

Introducción a Machine Learning

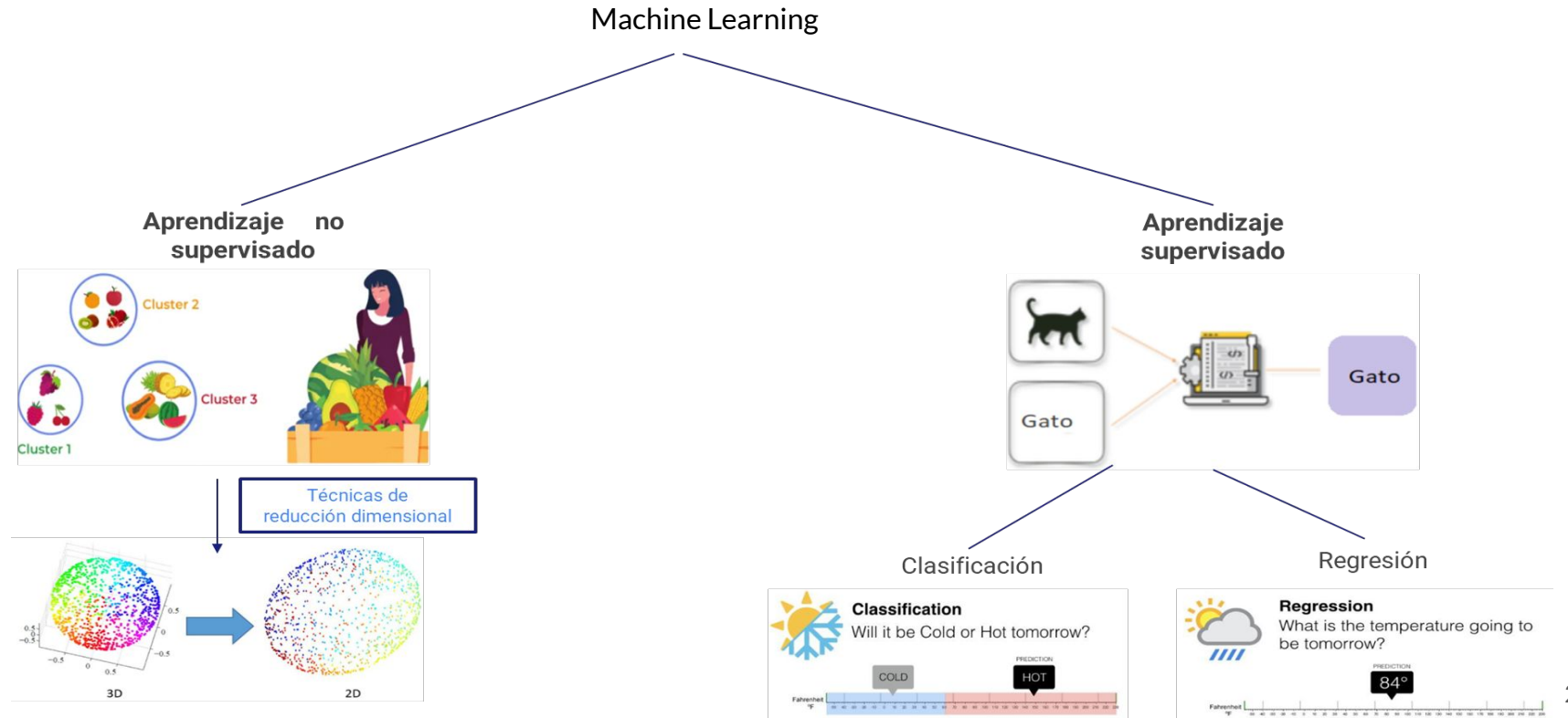
Contenido

- Redes Convolucionales
- Data augmentation
- Transfer learning
- Proyecto de Reconocimiento Facial

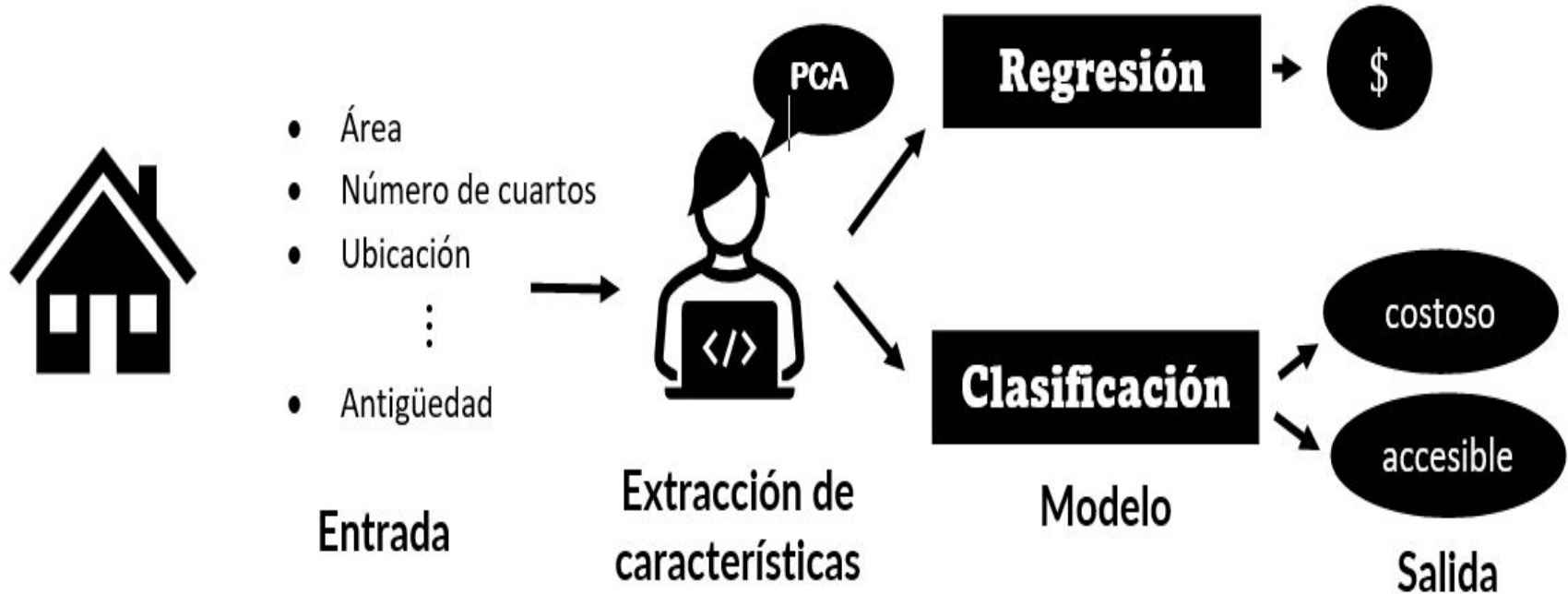


16/Octubre/2023

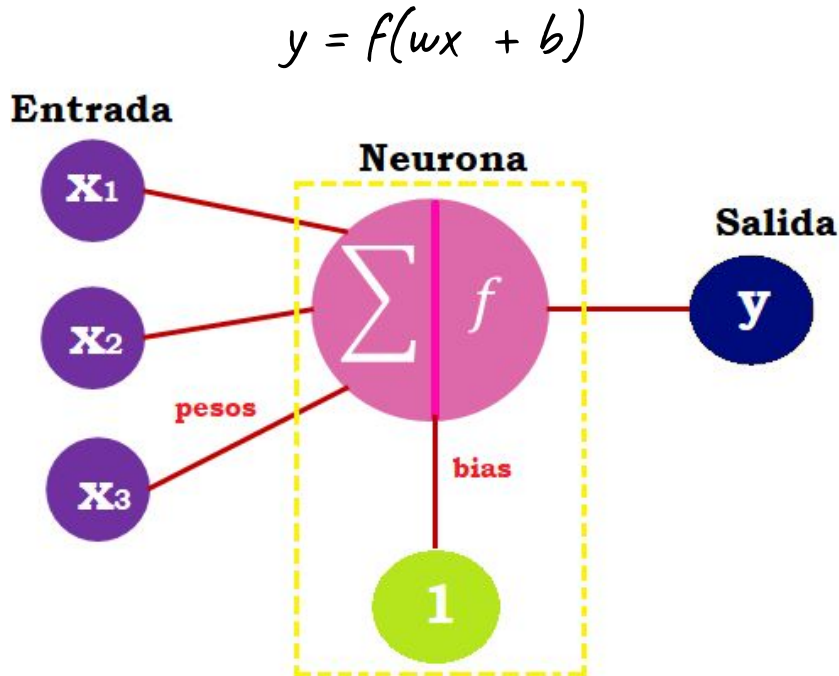
Machine Learning (recapitulación)



Machine Learning (recapitulación)



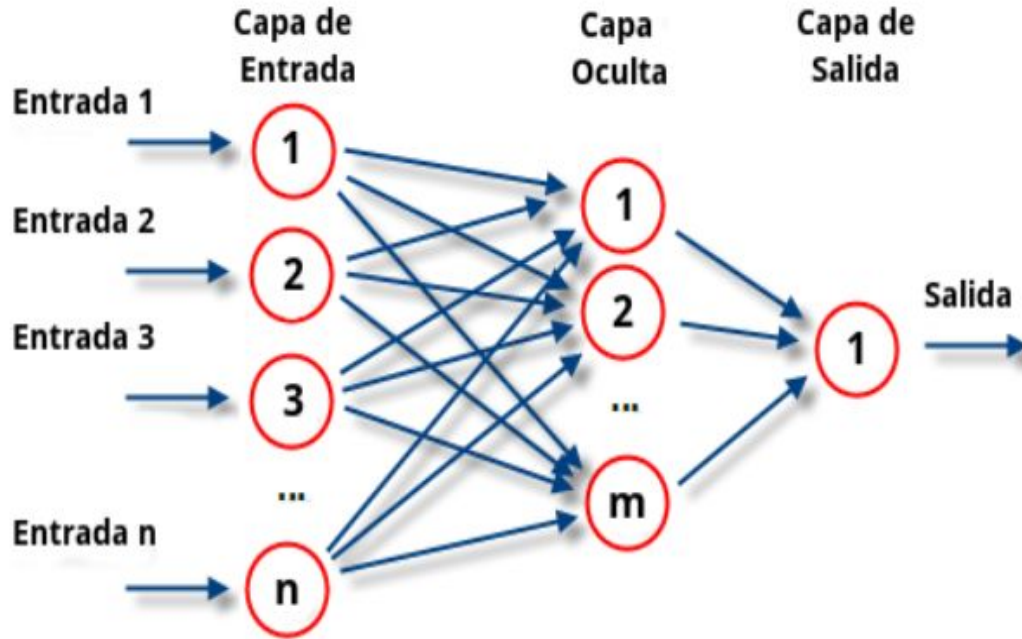
Redes Neuronales (recapitulación)



Una neurona artificial es una función matemática que toma una o más valores de entrada y regresa un valor numérico.

- x - Representa la entrada de los datos.
- w - Los pesos son valores que representan la intensidad de las entradas o la fuerza entre las conexiones entre las neuronas.
- b - Bias es una entrada artificial que siempre tiene por entrada el valor de 1.
- $f(n)$ - Función de activación determina cuando se produce una señal de salida.
- y - Salida de la red.

