

Jetson Nvidia

Inteligencia artificial
en tu bolsillo



www



Desde www.ra-ma.es podrá
descargar material adicional.

Ignacio D. Gómez Gracián • Francisco J. Orozco Jiménez
Axel D. Saldívar Zaldívar • Daniel Zaldivar Navarro
Erik Cuevas Jiménez • Marco A. Pérez Cisneros



Ra-Ma®

edü

Ediciones de la U

Jetson Nvidia

Inteligencia artificial en tu bolsillo

*Ignacio Daniel Gómez Gracián • Francisco Javier Orozco
Jiménez • Axel Daniel Saldívar Zaldívar • Daniel Zaldivar
Navarro • Erik Cuevas Jiménez • Marco Antonio Pérez Cisneros*



Ra-Ma®

edü[®]
Ediciones de la U

Gómez Gracián, Ignacio Daniel *et al*

Jetson Nvidia. Inteligencia artificial en tu bolsillo / Ignacio Daniel Gómez Gracián,
Francisco Javier Orozco Jiménez, Axel Daniel Saldivar Zaldívar, Daniel Zaldivar Navarro,
Erik Cuevas Jiménez, Marco Antonio Pérez Cisneros Bogotá: Ediciones de la U, 2026
200 p. ; 24 cm.

ISBN 978-628-7912-56-4 ePUB 978-628-7912-58-8 PDF 978-628-7912-57-1

1. Informática 2. Inteligencia artificial 3. Redes neuronales I. Tít.
006.3 cd. - Thema: UYQ

Edición original publicada por © Editorial Ra-Ma (España)

Edición autorizada a Ediciones de la U para Colombia

Área: Informática

Primera edición: Bogotá, Colombia, julio de 2026

ISBN 978-628-7912-56-4

© Ignacio Daniel Gómez Gracián, Francisco Javier Orozco Jiménez, Axel Daniel Saldivar Zaldívar,
Daniel Zaldivar Navarro, Erik Cuevas Jiménez, Marco Antonio Pérez Cisneros

© Ra-ma Editorial

Calle Jarama, 3-A (Polígono Industrial Igarsa) 28860 Paracuellos de Jarama

www.ra-ma.es y www.ra-ma.com / E-mail: editorial @ra-ma.com

Madrid, España

© Ediciones de la U - Carrera 27 #27-43 - Tel. (+57) 601 6455049

www.edicionesdelau.com - E-mail: editor@edicionesdelau.com

Bogotá, Colombia

Ediciones de la U es una empresa editorial que, con una visión moderna y estratégica de las tecnologías, desarrolla, promueve, distribuye y comercializa contenidos, herramientas de formación, libros técnicos y profesionales, e-books, e-learning o aprendizaje en línea, realizados por autores con amplia experiencia en las diferentes áreas profesionales e investigativas, para brindar a nuestros usuarios soluciones útiles y prácticas que contribuyan al dominio de sus campos de trabajo y a su mejor desempeño en un mundo global, cambiante y cada vez más competitivo.

Coordinación editorial: Daniela Parra G.

Carátula: Ediciones de la U

Impresión: DGP Editores SAS

Calle 63 No. 70 D - 34, Pbx. (571) 7217756

Impreso y hecho en Colombia

Printed and made in Colombia

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro y otros medios, sin el permiso previo y por escrito de los titulares del Copyright.

ÍNDICE

PRÓLOGO	9
¿PARÁ QUIÉN VA DIRIGIDO ESTE LIBRO?.....	10
REQUISITOS PARA TRABAJAR CON ESTE LIBRO	10
¿QUÉ APRENDEREMOS EN ESTE LIBRO?	10
¿CÓMO TRABAJAR CON ESTE LIBRO?.....	11
CONTACTO Y REDES SOCIALES	11
AGRADECIMIENTOS	12
CAPÍTULO 1. INTRODUCCIÓN	13
1.1 REDES NEURONALES	16
1.2 REDES NEURONALES CONVOLUCIONALES	18
1.3 PYTHON	20
1.4 CÁMARA INTEL REALSENSE	22
1.5 ACTUADORES.....	24
1.6 SEÑALES DIGITALES EN LA JETSON NVIDIA	29
1.7 DOCKER	30
1.8 GIT Y GITHUB	33
1.9 MEDIAPIPE	35
1.9.1 Reconocimiento de gestos y movimientos	36
1.9.2 Tareas de texto.....	39
1.10 TENSORFLOW.....	40

