

# Deep Learning

Tema 1 – Ejercicios con Tensores



POLITÉCNICA

UNIVERSIDAD  
POLITÉCNICA  
DE MADRID



Máster  
Deep Learning



1. Introducción a los tensores
2. Operaciones básicas en Pytorch
3. Manipulación de tensores
4. Operaciones avanzadas
5. Trabajar con GPU

# 1. Introducción a los tensores

## ¿Qué es un tensor?

- Son ***unidades fundamentales de datos*** utilizadas para el cálculo en Pytorch y otras librerías de Deep Learning, como TensorFlow.
- Son ***contenedores de datos***, donde los valores están organizados en un espacio multidimensional.

## Tipos de tensores según la dimensionalidad

- **Escalar**: tensor de dimensión 0.

$$x = 5$$

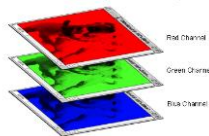
- **Vector**: tensor de dimensión 1.

$$x = [1, 2, 3]$$

- **Matriz**: tensor bidimensional.

$$x = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

- **Tensor de 3 o más dimensiones.**



## Tipos de tensores según el tipo de datos.

- **torch.IntTensor**

$$x = [1, 2, 3]$$

- **torch.FloatTensor**

$$x = [1.2, 2.3, 3.4]$$

- **torch.DoubleTensor**

$$x = [1.2000, 2.3000, 3.4000]$$

- **torch.BoolTensor**

$$x = [False, True, False]$$

## Propiedades de los tensores

- Forma
- Tamaño
- Tipo de datos
- Dispositivo en el que está
- Número de dimensiones
- Número de elementos

## 2. Operaciones básicas con tensores



## Principales operaciones básicas con tensores

- Suma
- Resta
- Multiplicación
- Multiplicación matricial
- División

## Principales operaciones básicas con tensores

$$\text{Tensor 1} = [1, 2, 3]$$

$$\text{Tensor 2} = [4, 5, 6]$$

$$\text{sum tensor} = \text{Tensor1} + \text{Tensor2}$$

$$\text{sum tensor} = \text{torch.add}(\text{Tensor1}, \text{Tensor2})$$

# Principales operaciones básicas con tensores: Multiplicación Matricial



$$\begin{bmatrix} 2 & 5 & 2 \\ 1 & 0 & -2 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 & 0 \\ -2 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

## Principales operaciones básicas con tensores: **Producto matricial**



$$\text{Tensor 1} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{Tensor 2} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$\text{multiplicación matricial} = \begin{bmatrix} 1 * 5 + 2 * 7 & 1 * 6 + 2 * 8 \\ 3 * 5 + 4 * 7 & 3 * 6 + 4 * 8 \end{bmatrix}$$

$$\text{multiplicación matricial} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

# Principales operaciones básicas con tensores:

## Producto matricial



- @
- torch.matmul
- torch.mm

# 3. Manipulación de tensores



## Principales operaciones básicas con tensores

- Cambiar la forma
- Transponer
- Expandir las dimensiones
- Combinación de tensores

Reshape

View



## Principales operaciones básicas con tensores

### Diferencias entre view y reshape

	View	Reshape
<b>Contigüidad</b>	Requiere que el tensor sea contiguo.	Funciona incluso si el tensor no es contiguo.
<b>Copia de datos</b>	Nunca hace una copia.	Puede hacer una copia si el tensor no es contiguo.
<b>Velocidad</b>	Más rápido (si el tensor es contiguo).	Puede ser más lento si necesita copiar datos.
<b>Flexibilidad</b>	Más restrictivo.	Más flexible.



## 4. Operaciones avanzadas

## Operaciones avanzadas con tensores

- Producto punto
- Suma de los elementos de un tensor
- Media
- Desviación estándar
- Mínimo
- Máximo
- Reemplazar valores utilizando ***torch.where***
- Suma de matriz y vector
- Rellenar una matriz con un valor definido



## Operaciones avanzadas con tensores: **Producto punto**

$$\textit{Tensor 1} = [1, 2, 3] \quad \textit{Tensor 2} = [4, 5, 6]$$

$$\textit{Producto punto} = 1 * 4 + 2 * 5 + 3 * 6$$

$$\textit{Producto punto} = 32$$



## Operaciones avanzadas con tensores: **torch.where**

*torch.where* (*condición*, *valor True*, *valor False*)

Ejemplo de condición:

Tensor =  $\begin{bmatrix} 1 & 3 \\ 0.5 & 2 \end{bmatrix}$

*torch.where*(*tensor*  $\leq 1$ , *tensor*, 5)

**Tensor resultante =  $\begin{bmatrix} 1 & 5 \\ 0.5 & 5 \end{bmatrix}$**

# 5. Trabajar con GPU



- Cómo conectarnos a la GPU
- Ver el número de GPUs disponibles
- Mostrar las propiedades de la GPU
- Mostrar la memoria utilizada y reservada
- Como crear tensores en GPU
- Como pasar tensores a GPU
- Como pasar tensores a CPU

# 6.EXTRA RESOURCES



## Documentación

- <https://pytorch.org/docs/stable/>
- <https://pytorch.org/docs/stable/generated/torch.where.html>
- <https://pytorch.org/docs/stable/generated/torch.mean.html>
- <https://www.run.ai/guides/gpu-deep-learning/pytorch-gpu>

## Ejercicios adicionales

- [TensorGym Exercises - Comprehensive Machine Learning Practice](#)
- <https://www.deep-ml.com/>



# Deep Learning

Tema 1 – Ejercicios con Tensores



POLITÉCNICA

UNIVERSIDAD  
POLITÉCNICA  
DE MADRID



Máster  
Deep Learning