{N}(0, 1) \quad \text{per } j = 1, \dots, N.$$

### 2.2.2 Etichettatura tramite rete teacher

Una volta generati gli input, le etichette sono ottenute mediante una rete neurale *teacher*  $T$  inizializzata casualmente e tenuta fissa per tutto il processo. Il mapping definito dalla rete è il seguente:

$$T : X \rightarrow (y, \alpha^n) \quad \text{per } n = 1, \dots, N_{\text{aux}}.$$

Si utilizza una semplice *forward pass* per etichettare i campioni, sfruttando lo stesso strato nascosto condiviso per il task principale  $y$  e quelli ausiliari  $\alpha^n$ .

### 2.2.3 Divisione del dataset

Nel nostro caso, sono stati generati 64000 campioni. I dati sono stati poi suddivisi come segue:

- **Training set:**  $N_{\text{train}} = 60\,000$ ;
- **Validation set:**  $N_{\text{val}} = 2\,000$ ;
- **Test set:**  $N_{\text{test}} = 2\,000$ .

### 2.2.4 Proprietà delle etichette

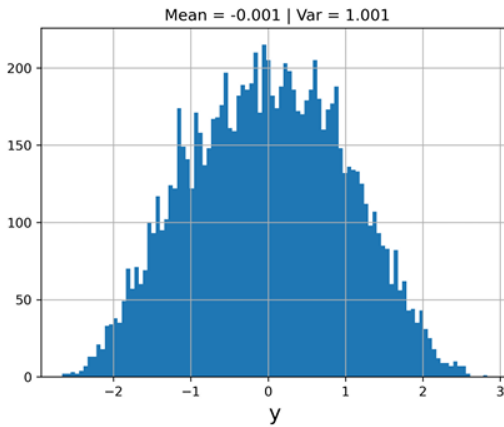
Poiché le etichette ausiliarie  $\alpha^n$  e quella principale  $y$  derivano da una base comune (i neuroni dello strato nascosto), esse risultano correlate. Questo aspetto è centrale per l'apprendimento multitask: ci si attende che l'informazione contenuta nei task ausiliari aiuti a migliorare la generalizzazione del task principale.

La forma generale del dataset è dunque:

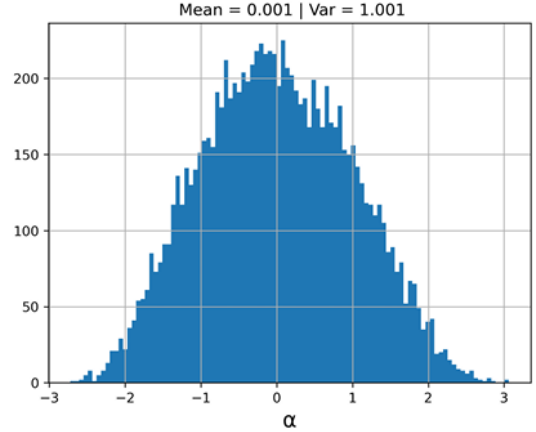
$$\mathcal{D} = \left\{ (\vec{x}_i; y_i, \alpha_i^1, \dots, \alpha_i^{N_{\text{aux}}}) \right\}_{i=1}^N.$$

### 2.2.5 Distribuzione delle etichette

La distribuzione dei target generati è illustrata in Figura 2.3, usando un binning pari a 100.



(a) Distribuzione della label principale  $y$ .



(b) Distribuzione di una delle labels ausiliarie  $\alpha^n$ .

Figura 2.3: Distribuzione delle labels

## 2.3 Metriche di valutazione delle performance

### 2.3.1 Errore sul task principale

Durante la fase di test, il nostro interesse è rivolto esclusivamente al task principale  $\hat{y}$ . La funzione di perdita valutata sui dati di test è definita come:

$$\mathcal{L}_{\text{test}}(\theta) = \frac{1}{2} \|\hat{y} - y(\theta)\|^2$$

Per stimare la bontà della predizione nel problema di regressione, si utilizza l'**errore quadratico medio** (MSE):

$$\text{MSE}(y, \hat{y}) = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (y_i - \hat{y}_i)^2$$

### 2.3.2 Valutazione del multitask learning

Per valutare l'efficacia del *multitask learning* (MTL), confrontiamo i suoi risultati con quelli ottenuti tramite *single-task learning* (STL), che costituisce il nostro baseline.

A tale scopo introduciamo l'**Advantage Score**, definito come:

$$\text{Adv}(\beta) = 1 - \frac{\mathbb{E}[L_{\text{test}}(\beta)]}{\mathbb{E}[L_{\text{test}}(\beta = 0)]}$$

dove:

- $L$  rappresenta il valore di RMSE (*Root Mean Squared Error*);
- $\mathbb{E}[\cdot]$  indica la media su 12 addestramenti indipendenti;
- $L_{\text{test}}(\beta = 0)$  è il punteggio del baseline STL;
- $L_{\text{test}}(\beta)$  è il punteggio del modello MTL addestrato con iperparametro  $\beta$ .

L'**Advantage Score** fornisce una misura relativa di miglioramento prestazionale del MTL rispetto a STL:

- $\text{Adv}(\beta) > 0$ : il modello MTL supera STL in termini di errore di generalizzazione;
- $\text{Adv}(\beta) < 0$ : il modello STL fornisce prestazioni migliori rispetto all'approccio multitask.

## Capitolo 3

# Ruolo di $\beta$ , del numero di task e della correlazione nel multitask learning

Per effettuare quest'analisi è stato addestrato il setup *teacher-student* descritto in precedenza, con i seguenti parametri:

- **Learning rate** : 0.1
- **Input size**: 50
- **Hidden size**: 10
- **Compiti**: task principale + task ausiliario
- **Numero massimo di step**: 60000
- **Early stopping**: attivato con *patience* pari a 5000

L'incertezza sull'**Advantage Score** (rappresentata dalle barre di errore nei grafici) è stata stimata tramite **propagazione dell'errore**, a partire dalla **deviazione standard** calcolata su **12 esperimenti indipendenti**. I punti riportati nei grafici rappresentano la **media** dei valori dell'Advantage Score ottenuti in queste 12 run.

### 3.1 Dipendenza dell'*Advantage Score* dal parametro $\beta$

Il primo risultato ottenuto in questo lavoro consiste nell'osservare l'andamento dell'Advantage Score al variare del parametro  $\beta$ .

- Per  $\beta = 0$ , l'*Advantage Score* è circa 0, come previsto: corrisponde al baseline STL.
- Nei valori bassi di  $\beta$  (fino a circa  $\beta \approx 2$ ), l'*Advantage Score* è positivo e crescente, suggerendo che un utilizzo moderato del *multitask learning* migliora le prestazioni rispetto al *single task learning*.
- Il valore massimo si raggiunge intorno a  $\beta \in [1,2]$ , dove Advantage Score  $> 0.6$ , con incertezze comunque contenute.
- Dopo  $\beta \approx 2$ , l'*Advantage Score* decresce progressivamente, diventando negativo per  $\beta \geq 5$ .

Il risultato più rilevante evidenziato da questo grafico è che, nel nostro setup, esiste un valore ottimale di  $\beta$  per cui il vantaggio è massimo. Scopo di questo lavoro sarà analizzare in dettaglio la dipendenza del valore ottimale di  $\beta$  dai diversi parametri della rete e cercare di fornirne una descrizione matematica.

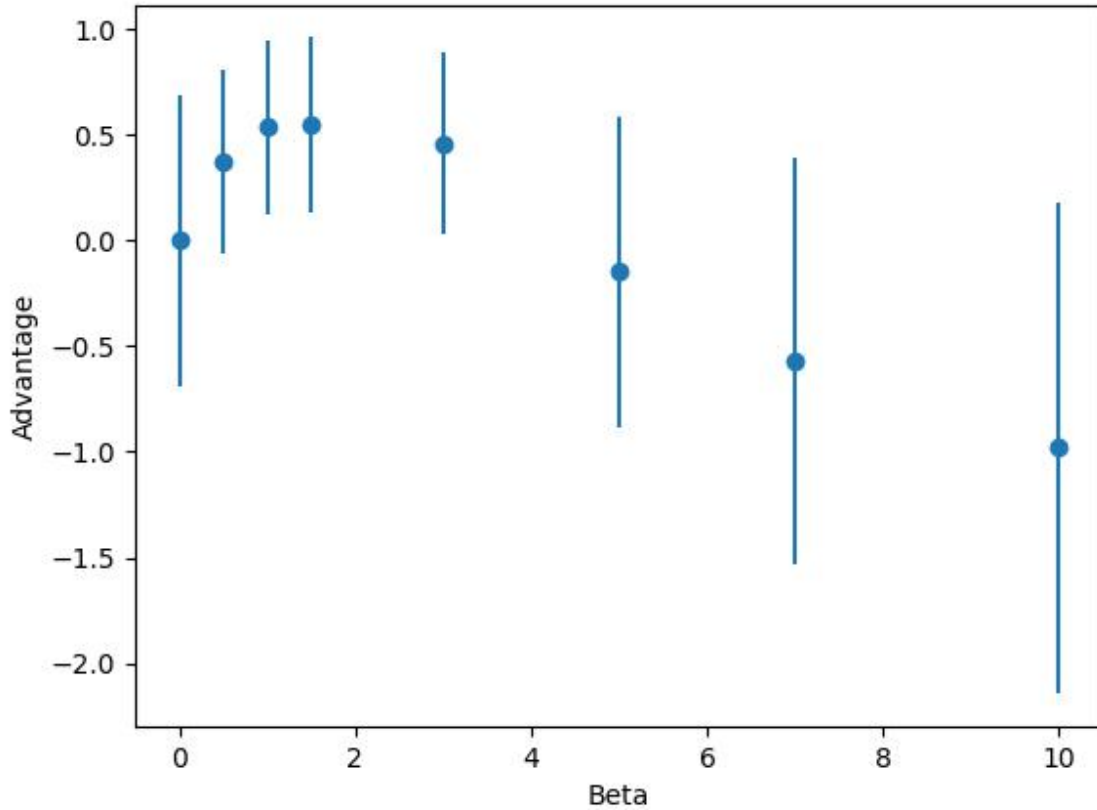


Figura 3.1: Andamento del vantaggio (*Advantage*) medio in funzione del parametro  $\beta$ .

### 3.2 Effetto del numero di task ausiliari sull'*Advantage Score*

In seguito, abbiamo voluto studiare come varia l'*Advantage Score* all'aumentare del numero di uscite ausiliarie.

A tal fine, è stata addestrata una rete identica a quella illustrata in precedenza, con l'unica differenza che il numero di uscite ausiliarie è stato incrementato. Per fare ciò, si è utilizzata la seguente funzione di loss per l'addestramento:

$$\mathcal{L}_{\text{train}}(\theta) = \mathcal{L}_{\text{main}}(\theta) + \beta \sum_{n=1}^{N_{\text{aux}}} \mathcal{L}_{\text{aux}}^{(n)}(\theta)$$

Di seguito è riportato il grafico ottenuto.