CAPÍTULO 2. LINUX.....	41
2.1 MOVERSE POR LINUX	42
2.2 CARACTERÍSTICAS DE LINUX	45
2.3 ESTRUCTURA DE LINUX.....	46
2.3.1 Administración de memoria	48
2.3.2 Espacio de intercambio	51
2.3.3 El Kernel	52
2.4 SISTEMA DE ARCHIVOS EN LINUX	54
2.4.1 Tipos de sistemas de archivos	55
2.4.2 Compatibilidad entre sistemas de archivos	57
2.4.3 Sistema de archivos en Linux.....	58
2.4.4 Archivos ocultos en Linux	60
2.5 COMANDOS ÚTILES.....	61
2.5.1 Concepto de recursividad	61
2.5.2 Comandos para el trabajo con archivos.....	62
2.5.3 Comandos para la obtener información del sistema.....	69
2.5.4 Comandos para la gestión de procesos.....	70
2.5.5 Comandos para redes	71
2.5.6 Comandos para instalación.....	71
2.5.7 Combinaciones de teclas útiles	72
2.6 USUARIOS.....	72
2.6.1 Super usuario	74
2.6.2 Gestión de usuarios	75
2.6.3 Grupos	75
2.7 PERMISOS DE LOS ARCHIVOS.....	76
2.7.1 Comando chmod	78
2.7.2 Comando chown.....	79
2.8 DISTRIBUCIÓN DE LINUX UBUNTU.....	79
2.8.1 Características del sistema	80
2.8.2 Requisitos para instalar Ubuntu	80
CAPÍTULO 3. INTRODUCCIÓN AL DESARROLLO DE INTELIGENCIA ARTIFICIAL CON TARJETAS RASPBERRY PI 4.....	81
3.1 ADVERTENCIAS DE USO DE RASPBERRY PI.....	85
3.2 INSTALACIÓN DEL SISTEMA OPERATIVO	86
3.3 CONFIGURACIÓN DE LA RASPBERRY PI 4	93
3.3.1 Control remoto de la Raspberry Pi 4.....	93
3.3.2 Obtención de la dirección IP de la Raspberry	93
3.3.3 Problemas de visualización	97
3.3.4 Instalación de Visual Studio Code	98
3.3.5 Instalación de librería Pyrealsense	100
3.3.6 Instalación de librería OpenCV	104
3.4 IMPLEMENTACIÓN DE CÓDIGO	106