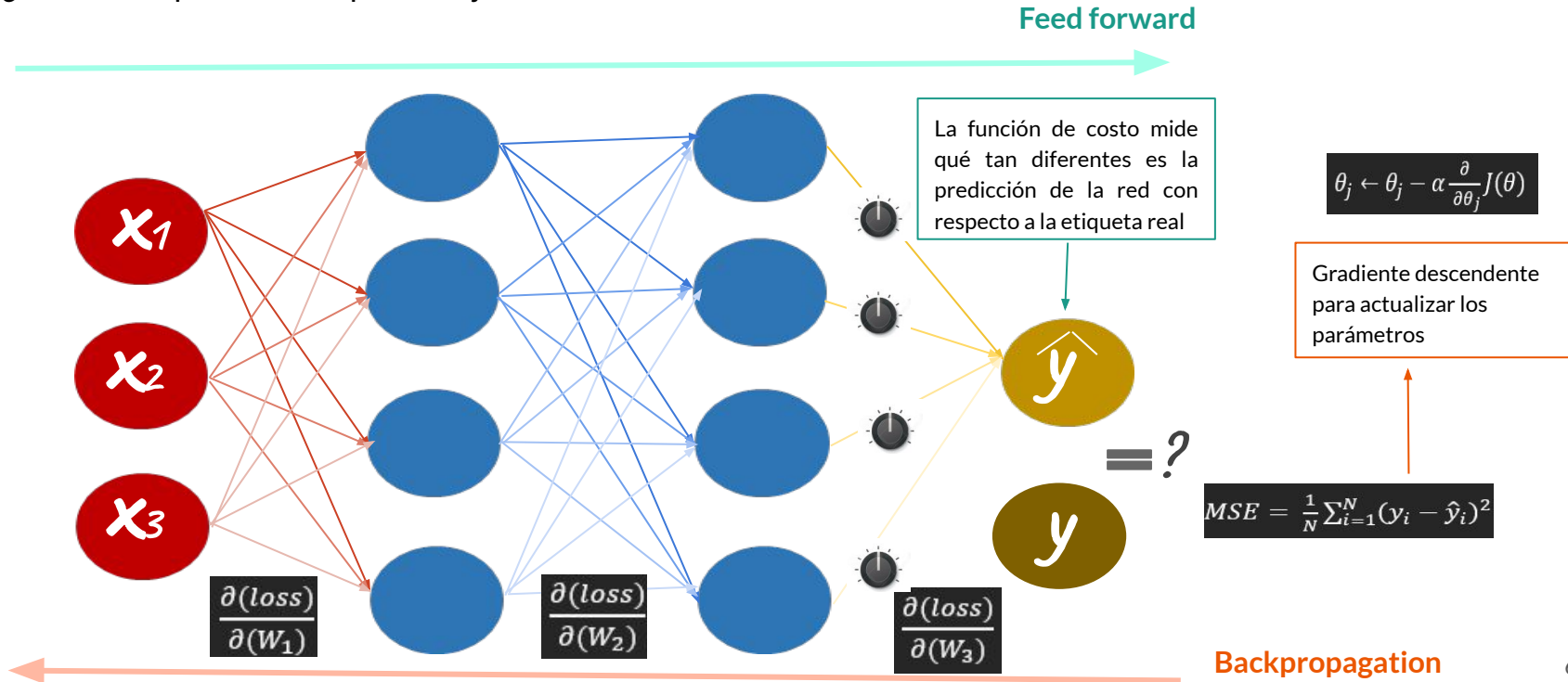
Redes Neuronales (recapitulación)



- **Capa de entrada:** La cantidad de neuronas en esta capa es igual a la cantidad total de características en nuestros datos.
- **Capas ocultas:** Capas intermedias. Puede haber muchas capas ocultas según el modelo y la cantidad de datos. Cada capa oculta puede tener diferentes números de neuronas.
- **Capa de salida:** La salida de la red neuronal. La cantidad de neuronas dependerá del problema.

Redes Neuronales (recapitulación)

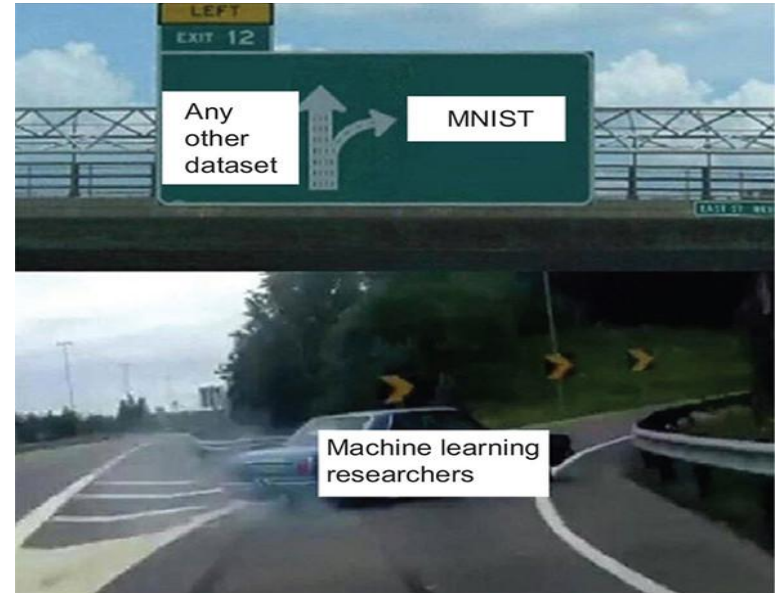
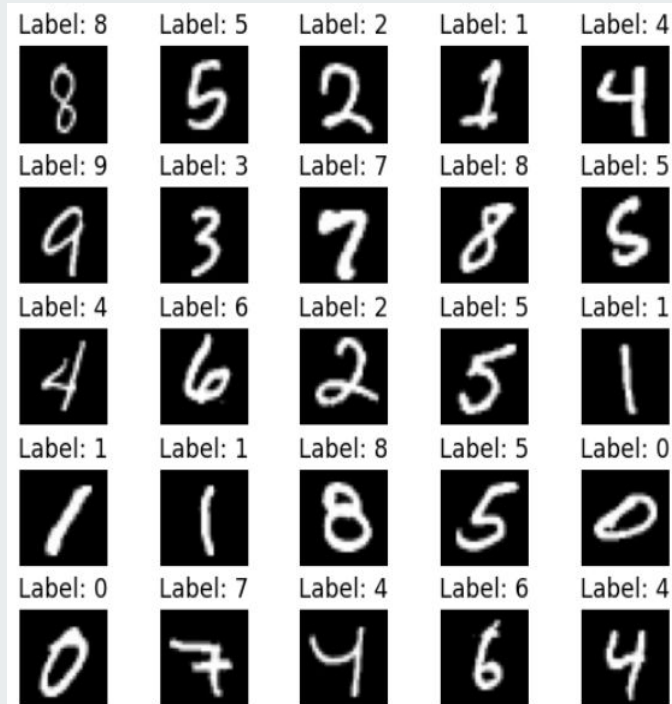
¿Cómo es el proceso de aprendizaje?



Actividad 1:

Red Neuronal Multicapa para Clasificación de
Imágenes de MNIST con Pytorch

 [MLP de MNIST](#)



¿Porque las MLP no es la mejor opción para clasificar imágenes?

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 |

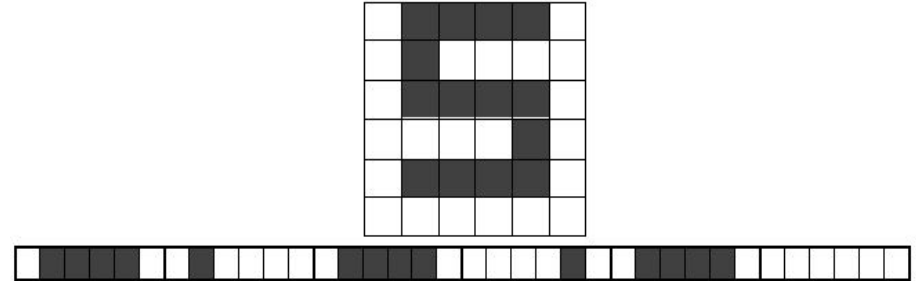
28x28x1

Flatten

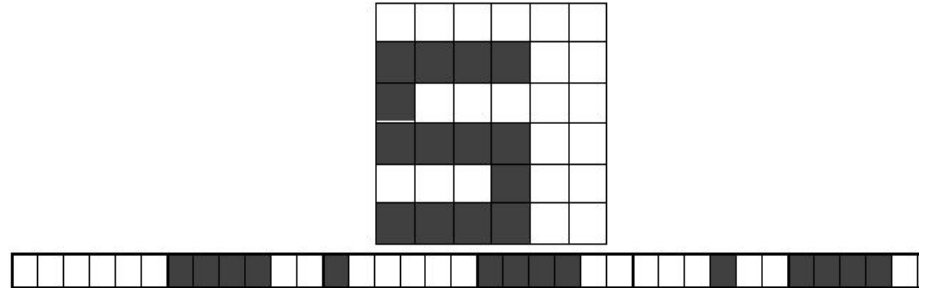


784

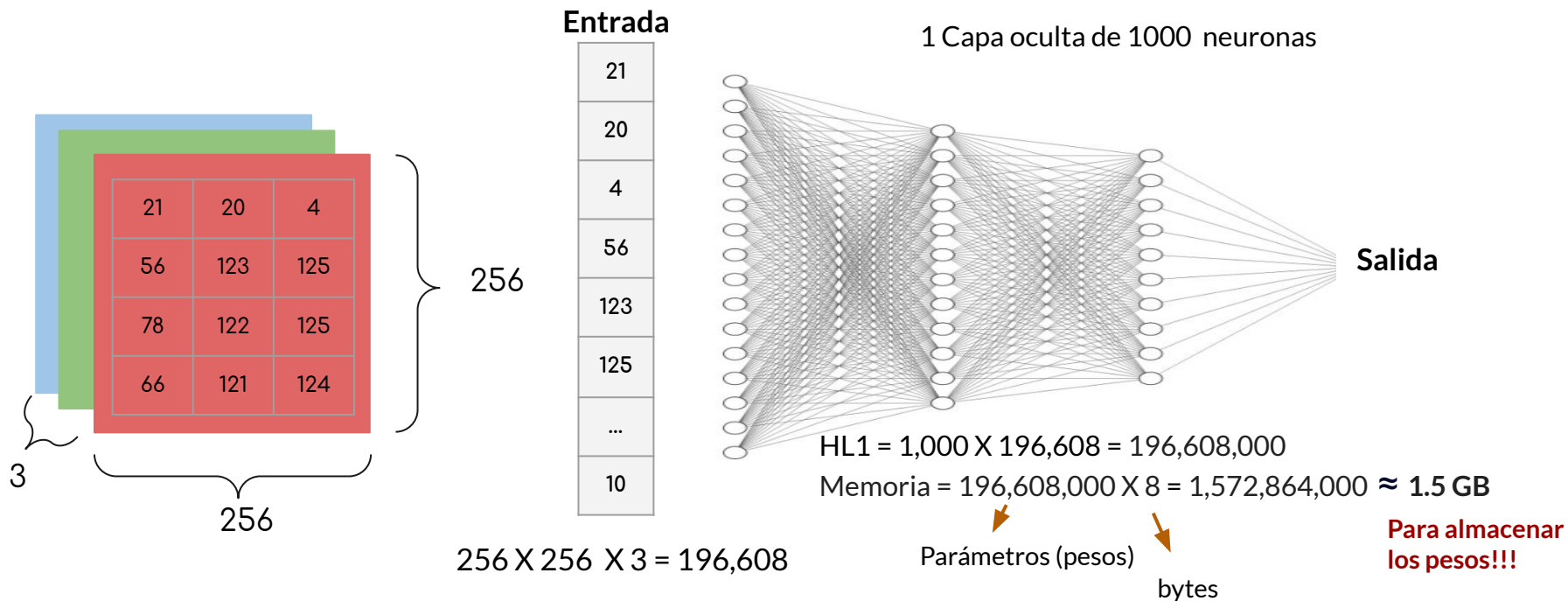
Al transformar las imágenes a vectores, se pierde la estructura espacial de las imágenes



Vectores muy diferentes para imágenes iguales



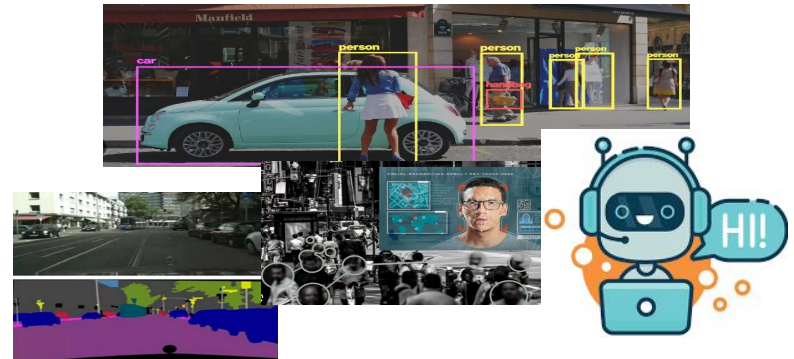
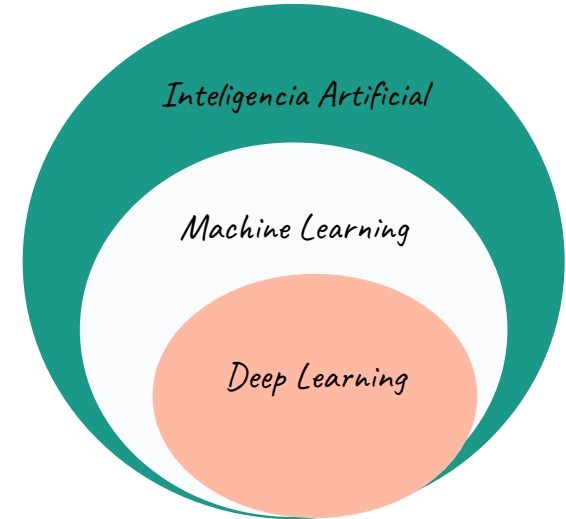
¿Porque las MLP no es la mejor opción para clasificar imágenes?



