Universidad Autónoma de Yucatán
Facultad de Matemáticas

Maestría en Ciencias de la Computación

Redes Neuronales Convolucionales

Trabajo Final

Título: RNC para clasificación de imágenes con CIFAR-10

Autor: Mario Herrera Almira

10 de mayo del 2023

Segundo Semestre

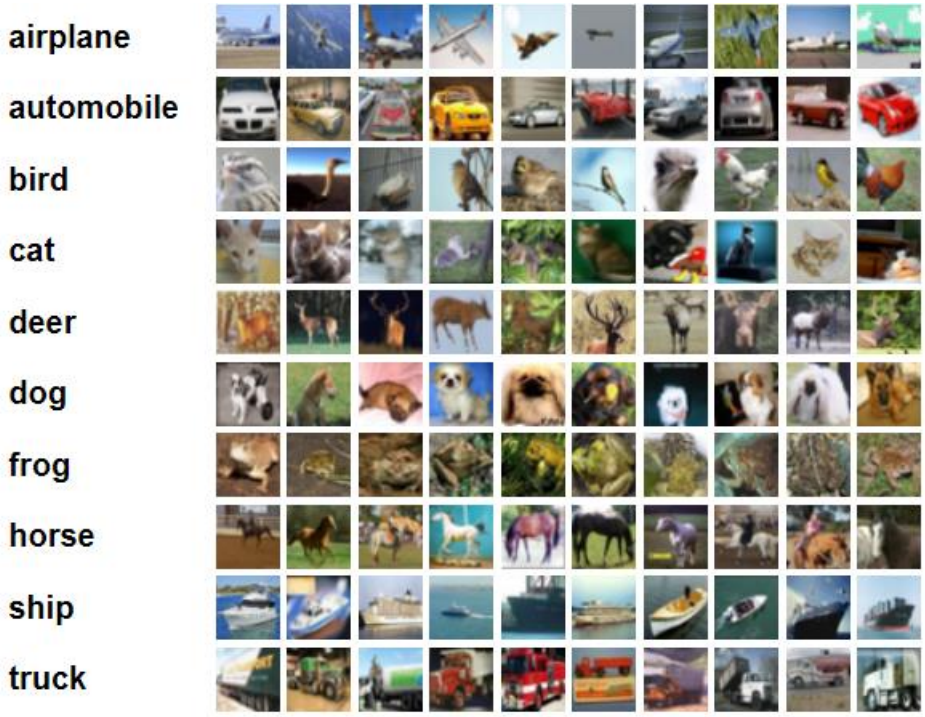
Introducción

El problema de clasificar imágenes de manera automatizada y rápida es uno que ha sido de gran interés por muchos años, ya que tiene diversas aplicaciones en diferentes ámbitos de la ciencia y la industria. Las redes neuronales convolucionales han sido la solución más popular para resolver este problema en los últimos años debido a su gran precisión y velocidad en clasificar imágenes luego de haber sido entrenadas debidamente para este propósito. La clasificación de imágenes es especialmente útil en ámbitos como la detección de productos defectuosos en una línea de producción, proceso que anteriormente necesitaba de la intervención humana, lo que necesitaba de más tiempo y esfuerzo. Otro caso en el que son muy útiles es el reconocimiento del contenido visual de las imágenes, con el aumento del volumen de contenido visual en los últimos años, es importante poder clasificar rápidamente si una imagen posee contenido inapropiado, o simplemente poder organizar la información en grupos o categorías. Las redes neuronales pueden identificar y clasificar objetos en imágenes con una alta precisión. Esto tiene aplicaciones en áreas como la conducción autónoma, la seguridad (detección de intrusos, armas, etc.), la medicina (detección de enfermedades en imágenes médicas) y la agricultura (detección de plagas o enfermedades en cultivos). En resumen, la clasificación de imágenes con redes neuronales es importante porque permite automatizar tareas, analizar y comprender el contenido visual, reconocer objetos, personalizar experiencias, mejorar la seguridad y facilitar el análisis forense. Estas aplicaciones tienen un impacto significativo en una amplia gama de sectores y contribuyen al avance tecnológico y la eficiencia en diversas áreas de estudio.

