Aplicación de técnicas de extracción de características en problemas de clasificación de imágenes

Autor: Lucas Fortun Iñurrieta

NIA: 140006

Asignatura: Visión Artificial

Índice

Introducción y objetivos

La motivación de este trabajo es crear un sistema que ayude a clasificar diferentes tipos de animales utilizando un sistema de clasificación supervisada basado en la extracción de características.

El objetivo principal es implementar varios modelos que aprendan de estos datos y puedan predecir a qué grupo pertenece una nueva imagen, utilizando las características aprendidas de cada clase.

Este trabajo se divide en cinco partes:

1. **Construcción de los algoritmos de extracción de características:**
   En este caso se van a desarrollar dos algoritmos para capturar las características que determinan la clasificación de una imagen en diferentes clases. Uno de ellos se centra en la generación de un histograma a partir de números binarios transformados a decimales, mientras que el otro se enfoca en la construcción de un histograma mediante la derivación.

2. **Preprocesado de datos:**
   El dataset abarca imágenes de 4 categorías de animales con variaciones en tamaño y color. Para optimizar la extracción de características y facilitar el entrenamiento de los modelos, se implementarán técnicas para estandarizar el formato de todas las imágenes, garantizando uniformidad en el conjunto de datos.

3. **Implementación y entrenamiento de modelos:**
   Se emplearán diversos modelos de aprendizaje automático, y se buscarán los hiperparámetros óptimos a través de la metodología de entrenamiento, validación y prueba. Estos modelos serán entrenados utilizando las características obtenidas en el proceso de extracción.

4. **Validación del sistema:**
   Se procederá a evaluar la eficiencia y capacidad de generalización de cada modelo utilizando los dos algoritmos de extracción de características, seleccionando el mejor rendimiento obtenido para cada uno de ellos.

5. **Comparativa de resultados:**
   Finalmente, se llevará a cabo un análisis comparativo de los resultados obtenidos, determinando cuál de los algoritmos de extracción ha demostrado un rendimiento superior.

Algoritmo de extracción de características 1

Implementación

Este algoritmo de extracción de características inicia dividiendo la imagen en bloques disjuntos de NxN píxeles, denominados celdas. Para cada píxel en la celda, se realiza una comparación con sus 8 vecinos, reemplazando sus valores con 0 o 1 según si son menores o mayores que el píxel central. Estos valores se convierten en un número binario que, al transformarse a decimal, genera un número entre 0 y 255. Posteriormente, se sustituye el valor del píxel por el decimal obtenido. Después de aplicar este proceso a cada píxel en la celda, se construye un histograma con los NxN píxeles resultantes. El valor del histograma se convierte en el vector de características asociado a la celda.

Finalmente, al concatenar todos los histogramas de la imagen según el orden de las celdas, se obtiene el vector de características completo de la imagen.

Modificaciones

Se ha implementado una mejora al algoritmo mediante la introducción del concepto de transición uniforme. En este escenario, los números binarios generados que presenten más de dos cambios de 0 a 1 o de 1 a 0 se identificarán como transiciones no-uniformes. Todas estas transiciones se codificarán de manera uniforme y se asignarán al mismo valor en el histograma. Esta implementación tiene como objetivo reducir la complejidad del histograma, limitándolo a un valor máximo de 59 (Imágenes utilizadas 128x128).

La introducción del concepto de transición uniforme en el algoritmo de extracción de características proporciona mejoras significativas en la simplicidad y compacidad del histograma resultante. Al considerar únicamente las transiciones no-uniformes y asignarles un único valor en el histograma, se logra una reducción la dimensionalidad del conjunto de características. Además de llevar a una interpretación más eficiente y discriminativa de las características, esta modificación facilita el proceso de clasificación de imágenes.

Algoritmo de extracción de características 2

Implementación

Este algoritmo se inicia aplicando un filtro espacial de aproximación de la derivada parcial en ambas direcciones mediante máscaras de convolución específicas (-1, 0, 1 y -1, 0, 1). Posteriormente, se calcula la magnitud y orientación del gradiente, asegurándose de que todas las orientaciones estén en el rango de 0 a 180 grados. Luego, al igual que en el algoritmo anterior, la imagen se divide en bloques disjuntos de NxN píxeles, conocidos como celdas.

