Convolutional Neural Networks

Prof. Cristian López Del Alamo

Junio de 2022

UNSA
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

Índice de Contenidos

Motivación y Contexto

Aprendizaje de Máquina

Representación de imagenes
Procesamiento de imagenes
Filtros de imagenes

Espacios Vectoriales
Tensores
Integral de Convolución
imagenes como tensores
Convolucionando...
¿imagenes?

¡Finalmente! CNN

CNN Completa
Funciónes de Activación
Extrayendo Características: Flattening
Fully connected layer - Capa Totalmente Conectada

Forward Propagation
Backpropagation
Transferencia de Aprendizaje

Motivación y Contexto

La Inteligencia Artifical ha ido presenciando un crecimiento monumental en la capacidad de conectar las capacidades humanas con las capacidades computacionales. Es así que investigadores y entusiastas trabajan numerosos aspectos del campo para lograr cosas asombrosas. Una de las muchas áreas de dicho dominio es la Visión Computacional.

Algo de Historia

Inicialmente, los fundamentos de las CNNs se basaron en el Neocognitron, introducido por Kunihiko Fukushima en 1980. Este modelo fue más tarde mejorado por Yann LeCun en 1983, al introducir un método de aprendizaje basado en backpropagation para poder entrenar el sistema correctamente. En el año 2012, fueron refinadas por Dan Ciresan y otros, y fueron implementadas oficialmente para correr en una unidad de procesamiento gráfico (GPU) consiguiendo así resultados impresionantes.

Definiciones

"El aprendizaje de máquina es el campo de estudio que le da a la computadora la capacidad de aprender sin ser programada explícitamente."

"Se dice que un programa aprende de la experiencia E con referencia a alguna clase de tarea T y con la medida de rendimiento P, si su rendimiento en la tarea T, medido por P, mejora con la experiencia E."

Datos

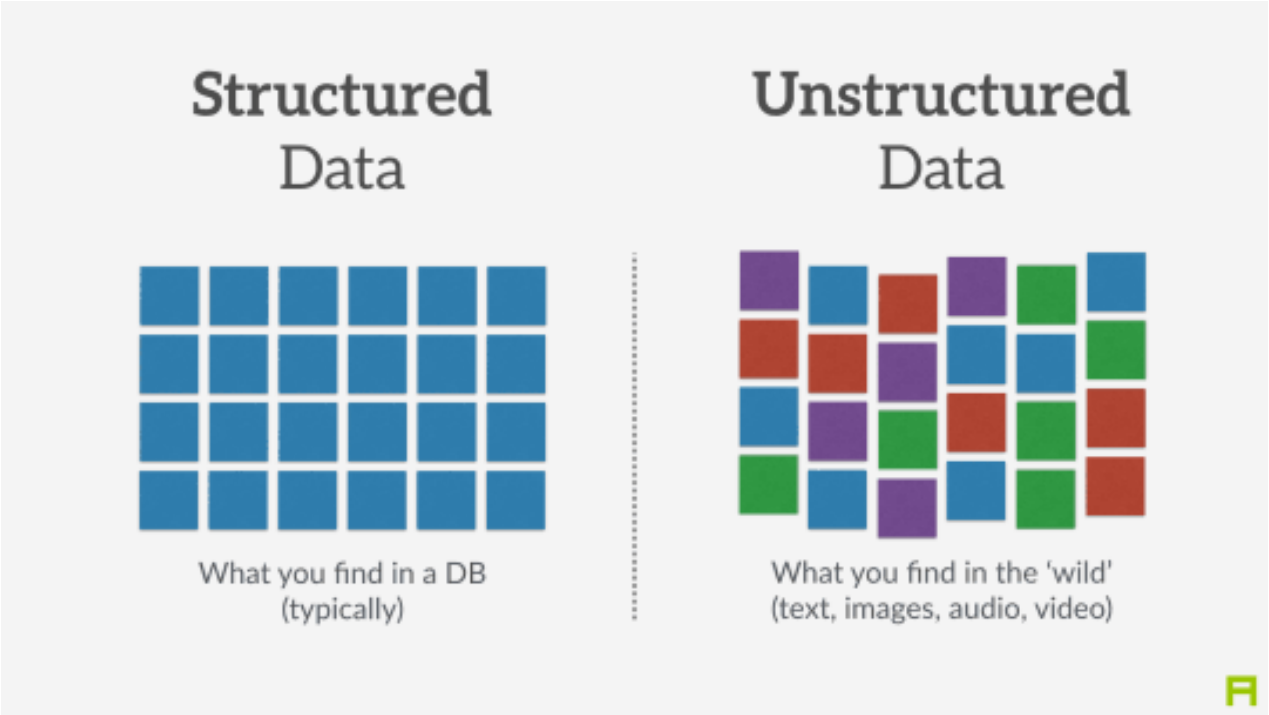


Figure: Las modelos trabajan con datos estructurados y no estructurados

Big Data Volume = Cálculos rápidos

Para la clasificación de imagenes, una regla general es 1000 imagenes por clase, donde este número puede disminuir significativamente si se usan modelos previamente entrenados.

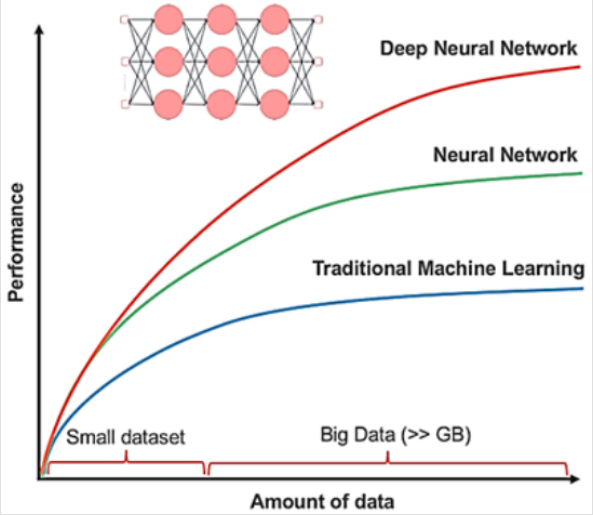


Figure: Cálculos en el Deep Learning

Aprendizaje Supervisado

[HTML]FFCB2F **Entrada: X**	**Salida: X**	**Tipo de NN**
->Precios de una Casa ->Información de Usuarios	->Precio (Regresión) ->Click del Usuario (0/1)	NN Standard
->imagen	->Clase (1, 2, 3, ..., 1000) ->Características	NN Convolucionales (CNN)
->Audio ->Texto	->Transcripción del Audio ->Traducción en otro lenguaje	NN Recurrente (RNN)

Redes Neuronales Artificiales (ANN)

Una neurona puede verse como una unidad computacional que recibe una cantidad de entradas a través de sus "cables" de entrada y realiza algunos cálculos, para después enviar salidas a través de su axón a otros nodos u otras neuronas en el cerebro. En general, una red neuronal artificial es un grupo interconectado de nodos similar a la vasta red de neuronas en un cerebro biológico.

Redes Neuronales Artificiales (ANN)

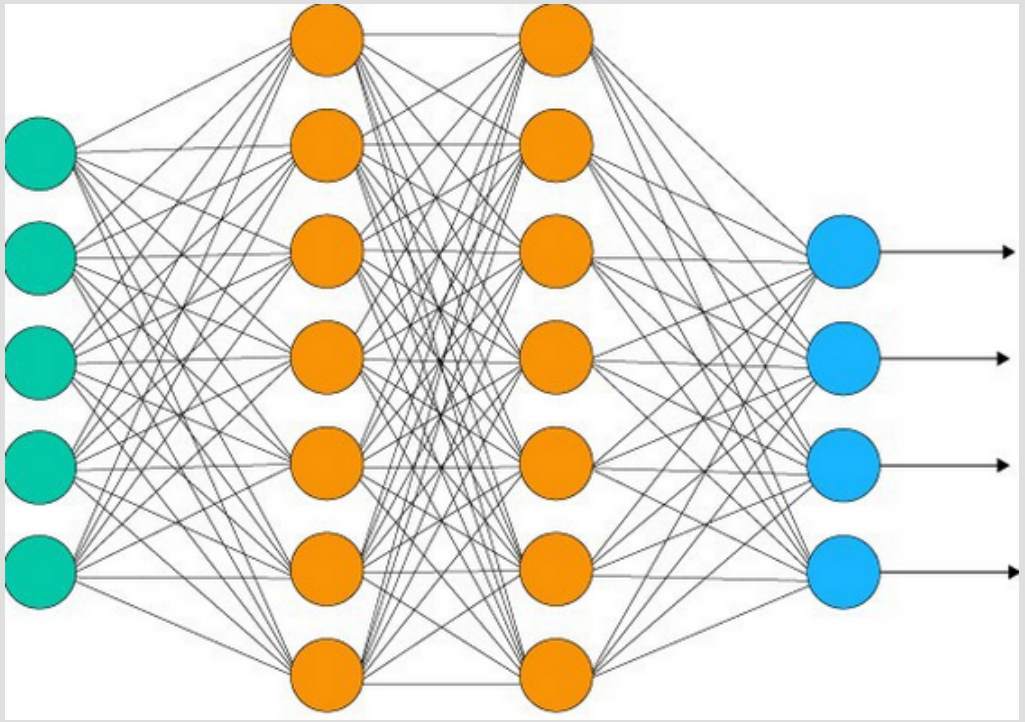


Figure: Red neuronal Artificial.

imagenes por computador

Una imagen digital es una representación bidimensional de una imagen a partir de una matriz numérica.

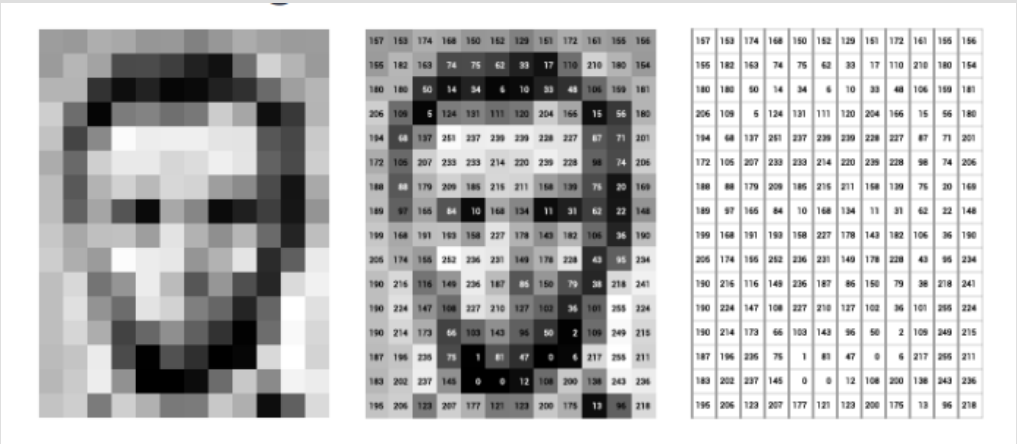


Figure: Imagen con 256 niveles de intensidad representada mediante una matriz.

imagenes como Funciónes

Una imagen se define como una función de dos dimensiones $f(x, y)$ donde x e y son las coordenadas de un plano que contiene todos los puntos de la misma, en donde al evaluar f en la posición (x,y) obtenemos un nivel de intensidad de brillo.

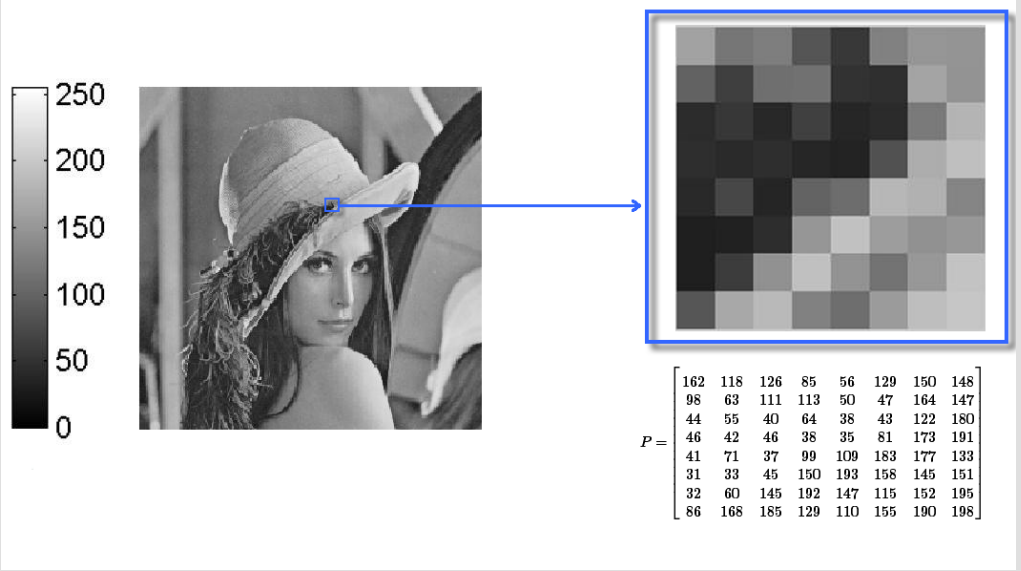


Figure: Imagen con 256 niveles de intensidad y representación numérica de un fragmento 8x8.

Procesamiento de imagenes

Es todo un conjunto de técnicas que tiene como objetivos, obtener una representación más adecuada de una imagen, o bien, es también utilizado en la extracción de características relevantes de esta, las cuales muchas veces no son perceptibles a primera vista.

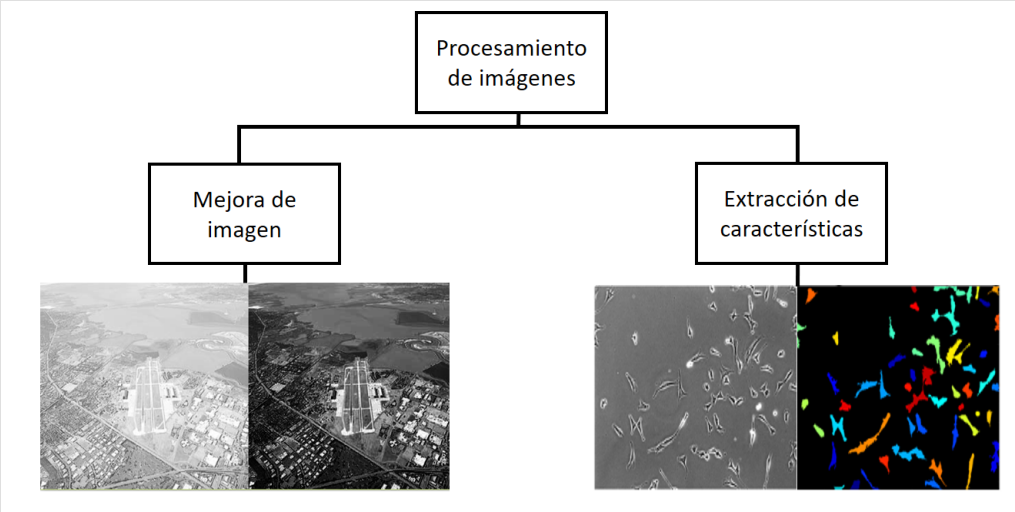


Figure: La mejora de la imagen (image enhancement) y la extracción de características (feature extraction) son los dos principales objetivos que plantea el procesamiento digital de imagenes.

Filtros de imagenes

Los filtros digitales constituyen uno de los principales modos de operar en el procesamiento de imagenes digitales. Pueden usarse para distintos fines, pero en todos los casos, el resultado sobre cada píxel depende de los píxeles de su entorno.

