

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

**Reti Neurali Convoluzionali
Geodetiche Per Approssimare
Simulazioni CFD**

**Relatore:
Chiar.mo Prof.
ANDREA ASPERTI**

**Presentata da:
LIAM BUSNELLI URSO**

**II Sessione
Anno Accademico 2023/2024**

Deep Learning

Convoluzione Geodetica

Approssimazioni

Superficie Tridimensionale

Informazione Posizionale

Abstract

Questo lavoro presenta lo sviluppo di un modello basato su Reti Neurali Convolutionali Geodetiche (GCNN) per la previsione di parametri fisici su geometrie tridimensionali di pistoni. L'obiettivo principale è ridurre i tempi di calcolo tipici delle simulazioni di fluidodinamica computazionale (CFD), mantenendo un alto livello di accuratezza nelle previsioni. Il modello proposto utilizza convoluzioni geodetiche per operare direttamente sulle superfici delle mesh triangolari, sfruttando coordinate geodetiche polari ottenute attraverso un algoritmo di approssimazione iterativa. L'architettura della rete integra strati specifici come la Convoluzione Geodetica e l'Angular Max-Pooling, permettendo al modello di catturare informazioni locali rispettando la geometria intrinseca delle superfici. I risultati ottenuti mostrano che il modello è in grado di prevedere con buona accuratezza parametri come la temperatura e il rapporto di ricircolo dei gas di scarico (EGR), offrendo un'alternativa efficiente alle simulazioni CFD tradizionali.

Indice

0	Introduzione	1
1	Concetti di Base e Background Teorico	1
1.1	Manifold	1
1.1.1	Definizione Formale	2
1.1.2	Esempi di Manifold	2
1.2	Manifold Differenziabili	3
1.3	Manifold di Riemann	3
1.3.1	Metrica di Riemann	4
1.3.2	La Distanza Geodetica	4
1.4	Coordinate Polari	5
1.5	Coordinate Geodetiche Polari	6
1.5.1	Discretizzazione	6
1.6	Coordinate Baricentriche	7
1.7	Reti Neurali convoluzionali	8
1.7.1	Convoluzioni	8
1.7.2	Reti Neurali Convoluzionali	9
1.7.3	Convoluzioni Geodetiche	9
1.7.4	Reti Convoluzionali Geodetiche	12
1.8	Il formato STL	13
2	Dataset e Preprocessing	15
2.1	Descrizione del Dataset	15
2.1.1	Preprocessing degli Input	18
2.2	Calcolo delle Coordinate Geodetiche Polari su Spazio Discreto	19
2.2.1	Definizione del Problema	19
2.2.2	Approssimazione delle Coordinate Geodetiche Polari	19
2.3	Preprocessing da File STL a Coordinate Baricentriche	21
2.3.1	Normalizzazione delle Geometrie	21
2.3.2	Calcolo del Raggio Geodetico	21

2.3.3	Calcolo delle Coordinate Baricentriche	22
2.3.4	Problema con La Dimensione del Raggio U_{\max}	23
2.3.5	Uniformare le Dimensioni dei Triangoli	25
3	Il modello	27
3.1	Architettura	27
3.2	Descrizione del Modello	27
3.2.1	Pipeline del Modello	28
3.3	Strati specifici	29
3.3.1	Angular Max-Pooling (AMP)	29
3.3.2	Convoluzione Geodetica (GC)	30
3.4	Addestramento del Modello	31
3.4.1	Dataset e Obiettivi	31
3.4.2	Ottimizzazione e Scheduler del Learning Rate	31
3.4.3	Funzione di Loss	33
3.4.4	Strategie per Evitare l'Overfitting	33
3.4.5	Configurazione dell'Addestramento	33
3.4.6	Monitoraggio e Salvataggio del Modello	34
3.5	La Maschera	34
3.5.1	Conclusioni sull'Utilizzo della Maschera	36
3.6	Risultati	37
4	Conclusioni	39
4.1	Sviluppi futuri	39
4.1.1	Convoluzioni Geodetiche Multi-Direzionali	39
4.1.2	Calcolo delle Coordinate Baricentriche	40
4.1.3	Parametri Allenabili nei Coefficienti del Kernel	40
4.2	Conclusioni	42

Elenco delle figure

1.1	Coordinate polari	5
1.2	Rappresentazione delle coordinate baricentriche nel piano. Ogni immagine mostra la variazione di una delle tre coordinate baricentriche. Per ogni punto nel piano, è visualizzato il valore della coordinata baricentrica associata a ciascun vertice del triangolo ABC preso come punto di riferimento. All'interno del triangolo, le coordinate variano tra 0 e 1. Al di fuori del triangolo, la coordinata aumenta oltre 1 in direzione del vertice corrispondente alla coordinata e scende sotto 0 nella direzione opposta.	8
2.1	Ground truth del modello	17
2.2	Illustrazione delle meccaniche di base dell'algoritmo DGPC. Immagini tratte da [18]	20
2.3	Diametro geodetico dello Stanford Bunny. Calcolato con PYGEODESIC. La mesh è colorata in base alla distanza dal punto selezionato sulla coda.	22
2.4	Le due caratteristiche predominanti nelle geometrie del dataset, dal punto di vista della lunghezza dei lati, che è altamente disomogenea .	24
2.5	La lunghezza dei lati delle geometrie si aggira attorno alle 7 unità. Tuttavia ogni geometria presenta lati di lunghezza pari a quasi 120 unità.	25
2.6	A seguito dell'algoritmo di regolarizzazione dei lati, gli outlier sono dimezzati in grandezza.	26
3.1	Architettura del modello	27

3.2	Addestramento dei modelli. Per ogni parametro da prevedere, sono stati addestrati 4 modelli, inizializzati con pesi diversi. Come si vede in un modello dell'EGR, esiste la possibilità che un addestramento rimanga bloccato in minimi locali, o incappi in altri tipi di problemi. Addestrare più modelli permette di aggirare agilmente eventuali inciampi di questo tipo, oltre che selezionare la versione più performante.	34
3.3	Maschera di uno degli elementi del dataset, costituita da valori binari (0 e 1).	35
3.4	Sezione lungo l'asse Y della previsione della turbolenza all'interno di un pistone del dataset.	37
3.5	Sezione lungo l'asse X della previsione della turbolenza all'interno di un pistone del dataset.	37
4.1	Confronto tra insiemi di livello (le linee rosse indicano il valore 0.5) dei kernel in tre generalizzazioni delle convoluzioni nelle varietà. I parametri del kernel sono fissi nel caso delle GCNN e delle ACNN[2] (<i>Anisotropic Convolutional Neural Network</i>), mentre sono invece parametri allenabili nel caso della MoNet. Con questa immagine si vuole evidenziare come la rete abbia imparato a dare più peso ad alcune zone della patch, e meno ad altre, come si evince dalla non simmetria delle curve di livello. Inoltre, la MoNet rappresenta una generalizzazione delle GCNN classiche, in quanto i parametri fissi utilizzati nelle GCNN sono un caso specifico dei parametri allenabili della MoNet. Immagine tratta da [19]	41

Capitolo 0

Introduzione

Le simulazioni numeriche in fluidodinamica computazionale (*Computational Fluid Dynamics*, CFD) sono strumenti fondamentali per l'analisi e la previsione dei fenomeni fluidodinamici in ambito ingegneristico. Tuttavia, tali simulazioni richiedono un elevato costo computazionale, spesso incompatibile con le esigenze di rapidità e ottimizzazione nei processi progettuali.

In questo contesto, l'utilizzo di reti neurali offre una valida alternativa per ridurre i tempi di calcolo, mantenendo al contempo un alto livello di accuratezza nelle previsioni. In particolare, le *Reti Neurali Convoluzionali Geodetiche* (GCNN) rappresentano un'estensione delle reti neurali convoluzionali tradizionali, adattate per operare su superfici e *manifold*, tipici delle geometrie complesse tridimensionali.

Il presente lavoro, svolto durante un tirocinio presso **NAIS**¹, si concentra sull'applicazione delle GCNN per la previsione di parametri fisici, come la temperatura, la turbolenza e il rapporto di ricircolo dei gas di scarico (*Exhaust Gas Recirculation*, EGR), su geometrie tridimensionali di pistoni. Il codice di partenza utilizzato è tratto dal progetto open source *GeoConv*[16]. L'obiettivo è sviluppare un modello in grado di sostituire, o integrare, le simulazioni CFD nelle fasi iniziali di progettazione, fornendo risultati affidabili con una significativa riduzione dei tempi di elaborazione.

La tesi è strutturata come segue:

Nel **Capitolo 1** vengono introdotti i concetti teorici di base necessari alla comprensione del lavoro. Si parte dalla definizione di *manifold*, *manifold* differenziabili e *manifold* di Riemann, e le coordinate geodetiche polari. Viene inoltre presentato il funzionamento delle reti neurali convoluzionali e la loro estensione alle convoluzioni geodetiche confrontandone le caratteristiche fondamentali.

¹<https://www.naisengineering.com/>

Il **Capitolo 2** descrive il dataset utilizzato e le fasi di preprocessing applicate alle geometrie dei pistoni. Si affronta il problema della disomogeneità delle dimensioni dei triangoli nelle mesh e si propone una soluzione per uniformare tali dimensioni, migliorando l'efficacia del modello.

Nel **Capitolo 3** viene presentato il modello sviluppato, dettagliandone l'architettura e gli strati specifici, come l'*Angular Max-Pooling* e la *Convoluzione Geodetica*. Si illustrano inoltre le strategie di addestramento adottate, comprese le tecniche di ottimizzazione e gli scheduler per la regolazione del learning rate.

Infine, nelle **Conclusioni** vengono discussi i risultati ottenuti, evidenziando i vantaggi e le possibili limitazioni del modello proposto, e si delineano potenziali sviluppi futuri del lavoro.

Capitolo 1

Concetti di Base e Background Teorico

1.1 Manifold

Un *manifold* (o varietà) è uno spazio matematico che generalizza il concetto di superficie in uno spazio euclideo, permettendo di descrivere geometrie più complesse[17]. In termini semplici, un manifold è uno spazio che, localmente, assomiglia a uno spazio euclideo di una certa dimensione, ma che può avere una struttura globale più complicata.

Per definire formalmente un manifold, sono necessari i seguenti concetti preliminari:

- **Spazio Topologico:** Uno spazio dotato di una struttura che consente di definire nozioni di vicinanza e continuità. Ogni manifold è uno spazio topologico.
- **Carta e Atlante:** Una carta è una funzione che mappa una porzione dello spazio topologico su uno spazio euclideo locale; un atlante è una collezione di carte che copre l'intero manifold. Una carta (o chart in inglese) è uno strumento matematico che permette di studiare un oggetto geometrico complesso, come una varietà (manifold), traducendo localmente le sue proprietà in qualcosa di più semplice, ovvero uno spazio euclideo. In altre parole, una carta è una mappa che descrive una porzione di una varietà in termini di coordinate "piatte" (simili a quelle usate nello spazio euclideo).

1.1.1 Definizione Formale

Sia M uno spazio topologico. Diciamo che M è un manifold n -dimensionale se soddisfa i seguenti criteri:

1. Ogni punto $p \in M$ ha un intorno aperto $U \subset M$ che è omeomorfo a un sottoinsieme aperto di \mathbb{R}^n , ossia esiste una funzione continua e invertibile:

$$\varphi : U \rightarrow V \subset \mathbb{R}^n,$$

dove V è un sottoinsieme aperto di \mathbb{R}^n , e φ è detta *carta locale*.

2. L'insieme delle carte $\{(U_i, \varphi_i)\}$ forma un *atlante*, ossia una copertura di M tale che ogni $p \in M$ appartiene almeno a una carta (U_i, φ_i) .
3. Se $U_i \cap U_j \neq \emptyset$, sono definibili le *funzioni di transizione* φ_i e φ_j tra le carte, che collegano le coordinate locali delle due carte. Le funzioni di transizione sono mappature del tipo:

$$\varphi_j \circ \varphi_i^{-1} : \varphi_i(U_i \cap U_j) \rightarrow \varphi_j(U_i \cap U_j),$$

e sono differenziabili.

Nello specifico, φ_i e φ_j sono due carte locali definite rispettivamente sugli aperti U_i e U_j del manifold M . L'intersezione $U_i \cap U_j$ rappresenta la regione dove entrambe le carte sono definite. La composizione $\varphi_j \circ \varphi_i^{-1}$ descrive come passare dalle coordinate della carta i alle coordinate della carta j in questa regione di sovrapposizione. Il fatto che queste funzioni siano differenziabili garantisce la regolarità della struttura del manifold nelle zone di transizione tra carte diverse.

Un manifold può essere visto come un'estensione del concetto di curva[12] ($n = 1$) e superficie ($n = 2$) in dimensioni arbitrarie, con applicazioni in geometria, fisica e machine learning.

1.1.2 Esempi di Manifold

- **Linea Retta \mathbb{R} :** La linea retta è un esempio di manifold 1-dimensionale, poiché ogni punto può essere mappato localmente su \mathbb{R}^1 .
- **Cilindro:** Il cilindro può essere visto come un manifold 2-dimensionale, in quanto localmente ogni regione della sua superficie assomiglia a \mathbb{R}^2 .
- **Superficie terrestre:** La superficie terrestre (approssimabile come una sfera) è un esempio di manifold bidimensionale. Localmente possiamo rappresentarla con coordinate geografiche (latitudine e longitudine) o proiezioni piane.

- **Spaziotempo nella relatività generale:** Lo spaziotempo nella teoria della relatività generale di Einstein è una varietà quadridimensionale (3 dimensioni spaziali + 1 dimensione temporale).

1.2 Manifold Differenziabili

Un *manifold differenziabile* è un tipo particolare di manifold che, localmente, è sufficientemente simile a uno spazio vettoriale da consentire l'applicazione degli strumenti del calcolo differenziale. L'idea centrale è che, mentre un manifold può avere una struttura globale complessa e non euclidea, localmente esso può essere descritto da coordinate che appartengono a uno spazio vettoriale \mathbb{R}^n , dove valgono le regole classiche del calcolo.

