

Tema 1.1

Presentación de la Asignatura

Deep Learning

Miguel Ángel Martínez del Amor

Máster Oficial en Ingeniería Informática

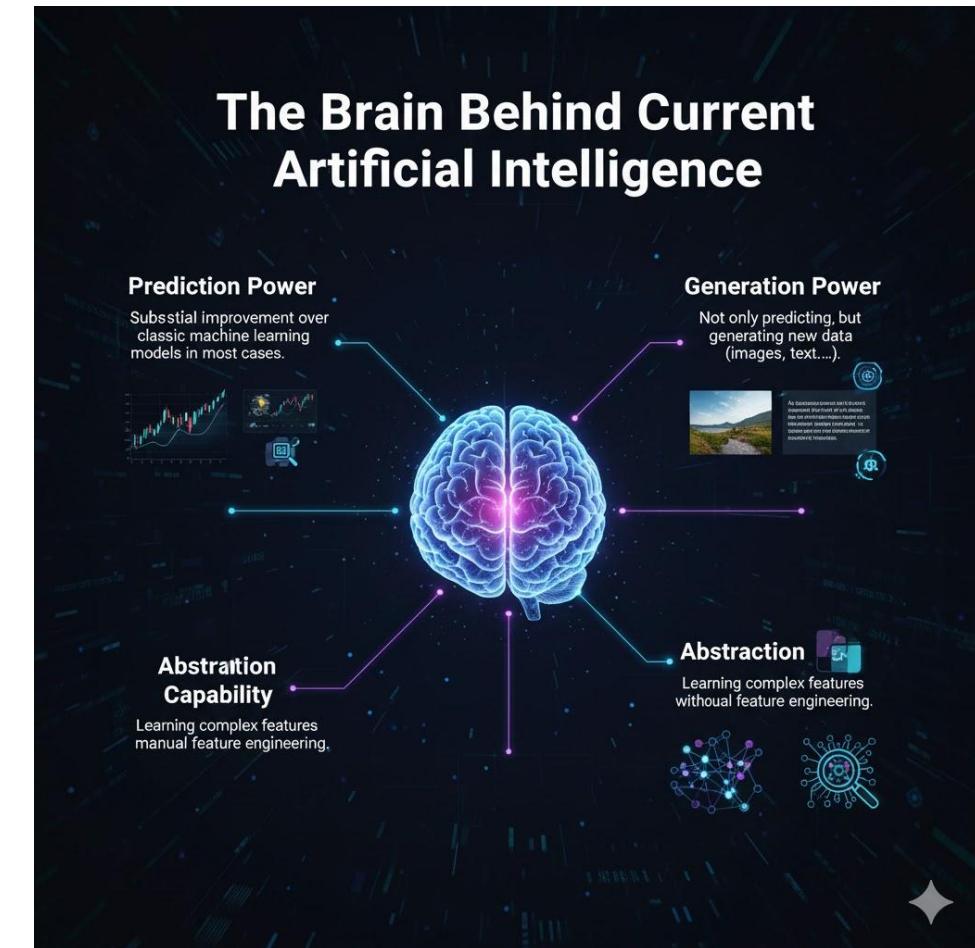
Universidad de Sevilla

Motivación

- Deep Learning es la rama que ha revolucionado la IA y sus aplicaciones reales
- Surge de la convergencia de tres vertientes:
 - **Datos**: vivimos en la era del big data, por el impacto social de Internet.
 - **Modelos teóricos**: desde los años 50, los investigadores han trabajado mucho en el área de la IA y del ML, desarrollando modelos cada vez más potentes, hambrientos de datos pero costosos de ejecutar.
 - **Hardware**: dispositivos paralelos como las GPUs ha democratizado la supercomputación.

Motivación

- El cerebro detrás de la **actualidad** en Inteligencia Artificial
 - **Poder de predicción:** mejora sustancial a modelos clásicos de machine learning en la mayoría de los casos
 - **Poder de generación:** no solo predecir, sino generación de nuevos datos (imágenes, texto...)
 - **Capacidad de abstracción:** aprender características complejas sin una ingeniería de características manual.



