



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

UBUSETAS



Presentado por Adrián Antón García
en Universidad de Burgos — 30 de noviembre
de 2017

Tutores:

Dr. José F. Díez Pastor

Dr. Raúl Marticorena Sánchez



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Adrián Antón García, con DNI 71300381J, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 30 de noviembre de 2017

Vº. Bº. del Tutor:

Vº. Bº. del Tutor:

Dr. José F. Díez Pastor

Dr. Raúl Marticorena Sánchez

Resumen

En este proyecto se va a desarrollar una aplicación Android que permita, mediante la fotografía de una seta, clasificar a que especie pertenece. Además se proporcionará información de las especies más probables, así como imágenes de estas especies para comparar con nuestra fotografía.

Para realizar la parte del reconocimiento de imágenes se aplicarán técnicas de minería de datos para usar los clasificadores de imágenes.

A la hora de recopilar la información necesaria, se usarán técnicas de Web semántica para conseguir una pequeña descripción e información interesante sobre cada especie de seta. Esta parte se realizará realizando consultas a la DBpedia.

Para completar el trabajo del clasificador de imágenes, se integrarán diferentes claves dicotómicas para el reconocimiento de géneros y especies de setas. Estas claves se conseguirán realizando técnicas de Web Scraping.

El objetivo del proyecto es el de crear una herramienta que sirva de apoyo en las tareas de micología relacionadas con el reconocimiento de setas.

Descriptores

Palabras clave. *Minería de datos, Clasificador, Web Semántica, Clave dicotómica, Android, Seta, Micología, Reconocimiento*

Abstract

In this project an Android application will be developed that allows, through the photograph of a mushroom, to classify which species it belongs to. In addition information about the most likely species will be provided, as well as images of these species to compare with our photograph.

To perform the image recognition part, data mining techniques will be applied to use the image classifiers.

When gathering the necessary information, semantic Web techniques will be used to obtain a short description and interesting information about each species of mushroom. This part will be done by consulting the DBpedia.

To complete the work of the image classifier, different dichotomous keys will be integrated for the recognition of genera and species of mushrooms. These keys will be achieved by performing Web Scraping techniques.

The objective of the project is to create a tool to support the mycology tasks related to the recognition of mushrooms.

Keywords

Keywords *Data Mining, Sorter, Semantic Web, Dichotomous Key, Android, Mushroom, Mycology, Recognition*

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos Generales	3
2.2. Objetivos Técnicos	4
Conceptos teóricos	7
3.1. Micología	7
3.2. Hongos	7
3.3. Setas	8
3.4. Inteligencia Artificial	9
3.5. Minería de datos	11
3.6. Redes neuronales	13
3.7. Modelo Inception	19
3.8. Modelo Mobilenet	20
3.9. Web Semántica	22
3.10. Web Scraping	24
3.11. Clave Dicotómica	24
Técnicas y herramientas	27
4.1. Java	27

4.2. Eclipse	27
4.3. Android	28
4.4. Android Studio	29
4.5. JavaScript	29
4.6. SLF4J	29
4.7. Apache Maven	29
4.8. RDF	30
4.9. Apache Jena	31
4.10. JSON	33
4.11. Jaunt	33
4.12. SQLite	33
4.13. SPARQL	34
4.14. DBpedia	34
4.15. Tensorflow	34
4.16. Git	35
4.17. GitHub	35
4.18. Zenhub	35
4.19. SourceTree	36
4.20. Latex	36
Aspectos relevantes del desarrollo del proyecto	37
Trabajos relacionados	39
Conclusiones y Líneas de trabajo futuras	41
Bibliografía	43

Índice de figuras

3.1. Partes de una seta	9
3.2. Flujos de trabajo, clasificación	13
3.3. Modelo de una neurona artificial	14
3.4. Función de propagación	14
3.5. Funciones de activación	15
3.6. Grafo de una RNA	16
3.7. Izquierda: grafo de una RNA. Derecha: Grafo de una Red neuronal convolucional.	17
3.8. Figura que muestra una convolución en dos dimensiones.	17
3.9. Técnica de máx pooling.	18
3.10. Ejemplo de un nodo Inception.	19
3.11. Aplicación de una convolucion 1x1 antes de ejecutar la convolucion 5x5.	20
3.12. Los filtros convolucionales normales son divididos en los filtros profundos y los puntuales.	23
4.13. Descripciones RDF.	31
4.14. Arquitectura de Apache Jena	32

Índice de tablas

3.1. Depthwise Separable vs Full Convolution MobileNet	22
--	----

Introducción

La tarea de clasificar una seta es altamente compleja ya que las diferencias entre algunas especies son mínimas y se necesitan de expertos que las analicen detalladamente hasta clasificar con seguridad la especie. Además dependiendo de la fase de crecimiento en la que se encuentre la propia seta y los factores a los que se encuentre expuesta, como puede ser la humedad del ambiente, pueden provocar que la apariencia de la seta cambie dificultando aún más esta tarea.

La clasificación de una seta debe realizarse de forma segura y minuciosa ya que hay que asegurarse que la seta no sea tóxica ni provoque perjuicios a aquella persona que la consuma.

Ante estas dificultades nace la idea de crear una aplicación que nos ayude con la identificación de la especie o género a la que pertenece una seta. Debe quedar claro desde el principio que esta aplicación solo debe servir de guía y no pretende sustituir los conocimientos de un experto, simplemente servir de apoyo o de ayuda en la difícil tarea de clasificar la especie a la que pertenece una seta.

Por estos motivos los resultados obtenidos por el clasificador deben tomarse sólo como una ayuda y usarlos para encaminar nuestra búsqueda de la especie correcta en el buen camino.

Para hacer uso de la aplicación, deberemos introducir una foto de la seta a clasificar, bien directamente sacando una foto desde la cámara o desde la galería de nuestro móvil. Con esta foto la aplicación nos mostrará las especies que el clasificador ha determinado como más probables. Si pulsamos sobre cada especie mostrada se nos mostrará información de esa especie, una breve descripción, la comestibilidad, así como diferentes imágenes que podremos comparar con la seta a clasificar. Además de mostrar esta información,

la aplicación nos hará preguntas mediante claves dicotómicas según los resultados obtenidos para identificar correctamente la seta. Las preguntas consistirán en una breve descripción de características de la seta, el usuario deberá elegir la que más se adecue a la seta para poder llegar a la especie concreta.

La aplicación se ejecuta íntegramente en el móvil, es decir, no necesitamos de una conexión a Internet para ejecutar el clasificador ni obtener información de la seta. Actualmente el clasificador es capaz de ejecutarse en el propio móvil sin necesidad de un servidor externo en un tiempo casi instantáneo permitiendo diferenciar entre 173 especies de setas diferentes. La aplicación cuenta con una pequeña base de datos que contiene información de cada especie, si lo deseamos podemos acceder por Internet al link proporcionado en cada especie a la página en la Wikipedia de la especie.

Objetivos del proyecto

En este apartado se va a proceder a explicar los objetivos marcados en el proyecto, diferenciando entre los objetivos generales que se requerían para llevar a cabo este proyecto y los objetivos técnicos que se han ido encontrando a lo largo de su ejecución.

2.1. Objetivos Generales

A continuación se muestran los principales objetivos del proyecto de forma general:

- El objetivo principal del proyecto es el de crear una aplicación Android que fuera capaz de identificar, mediante un clasificador de imágenes, la especie a la que pertenece una seta mediante una fotografía introducida por el usuario, bien por la cámara o desde la galería del móvil.
- Otro objetivo es el de mostrar información de las diferentes especies clasificadas. En esta información se incorporaría una breve descripción de la especie, la comestibilidad y otros datos de interés. Este objetivo se realizará mediante técnicas de web semántica.
- Como complemento al clasificador se debería implementar una serie de claves dicotómicas que mediante preguntas simples acerca de la seta conduzcan a la especie correcta de la que se trata. Este objetivo se implementará mediante técnicas de web scraping.
- Se intentará desarrollar una interfaz atractiva y amigable en la aplicación Android.

