



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**DESARROLLO DE APLICACIÓN PARA LA DETECCIÓN DE ENFERMEDADES EN
CULTIVOS DE MILPA UTILIZANDO REDES NEURONALES CONVOLUCIONALES**

Edgar Arnoldo Aldana Arriola

Asesorado por Ing. Sergio Leonel Gómez Bravo

Guatemala, mayo de 2023

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DESARROLLO DE APLICACIÓN PARA LA DETECCIÓN DE ENFERMEDADES EN
CULTIVOS DE MILPA UTILIZANDO REDES NEURONALES CONVOLUCIONALES**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

EDGAR ARNOLDO ALDANA ARRIOLA
ASESORADO POR ING. SERGIO LEONEL GÓMEZ BRAVO

AL CONFERIRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MAYO DE 2023

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton De León Bran
VOCAL IV	Br. Kevin Vladimir Cruz
VOCAL V	Br. Fernando José Paz González
SECRETARIA	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. Juan Álvaro Díaz Ardavin
EXAMINADOR	Ing. Álvaro Giovanni Longo Morales
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DESARROLLO DE APLICACIÓN PARA LA DETECCIÓN DE ENFERMEDADES EN CULTIVOS DE MILPA UTILIZANDO REDES NEURONALES CONVOLUCIONALES

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 15 de marzo de 2022.

A handwritten signature in black ink, appearing to read 'E. Aldana Arriola', with a large, stylized initial 'E' that loops around the start of the name.

Edgar Arnoldo Aldana Arriola

Guatemala, 16 de marzo de 2023

Ingeniero
Carlos Alfredo Azurdia
Coordinador de Privados y Trabajos de Tesis
Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería - USAC

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **Edgar Arnoldo Aldana Arriola** con carné **201602797** y CUI **3051 50685 0201** titulado **"DESARROLLO DE APLICACIÓN PARA LA DETECCIÓN DE ENFERMEDADES EN CULTIVOS DE MILPA UTILIZANDO REDES NEURONALES CONVOLUCIONALES"**, luego de corroborar que el mismo se encuentra finalizado, lo he revisado y doy fé de que el mismo cumple con los objetivos propuestos en el respectivo protocolo, por consiguiente, procedo a la aprobación correspondiente.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Sergio Leonel Gómez Bravo
Colegiado No. 12671

SERGIO LEONEL GÓMEZ BRAVO
INGENIERO EN CIENCIAS Y SISTEMAS
COL. 12671



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 22 de marzo de 2023

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **EDGAR ARNOLDO ALDANA ARRIOLA** con carné **201602797** y CUI **3051 50685 0201** titulado “**DESARROLLO DE APLICACIÓN PARA LA DETECCIÓN DE ENFERMEDADES EN CULTIVOS DE MILPA UTILIZANDO REDES NEURONALES CONVOLUCIONALES**”, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



SIST.LNG.DIRECTOR.1.EICCSS.2023

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área y la aprobación del área de lingüística del trabajo de graduación titulado: **DESARROLLO DE APLICACIÓN PARA LA DETECCIÓN DE ENFERMEDADES EN CULTIVOS DE MILPA UTILIZANDO REDES NEURONALES CONVOLUCIONALES**, presentado por: **Edgar Arnoldo Aldana Arriola**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

“ID Y ENSEÑAD A TODOS”



Ingeniero Carlos Gustavo Alonzo
DIRECTOR
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, mayo de 2023



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Decanato
Facultad e Ingeniería

24189101- 24189102

LNG.DECANATO.OIE.3.2023

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería En Ciencias Y Sistemas, al Trabajo de Graduación titulado: **DESARROLLO DE APLICACIÓN PARA LA DETECCIÓN DE ENFERMEDADES EN CULTIVOS DE MILPA UTILIZANDO REDES NEURONALES CONVOLUCIONALES**, presentado por: **Edgar Arnoldo Aldana Arriola** después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Firmado digitalmente por AURELIA ANABELA CORDOVA ESTRADA Fecha: 10/05/2023 06:17:45 p.m. Razón: Orden de impresión Ubicación: Facultad de Ingeniería, USAC.

Inga. Aurelia Anabela Córdova Estrada
Decana



Guatemala, mayo de 2023

Para verificar validez de documento ingrese a <https://www.ingenieria.usac.edu.gt/firma-electronica/consultar-documento>

Tipo de documento: Correlativo para orden de impresión Año: 2023 Correlativo: 3 CUI: 3051506850201

Escuelas: Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, - Escuela de Ciencias, Regional de Ingeniería Sanitaria y Recursos Hidráulicos (ERIS). Postgrado Maestría en Sistemas Mención Ingeniería Vial. Carreras: Ingeniería Mecánica, Ingeniería Electrónica, Ingeniería en Ciencias y Sistemas. Licenciatura en Matemática. Licenciatura en Física. Centro de Estudios Superiores de Energía y Minas (CESEM). Guatemala, Ciudad

ACTO QUE DEDICO A:

Mis padres

Quiero expresar mi gratitud a mis padres por su esfuerzo incansable y sacrificio a lo largo de mi carrera universitaria. Su amor y apoyo me han impulsado a seguir adelante incluso en los momentos más difíciles. Este logro es de ustedes tanto como mío.

Mis hermanos

Gracias a mis hermanos por estar siempre presentes, como hermano mayor espero servir como ejemplo de que todo esfuerzo rinde sus frutos independientemente del camino que se tome.

Mis amigos

Quiero agradecer a mis amigos por su apoyo constante y motivación, por acompañarme en todos los momentos de estrés y ansiedad en la carrera. Gracias por su amistad incondicional, por brindarme su tiempo y por compartir tantos momentos memorables que atesoraré por siempre.

AGRADECIMIENTOS A:

**Universidad de San
Carlos de Guatemala**

Expresar mi gratitud a la universidad por brindarme la oportunidad de cursar una carrera y adquirir conocimientos y habilidades valiosas que, utilizaré en mi recorrido profesional y en mi vida diaria.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	III
GLOSARIO	V
RESUMEN.....	VII
OBJETIVOS.....	IX
INTRODUCCIÓN	XI
1. CULTIVOS DE MILPA.....	1
1.1. Características.....	1
1.1.1. Hojas	1
1.1.2. Raíces	2
1.1.3. Tallo.....	3
1.1.4. Mazorca.....	4
1.2. Enfermedades	5
1.2.1. Roya común.....	6
1.2.2. Moteado clorótico	7
2. INTELIGENCIA ARTIFICIAL	9
2.1. Historia	9
2.2. Técnicas	11
2.2.1. Machine Learning	12
2.2.1.1. Aprendizaje por refuerzo.....	12
2.2.1.2. Aprendizaje supervisado.....	13
2.2.1.3. Aprendizaje no supervisado.....	14
2.2.2. Heurística	15
2.2.3. Sistemas expertos	16

2.2.4.	Redes neuronales	17
2.3.	Banco de datos.....	17
3.	REDES NEURONALES	19
3.1.	Redes neuronales biológicas	20
3.2.	Redes neuronales artificiales	23
3.3.	Redes neuronales convolucionales.....	26
3.3.1.	Capa convolucional	28
3.3.2.	Capa Poling.....	31
3.3.3.	Función de activación.....	32
4.	DESARROLLO DE APLICACIÓN WEB.....	35
4.1.	Desarrollo del Frontend.....	36
4.2.	Recopilación de imágenes	37
4.3.	Desarrollo del Backend	38
4.3.1.	Lenguaje Python para el desarrollo de redes neuronales.....	38
4.3.2.	Programación de redes neuronales convolucionales.....	39
4.3.3.	Entrenamiento de red neuronal convolucional	42
	CONCLUSIONES.....	47
	RECOMENDACIONES	49
	REFERENCIAS	51
	APÉNDICES.....	55

ÍNDICE DE ILUSTRACIONES

FIGURAS

Figura 1.	Hoja de la planta del maíz	2
Figura 2.	Raíces de la planta del maíz.....	3
Figura 3.	Tallo de la planta del maíz	4
Figura 4.	Mazorca	5
Figura 5.	Roya del maíz..	7
Figura 6.	Moteado clorótico.....	8
Figura 7.	Aprendizaje por refuerzo.....	13
Figura 8.	Aprendizaje supervisado.....	14
Figura 9.	Aprendizaje no supervisado.....	15
Figura 10.	Neurona biológica.....	21
Figura 11.	Sinapsis	23
Figura 12.	Neurona artificial	25
Figura 13.	Red neuronal artificial	26
Figura 14.	Red neuronal convolucional.....	27
Figura 15.	Capa convolucional.....	29
Figura 16.	Convoluciones	30
Figura 17.	Pooling.....	31
Figura 18.	Aplicación web.	36
Figura 19.	Librerías usadas en Python	39
Figura 20.	Creación de capas convolucionales y pooling en Python	40
Figura 21.	Creación de capas ocultas en Python.....	41
Figura 22.	Compilación de la red neuronal en Python	42
Figura 23.	Modificaciones a imágenes en Python.....	43

Figura 24.	Carga del banco de datos en Python	44
Figura 25.	Aplicación de modificaciones en Python	45
Figura 26.	Utilización de la red neuronal en Python	46

TABLAS

Tabla 1.	Cerebro humano vs computador	20
-----------------	------------------------------------	----

GLOSARIO

Convolución	Es una función, que de forma lineal y continua realiza que recibe una data de entrada y se encarga de convertirla en una nueva data de salida.
DENDRAL	Primer sistema experto en ser utilizado en propósitos reales.
ELIZA	ELIZA es el primer programa conversacional de la historia, conocidos hoy en día más comúnmente como chat box.
EURISKO	Sistema experto capaz de aprender a medida que funciona.
Fasciculadas	Es un tipo de raíz en la que todas las ramificaciones tienen la misma importancia, no hay tienen un eje principal.
GPS	Generador General de Problemas, programa capaz de resolver problemas: geométricos, trabajar con lógica posicional, jugar ajedrez, entre otros.
Hiperparámetros	Son parámetros que se pueden ajustar para controlar el proceso para el entreno de un modelo.

IA	Inteligencia Artificial
JPEG	Join Photographic Experts Group
LISP	Familia de lenguajes de programación que se caracterizan por ser multiparadigma.
Mycin	Sistema experto inspirado en DENDRAL.
Overfitting	Es el sobreajuste que se le da a un algoritmo de aprendizaje.
PNG	Portable Network Graphics
PROLOG	Lenguaje de programación interpretado que se suele utilizar en el área de inteligencia artificial.
Pústulas	Pequeñas protuberancias que se producen en la epidermis de la planta las que suelen cambiar de color.
ReLU	Rectified Linear Unit
Tanh	Tangente Hiperbólica
Test de Turing	Es un examen que pone a prueba a una máquina para determinar si esta posee cierto grado de inteligencia.

RESUMEN

En los cultivos, existen enfermedades que pueden perjudicar su desarrollo debido a un mal manejo o a la falta de tratamiento oportuno. Por lo tanto, el desarrollo de una herramienta capaz de detectar y proporcionar información sobre las enfermedades sería de gran beneficio, porque permitiría salvar la cosecha si las plantas son tratadas a tiempo.

La herramienta creada para la detección de enfermedades es una aplicación web desarrollada en React que consulta una API programada en Python. La API posee una red neuronal que recibe una imagen y se encarga de detectar la enfermedad, proporcionando el resultado a la aplicación web. Esta muestra la enfermedad, su causa y un posible tratamiento que puede aplicarse a las plantas.

La red neuronal fue diseñada para poder entrenar con un banco de datos pequeño, ya que en la recopilación de imágenes no se logró obtener una cantidad suficiente. Debido a esto, la precisión de la detección no es óptima en todas las ocasiones, y en algunos casos la red neuronal no proporciona un resultado correcto.

OBJETIVOS

General

Desarrollar una aplicación que detecte las enfermedades Roya Común y Moteado Clorótico en cultivos de milpa a partir de imágenes procesadas por redes neuronales convolucionales.

