David Parreño Barbuzano
Sergio del Ser Muñoz

Práctica 2 Redes Neuronales. Clasificador de peces.

Introducción.

En esta práctica se ha implementado una red neuronal convolutiva que es capaz de clasificar distintos grupos de peces. El dataset que se ha empleado está formado por cuatro clases de peces, en el que cada uno se compone de 1000 imágenes, obtenidas de un dataset de Kaggle (ver https://www.kaggle.com/crowww/a-large-scale-fish-dataset).

Hiperparámetros y parámetros.

Los modelos que se han probado para esta práctica han utilizado los mismos hiperparámetros y parámetros, lo que significa que difieren únicamente en la disposición, número y características de las capas ocultas, es decir, en el diseño de la red neuronal. A continuación, se muestra un conjunto de tablas que contienen el nombre, definición y valor establecido de cada uno de los hiperparámetros y parámetros.

Inicializador del Dataset

Nombre	Definición	Valor
Training percentage	Porcentaje de ficheros para el entrenamiento	80%
Validation percentage	Porcentaje de fichero para la validación	20%

- **Compilación de la red neuronal**

Nombre	Definición	Valor
Loss function	Función de pérdida	Categorical Cross Entropy
Optimizer	Optimizador	Ada Delta
Metrics	Unidad de medida	Accuracy

- **Entrenamiento y EarlyStopping**

Nombre	Definición	Valor
Number of epochs	Número máximo de iteraciones	30
Steps per epoch	Pasos totales para cada época	100
Batch size	Número de ficheros cargados por cada época	30
Patience	Número máximo de intentos para mejorar val_accuracy antes de que finalice el entrenamiento	3

- **Data augmentation**

Nombre	Definición	Valor
Target size	Ancho y alto para cada imagen cargada	250w, 250h
Rescale	Reescalado usado sobre las imágenes	1./255
Rotation range	Rango de rotación para las imágenes	30
Zoom range	Rango del zoom para las imágenes	0.7
Width shift range	Rango del desplazamiento del ancho	0.1
Height shift range	Rango del desplazamiento del alto	0.1
Brightness range	Rango del brillo de las imágenes	(0.2,0.8)
Horizontal flip	Volteo horizontal de las imágenes	True
Vertical flip	Volteo vertical de las imágenes	True

- **Modelos probados**

Nombre	Estructura
ModelDefinitionOneConv	• Conv2D(filters=32, kernel=(2, 2), activation=relu) • MaxPooling(pool_size=(2, 2)) • Dropout(0.25) • Flatten() • Dense(64, activation=relu) • Dropout(0.5) • Dense(32, activation=relu) • Dense(4, activation=softmax)
ModelDefinitionTwoConv	• Conv2D(filters=64, kernel=(2, 2), activation=relu) • MaxPooling(pool_size=(2, 2)) • Dropout(0.25) • Conv2D(filters=128, kernel=(2, 2), activation=relu) • MaxPooling(pool_size=(2, 2)) • Dropout(0.25) • Flatten() • Dense(128, activation=relu) • Dropout(0.25) • Dense(64, activation=relu) • Dense(4, activation=softmax)
ModelDefinitionThreeConv	• Conv2D(filters=32, kernel=(2, 2), activation=relu) • MaxPooling(pool_size=(2, 2)) • Dropout(0.25) • Conv2D(filters=64, kernel=(2, 2), activation=relu) • MaxPooling((2,2)) • Dropout(0.25)

	• Conv2D(filters=128, kernel=(2, 2), activation=relu) • MaxPooling((2,2)) • Dropout(0.25) • Flatten() • Dense(64, activation=relu) • Dropout(0.5) • Dense(32, activation=relu) • Dense(4, activation=softmax)

Los resultados que ofrecen los modelos diseñados se han obtenido a partir de la misma disposición del dataset, usando los mismos valores para los hiperparámetros. Cabe aclarar que puede que se obtengan resultados distintos si el dataset mostrase una organización diferente a la que se usó en las pruebas.

