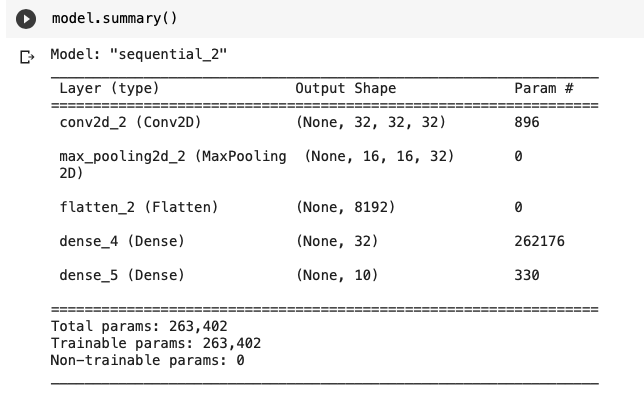
**Comentarios sobre CIFAR10**

Desde la más absoluta sinceridad tengo que decirte que no sabía qué estaba haciendo en los primeros modelos que he lanzado, he visto de nuevo tus clases, buscado en Google y preguntado a compañeros. Y aquí va lo mejor que he podido hacer. Espero que lo disfrutes.

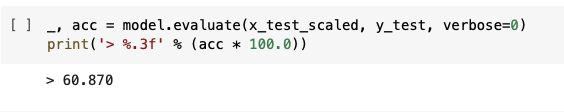
**V1.**

En esta primera versión simplemente ejecuto la plantilla que me viene dada. Mismos epochs, mismo batch size…

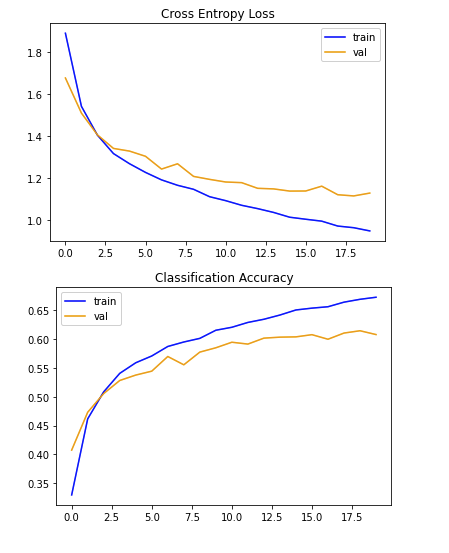
Quedando así los parámetros:



Con una accuracy bajita de 60.87.



Al tener una accuracy tan baja, las líneas de train/val salen bastante distantes:

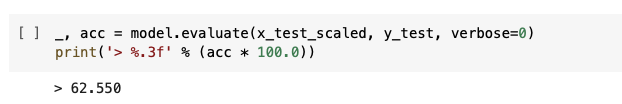


Está overfitteando ya que el train está por encima de validación, lo que significa que hemos entrenado el modelo para que acierte con unos valores muy concretos.

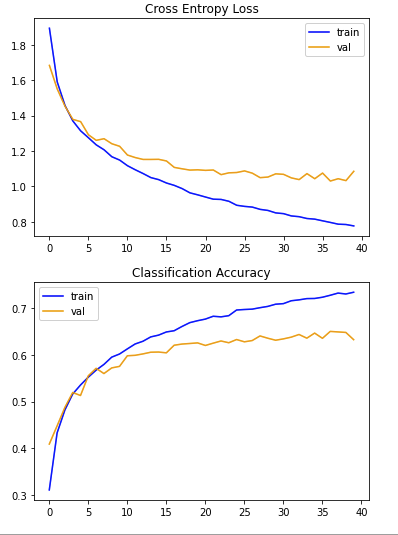
**V2**

Los únicos cambios que hago en la segunda versión es aumentar los epochs de 20 a 40, es decir el número de veces que se ejecutarán los algoritmos. Sé que no es ningún cambio que vaya a repercutir en el modelo pero como te he comentado al principio, he ido bastante a ciegas en los primeros modelos.

