

Redes Convolucionales

Capas Convolucionales



Principios Básicos

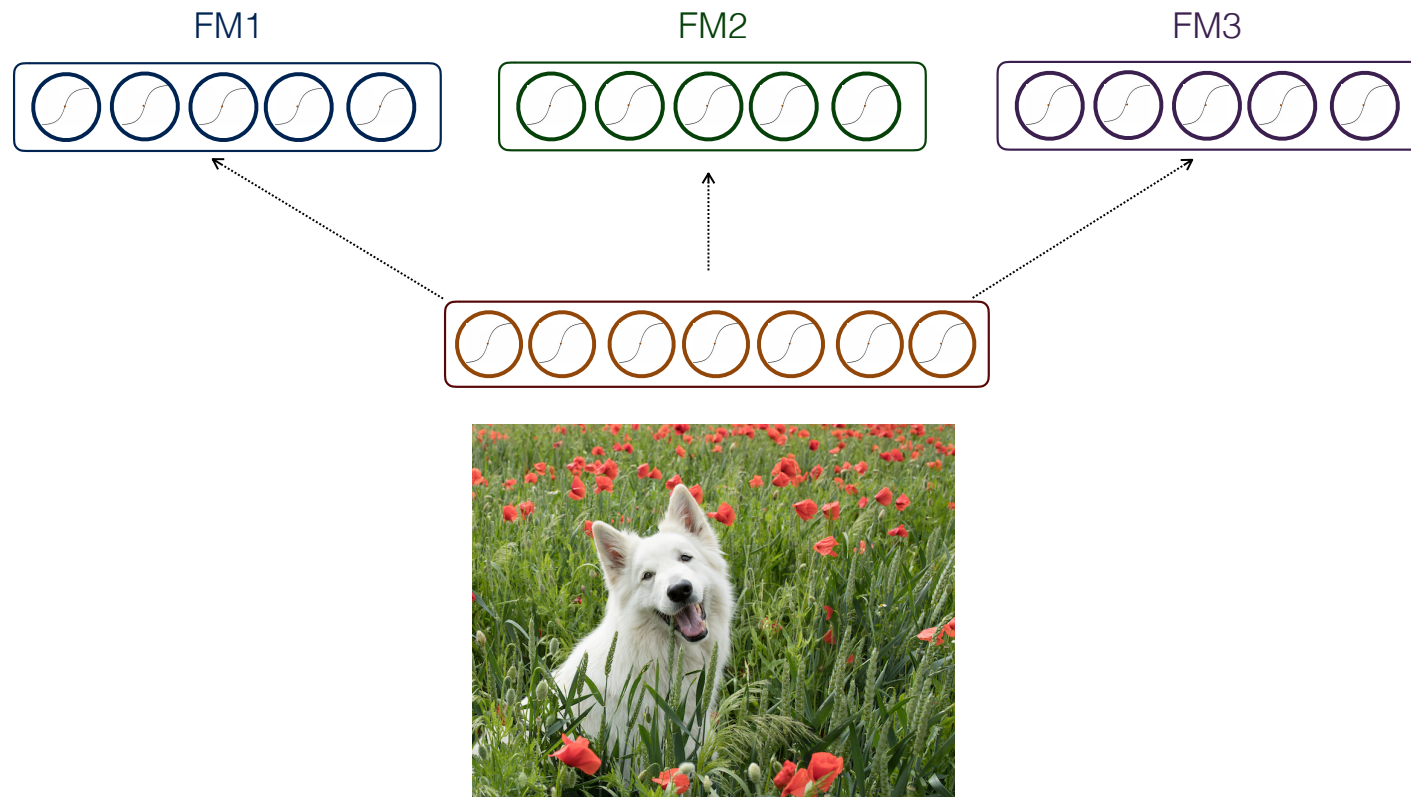
- Las capas de una red convolucional serán capaces de **patrones locales en grandes arreglos multi-dimensionales de datos**. Esto se logrará mediante 3 principios básicos de diseño:



- **Conectividad local.**
 - Compartición de pesos.
 - Pooling.
- A esto sumaremos un principio transversal de diseño: mantener la topología de los datos a medida que se propaga por la red.

