

Introducción a los Sistemas Inteligentes

Sistemas Inteligentes

***SISTEMAS COMPUTACIONALES QUE TIENE LA
HABILIDAD DE **RAZONAR Y APRENDER**
PARA TRATAR NUEVAS SITUACIONES.***

***FRECUENTEMENTE ESTA BASADA EN LA
EMULACIÓN DE UNO O MAS ASPECTOS DE
LOS SISTEMAS BIOLÓGICOS O SOCIALES***

Inteligencia Artificial

Inteligencia: capacidad de **adquirir y usar** conocimiento

¿Cómo el cerebro percibe, entiende, predice y manipula?

⇒ Razonar, Aprender, Comunicarse y Actuar

- *La Inteligencia Artificial trata de conseguir que los computadores **simulen en cierta manera la inteligencia humana.***
- *Se acude a sus técnicas cuando es necesario **incorporar en un sistema informático, conocimiento o características propias del ser humano.***

Aspectos y Características de la Inteligencia Artificial

Envuelve entender cómo el *conocimiento* es adquirido, representado y almacenado; cómo el *comportamiento inteligente* es generado; cómo *las, emociones, y pensamientos* son desarrollados y usados; cómo las *señales sensoriales* son transformadas en símbolos; cómo los *símbolos* son utilizados para actuar lógicamente, razonar, planear; cómo el *cerebro* produce la ilusión, las creencias, las esperanzas, el temor y los sueños

Entre otras cosas

Algunos objetivos de la Investigación en Inteligencia Artificial

- Emulación de la forma de **razonamiento humano** para la resolución de problemas,
- **Reconocimiento de patrones** que abarca la comprensión y la síntesis del habla, de imágenes y la visión artificial.
- **Representación del conocimiento**, conceptualización cognoscitiva, **procesamiento del lenguaje natural**.
- **Emulación de comportamientos y sistemas biológicos**: cerebro, proceso evolutivo, etc.
- Interés en el **control y coordinación autonómica**.
- Estudio de la **inteligencia colectiva** (técnicas bio-inspiradas): Colonias de Hormigas, Colonias de Abejas, ...

Aplicaciones



Es inimaginable

*En todas las áreas
puede ser usada*



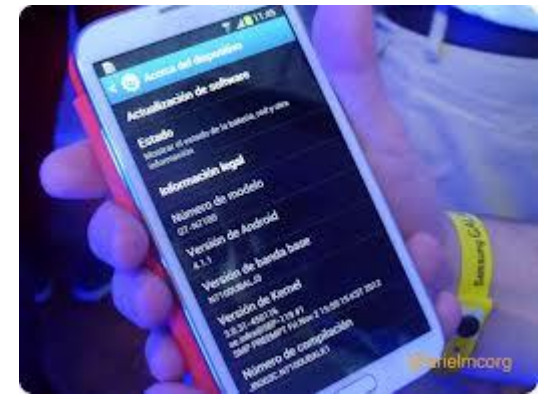
**Próxima década prácticamente
Todo el software tendrá algo de la IA**

Reconocimiento

Predicción

Optimización

Planificación



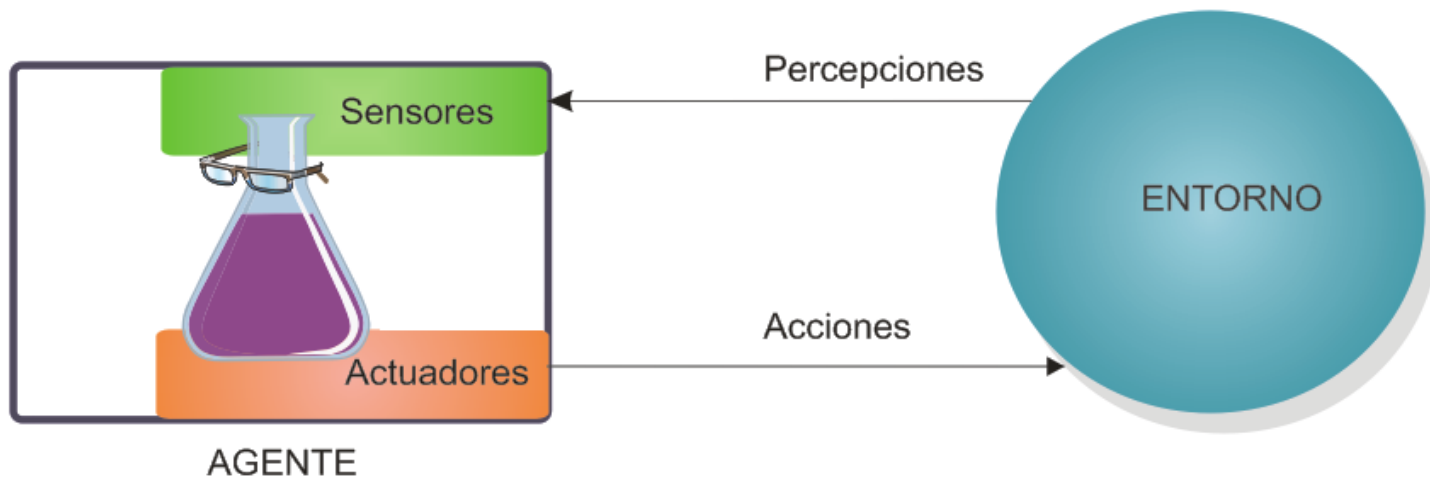
Áreas

- **Inteligencia Artificial**
 - Teoría de agentes
 - Sistemas multiagentes (Inteligencia Artificial Distribuidas)
- **Técnicas Inteligentes clásicas (computación inteligente)**
 - Redes neuronales artificiales
 - Lógica difusa
 - Computación Evolutiva
- **Técnicas Inteligentes distribuidas**
 - Algoritmos inspirados en colonias de insectos: PSO, ACO, etc.
 - Algoritmos inspirados en fenómenos físicos: Flujo de Agua, de gas, TS, etc.
 - Algoritmos inspirados en sistemas biológicos: Sistema Inmune Artificial, etc.
- **Técnicas inteligentes avanzadas autónomas:**
 - Sistemas emergentes y auto-organizados (Inteligencia colectica)
 - Computación autonómica

Agentes

Los agentes de software son capaces de ***decidir por sí mismos*** qué hacer en una determinada situación.

Mantienen información acerca de su entorno, y ***toman decisiones*** en función de su percepción del estado de dicho entorno, sus experiencias anteriores, y los objetivos que tienen planteados. Además, los agentes ***pueden comunicarse*** con otros agentes para colaborar y alcanzar objetivos comunes. [Denney 2008]



Procedimiento de base:

1. Percibo (Actualiza *Memoria*)
2. Decido (Escoge *Acción*)
3. Actúo (Actualiza *Memoria*)

Descripción práctica de un agente:

Sus Tareas.

Sus Conocimientos.

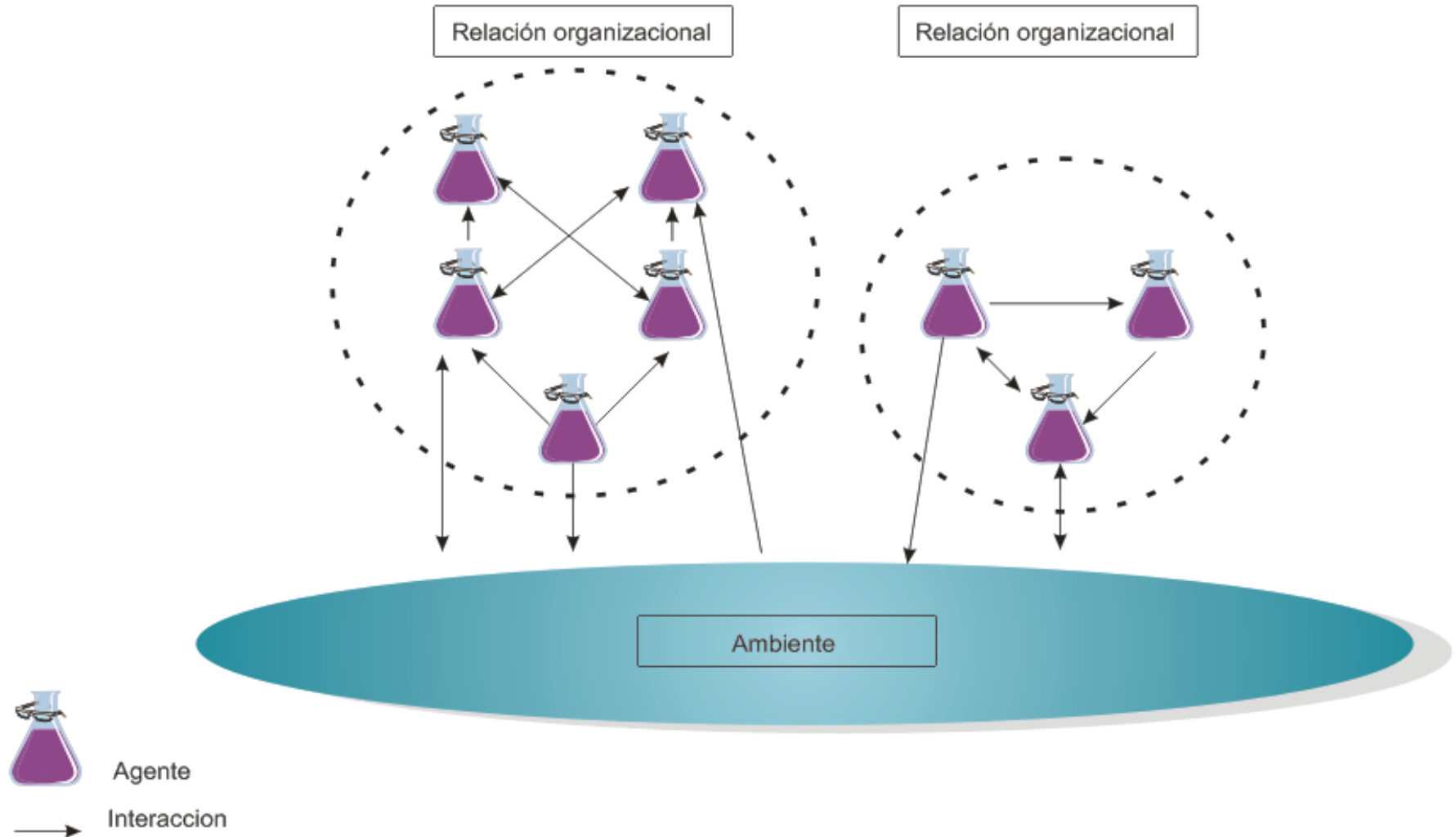
Su Comunicación

Otros aspectos importantes a considerar en los Agentes

- Mecanismos para **resolver un problema**
- Mecanismo para **planificar sus actividades /tareas**
- Mecanismos para **representar el conocimiento**
- Mecanismo de **razonamiento**
- Mecanismos de **aprendizaje**
- Mecanismos de **percepción**
- Mecanismos para **comunicarse**



Sistemas Multiagentes



<http://www.ing.ula.ve/~aguilar/actividad-docente/SMAAn.htm>

Temas en los Sistemas Multiagentes

1. Metodologías de desarrollo de SMA
2. Interacción y sus tipos: Cooperación, Coordinación, Negociación
3. Comunicación
4. Problema de Confianza y Reputación
5. Aprendizaje en los SMA (aprendizaje colectivo)
6. Plataformas de Despliegue
7. Arquitecturas de Referencia en los SMA.
8. Sistemas Emergentes Auto-organizados

Red Neuronal Artificial

Rafael Yuste, ideólogo del Proyecto BRAIN:



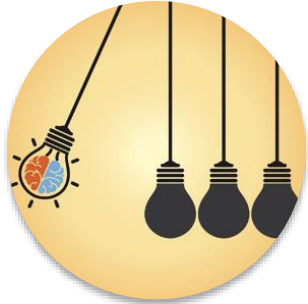
Cerebro

➤ “Sólo conocemos un 3 o 4% del cerebro”



Usan algoritmos muy superficiales del funcionamiento del cerebro

Muchos interrogantes sin responder
¿Conciencia,
Libre albedrío,
Qualía
pensamiento?



Como logra el optimo uso de energía?



✓ Tesis

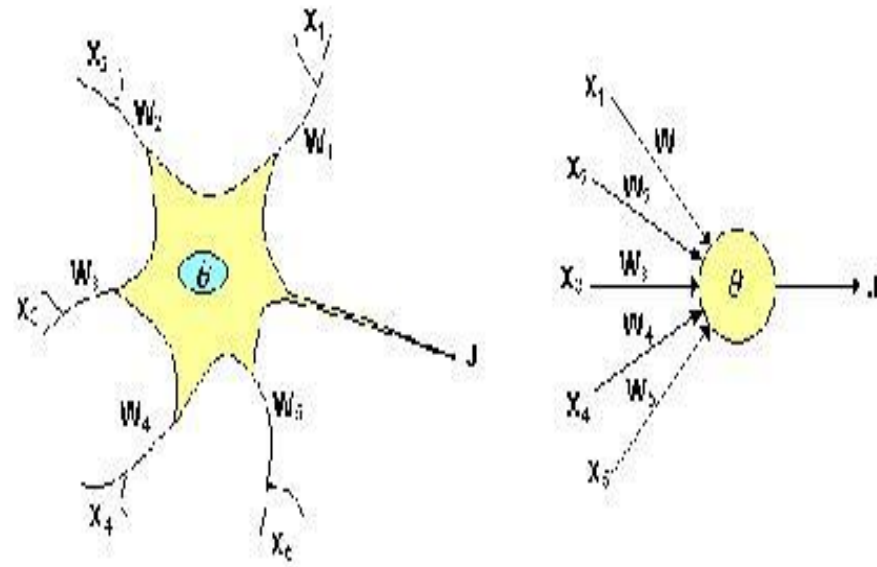
Modelos Neuronales Limitados.



Comportamiento, diversidad, Lesiones, alzheimer,..?

Red Neuronal Artificial

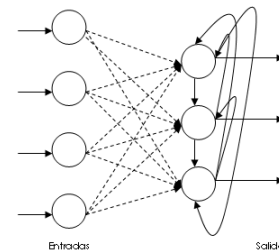
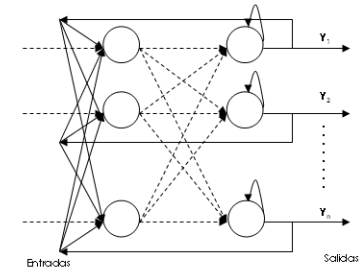
- ✓ Una **nueva forma de computación**, inspirada en el cerebro.
- ✓ Un **modelo matemático** compuesto por un gran **número de elementos de procesamiento**, eventualmente en niveles.
- ✓ Son **redes interconectadas masivamente** en paralelo de elementos simples y organización diversa.



Modelo de una RNA vs. el Modelo Biológico

COMO TRABAJA UNA RED NEURONAL

1. El conjunto de **unidades de procesamiento** (neuronas formales).
2. El **estado interno o de activación** de las neuronas.
3. Las **conexiones entre las neuronas**.
4. Las **conexiones con el ambiente**.
5. La regla de propagación
6. La función de activación
7. La función de transición o de salida
8. La **topología o arquitectura** de la red
- 9 El **algoritmo de Aprendizaje**



APRENDIZAJE

- Al finalizar la fase de entrenamiento/aprendizaje de una RNA, se espera que la red haya aprendido lo suficiente para **resolver otro problema similar** satisfactoriamente.