2.2. Objetivos Técnicos

En esta lista se van a detallar los objetivos técnicos que se han planteado para implementar los objetivos generales descritos anteriormente.

- Usar Android Studio para implementar la aplicación Android.
- Un objetivo técnico importante que se marcó fue el de ejecutar el clasificador en el propio móvil, para lo que se utilizaron las librerías de Tensorflow para Android Studio junto a los modelos de clasificación Mobilenet.
- Usar Java como lenguaje de programación principal a la hora de programar la parte de Web Semántica y la extracción de las claves dicotómicas mediante Web Scraping.
- Utilizar Sparql como lenguaje para realizar las consultas a la DBpedia.
- Usar las librerías de Apache Jena para realizar las consultas con sparql a la DBpedia en Java y extraer la información de las especies de setas.
- Utilizar las librerías de Jaunt en java para realizar la parte de Web Scraping y poder extraer las claves dicotómicas.
- Utilizar SQLite como base de datos para almacenar los datos de las especies y las claves en Android.
- Usar un sistema de control de versiones, se ha optado por utilizar el servicio GitHub.
- Utilizar la metodología Scrum para realizar un seguimiento de las tareas realizadas durante el proyecto. Se ha elegido usar la herramienta ZenHub para este propósito.
- Se intentará que la aplicación no ocupe demasiada memoria dentro del dispositivo, para ello se formatearán las imágenes para que ocupen lo necesario y se buscará un modelo de clasificador que no sea demasiado exigente, en nuestro caso se ha optado por el modelo Mobilenet-224-v1.
- Llevar a cabo un prototipo de la interfaz de usuario mediante alguna herramienta de prototipado.
- Utilizar herramientas de testing para probar el correcto funcionamiento de nuestra aplicación.

- Hacer uso de las directrices de diseño de Material design para desarrollar la interfaz gráfica.

Conceptos teóricos

Para la correcta comprensión del proyecto se van a explicar a continuación una serie de conceptos teóricos mínimos necesarios.

3.1. Micología

La micología es la ciencia que se dedica al estudio de los hongos. Es una de las áreas de la ciencia más extensas y diversificadas que aporta avances significativos a la investigación científica y al desarrollo tecnológico. [28]

Tiene varias aplicaciones y objetivos, se usa para determinar que especies de hongos son comestibles o no y si podrían usarse para diferentes tratamientos médicos, ciertas sustancias de las setas son estudiadas con estos fines curativos. [3]

3.2. Hongos

El hongo es el nombre común de los organismos del reino Fungi. El término Fungi en biología se refiere a un grupo de organismos eucariotas¹ que se clasifican en un reino distinto al de las plantas, animales y protistas. Se diferencian de las plantas en que son heterótrofos² y de los animales en que tienen paredes celulares, como las plantas, pero compuestas de quitina³ en vez de celulosa.

¹ Aquellos organismos formados por células con núcleo verdadero

² Seres vivos que necesitan de otros para alimentarse

³ <https://es.wikipedia.org/wiki/Quitina>

Los hongos se reproducen de forma sexual o asexual mediante esporas, que se dispersan en un estado latente y solo se interrumpe cuando se dan las condiciones adecuadas para su germinación. [22]

La mayoría de los hongos están formados por estructuras microscópicas, filamentosas y ramificadas llamadas hifas. El conjunto de estas hifas forma una red a la que llamaremos micelio. Cuando la acumulación de hifas es grande se pueden observar como una red algonodosa que se pueden reconocer por ejemplo cuando los alimentos empiezan a descomponerse. [6]

3.3. Setas

Las setas son los cuerpos fructíferos de algunos tipos de hongos (no todos los hongos producen setas), en otras palabras, son la parte reproductiva de los hongos. La principal función de la seta es dispersar las esporas del hongo. Normalmente es la parte visible del hongo ya que este suele estar bajo tierra.

Partes de las setas

A continuación se van a describir las partes más características de una seta que se usan para su clasificación.

- Sombrero: Situado sobre el pie, ejerce la función de protección en la formación y desarrollo de las esporas. El sombrero es un elemento clave a la hora de diferenciar las especies ya que puede adoptar diferentes formas, aspectos y colores.
- Himenio: Es la parte situada justo debajo del sombrero y que puede adoptar diferentes formas (láminas, tubos, aguijones o pliegues), estas diferentes formas nos ayudarán a diferenciar entre las especies. La función principal de esta parte es la de crear, desarrollar, almacenar y dispersar las esporas que generan nuevos hongos.
- Pie: Elemento que no tiene por que aparecer en todas las setas y que sujeta al sombrero e himenio.
- Volva: Es un fragmento en forma de membrana que envuelve la base del pie en algunas setas. Es un error común cortar el pie de la seta y no conservar la volva que nos podría dar indicios de a que especie pertenece la seta. [7]

En la siguiente figura 3.1 se muestran las partes más relevantes de una seta.

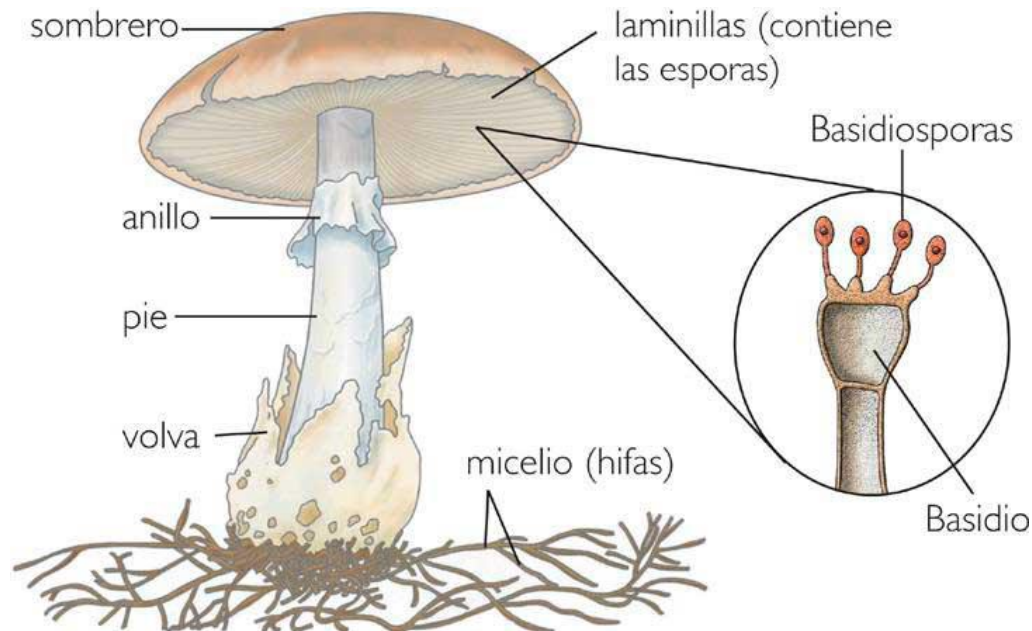


Figura 3.1: Partes de una seta

3.4. Inteligencia Artificial

Hay muchas definiciones para definir lo que es la inteligencia artificial, normalmente aparece bajo las siglas IA o AI (Del inglés Artificial Intelligence), pero de forma general la podemos definir como el desarrollo de algoritmos y métodos que otorgan a las computadoras la capacidad de realizar acciones o tareas como las realizaría un ser humano. La inteligencia artificial se aplica en multitud de campos como puede ser la robótica, automovilismo, reconocimiento de imágenes, medicina, sistemas de apoyo...

Stuart Russell y Peter Norving[23] diferencian estos cuatro tipos de inteligencia artificial:

- Sistemas que piensan como humanos: Son sistemas que intentan emular el pensamiento humano automatizando actividades que relacionamos con los procesos de pensamiento humano (toma de decisiones, resolución de problemas...), por ejemplo las redes neuronales artificiales.

- Sistemas que actúan como humanos: Este tipo de sistemas intentan imitar el comportamiento humano, por ejemplo la robótica.
- Sistemas que piensan racionalmente: Tratan de imitar con lógica el pensamiento lógico racional del ser humano, por ejemplo los sistemas expertos.
- Sistemas que actúan racionalmente: Intentan emular de forma racional el comportamiento humano.

