Analítica para la toma de decisiones: Redes Neuronales

David Diaz Rodríguez

Nicolas Niño Valderrama

Valentina Jiménez Torres

Daniel Perea Mercado

Estudiantes de Ingeniería Industrial

Manuela Londoño Ocampo

Docente

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería Industrial

2024

**Introducción**

Actualmente, los parques naturales y jardínes botánicos juegan un papel importante en la conservación de la biodiversidad y la educación botánica y ambiental. Sin embargo, la identificación de las diferentes especies de flores carece de precisión y necesita de talento humano necesariamente capacitado y experimentado para ello. Así mismo, se presentan limitaciones del factor humano; escasez de personal especializado, algunos parques y jardínes tienen personal limitado y no especializado. Limitaciones de tiempo; poco tiempo disponible para atender a los usuarios en la identificación de plantas y flores. Subjetividad de la identificación; principiantes y expertos botánicos pueden tener dificultades en la identificación de ciertas especies. Error humano; la identificación manual es propensa a errores lo que puede afectar la gestión en parques y jardínes y, la alta afluencia de público a estos lugares dificulta la gestión debido a la escasez de guías de campo, lo que hace que este proceso sea lento e inexacto.

Por otra parte, la correcta identificación de especies florales es indispensable en investigación, conservación y educación ambiental. Un sistema preciso y confiable para dicha identificación es trascendental para garantizar la integridad de la información y el éxito de la conservación. Entre tanto, el avance de la Inteligencia Artificial y el aprendizaje profundo ofrece soluciones de eficiencia y precisión en el proceso de identificación de flores, el cual puede ser usado en la gestión de visitantes de parques naturales y jardínes botánicos, programas educativos, gestión de recursos naturales e impacto en la experiencia del visitante.

**Problema:** Los usuarios de parques naturales y jardínes botánicos presentan dificultades para identificar diferentes especies de flores, inicialmente; Margarita, Diente de León, Rosa, Girasol y Tulipán. Además, los parques naturales (parque arví) y jardínes botánicos (en Medellín) presentan problemas en la gestión de clientes, solo cuentan con colaboradores (algunos no tan experimentados en botánica) que resultan insuficientes para atender la alta demanda de visitantes a estos lugares y hacer que su recorrido botánico se convierta en una experiencia satisfactoria.

**Solución:** Automatizar la identificación de especies de flores, inicialmente; Margarita, Diente de León, Rosa, Girasol y Tulipán. Entrenar un red neuronal que permita clasificar e identificar su especie. Esta red neuronal sería implementada en una aplicación móvil y usada por los usuarios para identificar las cinco especies de flores; Margarita, Diente de León, Rosa, Girasol y Tulipán, a partir de una imagen tomada con un dispositivo móvil o cargada por el usuario. Los usuarios de parques naturales y jardines botánicos pueden usar la aplicación para identificar rápidamente cada flor durante su visita. Esto permite optimizar la gestión de cliente, gestionar una identificación más eficiente y precisa, disminuir los errores humanos, reducir los costos en este proceso, proporcionar información adicional a los usuarios sobre la especie identificada y mejorar su experiencia en el recorrido.

Para el desarrollo de este proyecto se usa un data set tipo imagen, encontrado e importado desde kaggle.com el cual nos proporciona datos a cerca de 5 tipos de flores Margarita, Diente de León, Rosa, Girasol y Tulipán. En el análisis exploratorio se encuentran datos importantes que nos ayudan a comprender mejor el data set; 3670 observaciones y 2 columnas compuesto por 5 clases de rosas, dandelion 898 datos que representan un 24,4% , tulips 799 datos con un 21,77% , sunflowers 669 datos (19.05%), roses 641 datos (17,47%) y Daisy 633 datos (17,25%). Para estos datos se determina hacer un balanceo de clases con el fin de que la métrica tenga un mejor desempeño. Además, se hace la separación en datos considerando un porcentaje para cada uno, datos de entrenamiento 70%, datos de validación 20% y datos de testeo 10% y por último se realiza una estandarización de los datos esto, ayuda a que todas las características contribuyan de manera equitativa al proceso de aprendizaje, acelera el entrenamiento, evita el dominio de características y ayuda a la red neuronal a predecir con más exactitud.

Ahora, en el ajuste de una red neuronal convolucional (CNN) diseñada para procesar imágenes RGB de tamaño 32x32. Su estructura incluye las siguientes capas:

* Capa Convolucional: La primera capa aplica 32 filtros con un tamaño de kernel de 3×3, utilizando la función de activación ReLU. Se especifica un desplazamiento (𝑠𝑡𝑟𝑖𝑑𝑒) de 2×2, lo que reduce la dimensionalidad de la entrada. Esta capa es capaz de extraer características locales como bordes y texturas de las imágenes.
* Capa de Agrupación (MaxPooling): Reduce la dimensionalidad espacial de las características extraídas mediante un proceso de agrupación máxima con una ventana de tamaño 2×2 y un desplazamiento de 2×2. Esto ayuda a disminuir el tamaño de los datos y a mantener las características más relevantes.
* Capa de Aplanamiento (Flatten): Convierte el tensor bidimensional resultante en un vector unidimensional para conectarlo con las capas completamente conectadas (densas) posteriores.
* Capa Densa Oculta: Una capa totalmente conectada con 256 unidades y activación ReLU. Esta capa procesa las características extraídas, permitiendo que la red capture relaciones no lineales.
* Capa de Salida: Una capa totalmente conectada con 5 unidades y activación softmax.