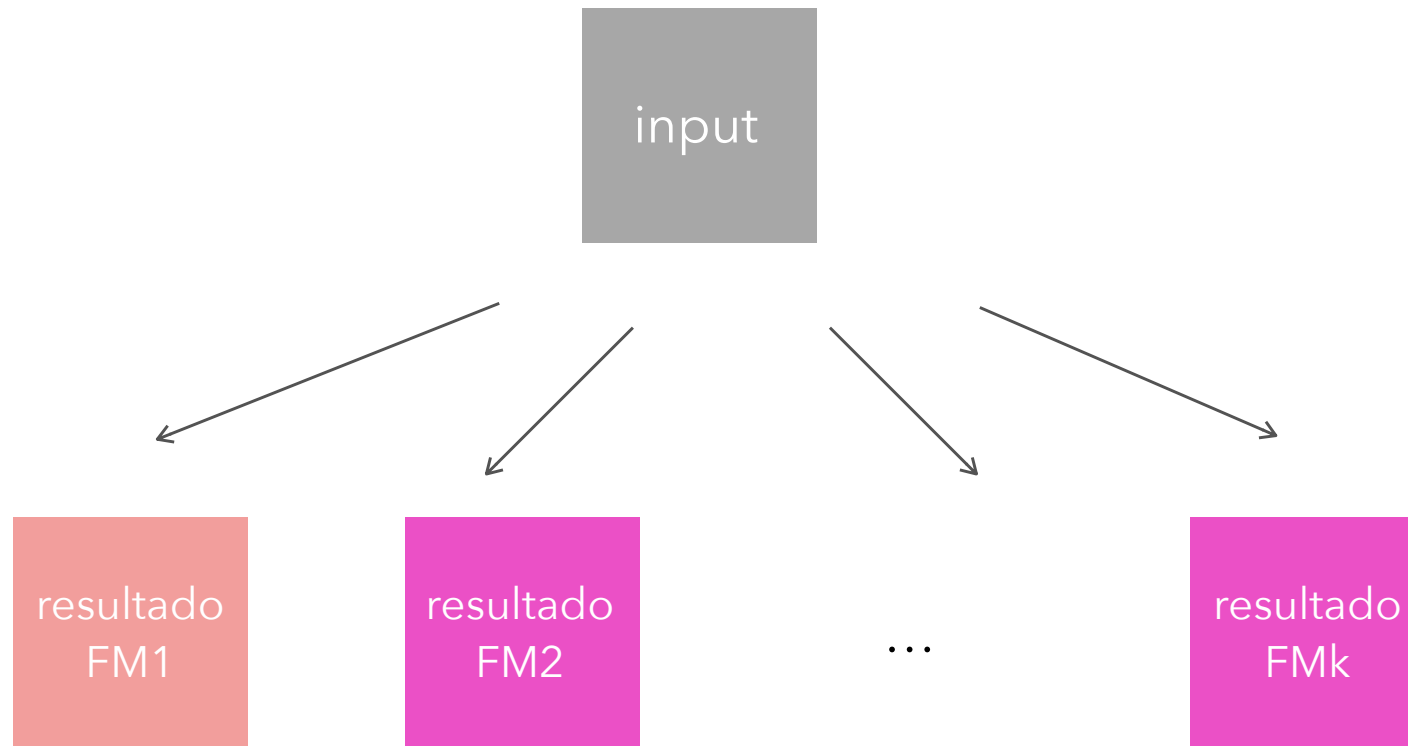
Capas Convolucionales

- Para poder detectar diferentes patrones, tendremos en general varios filtros o canales en una capa convolucional. En vez de hablar de neuronas hablaremos ahora del número de filtros, número de canales, o número de FMs de la capa.



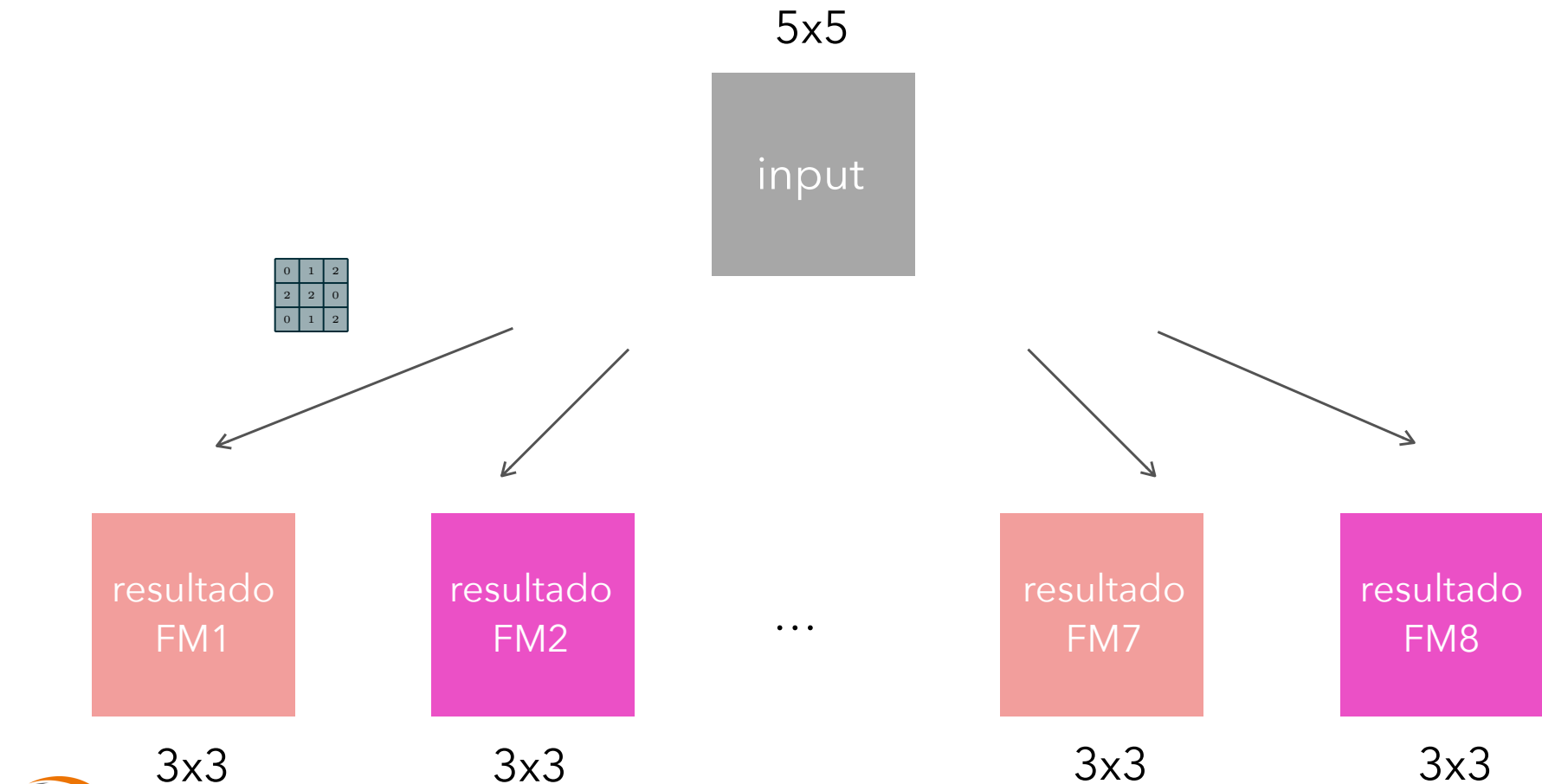
Capas Convolucionales

- Típicamente, los distintos canales estarán parametrizados uniformemente (e.g. mismo stride y tamaño de filtro) de manera que al aplicarlos sobre un determinado patrón de entrada **obtendremos resultados** (mapas de activación) **de la misma forma y dimensiones.**



