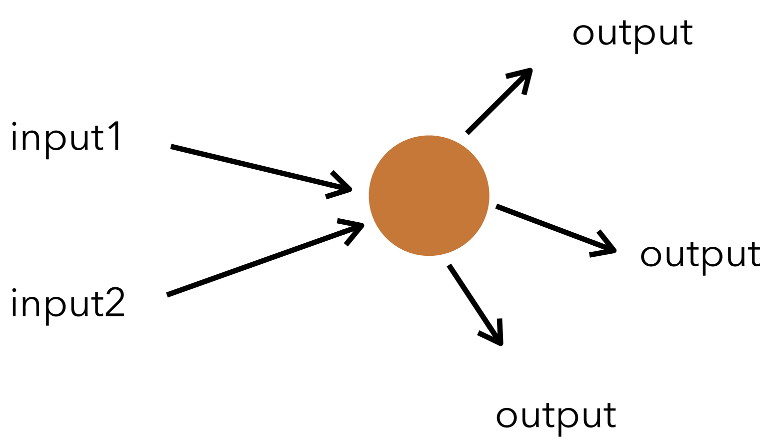
Arquitectura Básica de Redes Neuronales

Modelos de Neurona Artificial

Neuronas

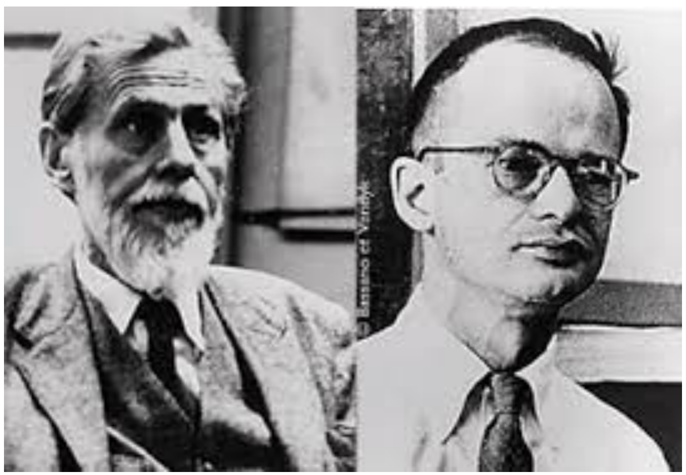
- La neurona es el componente básico de una red neuronal.
- Cada **neurona** (excepto las de entrada) implementa una regla de cómputo que que determina el ***estado o activación*** de la neurona. Ese estado **se propaga** a otras neuronas del grafo.



- La especificación matemática de esta regla de cálculo determina el **modelo de neurona** artificial adoptado en la red.

Modelo de McCulloc & Pitts (1943)

- Hacia 1943, Warren S. McCulloch & Walter Pitts publican el paper *"A logical calculus of the ideas immanent in nervous activity"*, formalizando un modelo de neurona que, con algunas modificaciones, es el modelo más utilizado actualmente.



Bulletin of Mathematical Biology Vol. 52, No. 1/2, pp. 99–115, 1990.
Printed in Great Britain.

0092–8240/90$3.00+0.00
Pergamon Press plc
Society for Mathematical Biology

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

■ WARREN S. MCCULLOCH AND WALTER PITTS
University of Illinois, College of Medicine,
Department of Psychiatry at the Illinois Neuropsychiatric Institute,
University of Chicago, Chicago, U.S.A.

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

Modelo de McCulloc & Pitts (1943)

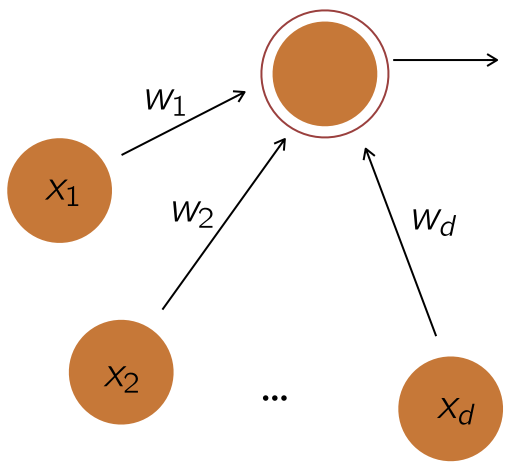
- Si denotamos por $x_1, x_2, \ldots, x_d$ los datos que la neurona recibe en un determinado instante de tiempo, y por $f(x)$ la respuesta o estado de la neurona después de recibir esta información, el modelo MP se presenta hoy en día como la siguiente ecuación

$$f(x) = g\left(\sum_{i=1}^{d} w_i x_i - b\right)$$

donde g(a) es la **función de Heaviside** o "escalón unitario", b es un parámetro característico de la neurona, denominado **umbral de excitación** y $w_1, w_2, \ldots, w_d$ denotan los **pesos de conexión** de la neurona con otras neuronas del grafo (de quienes recibe información).

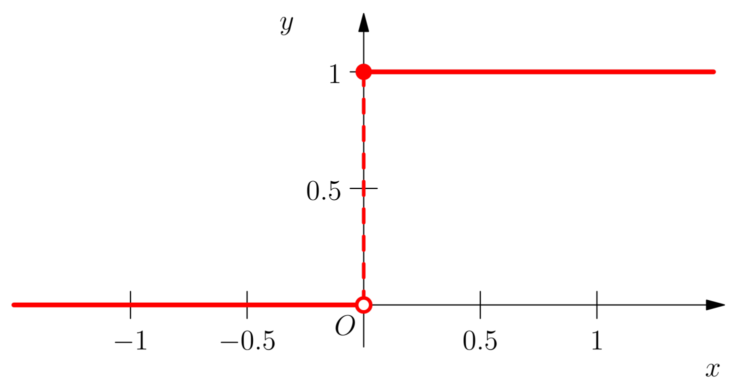
Modelo de McCulloc & Pitts (1943)

- Gráficamente:



$$f(x) = g\left(\sum_{i=1}^{d} w_i x_i - b\right)$$

$$g(a) = \begin{cases} 1 & a \geq 0 \\ 0 & a < 0 \end{cases}$$



Interpretación

- La neurona de McCulloch & Pitts primero **suma** la información que recibe, multiplicando cada dato por un peso característico de la conexión desde la que esa información llega. Si el estímulo final (suma ponderada) supera el **umbral de excitación** de la neurona, el resultado es 1. En otro caso, el resultado es 0.

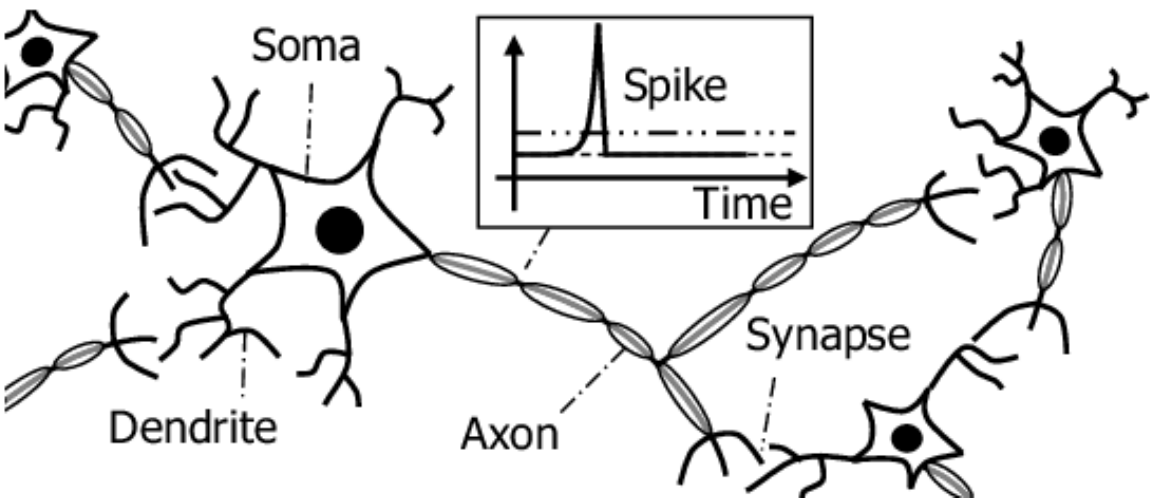
$$f(x) = g\left(\sum_{i=1}^{d} w_i x_i - b\right)$$

- Los pesos pueden ser **positivos o negativos** para representar conexiones excitatorias o inhibitorias.

Neuronas MP versus Neuronas Biológicas

- El modelo de McCulloch & Walter Pitts es consistente con algunas observaciones que se habían hecho hacia esos tiempos sobre el funcionamiento de las neuronas biológicas.

