Datos Masivos y la contribución de la Estadística a la Inteligencia Artificial

Daniel Peña

Universidad Carlos III de Madrid

20 de octubre: Día Mundial de la Estadística

Día de la Estadística en Cataluña, Universidad Autónoma de Barcelona

Índice

1. Datos y ciencia empírica.
2. Avances en su tratamiento.
3. La Estadística y los datos masivos.
4. El Aprendizaje Automático.
5. La Inteligencia Artificial.
6. Los Métodos híbridos.
7. Conclusiones.

Agradecimientos

- Llorenç Badiella, Universidad Autónoma de Barcelona
- Montserrat Guillen, Universidad de Barcelona
- Isabel Serra, Presidenta de la SoCE

1. Datos y ciencia empírica

Los datos sobre sucesos inciertos han sido la materia prima del conocimiento desde finales del siglo XV.

La recogida de datos de fenómenos aleatorios o inciertos se inicia en el renacimiento, al descubrir la regularidad en el azar (sexo al nacimiento, muertes, accidentes, lluvia). Con ellos se inicia la ciencia empírica.

Regularidad en el azar y la incertidumbre : G. Cardano *"Liber de ludo aleae"* 1553.

Copérnico (1473-1543) y su sistema heliocéntrico impulsa la medición en Astronomía.

En 1640 Pascal y Fermat crean el cálculo de probabilidades.

En el siglo XVIII, el empirismo británico (Berkeley, Hume) afirma que el conocimiento se genera principalmente buscando patrones y regularidades en los datos (Halley, Flanking, Volta, Boyle, Linneo, Jener).

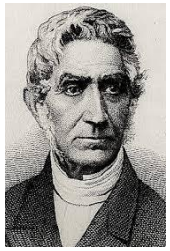
Gerolamo Cardano

Pascal y Fermat

Breve historia del análisis de datos

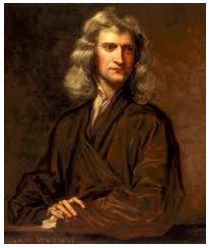
Demográficos 1662

J. Graunt (1620-74)



A. Quetelet (1796-1874)

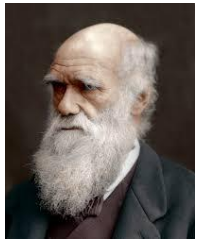
Astronómicos 1687



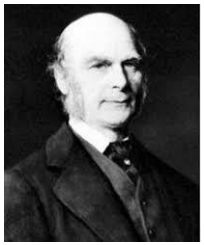
I. Newton (1643-1727)

C. Gauss (1777-1855)

Biológicos 1859



Ch. Darwin (1809-82)



F. Galton (1822-1911)

Médicos 1860

F. Nightingale (1820-1910)

D. Cox (1924-2022)

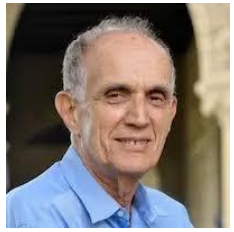
Agronomía e Industria 1925

R.Fisher (1890-1962)

G. Box (1919-2013)

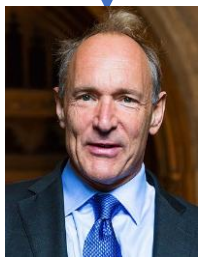
Simulados con ordenadores 1948

J. Von Neuman (1903-57)



B. Efron (1938-)

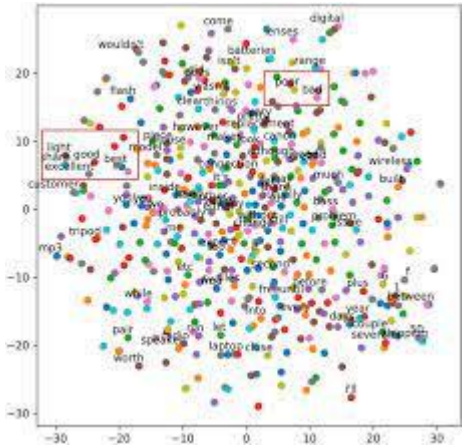
Automáticos y masivos 2000

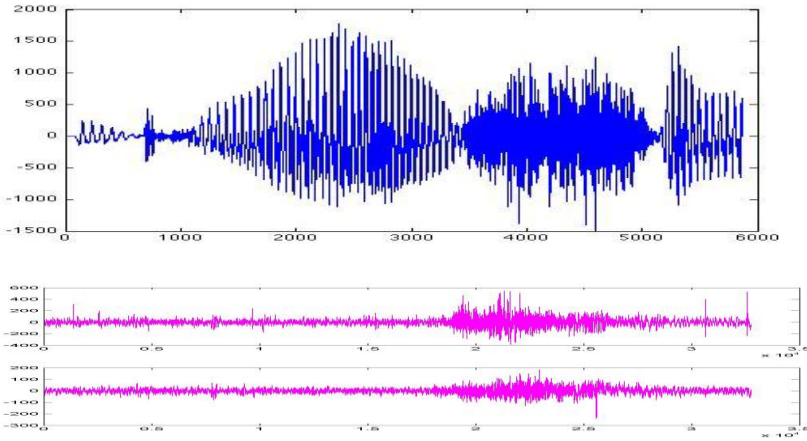


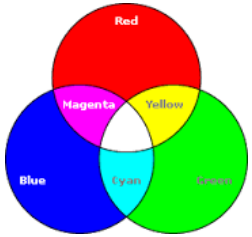
T. Berners-Lee (1959-)

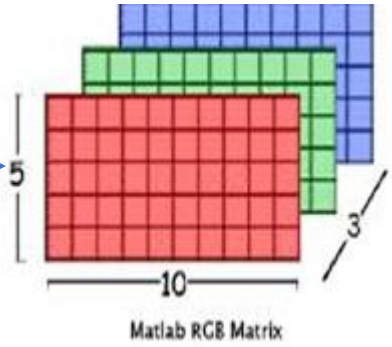
G. Hinton (1947-)

Los nuevos datos





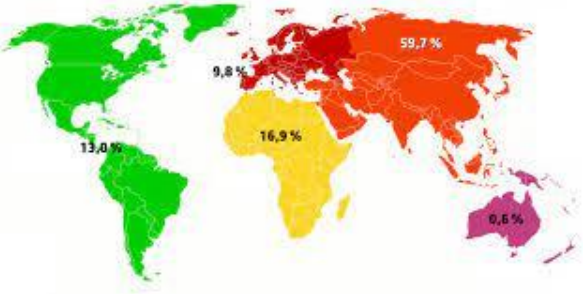




Datos digitales

Tamaño de los nuevos datos

- Ahora cada día se genera 400ZettaB de datos, una cantidad equivalente a que cada ser humano (8.000 millones) generara cada día 50TB.
- La biblioteca del congreso de EEUU tiene unos 20TB
- Todos los libros editados jamás en el mundo (unos 400TB).
- 1TB= 10^12B
- 1ZettaB= 10^21B.



2. Los datos y los avances en TIC

Grandes avances en las tecnologías de la información y las comunicaciones para transmitir, almacenar y procesar grandes masas de datos.

En este siglo el almacenamiento personal ha cambiado:

- Un PC en 2000: 10 GB (10.000 libros, 2.500 canciones, 5/10 películas) (GB= $10^9 bytes$)

- Un iphone 14, 2022 : 1TB (100 veces más que 10GB, 1 millón de libros) (TB= $10^{12} bytes$)

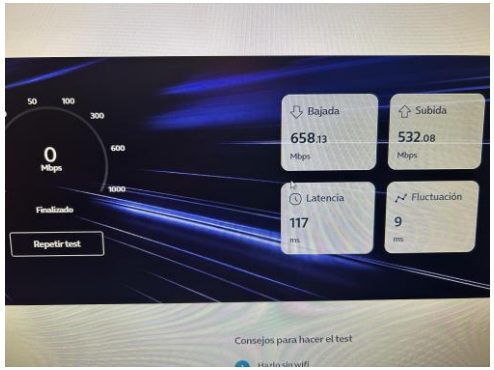
- Un PC actual: 10TB (media librería del Congreso de EEUU)

Almacenamiento personal en este siglo multiplicado por 10^3

Avances en transmisión de datos

Transmitir: Velocidad

- En 2000 1 MBps (MBps= $10^6 bytes/seg$)
- 2010 10 MBps
- 2025 1 GBps (GBps= $10^9 bytes/seg$)
- Hoy en Japón: 319 Terabits (Tbps= $10^{12} bits/seg$) en distancia de 3.000 kilómetros.



Velocidad de transmisión en este siglo multiplicado por 10^3

Avances en velocidad de cálculo:

- Velocidad de un PC en **2000:** 2x10^9 flops, (2 Gfps): ¼ de la población del mundo (8x10^9 $personas$) con calculadora
- La velocidad de un iphone 14 en **2022** es de 2 tera-flops=2x10^{12} flops: casi mil planetas con la población de la tierra calculando.
- PC CES **2025**(NVIDIA): 300TFLOP= 3x10^{15} flops: 1 millón de planetas calculando.

Velocidad de cálculo de un PC en este siglo multiplicado por más de 10^6

- El ordenador más rápido 2025 Cray 1.7x10^{18} flops (mil millones de planetas como la tierra, todos los de nuestra Galaxia)

3. La estadística y los datos masivos

- La Estadística ha creado los conceptos necesarios para aprender de los datos: representaciones gráficas, diseñar experimentos, construir modelos, estimar parámetros, cuantificar la incertidumbre, validar modelos y combinar muchas fuentes de información.

- Los métodos estadísticos han sido la herramienta principal de creación de conocimiento científico en el siglo XX, transformado datos numéricos interpretables en información mediante un análisis "experto y artesanal" de los mismos.

- Para trabajar de forma automática con millones de datos digitalizados con estructura compleja y no intuitiva hay que adaptar algunos métodos y desarrollar otros nuevos. Varios conceptos básicos (distribuciones de las variables, contrastes de hipótesis, modelos paramétricos lineales para relacionar variables, etc.) no son apropiados o tienen que reformularse, y desarrollar nuevos enfoques más generales que aprovechen la gran capacidad de cálculo disponible.

