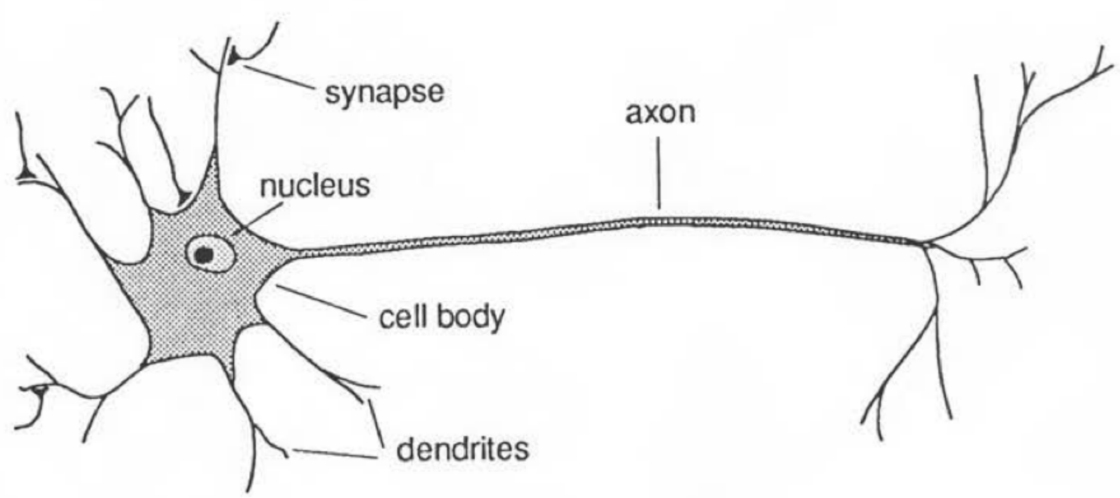
Redes Neuronales

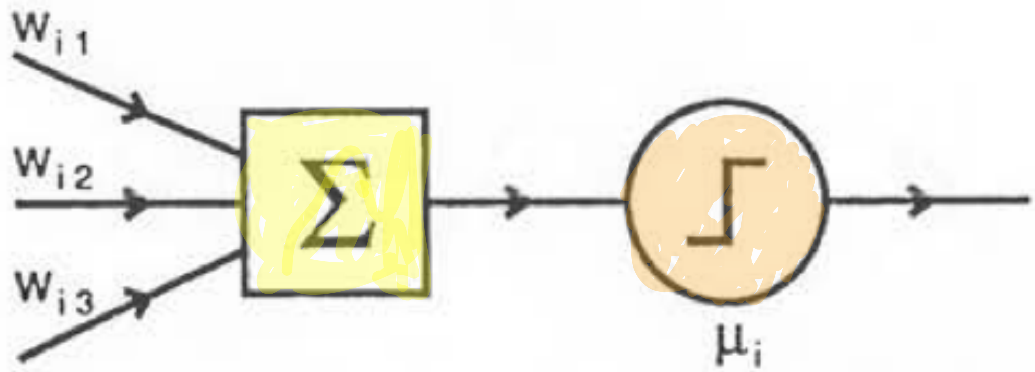
Parte 1: neuronas artificiales

Dado que una "RED NEURONAL" no es más que un conjunto de "NEURONAS ARTIFICIALES" ensambladas a través de una ARQUITECTURA de conexiones sinápticas también artificiales, es necesario comenzar hablando de estas últimas.

Ya vimos bastante sobre el funcionamiento y modelado dinámico de neuronas. Ahora tomaremos un camino diferente.

En lugar de procurar modelar los detalles, buscaremos modelar aquellos elementos escenciales (en nuestras creencias) para imaginar que un conjunto grande de neuronas interactuando pueden eventualmente adquirir capacidades computacionales similares a las que observamos en sistemas naturales, como son los sistemas nerviosos del reino animal.





Vamos a quedarnos con algunos elementos biológicos.

i) las sinapsis serán modeladas con números reales

$$W_{ij} \begin{cases} \text{sinapsis entre la} \\ \text{neurona presináptica } j \\ \text{a postsináptica } i \end{cases}$$

$W_{ij} > 0$ sinapsis excitatoria

$W_{ij} < 0$ sinapsis inhibitoria

$W_{ij} = 0$ ausencia de sinapsis

$|W_{ij}|$ medida de la eficiencia de la sinapsis

2) suponemos que en la región entre el cuerpo y el axón la neurona suma las contribuciones que le llegan a través de todos sus dendritos, para definir una medida del INPUT al que llamamos h.
Este input h eventualmente puede recibir también una contribución externa.

h emula al potencial de membrana

3) suponemos que la neurona artificial es una UNIDAD DE UMBRAL, o sea, que si h es mayor que cierto valor μ, dispara. Sino, no dispara.

4) Sólo nos interesa saber si dispara y cuándo dispara, y no modelar la forma de la espiga de potencial de acción.

1943

A logical calculus of the ideas immanent in nervous activity

Warren McCulloch y Walter Pitt

$n_i(t)$ describe el estado de la neurona i en el tiempo t.

