

Architectures of Topological Deep Learning

A Survey of Message-Passing Topological Neural Networks (2023)

Mathilde Papillon, Sophia Sanborn,
Mustafa Hajij, Nina Miolane

Rodrigo Fritz

7 de marzo de 2025

Contenido

Objetivo

Redes Neuronales Topológicas (TNNs)

Dominios del TDL

Vecindades

Paso de Mensajes

Resultados

Referencias

Objetivos

1. Hacer accesibles las técnicas y ventajas que ofrece el TDL.
2. Proveer una guía de las diferentes arquitecturas de TNNs, estableciendo criterios de comparación a través de una notación unificada así como de diagramas estandarizados.
3. Caracterizar las TNNs por el dominio de los datos y su modelo, el cual se especifica mediante las estructuras vecinas y su esquema pasa-mensajes.
4. Examinar el manejo de características topológicas (estructuras de orden superior) para procesar datos más allá de los enfoques tradicionales a nivel de grafos.

Objetivos

5. Identificar los retos para una evaluación comparativa con respecto a un estándar, enfatizando la dificultad de comparar las TNNs dentro y a través de diferentes dominios de aplicación. Mejorar los mecanismos de comparación es esencial para seleccionar adecuadamente los modelos y evaluar el progreso.
6. Destacar posibles direcciones de trabajo futuro: mejorar los mecanismos de evaluación comparativa, expandir el rango de tareas para las TNNs, y refinar sus fundamentos teóricos.

Contenido

Objetivo

Redes Neuronales Topológicas (TNNs)

Dominios del TDL

Vecindades

Paso de Mensajes

Resultados

Referencias

El Aprendizaje Topológico Profundo (TDL) provee un marco integral para procesar y extraer información asociado a los sistemas estudiados, como predecir la comunidad a la que pertenece un individuo en una red social, o si una proteína puede ser un target adecuado para el desarrollo de un fármaco.

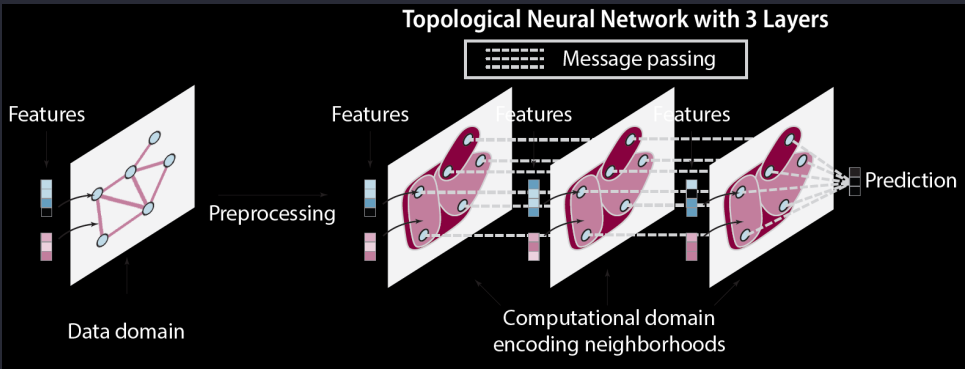
TNNs

Las Redes Neuronales Topológicas (TNNs) procesan señales definidas en dominios topológicos, los cuales generalizan las relaciones binarias de los grafos a **relaciones jerárquicas y relaciones de orden superior basadas en conjuntos**.

La flexibilidad y expresividad adicional de estas arquitecturas permite la representación de sistemas naturales complejos tales como:

- ▶ Proteínas
- ▶ Actividad neuronal
- ▶ Sistemas físicos de muchos cuerpos
- ▶ Mapas de tráfico en ciudades
- ▶ Redes de citas de artículos científicos

TNNs



Las TNNs, al igual que las Redes Neuronales en Grafos (GNNs), tienen varias capas que transforman y procesan características específicas de los datos en cada capa (Hajij et al. 2023).

Contenido

Objetivo

Redes Neuronales Topológicas (TNNs)

Dominios del TDL

Vecindades

Paso de Mensajes

Resultados

Referencias

Dominios discretos del TDL

- ▶ Complejos Simpliciales (SC): solo incluyen caras triangulares, volúmenes tetraedros, etc. Pueden introducir conexiones espurias si los datos no cumplen las condiciones geométricas específicas.
- ▶ Complejos Celulares (CC): las caras pueden ser cualquier polígono, los volúmenes cualquier poliedro, etc.
- ▶ Hipergrafos (HG): sus hiperaristas pueden conectar más de 2 nodos en relaciones tipo conjunto en las que la participación en una interacción no está implícita por cualquier otra relación en el sistema.
- ▶ Complejos Combinatorios (CCC): sus hiperaristas con rango jerárquico son más flexibles que los CC y HG. Fueron propuestos por Hajij et al. 2023.