Nombre	Loss	Accuracy	Val_Loss	Val_Accuracy
ModelDefinitionOneConv	1.3097	0.4203	1.1064	0.6775
ModelDefinitionTwoConv	1.2537	0.5282	1.0885	0.6775
ModelDefinitionThreeConv	1.3853	0.2657	1.3765	0.3663

De todos los modelos probados, se ha escogido el segundo, es decir, ModelDefinitionTwoConv. Los motivos de esta elección se resumen en el hecho de que, salvo la precisión del conjunto de validación, tiene la mejor precisión y valor de pérdida. Hay que aclarar que en la matriz de confusión de este modelo se puede observar que realmente es un clasificador binario, porque predice con gran exactitud dos clases mientras que las otras no tan bien, pero igualmente sigue siendo el modelo con la mejor precisión de todas las probadas. Cabe aclarar que es posible que los otros dos llegasen a alcanzar resultados mejores con otros hiperparámetros, como puede ser el número de épocas o pasos por época, que, si fuese mayor, seguramente se obtendrían resultados más altos.

Resultados obtenidos.

Como ya se mencionó anteriormente, de entre todas las posibles configuraciones que se han probado, la que mejor resultados ha ofrecido ha sido el modelo ModelDefinitionTwoConv, llegando a alcanzar alrededor del 48-55% de precisión, y un 65-70% en la validación. Las razones de que la red neuronal no llegue a ser más eficiente y precisa pueden ser varias, y se explicarán a continuación:

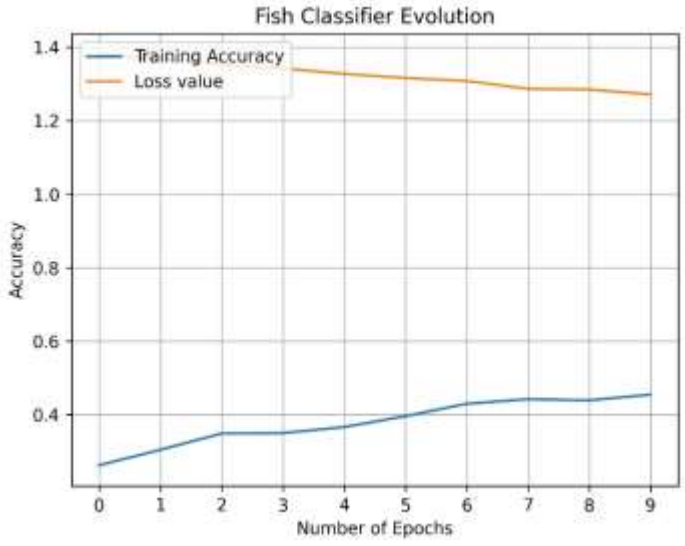
- **Número de pasos por época:** el proceso de entrenamiento se ha restringido de tal manera que en cada época se hagan sólo 100 pasos, cuando realmente este número debería corresponder a la fórmula nºmuestras / batch_size. La explicación de que se haya reducido tanto este número se debe más bien a limitaciones en el hardware, lo que significa que posiblemente la red neuronal llegaría a alcanzar mejores resultados si el número de pasos no se limitase.
- **Número de épocas:** el número de épocas que realiza la red neuronal se ha limitado a 30, por exactamente la misma razón que ocurría en el número de pasos por época, es decir, limitaciones en el hardware empleado durante el entrenamiento.
- **Mala configuración de la red neuronal:** aunque la arquitectura de red que se acabó escogiendo es la que mejor resultados ha ofrecido, ello no implica que no sea mejorable, por lo que puede

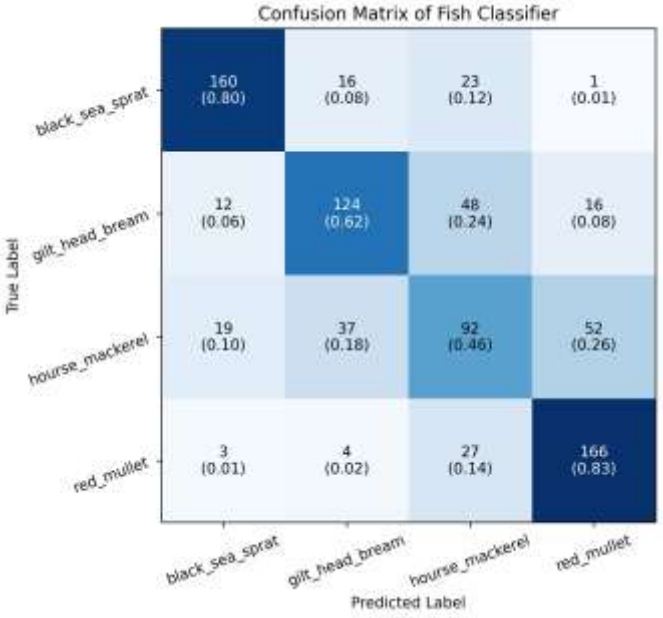
ser que hagan falta más capas o que las existentes se localicen en la red de distinta manera. Este puede ser el motivo más evidente, pues durante el entrenamiento la red neuronal llega en muchas ocasiones a necesitar dos épocas más para superar la mejor precisión alcanzada, lo que supone que, si no se cambiase la red, 30 épocas no serían suficientes.