3.5	DESCRIPCIÓN DEL PRIMER CÓDIGO A IMPLEMENTAR	109
3.6	CORRER CÓDIGOS EN PYTHON	114
CAPÍTULO 4. INTRODUCCIÓN AL DESARROLLO DE INTELIGENCIA ARTIFICIAL ACELERADA CON TARJETAS GRÁFICAS NVIDIA EN WINDOWS 11.....		115
4.1	COMPILACIÓN DE CUDA PARA WINDOWS.....	115
4.2	CÓDIGO DE VISIÓN ARTIFICIAL ACELERADA.....	122
CAPÍTULO 5. PROYECTO JETSON NANO: VISIÓN ARTIFICIAL CON RECONOCIMIENTO DE PROFUNDIDAD.....		129
5.1	INSTALACIÓN SISTEMA OPERATIVO.....	129
5.2	CONTROL REMOTO DE JETSON NANO.....	131
5.3	INSTALACIÓN VISUAL STUDIO CODE	134
5.4	INSTALACIÓN DE LIBRERÍAS CUDA OPENCV	135
5.5	INSTALACIÓN DE LIBRERÍAS PYREALSENSE	138
5.6	CÓDIGO DETECCIÓN DE DISTANCIA DE PERSONAS	139
5.7	CONEXIÓN DE JETSON NANO AL VEHÍCULO INTELIGENTE.....	146
CAPÍTULO 6. PROYECTO JETSON ORIN NANO: VISIÓN ARTIFICIAL MULTICÁMARA CON RECONOCIMIENTO DE PROFUNDIDAD.....		149
6.1	INSTALACIÓN SISTEMA OPERATIVO.....	149
6.2	CONTROL REMOTO	150
6.3	INSTALACIÓN DE LIBRERÍAS CUDA OPENCV	152
6.4	INSTALACIÓN DE LIBRERÍA PYREALSENSE.....	153
6.5	CÓDIGO DETECCIÓN DE DISTANCIA DE PERSONAS CON MÚLTIPLES CÁMARAS	154
CAPÍTULO 7. DOCKER EN LA JETSON NVIDIA.....		161
7.1	TENSORFLOW EN DOCKER.....	162
7.2	ENTRENAMIENTO DE RED NEURONAL EN TENSOFLOW.....	164
CAPÍTULO 8. MEDIAPIPE JETSON ORIN NANO.....		169
8.1	CREACIÓN DEL ENTORNO VIRTUAL E INSTALACIÓN DE MEDIAPIPE	170
8.2	CONTADOR DE DEDOS	171
8.3	CONTROL DE LEDS CON GESTOS	176
8.4	DETECCIÓN DE ROSTRO.....	180
CAPÍTULO 9. CONEXIÓN SERVER SOCKET.....		183
MATERIAL ADICIONAL		199

PRÓLOGO

¡Es importante que leas toda la introducción para evitar problemas y confusiones en el futuro!

Este libro te enseñará a manejar de manera profesional las plataformas Jetson de NVIDIA, líderes en inteligencia artificial y sistemas embebidos. Estas plataformas utilizan el sistema operativo Linux, ampliamente reconocido en el mercado del software libre y esencial en el mundo de los sistemas digitales.

Además, exploraremos el papel crucial de los servidores en la informática. Aprenderás qué son, cómo se utilizan y por qué son fundamentales para la comunicación y el flujo de datos eficaces e íntegros entre múltiples dispositivos. Esto es relevante tanto para organizaciones internas como para el público general que usa Internet diariamente como medio de acceso a la información.



Los proyectos se construirán de manera progresiva, por lo que es esencial seguir el orden del libro para comprender cómo desarrollar proyectos cada vez más complejos, sobre todo en los proyectos de visión artificial.

¿PARÁ QUIÉN VA DIRIGIDO ESTE LIBRO?

Está dirigido a cualquier persona entusiasta que quiera aprender sobre inteligencia artificial aplicada con visión artificial para señales de control. También está enfocado para aquellos que deseen aprender sobre Linux, la API de CUDA y demás bibliotecas.

En este libro se planea abordar un par de proyectos que resultan sumamente interesantes en el mundo no solo de la informática, sino también de la electrónica e incluso de la industria automotriz, dado que se tratarán temas que envuelven conceptos actuales tales como la inteligencia y la visión artificial embebidas en tarjetas Jetson de NVIDIA.

REQUISITOS PARA TRABAJAR CON ESTE LIBRO

Para desarrollar los proyectos propuestos en este libro, será necesario que el lector cuente con alguno de los siguientes hardware:

1. Jetson Nano.
2. Jetson Orin Nano.
3. Cámara de profundidad Intel Realsense.
4. Raspberry Pi 4 (opcional).

¿QUÉ APRENDEREMOS EN ESTE LIBRO?

En este libro vamos a desarrollar proyectos en el que implementaremos Visión Computacional con Inteligencia Artificial acelerada con la API de CUDA.

1. Implementación de Visión Computacional acelerada en ordenadores con Windows.
2. Introducción al sistema operativo Ubuntu utilizado en plataformas Jetson.
3. Implementación de Visión Computacional de Profundidad con cámara Intel Real-Sense con Jetson Nano.

4. Implementación de Visión Computacional de Profundidad multicámara Intel Real-Sense con Jetson Orin Nano.
5. Implementación de Docker en Jetson Orin Nano para desarrollo de TensorFlow.
6. Implementación de MediaPipe para la detección y conteo de dedos.

