# Primera ejecución del modelo CNN simple

## Cambios vs. Version anterior:

* 1. Se ha añadido una capa de Dropout, tras la única capa Dense, para desactivar algunas redes de neuronas en las diferentes iteraciones/steps del proceso de training para tratar de reducir el overfitting.
  2. Se ha aplicado a este Dropout un ratio de 0.3 (30%), que implica el “drop”/desactivación del 30% de las neuronas

## Estructura:

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras import Sequential

from tensorflow.keras.layers import Conv2D, MaxPool2D, Dropout, Flatten, \

Dense

import tensorflow as tf

physical\_devices = tf.config.experimental.list\_physical\_devices('GPU')

tf.config.experimental.set\_memory\_growth(physical\_devices[0], True)

training\_datagen = ImageDataGenerator(rescale=1. / 255,

rotation\_range=0.2,

shear\_range=0.05,

zoom\_range=[0.95, 1.2],

horizontal\_flip=True, )

test\_datagen = ImageDataGenerator(rescale=1. / 255)

training\_path = '/home/ruben/workspace/tfg/deep-learning-facial-recognition/data/age/training'

test\_path = '/home/ruben/workspace/tfg/deep-learning-facial-recognition/data/age/test'

training\_set = training\_datagen.flow\_from\_directory(training\_path,

target\_size=(64, 64),

batch\_size=32,

class\_mode='categorical',

shuffle=True,

seed=42)

test\_set = test\_datagen.flow\_from\_directory(test\_path,

target\_size=(64, 64),

batch\_size=32,

class\_mode='categorical',

shuffle=True,

seed=42)

age\_classifier = Sequential()

age\_classifier.add(

Conv2D(filters=32, kernel\_size=(3, 3), input\_shape=(64, 64, 3),

activation='relu'))

age\_classifier.add(MaxPool2D(pool\_size=(2, 2)))

age\_classifier.add(

Conv2D(filters=32, kernel\_size=(3, 3), activation='relu'))

age\_classifier.add(MaxPool2D(pool\_size=(2, 2)))

age\_classifier.add(Flatten())

age\_classifier.add(Dense(units=128, activation='relu'))

age\_classifier.add(Dropout(0.3))

age\_classifier.add(Dense(units=6, activation='softmax'))

age\_classifier.compile(optimizer='adam', loss='categorical\_crossentropy',

metrics=['accuracy'])

age\_classifier.fit(training\_set,

steps\_per\_epoch=(18966 // 32),

epochs=25,

validation\_data=test\_set,

validation\_steps=(4742//32))