MODIFICAR PESOS DE LAS CONEXIONES DE
LAS NEURONAS (CREAR, DESTRUIR,
MODIFICAR)

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$$

Modelos Neuronales

Realimentados :

feed-propagation

ART,

HOPFIELD

Unidireccionales

PERCEPTRON,

M RN,

BOLTZMAN,

backpropagation

KOHONEN

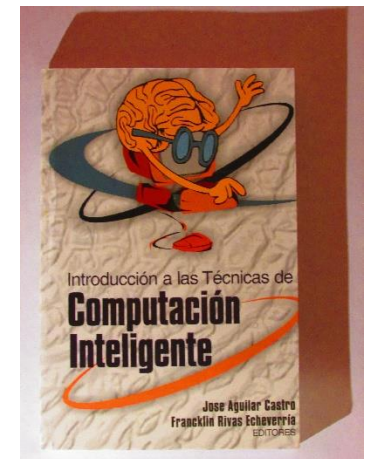
Híbridos:

RBF (RADIAL BASIC FUNCTION)

Redes basadas en DEEP LEARNING

Redes de Convolución

Extreme Learning

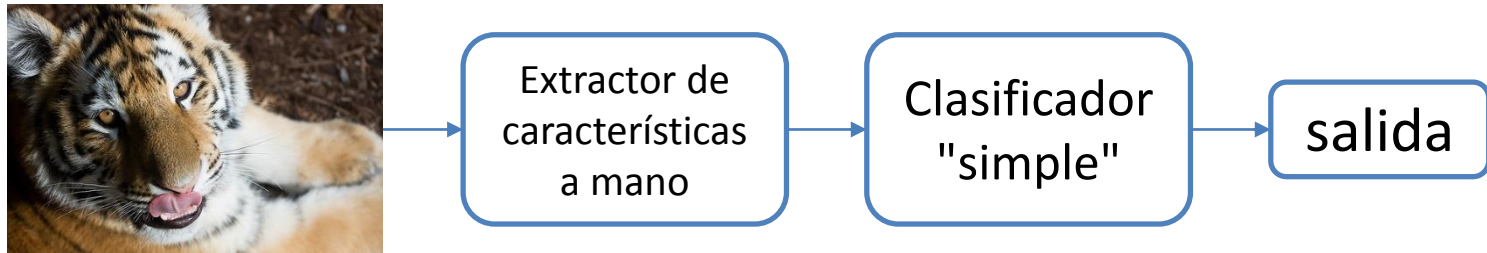


Temas Avanzados

- **RNA**
 - Deep Learning
 - Extreme Learning Machines
 - Online and Incremental Learning
 - Hybrid Learning

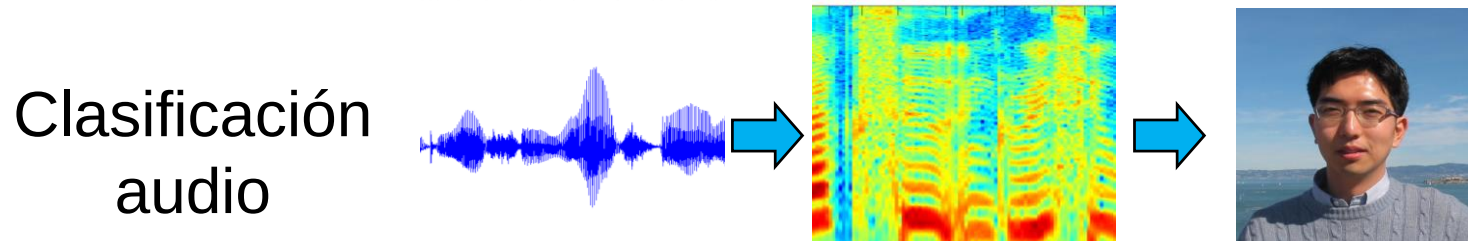
Introducción

- Los modelos tradicionales de reconocimiento de patrones usan características hechas a mano y un clasificador de entrenamiento relativamente simple.



- Este enfoque tiene las siguientes limitaciones:
 - Es muy tedioso y costoso desarrollar características hechas a mano
 - Las características hechas a mano suelen ser altamente dependientes en una aplicación, y no se pueden transferir fácilmente a otras aplicaciones

Representación de características



Problemas de características manuales

1. Necesita el conocimiento experto
2. Demasiado tiempo y costoso
3. No generaliza a otros dominios

nivel

Acción

Aprendizaje de características sin supervisión



Encuentre una mejor manera de representar las imágenes que los píxeles.

Representación de características

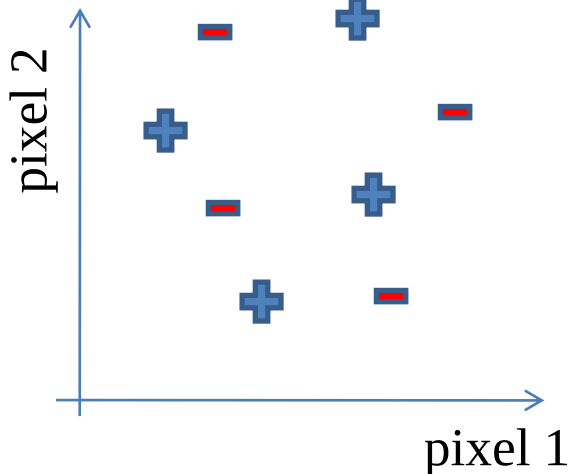


Algoritmo
aprendizaje

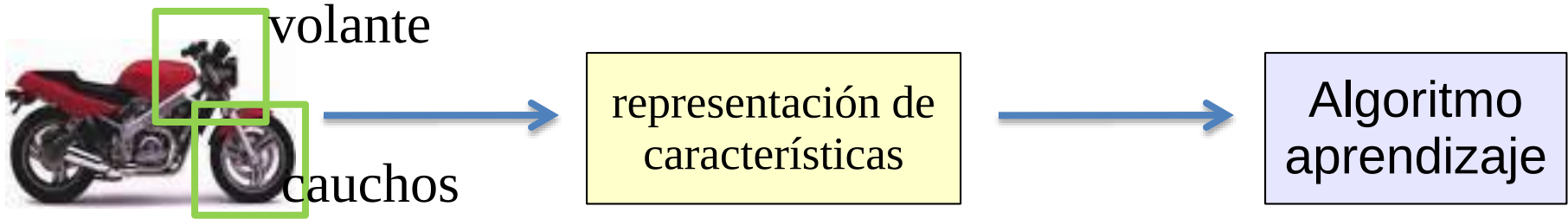
Entrada

Espacio entrada

+ Motos
- "No"-Motos

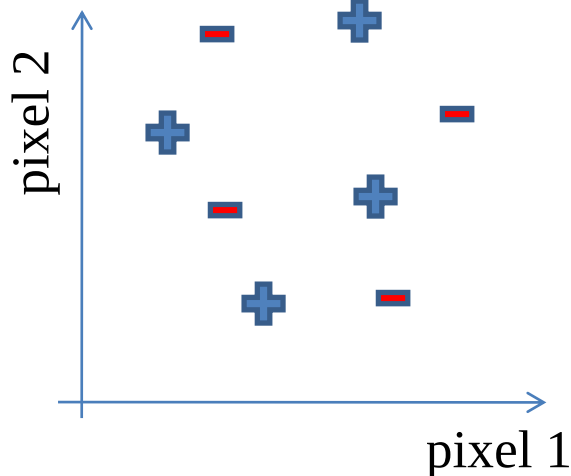


Representación de características

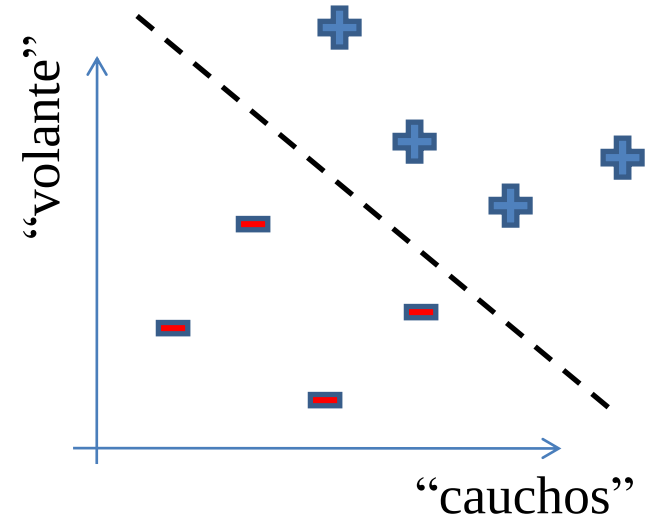


Entrada

Espacio entradas



Espacio Características

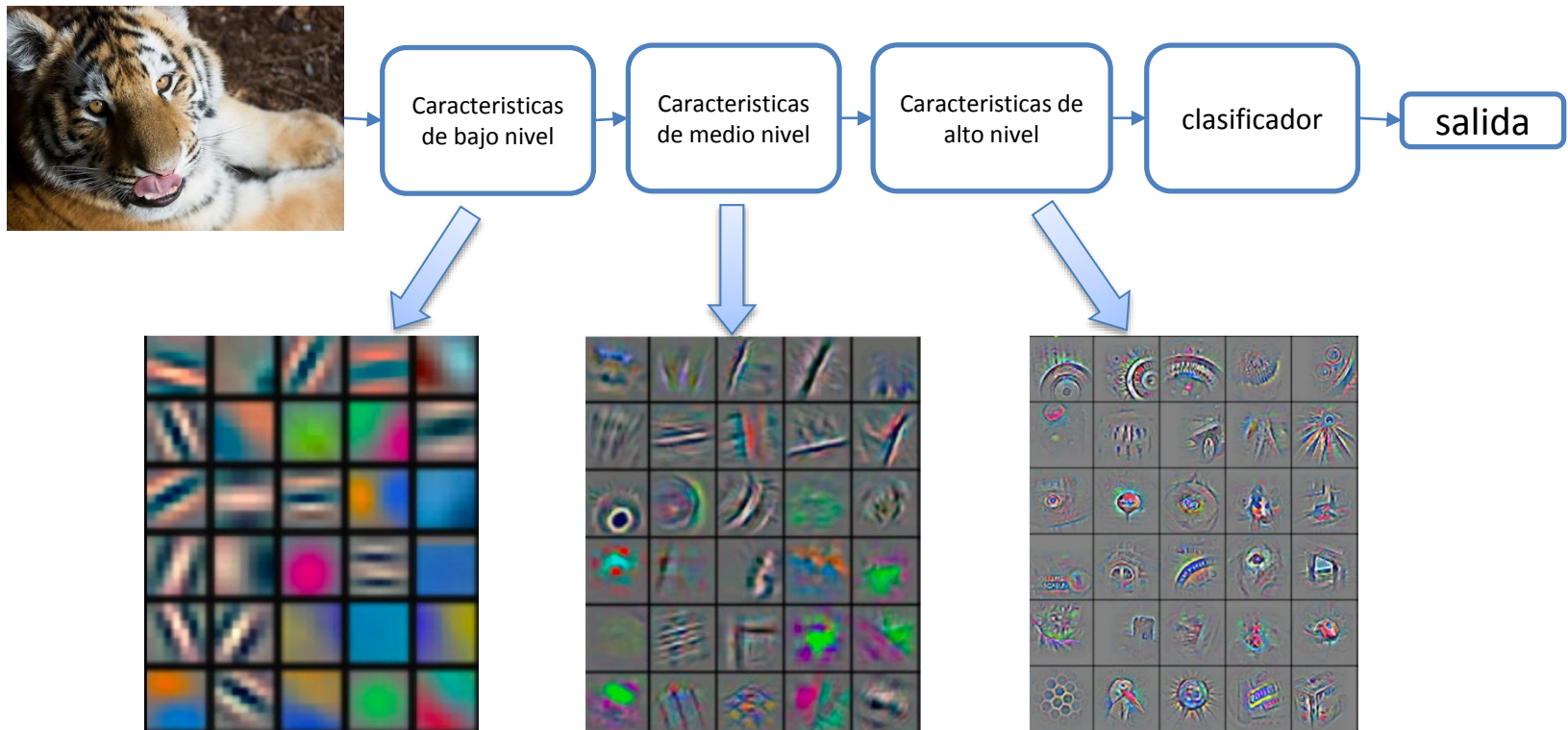


Aprendizaje de Representaciones

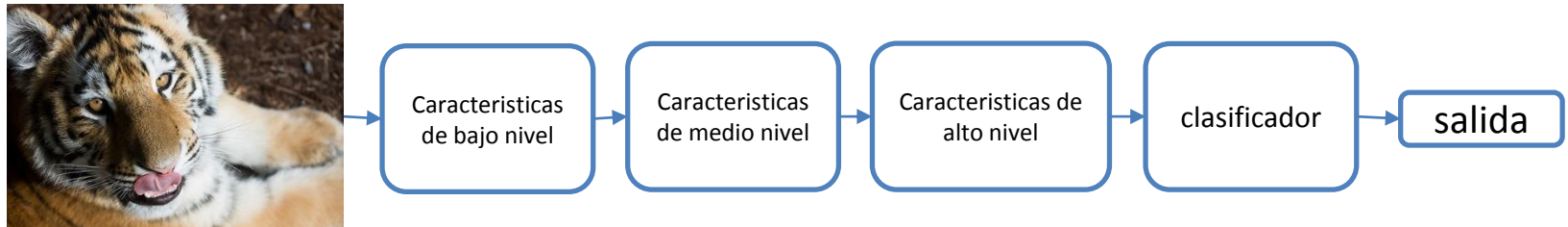


Deep Learning

- El aprendizaje profundo busca aprender ricas representaciones jerárquicas (es decir, características) automáticamente a través de un proceso de múltiples etapas del aprendizaje de rasgos.

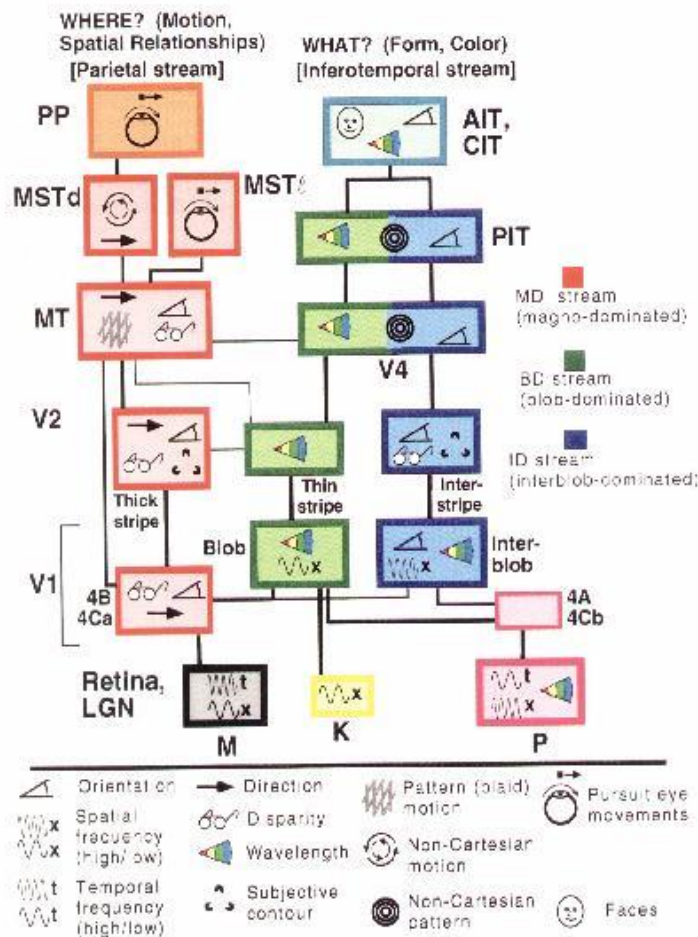


Aprendizaje de representaciones jerárquicas



- Jerarquía de representaciones con creciente nivel de abstracción. Cada etapa es un tipo de trama transformable no lineal manejable
- Reconocimiento de imagen
 - Píxel → borde → motivo → parte → objeto
- Texto
 - Carácter → palabra → grupo de palabras → oración → párrafo → historia

La corteza mamaria es jerárquica



(van Essen and Gallant, 1994)

- Es bueno inspirarse en la naturaleza,
- Necesitamos entender qué detalles son importantes y cuales no.
- Cada módulo en Aprendizaje Profundo transforma su representación de entrada en una de nivel superior, de manera similar a la corteza humana.

Aprendizaje profundo

algoritmos en aprendizaje automático para RNA multicapas.

- Los algoritmos pueden utilizar aprendizaje supervisado o no supervisado.
- Basados en el **aprendizaje de múltiples niveles de características**. Las características de más alto nivel se derivan de las características de nivel inferior para formar una representación jerárquica.
- Aprenden múltiples niveles de representación que corresponden a **diferentes niveles de abstracción**, que forman una jerarquía de conceptos.

Aprendizaje profundo

¿por qué es generalmente mejor?

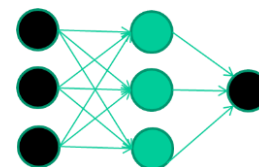
- Utilizan una red neuronal con **varias capas de nodos** entre la entrada y salida
- La serie de capas entre la entrada y salida hace una **identificación de característica y procesamiento en una serie de etapas**, al igual que nuestro cerebro .

Aprendizaje profundo

- Redes neuronales multicapas tienen muchísimos años.