Motivación

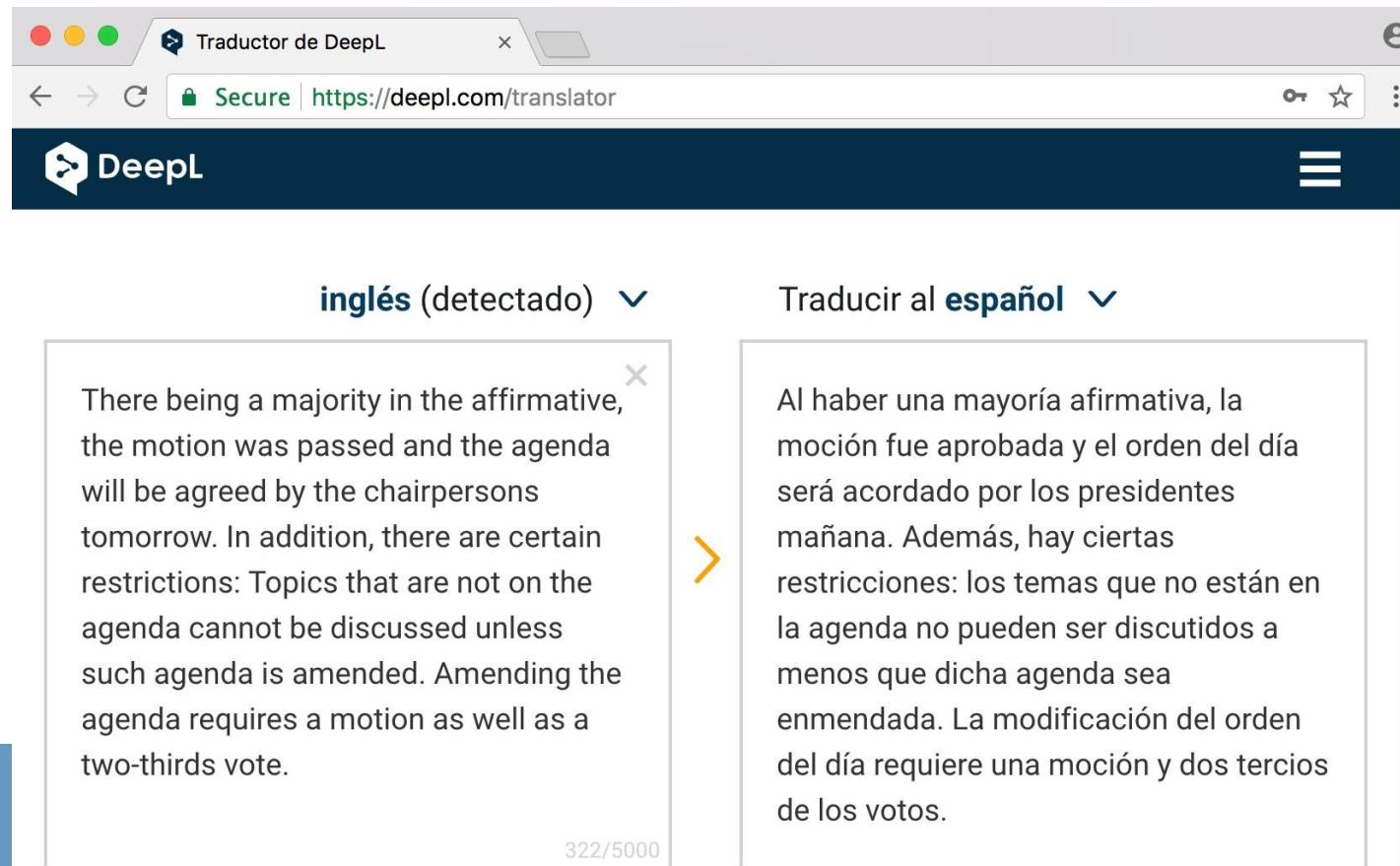
- Oportunidades profesionales:
 - **Alta demanda laboral:** perfil escaso y muy demandado
 - **Salarios competitivos**
 - **Roles:** científico de datos, IA Architect, o Ingeniero de: ML, MLOps, MLOps, LLM Fine-Tuning, orquestación, AI, ML platform, ...
 - Aunque la IA empieza a reemplazar algunos puestos automatizables...



