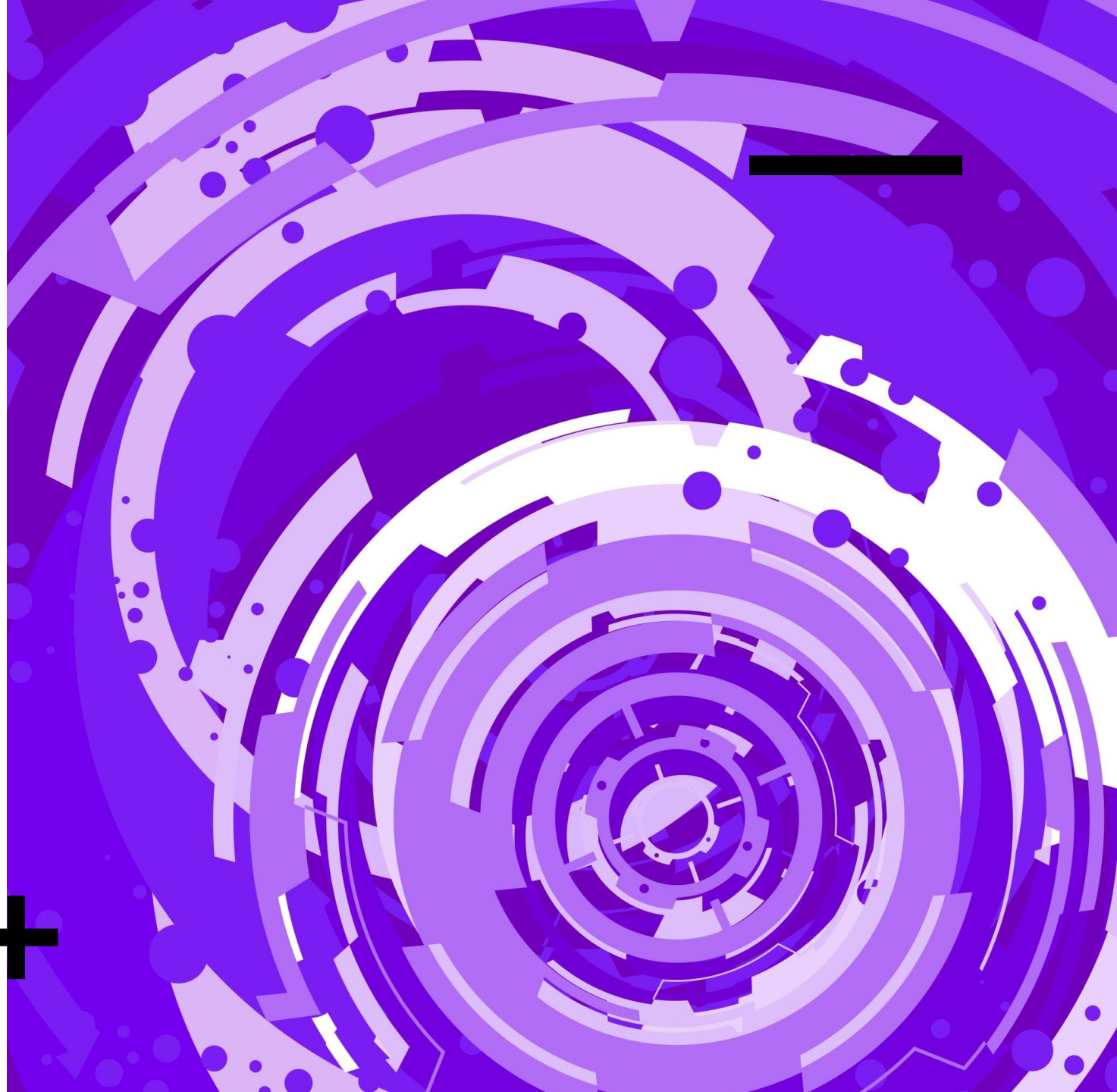


Redes Neuronales Convolucionales Equivariantes a Grupos (G-CNNs)

Luis Raúl Figueroa Martínez

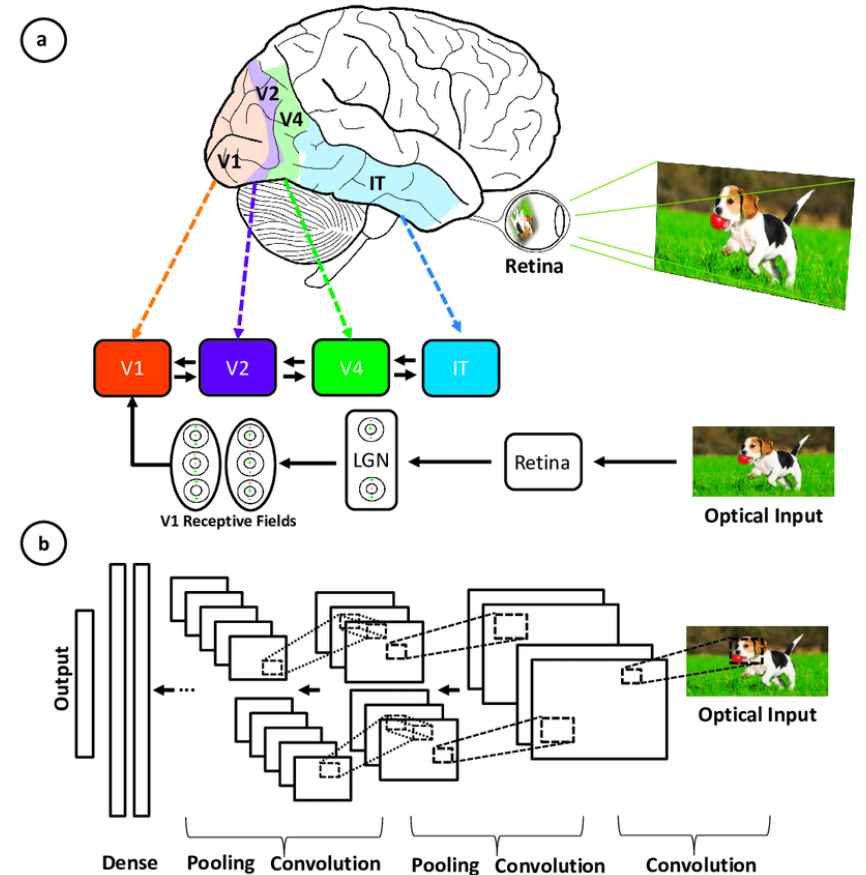


Contexto de las CNNs

Las **Redes Neuronales Convolucionales** (CNNs) son un tipo de red neuronal especialmente diseñada para el **procesamiento de datos estructurados en forma de cuadrícula**, como las imágenes.

Las CNNs se inspiran en el **corte visual** de los mamíferos, donde neuronas específicas reaccionan a patrones y características visuales (como bordes y formas).

(Ver Mariel)



La Convolución en una CNN:

- La **convolución** es el proceso mediante el cual la red aplica un conjunto de **filtros (kernels)** a una imagen para extraer características específicas.

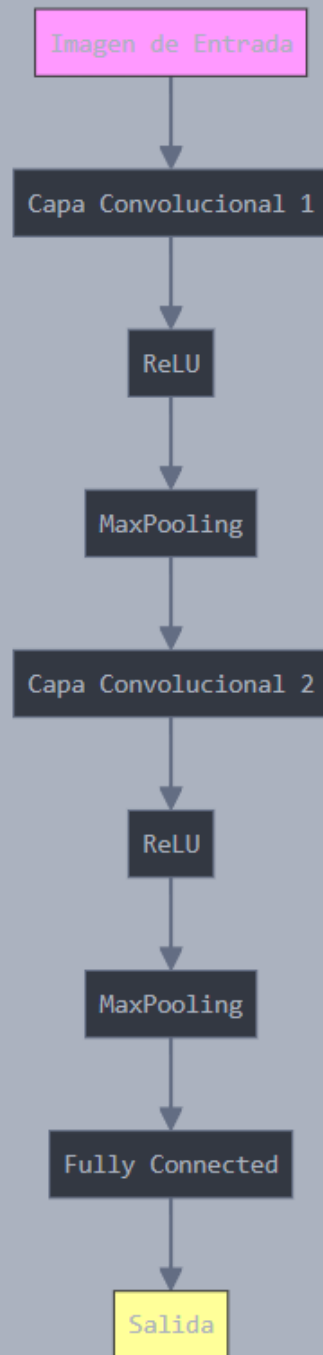
La misma característica puede aparecer en diferentes ubicaciones de una imagen. Los filtros convolucionales permiten detectar esa característica en cualquier posición, ya que recorren toda la imagen

3.1.2 Mapas de Características

La matriz resultante de aplicar un filtro a una imagen es conocido como **mapa de características** o **feature map**. Cada mapa de características resalta una propiedad específica de la imagen, como bordes horizontales, verticales, texturas, etc. Las primeras capas convolucionales tienden a aprender características básicas, mientras que las capas más profundas aprenden características más complejas, como formas y patrones completos.

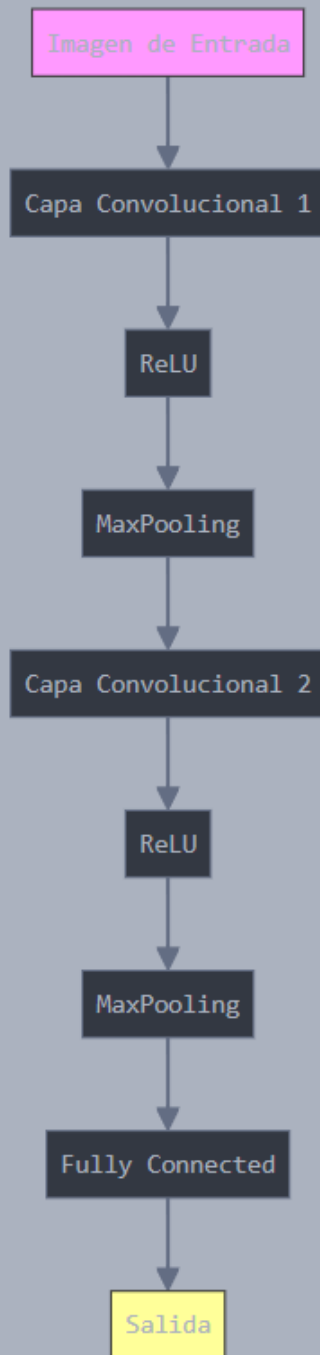


CNN Tradicional



Realiza una primera convolución sobre la imagen, aplicando filtros que detectan patrones básicos como bordes o texturas.

CNN Tradicional



- Realiza una primera convolución sobre la imagen, aplicando filtros que detectan patrones básicos como bordes o texturas.
- Introduce no linealidad, activando solo los valores positivos. Esto permite que la red modele relaciones no lineales en los datos.