El objetivo es aprender a programar inteligencia artificial acelerada, enfocada en visión computacional para detectar automóviles y personas, para enviar señales de control.

La mayor parte de los proyectos se llevarán a cabo en las plataformas Jetson que están diseñadas para correr inteligencia artificial en sistemas embebidos de forma eficiente.

¿CÓMO TRABAJAR CON ESTE LIBRO?

Primero que nada, las instrucciones para descargar el contenido vienen en el último apartado del libro llamado **Material adicional**.

En este material podrás encontrar los códigos de ejemplos que se mencionan a lo largo de los capítulos de este libro.

CONTACTO Y REDES SOCIALES

Síguenos en nuestras redes sociales para que estés al pendiente de novedades y actualizaciones, y para que puedas contactar con nosotros, en caso de que se te presente alguna duda. Búscanos en Facebook, estamos como @TipsExcel2020 y en nuestra página de Instagram, estamos como @tips_excel_2021. Podrás ubicarnos con el siguiente logotipo (podrá variar mínimamente según la temporada del año).



También puedes consultar nuestra página web (en caso de que el enlace llegase a cambiar, podrás checarlo en nuestras redes sociales).

<https://sites.google.com/view/tips-excel-2021/inicio>

AGRADECIMIENTOS

Esta obra no hubiera sido posible sin la importante colaboración de algunos integrantes del Laboratorio de Inteligencia Artificial, Optimización, Metaheurística y Tecnologías Sustentables del Centro Universitario de Ciencias Exactas e Ingenierías de la Universidad de Guadalajara, los cuales estuvieron apoyando activamente a cumplir con los objetivos planteados para este texto.

En primer lugar, reconocemos a las ingenieras y entrañables amigas que estuvieron apoyando con la parte de la implementación de la cámara de profundidad Intel Realsense para la detección de obstáculos en vehículos autónomos. Diana Miriam Escobedo Ibarra, Alondra Jocelyn Álvarez Americano y Amy Elizabeth Celis Sánchez.

Adicionalmente reconocemos el trabajo del maestro en ciencias Óscar Alan González Sánchez quien nos apoyó con algunos temas de inteligencia artificial, así como el trabajo realizado por el futuro ingeniero, Isaac Saldívar Zaldivar por su aporte a la revisión de algunos ejercicios prácticos que se implementaron en esta obra.

Agosto del 2025
Universidad de Guadalajara, CUCEI
Guadalajara, Jalisco, México

1

INTRODUCCIÓN

La inteligencia artificial (IA) está revolucionando el mundo en múltiples aspectos, transformando industrias, mejorando la eficiencia en procesos y la forma en la que interactuamos con la tecnología. Desde la automatización de procesos hasta la creación de vehículos autónomos.

La empresa líder que está llevando esta transformación es NVIDIA. La principal plataforma que impulsa las máquinas autónomas y aplicaciones integradas es la NVIDIA Jetson, las cuales son computadoras compactas de alto rendimiento, complementadas con librerías para un desarrollo de software acelerado en IA.

Cada NVIDIA Jetson es un Sistema en Módulo (SOM) completo, estas incluyen GPU (Unidad de Procesamiento Gráfico), CPU (Unidad Central de Procesamiento), memoria, interfaces y más. Existen en una amplia gama de rendimiento, eficiencia y precio para ser utilizados en distintos proyectos según la escala.

La principal razón sobre porque se utilizan Jetson para correr Inteligencia Artificial sobre otros sistemas embebidos como la Raspberry, es porque cuentan con una GPU que ejecuta IA de manera más rápida y eficiente que la CPU. La GPU procesa operaciones en paralelo, posee más núcleos que la CPU y puede manejar un gran volumen de datos de forma simultánea.

Los módulos que usaremos en este libro son Jetson Nano 4GB y Jetson Orin 8GB, en la tabla 1 se muestran sus principales características, en modo de resumen la tabla Jetson Orin Nano es hasta 80 veces más rápida que la Jetson Nano, aunque también se eleva su costo. Ambas dan un gran rendimiento en inteligencia artificial y se pueden considerar un módulo de bolsillo.

