Analisi Reti Neurali su Dataset DARWIN per Diagnosi dell'Alzheimer

Alfieri Giuseppe; **Cannavale Achille**; **Colacicco Nunziamaria**; **La Torre Noemi**

Università Degli Studi di Cassino e del Lazio Meridionale

Corso Di Laurea Magistrale in Ingegneria Informatica

Abstract: Questo lavoro esplora l'uso di reti neurali MLP per la classificazione precoce del morbo di Alzheimer a partire da dati calligrafici. Il dataset DARWIN, composto da campioni di scrittura di soggetti sani e pazienti, è stato analizzato su più esecuzioni indipendenti per valutare la stabilità e l'affidabilità del modello in scenari realistici.

1 Introduzione

La diagnosi precoce dell'Alzheimer è essenziale per migliorare la qualità della vita dei pazienti, ma le metodologie tradizionali risultano spesso invasive e costose. L'analisi della scrittura a mano rappresenta un'alternativa promettente, permettendo di individuare segnali precoci della malattia in modo non invasivo. In questo studio, utilizziamo reti neurali MLP per classificare i dati del dataset DARWIN, che contiene 451 caratteristiche estratte da scritture di 174 partecipanti (pazienti e soggetti sani). Analizziamo l'impatto di diverse architetture, funzioni di attivazione, strategie di learning rate e tecniche di regolarizzazione, valutando la stabilità e l'accuratezza della classificazione. L'obiettivo è identificare le configurazioni ottimali per migliorare la capacità predittiva del modello, fornendo un contributo utile all'uso dell'intelligenza artificiale nella diagnosi dell'Alzheimer.

2 Fasi del progetto

2.1 Analisi del Dataset

Riduzione dimensionale tramite PCA

Il dataset `DARWIN` presenta una configurazione sbilanciata tra il numero di campioni ($n = 174$) e il numero di feature ($d = 450$). Questa condizione, nota in letteratura come *curse of dimensionality*, può causare fenomeni di overfitting, degrado delle performance del modello e instabilità nei processi di generalizzazione.

In aggiunta, molte delle feature risultano fortemente correlate tra loro, introducendo ridondanza informativa che tende a ridurre il rapporto segnale-rumore. Per mitigare tali problematiche, è stata applicata l'**Analisi delle Componenti Principali (PCA)** come tecnica di riduzione dimensionale non supervisionata. L'obiettivo della PCA è quello di proiettare i dati in uno spazio a dimensionalità inferiore, preservando al contempo la massima varianza informativa. Questa trasformazione consente di:

- Eliminare le correlazioni lineari tra le feature originali;
- Migliorare l'efficienza computazionale nei processi di training;
- Ridurre il rischio di overfitting, rendendo il modello più robusto;
- Facilitare la visualizzazione e l'analisi qualitativa dei dati.

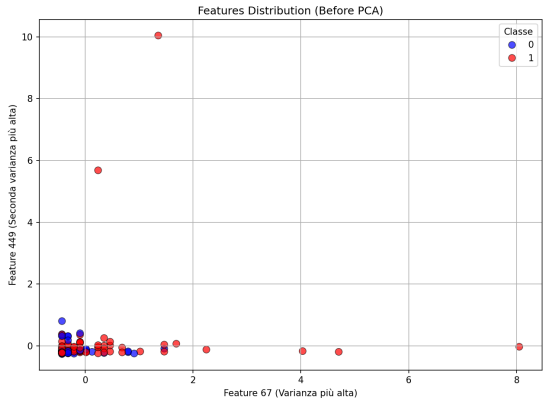
Per valutare l'affidabilità del modello, ogni configurazione di iperparametri è stata testata su **30 run indipendenti**, ciascuna eseguita con un diverso valore di `random_state`. In ogni run, il dataset viene suddiviso in training e test set differenti, e la PCA viene ricalcolata partendo dai dati di training, influenzando così leggermente il numero di componenti principali ottenute.

