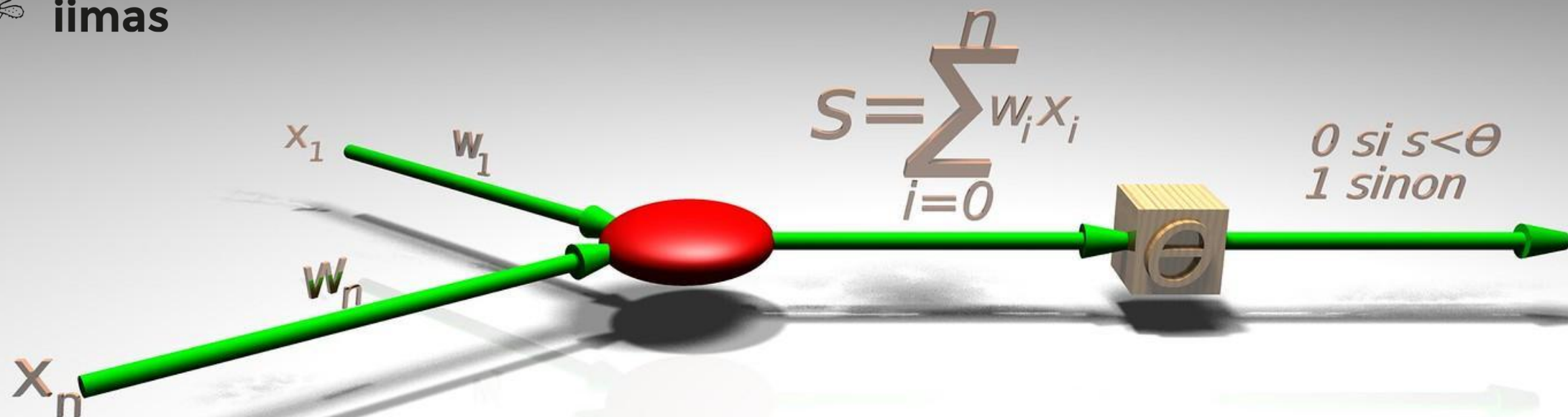




iimas

PROGRAMA INTENSIVO EN CIENCIA DE DATOS:
DE LOS FUNDAMENTOS AL DEEP LEARNING



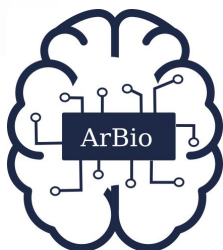
Módulo 3. Redes Neuronales Artificiales

Redes neuronales convolucionales

Dra. Blanca Vázquez

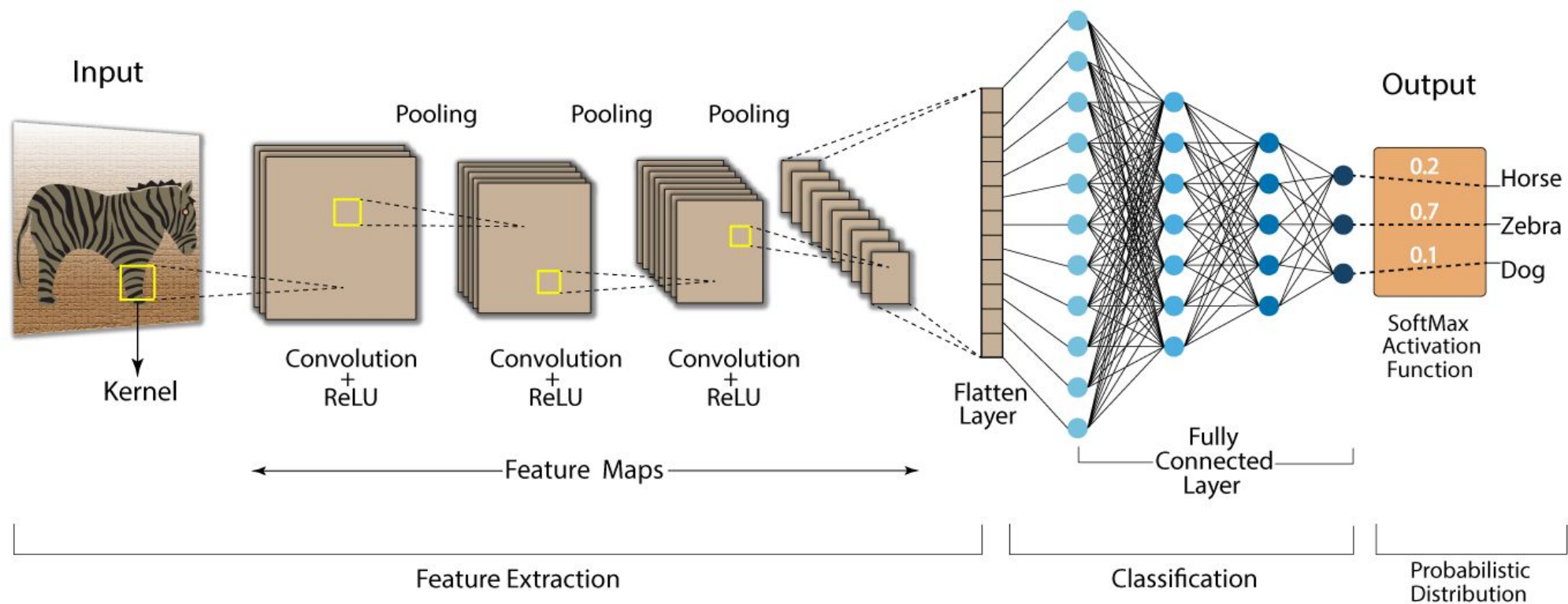
Dra. Nidiyare Hevia Montiel

Ing. Gabriel Carcedo Rodríguez