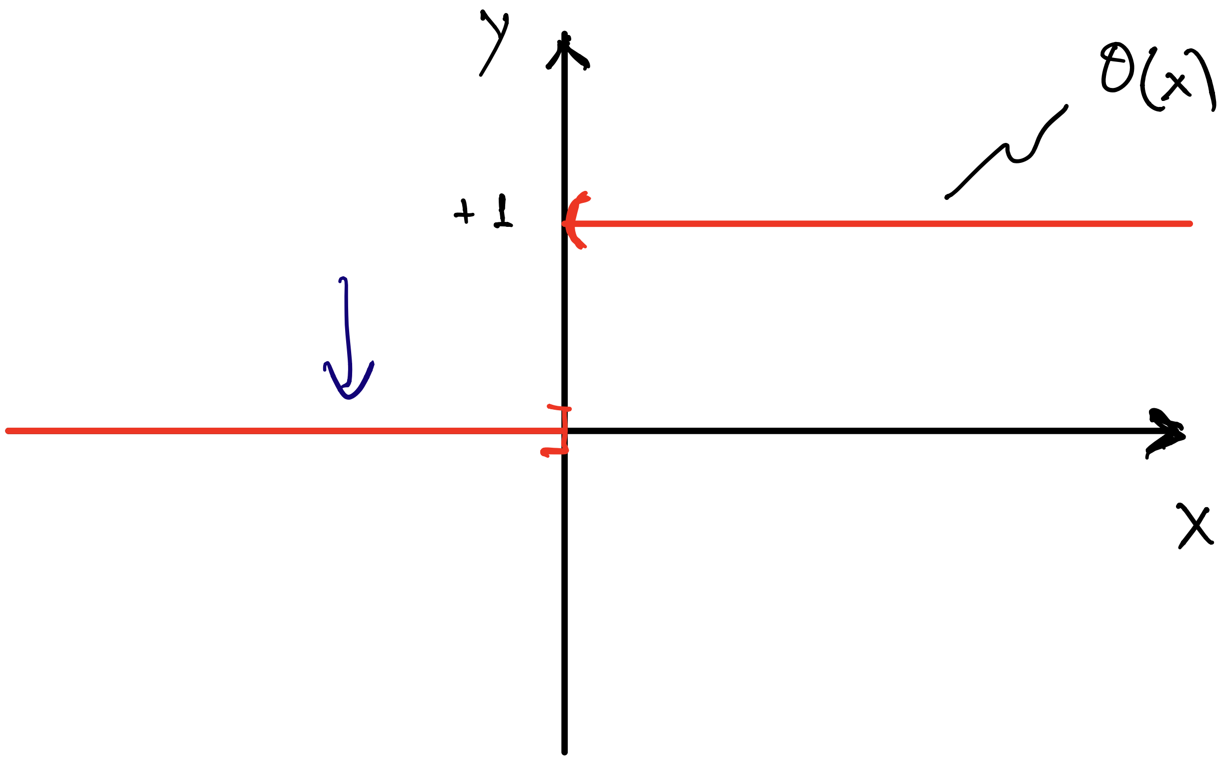
$$n_i(t) : \begin{cases} 0 & \text{si no dispara} \\ 1 & \text{si dispara} \end{cases}$$

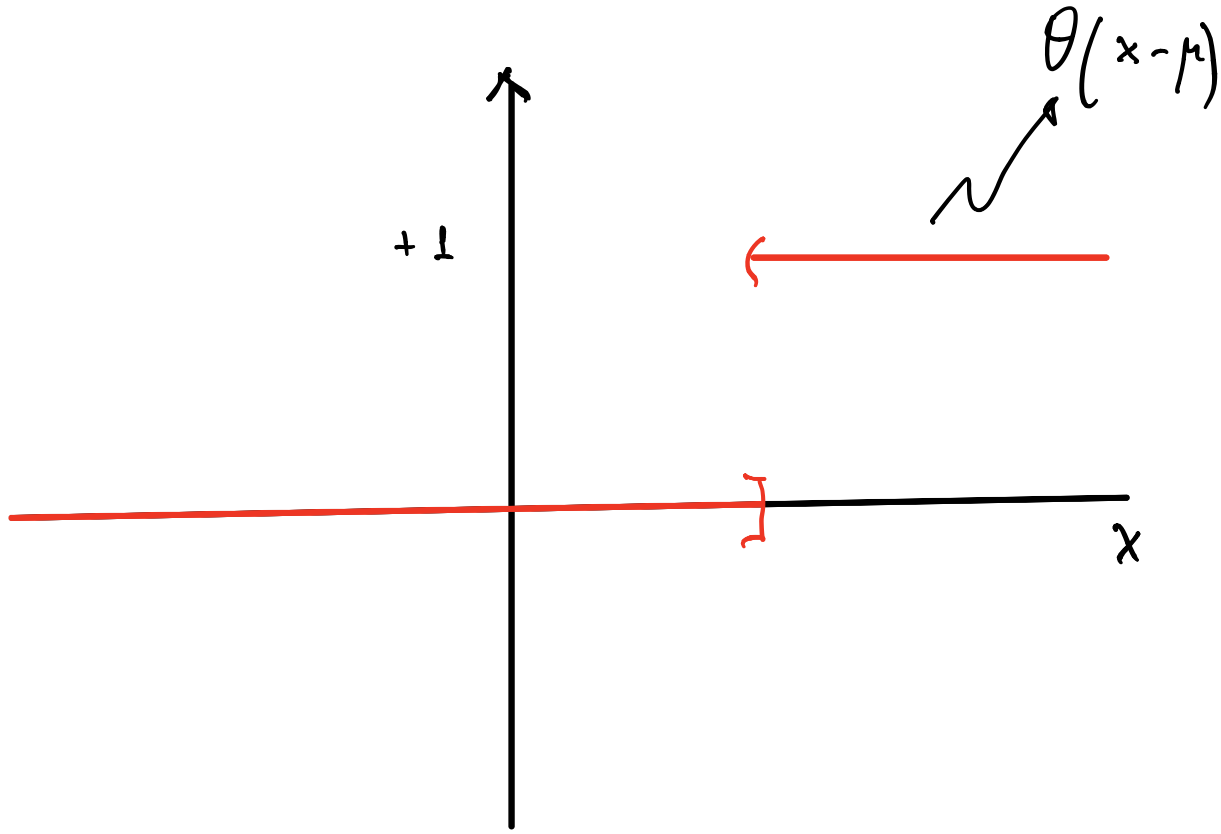
$$n_i(t+\Delta t) = \Theta\left(h_i(t) - \mu_i\right)$$