Algunos de estos sistemas necesitan desarrollar una capacidad de aprendizaje para poder imitar de forma correcta el comportamiento humano.

Aprendizaje automático

El aprendizaje automático (del inglés Machine Learning) es una rama de la inteligencia artificial que se encarga de desarrollar los algoritmos que permitan a las computadoras aprender ⁴. Para ello, se entrenan los sistemas con una serie de ejemplos, o información de entrenamiento, para generalizar comportamientos que deben realizar ante diferentes casos de entrada.

Hay una gran variedad de algoritmos de aprendizaje automático, a continuación se presentan algunos de los tipos más extendidos:

- Aprendizaje supervisado: En estos algoritmos se intenta crear una función que nos de la correspondencia entre los datos de entrada y una de las salidas deseadas. En este tipo de algoritmos se tiene información tanto de los valores de entrada como del los valores deseados. Para entrenar se suelen usar pares en los que un componente son los datos de entrada y otro los valores deseados. Los valores deseados suelen ser denominados como clases. [19]

El valor de salida de la función puede ser un número (Regresión) o una etiqueta de clase (Clasificación).

Los algoritmos de clasificación usados en este proyecto se corresponderían con este tipo de aprendizaje, siendo las etiquetas de clase las especies de setas que se quieren clasificar.

- Aprendizaje no supervisado: El aprendizaje no supervisado se diferencia del supervisado en que no se conoce a priori los resultados deseados. En este tipo de aprendizaje, normalmente, se tratan los datos de entrada

⁴Adquirir conocimiento por medio de la experiencia

como una serie de datos aleatorios y se intentan buscar relaciones ocultas entre ellos. Una forma de aprendizaje no supervisado es el clustering, el cuál se encarga de organizar los datos de entrada en grupos en los que comparten propiedades y los diferencian del resto. [18]

- Aprendizaje semisupervisado: Este tipo de aprendizaje mezcla las dos técnicas descritas anteriormente. Se usa cuando tenemos tanto datos conocidos como datos de los que no tenemos conocimiento.
- Aprendizaje por refuerzo: Este tipo de algoritmos realiza una aprendizaje mediante el modelo de prueba-error, es decir, el algoritmo realiza acciones y a partir de los resultados de estas el algoritmo evalúa cual han sido beneficiosas y cuales no.

En nuestro proyecto se hace uso del aprendizaje supervisado para entrenar el clasificador de imágenes que reconocerá las especies de las setas. [17]

3.5. Minería de datos

La minería de datos consiste en la aplicación de técnicas de inteligencia artificial sobre grandes cantidades de datos con el objetivo de descubrir patrones o relaciones ocultas entre los datos. La parte de la minería de datos más relevante para nuestro proyecto son los clasificadores previamente introducidos [13]. Básicamente los clasificadores se basan en la hipótesis del aprendizaje inductivo:

Cualquier hipótesis que aproxime bien una función objetivo sobre un conjunto de ejemplos de entrenamiento suficientemente grande también aproximará bien la función objetivo en ejemplos no observados //CITAR UBU

La minería de datos, típicamente, consta de los siguientes pasos:

- Selección del conjunto de datos: En esta parte del proceso se seleccionarán las variables objetivo, las variables independientes y se realizará un muestreo con los registros disponibles.
- Análisis de las propiedades de los datos: Se comprobarán los histogramas, diagramas de dispersión, si se encuentran valores atípicos entre el conjunto de datos y si faltan ciertos datos.

- Transformación de los datos: En esta etapa del proceso se procesarán los datos de entrada para adecuarlos a la estructura requerida por las técnicas de inteligencia artificial que se aplicarán sobre ellos.
- Selección de la técnica de minería de datos: Se construirá el modelo predictivo a aplicar sobre los datos. Clasificación o segmentación.
- Extracción del conocimiento: Esta es la parte donde se aplicarán las técnicas de inteligencia artificial sobre los datos para extraer el conocimiento buscado. La técnica que hayamos aplicado nos devolverá un modelo de conocimiento sobre los datos.
- Interpretación y evaluación de los datos: Esta última etapa consiste en validar si los resultados que nos ofrece el modelo entrenado son satisfactorios y se adecuan a los requisitos pedidos.

Evolución del flujo de trabajo

Al principio, el flujo de trabajo para entrenar un clasificador, era el de extraer unas características de los datos de entrenamiento que se proporcionaban al clasificador. En este modo de trabajo se usan métodos como el Bag-of-words⁵ para determinar las características de los datos de entrada.

Posteriormente, este flujo de trabajo se ha modificado, introduciendo redes neuronales que se encargan de extraer estas características y dividiéndolas en diferentes jerarquías para que no todas tengan el mismo peso a la hora de clasificar.

En la figura 3.2 se muestra como se han ido incorporando diferentes capas, todas de ellas entrenables, para extraer las características de una forma jerárquica mediante redes neuronales.

⁵Por ejemplo, tratar las características de una imagen como palabras y guardar las veces que se repiten estas características en cada imagen. https://es.wikipedia.org/wiki/Modelo_bolsa_de_palabras

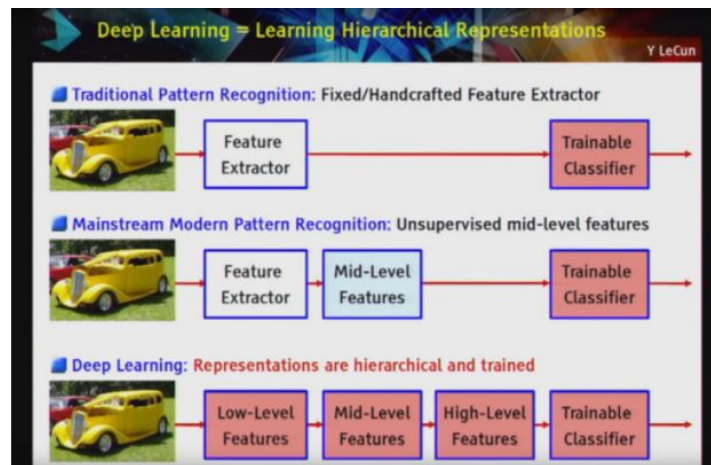


Figura 3.2: Flujos de trabajo, clasificación

3.6. Redes neuronales

En esta sección se explicarán algunos conceptos teóricos sobre redes neuronales necesarios para entender el funcionamiento de los modelos Mobilenet e Inception usados para clasificar imágenes en nuestro proyecto. Actualmente hay gran variedad de clasificadores que se pueden usar para esta tarea además de los dos mencionados.

Modelo de neurona artificial

Una neurona artificial de forma general se puede describir según el modelo de Rumelhart y McClelland (1986), el cual define la neurona o elemento de proceso (EP) como un dispositivo el cuál a partir de un conjunto de entradas, vector x , genera una única salida y . [5]

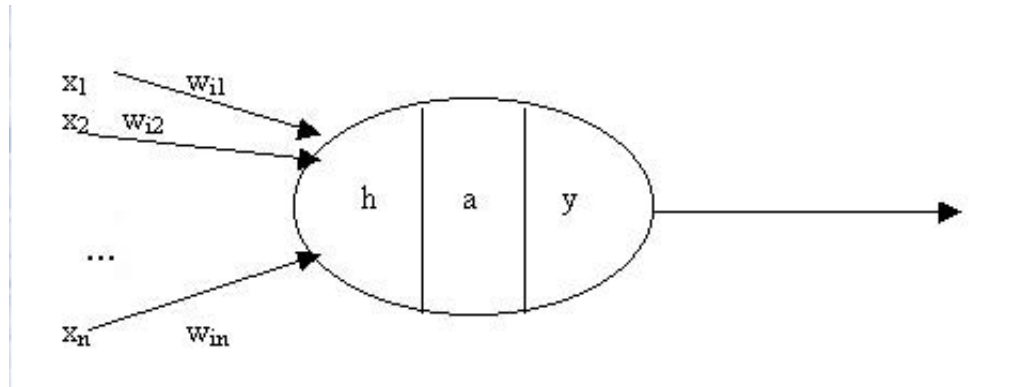


Figura 3.3: Modelo de una neurona artificial

La neurona artificial consta de los siguientes elementos:

- Conjunto de entradas x : El vector x .
- Conjunto de pesos sinápticos w_{ij} : Representan la relación entre la neurona i y j .
- Regla de propagación: proporciona el potencial postsináptico, $h_i(t)$. Se suele representar como una suma ponderada de la siguiente forma [3.4](#)

$$h_1(t) = \sum (w_{1j} * x_j)$$

Figura 3.4: Función de propagación

- Función de activación a : proporciona el estado de activación en función del estado anterior y el potencial postsináptico.
- Función de salida y : proporciona la salida y en función del valor de activación. En la figura [3.5](#) podemos ver diferentes funciones de activación.



