



Escuela de ingeniería y ciencias.

Campus **Monterrey**

Unidad de formación **TC3006C.102**

Semana 2: CNN + Data augmentation.

Equipo **3**

Grupo: **102**

Regina Reyes Juárez - **A01275790**
Nadia Salgado Alvarez - **A01174509**
Gilberto Angel Camacho Lara - **A01613895**
Santiago Miguel Lozano Cedillo - **A01198114**

20 de octubre

Introducción y exploración

En esta práctica se trabajó con redes neuronales convolucionales (CNN) utilizando el dataset CIFAR-10 (conteniendo 60 000 imágenes de 32×32 píxeles en color, divididas en 10 clases (avión, automóvil, pájaro, gato, ciervo, perro, rana, caballo, barco y camión). Todo esto con el objetivo de analizar cómo las técnicas de data augmentation influyen en el rendimiento del modelo. El data augmentation permite aumentar la cantidad y diversidad de imágenes de entrenamiento mediante transformaciones como rotaciones, desplazamientos, volteos o zoom, ayudando a que el modelo generalice mejor y evite el sobreajuste.

Desarrollo del Modelo de Deep Learning

Hiperparámetros:

- **Rotation Range:** indica los grados máximos de rotación que puede aplicarse a una imagen de forma aleatoria.
Por ejemplo, si es 30°, las imágenes se rotarán aleatoriamente entre -30° y +30°.
- **Width Shift Range:** define el rango de desplazamiento horizontal (izquierda o derecha) que se aplica a la imagen.
Se expresa como una fracción del ancho total (por ejemplo, 0.1 significa un 10% de desplazamiento).
- **Height Shift Range:** especifica el rango de desplazamiento vertical (arriba o abajo) de la imagen.
También se expresa como proporción de la altura (por ejemplo, 0.1 equivale a un 10%).
- **Horizontal Flip:** es un valor booleano (True o False) que indica si las imágenes deben invertirse horizontalmente (efecto espejo).
- **Zoom Range:** controla el nivel de zoom aleatorio aplicado a cada imagen, expresado como una proporción.
- Por ejemplo, 0.2 permite acercar o alejar hasta un 20% respecto al tamaño original.
- **Epochs:** representa la cantidad de veces que el modelo recorre todo el conjunto de datos de entrenamiento.
Cada *epoch* permite ajustar los pesos del modelo, por lo que más épocas suelen mejorar el aprendizaje

Parámetros	Rotation range	Width shift range	Height shift range	Horizontal flip	Zoom rang	epochs
Modelo 1	30	0.1	0.1	True	0.2	30
Modelo 2	40	0.1	0.1	False	0.1	20
Modelo 3	70	0.2	0.2	True	0.3	25

Resultados e interpretación

Para este experimento utilizamos el dataset CIFAR-10, que contiene imágenes de 10 clases distintas. El objetivo fue comparar tres modelos de redes neuronales convolucionales (CNN) entrenados con diferentes configuraciones de data augmentation, para observar cómo los cambios en los hiperparámetros afectan la precisión y la capacidad del modelo para generalizar.

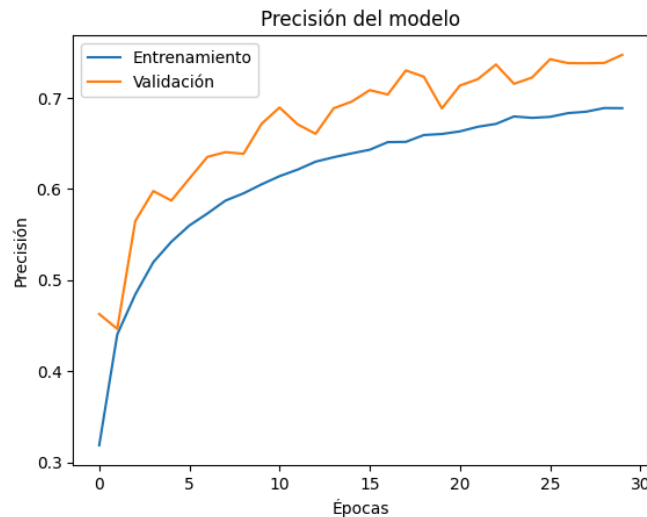


Figura 1: Precisión del modelo 1 - precisión: 0.7470

Modelo 1, figura 1.

En este caso se aplicaron rotaciones de 30° , desplazamientos del 10% tanto horizontal como vertical, volteo horizontal activado y un zoom del 20%, con 30 épocas de entrenamiento. Estos valores son moderados y equilibrados, lo que permitió que el modelo aprendiera sin distorsionar demasiado las imágenes originales. En la gráfica se observa un aumento constante en la precisión tanto de entrenamiento como de validación, sin una separación grande entre ambas curvas. Esto indica que el modelo aprendió de forma estable y sin sobreajustarse. La precisión final en prueba fue de 0.7470, la más alta de los tres modelos, lo que demuestra que esta configuración ayudó al modelo a reconocer bien los patrones visuales sin memorizar los datos.