Nel caso riportato nelle figure, relativo a una singola run rappresentativa, il dataset originario composto da 450 feature è stato ridotto a 99 componenti principali, preservando il 97.03% della varianza totale. Questa riduzione, basata su una soglia fissa di varianza da mantenere (97%), permette di comprimere

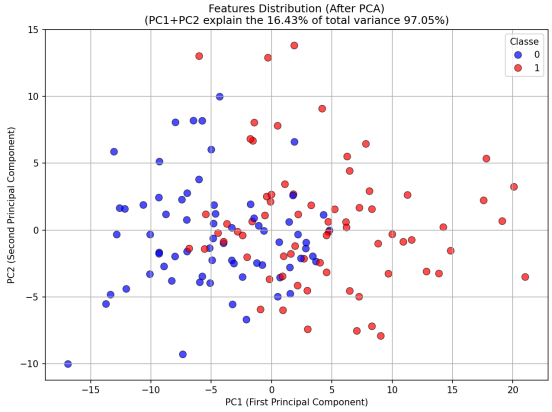
l'informazione riducendo la dimensionalità, pur garantendo una rappresentazione informativa utile per l'apprendimento automatico.

Le due proiezioni mostrate si riferiscono quindi a questa specifica run, e illustrano come la trasformazione tramite PCA abbia migliorato la dispersione dei dati nello spazio, facilitando la separazione tra le classi a beneficio del classificatore MLP.

Nella Figura 1 viene presentato un confronto tra lo spazio originale e quello trasformato.



(a) Distribuzione dei dati nello spazio originale usando le due feature con varianza più alta.



(b) Distribuzione dei dati proiettati sulle prime due componenti principali (PC1 e PC2) dopo la PCA.

Figura 1: Confronto tra la rappresentazione originale e quella ridotta tramite PCA, relativo a una singola run (su 30) eseguita con un particolare seed casuale.

In particolare:

- La **Figura 1a** mostra la distribuzione dei dati nello spazio originale, proiettando i punti sulle due feature con la varianza più alta.
- La **Figura 1b** mostra invece la proiezione dei dati sulle prime due componenti principali (PC1 e PC2) dopo la PCA. Queste due componenti da sole spiegano il **16.43%** della varianza.

È interessante osservare che, nonostante la bassa quota di varianza spiegata dalle sole due componenti (rispetto alla soglia del 97% mantenuta complessivamente), è già possibile intravedere una parziale separazione tra le due classi. Questo suggerisce che la trasformazione tramite PCA ha contribuito a una migliore dispersione e organizzazione dei dati nello spazio, facilitando il compito del classificatore MLP.

3 Metodologia sperimentale

Per ogni configurazione di iperparametri considerata, è stato adottato un approccio robusto alla valutazione delle prestazioni del modello mediante l'esecuzione di **30 run indipendenti**, ciascuna caratterizzata da un valore diverso di `random_state`, incrementato a partire da 42.

Tutte le 30 esecuzioni condividono esattamente la stessa configurazione di iperparametri; l'unica variabile che cambia è la partizione dei dati train/test e l'inizializzazione casuale del modello, al fine di valutare la variabilità dei risultati e la stabilità del modello.

Durante ogni run, i dati vengono divisi in training e test set tramite uno `stratified split`, la PCA viene ricalcolata a partire dal training set, e il classificatore viene addestrato e valutato. Le metriche ottenute (accuracy, AUC, matrice di confusione, curva di loss) vengono salvate per l'analisi aggregata finale.

3.1 Esperimento "ottimo"

Tra tutte le configurazioni esplorate, l'esperimento che ha mostrato le prestazioni più promettenti è quello descritto in Tabella 1, eseguito su 30 run indipendenti con seed incrementale.

Questa configurazione è risultata particolarmente efficace per diversi motivi.

- **Uso del solver** `adam`: ottimizzatore robusto che combina i vantaggi di AdaGrad e RMSProp, ideale per dataset rumorosi e non stazionari.
- **Attivazione** `tanh`: consente di modellare efficacemente le non linearità presenti nei dati, rivelandosi più espressiva rispetto alla `relu` in contesti con dati standardizzati.