Figura 3.5: Funciones de activación

Red Neuronal Artificial RNA

Una red neuronal Artificial RNA es un grafo dirigido en el que las neuronas artificiales están conectadas con otras y los enlaces pueden aumentar o disminuir el estado de activación de las neuronas conectadas. Cada neurona opera de forma individual mediante operaciones de suma.^[29]

Estas neuronas se organizan en capas o layers formando diferentes filtros para generar algoritmos que puedan resolver diferentes problemas. Dentro de una misma capa las neuronas suelen ser del mismo tipo y podemos diferenciar de forma general 3 tipos distintos de capas:

- De entrada: Reciben datos o señales externos del entorno.
- De salida: Proporcionan las respuestas de la red ante los estímulos recibidos por las capas de entrada.
- Ocultas: No interaccionan con el entorno, son nodos intermedios que ejecutan las operaciones internas de la red.

Como se puede observar en la figura 3.6, las neuronas representadas como nodos se organizan en las diferentes capas pudiéndose interconectar con varias neuronas a la vez. Estas redes pueden tener un número de capas y neuronas ilimitados. Pueden tener un número ilimitado de enlaces o solo estar conectadas con unas pocas.

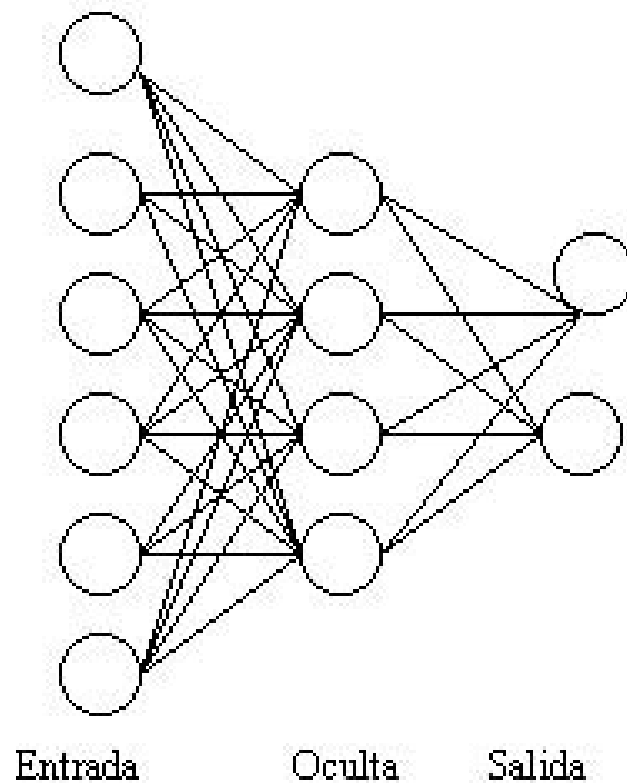


Figura 3.6: Grafo de una RNA

Redes neuronales convolucionales

Este tipo de redes neuronales son muy parecidas a las RNA normales pero tienen dos cambios significativos. El primero es que parten de la idea de que las entradas son imágenes lo que permite programar optimizaciones a los algoritmos para que se adecuen a este tipo de entradas, el segundo cambio es que intentan reducir en la mayor medida posible el número de parametros utilizados en cada capa.^[4]

Las redes neuronales convolucionales organizan sus neuronas en capas de 3 dimensiones normalmente coincidiendo con las características de la imagen, siendo height la altura de la foto, width el ancho y depth los colores de la imagen. Por ejemplo se podría aplicar a imágenes de 32x32x3.

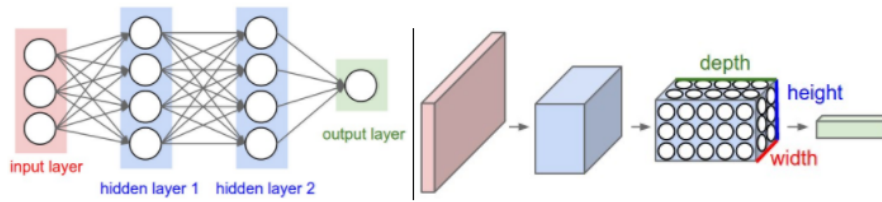


Figura 3.7: Izquierda: grafo de una RNA. Derecha: Grafo de una Red neuronal convolutiva.

Tipos de capas

En este apartado se van a explicar los tres tipos de capas más importantes a la hora de elaborar las redes neuronales convolucionales:

Capas convolucionales

En estas capas los píxeles de salida son combinaciones lineales de diferentes píxeles de entrada lo que generan nuevos mapas de píxeles más sencillos y que no son tan sensibles a cambios en las entradas.[11] Las convoluciones se pueden producir teniendo en cuenta solo 2 dimensiones (el alto y ancho) o teniendo en cuenta las 3 dimensiones (alto, ancho y profundidad). En algunas arquitecturas son llamadas capas RELU, ejecutan la función de activación de las neuronas.

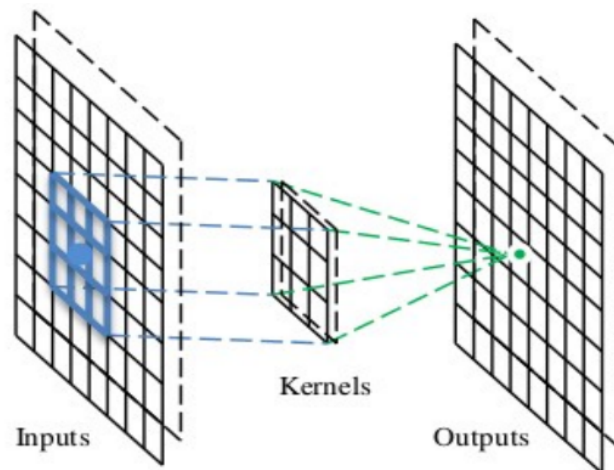


Figura 3.8: Figura que muestra una convolución en dos dimensiones.

Como vemos en la figura 3.8 Los kernels definen el área sobre el que se va a aplicar la convolucion, en el ejemplo 3x3.

Capas de pooling

Las capas de pooling o reescalado tiene el objetivo de disminuir el número de parámetros o pixeles de los mapas, para ello se aplican técnicas como la explicada en la siguiente figura.

Estas capas van a tener un gran peso en los modelos de Mobilenet e Inception, ya que nos van a permitir ajustar el número de operaciones realizadas por los modelos para que se puedan ajustar correctamente a las limitaciones hardware de los teléfonos móviles.

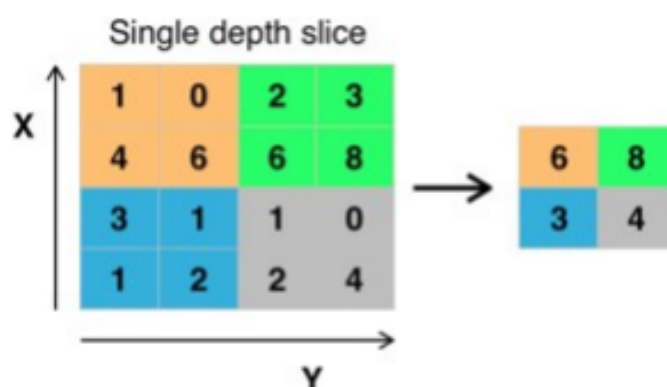


Figura 3.9: Técnica de máx pooling.