4. Los Métodos de Aprendizaje Automático (Machine Learning)

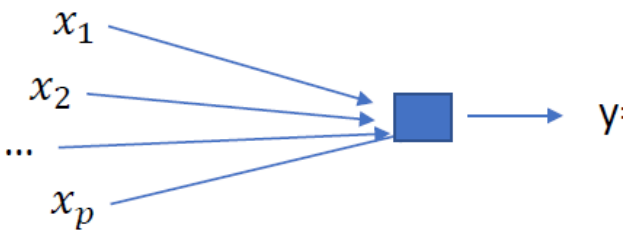
Métodos automáticos **para prever, clasificar y agrupar** datos masivos sin hacer ninguna hipótesis sobre ellos y basados en cálculos intensivos para:

	Métodos
1. Prever o clasificar con modelos no lineales con muchas variables (predicción y clasificación supervisada).	Árboles de clasificación y regresión (EST) Combinación de modelos (Ensemble Methods) (EST) Máquinas de vector soporte **Redes neuronales profundas**
2. Agrupar cualquier conjunto de datos digitales en grupos homogéneos (clusters, o clasificación no supervisada).	K- medias y otros métodos de estadística Métodos de procesamiento de lenguaje natural **Redes neuronales profundas**

Una neurona : varias variables de entrada y una de salida

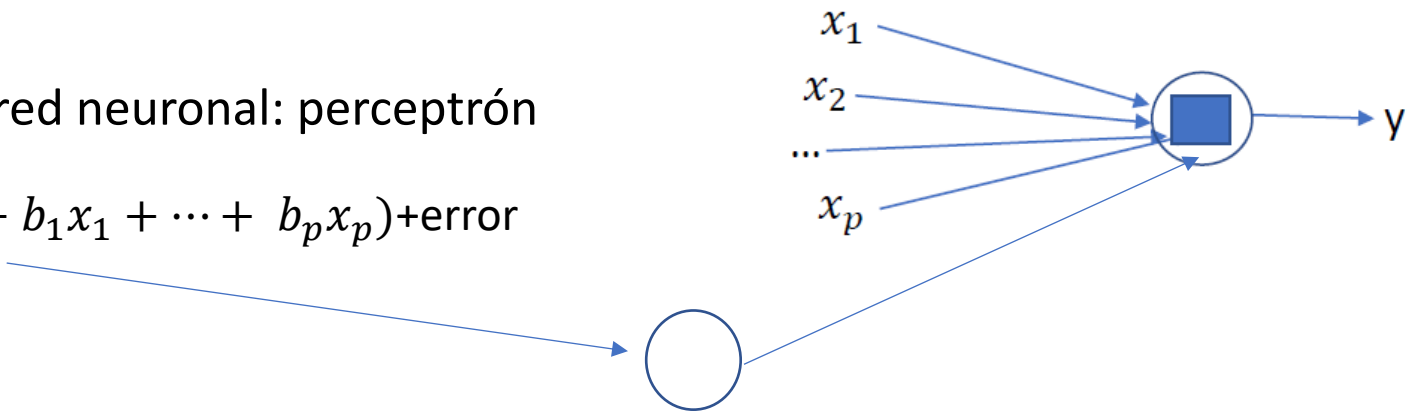
Regresión

$$y = a + b_1 x_1 + \cdots + b_p x_p + \text{error}$$

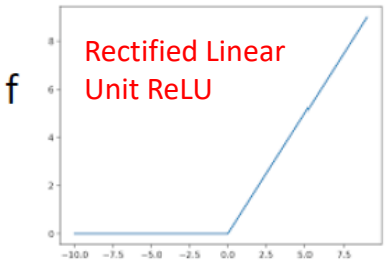


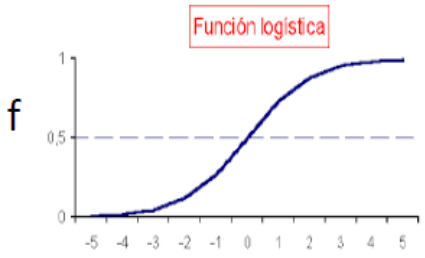
Unidad de red neuronal: perceptrón

$$y = f(a + b_1 x_1 + \cdots + b_p x_p) + \text{error}$$



Relación no lineal





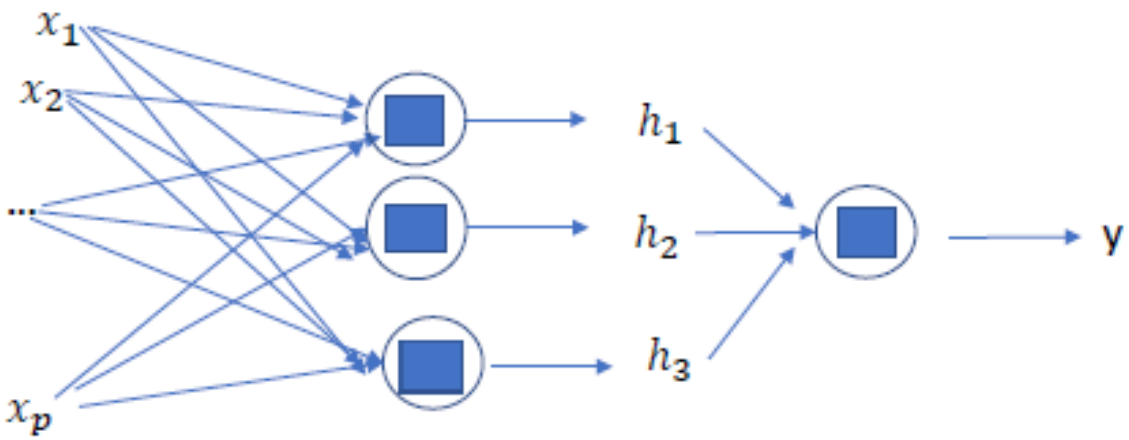
El perceptrón: varias neuronas

Los parámetros se estiman minimizando los errores de predicción

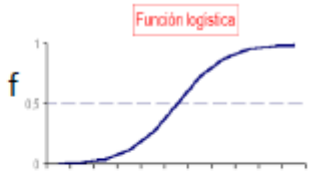
$$y=f(b_0 + b_1h_1 + b_2h_2 + b_3h_3)$$

$$h_i = f(a_o + a_{i1}x_1 + \cdots + a_{ip}x_p)$$

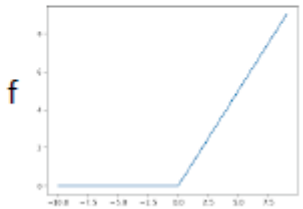
i=1,2,3



Logística para clasificación



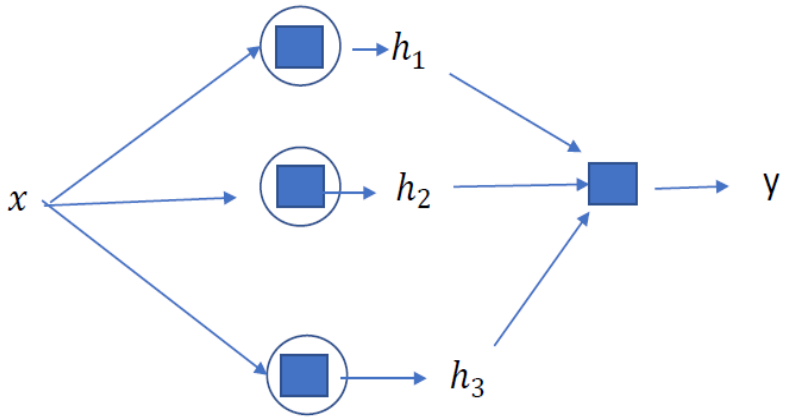
Lineal rectificada (Rectified Linear Unit /RELU)



Las RNA y las relaciones no lineales

Ejemplo: una variable de entrada, x , 3 neuronas con salida ReLU y una variable de salida

$$y=\varphi_1 h_1 + \varphi_2 h_1 + \varphi_3 h_1$$



h_i =RELU($\theta_{0i} + \theta_{1i} x$)
i=1,2,3

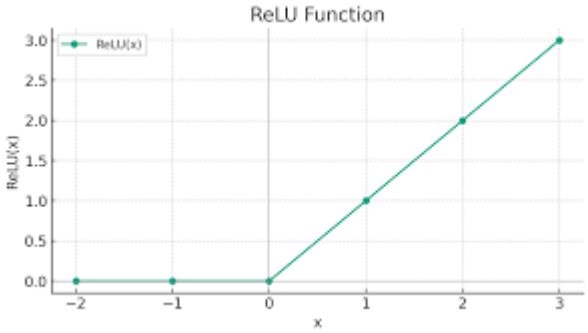
Da como salida una combinación Lineal de las variables de entrada

x ⟶ $\theta_0 + \theta_1 x$

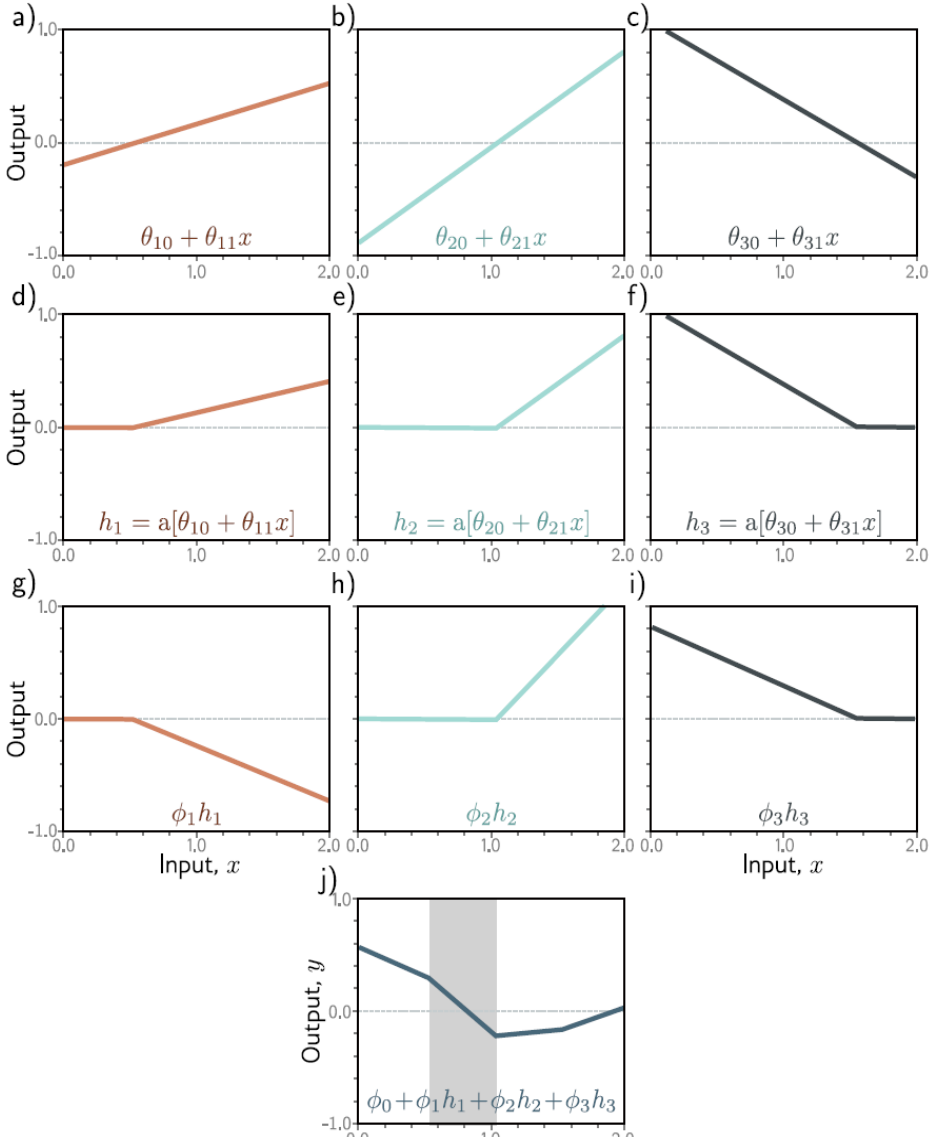
Transforma la combinación lineal de forma no lineal