Específicos

1. Recopilar imágenes de las enfermedades Roya Común y Moteado Clorótico en cultivos de milpa para la creación de una base de datos de entrenamiento.
2. Diseñar y entrenar una red neuronal convolucional que se encargue del reconocimiento de patrones específicos de imágenes de las enfermedades Roya Común y Moteado Clorótico.
3. Crear de una aplicación que permita la carga de imágenes JPEG o PNG para la detección de las enfermedades Roya Común y Moteado Clorótico, y muestre posibles tratamientos que puede aplicarse al cultivo.

INTRODUCCIÓN

Los cultivos de milpa son un sistema de cultivo que varía según la región, pero en Mesoamérica se compone de tres cultivos: maíz, frijol y calabaza. En Guatemala, el maíz es uno de los principales cultivos y es muy importante para muchas personas.

Los cultivos de milpa se componen de cuatro partes importantes: hojas, raíces, tallo y mazorca, cada una de estas partes puede ser atacada por diversas enfermedades que afectan la productividad y el desarrollo de la planta. En este artículo se discutirán las características de las partes de la planta del maíz y algunas de las enfermedades que pueden afectar la milpa, como la Roya Común y el Moteado Clorótico.

La inteligencia artificial es una rama de la computación que ha ganado mucha relevancia en la actualidad debido a su aplicación en distintos campos, como la medicina, la educación y el comercio, entre otros. Esta tecnología permite a las computadoras aprender y tomar decisiones basadas en la información que reciben.

La historia de la inteligencia artificial se remonta a la propuesta de Alan Turing de una prueba para comprobar si una máquina no biológica puede demostrar inteligencia. Desde entonces, se han desarrollado programas y sistemas capaces de realizar tareas cada vez más complejas, como mantener una conversación o diagnosticar enfermedades.

Existen dos enfoques principales en la inteligencia artificial: uno busca emular la estructura del cerebro humano mediante redes neuronales, mientras que el otro se enfoca en desarrollar algoritmos inteligentes sin seguir esta estructura. A pesar de los ciclos de éxito y desaparición de apoyos financieros, la inteligencia artificial se ha consolidado como un área del conocimiento de gran interés para la investigación científica.

Las redes neuronales artificiales son una técnica de inteligencia artificial que busca emular el comportamiento de las redes neuronales biológicas del cerebro humano. Se comienza describiendo el funcionamiento de las neuronas biológicas y la organización neuronal en el cerebro humano, para luego compararlas con las redes neuronales artificiales.

A pesar de las diferencias significativas entre ambas, las redes neuronales artificiales tienen la capacidad de aprender y tomar decisiones en base a lo aprendido, esto las hace una técnica muy poderosa en el campo de la inteligencia artificial.

1. CULTIVOS DE MILPA

La palabra milpa hace referencia a dos sistemas de cultivo que varían según la región; en las regiones de Mesoamérica la milpa se constituye de un agro ecosistema que se compone de maíz, frijol y calabaza; fuera de las regiones de Mesoamérica este nombre se les da solo a los cultivos que se componen únicamente de maíz.

El maíz es uno de los cereales con mayor volumen de producción a nivel mundial y es uno de los principales cereales que se cultivan en la región de Guatemala, esto les da bastante relevancia a los cultivos de milpa dentro del país porque muchas personas son dependientes de estos cultivos.

1.1. Características

Los cultivos de milpa de maíz se componen de planas de maíz y ésta se compone de 4 partes importantes: las hojas, raíces, tallo y mazorca. Cada una de estas partes puede ser atacada por ciertas enfermedades que afectan la productividad, el desarrollo o incluso causar la muerte de la misma planta.

1.1.1. Hojas

Las hojas de la planta del maíz son: largas, grandes y verdes; estas se encuentran adheridas directamente al tallo. Las hojas de la planta del maíz presentan bellos en la parte superior de la hoja y las hojas tienden a ser afiladas en los bordes.

Figura 1.

Hoja de la planta del maíz



Nota. Free Nature Stock (2022). *Campo de maíz verde* [Fotografía]. Pexels. (<https://www.canva.com/photos/MADGyeynBQA-selective-focus-photography-of-green-corn-fields/>). Derechos de autor 2023 por Free Nature Stock.

1.1.2. Raíces

Las raíces son fasciculadas y como las raíces de muchas plantas estas aportan nutrientes y aportan de un anclaje a la planta.

Figura 2.

Raíces de la planta del maíz



Nota. Corelens (2022). *Raíz de la planta de maíz* [Fotografía]. Canva. (<https://www.canva.com/photos/MAFU3I6JYmk-plant-with-bare-roots/>). Derechos de autor 2023 por Corelens.

1.1.3. Tallo

El tallo se compone de tres capas y estas son: una epidermis que es impermeable y transparente; una pared vegetal por la que circulan los nutrientes de la planta, y en la parte más interna se encuentra la medula, esta se compone de un tejido esponjoso que se encarga de almacenar los azúcares que necesita la planta.

Figura 3.

Tallo de la planta del maíz



Nota. L. Venini (2020). *Tallo del maíz* [Fotografía]. Elabcrural. (<https://elabcrural.com/green-snap-o-quebrado-en-verde-del-tallo-del-maiz-de-que-se-trata/>). Derechos de autor 2023 por L. Venini.

1.1.4. Mazorca

La mazorca del maíz, también conocida como elote, es una espiga larga y gruesa que tiene granos incrustados y se encuentra recubierta por hojas. Las mazorcas se pueden consumir de diferentes maneras, ya sea cocidas, asadas, enlatadas o en productos derivados como harina de maíz o tortillas.

Figura 4.

Mazorca



Nota. M. Mali (2016). *Maíz amarillo* [Fotografía]. Pexels. (<https://www.pexels.com/photo/yellow-corn-547263/>). Derechos de autor 2023 por M. Mali.

1.2. Enfermedades

Evitar las enfermedades del maíz puede considerarse como un reto, y se debe tener ciertas precauciones para mantener las enfermedades al margen. Son varias las razones por las que una planta de maíz puede enfermar y algunas son: falta de rotación de cultivos, prácticas deficientes, el medio ambiente, entre otros.

Existen diversas enfermedades y dependiendo de ellas, la planta de maíz puede verse afectada en ciertas zonas específicas, ya sea en las raíces, hojas, tallo o en la mazorca.

1.2.1. Roya común

La Roya Común es causada por un hongo llamado *Puccinia sorghi*. La enfermedad producida por el hongo reduce el rendimiento en la planta. Esta enfermedad reduce las áreas de la planta encargadas de realizar la fotosíntesis y disminuye el peso de los granos del maíz.

La Roya Común tiende a afectar la planta del maíz en zonas tropicales y subtropicales. La temperatura idónea para que este hongo cause una infección severa es de 27 a 33 grados centígrados y con una humedad considerablemente alta. Si la infección de este hongo llega a afectar la planta en una etapa temprana de su vida, puede afectar severamente el desarrollo de la misma.

Los síntomas que causa esta enfermedad inician con pequeñas manchas en las hojas que se observan por ambos lados. Posteriormente, estas manchas se convierten en manchas color café y un poco sobresalientes. Luego, estas se convierten en pústulas de color dorado, que cambian a negro a medida que la planta se va desarrollando. La infección afecta solo las hojas, y como efecto secundario, el tallo se ve afectado, creciendo débil.

Figura 5.

Roya del maíz



Nota. Pioneer Argentina (2016). *Roya común* [Fotografía]. Pioneer. (https://www.pioneer.com/ar/articulos/monitoreo_control_roya.html). Derechos de autor 2023 por Pioneer Argentina.

1.2.2. Moteado clorótico

El moteado clorótico afecta principalmente a las hojas, causando la aparición de grandes bandas de un color verde más claro que el verde normal de las hojas. Las hojas que se ven afectadas por el virus mueren, y debido a los problemas que la planta presenta en las hojas, esta puede presentar enanismo si se ve afectada en una etapa temprana de su crecimiento. El virus también afecta la producción de la planta, y el desarrollo de las mazorcas se altera, causando que no crezcan a su tamaño normal y que las espigas se deformen.

Esta enfermedad se transmite por lo general mediante heridas causadas en las plantas de maíz. Los insectos juegan un papel crucial en la propagación

del virus, al alimentarse de plantas infectadas y posteriormente transmitir el virus a una planta sana. Existe otra forma de contagio menos frecuente, que es mediante heridas en la semilla, causadas por las herramientas de trabajo.

Figura 6.

Moteado clorótico



Nota. P. Kodwaran (2016). *Moteado clorótico* [Fotografía]. P. Kodwaran. (<https://plantwiseplusknowledgebank.org/cms/10.1079/pwkb.20167800947/asset/c9b397c0-cfe6-45a7-95fc-18fa4c63823e/assets/graphic/maize-lethal-necrosis-disease-global-1.jpg>).

Derechos de autor 2023 por P. Kodwaran.

2. INTELIGENCIA ARTIFICIAL

La inteligencia artificial es una de las ramas de la computación que más está dando de qué hablar en esta época. La inteligencia artificial es aplicable en todos los campos, brindando soporte, simplificando procesos, automatizando, entre otras cosas. La inteligencia artificial no es más que la capacidad que tienen los ordenadores de aprender y tomar decisiones basándose en lo aprendido.

2.1. Historia

Los criterios de desarrollo de la IA se remontan a la intuición del genio matemático inglés Alan Turing y el apelativo Inteligencia Artificial se debe a McCarthy quien organizó una conferencia en el Dartmouth College (Estados Unidos) para discutir la posibilidad de construir máquinas inteligentes {...} Como resultado de esta reunión, se establecieron los primeros lineamientos de la hoy conocida como Inteligencia Artificial... (Ponce, 2014, p. 16)

Alan Turing es el primero en proponer una prueba para comprobar si una máquina no biológica puede demostrar ser inteligente. “Esta prueba conocida como “test de Turing” se fundamenta en la hipótesis de que, si una máquina se comporta en todos los aspectos como inteligente, entonces debe ser inteligente” (Ponce, 2014, p. 18). Alan Turing despertó el interés en las mentes científicas que en los años posteriores se centrarían en el desarrollo de las curiosas máquinas inteligentes. De las contribuciones que Alan Turing aportó hay una que es de las que más destaca y esta es la del diseño de una computadora capaz de jugar al ajedrez.

Tiempo después, en 1957, Alan Newell y Herbert Simon desarrollaron un programa llamado GPS, capaz de encontrar soluciones a problemas como las torres de Hanói. Sin embargo, el enfoque basado en prueba y error lo hacía inaplicable en la vida cotidiana. Por ello, en los años 70 se inició un proyecto para resolver problemas cotidianos, dando origen a los sistemas expertos.

En 1965, se desarrolló Eliza, el primer programa capaz de mantener una conversación con las personas mediante textos. En 1967, se creó el primer sistema experto, llamado Dendral, un intérprete de espectrograma de masa. En 1974, nació el primer sistema implementado en hospitales capaz de diagnosticar trastornos sanguíneos, llamado Mycin, inspirado en Dendral.

En los años 80, se preparó el terreno para la inteligencia artificial, creando los primeros lenguajes especializados para su desarrollo, como LISP y PROLOG. Esto permitió crear sistemas expertos más complejos, capaces incluso de mejorarse a sí mismos, como el Software Eurisko.

“También podemos destacar la importante intervención de Arthur Samuel, que desarrollo un programa de juego de damas capaz de aprender de su propia experiencia; Selfridge, que estudiaba el reconocimiento visual por computadora” (Ponce, 2014, p. 19).

A partir de los modelos existentes y todas las personas que aportaron en el desarrollo de la inteligencia artificial surgieron ideas que tenían enfoques distintos pero que tenían un mismo fin: hacer que la inteligencia artificial emule el pensamiento humano.

Los enfoques se centraban especialmente en cómo es que una IA puede o debería pensar, y aquí es donde el camino se divide en dos: uno se centra en

emular una estructura lo más parecida al cerebro humano mediante redes neuronales y el otro únicamente se enfocaba en que los algoritmos fueran inteligentes, no importando si esto se alejaba de la estructura o del cómo piensa un cerebro humano.

