

Práctica 3

Transferencia del conocimiento



Profesor:
David Campoy Miñarro



¿Qué es la transferencia del conocimiento?

La transferencia de conocimiento es el proceso de aplicar conocimientos adquiridos en una tarea a otra tarea relacionada, acelerando el aprendizaje en la segunda tarea.

Nuestro objetivo:

- Entrenar una red neuronal para que sea capaz de identificar el objeto coche.
- Con el conocimiento adquirido, utilizaremos la red neuronal para identificar el objeto camión.

automobile



Dataset: CIFAR-10

El conjunto de datos CIFAR-10 es un conjunto de imágenes utilizado comúnmente en el campo del aprendizaje automático y la visión por computadora. Contiene 60,000 imágenes a color de 32x32 píxeles, clasificadas en 10 clases distintas. Cada clase contiene 6.000 imágenes.

<https://www.cs.toronto.edu/~kriz/cifar.htm>

1

Categorías:

- Avión
- Automóvil
- Pájaro
- Gato
- Ciervo
- Perro
- Rana
- Caballo
- Barco
- Camión

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck



Modelo de aprendizaje

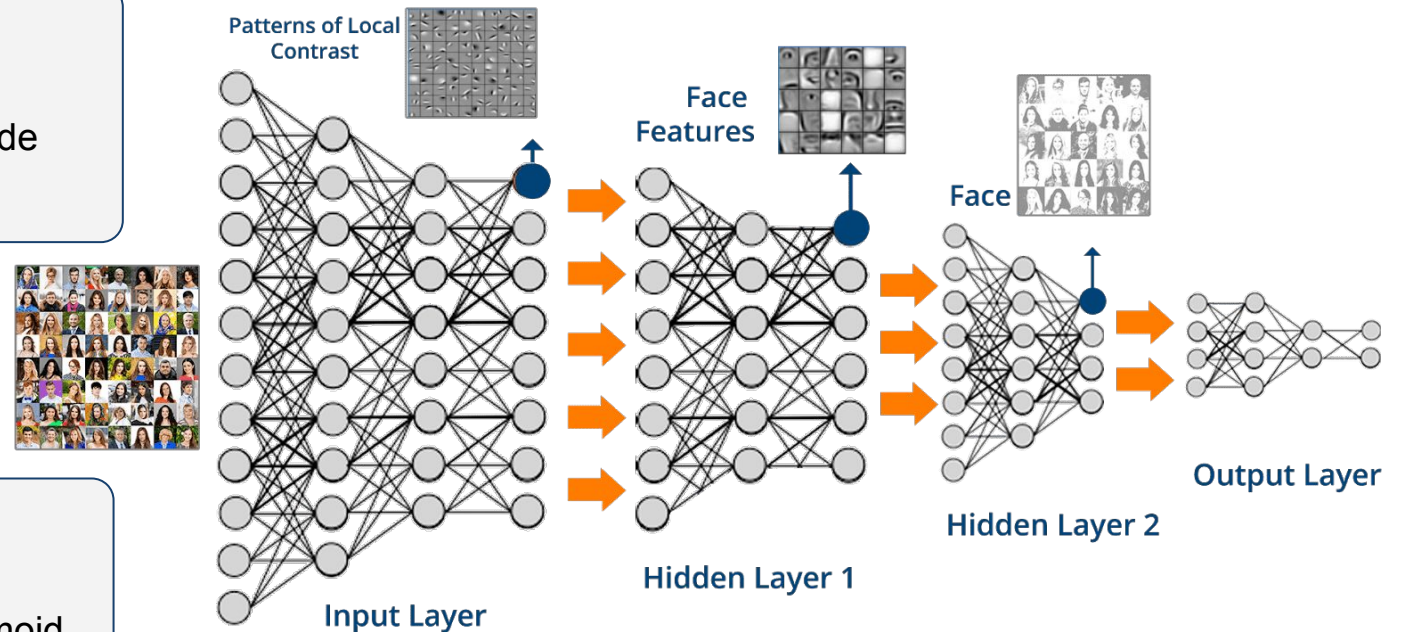
Se ha utilizado un modelo de redes **neuronales convolucionales (CNN)** para realizar la clasificación binaria de imágenes. Las redes neuronales convolucionales son especialmente útiles en tareas de visión por computadora debido a su capacidad para aprender patrones y características visuales en imágenes.

Función de activación: ReLU (Rectified Linear Unit)

Tamaño de entrada esperado: (32, 32, 3) (imágenes de 32x32 píxeles y 3 canales para RGB)

Número de neuronas: 128 en la primera capa densa.

Capa final con 1 neurona y función de activación Sigmoid para la clasificación binaria (automóvil o no automóvil).