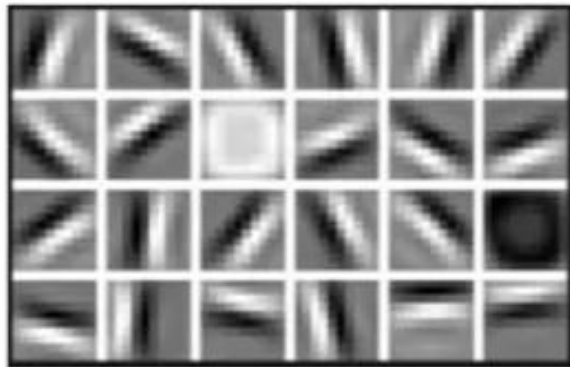
Motivación

Convolution Neural Network (CNN)



Motivación

Low Level Features



Lines & Edges

Mid Level Features



Eyes & Nose & Ears

High Level Features



Facial Structure

Operación de convolución

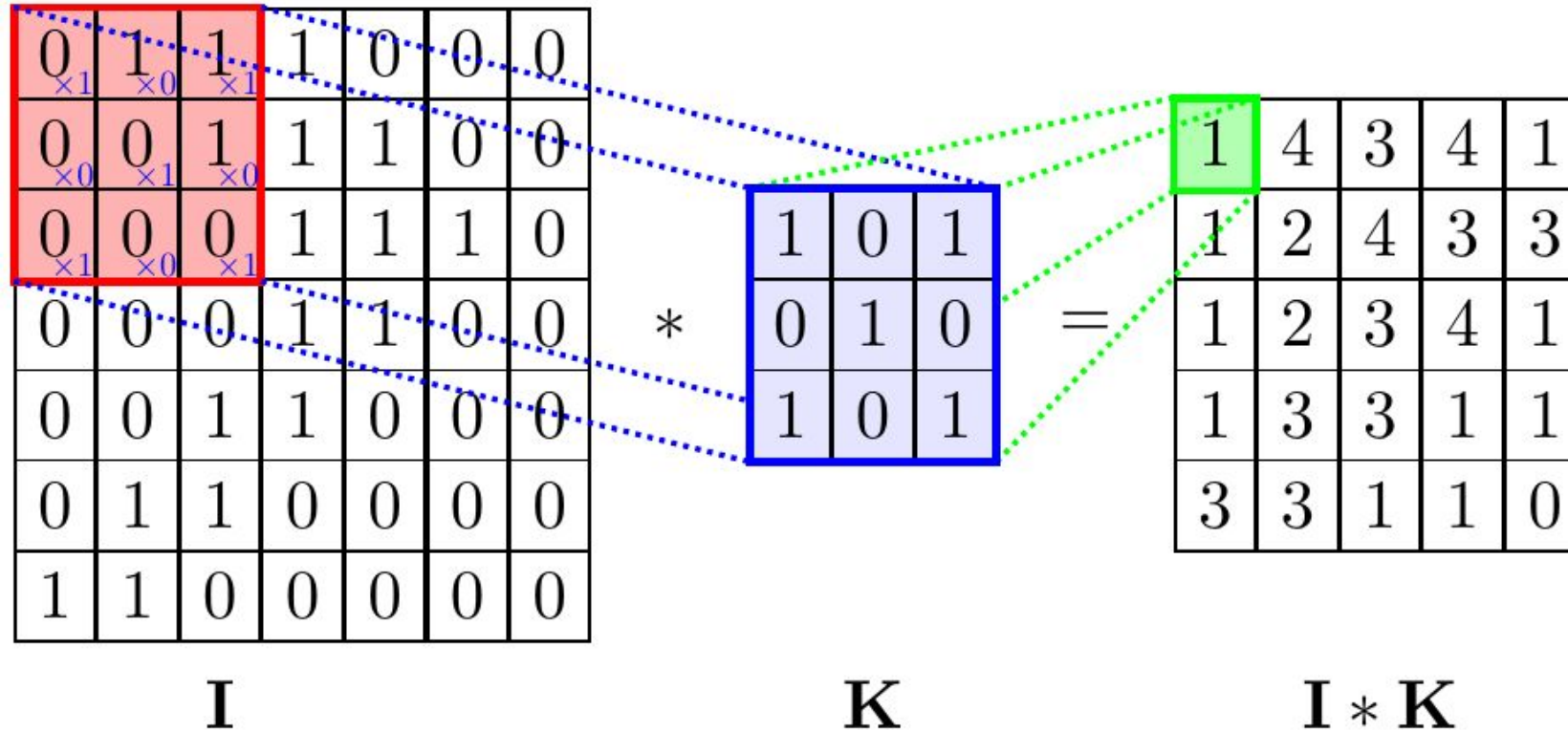
Es una operación **matemática** que combina dos funciones (f , g) para describir la superposición entre ambas.