Módulo	CPU	GPU	RAM
Jetso Nano 4GB	Quad-core ARM A57 de 4 núcleos	128-núcleos Maxwell	4 GB 64-bit
Jetson Orin Nano	Arm® Cortex®-A78AE de 6 núcleos	1024-núcleos Ampere y 32 núcleos Tensor	8 GB 64-bit

Tabla 1: Especificaciones técnicas de Jetson; página oficial de NVIDIA

En la Figura 1.1 de abajo se muestra una imagen de la Jetson Nano junto con sus puertos.

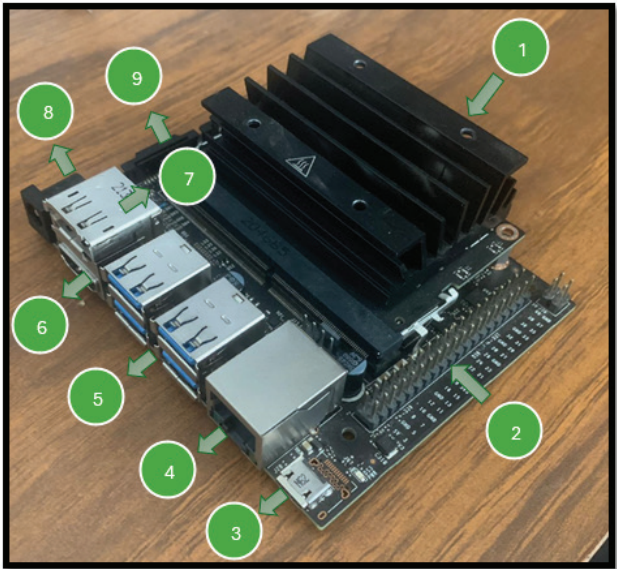


Figura 1.1. Jetson Nano

- 1. Puerto microSD (128GB espacio recomendado).
- 2. Cabecera de expansión de 40 pines.
- 3. Puerto micro-USB para entrada de alimentación de 5V o para modo dispositivo.
- 4. Puerto Gigabit Ethernet.
- 5. 4 puertos USB 3.0.
- 6. Puerto HDMI de salida.

7. Puerto Display.
8. Conector DC Barrel para entrada de alimentación 5V.
9. Conectores de cámara MIPI CSI-2.

A continuación, se muestra la Jetson Orin Nano, Figura 1.2:

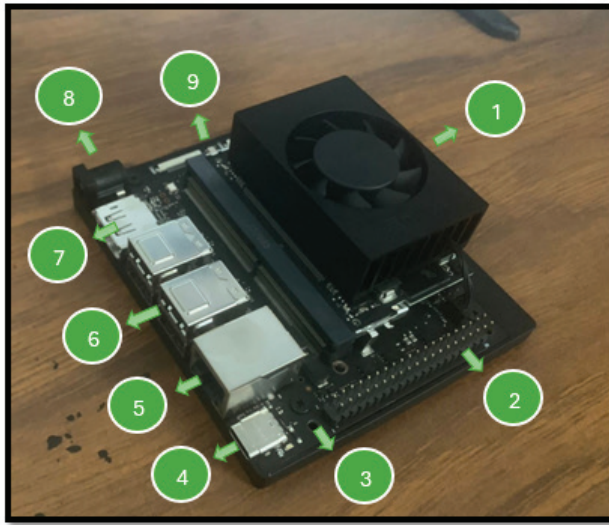


Figura 1.2. Jetson Nano

1. Puerto microSD (128GB espacio recomendado).
2. Cabecera de expansión de 40 pines.
3. Indicador Led de encendido.
4. Puerto USB-C solo para datos.
5. Puerto Gigabit Ethernet.
6. 4 puertos USB 3.1 tipo A.
7. Puerto Display Port.
8. Conector DC Barrel para entrada de alimentación 5V.
9. Conectores de cámara MIPI CSI.

1.1 REDES NEURONALES

Gran parte del auge de la inteligencia artificial se debe a las redes neuronales, un método que enseña a las computadoras a procesar datos de manera similar a cómo lo hace el cerebro humano. Estas redes aprenden a través de un proceso conocido como Machine Learning, y dentro de este campo, se usa una especialización particularmente importante el Deep Learning.

