

# PROPUESTA DE MEJORAS DEL ESTUDIO (MOLINA ET AL., 2021)

## PARTE 1: HIPERPARÁMETROS

CONTROL DE VERSIONES.

Versión	Descripción	Fecha de última modificación	Validado
1.0	Primera versión estable	25/03/21	No
1.1	Unificación en el formato	31/03/21	No

## TABLA DE CONTENIDO

Control de versiones.....	2
Introducción .....	4
Análisis de tamaño de lotes.....	4
Introducción e hipótesis .....	4
Planteamiento del problema.....	4
Resultados .....	5
Análisis del número de épocas .....	7
Introducción e hipótesis .....	7
Planteamiento del problema.....	7
Resultados .....	7
Aplicación del conocimiento .....	9
Conclusión .....	10
Bibliografía.....	11

## INTRODUCCIÓN

Este experimento constituye una propuesta de mejora al artículo (Molina et al., 2021) el cual tratamos en el experimento 0.

En el experimento 0, creamos la red SOCO, una red basada en la propuesta de arquitectura de (Molina et al., 2021). Solo se modificó lo estrictamente necesario, como la capa de salida.

Durante la realización del experimento, se detectaron algunos puntos dudosos como la selección de hiperparámetros. En este experimento pretendemos hacer una propuesta de mejora en función de la variación de estos.

Los hiperparámetros que vamos a estudiar son los siguientes:

- Tamaño del lote.
- Número de épocas.

## ANÁLISIS DE TAMAÑO DE LOTES

### INTRODUCCIÓN E HIPÓTESIS

Como apoyo durante esta primera parte del experimento nos basamos en las ideas del artículo (Kandel & Castelli, 2020), este artículo estudia la relación del tamaño del ratio de aprendizaje y el tamaño de lote, y concluye con el resultado de que un tamaño de lote estándar recomendable es de 32 imágenes y que el ratio de aprendizaje recomendable es 0.0001.

En (Molina et al., 2021) el tamaño de lote seleccionado es de 128 imágenes.

Este valor cumple una de las recomendaciones generales cuando se trabaja con CPU (usar un valor en base 2). No obstante, desoye recomendaciones sobre la relación de ratio de aprendizaje de optimizadores y tamaño de lotes.

Se plantea por tanto la siguiente hipótesis:

1. *Reducir el tamaño de lote puede incrementar la velocidad de convergencia y la precisión del modelo*

### PLANTEAMIENTO DEL PROBLEMA

Para comprobar si se cumple esta hipótesis y saber cuál es el tamaño del lote óptimo, realizaremos cuatro pruebas con distintos tamaños. Se usará la misma arquitectura y optimizador. Se realizarán 10 iteraciones con cada tamaño de lote de forma que los resultados sean robustos y se reduzca el efecto de la aleatoriedad lo máximo posible.

El esquema de pruebas será el siguiente:

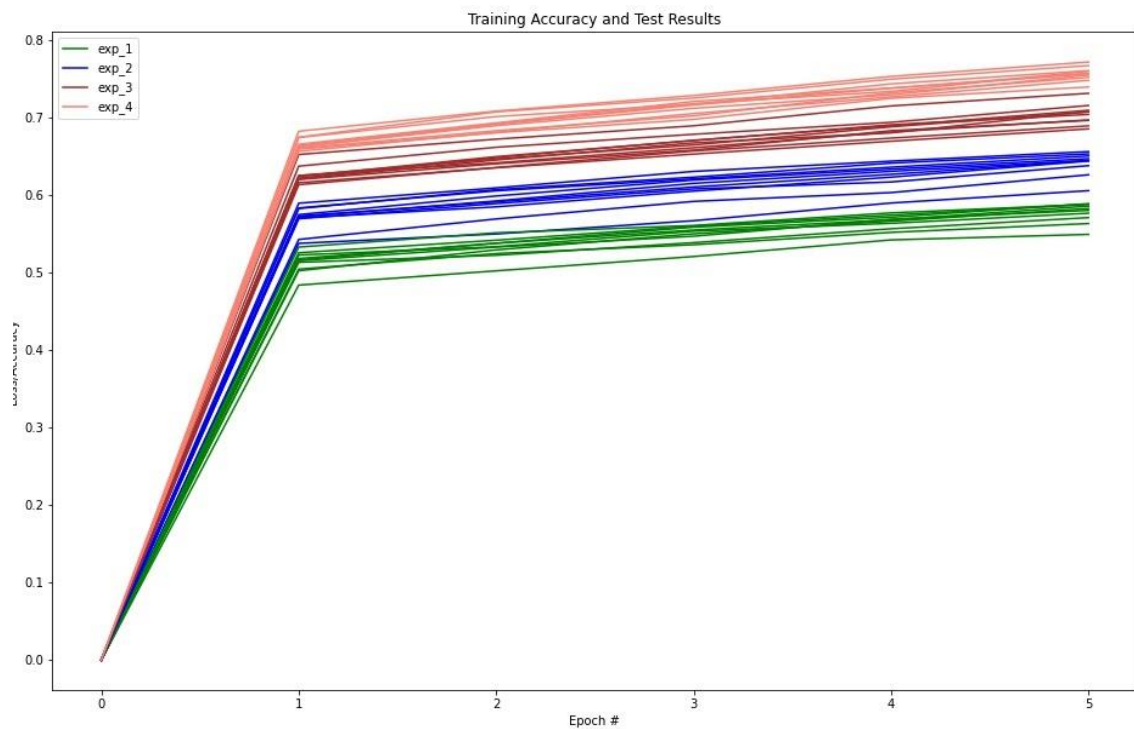
- exp\_1: Tamaño del lote = 128
- exp\_2: Tamaño del lote = 64
- exp\_3: Tamaño del lote = 32
- exp\_4: Tamaño del lote = 16

El esquema de entrenamiento que ha seguido ha sido el propuesto en el estudio criticado:

- El conjunto de entrenamiento = source (60% del total del conjunto de datos),
- El conjunto de reentrenamiento = train (70% del 40% restante del conjunto de datos),
- El conjunto de test = test (30% del 40% restante del conjunto de datos)

## RESULTADOS

En la siguiente imagen podemos observar el proceso de entrenamiento de la red (resultados sobre el test de entrenamiento) y cómo afecta el tamaño del lote.



Podemos apreciar que la rapidez con la que converge el modelo es inversamente proporcional al tamaño del lote.

No obstante, podría tratarse de un caso de sobreajuste, por ello a continuación mostramos una tabla en la que podemos encontrar los resultados obtenidos al evaluar contra el conjunto de test para cada iteración y para cada uno de los casos.

Iteración	Caso 1	Caso 2	Caso 3	Caso 4
1	0.478565	0.496063	0.556430	<b>0.720618</b>
2	0.496063	0.462234	0.605716	<b>0.762613</b>
3	0.512394	0.619131	0.579178	<b>0.624380</b>
4	0.429571	0.440653	0.652960	<b>0.736658</b>
5	0.419073	0.552056	0.714494	<b>0.748906</b>
6	0.468650	0.543015	0.686206	<b>0.695246</b>
7	0.403325	0.543015	0.513561	<b>0.569554</b>
8	0.482065	0.596967	0.643628	<b>0.682415</b>
9	0.426947	0.559055	0.689122	<b>0.710994</b>
10	0.417614	0.547098	0.668416	<b>0.685331</b>
Media	0.453427	0.535929	0.630971	<b>0.693672</b>

Como podemos ver en la tabla anterior los resultados son abrumadores. Para cada iteración el mejor resultado lo obtiene la red entrenada con un tamaño de lote de 16. La mejora con respecto a la configuración propuesta en (Molina et al., 2021) es de un 52%.

## ANÁLISIS DEL NÚMERO DE ÉPOCAS

### INTRODUCCIÓN E HIPÓTESIS

La hipótesis de partida en esta sección del experimento es:

1. *Al ser nuestro conjunto de datos más complejo, el número de épocas recomendadas en el artículo puede ser insuficiente para que el modelo tenga un desempeño óptimo. Por lo tanto, aumentar el número de épocas mejorará su precisión.*

### PLANTEAMIENTO DEL PROBLEMA

Se han planteado un experimento para el cuál los parámetros usados serán:

- RMSProp con ratio de aprendizaje de 0.0001 (recomendado por el estudio).
- El tamaño de lote será 128 (el recomendado por el estudio).
- El número de épocas será 30, tanto en el primer entrenamiento del modelo, como en la fase de transfer learning.