Los dos tipos de enfoques en la Inteligencia Artificial pueden diferir bastante, pero ambos pretenden alcanzar una misma meta, y esto es: llegar a emular el pensamiento humano.

La historia de la IA ha sido testigo de ciclos de éxito, injustificado optimismo y la consecuente desaparición de entusiasmo y apoyos financieros. También ha habido ciclos caracterizados por la introducción de nuevos y creativos enfoques y de un sistemático perfeccionamiento de los mejores. Por sus implicaciones con áreas como la medicina, psicología, biología, ética y filosofía entre otras, esta rama del conocimiento ha tenido que lidiar con fuertes grupos oponentes y críticas desde sus orígenes; sin embargo, siempre existió un grupo de personas interesadas en el área lo que permitió que se consolidara como un área del conocimiento de gran interés para la investigación científica. (Ponce, 2014, p. 20)

2.2. Técnicas

Existen diversas técnicas para trabajar con la inteligencia artificial y tiene sentido, pues no son pocas las áreas en las que se puede aplicar esta herramienta. No es como que exista una de estas técnicas sea mejor, sino que dependiendo del uso que se le quiera dar, existen técnicas que se acoplan mejor que otras a resolver cierto tipo de problemas, por lo que cada una tiene su relevancia y su aspecto inteligente.

2.2.1. Machine Learning

Machine Learning es un campo de la informática que se centra en el desarrollo de algoritmos y sistemas que pueden aprender y mejorar automáticamente a partir de la experiencia y los datos.

En esencia, el aprendizaje automático implica la programación de una computadora para que pueda detectar patrones y tendencias en los datos, y luego usar esta información para tomar decisiones o hacer predicciones. El aprendizaje automático se divide en tres categorías principales: supervisado, no supervisado y de refuerzo.

Una de las aplicaciones más populares del aprendizaje automático es la clasificación de datos. Por ejemplo, se puede usar el aprendizaje automático para identificar el spam en los correos electrónicos o para clasificar imágenes en diferentes categorías. Otro ejemplo es la predicción de resultados, como los precios de las acciones o el rendimiento de los estudiantes.

El aprendizaje automático se ha convertido en una herramienta clave para muchas empresas y organizaciones, que utilizan algoritmos de aprendizaje automático para analizar grandes conjuntos de datos y obtener información valiosa sobre los patrones de comportamiento de los clientes y los patrones de ventas. También se utiliza en la ciencia, la medicina y la investigación para descubrir patrones en grandes conjuntos de datos y hacer predicciones precisas.

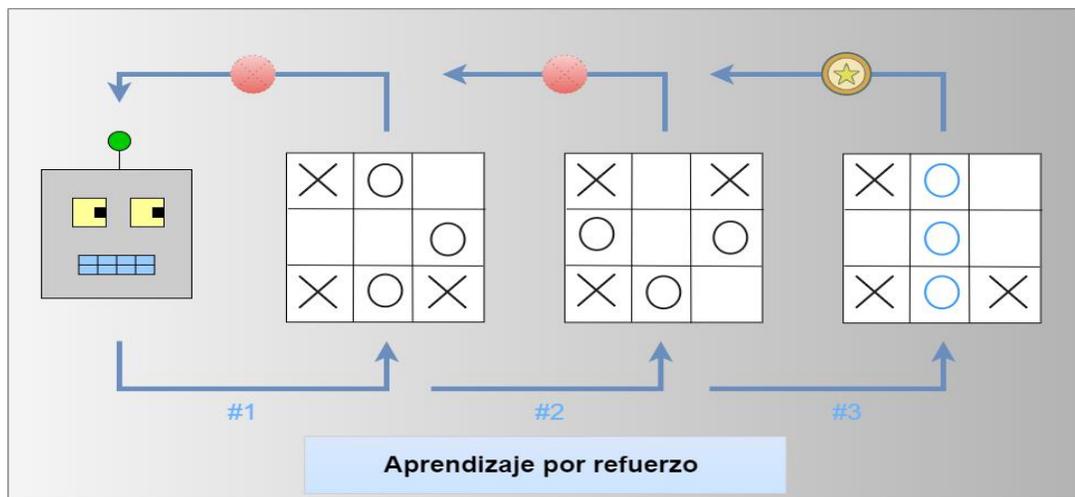
2.2.1.1. Aprendizaje por refuerzo

Este algoritmo se basa en la prueba y el error. El algoritmo sigue las condiciones impartidas y se encarga de realizar las pruebas, encontrando

recompensas a lo largo del camino. Teniendo una mayor recompensa el camino que lo lleve a realizar la tarea de forma más óptima. Este algoritmo, para poder funcionar, necesita la experiencia previa que él mismo genera.

Figura 7.

Aprendizaje por refuerzo

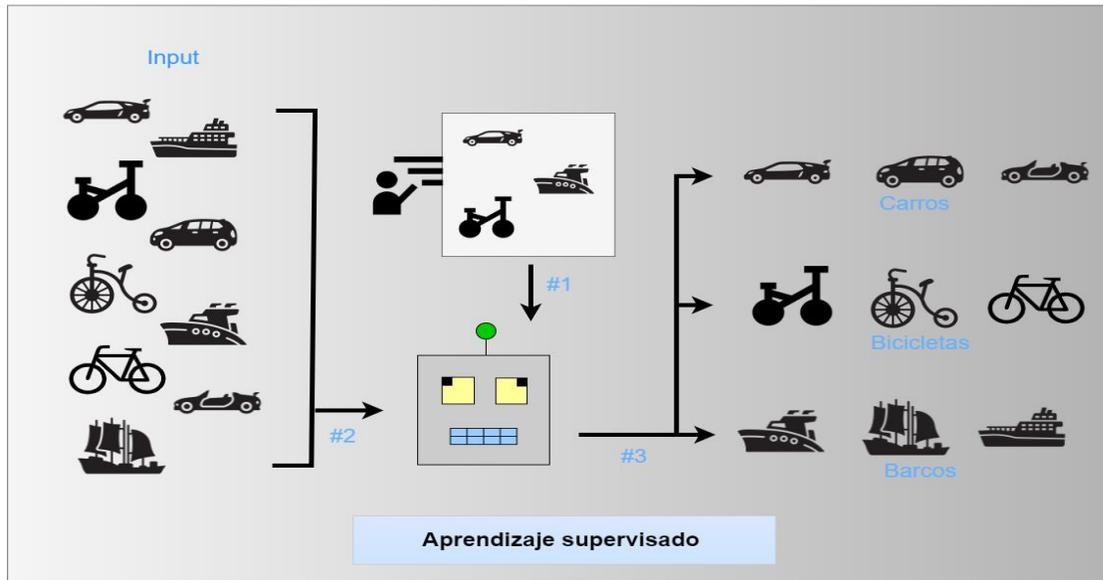


Nota. Muestra el proceso del aprendizaje por refuerzo. Elaboración propia, realizado con Drwa.io.

2.2.1.2. Aprendizaje supervisado

Estos algoritmos se caracterizan por utilizar modelos predictivos y por tener un entrenamiento previo antes de ponerlos en funcionamiento. El algoritmo recibe un banco de información en el que se le indican los patrones de ciertos elementos que debe reconocer. De esta forma, el modelo se ajusta hasta obtener los resultados adecuados.

Figura 8.
Aprendizaje supervisado



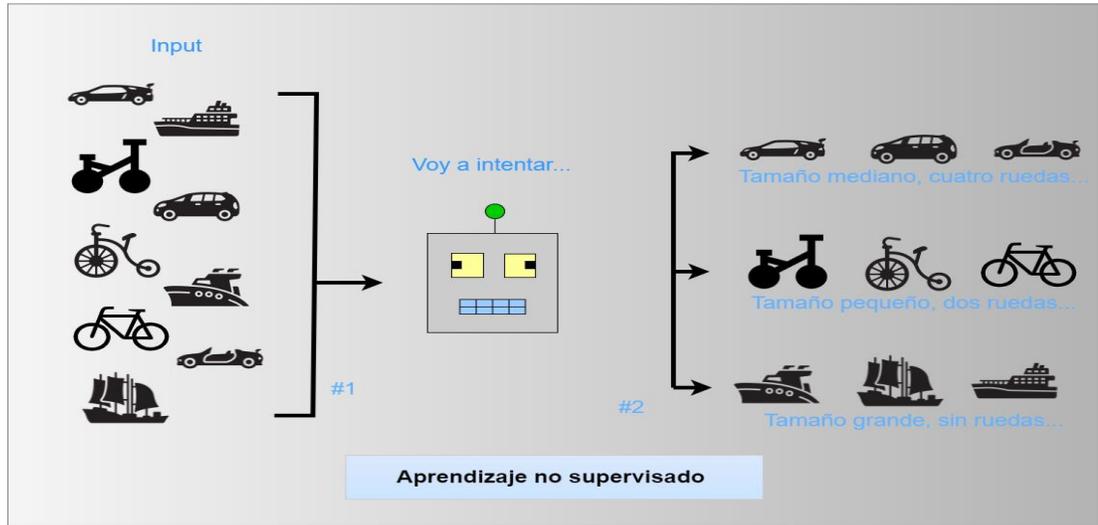
Nota. Muestra el proceso del aprendizaje supervisado. Elaboración propia, realizado con Drwa.io.

2.2.1.3. Aprendizaje no supervisado

Estos algoritmos son similares a los algoritmos de Aprendizaje Supervisado, la diferencia radica en que éstos no reciben un entrenamiento previo. El aprendizaje no supervisado no recibe retroalimentación externa, realiza su propio entrenamiento con los datos de entrada que recibe. En pocas palabras, el algoritmo va aprendiendo sobre la marcha.

Figura 9.

Aprendizaje no supervisado



Nota. Muestra el proceso del aprendizaje no supervisado. Elaboración propia, realizado con Drwa.io.

2.2.2. Heurística

La heurística es un proceso de resolución de problemas que se basa en la experiencia, la intuición y la capacidad de aprendizaje. Es una técnica que se utiliza para encontrar soluciones prácticas y eficaces a problemas complejos, incluso en situaciones en las que no se dispone de información completa o precisa.

En la vida diaria se utiliza la heurística en muchas situaciones, sin siquiera darse cuenta. Por ejemplo, al enfrentarse a una nueva tarea, se puede utilizar la heurística para identificar patrones y hacer inferencias sobre lo que se espera de las personas. En otros casos, se puede utilizar la heurística para tomar decisiones

rápidas y efectivas sin tener que pasar por un proceso completo de análisis y evaluación.

Aunque la heurística puede ser muy útil en muchas situaciones, también tiene sus limitaciones. En algunos casos, puede llevar a la toma de decisiones incorrectas o subóptimas. Además, la heurística puede sesgar el pensamiento y limitar la capacidad para considerar todas las opciones disponibles.

Una de las limitaciones más comunes de la heurística es la tendencia a utilizar la información más fácilmente disponible en lugar de buscar información adicional que podría ser más relevante o precisa. Por ejemplo, se pueden basar las decisiones en la información que se presenta con mayor frecuencia, o que es más fácil de recordar, en lugar de buscar información extra, que podría ser más precisa o relevante.

2.2.3. Sistemas expertos

Los sistemas expertos son un tipo de inteligencia artificial diseñados para imitar la toma de decisiones y la resolución de problemas de un experto humano en un campo específico. Estos sistemas se basan en un conjunto de reglas y conocimientos que se utilizan para analizar datos y tomar decisiones, con el objetivo de ayudar a los usuarios a resolver problemas complejos y tomar decisiones informadas.

Un sistema experto típicamente está compuesto de tres componentes principales: una base de conocimientos, un motor de inferencia y una interfaz de usuario. La base de conocimientos contiene la información y las reglas específicas de un dominio en particular, mientras que el motor de inferencia es el algoritmo que utiliza la información contenida en la base de conocimientos para

analizar los datos y tomar decisiones. La interfaz de usuario permite a los usuarios interactuar con el sistema experto y recibir recomendaciones basadas en el análisis de datos y la información proporcionada.

