

MODELO DE CLASIFICACIÓN DE IMÁGENES CON MLP

Universidad Militar Nueva Granada

Fiorella Trujillo 1401519, José Torres 1401499

I. Introducción.

Este proyecto utilizó el dataset PathMNIST, parte de la colección MedMNIST, que contiene imágenes médicas de histopatología de 28×28 píxeles distribuidas en nueve clases. El objetivo fue diseñar un modelo MLP capaz de clasificarlas de la mejor manera posible. Para ello, se optimizaron hiperparámetros con Optuna como número de neuronas, tasa de aprendizaje y tamaño de lote y se evaluó su desempeño en entrenamiento, validación y prueba, considerando tanto su capacidad de generalización como sus limitaciones.

II. Criterios de diseño y características del modelo.

Para este proyecto de simulación se utilizó el dataset PathMNIST de la librería MedMNIST, el cual contiene imágenes médicas de histopatología con resolución de 28×28 píxeles en RGB (3 canales) y está dividido en 9 clases. El objetivo fue diseñar un modelo de tipo MLP (Multi-Layer Perceptron) para clasificar correctamente cada imagen en su clase.

Para el preprocesamiento, se normalizaron los valores de píxeles entre -1 y 1, esto para mejorar la estabilidad del entrenamiento y acelerar la convergencia. Se trabajó con la división oficial del dataset en entrenamiento, validación y prueba, lo que nos asegura que los resultados sean representativos.

El modelo implementado fue un MLP sencillo con las siguientes características:

- Entrada de $28 \times 28 \times 3 = 2352$ características.
- Una capa densa con un número de neuronas variable (hidden dim).
- ReLU en la capa oculta, por su eficiencia y capacidad de aprendizaje no lineal (Función de activación).
- Capa de salida con 9 neuronas, una por clase, evaluadas con CrossEntropyLoss.
- Optimizador Adam, seleccionado ya que nos brinda rapidez y estabilidad en el ajuste de parámetros.

El MLP es un modelo básico que permite evaluar cómo un clasificador de redes densas procesa datos de imágenes.

III. Selección de hiperparámetros

La selección de hiperparámetros se realizó con la herramienta Optuna, probando distintas configuraciones y eligiendo la que maximizara la accuracy en validación.

Se probaron diez configuraciones (trials) variando tres parámetros principales:

- Número de neuronas ocultas (hidden dim): valores entre 64 y 512. Este parámetro controla la capacidad de representación del modelo. Con pocas neuronas el modelo

puede quedarse corto (subajuste), mientras que con demasiadas podría memorizar los datos (sobreajuste).

- Learning rate (lr): entre $1e-4$ y $1e-2$. Este valor define el tamaño de los pasos que da el optimizador Adam al ajustar los pesos. Un valor muy alto genera oscilaciones y pérdida de estabilidad; uno muy bajo hace que el modelo aprenda demasiado lento.
- Tamaño de lote (batch size): 64, 128 y 256. Este valor influye tanto en la estabilidad del gradiente como en el tiempo de entrenamiento. Los lotes grandes tienden a dar un gradiente más estable, pero consumen más memoria, mientras que los pequeños hacen que el modelo vea más variaciones en cada paso.

IV. Resultados

1. Resultados del barrido:

“Trial 8 finished with value: 0.6211515393842463 and parameters: {'hidden_dim': 512, 'lr': 0.00030963496104920445, 'batch_size': 128}. Best is trial 8 with value: 0.6211515393842463.”

- Los modelos con hidden dim bajos (64 y 128) tuvieron accuracy en validación cercana al 55 a 58%, lo que nos dice que capacidad de representación era insuficiente.
- Los modelos con valores intermedios (192 a 256) alcanzaron alrededor de 59 y 60% de accuracy en validación.
- El mejor desempeño se obtuvo con 512 neuronas ocultas, learning rate 0.00031 y batch size 128, logrando una accuracy de validación de 62.1%.

Este resultado muestra que un mayor número de neuronas ayudó a mejorar la representación de los datos, pero acompañado de un learning rate bajo, que evitó que el entrenamiento se volviera inestable, pero aprendía de manera lenta. El batch size de 128 ofreció un balance entre la estabilidad y la velocidad del entrenamiento.

