

Introduzione e Obiettivi della Tesi (Slide 1-2: 30 secondi)

Buongiorno a tutti. Sono Martin Tomassi, e sono qui oggi per presentarvi la mia tesi di laurea: **"Valutazione Metodologica ed Applicativa di KAN, MLP, Random Forest e XGBoost con Tecniche di Ottimizzazione su differenti casi di studio"**.

La tesi è stata strutturata sulla base di due obiettivi principali:

1. **Studio Teorico**, dove ho Analizzato i fondamenti matematici e le architetture dei modelli, dedicando particolare attenzione alle **Kolmogorov-Arnold Networks (KAN)**.
2. **Analisi Metodologica ed Applicativa**, dove ho valutato l'efficacia dei modelli in scenari reali differenti per tipologia di problema e natura dei dati, tramite:
 - **Addestramento e Valutazione**
 - **Studio di Ablazione**, dove ho analizzato l'impatto del **pruning** per misurare il trade-off tra la complessità del modello e le sue performance.

MLP e Backpropagation (Slide 3-4: 45 secondi)

Prima di esplorare le KAN, richiamo brevemente le **MLP**, che sono la base delle reti neurali tradizionali. Una MLP è composta da vari strati: uno di input, uno di output e vari strati nascosti. Il processo chiave in ogni neurone si svolge in due fasi: gli input subiscono una **somma ponderata** con i pesi ed il risultato viene poi fatto passare attraverso una **funzione di attivazione** non-lineare. L'apprendimento avviene tramite l'algoritmo di **Backpropagation** che propaga **all'indietro** l'errore calcolato dalla Loss nello strato di output. Sfruttando la **Chain Rule** ed il **Gradient Descent**, i pesi ed i bias vengono aggiornati in modo da minimizzare l'errore complessivo.

B-Spline e KART (Slide 5-6: 2 minuti e 30 secondi)

Il cuore matematico delle KAN risiede nelle **B-Spline** che sono **funzioni polinomiali a tratti** utilizzate nelle KAN per definire le attivazioni in modo flessibile ed appredibile. A differenza delle curve di Bézier, che non hanno controllo locale e diventano computazionalmente pesanti all'aumentare dei punti di controllo, le B-Spline sono:

- **Continue e differenziabili.**
- Hanno **controllo locale**.
- I **coefficienti** sono **addestrabili tramite backpropagation**.

Il teorema alla base dell'architettura delle KAN è il **Kolmogorov-Arnold Representation Theorem**. Questo teorema afferma che ogni funzione continua multivariata può essere espressa come una composizione di somme di funzioni continue univariate. Il teorema, però, è solo **esistenziale**, non costruttivo: ci dice che la rappresentazione esiste, ma non ci fornisce

un metodo esplicito per determinare le funzioni univariate.

KAN (Slide 7-8: 2 minuti)

La vera innovazione delle **KAN** si trova nella modifica dell'architettura neurale. Mentre le MLP hanno **pesi scalari** apprendibili sugli archi, le KAN sostituiscono questi pesi con **funzioni univariate parametriche ed apprendibili**. Inoltre, le MLP alternano operazioni lineari a non lineari. Le KAN, invece, sostituiscono ogni prodotto peso-input con una di queste funzioni parametriche basate su B-Spline.

Questo approccio permette alle KAN di unire i vantaggi di entrambi i mondi:

- Dalle **Spline**: **Efficacia a bassa dimensionalità** e **controllo locale**.
- Dalle **MLP**: la **Backpropagation**, la **composizionalità** e la **scalabilità**.

Il risultato più importante è che le KAN riescono a **superare la *curse of dimensionality*** che affligge le MLP, dato che l'errore di approssimazione dipende dalla risoluzione delle Spline e non dalla dimensionalità dell'input, portando a **scaling laws più favorevoli**.

Altri Modelli e Metodologia (Slide 9-12: 1 minuto e 15 secondi)

Durante lo studio, ho confrontato le KAN e MLP con altri modelli:

- **XGBoost**: Un algoritmo di **ensemble sequenziale** in cui ogni albero corregge gli errori del precedente.
- **Random Forest**: Un algoritmo di **ensemble parallelo** che combina le previsioni di alberi addestrati su sottoinsiemi casuali di features e dati.
- **CNN**: Nel caso delle immagini, ho applicato una modifica cruciale alle CNN, dove i classificatori *fully-connected* standard sono stati sostituiti da una **KAN**, per un confronto diretto.

Per l'ottimizzazione degli iperparametri, ho scelto il **Random Search** per la sua **efficienza, semplicità di implementazione, scalabilità e parallelizzazione**.

L'intero processo è stato implementato utilizzando **PyTorch, pykan, scikit-learn e XGBoost**, sfruttando il **Cluster HPC dell'Università** per l'allenamento intensivo.

Per la valutazione, ho standardizzato le metriche per regressione e classificazione, utilizzando gli **intervalli di confidenza bootstrap** per stimare l'incertezza delle metriche.

Casi di Studio (Slide 14-21: 2 minuti)

Passando ai risultati, la tesi si è concentrata su tre casi di studio, ma verranno presentati

solamente gli ultimi due.

Nel **Caso 2** mi sono occupato della **Classificazione dell'Inquinamento da PM2.5** in 6 classi AQI. Ho utilizzato un vasto dataset contenente misurazioni orarie degli inquinanti e variabili meteo da varie città indiane nel periodo 2010-2023.

Durante la fase di **Data Preparation**:

- Ho rimosso le colonne ridondanti e non significative alla classificazione.
- Ho gestito i Missing Values eliminando le colonne con più del 40% di dati mancanti ed i restanti sono stati sostituiti tramite forward-fill o con la media.
- Ho effettuato un ricampionamento a livello statale con media giornaliera per semplificare l'analisi.
- Ho rimosso gli Outlier, tramite Isolation Forest.

Da questi risultati, si può vedere che:

- **XGBoost** è il migliore a livello di performance.
- **KAN** ha ottenuto il **miglior compromesso** tra performance e complessità.
- Nello studio di ablazione, il Random Forest ha mostrato una **straordinaria resilienza al pruning**, mentre i restanti modelli hanno avuto risultati pressoché uguali.

Nel **Caso 3**, ho dovuto **Classificare l'Età di una persona a partire da immagini del volto**, utilizzando il **Dataset UTKFace** per classificare l'età in 4 fasce predefinite.

Durante la fase di **Data Preparation**:

1. Ho prima **bilanciato il dataset** campionando un numero quasi uguale di immagini per fascia d'età, per evitare che il modello privilegiasse le classi più rappresentate.
2. Successivamente, tutte le immagini sono state ridimensionate in forma quadrata e **normalizzate**.

Per quanto riguarda l'architettura, abbiamo confrontato **CNN standard** e **CNN+KAN**. La **CNN+KAN** ha avuto delle performance migliori, ma la **CNN standard** è stata considerata come la soluzione più **efficiente**, offrendo un eccellente compromesso tra performance e complessità.

Conclusioni (Slide 22: 30 secondi)

In conclusione, la scelta del modello ideale dipende sempre dal tipo del problema e dai vincoli operativi.

In questo studio:

- **XGBoost** è il modello più performante.
- **KAN** è il modello con il miglior compromesso tra performance e complessità.

- **CNN standard** è la soluzione più efficiente per l'analisi delle immagini.
- Infine, la resilienza al pruning dei vari modelli dipende dalla natura del problema.

Quindi, le KAN non sono un sostituto universale delle MLP, ma un'alternativa valida e promettente.