Algebra geometrica e applicazioni al Deep Learning

Giacomo Bencivinni, Alin Marian Habasescu, Riccardo Lo Iacono

9 gennaio 2025

Geometric Algebra

Autore: Riccardo Lo Iacono

Algebra di Clifford

Sia fissato uno spazio vettoriale $\mathbb{R}^n$, e sia $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ una sua base ortonormale.

Da questi è possibile definire un nuovo spazio vettoriale, detto spazio di Clifford n-dimensionale ($\mathcal{C}\ell_n$). Nello specifico, $\mathcal{C}\ell_n$ rappresenta l'insieme di tutti i possibili sottospazi k-dimensionali, con $k \leq n$, di tutte le possibili combinazioni dei vettori base.

Multivettori: il caso bi- e tri-dimensionale

Si consideri lo spazio $\mathbb{R}^2$ e sia $\{\mathbf{e}_1, \mathbf{e}_2\}$ una sua base ortonormale. É noto che comunque presi $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$, questi possano essere intesi come opportune combinazioni lineari dei vettori base. Ossia

$$\mathbf{a} = \alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2 \qquad \mathbf{a} = \beta_1\mathbf{e}_1 + \beta_2\mathbf{e}_2$$

inoltre, sappiamo che

$$\begin{aligned} \mathbf{ab} &= (\alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2)(\beta_1\mathbf{e}_1 + \beta_2\mathbf{e}_2) \\ &= \alpha_1\beta_1\mathbf{e}_1\mathbf{e}_1 + \alpha_1\beta_2\mathbf{e}_1\mathbf{e}_2 + \alpha_2\beta_1\mathbf{e}_1\mathbf{e}_2 + \alpha_2\beta_2\mathbf{e}_2\mathbf{e}_2. \end{aligned} \tag{1}$$

Definindo i seguenti assiomi:

1. $\mathbf{e}_i\mathbf{e}_i = 1$
2. $\mathbf{e}_j\mathbf{e}_i = -\mathbf{e}_i\mathbf{e}_j$

Equazione (1) può essere riscritta come

$$\mathbf{ab} = (\alpha_1\beta_1 + \alpha_2\beta_2) + (\alpha_1\beta_2 - \alpha_2\beta_1)\mathbf{e}_1\mathbf{e}_2 \tag{2}$$

Da *Equazione* (2) segue che, $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^2$ il loro prodotto risulti essere la somma di un termine scalare $(\alpha_1\beta_1 + \alpha_2\beta_2)$ e da un termine $(\alpha_1\beta_2 - \alpha_2\beta_1)\mathbf{e}_1\mathbf{e}_2$.

Dando un'interpretazione geometrica, $(\alpha_1\beta_2 - \alpha_2\beta_1)$ descrive l'area del rettandolo definita dai vettori $\mathbf{e}_1, \mathbf{e}_2$; segue che $(\mathbf{e}_1, \mathbf{e}_2)$ definiscono il piano su cui giace l'area.

In conclusione, $\mathbf{e}_1\mathbf{e}_2$ rappresenta un'area orientata in $\mathbb{R}^2$ ed è definita *bivettore*.

Un ragionamento analogo può essere fatto per $\mathbb{R}^3$.

Sia $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ una base ortonormale di $\mathbb{R}^3$. Considerati $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, questi saranno della forma

$$\mathbf{a} = \alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2 + \alpha_3\mathbf{e}_3 \qquad \mathbf{b} = \beta_1\mathbf{e}_1 + \beta_2\mathbf{e}_2 + \beta_3\mathbf{e}_3$$

Considerandone il prodotto, e applicando gli assiomi definti poco sopra, segue

$$\begin{aligned}\mathbf{ab} = &(\alpha_1\beta_1 + \alpha_2\beta_2 + \alpha_3\beta_3) \\ &+ (\alpha_1\beta_2 - \alpha_2\beta_1)\mathbf{e}_1\mathbf{e}_2 \\ &+ (\alpha_1\beta_3 - \alpha_3\beta_1)\mathbf{e}_1\mathbf{e}_3 \\ &+ (\alpha_2\beta_3 - \alpha_3\beta_2)\mathbf{e}_2\mathbf{e}_3\end{aligned}$$

Oltre le tre aree, in $\mathbb{R}^3$ è possibile definire un elemento dato dal prodotto dei vettori base: $\mathbf{e}_1\mathbf{e}_2\mathbf{e}_3$.

Similarmente all'area orientata rappresentata da $\mathbf{e}_1\mathbf{e}_2$, $\mathbf{e}_1\mathbf{e}_2\mathbf{e}_3$ rappresenta un volume orientata in $\mathbb{R}^3$.

Multivettori: il caso generale

In generale fissato un n, e assunti $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ una base ortonormale di uno spazio $\mathbb{R}^n$, un multivettore $\mathbf{a} \in \mathbb{R}^n$ è un elemento della forma

$$\mathbf{a} = \alpha_0 + \alpha_1 \mathbf{e}_1 + \cdots \alpha_n \mathbf{e}_n + \alpha_{1,2} \mathbf{e}_1 \mathbf{e}_2 + \cdots + \alpha_{1,\ldots,n} \mathbf{e}_1 \cdots \mathbf{e}_n$$

e lo spazio che contiene tutti questi multivettori è detto spazio di Clifford n-dimensionale ($\mathcal{C}\ell_n$). In ultimo, il multivettore dato dal prodotto di ogni vettore base è detto *pseudo-scalare*.

Operazioni tra multivettori

Fissata una qualche algebra di Clifford $\mathcal{C}\ell_n$, su gli elementi della stessa è possibile applicare diverse operazioni, quali

- prodotto geometrico
- prodotto interno
- operazioni unarie

Operazioni tra multivettori: il prodotto geometrico

Sia considerata $\mathcal{C}\ell_2$, (l'estensione al caso n-simo è immediata), e siano $\mathbf{a}, \mathbf{b}$ due multivettori.

Il prodotto geometrico tra i due consiste nel moltiplicare ciascuna delle componenti del primo multivettore per quelle del secondo, tenendo conto dei seguenti assiomi:

- $\mathbf{e}_i \mathbf{e}_i = \pm 1$
- $\mathbf{e}_i \mathbf{e}_j = -\mathbf{e}_i \mathbf{e}_j$
- $\lambda \mathbf{e}_i = \mathbf{e}_i \lambda$

Si ha cioè che

$$\begin{aligned}\mathbf{ab} &= (\alpha_0 + \alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2 + \alpha_{12}\mathbf{e}_{12})(\beta_0 + \beta_1\mathbf{e}_1 + \beta_2\mathbf{e}_2 + \beta_{12}\mathbf{e}_{12}) \\ &= \alpha_0\beta_0 + \alpha_0\beta_1\mathbf{e}_1 + \alpha_0\beta_2\mathbf{e}_2 + \alpha_0\beta_{12}\mathbf{e}_{12} \\ &+ \alpha_1\beta_0\mathbf{e}_1 + \alpha_1\beta_1\mathbf{e}_{11} + \alpha_1\beta_2\mathbf{e}_{12} + \alpha_1\beta_{12}\mathbf{e}_{112} \\ &+ \alpha_2\beta_0\mathbf{e}_2 + \alpha_2\beta_1\mathbf{e}_{21} + \alpha_2\beta_2\mathbf{e}_{22} + \alpha_2\beta_{12}\mathbf{e}_{212} \\ &+ \alpha_{12}\beta_0\mathbf{e}_{12} + \alpha_{12}\beta_1\mathbf{e}_{121} + \alpha_{12}\beta_2\mathbf{e}_{122} + \alpha_{12}\beta_{12}\mathbf{e}_{1212}\end{aligned}$$

diventa, applicando gli assiomi di cui sopra

$$\begin{aligned}\mathbf{ab} &= (\alpha_0\beta_0 + \alpha_1\beta_1 + \alpha_2\beta_2 - \alpha_{12}\beta_{12}) \\ &+ (\alpha_0\beta_1 + \alpha_1\beta_0 - \alpha_2\beta_{12} + \alpha_{12}\beta_2)\mathbf{e}_1 \\ &+ (\alpha_0\beta_2 + \alpha_2\beta_0 + \alpha_1\beta_{12} - \alpha_{12}\beta_1)\mathbf{e}_2 \\ &+ (\alpha_0\beta_{12} + \alpha_{12}\beta_0 + \alpha_1\beta_2 - \alpha_2\beta_1)\mathbf{e}_{12}\end{aligned}$$

