Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Querétaro

Momento de Retroalimentación: Módulo 2 Implementación de un modelo de deep learning.

(Portafolio Implementación)

Alumno:

Tomás Pérez Vera – A01028008

TC3007C.501

Inteligencia Artificial Avanzada Para la Ciencia de Datos II

Fecha:

8 de noviembre de 2025

Resumen

El presente trabajo, pretende realizar un análisis comparativo de dos arquitecturas de redes neuronales convolucionales (CNN) aplicadas a la clasificación binaria de audio, con el fin de realizar la distinción entre sonidos de gatos y perros. El conjunto de datos empleado consiste en grabaciones en formato WAV, recortadas a una duración máxima de un segundo y muestreadas a 16 kHz, balanceadas entre ambas clases. El primer modelo utiliza convoluciones en una dimensión sobre las formas de onda crudas, mientras que el segundo emplea representaciones espectrales mediante Coeficientes Cepstrales en la Frecuencia Mel (MFCC) combinados con convoluciones bidimensionales. Los resultados muestran que el modelo basado en MFCC alcanzó una mayor precisión de clasificación y una convergencia más rápida, demostrando la efectividad de las características del dominio frecuencial para representar información acústica relevante en tareas de clasificación de audio.

INTRODUCCIÓN

En el campo del aprendizaje profundo, las redes neuronales convolucionales (CNN) se han consolidado como una de las arquitecturas más utilizadas, especialmente en tareas de visión computacional. Sin embargo, aunque su popularidad moderna se asocia al procesamiento de imágenes, la operación fundamental que les da nombre —la convolución— se originó en el ámbito del procesamiento digital de señales.

Matemáticamente, la convolución entre una señal de entrada $x(t)$ y un filtro $h(t)$ se define como:

$$(y * t)(t) = \int_{-\infty}^{\infty} c(\tau)h(t-\tau)\,d\tau$$

o, en su versión discreta dentro del ámbito de sistemas digitales:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Esta operación mide la similitud local entre una señal y un patrón (kernel), permitiendo detectar transiciones, bordes, repeticiones, componentes de frecuencia y otras características estructurales. Dentro de sus principales aplicaciones, destacan: filtrado, suavizado, detección de picos, análisis espectral y reconocimiento de patrones en señales unidimensionales.

Dentro del ámbito del aprendizaje profundo, la idea de la convolución fue reinterpretada, en torno a su aplicación dentro de un marco neuronal, utilizando filtros aprendibles en lugar de filtros diseñados manualmente. Su aplicación a imágenes se volvió ampliamente exitosa ya que las CNN son capaces de detectar bordes, texturas y formas mediante correlaciones locales en dos dimensiones. Por ello, en la práctica moderna, la convolución se asocia principalmente al procesamiento de imágenes, aunque su base teórica proviene directamente del análisis de señales.

Dado este antecedente, resulta natural la aplicación de convoluciones al procesamiento de audio. Las ondas sonoras son señales temporales que contienen patrones locales, variaciones de energía, transiciones abruptas, y periodicidades que pueden capturarse de manera eficaz mediante convoluciones unidimensionales. De forma alternativa, al transformar el audio al dominio frecuencial mediante métodos como los Coeficientes Cepstrales en la Frecuencia Mel (MFCC), el problema se convierte en una estructura bidimensional similar a una imagen, lo que posibilita el uso de CNN 2D para aprender patrones espectrales.

En este trabajo se evalúan dos enfoques ampliamente utilizados en clasificación de audio:

1. CNN unidimensional aplicada a la forma de onda cruda, donde la convolución opera directamente sobre la señal temporal, aprovechando su naturaleza original como dato de una dimensión.
2. CNN bidimensional aplicada a MFCC, donde el audio se representa mediante un mapa tiempo–frecuencia que permite a los filtros bidimensionales capturar distribuciones energéticas y patrones armónicos característicos.

El objetivo es comparar la efectividad de ambas representaciones para la clasificación binaria entre sonidos de gatos y perros, analizando cómo la naturaleza matemática de la convolución y la estructura del dato influyen en el rendimiento y la capacidad de generalización de cada modelo.

