



UNIVERSIDAD DE CHILE

# Deep Learning

Deeper, Better, \_\_\_\_\_, Stronger than Machine Learning

---

Valentin Barriere

Universidad de Chile – DCC

CC6204, Primavera 2025

# Deep Learning

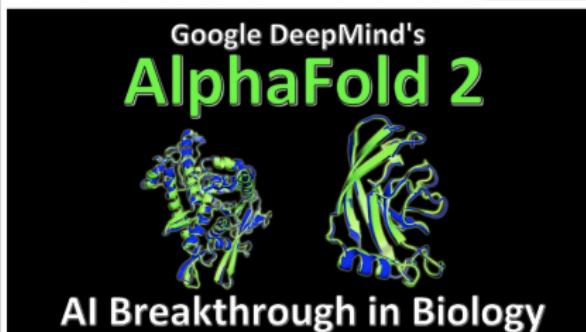
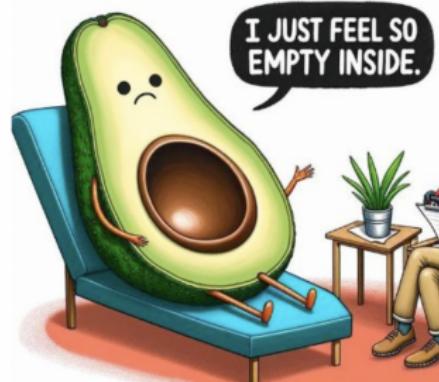
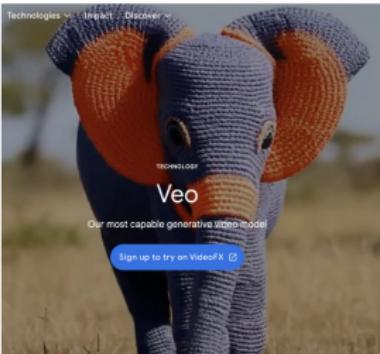


Figure 1: Conversational Agent, Image/Video generation, Protein structure

# Outline : Significacion de las termas

Significacion de las termas

Aplicaciones

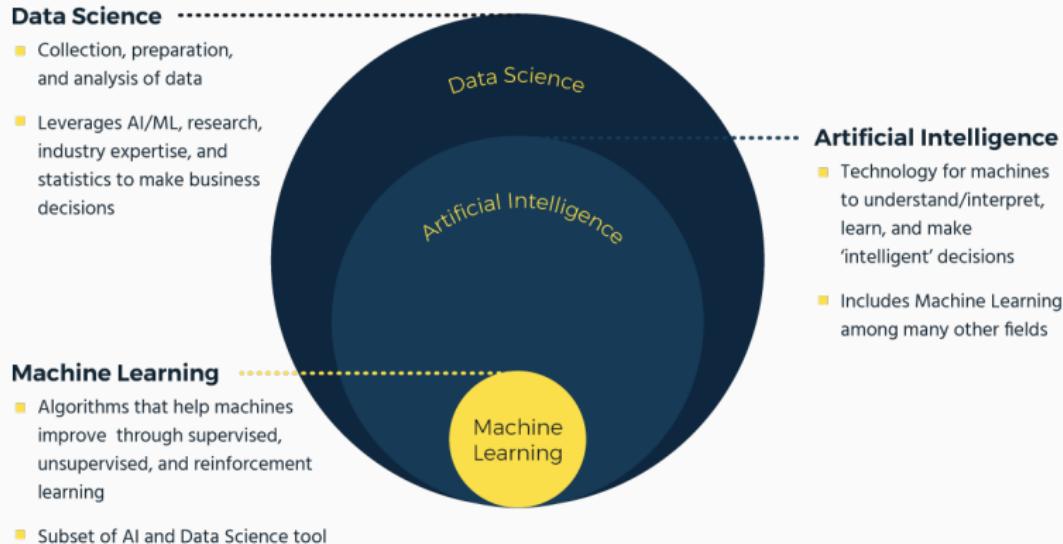
Historia

Los recursos

Principe

El curso

## AI vs. Data Science vs. Machine Learning



**Figure 2:** Diferencias entre campos

## En resumen

Data Science se centra en el análisis de datos para extraer conocimiento, Machine Learning utiliza algoritmos para hacer predicciones y tomar decisiones basadas en datos, y Artificial Intelligence se refiere al desarrollo de sistemas que pueden realizar tareas inteligentes de manera autónoma.

Definición (sobre) simplista:

- Data mining genera entendimiento.
- Machine learning genera predicciones.
- Artificial intelligence genera acciones.

# Ejemplo en plataforma de música

## Data Scientist

Recopila y analiza datos de usuarios de plataformas de música para identificar patrones y preferencias musicales.

## Machine Learner

Desarrolla y optimiza un modelo de recomendación de música utilizando algoritmos de aprendizaje automático para predecir las preferencias de los usuarios.

## Artificial Intelligence

Implementa un agente social que puede interagir con el usor, para mejorar la personalización de las recomendaciones musicales y proporcionar una experiencia más precisa y contextualizada.