En la figura 3.9 se puede observar que los diferentes pixeles se agrupan eligiendo sólo aquellos de mayor valor para el nuevo mapa.

Capas totalmente conectadas

En este tipo de capas las neuronas están totalmente conectadas con las neuronas de la capa anterior y tienen una profundidad igual al número de clases que se quieran clasificar. Por ejemplo si tenemos 10 clases diferentes, el último layer podría ser de tamaño 32x32x10. Estas capas son las encargadas de producir la señal de salida y clasificar la imagen eligiendo una de las clases.

Cuando re entrenamos un modelo, normalmente lo que estamos haciendo es modificar estas últimas capas que se encargan de clasificar las clases en vez de tener que re entrenar todas las demás capas que forman el modelo.

3.7. Modelo Inception

El modelo Inception es una red neuronal convolucional que se usa para la clasificación de imágenes. Inception ha sido desarrollado por Google y parte de la pregunta de que tipo de convolución aplicar en cada capa o filtro del modelo. Es decir, preguntarse que tamaño de convolución sería el adecuado para cada etapa, ¿3x3?, ¿1x1?, ¿5x5?.[2]

Ante esta pregunta, la respuesta que han aplicado es la de ejecutar los diferentes tamaños de convoluciones en cada capa de manera paralela y concatenar los diferentes mapas en uno solo que es el que se pasa a la siguiente capa del modelo. La idea es ejecutar todos los tipos de convoluciones a la vez y elegir el que mejor funcione.

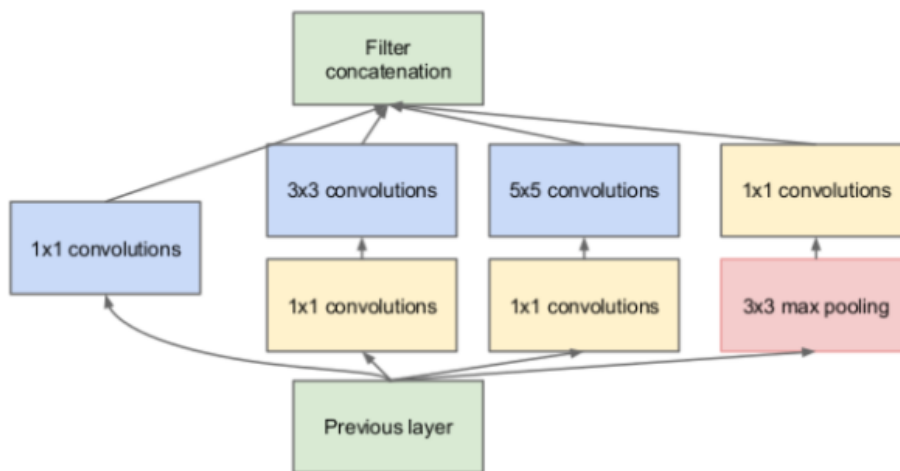


Figura 3.10: Ejemplo de un nodo Inception.

En la figura 3.10 podemos observar como se aplican las diferentes convoluciones en paralelo. Además podemos observar que se aplican unas convoluciones de 1x1, estas convoluciones tienen el objetivo de reducir el número de parámetros que se usan ya que aunque no reducen el área de los mapas si reducen la profundidad de las capas.[12]

Los modelos inception se construyen concatenando estos bloques o nodos Inception, en teoría cuanto más profundidad se da a estos modelos y más se concatenen más precisos deberían ser los modelos pero esto tiene un coste alto de computación (Aún reduciendo los computos con las convoluciones 1x1) y se corre el riesgo de sobreajustar los modelos.

Actualmente se pueden entrenar diferentes versiones de Inception, en el proyecto hemos trabajado reentrenando la versión 3 del modelo mediante Tensorflow.

Este modelo se sigue desarrollando y los desarrolladores lo van evolucionando por lo que no es de extrañar que en un futuro cercano aparezcan nuevas versiones mas optimizadas de este modelo.

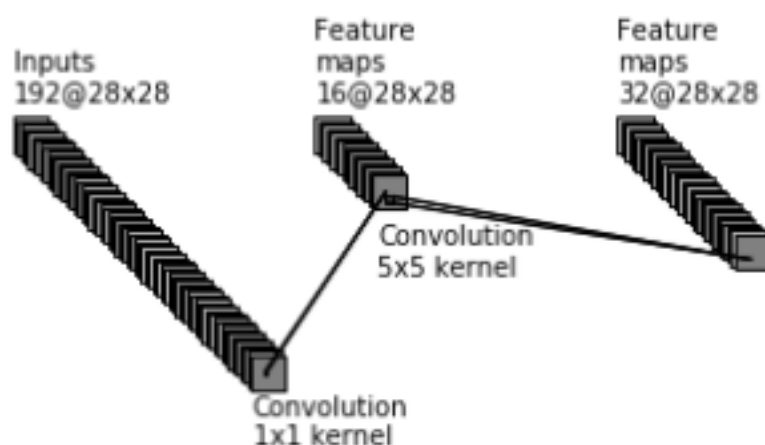


Figura 3.11: Aplicación de una convolucion 1x1 antes de ejecutar la convolucion 5x5.

En la figura 3.11 se puede observar el efecto de aplicar una convolucion 1x1 antes de efectuar la convolucion 5x5.

3.8. Modelo Mobilenet

Mobilenet hace referencia a una serie de modelos clasificadores de imágenes desarrollados por Google para ser embebidos en dispositivos móviles o sistemas con pocos recursos. La filosofía de este tipo de clasificadores nace

en contraposición a la tendencia que se estaba implantando en las redes neuronales de crear redes profundas y complejas con el objetivo de mejorar la precisión dejando a un lado cuestiones como el tamaño del modelo o la rapidez.[9]

Los modelos Mobilenet intentan ser precisos pero a la vez ser modelos de poco tamaño y rápidos que se adecuen a sistemas que necesiten de estas ventajas, como pueden ser los automóviles autónomos, realidad aumentada, sistemas de reconocimiento, entre otros.

La arquitectura de este tipo de modelo, al igual que el modelo Inception explicado anteriormente, se basa en una red neuronal convolucional que se explicará en el siguiente apartado.

Arquitectura Modelo Mobilenet

La arquitectura del modelo Mobilenet se basa en dividir los filtros convolucionales típicos de una RNA totalmente conectada en dos nuevos tipos de filtros, los filtros convolucionales profundos (Del inglés Depthwise Convolutional Filters) y los filtros convolucionales puntuales (Del inglés Pointwise Convolution Filters).

El objetivo de esta división de filtros es el de minimizar el número de operaciones realizada. La convoluciones normales tienen el siguiente coste:

$$DK * DK * M * N * DF * DF^6$$

Los Depthwise Convolutional Filters tienen el siguiente coste computacional:

$$DK * DK * M * DF * DF$$

Y los Pointwise Convolution Filters tienen el siguiente coste:

$$M * N * DF * DF$$

Si expresamos la convolución como un proceso de dos pasos de filtrado y combinación de los dos filtros anteriores obtenemos la siguiente reducción de costes de computación:

$$(DK * DK * M * DF * DF + M * N * DF * DF) / (DK * DK * M * N * DF * DF) = (1/N) + (1/(DK * DK))$$

Lo que supone una reducción respecto a los filtros convolucionales normales. Esta reducción la podemos observar en la siguiente tabla ??

⁶DK=Tamaño del kernel (alto x ancho), M=Número de canales de entrada, N=número de canales de salida, DF=Tamaño del mapa de salida

Modelos	Imagenet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7 %	4866	29.3
MobileNet	70.6 %	569	4.2

Tabla 3.1: Depthwise Separable vs Full Convolution MobileNet

Estos dos tipos de filtros se aplican de forma separada y paralela como sucedía en el modelo Inception. Para ver como surgen estas convoluciones vamos a fijarnos en la siguiente figura [3.12](#).

