

Categorías y Funtores: Mapa Jerárquico y Diagramas

con `tikz-cd`

(con ejemplos para LLMs, Multi-Head y TDA)

Índice

1. Introducción	2
2. Fundamentos: categoría, conmutatividad y funtores	2
2.1. Definición de categoría	2
2.2. Diagrama conmutativo (el cuadrado básico)	2
2.3. Definición de funtor (covariante)	2
2.4. Funtor contravariante	2
3. Mapa jerárquico de tipos de categorías	3
3.1. Mapa visual (alto nivel)	3
3.2. Árbol (familias principales)	3
3.3. Notas mínimas por familia	3
4. Diagramas listos para ML: LLM, TDA, multi-head, adjunciones	4
4.1. LLM como composición (Set → Vect → Prob)	4
4.2. Naturaleza de “comutar” en el pipeline	4
4.3. LLM + TDA (vector topológico)	4
4.4. Funtor de olvido (Metric → Top)	5
4.5. Adjunción (Free \dashv Forget) como diagrama	5
4.6. Multi-Head: suma directa de cabezas y proyección	5
4.7. Estructura monoidal: coherencia con tensor	6
4.8. Optimizador como endomorfismo (dinámica de entrenamiento)	6
4.9. Diagrama completo “stack ML moderno” (resumen)	6
5. Tips prácticos de <code>tikz-cd</code>	6
6. Cierre	6

1. Introducción

Este documento reúne un mapa jerárquico de tipos de categorías y, sobre todo, un conjunto de diagramas conmutativos listos en `tikz-cd` para modelar pipelines tipo LLM, multi-head attention y TDA.

2. Fundamentos: categoría, conmutatividad y funtores

2.1. Definición de categoría

Definición 1. Una *categoría* \mathcal{C} consiste en:

- una clase de objetos $\text{Ob}(\mathcal{C})$;
- para cada par A, B , un conjunto de morfismos $\text{Hom}_{\mathcal{C}}(A, B)$;
- composición $\circ : \text{Hom}(B, C) \times \text{Hom}(A, B) \rightarrow \text{Hom}(A, C)$;
- identidades $\text{id}_A \in \text{Hom}(A, A)$,

sujeto a asociatividad e identidad.

2.2. Diagrama comutativo (el cuadrado básico)

Un cuadrado

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ g \downarrow & & \downarrow h \\ C & \xrightarrow{k} & D \end{array}$$

commuta si

$$h \circ f = k \circ g.$$

2.3. Definición de funtor (covariante)

Definición 2. Un *funtor* $F : \mathcal{C} \rightarrow \mathcal{D}$ asigna:

- a cada objeto $A \in \mathcal{C}$ un objeto $F(A) \in \mathcal{D}$;
- a cada morfismo $f : A \rightarrow B$ un morfismo $F(f) : F(A) \rightarrow F(B)$;

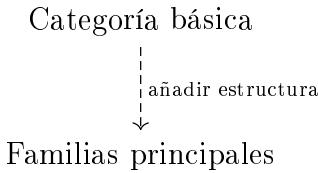
de modo que $F(\text{id}_A) = \text{id}_{F(A)}$ y $F(g \circ f) = F(g) \circ F(f)$.