h= $\theta_0 + \theta_1 x$ si $(\theta_0 + \theta_1 x) \geq 0$

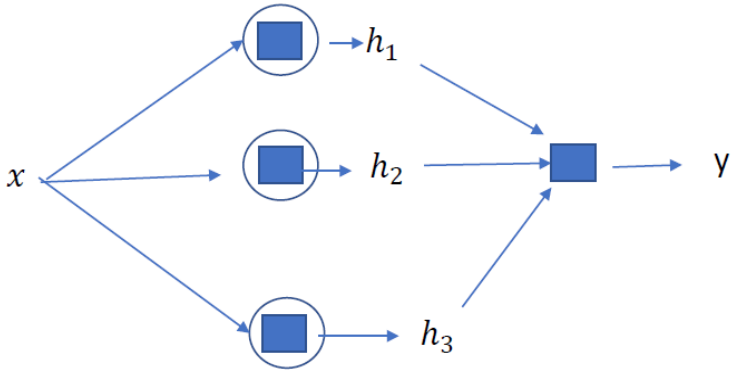
h=0 si $(\theta_0 + \theta_1 x) < 0$



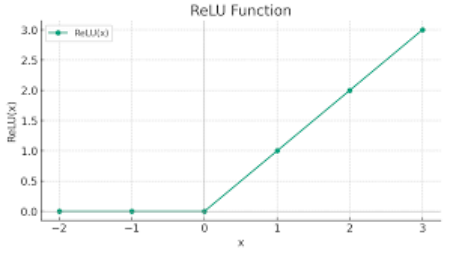
Ejemplo : una variable de entrada, $0 \leq x \leq 2$ y 3 neuronas con ReLU



$$y = \emptyset_1 h_1 + \emptyset_2 h_2 + \emptyset_3 h_3$$

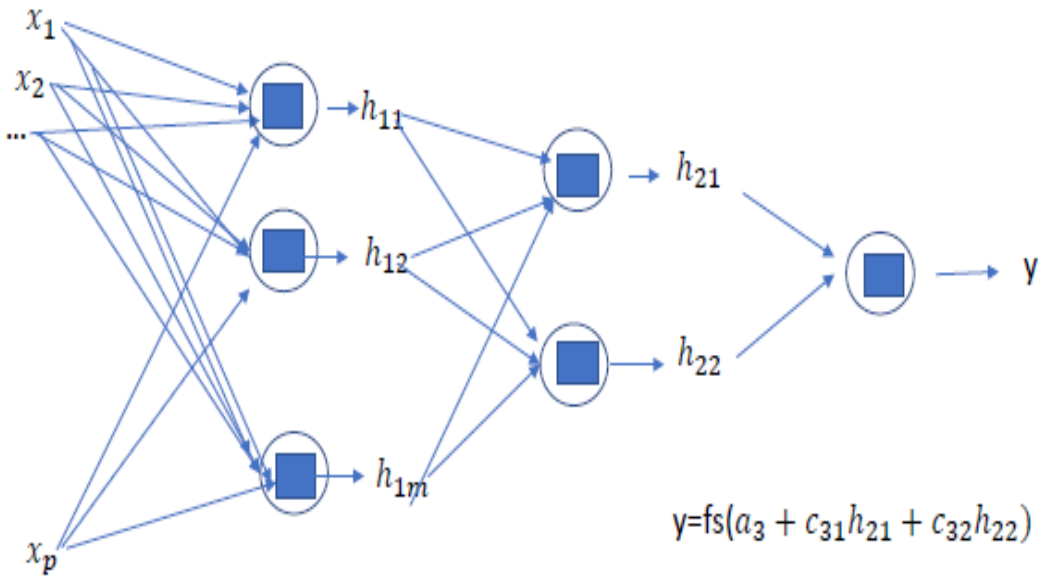


h_i =RELU($\theta_{0i} + \theta_{1i}x$)
i=1,2,3



Aprendizaje Profundo(Deep Learning):

Red neuronal profunda : varias capas intermedias modela relaciones no lineales con varias capas de neuronas, cada una con varias neuronas, y una capa final. Aproximan cualquier relación no lineal entre la entrada y la salida.



$$y=fs(a_3 + c_{31}h_{21} + c_{32}h_{22})$$

$$h_{2j} = f(b_{oj} + c_{1j}h_{11} + c_{2j}h_{12} + \cdots + c_{2m}h_{1m}),$$
$$J=1,2$$

$$h_{1i} = f(c_{0i} + c_{1i}x_1 + c_{2i}x_2 + \cdots + c_{pi}x_p),$$
$$i = 1,2,\ldots,m$$

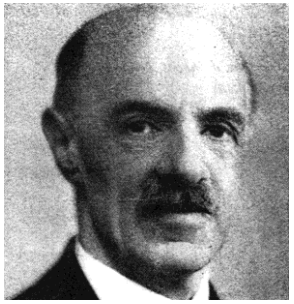
Problema:

Con muchas neuronas y capas es muy difícil interpretar la relación entre las variables de entrada y la salida

Ideas fundamentales de RR:

1. Utilizar en lugar de las variables originales ciertas combinaciones lineales como variables latentes no necesariamente ortogonales.

- Variables latentes o factores. Introducidos por Spearman para el análisis factorial de la inteligencia



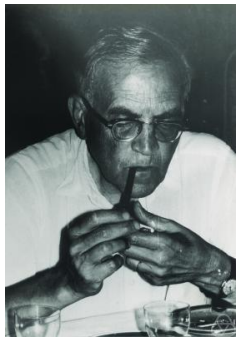
Charles Spearman
Londres, 1863-1945

- Componentes principales: nuevas variables combinaciones lineales de las originales con máximo poder predictivo del conjunto. Introducidas por Hotelling con variables estáticas.

Harold Hotelling
EEUU, 1895 -1973

2. Suponer siempre relaciones no lineales y aproximarlas por combinaciones de funciones no lineales

- Aproximar relaciones no lineales mediante polinómicos (splines) fue propuesta hacia 1940 por I. J. Schoenberg.
- En estadística los splines han sido estudiados por Grace Wahba (The International Prize in Statistics 2025)
- Ambos en University Wisconsin-Madison



Pero la idea del aprendizaje profundo: aproximar la no linealidad por superposición de muchas neuronas en distintas capas, es nueva.

Geoffrey Hinton (1947-)
PN Física 2024, P. Turing

Algunos tipos de redes neuronales

•**Redes Neuronales Artificiales (ANN) o Feed-Forward NN**: Aprenden a partir de datos para realizar tareas de clasificación, regresión, reconocimiento de patrones, etc.

•**Redes Convolucionales (CNN)**: Especializadas en procesar datos con estructura espacial, como imágenes y videos. Aplican filtros para encontrar características de la imagen.

•**Redes Recurrentes (RNN)**: Para secuencias de datos, como texto, series temporales o voz. La predicción depende de valores presentes y pasados.

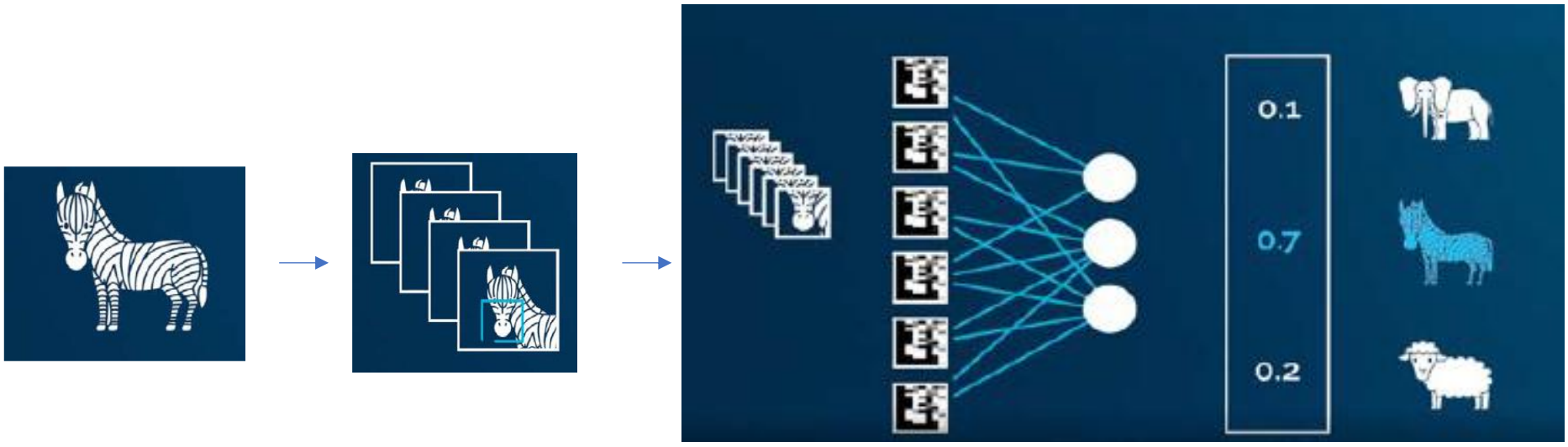
•**Transformers**: Modelos que manejan secuencias digitales con un mecanismo de relación entre ellas de "atención", muy utilizados en procesamiento de lenguaje natural.

Redes neuronales convolucionales para el reconocimiento de imágenes

1. Analizar imágenes buscando “patrones” (círculos, rectas, contornos, etc).
2. Aprenden patrones útiles para clasificar por entrenamiento que se representan

por bloques pequeños de pixeles que se aplican sobre la imagen para obtener una nueva por convolución (cada pixel se suaviza por los que le rodean de acuerdo al patrón o filtro aplicado).