El uso de estos tipos de filtros de forma paralela, más el uso de filtros de normalización y pooling, es lo que usa Mobilenet para ejecutarse de forma rápida y utilizando la menor cantidad de memoria posible.

3.9. Web Semántica

La Web semántica es una Web extendida desarrollada bajo los estándares W3C (World Wide Web Consortium). Esta Web extendida aplica metadatos a la información que recoge para que los usuarios puedan encontrar respuestas de una manera más rápida y sencilla gracias a esta información adicional.[\[14\]](#)

Estos metadatos son los que otorgan más semántica a la Web y permiten obtener soluciones a problemas habituales gracias al uso de una arquitectura común para toda la información.

Por ejemplo, en este proyecto, esta tecnología ha permitido automatizar todo el proceso de recogida de información de las diferentes especies de setas de una manera rápida y eficaz.

¿Para qué sirve?

A la Web actual que todos conocemos le podemos atribuir dos problemas que han surgido del enorme crecimiento que ha tenido. Estos problemas son una sobrecarga de información, mucha de la información esta repetida en diferentes lugares siendo la misma, y que la información no se muestra bajo una estructura común, sino de una forma heterogénea.

La web semántica busca solucionar estos problemas delegándolos en el software, este software es capaz de procesar su contenido, combinarlo y realizar deducciones lógicas para solucionar problemas habituales.

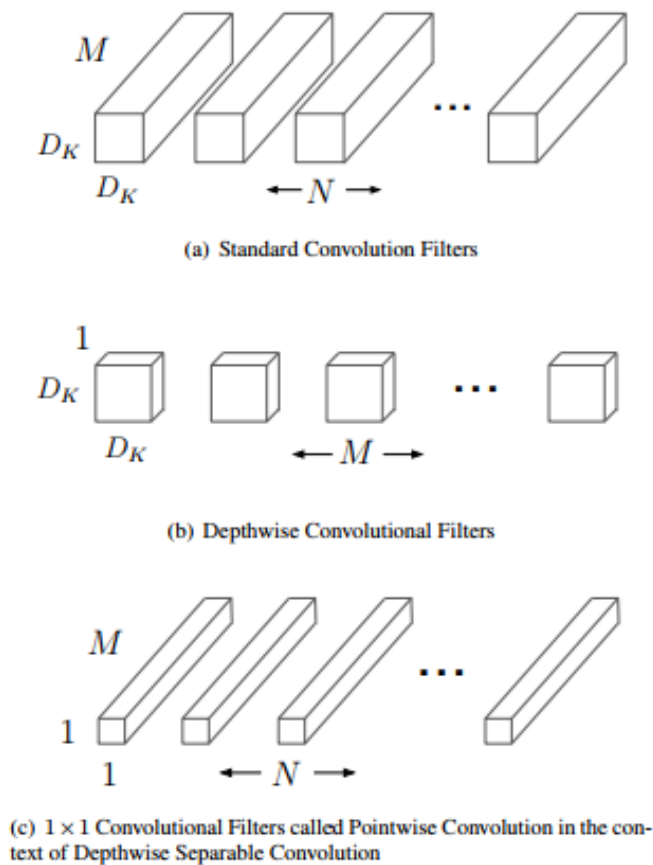


Figura 3.12: Los filtros convolucionales normales son divididos en los filtros profundos y los puntuales.

¿Cómo funciona?

El funcionamiento de la web semántica se basa esencialmente en RDF, SPARQL y OWL, mecanismos que permiten convertir la estructura de la Web para que se adecue al funcionamiento de una Web Semántica.

- RDF: RDF permite introducir metadatos descriptivos en los distintos tipos de información.
- SPARQL: Es el lenguaje de consultas que nos permite realizar búsquedas sobre la información que contiene estos metadatos.

- OWL:OWL permite crear vocabularios para asociar los diferentes recursos.

El uso conjunto de estos mecanismos es el que nos permite razonar sobre los datos de la Web.

3.10. Web Scraping

La herramienta del Web Scraping es una herramienta para extraer información o datos de páginas web de una manera rápida, eficiente y automatizada. La información extraída se presenta de una forma más estructurada y fácil de usar.^[1]

Los programas para realizar Web Scraping se pueden programar en diferentes lenguajes de programación, los más populares son Java, Python, Ruby y Node. Además existen diversos programas para realizar estas acciones mediante un interfaz gráfico.

Aproximadamente el 70 % de la información publicada en Internet esta en formato PDF, un formato no estructurado y difícil de manejar. Sin embargo, las páginas web si que tienen un formato estructurado ya que están programadas en código HTML, pero aun así no están presentadas de una manera totalmente reutilizable ya que no todas siguen el mismo esquema.

La técnica más extendida de Web Scraping es analizar la estructura del código HTML propio de una página web para poder extraer la información buscada. Por ejemplo, extraer todos los atributos text de los label <p>que se encuentren en una determinada página Web.

3.11. Clave Dicotómica

Las claves dicotómicas son herramientas que permiten identificar organismos. Las claves pueden determinar animales, plantas, hongos, moneras, protistas o cualquier otro ser vivo. Las claves pueden alcanzar diferentes profundidades identificando el género, especie, familia, entre otros.

La clave dicotómica se basa en ir mostrando al usuario diferentes dilemas (afirmaciones contrapuestas) entre los que deberá ir eligiendo el que más se adecue al organismo a identificar hasta alcanzar la categoría taxonómica⁷ deseada. Estas afirmaciones están enumeradas a lo largo de la clave.

⁷Grupos en los que se clasifican los seres vivos.

¿Cómo usar una clave?

Usar una clave consiste simplemente en leer los diferentes dilemas y optar solamente por uno de ellos. El dilema elegido no volverá a aparecer en el desarrollo de la clave. [20]

Técnicas y herramientas

4.1. Java

Java es un lenguaje de programación desarrollado por James Gosling de Sun Microsystems y publicado en 1995.

Es un lenguaje de programación orientado a objetos, de propósito general y orientado a objetos que fue diseñado con el objetivo de tener la menor cantidad de dependencias de implementación posibles. Un programa java es un programa multiplataforma que solo necesita ser compilado una vez para ser ejecutado en diferentes máquinas. Esto se consigue ya que el programa se ejecuta sobre una máquina virtual java(JVM), elemento que debe estar disponible en el sistema sobre el que se va a ejecutar el programa.

Java cuenta con una gran cantidad de documentación en línea así como una gran variedad de librerías de código abierto creadas por la comunidad que extienden el uso de Java a muchas aplicaciones diferentes, mejorando la funcionalidad original de Java. En este proyecto por ejemplo nos ha servido para realizar la parte de Web Scraping y de la Web Semántica.

Url de la herramienta: https://www.java.com/es/download/faq/whatis_java.xml

4.2. Eclipse

Eclipse es un entorno de desarrollo integrado (IDE, Integrated Development Environment) disponible para varios lenguajes de programación aunque el de uso más extendido sea el de Java.

En este proyecto se ha trabajado con el IDE de Java denominado Java EE (Eclipse IDE for Java EE Developers) en su versión "Neon.3".

Url de la herramienta: [<https://www.eclipse.org/ide/>](https://www.eclipse.org/ide/)

4.3. Android

Android es un sistema operativo basado en Linux. Fue diseñado con el objetivo de operar en dispositivos móviles con pantalla táctil, empezando por los teléfonos inteligentes y tabletas y después extendiéndose a nuevos dispositivos como relojes inteligentes, televisores y automóviles. Fue desarrollado por Android Inc., empresa respaldada económicamente por Google que en 2005 la compró. Android fue se lanzo en 2007 junto a la fundación Open Handset Alliance, que es un consorcio de compañías hardware, software y de telecomunicaciones.[15]

Los principales componentes del sistema operativo Android son los siguientes:

- Aplicaciones: Todas están escritas en el lenguaje de programación Java.
- Marco de trabajo de aplicaciones: Todos los desarrolladores tienen acceso al mismo entorno de trabajo donde cualquier aplicación puede publicar sus capacidades y luego cualquier otra aplicación puede hacer uso de estas capacidades. Este mecanismo a su vez permite que los componentes sean reemplazados por el usuario.
- Bibliotecas: Android dispone de una serie de bibliotecas del sistema escritas en C/C++ cuyas características se ponen a disposición de los desarrolladores a través del marco de trabajo.
- Runtime de Android: Android proporciona un set de librerías que proporcional la funcionalidad base de Java. Cada aplicación se ejecuta en una máquina virtual Dalvik.
- Núcleo Linux: Android depende de Linux para los servicios base del sistema.