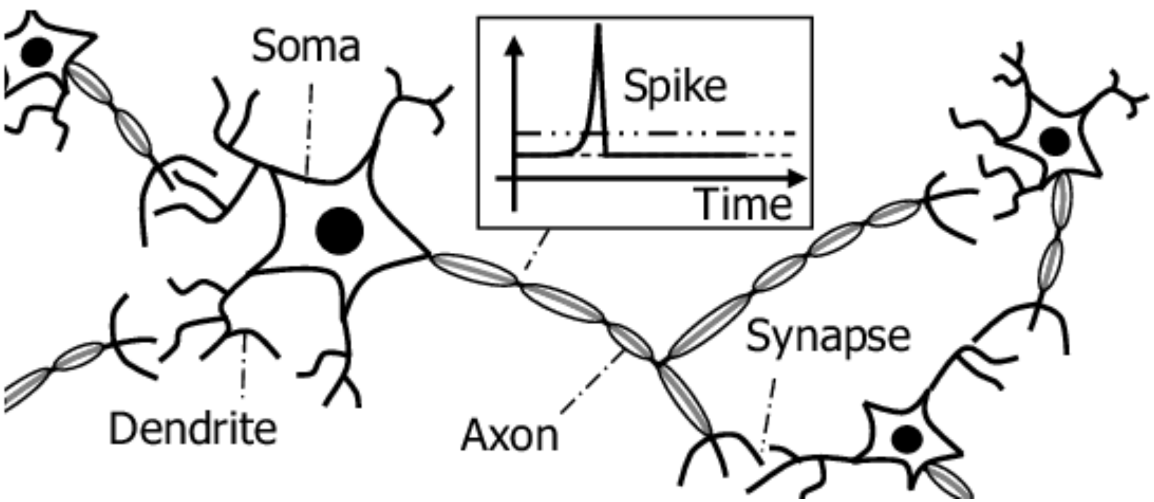
    1. **Efecto "todo o nada".** En ocasiones la neurona transmite una señal a través de su axón hacia otras neuronas y en otras no.



Neuronas MP versus Neuronas Biológicas

- El modelo de McCulloch & Walter Pitts es consistente con algunas observaciones que se habían hecho hacia esos tiempos sobre el funcionamiento de las neuronas biológicas.

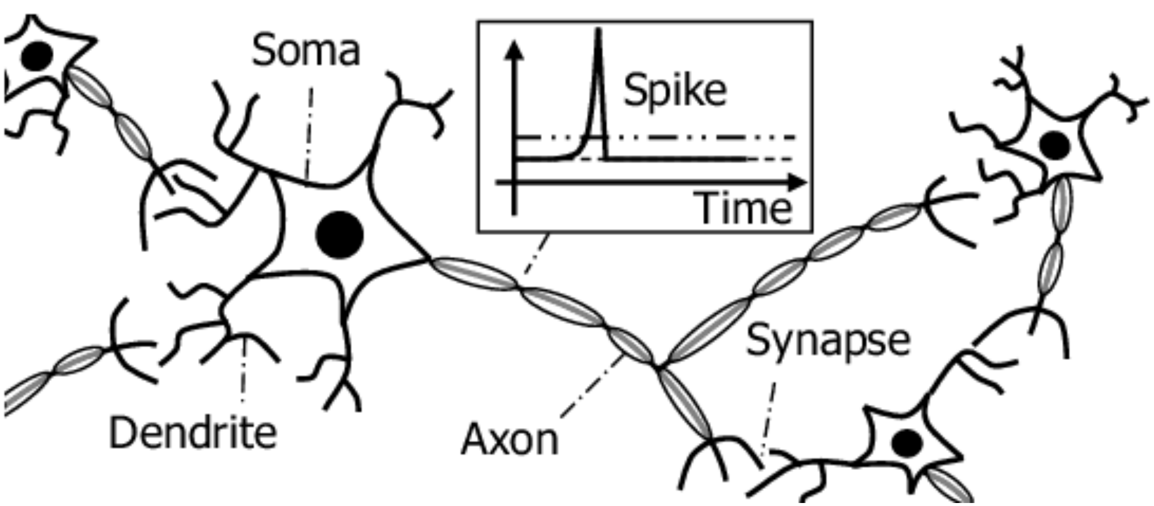
  2. **Conexiones excitatorias e inhibitorias.** La presencia de ciertos estímulos puede inhibir la reacción electro-química de la neurona o estimularla.



Neuronas MP versus Neuronas Biológicas

- El modelo de McCulloch & Walter Pitts es consistente con algunas observaciones que se habían hecho hacia esos tiempos sobre el funcionamiento de las neuronas biológicas.

3. **Efecto acumulativo.** La respuesta de una neurona depende de todos los estímulos a su alrededor, de manera de una mayor presencia de estímulos excitatorios aumenta la probabilidad de observar la transmisión de una señal hacia otras neuronas.

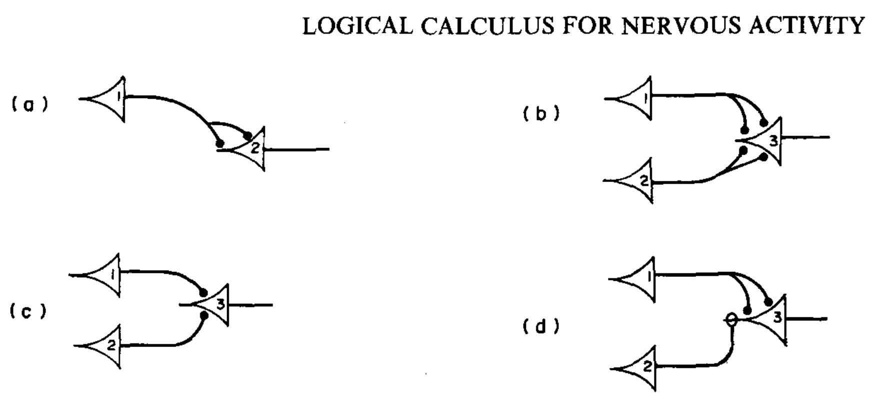


Neuronas MP versus Neuronas Biológicas

- Hoy en día, el modelo de McCulloch & Pitts se considera un **modelo muy poco realista de una neurona biológica** (aún así funciona muy bien en redes neuronales artificiales, que es lo que nos interesa).
- Uno de los aspectos más cuestionados ya en tiempos de su introducción fue el considerar un **modelo binario para la respuesta**: la neurona puede activarse (f(x) = 1) o no activarse (f(x) = 0).
- Como es *esa* la respuesta que se propaga a través de las red hacia otras neuronas, el modelo trata también los datos o estímulos que recibe cada neurona como binarios $x_i \in \{0, 1\} \, \forall i = 1, \ldots, d$.

Neuronas MP versus Neuronas Biológicas

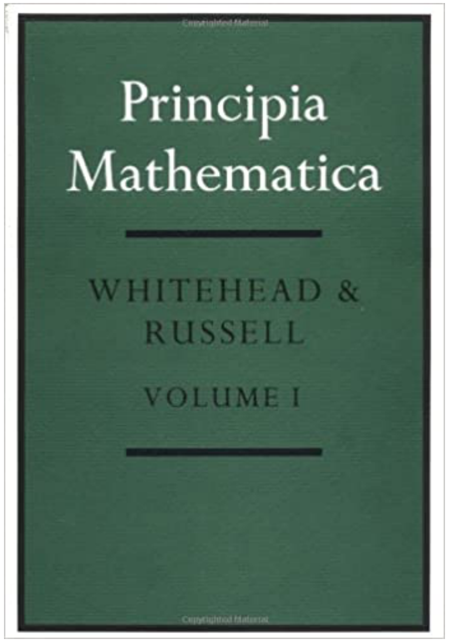
- Posiblemente, esta preferencia por **inputs binarios y** un **outputs binarios** se deba, más que a un intento de los autores por reproducir antecedentes biológicos disponibles en esos tiempos, a una fuerte convicción de que el cerebro es una **máquina basada en la lógica**.



(Redes neuronales expuestas en el paper de 1943)

Contexto del Modelo MP

- Recordemos que "pocos años antes" (1910), Russell & Whitehead habían publicado "Principia Mathematica", donde esencialmente sugieren que toda la matemática puede ser construida sobre el formalismo de la lógica (allí descrita).