Deep Learning

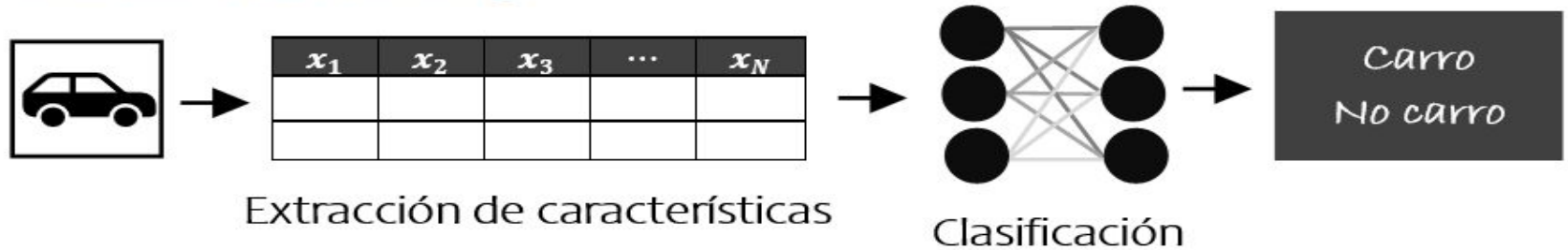
El aprendizaje profundo es un subconjunto del machine learning (aprendizaje automático) basado en redes neuronales artificiales.

Se llama aprendizaje *profundo* porque la estructura de redes neuronales artificiales se compone de varias capas ocultas. Cada capa contiene unidades que transforman los datos de entrada en información que la capa siguiente puede usar para realizar una tarea de predicción determinada. Gracias a esta estructura, una máquina puede aprender a través de su propio procesamiento de datos.



Deep Learning vs Machine Learning

Machine Learning



Deep Learning



Redes Neuronales Convolucionales



Una red neuronal convolucional (CNN) o convnet es un tipo de arquitectura de Deep Learning comúnmente utilizada en aplicaciones de visión por computadora.

El desarrollo de las CNN se remonta a la década de 1990 , cuando Yann LeCun et al. propusieron una novedosa arquitectura NN para clasificar dígitos escritos a mano a partir de imágenes (Handwriting Digit Recognition with a Back-Propagation Network,

Handwritten Digit Recognition with a Back-Propagation Network

Y. Le Cun, B. Boser, J. S. Denker, D. Henderson,
R. E. Howard, W. Hubbard, and L. D. Jackel
AT&T Bell Laboratories, Holmdel, N. J. 07733

ABSTRACT

We present an application of back-propagation networks to handwritten digit recognition. Minimal preprocessing of the data was required, but architecture of the network was highly constrained and specifically designed for the task. The input of the network consists of normalized images of isolated digits. The method has 1% error rate and about a 9% reject rate on zipcode digits provided by the U.S. Postal Service.

Redes Neuronales Convolucionales

Las CNN son una familia de modelos que se inspiraron originalmente en cómo funciona la corteza visual del cerebro humano al reconocer objetos.

El descubrimiento original de cómo funciona la corteza visual de nuestro cerebro fue realizado por David H. Hubel y Torsten Wiesel en 1959 en donde descubrieron que existen diferentes capas de la corteza visual.

Las capas primarias detectan principalmente bordes y líneas rectas, las capas de orden superior se centran más en extraer formas y patrones complejos.

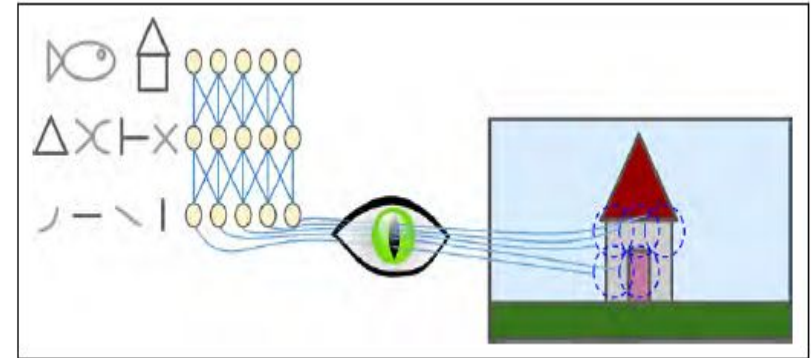
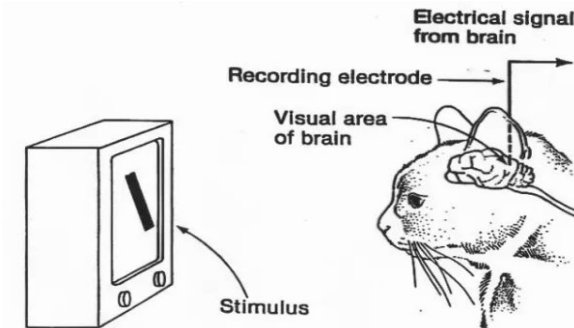
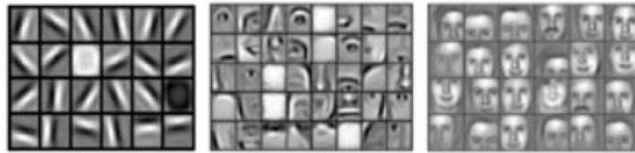
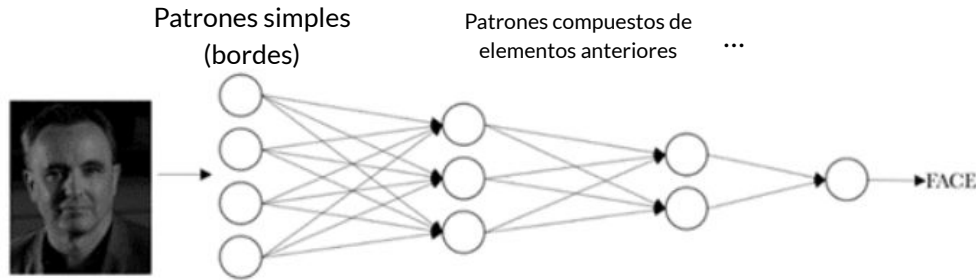


Imagen extraída de [hands on



Experimento
de Hubel and
Wiesel

Redes Neuronales Convolucionales (CNN)



Low-level features

Mid-level features

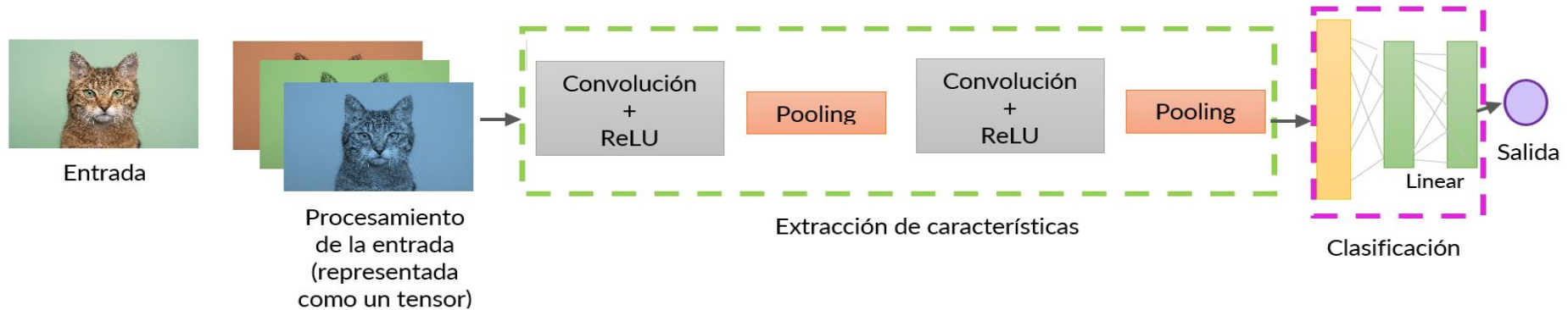
High-level features

Aprende jerarquías espaciales de patrones



Los patrones que aprende son invariantes a la traslación

Redes Neuronales Convolucionales (CNN)



Se componen de tres tipos principales de capas:

- Capa convolucional
- Capa pooling
- Capa completamente conectada

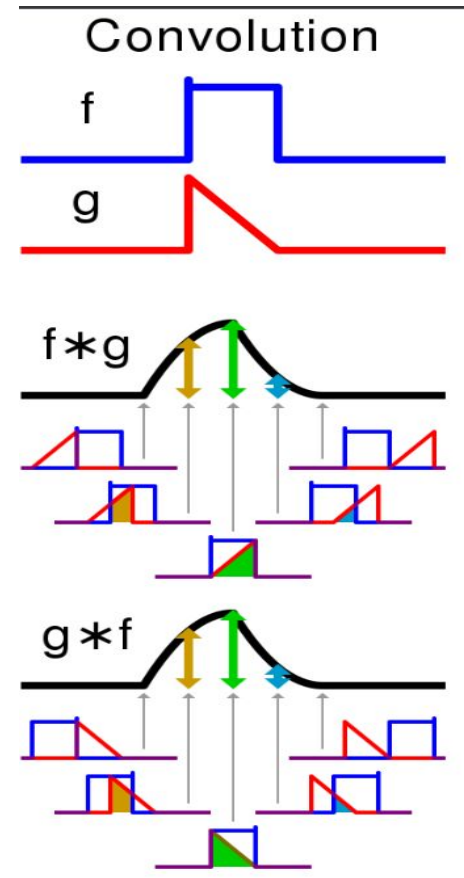
- Red Neuronal Convolucional
- CNN
- Convnet

Capa convolucional

Componente principal de una *convnet*.

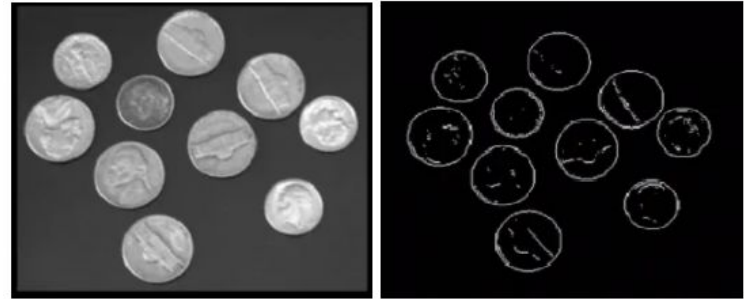
Una convolución es un operador matemático que transforma dos funciones f y g en una tercera función que muestra cómo la forma de una función está siendo influenciada y modificada por la otra función.

