

RELAZIONE PROGETTO DEEP LEARNING

Introduzione

Il progetto si colloca nell'ambito del deep learning applicato alla fluidodinamica, con l'obiettivo di ricostruire campi fluidodinamici bidimensionali a partire da dati acquisiti da microfoni ad alta frequenza. L'analisi di questi dati è particolarmente complessa a causa dell'elevata frequenza di campionamento, della distribuzione spaziale limitata dei sensori e della necessità di estrarre strutture coerenti nel tempo. Per affrontare queste sfide, è stato sviluppato un workflow che prevede il pre-processing avanzato dei dati, la loro interpolazione su una griglia regolare e l'addestramento di una rete neurale convoluzionale basata su U-Net per la ricostruzione dei campi fluidodinamici.

Il progetto non si limita a una semplice elaborazione dei dati grezzi, ma integra tecniche di filtraggio del segnale, riduzione della dimensionalità e interpolazione spaziale per creare un dataset strutturato idoneo all'addestramento di modelli di deep learning. La scelta di una rete U-Net è motivata dalla sua capacità di preservare sia le caratteristiche locali che la struttura globale dei dati, permettendo una ricostruzione accurata dei campi fluidodinamici.

Descrizione del Dataset

Il dataset (https://data.nasa.gov/Aerospace/Turbulent-boundary-layer-wall-pressure-flight-data/5pvz-kbwd/about_data) utilizzato è costituito da misurazioni acquisite tramite microfoni ad alta frequenza durante esperimenti fluidodinamici. I dati presentano le seguenti caratteristiche principali:

- **Frequenza di campionamento:** 204.8 kHz, successivamente ridotta a 50 kHz tramite filtraggio passa-basso per ridurre il rumore e il carico computazionale.
- **Formato dei dati:** File .mat contenenti registrazioni su più canali, con variazioni tra le sessioni di volo (B101, B106, B109).
- **Struttura temporale:** Le analisi si concentrano sui primi 5 secondi di registrazione per limitare l'uso della memoria e garantire un'elaborazione efficiente.
- **Interpolazione spaziale:** I dati acquisiti da una distribuzione limitata di microfoni vengono interpolati su una griglia regolare 2D (es. 128×128) per essere trattati come immagini in un framework di deep learning.

L'implementazione è stata realizzata in Python utilizzando librerie come numpy, scipy, tensorflow e matplotlib, con un'architettura modulare che permette di adattare facilmente i parametri di elaborazione e addestramento a diverse configurazioni hardware.

Obiettivo

L'obiettivo principale del progetto è la ricostruzione di un campo di dati fluidodinamici acquisiti da un insieme di microfoni, utilizzando un'architettura di rete neurale U-Net 2D in modalità autoencoder. I dati grezzi, forniti in formato .mat, contengono misurazioni di pressione registrate su un elevato numero di canali, con frequenze di campionamento molto alte (fino a 204.8 kHz) e durate di diversi secondi.

Fase 1: Caricamento e Preprocessing dei Dati

Per gestire al meglio i dati acquisiti, sono state adottate le seguenti strategie:

- **Limitazione del volume di dati** – Per prevenire errori di memoria, sono stati caricati solo i primi N secondi di registrazione (tipicamente 5-10) per ciascun canale, selezionando un massimo di 65 canali, corrispondenti al numero di microfoni fisici disponibili.
- **Normalizzazione** – Ogni canale è stato normalizzato nell'intervallo $[-1,1]$, per rendere il training più stabile e abilitare l'uso di tipi float16, riducendo così il consumo di memoria.
- **Downsampling** – La frequenza di campionamento è stata ridotta da 204.8 kHz a 50 kHz tramite un filtro passa-basso Butterworth seguito da un'operazione di *resampling*. Questo passaggio ha ridotto significativamente la dimensione dei dati, alleggerendo il carico computazionale.

Fase 2: Creazione del Dataset Multi-Time-Step

Dopo il preprocessing, i dati sono stati trasformati in un formato adatto all'addestramento della rete neurale:

- **Interpolazione spaziale** – A intervalli regolari (ad esempio ogni 1000 campioni temporali), è stato estratto un vettore di valori (uno per ciascun microfono) e interpolato su una griglia 2D 128x128. Questa griglia costituisce l'input alla rete neurale, consentendo di trattare il problema come una ricostruzione di immagini.

