UAB
Universitat Autònoma de Barcelona

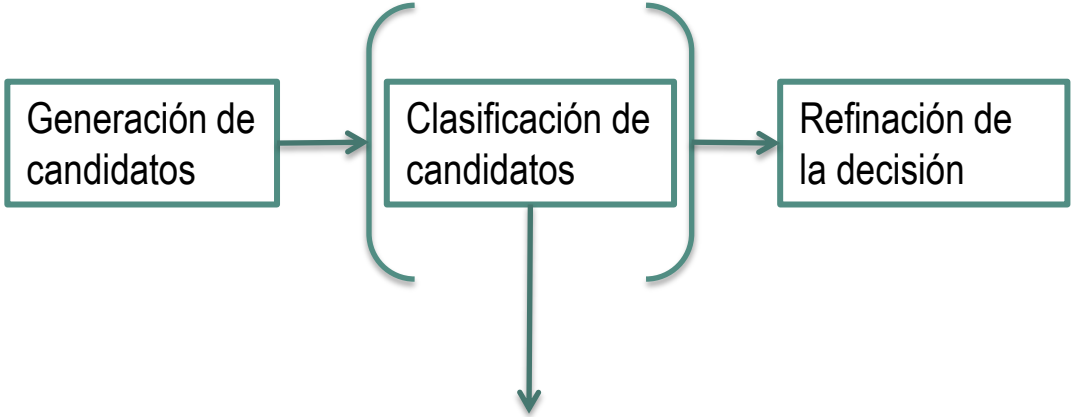
TÉCNICAS AVANZADAS

Convolutional Neural Networks (CNNs)

Antonio M. López

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

SECUENCIA DE PROCESAMIENTO:

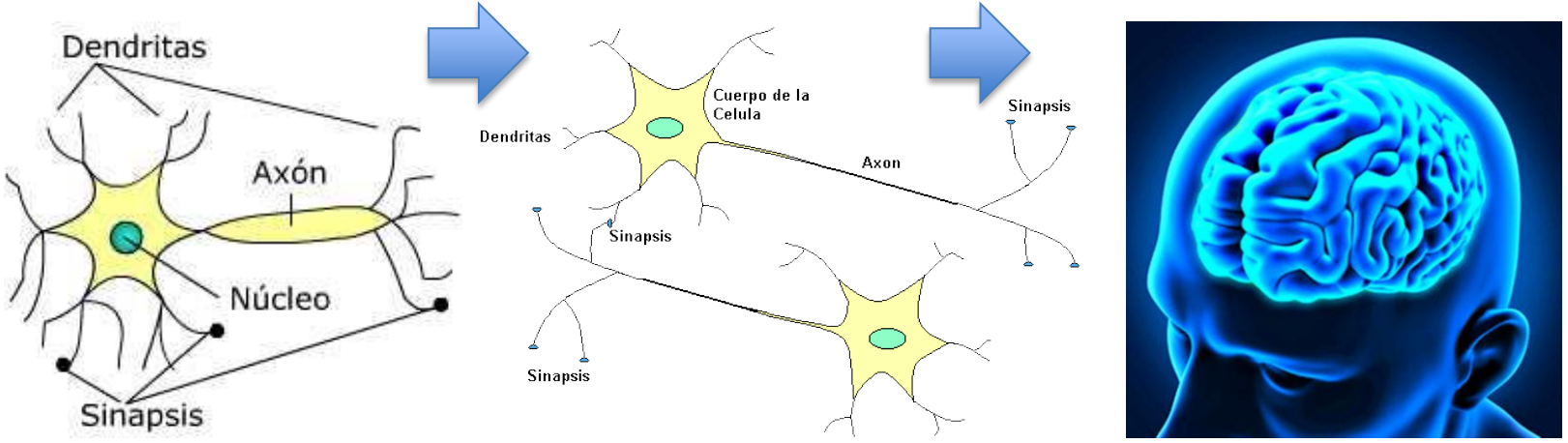


Clasificador ($\mathcal{C}_{\boldsymbol{w},\mathrm{T}}(\boldsymbol{x})$): Descriptor ($\boldsymbol{x}$) + Modelo ($\boldsymbol{w}$) + Umbral ($\mathrm{T}$).

Descriptores: HOG, LBP, Haar … ➔ diseñados "a mano" ➔ dos décadas haciendo propuestas.

