

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

PARTE 2: ARQUITECTURA

CONTROL DE VERSIONES

[illegible]

TABLA DE CONTENIDO

Experimento 1	1
Control de versiones.....	2
Introducción	4
Análisis de la precisión tras introducir una capa de Dropout.....	4
Planteamiento del estudio	4
Resultados	5
Análisis de la precisión tras reemplazar ReLU por PReLU	6
Planteamiento del estudio	6
Resultados	6
Análisis de la precisión adoptando ambas técnicas	7
Resultados	8
Conclusión del experimento.....	8
Bibliografía.....	9
Anexo I.....	10
Tabla de resultados red SOCO, ajustada con dropout.....	10
Tabla de resultados red SOCO, ajustada con PReLU	10
Tabla de resultados red SOCO, ajustada con PReLU y dropout.....	11

INTRODUCCIÓN

Este experimento es la segunda parte del experimento 1, el cual se basa en realizar una crítica al artículo (Molina et al., 2021)

En la primera parte de este estudio abordamos la elección de hiperparámetros. En esta segunda parte haremos una crítica constructiva a la elección arquitectónica del modelo.

Nos basaremos en dos artículos fundamentalmente:

- El artículo en el que se presentó la familia de redes VGG, (Molina et al., 2021), puesto que la red SOCO es una versión reducida de esta familia de redes.
- (Zhang et al., 2018) en este artículo se propone validar la siguiente hipótesis: “Un modelo se comporta de forma positiva ante la inclusión de técnicas de dropout y la sustitución de la función de activación ReLU por PReLU”

Las hipótesis que se pretenden validar en este estudio son:

- Modificar la arquitectura añadiendo una capa de dropout resultará en una mejora de la precisión del modelo.
- Reemplazar ReLU por PReLU resultará en una mejora de la precisión del modelo.

ANÁLISIS DE LA PRECISIÓN TRAS INTRODUCIR UNA CAPA DE DROPOUT

PLANTEAMIENTO DEL ESTUDIO

El estudio consistirá en una comparación de la precisión media entre la red SOCO, ajustada según los hiperparámetros convenidos en la parte I, y la red SOCO, ajustada según los hiperparámetros de la parte I y añadiendo una capa de dropout.

Recordemos que el esquema de entrenamiento será 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)

Por otra parte, recordemos que los hiperparámetros elegidos en el apartado anterior son:

- Optimizador: RMSProp con un ratio de aprendizaje de 0.0001.
- Épocas: 25 para entrenamiento y para transferencia.
- Tamaño de lote: 16

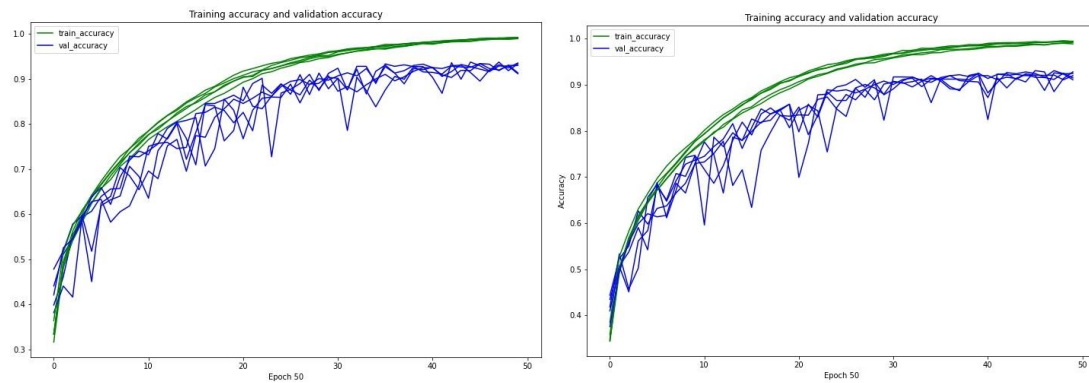
Como nuevo parámetro ajustable incluimos el ratio de desactivación de las neuronas, un valor que introduciremos en la capa de dropout. Se ha escogido usar un valor de 0.4 o lo que es lo mismo, de cada 10 neuronas se apagan 4.

Para asegurar la robustez de la prueba se ejecutará en diez ocasiones.