A partir de esto se llevará a cabo la construcción del histograma formado por 9 bins que representan distintos intervalos de orientaciones (El primer bin servirá para las orientaciones entre 0 y 20 grados, el segundo para las orientaciones entre 20 y 40, y así sucesivamente hasta las orientaciones entre 160 y 180). La particularidad aquí es que el histograma se llena no mediante un conteo de orientaciones, sino utilizando la suma acumulativa del valor de la magnitud del gradiente de los pixeles con orientación entre dichos rangos. Después de esta operación, se normaliza el histograma dividiéndolo por la norma euclidiana de los 9 valores.

Finalmente, al concatenar todos los histogramas de la imagen, siguiendo el orden de las celdas, se obtiene el vector de características completo que representa la imagen en su totalidad.

Modificaciones

Se ha implementado una mejora al algoritmo mediante la normalización del histograma de las celdas, considerando los histogramas de las celdas vecinas. Se define el concepto de bloque como un conjunto de MxM. Los MxM histogramas (9*MxM valores) se normalizan por la norma de dichos valores. Además, se añade el desplazamiento los bloques de MxM celdas de manera incremental, en este caso se ha implementado un incremento de 1 en 1, generando un solapamiento al calcular las normalizaciones.

Esta estrategia amplía el vector de características al permitir el solapamiento, lo que, a diferencia de la modificación del algoritmo anterior, aumenta su tamaño. Sin embargo, este enfoque tiene el potencial de proporcionar características más detalladas y de mayor calidad.

Conjunto de datos y particionado

El dataset está formado por diversas imágenes de 4 clases, las cuales son: gatos, elefantes, caballos y arañas. Este dataset ya se encuentra dividido en 3 conjuntos distintos, concretamente 3 conjuntos:

- **Test:** Este conjunto se utilizará para analizar la generalización de los modelos, es decir, para ver que tan bien predicen los modelos con datos/ejemplos que no ha visto en el entrenamiento.

- **Train_100:** Este conjunto será el de validación, su función será ajustar los hiperparamteros y validar el rendimiento de los modelos durante el entrenamiento. El conjunto consta de 100 imágenes de cada clase.

- **Train_500:** Este conjunto será el conjunto de entrenamiento, su función será que los distintos modelos aprendan las características de cada clase para así en un saber clasificarlas. El conjunto consta de 500 imágenes de cada clase.

Lo que he comentado ahora son los usos teóricos de cada conjunto, sin embargo, he decidido juntar train_100 y train_500, en un único conjunto de entrenamiento. Esto se debe a que a la hora de entrenar los modelo voy aplicar GridSearch de la librería sklearn.model_selection el cual me permitirá encontrar los mejores hiperparamtros para cada modelo realizando un validación cruzada. Concretamente utilizare el parámetro cv = 5, el cual realizara una validación cruzada de 5 pliegues evaluando y entrenando el modelo 5 veces con 5 conjuntos distintos, es por lo que no necesitare un conjunto de validación separado, ya que esta validación se realizara sobre el conjunto de entrenamiento, el cual contendrá 2400 ejemplos tras la unión de train_100 y train_500.

Preprocesado de imágenes

Como mencioné previamente, las imágenes en nuestro conjunto de datos no siguen un estándar, es decir, difieren en cuanto a su color y tamaño. Por esta razón, resulta muy importante llevar a cabo un proceso de preprocesamiento de imágenes antes de extraer sus características. Este procedimiento garantiza que todas las imágenes estén en condiciones uniformes, siendo tratadas de manera equitativa sin asignar una importancia distinta a imágenes debido a sus variaciones en tamaño o color. La estandarización de las imágenes establece una base homogénea, fundamental para obtener resultados coherentes y significativos durante la extracción de características y con ello permitir un mejor entrenamiento de los modelos y posterior clasificación.

Concretamente, en el proceso de preprocesamiento de imágenes, se ha implementado una función que transforma las imágenes de entrada. Primero, cada imagen se convierte a escala de grises mediante la función rgb2gray. Luego, se redimensiona a un nuevo tamaño especificado, concretamente se ha elegido 128x128 píxeles, utilizando la función resize de la biblioteca scikit-image. El resultado final es una lista de imágenes procesadas, preparadas para ser utilizadas en la extracción de características y posterior entrenamiento de modelos, contribuyendo así a la homogeneización y mejor manejo de los datos.