Metodología

La Base de Datos

CIFAR-10 es la base de datos que se utiliza para entrenar, validar y probar la red neuronal. Esta base de datos contiene un total de 60 mil imágenes a color de 32x32 píxeles de dimensión y se dividen en 50 mil imágenes de entrenamiento y 10 mil imágenes de prueba. Luego del conjunto de entrenamiento se extrae el 10% de las imágenes para realizar la validación durante el proceso de entrenamiento de la red neuronal. CIFAR-10 contiene 10 clases con 6000 imágenes por cada clase, exactamente 1000 imágenes de cada clase fueron seleccionadas de manera aleatoria para conformar el conjunto de prueba. En la siguiente imagen se muestran ejemplos de imágenes que pertenecen a cada una de las clases:



Las Redes Neuronales

Para llevar a cabo la tarea de clasificación se probaron tres redes neuronales diferentes:

1.Red Personalizada

Una red neuronal personalizada que consiste en 4 capas convolucionales y dos completamente conectadas, su estructura está basada en dos capas convolucionales seguida de un Maxpooling, este módulo se repite una vez más y luego se pasa a una red completamente conectada. Se utiliza una capa de Dropout antes de entrar a la red completamente conectada y otro antes de la última capa de esta red, las capas de Dropout permiten apagar algunas neuronas aleatoriamente para ayudar a evitar el sobreajuste. La estructuran de esta red se puede observar en la siguiente tabla:

Layer	Output Shape	Params
Conv2D	(None, 32, 32, 16)	448
LeakyReLU	(None, 32, 32, 16)	0
Conv2D	(None, 32, 32, 32)	4640
LeakyReLU	(None, 32, 32, 32)	0
MaxPooling2D	(None, 16, 16, 32)	0
Dropout	(None, 16, 16, 32)	0

Conv2D	(None, 16, 16, 32)	9248
LeakyReLU	(None, 16, 16, 32)	0
Conv2D	(None, 16, 16, 64)	18496
LeakyReLU	(None, 16, 16, 64)	0
MaxPooling2D	(None, 8, 8, 64)	0
Dropout	(None, 8, 8, 64)	0
Flatten	(None, 4096)	0
Dense	(None, 256)	1048832
LeakyReLU	(None, 256)	0
Dropout	(None, 256)	0
Dense	(None, 10)	2570

Total params: 1,084,234
Trainable params: 1,084,234
Non-trainable params: 0

2.AlexNet

La segunda red con la que se probó fue con AlexNet que consta de una capa convolucional seguida de un Maxpooling, luego se repite esta secuencia y se continúa con tres capas convolucionales y otro Maxpooling. Finalmente se pasa por una red completamente conectada de tres capaz, y se utiliza una capa de Dropout luego de cada una de estas capas, excepto por supuesto después de la capa de salida. La tabla siguiente muestra la estructura de la AlexNet:

Layer	**Output Shape**	**Params**
Conv2D	(None, 8, 8, 96)	2688
MaxPooling2D	(None, 4, 4, 96)	0
Conv2D	(None, 4, 4, 256)	614656
MaxPooling2D	(None, 2, 2, 256)	0
Conv2D	(None, 2, 2, 384)	885120
Conv2D	(None, 2, 2, 384)	1327488
Conv2D	(None, 2, 2, 256)	884992
MaxPooling2D	(None, 1, 1, 256)	0
Flatten	(None, 256)	0
Dense	(None, 4096)	1052672

Dropout	(None, 4096)	0
Dense	(None, 4096)	16781312
Dropout	(None, 4096)	0
Dense	(None, 10)	40970

