

# Homología persistente como herramienta de análisis de redes neuronales

José Manuel Ros Rodrigo

Facultad de Ciencia y Tecnología  
Universidad de La Rioja

22/07/2022

## «Topological measurement of deep neural networks using persistent homology»

Annals of Mathematics and Artificial Intelligence  
<https://doi.org/10.1007/s10472-021-09761-3>



### Topological measurement of deep neural networks using persistent homology

Satoru Watanabe<sup>1</sup> · Hayato Yamana<sup>1</sup>

Accepted: 10 June 2021 / Published online: 03 July 2021  
© The Author(s) 2021

#### Abstract

The inner representation of deep neural networks (DNNs) is indecipherable, which makes it difficult to tune DNN models, control their training process, and interpret their outputs. In this paper, we propose a novel approach to investigate the inner representation of DNNs through topological data analysis (TDA). Persistent homology (PH), one of the outstanding methods in TDA, was employed for investigating the complexities of trained DNNs. We constructed clique complexes on trained DNNs and calculated the one-dimensional PH of DNNs. The PH reveals the combinational effects of multiple neurons in DNNs at different resolutions, which is difficult to be captured without using PH. Evaluations were conducted using fully connected networks (FCNs) and networks combining FCNs and convolutional neural networks (CNNs) trained on the MNIST and CIFAR-10 data sets. Evaluation results demonstrate that the PH of DNNs reflects both the excess of neurons and problem difficulty, making PH one of the prominent methods for investigating the inner representation of DNNs.

**Keywords** Deep neural network · Convolutional neural network · Persistent Homology · Topological data analysis

**Mathematics Subject Classification (2010)** 68T07 · 55N31

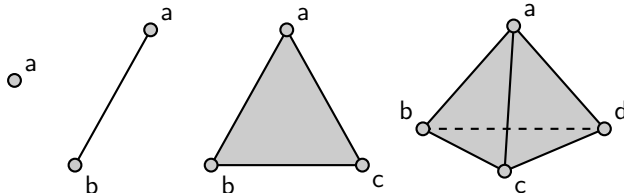
- 1 Complejos simpliciales, homología, y homología persistente
- 2 Redes neuronales
- 3 El artículo: definiciones y resultados
- 4 Las dos interpretaciones: un ejemplo
- 5 Interpretación global
- 6 Interpretación local
- 7 Conclusiones

# Complejos simpliciales

## Definición

Un **complejo simplicial abstracto** es una colección  $\mathcal{V}$ , de subconjuntos no vacíos de un conjunto  $\mathcal{V}_0$ , que verifica las siguientes propiedades:

- a) Si  $v \in \mathcal{V}_0$ , entonces  $\{v\} \in \mathcal{V}$ .
- b) Si  $\sigma \in \mathcal{V}$  y  $\tau \subset \sigma$ , entonces  $\tau \in \mathcal{V}$ .



### Definición

Una **aplicación simplicial entre complejos simpliciales**,  $f: \mathcal{V} \rightarrow \mathcal{V}'$ , es una aplicación tal que  $f(\sigma) = \{g(u_1), g(u_2), \dots, g(u_k)\} = \{v_1, v_2, \dots, v_k\}$ ; donde  $g: \mathcal{V}_0 \rightarrow \mathcal{V}'_0$  es una aplicación entre 0-símplices,  $\sigma = \{u_1, u_2, \dots, u_k\} \in \mathcal{V}$  y  $\{v_1, v_2, \dots, v_k\} \in \mathcal{V}'$ .

# Homología

## Definición

Sea  $R$  un anillo. Decimos que un **complejo de cadenas** sobre  $R$  es un conjunto  $\mathcal{C}_* = \{(C_p, \partial_p) \mid p \in \mathbb{Z}\}$  de  $R$ -módulos y  $R$ -homomorfismos  $\{\partial_p: C_p \rightarrow C_{p-1} \mid p \in \mathbb{Z}\}$ , que satisfacen la siguiente relación:  $\partial_p \circ \partial_{p+1} = 0$ . Se denota por  $(\mathcal{C}_*, \partial)$  y a  $\partial$  se le llama el **diferencial** del complejo.