Alfieri Giuseppe, Cannavale Achille, Colacicco Nunziamaria, La Torre Noemi

Tabella 1: Configurazione iperparametri dell'esperimento migliore.

Parametro	Valore
Hidden Layers	[400, 200]
Attivazione	`tanh`
Solver	`adam`
α (L2 regularization)	0.001
Batch Size	16
Learning Rate	`adaptive`
Learning Rate Init	0.1
Validation Fraction	0.1
Early Stopping	`true`
n_iter_no_change	10

- **Regolarizzazione L2 con** $\alpha = 0.001$: scelta intenzionale per contrastare l'overfitting, particolarmente importante in questo dataset caratterizzato da un numero ridotto di campioni (174) e un'elevata dimensionalità (451 feature), molte delle quali fortemente correlate.

- **Learning rate** `adaptive` con inizializzazione alta (0.1): ha permesso al modello di adattare dinamicamente la velocità di apprendimento, iniziando con aggiornamenti rapidi per poi stabilizzarsi.

- **Early stopping**: ha garantito una buona generalizzazione, evitando il sovra-addestramento sulle epoche successive.

Complessivamente, questa configurazione ha mostrato un buon equilibrio tra accuratezza in training e test, nonché una ridotta varianza tra le run, suggerendo un comportamento robusto e stabile del modello. Qui di seguito vengono riportati i grafici valutativi del miglior esperimento (**Figure 2 e 3**):

3.2 Esperimenti sub-ottimi

Oltre alla configurazione ottimale, sono stati analizzati anche diversi setup alternativi che, pur avendo mostrato prestazioni inferiori, hanno fornito informazioni preziose sull'influenza di singoli iperparametri. Variando un iperparametro sono stati fissati tutti gli altri. Qui di seguito riportiamo due di queste configurazioni.

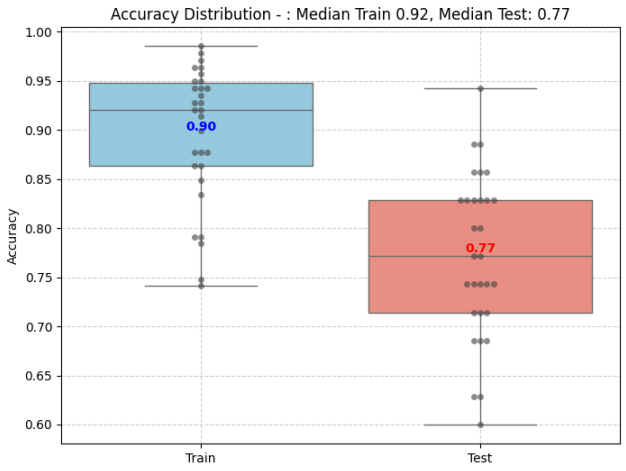


Figura 2: Distribuzione di Accuratezza del Training Set e del Test Set per il miglior esperimento.

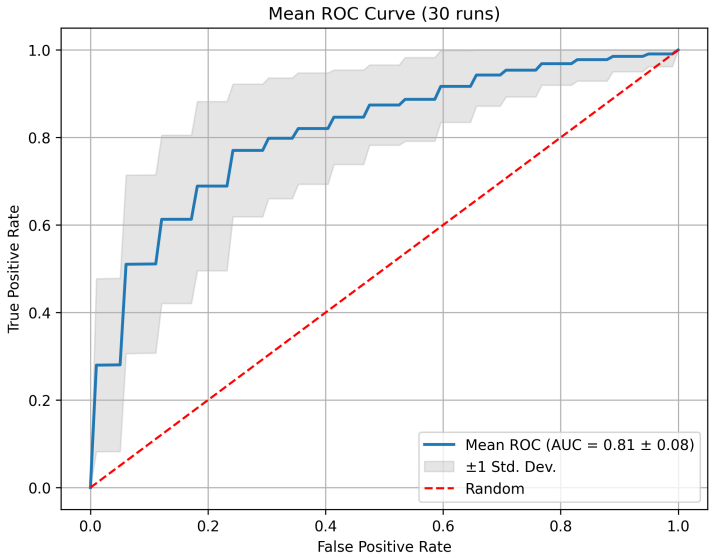


Figura 3: Curva ROC e AUC per il miglior esperimento.