El Machine Learning es una técnica que implica ajustar o aproximar un modelo mediante un conjunto de datos de entrenamiento. Estos datos pueden ser documentos, archivos de audio, imágenes, entre otros. El **modelo** es el resultado final obtenido tras el entrenamiento con los datos de entrada. Este proceso de entrenamiento del modelo es una etapa clave del Machine Learning. Cuando se cuenta con un modelo ya entrenado, se puede aplicar a nuevos datos para generalizar nuevas salidas de datos. En la Figura 1.3 se muestra de manera simplificada este proceso.

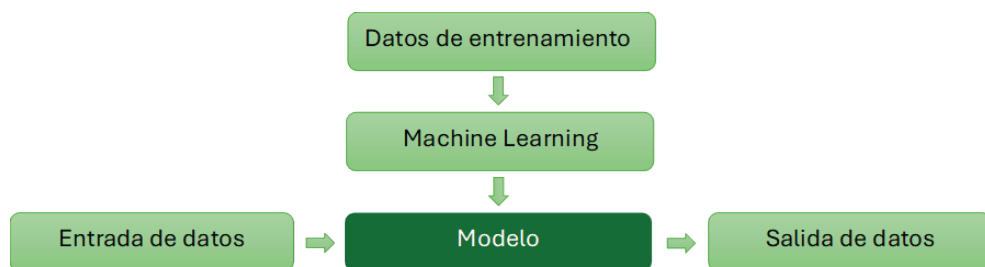


Figura 1.3. Modelo básico de Machine Learning

En el campo del Machine Learning, los métodos se pueden clasificar en tres categorías principales: aprendizaje supervisado, no supervisado y reforzado. Nos centraremos en el aprendizaje supervisado, el cual se caracteriza por utilizar datos de entrenamiento acompañados de sus correspondientes salidas o etiquetas. Esta información es esencial para que el modelo aprenda a realizar predicciones o aproximaciones precisas. Dentro del aprendizaje supervisado, los usos más comunes incluyen la clasificación, donde se categorizan los datos en diferentes grupos, y la regresión, donde se predicen valores.

Las redes neuronales son un tipo de implementación del aprendizaje automático que utiliza el aprendizaje supervisado. En la Figura 1.3, se reemplaza el modelo tradicional por una red neuronal. Las redes neuronales imitan el funcionamiento del cerebro, que consta de múltiples neuronas; en el contexto del aprendizaje automático, cada neurona se denomina **nodo** y las conexiones entre ellas se llaman **pesos**. Al igual

que el cerebro posee un gran número de conexiones neuronales, las redes neuronales consisten en una red de nodos con sus respectivos pesos de conexión. Estas redes incluyen una capa de entrada, varias capas ocultas y una capa de salida. La Figura 1.4 ilustra una red neuronal.

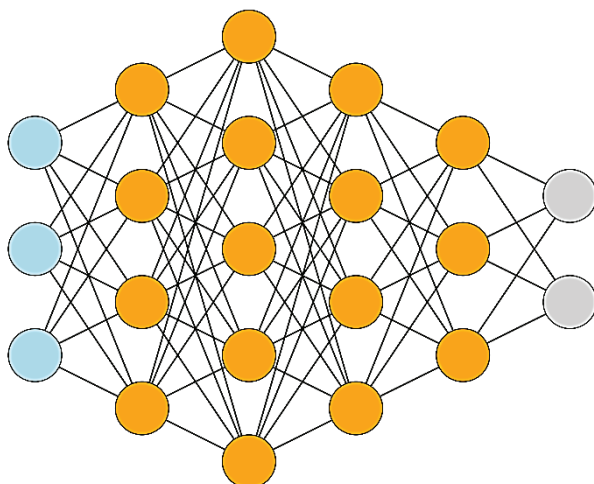


Figura 1.4. Red neuronal profunda: primera capa (entrada), cuatro capas intermedias (capas ocultas) y última capa (salida)

Los modelos iniciales de redes neuronales presentaban una estructura simple, con una única entrada y salida, conocidas como redes neuronales de una sola capa. La evolución de estas estructuras incorporó capas ocultas, dando origen a las redes neuronales multicapa. Las que poseen una capa oculta se denominan redes neuronales poco profundas, mientras que las que incluyen dos o más capas ocultas se clasifican como redes neuronales profundas.