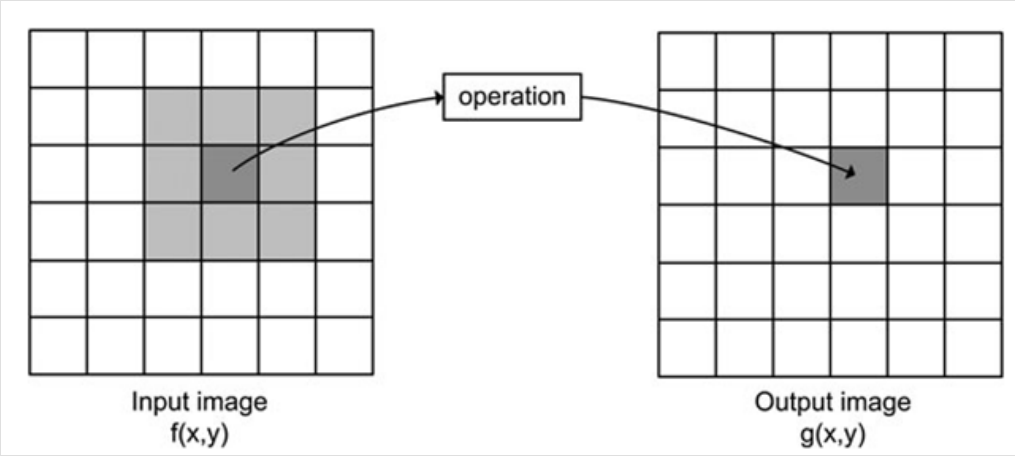


Figure: Filtro.

Filtro de la Media

El filtro medio es un filtro espacial de ventana deslizante simple que reemplaza el valor central en la ventana con el promedio (media) de todos los valores de píxeles en la ventana.

Figure: Filtro de la media.

Filtro de la Mediana

El filtro de la mediana reemplaza el valor central en la ventana con la mediana de todos los valores de píxeles en la ventana.

Figure: Filtro de la Mediana.

Filtro de Sobel

El operador Sobel calcula el gradiente de la intensidad de una imagen en cada punto.

Figure: Detección de bordes con sobel.

Espacios Vectoriales

En general, un **espacio vectorial** es una **estructura algebráica** creada a partir de un conjunto no vacío, una **operación interna** (suma) y una **operación externa** (multiplicación) con otro conjunto (con estructura de cuerpo) que satisface 8 propiedades fundamentales; a los elementos de un espacio vectorial se les llama **vectores** y a los elementos del cuerpo se les conoce como **escalares**.

Es una n-tupla $(a_1, a_2, ..., a_n)$ donde a_1 es un conjunto dado no vacío y $\{a_2, ..., a_n\}$ un conjunto de operaciones aplicables a los elementos de dicho conjunto. En general, una **estructura algebráica** es la abstracción de un conjunto no vacío y sus operaciones.

Entendiendo Espacios Vectoriales (Ejemplo 1)

Figure: Conjunto no vacío: Acuarela.

Entendiendo Espacios Vectoriales (Ejemplo 1)

Figure: Conjunto no vacío: Agua.

Entendiendo Espacios Vectoriales (Ejemplo 1)

Figure: Operación Interna en las acuarelas

Entendiendo Espacios Vectoriales (Ejemplo 1)

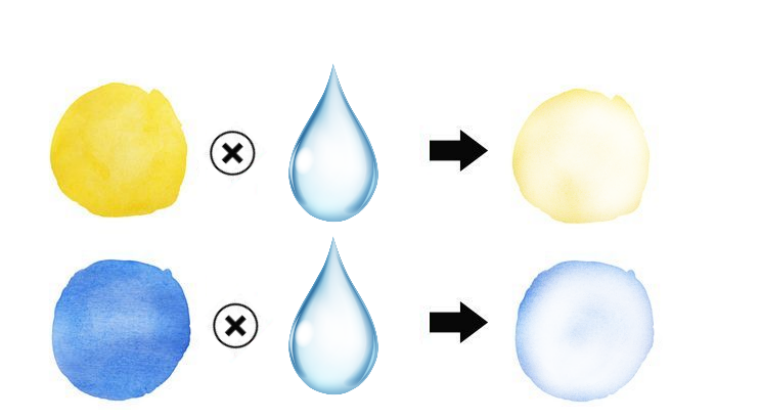


Figure: Operación Externa de las acuarelas con el agua

Entendiendo Espacios Vectoriales (Ejemplo 2)

Figure: Conjunto no vacío: imagenes.

Entendiendo Espacios Vectoriales (Ejemplo 2)

Figure: Conjunto no vacío (cuerpo): Números Reales.

Entendiendo Espacios Vectoriales (Ejemplo 2)

Figure: Operación Interna en imagenes (blending).

Entendiendo Espacios Vectoriales (Ejemplo 2)

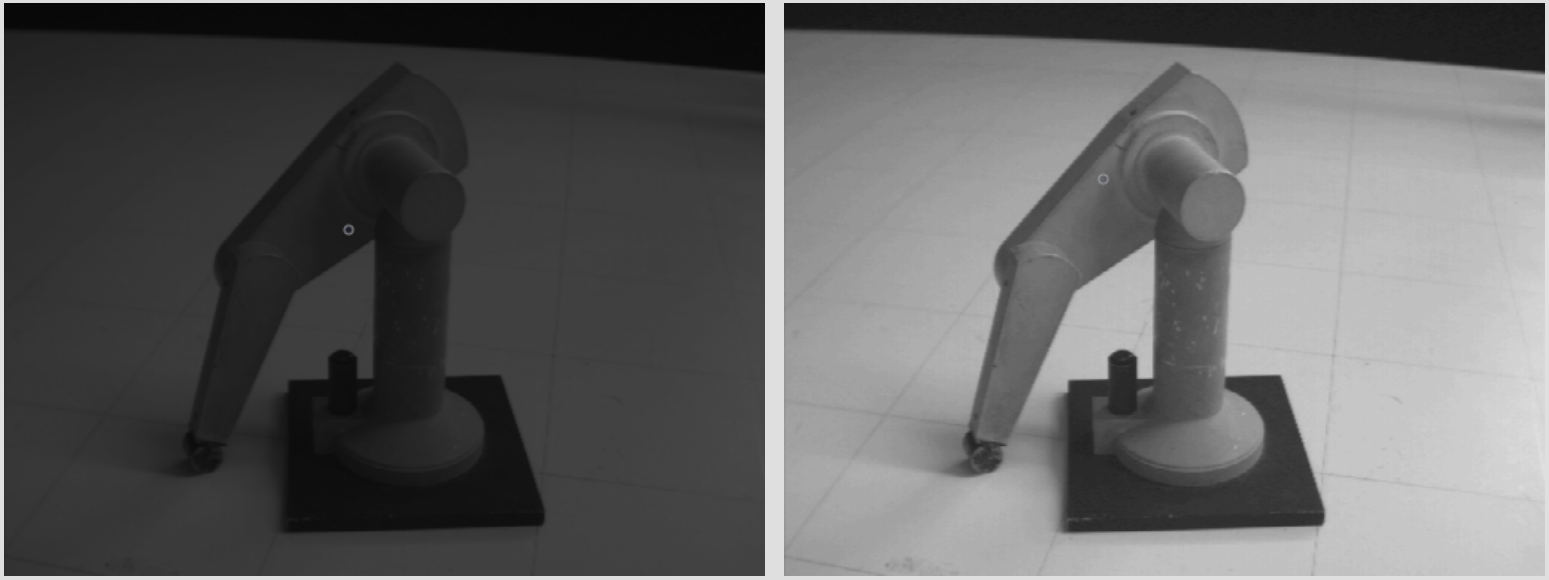


Figure: Operación Externa de imagenes con reales (multiplicación).

Espacios Vectoriales

Un espacio vectorial es un conjunto no vacío V cuyos objetos se denominan vectores, en el que hay definidas dos operaciones: suma y multiplicación por un escalar (números reales o complejos) que satisfacen los axiomas que serán determinados a continuación. Los axiomas deben ser válidos para todos los vectores u, v y w en V y todos los escalares α y β reales. Llamamos $u + v$ a la suma de vectores en V, y αv al producto de un número real α por un vector $v \in V$. Los axiomas 1 y 6 son redundantes a la definición de operación interna y externa.

Espacios Vectoriales

1. *La suma es una operación interna:* $u + v \in V$.
2. *La suma es conmutativa:* $u + v = v + u$.
3. *La suma es asociativa:* $(u + v) + w = u + (v + w) = u + v + w$.
4. *Elemento neutro de la suma:* $\exists 0 \in V \mid v + 0 = v, \ \forall v \in V$.
5. *Elemento inverso en la suma:* $\forall v \in V, \ \exists v' \in V \mid v + v' = 0$, se escribe $v' = (-v)$.
6. *La multiplicación por un escalar produce un vector:* $cv \in V$.
7. *Distributividad I:* $\alpha(u + v) = \alpha u + \alpha v$
8. *Distributividad II:* $(\alpha + \beta)v = \alpha u + \beta v$
9. *Asociatividad:* $\alpha(\beta v) = (\alpha\beta)v = \alpha\beta v$
10. $1 \cdot v = v$

El vector 0 es único y, dado v, también lo es $-v$. Además $0v = 0$, $\alpha 0 = 0$, $-v = (-1)v$.

Espacios Vectoriales

En matemáticas, una aplicación lineal es una aplicación entre dos espacios vectoriales que preserva las operaciones de adición de vectores y multiplicación por un escalar.

Sean VyW espacios vectoriales sobre el mismo cuerpo $\mathbb{K}$. Una aplicación T de V en W, es decir, $T: \ V \to W$, es una aplicación lineal si para todo par de vectores $u, v \in V$ y para todo escalar $k \in \mathbb{K}$, se satisface que:

1. $T(u+v) = T(u) + T(v)$
2. $T(ku) = kT(u)$

Tensores

Un tensor es un objeto matemático que pertenece a un espacio vectorial (tensorial) y que es invariante ante un cambio de base. Cuando cambia la base, sus componentes se pueden recalcular a través de una fórmula conocida.

Es de capital importancia asumir que un tensor es invariante ante un cambio de base, pero sus componentes NO. Precisamente su potencia radica en que sirve para representar magnitudes invariantes ante rotaciones (imagenes), cambios de escala (unidades de medida), cambios de base (inclinación de ejes), etc. Si rotamos el espacio las componentes del tensor podrán ser diferentes, pero el tensor en sí será el mismo.

Aclaración

Una matriz es un concepto restringido de tensor. Es un arreglo rectangular de mn números que pertenecen a un cuerpo (generalmente números reales o complejos). Un tensor es un arreglo de $mn...p$. Así, un escalar es un tensor de orden 0, un vector es un tensor de orden 1, una matriz es un tensor de orden 2, y así hasta el infinito. Los tensores surgen naturalmente en matemáticas debido a las Aplicaciónes lineales entre diversos tipos de espacios vectoriales. Es por eso que lo que tenemos es una aplicación multilineal del tipo:

$$f : V_1 V_2 V_3 ... V_{n-1} \rightarrow W$$

Grado de un Tensor

1. **Escalar** adimensional o Tensor de grado 0: $\begin{bmatrix} A \end{bmatrix}$
2. **Vector** unidimensional o Tensor de grado 1: $\begin{bmatrix} A_1 & A_2 & A_3 \end{bmatrix}$
3. **Matriz** bidimensional, o Tensor de grado 2:
$$\begin{bmatrix} A_{11} & A_12 & A_13 \\ A_{21} & A_22 & A_23 \\ A_{31} & A_32 & A_33 \end{bmatrix}$$

Grado de un Tensor

1. **Tensor de grado 3**. Cada entrada es un vector, o sea se podría imaginar como un "cubo de números":

$$\begin{bmatrix} \begin{bmatrix} A_{111} & A_{112} & A_{113} \\ A_{211} & A_{212} & A_{213} \\ A_{311} & A_{312} & A_{313} \end{bmatrix} & \begin{bmatrix} A_{121} & A_{122} & A_{123} \\ A_{221} & A_{222} & A_{223} \\ A_{321} & A_{322} & A_{323} \end{bmatrix} & \begin{bmatrix} A_{131} & A_{132} & A_{133} \\ A_{231} & A_{232} & A_{233} \\ A_{331} & A_{332} & A_{333} \end{bmatrix} \end{bmatrix}$$

2. **Tensor de grado 4**: $\begin{bmatrix} \begin{bmatrix} A_{1111} & A_1112 \\ A_{1121} & A_1122 \end{bmatrix} & \begin{bmatrix} A_{1211} & A_1212 \\ A_{1221} & A_1222 \end{bmatrix} \\ \begin{bmatrix} A_{2111} & A_2112 \\ A_{2121} & A_2122 \end{bmatrix} & \begin{bmatrix} A_{2211} & A_2212 \\ A_{2221} & A_2222 \end{bmatrix} \end{bmatrix}$

Frame Title

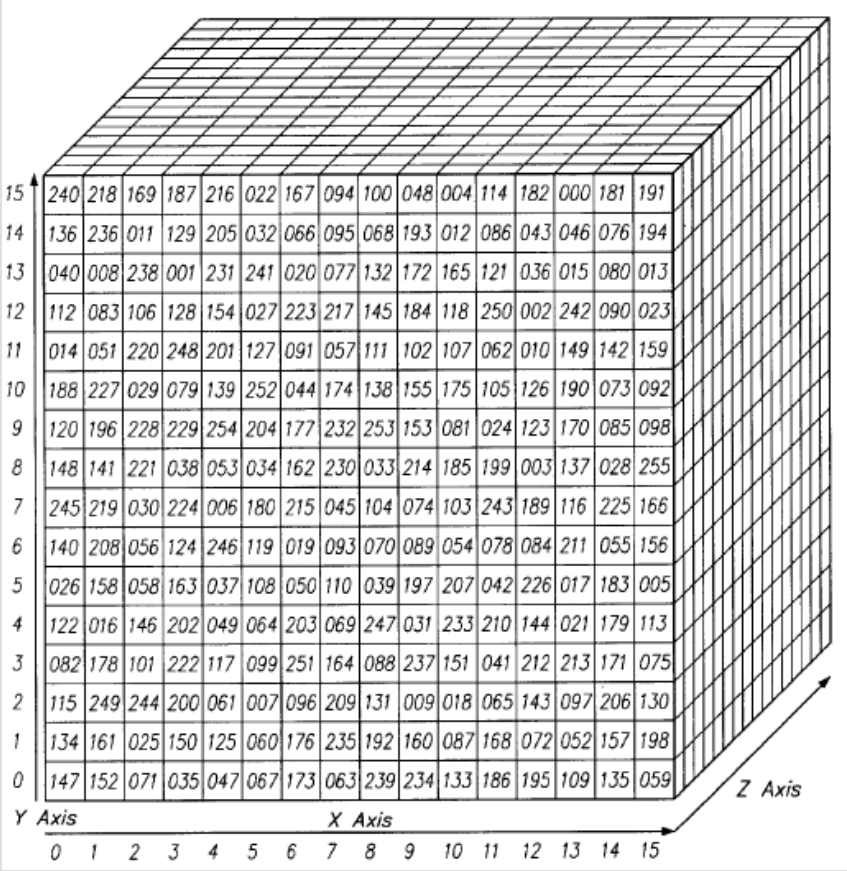


Figure: Abstracción de un Tensor.