Feature Map de Entrada

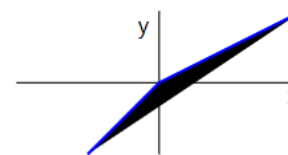
| | | |
|------|------|------|
| 2.5 | -1.2 | 0.8 |
| -0.5 | 1.7 | -2.1 |
| 0.3 | -0.9 | 1.4 |

ReLU
 $f(x) = \max(0, x)$

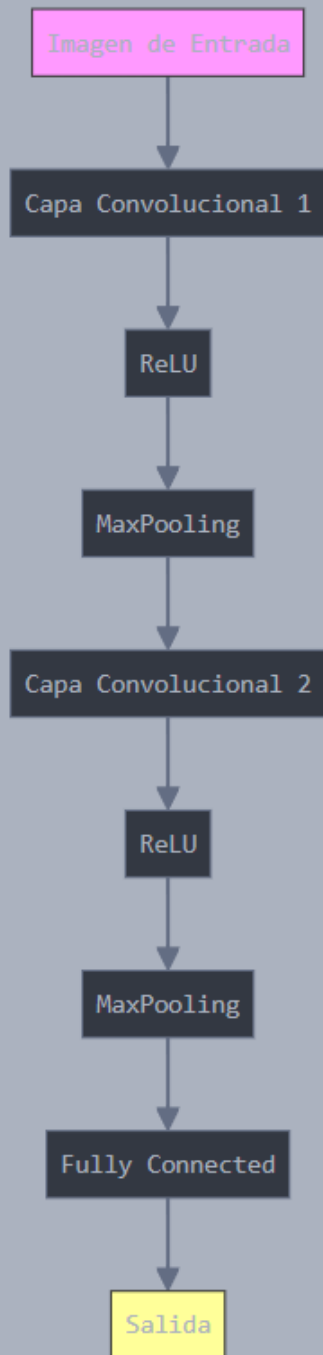
Feature Map después de ReLU

| | | |
|-----|-----|-----|
| 2.5 | 0.0 | 0.8 |
| 0.0 | 1.7 | 0.0 |
| 0.3 | 0.0 | 1.4 |

Función ReLU



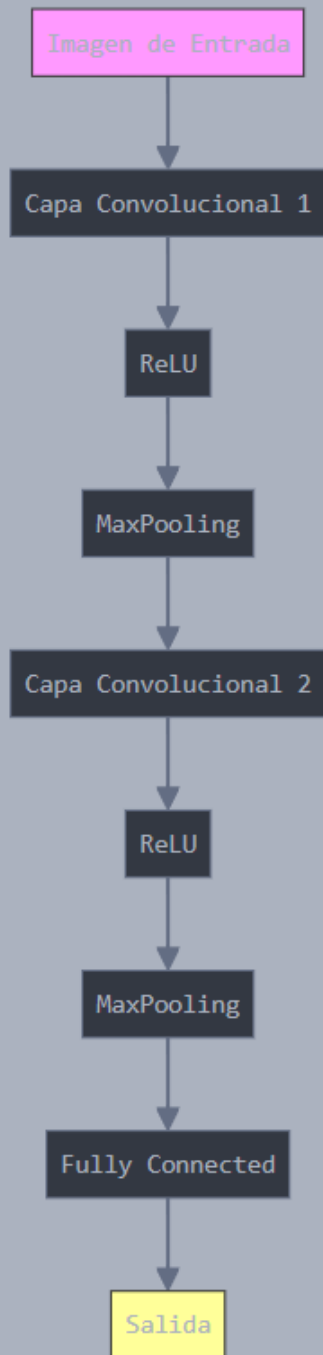
CNN Tradicional



- Realiza una primera convolución sobre la imagen, aplicando filtros que detectan patrones básicos como bordes o texturas.
- Introduce no linealidad, activando solo los valores positivos. Esto permite que la red modele relaciones no lineales en los datos.
- Reduce la resolución espacial de los mapas de características, ayudando a reducir el tamaño de los datos mientras mantiene la información relevante.



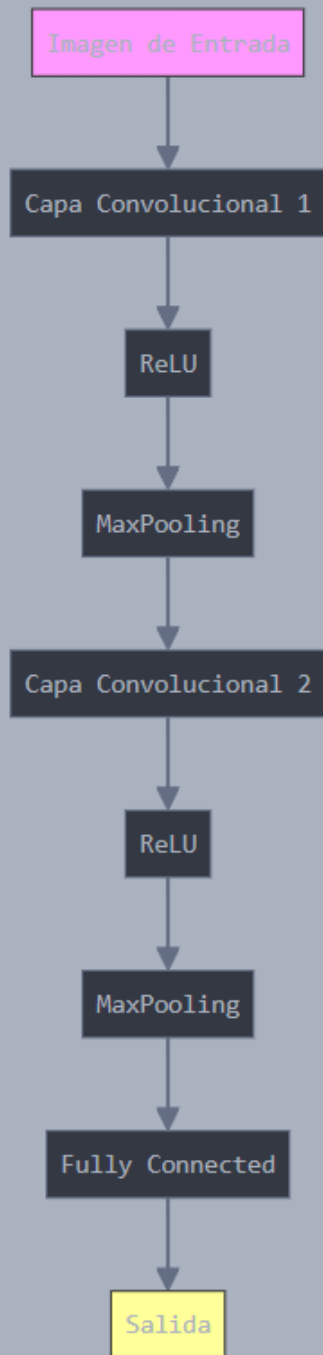
CNN Tradicional



- Realiza una primera convolución sobre la imagen, aplicando filtros que detectan patrones básicos como bordes o texturas.
- Introduce no linealidad, activando solo los valores positivos. Esto permite que la red modele relaciones no lineales en los datos.
- Reduce la resolución espacial de los mapas de características, ayudando a reducir el tamaño de los datos mientras mantiene la información relevante.
- La salida de la red CNN, que puede ser la clasificación, la detección, o cualquier otra tarea según el propósito del modelo.

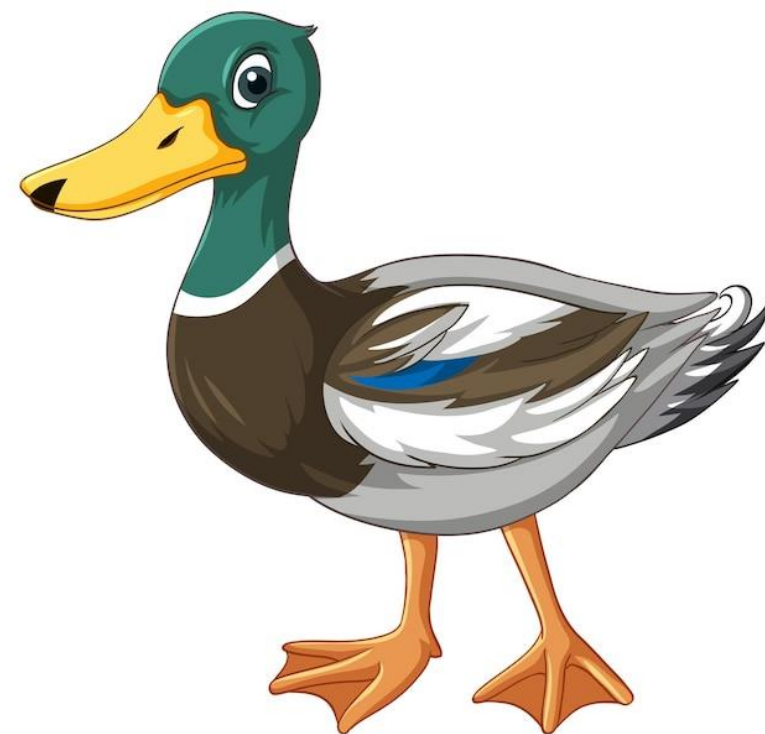
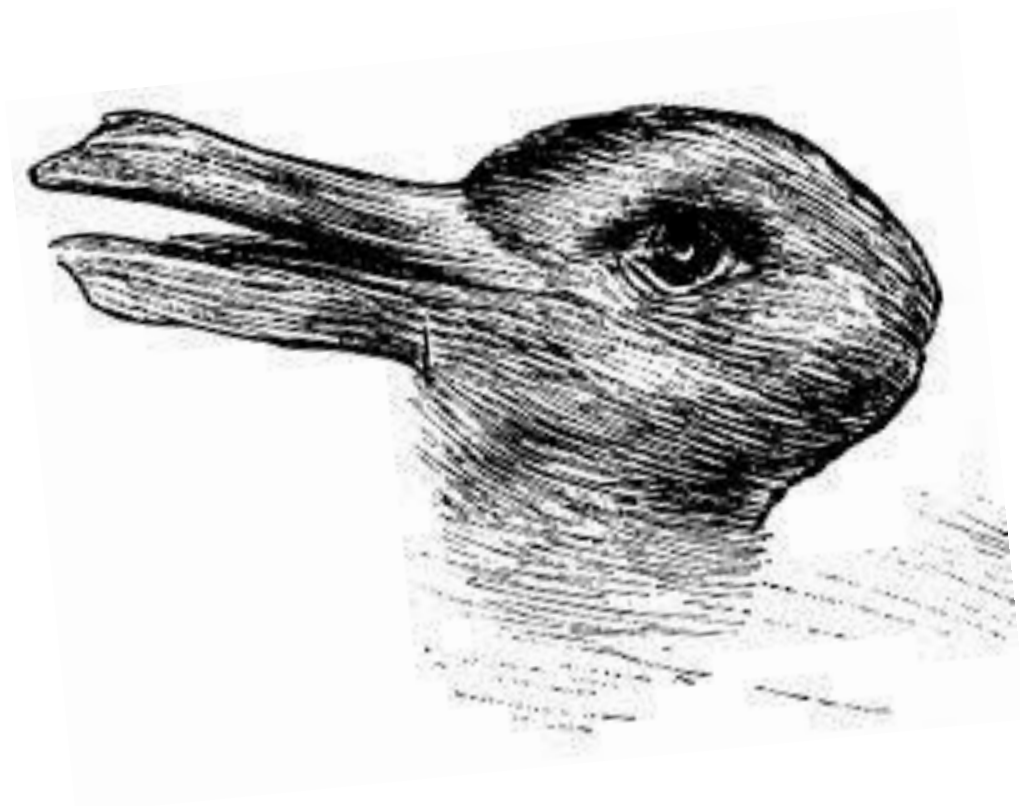


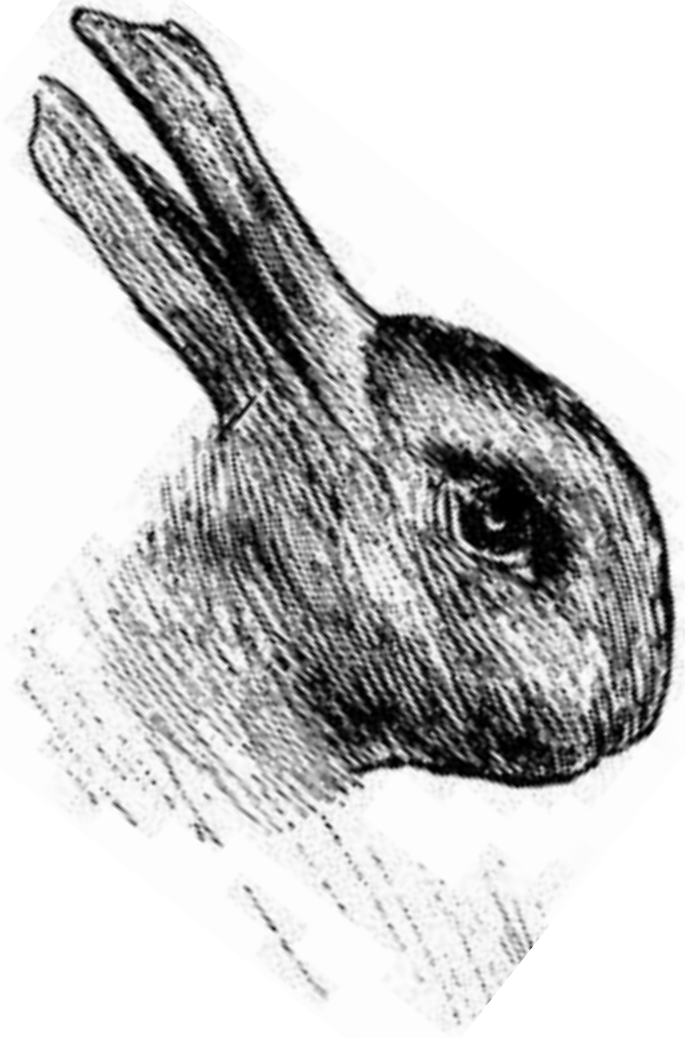
CNN Tradicional



- Realiza una primera convolución sobre la imagen, aplicando filtros que detectan patrones básicos como bordes o texturas.
- Introduce no linealidad, activando solo los valores positivos. Esto permite que la red modele relaciones no lineales en los datos.
- Reduce la resolución espacial de los mapas de características, ayudando a reducir el tamaño de los datos mientras mantiene la información relevante.
- Capa que conecta todas las características aprendidas con el objetivo final de la red, como una clasificación o detección.
- La salida de la red CNN, que puede ser la clasificación, la detección, o cualquier otra tarea según el propósito del modelo.