Per costruire un manifold differenziabile, si utilizza un insieme di *carte* (coordinate locali) organizzate in un *atlante*. Ogni carta associa una porzione del manifold a un sottoinsieme di \mathbb{R}^n attraverso una mappa continua e invertibile. Questo permette di utilizzare il calcolo differenziale all'interno della regione coperta dalla carta, poiché essa "trasforma" localmente il manifold in uno spazio vettoriale.

Tuttavia, per garantire che questa struttura funzioni globalmente sul manifold, le carte non devono essere indipendenti le une dalle altre: esse devono *combaciare* correttamente nelle regioni in cui si sovrappongono. Questo significa che le *mappe di transizione*, ovvero le funzioni che collegano le coordinate locali di una carta con quelle di un'altra nella zona di sovrapposizione, devono essere *differenziabili*. Se queste mappe di transizione soddisfano il requisito di differenziabilità, allora il manifold è considerato differenziabile.

In altre parole, un manifold differenziabile è un manifold topologico arricchito da una *struttura differenziabile*, che permette di estendere il calcolo differenziale a tutto il manifold in modo coerente. Ogni calcolo eseguito in una carta rimane valido in tutte le altre carte, grazie alla compatibilità differenziabile tra le loro mappe di transizione.

Questa proprietà rende i manifold differenziabili strumenti fondamentali in numerosi campi della matematica[6] e della fisica, poiché consentono di studiare oggetti complessi e non euclidei utilizzando strumenti del calcolo classico.

1.3 Manifold di Riemann

Un *manifold di Riemann* è un manifold differenziabile su cui è definita una struttura geometrica aggiuntiva chiamata *metrica di Riemann*. Questa metrica consente

di estendere i concetti classici della geometria euclidea, come distanza, angoli, lunghezza, volume e curvatura, a spazi più complessi. Manifold come lo spazio euclideo \mathbb{R}^n , la sfera S^n , lo spazio iperbolico e superfici lisce tridimensionali (ad esempio, ellipsoidi o paraboloidi) sono tutti esempi di manifold di Riemann.

Questi manifold prendono il nome dal matematico tedesco Bernhard Riemann, che per primo li ha formalizzati, gettando le basi per la geometria differenziale moderna.

1.3.1 Metrica di Riemann

Formalmente, una *metrica di Riemann* su un manifold liscio¹ M è una scelta di prodotto scalare definito positivo g_p su ogni spazio tangente $T_p M$ associato a ciascun punto $p \in M$. Questo prodotto scalare consente di misurare lunghezze e angoli tra vettori tangenti al manifold. In termini matematici, la metrica di Riemann è una funzione che assegna, in modo liscio, una forma bilineare simmetrica e definita positiva:

$$g_p : T_p M \times T_p M \rightarrow \mathbb{R},$$

dove $g_p(v, w)$ rappresenta il prodotto scalare tra i vettori tangenti v e w nello spazio tangente $T_p M$.

L'aggiunta di una metrica di Riemann a un manifold differenziabile consente di eseguire calcoli geometrici e analitici, come:

- La lunghezza di curve definite sul manifold.
- L'integrazione per definire distanze e volumi.
- La derivazione per studiare proprietà locali come la curvatura.

Un manifold differenziabile dotato di una metrica di Riemann (M, g) prende il nome di *manifold di Riemann*[14].

1.3.2 La Distanza Geodetica

Uno dei concetti fondamentali definiti da una metrica di Riemann è la *distanza geodetica*, che generalizza il concetto di distanza euclidea a spazi curvi. La distanza geodetica tra due punti $p, q \in M$ è la lunghezza della curva più corta che li collega, chiamata *geodetica*.

La distanza geodetica è il concetto che generalizza il concetto di distanza "diretta" tra due punti in spazi curvi. In geometria euclidea, la distanza tra due punti è

¹"Liscio" significa che l'oggetto matematico è sufficientemente regolare da permettere di fare calcoli differenziali senza problemi. Indica un alto livello di regolarità, continuità e derivabilità.

semplicemente la lunghezza del segmento di retta che li congiunge. Tuttavia, in uno spazio curvo, la "linea più breve" tra due punti non è sempre una retta, ma una geodetica, cioè una curva che minimizza la distanza tra i punti.

Consideriamo per esempio una sfera in uno spazio tridimensionale. La distanza geodetica tra due punti sulla superficie della sfera passa sempre dagli archi di cerchio massimo².

La distanza geodetica soddisfa le proprietà di una *metrica*:

- $d(p, q) \geq 0$ e $d(p, q) = 0$ se e solo se $p = q$.
- $d(p, q) = d(q, p)$ (simmetria).
- $d(p, r) \leq d(p, q) + d(q, r)$ (disuguaglianza triangolare).

1.4 Coordinate Polari

Le coordinate polari costituiscono un sistema di coordinate bidimensionale in cui ogni punto del piano viene identificato attraverso due componenti: la distanza da un punto fisso, detto *polo* (o origine), e l'angolo formato con una direzione di riferimento, detta *asse polare*.

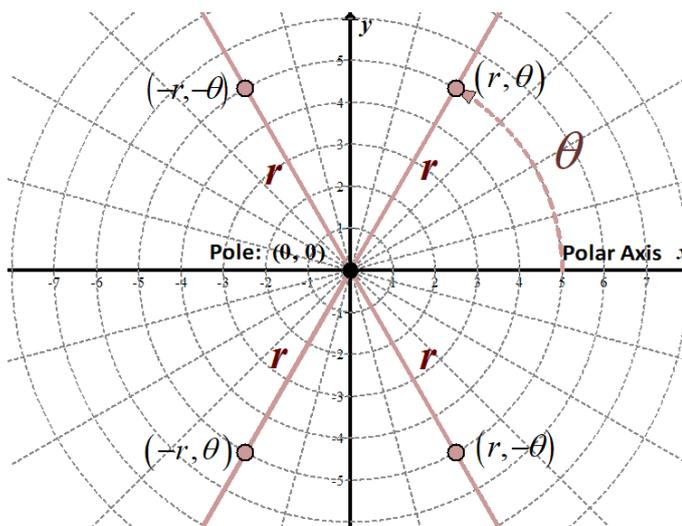


Figura 1.1: Coordinate polari

Formalmente, dato un punto P nel piano, le sue coordinate polari (ρ, θ) sono definite da:

²Il cerchio determinato dall'intersezione fra la sfera ed il piano passante per i due punti ed il centro della sfera.

- ρ (raggio): la distanza non negativa dal polo al punto P .
- θ (angolo polare): l'angolo formato tra l'asse polare e la retta che congiunge il polo con il punto P .

L'asse polare viene comunemente fatto coincidere col semiasse positivo delle ascisse del piano cartesiano, e l'angolo θ viene misurato in senso antiorario a partire da tale asse.

In questo modo, è possibile stabilire una corrispondenza biunivoca tra coordinate polari e coordinate cartesiane attraverso le seguenti relazioni:

$$\begin{cases} x = \rho \cos \theta, \\ y = \rho \sin \theta. \end{cases}$$

E, viceversa:

$$\begin{cases} \rho = \sqrt{x^2 + y^2}, \\ \theta = \arctan\left(\frac{y}{x}\right). \end{cases}$$

1.5 Coordinate Geodetiche Polari

Le coordinate geodetiche polari costituiscono un sistema di coordinate definito su una varietà di Riemann. In questo sistema, si sceglie un punto di origine sulla varietà e, a partire da esso, si rappresentano gli altri punti della varietà attraverso due parametri: la distanza geodetica dal polo e l'angolo rispetto a una direzione di riferimento definita localmente. Questo approccio permette di estendere il concetto di coordinate polari standard a superfici curve e geometrie più complesse.

Formalmente, dato un punto di origine p sulla varietà \mathcal{M} , ogni punto $q \in \mathcal{M}$ sufficientemente vicino a p può essere rappresentato attraverso le coordinate geodetiche polari (ρ, θ) , dove ρ indica la distanza geodetica tra p e q , mentre θ è l'angolo rispetto a una direzione di riferimento fissata presso il punto p .

1.5.1 Discretizzazione

Nell'ambito discreto, la superficie \mathcal{X} viene campionata in N punti $\{x_1, \dots, x_N\}$. Su questi punti si costruisce una *mesh triangolare* (V, E, F) , dove:

- $V = \{1, \dots, N\}$ è l'insieme dei vertici della mesh;
- E è l'insieme degli spigoli che collegano coppie di vertici;
- F è l'insieme delle facce triangolari definite da terne di vertici.

Ogni spigolo interno $ij \in E$ appartiene esattamente a due facce triangolari $ikj \in F$ e $jhi \in F$, mentre gli spigoli al bordo appartengono a una sola faccia triangolare.

Per ogni vertice $i \in V$, l'insieme dei vertici connessi direttamente a i tramite uno spigolo è detto *1-ring* di i ed è definito come:

$$1\text{-ring}(i) = \{j \in V \mid ij \in E\}.$$

Questa rappresentazione discreta consente di approssimare le proprietà geometriche e topologiche della superficie continua \mathcal{X} attraverso la mesh, mantenendo una stretta corrispondenza con la struttura originale.

La qualità della discretizzazione, e quindi l'accuratezza del sistema di coordinate geodetiche polari discreto, dipende dalla densità del campionamento e dalla regolarità della triangolazione. Una mesh con triangoli il più possibile equilateri e una distribuzione uniforme dei vertici garantisce una migliore approssimazione delle proprietà geometriche[9] della varietà sottostante.

1.6 Coordinate Baricentriche

In geometria, un sistema di coordinate baricentriche è un sistema in cui la posizione di un punto viene specificata rispetto a un *simpletso*³.

A differenza del sistema di coordinate cartesiane, in cui la posizione di un punto nel piano viene specificata attraverso due valori (x, y) che rappresentano le distanze lungo gli assi ortogonali X e Y a partire da un punto di origine comune, nelle coordinate baricentriche un punto viene espresso come una combinazione lineare dei vertici di un simpletso, ad esempio un triangolo nel piano bidimensionale. In particolare, un punto P nel piano può essere rappresentato attraverso tre coefficienti $(\lambda_1, \lambda_2, \lambda_3)$ associati ai vertici A, B e C di un triangolo scelto, tali che:

$$P = \lambda_1 A + \lambda_2 B + \lambda_3 C,$$

con la condizione che $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Questo sistema di coordinate è equivalente a quello cartesiano, in quanto permette di individuare qualsiasi punto nel piano. La differenza è che invece di definire come punto di riferimento un punto d'origine e due assi, si utilizza un triangolo (e i suoi vertici).

Le coordinate baricentriche sono utili per determinare la posizione relativa di un punto rispetto al simpletso. Se tutte le coordinate baricentriche sono positive e

³Generalizzazione di un triangolo a dimensioni arbitrarie. Ad esempio, un 0-simpletso è un punto, un 1-simpletso è un segmento, un 2-simpletso è un triangolo e un 3-simpletso è un tetraedro.

comprese tra 0 e 1, il punto si trova all'interno del semplice. Se una o più coordinate sono negative o maggiori di 1, il punto si trova al di fuori del semplice.

Ogni punto nel piano ha coordinate baricentriche^{1,2}, e la loro somma non è mai zero.

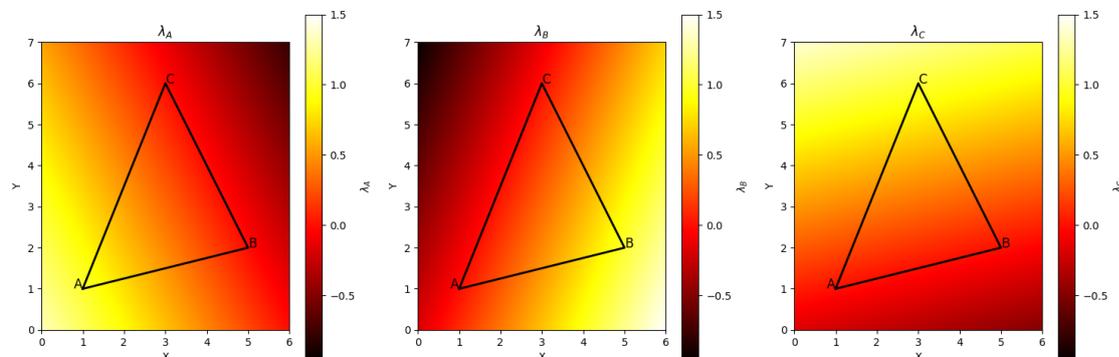


Figura 1.2: Rappresentazione delle coordinate baricentriche nel piano. Ogni immagine mostra la variazione di una delle tre coordinate baricentriche. Per ogni punto nel piano, è visualizzato il valore della coordinata baricentrica associata a ciascun vertice del triangolo ABC preso come punto di riferimento. All'interno del triangolo, le coordinate variano tra 0 e 1. Al di fuori del triangolo, la coordinata aumenta oltre 1 in direzione del vertice corrispondente alla coordinata e scende sotto 0 nella direzione opposta.

1.7 Reti Neurali convoluzionali

1.7.1 Convoluzioni

La convoluzione è un'operazione matematica che combina due funzioni per produrne una terza. Nel contesto dell'elaborazione di segnali e immagini, una delle funzioni rappresenta il segnale di input, mentre l'altra, detta kernel o filtro, definisce come l'input debba essere trasformato. Nel caso discreto bidimensionale, come nelle immagini digitali, la convoluzione produce un nuovo valore per ogni pixel combinando i valori dei pixel vicini con i coefficienti del kernel.

Formalmente, dato un input f e un kernel k di dimensione $(2a + 1) \times (2b + 1)$, il valore del pixel di coordinate (x, y) nell'immagine di output è:

$$(f \star k)(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b f(x + i, y + j) \cdot k(i, j)$$

dove la sommatoria considera tutti i pixel nel vicinato centrato in (x, y) e li moltiplica per i corrispondenti valori del kernel. Nell'elaborazione delle immagini, questa

operazione permette di applicare vari effetti come sfocatura, rilevamento dei bordi o enfaticizzazione dei dettagli, a seconda dei valori specificati nel kernel. La convoluzione presenta proprietà fondamentali come la linearità e l'invarianza alla traslazione, che la rendono particolarmente utile nell'estrazione di caratteristiche locali.