Modelos: Regresión Logística, SVM, AdaBoost, RF … ➔ genéricos, se aprenden.

Umbral: fijado para operar en un rango de FPPI vs Tasa de Error.

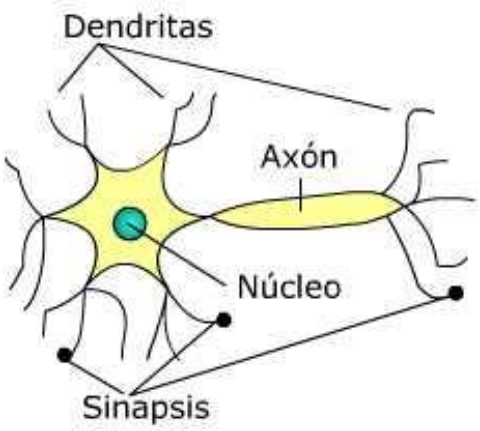


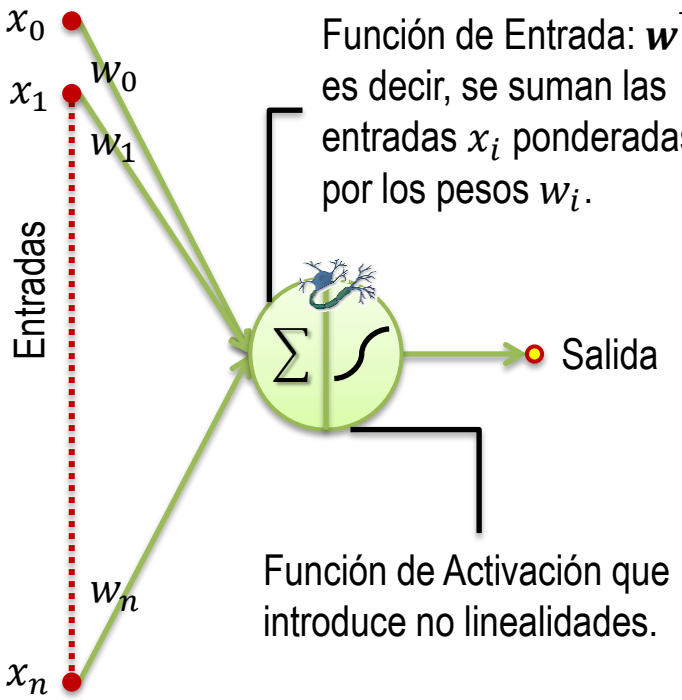
El cerebro como inspiración: redes neuronales artificiales (ANN: *artificial neural networks*)

➔ Aprender el clasificador en su totalidad, es decir, **aprender los descriptores** en lugar de diseñarlos a mano.

➔ La idea tiene más de 40 años, pero hasta hace poco era muy complicado llevarla a la práctica en problemas de visión por computador debido a que la capacidad de cálculo no estaba “a la altura” por un coste relativamente bajo. Las GPUs (procesadores gráficos) han tirado abajo ese muro.

Neurona Artificial: modelo matemático simple de una neurona (McCulloch & Pitts, 1943)