En las CNN, una convolución es el producto punto entre dos matrices (el kernel y una pequeña región de la imagen de entrada) para producir una tercera matriz.

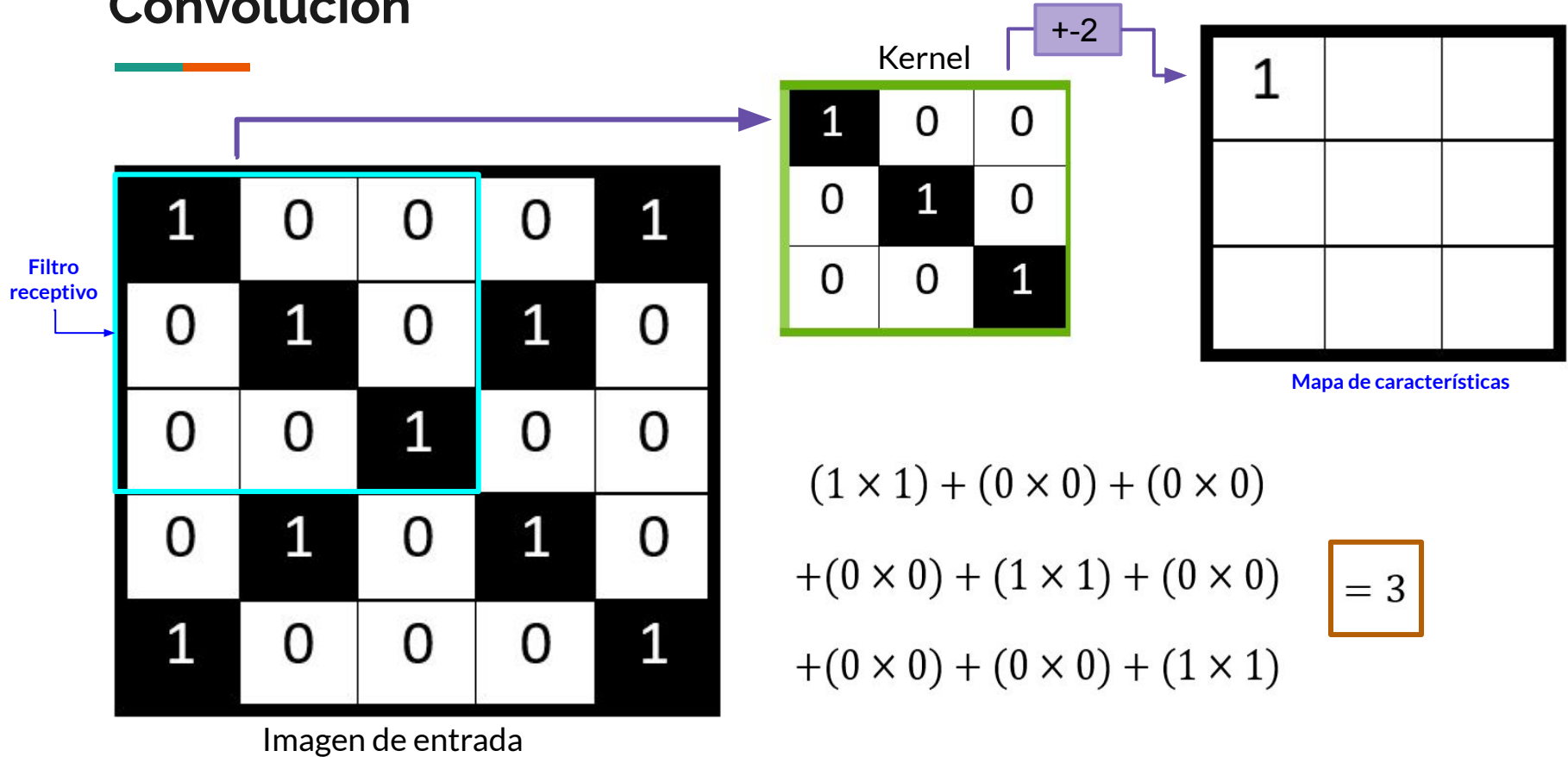


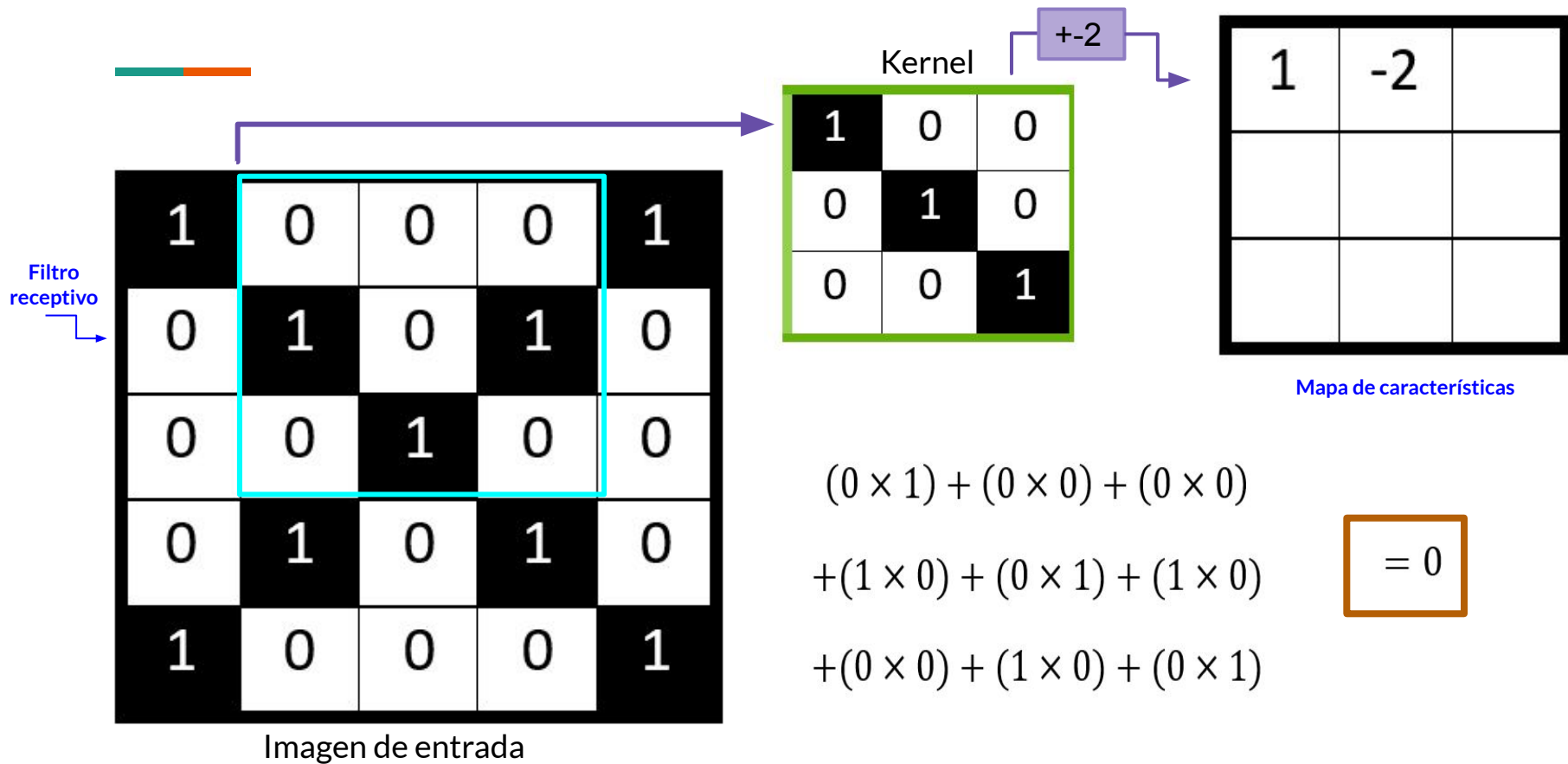
Kernel

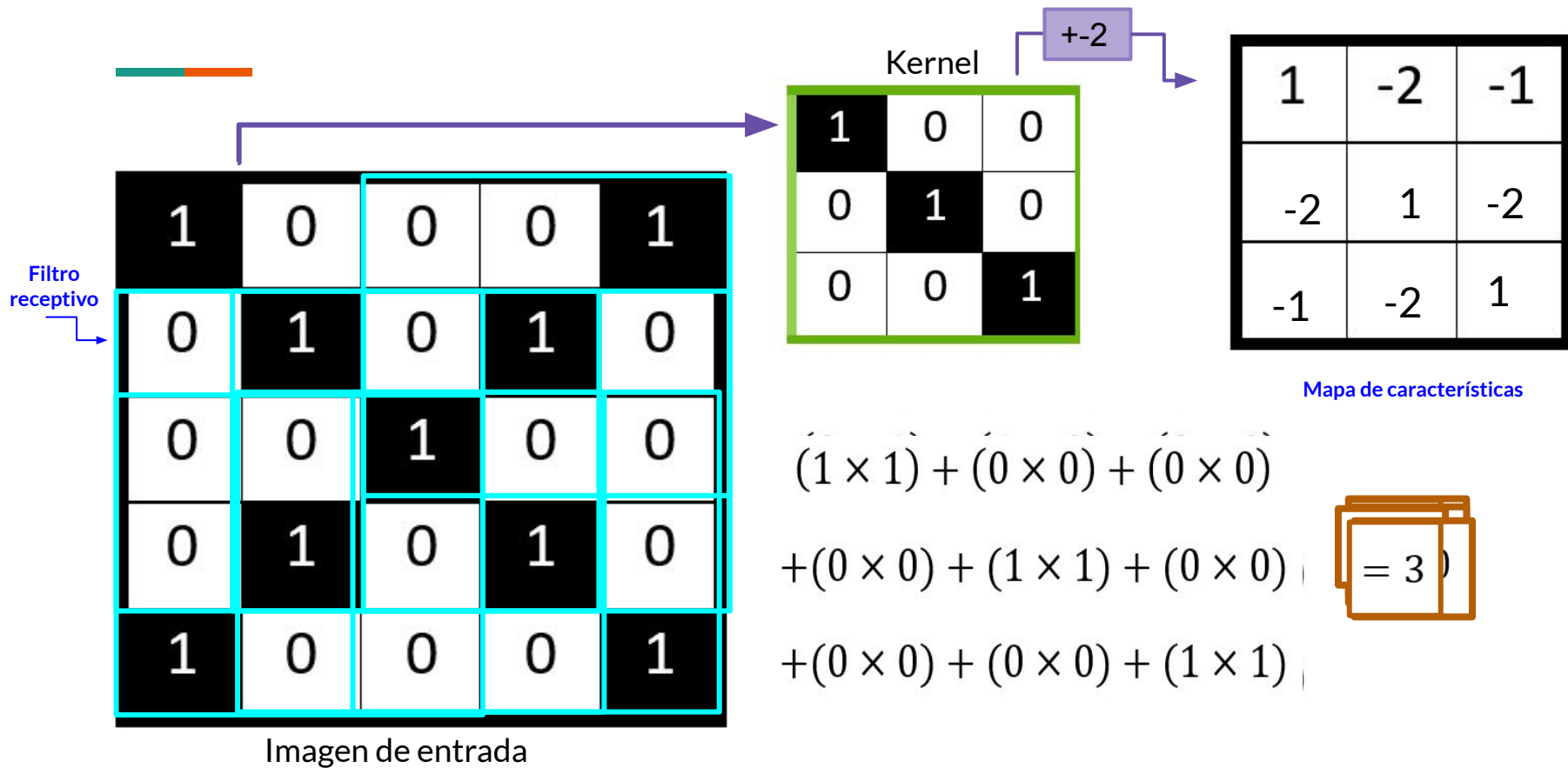
El kernel es un tipo de filtro que se aplica a una imagen para extraer ciertas características importantes o patrones.