La convolución toma dos funciones:

- “**Desliza**” una función sobre la otra,
- **Multiplica** los valores de las funciones en todos los puntos de superposición, y
- **Suma** los productos para crear una nueva función.

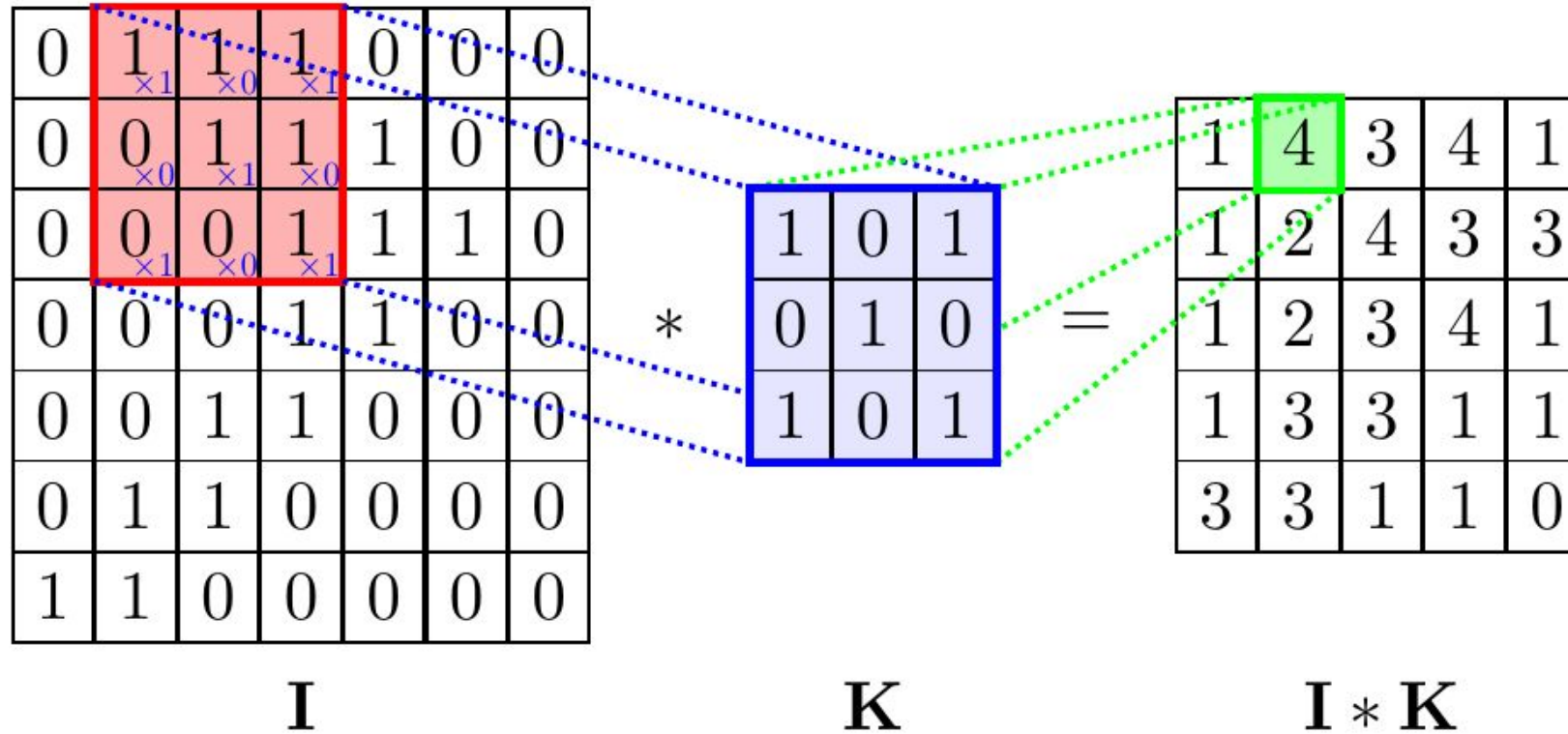
La nueva función representa cómo interactúan las dos funciones originales entre sí.

