



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA INFORMÁTICA
MENCIÓN SISTEMAS DE INFORMACIÓN

**Inteligencia Artificial aplicada a la agricultura de
precisión. Control de hongos en la planta de tomate**

**Artificial Intelligence applied to Precision Agriculture.
Control of fungi in the tomato plant**

Realizado por
Fabrice Fouquet Calderón

Tutorizado por
Francisco López Valverde

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE DE 2021

Fecha defensa: XXXX de octubre de 2021

Agradecimientos

En primer lugar, me gustaría agradecerles a mis padres todo lo que han hecho por mí, en todo momento habéis estado ahí confiando y apoyando como los que más, en los buenos momentos y en los no tan buenos. A mi Luna, mi hermana, gracias por estar siempre ahí como un equipo, un equipazo. Nos costó empezar y estimamos que fueran más años pero se luchó y se logró, lo logramos juntos los cuatro.

A Francisco, gracias por darme la oportunidad de trabajar con usted y confiar en mí para realizar este TFG. Tras haber sido alumno suyo supe que debería abordar este trabajo con usted y no me he equivocado. Después de todo, aunque se pensará que no se iba a llegar o que íbamos a estar super justos, hemos conseguido terminarlo sin ningún problema, con trabajo duro.

Para Sarvio, obrigado por se interessar sem realmente ter que fazê-lo e, acima de tudo, me ajudar na compreensão do conjunto de dados e do problema. Graças a vocês, consegui focalizar a direção do trabalho. Sem sua ajuda, não creio que teria sido possível. Muito obrigado, de fato.

A Juan Carlos y Alberto, ha sido un placer compartir estos cuatro años con ustedes. Remando juntos siempre en la misma dirección, no he podido tener mejores compañeros. Gracias.

A JJ. Gallardo, David y Ramón, gracias por estar siempre ahí en todo momento, es una alegría tener amigos con ustedes.

A mi familia, a compañeros de piso y carrera, a mis amigos y cada una de las personas que siempre ha estado ahí apoyándome en todos años y confiando en todo momento incluso cuando ni yo lo hacía, gracias.

Por último y especialmente a mi Andrea, gracias por mantener la cordura en mí. Aguantándome siempre incluso ni cuando yo mismo me aguanto. Me has entendido, me has ayudado y me has dado ese empujón que en ocasiones me ha hecho falta. Gracias por acompañarme en esta aventura y en las siguientes.

Resumen

Actualmente, el autoabastecimiento del sector alimentario en el planeta pelagra de forma temerosa debido al alto crecimiento de la población en el mundo. Este aumento de la población conlleva el aumento de la necesidad de productos agrícolas.

Las plagas y enfermedades que se producen en los cultivos son una de las principales amenazas que afectan a esta necesidad. Es por ello que la industria agrícola ya usa la automatización y precisión en sus prácticas para optimizar la producción. El uso de este tipo de técnicas, agricultura de precisión, junto con la Inteligencia Artificial ha hecho posible la predicción y reacción rápida ante situaciones impredecibles como esta.

Por lo tanto, este trabajo se centra en la creación de una herramienta basada en la rama de *Machine Learning* que nos permita realizar predicciones sobre la afectación del grado de severidad de un hongo en el cultivo del tomate.

El desarrollo de esta herramienta conlleva un preprocesamiento de un conjunto de datos que contiene determinados índices de vegetación de esta planta. Para su implementación se llevará a cabo una red neuronal que nos permita predecir dicho grado de severidad.

Los resultados obtenidos y las gráficas desarrolladas determinan el buen y eficiente funcionamiento de nuestro motor de inteligencia para realizar predicciones.

Palabras clave:

Inteligencia Artificial, Machine Learning, Agricultura de Precisión, Phytophthora Infestans, Tomate, Severidad

Abstract

The self-sufficiency of the world's food supply is currently under threat due to the high population growth in our planet. This population growth leads to an increasing need for agricultural products.

Crop pests and diseases are one of the main threats to this need. This is the reason why the agricultural industry is already using automation and precision in its practices to optimise production. The use of the precision agriculture techniques together with Artificial Intelligence has made it possible to predict and react quickly to above-mentioned unpredictable situations.

Therefore, this project focuses on the creation of a Machine Learning based tool that allows us to make predictions about the degree of severity of a fungus in tomato crops.

The development of this tool involves the pre-processing of a dataset containing certain vegetation indexes of this plant. For its implementation, a neural network will be used to predict the degree of severity of the fungus.

The results and the graphs obtained confirm the good and efficient functioning of our intelligence engine to make predictions.

Keywords:

Artificial Intelligence, Machine Learning, Precision Agriculture, Phytophthora Infestans, Tomato, Severity

Índice

Agradecimientos	1
Resumen	1
Abstract	1
Índice	1
Introducción.....	1
1.1 Motivación	2
1.2 Objetivos	3
1.3 Estructura de la memoria	3
Estado del arte	5
2.1 Tecnología en la agricultura de precisión	5
2.2 La IA en la agricultura de precisión	12
2.3 Índices de vegetación hiperespectral en referencia al cultivo del tomate en la agricultura de precisión.....	19
2.4 Hongos en el cultivo del tomate	21
Metodología.....	25
Análisis de requisitos.....	31
4.1 Requisitos funcionales.....	31
4.2 Requisitos no funcionales.....	33
Diseño.....	35
5.1 Arquitectura del Software	35
5.2 Tecnologías.....	37
Implementación	41
6.1 Extracción de características y carga de datos	42
6.2 Elección del modelo de aprendizaje	44
6.3 Optimización de parámetros	48
6.4 Proceso de aprendizaje y entrenamiento	50
6.5 Evaluación del proceso.....	54
6.6 Validación del proceso	56
Resultados	59
7.1 Resultados fase de entrenamiento	59
7.2 Resultados fase de test	61
7.3 Resultados fase de validación.....	63
Conclusiones y líneas futuras	67
8.1 Conclusiones.....	67
8.2 Líneas futuras	68

Referencias..... 71
 Bibliografía71
 Webgrafía75
Manual de Instalación..... 77
 Requerimientos:.....77
 Instalación:78

1

Introducción

De una forma cada vez más inmensurable se estima que la población del planeta irá siempre a más, estimando que para 2050 sea de un total de 9.3 millones de habitantes, lo que conlleva que la producción de alimentos deberá duplicarse en un pequeño período de tiempo para poder llegar a satisfacer las necesidades de la población.

Este aumento de la producción puede verse afectado por múltiples amenazas, entre ellas las plagas y enfermedades pueden llegar a ser las más devastadoras. Los brotes y recrudescimientos pueden llegar a provocar grandes pérdidas en los cultivos llegando a poner en peligro los medios de vida de los agricultores y la seguridad alimentaria y nutricional de la población mundial.

Estas amenazas pueden llegar a tener una propagación muy rápida. Entre los diferentes tipos podemos destacar el comercio o los movimientos migratorios de las personas, los factores ambientales y los insectos u otros patógenos, como es nuestro caso.

Aquí es donde entra en juego la agricultura de precisión, debido a que esta será una gran forma para poder llegar a agilizar la sostenibilidad y seguridad del planeta frente a estos cambios que irán surgiendo.

Como bien sabemos, la inteligencia artificial (IA) y el aprendizaje automático facilitan múltiples facetas de la vida cotidiana y es poco inusual aquellos sectores empresariales y domésticos dónde no se utilice.

Aunque la agricultura se considere como una de las industrias más antiguas, requiere un enfoque más inteligente y eficaz para poder llegar a cubrir estas necesidades. Gracias a las tecnologías de IA es posible mejorar múltiples aspectos de la agricultura como puede ser la optimización de la monitorización del estado del suelo, llevar un control más efectivo de las plagas, utilizar índices de vegetación para mejorar diversos aspectos, entre otros.

Por ello, la dupla de la agricultura de precisión e IA es cada vez más sonada. La industria agrícola ya usa la automatización y la precisión en sus prácticas con el objetivo principal de mejorar la producción y la vida de los cultivos.

1.1 Motivación

El concepto fundamental sobre el que se basa la agricultura de precisión es saber aplicar la cantidad correcta de insumos, en el momento adecuado y en el lugar exacto. De forma más específica la agricultura de precisión es la estrategia que usa la tecnología de información para adecuar el manejo de los suelos y cultivos a la variabilidad presente (García & Flego, 2007). Es decir, se entiende como la aplicación a tareas agrícolas de las nuevas tecnologías de información con el objetivo principal de mejorar la productividad, la vida de los cultivos y disminuir el impacto medioambiental.

De forma cada vez más sonada, las aplicaciones de procesamiento de imágenes y *computer vision* han crecido en la agricultura debido a la reducción de los costos de los equipos, al gran aumento de la potencia computacional y al creciente interés en los métodos de evaluación de alimentos no destructivos (Mahajan et al., 2015).

Y es que cada vez es más común el uso de estas técnicas. En comparación con los métodos tradicionales basados en el trabajo manual estas técnicas presentan numerosas ventajas, sin embargo, aún existen muchos desafíos por abordar y superar (Barbedo, 2016).

Las plagas y enfermedades de las plantas son una de las amenazas principales que pueden acabar de forma rápida y contundente con el duro trabajo de los agricultores llegando a generar pérdidas muy significativas en los rendimientos e ingresos, representando un peligro importante para la seguridad alimentaria.

Debido a estos factores nombrados anteriormente, a la vinculación de mi localidad natal con la agricultura y a la de mi propia familia hacen que el enfoque de este TFG sea utilizar los beneficios y funcionalidades de la Inteligencia artificial. El objetivo principal es crear una

herramienta que por medio de algoritmos de *Machine Learning* tenga la capacidad de utilizar y analizar un conjunto de datos pertenecientes al cultivo del tomate. Además de permitirnos predecir el grado de severidad de un hongo tan trascendente como es el protista fungoide *Phytophthora Infestans*.

1.2 Objetivos

El principal objetivo de este trabajo es desarrollar un motor de Inteligencia Artificial que implemente algoritmos de Machine Learning basado en el lenguaje de programación de Python con el fin de poder predecir el nivel de severidad de un hongo en el cultivo del tomate y a partir de ahí ser capaz de elaborar informes y conclusiones.

Para el estudio y análisis del grado de severidad de este hongo se utilizará un conjunto de datos proporcionado por la Universidad de Vicosa (Brasil). Este conjunto contiene ciertos índices de vegetación extraídos de una plantación del cultivo del tomate tras la implantación de este hongo.

Con estos índices se pretenderá ser capaz de predecir el grado de severidad ya establecido por especialistas por medio del uso de imágenes.

Scrum, ha sido la metodología ágil utilizada para la organización y realización del proyecto. Esta metodología nos ofrece una gran capacidad de adaptación para poder corregir cualquier tipo de inconveniente que perjudique el desarrollo de nuestro proyecto.

El IDE de PyCharm es el entorno utilizado para el desarrollo del motor de inteligencia, para ello se utilizará la librería de *Machine Learning* Pytorch.

1.3 Estructura de la memoria

Este documento se estructura de la siguiente forma: En la introducción se muestra la motivación y objetivos de este trabajo. El segundo apartado trata sobre el estado del arte de la agricultura de precisión que hablará sobre la tecnología, la inteligencia artificial, índices de vegetación hiperespectral y plagas y enfermedades en referencia al cultivo del tomate en la agricultura de precisión. El tercer punto indicará la metodología aplicada al desarrollo del trabajo. El cuarto apartado contendrá la lista de requisitos funcionales y no funcionales que deberá de cumplir el código desarrollado y la herramienta implementada.

El diseño del trabajo consta de las tecnologías, aplicaciones y software utilizado y necesario para su desarrollo. En el apartado de implementación se explica el proceso completo y finalmente se muestran las conclusiones junto a los resultados obtenidos de los apartados anteriores y se introducen posibles líneas futuras.

2

Estado del arte

2.1 Tecnología en la agricultura de precisión

Antes de desarrollar las páginas siguientes, dejaremos más o menos claro la técnica IoT. Dicha técnica combina los recursos existentes de Internet para obtener el control de todo tipo de dispositivos. Estos dispositivos pueden comunicarse entre sí y realizar una serie de actividades según lo dirigido o programado, de acuerdo con el diseño y con las capacidades funcionales de varios objetos.

Sin embargo, la era del IoT no tiene una gran prosperidad debido a que los investigadores están intentando incorporar los conceptos de Inteligencia artificial (IA) sobre estos dispositivos. El objetivo de esto es que puedan tomar las decisiones necesarias y actuar sin la intervención de los humanos.

IoT tiene un gran potencial social, medioambiental y un impacto económico en su adaptación. Han sido numerosos los campos en los que se han aplicado los conceptos de IoT como son los campos de movilidad, de atención médica y sanitaria, agricultura, procesamiento industrial entre otros. Al fin y al cabo, todas las actividades en estos campos están asociadas a los humanos de una forma o de otra. Pasando a ser un aspecto indispensable en la vida de estos debido al uso de estas aplicaciones y a los beneficios producidos.

La agricultura de precisión es una especie de enfoque de gestión agrícola que utiliza la tecnología de la información (TI) para garantizar que los cultivos y la superficie reciban

exactamente lo que necesitan para una salud y productividad óptimas. En este campo se han producido numerosos y enormes cambios en las técnicas y metodologías de realización de actividades agrícolas. Los investigadores han elaborado prácticas y teorías que incorporan dispositivos inteligentes para evaluar los parámetros que contribuyen a aspectos como el desarrollo de las plantas, beneficios de producción, etc.

A continuación, vamos a ver algunas de las principales contribuciones en este ámbito pertenecientes al siguiente artículo (Khanna & Kaur, 2019).

En el trabajo de Zhao et al. (2010) se proyectó los beneficios de realizar actividades agrícolas dentro de una tecnología de invernadero. Se propuso un marco que combinaba el Sistema de Monitoreo Remoto (RMS) con Internet y comunicaciones inalámbricas. Este marco dio como resultado un buen crecimiento y una buena salud de las plantas cultivadas en invernadero. Las grandes variaciones de temperatura ambiental y factores de humedad pueden provocar un menor rendimiento en la precisión de monitoreo y control.

El estudio de Bo & Wang (2011) propuso el uso de técnicas de Cloud Computing y la importancia del significado de IoT. Este estudio planteó la colaboración de dos técnicas con el objetivo de analizar la viabilidad y las perspectivas de futuro en la combinación entre ambas. Los experimentos llevados a cabo demostraron que la combinación de estas dos tecnologías podría resolver problemas con bastante éxito y eliminar los cuellos de botella del desarrollo.

La aplicación de IOT basada en Cloud Computing no se ha generalizado completamente y todavía existen algunos problemas por resolver. Entre ellos podemos destacar: el costo de escala de las instalaciones de IOT, los precios de dispositivos de detección y la dificultad de implementación; respecto a seguridad y riesgo, aunque esta aplicación haya logrado eficiencia y flexibilidad, todavía existen muchas incertidumbres en la seguridad humana y de la información.

En un estudio se presentó un marco vanguardista que destacó los impulsores tecnológicos clave de IoT. También visualizó las ayudas que podrían obtener los agricultores de las aplicaciones IoT para poder adquirir información para la entrega de cultivos dentro de pequeñas regiones (Bandyopadhyay & Sen, 2011).

En esta aplicación se observa que existe una necesidad urgente de un trabajo significativo en el área de gobernanza de IoT, debido a que sin un enfoque estandarizado se producirá una proliferación de arquitecturas, esquemas de identificación, protocolos, etc. Esta

proliferación conducirá inevitablemente a una fragmentación de IoT, lo que podría obstaculizar su popularidad y convertirse en un obstáculo importante en su implementación.

El trabajo realizado por Tong-Ke (2013) propuso el uso de conceptos IoT y técnicas de computación en la nube para realizar actividades agrícolas, tecnologías SOA y RFID también se incorporaron al estudio.

En esta aplicación se tuvieron que integrar múltiples aspectos de seguridad y riesgo debido a que los drones deben estar integrados con software programable de alta eficiencia que ayude a hacer todas las actividades necesarias.

Las redes de sensores inalámbricas (WSN) son aquellas que están formadas por dispositivos autónomos, distribuidos en un área de interés y cuyo objetivo es monitorizar parámetros físicos o ambientales.

El artículo de Ojha et al. (2015) destacó las posibles aplicaciones de WSN junto con varios desafíos y problemas asociados en el despliegue de esta plataforma con el fin de mejorar la agricultura. El advenimiento de las técnicas de WSN produjo un impulso en la investigación del dominio agrícola.

Este tipo de aplicaciones de WSN presentan soluciones de bajo coste con características como operación autónoma y bajo rendimiento. Estas se ven un poco más limitadas para superar los problemas en los países de ingresos bajos y medianos, debido a que se requiere una planificación más exhaustiva para asegurar el éxito.

En la investigación realizada por Bauer et al. (2016) se estudió el alcance de la implementación de WSN para derivar información vital sobre los cultivos, ya que es un buen indicador para aquellos procesos que limitan el rendimiento. Este estudio describe el uso de sensores de radiación fotosintéticamente activa (PAR) que mejora en gran porcentaje el rendimiento potencial de la tecnología WSN para la evaluación del índice de área foliar (LAI).

Como futuro trabajo se espera llegar a poder tener una mayor diversidad de cultivos. Para este propósito, se llevarán mediciones de LAI de forma más instructiva para superar el problema de saturación del enfoque LI-COR y se procesarán una mayor cantidad de datos de detección remota para validar ampliamente sus resultados.

En la enseñanza realizada por Ferrández-Pastor et al. (2016) se presentó una plataforma de interconexión de sensores actuadores de bajo coste, basados en IoT que incluían protocolos integrados M2M y una interfaz de persona a máquina. Aunque esta actividad estuvo desarrollada dentro de un invernadero, los resultados del experimento

mostraron que las tecnologías de Internet y los patrones de objetos inteligentes se pueden combinar para fomentar el desarrollo y mejorar aún más las actividades de agricultura de precisión.

Con el fin de lograr una gran polivalencia y diversidad de estas aplicaciones se espera que se utilicen algoritmos basados en sistemas expertos e inteligencia artificial (IA). Además de que se desarrollen nuevos cultivos agrícolas para integrar y probar nuevos servicios utilizando el método y plataforma desarrollados.

El estudio desarrollado por Jayaraman et al. (2016) enfatizó en una recopilación de datos agrícolas a través de varios dispositivos de IoT como WSN, estaciones meteorológicas conectadas a la red, cámaras y teléfonos inteligentes. Además, el estudio propuso una SmartFarmNet, una plataforma basada en IoT que era capaz de automatizar el proceso de recopilación de datos a partir de varios parámetros relacionados con la agricultura. El marco propuesto también fue capaz de correlacionar los datos y filtrar datos no válidos desde la perspectiva de evaluar el rendimiento de los cultivos y calcular más pronósticos de cultivo.

Los autores Paustian & Theuvsen (2017) destacaron en su trabajo la importancia de diversas técnicas disponibles en la agricultura de precisión. El estudio también tiende a comprender los aspectos relevantes de la adopción de la agricultura de precisión entre varios agricultores en Alemania, mediante diversas características y datos demográficos de los agricultores. El análisis de regresión muestra una influencia positiva en la adopción de la agricultura de precisión entre los agricultores. Los resultados del artículo de investigación proporcionaron múltiples puntos de partida para una mayor proliferación de la agricultura de precisión en varias direcciones.

Uno de los grandes inconvenientes de esta proliferación es la falta de capacitación de muchos agricultores a la hora de implantar estas tecnologías, ya sea por falta de conocimiento o por la falta de recursos.

El aprendizaje realizado por Balducci et al. (2018) expuso un estudio destinado a gestionar datos heterogéneos procedentes de varios conjuntos de datos en virtud de valores sensoriales. El estudio también describe cómo las empresas productivas están en una carrera por el aumento de los beneficios de estabilidad en virtud de la reducción de costes. Descubrir formas apropiadas de explotar los datos que se registran continuamente y se ponen a disposición puede ser la opción correcta para lograr esta meta. Además, sugirió el uso de

modelos de aprendizaje automático de regresión lineal y polinomial de redes neuronales para la toma de decisiones y el manejo de datos.

En este estudio se puede comprobar que para los casos en los que existen muy pocos modelos estadísticos de datos, como los lineales o polinomiales, las detecciones de monitoreo suelen ser defectuosas.

En la siguiente indagación destacaron el papel del uso de teléfonos inteligentes para acceder a información agrícola sobre diversos parámetros y sus ventajas en el campo de la agricultura. Se entrevistó a un total de 230 agricultores de la región y se realizaron cuestionarios estructurados primarios. El estudio destacó que la mayoría de los agricultores sugirieron el uso de teléfonos inteligentes para adquirir datos sobre las condiciones actuales de la granja. También se basaron en la adopción de técnicas agrícolas más nuevas que se muestran en los videos disponibles en el campo de la agricultura de precisión (Hamad et al., 2018).

Los problemas que se pueden presentar con el uso de teléfonos son la barrera del idioma, la batería del dispositivo, robos en las fincas, difusión de *fake news* o como bien se ha nombrado anteriormente, las dificultades para lidiar con la tecnología o la incapacidad de poder permitirse el uso de estas.