DESCRIPCIÓN DEL CONJUNTO DE DATOS

El conjunto de datos utilizado corresponde a una colección de grabaciones de audio disponible en la plataforma **Kaggle**, particularmente en la competencia de *Audio Cats and Dogs*. Este dataset tiene como propósito evaluar la capacidad de los modelos de aprendizaje profundo para distinguir entre sonidos de gatos y perros.

El conjunto está conformado por archivos en formato WAV con una frecuencia de muestreo uniforme de 16 kHz y duraciones variables. Cada clase presenta un número diferente de muestras:

- Clase gato: 164 archivos WAV, equivalentes a 1,323 segundos de audio.
- Clase perro: 113 archivos WAV, equivalentes a 598 segundos de audio.

Las grabaciones provienen de la base AE-Dataset (Acoustic Events Dataset) y fueron extraídas originalmente de la plataforma FreeSound, una comunidad colaborativa de sonidos abiertos.

Para el presente trabajo, los audios fueron normalizados en amplitud, convertidos a una duración estándar de un segundo y segmentados en tres subconjuntos:

- Entrenamiento (80%)

- Validación (10%)
- Prueba (10%)

El objetivo de este preprocesamiento fue garantizar un equilibrio entre clases y homogeneidad en la longitud de las señales, permitiendo un entrenamiento más estable y una comparación justa entre los modelos.

PREPROCESAMIENTO DE DATOS

El preprocesamiento aplicado a las grabaciones de audio constituye una etapa esencial para garantizar que los modelos convolucionales reciban señales uniformes, comparables y adecuadamente representadas. En ambos enfoques se aplicó un conjunto de pasos comunes de normalización temporal y espectral que aseguran que cada muestra de audio tenga la misma estructura dimensional y una calidad homogénea para el entrenamiento.

Inicialmente, se hace la lectura de los archivos .wav utilizando la librería *SoundFile*. Dado que las señales pueden tener uno o dos canales; por ello, en caso de encontrarse un archivo estéreo, se convierte a mono mediante la media aritmética de los canales.

Matemáticamente, si el audio posee dos canales $x_L(t)$ y $x_R(t)$, la señal mono se puede obtener mediante:

$$x_{mono} = \frac{1}{2}(x_L(t) + x_R(t))$$

Esto garantiza que la información esté distribuida de forma uniforme sin introducir sesgos por diferencias entre canales.

Posteriormente, todas las señales son remuestreadas a una frecuencia estándar de 16 kHz. Debido a que los audios pueden provenir de diferentes dispositivos o fuentes con tasas de muestreo distintas, se aplica un resampling mediante interpolación y filtrado anti–aliasing. El proceso de remuestreo consiste en transformar una señal registrada a frecuencia f_s hacia una nueva frecuencia f'_s generando una representación temporal consistente:

$$x'(t) = Resample(x(t), f_s \rightarrow f'_s)$$

Esto asegura que todos los audios tengan la misma densidad temporal y para prevenir distorsiones en las convoluciones posteriores.

Una vez remuestreadas las señales, se procede a estandarizar su longitud. Dado que los sonidos presentan duraciones variables, cada señal es ajustada a una duración máxima de un segundo. Esto equivale a un vector de longitud fija de $N = 160000\ muestras$.

Si una señal de audio dentro del conjunto de datos excede el tamaño de muestras, esta se recorta:

$$x_{trim} = x[0:N]$$

En cambio, si la señal de audio es más corta, se aplica padding con ceros al final de la forma de onda:

$$x_{pdding} = \begin{cases} [x,0,0,0,\ldots 0]\ si\ |x| < N \\ x[0:N]\ |x| > N \end{cases}$$

Este proceso permite que las señales presenten la misma dimensión, de tal

forma que puedan ser procesadas por lotes dentro de redes neuronales.

Tras la estandarización de longitud y frecuencia, se aplica normalización estadística con el objetivo de estabilizar el entrenamiento y eliminar dependencias relativas a la intensidad del sonido original, mediante:

$$\tilde{x} = \frac{x - \mu}{\sigma - \epsilon}$$

donde μ es la media de la señal; σ es su desviación estándar; y $\epsilon = 10^{-6}$ un valor pequeño para evitar divisiones entre cero. Ello permite que el comportamiento de cada señal sea semejante al de una distribución estándar, lo que resulta en optimizar la convergencia del modelo.

