



PROYECTO

REDES NEURONALES CIFAR 10

Autor: Carlos Cabañó

Máster en Data Science
Nuclo Digital School

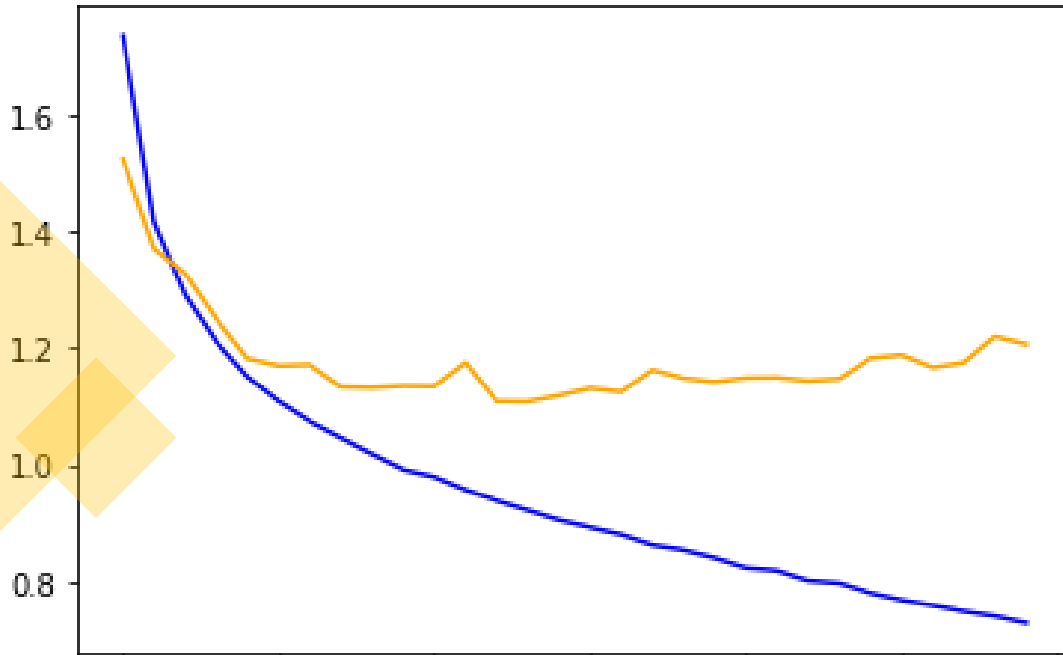
MUESTRA 1

Cambios introducidos y comentarios:

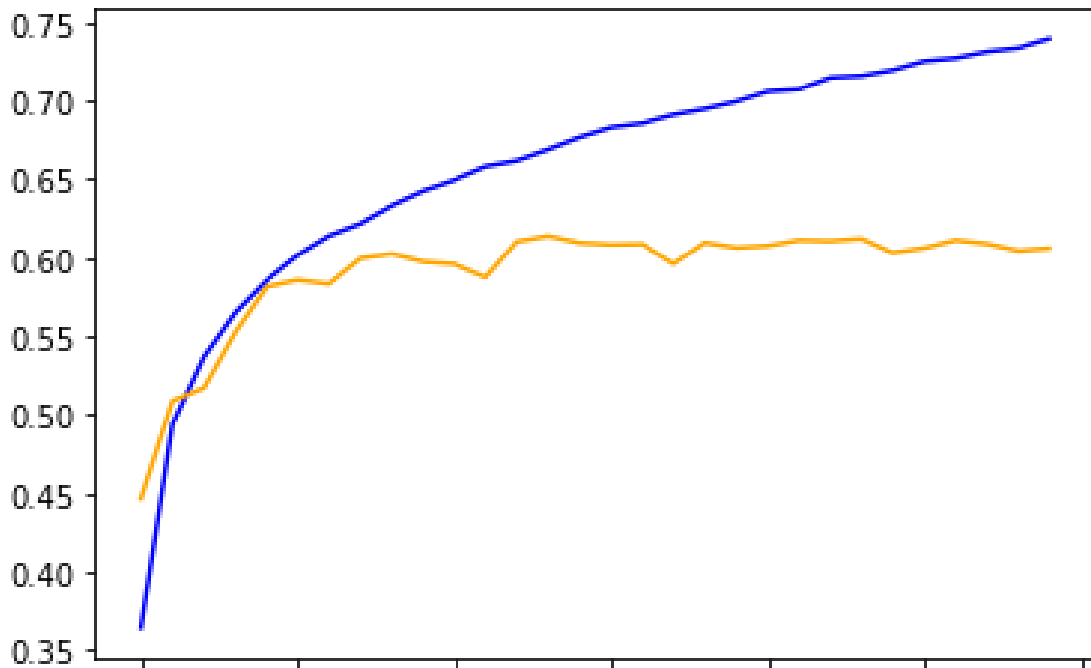
- Aumento de Epochs de 5 a 30, ya que aún no vemos overfitting en los tramos finales de la versión 0.
- Vemos cómo aumenta el Accuracy de 55 (versión 0) a 60,48, aunque el accuracy óptimo se obtiene a los 12 epoch. En el modelo se genera mucho *overfitting* si observamos la diferencia entre el modelo entrenado y la validación.

Resultado en accuracy: 60,48

Cross Entropy Loss



Classification Accuracy



MUESTRA 1 - Arquitectura

Model: "sequential"

Layer (type)	Output Shape	Param #
--------------	--------------	---------

conv2d (Conv2D)	(None, 32, 32, 32)	896
-----------------	--------------------	-----

max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
------------------------------	--------------------	---

flatten (Flatten)	(None, 8192)	0
-------------------	--------------	---

dense (Dense)	(None, 32)	262176
---------------	------------	--------

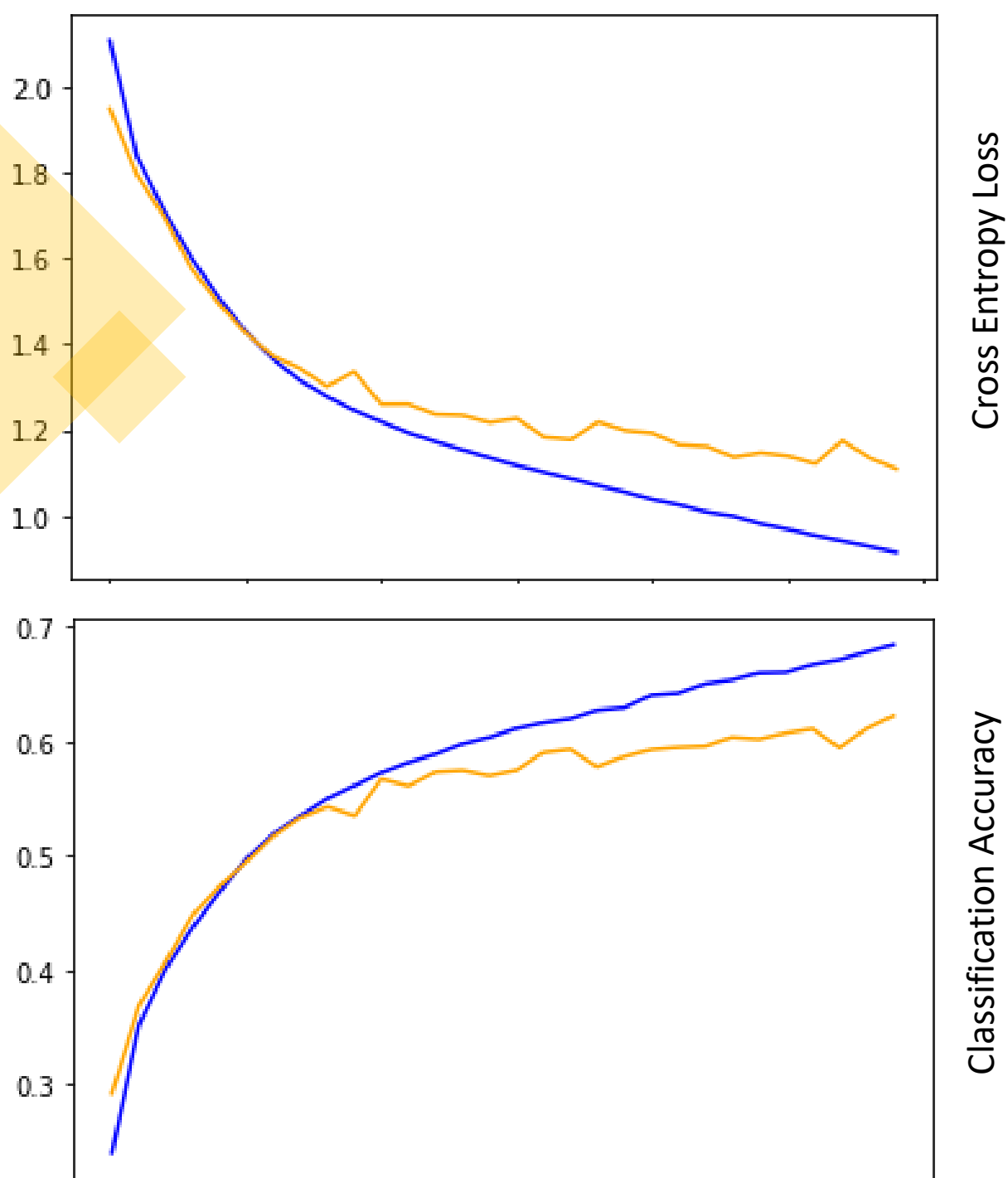
dense_1 (Dense)	(None, 10)	330
-----------------	------------	-----