# Datos

Diferentes tipos de datos:

- Datos estructurados:
  - Datos sociales: Edad, Salario, Color de piel, Lugar de residencia
  - Datos métricos: Likes de una publicación, Tiempo pasado en una página, Número de amigos en común
- Datos no estructurados:
  - Texto: Frase, Párrafo, Documento
  - Sonido: Canción, Discurso
  - Imagen: Foto, Vídeo

Diferentes tipos de Minería:

- Exploración de datos: Detectar valores simples, sesgos
- Tarea de clasificación/regresión: Alimentarse de datos para caracterizar nuevos datos **por clase o con un valor**, de manera supervisada
- Tarea de agrupamiento: Caracterizar datos por clase de manera no supervisada
- Reducción de dimensiones: Desarrollar estructuras comunes para representaciones comprimidas de datos

Diferentes tipos de Minería:

- Exploración de datos: Detectar valores simples, sesgos
- Tarea de clasificación/regresión: Alimentarse de datos para caracterizar nuevos datos **por clase o con un valor**, de manera supervisada
- Tarea de agrupamiento: Caracterizar datos por clase de manera no supervisada
- Reducción de dimensiones: Desarrollar estructuras comunes para representaciones comprimidas de datos
- **Generacion: generar datos segun una distribucion particular**  
(mas complejo)

# Outline : Historia

---

Significacion de las termas

**Historia**

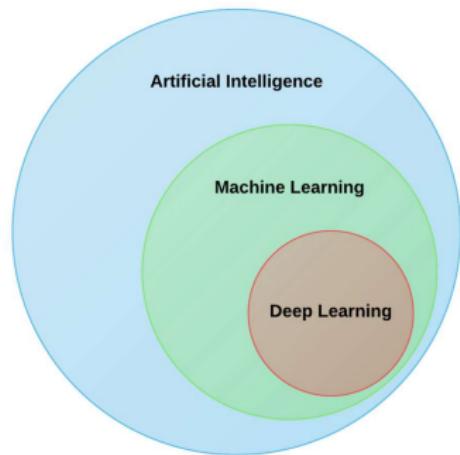
Principe

Aplicaciones

Los recursos

El curso

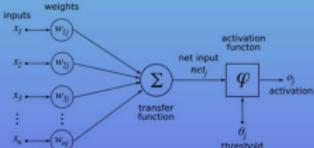
## Qué es Deep learning?



## Redes neuronales no siempre fueron profundas

**1943** El primer modelo matemático de una neurona (McCulloch-Pitts)

**1957** El perceptrón



**1959** Hubel y Wiesel descubrieron las células simplex y complex en sistemas de visión biológica.

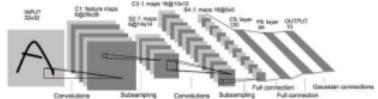
**1965** Primera red profunda (8 capas) – Ivakhnenko and Lapa

**1979** Red neuronal para reconocer patrones visuales (Neocognitron) – Fukushima

**1982** Primera red recurrente – Hopfield

**1986** Algoritmo Backpropagation

**1989** Redes neuronales convolucionales – Handwritten recognition



**1989** Reinforcement learning – Q-learning

## Redes neuronales no siempre fueron profundas (no tanto)



**2011** Alexnet

**2012-2015** Arquitecturas mejoradas para reconocimiento visual –

**2014** Redes generativas adversarias (GAN)



**2015 ~** Explosión de la industria DL

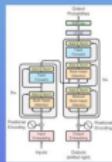


# Historia

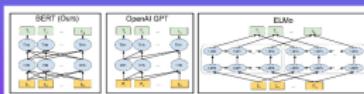
## Redes neuronales profundas

**2014 Mecanismo de atención – Bahdanau et al.**

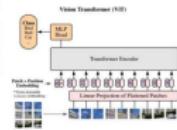
**2017 Arquitectura Transformer – Machine translation.**



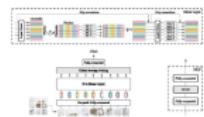
**2018-2020 Modelos de NLP: BERT, GPTs, XLM**



**2020 Vision Transformer**

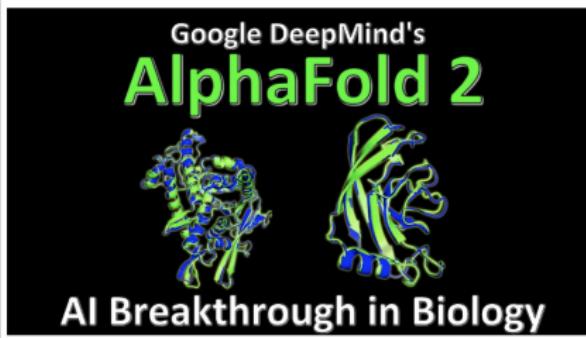
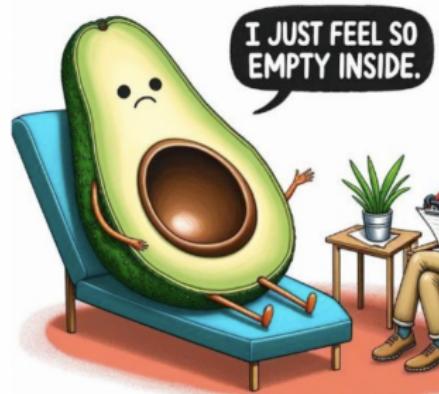
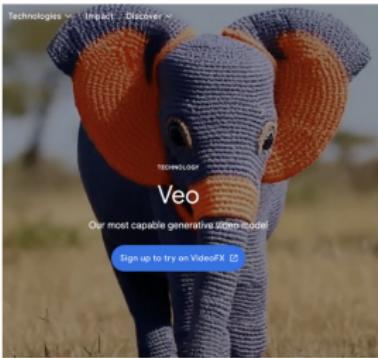