Fuente: Informe «caminos que convergen» de KPMG, S.A

Motivación

- Traductores automáticos (deepL translator)



The screenshot shows a web browser window for the DeepL Translator. The URL is <https://deepl.com/translator>. The source text is in English and reads:

inglés (detectado) ▾
There being a majority in the affirmative, the motion was passed and the agenda will be agreed by the chairpersons tomorrow. In addition, there are certain restrictions: Topics that are not on the agenda cannot be discussed unless such agenda is amended. Amending the agenda requires a motion as well as a two-thirds vote.

Traducir al **español** ▾
Al haber una mayoría afirmativa, la moción fue aprobada y el orden del día será acordado por los presidentes mañana. Además, hay ciertas restricciones: los temas que no están en la agenda no pueden ser discutidos a menos que dicha agenda sea enmendada. La modificación del orden del día requiere una moción y dos tercios de los votos.

322/5000

Motivación

- Chatbots inteligentes ([chatGPT](#))

¿Sabes recetas?

Sí, conozco algunas recetas básicas. Por ejemplo, una receta simple para hacer una tortilla española es la siguiente:

- Ingredientes:
 - 4 huevos
 - 1 cebolla
 - 1 pimiento verde
 - Aceite de oliva
- Preparación:
 - Pelar y picar la cebolla y el pimiento verde en trozos pequeños.
 - Calentar un poco de aceite de oliva en una sartén y añadir la cebolla y el pimiento. Cocinar a fuego medio hasta que estén transparentes.

La tortilla de patatas mejor sin cebolla

La tortilla de patatas es una receta muy versátil y se puede adaptar a diferentes gustos y preferencias. Si no te gusta la cebolla, simplemente puedes omitirla de la receta. La cebolla aporta un sabor dulce y suave a la tortilla, pero si no te gusta, puedes probar añadiendo otras verduras en su lugar, como pimientos, calabacín, espinacas o incluso champiñones. También puedes agregar otras especias o ingredientes para darle un toque distintivo a tu tortilla, como ajo, perejil, queso rallado o jamón picado. La tortilla de patatas es una receta muy versátil, así que sientete libre de experimentar y adaptarla a tus preferencias.

AB

Pasa de Python a Haskell el siguiente código:

```
from dataclasses import dataclass
from typing import Generic, TypeVar
```

```
A = TypeVar("A")
```

```
@dataclass
class Arbol(Generic[A]):
```

```
    pass
```

```
@dataclass
class Hoja(Arbol[A]):
```

```
    x: A
```

```
@dataclass
class Nodo(Arbol[A]):
```

```
    i: Arbol[A]
    d: Arbol[A]
```

```
def altura(a: Arbol[A]) -> int:
    match a:
        case Hoja(_):
            return 0
        case Nodo(i, d):
            return 1 + max(altura(i), altura(d))
    assert False
```

rn

```
data Arbol a = Hoja a | Nodo (Arbol a) (Arbol a)
```

```
altura :: Arbol a -> Int
```

```
altura (Hoja _) = 0
```

```
altura (Nodo i d) = 1 + max (altura i) (altura d)
```