$$h_i(t) = \sum_{j \in V_j} \omega_{ij} n_j(t)$$

V_j: conjunto de los presinápticos

donde
$$\Theta(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$





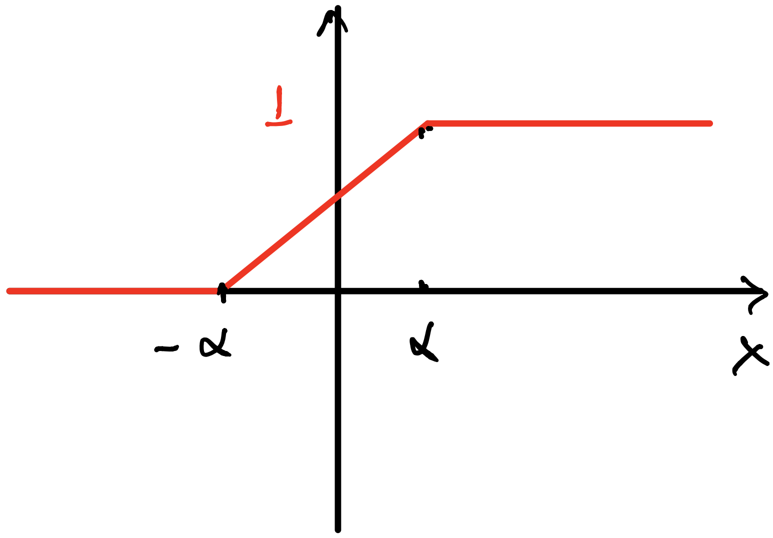
Esta neurona de McCulloch-Pitts anda muy bien. Volveremos sobre ella.

Ahora podemos generalizar la neurona, cambiando Θ por otras

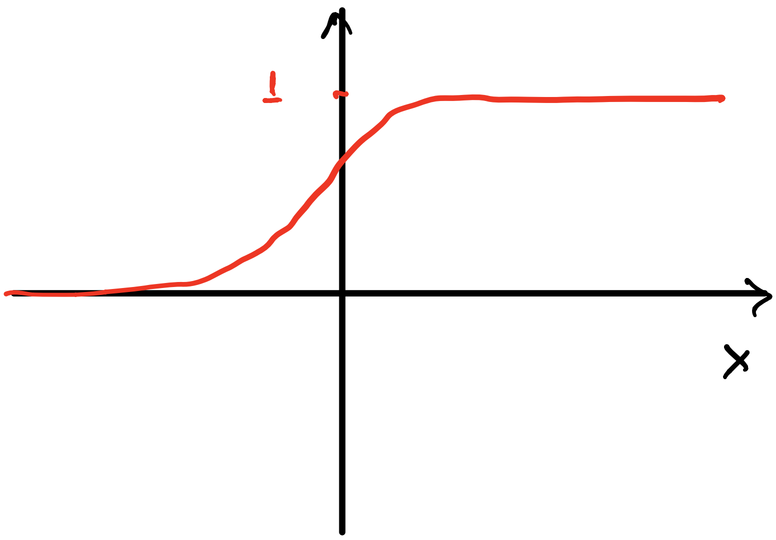
$$n_i(t+\Delta t) = g\left(h_i(t) - \mu_i\right)$$

