



Escuela Técnica Superior de  
**Ingeniería Informática**

TRABAJO FIN DE GRADO

## **Título del trabajo**

Realizado por  
**Jaime Rodríguez Albuín**

Para la obtención del título de  
Grado en Ingeniería Informática - Tecnologías Informáticas

Dirigido por  
Isabel De Los Ángeles Nepomuceno Chamorro

Realizado en el departamento de  
Lenguajes y Sistemas Informáticos

**Convocatoria de Junio, curso 2020/21**

---

# Agradecimientos

---

Quiero agradecer sobre todo a mi hermano y a mi madre por estar apoyándome desde el minuto cero incondicionalmente, siempre dispuestos a llevarme y a traerme de Sevilla y siendo sin duda piezas clave en esta etapa de mi vida.

También quiero agradecer a mis tíos Victoria y Manolo por ayudarnos en momentos tan complicados y haber ejercido siempre de familia.

Agradezco a mi tutora Isabel el hecho de aceptarme como alumno incluso llegando un poco tarde. También a todos los profesores que he tenido a lo largo de la carrera, puesto que sé con certeza que la formación que he recibido a lo largo de estos cuatro años me servirá, tanto en los conocimientos como en la forma de afrontar los problemas.

No me voy a olvidar de los maravillosos compañeros que he conocido a lo largo de este viaje, y sobre todo agradezco las amistades con Arturo, Ventura, Emilio, Jesús, Juan, Juan Carlos, Pedro García, Pedro Escobar, Curro, Antonio, Manuel, Javi y Josemi.

---

# Resumen

---

Trabajo de investigación sobre redes neuronales convolucionales y la detección de pacientes con neumonía, posiblemente provocadas por Covid.

**Palabras clave:** Redes neuronales, Convolución, Pandas, Keras, Tensorflow, Covid, Kaggle, Google Colab, Inteligencia Artificial, Deep learning, Neumonía, Salud.

---

# Abstract

---

Research project about convolutional neural networks and pneumonia detection on patients, possibly provoked by Covid

**Keywords:** Neural networks, Convolution, Pandas, Keras, Tensorflow, Covid, Kaggle, Google Colab, Artificial Intelligence, Deep learning, Pneumonia, Health.

---

# Índice general

---

<b>1. Ejemplos de uso LaTeX</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Estilos . . . . .	1
1.3. Listados . . . . .	1
1.4. Subsecciones . . . . .	1
1.4.1. Primera subsección . . . . .	2
1.4.2. Segunda subsección . . . . .	2
1.5. Imágenes y figuras . . . . .	2
1.6. Referencias . . . . .	3
1.7. Extractos de código . . . . .	3
1.8. Enlaces . . . . .	4
1.9. Citas y bibliografía . . . . .	4
1.10. Ecuaciones . . . . .	4
1.11. Caracteres y símbolos especiales . . . . .	4
<b>2. Introducción</b>	<b>6</b>
<b>3. Estudio Previo</b>	<b>7</b>
3.1. Introducción . . . . .	7
3.2. Aprendizaje supervisado y no supervisado . . . . .	7
3.2.1. Aprendizaje supervisado . . . . .	7
3.2.2. Aprendizaje no supervisado . . . . .	9
3.3. Red neuronal convolucional . . . . .	11
3.3.1. Inspiración biológica . . . . .	11
3.3.2. El perceptrón . . . . .	13
3.4. Planificación . . . . .	17
3.5. Presupuesto . . . . .	17
3.6. Conclusiones . . . . .	17
<b>4. Análisis del problema</b>	<b>18</b>
4.1. Introducción . . . . .	18
4.2. Requisitos de información . . . . .	18
4.3. Requisitos funcionales . . . . .	18
4.4. Requisitos no funcionales . . . . .	18
4.5. Conclusiones . . . . .	18
<b>5. Diseño del problema</b>	<b>19</b>
5.1. Introducción . . . . .	19
5.2. Conclusiones . . . . .	19
<b>6. Implementación</b>	<b>20</b>
6.1. Introducción . . . . .	20
6.2. Herramientas . . . . .	20

6.3. Conclusiones . . . . .	20
<b>7. Pruebas</b>	<b>21</b>
7.1. Introducción . . . . .	21
7.2. Conclusiones . . . . .	21
<b>8. Conclusiones</b>	<b>22</b>
<b>9. Bibliografía</b>	<b>23</b>

