|  |
| --- |
| Asignatura: |
| Inteligencia Artificial II |

|  |
| --- |
| Título del documento: |
| **Laboratorio 3:**  **Implementación de modelos**  **neuronales profundos** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Fecha |
| Grupo: |  | 2 |  | 3/04/18 |
|  |  |  |  |  |
| Miembros: | 1- | Gonzalo de las Heras |  |  |
|  |  |  |  |  |
|  | 2- | Jorge de la Fuente |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre de fichero: |  | Fecha: |  |  |  |  |
| LAB03-GRUPO02-MEMORIA.docx **(.pdf)** |  | **3/04/18** |  |  |  |  |

Índice

[1 Laboratorio 3. Implementación de modelos neuronales profundos 5](#_Toc510520829)

[1.1 Práctica 1. MLP multicapa con Keras: supervivientes del Titanic 5](#_Toc510520830)

[1.1.1 Objetivo 5](#_Toc510520831)

[1.1.2 Cuestiones 5](#_Toc510520832)

[1.1.2.1 Cuestión 1 5](#_Toc510520833)

[1.1.2.2 Cuestión 2 5](#_Toc510520834)

[1.1.2.3 Cuestión 3 5](#_Toc510520835)

[1.1.2.4 Cuestión 4 5](#_Toc510520836)

[1.2 Práctica 2. DNN para clasificar imágenes 6](#_Toc510520837)

[1.2.1 Objetivo 6](#_Toc510520838)

[1.2.2 Cuestiones 6](#_Toc510520839)

[1.2.2.1 Cuestión 1 6](#_Toc510520840)

[1.2.2.2 Cuestión 2 6](#_Toc510520841)

[1.3 Práctica 3. DNN para clasificar imágenes 6](#_Toc510520842)

[1.3.1 Objetivo 6](#_Toc510520843)

[1.3.2 Cuestiones 6](#_Toc510520844)

[1.3.2.1 Cuestión 1 6](#_Toc510520845)

[1.3.2.2 Cuestión 2 6](#_Toc510520846)

[2 Resultados 7](#_Toc510520847)

[2.1 Práctica 1. MLP multicapa con Keras: supervivientes del Titanic 7](#_Toc510520848)

[2.2 Práctica 2. DNN para clasificar imágenes 11](#_Toc510520849)

[2.3 Práctica 3. DNN para clasificar imágenes 17](#_Toc510520850)

[3 Discusión general 22](#_Toc510520851)

[3.1 Práctica 1. Fundamentos de redes neuronales 22](#_Toc510520852)

[3.1.1 Cuestión 1 22](#_Toc510520853)

[3.1.2 Cuestión 2 23](#_Toc510520854)

[3.1.3 Cuestión 3 23](#_Toc510520855)

[3.1.4 Cuestión 4 24](#_Toc510520856)

[3.2 Práctica 2. DNN para clasificar imágenes 24](#_Toc510520857)

[3.2.1 Cuestión 1 24](#_Toc510520858)

[3.2.2 Cuestión 2 24](#_Toc510520859)

[3.3 Práctica 3. DNN para clasificar imágenes 24](#_Toc510520860)

[3.3.1 Cuestión 1 24](#_Toc510520861)

[3.3.2 Cuestión 2 25](#_Toc510520862)

[4 Reflexiones finales 26](#_Toc510520863)

[5 Bibliografía 27](#_Toc510520864)

[6 ANEXO A. Documentos adjuntos 28](#_Toc510520865)

Índice de ilustraciones

[Ilustración 2‑1: Práctica 1. Evolución loss y val\_loss con respecto a los epochs. 9](#_Toc510520866)

[Ilustración 2‑2: Práctica 1. Evolución acc y val\_acc con respecto a los epochs. 9](#_Toc510520867)

[Ilustración 2‑3: Práctica 3. Arquitectura de la red usada. 21](#_Toc510520868)

[Ilustración 3‑1: Práctica 1. Instrucción para rellenar los datos de edad vacíos. 22](#_Toc510520869)