Convolución en el procesamiento de imágenes

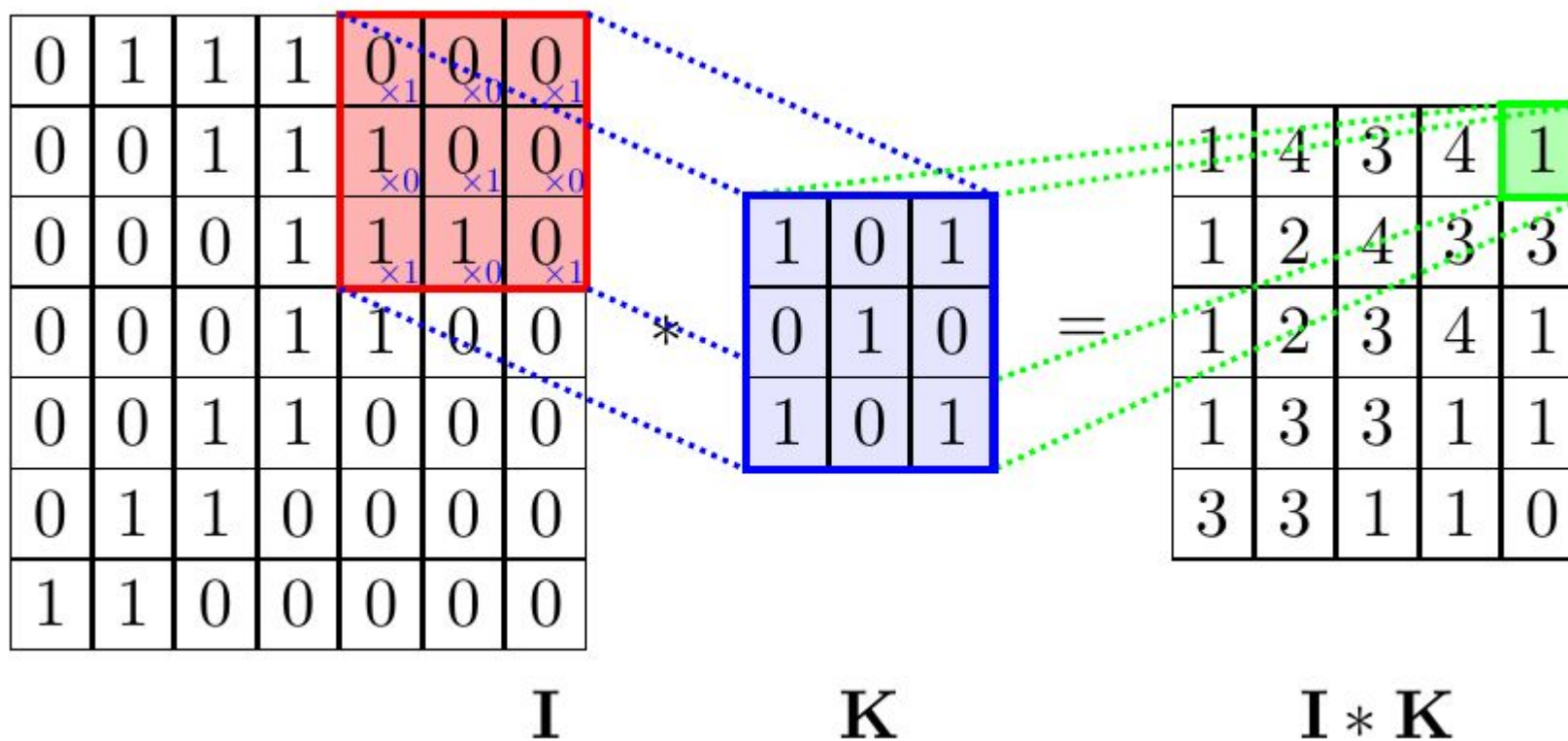


La salida de una convolución se le conoce como: mapa de características

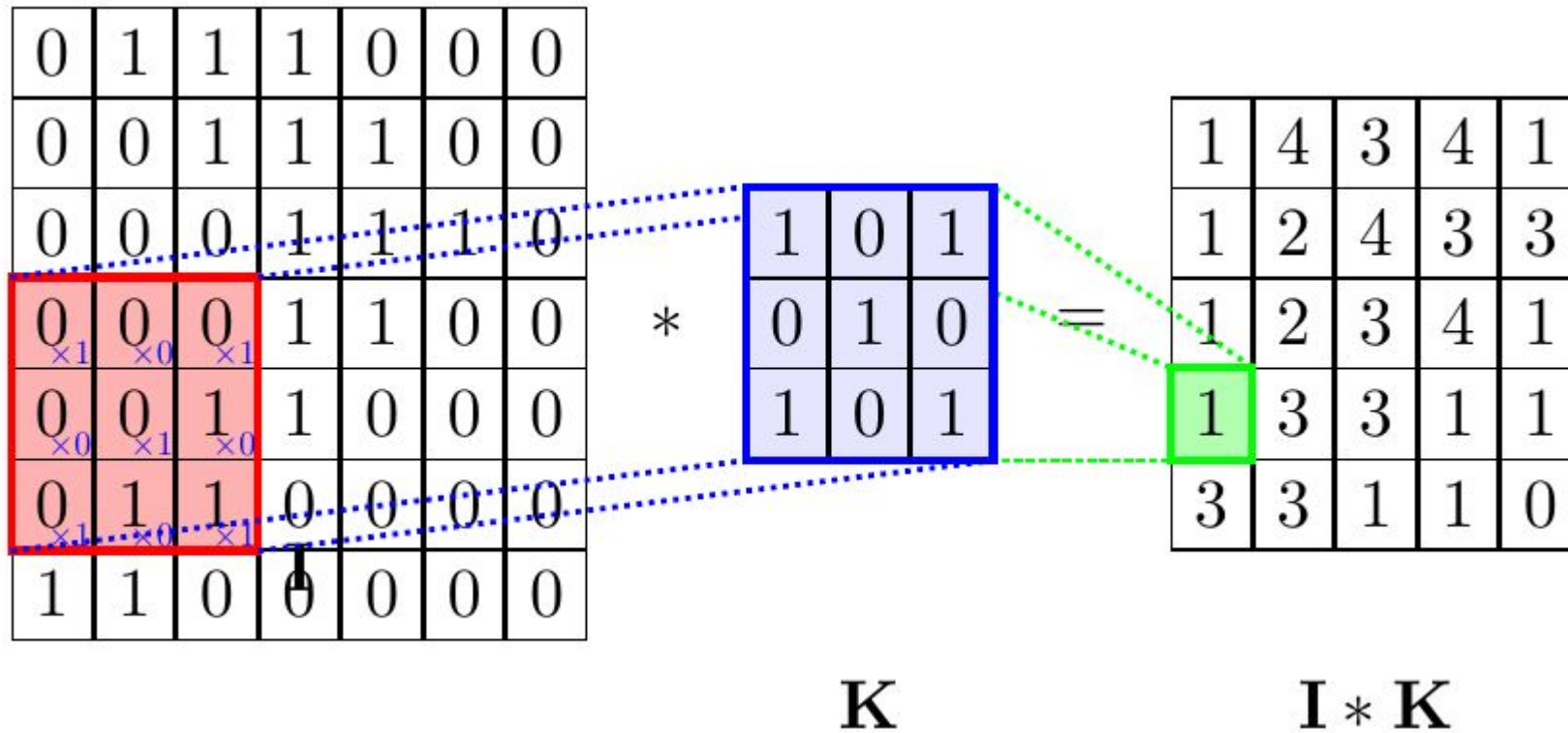
Convolución en el procesamiento de imágenes



Convolución en el procesamiento de imágenes

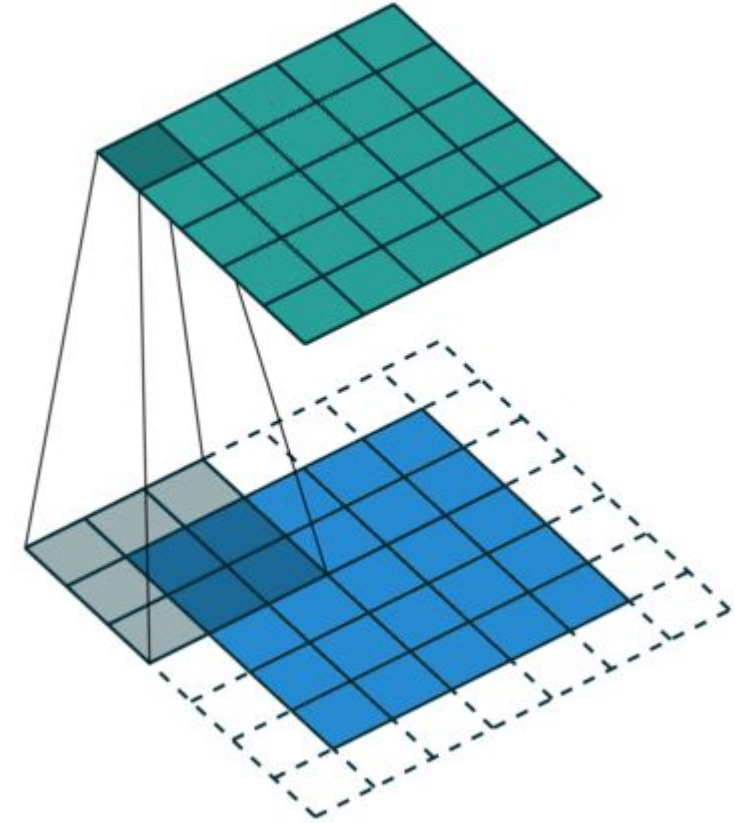


Convolución en el procesamiento de imágenes



Stride y padding

- **Stride:** define el tamaño del desplazamiento del kernel a través de los datos de entrada.
- **Padding:** determina si existe o no un aumento en la resolución del mapas de segmentación de entrada. Normalmente este aumento se consigue añadiendo píxeles de valor nulo.
- **Kernel:** define el tamaño del campo de visión de la convolución. Un tamaño común es $k = 3$.



Convolución 2D usando un kernel de tamaño 3, stride de 1 and padding.

Submuestreo: capas de agrupación

Son utilizadas para sintetizar / reducir la información proveniente de capas anteriores mediante una operación en particular.

Existen dos tipos principales
de capas de agrupación:



- **Max Pooling:** Toma el máximo elemento dentro de la ventana de aplicación.
- **Average Pooling:** Toma el promedio de los elementos dentro de la ventana de aplicación.