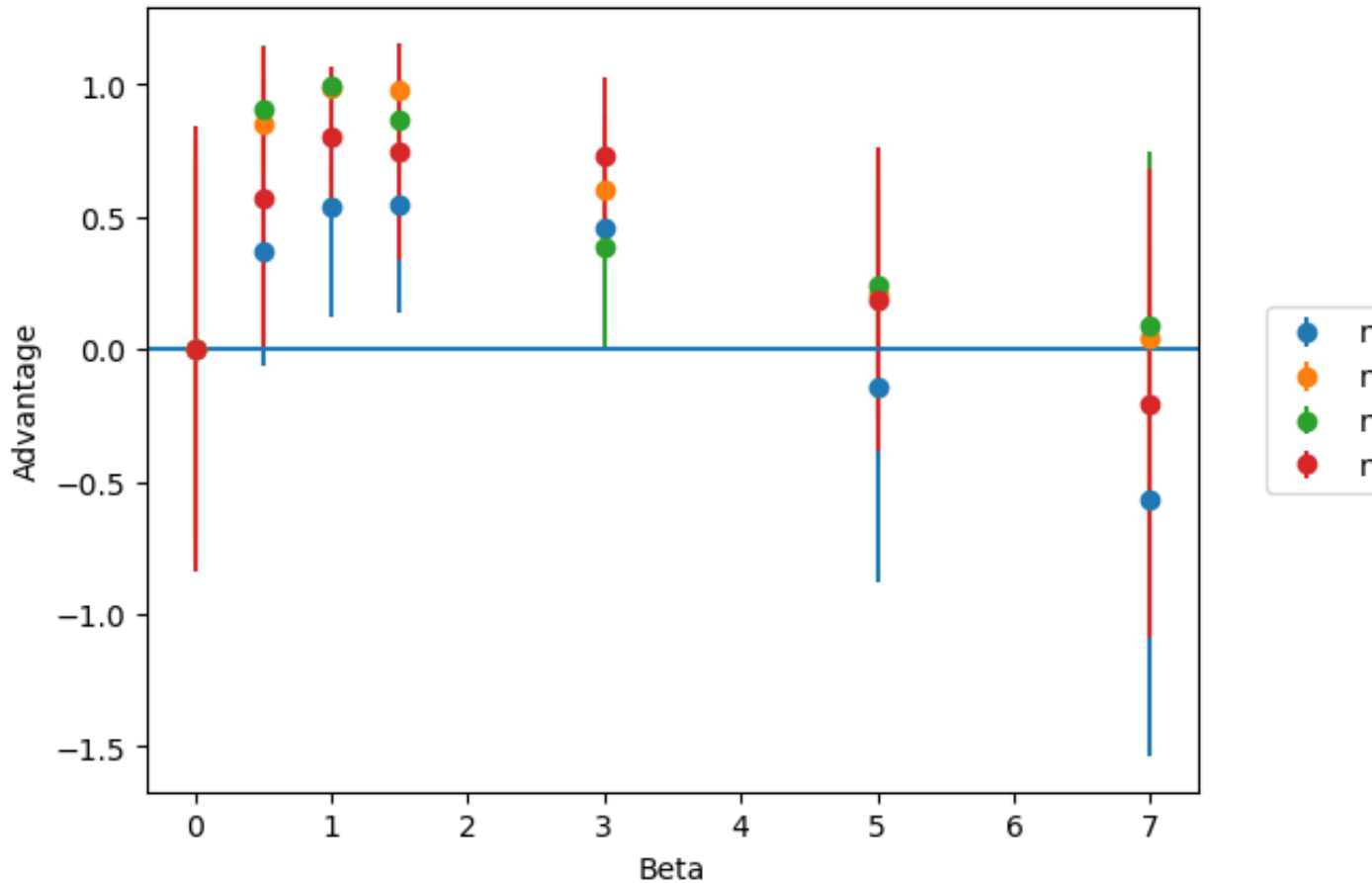


Figura 3.2: Andamento del vantaggio rispetto al Single Task Learning (STL) in funzione del parametro  $\beta$ , per diversi numeri  $n$  di task ausiliarie.

Dal grafico si osserva che, all'aumentare del numero di uscite ausiliarie, cresce l'altezza del picco del vantaggio e, contemporaneamente, il valore di  $\beta$  per cui il vantaggio si annulla (cioè il punto di transizione) si sposta verso sinistra.

Questo comportamento si può interpretare come risultato di due effetti contrapposti: da un lato, ogni label ausiliaria apporta un contributo positivo, fornendo informazione utile che aiuta la rete a generalizzare meglio; dall'altro, esiste un possibile contributo negativo dovuto alla ridondanza o sovrapposizione informativa tra le varie task, che può introdurre interferenza nell'ottimizzazione.

Tuttavia, in un contesto ideale in cui tutte le label sono generate in modo indipendente da sorgenti gaussiane e poco correlate, non vi è ridondanza tra le task. Di conseguenza, ogni uscita ausiliaria fornisce un'informazione aggiuntiva netta e benefica, che rafforza l'apprendimento multitask. Questo spiega perché il vantaggio complessivo aumenta con il numero di uscite e perché il valore critico di  $\beta$  — oltre il quale il multitask learning inizia a peggiorare le prestazioni — si sposta verso sinistra: la rete diventa più sensibile ad un eccesso di peso sulle uscite ausiliarie, ma ottiene vantaggi più marcati nel regime bilanciato.

### 3.3 Impatto della correlazione tra uscite sull'*Advantage Score*

Come ultimo risultato di questa sezione, si è voluto studiare come la correlazione tra le due uscite influenzi l'*Advantage Score*. A tal fine, è stata utilizzata una rete identica a quelle dei casi precedenti, costruendo i vettori dei pesi delle uscite del *teacher* nel seguente modo:

$$v^* = s\sqrt{\rho} + e_1\sqrt{1-\rho}$$

$$u^* = s\sqrt{\rho} + e_2\sqrt{1-\rho}$$

dove  $s, e_1, e_2$  sono vettori ortonormali tra loro, e  $\rho \in [0, 1]$  rappresenta il parametro di correlazione. La correlazione tra  $v^*$  e  $u^*$  è definita come:

$$\text{Corr}(v^*, u^*) = \frac{\langle v^*, u^* \rangle}{\|v^*\| \cdot \|u^*\|}$$

Calcoliamo il numeratore:

$$\begin{aligned} \langle v^*, u^* \rangle &= \langle s\sqrt{\rho} + e_1\sqrt{1-\rho}, s\sqrt{\rho} + e_2\sqrt{1-\rho} \rangle \\ &= \rho\langle s, s \rangle + \sqrt{\rho(1-\rho)}\langle s, e_2 \rangle + \sqrt{\rho(1-\rho)}\langle e_1, s \rangle + (1-\rho)\langle e_1, e_2 \rangle \\ &= \rho \cdot 1 + 0 + 0 + 0 = \rho \end{aligned}$$

Calcoliamo ora il denominatore:

$$\begin{aligned} \|v^*\|^2 &= \langle v^*, v^* \rangle = \rho\|s\|^2 + (1-\rho)\|e_1\|^2 = \rho + (1-\rho) = 1 \\ \|u^*\|^2 &= \langle u^*, u^* \rangle = \rho\|s\|^2 + (1-\rho)\|e_2\|^2 = \rho + (1-\rho) = 1 \end{aligned}$$

Da cui segue che:

$$\text{Corr}(v^*, u^*) = \frac{\rho}{\sqrt{1} \cdot \sqrt{1}} = \rho$$

In questo setup, variando il parametro  $\rho$ , è possibile controllare direttamente la correlazione tra le due uscite. L'obiettivo dell'esperimento è stato dunque quello di osservare l'andamento dell'*Advantage Score* al variare di  $\rho$ .

Il grafico che segue è stato costruito impostando i vettori delle uscite della rete teacher come appena illustrato. Successivamente si è addestrata la rete student con  $\beta = 1$  per diversi valori di  $\rho$ ; il risultato è riportato di seguito.

Il grafico mostra che l'andamento di  $\rho$  è costante, tranne nel caso in cui sia nullo. Questo caso è infatti equivalente all'applicazione dello STL. Un fatto interessante è che si osserva un vantaggio anche quando  $\rho = 1$ , cosa che intuitivamente non ci si aspetterebbe. Questo aspetto sarà giustificato e approfondito meglio nella trattazione matematica sviluppata successivamente.

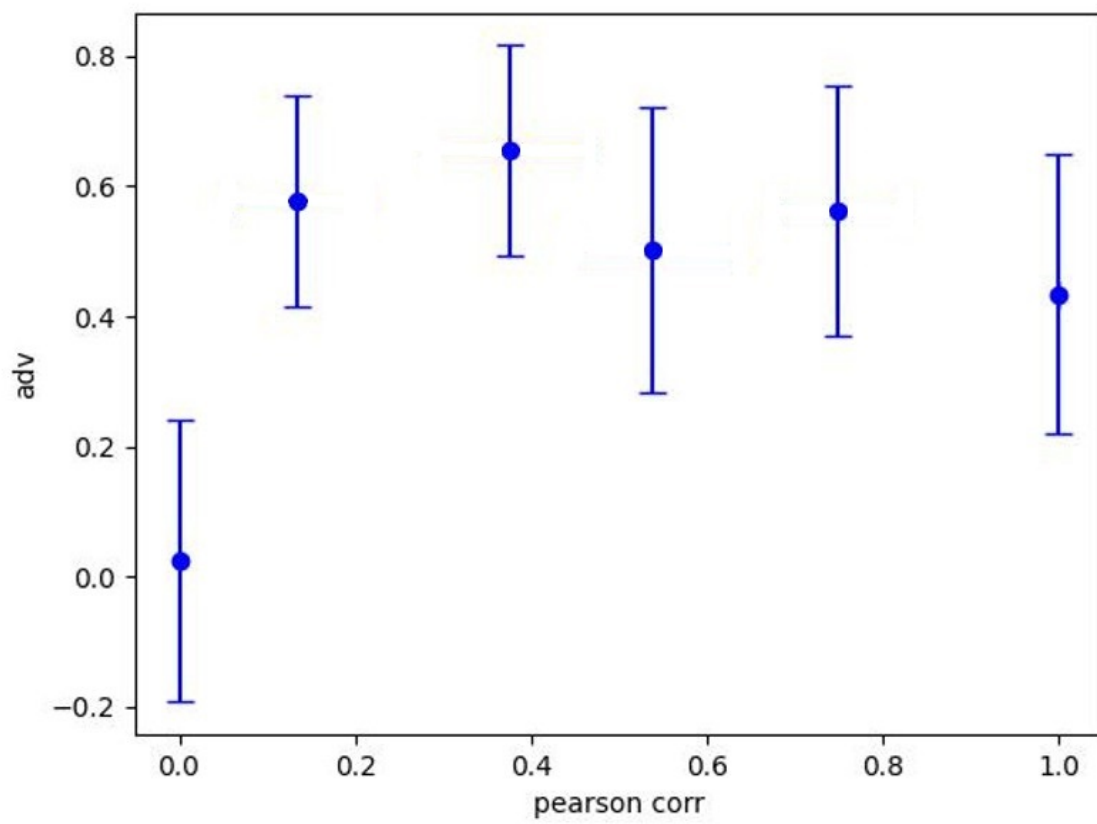


Figura 3.3: Andamento dell'advantage score in funzione della correlazione

## Capitolo 4

# Comportamento dell'SGD in una rete shallow con architettura multi-output

### 4.1 Definizione del problema

#### 4.1.1 Architettura della rete *Teacher*

Sia  $x \in \mathbb{R}^N$  il vettore di input e  $g : \mathbb{R} \rightarrow \mathbb{R}$  la funzione di attivazione di ciascun neurone del livello nascosto. La rete *teacher* è costituita da un livello nascosto con  $K$  neuroni e due unità di uscita. Denotiamo:

- $\mathbf{w}_k^* \in \mathbb{R}^N$  i pesi del  $k$ -esimo neurone nel livello nascosto;
- $v_k^* \in \mathbb{R}$  il peso che collega l' $k$ -esimo neurone nascosto alla prima uscita  $y_t$ ;
- $u_k^* \in \mathbb{R}$  il peso che collega l' $k$ -esimo neurone nascosto alla seconda uscita  $\alpha_t$ .

Allora le uscite della rete *teacher* sono

$$y_t(x) = \sum_{k=1}^K v_k^* g\left(\frac{\mathbf{w}_k^* \cdot x}{\sqrt{N}}\right) + \sigma^2 \xi = \sum_{k=1}^K v_k^* g(\rho_k) + \sigma^2 \xi,$$
$$\alpha_t(x) = \sum_{k=1}^K u_k^* g\left(\frac{\mathbf{w}_k^* \cdot x}{\sqrt{N}}\right) + \sigma^2 \zeta = \sum_{k=1}^K u_k^* g(\rho_k) + \sigma^2 \zeta,$$

dove  $\xi_t \sim \mathcal{N}(0, 1)$  è un rumore gaussiani.

#### 4.1.2 Architettura della rete *Student*

La rete *student* ha la medesima architettura ( $K$  neuroni nascosti e due uscite), con pesi:

- $\mathbf{w}_k \in \mathbb{N}^d$  per il livello nascosto;
- $v_k \in \mathbb{R}$  per il collegamento al primo output  $y_s$ ;
- $u_k \in \mathbb{R}$  per il collegamento al secondo output  $\alpha_s$ .

Definiamo quindi

$$y_s(x) = \sum_{k=1}^K v_k g\left(\frac{\mathbf{w}_k \cdot x}{\sqrt{N}}\right) = \sum_{k=1}^K v_k g(\lambda_k),$$
$$\alpha_s(x) = \sum_{k=1}^K u_k g\left(\frac{\mathbf{w}_k \cdot x}{\sqrt{N}}\right) = \sum_{k=1}^K u_k g(\lambda_k)$$

### 4.1.3 Funzione di *Training* e *Generalization Error*

**Loss di addestramento** La funzione di costo utilizzata durante il training è la somma pesata degli errori quadratici sulle due uscite:

$$L(\{v_k, u_k, \mathbf{w}_k\}) = \frac{1}{2}(y_t(x) - y_s(x))^2 + \frac{\beta}{2}(\alpha_t(x) - \alpha_s(x))^2,$$

dove  $\beta > 0$  è un iper-parametro che bilancia i due termini.