Conceptos importantes



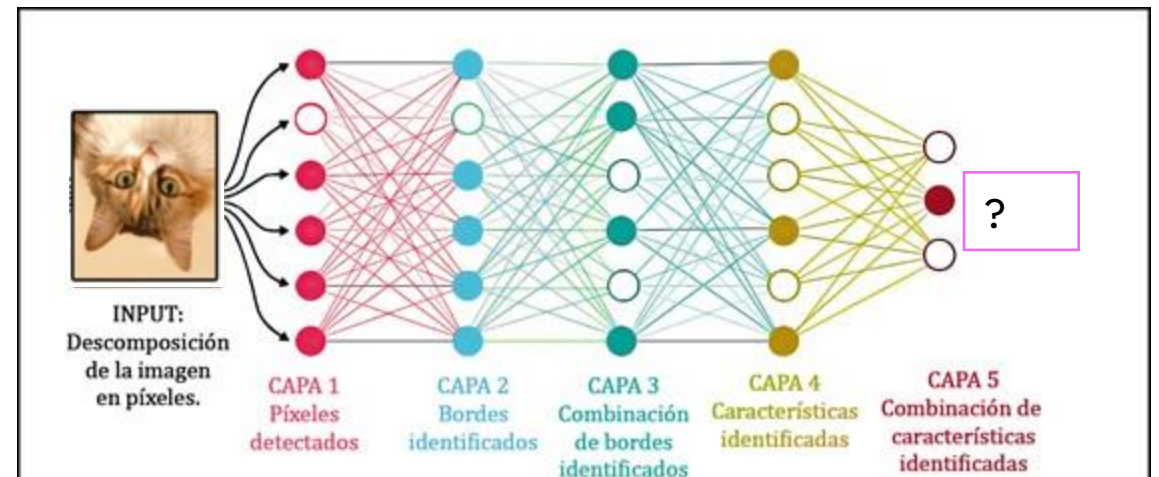
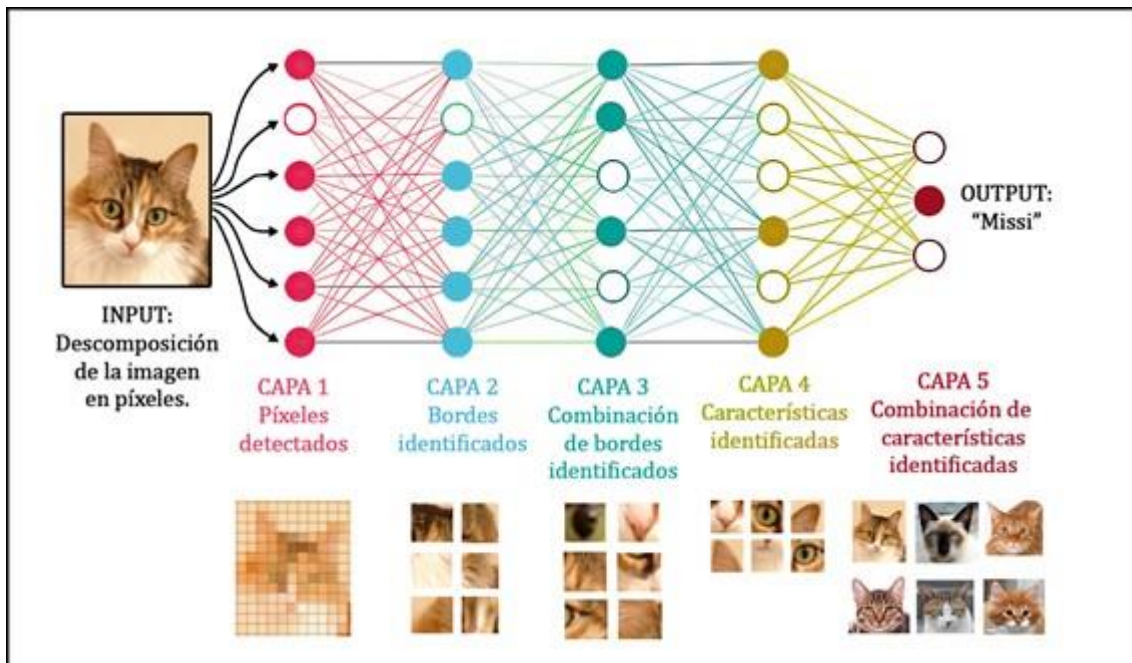
la **equivarianza** es la capacidad de la red para **responder de manera predecible** ante ciertas transformaciones de la entrada.

Por ejemplo, si se rota una imagen de un objeto, el **mapa de características** de las primeras capas también rota de manera equivalente, pero sigue representando las características originales.

La **invariancia** se refiere a que, a nivel de clasificación, la red puede **reconocer el mismo objeto sin importar su orientación**. Aunque el objeto esté rotado o reflejado, la red debería clasificarlo en la misma categoría.

Las CNNs convencionales no son invariantes a rotaciones y reflexiones

Aunque las CNNs tradicionales son invariantes a la **traslación** no son invariantes a **rotaciones o reflexiones**.

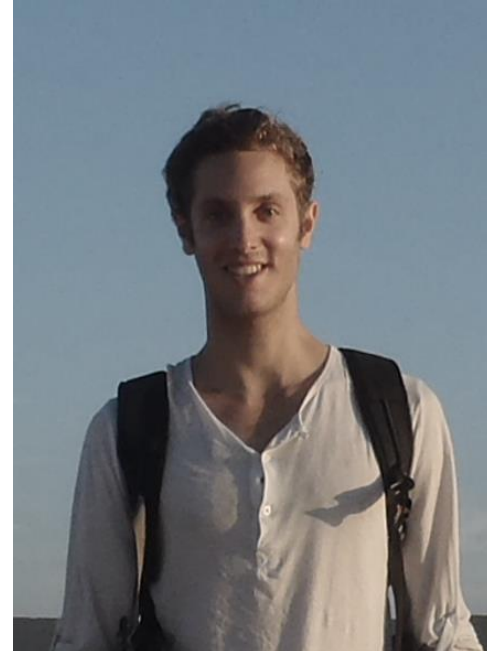


Y se hizo la luz ... (G-CNNs)

Las Redes Convolucionales Grupales (G-CNNs) fueron desarrolladas para abordar esta limitación al **incorporar la simetría en la estructura de la red**.

En lugar de aprender por separado cada orientación o simetría, las G-CNNs **incorporan grupos de simetría** (como rotaciones y reflexiones) en el proceso de convolución, haciendo que la red sea inherentemente capaz de procesar transformaciones.

Max Welling y Taco Cohen, investigadores en el campo de la inteligencia artificial y el aprendizaje profundo, introdujeron el concepto de G-CNNs en **2016**.



Conceptos importantes



la **equivarianza** es la capacidad de la red para **responder de manera predecible** ante ciertas transformaciones de la entrada.

Por ejemplo, si se rota una imagen de un objeto, el **mapa de características** de las primeras capas también rota de manera equivalente, pero sigue representando las características originales.

La **invariancia** se refiere a que, a nivel de clasificación, la red puede **reconocer el mismo objeto sin importar su orientación**. Aunque el objeto esté rotado o reflejado, la red debería clasificarlo en la misma categoría.

Conceptos importantes

- Un grupo de simetría es un conjunto de transformaciones que incluye operaciones como rotaciones, reflexiones o traslaciones, y que mantienen algunas propiedades de la imagen.



$$r \begin{array}{|c|} \hline \text{cactus} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{rotated cactus} \\ \hline \end{array}$$

$$m \begin{array}{|c|} \hline \text{cactus} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{cactus} \\ \hline \end{array}$$

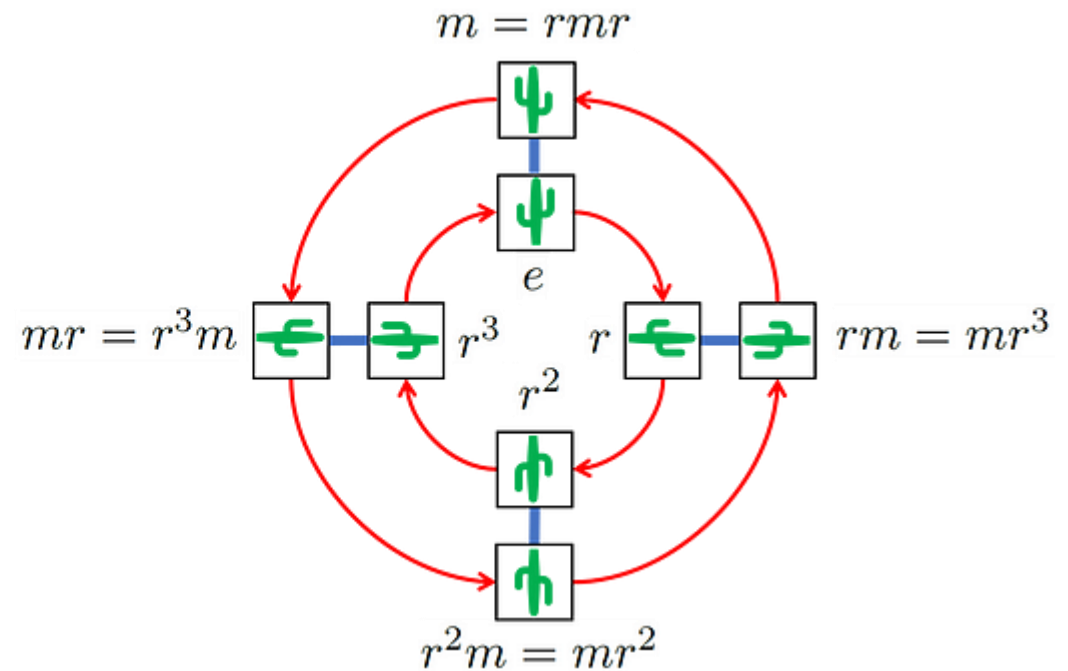
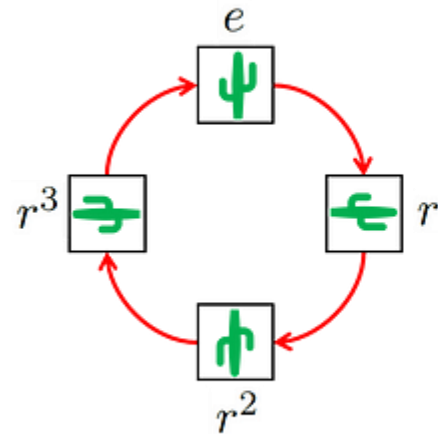
$$r^2 m \begin{array}{|c|} \hline \text{cactus} \\ \hline \end{array} = r^2 \begin{array}{|c|} \hline \text{cactus} \\ \hline \end{array} = r \begin{array}{|c|} \hline \text{rotated cactus} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{rotated cactus} \\ \hline \end{array}$$



Conceptos importantes

- Ejemplos de grupos de simetría:**Grupo p4**: Incluye rotaciones de 90 grados (0°, 90°, 180° y 270°).

- Grupo p4m**: Incluye rotaciones de 90 grados y reflejos. En este caso, la red puede aprender a detectar patrones que sean **equivariantes** a rotaciones y reflexiones.



Las G-convoluciones



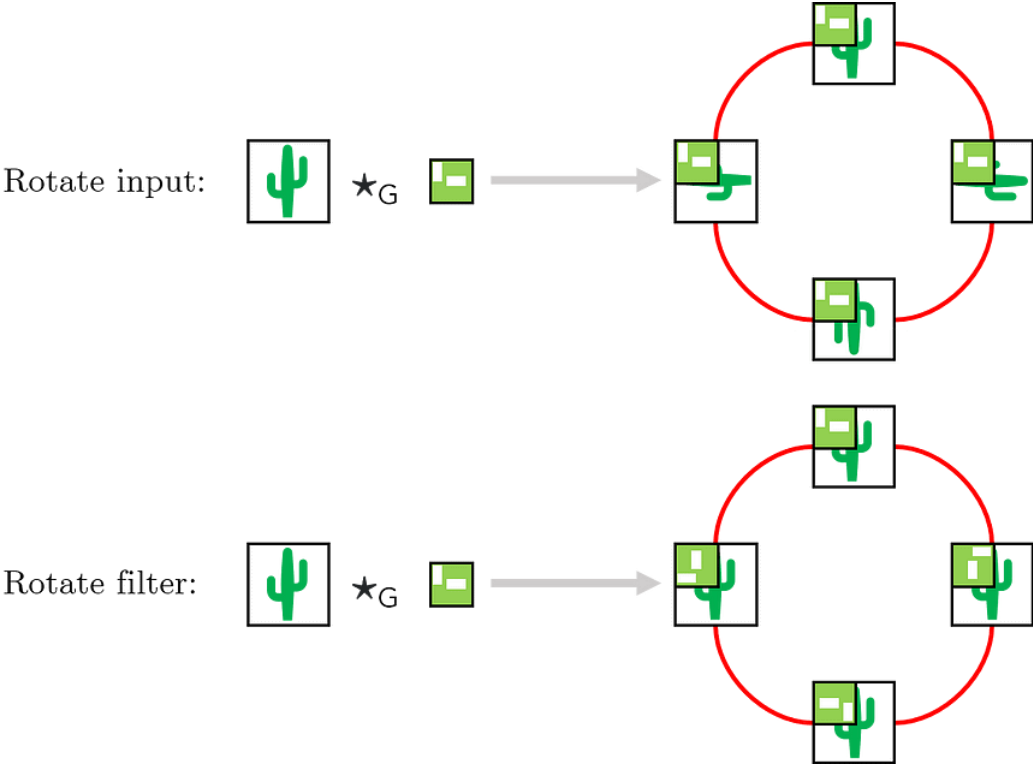
Las **G-Convoluciones** (Group Convolutions) son una extensión de las convoluciones tradicionales en redes neuronales.