2.4. Funtor contravariante

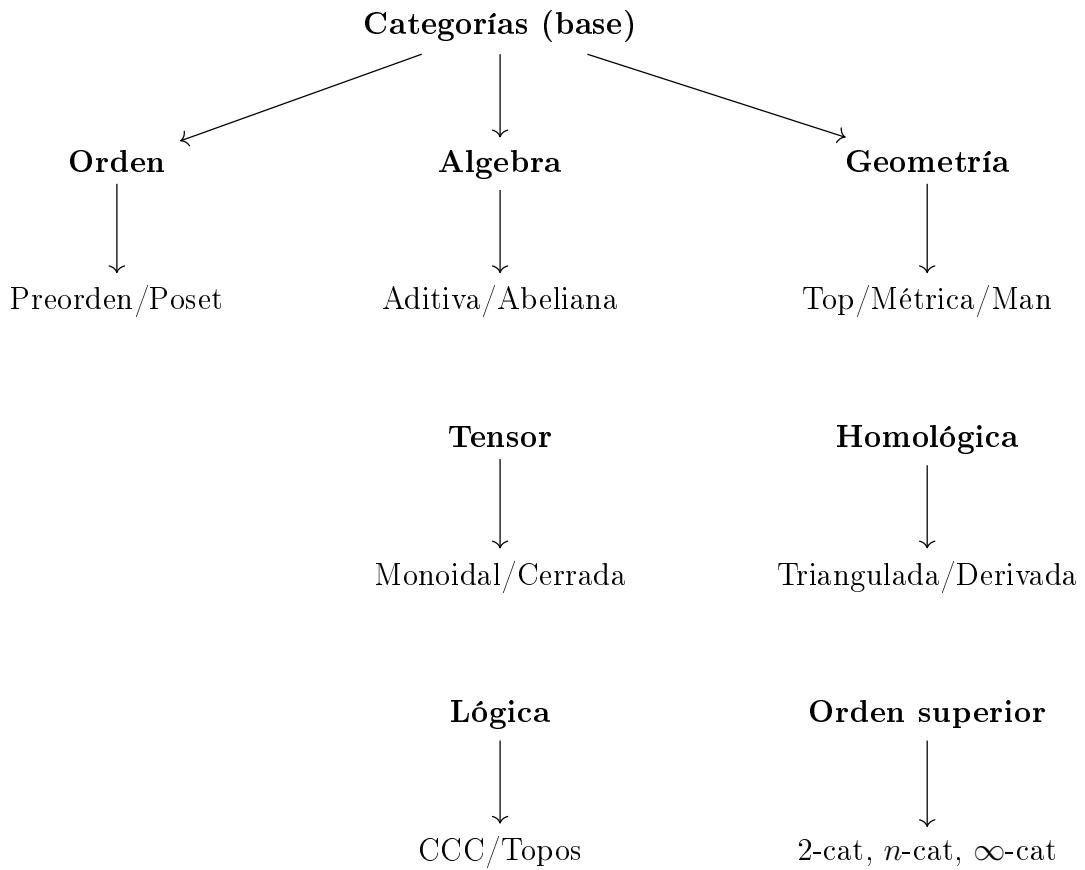
Un funtor contravariante es $F : \mathcal{C}^{\text{op}} \rightarrow \mathcal{D}$, es decir invierte flechas.

3. Mapa jerárquico de tipos de categorías

3.1. Mapa visual (alto nivel)



3.2. Árbol (familias principales)



3.3. Notas mínimas por familia

- **Orden**: categorías delgadas (preorden/poset).
- **Álgebra**: aditivas/abelianas (núcleos, cokernels, exactitud).
- **Tensor**: monoidales (producto \otimes), monoidales cerradas (hom interno).
- **Geometría**: Top, Metric, Man.

- **Homológica:** simpliciales, trianguladas, derivadas.
- **Lógica:** CCC y topos.
- **Orden superior:** 2-categorías, n -categorías, ∞ -categorías.

4. Diagramas listos para ML: LLM, TDA, multi-head, adjunciones

4.1. LLM como composición ($\mathbf{Set} \rightarrow \mathbf{Vect} \rightarrow \mathbf{Prob}$)

Interpretación estándar:

$$LLM = Dec \circ T \circ Emb.$$

Diagrama:

$$\mathbf{Set} \xrightarrow{\text{Emb}} \mathbf{Vect} \xrightarrow{T} \mathbf{Vect} \xrightarrow{\text{Dec}} \mathbf{Prob}$$

Con la composición explícita:

$$\begin{array}{ccccccc} & & & \text{LLM} & & & \\ & \nearrow \text{Emb} & & \curvearrowright & & \searrow \text{Dec} & \\ \mathbf{Set} & \xrightarrow{\text{Emb}} & \mathbf{Vect} & \xrightarrow{T} & \mathbf{Vect} & \xrightarrow{\text{Dec}} & \mathbf{Prob} \end{array}$$

4.2. Naturaleza de “conmutar” en el pipeline

Si tienes una transformación $f : x \rightarrow y$ en \mathbf{Set} , el requisito functorial dice:

$$LLM(f) = Dec(T(Emb(f))).$$

Diagrama:

$$\begin{array}{ccc} x & \xrightarrow{f} & y \\ \downarrow LLM & & \downarrow LLM \\ LLM(x) & \xrightarrow{LLM(f)} & LLM(y) \end{array}$$

4.3. LLM + TDA (vector topológico)

Pipeline típico:

$$\mathbf{Vect} \rightarrow \mathbf{Metric} \rightarrow \mathbf{SimpComp} \rightarrow \mathbf{Ab} \rightarrow \mathbf{Vect}.$$