Los parámetros son los mismos que en la V1.

La accuracy ha subido un poquito:

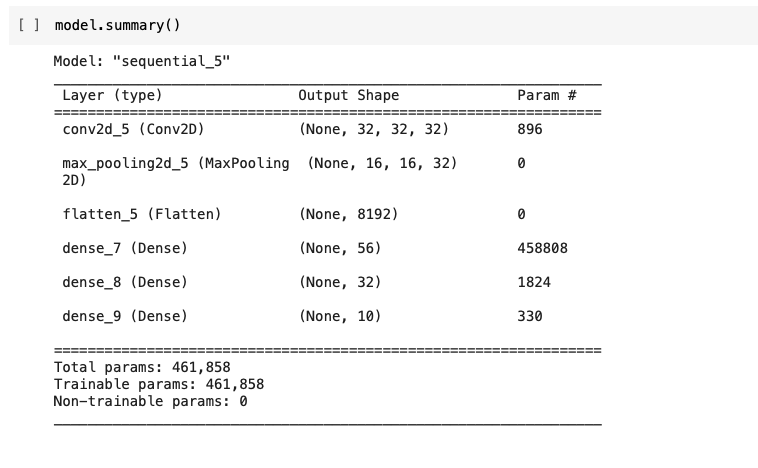
También podemos ver que las líneas de validación y training se han distanciado aún más:

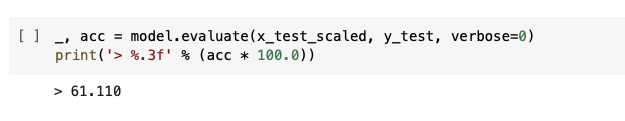


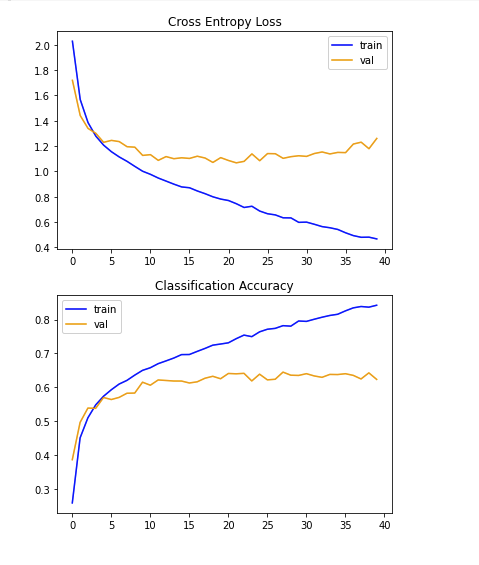
Está overfitteando ya que el train está por encima de validación, lo que significa que hemos entrenado el modelo para que acierte con unos valores muy concretos.

**V3**

En esta versión añadiré una neurona lineal que es la más simple de todas y la base para encontrar nuevas funciones de paso que nos permitan aprender. La introduzco entre la Flatten y Relu. Sigo dejando 40 epochs para que vaya aprendiendo el modelo.





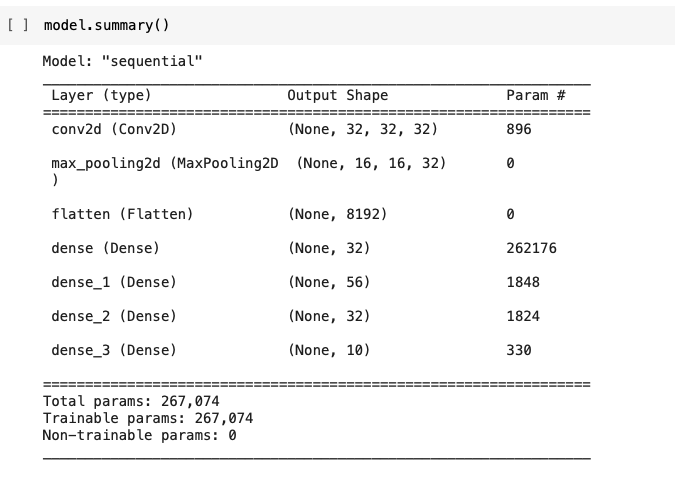


El rate de loss ha aumentado y la accuracy ha disminuido lo que hace que se separen aún más.