**2021 MLP Mixer**



**2022 Dall-e, Imagen, LaMDA**

# Y ahora eso



# Outline : Principe

---

Significacion de las termas

Historia

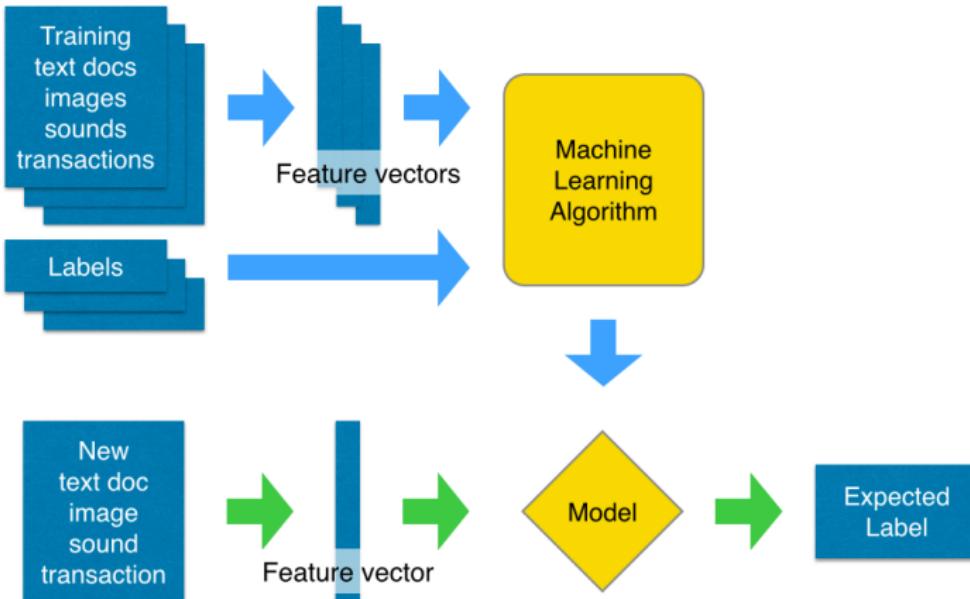
Principe

Aplicaciones

Los recursos

El curso

# Supervised Machine Learning

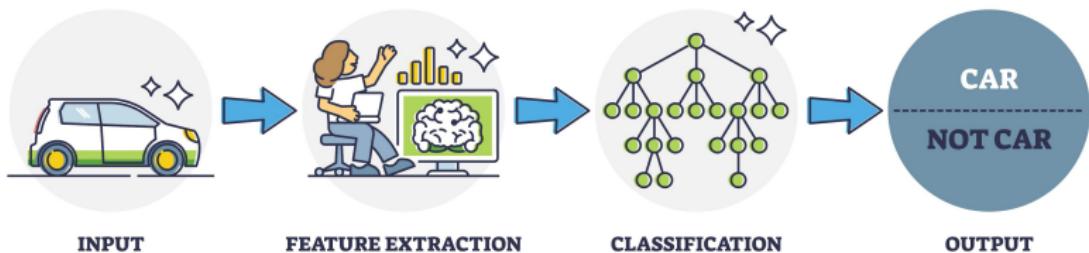


Predictive Modeling Data Flow

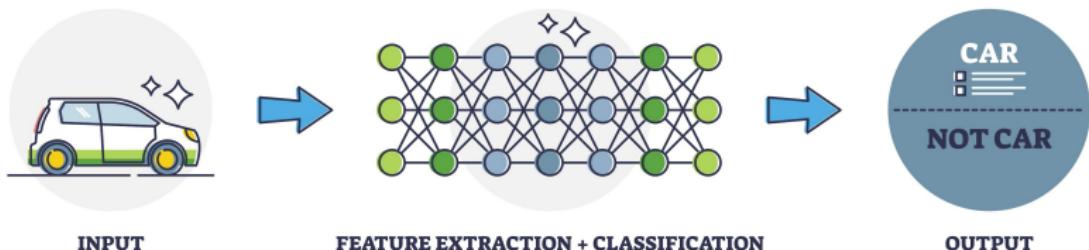
ML: Es necesario **crear los vectores de características** para utilizarlos como entrada.

# Diferencia entre ML y DL: Extracción de features

## MACHINE LEARNING



## DEEP LEARNING



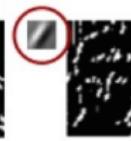
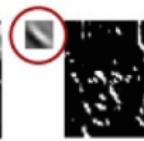
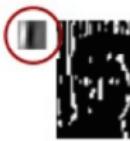
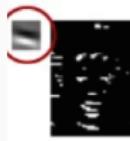
## Complejidad de las representaciones: Ejemplo



Layer 3 activation (coefficients)



Layer 2 activation (coefficients)