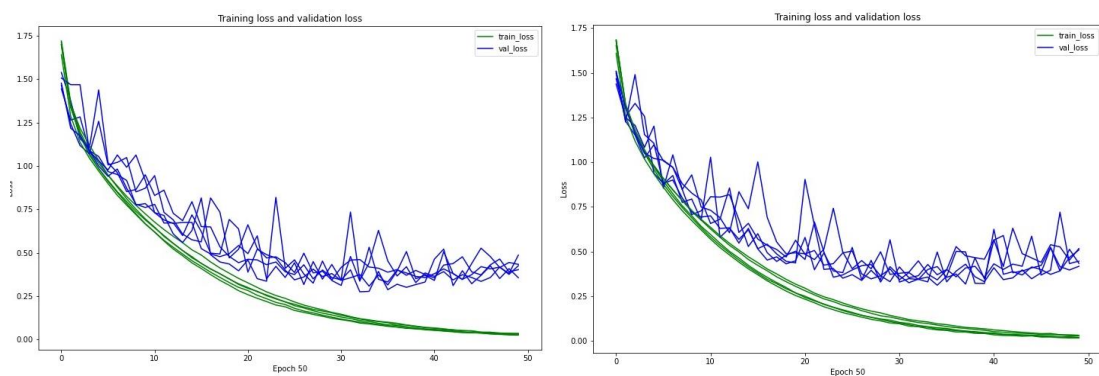
RESULTADOS

En primer lugar, se compara la curva de aprendizaje entre los dos modelos. En la primera imagen tenemos los resultados obtenidos durante el aprendizaje con dropout y en el segundo sin dropout.

A simple vista podemos ver como en la primera imagen, la línea azul (validación) tarda más en despegarse de la verde (entrenamiento), esto significa que el modelo tarda más en sobreajustarse.



Por otra parte, esta misma tendencia se puede observar en las gráficas de pérdida:



De cara a estudiar los resultados sobre el set de datos de test tenemos lo siguiente:

Mientras que tras los ajustes del apartado anterior la red SOCO obtuvo una precisión del $92.59\% \pm 0.63$. La red tras incluir una capa de dropout obtuvo una precisión media de $93.33\% \pm 1.00$ (Tabla de resultados en el Anexo I). Lo que supone de media un incremento de 1%.

ANÁLISIS DE LA PRECISIÓN TRAS REEMPLAZAR RELU POR PRELU

PLANTEAMIENTO DEL ESTUDIO

El estudio consistirá en una comparación de la precisión media entre la red SOCO, ajustada según los hiperparámetros convenidos en la parte I, y la red SOCO, ajustada según los hiperparámetros de la parte I y reemplazando las funciones de activación ReLU por PReLU.

Recordemos que el esquema de entrenamiento será 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)

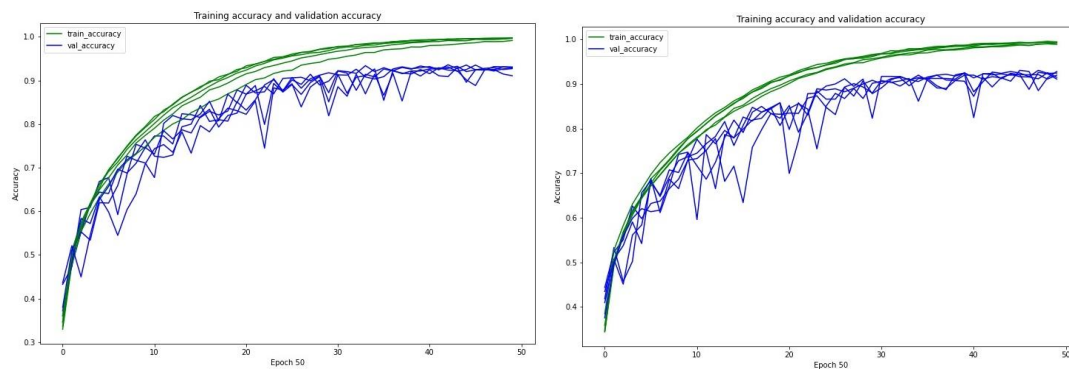
Por otra parte, recordemos que los hiperparámetros elegidos en el apartado anterior son:

- Optimizador: RMSProp con un ratio de aprendizaje de 0.0001.
- Épocas: 25 para entrenamiento y para transferencia.
- Tamaño de lote: 16

Para asegurar la robustez de la prueba se ejecutará en diez ocasiones.