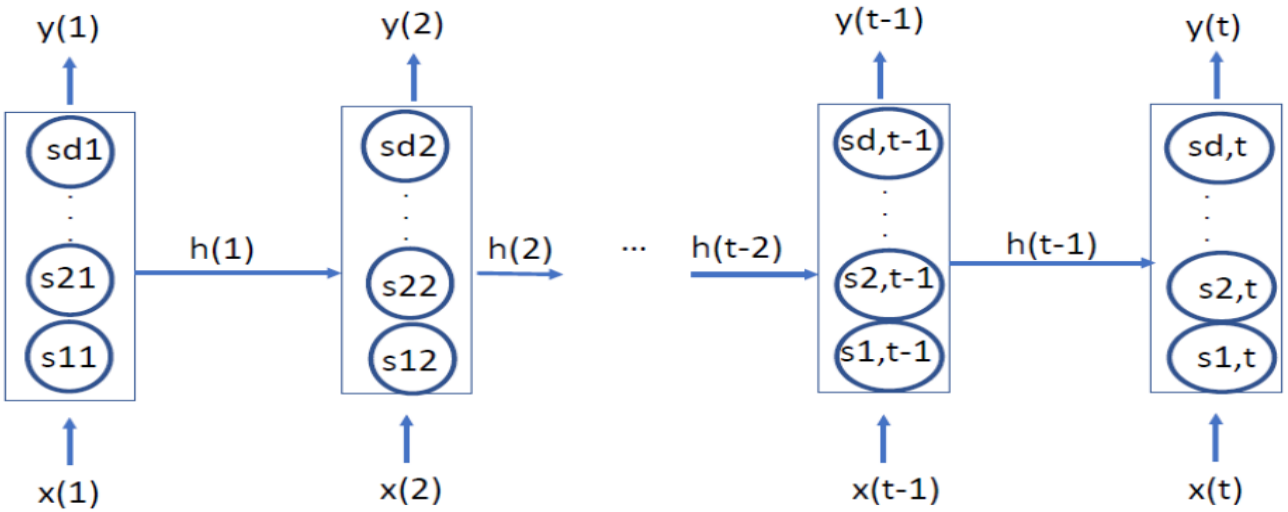
3. Se buscan en paralelo muchos “patrones” cada uno por su filtro.
4. El conjunto de “patrones” encontrados en la imagen, el mapa de caracteres, se vectoriza y clasifica con una red FF de acuerdo con las imágenes guardadas.



Recurrent Neural Networks

- Los datos se procesan secuencialmente
- La predicción en t depende de los datos en t-1 pero también de los anteriores con memoria decreciente

Llamando y(t) a la predicción de x(t) $y(t)=f(x(t))+g(y(t-1)$



Las redes RNN clásicas tiene dificultades para incorporar dependencias con muchos retardos. Las **redes Short Long Term Memory (LSTM)** incorporan una celda de memoria y tres compuertas (olvido, entrada, salida) que controlan el flujo de información y permiten conservar datos relevantes durante períodos más largos (véase por ejemplo, Peña and Tsay (2021). Una mejora de estos modelos son los **Gated Recurrent Unit (GRU)** que incorporan una compuerta recurrente)

Ventajas y limitaciones de las redes neuronales profundas (RNP o DNN)

- Pueden aproximar relaciones muy complejas entre la entrada y la salida y obtienen muy buenos resultados de predicción y clasificación en problemas muy distintos. Funcionan en general mejor que los métodos estadísticos clásicos con datos desagregados y no lineales.
- Representan la relación entre las variables con muchos parámetros en distintas capas superpuestas, y es muy difícil ver la relación entre las variables de entrada y las de salida. Interpretabilidad
- Las predicciones no incorporan medidas de su precisión, o incertidumbre asociada, como lo hacen los métodos estadísticos. Incertidumbre resultados.
- Como las RN siempre suponen relaciones no lineales cuando la relación entre las variables de entrada y la salida es lineal se obtiene un ajuste muy complicado y menos eficiente y preciso que utilizar directamente un modelo lineal. También con datos linealizables, como datos atípicos o clusters de relaciones, la RN obtiene un modelo complicado y menos eficiente que un modelo estadístico simple para esos datos. Complejidad no necesaria
- Entrenar RN complejas require muchos datos y consumo de energía y tiempo de cálculo. Sostenibilidad

5. Inteligencia Artificial

Tiene como objetivo crear sistemas que:

Actúen como humanos

- Robots que se desplazan y actúan

Tengan capacidades humanas

- Entender el lenguaje y hablar
- Ver y comprender las imágenes
- Responder por escrito a preguntas

Razonen como seres humanos capaces de

- Predecir
- Tomar decisiones
- Clasificar

Generar datos, imágenes, textos, videos, relatos

- IA Generativa

Definición de Inteligencia Artificial

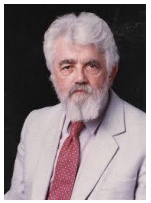
En 2019 la Comisión Mundial de Ética del Conocimiento Científico y la Tecnología (COMEST) de la UNESCO definió la inteligencia artificial como:

Máquinas capaces de imitar determinadas funcionalidades de la inteligencia humana, como la percepción, el aprendizaje, el razonamiento, la resolución de problemas, la interacción lingüística e incluso la producción de trabajos creativos.

Reglamento IA de 2024 de la UE: «sistema de IA»: un sistema basado en una máquina que está diseñado para funcionar con distintos niveles de autonomía y que puede mostrar capacidad de adaptación tras el despliegue, y que, para objetivos explícitos o implícitos, infiere de la información de entrada que recibe la manera de generar resultados de salida, como predicciones, contenidos, recomendaciones o decisiones, que pueden influir en entornos físicos o virtuales.

Breve historia de la IA (siglo XX)

Reunión de IA en Darmouth 1956



John McCarthy (1927-2011)

Perceptrón 1957

F. Rosenblatt (1928-71)

Chatbot ELIZA 1966

J Weizenbaum (1923-2008))

Robótica 1976

Robot Sejourne en Marte

Empuje desde Estadística y ML
RN Profundas 1986

Geoffrey Hinton (1947-)

Estimación de RN Profundas

Reconocimiento de voz 1990

C. Manning (1965-)

- Asistentes Virtuales (SIRI o Alexa)
- Contestadores
- Traductores

Deeper Blue (IBM) derrota a Kasparov Campeón del mundo en ajedrez 1997

Breve historia de la IA (siglo XXI)

Visión artificial

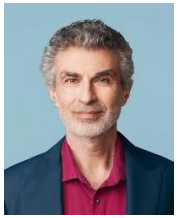
2002

Yann LeCun
(1960 -)

Redes Convolucionales para imágenes

Lenguaje natural

2009



Yoshua Bengio
(1964 -)
Imágenes y lenguaje natural

Alexnet

2012

CNN +Deep Learning

Google Brain Neural Machine Translation

2016

Probabilidad condicionada y correlación Transformers en RNA

2017

AlphaZero
Vencedor del juego Go

Open AI lanza ChatGTP

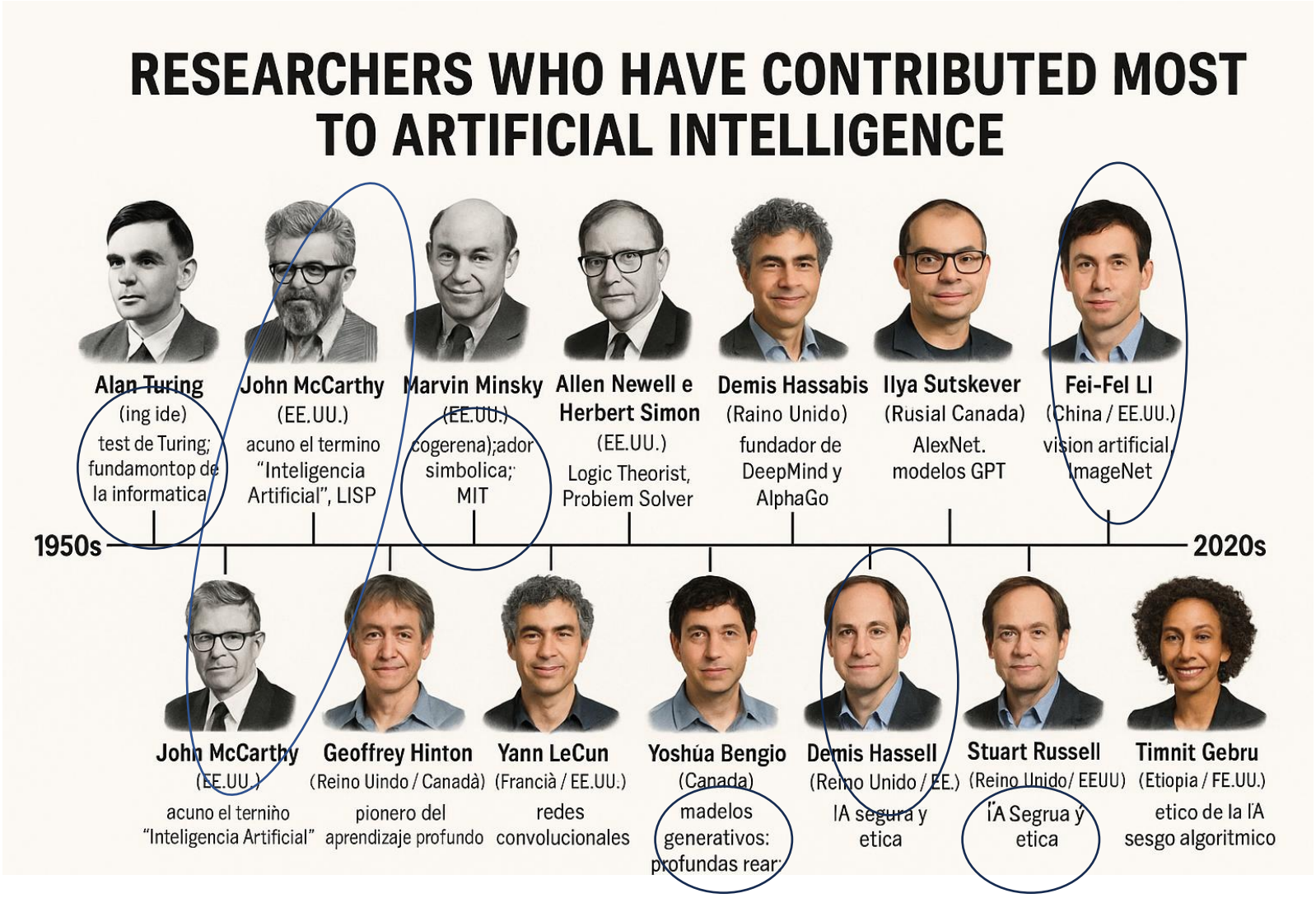
2022

Deep Seek lanza su Chat de IA

2025

deepseek

Imagen creada por ChatGTP



IA Generativa : Procesado de Lenguaje Natural (I.Word Embedding)

Los grandes modelos de lenguaje (ChatGTP, Gemini, DeepSeek,..) generan respuestas a preguntas.

No son buscadores de información sino generadores de respuestas que pueden no ser ciertas.

Requieren un entrenamiento previo para convertir cada palabra de un idioma en un gran vector de números (**Word embedding)**.

En los textos de entrenamiento toman ventanas de tamaño T, normalmente 5 o 6 palabras, y cuentan las veces que las palabras aparecen con otras.

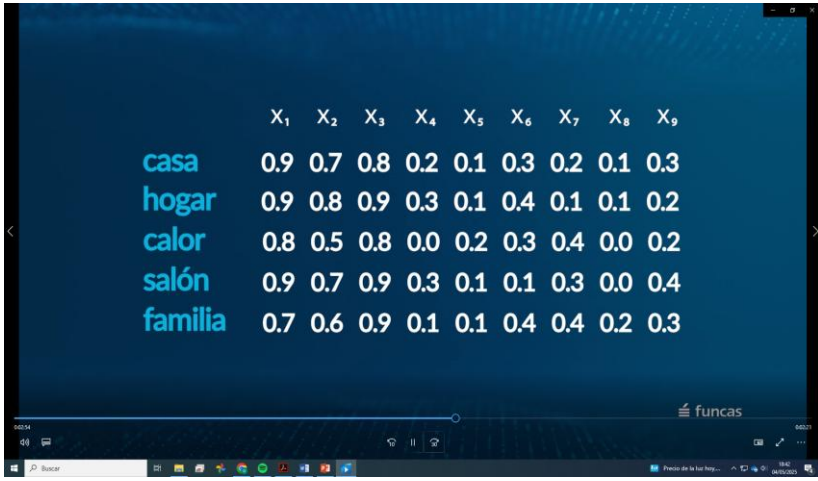
Con las **NP** palabras del idioma se construye una matriz NPxNP cuyas casillas son las frecuencia relativas con las que cada palabra aparece **cerca** del resto de palabras del idioma (donde **cerca** es en la ventana definida y NP es muy grande).