Integral de Convolución

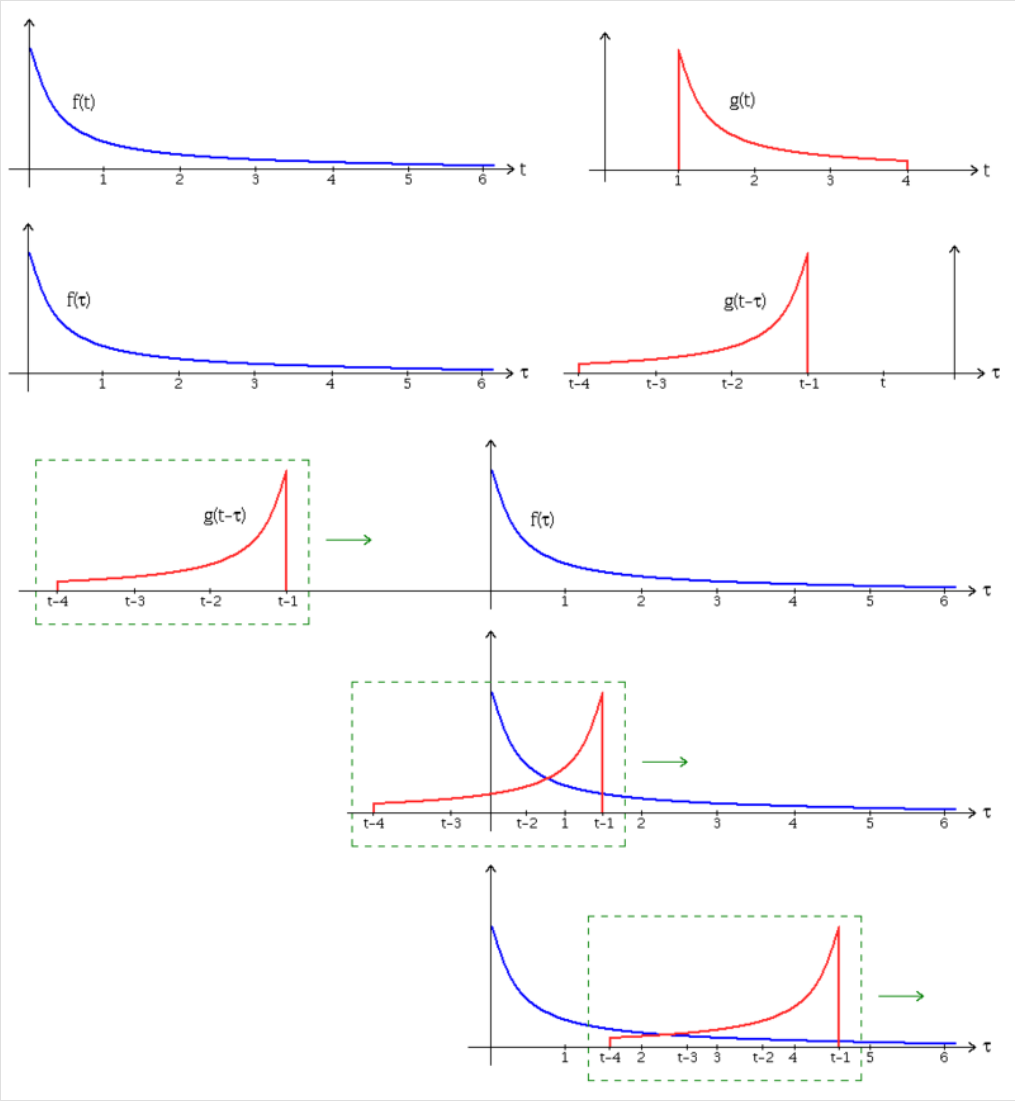
En matemáticas, y en particular análisis Funciónal, una convolución es un operador matemático que transforma dos Funciónes f y g en una tercera función que en cierto sentido representa la magnitud en la que se superponen f y una versión trasladada e invertida de g. Una convolución es un tipo muy general de media móvil, como se puede observar si una de las Funciónes se toma como la función característica de un intervalo. Está definida de la forma:

$$f(t) * g(t) = (f * g)(t) = \int_{-\infty}^{+\infty} f(\tau) \cdot g(-\tau + t) d\tau$$

Integral de Convolución

Podemos interpretar la convolución imaginando un desplazamiento de la función "kernel" sobre una fija y calcular progresivamente el área bajo la curva de la función producto de ambas. Esto será explicado mejor en la sesión interactiva en Geogebra.

Integral de Convolución



¡Aquí las imagenes son tensores!

Dada la naturaleza de los tensores, es que se empezó a representar las imagenes en forma de tensores, debido a las propiedades de los mismos e incluso la posibilidad de unificar las imagenes con múltiples canales.

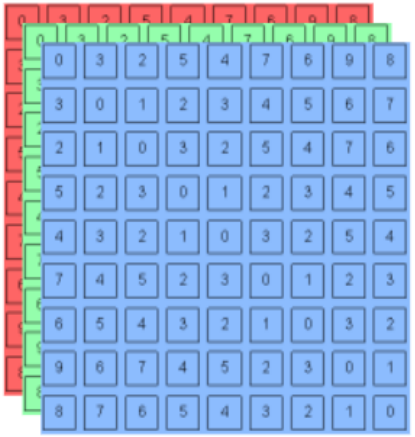


Figure: Imagen RGB como tensor.

Formulación discreta

Como ya vimos anteriormente, la integral de convolución se define como:

$$f(t) * g(t) = (f * g)(t) = \int_{-\infty}^{+\infty} f(\tau) \cdot g(-\tau + t) d\tau$$

Pero podemos partir de ese planteamiento para convolucionar... ¿imagenes?. ¡Sí! Y esto se debe a la maravillosa idea de aplicar filtros en las imagenes, ya que de alguna forma los mismos son Funciónes que deseamos "superponer" encima de una imagen para obtener su "reacción" a la misma, ¡y de eso trata convolucionar!

Formulación discreta

Es por ello que, reformulando discretamente, podríamos decir que para convolucionar un conjunto de puntos debemos calcular la sumatoria en vez de la integral, pues discretamente es la misma idea, teniendo así:

$$(f * g)(t) = \sum_{i=-\infty}^{+\infty} f(i) \cdot g(-i + t)$$

Ahora, considerando que el dominio de una imagen es finito, podemos limitarlo de 0 a N de donde $N = m \cdot n$ siendo m la altura y n el ancho, teniendo ahora:

$$(f * g)(t) = \sum_{i=0}^{N} f(i) \cdot g(-i + t)$$

Formulación discreta

Ahora, f representa la función que nos retorna los píxeles en cierta posición y g representa el kernel: lo renombraremos w. Luego, nos damos cuenta que nuestra imagen está siendo representada en una sola dimensión, lo que no nos ayuda mucho intuitivamente, por lo que convertiremos esto en una sumatoria doble para el ancho y el alto y ahora recibiremos 2 variables, x e y para posicionar el pixel:

$$(f * w)(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} f(i, j) \cdot w(-i + x, -j + y)$$

Formulación discreta

Sin embargo, nos damos cuenta que al ser discreto también el kernel, este mismo podrá tener un tamaño finito, lo que limita nuestra sumatoria a las dimensiones del kernel $w_m w_n$, dejándonos con:

$$(f * w)(x, y) = \sum_{i=-\frac{w_m}{2}}^{\frac{w_m}{2}} \sum_{j=-\frac{w_n}{2}}^{\frac{w_n}{2}} f(i, j) \cdot w(-i + x, -j + y)$$

Y si lo reordenamos:

$$(f * w)(x, y) = \sum_{i=-\frac{w_m}{2}}^{\frac{w_m}{2}} \sum_{j=-\frac{w_n}{2}}^{\frac{w_n}{2}} f(i, j) \cdot w(x - i, y - j)$$

Formulación discreta

Ahora, si consideramos que nuestro kernel de kernel $w_m w_n$ es un cuadrado, entonces $w_m = w_n$, y no solo eso, además podemos definir el radio cuadrangular como $r = \left\lfloor \frac{w_m}{2} \right\rfloor = \left\lfloor \frac{w_n}{2} \right\rfloor$, lo que nos permite tener:

$$(f * w)(x, y) = \sum_{i=-r}^{r} \sum_{j=-r}^{r} f(i, j) \cdot w(x - i, y - j)$$

Y finalmente, si nuestro kernel w es simétrico, entonces $w(x - i, y - j) = w(x + i, y + j)$ con lo que finalmente tendríamos:

$$(f * w)(x, y) = \sum_{i=-r}^{r} \sum_{j=-r}^{r} f(i, j) \cdot w(x + i, y + j)$$

¡Lo que nos deja con un planteamiento maravilloso! ya que si lo

Capa Convolucional

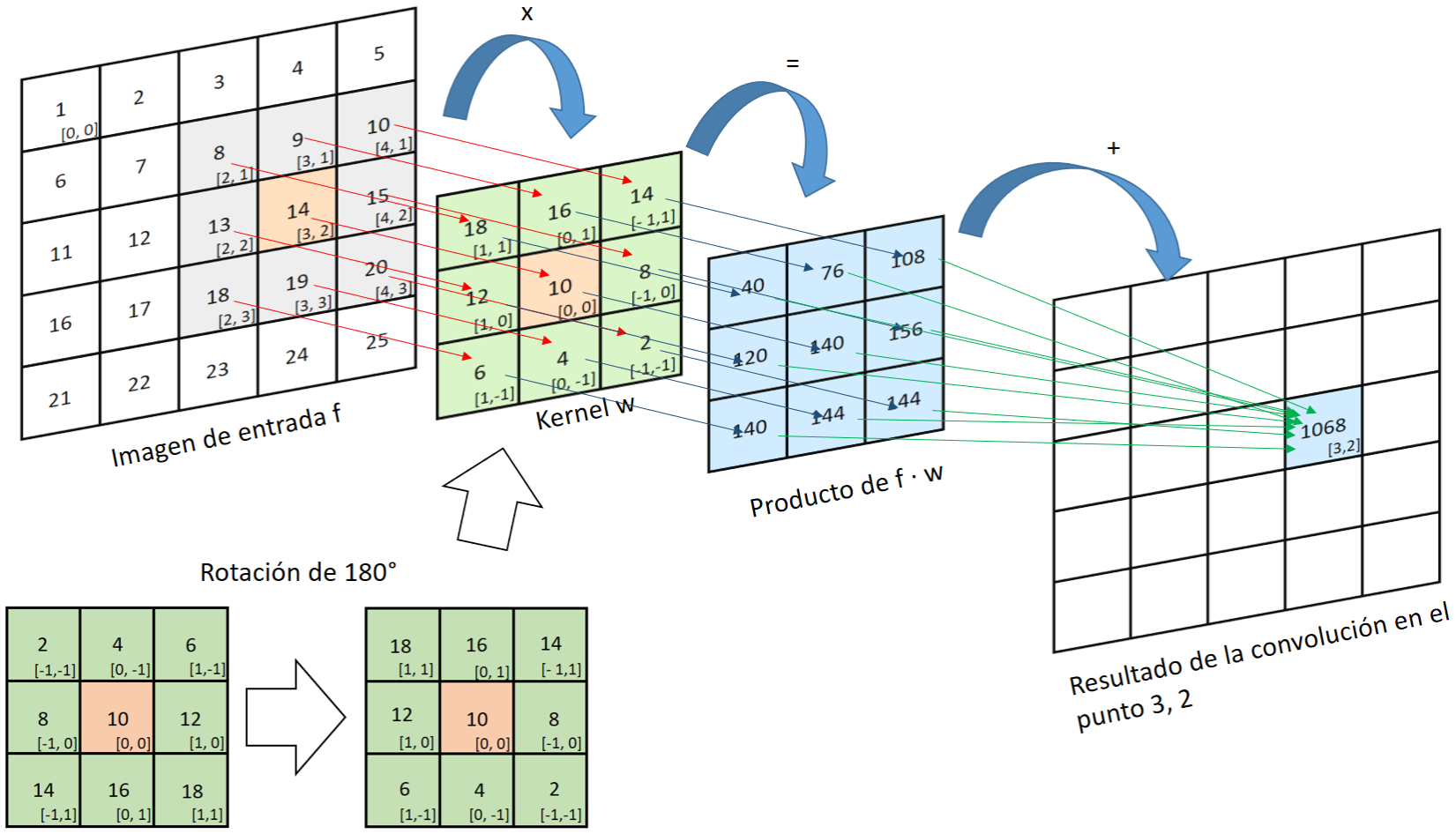


Figure: Representación visual del "deslizamiento"

Capa Convolucional

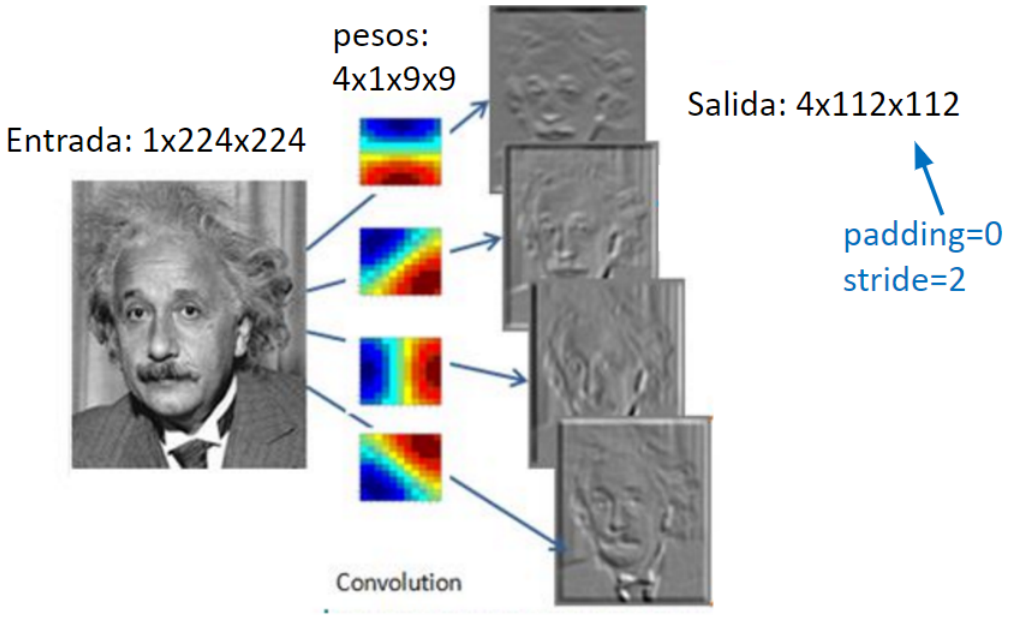


Figure: Redes Convolucionales.

Capa Convolucional

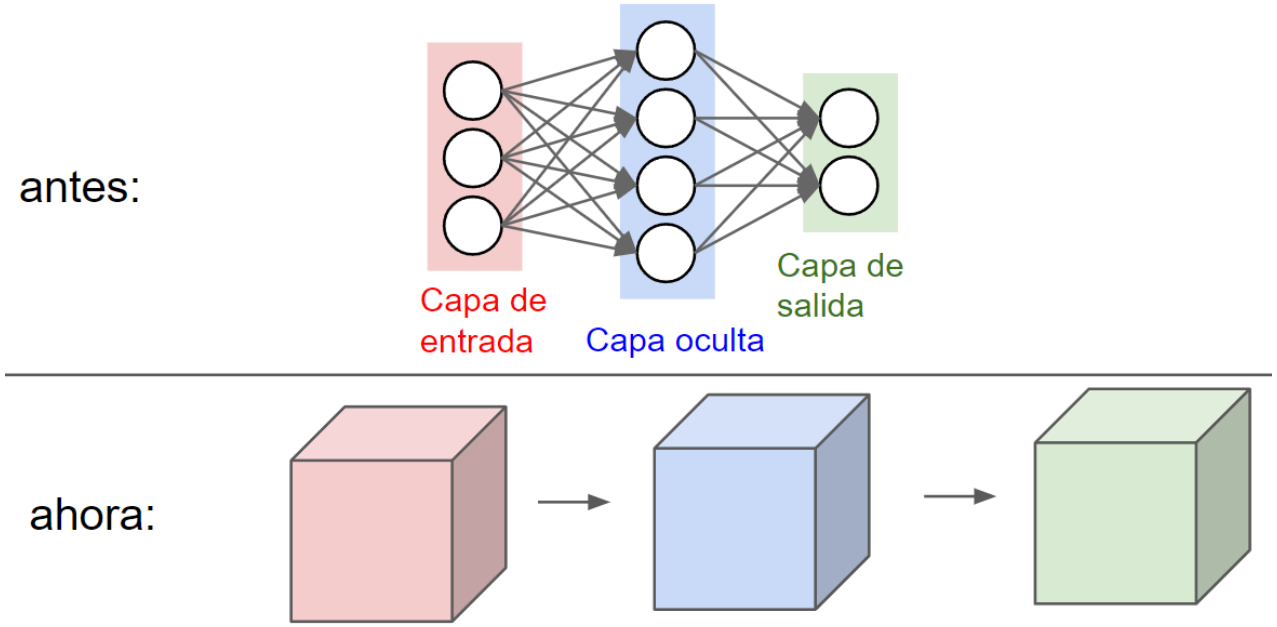


Figure: Redes Convolucionales.