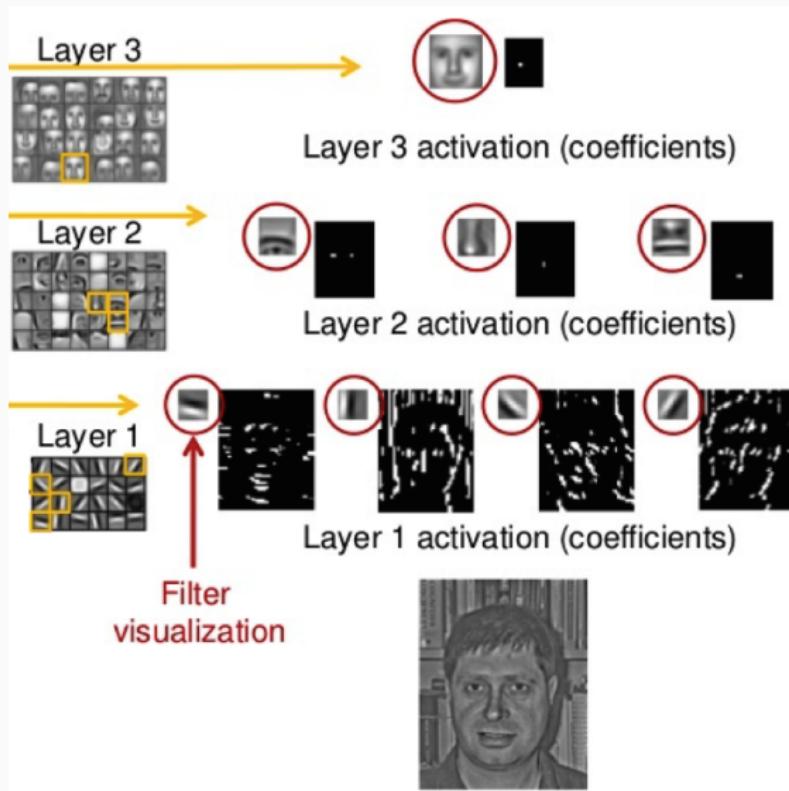


Layer 1 activation (coefficients)



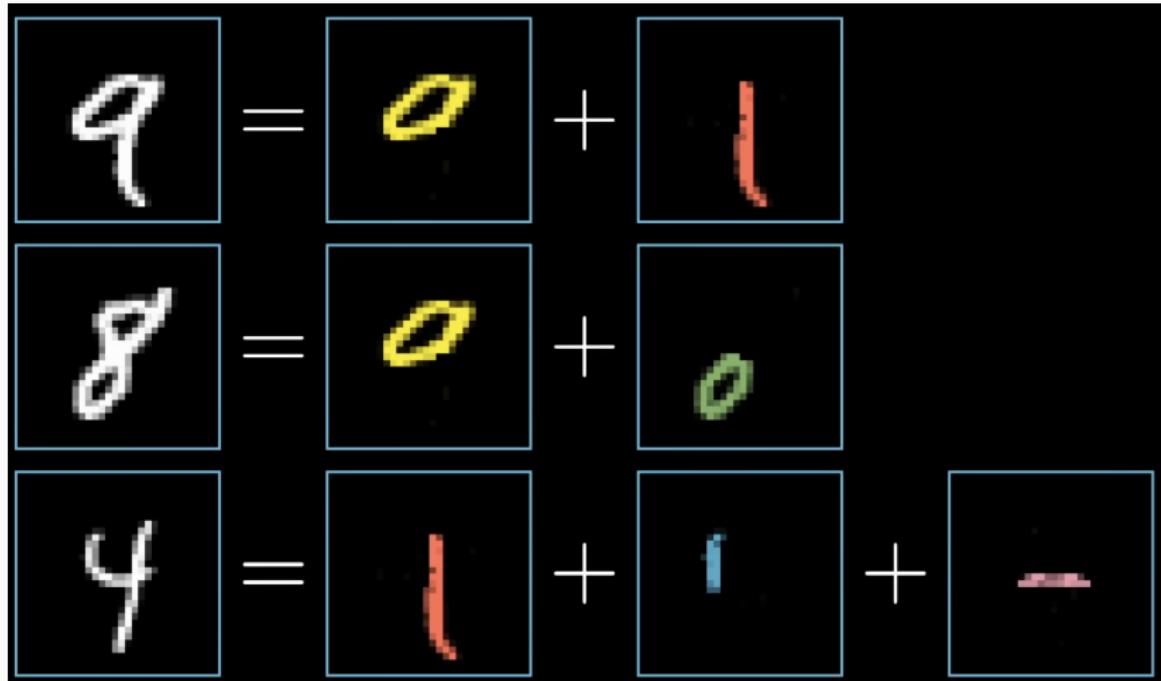
DL **extrae los vectores de características de datos brutos**

# Complejidad de las representaciones: Composicionalidad



DL extrae los vectores de características de datos brutos y **los combina**

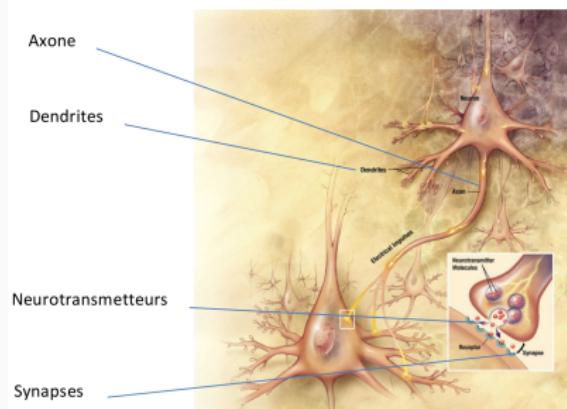
## Complejidad de las representaciones: Composicionalidad



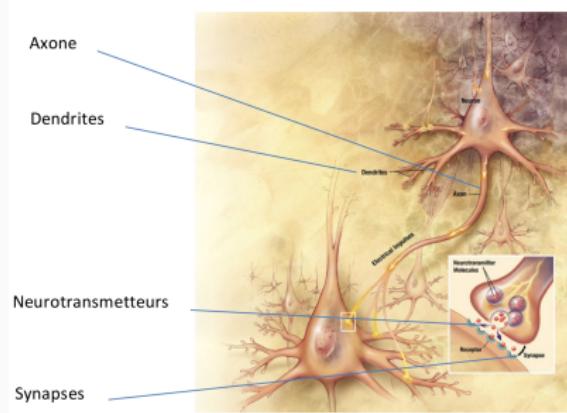
Mas informacion en [este video](#).

