



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Redes Neuronales

Fernando Berzal, berzal@acm.org

Redes Neuronales



- Introducción
 - El cerebro humano
 - Neuronas
 - Redes neuronales artificiales
- Historia
- Aplicaciones
 - MNIST
 - ImageNet



Introducción



¿Por qué estudiar redes neuronales?

- Para comprender cómo funciona realmente el cerebro.
- Para diseñar un modelo de cómputo paralelo inspirado en las neuronas y sus sinapsis [conexiones] adaptativas.
- **Para resolver problemas prácticos utilizando algoritmos de aprendizaje inspirados en el cerebro.**

NOTA: Incluso aunque no sepamos realmente cómo funciona el cerebro, los algoritmos de aprendizaje nos serán muy útiles.

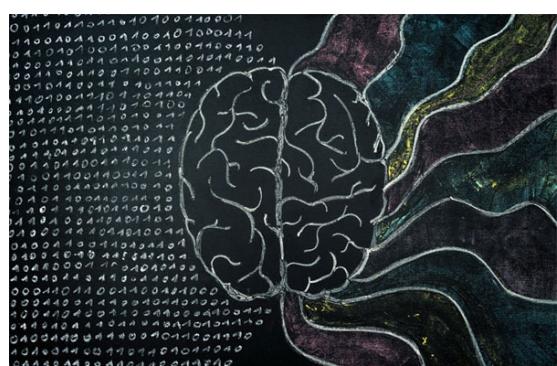
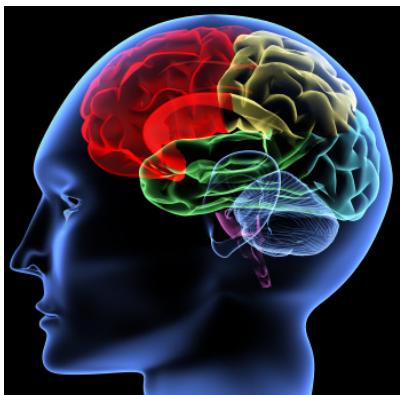


Introducción



El cerebro humano

Inspiración de las redes neuronales artificiales



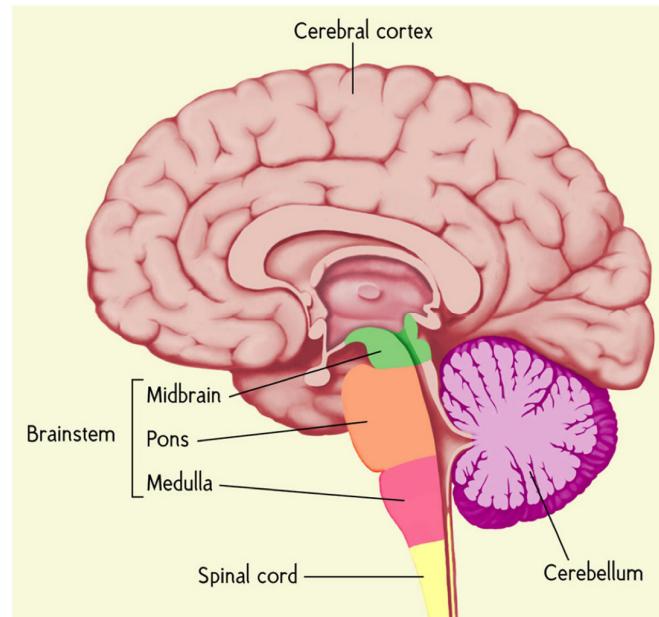
Las RNA intentan modelar la estructura y funcionamiento de algunas partes del sistema nervioso animal.



Introducción



El cerebro humano Anatomía del cerebro



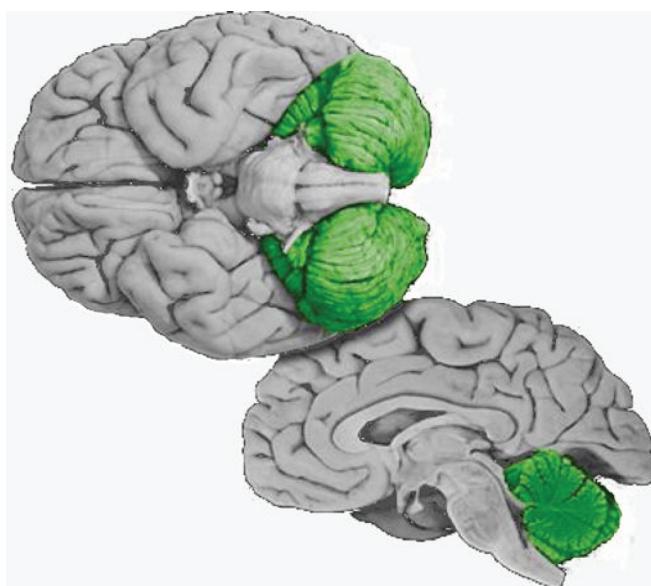
[Psychological Science, 2nd edition, W.W. Norton & Company]



Introducción



El cerebro humano Anatomía del cerebro



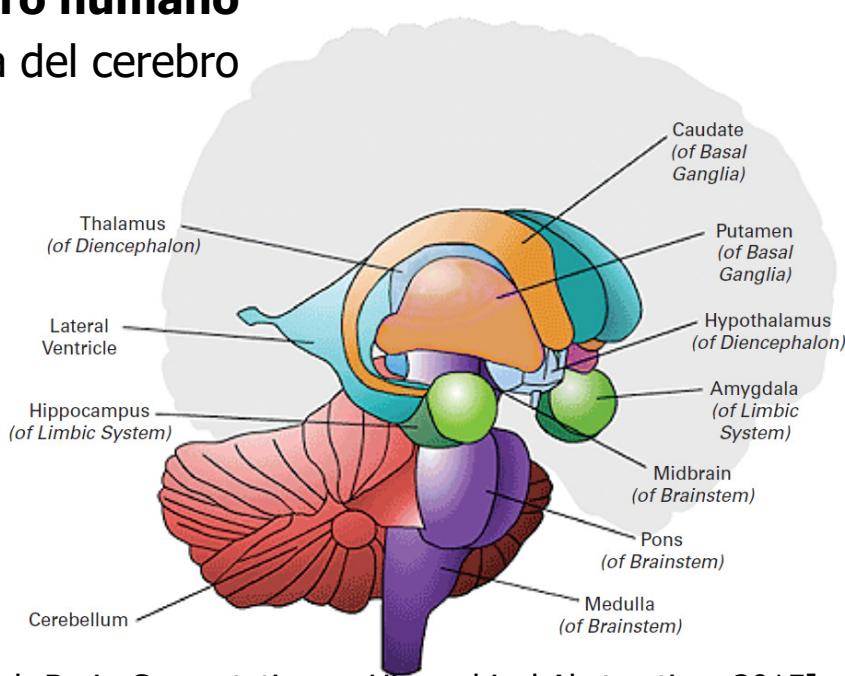
El cerebelo
[Sylvius 4 Online, Sinauer Associates]



Introducción



El cerebro humano Anatomía del cerebro



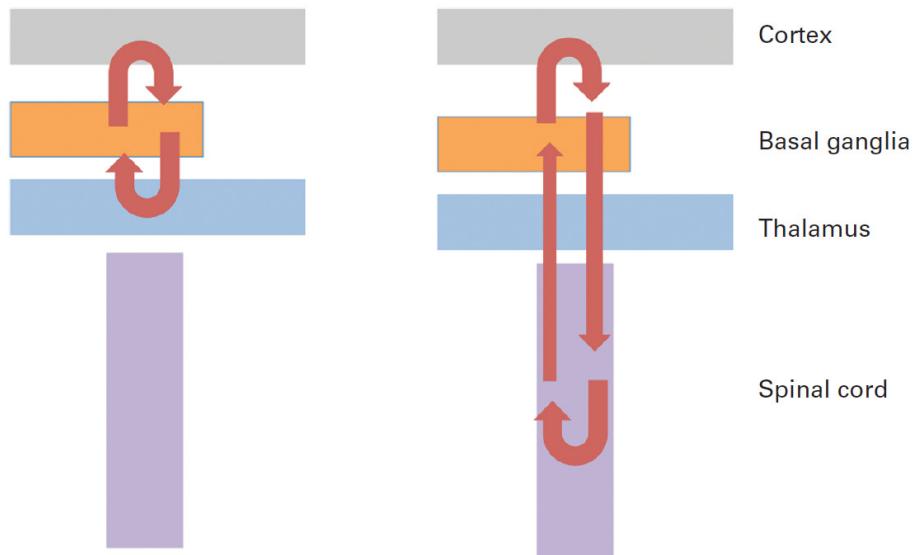
[Dana Ballard: Brain Computation as Hierarchical Abstraction, 2015]



Introducción



El cerebro humano El ciclo de cálculo del cerebro



[Dana Ballard: Brain Computation as Hierarchical Abstraction, 2015]

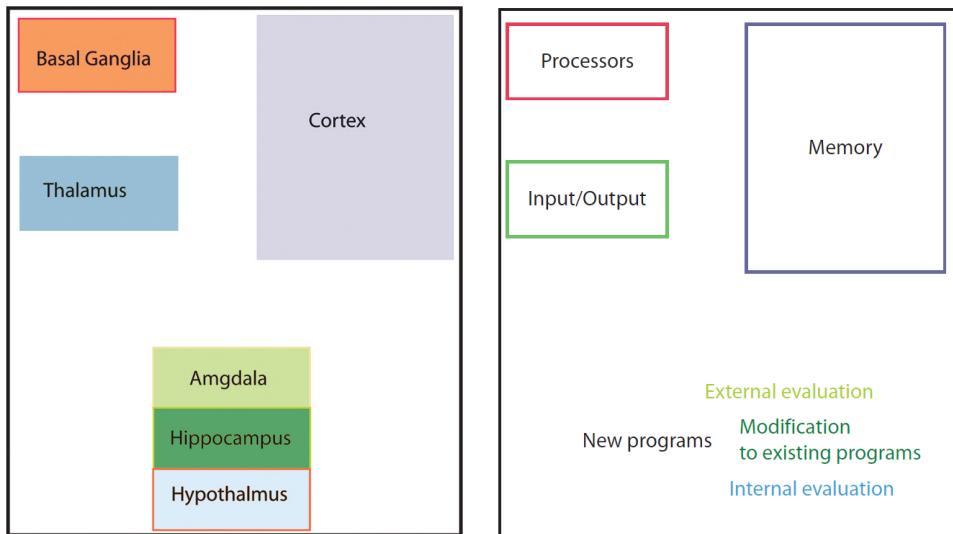


Introducción



El cerebro humano

Analogía entre el cerebro humano y un ordenador



[Dana Ballard: Brain Computation as Hierarchical Abstraction, 2015]



Introducción



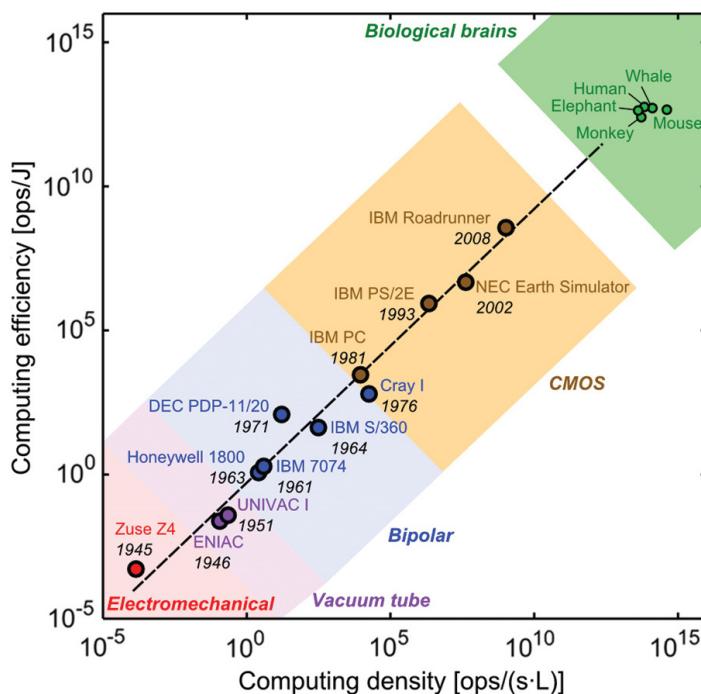
El cerebro humano

Diferencias entre un ordenador y el cerebro humano

Ordenador	Cerebro humano
Computación en serie	Computación en paralelo
Poco robusto	Tolerancia a fallos
Programable	Aprendizaje autónomo
Digital	Analógico
10⁹ transistores	10¹¹ neuronas 10¹⁴ ~ 10¹⁵ sinapsis
Nanosegundos (3.6GHz)	Milisegundos (4~90Hz)
51.2 GB/s	10 spikes/s
210,000,000 m/s	1 ~ 100 m/s
2.3x10¹³ TEPS	6.4x10¹⁴ TEPS



Introducción



IBM Journal of Research and Development, 2011

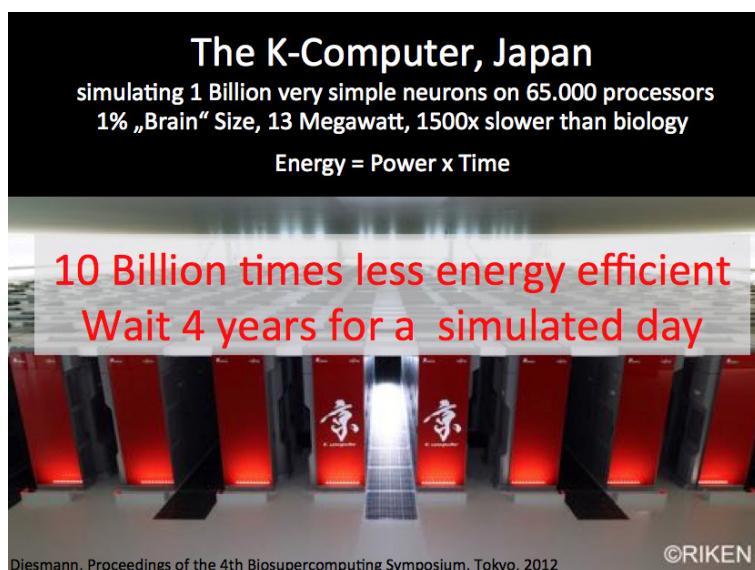


Introducción



Arquitecturas basadas en el cerebro

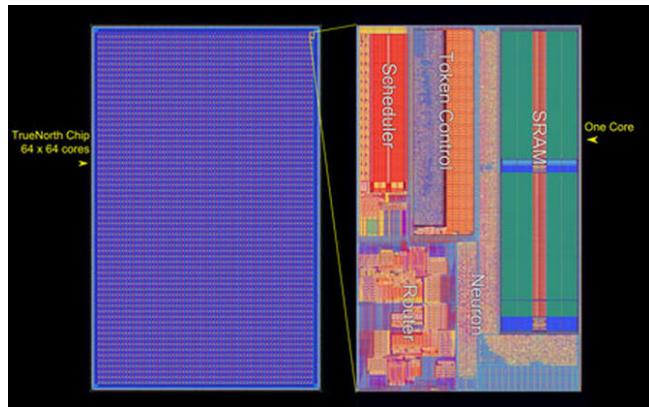
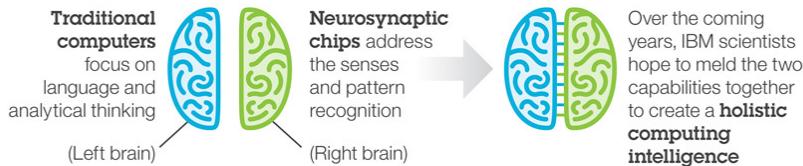
Simulación (muy ineficiente) → “Neuromorphic Computing”