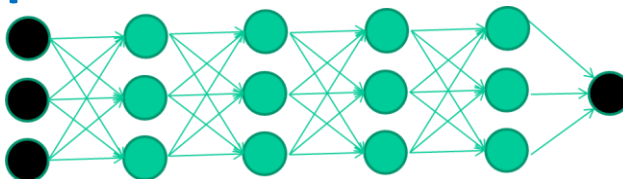
Qué es lo nuevo?

siempre han habido buenos algoritmos para el aprendizaje de la pesos en redes con 1 capa oculta

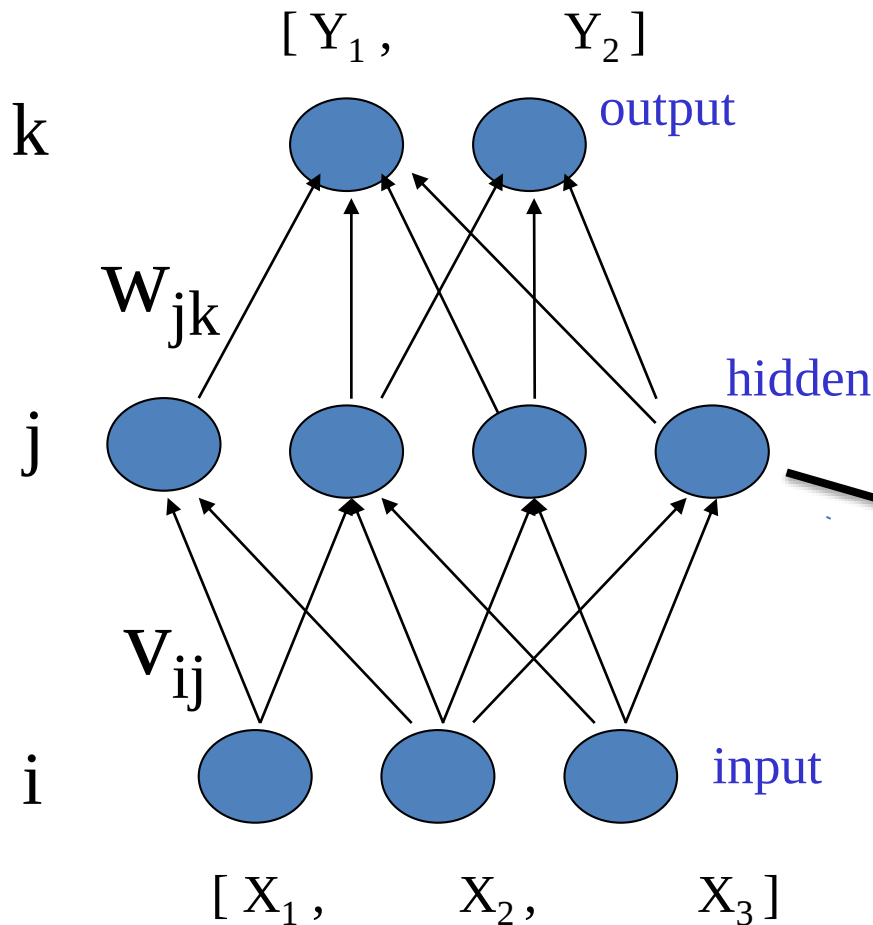


pero estos algoritmos no son buenos para aprender las ponderaciones de redes con más capas ocultas

Lo nuevo es: algoritmos para entrenar redes “mucho-más tarde”

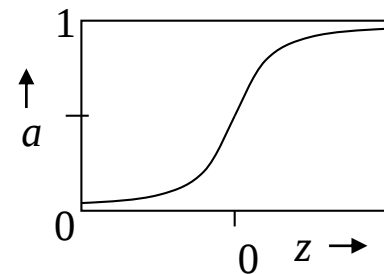


Perceptron Multicapa



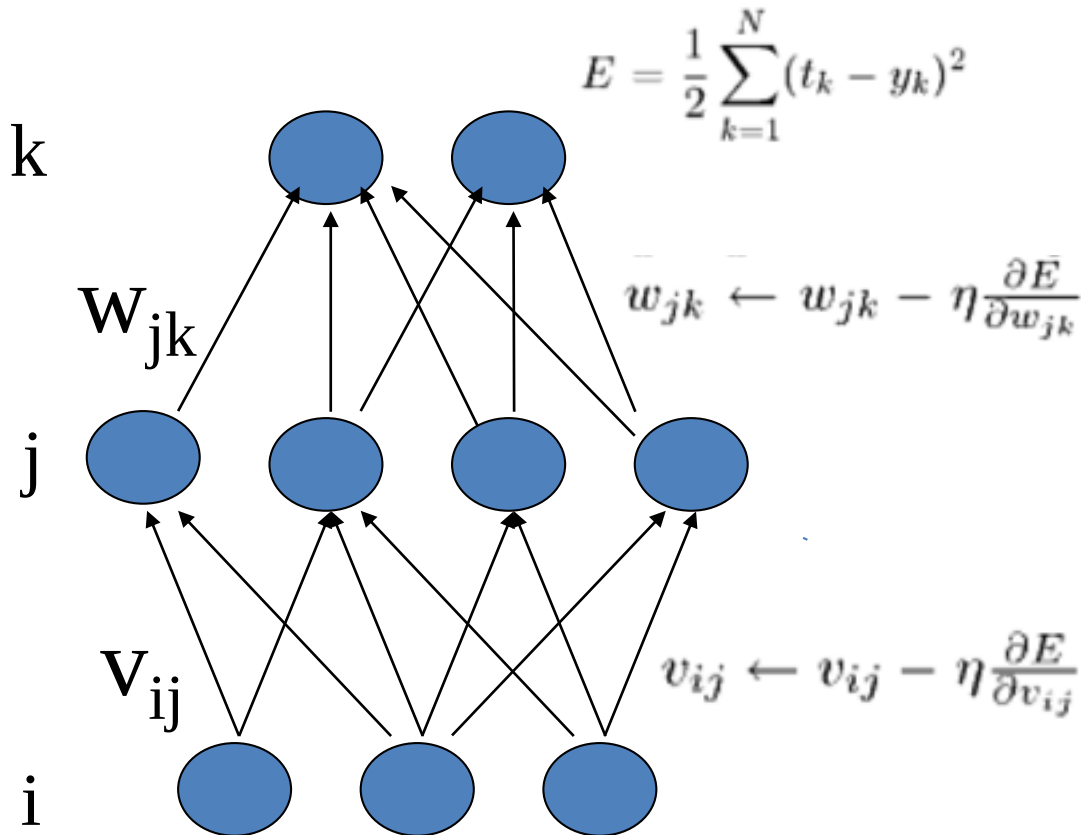
- Multiples capas
- Hacia adelante

- Pesos de Conexión $z_j = \sum_i x_i w_{ij}$



$$a = \frac{1}{1 + e^{-z}}$$

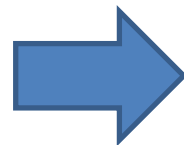
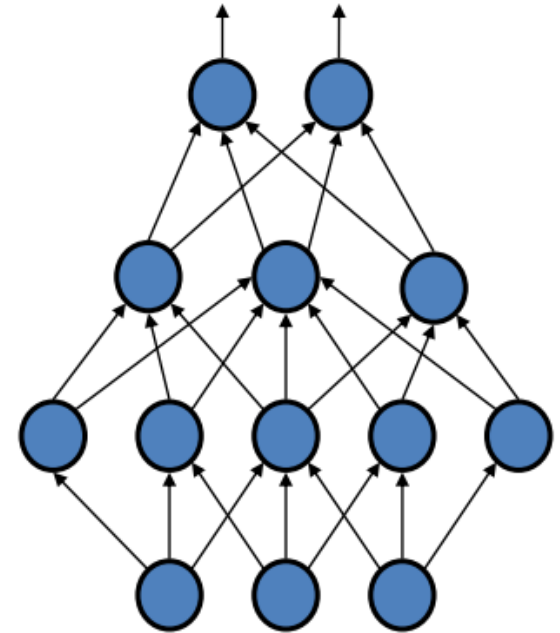
Backpropagation



- Minimiza error de salida
- Ajuste de pesos
 - Descenso de Gradiente
- Pasos
 - Hacia adelante
 - Retropropagación del error
- Para cada ejemplo, Varias corridas

Problemas con Backpropagation

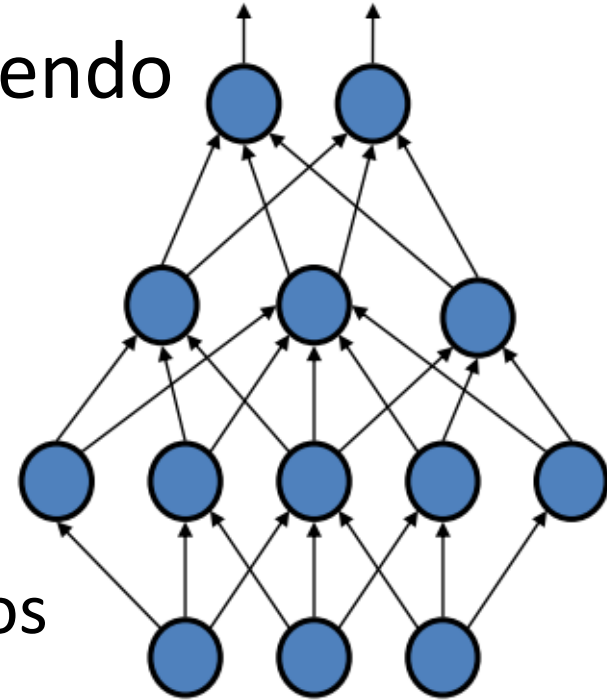
- **Multiples capas escondidas**
- **Cae en optimos locales**
 - Segun donde se inician pesos
- **Converge lento al optimo**
 - Necesita mucho entrenamiento
- **Solo usa datos etiquetados**
 - La mayoría de los datos son no etiquetados



Otro enfoque

Arquitectura Deep

- Múltiples capas escondidas
- Backpropagation, como construyendo bloques
- Motivación
 - aprendizaje jerárquico
 - funciones cada vez más complejas
 - trabajar bien en diferentes dominios
 - Visión, audio, ...



Aprendizaje supervisado



Carros



Motocicletas



Aprendizaje semi-supervisado



Imágenes sin etiquetas (solo carros/motocicletas)



Carro



Motocicleta



Aprendizaje autodidacta



Imágenes sin etiquetas (p.e., tomadas de internet)



Carro

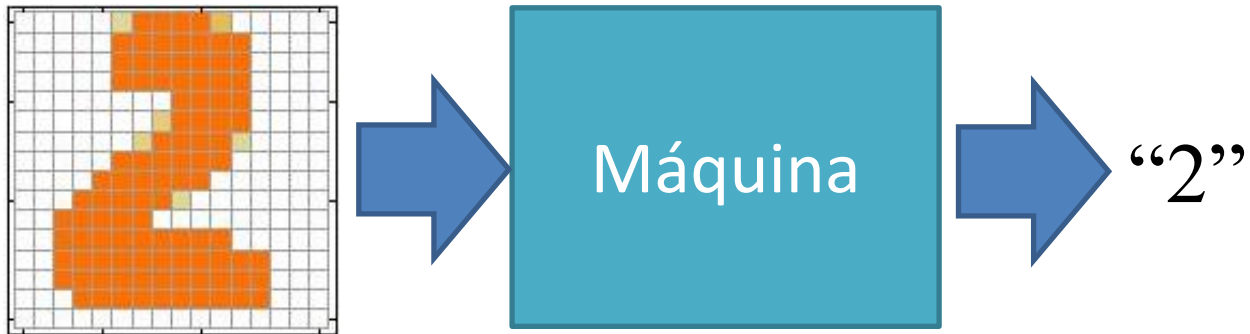


Motocicleta



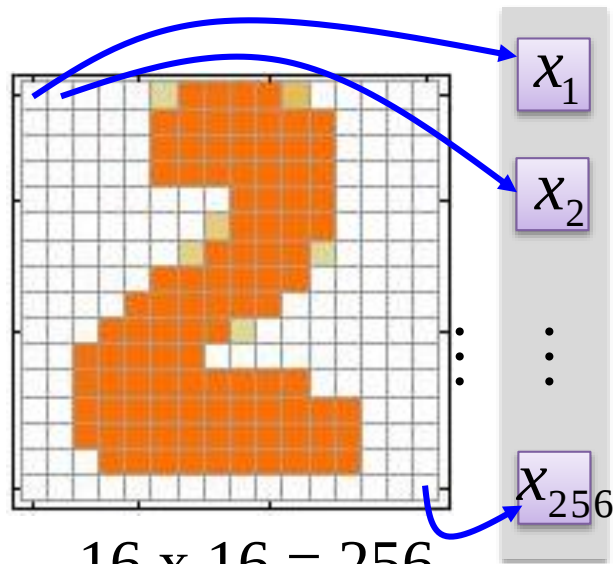
Ejemplo

- Reconocimiento de dígitos escritos



Reconocimiento de dígitos escritos

Entrada



$16 \times 16 = 256$

Ink $\rightarrow 1$

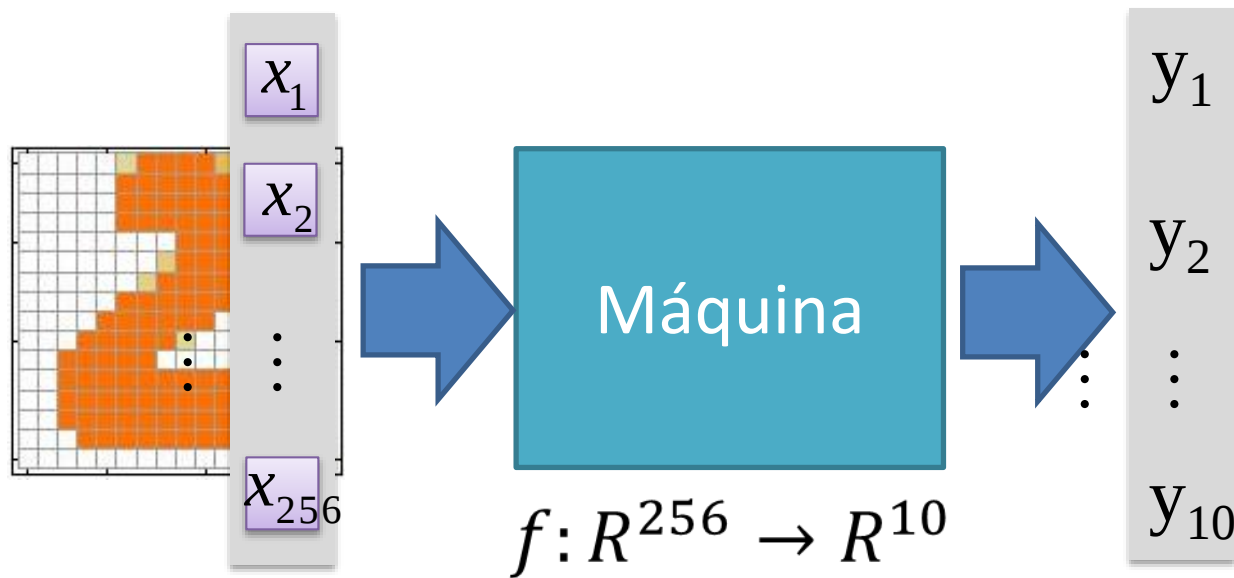
No ink $\rightarrow 0$

Salida



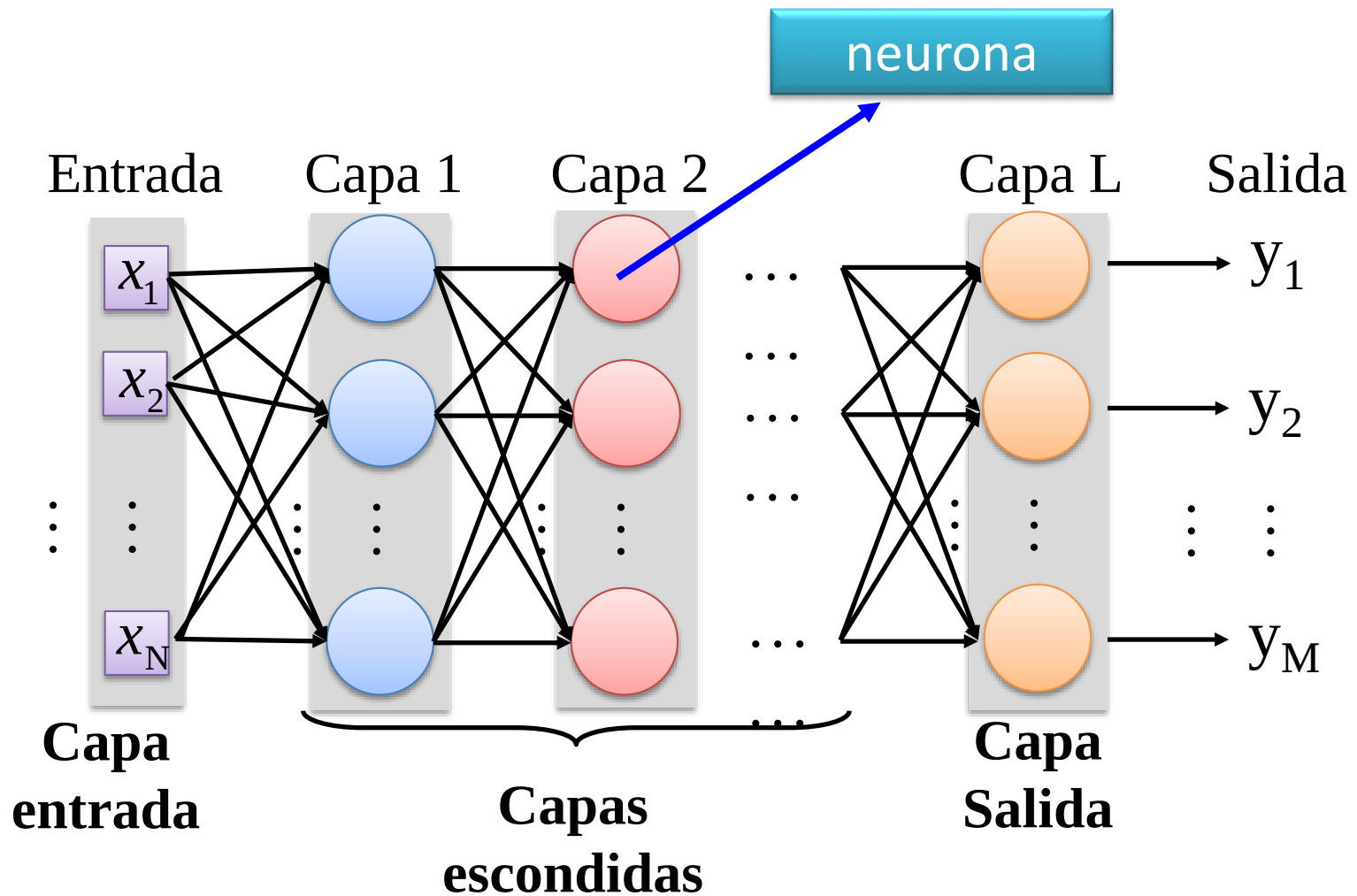
Cada dimensión representa la confianza de un dígito.