[Ilustración 3‑2: Práctica 1. Instrucción para rellenar los datos de género vacíos. 22](#_Toc510520870)

Índice de tablas

[Tabla 2‑1: Práctica 1. Resultados pruebas 200 epochs. 7](#_Toc510520871)

[Tabla 2‑2: Práctica 1. Resultados pruebas 250 epochs. 7](#_Toc510520872)

[Tabla 2‑3: Práctica 1. Resultados pruebas 300 epochs. 8](#_Toc510520873)

[Tabla 2‑4: Práctica 1. Resultados pruebas 350 epochs. 8](#_Toc510520874)

[Tabla 2‑5: Práctica 1. Media de los pesos de la capa Entrada-Oculta según característica. 10](#_Toc510520875)

[Tabla 2‑6: Práctica 1. Predicción de la probabilidad de supervivencia. 10](#_Toc510520876)

[Tabla 2‑7: Práctica 2. Resultados experimentación con diferentes redes. 11](#_Toc510520877)

[Tabla 2‑8: Práctica 2. Arquitectura de las distintas redes experimentadas. 12](#_Toc510520878)

[Tabla 2‑9: Práctica 2. Resultados clasificación. 13](#_Toc510520879)

[Tabla 2‑10: Práctica 2. Resultados clasificación (continuación). 14](#_Toc510520880)

[Tabla 2‑11: Práctica 2. Resultados clasificación (continuación). 15](#_Toc510520881)

[Tabla 2‑12: Práctica 2. Clases posibles de clasificación. 16](#_Toc510520882)

[Tabla 2‑13: Práctica 3. Resultados clasificación. 17](#_Toc510520883)

[Tabla 2‑14: Práctica 3. Resultados clasificación (continuación). 18](#_Toc510520884)

[Tabla 2‑15: Práctica 3. Resultados clasificación (continuación). 19](#_Toc510520885)

[Tabla 2‑16: Práctica 3. Resultados clasificación (continuación). 20](#_Toc510520886)

[Tabla 2‑17: Práctica 3. Clases posibles de clasificación. 20](#_Toc510520887)

Listado de abreviaturas

|  |  |
| --- | --- |
| DNN | *Deep Neural Network* |
|  |  |
|  |  |
|  |  |
|  |  |

# Laboratorio 3. Implementación de modelos neuronales profundos

Desarrollar de forma autónoma distintas implementaciones de redes neuronales de aprendizaje supervisado que permitan resolver distintos casos de uso. La práctica comenzará con la construcción de un MLP capaz de hacer predicciones para un caso concreto planteado y continúa con el desarrollo de un modelo de DNN para clasificación de imágenes usando redes de convolución.

## Práctica 1. MLP multicapa con Keras: supervivientes del Titanic

### Objetivo

Utiliza la librería Keras para construir y entrenar un MLP para predecir si los pasajeros del Titanic, en base a varias características, sobreviven o por el contrario perecen. En vez de usar la Regla Delta Generalizada, usaremos Adam como función de modificación de matriz de pesos (optimizer) de la forma que se indica en el apartado de “Implementación”. Responde a las preguntas que se plantean en “Cuestiones”.

### Cuestiones

#### Cuestión 1

Explica cómo has llevado a cabo la normalización de los datos de entrada.

#### Cuestión 2

¿Cuál es la mejor arquitectura? Justifícalo con una tabla que recoja los valores de loss y accuracy para el conjunto de entrenamiento y el de validación para las distintas pruebas que has llevado a cabo.

#### Cuestión 3

Con esa arquitectura, determina el error que se obtiene para el dataset predict\_titanic usando los valores de survive de ese dataset.

#### Cuestión 4

¿Es verdad aquello de las mujeres y los niños primero? ¿Hay más posibilidades siendo pasajero de primera clase (se consideran de primera clase aquellos billetes de más de 500 libras) debido a que los camarotes están más cerca de la cubierta? Justifícalo en base a los resultados de la predicción

## 

## Práctica 2. DNN para clasificar imágenes

### Objetivo

Utiliza lo aprendido en clase respecto a Deep Learning y redes convolucionales para construir un clasificador de imágenes. El dataset usado será CIFAR-10, un dataset de imágenes a color de tamaño 32x32 que están clasificadas en 10 categorías. Crea una red convolucional de la forma que se indica en el apartado de “Implementación”. Responde a las preguntas que se plantean en “Cuestiones”.