## Definición

Sea  $(\mathcal{C}_*, \partial)$  un complejo de cadenas sobre  $R$  anillo. Definimos el  **$p$ -ésimo  $R$ -módulo de homología** de  $\mathcal{C}_*$  como el cociente:

$$H_p(\mathcal{C}_*) := \ker \partial_p / \operatorname{im} \partial_{p+1}.$$

# Homología

## Definición

Sea  $R$  un anillo. Decimos que un **complejo de cadenas** sobre  $R$  es un conjunto  $\mathcal{C}_* = \{(C_p, \partial_p) \mid p \in \mathbb{Z}\}$  de  $R$ -módulos y  $R$ -homomorfismos  $\{\partial_p: C_p \rightarrow C_{p-1} \mid p \in \mathbb{Z}\}$ , que satisfacen la siguiente relación:  $\partial_p \circ \partial_{p+1} = 0$ . Se denota por  $(\mathcal{C}_*, \partial)$  y a  $\partial$  se le llama el **diferencial** del complejo.

## Definición

Sea  $(\mathcal{C}_*, \partial)$  un complejo de cadenas sobre  $R$  anillo. Definimos el  **$p$ -ésimo  $R$ -módulo de homología** de  $\mathcal{C}_*$  como el cociente:

$$H_p(\mathcal{C}_*) := \ker \partial_p / \operatorname{im} \partial_{p+1}.$$

Complejo de cadenas  $\longrightarrow$  Diferencial del complejo  $\longrightarrow$  Homología simplicial

# Complejo de cadenas

## Definición

Sea  $R$  un anillo y  $\mathcal{V}$  un complejo simplicial abstracto. Definimos el  **$R$ -módulo de  $p$ -cadenas** de  $\mathcal{V}$  como el conjunto de todas las sumas formales de  $p$ -símplices  $\mathcal{V}$ , con la operación suma componente a componente con coeficientes en  $R$ , y el producto por escalares por elementos de  $R$ . Lo denotaremos por  $C_p(\mathcal{V})$ .



# Diferencial del complejo

## Definición

Sea  $R$  un anillo,  $\mathcal{V}$  un complejo simplicial abstracto,  $\sigma \in \mathcal{V}$  y  $\sigma = \{u_0, \dots, u_p\}$  un  $p$ -símplice escogido de manera ordenada. Definimos el **operador borde para un símplice** como:

$$B_p(\sigma) := \sum_{j=0}^p (-1)^j \{u_0, \dots, \widehat{u}_j, \dots, u_p\}.$$

La suma anterior es una suma formal, donde  $\widehat{u}_j$  indica que omitimos  $u_j$ . A continuación definimos el **homomorfismo borde**:

$$\begin{aligned} \partial_p: C_p(\mathcal{V}) &\rightarrow C_{p-1}(\mathcal{V}) \\ c = \sum a_i \sigma_i &\mapsto \partial_p(c) = \sum a_i B_p(\sigma_i) \end{aligned} \tag{1.1}$$

donde  $a_i \in R$ . Notemos que, si escogemos  $R = \mathbb{Z}_2$ , no es necesaria la alternancia de signos ni el uso de coeficientes escalares.

## Teorema

Sea  $\partial_p$  definida como en la expresión 1.1. Entonces para todo  $p \in \mathbb{N} \cup \{0\}$ ,  
 $\partial_p \circ \partial_{p+1} = 0$ .

# Homología simplicial

## Definición

Sea  $R$  un anillo,  $p \in \mathbb{N} \cup \{0\}$  y  $\mathcal{V}$ , un complejo simplicial. Definimos el **p-ésimo  $R$ -módulo de homología simplicial** del complejo simplicial  $\mathcal{V}$  como el  $R$ -módulo cociente  $\ker \partial_p / \operatorname{im} \partial_{p+1}$ , donde  $\partial_p$  está definida como en la expresión 1.1. Lo denotaremos por  $H_p(\mathcal{V})$ .