Convolución







CO Convolución

Ver:
[Image Kernels](#)

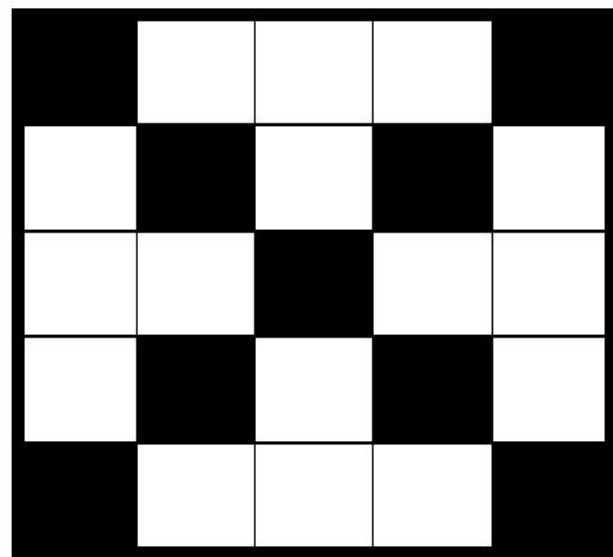
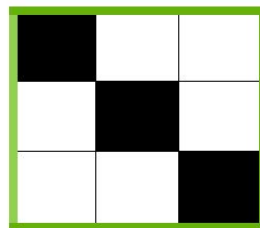


Imagen de entrada

Kernel



El kernel desempeña el papel de detector de características

+2

Mapa de características

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

ReLU






Los mapas de características indican las regiones donde se han detectado características específicas del kernel en la entrada.

| | | |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

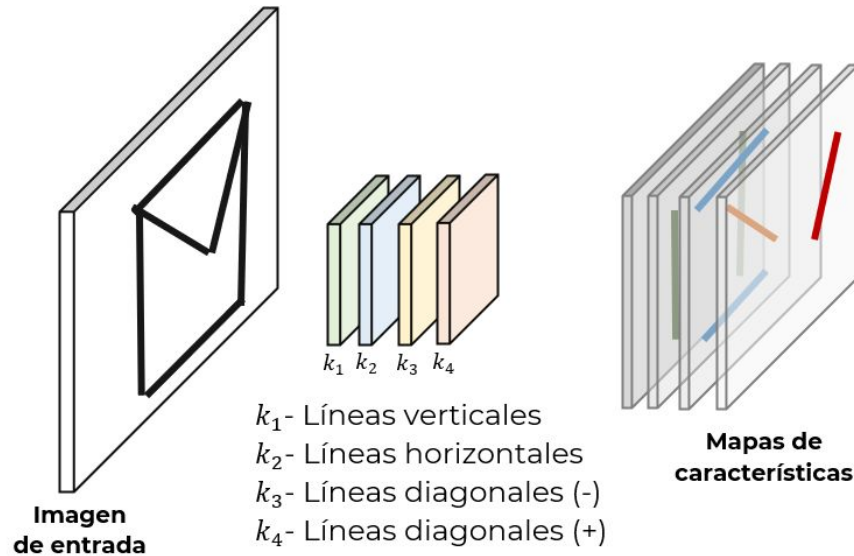
¿Cómo se escogen los valores del kernel?

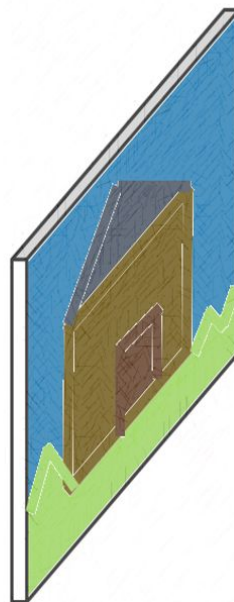
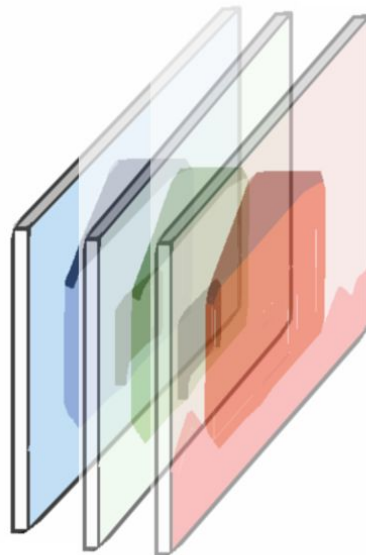
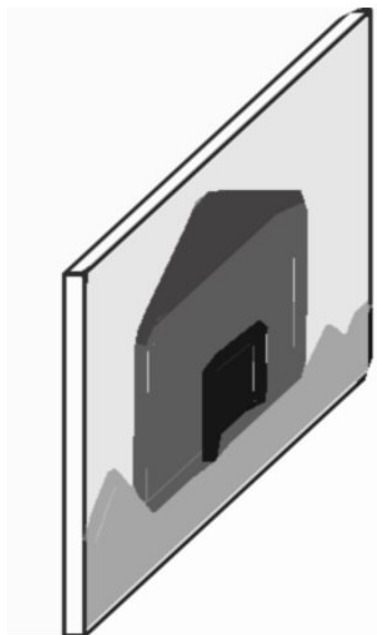
Al entrenar una CNN no se configuran los pesos del kernel manualmente. Estos valores son aprendidos por la red.

La red aprende a ajustar sus pesos (a partir de valores aleatorios), permite que el modelo genere los filtros adecuados capaces de adaptarse al problema y al conjunto de datos.

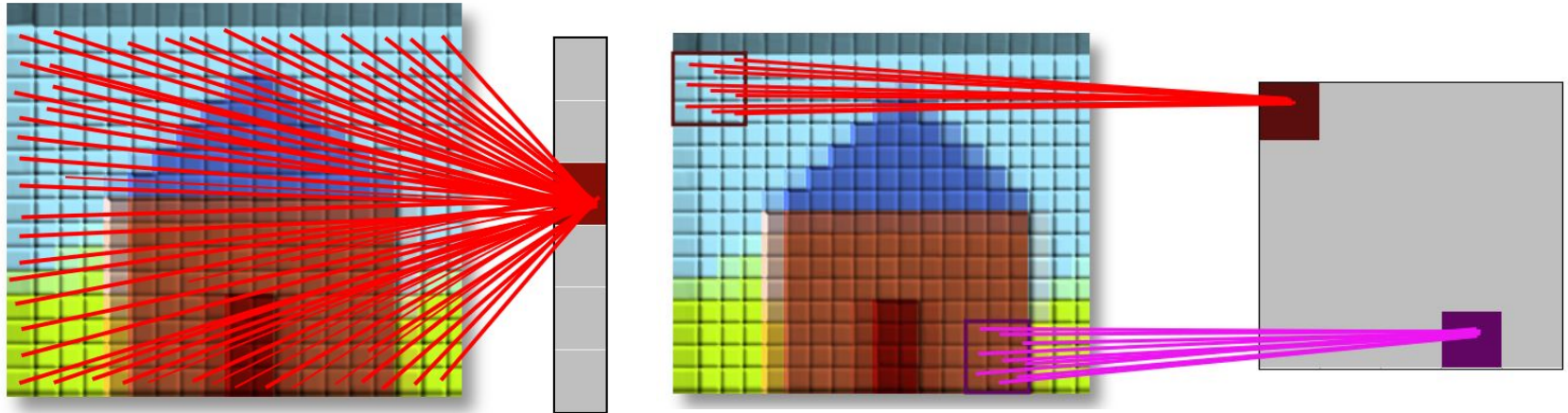
| Nombre | Kernel | Imagen resultante |
|------------|--|--|
| Identidad | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |  |
| Laplaciano | $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ |  |
| Desenfoque | $\begin{bmatrix} 1/16 & 1/16 & 1/16 \\ 1/16 & 1/16 & 1/16 \\ 1/16 & 1/16 & 1/16 \end{bmatrix}$ |  |

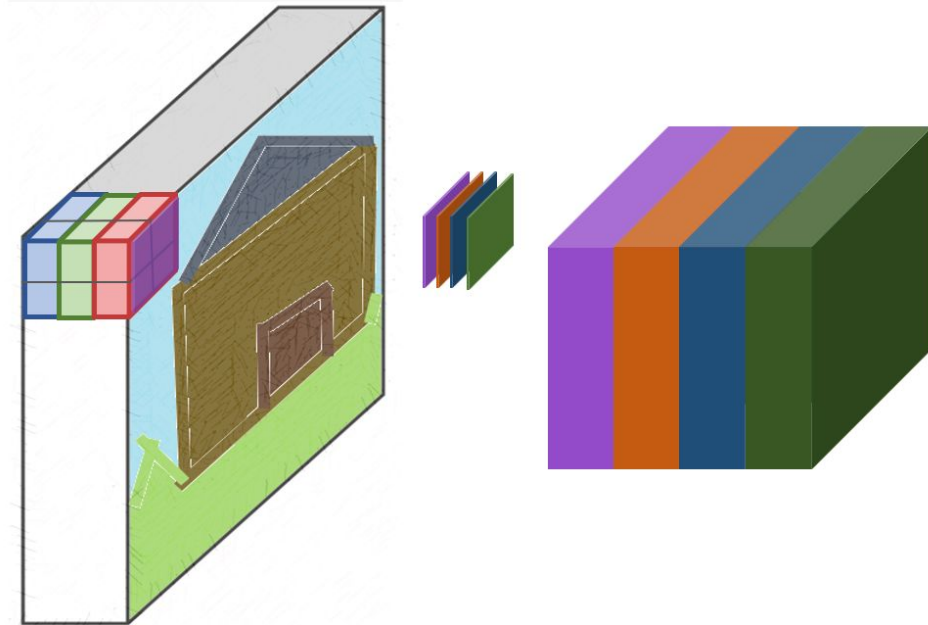
Las capas convolucionales suelen contener muchos filtros, lo que significa que cada capa convolucional produce múltiples mapas de activación. La idea es generar diferentes mapas de características que sean capaces de localizar diferentes características específicas en la imagen.