Noten que el Tiempo es discreto

Continua a Trozos



Sigmoidea

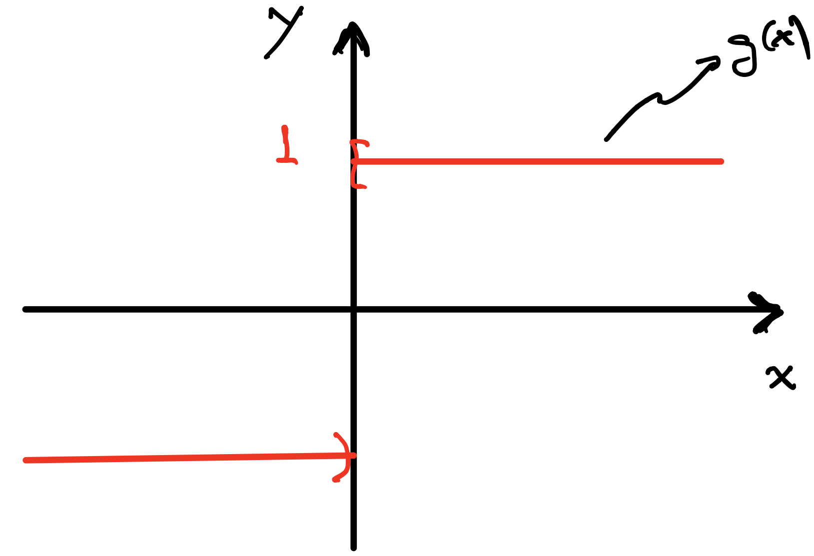


$$g(x) = \frac{1}{1 + e^{-\beta x}}$$

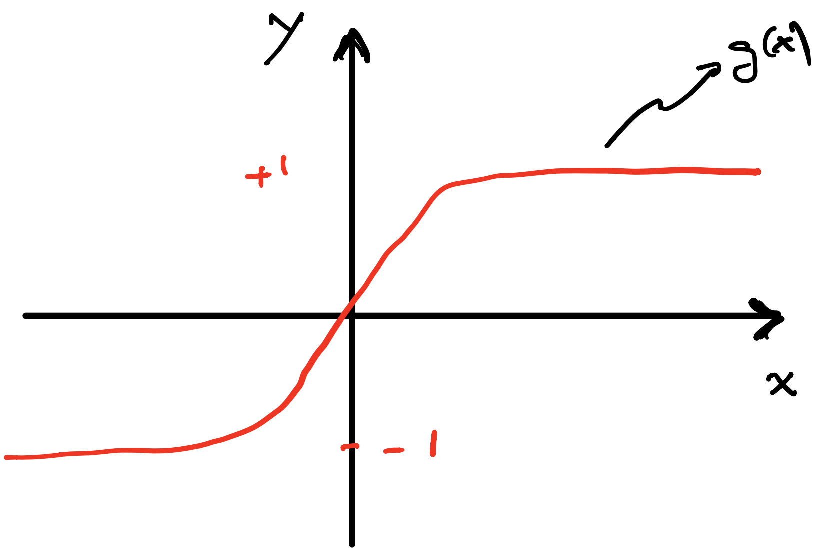
$\beta > 0$

Podemos también trabajar con ± 1,

$$n_i(t) = \begin{cases} +1 & \text{si dispara} \\ -1 & \text{si no dispara} \end{cases}$$



$$g(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x \leq 0 \end{cases}$$

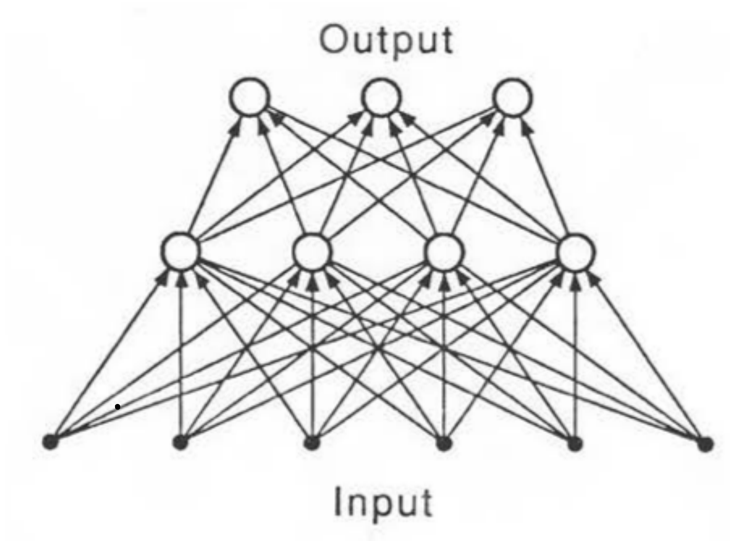


$$g(x) = \tanh(\beta x)$$

$$= \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}$$

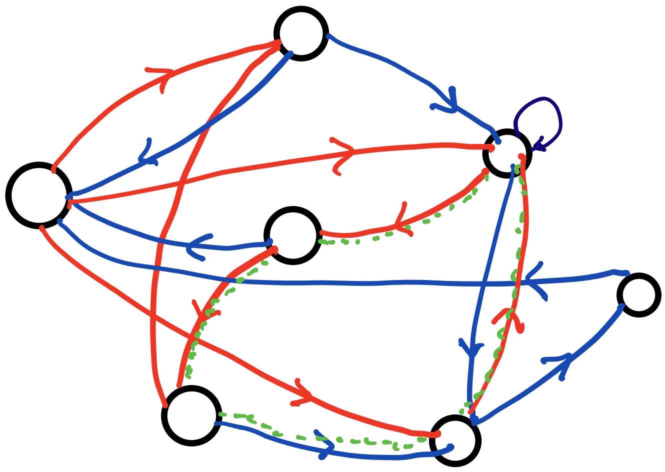
Ahora podemos comenzar a armar redes neuronales ensamblando neuronas artificiales. Hay muy variadas arquitecturas, pero nosotros las clasificamos en esta etapa en dos tipos

Redes feed_forward



Noten que las neuronas se organizan jerárquicamente en "CAPAS" y que las sinapsis son de las neuronas de una capa a las de las siguientes. No vuelve la información hacia atrás

Redes Recurrentes



No hay jerarquías en la disposición. Se pueden formar loops. Todas las neuronas representan el input y todas representan el output.