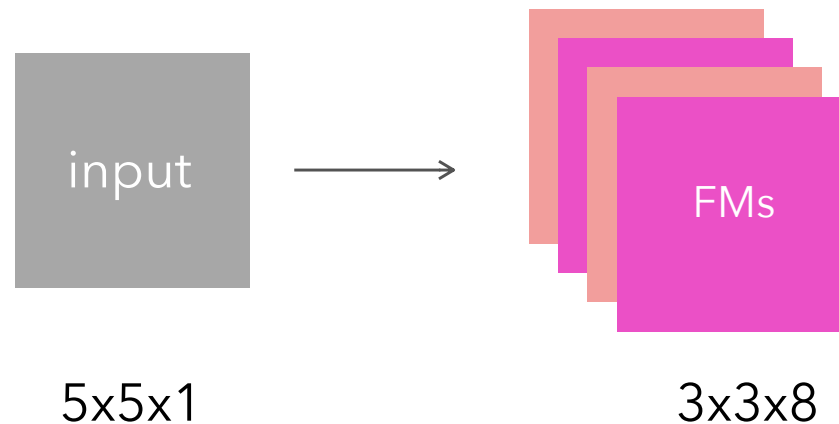
Capas Convolucionales

- Por ejemplo, en el caso anterior, teníamos un filtro de 3x3 que producía un resultado de 3x3 al aplicarlo sobre un volumen de 5x5. Si tenemos 8 filtros en vez de 1, obtendremos 8 resultados de 3x3.



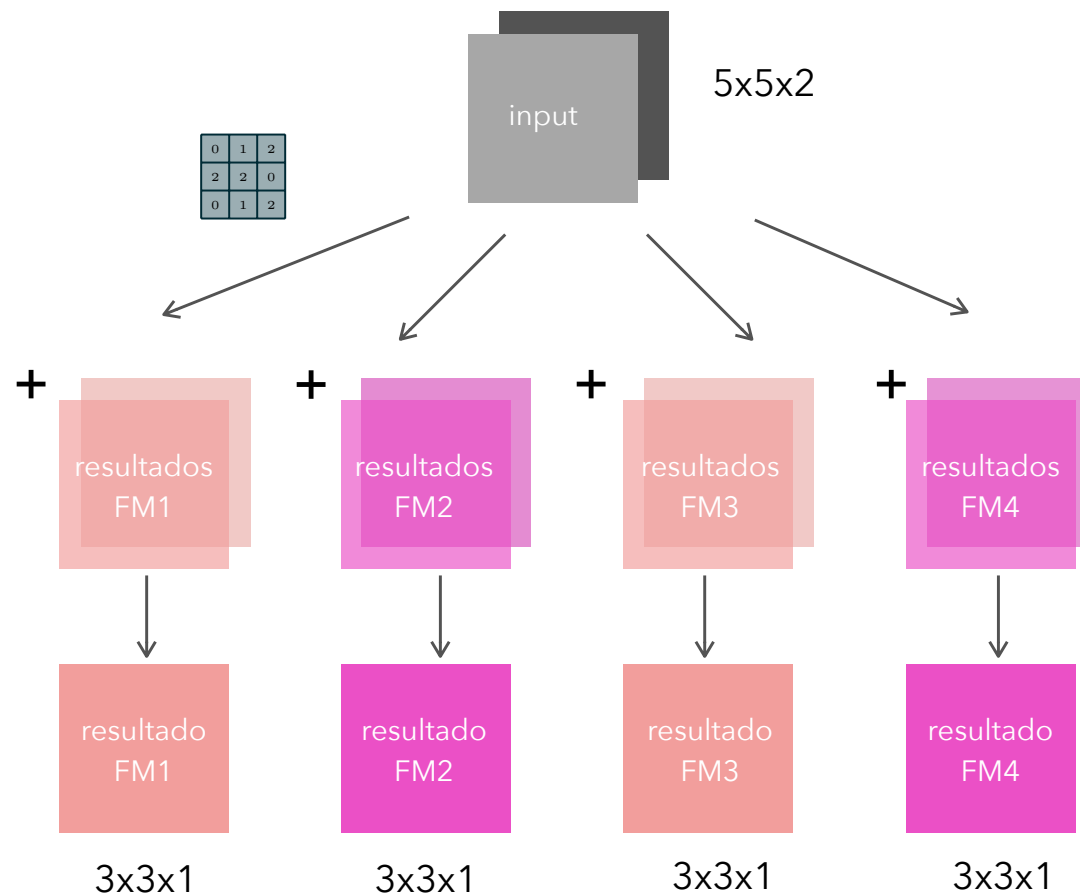
Capas Convolucionales

- Esos 8 resultados se pueden **apilar en una dimensión extra** formando un arreglo/volumen de $3 \times 3 \times 8$. Tenemos entonces que la capa transforma un volumen de $5 \times 5 \times 1$ en uno de $3 \times 3 \times 8$.