### Cuestiones

#### Cuestión 1

Entrena la red usando como criterio de parada un loss ≤ 0,5. Recoge en la memoria los valores de loss y accuracy tanto del conjunto de entrenamiento como del de validación de los distintos experimentos que hayas llevado a cabo indicando la arquitectura de cada una de las redes empleadas. Dibuja la arquitectura de red que mejor clasifica.

#### Cuestión 2

Clasifica el conjunto de imágenes de prueba y recoge en la memoria la salida obtenida para cada imagen tal y como se indica en el apartado implementación.

## Práctica 3. DNN para clasificar imágenes

### Objetivo

Utilizando tu propio criterio, selecciona, explora y analiza un dataset de entre los disponibles en KAGGLE.

### Cuestiones

#### Cuestión 1

Motivación del problema seleccionado y justificación de la solución que se quiere obtener.

#### Cuestión 2

Resultados obtenidos.

# Resultados

## Práctica 1. MLP multicapa con Keras: supervivientes del Titanic

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | **Nº Capas ocultas** | | | |
|  | **Epochs** | **Nº Neuronas Salida** | **1 – 260** | **2 – 260/130** | **3 – 260/130/66** | **4 – 260/130/66/32** |
| **Loss** | 200 | 1 | 0,1636 | 0,1284 | 0,1313 | 0,1237 |
| **Acc** | 0,7743 | 0,8192 | 0,8204 | 0,8329 |
| **Val\_loss** | 0,1280 | 0,1232 | 0,1222 | 0,1378 |
| **Val\_acc** | 0,8202 | 0,7978 | 0,7978 | 0,7978 |

Tabla 2‑1: Práctica 1. Resultados pruebas 200 epochs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | **Nº Capas ocultas** | | | |
|  | **Epochs** | **Nº Neuronas Salida** | **1 – 260** | **2 – 260/130** | **3 – 260/130/66** | **4 – 260/130/66/32** |
| **Loss** | 250 | 1 | 0,1411 | 0,1357 | 0,1379 | 0,1183 |
| **Acc** | 0,8192 | 0,8080 | 0,8204 | 0,8429 |
| **Val\_loss** | 0,1134 | 0,1305 | 0,1192 | 0,1407 |
| **Val\_acc** | 0,8427 | 0,8090 | 0,8090 | 0,7978 |

Tabla 2‑2: Práctica 1. Resultados pruebas 250 epochs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | **Nº Capas ocultas / Nº Neuronas** | | | |
|  | **Epochs** | **Nº Neuronas Salida** | **1 - 260** | **2 – 260/130** | **3 – 260/130/66** | **4 – 260/130/66/32** |
| **Loss** | 300 | 1 | 0,1307 | 0,1178 | 0,1138 | 0,1086 |
| **Acc** | 0,8279 | 0,8367 | 0,8529 | 0,8516 |
| **Val\_loss** | 0,1151 | 0,1209 | 0,1331 | 0,1385 |
| **Val\_acc** | 0,8315 | 0,8315 | 0,8202 | 0,8090 |

Tabla 2‑3: Práctica 1. Resultados pruebas 300 epochs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | **Nº Capas ocultas / Nº Neuronas** | | | |
|  | **Epochs** | **Nº Neuronas Salida** | **1 - 260** | **2 – 260/130** | **3 – 260/130/66** | **4 – 260/130/66/32** |
| **Loss** | 350 | 1 | 0,1397 | 0,1353 | 0,1056 | 0,0955 |
| **Acc** | 0,8142 | 0,8092 | 0,8603 | 0,8666 |
| **Val\_loss** | 0,1234 | 0,1367 | 0,1331 | 0,1351 |
| **Val\_acc** | 0,8315 | 0,8202 | 0,8090 | 0,8315 |

Tabla 2‑4: Práctica 1. Resultados pruebas 350 epochs.