# Primero, ¿por qué este nombre?

Una neurona recibe varias informaciones (neurotransmisores) a nivel de sus **dendritas**. Estos neurotransmisores son liberados por las **sinapsis**. Cuando la cantidad de información supera un cierto **umbral**, la neurona se "activa", enviando una corriente eléctrica a través de su axón, lo que le permite **emitir a su vez** neurotransmisores a través de sus sinapsis.



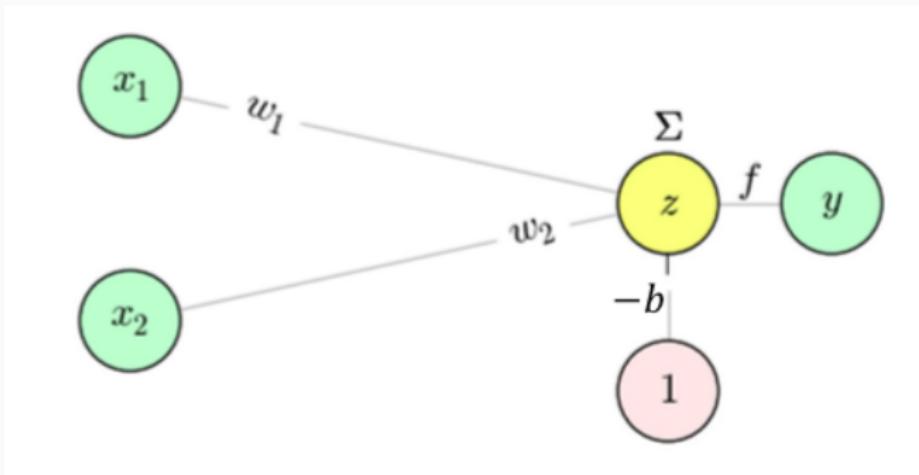
# Primero, ¿por qué este nombre?



## Para resumir:

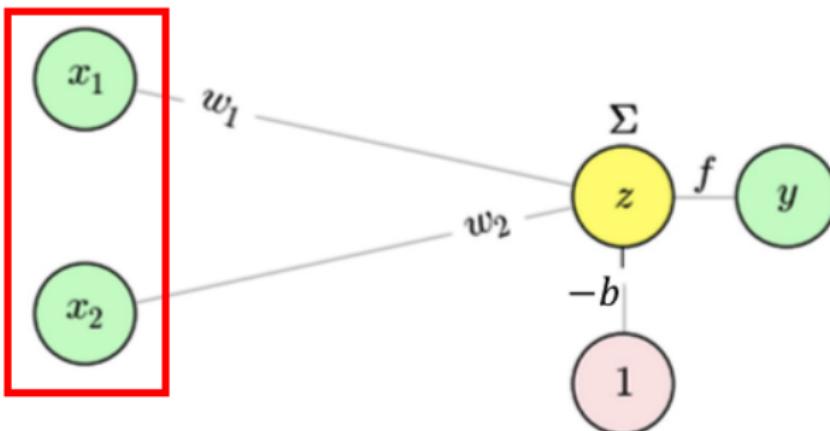
- Las salidas de una neurona son las entradas de otra.
- Una neurona emite cuando recibe una cantidad de información que supera un umbral.
- La cantidad de información emitida a la siguiente neurona es gestionada por las sinapsis, que se activan a partir de un umbral.