Independientemente del gran aumento de IoT y de sus numerosas aplicaciones a lo largo de estos años, existen cuestiones, desafíos e investigaciones futuras sobre este campo.

Dentro de los problemas ya existentes podemos ver que la subestimación del hardware debido que la mayoría contempla a IoT como una aplicación software, pero la parte del hardware es de igual importancia y está asociada a múltiples protocolos.

Los fallos de conectividad entre dispositivos son un gran problema para IoT por no poder transferir los datos correctamente, la conectividad juega un papel importante en la movilidad de los datos.

Por la gran cantidad de dispositivos conectados en la mayoría de las ocasiones, se presentan problemas a la hora de lograr una buena sincronización entre diferentes plataformas.

Las rápidas transformaciones de los sistemas de comunicación provocan que las innovaciones y aplicaciones sean reprimidas por los problemas de técnicas de comunicación como 5G, algunos de estos problemas son el manejo de múltiples entradas y salidas, la modulación de la capa física, etc.

Para los desafíos presentados, podemos destacar la comprensión completa de la arquitectura de IoT, en dónde volverse imperativo en una comunicación independiente de su naturaleza se ha convertido en el mayor desafío de SOA. Al igual que identificar la tecnología correcta para establecer la comunicación, en dónde mapear un identificador a una entidad sin una ambigüedad identificable se ha convertido en su principal desafío.

Por cómo ha avanzado el concepto de IoT, los recursos disponibles se han quedado cortos y los escenarios existentes dan paso a nuevas consultas que deben abordarse antes de pasar a una nueva dimensión. Entre este tipo de futuras investigaciones podemos destacar que tipo de modelo de negocio de IoT impulsará la próxima generación de negocios y comercio, y cuál será el próximo paso para identificar y manejar la gran cantidad de dispositivos que están surgiendo.

A lo largo del tiempo, está continua necesidad de información en tiempo real e in situ ha dado lugar a un nuevo tipo de IoT: Internet Of Underground Things (IOUT) (Vuran et al., 2018).

Esta modalidad representa aquellos dispositivos autónomos que recopilan información relevante sobre la Tierra y están interconectados con redes de comunicación que facilitan el envío de la información de los campos hacia los productores y los mecanismos de decisión. Este campo no solo proporciona información y capacidades de monitoreo desde el propio lugar, sino que puede llegar a una autonomía completa del campo cuando se conecta con la maquinaria existente.

Dentro de la arquitectura de IOUT podemos destacar las funcionalidades como la detención in situ donde se requieren sensores de humedad, temperatura y salinidad de la superficie para un conocimiento focalizado, y la comunicación inalámbrica en entornos con altas dificultades.

Estas soluciones deben tener la accesibilidad y capacidad de almacenar datos en el caso de comunicarse por vía aérea. Requieren la facultad de comunicarse a través del suelo y ajustar sus parámetros para adaptarse a los cambios dinámicos en vía subterránea, entre muchas otras.

Entre las principales funcionalidades de este campo, podemos destacar la detección en tiempo real y la conectividad inalámbrica. La cual ha sido el desencadenante en la adopción de la tecnología en la agricultura de precisión y ha permitido mejorar la eficiencia en producción y en prácticas agrícolas. Dentro de las tecnologías de detección podemos destacar

el uso de sensores de humedad del suelo para medir el contenido de agua y lograr una gestión del agua más efectiva o el uso de sensores para detectar la temperatura del suelo y aire, la dirección y velocidad del viento, etc. John Deere introdujo sensores para evaluar estos fenómenos en su solución comercial Field Connect.

Partiendo de esta detección, las comunicaciones inalámbricas entre equipos heterogéneos utilizados en la detección tienen un papel fundamental en la toma de decisiones en tiempo real. Las soluciones presentadas de conectividad para IOU se clasifican por comunicaciones en el campo y en la nube.

Como sistemas académicos, los IOU se pueden utilizar para determinar la cantidad de agua y fertilizante que se aplicará mediante un sistema de control de riego. Para ello, se desarrolló un banco de pruebas en un laboratorio de Nebraska, en este se instaló un avanzado sistema de riego de pivote central para investigar aspectos como la absorción de nutrientes y agua de los cultivos, la producción de los cultivos, el estrés hídrico, etc. Este banco es un sistema completamente funcional desarrollado para investigar las capacidades de comunicación y detección de IOU en un campo agrícola utilizando riego de pivote central, sensores y dispositivos de comunicación subterránea y aérea (Dong et al., 2013).

En este tipo de soluciones se ha podido comprobar la importancia de considerar los impactos de las propiedades del suelo en la calidad de la comunicación y a partir de ahí diseñar la antena, ya que se ha podido ver interferencias y problemas. Además de otros factores como la humedad del suelo, la irregularidad del suelo en la propagación de ondas y el rendimiento de la comunicación. Por último, en esta aplicación en concreto, y no menos importante, se ha notado la necesidad de un marco de ajuste del ciclo de trabajo para mejorar aún más la eficiencia energética del sistema. Este ciclo debe adaptarse a la variación de la humedad del suelo y a la variación de la temperatura del medio ambiente.

Dentro de estos sistemas académicos, se ha desarrollado una IOU basada en Zigbee en el campo de la agricultura de precisión. En ella las propiedades relacionadas con la superficie de los campos como la humedad y el pH eran detectadas por esta aplicación a múltiples profundidades. Más concretamente, esta arquitectura consiste en la unión de nodos inalámbricos, sensores y pasarelas de transferencia para la detección, comunicación y visualización a tiempo real en el campo (Zhang & Wang, 2021).

Como soluciones agrícolas de IOU podemos destacar el Field Connect de John Deere que usa conexiones 3G para transmitir información con sensores de 8 sondas ubicados a una

milla de distancia que miden la humedad del suelo a varias profundidades, la temperatura, velocidad y dirección del viento, etc. (*Precision Ag Technology | Field and Water Management | John Deere US*, n.d.)

Aunque se ha diseñado un conjunto completo de dispositivos de control remoto, los transceptores de frecuencia no pueden llegar a percibir un rango amplio, cosa que también está relacionada con la conexión de cada módulo en hardware, si se espera que la transmisión de información se puede llegar a completar en distancias mayores en el futuro. Por otro lado, se ha percibido que los sensores inalámbricos pueden llegar a recolectar todos los datos en el proceso, pero no llegan a valores críticos, lo cual es inútil en la toma de decisiones. Se espera que esta parte de los datos se pueda llegar a filtrar en trabajos futuros.

Dentro de este campo podemos destacar una serie de desafíos en el diseño y en la implementación de IOU en la agricultura de precisión. Por una parte, la necesidad de mecanismos de seguridad avanzados para proteger la transferencia de información en los campos. Además de soluciones privadas para que la información de múltiples campos se pueda fusionar para tomar decisiones más precisas mientras se mantiene la privacidad de los productores. Por otro lado, debido a la gran área despliegue en estos campos, los dispositivos IOU de menor coste y complejidad deben de tener la capacidad de soportar terrenos accidentados en cualquier tipo de circunstancia, pero sobre todo en cualquier tipo de humedad del suelo.

En definitiva, podemos ver que la *aplicación Internet of Underground Things* (IOU) se introdujo para la toma de decisiones en tiempo real para los campos agrícolas. La detección y comunicación son sus componentes principales y dónde se han desarrollado múltiples aplicaciones y soluciones a lo largo del tiempo.

2.2 La IA en la agricultura de precisión

En este marco de la agricultura de precisión podemos clasificar las aplicaciones de IA que han tenido un gran impacto en tres grandes bloques: las enfermedades y plagas, la calidad del grano, y la fenología y el fenotipado, pertenecientes al siguiente artículo (Patrício & Rieder, 2018).

Las tres tienen una gran repercusión en la producción de cereales, la cual juega un papel importante en la economía mundial. Esta se ve afectada por la aparición de

enfermedades o infecciones provocadas por insectos y plagas. Lo que repercute de manera directa produciendo aspectos negativos en el desarrollo del cultivo y en la calidad del grano, provocando una variación en el precio de mercado. Por tanto, esta identificación, gestión y detección de enfermedades de forma efectiva garantiza una agricultura productiva y sostenible.

El **diagnóstico de enfermedades** se realiza de forma visual y tiene un carácter laborioso y subjetivo. Por ello, se ha trabajado con el objetivo de proponer métodos de visión artificial con inteligencia artificial para automatizar este proceso. La detección de enfermedades se realiza a partir de imágenes e incluye la determinación de características discriminatorias.

Para hallar las plagas se pueden utilizar trampas para mejorar la calidad de segmentación de la imagen y el especificar los insectos que intervengan. Pero antes de esto hay que hacer un seguimiento previo de la especie, ya que muchas de ellas pueden tener una coloración parecida.

Más específicamente, se llegó a explorar la aptitud de los métodos de *computer vision* para detectar y categorizar las enfermedades foliares de la planta de soja (Shrivastava et al., 2017).

En el estudio analizaron los siguientes descriptores:

- WDH (histograma de color descompuesto de ondas).
- BIC (Border /Interior Clasificación) es un método compacto para caracterizar cada píxel de la imagen como borde o interior.
- CDH (histograma de diferencia de color) que permite codificar la apariencia visual de las imágenes.
- LAP (fase angular localizada) que calcula la magnitud y la fase de la transformada de Fourier de los vecinos de cada píxel y sirve para codificar la información de la textura.

Y para la clasificación se analizaron SVM (Support Vector Machine), KNN (k-Nearest Neural Network) y PNN (Probabilistic Neural Network).

Entre los principales resultados se puede destacar que BIC, histograma y WDH presentaron los mejores desempeños en la detección de enfermedades de la hoja de la soja analizadas en el estudio. Los descriptores CDH y LAP fueron más robustos para imágenes

desenfocadas. Finalmente, entre los métodos de clasificación de cationes utilizados el mejor rendimiento fue obtenido por KNN y SVM.

Dentro de las ventajas e inconvenientes de cada uno de los descriptores, se ha podido observar que en los casos de que las imágenes coincidentes de las medidas de similitud de cada descriptor sean menores a la base de datos, este tipo de proceso presenta deficiencias. También se ha podido notar que ciertos descriptores, sobre todo los bidimensionales, requieren un mayor tiempo promedio de recuperación por lo que hace que sean menos eficientes.

Los pulgones son unas de las plagas que más afecta en la cosecha del trigo. Estos insectos se alimentan de la savia del floema del trigo, afectando a su desarrollo y pudiendo transmitir una serie de patologías vitales.

Para su detección propusieron el método SMH, el cuál combina el uso de SVM utilizado para la presencia o ausencia de pulgones; el algoritmo MSER, que tiene objetivo simplificar el fondo de las imágenes y HOG (Histograma de gradientes), que se utiliza como un parámetro de clasificación utilizando SVM en la extracción de los atributos de la imagen. Este método fue probado con diferentes densidades de pulgones, color o ubicación en la pantalla, en dónde comparándose con otros cinco métodos comúnmente utilizados, obtuvo unos resultados que mostraban un mayor rendimiento (Liu et al., 2016).

En este tipo de aplicaciones se ha podido comprobar que se necesita un gran número de ejemplos, tanto positivos como negativos, para poder establecer un buen clasificador de reconocimiento de hojas, lo que dificulta la aplicación para la identificación y el recuento de plagas en otros campos de cultivo.

Respecto a la **calidad del grano** vemos que influyen varios factores genéticos y ambientales. En este caso, el uso de técnicas convencionales suele presentar errores de clasificación, así que se opta por emplear técnicas como Machine Learning. La cual a la hora de la clasificación de imágenes ha destacado como una forma de construir clasificaciones precisas e inteligentes. Para ello, se requiere información sobre el tipo y la calidad del grano en las etapas de procesamiento de este. Este método de clasificación automática puede prevenir errores humanos en el proceso de evaluación de calidad, convirtiéndolo en una alternativa a la inspección manual.

Por ello, se presentó un sistema de computer vision que usa un enfoque simplificado de clasificación con un alto índice de precisión. El objetivo del sistema es clasificar los granos

de trigo de la especie *Triticum aestivum* y *Triticum durum* según sus características visuales utilizando una red neuronal artificial del tipo MLP (Sabanci et al., 2017).

Sobre el sistema, se ha obtenido que la peor precisión viene de los subconjuntos relacionados con las características de textura, esta situación estimula a investigar un nuevo conjunto de datos teniendo en cuenta este tipo de características para lograr un mayor impacto en los resultados.

Se combinaron la *computer vision* y los métodos de inferencia difusa en la construcción de un sistema de apoyo a la decisión para la clasificación de la calidad del arroz pulido (Zareiforush et al., 2015).

Para evaluar el rendimiento del modelo desarrollado, los resultados de la clasificación de calidad de las muestras de arroz elaborado se compararon con los de expertos humanos, y aunque se obtuvieron resultados con gran confianza, se estudia poder llegar a mejorar todavía más esta eficiencia.

La **fenología** es la rama de la ecología que estudia los eventos que ocurren durante el ciclo de vida de la planta en reacción a las condiciones del medio. La aplicación de un sistema de *computer vision*, en este caso, sería una alternativa productiva y no invasiva al trabajo de observación manual requerido en el **proceso de fenotipado** de plantas.

En la siguiente investigación proponen un sistema de observación automático para el desarrollo de espigas de trigo basado en *computer vision*. Las imágenes se obtienen en condiciones naturales con frecuentes cambios de iluminación directamente desde el campo (Zhu et al., 2016).

El desarrollo de este sistema ha estado más enfocado en la aparición de espigas que en las propias características físicas. Por este motivo, esta investigación se puede ampliar con el fin de extraer más características biológicas relacionadas con el rendimiento de los cultivos. Por ejemplo, se pueden obtener más rasgos esenciales contando y midiendo espigas.

Para ello se propuso un método de visión artificial para detectar el estado nutricional del nitrógeno en el maíz. La identificación del estado nutricional mediante la química de las hojas requiere el muestreo de hojas cuando la planta se encuentra en una etapa avanzada de desarrollo. De esta forma, impide la acción correctiva en el cultivo en curso si se detecta deficiencia de un nutriente específico (Romualdo et al., 2014).

Cabe destacar que la idea principal presentada en este trabajo podría llegar a ser adoptada para otras plantas, pero se resalta que el desempeño de este método es totalmente

válido para las condiciones en las que se realizó, por lo que se debería llegar a considerar como desafío extrapolar este trabajo a diferentes condiciones.

Entre todas estas aplicaciones podemos ver que no hay un trabajo predeterminado que se pueda aplicar de forma genérica. Por ejemplo, la segmentación de imágenes es fundamental para lograr el resultado deseado, pero no es una tarea nada trivial ya que la iluminación, el fondo de la imagen o la posición de esta son algunos de los aspectos que hacen que esta tarea sea un desafío.

Otro gran desafío es al que se presentan los algoritmos en donde debido a la gran variedad de características fisiológicas de las plantas aumenta aún más la diversidad de los factores que son analizados en el procesamiento de imágenes.

Estos algoritmos de *Machine Learning* son bastante sensibles al volumen y la calidad de la información que utilizan, algo que se ve afectado cuando estos “aprenden” debido a que en ciertas ocasiones el ruido o los detalles excesivos del conjunto de entrada pueden influir negativamente en la generalización del modelo construido.

También hemos podido comprobar como en el campo de la agricultura de precisión se utilizan herramientas y tecnologías para identificar la superficie del terreno y la variabilidad de los cultivos con el fin de mejorar las prácticas agrícolas y optimizar los insumos agrícolas.

La temperatura de la superficie se considera una variable de respuesta rápida que puede indicar el estrés del cultivo previamente de sus síntomas visuales. Por lo que *Thermal remote sensing (RS)* se puede utilizar en muchos aspectos del monitoreo de cultivos y superficies en el sector agrícola. Como por ejemplo, la estimación del estrés hídrico del suelo y del cultivo para la programación del riego, el mapeo de la textura de la superficie, etc.

Hoy en día, el **riego** es una parte fundamental de la producción agrícola, sobre todo en aquellas regiones donde las lluvias son irregulares o inadecuadas durante la temporada para poder satisfacer a los cultivos. Saber dónde, cuándo y cuánto ayuda a minimizar la pérdida de rendimiento de los cultivos y optimizar la cantidad de agua utilizada, provocando un gran beneficio para el agricultor.

La necesidad de riego está determinada principalmente por cuatro factores: disponibilidad de agua en el suelo, necesidad de agua para los cultivos, cantidad de lluvia y eficiencia del sistema de riego. Aparte de estos factores, también se suele cuantificar la humedad de la superficie, la temperatura de las plantas y la evapotranspiración.

La monitorización de la humedad del suelo es algo esencial, ya que aporta nutrientes en el crecimiento de las plantas y regula la temperatura del suelo que influyen en las tareas agrícolas. Como podemos ver en el siguiente aprendizaje que utilizó la aplicación de imágenes térmicas para estimar con precisión la distribución espacial de la humedad superficial del suelo en una granja de Utah plantada con alfalfa y avena (Hassan-Esfahani et al., 2015).

En este estudio se muestra como la capacidad de retención de agua del suelo en función de su textura juega un papel importante en la determinación de los valores de la humedad del suelo, aunque en ciertos casos se ha podido ver como el efecto de este parámetro puede llegar a confundir con otros insumos importantes en la distribución espacial de la humedad del suelo.

Otra de las propiedades del suelo es la textura afectada por factores como la porosidad, la capacidad de retención de nutrientes entre otros, lo que a su vez influye en la productividad del cultivo. La temperatura y la textura están bastante correlacionadas. La segunda de ellas influye en el contenido de agua y esto a su vez influye en la temperatura de la superficie.

Por tanto, al examinar diferencias de temperatura en condiciones climáticas parecidas, los siguientes estudios demostraron también el uso de Thermal RS en la cuantificación de la textura del suelo a escala regional (Wang et al., 2012, 2015).

Aunque entre los diferentes estudios de cada hemos podido ver una mejora, el método utilizado tiene un gran potencial para mejorar el mapeo digital de las propiedades del suelo en áreas de bajo relieve.

RS proporciona información espacial y temporal de la temperatura de la superficie del suelo y del cultivo, pero al utilizar imágenes térmicas de RS existen ciertos factores que hay que tener en cuenta. Entre estos podemos destacar los efectos de absorción y dispersión atmosférica para una representación precisa de la temperatura de la superficie; o las condiciones de niebla y nubosidad que pueden influir en los datos térmicos recopilados de las plataformas aéreas y satelitales (Khanal et al., 2017).

La detección y clasificación de las características de calidad de los cultivos pueden aumentar el precio del producto y reducir los desperdicios. Debemos mencionar entre la gran variedad de estudios uno en concreto referente a la producción de peras, en donde se presentó un método para la identificación y diferenciación de las peras aromáticas. Este estudio aplicó métodos de Machine Learning con imágenes espectrales (Hu et al., 2017). Tras

varios resultados se perciben ciertas, más bien mínimas, deficiencias en la precisión y la solidez de los modelos, cosa que podría mejorarse aún más con tamaños de muestra más grandes y situaciones más reales.

Además de la importancia de estos aspectos, destaca el rendimiento y calidad de los cultivos. La predicción del rendimiento es uno de los aspectos más importantes en la agricultura de precisión y es de gran importancia para la adecuación de la oferta de cultivos con la demanda y el manejo de cultivos para aumentar la productividad. Entre las posibles aplicaciones podemos destacar un sistema de visión artificial para automatizar la recogida de cerezas durante la cosecha (Amatya et al., 2016). Este sistema segmenta y detecta las ramas de cerezo con el objetivo de reducir los requisitos de mano de obra en las operaciones manuales de recolección y manipulación.

Como paso futuro, se trabajará en la identificación y localización de las cerezas para tomar decisiones en términos de ubicación de agitación dentro de una rama de cereza. Integración con localización 3D, el efecto final de agitación y los métodos de planificación de ruta tendrían que seguir para desarrollar y evaluar un prototipo de recolección automatizada.

La detección y el manejo de malezas es otro problema y aspecto de la agricultura, se clasifican entre los productores como la amenaza más destacable. Una detección precisa asegura una agricultura sostenible, debido a la dificultad para tratarlas.

En un estudio desarrollaron un método basado en técnicas de Machine Learning e imágenes espectrales para el reconocimiento de especies de cultivos y malezas. Es decir, estos autores crearon un sistema de aprendizaje activo para el reconocimiento del maíz con el objetivo de reconocer y discriminar estas especies con fines económicos y ambientales (Pantazi et al., 2016).

Este sistema propone una aplicación en tiempo real por lo que se sugiere un ajuste de los algoritmos con respecto al tiempo de ejecución y la precisión correspondiente para no tener grandes variaciones en los resultados finales.

Los resultados de este tipo de aplicaciones son bastante variables debido al uso de imágenes al aire libre, un buen uso tanto de hardware como software es fundamental. Además, sirve para ver la verdadera eficiencia de estas aplicaciones que deben abordarse en diferentes situaciones e ir mejorándose con sistemas de comunicación más eficientes.

La rapidez del procesamiento depende de las dimensiones de la imagen, debido a que en algunos casos es necesario redimensionar la imagen antes de procesarla. Y por otro lado,

la calidad de la imagen suele verse afectada por la oscuridad o por el nivel de intensidad, así que hay que tener en cuenta estos factores en el desarrollo de sistemas de este tipo.