<https://medium.com/@mohamedhedifkir/convolutional-neural-network-23a6acb08d6a>

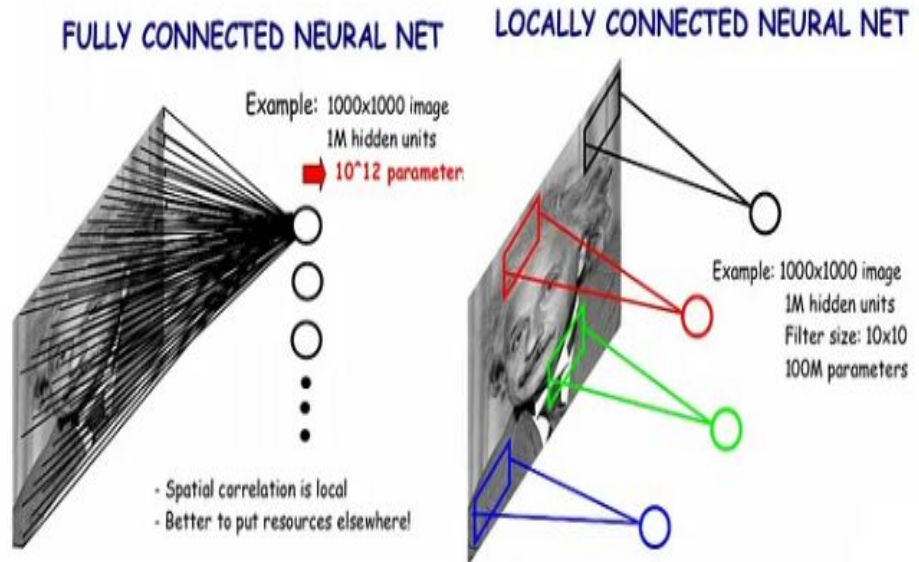




Capa convolucional

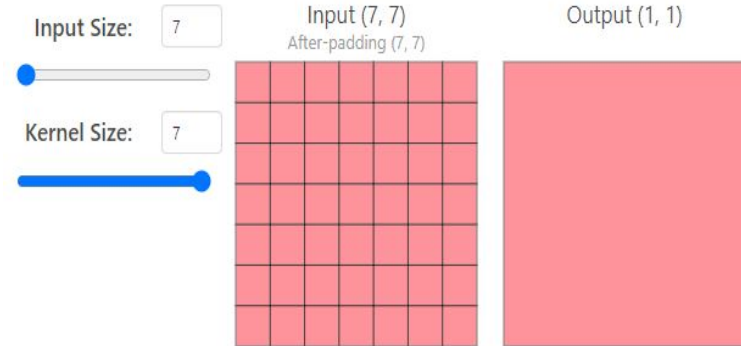
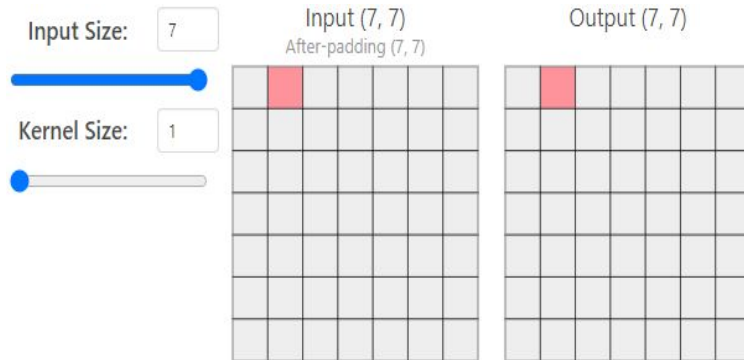
La conectividad local es cuando cada neurona está conectada solo a un subconjunto de la imagen de entrada (a diferencia de una red neuronal donde todas las neuronas están completamente conectadas).

Esto ayuda a reducir la cantidad de parámetros en todo el sistema y hace que el cálculo sea más eficiente.



Tamaño del kernel

Tanto el tamaño del kernel como la cantidad de filtros son hiperparámetros muy importantes pues determinan que tanta información se extrae y cuantos mapas de activación se producen en la salida

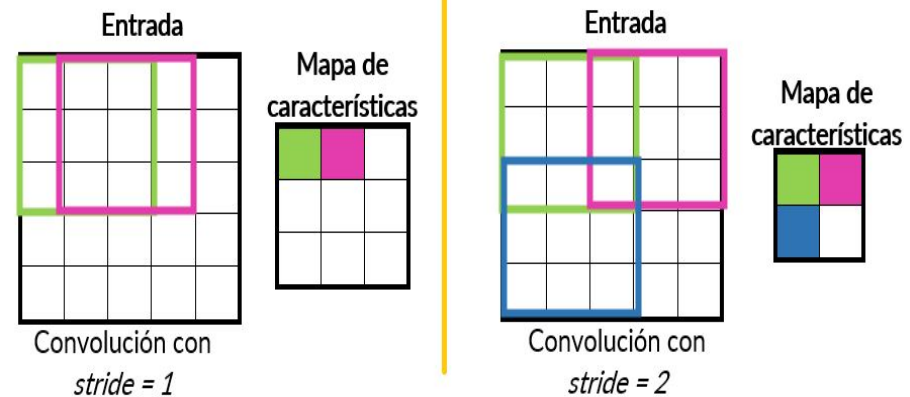


Hiperparámetros de capa convolucional: STRIDE

El “stride” es un hiperparámetro que indica el tamaño de paso para mover el kernel a través de la entrada.

Si se aumenta el stride, se reducirá el tamaño del tensor de salida.

Esto es útil para reducir el tamaño espacial del tensor a medida que pasa a través de la red.



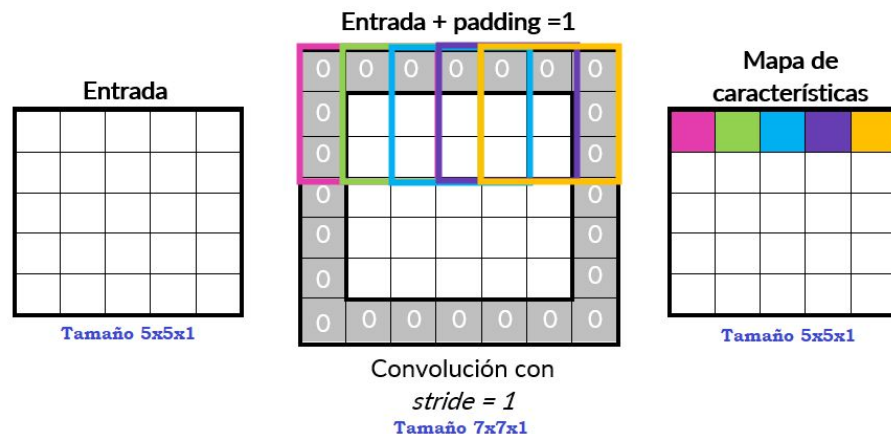
$$Dim_{map\ features} = \left(\frac{Dim_{input} - Dim_{kernel}}{stride} \right) + 1$$

Ver:
[CNN explainer](#)

Hiperparámetros de capa convolucional: PADDING

El “padding” se aplica agregando píxeles de valor cero alrededor de la imagen original.

Lo que permite que, al realizar la convolución, el mapa de características sea de igual tamaño que la entrada.



$$Dim_{map\ features} = \left(\frac{Dim_{input} - Dim_{kernel} + (2 \times padding)}{stride} \right) + 1$$



Capas Pooling

Pooling es una técnica para generalizar características extraídas por los kernels y ayudar a la red a reconocer características independientemente de su ubicación en la imagen.

Reducen el tamaño espacial del mapa de características, lo que disminuye la cantidad requerida de cálculos y pesos.

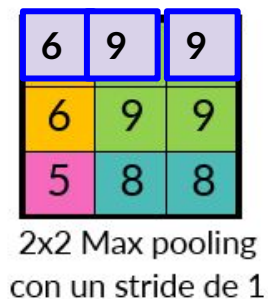
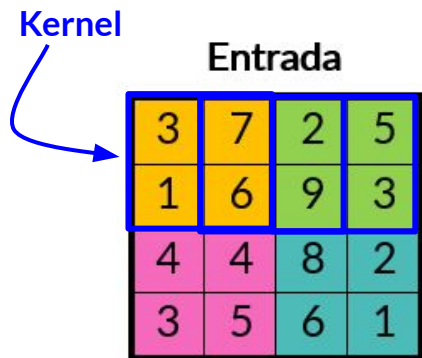
Tipos de pooling más comunes:

- Max-pooling:
- Average pooling

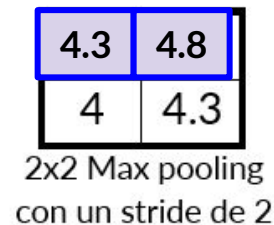
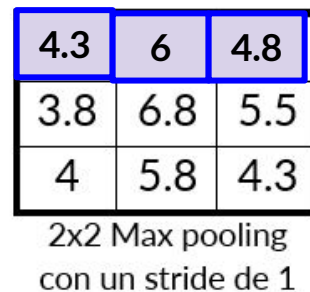
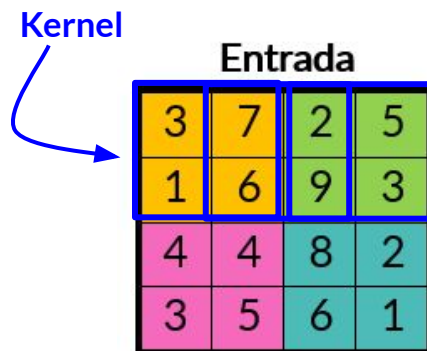
Se realiza para cada mapa de características