## Definición

Sea  $H_p(\mathcal{V})$  el  $p$ -ésimo  $\mathbb{Z}_2$ -espacio vectorial de homología de un complejo simplicial  $\mathcal{V}$ . A su dimensión,  $\dim H_p(\mathcal{V}) = \dim \ker \partial_p - \dim \operatorname{im} \partial_{p+1}$ , la llamaremos **p-ésimo número de Betti**, y la denotaremos por  $\beta_p(\mathcal{V})$ .

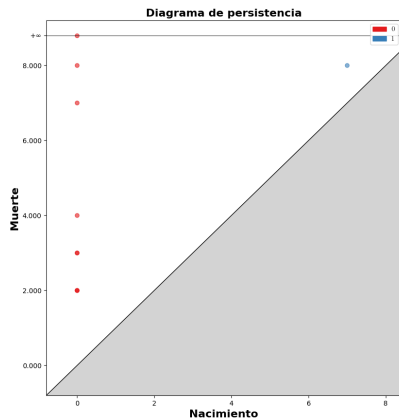
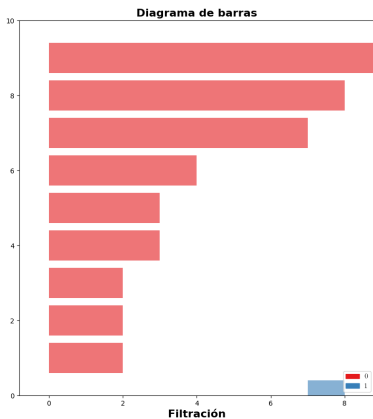
# Homología persistente

## Definición

Sea  $\mathcal{V}$  un complejo simplicial abstracto finito. Consideremos la secuencia  $\mathcal{V}^1 \subset \mathcal{V}^2 \subset \dots \subset \mathcal{V}^{k-1} \subset \mathcal{V}^k = \mathcal{V}$  de subcomplejos simpliciales cualesquiera de  $\mathcal{V}$ . A  $\mathcal{V}$  junto con su secuencia de subcomplejos simpliciales encajados lo llamaremos **complejo simplicial filtrado**.

## Definición

Sea  $\mathcal{V}^1 \subset \mathcal{V}^2 \subset \dots \subset \mathcal{V}^{k-1} \subset \mathcal{V}^k = \mathcal{V}$  un complejo simplicial filtrado. Definimos los **p-ésimos espacios vectoriales de homología persistente** como las imágenes de los homomorfismos inducidos por la inclusión,  $H_p^{i,j} := \text{im } f_p^{i,j}$ , con  $0 \leq i \leq j \leq k$ . A su dimensión,  $\dim H_p^{i,j}$ , la llamaremos **p-ésimo número de Betti persistente** y la denotaremos por  $\beta_p^{i,j}$ .



- 1 Complejos simpliciales, homología, y homología persistente
- 2 Redes neuronales**
- 3 El artículo: definiciones y resultados
- 4 Las dos interpretaciones: un ejemplo
- 5 Interpretación global
- 6 Interpretación local
- 7 Conclusiones