Capa Convolucional

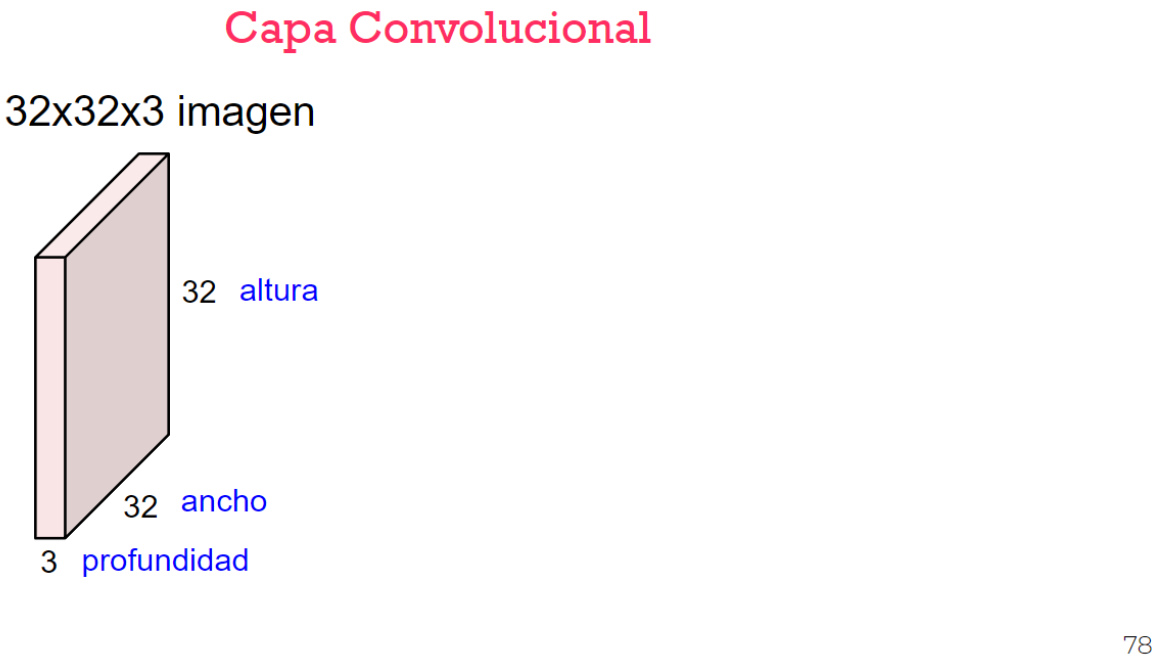


Figure: Redes Convolucionales.

Capa Convolucional

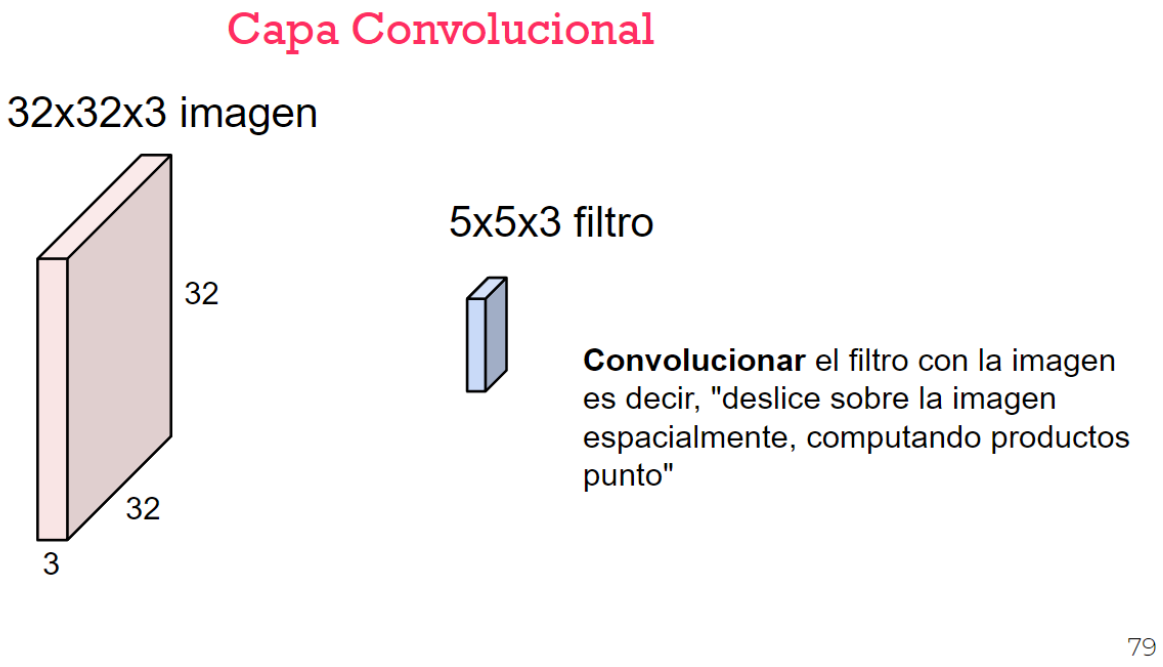


Figure: Redes Convolucionales.

Capa Convolucional

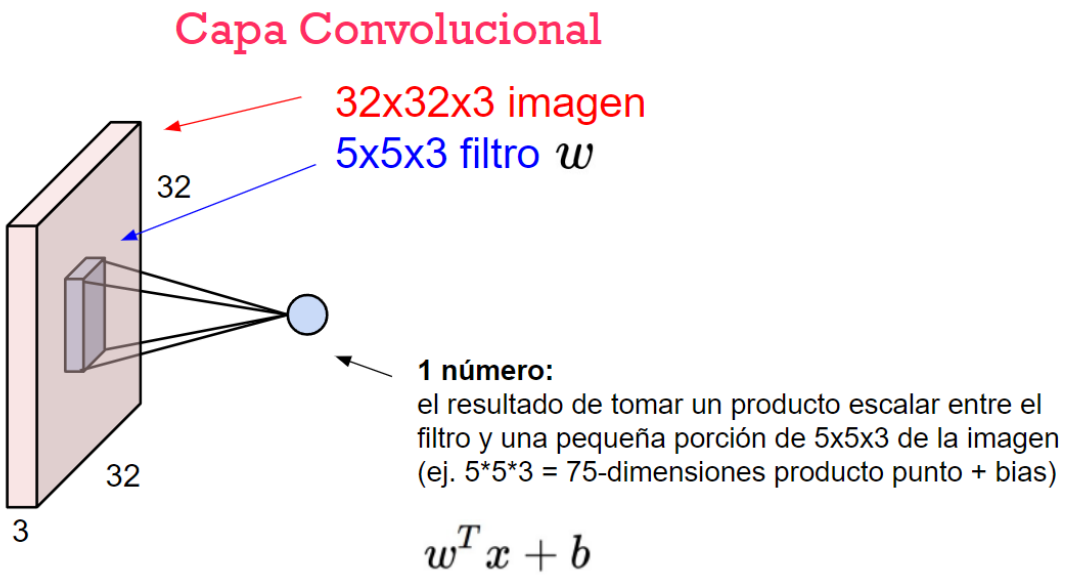


Figure: Redes Convolucionales.

Capa Convolucional

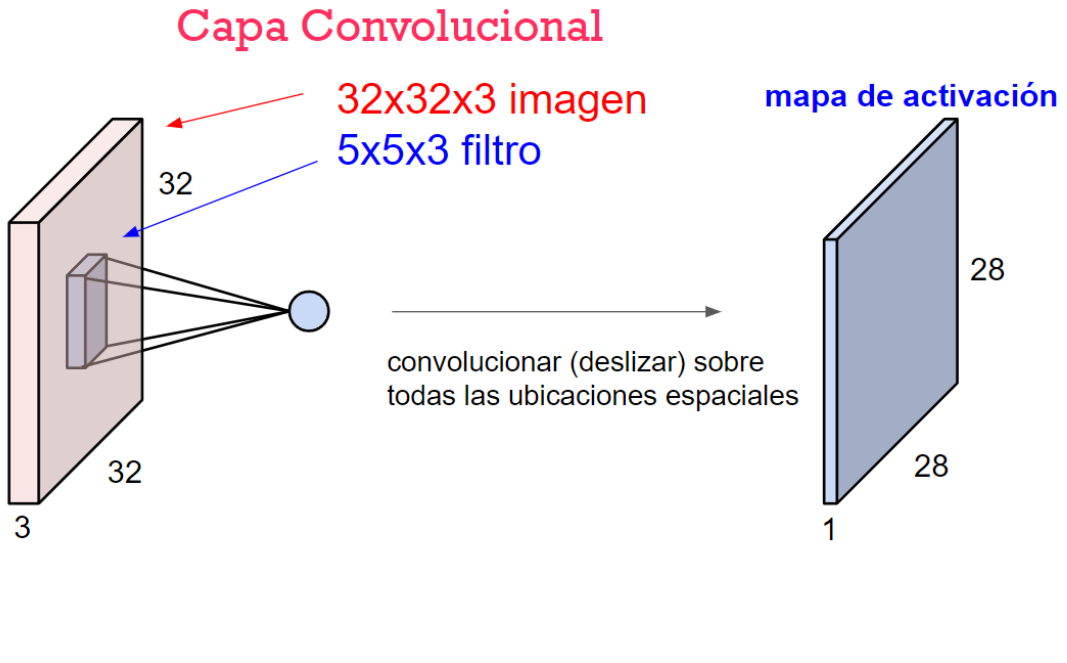


Figure: Redes Convolucionales.

Capa Convolucional

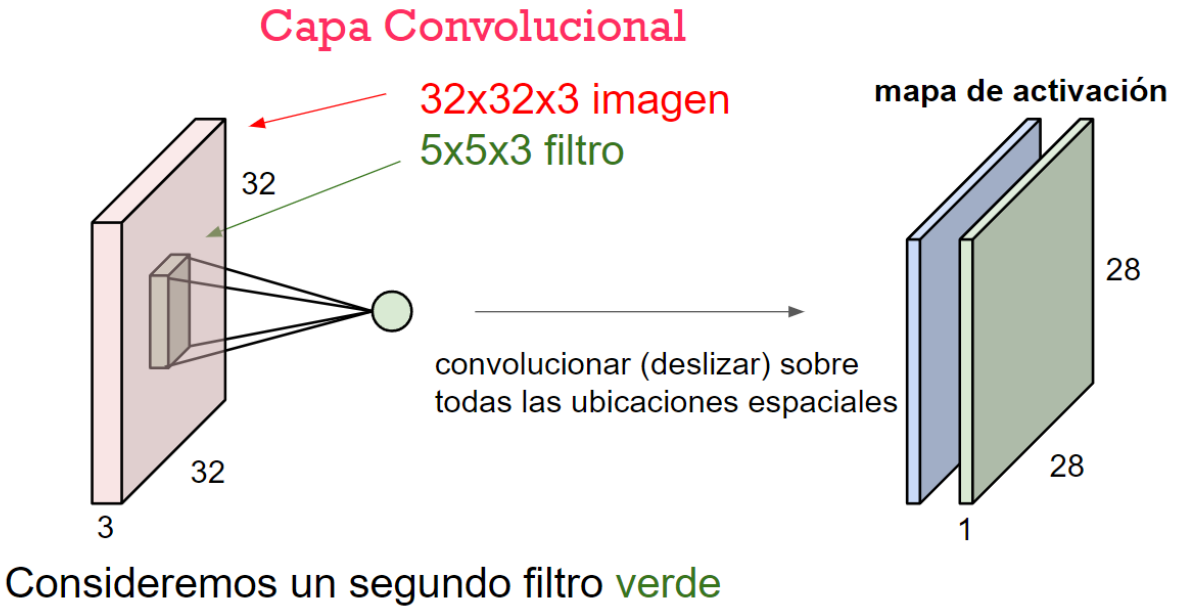


Figure: Redes Convolucionales.

Capa Convolucional

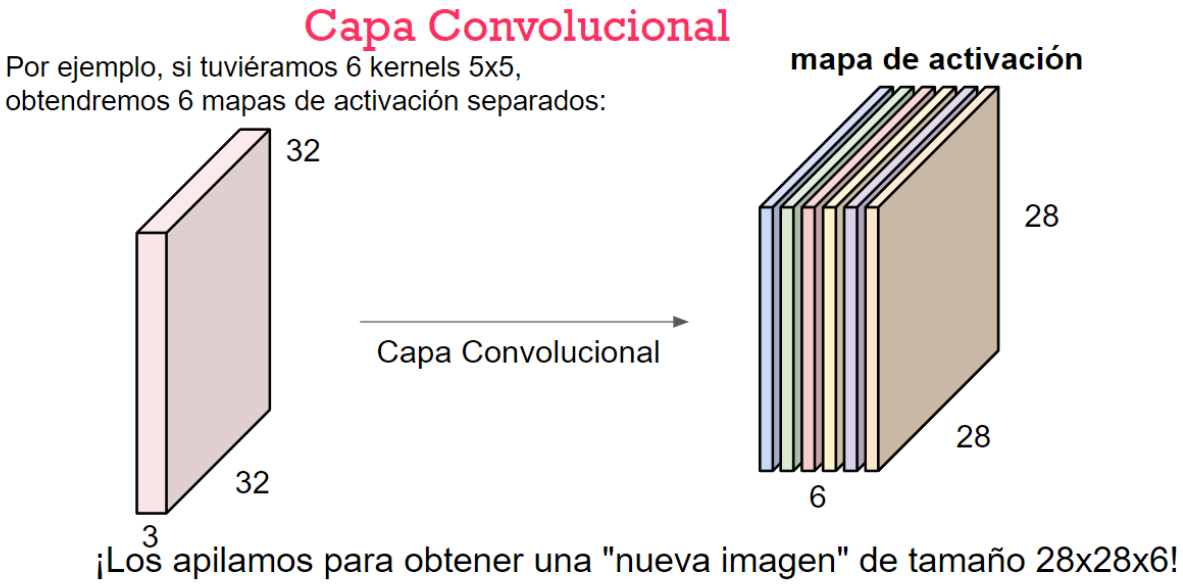
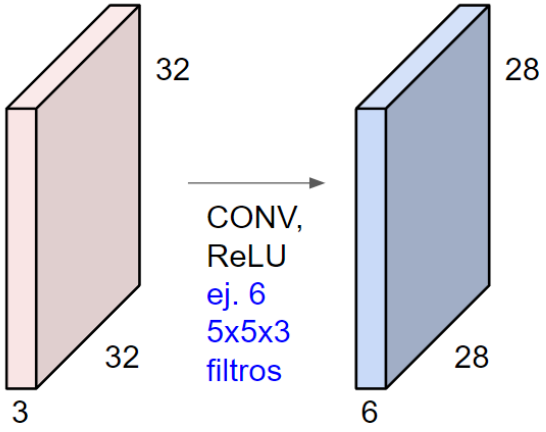


Figure: Redes Convolucionales.

CNN

CNN o ConvNet

CNN o ConvNet es una secuencia de capas de convolución, intercaladas con funciones de activación



85

Figure: Redes Convolucionales.