Introducción



Arquitecturas basadas en el cerebro IBM TrueNorth Brain-inspired Computer



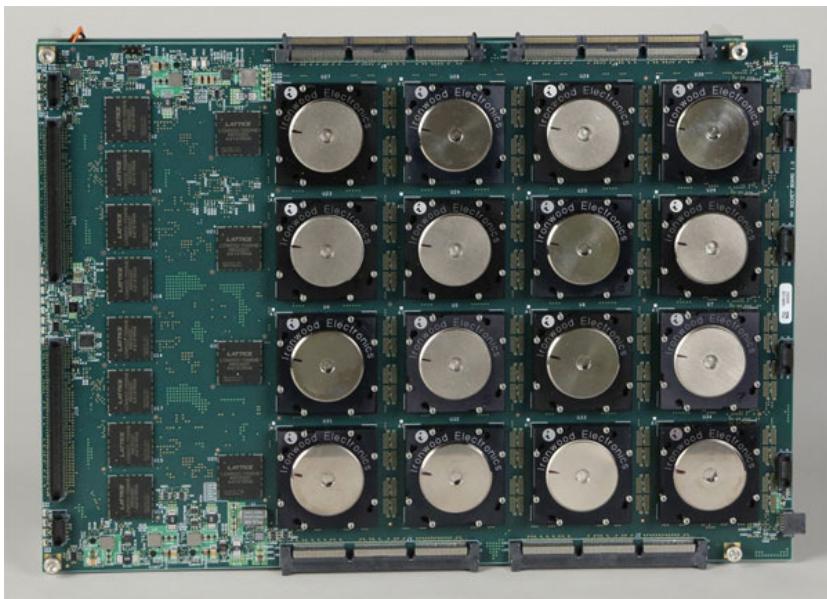
4096 cores
1M neurons
256M synapses
5.4B transistors
CMOS
70mW



Introducción



Arquitecturas basadas en el cerebro IBM TrueNorth Brain-inspired Computer



Synapse 16
16M neurons
4B synapses

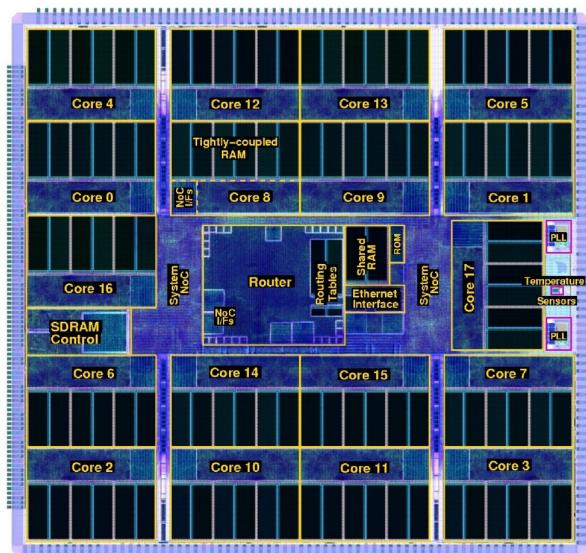
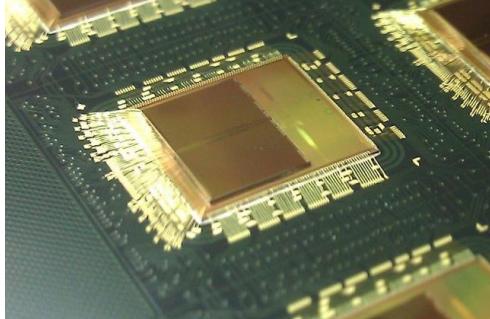


Introducción



Arquitecturas basadas en el cerebro

SpiNNaker project (UK)



Globally Asynchronous Locally Synchronous (GALS) chip:
18 ARM968 processor nodes + 128MB Mobile DDR SDRAM
<http://apt.cs.manchester.ac.uk/projects/SpiNNaker/project/>

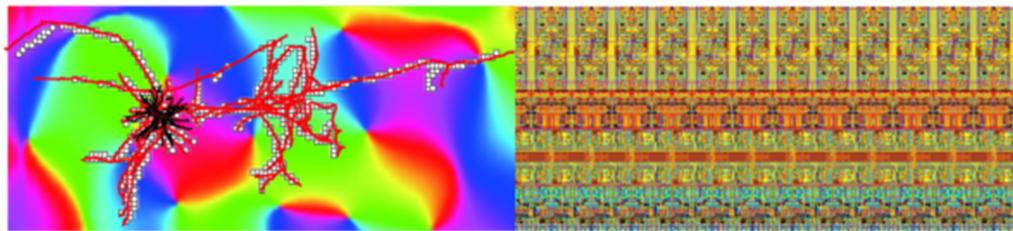
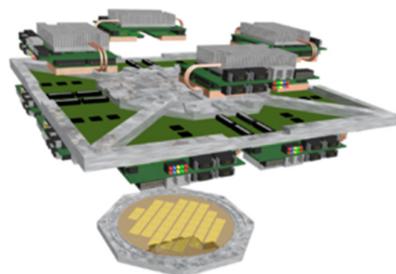


Introducción



Arquitecturas basadas en el cerebro

BrainScaleS (Germany)



Mixed CMOS signals

<https://brainscales.kip.uni-heidelberg.de/>

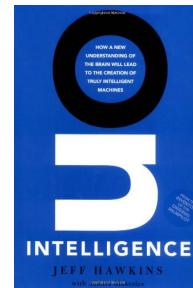
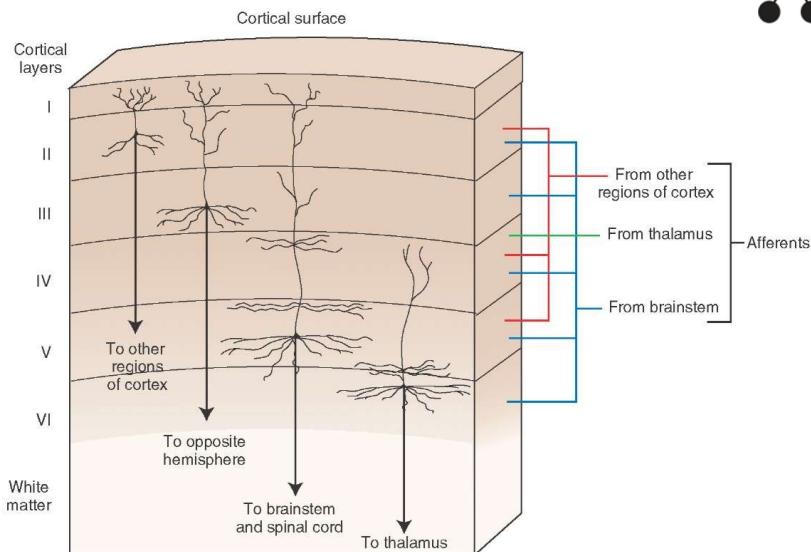


Introducción



Arquitecturas basadas en el cerebro

HTM [Hierarchical Temporal Memory]

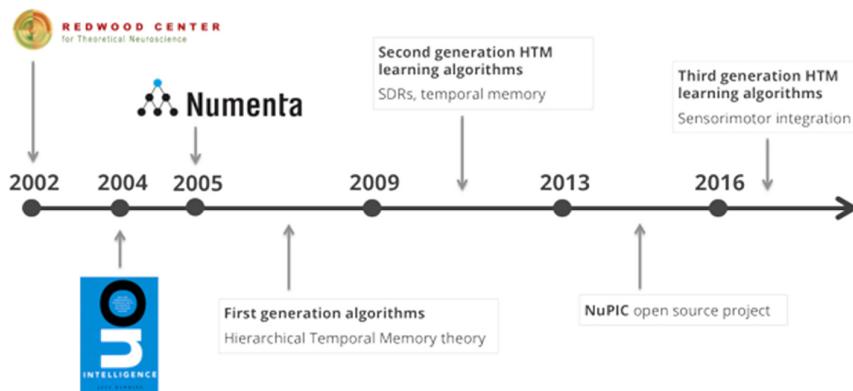


Introducción



Arquitecturas basadas en el cerebro

HTM [Hierarchical Temporal Memory]



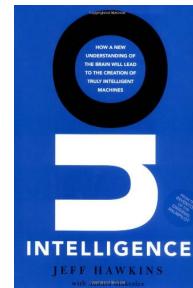
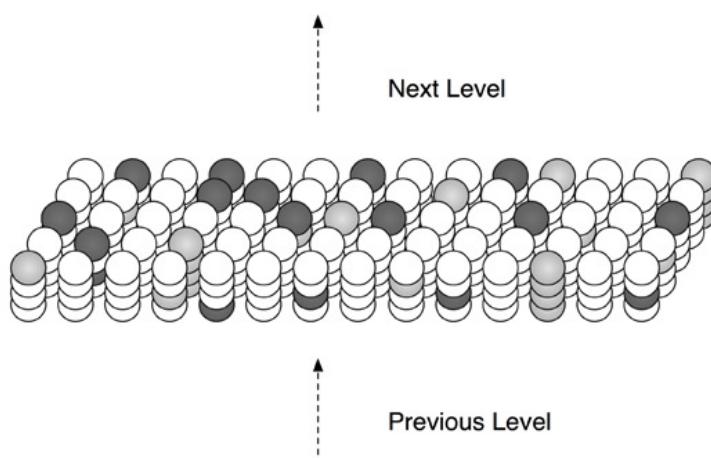
<http://numenta.com/>

Introducción



Arquitecturas basadas en el cerebro

HTM [Hierarchical Temporal Memory]



<http://numenta.com/>

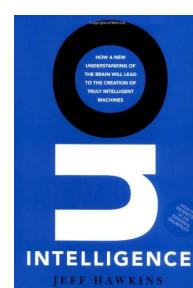
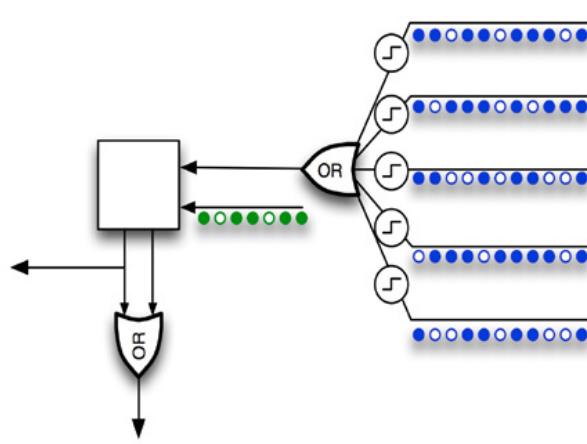
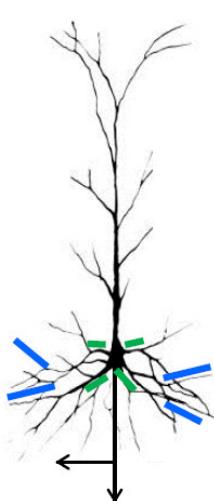


Introducción



Arquitecturas basadas en el cerebro

HTM [Hierarchical Temporal Memory]



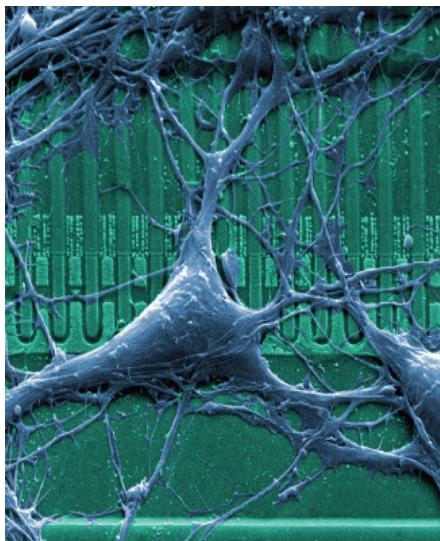
<http://numenta.com/>



Introducción



Neuronas



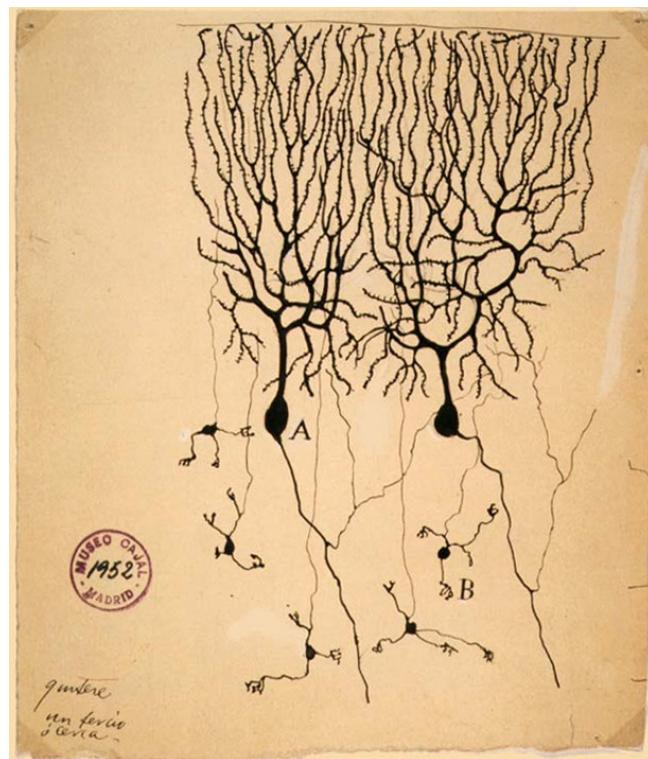
Microfotografía de una neurona “cultivada” sobre una oblea de silicio.
[Peter Fromherz, Max Planck Institute]