El español tiene un núcleo (token) de unas 50.000 palabras, sin incluir tiempos verbales, derivados etc.

Esta matriz de probabilidades se aproxima por otra **matriz reducida** de dimensiones NPxD donde D<<NP que resume la información obtenida en factores del lenguaje.

Cada fila de esta **matriz reducida** se utiliza para representar cada palabra con un vector de dimensión r (Word embeding) que se interpretan como factores. En ChatGTP unas 12.000 dimensiones o factores.

Vectores próximos en el espacio corresponden a palabras relacionadas entre sí, ya que suelen aparecer próximas.



IA Generativa : Procesado de Lenguaje Natura (II.Definir el contexto con Transformers)

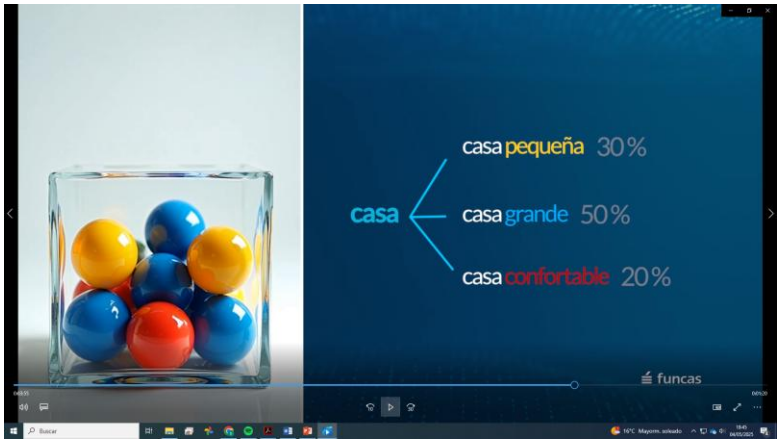
Dada una pregunta las palabras que contiene definen el contexto (o tema) de la pregunta.

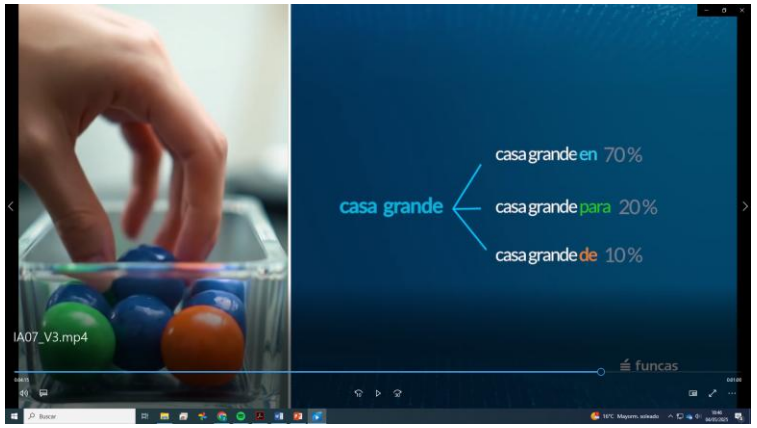
Los vectores que representan las palabras en general hay que adaptarlos **al contexto** definido por la pregunta. Esto se hace con una red neuronal tipo **Transformer.**

El contexto aclara el significado de las palabras en la frase (banco para sentarse, para guardar dinero de un río, de peces, de organos) y modifica su representación. Para ello analizamos la información que puede recibir cada palabra de todas las demás y la que aporta cada palabra a todas las demás. De esta forma modificamos el vector de cada palabra para adaptarlo al contexto.

A continuación para responder a la pregunta se toma la última palabra de la secuencia y se genera la siguiente utilizando las probabilidades con las que todas las palabras pueden aparecen detrás de ella en el contexto de la pregunta. En el sorteo se eliminan las muy improbables para evitar incoherencias. Con este proceso se siguen generando las palabras siguientes, una tras otra sucesivamente.

No hay garantía de que el texto generado de datos ciertos y sea veraz, pero si coherente con el uso del idioma.





Embedding para la predicción de palabras o tokens

Partimos de unos vectores iniciales que representan las palabras de dimensión NPx1 y cuyas coordenadas son la frecuencia relativa con las que las palabras aparecen a continuación de cada una $\boldsymbol{o}_i, i = 1, \dots, NP$. Estos vectores se convierte en otros de dimensión menor Dx1,$\boldsymbol{x_i}, i = 1, \dots, NP$ cuyas coordenadas representan factores o dimensiones del lenguaje, como género (másculino, neutro, femenino) , color (verde, rojo,..), contexto (hogar, geografía, derecho..), valoración(positivo, indiferente, negativo..) etc. Estos factores se obtienen aprendiendo con muchos textos del idioma que factores son útiles para prever la palabra siguiente de una secuencia de palabras. La operación de pasar de los vectores originales de frecuencias, $\boldsymbol{o}_i$, a los vectores de factores ,$\boldsymbol{x_i}$, es

$$\boldsymbol{x_i} = \boldsymbol{E}\, \boldsymbol{o}_i$$

Donde $\boldsymbol{E}$ es una matriz DxNP. Las palabras $\boldsymbol{x_i}$ pueden sumarse y restarse y dar lugar, aproximadamente, a otras palabras, por ejemplo:

Presidenta-mujer = presidente

Las coordenada de los vectores se estandarizan a media cero y varianza unidad. Entonces, el producto escalar de dos vectores, o variables, es su correlación. Las palabras relacionadas porque aparecen juntas tendrán alta correlación.

Dada la representación de una palabra por factores podemos llevarla a frecuencias con la matriz inversa generalizada, $\boldsymbol{E}^-$

$$\boldsymbol{o}_i = \boldsymbol{E}^-\boldsymbol{x_i}$$

Mecanismo de adaptación de cada palabra al contexto con Transformer: self-attention (I)

Las palabras o tokens que utiliza el mecanismo de self –attention son el conjunto de N vectores $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ de dimensión, D, que forman la secuencia que se quiere ampliar.

Primero, estos vectores, $\boldsymbol{x}_m$, m=1,…,N, se transforman en otro conjunto de vectores llamados values de la misma dimensión y estandarizados:

$$\boldsymbol{v}_m = \boldsymbol{b}_v + \boldsymbol{C}_v\, \boldsymbol{x}_m$$

Y con estos N values se generan otros N vectores de salida, $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ adaptados al contexto dado por la secuencia, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, y de la misma dimensión , D. Cada nuevo vector $\boldsymbol{y}_m$ se obtiene ponderando los N valores de los values con un peso dado para cada uno y que suman uno.

$$\boldsymbol{y}_m = \sum_{i=1}^{i=N} w(i,m)\, \boldsymbol{v}_m$$

Los pesos $w(i,m)$ miden cuánto afecta a la palabra m el resto de las palabras de la secuencia, i=1,..,N, donde $\sum_{i=1}^{i=N} w(i,m)$ =1. Son el mecanismo de self-attention y tienen en cuanta la relación entre cada palabra y todas las demás de la secuencia, como se explica a continuación.

Mecanismo de dependencia entre palabras del Transformer: self-attention (II)

Los pesos $w(i,m)$ se determinan definiendo para cada palabra (o token) dos conjuntos de variables:

Queries: (preguntas) cómo influye una palabra en todas las otras en una determinda dimensión definida. Se calculan como vectores $\boldsymbol{q_m}$, para m=1,…N, con dimensión d menor que la entrada, D>d, con

$$\boldsymbol{q_m} = \boldsymbol{b}_q + \boldsymbol{C}_q\, \boldsymbol{x}_m$$

Keys : (claves) cómo es influida una palabra por las demás en una dimensión d, y se calcula como:

$$\boldsymbol{k_m} = \boldsymbol{b_k} + \boldsymbol{C_k}\, \boldsymbol{x}_m$$

Dados estos dos vectores para cada palabra podemos calcular la importancia de la secuencia (todos ellos) sobre cada una de las palabras por unos pesos. El peso de cada palabra, i=1,…N, en una cierta dimensión para la palabra m, se calcula con

$$w(i,m) = \frac{e^{(\boldsymbol{q_{i'}}\boldsymbol{k_m})}}{\sum_{i=1}^{i=N} e^{(\boldsymbol{q_{i'}}\boldsymbol{k_m})}}$$

Se pueden definir muchas combinaciones de Queries and Keys de distinta dimensión que se aplican sucesivamente. Cada token compara su Q con los K de todos los demás tokens anteriores . Cuanto más parecidos, mayor será el peso que ese token da a la información, en el value (V) de los otros. Este proceso se repite en paralelo en varias «cabezas» de atención (multi-head attention).

En GPT, si el embedding es D y tenemos H cabezas, cada una maneja vectores D/H.

Predicción de la próxima palabra

Una vez obtenidos los vectores transformados de la secuencia , $\boldsymbol{y_i}$, i $= 1, \ldots,$ N

el vector de la última palabra se proyecta sobre el espacio de todas las palabras con la matriz del embedding $\boldsymbol{E_F}$ para palabras futuras

$$\boldsymbol{z_{N+1} = E_F y_N}$$

Dando lugar a un vector NPx1 sobre todas las palabras posibles futura cuyas coordenadas, $z_1, \ldots, z_{NP}$, se llaman logit.