CNN

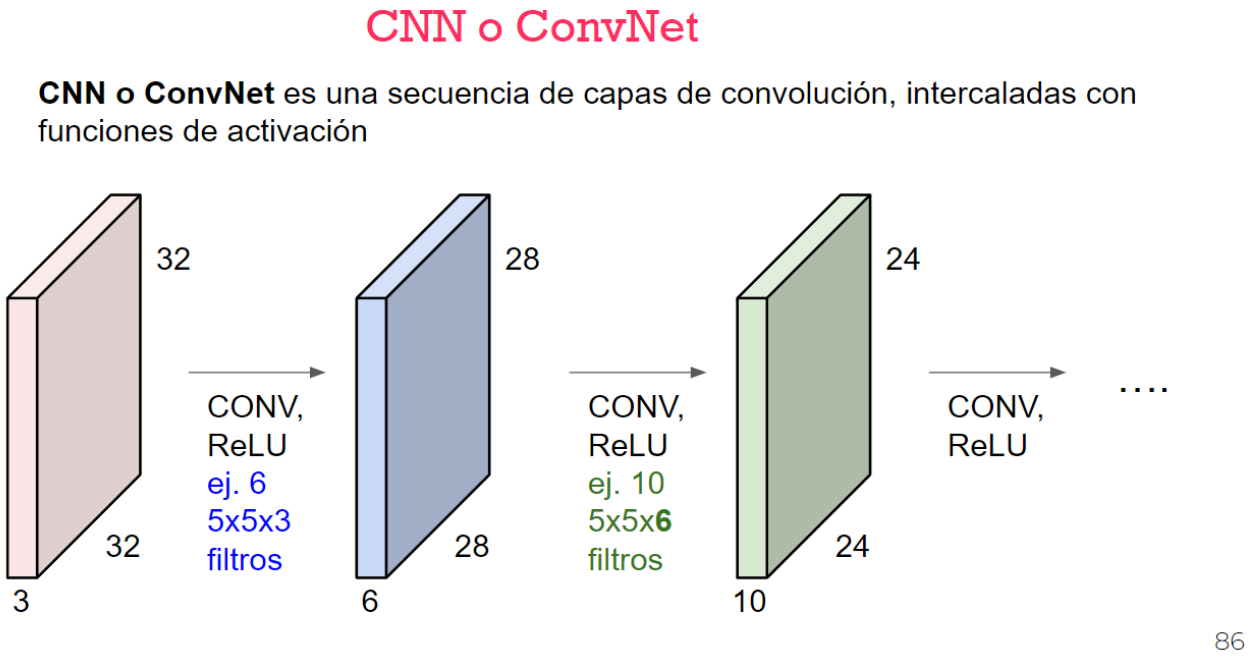


Figure: Redes Convolucionales.

CNN

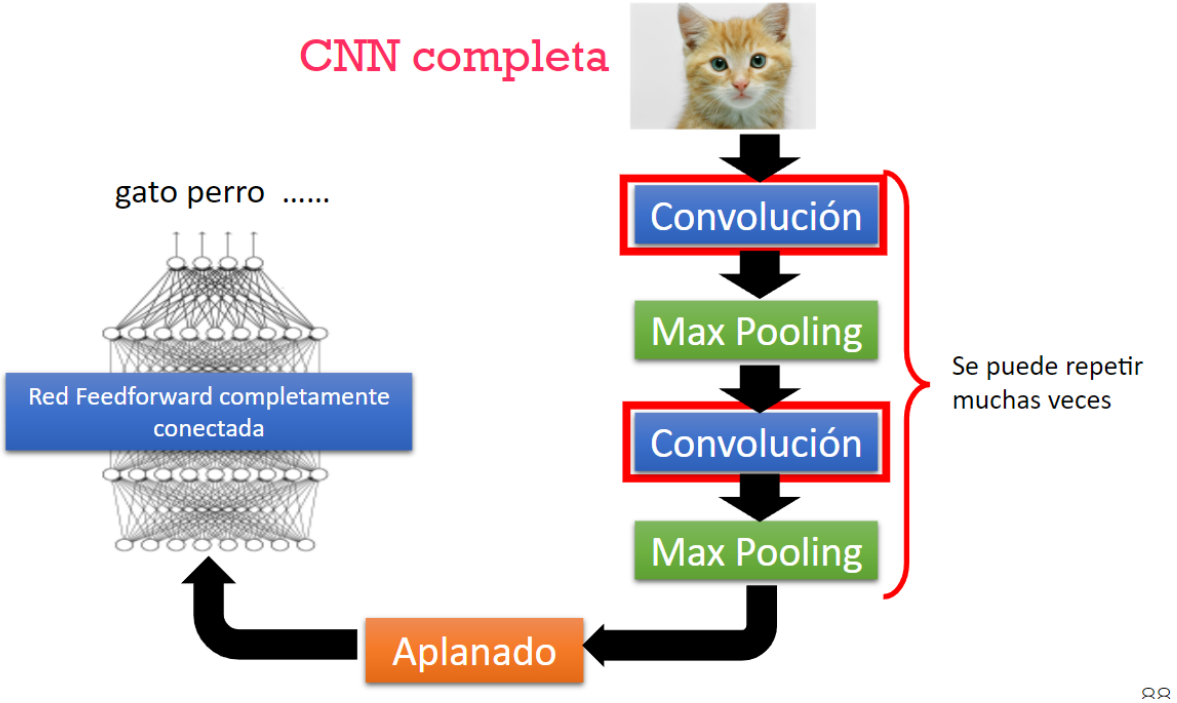
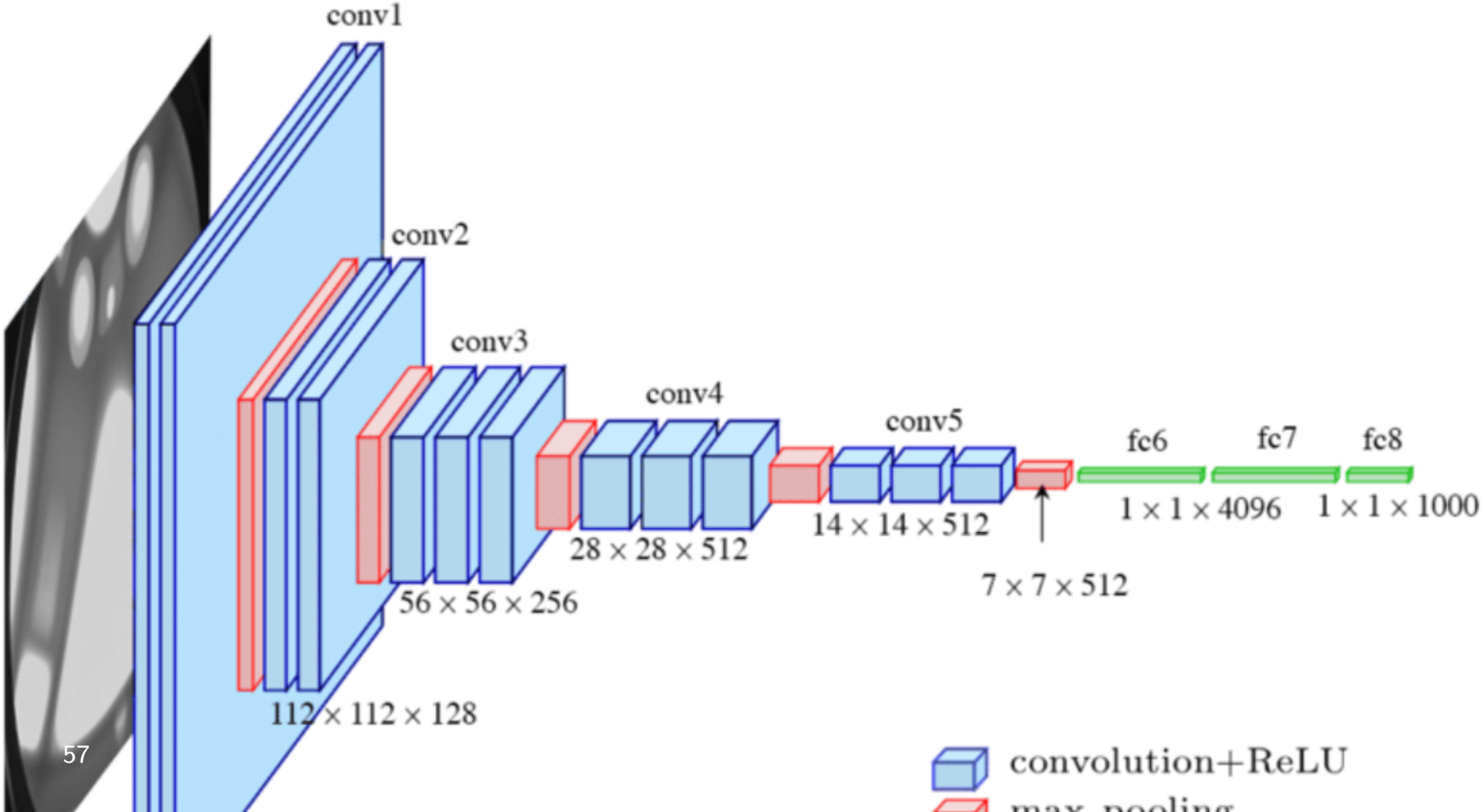


Figure: Redes Convolucionales.

Hiperparametros

- Kernel size: Tamaño del kernel
- Stride: Desplazamiento del kernel
- Dilation: Dilatación del kernel
- Number of filters: Cantidad de filtros
- Padding: Agregar 0's a la imagen para evitar que se reduzca
- Pooling type and size

Arquitectura de CNN



Aplicación del kernel

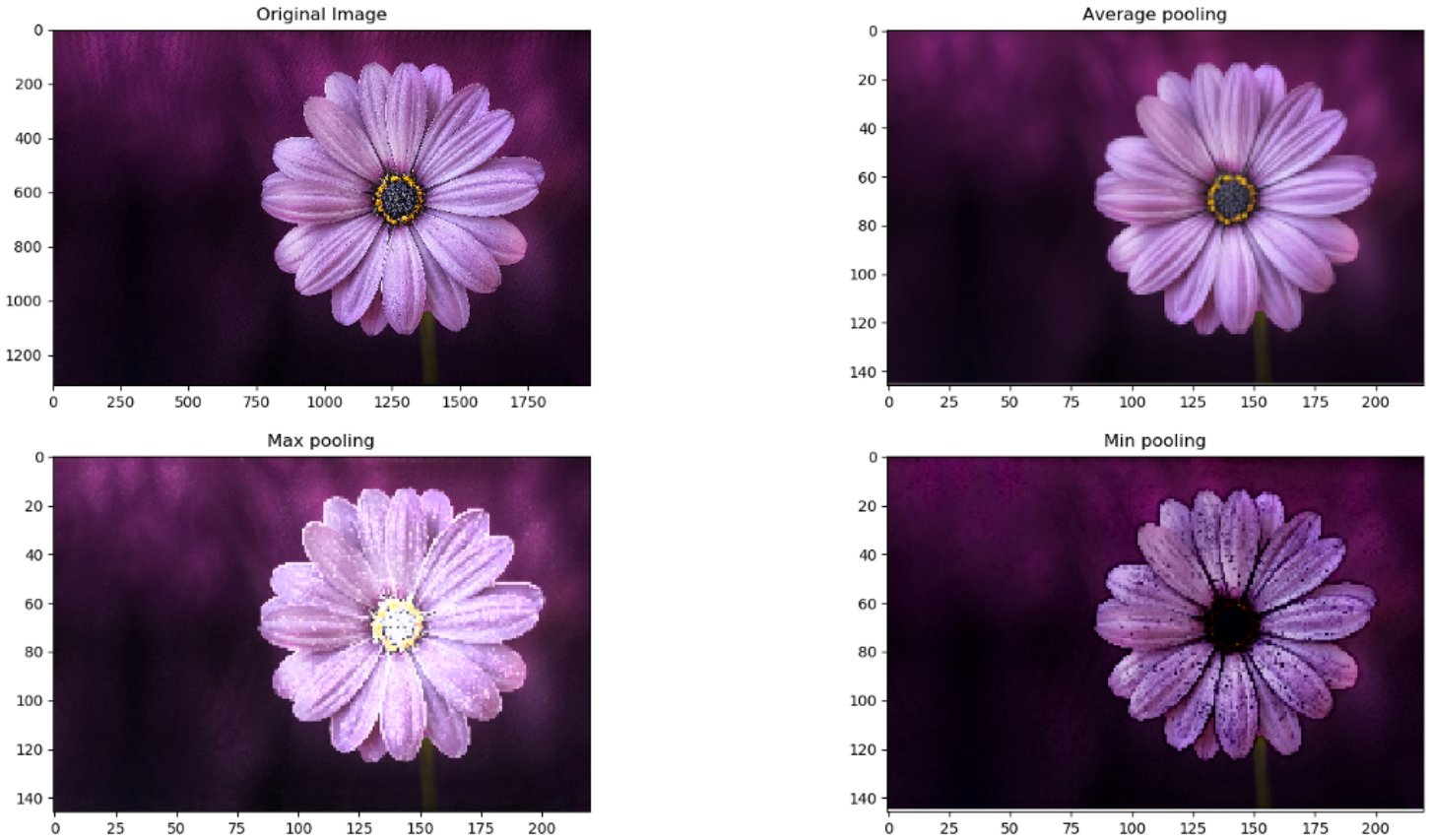
-

Pooling

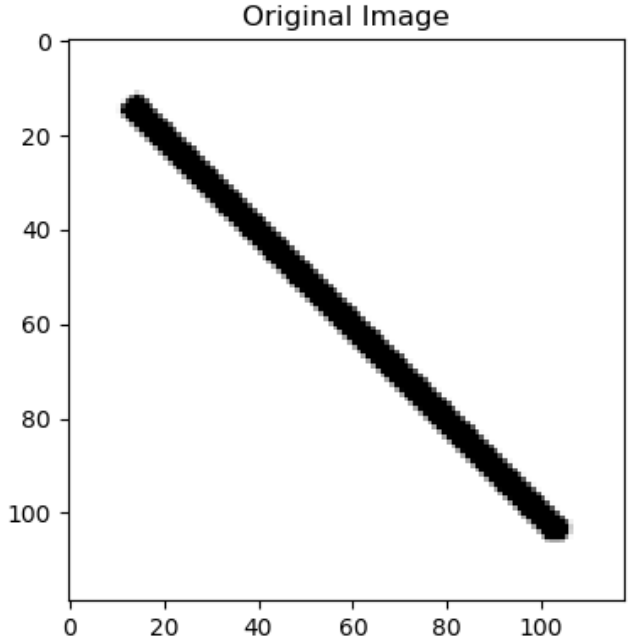
Su función es reducir progresivamente el tamaño espacial de la representación para reducir la cantidad de parámetros y cómputo en la red.