Url de la herramienta: [<https://developer.android.com/index.html?hl=es-419>](https://developer.android.com/index.html?hl=es-419)

4.4. Android Studio

Android Studio es el entorno de desarrollo integrado(IDE) oficial de Android. Tiene el objetivo de facilitar el desarrollo en Android proporcionando herramientas personalizadas a los usuarios así como herramientas completas de edición, depuración, pruebas y perfilamiento de código.

Url de la herramienta: <https://developer.android.com/studio/index.html?hl=es-419>

4.5. JavaScript

JavaScript es un lenguaje de programación orientado a objetos, basado en prototipos, débilmente tipado y dinámico. Puede ser aplicado sobre un documentos HTML con el fin de crear interactividad dinámica con las páginas Web.

JavaScript se desarrollo inicialmente con una sintaxis parecida a C y adopta nombres y convenciones del lenguaje de programación Java.[24]

Url de la herramienta: <https://www.javascript.com/>

4.6. SLF4J

SLF4J(The Simple Logging Facade for Java) proporciona una API de registro a Java mediante un patrón de fachada que se aplica a varios marcos de registro (logging frameworks) que permite al usuario mostrar el marco de registro deseado en el momento de despliegue de la aplicación.

SLF4J nos permite filtrar los mensajes de log para visualizar de una forma más oportuna la información que nos muestra Java.

Url de la herramienta: <https://www.slf4j.org/>

4.7. Apache Maven

Apache Maven es una herramienta para manejar e interpretar nuestros proyectos software mediante un archivo de configuración XML. Maven se basa en el concepto de *project object model (POM)* para describir como y en que orden se van a construir los elementos de un proyecto y para manejar las dependencias de ese proyecto.

Apache Maven permite manejar estas dependencias por Internet descargando y actualizando los diferentes repositorios.[16]

Url de la herramienta: [<https://maven.apache.org/>](https://maven.apache.org/)

4.8. RDF

El Resource Description Framework (RDF) es una definición de tipo de documento de XML, es decir, una serie de metadatos que usan XML para poder relacionar contenidos entre las diferentes páginas web.

Los objetos de información o recursos se describen a través de un conjunto de propiedades denominadas "descripción RDF"(<rdf:description>). Estas propiedades se pueden entender como atributos de los recursos correspondiéndose a los pares atributo-valor tradicionales.

En el modelo de datos de RDF podemos distinguir tres tipos de objetos:

- Recursos: Cualquier recurso Web identificado unívocamente por un URI.
- Propiedades: Son características o atributos usadas para describir los recursos. En ellas se definen los valores permitidos y sobre que recursos se pueden aplicar.
- Descripciones: Son una triada formada por un recurso, un nombre de propiedad y un valor para esa propiedad. (Sujeto, predicado y objeto). Ver siguiente figura 4.13



Figura 4.13: Descripciones RDF.

RDF usa la sintaxis básica de XML1.0. Ejemplo: «xmlns:rdf="http://www.w3.org/TR/REC-rdf-syntax"»

El modelo de datos y la sintaxis no son necesarios para definir la relación entre los diferentes recursos y predicados, por ello hace falta un esquema. Un esquema son una serie de informaciones relativas a la clase a la que pertenecen los recursos para establecer relaciones jerárquicas entre ellos.

4.9. Apache Jena

Apache Jena es un entorno de desarrollo de código abierto para Java cuyo objetivo es el de permitir construir aplicaciones que hagan uso de la Web semántica y del *Linked Data*⁸.

El framework hace uso de diferentes APIs para extraer y escribir información en grafos RDF⁹.[\[25\]](#)

La arquitectura de Apache Jena contempla las siguientes características:

- Arquitectura para manejar (leer, procesar, escribir) las ontologías RDF y OWL.
- Motor para razonar sobre estas ontologías RDF y OWL.

⁸Método por el cuál se publican datos estructurados para que puedan ser interconectados con facilidad.

⁹Modelo de datos para los metadatos basado en triadas de información

- API para almacenar las tripletas RDF en ficheros o en memoria.
- Motor de queries para realizar consultas bajo la especificación SPARQL.[8]

En la siguiente figura se muestra un esquema de la arquitectura de Apache Jena.

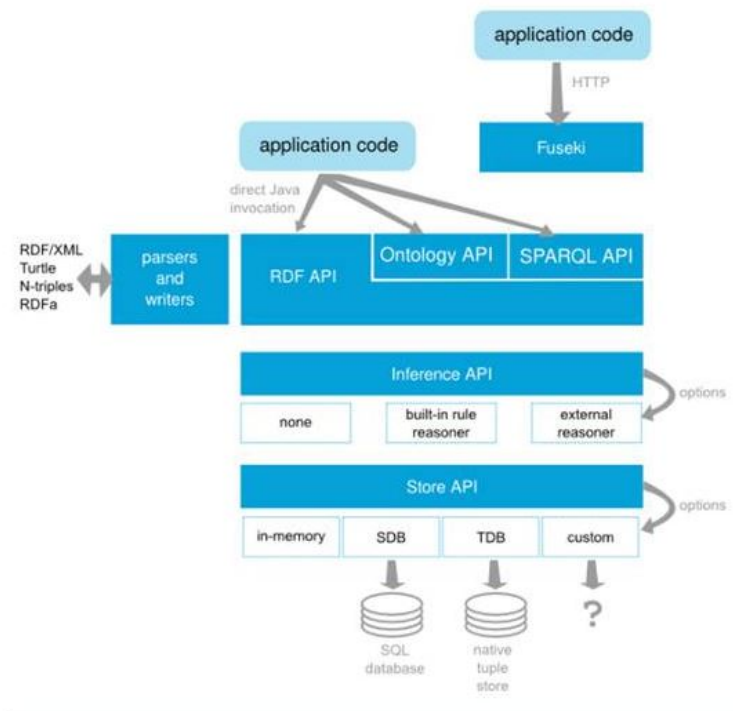


Figura 4.14: Arquitectura de Apache Jena

Url de la herramienta: [<https://jena.apache.org/>](https://jena.apache.org/)

4.10. JSON

JSON (acrónimo de JavaScript Object Notation) es un formato de texto ligero para el intercambio de datos. Es un subconjunto de la notación de JavaScript pero debido a la extensión de su uso como alternativa a XML se considera como un formato de lenguaje independiente.

La principal ventaja que ofrece JSON es la facilidad para crear analizadores sintácticos (parsers). [26]

Url de la herramienta: [<https://www.json.org/json-es.html>](https://www.json.org/json-es.html)

4.11. Jaunt

Jaunt es una librería gratuita de Java desarrollada para tareas de Web Scraping y automatización Web así como la realización de consultas JSON. Con Jaunt los programas Java pueden operar a nivel de navegador, documento o operaciones DOM (Del inglés Document Object Model).

JSON esta diseñado para realizar las siguientes tareas:

- Completar y enviar formularios Web.
- Crear programas automáticos Web o programas de Web-Scraping.
- Escribir clientes http para REST-APIS o Web-apps(JSON,HTML,XHTML o XML).

Url de la herramienta: [<http://jaunt-api.com/>](http://jaunt-api.com/)

4.12. SQLite

SQLite es un sistema de gestión de bases de datos relacional que contempla las características ACID(Atomicidad, Consistencia, Aislamiento,Durabilidad/Persistencia). SQLite esta contenida en una pequeña librería escrita en C que se enlaza con el propio programa pasando a ser parte del mismo, a diferencia de otros sistemas gestores de bases de datos que se ejecutan en un proceso separado e independiente al programa.