2.3 Índices de vegetación hiperespectral en referencia al cultivo del tomate en la agricultura de precisión

Tal y como ya podemos saber, la agricultura de precisión utiliza el conocimiento específico de cada sitio para ayudar a los agricultores y a las partes interesadas, con el fin de administrar mejor sus recursos limitados con las nuevas tecnologías.

En el momento en el que se puede medir y monitorear la variabilidad del campo, vemos que se pueden adaptar prácticas específicas para aplicar de una manera más inteligente y eficaz los insumos, para mejorar la eficiencia en general en los sistemas agrícolas. Aquí, entra una de las prácticas más importantes que es la mejora de la gestión del riego, la cual es clave para hacer frente a los efectos actuales y futuros del cambio global en los recursos hídricos.

A lo largo de los años se han propuesto muchos procedimientos cuantitativos para la programación del riego, entre ellos destaca el uso de técnicas de detección remota y proximal, así como la reflectancia multispectral, que pueden proporcionar información instantánea y cuantitativa sobre el estado hídrico de muchos cultivos. Esta detección proporciona datos reales del suelo de los cultivos para poder mapear y solucionar problemas y errores de riego, llegando hasta el punto de que la productividad del agua relaciona la producción del cultivo con el uso del agua y es un indicador clave en la gestión de esta.

Con el objetivo de hacer un mejor uso de este tipo de datos espectrales se han desarrollado múltiples índices espectrales de vegetación (VI), utilizando la reflectancia de los cultivos a diferentes longitudes de onda. Estos índices están fuertemente modulados por interacciones de la radiación solar con tejidos fotosintéticamente activos, por lo que son fuertes indicativos de propiedades biofísicas dinámicas relacionadas con la productividad y el equilibrio energético de la superficie (Basso et al., 2004). Los VI se pueden llegar a usar para monitorear las condiciones de las plantas, evaluar el estado de los nutrientes, estimar la tasa de crecimiento de las plantas, pronosticar los rendimientos de los cultivos, etc.

Dentro de la gran variedad de cultivos, vemos que el tomate tiene un gran dominio en el mercado mundial, aunque no se han realizado muchas investigaciones sobre el crecimiento

y la fisiología de este con diferentes disponibilidades de agua. Los estudios se han dirigido principalmente a diagnosticar los efectos de la variabilidad de la humedad del suelo y la calidad del fruto. Se requieren unos estudios más exhaustivos en la fisiología del cultivo para modelar el comportamiento general del tomate cultivado en el campo bajo una disponibilidad variable de agua.

Por ello, vamos a analizar y destacar los resultados de estos índices de vegetación en dos estudios realizados sobre el estudio del tomate:

El primer experimento se llevó a cabo durante la temporada de cultivo de 2011 en el sur de Italia en una llanura dedicada a cultivos hortícolas (Marino et al., 2015). Sin entrar en todo tipo de detalle sobre el área de estudio, las mediciones realizadas y los análisis geoestadísticos y de regresión, se puede observar que este estudio fue una representación real de un cultivo de tomate de campo con riego expuesto a problemas comunes indeseables como pueden ser el riego excesivo y los anegamientos.

La originalidad de esta investigación radica en el uso de una herramienta geoestadística para producir una distribución espacial de los VI y los rasgos de los cultivos, llegando a proporcionar información importante para mejorar la rentabilidad de los sistema de riego y para hacer que se puedan hacer en diferentes condiciones y lugares. En definitiva, se pudo ver la idoneidad de estos VI para detectar la eficiencia espacial del agua del riego, como pueden ser en este caso concreto en los mapas de TSAVI, OSAVI y SAVI, e incluso estos índices pueden llegar a mejorar más aún la precisión de los estudios de teledetección sobre la productividad de los cultivos.

En el segundo estudio se utilizaron imágenes hiperespectrales en entornos de laboratorio y de campo para detectar enfermedades (Abdulridha et al., 2020). Las plantas de tomate enfrentan desafíos de numerosas enfermedades en la producción de Florida como pueden ser la mancha de objetivo (TS) y la enfermedad de la mancha bacteriana (BS) llegando a reducir de manera considerada el rendimiento comercial.

Al igual que en el anterior estudio, sin entrar en detalle en los métodos de clasificación y los materiales y mediciones utilizados, se calcularon treinta y cinco índices espectrales para poder llegar a seleccionar un conjunto óptimo de índices para la detección e identificación de enfermedades. Entre estos, se pudo observar que los VI más significativos que pudieron distinguir entre hojas sanas y enfermas fueron PRI para las dos enfermedades, el NDVI para la enfermedad de la mancha bacteriana y el TVI y el Chle green para la enfermedad TS. Sobre

este tipo de índice se profundizará más a lo largo del documento y en especial en la etapa de implementación ya que conjunto de datos contiene valores sobre estos índices.

2.4 Hongos en el cultivo del tomate

Como venimos hablando a lo largo de este estudio, el tomate es uno de los cultivos más importante en el mundo por su impacto alimenticio y económico. De ahí su importancia en intentar prevenir todo tipo de enfermedades y problemas que perjudiquen su producción. Además, este cultivo suele ser propenso a la agresión de varios patógenos.

Entre todas las enfermedades que puedan afectar a este cultivo podemos destacar como más importantes:

- **Oidio** (*Leveillula taurica*): Es una enfermedad criptogámica producida por varios géneros de hongos que atacan principalmente a hojas y tallos. Esta enfermedad se muestra en la aparición de manchas amarillas sobre las hojas llegando a secarlas.
- **Podredumbre gris** (*Botrytis cinerea*): Es una enfermedad fúngica que puede atacar hojas, tallos, frutos, etc. En hojas y flores se producen lesiones pardas y en los frutos se observa una vellosidad gris del hongo.
- **Mildiu o tizón tardío** (*Phytophthora Infestans*): es un protista fungoide y suele infectar principalmente a las papas y a los tomates atacando a la parte aérea de la planta y en cualquier etapa de desarrollo.
- **Alternariosis** (*Alternaria solani*): Es una enfermedad basada en el ataque de hongos del género *Alternaria*. Esta se visualiza cuando en las hojas se producen manchas pequeñas circulares o angulares.
- **Fusarium oxysporum**: Es una especie de hongo que coloniza los conductos de la planta bloqueando y tapando los vasos llegando a producir un amarilleo que comienza en las hojas más bajas y que termina por secar la planta.

Debido al estudio realizado en este trabajo nos centraremos en la enfermedad del tizón tardío por medio del hongo *Phytophthora Infestans*.



Figura 2.1. Imagen representativa de la afectación del hongo en una planta.

Repasando el historial de este parásito podemos destacar que fue el causante principal de la hambruna irlandesa de 1845 a 1849 y de la escocesa de 1846 a 1857. Más recientemente a principios de la década de los 2000s causó una gran pérdida de la producción del tomate en Mongolia Interior de China y EE.UU.

Por lo que no cabe duda que este hongo se ha convertido en uno de los patógenos más devastadores que pueden causar grandes pérdidas en la producción del tomate. Por ello, es necesario estudiar los diferentes mecanismos de resistencia del tomate frente a este.

Hasta ahora, los mecanismos moleculares del tomate-*Phytophthora Infestans* están más allá de la comprensión. El método para controlar el tizón tardío aún no es lo suficientemente eficaz. Debido a este tipo de aspectos y situaciones aumenta la necesidad de estudiar los mecanismo de resistencia del tomate contra este hongo y ser capaz de identificar aquellos genes que muestran una resistencia clave que se utilizarán en el mejoramiento del transgén del tomate resistente a otras enfermedades (Jiang et al., 2018).

En un estudio realizado para ver los efectos del tizón tardío sobre el cultivo del tomate y de la patata se obtuvieron resultados y conclusiones bastante significantes (Nowinck Marcin, R. Foolad Majid , Nowakowska Marzena, 2012):

- *Phytophthora Infestans* es uno de los patógenos vegetales más devastadores de todos los tiempos.
- Si las condiciones son adecuadas, el ritmo de expansión es enorme llegando a amenazar la producción de cultivos a gran escala.

- Gracias a las investigaciones realizadas sobre las interacciones entre las plantas hospedadoras y este patógeno se ha contribuido con una gran cantidad de conocimientos a nuestra comprensión de la biología celular y molecular de las estrategias de infección por patógenos y por componentes claves de estas.

- Estudios de inmunocitoquímica junto al marcado de proteínas vegetales han proporcionado un conocimiento detallado a nivel celular de los cambios producidos durante el inicio y la producción de la infección, la estructura y comportamiento de las esporas de este hongo.

- Los estudios realizados sobre los efectos de los oomicetos aumentan nuestro conocimiento sobre cómo estos agentes inhiben las defensas de las plantas y aumentan la fuerza de los patógenos.

En definitiva, la secuenciación del genoma de *Phytophthora Infestans* ha abierto un capítulo que ya lleva abierto en los estudios de la patogenicidad, permitiendo una clasificación y análisis de alto rendimiento de las proteínas efectoras putativas. Lo que ha llevado al descubrimiento de genes implicados en la virulencia que antes se pasaban por alto.

Genes que afectan directamente al desarrollo y a la producción del cultivo del tomate en este caso y otros múltiples cultivos. De ahí la importancia en ser capaz de identificar todo tipo de posibles mejoras e inconvenientes que afecten directamente en este cultivo.

3

Metodología

Scrum será la metodología de trabajo utilizada en este proyecto. La gran flexibilidad, la capacidad de adaptación y organización son los principales motivos por los que se ha escogido esta metodología. Scrum nos permite adaptarnos y corregir cualquier tipo de inconveniente por medio de los sprints, pequeñas tareas que tienen un tiempo específico para cumplirse.

En el siguiente diagrama se puede observar las actividades que formarán parte del proyecto.

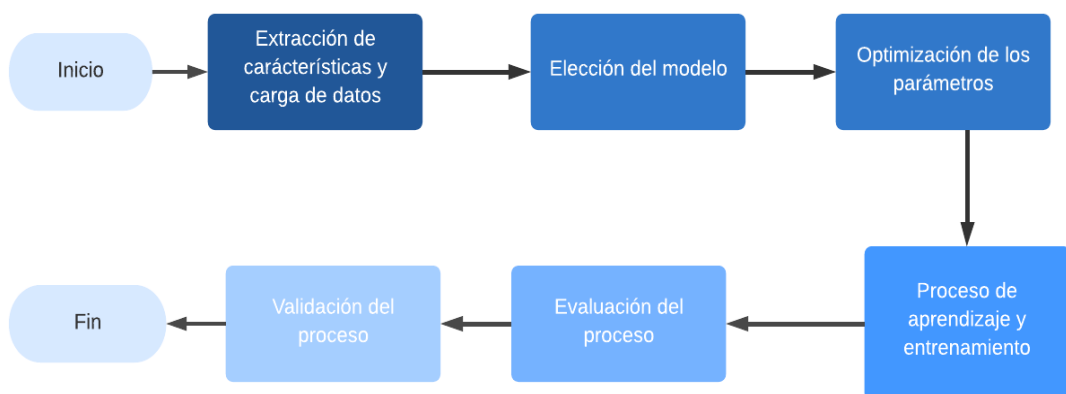


Figura 3.1. Actividades que forman el proyecto.

En este proyecto se desarrolla un estudio de cómo afecta la inyección del hongo *Phytophthora Infestans* en el cultivo del tomate. Este hongo es un parásito de las plantas que produce la enfermedad conocida como tizón tardío.

Para ello se utiliza un conjunto de datos que contiene ciertos índices de vegetación en relación a una plantación del cultivo del tomate proporcionado por una universidad de Brasil. Los índices representados son los siguientes:

- **NDVI** (Normalized Difference Vegetation Index): El Índice de Vegetación de Diferencia Normalizada es un índice de vegetación que se utiliza para poder estimar la calidad, cantidad y desarrollo de la vegetación en base a la medición de la intensidad de la radiación de ciertas bandas del espectro electromagnético.

Es un buen indicador de la vigorosidad de la planta de forma directa, es decir, informa del estrés del árbol (hídrico, nutricional, etc.).

- **SAVI** (Soil-Adjusted Vegetation Index): Tras la demostración de la inestabilidad del NDVI en relación a los efectos de saturación, al color y a la humedad del suelo. Este índice es una técnica de transformación que minimiza las influencias del brillo del suelo de los índices espectrales de vegetación que involucran longitudes de onda roja e infrarroja cercana.

- **GNDVI** (Green Normalized Difference Vegetation Index): El Índice de Vegetación de Diferencia Normalizada Verde es un índice que indica el “verdor” de la planta, es decir, su actividad fotosintética. Destaca por ser uno de los índices más utilizados para determinar la captación de agua y nitrógeno en el cultivo.

- **MCARI 1** (Modified Chlorophyll Absorption in Reflectance Index): Este índice surgió como modificación del CARI, desarrollado para reflejar la variación de la cantidad de clorofila existente en la vegetación. Bastante sensible al índice foliar.

- **SR** (Simple Ratio): Índice ubicado entre la banda del infrarrojo cercano y el rojo para resaltar las cubiertas vegetales. Tiene la capacidad de reducir o eliminar la influencia del suelo sobre los valores de reflectancia solar.

Estos índices del conjunto aparecen en relación a una secuencia de temporal, obteniendo para cada índice cuatro resultados de cuatro días distintos por cada cultivo, todos ellos tomados de la misma forma y a la misma hora en un intervalo de doce días, con la misma diferencia entre estos.

Al mismo tiempo se tomaron fotografías del estado de las plantas en los días correspondientes para que especialistas pudieran estimar el grado de severidad del hongo visible en el cultivo.

El principal objetivo de este trabajo es la creación de un motor de inteligencia que nos permita predecir el grado de severidad del hongo *Phytophthora Infestans* en el cultivo del tomate en función de estos índices de vegetación.

La evaluación final del grado de severidad se deducirá través de la escala mostrada a continuación. Nuestro objetivo será utilizar estos índices de vegetación para lograr llegar a los resultados de esta escala.

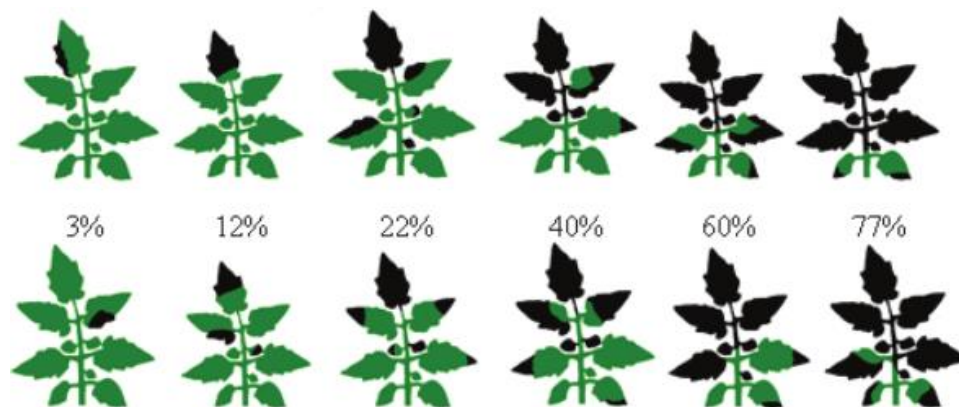


Figura 3.2. Escala utilizada para evaluar la gravedad del tizón tardío en plantas de tomate causado por *Phytophthora Infestans*. Las afectaciones producidas por el tizón tardío se destacan en negro. Fuente: (Corrêa et al., 2009)

Se estima que aquellos que superen en un 40 % la población del hongo se cataloga como *Classe = 1*, lo que significa que no han sido capaces de ser lo suficientemente fuertes y fértiles para combatir este hongo. En cambio, aquellos que no superen ese 40% se clasifican como *Classe=0*, y mostrarán una gran resiliencia frente al hongo.

Por medio de este umbral, se determina la diferencia entre los cultivos que han sido capaces de combatir al hongo y los que no, esto nos ayudará a tomar medidas con el objetivo principal de lograr una mayor fertilidad en el cultivo y obtener un mayor rendimiento en la vida y producción de estos.

En el conjunto de datos también se dictamina que para obtener mayor precisión en los resultados y en las predicciones se establecerá otro umbral para uno de los índices, quedándonos solo con aquellos que muestren tener un NDVI > 0.6.

Cabe destacar que también se ofrecerá por medio de gráficas una evaluación del rendimiento y de la efectividad del modelo y motor implementado.

Para llevar un correcto desarrollo se dividirá el conjunto de datos en tres conjuntos diferentes. El primer subconjunto será el de entrenamiento que consta del 60 % de los datos con el objetivo de desarrollar una buena etapa de aprendizaje con una amplia cantidad de datos. Posteriormente, en el segundo subconjunto se utilizó un 30 % de los datos con la finalidad de determinar si este modelo cumple los requisitos necesarios, esta fase se denomina test. Y para el tercero que será el de validación tenemos un 10 % de los datos. Esta fase ayudará a saber si realmente este motor de inteligencia puede trabajar con otros conjuntos en el mundo, para ello los datos utilizados no se habrán utilizado anteriormente.

Gracias a la información obtenida por los responsables del conjunto se ha podido disminuir la cantidad de datos para el motor de inteligencia, quedándonos solo con los valores obtenidos de los índices del último día, concretamente el 8 de julio de 2021. Esto es debido a la gran variabilidad que tiene el hongo en expandirse, en dónde en los primeros días puede llegar a obtener un grado de severidad bastante bueno y repentinamente crecer de forma inconmensurable. Es por ello por lo que a partir de este último día se da a entender que el hongo ya ha tenido suficiente tiempo para desarrollarse y podemos llegar a determinar el grado de población y fertilidad del cultivo.

A continuación, vamos a explicar con más detalle cada una de las actividades del diagrama anterior que forman al proyecto:

- **Extracción de características y carga de datos:** En el desarrollo del motor de inteligencia la parte de extracción, carga y procesamiento es fundamental debido a que es la parte en donde se lleva el aprendizaje que va a tener nuestro sistema de inteligencia y la que determinará gran parte del éxito.

Al comenzar, en el preprocesamiento de los datos, se tuvo que hacer una normalización de estos debido a la diferencia en los valores entre índices.

El tiempo necesario para completar esta etapa fue de 15 días, ya que se partía con un gran desconocimiento sobre el significado de estos índices y su posterior

utilización. Cabe destacar como hemos nombrado anteriormente el proceso de normalización de estos datos.

- **Elección del modelo:** En segundo lugar, tenemos la elección del modelo de aprendizaje, esta etapa duró 7 días, en los que se llevó a cabo la búsqueda del modelo que mejor se adapte a nuestro conjunto de datos.

Este modelo es de tipo supervisado ya que al trabajar con un conjunto de índices de vegetación de una planta concreta deberá determinar si esa planta es lo suficientemente fuerte para vencer y no mostrar debilidad sobre el hongo basándose en el grado de severidad ya especificado por medio de la escala. En función de las características y parámetros del problema será la red neuronal utilizada.

- **Optimización de los parámetros:** Posteriormente pasaremos a la optimización de los parámetros, esta etapa es de gran importancia, porque nos permite conectar la información extraída con el modelo escogido y nos ayuda a reducir y mantener un buen valor de la función de pérdida que asegure un buen aprendizaje con el fin de cumplir todos los requisitos y lograr todos los objetivos del modelo.

En el cómputo global se puede decir que esta etapa y la anterior llegaron a durar unos 10 días, ya que la investigación de la optimización de los datos está unida a la investigación de la red neuronal a utilizar.

En la elección de este tipo de funciones y características a llevar en la arquitectura del proceso se tienen en cuenta múltiples factores como son el número de datos, el aprendizaje supervisado llevado a cabo y los valores de los índices normalizados todos entre 0 y 1.

Tanto en las selecciones anteriores como en la selección de los hiperparámetros de la tasa de aprendizaje, el número de épocas, el tamaño de la capa oculta, la función de optimización y el tamaño de los lotes de datos se han establecido en función de los resultados obtenidos. Es decir, se han ido graduando y ajustando conforme el modelo iba obteniendo mejor rendimiento, esta etapa fue de una duración de 3 días.

A continuación, pasamos a las tres fases finales.

- **Proceso de aprendizaje y entrenamiento:** En primer lugar, tenemos el proceso de aprendizaje de nuestro motor que nos ayudará a alcanzar el rendimiento esperado llegando a ser capaz de interpretar los datos proporcionados de forma

correcta. Para ello utilizaremos el conjunto de entrenamiento con el fin de que el modelo sea capaz de adaptarse a este llegando a conseguir los mejores resultados posibles.

Los valores seleccionados anteriormente como hiperparámetros tienen bastante relevancia en esta fase del proceso, debido a que determinarán que nuestro modelo sea capaz de presentar en las fases posteriores unos buenos niveles de precisión y efectividad. Como se ha nombrado anteriormente, la formación de esta etapa viene determinada por la prueba de diferentes valores y a partir de ahí ver qué resultados son mejores.

- **Evaluación del proceso:** La siguiente fase consiste en una evaluación del sistema desarrollado con el objetivo de dar veracidad a los resultados obtenidos llegando a cumplir unos estándares de calidad que nos aseguren que el modelo ha sido implementado de forma correcta. Para esta fase se utilizará un conjunto de menor tamaño y distinto que el de entrenamiento llamado test.