Dominios

Traditional Discrete Domains



Set

No
Relation



Graph

Pairwise
Relations

● : Nodes

— : Edges

Domains of Topological Deep Learning



Simplicial
complex



Cellular
complex

Part-Whole
Relations

— is part of —▶



Combinatorial
complex

Set-Type
Relations

◐ not necessarily part of ◑



Hypergraph

Hajij et al. 2023

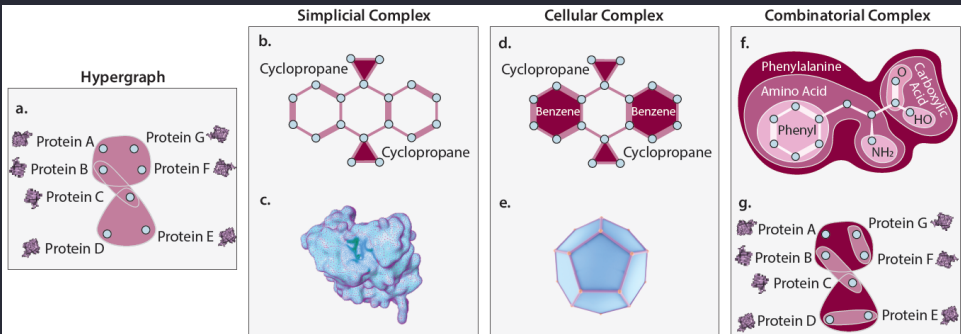


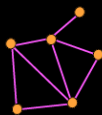
Fig. 3. Examples of Data on Topological Domains. (a) Higher-order interactions in protein networks. (b) Limited molecular representation: rings can only contain three atoms. (c) Triangular mesh of a protein surface. (d) More flexible molecular representation, permitting the representation of any ring-shaped functional group. (e) Flexible mesh which includes arbitrarily shaped faces. (f) Fully flexible molecular representation, permitting the representation of the complex nested hierarchical structure characteristic of molecules and other natural systems. (g) Hierarchical higher-order interactions in protein networks.

Dominios

Geometric deep learning domains



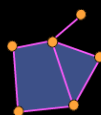
Set of entities



Graph



Simplicial complex



Cell complex



Hypergraph

● Rank-0 cells

— Rank-1 cells

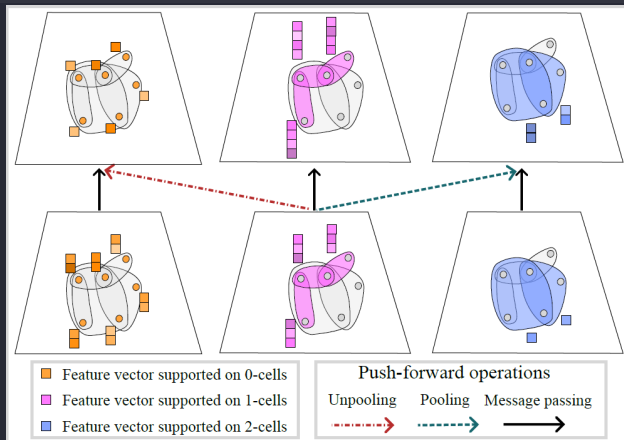
● Rank-2 cells



Combinatorial complex

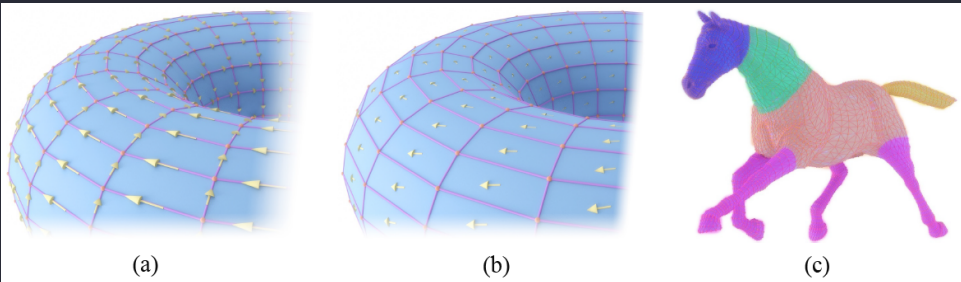
Topological Deep Learning: Going Beyond Graph Data
(*Higher-Order Attention Networks (2022)*) Hajij et al. 2023.

Dominios



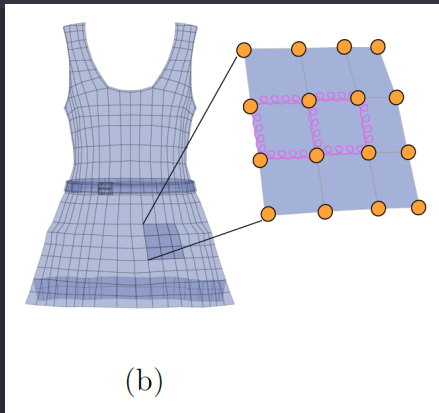
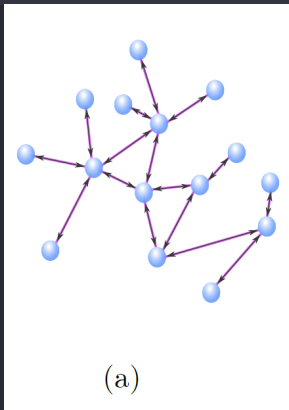
A partir de la estructura jerárquica y topológica de los CCs se aplican las operaciones push-forward en el protocolo de transmisión de mensajes de orden superior, y para las operaciones de pooling y unpooling. Estas operaciones permiten la construcción de las CCNNs.