---

# Índice de figuras

---

1.1. Un feroz depredador . . . . .	2
3.1. Resultado de aplicar la detección de objetos . . . . .	7
3.2. Ejemplo de segmentación de imágenes . . . . .	8
3.3. Imagen que demuestra el flujo de la generación de subtítulos . . . . .	8
3.4. Resultado de la aplicación de clustering sobre un conjunto de individuos . . . . .	9
3.5. Etapas en la reducción de dimensiones. . . . .	10
3.6. Estructura de una neurona biológica. . . . .	11
3.7. Diagrama que muestra el disparo de una neurona . . . . .	12
3.8. Comparativa de la estructura de la neurona biológica con el perceptrón . . . . .	13
3.9. Gráfica de la función escalón de Heaviside. . . . .	14
3.10. Gráfica de la función ReLU. . . . .	15
3.11. Gráfica de la función sigmoide. . . . .	15
3.12. Gráfica de la función softmax. . . . .	16

---

# Índice de extractos de código

---

1.1. Código Python . . . . .	3
1.2. Código Java . . . . .	3



---

# 1. Ejemplos de uso LaTeX

---

Este capítulo se incluye únicamente como ayuda y referencia de uso de LaTeX. No debe aparecer en el documento final.

## 1.1. Introducción

En este capítulo se muestran ejemplos de uso de LaTeX para operaciones comunes.

## 1.2. Estilos

Se pueden aplicar estilos al texto como **negritas**, *cursiva* y subrayado. También se **pueden aplicar colores**, y combinar estilos. Se recomienda usar sólo negritas para hacer énfasis, y no abusar de este recurso.

## 1.3. Listados

Con itemize se pueden crear listas no numeradas:

- Fresas
- Melocotones
- Piñas
- Nectarinas

De manera similar, enumerate permite crear listas numeradas:

1. Elaborar la memoria del TFG
2. Elaborar la presentación
3. Presentar el TFG
4. Solicitar el título de Grado

## 1.4. Subsecciones

Se pueden definir subsecciones con el comando subsection:

### 1.4.1. Primera subsección

Esto es una subsección

### 1.4.2. Segunda subsección

Esto es otra subsección.

## 1.5. Imágenes y figuras

Todas las imágenes y figuras del documento se incluirán en la carpeta “fig”. Se pueden incluir de la siguiente manera:



Figura 1.1: Un feroz depredador

Observe que las figuras se numeran automáticamente según el capítulo y el número de figuras que hayan aparecido anteriormente en dicho capítulo. Existen muchas maneras de definir el tamaño de una figura, pero se aconseja utilizar la mostrada en este ejemplo: se define el ancho de la figura como un porcentaje del ancho total de la página, y la altura se escala automáticamente. De esta manera, el ancho máximo de una figura sería  $1.0 * \text{textwidth}$ , lo que aseguraría que se muestra al máximo tamaño posible sin sobrepasar los márgenes del documento.

Tenga en cuenta que LaTeX intenta incluir las figuras en el mismo sitio donde se declaran, pero en ocasiones no es posible por motivos de espacio. En esos casos, LaTeX colocará la figura lo más cerca posible de su declaración, puede que en una página diferente. Esto es un comportamiento normal y no debe ser evitado.

## 1.6. Referencias

Observe cómo en el código fuente de esta sección se ha usado varias veces el comando “label”. Este comando permite marcar un elemento, ya sea capítulo, sección, figura, etc. para hacer una referencia numérica al mismo. Para referenciar una “label” se usa el comando “ref” incluyendo el nombre de la referencia:

Este es el capítulo [1](#).

En la sección [1.2](#) se muestran ejemplos de estilos.

La subsección [1.4.1](#) explica...

En la Figura [1.1](#) vemos que...

Esto evita que tengamos que escribir directamente los índices de las secciones y figuras que queremos mencionar, ya que LaTeX lo hace por nosotros y además se encarga de mantenerlos actualizados en caso de que cambien (pruebe a mover este capítulo al final del documento y observe cómo se actualizan automáticamente todos los índices referenciados). Además, las referencias mediante “ref” actúan como hipervínculos dentro del documento que llevan al elemento referenciado al pulsar en ellas.

Es habitual nombrar las “label” con un prefijo que indica el tipo de elemento para encontrarlo luego más fácilmente, pero no es obligatorio.

## 1.7. Extractos de código

Se pueden incluir extractos de código mediante lstlisting:

```
num = float(input("Enter_a_number:_"))
if num > 0:
    print("Positive_number")
elif num == 0:
    print("Zero")
else:
    print("Negative_number")
```

Extracto de código 1.1: Código Python

Se admite gran variedad de lenguajes:

```
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello,_world!");
    }
}
```

Extracto de código 1.2: Código Java

Los extractos de código también se pueden referenciar mediante label/ref: Extractos de código [1.1](#) y [1.2](#).