Reconocimiento de dígitos escritos



En aprendizaje profundo, la
función f es una RNA

RNA



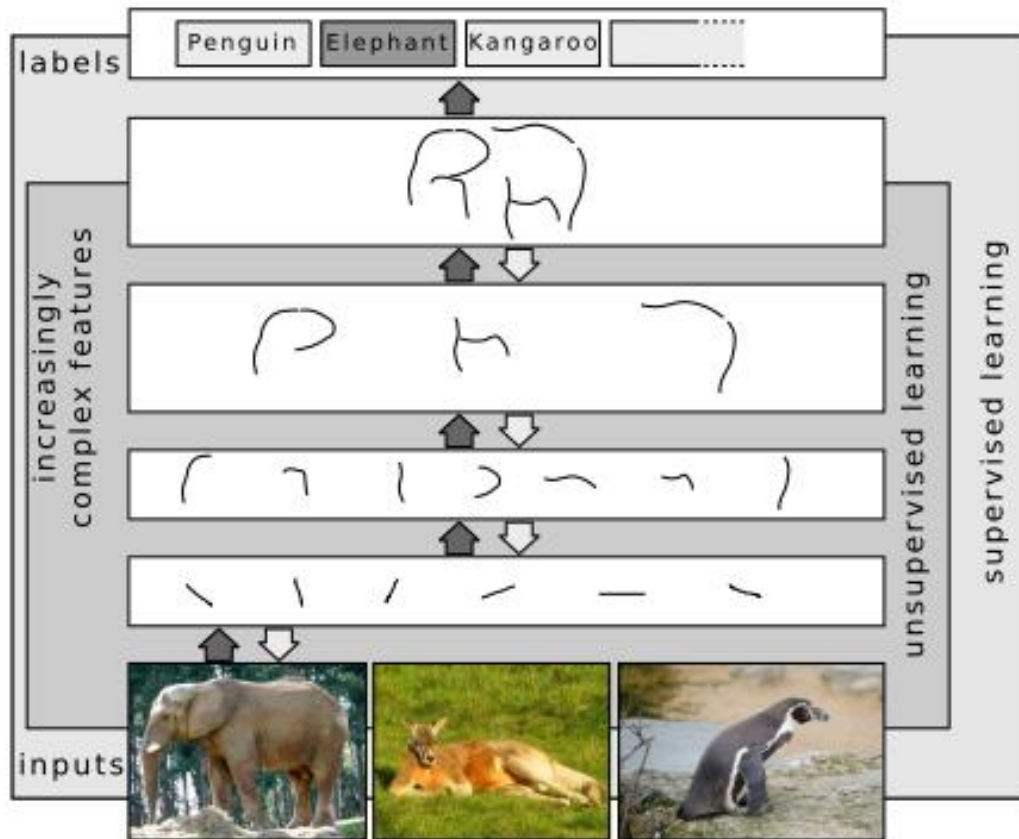
Deep significa muchas capas ocultas

¿Más profunda es mejor?

Capa x tamaños	Tasa error palabra (%)
1 X 2k	24.2
2 X 2k	20.4
3 X 2k	18.4
4 X 2k	17.8
5 X 2k	17.2
7 X 2k	17.1

No sorprende, más parámetros, un mejor rendimiento

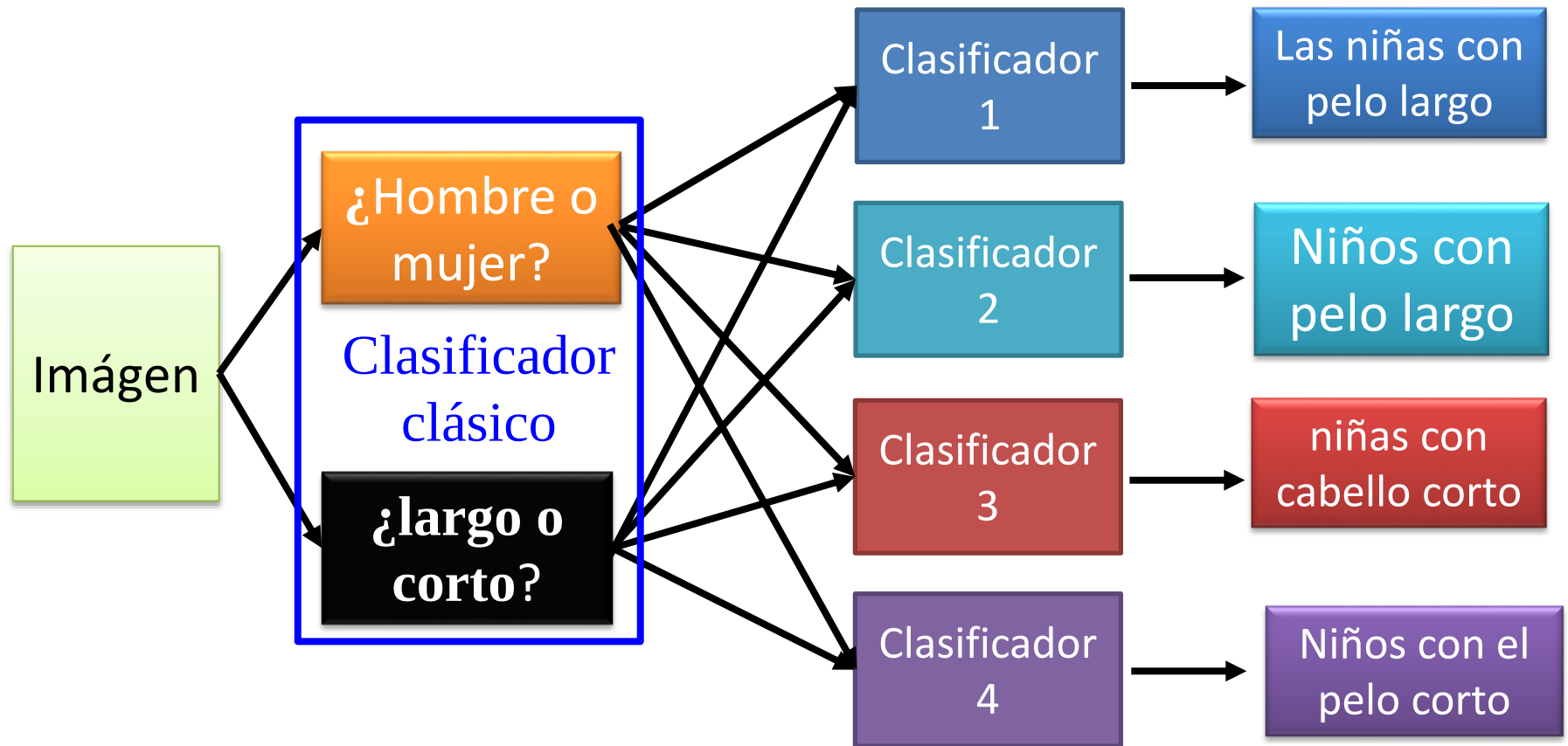
Aprendizaje jerárquica



- progresión natural desde el bajo nivel a la estructura de alto nivel, como se ve en la complejidad natural
- Más fácil de controlar lo que se está aprendiendo y guiar la máquina a mejores subespacios

¿Por qué Deep?

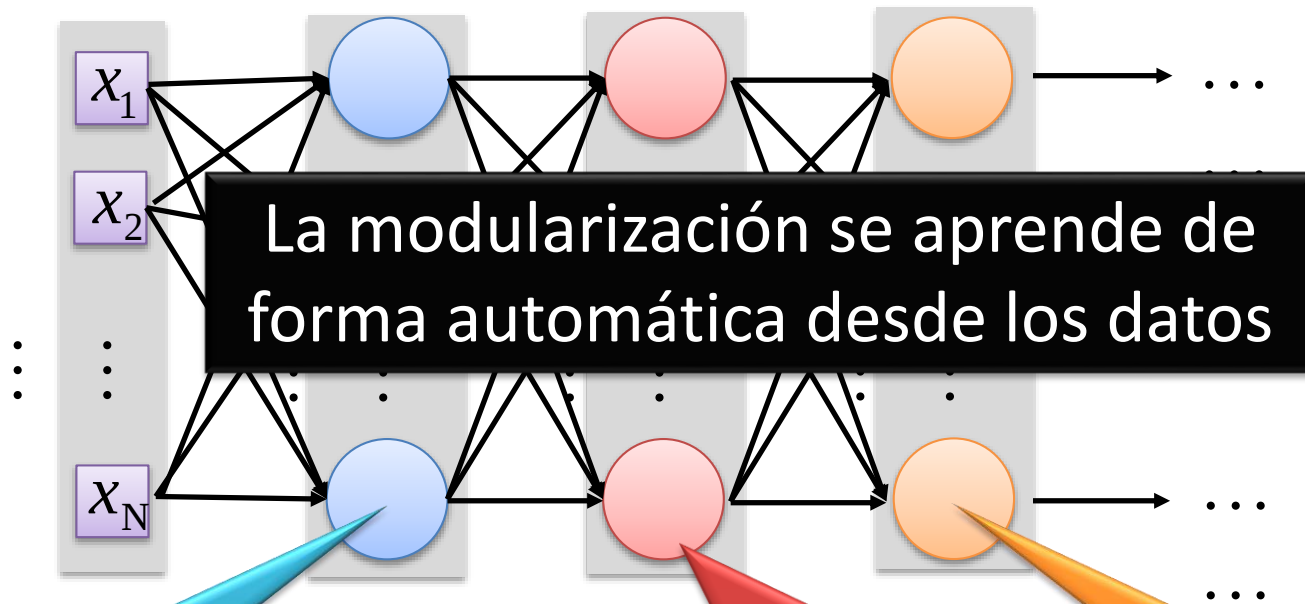
- Modularidad



¿Por qué?

trabaja con conjuntos
pequeños de datos

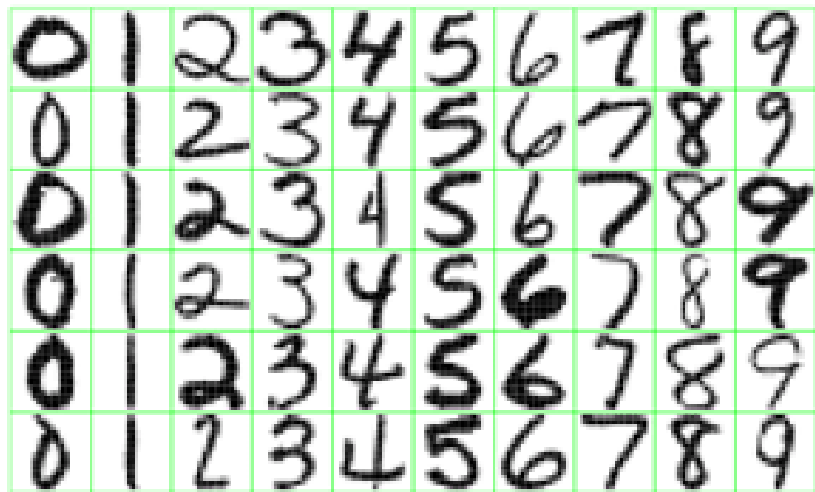
- Modularidad



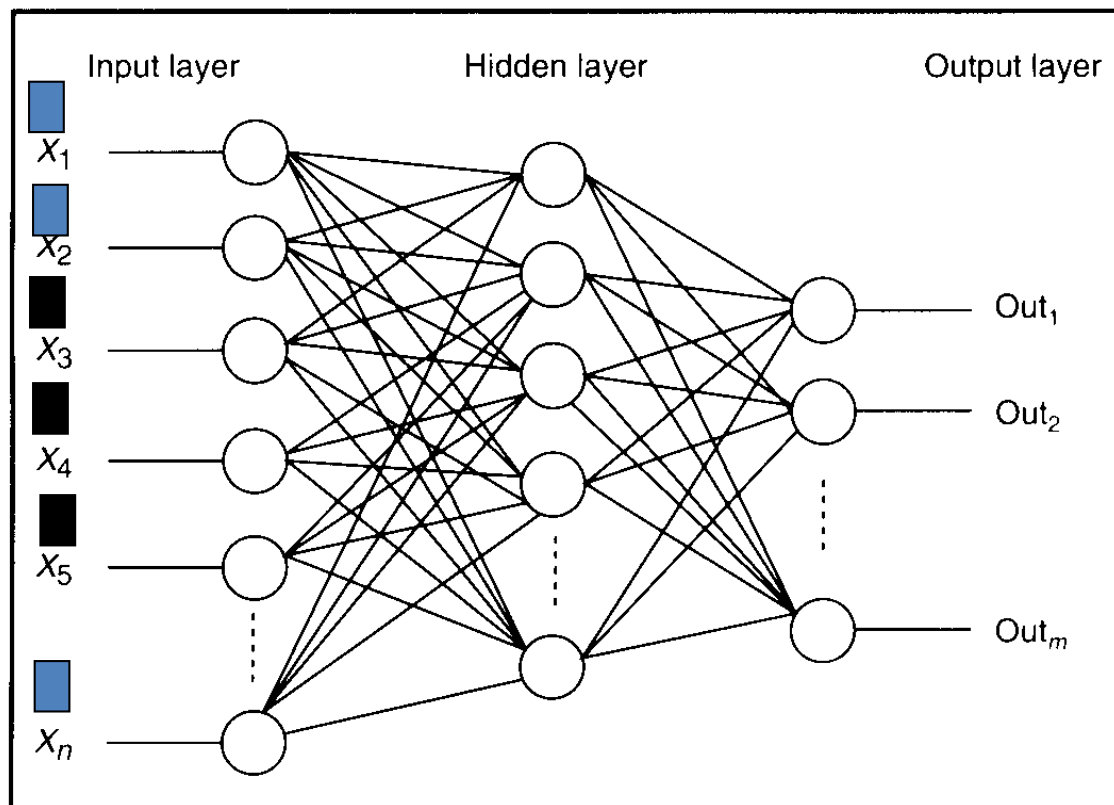
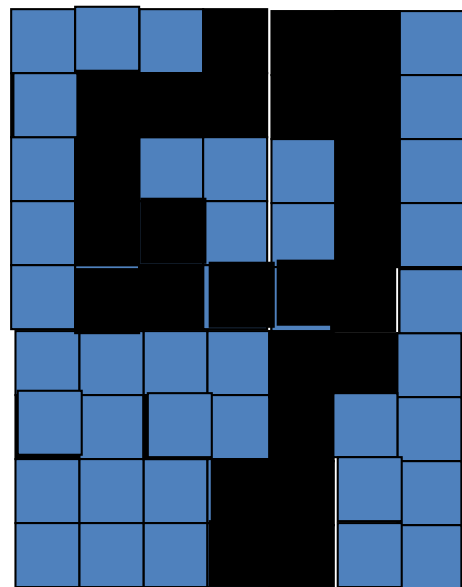
Los clasificadores
más básicas

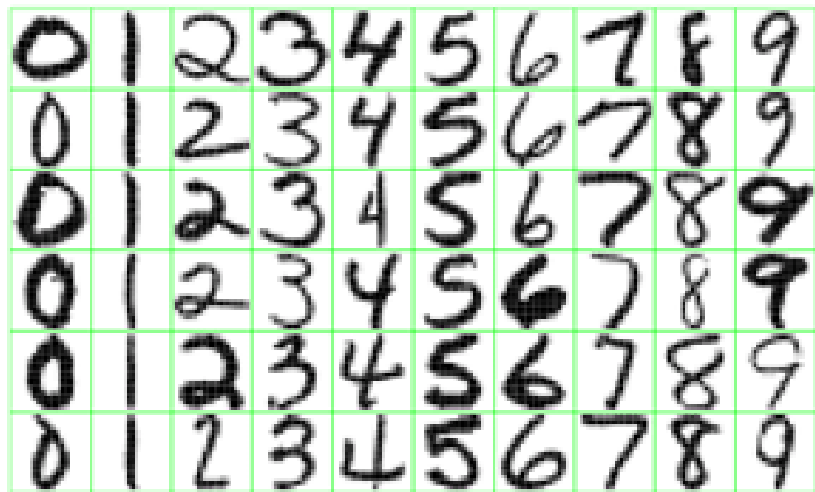
Usa primera capa como
módulo para construir
clasificadores