## Resultado:

1. Epoch 1/25
2. 2020-05-20 13:16:39.136915: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:44] Successfully opened dynamic library libcublas.so.10
3. 2020-05-20 13:16:39.305077: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:44] Successfully opened dynamic library libcudnn.so.7
4. 592/592 [==============================] - 36s 61ms/step - loss: 1.2717 - accuracy: 0.4782 - val\_loss: 1.0315 - val\_accuracy: 0.5505
5. Epoch 2/25
6. 592/592 [==============================] - 37s 63ms/step - loss: 1.0457 - accuracy: 0.5542 - val\_loss: 0.9609 - val\_accuracy: 0.5830
7. Epoch 3/25
8. 592/592 [==============================] - 37s 63ms/step - loss: 0.9675 - accuracy: 0.5884 - val\_loss: 0.9052 - val\_accuracy: 0.6073
9. Epoch 4/25
10. 592/592 [==============================] - 37s 62ms/step - loss: 0.9194 - accuracy: 0.6031 - val\_loss: 0.8898 - val\_accuracy: 0.6195
11. Epoch 5/25
12. 592/592 [==============================] - 38s 64ms/step - loss: 0.8821 - accuracy: 0.6184 - val\_loss: 0.8713 - val\_accuracy: 0.6237
13. Epoch 6/25
14. 592/592 [==============================] - 38s 64ms/step - loss: 0.8562 - accuracy: 0.6257 - val\_loss: 0.8842 - val\_accuracy: 0.6060
15. Epoch 7/25
16. 592/592 [==============================] - 38s 64ms/step - loss: 0.8380 - accuracy: 0.6355 - val\_loss: 0.8376 - val\_accuracy: 0.6299
17. Epoch 8/25
18. 592/592 [==============================] - 38s 64ms/step - loss: 0.8218 - accuracy: 0.6399 - val\_loss: 0.8334 - val\_accuracy: 0.6326
19. Epoch 9/25
20. 592/592 [==============================] - 39s 65ms/step - loss: 0.8108 - accuracy: 0.6446 - val\_loss: 0.8295 - val\_accuracy: 0.6404
21. Epoch 10/25
22. 592/592 [==============================] - 38s 65ms/step - loss: 0.7940 - accuracy: 0.6505 - val\_loss: 0.8059 - val\_accuracy: 0.6461
23. Epoch 11/25
24. 592/592 [==============================] - 38s 65ms/step - loss: 0.7807 - accuracy: 0.6522 - val\_loss: 0.7992 - val\_accuracy: 0.6518
25. Epoch 12/25
26. 592/592 [==============================] - 38s 65ms/step - loss: 0.7718 - accuracy: 0.6600 - val\_loss: 0.8407 - val\_accuracy: 0.6208
27. Epoch 13/25
28. 592/592 [==============================] - 39s 65ms/step - loss: 0.7645 - accuracy: 0.6621 - val\_loss: 0.7825 - val\_accuracy: 0.6560
29. Epoch 14/25
30. 592/592 [==============================] - 38s 64ms/step - loss: 0.7509 - accuracy: 0.6682 - val\_loss: 0.8032 - val\_accuracy: 0.6448
31. Epoch 15/25
32. 592/592 [==============================] - 39s 65ms/step - loss: 0.7453 - accuracy: 0.6741 - val\_loss: 0.8010 - val\_accuracy: 0.6484
33. Epoch 16/25
34. 592/592 [==============================] - 38s 64ms/step - loss: 0.7343 - accuracy: 0.6784 - val\_loss: 0.7904 - val\_accuracy: 0.6582
35. Epoch 17/25
36. 592/592 [==============================] - 38s 65ms/step - loss: 0.7321 - accuracy: 0.6818 - val\_loss: 0.8096 - val\_accuracy: 0.6432
37. Epoch 18/25
38. 592/592 [==============================] - 39s 65ms/step - loss: 0.7223 - accuracy: 0.6823 - val\_loss: 0.7974 - val\_accuracy: 0.6586
39. Epoch 19/25
40. 592/592 [==============================] - 38s 65ms/step - loss: 0.7122 - accuracy: 0.6862 - val\_loss: 0.7970 - val\_accuracy: 0.6620
41. Epoch 20/25
42. 592/592 [==============================] - 38s 65ms/step - loss: 0.7140 - accuracy: 0.6889 - val\_loss: 0.7977 - val\_accuracy: 0.6535
43. Epoch 21/25
44. 592/592 [==============================] - 39s 66ms/step - loss: 0.7008 - accuracy: 0.6876 - val\_loss: 0.7982 - val\_accuracy: 0.6636
45. Epoch 22/25
46. 592/592 [==============================] - 39s 65ms/step - loss: 0.6936 - accuracy: 0.6914 - val\_loss: 0.8103 - val\_accuracy: 0.6586
47. Epoch 23/25
48. 592/592 [==============================] - 41s 70ms/step - loss: 0.6874 - accuracy: 0.6983 - val\_loss: 0.8138 - val\_accuracy: 0.6628
49. Epoch 24/25
50. 592/592 [==============================] - 44s 75ms/step - loss: 0.6761 - accuracy: 0.7022 - val\_loss: 0.8556 - val\_accuracy: 0.6446
51. Epoch 25/25
52. 592/592 [==============================] - 41s 69ms/step - loss: 0.6797 - accuracy: 0.7020 - val\_loss: 0.8070 - val\_accuracy: 0.6594
53. Process finished with exit code 0

## Conclusiones y Resultado vs. Version Anterior:

Las diferencias entre la precisión de Training y de Test se han reducido, estando ambas más cercanas. Concretamente, lo que ha sucedido es que la tasa acierto en training ha disminuido con respecto a la arquitectura anterior, donde tenia un 76.5% de acierto en training.

Valores actuales:

- Acc. Training: 70.2%

- Acc. Test: 65.94%