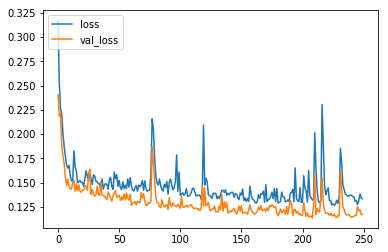


Ilustración ‑: Práctica 1. Evolución loss y val\_loss con respecto a los epochs.

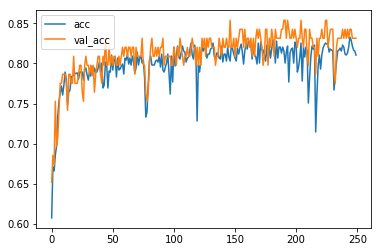


Ilustración 2‑2: Práctica 1. Evolución acc y val\_acc con respecto a los epochs.

|  |  |
| --- | --- |
|  | **Media de los pesos** |
| **Pclass** | -0.04684532 |
| **Sex** | 0.02486094 |
| **Age** | -0.01937371 |
| **Fare** | -0.01995984 |
| **Embarked** | -0.02719517 |

Tabla 2‑5: Práctica 1. Media de los pesos de la capa Entrada-Oculta según característica.

|  |  |
| --- | --- |
|  | **Prob. media** |
| **Mujeres** | 72.67 % |
| **Hombres** | 14.92 % |
| **Niños (15 años o menos)** | 32 % |
| **Primera clase** | 99.88 % |

Tabla 2‑6: Práctica 1. Predicción de la probabilidad de supervivencia.

## Práctica 2. DNN para clasificar imágenes

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Red 1** | **Red 2** | **Red 3** |
| **Loss** | 0.8325 | 0.8142 | 0.4917 |
| **Accuracy** | 0.7161 | 0.7253 | 0.8271 |
| **Val\_Loss** | 0.8124 | 0.8330 | 0.7245 |
| **Val\_Accuracy** | 0.7207 | 0.7147 | 0.7677 |

Tabla 2‑7: Práctica 2. Resultados experimentación con diferentes redes.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Red 1** | **Red 2** | **Red 3** |

Tabla 2‑8: Práctica 2. Arquitectura de las distintas redes experimentadas.

|  |  |
| --- | --- |
| Imagen | Predicción |
|  |  |
|  |  |
|  |  |

Tabla 2‑9: Práctica 2. Resultados clasificación.

|  |  |
| --- | --- |
| Imagen | Predicción |
|  |  |
|  |  |
|  |  |

Tabla 2‑10: Práctica 2. Resultados clasificación (continuación).

|  |  |
| --- | --- |
| Imagen | Predicción |
|  |  |
|  |  |

Tabla 2‑11: Práctica 2. Resultados clasificación (continuación).

|  |
| --- |
| Clase |
| Avión |
| Coche |
| Pájaro |
| Gato |
| Ciervo |
| Perro |
| Rana |
| Caballo |
| Barco |
| Camión |

Tabla ‑: Práctica 2. Clases posibles de clasificación.

## Práctica 3. DNN para clasificar imágenes

|  |  |
| --- | --- |
| Imagen | Predicción |
|  |  |
|  |  |
|  |  |

Tabla ‑: Práctica 3. Resultados clasificación.

|  |  |
| --- | --- |
| Imagen | Predicción |
|  |  |
|  |  |
|  |  |

Tabla ‑: Práctica 3. Resultados clasificación (continuación).

|  |  |
| --- | --- |
| Imagen | Predicción |
|  |  |
|  |  |
|  |  |

Tabla ‑: Práctica 3. Resultados clasificación (continuación).

|  |  |
| --- | --- |
| Imagen | Predicción |
|  |  |

Tabla ‑: Práctica 3. Resultados clasificación (continuación).

|  |  |
| --- | --- |
| Clase | Situación |
| C0 | Normal driving |
| C1 | Texting - right |
| C2 | Talking on the phone - right |
| C3 | Texting - left |
| C4 | Talking on the phone - left |
| C5 | Operating the radio |
| C6 | Drinking |
| C7 | Reaching behind |
| C8 | Hair and makeup |
| C9 | Talking to passenger |

Tabla ‑: Práctica 3. Clases posibles de clasificación.