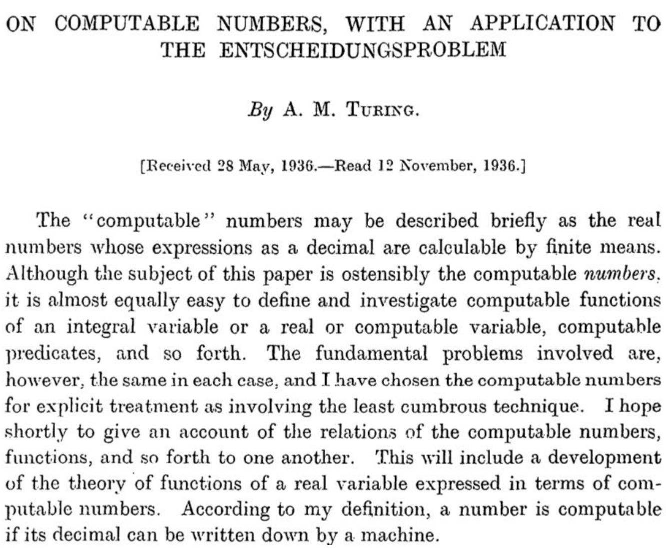


- Este texto fue determinante no sólo en el desarrollo de la matemática, sino también de la computación y la inteligencia artificial.

(Se cree que Pitts, conocido autodidacta, estudió el texto entre los 12 y los 15 años).

Contexto del Modelo MP

- En 1936, Turing había publicado *"On Computable Numbers with an Application to the Entscheidungs problem",* trabajo en que formaliza la idea de computabilidad, y define una máquina (abstracta) capaz de **leer y ejecutar instrucciones**, es decir **capaz de simular otras máquinas de cómputo.**



ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

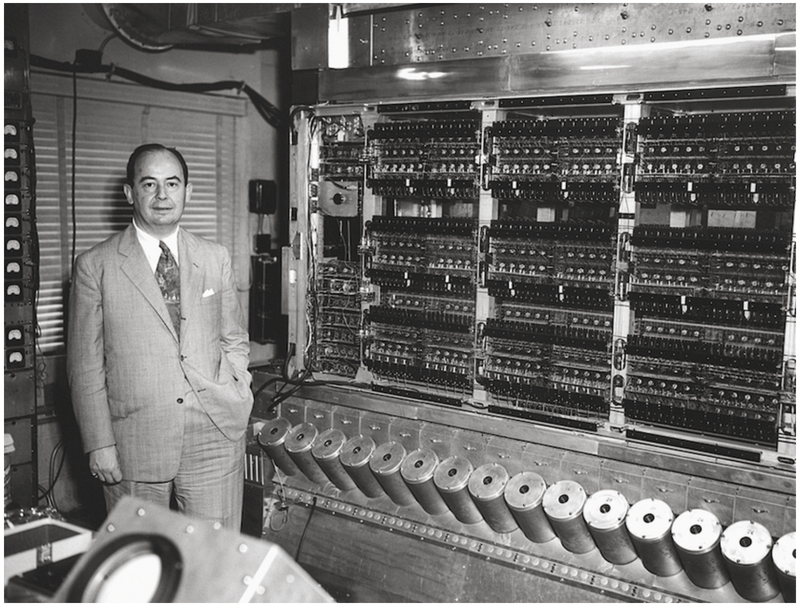
[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

- La máquina de Turing demuestra la existencia (teórica) de máquinas de cómputo universales, idea que se materializa más tarde (1945) en los computadores programables modernos.

Contexto del Modelo MP

- Es interesante mencionar que hacia 1945 (2 años después de M&C), Von Neumman crea el EDVAC, el primer computador digital programable en el sentido moderno (ENIAC debía re-cablearse físicamente).

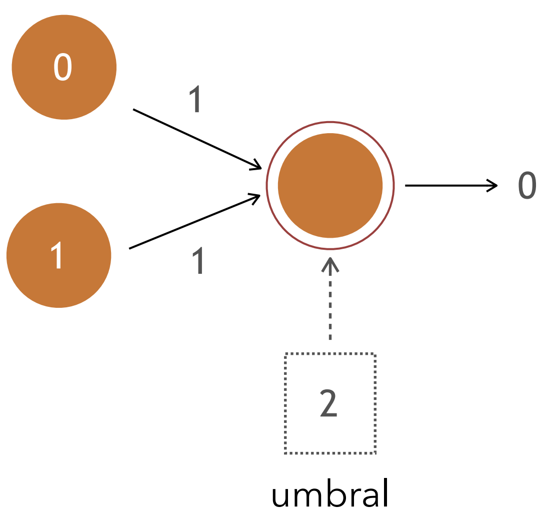


Se cree que las ideas de Pitts fueron muy influyentes en el diseño de esta máquina, incluso en la implementación de la idea de memoria externa (el paper de M&C es la única referencia en al paper de Von Neumman).

Cerebro como Máquina Lógica

- Todo este contexto consolida en Pitts la idea de que el cerebro no es más que una **máquina de cómputo universal basada en la lógica** (binaria). Para entender (o reproducir) su funcionamiento es suficiente considerar neuronas binarias que puedan reproducir las operaciones lógicas fundamentales.

AND

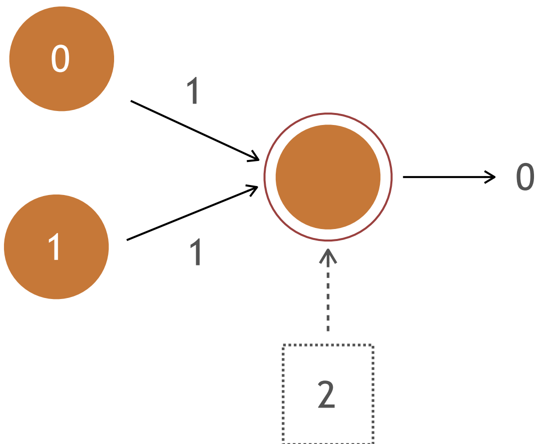


OR?

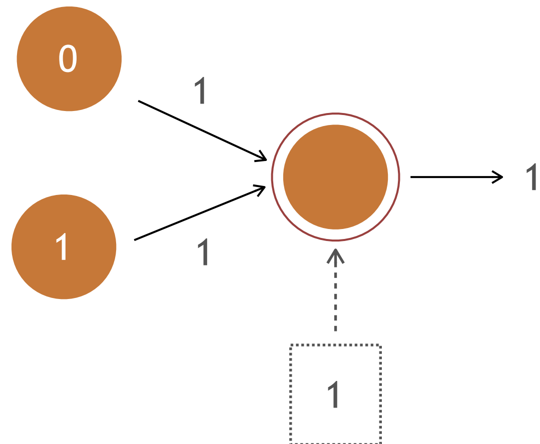
NOT?

Cerebro como Máquina Lógica

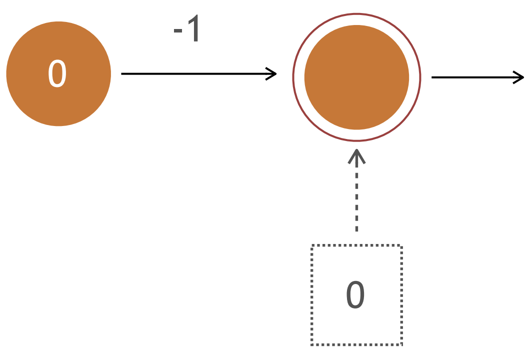
AND



OR



NOT



Ejercicio

- Construya una red neuronal que implemente la operación XOR (función de paridad binaria)

AND

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

OR

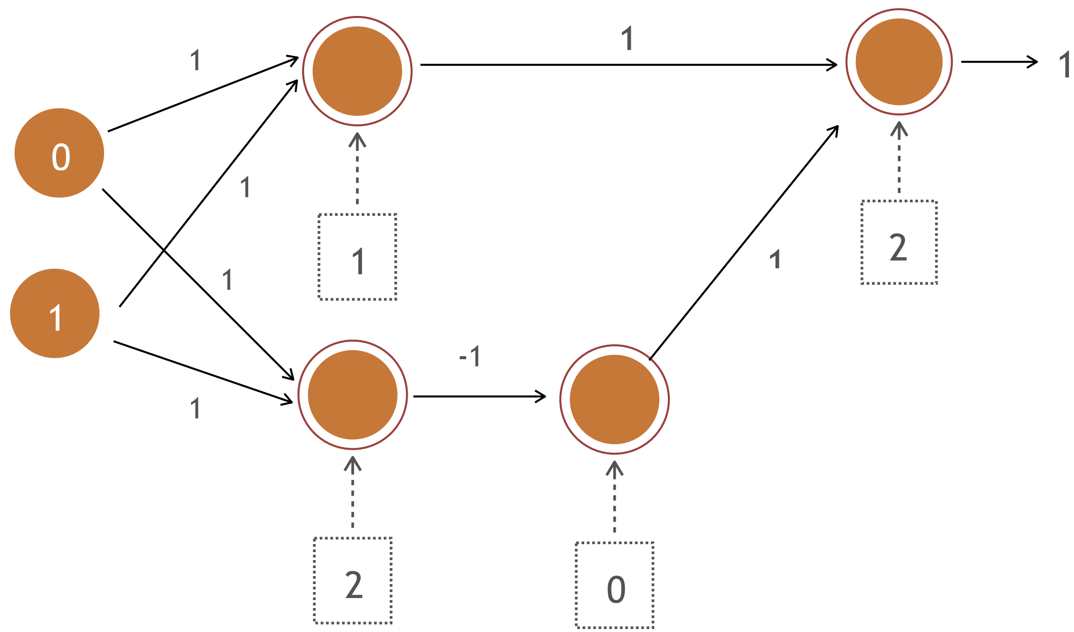
x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