En las convoluciones estándar, los filtros se aplican solo en distintas posiciones de la imagen.

En cambio, en las G-Convoluciones, los filtros se aplican en distintas posiciones **y** en diferentes transformaciones de grupo, como rotaciones y reflexiones de la imagen.



Las G-convoluciones en grupos

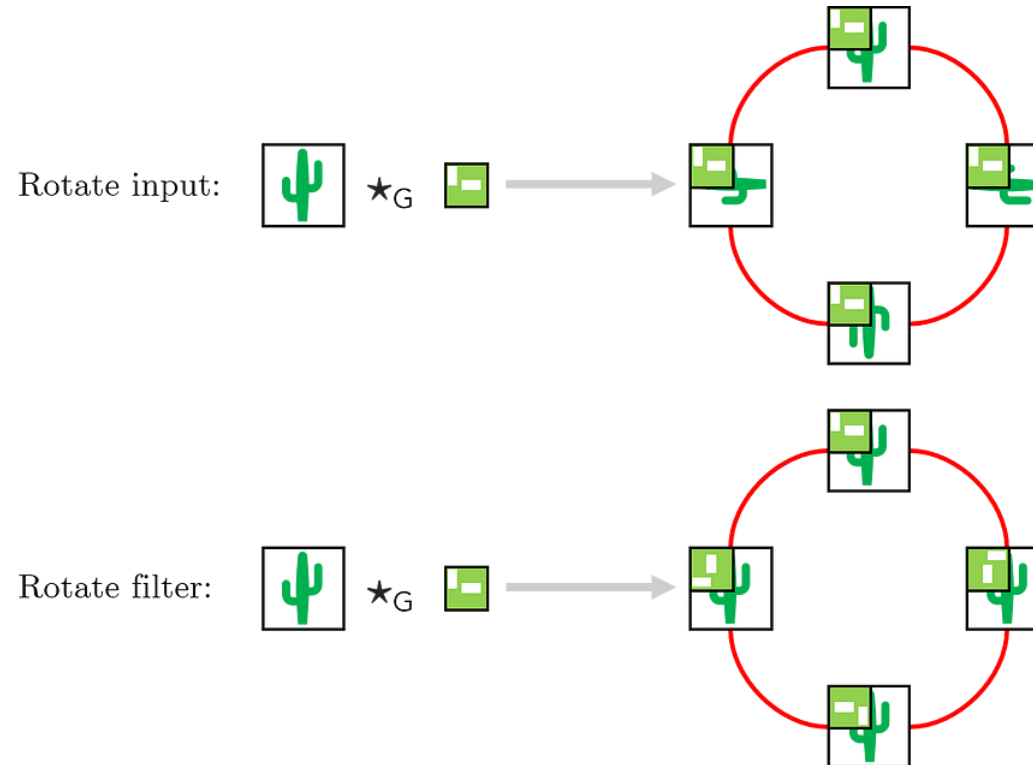


P4ConvZ2



Las G-convoluciones en grupos

el objetivo es **capturar características de la imagen en diferentes orientaciones**



Cada uno de los mapas generados se organiza en un círculo (o "rejilla estructurada")

P4ConvZ2



Segunda convolución

¿Qué pasa si la entrada es un mapa de características estructurado?

Transformación de un objeto estructurado: cómo las características en cada orientación se rotan y organizan.

Producto escalar entre dos objetos estructurados: cómo combinamos los resultados de diferentes rotaciones.

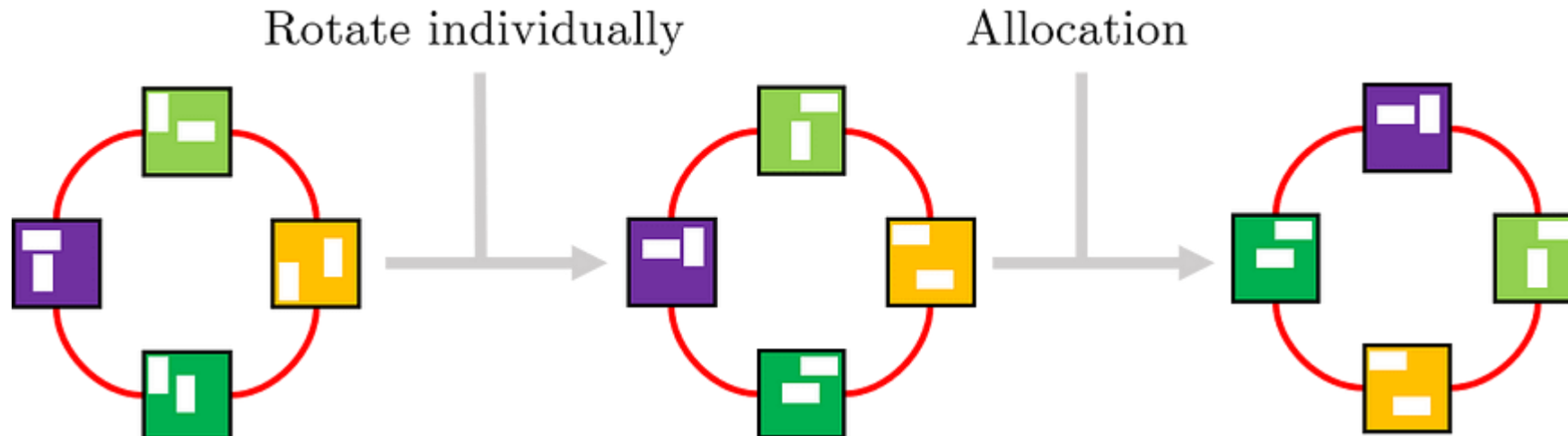


Transformación de un objeto estructurado: cómo las características en cada orientación se rotan y organizan

La transformación ocurre en dos pasos paralelos:

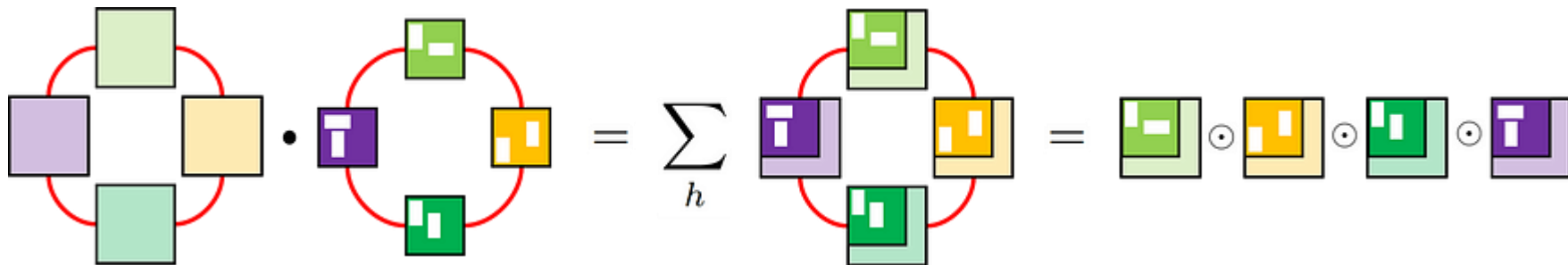
1. Rotación individual de cada característica dentro de su posición.

2. Permutación de posiciones siguiendo una flecha, lo que reorganiza cada nodo en el círculo o gráfico.



Producto escalar entre dos objetos estructurados: cómo combinamos los resultados de diferentes rotaciones.

el producto escalar se realiza superponiendo ambos gráficos y sumando punto por punto los valores de las convoluciones.



Ejemplo

$$\text{Imagen} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Esta matriz representa un objeto central con un borde alrededor

$$\text{Filtro} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Este filtro es un detector de bordes, resaltará los cambios en la intensidad (o bordes) en la imagen.



0° (original):

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$


$$\text{Imagen} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

90°:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{aligned} \text{Resultado} &= (-1 \cdot 0) + (-1 \cdot 1) + (-1 \cdot 0) + (-1 \cdot 1) + (8 \cdot 1) + (-1 \cdot 1) + (-1 \cdot 0) + (-1 \cdot 1) + (-1 \cdot 0) = \\ &= 0 - 1 + 0 - 1 + 8 - 1 + 0 - 1 + 0 = 4 \end{aligned}$$

Entonces, el mapa de características en esta posición central sería: [4]



Entonces, tenemos cuatro mapas de características, uno para cada rotación del filtro:

- **Mapa de Características 0°: 4**
- **Mapa de Características 90°: 4**
- **Mapa de Características 180°: 4**
- **Mapa de Características 270°: 4**

El producto escalar entre estos mapas estructurados se calcula sumando los valores correspondientes en cada mapa. Aquí, el producto escalar será:

$$4+4+4+4=16$$

Ejercicio

Imaginemos que tenemos una **imagen de entrada** y un **filtro** 3x3 que aplicaremos con el grupo de rotaciones p4 (0°, 90°, 180°, y 270°).

1. Imagen de Entrada (3x3):

$$\text{Imagen} = \begin{bmatrix} 2 & 1 & 2 \\ 0 & 3 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

2. Filtro de Detección (3x3):

$$\text{Filtro} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



Rotación 0°:

El filtro aplicado a la imagen en su orientación original (0°):

$$\begin{aligned}\text{Resultado} &= (-1 \cdot 2) + (0 \cdot 1) + (1 \cdot 2) + (-2 \cdot 0) + (0 \cdot 3) + (2 \cdot 0) + (-1 \cdot 1) + (0 \cdot 2) + (1 \cdot 1) \\ &= -2 + 0 + 2 + 0 + 0 + 0 - 1 + 0 + 1 = 0\end{aligned}$$

Rotación 90°:

Rotamos el filtro 90° en sentido horario:

$$\text{Filtro } (90^\circ) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Aplicamos este filtro a la imagen:

$$\begin{aligned}\text{Resultado} &= (1 \cdot 2) + (2 \cdot 1) + (1 \cdot 2) + (0 \cdot 0) + (0 \cdot 3) + (0 \cdot 0) + (-1 \cdot 1) + (-2 \cdot 2) + (-1 \cdot 1) \\ &= 2 + 2 + 2 + 0 + 0 + 0 - 1 - 4 - 1 = 0\end{aligned}$$

Rotación 180°:

Rotamos el filtro otros 90°, para llegar a 180°:

$$\text{Filtro } (180^\circ) = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Aplicamos este filtro a la imagen:

$$\begin{aligned} \text{Resultado} &= (1 \cdot 2) + (0 \cdot 1) + (-1 \cdot 2) + (2 \cdot 0) + (0 \cdot 3) + (-2 \cdot 0) + (1 \cdot 1) + (0 \cdot 2) + (-1 \cdot 1) \\ &= 2 + 0 - 2 + 0 + 0 + 0 + 1 + 0 - 1 = 0 \end{aligned}$$


Rotación 270°:

Rotamos el filtro otros 90°, para llegar a 270°:

$$\text{Filtro } (270^\circ) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Aplicamos este filtro a la imagen:


$$\begin{aligned} \text{Resultado} &= (-1 \cdot 2) + (-2 \cdot 1) + (-1 \cdot 2) + (0 \cdot 0) + (0 \cdot 3) + (0 \cdot 0) + (1 \cdot 1) + (2 \cdot 2) + (1 \cdot 1) \\ &= -2 - 2 - 2 + 0 + 0 + 0 + 1 + 4 + 1 = 0 \end{aligned}$$



Ahora tenemos cuatro mapas de características, uno para cada rotación. En este caso, cada rotación produjo el mismo valor, **0**, en el mapa de características.