Total params: 263,402 Trainable params: 263,402 Non-trainable params: 0		
---	--	--

MUESTRA 2

Cambios introducidos y comentarios:

- Cambiamos el optimizador de ADAM (preestablecido) a SGD. Con SGD establecemos tanto learning rate como momentum.
- El Accuracy mejora ligeramente pero la principal diferencia con la versión anterior está en la mejora del *overfitting*. Aunque sigue generándose, este es menor que con el optimizador ADAM.
- **Resultado en accuracy: 61,79**



MUESTRA 2 - Arquitectura

Model: "sequential"

Layer (type)	Output Shape	Param #
--------------	--------------	---------

conv2d (Conv2D)	(None, 32, 32, 32)	896
-----------------	--------------------	-----

max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
------------------------------	--------------------	---

flatten (Flatten)	(None, 8192)	0
-------------------	--------------	---

dense (Dense)	(None, 32)	262176
---------------	------------	--------

dense_1 (Dense)	(None, 10)	330
-----------------	------------	-----

=====
Total params: 263,402 Trainable params: 263,402 Non-trainable
params: 0

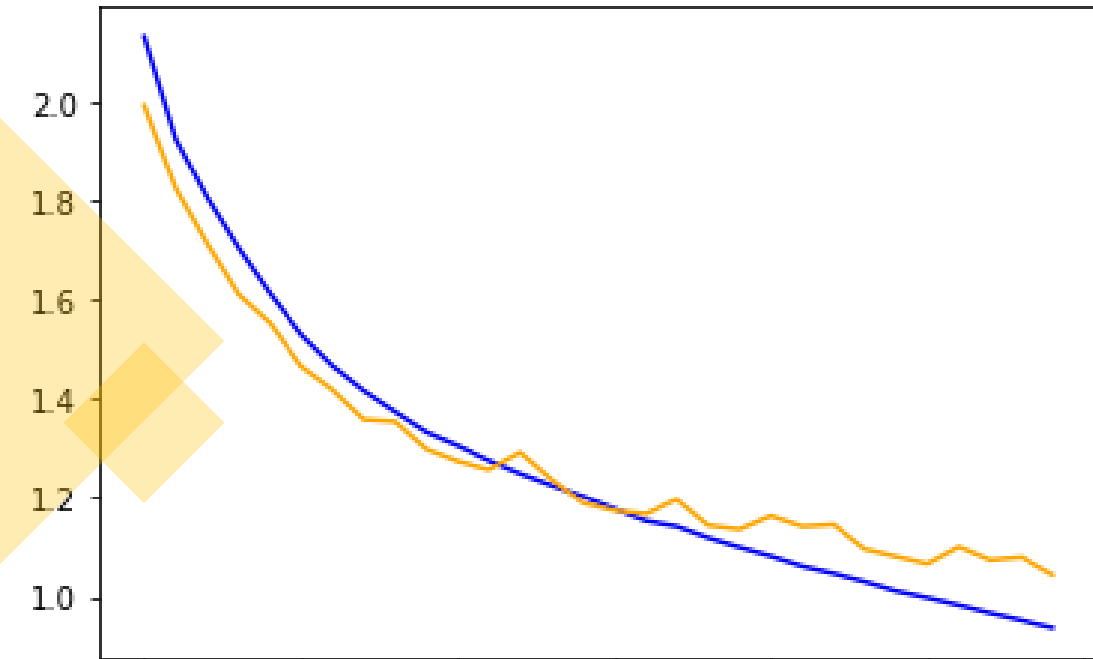
MUESTRA 3

Cambios introducidos y comentarios:

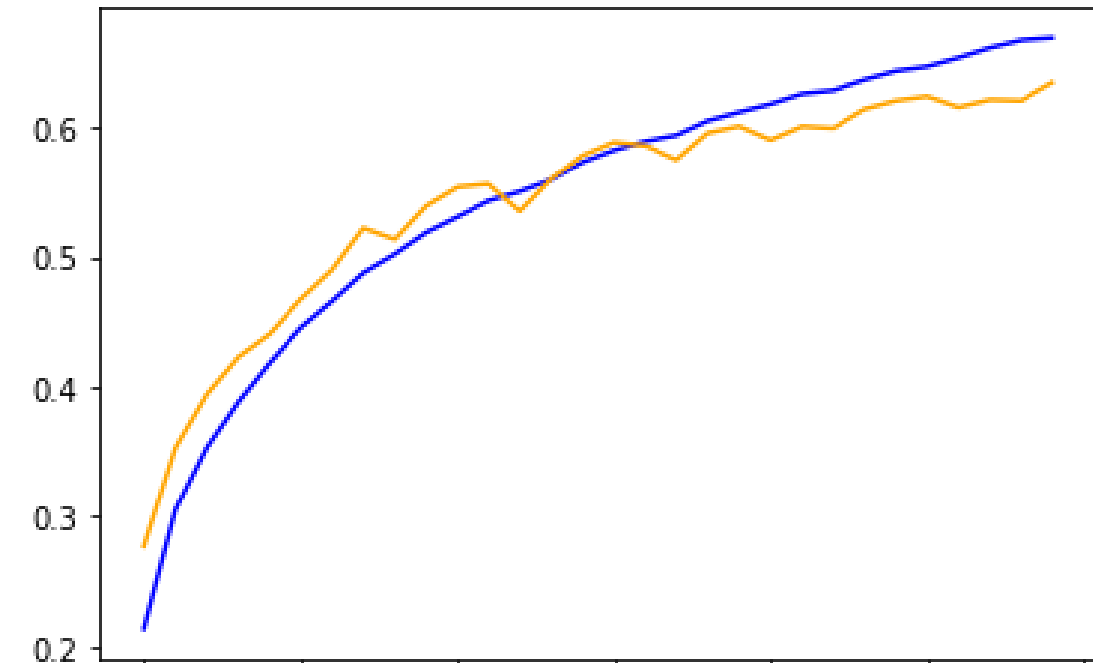
- Añadimos más capas y más profundidad (una de 64), para darle una mayor complejidad a la arquitectura.
- Añadimos un drop-out (20%) y dejamos el batch a 64, ya que aumentarlo lo único que se consigue es empeorar los resultados de accuracy del test.
- Con el drop-out conseguimos que ciertas neuronas seleccionadas aleatoriamente sean ignoradas durante el entrenamiento, lo que previene la aparición de overfitting.

Resultado en accuracy: 63,81

Cross Entropy Loss



Classification Accuracy



MUESTRA 3 - Arquitectura

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	896
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 32)	262176
dense_1 (Dense)	(None, 64)	2112
dropout (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 10)	650

=====
Total params: 265,834 Trainable params: 265,834 Non-trainable
params: 0
=====

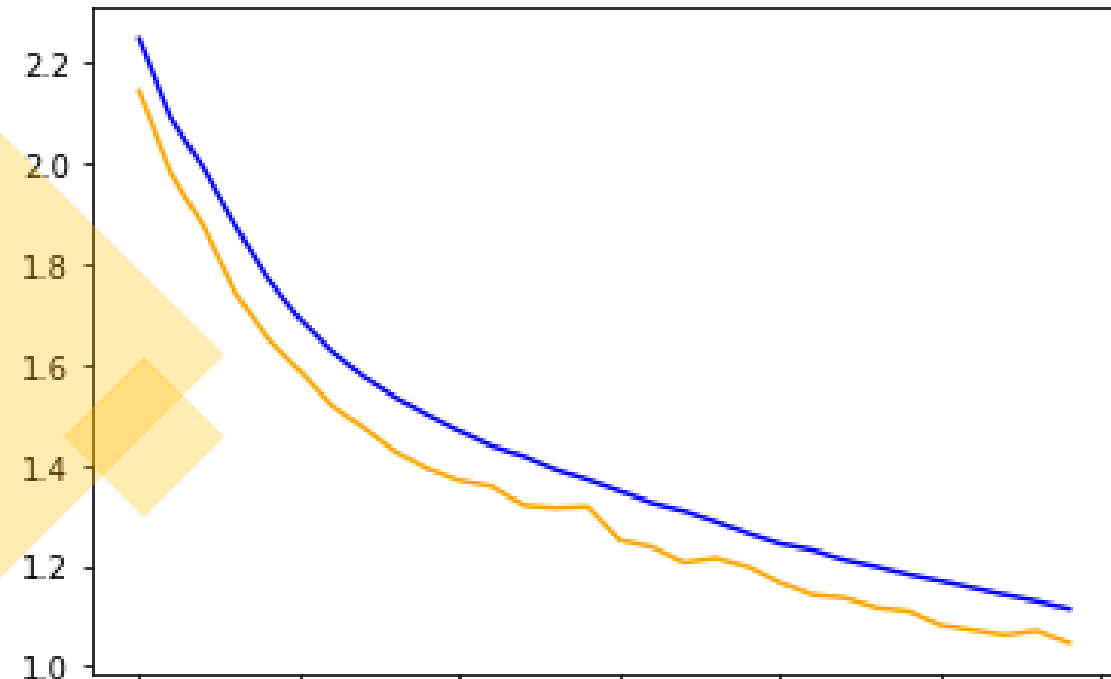
MUESTRA 4

Cambios introducidos y comentarios:

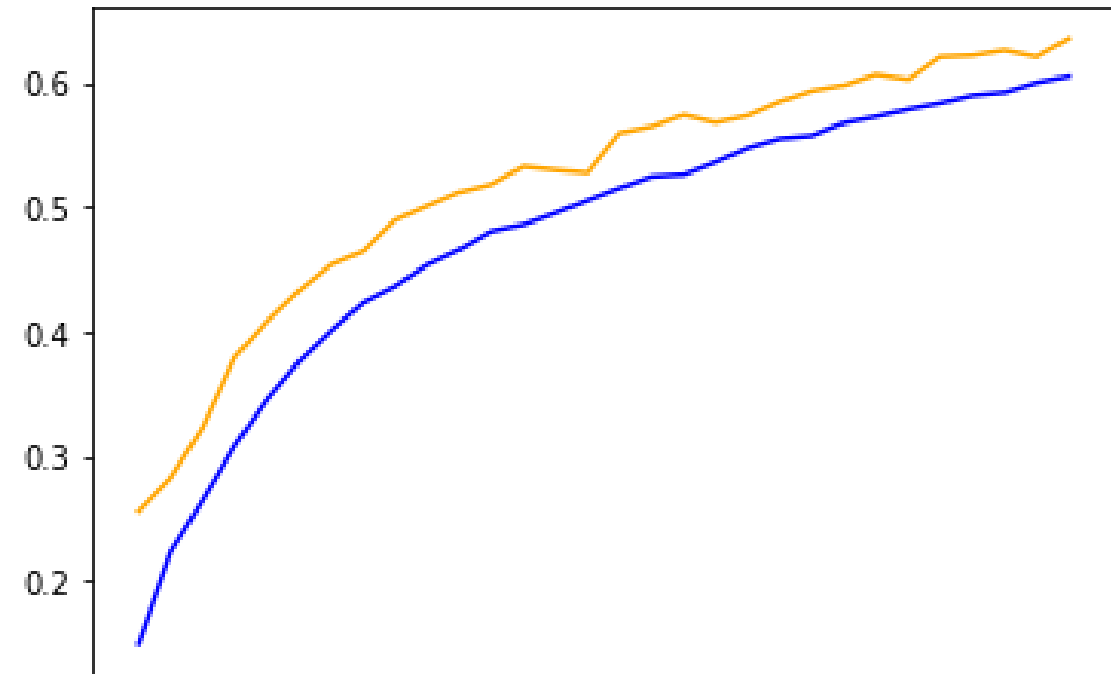
- Añadimos una capa de convolución 2D y un MaxPooling más para añadirle complejidad a la arquitectura, que será lo que mejore el modelo.
- Añadimos un Drop-out más después de la última capa de convolución.
- Aumentamos los Drop-out anteriores a 0.3, de forma que le sea más difícil al modelo aprender.
- El resultado empeora ligeramente respecto de la anterior muestra porque el modelo puede dar más de sí con mas Epochs.

Resultado en accuracy: 62,95

Cross Entropy Loss



Classification Accuracy



MUESTRA 4 - Arquitectura

Model: "sequential"

Layer	(type)	Output Shape	Param #
-------	--------	--------------	---------

conv2d	(Conv2D)	(None, 32, 32, 32)	896
--------	----------	--------------------	-----

max_pooling2d	(MaxPooling2D)	(None, 16, 16, 32)	0
---------------	----------------	--------------------	---

conv2d_1	(Conv2D)	(None, 16, 16, 32)	9248
----------	----------	--------------------	------

max_pooling2d_1	(MaxPooling2D)	(None, 8, 8, 32)	0
-----------------	----------------	------------------	---

dropout	(Dropout)	(None, 8, 8, 32)	0
---------	-----------	------------------	---

flatten	(Flatten)	(None, 2048)	0
---------	-----------	--------------	---

dense	(Dense)	(None, 32)	65568
-------	---------	------------	-------

dense_1	(Dense)	(None, 64)	2112
---------	---------	------------	------

dropout_1	(Dropout)	(None, 64)	0
-----------	-----------	------------	---

dense_2	(Dense)	(None, 10)	650
---------	---------	------------	-----

=====
Total params: 78,474 Trainable params: 78,474 Non-trainable
params: 0
=====

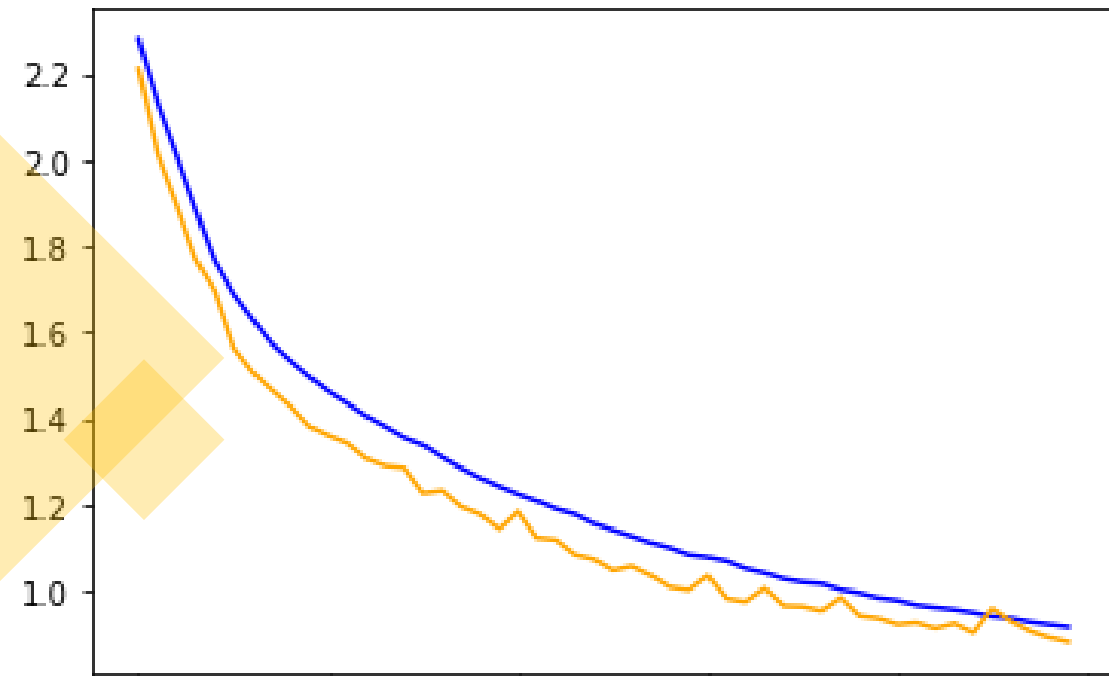
MUESTRA 5

Cambios introducidos y comentarios:

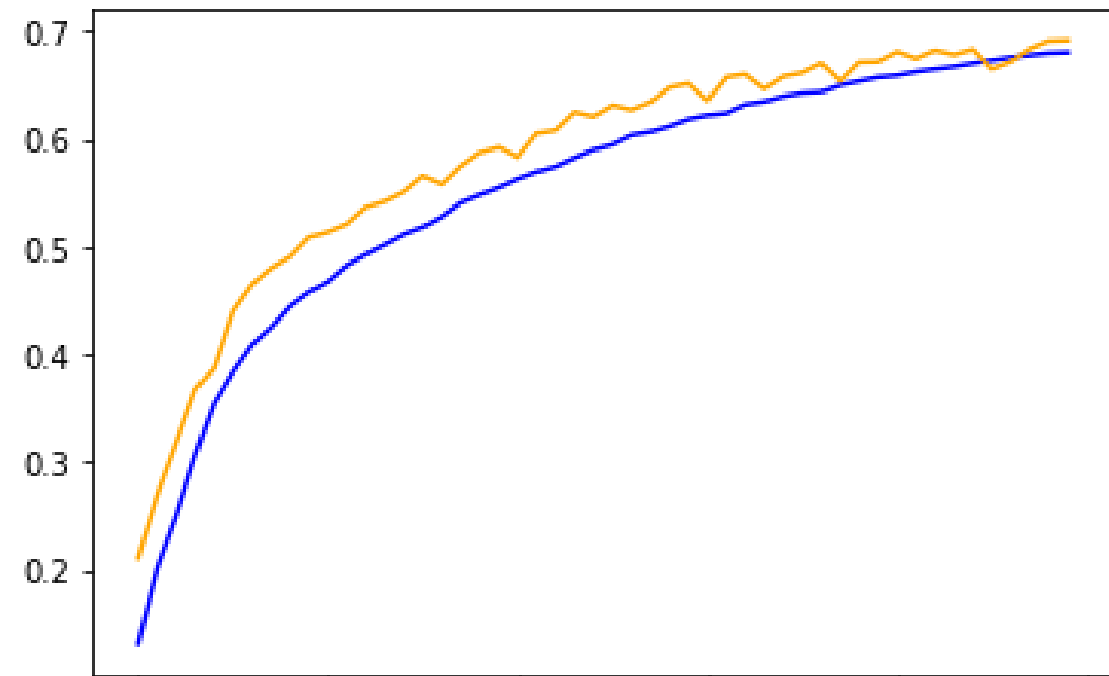
- Aumentamos los Epochs a 50, ya que en la anterior muestra el modelo parecía poder dar más de sí. Efectivamente, se produce convergencia.
- Añadimos callbacks, concretamente un Early Stopping, mediante el cual lograremos que el modelo pare cuando se cumplan los parámetros que le indiquemos, en este caso, un patience de 5.