Motivación

- Localización de objetos en imágenes/video



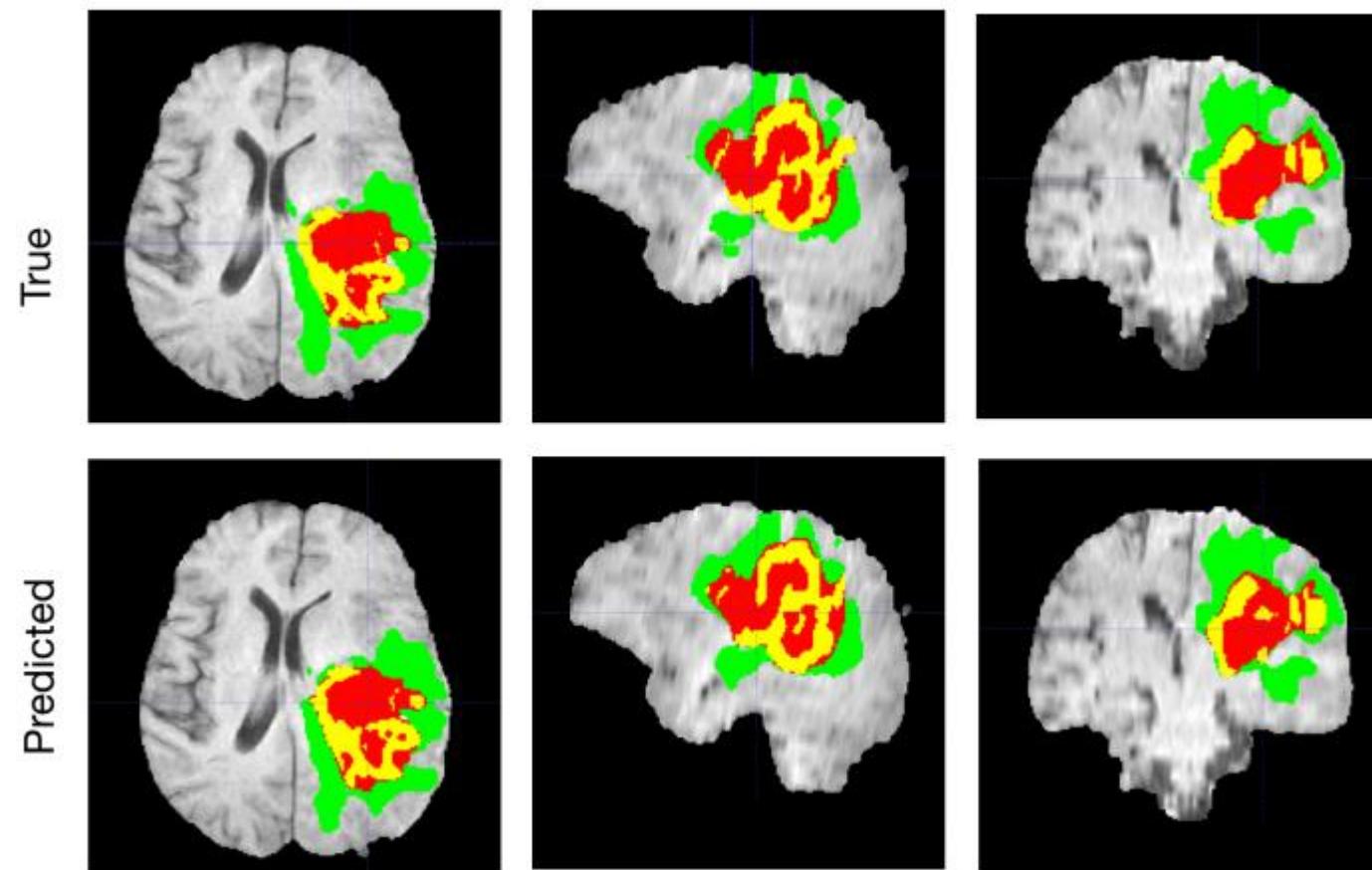
Motivación

- Conducción autónoma ([Tesla autopilot](#))