1.7.2 Reti Neurali Convoluzionali

Le reti neurali convoluzionali (Convolutional Neural Networks, CNN) rappresentano una specializzazione delle reti neurali tradizionali, ottimizzate per l'elaborazione di dati che presentano una chiara struttura topologica, come le immagini. La loro peculiarità risiede nell'utilizzo di strati di convoluzione che, a differenza degli strati completamente connessi, applicano un insieme di filtri convoluzionali localmente sulla struttura di input. Gli strati convoluzionali operano facendo scorrere i filtri (o kernel) attraverso l'input, producendo mappe di attivazione che preservano le relazioni spaziali dei dati originali. Questa caratteristica è fondamentale poiché:

- mantiene l'informazione sulla località spaziale degli input
- garantisce l'invarianza traslazionale⁴ delle feature estratte
- riduce significativamente il numero di parametri rispetto a una rete completamente connessa

Nelle implementazioni classiche, le CNN operano su griglie regolari bidimensionali, dove la definizione di convoluzione è naturale e ben definita. I filtri convoluzionali scorrono sull'immagine applicando la stessa operazione in ogni posizione, sfruttando la regolarità della struttura sottostante. Questa regolarità della griglia permette di definire in modo non ambiguo concetti come il vicinato di un pixel e le direzioni di scorrimento del filtro.

Tuttavia, questa formulazione standard presenta delle limitazioni quando si vuole operare su domini più complessi, come le superfici tridimensionali, dove la struttura regolare della griglia viene meno. Questo limite motiva l'estensione del concetto di convoluzione a domini più generali, come le varietà differenziabili, dove la definizione di vicinato e di operazione di convoluzione richiede una formulazione più sofisticata.

1.7.3 Convoluzioni Geodetiche

Le convoluzioni geodetiche generalizzano le convoluzioni classiche alle varietà di Riemann. Mentre nelle convoluzioni classiche si opera su griglie regolari di pixel (ad esempio immagini 2D), le convoluzioni geodetiche permettono di considerare la

⁴Capacità di riconoscere una caratteristica (feature) indipendentemente dalla sua posizione nell'immagine

struttura geometrica del manifold, utilizzando coordinate locali costruite attorno a ciascun punto.

Patch operator

Per definire la convoluzione geodetica, consideriamo un intorno circolare $B_{\rho_0}(x)$ centrato in un punto x del manifold, di raggio ρ_0 . Utilizziamo un sistema di coordinate geodetiche polari (ρ, θ) per descrivere i punti di $B_{\rho_0}(x)$, dove ρ rappresenta la distanza geodetica da x e θ è l'angolo rispetto a un asse di riferimento arbitrario. Sia $\Omega(x) : B_{\rho_0}(x) \rightarrow [0, \rho_0] \times [0, 2\pi]$ una mappa biettiva che associa a ciascun punto le sue coordinate geodetiche polari.

Definiamo ora il *patch operator* $D(x)f$ come una funzione che, dato un punto (ρ, θ) nelle coordinate locali, restituisce il valore della funzione f definita sul manifold nel punto corrispondente:

$$(D(x)f)(\rho, \theta) = f \circ \Omega^{-1}(x)(\rho, \theta),$$

dove $\Omega^{-1}(x)$ è la mappa inversa che riporta dal sistema di coordinate locali al manifold.

Convoluzione Geodetica

La convoluzione geodetica tra una funzione f definita sul manifold e un filtro a si esprime come:

$$(f \star a)(x) = \sum_{\theta, r} a(\theta + \Delta\theta, \rho) \cdot (D(x)f)(\rho, \theta)$$

dove:

- $a(\theta, \rho)$ è un filtro definito sulle coordinate locali (θ, ρ) ;
- $(D(x)f)(\rho, \theta)$ è il valore della funzione f interpolato tramite il patch operator;
- $\Delta\theta$ rappresenta una rotazione arbitraria del filtro, introdotta per gestire l'ambiguità angolare.

In analogia alle convoluzioni classiche, questa operazione combina i valori della funzione f in un intorno locale con i pesi del filtro a , tenendo conto della struttura geometrica del manifold. Tuttavia, anziché considerare pixel disposti su una griglia regolare, il filtro agisce su punti definiti in coordinate geodetiche polari. Un insieme di punti all'interno di un sistema di coordinate polari geodetiche è detto *patch*.

Nel caso delle convoluzioni geodetiche, il kernel è definito come una funzione continua $a(\theta, r)$ su coordinate polari, e i punti della patch sono determinati dalle coordinate (ρ, θ) nel sistema polare geodetico.

Mentre nelle immagini digitali la struttura regolare delle griglie facilita il calcolo, nei manifold si deve ricorrere a interpolazioni per calcolare il valore di f in punti arbitrari. La scelta del raggio ρ_0 è cruciale: un valore troppo grande potrebbe compromettere la validità topologica del disco geodetico, mentre un valore troppo piccolo limiterebbe l'efficacia del filtro.

Discretizzazione

Per implementare le convoluzioni geodetiche su mesh triangolari, è necessario discretizzare il *patch operator* e il dominio locale in coordinate polari geodetiche. Questo approccio si basa su una suddivisione locale del manifold attorno a ciascun vertice utilizzando N_θ bin angolari e N_ρ bin radiali.

Costruzione delle Coordinate Geodetiche Polari Per ogni vertice i della mesh triangolare, consideriamo il suo *1-ring*, ovvero l'insieme di vertici adiacenti connessi a i tramite un lato. Questo intorno locale viene diviso in N_θ bin equiangolari, a partire da un raggio iniziale allineato con uno degli spigoli del triangolo. Successivamente, questi raggi vengono propagati attraverso i triangoli adiacenti utilizzando un procedimento di *unfolding*, formando le linee angolari che delimitano ciascun bin angolare.

La suddivisione in bin radiali si realizza mediante N_ρ insiemi di livello equidistanti della funzione distanza geodetica, centrata nel vertice i . Formalmente, dato l'intervallo $[0, \rho_0]$, con ρ_0 raggio del disco geodetico, si definisce ciascun bin come $\{x' : d_X(x, x') = \rho\}$, dove i valori di ρ sono equispaziati di $\frac{\rho_0}{N_\rho}$, in modo da garantire una suddivisione regolare e consistente del dominio. La distanza può essere calcolata utilizzando algoritmi efficienti come il *fast marching method*[13], sfruttando l'esecuzione parallela di più processi[23], basati sulla propagazione di insiemi di livello[5], o persino con approcci basati sul deep learning[20].

Rappresentazione del Patch Operator Il *patch operator* discreto può essere rappresentato come una matrice $N_\theta N_\rho N \times N$, dove:

- N_θ è il numero di bin angolari;
- N_ρ è il numero di bin radiali;
- N è il numero totale di vertici della mesh.

Questa matrice agisce sostanzialmente come una funzione che, dato un vertice i , mappa i suoi vicini (definiti in coordinate geodetiche polari) sui valori della funzione f discreta definita sui vertici della mesh. Essendo i valori di f rilevanti solo per i vertici vicini a i , la matrice è altamente sparsa: la maggior parte degli elementi è

nulla, e i valori significativi si concentrano nelle righe corrispondenti ai bin angolari e radiali del vertice i .

1.7.4 Reti Convoluzionali Geodetiche

Le reti convoluzionali geodetiche rappresentano un'estensione delle reti neurali convoluzionali tradizionali (CNN) a domini tridimensionali e, più in generale, a manifold. Queste reti mantengono le componenti fondamentali delle CNN classiche: l'informazione posizionale, la feature map e il kernel. Tuttavia, il loro funzionamento si adatta alla geometria irregolare del dominio di input. Di seguito si propone un confronto tra le convoluzioni classiche e quelle geodetiche, seguito da un'illustrazione di alcune applicazioni pratiche.

Confronto tra Convoluzioni Classiche e Geodetiche

- **Informazione posizionale:**

- Nelle convoluzioni classiche, l'informazione posizionale è fornita dalle coordinate regolari della griglia (x, y) , corrispondenti ai pixel dell'immagine.
- Nelle convoluzioni geodetiche, l'informazione posizionale è derivata dalle coordinate baricentriche dei punti appartenenti alla "patch" di una superficie tridimensionale. Queste coordinate sono combinate con l'informazione dei vertici del triangolo contenente il punto.

- **Feature map:**

- Nelle CNN classiche, la feature map contiene valori scalari (ad esempio, intensità dei pixel) o vettoriali (ad esempio, canali RGB).
- Nelle CNN geodetiche, la feature map rappresenta funzioni definite sui punti del manifold, spesso costruite sulla base di attributi geometrici o fisici della superficie.

- **Kernel:**

- Nelle convoluzioni classiche, il kernel è una matrice bidimensionale, con ciascun elemento che rappresenta un peso applicato ai pixel della griglia.
- Nelle convoluzioni geodetiche, il kernel è una funzione definita sulle coordinate locali del manifold (tipicamente in un sistema polare geodetico). La struttura del kernel rimane analoga, ma la sua applicazione richiede interpolazioni locali per gestire la geometria irregolare.

- **Dominio di applicazione:**

- Le convoluzioni classiche operano su griglie regolari, come immagini o dati volumetrici regolari.
- Le convoluzioni geodetiche sono progettate per domini irregolari, come superfici 3D rappresentate da mesh triangolari o varietà differenziabili.

Applicazioni delle Reti Convoluzionali Geodetiche

Le reti convoluzionali geodetiche trovano applicazione in contesti che richiedono l'analisi di superfici tridimensionali e geometrie complesse. Grazie alla capacità di sfruttare simmetrie e regolarità geometriche nei dati, rappresentano una naturale estensione delle reti convoluzionali classiche, adattandosi a strutture non euclidee e manifold tridimensionali. Questo tipo di modello è incluso nel cosiddetto panorama del *Geometric Deep Learning*[4], di cui fanno parte svariati metodi di studio di geometrie complesse.

Esistono anche approcci alternativi per l'apprendimento su superfici 3D[3] e la generazione di geometrie tridimensionali euclidee. Ad esempio, AtlasNet utilizza elementi parametrici per generare superfici 3D con alta precisione e flessibilità, applicabili alla ricostruzione da immagini 2D e al morphing[11]. Un altro approccio è MeshSDF, che adotta rappresentazioni mesh differenziabili per applicazioni come la ricostruzione da immagini singole e l'ottimizzazione di forme con vincoli fisici[22].

Un'applicazione rilevante delle reti geodetiche è l'ottimizzazione aerodinamica[1], cruciale per il design industriale, così come la modellazione di fenomeni di trasferimento termico in strutture macro-porose. Queste reti possono essere utilizzate per predire distribuzioni di temperatura, pressione e velocità in materiali con cambiamento di fase[15], fornendo un metodo efficace per ottimizzare unità porose tridimensionali ad alta densità di potenza in base a requisiti specifici.

1.8 Il formato STL

Il formato STL (STereoLithography)⁵ è un formato di file ampiamente utilizzato per rappresentare geometrie tridimensionali attraverso mesh triangolari[8]. Un file STL descrive la superficie di un oggetto tridimensionale come un insieme non strutturato di facce triangolari orientate. Il formato può essere rappresentato sia in formato ASCII che binario, dove la versione binaria risulta più compatta e quindi più efficiente in termini di spazio di archiviazione. Tuttavia, entrambe le versioni

⁵La stereolitografia è una tecnica di stampa 3D che realizza oggetti attraverso la solidificazione di resine fotosensibili mediante raggi UV, partendo da modelli digitali CAD.

contengono le stesse informazioni geometriche essenziali. È importante notare che il formato STL presenta alcune limitazioni intrinseche:

- non memorizza informazioni su colori, texture o materiali
- non mantiene esplicitamente le relazioni di adiacenza tra i triangoli
- può contenere difetti topologici come triangoli duplicati o non connessi

Altri formati come il *.ply* permettono di memorizzare una descrizione più ricca dell'oggetto, includendo non solo la lista di poligoni che ne compongono la superficie, ma anche proprietà aggiuntive come il colore, la trasparenza, le normali alle superfici, le coordinate delle texture e valori di confidenza dei dati, con la possibilità di specificare proprietà differenti per il fronte e il retro di ogni poligono.

Nell'ambito di questa tesi, i file STL rappresentano il punto di partenza per la creazione del dataset, fornendo una descrizione geometrica delle superfici tridimensionali su cui viene applicato il processo di pre-processing. I dati contenuti nei file STL vengono successivamente trasformati per essere compatibili con i sistemi di coordinate geodetiche polari descritti nelle sezioni precedenti.

Capitolo 2

Dataset e Preprocessing

2.1 Descrizione del Dataset

Il dataset utilizzato per l'addestramento e la valutazione del modello è di dimensioni relativamente ridotte, composto da 58 elementi, a cui si aggiungono alcuni ulteriori campioni utilizzati esclusivamente per l'*evaluation*.

Le geometrie dei pistoni originali, fornite da NAIS, sono modelli tridimensionali ad alta risoluzione. Per l'utilizzo nel dataset, è stato ridotto il numero di facce delle mesh, al fine di ottenere un numero più gestibile di triangoli e assicurando che tutte le mesh avessero il medesimo numero di vertici, per una questione di omogeneità dei dati nel dataset.

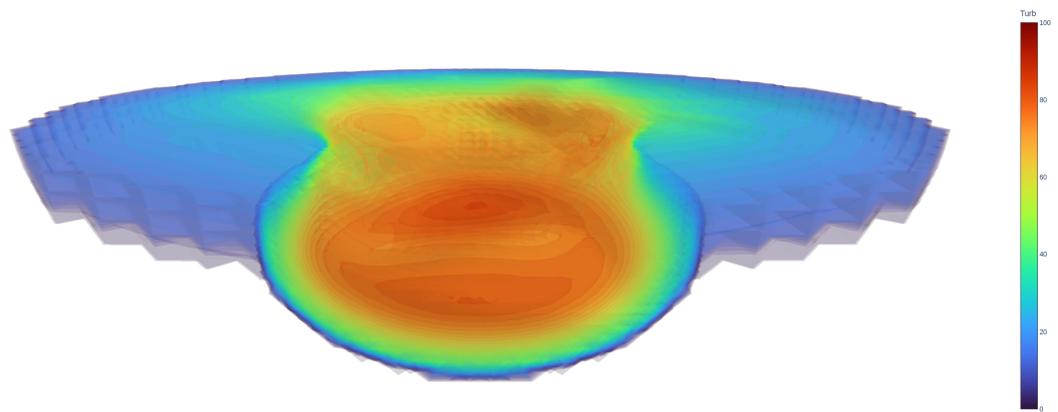
Le geometrie semplificate sono state sottoposte a simulazioni numeriche di fluidodinamica computazionale (CFD, *Computational Fluid Dynamics*). Tali simulazioni, effettuate dall'azienda, hanno fornito, per ogni punto all'interno del pistone, i valori dei parametri fisici di interesse. I valori ottenuti dalle simulazioni sono considerati come *ground truth* per il modello.