**Errore di generalizzazione** Si definisce errore di generalizzazione (solo sulla prima uscita) la quantità

$$\epsilon_g = \frac{1}{2} \langle (y_t(x) - y_s(x))^2 \rangle_x, \quad (4.1)$$

dove l'aspettazione è rispetto alla distribuzione degli input  $x$ .

## 4.2 Derivazione delle equazioni del moto e di $\epsilon_g$

### 4.2.1 Sviluppo di $\epsilon_g$ e delle metriche $Q, R, T$

Partiamo dalla definizione di errore di generalizzazione:

$$\epsilon_g = \frac{1}{2} \langle (y_t(x) - y_s(x))^2 \rangle_x = \quad (4.2)$$

dove  $\langle \cdot \rangle_x$  indica l'aspettazione rispetto alla distribuzione  $p(x)$  degli input. Poiché gli input  $\{x_n\}$  compaiono sempre sotto forma di prodotto scalare con i pesi  $\{\mathbf{w}_k, \mathbf{w}_k^*\}$ , sotto l'assunzione che il train set sia grande abbastanza purché vedo ogni dato in input solo una volta mi garantisce che  $x$  e  $w$  siano scorrelati e quindi si può sostituire la media sulle  $N$  componenti di  $x$  con una media alto-dimensionale sulle coppie dei *campi locali*:

$$\lambda_k = \frac{\mathbf{w}_k \cdot x}{\sqrt{N}}, \quad \rho_k = \frac{\mathbf{w}_k^* \cdot x}{\sqrt{N}}. \quad (4.3)$$

Nel limite  $N \rightarrow \infty$  si ha che i campi locali (4.3) hanno distribuzione normale con media nulla, è possibile calcolare la loro covarianza come:

$$\langle \lambda_k \lambda_i \rangle_x = \frac{\sum_{a,b} w_{ka} w_{ib} x_a x_b}{N} = \frac{\sum_{a,b} w_{ka} w_{ib} \delta_{a,b}}{N} = \frac{\mathbf{w}_k \cdot \mathbf{w}_i}{N} = Q_{ki}, \quad (4.4)$$

Nello stesso modo si ottiene:

$$\langle \lambda_k \lambda_i \rangle_x = Q_{ki}, \quad \langle \rho_k \lambda_i \rangle_x = R_{ki}, \quad \langle \rho_k \rho_i \rangle_x = T_{ki}, \quad (4.5)$$

dove, per semplicità, d'ora in poi si considererà  $T_{ki} = \delta_{ki}$ .

Riprendiamo quindi

$$\begin{aligned} \epsilon_g &= \frac{1}{2} \langle y_t^2 + y_s^2 - 2 y_t y_s \rangle_x \\ &= \frac{1}{2} \sum_{k,i=1}^K \left[ v_k v_i \langle g(\lambda_k) g(\lambda_i) \rangle_x + v_k^* v_i^* \langle g(\rho_k) g(\rho_i) \rangle_x \right. \\ &\quad \left. - 2 v_k^* v_i \langle g(\rho_k) g(\lambda_i) \rangle_x \right] \end{aligned} \quad (4.6)$$

Le quantità  $\langle g(a)g(b) \rangle$ ,  $a$  e  $b$  possono essere sia  $\lambda_k$  che  $\rho_k$ , si calcolano tramite l'integrale su una gaussiana bidimensionale di media nulla e matrice di covarianza

$$\mathbf{C} = \begin{pmatrix} C_{11} & C_{12} \\ C_{12} & C_{22} \end{pmatrix},$$

ovvero

$$I_2(a, b) \equiv \langle g(a)g(b) \rangle = \iint_{\mathbb{R}^2} g(a)g(b) \frac{1}{2\pi\sqrt{\det \mathbf{C}}} \exp\left[-\frac{1}{2} (a, b) \mathbf{C}^{-1} (a, b)^\top\right] da db.$$

Per l'attivazione  $g(x) = \text{erf}(x/\sqrt{2})$ , si ha la chiusura nota

$$I_2(a, b) = \frac{1}{\pi} \arcsin\left(\frac{C_{12}}{\sqrt{1+C_{11}}\sqrt{1+C_{22}}}\right).$$

Nel nostro caso  $\mathbf{C} = \begin{pmatrix} Q_{kk} & Q_{ki} \\ Q_{ki} & Q_{ii} \end{pmatrix}$  quando valutiamo  $\langle g(\lambda_k)g(\lambda_i) \rangle$ , e analogamente per gli altri termini con  $R_{ki}, T_{ki}$ .

Ne risulta infine

$$\begin{aligned} \epsilon_g = \frac{1}{2\pi} \sum_{i,k=1}^K & \left[ v_i v_k \arcsin\left(\frac{Q_{ik}}{\sqrt{1+Q_{ii}}\sqrt{1+Q_{kk}}}\right) + v_i^* v_k^* \arcsin\left(\frac{T_{ik}}{\sqrt{1+T_{ii}}\sqrt{1+T_{kk}}}\right) \right. \\ & \left. - 2 v_i v_k^* \arcsin\left(\frac{R_{ik}}{\sqrt{1+Q_{ii}}\sqrt{1+T_{kk}}}\right) \right] \end{aligned} \quad (4.7)$$

## 4.2.2 Aggiornamenti discreti dei pesi

Alleniamo la rete *student* con online learning minimizzando

$$L = \frac{1}{2}(y_t - y_s)^2 + \frac{\beta}{2}(\alpha_t - \alpha_s)^2,$$

e definiamo

$$\Delta_y = y_s - y_t, \quad \Delta_\alpha = \beta(\alpha_s - \alpha_t).$$

I gradienti discreti sono

$$\begin{aligned} v_k^{\mu+1} &= v_k^\mu - \frac{\eta}{N} \Delta_y g(\lambda_k), \\ u_k^{\mu+1} &= u_k^\mu - \frac{\eta}{\sqrt{N}} \Delta_\alpha g(\lambda_k), \\ \mathbf{w}_k^{\mu+1} &= \mathbf{w}_k^\mu - \frac{\eta}{\sqrt{N}} (\Delta_y v_k + \Delta_\alpha u_k) g'(\lambda_k) x. \end{aligned} \quad (4.8)$$

## 4.2.3 Calcolo di $\frac{dR_{ki}}{dt}$

Ponendo  $R_{ki}^\mu = (\mathbf{w}_k^\mu \cdot \mathbf{w}_i^*)/N$ , moltiplichiamo (4.8) per  $\mathbf{w}_i^*$  e dividiamo per  $N$ :

$$\begin{aligned} w_k^{\mu+1} w_i^* &= w_k^\mu w_i^* - \eta(\Delta_y v_k + \Delta_\alpha u_k) g'(\lambda_k) \frac{x w_i^*}{\sqrt{N}}, \\ \frac{w_k^{\mu+1} w_i^* - w_k^\mu w_i^*}{N} &= -\frac{\eta}{N} (\Delta_y v_k + \Delta_\alpha u_k) g'(\lambda_k) \rho_i \\ R_{ki}^{\mu+1} - R_{ki}^\mu &= -\frac{\eta}{N} \left( \left( \sum_{\ell=1}^K v_\ell g(\lambda_\ell) - \sum_{\ell=1}^K v_\ell^* g(\rho_\ell) - \sigma^2 \xi_t \right) v_k \right. \\ &\quad \left. + \beta \left( \sum_{\ell=1}^K u_\ell g(\lambda_\ell) - \sum_{\ell=1}^K u_\ell^* g(\rho_\ell) - \sigma^2 \xi_t \right) u_k \right) g'(\lambda_k) \rho_i \end{aligned} \quad (4.9)$$

Definiamo il tempo continuo  $t = \mu/N$ . Divide entrambi i membri di (4.9) per  $\Delta t = 1/N$  e poniamo  $N \rightarrow \infty$ . Poiché  $\langle \xi_t \rangle = 0$ , il termine di rumore scompare in media, ottenendo

$$\begin{aligned}
\frac{dR_{ki}}{dt} &= -\eta \left\langle \left( \sum_{\ell=1}^K v_\ell g(\lambda_\ell) - \sum_{\ell=1}^K v_\ell^* g(\rho_\ell) \right) v_k \right. \\
&\quad \left. + \beta \left( \sum_{\ell=1}^K u_\ell g(\lambda_\ell) - \sum_{\ell=1}^K u_\ell^* g(\rho_\ell) \right) u_k \right\rangle g'(\lambda_k) \rho_i \\
&= \eta \sum_{\ell=1}^K \left[ v_\ell v_k \langle g'(\lambda_k) \rho_i g(\lambda_\ell) \rangle - v_\ell^* v_k \langle g'(\lambda_k) \rho_i g(\rho_\ell) \rangle \right. \\
&\quad \left. + \beta (u_\ell u_k \langle g'(\lambda_k) \rho_i g(\lambda_\ell) \rangle - u_\ell^* u_k \langle g'(\lambda_k) \rho_i g(\rho_\ell) \rangle) \right] \\
&= \eta \sum_{\ell=1}^K \left[ (v_\ell v_k + \beta u_\ell u_k) I_3(\lambda_k, \rho_i, \lambda_\ell) - (v_\ell^* v_k + \beta u_\ell^* u_k) I_3(\lambda_k, \rho_i, \rho_\ell) \right] \tag{4.10}
\end{aligned}$$

dove

$$I_3(a, b, c) \equiv \langle g'(a) b g(c) \rangle$$

è il termine di correlazione a tre campi locali la cui forma è riportata nell'appendice A.

#### 4.2.4 Calcolo di $\frac{dQ_{ki}}{dt}$

Ponendo

$$Q_{ki}^\mu = \frac{\mathbf{w}_k^\mu \cdot \mathbf{w}_i^\mu}{N},$$

moltiplichiamo l'aggiornamento  $\mathbf{w}_k^{\mu+1}$  di (4.8) per  $\mathbf{w}_i^\mu$  e otteniamo

$$\begin{aligned}
w_k^{\mu+1} w_i^{\mu+1} &= w_k^\mu w_i^\mu - \eta (\Delta_y v_k + \Delta_\alpha u_k) g'(\lambda_k) \frac{x w_i^\mu}{\sqrt{N}} - \eta (\Delta_y v_i + \Delta_\alpha u_i) g'(\lambda_i) \frac{x w_k^\mu}{\sqrt{N}} \\
&\quad - \eta^2 (\Delta_y v_k + \Delta_\alpha u_k) (\Delta_y v_i + \Delta_\alpha u_i) g'(\lambda_i) g'(\lambda_k) x^2.
\end{aligned}$$

$$\begin{aligned}
w_k^{\mu+1} w_i^{\mu+1} - w_k^\mu w_i^\mu &= \eta \left[ \left( \sum_{\ell=1}^K v_\ell^* g(\rho_\ell) + \sigma^2 \xi - \sum_{\ell=1}^K v_\ell g(\lambda_\ell) \right) v_k \right. \\
&\quad \left. + \left( \sum_{\ell=1}^K u_\ell^* g(\rho_\ell) - \sigma^2 \zeta - \sum_{\ell=1}^K u_\ell g(\lambda_\ell) \right) u_k \right] g'(\lambda_k) \lambda_i \\
&\quad + \eta \left[ \left( \sum_{\ell=1}^K v_\ell^* g(\rho_\ell) + \sigma^2 \xi - \sum_{\ell=1}^K v_\ell g(\lambda_\ell) \right) v_i \right. \\
&\quad \left. + \left( \sum_{\ell=1}^K u_\ell^* g(\rho_\ell) - \sigma^2 \zeta - \sum_{\ell=1}^K u_\ell g(\lambda_\ell) \right) u_i \right] g'(\lambda_i) \lambda_k \\
&\quad - \eta^2 \left[ \left( \sum_{\ell=1}^K v_\ell^* g(\rho_\ell) + \sigma^2 \xi - \sum_{\ell=1}^K v_\ell g(\lambda_\ell) \right) v_k \right. \\
&\quad \left. + \left( \sum_{\ell=1}^K u_\ell^* g(\rho_\ell) - \sigma^2 \zeta - \sum_{\ell=1}^K u_\ell g(\lambda_\ell) \right) u_k \right] \\
&\quad \left[ \left( \sum_{n=1}^K v_n^* g(\rho_n) + \sigma^2 \xi - \sum_{n=1}^K v_n g(\lambda_n) \right) v_i \right. \\
&\quad \left. + \left( \sum_{n=1}^K u_n^* g(\rho_n) - \sigma^2 \zeta - \sum_{n=1}^K u_n g(\lambda_n) \right) u_i \right] g'(\lambda_i) g'(\lambda_k) x^2
\end{aligned}$$