2.2.4. Redes neuronales

Las redes neuronales artificiales no son más que una réplica de menor calidad de las redes neuronales que poseen los cerebros. Estas redes neuronales artificiales intentan emular el funcionamiento de las redes neuronales orgánicas, teniendo múltiples neuronas conectadas entre ellas, estas son capaces de aprender y de generar respuestas en base a las entradas que estas tengan.

2.3. Banco de datos

Los bancos de datos son el combustible para la inteligencia artificial, es gracias a los datos que se tienen disponibles que los algoritmos de la inteligencia artificial pueden desarrollar sus tareas. Si se tiene un banco de información de mala calidad, el algoritmo se ve severamente afectado, debido a que las respuestas que estos se basan en la información que aprenden, y como la fuente no es confiable, las respuestas del algoritmo tampoco lo serán.

3. REDES NEURONALES

En el cerebro humano existen millones de neuronas y cada una de estas neuronas se encuentra conectada a otras neuronas formando una red gigante con millones de conexiones. Si una neurona recibe varios estímulos de otras neuronas y dependiendo de los estímulos, esta puede decidir si se activa o no; al momento de que una neurona se activa envía señales a todas las neuronas a las que está conectada.

Las redes neuronales artificiales no tienen otro objetivo más que intentar recrear este comportamiento que les permite aprender y tomar decisiones en base a lo aprendido.

Las redes neuronales artificiales no tratan sino de emular el cerebro humano el cual funciona a base de aprendizaje por experiencia y de la extracción del conocimiento. Si bien se intenta imitar la estructura del cerebro humano existen diferencias bastante significativas entre las redes neuronales biológicas del cerebro humano y las redes neuronales artificiales.

Las computadoras actuales cuentan con procesadores que por lo general tienen de 4 u 8 núcleos que por lo general trabajan bastante rápido pero no son rivales para competir contra un cerebro que cuenta con millones de procesadores, también conocidos como neuronas.

Tabla 1.

Cerebro humano vs computador

Características	Cerebro humano	Computador
Velocidad de procesamiento	Entre 10^{-3} y 10^{-2} segundos	Entre 10^{-8} y 10^{-9} segundos
Estilo de procesamiento	Paralelo	Secuencial
Numero de procesadores	Entre 10^{11} y 10^{14}	Pocos
Conexiones	10,000 por procesador	Pocos
Almacenamiento del conocimiento	Distribuido	En direcciones fijas
Tolerancia a fallos	Amplia	Poca o nula
Tipo de control del proceso	Autoorganizado	Centralizado
Consumo de energía para ejecutar una operación por segundo	10^{-16} Julios	10^{-6} Julios

Nota: Comparación entre el cerebro humano y una computadora. Elaboración propia, realizado con Visio.

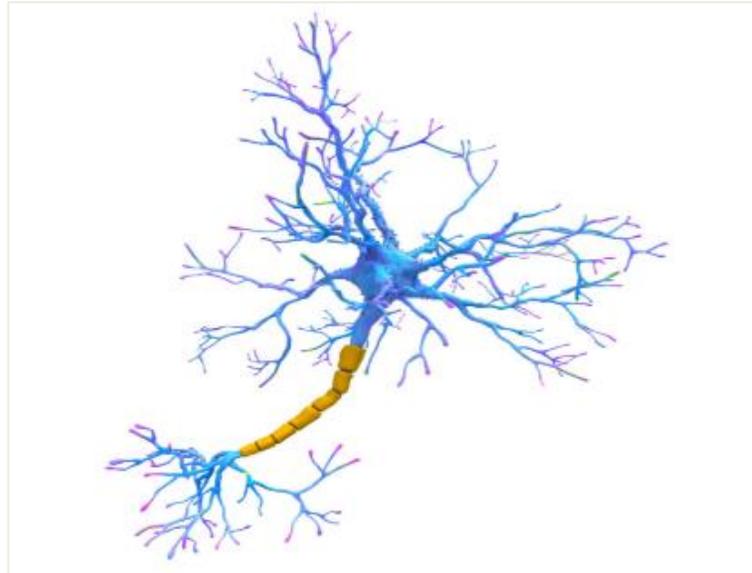
3.1. Redes neuronales biológicas

La neurona es el elemento fundamental de los sistemas neuronales en los cerebros humanos. Una neurona no es más que un tipo de célula que cuenta con algunos de los elementos típicos de una célula normal que por lo general se encuentran en el soma, pero cuenta con un axón y un árbol que tiene muchas ramificaciones que se componen de dendritas.

El axón también puede contar con ramificaciones en su extremo. No existe solo un tipo de neurona, estas pueden variar su forma dependiendo de su función, pero por lo general se componen de las partes antes descritas.

Figura 10.

Neurona biológica



Nota. Representación de una neurona biológica que muestra las partes más importantes para la comunicación en una red neuronal. Obtenida de P. Alfred (2018). *Neuron* [La neurona]. (<https://www.sciencephoto.com/media/854238/view/neuron-illustration>), consultado el 06 de julio de 2023. De dominio público.

En el córtex se aprecia la existencia de una organización neuronal horizontal en capas, que coexiste con una organización vertical en forma de columnas de neuronas. Así mismo, existen grupos neuronales, localizados en zonas específicas del cerebro, especializados en ciertas áreas (área visual, córtex censo-motor, área auditiva área olfativa, etc.); cada subgrupo incluye varios millones de neuronas, existiendo de que el procesamiento de la información involucra la actuación de conjuntos de varios subsistemas, que intercambian y comparten información entre sí. (López, 2008, p. 13)

Las neuronas son pequeños procesadores que funcionan a base de señales eléctricas y actividad bioquímica, estos pequeños procesadores se componen de:

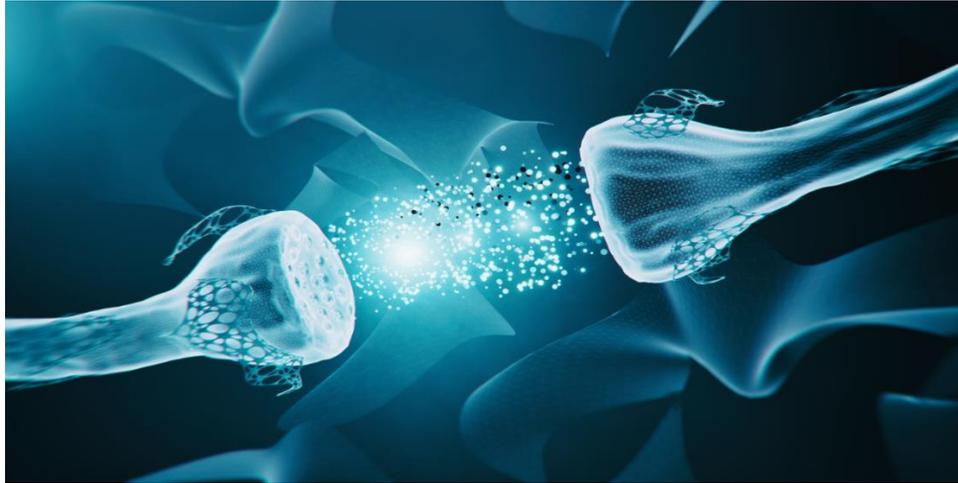
- Un canal de salida, el cual se compone por el axón el que con sus ramificaciones es capaz de conectarse con las dendritas de otras neuronas.
- Un procesador, el soma que se encarga de recolectar todos los inputs recibidos y emite las respectivas señales de salida.
- Un canal de entrada, que serían las ramificaciones de dendritas que se encargan de recibir alrededor de 10,000 señales de otras neuronas.

Sinapsis es el nombre que recibe la conexión del axón de una neurona con una dendrita de otra neurona. Que una sinapsis se vea excitada depende únicamente del aprendizaje que la neurona tuvo.

Al momento de que la sinapsis se ve excitada, se produce una comunicación curiosa, esta funciona mediante impulsos eléctricos que estimulan al axón de la neurona para que libere sustancias bioquímicas en la dendrita (*inputs*), las neuronas las captan y procesan para decidir si activarse o no.

Figura 11.

Sinapsis



Nota. Imagen ilustrativa que muestra la conexión de un axón terminal y una dendrita de dos neuronas realizando sinapsis. Obtenida de K. Feja (2020). *neuron synapse hologram* [Holograma de la neurona sinapsis]. (<https://www.istockphoto.com/es/foto/holograma-de-la-neuron-sinapsis-gm1223792768-359603324>), consultado el 06 de julio de 2023. De dominio público.

3.2. Redes neuronales artificiales

Al igual que en las redes neuronales biológicas, en las redes neuronales artificiales, las neuronas son el componente elemental que las compone. Estas neuronas se componen de una parte receptora de la información, una unidad de decisión y de las salidas.

En este tipo de redes neuronales se ven diferenciadas tres tipos de neuronas las cuales son:

- Neuronas de entrada: se encargan de recibir la información del exterior y comunicarlas a las neuronas ocultas.

- Neuronas ocultas: se encargan de realizar el procesamiento de la información recibida de las neuronas de entrada.
- Neuronas de salida: se encargan de enviar una señal directamente fuera del sistema, evitando que las neuronas ocultas tengan contacto con el entorno externo.

Las neuronas artificiales funcionan a base de fórmulas y operaciones matemáticas, con estas se determina si la neurona realizara su activación y enviara señales a otras neuronas. cada neurona cuenta con múltiples entradas a las cuales les corresponde un peso y este peso se multiplica por el valor de entrada que reciba. Para calcular si la neurona se activa o no se hace una sumatoria de todas las operaciones en las entradas y esto determina el nivel de excitación o activación de la neurona. Una representación vectorial del funcionamiento básico de una entrada en una neurona se muestra en la ecuación. Normalmente la señal de salida NET suele ser procesada por una función de activación F para producir la señal de salida de la neurona OUT. La función F puede ser una función lineal, o una función umbral o una función no lineal que simula con mayor exactitud las características de transferencia no lineales de las neuronas biológicas. (Basogain, 2008, p. 14)

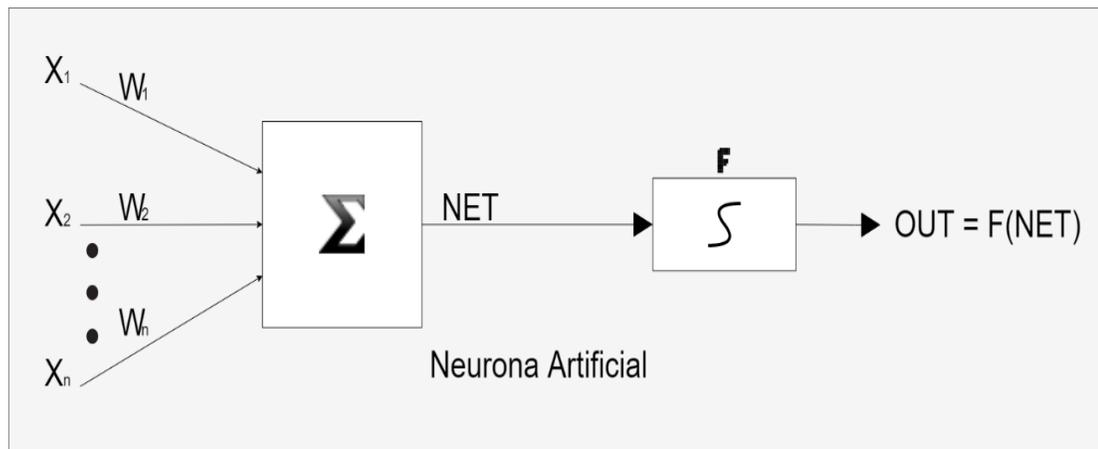
$NET = X*W$ es la función empleada para la activación de las redes neuronales es en donde NET es el indicador para validar si la neurona se enciende o no, X el vector de entrada y W el vector de pesos.

Este tipo de neurona artificial ignora muchas de las características de las neuronas biológicas, pero a pesar de las limitaciones de las neuronas, las redes

neuronales artificiales llegan a presentar similitudes con las redes neuronales biológicas.

Figura 12.