I parametri di interesse considerati sono tre:

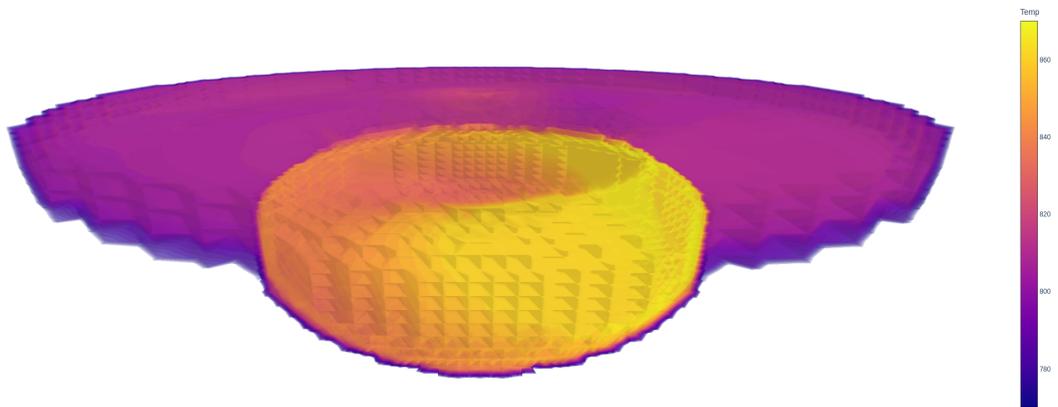
- la temperatura;
- la turbolenza;
- il rapporto di ricircolo dei gas di scarico (EGR, *Exhaust Gas Recirculation*).

Per ciascun parametro è presente un set distinto di *ground truth*, ciascuno dei quali è un array tridimensionale di dimensione $64 \times 64 \times 64$, tramite il quale sono state

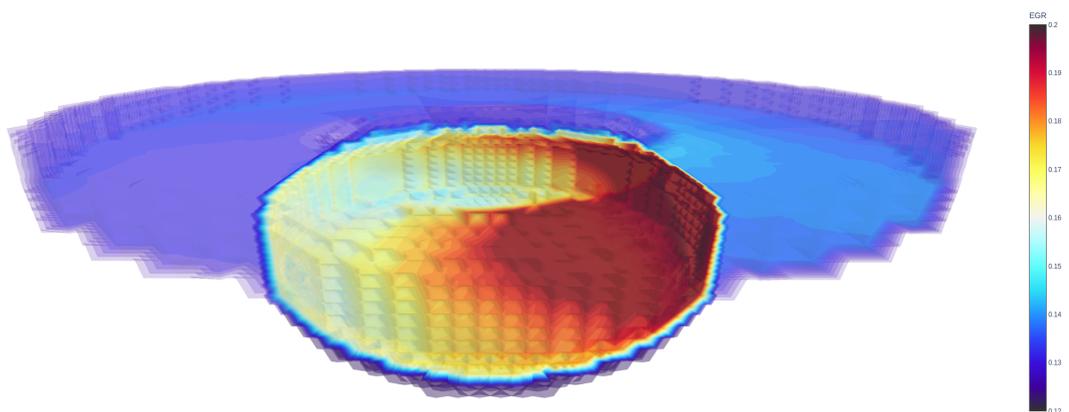
addestrate tre reti neurali differenti, ciascuna con la medesima architettura. Gli input, invece, sono identici per tutte e tre le reti.



(a) Turbolenza



(b) Temperatura



(c) EGR

Figura 2.1: Ground truth del modello

Il modello, data una geometria in input, è progettato per predire, per ogni punto dello spazio tridimensionale, il valore del parametro corrispondente, replicando i risultati che sarebbero altrimenti ottenuti tramite simulazioni CFD, ma con un costo computazionale significativamente inferiore.

2.1.1 Preprocessing degli Input

Gli input per il modello derivano da file in formato STL che rappresentano la geometria tridimensionale del pistone. Il preprocessing applicato a queste geometrie produce, per ogni segmento di ogni raggio di ogni patch associata alla geometria, un tensore che contiene:

- le coordinate baricentriche (3 numeri reali per ogni punto),
- gli indici dei vertici del triangolo contenente il punto (3 numeri interi appartenenti all'insieme dei vertici della geometria).

Il risultato del preprocessing è quindi una matrice multidimensionale di forma $N \times N_{\text{radial}} \times N_{\text{angular}} \times 3 \times 2$, dove:

- N rappresenta il numero di vertici della mesh triangolare,
- N_{radial} è il numero di cerchi concentrici considerati nella patch associata a ciascun vertice,
- N_{angular} è il numero di raggi angularmente equidistanti (calcolati rispetto al polo della patch),
- 3×2 rappresenta le coordinate baricentriche e gli indici del triangolo associato al punto.

Le geometrie utilizzate sono rappresentate da mesh triangolari, ognuna delle quali contiene 752 vertici. Sebbene una rete convoluzionale geodetica possa tecnicamente gestire geometrie con un numero arbitrario di vertici (grazie a modifiche introdotte durante il lavoro), per garantire omogeneità nei dati durante l'addestramento è stato deciso di utilizzare geometrie con un numero fisso di vertici. In particolare, tutte le geometrie del dataset sono state normalizzate a 752 vertici.

L'input effettivo della rete neurale non è il file STL contenente la geometria, ma bensì l'insieme delle coordinate baricentriche polari che compongono le *patch*, su cui poi il modello effettuerà le convoluzioni, e l'insieme dei segnali, uno per ogni punto indicato nelle *patch*.

2.2 Calcolo delle Coordinate Geodetiche Polari su Spazio Discreto

In questa sezione si introduce l'algoritmo DGPC[18] (*Discrete Geodesic Polar Coordinates*), un metodo per calcolare un sistema locale di coordinate polari geodetiche su una mesh triangolare \mathcal{T} , definita in uno spazio tridimensionale.

2.2.1 Definizione del Problema

Sia \mathcal{T} una mesh triangolare con vertici $\{v_i\}_{i=1}^n$ e triangoli $T_{ijk} = [v_i, v_j, v_k]$. Si consideri un punto di origine $s \in \mathcal{T}$ e un vettore tangente \mathbf{x} , che funge da riferimento per gli angoli del sistema di coordinate polari. La mappa polare geodetica (GPM) $P_s : \mathcal{T} \rightarrow \mathbb{R}^2$ associa a ciascun punto $v \in \mathcal{T}$ una coppia (ρ, θ) , dove:

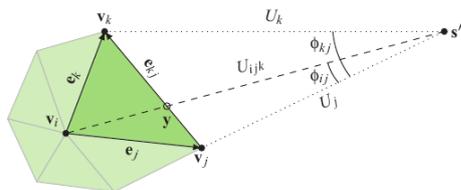
- $\rho(v)$ è la distanza geodetica tra v e s ;
- $\theta(v)$ è l'angolo tra la curva geodetica che collega v e s e la direzione \mathbf{x} .

La GPM è un'isometria radiale, il che implica che le distanze rispetto al punto di origine s sono preservate. Tuttavia, su geometrie non piane, gli angoli possono subire distorsioni.

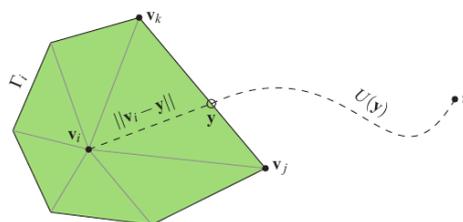
2.2.2 Approssimazione delle Coordinate Geodetiche Polari

Il calcolo esatto delle coordinate polari geodetiche su un'intera mesh è computazionalmente oneroso. L'algoritmo DGPC affronta questo problema mediante un'approssimazione iterativa basata sulla programmazione dinamica, simile all'algoritmo di Dijkstra. La differenza fondamentale risiede nel fatto che, mentre Dijkstra calcola le distanze una sola volta, DGPC aggiorna iterativamente le soluzioni fino al raggiungimento di una precisione definita.

Algoritmo per l'Approssimazione delle Distanze Geodetiche



(a) L'algoritmo estende il piano del triangolo $v_i v_j v_k$ e crea un punto d'origine fittizio sul piano, dopodiché proietta le distanze ed approssima progressivamente la distanza geodetica e l'angolo tra v_i ed il punto d'origine fittizio.



(b) La distanza viene approssimata a partire da un punto y appartenente al perimetro dell'1-ring.

Figura 2.2: Illustrazione delle meccaniche di base dell'algoritmo DGPC. Immagini tratte da [18]

Dato il punto di origine s e un vertice v_i , consideriamo il perimetro dell'1-ring di v_i , indicato con Γ_i . La distanza approssimata U_i soddisfa la seguente equazione:

$$U_i = \min_{y \in \Gamma_i} \{U(y) + \|v_i - y\|\},$$

dove $U(y)$ è l'approssimazione della distanza geodetica tra s e un punto y appartenente all'1-ring di v_i .

L'algoritmo utilizza una lista di candidati che viene inizializzata al solo punto d'origine, ed un array in cui sono contenute le distanze dal punto d'origine a ciascun vertice della geometria, inizializzate ad un valore molto grande, e procede iterativamente:

1. Rimuovi dalla lista il candidato con distanza approssimata minore
2. Per ogni vertice connesso tramite un lato al candidato, calcolane la distanza e salvala in un array
3. Se la distanza calcolata migliora oltre una certa tolleranza rispetto all'ultima distanza calcolata, aggiungi il vertice alla lista dei candidati

L'algoritmo calcola il sistema di coordinate geodetiche polari di tutti i vertici entro un certo raggio, detto U_{\max} , ignorando i punti la cui distanza calcolata supera questa soglia

L'implementazione scelta dell'algoritmo per questo lavoro è tratta dal progetto open-source *GeoConv*[16]

Estensione per il Calcolo degli Angoli Polari

L'algoritmo DGPC non si limita all'approssimazione delle distanze geodetiche ma include anche il calcolo degli angoli.

Il calcolo degli angoli polari si basa su un principio di correttezza nel caso planare. Per ogni vertice v_i , l'angolo θ_i , si utilizza un'interpolazione lineare degli angoli:

$$\theta_i = (1 - \alpha)\theta_j + \alpha\theta_k$$

dove $\alpha = \phi_{ij}/\phi_{kj}$, con ϕ_{ij} che rappresenta l'angolo tra i vettori $v_i - s$ e $v_j - s$. Questa approssimazione risulta esatta per mesh planari e fornisce una buona approssimazione per mesh con curvatura moderata. In particolare, se la mesh è localmente isometrica al piano, l'approssimazione coincide con le coordinate polari geodetiche esatte.

2.3 Preprocessing da File STL a Coordinate Baricentriche

Per preparare i dati destinati alla rete neurale convoluzionale geodetica, è stato utilizzato un programma che esegue il preprocessing delle geometrie tridimensionali contenute nei file STL. L'obiettivo è trasformare queste geometrie in una matrice di coordinate baricentriche, che costituirà l'informazione posizionale della rete, che verrà utilizzata dalla rete per estrarre le patch, sulle quali effettuare le operazioni di convoluzione.

2.3.1 Normalizzazione delle Geometrie

Per facilitare l'analisi dei dati da parte del modello, il programma inizia normalizzando tutte le geometrie, in modo da omogenizzare le misure. Si calcola il *diametro geodetico*¹ della geometria, utilizzato per scalare la mesh affinché tutte le distanze, inclusa la più grande, siano comprese tra 0 e 1. Inoltre, i vertici vengono traslati in modo che la geometria abbia al centro l'origine degli assi cartesiani.

2.3.2 Calcolo del Raggio Geodetico

Successivamente, viene calcolato il *raggio geodetico* (ossia, il raggio del disco geodetico dentro il quale sono definite le coordinate geodetiche polari), necessario per determinare l'estensione del sistema di coordinate geodetiche polari attorno a ogni vertice della mesh. Il raggio geodetico è definito come il lato più lungo della geometria. Questo parametro è fondamentale poiché delimita l'area entro la quale verrà calcolato il sistema di coordinate geodetiche polari per ciascun vertice.

¹Con diametro geodetico si intende la massima distanza geodetica misurabile tra due punti qualsiasi sulla superficie.

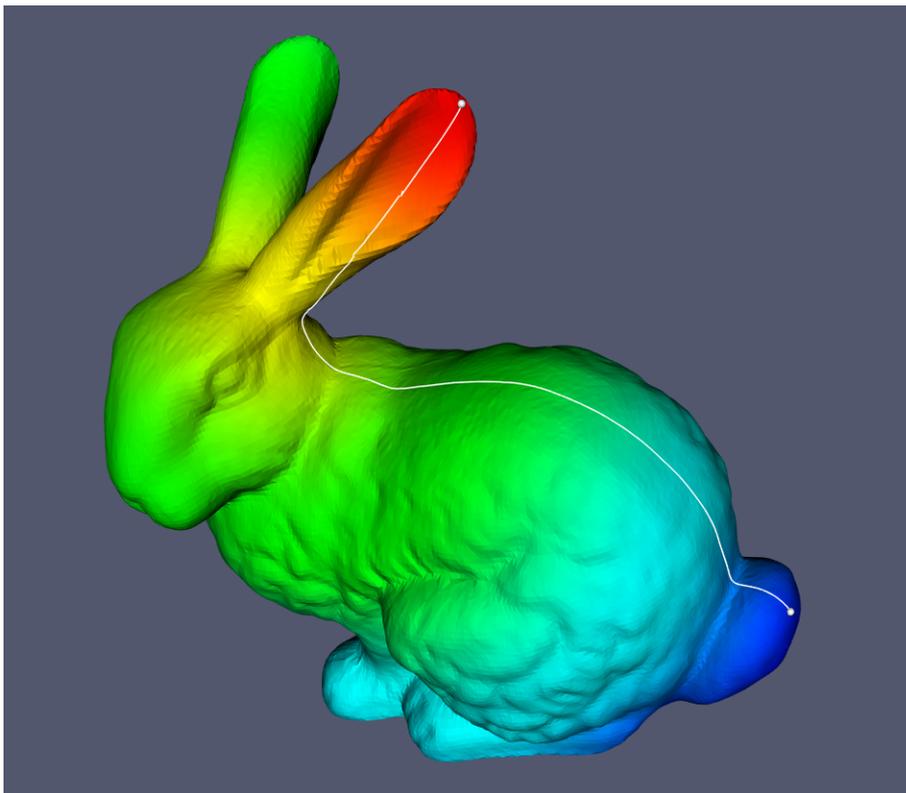


Figura 2.3: Diametro geodetico dello Stanford Bunny. Calcolato con PYGEODESIC. La mesh è colorata in base alla distanza dal punto selezionato sulla coda.