Motivación

- Segmentación de tumores



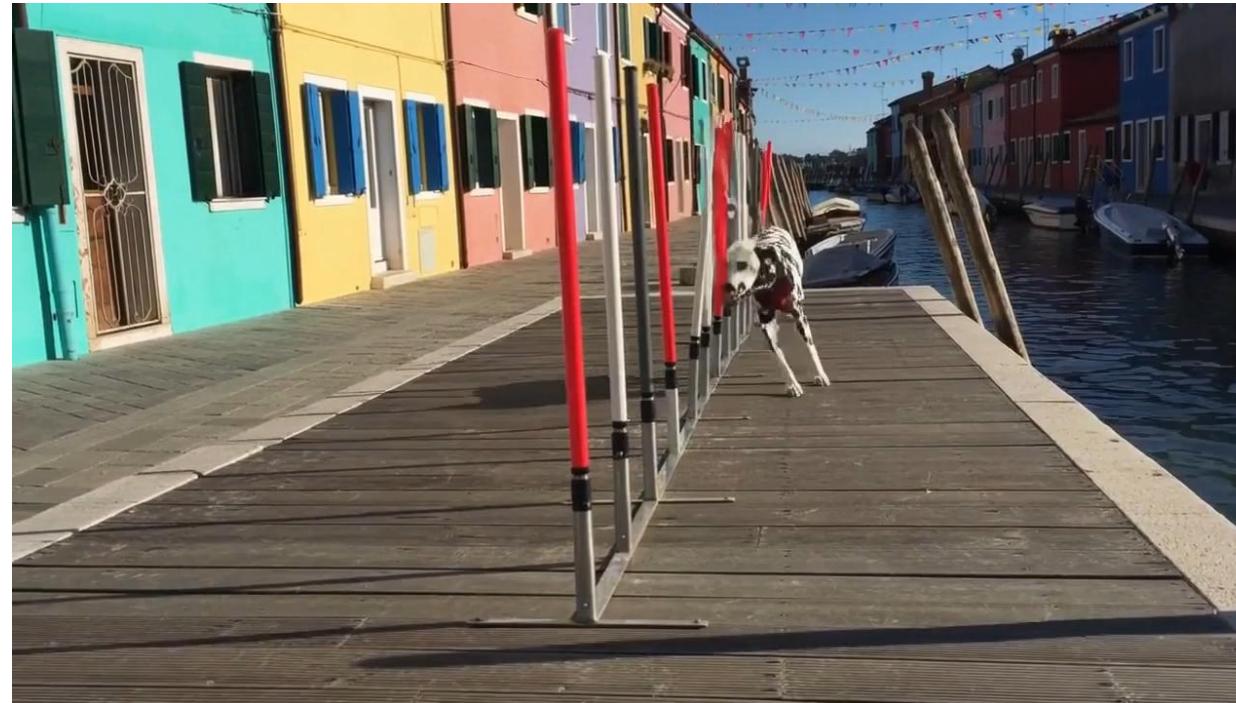
Motivación



- Generación de imágenes
 - Ejemplo: “una imagen hiperrealista sobre un estudiante estudiando un libro de deep learning en Sevilla”. Nanobanana dentro de Gemini



Motivación



- Generación de video con audio
 - Ejemplo con Sora 2 (OpenAI)



Motivación



- Generación de mundos
 - Ejemplos con Genie 3 (Google)

Motivación

- Descubrir ciencia (DeepMind)

A Problem (input)

You are given two strings s and t , both consisting of lowercase English letters. You are going to type the string s character by character, from the first character to the last one.

When typing a character, instead of pressing the button corresponding to it, you can press the "Backspace" button. It deletes the last character you have typed among those that aren't deleted yet (or does nothing if there are no characters in the current string). For example, if s is "abcd" and you press Backspace instead of typing the first and the fourth characters, you will get the string "bd" (the first press of Backspace deletes no character, and the second press deletes the character 'c'). Another example, if s is "abca" and you press Backspace instead of the last two letters, then the resulting text is "a".

Your task is to determine whether you can obtain the string t , if you type the string s and press "Backspace" instead of typing several (maybe zero) characters of t .

Input

The first line contains a single integer q ($1 \leq q \leq 10^5$) — the number of test cases.

The first line of each test case contains the string s ($1 \leq |s| \leq 10^5$). Each character of s is a lowercase English letter.

The second line of each test case contains the string t ($1 \leq |t| \leq 10^5$). Each character of t is a lowercase English letter.

It is guaranteed that the total number of characters in the strings over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print "YES" if you can obtain the string t by typing the string s and replacing some characters with presses of "Backspace" button, or "NO" if you cannot.

You may print each letter in any case (YES, yes, Yes will all be recognized as positive answer, NO, no and nO will all be recognized as negative answer).

Example

Note

Consider the example test from the statement.