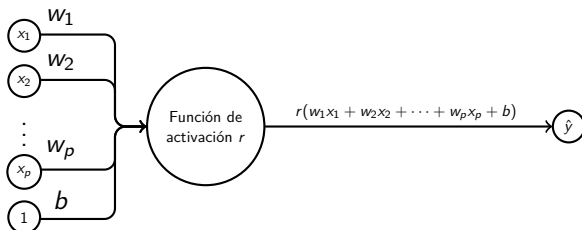
# Redes neuronales

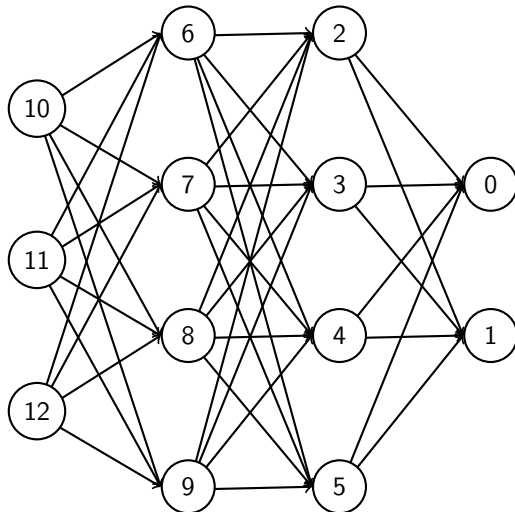
$$\hat{y} = r(w_1x_1 + w_2x_2 + \cdots + w_px_p + b)$$

$$MSE(w_1, w_2, \cdots, w_p, b) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$w_i^{siguiente} = w_i^{actual} - \alpha \frac{\partial MSE(w, b)}{\partial w_i}$$

$$b^{siguiente} = b^{actual} - \alpha \frac{\partial MSE(w, b)}{\partial b}$$







- 1 Complejos simpliciales, homología, y homología persistente
- 2 Redes neuronales
- 3 El artículo: definiciones y resultados**
- 4 Las dos interpretaciones: un ejemplo
- 5 Interpretación global
- 6 Interpretación local
- 7 Conclusiones

# Definiciones y resultados

## Definición

Definimos la **importancia** de  $u_i$  (salida) para  $u_j$  (llegada) como:

$$R_{ij} := \begin{cases} 1 & \text{si } i = j \\ w_{ij}^+ / \sum_{k, k \neq j} w_{kj}^+ & \text{si } i \neq j \end{cases}$$

## Definición

Definimos la **importancia extendida** entre neuronas, y la denotamos por  $\overline{R}_{ij}$ , como:

$$\overline{R}_{ij} := \max\{R_{u_i u_{m_1}} \cdots R_{u_{m_n} u_j} \mid (u_i, u_{m_1}, \dots, u_{m_n}, u_j) \in C_{ij}\} \quad (3.1)$$

donde  $C_{ij}$  denota el conjunto de todos los posibles caminos de  $u_i$  a  $u_j$ .

## Definición

Sea  $0 \leq t \leq 1$  un parámetro real. Definimos los  **$p$ -símplices asociados a un DAG con pesos** a partir del conjunto de vértices del DAG como sigue:

$$\mathcal{V}_p^t := \begin{cases} \mathcal{V}_0 & \text{si } p = 0 \\ \{\{u_{a_0}, \dots, u_{a_p}\} \mid u_{a_i} \in \mathcal{V}_0, \overline{R_{a_i a_j}} \geq t, \forall a_i > a_j\} & \text{si } p \geq 1 \end{cases} \quad (3.2)$$

donde cada  $u_{a_i}$  pertenece al nivel  $a_i$  del DAG y  $a_0 < \dots < a_i < \dots < a_p$ .

## Proposición

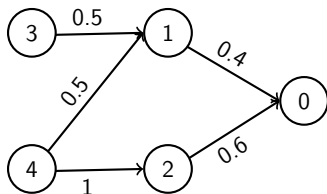
Sea  $\mathcal{V}_0 = \{u_0, \dots, u_n\}$  un conjunto finito, y  $\{w_{ij}\}_{0 \leq j \leq i \leq n}$  un conjunto de números reales. Sea  $\overline{R_{ij}}$  la importancia entre neuronas definida en 3.1, y  $\mathcal{V}_p^t$  los  $p$ -símplices definidos en 3.2 con  $t$  parámetro real entre 0 y 1. Entonces  $\mathcal{V}^t = \bigcup_{s=0}^{s=n} \mathcal{V}_s^t$  es un complejo simplicial abstracto.

