Funtores y su Aplicación Conceptual en Inteligencia Artificial

Equipo de Ingeniería

1. Introducción

En este documento presentamos una explicación intuitiva y formal del concepto de **funtor** desde teoría de categorías, junto con una interpretación aplicada al contexto de modelos de Inteligencia Artificial (IA), especialmente embeddings y redes neuronales.

El objetivo es mostrar que muchas arquitecturas modernas pueden entenderse como estructuras que preservan composición, lo cual es precisamente la esencia de un funtor.

2. Recordatorio: ¿Qué es una categoría?

Una categoría $\mathcal{C}$ está compuesta por:

- Objetos
- Morfismos (flechas) entre objetos

Cumpliendo:

- Existencia de identidad: $\mathrm{id}_X : X \to X$
- Composición asociativa: si $f : X \to Y$ y $g : Y \to Z$, entonces existe $g \circ f : X \to Z$

3. Definición Formal de Funtor

Un **funtor covariante**

$$F : \mathcal{C} \to \mathcal{D}$$

asigna:

- A cada objeto $X \in \mathcal{C}$ un objeto $F(X) \in \mathcal{D}$
- A cada morfismo $f : X \to Y$ un morfismo $F(f) : F(X) \to F(Y)$

Preservando:

$$F(\mathrm{id}_X) = \mathrm{id}_{F(X)}$$

$$F(g \circ f) = F(g) \circ F(f)$$

Es decir, un funtor preserva identidad y composición.

4. Modelo Conceptual Aplicado a IA

Consideremos dos categorías:

4.1. Categoría de Datos $\mathcal{D}\dashv\sqcup\dashv$

- Objetos: conjuntos de datos (texto, imágenes, audio)
- Morfismos: transformaciones válidas (tokenización, normalización, data augmentation)

4.2. Categoría Vectorial $\mathcal{V}\rceil\rfloor\sqcup$

- Objetos: espacios vectoriales $\mathbb{R}^n$
- Morfismos: transformaciones lineales (capas neuronales)

5. El Embedding como Funtor

Definimos:

$$F : \mathcal{D}\dashv\sqcup\dashv \to \mathcal{V}\rceil\rfloor\sqcup$$

Tal que:

- A cada dataset D le asigna un embedding $F(D) \subseteq \mathbb{R}^n$
- A cada transformación $T : D_1 \to D_2$ le asigna una transformación lineal $F(T)$

5.1. Preservación de Composición

Supongamos:

$$T_1 : D \to D'$$
$$T_2 : D' \to D''$$

Entonces:

$$F(T_2 \circ T_1) = F(T_2) \circ F(T_1)$$

Esto refleja exactamente cómo funcionan las redes neuronales:

$$\text{Input} \longrightarrow \text{Embedding} \longrightarrow W_1 \longrightarrow \sigma \longrightarrow W_2$$

Cada capa es composición de transformaciones.

6. Interpretación en Redes Neuronales

Una red neuronal puede entenderse como la composición:

$$f(x) = W_k\sigma(W_{k-1}\sigma(\dots W_1x))$$

La arquitectura respeta estructura composicional:

$$F(g \circ f) = F(g) \circ F(f)$$

Es decir, el modelo completo es composición de morfismos en $\mathcal{V}\rceil\rfloor\sqcup$.

7. Caso Especial: Graph Neural Networks

Podemos definir:

$$F : \mathbf{Graph} \to \mathbf{Vect}$$

Donde:

- Objetos: grafos
- Morfismos: homomorfismos de grafos
- Imagen: embeddings del grafo

Una GNN puede verse como un funtor que transforma estructura relacional en representación vectorial preservando composición estructural.

8. Conclusión

Un funtor es una estructura que:

- Traduce objetos entre categorías
- Traduce transformaciones
- Preserva identidad
- Preserva composición

Muchos modelos modernos de IA pueden interpretarse como funtores:

- Embeddings
- Redes neuronales profundas
- GNNs
- Modelos kernelizados

Esta perspectiva permite:

- Comprender mejor la composición de modelos
- Diseñar arquitecturas más estructurales
- Conectar IA con teoría matemática avanzada