- **Data augmentation** – Sono stati applicati trasformazioni casuali, come flip orizzontali/verticali e leggere rotazioni, per aumentare la varietà dei dati e ridurre il rischio di overfitting.

Fase 3: Architettura della Rete Neurale

Per la ricostruzione dei dati fluidodinamici, è stata implementata una rete **U-Net 2D** profonda, caratterizzata da:

- **Struttura a più livelli** – La rete include tre livelli di downsampling con convoluzioni che utilizzano 32, 64, 128 e 256 filtri, seguiti dai corrispondenti livelli di upsampling.
- **Regolarizzazione e stabilizzazione** – Sono stati impiegati BatchNormalization e Dropout (con valori compresi tra 0.1 e 0.2) per prevenire il sovradattamento e migliorare la stabilità del training.
- **Funzione di loss combinata** – La funzione di perdita adottata è una combinazione di MSE (Mean Squared Error) e SSIM (Structural Similarity Index), espressa come:

$$\text{Loss} = \text{MSE} + \alpha(1 - \text{SSIM})$$

La MSE garantisce fedeltà numerica nella ricostruzione dei dati, mentre la SSIM enfatizza la conservazione delle strutture spaziali nell'output della rete.

Fase 4: Training Avanzato

Per ottimizzare l'addestramento della rete e garantire una generalizzazione efficace, sono stati adottati i seguenti accorgimenti:

1. Callback per il controllo del training

- **EarlyStopping** (*patience=10*) – Interrompe il training se la *validation loss* non migliora per 10 epoche consecutive, evitando overfitting e spreco di risorse computazionali.
- **ReduceLROnPlateau** – Riduce automaticamente il *learning rate* se la *validation loss* rimane stagnante per diverse epoche, migliorando la convergenza.
- **ModelCheckpoint** – Salva il modello con le migliori prestazioni sulla validazione, permettendo di riprendere l'addestramento o effettuare test senza doverlo ritrainare completamente.

2. Impostazioni di training

- **Numero di epoche** – Tipicamente tra **50 e 100**, a seconda della convergenza del modello.
- **Batch size** – Impostato a **16**, bilanciando efficienza computazionale e stabilità del gradiente.
- **Validation split** – Il **20%** dei dati è stato riservato alla validazione, per monitorare eventuali fenomeni di overfitting durante l'addestramento.

Difficoltà e Soluzioni Implementate

Lo sviluppo di questo progetto ha presentato diverse sfide, che sono state affrontate con strategie mirate per ottimizzare l'uso della memoria, la qualità delle interpolazioni e la stabilità del training.

Uno degli ostacoli principali è stato l'enorme quantità di dati da gestire. I file .mat contenevano registrazioni ad alta frequenza su più canali, rendendo difficile il caricamento e la manipolazione in RAM senza incorrere in errori di memoria. Per superare questo problema, è stata adottata una strategia di caricamento selettivo, limitando i dati ai primi cinque secondi di registrazione e riducendo il numero di canali considerati a un massimo di 65, corrispondente al numero di microfoni fisici presenti nel sistema. Inoltre, l'uso del formato float16 ha permesso di dimezzare il consumo di memoria senza una significativa perdita di precisione. Un altro passaggio fondamentale è stato l'applicazione di un filtro passa-basso Butterworth, seguito da downsampling da 204.8 kHz a 50 kHz, riducendo il numero di campioni da elaborare e alleggerendo il carico computazionale.

Un'altra difficoltà è derivata dalla disposizione irregolare dei microfoni, che non consentiva di ottenere direttamente una rappresentazione spaziale continua del campo fluidodinamico. Per affrontare questa problematica, è stata implementata un'interpolazione 2D su una griglia regolare, in modo da convertire i dati grezzi in un formato compatibile con l'input di una rete neurale convoluzionale. Tuttavia, questo processo ha introdotto il rischio di generare valori NaN in zone prive di dati. Per ovviare a questo inconveniente, tutti i valori mancanti sono stati sostituiti utilizzando `np.nan_to_num()`, garantendo così che la rete potesse apprendere senza distorsioni nei dati.

