

Clasificación de cartas de una baraja utilizando redes neuronales convolucionales

Emiliano Vásquez Olea – A01707035

1. Introducción

En el entorno del aprendizaje profundo y redes neuronales, encontramos una gran diversidad de métodos y prácticas para resolver diferentes problemas. Con la libertad de implementar redes de distintos tamaños y con diferentes operaciones, de igual forma surgen aquellos métodos que ven mejores resultados en el procesamiento de imágenes.

Debido a que los datos definen el problema que estamos abarcando, es crucial identificar estas mismas técnicas, integrando las investigaciones y conocimientos pasados con las adaptaciones necesarias para abarcar el tipo de datos que se maneje.

El modelo que se presenta en esta entrega tiene como objetivo clasificar el tipo de carta (de una baraja con 53 cartas) a partir de su imagen. El *dataset* utilizado se encuentra en la plataforma *Kaggle* dentro de la siguiente liga: <https://www.kaggle.com/datasets/gpiose nka/cards-image-datasetclassification>.

2. Información del conjunto de datos

El *dataset* está compuesto por una serie de carpetas y documentos, para el desarrollo de este modelo nos enfocamos en las tres carpetas *test*, *train* y *valid*, que contienen una serie de imágenes correspondientes a las

diferentes cartas de una baraja, incluyendo el Joker. Entre los demás documentos encontrados en *Kaggle* vemos un archivo llamado *cards.csv*, que contiene información sobre cada imagen, ubicación y label, sin embargo, gracias a diferentes herramientas de procesamiento, no es necesario utilizar estos registros.

Como podemos ver, el conjunto de imágenes ya se encuentra agrupado en subconjuntos de entrenamiento, validación y prueba, que, cómo se menciona en la descripción del *dataset*, se han distribuido de la siguiente forma:

- *train*: 7624 imágenes de entrenamiento.
- *valid*: 265 imágenes para validación.
- *test*: 265 imágenes para pruebas.

Cada uno de estos conjuntos sigue la misma estructura, con 53 distintas carpetas que indican las diferentes clases o tipos de cartas que existen, estás también son las distintas clases que se toman en cuenta dentro del modelo de predicción.

Es importante mencionar que los grupos para cada tipo de carta, a pesar de no contar con el mismo número de imágenes, se mantienen en un rango mayor a 100 y menor a 200 para el conjunto de entrenamiento. Esto nos ayuda a indicar que hay un balance entre la cantidad de registros por clase. Mientras tanto, para los conjuntos de

validación y prueba se presentan 5 imágenes por clase.

Los conjuntos de entrenamiento y validación son utilizados en el ajuste de parámetros e hiperparámetros, mientras que el conjunto de prueba permite evaluar el modelo final con datos que no han sido vistos por el modelo.

3. Modelo de clasificación

Como fue mencionado anteriormente, el modelo utilizado es una red neuronal convolucional, la cuál clasificará la clase o tipo de carta a partir de una imagen. La separación entre clases ya se ve representada por la división en carpetas del *dataset*, pero dentro del modelo estas son asignadas a un valor entero para su predicción.

Las redes neuronales convolucionales están especializadas en el manejo de datos que pueden ser representados como una matriz, por ejemplo, las imágenes. La diferencia principal entre las redes convolucionales y redes neuronales generales es, como su nombre lo indica, el uso de capas convolucionales, que permiten extraer diferentes características de una imagen generando nuevas matrices abstractas para el modelo. Estas capas de igual forma siguen un proceso de aprendizaje junto con el resto de la red y se suman al resto de capas densas o con otro tipo de operaciones para crear la red neuronal completa, finalizando con una capa de salida.