## Proposición

Sea  $(t_i)_{i=1}^n$  una sucesión, monótona decreciente, de números reales entre 1 y 0, e indexada sobre los naturales. Entonces  $\mathcal{V}_0 = \emptyset$  y  $\mathcal{V}_i = \mathcal{V}^{t_i}$  con  $1 \leq i \leq n$ , es un complejo simplicial filtrado.

- 1 Complejos simpliciales, homología, y homología persistente
- 2 Redes neuronales
- 3 El artículo: definiciones y resultados
- 4 Las dos interpretaciones: un ejemplo**
- 5 Interpretación global
- 6 Interpretación local
- 7 Conclusiones

# Interpretación global frente a interpretación local



Calcularemos  $\mathcal{V}_2^{0.4}$ . Listamos los posibles 2-símplices:

$$\{4, 2, 0\}, \{4, 1, 0\}, \{3, 1, 0\}$$

## ■ Interpretación local

$$\mathcal{V}_2^{0.4} = \{\{4, 2, 0\}\}$$

## ■ Interpretación global

$$\mathcal{V}_2^{0.4} = \{\{4, 2, 0\}, \{4, 1, 0\}\}$$

- 1 Complejos simpliciales, homología, y homología persistente
- 2 Redes neuronales
- 3 El artículo: definiciones y resultados
- 4 Las dos interpretaciones: un ejemplo
- 5 Interpretación global**
- 6 Interpretación local
- 7 Conclusiones

# Interpretación global

- 1) Se calculan recursivamente todos los posibles caminos para un vértice de origen.



# Interpretación global

- 1) Se calculan recursivamente todos los posibles caminos para un vértice de origen.
- 2) Se calcula la importancia de cada camino, es decir, la importancia entre el vértice de origen y de destino de la siguiente manera: se van multiplicando las importancias entre cada par de vértices; y si dos caminos comparten un vértice intermedio, entonces la importancia hasta ese vértice será la máxima producida por ambos caminos hasta ese vértice. La importancia obtenida tras esta multiplicación será la importancia del camino.

# Interpretación global

- 1) Se calculan recursivamente todos los posibles caminos para un vértice de origen.
- 2) Se calcula la importancia de cada camino, es decir, la importancia entre el vértice de origen y de destino de la siguiente manera: se van multiplicando las importancias entre cada par de vértices; y si dos caminos comparten un vértice intermedio, entonces la importancia hasta ese vértice será la máxima producida por ambos caminos hasta ese vértice. La importancia obtenida tras esta multiplicación será la importancia del camino.
- 3) Habiendo calculado las importancias de cada camino, se descartan aquellos caminos que no superan el umbral fijado.

# Interpretación global

- 1) Se calculan recursivamente todos los posibles caminos para un vértice de origen.
- 2) Se calcula la importancia de cada camino, es decir, la importancia entre el vértice de origen y de destino de la siguiente manera: se van multiplicando las importancias entre cada par de vértices; y si dos caminos comparten un vértice intermedio, entonces la importancia hasta ese vértice será la máxima producida por ambos caminos hasta ese vértice. La importancia obtenida tras esta multiplicación será la importancia del camino.
- 3) Habiendo calculado las importancias de cada camino, se descartan aquellos caminos que no superan el umbral fijado.
- 4) Como cada camino constituye un símplex maximal, se calculan todos los subsímplices asociados al camino.

```

def __calcula_simplices_global(self,t)->List[List[int]]:
    simp=[]
    P=self.filtracion(t)
    T=P.clausura_transitiva()
    for i in range(len(self.M)):
        for j in T.G.adj[i]:
            if(self.M[i][j]==0):
                p=[]
                self.busca_pesos(i,j,p)
                if(len(p)>0):
                    m=max([v[0] for v in p])
                    if(m>=t):
                        l = [k[1] for k in p]
                        simp.extend(l)
                        simp.append([i,j])
            elif (self.M[i][j]>=t):
                simp.append([i,j])
    return simp