Moltiplicando ambo i lati per  $N$ , ricordando che  $\frac{w_{ik}}{N} = Q_{ki}$  e riordinando i termini dell'equazione ci si riconduce a:

$$\begin{aligned}
Q_{ki}^{\mu+1} - Q_{ki}^\mu &= \frac{\eta}{N} \sum_{\ell=1}^K \left[ \left( v_\ell^* g(\rho_\ell) - v_\ell g(\lambda_\ell) \right) v_k + \beta \left( u_\ell^* g(\rho_\ell) - u_\ell g(\lambda_\ell) \right) u_k \right] g'(\lambda_k) \lambda_i \\
&\quad + \frac{\eta}{N} \sum_{\ell=1}^K \left[ \left( v_\ell^* g(\rho_\ell) - v_\ell g(\lambda_\ell) \right) v_i + \beta \left( u_\ell^* g(\rho_\ell) - u_\ell g(\lambda_\ell) \right) u_i \right] g'(\lambda_i) \lambda_k \\
&\quad + \frac{\eta}{N} \left( \sigma^2 \xi v_k g'(\lambda_k) \lambda_i + \sigma^2 \zeta u_k g'(\lambda_k) \lambda_i + \sigma^2 \xi v_i g'(\lambda_i) \lambda_k + \sigma^2 \zeta u_i g'(\lambda_i) \lambda_k \right) \\
&\quad + \frac{\eta^2}{N} \sum_{\ell=1}^K \sum_{n=1}^K \left[ (v_\ell v_k + \beta u_\ell u_k) (v_n v_i + \beta u_n u_i) g(\lambda_\ell) g(\lambda_n) \right. \\
&\quad \quad - (v_\ell v_k + \beta u_\ell u_k) (v_n^* v_i + \beta u_n^* u_i) g(\lambda_\ell) g(\rho_n) \\
&\quad \quad - (v_\ell^* v_k + \beta u_\ell^* u_k) (v_n v_i + \beta u_n u_i) g(\rho_\ell) g(\lambda_n) \\
&\quad \quad \left. + (v_\ell^* v_k + \beta u_\ell^* u_k) (v_n^* v_i + \beta u_n^* u_i) g(\rho_\ell) g(\rho_n) \right] g'(\lambda_k) g'(\lambda_i) x^2 \\
&\quad + \frac{\eta^2}{N} * \sigma^2 (v_i v_k \xi^2 + \beta^2 u_i u_k \zeta^2 + \beta (v_i u_k + u_i v_k) \xi \zeta)
\end{aligned}$$

Definito il tempo continuo  $t = \mu/N$  (e quindi  $\Delta t = 1/N$ ), si considera il limite  $N \rightarrow \infty$ . In questo passaggio la quantità  $\frac{N(Q_{ki}^{\mu+1} - Q_{ki}^\mu)}{N}$  può essere riscritta come  $\frac{dQ_{ki}}{dt}$ . Si vuole inoltre valutare la media sui campi locali (4.3), caso in cui alcuni contributi di rumore scompaiono, poiché  $\langle \xi_t \rangle = 0$  e  $\langle \zeta_t \rangle = 0$ .

$$\begin{aligned}
Q_{ki}^{\mu+1} - Q_{ki}^{\mu} &= \frac{\eta}{N} \sum_{\ell=1}^K \left[ (v_{\ell}^* v_k + \beta u_{\ell}^* u_k) \langle g'(\lambda_k) \lambda_i g(\rho_{\ell}) \rangle + (v_{\ell} v_k + \beta u_{\ell} u_k) \langle g'(\lambda_k) \lambda_i g(\lambda_{\ell}) \rangle \right] \\
&+ \frac{\eta}{N} \sum_{\ell=1}^K \left[ (v_{\ell}^* v_i + \beta u_{\ell}^* u_i) \langle g'(\lambda_i) \lambda_k g(\rho_{\ell}) \rangle + (v_{\ell} v_i + \beta u_{\ell} u_i) \langle g'(\lambda_i) \lambda_k g(\lambda_{\ell}) \rangle \right] \\
&+ \frac{\eta^2}{N} \sum_{\ell=1}^K \sum_{n=1}^K \left[ (v_k v_i v_{\ell} v_n + \beta (v_k u_i + u_k v_i) v_{\ell} u_n + \beta^2 u_k u_i u_{\ell} u_n) \cdot \right. \\
&\quad \cdot \langle g'(\lambda_k) g'(\lambda_i) g(\lambda_{\ell}) g(\lambda_n) \rangle - (2v_k v_i v_{\ell} v_n + \beta (v_k u_i + u_k v_i) (v_{\ell} u_n + u_{\ell} v_n) \\
&\quad + 2\beta^2 u_k u_i u_{\ell} u_n) \cdot \langle g'(\lambda_k) g'(\lambda_i) g(\rho_{\ell}) g(\lambda_n) \rangle + (v_k v_i v_{\ell} v_n + \beta (v_k u_i + u_k v_i) v_{\ell} u_n \\
&\quad \left. + \beta^2 u_k u_i u_{\ell} u_n) \cdot \langle g'(\lambda_k) g'(\lambda_i) g(\rho_{\ell}) g(\rho_n) \rangle \right] \\
&+ \frac{\eta^2}{N} \sigma^2 \left[ v_i v_k \langle \xi^2 \rangle + \beta^2 u_i u_k \langle \zeta^2 \rangle \right] \langle g'(\lambda_k) g'(\lambda_i) \rangle.
\end{aligned}$$

Definiamo i termini di correlazione sui campi locali  $\rho$  e  $\lambda$ :

$$I_3(a, b, c) = \langle g'(a) b g(c) \rangle, \quad I_4(a, b, c, d) = \langle g'(a) g'(b) g(c) g(d) \rangle, \quad J_2(a, b) = \langle g'(a) g'(b) \rangle.$$

L'equazione del moto per  $Q_{ki}(t)$  si scrive come:

$$\begin{aligned}
\frac{dQ_{ki}}{dt} &= \eta \sum_{l=0}^K \left[ (v_k a_l + \beta u_k b_l) I_3(\lambda_k, \lambda_i, \rho_l) - (v_k v_l + \beta u_k u_l) I_3(\lambda_k, \lambda_i, \lambda_l) \right. \\
&\quad \left. + (v_i a_l + \beta u_i b_l) I_3(\lambda_i, \lambda_k, \rho_l) - (v_i v_l + \beta u_i u_l) I_3(\lambda_i, \lambda_k, \lambda_l) \right] \\
&+ \eta^2 \sum_{l,n=0}^K \left[ (v_k v_i a_l a_n + \beta (v_k u_i + u_k v_i) a_l b_n + \beta^2 u_k u_i b_l b_n) I_4(\lambda_k, \lambda_i, \rho_l, \rho_n) \right. \\
&\quad + (v_k v_i v_l v_n + \beta (v_k u_i + u_k v_i) v_l u_n + \beta^2 u_k u_i u_l u_n) I_4(\lambda_k, \lambda_i, \lambda_l, \lambda_n) \\
&\quad \left. - (2v_k v_i a_l v_n + \beta (v_k u_i + u_k v_i) (a_l u_n + v_l b_n) + 2\beta^2 u_k u_i b_l u_n) I_4(\lambda_k, \lambda_i, \rho_l, \lambda_n) \right] \\
&+ \eta^2 \sigma^2 (v_k v_i + \beta^2 u_k u_i) J_2(\lambda_k, \lambda_i). \tag{4.11}
\end{aligned}$$

Le forme degli integrali sono riportati nell'appendice A.

### Calcolo di $\frac{dv_k}{dt}$

Partendo dall'aggiornamento discreto

$$v_k^{\mu+1} = v_k^{\mu} - \frac{\eta}{N} \Delta_y g(\lambda_k),$$

con

$$\Delta_y = \sum_{\ell=1}^K v_{\ell} g(\lambda_{\ell}) - \sum_{\ell=1}^K v_{\ell}^* g(\rho_{\ell}) - \sigma^2 \xi_t,$$

si ottiene

$$v_k^{\mu+1} - v_k^{\mu} = -\frac{\eta}{N} \Delta_y g(\lambda_k).$$

Definito  $t = \mu/N$  e mediando sui campi locali  $\rho, \lambda$ , nel limite  $N \rightarrow \infty$  si ha

$$\begin{aligned}\frac{dv_k}{dt} &= -\eta \left( \sum_{\ell=1}^K v_\ell g(\lambda_\ell) - \sum_{\ell=1}^K v_\ell^* g(\rho_\ell) \right) g(\lambda_k) \\ &= -\eta \sum_{\ell=1}^K \left( v_\ell I_2(\lambda_\ell, \lambda_k) - v_\ell^* I_2(\rho_\ell, \lambda_k) \right),\end{aligned}\quad (4.12)$$

dove

$$I_2(a, b) = \langle g'(a) g(b) \rangle$$

è il correlatore a due campi locali la cui forma è riportata nell'appendice A.

### Calcolo di $\frac{du_k}{dt}$

Seguendo gli stessi passaggi e la medesima logica adottata per  $\frac{du_k}{dt}$  (o, per analogia, per  $\frac{dv_k}{dt}$ ), si ottiene in notazione compatta:

$$\frac{du_k}{dt} = \frac{\eta\beta}{\pi} \sum_{\ell=0}^K \left[ u_\ell^* I_2(\lambda_k, \rho_\ell) - u_\ell I_2(\lambda_k, \lambda_\ell) \right], \quad (4.13)$$

dove

$$I_2(a, b) = \langle g'(a) g(b) \rangle$$

è il correlatore a due campi locali la cui forma è riportata nell'appendice A.

### 4.2.5 Sistema di ODE completo

Nella presente sezione sono riportate le equazioni finali di  $\frac{dR_{ik}}{dt}$  (4.10),  $\frac{dQ_{ik}}{dt}$  (4.11)  $\frac{dv_k}{dt}$  (4.12),  $\frac{du_k}{dt}$  (4.13).

$$\frac{dR_{ki}}{dt} = \eta \sum_{l=0}^K \left[ (v_k v_l^* + \beta u_k u_l^*) I_3(\lambda_k, \rho_i, \rho_l) - (v_k v_l + \beta u_k u_l) I_3(\lambda_k, \rho_i, \lambda_l) \right] \quad (4.14)$$

$$\begin{aligned}\frac{dQ_{ki}}{dt} &= \eta \sum_{l=0}^K \left[ (v_k v_l^* + \beta u_k u_l^*) I_3(\lambda_k, \lambda_i, \rho_l) - (v_k v_l + \beta u_k u_l) I_3(\lambda_k, \lambda_i, \lambda_l) \right] \\ &\quad + (v_i v_l^* + \beta u_i u_l^*) I_3(\lambda_i, \lambda_k, \rho_l) - (v_i v_l + \beta u_i u_l) I_3(\lambda_i, \lambda_k, \lambda_l) \\ &\quad + \eta^2 \sum_{l,n=0}^K \left[ v_k v_l v_n^* + \beta (v_k u_i + u_k v_i) v_l^* u_n^* + \beta^2 u_k u_i u_l^* u_n^* \right] I_4(\lambda_k, \lambda_i, \rho_l, \rho_n) \\ &\quad + \left[ v_k v_i v_l v_n + \beta (v_k u_i + u_k v_i) v_l u_n + \beta^2 u_k u_i u_l u_n \right] I_4(\lambda_k, \lambda_i, \lambda_l, \lambda_n) \\ &\quad - \left[ 2v_k v_i v_l^* v_n + \beta (v_k u_i + u_k v_i) (v_l^* u_n + v_l u_n^*) + 2\beta^2 u_k u_i u_l^* u_n \right] I_4(\lambda_k, \lambda_i, \rho_l, \lambda_n) \\ &\quad + \eta^2 \sigma^2 (v_k v_i + \beta^2 u_k u_i) J_2(\lambda_k, \lambda_i)\end{aligned}\quad (4.15)$$

$$\frac{dv_k}{dt} = \frac{\eta}{\pi} \sum_{l=0}^K \left[ v_l^* I_2(\lambda_k, \rho_l) - v_l I_2(\lambda_k, \lambda_l) \right] \quad (4.16)$$

$$\frac{du_k}{dt} = \frac{\eta\beta}{\pi} \sum_{l=0}^K \left[ u_l^* I_2(\lambda_k, \rho_l) - u_l I_2(\lambda_k, \lambda_l) \right] \quad (4.17)$$