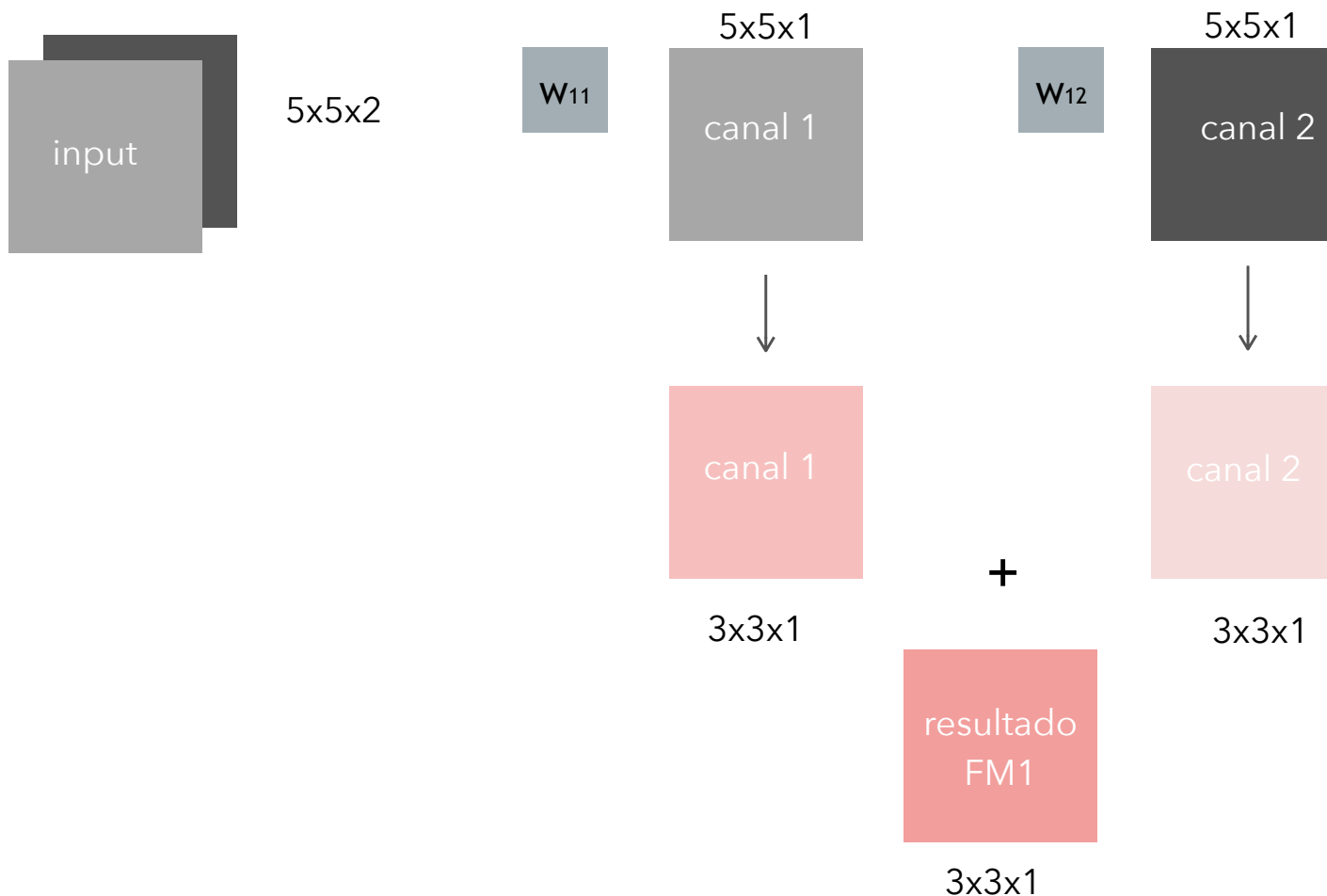
Capas Convolucionales

- Cuando una capa convolucional 2D recibe un patrón de entrada 3D (o de más dimensiones), convendremos que la acción de cada FM se aplica sobre **cada dimensión extra por separado y que los resultados se suman.**



Capas Convolucionales

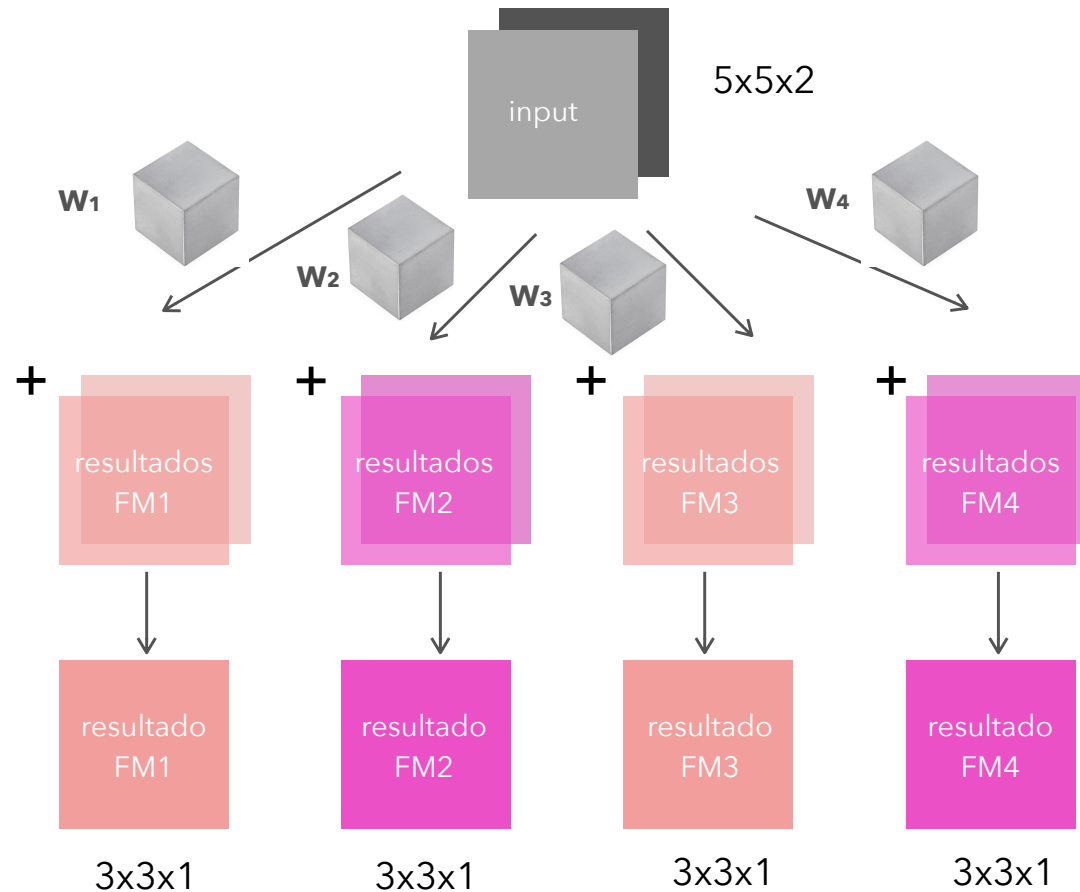
- En general, la acción sobre cada dimensión adicional (denominada canal), se hace **con una matriz de pesos diferente**.



(*) En ciertas librerías se usa la misma matriz para cada canal.