```

**BC:** Función para el cálculo de simplices maximales (**global**).

```

def busca_pesos(self, o, d, p=[], q=1, c=[]):
    if(o==d):
        c.append(o)
        p.append((q.__round__(4),c))
        q=1
        c=[]
    else:
        for j in range(d,o):
            if(self.M[o][j]!=0):
                self.busca_pesos(j,d,p,q*self.M[o][j],c+[o])

```

**BC:** Función para la búsqueda de pesos.

```

def get_simplex_global(self, t : float)->List[List[int]]:
    simplices=self.__calcula_simplices_global(t)
    simplices.extend([[i] for i in range(len(self.M))])
    for s in simplices:
        for L in range(3,len(s)+1):
            for sub in it.combinations(s,L):
                if(list(sub) not in simplices):
                    simplices.append(list(sub))
    tr=[s for s in simplices if len(s)>2]
    for e in tr:
        if(any([list(sub) not in simplices for sub in
                it.combinations(e,len(e)-1)])):
            simplices.remove(e)
    return simplices

```

**BC:** Función para el cálculo de subsímplices (global).

- 1 Complejos simpliciales, homología, y homología persistente
- 2 Redes neuronales
- 3 El artículo: definiciones y resultados
- 4 Las dos interpretaciones: un ejemplo
- 5 Interpretación global
- 6 Interpretación local**
- 7 Conclusiones

# Interpretación local

- 1) Se calculan recursivamente todos los posibles caminos para un vértice de origen.

# Interpretación local

- 1) Se calculan recursivamente todos los posibles caminos para un vértice de origen.
- 2) Se calcula la importancia de cada camino multiplicando la importancia entre cada par vértices, sin tener en cuenta si existe otro camino que comparta vértices.



# Interpretación local

- 1) Se calculan recursivamente todos los posibles caminos para un vértice de origen.
- 2) Se calcula la importancia de cada camino multiplicando la importancia entre cada par vértices, sin tener en cuenta si existe otro camino que comparta vértices.
- 3) Al igual que en la interpretación global, se descartan aquellos caminos que no superan el umbral fijado.

# Interpretación local

- 1) Se calculan recursivamente todos los posibles caminos para un vértice de origen.
- 2) Se calcula la importancia de cada camino multiplicando la importancia entre cada par vértices, sin tener en cuenta si existe otro camino que comparta vértices.
- 3) Al igual que en la interpretación global, se descartan aquellos caminos que no superan el umbral fijado.
- 4) Se calculan todos los subsímplices asociados a cada camino.

```

def __calcula_simplices(self,t)->List[List[int]]:
    simp=[]
    P=self.filtracion(t)
    T=P.clausura_transitiva()
    for i in range(len(self.M)):
        for j in T.G.adj[i]:
            if(self.M[i][j]==0):
                p=[]
                self.busca_pesos(i,j,p)
                if(len(p)>0):
                    l = [k[1] for k in p if k[0]>=t]
                    simp.extend(l)
            elif (self.M[i][j]>=t):
                simp.append([i,j])
    return simp

```

**BC:** Función para el cálculo de símplexes maximales (**local**).

```

def get_simplex_local(self, t : float)->List[List[int]]:
    simplices=self.__calcula_simplices(t)
    simplices.extend([[i] for i in range(len(self.M))])
    for s in simplices:
        for L in range(2,len(s)+1):
            for sub in it.combinations(s,L):
                if(list(sub) not in simplices):
                    simplices.append(list(sub))
    return simplices

```

**BC:** Función para el cálculo de subsímplexes (**local**).

- 1 Complejos simpliciales, homología, y homología persistente
- 2 Redes neuronales
- 3 El artículo: definiciones y resultados
- 4 Las dos interpretaciones: un ejemplo
- 5 Interpretación global
- 6 Interpretación local
- 7 Conclusiones

## Conclusiones

## Conclusiones

¡Gracias por su atención!