Los logit se convierten en una distribución de probabilidad sobre estas posibles palabras futuras con:

$$p_i = e^{z_i} / \sum e^{z_i}$$

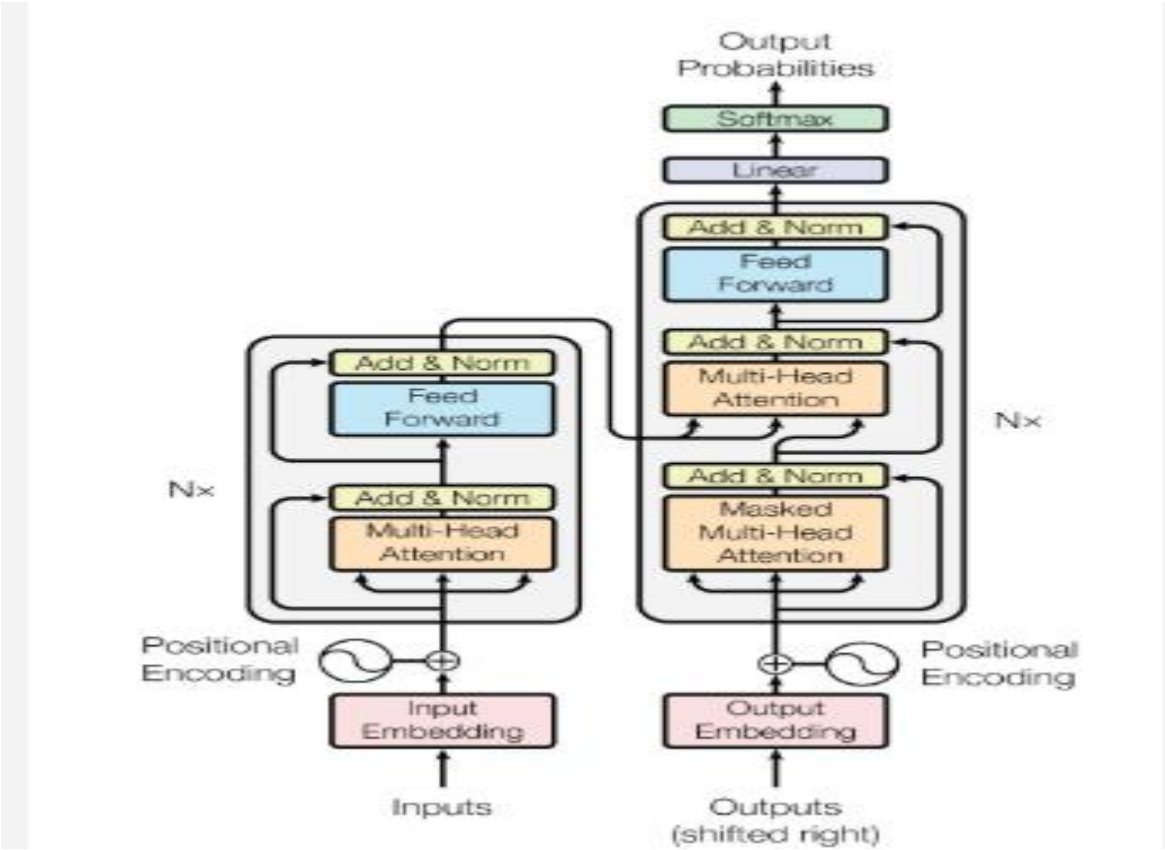
y estas p_i definen la distribución de probabilidad sobre las palabras futuras de la última observada en la secuencia.

Se suele ahora seleccionar la palabra con máxima probabilidad (absoluta o relativa). Esta palabra se añade a la secuencia anterior y la nueva secuencia con una palabra más es el input que se utiliza para prever la palabra siguiente.

El proceso continua hasta que se genera al azar un fin de secuencia o se alcanza una longitud predeterminada.

Red para prever palabras tipo Transformers

La red tiene cuatro partes importantes: el embedding de las palabras (tokens), el mecanismo de atención (Multi-Head attention), redes Feed Forward para la predicción y el output obtenido son las probabilidades de las siguientes palabras (tokens)



IA Generativa : Generación de imágenes

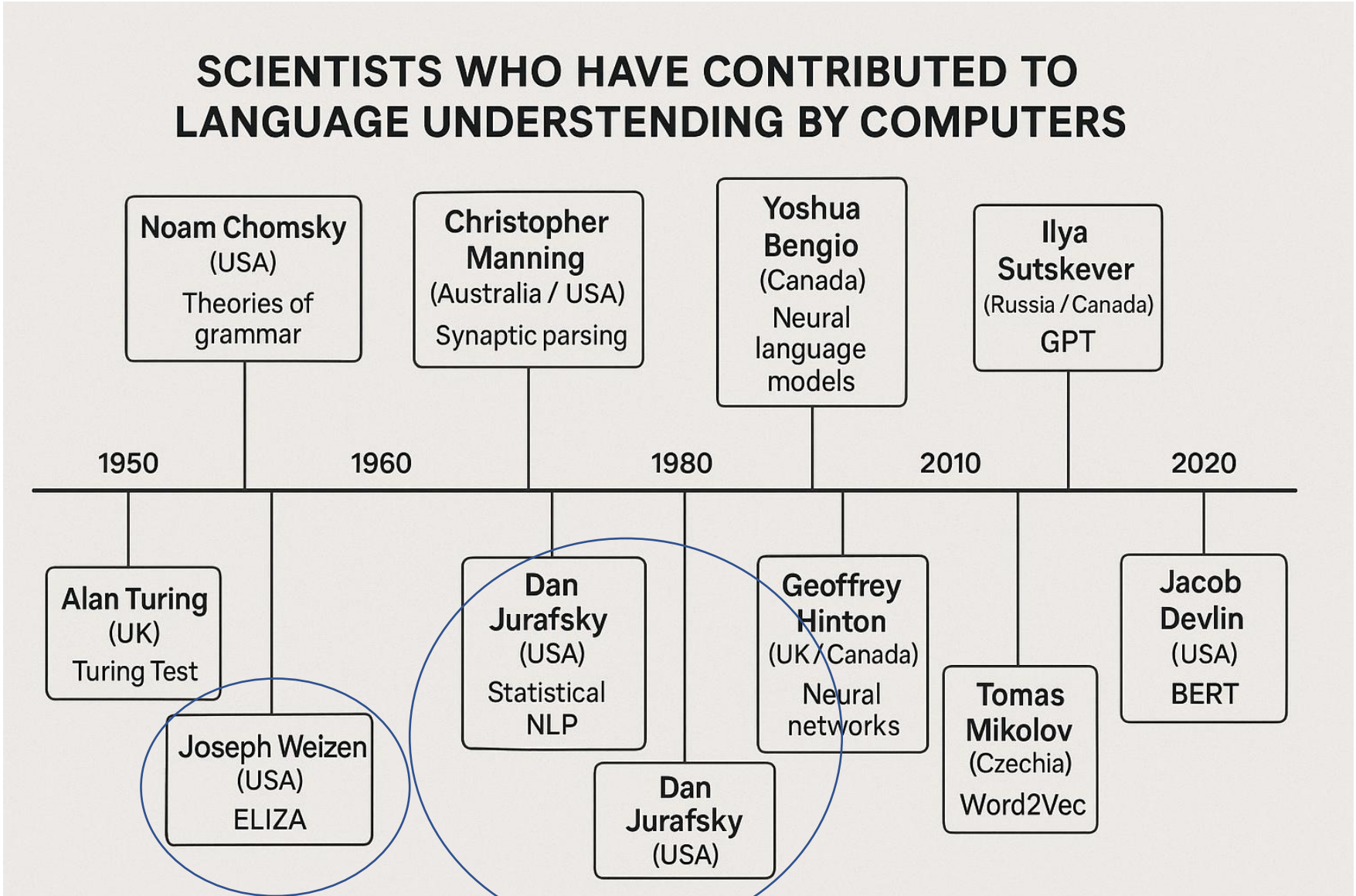
Para generar imágenes se utilizan los elementos que deben aparece en la imagen de acuerdo con la solicitud hecha.

El sistema esta entrenado con muchas imágenes y “genera” un elemento de acuerdo con los patrones que tiene establecidos. No selecciona al azar ninguno de los que tiene sino que hace una especie de “promedio” de los que tiene que corresponde a la pregunta hecha.

Los elementos que acompañan la imagen solicitada y no especificados se seleccionan entre los más probables en las imágenes guardadas.

Se trata de que la imagen no sea ninguna de las existentes pero “parecida” a las más frecuentes de las existentes.

Imagen creada por ChatGTP



6. Los métodos híbridos de IA

Los métodos estadísticos tienen limitaciones para modelar bien relaciones no lineales con muchas variables, problema que resuelven bien las redes neuronales. ¿Podríamos aprovechar lo mejor de cada enfoque?

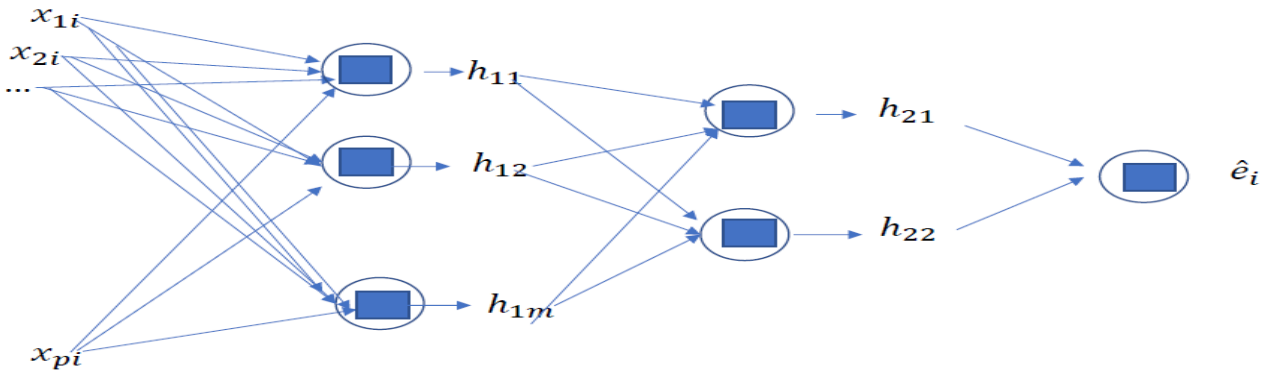
1. Por ejemplo: tomar los residuos de un modelo estadístico y predecirlos mediante una red neuronal feed-forward o recurrente. Esto permite además ver la contribución de la parte lineal y no lineal a la predicción.

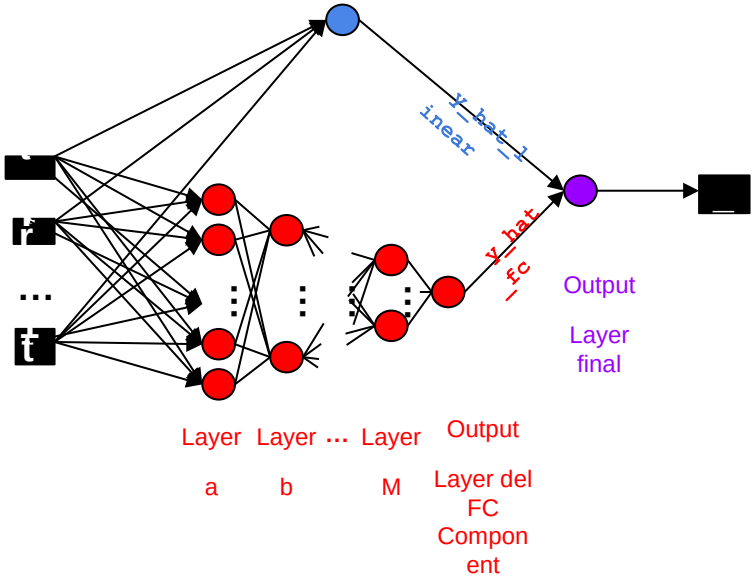
2. Por ejemplo: explorar relaciones más complejas entre las variables con Transformers. Los grandes modelos de lenguaje como ChatGTP predicen palabras en una secuencia buscando relaciones complejas entre las palabras en un idioma. ¿Podemos explorar con estas ideas relaciones de dependencia no lineales entre conjuntos de observaciones? . Quizás podemos modelar con series temporales la parte lineal y explorar los residuos con redes tipo Transformers, como se utilizan en LLM.