XOR

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

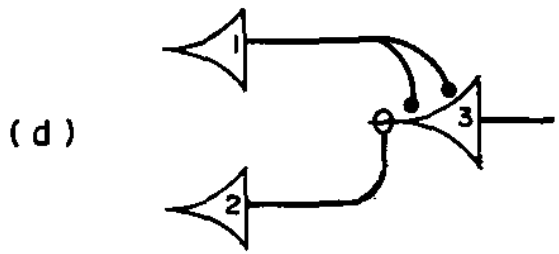
Universalidad del Modelo MP

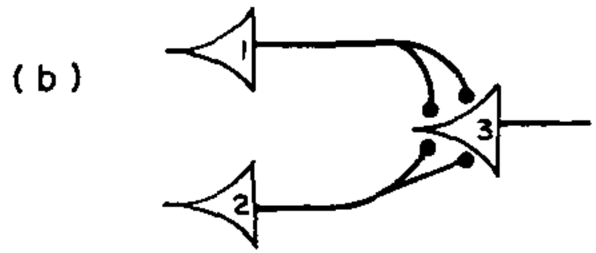
- **Conclusión:** las neuronas MP son funcionalmente completas en$\{0,1\}^2$
- Mostraremos más tarde que las neuronas MP son completas en $\{0,1\}^d$, y por lo tanto permiten implementar cualquier circuito digital, que es básicamente lo que quiere demostrar Pitts en el paper de 1943.



Circuitos de Neuronas

- La notación circuit-friendly en e paper de 1943 es evidente:





Los pesos se representan como multiplicaciones de la conexión a una neurona.

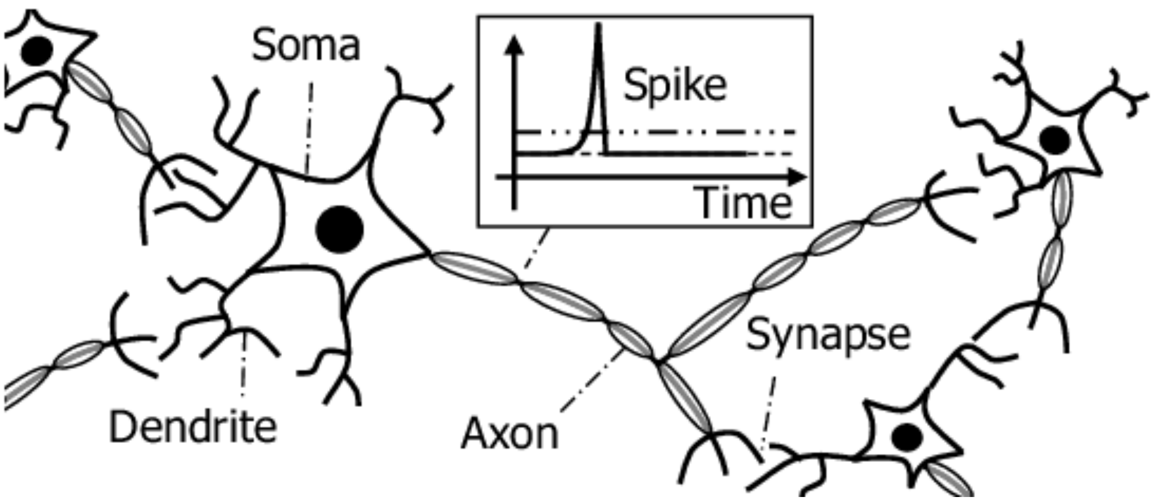
Las conexiones inhibitorias se marcan con un anillo y las demás con un círculo negro.

Los pesos son siempre +1 ó -1.

En los dibujos el umbral de excitación es siempre 2.

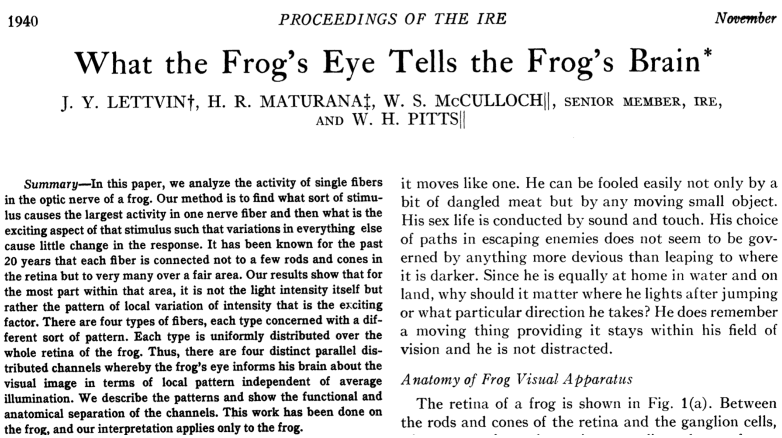
Cerebro como Máquina Lógica?

- Hoy es bastante claro que las neuronas biológicas no son dispositivos digitales sino analógicos: el potencial que trasmite la neurona no entrega información binaria (ausencia/presencia) sino análoga (frecuencia o fase de la señal).



Cerebro como Máquina Lógica?

- Uno de los primeros antecedentes contundentes acerca de esto fue entregado en el paper ***"What the Frog's Eye Tells the Frog's Brain"*** publicado en 1959 por Lettvin, Humberto Maturana (chileno), Pitts y McCulloch.



1940 *PROCEEDINGS OF THE IRE* *November*

What the Frog's Eye Tells the Frog's Brain*

J. Y. LETTVIN†, H. R. MATURANA‡, W. S. McCULLOCH||, SENIOR MEMBER, IRE, AND W. H. PITTS||

***Summary*—In this paper, we analyze the activity of single fibers in the optic nerve of a frog. Our method is to find what sort of stimulus causes the largest activity in one nerve fiber and then what is the exciting aspect of that stimulus such that variations in everything else cause little change in the response. It has been known for the past 20 years that each fiber is connected not to a few rods and cones in the retina but to very many over a fair area. Our results show that for the most part within that area, it is not the light intensity itself but rather the pattern of local variation of intensity that is the exciting factor. There are four types of fibers, each type concerned with a different sort of pattern. Each type is uniformly distributed over the whole retina of the frog. Thus, there are four distinct parallel distributed channels whereby the frog's eye informs his brain about the visual image in terms of local pattern independent of average illumination. We describe the patterns and show the functional and anatomical separation of the channels. This work has been done on the frog, and our interpretation applies only to the frog.**

it moves like one. He can be fooled easily not only by a bit of dangled meat but by any moving small object. His sex life is conducted by sound and touch. His choice of paths in escaping enemies does not seem to be governed by anything more devious than leaping to where it is darker. Since he is equally at home in water and on land, why should it matter where he lights after jumping or what particular direction he takes? He does remember a moving thing providing it stays within his field of vision and he is not distracted.

Anatomy of Frog Visual Apparatus

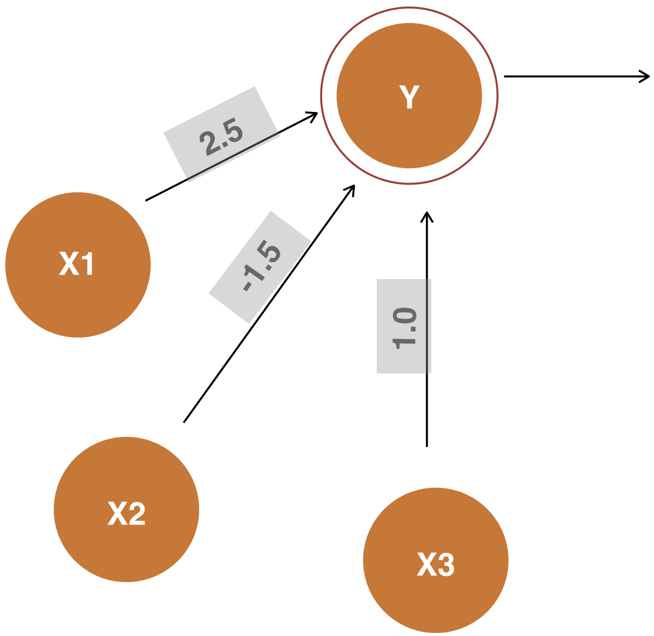
The retina of a frog is shown in Fig. 1(a). Between the rods and cones of the retina and the ganglion cells,

Cerebro como Máquina Lógica?