Introducción



Neuronas



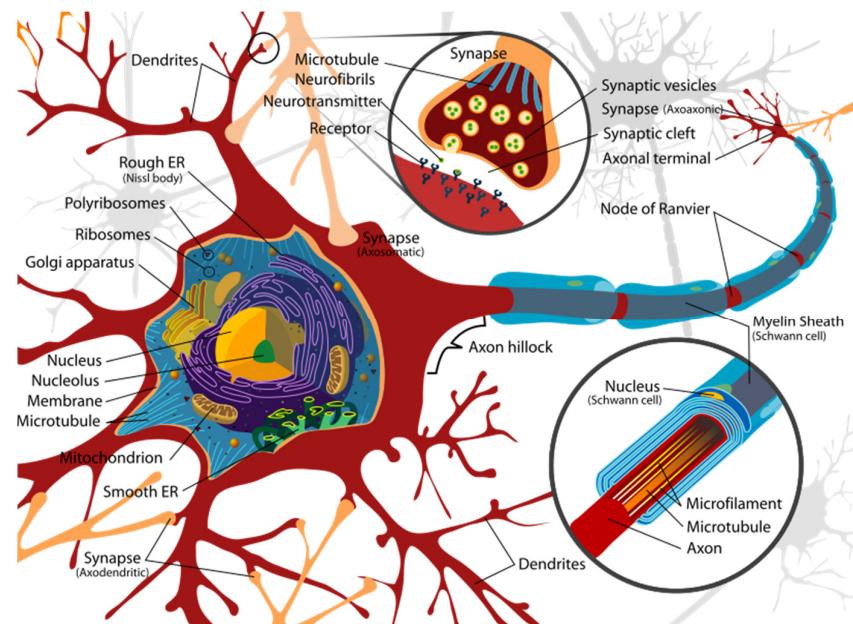
Neuronas del cerebelo
[Dibujo de Santiago
Ramón y Cajal, 1899]



Introducción



Neuronas



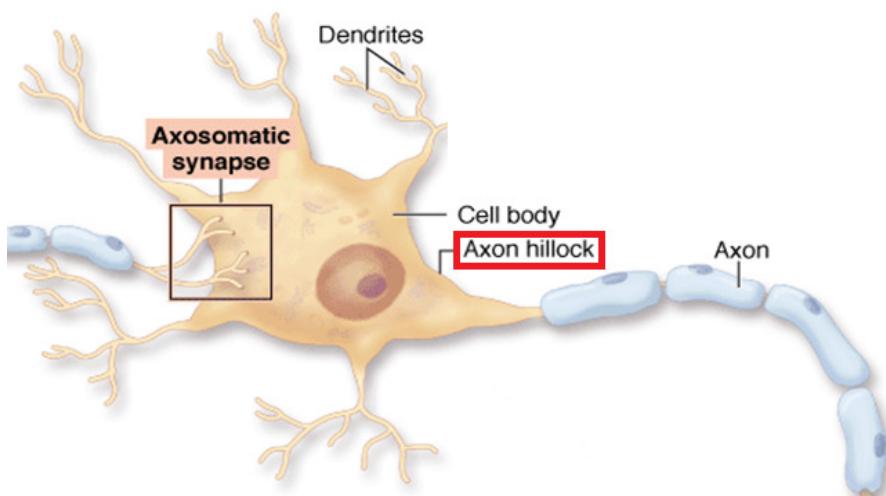
[Wikipedia]



Introducción



Neuronas



https://en.wikipedia.org/wiki/Axon_hillock

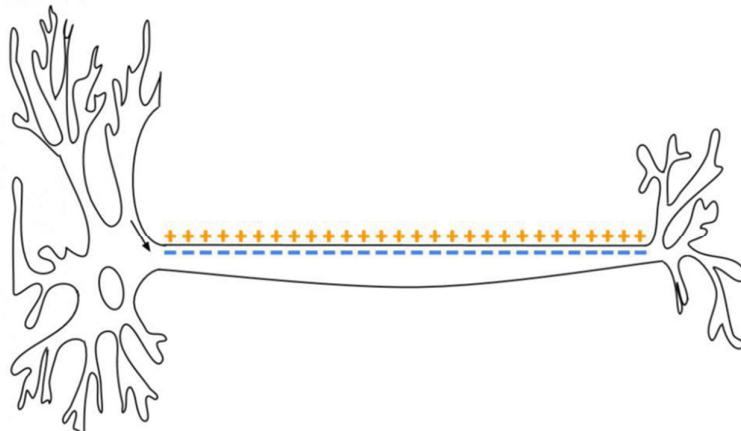


Introducción



Neuronas

Spike, a.k.a. action potential [potencial de acción]



MakeAGIF.com

https://en.wikipedia.org/wiki/Action_potential



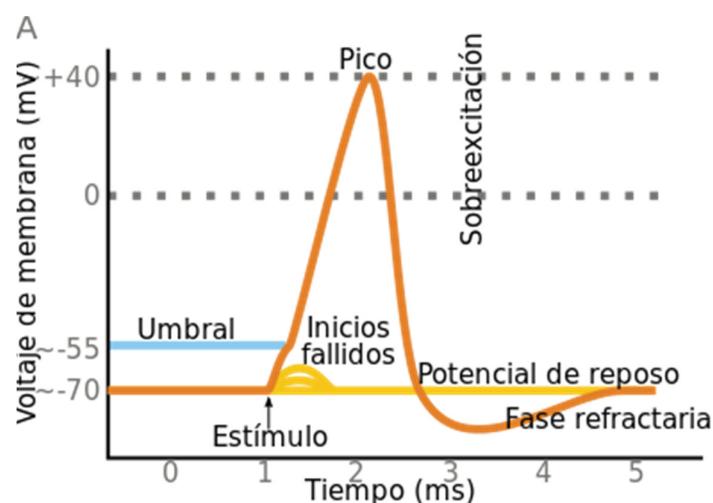
24

Introducción



Neuronas

Spike, a.k.a. action potential [potencial de acción]



https://en.wikipedia.org/wiki/Action_potential

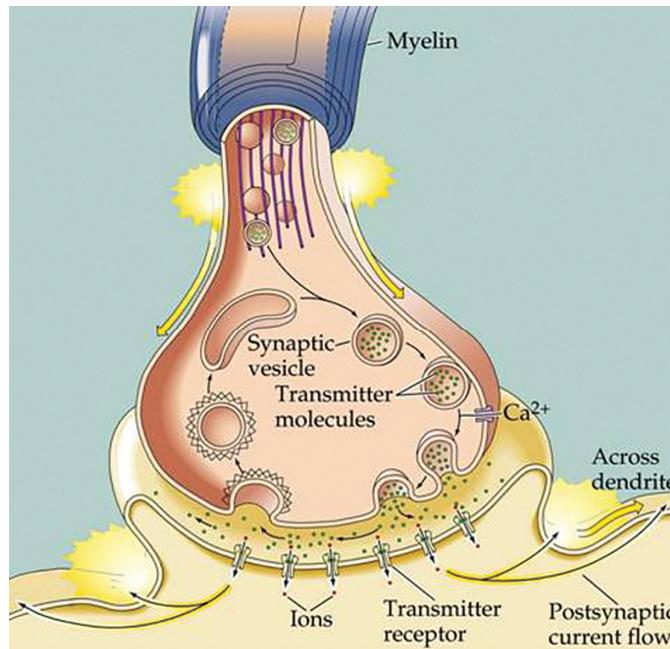


25

Introducción



Neuronas Sinapsis



[Purves et al.: Neuroscience, 3rd edition, 2004]



Introducción



Neuronas Sinapsis

Las sinapsis son lentas (en comparación con los transistores de un ordenador), pero...

- Son muy pequeñas y consumen muy poca energía.
- Se adaptan utilizando señales locales.

Como tenemos cerca de 10^{11} neuronas y de 10^{14} a 10^{15} sinapsis, muchas sinapsis pueden influir en un “cálculo” en un período de tiempo muy breve:

Ancho de banda muy superior al de un ordenador.



Introducción



Neuronas

Sinapsis

- El efecto de cada entrada sobre una neurona depende de un peso sináptico (positivo o negativo)
- Los pesos sinápticos se adaptan [plasticidad]: La “efectividad” de una sinapsis puede cambiar.
 - Neurona pre-sináptica:
Número de vesículas de neurotransmisores.
 - Neurona post-sináptica:
Número de receptores de neurotransmisores.

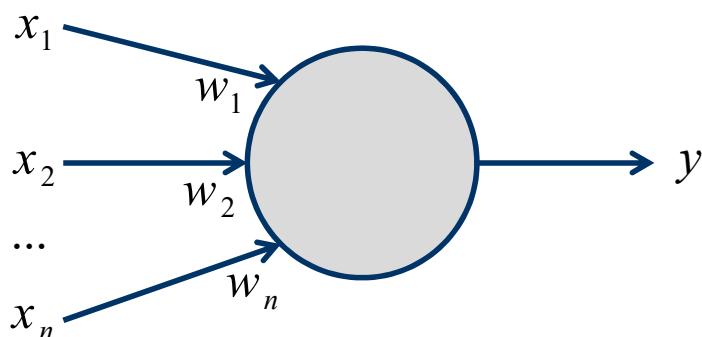


Introducción



Neuronas

El modelo computacional más simple de una neurona



$$y = \sum_i x_i w_i = x_1 w_1 + x_2 w_2 + \dots + x_n w_n$$



Introducción



Neuronas

- Diferentes partes del córtex se encargan de distintas tareas (daños locales tienen efectos específicos y la realización de tareas concretas aumenta el consumo de oxígeno en regiones determinadas).
- La estructura de todo el córtex es similar (6 capas de neuronas en una “servilleta arrugada” [Hawkins])

HIPÓTESIS

El córtex es un sistema de propósito general capaz de convertirse en hardware de propósito específico usando un algoritmo de aprendizaje (¿único?).



Introducción



Redes neuronales artificiales

Modelos de neuronas

- Para modelar las neuronas tenemos que idealizarlas: Eliminar de detalles irrelevantes que no son esenciales para entender su funcionamiento.
- La idealización nos permitirá utilizar herramientas (p.ej. matemáticas) y establecer analogías.



Introducción



Redes neuronales artificiales

Modelos de neuronas

- Una vez que tengamos un modelo básico, será más sencillo añadirle detalles y hacerlo más complejo para que sea más fiel a la realidad.
- Incluso modelos que son incorrectos de partida pueden resultarnos útiles.
p.ej. Asumir que las neuronas transmiten números reales en vez de potenciales de acción [spikes].



Introducción

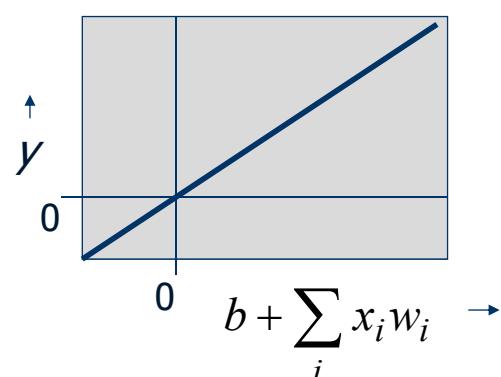


Redes neuronales artificiales

Modelos de neuronas: Neuronas lineales

$$y = b + \sum_i x_i w_i$$

y Salida
x Entradas
w Pesos
b Sesgo [bias]



- Sencillas, pero computacionalmente limitadas.



Introducción



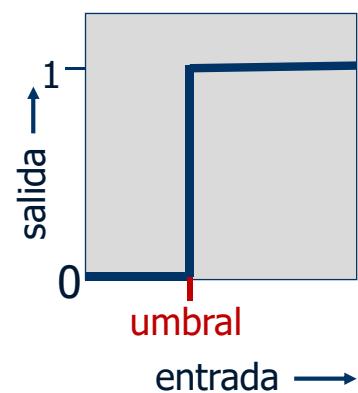
Redes neuronales artificiales

Modelos de neuronas: Neuronas binarias con umbral

[McCulloch & Pitts, 1943]

$$z = b + \sum_i x_i w_i$$

$$y = \begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$



Umbral $\theta=-b$



Introducción



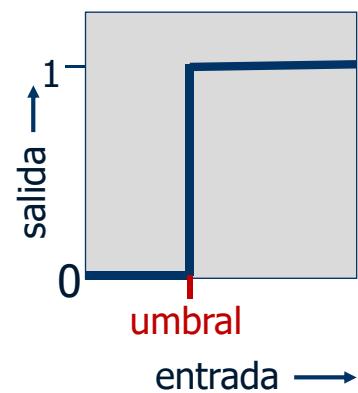
Redes neuronales artificiales

Modelos de neuronas: Neuronas binarias con umbral

[McCulloch & Pitts, 1943]

$$z = \sum_i x_i w_i$$

$$y = \begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$



Asumiendo $x_0=1$ y $w_0=b$ (umbral $\theta=-b$)



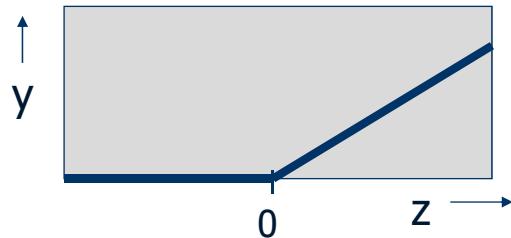
Introducción



Redes neuronales artificiales

Modelos de neuronas: Neuronas lineales rectificadas

$$z = \sum_i x_i w_i$$
$$y = \begin{cases} z & \text{si } z \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$



Asumiendo $x_0=1$ y $w_0=b$ (umbral $\theta=-b$)



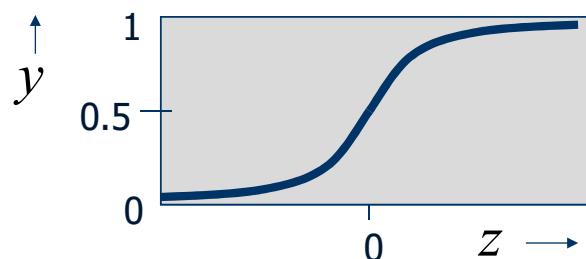
Introducción



Redes neuronales artificiales

Modelos de neuronas: Neuronas sigmoidales

$$z = \sum_i x_i w_i$$
$$y = \frac{1}{1 + e^{-z}}$$



- Función de activación suavizada y acotada.
p.ej. Función logística, tangente hiperbólica...
- El uso de sus derivadas facilita el aprendizaje.