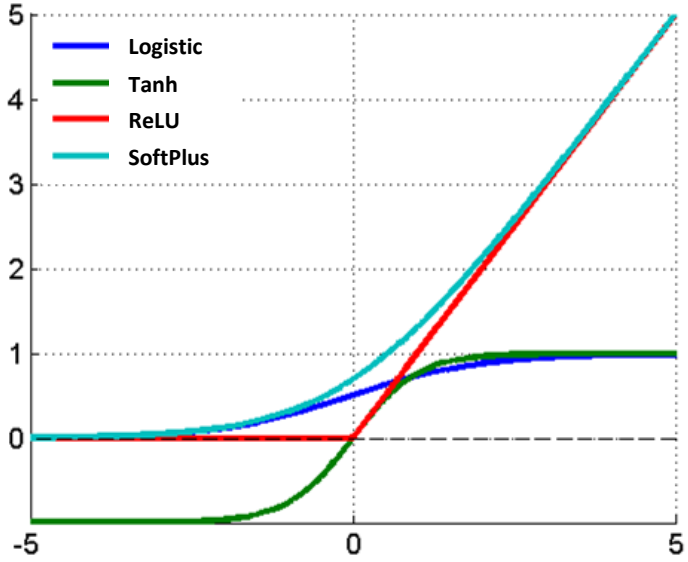




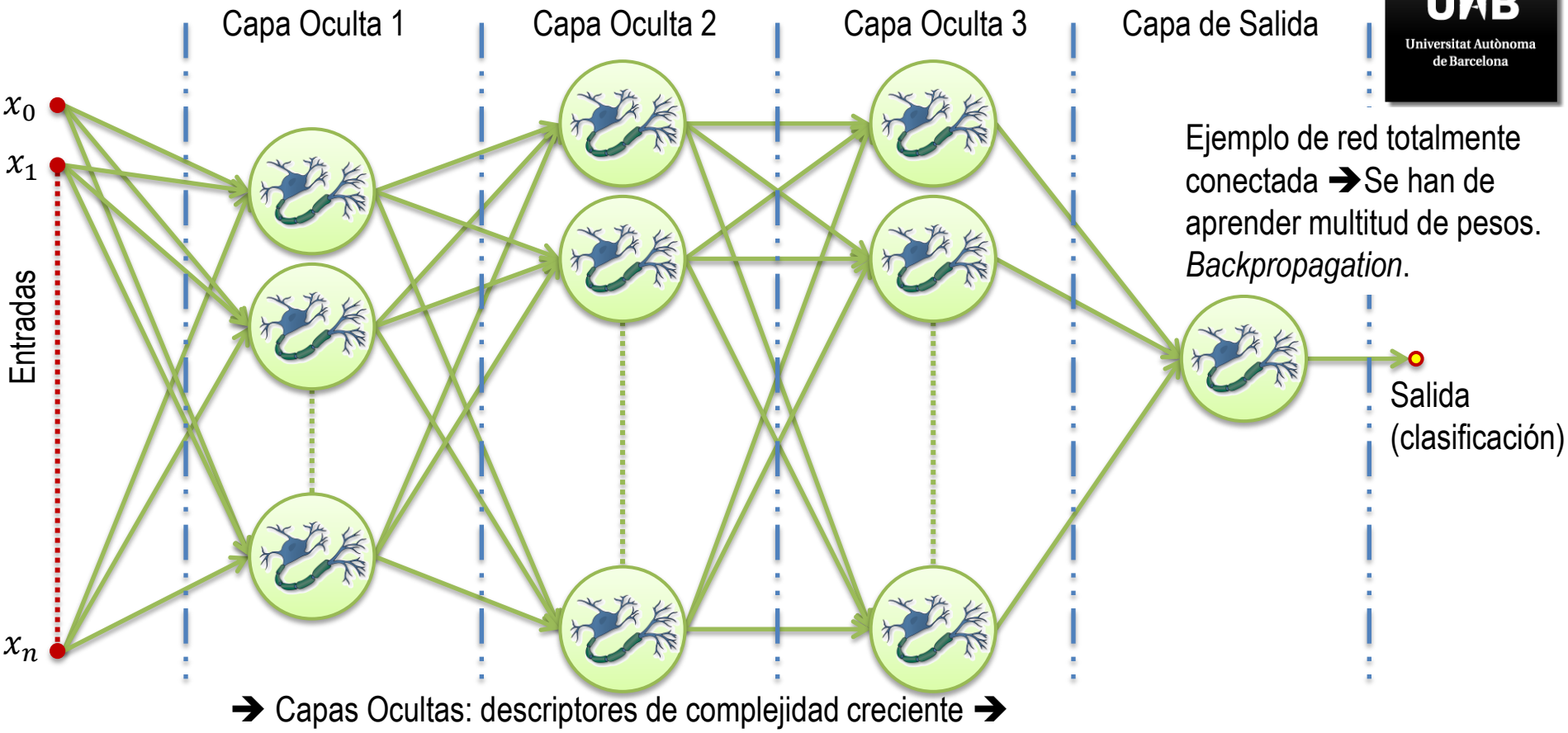
Función de Entrada: $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$, es decir, se suman las entradas x_i ponderadas por los pesos w_i.