Más adelante veremos que la topología de las arquitecturas, tanto en redes feed-forward como en recurrentes, es muy importante para el desempeño de la red.

Parte 2: el problema de memoria asociativa

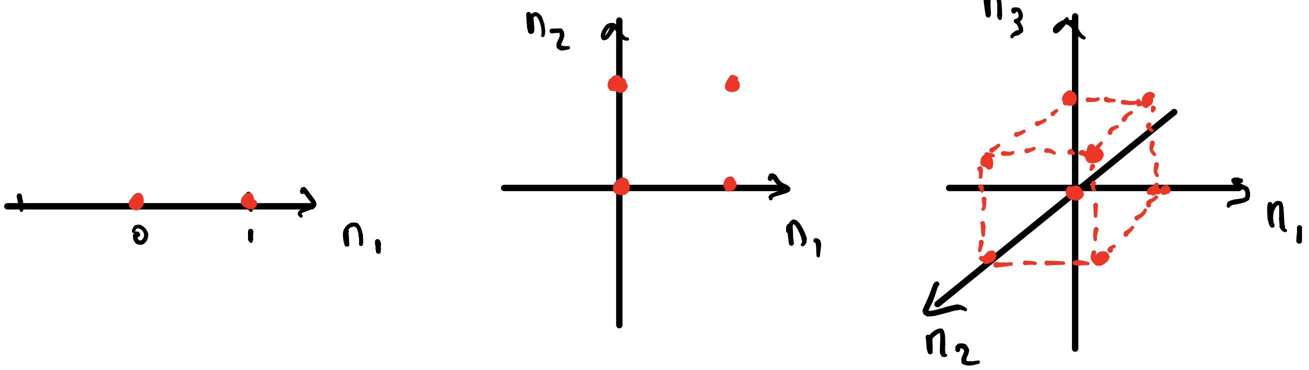
Comencemos considerando una red recurrente de N neuronas binarias $n_i(t)$, con $i = 1, 2, \dots N$. La podemos pensar como un numero binario

$$\left(n_1(t),\ n_2(t), \dots,\ n_N(t)\right)$$

Por ejemplo:

$$\left(1, 0, 0, 1, 0, \dots\ 1\right)$$

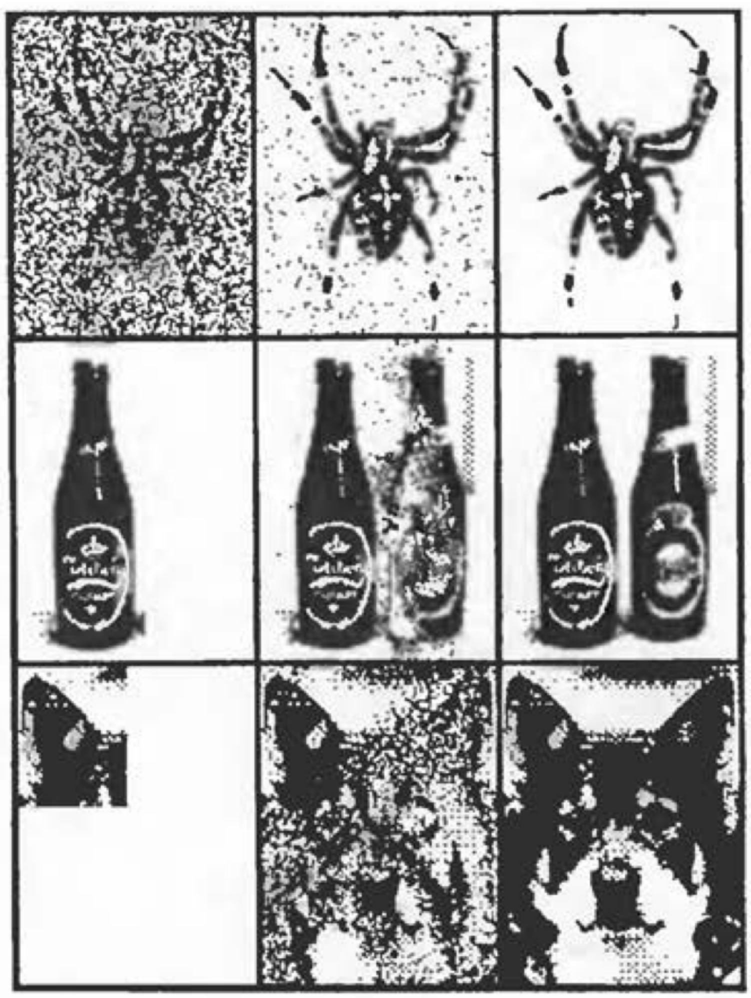
Podemos pensarlo como un vertice de un hipercubo en N dimensiones



Una red neuronal de este tipo puede pensarse como un caminante entre los vertices del hipercubo de dimensión N.

¿Qué esperamos que haga la red?

Que almacene y recupere información como nosotros



Por memoria asociativa entendemos la capacidad de ALMACENAR y RECUPERAR información por asociación con otra información.

Esto difiere mucho de una computadora convencional, las cuales almacenan por dirección.