Submuestreo: capas de agrupación

Salida de una capa de convolución

| | | | |
|---|---|---|---|
| 1 | 2 | 2 | 1 |
| 4 | 9 | 1 | 0 |
| 1 | 5 | 2 | 3 |
| 4 | 6 | 1 | 2 |

Max pooling

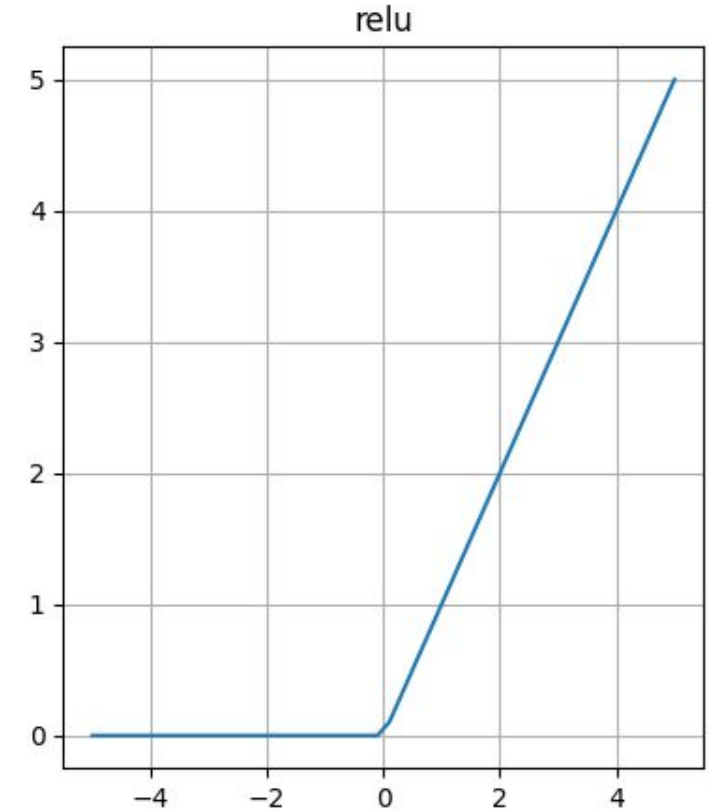
| | |
|---|---|
| 9 | 2 |
| 6 | 3 |

Average pooling

| | |
|---|---|
| 4 | 1 |
| 4 | 2 |

Función de activación

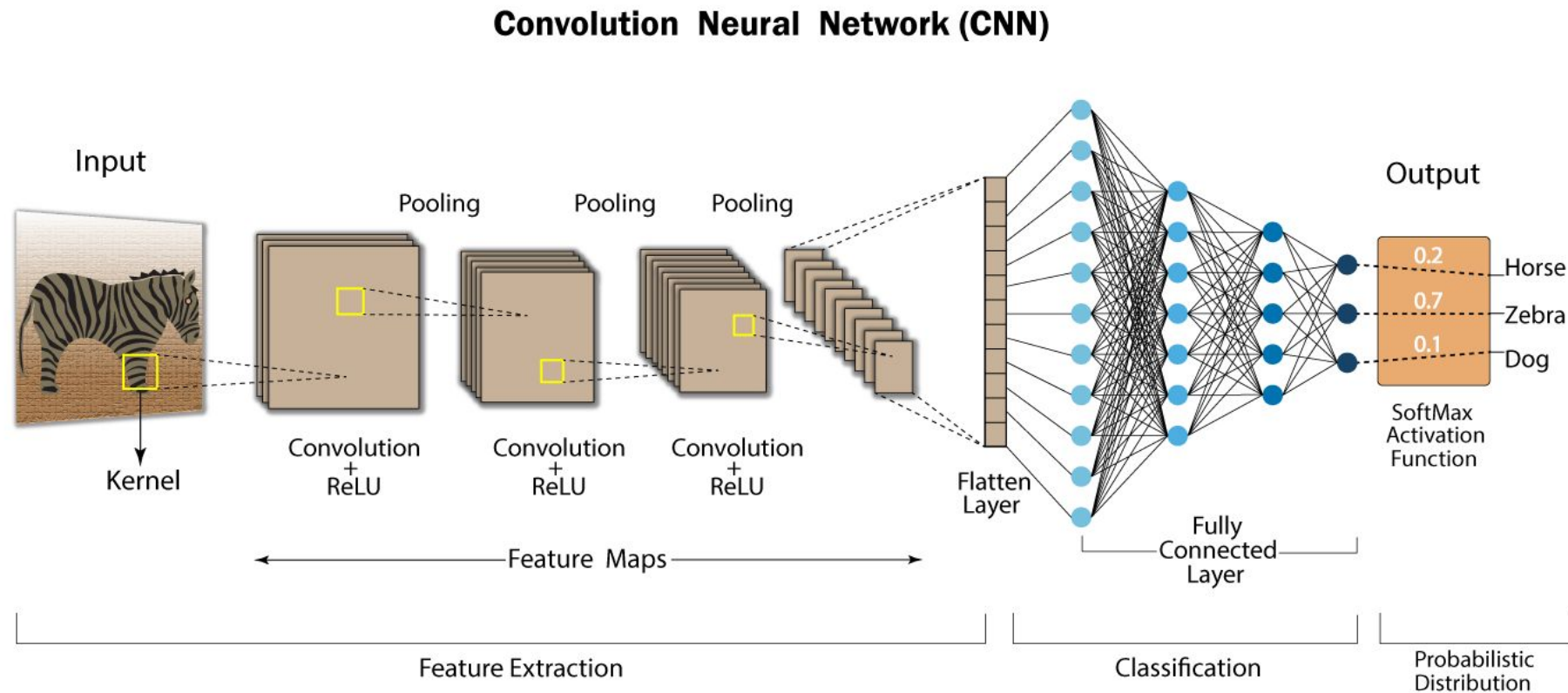
- Se aplica una función de activación ReLU después de cada operación de convolución.
- Esta función ayuda a la red a aprender relaciones no lineales entre las características de la imagen, lo que hace la red más robusta para identificar distintos patrones.