2.3.3 Calcolo delle Coordinate Baricentriche

La funzione principale utilizzata per il calcolo delle coordinate baricentriche applica il metodo delle coordinate affini² elaborando i sistemi di coordinate polari geodetiche (GPC-systems) precedentemente calcolati e restituisce le coordinate necessarie per le convoluzioni della rete.

Descrizione della Funzione `compute_barycentric_coordinates` La funzione `compute_barycentric_coordinates` prende in input:

- `gpc_systems`: Il gruppo di GPC-systems calcolati per la mesh.
- `n_radial`: Il numero di coordinate radiali desiderate per il template (ad esempio, 10).
- `n_angular`: Il numero di coordinate angolari desiderate per il template (ad esempio, 15).

²*Barycentric Technique*, descritta in <https://blackpawn.com/texts/pointinpoly/>

- **radius:** Il raggio del template utilizzato per delimitare l'area attorno a ogni vertice.

Formato e Dimensioni dell'Output La funzione restituisce un array a 5 dimensioni contenente le coordinate baricentriche per ogni punto del template e per ogni GPC-system. La struttura dell'array è $[N, R, A, 3, 2]$, dove:

- N è il numero di GPC-systems calcolati (uno per ciascun vertice della mesh).
- R è il numero di coordinate radiali del template (`n_radial`).
- A è il numero di coordinate angolari del template (`n_angular`).
- Il valore 3 rappresenta i tre vertici del triangolo su cui ricade ogni punto del template.
- Il valore 2 distingue tra indici dei vertici e coordinate baricentriche.

Quindi, se per esempio si volesse conoscere in quale triangolo è situato il punto con coordinate polari (r, α) del sistema con punto di origine p , si potrebbe accedere a `array[p, r, alpha, :, 0]`, ottenendo così gli indici dei vertici in cui tal punto è contenuto.

Se invece si volessero conoscere le coordinate baricentriche dello stesso punto, in modo da identificare la sua posizione all'interno del triangolo, si potrebbe accedere a `array[p, r, alpha, :, 1]`

Utilizzo delle Coordinate Baricentriche Le coordinate baricentriche calcolate rappresentano le *patch* su cui la rete convoluzionale eseguirà le operazioni di convoluzione. Il *segnale* associato a ciascuna patch è costituito dalle coordinate tridimensionali dei punti della mesh. Utilizzando le coordinate baricentriche e gli indici dei vertici (e le loro coordinate tridimensionali), è possibile interpolare le coordinate spaziali precise dei punti della patch all'interno dei triangoli della mesh.

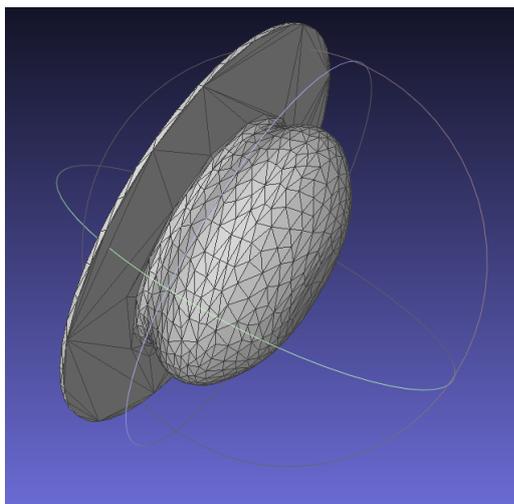
2.3.4 Problema con La Dimensione del Raggio U_{\max}

Per raggio U_{\max} si intende la distanza massima entro la quale vengono calcolate le coordinate geodetiche polari attorno a un vertice della mesh. In altre parole, U_{\max} delimita l'area di influenza di ciascun vertice, definendo un disco geodetico all'interno del quale vengono considerate le distanze e gli angoli per il calcolo delle coordinate polari.

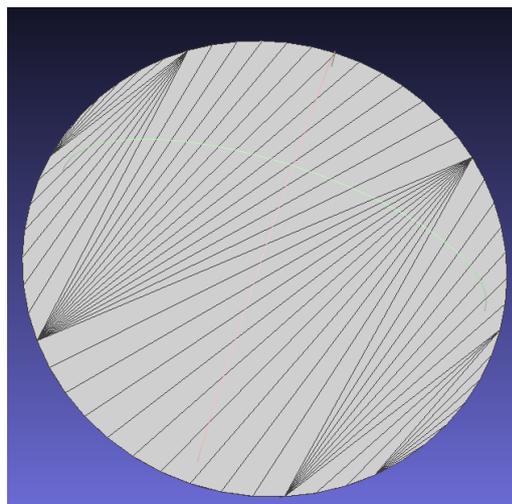
Le geometrie presenti nel dataset derivano da modelli tridimensionali complessi, successivamente semplificati utilizzando la libreria Python `trimesh`. Tale libreria offre

una funzione specifica per ridurre il numero di triangoli nella mesh fino a un valore desiderato, privilegiando la semplificazione nelle regioni più piatte. Questo approccio consente di preservare le informazioni relative alla curvatura della geometria, evitando di comprometterne la rappresentazione.

Tuttavia, questa caratteristica dell'algoritmo di `trimesh`, pur essendo generalmente appropriata, si scontra con una peculiarità delle geometrie presenti nel dataset utilizzato. Ogni elemento del dataset rappresenta la testa di un pistone, che nella parte inferiore è stato “chiuso” artificialmente con uno strato orizzontale nella mesh, al fine di completare la geometria per il calcolo. Ne consegue che la maggior parte degli elementi del dataset presenta 2.4 due caratteristiche predominanti:



(a) Una forma curva e simmetrica, simile a un solido di rotazione, nella parte superiore;



(b) Una superficie piatta molto larga nella parte inferiore

Figura 2.4: Le due caratteristiche predominanti nelle geometrie del dataset, dal punto di vista della lunghezza dei lati, che è altamente disomogenea

Queste particolarità geometriche hanno contribuito alla formazione di triangoli con dimensioni estremamente disomogenee: i lati più lunghi della mesh raggiungono lunghezze di circa 100 unità, mentre la media e la mediana si aggirano attorno a valori pari a 5 unità. Tale disparità è evidenziata da numerosi *outlier* nella distribuzione delle lunghezze dei lati.2.5

Il problema principale si manifesta durante il preprocessing delle geometrie. In questa fase, l'algoritmo utilizza il lato più lungo della mesh come punto di riferimento per il raggio del sistema di coordinate polari geodetiche associato a ciascun vertice. Questo comporta la generazione di sistemi di grandi dimensioni, che, in molte regioni della mesh, includono un numero elevato di vertici. Ciò aumenta significativamente

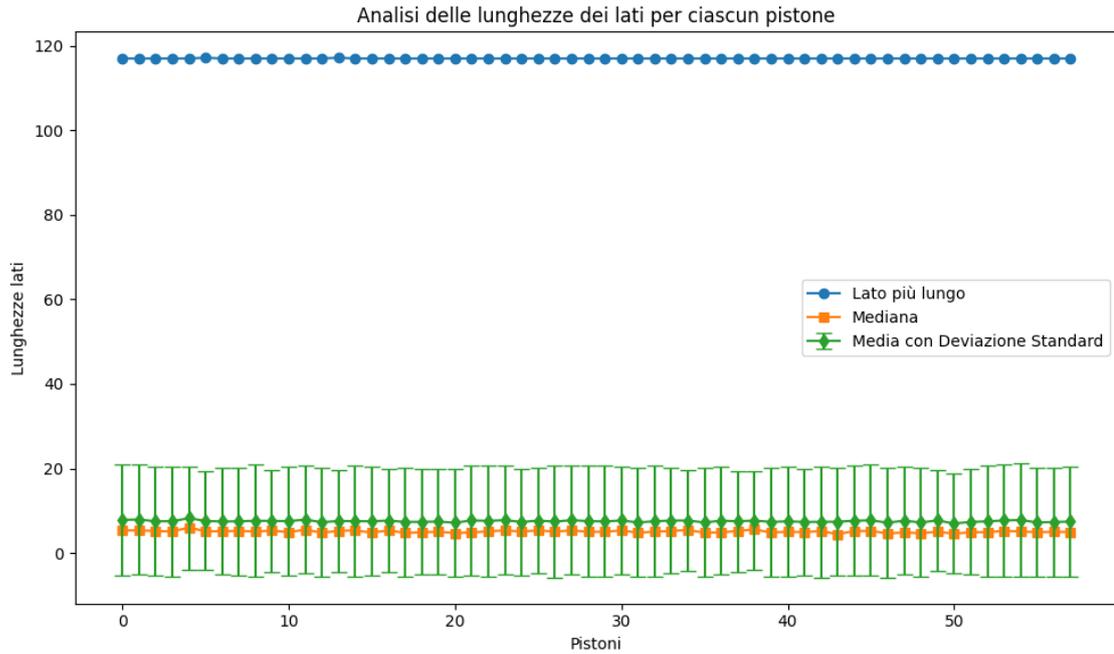


Figura 2.5: La lunghezza dei lati delle geometrie si aggira attorno alle 7 unità. Tuttavia ogni geometria presenta lati di lunghezza pari a quasi 120 unità.

il costo computazionale per il calcolo delle coordinate polari geodetiche associate a ogni *patch*.

2.3.5 Uniformare le Dimensioni dei Triangoli

Per affrontare il problema, è stato sviluppato uno script Python in grado di ridurre la disomogeneità delle dimensioni dei triangoli, accettando un compromesso in termini di precisione nella descrizione delle curve della geometria. Lo script opera in due fasi principali:

1. Inizialmente, la mesh viene ulteriormente semplificata fino a raggiungere un numero di vertici inferiore rispetto a quello desiderato. Questa fase può portare alla formazione di triangoli con dimensioni disomogenee.
2. Successivamente, i triangoli più grandi vengono suddivisi aggiungendo nuovi vertici a metà dei lati più lunghi, procedendo iterativamente fino a ottenere una mesh con un numero desiderato di vertici e triangoli di dimensioni molto più omogenee.

Questo approccio consente di ottenere una mesh con triangoli maggiormente uniformi 2.6, riducendo drasticamente il costo computazionale del preprocessing. Sebbene

questa soluzione comporti una riduzione della precisione geometrica, non è stato osservato alcun impatto significativo sulla qualità delle previsioni effettuate dal modello.

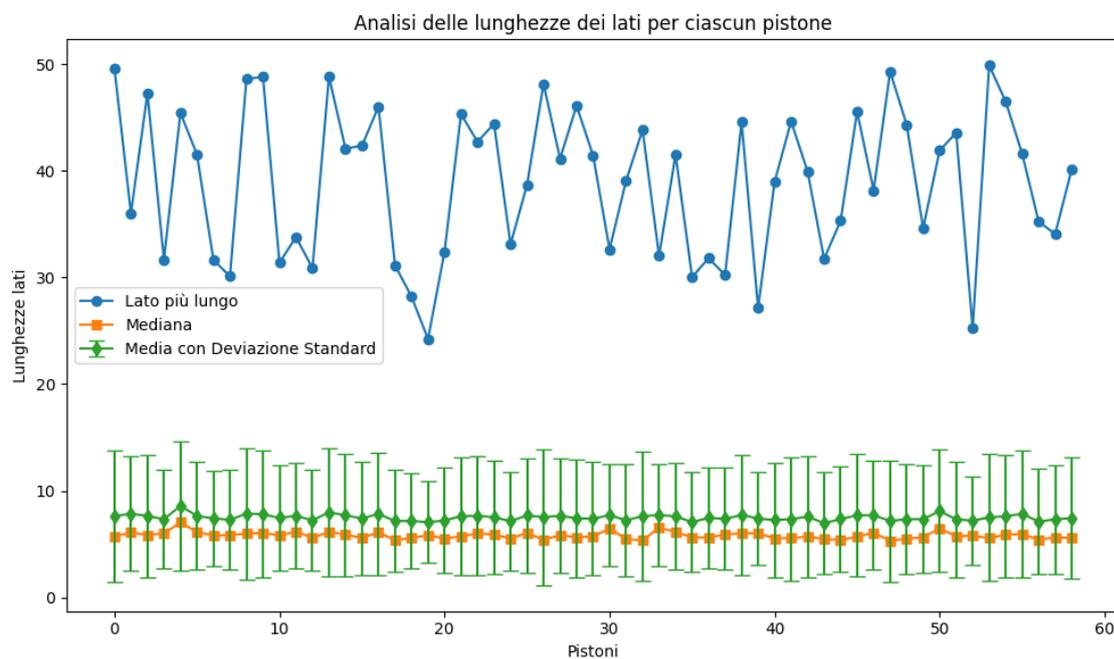


Figura 2.6: A seguito dell’algoritmo di regolarizzazione dei lati, gli outlier sono dimezzati in grandezza.