Introducción

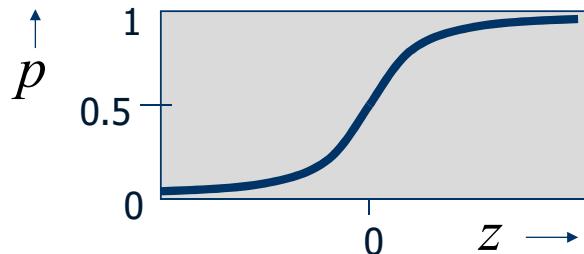


Redes neuronales artificiales

Modelos de neuronas: Neuronas binarias estocásticas

$$z = \sum_i x_i w_i$$

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



Las mismas ecuaciones que las neuronas sigmoidales, si bien su salida se interpreta como una probabilidad (de producir un spike en una pequeña ventana de tiempo)



Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)

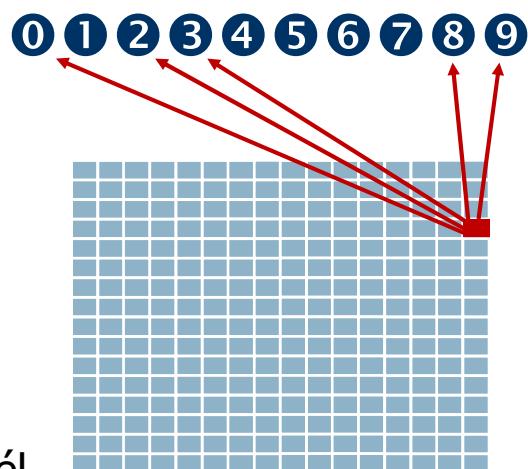
Red neuronal con 2 capas de neuronas:

- Capa de salida: Símbolos reconocidos.
- Capa de entrada: Píxeles de la imagen

Cada píxel vota si tiene tinta en él.

Cada píxel puede votar a varios símbolos.

El símbolo con más votos gana.

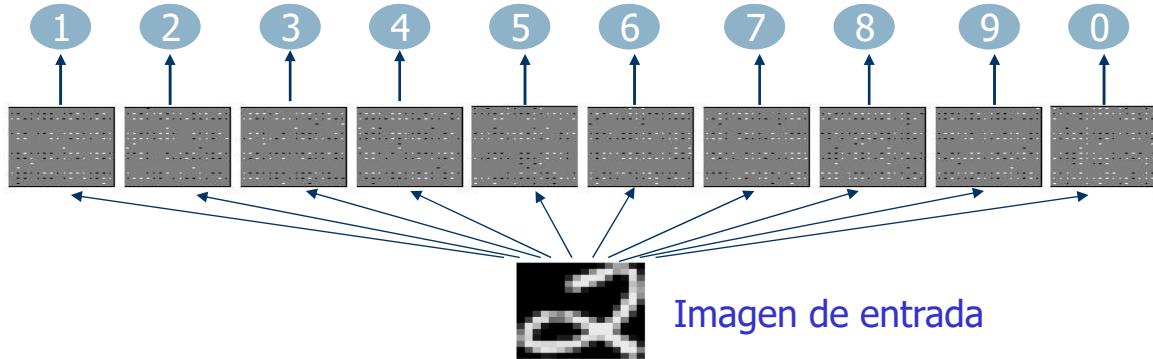


Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)



Visualización: Cada unidad de salida tiene su propio “mapa” de la imagen de entrada que muestra el peso asociado a cada píxel de la imagen de entrada.

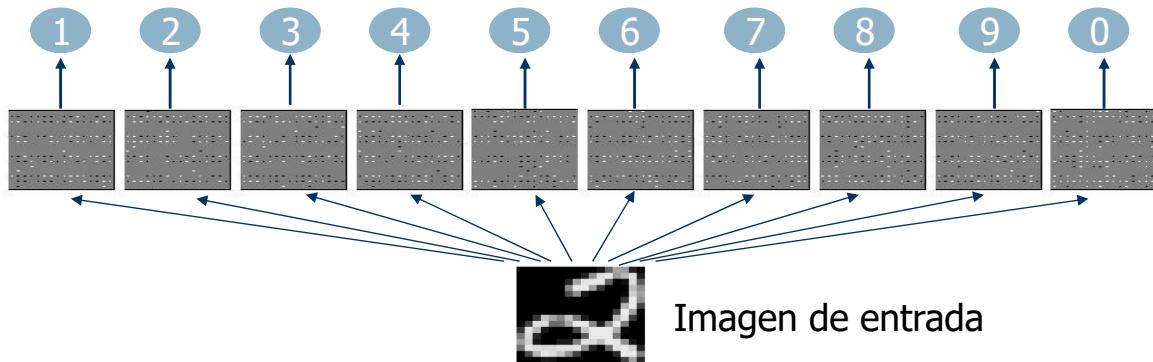


Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)



Entrenamiento: Se le enseña una imagen a la red...

- Se incrementan los pesos asociados a los píxeles activos para el símbolo de la imagen (clase correcta).
- Se decrementan los pesos de los píxeles activos de la imagen si la red se equivoca y predice un símbolo equivocado (error).

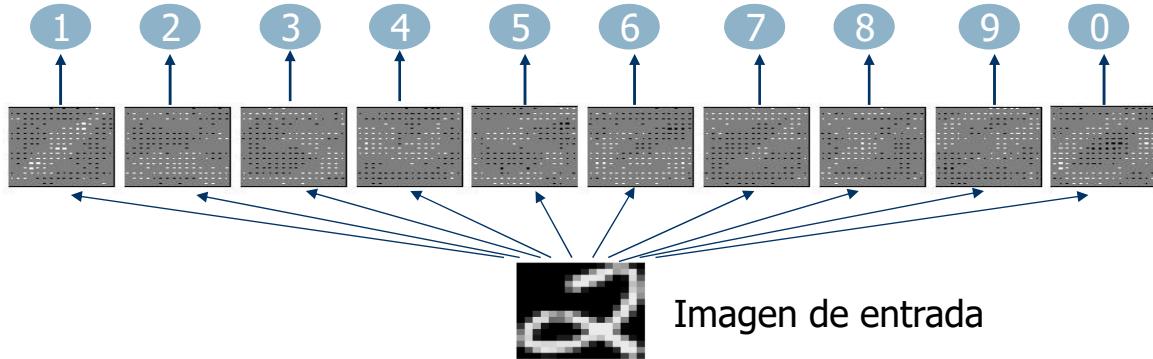


Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)



Aprendizaje...

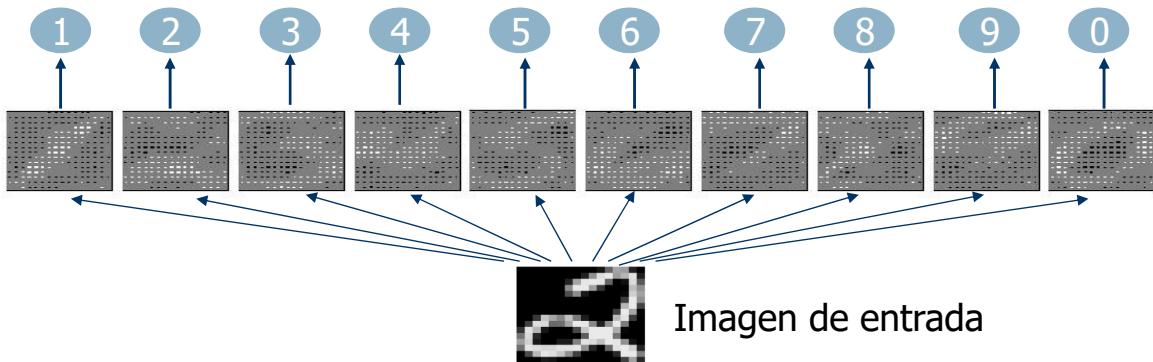


Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)



Aprendizaje...

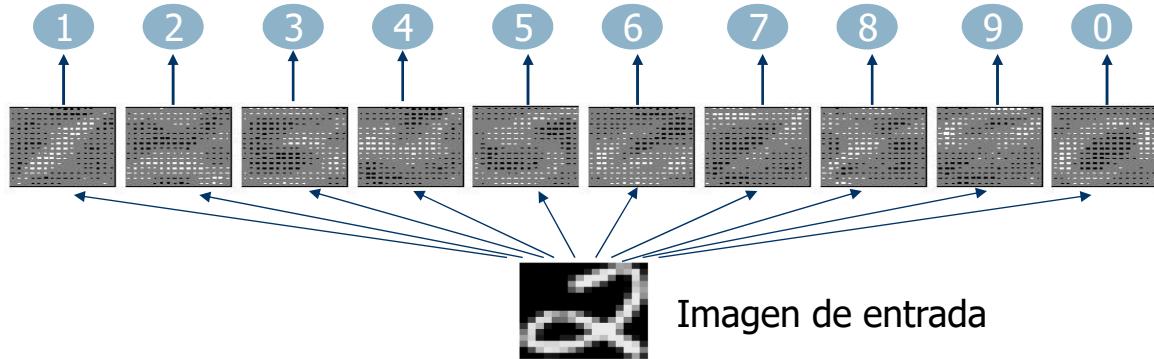


Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)



Aprendizaje...

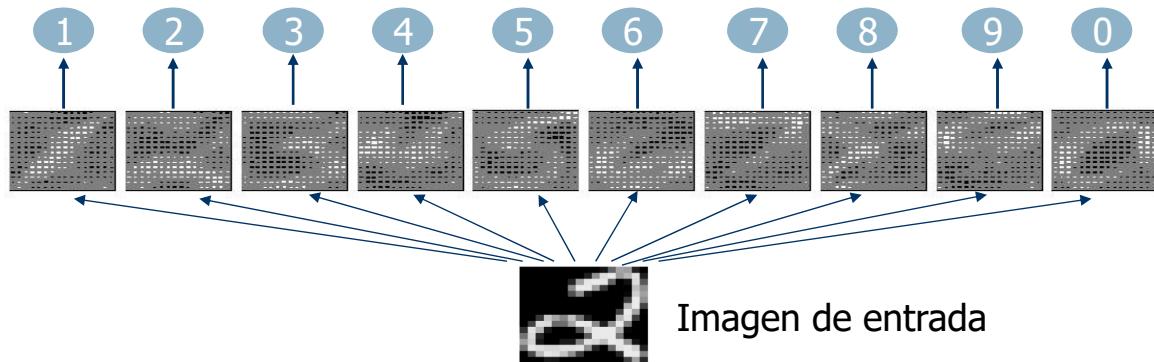


Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)



Aprendizaje...

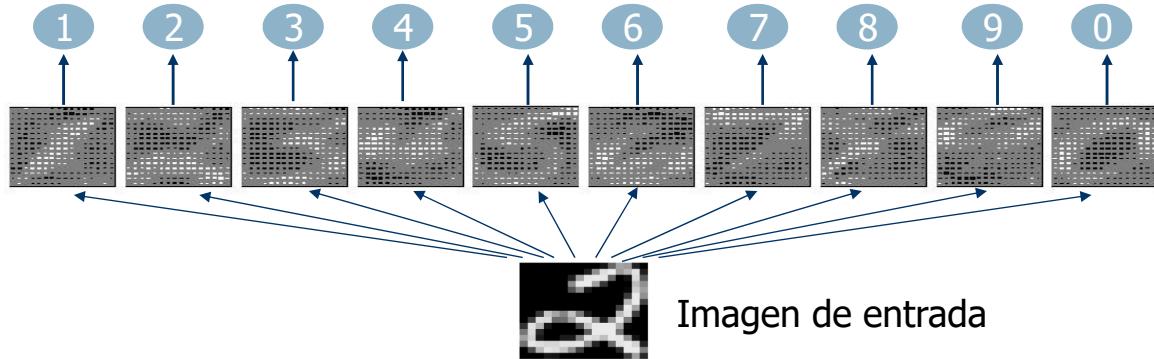


Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)



Aprendizaje...

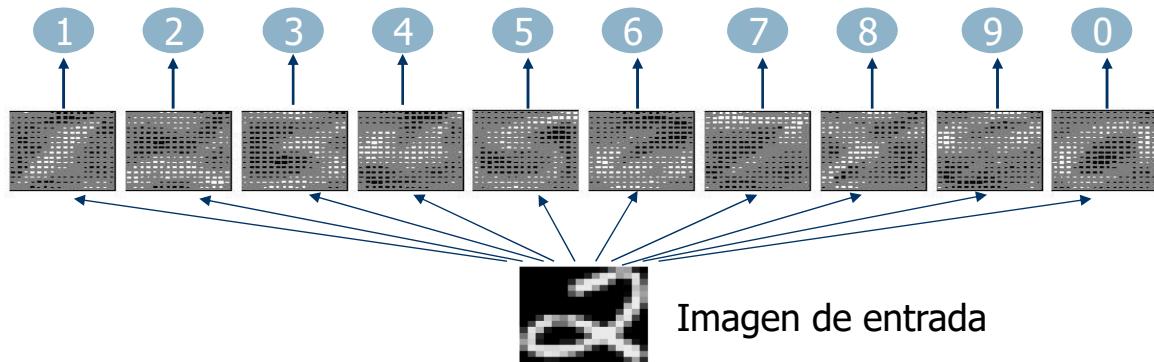


Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)



Aprendizaje...



Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)

- Una red neuronal tan simple, con una capa de entrada y una de salida, es equivalente a tener una plantilla rígida para cada símbolo (se elige el símbolo cuya plantilla se solapa más con la imagen de entrada).
- Las distintas formas en que pueden variar los dígitos manuscritos son demasiado complicadas para que se puedan capturar con plantillas tan simples.



Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)

- Aprenderemos formas de capturar las variaciones de los símbolos aprendiendo sus características: capas intermedias de neuronas, a.k.a. capas ocultas.
- Podremos incluso hacerlo utilizando técnicas no supervisadas, creando una representación interna de la entrada que luego sea útil en otras tareas (p.ej. aprendizaje supervisado para clasificar símbolos).



Introducción



Redes neuronales artificiales

Ejemplo de aprendizaje: Reconocimiento de dígitos (OCR)

- Seremos capaces de clasificar correctamente símbolos como los siguientes la primera vez que los veamos:

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



Historia



1943	Neurona de McCulloch-Pitts
1957	Perceptrón
1960	ADALINE
1969	Minsky & Papert: "Perceptrons"
1974-1986	Backpropagation
1982	Redes de Hopfield
1985	Máquinas de Boltzmann
1986	Harmonium [Restricted Boltzmann Machines]
2006	Deep Learning



Historia



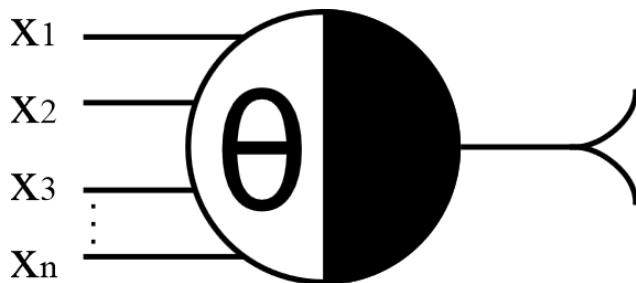
Nacimiento de las redes neuronales artificiales

1943 Circuitos booleanos como modelos del cerebro

Warren McCulloch & Walter Pitts:

"A logical calculus of the ideas immanent in nervous activity."

Bulletin of Mathematical Biophysics, 5:115-133.