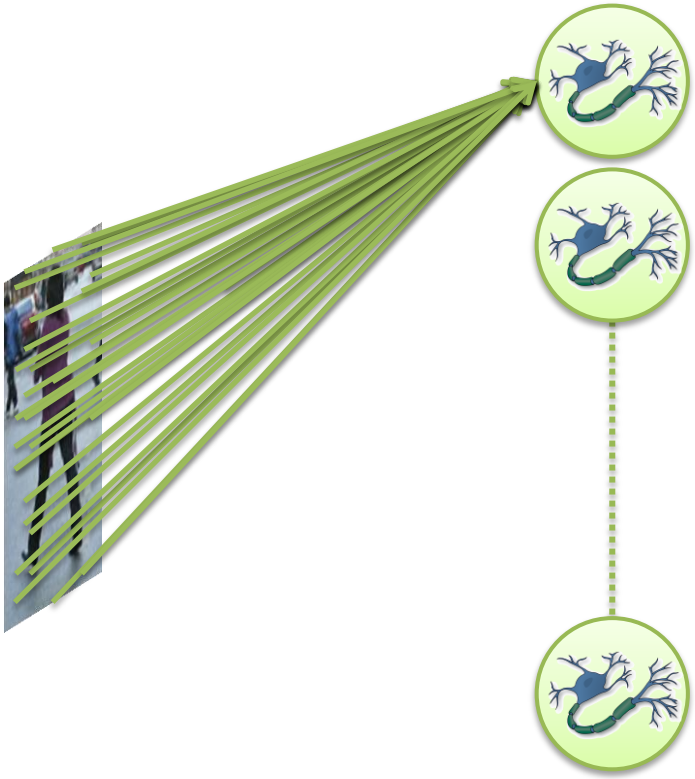
Función de Activación que introduce no linealidades.

Ejemplo ➔ *Logistic* ($\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$)



Perceptrón: no hay capas ocultas; Perceptrón Multicapa: hay capas ocultas (MLP: *multi-layer perceptron*)





Caso de ANN totalmente conectada

Imagen: 64 x 128 ➔ 8.192 píxeles

Si

tenemos 1000 neuronas en esta 1ª capa,

Entonces,

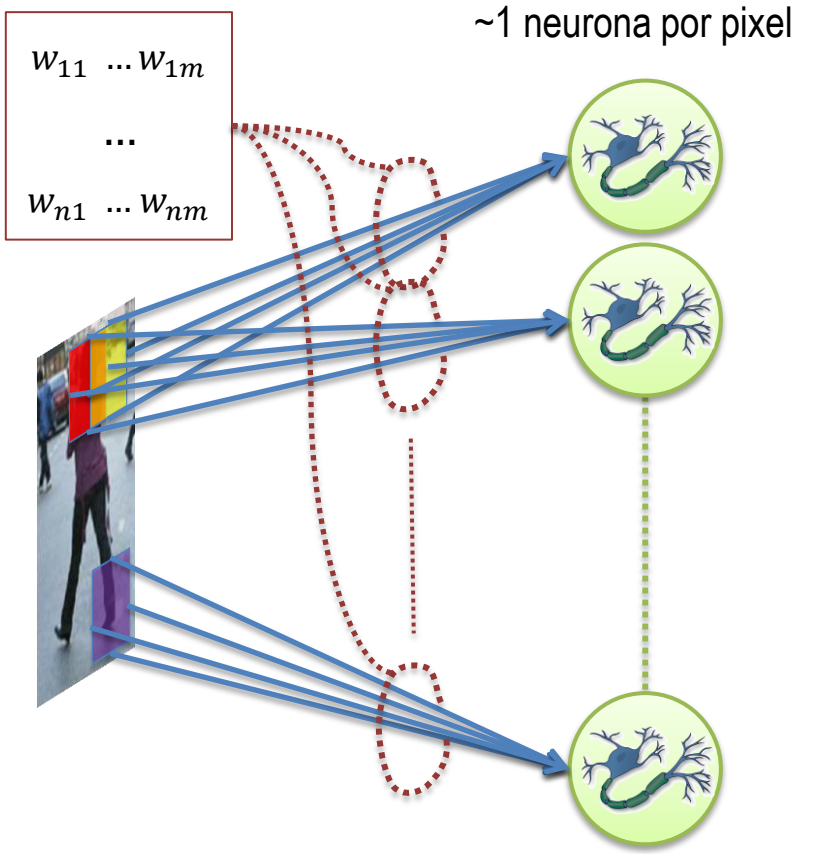
solo para esta capa, tenemos ~8M de parámetros.

Como los descriptores se aprenden de los datos, la cantidad de ejemplos que necesitaríamos sería “descomunal”.

- En visión por computador podemos explotar algunas observaciones para reducir el número de parámetros y definir una topología de red más parecida a sistemas visuales de inspiración biológica.

