

# Informe Técnico – Red Neuronal Convolutiva para Clasificación de Imágenes (CIFAR-10)

Este proyecto aplica una red neuronal convolutiva (CNN) para el reconocimiento de imágenes, analizando su rendimiento, precisión y los desafíos del sobreajuste en conjuntos de datos complejos.

Por Daniel Serrano y  
Alexander Arrosquipa

Contenido

Introducción ..... 2

Arquitectura del modelo ..... 2

    Fases ..... 2

Resultados y análisis..... 3

Conclusiones ..... 3

## Introducción

En este proyecto hemos desarrollado una red neuronal convolucional (CNN) capaz de clasificar imágenes del conjunto de datos CIFAR-10, compuesto por 60.000 imágenes a color.

El problema de clasificación en CIFAR-10 es que el modelo debe reconocer y asignar cada imagen a una de las 10 categorías.

Nuestro objetivo es que la red neuronal aprenda a distinguir correctamente los objetos en imágenes pequeñas (32x32 píxeles en color), a pesar de tener fondos variados, diferentes ángulos, tamaños y niveles de iluminación, y para ello lo más adecuado es una CNN ya que imitan la forma en que el córtex visual humano procesa la información visual, aprendiendo de manera automática características como bordes, texturas y formas.

## Arquitectura del modelo

El modelo se compone de dos bloques convolucionales seguidos de un clasificador denso. Se implementó con las bibliotecas TensorFlow / Keras en Python.

### Fases

#### *Fase 1 – Preparación del Dataset (CIFAR-10)*

- Normalización de píxeles al rango [0,1] para mejorar la estabilidad numérica y acelerar la convergencia del optimizador.
- Codificación one-hot de las etiquetas mediante `to_categorical()`. Esto convierte etiquetas a vectores binarios compatibles con `categorical_crossentropy`.
- Visualización exploratoria de muestras por clase para verificar equilibrio y variedad visual.

La estructura tridimensional de las imágenes se mantuvo sin aplanar, ya que las CNN requieren la relación espacial entre píxeles para extraer patrones visuales.

#### *Fase 2 – Construcción del Modelo CNN*

Se implementó un modelo secuencial en Keras con dos bloques convolucionales y un clasificador denso.

Capas principales:

1. Conv2D(32, 3x3, ReLU) + MaxPooling(2x2)

→ Detecta bordes y reduce la dimensionalidad espacial

2. Conv2D(64, 3x3, ReLU) + MaxPooling(2x2)

→ Aprende combinaciones más complejas de texturas y formas

3. Flatten() → Dense(64, ReLU) → Dense(10, Softmax)

→ Clasificación final multiclase

El modelo tiene 167.562 parámetros para entrenar.

Durante el entrenamiento los pesos se ajustan mediante descenso del gradiente y retropropagación, minimizando la entropía cruzada.

Las capas convolucionales actúan como extractores automáticos de características, sustituyendo el preprocesamiento manual tradicional.

### Fase 3 – Entrenamiento y Evaluación

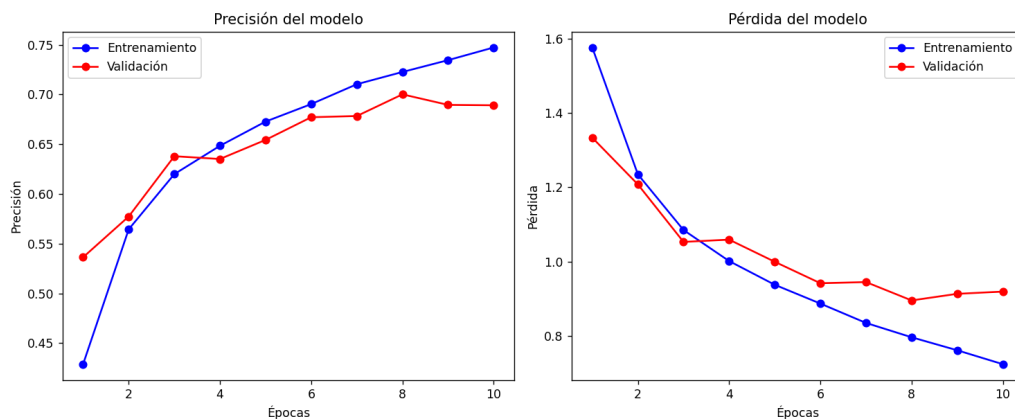
El modelo se entrenó durante 10 épocas, con batch size = 64 y validation\_split = 0.1.

El optimizador Adam ajustó dinámicamente la tasa de aprendizaje, garantizando una convergencia estable.

Métrica	Valor
Precisión entrenamiento	0.76
Precisión Validación	0.70
Precisión test	0.70
Pérdida test	0.85

## Resultados y análisis

Las curvas de pérdida y precisión muestran una convergencia estable y sin signos de sobreajuste. Por lo que podemos decir que la CNN generaliza correctamente las características visuales de las imágenes. Aunque podemos observar un ligero aprendizaje después de la 8ª época.



El nuevo conjunto (CIFAR-10) es más complejo y variado que MNIST; las imágenes tienen más ruido, diferentes fondos, formas y colores, lo que hace que el modelo tienda a memorizar detalles en lugar de generalizar bien.

## Conclusiones

El modelo CNN desarrollado logra una precisión del 70% sobre el conjunto de prueba, demostrando la eficacia de las redes convolucionales para tareas de reconocimiento de imágenes.

Ademas este proyecto nos ha hecho comprender:

- Cómo el procesamiento adecuado (normalización y one-hot encoding) influye en la estabilidad del entrenamiento.
- El rol de las convoluciones y del pooling en la extracción jerárquica de características visuales.
- Cómo el descenso del gradiente y Adam ajustan los pesos de la red para minimizar la entropía cruzada.