Usa segunda capa
como módulo

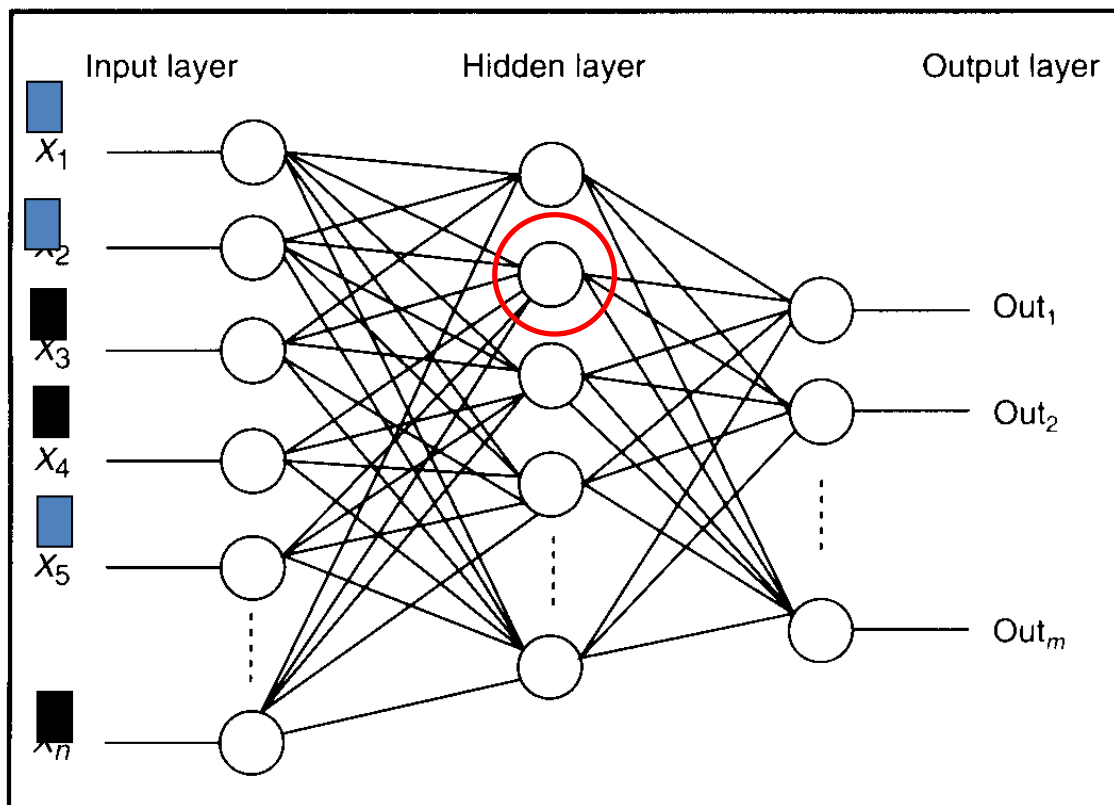
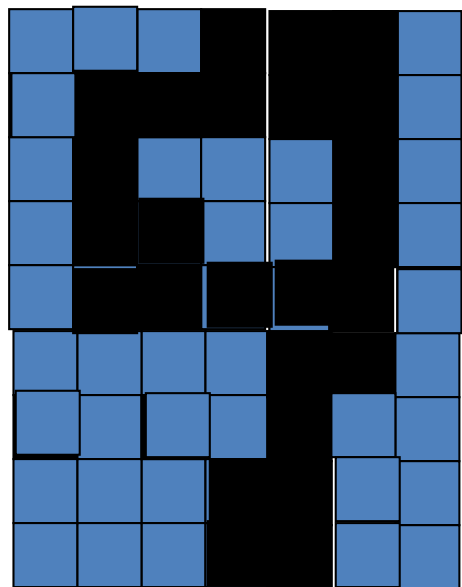


detectores de características

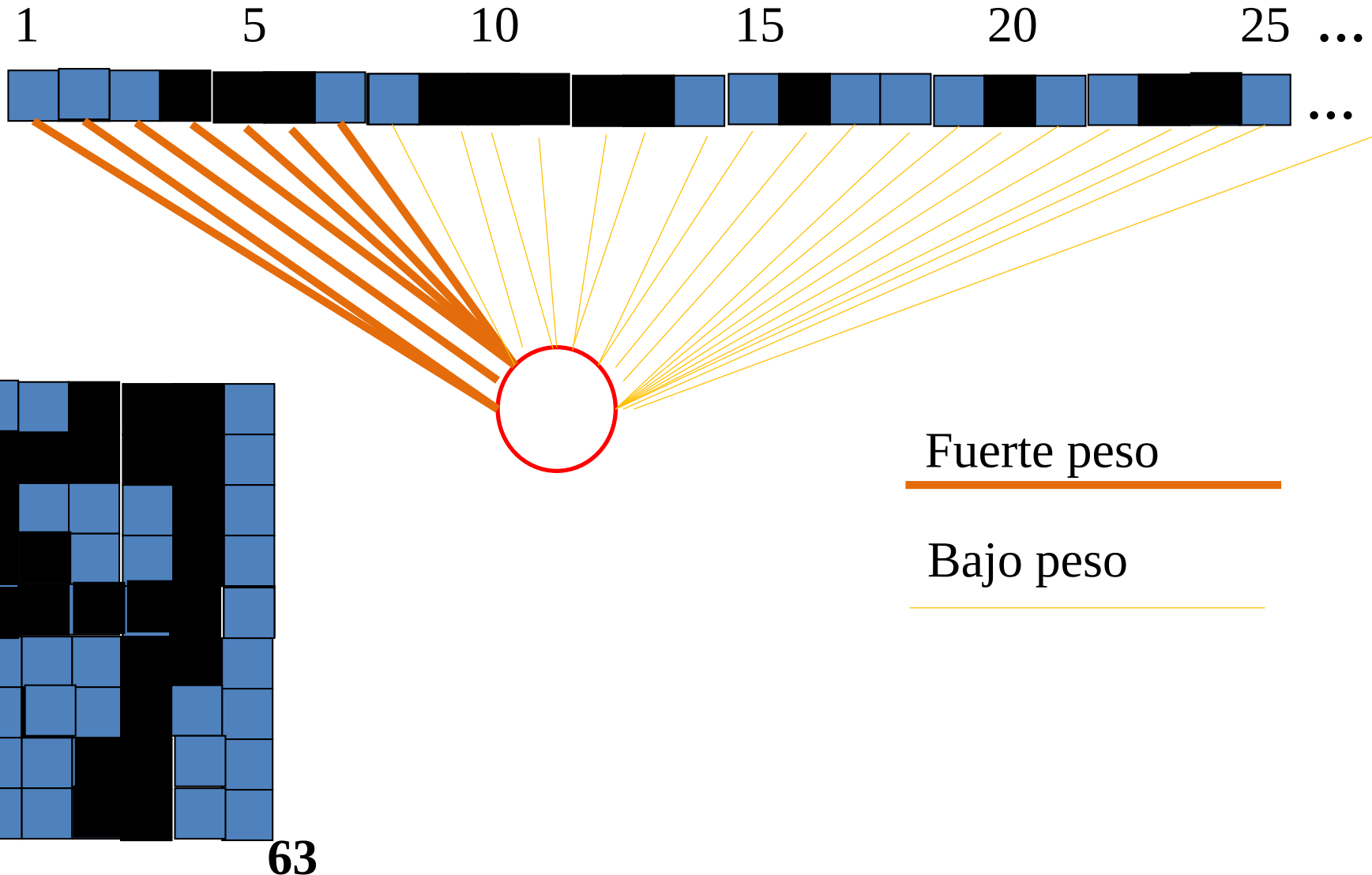




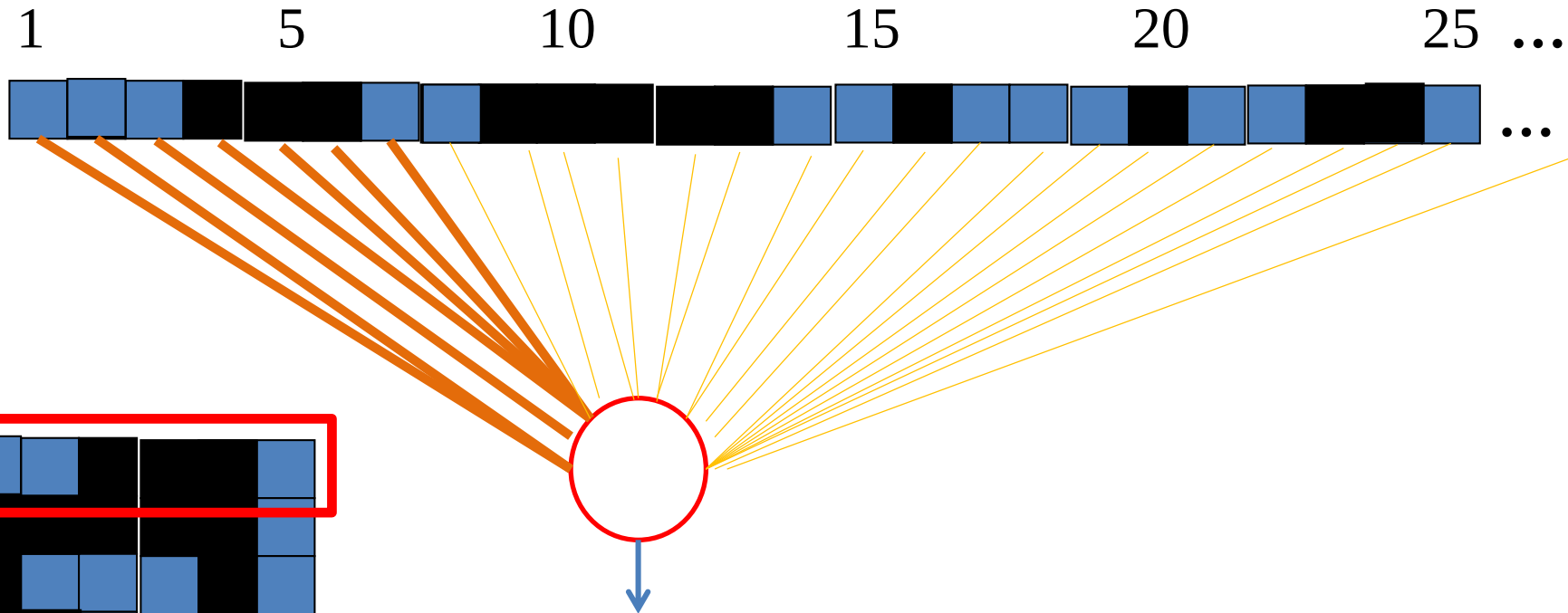
¿Qué hace?



Redes Capas ocultas son detectores auto-organizadas de características

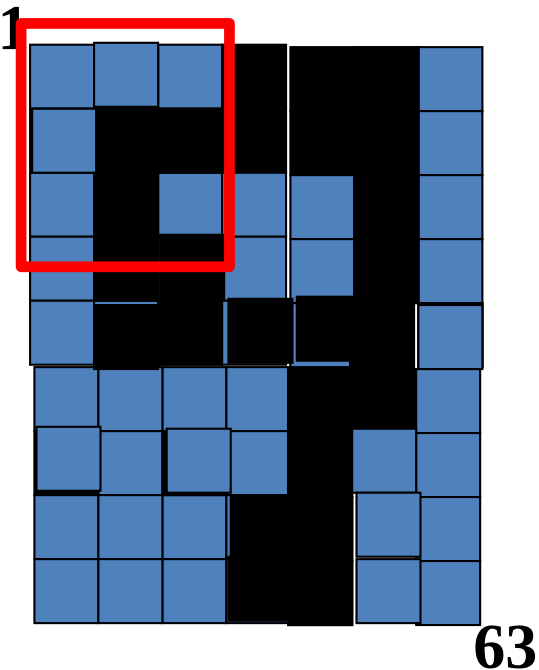
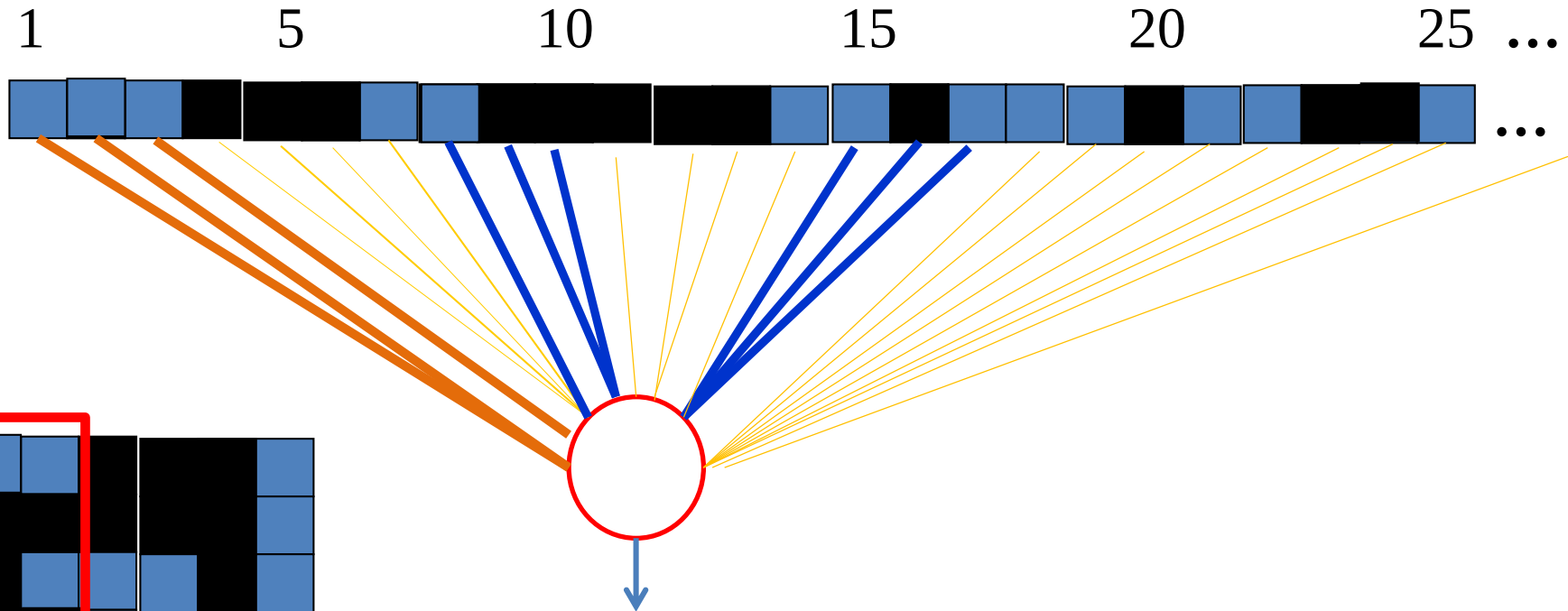


¿Qué detecta?



Enviaré la señal fuerte para una línea horizontal en la fila superior, haciendo caso omiso de todos los demás sitios

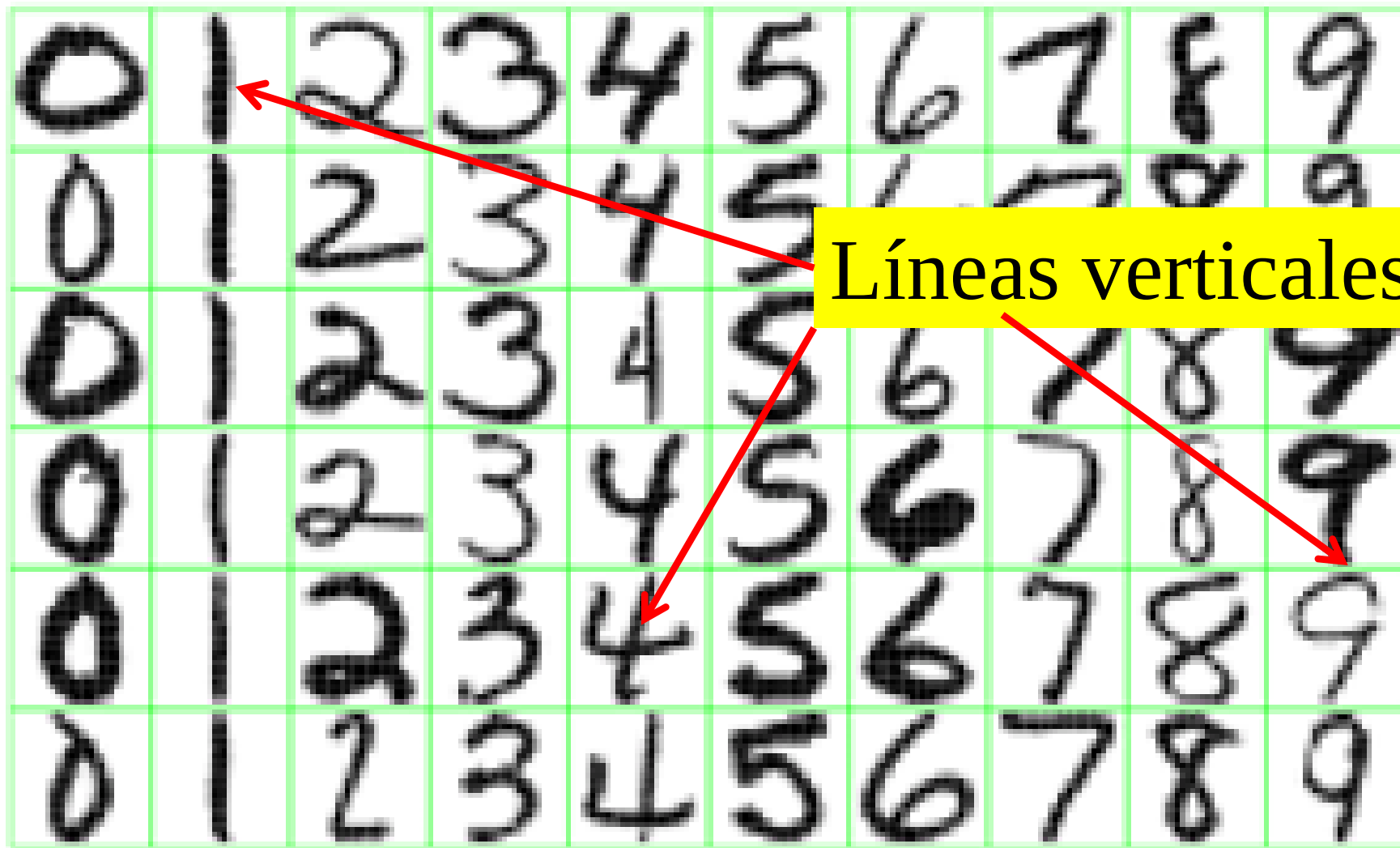
¿Qué detecta?