Resultado en accuracy: 69,06

Cross Entropy Loss



Classification Accuracy



MUESTRA 5 - Arquitectura

Layer	(type)	Output Shape	Param #
conv2d	(Conv2D)	(None, 32, 32, 32)	896
max_pooling2d	(MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_1	(Conv2D)	(None, 16, 16, 32)	9248
max_pooling2d_1	(MaxPooling2D)	(None, 8, 8, 32)	0
dropout	(Dropout)	(None, 8, 8, 32)	0
flatten	(Flatten)	(None, 2048)	0
dense	(Dense)	(None, 32)	65568
dense_1	(Dense)	(None, 64)	2112
dropout_1	(Dropout)	(None, 64)	0
dense_2	(Dense)	(None, 10)	650
Total params: 78,474 Trainable params: 78,474 Non-trainable params: 0			

MUESTRA 6

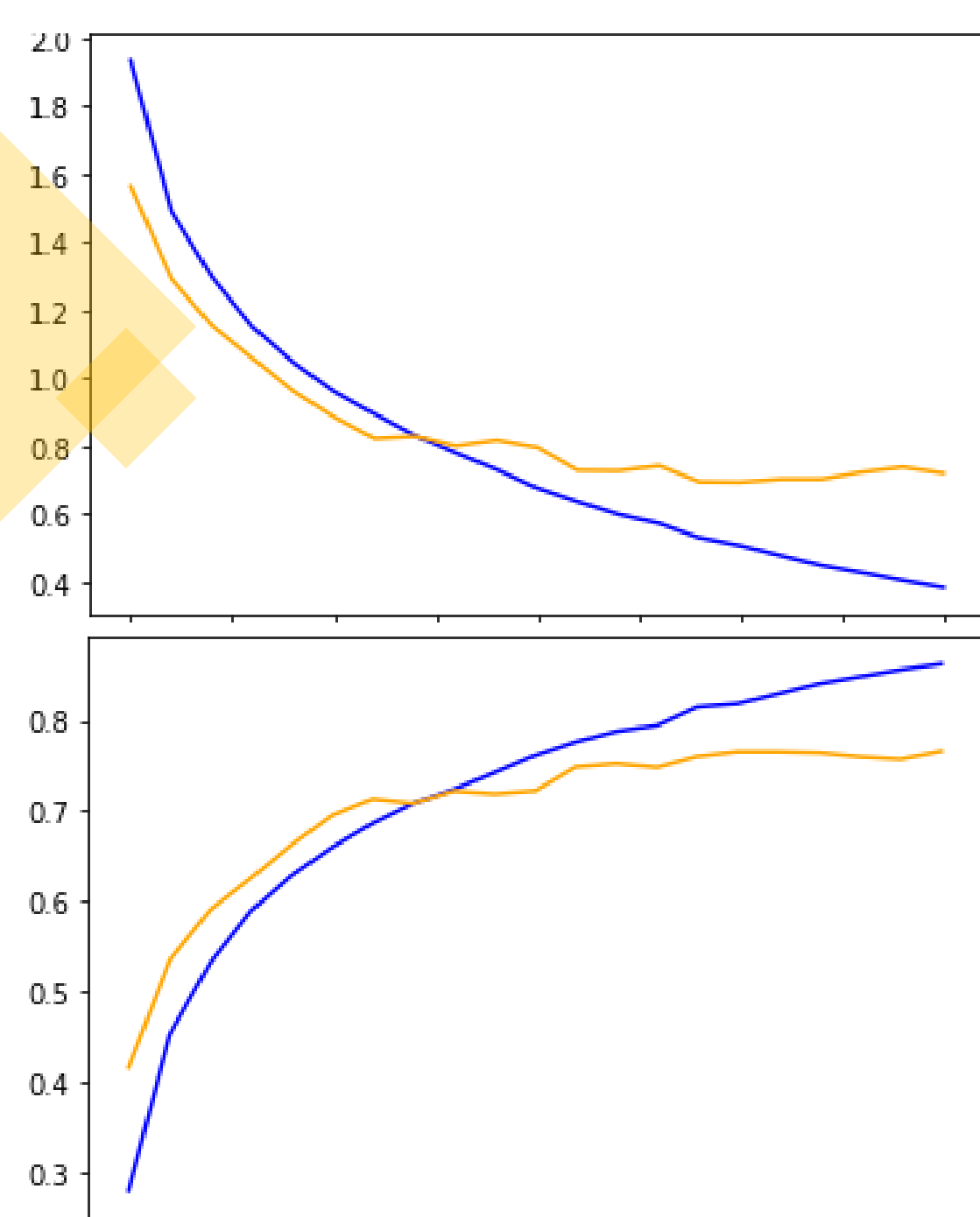
Cambios introducidos y comentarios:

- Mejoramos el código de la convolución añadiendo más capas de 2D y maxpooling. Las capas tienen mayor profundidad: 32, 64 y 128.
- Aumentamos el nº de epochs a 100 pues el modelo anterior aún podía dar más de sí y las convoluciones aprenden más rápido. Dejamos el Callback (early stopping) anterior.
- Añadimos un learning rate de 0.1 (es bastante alto) y un momentum de 0.9 al optimizador SGD.
- El resultado mejora ostensiblemente respecto de la anterior muestra, por lo que nos enfocaremos en mejorar la estructura y añadirle complejidad.

Resultado en accuracy: 75,76

Cross Entropy Loss

Classification Accuracy



MUESTRA 6 - Arquitectura

Model: "sequential"

Layer	(type)	Output Shape	Param #
=====			
conv2d	(Conv2D)	(None, 32, 32, 32)	896

max_pooling2d	(MaxPooling2D)	(None, 16, 16, 32)	0

conv2d_1	(Conv2D)	(None, 16, 16, 64)	18496

max_pooling2d_1	(MaxPooling2D)	(None, 8, 8, 64)	0

conv2d_2	(Conv2D)	(None, 8, 8, 128)	73856

max_pooling2d_2	(MaxPooling2D)	(None, 4, 4, 128)	0

dropout	(Dropout)	(None, 4, 4, 128)	0

flatten	(Flatten)	(None, 2048)	0

dense	(Dense)	(None, 512)	1049088

dropout_1	(Dropout)	(None, 512)	0

dense_1	(Dense)	(None, 512)	262656

dropout_2	(Dropout)	(None, 512)	0

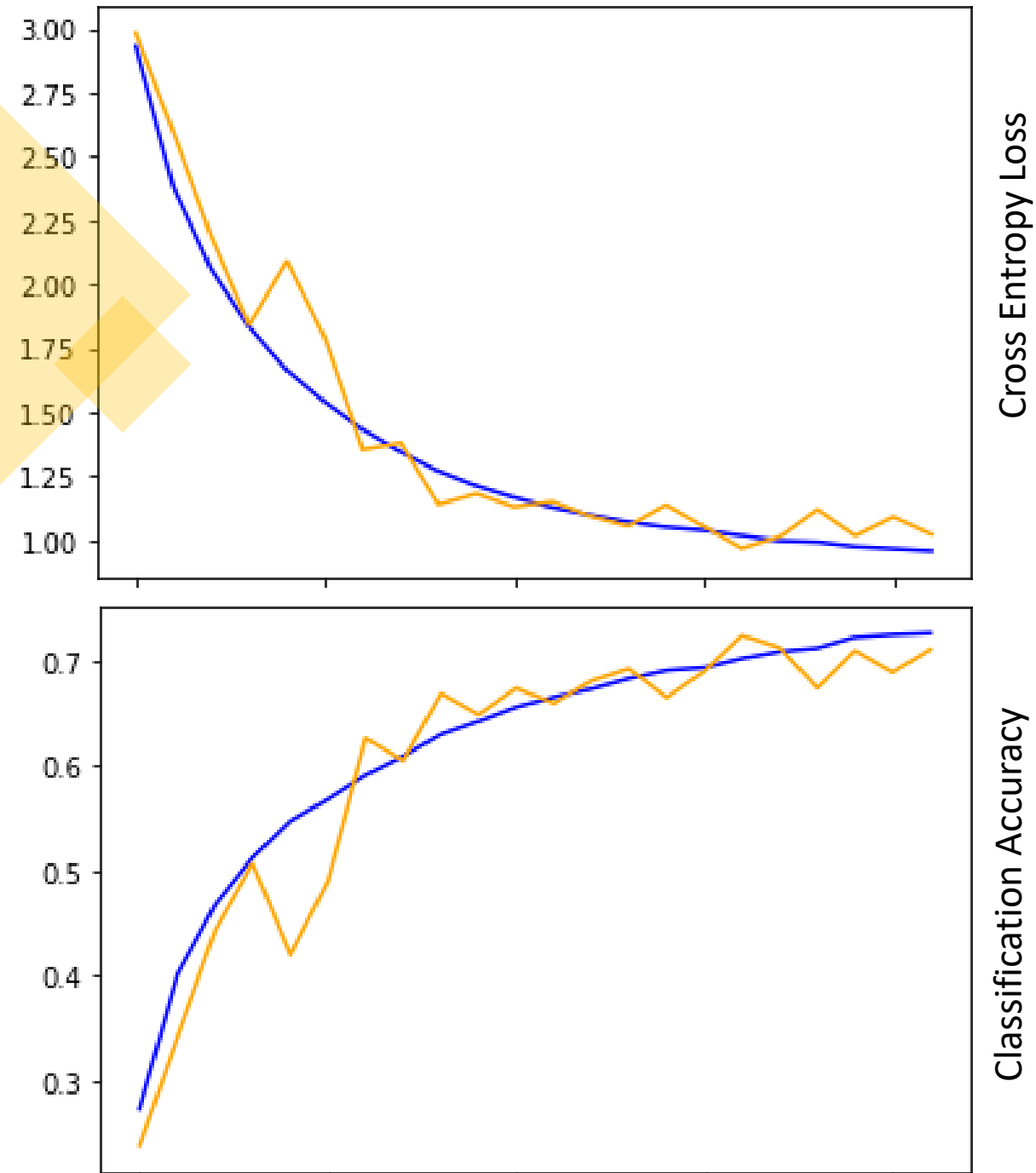
dense_2	(Dense)	(None, 10)	5130
=====			
Total params: 1,410,122 Trainable params: 1,410,122 Non-trainable params: 0			

MUESTRA 7

Cambios introducidos y comentarios:

- Añadimos capas de Batch normalization para ayudar a estandarizar las activaciones
- Añadimos Weight Decay para implementar penalizaciones al modelo y prevenir el overfitting. Lo hacemos con el Kernel regularization de Keras a 0,001.
- Añadimos Kernel initializer (“he_uniform”) para inicializar los pesos.
- Añadimos drop-out después de todas las convoluciones, lo que le dificultará más aún el aprendizaje. El último drop-out justo antes de la última capa lo establecemos en 0.1.
- El resultado empeora respecto del anterior modelo, posiblemente por el exceso de penalizaciones para overfitting a pesar de que nuestra red aún no es muy compleja. Vamos a aplicar una mayor complejidad a la arquitectura.