Dominios

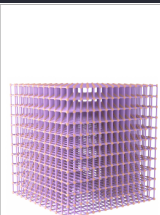


Campos vectoriales (a) en las aristas y (b) en las caras de una triangulación del toro, que es un complejo celular. (c) Etiquetas de segmentación de mallas para 2-caras. La malla también es un complejo celular.

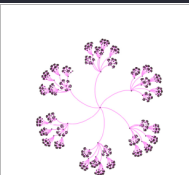
Dominios



(a) Modelo de la interacción de partículas en dinámica de fluidos mediante un grafo con transmisión de mensajes entre vértices. (b) Modelo de tensión y compresión mediante resortes en las aristas. Para modelar la interacción entre múltiples aristas, se utilizan caras poligonales para representar la geometría local, con las cuales se puede calcular la transmisión de mensajes de orden superior entre las aristas.



(a)



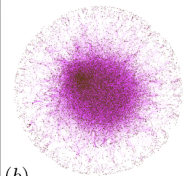
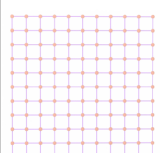
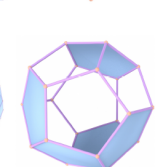
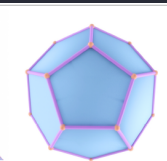
(b)



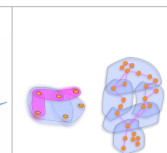
(c)



(d)



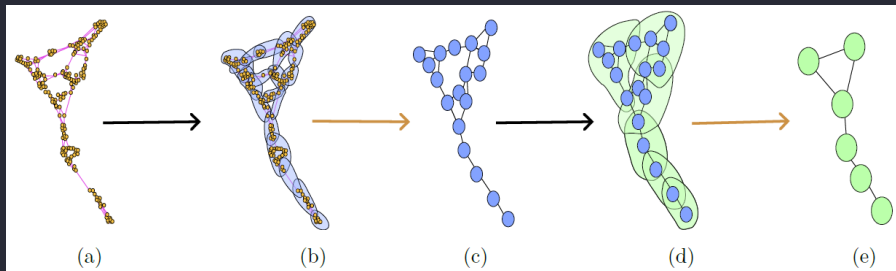
(e)



(f)

(a) secuencias e imágenes, (b) grafos, (c) figuras 3D y complejos simpliciales,
 (d) complejos cúbicos y celulares, (e) variedades discretas, (f) hipergrafos.
 (Hajij et al. 2023)

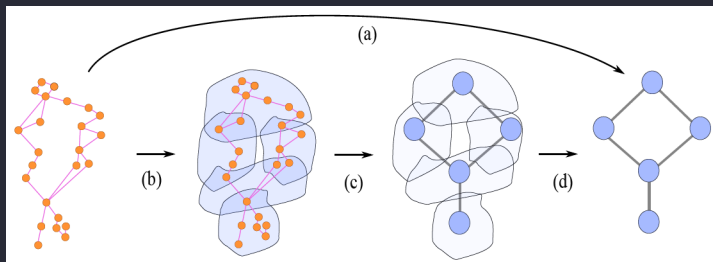
Relaciones jerárquicas de orden superior



Redes de orden superior que representan relaciones jerárquicas. Las flechas negras indican aumentación del grafo mediante relaciones de orden superior, mientras que las flechas naranjas indican la extracción del grafo engrosado.

Este proceso ilustra las relaciones jerárquicas de orden superior, relaciones entre relaciones, las cuales extraen significado y contenido, incluyendo la *forma de los datos*.

Pooling a nivel de grafos en los complejos combinatorios



La habilidad de los CCs para modelar células de tipo conjunto permite escoger la forma de los conglomerados que definen la operación de pooling, esto es muy útil para tareas específicas.

Las células con rango de orden superior en un CC corresponden a una versión más gruesa del espacio subyacente. Estructuras menos generales como los hipergrafos y los complejos celulares no permiten la selección de la forma de los conglomerados junto con la generación de representaciones más gruesas del espacio subyacente de manera simultánea.

Levantamiento

Los datos de entrada pueden ser preprocesados en un dominio computacional distinto, a este proceso se le llama levantamiento (lifting).