- **Alto sobreajuste:** en muchas ocasiones la precisión de la validación llega a superar a la del entrenamiento, lo que puede significar que la red no está realmente aprendiendo, sino que trata de mejorar la precisión en la validación, pero no tanto en el entrenamiento. Una solución podría ser añadir más ficheros de test que no se empleen durante en el entrenamiento.
- **Similitud de las clases:** el dataset que se ha escogido para la realización de la práctica está compuesto por imágenes de peces que puede que la red neuronal no sea capaz de diferenciarlos por su gran similitud, siendo en muchas ocasiones el tamaño y el color lo que realmente los hace distintos. Esto se ha intentado arreglar con el data augmentation, usando un dataset amplio, y añadiendo todas las capas convolutivas y filtros que se pueda.

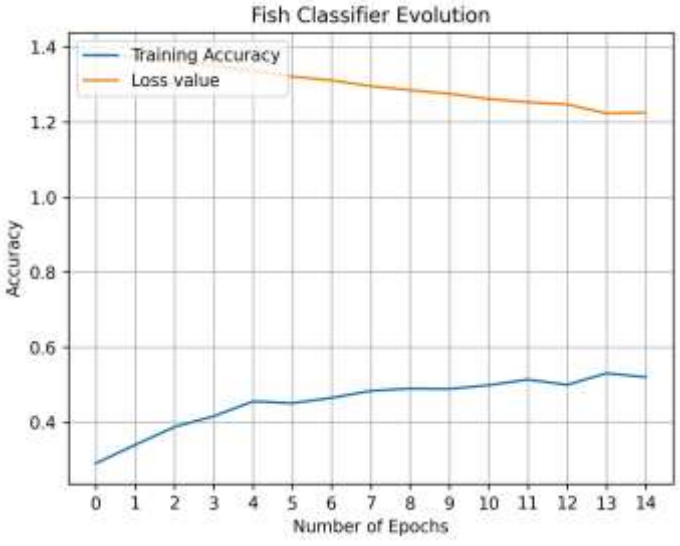
Las siguientes imágenes muestran las gráficas de la precisión y valor de pérdida, así como la matriz de confusión, de cada uno de los modelos diseñados para esta práctica.

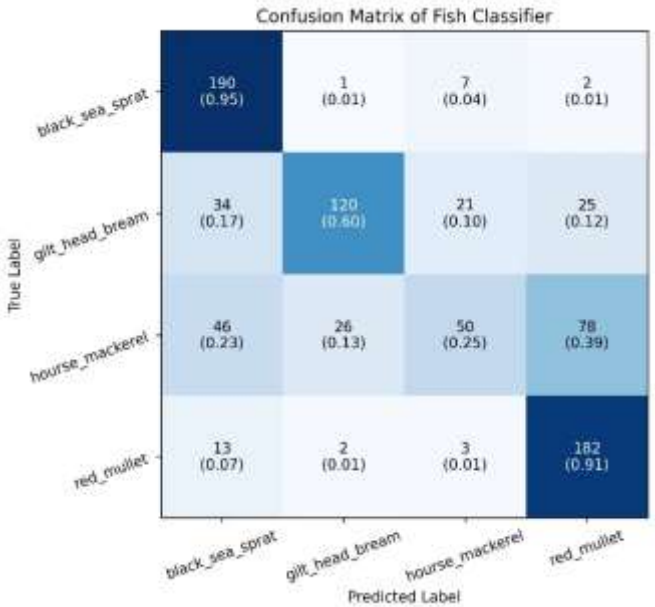
ModelDefinitionOneConv





ModelDefinitionTwoConv





David Parreño Barbuzano
Sergio del Ser Muñoz

ModelDefinitionThreeConv