Alfieri Giuseppe, Cannavale Achille, Colacicco Nunziamaria, La Torre Noemi

3.3 Activation: ReLU

In questa configurazione, l'unico cambiamento rispetto all'esperimento ottimale è l'adozione della funzione di attivazione `ReLU` al posto di `tanh`.

Nonostante la `ReLU` sia ampiamente utilizzata per la sua semplicità computazionale e per la capacità di mitigare il problema del vanishing gradient, in questo caso ha mostrato un comportamento instabile, con un'elevata varianza nei risultati tra le diverse run.

Questo è attribuibile alla natura del dataset, che presenta feature standardizzate ma anche una significativa componente di rumore e correlazione: in tali condizioni, la funzione `tanh` ha dimostrato una maggiore capacità di modellazione.

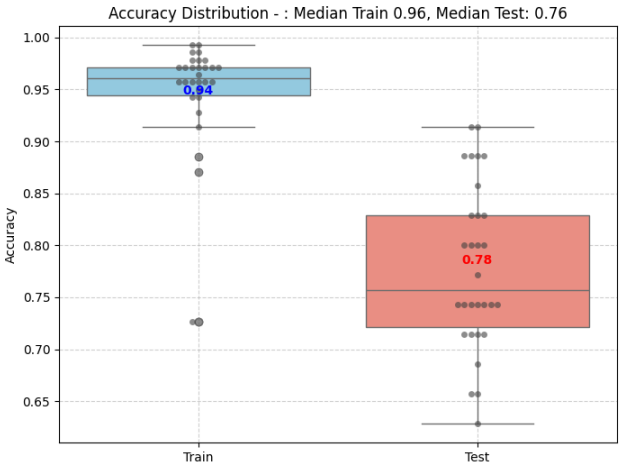


Figura 4: Distribuzione di Accuratezza del Training Set e del Test Set per un esperimento peggiorativo (Attivazione: ReLU).

3.4 Batch Size: 36

L'incremento del **batch size** ha influito negativamente sul processo di ottimizzazione. Un batch più grande comporta aggiornamenti dei pesi più stabili ma meno frequenti, riducendo la capacità del modello di adattarsi rapidamente alle variazioni nei dati. Questo ha portato a un apprendimento meno reattivo e a una tendenza più marcata all'overfitting: il modello ottiene buoni risultati sul training set, ma generalizza peggio sui dati di test.

In particolare, si è osservato che durante molte run il modello continuava a migliorare la performance in training senza ottenere benefici significativi in validazione, segno di un adattamento eccessivo ai pattern specifici del training set.

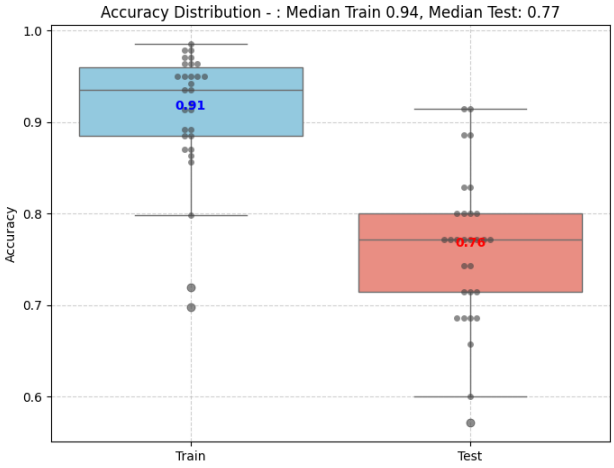


Figura 5: Distribuzione di Accuratezza del Training Set e del Test Set per un esperimento peggiorativo (Batch-Size: 36).

Alfieri Giuseppe, Cannavale Achille, Colacicco Nunziamaria, La Torre Noemi