Si dimostra inoltre che il prodotto geometrico gode delle seguenti proprietà:

Associatività:

$$\forall \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n, (\mathbf{ab})\mathbf{c} = \mathbf{a}(\mathbf{bc})$$

Commutatività rispetto uno scalare:

$$\forall \mathbf{a} \in \mathbb{R}^n, \lambda \in \mathbb{R}, \lambda\mathbf{a} = \mathbf{a}\lambda$$

Distributività rispetto la somma:

$$\forall \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n, \mathbf{a}(\mathbf{b} + \mathbf{c}) = \mathbf{ab} + \mathbf{ac}$$

Operazioni tra multivettori: prodotto interno

Sebbene ne esistano varie formulazioni, di interesse è la forma contratta di Lounesto ($\rfloor$). Nello specifico, questa può essere intesa come una generalizzazione del prodotto scalare.

Siano α, β scalari, a, b vettori e $\mathsf{A}, \mathsf{B}, \mathsf{C}$ multivettori, allora vale quanto segue:

$$\begin{aligned}
\alpha \rfloor \beta &= \alpha\beta \\
\alpha \rfloor \mathsf{b} &= \alpha\mathsf{b} \\
\mathsf{a} \rfloor \beta &= 0 \\
\mathsf{a} \rfloor \mathsf{b} &= \mathsf{a}\mathsf{b} \\
\mathsf{a} \rfloor (\mathsf{b} \wedge \mathsf{C}) &= (\mathsf{a} \rfloor \mathsf{b}) \wedge \mathsf{C} - \mathsf{b} \wedge (\mathsf{a} \rfloor \mathsf{C})
\end{aligned}$$

Operazioni tra multivettori: il prodotto geometrico v2

Formulazione alternativa del prodotto geometrico è la seguente:

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b}$$

ove $\mathbf{a} \cdot \mathbf{b}$, detto *prodotto inteno*, determina la componente scalare del prodotto; $\mathbf{a} \wedge \mathbf{b}$, detto *prodotto esterno*, ne determina la componente multivettoriale.

Più correttamente sono $\mathbf{a} \cdot \mathbf{b}$ e $\mathbf{a} \wedge \mathbf{b}$ ad essere definiti in termini di prodotto geometrico. Nello specifico

$$\mathbf{a} \wedge \mathbf{b} = \frac{1}{2}(\mathbf{ab} - \mathbf{ba})$$

$$\mathbf{a} \cdot \mathbf{b} = \frac{1}{2}(\mathbf{ab} + \mathbf{ba})$$

Operatori unari

Di interesse risultano essere l'operatore duale ($\mathbf{x}^*$) e l'inverso ($\mathbf{x}^{-1}$).

Sia considerato l'inverso: questi, dato $\mathbf{a}$ un multivettore, definisce il multivettore $\mathbf{a}^{-1}$ tale che $\mathbf{a}\mathbf{a}^{-1} = 1$. Come nel caso matriciale, l'esistenza di $\mathbf{a}^{-1}$ non è garantita.

Sono per tanto considerati i cosidetti versori: multivettori esprimibili nella forma $\mathbf{a} = v_1 v_2 \cdots v_k$; per i quali è possibile garantire l'esistenza dell'inverso.

Considerando il duale: sia $\mathbf{a}$ un multivettore, il suo *duale* $\mathbf{a}^* = \mathbf{a}\mathbf{I}^{-1}$ rappresenta il vettore normale al multivettore.

Sia ad esempio considerato $\mathbf{e}_{12}$, calcolandone il duale si ha

$$\begin{aligned}
\mathbf{e}_{12}^* &= \mathbf{e}_1\mathbf{e}_1\mathbf{e}_3\mathbf{e}_2\mathbf{e}_1 \\
&= -\mathbf{e}_1\mathbf{e}_3\mathbf{e}_2\mathbf{e}_2\mathbf{e}_1 \\
&= -\mathbf{e}_1\mathbf{e}_3\mathbf{e}_1 \\
&= \mathbf{e}_3\mathbf{e}_1\mathbf{e}_1 \\
&= \mathbf{e}_3
\end{aligned}$$

Bibliografia

- Jaap Suter, *Geometric Algebra Primer*, 2003
- Alan Macdonald, *A Survey of Geometric Algebra and Geometric Calculus, applied Geometric Algebra in Computer Science and Engineering*, Barcelona, Spain, 29-31 Luglio, 2015
- Sivia Franchini, Giorgio Vassallo, Filippo Sorbello, *A brief introduction to Clifford algebra*, Febbraio 2010.

CGAPoseNet+GCAN

Autore: Giacomo Bencivinni

Isometrie

Dato V uno spazio vettoriale, definiamo *isometria* una funzione $f : V \to V$ definita come segue:

$$\forall \mathbf{u}, \mathbf{v} \in V, d(\mathbf{u}, \mathbf{v}) = d(f(\mathbf{u}), f(\mathbf{v}))$$

con d funzione per il calcolo della distanza tra i multivettori. In altri termini, ogni trasformazione geometrica che preserva le distanze è un'isometria. In generale, le isometrie sono classificate sulla base delle trasformazioni geometriche che rappresentano, da ciò si parla di: riflessioni, rotazioni, traslazioni, roto-riflessioni

Teorema di Cartan-Dieudonné

Il teorema di Cartan-Dieudonné afferma che ogni trasformazione ortogonale di uno spazio n-dimensionale può essere decomposto in, al più, n riflessioni su iperpiani. Alla luce di questo teorema introduciamo la definizione di isometria impropria: isometria composta da un numero dispari di riflessioni. È importante notare che tale tipo di isometria altera la chiralità dello spazio

Pin(n) Group

In uno spazio n-dimensionale, le composizioni di riflessioni formano il Pin(n) group, un potente strumento per modellare le isometrie: Il gruppo $(G, \circ)$ è l'insieme non vuoto dotato di un'operazione binaria $\circ : G \times G \to G$ che soddisfa le seguenti proprietà:

1. **Chiusura:** $uv \in G,\ \forall u, v \in G$
2. **Associatività:** $(uv)w = u(vw),\ \forall u, v, w \in G$
3. **Identità:** $\exists 1 \in G \mid 1u = u = u1,\ \forall u \in G$
4. **Inverso:** $\exists\, u^{-1} \in G \mid uu^{-1} = 1 = u^{-1}u,\ \forall u \in G$

Poichè il Pin(n) Group è definito come l'insieme di tutte le possibili composizioni di riflessioni in uno spazio euclideo n-dimensionale e soddisfa tutte le proprietà precedentemente elencate, esso forma un gruppo. Di conseguenza ogni elemento di Pin(n) può essere definito da una composizione di k riflessioni linearmente indipendenti.