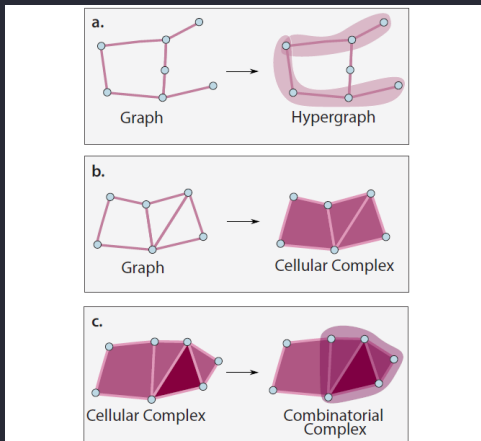
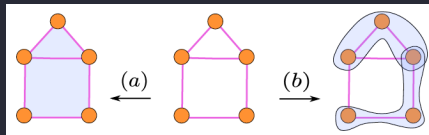
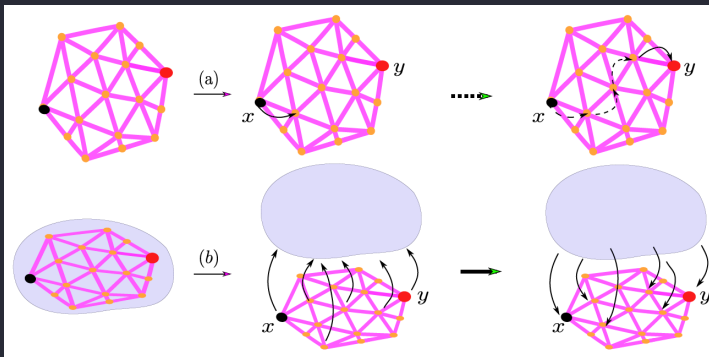


Fig. 4. **Lifting Topological Domains.** (a) A graph is lifted to a hypergraph by adding hyperedges that connect groups of nodes. (b) A graph can be lifted to a cellular complex by adding faces of any shape. (c) Hyperedges can be added to a cellular complex to lift the structure to a combinatorial complex. Figure adopted from [5].

El levantamiento es un sesgo inductivo



(a) Agregar las caras ayuda a mejorar la clasificación de datos moleculares (Bodnar, Frasca, Otter et al. 2022). (b) Considerar vecindades de 1-salto para algunos vértices y levantar así el grafo a un hipergrafo contribuye a mejorar el desempeño en tareas a nivel de vértices (Feng et al. 2019).



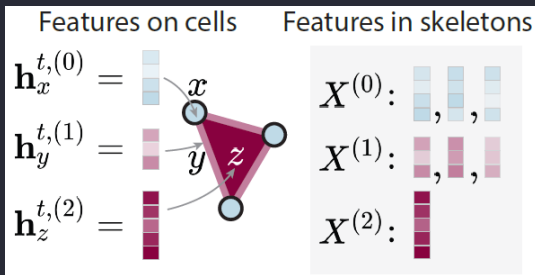
Tipos de Sesgos Inductivos

- ▶ Invarianza de traslación en CNNs. Orillas, texturas y formas lleva a detectar relaciones de más alto nivel como escenas, objetos y caras.
- ▶ Dependencia temporal en Transformers y RNNs.
- ▶ Lisura y simplicidad en la regularización e inicialización de pesos.
La regularización L2 (decaimiento de pesos) o la normalización por lotes introduce el sesgo de que pesos pequeños llevan a funciones más simples y más lisas, evitando el sobreajuste.
- ▶ Aprendizaje Jerárquico de Características.
Las redes profundas asumen que las características complejas están compuestas por características más simples, de modo que las capas más profundas extraen abstracciones de alto nivel a partir de los datos.
- ▶ Hipótesis de la Variedad en representaciones latentes de los VAEs y las GANs.
Se asume que los datos del mundo real yacen en una variedad de dimensión baja, en contraste con el espacio de dimensiones altas de los datos de entrada.

Características en un Dominio

Denotamos por $h_x^{t,(r)}$ al vector de características de la célula $x \in \mathcal{X}$ de rango r en la capa t de la TNN.

El dominio se descompone en rangos y el conjunto de todas las células de un mismo rango r se llama el r -esqueleto y se denota $X^{(r)}$.



Características en un Dominio

Las características pueden ser categóricas o cuantitativas, y pueden provenir de los datos o ser asignadas a mano con un preprocesamiento de métodos de encaje, los cuales producen vectores con las características de la célula que codifican la estructura local del espacio.

Para grafos, métodos como DeepWalk y Node2Vec se usan para encajar nodos.

Las generalizaciones recientes a dominios topológicos incluyen:

- ▶ HG: Hyperedge2Vec y Deep Hyperedge
- ▶ SC: Simplex2Vec y k-Simplex2Vec
- ▶ CC: Cell2Vec

Dominios Dinámicos y Heterogéneos

- ▶ Usualmente el dominio de los datos al pasar por las capas es estático, solo las características de los datos son transformadas. En un **dominio dinámico**, los datos cambian de una capa a otra, p.ej. pueden agregarse o quitarse nodos, las aristas pueden conectarse o desconectarse, etc.
- ▶ Un **dominio homogéneo** tiene la misma dimensión d para todos los rangos r .
- ▶ En un **dominio heterogéneo** la dimensión d_r de una característica $h_x^{(r)}$ depende del rango r de la célula x que lo soporta.