Resultado en accuracy: 70,83



MUESTRA 7 - Arquitectura

Model: "sequential"

Layer	(type)	Output Shape	Param #
-------	--------	--------------	---------

conv2d	(Conv2D)	(None, 32, 32, 32)	896
--------	----------	--------------------	-----

max_pooling2d	(MaxPooling2D)	(None, 16, 16, 32)	0
---------------	----------------	--------------------	---

dropout	(Dropout)	(None, 16, 16, 32)	0
---------	-----------	--------------------	---

conv2d_1	(Conv2D)	(None, 16, 16, 64)	18496
----------	----------	--------------------	-------

max_pooling2d_1	(MaxPooling2D)	(None, 8, 8, 64)	0
-----------------	----------------	------------------	---

dropout_1	(Dropout)	(None, 8, 8, 64)	0
-----------	-----------	------------------	---

conv2d_2	(Conv2D)	(None, 8, 8, 128)	73856
----------	----------	-------------------	-------

max_pooling2d_2	(MaxPooling2D)	(None, 4, 4, 128)	0
-----------------	----------------	-------------------	---

dropout_2	(Dropout)	(None, 4, 4, 128)	0
-----------	-----------	-------------------	---

flatten	(Flatten)	(None, 2048)	0
---------	-----------	--------------	---

dense	(Dense)	(None, 512)	1049088
-------	---------	-------------	---------

batch_normalization	(Batch Normalization)	(None, 512)	2048
---------------------	-----------------------	-------------	------

dense_1	(Dense)	(None, 512)	262656
---------	---------	-------------	--------

dropout_3	(Dropout)	(None, 512)	0
-----------	-----------	-------------	---

batch_normalization_1	(Batch Normalization)	(None, 512)	2048
-----------------------	-----------------------	-------------	------

dense_2	(Dense)	(None, 512)	262656
---------	---------	-------------	--------

dropout_4	(Dropout)	(None, 512)	0
-----------	-----------	-------------	---

dense_3	(Dense)	(None, 10)	5130
---------	---------	------------	------

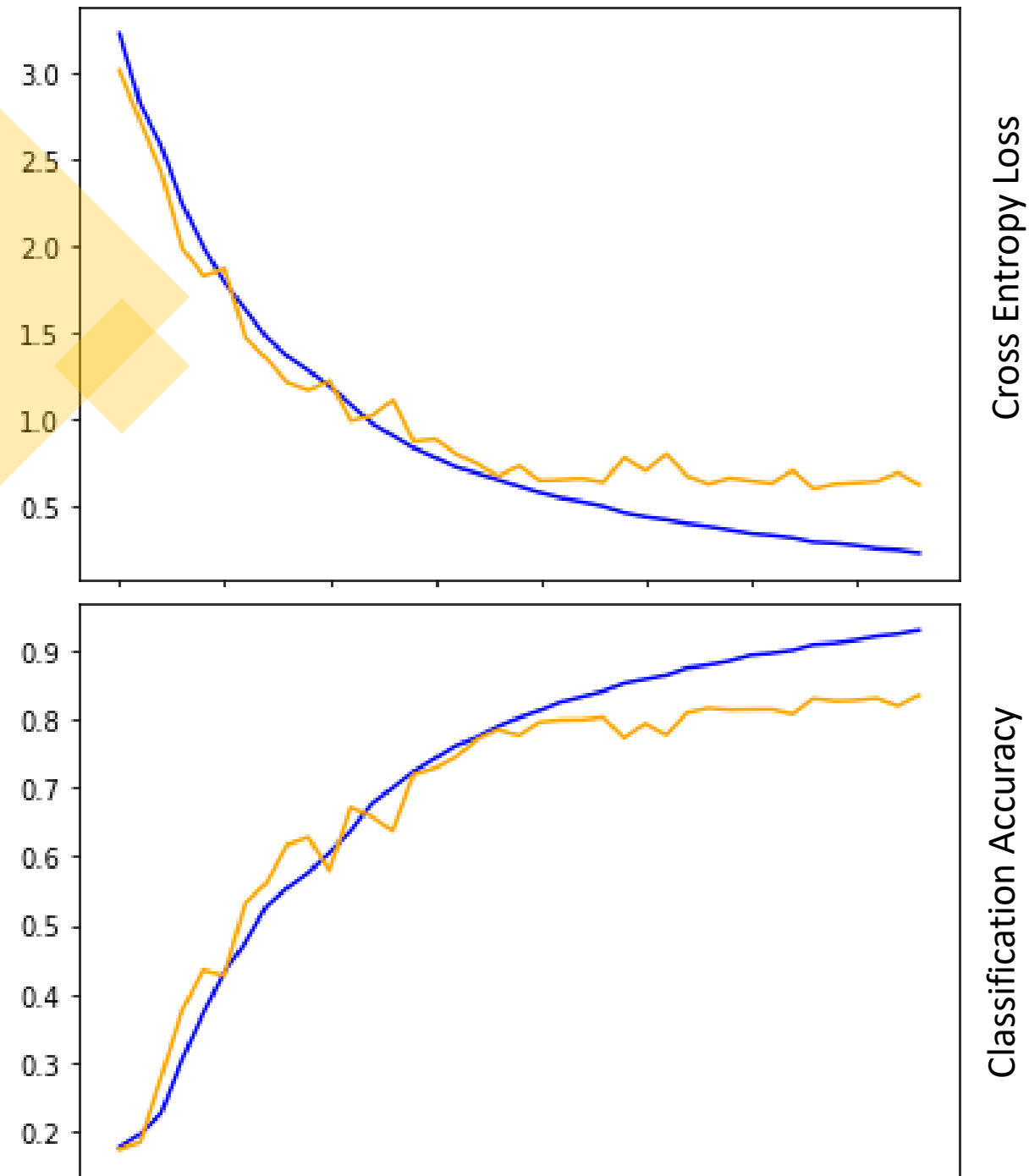
Total params: 1,676,874 Trainable params: 1,674,826 Non-trainable params: 2,048			
---	--	--	--

MUESTRA 8

Cambios introducidos y comentarios:

- Replicamos la arquitectura del modelo VGG16.
- Dejamos solo dos capas dense y dejamos el drop-out final a 0.1, para que los drop-outs sean descendentes en la arquitectura.
- Los resultados mejoran de forma exponencial gracias a la réplica del modelo VGG16.
- Con Adam los resultados son ligeramente inferiores (81,16), por lo que seguiremos con SGD.

Resultado en accuracy: 82,82



MUESTRA 8 - Arquitectura

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 32, 32, 64)	1792
conv2d_1 (Conv2D)	(None, 32, 32, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
dropout (Dropout)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73856
conv2d_3 (Conv2D)	(None, 16, 16, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout_1 (Dropout)	(None, 8, 8, 128)	0
conv2d_4 (Conv2D)	(None, 8, 8, 256)	295168
conv2d_5 (Conv2D)	(None, 8, 8, 256)	590080
conv2d_6 (Conv2D)	(None, 8, 8, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 256)	0
dropout_2 (Dropout)	(None, 4, 4, 256)	0

conv2d_7 (Conv2D)	(None, 4, 4, 512)	1180160
conv2d_8 (Conv2D)	(None, 4, 4, 512)	2359808
conv2d_9 (Conv2D)	(None, 4, 4, 512)	2359808
conv2d_10 (Conv2D)	(None, 4, 4, 512)	2359808
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 512)	0
conv2d_11 (Conv2D)	(None, 2, 2, 512)	2359808
conv2d_12 (Conv2D)	(None, 2, 2, 512)	2359808
dropout_3 (Dropout)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 512)	1049088
batch_normalization (Batch Normalization)	(None, 512)	2048
dense_1 (Dense)	(None, 512)	262656
dropout_4 (Dropout)	(None, 512)	0
batch_normalization_1 (Batch Normalization)	(None, 512)	2048
dense_2 (Dense)	(None, 10)	5130
=====		
Total params: 16,035,658 Trainable params: 16,033,610 Non-trainable params: 2,048		