Es así como esta arquitectura tiene 1,715,909 de parámetros y se hace la representación esquemática de la arquitectura, para entender mejor la arquitectura de la red neorunal. Así mismo, se entrena la red con 10 epochs en donde nos entrega resultados de accuracy el cual aumenta constantemente en el entrenamiento a lo largo de las 10 epochs alcanzando un 0.97 , la precisión del conjunto de validación se mantiene baja disminuyendo en las últimas epochs y en la función de pérdida aumenta considerablemente es así, como podemos decir que la red está memorizando los datos de entrenamiento en vez de aprender además, una perdida baja en el conjunto de entrenamiento y una perdida alta en validación es una característica de overfitting. Entre tanto, al utilizar el metodo evaluate para evaluar la red neuronal los resultados nos muestran una precisión del conjunto de prueba de 65.8% y la pérdida es de 1.5 que nos indica un alto valor de error en la predicción

Para la optimización de hiperparámetros se hace una búsqueda a través de Random Search con el objetivo de mejorar la precisión de la validación, en dicho procedimiento se realizaron 7 pruebas obteniendo el mejor accuracy de 0.28, la precisión del conjunto de validación aumenta durante las 5 epochs llegando a 0.54 no siendo este un valor muy alto y no supera al valor de la validación de forma significativa. Estas diferencias se reducen notablemente y no se nota el sobreajuste anterior.

Entonces, como solución para el sobreajuste que se está presentando usamos la técnica de regularización que disminuye la complejidad del modelo y lo hace menos sensible al ruido para los datos de entrenamiento lo que sirve para que el aprendizaje del modelo en las características sea de manera general esto, lleva a reducir el sobreajuste.

Entonces, usamos la técnica de regularización Dropout que obliga a la red neuronal a aprender representaciones de características más robustas y que no dependan de una neurona. Además, es una herramienta muy eficaz para corregir el sobreajuste.

Posteriormente, se empela la optimización de hiperparámetros y se entrena la red con los mejores hiperparámetros encontrados. Así mismo, obtenemos resultados del entrenamiento durante 10 epochs en donde, los epochs 1 al 3 se nota que la precisión aumenta rápidamente en el conjunto de entrenamiento que en el conjunto de validación. Por otro lado, los epochs 4 al 7 la precisión crece en ambos conjuntos e indica que el modelo está aprendiendo lento pero consistente. Finalmente, en los epochs 8 al 10 la precisión del conjunto sigue mejorando mientras, que el conjunto de validación aumenta poco y es posible que se esté acercando a su límite de aprendizaje. Por tanto, el uso de Dropout parece que está siendo efectivo para prevenir el sobreajuste esto, debido a que la diferencia entre el entrenamiento y la validación es pequeña en el accuracy. De todas maneras, la precisión final (0.85 en entrenamiento y 0.67 en validación podría mejorarse. Finalmente, la función los muestra que el modelo está aprendiendo de forma efectiva y está generalizando los datos, la disminución en sus valores de los y validación los indican un entrenamiento correcto y confirma que la precisión del modelo está mejorando y que el uso de Dropout ha sido beneficioso.

Por otra parte, entrenamos el primer modelo con la regularización L2,

El trabajo consiste en desarrollar los siguientes puntos:

a. Descripción del problema de negocio: Incluir una descripción del problema a resolver,

puede ser un caso de estudio hipotético planteado en un contexto empresarial que permita

entender por qué es un problema para la empresa y los beneficios o posible impacto que

tendría resolverlo. Un ejemplo de un problema de negocio es: reducir costos, automatizar

procesos, aumentar las ventas, etc.

b. Proponer una solución: a partir del planteamiento realizado en el punto anterior, el equipo

debe proponer una solución analítica, en la que va desarrollar un modelo predictivo (redes

neuronales) y va a explicar cómo ese modelo va a servir para resolver o apoyar la solución

del problema.

c. Limpieza y transformación de datos: Aplicar limpieza y transformación según la revisión

realizada a los datos elegidos y que considere pertinente para los demás puntos a

desarrollar.

d. Ajustar redes neuronales: Pruebe una arquitectura de red neuronal con sus hiperparámetros,

explicando su elección. Con base en los resultados que obtuvo en esta primera corrida, vaya

modificando la arquitectura e hiperparámetros para mejorar el desempeño. Lo principal en

este punto, más allá del resultado, son las explicaciones a la arquitectura de red neuronal

propuesta y el análisis de los resultados preliminares obtenidos. Algunos elementos se

deben decidir al azar, pero es importante tener un criterio de selección claro para la mayoría

de elementos. que se van definiendo.

e. Afinar hiperparámetros: Después de encontrar unas arquitecturas e hiperparámetros

prometedores en el punto anterior, realice un afinamiento de hiperparámetros con una grilla

pequeña para que el proceso de búsqueda no. Tome mucho tiempo. Finalmente, deje una

red neuronal ganadora.

f. Ajustar modelo de Shallow Learning: Escoja un modelo de shallow learning y compare los

resultados de este modelo con los que obtuvo con la red neuronal ganadora. Explique por

qué decidió ajustar el modelo elegido.

g. Conclusiones y resultados: Analice los resultados que obtuvo, genere conclusiones

relacionadas con el ajuste del modelo y otras relacionadas con la solución del problema

planteado al inicio.