

Práctica 5 - Reporte

Cuaderno Movenet.ipynb:

1. ¿Cómo se determina el modelo de red que se usa para hacer la inferencia de postura en este programa?

En este ejemplo, podemos ver en el bloque de código 5 (al lado tiene `In [5]`) que se utiliza el modelo `movenet_lightning` por defecto, sin embargo, más abajo (en el mismo bloque de código) vemos que se encuentra un selector que permite cambiar el modelo a utilizar. Esto se debe a que el modelo `movenet_lightning` es el modelo por defecto en este ejemplo, pero se puede cambiar a otro modelo de la lista de modelos disponibles en el repositorio de TF Hub.

2. ¿Qué procesamiento se le hace a la imagen de entrada para hacer la inferencia? ¿por qué?

La imagen de entrada se preprocesa antes de realizar la inferencia. Este preprocesamiento implica cambiar el tamaño de la imagen a un tamaño estándar, convertirla a escala de grises y normalizar los valores de los píxeles. Esto se hace para reducir la complejidad de la imagen, estandarizar el formato de entrada y mejorar la precisión del modelo.

Cuaderno Transfer-learning-ES.ipynb:

1. ¿En qué consiste un modelo preentrenado?

Un modelo preentrenado es un modelo de aprendizaje automático que se ha entrenado en un gran conjunto de datos para una tarea específica, como la clasificación de imágenes o el procesamiento del lenguaje natural, y que ya ha aprendido a reconocer determinados patrones o características en los datos.

2. Sintetice en qué consiste el aprendizaje por transferencia.

El aprendizaje por transferencia consiste en utilizar un modelo preentrenado como punto de partida para una nueva tarea en lugar de entrenar un nuevo modelo desde cero. El modelo preentrenado suele estar entrenado en un conjunto de datos amplio y diverso, por lo que ya ha aprendido características útiles que pueden aplicarse a la nueva tarea. El modelo preentrenado se modifica añadiendo nuevas capas sobre él, y estas nuevas capas se entrenan en un conjunto de datos más pequeño y específico para la nueva tarea.

3. Indique las maneras en que se realiza el aprendizaje por transferencia en el cuaderno.

El aprendizaje por transferencia se realiza en el código cargando un modelo preentrenado (MobileNetV2) mediante la API Keras y añadiendo una nueva capa de clasificación sobre él. A continuación, el modelo preentrenado se congela y sólo se entrena la nueva capa de clasificación en un conjunto de datos más pequeño.

4. Explique por qué se usa el aumento de datos en el ejercicio y en qué consiste.

El aumento de datos se utiliza para aumentar el tamaño y la diversidad del conjunto de datos de entrenamiento. Consiste en aplicar transformaciones aleatorias a las imágenes, como rotación, volteo o zoom,

para crear nuevas variaciones de la misma imagen. Esto ayuda a evitar el sobreajuste y mejora la generalización del modelo.

5. ¿Qué procesamiento se le hace a las imágenes de entrada para alimentarlas al modelo? ¿por qué?

Las imágenes de entrada se preprocesan redimensionándolas a un tamaño fijo de 224x224 píxeles y normalizando los valores de los píxeles para que estén en el intervalo $[-1, 1]$. Esto se hace porque el modelo preentrenado (MobileNetV2) espera imágenes de entrada de este tamaño y también está entrenado con esta normalización.

6. ¿De qué manera se construye el modelo de clasificación encima del modelo preentrenado?

El modelo de clasificación se construye sobre el modelo MobileNetV2 preentrenado añadiendo una nueva capa de clasificación con dos nodos de salida (para las dos clases del conjunto de datos) y utilizando el modelo preentrenado como extractor de características. Las imágenes de entrada se introducen en el modelo preentrenado, y la salida de la última capa convolucional se aplanan y pasa por la nueva capa de clasificación para obtener las predicciones finales.

Esto lo vemos en los fragmentos de código 19 y 20, respectivamente:

```
global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
feature_batch_average = global_average_layer(feature_batch)
print(feature_batch_average.shape)
```

```
prediction_layer = tf.keras.layers.Dense(1)
prediction_batch = prediction_layer(feature_batch_average)
print(prediction_batch.shape)
```

7. ¿Cuántos parámetros se necesitan entrenar en este nuevo modelo?

Sólo es necesario entrenar los parámetros de la nueva capa de clasificación en este nuevo modelo, ya que los parámetros del modelo MobileNetV2 preentrenado están congelados y no se actualizan durante el entrenamiento. En total son **1,281** parámetros. En el refinamiento son un total de **1,862,721** parámetros entrenables.

8. ¿En qué consiste el proceso de refinamiento de este nuevo modelo?

El proceso de refinamiento de este nuevo modelo consta de dos etapas: en primer lugar, la nueva capa de clasificación se entrena en el conjunto de datos de entrenamiento durante un número fijo de épocas, mientras que el modelo MobileNetV2 preentrenado se congela; a continuación, todo el modelo se descongela y se ajusta en el conjunto de datos de entrenamiento durante otro número fijo de épocas, con una tasa de aprendizaje menor que en la primera etapa. Esto permite al modelo ajustar los pesos preentrenados y aprender características específicas de la tarea a partir de un conjunto de datos más pequeño.