Capas Convolucionales

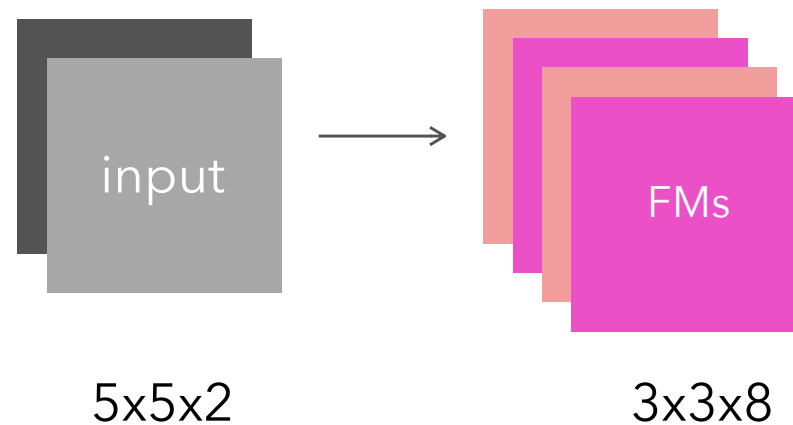
- Con esta convención, las neuronas de un FM bi-dimensional con inputs 3D de forma $I \times J \times K$, tienen en realidad un cubo de pesos de forma $S_i \times S_j \times K$ en vez de una matriz de pesos.



(*) En ciertas librerías se forma el cubo replicando la matriz K veces.

Capas Convolucionales

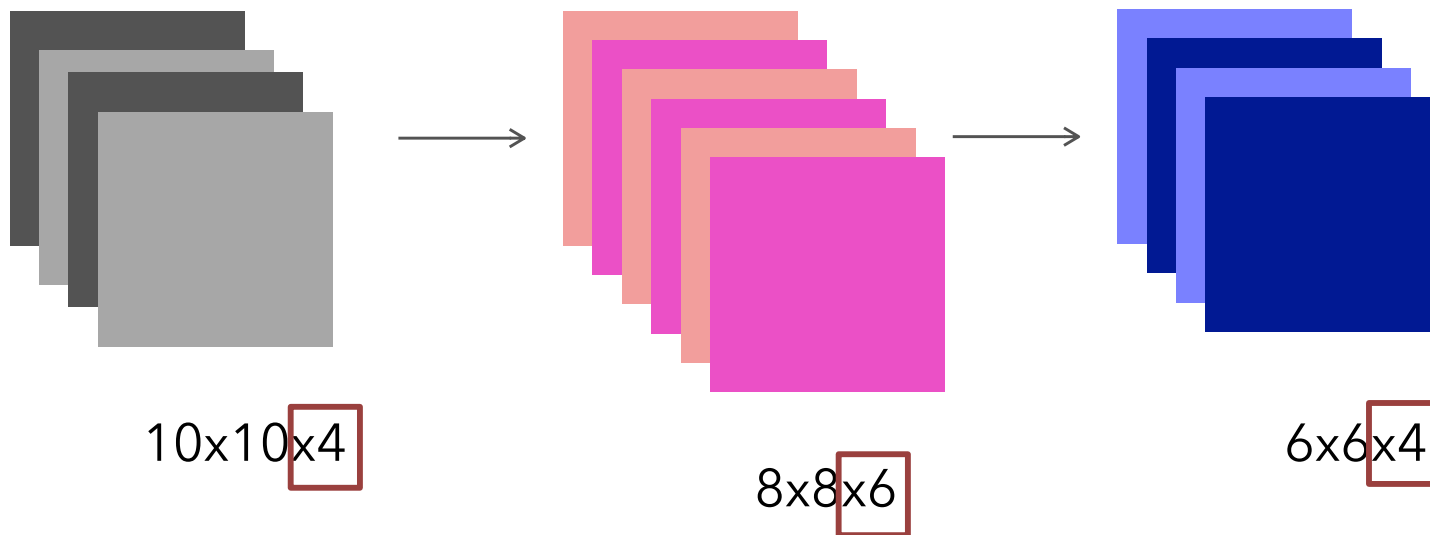
- De este modo, los resultados se siguen pudiendo **apilar en una sola dimensión extra** independiente de cuántas dimensiones adicionales de entrada se presenten.



- Permitir el ingreso de volúmenes K+1-D a una capa K-D nos permitirá **apilar en profundidad varias capas convolucionales**, independientemente de número de FM en cada capa.

Capas Convolucionales

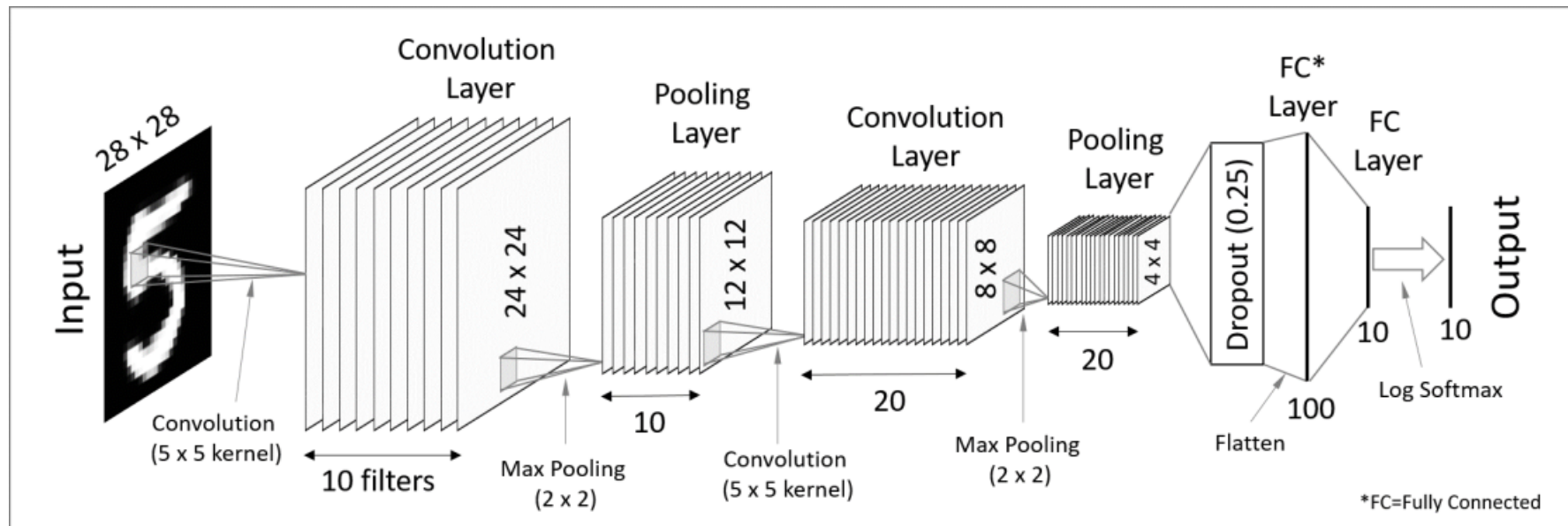
- Esta convención nos permitirá **apilar en profundidad varias capas** convolucionales, independientemente de número de FM en cada capa.