Ante este punto, esto ha sido el preprocesamiento que se le ha hecho al conjunto de datos para alimentar ambos modelos, sin embargo, hay ciertas condiciones de los datos que fueron ajustadas para cada modelo.

Para el modelo CNN unidimensional, la representación final es simplemente la forma de onda normalizada $\tilde{x}(t)$, que se mantiene como un vector unidimensional para ser procesado mediante convoluciones unidimensionales.

Por otro lado, en el modelo CNN bidimensional, se transforma la señal normalizada en un conjunto de coeficientes MFCC. Este proceso implica primero una representación espectral mediante la transformada corta de Fourier (STFT):

$$X(\tau, \omega) = \sum_{n=0}^{N-1} x[n + \tau H]\, w[n] e^{-j\omega n}$$

donde $w[n]$ es una ventana, H es el *hop length* y τ indexa cada cuadro temporal. Luego, la energía espectral se proyecta sobre la escala de Mel mediante bancos de filtros triangulares, obteniendo:

$$M(K, \tau) = \sum_{\omega} |X(\tau, \omega)|^2 H_k(\omega)$$

Finalmente, los MFCC se calculan aplicando una transformada discreta del coseno (DCT):

$$c_m(\tau) = \sum_{k=1}^{K} \log\big(M(k, \tau)\big) \cos\left[\frac{\pi m}{K}(k - \frac{1}{2})\right]$$

Donde el resultado es una matriz bidimensional que captura información acústica compacta y perceptualmente relevante.

METODOLOGÍA DE TRABAJO

Modelo 1 – CNN 1D

El primer modelo emplea convoluciones unidimensionales directamente sobre las formas de onda crudas. Esta arquitectura busca aprender patrones temporales locales, tales como cambios súbitos en la amplitud, onsets y texturas acústicas, sin realizar una transformación previa al dominio frecuencial.

Arquitectura propuesta:

1. Entrada
    - Onda de audio en formato 1D con una sola canalización y longitud fija de 16 000 muestras

(equivalente a 1 segundo a 16 kHz).

2. Bloque Convolucional 1
    - Conv1D: 1 → 16 canales, *kernel* = 5, *stride* = 2, *padding* = 2.
    - Activación: ReLU.
3. Bloque Convolucional 2
    - Conv1D: 16 → 32 canales, *kernel* = 5, *stride* = 2, *padding* = 2.
    - Activación: ReLU.
4. Bloque Convolucional 3
    - Conv1D: 32 → 64 canales, *kernel* = 5, *stride* = 2, *padding* = 2.
    - Activación: ReLU.
5. Capa de Reducción Global
    - AdaptiveAvgPool1d(1): promedia cada mapa de características a una sola muestra.
6. Aplanamiento
    - Flatten: convierte el tensor a un vector de tamaño 64.
7. Capas Densas
    - Linear: 64 → 32 unidades.
    - Activación: ReLU.
8. Capa de Salida
    - Linear: 32 → 2 unidades (correspondiente a las dos clases: gato y perro).
    - Función de activación aplicada durante la inferencia mediante *softmax* implícito en la pérdida CrossEntropy**.**

Este modelo aprovecha la capacidad de la convolución 1D para identificar patrones locales en señales temporales. A través de sucesivos niveles de abstracción, las capas extraen características acústicas de bajo nivel (microvariaciones temporales) que se combinan en representaciones más globales útiles para la clasificación binaria.

Modelo 2 – CNN 2D (MFCCs)

En este modelo, la señal se transforma primero al dominio frecuencial mediante la extracción de 40 coeficientes MFCC por ventana, generando un mapa tiempo–frecuencia similar a una imagen. Este mapa es procesado mediante convoluciones bidimensionales, aprovechando la capacidad de la CNN 2D para capturar patrones espaciales correlacionados.

