



Il Tuo Nome

**Università degli Studi di Firenze**  
Dipartimento di Ingegneria / Informatica

27 febbraio 2026

# Agenda della Presentazione

- 1 Introduzione Teorica
- 2 Setup Sperimentale
- 3 Analisi dei Risultati (Exp 1, 2, 3)
- 4 Analisi delle Rappresentazioni (CKA)
- 5 Conclusioni

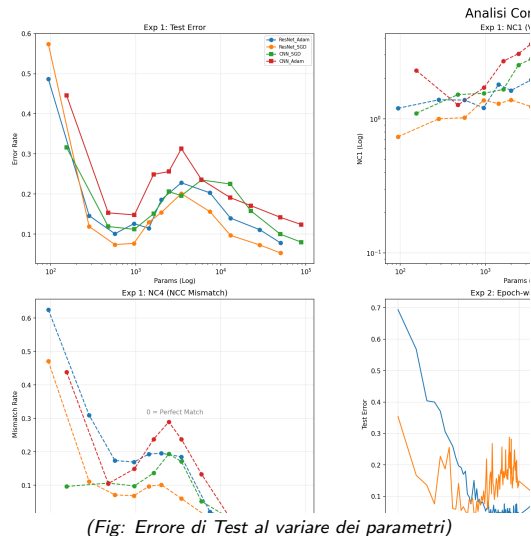
# Il Paradigma Moderno: Deep Double Descent

## La Teoria Classica:

- Trade-off Bias-Varianza a forma di "U".
- Modelli troppo complessi (*overparameterized*) memorizzano il rumore e falliscono sul test set (*overfitting*).

## La Realtà del Deep Learning:

- Superata la **soglia di interpolazione** (capacità limite), l'errore di test torna a scendere.
- Modelli massivamente sovra-parametrizzati generalizzano in modo eccellente.



# Cos'è il Neural Collapse (NC)?

Nelle fasi terminali dell'addestramento, le rappresentazioni nell'ultimo strato nascosto mostrano 4 proprietà geometriche sorprendenti (Papayan et al., 2020):

- ① **NC1 (Variability Collapse):** Le attivazioni della stessa classe convergono verso il loro centro medio (varianza intra-classe  $\rightarrow 0$ ).
- ② **NC2 (Simplex ETF):** Le medie delle classi si dispongono simmetricamente nello spazio, massimizzando la distanza tra loro (Equiangular Tight Frame).
- ③ **NC3 (Self-Duality):** I pesi del classificatore lineare si allineano perfettamente ai centri di classe.
- ④ **NC4 (NCC):** Le previsioni della rete sono identiche a quelle basate sulla distanza dal centro di classe più vicino.

**Obiettivo:** Unire empiricamente Double Descent, Neural Collapse e analisi delle similarità tra strati (CKA).

- **Dataset:** MNIST-1D. Dati a bassa dimensionalità ma con proprietà morfologiche simili a quelle delle immagini.
- **Rumore introdotto:** **20% Label Noise**. Fondamentale per esacerbare il picco di interpolazione (forza la rete a memorizzare etichette errate).
- **Architetture:** ResNet1D e StandardCNN1D.
- **Scaling:** Larghezza (*width*) variabile da  $W = 2$  a  $W = 64$ , portando il modello da  $10^2$  a  $10^5$  parametri.
- **Ottimizzatori:** Confronto tra SGD (con momento e Cosine Annealing) e Adam.

# Risultati: Double Descent e Rottura del Collapse

- Attorno ai  $10^3 - 10^4$  parametri, l'Errore di Test presenta il **picco critico**. Il modello usa tutta la sua capacità per memorizzare il 20% di rumore.
- **Scoperta chiave:** L'errore di generalizzazione correla perfettamente con la metrica **NC1**. Nel regime critico, la varianza intra-classe esplode.
- Nel regime sovra-parametrizzato ( $> 10^4$  param), il Neural Collapse si ripristina: lo spazio latente si riorganizza geometricamente.