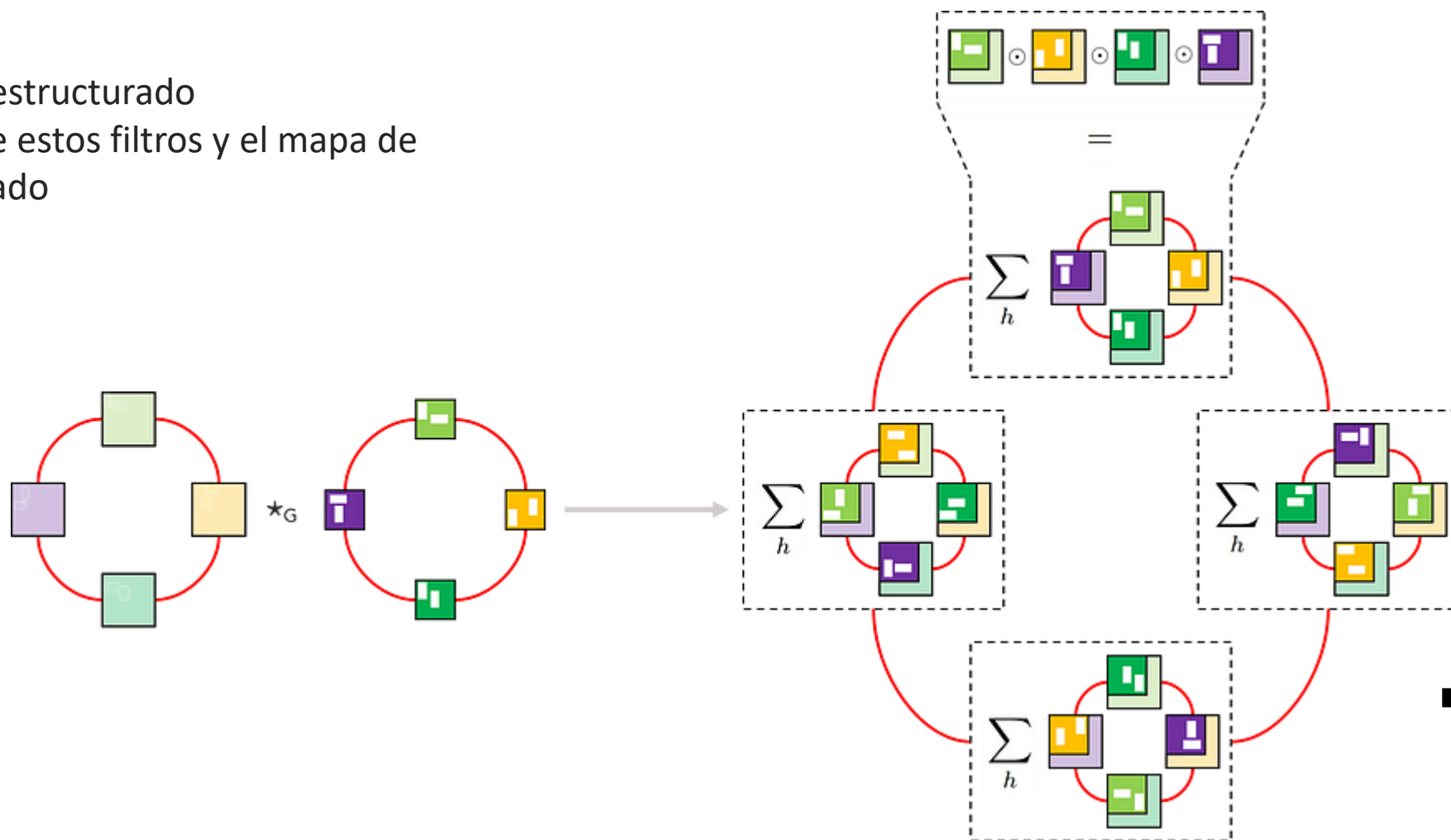
- **Mapa de Características 0°: 000**
- **Mapa de Características 90°: 000**
- **Mapa de Características 180°: 000**
- **Mapa de Características 270°: 000**

El producto escalar entre los mapas estructurados se calcula sumando los valores correspondientes en cada mapa:

$$0+0+0+0=0$$


la G-convolución completa

- un filtro estructurado
- Transformamos el filtro estructurado
- el producto escalar entre estos filtros y el mapa de características estructurado



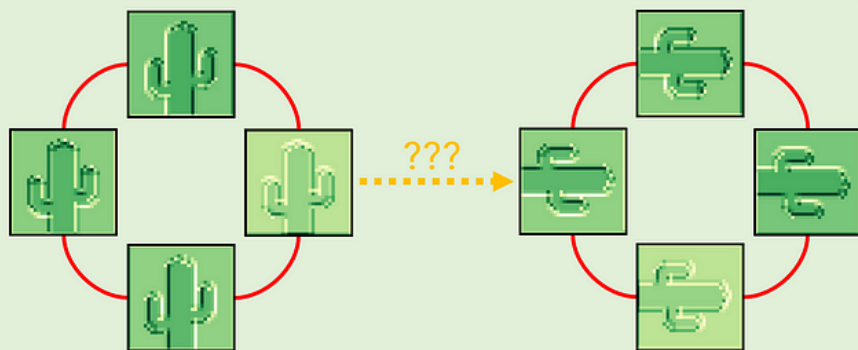
Structured input:
[32, 32]



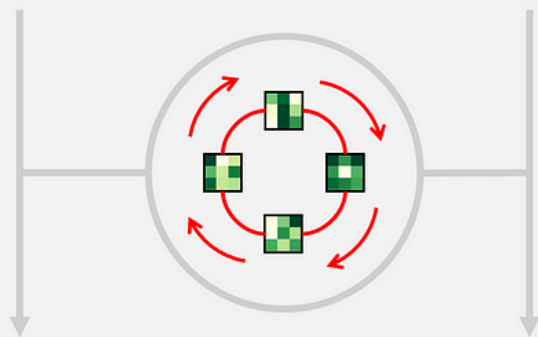
Layer 1: P4ConvZ2
[3, 3] filter, zero pad
+ normalization



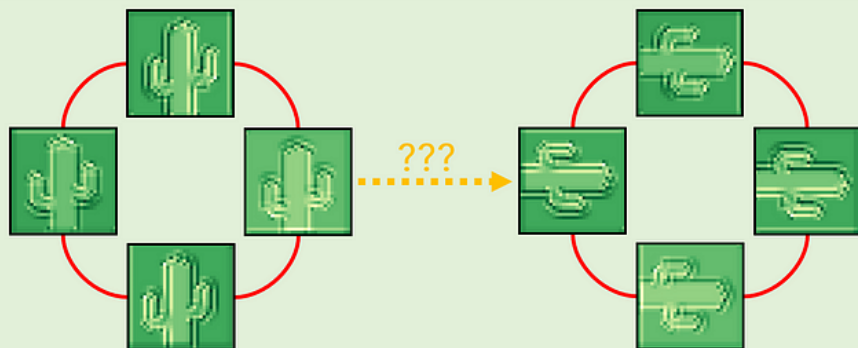
Structured data:
[4, 32, 32]



Layer 2: P4ConvP4
[4, 3, 3] filter, zero pad
+ normalization



Structured data:
[4, 32, 32]



- imagen de 32x32 píxeles de un cactus
- Capa 1: P4ConvZ2 (Filtro [3, 3])

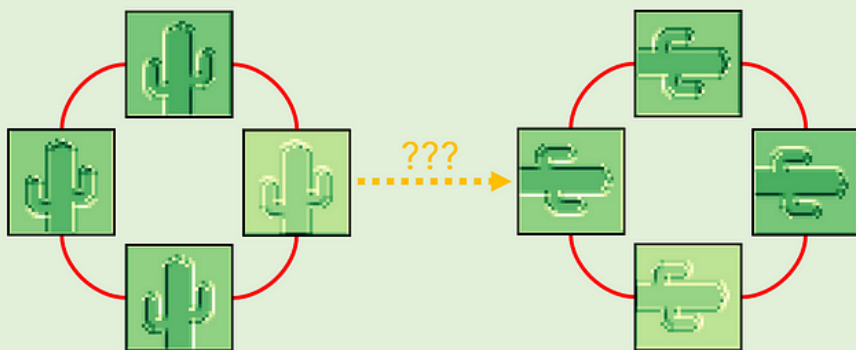
Structured input:
[32, 32]



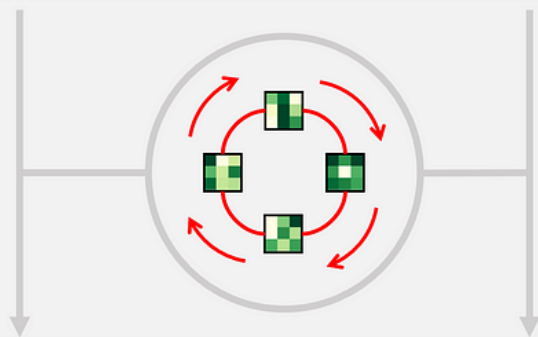
Layer 1: P4ConvZ2
[3, 3] filter, zero pad
+ normalization



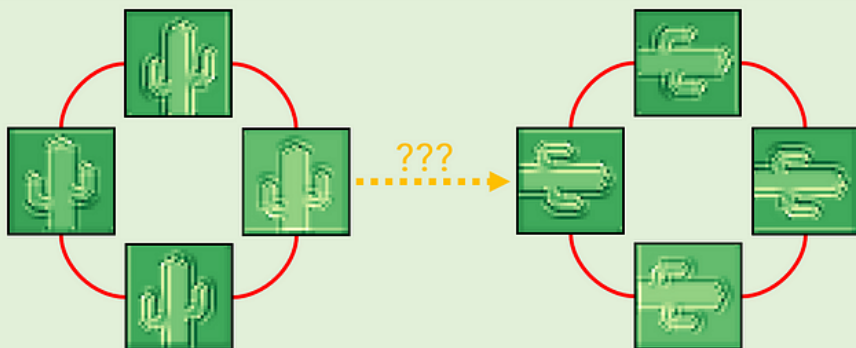
Structured data:
[4, 32, 32]



Layer 2: P4ConvP4
[4, 3, 3] filter, zero pad
+ normalization



Structured data:
[4, 32, 32]



- imagen de 32x32 píxeles de un cactus
- Capa 1: P4ConvZ2 (Filtro [3, 3])

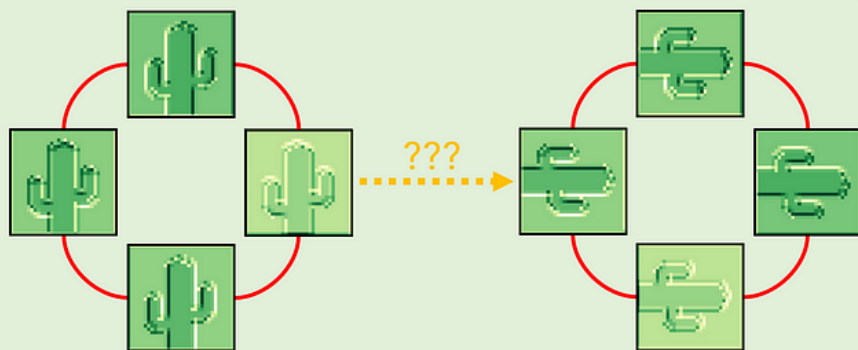
Structured input:
[32, 32]



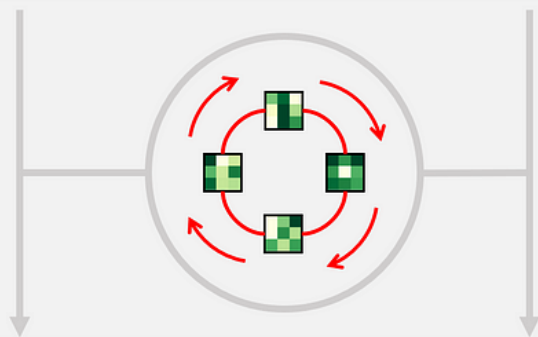
Layer 1: P4ConvZ2
[3, 3] filter, zero pad
+ normalization