- Neuronas del córtex visual de los gatos (Hubel & Wiesel, 1959, 1962):

  ➔ Cada neurona “simple” del córtex se responsabiliza de una pequeña región “bidimensional” del campo visual.
  ➔ Hay solapamiento para cubrir todo el campo visual.
  ➔ Estas neuronas actúan como filtros locales basados en convolución.
  ➔ El filtro se repite actuando localmente.
  ➔ Hay distintos tipos de filtros (p.e., detectores de contornos orientados).
  ➔ Las neuronas “complejas” abarcan mayor campo visual y son más robustas a la posición exacta de los estímulos: capa con sub-muestreo.

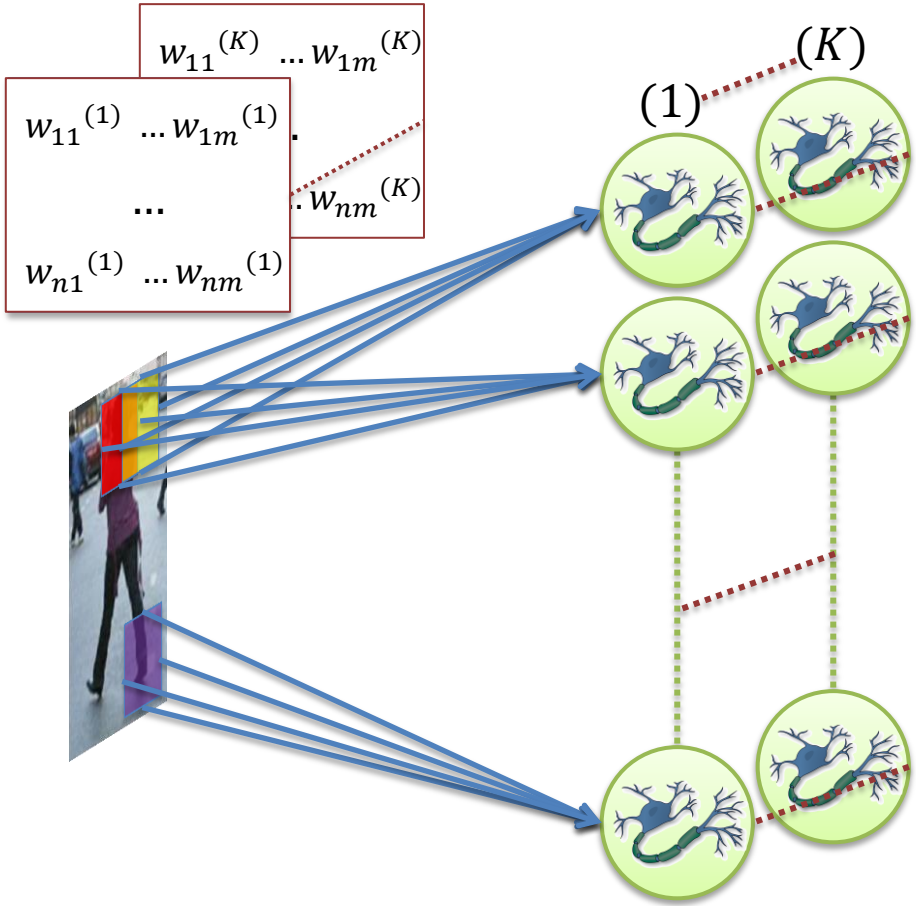


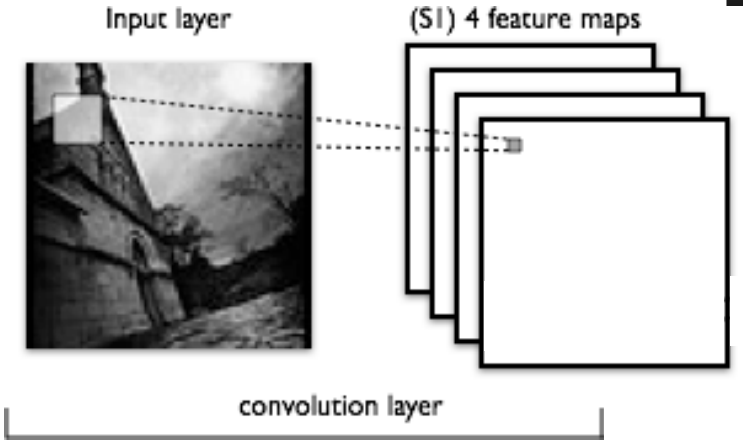
- Cada neurona “simple” del córtex se responsabiliza de una pequeña región “bidimensional” del campo visual.
- Hay solapamiento para cubrir todo el campo visual.
- Estas neuronas actúan como filtros locales basados en convolución.
- El filtro se repite actuando localmente.