Volvamos a nuestra red de N neuronas binarias 0 ó 1. Con ella tenemos

$$2^N$$

posibles configuraciones de disparos globales

$$
\begin{aligned}
2^N &= \exp(\ln(2^N)) \\
&= \exp(N \ln(2)) \\
&= e^{\ln(2) \cdot N} \approx e^{0{,}69 \cdot N}
\end{aligned}
$$

O sea, el número de posibles configuraciones crece exponencialmente.

Con

$N = 10$ hay 1024 configuraciones

$N = 100$ hay $\approx 1{,}26 \times 10^{30}$ conf.

$N = 1000$ hay $\approx 1{,}07 \times 10^{301}$ conf.

El universo tiene aproximadamente

$4{,}7 \times 10^{21}$ segundos

El universo tiene aproximadamente

1×10^{25} estrellas.

El modelo de HOPFIELD

John Joseph Hopfield

Nació en 1933

PhD in Physics, Cornell University, 1958

Trabajó en
- Bell Laboratories
- University of California
- Princeton University
- California Institute of Technology
- Princeton University

Su paper relevante para nosotros es

"Neural Networks and physical systems with emergent collective computational abilities"

J. J. Hopfield

Proceedings of the National Academy of Science USA

PANS 79 (8) pp 2554-2558 (1 April, 1982)

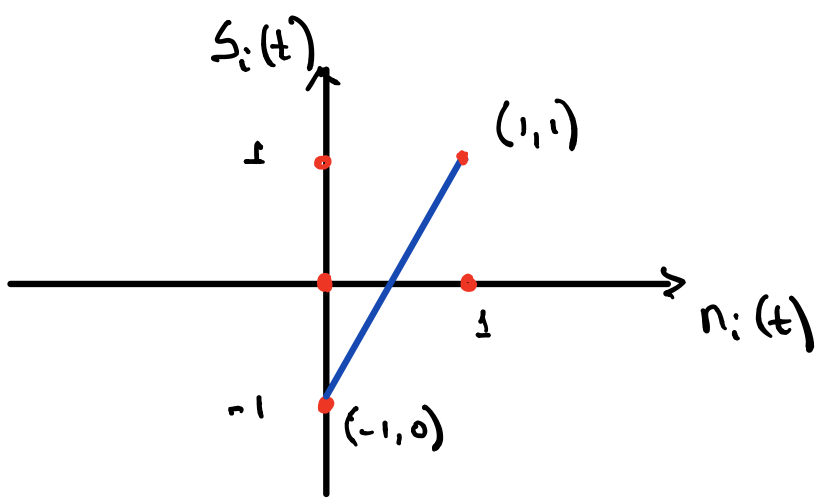
Lo primero que haremos en pasar de la representación $(0,1)$ a la representación $(-1,1)$ y llamaremos $S_i(t)$ al estado de la neurona i en el tiempo t.

$$S_i(t) : \begin{cases} +1 & \text{si la neurona } i \text{ dispara en } t \\ -1 & \text{si la neurona } i \text{ no dispara en } t \end{cases}$$

De $(0,1) \longrightarrow (-1,1)$ pasamos con una simple transformación lineal

$$n_i(t) \longrightarrow S_i(t)$$

$$S_i(t) = 2n_i(t) - 1$$

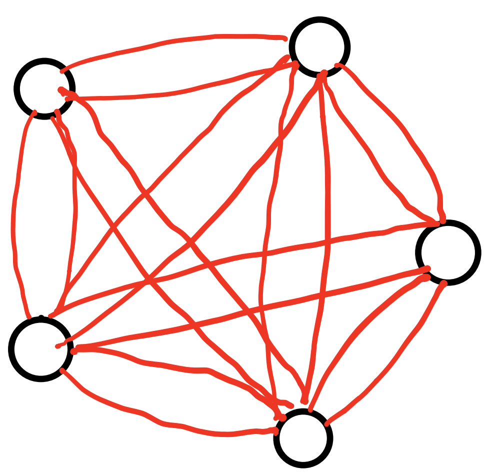


- La dinámica ahora toma la forma

$$S_i(t+\Delta t) = \text{signo}\left(h_i(t) - \mu_i \right)$$

$$= \text{signo}\left(\sum_{j=1}^{N} w_{ij} S_j(t) - \mu_i \right)$$

- En el modelo de Hopfield cada neurona se conecta con todas las otras, tanto para recibir estímulos como para enviar estímulos. Se denomina RED NEURONAL TOTALMENTE CONECTADA (FULLY CONNECTED).



- El modelo de Hopfield no contempla auto interacciones, o sea:

$$W_{ii} = 0 \qquad \forall\ i \ (i=1,\dots,N)$$

En otras palabras, la vecindad de la neurona i es el conjunto de las $(N-1)$ otras neuronas

$$V_i = \{ j \text{ tal que } j \neq i \}.$$

Así

$$h_i(t) = \sum_{\substack{j \neq i \\ j=1}}^{N} w_{ij}\, S_j(t)$$

al cual denotaremos como

$$h_i(t) = \sum_{j \neq i} w_{ij}\, S_j(t)$$

- En el modelo de Hopfield el tiempo es discreto y las neuronas se actualizan en forma secuencial, o sea, una por vez.

Esta actualización puede ser ORDENADA (vamos de la 1 a la N) o aleatoria.