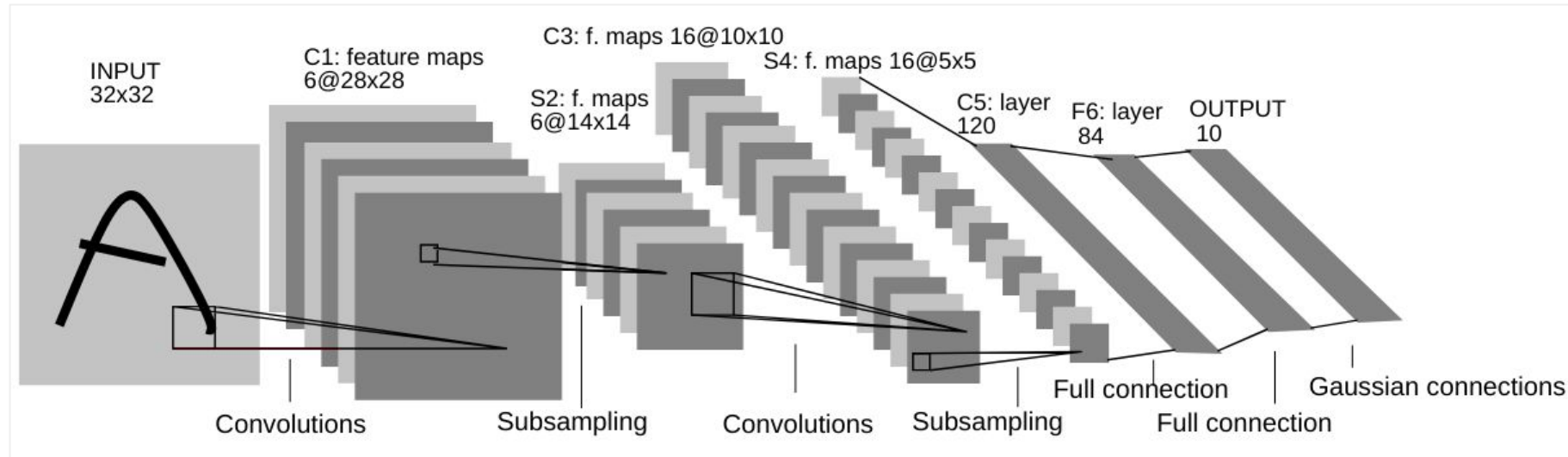


Time to Code

Arquitecturas de redes neuronales convolucionales



Arquitecturas de redes neuronales convolucionales: LeNet

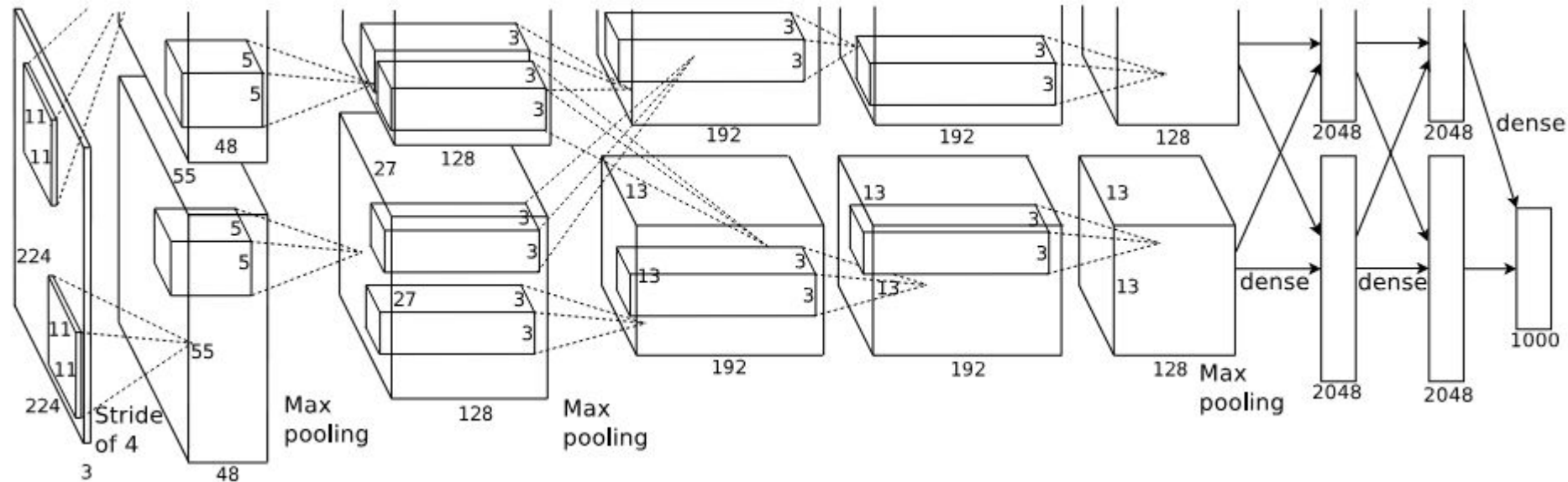


Total de parámetros: 60,000

Características:

- Propuesta por LeCun et al. en 1998.
- Conjunto de entrenamiento: imágenes de dígitos a mano (MNIST).
- Entrada: imágenes de tamaño 32 x 32 en escala de grises.
- Salida: 10 clases (una por cada dígito)

Arquitecturas de redes neuronales convolucionales: AlexNet

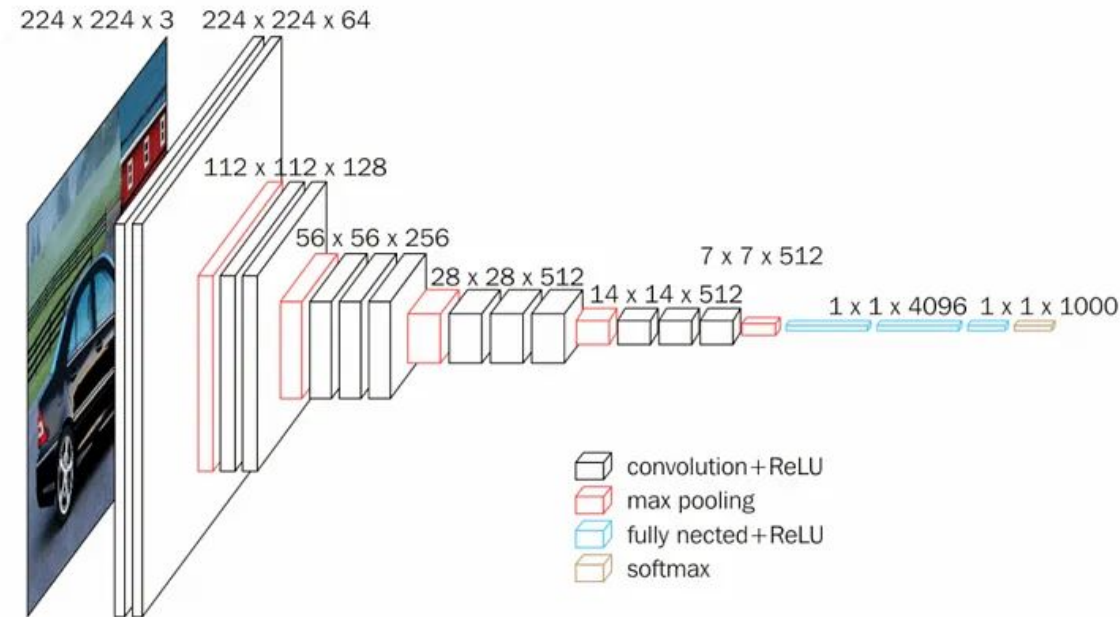


Características:

- Propuesta por Krizhevsky et al. en 2012.
- Conjunto de entrenamiento: ImageNet LSVRC-2010 (1.2 millones de imágenes).
- Entrada: imágenes de tamaño 227 x 227 a color.
- Salida: 1000 clases.

Total de parámetros: 62.3 millones

Arquitecturas de redes neuronales convolucionales: VGGNet-16



Características:

- Propuesta por Visual Geometry Group en el 2014.
- Conjunto de entrenamiento: ImageNet (14 millones de imágenes).
- Entrada: imágenes de tamaño 224 x 224 a color.
- Salida: 1000 clases.

Total de parámetros: 138.4 millones

Total de capas: 16 capas (13 convolucionales y 3 completamente conectadas).



Time to Code

Sitios importantes a considerar

- [Hugging Face](#)
- [Pytorch](#)
- [Vision group](#)
- [SAM](#)
- [Robotic Instrument Segmentation Sub-Challenge](#)

Referencias

- Zhang A, Lipton Z, Li M, and Smola J. Dive into Deep Learning. 2020. Disponible en <https://d2l.ai/>
- Murphy, K. P. (2022). Probabilistic Machine Learning: An introduction. MIT Press. Capítulo 8, 10 y 11. Disponible en <https://probml.github.io/pml-book/book1.html>
- Nielsen, M. (2019). Neural Networks and Deep Learning. Capítulo 1. Disponible en <http://neuralnetworksanddeeplearning.com/index.html>

Contacto

Dra. Blanca Vázquez

blanca.vazquez@iimas.unam.mx

<https://github.com/blancavazquez>