Dal punto di vista dell'addestramento della rete, uno degli aspetti più critici è stato garantire la stabilità della convergenza e ottenere una ricostruzione fedele del campo fluidodinamico. La scelta della funzione di loss combinata, basata su MSE + SSIM, si è rivelata efficace per preservare sia l'errore numerico che la coerenza strutturale delle ricostruzioni. Inoltre, per migliorare il training e prevenire overfitting,

sono stati implementati meccanismi di controllo come EarlyStopping, che interrompe il processo se la *validation loss* smette di migliorare, e ReduceLROnPlateau, che riduce il *learning rate* in caso di stagnazione della performance. Questi accorgimenti hanno permesso di ottimizzare il tempo di addestramento senza compromettere la qualità dei risultati.

Risultati e Analisi

L'analisi delle predizioni della rete ha mostrato risultati promettenti. Le mappe ricostruite risultano coerenti con quelle di input, indicando che la rete ha effettivamente appreso le principali strutture spaziali del campo fluidodinamico. Nelle regioni centrali, le ricostruzioni si sovrappongono quasi perfettamente con i dati target, mentre nelle zone periferiche si osserva un leggero effetto di smooth, dovuto alla capacità della rete di generalizzare in assenza di informazioni dirette dai microfoni. Tuttavia, alcune discrepanze più marcate si notano nei dettagli più fini e nelle regioni meno coperte dai sensori, dove il modello fatica a ricostruire variazioni locali rapide.

Dal punto di vista numerico, la loss combinata (MSE + SSIM) ha mostrato una buona convergenza, stabilizzandosi dopo circa 50-100 epoche di addestramento. Il Mean Absolute Error (MAE) si è ridotto progressivamente, mentre il valore di $1 - \text{SSIM}$ ha evidenziato un miglioramento rispetto all'uso della sola MSE, suggerendo che la rete è riuscita a preservare non solo l'accuratezza numerica, ma anche la coerenza strutturale dell'output.

L'analisi visiva delle predizioni conferma queste osservazioni: le mappe Input e Target risultano molto simili, mentre la Prediction segue fedelmente i pattern principali, con lievi differenze nella saturazione del segnale. Inoltre, riportando i valori ricostruiti nella loro scala originale (0–350), si verifica che il modello mantiene un'aderenza accettabile ai dati fisici reali, con un errore contenuto.

Confronto tra il modello iniziale e la versione definitiva

Nel corso dello sviluppo del progetto, sono state realizzate due versioni del modello per la ricostruzione del campo fluidodinamico a partire dai dati acquisiti dai microfoni. La prima versione (FirstVersion) ha rappresentato un punto di partenza utile per comprendere la struttura del dataset e testare le prime strategie di preprocessing e addestramento. Successivamente, sono state introdotte diverse ottimizzazioni che hanno portato alla versione definitiva del modello, migliorando la qualità delle predizioni e l'efficienza del processo di training.

1. Preprocessing e gestione dei dati

Versione iniziale:

- Caricava l'intero dataset senza alcuna riduzione della dimensionalità, risultando in problemi di memoria (*MemoryError*).
- Non applicava alcun filtro ai dati acquisiti, mantenendo la frequenza di campionamento originale (204.8 kHz).
- L'interpolazione dei dati era gestita in modo meno efficiente, con il rischio di generare valori NaN non adeguatamente trattati.

Versione definitiva:

- Introduzione di un downsampling con filtro passa-basso (da 204.8 kHz a 50 kHz), riducendo significativamente il numero di campioni da elaborare.
- Limitazione dei dati ai primi 5 secondi per ogni canale, riducendo il carico computazionale senza perdere informazioni essenziali.
- Normalizzazione più efficiente dei dati (float16), dimezzando l'occupazione di memoria.
- Interpolazione migliorata con gestione robusta dei valori NaN, garantendo una rappresentazione continua e utilizzabile dal modello.