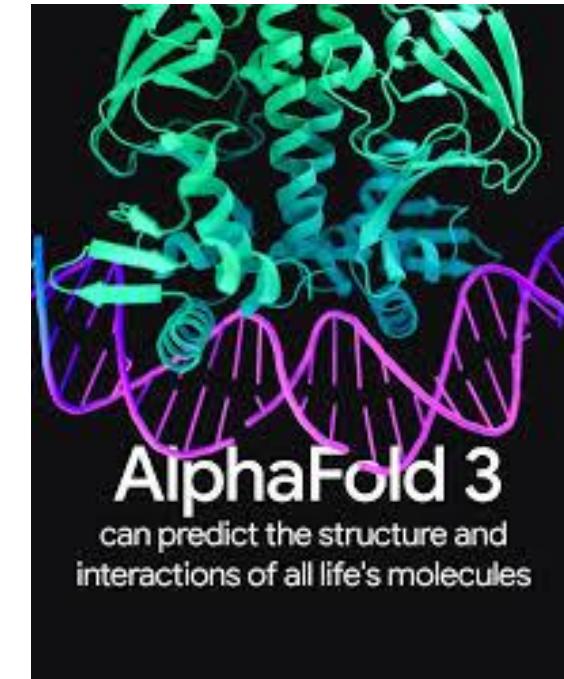
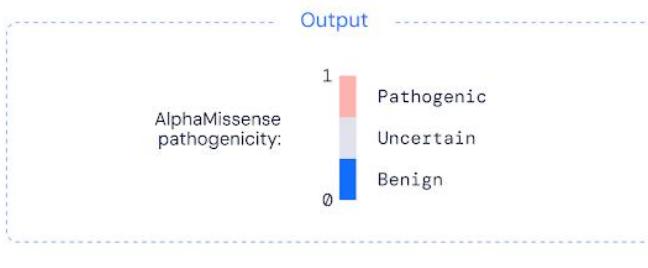
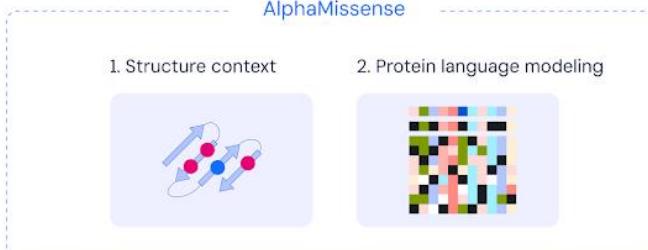
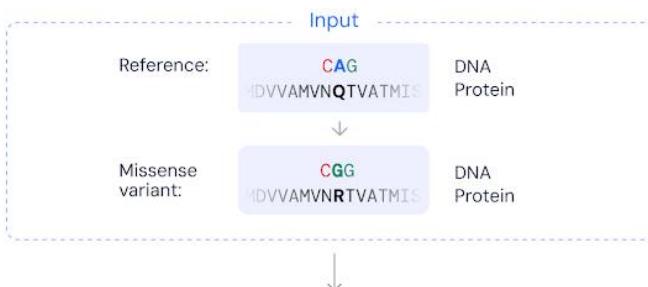
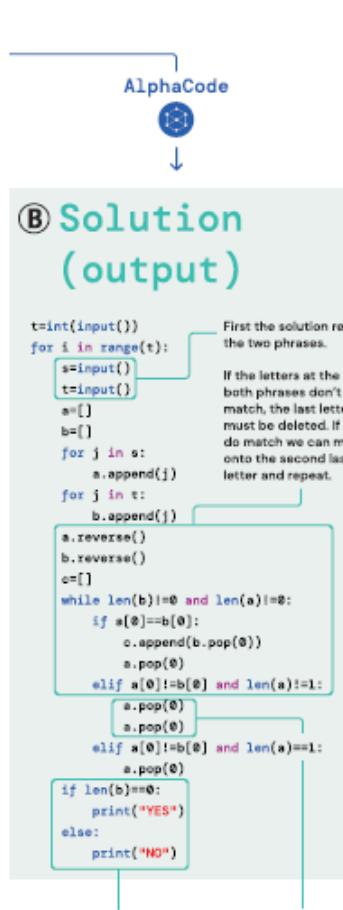
In order to obtain "ba" from "abab", you may press Backspace instead of typing the first and the fourth characters.

There's no way to obtain "bb" while typing "ababa".

There's no way to obtain "aaa" while typing "aa".

In order to obtain "ababa" while typing "aababa", you have to press Backspace instead of typing the first character, then type all the remaining characters.

Input	Output
4	YES
ababa	NO
ba	NO
aa	YES
aaaa	NO
aababa	YES



Objetivos de la asignatura

- Comprender los **fundamentos** teóricos de las redes neuronales:
 - Conocer sus variantes, sus ventajas e inconvenientes en cada caso
 - Saber manejar las principales técnicas de entrenamiento y de regularización para evitar los típicos problemas.
 - Entender cuándo y cómo aplicar transferencia de aprendizaje.
- Aprender a **utilizar** redes neuronales:
 - Hacer regresión, clasificación, localización y generación.
 - Trabajar con redes MLP, convolucionales, recurrentes y Transformers (LLMs).
- Fundamentos prácticos con el framework más popular actualmente (**PyTorch**) y la aceleración por **GPUs**