Azione di gruppo: Coniugazione

Un'azione di gruppo è un omomorfismo dal gruppo stesso al gruppo delle trasformazioni dello spazio, in altre parole un'azione di gruppo associa ad ogni elemento del gruppo una specifica trasformazione dello spazio. Nel caso in cui un gruppo agisca su se stesso si parla di coniugazione, cioè una specifica mappa $G \times G \to G$ che, dati $u, v \in G$, definisce come un elemento del gruppo, che rappresenta una trasformazione, agisce su un altro elemento dello stesso gruppo, che rappresenta un oggetto geometrico, nel seguente modo:

$$u[v] \to uvu^{-1} \tag{3}$$

Concludiamo quindi che la coniugazione offre un approccio elegante e efficiente per unificare le trasformazioni e gli oggetti e assicura che gli oggetti trasformati mantengano le loro proprietà fondamentali.

Rappresentazione degli elementi di Pin(p,q,r)

Generalizzando il concetto di $Pin(n)$ e considerando la firma dello spazio vettoriale, possiamo definire l'algebra geometrica G_{pqr} che rappresenta gli elementi di $Pin(p, q, r)$ in cui:

1. p definisce il numero di dimensioni positive
2. q definisce il numero di dimensioni negative
3. r definisce il numero di dimensioni nulle

Per uno spazio n-dimensionale si ha sempre che $p + q + r = n$

Si noti che la firma dello spazio influisce sulle trasformazioni geometriche rappresentabili e la preferenza di una specifica algebra geometrica dipende dalla geometria del problema e dalle trasformazioni da modellare.
Come discusso precedentemente l'operazione fondamentale per la rappresentazione di isometrie è la riflessione, di conseguenza definendo la rappresentazione della riflessione sarà possibile definire tutti gli elementi di $Pin(p, q, r)$

Consideriamo un iperpiano passante per l'origine rappresentato da un vettore di grado 1 in G_{pqr}. Tale vettore ha due componenti: la parte vettoriale che rappresenta la normale all'iperpiano $n = [a\ b\ c]^T$ e la distanza dall'origine δ (=0 in nel caso che stiamo considerando)

$$ax + by + cz + ... + \delta = 0 \iff u := ae_1 + be_2 + ce_3 + ... + \delta e_0$$

Riflessione in $Pin(p, q, r)$

Si noti che considerati un vettore e il suo negativo essi hanno stessa parte vettoriale e di conseguenza identificano lo stesso iperpiano. Questa considerazione è fondamentale per definire in maniera corretta e coerente l'operazione di riflessione su un iperpiano indipendentemente dal vettore scelto per rappresentare l'iperpiano. A questo scopo introduciamo il segno meno e definiamo una riflessione:

$$v \to -uvu^{-1} \tag{4}$$

Composizioni di riflessioni in $Pin(p, q, r)$

Componendo due o più riflessioni si ottengono altri elementi del gruppo $Pin(n)$, di seguito mostriamo che una biriflessione equivale a una rotazione come suggerito dalla presenza del bivettore $u_1 u_2$

$$\begin{aligned} v &\to -u_1 v u^{-1} \\ &= -u_2(-u_1 v u^{-1})u_2^{-1} \\ &= (u_2 u_1) v (u_2 u_1)^{-1} \end{aligned}$$

Azione di gruppo in $Pin(p, q, r)$

Infine notando che la composizione di riflessioni modifica la chiralità dello spazio, definiamo un azione di gruppo nel seguente modo

$$v \to u[v] := (-1)^{kl} uvu^{-1} \tag{5}$$

dove u e v sono espressi come combinazioni lineare di, rispettivamente, k e l riflessioni. Il -1 assicura che l'orientazione dello spazio sia corretta.

Geometric Clifford Algebra Networks (GCAN)

Le Geometric Clifford Algebra Networks (GCAN) rappresentano una nuova frontiera nell'ambito delle reti neurali, utilizzando l'algebra geometrica come base per la rappresentazione e manipolazione dei dati geometrici. Questo approccio si fonda su un potente framework matematico che consente di trattare in modo unificato ed efficiente oggetti e trasformazioni geometriche come rotazioni, traslazioni e riflessioni.

Le GCAN sfruttano i vantaggi di questa rappresentazione per costruire modelli capaci di apprendere e applicare trasformazioni geometriche in modo intrinsecamente coerente con le proprietà dello spazio dei dati. A differenza degli approcci tradizionali basati sull'algebra lineare, dove oggetti e trasformazioni devono essere rappresentati separatamente, il framework dell'algebra geometrica riduce la complessità e minimizza le incongruenze geometriche.

Principali vantaggi di GCAN

- Rappresentazione unificata di oggetti e trasformazioni: L'algebra geometrica consente di trattare in modo uniforme diversi tipi di oggetti geometrici e le loro trasformazioni. Questo approccio semplifica l'implementazione dei modelli e migliora la loro efficienza operativa.
- Trasformazioni covarianti: Gli oggetti geometrici, rappresentati nell'algebra geometrica, si trasformano in modo coerente rispetto alle trasformazioni dello spazio. Ciò significa che le stesse regole di trasformazione possono essere applicate a diversi tipi di oggetti, mantenendo una coerenza intrinseca.

- Generalizzazione dimensionale: Le GCAN sono intrinsecamente indipendenti dalla dimensionalità dello spazio. Questo consente di applicare lo stesso modello a dati di diverse dimensioni senza modifiche sostanziali all'architettura della rete.

Concetti fondamentali delle GCAN

Le GCAN si basano sull'idea di preservare la coerenza geometrica durante le trasformazioni. Questo si riflette in due proprietà chiave:

- Conservazione del tipo degli oggetti geometrici: Dopo una trasformazione gli oggetti geometrici mantengono il loro tipo originario. Questo corrisponde, in algebra geometrica, alla conservazione del grado dei multivettori.
- Trasformazioni basate su azioni di gruppo Le trasformazioni sono eseguite come combinazioni lineari di azioni di gruppo, che garantiscono la validità geometrica delle operazioni. In particolare, le GCAN sfruttano il gruppo$Pin(p, q, r)$ che preserva le proprietà geometriche dello spazio durante le trasformazioni.

Concetti fondamentali delle GCAN: Group Action Layers

I Group Action Layers sono strati neurali che implementano le trasformazioni geometriche nelle GCAN. Questi strati utilizzano le azioni di gruppo per applicare le trasformazioni geometriche agli input in modo controllato e coerente. Dato un gruppo G, uno spazio vettoriale X, un'azione di gruppo $\alpha : G \times X \to X$, essi sono definiti in termini generali dalla seguente scrittura:

$$x \to T_{g,w}(x) := \sum_{i=1}^{c} w_i \cdot \alpha(g_i, x_i)$$

Questa trasformazione calcola una combinazione lineare pesata delle azioni di gruppo, modulata dai pesi w_i ottimizzati durante l'addestramento della rete

Esperimento: Tetris

Obiettivo: Modellare traiettorie complesse di oggetti rigidi
Gli oggetti, inizialmente posizionati all'origine, vengono sottoposti a una serie di traslazioni e rotazioni casuali attorno ai rispettivi centri di massa. Tali movimenti non sono indipendenti: le rotazioni e le traslazioni risultano correlate, generando una dipendenza tra le varie trasformazioni. Inoltre per rendere il modello ancora più realistico, a ogni parte dell'oggetto viene aggiunto un rumore gaussiano condizionato, simulando piccole deformazioni che potrebbero verificarsi durante il movimento.

Modelli utilizzati

L'esperimento utilizza due tipi di reti basati su GCAN:

- GCA-MLP: una rete multistrato perceptron (MLP) opportunamente modificata per integrare i GCA Linear Layers
- GCA-GNN: una rete neurale a grafi (GNN) adattata per sfruttare i vantaggi dei layer GCA.

Entrambi i modelli sfruttano l'algebra geometrica $G_{3,0,1}$, particolarmente adatta per descrivere movimenti rigidi nello spazio euclideo. All'interno di questa algebra, i punti degli oggetti Tetris sono rappresentati come trivettori, che codificano l'intersezione di tre piani per determinare un punto nello spazio. Le trasformazioni geometriche, come rotazioni e traslazioni, vengono applicate utilizzando l'operazione di "sandwich".