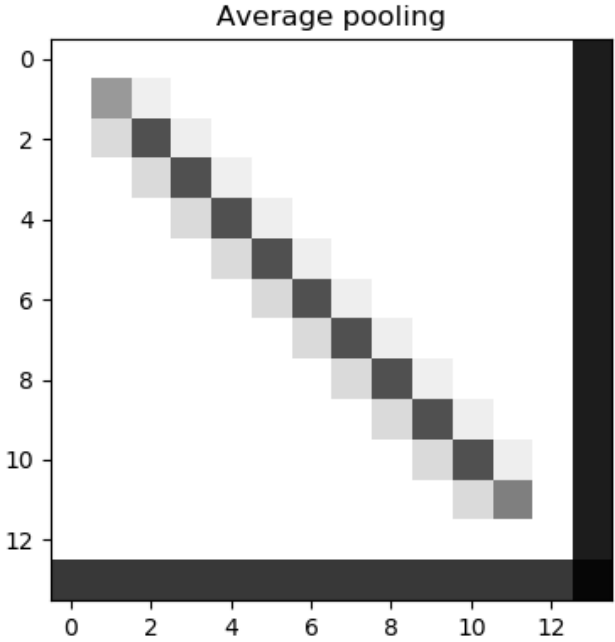
- Max Pooling
- Min Pooling
- Average Pooling

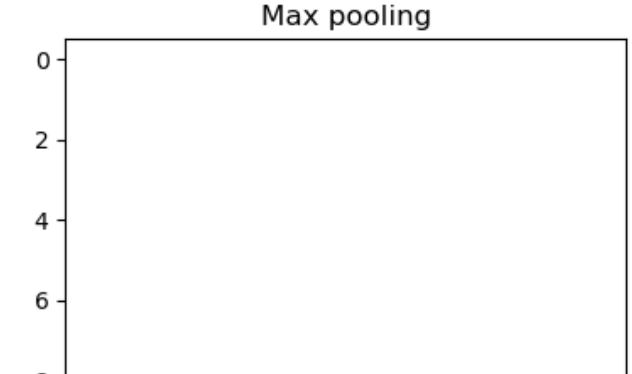
Pooling - ejemplo 1

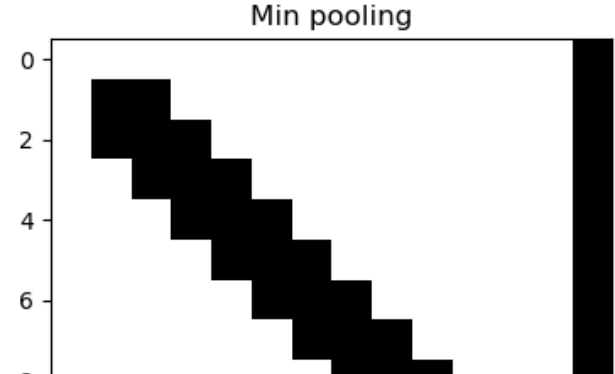


Pooling - ejemplo 2

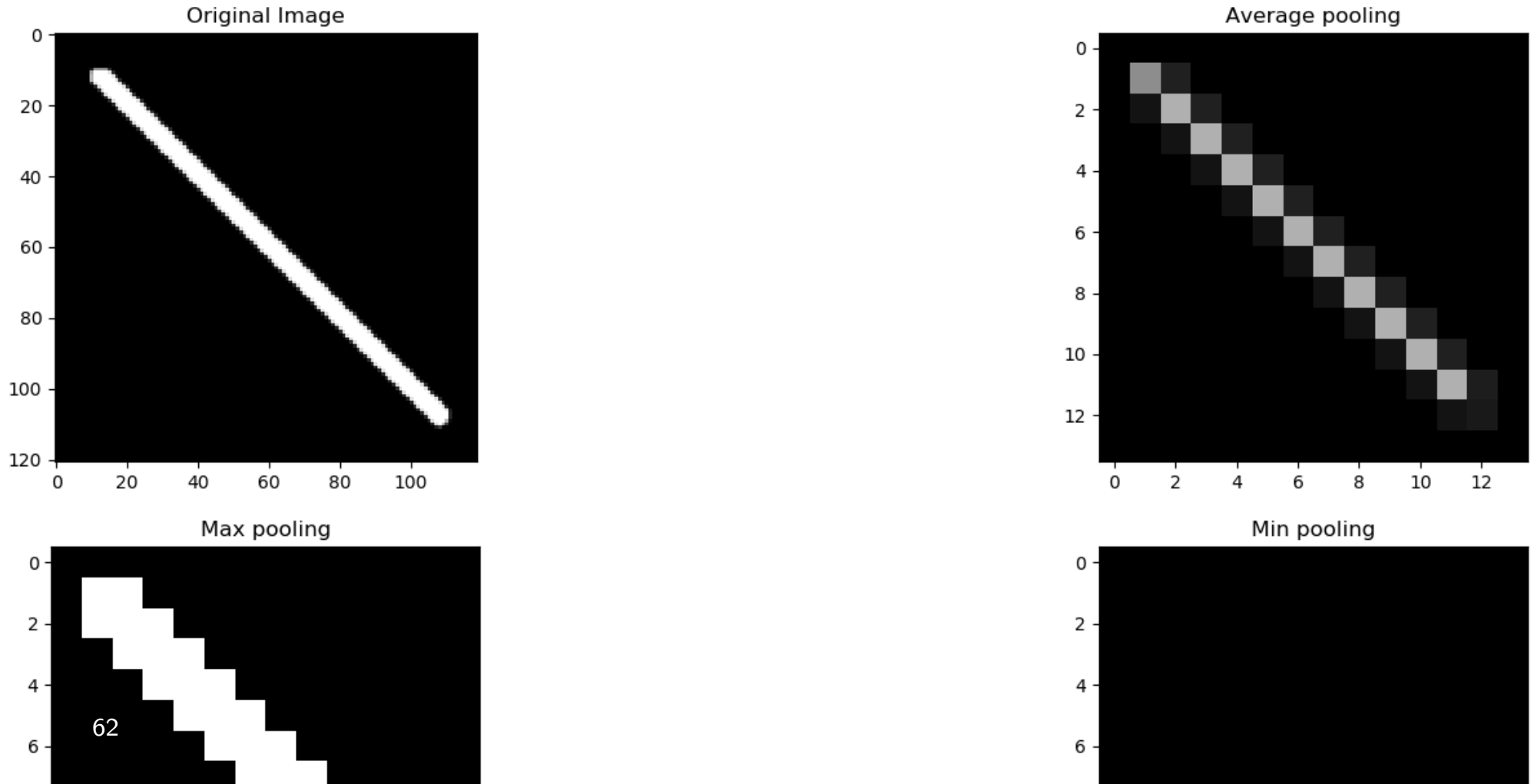








Pooling - ejemplo 3



Funciónes de Activación - Sigmoid

- Ampliamente utilizado en problemas de clasificación.
- Introduce la no linealidad.
- La derivada/gradiente se puede representar mediante sigmoide, lo que facilita el cálculo.
- Es una función continua y derivable en todos sus puntos.

- Un inconveniente importante de la función de activación sigmoidea es que sufre el problema del gradiente de fuga. No se preocupe si no sabe sobre el gradiente de fuga, lo explicaré. Se refiere a un punto en el que nuestro gradiente se vuelve minúsculo. Cuando el número de capas ocultas aumenta, durante la retropropagación, el producto de los gradientes se vuelve muy, muy pequeño, ya que el gradiente está en un

Funciónes de Activación - Sigmoid

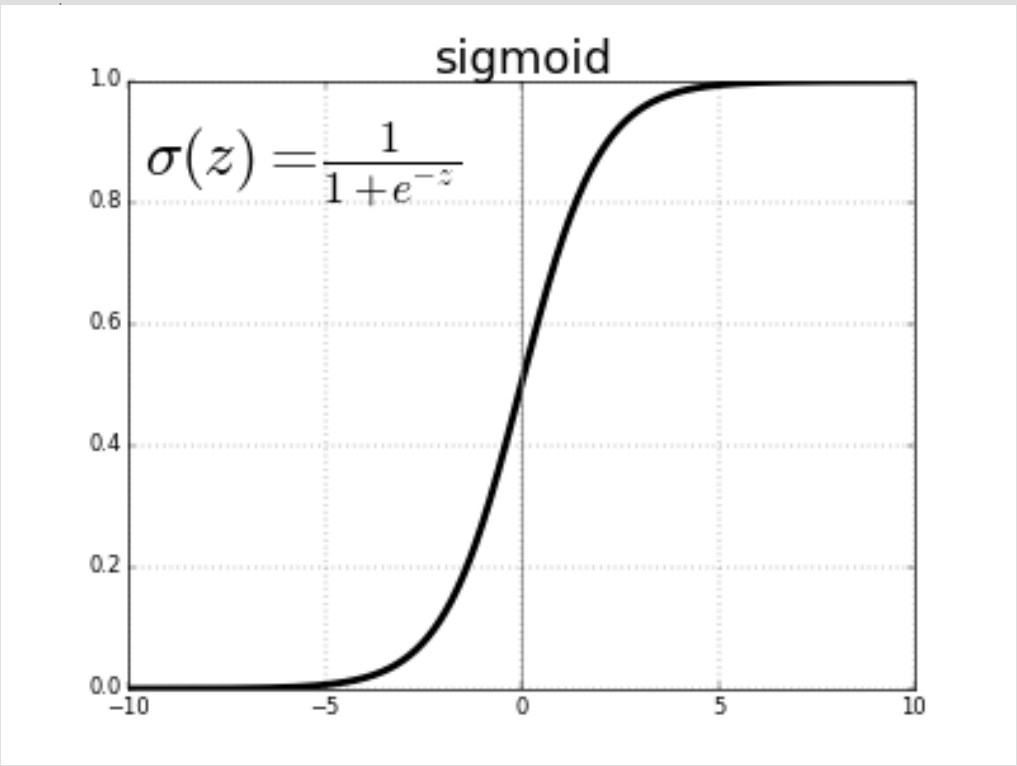


Figure: Función de activación sigmoide.

Funciónes de Activación - TanH

- Cero: función centrada.
- La derivada se puede representar en sus términos.
- Tanh es una función continua y es diferenciable en todos los puntos.
-

- También puede sufrir el problema de la desaparición del gradiente.

Funciónes de Activación - TanH

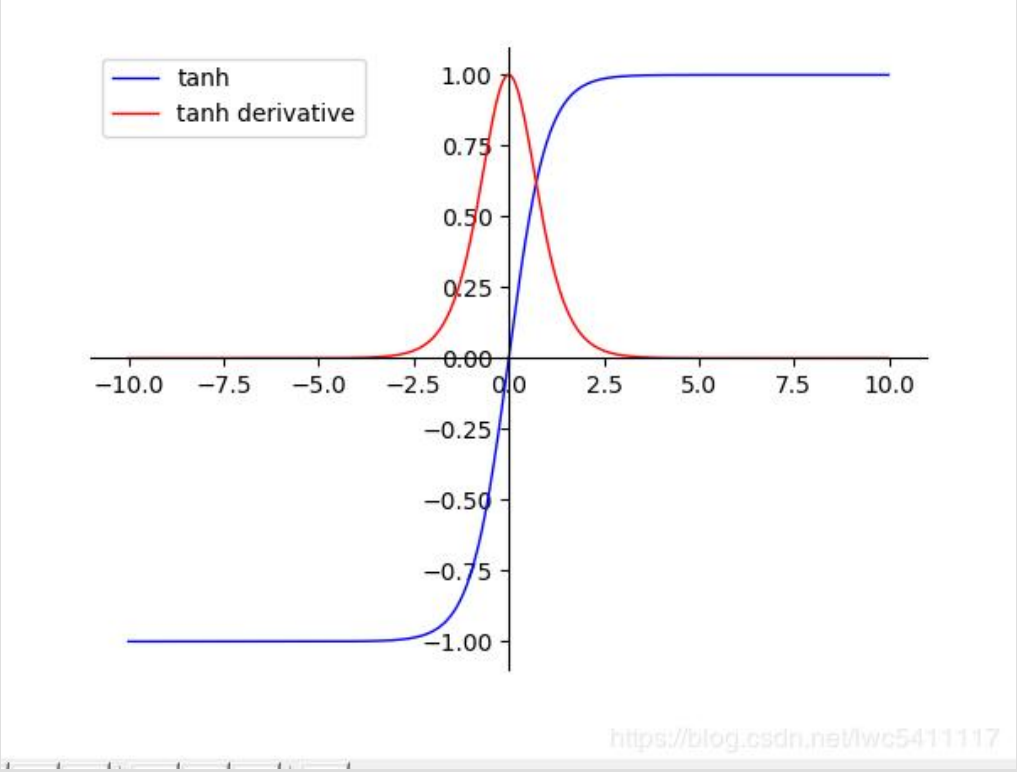


Figure: Función de activación TanH.

Funciónes de Activación - RELU

- Es computacionalmente eficiente, es decir; permite una rápida convergencia.
- También introduce la no linealidad.

- Sufre de un problema de activación muerta, también conocido como neuronas moribundas. Debido a que la salida para cualquier número no positivo es 0, lo que desactiva la neurona o podemos decir que la neurona está muerta, esto restringe el flujo de información de una capa a otra.

Funciónes de Activación - RELU

images/Funactivacin/RELU.png

Funciónes de Activación - Leaky RELU

- Superar el problema de las neuronas moribundas.
- Converger rápidamente También es una función no lineal.

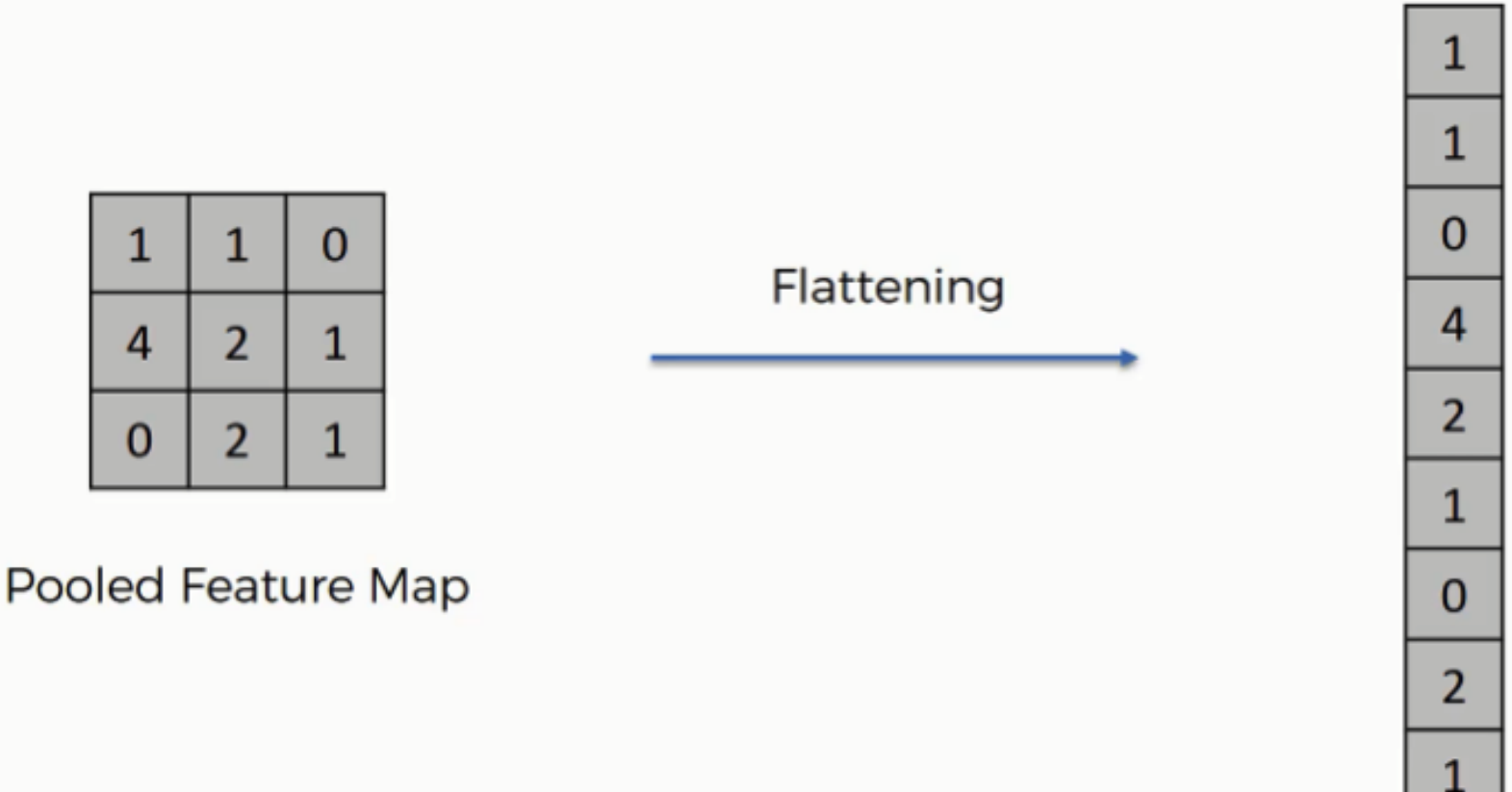
- Las predicciones no son consistentes para valores negativos.