Diremos que una unidad de tiempo corresponde a N actualizaciones sucesivas. De esta forma,

$$\Delta t = \frac{1}{N}$$

Luego veremos la versión sincrónica del modelo de Hopfield, que se conoce como Modelo de Little.

Planteamos ahora el problema básico de memoria asociativa en términos de la notación del modelo de Hopfield. Recordemos que con N neuronas binarias tenemos 2^N posibles configuraciones.

De entre las 2^N posibles configuraciones deseamos elegir p y buscamos la matriz $N \times N$ w_{ij} y el conjunto de N umbrales μ_j que hagan de estos p patrones elegidos atractores de la dinámica.

Los p patrones son un conjunto de $p \times N$ variables $\xi_i^\mu = +1$ o -1.

$$(\xi_1^1, \xi_2^1, \ldots\ldots, \xi_N^1) \quad \text{configuración } 1$$

$$(\xi_1^2, \xi_2^2, \ldots\ldots, \xi_N^2) \quad \text{configuración } 2$$

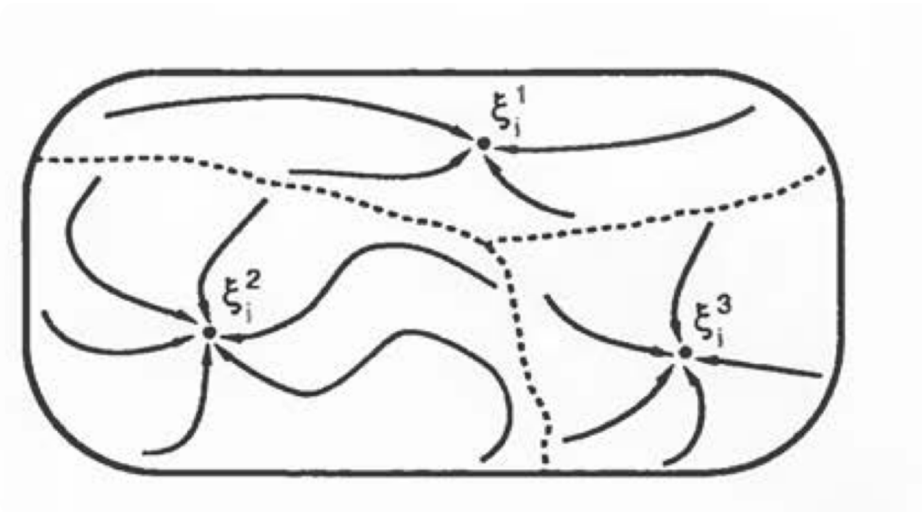
$$\vdots$$

$$(\xi_1^p, \xi_2^p, \ldots\ldots, \xi_N^p) \quad \text{configuración } p$$

$$\xi_i^\mu = \pm 1$$

$\mu = 1, 2, \ldots p$ denota configuraciones

$i = 1, 2, \ldots, N$ denota neuronas



En este esquema del libro de Hertz y colaboradores vemos el caso en que eligen 3 configuraciones $\{\xi_i^1\}$, $\{\xi_j^2\}$ y $\{\xi_j^3\}$

La idea es que si preparamos al sistema en una configuración inicial arbitraria $\{S_i(0)\}$, la dinámica simple que definimos

$$S_i(t+\Delta t) = \text{signo}\left(h_i(t) - \mu_i\right)$$

lleva el sistema AUTONOMAMENTE al atractor más parecido a $\{S_i(0)\}$. El criterio de similitud lo definirá la red.

$$\{S_i(0)\} \longrightarrow \{\xi_i^\nu\}$$

asocia el INPUT $\{S_i(0)\}$ con el OUTPUT $\{\xi_i^\nu\}$

Esto se podría resolver por fuerza de cálculo. Calcularíamos la distancia de Hamming entre la configuración inicial $\{S_i(0)\}$ y cada configuración $\mu = 1, 2, \ldots p$ almacenada, y nos quedamos con la "más próxima", o sea, con la que tiene menor distancia de Hamming con $\{S_i(0)\}$

$$d^{\mu} = \frac{1}{4N} \sum_{i=1}^{N} \left(\xi_i^{\mu} - S_i(0) \right)$$

$$\{ S_i(0) \} \longrightarrow \{ \xi_i^{\nu} \}$$

donde ν es tal que d^{ν} es el mínimo entre todos los $\mu = 1, 2, \ldots, P$.

NOTACIÓN: $\{ S_i \}$ denota el estado de la red, o sea

$$\{ S_i \}_{i=1,2,\ldots N}$$

pero omitimos $i = 1, 2, \ldots, N$

UN PATRÓN

Solo elegimos una configuración para almacenar, o sea, queremos tener un único atractor $\{\xi_i^1\}$.

Llamamos **patrones** o **memorias** a las configuraciones que queremos almacenar.

Hopfield eligió

$$W_{ij} = \frac{1}{N} \xi_i^1 \xi_j^1$$

$$W_{ii} = 0 \quad \forall i$$ no hay auto interacción

$$\mu_i = 0 \quad \forall i$$ no hay umbrales

Noten que es una regla LOCAL: el acoplamiento entre j e i depende de la configuración almacenada en i y en j. Podría depender de otras neuronas, pero es una regla local

Ponemos el sistema en $t=0$ en una configuración $\{S_i(0)\}$ cualquiera.