Obiettivo e valutazioni

L'obiettivo principale dei modelli è prevedere, a partire dalle posizioni degli oggetti in quattro istanti temporali consecutivi, le loro posizioni nei momenti successivi. Per valutare le prestazioni dei modelli, viene utilizzato l'errore quadratico medio (MSE) calcolato tra le posizioni previste e quelle reali.

Risultati

Come si può vedere dal grafico, le GCAN hanno dimostrato una capacità di generalizzazione superiore rispetto ad approcci tradizionali come reti MLP standard, MotorMLP, SO(3)-MLP, O(3)-MLP, EdgeConv GNN e altre varianti di GNN. Questa superiorità è stata particolarmente evidente in scenari con quantità limitata di dati di addestramento. Tale risultato è attribuibile al forte bias induttivo introdotto dai layer GCA, che vincolano le trasformazioni a rispettare rigorosamente la struttura geometrica dei dati.

Limitazioni GCAN

Le principali limitazioni legate alle GCAN riguardano la complessità computazionale delle operazioni algebriche e la necessità di ottimizzare l'implementazione per l'hardware. Infatti le operazioni algebriche, in particolare il prodotto geometrico, richiedono un maggiore carico computazionale rispetto alle operazioni standard dell'algebra lineare. Questo si traduce in un aumento dei tempi di esecuzione e maggiori requisiti di memoria. Dall'altro lato vi è la necessità di ottimizzare l'implementazione per sfruttare al meglio l'hardware disponibile. Infatti le operazioni algebriche complesse possono non essere ben supportate dalle librerie ottimizzate per calcoli lineari, richiedendo adattamenti specifici o sviluppo di soluzioni personalizzate per garantire l'efficienza.

Rotori

Definendo il rotore coome il prodotto geometrico di un numero pari di vettori con $R = u_1 u_2 ... u_k$ e il suo inverso $\tilde{R} = u_k u_{k-1} ... u_1$ possiamo definire il prodotto sandwich $v' = Rv\tilde{R}$.
Tale prodotto rappresenta la rotazione di un oggetto in uno spazio 3D e appartiene al gruppo $Spin(n)$
Si noti che nel caso in cui k non sia pari, v' rappresenta una roto-riflessione ovvero una trasformazione che combina una rotazione e una riflessione

Algebra geometrica conforme (CGA)

L'algebra geometrica conforme è un potente strumento matematico che estende l'algebra geometrica standard per modellare in modo efficiente la geometria euclidea incorporando lo spazio euclideo 3d in uno spazio di dimensioni superiori introducendo due punti speciali: il punto all'origine e il punto all'infinito

- punto all'origine : Fornisce un punto di riferimento da cui vengono misurate le distanze e le direzioni
- punto all'infinito : Consente di rappresentare le traslazioni come rotazioni attorno a punti all'infinito permettendo di unificare la rappresentazioni delle trasformazioni di rotazione e traslazione.

1D-Up CGA

Nel caso di 1D-Up CGA, lo spazio euclideo viene incorporato in uno spazio 4D aggiungendo una sola dimensione extra. La 1D-UP CGA utilizza l'algebra $G_{4,0,0}$ che abbrevieremo in $G_{4,0}$. In questa algebra i punti nello spazio 3D vengono mappati nello spazio 4D utilizzando la seguente funzione specifica:

$$X = f(x) = \frac{2\lambda}{\lambda^2 + x^2} x + \frac{\lambda^2 - x^2}{\lambda^2 + x^2} e_4 \tag{6}$$

Dove λ è uno scalare che indica la curvatura dello spazio sferico 4D e e_4 è il vettore di base per la quarta dimensione

Motori

In $G_{4,0}$ un rotore può rappresentare sia una traslazione che una rotazione.
Dato un vettore di traslazione $t \in G_{3,0}$ il rotore corrispondente è dato da:

$$T = g(t) = \frac{\lambda + te_4}{\sqrt{\lambda^2 + t^2}}$$

Si può quindi esprimere la traslazione o rotazione di un oggetto X in X' tramite la combinazione di due sandwich products

$$X' = TRX\tilde{R}\tilde{T} = MX\tilde{M} \tag{7}$$

Motori

Il prodotto geometrico $M = TR$ rappresenta un motore I motori permettono di unificare la rappresentazione di rotazione e traslazione in unica entità permettendo di semplificare la rappresentazione della posa e nell'algebra considerata un motore è definito da 1 scalare, 6 bivettori e 1 quadrivettore:

$$\begin{aligned} M = & x_{01} \\ & + x_{12}e_{12} + x_{13}e_{13} + x_{14}e_{14} + x_{23}e_{23} + x_{24}e_{24} + x_{34}e_{34} \\ & + x_{1234}e_{1234} \end{aligned}$$

Dove i bivettori rappresentano i piani di rotazione e l'ampiezza della rotazione è determinata dai rispettivi coefficienti mentre il quadrivettore rappresenta la traslazione nello spazio euclideo

CGAPoseNet+GCAN

L'architettura CGAPoseNet+GCAN è progettata per la regressione della posa della telecamera da immagini RGB e si basa sull'algebra CGA 1D-Up vista in precedenza.
Il principale obiettivo di questa architettura è superare il modello CGAPoseNet che si limita a predire i coefficienti dei motori senza sviluppare una comprensione del loro significato geometrico o di come eseguire trasformazioni geometriche su di essi.

La consapevolezza geometrica è fondamentale per la regressione accurata della posa della telecamera e la capacità di comprendere le relazioni spaziali tra gli oggetti nella scena e di eseguire trasformazioni geometriche sulle pose previste, consente alla rete di:

- Generalizzare meglio dati non visti in precedenza
- Interpretare meglio i risultati

CGAPoseNet+GCAN: Pipeline

La pipeline inizia con l'estrazione delle caratteristiche significative da un dataset di immagini tramite un modello pre-addestrato(InceptionV3).
Queste caratteristiche vengono utilizzate nel penultimo strato di InceptionV3 per generare una matrice 256×8 dove ogni riga rappresenta una proposta di motore, ovvero una possibile posa della fotocamera, e ogni colonna i suoi coefficienti.

A questo punto la GCAN elabora le proposte dei motori attraverso tre strati densi con prodotto sandwich, raffinandole progressivamente secondo la seguente funzione, fino ad ottenere il motore ottimale

$$h(M) = \sum_{i=1}^{c} W_i M_i \tilde{W}_i + B_i$$

Dove c è il numero di canali, $M = \{M_i\}_{i=1}^{c}$ è l'insieme di motori per canale, W_i rappresentano i pesi e B_i rappresentano i bias.

Inoltre la notazione maiuscola indica che $W_i, M_i, B_i \in G_{4,0}$ e che contengono un numero pari di blades. Questo implica che:

- Ogni neurone in un layer rappresenta una trasformazione geometrica del suo input preservando il grado degli oggetti
- Ogni ouput del GCAN layer è anch'esso un motore

Dettagli esperimento: dataset

Il setup sperimentale presenta un aspetto particolarmente interessante nella differenza di trattamento tra i dataset indoor e outdoor. Per i dataset outdoor, come Cambridge Landmarks, il modello è stato ri-addestrato due volte utilizzando learning rate progressivamente decrescenti, mentre per i dataset indoor, come 7 Scenes, è stato sufficiente un unico ciclo di addestramento.

Dettagli esperimento: ottimizzazione

- Durante il training, il modello è stato ottimizzato utilizzando il Mean Squared Error (MSE) come funzione di perdita, con pesi inizializzati da un modello pre-addestrato su ImageNet.
- L'addestramento, condotto per 100 epoche con batch size 64, ha impiegato l'ottimizzatore Adam con un learning rate iniziale di 10^{-4} e decadimento esponenziale per una convergenza stabile.
- Per prevenire l'overfitting, è stato adottato l'early stopping con una pazienza di 12 epoche.