Threshold Logic Unit (TLU):

Primer modelo de neurona artificial



52

Historia



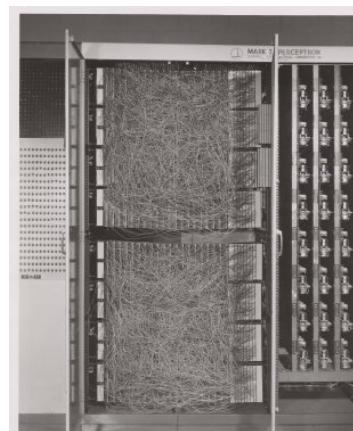
Perceptrón

1957 Algoritmo de aprendizaje supervisado

Frank Rosenblatt:

"The Perceptron--a perceiving and recognizing automaton".

Report 85-460-1, Cornell Aeronautical Laboratory, 1957.



Mark I Perceptron machine



53

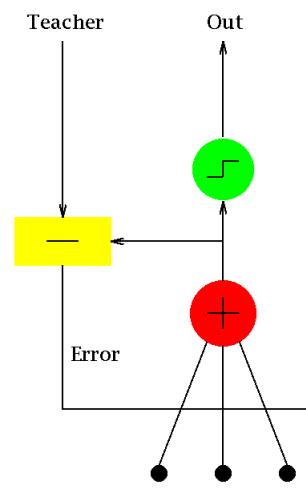
Historia



ADALINE [Adaptive Linear Element/Neuron]

1960 Red neuronal de una sola capa y dispositivo físico construido con memristores.

Bernard Widrow:
An adaptive “ADALINE” neuron
using chemical “memristors”
Technical Report 1553-2
Stanford University, 1960

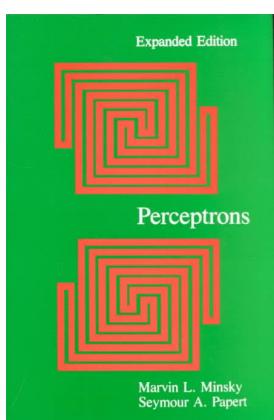


Historia



El invierno de la I.A.

1969 Marvin Minsky & Seymour Papert: “Perceptrons”
Análisis de las capacidades y limitaciones del perceptrón
- Abandono de modelos conexiónistas.
- La investigación en redes neuronales casi desaparece.



Marvin Minsky & Seymour Papert:
“Perceptrons:
An Introduction to Computational Geometry”
MIT Press, expanded edition, 1987
ISBN 0262631113



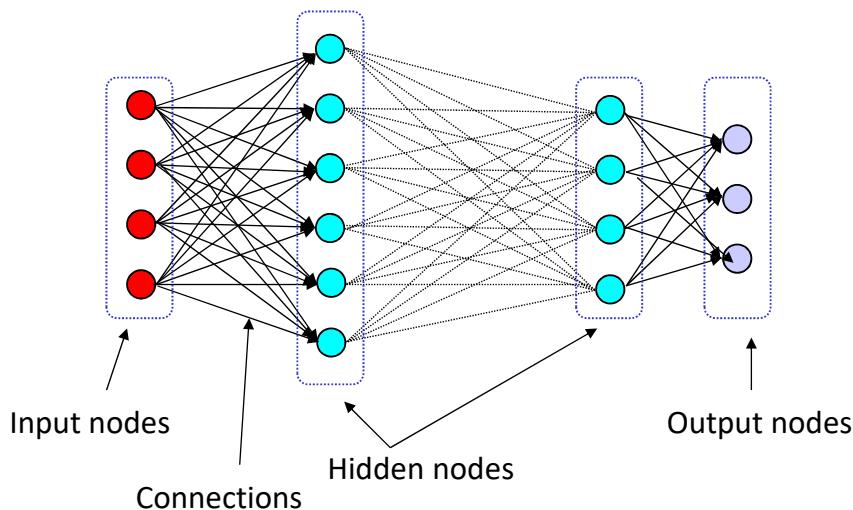
Historia



Renacimiento de las redes neuronales artificiales

1986 "Backpropagation"

(algoritmo de entrenamiento de redes neuronales multicapa)



Historia



Renacimiento de las redes neuronales artificiales

1986 "Backpropagation"

(las redes neuronales retoman su popularidad)

David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams

"Learning representations by back-propagating errors"

Nature 323(6088):533–536, 1986. DOI [10.1038/323533a0](https://doi.org/10.1038/323533a0)

NOTA: El algoritmo en sí fue "redescubierto" varias veces y utilizado en redes neuronales desde 1974.

Arthur E. Bryson, W.F. Denham & S.E. Dreyfus: "Optimal programming problems with inequality constraints. I: Necessary conditions for extremal solutions." AIAA J. 1(11):2544-2550, 1963.

Arthur Earl Bryson & Yu-Chi Ho: "Applied optimal control: optimization, estimation, and control." Blaisdell Publishing Company / Xerox College Publishing, p. 481, 1969.

Paul John Werbos: "The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting." John Wiley & Sons, Inc., 1994. ISBN 0471598976

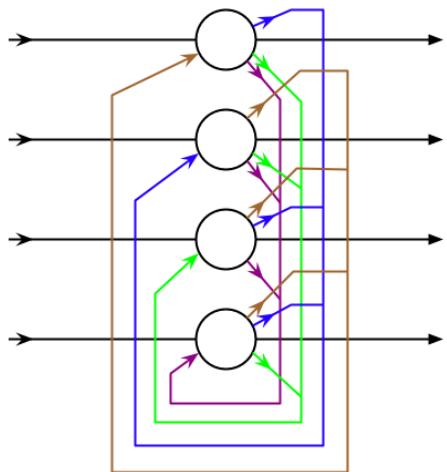


Historia



Redes de Hopfield

1982 Redes recurrentes que funcionan como memorias asociativas



John J. Hopfield:
"Neural networks and physical systems
with emergent collective computational abilities"
Proceedings of the National Academy of Sciences
PNAS 79(8):2554–2558, 1982



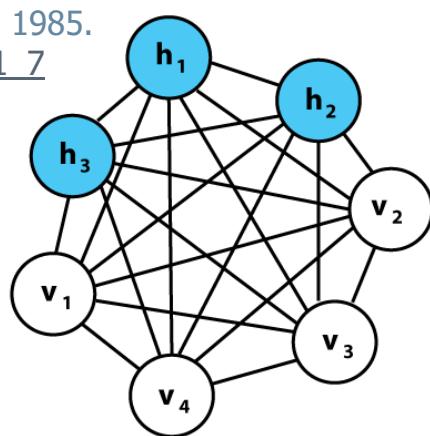
Historia



Máquinas de Boltzmann

1985 Máquinas de Boltzmann
(redes de Hopfield con neuronas ocultas)

David H. Ackley, Geoffrey E. Hinton & Terrence J. Sejnowski:
"A Learning Algorithm for Boltzmann Machines"
Cognitive Science 9(1):147–169, 1985.
[DOI 10.1207/s15516709cog0901_7](https://doi.org/10.1207/s15516709cog0901_7)

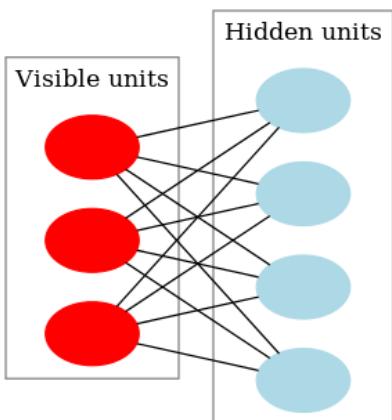


Historia



Máquinas de Boltzmann restringidas

1986 Harmonium = Restricted Boltzmann Machines



(máquinas de Boltzmann con estructura fija: grafos bipartidos con una capa de neuronas ocultas y una capa de neuronas "visibles", sin conexiones entre las neuronas de la misma capa)

Paul Smolensky: "Information Processing in Dynamical Systems: Foundations of Harmony Theory". In David E. Rumelhart & James L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations. MIT Press, chapter 6, pp. 194-281. ISBN 0-262-68053-X.

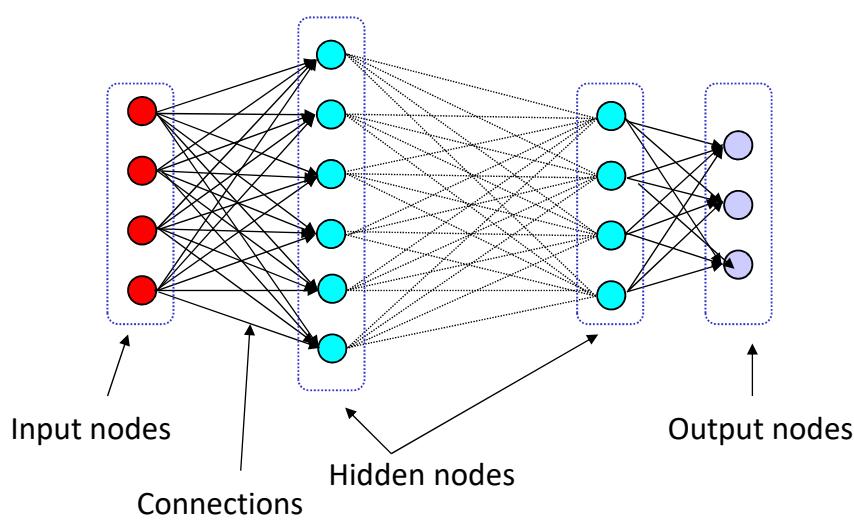


Historia



Deep Learning

Backpropagation no funciona bien con redes que tengan varias capas ocultas...



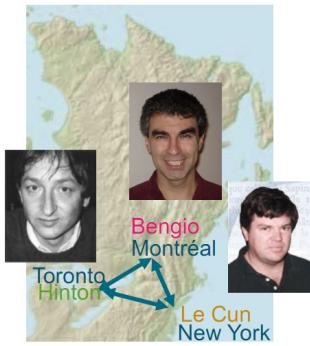
Historia



Deep Learning

2006 Nuevos algoritmos de aprendizaje

2006: The Deep Breakthrough



- Hinton, Osindero & Teh
« [A Fast Learning Algorithm for Deep Belief Nets](#) », *Neural Computation*, 2006
- Bengio, Lamblin, Popovici, Larochelle
« [Greedy Layer-Wise Training of Deep Networks](#) », *NIPS'2006*
- Ranzato, Poultney, Chopra, LeCun
« [Efficient Learning of Sparse Representations with an Energy-Based Model](#) », *NIPS'2006*

[Yoshua Bengio]



Historia



Otros modelos clásicos de redes neuronales artificiales

- Neocognitron
[Kunihiko Fukushima, 1980](#)
- Self-Organizing Map [SOM]
[Teuvo Kohonen, 1982](#)
- Counter-Propagation
[Robert Hecht-Nielsen, 1986](#)
- Adaptive Resonance Theory [ART]
[Stephen Grossberg & Gail Carpenter, 1987](#)
- Bidirectional Associative Memory [BAM]
[Bart Kosko, 1988](#)



Historia



- 1956: Psychologist Frank Rosenblatt uses theories about how brain cells work to design the perceptron, an artificial neural network that can be trained to categorize simple shapes.
- 1969: AI pioneers Marvin Minsky and Seymour Papert write a book critical of perceptrons that quashes interest in neural networks for decades.
- 1986: Yann LeCun and Geoff Hinton perfect backpropagation to train neural networks that pass data through successive layers of artificial neurons, allowing them to learn more complex skills.
- 1987: Terry Sejnowski at Johns Hopkins University creates a system called NETtalk that can be trained to pronounce text, going from random babbling to recognizable speech.
- 1990: At Bell Labs, LeCun uses back-propagation to train a network that can read handwritten text. AT&T later uses it in machines that can read checks.
- 1995: Bell Labs mathematician Vladimir Vapnik publishes an alternative method for training software to categorize data such as images. This sidelines neural networks again.
- 2006: Hinton's research group at the University of Toronto develops ways to train much larger networks with tens of layers of artificial neurons.
- June 2012: Google uses deep learning to cut the error rate of its speech recognition software by 25 percent.
- October 2012: Hinton and two colleagues from the University of Toronto win the largest challenge for software that recognizes objects in photos, almost halving the previous error rate.
- March 2013: Google buys DNN Research, the company founded by the Toronto team to develop their ideas. Hinton starts working at Google.
- March 2014: Facebook starts using deep learning to power its facial recognition feature, which identifies people in uploaded photos.
- May 2015: Google Photos launches. The service uses deep learning to group photos of the same people and let you search your snapshots using terms like "beach" or "dog."

MIT Technology Review

MIT Technology Review: "Teaching Machines to Understand Us", August 2015



Aplicaciones



¿Para qué sirven...

- ... las técnicas de aprendizaje automático?
- ... la inteligencia computacional?
- ... las redes neuronales artificiales?

Existen problemas para los que es extremadamente difícil desarrollar manualmente un programa de ordenador que los resuelva.

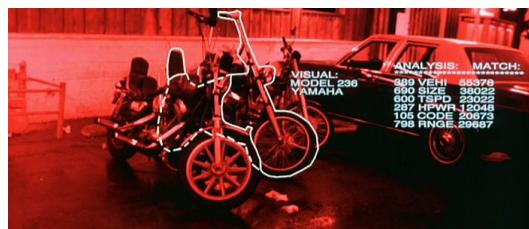


Aplicaciones



Ejemplos: Visión artificial

Reconocer un objeto tridimensional dentro de una escena, de la que tenemos un punto de vista nuevo, con condiciones de iluminación cambiantes...



[Terminator, 1984]



Aplicaciones



Ejemplos: Visión artificial

De hecho, el reconocimiento de objetos...

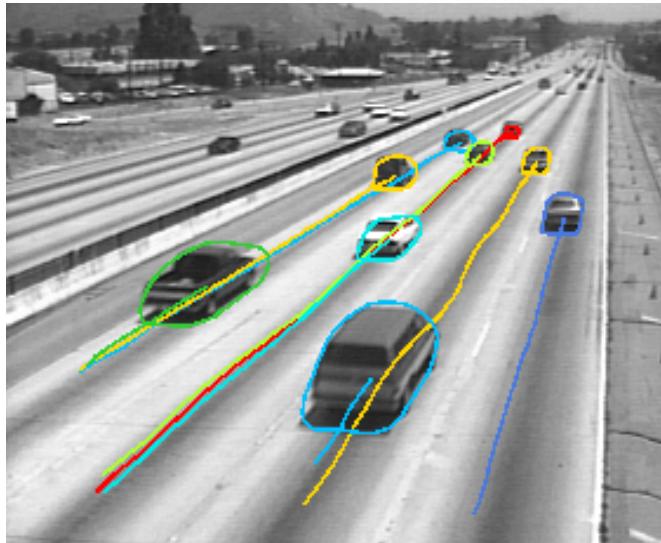
- Ni siquiera sabemos cómo se hace realmente en nuestro cerebro (por lo que difícilmente podremos diseñar un algoritmo que haga exactamente lo mismo).
- Incluso aunque tuviésemos una idea más precisa de cómo se hace en nuestro cerebro, el programa necesario podría ser tremadamente complicado :-(



Aplicaciones



Ejemplos: Visión artificial...