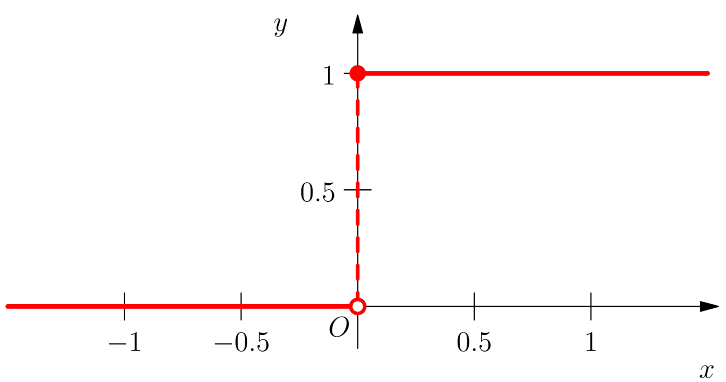
- El trabajo presenta evidencia de que las neuronas del sistema visual de la rana no transmiten información binaria al cerebro como pensaba Pitts sino información analógica altamente especializada (como posición de insectos en el campo visual del animal).

Generalización de la Neurona MP

- Hoy en día el modelo MP de neurona se suele introducir sin la restricción de inputs binarios y pesos discretos (+1,-1)

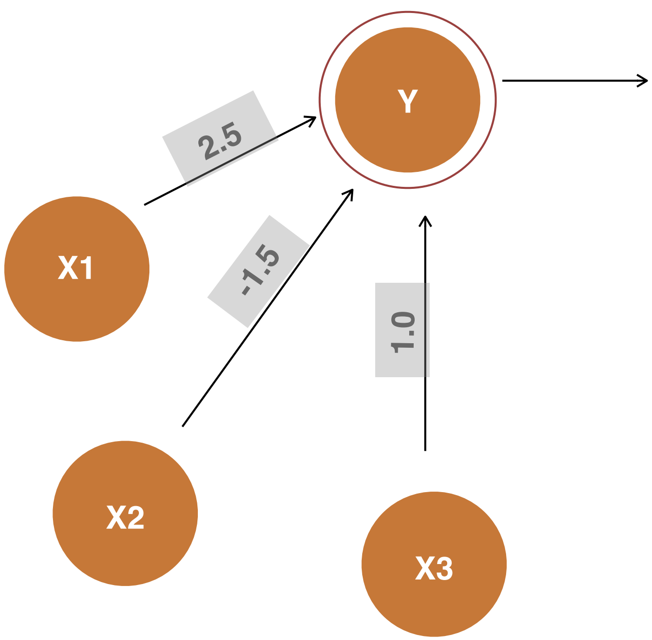


$$f(x) = g\left(\sum_{i=1}^{d} w_i x_i - b\right)$$



Función de Activación

- Es también usual hacer referencia a modelo MP sin la restricción de salidas binarias, generalización que se obtiene sustituyendo la función de Heaviside en el modelo original por otra función (típicamente no-lineal) denominada **función de activación** de la neurona.

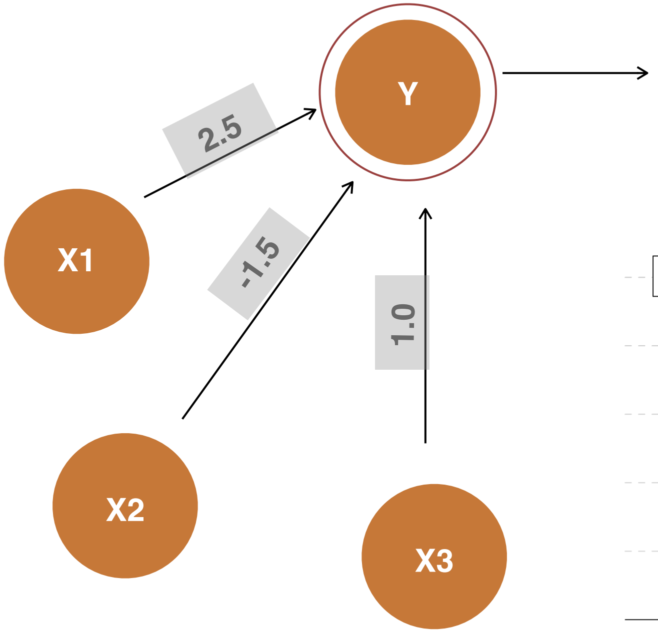


$$f(x) = g\left(\sum_{i=1}^{d} w_i x_i - b\right)$$

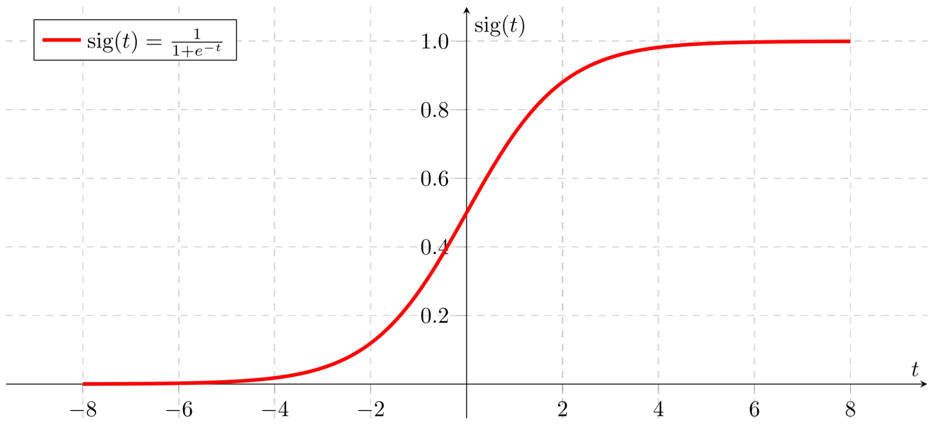
?

Neurona Sigmoidal

$$g(\xi) = \sigma(\xi) = \frac{1}{1 + e^{-x}}$$

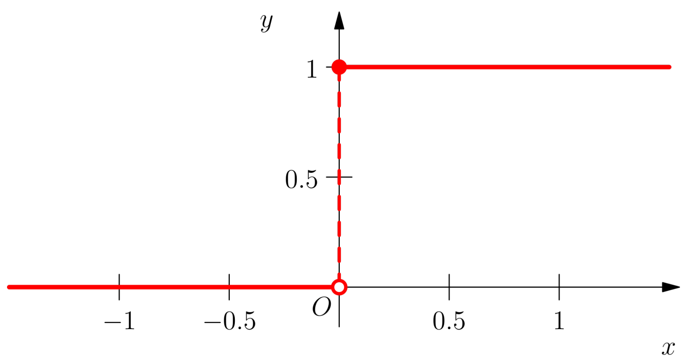


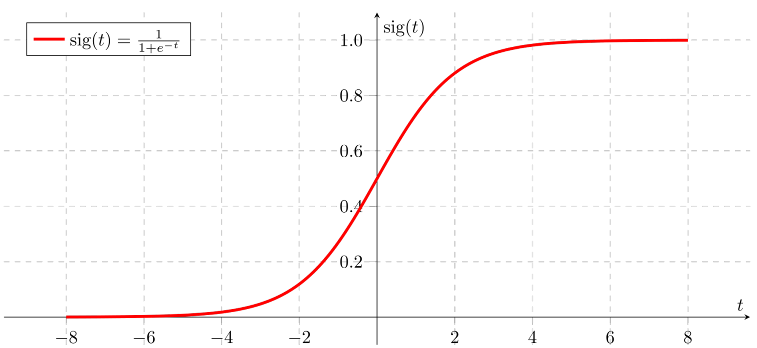
$$f(x) = g\left(\sum_{i=1}^{d} w_i x_i - b\right)$$



Neurona Sigmoidal

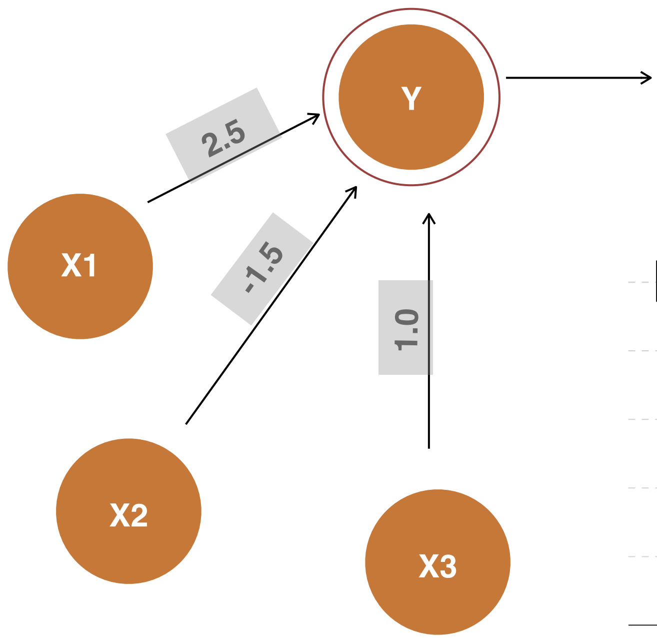
- Opción popular durante los años 80.
- Puede considerarse una aproximación suave (diferenciable) de la activación binaria clásica, que permite entrenar la neurona usando métodos basados en gradiente.