Además de ayudar a medir la precisión del modelo, esta fase nos servirá de ayuda para poder elaborar diferentes gráficas que nos permitan facilitar la comprensión de los datos y de los resultados obtenidos.

- **Validación del proceso:** Finalmente, tenemos la fase de validación, esta fase nos servirá para poder garantizar que la implementación del modelo se ha realizado con éxito y se puede llevar a utilizar con otros datos del mundo real. Esta fase contiene un conjunto de datos más pequeño que no ha sido utilizado en ninguna de las otras dos fases lo que nos asegurará que este modelo cumple con los requisitos y objetivos establecidos.

En conjunto, se puede estimar que el desarrollo de estas tres fases duró aproximadamente 10 días.

Después de tener toda la implementación completa, se pasó a desarrollar como última etapa el conjunto de gráficas que nos ayudará a darle veracidad y comprensión al modelo.

4

Análisis de requisitos

A continuación, se detallarán de una manera clara y concisa cada uno de los objetivos que presenta y debe cumplir el proyecto. Estos requisitos se han especificado a lo largo del proyecto y se clasifican dos tipos: funcionales y no funcionales. Todos son indispensables para el funcionamiento correcto de este.

4.1 Requisitos funcionales

En este primer subapartado se ofrece la lista de requisitos funcionales que ha de cumplir el motor de inteligencia artificial para poder llevar a cabo un buen funcionamiento y desempeñar sus funcionalidades de forma correcta para lograr los objetivos especificados.

En la siguiente tabla se describen y proporcionan información de cada uno de los requisitos de forma breve y concisa para ayudar al lector a entender el por qué la necesidad de cada uno de estos en el sistema.

Requisitos	Descripción
RF-01	Herramienta colaborativa para el uso académico y universitario.
RF-02	La aplicación deberá determinar si cada cultivo es lo suficientemente fuerte y fértil para superar con un buen resultado el grado de severidad del hongo <i>Phytophthora Infestans</i> y a partir de ahí determinar conclusiones.
RF-03	La aplicación deberá generar un conjunto de gráficas que permitan visualizar los resultados obtenidos con el fin de lograr una mayor comprensión sobre el sistema.
RF-04	Esta herramienta deberá de poder trabajar con el formato de los datos del conjunto para poder elaborar de forma correcta sus funciones de correlación de índices y a partir ahí obtener conclusiones elaboradas.
RF-05	La herramienta deberá validar los resultados obtenidos de los algoritmos de inteligencia artificial implementados con el conjunto de datos con el fin de demostrar que pueden ser utilizados para entornos de producción y para verificar los objetivos y conclusiones expuestas.
RF-06	La aplicación deberá mostrar cuál es la efectividad real de nuestro motor.

4.2 Requisitos no funcionales

Para este subapartado se muestra la lista de requisitos no funcionales que especificarán ciertos criterios que podrán usarse para juzgar las operaciones ha de cumplir este motor de inteligencia para garantizar uno mínimos de calidad que debe tener la funcionalidad de la aplicación.

Requisitos	Descripción
RNF-01 (Accesibilidad)	En toda su totalidad la herramienta deberá de poder ejecutarse en cualquier dispositivo independientemente de cuál sea el sistema operativo (Windows, Linux o Mac). Solo será necesario tener la versión más reciente de Python con las librerías necesarias.
RNF-02 (Usabilidad)	La herramienta será fácil de usar y debe estar totalmente disponible, ya que solo dependerá de la máquina o dispositivo en el que se ejecute.
RNF-03 (Mantenimiento)	La herramienta no necesitará ningún tipo de mantenimiento periódico.
RNF-04 (Confianza)	Los resultados obtenidos de las predicciones y correlaciones deben de ser cercanos a un porcentaje determinado con el fin de poder justificar la validez de la aplicación.
RNF-05 (Eficiencia)	Los tiempos de respuesta y ejecución deben de ser mínimos a la hora de elaborar informes y predicciones con el fin de que el motor sea todo lo eficiente posible y evite tiempos de pérdida.

Como bien se ha mostrado anteriormente, tener una buena identificación de los requisitos del proyecto, tanto funcionales como no funcionales, es imprescindible debido a que nos ayudará a prevenir y anticiparnos a posibles errores, a mejorar la imagen pública del proyecto, a cumplir unos mínimos de calidad y lograr los objetivos esperados.

5

Diseño

5.1 Arquitectura del Software

A continuación, en este apartado se muestra y detalla los componentes de la arquitectura software desarrollada en este proyecto y que conllevarán el correcto funcionamiento de la aplicación, cómo son:

- **Motor de Inteligencia Artificial:** Componente principal de la arquitectura encargado de dar el comportamiento inteligente a la aplicación. Para esto, utilizaremos un entorno determinado llamado *PyCharm Community Edition* que nos ayudará desarrollar el código de nuestro motor proporcionándonos librerías, herramientas y una amigable interfaz.

A partir de estas librerías podremos realizar las actividades y acciones oportunas que nos servirán a la hora de realizar los cálculos necesarios para implementar el conjunto de modelos que nos harán falta. Para el conjunto de modelos haremos uso de redes neuronales y los resultados obtenidos que se caracterizan como prototipos se almacenarán en un gestor de archivos para llevar un control de versiones y mantener unos mínimos de seguridad.

- **Conjuntos de Datos:** Con este componente nos referimos a los datos proporcionados sobre los cuáles trabajará el motor de inteligencia. Este conjunto de

datos en formato .csv estará dividido en tres partes, cada una correspondientes a una etapa del proceso. Cada conjunto está formado por una serie temporal de índices de vegetación, estos datos vendrán representados en un archivo .xlsx de Excel. A partir de estos realizaremos las operaciones necesarias para determinar el grado de severidad del hongo y a raíz de esto obtener conclusiones.

- **Informes (Gráficas):** Este componente se compondrá de los resultados elaborados y generados de todo el proceso. Además, nos proporcionará la efectividad del motor de inteligencia por medio de una evaluación de los algoritmos utilizados.

A continuación, en los siguientes diagramas se mostrará la interacción de los componentes utilizados:

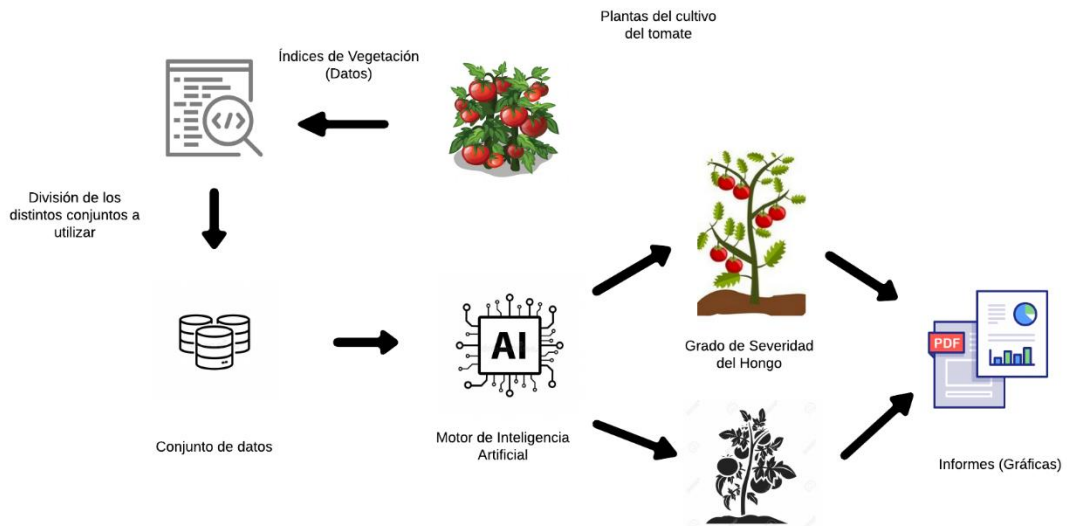


Figura 5.1. Componentes que forman la Arquitectura del Software del proceso y su interacción entre ellos.

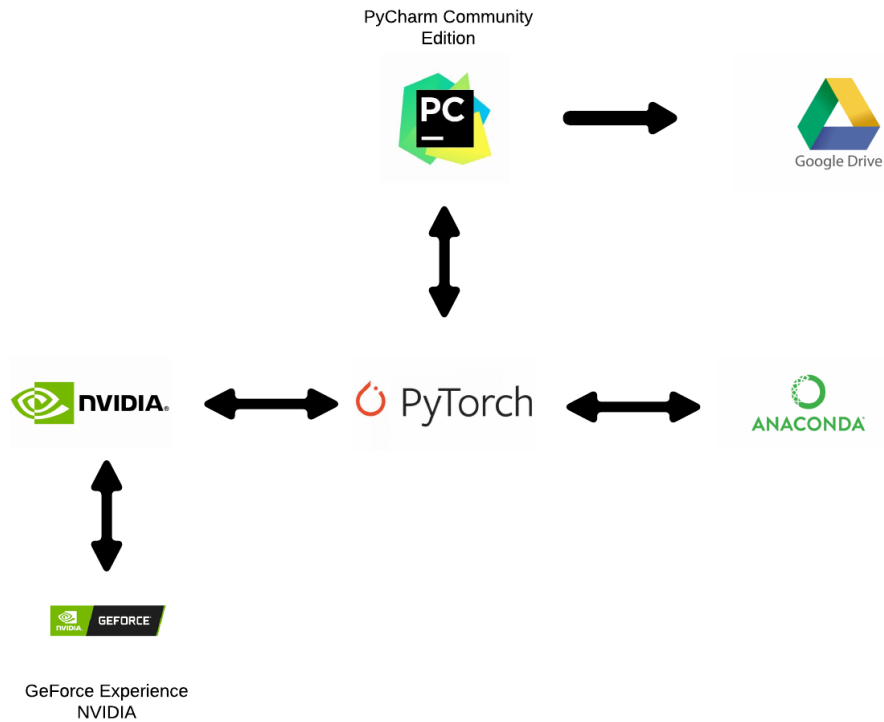


Figura 5.2. Componentes que forman al motor de inteligencia artificial junto a su interacción entre ellos.

5.2 Tecnologías

A continuación, en este apartado se muestran todas las tecnologías usadas en el proceso de implementación, es decir, el conjunto de herramientas y librerías usadas para la carga de los datos en el entorno y el desarrollo del código del proceso.

Antes de profundizar en aspectos del entorno de entorno de prototipado cabe destacar la instalación y uso de aplicaciones como:

- **Anaconda:** Una interfaz gráfica de usuario GUI que nos permite gestionar de manera avanzada paquetes relacionados a la ciencia de datos. Anaconda funciona como un programa de Python que contiene los paquetes más utilizados.
- **NVIDIA:** Es una plataforma que proporciona tarjetas gráficas y diseñadores de unidades de procesamiento de gráficos (GPU).
- **GeForce Experience:** Es un paquete de utilidades para las tarjetas gráficas de NVIDIA y una herramienta bastante versátil para administrar GPU Nvidia.

El uso e instalación de estas aplicaciones se debe a que el interpretador que traía incorporado el entorno de PyCharm ocasionó ciertos problemas a la hora de descargar todas las librerías necesarias, por lo que se optó por la instalación de un nuevo intérprete. El propio *Pytorch* indica en su propia página principal que este problema puede llegar a surgir en diversos equipos y es por lo que recomienda el uso de estas aplicaciones.

La versión del **lenguaje de programación Python** utilizado por este interpretador de Anaconda es la 3.08. Esta versión ofrece el buen uso de múltiples librerías con una buena relación sobre aspectos de Inteligencia artificial que veremos a continuación y que nos facilitarán la elaboración de nuestro motor de inteligencia artificial.

Principalmente *Python* destaca por fácil legibilidad y limpieza de sus códigos al ser un lenguaje de programación versátil multiplataforma y multiparadigma. Es ideal para trabajar con grandes volúmenes de datos, ya que como bien hemos dicho, al ser multiplataforma favorece la extracción y el procesamiento de los datos.

De ahí el uso de la librería de aprendizaje automático de código abierto *Pytorch* basada en la biblioteca de *Torch*. Esta es utilizada para aquellas aplicaciones que implementan cosas relacionadas con la visión artificial y procesamiento de lenguajes naturales.

Dentro del mundo de la informática, este lenguaje de programación es uno de los más usados y de los que mayor crecimiento tiene por las características nombradas anteriormente y por ser uno de los más fáciles y rápidos de entender, de ahí su elección, la curva de aprendizaje que nos proporciona nos permite conocer numerosas librerías y es uno de los más usados en aplicaciones de Deep Learning.

Como medio **de gestión de versiones** del proceso y de todos los archivos necesarios se hace el uso de la combinación de la plataforma de Google Drive y de los repositorios del propio equipo. Esto nos facilitará el hecho de tener diferentes versiones que nos proporcionan escalabilidad y seguridad dentro del proceso.

Además, esta gestión de versiones nos facilita la compartición de los elementos y archivos necesarios sin tener que estar reenviando diferentes versiones tras ofrecernos una sencilla organización de los archivos. Por último, además de llevar esta gestión de versiones se almacenará toda la información del proceso en una unidad extraíble aparte (USB) con el fin de mantener toda la seguridad posible sobre el proceso.

Una vez explicado el uso de todas estas aplicaciones anteriores, pasaremos a la parte más importante de este apartado de tecnologías que son el entorno de prototipado utilizado junto a las librerías correspondientes al desarrollo de nuestro motor de inteligencia.

- **Entorno de prototipado:** para este componente se usa la aplicación de Pycharm: el IDE de Python para programadores en la versión de 2021.1.3.

Este entorno nos ofrece dos ediciones diferentes para poder utilizar la aplicación. Estas son Professional o Community Edition, en el caso de ser estudiantes de la UMA, podemos utilizar ambas debido a que se nos ofrece licencia de la edición Profesional pero lo recomendable es usar la edición Community Edition, ya que es totalmente gratuita y nos ofrece todas las funcionalidades necesarias para la implementación de nuestro proyecto.

Entre las funcionalidades ofrecidas que nos servirán de ayuda podemos destacar la compatibilidad con VCS, soporte para base de datos y SQL, desarrollo web, depurador gráfico y ejecutor de pruebas, inspecciones de código, etc.

Más concretamente, este entorno nos permite ir elaborando prototipos de nuestro motor de inteligencia con el fin de poder determinar la mejor efectividad entre estos.

El entorno nos ofrece una interfaz bastante amigable que nos facilita bastante la programación, un editor de código inteligente para ayudarnos a mejorar la redacción para hacer que sea más fácil de leer, entender y a solucionar los posibles errores en él.

- **Librerías:** Para el uso de este componente nos basaremos en la librería de aprendizaje automático nombrada anteriormente *Pytorch* con la versión 1.9.0.

Esta librería está basada en la biblioteca de *Torch* y se ha utilizado por las razones como la interfaz tan sencilla que ofrece para la creación de redes neuronales sin tener que usar librería de nivel superior como puede ser *Keras* para *Tensor Flow*.

Pytorch trabaja con grafos dinámicos en vez de estáticos a diferencia de otros paquetes lo que nos permite ir modificando las funciones en tiempos de ejecución y hace que el cálculo varíe con ellas. Además, es un paquete de *Python* que haciendo uso de la programación de tensores está diseñado para realizar cálculos numéricos.

Al instalar esta librería se dispone de soporte para su ejecución en tarjetas gráficas (GPU), utiliza internamente CUDA, una API que conecta la CPU con la GPU, en

nuestro proyecto para su correcto funcionamiento hubo instalar previamente estas aplicaciones por medio de NVIDIA.

Dentro de los paquetes ofrecidos por *Pytorch* vamos a destacar los que son usados en este proyecto:

- **Nn:** El paquete `torch.nn` contiene multitud de clases que nos permiten crear redes neuronales y al mismo tiempo tener un gran nivel de control y detalle sobre estas. Una de las más importantes es la clase `torch.nn.Module`, esta es la clase básica que nos permitirá que todos los modelos hereden de ella.

Este paquete debe implementar los métodos de inicialización y computación de los datos. Además, nos ofrece la opción de seleccionar los tipo de neuronas, en nuestro `Linear`, y las funciones de activación que en nuestro caso serán `ReLU` y `Softplus`.

- **Tensor:** Con este paquete podemos crear las estructuras de tensores para poder almacenar los datos obtenidos.

- **Numpy:** Este paquete nos permite trabajar con vectores multidimensionales. La versión utilizada es la 1.20.1. Numpy nos ofrece un gran conjunto de operaciones matemáticas para trabajar con arrays y tensores.

- **Matplotlib:** Esta librería es utilizada para la representación de los datos en Python. Por medio de esta se crearán los gráficos necesarios para representar los datos obtenidos como puede ser la función de pérdida de cada una de las fases. La versión utilizada es la 3.4.2.

- **Pandas:** Este paquete nos proporciona los métodos necesarios para la carga de los datos de los diferentes conjuntos. Para ello utilizaremos la función `read_excel(file)`. La versión utilizada es la 1.2.4.

- **Utils.data:** A través de este paquete podemos almacenar los conjuntos de datos y trabajar sobre esto. Para ello nos ofrece las clases de `Dataset`, para guardar los datos y `DataLoader`, para poder iterar sobre estos que será necesario en las fases de entrenamiento, evaluación y validación.

- **Optim:** Por medio de este paquete usaremos el optimizador para entrenar a la red neuronal y modificar sus pesos.

6

Implementación

Este apartado se compone de todo el proceso de desarrollo del código de nuestro motor de inteligencia. Para esto nos basaremos en las redes neuronales que son modelos matemáticos y computacionales inspirados en sistemas biológicos, adaptados y simulados en computadoras; han surgido como un intento de desarrollar modelos que emulen características del comportamiento del cerebro humano (Menacho, 2014).

Las unidades básicas de estas redes son las neuronas que se organizan por capas conectadas entre sí. Normalmente suelen constar de tres capas principales, aunque en nuestro caso hay una capa más, podemos clasificarlo como tres capas, éstas son entrada, oculta y de salida.

Las redes neuronales tienen múltiples funcionalidades como procesamiento de datos, clasificación, regresión, etc. Además, cuenta con un gran marco de aplicaciones, concretamente en nuestro trabajo, el análisis y predicción por medio de series temporales de unos índices de vegetación extraídos de una plantación de tomate en Brasil.

La facilidad de aprendizaje, manejo e implementación proporcionados por la librería de Machine Learning de Pytorch junto a estos factores nombrados anteriormente, han sido determinantes para decidirnos y enfocarnos como medio de desarrollo de nuestro proyecto.

Este apartado se compondrá de las etapas establecidas en el apartado 3 de este proyecto, la metodología, y consta de: extracción de características y carga de datos, elección

del modelo, optimización de los parámetros, proceso de aprendizaje y entrenamiento, evaluación del proceso, validación del proceso.

6.1 Extracción de características y carga de datos

Esta etapa corresponde a una de las más extensas y complejas de este proyecto. Y es que el procesamiento de los datos es algo fundamental para poder desarrollar con éxito un motor de inteligencia que implemente un algoritmo de aprendizaje de este calibre.

Basándonos en nuestros datos, tenemos un conjunto principal que se trata de una serie temporal de índices de vegetación obtenidos en una plantación del cultivo del tomate tras la inyección del hongo *Phytophthora Infestans* con el objetivo principal de predecir el grado de severidad de este tras el paso de los días.

Este conjunto tuvo que ser reducido por una peculiaridad del hongo, y es que los índices obtenidos son desde el 28 de junio hasta el 8 de julio, ambos días inclusive, con otras dos mediciones entre ambos días. Es decir, de un mismo índice tenemos una serie temporal con cuatro valores de cuatro días distintos, pero por este patógeno puede llegar a darse que tras las primeras dos mediciones o tres el hongo no muestre mucha relevancia e implicación en la planta y partir de la cuarta tenga una gran variación considerable.

Esto provoca que nos centremos principalmente en la última medición realizada de los índices, es decir, la medición del 8 julio. Estableciendo así un período de tiempo considerable para que el hongo llegue a desarrollarse y podamos analizarlo de forma correcta. Además, solo teníamos como referencia la severidad del último día que había sido establecida por medio de especialistas a través de una escala determinada.

Una vez reducido el conjunto de los índices, se tuvo que realizar una normalización de los datos debido a que el rango de resultado de los índices era muy disperso. Esta normalización se hizo con el objetivo principal de facilitar el aprendizaje de la red neuronal, ya que será más efectivo su trabajo y se adaptará mejor a las distintas variaciones que pueda producir si todos los valores se encuentran comprendidos en un rango de 0 a 1.

Para ello se utilizará la siguiente función:

Función de Normalización

$$X_i = \frac{(X_i - \text{Min}(X))}{(\text{Max}(X) - \text{Min}(X))}$$

Figura 6.1. Definición de la función de normalización.

Como último paso con el conjunto de datos se realiza una división de este para tener un conjunto distinto para cada una de las etapas del proceso de aprendizaje : entrenamiento, evaluación y validación.

En la siguiente imagen podemos ver de una forma más visual el proceso llevado a cabo para el procesamiento de los datos.