2. Architettura del modello

Versione iniziale:

- Utilizzava una U-Net 2D, ma in maniera troppo basilare.
- Struttura più semplice, con un numero inferiore di livelli convoluzionali e senza tecniche avanzate di regolarizzazione.
- Perdita basata solo su Mean Squared Error (MSE), che penalizzava l'errore numerico senza considerare la qualità strutturale della ricostruzione.

Versione definitiva:

- Passaggio a una U-Net 2D più avanzata, che tiene conto della disposizione spaziale dei microfoni, migliorando la qualità della ricostruzione visiva.
- Architettura più profonda con BatchNormalization e Dropout, garantendo maggiore stabilità durante il training.
- Introduzione di una loss combinata (MSE + SSIM), che non solo minimizza l'errore numerico, ma preserva la coerenza strutturale dei campi ricostruiti.

3. Training e convergenza

Versione iniziale:

- Il training soffriva di instabilità e fenomeni di overfitting, dovuti alla mancanza di un adeguato monitoraggio della validazione.
- Non era implementato alcun meccanismo per adattare il learning rate.
- Batch size e numero di epoche ottimizzati solo in base alle risorse disponibili, senza particolare attenzione alla generalizzazione del modello.

Versione definitiva:

- Introduzione di EarlyStopping per interrompere l'addestramento in caso di mancati miglioramenti nella *validation loss*.
- Implementazione di ReduceLROnPlateau, che abbassa il learning rate se la perdita di validazione non migliora, facilitando la convergenza.
- Aumento del numero di epoche (da 20 a 50-100) e tuning del batch size, portando a una migliore capacità di generalizzazione.

4. Qualità delle predizioni

Versione iniziale:

- Le mappe predette risultavano meno definite e prive di dettagli strutturali precisi.
- Effetto di oversmoothing, con perdita di dettagli nelle zone di maggiore variazione del campo.

Versione definitiva:

- Le ricostruzioni mostrano un gradiente più coerente e dettagli più nitidi, con una netta riduzione dell'errore.
- L'aggiunta della loss basata su SSIM ha permesso di preservare meglio la struttura spaziale dei dati.
- La denormalizzazione dei risultati ha confermato che il modello riesce a ricostruire valori compatibili con quelli fisici reali.

Conclusioni

L'ottimizzazione della U-Net 2D, insieme a un preprocessing più accurato e un training più controllato, ha portato a un notevole miglioramento della qualità delle ricostruzioni. La versione definitiva del modello è più stabile, efficiente e fornisce risultati significativamente migliori rispetto alla versione iniziale, rendendola più adatta per applicazioni concrete nel campo della fluidodinamica computazionale.

Conclusioni e Sviluppi Futuri

I risultati ottenuti dimostrano che l'approccio basato su U-Net 2D, combinato con l'interpolazione spaziale dei dati microfonici, permette di ottenere una buona ricostruzione del campo fluidodinamico anche a partire da dati dispersi. La scelta di una loss combinata MSE + SSIM si è rivelata efficace nel bilanciare fedeltà numerica e coerenza strutturale, mentre l'utilizzo di float16 e la selezione di una finestra temporale limitata hanno permesso di superare i problemi di memoria, rendendo il training gestibile su hardware standard.

Tuttavia, ci sono diversi aspetti che potrebbero essere migliorati. Un primo passo potrebbe essere aumentare la risoluzione della griglia, portandola a 256×256 , per catturare più dettagli, compatibilmente con le risorse computazionali disponibili. Inoltre, un'interessante evoluzione del progetto potrebbe essere l'integrazione di parametri di volo (come Mach number, altitudine e condizioni atmosferiche) in un modello multimodale, in grado di condizionare la predizione in base alle specifiche condizioni sperimentali.

Un ulteriore miglioramento potrebbe derivare dall'adozione di architetture più avanzate, come Attention U-Net o Physics-Informed Neural Networks (PINNs), che permetterebbero di vincolare l'output della rete a leggi fisiche note, aumentando la fedeltà della ricostruzione. Infine, un'accurata denormalizzazione dei dati consentirebbe di confrontare i valori predetti direttamente con misurazioni fisiche reali, per validare ulteriormente le prestazioni del modello.