- Deja de ser necesario que el tamaño de la última dimensión se mantenga constante entre las capas.**

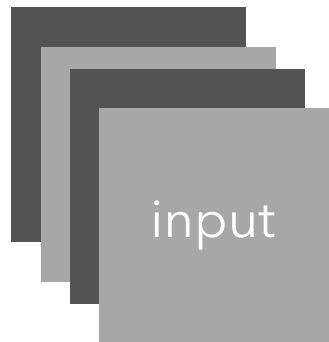
Ejercicio

- Mostrar los cambios de forma que experimenta el patrón de entrada mientras atraviesa las capas de la red, es la forma más común de describir la arquitectura de una red convolucional.



Ejercicio

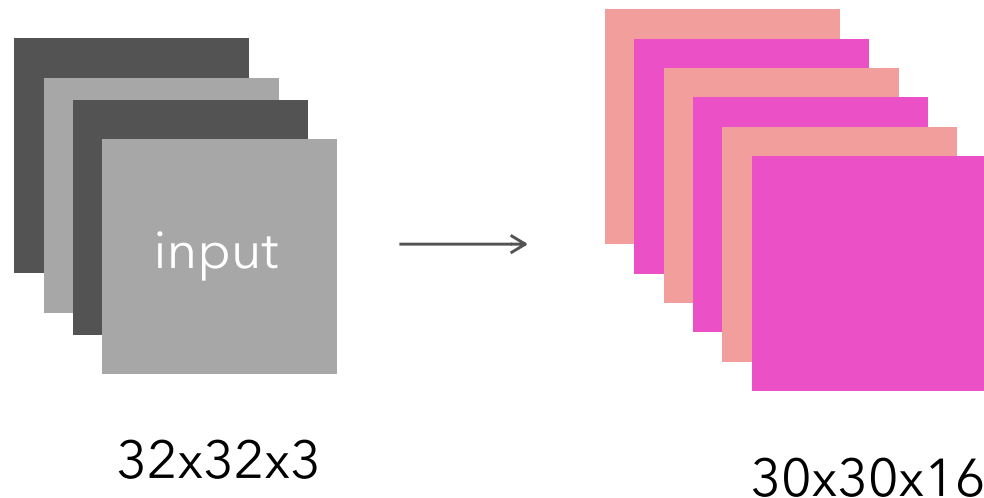
- Consideremos por ejemplo una capa 2D con 16 filtros (FMs) de 3×3 (campos receptivos) y strides de 1×1 como en el último ejemplo. Si queremos procesar volúmenes de $32 \times 32 \times 3$ ¿Qué forma tendrá el arreglo de neuronas, la matriz de pesos y el volumen de datos una vez que pase por la capa?



$32 \times 32 \times 3$

Ejercicio

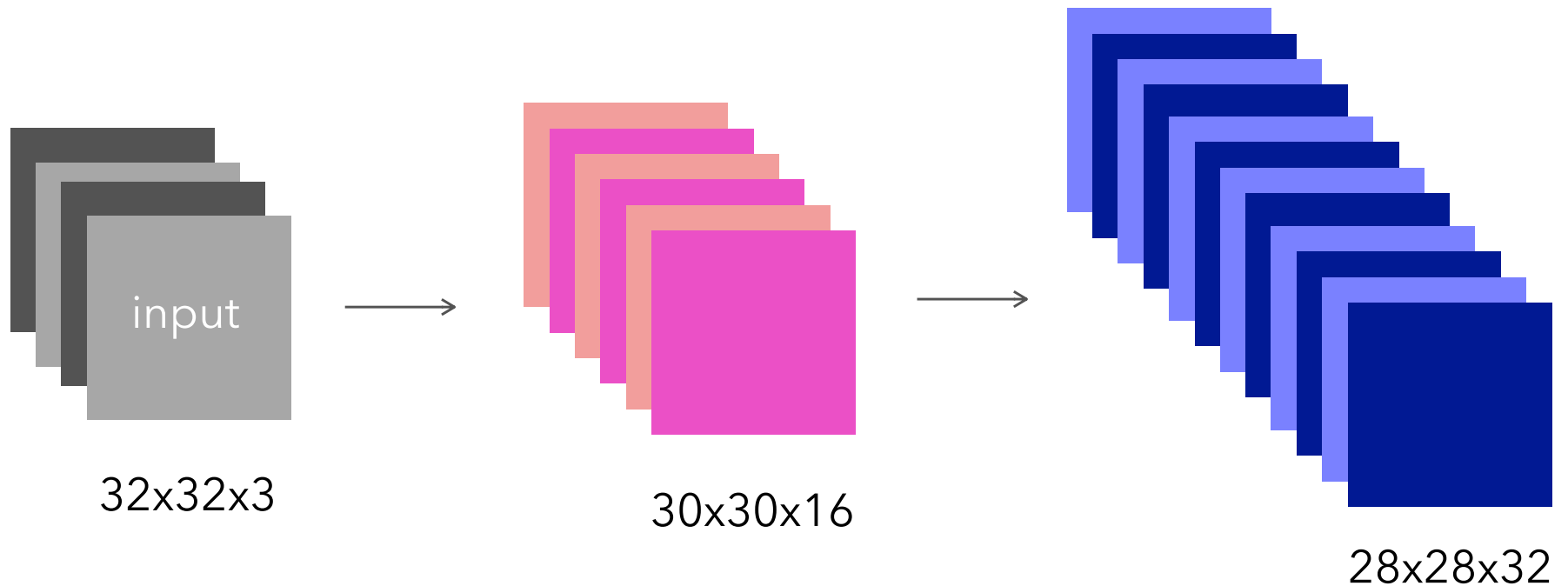
- **Solución:** Cada FM tendría 30×30 neuronas. La acción de cada filtro queda definida por una matriz de $3 \times 3 \times 3$ que genera un resultado de 30×30 . Al apilar los resultados obtenemos un volumen de $30 \times 30 \times 16$.