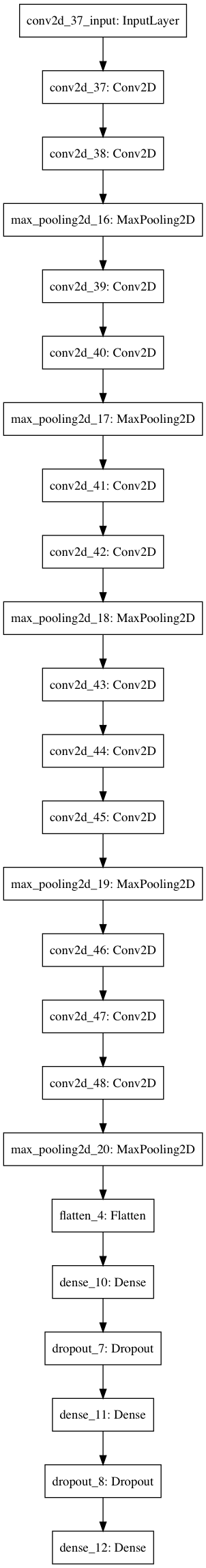


Ilustración ‑: Práctica 3. Arquitectura de la red usada.

# Discusión general

## Práctica 1. Fundamentos de redes neuronales

### Cuestión 1

En la columna de la edad de los pasajeros del Titanic existen algunos valores que presentan el valor “nan” (not a number). Este valor indica que no hay ningún registro sobre la edad del pasajero, de tal manera que vamos a sustituir es valor por la media de edad de todos los pasajeros, con el fin de que este nuevo valor creado artificialmente afecte lo mínimo. Para ello usamos el siguiente código:



Ilustración 3‑1: Práctica 1. Instrucción para rellenar los datos de edad vacíos.

Del mismo, los datos contenidos en la columna “Age” que contengan el valor “nan” son sustituidos por la media de las edades de todos los pasajeros.

En el enunciado de la práctica se nos pedía usar el método *to\_categorical* para las variables *Embarked* (el puerto donde se ha subido el pasajero) y *Sexo* (el género del pasajero).

Tras investigar lo que hacía dicho método, descubrimos que ese método categoriza valores números enteros, pero las variables eran de tipo string. Por lo que no pudimos hacer uso de ese método.

Es por ello por lo que decidimos emplear el método *replace* para que sustituya en los datos los valores que queremos por otros (de esta manera además controlamos el valor que exacto en el que se van a categorizar los datos). El código usado sería el siguiente:

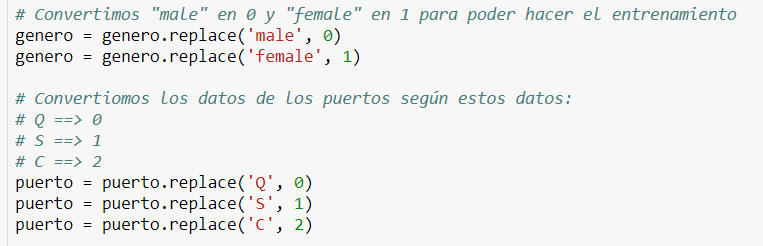


Ilustración 3‑2: Práctica 1. Instrucción para rellenar los datos de género vacíos.

De tal manera que, en la variable Sexo convertimos el valor “male” con 0 y “female” con 1, y en el caso de Embarked sustituimos el valor “Q” que corresponde al puerto Queenstown con 0, el valor “S” que corresponde al puerto Southhampton con 1 y el valor “C” que corresponde al puerto Cherbourg con el valor 2.