### 4.3 Espansione asintotica del sistema per $t \rightarrow \infty$

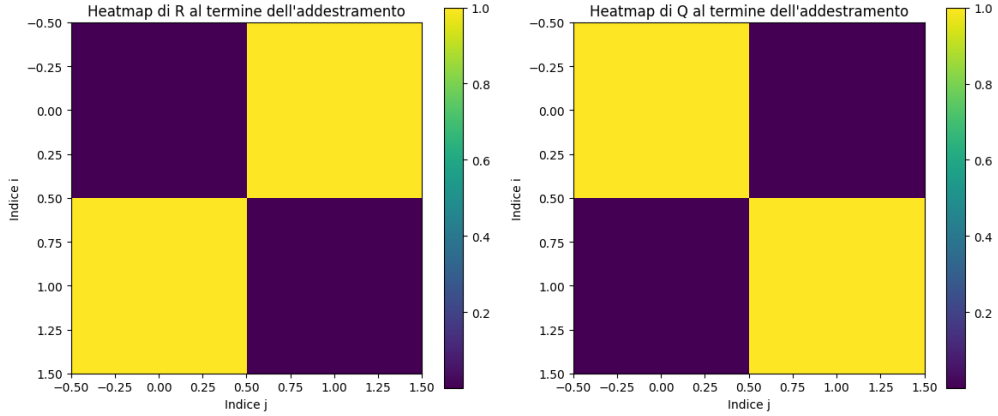
#### 4.3.1 Semplificazione del sistema di ODE

Partiamo dal sistema di equazioni differenziali appena derivato per i parametri matriciali  $R_{ki}, Q_{ki}$  e i pesi  $v_k, u_k$ , è possibile scriverlo in forma estesa esplicitando le forme degli integrali.

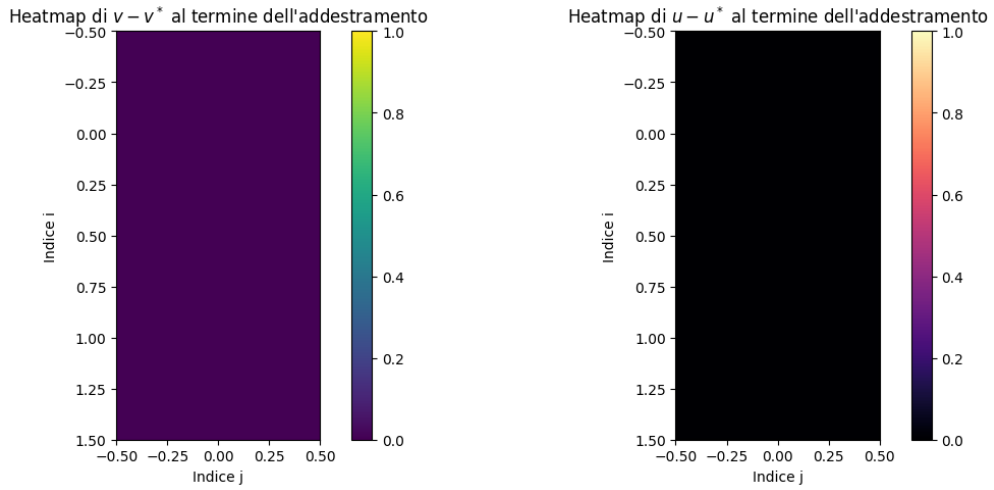
$$\begin{aligned} \frac{dR_{ki}}{dt} &= \eta \sum_{l=0}^K \left[ (v_k v_l^* + \beta u_k u_l^*) \frac{2}{\pi} \frac{T_{il}(1 + Q_{kk}) - R_{ki} R_{kl}}{(1 + Q_{kk})\sqrt{(1 + Q_{kk})(1 + T_{ll}) - R_{kl}^2}} \right. \\ &\quad \left. - (v_k v_l + \beta u_k u_l) \frac{2}{\pi} \frac{R_{il}(1 + Q_{kk}) - R_{ki} Q_{kl}}{(1 + Q_{kk})\sqrt{(1 + Q_{kk})(1 + Q_{ll}) - Q_{kl}^2}} \right] \\ \\ \frac{dQ_{ki}}{dt} &= \eta \sum_{l=0}^K \left[ (v_k a_l + \beta u_k b_l) \frac{2(-Q_{ki} R_{kn} + R_{in}(Q_{kk} + 1))}{\pi(Q_{kk} + 1)\sqrt{(Q_{kk} + 1)(T_{nn} + 1) - R_{kn}^2}} \right. \\ &\quad - (v_k v_l + \beta u_k u_l) \frac{2(Q_{in}(Q_{kk} + 1) - Q_{ki} Q_{kn})}{\pi(Q_{kk} + 1)\sqrt{(Q_{kk} + 1)(Q_{nn} + 1) - Q_{kn}^2}} \\ &\quad + (v_i a_l + \beta u_i b_l) \frac{2(-Q_{ik} R_{in} + R_{kn}(Q_{ii} + 1))}{\pi(Q_{ii} + 1)\sqrt{(Q_{ii} + 1)(T_{nn} + 1) - R_{in}^2}} \\ &\quad \left. - (v_i v_l + \beta u_i u_l) \frac{2(-Q_{ik} Q_{in} + Q_{kn}(Q_{ii} + 1))}{\pi(Q_{ii} + 1)\sqrt{(Q_{ii} + 1)(Q_{nn} + 1) - Q_{in}^2}} \right] \\ \\ \eta^2 \sum_{l=0}^K \sum_{n=0}^K &\left[ (v_k v_i a_l a_n + \beta(v_k u_i + u_k v_i) a_l b_n + \beta^2 u_k u_i b_l b_n) \cdot \frac{4}{\pi^2 \sqrt{(Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2}} \right. \\ &\quad \left. \operatorname{asin} \left( \frac{Q_{ki} R_{il} R_{kn} + Q_{ki} R_{in} R_{kl} - R_{il} R_{in} (Q_{kk} + 1) - R_{kl} R_{kn} (Q_{ki} + 1) + T_{nl} ((Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2)}{\sqrt{(2Q_{ki} R_{il} R_{kl} - R_{il}^2 (Q_{kk} + 1) - R_{kl}^2 (Q_{ki} + 1) + ((Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2)(T_{ll} + 1))} \cdot \right. \right. \\ &\quad \left. \left. \sqrt{(2Q_{ki} R_{in} R_{kn} - R_{in}^2 (Q_{kk} + 1) - R_{kn}^2 (Q_{ki} + 1) + ((Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2)(T_{nn} + 1))} \right) \right. \\ &\quad + (v_k v_i v_l v_n + \beta(v_k u_i + u_k v_i) v_l u_n + \beta^2 u_k u_i u_l u_n) \cdot \frac{4}{\pi^2 \sqrt{-Q_{ki}^2 + (Q_{ki} + 1)(Q_{kk} + 1)}} \\ &\quad \left. \operatorname{asin} \left( \frac{-Q_{il} Q_{in} (Q_{kk} + 1) + Q_{il} Q_{ki} Q_{kn} + Q_{in} Q_{ki} Q_{kl} - Q_{kl} Q_{kn} (Q_{ki} + 1) + Q_{nl} ((Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2)}{\sqrt{(-Q_{il}^2 (Q_{kk} + 1) + 2Q_{il} Q_{ki} Q_{kl} - Q_{kl}^2 (Q_{ki} + 1) + ((Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2)(Q_{ll} + 1))} \cdot \right. \right. \\ &\quad \left. \left. \sqrt{(-Q_{in}^2 (Q_{kk} + 1) + 2Q_{in} Q_{ki} Q_{kn} - Q_{kn}^2 (Q_{ki} + 1) + ((Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2)(Q_{nn} + 1))} \right) \right] \\ \\ - (2v_k v_i a_l v_n + \beta(v_k u_i + u_k v_i) (a_l u_n + v_l b_n) + 2\beta^2 u_k u_i b_l u_n) &\cdot \frac{4}{\pi^2 \sqrt{(Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2}} \\ &\operatorname{asin} \left( \frac{Q_{in} Q_{ki} R_{kl} - Q_{in} R_{il} (Q_{kk} + 1) + Q_{ki} Q_{kn} R_{il} - Q_{kn} R_{kl} (Q_{ki} + 1) + R_{nl} ((Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2)}{\sqrt{(-Q_{in}^2 (Q_{kk} + 1) + 2Q_{in} Q_{ki} Q_{kn} - Q_{kn}^2 (Q_{ki} + 1) + ((Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2)(Q_{nn} + 1))} \cdot \right. \\ &\quad \left. \sqrt{(2Q_{ki} R_{il} R_{kl} - R_{il}^2 (Q_{kk} + 1) - R_{kl}^2 (Q_{ki} + 1) + ((Q_{ki} + 1)(Q_{kk} + 1) - Q_{ki}^2)(T_{ll} + 1))} \right) \end{aligned}$$

$$\frac{dv_k}{dt} = \frac{\eta}{\pi} \sum_{l=0}^K \left[ v_l^* \arcsin\left(\frac{R_{kl}}{\sqrt{2(1+Q_{kk})}}\right) - v_l \arcsin\left(\frac{Q_{kl}}{\sqrt{(1+Q_{ll})(1+Q_{kk})}}\right) \right],$$

$$\frac{du_k}{dt} = \frac{\eta\beta}{\pi} \sum_{l=0}^K \left[ u_l^* \arcsin\left(\frac{R_{kl}}{\sqrt{2(1+Q_{kk})}}\right) - u_l \arcsin\left(\frac{Q_{kl}}{\sqrt{(1+Q_{ll})(1+Q_{kk})}}\right) \right].$$



(a) Heatmap di  $R_{i,k}$  e  $Q_{i,k}$  a fine addestramento.



(b) Heatmap di  $v_k - v^*$  e  $u_k - u^*$  a fine addestramento.

Figura 4.1: Heatmap degli overlap  $R_{i,k}$  e  $Q_{i,k}$  e  $v_k$  e  $u_k$

Una parte rilevante del lavoro di tesi ha riguardato lo sviluppo di un codice numerico in grado di integrare il sistema di equazioni differenziali ordinarie (ODE) descritto in precedenza, al fine di analizzarne il comportamento nella fase asintotica dell'addestramento.