- **Continuación:** Si ahora apilamos en profundidad una capa convolucional 2D con 32 filtros (FMs) de 3×3 (campos receptivos) y strides de 1×1 , qué forma tomará el volumen de datos?

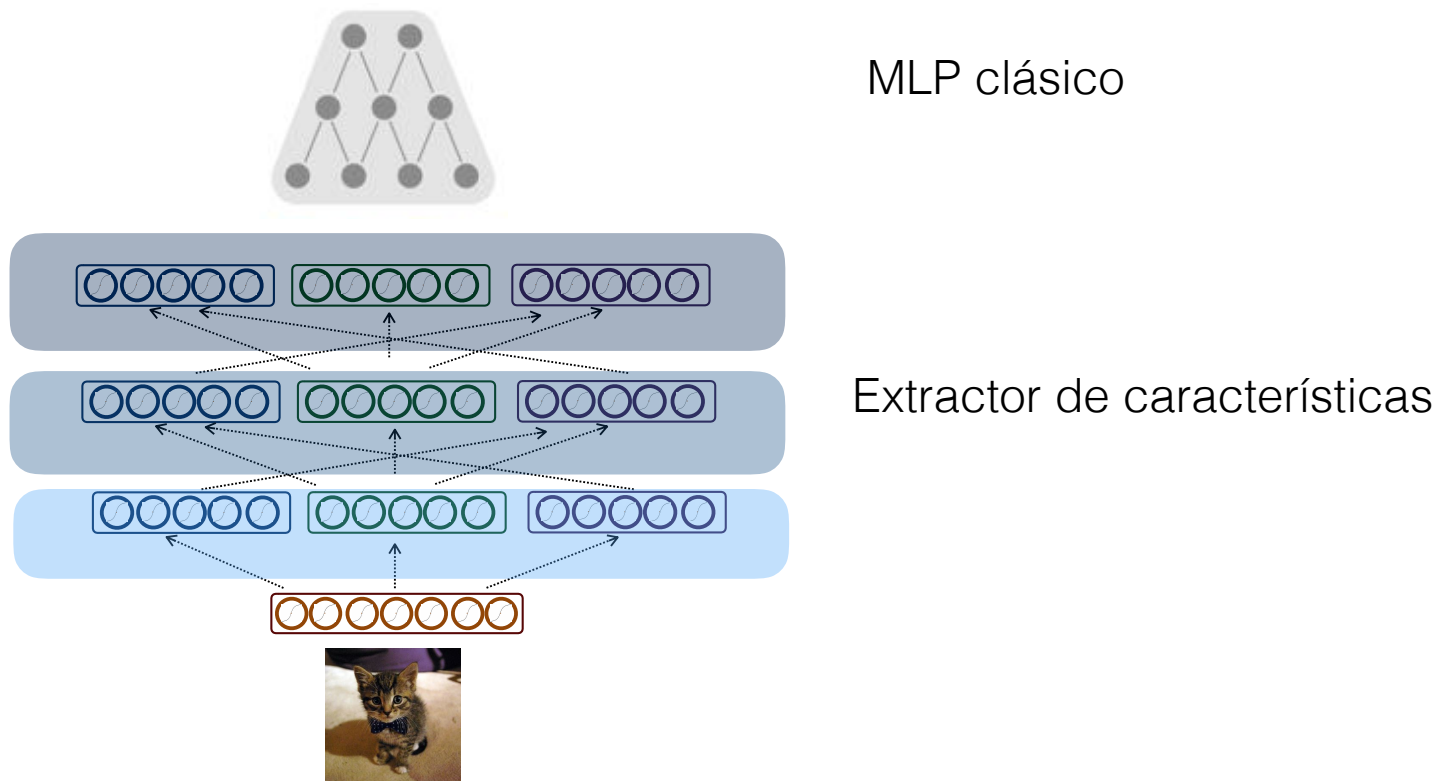
Ejercicio

- **Solución:** Cada FM tendría 28×28 neuronas. La acción de cada filtro sobre el volumen de entrada la controla una matriz de $3 \times 3 \times 16$ que genera un resultado de 28×28 . Al apilar los resultados obtenemos un volumen de $28 \times 28 \times 32$.



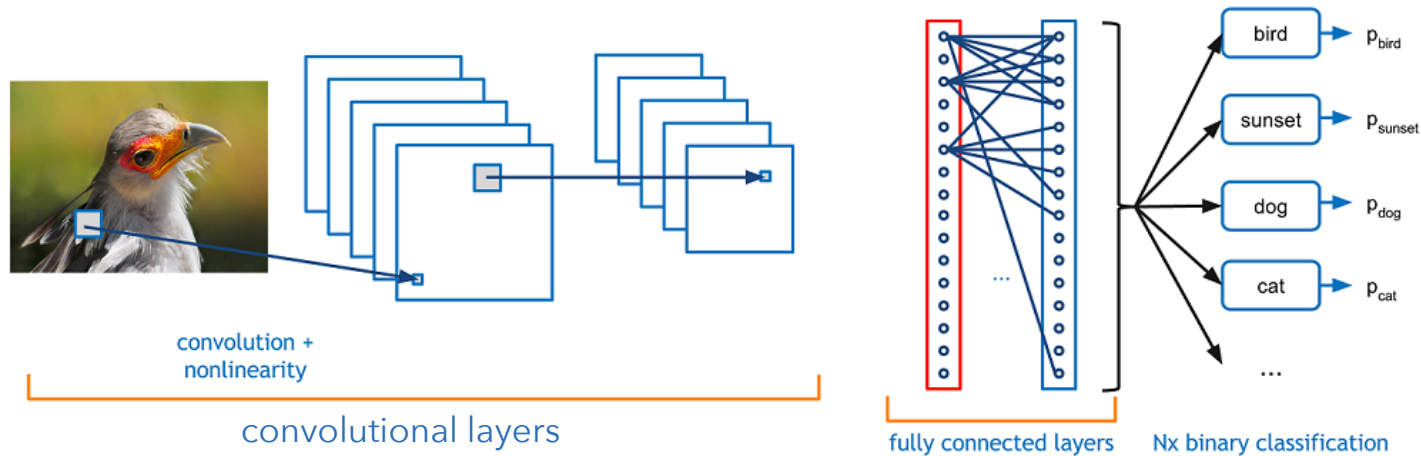
Capas Convolucionales

- Después de apilar un determinado número de capas convolucionales, tendremos características suficientemente potentes como para alimentar una red feed-forward densa.