Deep Learning es el uso y desarrollo de redes neuronales profundas. Su impacto ha sido significativo y transformador. Mejora drásticamente el rendimiento de las tecnologías de reconocimiento de voz e imagen. Impulsa los sistemas de vehículos autónomos y sistemas de recomendación personalizados en aplicaciones.

Referencias

Kim, P. (2017). Matlab deep learning. With machine learning, neural networks and artificial intelligence, 130 (21).

1.2 REDES NEURONALES CONVOLUCIONALES

Las redes neuronales son un subconjunto del Machine Learning, y se encuentran en el centro de los algoritmos del Deep Learning. Las Redes Neuronales Convolucionales (por sus siglas en inglés CNN, *Convolutional Neuronal Network*) se distinguen de otras redes neuronales por un mejor desempeño con entradas de señal de imagen, voz o audio. Contiene tres tipos de capas:

- Capa convolucional.
- Capa Pooling.
- Capa totalmente conectada (FC).

La capa convolucional es el núcleo de una CNN, ya que en ella se realiza la mayor parte del procesamiento. Está conformada por elementos como los datos de entrada, el filtro y el mapa de características. Por ejemplo, si la entrada es una imagen —representada como una matriz tridimensional de píxeles—, se aplica un detector de características, conocido como kernel o filtro, que se desplaza a lo largo de la imagen. Este procedimiento recibe el nombre de **convolución** y en la figura 1.5 se ilustra cómo se lleva a cabo.

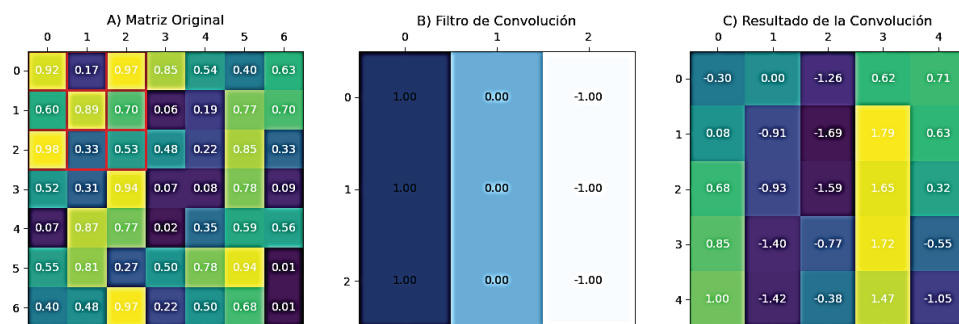


Figura 1.5. Proceso de convolución

En la Figura 1.5 de arriba el primer resultado que se muestra en la matriz C) se obtuvo con la siguiente operación: $\text{salida}[0][0] = (0.92) \cdot (1.00) + (0.17) \cdot (0.00) + (0.97) \cdot (-1.00) + (0.60) \cdot (1.00) + (0.89) \cdot (0.00) + (0.70) \cdot (-1.00) + (0.98) \cdot (1.00) + (0.33) \cdot (0.00) + (0.53) \cdot (-1.00) = 0.30$. La matriz A) es la matriz de entrada, puede ser la imagen de un perro, un gato o una calle. La matriz B) es un filtro especial para detección de bordes de una imagen. Recorriendo el filtro a través de toda la imagen se obtiene la matriz final C).

La capa *pooling* tiene como objetivo reducir las dimensiones espaciales (ancho y alto) de las características extraídas por las capas de convolución, mientras se preservan los componentes más importantes. Esto sirve para reducir el número de parámetros, la carga computacional y evitar el sobreajuste.