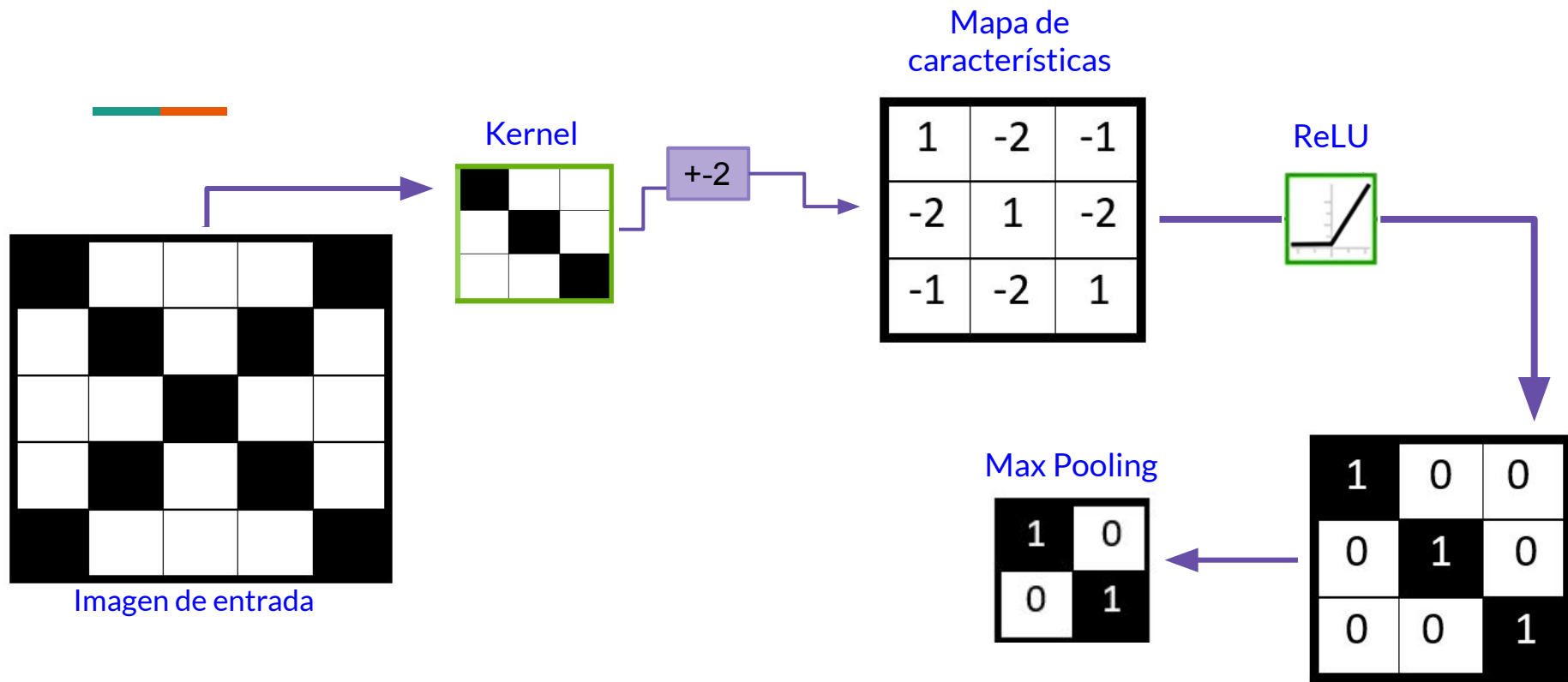
Capas Pooling

Max Pooling

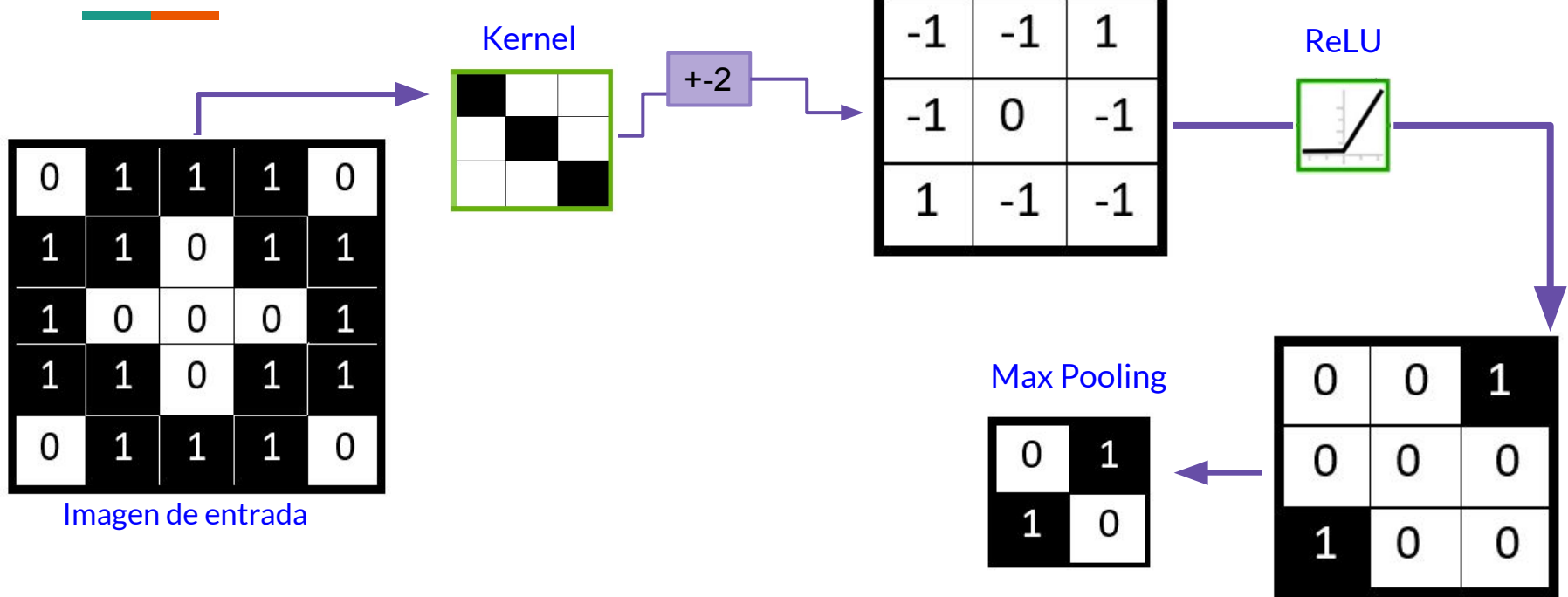


Average Pooling





Pooling

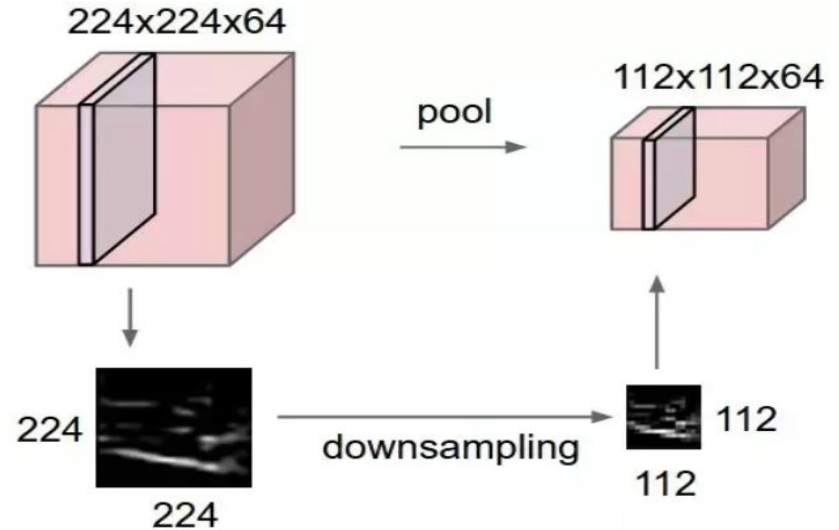


Max-pooling:

Reduce el tamaño de los mapas de características por lo tanto la cantidad de parámetros entrenables se reducen, ayudándonos a controlar el overfitting.

Condensa los mapas de características conservando las características más importantes.

Hace que la red sea invariante a pequeñas transformaciones, distorsiones y traducciones en la imagen de entrada (una pequeña distorsión en la entrada no cambiará la salida de Pooling, ya que tomamos el valor máximo en un vecindario local)

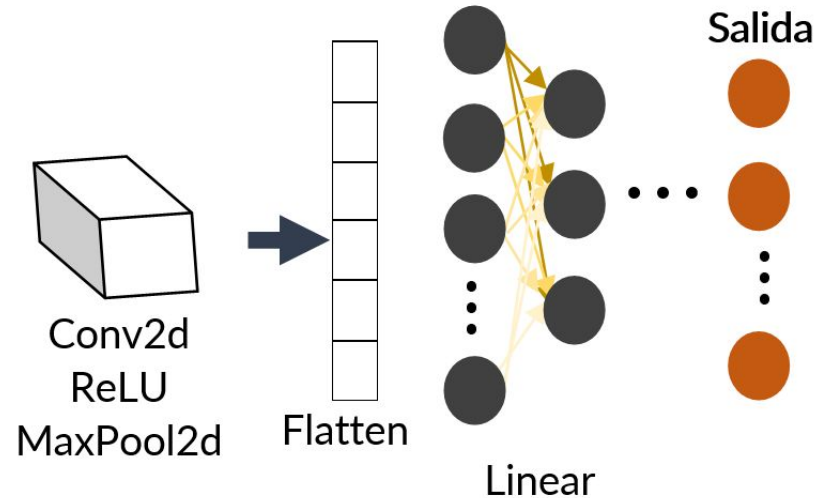



Capa Completamente Conectada


























Las capas completamente conectadas desempeñan el papel de la tarea de clasificación, mientras que las capas anteriores actúan como extractores de características.

Las capas completamente conectadas toman como entrada el arreglo resultante de las capas convolucionales y las capas pooling para generar un vector que combina estos elementos y así poder realizar una clasificación.

Además de la clasificación, agregar una capa completamente conectada también es una forma de aprender combinaciones no lineales de estas características.



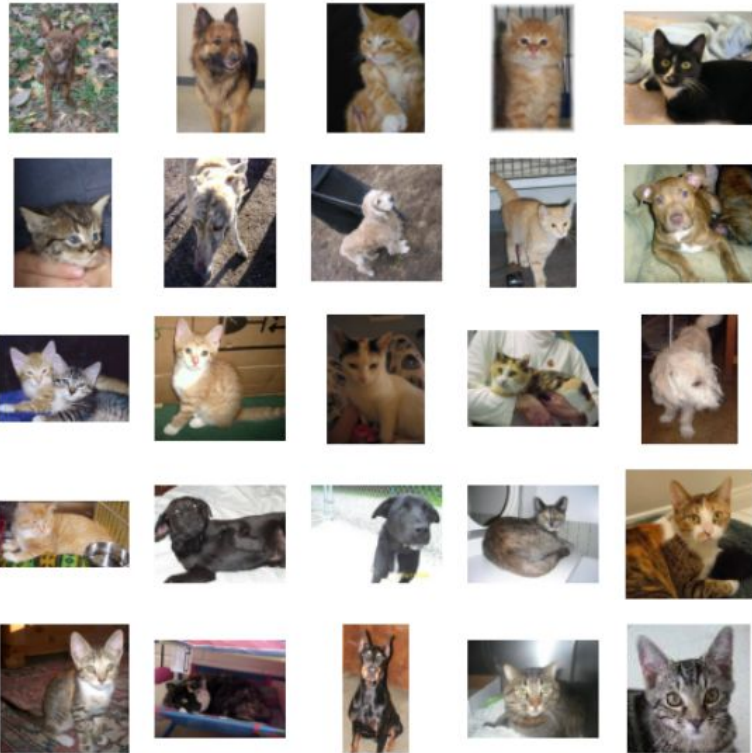


| | | | | |
|--|--|--|--|--|
| Label: 8  | Label: 5  | Label: 2  | Label: 1  | Label: 4  |
| Label: 9  | Label: 3  | Label: 7  | Label: 8  | Label: 5  |
| Label: 4  | Label: 6  | Label: 2  | Label: 5  | Label: 1  |
| Label: 1  | Label: 1  | Label: 8  | Label: 5  | Label: 0  |
| Label: 0  | Label: 7  | Label: 4  | Label: 6  | Label: 4  |

Actividad 2:

Red Neuronal Convolutacional para Clasificación
de Imágenes de MNIST con Pytorch

[CNN de MNIST](#)



Ejercicio 1:

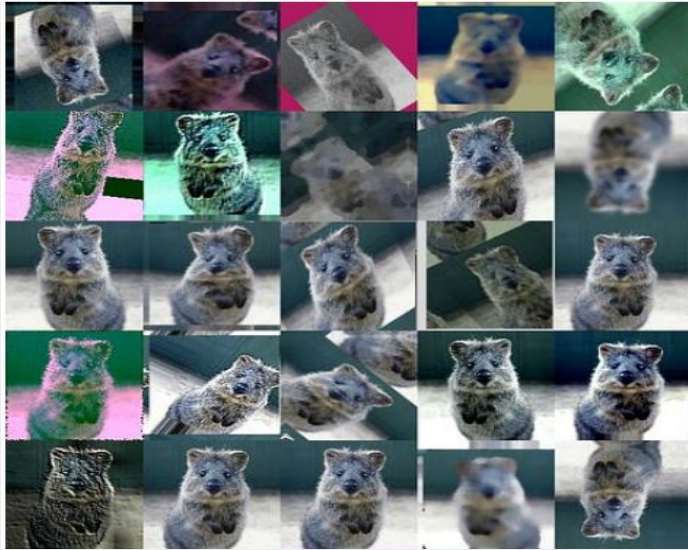
Red Neuronal Convolutacional para Clasificación de perros y gatos con Pytorch.

People with no idea about AI saying it will take over the world:

My Neural Network:



Data augmentation



El aumento de datos es una técnica que se utiliza para aumentar la diversidad del conjunto de datos de entrenamiento mediante la aplicación de varias transformaciones a los datos existentes.

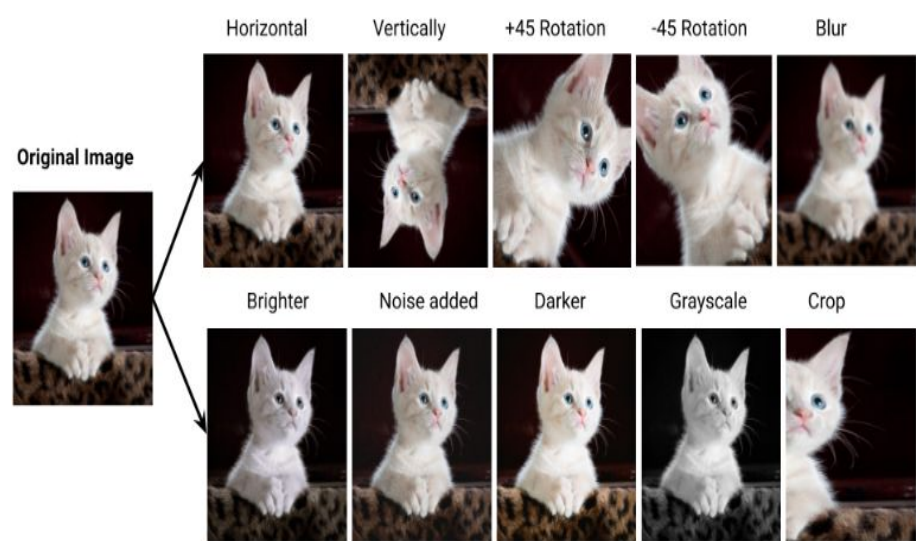
La idea detrás del aumento de datos es crear nuevas variaciones de los datos de entrenamiento que puedan ayudar al modelo a generalizar mejor.

¿Por qué utilizar el aumento de datos?