➔ Primera capa de la red: por cada pixel una neurona de CONVOLUCIÓN (excepto bordes).

➔ Hablamos de redes neuronales de convolución (*convolutional neural networks*: CNNs).

Es el mismo filtro de convolución para todas las neuronas ➔ aprender $n \times m$ parámetros (ignorando el sesgo/*bias*).

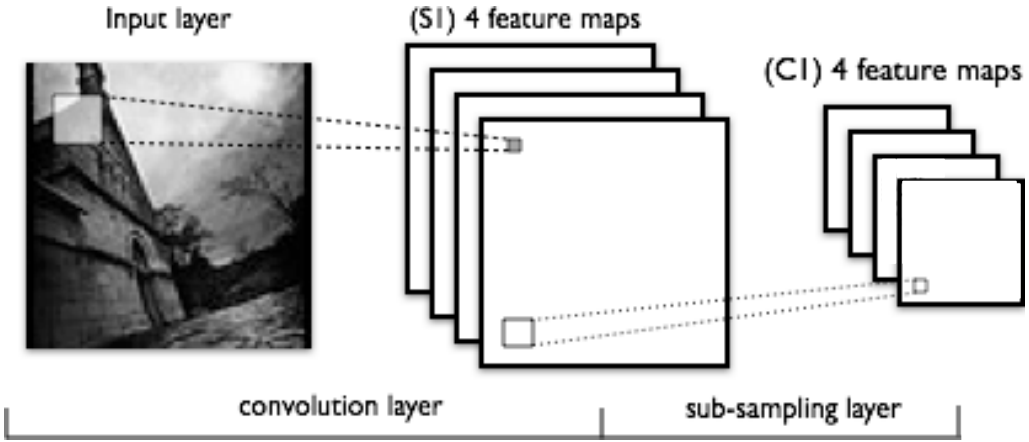




- Hay distintos tipos de filtros.

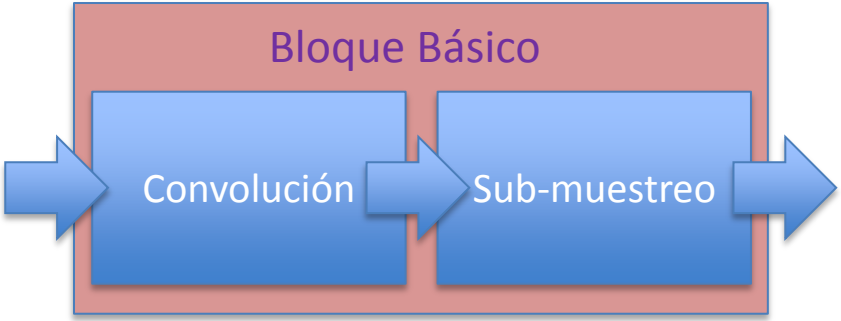
➔ Primera capa de la red: por cada pixel varias Neuronas diferentes de CONVOLUCIÓ (excepto bordes) ➔ BANCO de FILTROS.

➔ Aprender $K \times n \times m$ parámetros (ignorando el sesgo/*bias*).



- Las neuronas “complejas” abarcan mayor campo visual y son más robustas a la posición exacta de los estímulos: capa con **sub-muestreo**.