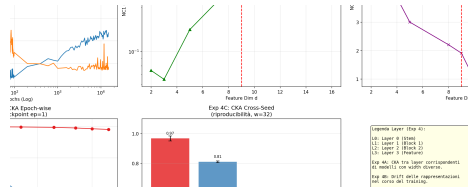
# L'Impatto dell'Ottimizzatore: SGD domina

- Dai grafici si nota una netta distinzione tra ottimizzatori nel regime "overparameterized".
- **SGD:** Mostra un Double Descent più morbido e raggiunge metriche di Neural Collapse (NC2, NC3) nettamente migliori (valori prossimi allo zero).
- **Adam:** Spesso si ferma in minimi locali "peggiori" geometricamente.
- **Spiegazione:** SGD possiede un *bias induttivo implicito* che penalizza la norma dei pesi e favorisce confini decisionali ampi e simmetrici (massimo margine).

# I Limiti Dimensionali del Neural Collapse

La matematica ci dice che, per ospitare un Simplex ETF perfetto con  $K$  classi, lo spazio deve avere dimensione  $d \geq K - 1$ .

- In MNIST-1D abbiamo  $K = 10$  classi.
- Ci aspettiamo il collasso solo per dimensioni latenti  $d \geq 9$ .
- Variando la dimensione finale  $d$  di una ResNet, la metrica **NC2 crolla verticalmente esattamente a  $d = 9$** , confermando la teoria in modo magistrale.





# Come evolve la Rete all'interno? Il CKA

**CKA (Centered Kernel Alignment)** è una metrica ( $0 \rightarrow 1$ ) che misura la similarità tra due spazi di attivazioni, essendo invariante a rotazioni ortogonali e allo scaling.

Abbiamo usato il CKA per rispondere a 3 domande:

- 1 Reti "piccole" e "grandi" imparano le stesse cose? (*Cross-Architettura*)
- 2 Come cambiano le rappresentazioni nel tempo? (*Epoch-wise Drift*)
- 3 Se cambio il seed casuale iniziale, ottengo la stessa geometria? (*Cross-Seed Reproducibility*)

# CKA: Differenze Strutturali (Small vs Large)

Confrontando una rete critica ( $W = 8$ ) e una sovra-parametrizzata ( $W = 64$ ):

- **Primi Strati (L0, L1):** CKA moderato/alto ( $\approx 0.9 - 0.6$ ). Entrambi i modelli estraggono feature a basso livello simili (frequenze, bordi base del segnale 1D).
- **Strati Profondi (L3):** CKA crolla ( $\approx 0.34$ ).
- **Perché?** Il modello piccolo fallisce nel separare le classi (manca di gradi di libertà per collassare), mentre il grande riesce a formare un Simplex ETF. Le topologie prima del classificatore sono quindi diametralmente opposte.

# CKA: La Deriva delle Rappresentazioni (Epoch Drift)

Confrontando le attivazioni durante il training rispetto alla prima epoca:

- **Lo Stem (L0)** converge quasi subito. I filtri di convoluzione primari non necessitano di grandi aggiornamenti.
- **Gli strati profondi (L2, L3)** subiscono una deriva costante e massiccia (*drift*).
- Anche quando la loss di classificazione è quasi zero, lo spazio latente in L3 continua a modificarsi per migliaia di epoche, guidato dalla compressione della varianza tipica del Neural Collapse.

Addestrando lo stesso modello identico, cambiando solo il Seed iniziale:

- L0 è quasi deterministico ( $CKA \approx 0.97$ ).
- **L3 ha un CKA di appena 0.59.**
- **Il motivo:** Degenerazione dello spazio latente. Ci sono infiniti modi per ruotare uno spazio a  $N$  dimensioni mantenendo un Simplex perfetto. Modelli con seed diversi convergono a "rotazioni isomorfe" differenti: risolvono il problema allo stesso modo, ma le coordinate assolute dei neuroni non corrispondono.

- ❶ **Il legame Empirico:** Il fenomeno del Double Descent è intimamente legato alla distruzione e successiva ricostruzione del Neural Collapse nel regime sovra-parametrizzato.
- ❷ **Ottimizzazione:** SGD, grazie al suo bias implicito di regolarizzazione, risulta decisivo per la convergenza verso un Simplex ETF perfetto rispetto ad Adam.
- ❸ **Geometria Profonda:** L'analisi CKA rivela che la rete non apprende uniformemente. Gli strati iniziali fungono da estrattori statici, mentre l'ultimo strato compie una complessa e continua danza per comprimere lo spazio e massimizzare i margini geometrici.

# Grazie per l'attenzione!

Domande?