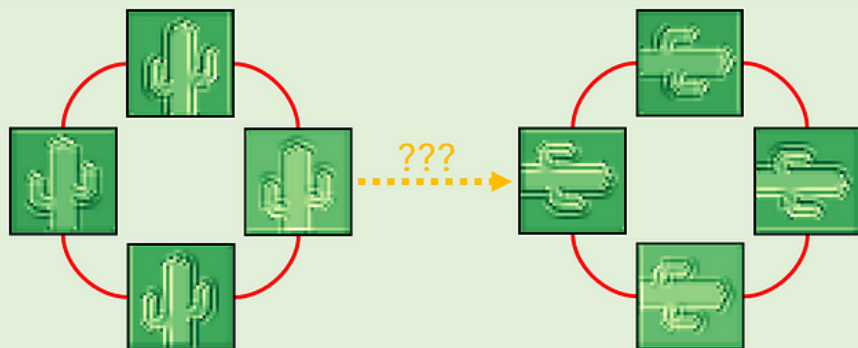
Structured data:
[4, 32, 32]



Layer 2: P4ConvP4
[4, 3, 3] filter, zero pad
+ normalization



Structured data:
[4, 32, 32]

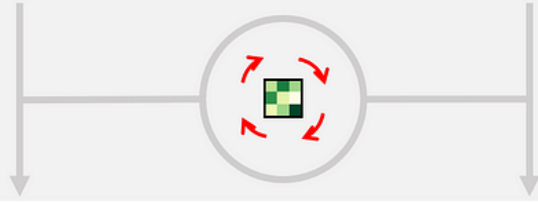


- imagen de 32x32 píxeles de un cactus
- Capa 1: P4ConvZ2 (Filtro [3, 3])
- "mapa estructurado de datos"

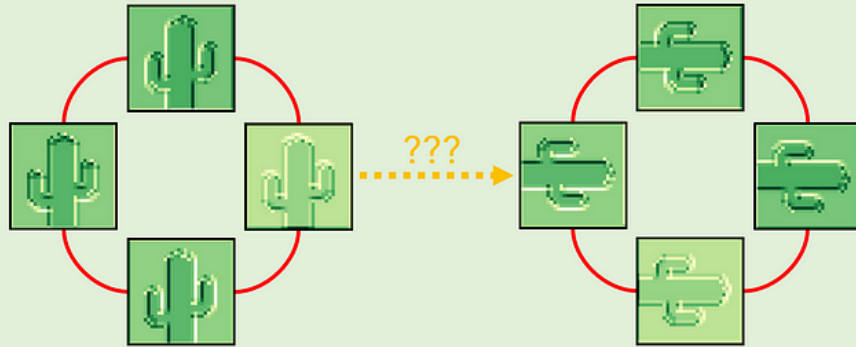
Structured input:
[32, 32]



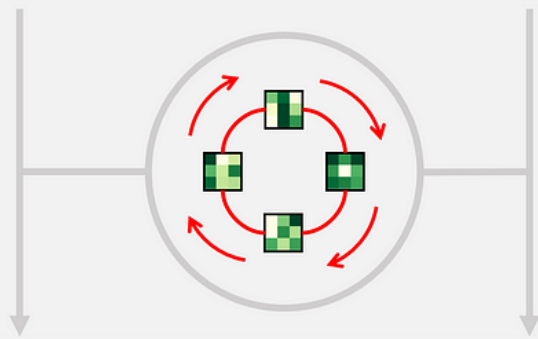
Layer 1: P4ConvZ2
[3, 3] filter, zero pad
+ normalization



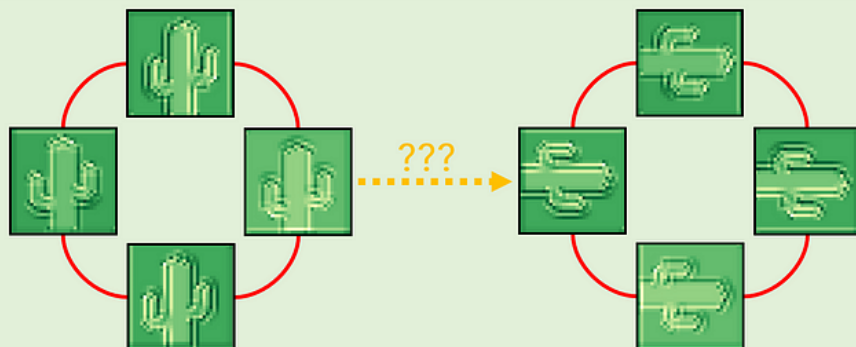
Structured data:
[4, 32, 32]



Layer 2: P4ConvP4
[4, 3, 3] filter, zero pad
+ normalization



Structured data:
[4, 32, 32]

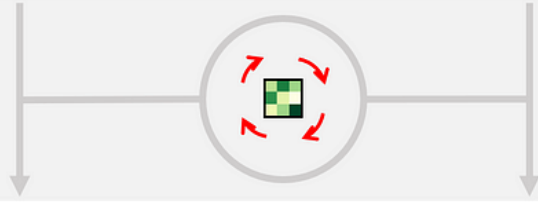


- imagen de 32x32 píxeles de un cactus
- Capa 1: P4ConvZ2 (Filtro [3, 3])
- "mapa estructurado de datos"
- P4ConvP4 (Filtro [4, 3, 3])
- Datos Estructurados Finales ([4, 32, 32])

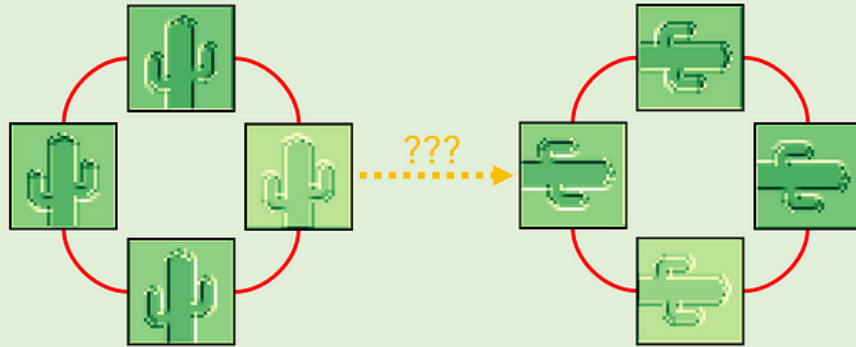
Structured input:
[32, 32]



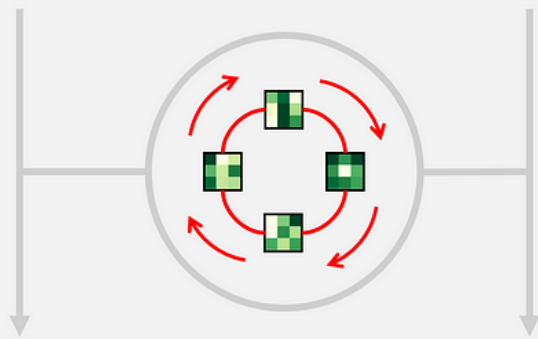
Layer 1: P4ConvZ2
[3, 3] filter, zero pad
+ normalization



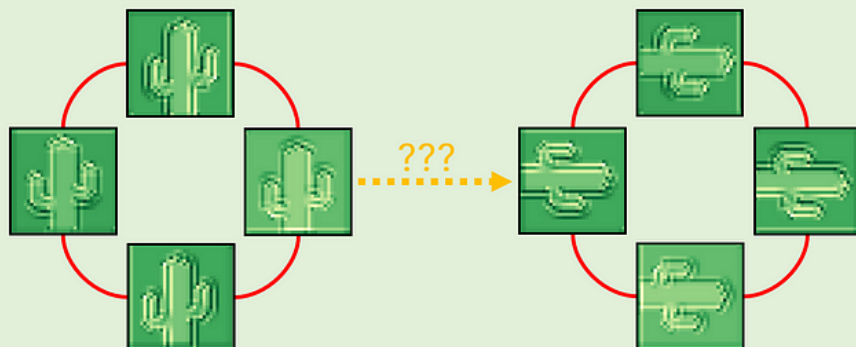
Structured data:
[4, 32, 32]



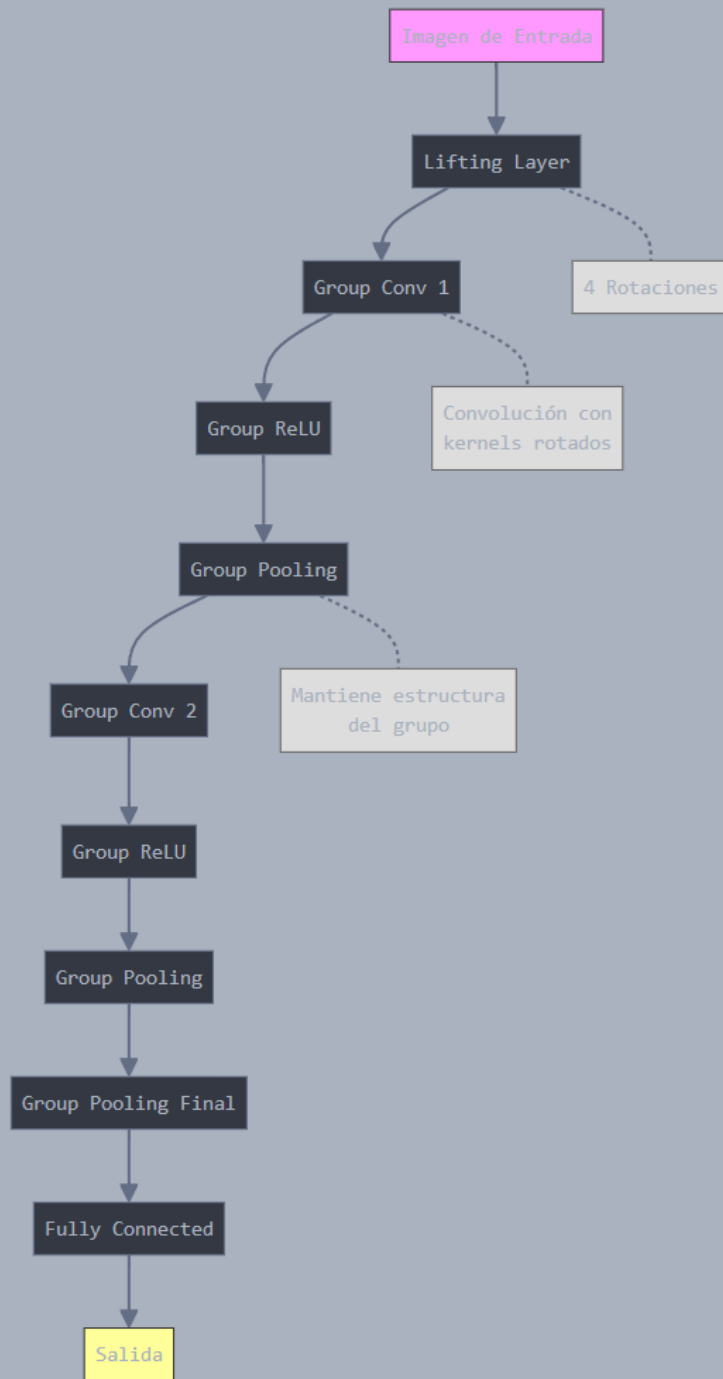
Layer 2: P4ConvP4
[4, 3, 3] filter, zero pad
+ normalization