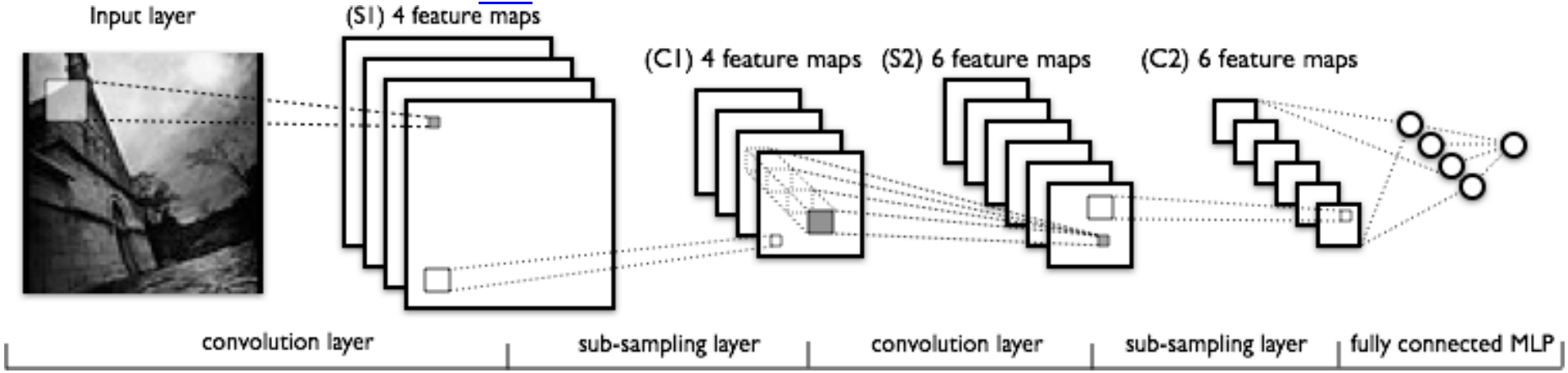
➔ Se define una rejilla sin solapamiento, con celdas de tamaño constante.
➔ De cada celda sacaremos un solo valor.
➔ Ejemplos: *max-pooling*; promediado; parametrizados.



- Hiperparámetros (decisiones):
  - Tamaño filtros ($n \times m$).
  - Número de filtros (K).
  - Tipo de sub-muestreo.
  - Tamaño celda de sub-muestreo.

Ejemplo muy popular: **LeNet**.

http://yann.lecun.com/exdb/lenet/
http://deeplearning.net/tutorial/lenet.html

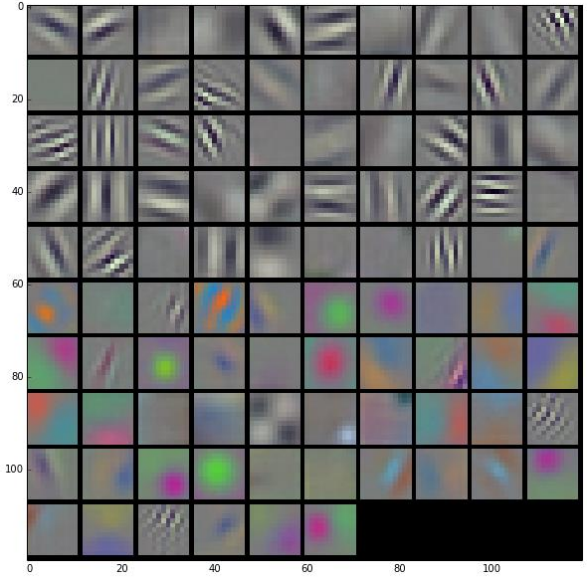


➔ Capas Ocultas: descriptores de complejidad creciente ➔

Clasificación (multiclase)

➔ Cuestión clave: topología de la red (capas y conexiones)

Visualización de algunas capas en distintas CNNs que podemos encontrar en la literatura.
http://cs231n.github.io/understanding-cnn/



Filtros para detectar altas frecuencias en escalas de gris.

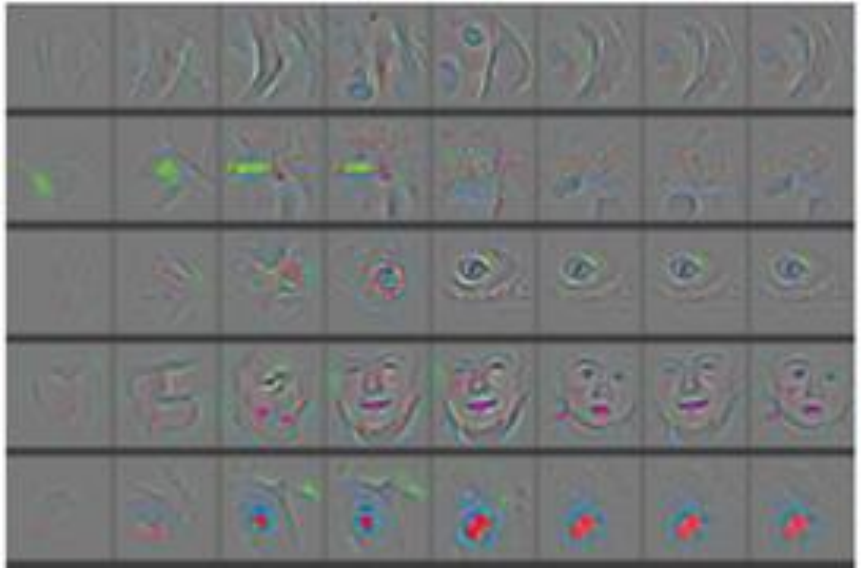
Filtros para detectar características de baja frecuencia de los canales de color.