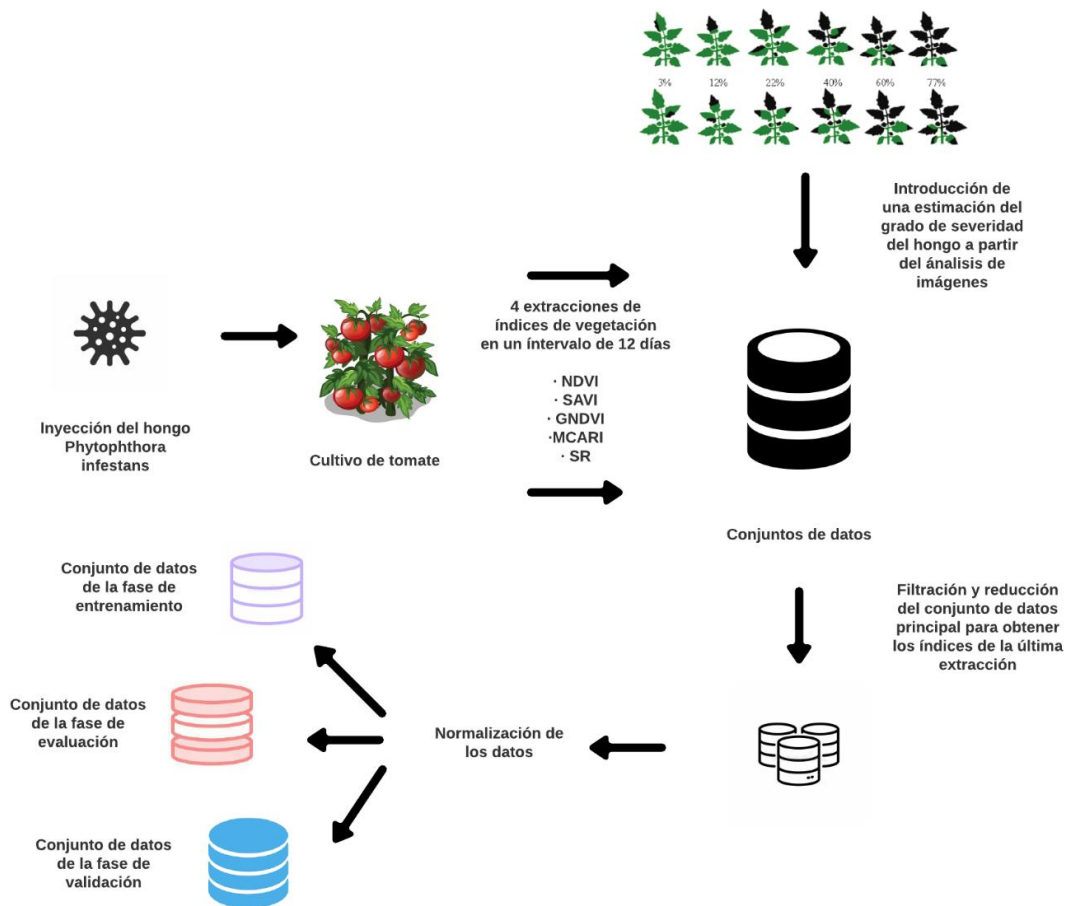


Figura 6.2. Diagrama del procesamiento de los datos

Después de haber realizado toda la extracción de características del conjunto y el procesamiento de los datos, pasamos a realizar la carga de los datos al entorno prototipado, PyCharm. Cabe destacar que el conjunto inicial de los datos era de formato .csv pero tras realizar las operaciones y divisiones correspondientes disponemos de tres conjuntos de datos de Excel con un formato .xlsx por lo que utilizaremos la función proporcionada por la librería de Pandas *read_excel*.

A continuación, en la siguiente imagen podemos visualizar la carga de los datos. Para la X utilizaremos los valores de los cuatro índices normalizados correspondientes al día 8 de julio y para la Y utilizaremos solo la columna del grado de severidad de ese mismo día, con esto nuestra trabajará con los índices con el objetivo de llegar a predecir la severidad establecida previamente.

```
# Creación de la clase de los conjuntos
class TomatoData(Dataset):
    def __init__(self, excel_file):
        data_excel = pd.read_excel(excel_file, 'Hoja1', engine='openpyxl')

        self.x = torch.tensor(data_excel[['NDVI_d08_Norm', 'GNDVI_d08_Norm', 'MCARI1_d08_Norm', 'SR_d08_Norm']].values,
                               dtype=torch.float32, requires_grad=True)

        self.y = torch.tensor(data_excel[['Severity (08 de Julio)']].values, dtype=torch.float32, requires_grad=True)

    def __len__(self):
        return len(self.y)

    def __getitem__(self, idx):
        if isinstance(idx, torch.Tensor):
            idx = idx.tolist()
        return [self.x.data[idx], self.y.data[idx]]
```

Figura 6.3. Creación de la clase TomatoData para la carga de los datos.

6.2 Elección del modelo de aprendizaje

El siguiente paso a seguir después de tener preprocesado y cargado el conjunto de datos sobre el que trabajaremos es elegir el modelo de aprendizaje de nuestro motor de inteligencia.

Para elección de nuestro modelo hemos nos basaremos en la tabla comparativa mostrada a continuación con el fin de saber que estructura neuronal se adecua mejor a nuestro motor. Antes de ello pasaremos a definir primeramente los dos tipos de redes a elegir.

En primer lugar, tenemos a la red que hemos elegido como estructura neuronal para nuestro trabajo, Feed-Forward; es una red neuronal artificial donde las conexiones producidas entre las neuronas no forman ciclos.

El motivo principal por el que hemos escogido esta arquitectura es que esta se adecuaba perfectamente al funcionamiento de nuestro trabajo ya que va a ser alimentada por un conjunto de datos, los índices de vegetación, en el que prácticamente ya conoce la salida que es el grado de severidad del hongo. Este tipo de red es bastante característica y conocida debido a que es una de las formas más simples, efectivas y fáciles de implementar.

Además de estos motivos, para la elección de este tipo de red también hemos tenido en cuenta que vamos a trabajar con una red multicapa, es decir, entre la entrada y salida puede llegar a existir diversas capas. Como bien hemos nombrado anteriormente, esta red neuronal será no recurrente, solo trabajaremos en un solo sentido y con una alta conectividad entre las neuronas.

Como siguiente tipo tenemos la red neuronal recurrente. Este tipo de red puede llegar a tener varias señales viajando en ambas direcciones llegando a introducir bucles en la red. Estas redes son bastantes potentes pero al mismo tiempo bastante complejas.

De forma más visual, en la siguiente imagen podemos diferenciar este tipo de redes neuronales.

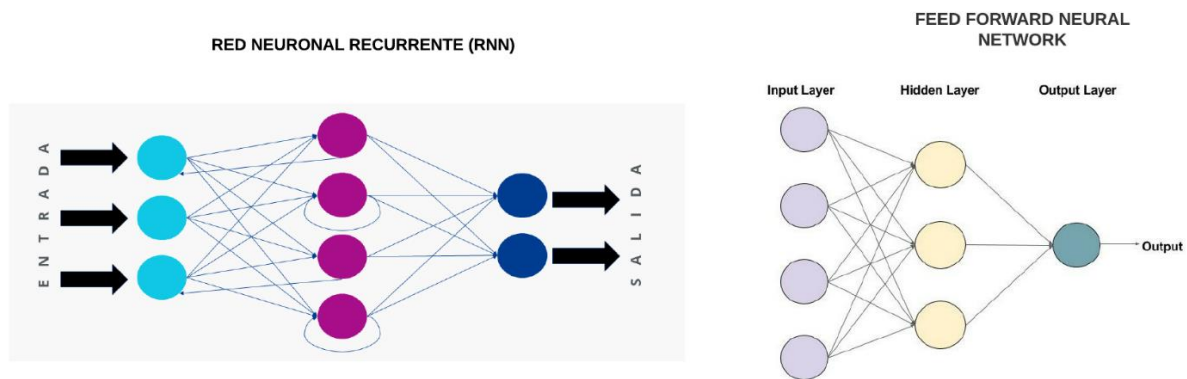


Figura 6.4. Imagen comparativa de los tipos de modelos de aprendizaje.

Tras la visualización de la imagen anterior y junto a los motivos nombrados anteriormente en la siguiente tabla mostraremos las principales diferencias que nos han hecho decantarnos por la primera de ellas entre estos dos tipos de redes.

Feed-Forward	Recurrentes
<ul style="list-style-type: none"> • Unión completa y en un solo sentido de las neuronas entre cada capa de la red. 	<ul style="list-style-type: none"> • Unión completa y recurrente entre las neuronas llegando a crear bucles entre capas.
<ul style="list-style-type: none"> • Gran número de neuronas necesarias y considerable complejidad computacional. 	<ul style="list-style-type: none"> • Gran número de neuronas necesarias y alta complejidad computacional.
<ul style="list-style-type: none"> • La alta conectividad entre las neuronas hace que los resultados puedan llegar a ser los esperados. 	<ul style="list-style-type: none"> • Debido a su alta conectividad y recurrencia provoca que los resultados tengan más probabilidades de ser los esperados.

<ul style="list-style-type: none"> • Redes sencillas y eficientes utilizadas para series temporales de menor tamaño. 	<ul style="list-style-type: none"> • Redes complejas y poderosas utilizadas para series temporales de gran tamaño.
---	---

Figura 6.5. Tabla comparativa de las diferencias entre los tipos de redes neuronales a utilizar.

El desarrollo de esta red neuronal muestra varias peculiaridades en relación al desarrollo de otras redes en diferentes problemas. Y es que para esta red se han definido tres transformaciones lineales debido a que nuestro proyecto contará con dos capas ocultas en vez de con una como suele ser normalmente, es decir, constará de una transformación lineal entre las dos capas ocultas.

Aunque existe un consenso en el que la diferencia de rendimiento al añadir una capa oculta más es mínimo, para nuestro problema aumenta considerablemente al obtener bastantes mejores resultados con una capa oculta extra, he de ahí el motivo por el que realizamos tres transformaciones lineales en vez de dos.

Las otras dos transformaciones lineales serán para los datos recibidos por entrada y para producir la salida de la última capa.

Cabe destacar que debido a nuestro sofisticado modelo, el hecho de añadir una capa oculta más aparte de aumentar el rendimiento también aumentarán recursos como tiempo y poder computacional.

Además, como funciones de activación de las capas, se eligen las funciones ReLU (Rectificador Lineal Unario) y Softplus. La primera, ReLU, es la más utilizada debido a que permite un rápido aprendizaje en las redes neuronales. Esta nos elimina los valores negativos y nos permite hacer predicciones de datos continuos. La segunda, Softplus, es la función continua y derivable de la anterior. La principal diferencia es que es más suavizada que la anterior, de ahí su elección, obteniendo mejores resultados.

Aunque en varios problemas se suele utilizar una fórmula para determinar el número de neuronas de la capa oculta por medio del número de valores de entrada, debido a la dificultad de nuestra red en abordar este problema por su complejidad hemos utilizado un número bastante alto.

A continuación, en la siguiente imagen podemos visualizar una representación abstracta de nuestra red neuronal con su correspondiente división entre capas.

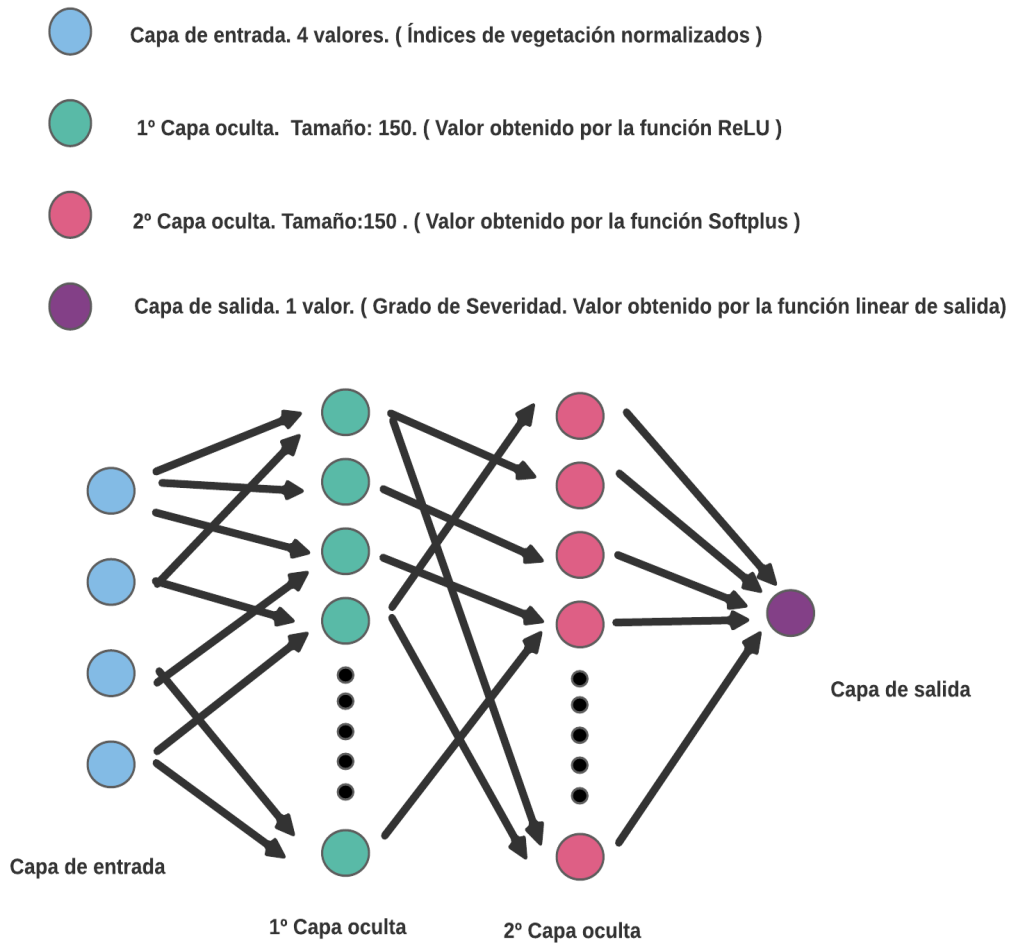


Figura 6.6. Diagrama de la red neuronal desarrollada.

Posteriormente, utilizando esta representación visual solo nos quedaría redefinir los siguientes métodos de nuestra red en nuestro motor de inteligencia:

- **__init__()** : En este método inicializamos las funciones de activación (Linear, ReLU y Softplus) proporcionadas por la librería torch.nn. A este método se le pasarán como parámetros self (parámetro especial que siempre se utilizará como una referencia al objeto mismo), input (valores de entrada), hidden (tamaño de la capa oculta) y nclasses (valor de salida).
- **forward()**: Este método es el encargado de calcular el valor de salida de la red por medio de los valores de entrada, concretamente, esta clase visualiza el paso que tienen que hacer los datos en el diagrama mostrado anteriormente, con sus

correspondientes funciones de activación. A este método se pasa como parámetros self y x (conjunto de datos).

En la siguiente imagen se puede ver el resultado final de la creación de la red neuronal en el entorno de prototipado.

```
# Creación de la Neural Network

class NeuralNetwork(nn.Module):
    def __init__(self, input_parameters, hidden, nclasses):
        super(NeuralNetwork, self).__init__()
        self.l1 = nn.Linear(input_parameters, hidden)
        self.relu = nn.ReLU()
        self.l2 = nn.Linear(hidden, hidden)
        self.l3 = nn.Linear(hidden, nclasses)
        self.soft = nn.Softplus()

    def forward(self, x_data):
        out = self.l1(x_data)
        out = self.relu(out)
        out = self.l2(out)
        out = self.soft(out)
        out = self.l3(out)

        return out
```

Figura 6.7. Creación y configuración de la red neuronal.

6.3 Optimización de parámetros

Las decisiones tomadas en esta etapa son fundamentales en la eficacia de nuestro proyecto y es que el algoritmo de optimización elegido en nuestra red juega un papel determinante debido a que actualiza los parámetros de peso para minimizar el valor de la función de pérdida.

Para la elección de este algoritmo se han tenido en cuenta varios experimentos realizados en la red y además se han realizado pruebas propias para ver qué algoritmo nos proporciona mejores resultados.

Además de estas pruebas realizadas con el optimizador, al final del subapartado siguiente (proceso de aprendizaje) se mostrará una tabla con los resultados obtenidos tras haber realizado pruebas con distintos parámetros y funciones.

La elección de este algoritmo ha estado condicionada por la comparación de los resultados obtenidos con dos tipos de optimizadores. Entre ellos tenemos al algoritmo de Adam que combina las bondades de RMSProp y AdaGrad. Adam está diseñado para acelerar el proceso de optimización, esto se logra calculando un factor de entrenamiento para cada parámetro de entrada que se optimiza. Y el algoritmo de RMSProp (Root Mean Square Propagation), este mantiene un factor de entrenamiento diferente para cada dimensión en dónde el escalado del factor de entrenamiento se calcula por la división por la media del declive exponencial del cuadrado de los gradientes.

Para la comparación de estos dos algoritmos nos hemos basado en el resultado de la función de pérdida de la primera época y en la pérdida media de la fase de entrenamiento llegando a obtener los siguientes valores:

- **Adam**

```
----- Fase de entrenamiento -----  
  
Pérdida media por Epoch 1 : 20.848714476021435
```

Figura 6.8. Resultado de la primera época en la fase de entrenamiento con el algoritmo Adam.

```
Pérdida media en la fase de entrenamiento con el algoritmo de optimización Adam : 8.555915784433914
```

Figura 6.9. Resultado de la pérdida media en la fase de entrenamiento con el algoritmo Adam.

- **RMSProp**

```
----- Fase de entrenamiento -----  
  
Pérdida media por Epoch 1 : 16.35396874180207
```

Figura 6.10. Resultado de la primera época en la fase de entrenamiento con el algoritmo RMSProp.

```
Perdida media en la fase de entrenamiento con el algoritmo de optimización RMSProp : 8.182771609952635
```

Figura 6.11. Resultado de la pérdida media en la fase de entrenamiento con el algoritmo RMSProp.

Como hemos podido observar en las imágenes de la comparación de los algoritmos con el segundo de ellos, RMSProp, llegamos a obtener mejores resultados, sobre todo en la primera época de la fase de entrenamiento. Por ello hemos decidido utilizar este algoritmo en nuestro motor de inteligencia.

A continuación, podemos ver la declaración del algoritmo en el entorno:

```
optimizer = optim.RMSprop(model.parameters(), lr=learning_rate)
```

Figura 6.12. Declaración de la función de optimización

6.4 Proceso de aprendizaje y entrenamiento

Esta etapa constituye todo el proceso de aprendizaje de nuestro motor de inteligencia. Para ello, le suministraremos toda la información necesaria y utilizaremos ciertos parámetros y funciones, para que este motor pueda llegar a adquirir y a utilizar todo el conocimiento posible sobre los datos de entrada y de la red con el objetivo principal de poder llegar a realizar buenas predicciones en las fases posteriores.

Dentro de la división del conjunto de datos principal el desarrollo de esta etapa utilizará el conjunto más grande de datos como podemos ver en la siguiente imagen, que equivale al 60% de los datos totales.

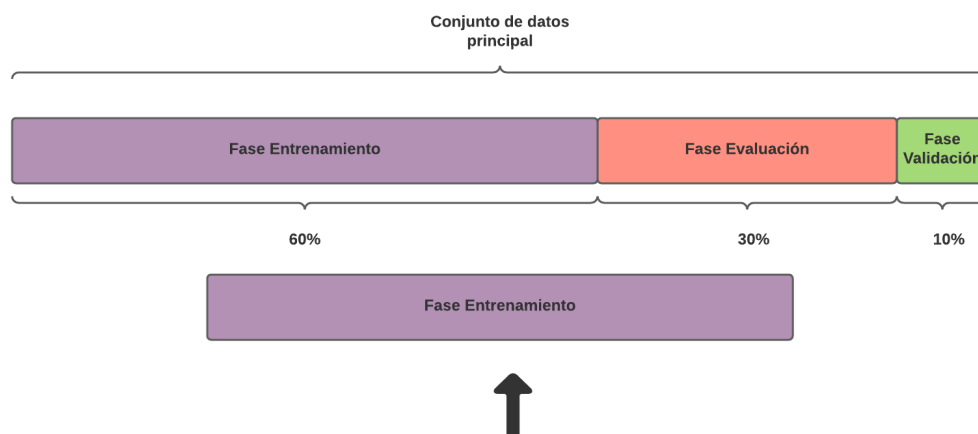


Figura 6.13. Diagrama de la distribución del conjunto de datos utilizado en la fase de entrenamiento.

Respecto a la hora de elegir nuestra función de pérdida seguiremos un procedimiento parecido a la etapa anterior, elección de la función de optimización, y es que al ser un problema de regresión nos hemos basado en aquellas funciones más utilizadas y con mejores resultados en estos problemas.

Esta función tiene como objetivo calcular la diferencia entre el valor de la etiqueta y el valor predicho, es decir, la diferencia entre el valor al que tenemos que llegar y el valor que produce nuestra red. Además, nos hará ver si nuestra red va realmente por el buen camino.