Contenido

Objetivo

Redes Neuronales Topológicas (TNNs)

Dominios del TDL

Vecindades

Paso de Mensajes

Resultados

Referencias

Vecindades

Las TNN actualizan sucesivamente las características de las células a través de sus capas mediante una noción de cercanía: la **estructura de vecindad**, la cual está definida por **relaciones de frontera** que describen cómo se relacionan entre sí células de distinto rango.

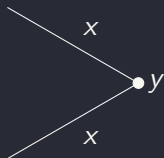
Una célula y de rango r se dice que está en la frontera de la célula x de rango R si está conectada a x y se cumple que $r < R$. Esto se denota $y \prec x$.

P.ej. un nodo conectado a una arista está en la frontera de la arista.

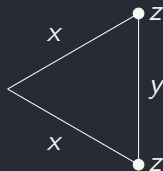
Estructuras de Vecindad



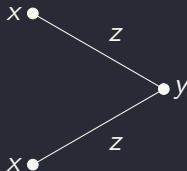
(a)



(b)



(c)



(d)

(a) Vecindad Adyacente a la Frontera: el conjunto de células x del siguiente rango más bajo que y conectadas a y .

(b) Vecindad Adyacente a la Co-Frontera: el conjunto de células x del siguiente rango más alto que y conectadas a y .


(c) Vecindad Adyacente Inferior: el conjunto de células x que comparten una frontera z con y .


(d) Vecindad Adyacente Superior: el conjunto de células x que comparten una co-frontera z con y .

Neighborhood structure



Rank 0 : 

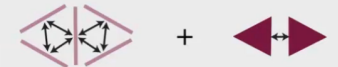
Rank 1 : 

Rank 2 : 

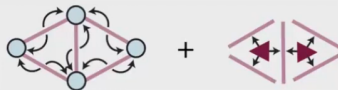
Upper adjacency: $\mathcal{N}_{A,\uparrow}$



Lower adjacency: $\mathcal{N}_{A,\downarrow}$



Boundary: $\mathcal{N}_{I,\downarrow}$



Co-Boundary: $\mathcal{N}_{I,\uparrow}$



GEOTOP-A | Make any GNN Go Topological with TopoTune | Mathilde Papillon

Vecindad Adyacente Superior: tres aristas comparten una cara como co-frontera.
Cuatro caras comparten un volumen como co-frontera.

Vecindad Adyacente Inferior: dos volúmenes comparten una cara como frontera.

Matriz de Incidencia

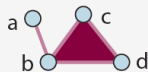
Las relaciones de frontera están codificadas en **matrices de incidencia** B_r , las cuales registran qué células x_i^{r-1} de rango $r-1$ están en la frontera de qué células x_j^r de rango r .

B_r es una matriz de tamaño $n_{r-1} \times n_r$, donde n_r es el número de células de rango $r \geq 1$ definida como:

$$(B_r)_{i,j} = \begin{cases} \pm 1 & \text{si } x_i^{r-1} \prec x_j^r \\ 0 & \text{e.o.c.} \end{cases} \quad (1)$$

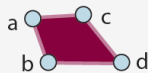
El signo de ± 1 codifica una noción de orientación que requieren los SC y los CC, mientras que siempre es $+1$ para los HG y los CCC.

Simplicial Complex



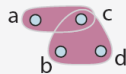
	ab	bc	bd	cd
a	-1	0	0	0
b	1	-1	-1	0
c	0	1	0	-1
d	0	0	1	1

Cellular Complex



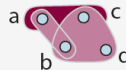
	ab	ac	bd	cd
a	-1	-1	0	0
b	1	0	-1	0
c	0	1	0	-1
d	0	0	1	1

Hypergraph



	ac	bcd
a	1	0
b	0	1
c	1	1
d	0	1

Combinatorial Complex



	ab	bcd
a	1	0
b	1	1
c	0	1
d	0	1

Contenido

Objetivo

Redes Neuronales Topológicas (TNNs)

Dominios del TDL

Vecindades

Paso de Mensajes

Resultados

Referencias

Paso de Mensajes

El **paso de mensajes** define las operaciones realizadas por una capa t de la TNN.

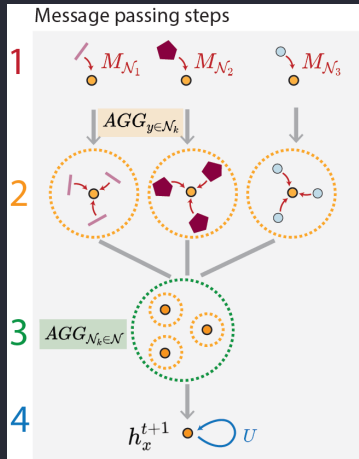
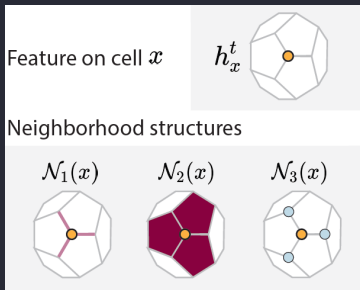
Durante el paso de mensajes, cada característica $h_x^{t,(r)}$ se actualiza con:

Las características asociadas con las células vecinas

Los parámetros aprendibles de la capa Θ^t

El vector de salida h^{t+1} de la capa t es ahora el vector de entrada de la capa $t + 1$, incorporando la información de células lejanas, mientras la información se difunde a través de la red.