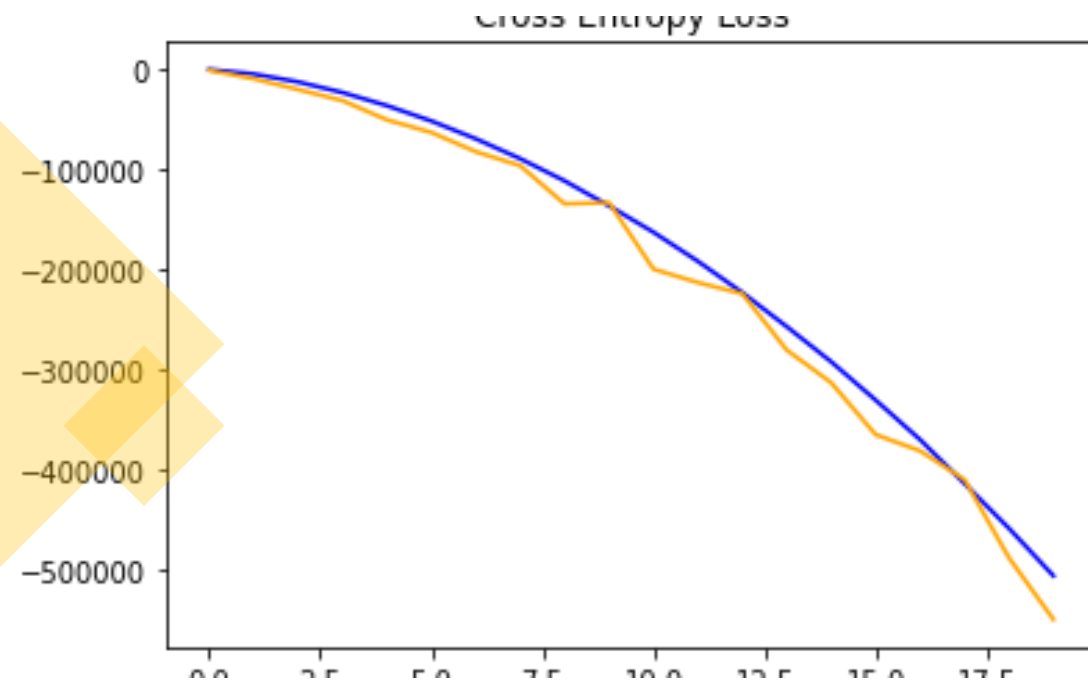
MUESTRA 9

Cambios introducidos y comentarios:

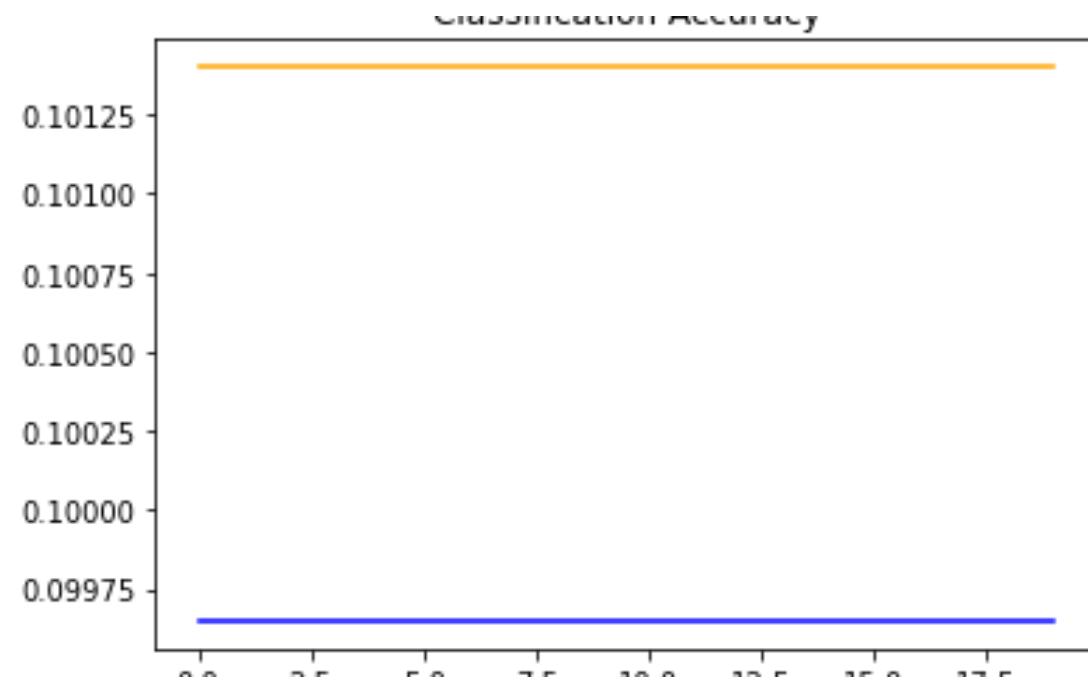
- Implementamos un transfer learning con Imagenet mediante la VGG16.
- Los resultados son excesivamente bajos. El transfer learning con VGG16 no funciona bien con el proyecto, por lo que lo omitiremos. En la muestra v9_2 también lo hemos intentado con ResNet50, pero los resultados son similares.

Resultado en accuracy: 10.00

Cross Entropy Loss



Classification Accuracy



MUESTRA 9 - Arquitectura

Model: "model"	Layer
(type) Output Shape Param #	
=====	input_1
(InputLayer) [(None, 32, 32, 3)] 0	
block1_conv1 (Conv2D) (None, 32, 32, 64) 1792	
block1_conv2 (Conv2D) (None, 32, 32, 64) 36928	
block1_pool (MaxPooling2D) (None, 16, 16, 64) 0	
block2_conv1 (Conv2D) (None, 16, 16, 128) 73856	
block2_conv2 (Conv2D) (None, 16, 16, 128) 147584	
block2_pool (MaxPooling2D) (None, 8, 8, 128) 0	
block3_conv1 (Conv2D) (None, 8, 8, 256) 295168	
block3_conv2 (Conv2D) (None, 8, 8, 256) 590080	
block3_conv3 (Conv2D) (None, 8, 8, 256) 590080	
block3_pool (MaxPooling2D) (None, 4, 4, 256) 0	
block4_conv1 (Conv2D) (None, 4, 4, 512) 1180160	
block4_conv2 (Conv2D) (None, 4, 4, 512) 2359808	
block4_conv3 (Conv2D) (None, 4, 4, 512) 2359808	
block4_pool (MaxPooling2D) (None, 2, 2, 512) 0	
block5_conv1 (Conv2D) (None, 2, 2, 512) 2359808	
block5_conv2 (Conv2D) (None, 2, 2, 512) 2359808	
block5_conv3 (Conv2D) (None, 2, 2, 512) 2359808	
block5_pool (MaxPooling2D) (None, 1, 1, 512) 0	
(Flatten) (None, 512) 0	flatten
=====	Total
params: 14,714,688 Trainable params: 14,714,688 Non-trainable params: 0	

Model: "sequential"
_ Layer (type) Output Shape Param #
=====
= dense (Dense) (None, 512) 262656
_ batch_normalization (BatchNo (None, 512) 2048
_ dropout (Dropout) (None, 512) 0
_ dense_1 (Dense) (None, 512) 262656
_ batch_normalization_1 (Batch (None, 512) 2048
_ dropout_1 (Dropout) (None, 512) 0
_ dense_2 (Dense) (None, 1) 513
=====
= Total params: 529,921 Trainable params: 527,873 Non-trainable params: 2,048
=====
_

MUESTRA 10

Cambios introducidos y comentarios:

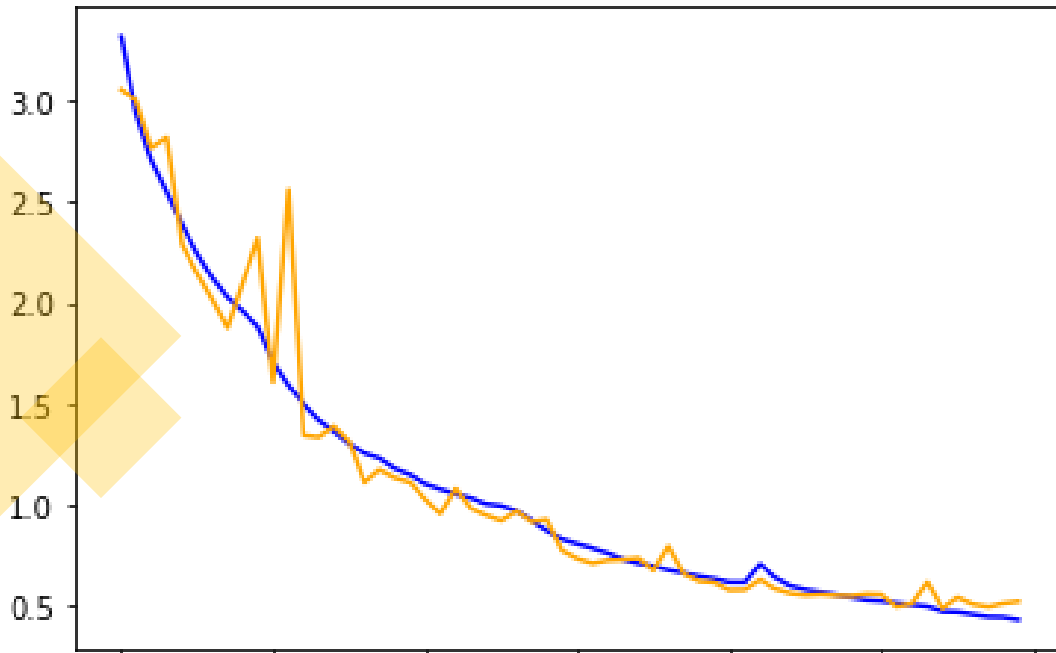
- Aplicamos técnicas de Data Augmentation al modelo de la muestra 8.
- El Image Data Generator tendrá los parámetros:
 - Horizontal_flip
 - Width_shift
 - Height_shift_range

Y quitaremos los rescale, que provocan que las imágenes se vean en negro.