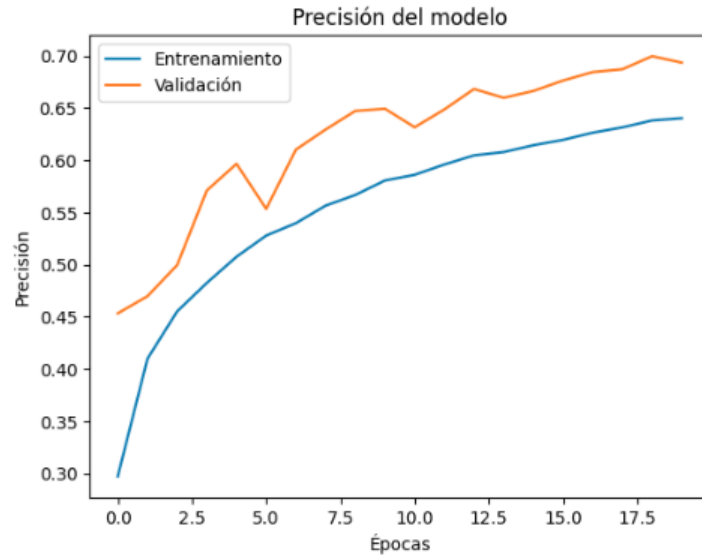


Figura 2: Precisión del modelo 2 - precisión: 0.6933

Modelo 2, figura 2.

Aquí se incrementó la rotación a 40° , se desactivó el volteo horizontal y se redujo el zoom a 0.1, con solo 20 épocas de entrenamiento. Estas modificaciones afectaron el desempeño: al no incluir el volteo horizontal, el modelo tuvo menos ejemplos variados para aprender, y al tener menos épocas, no alcanzó a ajustar completamente los pesos. En la gráfica se ve que la curva de validación crece más rápido al principio, pero luego se estabiliza por debajo del 0.70, mostrando que el aprendizaje fue más limitado. También se nota una ligera distancia entre las líneas de entrenamiento y validación, lo que sugiere que el modelo aprendió los patrones básicos, pero no logró generalizar tan bien como el primero.

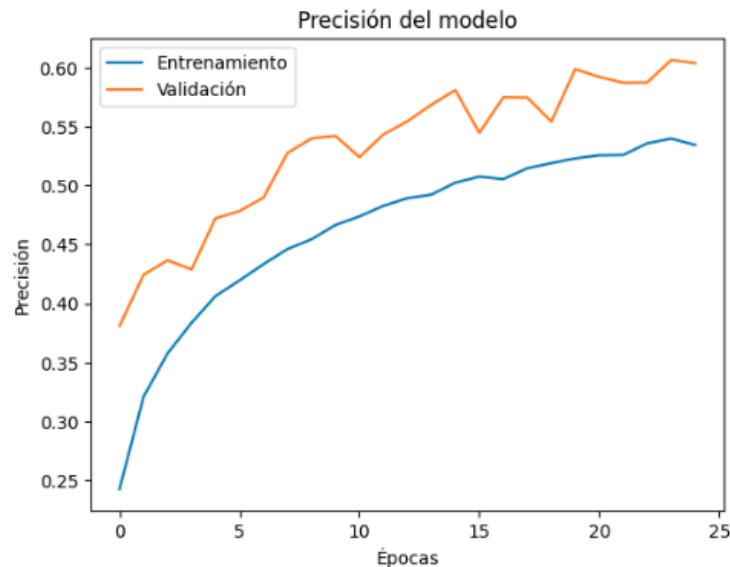


Figura 3: Precisión del modelo 3 - precisión: 0.6037

Modelo 3, figura 3.

En este modelo se probaron valores más extremos: rotaciones de 70° , desplazamientos del 20% en ambas direcciones, volteo horizontal activado y zoom de 0.3, con 25 épocas. Estas transformaciones generaron mucha variación en las imágenes, al punto que el modelo tuvo

dificultades para reconocer correctamente los objetos. En la gráfica se ve que ambas curvas suben de forma lenta y no superan el 0.60 de precisión. Esto muestra que el exceso de aumentación confundió al modelo, ya que las imágenes rotadas y desplazadas en exceso se alejaban demasiado del patrón real de cada clase. Aunque el entrenamiento avanzó de manera estable, el rendimiento final fue bajo porque el modelo no logró aprender representaciones claras de las imágenes originales.

Después de analizar los tres casos, el Modelo 1 fue el más equilibrado y con mejor rendimiento general. Su configuración moderada permitió que la red aprendiera los patrones de las imágenes sin distorsionarlas demasiado, logrando una buena precisión tanto en entrenamiento como en validación. En cambio, el Modelo 2 se quedó corto por falta de variabilidad y menor número de épocas, mientras que el Modelo 3 presentó demasiada alteración visual, reduciendo su capacidad de reconocimiento.

Elegimos el Modelo 1 como el mejor porque alcanzó una precisión de 0.7470, mostró una curva de aprendizaje estable y no presentó signos de sobreajuste. Esto significa que el modelo aprendió correctamente los rasgos importantes de las imágenes y generalizó bien en datos nuevos, cumpliendo con el objetivo de lograr una clasificación efectiva sin sacrificar estabilidad ni rendimiento.

Conclusión

En este trabajo se demostró la importancia de aplicar técnicas de data augmentation en redes neuronales convolucionales (CNN) para mejorar la capacidad de generalización del modelo. A través del análisis de tres configuraciones diferentes, se observó que un uso moderado de las transformaciones como rotación, desplazamiento, volteo y zoom. Permite al modelo aprender de manera más estable y efectiva, evitando tanto el sobreajuste como la pérdida de información relevante.

El Modelo 1, con parámetros equilibrados, logró el mejor desempeño con una precisión del 74.7%, evidenciando que un aumento de datos controlado genera ejemplos suficientemente variados sin distorsionar el patrón original. En cambio, los modelos con menor o mayor grado de aumentación mostraron un aprendizaje limitado o ineficiente.

En conclusión, el uso adecuado del data augmentation es clave para optimizar el entrenamiento de CNNs, y la elección de los hiperparámetros debe buscar siempre un balance entre la variabilidad de las imágenes y la preservación de sus características esenciales.

🔗 M-Itzel.ipynb

Github: https://github.com/santiagolc02/deep_learning