## 1.8. Enlaces

Puede enlazar una web externa mediante el comando url: <https://www.example.com>. También se puede vincular un enlace a un texto mediante el comando href: [dominio de ejemplo](#).

## 1.9. Citas y bibliografía

En LaTeX, los elementos de la bibliografía se almacenan en un fichero bibliográfico en un formato llamado BibTeX, en el caso de este proyecto se encuentran en “bibliografia.bib”. Para citar un elemento se usa el comando “cite”. Se pueden citar tanto artículos científicos [1] como enlaces web [2]. Las citas se numeran automáticamente y se incluyen en la sección de bibliografía del documento.

Observe cómo los elementos bibliográficos almacenados en “bibliografia.bib” tienen una etiqueta asociada, que es la que se incluye al citarlos mediante cite. Añadir una referencia al fichero bibliográfico no hace que ésta aparezca automáticamente en la sección de bibliografía del trabajo, es necesario citarla en algún lugar del mismo.

## 1.10. Ecuaciones

LaTeX tiene un potente motor para mostrar ecuaciones matemáticas y un amplio catálogo de símbolos matemáticos. El entorno matemático se puede activar de muchas maneras. Para incluir ecuaciones simples en un texto se pueden rodear de símbolos dólar:  $1 + 2 = 3$ ,  $\sqrt{81} = 3^2 = 9$ ,  $\forall x \in y \exists z : S_z < 4$ .

Las ecuaciones más complejas pueden expresarse aparte y son numeradas: ecuación 1.1.

$$\lim_{x \rightarrow 0} \frac{e^x - 1}{2x} \stackrel{\left[\frac{0}{0}\right]}{=} \lim_{x \rightarrow 0} \frac{e^x}{2} = \frac{1}{2} + 7 \int_0^2 \left( -\frac{1}{4} (e^{-4t_1} + e^{4t_1-8}) \right) dt_1 \quad (1.1)$$

Dispone [aquí](#) de un amplio listado de símbolos que pueden usarse en modo matemático.

## 1.11. Caracteres y símbolos especiales

Algunos caracteres y símbolos deben ser escapados para poder representarse en el documento, ya que tienen un significado especial en LaTeX. Algunos de ellos son:

- El símbolo dólar \$ se usa para ecuaciones.
- El tanto por ciento % se usa para comentarios en el código fuente.

- El símbolo euro € suele dar problemas si se escribe directamente.
- El guión bajo \_ se usa para subíndices en modo matemático.
- Las comillas deben expresarse ‘así’ para comillas simples y “así” para comillas dobles. Las comillas españolas pueden expresarse «así».
- La barra invertida o contrabarra \ se usa para comandos LaTeX.
- Otros símbolos que deben escaparse son las llaves { }, el ampersand &, la almohadilla # y los símbolos mayor que > y menor que <.

---

## 2. Introducción

---

### Motivación

La principal fuente de motivación ha sido la pandemia. Mi principal objetivo fué encontrar una manera de predecir un diagnóstico sobre las personas que pudieran padecer de COVID. Entre síntomas hay uno que siempre ha sido uno de los principales indicios de dicha enfermedad, y es la neumonía provocada por el virus. Esta neumonía es la que provoca la tos en los pacientes, y el hecho de encontrar esta propuesta en la página web de los TFG de la ETSII fué un flechazo.

Además, el hecho de ver las redes neuronales tan por encima en IA y AIA me dejó con las ganas de investigar. No conocía las redes convolucionales, y me han parecido un descubrimiento espectacular.

También el hecho de desarrollar algo que puede ayudar en un campo tan noble como la salud es una motivación enorme, aunque tengo muy claro que hay gente con mucha más experiencia que yo en el sector y no sé si podría a llegar a competir con ellos.

---

## 3. Estudio Previo

---

### 3.1. Introducción

En este capítulo introduciremos las redes convolucionales desde el perceptrón hasta el estado del arte y el dataset que trabajaremos.

### 3.2. Aprendizaje supervisado y no supervisado

Primero deberemos preguntarnos, ¿qué significa el "aprendizaje" en máquinas? La respuesta depende de qué tipo de aprendizaje buscamos:

#### 3.2.1. Aprendizaje supervisado

Este acercamiento al aprendizaje automático busca obtener una función a partir de un conjunto de datos, es decir, relaciones que asocian entradas con salidas. Este acercamiento al aprendizaje automático incluso llega a proporcionar las formas de clasificar más comunes. Dependiendo de la salida diferenciaremos entre modelos de clasificación, si la salida es un valor categórico (como una enumeración, o un conjunto finito de clases), y modelos de regresión, si la salida es un valor en un espacio continuo.