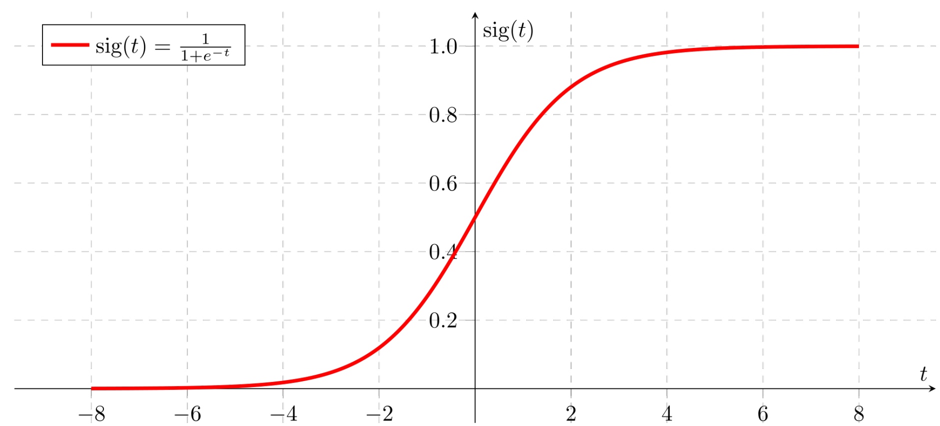


Neurona Sigmoidal

- Podemos también pensar la neurona sigmoidal como una neurona estocástica (Bernoulli) donde g() determina la probabilidad de activación. Explotaremos esto en capítulos más avanzados.



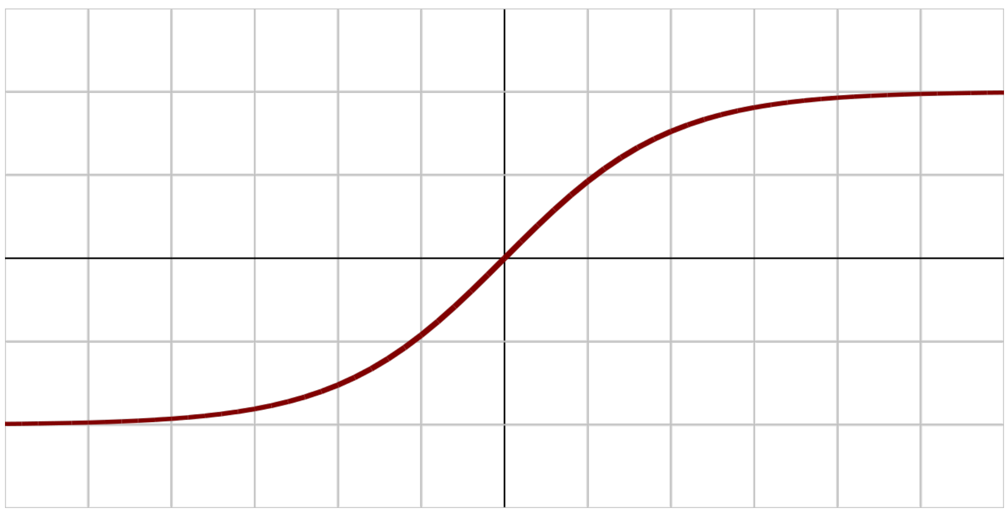
$$f(x) = g\left(\sum_{i=1}^{d} w_i x_i - b\right)$$



Neurona Tanh

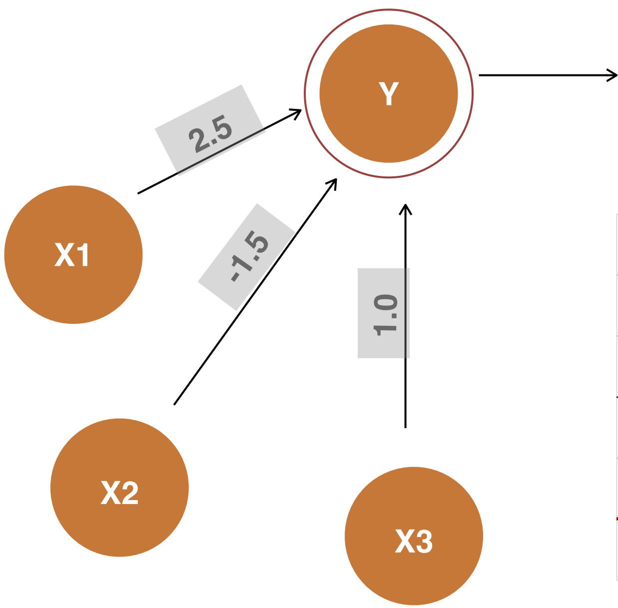
- Otra opción popularizada a fines de los años 80 es la función tangente hiperbólica

$$g(\xi) = \tanh(\xi) = \frac{2}{1 + e^{-2\xi}} - 1$$

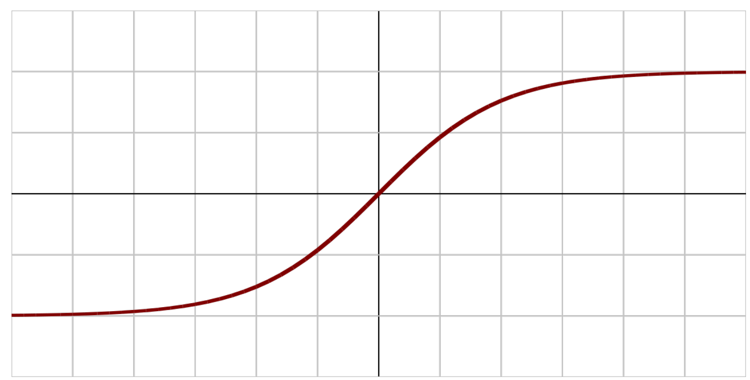


Neurona Tanh

- Se ha argumentado que al permitir activaciones positivas y negativas la unidad entrega mayor versatilidad a la red: con una neurona sigmoidal es necesario cambiar el signo de una conexión para entregar un valor negativo (la siguiente unidad en el forward pass y la misma en el backward pass). Con la neurona tanh el signo es función del estímulo.

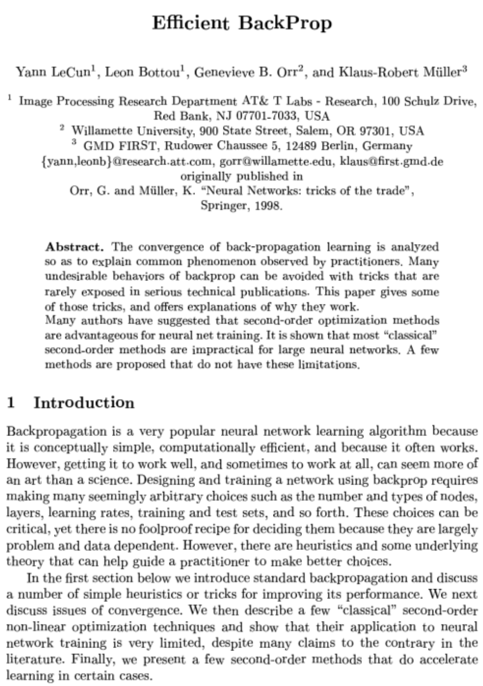


$$f(x) = g\left(\sum_{i=1}^{d} w_i x_i - b\right)$$



Neurona Tanh

- Como argumenta Yan LeCunn en "Efficient BackProp" (2002), es probable que la alta efectividad práctica de la neurona Tanh se deba también a la siguiente propiedad:



Efficient BackProp

Yann LeCun[1], Leon Bottou[1], Genevieve B. Orr[2], and Klaus-Robert Müller[3]

[1] Image Processing Research Department AT& T Labs - Research, 100 Schulz Drive, Red Bank, NJ 07701-7033, USA
[2] Willamette University, 900 State Street, Salem, OR 97301, USA
[3] GMD FIRST, Rudower Chaussee 5, 12489 Berlin, Germany
{yann,leonb}@research.att.com, gorr@willamette.edu, klaus@first.gmd.de

originally published in
Orr, G. and Müller, K. "Neural Networks: tricks of the trade", Springer, 1998.