Para crear la red neuronal inicial se utiliza como base la arquitectura de *MobileNet V3*, una red convolucional relativamente ligera debido a su enfoque para el procesamiento dentro de teléfonos celulares. Integrando este modelo previamente utilizado y probado en la clasificación de otros tipos de imágenes, es posible contar con una base robusta con diferentes operaciones y capacidad de extracción de información de las imágenes, sin embargo, para completar la red neuronal y realizar predicciones, es necesario agregar algunas capas finales al modelo. Después de la base de *MobileNet* se agrega una capa de *Global Average Pooling* y dos capas densas, una con 128 neuronas y la función de activación *ReLU*, y la última teniendo el *output* final de 53 clases.

A comparación de otras redes convolucionales que pudieron ser utilizadas como base, como *VGG* o *ResNet*, *MobileNet* permite contar con una arquitectura de menor tamaño para su uso en una mayor variedad de dispositivos.

4. Resultados y Evaluación

Utilizando los parámetros iniciales (mencionados en *README.md*) y entrenando el modelo con el conjunto de datos de entrenamiento, se obtienen los siguientes resultados de precisión y tiempo de entrenamiento al finalizar la ejecución del método *fit*:

- Un puntaje del 94.9% de precisión para los datos de entrenamiento.

- Puntaje de 77.7% con los datos de validación.
- Puntaje de 87.5% para los datos de prueba.
- Tiempo de entrenamiento de 39 minutos.

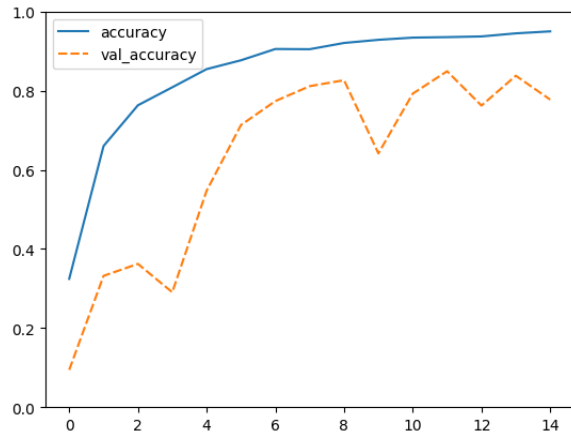


Fig 1. Precisión a lo largo de las épocas con parámetros iniciales.

Vemos que la precisión en los subconjuntos de validación y pruebas es menor que en el entrenamiento. Se realizarán una serie de cambios a los hiperparámetros o arquitectura del modelo para mejorar estos puntajes.

5. Ajuste de Parámetros

Se exploraron una serie de cambios sobre los parámetros iniciales del modelo de clasificación. A continuación, se presentan los resultados en precisión y tiempo de ejecución al evaluar distintos ajustes:

Uno de estos valores básicos que podemos modificar es el número de iteraciones para el entrenamiento del modelo (*epochs*). Al incrementar de 15 a 20 épocas vemos una mayor precisión sobre el conjunto de validación de 84.9%,

sin embargo, también se incrementó el tiempo de entrenamiento a alrededor de 55 minutos.

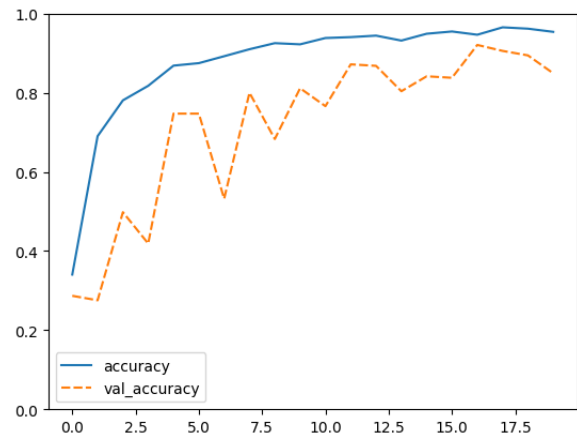


Fig 2. Precisión al entrenar con un número mayor de iteraciones.

Ahora, probando una diferente cantidad de pasos por iteración (*steps_per_epoch*) podemos ver de igual forma, un mayor tiempo de entrenamiento, llegando a una hora con 20 minutos. A pesar de este mayor tiempo de ejecución, la precisión con el conjunto de validación llegó a un 93.5% Eliminando el límite de 200 para este parámetro, pasa a un valor de 382.