UC Berkeley project, funded by Caltrans, 1994



Aplicaciones



Ejemplos: Vehículos autónomos



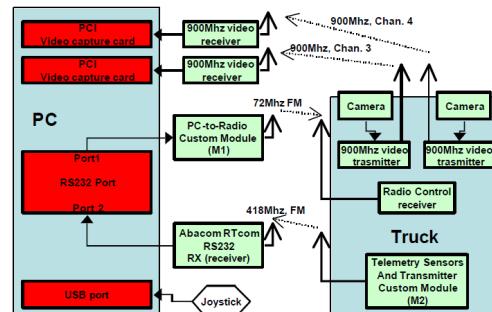
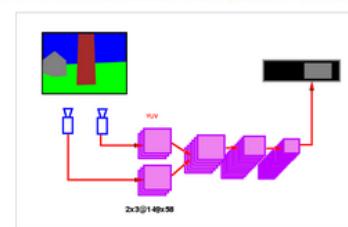
Autonomous Land Vehicle In a Neural Network (ALVINN)
NAVigational LABoratory II (NAVLAB II)
Control de dirección de un vehículo, CMU Ph.D. thesis, 1992



Aplicaciones



Ejemplos: Vehículos autónomos



DAVE, 2004

Autonomous Off-Road Vehicle Control using End-to-End Learning

NYU Courant Institute / CBLL [Computational & Biological Learning Lab]

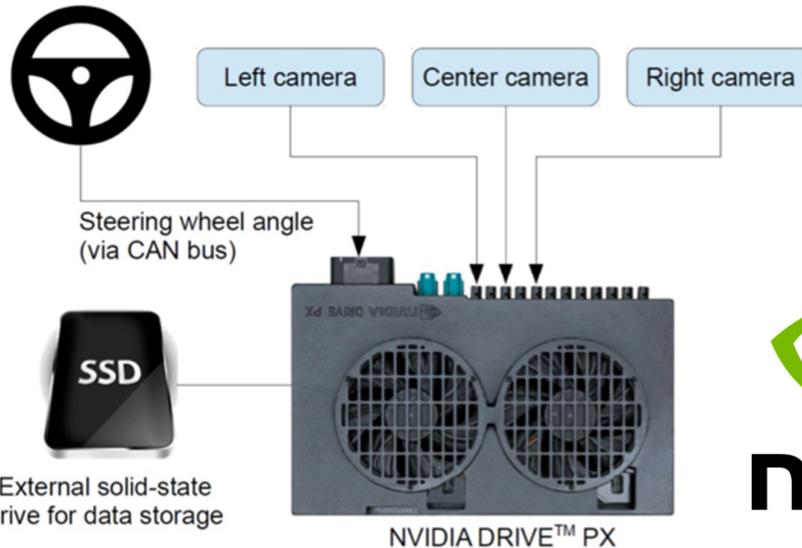
<http://www.cs.nyu.edu/~yann/research/dave/>



Aplicaciones



Ejemplos: Vehículos autónomos



NVIDIA.®

DAVE2, after DARPA Autonomous Vehicle (DAVE) project

NVIDIA, 2016. <http://arxiv.org/abs/1604.07316>

Completamente autónomo con sólo 100 horas de entrenamiento!!!



Aplicaciones



Ejemplos: Detección de fraudes

Calcular la probabilidad con la que una transacción realizada con una tarjeta de crédito es fraudulenta...

- No existen reglas simples que sean fiables (hay que combinar múltiples reglas que no siempre indican la presencia de fraude).
- Los tipos de fraude van cambiando, por lo que el programa que los detecte debe ir evolucionando.



Aplicaciones



La solución

Aprendizaje automático / Inteligencia computacional / RNA

- En vez de diseñar un algoritmo que resuelva el problema, recopilamos un montón de datos (ejemplos).
- Diseñamos un algoritmo que aprenda de esos datos y cree el programa necesario para resolver el problema.



Aplicaciones



La solución

- El programa generado automáticamente no tiene por qué parecerse a un programa implementado manualmente (en el caso de las redes neuronales, puede contener millones de números reales).
- Si tenemos éxito, el programa funcionará bien para nuevos ejemplos, aunque sean diferentes a los que utilizamos para su entrenamiento.
- Si los datos cambian, el programa puede cambiar entrenándolo de nuevo.



Aplicaciones



Más ejemplos

Tareas que se resuelven con éxito usando ML/CI/ANN

- Reconocimiento de patrones
 - Reconocimiento de objetos en imágenes
 - Reconocimiento facial
 - Reconocimiento de voz
- Detección de anomalías
 - Transacciones bancarias / tarjetas de crédito
 - Sensores en sistemas de control (p.ej. centrales nucleares)
 - Sistemas de detección de intrusiones
- Predicción
 - Cotizaciones de acciones / cambios de divisa
 - Sistemas de recomendación (películas, libros...)



Aplicaciones



The MNIST database of handwritten digits

<http://yann.lecun.com/exdb/mnist/>

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



Aplicaciones



The MNIST database of handwritten digits

<http://yann.lecun.com/exdb/mnist/>

- Benchmark estándar.

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

- Usaremos esta base de datos en prácticas para probar distintos algoritmos de aprendizaje y diferentes tipos de redes neuronales.

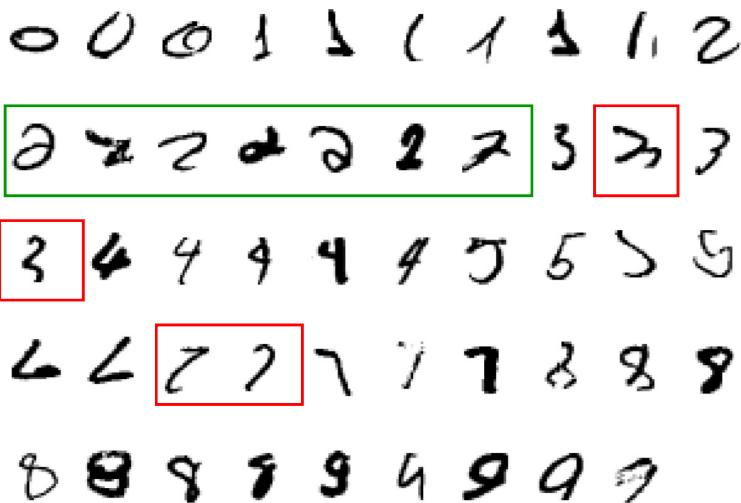


Aplicaciones



The MNIST database of handwritten digits

<http://yann.lecun.com/exdb/mnist/>



No siempre es tan fácil como puede parecer...



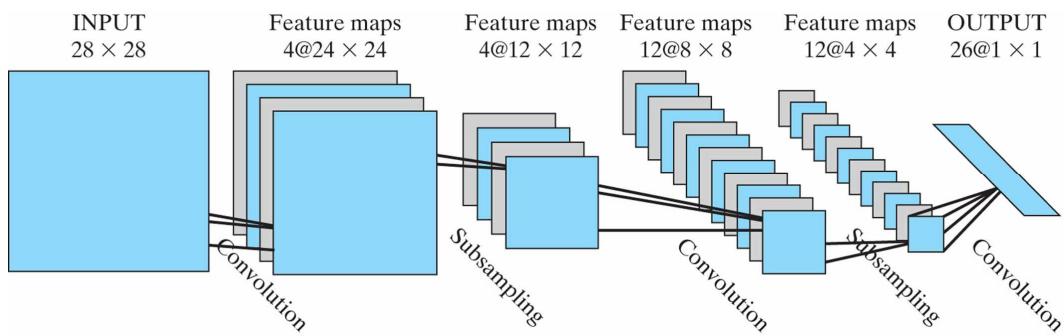
Aplicaciones



LeNet

<http://yann.lecun.com/exdb/lenet/>

Red neuronal con varias capas diseñada específicamente para reconocer dígitos y que, en su momento, se encargó de procesar automáticamente sobre el 10% de los cheques de EEUU.

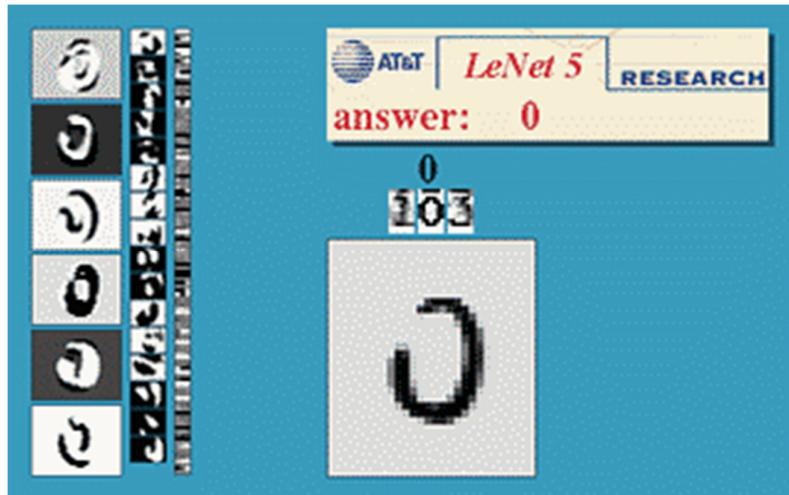


Aplicaciones



LeNet

<http://yann.lecun.com/exdb/lenet/>



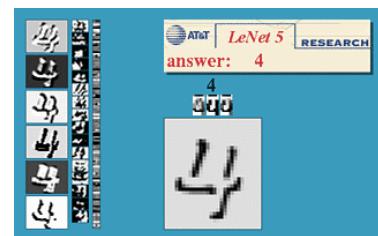
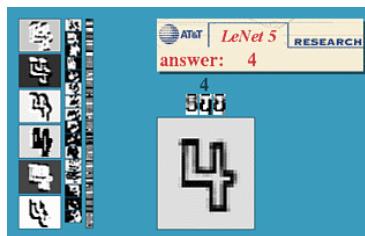
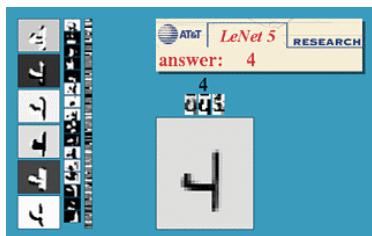
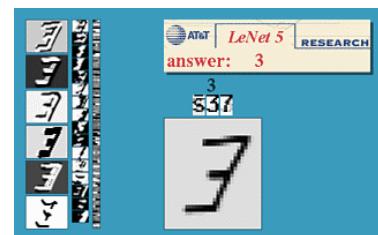
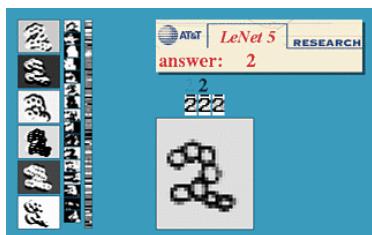
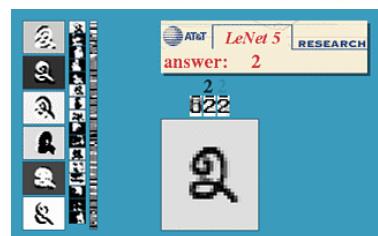
Aplicaciones



LeNet

<http://yann.lecun.com/exdb/lenet/>

Ejemplos



Aplicaciones



LeNet

<http://yann.lecun.com/exdb/lenet/>

Ejemplos

The image displays six examples of digit recognition from the LeNet-5 dataset. Each example consists of a small input image, a processing step, and the final output.

- Example 1:** Input is a handwritten digit '5'. The output shows a confidence score of 5.00 and the digit '5'.
- Example 2:** Input is a handwritten digit '5'. The output shows a confidence score of 5.00 and the digit '5'.
- Example 3:** Input is a handwritten digit '5'. The output shows a confidence score of 5.00 and the digit '5'.
- Example 4:** Input is a handwritten digit '6'. The output shows a confidence score of 6.00 and the digit '6'.
- Example 5:** Input is a handwritten digit '7'. The output shows a confidence score of 7.00 and the digit '7'.
- Example 6:** Input is a handwritten digit '8'. The output shows a confidence score of 8.00 and the digit '8'.



Aplicaciones



LeNet

<http://yann.lecun.com/exdb/lenet/>

Variaciones en los datos de entrada

The image displays three examples of digit recognition where the input data is varied.

- Example 1:** Input is a handwritten digit '00'. The output shows a confidence score of 0.00 and the digit '00'.
- Example 2:** Input is a handwritten digit '30'. The output shows a confidence score of 3.00 and the digit '30'.
- Example 3:** Input is a handwritten digit '384'. The output shows a confidence score of 384.00 and the digit '384'.



Aplicaciones



LeNet

<http://yann.lecun.com/exdb/lenet/>

Robustez frente a la presencia de ruido en la imagen...



Aplicaciones



LeNet

<http://yann.lecun.com/exdb/lenet/>

Casos curiosos



Aplicaciones



IM²GENET

Large Scale Visual Recognition Challenge

Reconocimiento de objetos reales en imágenes.

Mucho más difícil que OCR:

- Muchas más clases (1000 ImageNet vs. 10 MNIST).
- Más píxeles (256x256 color vs. 28x28 escala de grises)
- Vista bidimensional de una escena tridimensional.
- Múltiples objetos en la misma imagen.
- Solapamientos y desorden.



Aplicaciones



IM²GENET

Large Scale Visual Recognition Challenge

ILSVRC'2012: 1.2M imágenes, 1000 clases

Problemas planteados:

- **Clasificación:** Acertar la clase “correcta” (dentro de las cinco más probables).
- **Localización:** Identificar la zona de la imagen en la que se encuentra el objeto de interés (la región debe solaparse al menos al 50% con la zona correcta).