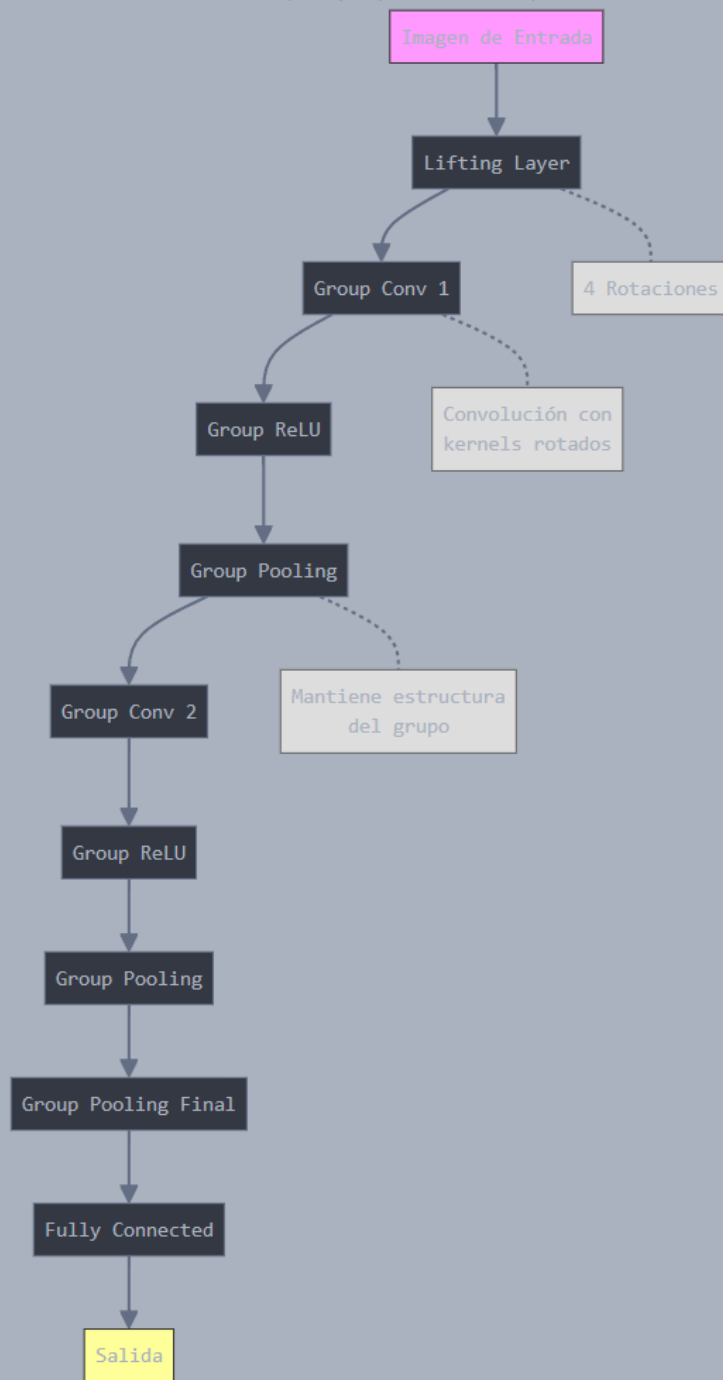
Structured data:
[4, 32, 32]



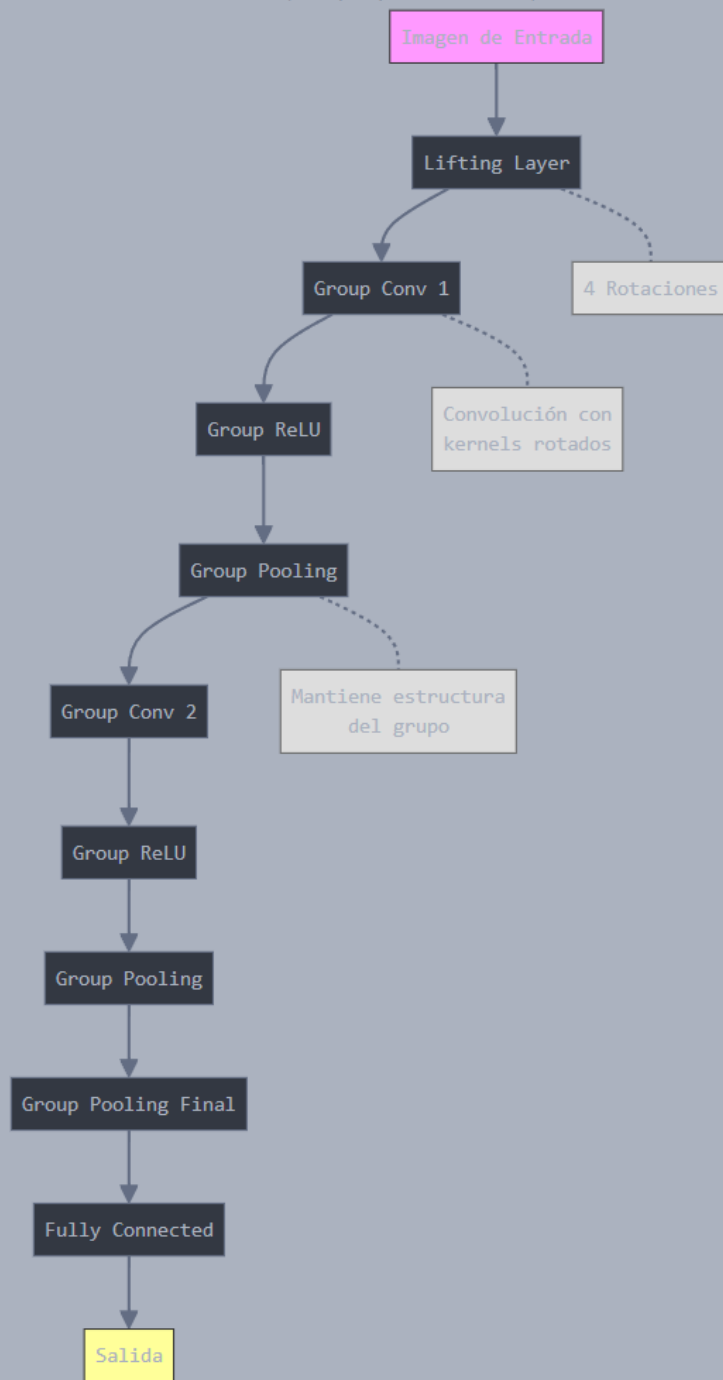
- imagen de 32x32 píxeles de un cactus
- Capa 1: P4ConvZ2 (Filtro [3, 3])
- "mapa estructurado de datos"
- P4ConvP4 (Filtro [4, 3, 3])
- Datos Estructurados Finales ([4, 32, 32])



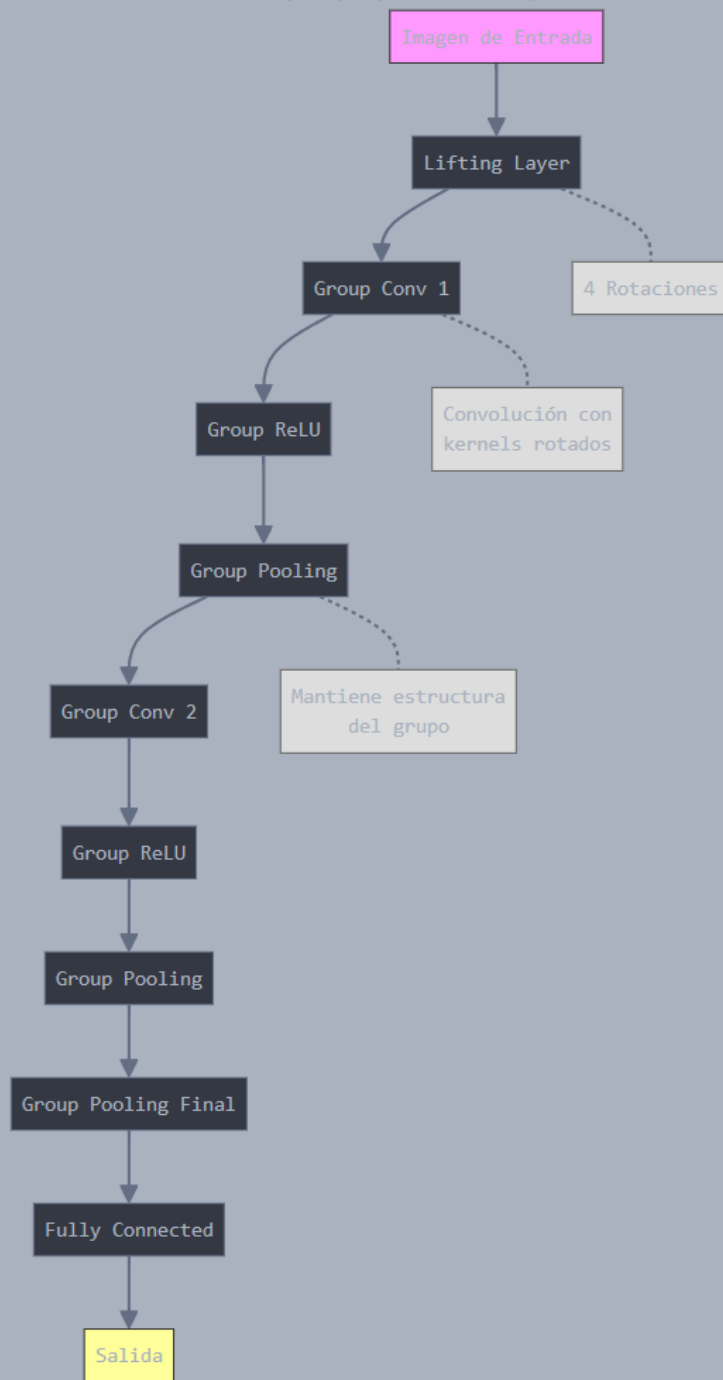
- La Lifting Layer proyecta la imagen en un espacio de grupo donde se aplican las simetrías deseadas.



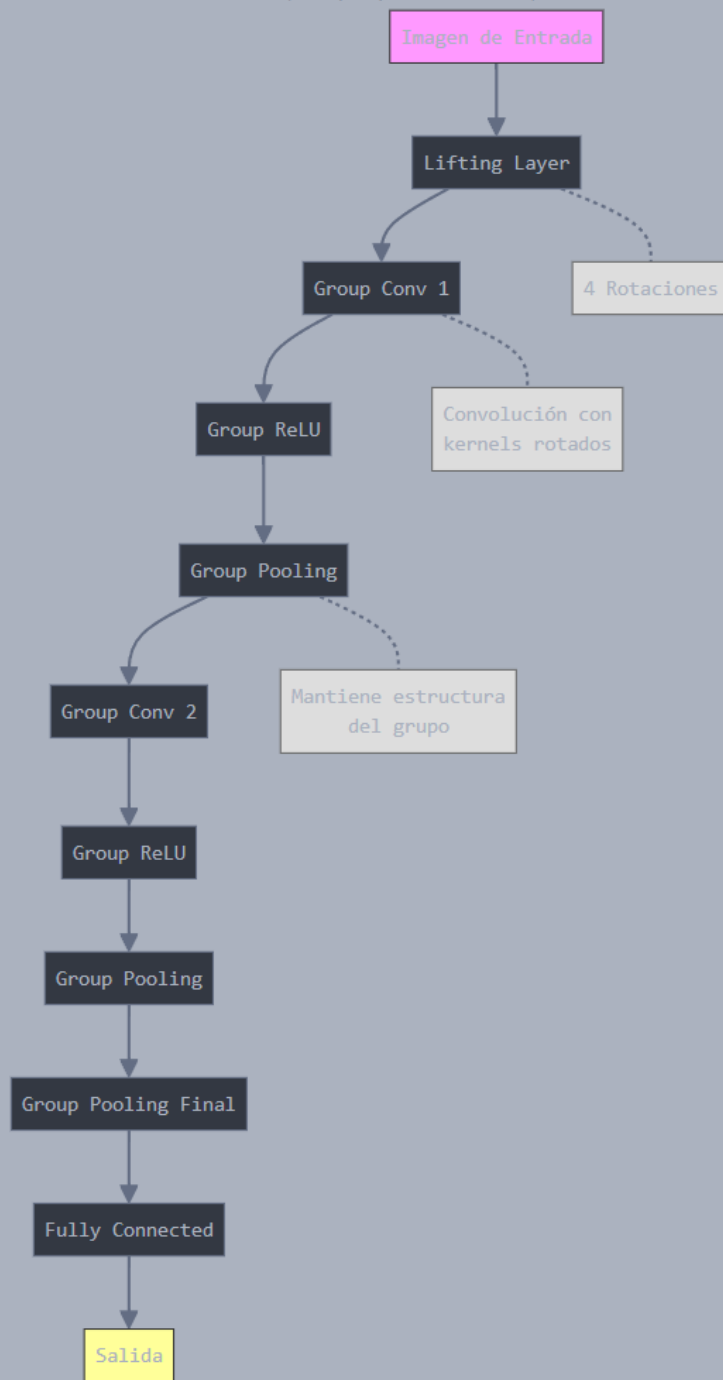
- La Lifting Layer proyecta la imagen en un espacio de grupo donde se aplican las simetrías deseadas
- Es una capa convolucional especial diseñada para trabajar en el espacio de grupo