### Cuestión 2

Para el proceso de búsqueda del mejor predictor, se ha empleado el método de validación cruzada. Para ello, se ha empleado el 90% de los datos proporcionados, como conjunto de entrenamiento, y el 10% restante, como validación.

El resultado de las pruebas realizadas junto a las arquitecturas empleadas se puede encontrar en las tablas 2-7 y 2-8.

Después del proceso de experimentación, el predictor más eficiente conseguido tiene 0,8427 de precisión (valor conseguido con el conjunto de validación). La red neuronal consta de las siguientes características:

* Capa de entrada: 5 elementos (requeridos por el enunciado de la práctica).
* 1 capa oculta de 260 neuronas (concuerda con el Teorema de Aproximación Universal).
* Capa de salida: 1 neurona.
* Función de error: Mean Squared Error.
* Función de activación de la capa oculta: ReLU.
* Función de activación de la capa de salida: Sigmoide.
* Iteraciones: 250.
* Optimizador: Adam.
* Coeficiente de aprendizaje: 0,001.

El resto de hiperparámetros del optimizador Adam son los propuestos por defecto por la librería Keras:

* B1: 0,9.
* B2: 0,999
* E: 1e-08.
* Decay: 0.0

### Cuestión 3

Para calcular el error realizaremos una comparación entre la columna Survived procedente del dataset titanic.csv y de los resultados obtenidos en el predict de ese mismo dataset (el dataset predict\_titanic.csv no posee esa columna). Se ha realizado la comparación con el 80% del dataset, que es el empleado para el conjunto de entrenamiento (el restante 20% se ha empleado como validación).

Debido a que los resultados del predict son valores de 0 o 1, tenemos que transformar el porcentaje de supervivencia de la predicción a valores 0 o 1. Para ello, aplicaremos una función escalón estableciendo el umbral en 0.50, correspondiente a que tiene más del 50% de posibilidades de sobrevivir, lo pasaremos a 1, y si tiene un valor inferior a 0.50, lo pasamos a 0.

Una vez realizada la transformación de los datos, realizamos la comprobación de cuántos datos coinciden, para poder calcular el porcentaje de error.

Tras comprobar cuantos valores son iguales, y cuantos son distintos, aplicamos una regla de tres para saber cuánto es el porcentaje de error.

El error calculado es de 22%.

### Cuestión 4

Para responder a las preguntas del enunciado se han realizado dos cálculos:

* La media de los pesos que salen de la capa de entrada.
* La media de las probabilidades de supervivencia predichas, según categorías (hombres, mujeres, niños y primera clase).

Los resultados se pueden consultar en las tablas 2-5 y 2-6.

Observando los resultados de la media de supervivencia según categorías, podemos afirmar que las mujeres y los niños fueron puesto a salvo antes que los hombres, ya que tienen mucha más probabilidad de supervivencia.

Además, los pasajeros de primera clase tienen claramente muchas más probabilidades de supervivencia que el resto. Esto también se confirma con la media de los pesos que salen de la entrada correspondiente a esta característica, al ser la media de los pesos más alta.

## Práctica 2. DNN para clasificar imágenes

### Cuestión 1

Para alcanzar la red neuronal que cumple los requisitos del enunciado se ha experimentado con 3 arquitecturas diferentes, las cuales se pueden consultar en la tabla 2-8.

Finalmente, la red 3 se ha establecido como la más eficiente debido al estudio de las métricas obtenidas y que se pueden consultar en la tabla 2-7. En ella vemos como la red 3 tiene los valores de accuracy más altos y loss de los más bajos.

Como apunte, se ha utilizado el “padding = same”, es decir, con zero padding y un “early\_stop” para evitar el sobreentrenamiento de la red.

### Cuestión 2

El resultado de la clasificación según las clases de la tabla 2-12, se puede consultar en las tablas 2-9, 2-10, 2-11, las cuales reflejan un alto rendimiento en la predicción.