SQLite destaca por crear bases de datos ligeras que se almacenan en un único fichero, en la propia máquina donde se ejecuta. Además reduce la latencia ya que los accesos a la base de datos se realizan mediante subrutinas

o funciones que implementan la funcionalidad SQL, en vez de realizar comunicación entre procesos como ocurre con otros gestores.[30]

Url de la herramienta: [<https://www.sqlite.org/>](https://www.sqlite.org/)

4.13. SPARQL

SPARQL (Del inglés Protocol and RDF Query Language) es un lenguaje de consultas para grafos RDF. SPARQL permite crear consultas sobre diferentes fuentes que contengan la información nativamente en formato RDF o que la muestren en este formato mediante algún middleware.[10]

SPARQL soporta una serie de capacidades sobre los grafos como las uniones, intersecciones, agregaciones, sub-consultas entre otros. Los resultados obtenidos por estas consultas pueden ser o nuevos grafos RDF o *Result sets*.

Url de la herramienta: [<https://www.w3.org/TR/rdf-sparql-query/>](https://www.w3.org/TR/rdf-sparql-query/)

4.14. DBpedia

La DBpedia es un proyecto comunitario que tiene como objetivo poder extraer de forma estructurada la información contenida en la Wikipedia y crear una Web semántica con esta información. Este proyecto esta suendo realizado por la Universidad de Leipzig, Universidad Libre de Berlín y la compañía OpenLink Software. [21]

La DBpedia se fundamente en la estructura RDF y las consultas en SPARQL para lograr estos objetivos.

Url de la herramienta: [<http://wiki.dbpedia.org/>](http://wiki.dbpedia.org/)

4.15. Tensorflow

Tensorflow es una biblioteca de código abierto para realizar cálculos numéricos usando grafos de flujo de datos. Los nodos de los grafos representan operaciones matemáticas mientras que los enlaces representan conjuntos de datos multidimensionales que relacionan los nodos. Estos arrays son referidos como "Tensores".[31]

La arquitectura flexible de Tensorflow permite ejecutar los cálculos en una o varias CPUs o GPUs, ya sea en servidores, ordenadores de sobremesa o dispositivos móviles mediante una sola API.

Tensorflow ha sido desarrollado para crear y entrenar diferentes modelos de redes neuronales.

TensorFlow fue desarrollado originalmente por investigadores e ingenieros que trabajan en el equipo Brain de Google dentro de la organización de investigación de Inteligencia Artificial de Google para realizar investigaciones de aprendizaje automático y redes neuronales profundas.

Actualmente el sistema es lo suficientemente general para aplicarse a otra serie de dominios.

Url de la herramienta: [<https://www.tensorflow.org/>](https://www.tensorflow.org/)

4.16. Git

Git es un sistema de control de versiones distribuido de código abierto. El control de versiones nos permitirá retornar a algún punto anterior del desarrollo de nuestra aplicación en caso de sufrir errores o pérdidas de información y nos permitirá llevar un registro de cuando y por quién se modifican los archivos en todo momento.

Actualmente Git es uno de los sistemas más utilizados para este propósito y nos proporciona todas las herramientas que necesitamos para llevar el control de versiones de una manera simple y eficiente.

Url de la herramienta: [<https://git-scm.com/>](https://git-scm.com/)

4.17. GitHub

Github es un sistema de código abierto que nos permite alojar nuestro repositorio central del proyecto usando el control de versiones Git. Además permite la integración con diferentes herramientas como Zenhub, que se explicará a continuación, y nos ofrece las características que necesitamos para llevar a cabo esta tarea.

Url de la herramienta: [<https://github.com/>](https://github.com/)

4.18. Zenhub

Zenhub es una herramienta de gestión de proyectos integrada nativamente con GitHub, lo que hará que no necesitemos configuraciones extra a la hora de integrarla en nuestro repositorio.

Esta herramienta nos ayuda a la hora de planificar nuestros proyectos mediante un tablero kanban integrado y una serie de diagramas (Burndown y Velocity tracking) que se corresponden con las historias de usuarios y tareas que hayamos definido en nuestro proyecto.

Con el podremos definir puntos de historia en nuestras tareas para poder planificar de una manera más eficaz nuestros sprints.

Url de la herramienta: [<https://github.com/marketplace/zenhub>](https://github.com/marketplace/zenhub)

4.19. SourceTree

SourceTree es una herramienta que nos proporciona una interfaz gráfica para poder realizar las tareas relacionadas con el control de versiones de nuestro repositorio, como pueden ser realizar operaciones de commits, pull, push, Fetch, Branch, Merge entre otras.

SourceTree nos permite ver de una manera fácil y cómoda las ramas de nuestro proyecto, así como todos los commits realizados y los cambios producidos en cada archivo modificado.

Url de la herramienta: [<https://es.atlassian.com/software/sourcetree>](https://es.atlassian.com/software/sourcetree)

4.20. Latex

Latex es un sistema de composición de textos de código abierto, orientado a crear documentos con una alta calidad tipográfica, como pueden ser artículos o libros científicos.[27]

Url de la herramienta: [<https://www.latex-project.org/>](https://www.latex-project.org/)

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros³, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] Osmar Castrillo-Fernández. Web scraping: Applications and tools, Diciembre, 2015. [Internet; descargado 12-noviembre-2017].
- [2] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.
- [3] CONCEPTODEFINICION.DE. Definición de micología, 2015. [Internet; descargado 28-noviembre-2017].
- [4] cs231n. Convolutional neural networks (cnns / convnets). [Internet; descargado 30-noviembre-2017].
- [5] Universidad de Salamanca. Redes neuronales. [Internet; descargado 9-noviembre-2017].
- [6] dipualba. El reino de los hongos, 2010. [Internet; descargado 8-noviembre-2017].
- [7] Florencio. Las setas y sus características. [Internet; descargado 8-noviembre-2017].
- [8] Luis Miguel Gracia. Qué es apache jena, 27 julio 2012. [Internet; descargado 15-noviembre-2017].
- [9] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

- [10] SPARQL 1.1 Query Language. Steve harris, garlik, a part of experian andy seaborne, the apache software foundation, 21 Marzo 2013. [Internet; descargado 30-noviembre-2017].
- [11] Patricio Loncomilla. Deep learning: Redes convolucionales, 2016. [Internet; descargado 30-noviembre-2017].
- [12] Tommy Mulc. Inception modules: explained and implemented, 25 Septiembre 2016. [Internet; descargado 30-noviembre-2017].
- [13] Sergio Ventura. El proceso en la minería de datos, 18 julio 2011. [Internet; descargado 11-noviembre-2017].
- [14] W3C. Guía breve de web semántica. [Internet; descargado 12-noviembre-2017].
- [15] Wikipedia. Android — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 13-noviembre-2017].
- [16] Wikipedia. Apache maven — wikipedia, the free encyclopedia, 2017. [Online; accessed 13-November-2017].
- [17] Wikipedia. Aprendizaje automático — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 10-noviembre-2017].
- [18] Wikipedia. Aprendizaje no supervisado — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 10-noviembre-2017].
- [19] Wikipedia. Aprendizaje supervisado — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 10-noviembre-2017].
- [20] Wikipedia. Clave dicotómica — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 12-noviembre-2017].
- [21] Wikipedia. Dbpedia — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 30-noviembre-2017].
- [22] Wikipedia. Fungi — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 8-noviembre-2017].
- [23] Wikipedia. Inteligencia artificial — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 10-noviembre-2017].
- [24] Wikipedia. Javascript — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 13-noviembre-2017].

- [25] Wikipedia. Jena (framework) — wikipedia, the free encyclopedia, 2017. [Online; accessed 13-November-2017].
- [26] Wikipedia. Json — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 13-noviembre-2017].
- [27] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 30-noviembre-2017].
- [28] Wikipedia. Micología — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 8-noviembre-2017].
- [29] Wikipedia. Red neuronal artificial — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 9-noviembre-2017].
- [30] Wikipedia. Sqlite — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 13-noviembre-2017].
- [31] Wikipedia. Tensorflow — wikipedia, la enciclopedia libre, 2017. [Internet; descargado 30-noviembre-2017].