Total params: 21,589,898
Trainable params: 21,589,898
Non-trainable params: 0

3.VGG-16

Por último, la tercera red neuronal que se probó fue una VGG-16 que consiste en módulos que contienen dos capas convolucionales, seguidas de un Maxpooling, luego se hace un BatchNormalization que permite mantener la media y la desviación estándar de la salida de una capa cercanos a cero y uno respectivamente, por último, se utiliza una capa de Dropout. Esta secuencia se repite una vez más y luego se hace otro módulo prácticamente igual, pero con tres convoluciones contiguas en vez de dos. Por último, se pasa a una red completamente conectada poniendo una capa de Dropout luego de cada capa. La tabla siguiente muestra la estructura descrita anteriormente:

Layer	**Output Shape**	**Params**
Conv2D	(None, 32, 32, 64)	1792
Conv2D	(None, 32, 32, 64)	36928
MaxPooling2D	(None, 16, 16, 64)	0
BatchNormalization	(None, 16, 16, 64)	256
Dropout	(None, 16, 16, 64)	0
Conv2D	(None, 16, 16, 128)	73856
Conv2D	(None, 16, 16, 128)	147584
MaxPooling2D	(None, 8, 8, 128)	0
BatchNormalization	(None, 8, 8, 128)	512
Dropout	(None, 8, 8, 128)	0
Conv2D	(None, 8, 8, 256)	295168
Conv2D	(None, 8, 8, 256)	590080
Conv2D	(None, 8, 8, 256)	590080
MaxPooling2D	(None, 4, 4, 256)	0
BatchNormalization	(None, 4, 4, 256)	1024

Dropout	(None, 4, 4, 256)	0
Conv2D	(None, 4, 4, 512)	1180160
Conv2D	(None, 4, 4, 512)	2359808
Conv2D	(None, 4, 4, 512)	2359808
MaxPooling2D	(None, 2, 2, 512)	0
BatchNormalization	(None, 2, 2, 512)	2048
Dropout	(None, 2, 2, 512)	0
Conv2D	(None, 2, 2, 512)	2359808
Conv2D	(None, 2, 2, 512)	2359808
Conv2D	(None, 2, 2, 512)	2359808
MaxPooling2D	(None, 1, 1, 512)	0
BatchNormalization	(None, 1, 1, 512)	2048
Dropout	(None, 1, 1, 512)	0
Flatten	(None, 512)	0
Dense	(None, 4096)	2101248
Dropout	(None, 4096)	0
Dense	(None, 512)	2097664
Dropout	(None, 512)	0
Dense	(None, 10)	5130

Total params: 18,924,618
Trainable params: 18,921,674
Non-trainable params: 2,944

De las tres redes neuronales que se probaron inicialmente la que mejor resultados arrojó fue la VGG-16, es necesario recalar que tanto la AlexNet como la VGG-16 fueron tomadas exactamente como se encontraron en otros repositorios de Colab. La red neuronal personalizada fue creada a mano con la finalidad de comprender el funcionamiento de la librería Keras y para hacer pruebas de manera rápida. La tabla siguiente muestra una comparación entre las tres redes en cuanto a:

- Cantidad de parámetros a entrenar.
- Tiempo que demora el entrenamiento aproximadamente.
- Precisión alcanzada durante el entrenamiento y durante la prueba.

Red Neuronal	Cantidad Parámetros	Tiempo (minutos)	Precisión Entrenamiento	Precisión Prueba
Personalizada	1,084,234	10	80.70%	77.54%
AlexNet	21,589,898	65	83.80%	58.00%
VGG-16	18,921,674	48	92,48%	85.91%

La red VGG-16 fue la que mostró los mejores resultados y por lo tanto se seleccionó como la indicada para continuar con el proceso. Esta red por defecto estaba configurada para realizar 40 épocas de entrenamiento con una taza de aprendizaje de 0.001 que disminuye en un factor de 10 cada 7 épocas aproximadamente. Se utilizó el optimizador Adam de la librería Keras para ajustar los parámetros y minimizar la función de pérdida de la red neuronal. Se realizaron una serie de experimentos de prueba y error para intentar encontrar el mejor ajuste para los hiperparámetros, tomando en cuenta el número de épocas, la taza de aprendizaje, el factor de disminución de taza de aprendizaje y el optimizador. Los resultados de estas pruebas se pueden observar en la siguiente tabla:

Épocas	TA	Factor	Opt.	Precisión Entrenamiento	Precisión Prueba
40	0.001	10	Adam	92,48%	85.91%
25	0.001	10	SGB	78.42%	72.96%
25	0.005	10	Adam	10.05%	10.00%
25	0.002	10	Adam	10.03%	10.00%
15	0.001	10	Adamax	92.57%	86.16%
15	1.00	10	Adadelta	79.48%	79.84%
15	0.001	12	Adamax	87.41%	83.63%
15	0.001	15	Adamax	94.27%	85.95%
15	0.001	8	Adamax	96.02%	86.15%
15	**0.001**	**6**	**Adamax**	**94.07%**	**86.47%**
15	0.001	4	Adamax	94.55%	85.98%
15	0.001	1	Adamax	91.64%	83.22%

Leyenda:

- TA: Taza de aprendizaje.
- Factor: Factor de disminución de la taza de aprendizaje.
- Opt: Optimizador.

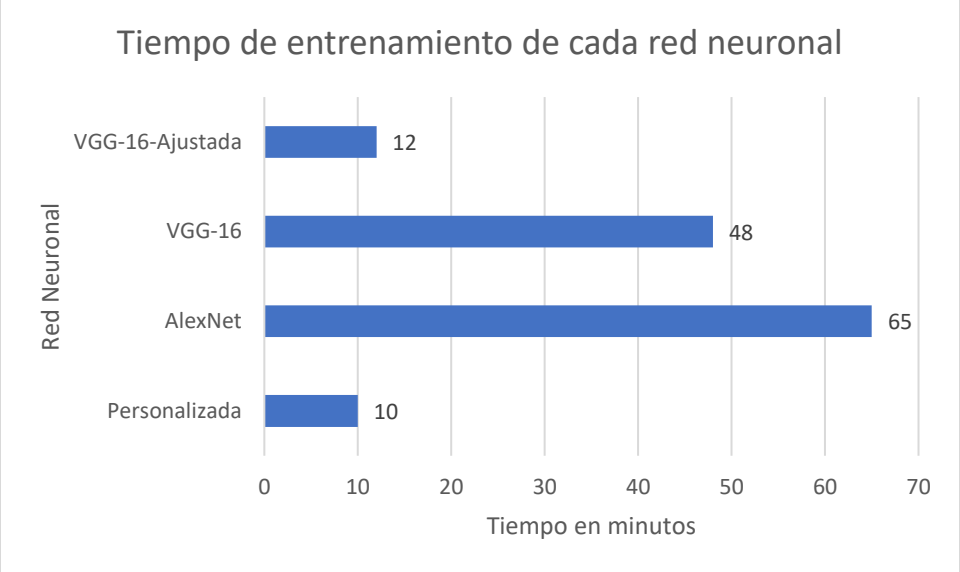
Tras realizar todas las pruebas se obtuvo que la mejor configuración fue la de 15 épocas, una taza de aprendizaje de 0.001 con un decremento de 6 y el optimizador

de Adamax que logra una precisión de 86.47% en el conjunto de prueba. La tabla siguiente muestra esta configuración:

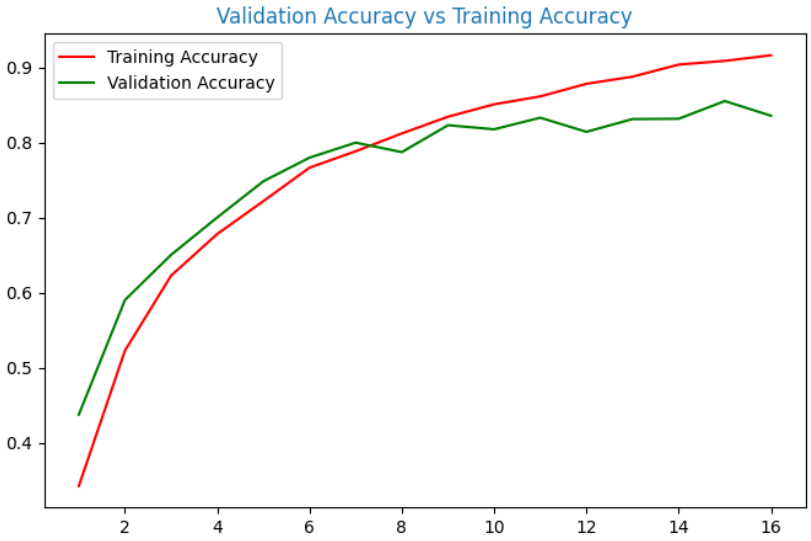
Épocas	TA	Factor	Opt.	Precisión Entrenamiento	Precisión Prueba
15	0.001	6	Adamax	94.07%	86.47%

Es necesario mencionar que esta red no realiza ningún tipo de procesamiento sobre las imágenes ni lleva a cabo aumentación de datos, la única modificación que se realiza es que las imágenes se normalizan para llevarlas a valores entre 0 y 1 lo que aumenta considerablemente la efectividad del entrenamiento de los pesos en la red neuronal.

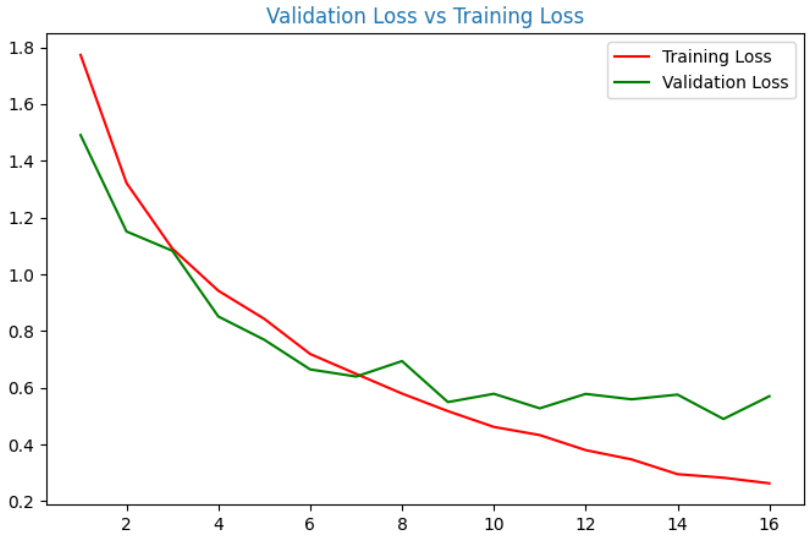
Como se puede observar esta nueva configuración no aumenta considerablemente la precisión de la red a la hora de clasificar las imágenes del conjunto de prueba, pero sí disminuye en gran medida el tiempo que demora la red en entrenar. El tiempo de entrenamiento bajó de 48 minutos a tan solo 12 minutos únicamente por cambiar el número de épocas de 40 a 15 y cambiar el optimizador de Adam por Adamax. En la siguiente gráfica se puede observar una comparación entre las tres primeras redes utilizadas y la VGG-16 con los ajustes finales:



En las siguientes gráficas se pueden observar las curvas que representan el proceso de aprendizaje y de validación durante la etapa de entrenamiento de la VGG-16-Ajustada:



En esta gráfica de comparación entre la precisión de la red neuronal sobre el conjunto de entrenamiento y de validación se puede observar que al final las curvas comienzan a divergir lo que provoca que haya una diferencia notable de un 8% aproximadamente cuando se compara contra los resultados del conjunto de prueba.