Las aplicaciones son muy numerosas y siguen estando a la orden del día, tales como reconocimiento de voz, clasificación de imágenes (como en este trabajo) o traducción de idiomas. También existen otras aplicaciones más exóticas que merecen la pena mencionar:

- Detección de objetos: dada una imagen dibujar una caja que delimite los objetos a detectar. Este problema puede ser presentado como un problema de clasificación (obteniendo muchos candidatos a los que dibujar la caja, elegir si se dibujan o no) o como una unión entre un problema de clasificación y uno de regresión, donde las esquinas de dicha caja se predigan mediante vectores de regresión.

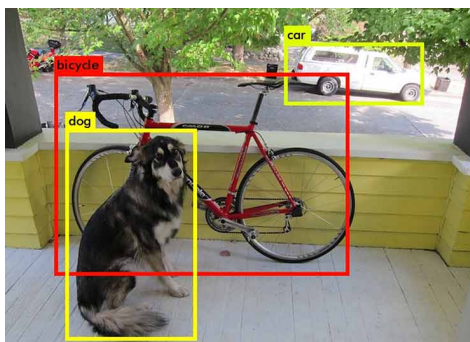


Figura 3.1: Resultado de aplicar la detección de objetos

- Segmentación de imágenes: dada una imagen, dibujar una máscara píxel a píxel sobre un objeto en específico.

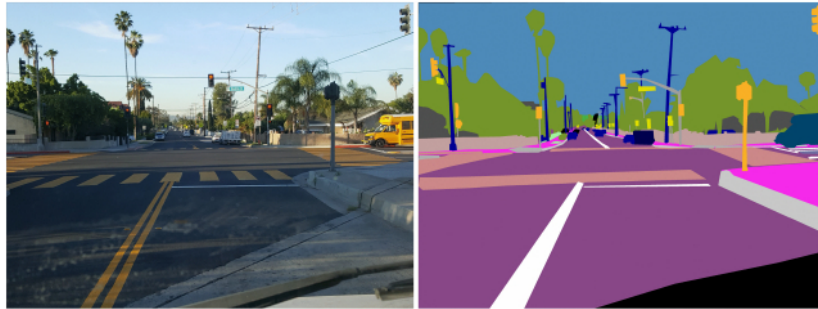


Figura 3.2: Ejemplo de segmentación de imágenes

- Generación de secuencias: dada una imagen, predecir un subtítulo que la describa. Este problema a veces puede ser reformulado como una secuencia de

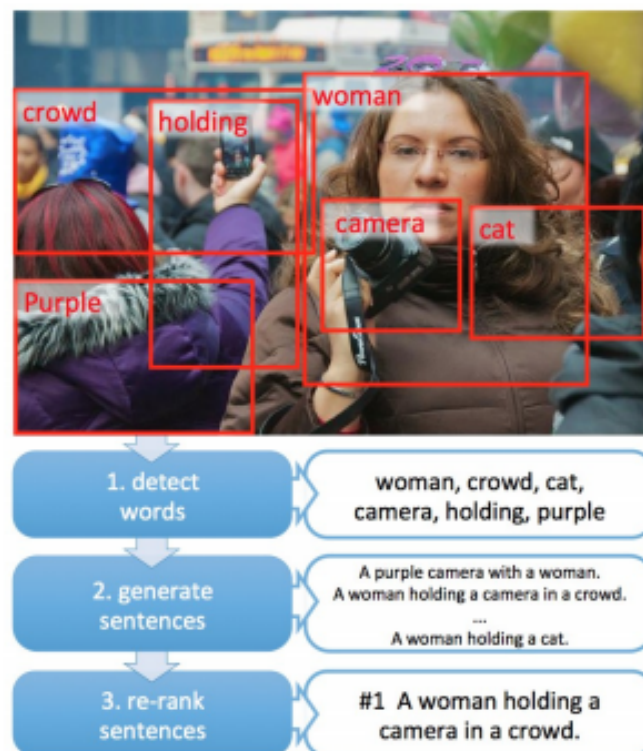


Figura 3.3: Imagen que demuestra el flujo de la generación de subtítulos



### 3.2.2. Aprendizaje no supervisado

Esta rama del aprendizaje automático se utiliza cuando no nos interesa ajustar pares (entrada/salida), sino en aumentar el conocimiento estructural de los datos de entrada. Consiste en encontrar transformaciones interesantes de los datos de entrada sin la ayuda de etiquetas, con el propósito de visualizar los datos, comprimirlos, eliminar ruido o simplemente aumentar el entendimiento de los datos en cuestión.

Su principal aplicación se encuentra en el análisis de datos, a suele ser un paso necesario en la mejora del entendimiento de los datos de un dataset antes de la resolución por un algoritmo de aprendizaje supervisado.