Arquitectura propuesta:

1. Entrada
    - Matriz MFCC de tamaño aproximadamente (40 × T), donde *T* corresponde al número de ventanas temporales después del proceso de extracción (≈ 100–110 para 1 s a 16 kHz con hop de 160).
    - La entrada se trata como un tensor 2D con un canal: (1 × 40 × T).
2. Bloque Convolucional 1
    - Conv2D: 1 → 16 canales, *kernel* = (3, 3), *padding* = 1.

- Batch Normalization (BN2D).
- Activación: ReLU.
- MaxPooling2D: *pool* = (2, 2).

3. Bloque Convolucional 2
   - Conv2D: 16 → 32 canales, *kernel* = (3, 3), *padding* = 1.
   - Batch Normalization.
   - Activación: ReLU.
   - MaxPooling2D: *pool* = (2, 2).
4. Bloque Convolucional 3
   - Conv2D: 32 → 64 canales, *kernel* = (3, 3), *padding* = 1.
   - Batch Normalization.
   - Activación: ReLU.
   - AdaptiveAvgPool2D: salida fija de tamaño (1 × 1) por canal, independientemente de *T*.
5. Capa de Aplanamiento (Flatten)
   - Convierte el tensor resultante (64 × 1 × 1) en un vector de 64 características.
6. Capas Densas
   - Linear: 64 → 128 unidades.
   - Activación: ReLU.
   - Dropout: 0.3.
7. Capa de Salida
   - Linear: 128 → *num_classes* (2 unidades para clasificación binaria).
   - La activación *softmax* está implícita al usar *CrossEntropyLoss* durante el entrenamiento.

Al trabajar sobre MFCC, el modelo recibe una representación compacta del espectro de energía perceptualmente relevante. Las CNN 2D capturan correlaciones tanto en el eje temporal como en el frecuencial, permitiendo identificar texturas espectrales asociadas a patrones vocales típicos de gatos y perros, lo que favorece un aprendizaje más rápido y robusto.

A continuación, se muestran los Coeficientes Cepstrales en la Frecuencia Mel para audios de gato y perro:

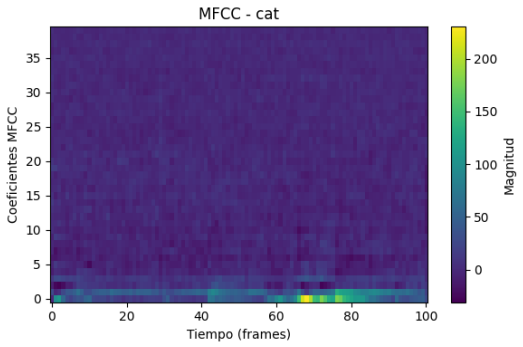

Figura 1. MFCC de gato

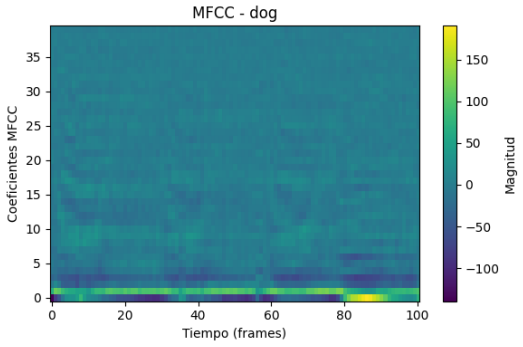

Figura 2. MFCC de perro

RESULTADOS EXPERIMENTALES

Para la evaluación del desempeño de ambos modelos se empleó la métrica *accuracy*, debido a que la tarea consiste en una clasificación binaria con clases balanceadas, por lo que la proporción de predicciones correctas resulta una medida directa y adecuada del rendimiento global. Asimismo, se utilizó *Binary Cross Entropy* como función de pérdida, ya que es la más apropiada para problemas de clasificación binaria al cuantificar la discrepancia entre las probabilidades predichas y las etiquetas reales, penalizando con mayor intensidad las predicciones con alta confianza, pero incorrectas y favoreciendo una convergencia estable durante el entrenamiento.