Si comparamos esta configuración con la memoria $\mu=1$ habrá n neuronas en las cuales

$$S_i(0) = \xi_i^1$$

y habrá $(N-n)$ neuronas en las cuales

$$S_i(0) = -\xi_i^1$$

No hay otra posibilidad. Sin perder generalidad, llamamos $i=1,\dots n$ a las n que coinciden, e $i=n+1,\dots,N$ a las $(N-n)$ que no coinciden

Veamos a dónde lleva la dinámica a la configuración inicial, que es el INPUT

$$S_i(0+\Delta t) = \text{signo}\left(h_i(0)\right)$$

$$= \text{signo}\left(\sum_{j\neq i} w_{ij}\, S_j(0)\right)$$

$$= \text{signo}\left(\frac{1}{N}\sum_{j\neq i} \xi_i^1 \xi_j^1\, S_j(0)\right)$$

$$= \text{signo}\left(\frac{1}{N}\right)\text{signo}\left(\sum_{j\neq i} \xi_i^1 \xi_j^1\, S_j(0)\right)$$

$$= \text{signo}\left(\sum_{j=1}^{n} \xi_i^1 \xi_j^1 \xi_j^1 + \sum_{j=n+1}^{N} \xi_i^1 \xi_j^1 (-\xi_j^1)\right)$$

$$= \text{signo}\left(\sum_{j=1}^{n} \xi_i^1 - \sum_{j=n+1}^{N} \xi_i^1\right)$$

$$= \text{signo}\left(\left(n - N + n\right)\xi_i^1\right)$$

$$= \text{signo}\left(2n - N\right)\,\text{signo}\left(\xi_i^1\right)$$

$$= \xi_i^1\ \text{signo}\left(n - \frac{N}{2}\right)$$

Aquí usamos que

$$\text{signo}\left(\frac{1}{N}\right) = 1 \quad \text{pues } N > 0$$

$$\xi_j^1 \xi_j^1 = 1 \text{ siempre, pues } \xi_i^1 = \pm 1$$

$$\text{signo}(a \cdot b) = \text{signo}(a) \cdot \text{signo}(b)$$

$$\text{signo}(\xi_i^1) = \xi_i^1$$

Entonces:

$$S_i(\Delta t) = \xi_i^1 \ \text{signo}\left(n - \frac{N}{2}\right)$$

Supongamos por un momento N impar.

- Si $n > \frac{N}{2}$, $\text{signo}\left(n - \frac{N}{2}\right) = +1$

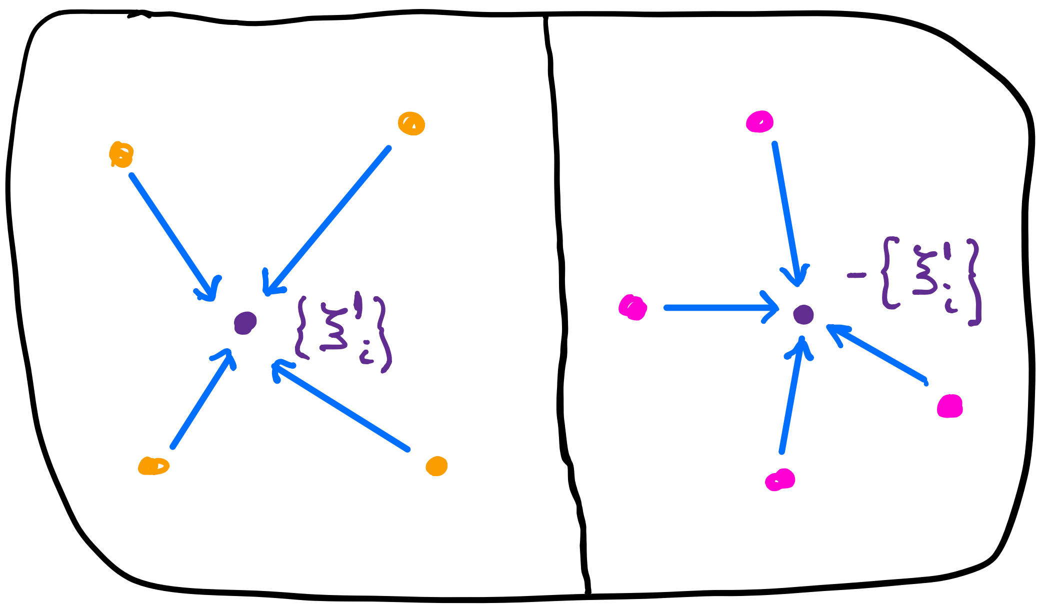
$$S_i(\Delta t) = \xi_i^1 \qquad \forall i$$

O sea, en una actualización llegamos a la memoria almacenada

- Si $n < \frac{N}{2}$, signo $\left(n - \frac{N}{2}\right) = -1$ y

$$S_i(\Delta t) = -\xi_i^1 \quad \forall i$$

O sea, en una actualización llegamos al negativo de la memoria almacenada



Resumiendo, la regla anda bien para un patrón o memoria, con la salvedad de que almacena la memoria y su negativo. Esto no es grave, y responde a la simetría $S_i \to -S_i \;\; \forall i$ de la dinámica $S_i = \text{signo}(h_i)$