Las funciones de pérdida más utilizadas para los problemas de regresión son la L1Loss y la MSELoss. Por los resultados obtenidos utilizaremos la primera que tiene un método de cálculo muy simple ya que simplemente toma el promedio del error absoluto entre el valor predicho y el valor real.

La comparación de estas dos funciones también se verá reflejada en la tabla nombrada anteriormente.

A continuación, podemos ver la declaración de la función de pérdida en el entorno de prototipado:

```
loss_function = nn.L1Loss()
```

Figura 6.14. Declaración de la función de pérdida

Tras tener claro cómo va a estar estructura la red neuronal debemos establecer los valores de los hiperparámetros, éstos son una serie de valores que determinan el comportamiento, funcionamiento y precisión para nuestros conjuntos de datos. Para nuestra red utilizaremos los siguientes:

- **input_size:** Con este parámetro nos referimos al número de valores de entrada que tendrá la red neuronal. En nuestro caso su valor será 4, por los cuatro índices de vegetación correspondientes al último día.
- **batch_size:** Este parámetro determina el número de muestras de cada lote que se irán propagando. Para esta red el valor será 1 para ir iterando por cada una de las muestras del conjunto de datos.
- **hidden_size:** Este valor constituye al número de neuronas que tendrá la capa oculta. Su valor es de 150.

- **learning_rate:** Es un parámetro de ajuste para el algoritmo de optimización y determina la frecuencia con la que variarán los pesos de las neuronas. Su valor será 0.005.
- **num_epochs:** Este parámetro consiste en el número de iteraciones que realizará la red neuronal en la etapa de entrenamiento. Nuestra red realizará 25 iteraciones.
- **num_classes:** Con este parámetro establecemos el número de valores de salida que tendrá la red neuronal. En nuestro caso su valor será 1 que corresponderá al grado de severidad predicho.

Para la decisión de estos hiperparámetros hemos realizado varias pruebas hasta ver qué tipo de valores se ajusta más a la red neuronal y con los que obtenemos mejores resultados.

Una vez elegida la función de pérdida y establecidos los valores de los hiperparámetros, pasamos a realizar la fase de entrenamiento de la red neuronal (proceso de aprendizaje).

Para cada época se calculará el valor predicho por las funciones de activación tras el paso por la red neuronal (forward pass). A continuación, calculamos el error por la función de pérdida y la diferencia de este valor y el valor predicho será almacenada y acumulada para calcular la pérdida media por cada época y para la fase de entrenamiento. La última parte de esta etapa consiste en propagar el error de la red (backpropagation) y realizar la optimización de los parámetros.

En la siguiente imagen podemos ver la declaración de los pasos descritos anteriormente en el motor de inteligencia.

```

# Fase de Entrenamiento
print("\n----- Fase de entrenamiento -----\n")
for epoch in range(num_epochs):
    diferenciaDeError = 0
    for i, (data, label) in enumerate(training_load):

        # Forward pass
        outputs = model(data)

        # Cálculo del error
        loss = loss_function(outputs, label)

        # Backpropagation
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    diferenciaDeError = loss.item() + diferenciaDeError

    val_loss.append(diferenciaDeError / len(training_load))
    num_epochs_val.append(epoch + 1)
    mediaTraining = mediaTraining + (diferenciaDeError / len(training_load))
    print(f'\nPérdida media por Epoch {epoch + 1} : {diferenciaDeError / len(training_load)} \n')

print(f'\nPérdida media en la fase de entrenamiento : {mediaTraining / num_epochs} \n')

```

Figura 6.15. Declaración de la fase de entrenamiento

En la siguiente tabla podemos comparar los resultados obtenidos en las pruebas realizadas con diferentes configuraciones en la fase de entrenamiento:

Hidden_size	learning_rate	Num_epochs	Funciones de Activación	F. de pérdida	Optimizador	Pérdida media Fase Entrenamiento
150	0.005	25	Linear ReLU Softplus	MSELoss	RMSprop	115.567
200	0.005	25	Linear Softplus	L1Loss	RMSprop	11.326
100	0.001	25	Linear ReLU Softplus	L1Loss	RMSprop	11.309
150	0.005	10	Linear ReLU Softplus	L1Loss	RMSprop	9.525
150	0.005	25	Linear ReLU Softplus	L1Loss	Adam	8.727

150	0.005	25	Linear ReLU	L1Loss	RMSprop	8.611
100	0.005	25	Linear ReLU Softplus	L1Loss	RMSprop	8.41
200	0.005	25	Linear ReLU Softplus	L1Loss	RMSprop	8.362
150	0.005	25	Linear ReLU Softplus	L1Loss	RMSprop	8.088

Figura 6.16. Resultados obtenidos de la pérdida media en la fase de entrenamiento con diferentes configuraciones (hiperparámetros y funciones).

Tras la visualización de los datos mostrados en la tabla anterior hemos podido decantarnos por los valores de los parámetros a utilizar y de las funciones para tener el mejor desarrollo posible en nuestro motor de inteligencia.

El desarrollo y selección de estos parámetros ha estado basado principalmente en la exploración de todas las alternativas posibles y ver cuál es la que nos ofrece un mejor funcionamiento.

Cabe destacar que no se han incluido los resultados de aquellas redes de una sola capa oculta debido a sus grandes valores de pérdida y mala eficiencia.

6.5 Evaluación del proceso

El paso a seguir después de realizar el entrenamiento de la red neuronal consiste en realizar una evaluación para ver si esta logra y tiene la capacidad para realizar buenas predicciones.

Para esta etapa se utilizará un conjunto diferente y reducido como podremos apreciar en la imagen que aparece a continuación, ya que se supone que la red tiene el conocimiento y la preparación suficiente para enfrentarse a otros datos. Este conjunto constituirá el 30% de los datos totales.

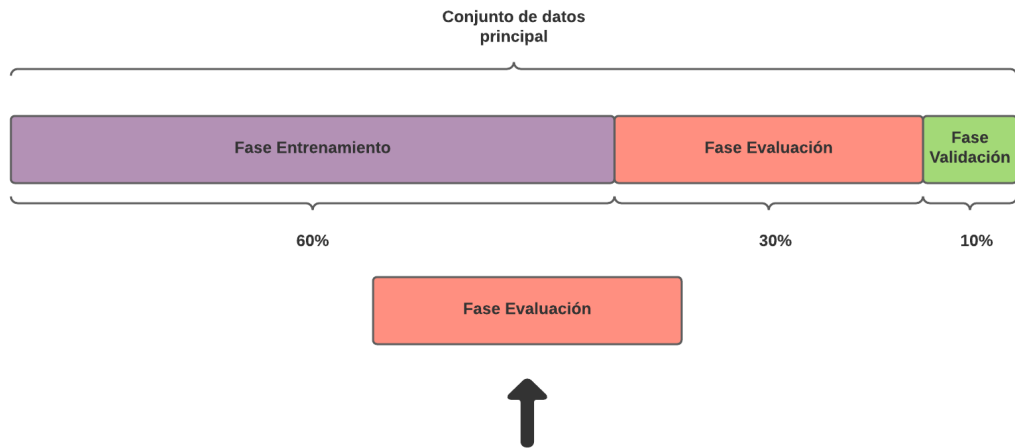


Figura 6.17. Diagrama de la distribución del conjunto de datos utilizado en la fase de evaluación.

Como principal diferencia entre esta fase y la anterior, podemos decir que ya no es necesario propagar el error de la red debido a que se supone que el algoritmo está lo suficientemente preparado para enfrentarse a un conjunto de datos diferente.

Esta evaluación será determinada por la pérdida media de los valores a predecir y los predichos del conjunto de datos. Este valor nos ayudará a determinar qué precisión tiene el algoritmo desarrollado para predecir el grado de severidad de este hongo.

La siguiente imagen nos mostrará la fase de evaluación establecida para la red neuronal.

```
# Fase de Evaluación
with torch.no_grad():
    n_samples = 0
    diferenciaDeError = 0

    for i, (data, labels) in enumerate(test_load):
        # Predict
        outputs = model(data)

        loss = loss_function(outputs, labels)

        n_samples += 1
        diferenciaDeError = diferenciaDeError + loss

    print("\n-----Fase de Evaluación-----\n")
    print(f"Error cuadrático medio de la fase evaluación : {diferenciaDeError / n_samples}")
```

Figura 6.18. Declaración de la fase de evaluación

6.6 Validación del proceso

Finalmente, tenemos la etapa de validación, que corresponde a la etapa final de nuestro motor de inteligencia y es la que determinará la validez, como bien su nombre indica, de esta herramienta trabajando con un conjunto de datos que no conoce la red.

Para esta fase se lleva a cabo un desarrollo que es prácticamente igual a la etapa de evaluación, etapa anterior, en donde la única diferencia como se ha dicho anteriormente es que el conjunto tratado aparte de ser un poco más reducido es totalmente nuevo. En la imagen mostrada posteriormente podemos visualizar su distribución.

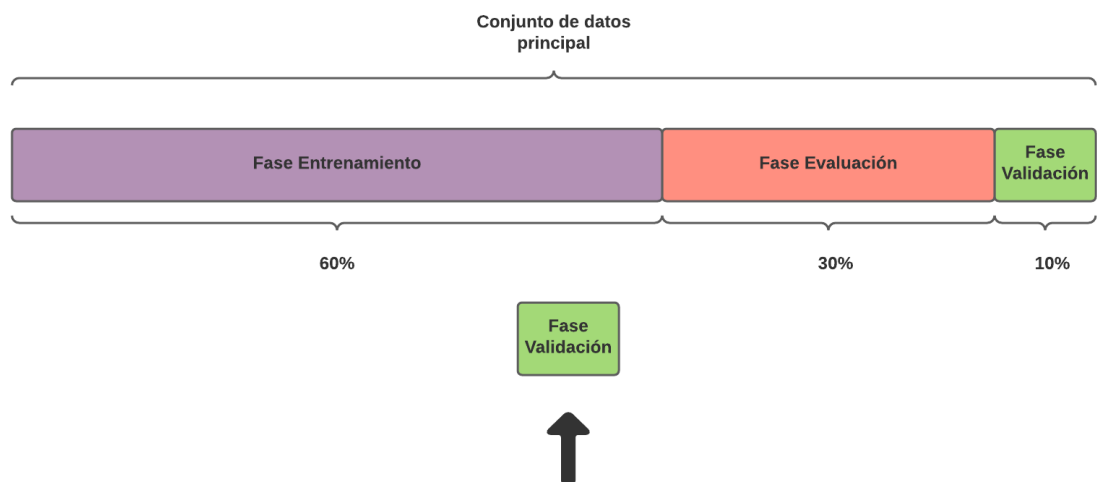


Figura 6.19. Diagrama de la distribución del conjunto de datos utilizado en la fase de validación.

Por medio de este proceso de validación conseguiremos garantizar que la implementación ha sido bastante buena y realizada con éxito. Además, nos indicará que el motor está totalmente preparado para usarse en un caso del mundo real.

El hecho de que el conjunto usado en esta fase sea totalmente nuevo es lo que asegura que nuestro modelo cumple todos los requisitos especificados con éxito.

En la imagen mostrada a continuación podemos ver cómo ha sido la declaración de esta etapa en el motor de inteligencia.

```

# Fase de Validación
with torch.no_grad():
    n_correct_val = 0
    n_samples_val = 0
    diferenciaDeError = 0

    for i, (data, labels) in enumerate(validation_load):
        # Predict
        outputs = model(data)

        loss = loss_function(outputs, labels)

        diferenciaDeError = diferenciaDeError + loss

        n_samples_val += 1

print("\n-----Fase de Validación-----\n")
print(f"Error cuadrático medio de la fase validación : {diferenciaDeError / n_samples_val}")

```

Figura 6.20. Declaración de la fase de validación

7

Resultados

El resultado principal y final de este proyecto es elaborar una herramienta que consiga estimar el grado de severidad del hongo *Phytophthora infestans* en el cultivo del tomate, teniendo en cuenta una serie de índices de vegetación correspondientes a estas plantas tras la implantación de este patógeno.

Para lograr este objetivo se ha desarrollado un motor de inteligencia que implementa algoritmos de Machine Learning y Deep Learning. Y tras todo el proceso de implantación explicado se logra el objetivo esperado de este proyecto.

7.1 Resultados fase de entrenamiento

Para esta fase se estiman como principales objetivos el desarrollo y aprendizaje de este motor de inteligencia, en dónde en esta etapa se tiene que dotar a la herramienta con un aprendizaje para que esta tenga la capacidad de saber interpretar y manipular estos datos con el fin de poder lograr estimar un buen grado de severidad del hongo.

Los valores de pérdida obtenidos por cada época de esta fase nos servirán de ayuda para indicar que nuestro algoritmo va por el buen camino debido a que en el caso de obtener de forma progresiva uno valores de pérdida cada vez más pequeños estaremos logrando el objetivo y resultado esperado.

Para ello, en la siguiente gráfica se muestra la evolución de la tasa de pérdida obtenida en esta fase de entrenamiento.

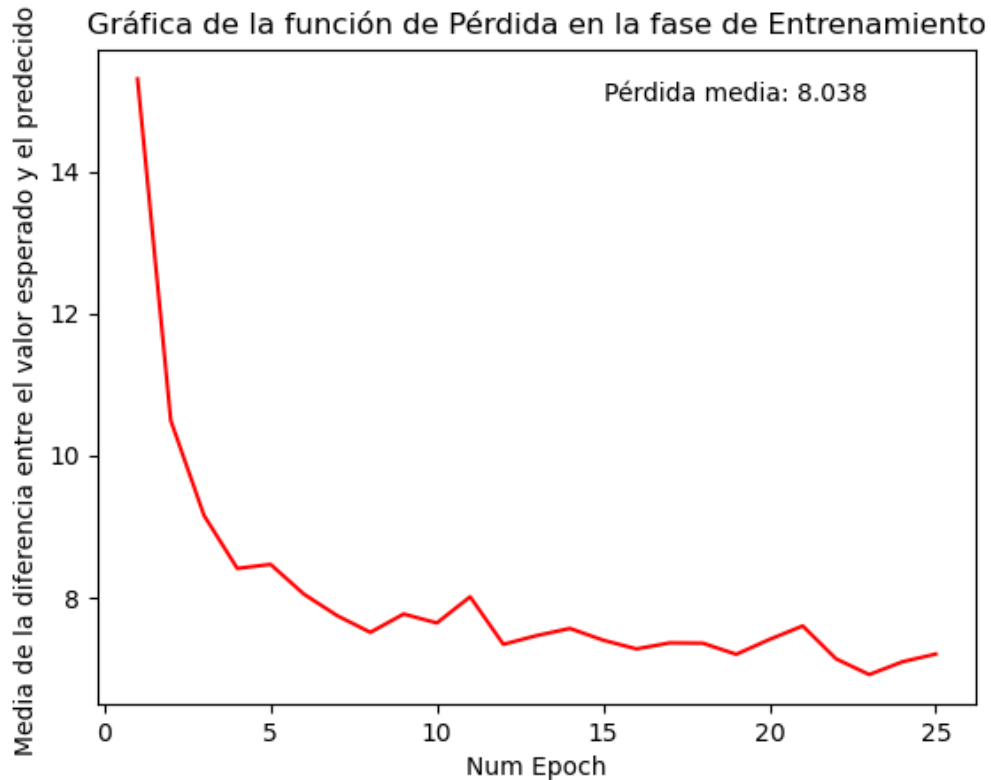


Figura 7.1. Gráfica que muestra la evolución de la tasa de pérdida en la fase de entrenamiento.

En esta gráfica, como bien hemos indicado, podemos visualizar la evolución de la pérdida media de cada época en la fase de entrenamiento. En dónde el eje de la Y muestra los valores obtenidos por cada época y el eje de la X indica el número de las épocas.

Tras el análisis de esta gráfica podemos ver que el valor de la función de pérdida tiende a descender tras el paso de las épocas hasta lograr un resultado aceptable, asegurándonos así que la etapa de aprendizaje ha sido correcta y nuestra herramienta está totalmente preparada para pasar a la siguiente fase. Obteniéndose como valor medio de pérdida en la fase de 8.038.

Un valor de pérdida menor que 10 será estimado como un buen valor en el funcionamiento de este motor de inteligencia.

7.2 Resultados fase de test

Los resultados obtenidos a partir del conjunto proporcionado en esta etapa se utilizarán con la finalidad de comprobar si nuestro motor de inteligencia artificial es capaz de realizar buenas predicciones.

Para ello, se iterará el conjunto de test y se realizará una evaluación para ver si el algoritmo realiza su función de forma satisfactoria. La evaluación de estos resultados obtenidos se realizará de las siguientes dos formas.

En primer lugar, analizaremos la evolución de la función de pérdida para esta fase en la siguiente imagen.

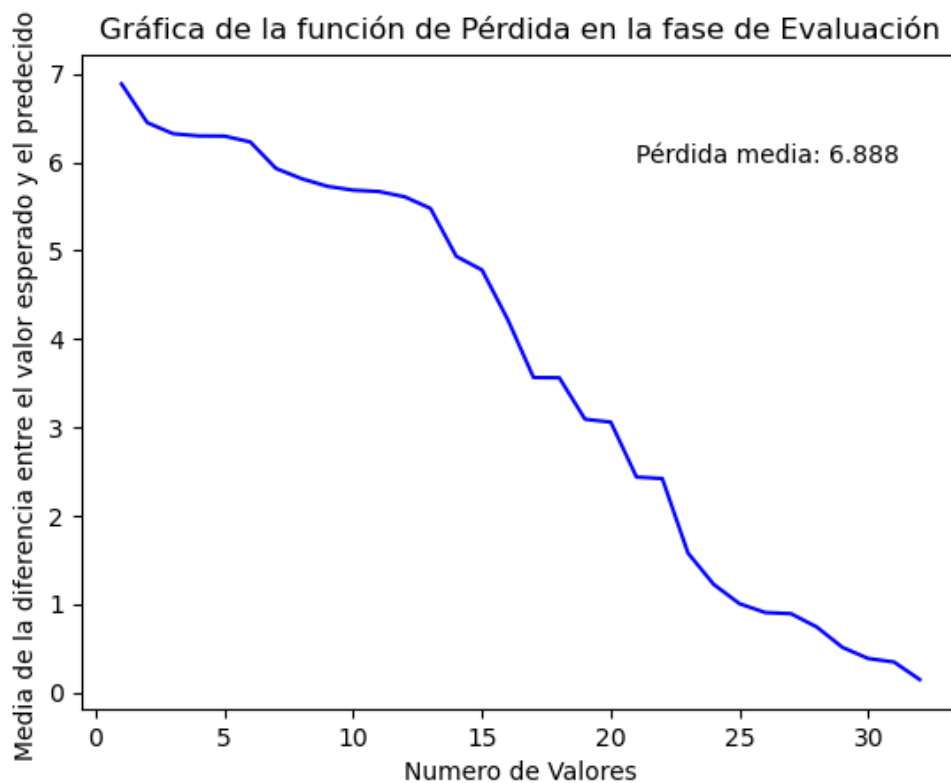


Figura 7.2. Gráfica que muestra la evolución de la tasa de pérdida en la fase de evaluación.

En relación a la gráfica mostrada en la fase de entrenamiento solo hay una única diferencia con esta, y es que en el eje X en lugar de estar el número de épocas sobre las que se iteran los valores, en esta gráfica tenemos simplemente a los valores.

En la visualización de esta gráfica podemos seguir viendo la tendencia descendente de la función de pérdida y un valor de pérdida menor que en la fase anterior de 6.899 lo que llega a indicar que la evaluación tiene un correcto funcionamiento.

Como segunda forma de analizar esta fase de evaluación tenemos la siguiente gráfica que muestra una dispersión entre los datos a predecir (color azul) y los obtenidos (color rojo).

Gráfica de dispersión de los datos por predecir y predichos Fase Evaluación

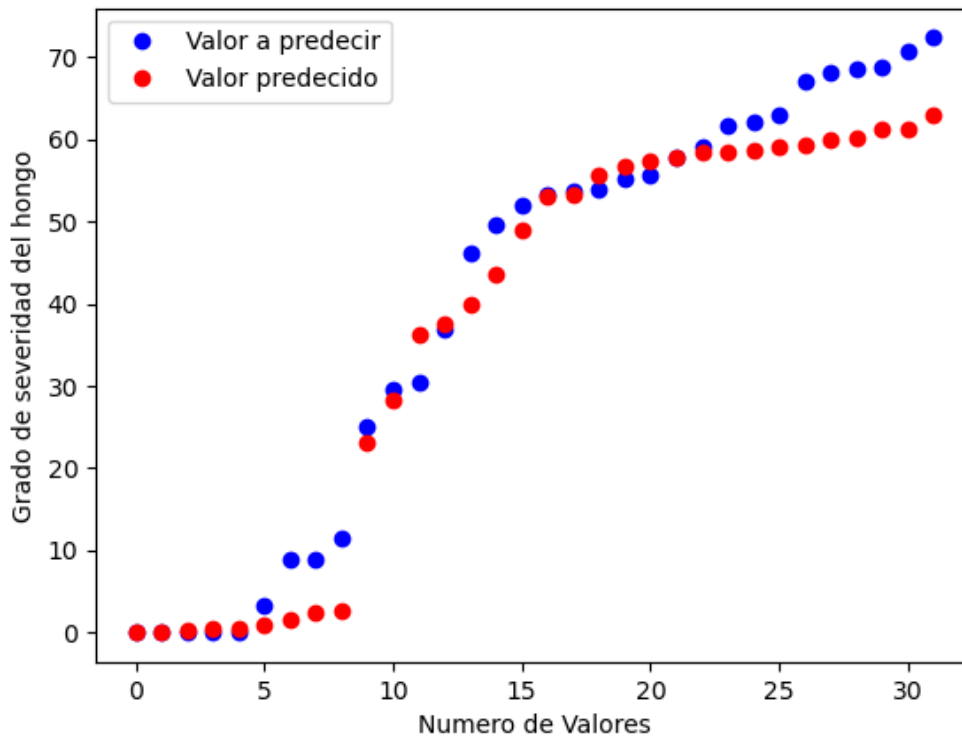


Figura 7.3. Gráfica que muestra la dispersión de los datos a predecir y los predichos en la fase de evaluación.