Neurona artificial



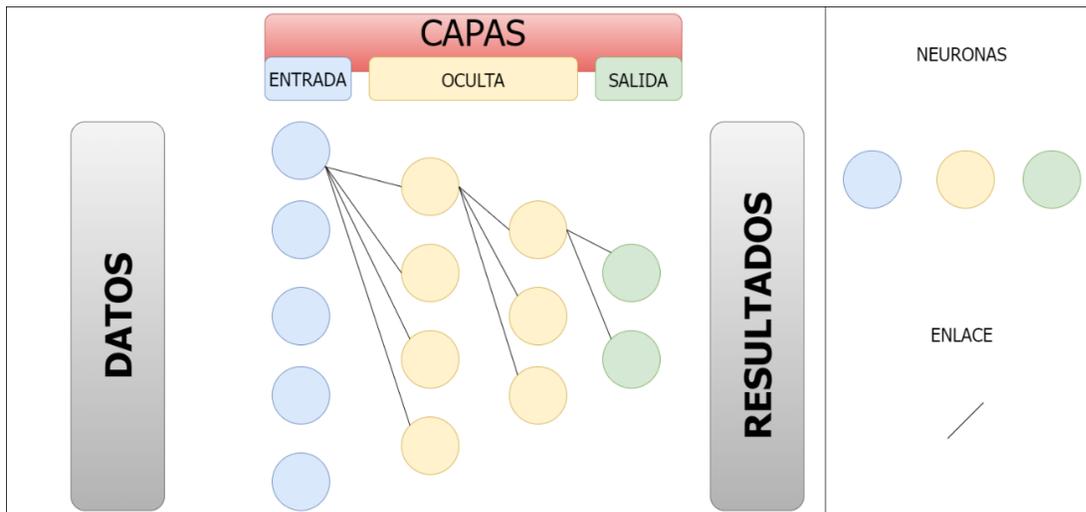
Nota. Diagrama de una neurona artificial. Elaboración propia, realizado con Drwa.io.

Las redes neuronales suelen constar de tres capas, y estas son: capa de entrada, capa oculta y capa de salida. La capa de entrada es la encargada de recibir toda la información para luego mandarla a la siguiente capa, que es la capa oculta; la capa oculta puede constar de múltiples capas y la tarea de estas es realizar todo el aprendizaje correspondiente; la capa de salida únicamente recibe la información de la capa oculta para luego elegir la posible mejor solución.

Las conexiones entre las neuronas se llevan a cabo de esta forma: todas las neuronas de una capa de entrada se conectan con todas las neuronas de la capa oculta, y la capa oculta se conecta con la capa de salida.

Figura 13.

Red neuronal artificial



Nota. Diagrama de una red neuronal artificial. Elaboración propia, realizado con Drwa.io.

Las redes neuronales son herramientas que se pueden aplicar en una gran cantidad de campos gracias a su capacidad para aprender. Las redes neuronales tienen sus variaciones, estas variaciones se adaptan mejor a ciertos tipos de tareas, y estas son: monocapa, perceptrón multicapa, convolucionales, recurrentes y retroalimentación.

3.3. Redes neuronales convolucionales

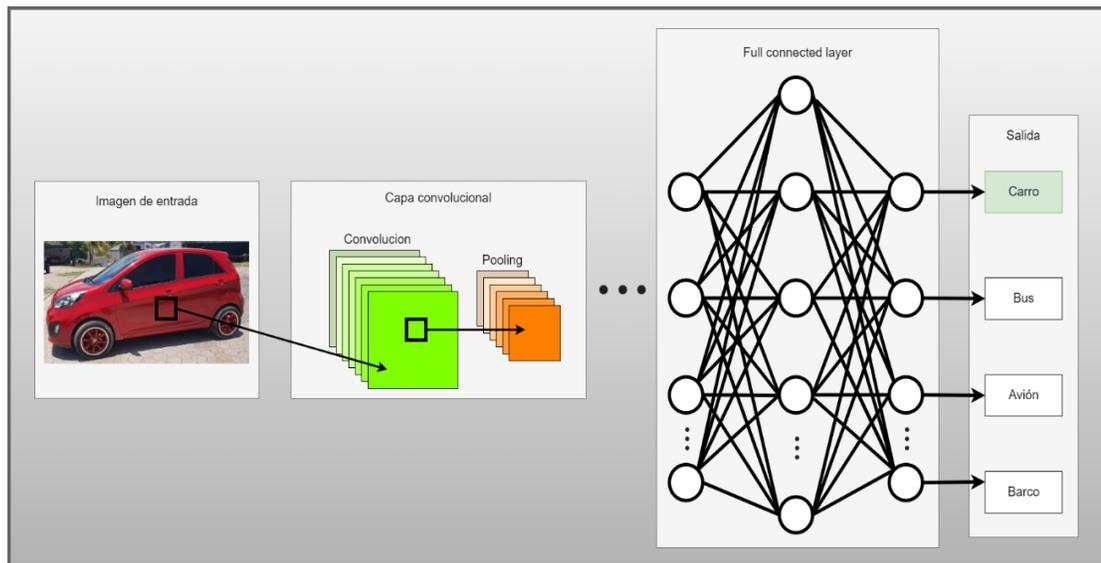
Las redes neuronales convolucionales aparecen debido a la exploración de capas de procesamiento no lineal con la que se pretende la extracción de características, transformación, análisis de patrones y la clasificación de los mismos.

Las redes neuronales convolucionales son las que han demostrado ser las idóneas para el reconocimiento de imágenes, ya que gracias a las convoluciones que se van aplicando a la imagen se obtiene un banco de datos con los que es más simple la detección de patrones.

La arquitectura de este tipo de redes permite que la información fluya únicamente en una dirección, y es desde las entradas a las salidas. Esta arquitectura consiste en las capas normales que posee una red neuronal, y se agregan otras capas que son las capas de convoluciones y pooling. Las redes neuronales convolucionales únicamente se encargan de agregar una capa extra (capa convolucional) y se acopla a una red neuronal multicapa para que esta pueda procesar los datos.

Figura 14.

Red neuronal convolucional



Nota. Diagrama de una red neuronal convolucional. Elaboración propia, realizado con Drwa.io.

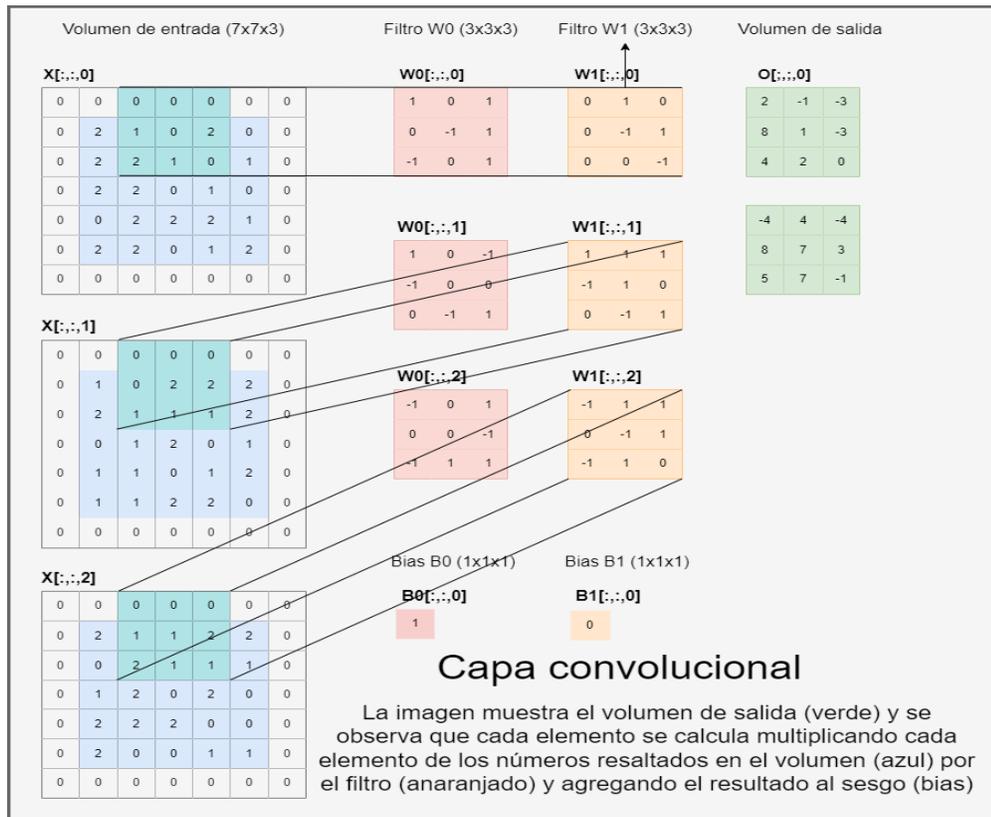
3.3.1. Capa convolucional

Los parámetros de la capa convolucional consisten en un conjunto de filtros aprensibles. Cada filtro es pequeño espacialmente (en alto y ancho), pero se extienden a través de la profundidad total del volumen de entrada. Por ejemplo, un filtro típico en la primera capa de una red neuronal convolucional podría ser de tamaño 5x5x3 (5 píxeles de ancho y alto, 3 por 3 canales de color).

Cada filtro se desliza (más preciso, se convoluciona) a través del alto y ancho de la entrada y se calcula el producto punto entre las entradas del filtro y la imagen de entrada en cualquier posición. Mientras desliza el filtro por el alto y ancho del volumen de entrada se produce un mapa de activación bidimensional que obtiene las respuestas del filtro en cada posición espacial. Intuitivamente, la red aprenderá los filtros que se activan cuando ven algún tipo de característica visual como el borde de alguna orientación o una mancha de algún color en la primera capa o eventualmente patrones completos como un panal de abeja o algo parecido a una llanta en capas más altas de la red. Si tenemos un conjunto de filtros en cada capa convolucional (*Conv layer*) por ejemplo, 12 filtros, cada uno de estos producirá un mapa de activación bidimensional. Estos mapas de activación se apilan a lo largo de la dimensión de profundidad y producirá un volumen de salida. (Contreras, 2018, p. 19)

Figura 15.

Capa convolucional



Nota. Muestra la forma en la que la capa convolucional aplica las convoluciones. Elaboración propia, realizado con Drwa.io.

La capa convolucional funciona con tres hiperparametros para el control del volumen de salida y estos son: profundidad, *stride* y *zero-passing*.

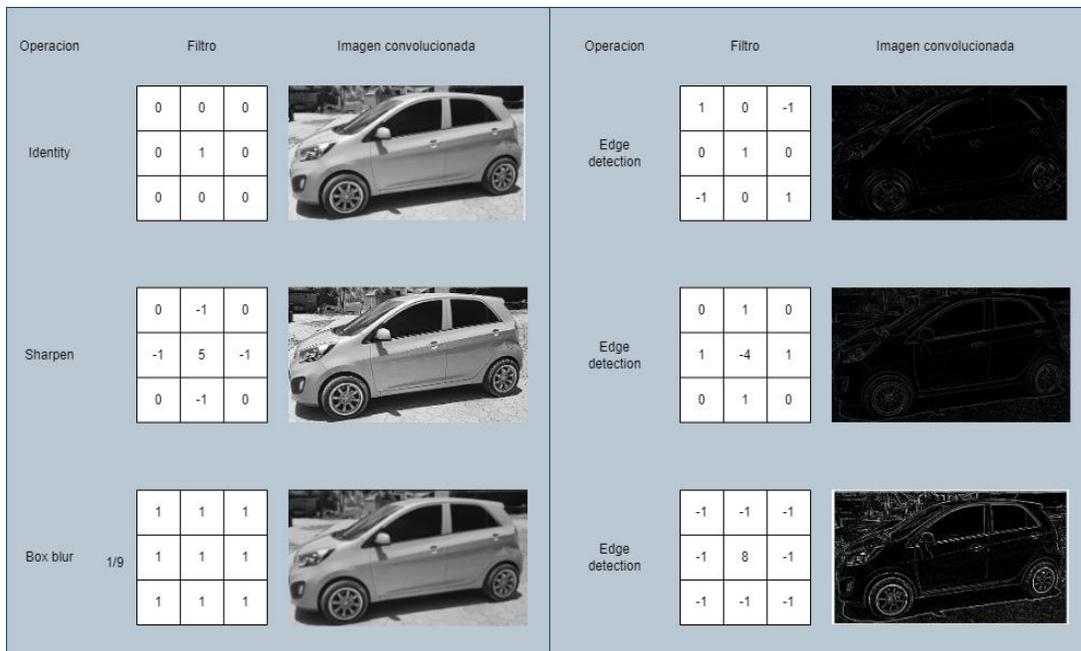
- Profundidad: corresponde al número de filtros que se quieren aplicar, cada filtro se encarga de buscar algo diferente en la entrada.