# Perceptrón – Presentación



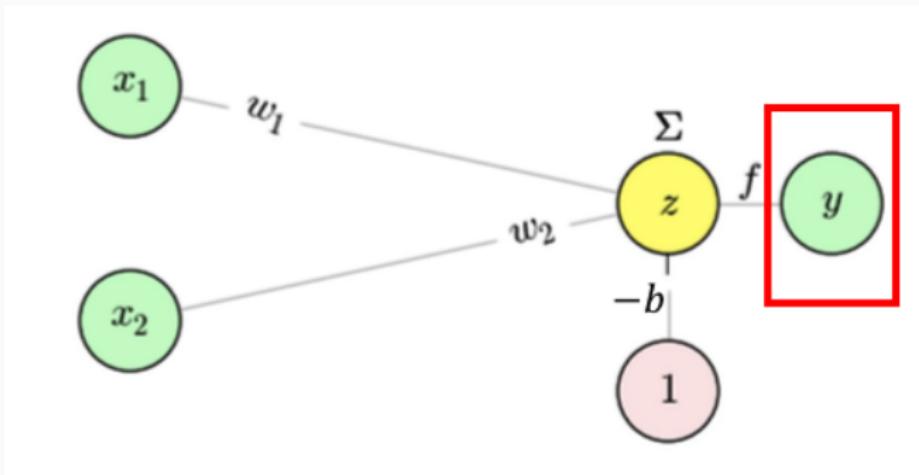
$$y = f(x_1 w_1 + x_2 w_2 - b)$$

# Perceptrón – Presentación



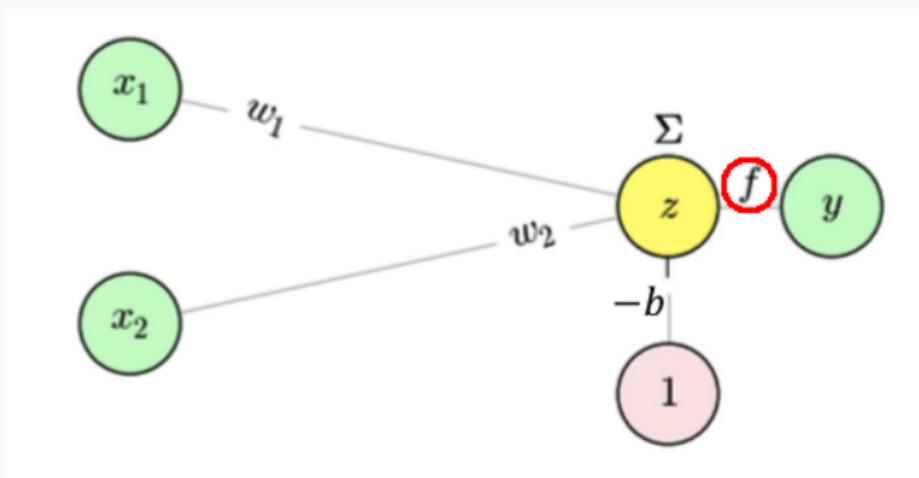
$$y = f(x_1 w_1 + x_2 w_2 - b)$$

# Perceptrón – Presentación



$$y = f(x_1 w_1 + x_2 w_2 - b)$$

# Perceptrón – Presentación



$$y = f(x_1 w_1 + x_2 w_2 - b)$$

En teoría  $f$  es la función de Heaviside:  $f(z) = \mathbb{1}_{\mathbb{R}_+} = \begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{si } z < 0 \end{cases}$

# Outline : Aplicaciones

Significacion de las termas

Historia

Principe

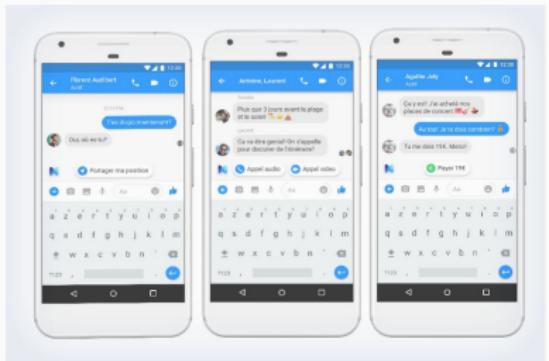
Aplicaciones

Los recursos

El curso

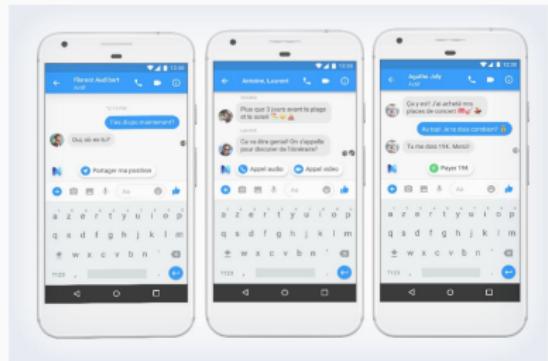
# Aplicaciones (I/II)

- Detección de eventos en un texto



# Aplicaciones (I/II)

- Detección de eventos en un texto

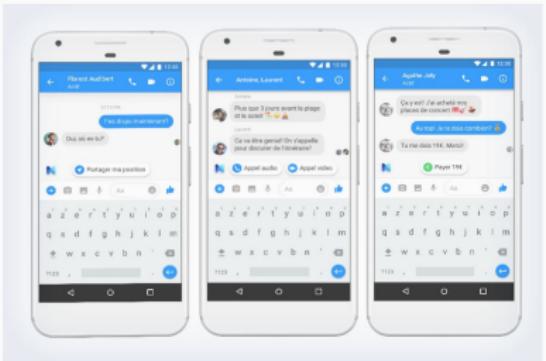


- Procesamiento automático de opiniones de usuarios



# Aplicaciones (I/II)

- Detección de eventos en un texto



- Procesamiento automático de opiniones de usuarios



Les connaissez-vous ?



**Marina Dunion**  
Digital Marketing @Air France  
& Co-Founder @FlexiFly  
● Teddy Viraye-Chevalier et 3 autres relations



**Salvatore Anzalone**  
Post-Doc at ISIR, University Pierre et Marie Curie, Paris  
● Thomas Janssone et 2 autres relations

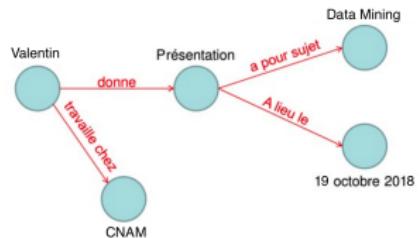


**Halla Olafsdottir**  
Medical Solutions Project Manager | Chef de Projet  
● Télécom ParisTech

- Propuesta de recomendaciones a un usuario

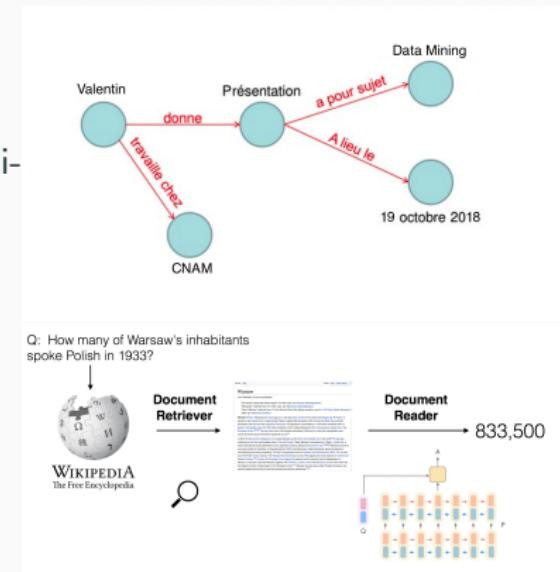
## Aplicaciones (II/II)