## Práctica 3. DNN para clasificar imágenes

### Cuestión 1

Los accidentes de tráfico son un problema de la actualidad y que se dan en el día a día. Muchos factores afectan a la probabilidad de que ocurra un accidente, empezando por el conductor. No es raro contestar un mensaje, hablar con el resto de los acompañantes, comer, manipular la radio… todo ello en mitad de la conducción aumentan considerablemente el riesgo de accidente.

Con ese dataset de conductores realizando diversas tareas, además de la propia de conducir, crearemos un clasificador gracias a la visión por computador que sirva como base para otro software que alerte de cuando un conductor se está distrayendo de la conducción, para corregir su actitud y disminuir la probabilidad de accidentes.

Para ello se ha experimentado con varias arquitecturas de redes, quedando como la mejor la mejor la de la ilustración 2-3.

También se ha empleado la librería de OpenCV para transformar las imágenes en arrays de píxeles, un optimizador RMSprop, la instrucción “padding = same”, es decir, con zero padding, un “early\_stop” para evitar el sobreentrenamiento de la red y “data\_format='channels\_first'” para mantener el orden de las capas RGB. Todo ello recomendado por el creador de Keras, Francois Chollet.

### Cuestión 2

El resultado de la clasificación según las clases de la tabla 2-17, se puede consultar en las tablas 2-13, 2-14, 2-15, 2-16, las cuales reflejan un alto rendimiento en la predicción.

# Reflexiones finales

* Es importante una buena modelización del problema que se pretende resolver para diseñar correctamente la red neuronal que lo resuelva.
* Para el entrenamiento de redes neuronales convoluciones es necesario un gran dataset de imágenes para obtener un buen clasificador.
* Los datos provistos a la red neuronal no siempre son útiles para la predicción. Si previamente se eliminaran estos datos irrelevantes, el rendimiento del predictor mejoraría.
* Antes de entrenar la red, los datos han de ser preprocesados para la óptima utilización de estos.
* Por mucho que se entrene la red, siempre existirá un error, mínimo (local o global, dentro de la superficie de error) pero insalvable.
* La búsqueda de la mejor red neuronal es un proceso de experimentación.
* El tiempo de entrenamiento de una red convolución es muy alto, estudios de rendimiento y optimización de recursos pueden resultar de extrema utilidad.

# Bibliografía

**Brownlee, Jason.** Machine Learning Mastery. *Object Recognition with Convolutional Neural Networks in the Keras Deep Learning Library.* [En línea] https://machinelearningmastery.com/object-recognition-convolutional-neural-networks-keras-deep-learning-library/.

**Chollet, François.** Keras: The Python Deep Learning library. [En línea] https://keras.io/.

**García-Tejedor, Álvaro.** Moodle UFV. Recursos de la asignatura. [En línea] http://moodleufv.ufv.es/moodle/.

**Kaggle Inc.** Kaggle. *State Farm Distracted Driver Detection.* [En línea] https://www.kaggle.com/c/state-farm-distracted-driver-detection.

**ruder.** An overview of gradient descent optimization algorithms. [En línea] http://ruder.io/optimizing-gradient-descent/.

**shaoanlu.** SGD > Adam?? Which One Is The Best Optimizer: Dogs-VS-Cats Toy Experiment. [En línea] https://shaoanlu.wordpress.com/2017/05/29/sgd-all-which-one-is-the-best-optimizer-dogs-vs-cats-toy-experiment/.

# ANEXO A. Documentos adjuntos

* LAB03-GRUPO02-MEMORIA.docx (presente documento).
* Práctica 1 – Titanic:
  + L3P1-Titanic.ipynb
  + predict\_titaniccsv
  + titanic.csv
* Práctica 2 - Cifar 10:
  + L3P2-Imagenes.ipynb
  + red\_conv\_cifar10.h5
  + images/
* Práctica 3 - Conductores distraídos:
  + L3P3-kaggle.ipynb
  + red\_conv\_conductores.h5
  + driver\_imgs\_list.csv

Nota: Los datasets de imágenes de la Práctica 3 - Conductores distraídos no se adjuntan debido al tamaño de las carpetas de varios GB.