Capitolo 3

Il modello

3.1 Architettura

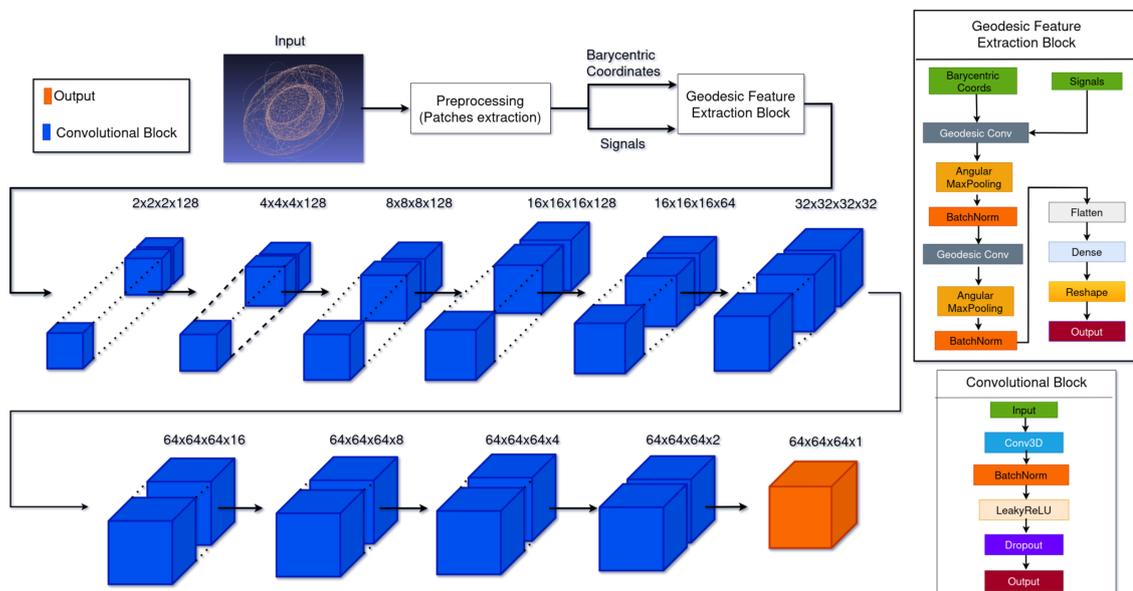


Figura 3.1: Architettura del modello

3.2 Descrizione del Modello

Il modello sviluppato è progettato per elaborare dati su superfici tridimensionali utilizzando convoluzioni geodetiche, con l'obiettivo di catturare informazioni locali rispettando la geometria intrinseca delle superfici.

3.2.1 Pipeline del Modello

Il flusso di elaborazione del modello può essere descritto attraverso i seguenti passaggi fondamentali:

1. Strato di Input:

- *Segnale di Input*: Viene fornito un input di forma $(N, 3)$, dove N rappresenta il numero di vertici della superficie e 3 corrisponde alle coordinate spaziali di ciascun vertice. Ciò costituisce il segnale dell'input.
- *Pseudo-Coordinate Geodetiche*: Si utilizzano le pseudo-coordinate geodetiche locali, necessarie per le convoluzioni geodetiche successive. Ciò costituisce l'informazione posizionale dell'input.

2. Convoluzione Geodetica:

- Si applica uno *strato di convoluzione geodetica* che consente di effettuare convoluzioni direttamente sulla superficie tridimensionale, sfruttando le pseudo-coordinate geodetiche.
- Questo strato estrae le caratteristiche locali iniziali dall'input utilizzando il segnale ricevuto, e ritorna una serie di convoluzioni, ciascuna ruotata di un certo angolo.

3. Batch Normalization e Attivazione:

- Viene applicato uno *strato di Batch Normalization*, che normalizza le attivazioni dello strato precedente, stabilizzando e accelerando il processo di apprendimento.
- Successivamente, si utilizza la funzione di attivazione *Leaky ReLU* con parametro $\alpha = 0,01$, introducendo non linearità nel modello e prevenendo il problema della *morte dei neuroni* associato alla ReLU classica.

4. Dropout:

- Si inserisce uno *strato di Dropout* con un tasso di 0,1 (10%), che omette casualmente alcuni neuroni durante l'addestramento.

5. Angular Max Pooling:

- Lo *strato di Angular Max Pooling* (AMP) aggrega le risposte della convoluzione geodetica provenienti da diverse rotazioni del filtro.
- Per ogni vertice, l'AMP seleziona la risposta massima tra tutte le rotazioni considerate, garantendo l'*invarianza alle rotazioni* del modello.

6. Seconda Convoluzione Geodetica e Angular Max Pooling:

- Si applica un ulteriore *strato di convoluzione geodetica* e *Angular Max Pooling*.

7. Decoder:

- Le caratteristiche processate vengono appiattite tramite uno *strato di Flatten* e passate attraverso uno *strato denso* e attivazione *Leaky ReLU*.
- Si procede poi con una serie di strati di *deconvoluzione tridimensionale* (*Conv3DTranspose*) che ricostruiscono l'output desiderato, trasformando le informazioni apprese in una rappresentazione tridimensionale.

L'architettura descritta è progettata per sfruttare le potenzialità delle *convoluzioni geodetiche* nella gestione di dati definiti su superfici tridimensionali. L'utilizzo sequenziale di strati di convoluzione geodetica e di *Angular Max Pooling* consente al modello di estrarre e aggregare efficacemente le caratteristiche locali, mantenendo l'invarianza alle rotazioni.

L'inserimento di strati di *Batch Normalization* e *Dropout* dopo ogni convoluzione geodetica contribuisce a stabilizzare l'apprendimento e a migliorare la generalizzazione del modello, riducendo il rischio di overfitting. La funzione di attivazione *Leaky ReLU* introduce non linearità, permettendo al modello di apprendere rappresentazioni più complesse.

La fase finale di decodifica trasforma le caratteristiche apprese in un output tridimensionale, grazie all'utilizzo di strati di deconvoluzione tridimensionale. Questa struttura permette al modello di produrre previsioni che preservano la dimensionalità e la natura dei dati originali.

3.3 Strati specifici

3.3.1 Angular Max-Pooling (AMP)

Lo strato di *Angular Max-Pooling* (AMP) è una componente essenziale nelle reti neurali convoluzionali geodetiche (GCNN), progettata per aggregare le risposte delle convoluzioni geodetiche applicate su diverse rotazioni. L'obiettivo principale di questo strato è garantire l'invarianza alle rotazioni del filtro sulla superficie tridimensionale, selezionando per ogni vertice la risposta più rilevante tra quelle generate da tutte le rotazioni considerate.

Funzionamento Data l'arbitrarietà della scelta dell'angolo iniziale, dopo aver applicato la convoluzione geodetica, per ciascun vertice sulla superficie tridimensionale

sono disponibili molteplici risposte, corrispondenti a diverse rotazioni del filtro convoluzionale. Lo strato AMP seleziona la risposta massima per ciascun vertice, secondo la norma euclidea, garantendo che l'informazione più rilevante venga mantenuta indipendentemente dalla rotazione.

Formalmente, dato un vertice x , una feature convoluzionale p , e un insieme di rotazioni $\Delta\theta$, l'output dello strato AMP è dato da:

$$f_p^{\text{out}}(x) = \max_{\Delta\theta} \|f_{\Delta\theta,p}^{\text{in}}(x)\|,$$

dove:

- $p = 1, \dots, Q$ con Q numero di canali della feature map.
- $f_{\Delta\theta,p}^{\text{in}}(x)$ rappresenta il risultato della convoluzione geodetica applicata al vertice x , con una specifica rotazione $\Delta\theta$ e per un determinato filtro p ,
- $\|\cdot\|$ denota la norma euclidea.

In altre parole, per ciascun vertice x , viene considerata la risposta convoluzionale relativa a tutte le rotazioni $\Delta\theta$ e viene selezionata quella di intensità massima.

3.3.2 Convoluzione Geodetica (GC)

Lo strato convoluzionale geodetico (*Geodesic Convolutional Layer*) rappresenta un'estensione delle convoluzioni classiche utilizzate nelle reti neurali convoluzionali, adattata per operare su manifold non euclidei. Questo strato è costruito sulla nozione di convoluzione geodetica, come definita nelle sezioni precedenti, ed è progettato per lavorare direttamente con coordinate polari geodetiche.

Definizione della Convoluzione Geodetica

La convoluzione geodetica (GC) sostituisce lo strato convoluzionale utilizzato nelle reti neurali convoluzionali classiche. A causa dell'ambiguità associata alle coordinate angolari, la convoluzione viene calcolata per tutte le rotazioni del filtro lungo N_θ , ovvero:

$$f_{\Delta\theta,q}^{\text{out}}(x) = \sum_{p=1}^P (f_p \star a_{\Delta\theta,qp})(x), \quad q = 1, \dots, Q,$$

dove:

- f_p è il valore della funzione in ingresso (feature map) associata al canale p ;
- $a_{\Delta\theta,qp}(\theta, r)$ rappresenta i coefficienti del filtro p nel banco di filtri q , ruotati di $\Delta\theta$;

- P è il numero dei canali in ingresso della feature map;
- Q è il numero di canali in uscita (dimensione della feature map in uscita);
- \star denota l'operazione di convoluzione geodetica come definita nelle sezioni precedenti.

Il filtro $a_{\Delta\theta,qp}$ è definito come:

$$a_{\Delta\theta,qp}(\theta, r) = a_{qp}(\theta + \Delta\theta, r),$$

dove $\Delta\theta$ varia tra 0 e 2π , con N_θ rotazioni equi-angolari.

Calcolo dei Coefficienti del Filtro

Per calcolare i coefficienti di interpolazione, lo strato utilizza una distribuzione normale su coordinate polari geodetiche. Un approccio alternativo[19], prevede l'utilizzo di kernel gaussiani con parametri apprendibili, permettendo al modello di adattare automaticamente la forma dei filtri durante l'addestramento, creando una versione più generica (i parametri fissi utilizzati in questa rete sono casi specifici dei parametri addestrabili) e flessibile del modello.

3.4 Addestramento del Modello

3.4.1 Dataset e Obiettivi

Per l'addestramento del modello, è stato utilizzato un dataset contenente le coordinate tridimensionali dei punti dei pistoni come segnali di input, oltre alle coordinate baricentriche dei punti appartenenti alle varie patch. Queste ultime sono state utilizzate per fornire l'informazione posizionale per svolgere le convoluzioni. Il modello è stato addestrato per predire tre diverse grandezze fisiche: temperatura, turbolenza ed EGR (Exhaust Gas Recirculation). L'architettura del modello rimane la stessa per tutte le grandezze, variando solamente la ground truth fornita in input.

3.4.2 Ottimizzazione e Scheduler del Learning Rate

L'ottimizzazione del modello è stata eseguita utilizzando l'algoritmo Adam, con un learning rate iniziale $\alpha = 1 \times 10^{-3}$. Per migliorare la convergenza e prevenire l'overfitting, è stato implementato uno scheduler del learning rate, definito come:

$$\alpha(e) = \begin{cases} 1 \times 10^{-3}, & \text{se } e < 150 \\ 1 \times 10^{-4}, & \text{se } e \geq 150 \end{cases}$$

dove e rappresenta l'epoca corrente di addestramento. Questa strategia permette di iniziare con un learning rate più alto per facilitare l'esplorazione dello spazio dei parametri, riducendolo successivamente per affinare l'ottimizzazione.

Altri scheduler Durante la fase di sperimentazione, oltre allo scheduler descritto in precedenza, sono stati testati altri due scheduler del learning rate: il *decadimento esponenziale* e il *decadimento polinomiale*.

Il **decadimento esponenziale** del learning rate è definito dalla seguente espressione:

$$\alpha(e) = \alpha_0 \cdot \lambda^{\lfloor \frac{e}{s} \rfloor}$$

dove:

- α_0 è il learning rate iniziale (nel nostro caso 1×10^{-3}),
- λ è il fattore di decadimento (in questo caso 0,99),
- e è l'epoca corrente di addestramento,
- s è il numero di epoche dopo le quali il learning rate viene ridotto (qui 1000).

Applicando i valori specifici:

$$\alpha(e) = 1 \times 10^{-3} \cdot 0,99^{\lfloor \frac{e}{1000} \rfloor}$$

Questo scheduler diminuisce il learning rate in modo esponenziale ogni s epoche, consentendo una diminuzione graduale nel tempo.

Il **decadimento polinomiale** è invece definito dalla formula:

$$\alpha(e) = \alpha_0 \left(1 - \frac{e}{e_{\max}} \right)^p + \alpha_{\text{end}}$$

dove:

- α_0 è il learning rate iniziale (1×10^{-3}),
- α_{end} è il learning rate finale (1×10^{-5}),
- e_{\max} è il numero totale di epoche di addestramento (qui 2000),
- p è la potenza del polinomio (in questo caso 0,5),

- e è l'epoca corrente.

La formula specifica diventa:

$$\alpha(e) = (1 \times 10^{-3} - 1 \times 10^{-5}) \left(1 - \frac{e}{2000}\right)^{0,5} + 1 \times 10^{-5}$$

Questo scheduler riduce il learning rate seguendo un andamento polinomiale, permettendo un controllo più accurato sulla velocità di decadimento lungo le epoche.

Nonostante questi schemi di schedulazione del learning rate offrano teoricamente vantaggi nella convergenza, le sperimentazioni hanno mostrato che non apportavano benefici significativi rispetto allo scheduler più semplice a due fasi precedentemente descritto. Di conseguenza, si è preferito adottare quello che mantenesse il modello il più semplice possibile senza sacrificare le performance.

3.4.3 Funzione di Loss

La funzione di loss utilizzata è l'Errore Assoluto Medio (Mean Absolute Error, MAE), definito come:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

dove N è il numero di campioni nel batch, y_i è il valore target e \hat{y}_i è la previsione del modello per il campione i -esimo.

3.4.4 Strategie per Evitare l'Overfitting

Per evitare l'overfitting, è stata utilizzata la tecnica di *Early Stopping* con una *patience* di 20 epoche, attivata a partire dalla 200-esima epoca. Il criterio di arresto anticipato monitora la perdita sul set di validazione e interrompe l'addestramento se non si osservano miglioramenti significativi, ripristinando i pesi del modello alla migliore iterazione osservata.

3.4.5 Configurazione dell'Addestramento

Il modello è stato addestrato per un massimo di 450 epoche con un *batch size* di 7. Per ogni esperimento sono state addestrate 4 reti, ciascuna inizializzata con un diverso *seed*, in modo da irrobustire il modello rispetto all'inizializzazione casuale dei pesi, ed è stata poi selezionata la migliore delle quattro.