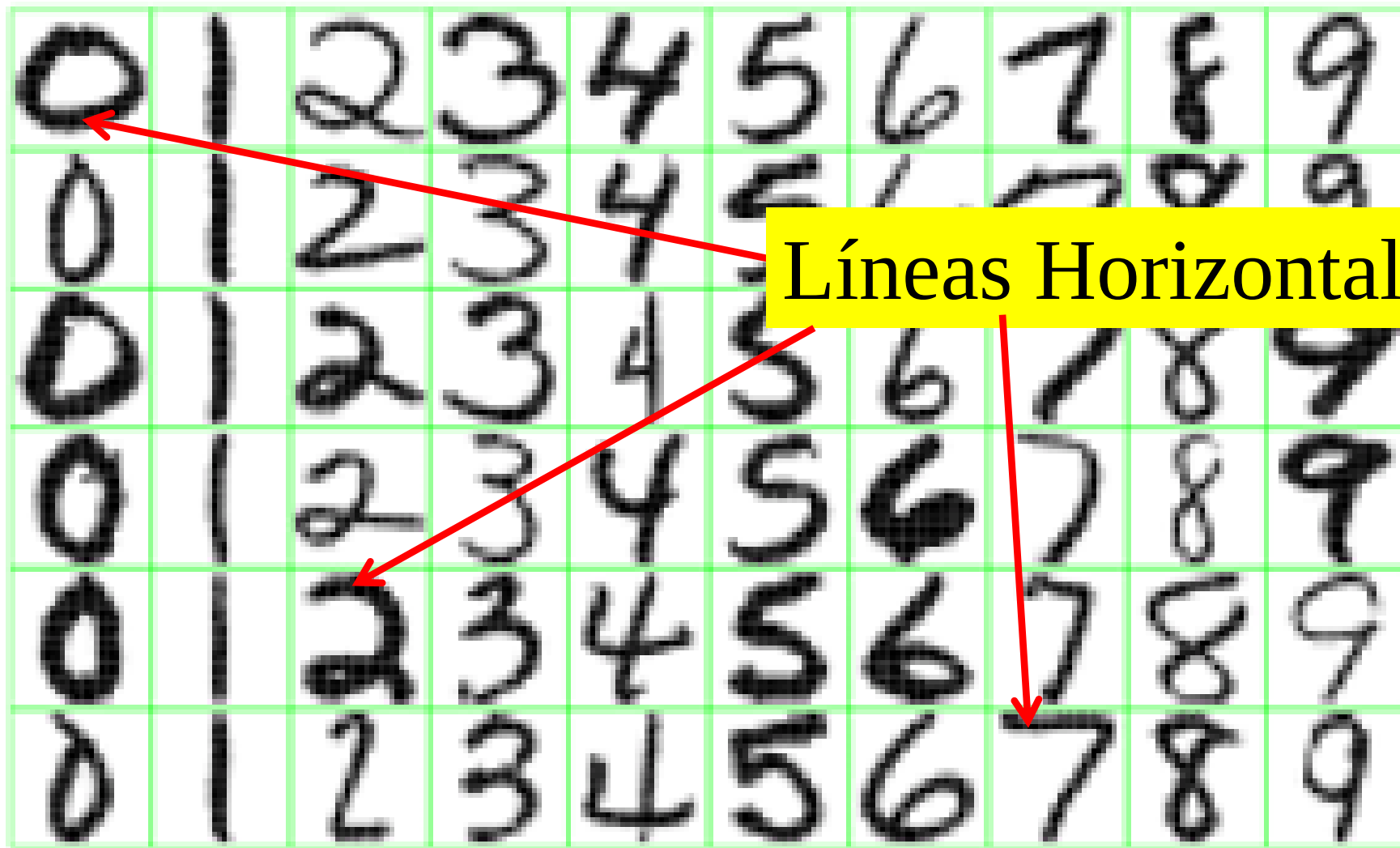
señal fuerte para un área oscura en la esquina superior izquierda

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

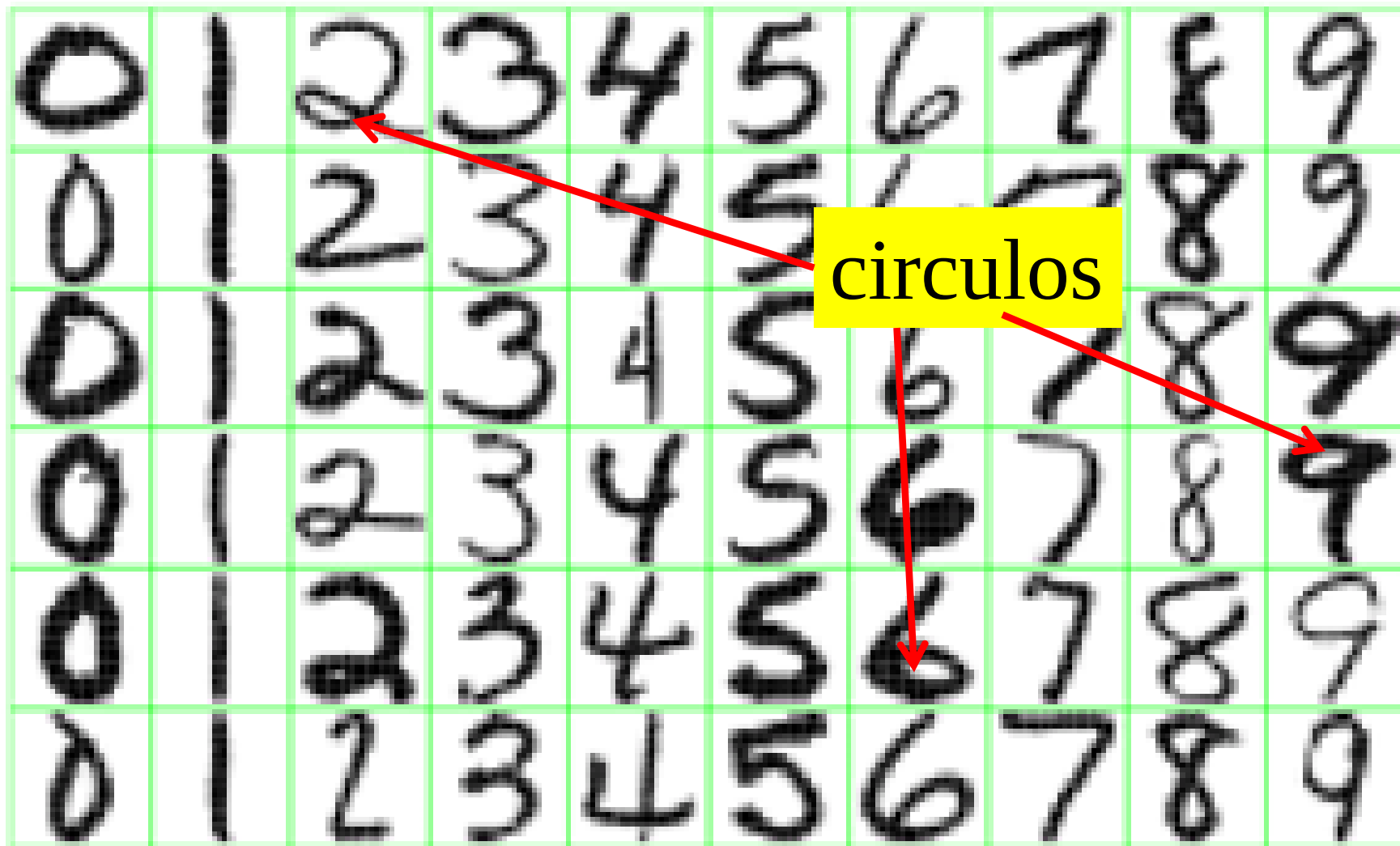
¿Qué características podría esperar una buena NN aprender, entrenados con datos como este?



Líneas verticales

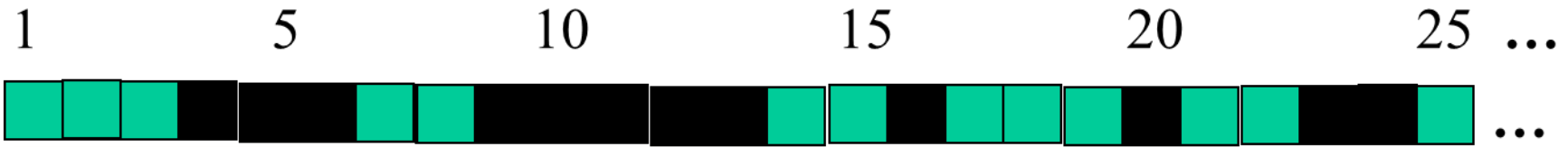


Líneas Horizontales

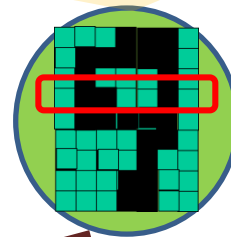
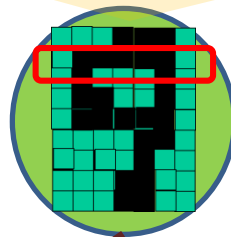
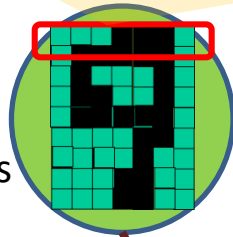


circuitos

capas sucesivas pueden aprender características de nivel superior ...

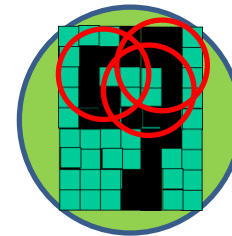
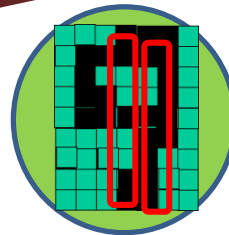
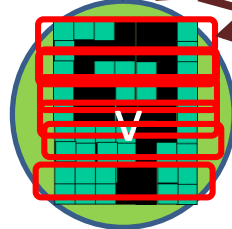


detectar líneas en
posiciones específicas



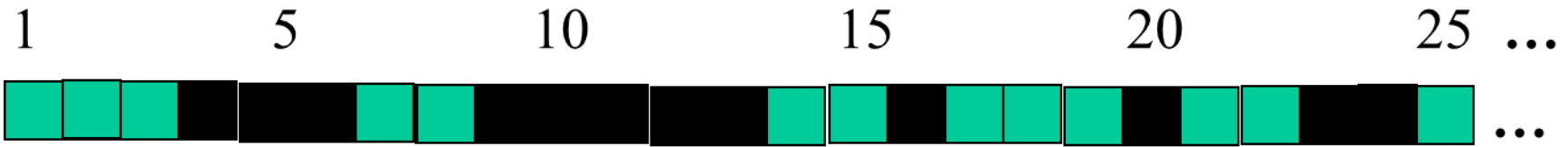
etc ...

Los detectores de nivel
superior
(línea horizontal,
"Línea vertical del lado
derecho"
"Bucle superior", etc., ...

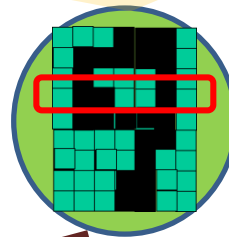
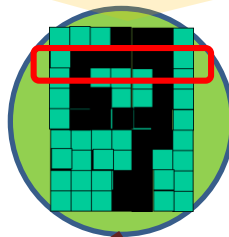
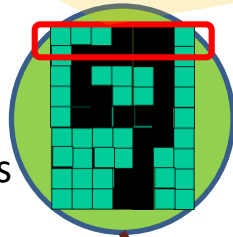


etc ...

capas sucesivas pueden aprender características de nivel superior ...

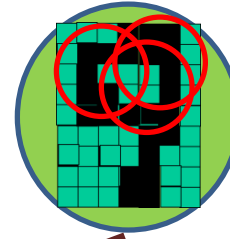
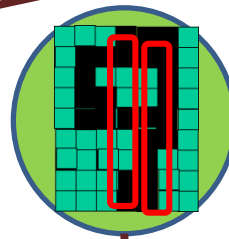
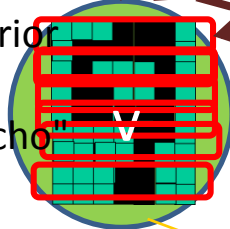


detectar líneas en
posiciones específicas



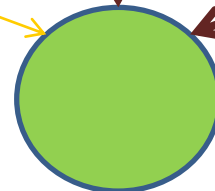
etc ...

Los detectores de nivel superior
(línea horizontal,
"Línea vertical del lado derecho"
"Bucle superior", etc., ...



etc ...

¿Qué detecta esta unidad?

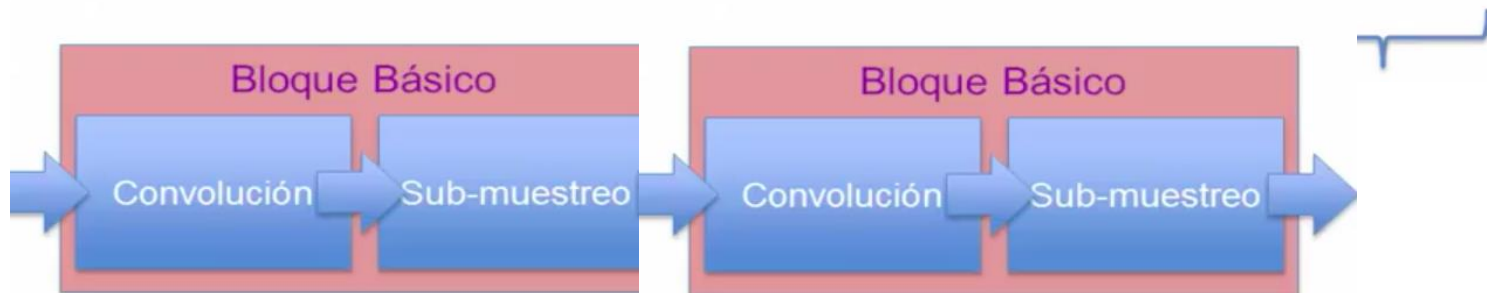
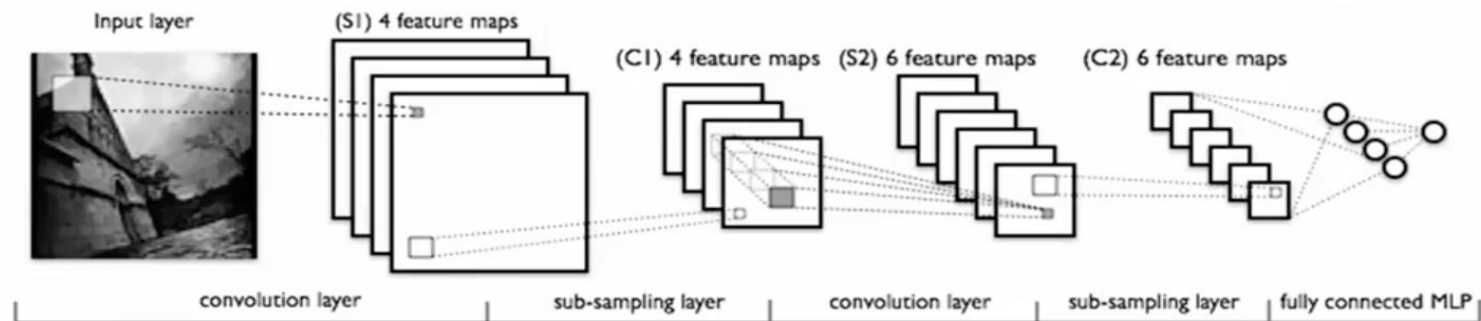


Algunas Deep Learning NN

- Convolutional neural networks
- Recursive neural networks
- Deep belief networks
- Deep Boltzmann machines
- Deep Q-networks

CNN

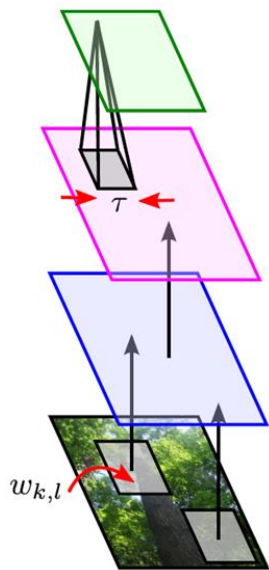
Yann LeCun, introduce el concepto de Redes Neuronales de Convolución o Neural Network (CNN del Inglés) en 1998.