Dettagli esperimento: risultati

- I risultati sperimentali confermano la superiorità del modello CGAPoseNet+GCAN rispetto alle reti PoseNet tradizionali e alle loro varianti.
- Ha registrato errori significativamente inferiori nella stima delle pose sia in termini di rotazione che di traslazione: su 13 dataset, ha ottenuto prestazioni migliori in 11 per la rotazione e in 8 per la traslazione.
- Dal punto di vista dell'efficienza, CGAPoseNet+GCAN ha ridotto il numero di parametri del modello del 17% rispetto alla versione base di CGAPoseNet, mantenendo invariato il costo computazionale

Dettagli esperimento: conclusioni

Queste caratteristiche rendono la rete particolarmente interessante non solo per la sua accuratezza, ma anche per la leggerezza e la scalabilità. Inoltre, l'integrazione tra l'output geometrico del backbone e la manipolazione esplicita tramite i GCAN layers rappresenta un'innovazione che unisce deep learning e geometria computazionale. Tale approccio conferma l'efficacia del modello e apre nuove prospettive per il miglioramento delle reti neurali in compiti di visione artificiale che richiedono una consapevolezza geometrica avanzata.

Bibliografia

- David Ruhe, Jayesh K. Gupta, Steven de Keninck, Max Welling, Johannes Brandstetter *Geometric Clifford Algebra Networks*
- Alberto Pepe, Joan Lasenby, *CGAPoseNet+GCAN: A Geometric Clifford Algebra Network for Geometry-aware Camera Pose Regression*

GATr

Autore: Alin Habasescu Marin

GATr - Introduzione

In diverse situazioni si ha a che fare con dati di natura geometrica, soprattutto in ambiti scientifici ed ingegneristici. Il vantaggio di utilizzare dati geometrici risiede nel comportamento definito dei dati sotto trasformazioni ben definite, come rotazioni e traslazioni.

Con l'obiettivo di sfruttare al meglio questi dati, si introduce il Geometric Algebra Transformer (GATr), un'architettura di rete general-purpose che sfrutta dati geometrici.

GATr - Idee Chiave

GATr integra tre concetti fondamentali:

- Algebra Geometrica
- Equivarianza
- Transformer

Per rappresentare oggetti tridimensionali e applicare rotazioni e traslazioni su di essi, la 3D GA non è sufficiente, poichè i multivettori dell'algebra tradizionale possono rappresentare solo sottospazi lineari che passano per l'origine e rotazioni intorno ad essa.

GATr consente di rappresentare i dati come multivettori dell'algebra geometrica proiettiva $\mathbb{G}_{3,0,1}$. In pratica lo spazio di interesse viene inserito in uno spazio di dimensione superiore, aggiungendo una quarta coordinata omogenea $x_0\mathbf{e}_0$, ottenendo un'algebra di dimensione $2^4 = 16$ dimensioni, capace di rappresentare vari tipi geometrici e pose $E(3)$.

Nell'algebra geometrica, un vettore $\mathbf{u}$ può agire come operatore riflettendo altri elementi nel piano ortogonale a $\mathbf{u}$. Poiché ogni trasformazione ortogonale può essere espressa come una sequenza di riflessioni, possiamo rappresentare qualsiasi trasformazione come prodotto geometrico di vettori unitari, detti *versori* $u = u_1 \cdots u_k$.

Il prodotto di versori unitari genera un gruppo, chiamato *Pin group*, in cui i versori unitari si comportano come propri inversi ($u^2 = 1$). I prodotti di un numero pari di riflessioni formano il *Spin group*. Nell'algebra geometrica proiettiva $\mathbb{G}_{3,0,1}$, questi gruppi permettono di coprire le trasformazioni $E(3)$ e $SE(3)$, rispettivamente.

Algebra Geometrica: Prodotto Sandwich

Le simmetrie tridimensionali (rotazioni, traslazioni, riflessioni) si rappresentano con i multivettori di $\mathbb{G}_{3,0,1}$. Il prodotto *sandwich* applica un versore u ad un elemento x:

$$\rho_u(x) = \begin{cases} uxu^{-1} & \text{se } u \text{ è pari} \\ ux\hat{u}^{-1} & \text{se } u \text{ è dispari} \end{cases} \tag{8}$$

dove $\hat{x}$ è l'involuzione di grado, che inverte il segno degli elementi dispari (come vettori e trivettori) e lascia invariati quelli di grado pari. Questo prodotto fornisce una rappresentazione lineare dei gruppi *Pin* e *Spin*.

Per rappresentare oggetti 3D, i piani sono descritti da vettori, e l'intersezione di due oggetti geometrici è data dal prodotto wedge delle loro rappresentazioni. Si crea una dualità tra oggetti e operatori, in cui gli oggetti si comportano come trasformazioni che li lasciano invariati.

La Tabella mostra una corrispondenza tra queste rappresentazioni, che risultano compatibili con il prodotto *sandwich* utilizzato per le trasformazioni.

Algebra Geometrica: Tabella di Oggetti e Operatori

Object / operator	Scalar	Vector		Bivector		Trivector		PS
	1	e_0	e_i	e_{0i}	e_{ij}	e_{0ij}	e_{123}	e_{0123}
Scalar $\lambda \in \mathbb{R}$	λ	0	0	0	0	0	0	0
Plane w/ normal $n \in \mathbb{R}^3$, origin shift $d \in \mathbb{R}$	0	d	n	0	0	0	0	0
Line w/ direction $n \in \mathbb{R}^3$, orthogonal shift $s \in \mathbb{R}^3$	0	0	0	s	n	0	0	0
Point $p \in \mathbb{R}^3$	0	0	0	0	0	p	1	0
Pseudoscalar $\mu \in \mathbb{R}$	0	0	0	0	0	0	0	μ
Reflection through plane w/ normal $n \in \mathbb{R}^3$, origin shift $d \in \mathbb{R}$	0	d	n	0	0	0	0	0
Translation $t \in \mathbb{R}^3$	1	0	0	$\frac{1}{2}t$	0	0	0	0
Rotation expressed as quaternion $q \in \mathbb{R}^4$	q_0	0	0	0	q_i	0	0	0
Point reflection through $p \in \mathbb{R}^3$	0	0	0	0	0	p	1	0

Figura 1: La tabella mostra la relazione tra oggetti geometrici fondamentali (come vettori, piani) e gli operatori che li trasformano. Aiuta a visualizzare come diversi tipi di oggetti interagiscono tra loro tramite operazioni geometriche.

Equivarianza

GATr è progettato per essere equivariante rispetto al gruppo di simmetria $E(3)$, che descrive le trasformazioni nello spazio tridimensionale.

A tale scopo, sono state sviluppate diverse nuove primitive $E(3)$-equivarianti, tra cui mappe lineari equivarianti, un meccanismo di attenzione, non-linearità e strati di normalizzazione.

Vengono costruiti networks layers che sono equivarianti rispetto a $E(3)$. Una funzione $f : \mathbb{G}_{3,0,1}$ è $Pin(3,0,1)$-equivariante rispetto alla rappresentazione p se

$$f(p_u(x)) = p_u(f(x))$$

per ogni $u \in Pin(3,0,1)$ e $x \in \mathbb{G}_{3,0,1}$, dove $p_u(x)$ è il sandwich product definito in (8)

Trasformer

Per il GATr si è scelto di utilizzare l'architettura Transformer grazie alle sue favorevoli proprietà di scalabilità, espressività, addestrabilità e versatilità.

Di seguito si analizza più nel dettagli la struttura di questo modello.