Clasificación

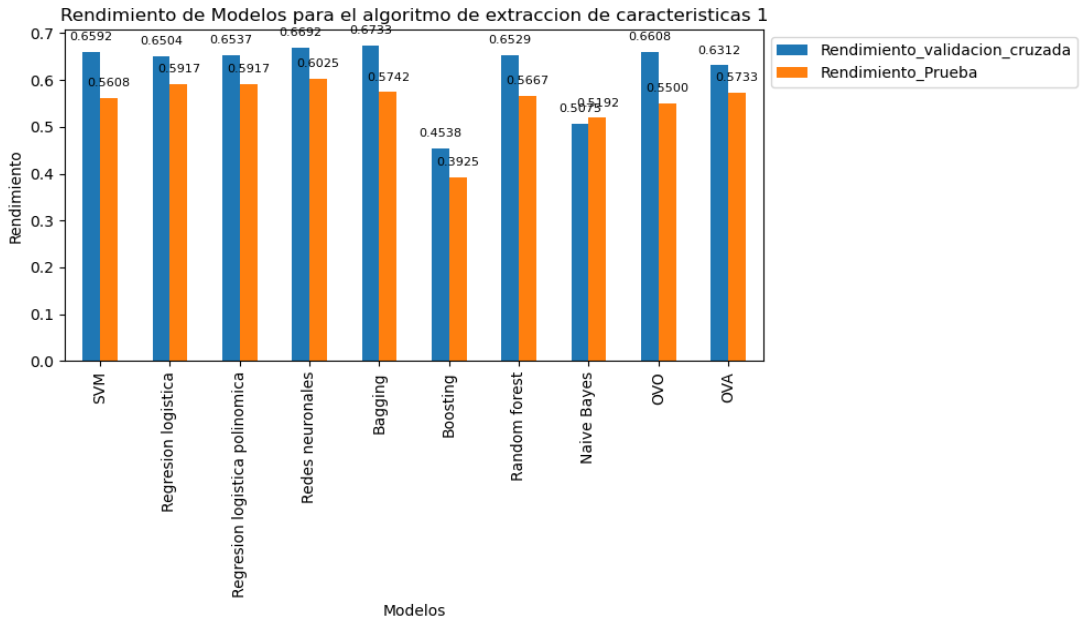
Modelos implementados

1. **SVM (Support vector machine):**
   Las Máquinas de Soporte Vectorial (SVM) son algoritmos de aprendizaje supervisado utilizados para clasificación y regresión. Buscan encontrar un hiperplano que mejor separe las clases en el espacio de características. Se centran en maximizar el margen entre clases.
2. **Regresión logística:**
   La regresión logística es un modelo utilizado comúnmente para problemas de clasificación binaria, donde el objetivo es predecir una de dos clases posibles mediante funciones logísticas que transforman las salidas en un numero entre 0 y 1. Sin embargo, cuando se trata de problemas de clasificación multiclase, se pueden realizar extensiones de la regresión logística para manejar varias clases.
3. **Regresión logística con variables polinómicas:**
   La Regresión Logística con variables polinómicas es una extensión de la Regresión Logística que permite modelar relaciones no lineales entre las características y la variable objetivo. Utiliza la función logística para prever probabilidades de pertenencia a una clase y puede incluir características polinómicas para capturar patrones más complejos.
4. **Redes neuronales:**
   Las redes neuronales son modelos de aprendizaje automático que imitan el funcionamiento del cerebro humano. Consisten en capas de nodos (neuronas) conectados, donde cada conexión tiene un peso. Las redes neuronales aprenden ajustando estos pesos durante el entrenamiento para hacer predicciones o clasificaciones. Utilizan funciones de activación para introducir no linealidades y pueden tener capas ocultas para aprender representaciones más complejas.
5. **Bagging:**
   Bagging (Bootstrap Aggregating) es una técnica de ensamblado que combina múltiples modelos para mejorar la estabilidad y precisión de las predicciones. En el contexto de Bagging, se crean múltiples conjuntos de datos de entrenamiento mediante muestreo con reemplazo (bootstrap) y se entrena un modelo base en cada conjunto. Luego, las predicciones de cada modelo se promedian (en el caso de regresión) o se vota (en el caso de clasificación) para obtener una predicción final más robusta y generalizable.
6. **Boosting (AdaBoost):**
   Boosting, como AdaBoost, es un método de ensamblado que mejora modelos débiles. Se ajusta iterativamente a los datos, dando más peso a las instancias mal clasificadas en cada paso. Combina modelos débiles para formar un modelo fuerte.