Ejemplo de Paso de Mensajes



Pasos del Paso de Mensajes

The Steps of Message Passing

1. Message (red): First, a message $m_{y \rightarrow x}^{(r' \rightarrow r)}$ travels from a r' -cell y to a r -cell x through a neighborhood k of x denoted $\mathcal{N}_k(x)$:

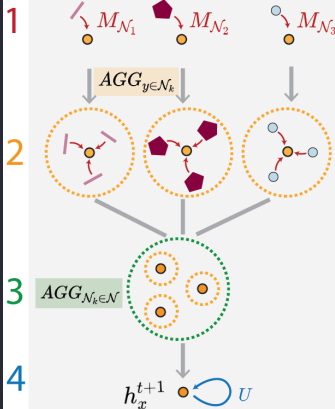
$$m_{y \rightarrow x}^{(r' \rightarrow r)} = M_{\mathcal{N}_k} \left(\mathbf{h}_x^{t,(r)}, \mathbf{h}_y^{t,(r')}, \Theta^t \right). \quad (2)$$

via the function $M_{\mathcal{N}_k}$ depicted in red in Figure 8. Here, $\mathbf{h}_x^{t,(r)}$ and $\mathbf{h}_y^{t,(r')}$ are features of dimension d_r and $d_{r'}$ on cells y and x respectively, and Θ^t are learnable parameters. In the simplest case, this step looks like a neighborhood matrix M propagating a feature $\mathbf{h}_y^{t,(r')}$ on r' -cell y to r -cell x :

$$m_{y \rightarrow x}^{(r' \rightarrow r)} = M_{xy} \cdot \mathbf{h}_y^{t,(r')} \cdot \Theta^t, \quad (3)$$

where M_{xy} is the scalar entry of matrix M at the row corresponding to cell x and column corresponding to cell y and $m_{y \rightarrow x}^{(r' \rightarrow r)}$ and Θ is a $d_{r'} \times d_r$ matrix. If y is not in the neighborhood structure of x , then M_{xy} will be 0, and x cannot receive any message from y .

Message passing steps



Pasos del Paso de Mensajes

2. Within-Nighborhood Aggregation (orange):

Next, messages are aggregated across all cells y belonging to the neighborhood $\mathcal{N}_k(x)$:

$$m_x^{(r' \rightarrow r)} = AGG_{y \in \mathcal{N}_k(x)} m_{y \rightarrow x}^{(r' \rightarrow r)}, \quad (4)$$

resulting in the *within-neighborhood aggregated message* $m_x^{(r' \rightarrow r)}$. Here, AGG is an aggregation function, depicted in orange in Figure 8, analogous to pooling in standard convolutional networks.

3. Between-Nighborhood Aggregation (green):

Then, messages are aggregated across neighborhoods in a neighborhood set \mathcal{N} :

$$m_x^{(r)} = AGG_{\mathcal{N}_k \in \mathcal{N}} m_x^{(r' \rightarrow r)}, \quad (5)$$

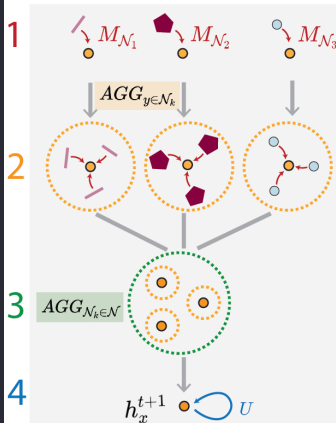
where AGG is a (potentially different) aggregation function depicted in green in Figure 8 and $m_x^{(r)}$ is the message received by cell x that triggers the update of its feature.

4. Update (blue): Finally, the feature on cell x is updated via a function U depicted in blue in Figure 8, which may depend on the previous feature $\mathbf{h}_x^{t,(r)}$ on cell x :

$$\mathbf{h}_x^{t+1,(r)} = U(\mathbf{h}_x^{t,(r)}, m_x^{(r)}), \quad (6)$$

The result $\mathbf{h}_x^{t+1,(r)}$ is the updated feature on cell x that is input to layer $t + 1$.

Message passing steps



Los tres tipos de GNN

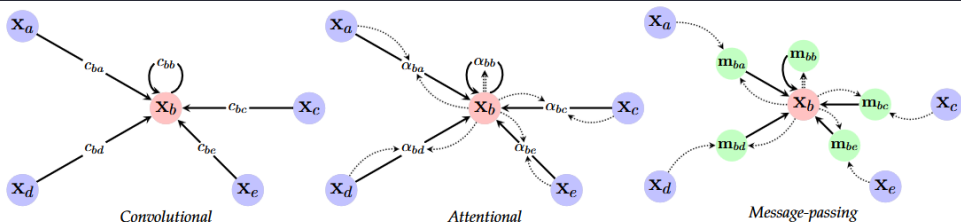
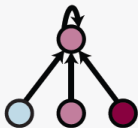
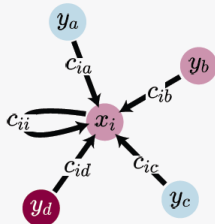