Los principales tipos de pooling son el **Max pooling** y el **Average Pooling**. El Max Pooling selecciona el valor máximo dentro de una región en la matriz de entrada, nos da la ventaja de conservar las características más importantes de bordes y texturas. Mientras el Average Pooling calcula igualmente ciertos valores dentro de una región de la matriz, con la diferencia de suavizar las características y nos sirve principalmente para tareas donde la información detallada no es estrictamente necesaria. En la Figura 1.6 se muestra un ejemplo de Max Pooling.

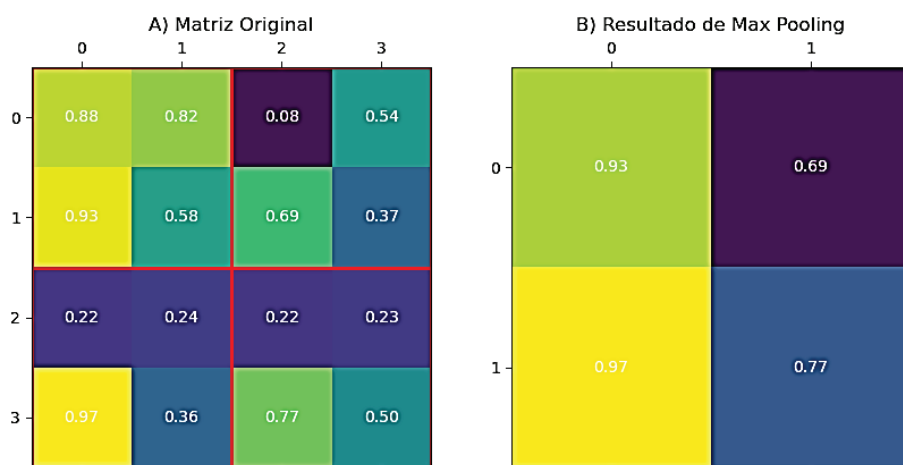


Figura 1.6. Proceso de Max Pooling

En la Figura 1.6 para llevar a cabo el Max Pooling, primero se elige el tamaño de la ventana en nuestro caso es 4x4, de esa ventana se selecciona el valor más alto y se pasa a la nueva matriz B), el mismo procedimiento se repite para cada cuadrante de la matriz A) generando de esta manera, la nueva matriz ya compactada y más sencilla de procesar.

La capa *Fully Connected*, está conectada a todas las neuronas de la capa anterior. Esta conexión densa permite que la red combine las características extraídas de capas anteriores como las convolucionales y de pooling, las combina para formar representaciones más abstractas y de mayor nivel.

Todos estos conceptos los pondremos en práctica conforme desarrollemos los proyectos propuestos a lo largo de todo el libro.

1.3 PYTHON

Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad del código. Es un lenguaje multiparadigma, esto quiere decir, que soporta programación orientada a objetos, imperativa y en menor medida funcional. Es fácil de aprender e implementar en una variedad de plataformas, además cuenta con un gran ecosistema de paquetes de terceros, haciéndolo adecuado para una amplia gama de aplicaciones.



Figura 1.7. Logotipo de Python

La evolución y popularidad de Python en el campo de la Inteligencia Artificial (IA) ha generado una convergencia entre este lenguaje de programación y las aplicaciones avanzadas de IA. Python se destaca especialmente en el procesamiento de Lenguaje Natural (NLP), donde librerías como *Natural Language Toolkit* (NLTK) y *SpaCy* proporcionan herramientas potentes y accesibles para el análisis y procesamiento del lenguaje humano.

En el ámbito de la visión por computadora, librerías como *OpenCV* y *TensorFlow* facilitan la implementación de tareas complejas como el reconocimiento de imágenes y el seguimiento de objetos, haciéndolas más accesibles y eficientes, entre otros campos.

Otra razón por la cual Python es ampliamente utilizado en el campo de la IA, es gracias a la API (Interfaz de Programación de Aplicaciones) de *CUDA* de NVIDIA. *CUDA Python* provee una amplia y uniforme caja de herramientas para utilizar simplícidamente el procesamiento paralelo basado en GPU. Eso crea un camino para utilizar código acelerado en paralelo de forma más sofisticado y con un mínimo de sintaxis.

Python, siendo un lenguaje de programación versátil, soporta programación orientada a objetos (POO), permitiendo no solo estructurar el software en torno