

# Otras arquitecturas

---

Otras arquitecturas

---

## Redes siamesas

Otras arquitecturas

---

## Sistemas de recomendación

Otras arquitecturas

---

## Arquitecturas diseñadas para el procesamiento de conjuntos

# Arquitecturas diseñadas para conjuntos

Existen arquitecturas de red especialmente diseñadas para tratar y modelar **conjuntos**. La clave en este tipo de problemas es que el orden de los elementos del conjunto no debería de influenciar la salida de la red. Algunas aplicaciones podrían ser:

- Detección de anomalías en un conjunto.
- Predicción de estadísticos de un conjunto.
- Cuantificación



## Definición del problema:

Dado un conjunto de entrenamiento  $D = (\{x_i\}_{i=1}^n, y)$ , es decir, un conjunto de muestras etiquetadas, tratamos de aprender un modelo que sea capaz de predecir una nueva muestra de test  $T = \{x_i\}_{i=1}^m$ .

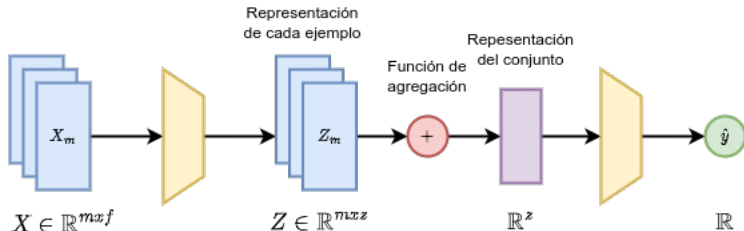
Para resolver este problema necesitamos arquitecturas que sean capaces de procesar conjuntos de datos y representarlos de manera **invariante al orden**. Es decir, la evaluación de la red para una muestra  $T$  debería de cumplir:

$$f(\{x_i\}_{i=1}^m) = f(\phi(\{x_i\}_{i=1}^m)),$$

siendo  $\phi$  una permutación cualquier de los ejemplos de la muestra.

# Arquitecturas diseñadas para conjuntos: DeepSets

**DeepSets** es una arquitectura de red que propone una capa de representación basada en **funciones de agregación simples**:



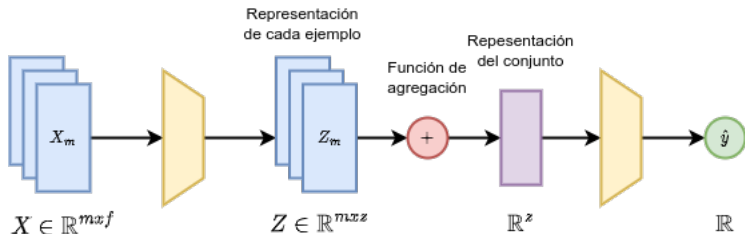
donde  $m$  sería el tamaño del conjunto de entrada y  $f$  la dimensión de cada ejemplo.

## Funciones de agregación

Las funciones de agregación a utilizar pueden ser: suma, máximo, mínimo, media, mediana, etc.

# Arquitecturas diseñadas para conjuntos: DeepSets

Las funciones de agregación utilizadas por DeepSets tienen como objetivo obtener una **representación única de la muestra en un espacio latente**.



A partir de esta representación la segunda parte de la red puede aprender a asociar estos vectores con su etiqueta final.



# Arquitecturas diseñadas para conjuntos: DeepSets

Algunos puntos importantes sobre DeepSets son:

- La arquitectura está pensada funcionar como un **aproximador universal** de cualquier función que trabaje con conjuntos.
- La función de agregación es una manera **demasiado simple** de representar los ejemplos del conjunto, perdiéndose mucha información en el proceso.
- La función de agregación no captura **iteraciones entre los elementos del conjunto**.
- Algunas funciones de agregación como la suma pueden no sirven para **conjuntos de tamaño variable**.

# Arquitecturas diseñadas para conjuntos: SetTransformers

SetTransformers es otra arquitectura diseñada para trabajar con conjuntos que explota el concepto de **atención** de los **transformers**.

- **Modelan relaciones** entre los diferentes ejemplos del conjunto.
- Aprovechan la atención para capturar **relaciones entre elementos**.
- Flexibilidad para manejar **conjuntos de tamaños variables**.
- Mejoran el rendimiento de DeepSets.

# Arquitecturas diseñadas para conjuntos: SetTransformers

La idea básica de los SetTransformers es calcular la atención entre los elementos del conjunto. El principal problema de esta arquitectura es que su complejidad es  $O(n \times n)$ , siendo  $n$  el número de ejemplos del conjunto.

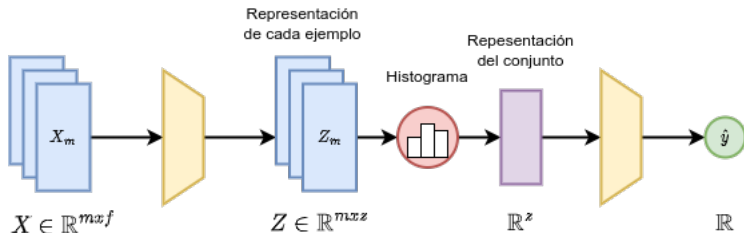
Para resolver este problema los autores de SetTransformers calculan la atención de cada ejemplo a una serie de puntos llamados **inducing points**, reduciendo por tanto la complejidad a  $O(n \times i)$ , siendo  $i$  el número de inducing points.

## Nota importante

Ten en cuenta que **los transformers por defecto son invariantes al orden**. Es por ello que en problemas de NLP se les añade la posición de cada palabra en la frase para poder capturar esa información que es importante a la hora de procesar texto.

# Arquitecturas diseñadas para conjuntos: HistNetQ

HistNetQ es una arquitectura de red pensada específicamente para **cuantificación**. En esta arquitectura se utiliza como capa de representación de la muestra **histogramas**. La idea es que un histograma es una herramienta de representación adecuada para este problema.



## Cuantificación

Recuerda que la cuantificación consiste en estimar la **prevalencia de las clases** en un conjunto de test.

# Arquitecturas diseñadas para conjuntos: HistNetQ

La arquitectura es similar a la de **DeepSets** pero utilizando histogramas para representar las muestras. La red calcula un histograma por cada una de las características de la representación de ejemplos.

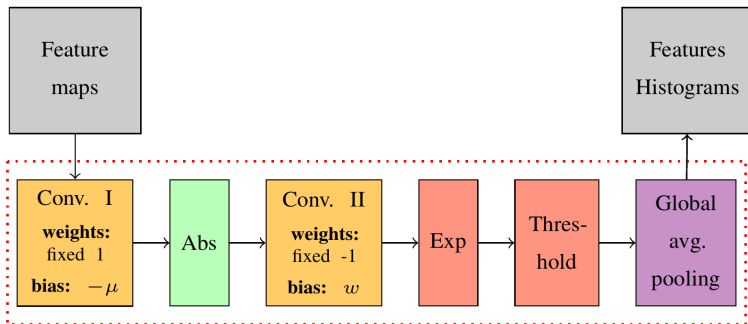


Como usar histogramas en una red

El principal problema de usar histogramas es que **el cálculo de un histograma no es una operación diferenciable**.

# Arquitecturas diseñadas para conjuntos: HistNetQ

La solución es utilizar **histogramas diferenciables**.



## Histogramas diferenciables

Los histogramas diferenciables utilizan operaciones comunes (como las convoluciones), para calcular una aproximación a un histograma.