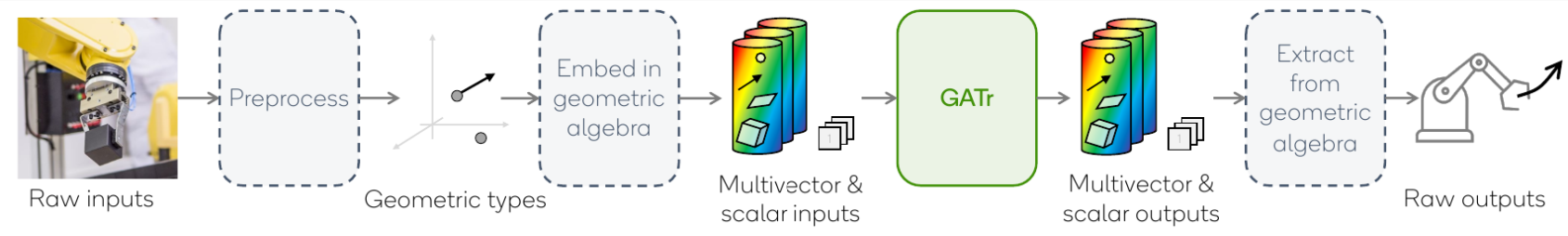
Principi di design

GATr è progettato per rappresentre vari oggetti e trasformazioni geometriche in modo efficiente, permettendo il calcolo di diverse trasformazioni utilizzando pochi strati e pochi dati.

GATr è stato progettato anche per essere simmetrico al gruppo di simmetria $E(3)$, che include traslazioni, rotazioni e riflessioni. Permette quindi trasformazioni arbitrarie sotto $E(3)$, anche rispetto a traslazione degli input, permettendo di rappresentare posizioni assolute.

GATr deve essere anche flessibile ripetto a più tipi di input geometrici, che vanno da scene statiche a serie temporali. Il meccanismo di attenzione tramite prodotto scalare, applicabile a più oggetti.

Panoramica del Flusso di Lavoro di GATr



Fasi del Flusso di Lavoro:

- Se necessario gli input grezzi vengono preprocessati in tipi geometrici.
- Gli oggetti geometrici vengono incorporati in multivettori dell'algebra geometrica $G_{3,0,1}$, come descritto nella Figura 1.
- I dati multivettoriali vengono elaborati con una rete GATr.

Architettura di GATr

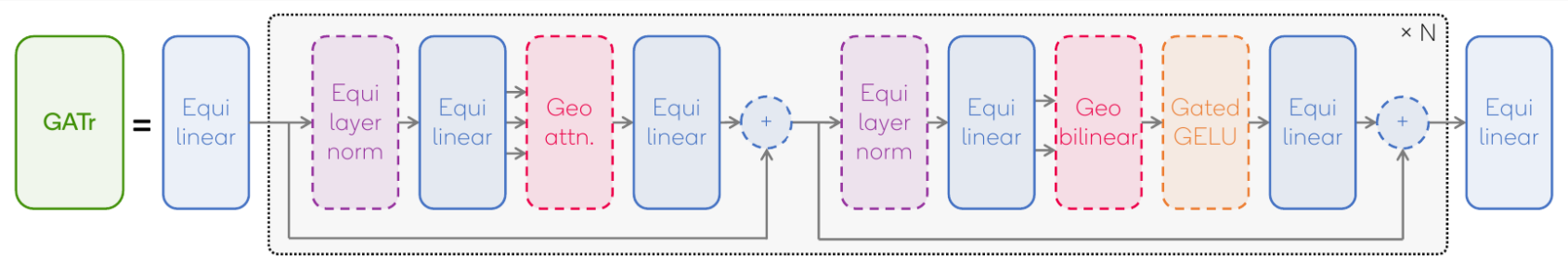


Figura 2: Architettura dettagliata di GATr.

L'architettura di GATr è composta da *N* blocchi trasformer, ognuno dei quali include: un **LayerNorm** e un **meccanismo di auto-attenzione** per multivettori equivarianti, una **connessione residua**, un altro LayerNorm, un **MLP con interazioni bilineari geometriche** e una seconda connessione residua. Dall'output di GATr si estraggono le variabili target, come illustrato in Figura 1.

LayerNorm - Layer Lineari

Per preservare l'equivarianza, qualsiasi trasformazione lineare applicata ai multivettori deve mantenere la coerenza sotto trasformazioni geometriche. Qualsiasi **trasformazione lineare** $\phi : \mathbb{G}_{d,0,1} \to \mathbb{G}_{d,0,1}$ deve essere della forma:

$$\phi(x) = \sum_{k=0}^{d+1} w_k \langle x \rangle_k + \sum_{k=0}^{d} v_k e_0 \langle x \rangle_k$$

- La trasformazione combina diverse parti del multivettore, filtrate in base al loro grado k
- w_k e v_k sono parametri apprendibili che controllano come queste parti vengono combinate.
- e_0 funge da "base omogenea" per preservare la struttura geometrica.

Prodotti Geometrici Bilineari

Per permettere alla rete di costruire nuove caratteristiche geometriche da quelle esistenti, come il vettore di traslazione tra due punti, sono necessarie due primitive aggiuntive.

- **Prodotto geometrico** $x, y \to xy$, consente di mescolare fa loro diversi tipi di componenti geometriche, come vettori e piani.
- **Join** $\text{EquiJoin}(x, y, z) = z_{0123}(x^* \wedge y^*)^*$ operazione bilineare che include il duale per rappresentare caratteristiche semplici come la distanza euclidea. Il join viene reso equivariante anche rispetto alle riflessioni, moltiplicandolo per un componente pseudoscalare di un multivettore z

Non-linearità e Normalizzazione

Nonlinearità:
Utilizziamo la **Gated GELU** per introdurre non linearità:

$$\text{GatedGELU}(x) = \text{GELU}(x_1) \cdot x$$

dove x_1 è la componente scalare di x.

Normalizzazione:
Definiamo una **LayerNorm equivariante**:

$$\text{LayerNorm}(x) = \frac{x}{\sqrt{\mathbb{E}_c \langle x, x \rangle}}$$

con il prodotto interno invariato $\langle \cdot, \cdot \rangle$.

Attenzione Multivettoriale Equivariante E(3)

Attenzione multivettoriale equivariante:
Dato un insieme di **query**, **key**, e **value** rappresentati da tensori multivettoriali, la **Attenzione E(3)-equivariant** è definita come:

$$\text{Attention}(q, k, v)_{i'c'} = \sum_i \text{Softmax}_i \left(\frac{\sum_c \langle q_{i'c}, k_{ic} \rangle}{\sqrt{8n_c}} \right) v_{ic'}$$

dove i e i' indicano gli oggetti (o token), c e c' indicano i canali, e $\langle \cdot, \cdot \rangle$ è il prodotto interno invariato dell'algebra geometrica.

Il meccanismo utilizza il prodotto interno di $G_{3,0,1}$, simile all'attenzione scalare nel trasformatore originale, ma applicato su 8 delle 16 dimensioni (escludendo le dimensioni contenenti e_0).

Estensioni dell'Architettura GATr

Diverse estensioni possono essere applicate all'architettura di GATr per aumentarne la flessibilità e l'espressività:

- **Rappresentazioni scalari ausiliarie:** Integrazione di dati non geometrici, come codifiche posizionali, con scalari aggiuntivi.
- **Attenzione basata sulla distanza:** Meccanismo di attenzione che considera la distanza euclidea tra i punti per migliorare l'espressività del modello.
- **Incorporamenti posizionali:** Utilizzo di embedding posizionali rotanti per sequenze di oggetti.
- **Attenzione assiale:** Gestione di dati su più dimensioni (oggetti, sequenze temporali) attraverso attenzione assiale.