Ambos modelos fueron entrenados bajo las mismas condiciones:

- Optimizador: Adam (lr = 1e-3)
- Función de pérdida: CrossEntropyLoss
- Tamaño de lote: 8
- Número de épocas: 150 para el modelo 2, 150 para el modelo 1.
- Tasa de muestreo: 16 kHz
- Duración máxima: 1 segundo

Tabla 1. Comparativa de Métricas entre Modelos

Métrica	Modelo 1 – CNN 1D	Modelo 2 – MFCC + CNN 2D
Precisión del modelo	0.9104	0.8602
Pérdida de validación mínima	0.46–0.50	≈ 0.38–0.40
Épocas totales	150	150

Los resultados preliminares muestran diferencias relevantes entre ambas arquitecturas. El Modelo 1 (CNN 1D) obtuvo una precisión ligeramente superior, alcanzando un 91.04 %, mientras que el Modelo 2 (MFCC + CNN 2D) registró un 86.02 %. Sin embargo, el segundo modelo presentó una *pérdida de validación mínima* más baja (≈0.38–0.40 frente a 0.46–0.50), lo que sugiere una mejor estabilidad y una convergencia más consistente durante el entrenamiento. A pesar de que ambos modelos fueron entrenados con el mismo número de épocas, los resultados indican que el enfoque basado en MFCC logra representar de manera más compacta y eficiente la información acústica, mientras que la CNN 1D logra un mejor desempeño final en términos de precisión directa sobre las predicciones.

GRAFICAS OBTENIDAS

Modelo 1 – CNN 1D

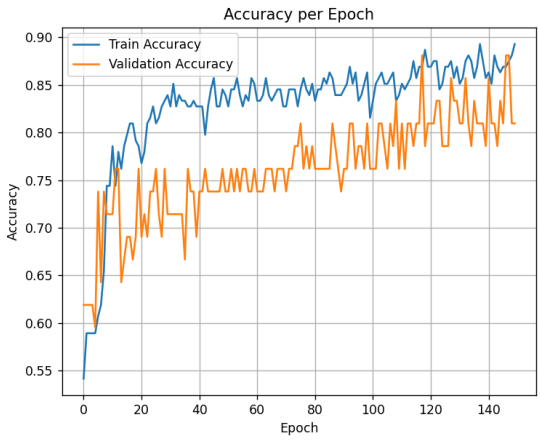

Figura 3. Certeza por época Modelo CNN 1D

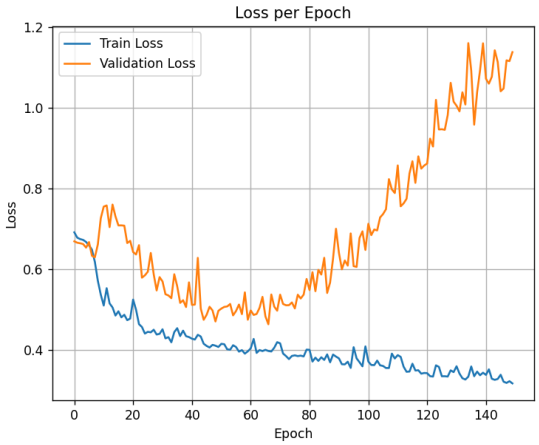

Figura 4. Pérdida por época Modelo CNN 1D

Modelo 2 – CNN 2D (MFCCs)