Tipos de algoritmos no supervisados más comunes:

- Clustering: busca agrupar una serie de elementos según su semejanza. Por ello, un *cluster* es una agrupación de elementos similares.

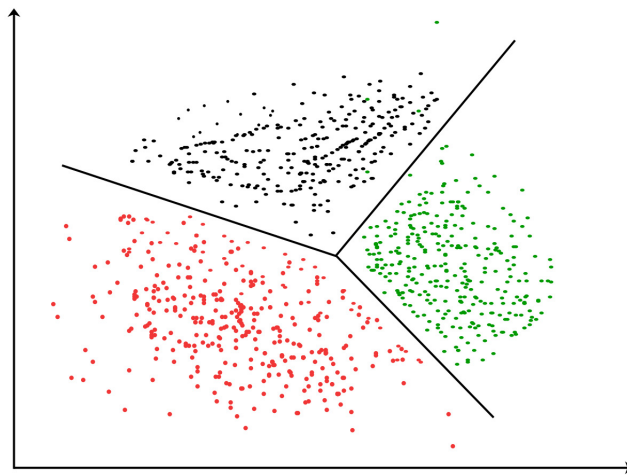


Figura 3.4: Resultado de la aplicación de clustering sobre un conjunto de individuos

- Reducción de dimensiones: es la transformación de datos de un espacio dimensional elevado a otro de menores dimensiones, manteniendo en este nivel menor de dimensionalidad características importantes.

Este tipo de algoritmos es muy común en campos que manejan grandes cantidades de observaciones y/o variables, como el procesamiento de señales, el reconocimiento de voz o la bioinformática.

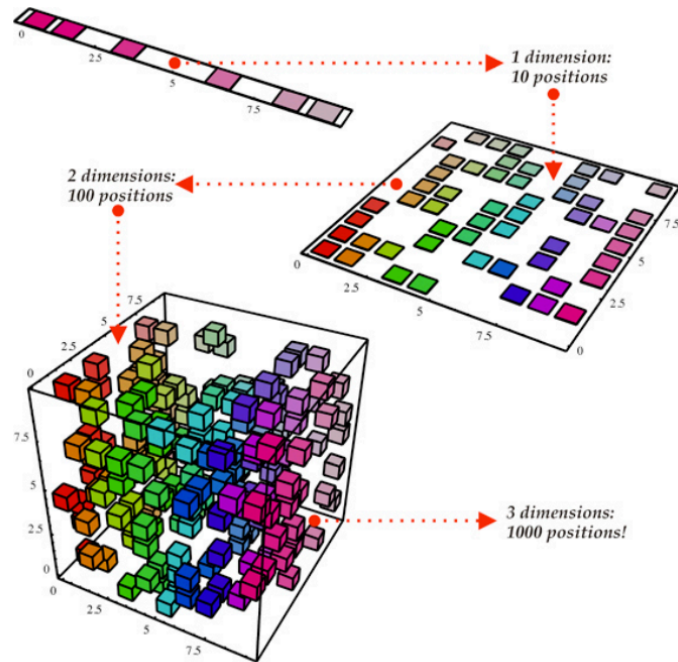


Figura 3.5: Etapas en la reducción de dimensiones.

### 3.3. Red neuronal convolucional

En esta sección profundizaremos en las redes neuronales, desde la inspiración biológica hasta las redes convolucionales más famosas y desarrolladas.

#### 3.3.1. Inspiración biológica

Una red neuronal biológica está compuesta por aproximadamente 86 billones de neuronas, conectadas entre ellas. Los científicos estiman que existen más de 500 millones de billones de conexiones en el cerebro humano. Incluso las redes neuronales artificiales más largas ni se acercan a este número.

Desde un punto de vista informatizado, podemos decir que una neurona es una unidad excitable que puede procesar y transmitir información mediante señales eléctricas y químicas. Esta unidad (la neurona) es el principal componente en nuestro sistema nervioso.

#### Estructura

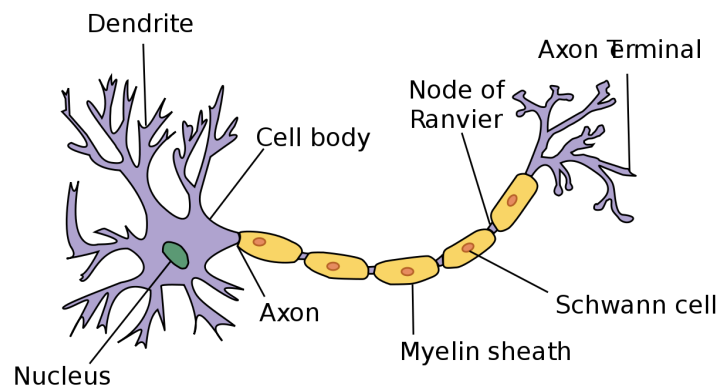


Figura 3.6: Estructura de una neurona biológica.