Queste estensioni ampliano le capacità di GATr, mantenendo i

Esperimenti

Si presentano una serie di esperimenti volti a dimostrare l'efficacia di GATr in diversi ambiti. Questi esperimenti valutano le prestazioni di GATr rispetto ad altri modelli, sia equivarianti che non, in termini di accuratezza, efficienza dei dati e scalabilità.

- n-body dynamics,
- Stima dello stress di parete su grandi mesh di arterie umane,
- Pianificazione robotica tramite diffusione invariante,

n -body dynamics

- **Obiettivo:** Valutare la capacità di GATr di prevedere le posizioni finali di corpi celesti (una stella e pianeti) in un sistema gravitazionale.
- **Dataset:** Dati sintetici generati con posizioni, velocità e masse campionate da distribuzioni specifiche, con trasformazioni casuali per migliorare la generalizzazione.
- **Modelli di Confronto:**
  - Transformer e MLP
  - SEGNN, SE(3)-Transformer (modelli equivarianti)
  - GCA-GNN (non equivariante)

Risultati dell'Esperimento

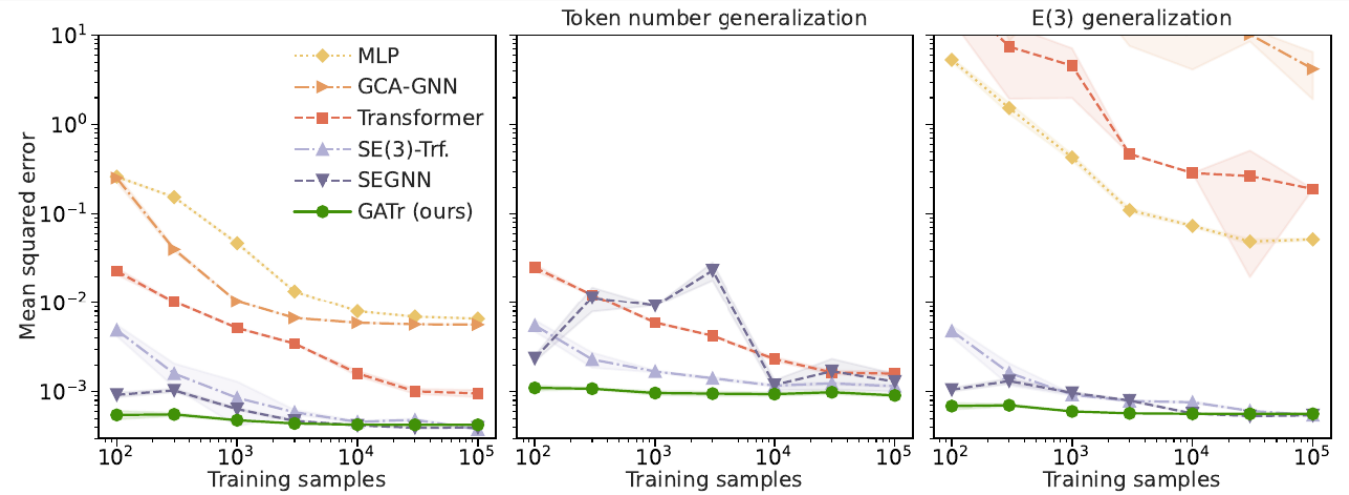


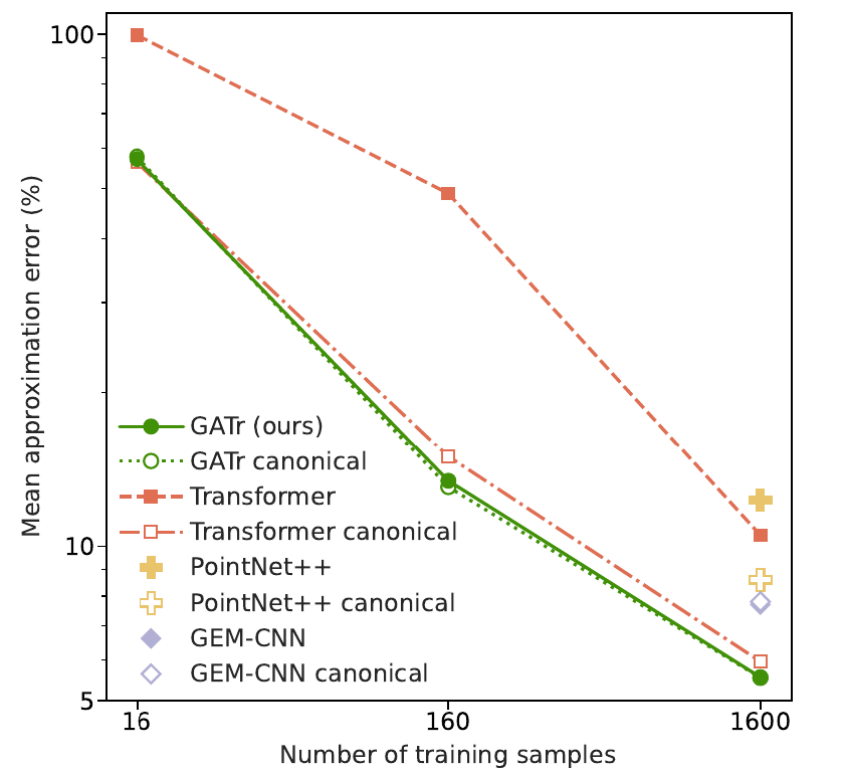
Figura 3: Errore nella previsione delle posizioni dei pianeti in funzione delle dimensioni del dataset. A sinistra: senza spostamento di distribuzione; al centro: con più pianeti di quelli addestrati; a destra: su dati traslati.

GATr supera i modelli di riferimento in efficienza dei dati, mostrando errori di predizione più bassi con meno campioni. Generalizza bene anche su sistemi con più pianeti e su dati traslati.

Stima dello Stress di Parete nelle Arterie Umane

- **Obiettivo:** Testare la scalabilità di GATr su mesh di grande dimensione per stimare lo stress di parete nelle arterie umane, importante per la formazione di aneurismi.
- **Dataset:** 2000 mesh di arterie, 1600 per l'addestramento e 400 per la valutazione.
- **Modelli di Confronto:**
  - Transformer, PointNet++
  - GEM-CNN (modello equivariante)

Risultati dell'Esperimento

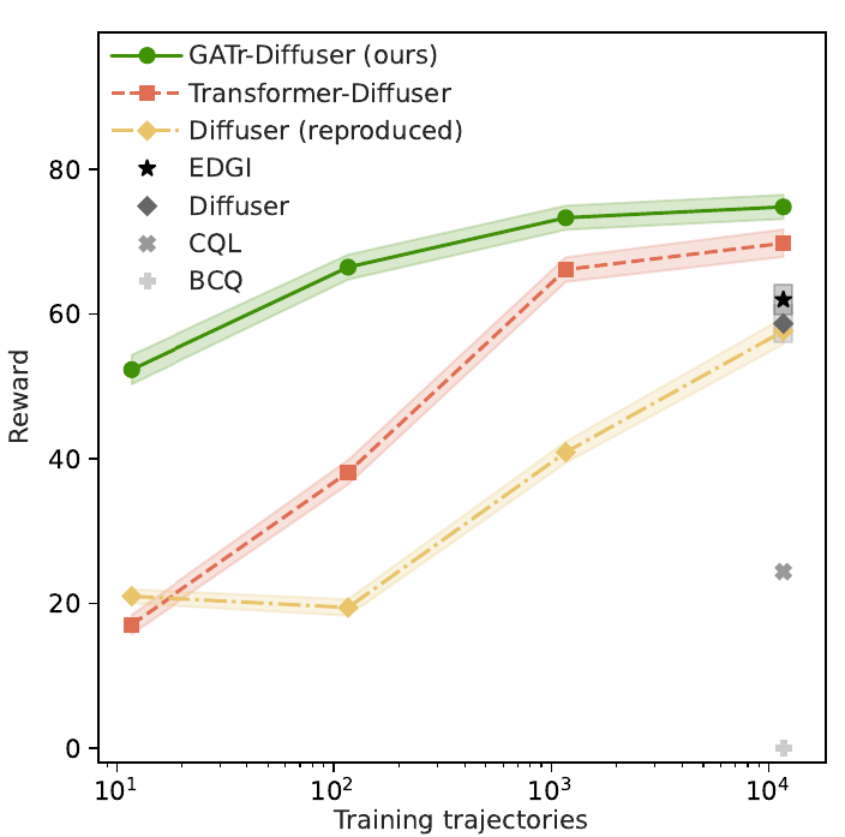


GATr supera tutti i modelli di riferimento in termini di accuratezza e di efficienza nell'utilizzo dei dati, sia su dati non canonizzati (con mesh ruotate casualmente) che su dati canonizzati (con mesh orientate nella stessa direzione).