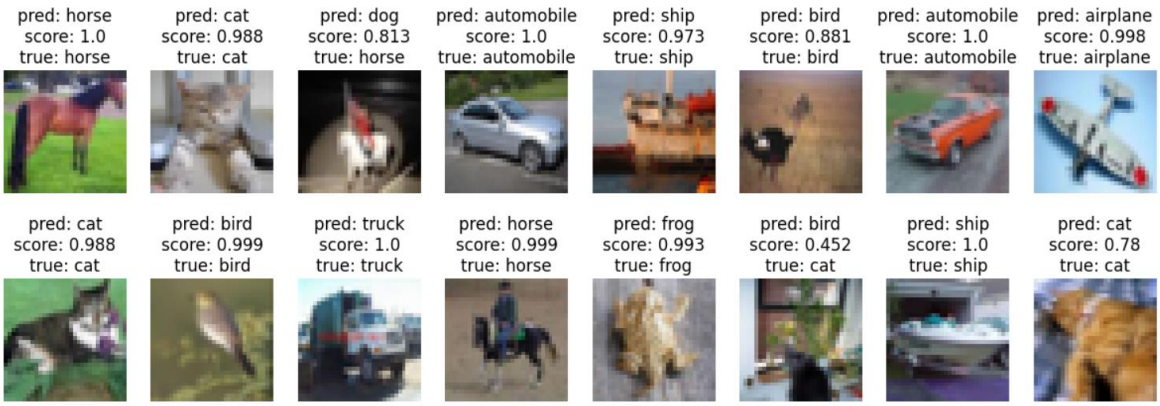


En la gráfica que representa la pérdida también se observa un comportamiento similar entre la curva del conjunto de entrenamiento y la curva del conjunto de validación.

La siguiente figura muestra la matriz de confusión asociada a la red VGG-16-Ajustada:

Como se puede observar en la diagonal de la matriz se encuentras los valores más altos de cada fila lo que quiere decir que en la mayoría de los casos se obtienen buenas predicciones para cada clase. La clase “Cat” tiene un valor de 0.69 siendo la más baja de las diez clases, pero el resto sí presentan valores bastante altos por lo que esta matriz de confusión vuelve a ratificar los resultados obtenidos en los experimentos.

A continuación, se muestra una imagen donde se seleccionaron 16 muestras de manera aleatoria del conjunto de prueba, y se realizó la predicción de las clases a las que pertenecen estas imágenes utilizando la Red Neuronal Ajustada. Como se puede observar solamente se equivoca en dos ocasiones dando un total de 14 predicciones correctas de 16 que representa un 87.50% de precisión en esta muestra aleatoria, por lo que se corresponde con los resultados obtenidos durante el proceso de comprobación de la precisión de la red neuronal.



Conclusiones

Todo el proceso de selección de la base de datos, la red neuronal y el refinamiento de los parámetros permitió obtener de manera correcta una red neuronal convolucional que permite clasificar 10 clases diferentes de objetos con una precisión relativamente alta de 86.47%. Se pueden concluir los siguientes elementos:

- La red AlexNet no funcionó correctamente con la base de datos CIFAR-10 porque se estaba sobre ajustando, arrojando un 83.80% de precisión con el conjunto de entrenamiento, pero solo un 58.00% en el conjunto de prueba.

- La red VGG-16 fue la que mejores resultados arrojó desde un inicio con una precisión de 85.91% sobre el conjunto de prueba.

- El proceso de ajuste de parámetros sobre la VGG-16 solo aumentó la precisión de la red en 0.56%, pero hizo que el tiempo de entrenamiento bajara de 48 minutos a tan solo 12 minutos aproximadamente.

- A pesar de que las curvas de precisión y pérdida entre los conjuntos de entrenamiento y validación tienen una liviana divergencia en las últimas épocas los resultados siguen siendo favorables.

- La matriz de confusión muestra que la clase que le cuesta más trabajo a la red predecir es la clase “Cat” con un 69% de precisión y la que mejor se predice es “Ship” con un 94% de precisión.

- De 16 muestras aleatorias del conjunto de prueba la red predijo correctamente 14 que equivale a un 87.50% de precisión lo cual está acorde a los resultados de los experimentos donde se obtuvo un 86.47% sobre el conjunto de prueba completo.