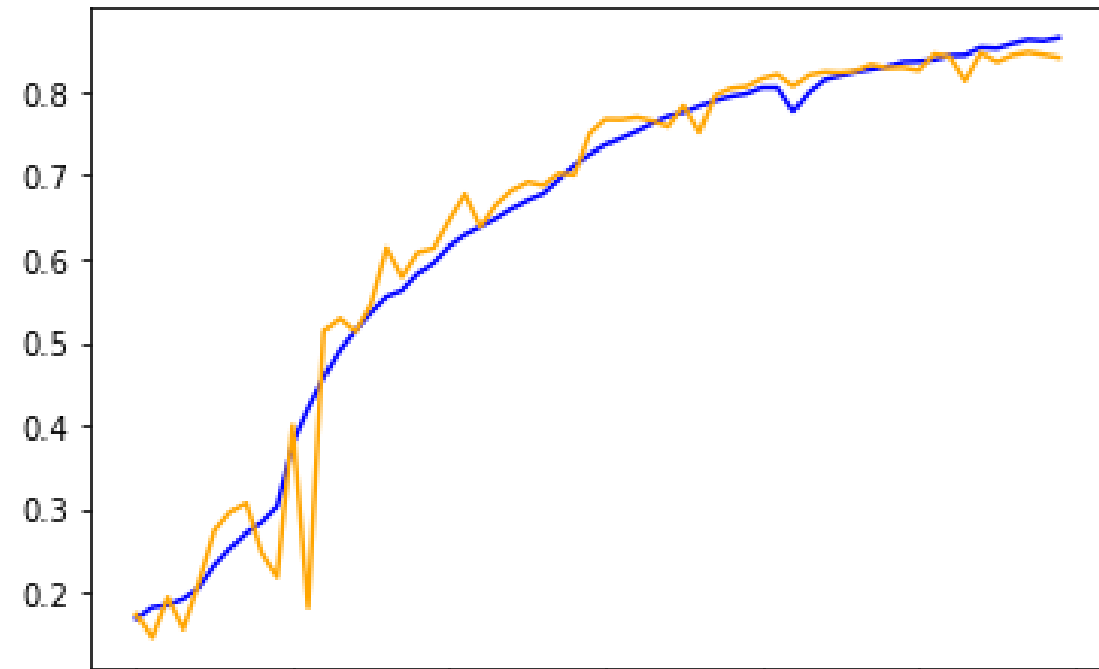
- Además, los steps_per_epoch tendrán el valor de entrada del shape dividido por el batch.
- Los resultados mejoran ligeramente el modelo anterior.

Resultado en accuracy: 83.84

Cross Entropy Loss



Classification Accuracy



MUESTRA 10 - Arquitectura

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 64)	1792
conv2d_1 (Conv2D)	(None, 32, 32, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
dropout (Dropout)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73856
conv2d_3 (Conv2D)	(None, 16, 16, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout_1 (Dropout)	(None, 8, 8, 128)	0
conv2d_4 (Conv2D)	(None, 8, 8, 256)	295168
conv2d_5 (Conv2D)	(None, 8, 8, 256)	590080
conv2d_6 (Conv2D)	(None, 8, 8, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 256)	0
dropout_2 (Dropout)	(None, 4, 4, 256)	0
conv2d_7 (Conv2D)	(None, 4, 4, 512)	1180160
conv2d_8 (Conv2D)	(None, 4, 4, 512)	2359808
conv2d_9 (Conv2D)	(None, 4, 4, 512)	2359808
conv2d_10 (Conv2D)	(None, 4, 4, 512)	2359808

max_pooling2d_3 (MaxPooling2D) (None, 2, 2, 512) 0

conv2d_11 (Conv2D) (None, 2, 2, 512) 2359808

conv2d_12 (Conv2D) (None, 2, 2, 512) 2359808

dropout_3 (Dropout) (None, 2, 2, 512) 0

flatten (Flatten) (None, 2048) 0

dense (Dense) (None, 512) 1049088

batch_normalization (Batch Normalization) (None, 512) 2048

dense_1 (Dense) (None, 512) 262656

dropout_4 (Dropout) (None, 512) 0

batch_normalization_1 (Batch Normalization) (None, 512) 2048

dense_2 (Dense) (None, 10) 5130

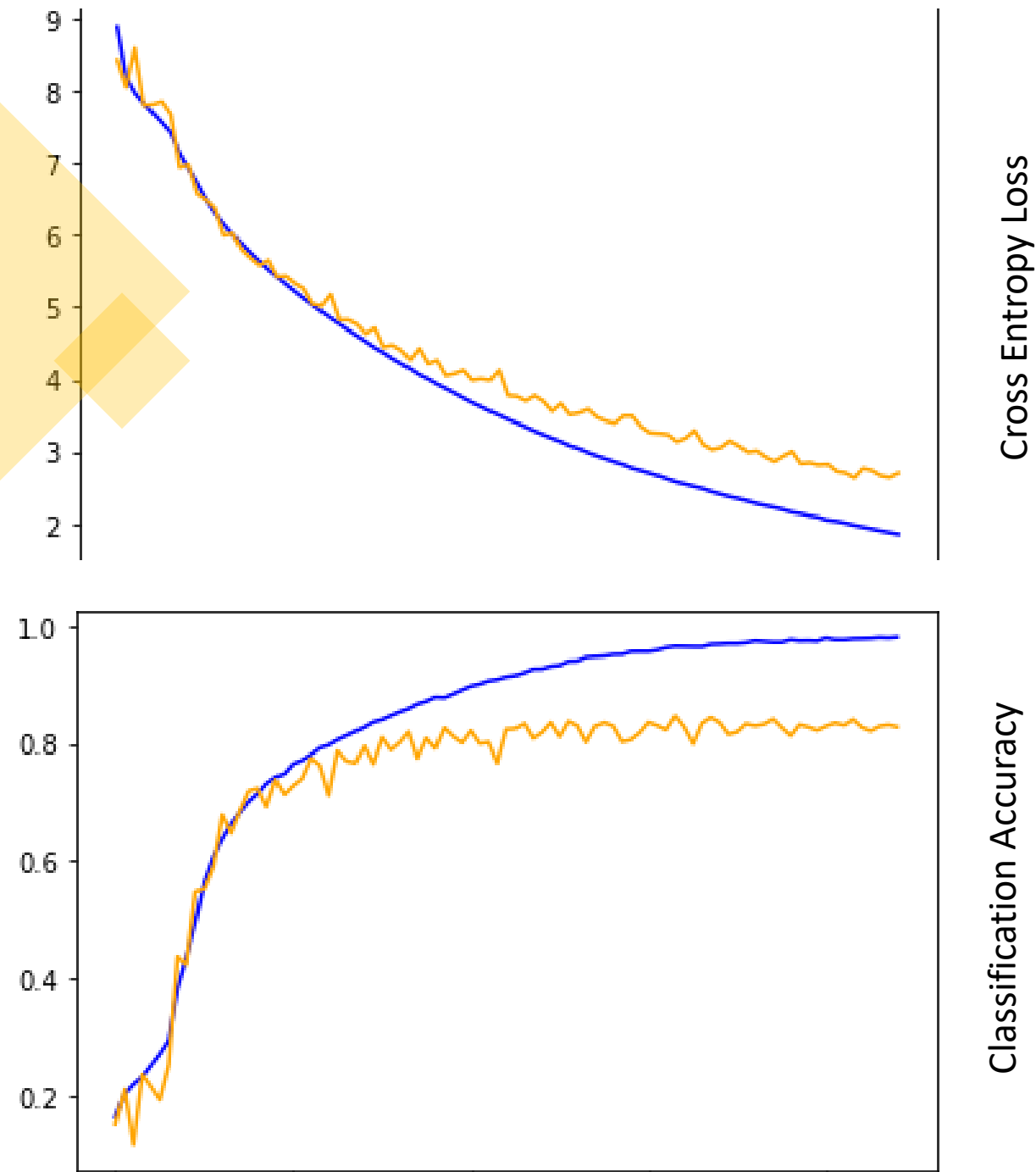
=====
Total params: 16,035,658 Trainable params: 16,033,610 Non-trainable params: 2,048

MUESTRA 11

Cambios introducidos y comentarios:

- Volvemos a la muestra n° 8 para realizar los cambios y asegurar una arquitectura final completa y bien ajustada antes de implementar el Data Augmentation.
- Modificamos el weight decay a 0,0005 en las convoluciones porque parece funcionar mejor con un parámetro más pequeño.
- Modificamos los drop-out de forma que sean ascendentes en la arquitectura, empezando por 0.2 en las convoluciones y terminando por 0.5 en las capas dense.
- El modelo empeora ligeramente aunque puede deberse al balanceo de los pesos. Vamos a juntar esta arquitectura con el Data Augmentation.