6.1 Comparar dos RNA (trabajo con Andrés Alonso y Matías Avila)

First, fit by regression $e_i = y_i - \boldsymbol{x}_i'\beta$ $\boldsymbol{x}_i'=(x_{1i},\ldots, x_{pi})$

Second, explain the residuals e_i by a non linear model





Final output: **Es la combinación lineal de los componentes Linear + FC ponderados tal que:**

y_hat_final =
weight_for_linear_component*y_hat_linear + weight_for_non_linear_component*y_hat_fc + constant

Estructura de la red con dos capas:

- El output final de la red neuronal es la suma de la salida del componente lineal y el componente no lineal . Esta predicción final se genera en la output layer final (la última neurona de toda nuestra red neuronal). Fijando los pesos de la combinación a uno.
- Para interpretar el resultado el único sesgo o constante que se añade es la del componente lineal. El sesgo del componente no lineal se fija en ceros.

Dos ejemplos de simulación

Se consideran tres conjuntos de datos:

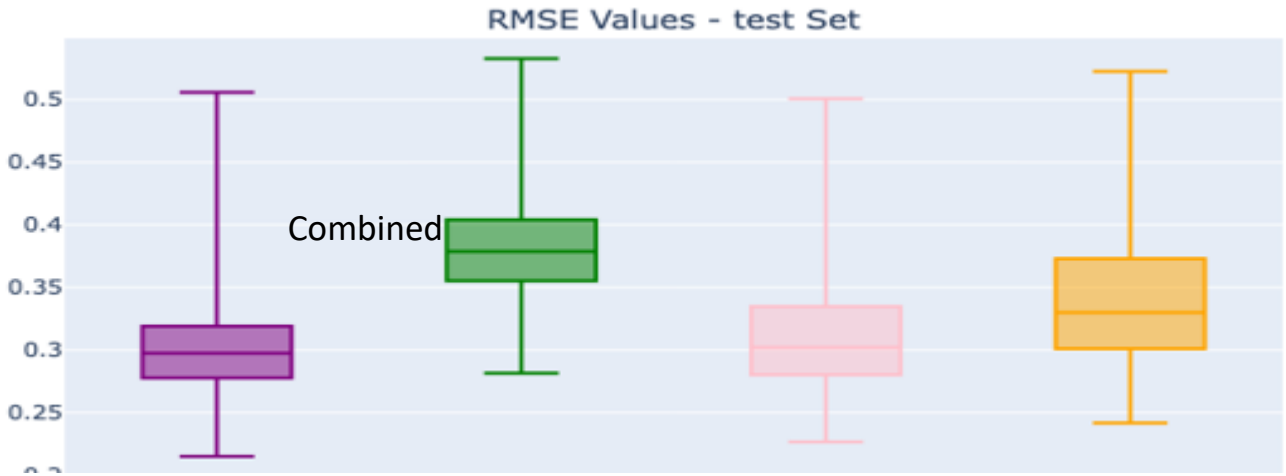
- **Linear**: la variable respuesta se genera como una combinación lineal de 10 variables de entrada, con pesos aleatorios y distribución normal más un término de ruido normal.
- **Non-linear**: La variable respuesta se genera como una combinación lineal de 10 variables que son transformaciones no lineales (como funciones seno, logaritmo, raíz cuadrada, exponencial, etc.), de las variables en el caso lineal más un ruido normal. En la primera simulación la no linealidad resulta de transformar las variables iniciales y en la segunda la transformación se realiza para asegurar una clara no linealidad en el rango de valores generados.
- **Combined**: La variable objetivo se genera combinando linealmente las variables respuesta anteriores **Linear**, y **Non-linear** más un término de ruido normal.

- Las variables explicativas son continuas, sin ninguna categórica

Tabla para RMSE:

Resultados primera simulación

	MSE en función del conjunto de datos y el modelo de predicción utilizado			
Dataset	RedN Híbrida	Regresión lineal	Regresión y RN en residuos	RN en datos
Combined	**0.3043**	0.3810	0.3116	0.3403
Linear	0.1854	**0.1809**	0.1825	0.1886
non_linear	**0.5227**	0.6649	0.5415	0.5494



Primera simulación: El coeficiente de no linealidad en este ejemplo para los datos Combinados es del 20%, ((.381-.304)/.381) y para los no lineales puros del 21% ((.665-.523)/.665).

Medir la no linealidad (NL) en la predicción de los datos

Comparar el error de una predicción lineal con el error de la mejor predicción posible, probando tanto el modelo lineal como muchos modelos no lineales.

Medida del error: error cuadrático medios (ECM)

ECM(lineal)= con una predicción lineal
ECM(óptimo)= mejor predicción lineal o no lineal encontrada

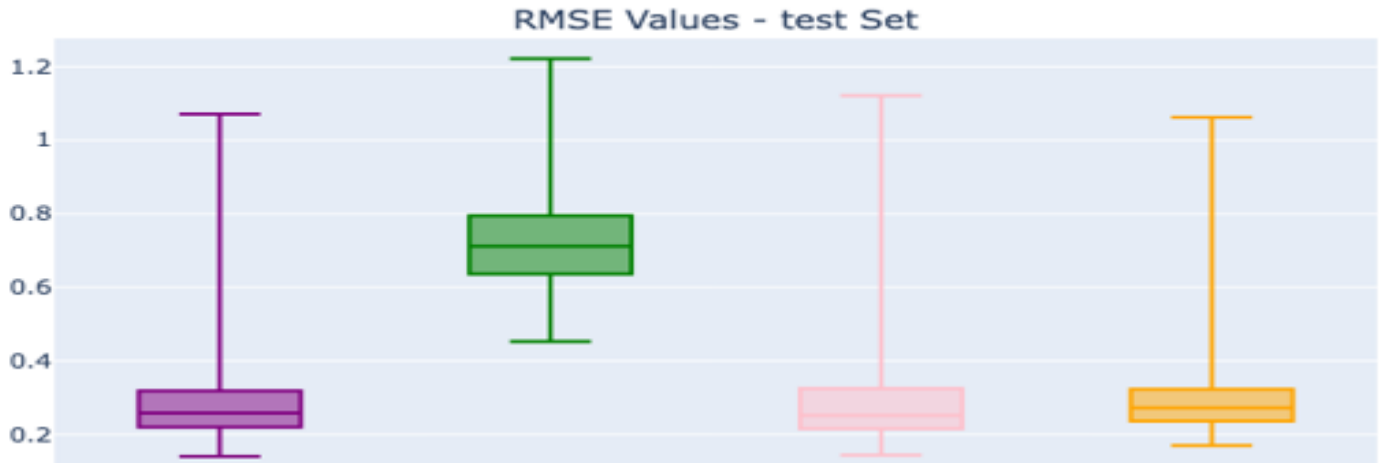
$$\text{CNL}=Coeficiente\ NL = \frac{ECM(lineal)-ECM(\acute{o}ptimo)}{ECM(lineal)}$$

CNL=0 . EL óptimo es el modelo lineal
CNL=1 Un modelo no lineal tiene error cero y menor que el lineal

Tabla para RMSE:

Resultados segunda simulación

	MSE en función del conjunto de datos y el modelo de predicción utilizado			
Dataset	RedN Híbrida	Regresión lineal	Regr +RN residuos	RN s datos
Combined	**0.3025**	0.7226	0.3198	**0.2996**
Linear	0.1292	**0.1256**	0.1268	0.1347
non_linear	**0.2320**	0.5794	0.2608	0.2429



Segunda simulación: El coeficiente de no linealidad para los datos Combinados es del 58%, ((.7226-.2996)/ .7226), y para los no lineales puros del 60%, ((.5794-.2320)/ .5794).

Resultados MES en datos reales

Se utiliza el 80% de los datos para muestra de estimación, 10% para validación y 10% para muestra de predicción

Los datos se dividen al azar en estas tres muestras y se realizan 1000 replicaciones. Se dan los valores medios de MSE.

Datos	N	p	RNA Híbrida	Regresión lineal	Regr +RNA residuos	RNA datos	CNL
Diabetes Efron et al (2004)	442	11	0.7120	0.7119	**0.7114**	0.7185	.001
California Housing Statlib Rep.	20640	8	**0.5297**	0.6289	0.5408	0.5309	.158
Boston Housing Belsley et al 1978	506	13	**0.2934**	0.3710	0.3104	0.3019	.194

- Las mismas ideas pueden aplicarse para redes con dinámica para series temporales. Como recurrentes tipo LSTM (Long short term memory) y este trabajo está en realización con Andrés Alonso y Matías Avila.

6.2 Explorando dependencias complejas en series temporales (con Nicolas Buhringer)

Otra forma de captar relaciones de dependencia complejas en datos de series temporales muy desagregados es con ayuda de los métodos utilizados en los grandes modelos de lenguaje, como ChatGTP.

Estos modelos expresan la dependencia entre las palabras para prever la siguiente palabra en una secuencia convirtiendo cada palabra en un gran vector y estudiando la correlación entre estos vectores.

Dadas las palabras

$$\boldsymbol{w}_1, \ldots, \boldsymbol{w}_t \text{ prever } \boldsymbol{w}_{t+1}$$

Teniendo en cuenta que la dependencia en secuencias de palabras entre ellas es mucho más compleja que la que esperamos entre observaciones en una serie temporal.

Predicción de Realized volatility (Tesis de Master de Nicolas Buhringer

Pregunta: ¿Hay una relación compleja entre los valores desagregados pasados de una serie temporal y el valor agregado futuro?

The dataset consists of high-frequency stock price data from 28 stocks of the Dow Jones Industrial Average index. The observation period reaches from January 4th, 2010 to December 31st, 2019 with minutely sampled values of the stock price from 9:35:00 am to 15:55:00 pm. Thus, the dataset of each stock consists of 2516 trading days each containing 381 stock price observations resulting in 958,596 rows. To calculate the Log Returns, the closing price of each minute was used.

The dataset for each stock was split into a training set a validation set and a test, where forecast are generated. A 85%-15% split was used resulting in 2,138 observed days in the training set and validation set, and 378 observations in the test set.

Application: time series prediction

The objective is to forecast daily Realized Volatility, defined as the square root of the sum of squared intraday log returns

$$\mathrm{RV_t} = \sqrt{\sum_{i=1}^{M} r_{t,i}^2}$$

Where M as the number of intraday price observations, $P_{t,i}$ is the price at day t and interval I and $r_{t,i} = \log(P_{t,i}) - \log(P_{t,i-1})$.

The input for forecasting the daily volatility are consecutive minutes patches of intraday log returns , $r_{t,i}$, during one day, that is only the 380 log returns of the previous day are used as input sequence.

The model first projects each scalar input to D=64 linear embedding and adds learned positional embeddings. After the final encoder layer, the hidden state of the CLS token is passed through a prediction head to produce the next-day log realized volatility residual.