Práctica



Permitido

- Deberás modificar el código para obtener una predicción lo más cercana a 1, identificando el 100% de las imágenes de prueba como “coches”.
- ¿Qué tipo de aprendizaje está utilizando el programa?
- Explica cada una de las pruebas que has realizado y las mejoras o no de cada cambio.
- Estudia el código y lo que hace cada una de las funciones.

Una vez conseguido, deberás:

- Desarrollar un nuevo programa que a partir del fichero `.keras` de coches, puedas reconocer las imágenes de los camiones.

¿Y si quieres hacer un programa capaz de reconocer todos los objetos del dataset?

Objetivo: Estudiar el concepto de “*Transferencia del conocimiento*”.




```

import tensorflow as tf
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
import matplotlib.pyplot as plt
import numpy as np
import random

def show_images_with_predictions(images, predictions):
    for i in range(len(images)):
        plt.figure()
        plt.imshow(images[i])
        if predictions[i] >= 0.1:
            plt.title(f"Predicción: Es un coche
({predictions[i][0]:.4f})")
        else:
            plt.title(f"Predicción: No es un coche
({predictions[i][0]:.4f})")
        plt.xlabel("Imagen")
        plt.show()

(train_images, train_labels), (test_images, test_labels) =
cifar10.load_data()

car_indices = np.where(train_labels == 1)[0][:2000]
non_car_indices = np.where(train_labels != 1)[0][:15000]

selected_indices = np.concatenate([car_indices, non_car_indices])
np.random.shuffle(selected_indices)

train_images = train_images[selected_indices]
train_labels = train_labels[selected_indices]

train_images, test_images = train_images / 255.0, test_images / 255.0
train_labels = np.array([1 if label == 1 else 0 for label in
train_labels]) # 1 si es un coche, 0 si no lo es
test_labels = np.array([1 if label == 1 else 0 for label in
test_labels]) # 1 si es un coche, 0 si no lo es