Las partes más importantes que componen una neurona son:

- El soma (o cuerpo celular): procesa las activaciones entrantes y las convierte en activaciones de salida.
- El núcleo: contiene material genético en forma de ADN.
- Las dendritas: son fibras que nacen desde el soma y provee de zonas receptoras para recibir las activaciones de otras neuronas.
- Axones: son fibras que actúan como líneas transmisoras que envían la activación a otras neuronas.

## Funcionamiento

Los pasos clave en el funcionamiento de una neurona son:

1. Las señales de otras neuronas se almacenan en las dendritas.
2. El soma agrupa las señales entrantes (espacial y temporalmente).
3. Cuando se recibe la suficiente cantidad de señal la neurona se "dispara" (produce una diferencia de potencial).
4. Esta diferencia de potencial es transmitido a lo largo del axón hacia otras neuronas o hacia estructuras fuera del sistema nervioso (Por ejemplo músculos).
5. En caso de no recibir suficiente señal la señal almacenada en las dendritas caerá rápidamente y la neurona no se disparará.

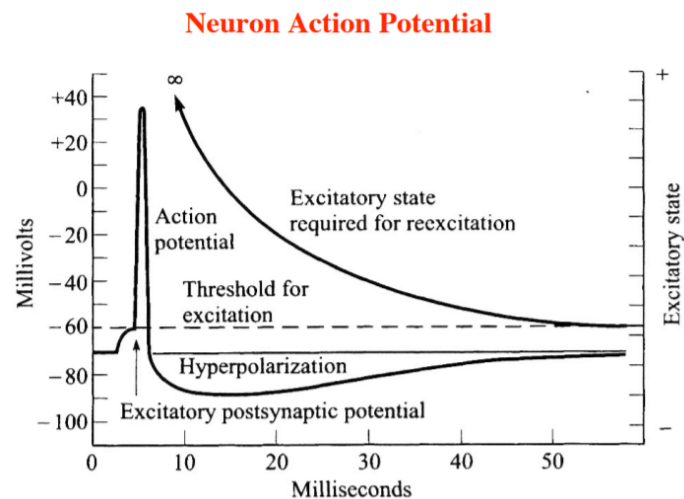


Figura 3.7: Diagrama que muestra el disparo de una neurona

### 3.3.2. El perceptrón

El perceptrón es un modelo lineal usado para la clasificación binaria. En el campo de las redes neuronales el perceptrón es considerado la neurona artificial.

El primer modelo matemático de dicha neurona artificial fué presentado por el psiquiatra y neuroanatomista Warren McCulloch y el matemático Walter Pitts en 1943. Este modelo se considera el precursor del perceptrón, y es llamado "Threshold Logic Unit." "TLU". Dicho modelo era capaz de simular puertas lógicas, como la AND, OR o la XOR.

#### Estructura

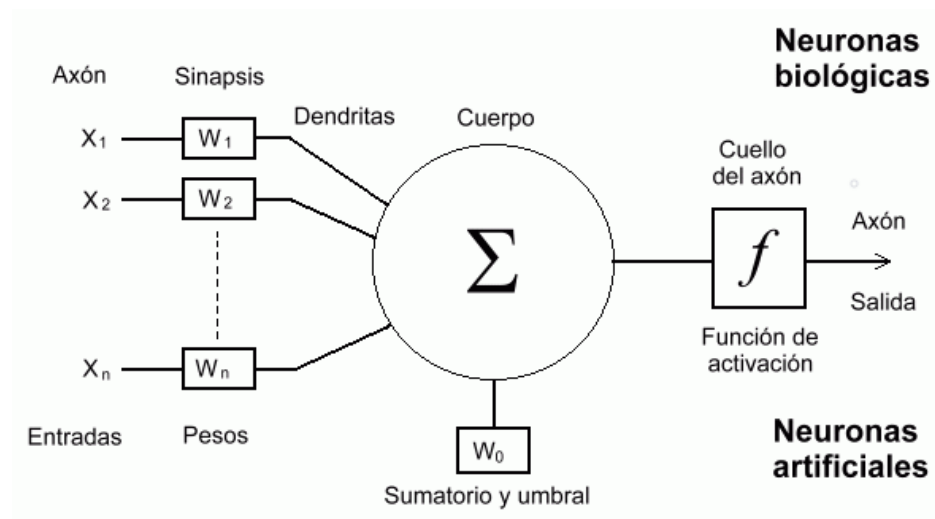


Figura 3.8: Comparativa de la estructura de la neurona biológica con el perceptrón

En la figura anterior podemos apreciar las distintas partes de la estructura de una neurona artificial:

- El conjunto de elementos de entrada  $(x_1, x_2, \dots, x_n)$ .
- Los pesos asociados a cada entrada  $(w_1, w_2, \dots, w_n)$ , los cuales se multiplicarán por cada elemento de entrada.
- Una función de agregación ( $\Sigma$ ) que sumará las multiplicaciones de los pesos con sus respectivos elementos de entrada. A esta agregación se le suma un factor de parcialidad (bias).
- Una función de activación ( $f$ ).

