Práctica Clasificación de animales

Miguel Gómez Prieto

Clasificación de un dataset de animales para una competición en Kaggle

---

Análisis del Dataset

El dataset contiene imágenes de **10 clases de animales**. Durante el análisis inicial se observó:

- **Número de clases**: 10 categorías balanceadas de forma desigual.
- **Número de muestras**: variable por clase.
- **Dimensiones de las imágenes**: no homogéneas, lo que obligó a aplicar un preprocesamiento de redimensionado.

Se aplicaron las siguientes técnicas de preprocesamiento:

- Redimensionado de todas las imágenes a **128x128 píxeles**.
- Normalización de valores de píxeles en el rango [0,1].
- Aumento de datos (*data augmentation*) con rotaciones, espejado horizontal y zoom para mejorar la generalización.

Modelos Probados

Modelo Grande (CNN desde cero)

Se diseñó una red convolucional profunda con varias capas `Conv2D`, `BatchNormalization`, `MaxPooling` y capas densas al final.

Este modelo fue el primero en ofrecer resultados aceptables, aunque requirió **muchas épocas de entrenamiento** debido a la complejidad del problema.

Se diseñó siguiendo principios básicos de visión por computador: capas convolucionales iniciales con pocos filtros para captar bordes y texturas simples, aumentando progresivamente la profundidad para extraer patrones más complejos.

Se incluyeron capas de BatchNormalization y MaxPooling para estabilizar y reducir dimensionalidad.

Finalmente se incluyeron capas densas con Dropout para combinar características e intentar disminuir el sobreajuste. Esta arquitectura buscaba un equilibrio entre capacidad de representación y generalización.

Resultados principales:

- Alcanzó una **accuracy de validación del 96.5%** tras 120 epochs.
- El entrenamiento fue más lento y con bastante sobreajuste, sobretodo en las últimas epochs.

Oversampling

Se probó el mismo modelo aplicando técnicas de balanceo de clases mediante oversampling, cambiando el imapcto de la función de loss según cuantas muestras tiene la clase.

- **Resultado:** no se obtuvieron mejoras significativas respecto al modelo base.

Transfer Learning (VGG16 + ImageNet)

Se aplicó *transfer learning* utilizando **VGG16** con pesos preentrenados en ImageNet.

- Se congelaron la mayoría de capas, excepto el último bloque convolucional.
- Se redujo la tasa de aprendizaje a **1e-4** y posteriormente a **1e-5**.

Para VGG, las imagenes se tenían que preprocesar, según indica VGG (sencillo con la función de Keras)

Resultados principales:

- Accuracy de validación del **94.3%** en 20 épocas.
- Con fine-tuning y menor *learning rate*, se alcanzó un **95.7%**.
- Mejor generalización y menor tiempo de entrenamiento que el modelo desde cero.

Ensemble

Se combinaron los resultados de:

- El modelo grande entrenado desde cero.
- Los dos modelos basados en VGG16.

Se utilizó un **clasificador por votación**.

- **Resultado:** el ensemble no superó al mejor modelo individual (VGG16 ajustado), pero sí se convirtió en la **segunda mejor entrega**.

Evaluación y Conclusiones

- El **modelo a mano** demostró que es posible entrenar desde cero, pero requiere más recursos y tiempo y aun así se alcanzan peores resultados.
- El **transfer learning con VGG16** fue la mejor estrategia. Al utilizar una arquitectura y pesos comprobados, alcanza mayor precisión con menos esfuerzo computacional.
- El **ensemble** aportó robustez, aunque no mejoró la métrica final.