



## Modelación Numérica de Sistemas Estocásticos

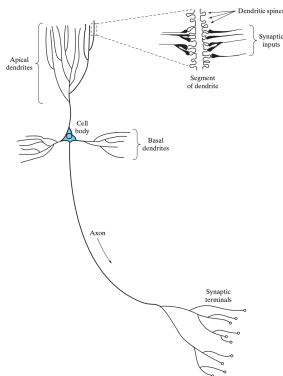
Daniel Otero Fadul

*Departamento de Matemáticas y Ciencia de Datos  
Escuela de Ingeniería y Ciencias*

# Redes Neuronales

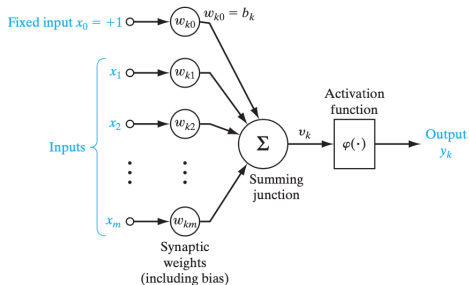
Las **redes neuronales**, también conocidas como “Artificial Neural Networks” (ANN) en inglés, son sistemas computacionales que modelan una relación entre un conjunto de señales de entrada y un conjunto de señales de salida. Estas redes están conformadas por *neuronas* que se conectan a otras por medio de *sinapsis*.

# Redes Neuronales



**Figura:** Como su nombre lo indica, las redes neuronales son sistemas que fueron desarrollados tomando como inspiración el cerebro biológico. La neurona biológica recibe señales eléctricas por medio de sus dendritas, las cuales reciben un distinto grado de importancia gracias a un proceso bioquímico. La acumulación de estas señales hace que se active una respuesta una vez se alcance un umbral, la cual se transmite a través del axón. Esta señal de salida se transmite a otras neuronas pues el axón de una neurona se conecta a las dendritas de neuronas vecinas. Imagen tomada de [2].

# Redes Neuronales



**Figura:** La neurona biológica se modela como una neurona artificial como la que se ve en la figura. Imagen tomada de [2].

Dado lo anterior, la expresión que relaciona las entradas y la salida de una neurona está dada por

$$\begin{aligned} y &= \phi \left( \sum_{i=0}^n w_i x_i \right) \\ &= \phi(w^T x), \end{aligned}$$

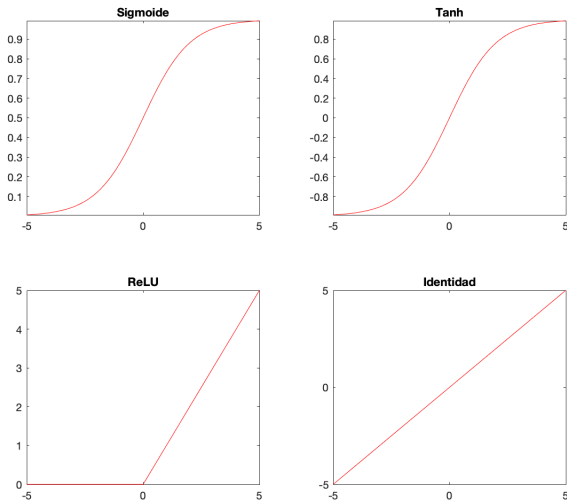
donde  $w \in \mathbb{R}^{n+1}$  es un vector de pesos,  $x_0 = 1$  y  $\phi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$  es la **función de activación**.

## Función de Activación

La función de activación es el mecanismo por medio del cual la neurona procesa la información de entrada y genera una señal de salida. Las funciones de activación más comunes son las siguientes:

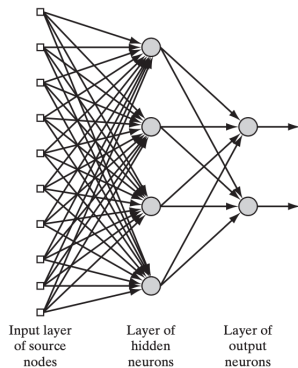
- **Sigmoide:**  $\sigma(z) = \frac{1}{1+e^{-z}}$ .
- **Tangente hiperbólica:**  $\phi(z) = 2\sigma(z) - 1$ .
- **Unidad Lineal Rectificada (ReLU):**  $\phi(z) = \max(0, z)$ .
- **Identidad:**  $\phi(z) = z$ .

# Redes Neuronales



**Figura:** Algunas de las funciones de activación más usadas.

# Redes Neuronales



**Figura:** En la figura se puede ver un ejemplo de una red totalmente conectada que tiene una capa de entrada, una capa oculta y una capa de salida. Imagen tomada de [2].