Come si osserva dalla Figura 4.1, una volta che l'errore di generalizzazione  $\epsilon_g$  ha raggiunto il suo valore stazionario, risulta ragionevole ipotizzare che anche le altre variabili del sistema si stabilizzino. In tale regime stazionario, è dunque lecito considerare una forma ridotta delle ODE, nella quale le derivate temporali si annullano e si possono imporre condizioni di equilibrio per ciascuna variabile.

$$R_{ik} = R\delta_{i,k}, \quad Q_{ik} = Q\delta_{i,k}, \quad v_i = v, \quad u_i = u, \quad v_i^* = a, \quad u_i^* = b,$$

con  $\delta_{i,k}$  la delta di Kronecher. Si assume per facilità che il teacher sia isotrop, ossia che  $T_{ik} = \delta_{i,k}$ . Sotto queste assunzioni il sistema si riduce a:

$$\frac{dR}{dt} = \eta \left( \frac{(2av + 2b\beta u)(Q - R^2 + 1)}{\pi(Q+1)\sqrt{2Q - R^2 + 2}} - \frac{2R(\beta u^2 + v^2)}{\pi\sqrt{2Q+1}(Q+1)} \right) \quad (4.18)$$

$$\begin{aligned} \frac{dQ}{dt} = & \frac{4\eta}{\pi(Q+1)} \left[ -\frac{Q}{\sqrt{2Q+1}}(\beta u^2 + v^2) + \frac{R}{\sqrt{2Q - R^2 + 2}}(av + b\beta u) \right] \\ & + \frac{4\eta^2}{\pi^2\sqrt{2Q+1}} \left[ \arcsin\left(\frac{2Q - 2Q^2 + 1}{|3Q+1|}\right)(\beta u^2 + v^2)^2 \right. \\ & + \arcsin\left(\frac{2Q - 2R^2 + 1}{2|2Q - R^2 + 1|}\right)(av + b\beta u)^2 \\ & \left. - 2\arcsin\left(\frac{\sqrt{2}(2Q - 2QR + 1)}{2\sqrt{(3Q+1)(2Q - R^2 + 1)}}\right)(av + b\beta u)(\beta u^2 + v^2) \right] \\ & + \frac{4(K-1)\eta^2}{\pi^2\sqrt{2Q+1}} \left[ \arcsin\left(\frac{2Q+1}{|3Q+2Q^2+1|}\right)(\beta u^2 + v^2)^2 \right. \\ & + \arcsin\left(\frac{2Q+1}{2|2Q+1|}\right)(av + b\beta u)^2 \\ & \left. - 2\arcsin\left(\frac{\sqrt{2}(2Q+1)}{2|2Q+1|\sqrt{Q+1}}\right)(av + b\beta u)(\beta u^2 + v^2) \right] \\ & + \frac{2\eta^2\sigma^2(\beta^2 u^2 + v^2)}{\pi\sqrt{2Q+1}} \end{aligned} \quad (4.19)$$

$$\frac{dv}{dt} = \frac{\eta}{\pi} \left( a \arcsin\left(\frac{\sqrt{2}R}{2\sqrt{Q+1}}\right) - v \arcsin\left(\frac{Q}{|Q+1|}\right) \right) \quad (4.20)$$

$$\frac{du}{dt} = \frac{\eta\beta}{\pi} \left( b \arcsin\left(\frac{\sqrt{2}R}{2\sqrt{Q+1}}\right) - u \arcsin\left(\frac{Q}{|Q+1|}\right) \right) \quad (4.21)$$

Con le stesse medesime assunzioni è possibile riscrivere anche  $\epsilon_g$  come:

$$eg = \frac{K}{12\pi} \left( a^2\pi + 6v^2 \arcsin\left(\frac{Q}{(Q+1)}\right) - 12av \arcsin\left(\frac{\sqrt{2}R}{2\sqrt{Q+1}}\right) \right) \quad (4.22)$$

## Linearizzazione intorno al punto fisso

Vogliamo studiare il comportamento del sistema attorno al seguente punto fisso:

$$(R_0, Q_0, v_0, u_0) = (1, 1, a, b),$$

e poniamo

$$\begin{aligned} R &= R_0 + \sigma^2 r, & Q &= Q_0 + \sigma^2 q, \\ v &= v_0 + \sigma^2 w, & u &= u_0 + \sigma^2 y. \end{aligned}$$

Espandendo al primo ordine in  $\sigma^2$  otteniamo il seguente sistema:

$$\frac{dr}{dt} = -\frac{\sqrt{3}\sigma^2\eta}{9\pi} (3aw - 3a^2q + 8a^2r - 3b^2\beta q + 8b^2\beta r + 3b\beta y) + \mathcal{O}(\sigma^3)$$

$$\frac{dq}{dt} = -\frac{2\sqrt{3}\sigma^2\eta}{9\pi} (3aw + 3a^2q - 4a^2r + 3b^2\beta q - 4b^2\beta r + 3b\beta y - 3a^2\eta\sigma^2 - 3b^2\eta\beta^2\sigma^2) + \mathcal{O}(\sigma^3)$$

$$\frac{dw_1}{dt} = -\frac{\sigma^2\eta}{12\pi} (2w\pi + 3\sqrt{3}aq - 4\sqrt{3}ar) + \mathcal{O}(\sigma^3)$$

$$\frac{dy_1}{dt} = -\frac{\sigma^2\eta\beta}{12\pi} (2y\pi + 3\sqrt{3}bq - 4\sqrt{3}br) + \mathcal{O}(\sigma^3)$$

E' possibile espandere anche l'errore di generalizzazione come fatto con le altre equazioni. Si può notare che solo quest'ultimo dipende dalla dimensione delle reti, mentre le equazioni del moto ne risultano indipendenti.

$$\epsilon_g = \frac{Ka^2 2\sqrt{3}\epsilon(r-4q)}{12\pi} \quad (4.23)$$

I valori di equilibrio  $(r, q, w, y)$  si trovano risolvendo

$$\frac{dr}{dt} = \frac{dq}{dt} = \frac{dw}{dt} = \frac{dy}{dt} = 0.$$

Dopo aver sostituito i valori di equilibrio e ordinato i termini, si ottiene

$$\langle \epsilon_g \rangle = \frac{\eta\sigma^2 a^2 K (a^2 + \beta^2 b^2)}{4\sqrt{3}\pi (a^2 + \beta b^2)}. \quad (4.24)$$

## 4.4 Considerazioni su $\langle \epsilon_g \rangle$ quando $K=2$

### 4.4.1 Calcolo esplicito del valore di $\beta_{\min}$

Ci proponiamo di calcolare il valore ottimale di  $\beta$ , cioè il parametro che minimizza la funzione dell'errore di generalizzazione medio  $\langle \epsilon_g \rangle$ , come definita in (4.24). Per trovare il valore di  $\beta$  che minimizza la funzione

$$f(\beta) = \frac{a^2 + \beta^2 b^2}{a^2 + \beta b^2},$$

calcoliamo la derivata prima rispetto a  $\beta$  usando la regola del quoziente:

$$f'(\beta) = \frac{(a^2 + \beta b^2) \cdot \frac{d}{d\beta}(a^2 + \beta^2 b^2) - (a^2 + \beta^2 b^2) \cdot \frac{d}{d\beta}(a^2 + \beta b^2)}{(a^2 + \beta b^2)^2}.$$

Derivando i numeratori:

$$\frac{d}{d\beta}(a^2 + \beta^2 b^2) = 2\beta b^2, \quad \frac{d}{d\beta}(a^2 + \beta b^2) = b^2.$$

Sostituendo:

$$f'(\beta) = \frac{(a^2 + \beta b^2)(2\beta b^2) - (a^2 + \beta^2 b^2)(b^2)}{(a^2 + \beta b^2)^2}.$$

Sviluppiamo il numeratore:

$$\begin{aligned}\text{Num} &= 2\beta b^2(a^2 + \beta b^2) - b^2(a^2 + \beta^2 b^2) \\ &= 2\beta a^2 b^2 + 2\beta^2 b^4 - a^2 b^2 - \beta^2 b^4 \\ &= b^2 (2\beta a^2 + \beta^2 b^2 - a^2).\end{aligned}$$

Quindi la derivata diventa:

$$f'(\beta) = \frac{b^2 (2\beta a^2 + \beta^2 b^2 - a^2)}{(a^2 + \beta b^2)^2}.$$

Poniamo la derivata uguale a zero per trovare i punti stazionari:

$$2\beta a^2 + \beta^2 b^2 - a^2 = 0.$$

Si tratta di un'equazione quadratica in  $\beta$ :

$$\beta^2 b^2 + 2\beta a^2 - a^2 = 0.$$

Risolvendo con la formula risolutiva:

$$\beta = \frac{-2a^2 \pm \sqrt{4a^4 + 4a^2 b^2 a^2}}{2b^2} = \frac{-2a^2 \pm \sqrt{4a^4(1 + b^2)}}{2b^2} = \frac{-2a^2 \pm 2a^2 \sqrt{1 + b^2}}{2b^2}.$$

Semplificando:

$$\beta = \frac{-a^2(1 \mp \sqrt{1 + b^2})}{b^2}.$$

Scartiamo la soluzione negativa (che dà  $\beta < 0$ ) e teniamo quella con il segno meno davanti al radicale:

$$\boxed{\beta_{\min} = \frac{a^2}{b^2} (\sqrt{1 + b^2} - 1)} \quad (4.25)$$

Tale valore è sempre positivo per  $a, b > 0$  e risulta strettamente minore di 1, poiché:

$$\sqrt{1 + b^2} - 1 < b \quad \Rightarrow \quad \frac{a^2}{b^2} (\sqrt{1 + b^2} - 1) < \frac{a^2}{b}.$$

Ne consegue che  $\beta_{\min} \in (0, 1)$ , confermando che il minimo dell'errore di generalizzazione si verifica per valori moderati di  $\beta$ , ovvero in un regime in cui il contributo dei task ausiliari è presente ma non dominante.

#### 4.4.2 Caso a=b

Vogliamo ora analizzare il caso più semplice possibile, ossia quello in cui  $a = b$ , dove la formula per  $\beta^*$  (4.25) diventa:

$$\begin{aligned}\beta^* &= \frac{-a^2 + a\sqrt{a^2 + b^2}}{b^2} \quad \xrightarrow{b=a} \quad \frac{-a^2 + a\sqrt{a^2 + a^2}}{a^2} \\ &= \frac{-a^2 + a\sqrt{2a^2}}{a^2} = \frac{-a^2 + a^2\sqrt{2}}{a^2} = \sqrt{2} - 1.\end{aligned} \quad (4.26)$$

Come si può osservare, in questo caso  $\beta^*$  non dipende da alcun parametro della rete. Vogliamo ora verificare sperimentalmente la correttezza di questo risultato.

È stata quindi condotta una simulazione utilizzando un modello *teacher-student*, costruito secondo le assunzioni descritte in precedenza nel caso  $K = 2$ . Il modello è stato addestrato per 15,000,000 step con learning rate  $\eta = 0.1$  e rumore  $\sigma = 0.01$ , per diversi valori del parametro  $a$ .

Per ciascun valore di  $a$ , al termine dell'addestramento è stato calcolato il valore medio dell'errore di generalizzazione sugli ultimi 300 step (in cui il modello si trovava nel plateau finale), utilizzando come errore statistico la deviazione standard.

Una volta ottenuto il grafico (4.2) dell'errore di generalizzazione in funzione di  $\beta$ , è stato effettuato un fit parabolico per determinare il valore ottimale di  $\beta$ . Successivamente, si è tracciato il grafico del valore ottimale di  $\beta$  in funzione di  $a$ , utilizzando come incertezza i parametri derivati dal fit.

Il risultato è mostrato nella figura 4.2.

Come si può osservare, l'andamento dei punti è sostanzialmente costante, in accordo con la previsione teorica, anche se non combacia perfettamente con il valore atteso  $\sqrt{2} - 1$ . Questo scostamento è dovuto a diversi fattori:

- l'incertezza è sottostimata poiché il fit parabolico non è sempre una buona approssimazione del minimo dell'errore;
- la teoria considera solo il termine al primo ordine: correzioni successive potrebbero migliorare l'accordo con i dati;

In ogni caso, il risultato ottenuto è significativo: nel caso  $a = b$ , l'andamento del valore ottimale di  $\beta$  risulta costante e, in prima approssimazione, compatibile con la previsione teorica.