Aplicaciones



IM²GENET

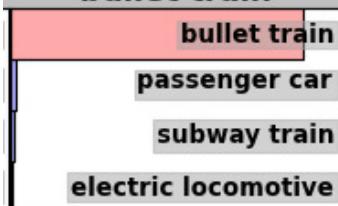
Large Scale Visual Recognition Challenge



cheetah



bullet train



hand glass



Aplicaciones

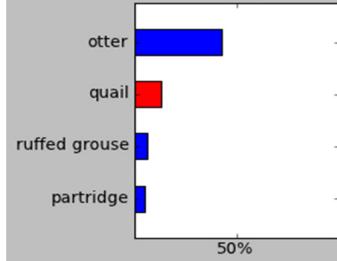


IM²GENET

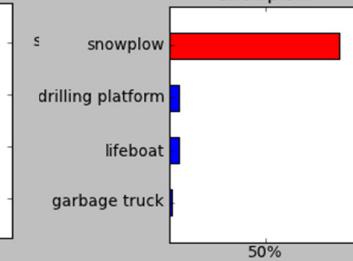
Large Scale Visual Recognition Challenge



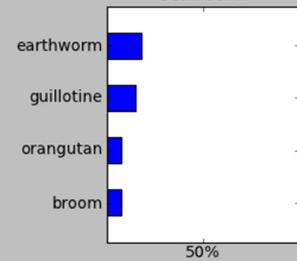
quail



snowplow



scabbard

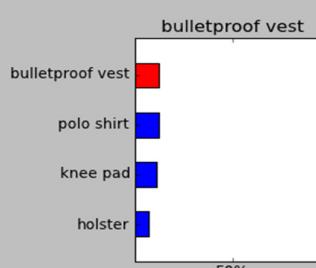
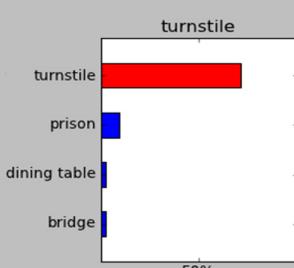
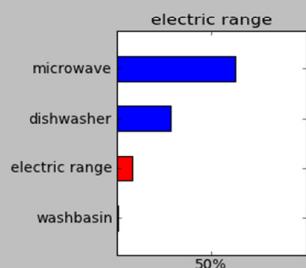


Aplicaciones



IMAGENET

Large Scale Visual Recognition Challenge

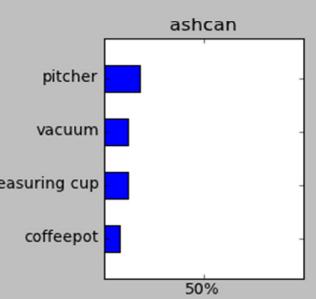
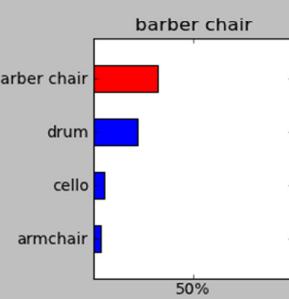
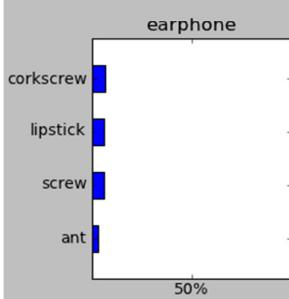
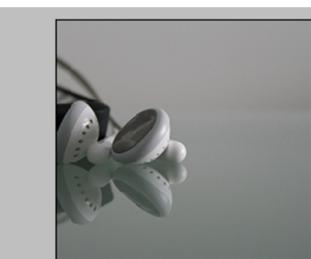


Aplicaciones



IMAGENET

Large Scale Visual Recognition Challenge



Aplicaciones

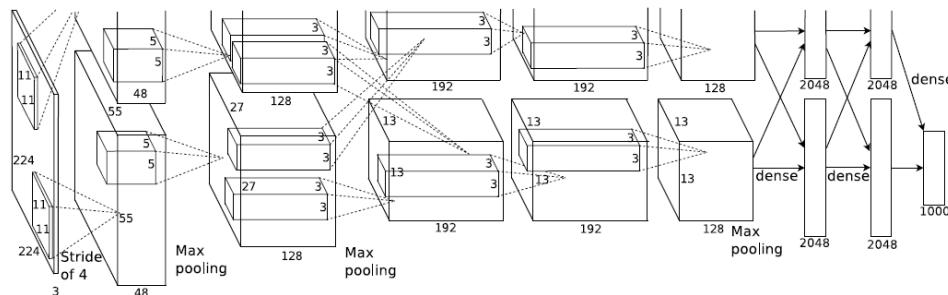


IM²GENET

Large Scale Visual Recognition Challenge

Red neuronal diseñada por Alex Krizhevsky (NIPS 2012)

- “Deep network” con topología compleja: 7 capas ocultas (las primeras convolutivas, las últimas completamente conectadas).



- Múltiples trucos para mejorar su capacidad de generalización (“image patches” & “dropout”).



Aplicaciones



IM²GENET

Large Scale Visual Recognition Challenge

Red neuronal diseñada por Alex Krizhevsky (NIPS 2012)

- Implementación usando 2 GPUs NVIDIA (procesadores SIMD con miles de núcleos y un ancho de banda muy alto con memoria, para acelerar la multiplicación de matrices y el tiempo de entrenamiento [1 semana]).



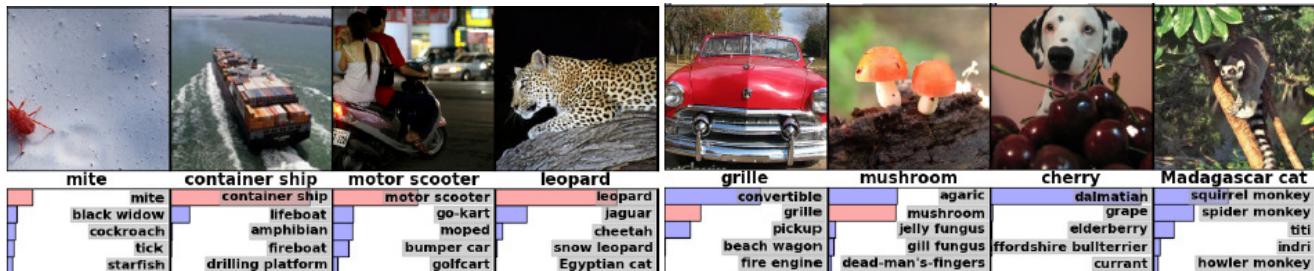
Aplicaciones



IMAGENET

Large Scale Visual Recognition Challenge

Red neuronal diseñada por Alex Krizhevsky (NIPS 2012)



Tasa de error

- Clasificación: **16.4%** vs. 25% (2010)
- Clasificación & localización: **34.1%**



Aplicaciones



IMAGENET

Large Scale Visual Recognition Challenge

MSR'2015



GT: horse cart
1: horse cart
2: minibus
3: oxcart
4: stretcher
5: half track



GT: birdhouse
1: birdhouse
2: sliding door
3: window screen
4: mailbox
5: pot



GT: forklift
1: forklift
2: garbage truck
3: tow truck
4: trailer truck
5: go-kart



GT: letter opener
1: drumstick
2: candle
3: wooden spoon
4: spatula
5: ladle



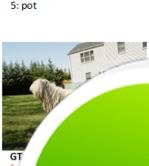
GT: letter opener
1: Band Aid
2: ruler
3: rubber eraser
4: pencil box
5: wallet



GT: letter opener
1: fountain pen
2: ballpoint
3: hammer
4: can opener
5: ruler



GT: coucal
1: coucal
2: indigo bunting
3: lorikeet
4: walking stick
5: custard apple



GT: 1
1: 2
2: 3
3: 4
4: 5



GT: go-kart
1: acoustic guitar
2: shot
3: bow tie
4: cowboy hat
5: banjo



GT: restaurant
1: wine bottle
2: candle
3: crash helmet
4: racer
5: sports car



GT: restaurant
1: goblet
2: plate
3: candle
4: French loaf
5: wooden spoon



GT: spotlight
1: altar
2: candle
3: perfume
4: restaurant
5: confectionery



GT: torch
1: stage light
2: spotlight
3: torch
4: microphone
5: feather boa



GT: banjo
1: acoustic guitar
2: shot
3: bow tie
4: cowboy hat
5: banjo



GT: go-kart
1: acoustic guitar
2: shot
3: bow tie
4: cowboy hat
5: banjo



GT: restaurant
1: wine bottle
2: candle
3: red wine
4: French loaf
5: wooden spoon



GT: restaurant
1: plate
2: meat loaf
3: ice cream
4: chocolate sauce
5: dining table



Aplicaciones



IM²GENET

Large Scale Visual Recognition Challenge

Tasa de error

16.4% Alex Krizhevsky @ NIPS 2012

6.66% GoogLeNet @ ILSVRC'2014

4.94% PreLU-nets (MSR) @ 2015

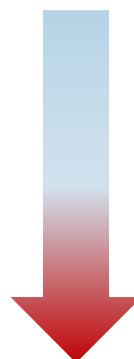
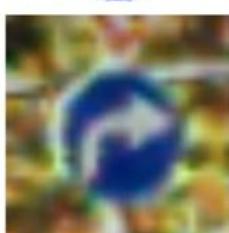
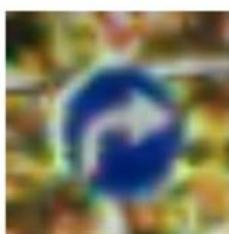
"Delving Deep into Rectifiers:
Surpassing Human-Level Performance on ImageNet Classification"
<http://arxiv.org/pdf/1502.01852v1.pdf> (February 2015)



Aplicaciones



¡OJO! No sólo se equivocan,
sino que se les puede engañar a propósito...

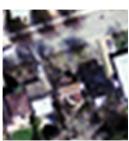
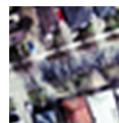


Aplicaciones



Detección de carreteras

- Se dispone de muchos datos ya etiquetados.
- Problemas prácticos:
 - Oclusiones (edificios, árboles, coches...)
 - Sombras y cambios de iluminación
 - Distintos puntos de vista (objetivos & subjetivos)
 - Errores en los mapas
- Las redes neuronales artificiales son “nuestra única esperanza”...

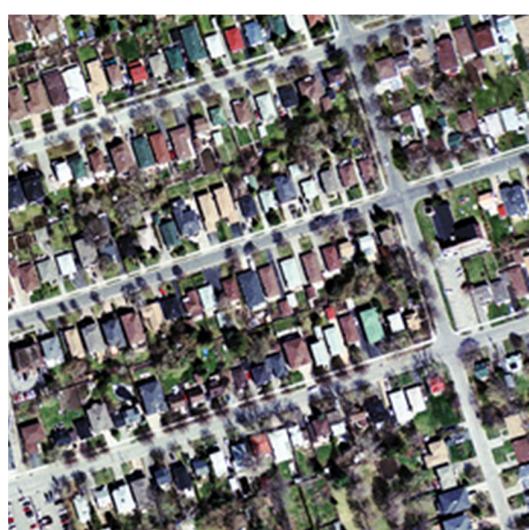


Aplicaciones



Detección de carreteras

Vlad Mnih (ICML 2012)



Aplicaciones



Descripción textual de imágenes

"Google's Brain-Inspired Software Describes What It Sees in Complex Images"
MIT Technology Review, November 2014

Describes without errors	Describes with minor errors	Somewhat related to the image	Unrelated to the image



Aplicaciones



Descripción textual de imágenes

"La inteligencia visual de un niño de 3 años..."
-- Fei-Fei Li (Stanford)



"The search for a thinking machine"

BBC News, 17 September 2015
<http://www.bbc.com/news/technology-32334573>



Aplicaciones



Microsoft
Common Objects in Context



Microsoft COCO Competition

<http://mscoco.org/>

ICCV'2015

Reconocimiento, segmentación y descripción textual de imágenes

- > 300 000 imágenes
- > 2M objetos
- 80 categorías
- 5 “captions” [leyendas] por imagen



Aplicaciones



Microsoft
Common Objects in Context

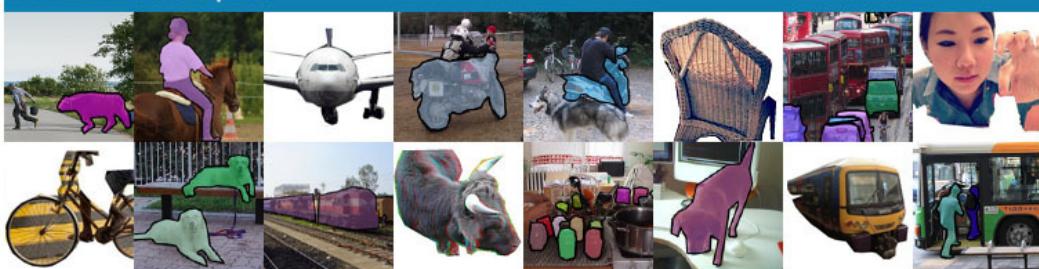


Microsoft COCO Competition

<http://mscoco.org/>

Reconocimiento, segmentación y descripción textual de imágenes

Dataset examples



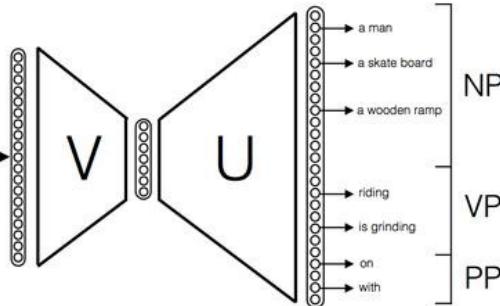
Aplicaciones



Microsoft
Common Objects in Context



A man in a helmet skateboarding before an audience.
Man riding on edge of an oval ramp with a skate board.
A man riding a skateboard up the side of a wooden ramp.
A man on a skateboard is doing a trick.
A man is grinding a ramp on a skateboard.



Rémi Lebret, Pedro O. Pinheiro & Ronan Collobert:
Phrase-based Image Captioning. ICML 2015:2085-2094
<http://jmlr.org/proceedings/papers/v37/lebret15.pdf>

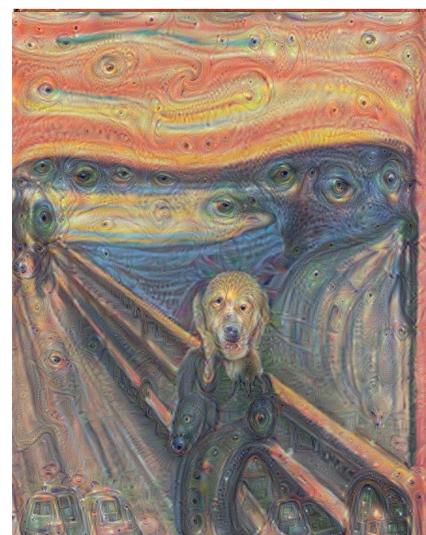


Aplicaciones



Síntesis de imágenes: “Inceptionism”

Usando una red ya entrenada para reconocer objetos...