Figure 17: A visualisation of the dataflow for the three flavours of GNN layers, g . We use the neighbourhood of node b from Figure 10 to illustrate this. Left-to-right: **convolutional**, where sender node features are multiplied with a constant, c_{uv} ; **attentional**, where this multiplier is *implicitly* computed via an attention mechanism of the receiver over the sender: $\alpha_{uv} = a(\mathbf{x}_u, \mathbf{x}_v)$; and **message-passing**, where vector-based messages are computed based on both the sender and receiver: $\mathbf{m}_{uv} = \psi(\mathbf{x}_u, \mathbf{x}_v)$.

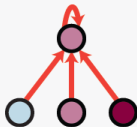
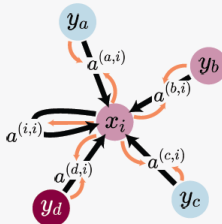
Bronstein et al. 2021

Funciones Pasa-Mensajes

Standard Convolutional



Attentional Convolutional



General

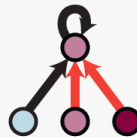
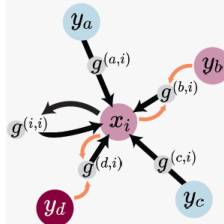


Diagrama de Tensores

Step	Purpose	Tensor diagram
$m_{y \rightarrow x}^{(r' \rightarrow r)}$	Defines message sent from cell y to cell x (ranks of cells, type of message, and neighborhood matrix).	<p>Feature on cell x, rank 0 at layer $t + 1$</p> <p>Feature on cell y, rank 0 at layer t</p> <p>Feature on cell y, rank 1 at layer t</p> <p>Feature on cell y, rank 1 at layer t</p>
$m_x^{(r' \rightarrow r)}$	Defines aggregation of messages received by cell x per neighborhood k .	<p>Sums over all messages from this neighborhood</p> <p>Uses general function to combine messages from this neighborhood</p>
$m_x^{(r)}$	Defines aggregation of k neighborhood-specific messages into one message.	<p>Combine messages from both neighborhoods via a sum</p> <p>Combine messages from both neighborhoods via a general function</p>
$h_x^{t+1, (r)}$	Defines update of the feature on cell x .	<p>Update function</p> <p>Update function, depends on cell's feature at layer t</p>

Fig. 9. **Tensor Diagrams:** a graphical notation for the four steps of a message passing scheme. A diagram depicts how a feature on cell y at layer t , $\mathbf{h}_y^{(t)}$, becomes a feature on cell x at layer $t + 1$, $\mathbf{h}_x^{(t+1)}$.

Contenido

Objetivo

Redes Neuronales Topológicas (TNNs)

Dominios del TDL

Vecindades

Paso de Mensajes

Resultados

Referencias

Análisis de las TNN

Arquitecturas

Tareas:

- ▶ Nodos: clasificación, regresión, aglomeración
- ▶ Aristas: clasificación, predicción de enlaces
- ▶ Complejos: clasificación de hiperaristas

Simetrías:

- ▶ Invarianza Permutaciones
- ▶ Equivarianza Orientación
- ▶ Localidad Simplicial
- ▶ Cardinalidad de las hiperaristas

Resultados para Complejos Celulares (CC) y Complejos Combinatorios (CCC)

Domain	Model	Task Level			Task Purpose	Comparisons
		<i>Node</i>	<i>Edge</i>	<i>Complex</i>		
CC	CWN [8]		✓	✓	Classification, prediction, regression	GNN SOTA
	CAN [79]			✓	Classification	GNN SOTA
CCC	HOAN* [11]		✓	✓	Classification	GNN SOTA

CWN = CW Networks (Bodnar, Frasca, Otter et al. 2022).

CAN = Cell Attention Networks (Giusti et al. 2022).

HOAN = Higher-Order Attention Networks (Hajij et al. 2023).

* indica que no se compartió una implementación.

SOTA = state of the art.

Resultados con Complejos Simpliciales

Domain	Model	Task Level			Task Purpose	Comparisons
		<i>Node</i>	<i>Edge</i>	<i>Complex</i>		
SC	MPSN [7]	✓	✓		Classification, Trajectory Classification	GNN SOTA
	SCCONV [66]			✓	Classification	Graph
	BScNet [13]	✓			Link prediction	GNN SOTA
	SNN [65]	✓			Imputation	None
	SAN [73]	✓			Classification, Trajectory Classification	TNN SOTA
	SAT [74]	✓	✓		Classification, Trajectory Classification	TNN SOTA
	HSN* [72]	✓	✓	✓	Classification, Link prediction, Vector embedding	Graph
	SCA* [76]			✓	Clustering	Graph
	Dist2Cycle [71]	✓			Homology Localization	GNN SOTA
	SGAT [75]	✓			Classification	GNN SOTA
	SCoNe [68]	✓			Trajectory Classification	TNN SOTA
	SCNN* [62]	✓			Imputation	TNN SOTA
	SCCNN [70]	✓			Link prediction, Trajectory Classification	TNN SOTA
	SCN [67]	✓			Classification	TNN SOTA