Como hemos dicho, el perceptrón es un algoritmo para **aprender** un clasificador binario. Los elementos externos llegarán al perceptrón para devolver un valor. Por ello, teniendo nuestro perceptrón  $f(\mathbf{x})$ , siendo  $\mathbf{x}$  dichos elementos externos, siendo  $\mathbf{b}$  el factor de parcialidad (bias), siendo  $\mathbf{w}$  los pesos asociados a cada elemento externo y siendo  $\mathbf{w} * \mathbf{x}$  el producto escalar de  $\mathbf{x}$  y  $\mathbf{w}$ :

$$f(x) = \begin{cases} 1 & \text{si } w * x + b > 0 \\ 0 & \text{E.O.C} \end{cases}$$

Este clasificador funciona solo en caso de que el conjunto de datos de entrada sea linealmente separable. Este hecho produce que durante el entrenamiento de dicho perceptrón la clasificación erre, produciendo un clasificador de muy baja confianza.

Para poder entender cómo funciona dicho algoritmo primero tendremos que observar las distintas funciones de activación que podemos utilizar.

### Funciones de activación

Las funciones de activación son funciones que dependiendo de lo que reciban se comportarán de una manera u otra. Un ejemplo es la función escalón de Heaviside, que devuelve 0 si el número que recibe es negativo o 1, si el número es mayor o igual que 0:

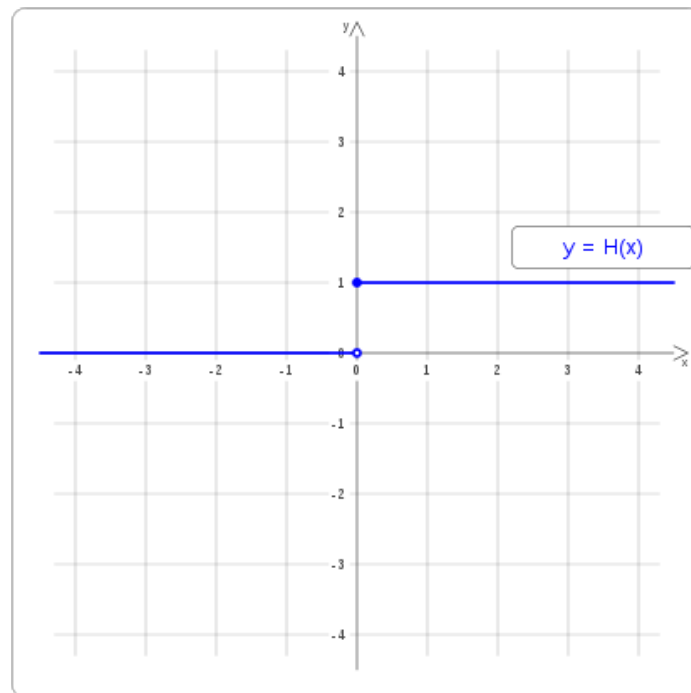


Figura 3.9: Gráfica de la función escalón de Heaviside.

Empezaremos por comentar las funciones de activación más comunes en el campo de las redes neuronales:

- ReLU (Rectifier Linear Unit): es la función que define la parte positiva de sus argumentos.

$$f(x) = x^+ = \max(0, x)$$

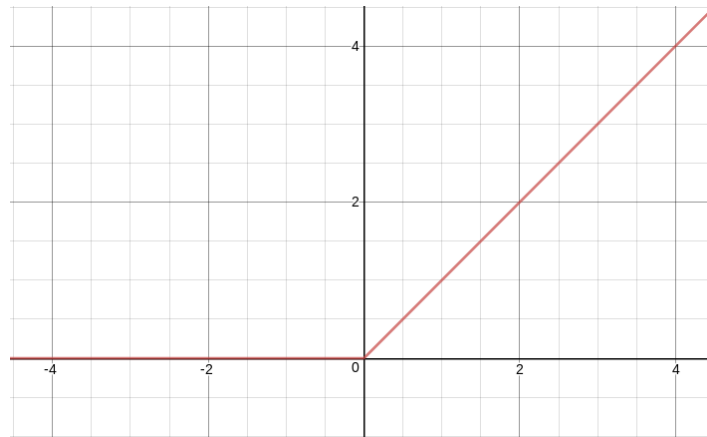


Figura 3.10: Gráfica de la función ReLU.