Este proceso de selección de hiperparámetros nos sirvió tanto para encontrar los mejores valores, como para entender cómo cada decisión influía en el rendimiento final.

2. Resultados de generalización

Una vez definidos los hiperparámetros óptimos, se entrenó el modelo nuevamente durante 10 épocas. Los resultados obtenidos fueron:

- Accuracy en entrenamiento: 66.6%
- Accuracy en validación: 61.2%
- Accuracy en prueba (test): 60% aprox.

La diferencia entre entrenamiento y validación (aproximadamente del 5%) nos dice que puede ser un sobreajuste, es decir, que el modelo aprende mejor en los datos que ya ha visto que en los nuevos. A pesar de esto, la accuracy de prueba fue cercana a la de validación, entonces el modelo tiene una capacidad de generalización aceptable.

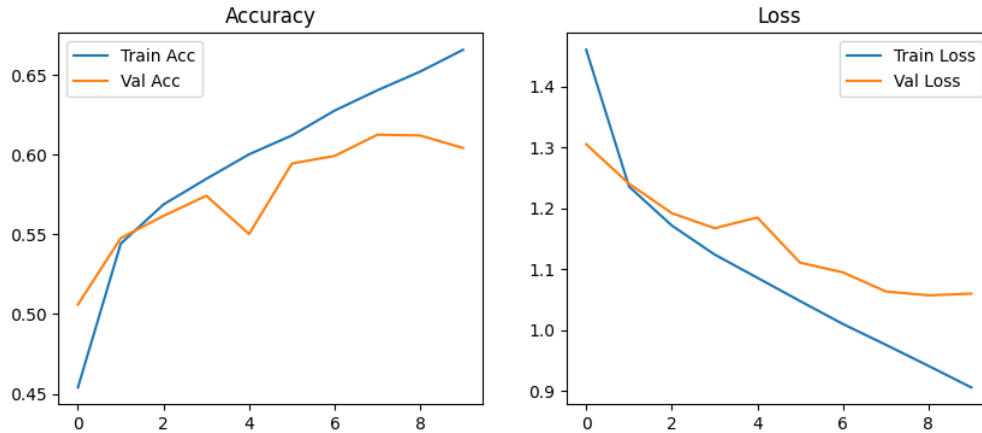


Fig. 1. Configuración del nombre para la primera antena.

En la comparación de las curvas de entrenamiento y validación (Fig. 1), se observa que ambas se estabilizan sin un colapso notable en validación, lo cual es una buena señal. El valor de loss en validación se mantiene más alto que el de entrenamiento, lo cual es esperado y nos dice a cerca de un sobreajuste leve. Un valor del 60% de accuracy no es muy alto, pero es lo esperado de un MLP simple en imágenes médicas, ya que suelen ser difíciles de clasificar por sus patrones complejos.

V. Conclusiones

El proceso de realización de este modelo nos ayudó a ver de manera concreta cómo modelar y probar un modelo de aprendizaje profundo para la clasificación de imágenes radiológicas. En primer lugar, comprobamos que una MLP básica, aunque no es la arquitectura más compleja para imágenes, podía alcanzar una precisión del 60 % o ligeramente superior, lo cual es razonable para un modelo básico.

Un hallazgo significativo fue el impacto directo de los hiperparámetros en el rendimiento del modelo. El uso de Optuna facilitó la selección automática de la configuración, y se logró el resultado óptimo con un número elevado de neuronas ocultas (512), una tasa de aprendizaje baja (0,00031) y un tamaño de lote medio (128). Esto respalda que la capacidad de representación y la estabilidad del entrenamiento son factores determinantes para aumentar la precisión.

En cuanto a la generalización, se observó un ligero sobreajuste, reflejado en la diferencia entre entrenamiento y validación. Sin embargo, el hecho de que el desempeño en prueba se mantuviera cercano indica que el modelo sí logró aprender patrones útiles sin depender únicamente de los datos de entrenamiento.

Por último, este proyecto destaca que, aunque MLP tiene sus inconvenientes en comparación con los datos visuales, es una buena base para comprender todo el proceso de un experimento de aprendizaje profundo: desde la preparación de los datos, pasando por el ajuste de los hiperparámetros, hasta el análisis de los resultados. Una clara implicación es que, para mejorar el rendimiento en conjuntos de datos como PathMNIST, sería conveniente investigar arquitecturas más sofisticadas.