En esta gráfica en el eje de la X tenemos el número de valores como en la anterior, pero en el eje de la Y tenemos el grado de severidad del hongo. Tras su visualización podemos objetar que nuestro modelo tiene menor tendencia a fallar más en aquellos valores en los que el hongo tenga un grado de severidad bastante alto. En cambio, para aquellos valores situados por debajo de 60 vemos que nuestro modelo prácticamente no falla de una forma tan destacada como en algunos valores por encima de este umbral.

7.3 Resultados fase de validación

Finalmente, en la última etapa de este proceso se analizarán los resultados obtenidos siguiendo el mismo procedimiento que la fase anterior pero con la principal diferencia que se ha ido explicando a lo largo del trabajo y es que el conjunto de datos sobre el que operará esta fase es totalmente nuevo.

Tras el análisis de los resultados de este conjunto podremos garantizar que la implementación se ha llevado a cabo con éxito y que nuestro motor está completamente preparado para pasar a casos del mundo real.

En este análisis tenemos en primer lugar la gráfica de la evolución de pérdida obtenida para esta fase de validación que podemos visualizar en la siguiente imagen.

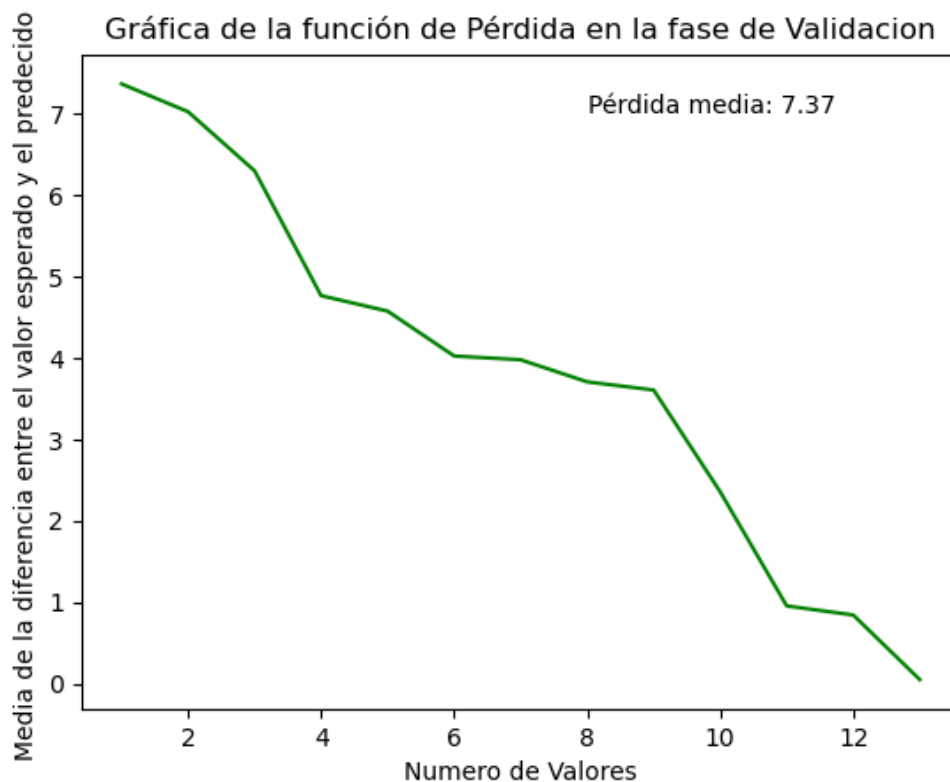


Figura 7.4. Gráfica que muestra la evolución de la tasa de pérdida en la fase de validación.

Por medio de esta gráfica podemos ver que nuestro responde ante este nuevo conjunto de datos debido a que sigue mostrando esa tendencia decreciente que hemos visto en las dos fases anteriores. Aunque el valor de la pérdida media de esta fase es un poco más

elevado que en la fase de validación obtener un resultado de 7.37 sigue siendo muy buen valor para el funcionamiento de esta herramienta.

A continuación, tras analizar la función de pérdida también analizaremos la dispersión de los valores obtenidos frente a los que hay que predecir. Los datos obtenidos por nuestra herramienta se muestra por medio del color verde mientras que los datos a predecir de amarillo como podemos ver en la siguiente imagen.

Gráfica de dispersión de los datos por predecir y predichos Fase Validación

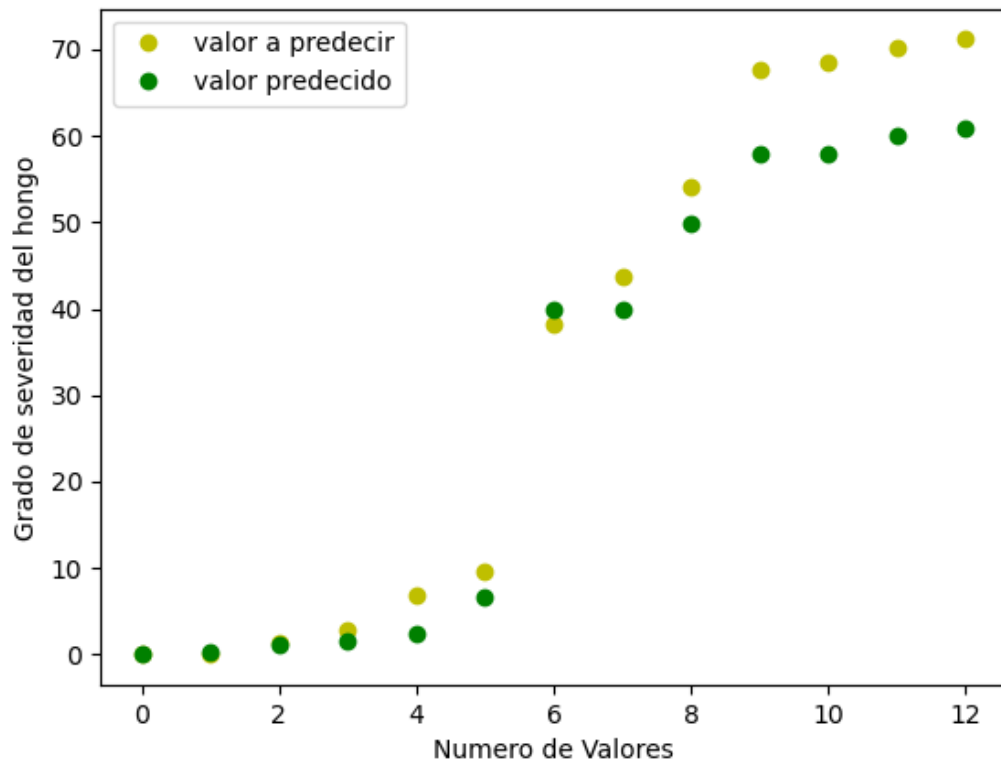


Figura 7.5. Gráfica que muestra la dispersión de los datos a predecir y los predichos en la fase de valuación.

En el análisis de los resultados obtenidos en esta gráfica afianzamos la suposición dictaminada en la anterior fase y es que la herramienta muestra una gran fiabilidad hasta valores de 50 en el grado de severidad, a partir de este umbral tiende a quedarse corta con estos valores.

Finalmente, con el objetivo de poder llegar a clasificar las predicciones realizadas como correctamente clasificadas o incorrectas se establecerá el siguiente umbral:

Aquellas predicciones que al redondearse o truncarse a la decena sean iguales al valor a predecir con las mismas operaciones realizadas se clasificarán como buenas predicciones.

En la gráfica mostrada a continuación podemos visualizar la clasificación realizada entre las predicciones obtenidas.

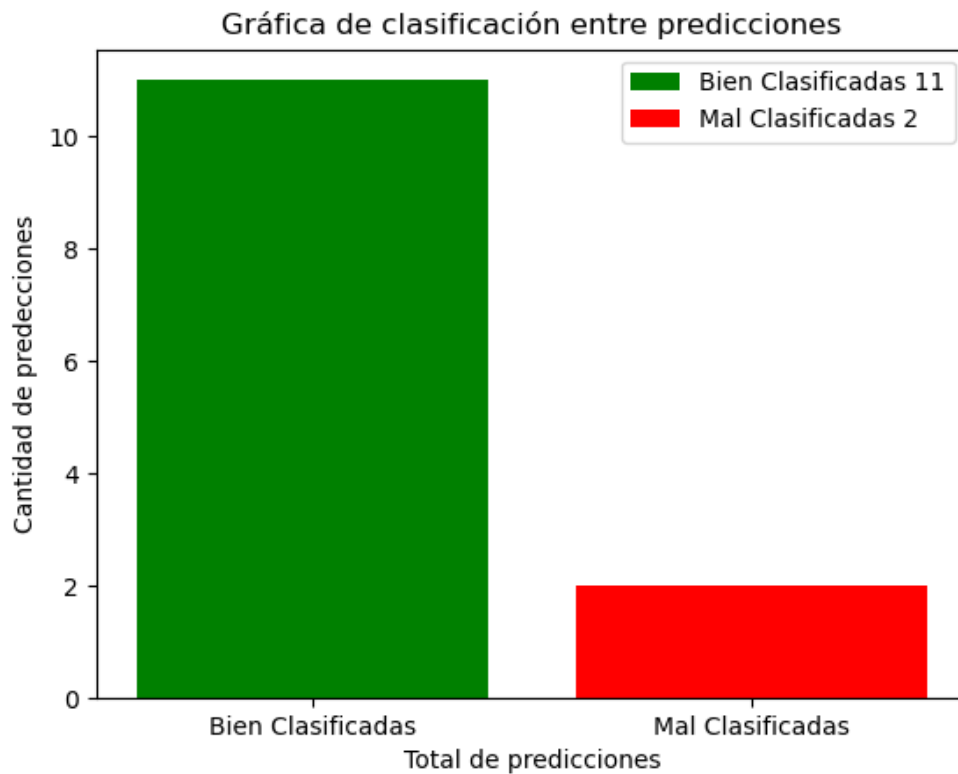


Figura 7.6. Gráfica que muestra la clasificación entre las predicciones realizadas.

Después de analizar los resultados en esta última gráfica podemos ver el buen rendimiento de nuestra herramienta debido al alto número de predicciones bien clasificadas. Además, estos resultados nos servirán de ayuda para determinar la efectividad de nuestro motor de inteligencia desarrollado como podemos ver en la siguiente imagen.

```
-----Fase de Validación-----  
  
Número predicciones bien clasificadas del total : 11 / 13  
Precisión de la fase de validación : 91.67
```

Figura 7.7. Resultados obtenidos en la fase de validación.

En conclusión, tras la visualización y análisis de los resultados obtenidos, principalmente en esta parte, pero teniendo en cuenta los de las fases anteriores podemos concluir que la implementación de la red de la neuronal Feed Forward junto a la función de pérdida, el algoritmo de optimización y la elección de los hiperparámetros ha sido un éxito. Debido a que el valor de pérdida media obtenido de este nuevo conjunto nos indica que las estimaciones son bastante próximas y fiables, lo que conlleva a que la implementación haya ido como se esperaba cumpliendo todas las expectativas establecidas.

8

Conclusiones y líneas futuras

8.1 Conclusiones

Las plagas y enfermedades de las plantas son una de las principales amenazas de los cultivos alimentarios llegando a provocar pérdidas significativas para los agricultores y repercutiendo de forma devastadora en la seguridad alimentaria.

La propagación de estas en las plantas ha aumentado de forma drástica en los últimos años. Entre los factores que llegan a influir en esta propagación podemos destacar la globalización, el comercio y el cambio climático y la falta de capacidad de los sistemas de reproducción entre otros. Estas amenazas tienen una gran facilidad para propagarse en varios países llegando a alcanzar dimensiones de epidemias.

Debido a mi vinculación con esta área junto a estas grandes causas incentivaron en mí el interés de la investigación y el desarrollo de una herramienta que agilice, resuelva y ayude en problemas existentes en la agricultura.

Por ello, tras establecerse como objetivo principal en el anteproyecto del trabajo se ha realizado la implementación de un motor de inteligencia artificial. Este nos permite determinar el grado de severidad del hongo *Phytophthora Infestans* en el cultivo del tomate.

A través de este documento se ha podido ver cómo ha sido la investigación y el desarrollo completo llevado a cabo de este motor desde sus fases iniciales como la extracción de características y carga de datos hasta finalmente su validación.

Afianzando los resultados y conclusiones obtenidos por nuestra herramienta cabe destacar una peculiaridad sobre el estudio y análisis de este hongo. Y es que los encargados de este conjunto de datos establecieron un umbral en el grado del hongo. Este dictamina que aquellas plantas que tengan un grado inferior a 40 se clasificarán como aquellas que han tenido la capacidad y fuerza de combatir este hongo, las que no serán clasificadas por débiles ante el patógeno.

Como se ha podido comprobar en los resultados obtenidos y mostrados previamente nuestra herramienta demuestra un alto grado de fiabilidad para establecer este umbral como punto de partida de clasificación llegando a obtener un porcentaje del 90% de éxito en las predicciones realizadas.

Esta fiabilidad nos permite determinar qué tipo de acciones tomar en aquellas zonas más afectadas por el hongo, llegando a lograr una destacable mejoría en su rendimiento y obteniendo una mayor fuente de beneficios. Para ello, no será necesario ir tomando fotos de los cultivos como se hizo previamente, este era uno de los objetivos expuestos en el anteproyecto que dictamina la rentabilidad obtenida entre los resultados sin el motor de inteligencia y con él.

En conclusión, la investigación y el aprendizaje llevado a cabo en este trabajo de fin de grado en campos tan relevantes e influyentes como son la inteligencia artificial y la agricultura de precisión en el desarrollo de una herramienta que dictamine el grado de severidad de un hongo tan devastador en el cultivo del tomate y en la seguridad alimentaria de la sociedad llena de una gran satisfacción y orgullo.

8.2 Líneas futuras

La propagación de este hongo suele darse de forma rápida y severa en aquellas localizaciones que se caracterizan por ser zonas cálidas y húmedas. Más específicamente, las

esporas de este hongo se desarrollan en las hojas de los cultivos que se encuentran en zonas de temperaturas superiores a 10 °C y que tienen una humedad superior al 75% durante varios días.

La lluvia puede llegar a trasladar estas esporas al suelo llegando a infectar a los tubérculos jóvenes y el viento tiene la capacidad de arrastrar estas esporas a varios kilómetros de distancia.

Es por ello, que la inclusión de datos con precipitaciones e intensidades del viento a zonas analizadas por este hongo junto a los datos utilizados en este proyectos servirían de gran ayuda en el estudio y análisis de crecimiento y expansión del hongo.

Además, el análisis de la extracción de varias series temporales en diferentes épocas sobre estos índices de vegetación y la expansión de este estudio a otras zonas nos ayudarían a poder asegurar con mayor eficacia la elaboración de un algoritmo, que haya tenido mucho más rodaje llegando a obtener un motor de inteligencia artificial que pueda llegar a ser más preciso.

Finalmente, dentro de las posibles líneas futuras también destaca con bastante fuerza extrapolar esta herramienta a otros cultivos que también sufran los efectos de este hongo con el objetivo principal de afianzar la confianza y fiabilidad obtenida en el transcurso de este estudio llevado a cabo. Entre las posibles opciones podemos destacar el cultivo de la patata y otras solanáceas.

Referencias

Bibliografía

- Abdulridha, J., Ampatzidis, Y., Kakarla, S. C., & Roberts, P. (2020). Detection of target spot and bacterial spot diseases in tomato using UAV-based and benchtop-based hyperspectral imaging techniques. *Precision Agriculture, 21*(5), 955–978. <https://doi.org/10.1007/s11119-019-09703-4>
- Amatya, S., Karkee, M., Gongal, A., Zhang, Q., & Whiting, M. D. (2016). Detection of cherry tree branches with full foliage in planar architecture for automated sweet-cherry harvesting. *Biosystems Engineering, 146*, 3–15. <https://doi.org/10.1016/j.biosystemseng.2015.10.003>
- Balducci, F., Impedovo, D., & Pirlo, G. (2018). Machine learning applications on agricultural datasets for smart farm enhancement. *Machines, 6*(3). <https://doi.org/10.3390/machines6030038>
- Bandyopadhyay, D., & Sen, J. (2011). Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications, 58*(1), 49–69. <https://doi.org/10.1007/s11277-011-0288-5>
- Barbedo, J. G. A. (2016). A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering, 144*, 52–60. <https://doi.org/10.1016/j.biosystemseng.2016.01.017>
- Basso, B., Cammarano, D., & De Vita, P. (2004). Remotely Sensed Vegetation Indices: Theory and Applications for Crop Management. *Rivista Italiana Di Agrometeorologia, 1*(1), 36–53.
- Bauer, J., Siegmann, B., Jarmer, T., & Aschenbruck, N. (2016). On the potential of Wireless Sensor Networks for the in-situ assessment of crop leaf area index. *Computers and Electronics in Agriculture, 128*, 149–159. <https://doi.org/10.1016/j.compag.2016.08.019>
- Bo, Y., & Wang, H. (2011). The application of cloud computing and the internet of things in agriculture and forestry. *Proceedings - 2011 International Joint Conference on Service Sciences, IJCSS 2011*, 168–172. <https://doi.org/10.1109/IJCSS.2011.40>
- Corrêa, F. M., Bueno Filho, J. S. S., & Carmo, M. G. F. (2009). Comparison of three diagrammatic keys for the quantification of late blight in tomato leaves. *Plant Pathology, 58*(6), 1128–1133. <https://doi.org/10.1111/j.1365-3059.2009.02140.x>
- Dong, X., Vuran, M. C., & Irmak, S. (2013). Autonomous precision agriculture through integration of wireless underground sensor networks with center pivot irrigation systems. *Ad Hoc Networks, 11*(7), 1975–1987. <https://doi.org/10.1016/j.adhoc.2012.06.012>

- Ferrández-Pastor, F. J., García-Chamizo, J. M., Nieto-Hidalgo, M., Mora-Pascual, J., & Mora-Martínez, J. (2016). Developing ubiquitous sensor network platform using internet of things: Application in precision agriculture. *Sensors (Switzerland)*, *16*(7). <https://doi.org/10.3390/s16071141>
- García, E., & Flego, F. (2007). Agricultura de precision. *Tecnología Agropecuaria*, *24*–31.
- Hamad, M. A. A., Eltahir, M., Ali, A. E. M., Hamdan, A. M., & Elsafi, A. A. H. (2018). Efficiency of Using Smart-Mobile Phones in Accessing Agricultural Information by Smallholder Farmers in North Kordofan – Sudan. *SSRN Electronic Journal*, *September*. <https://doi.org/10.20944/preprints201809.0044.v1>
- Hassan-Esfahani, L., Torres-Rua, A., Jensen, A., & McKee, M. (2015). Assessment of surface soil moisture using high-resolution multi-spectral imagery and artificial neural networks. *Remote Sensing*, *7*(3), 2627–2646. <https://doi.org/10.3390/rs70302627>
- Hu, H., Pan, L., Sun, K., Tu, S., Sun, Y., Wei, Y., & Tu, K. (2017). Differentiation of deciduous-calyx and persistent-calyx pears using hyperspectral reflectance imaging and multivariate analysis. *Computers and Electronics in Agriculture*, *137*, 150–156. <https://doi.org/10.1016/j.compag.2017.04.002>
- Jayaraman, P. P., Yavari, A., Georgakopoulos, D., Morshed, A., & Zaslavsky, A. (2016). Internet of things platform for smart farming: Experiences and lessons learnt. *Sensors (Switzerland)*, *16*(11), 1–17. <https://doi.org/10.3390/s16111884>
- Jiang, N., Meng, J., Cui, J., Sun, G., & Luan, Y. (2018). Function identification of miR482b, a negative regulator during tomato resistance to *Phytophthora infestans*. *Horticulture Research*, *5*(1). <https://doi.org/10.1038/s41438-018-0017-2>
- Khanal, S., Fulton, J., & Shearer, S. (2017). An overview of current and potential applications of thermal remote sensing in precision agriculture. *Computers and Electronics in Agriculture*, *139*(June 2017), 22–32. <https://doi.org/10.1016/j.compag.2017.05.001>
- Khanna, A., & Kaur, S. (2019). Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. *Computers and Electronics in Agriculture*, *157*(December 2018), 218–231. <https://doi.org/10.1016/j.compag.2018.12.039>
- Liu, T., Chen, W., Wu, W., Sun, C., Guo, W., & Zhu, X. (2016). Detection of aphids in wheat fields using a computer vision technique. *Biosystems Engineering*, *141*, 82–93. <https://doi.org/10.1016/j.biosystemseng.2015.11.005>
- Mahajan, S., Das, A., & Sardana, H. K. (2015). Image acquisition techniques for assessment of legume quality. *Trends in Food Science and Technology*, *42*(2), 116–133. <https://doi.org/10.1016/j.tifs.2015.01.001>
- Marino, S., Coccozza, C., Tognetti, R., & Alvino, A. (2015). Use of proximal sensing and vegetation indexes to detect the inefficient spatial allocation of drip irrigation in a spot area of tomato field crop. *Precision Agriculture*, *16*(6), 613–629. <https://doi.org/10.1007/s11119-015-9396-7>
- Menacho, C. (2014). Lineal regression models with neural networks. *Anales Científicos*, *75*(2), 253–260.
- Nowinck Marcin, R., Foolad Majid, Nowakowska Marzena, U. K. E. (2012). Potato and Tomato Late Blight by *Phytophthora infestans*: An Overview of Pathology and Resistance Breeding. *Plant Disease*, *96*(1), 4–17.
- Ojha, T., Misra, S., & Raghuwanshi, N. S. (2015). Wireless sensor networks for

