Categorías y Funtores: Mapa Jerárquico y Diagramas con `tikz-cd`

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

Índice

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* $\mathrm{Ob}(\mathcal{C})$*;*
- *para cada par* A, B*, un conjunto de morfismos* $\mathrm{Hom}_{\mathcal{C}}(A, B)$*;*
- *composición* $\circ : \mathrm{Hom}(B, C) \times \mathrm{Hom}(A, B) \to \mathrm{Hom}(A, C)$*;*
- *identidades* $\mathrm{id}_A \in \mathrm{Hom}(A, A)$*,*

sujeto a asociatividad e identidad.

2.2. Diagrama conmutativo (el cuadrado básico)

Un cuadrado

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "h"] \\ C \arrow[r, "k"'] & D \end{tikzcd}$$

conmuta si

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

2.3. Definición de funtor (covariante)

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

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

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

2.4. Funtor contravariante

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

3. Mapa jerárquico de tipos de categorías

3.1. Mapa visual (alto nivel)

$$
\begin{tikzcd}
\text{Categoría básica} \arrow[d, dashed, "\text{añadir estructura}"] \\
\text{Familias principales}
\end{tikzcd}
$$

3.2. Árbol (familias principales)

$$
\begin{tikzcd}
 & \textbf{Categorías (base)} \arrow[dl] \arrow[d] \arrow[dr] & \\
\textbf{Orden} \arrow[d] & \textbf{Algebra} \arrow[d] & \textbf{Geometría} \arrow[d] \\
\text{Preorden/Poset} & \text{Aditiva/Abeliana} & \text{Top/Métrica/Man} \\
 & \textbf{Tensor} \arrow[d] & \textbf{Homológica} \arrow[d] \\
 & \text{Monoidal/Cerrada} & \text{Triangulada/Derivada} \\
 & \textbf{Lógica} \arrow[d] & \textbf{Orden superior} \arrow[d] \\
 & \text{CCC/Topos} & \text{2-cat, } n\text{-cat, } \infty\text{-cat}
\end{tikzcd}
$$

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 (Set → Vect → Prob)

Interpretación estándar:

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

Diagrama:

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

Con la composición explícita:

$$\begin{tikzcd}
\mathbf{Set} \arrow[r, "Emb"] \arrow[rrr, bend left, "LLM"] & \mathbf{Vect} \arrow[r, "T"] & \mathbf{Vect} \arrow[r, "Dec"] & \mathbf{Prob}
\end{tikzcd}$$

4.2. Naturaleza de "conmutar" en el pipeline

Si tienes una transformación $f : x \to y$ en **Set**, el requisito funtorial dice:

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

Diagrama:

$$\begin{tikzcd}
x \arrow[r, "f"] \arrow[d, "LLM"'] & y \arrow[d, "LLM"] \\
LLM(x) \arrow[r, "LLM(f)"'] & LLM(y)
\end{tikzcd}$$

4.3. LLM + TDA (vector topológico)

Pipeline típico:

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

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 (Metric → Top)

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

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

4.5. Adjunción (Free ⊣ Forget) como diagrama

Una forma compacta de recordarlo:

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

Diagrama:

$$\begin{tikzcd} \mathbf{Set} \arrow[r, shift left, "Free"] & \mathbf{Vect} \arrow[l, shift left, "Forget"] \end{tikzcd}$$

Y se anota $Free \dashv Forget$.

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

Modelado estructural:

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

Diagrama:

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

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

$$\begin{tikzcd} & V \arrow[dl, "h_1"'] \arrow[dr, "h_h"] & \\ V \arrow[dr, dotted] & & V \arrow[dl, dotted] \\ & V^{\oplus h} \arrow[d, "W_O"] & \\ & V & \end{tikzcd}$$

(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):

$$\begin{tikzcd}
V \otimes W \arrow[r, "T"] \arrow[d, "\cong"'] & T(V \otimes W) \arrow[d, dashed, "\text{estructura monoidal}"] \\
V \otimes W \arrow[r, dashed, "T(V)\otimes T(W)"'] & T(V) \otimes T(W)
\end{tikzcd}$$

4.8. Optimizador como endomorfismo (dinámica de entrenamiento)

Un optimizador define una iteración:

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

Diagrama:

$$\begin{tikzcd}
\Theta \arrow[r, "\Phi"] & \Theta \arrow[r, "\Phi"] & \Theta \arrow[r, "\Phi"] & \cdots
\end{tikzcd}$$

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:

$$\begin{tikzcd}
\mathbf{Set} \arrow[r, "Emb"] & \mathbf{Vect} \arrow[r, "T\ \text{(Transformer)}"] & \mathbf{Vect} \arrow[r, "d"] & \mathbf{Metric} \arrow[r, "VR_\varepsilon"] & \mathbf{SimpComp} \arrow[r, "H_k"] & \mathbf{Ab} \arrow[r, "\Phi"] & \mathbf{Vect}
\end{tikzcd}$$

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.