- Sigmoide ( $\sigma$ ): es una de las funciones de activación más utilizadas. En este campo sirven para introducir la "no linealidad" en un modelo. Ya que el producto escalar entre los pesos y  $x$  es una combinación lineal, al añadirle esta "no linealidad" el resultado de dicha combinación se suaviza.

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

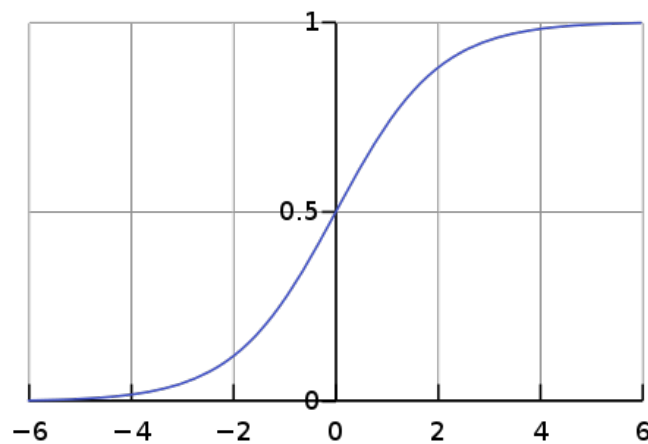


Figura 3.11: Gráfica de la función sigmoide.

- Softmax (función exponencial normalizada): es utilizada en la última capa de los clasificadores basados en redes neuronales, debido a que devuelve un vector de  $K$  valores reales a otro vector con  $K$  valores reales que suman 1. Si uno de los valores es negativo o pequeño, la función le dará poca probabilidad, mientras que si algún valor es más grande que el resto le dará más probabilidades.

Esta función solo se puede utilizar cuando las clases del clasificador son mutuamente exclusivas.

$$f(x) = \ln 1 + e^x$$

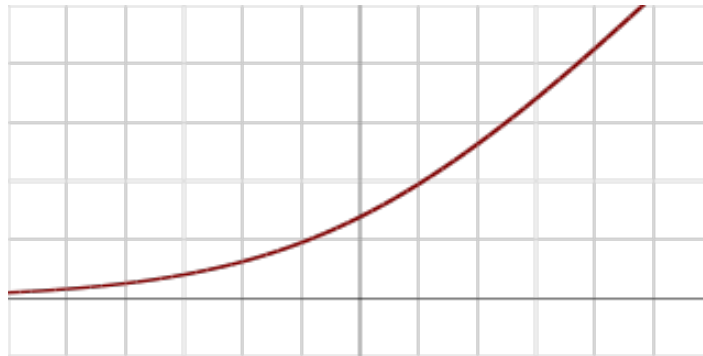


Figura 3.12: Gráfica de la función softmax.



Entrenamiento

**3.4. Planificación**

**3.5. Presupuesto**

**3.6. Conclusiones**

---

## 4. Análisis del problema

---

### 4.1. Introducción

En este capítulo explicaremos...

### 4.2. Requisitos de información

Los requisitos de información son...

### 4.3. Requisitos funcionales

Los requisitos funcionales son...

### 4.4. Requisitos no funcionales

Los requisitos no funcionales son...

### 4.5. Conclusiones

En este capítulo concluimos que...

---

## 5. Diseño del problema

---

### 5.1. Introducción

En este capítulo explicaremos...

### 5.2. Conclusiones

En este capítulo concluimos que...

---

## 6. Implementación

---

### 6.1. Introducción

En este capítulo explicaremos...

### 6.2. Herramientas

### 6.3. Conclusiones

En este capítulo concluimos que...

---

## 7. Pruebas

---

### 7.1. Introducción

En este capítulo explicaremos...

### 7.2. Conclusiones

En este capítulo concluimos que...

---

## 8. Conclusiones

---

---

## 9. Bibliografía

---

- [1] Agustín Borrego, Daniel Ayala, Inma Hernández, Carlos R. Rivero, and David Ruiz. Generating rules to filter candidate triples for their correctness checking by knowledge graph completion techniques. In *K-CAP*, pages 115–122, 2019. doi: 10.1145/3360901.3364418.
- [2] Universidad de Sevilla. Página principal de la escuela técnica de ingeniería informática, 2020. URL <https://www.informatica.us.es>.