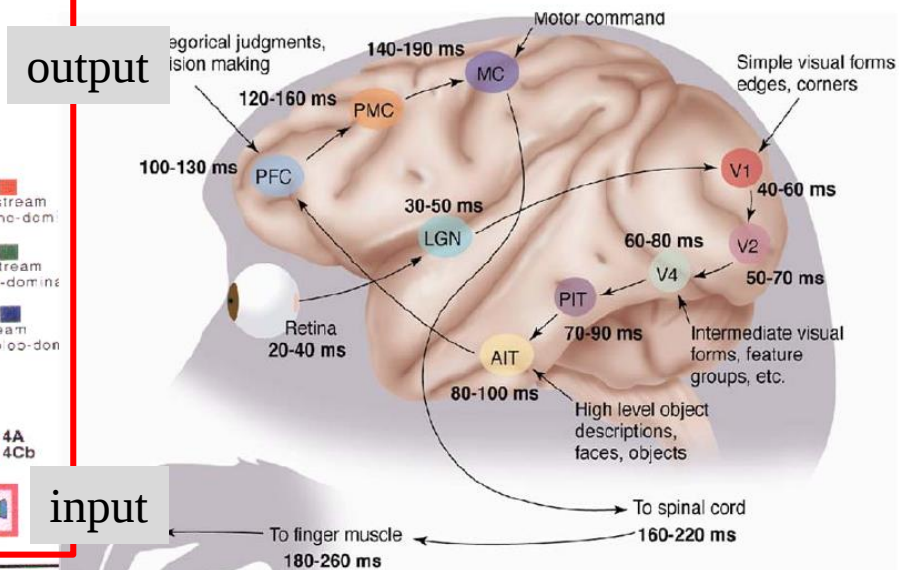
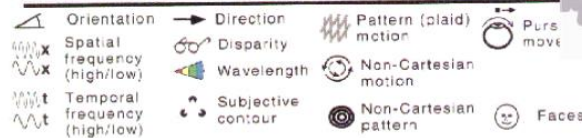
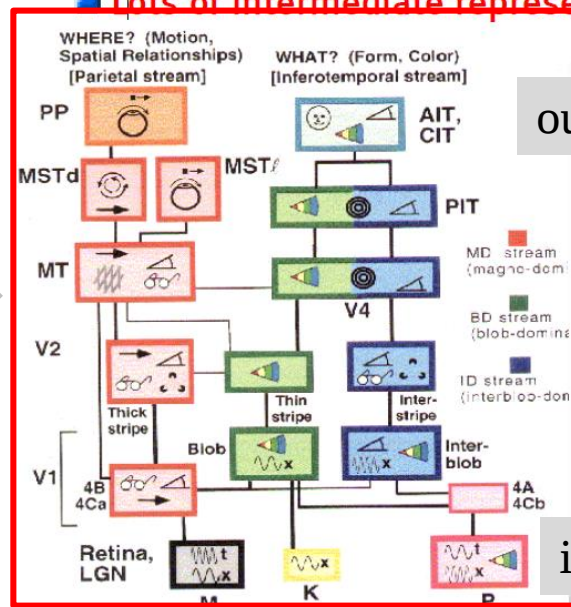
- Hiperparámetros (decisiones):
 - Tamaño filtros ($n \times m$).
 - Número de filtros (K).
 - Tipo de sub-muestreo.
 - Tamaño celda de sub-muestreo.
- Hiperparámetros (decisiones):
 - Tamaño filtros ($n \times m$).
 - Número de filtros (K).
 - Tipo de sub-muestreo.
 - Tamaño celda de sub-muestreo.

La corteza mamaria inspira CNN

Convolutional Neural Net



- The ventral (recognition) pathway in the visual cortex has multiple stages
- Retina - LGN - V1 - V2 - V4 - PIT - AIT
- Lots of intermediate representations



[picture from Simon Thorpe]

[Gallant & Van Essen]

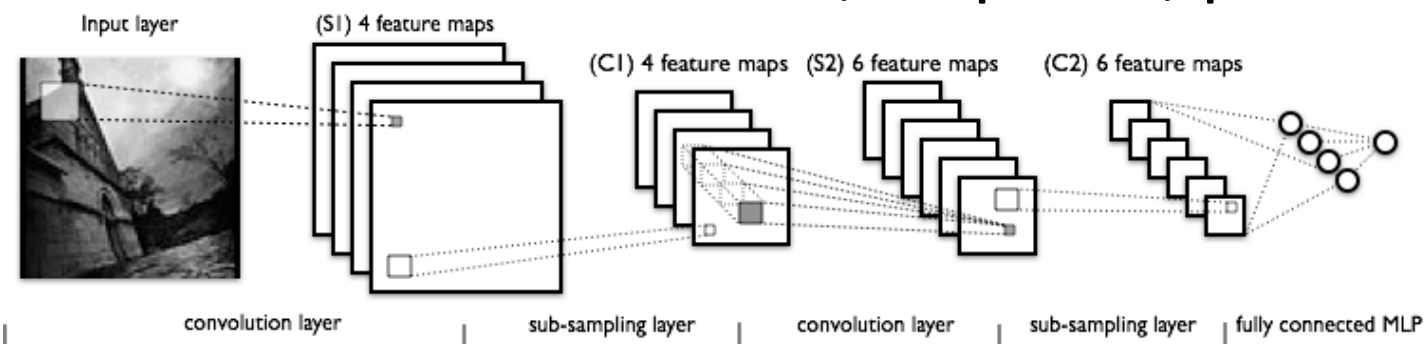
CNN

- Intuición: Red neuronal con estructura de conectividad especializada,
 - Apilamiento de múltiples capas de extractores de características
 - Las capas de bajo nivel extraen de características locales.
 - Las capas de alto nivel extraen los patrones globales de aprendizaje.
- Una CNN es una lista de capas que transforman los datos de entrada en una clase de salida / predicción.
- Tipos de de capas:
 - Capa convolucional
 - Capa no lineal
 - Capa de agrupamiento

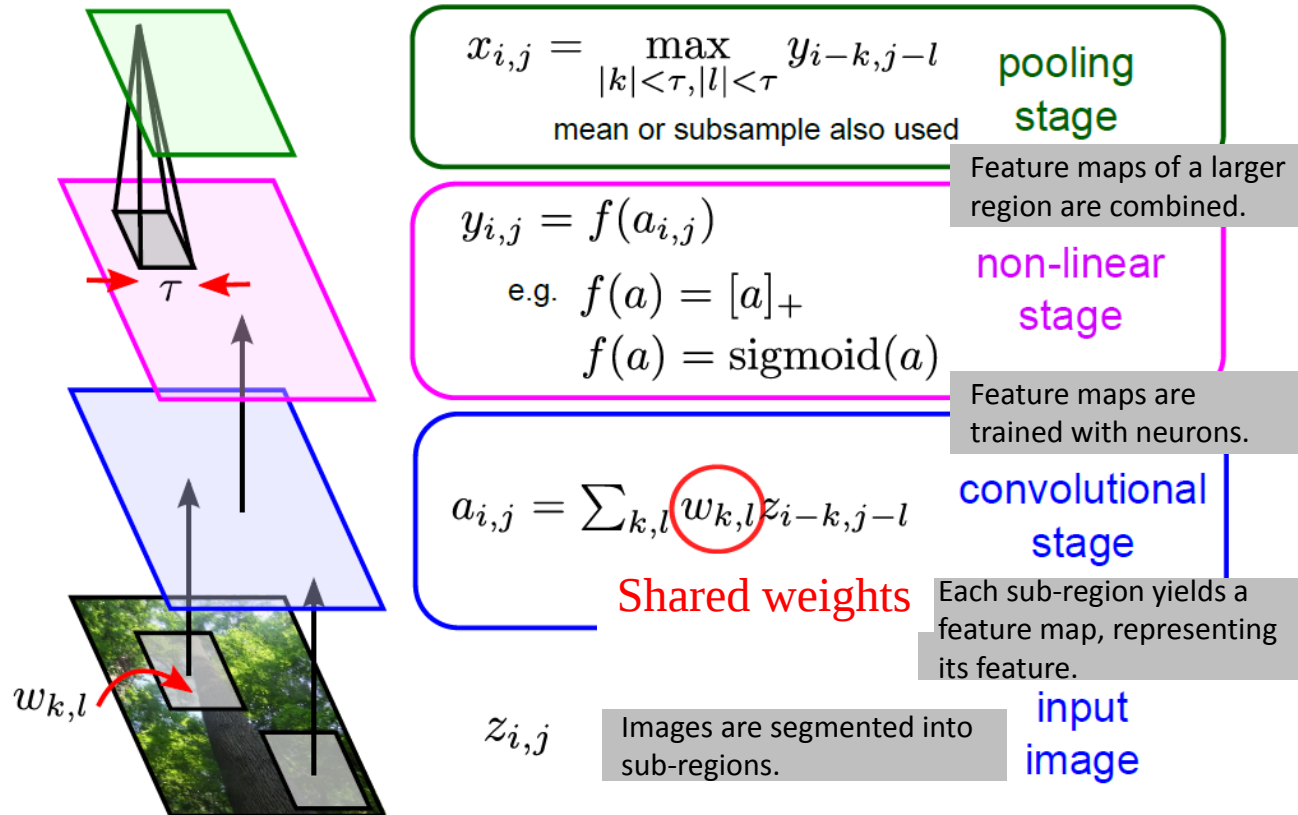
Convolutional Neural Networks

Mientras que los nodos en una estándar RNA obtienen información de todos los nodos de la capa anterior, **en CNN un nodo recibe sólo un pequeño conjunto de características que son espacial o temporalmente cercanas unos de otros**, llamados **campos receptivos** de una capa a la siguiente (por ejemplo, 3x3, 5x5) , haciendo cumplir así la capacidad de manejar la estructura local 2-D.

Puede definir bordes, esquinas, puntos finales,



Bloques de CNN

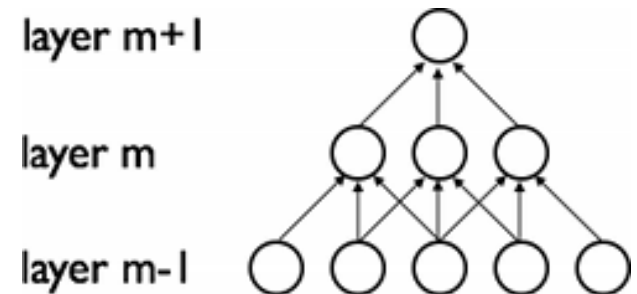


Capa convolucional

- La capa central de CNNs
- La capa convolucional consta de un conjunto de filtros.
 - Cada filtro cubre una parte espacialmente pequeña de los datos de entrada.
- Cada filtro es convolucionado a través de las dimensiones de los datos de entrada, produciendo un mapa de características multidimensional.
- Base: la red aprenderá los filtros que se activan cuando ven algún tipo específico de característica en alguna posición espacial en la entrada.
- Las principales características arquitectónicas de la capa convolucional son la conectividad local y los pesos compartidos.

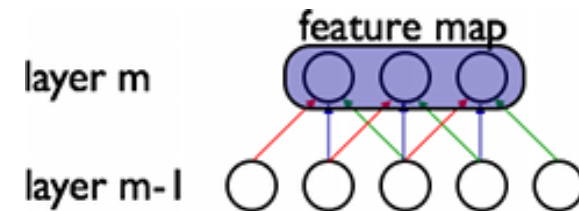
Capa convolucional: conectividad local

- Las neuronas en la capa m están conectadas solamente a 3 neuronas adyacentes en la capa $m-1$.
- Las neuronas en la capa $m + 1$ tienen una conectividad similar con la capa de abajo.
- Cada neurona no responde a variaciones fuera de su campo receptivo con respecto a la entrada.
 - Campo receptivo: colecciones de neuronas pequeñas que procesan porciones de los datos de entrada
- Por lo tanto, la arquitectura asegura que los extractores de función aprendidos produzcan la respuesta más fuerte a un patrón de entrada espacialmente local.



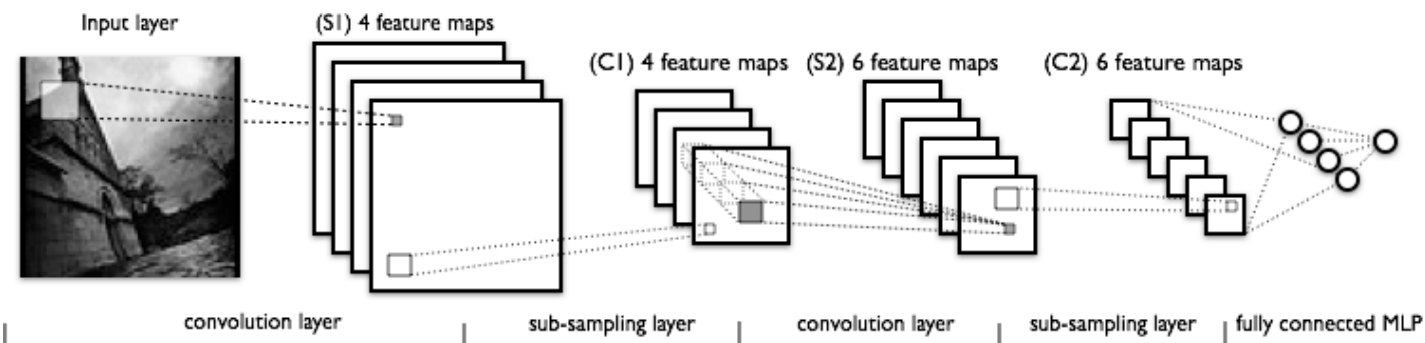
Capa convolucional: pesos compartidos

- Se muestran 3 neuronas ocultas pertenecientes al mismo mapa de características (la capa justo encima de la capa de entrada).
- Pesos del mismo color son compartidos, restringidos para ser idénticos.
- El descenso de gradiente todavía se puede utilizar para aprender tales parámetros compartidos, con sólo un pequeño cambio en el algoritmo original.
- El gradiente de un peso compartido es simplemente la suma de los gradientes de los parámetros que se comparten.
- Replicar las neuronas de esta manera permite que las características sean detectadas independientemente de su posición en la entrada.
- Además, compartir el peso aumenta la eficiencia de aprendizaje al reducir en gran medida el número de parámetros libres que se están aprendiendo.