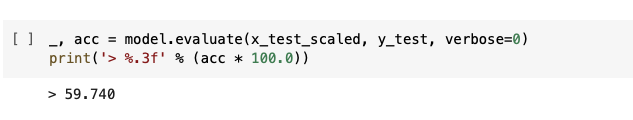
Está overfitteando ya que el train está por encima de validación, lo que significa que hemos entrenado el modelo para que acierte con unos valores muy concretos.

**V4**

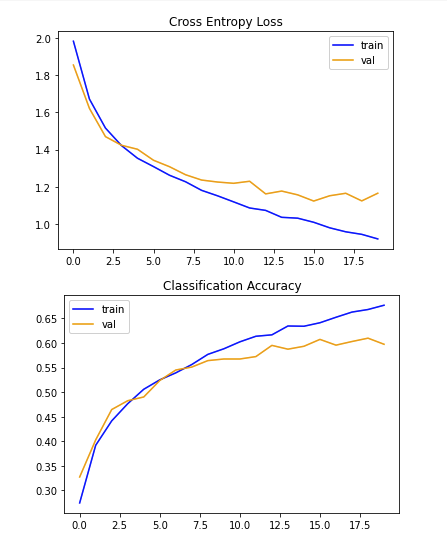
En este modelo sigo añadiendo capas , relu y linear, he cambiado el peso de las mismas a 32 y 56 respectivamente. Hago un modelo más sencillo para ver si soluciono el overfitting.



La accuracy me sale aún más baja:

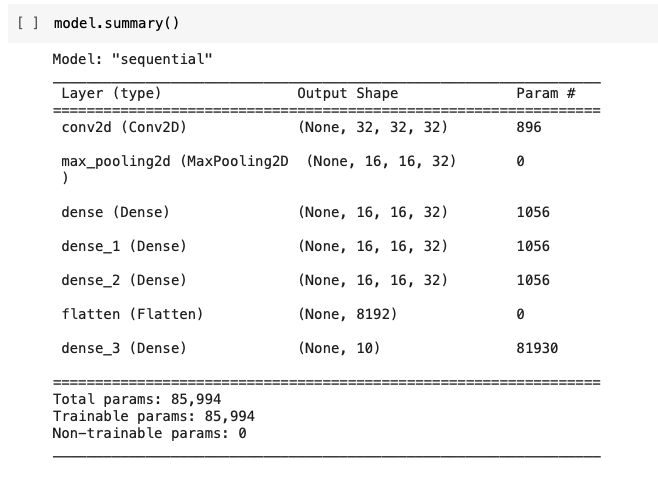


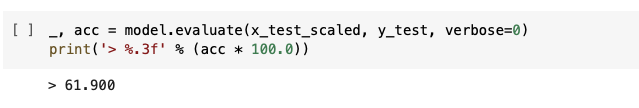
Pero me está overfitteando menos que es lo que buscaba al añadir estas capas:

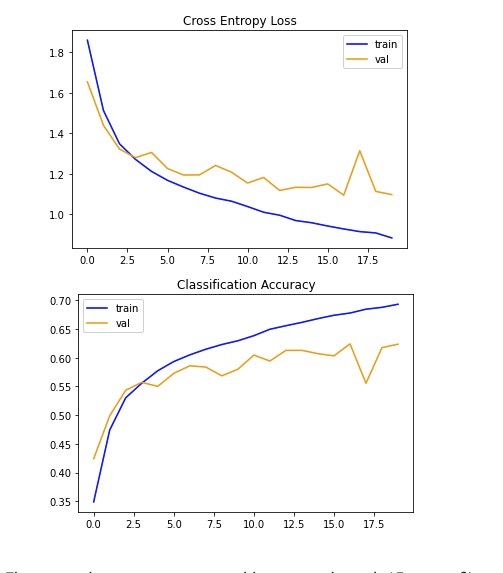


**V5**

En esta versión seguiré añadiendo capas Flatten en este caso para que me disminuya la imagen.



