
Clase 017 — NumPy: broadcasting

Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 2 § 2.5 Computation on Arrays: Broadcasting. · Duración estimada: 75 min.

Clase 017 — NumPy: broadcasting

Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 2 § 2.5 Computation on Arrays: Broadcasting. · Duración estimada: 75 min.

Objetivo

Que el alumno internalice las reglas de broadcasting — el mecanismo por el que NumPy operó arrays de shapes distintos sin copiar datos. Es lo que hace que $M - M.mean(axis=0)$ centrado por columna sea una línea, no un bucle anidado.

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Recitar las 3 reglas de broadcasting (alineación por la derecha, dim 1 estira, falla si no es 1 ni igual).
2. Predecir la shape del resultado de una operación entre arrays de shapes distintos.
3. Centrar y escalar matrices por fila/columna sin loops.
4. Usar `np.newaxis` (o `None`) para promover un vector a matriz fila/columna.
5. Diagnosticar un `ValueError`: `operands could not be broadcast together` leyendo las shapes.

Temas

#	Tema	Por qué importa
1	Las 3 reglas	Padding a la derecha, dim 1 estira, error
2	Vector + matriz	Vector como fila o como columna.
3	<code>np.newaxis</code> / <code>None</code>	Insertar eje de tamaño 1.
4	Caso canónico: centrar/escalar	$X - X.mean(axis=0)$ y $(X - \mu) / \sigma$.
5	Outer product sin loop	<code>a[:, None] * b[None, :]</code> .
6	<code>ValueError</code> común: <code>"operands could not be b</code>	Cómo leerlo.

Definiciones y características

Broadcasting

: Mecanismo por el que NumPy opera arrays de shapes distintos sin copiar memoria, estirando virtualmente las dimensiones de tamaño 1. Lo que hace posible $X - X.mean(axis=0)$ (centrado por columna) en una línea sin loops.

Regla 1 — padding por la izquierda

: Si los arrays tienen distinta cantidad de dimensiones, la shape del menor se rellena con 1s a la izquierda. $(4,)$ operado con $(3, 4)$ se trata como $(1, 4)$ vs $(3, 4)$.

Regla 2 — estirar dim 1

: En cada dimensión donde los tamaños difieren, si uno es 1 se estira al otro. $(3, 1)$ y $(3, 4) \rightarrow$ ambos $(3, 4)$ (la primera se estira en eje 1).

Regla 3 — fallo

: Si en alguna dimensión los tamaños son distintos y ninguno es 1, lanza `ValueError: operands could not be broadcast together`. No hay forma de inferir qué hacer.

`np.newaxis` (alias `None`)

: Inserta una dimensión de tamaño 1 donde lo pongas. `v[:, None]` convierte vector (3,) en columna (3, 1). Crítico para forzar broadcasting en la dirección correcta.

Outer product vía broadcasting

: `a[:, None] * b[None, :]` produce matriz `(len(a), len(b))` con todos los productos par a par — equivalente a `np.outer(a, b)` pero usando broadcasting puro.

Dataset / recursos

Sintético: matriz de features 100×5 para estandarización. Sin descarga.

Ejercicios

1. Predice antes de ejecutar. Para shapes (3,), (3,1), (1,3), (2,3,4) × (4,), predice la shape del resultado. Verifica.
2. Estandariza features. Matriz 100×5 aleatoria. Resta media por columna y divide por std por columna en una línea.
3. Outer product. Vectores `a=[1,2,3]`, `b=[10,20,30,40]`. Calcula la matriz outer (3×4) sin `np.outer`, solo broadcasting.
4. Distance matrix. Dados 5 puntos 2D, construye matriz 5×5 de distancias euclídeas entre pares — sin `cdist`, solo broadcasting.
5. Diagnostica error. Intenta `np.ones((3,4)) + np.ones((4,3))`. Lee el `ValueError` y explica.

Homework verificable

Notebook que: (a) predice shapes de 4 operaciones broadcasting y verifica; (b) estandariza una matriz feature por columna en una línea; (c) construye distance matrix de 100 puntos sin loop; (d) provoca y explica un error de broadcasting.

Criterio de aceptación: Las predicciones coinciden. Estandarización: media≈0, std≈1 por columna.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
<code>ValueError: operands could not be broadcas</code>	Las shapes alineadas por la derecha no son
Resté <code>M.mean(axis=0)</code> y los promedios queda	<code>mean(axis=0)</code> devuelve shape <code>(n_cols,)</code> — se
<code>a + b</code> con shapes (3,) y (3,) da escalar (s	No — da array (3,) elementwise. Producto p
Memoria explota en una operación 'inocente	<code>X[:, None, :] - X[None, :, :]</code> produce arra
<code>a[None] + b</code> no se broadcastea como espero	<code>a[None]</code> añade dim al inicio. Quizás quería

Preguntas frecuentes

¿Cómo predigo la shape del resultado?

(1) Alinea las shapes por la derecha. (2) En cada columna alineada: si son iguales o uno es 1, OK; si no, error. (3) Resultado: por dimensión, toma el max de las dos.

¿Broadcasting copia memoria?

No — es virtual. NumPy itera con strides 0 en las dims estiradas. Por eso es tan eficiente: cero alloc extra (excepto el array resultado).

¿`X - X.mean(0)` o `X - X.mean(0, keepdims=True)`?

Para 2D ambos funcionan (broadcasting alinea). En 3D+, `keepdims=True` preserva la dimensión como 1 y evita confusiones. Recomendado en general.

¿`np.newaxis` o `None`?

Aliases — `arr[:, None]` y `arr[:, np.newaxis]` son idénticos. `None` es más conciso; muchos prefieren `np.newaxis` por explicitud.

¿Qué hago si broadcasting no me sirve?

Operaciones que no se ajustan a las reglas: usa `np.einsum` (más expresivo), `np.tensordot`, o `reshape` explícito. Como último recurso, loop Python — pero busca librería específica antes.

Referencias

- VanderPlas, cap. 2 § 2.5 Broadcasting.
- NumPy broadcasting docs

Siguiente clase

Clase 018 — NumPy: boolean masks y fancy indexing

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
import numpy as np
rng = np.random.default_rng(42)
```

Archivos complementarios

- notebook.ipynb