## Regla Delta

Supongamos que tenemos solo una neurona y un conjunto de entrenamiento con  $m$  ejemplos. Además, tenemos la siguiente función de costo que nos cuantifica el error entre unas observaciones objetivo  $t_i$  y la salida de nuestra neurona:

$$C(w) = \frac{1}{2} \sum_{j=1}^m (\phi(w^T x_j) - t_j)^2,$$

Nótese que  $t_i$  es la  $i$ -ésima observación objetivo asociada al ejemplo  $x_j$ . Para encontrar los pesos óptimos que nos minimizan la función de costo se puede utilizar la *regla delta*.

Esta regla está basada en el método de gradiente descendente, por lo cual la iteración que actualiza nuestros pesos está dada por

$$w_{n+1} = w_n - \alpha \nabla C(w_n),$$

donde  $\alpha$  se conoce como la **taza de aprendizaje**.

Consideremos la derivada parcial de la función de costo respecto a el  $j$ -ésimo peso  $w_j$ :

$$\begin{aligned}\frac{\partial C(w)}{\partial w_i} &= \frac{1}{2} \sum_{j=1}^m 2(\phi(w^T x_j) - t_j) \frac{\partial}{\partial w_i} (\phi(w^T x_j) - t_j) \\ &= \sum_{j=1}^m (\phi(w^T x_j) - t_j) \phi'(w^T x_j) x_{ij}.\end{aligned}$$

Esta última expresión es la que determina la cantidad con la que debemos actualizar los pesos de la neurona. Nótese que depende de la diferencia entre la salida de nuestra neurona y la salida objetivo, por esta razón este método recibe el nombre de regla delta.

Lo anterior nos permite definir el siguiente algoritmo:

**inicializar**  $w = w_0$ ,  $\alpha \in (0, 1)$ ;

**repetir**  $n$  veces

**calcular**  $\sum_{j=1}^m (\phi(w^T x_j) - t_j) \phi'(w^T x_j) x_{ij}$  para cada peso  $w_i$ ;

$w = w - \alpha \nabla C(w)$ ;

**retornar**  $w$ .

La regla delta se puede emplear cuando tenemos una red neuronal que tiene solo una capa de entrada y una capa de salida de dos neuronas o más. Si tenemos  $m$  ejemplos en nuestro conjunto de entrenamiento y  $p$  neuronas en la capa de salida, la nueva función de costo es igual a

$$C(w) = \frac{1}{2} \sum_{k=1}^m \sum_{j=1}^p (\phi(w_j^T x_k) - t_{jk})^2,$$

donde  $w_j \in \mathbb{R}^{n+1}$  es el vector de pesos de la  $j$ -ésima neurona, y  $t_{jk}$  es la  $k$ -ésima observación objetivo de la  $j$ -ésima neurona asociada al ejemplo  $x_k \in \mathbb{R}^{n+1}$ .

Consideremos la derivada parcial de la función de costo respecto al peso  $w_{ij}$ :

$$\begin{aligned}\frac{\partial C(w)}{\partial w_{ij}} &= \frac{1}{2} \sum_{k=1}^m \sum_{j=1}^p 2(\phi(w_j^T x_k) - t_{jk}) \frac{\partial}{\partial w_{ij}} (\phi(w_j^T x_k) - t_{jk}) \\ &= \sum_{k=1}^m (\phi(w_j^T x_k) - t_{jk}) \phi'(w_j^T x_k) x_{ik}.\end{aligned}$$

Nótese que, nuevamente, el término con que actualizamos los pesos de la red neuronal depende de la diferencia entre la salida de nuestra neurona y la salida objetivo.

Así como en el caso de una red de una sola neurona, podemos utilizar un algoritmo muy similar al anterior para obtener los pesos óptimos cuando la capa de salida tiene más de una neurona:

**inicializar**  $w = w_0$ ,  $\alpha \in (0, 1)$ ;

**repetir**  $n$  veces

**calcular**  $\sum_{k=1}^m (\phi(w_j^T x_k) - t_{jk}) \phi'(w_j^T x_k) x_{ik}$  para cada peso  $w_{ij}$ ;

$w = w - \alpha \nabla C(w)$ ;

**retornar**  $w$ .

## Neurodinámica

Las redes neuronales se pueden ver como sistemas dinámicos no lineales. Esto implica que la teoría de estabilidad de los sistemas dinámicos también se puede aplicar en el caso de redes neuronales. El área de estudio en la cual las redes neuronales se modelan como un sistema dinámico, prestándole especial atención a su estabilidad, se le conoce como **neurodinámica**.