The results in RMSE (root mean squared forecast error) of the Transformer model are compared to those of a Naïve (Random Walk), N, Forecast and those of a HAR model (Corsi, 2009):

$$N_t = RV_{t-1}$$

And the HAR model is built by the three features for daily, weekly and monthly realized volatility as:

$$RV_{t-1}^{(d)} = RV_{t-1}, \quad RV_{t-1}^{(w)} = \frac{1}{5}\sum_{i=1}^{5} RV_{t-i}, \quad RV_{t-1}^{(m)} = \frac{1}{22}\sum_{i=1}^{22} RV_{t-i}$$

The HAR forecast is then generated as:

$$\widehat{RV_{HAR,t}} = \widehat{\beta_0} + \widehat{\beta_d} RV_{t-1}^{(d)} + \widehat{\beta_w} RV_{t-1}^{(w)} + \widehat{\beta_m} RV_{t-1}^{(m)}$$

Figure 6: RMSE out-of-sample results

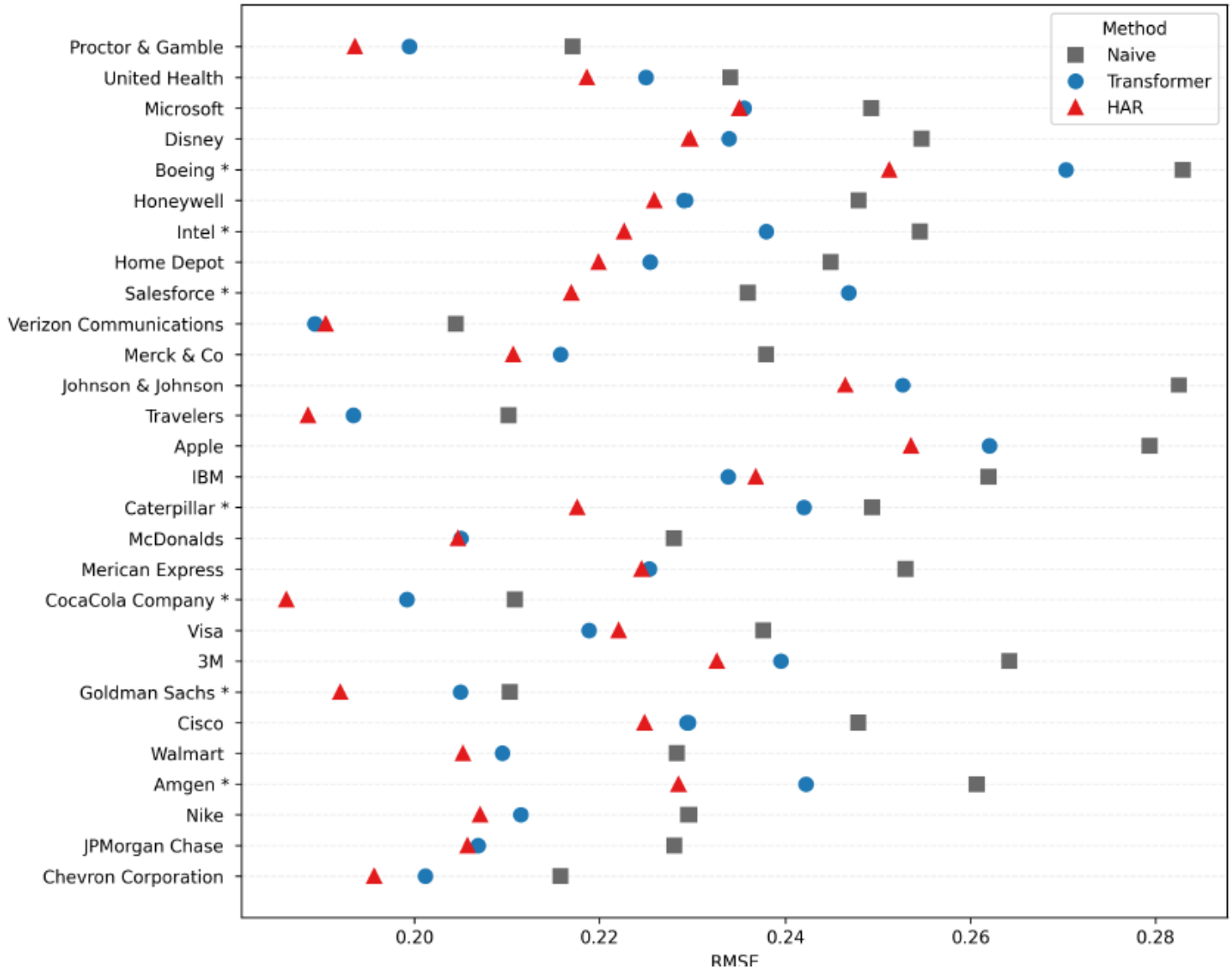


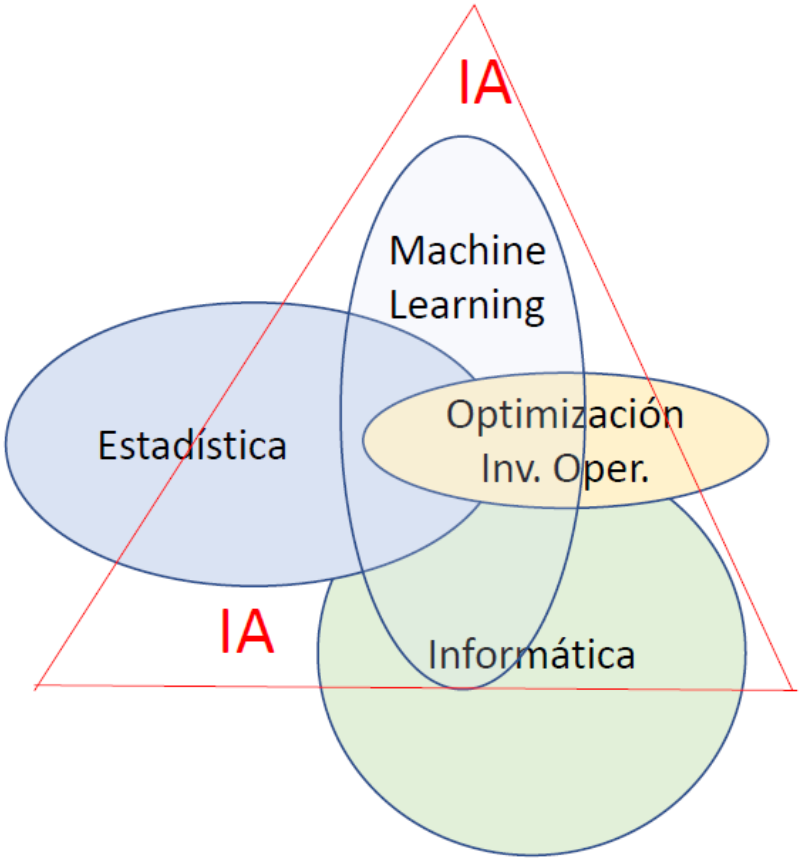
Table 2: Diebold-Mariano Test Results

	Naive-Transformer		HAR-Transformer	
Stock	Test Statistic	p-value	Test Statistic	p-value
Proctor & Gamble	4.268	.00003*	-1.290	.198
United Health	1.431	.153	-.947	.344
Microsoft	2.693	.007*	-.095	.924
Disney	3.886	.0001*	-1.029	.304
Boeing	1.532	.126	-2.859	.004*
Honeywell	3.947	.0001*	-.778	.437
Intel	2.432	.015*	-2.850	.005*
Home Depot	3.614	.0003*	-1.411	.159
Salesforce	-1.462	.144	-4.335	.00002*
Verizon Communications	3.037	.003*	.329	.742
Merck & Co	4.372	.00002*	-1.156	.248
Johnson & Johnson	4.518	.00001*	-1.358	.175
Travelers	3.547	.0004*	-1.183	.238
Apple	2.769	.006*	-1.960	.051
IBM	3.659	.0003*	.525	.600
Caterpillar	1.194	.233	-4.110	.00005*
McDonalds	4.445	.00001*	-.084	.933
American Express	4.154	.00004*	-.180	.857
CocaCola Company	2.879	.004*	-3.171	.002*
Visa	3.716	.0002*	.650	.516
3M	3.310	.001*	-1.511	.132
Goldman Sachs	1.132	.258	-2.767	.006*
Cisco	2.917	.004*	-1.010	.313
Walmart	3.800	.0002*	-.898	.370
Amgen	2.725	.007*	-2.456	.014*
Nike	3.076	.002*	-1.012	.312
JPMorgan Chase	4.807	0.00000*	-.288	.773
Chevron Corporation	3.380	.001*	-1.421	.156

Note: p-values smaller than 0.05 are marked with an asterisk

Conclusiones

Los métodos de Inteligencia Artificial con datos masivos están integrando los conceptos estadísticos y los métodos de Machine Learning y de optimización e IO en un marco de computación intensiva para obtener sistemas de predicción y decisión de carácter general.



Conclusiones

- El análisis estadístico durante el siglo XX ha impulsado los avances científicos, en todos los campos, combatiendo los prejuicios e incrementado la salud, la riqueza y la igualdad en el mundo.
- Sin embargo, en este siglo los estadísticos hemos dedicado pocos esfuerzos al análisis de datos masivos digitales, que han impulsado los métodos de aprendizaje automático (ML) y el desarrollo de la Inteligencia Artificial IA.
- La estadística es imprescindible para medir los sesgos en el entrenamiento, medir la incertidumbre en los resultados y facilitar la su interpretación. Puede también contribuir a la sostenibilidad de los modelos controlando la complejidad necesaria para el análisis.
- El análisis estadístico podría explicar en que situaciones van a tener éxito los modelos de aprendizaje de IA y en cuáles no y cómo simplificar el proceso para hacerlo más rápido y eficiente.
- Debemos desarrollar métodos híbridos combinando los métodos estadísticos y los de aprendizaje automático, y orientados a aprovechar las ventajas de cada enfoque.

Referencias

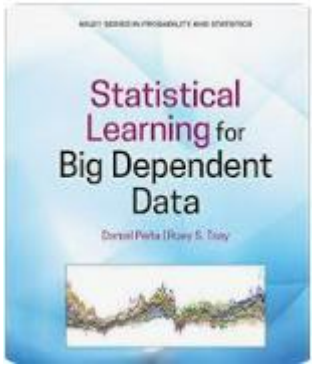
- Peña, D. and Tsay, R.S. (2021). Statistical Learning for Big Dependent Data. Wiley.

- Peña, D. (2025). Investigación Económica con Datos Masivos (Coordinador). Fundación Ramón Areces.

- Peña, D. (2025). Comprender la IA. *Big data y aprendizaje estadístico automático*

Funcas: https://www.funcas.es/articulos/big-data-aprendizaje-estadistico-automatico-e-inteligencia-artificial/

YouTube: https://www.youtube.com/playlist?list=PLQR2nth3aeQRdcUKsIuMzKIbFEeeSgzEC



moltes gràcies per la vostra atenció

i

molt feliç dia de l'estadistica