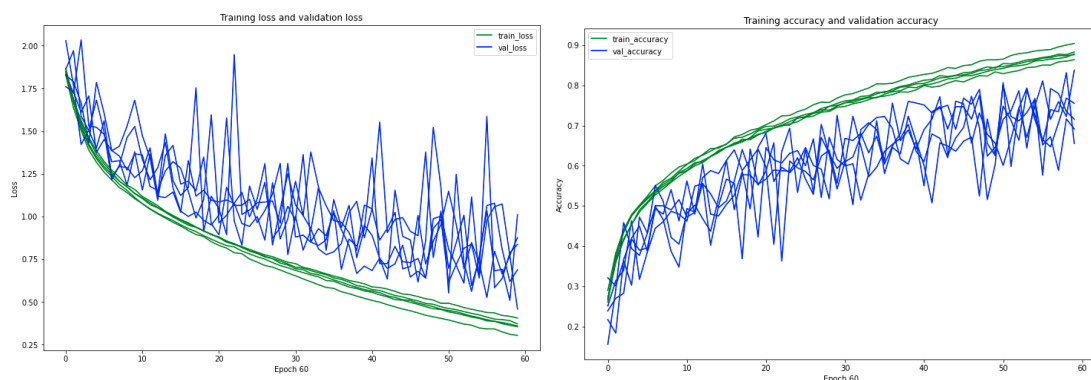
Se realizarán 5 iteraciones y el esquema de entrenamiento que se seguirá es el que mejor ha funcionado en el estudio anterior, es decir el caso 4. Tan solo introduciremos una pequeña variación y es que tomaremos el 30 % del conjunto de test como conjunto de validación. Hacemos esto para poder graficar el momento en el que la red comienza a sobre ajustarse.

El esquema de entrenamiento sería el siguiente:

- El conjunto de entrenamiento = source (60% del total del conjunto de datos),
- El conjunto de reentrenamiento = train (70% del 40% restante del conjunto de datos),
- El conjunto de test = test (21% del 40% restante del conjunto de datos)
- El conjunto de validación = validación (9% del 40% restante del conjunto de datos)

### RESULTADOS

A continuación, mostraremos dos imágenes, ambas representan los valores obtenidos al evaluar el modelo sobre el conjunto de entrenamiento y validación durante la fase de entrenamiento. A la izquierda tenemos la pérdida y en la derecha la precisión.



Estas dos imágenes nos dan una visión global sobre el comportamiento del modelo a lo largo de muchas iteraciones. Como podemos observar el modelo comienza a sobre ajustarse en torno a la época 30, pero tampoco encontramos unos resultados catastróficos.

Por otra parte, a continuación, se muestra una tabla con los resultados de evaluar el modelo contra el conjunto de datos de test.

Iteración	Precisión obtenida al evaluar el modelo
1	68.7372 %
2	75.2114 %
3	65.5585 %
4	69.2038 %
5	85.0685 %
Media	72.7559 %
Desviación estándar	0.4763

La media de la precisión es 72.76 %. El resultado medio obtenido con los parámetros indicados en el estudio son 43.53%. Lo que implica una mejora de un 29.3% o lo que es lo mismo ha aumentado en un 67% con respecto a la configuración propuesta.



## APLICACIÓN DEL CONOCIMIENTO

En esta sección pretendemos medir el rendimiento de aplicar el conocimiento adquirido. Por un lado, usaremos un tamaño de lote más bajo y usaremos un número de épocas mayor.

La hipótesis inicial es:

1. *¿Al aplicar ambas técnicas (reducción de tamaño de lote e incremento de épocas) mejorará el modelo más que al aplicar una sola de ellas?*

Para responder a ambas preguntas vamos a crear un experimento que involucre ambas variables.