## Sistemas Dinámicos

Un sistema dinámico se puede modelar como un conjunto de ecuaciones diferenciales no lineales:

$$\begin{aligned}\dot{x}_i &= f_i(t; x_1, x_2, \dots, x_n), \quad i = 1, 2, \dots, n \\ x_i(t_0) &= x_{i0}.\end{aligned}$$

Utilizando notación vectorial tenemos que:

$$\begin{aligned}\dot{x} &= f(t; x) \\ x(t_0) &= x_0,\end{aligned}$$

donde el estado del sistema es  $x \in \mathbb{R}^n$ .

Este sistema de ecuaciones diferenciales no lineales tiene solución única si la función  $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  satisface la *condición de Lipschitz*: existe una constante  $K > 0$  tal que

$$\|f(x) - f(y)\| \leq K\|x - y\|,$$

$$\forall x, y \in \mathbb{R}^n.$$

## Linealización

Recordemos que, según Taylor, una función  $f : \mathbb{R} \rightarrow \mathbb{R}$  puede, en algunos casos, tener la siguiente serie:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2!}f''(x_0)(x - x_0)^2 + \dots$$

Dado esto, una aproximación de primer orden de  $f$  estaría dada por

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0),$$

la cual corresponde a la ecuación de una recta.

Para una función  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  se puede tener la siguiente expansión:

$$f(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2!} (x - x_0) H(x_0) (x - x_0)^T + \cdots,$$

donde  $H(x_0)$  es el *Hessiano* de  $f$  evaluado en  $x_0$ . En este caso, la aproximación de primer orden corresponde a la ecuación de un plano:

$$f(x) \approx f(x_0) + \nabla f(x_0)^T (x - x_0).$$

Si  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , entonces

$$f(x) = f(x_0) + J(x_0)(x - x_0) + \cdots,$$

donde  $J(x_0)$  es el *Jacobiano* de  $f$  evaluado en  $x_0$ :

$$J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

Nótese que si  $f(x_0) = 0$  y  $x_0 = 0$  la aproximación de primer orden es igual a

$$f(x) \approx J(x_0)x.$$

Sin embargo, esto es muy restrictivo. Una alternativa es obtener la aproximación de primer orden alrededor de puntos en el espacio de estados tales que  $f(\bar{x}) = 0$ .

Sea  $\hat{x} = x - \bar{x}$ . Entonces, la aproximación de primer orden de  $f$  alrededor del punto  $\bar{x}$  está dada por

$$f(\hat{x} + \bar{x}) \approx J(\bar{x})\hat{x}.$$

Dado el siguiente sistema no lineal

$$\dot{x} = f(t; x),$$

un *punto de equilibrio* se define como la configuración de estados y entradas tales que

$$\dot{\bar{x}} = f(t; \bar{x}) = 0 \quad \forall t \in T.$$

En general, un sistema se modela con un conjunto de ecuaciones diferenciales de algún orden. Considérese la siguiente ODE lineal de orden  $n$ :

$$\frac{d^n}{dt^n}x(t) + a_{n-1}\frac{d^{n-1}}{dt^{n-1}}x(t) + \cdots + a_1\frac{d}{dt}x(t) + a_0x(t) = 0.$$

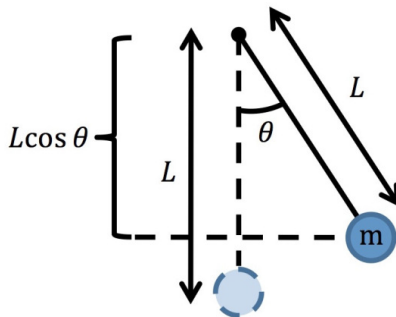


Las *variables de estado* se definen como

$$\begin{aligned}x_1(t) &= x(t) \\x_2(t) &= \frac{d}{dt}x(t) \\&\vdots \\x_n(t) &= \frac{d^{n-1}}{dt^{n-1}}x(t).\end{aligned}$$

Entonces, la ODE lineal de orden  $n$  se puede reescribir en términos de las variables de estado como sigue:

$$\begin{aligned}\dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= x_3(t) \\ &\vdots \\ \dot{x}_n(t) &= -\sum_{i=0}^{n-1} a_i x_{i+1}(t).\end{aligned}$$



**Figura:** Péndulo simple. Imagen tomada de <https://jeremykao.net/2017/07/03/simulating-a-simple-pendulum-in-python-part-1/>