1. Reducir el sobreajuste. Al aumentar los datos de entrenamiento, podemos introducir más variaciones en el conjunto de datos.

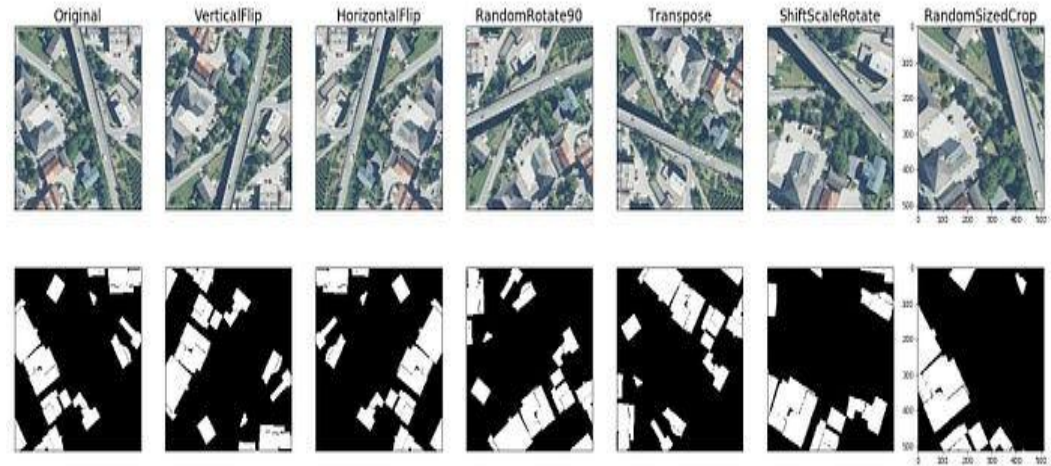
2. Aumentar el tamaño del conjunto de datos. Al aumentar el conjunto de datos, podemos generar más datos a partir de los datos de entrenamiento existentes, lo que puede ayudarnos a crear un conjunto de datos más grande y diverso.

3. Desbalance de clases. El desbalance de clases es una situación en la que una o más clases en el conjunto de datos tienen significativamente menos muestras que las otras clases. Al aumentar los datos, podemos aumentar el número de muestras en la clase minoritaria, reduciendo así el desequilibrio de clases.

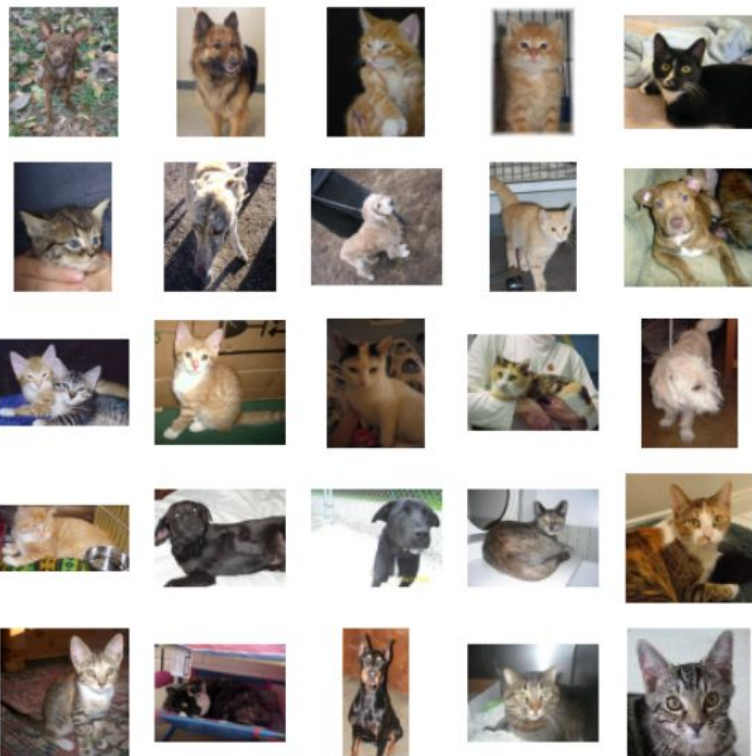


Limitaciones del aumento de datos

- Si se tiene algún tipo de sesgo en el conjunto de datos original, el sesgo persistirá en los datos aumentados.
- Encontrar un enfoque eficaz de aumento de datos puede resultar un desafío, en algunos casos si no se selecciona correctamente el método para aumentar datos, pudiera afectar el desempeño del modelo



La selección de las transformaciones para aumentar datos dependerá mucho del problema.



Ejercicio 2:

Aplicar data augmentation al conjunto de datos de perros y gatos para el problema de clasificación

People with no idea about AI saying it will take over the world:

My Neural Network:



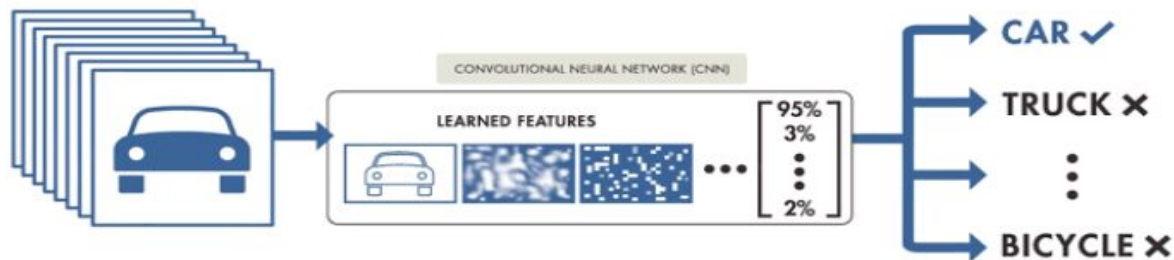
Transfer learning



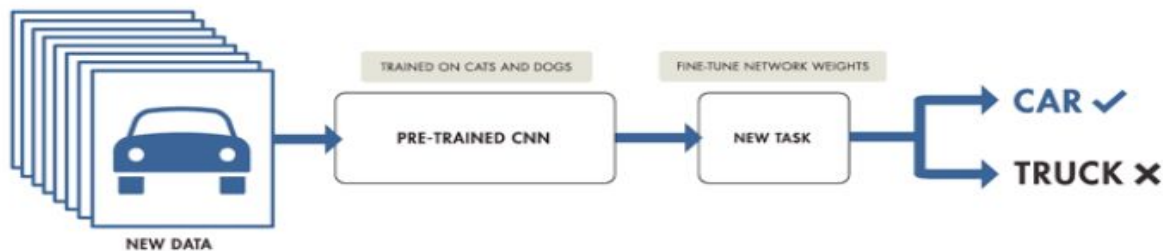
Transfer Learning es una técnica que consiste en utilizar un modelo pre-entrenado para resolver otra tarea relacionada.

Almacena el conocimiento adquirido al resolver un problema particular y utiliza el mismo conocimiento para resolver otro problema diferente pero que estén relacionados. Esto mejora la eficiencia al reutilizar la información recopilada de la tarea previamente aprendida.

TRAINING FROM SCRATCH




TRANSFER LEARNING



Ventajas:

- Se requieren menos datos para poder entrenar a nuestros modelos.
- Menor tiempo para entrenar a los modelos.

Si el conjunto de datos original con el que se entrenó la red neuronal pre-entrenada es suficientemente grande y diverso, entonces las características aprendidas permiten al modelo pre-entrenado actuar como un modelo genérico de extracción de características.



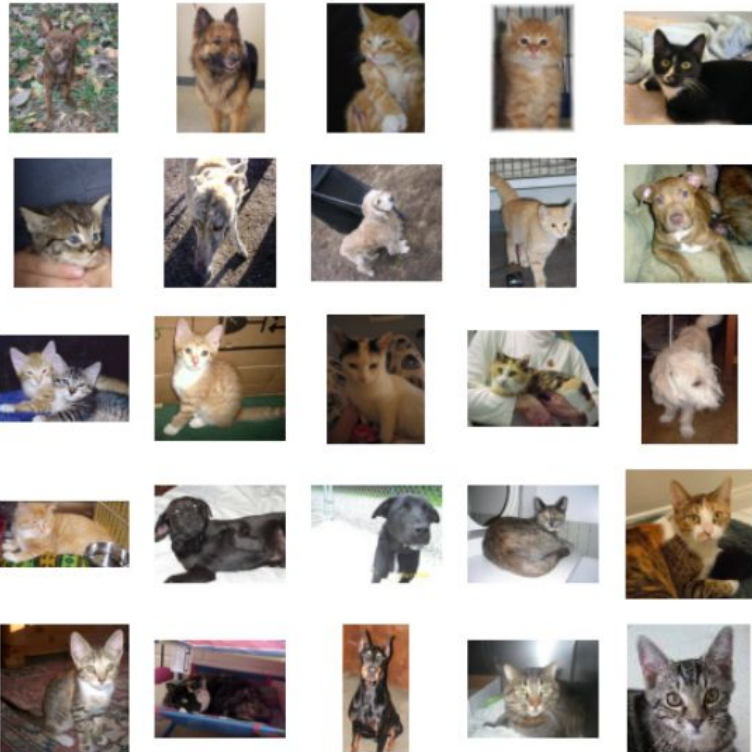
Específicamente, en el caso de la visión por computadora, muchos modelos previamente entrenados están disponibles públicamente para su descarga y se pueden usar para crear potentes modelos de visión con muy pocos datos.

Algunas redes que se utilizan como modelos pre entrenados en problemas de visión por computadora:

- VGG-16
- VGG-19
- Inception V3
- Xception
- ResNet-50



En IMAGENET las imágenes se categorizan en 22.000 clases distintas, sumando un total de aproximadamente 14 millones de imágenes al día de hoy.



Ejercicio 3:

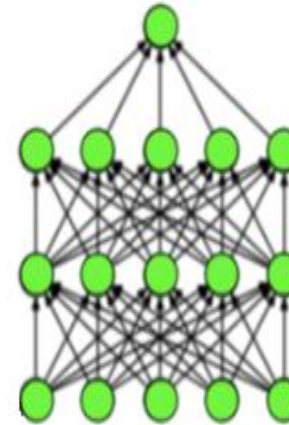
Hacer Transfer Learning al conjunto de datos de perros y gatos para el problema de clasificación

Herramientas útiles

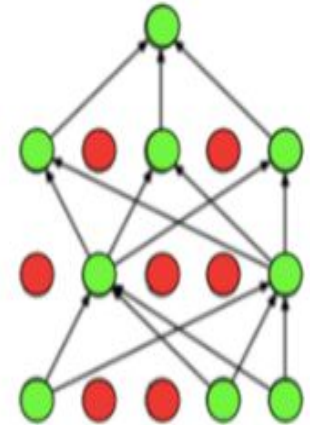
Dropout

Dropout es una técnica de regularización para modelos de redes neuronales

Dropout es un método que desactiva un número de neuronas de una red neuronal de forma aleatoria. En cada iteración de la red neuronal dropout desactiva diferentes neuronas, las neuronas desactivadas no se toman en cuenta para la propagación forward ni para el backpropagation lo que obliga a las neuronas cercanas a no depender tanto de las neuronas desactivadas.



**Red Neuronal
estándar**



**Después de
aplicar Dropout**



Batch Normalization

Batch normalization es una técnica general que se puede utilizar para normalizar las entradas de una capa.

Esto tiene el efecto de estabilizar el proceso de aprendizaje y reducir drásticamente la cantidad de épocas de entrenamiento necesarias para entrenar redes profundas.

Escala la salida de la capa, específicamente estandariza las activaciones de cada variable de entrada la estandarización se refiere a reescalar los datos para que tengan una media de cero y una desviación estándar de 1.

- Utilizarse antes de la función de activación ReLU.
- La estabilidad del entrenamiento aportada por batch normalization puede hacer que el entrenamiento de redes profundas sea menos sensible a la inicialización de pesos.
- Batch normalization ofrece cierto efecto de regularización, lo que reduce el error de generalización por eso no se recomienda mucho el uso de Batch Normalization con Dropout.