- Stride: este parámetro indica los campos que se desean que se mueva el filtro. Si se selecciona 1 el filtro se mueve de uno en uno, si se selecciona 2 el filtro se mueve de dos en dos.
- Zero-padding: permite el cambio en el volumen espacial del volumen de salida.

En la siguiente figura se pueden apreciar los filtros que la capa convolucional puede aplicar. Mediante estos filtros es posible el reconocimiento de bordes, colores, formas, entre otros.

Figura 16.

Convoluciones



Nota. Diferentes tipos de convoluciones. Elaboración propia, realizado con Drwa.io.

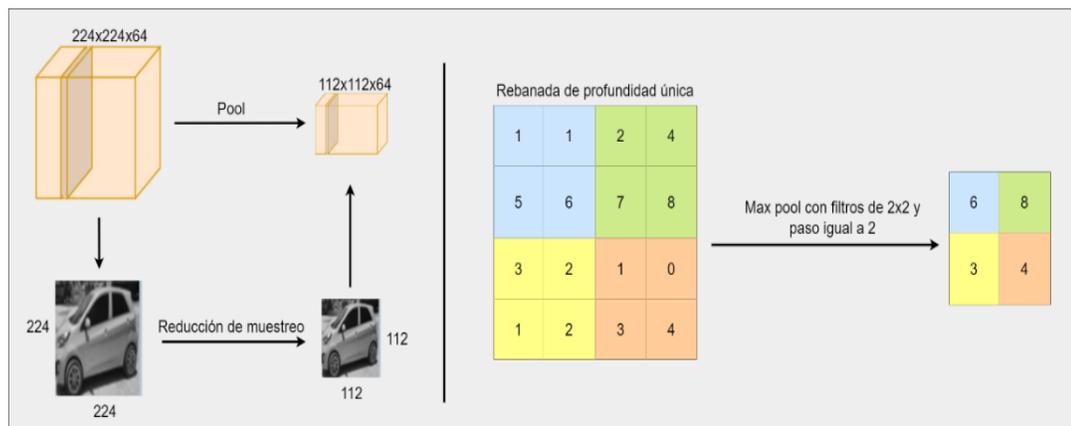
Cada red neuronal decide qué filtros son los que se aplicarán, debido a que los filtros pueden ser variables. Esto se debe a que no siempre los mismos filtros funcionarán para todas las cosas, por lo que, para evitar el menor error posible, cada red neuronal decide qué filtros son los que mejores se adaptan

3.3.2. Capa Poling

La capa *pooling* se suele agregar en el medio de las capas convolucionales. La función principal de esta capa es reducir el tamaño de la imagen, esto disminuye la cantidad de cálculos que se deben realizar y, por lo tanto, se reduce el *overfitting*. De esta forma, se logra una mejora en la detección de las características de la imagen, sin importar en qué lugar de la imagen se encuentren. Por lo general, para realizar la reducción se aplican filtros de tamaño 2x2 y un *stride* de 2. Para aplicar el filtro, se utiliza la operación MAX, la cual extrae el mayor de los 4 números de la matriz de 2x2.

Figura 17.

Pooling



Nota. Muestra el proceso pooling. Elaboración propia, realizado con Drwa.io.

3.3.3. Función de activación

Las funciones de activación son un componente fundamental en el diseño y entrenamiento de redes neuronales. Una función de activación se utiliza para determinar la salida de una neurona en función de su entrada y de los pesos asociados a las conexiones sinápticas.

Una de las funciones de activación más utilizadas es la función sigmoidea. Esta función toma un valor de entrada y lo transforma en un valor entre 0 y 1, lo que la hace útil para modelar la probabilidad de que una neurona se active. Sin embargo, la función sigmoidea tiene una limitación importante: satura a medida que el valor de entrada se acerca a cero o a uno, lo que puede dificultar el entrenamiento de redes neuronales profundas.

Otra función de activación común es la función ReLU. Esta función es muy simple: toma un valor de entrada y devuelve el mismo valor si es mayor que cero, y cero si es menor que cero. La función ReLU tiene la ventaja de ser muy eficiente computacionalmente y de evitar la saturación, lo que la hace adecuada para redes neuronales profundas.

Además de estas funciones, existen otras opciones como la función tanh y la función softmax, que se utilizan en redes neuronales para clasificación multiclase.

La elección de una función de activación depende en gran medida del tipo de problema que se está intentando resolver y de la arquitectura de la red neuronal en sí. Las funciones de activación influyen en la capacidad de una red neuronal para aprender patrones complejos y para generalizar a datos no vistos previamente. Por lo tanto, es importante elegir la función de activación adecuada

para cada tarea y ajustar sus parámetros para obtener el mejor rendimiento posible.

En conclusión, las funciones de activación son un elemento crucial en el diseño y entrenamiento de redes neuronales. Las funciones más comunes son la sigmoidea y la ReLU, cada una con sus ventajas y desventajas. La elección de la función de activación correcta puede mejorar significativamente el rendimiento de una red neuronal y hacer que sea más eficaz para resolver una amplia gama de problemas.

4. DESARROLLO DE APLICACIÓN WEB

Las aplicaciones web constan de dos partes muy importantes para su funcionamiento, y estas son el *frontend* y el *backend*; el *frontend* es la parte visual de la aplicación, es la parte que permite la interacción con el usuario, esta consta de: botones, cajas de texto, etiquetas, entre otros; el *backend* es el que se encarga del procesamiento de la información, esta parte es con la que se comunica el *frontend* ya sea enviando o pidiendo información.

El uso de aplicaciones web se ha vuelto popular en los últimos años gracias a todas las ventajas que ésta brinda, y proporciona un fácil acceso a las personas mediante el uso de internet y no requieren de instalación.

El desarrollo de estas aplicaciones se ha popularizado debido a las herramientas que facilitan en gran medida su desarrollo, algunas de las herramientas más populares para el desarrollo del *frontend* en aplicaciones web son: Angular, React y Vue.

Cada herramienta cuenta con su propio enfoque para el desarrollo y cuentan con amplia documentación que facilita su uso. Y en el caso del *backend*, las herramientas más populares son: Node.js, .NET y Java. Estas herramientas permiten el trabajo asíncrono, que permite un buen rendimiento.

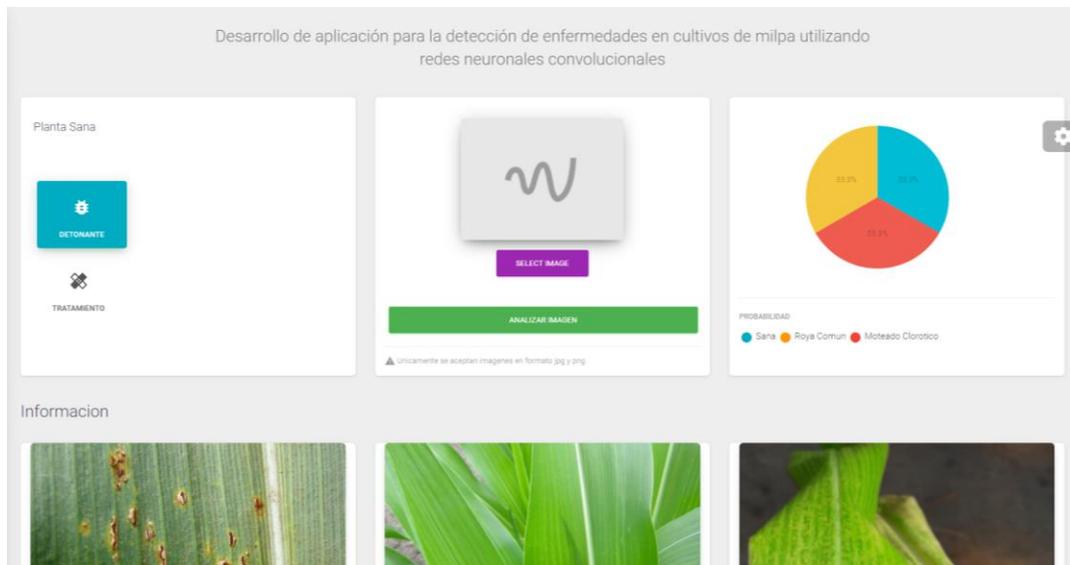
4.1. Desarrollo del Frontend

Para el desarrollo del *frontend* se utilizó React y también se utilizó una plantilla que cuenta con componentes como botones, tarjetas, gráficos, entre otros.

La interfaz gráfica cuenta con posibles indicaciones de los detonantes, tratamientos, una descripción de las enfermedades, una gráfica con los resultados de la red neuronal y también cuenta con una sección en la que se pueden subir imágenes para realizar el respectivo análisis.

Figura 18.

Aplicación web



Nota. Pantalla de inicio de la aplicación desarrollada. Elaboración propia, realizado con Windows 11.

4.2. Recopilación de imágenes

Para concluir con el desarrollo del *backend*, es necesario entrenar a la red neuronal convolucional y para realizar esta tarea, es necesario tener un banco de datos con el que se pueda realizar el entrenamiento. La recolección de imágenes se divide en tres partes y estas son: realizar búsquedas de imágenes de las enfermedades del maíz correspondientes en internet, consultas a distintas instituciones por una base de datos de enfermedades en plantas del maíz y la toma de fotografías a cultivos de milpa.

El método para realizar la búsqueda de imágenes en internet es realizar la búsqueda en diversos idiomas y búsquedas mediante imágenes similares. Este tipo de búsqueda tiende a ser tediosa y conlleva mucho tiempo realizarla. La cantidad de imágenes recopiladas mediante este método son de 180, distribuidas en las enfermedades de la roya común, moteado clorótico y la planta sana. Debido a que la enfermedad de roya común, es una de las enfermedades más comunes de la roya del maíz, es de la que se logró recopilar más imágenes; seguida del moteado clorótico y, al final, la menor cantidad de imágenes encontrada en internet fue de la planta sana.

Para la consulta a las instituciones se contactó a SAGARPA, MAGA y a facultades de agronomía de varias universidades. Estas instituciones no contaban con un banco de datos con las enfermedades del maíz que se necesitan para el entrenamiento de la red neuronal.

Para la toma de fotografías a las plantas del maíz, se procedió a visitar diversas propiedades en las que se estuviera sembrando maíz y pedir permiso para realizar las fotografías que ayudaron a complementar el banco de datos. Mediante este método únicamente se obtuvieron imágenes de plantas sanas; se

encontraron plantas enfermas, pero no con las enfermedades requeridas para realizar las capturas.

El banco de datos recolectado cuenta con un total de 125 imágenes de la roya común, 55 imágenes de moteado clorótico y 39 imágenes de la planta sana. El banco de datos está disponible en el enlace que aparece a continuación:

<https://drive.google.com/drive/folders/1-zBTDJIP4zWzEy8rPIMntU95MY-PdgX1?usp=sharing>.

4.3. Desarrollo del Backend

Debido a que la aplicación a desarrollar es una aplicación sencilla, y no recibirá un gran flujo de datos de entrada y de salida, por simplicidad se utilizará Python con la librería Flask.

4.3.1. Lenguaje Python para el desarrollo de redes neuronales

Python ha sido el lenguaje por excelencia que se ha utilizado durante los últimos años para la creación de aplicaciones que implementen inteligencia artificial, y este lenguaje permite una fácil implementación, lo cual ha llevado a que sea uno de los lenguajes con mayor crecimiento en la actualidad. Se caracteriza por ser eficiente al momento de trabajar con grandes volúmenes de datos y también cuenta con múltiples librerías (incluyendo librerías de IA que facilitan el desarrollo de aplicaciones).