Diagrama:

$$\mathbf{Vect} \xrightarrow{d \text{ (métrica)}} \mathbf{Metric} \xrightarrow{VR_\varepsilon \text{ (Rips)}} \mathbf{SimpComp} \xrightarrow{H_k \text{ (homología)}} \mathbf{Ab} \xrightarrow{\Phi \text{ (vectorizar)}} \mathbf{Vect}$$

4.4. Funtor de olvido ($\text{Metric} \rightarrow \text{Top}$)

La métrica induce topología (olvidas distancias exactas, conservas abiertos):

$$\text{Metric} \xrightarrow{\text{Forget}} \text{Top}$$

4.5. Adjunción ($\text{Free} \dashv \text{Forget}$) como diagrama

Una forma compacta de recordarlo:

$$\text{Hom}_{\text{Vect}}(\text{Free}(S), V) \cong \text{Hom}_{\text{Set}}(S, \text{Forget}(V)).$$

Diagrama:

$$\text{Set} \begin{array}{c} \xrightarrow{\text{Free}} \\[-1ex] \xleftarrow{\text{Forget}} \end{array} \text{Vect}$$

Y se anota $\text{Free} \dashv \text{Forget}$.

4.6. Multi-Head: suma directa de cabezas y proyección

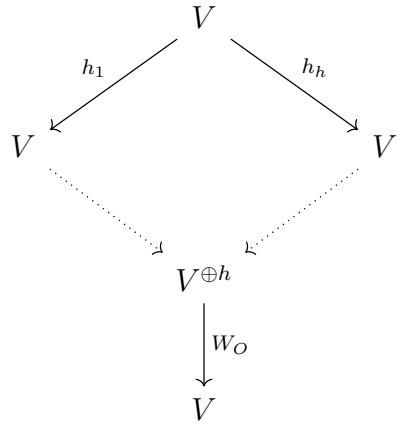
Modelado estructural:

$$V \xrightarrow{\Delta} V^{\oplus h} \xrightarrow{W_O} V, \quad \Delta(x) = (h_1(x), \dots, h_h(x)).$$

Diagrama:

$$V \xrightarrow{\Delta=h_1 \oplus \dots \oplus h_h} V^{\oplus h} \xrightarrow{W_O} V$$

También puedes dibujar “fork” (una cabeza arriba y otra abajo, sugerencia visual):



(Las punteadas solo indican “agrupación” en la suma directa.)

4.7. Estructura monoidal: coherencia con tensor

Si T fuera monoidal (idealización), esperas compatibilidad:

$$T(V \otimes W) \cong T(V) \otimes T(W).$$

Diagrama (como cuadrado de coherencia):

$$\begin{array}{ccc} V \otimes W & \xrightarrow{T} & T(V \otimes W) \\ \cong \downarrow & & \downarrow \text{estructura monoidal} \\ V \otimes W & \xrightarrow[T(V) \otimes T(W)]{} & T(V) \otimes T(W) \end{array}$$

4.8. Optimizador como endomorfismo (dinámica de entrenamiento)

Un optimizador define una iteración:

$$\theta_{t+1} = \Phi(\theta_t), \quad \Phi : \Theta \rightarrow \Theta.$$

Diagrama:

$$\Theta \xrightarrow{\Phi} \Theta \xrightarrow{\Phi} \Theta \xrightarrow{\Phi} \dots$$

Si Θ es variedad (Man) o espacio vectorial (Vect) depende de tu formalización.

4.9. Diagrama completo “stack ML moderno” (resumen)

Uniendo idea de embeddings, geometría y TDA:

$$\text{Set} \xrightarrow{\text{Emb}} \text{Vect} \xrightarrow{\text{Transformer}} \text{Vect} \xrightarrow{d} \text{Metric} \xrightarrow{VR_\varepsilon} \text{SimpComp} \xrightarrow{H_k} \text{Ab} \xrightarrow{\Phi} \text{Vect}$$

5. Tips prácticos de tikz-cd

- Direcciones: `r,l,u,d,dr,dl,ur,ul`.
- Etiqueta al otro lado: "`f`".
- Curvar flecha: `bend left=20` o `bend right=20`.
- Punteada: `dashed`.
- Ajustar espacios: `[row sep=large, column sep=huge]`.
- Mono/Epi: `hookrightarrow`, `twoheadrightarrow`.

6. Cierre

LLM como funtor compuesto, TDA como cadena de funtores, multi-head como suma directa + proyección, adjunciones (Free \dashv Forget), coherencia monoidal y optimización como dinámica.