

# Práctica

## Transfer Learning para clasificación de imágenes

En esta práctica, explorarás el concepto de Transfer Learning utilizando el dataset *CIFAR-10*. Implementarás modelos con tres arquitecturas diferentes preentrenadas, comparando su rendimiento, con el objetivo de aprender a aprovechar modelos preentrenados para resolver problemas de clasificación.

### Descripción del dataset a utilizar

El dataset *CIFAR-10* contiene 60,000 imágenes de 32x32 píxeles distribuidas en 10 clases (por ejemplo, aviones, automóviles, pájaros, gatos, etc.). Este dataset es adecuado para tareas de clasificación de imágenes y es ampliamente utilizado en la investigación de visión por computadora.

### Parte 1: Preparación

1. **Carga del dataset:** Utiliza la función `load_data()` para cargar el dataset *CIFAR-10* (la función está en `tensorflow.keras.datasets.cifar10`).
2. **Preprocesamiento de Datos:** normaliza las imágenes (escalar los valores de píxeles entre 0 y 1) y convierte las etiquetas a formato *one-hot*.

### Parte 2: Implementación y entrenamiento

1. **Arquitecturas:** utilizaremos VGG16, ResNet50 e InceptionV3.
2. **Preparación de las arquitecturas:** carga cada uno de los modelos preentrenados sin las capas superiores (sin la capa de clasificación) y congela todas las capas.
3. **Construcción:**
  - **VGG16:** Usa VGG16 preentrenado en imágenes de *ImageNet*.
  - **ResNet50:** Usa ResNet50 preentrenado.
  - **InceptionV3:** Usa InceptionV3 preentrenado.
4. **Compilación:** Utiliza el optimizador Adam y la función de pérdida `categorical_crossentropy`.
5. **Entrenamiento:** entrena cada modelo con las imágenes del conjunto de entrenamiento y evalúa en el conjunto de test. Ajusta el número de épocas según sea necesario.

### Parte 3: Evaluación

1. **Obtención de métricas:** evalúa los tres modelos utilizando el conjunto de prueba y compara su precisión. Genera las gráficas de pérdida y precisión, así como la matriz de confusión y el informe de clasificación.
2. **Comparación de resultados:** compara las métricas de precisión de los tres modelos. ¿Cuál de los modelos tiene mejor rendimiento? ¿Por qué crees que ocurre esto? ¿Cómo afecta la arquitectura del modelo preentrenado al rendimiento en el dataset *CIFAR-10*? ¿Crees que el modelo con más parámetros siempre es el mejor? Justifica tu respuesta.