Para el desarrollo de redes neuronales en Python se utilizan dos herramientas que brindan facilidades para la creación de las mismas, y estas son TensorFlow y Keras.

TensorFlow es una biblioteca de código abierto y se desarrolló para facilitar el aprendizaje automático. Esta biblioteca permite descifrar, detectar y corregir patrones.

Keras es un *framework* de alto nivel que se desarrolló en Python, y se dedica al aprendizaje profundo. Keras se desarrolló para la creación de redes neuronales, ya que con este framework es posible la creación de la red neuronal mediante bloques, y Keras es capaz de correr sobre TensorFlow, CNTK y Theano.

4.3.2. Programación de redes neuronales convolucionales

Para el desarrollo de la red neuronal convolucional se utilizó Python junto con la librería Keras que usa TensorFlow como motor. Las importaciones necesarias para el desarrollo de la red neuronal son las que se muestran en la figura 18, estas herramientas son de utilidad para el entrenamiento, la creación de la capa convolucional, la creación de las capas ocultas y la capa de salida.

Figura 19.

Librerías usadas en Python

A screenshot of a terminal window with a dark background and light blue text. The code shows imports for Keras models and layers, Keras preprocessing, TensorFlow, and NumPy. The terminal window has three colored window control buttons (red, yellow, green) at the top left.

```
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.preprocessing import image
from tensorflow import keras
import numpy as np
```

Nota. Librerías utilizadas en la red neuronal. Elaboración propia, realizado con carbon.now.sh.

El primer paso realizado para la creación de la red neuronal convolucional es crear la capa convolucional, esta se encarga de preparar la información de entrada para que sea posible la detección de patrones en las imágenes. En la primera línea se inicializa el modelo, luego se crea una capa convolucional 2D a la cual se le pasan como parámetros la cantidad de filtros que se desean aplicar a las imágenes de entrada, la función de activación y las dimensiones de las imágenes de entrada (únicamente es necesario aplicar en la primera capa convolucional). Luego se aplica una capa de *maxpooling* que se encarga de reducir las imágenes y se le asigna el tamaño con el que se desea realizar la reducción de la imagen.

Figura 20.

Creación de capas convolucionales y pooling en Python

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation = 'relu', input_shape = (128, 128, 3)))
model.add(MaxPooling2D(pool_size = (2,2)))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size = (2,2)))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
```

Nota. Capas convolucionales creadas en la red neuronal. Elaboración propia, realizado con carbon.now.sh.

El segundo paso consiste en crear la capa oculta, la cual se inicia agregando una capa *flatten* que se encarga de convertir los datos de salida de la capa convolucional en datos de entrada para las capas ocultas. Posteriormente,

se agregaron un total de tres capas ocultas que cuentan con una función de activación relu y las capas poseen 512, 512 y 256 nodos. Al final, se agregó la capa de salida, y cuenta con una función de activación softmax y consta de 3 nodos de salida. La capa identifica dos enfermedades y a la planta sana.

Figura 21.

Creación de capas ocultas en Python

```
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dense(512, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(3, activation='softmax'))
```

Nota. Capas ocultas creadas en la red neuronal. Elaboración propia, realizado con carbon.now.sh.

El último paso consiste únicamente en compilar la red neuronal y para eso se utilizó el método compile del modelo, al cual se le asignó el optimizador, la función de pérdida y las métricas que se tomarán en cuenta para la compilación. Con estos pasos se creó la red neuronal y se dejó lista para ser entrenada.

Figura 22.

Compilación de la red neuronal en Python

```
model.compile(  
    optimizer='adam',  
    loss='sparse_categorical_crossentropy',  
    metrics=['accuracy']  
)
```

Nota. Compilación de la red neuronal. Elaboración propia, realizado con carbon.now.sh.

4.3.3. Entrenamiento de red neuronal convolucional

El primer paso para el entrenamiento de las redes neuronales convolucionales consistió en importar ImageDataGenerator, lo cual permite asignar ciertas modificaciones que se realizarán a las imágenes de entrada para que el modelo cuente con una mayor cantidad de datos en el entrenamiento.

Figura 23.

Modificaciones a imágenes en Python

```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)
```

Nota. Modificación aplicada a las imágenes para agrandar el banco de información de entrada. Elaboración propia, realizado con carbon.now.sh.

El siguiente paso es cargar los conjuntos de imágenes que se utilizarán para el entrenamiento de la red neuronal. Se cargan dos conjuntos de imágenes: uno para el entrenamiento y otro para realizar pruebas y ajustar los pesos de la red neuronal.

Figura 24.

Carga del banco de datos en Python

```
batch_size = 5

training_dataset = train_datagen.flow_from_directory(
    r'path of images',
    target_size=(128,128),
    batch_size=batch_size,
    class_mode='binary'
)
testing_dataset = test_datagen.flow_from_directory(
    r'path of images',
    target_size=(128,128),
    batch_size=batch_size,
    class_mode='binary'
)
```

Nota. Carga del banco de imágenes. Elaboración propia, realizado con carbon.now.sh.

Como parte final del entrenamiento se procede a utilizar el método fit del modelo, lo cual inicia con el entrenamiento de la red neuronal. Para tener un entrenamiento apropiado, se debe asignar ciertos valores a la función que permitan tener el mejor entrenamiento posible, y para esto se asignan los siguientes datos al método. Luego de que finaliza el entrenamiento de la red neuronal, se procede a guardar el modelo entrenado para poder acceder a él posteriormente.

Figura 25.

Aplicación de modificaciones en Python

```
model.fit(  
    training_dataset,  
    steps_per_epoch=int(50/batch_size),  
    epochs=150,  
    validation_data=testing_dataset,  
    validation_steps=int(10/batch_size)  
)  
  
model.save("my_model")
```

Nota. Aplicación de las modificaciones a imágenes para aumentar el banco de información. Elaboración propia, realizado con carbon.now.sh.

Al finalizar el entrenamiento únicamente queda acceder a la red neuronal entrenada y posteriormente pasarle las imágenes para que la red neuronal dé la predicción. En la primera línea se carga el modelo generado en el entrenamiento, luego se carga la imagen deseada y se prepara para poder insertarla en el modelo y que este la acepte y muestre los resultados.

Figura 26.

Utilización de la red neuronal en Python

```
model = keras.models.load_model("my_model")