```

```

# Definir el modelo de la red neuronal
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(1, activation='sigmoid') # Capa de salida con activación sigmoid
para obtener valores entre 0 y 1
])

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
model.fit(train_images, train_labels, epochs=1, batch_size=64,
validation_data=(test_images, test_labels))
test_loss, test_accuracy = model.evaluate(test_images, test_labels)
print(f'Precisión en el conjunto de pruebas: {test_accuracy}')

car_indices = [i for i, label in enumerate(test_labels) if label == 1][:5]
car_images = [test_images[i] for i in car_indices]
car_predictions = model.predict(np.array(car_images))
show_images_with_predictions(car_images, car_predictions)

non_car_indices = [i for i, label in enumerate(test_labels) if label != 1]

random_non_car_indices = random.sample(non_car_indices, 5)

non_car_images = [test_images[i] for i in random_non_car_indices]

non_car_predictions = model.predict(np.array(non_car_images))
show_images_with_predictions(non_car_images, non_car_predictions)

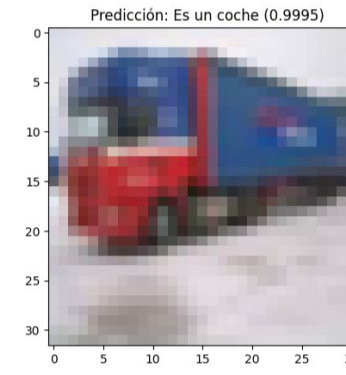
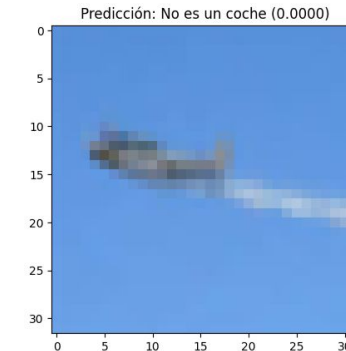
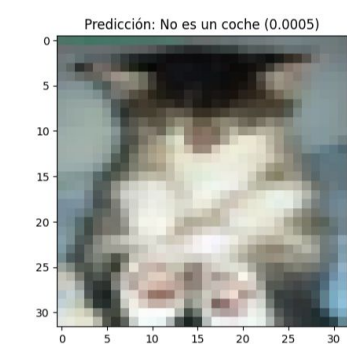
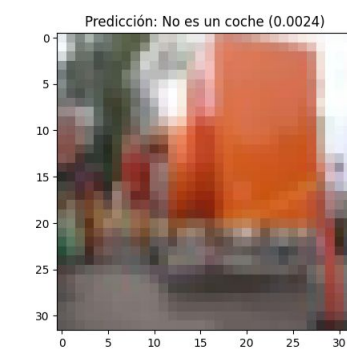
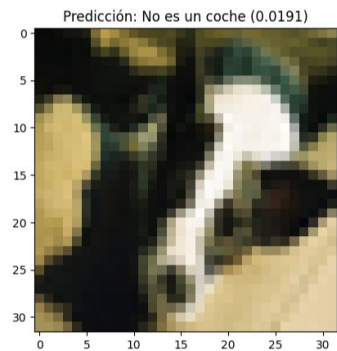
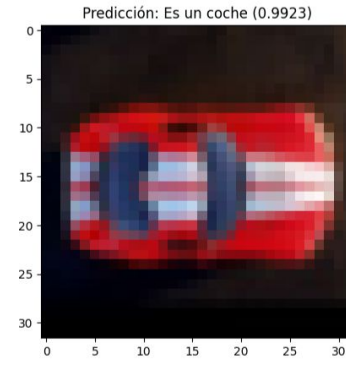
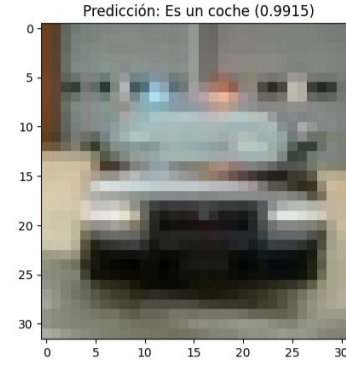
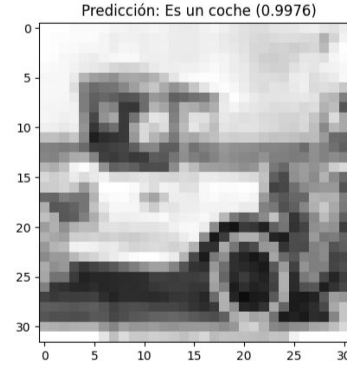
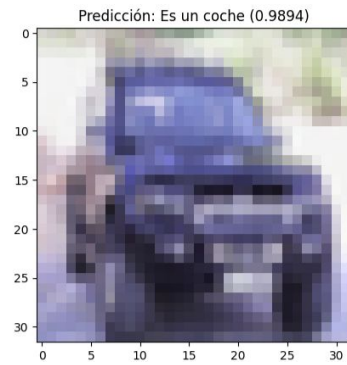
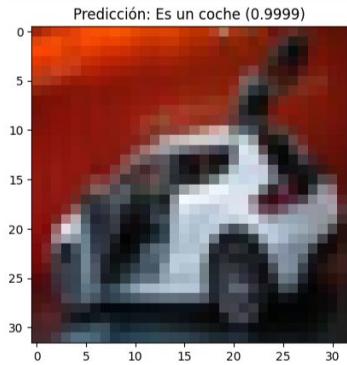
model.save("modelo_entrenado_coche.keras")
print("Modelo guardado exitosamente como 'modelo_entrenado_coche.keras'")

```

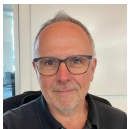
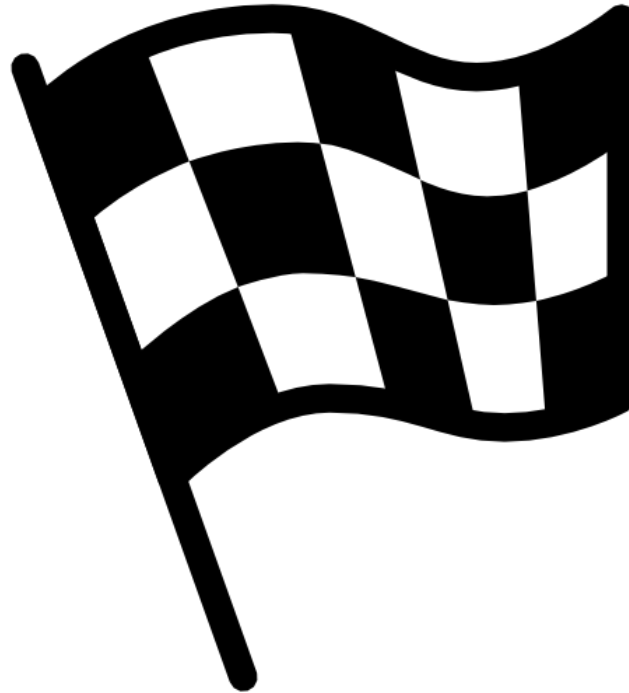


Mis resultados

Precisión en el conjunto de pruebas: 0.9251999855041504



¿Puedes mejorar los resultados? Seguro que sí.



“Antes de trabajar en inteligencia artificial, ¿por qué no hacemos algo sobre la estupidez natural?”

Steve Polyak, fue un neuroanatomista y neurólogo estadounidense