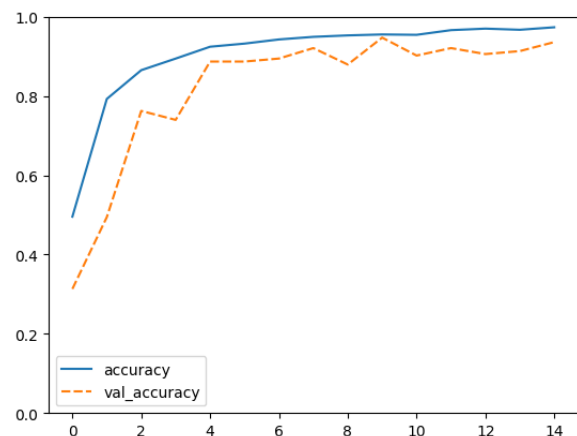


Fig 3. Precisión al eliminar el límite de pasos por iteración

Por último, al implementar un cambio en la arquitectura, similar a con los cambios anteriores, se vio una mejora en precisión junto con un incremento en el tiempo de entrenamiento, además de agregar complejidad al modelo. La precisión con el subconjunto de validación pasó a 89%, mientras que el tiempo de entrenamiento se incrementó a 44 minutos. Este cambio consistió en agregar una capa densa adicional de 64 neuronas previo a la capa de salida.

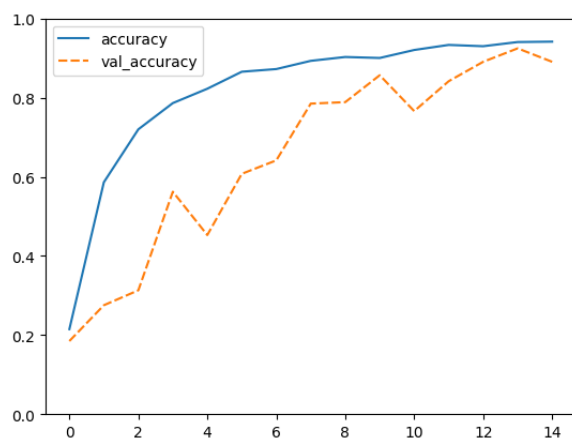


Fig 4. Resultados con un cambio en la arquitectura de la red neuronal.

Estas pruebas fueron aplicadas sobre el mismo código de entrenamiento inicial, manteniendo los demás parámetros igual que con el modelo inicial. El *notebook* final de la entrega contiene la última versión seleccionada para el modelo.

6. Conclusión y resultados finales

El ajuste seleccionado de las pruebas anteriores fue el cambio sobre la arquitectura de la red neuronal. Este cambio, a pesar de representar un impacto muy pequeño en la complejidad del modelo, fue capaz de mejorar de forma considerable la precisión con los datos de validación. Además, el tiempo

de entrenamiento no presentó un cambio tan alto en comparación con el modelo generado a partir de los parámetros iniciales.

Al agregar esta capa densa en el modelo de clasificación, se obtienen los siguientes resultados:

- Precisión del 94% para los datos de entrenamiento.
- Precisión de 89% con los datos de validación.
- Precisión de 97.5% para los datos de prueba.
- Tiempo de entrenamiento de 44 minutos

Vemos que ambos resultados sobre el conjunto de pruebas y de validación presentaron mejoras notables, con el conjunto de pruebas obteniendo el mejor puntaje. Es posible que otros parámetros u otros valores presenten mejoras a estos resultados, sin embargo, es posible que de igual forma creen un impacto sobre los tiempos de entrenamiento o peso del modelo.

Este modelo de clasificación con redes neuronales convolucionales muestra ser confiable al identificar tipos de carta en una imagen, sin embargo, algunos diseños o posiciones de las cartas pueden llegar a generar errores de predicción.