test_image = image.load_img('path image', target_size = (128,128))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = model.predict(test_image)
```

Nota. Forma de uso de la red neuronal creada. Elaboración propia, realizado con carbon.now.sh.

CONCLUSIONES

1. La precisión de los modelos de detección de enfermedades en plantas depende en gran medida de la calidad y cantidad de los datos de entrenamiento utilizados. Es importante recopilar conjuntos de datos grandes y bien etiquetados para asegurar que los modelos sean confiables, para el entrenamiento se logró recopilar un total de 219 imágenes, 55 de moteado clorótico, 125 de roya común y 39 de la planta sana por lo que obtuvo una red neuronal convolucional la cual no cuenta con un nivel de precisión muy alto.
2. Se desarrolló una aplicación web la que permite cargar imágenes para procesarlas y verificar el tipo de enfermedad Roya Común, Moteado Clorótico o que la planta se encuentre sana, debido a la cantidad reducida de imágenes que se tiene para el entrenamiento de la red neuronal no se logra obtener un porcentaje alto en el éxito de las detecciones
3. La inteligencia artificial puede ser una herramienta eficaz para la detección de enfermedades en plantas debido a su capacidad para procesar grandes cantidades de datos y detectar patrones que podrían no ser evidentes para los humanos.
4. Los algoritmos de aprendizaje automático pueden entrenarse con conjuntos de datos de imágenes de plantas infectadas y no infectadas para identificar patrones que indican la presencia de enfermedades. Estos modelos pueden utilizarse para detectar enfermedades en plantas en tiempo real.

RECOMENDACIONES

1. Elegir un conjunto de datos y técnicas de inteligencia artificial adecuadas puede ser crucial en el éxito de la tesis. Es importante seleccionar un conjunto de datos que sea representativo de las enfermedades que se están buscando detectar y elegir técnicas de inteligencia artificial que sean adecuadas para el problema.
2. Evaluar el modelo en diferentes conjuntos de datos para asegurarse de que sea capaz de detectar enfermedades en plantas de manera confiable y precisa.
3. Interpretar los resultados para obtener información valiosa sobre la detección de enfermedades en plantas una vez que se haya desarrollado el modelo y se hayan evaluado sus resultados.

REFERENCIAS

- Alfred. P. (2018). *Neuron* [La neurona]. Sciencephoto.
<https://www.sciencephoto.com/media/854238/view/neuron-illustration>
- Basogain, X. (2008). *Redes neuronales artificiales y sus aplicaciones*. Escuela Superior de Ingeniería de Bilbao.
https://ocw.ehu.eus/pluginfile.php/40137/mod_resource/content/1/edes_neuro/contenidos/pdf/libro-del-curso.pdf
- Bonilla, C. (2020). *Redes convolucionales*. [Tesis de licenciatura, Universidad de Sevilla]. Repositorio institucional.
- Contreras, O. (2018). *Desarrollo de una red neuronal convolucional para el procesamiento de imágenes placentarias*. [Tesis de licenciatura, Universidad Nacional Autónoma de México]. Repositorio institucional.
- Corelens (2022). *Raíz de la planta de maíz*. Canva.
<https://www.canva.com/photos/MAFU3l6JYmk-plant-with-bare-roots/>
- Feja, K. (2020). *Neuron synapse hologram* [Holograma de sinapsis de neuronas]. Istock.
<https://www.istockphoto.com/es/foto/holograma-de-la-neuron-sinapsis-gm1223792768-359603324>

Free Nature Stock (2022). *Campo de maíz verde*. Pexels.
<https://www.canva.com/photos/MADGyeynBQA-selective-focus-photography-of-green-corn-fields/>

Guillén, B. (septiembre de 2016). *El verdadero padre de la inteligencia artificial*.
<https://www.bbvaopenmind.com/tecnologia/inteligencia-artificial/el-verdadero-padre-de-la-inteligencia-artificial/>

Kodwaran, P. (septiembre de 2016). *Moteado clorótico*.
<https://plantwiseplusknowledgebank.org/cms/10.1079/pwkb.20167800947/asset/c9b397c0-cfe6-45a7-95fc-18fa4c63823e/assets/graphic/maize-lethal-necrosis-disease-global-1.jpg>

Lopez, F., Fernández, J. (2008). *Las redes neuronales artificiales*. Netbiblo.

Mali, M. (2016). *Maíz amarillo*. Pexels. <https://www.pexels.com/photo/yellow-corn-547263/>.

Meseguer, P. (19 de julio de 2012). *Turing y el ajedrez*. El País.
<https://blogs.elpais.com/turing/2012/07/turing-y-el-ajedrez.html>

Pioneer Argentina (2016). *Roya común*. Pionner
https://www.pioneer.com/ar/articulos/monitoreo_control_roya.html.

Ponce, J., Torres, A., Quezada, F., Sprock, A., Martínez, E., & Casali, A. (2014).
Inteligencia artificial. Proyecto LATIn.

Programa de maíz del CIMMYT. (2004). *Enfermedades del maíz*. CIMMYT.

<https://repository.cimmyt.org/xmlui/bitstream/handle/10883/715/25905.pdf>

Shulca, J. (2020). *Desarrollo de una red neuronal convolucional para la detección de fallos en maquinaria rotativa en tiempo real*. [Tesis de licenciatura, Universidad Técnica de Ambato]. Repositorio institucional.

Torres, J. (2020). *Python Deep learning* [Aprendizaje profundo de Python]. Marcombo.

Venini, J. (2020). *Tallo del maíz*. Elabcrural. <https://elabcrural.com/green-snap-o-quebrado-en-verde-del-tallo-del-maiz-de-que-se-trata/>

APÉNDICES

Apéndice 1.

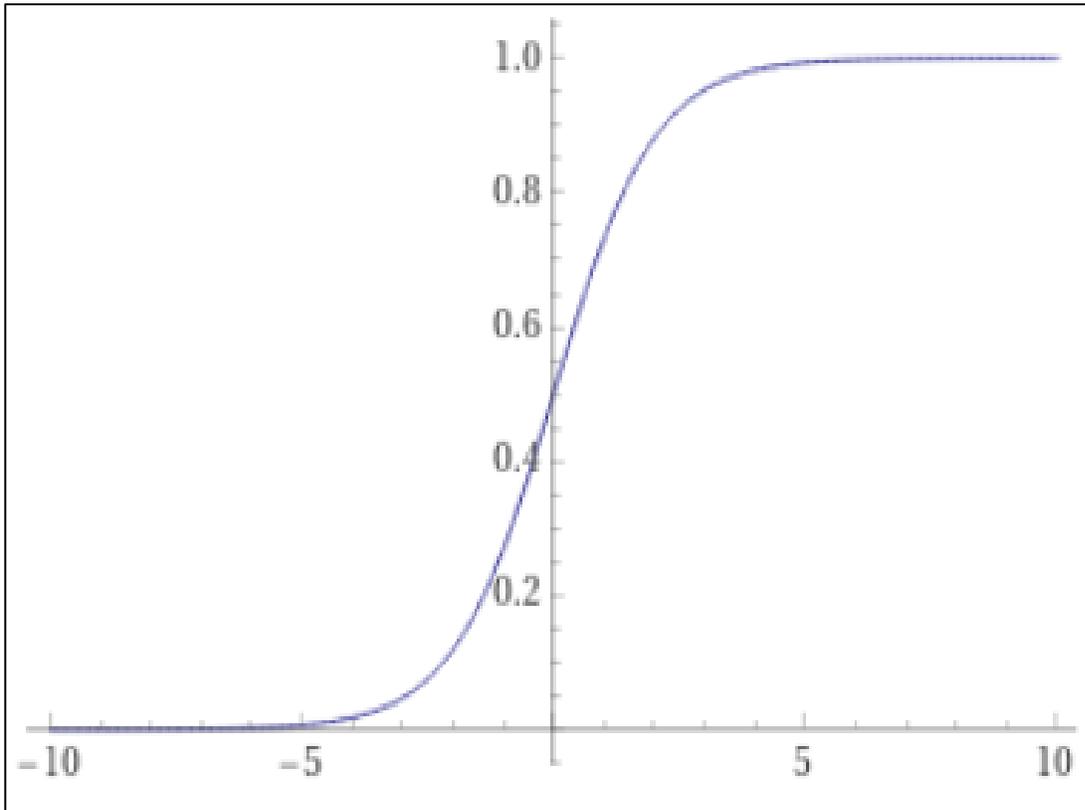
Tabla comparativa funciones de activación

Función de activación	Rango de salida	Función matemática
Sigmoidea	0 a 1	$f(X) = 1/(1 + e^{-X})$
ReLU	0 a infinito	$f(X) = \max(0, X)$
Tanh	-1 a 1	$f(X) = (2/(1 + e^{-2X}) - 1)$
Softmax	0 a 1	$f(X_i) = (e^{X_i})/(\sum_{j=0}^n e^{X_j})$

Nota. Tabla comparativa de las funciones de activación utilizadas en las neuronas artificiales.
Elaboración propia.

Apéndice 2.

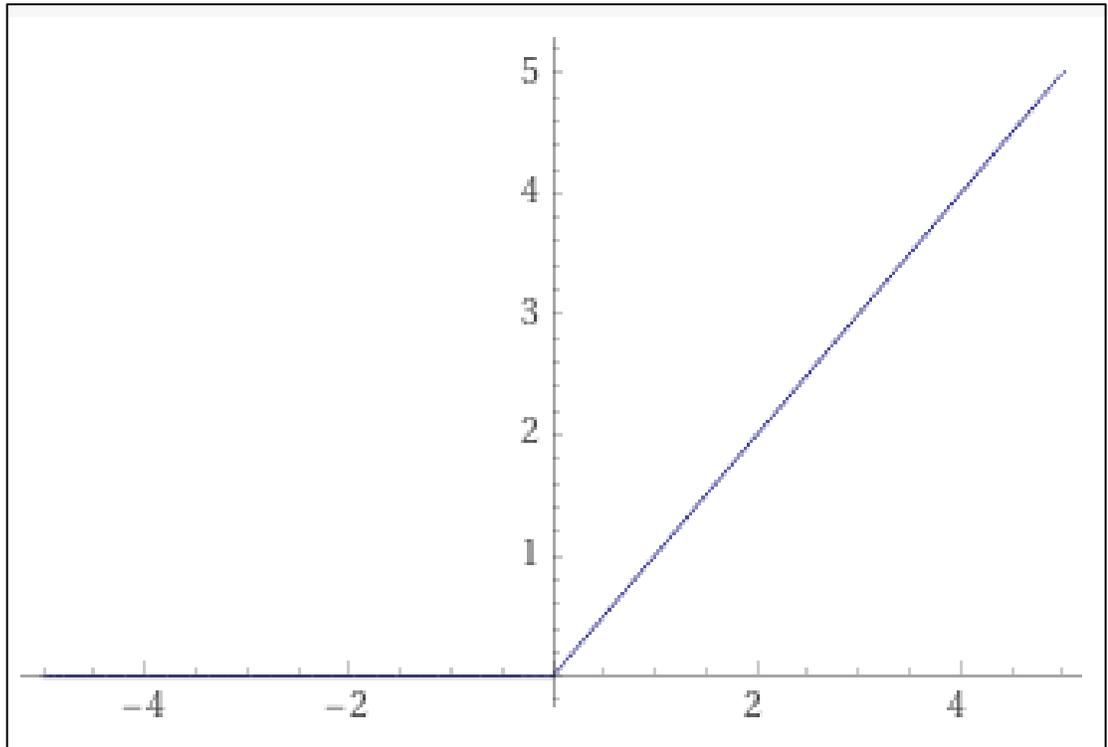
Función sigmoidea



Nota. Gráfica de la función sigmoidea utilizada para la activación en neuronas artificiales. Elaboración propia, realizado con WolframAlpha.

Apéndice 3.

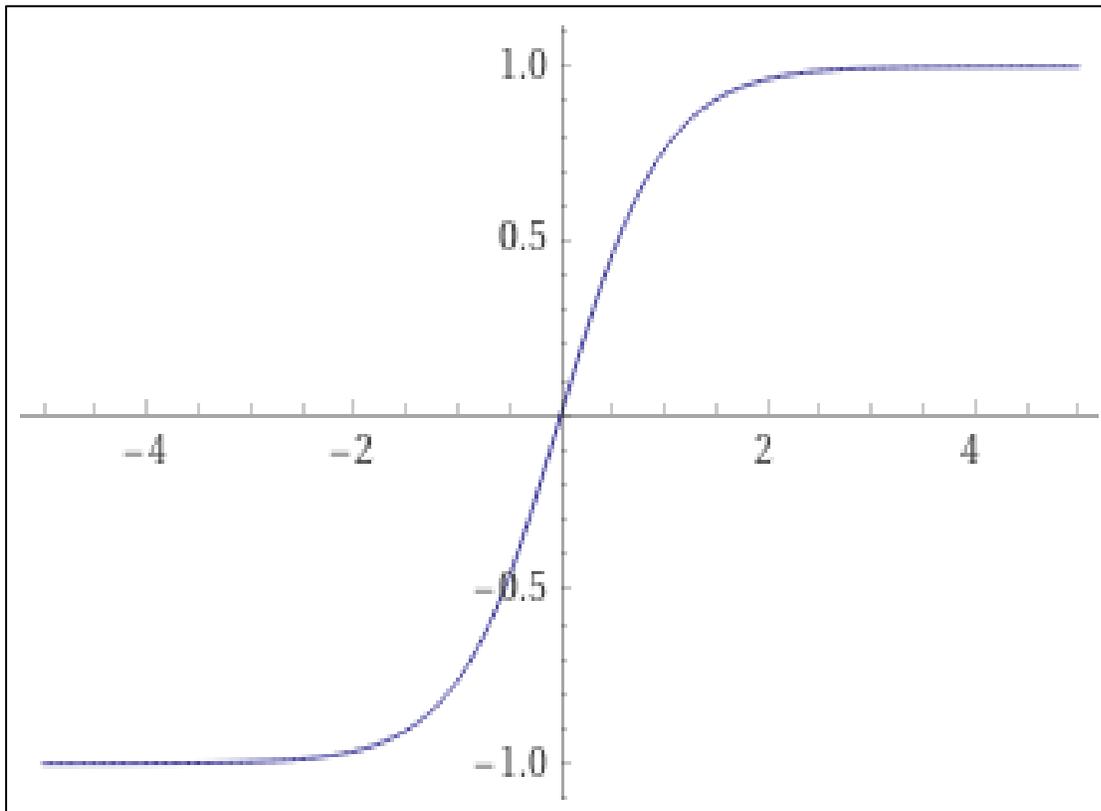
Función ReLu



Nota. Gráfica de la función ReLu utilizada para la activación en neuronas artificiales. Elaboración propia, realizado con WolframAlpha.

Apéndice 4.

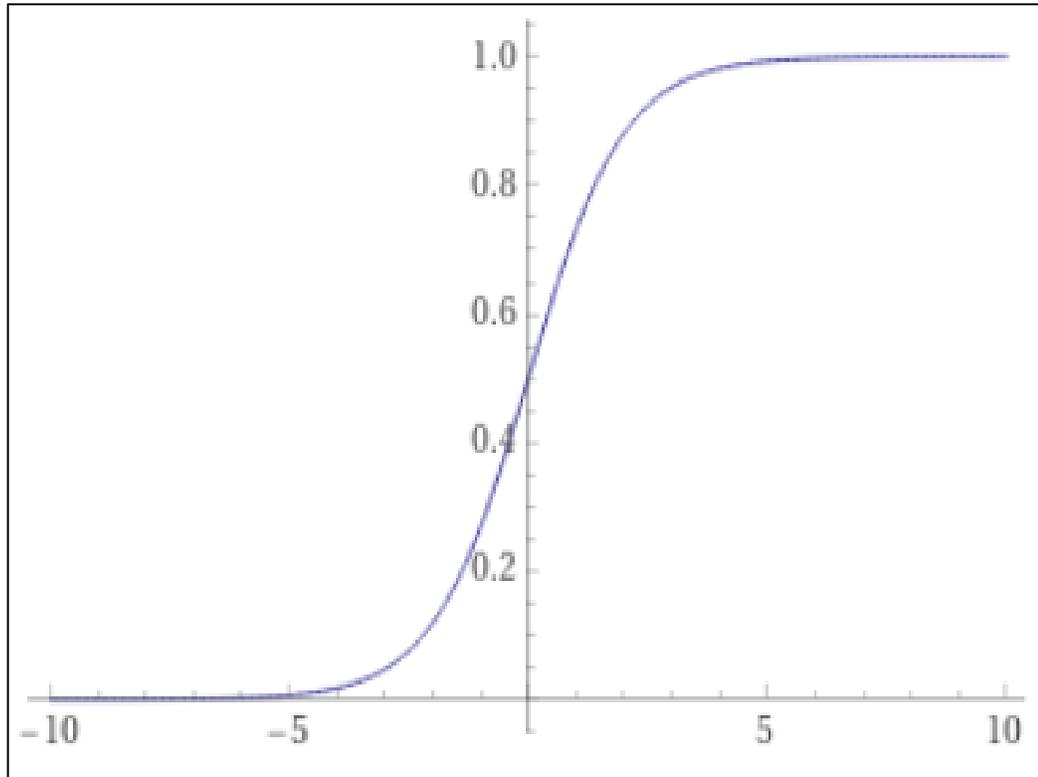
Función Tanh



Nota. Gráfica de la función Tanh utilizada para la activación en neuronas artificiales. Elaboración propia, realizado con WolframAlpha.

Apéndice 5.

Función softmax



Nota. Gráfica de la función softmax utilizada para la activación en neuronas artificiales. Elaboración propia, realizado con WolframAlpha.