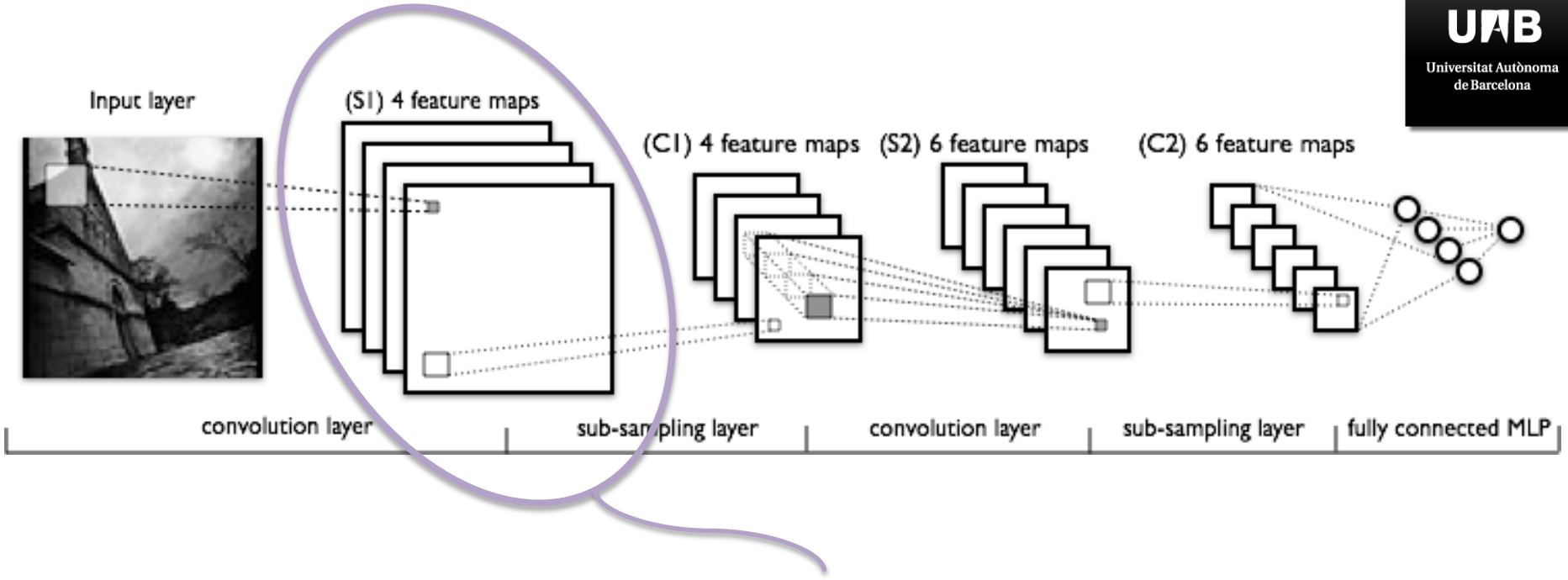
Primera capa de convolución en AlexNet

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," NIPS'2012.

Quinta capa en la CNN propuesta en:

Matthew D. Zeiler, Rob Fergus, "Visualizing and Understanding Convolutional Netwroks," ECCV'2014.





➔ "**Off-the-self features**": usar el esquema clásico pero con descriptores aprendidos.

➔ Cogemos el banco de filtros, se aplica en la imagen y luego se usa SVM, AdaBoost, RF, DPM, etc.

Ali S. Razavian, Hossein Azizpour, Josephine Sullivan, Stefan Carlsson, "CNN Features off-the-self: an Astounding Baseline for Recognition," CVPR'2014.

- Conceptos clave de este vídeo:
  - Descriptores diseñados a “mano” vs “aprendidos”.
  - Las CNNs permiten aprender esos descriptores, con niveles de complejidad crecientes, e incorporando la etapa de clasificación, es decir, todo bajo el mismo paradigma.