Pianificazione Robotica tramite Diffusione Invariante

- **Obiettivo:** Usare GAT come Diffuser $E(3)$-invariante per la pianificazione robotica.
- **Compito:** Un braccio robotico Kuka deve impilare blocchi su un tavolo. Il reward è la probabilità di successo.
- **Dataset:** 11,000 dimostrazioni effettuate in precedenza
- **Modelli di Confronto:**
  - Modelli di diffusione tradizionali (Diffuser, Transformer)
  - EDGI (modello equivariante)
  - CQL e BCQ (algoritmi di apprendimento per rinforzo offline)

Risultati dell'Esperimento



GATr-Diffuser supera tutti i modelli di riferimento nel compito di impilamento dei blocchi, dimostrando una maggiore efficienza nell'utilizzo dei dati e ottenendo risultati migliori con un minor numero di traiettorie di addestramento.

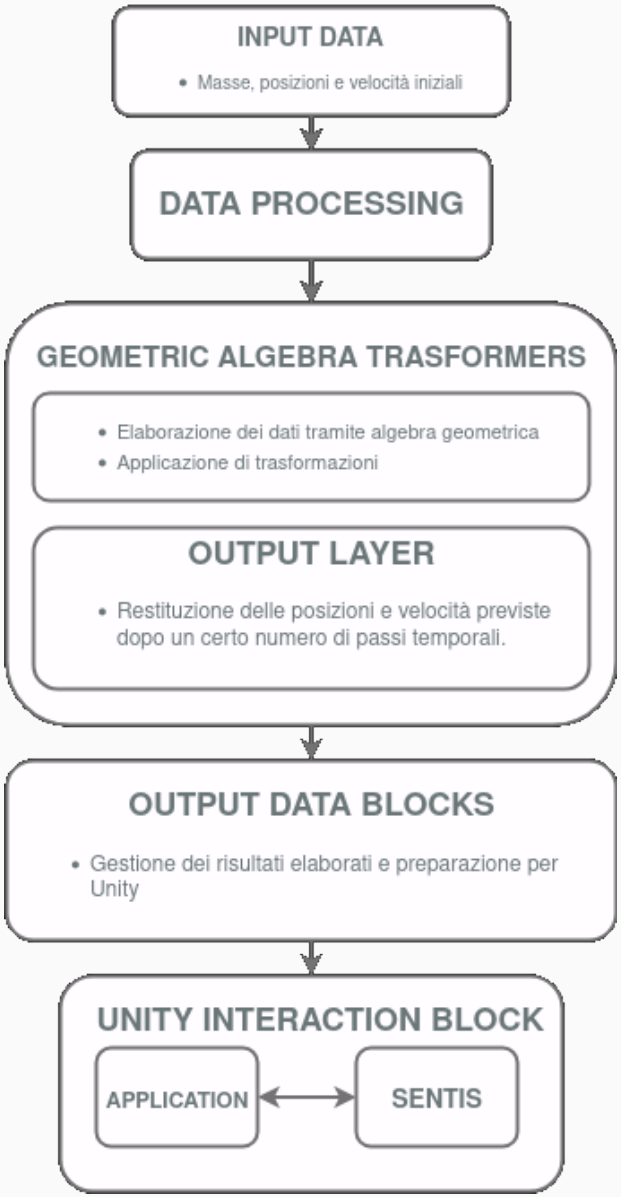
Conclusioni

- GATr si dimostra versatile ed efficace in vari contesti applicativi:
  - Previsione di sistemi dinamici complessi.
  - Gestione di grandi set di dati geometrici.
  - Pianificazione robotica più efficiente.
- GATr rappresenta un importante passo avanti nell'elaborazione di dati geometrici tramite deep learning.

Bibliografia

- Johann Brehmer, Pim de Haan, Sönke Behrends, Taco Cohen *Geometric Algebra Transformer*, Conference on Neural Information Processing Systems.

Architettura



La precedente architettura mostra come simulare e visualizzare l'N-body problem combinando una rete neurale basata su Geometric Algebra Transformer e Unity Sentis. Nel dettaglio l'architettura è composta dai seguenti moduli principali:

1. Input Data: riceve i parametri fondamentali per rappresentare il problema
2. Data Preprocessing: i dati grezzi vengono normalizzati e codificati in algebra geometrica proiettiva
3. Geometric Algebra Transformer: modella in modo efficiente le interazioni gravitazionali tra i corpi in moto e gli outliers vengono rimossi
4. Output Data Block: i dati generati dal modello vengono gestiti per permettere una corretta comunicazione con Unity
5. Unity Interaction Block: stabilisce la comunicazione bidirezionale tra la rete neurale e l'applicazione Unity

Input Data

Questo modulo riceve in input:

- le masse dei pianeti
- le coordinate spaziali dei pianeti all'inizio della simulazione
- le velocità iniziali descritte da vettori che ne indicano direzione e modulo, fornite dalla velocità di un'orbita circolare stabile.

Le posizioni iniziali dei pianeti vengono campionate su un cerchio bidimensionale attorno all'origine sul piano x-y, mentre la stella è posizionata in (0, 0) e inizialmente a riposo.

Data Preprocessing

I dati in input vengono rappresentati secondo due tipi principali di rappresentazione:

- multivettori basati sull'algebra geometrica proiettiva G(3,0,1)
- rappresentazioni scalari ausiliarie, che forniscono informazioni supplementari ai multivettori

I multivettori vengono quindi rappresentati tramite tensori a 16 componenti mentre le rappresentazioni scalari non hanno una struttura fissa ma devono comunque corrispondere nelle dimensioni di batch e numero di elementi ai multivettori a cui sono associate

Geometric Algebra Transformer

Questo modulo si occupa di applicare trasformazioni casuali e permutazioni alle posizioni fornite, in modo da generare configurazioni casuali del sistema. La posizione e velocità finale dei pianeti vengono calcolate applicando l'equazioni del moto di Newton, i campioni la cui distanza tra la posizione iniziale e finale supera un limite vengono considerati outliers e rimossi. Quindi il modulo restituisce la configurazione dei pianeti dopo l'evoluzione temporale e la traiettoria: un array 4D contenente le posizioni di tutti gli oggetti per ogni passo temporale.

Output Data Block

In questa fase le posizioni ad ogni istante temporale, codificate tramite multivettori, vengono trasformate in punti grezzi per consentire l'interfacciamento con Unity. Inoltre la rete crea un file ONNX che verrà importato in Sentis e permetterà la comunicazione tra l'applicazione e la rete neurale

Unity Interaction Block

I dati forniti dall'Output Data Block vengono utilizzati per fornire una rappresentazione 3D e interattiva dell'N-Body problem in cui l'utente può scegliere la posizione iniziale, massa e velocità dei pianeti e della stella e osservare la simulazione tramite visore