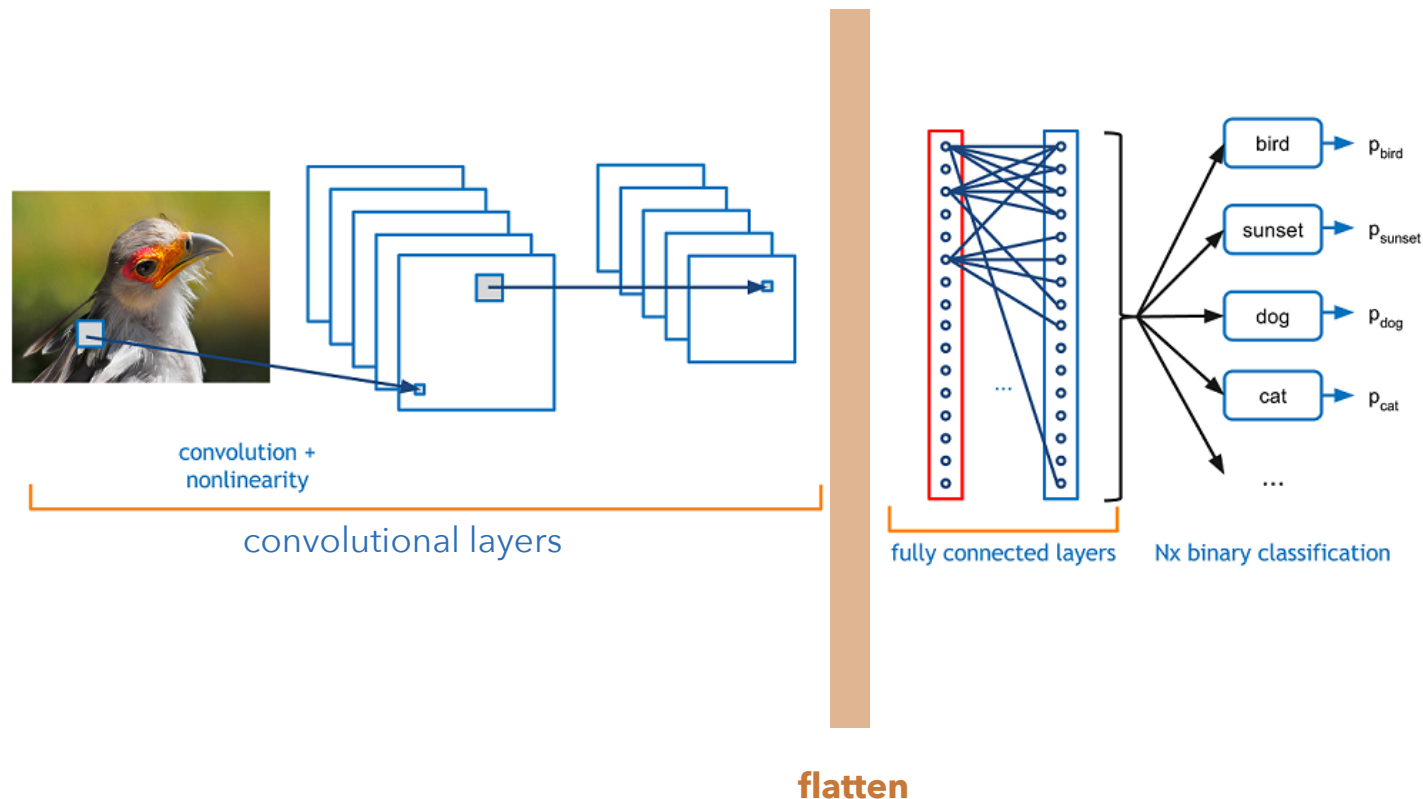
Capas Convolucionales

- Después de apilar un determinado número de capas convolucionales, tendremos características suficientemente potentes como para alimentar una red feed-forward densa.



Capas Convolucionales

- Normalmente la unión entre la parte convolucional y la no convolucional se hace re-organizando el volumen de datos de forma uni-dimensional.



Entonces ...

- Una red convolucional (CNN) estará formada en la práctica por múltiples filtros o canales de forma consistente que producen mapas de activación apilables en una dimensión extra.
- Una capa 2D puede procesar arreglos 3D actuando sobre cada canal extra por separado y luego sumando los resultados. De este modo la capa puede recibir un volumen 3D y entregar un volumen 3D.
- Una capa 1D puede procesar arreglos 2D actuando sobre cada canal extra por separado y luego sumando los resultados. De este modo la capa 1D puede recibir un volumen 2D y entregar un volumen 2D.
- Estas ideas permiten componer capas convolucionales en profundidad extrayendo patrones cada vez más abstractos y globales sobre el volumen de entrada.
- Normalmente una secuencia de capas convolucionales se termina con una o más capas densas especializadas en regresión o clasificación.



Tarea

- Suponga que tenemos un patrón de entrada de 1024×30 y aplicamos una capa convolucional 1D de 60 FC sobre la primera dimensión con campo vista de 10 (10×1) y stride unitario. Explique la forma de arreglo de neuronas de cada FC, la forma del arreglo de parámetros de cada FC y la forma del volumen de datos después atravesar la capa.
- ¿Es posible apilar una capa convolucional 1D sobre una 2D?
- ¿Puede dar un ejemplo en que le parece sensato utilizar capas convolucionales 1D? Explique.