Funciónes de Activación - Leaky RELU

images/Funactivacin/leakyRELU.png

Extrayendo Características: Flattening

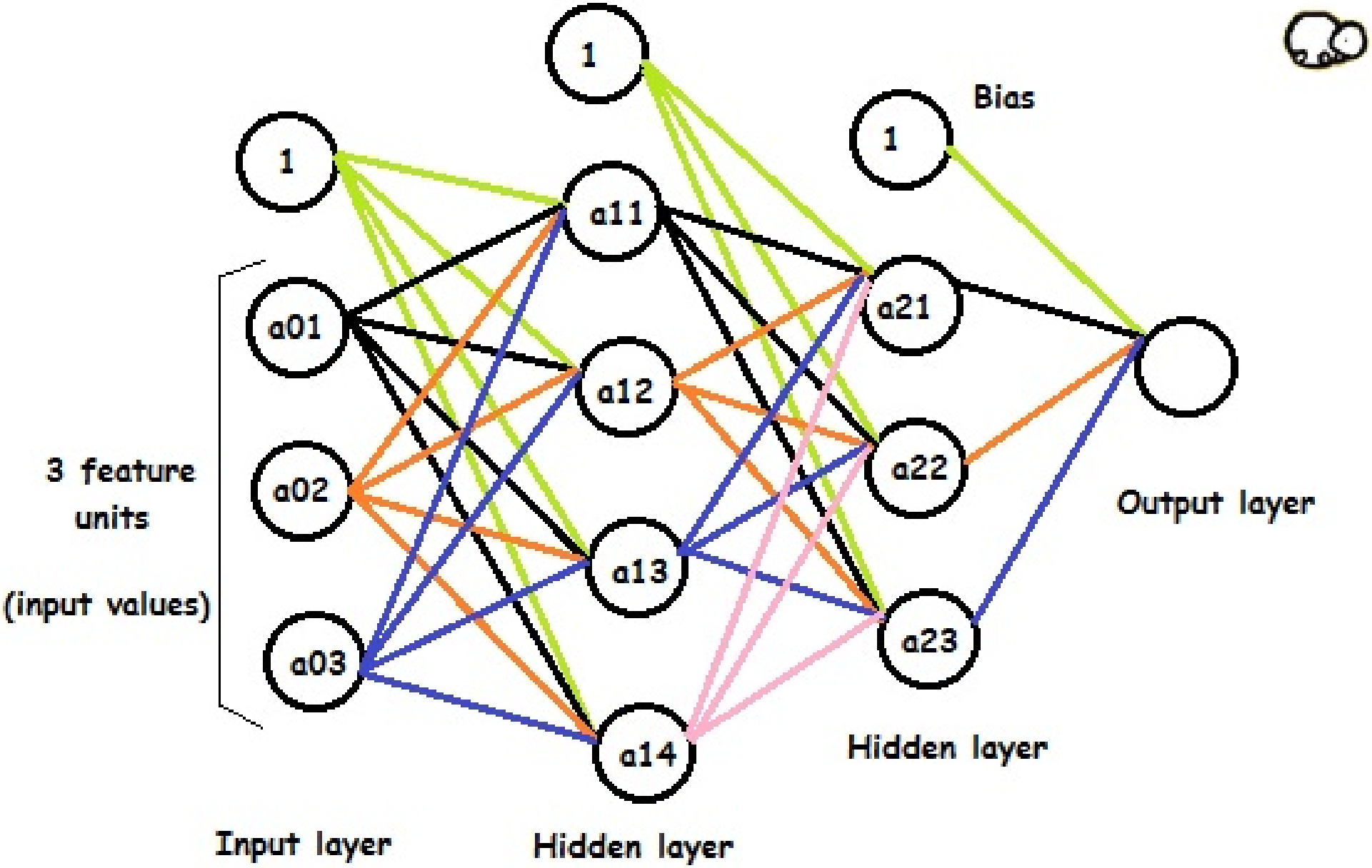
Es el aplanamiento de los datos, para usarlos como entrara en las capas de la Red Neuronal.



Flattening - Función de activación softmax

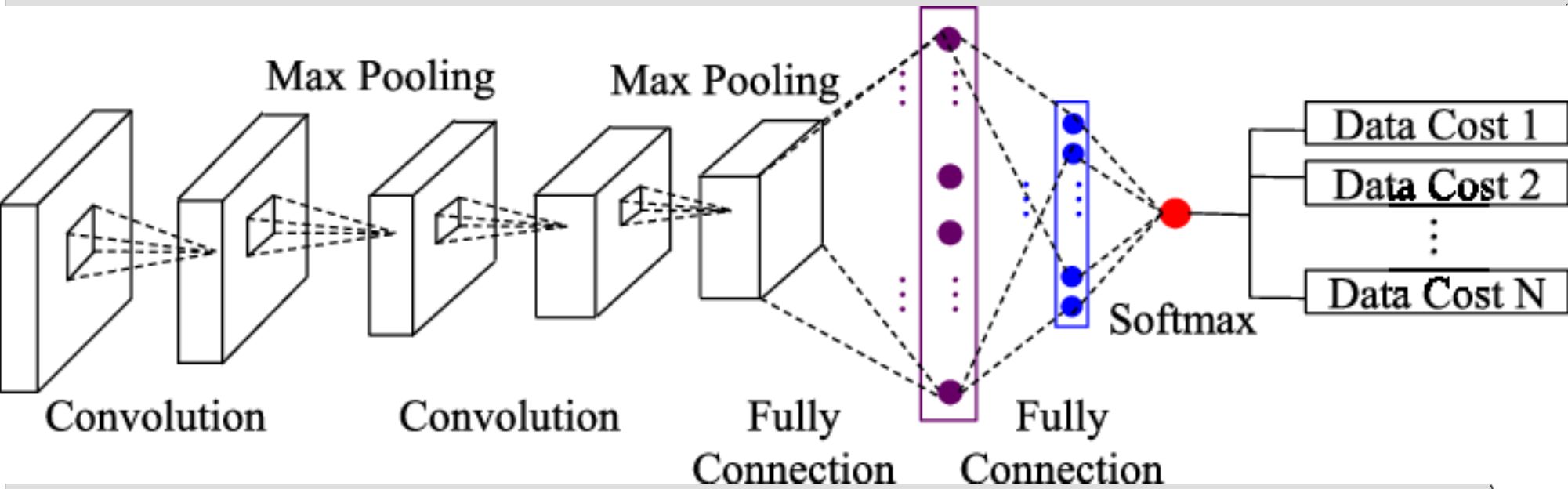
Es una Función de activación que pondera los pesos que tendrá cada características a una probabilidad.

Es la capa de la Red Neuronal, que clasifica la imagen.

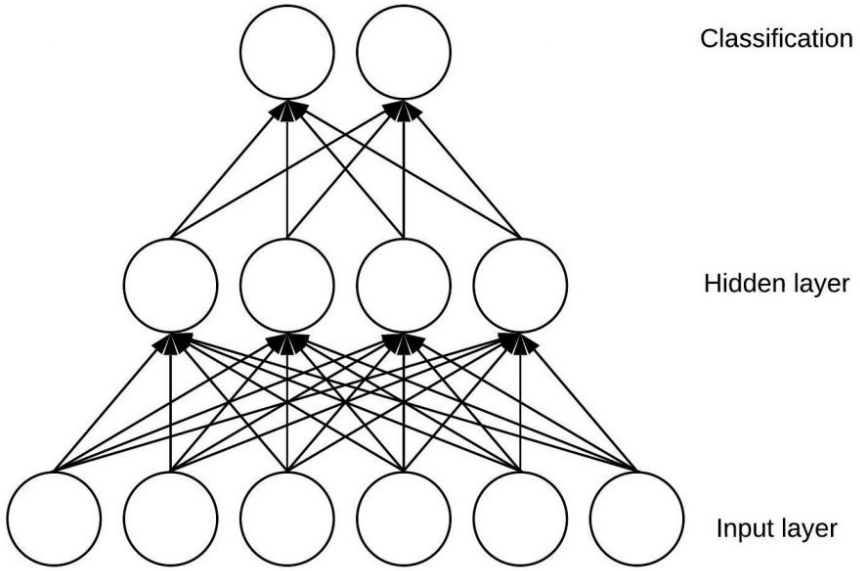


4 Neural Network with Fully Connected Layers

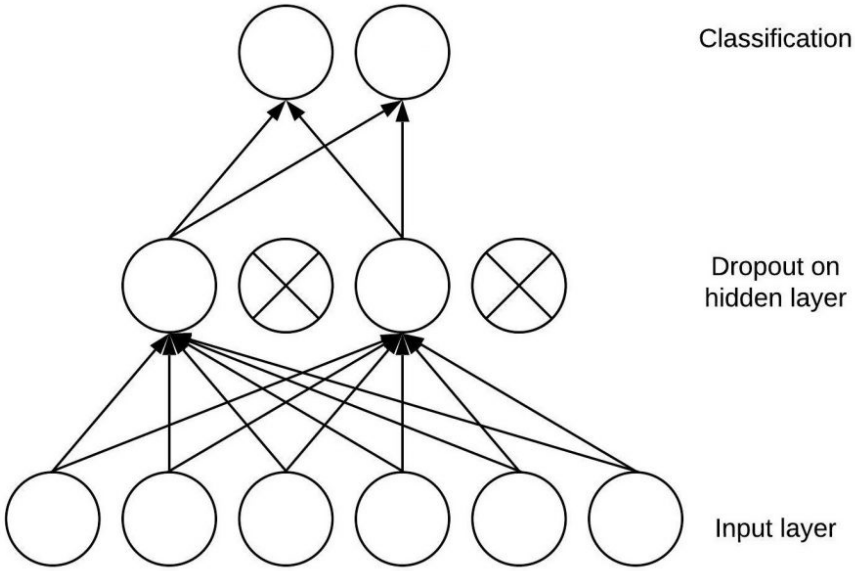
Arquitectura completa



Dropout

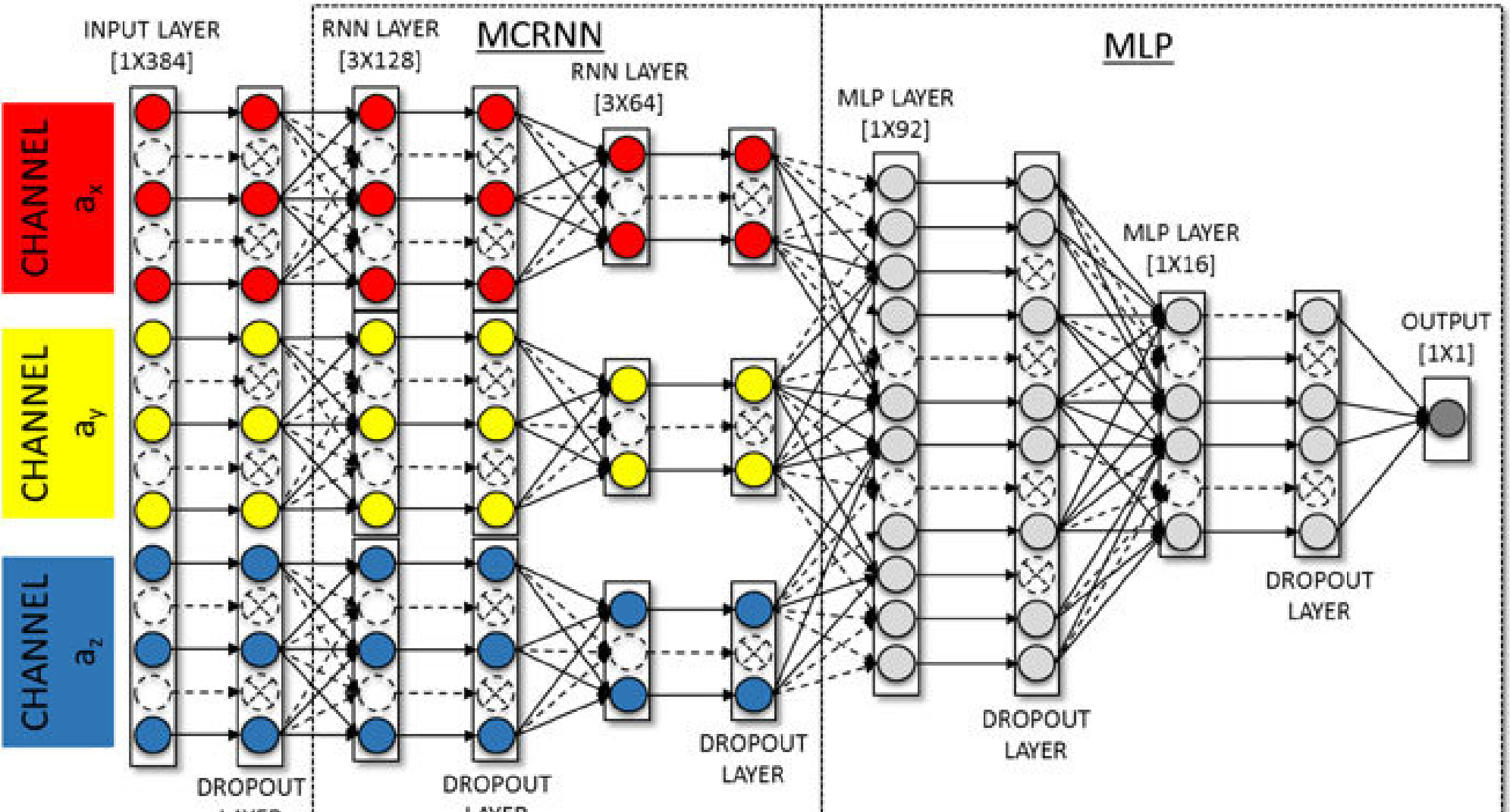


Without Dropout



With Dropout

Dropout - ejemplo de arquitectura



Forward Propagation

Es el proceso de izquierda a derecha, con el fin de obtener una predicción

Backpropagation

Es el proceso de derecha a izquierda, con el fin de entrenar a la CNN, de modo que cada capa se adapta sus pesos a su siguiente capa.

GIF Entrenamiento por back propagation

-

Transferencia de Aprendizaje

Una CNN puede aprender a clasificar nuevas clases, a partir de una existente.