NO cubriremos

- Uso de modelos privativos (chatGPT, Gemini, Claude...)
- Uso de APIs de modelos para crear aplicaciones
- Uso/despliegue de sistemas agénticos
- **Una vez afianzados los fundamentos, entenderás mejor estas herramientas y cómo usarlas para tus propósitos**
 - Los fundamentos te darán los conceptos para que te puedas adaptar al futuro de la IA

Contenidos de la asignatura

- Dividido en 7 módulos:
 - **Módulo 1:** Introducción
 - **Módulo 2:** Fundamentos Teóricos
 - **Módulo 3:** Fundamentos Prácticos
 - **Módulo 4:** Entrenamiento eficiente
 - **Módulo 5:** Modelos para Visión por Computador
 - **Módulo 6:** Modelos para Procesamiento del Lenguaje Natural
 - **Módulo 7:** Modelos Generativos

Contenidos de la asignatura

Semana	Módulos	Entregables
Primera	Módulos 1, 2 y 3	Ejercicio 1
Segunda	Módulos 4 y 5	Ejercicio 2
Tercera	Módulos 6 y 7	Ejercicio 3
Cuarta	Trabajo*	Mini-proyecto

* Se reserva la última semana para trabajar en los ejercicios propuestos.

Contenidos de la asignatura

- Formato de cada módulo:
 - **Introducción** al módulo.
 - Temas y sesiones prácticas, que pueden estar intercaladas:
 - **Temas de teoría**, introduciendo los conceptos fundamentales.
 - Diapositivas en formato pdf
 - Vídeos (píldoras).
 - **Sesiones de trabajo** con notebooks de Jupyter, donde se pondrá en práctica los conceptos y se afianzarán de nuevo los conceptos teóricos.
 - Ficheros notebook y, según el caso, el dataset, que debes leer y reproducir por tu cuenta.
 - **Ejercicio de autoevaluación.**
 - Material complementario.

Evaluación de la asignatura

- Entregables: **80%** (8 puntos)
- Especialización: **20%** (2 puntos)

Evaluación de la asignatura

- **Entregables: 80% (8 puntos)**
 - **3 ejercicios básicos** por semana:
 - **Semana 1:** entrenamiento básico de redes neuronales (1 punto)
 - **Semana 2:** entrenamiento de modelo para visión (2 puntos)
 - **Semana 3:** entrenamiento de modelo para lenguaje natural (2 puntos)
 - Un **mini-proyecto** final (3 puntos). A elegir por el alumno:
 - Entrenamiento de un detector de objetos
 - Entrenamiento de un modelo de segmentación
 - Entrenamiento de un LLM
 - Propuesta del alumno (consultar con el profesor)
- Especialización: 20% (2 puntos)

Evaluación de la asignatura

- **Entregables: 80% (8 puntos)**
 - **4 ejercicios** entregables: 3 por semana y un mini-proyecto final
 - **Fechas de entrega:**
 - Evaluación continua:
 - Plan: uno por semana, hasta el martes de la semana siguiente.
 - Límite: todos hasta el martes de la semana después del curso (semana 4).
 - Convocatoria: por lo general, el segundo día de la semana de exámenes.
 - **Defensa:**
 - Evaluación continua: después de la entrega del último ejercicio (todos juntos)
 - Convocatoria: por lo general, a partir del tercer día de la semana de exámenes.
- Especialización: 20% (2 puntos)

Evaluación de la asignatura

- Proyecto práctico: 80% (8 puntos)
- **Especialización: 20% (2 puntos)**
 - A elegir entre las siguientes opciones:
 - a) Un **examen oral de dos preguntas** sobre conceptos generales vistos en la asignatura
 - b) Realización de **talleres del NVIDIA Deep Learning Institute** (código disponible en Enseñanza Virtual)
 - **Realización/Entrega:** el día de la defensa.
 - Realización/entrega obligatoria para superar la asignatura.

Recomendaciones para superar el curso

- **Superación:** al menos 5 puntos en total, y una aportación a la especialización.
- Visualizar el contenido teórico y entender los conceptos fundamentales.
- Trabajar las prácticas para comprender el código y el comportamiento de los entornos.
- Desarrollar los ejercicios y el trabajo con tiempo, dedicación y ganas:
 - La calidad de los resultados se nota, y se valorará en la nota final.
- Usa el espacio en Teams, y participa activamente. Compartir conocimientos con tus compañeros te ayudará a afianzarlos.