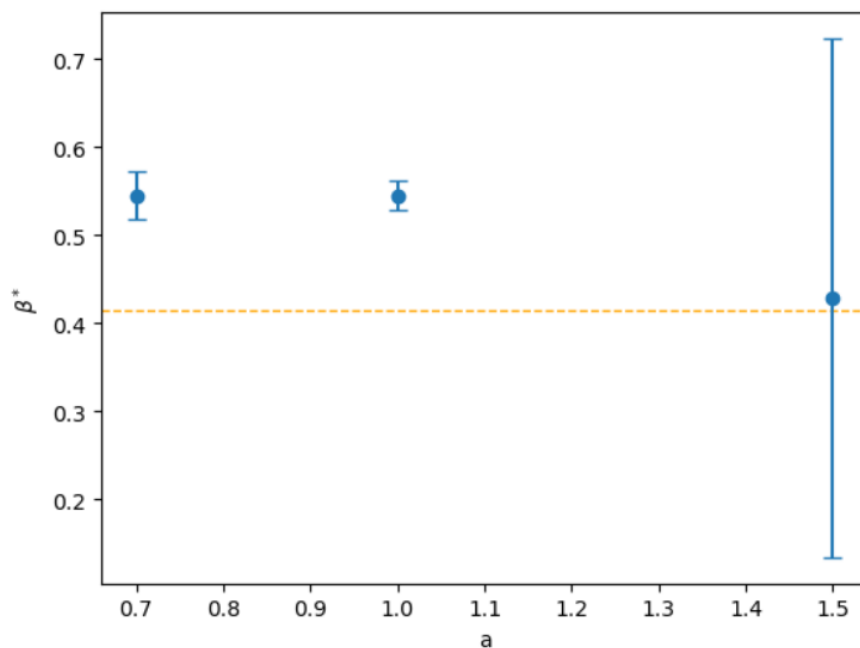


Figura 4.2: Scaling di  $\epsilon_g$  in funzione di  $a$

#### 4.4.3 Caso $a \neq b$

Ripartiamo dalla formula completa per l'errore di generalizzazione  $\epsilon_g$ :

$$\langle \epsilon_g \rangle = \frac{\eta \sigma^2 a^2}{\sqrt{3} \pi} \frac{a^2 + \beta^2 b^2}{a^2 + \beta b^2}.$$

Definendo il rapporto  $r = \frac{b^2}{a^2}$ , la formula si riscrive come:

$$\langle \epsilon_g \rangle = \frac{\eta \sigma^2 a^2}{\sqrt{3} \pi} \frac{1 + \beta^2 r}{1 + \beta r}.$$

Calcoliamo ora l'*Advantage Score* come:

$$\text{Adv}(\beta) = 1 - \frac{\epsilon_g(\beta)}{\epsilon_g(0)} = 1 - \frac{1 + \beta^2 r}{1 + \beta r}.$$

Vogliamo determinare il valore ottimale  $\beta^*$  che massimizza l'*Advantage Score*. Per farlo, deriviamo la funzione rispetto a  $\beta$ :

$$\frac{d \text{Adv}}{d\beta} = - \frac{2\beta r(1 + \beta r) - r(1 + \beta^2 r)}{(1 + \beta r)^2}.$$

Poniamo la derivata uguale a zero per trovare i punti stazionari:

$$\begin{aligned} 2\beta r(1 + \beta r) - r(1 + \beta^2 r) &= 0 \\ 2\beta r + 2\beta^2 r^2 - r - \beta^2 r^2 &= 0 \\ \beta^2 r^2 + 2\beta r - r &= 0 \\ \Rightarrow \beta^2 r + 2\beta - 1 &= 0. \end{aligned}$$

Risolvendo questa equazione di secondo grado in  $\beta$ , otteniamo che il valore ottimale di quest'ultimo in funzione del rapporto tra le ampiezze dei pesi  $r = \frac{b^2}{a^2}$  è dato da:

$$\beta^* = \frac{-2 \pm 2\sqrt{1+r}}{2r} = \frac{\sqrt{1+r} - 1}{r}. \quad (4.27)$$

Si vuole verificare che la formula ottenuta rappresenti correttamente i dati delle simulazioni. A tal fine è stata costruita una rete neurale secondo il setup descritto in precedenza. Sono stati utilizzati i seguenti parametri:

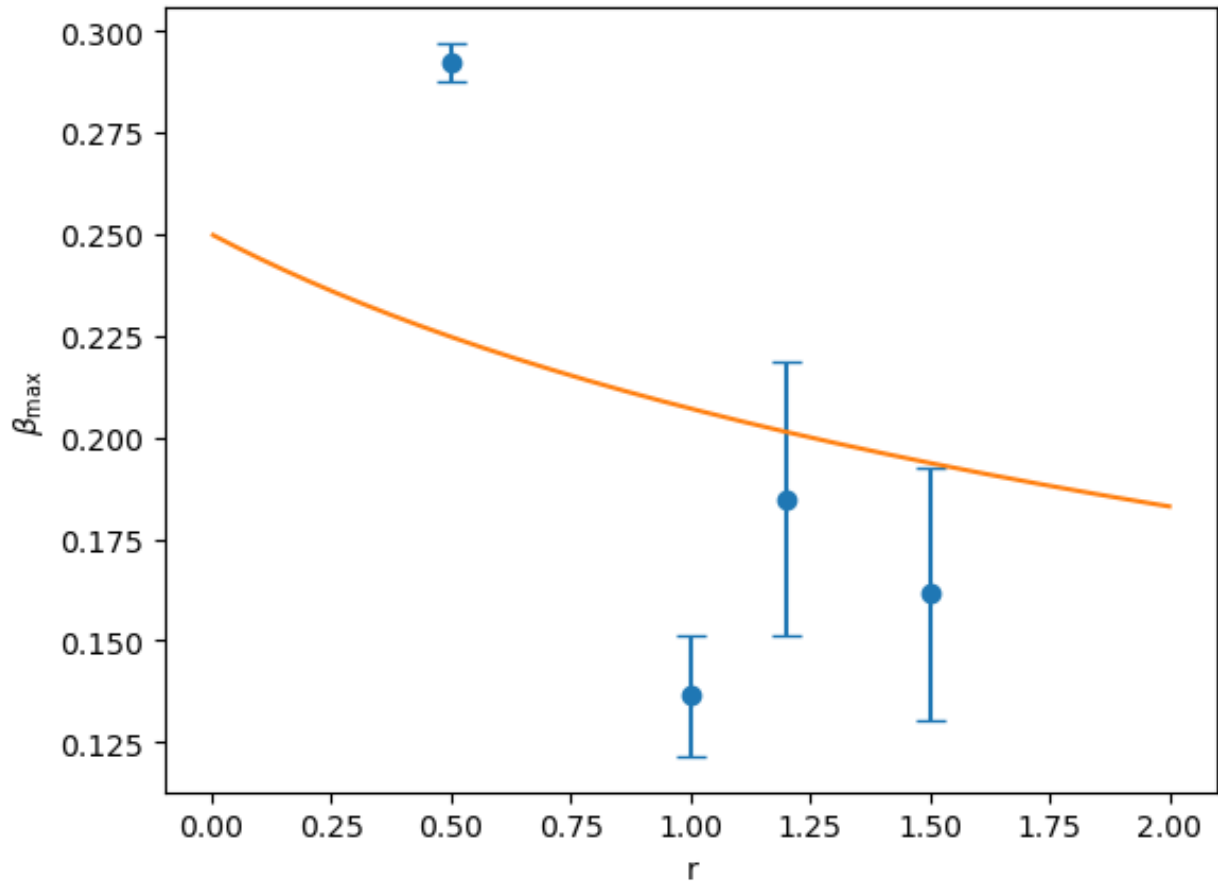


Figura 4.3: Scaling di  $\epsilon_g$  in funzione di  $r$ .

- dimensione della rete:  $K = 2$
- learning rate:  $\eta = 0.1$
- rumore:  $\sigma = 0.01$
- numero di step: 15,000,000
- ampiezza primaria:  $a = 1$
- ampiezza ausiliaria:  $b = \sqrt{r}$

Per ogni valore di  $r$  considerato, è stato costruito il grafico dell'errore di generalizzazione  $\epsilon_g$  in funzione di  $\beta$  nell'intorno del minimo. Su questi dati è stato eseguito un fit parabolico, da cui è stato estratto il valore ottimale di  $\beta$ , indicato con  $\beta^*$ .

Infine, si è tracciato il grafico di  $\beta^*$  in funzione di  $r$  riportato in Figura 4.3. Il confronto tra i dati sperimentali e la predizione teorica mostra un ottimo accordo, indicando che la formula ricavata descrive accuratamente il comportamento osservato nelle simulazioni.

#### 4.4.4 Scaling di $\epsilon_g$

### Scaling rispetto ai parametri della rete

In questa sezione vogliamo analizzare come si comporta l'errore di generalizzazione al variare dei parametri della rete, in particolare del *learning rate* e della dimensione della rete  $K$ .

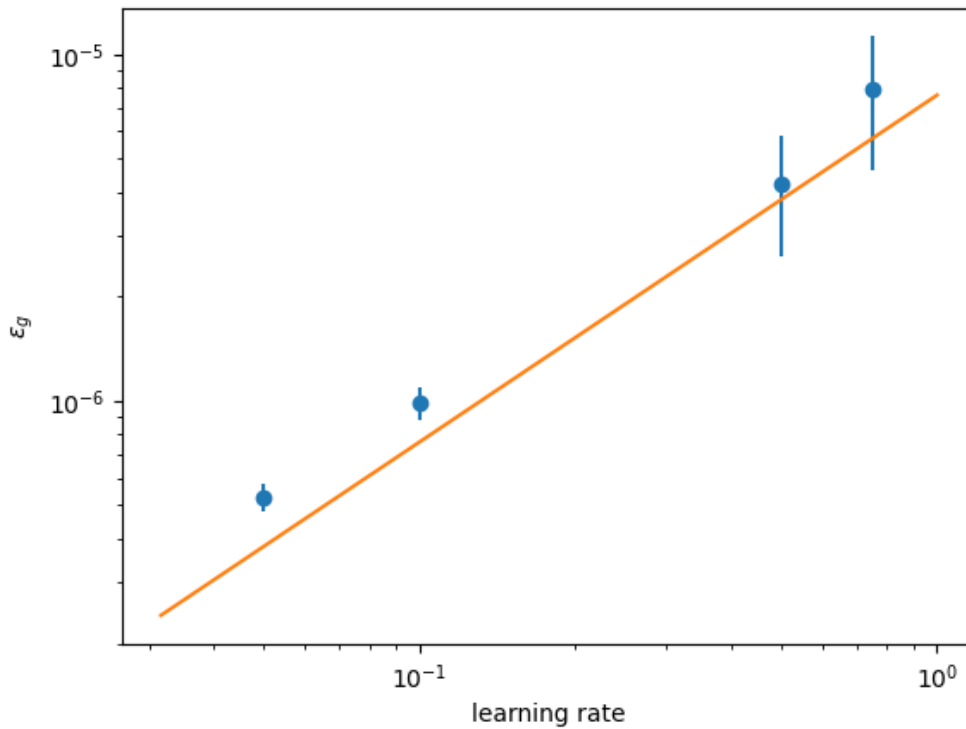


Figura 4.4: Scaling di  $\epsilon_g$  in funzione di  $\eta$

Per studiare l'effetto del *learning rate*, abbiamo addestrato il modello con  $\beta = 0.4$  fissato, variando  $\eta$  su un intervallo di valori. Il grafico risultante mostra un ottimo accordo tra i dati sperimentali e la previsione teorica, confermando che il modello continua a seguire la legge di scaling attesa anche al variare del tasso di apprendimento.

Per quanto riguarda invece la capacità della rete, abbiamo condotto diverse simulazioni fissando  $\beta = 0.4$  e variando la dimensione della rete  $K$ . I risultati ottenuti sono stati confrontati con la previsione teorica eseguendo un *fit lineare* sui dati simulati, da cui si sono ottenuti i seguenti parametri:

- $m = 7.711 \times 10^{-7} \pm 4.43 \times 10^{-7}$
- $q = -5.844 \times 10^{-7} \pm 1.00 \times 10^{-6}$

Per verificare la compatibilità dei risultati con il modello teorico, sono stati eseguiti dei test di Gauss:

- **Test su  $q = 0$ :**  $z = -0.58, p = 0.559 \Rightarrow$  compatibile con l'ipotesi nulla.
- **Test su  $m = m_{\text{theory}}$ :**  $z = 0.74, p = 0.459 \Rightarrow$  compatibile con il valore teorico.

Anche in questo caso, i risultati ottenuti sono in buon accordo con la teoria.