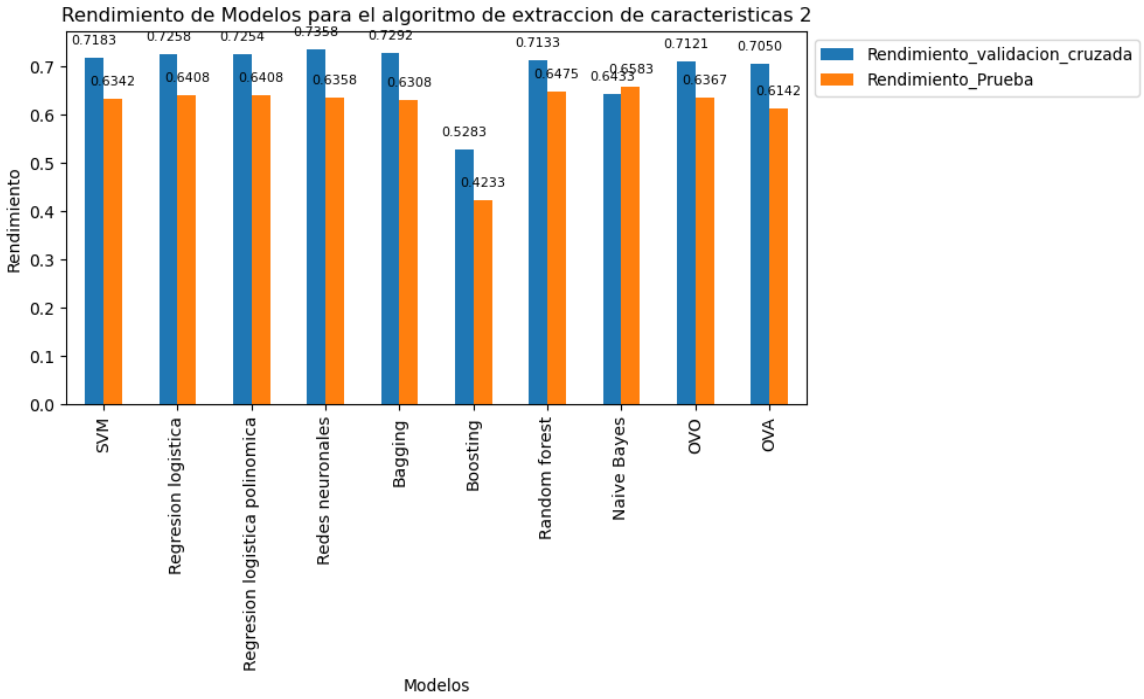
7. **Random forest:**
   Random Forest es un algoritmo de ensamblado que construye múltiples árboles de decisión durante el entrenamiento y los combina para obtener una predicción más robusta. Cada árbol se entrena en un conjunto de datos diferente y utiliza subconjuntos aleatorios de características. Luego, las predicciones se combinan mediante votación (clasificación) o promedio (regresión).
8. **Naive Bayes:**
   Naive Bayes es un algoritmo de clasificación probabilístico basado en el teorema de Bayes, asumiendo independencia condicional entre las características. Se calculan las probabilidades condicionales de pertenencia a cada clase dada la observación de las características. Luego, se realiza la clasificación seleccionando la clase con la probabilidad más alta.
9. **OVO (One vs One):**
   One-vs-One (OVO) es un enfoque de clasificación multiclase que implica entrenar clasificadores binarios para cada par único de clases. En lugar de abordar directamente la clasificación en un problema de varias clases, OVO descompone el problema en subproblemas más manejables. Durante la predicción, cada clasificador binario emite su veredicto y la clase que recibe más votos positivos se considera la predicción final.
10. **OVA (One vs All):**
    One-vs-All (OVA), también conocido como One-vs-Rest (OVR), es un método de clasificación multiclase que simplifica el problema al entrenar un clasificador binario para cada clase, considerando dicha clase como positiva y el resto como negativas. Durante la predicción, se evalúa cada clasificador y se selecciona la clase asociada al clasificador con la probabilidad más alta.