“El grito”

Edvard Munch

... visto por una red neuronal

<http://deepr dreamgenerator.com>

<https://github.com/google/deepr dream>



Aplicaciones



Síntesis de imágenes

... combinando fotografías con distintos estilos:

A



D



Leon A. Gatys, Alexander S. Ecker & Matthias Bethge:

A Neural Algorithm of Artistic Style

arXiv, 2015. <http://arxiv.org/abs/1508.06576>



Aplicaciones



Síntesis de imágenes

... combinando fotografías con distintos estilos:

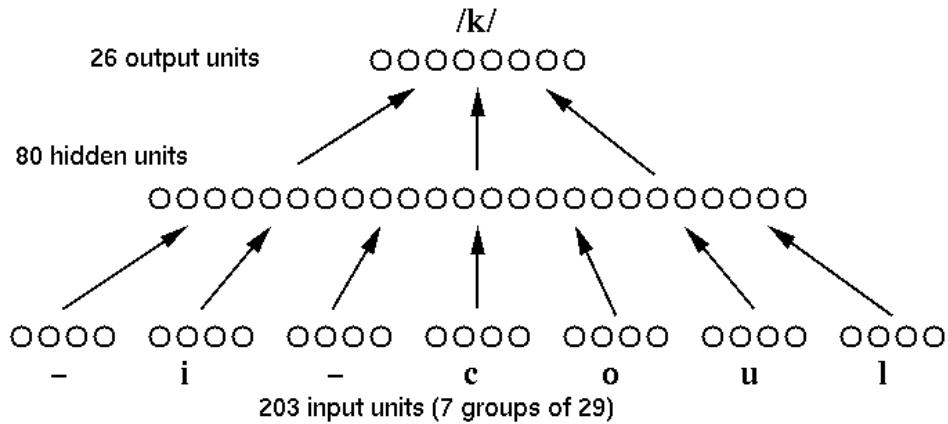


Aplicaciones



Síntesis de voz: NETtalk

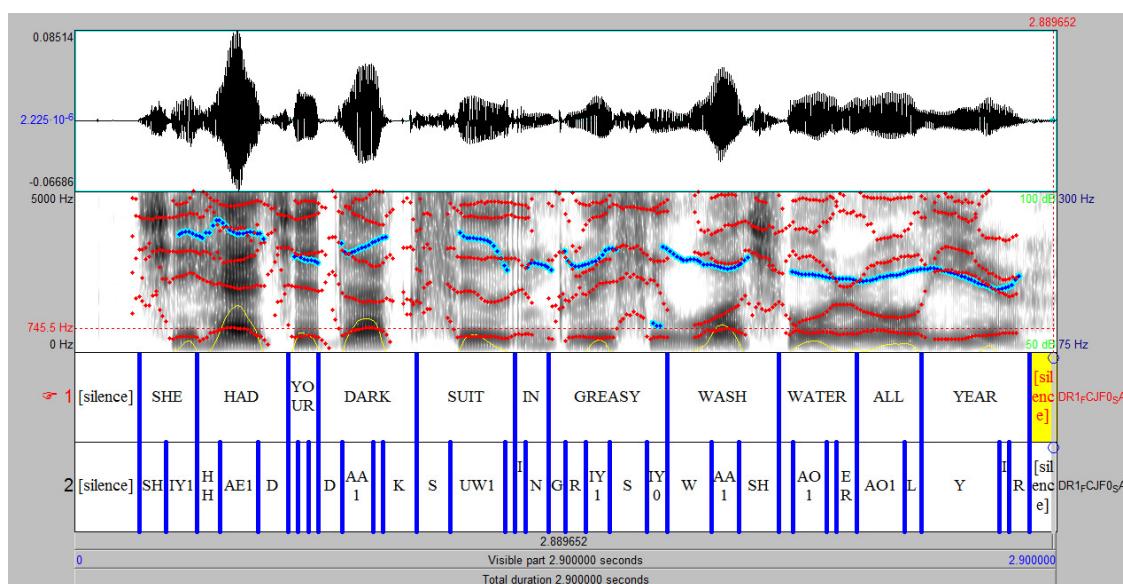
Terrence J. Sejnowski & Charles Rosenberg:
NETtalk: a parallel network that learns to read aloud,
Cognitive Science, 14, 179-211, 1986.



Aplicaciones



Reconocimiento de voz

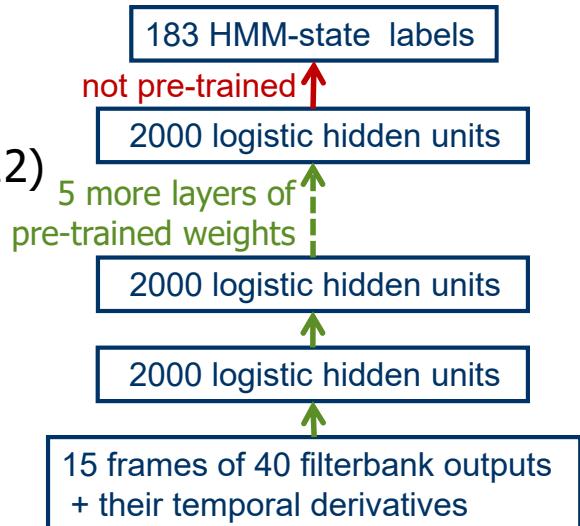


Aplicaciones



Reconocimiento de voz

Identificación de fonemas
usando “deep networks”
(Mohamed, Dahl, & Hinton, 2012)



TIMIT benchmark

- “Deep network”: 20.7% error en la identificación de fonemas (frente al 24.4% del mejor sistema previo)



Aplicaciones



Reconocimiento de voz

(Hinton et al.: IEEE Signal Processing Magazine, 2012)

Microsoft®

Research



Task	Hours of training data	Deep Neural Network	Gaussian Mixture Model	GMM with more data
Switchboard (Microsoft Research)	309	18.5%	27.4%	18.6% (2000 hrs)
English broadcast news (IBM)	50	17.5%	18.8%	
Google Voice Search (Android 4.1)	5,870	12.3% (and falling)		16.0% (>>5,870 hrs)



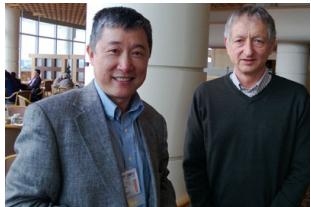
Aplicaciones



Reconocimiento de voz

Traducción simultánea [Machine Translation]

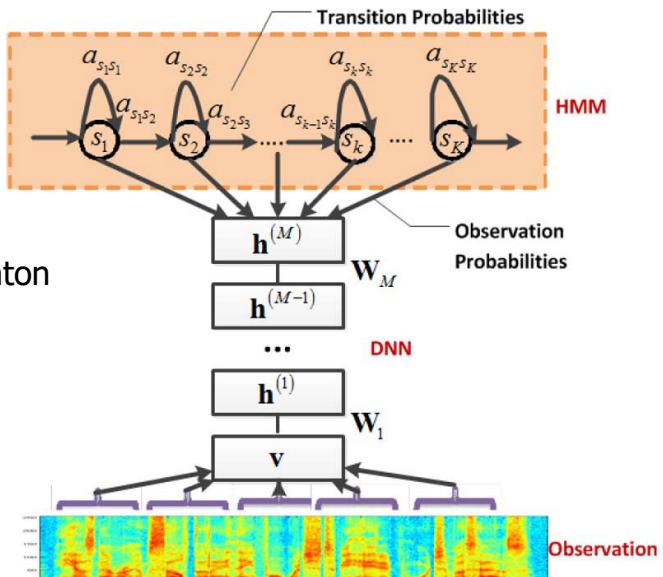
Microsoft®
Research



Li Deng (MSR) & Geoff Hinton



Dong Yu (MSR)



Demos

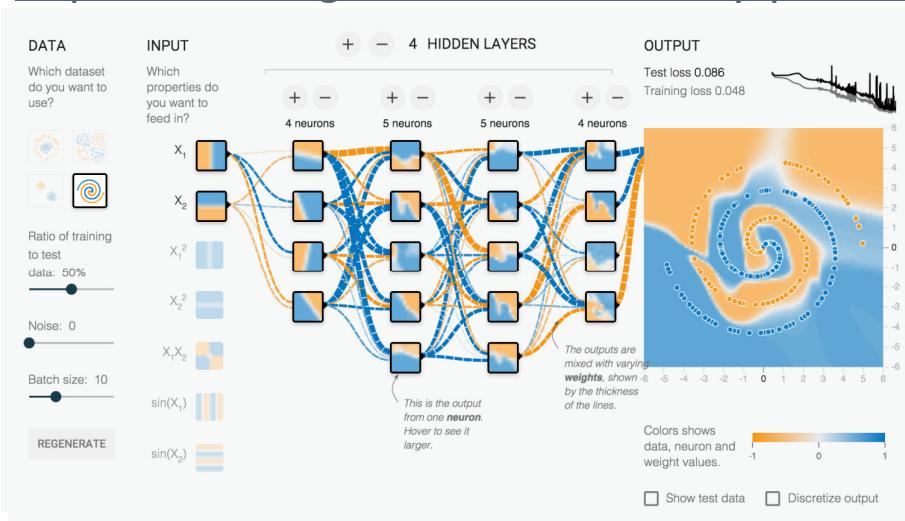
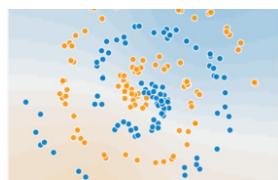


Para jugar un poco...

<http://playground.tensorflow.org/>

<http://ml4a.github.io/demos/>

<http://demos.algorithmia.com/classify-places/>



Referencias

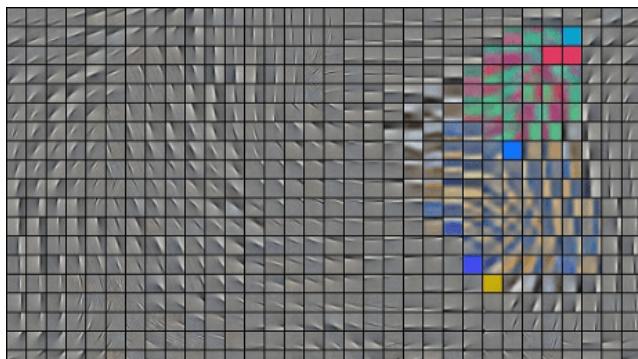


Neural Networks for Machine Learning

by Geoffrey Hinton

(University of Toronto & Google)

<https://www.coursera.org/course/neuralnets>



Referencias



■ Deep Learning Tutorial

Andrew Ng et al. (Stanford University)

<http://ufldl.stanford.edu/tutorial/>

■ Deep Learning: Methods and Applications

Li Deng & Dong Yu (Microsoft Research)

<http://research.microsoft.com/apps/pubs/default.aspx?id=209355>

■ Deep Learning

Joshua Bengio et al. (University of Montréal)

<http://www.iro.umontreal.ca/~bengioy/dlbook/>

■ Deep Learning for Natural Language Processing

Richard Socher et al. (Stanford University CS224d)

<http://cs224d.stanford.edu/>



Investigadores destacados



Geoffrey Hinton
(University of Toronto & Google)



Yann LeCun
(AT&T Labs → New York University → Facebook)



Joshua Bengio
(University of Montréal & IBM Watson)



Andrew Ng
(Stanford University → Coursera → Baidu)

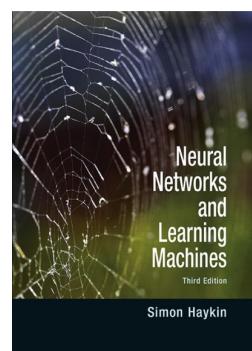


Bibliografía

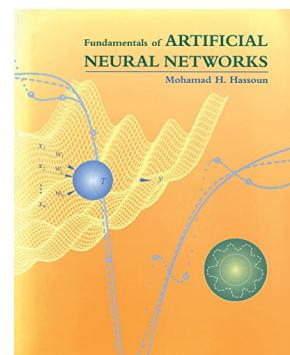


Lecturas recomendadas

- Simon Haykin:
**Neural Networks
and Learning Machines**
Prentice Hall, 3rd edition, 2008
ISBN 0131471392



- Mohamad Hassoun:
**Fundamentals of
Artificial Neural Networks**
MIT Press, 2003
ISBN 0262514672

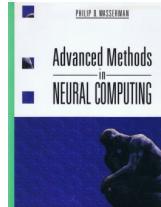
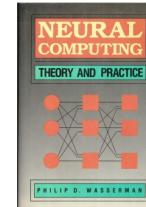
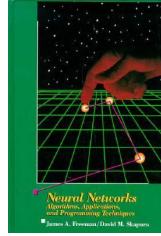
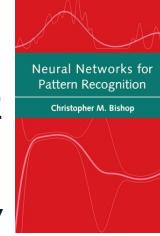


Bibliografía complementaria



Redes neuronales artificiales

- Christopher M. Bishop:
Neural Networks for Pattern Recognition
Oxford University Press, 1996. ISBN 0198538642
- James A. Freeman & David M. Skapura:
Neural Networks: Algorithms, Applications, and Programming Techniques
Addison-Wesley, 1991. ISBN 0201513765
- Philip D. Wasserman:
Neural Computing: Theory and Practice,
Van Nostrand Reinhold, 1989. ISBN 0442207433
- Philip D. Wasserman:
Advanced Methods in Neural Computing
Van Nostrand Reinhold, 1993. ISBN 0442004613

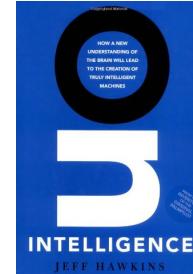
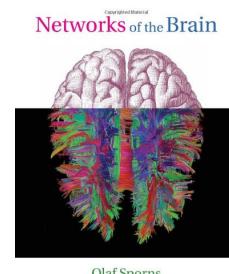
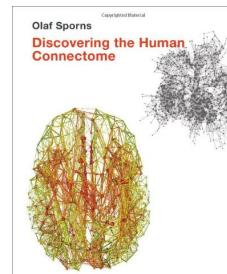
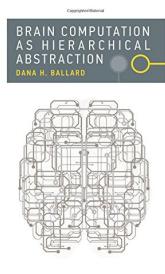


Bibliografía complementaria



“wetware”

- Dana H. Ballard: **Brain Computation as Hierarchical Abstraction.** MIT Press, 2015. ISBN 0262028611
- Olaf Sporns: **Discovering the Human Connectome.** MIT Press, 2012. ISBN 0262017903
- Olaf Sporns: **Networks of the Brain.** MIT Press, 2010. ISBN 0262014696
- Jeff Hawkins: **On Intelligence.** Times Books, 2004. ISBN 0805074562

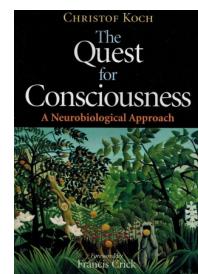
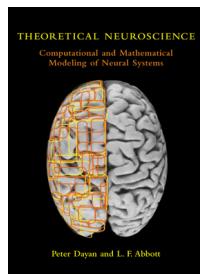
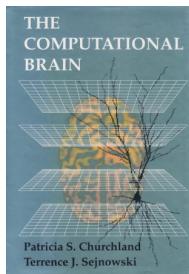


Bibliografía complementaria



Computational Neuroscience

- Patricia S. Churchland & Terrence J. Sejnowski: **The Computational Brain**. MIT Press, 1992. ISBN 0262031884
- Peter Dayan & L.F. Abbott: **Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems**. MIT Press, 2001. ISBN 0262041995.
- Christof Koch: **The Quest for Consciousness: A Neurobiological Approach**. Roberts & Company Publishers, 2004. ISBN 0974707708



Bibliografía



Bibliografía en castellano

- James A. Freeman & David M. Skapura: **Redes Neuronales: Algoritmos, aplicaciones y técnicas de programación** Addison-Wesley / Díaz de Santos, 1993. ISBN 020160115X