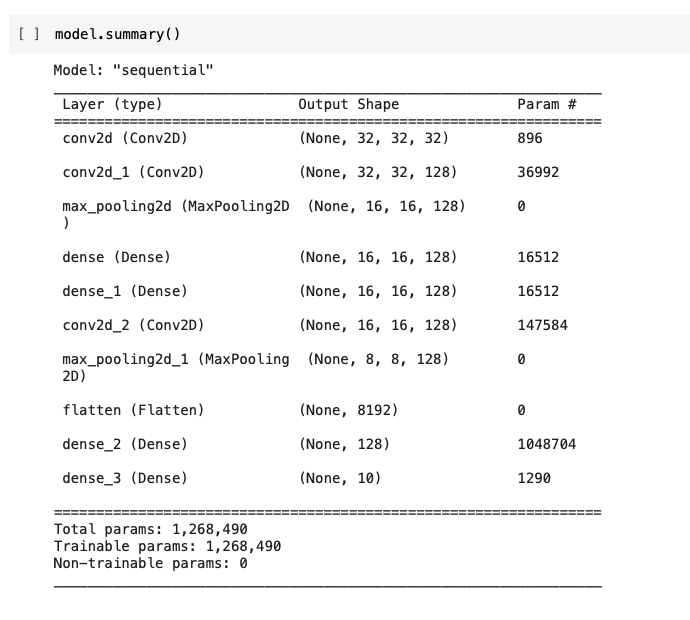




Esta versión es bastante irrelevante, no ha mejorado nada el modelo, lo que quiere decir que lo que estoy haciendo no está funcionando como me gustaría.

**V6**

En esta versión sigo puramente con arquitectura, añadiendo capas, todavía no meto dropouts para mejorar el overfitting.



La accuracy ha mejorado un poquito.

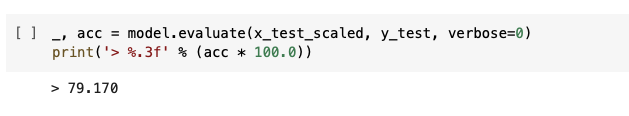


**V7**

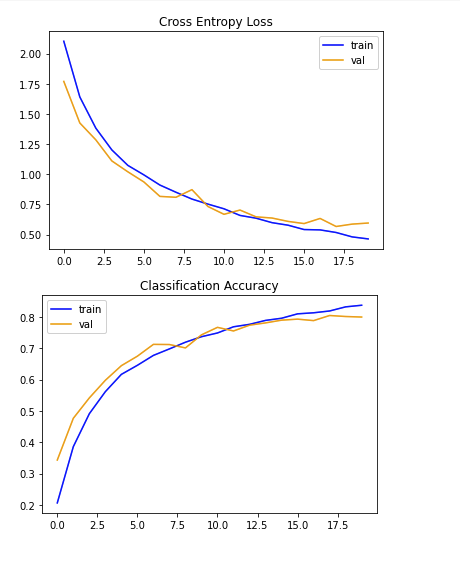
En esta versión ya añado dropouts para mejorar el overfitting. Y continúo añadiendo capas entre ellas convolución y subiendo el peso.



Podemos ver como la accuracy mejora notablemente.