Resultados

- **Algoritmo de extracción de características 1**



- **Algoritmo de extracción de características 2**



Análisis de los resultados

Previamente al análisis de los resultados mencionar que los mejores hiperparametros obtenidos para cada modelo se reflejan en el notebook.

Al observar los resultados de los modelos para el primer algoritmo de extracción, se puede ver que el modelo de Bagging ha demostrado tener el rendimiento más alto en la validación cruzada (100 estimadores) con un rendimiento de 0.6733, mientras que el modelo de Boosting tuvo el rendimiento más bajo con 0.4537. Sin embargo, cuando se trata del rendimiento del conjunto prueba, el modelo de Redes Neuronales (Dos capas ocultas de 50 neuronas) superó a los demás con un rendimiento de 0.6025, y nuevamente, el modelo de Boosting quedó atrás con el rendimiento más bajo de 0.3925.

Por otro lado, al observar los resultados de los modelos para el segundo algoritmo de extracción, se puede ver que, el modelo de Redes Neuronales (Dos capas ocultas de 50 neuronas) ha demostrado tener el rendimiento más alto en la validación cruzada con un rendimiento de 0.7358. Sin embargo, el modelo de Boosting tuvo el rendimiento más bajo en la validación cruzada con 0.5283. En cuanto al rendimiento del conjunto prueba, el modelo de Regresión Logística (parámetro de regularización C = 1) y Regresión Logística Polinómica (parámetro de regularización C = 1 y grado = 1) empataron con el rendimiento más alto de 0.6408, mientras que el modelo de Boosting tuvo el rendimiento más bajo con 0.4233.

Conclusión

En este trabajo, se ha abordado el problema de clasificación de imágenes de animales mediante la aplicación de técnicas de extracción de características. Se han implementado dos algoritmos de extracción de características distintos, cada uno con sus respectivas implementaciones y modificaciones.

El primer algoritmo se ha centrado en la generación de histogramas a partir de la transformación de píxeles a números binarios, con la introducción de mejoras como el concepto de transición uniforme. Por otro lado, el segundo algoritmo se ha basado en la aproximación de la derivada parcial y la construcción de histogramas de orientaciones con la incorporación de la normalización de histogramas de celdas vecinas.

En la fase de clasificación, se han implementado diversos modelos de aprendizaje automático, desde SVM y regresión lineal hasta redes neuronales y técnicas de ensamblado como Bagging y Boosting. Se han realizado pruebas y se han evaluado los modelos utilizando los dos algoritmos de extracción de características.

Los resultados han mostrado que el rendimiento de los modelos variaba según el algoritmo de extracción de características utilizado. Priorizando el rendimiento obtenido en los conjuntos de validación cruzada, el modelo que ha obtenido mejores resultados para el primer algoritmo de extracción de características ha sido Bagging, con un rendimiento en la validación cruzada del 67,33% y un 57,42% para el conjunto de prueba, mientras que, para el segundo algoritmo, ha sido el modelo de Redes Neuronales con un rendimiento en la validación cruzada del 73,58% y un 63,58% para el conjunto de prueba.

En general, se ha observado que la elección del algoritmo de extracción de características tiene un impacto significativo en el rendimiento de los modelos. La introducción de mejoras, como la transición uniforme en el primer algoritmo y la normalización de histogramas en el segundo, han demostrado ser beneficioso en términos de simplicidad, compacidad y, en algunos casos, rendimiento. Sin embargo, los datos son claros en cuanto a rendimiento, es por ello, que, para este problema, si tenemos en cuenta todos los rendimientos obtenidos para cada algoritmo de extracción, el algoritmo de extracción de características que mejor rendimiento ha dado ha sido el segundo. Además, si no nos ceñimos al total de los rendimientos de todos los modelos si no que nos fijamos solo en el mejor de cada algoritmo, el algoritmo de extracción 2 sigue siendo el que mejor rendimiento ha dado, y por tanto el más óptimo para este problema, concretamente utilizando el modelo de redes neuronales con 2 capas de 50 neuronas cada 1.