Se puede mostrar que la ecuación diferencial que modela este sistema está dada por

$$\ddot{\theta} + \frac{g}{L} \sin \theta = 0.$$

Sean  $\theta_1 = \theta$  y  $\theta_2 = \dot{\theta}$ . Entonces,

$$\begin{aligned}\dot{\theta}_1 &= \theta_2 \\ \dot{\theta}_2 &= -\frac{g}{L} \sin \theta_1.\end{aligned}$$

Expresando este sistema de ecuaciones en forma vectorial se tiene que

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \theta_2 \\ -\frac{g}{L} \sin \theta_1 \end{bmatrix}.$$

Ahora hallamos el punto de equilibrio:

$$\begin{bmatrix} \bar{\theta}_2 \\ -\frac{g}{L} \sin \bar{\theta}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

En este caso, se puede ver que  $\bar{\theta} = [0, 0]^T$ .

El Jacobiano está dado por

$$J(\theta_1, \theta_2) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} \cos \theta_1 & 0 \end{bmatrix}.$$

Así que

$$J(\bar{\theta}_1, \bar{\theta}_2) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{bmatrix}.$$

Por lo tanto, las ecuaciones de estado del péndulo simple se pueden definir como

$$\dot{\theta} = A\theta,$$

donde  $A$  es igual a

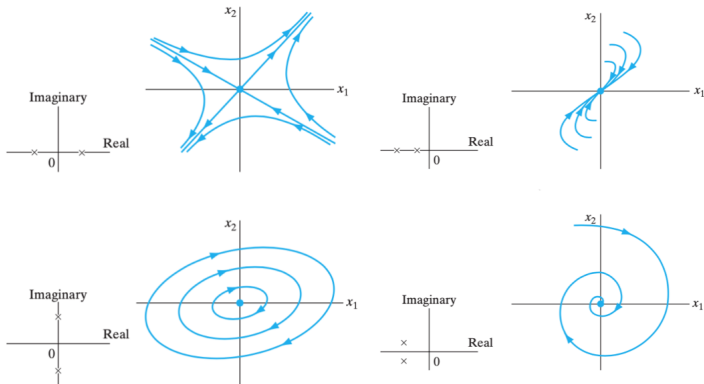
$$A = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{bmatrix}.$$

Esta matriz se le conoce como la *matriz de estados*.

## Estabilidad

Si la matriz de estados  $A$  de un sistema dinámico es invertible, su estabilidad alrededor de un punto de equilibrio se puede investigar por medio de los *valores propios* de la matriz  $A$ . En el caso de un sistema de segundo orden tenemos la siguiente clasificación:

Tipo de Equilibrio	Valores propios de $A$
Nodo estable	Reales y negativos
Foco estable	Complejos conjugados con parte real negativa
Nodo inestable	Reales y positivos
Foco Inestable	Complejos conjugados con parte real positiva
Punto de silla	Reales con signos opuestos
Centro	Conjugados puramente imaginarios



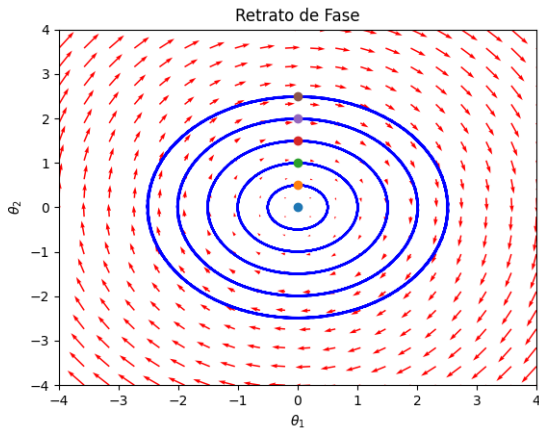
**Figura:** Las gráficas del conjunto de trayectorias que un sistema recorre en el espacio de estados se les llama **retratos de fase**. En esta figura se pueden ver las formas típicas de los retratos de fase para un sistema de segundo orden; estas cambiarán dependiendo de la ubicación de los valores propios de la matriz de estados  $A$ .