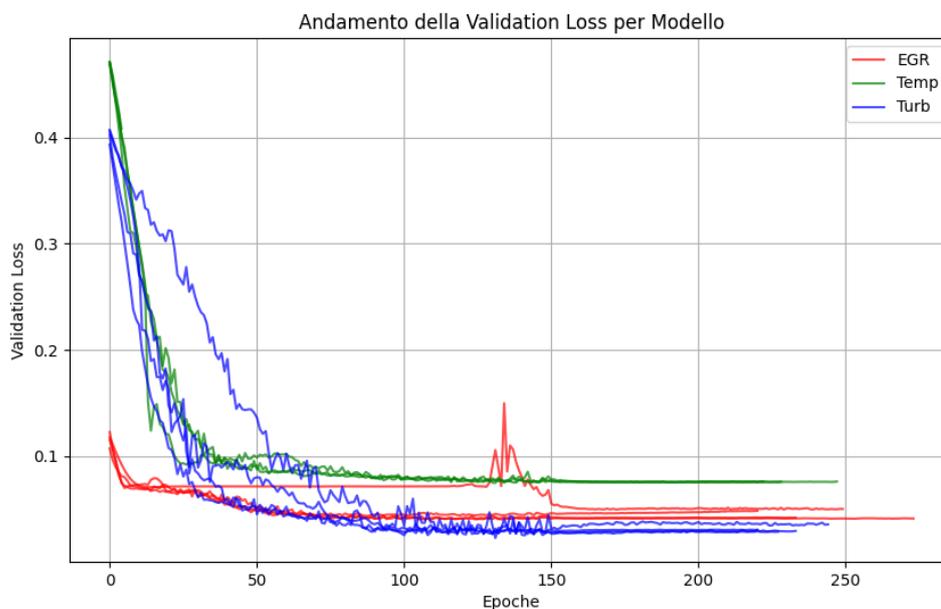


Figura 3.2: Addestramento dei modelli. Per ogni parametro da prevedere, sono stati addestrati 4 modelli, inizializzati con pesi diversi. Come si vede in un modello dell'EGR, esiste la possibilità che un addestramento rimanga bloccato in minimi locali, o incappi in altri tipi di problemi. Addestrare più modelli permette di aggirare agilmente eventuali inciampi di questo tipo, oltre che selezionare la versione più performante.

3.4.6 Monitoraggio e Salvataggio del Modello

Durante l'addestramento, sono stati utilizzati i *callback* di TensorBoard per monitorare in tempo reale le metriche di performance e il *CSVLogger* per registrare le informazioni sull'andamento della perdita e delle altre metriche a ogni epoca.

Al termine dell'addestramento, il modello addestrato è stato salvato per consentire una successiva fase di valutazione e inferenza sulle grandezze fisiche di interesse.

3.5 La Maschera

Il modello sviluppato produce come output un array tridimensionale di dimensioni $64 \times 64 \times 64$, rappresentante una discretizzazione volumetrica delle grandezze fisiche previste (temperatura, turbolenza, EGR) all'interno del pistone. Per garantire che le previsioni siano limitate esclusivamente all'area fisica del pistone e per evitare l'assegnazione di valori a punti esterni alla geometria reale, è stata applicata una maschera al risultato del modello. Questa maschera corrisponde all'area interna del

pistone fornito in input e permette di filtrare gli output indesiderati, assicurando che le previsioni siano rilevanti solo per le regioni di interesse.

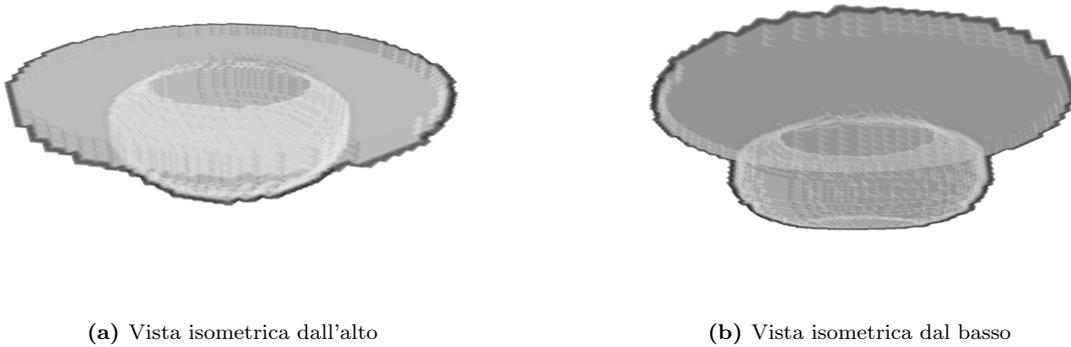


Figura 3.3: Maschera di uno degli elementi del dataset, costituita da valori binari (0 e 1).

Inizialmente, la maschera non è stata utilizzata unicamente come filtro post-process, ma è stata anche fornita in input al modello. L'ipotesi alla base di questa scelta era che l'inclusione della maschera potesse migliorare l'apprendimento della forma tridimensionale del pistone, fornendo al modello informazioni spaziali aggiuntive. In altre parole, si riteneva che il modello avrebbe potuto beneficiare di un riferimento tridimensionale supplementare durante la fase di apprendimento, potenziando la qualità delle previsioni.

La sezione del modello dedicata all'analisi della maschera era costituita da una serie di blocchi di strati ripetuti, ciascuno composto da:

- Uno strato di convoluzione tridimensionale (*3D convolutional layer*) con kernel $3 \times 3 \times 3$, stride pari a 2 e padding "same", per ridurre progressivamente le dimensioni spaziali dell'input mantenendo l'informazione sulla struttura volumetrica.
- Uno strato di *Batch Normalization*, per stabilizzare e accelerare il processo di apprendimento normalizzando le attivazioni degli strati precedenti.
- Una funzione di attivazione *Leaky ReLU* con parametro $\alpha = 0,01$, che introduce non linearità nel modello e allevia il problema della "morte dei neuroni" associato alla ReLU classica.
- Uno strato di *Dropout* con rate pari a 0,1, per ridurre l'overfitting omettendo casualmente alcuni neuroni durante la fase di addestramento.

Sono stati implementati sei blocchi di questo tipo, in cui il numero di filtri negli strati di convoluzione aumentava esponenzialmente, seguendo potenze di due (ad esempio, 4, 8, 16, 32, 64, 128 filtri). Dopo l'ultimo blocco, l'output risultante in un tensore di dimensioni $1 \times 1 \times 1 \times 64$ è stato rimodellato (*reshaped*) in un vettore unidimensionale di 64 elementi. Questo vettore è stato quindi concatenato all'output della sezione del modello responsabile delle convoluzioni geodetiche, prima di entrare nella fase di decodifica.

Per valutare l'effettiva utilità dell'inclusione della maschera come input, è stato condotto un esperimento in cui l'intera sezione del modello dedicata all'analisi della maschera è stata rimossa. In questa configurazione, al modello sono stati forniti in input esclusivamente le *patch* (le regioni locali della mesh) e i segnali (le coordinate tridimensionali dei punti), eliminando completamente l'informazione relativa alla maschera durante la fase di addestramento. La maschera è stata comunque utilizzata in fase di post-process per filtrare gli output.

3.5.1 Conclusioni sull'Utilizzo della Maschera

I risultati di questo esperimento hanno evidenziato che la rimozione della sezione relativa alla maschera non ha comportato variazioni significative nelle performance del modello. Le previsioni ottenute senza l'ausilio della maschera in input erano pressoché identiche a quelle del modello originale. Questo suggerisce che le convoluzioni geodetiche implementate nel modello siano sufficientemente efficaci nel catturare la curvatura della superficie e ricostruire la forma globale del pistone attraverso l'aggregazione delle informazioni provenienti dalle diverse *patch*. In sostanza, il modello è stato in grado di apprendere la forma tridimensionale dell'oggetto nella sua interezza senza la necessità di informazioni spaziali aggiuntive fornite dalla maschera.

Di conseguenza, si è ritenuto opportuno semplificare l'architettura rimuovendo la sezione relativa all'analisi della maschera, mantenendo comunque l'applicazione della stessa in fase di post-process per garantire la coerenza spaziale delle previsioni. Questa semplificazione contribuisce ad alleggerire il modello, rendendolo più efficiente senza compromettere le performance, riducendo la complessità computazionale e il numero di parametri da apprendere. Un modello più leggero facilita non solo il processo di addestramento, ma può anche portare a un miglioramento nella generalizzazione, riducendo il rischio di overfitting. Inoltre, sottolinea l'efficacia delle convoluzioni geodetiche nel trattamento di dati tridimensionali su superfici mesh.

3.6 Risultati

I risultati ottenuti dal modello sviluppato sono stati valutati confrontando le previsioni con i dati di ground truth forniti dalle simulazioni CFD. Per ogni parametro fisico di interesse (temperatura, turbolenza ed EGR), il modello produce un array tridimensionale di dimensioni $64 \times 64 \times 64$, rappresentante la distribuzione spaziale del parametro all'interno del pistone.

Per una valutazione visiva dei risultati, sono state estratte delle sezioni bidimensionali che attraversano il centro della geometria. Queste sezioni permettono di confrontare direttamente le previsioni del modello con i dati di ground truth, evidenziando le somiglianze e le eventuali discrepanze.

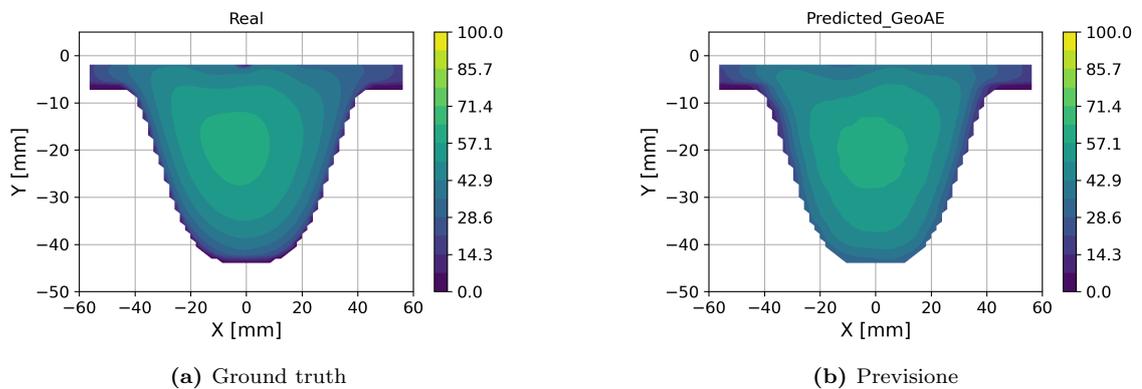


Figura 3.4: Sezione lungo l'asse Y della previsione della turbolenza all'interno di un pistone del dataset.

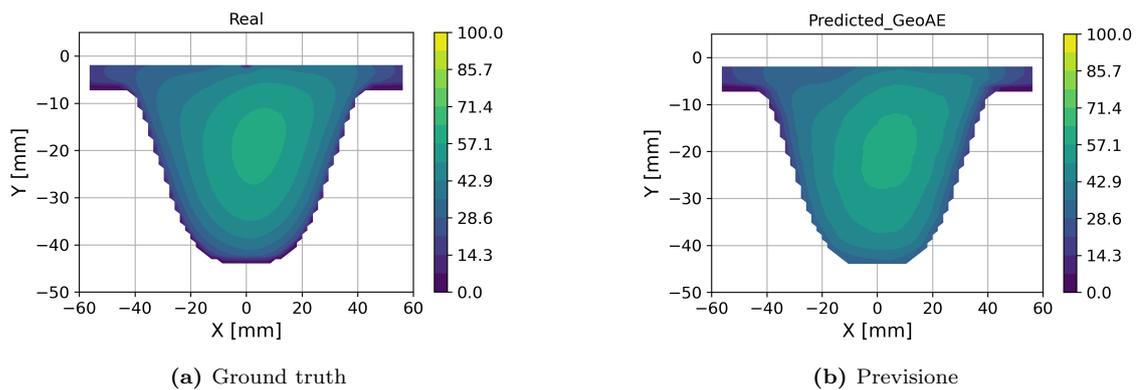


Figura 3.5: Sezione lungo l'asse X della previsione della turbolenza all'interno di un pistone del dataset.

Dalle immagini si può osservare che le predizioni del modello sono soddisfacenti e mostrano una buona corrispondenza con i dati di ground truth. Le principali caratteristiche spaziali dei parametri fisici sono ben catturate dal modello, con piccole differenze rispetto alle simulazioni CFD.

Un aspetto particolarmente rilevante è il tempo di calcolo. Mentre le simulazioni CFD richiedono un tempo computazionale significativo, il modello sviluppato è in grado di fornire previsioni in pochi secondi, seppur al costo della ridotta precisione. Infatti, programmi classici producono un maggior numero di punti nello spazio tridimensionale, che permettono di studiare meglio i fenomeni fisici all'interno dei pistoni.

In conclusione, i risultati ottenuti dimostrano che il modello sviluppato è in grado di predire con buona accuratezza i parametri fisici di interesse, offrendo un appoggio a studi preliminari nell'ambito della *Computational Fluid Dynamics* tradizionale.

Capitolo 4

Conclusioni

4.1 Sviluppi futuri

4.1.1 Convoluzioni Geodetiche Multi-Direzionali

Nell'ambito delle reti neurali convoluzionali geodetiche (GCNN), uno degli aspetti critici riguarda la gestione dell'ambiguità angolare durante l'applicazione delle convoluzioni su superfici tridimensionali. Nella rete sviluppata, questo problema è stato affrontato utilizzando lo strato *Angular Max-Pooling* (AMP), che seleziona, per ogni vertice, la risposta convoluzionale con valore massimo tra quelle ottenute dalle diverse rotazioni del filtro. Questo approccio garantisce l'invarianza alle rotazioni del filtro, ma comporta la perdita di informazioni direzionali potenzialmente rilevanti.

Un'alternativa promettente è rappresentata dalle *Convoluzioni Geodetiche Multi-Direzionali*[21], che mantengono tutte le risposte derivanti dalle diverse rotazioni del filtro convoluzionale, anziché selezionare esclusivamente quella massima. Questo metodo permette di propagare attraverso i vari strati della rete l'intera gamma di informazioni direzionali, consentendo al modello di apprendere caratteristiche più ricche e dettagliate delle geometrie analizzate.

L'utilizzo delle convoluzioni geodetiche multi-direzionali potrebbe migliorare le prestazioni del modello, in quanto permette di correlare informazioni direzionali tra diverse regioni della superficie. Mantenendo tutte le risposte alle diverse rotazioni, il modello acquisisce una rappresentazione più completa delle proprietà locali della superficie.