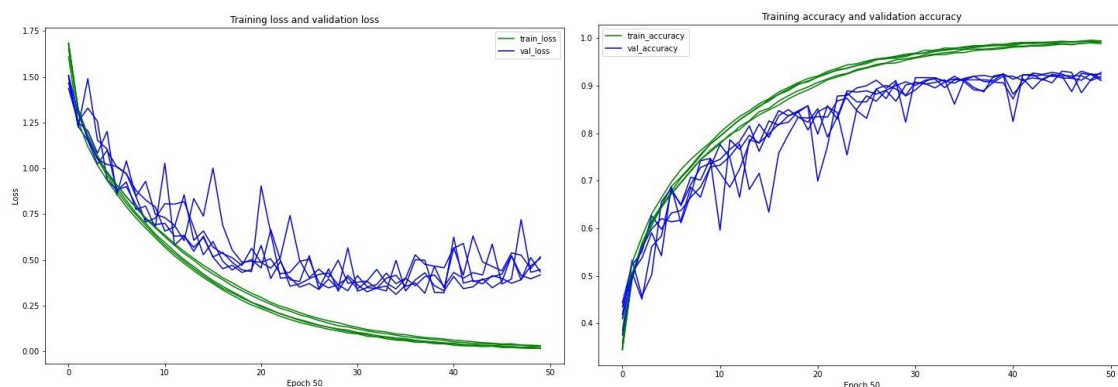
El esquema de entrenamiento será el mismo que en el apartado anterior:

- El conjunto de entrenamiento = source (60% del total del conjunto de datos),
- El conjunto de reentrenamiento = train (70% del 40% restante del conjunto de datos),
- El conjunto de test = test (21% del 40% restante del conjunto de datos)
- El conjunto de validación = validación (9% del 40% restante del conjunto de datos)

Los parámetros que usaremos serán:

- RMSProp con ratio de aprendizaje de 0.0001 (el recomendado por el estudio).
- El tamaño de lote será 16 (el recomendado por el estudio).
- El número de épocas será 25 tanto en el primer entrenamiento del modelo como en la fase de transfer learning.

Los resultados obtenidos durante el entrenamiento se resumen en las siguientes gráficas, ambas representan los valores obtenidos al evaluar el modelo sobre el conjunto de entrenamiento y validación durante la fase de entrenamiento. A la izquierda tenemos la pérdida y en la derecha la precisión.



Si nos fijamos en la imagen de la izquierda (pérdida), podemos observar un serio sobreajuste cuando nos aproximamos a las 30 iteraciones. Por otra parte podemos ver en la grafica de la derecha (precisión), como los valores se estabilizan entorno al 90 % lo cual es una excelente puntuación.

A continuación, trataremos los datos obtenidos al evaluar sobre el set de datos de test:

Iteración	Precisión obtenida al evaluar el modelo
1	93.4675 %
2	92.3010 %
3	92.7676 %
4	92.8259 %
5	91.5719 %
Media	92.5868 %
Desviación estándar	0.6288

Podemos observar que el resultado obtenido es considerablemente mejor que de cada mejora individual. El mejor resultado medio obtenido al replicar el estudio fue de 43.53%. La precisión media de este experimento ha sido del 92.59%. Esto quiere decir que el porcentaje de mejora es del 112,7%, es decir, nuestra configuración de hiperparámetros ha duplicado la precisión del modelo.

## CONCLUSIÓN

A lo largo de este estudio se han validado dos hipótesis:

1. Un tamaño de lote pequeño acelera la convergencia del modelo. Lo ideal es el uso de un valor en torno a 16, y es recomendable acompañarlo de un optimizador con un ratio de aprendizaje bajo (0.0001).
2. El estudio del número de épocas que se va a estar entrenando el modelo es de importancia capital, y a menudo el valor óptimo estará dado por factores como el tamaño del conjunto de imágenes o su complejidad. En nuestro caso 25 épocas (recordemos que son 25 épocas para entrenar el modelo por scratching y 25 más aplicando transfer learning) parece generar cierto sobreajuste. Quizás, un número de épocas óptimo es 15 para cada fase.

Además de la hipótesis validadas, también se ha dado con un modelo con una precisión media del 92.59 % con una variación de 0.62 a lo largo de 5 ejemplos, lo que invita a pensar la robustez de este. El gran contra de este modelo es la tendencia al sobre ajuste, profundizaremos en algunas técnicas para evitar esto en la segunda parte del estudio.

## BIBLIOGRAFÍA

- Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, 6(4), 312–315. <https://doi.org/10.1016/j.ict.2020.04.010>
- Molina, M. Á., Asencio-Cortés, G., Riquelme, J. C., & Martínez-Álvarez, F. (2021). A Preliminary Study on Deep Transfer Learning Applied to Image Classification for Small Datasets. *Advances in Intelligent Systems and Computing*, 1268 AISC, 741–750. [https://doi.org/10.1007/978-3-030-57802-2\\_71](https://doi.org/10.1007/978-3-030-57802-2_71)