Transferencia de Aprendizaje - visual

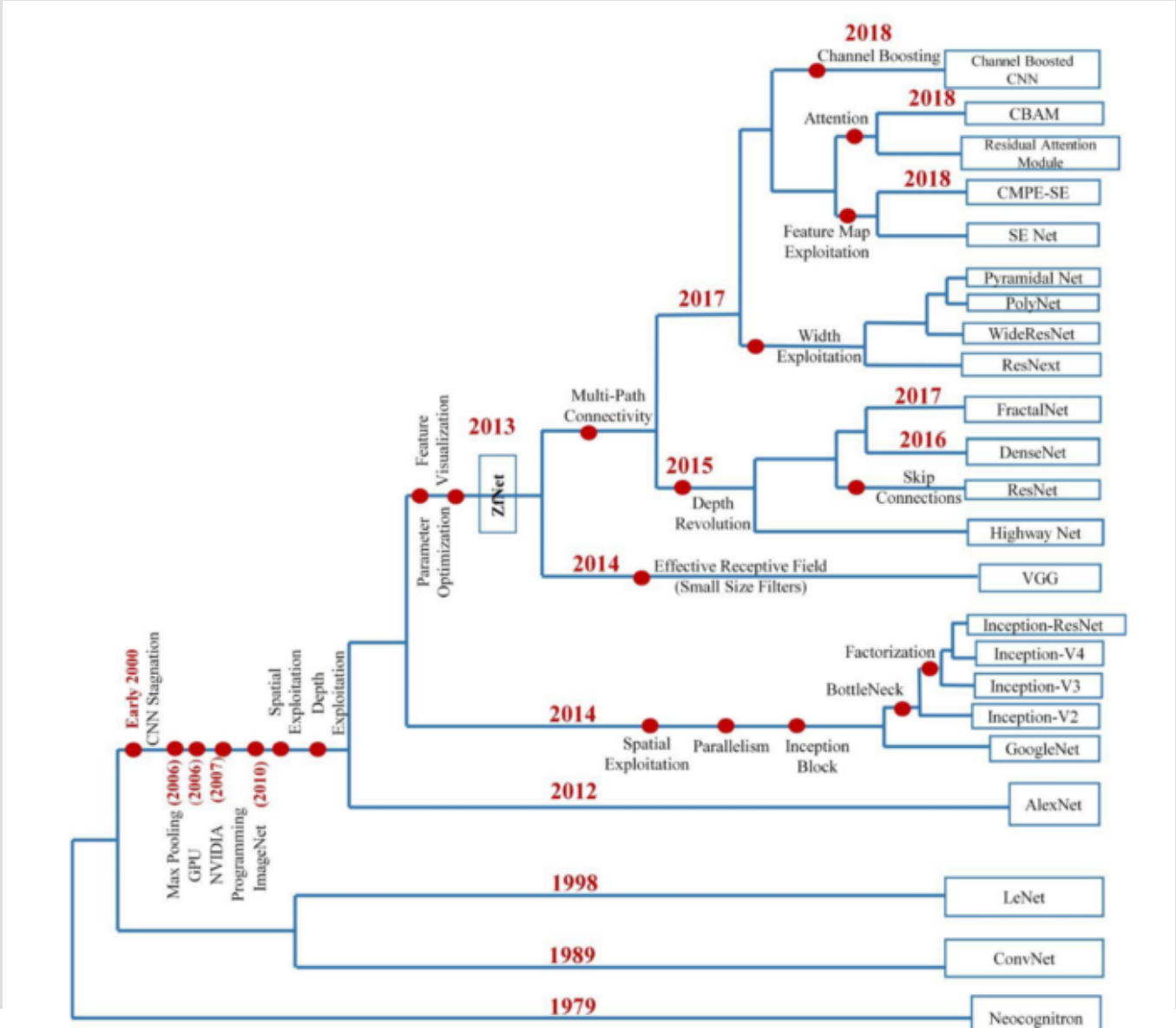
1. Entrenar en ImageNet

2. Si el conjunto de datos es pequeño: corrija todas las ponderaciones (trate a CNN como extractor de características fijas), reentrene solo al clasificador
o sea cambie la capa Softmax al final

3. Si el conjunto de datos es medio, utulize los pesos antiguos como inicializacion:

reentrene una mayor parte de la red, o incluso toda ella.

CNNs famosas



Inicios

CNN tiene sus raíces en el Neocognitron de Fukushima, este se utilizó para el reconocimiento de caracteres escritos a mano en japonés y otras tareas de reconocimiento de patrones.

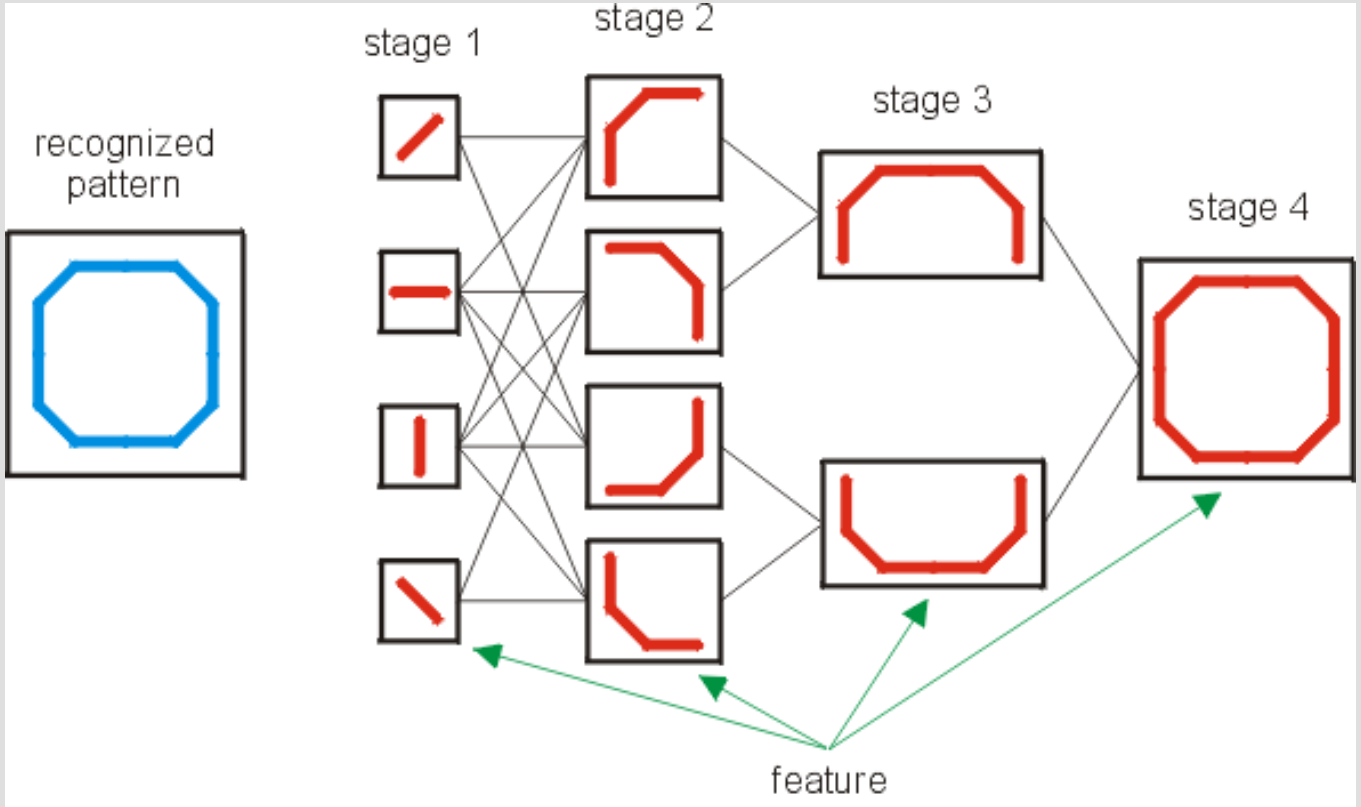


Figure: Principio de la extracción jerárquica de características de

ConvNet

LeCuN propuso un entrenamiento supervisado de ConvNet utilizando el algoritmo de retro-propagación, el trabajo de LeCuN sentó las bases para las CNN 2D modernas.

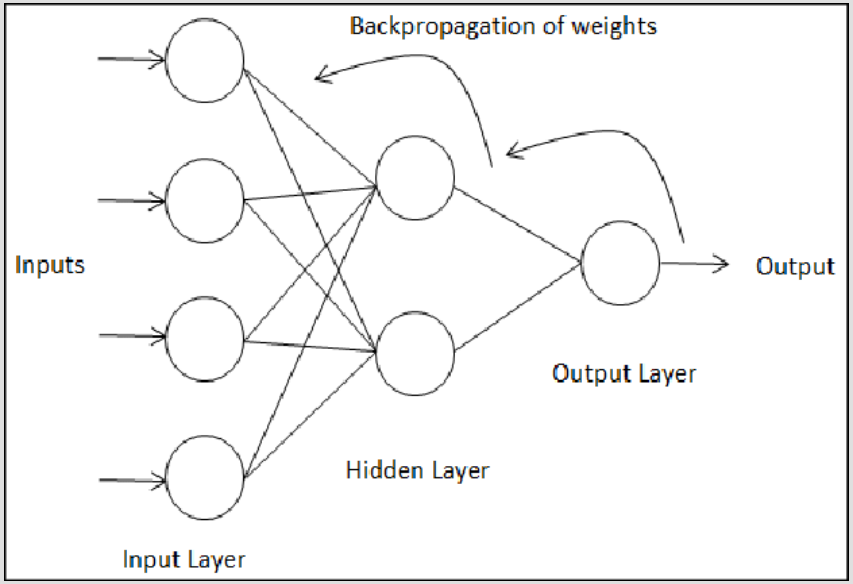


Figure: Backpropagation.

LeNet

En 1998, LeCuN propuso una versión mejorada de ConvNet, conocida como LeNet-5, y comenzó el uso de CNN para clasificar caracteres en Aplicaciónes relacionadas con el reconocimiento de documentos.

Figure: Se usó comercialmente para reconocimiento óptico de caracteres y huellas dactilares en cajeros automáticos y bancos.

AlexNet

La disponibilidad de extensos datos de capacitación y avances en hardware son los factores que contribuyeron al avance en la investigación de CNN. Sus principales características: Es más profundo y más ancho que LeNet, utiliza Relu, dropout y overlap Pooling.

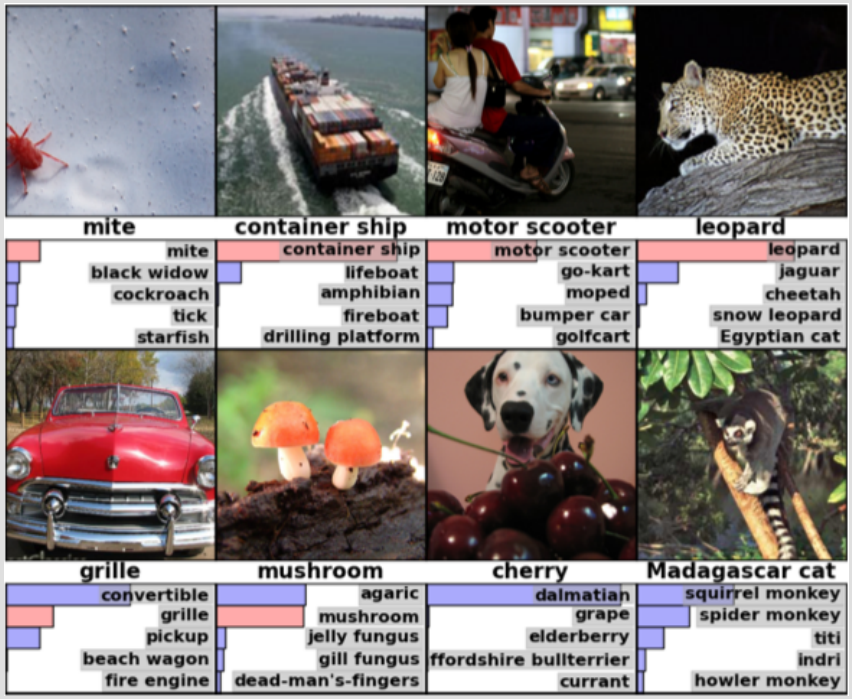


Figure: El reconocimiento de imagenes mejoró notablemente con AlexNet.

GoogleNet

El objetivo principal de la arquitectura de GoogleNet era lograr una alta precisión con un costo computacional reducido. Introdujo el nuevo concepto de inception block en CNN, mediante el cual incorpora transformaciones convolucionales de múltiples escalas utilizando la idea de dividir, transformar y fusionar.

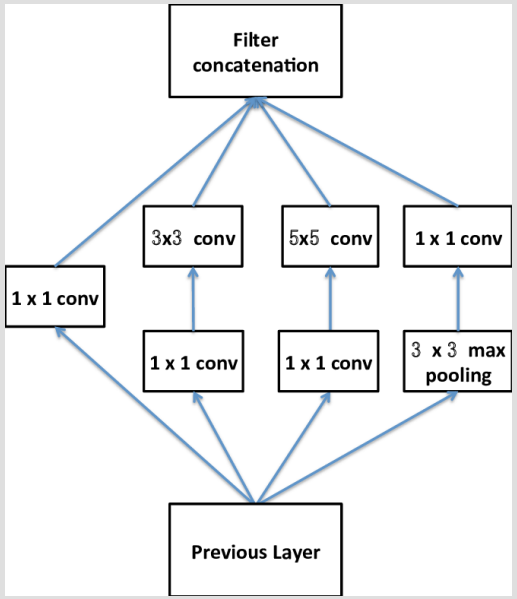


Figure: Arquitectura básica de inception block que muestra el concepto de división, transformación y fusión.

ResNet

ResNet introdujo el concepto de aprendizaje residual en las CNN y se coloca bajo las CNN basadas en rutas múltiples.

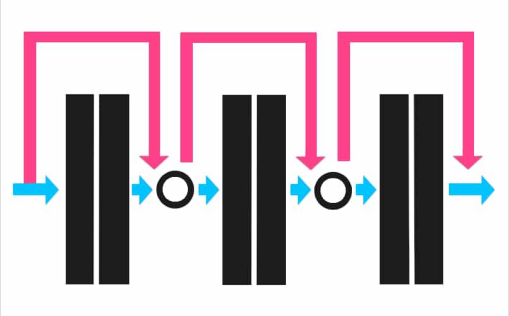


Figure: Las "conexiones atajo" coloreadas de magenta permiten a la información "saltarse" bloques de capas denominados por ello "bloques residuales".

DenseNet

DenseNet concatena las Funciónes de la capa anterior en lugar de agregarlas, la red puede obtener la capacidad de diferenciar explícitamente entre la información que se agrega a la red y la información que se conserva.

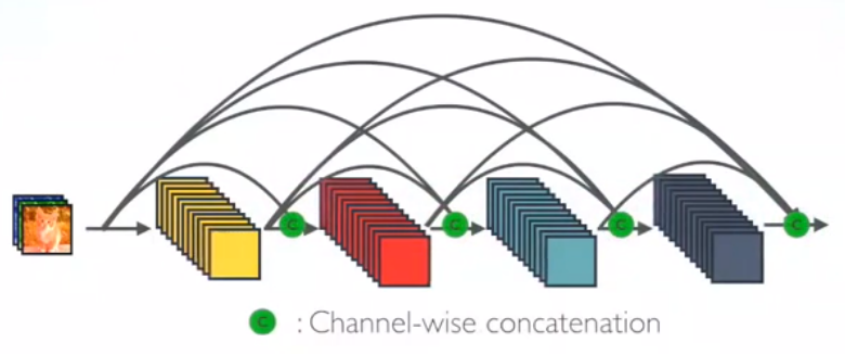


Figure: Dense Block in DenseNet.

SENet

Se propuso un nuevo bloque para la selección de mapas de características relevantes. Este nuevo bloque se denominó SE-block, que suprime los mapas de características menos importantes, pero otorga un alto peso a la clase que especifica los mapas de características.

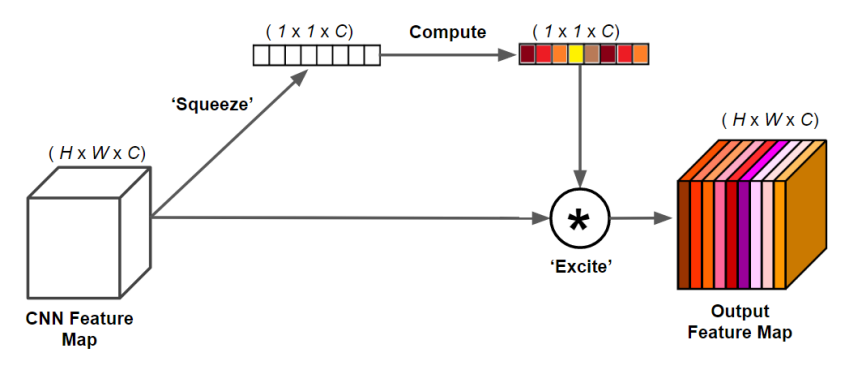


Figure: SE Block Visualization.

Gratidão!
Pergun
e
Comenta

Referencias Bibliogrŕficas I