Resultados con Hipergrafos

Domain	Model	Task Level			Task Purpose	Comparisons
		<i>Node</i>	<i>Edge</i>	<i>Complex</i>		
HG	HyperSage [47]	✓			Classification (Inductive + Transductive)	GNN SOTA
	AllSet [51]	✓			Classification	TNN SOTA
	HyperGat [54]	✓			Classification	GNN SOTA
	HNHN [10]	✓	✓		Classification, Dimensionality Reduction	GNN SOTA
	HMPNN* [31]	✓			Classification	TNN SOTA
	UniGNN [9]	✓			Classification (Inductive + Transductive)	TNN SOTA
	DHGNN [53]	✓			Classification (Multimodal)	GNN SOTA
	EHNN [52]	✓			Classification, Keypoint Matching	TNN SOTA
	HHNN [55]	✓			Link prediction	TNN SOTA
	HTNN [56]	✓			Classification	TNN SOTA
	SHARE* [36]	✓			Prediction	GNN SOTA
	DHGCN* [49]			✓	Classification	GNN SOTA
	HGC-RNN* [48]	✓			Prediction	GNN SOTA

- El dominio de los hipergrafos se ha beneficiado de los avances en las GNNs, en particular en los dominios dinámicos (estimación de poses, modelado de tránsito ferroviario, predicción de coautoría). Sin embargo, otros dominios discretos aún no han explorado el dinamismo. Además, fuera del dominio de los hipergrafos, la mayoría de las TNNs se implementan como redes homogéneas, lo que deja espacio para generalizaciones heterogéneas y no euclidianas.

- **Going Deeper.** El sobrealisamiento (oversmoothing) ocurre cuando una red es muy eficiente en la colección de señales a través de múltiples capas. Esto produce características muy similares en todas las células y un desempeño deficiente en la tarea de aprendizaje posterior. Mientras que esto ha llamado la atención en la comunidad de las GNNs, poco de este trabajo se ha generalizado para las TNNs, lo que ocasiona que permanezcan poco profundas.

UniGCNII (Huang y Yang 2021) consiste en una TNN de 64 capas que generaliza en hipergrafos las soluciones para el sobrealisamiento en GNNs (Chen et al. 2020).

Las High Skip Networks (HSNs) generalizan las conexiones residuales (skip connections) pero están implementadas en redes poco profundas.

Sobreaplastamiento

- ▶ En contraste, el sobreaplastamiento (oversquashing) se puede evitar empleando arquitecturas de GNNs jerárquicas o multiescala para distribuir la carga del aprendizaje.
- ▶ También se pueden utilizar mecanismos de atención en los nodos importantes para reducir una acumulación innecesaria de mensajes.
- ▶ Modificar la topología del grafo es también una buena opción, se pueden agregar nodos virtuales, reconectar aristas, o dispersar el grafo para aliviar los cuellos de botella.
- ▶ Otra opción es utilizar redes más amplias, incrementando las dimensiones ocultas de los vectores de características para que la red pueda aprender características más complejas y así no perder la información a través de las capas.

Contenido

Objetivo

Redes Neuronales Topológicas (TNNs)

Dominios del TDL





Vecindades

Paso de Mensajes





Resultados

Referencias

Referencias (Parte 1)

-  Bodnar, Cristian (2022). «Topological Deep Learning: Graphs, Complexes, Sheaves». Tesis doct. Apollo - University of Cambridge Repository. DOI: 10.17863/CAM.97212.
-  Bodnar, Cristian, Fabrizio Frasca, Nina Otter et al. (2022). *Weisfeiler and Lehman Go Cellular: CW Networks*. arXiv: 2106.12575 [cs.LG] (vid. págs. 21, 42).
-  Bodnar, Cristian, Fabrizio Frasca, Yu Guang Wang et al. (2021). *Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks*. arXiv: 2103.03212 [cs.LG].
-  Hajij, Mustafa et al. (2023). *Topological Deep Learning: Going Beyond Graph Data*. arXiv: 2206.00606 [cs.LG] (vid. págs. 8, 10-13, 17, 42).

Referencias (Parte 2)

-  Chen, Ming et al. (2020). *Simple and Deep Graph Convolutional Networks*. arXiv: 2007.02133 [cs.LG] (vid. pág. 46).
-  Feng, Yifan et al. (jul. de 2019). «Hypergraph Neural Networks». En: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01, págs. 3558-3565. DOI: 10.1609/aaai.v33i01.33013558 (vid. pág. 21).
-  Giusti, Lorenzo et al. (2022). *Cell Attention Networks*. arXiv: 2209.08179 [cs.LG] (vid. pág. 42).
-  Huang, Jing y Jie Yang (2021). *UniGNN: a Unified Framework for Graph and Hypergraph Neural Networks*. arXiv: 2105.00956 [cs.LG] (vid. pág. 46).