- Detección de relaciones entre entidades en un texto



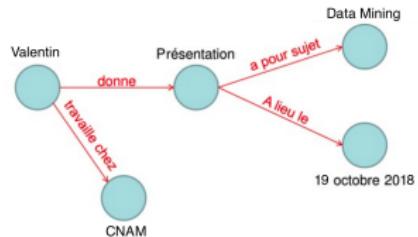
# Aplicaciones (II/II)

- Detección de relaciones entre entidades en un texto
- Respuesta a una pregunta

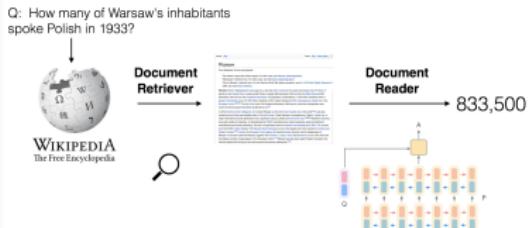


# Aplicaciones (II/II)

- Detección de relaciones entre entidades en un texto



- Respuesta a una pregunta

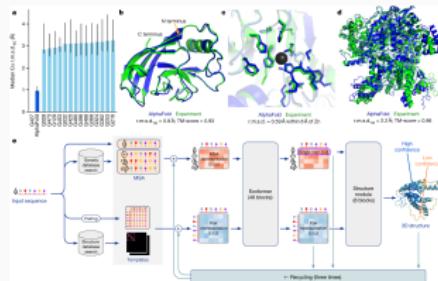


- Módulo de IE para un agente conversacional



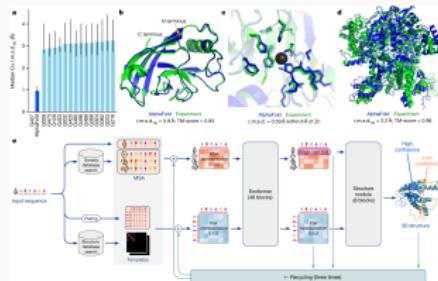
# Significación del trabajo: Porque hacer eso?

- Avance científico



# Significación del trabajo: Porque hacer eso?

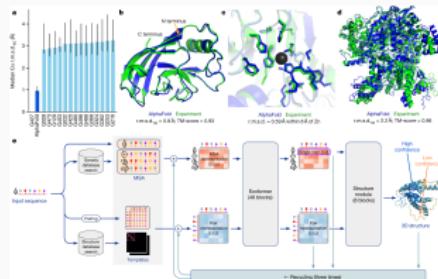
- Avance científico



- Prevención y gestión de desastres naturales

## Significación del trabajo: Porque hacer eso?

- Avance científico



- Prevención y gestión de desastres naturales

- Impacto en la salud pública



Open Chronic

## **Améliorer la prise en charge des malades chroniques**

Bivariate distributions

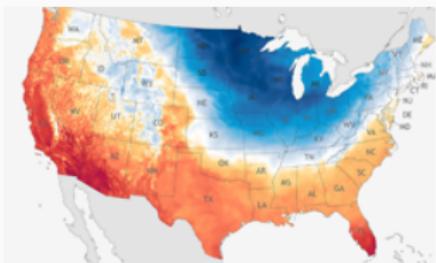
Ministère de la santé, Direction de la recherche, des études, de  
et des statistiques



#### **Beta selection**

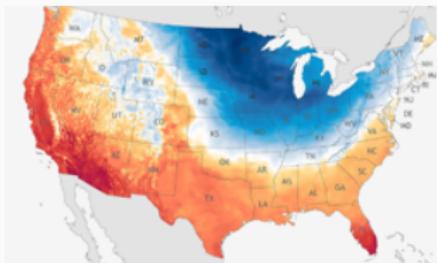
# Significación del trabajo: Porque hacer eso?

- Sostenibilidad ambiental

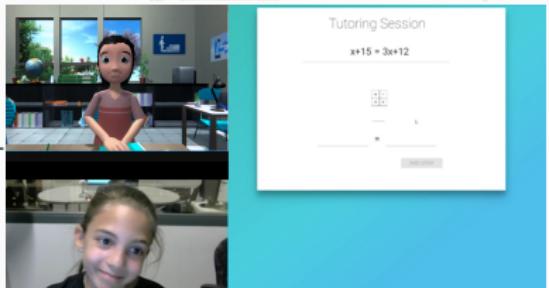


# Significación del trabajo: Porque hacer eso?

- Sostenibilidad ambiental

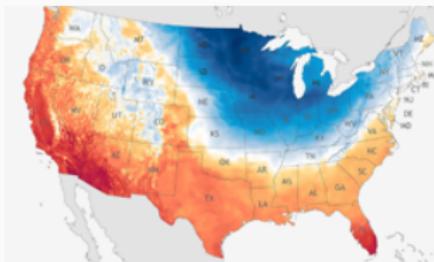


- Impulso a la educación y la investigación

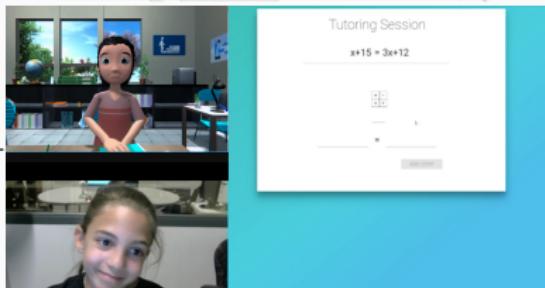


# Significación del trabajo: Porque hacer eso?

- Sostenibilidad ambiental



- Impulso a la educación y la investigación



- Democracia participativa



# Outline : Los recursos

---

Significacion de las termas

Historia

Principe

Aplicaciones

**Los recursos**

El curso

# Material

- Computadora
- Jupyter Notebook y Anaconda:  
<https://www.anaconda.com/download/>
- Los notebooks y las cheatsheets disponibles online:

**Python For Data Science Cheat Sheet**  
NumPy Basics  
Learn Python for Data Science interactively at [www.DataCamp.com](http://www.DataCamp.com)

### NumPy

The NumPy library is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Use the following import convention:

```
>>> import numpy as np
```

### NumPy Arrays



### Creating Arrays

Learn more about NumPy arrays at [www.DataCamp.com](#)

**Python For Data Science Cheat Sheet**  
Pandas Basics  
Learn Python for Data Science interactively at [www.DataCamp.com](http://www.DataCamp.com)

### Pandas

The Pandas library is built on NumPy and provides easy-to-use data structures and data analysis tools for the Python programming language.

pandas

Use the following import convention:

```
>>> import pandas as pd
```

### Pandas Data Structures

#### Series

A one-dimensional labeled array capable of holding any data type



**Python For Data Science Cheat Sheet**  
Matplotlib  
Learn Python interactively at [www.DataCamp.com](http://www.DataCamp.com)

### Matplotlib

Matplotlib is a Python 2D plotting library which produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms.

#### 1) Prepare the Data

##### 1D Data

```
>>> import numpy as np  
>>> x = np.linspace(0, 10, 100)  
>>> y = np.cos(x)  
>>> z = np.sin(x)
```

2D Data or Images

```
>>> data = 2 * np.random.random((100, 100))  
>>> data -= data.mean() / 100  
>>> Y, X = np.mgrid[-3:3:100j, -3:3:100j]  
>>> Z = np.exp(-X**2 - Y**2)  
>>> Z = Z / Z.max()
```

##### Also see Lists & NumPy

### 4) Colors

#### Colors

##### ax

##### plt

##### ax

##### fig

##### im

##### Marker

##### sc

### 5) Markers

#### Marker

##### sc

##### marker

##### markeredgecolor

##### markerfacecolor

##### markeredgecolor

# Recursos

- El Calendario y las slides en el github
- Libros
  - Deep learning, Goodfellow, Bengio, Courville ( <https://www.deeplearningbook.org/>)
  - Neural networks and Deep learning, Nielsen ( <http://neuralnetworksanddeeplearning.com/>)
  - Dive into Deep Learning. Zhang, Lipton, Li, Smola ( <https://d2l.ai/>)
- Otros
  - CS231n - Stanford: Deep Learning for Computer Vision ( <http://cs231n.stanford.edu/>)
  - CS224N: Natural Language Processing with Deep Learning ( <https://web.stanford.edu/class/cs224n/>)
  - Deep learning – New York University ( <https://atcold.github.io/pytorch-Deep-Learning/>)
  - CS224W: Machine Learning with Graphs ( <http://web.stanford.edu/class/cs224w/>)
- Canal Discord: <https://discord.gg/hUJqa4AHCd>

Biblioteca de Python para Deep Learning (Redes Neuronales Profundas).  
Es de alto nivel que corre sobre Tensorflow y otros backends.



- Visión: [Tutorial de Keras para afinar un VGG16 preentrenado](#), que puede ser utilizado con:  
[Diferentes modelos de CNN preentrenados disponibles en Keras](#)
- Texto y audio:  
[Tutorial RNN-LSTM Seq2seq para traducción automática](#)
- Texto: [Uso de embeddings de palabras preentrenados](#)

Biblioteca de Python para Deep Learning (Redes Neuronales Profundas)  
concurrente de Tensorflow



- Visión: [Tutorial para afinar un ResNet18 preentrenado](#)
- Audio: [Reconocimiento de voz con Wav2Vec2](#)
- Texto: [Tutorial RNN-GRU Seq2seq para traducción automática](#)

# HuggingFace's Transformers

Biblioteca de Python para **Transformers** (Tipo de Modelos de Redes Neuronales Profundas)



## Hugging Face

- Muchos modelos ya pre-entrenados, para imágenes, audio, multimedial, texto, ...
- Modelos clásicos, modelos generativos, embeddings, ... hasta los más grandes y nuevos (tipo LLaMA3-70B)
- Otras librerías: Diffuser, Datasets, Accelerate, PEFT, bitsandbytes, TRL, ...

# Outline : El curso

---

Significacion de las termas

Historia

Principe

Aplicaciones

Los recursos

**El curso**

# Evaluaciones

---

La evaluacion de este clase sera hecho con 4 o 5 Tareas que tendran que hacer en la casa. **Cuidado: las ponderaciones de las tareas varian con su dificultad!**

# El programa: Que quieren ver después las bases?

Todo no está definido! El año pasado, he añadido dos clases:

- **Modelización del lenguaje**: de Shannon al chatbot basado en LLM.
- **Modelos multimodales**: clásicos, CLIP, difusión, basados en LLM, generativos para visión o audio.
- **Modelos generativos de texto grandes (LLM)**: instrucciones, RLHF, agente que utiliza herramientas, generación aumentada recuperada.

Este año podemos hacer una otra nueva:

- **Modelos de audio**: reconocimiento de voz, separación de fuentes, clasificación de escenas de audio, modelos generales.
- **Eficiencia**: cómo entrenar/implementar un LLM con un presupuesto reducido (ajuste fino eficiente en parámetros, cuantificación, bibliotecas específicas optimizadas como vLLM).
- **Razonamiento**: Técnicas avanzadas de *prompt engineering* y métodos para mejorar las capacidades de razonamiento de los LLMs.

**Questions?**

## References i