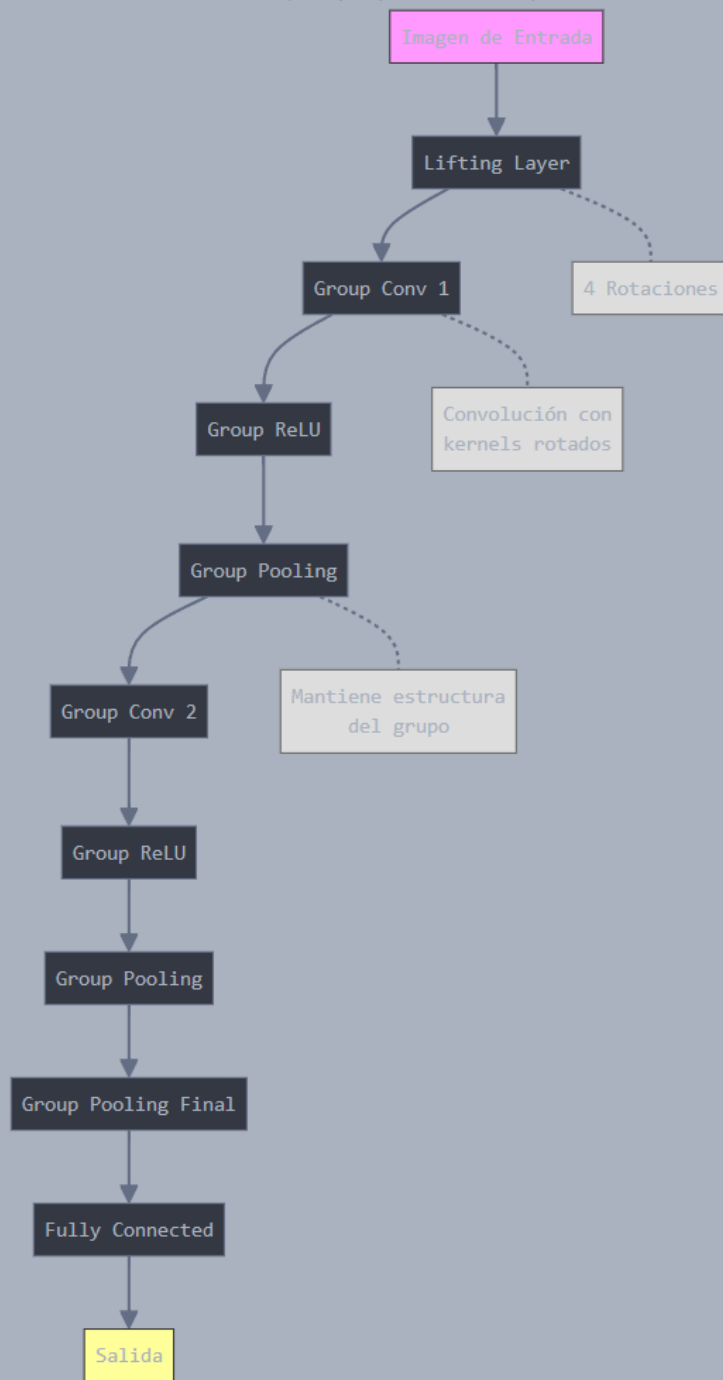
- La Lifting Layer proyecta la imagen en un espacio de grupo donde se aplican las simetrías deseadas
- Es una capa convolucional especial diseñada para trabajar en el espacio de grupo
- Introduce no linealidad en la red al activar solo los valores positivos, lo cual ayuda a modelar relaciones complejas



- La Lifting Layer proyecta la imagen en un espacio de grupo donde se aplican las simetrías deseadas
- Es una capa convolucional especial diseñada para trabajar en el espacio de grupo
- Introduce no linealidad en la red al activar solo los valores positivos, lo cual ayuda a modelar relaciones complejas
- Reduce la resolución espacial, similar al MaxPooling en una CNN tradicional, pero lo hace manteniendo la estructura de grupo..



- La Lifting Layer proyecta la imagen en un espacio de grupo donde se aplican las simetrías deseadas
- Es una capa convolucional especial diseñada para trabajar en el espacio de grupo
- Introduce no linealidad en la red al activar solo los valores positivos, lo cual ayuda a modelar relaciones complejas
- Reduce la resolución espacial, similar al MaxPooling en una CNN tradicional, pero lo hace manteniendo la estructura de grupo..
- Una última capa de pooling en el espacio de grupo que reduce la representación final para pasarla a la capa totalmente conectada.



- La Lifting Layer proyecta la imagen en un espacio de grupo donde se aplican las simetrías deseadas
- Es una capa convolucional especial diseñada para trabajar en el espacio de grupo
- Introduce no linealidad en la red al activar solo los valores positivos, lo cual ayuda a modelar relaciones complejas
- Reduce la resolución espacial, similar al MaxPooling en una CNN tradicional, pero lo hace manteniendo la estructura de grupo..
- Una última capa de pooling en el espacio de grupo que reduce la representación final para pasarla a la capa totalmente conectada.
- Igual que en la CNN tradicional



Usos

1. Reconocimiento de Imágenes y Clasificación

Las G-CNNs se han utilizado ampliamente en el reconocimiento de imágenes, particularmente cuando las imágenes pueden estar rotadas o reflejadas.

2. Análisis de Imágenes Biomédicas

En aplicaciones biomédicas, como la clasificación de tejidos y el análisis de células, las G-CNNs pueden mejorar la precisión al reconocer patrones en diferentes orientaciones.

3. Visión por Computadora en Robótica

Las G-CNNs permiten a los robots identificar objetos en diversas orientaciones, facilitando su reconocimiento en entornos tridimensionales.

4. Reconocimiento de Texto y Símbolos en Documentos Escaneados

En el procesamiento de documentos y OCR (Optical Character Recognition), las G-CNNs pueden reconocer caracteres rotados y símbolos en diferentes orientaciones.

5. Astronomía y Análisis de Imágenes Satelitales

Las G-CNNs son útiles en el análisis de imágenes astronómicas y satelitales, donde los objetos pueden aparecer en cualquier orientación.





Pros:

- Entienden relaciones complejas:** Las GNNs son ideales para datos donde las conexiones entre elementos son importantes, como redes sociales o estructuras moleculares.
- Adaptabilidad:** Funcionan bien en diferentes tipos de grafos y pueden aprender patrones en datos de distintos dominios, como biología, química y recomendaciones.
- Buen manejo de estructuras desordenadas:** A diferencia de las redes tradicionales, las GNNs no necesitan que los datos tengan un orden específico. Esto significa que son robustas ante cambios en el orden de los nodos.
- Interpretables:** En algunos casos, es posible ver cuáles nodos o conexiones influyen más en las predicciones, lo que ayuda a entender mejor el modelo.

Contras:




- **Consumen muchos recursos:** Procesar grafos grandes es lento y requiere mucha memoria, lo que hace difícil su uso en grafos muy extensos.
 - **Difíciles de entrenar si son profundas:** Si la red tiene muchas capas, los nodos empiezan a parecerse demasiado unos a otros, lo que hace que el modelo pierda precisión (esto se llama "sobreamortiguamiento").
 - **Sensibles a grafos incompletos:** Si el grafo está mal construido o tiene información faltante, las predicciones pueden ser menos precisas.
 - **Complicadas de interpretar en grafos grandes:** En grafos muy conectados o complejos, es más difícil entender qué está "aprendiendo" el modelo.
 - **Limitación en profundidad:** Si se añaden demasiadas capas, el modelo puede empezar a tener problemas para aprender de manera efectiva.
- 



Referencia: Cohen, T. S., & Welling, M. (2016). *Group equivariant convolutional networks*. In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (pp. 2990–2999).

Referencia: Bekkers, E. J., Lafarge, M. W., Veta, M., Eppenhof, K. A. J., Pluim, J. P. W., & Duits, R. (2018). *Rotation equivariant convolutional networks for medical image analysis*. Medical Image Analysis, 51, 16–29.

Referencia: Marcos, D., Volpi, M., Komodakis, N., & Tuia, D. (2017). *Rotation equivariant vector field networks*. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 5048–5057).



Referencia: Zhang, S., Wang, Z., & Xiao, J. (2019). *G-CNN for document image recognition*. Journal of Document Analysis and Recognition, 22(3), 189–202.

Referencia: Dieleman, S., Willett, K. W., & Dambre, J. (2015). *Rotation-invariant convolutional neural networks for galaxy morphology prediction*. Monthly Notices of the Royal Astronomical Society, 450(2), 1441–1459.



Relación con matrices de rotación

Las rotaciones están basadas en las transformaciones lineales definidas por **matrices de rotación** en 2D. Para un ángulo θ , la matriz de rotación es:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Ejemplo:

- Para $\theta = 90^\circ$, la matriz de rotación sería:

$$R(90^\circ) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Si aplicas esta matriz a un filtro, el resultado es el filtro rotado 90° .

Ejemplo

Filtro original

Imagina que tienes un filtro 3×3 , que representa los valores en sus posiciones espaciales originales:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Aquí, cada valor corresponde a una posición relativa al centro del filtro. Supongamos que el centro del filtro es el origen (0,0). Las coordenadas de los puntos son:

| Valor | Coordenada (x, y) |
|-------|---------------------|
| 1 | $(-1, 1)$ |
| 2 | $(0, 1)$ |
| 3 | $(1, 1)$ |
| 4 | $(-1, 0)$ |
| 5 | $(0, 0)$ |
| 6 | $(1, 0)$ |
| 7 | $(-1, -1)$ |
| 8 | $(0, -1)$ |
| 9 | $(1, -1)$ |



La matriz de rotación para un ángulo $\theta = 45^\circ = \frac{\pi}{4}$ es:

$$R(45^\circ) = \begin{bmatrix} \cos(45^\circ) & -\sin(45^\circ) \\ \sin(45^\circ) & \cos(45^\circ) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

El punto inicial del filtro es $(1, 0)$. Lo representamos como un vector columna:

$$\text{Punto original: } \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Paso 2: Multiplicamos por la matriz de rotación

La nueva posición (x', y') se calcula multiplicando la matriz de rotación $R(45^\circ)$ por el vector posición original:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R(45^\circ) \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Matriz de rotación

La matriz de rotación para un ángulo $\theta = 45^\circ$ es:

$$R(45^\circ) = \begin{bmatrix} \cos(45^\circ) & -\sin(45^\circ) \\ \sin(45^\circ) & \cos(45^\circ) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

Cada coordenada (x, y) del filtro será transformada usando:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R(45^\circ) \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Punto $(-1, 1)$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Cálculo del nuevo x' :

$$x' = \left(\frac{\sqrt{2}}{2} \cdot -1 \right) + \left(-\frac{\sqrt{2}}{2} \cdot 1 \right) = -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2} = -\sqrt{2}$$

Cálculo del nuevo y' :

$$y' = \left(\frac{\sqrt{2}}{2} \cdot -1 \right) + \left(\frac{\sqrt{2}}{2} \cdot 1 \right) = -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} = 0$$

Nueva posición:

$$(-1, 1) \rightarrow (-\sqrt{2}, 0)$$

Punto (1, 1)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Cálculo del nuevo x' :

$$x' = \left(\frac{\sqrt{2}}{2} \cdot 1 \right) + \left(-\frac{\sqrt{2}}{2} \cdot 1 \right) = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2} = 0$$

Cálculo del nuevo y' :

$$y' = \left(\frac{\sqrt{2}}{2} \cdot 1 \right) + \left(\frac{\sqrt{2}}{2} \cdot 1 \right) = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} = \sqrt{2}$$

Nueva posición:

$$(1, 1) \rightarrow (0, \sqrt{2})$$

Punto $(0, -1)$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

Cálculo del nuevo x' :

$$x' = \left(\frac{\sqrt{2}}{2} \cdot 0 \right) + \left(-\frac{\sqrt{2}}{2} \cdot -1 \right) = 0 + \frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2}$$

Cálculo del nuevo y' :

$$y' = \left(\frac{\sqrt{2}}{2} \cdot 0 \right) + \left(\frac{\sqrt{2}}{2} \cdot -1 \right) = 0 - \frac{\sqrt{2}}{2} = -\frac{\sqrt{2}}{2}$$

Nueva posición:

$$(0, -1) \rightarrow \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right)$$

- Filtro original:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- Filtro rotado 45° (tras interpolación):

$$\begin{bmatrix} 0 & 1.5 & 3 \\ 3.5 & 5 & 6.5 \\ 7 & 8.5 & 0 \end{bmatrix}$$

