

Red convolucional creada por usuario V.S transfer learning

Alvaro Pequeño Mondragón

20 de Febrero de 2025

Introducción

La clase pasada se vio un repaso de lo que son las redes convolucionales y su funcionamiento dentro de la clasificación de imágenes. Una vez hecho el repaso fue explicada una técnica en la cual se puede usar un modelo ya entrenado como base para poder entrenarlo con nuestro conjunto de datos que puede variar del conjunto con el que se entrenó este modelo base. Esta técnica se conoce como transfer learning y nos permite usar estos modelos entrenados con grandes cantidades de datos y aplicarlos a nuestro set de datos con la ventaja de que al tener una parte pre-entrenada nuestro conjunto de datos ya no necesita ser tan grande para entrenar de buena manera el modelo.

Objetivo

El objetivo de esta tarea es comparar el desempeño de una red con una arquitectura creada por nosotros contra el desempeño usando la técnica de transfer learning. Se usará el mismo conjunto de datos para ambos casos y se usaran los mismos datos de entrenamiento y prueba.

Resultados

Los modelos se entrenaron con imágenes de 3 animales diferentes: Mariposa, Gato y Caballo.

A continuación se muestra el desempeño de la red convolucional creada por nosotros:

```

Epoch 1/10
24/24 ————— 3s 77ms/step - accuracy: 0.4783 - loss: 2.3444 - val_accuracy: 0.5000 - val_loss: 0.9236
Epoch 2/10
24/24 ————— 2s 72ms/step - accuracy: 0.4825 - loss: 1.8553 - val_accuracy: 0.5000 - val_loss: 1.5776
Epoch 3/10
24/24 ————— 2s 71ms/step - accuracy: 0.5979 - loss: 0.9125 - val_accuracy: 0.6333 - val_loss: 0.8834
Epoch 4/10
24/24 ————— 2s 71ms/step - accuracy: 0.6166 - loss: 0.6873 - val_accuracy: 0.6667 - val_loss: 1.1184
Epoch 5/10
24/24 ————— 2s 71ms/step - accuracy: 0.7197 - loss: 0.5606 - val_accuracy: 0.7667 - val_loss: 0.8689
Epoch 6/10
24/24 ————— 2s 69ms/step - accuracy: 0.8878 - loss: 0.4208 - val_accuracy: 0.8000 - val_loss: 0.8255
Epoch 7/10
24/24 ————— 2s 69ms/step - accuracy: 0.8851 - loss: 0.3422 - val_accuracy: 0.8000 - val_loss: 0.7838
Epoch 8/10
24/24 ————— 2s 71ms/step - accuracy: 0.9007 - loss: 0.2859 - val_accuracy: 0.8000 - val_loss: 0.6659
Epoch 9/10
24/24 ————— 2s 69ms/step - accuracy: 0.9038 - loss: 0.2394 - val_accuracy: 0.7667 - val_loss: 0.5826
Epoch 10/10
24/24 ————— 2s 69ms/step - accuracy: 0.9063 - loss: 0.2377 - val_accuracy: 0.7000 - val_loss: 0.5597
<keras.src.callbacks.history.History at 0x124dd25b860>

```

Después observamos el desempeño de la tecnica de transfer learning:

```

Epoch 1/10
24/24 ————— 3s 96ms/step - accuracy: 0.5626 - loss: 1.0719 - val_accuracy: 0.6333 - val_loss: 1.0201
Epoch 2/10
24/24 ————— 2s 80ms/step - accuracy: 0.8260 - loss: 0.9287 - val_accuracy: 0.8667 - val_loss: 0.8436
Epoch 3/10
24/24 ————— 2s 82ms/step - accuracy: 0.8762 - loss: 0.6703 - val_accuracy: 0.8667 - val_loss: 0.5731
Epoch 4/10
24/24 ————— 2s 81ms/step - accuracy: 0.9306 - loss: 0.3747 - val_accuracy: 0.9000 - val_loss: 0.3776
Epoch 5/10
24/24 ————— 2s 83ms/step - accuracy: 0.9891 - loss: 0.2110 - val_accuracy: 0.9667 - val_loss: 0.2624
Epoch 6/10
24/24 ————— 2s 83ms/step - accuracy: 1.0000 - loss: 0.1211 - val_accuracy: 0.9667 - val_loss: 0.2019
Epoch 7/10
24/24 ————— 2s 81ms/step - accuracy: 1.0000 - loss: 0.0734 - val_accuracy: 0.9667 - val_loss: 0.1704
Epoch 8/10
24/24 ————— 2s 81ms/step - accuracy: 1.0000 - loss: 0.0466 - val_accuracy: 0.9667 - val_loss: 0.1542
Epoch 9/10
24/24 ————— 2s 82ms/step - accuracy: 1.0000 - loss: 0.0312 - val_accuracy: 0.9667 - val_loss: 0.1454
Epoch 10/10
24/24 ————— 2s 81ms/step - accuracy: 1.0000 - loss: 0.0217 - val_accuracy: 0.9667 - val_loss: 0.1401

```

Conclusiones

Viendo los resultados que muestran los modelos podemos concluir que usar algún modelo base nos permite poder mejorar la precisión de nuestro modelo que en este caso es para clasificar imágenes. Nuestro conjunto de datos se componía de 150 fotos donde hay 50 por cada animal que tenemos.

El modelo de transfer learning tiene las mismas capas que en el modelo creado por nosotros con la diferencia de que en transfer learning antes de estas capas se encuentra el modelo base pre-entrenado.