Resultado en accuracy: 82.67



MUESTRA 11 - Arquitectura

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 32, 32, 64)	1792
conv2d_1 (Conv2D)	(None, 32, 32, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
dropout (Dropout)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73856
conv2d_3 (Conv2D)	(None, 16, 16, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout_1 (Dropout)	(None, 8, 8, 128)	0
conv2d_4 (Conv2D)	(None, 8, 8, 256)	295168
conv2d_5 (Conv2D)	(None, 8, 8, 256)	590080
conv2d_6 (Conv2D)	(None, 8, 8, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 256)	0
dropout_2 (Dropout)	(None, 4, 4, 256)	0

conv2d_7 (Conv2D)	(None, 4, 4, 512)	1180160
conv2d_8 (Conv2D)	(None, 4, 4, 512)	2359808
conv2d_9 (Conv2D)	(None, 4, 4, 512)	2359808
conv2d_10 (Conv2D)	(None, 4, 4, 512)	2359808
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 512)	0
conv2d_11 (Conv2D)	(None, 2, 2, 512)	2359808
conv2d_12 (Conv2D)	(None, 2, 2, 512)	2359808
dropout_3 (Dropout)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 512)	1049088
batch_normalization (Batch Normalization)	(None, 512)	2048
dropout_4 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
batch_normalization_1 (Batch Normalization)	(None, 512)	2048
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130
=====		
Total params: 16,035,658 Trainable params: 16,033,610 Non-trainable params: 2,048		

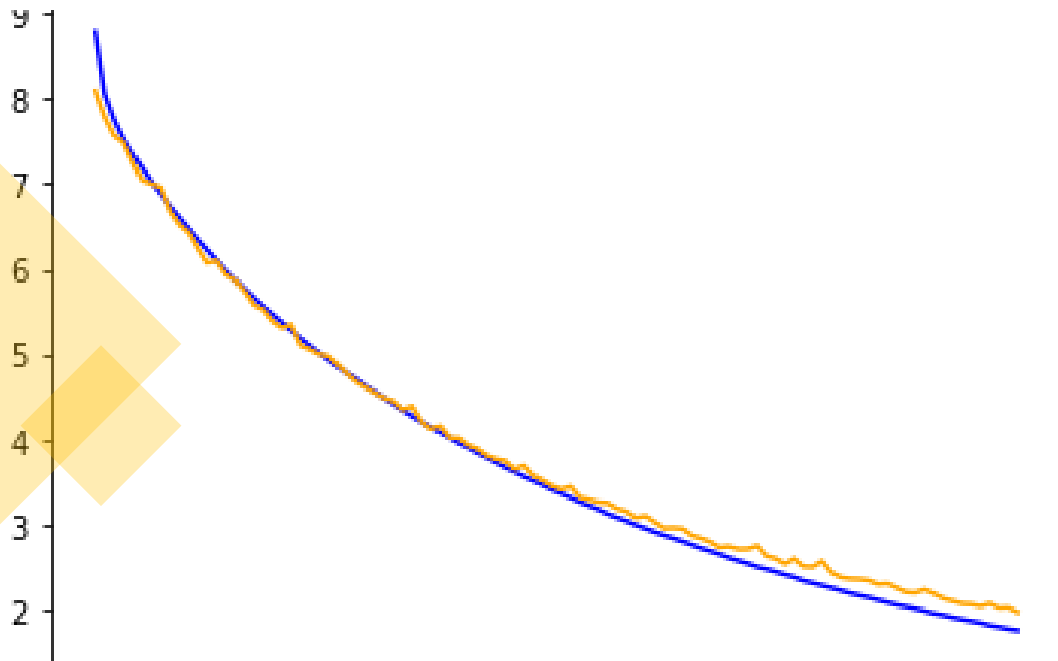
MUESTRA 12

Cambios introducidos y comentarios:

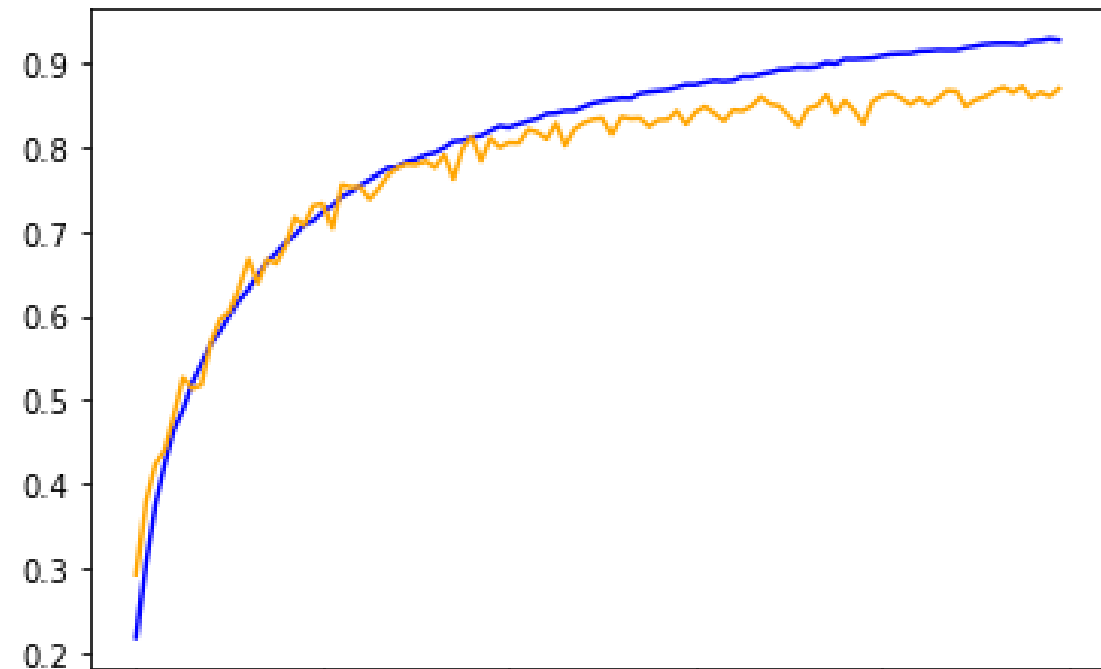
- Adoptamos la arquitectura del modelo 11 con los ajustes en Batch Normalization, Kernel Regularizer y Kernel Initializer.
- Añadimos Batch normalization a todas las capas de convolución.
- El modelo mejora el accuracy ostensiblemente en esta última versión gracias a la combinación de Data Augmentation, una arquitectura profunda basada en VGG16, un optimizador SGD con un learning rate de 0,001 y unos ajustes en la arquitectura que permiten afinar el resultado, como drop-outs de valor ascendente, batch normalization y weight decay.

Resultado en accuracy: 87.29

Cross Entropy Loss



Classification Accuracy



MUESTRA 12 - Arquitectura

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 32, 32, 64)	1792
batch_normalization (Batch Normalization)	(None, 32, 32, 64)	256
conv2d_1 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_1 (Batch Normalization)	(None, 32, 32, 64)	256
max_pooling2d (MaxPooling2D)	(None, 16, 16, 64)	0
dropout (Dropout)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73856
batch_normalization_2 (Batch Normalization)	(None, 16, 16, 128)	512
conv2d_3 (Conv2D)	(None, 16, 16, 128)	147584
batch_normalization_3 (Batch Normalization)	(None, 16, 16, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout_1 (Dropout)	(None, 8, 8, 128)	0
conv2d_4 (Conv2D)	(None, 8, 8, 256)	295168
batch_normalization_4 (Batch Normalization)	(None, 8, 8, 256)	1024
conv2d_5 (Conv2D)	(None, 8, 8, 256)	590080
batch_normalization_5 (Batch Normalization)	(None, 8, 8, 256)	1024
conv2d_6 (Conv2D)	(None, 8, 8, 256)	590080
batch_normalization_6 (Batch Normalization)	(None, 8, 8, 256)	1024
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 256)	0
dropout_2 (Dropout)	(None, 4, 4, 256)	0

conv2d_7 (Conv2D)	(None, 4, 4, 512)	1180160	
batch_normalization_7 (Batch Normalization)	(None, 4, 4, 512)	2048	conv2d_8
(Conv2D)	(None, 4, 4, 512)	2359808	
batch_normalization_8 (Batch Normalization)	(None, 4, 4, 512)	2048	conv2d_9
(Conv2D)	(None, 4, 4, 512)	2359808	
batch_normalization_9 (Batch Normalization)	(None, 4, 4, 512)	2048	conv2d_10
(Conv2D)	(None, 4, 4, 512)	2359808	
batch_normalization_10 (Batch Normalization)	(None, 4, 4, 512)	2048	conv2d_11
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 512)	0	
(Conv2D)	(None, 2, 2, 512)	2359808	conv2d_12
batch_normalization_11 (Batch Normalization)	(None, 2, 2, 512)	2048	
(Conv2D)	(None, 2, 2, 512)	2359808	dropout_3
batch_normalization_12 (Batch Normalization)	(None, 2, 2, 512)	2048	
(Dropout)	(None, 2, 2, 512)	0	flatten
(Flatten)	(None, 2048)	0	
(None, 512)	1049088		dense (Dense)
batch_normalization_13 (Batch Normalization)	(None, 512)	2048	dropout_4
(Dropout)	(None, 512)	0	
(Dense)	(None, 512)	262656	dense_1
batch_normalization_14 (Batch Normalization)	(None, 512)	2048	dropout_5
(Dropout)	(None, 512)	0	
(Dense)	(None, 10)	5130	dense_2
=====			Total params:
16,052,554 Trainable params: 16,042,058 Non-trainable params: 10,496			