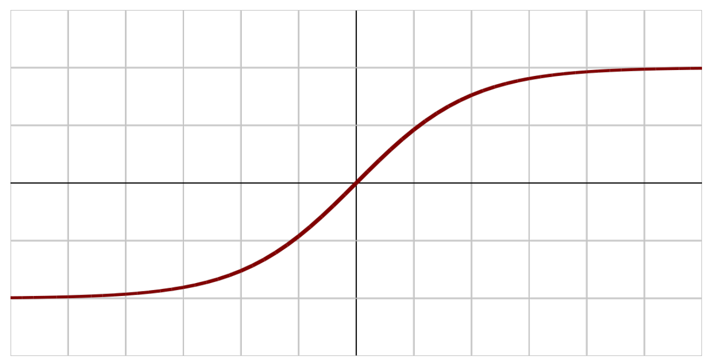
Abstract. The convergence of back-propagation learning is analyzed so as to explain common phenomenon observed by practitioners. Many undesirable behaviors of backprop can be avoided with tricks that are rarely exposed in serious technical publications. This paper gives some of those tricks, and offers explanations of why they work.
Many authors have suggested that second-order optimization methods are advantageous for neural net training. It is shown that most "classical" second-order methods are impractical for large neural networks. A few methods are proposed that do not have these limitations.

1 Introduction

Backpropagation is a very popular neural network learning algorithm because it is conceptually simple, computationally efficient, and because it often works. However, getting it to work well, and sometimes to work at all, can seem more of an art than a science. Designing and training a network using backprop requires making many seemingly arbitrary choices such as the number and types of nodes, layers, learning rates, training and test sets, and so forth. These choices can be critical, yet there is no foolproof recipe for deciding them because they are largely problem and data dependent. However, there are heuristics and some underlying theory that can help guide a practitioner to make better choices.

In the first section below we introduce standard backpropagation and discuss a number of simple heuristics or tricks for improving its performance. We next discuss issues of convergence. We then describe a few "classical" second-order non-linear optimization techniques and show that their application to neural network training is very limited, despite many claims to the contrary in the literature. Finally, we present a few second-order methods that do accelerate learning in certain cases.

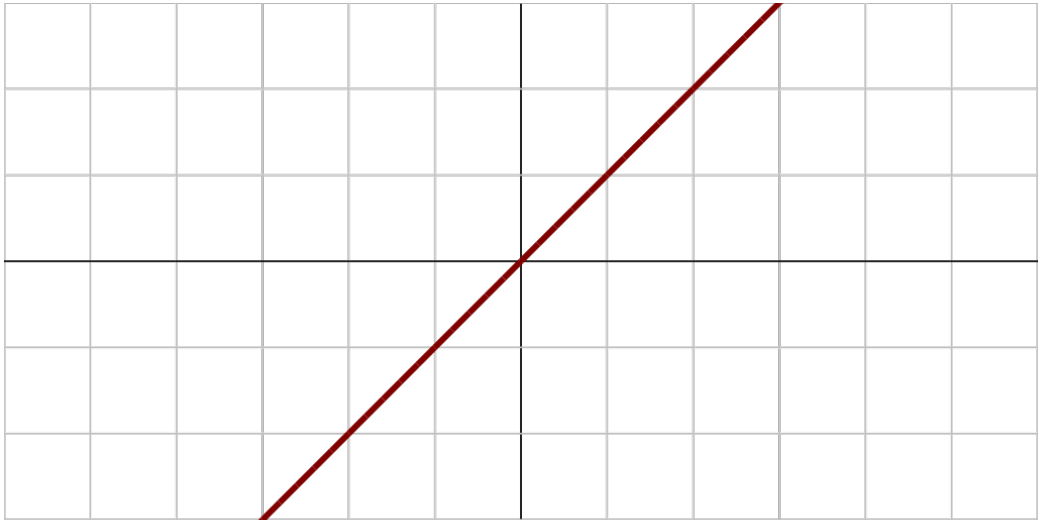
- Si la magnitud de los estímulos de entrada tiene valor esperado 0, la unidad tiende mantener esa propiedad para las neuronas que siguen hacia adelante en la red.



Neurona Lineal

- La neurona lineal tiende a usarse sólo en partes específicas de una red.

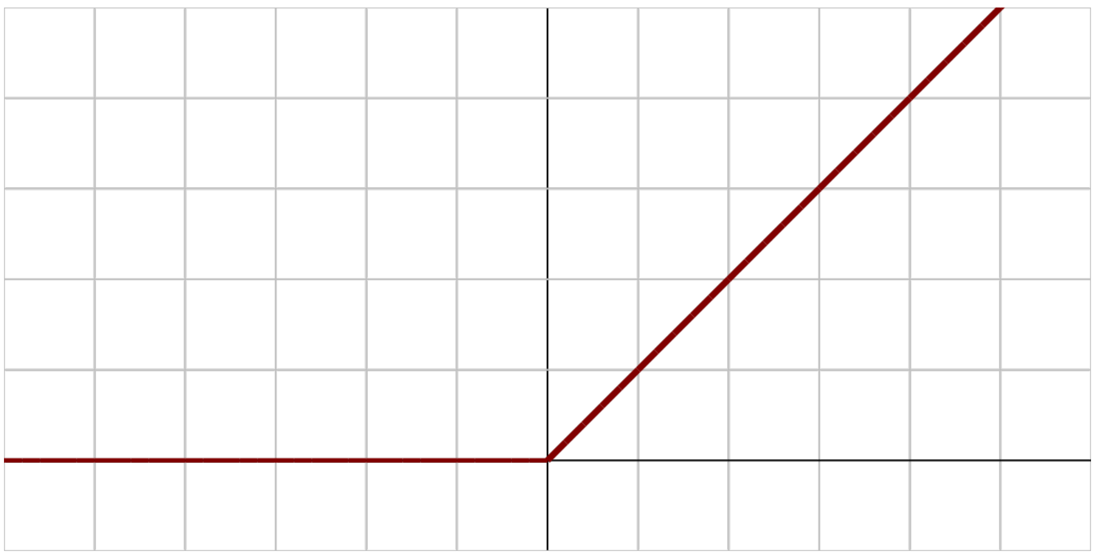
$$g(\xi) = \xi$$



Neurona Rectificadora

- Una opción popularizada hacia 2011 después de la publicación de Glorot et al. *"Deep Rectifier Linear Units"* es la **neurona ReLU**

$$g(\xi) = \begin{cases} \xi & \xi > 0 \\ 0 & \xi \leq 0 \end{cases} = \max(0, \xi)$$



Neurona Rectificadora

- Los autores argumentan que esta función de activación es una mejor aproximación de la respuesta que se observa en neuronas biológicas que la que proveen las alternativas populares a la fecha (sigmoid, tanh).

Deep Sparse Rectifier Neural Networks

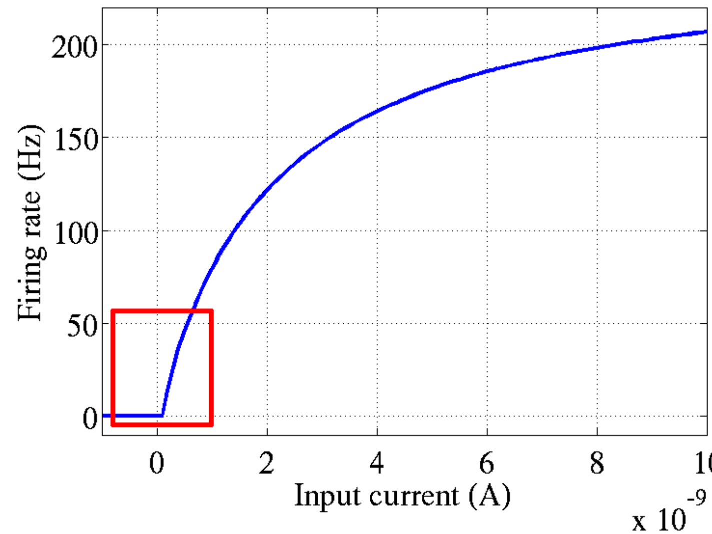
Xavier Glorot
DIRO, Université de Montréal
Montréal, QC, Canada
glorotxa@iro.umontreal.ca

Antoine Bordes
Heudiasyc, UMR CNRS 6599
UTC, Compiègne, France
and
DIRO, Université de Montréal
Montréal, QC, Canada
antoine.bordes@hds.utc.fr