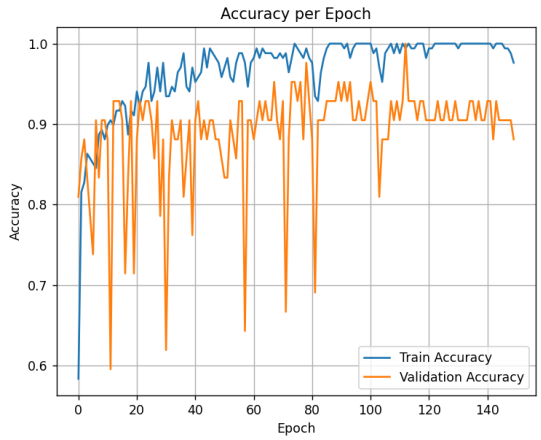

Figura 5. Certeza por época Modelo CNN 2D

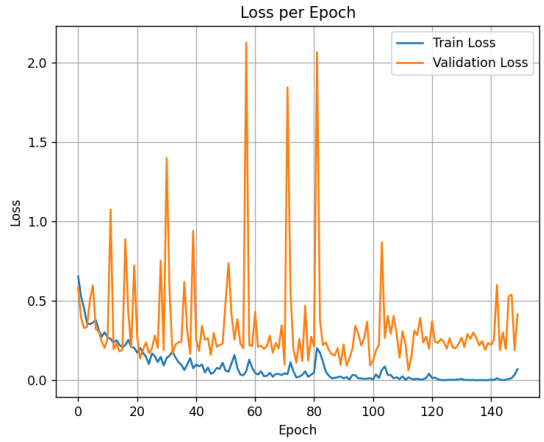

Figura 6. Pérdida por época Modelo CNN 2D

ANÁLISIS DE RESULTADOS

El análisis comparativo de ambas arquitecturas evidencia diferencias relevantes tanto en el comportamiento durante el entrenamiento como en su rendimiento final. Aunque en las curvas de aprendizaje se observa una divergencia abrupta entre la pérdida de entrenamiento y validación —especialmente en el caso del modelo basado en MFCC—, esta discrepancia no impidió que ambos modelos alcanzaran valores estables de precisión. Sin embargo, el modelo de convolución en una dimensión mostró una capacidad más consistente para generalizar, reflejada en su mayor precisión final sobre el conjunto de validación. Este resultado sugiere que, pese a no trabajar con una representación espectral explícitamente diseñada para resaltar patrones acústicos, la CNN 1D logró capturar características discriminativas directamente desde la forma de onda. Por lo tanto, la arquitectura 1D demostró un mejor equilibrio entre complejidad, capacidad de ajuste y robustez frente a los datos de validación, incluso frente a las oscilaciones observadas en las curvas de aprendizaje.

CONCLUSIÓN

Los resultados obtenidos permiten concluir que, dentro del contexto de este estudio comparativo, el modelo basado en convoluciones unidimensionales demostró una mayor capacidad de generalización en la clasificación de sonidos de gatos y perros. Un aspecto destacado es que, aunque el entrenamiento del modelo 1D presentó una divergencia notable entre la

pérdida de entrenamiento y validación —lo que típicamente podría interpretarse como un indicio de sobreajuste o inestabilidad—, dicha divergencia no se tradujo en un deterioro significativo del rendimiento final. Por el contrario, el modelo alcanzó la mayor precisión entre ambos enfoques, lo que evidencia que fue capaz de aprender patrones relevantes en la forma de onda que se mantuvieron robustos al ser evaluados con datos no vistos.

Este comportamiento resulta especialmente relevante si se considera que el modelo basado en MFCC, con una representación frecuencial más estructurada y una pérdida de validación menor, no logró superar la precisión del modelo 1D. Esto sugiere que, en este caso, la complejidad adicional introducida por la transformación al dominio espectral no garantizó un mejor desempeño, mientras que el modelo 1D supo aprovechar de manera efectiva la información temporal cruda del audio. En otras palabras, incluso frente a señales de entrenamiento que podrían indicar sobreajuste, la arquitectura 1D no solo evitó degradarse, sino que superó en exactitud a la alternativa espectral.

En conjunto, estos hallazgos muestran que el procesamiento directo de la forma de onda puede ser una estrategia competitiva y, en este caso, superior para tareas de clasificación binaria de audio. También evidencian que la interpretación de las curvas de aprendizaje debe complementarse con una evaluación cuantitativa rigurosa, pues una divergencia aparente no necesariamente implica un bajo poder de generalización. Finalmente, los resultados sugieren que futuras investigaciones podrían explorar configuraciones híbridas o técnicas de regularización específicas que permitan estabilizar las curvas del modelo 1D sin afectar su capacidad predictiva, fortaleciendo aún más su potencial en aplicaciones de reconocimiento acústico.

REFERENCIAS

GeeksforGeeks. (2025, 23 de julio). *Melfrequency cepstral coefficients (MFCC) for speech recognition*. GeeksforGeeks. https://www.geeksforgeeks.org/nlp/mel-frequency-cepstral-coefficients-mfcc-for-speech-recognition/

NVIDIA Developer. (n.d.). *Convolution*. https://developer.nvidia.com/discover/convolution