RESULTADOS

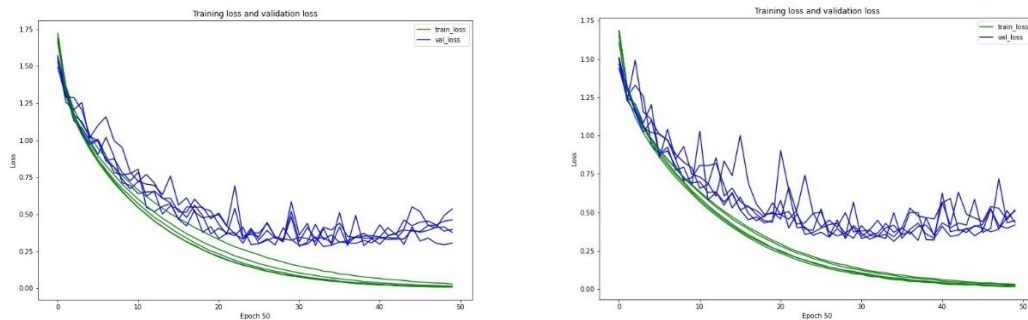
En primer lugar, vamos a comparar la curva de aprendizaje entre los dos modelos. En la imagen de la izquierda tenemos los resultados obtenidos con PReLU y en la derecha con ReLU.



Respecto a estas gráficas hay dos detalles que se pueden apreciar visualmente:

- La precisión de entrenamiento (verde) se encuentra algo separada de la mitad de la ejecución, esto puede ser debido a la parametrización de las funciones de activación.
- Por otra parte, podemos observar un mayor ajuste de la precisión de entrenamiento y la de validación.

Si nos fijamos en las gráficas de pérdida (en la imagen de la izquierda tenemos los resultados obtenidos con PReLU y en la de la derecha con ReLU)



Podemos observar la misma tendencia que en las gráficas de precisión.

Recordemos que la red SOCO obtuvo una precisión del $92.59\% \pm 0.63$. La red tras incluir una capa de dropout y ReLU obtuvo una precisión media de $93.03\% \pm 1.3$ (Tabla de resultados en el Anexo I). Lo que supone de media un incremento de 0.44%.

ANÁLISIS DE LA PRECISIÓN ADOPTANDO AMBAS TÉCNICAS

El estudio consistirá en una comparación de la precisión media entre la red SOCO, ajustada según los hiperparámetros convenidos en la parte I, y la red SOCO, ajustada según los hiperparámetros de la parte I y añadiendo una tanto dropout, como sustituyendo ReLU por PReLU.

Recordemos que el esquema de entrenamiento será 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)

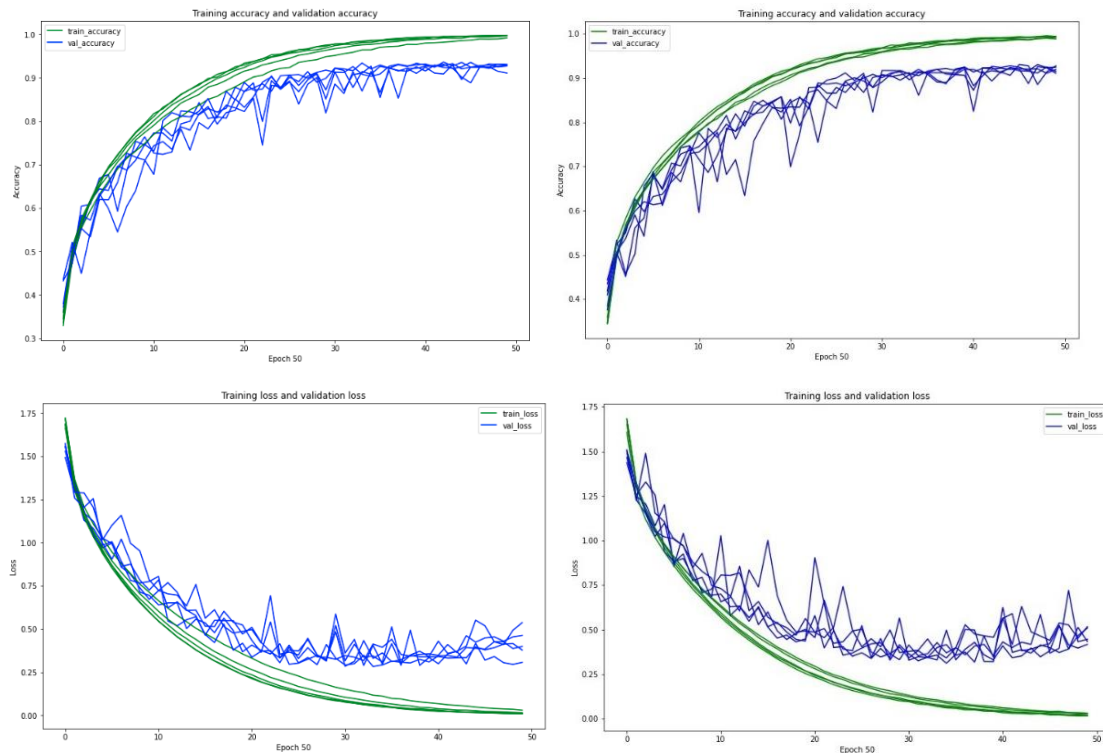
Por otra parte, recordemos que los hiperparámetros elegidos en el apartado anterior son:

- Optimizador: RMSProp con un ratio de aprendizaje de 0.0001.
- Épocas: 25 para entrenamiento y para transferencia.
- Tamaño de lote: 16
- Ratio de desactivación de neuronas en la capa de dropout es de 0.4

Para asegurar la robustez de la prueba se ejecutará en diez ocasiones.

RESULTADOS

En primer lugar, vamos a comparar la curva de aprendizaje entre los dos modelos. En la imagen de la izquierda tenemos los resultados obtenidos con PReLU y dropout y en la derecha la red SOCO.



Recordemos que la red SOCO obtuvo una precisión del $92.59\% \pm 0.63$. La red tras incluir una capa de dropout y ReLU obtuvo una precisión media de $92.75\% \pm 0.7$ (Tabla de resultados en el Anexo I). Lo que supone de media un incremento de 0.17%.

CONCLUSIÓN DEL EXPERIMENTO

Se esperaba conseguir unos resultados semejantes a los obtenidos en (Zhang et al., 2018), es decir, una mejora aproximada de 1.5 % en la precisión al usar dropout y 0.8% al usar PReLU. No obstante, nuestros resultados han sido bastante más modestos.

Al usar dropout la precisión de la red ha mejorado en un 1.0%, y al usar PReLU mejoró un 0.44%. Pero lo más extraño es que al utilizar ambas técnicas la mejora se vuelve casi imperceptible con un incremento de tan solo el 0.17%

BIBLIOGRAFÍA

- 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
- Zhang, Y. D., Pan, C., Sun, J., & Tang, C. (2018). Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU. *Journal of Computational Science*, 28, 1–10. <https://doi.org/10.1016/j.jocs.2018.07.003>

ANEXO I

TABLA DE RESULTADOS RED SOCO, AJUSTADA CON DROPOUT

Iteración	Precisión obtenida al evaluar el modelo
1	94.0216 %
2	93.8758 %
3	93.4675 %
4	91.3677 %
5	93.0884 %
6	93.8758 %
7	94.2840 %
8	93.7008 %
9	94.2549 %
10	91.7469 %
Media	93.3368 %
Desviación estándar	1.00218

TABLA DE RESULTADOS RED SOCO, AJUSTADA CON PRELU

Iteración	Precisión obtenida al evaluar el modelo
1	93.4092 %
2	93.5550 %
3	93.0009 %
4	93.9924 %
5	89.5013 %
6	93.3508 %
7	93.0300 %
8	93.2342 %
9	92.8259 %
10	94.4299 %
Media	93.033 %
Desviación estándar	1.331

TABLA DE RESULTADOS RED SOCO, AJUSTADA CON PRELU Y DROPOUT

Iteración	Precisión obtenida al evaluar el modelo
1	90.9303 %
2	93.2633 %
3	92.8259 %
4	92.8842 %
5	92.4759 %
6	93.3508 %
7	92.8551 %
8	93.3217 %
9	92.6509 %
10	92.9134 %
Media	92.7472 %
Desviación estándar	0.6999