Yoshua Bengio
DIRO, Université de Montréal
Montréal, QC, Canada
bengioy@iro.umontreal.ca

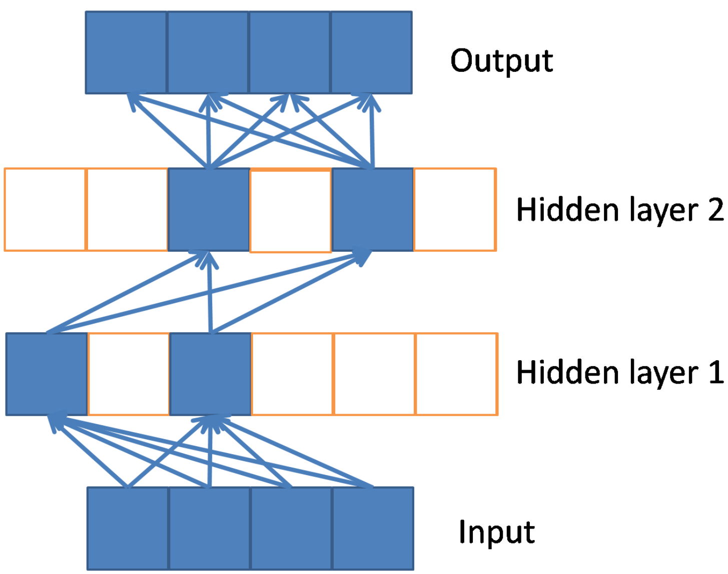
Abstract

While logistic sigmoid neurons are more biologically plausible than hyperbolic tangent neurons, the latter work better for training multi-layer neural networks. This paper shows that rectifying neurons are an even better model of biological neurons and yield equal or better performance than hyperbolic tangent networks in spite of the hard non-linearity and non-differentiability at zero, creating sparse representations with true zeros, which seem remarkably suitable for naturally sparse data. Even though they can take advantage of semi-supervised setups with extra-unlabeled data, deep rectifier networks can reach their best performance without requiring any unsupervised pre-training on purely supervised tasks with large labeled datasets. Hence, these results can be seen as because the objective of the former is to obtain computationally efficient learners, that generalize well to new examples, whereas the objective of the latter is to abstract out neuroscientific data while obtaining explanations of the principles involved, providing predictions and guidance for future biological experiments. Areas where both objectives coincide are therefore particularly worthy of investigation, pointing towards computationally motivated principles of operation in the brain that can also enhance research in artificial intelligence. In this paper we show that two common gaps between computational neuroscience models and machine learning neural network models can be bridged by using the following linear by part activation : $\max(0, x)$, called the rectifier (or hinge) activation function. Experimental results will show engaging training behavior of this activation function, especially for *deep architectures* (see Bengio (2009) for a review), i.e., where the number of hidden layers in the neural network is 3 or more.



Neurona Rectificadora

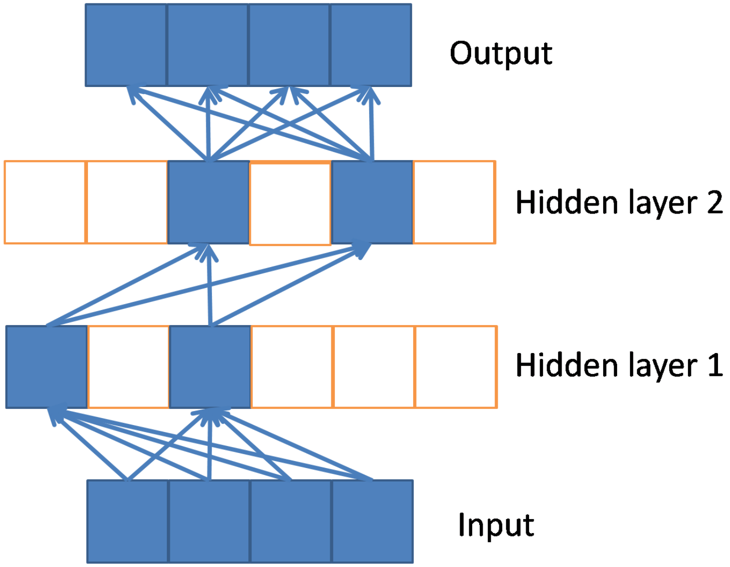
- Probablemente, aquello que hace tan efectiva a esta neurona en redes neuronales modernas (que más tarde llamaremos “profundas”) es que (su saturación exacta en 0) permite o induce patrones de activación dispersos en el grafo.



- Aunque la red sea muy “grande”, el número de unidades activas es pequeño ante un determinado dato de entrada. En otras palabras, la **complejidad efectiva** de modelo es menor que su complejidad aparente y puede calibrarse automáticamente según la complejidad de los datos por procesar.

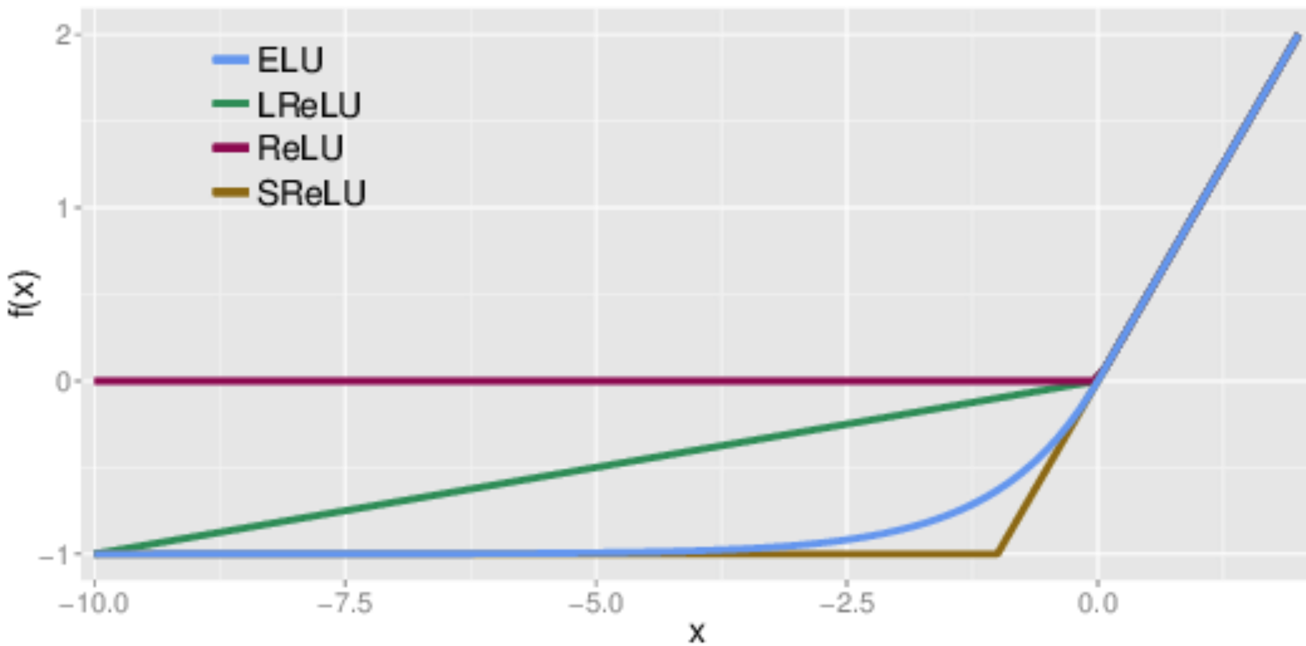
Neurona Rectificadora

- Limitar los canales por los que transmite una señal en la red ayuda también (como veremos más adelante) a evitar que la señal diverja durante el paso por el grafo hasta las neuronas de salida.



Variantes de la Rectificadora

- La efectividad de la neurona ReLU ha hecho proliferar variantes de este modelo en los últimos 10 años. La mayoría de las variantes rompe la propiedad de inducir activaciones dispersas en la red para evitar problemas que se tienden a ver durante el entrenamiento (dead units).



Entonces …

- Un modelo de neurona es una especificación matemática de la operación o cálculo que hace cada unidad del grafo con los datos que recibe.
- El modelo estándar hoy en día es (esencialmente) el modelo de McCulloch & Pitts: **la neurona computa una combinación lineal de los estímulos que recibe y aplica a eso una no-linealidad denominada función de activación.**
- Funciones populares son: sigmoid, tanh, ReLU y sus variantes.
- ReLU es considerada la opción por defecto en redes profundas.
- Pregunta: ¿Es suficiente especificar el modelo de neuronas para que la red esté completamente definida?
- Tarea: Implemente (programe) la red que implementa en XOR.