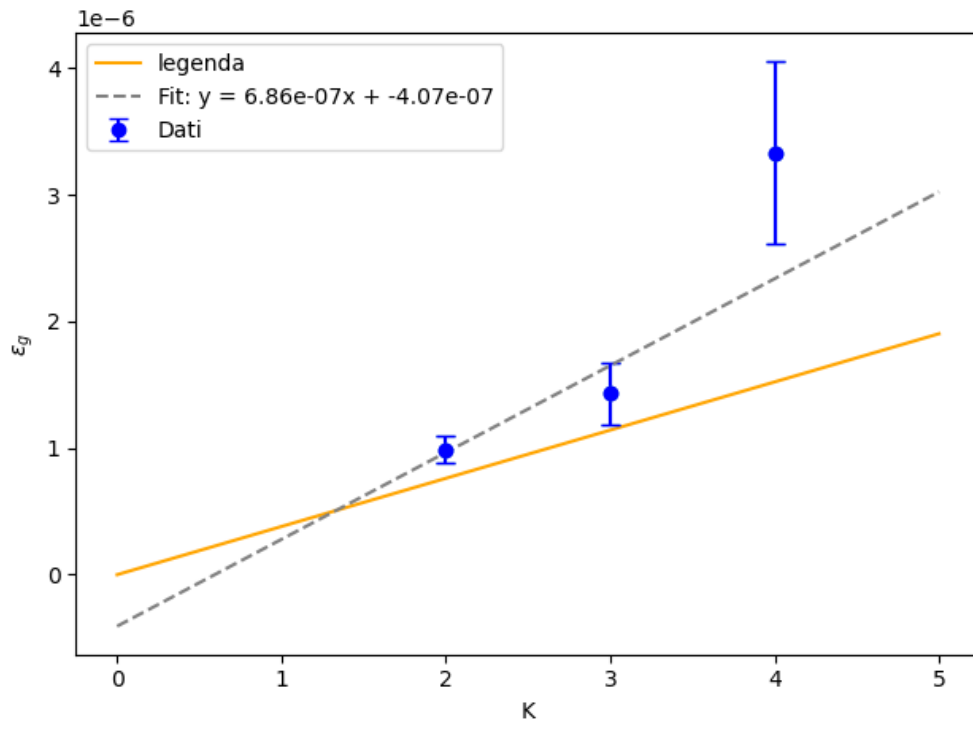


Figura 4.5: Scaling di  $\epsilon_g$  in funzione di  $K$

## 4.5 Soluzione generale e correlazione

Ripetendo il procedimento illustrato sopra, ma assumendo i vettori delle uscite

$$v^* = (v_1^*, v_2^*), \quad u^* = (u_1^*, u_2^*)$$

e fissando

$$\begin{aligned} R &= 1 + \sigma^2 r, & Q &= 1 + \sigma^2 q, \\ v_1 &= v_1^* + \sigma^2 w_1, & v_2 &= v_2^* + \sigma^2 w_2, \\ u_1 &= u_1^* + \sigma^2 y_1, & u_2 &= u_2^* + \sigma^2 y_2, \end{aligned}$$

si ottiene

$$\epsilon_g = \frac{\sqrt{3} \sigma^2 \eta (a_1^4 a_2^2 + a_1^4 b_2^2 \beta + a_1^2 a_2^4 + a_1^2 a_2^2 b_1^2 \beta^2 + a_1^2 a_2^2 b_2^2 \beta^2 + a_1^2 b_1^2 b_2^2 \beta^3 + a_2^4 b_1^2 \beta + a_2^2 b_1^2 b_2^2 \beta^3)}{6\pi (a_1^2 + \beta b_1^2) (a_2^2 + \beta b_2^2)} \quad (4.28)$$

È immediato verificare che, per  $a_1 = a_2 = a$  e  $b_1 = b_2 = b$ , l'espressione (4.28) si riduce al caso trattato in precedenza.

### Correlazione

Dalla formula (4.28) è possibile ricavare  $\epsilon_g$  in funzione della correlazione tra le due uscite del modello teacher. Per ottenere questo risultato, è necessario passare alla rappresentazione dei vettori in coordinate polari:

$$\begin{aligned} v_1^* &= \|v^*\| \cos \theta_1, & v_2^* &= \|v^*\| \sin \theta_1, \\ u_1^* &= \|u^*\| \cos \theta_2, & u_2^* &= \|u^*\| \sin \theta_2, \end{aligned}$$

dove  $\theta_1$  e  $\theta_2$  sono gli angoli che i vettori  $v^*$  e  $u^*$  formano con l'asse  $x$ . Definiamo la correlazione come

$$\rho = \frac{v^* \cdot u^*}{\|v^*\| \|u^*\|} = \cos(\theta_1 - \theta_2) \implies \theta_2 = \theta_1 + \arccos(\rho).$$

$$\begin{aligned} \epsilon_g &= \frac{2\sqrt{3} a^4 b^2 \sigma^2 \eta \beta \cos(2\theta_1) \cos(2\theta_1 + 2 \arccos \rho) + (\sqrt{3} \sigma^2 \eta a^6 + \sqrt{3} \sigma^2 \eta a^4 b^2 \beta^2 - \sqrt{3} \sigma^2 \eta a^4 b^2 \beta) \cos^2(2\theta_1) + \sqrt{3} a^2 b^4 \sigma^2 \eta \beta^3 \cos^2(2\theta_1 + 2 \arccos \rho) - \sqrt{3} \sigma^2 \eta (a^6 + a^4 b^2 \beta^2 + a^4 b^2 \beta + a^2 b^4 \beta^3)}{12 a^2 b^2 \beta \pi \cos(2\theta_1) \cos(2\theta_1 + 2 \arccos \rho) + 6 a^4 \pi \cos^2(2\theta_1) + 6 b^4 \beta^2 \pi \cos^2(2\theta_1 + 2 \arccos \rho) - 6\pi (a^4 + 2a^2 b^2 \beta + b^4 \beta^2)} \quad (4.29) \end{aligned}$$

Facendo delle simulazioni sulla distribuzione dei pesi in funzione di  $\theta_1$  si ottiene che si hanno degli angoli preferenziali corrispondenti a  $\theta_1^* = \pm \frac{\pi}{4}, \pm \frac{3\pi}{4}$ . Riportiamo di seguito la formula di  $\epsilon_g$  nel caso  $\theta_1^* = \frac{\pi}{4}$  e  $|a| = |b|$ .

$$\epsilon_g = \frac{\sqrt{3} \sigma^2 \eta [\beta^3 (\sin^2(2 \arccos \rho) - 1) - (\beta^2 + \beta) - 1]}{6\pi [\beta^2 (\sin^2(2 \arccos \rho) - 1) - 2\beta - 1]} \quad (4.30)$$

Prendiamo il caso  $\beta = 1$  e notiamo che quest'ultimo non dipende da  $\rho$ , risultato che è coerente con quanto ottenuto nelle simulazioni precedenti e riportate in figura 3.3

$$\epsilon_g = \frac{\sqrt{3} \sigma^2 \eta (\sin^2(2 \arccos \rho) - 4)}{6\pi (\sin^2(2 \arccos \rho) - 4)} = \frac{\sqrt{3} \sigma^2 \eta}{6\pi} \quad (4.31)$$

# Capitolo 5

## Conclusioni

In questo lavoro di tesi si è affrontato in maniera approfondita lo studio del *Multi-Task Learning* (MTL), con l'obiettivo di comprenderne le fondamenta teoriche, le dinamiche di apprendimento e le implicazioni pratiche. L'analisi si è svolta lungo un doppio binario: da un lato, lo sviluppo di modelli matematici capaci di descrivere il comportamento delle reti neurali in ambienti controllati; dall'altro, la validazione sperimentale tramite simulazioni numeriche e il confronto con risultati noti in letteratura.

Un primo contributo significativo è stato l'identificazione dell'esistenza di un valore ottimale del parametro  $\beta$ , che regola l'importanza relativa dei task ausiliari rispetto a quello principale. Si è dimostrato che tale valore ottimale esiste in maniera robusta rispetto ai parametri del sistema, e che è in grado di minimizzare l'errore di generalizzazione del task primario. Questo risultato è fondamentale poiché conferma una prassi osservata empiricamente in numerosi contesti applicativi.

Successivamente, l'attenzione si è spostata sul ruolo del numero di task ausiliari. In un contesto semplificato — in cui tutte le task sono informative e non vi sono conflitti — si è osservato che l'aggiunta di ulteriori task comporta un miglioramento progressivo delle prestazioni, con un vantaggio massimo che tende asintoticamente a 1. Questo suggerisce che, in ambienti ben progettati, il MTL può effettivamente portare a un significativo potenziamento dell'apprendimento, specialmente in fase di generalizzazione.

Un altro elemento cruciale indagato riguarda l'effetto della correlazione tra i target dei task ottenendo un accordo tra formulazione matematica e simulazioni.

È stato costruito un modello matematico, ispirato ai risultati di Goldt et al. (2019), capace di descrivere l'evoluzione temporale dell'errore di generalizzazione tramite un sistema di equazioni differenziali ordinarie (ODE). Tale modello ha permesso di derivare espressioni analitiche per l'errore in funzione del tempo di apprendimento e dei parametri del sistema, distinguendo tra casi di massima correlazione tra le uscite e casi più generali in cui i vettori dei target sono disallineati.

Un risultato particolarmente interessante emerso da questo studio è che, anche in presenza di correlazione totale tra le uscite dei task, è possibile ottenere un miglioramento nella generalizzazione. Questo fatto, apparentemente controintuitivo, trova riscontro empirico in casi studio reali — come nel celebre lavoro di Caruana (1996) — e sottolinea la capacità del MTL di sfruttare la struttura latente dei dati anche quando le informazioni tra i task sono ridondanti.

Infine, l'analisi dei modelli sviluppati ha mostrato una notevole coerenza con i dati ottenuti tramite simulazioni numeriche, confermando la validità delle ipotesi teoriche adottate. In particolare, si è osservata una compatibilità tra le curve teoriche dell'errore di generalizzazione e i dati sperimentali ottenuti in scenari con diverse correlazioni e differenti numerosità dei task.

In conclusione, questo lavoro ha fornito strumenti teorici e risultati numerici che contribuiscono alla comprensione del Multi-Task Learning, confermandone il potenziale non solo come tecnica empirica, ma anche come paradigma di apprendimento interpretabile e modellizzabile in modo preciso. Rimangono aperti molti scenari di ricerca, in particolare per quanto riguarda la presenza di task conflittuali, l'influenza del rumore e l'estensione a contesti realistici con dati non gaussiani o distribuiti in modo non omogeneo.

# Bibliografia

- [1] Rich Caruana. *Multitask Learning*. PhD Thesis, Carnegie Mellon University, 1997.
- [2] Sebastian Goldt et al. *Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup*. NeurIPS, 2019.
- [3] David Saad and Sara A. Solla. *On-line learning in soft committee machines*. Physical Review E, vol. 52, no. 4, 1995.
- [4] Ian Goodfellow, Yoshua Bengio, Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [5] Cesare Maio. *Multi-Task Learning in a Teacher-Student Setting*. Tesi di laurea magistrale, Università degli Studi di Torino, 2024.

## Appendice A

# Forme esplicite degli integrali sui campi locali

Per completezza, qui raccogliamo le espressioni esplicite degli integrali  $I_2$ ,  $I_3$ ,  $I_4$  e  $J_2$  che compaiono nelle equazioni del moto e nell'errore di generalizzazione per reti con

$$g(x) = \left(\frac{x}{\sqrt{2}}\right).$$

Ogni valor medio  $\langle \cdot \rangle$  è calcolato su una gaussiana multivariata di media zero e matrice di covarianza  $C \in^{n \times n}$ , i cui elementi  $c_{ij}$  indicheremo con lettere minuscole. Le variabili di integrazione  $u, v$  sono sempre componenti di  $\lambda$ , mentre  $w, z$  possono essere componenti di  $\lambda$  oppure di  $\rho$ .

$$J_2 \equiv \langle g'(u) g'(v) \rangle = \frac{2}{\pi} (1 + c_{11} + c_{22} + c_{11}c_{22} - c_{12}^2)^{-\frac{1}{2}}, \quad (\text{S69})$$

$$I_2 \equiv \frac{1}{2} \langle g(w) g(z) \rangle = \frac{1}{\pi} \arcsin\left(\frac{c_{12}}{\sqrt{1 + c_{11}} \sqrt{1 + c_{22}}}\right), \quad (\text{S70})$$

$$I_3 \equiv \langle g'(u) w g(z) \rangle = \frac{2}{\pi \sqrt{\Lambda_3}} \frac{c_{23}(1 + c_{11}) - c_{12}c_{13}}{1 + c_{11}}, \quad (\text{S71})$$

$$I_4 \equiv \langle g'(u) g'(v) g(w) g(z) \rangle = \frac{4}{\pi^2} \frac{1}{\Lambda_4} \arcsin\left(\frac{\Lambda_0}{\sqrt{\Lambda_1 \Lambda_2}}\right), \quad (\text{S72})$$

dove abbiamo posto

$$\Lambda_4 = (1 + c_{11})(1 + c_{22}) - c_{12}^2, \quad (\text{S73})$$

$$\begin{aligned} \Lambda_0 &= \Lambda_4 c_{34} - c_{23}c_{24}(1 + c_{11}) - c_{13}c_{14}(1 + c_{22}) + c_{12}c_{13}c_{24} + c_{12}c_{14}c_{23}, \\ \Lambda_1 &= \Lambda_4(1 + c_{33}) - c_{23}^2(1 + c_{11}) - c_{13}^2(1 + c_{22}) + 2c_{12}c_{13}c_{23}, \\ \Lambda_2 &= \Lambda_4(1 + c_{44}) - c_{24}^2(1 + c_{11}) - c_{14}^2(1 + c_{22}) + 2c_{12}c_{14}c_{24}. \end{aligned} \quad (\text{S74-S76})$$

Tali formule sono state ricavate in

40, 41

e garantiscono la valutazione analitica dei contributi di ordine fino a quattro campi locali nelle equazioni di moto.