L'implementazione di questo approccio comporterebbe una revisione dell'architettura della rete, in particolare degli strati convoluzionali e di pooling. Sarebbe necessario gestire un numero maggiore di canali derivanti dalle diverse rotazioni, con un

impatto potenziale sul costo computazionale. Tuttavia, questa complessità aggiuntiva potrebbe essere giustificata dai benefici in termini di accuratezza e capacità di generalizzazione del modello.

4.1.2 Calcolo delle Coordinate Baricentriche

Le coordinate baricentriche sono uno strumento matematico fondamentale con numerose applicazioni in vari campi dell'ingegneria e della scienza, con utilizzi anche molto recenti in settori avanzati della robotica[7].

Nel contesto di questo lavoro, il calcolo efficiente delle coordinate baricentriche è cruciale per la generazione delle patch necessarie alle convoluzioni geodetiche. L'approccio tradizionale richiede un tempo computazionale non irrilevante, soprattutto quando si lavora con mesh complesse.

Per affrontare questo problema, è stato considerato un nuovo metodo per il calcolo delle coordinate baricentriche basato su campionamento stocastico [10]. Questo approccio offre una significativa riduzione dei tempi di calcolo senza sacrificare la precisione lineare. Il metodo riformula l'integrale del kernel che definisce le coordinate baricentriche in un problema di minimizzazione dei minimi quadrati pesati, consentendo l'utilizzo dell'integrazione Monte Carlo.

L'adozione di questa tecnica renderebbe la fase di preprocessing più veloce, permettendo di analizzare un maggior numero di geometrie in tempi ridotti.

4.1.3 Parametri Allenabili nei Coefficienti del Kernel

Nel modello sviluppato, i coefficienti del kernel utilizzato nelle convoluzioni geodetiche sono stati definiti con parametri fissi. Questo approccio limita la flessibilità del modello nell'adattarsi alle diverse caratteristiche delle geometrie analizzate. Un'alternativa promettente è proposta da Monti et al. [19], che introducono l'idea di utilizzare parametri apprendibili all'interno del kernel, permettendo al modello di apprendere direttamente dai dati la forma ottimale del filtro convoluzionale.

In particolare, il metodo prevede l'utilizzo di kernel parametrici definiti come una combinazione di funzioni gaussiane con parametri apprendibili. La formulazione generale dei coefficienti del filtro è la seguente:

$$w_j(\mathbf{u}) = \exp\left(-\frac{1}{2}(\mathbf{u} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\mathbf{u} - \boldsymbol{\mu}_j)\right),$$

dove:

- \mathbf{u} sono le pseudo-coordinate locali associate ai punti della superficie (ad esempio, coordinate geodetiche polari);
- $\boldsymbol{\mu}_j$ e $\boldsymbol{\Sigma}_j$ sono rispettivamente la media e la covarianza del kernel gaussiano, parametri apprendibili durante l'addestramento;
- $j = 1, \dots, J$ indica il numero di componenti nella combinazione del kernel.

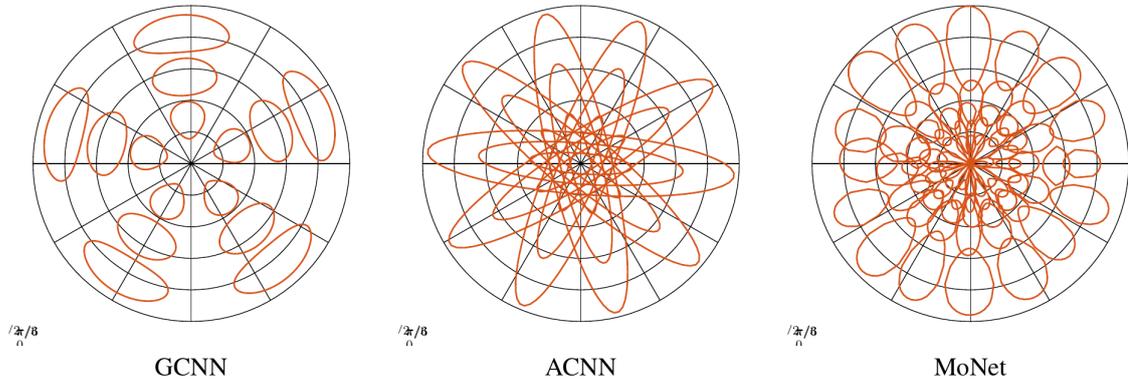


Figura 4.1: Confronto tra insiemi di livello (le linee rosse indicano il valore 0.5) dei kernel in tre generalizzazioni delle convoluzioni nelle varietà. I parametri del kernel sono fissi nel caso delle GCNN e delle ACNN[2] (*Anisotropic Convolutional Neural Network*), mentre sono invece parametri allenabili nel caso della MoNet. Con questa immagine si vuole evidenziare come la rete abbia imparato a dare più peso ad alcune zone della patch, e meno ad altre, come si evince dalla non simmetria delle curve di livello. Inoltre, la MoNet rappresenta una generalizzazione delle GCNN classiche, in quanto i parametri fissi utilizzati nelle GCNN sono un caso specifico dei parametri allenabili della MoNet. Immagine tratta da [19]

L'utilizzo di parametri apprendibili consente al modello di adattare dinamicamente la forma del kernel alle specifiche del dato in ingresso, migliorando la capacità di catturare caratteristiche locali della geometria e potenzialmente aumentando l'accuratezza delle previsioni. Tale approccio rende la convoluzione un processo adattivo, dove la rete apprende simultaneamente sia i pesi delle feature che la struttura spaziale dell'aggregazione.

Implementare questo metodo nel modello sviluppato richiederebbe una revisione degli strati convoluzionali per includere la parametrizzazione apprendibile dei kernel. Nonostante ciò possa comportare un aumento della complessità computazionale e del numero di parametri da addestrare, i benefici in termini di capacità di generalizzazione e adattabilità del modello potrebbero essere significativi.

4.2 Conclusioni

Il presente lavoro ha esplorato l'applicazione delle Reti Neurali Convolutionali Geometriche (GCNN) nel contesto della fluidodinamica computazionale, focalizzandosi sulla previsione di parametri fisici come la temperatura e il rapporto di ricircolo dei gas di scarico (EGR) su geometrie tridimensionali di pistoni. L'obiettivo principale era sviluppare un modello in grado di fornire previsioni accurate riducendo significativamente i tempi computazionali rispetto alle tradizionali simulazioni CFD.

Durante il percorso di sviluppo, il modello è stato notevolmente snellito sia nella fase di creazione del dataset, affrontando il problema della disomogeneità delle dimensioni dei triangoli nelle mesh, sia nell'architettura della rete neurale stessa. In particolare, è stata implementata una soluzione per uniformare le dimensioni dei triangoli, migliorando l'efficacia del modello e riducendo i costi computazionali. Parallelamente, l'architettura della rete è stata semplificata senza compromettere la qualità dei risultati, mantenendo elevate le prestazioni del modello nella previsione dei parametri fisici di interesse.

I risultati ottenuti indicano che l'utilizzo delle GCNN è adatto per questo tipo di applicazioni nel settore della fluidodinamica computazionale. Il modello proposto ha mostrato la capacità di predire con buona accuratezza la distribuzione dei parametri fisici, riducendo significativamente i tempi di elaborazione. Questo approccio può contribuire a ottimizzare i processi progettuali, fornendo supporto agli ingegneri nelle fasi iniziali senza la necessità di eseguire costose simulazioni.

Nonostante i risultati positivi, sono emerse alcune limitazioni legate alla dimensione ridotta del dataset e alla variabilità delle geometrie analizzate. La difficoltà nell'espandere il dataset è dovuta all'elevato costo computazionale delle simulazioni CFD necessarie per generare i dati di training, poiché ogni pistone richiede una simulazione complessa del circolo dei carburanti al suo interno. Questi aspetti evidenziano la necessità di ulteriori ricerche e perfezionamenti al modello per migliorarne la generalizzabilità e l'accuratezza.

In conclusione, l'impiego delle Reti Neurali Convolutionali Geometriche rappresenta una soluzione promettente per la previsione di parametri fisici in geometrie complesse. Pur non sostituendo completamente le simulazioni CFD, questi modelli possono integrarle efficacemente nelle fasi preliminari di progettazione, contribuendo a ridurre i tempi e i costi associati al processo.

Bibliografia

- [1] Pierre Baqué et al. “Geodesic Convolutional Shape Optimization”. In: *CoRR* abs/1802.04016 (2018). arXiv: 1802.04016. URL: <http://arxiv.org/abs/1802.04016>.
- [2] Davide Boscaini et al. “Learning shape correspondence with anisotropic convolutional neural networks”. In: *CoRR* abs/1605.06437 (2016). arXiv: 1605.06437. URL: <http://arxiv.org/abs/1605.06437>.
- [3] Michael M. Bronstein et al. “Geometric deep learning: going beyond Euclidean data”. In: *CoRR* abs/1611.08097 (2016). arXiv: 1611.08097. URL: <http://arxiv.org/abs/1611.08097>.
- [4] Michael M. Bronstein et al. “Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges”. In: *CoRR* abs/2104.13478 (2021). arXiv: 2104.13478. URL: <https://arxiv.org/abs/2104.13478>.
- [5] Luciano Arnaldo Romero Calla, Lizeth J. Fuentes Perez e Anselmo Antunes Montenegro. “A minimalistic approach for fast computation of geodesic distances on triangular meshes”. In: *CoRR* abs/1810.08218 (2018). arXiv: 1810.08218. URL: <http://arxiv.org/abs/1810.08218>.
- [6] Rudrasis Chakraborty et al. “ManifoldNet: A Deep Neural Network for Manifold-Valued Data With Applications”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.2 (2022), pp. 799–810. DOI: 10.1109/TPAMI.2020.3003846. URL: <https://doi.org/10.1109/TPAMI.2020.3003846>.
- [7] Ivan Chavdarov. “Application of Barycentric Coordinates and the Jacobian Matrix to the Analysis of a Closed Structure Robot”. In: *Robotics* 13.10 (2024). ISSN: 2218-6581. DOI: 10.3390/robotics13100152. URL: <https://www.mdpi.com/2218-6581/13/10/152>.
- [8] IANCU Cătălin, IANCU Daniela e Alin Stancioiu. “From cad model to 3d print via stl file format”. In: *Revista Fiabilitate si Durabilitate* 1 (giu. 2010).
- [9] Vladimir Florinski et al. “Technologies for supporting high-order geodesic mesh frameworks for computational astrophysics and space sciences”. In: *Computational Astrophysics and Cosmology* 7.1 (2020), pp. 1–24. DOI: 10.1186/

- s40668-020-00033-7. URL: <https://doi.org/10.1186/s40668-020-00033-7>.
- [10] Fernando de Goes e Mathieu Desbrun. “Stochastic Computation of Barycentric Coordinates”. In: *ACM Trans. Graph.* 43.4 (lug. 2024). ISSN: 0730-0301. DOI: 10.1145/3658131. URL: <https://doi.org/10.1145/3658131>.
- [11] Thibault Groueix et al. “AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation”. In: *CoRR* abs/1802.05384 (2018). arXiv: 1802.05384. URL: <http://arxiv.org/abs/1802.05384>.
- [12] K. Kanakoglou. *The notion of abstract Manifold: a pedagogical approach*. 2012. arXiv: 1204.2191 [math-ph]. URL: <https://arxiv.org/abs/1204.2191>.
- [13] R. Kimmel e J. A. Sethian. “Computing geodesic paths on manifolds”. In: *Proceedings of the National Academy of Sciences* 95.15 (1998), pp. 8431–8435. DOI: 10.1073/pnas.95.15.8431. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.95.15.8431>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.95.15.8431>.
- [14] Tong Lin e Hongbin Zha. “Riemannian Manifold”. In: *Computer Vision: A Reference Guide*. Cham: Springer International Publishing, 2021, pp. 1081–1086. ISBN: 978-3-030-63416-2. DOI: 10.1007/978-3-030-63416-2_801. URL: https://doi.org/10.1007/978-3-030-63416-2_801.
- [15] Nithin Mallya et al. “Geodesic Convolutional Neural Network Characterization of Macro-Porous Latent Thermal Energy Storage”. In: *ASME Journal of Heat and Mass Transfer* 145 (feb. 2023). DOI: 10.1115/1.4056663.
- [16] Andreas Mazur. *GeoConv - Geodesic Convolutions with Tensorflow*. <https://github.com/andreasMazur/geoconv>. URL: <https://github.com/andreasMazur/geoconv>.
- [17] Luke Melas-Kyriazi. “The Mathematical Foundations of Manifold Learning”. In: *CoRR* abs/2011.01307 (2020). arXiv: 2011.01307. URL: <https://arxiv.org/abs/2011.01307>.
- [18] Eivind Melvaer e Martin Reimers. “Geodesic Polar Coordinates on Polygonal Meshes”. In: *Computer Graphics Forum* 31 (dic. 2012), p. 2423. DOI: 10.1111/j.1467-8659.2012.03187.x.
- [19] Federico Monti et al. *Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs*. Lug. 2017. DOI: 10.1109/CVPR.2017.576.
- [20] Bo Pang et al. “Learning the Geodesic Embedding with Graph Neural Networks”. In: *ACM Transactions on Graphics* 42 (dic. 2023), pp. 1–12. DOI: 10.1145/3618317.
- [21] Adrien Poulencard e Maks Ovsjanikov. “Multi-directional Geodesic Neural Networks via Equivariant Convolution”. In: *CoRR* abs/1810.02303 (2018). arXiv: 1810.02303. URL: <http://arxiv.org/abs/1810.02303>.

BIBLIOGRAFIA

- [22] Edoardo Remelli et al. “MeshSDF: Differentiable Iso-Surface Extraction”. In: *CoRR* abs/2006.03997 (2020). arXiv: 2006.03997. URL: <https://arxiv.org/abs/2006.03997>.
- [23] Ofir Weber et al. “Parallel algorithms for approximation of distance maps on parametric surfaces”. In: *ACM Trans. Graph.* 27.4 (nov. 2008). ISSN: 0730-0301. DOI: 10.1145/1409625.1409626. URL: <https://doi.org/10.1145/1409625.1409626>.