CNN

- En la capa de submuestreo, cada nodo es calculado para cada **campo receptivo** en la capa anterior
 - Durante el entrenamiento, así, un número relativamente pequeño de parámetros se aprenden,
 - Cada nodo de salida de CNN es sigmoide ($\sum xw + b$)
 - mapas de características múltiples en cada capa
 - Cada mapa de características aprende una característica invariable diferente traducción
- Capa de convolución mantiene el total de características



Capa no lineal

- Base: Aumentar la no linealidad de toda la arquitectura sin afectar los campos receptivos de la capa de convolución

Una capa de neuronas que aplica la función de activación no lineal, como,

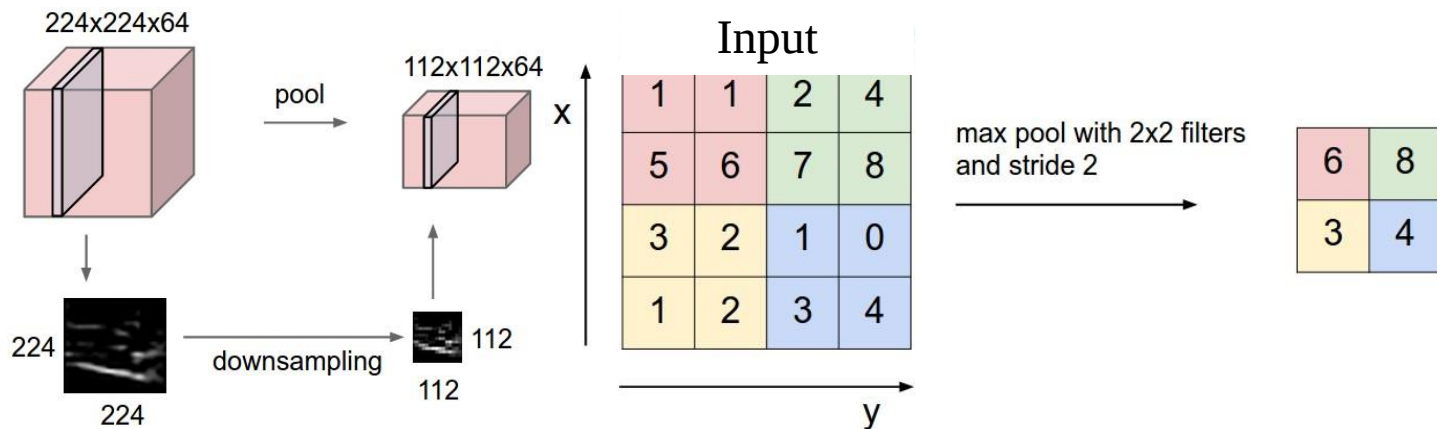
- $f(x) = \max(0, x)$

- $f(x) = \tanh x$

- $f(x) = |\tanh x|$

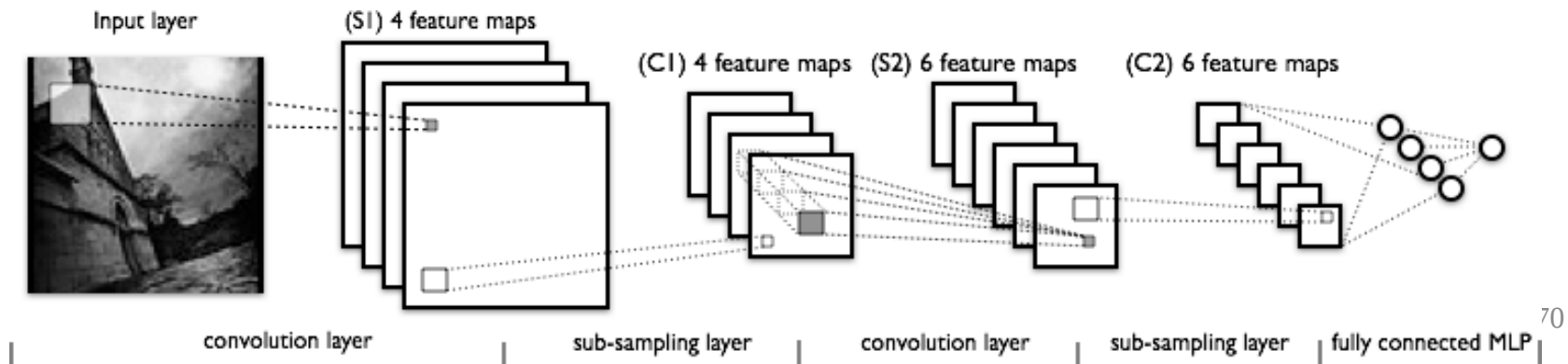
Capa Pooling

- Base: reducir progresivamente el tamaño espacial de la representación para reducir la cantidad de parámetros y de cálculo en la red, y por lo tanto también para controlar el overfitting
- Agrupando particiones de la imagen de entrada en un conjunto de rectángulos que no se superponen y, para cada una de esas subregiones, emite el valor máximo de las características

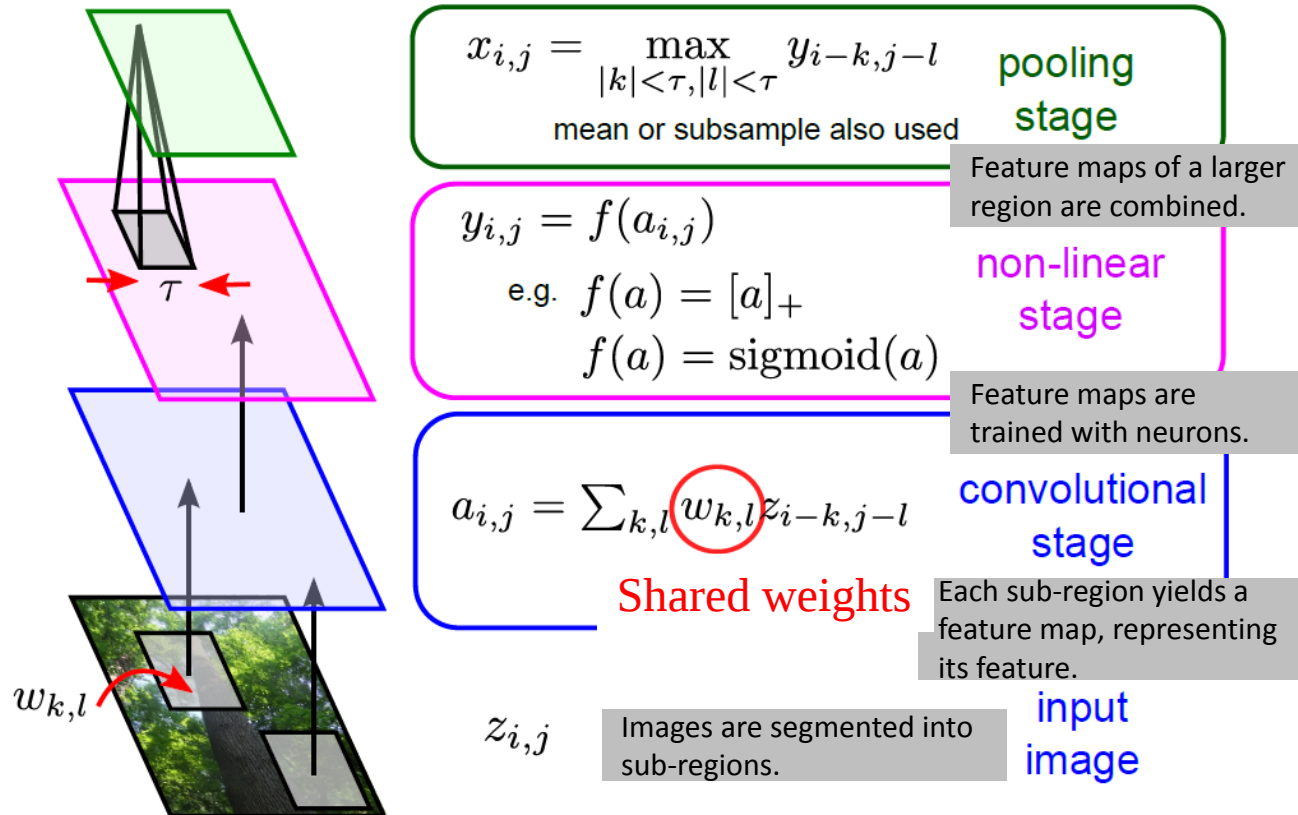


Submuestreo (agrupaciones)

- Capas de convolución y sub-muestreo se intercalan
- Submuestreo agrupa varias características
 - Reduce la resolución espacial, y por lo tanto, disminuye naturalmente importancia de una característica,
 - la agrupación de 2x2 haría compresión 4: 1, 3x3 9: 1, etc.
 - El agrupamiento suaviza los datos

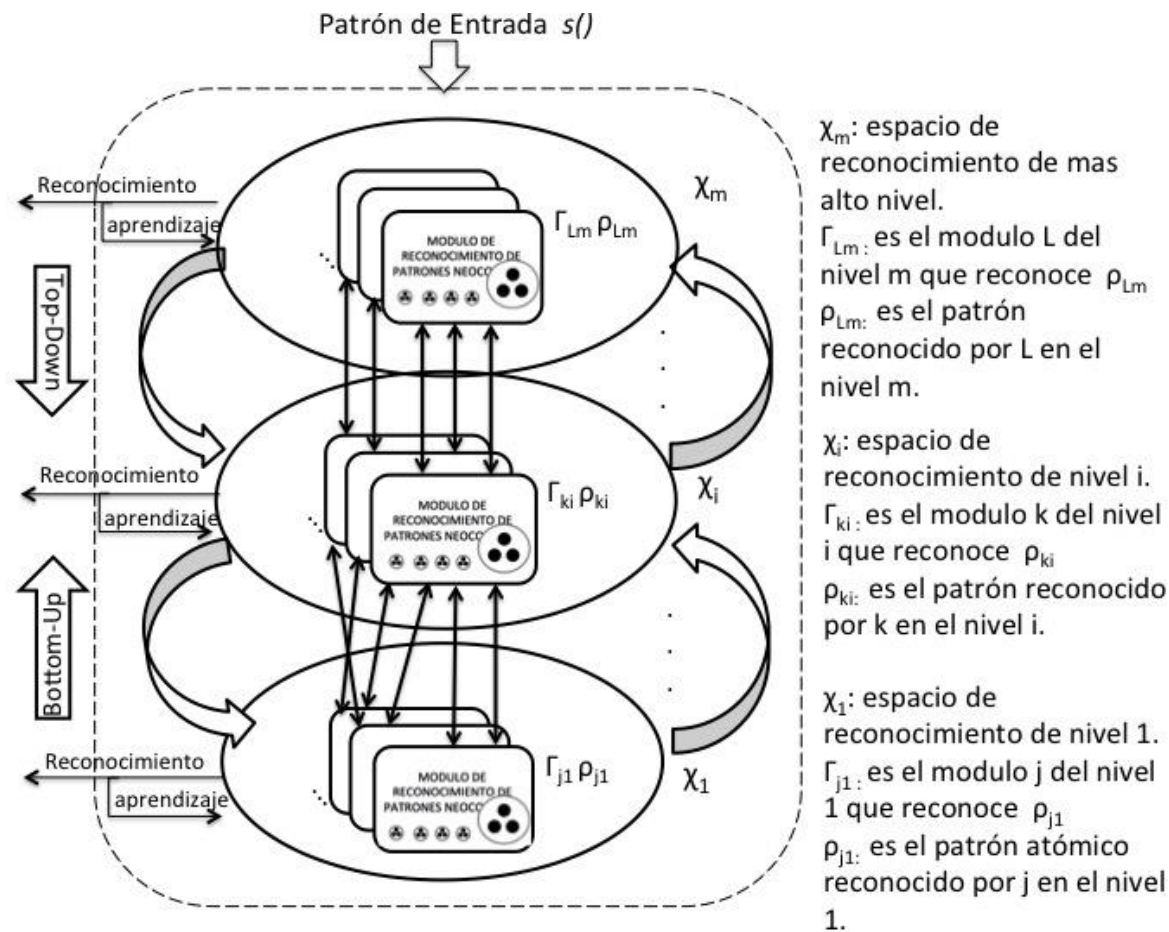


Bloques CNN

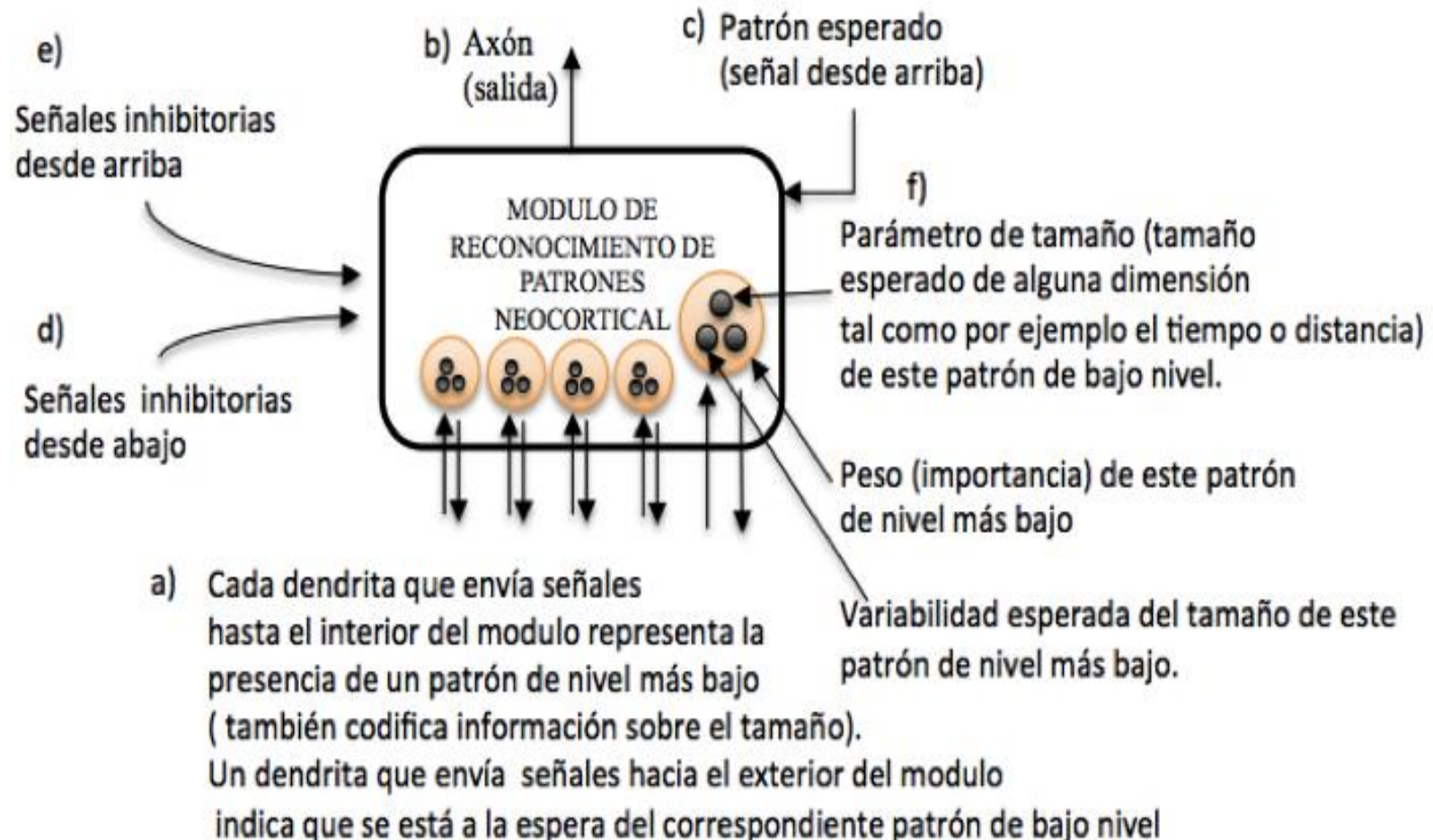


PRTM

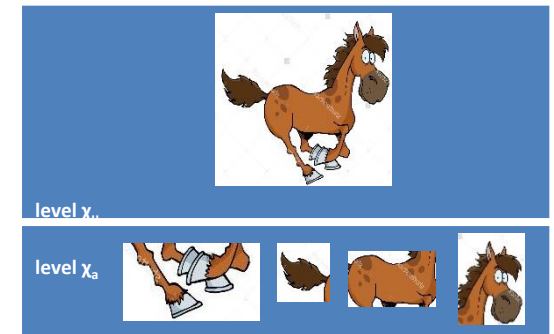
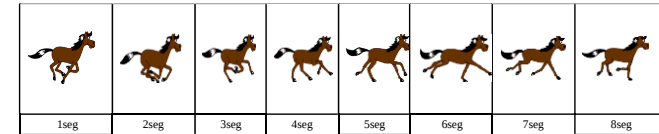
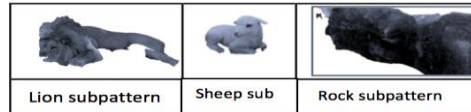
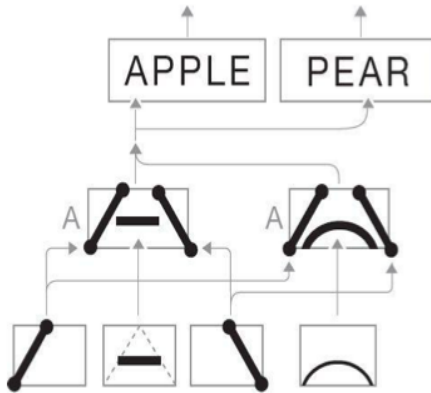
K. Raymond. How to Create a Mind (2013). Modelo de la Mente Basada en el Reconocimiento de Patrones



PRTM



PRTM



DEEP LEARNING

Begin

Input: Object

Disaggregate input until reach atomic component

Aggregation of descriptors

Classification input

End

Aplicaciones Deep Learning

Ingeniería:

- Visión por Computador (por ejemplo, la clasificación de imágenes, segmentación)
- Reconocimiento de voz
- Procesamiento del Lenguaje Natural (por ejemplo, el análisis de opiniones, la traducción)
- Sistemas de recomendación
- bioinformática

Ciencia:

- Biología (por ejemplo, predicción de estructura de la proteína, análisis de datos genómicos)
- Química (por ejemplo, la predicción de las reacciones químicas)
- Física (por ejemplo, la detección de partículas exóticas)

y muchos más

Herramientas de Deep Learning

Nombre	Lenguaje	Sitio Web	Descripción
Pylearn2	Python	http://deeplearning.net/software/pylearn2/	Librería de machine learning en Theano
Theano	Python	http://deeplearning.net/software/theano/	Librería en python de deep learning
Caffe	C++	http://caffe.berkeleyvision.org/	Framework de deep learning de Berkeley
Torch	Lua	http://torch.ch/	Open source Framework de machine learning
Overfeat	Lua	http://cilvr.nyu.edu/doku.php?id=code:st_art	Modelo de CNN para imagenes
Deeplearning4j	Java	http://deeplearning4j.org/	Librería commercial de deep learning
Word2vec	C	https://code.google.com/p/word2vec/	Framework para Word embedding
GloVe	C	http://nlp.stanford.edu/projects/glove/	Framework para Word embedding
Doc2vec	C	https://radimrehurek.com/gensim/models/doc2vec.html	PLN
StanfordNLP	Java	http://nlp.stanford.edu/	Librería de deep learning para PLN
TensorFlow	Python	http://www.tensorflow.org	Librería en python de deep learning