- agriculture: The state-of-the-art in practice and future challenges. *Computers and Electronics in Agriculture*, 118, 66–84.
<https://doi.org/10.1016/j.compag.2015.08.011>
- Pantazi, X. E., Moshou, D., & Bravo, C. (2016). Active learning system for weed species recognition based on hyperspectral sensing. *Biosystems Engineering*, 146, 193–202. <https://doi.org/10.1016/j.biosystemseng.2016.01.014>
- Patrício, D. I., & Rieder, R. (2018). Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and Electronics in Agriculture*, 153(August), 69–81.
<https://doi.org/10.1016/j.compag.2018.08.001>
- Paustian, M., & Theuvsen, L. (2017). Adoption of precision agriculture technologies by German crop farmers. *Precision Agriculture*, 18(5), 701–716.
<https://doi.org/10.1007/s11119-016-9482-5>
- Precision Ag Technology | Field and Water Management | John Deere US*. (n.d.). Retrieved July 27, 2021, from <https://www.deere.com/en/technology-products/precision-ag-technology/field-and-water-management/>
- Romualdo, L. M., Luz, P. H. C., Devechio, F. F. S., Marin, M. A., Zúñiga, A. M. G., Bruno, O. M., & Herling, V. R. (2014). Use of artificial vision techniques for diagnostic of nitrogen nutritional status in maize plants. *Computers and Electronics in Agriculture*, 104, 63–70. <https://doi.org/10.1016/j.compag.2014.03.009>
- Sabanci, K., Kayabasi, A., & Toktas, A. (2017). Computer vision-based method for classification of wheat grains using artificial neural network. *Journal of the Science of Food and Agriculture*, 97(8), 2588–2593. <https://doi.org/10.1002/jsfa.8080>
- Shrivastava, S., Singh, S. K., & Hooda, D. S. (2017). Soybean plant foliar disease detection using image retrieval approaches. *Multimedia Tools and Applications*, 76(24), 26647–26674. <https://doi.org/10.1007/s11042-016-4191-7>
- Tong-ke, F. (2013). Smart Agriculture Based on Cloud Computing and IOT. *Journal of Convergence Information Technology*, 8, 210-216.
- Vuran, M. C., Salam, A., Wong, R., & Irmak, S. (2018). Internet of underground things in precision agriculture: Architecture and technology aspects. *Ad Hoc Networks*, 81, 160–173. <https://doi.org/10.1016/j.adhoc.2018.07.017>
- Wang, D. C., Zhang, G. L., Pan, X. Z., Zhao, Y. G., Zhao, M. S., & Wang, G. F. (2012). Mapping Soil Texture of a Plain Area Using Fuzzy-c-Means Clustering Method Based on Land Surface Diurnal Temperature Difference. *Pedosphere*, 22(3), 394–403. [https://doi.org/10.1016/S1002-0160\(12\)60025-3](https://doi.org/10.1016/S1002-0160(12)60025-3)
- Wang, D. C., Zhang, G. L., Zhao, M. S., Pan, X. Z., Zhao, Y. G., Li, D. C., & Macmillan, B. (2015). Retrieval and mapping of soil texture based on land surface diurnal temperature range data from MODIS. *PLoS ONE*, 10(6), 1–14.
<https://doi.org/10.1371/journal.pone.0129977>
- Zareiforush, H., Minaei, S., Alizadeh, M. R., & Banakar, A. (2015). A hybrid intelligent approach based on computer vision and fuzzy logic for quality measurement of milled rice. *Measurement: Journal of the International Measurement Confederation*, 66, 26–34. <https://doi.org/10.1016/j.measurement.2015.01.022>
- Zhang, J., & Wang, Y. (2021). Design of remote control device using wireless sensor network and its use in intelligent monitoring of farmland information. *Eurasip Journal on Wireless Communications and Networking*, 2021(1), 1–14.
<https://doi.org/10.1186/s13638-021-01997-1>
- Zhao, J. C., Zhang, J. F., Feng, Y., & Guo, J. X. (2010). The study and application of the

IOT technology in agriculture. *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*, 2, 462–465. <https://doi.org/10.1109/ICCSIT.2010.5565120>

Zhu, Y., Cao, Z., Lu, H., Li, Y., & Xiao, Y. (2016). In-field automatic observation of wheat heading stage using computer vision. *Biosystems Engineering*, 143, 28–41. <https://doi.org/10.1016/j.biosystemseng.2015.12.015>

Webgrafía

- Actualizar controladores y configuración jugable óptima | EXPERIENCIA NVIDIA GeForce.* (n.d.). Retrieved July 9, 2021, from <https://www.nvidia.com/es-es/geforce/geforce-experience/>
- Agriculturers. (2021). Inteligencia Artificial en la agricultura de precisión. <https://agriculturers.com/inteligencia-artificial-en-la-agricultura-de-precision/>
- Anaconda Inc. (2020). *Anaconda Individual Edition*. Anaconda Website. <https://www.anaconda.com/products/individual-d>
- Beginners Ask “How Many Hidden Layers/Neurons to Use in Artificial Neural Networks?” | by Ahmed Gad | Towards Data Science. (n.d.). Retrieved September 6, 2021, from <https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e>
- B.V, E. (2021). Scopus - Document search | Signed in.
- Calo, F. (2018). ¿Cuál es la diferencia entre el índice NDVI y GNDVI? - Soporte. <https://soporte.auravant.com/soporte/cual-es-la-diferencia-entre-el-indice-ndvi-y-gndvi/>
- Cubert GmbH. (2020). Vegetation Indices - Cubert GmbH - Germany. Cubert GmbH. <https://cubert-gmbh.com/applications/vegetation-indices/>
- Descripción general de diferentes optimizadores para redes neuronales.* (n.d.). Retrieved September 9, 2021, from <https://ichi.pro/es/descripcion-general-de-diferentes-optimizadores-para-redes-neuronales-247308806799555>
- Document search - All Databases. (n.d.). Retrieved May 24, 2021, from <https://www.webofscience.com/wos/alldb/basic-search>
- Drumond, C. (2021). Scrum: qué es, cómo funciona y por qué es excelente. <https://www.atlassian.com/es/agile/scrum>
- Explicación de las Funciones de activación en Redes Neuronales y práctica con Python. | by Rubiales Alberto | Medium. (n.d.). Retrieved September 4, 2021, from <https://rubialesalberto.medium.com/explicación-funciones-de-activación-y-práctica-con-python-5807085c6ed3>
- Función de activación Relu — Aprendizaje automático — DATA SCIENCE. (n.d.). Retrieved September 4, 2021, from <https://datascience.eu/es/aprendizaje-automatico/funcion-de-activacion-relu/>
- Función de pérdida de Pytorch Función de pérdida utilizar una explicación detallada - programador clic.* (n.d.). Retrieved September 9, 2021, from <https://programmerclick.com/article/26841407070/>
- IBM Corporation. (2019). El modelo de redes neuronales - Documentación de IBM. IBM Corporation. <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model>
- Ionos. (2020). ¿Qué es una neural network? - IONOS. <https://www.ionos.es/digitalguide/online-marketing/marketing-para-motores-de-busqueda/que-es-una-neural-network/>
- Jábega, Catálogo de la Biblioteca de la Universidad de Málaga. (n.d.). Retrieved July 12, 2021, from https://jabega.uma.es/discovery/search?vid=34CBUA_UMA:VU1
- JetBrains s.r.o. (2020). *PyCharm: el IDE de Python para desarrolladores profesionales, por JetBrains*. <https://www.jetbrains.com/es-es/pycharm/>

Lucidchart. (n.d.). *Software de diagramas de red* | Lucidchart. Retrieved August 21, 2021, from <https://www.lucidchart.com/pages/es>

Luis Velasco. (2020). *Optimizadores en Redes Neuronales Profundas: un Enfoque Práctico*. Medium. <https://velascoluis.medium.com/optimizadores-en-redes-neuronales-profundas-un-enfoque-práctico-819b39a3eb5>

neural-network — ¿Cuál es la diferencia entre redes neuronales convolucionales y recurrentes? (n.d.). Retrieved September 17, 2021, from <https://qastack.mx/stats/2213/whats-the-difference-between-feed-forward-and-recurrent-neural-networks>

NVIDIA. (2006). *Liderazgo en computación de inteligencia artificial de NVIDIA*. NVIDIA. <https://www.nvidia.com/es-es/>

Plagas y enfermedades del tomate | *El huerto urbano :: El huerto en casa*. (n.d.). Retrieved September 2, 2021, from <https://www.elhuertourbano.net/hortalizas/plagas-y-enfermedades-del-tomate/>

Plagas y Enfermedades | CIAT. (n.d.). Retrieved September 1, 2021, from <https://ciat.cgiar.org/lo-que-hacemos/plagas-y-enfermedades/?lang=es>

Precision Ag Technology | *Field and Water Management* | *John Deere US*. (n.d.). Retrieved July 27, 2021, from <https://www.deere.com/en/technology-products/precision-ag-technology/field-and-water-management/>

PyTorch. (n.d.). Retrieved July 9, 2021, from <https://pytorch.org/>

Rodríguez Manuel. (2017). *Agricultura de precisión: cómo la inteligencia artificial revoluciona la agricultura*. *Agricultura de Precision / Como La Inteligencia Artificial*.

TIPOS DE REDES NEURONALES – INTELIGENCIA ARTIFICIAL. (n.d.). Retrieved September 5, 2021, from <https://inteligencia-artificial.dev/tipos-redes-neuronales/>

Universidad de Sevilla. (2020). *Conceptos Básicos sobre Redes Neuronales*. Universidad de Sevilla, 1–63. <https://vincentblog.xyz/posts/conceptos-basicos-sobre-redes-neuronales>

Apéndice A

Manual de Instalación

Requerimientos:

En este apartado se detalla los requisitos necesarios que debe cumplir nuestro sistema o dispositivo para poder llevar a cabo la instalación del entorno de prototipado del proyecto, IDE: PyCharm, dependiendo del sistema operativo utilizado:

- **Windows**
 - Versiones de 64 bits de Microsoft Windows 10, 8.
 - 2 GB de RAM como mínimo, 8 GB recomendados.
 - 2,5 GB de espacio en disco duro, SSD recomendado.
 - 1024x768 mínimo de resolución de pantalla.
 - Python 2.7, Python 3.5 o más reciente.
- **macOS**
 - macOS 10.14 o superior.
 - 2 GB de RAM como mínimo, 8 GB recomendados.
 - 2,5 GB de espacio en disco duro, SSD recomendado.
 - 1024x768 mínimo de resolución de pantalla.
 - Python 2.7, Python 3.5 o más reciente.

JetBrains Runtime (JBR) 11 está incluido en PyCharm. así que no será necesario instalar Java en el dispositivo para ejecutar PyCharm.

- **Linux**
 - Escritorio GNOME o KDE.
 - GB de RAM como mínimo, 8 GB recomendados.
 - 2,5 GB de espacio en disco duro, SSD recomendado.
 - 1024x768 mínimo de resolución de pantalla.
 - Python 2.7, Python 3.5 o más reciente.

JetBrains Runtime (JBR) 11 está incluido en PyCharm. así que no será necesario instalar Java en el dispositivo para ejecutar PyCharm.

Instalación:

Tras ver los requerimientos necesarios, se mostrará los pasos a seguir para la instalación de *PyCharm* dependiendo del sistema operativo.

- **Windows**

1. Ejecute el archivo `pycharm-2021.1.2.exe` para iniciar el Asistente de instalación.
2. Siga todos los pasos sugeridos por el asistente. Preste especial atención a las opciones de instalación correspondientes.

- **macOS**

1. Descargue el archivo de imagen de disco para macOS `pycharm-2021.1.2.dmg`
2. Móntelo como un nuevo disco en su sistema
3. Copie PyCharm a su carpeta de Aplicaciones

- **Linux**

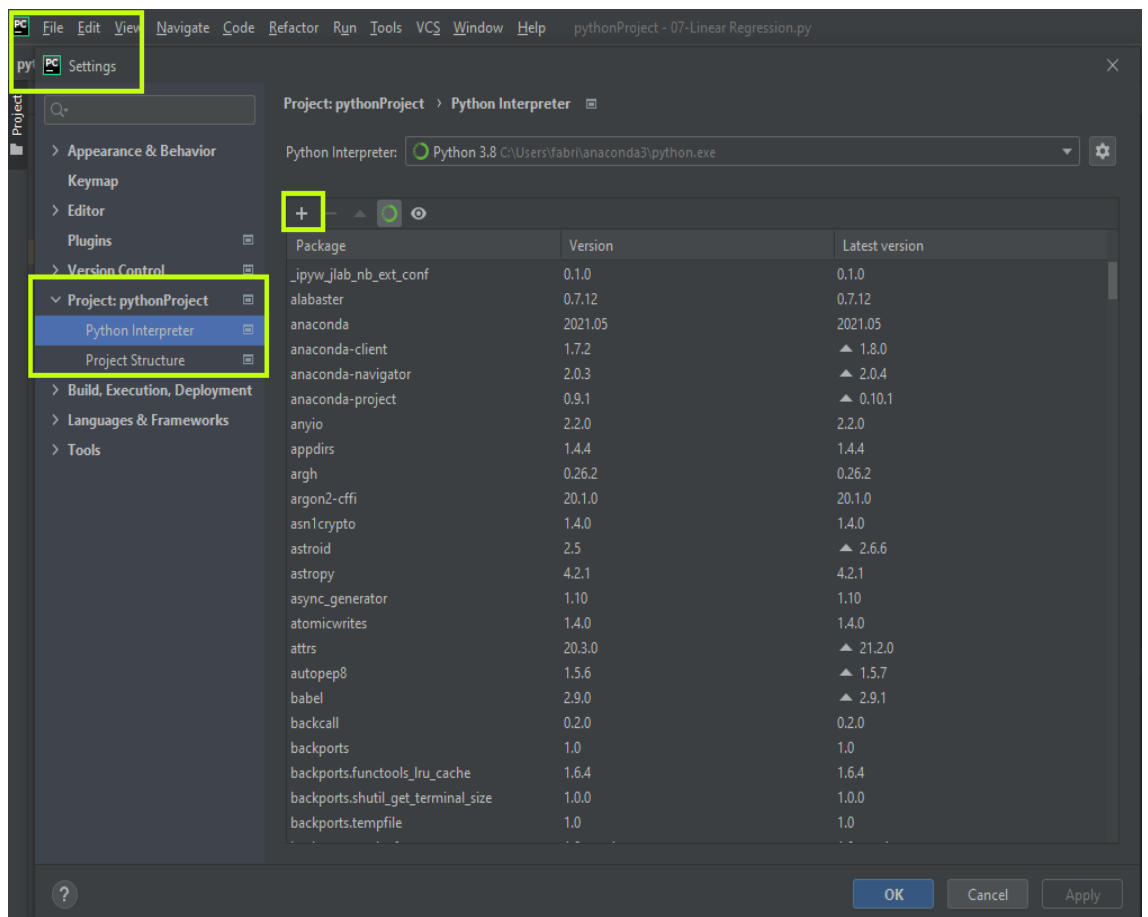
1. Copie el archivo `pycharm-2021.1.2.tar.gz` a la ubicación de instalación deseada (asegúrese de tener permisos de rw para ese directorio).
2. Descomprima el archivo `pycharm-2021.1.2.tar.gz` en un directorio vacío utilizando el siguiente comando: `tar -xzf pycharm-2021.1.2.tar.gz`

Nota: NO SE DEBE extraer una nueva instancia sobre una ya existente. La carpeta de destino debe estar vacía.

3. Ejecute `pycharm.sh` desde el subdirectorio `bin`.

Finalmente, después de tener instalado el IDE Pycharm, solo será necesario añadir las librerías necesarias para el correcto funcionamiento del proyecto. Tendremos dos opciones, en ciertos casos podemos incluirlas desde la propia plataforma sin tener que realizar ningún tipo de instalación adicional o tendremos que añadir un interpretador diferente.

En la primera sólo será necesario acceder al apartado *Files->Settings*, posteriormente ir a la sección *Project->pythonProject* y por medio de + añadir la librería que necesitemos.



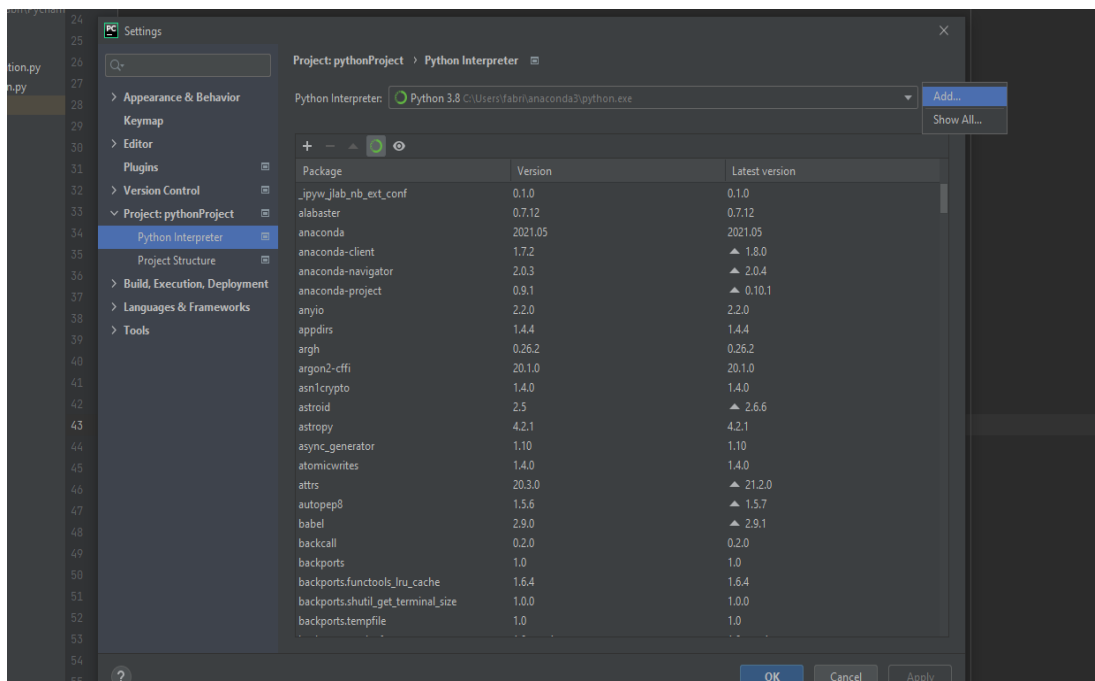
Aunque hay ciertos casos en dónde esto no es posible, como es este, y es necesario descargar un interpretador aparte.

Por ello, desde la propia página de Pytorch se nos recomienda descargar el interpretador de Anaconda por medio de CUDA para poder utilizar esta librería. Por lo que en primer lugar será necesario descargar la plataforma de computación CUDA y posteriormente instalar el paquete de Anaconda.

Después de esto solo nos quedaría ejecutar los siguientes comandos en el símbolo del sistema y ya tendríamos la librería instalada.

PyTorch Build	Stable (1.9.0)	Preview (Nightly)	LTS (1.8.1)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 10.2	CUDA 11.1	ROCm 4.2 (beta)	CPU
Run this Command:	<code>conda install pytorch torchvision torchaudio cudatoolkit=10.2 -c pytorch</code>			

Finalmente, una vez instalado el paquete de Anaconda con la correcta ejecución de los comandos en el símbolo del sistema, solo quedaría añadir el nuevo interpretador en el apartado de *Project->pythonProject* nombrado anteriormente.



Con la instalación del interpretador junto con las librerías, solo es necesario recordar realizar la importación de cada una de estas al principio del código como en el siguiente ejemplo.

```
import torch
import torch.nn as nn
import torch.optim as optim
import numpy as np
import pandas as pd
from torch.utils.data import DataLoader, Dataset
import matplotlib.pyplot as plt
import math
```