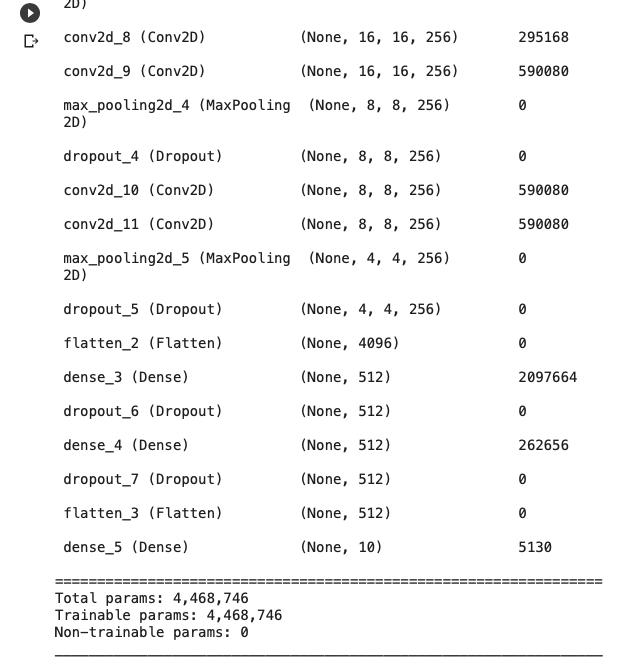


También podemos ver como el modelo se va ajustando más a la realidad y el overfit se reduce, todo va sobre ruedas.

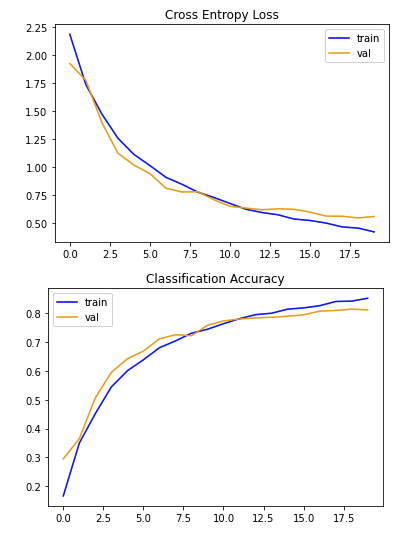


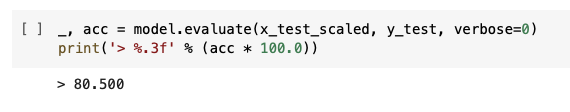
**V8**

En esta versión en la que ya estamos con técnicas avanzadas, dejo las mismas capas pero aumento los pesos.



También he añadido callbacks porque no los añadí antes y el modelo ya me está tardando demasiado en ejecutar.



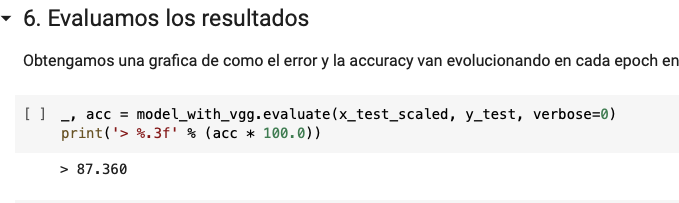


Todo está mejorando.

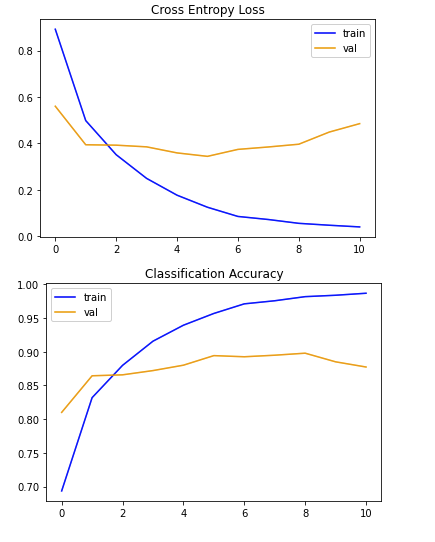
**V9**

Aquí añado el VGG16, redes previamente preentenadas con bases de datos grandes que adaptaré a nuestro problema. Tengo que hacer resize de las imágenes para poder utilizar el VGG.

Es el modelo con mejor accuracy de momento.



Pero me vuelve a hacer overfitting.



**V10**

He ejecutado con el mismo modelo de la versión 9 añadiendo a mayores el kernel\_initialiazer que como dijiste en las clases ayuda al reconocimiento de imágenes y va de perlas. Vuelve a overfittear pero he conseguido la mejor accuracy.