En el caso del péndulo simple, tenemos que los valores propios del sistema están dados por

$$\begin{aligned}\det(\lambda I - A) &= \det\left(\lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{bmatrix}\right) \\ &= \det\left(\begin{bmatrix} \lambda & -1 \\ \frac{g}{L} & \lambda \end{bmatrix}\right) \\ &= \lambda^2 + \frac{g}{L}.\end{aligned}$$

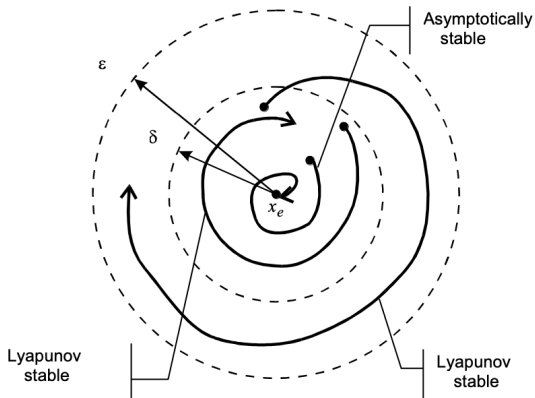
Así que los valores propios son iguales a  $\lambda = \pm i\sqrt{\frac{g}{L}}$ .



**Figura:** Retrato de fase del péndulo simple. Cada elipse corresponde a una condición inicial diferente.

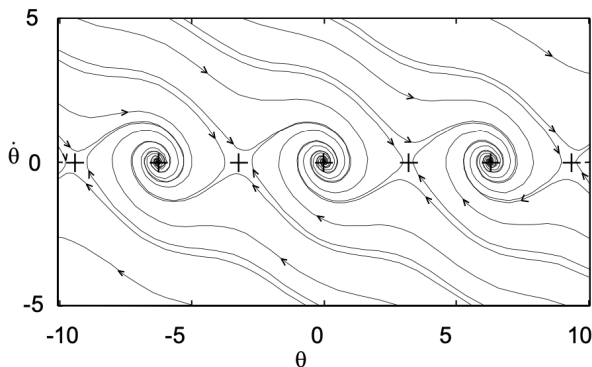
## Tipos de Estabilidad

- El punto de equilibrio  $\bar{x}$  es **estable** si, para cualquier constante  $\epsilon > 0$ , existe  $\delta(\epsilon)$  tal que  $\|x(0) - \bar{x}\| < \delta(\epsilon)$  implica  $\|x(t) - \bar{x}\| < \epsilon$  para todo  $t \in T$ . Si  $\delta = \delta(\epsilon)$ , el punto de equilibrio  $\bar{x}$  es **uniformemente estable**.
- El punto de equilibrio  $\bar{x}$  es **convergente** si existe  $\delta > 0$  tal que  $\|x(0) - \bar{x}\| < \delta$  implica  $x(t) \rightarrow \bar{x}$  a medida que  $t \rightarrow \infty$ .
- El punto de equilibrio  $\bar{x}$  es **asintóticamente estable** si es estable y convergente.
- El punto de equilibrio  $\bar{x}$  es **globalmente estable** si es estable y todas las trayectorias del sistema convergen a  $\bar{x}$  cuando  $t \rightarrow \infty$ .



Por otro lado, para sistemas lineales, los puntos de equilibrio son, ya sea, puntos aislados o subespacios del espacio de estados. En cualquier caso, estos sistemas son siempre globalmente estables.

En el caso de sistemas no lineales, en los que múltiples puntos de equilibrio pueden existir, puede haber regiones alrededor de estos puntos en las que el sistema se comporte de forma estable. Sin embargo, si el sistema se aleja de estas *regiones de atracción*, la trayectoria del sistema podría ir en cualquier otra dirección.



**Figura:** Retrato de fase de un péndulo con amortiguamiento. La ecuación diferencial que describe el comportamiento del sistema es  $\ddot{\theta} + 0.4\dot{\theta} + \sin(\theta) = 0$ . Las cruces muestra la ubicación de los puntos de equilibrio:  $\theta_e = n\pi$ , donde  $n \in \mathbb{Z}$ . Imagen tomada de [4].

Las anteriores definiciones de estabilidad requerirían resolver la ecuación de estados para determinar qué tipo de estabilidad tiene el sistema correspondiente. Un enfoque más práctico para ver qué tipo de estabilidad tiene un sistema es utilizar el **método directo de Lyapunov**:

**Teorema I:** El punto de equilibrio  $\bar{x}$  es **estable** si, en una vecindad de  $\bar{x}$ , existe una función definida positiva  $V(x)$  tal que su derivada respecto al tiempo es semidefinida negativa en esta misma vecindad.

**Teorema II:** El punto de equilibrio  $\bar{x}$  es **asintóticamente estable** si, en una vecindad de  $\bar{x}$ , existe una función definida positiva  $V(x)$  tal que su derivada respecto al tiempo es definida negativa en esta misma región.

La función  $V(x)$  se le conoce como **función de Lyapunov**. Dados los teoremas anteriores, esta función debe ser definida positiva en una vecindad  $U$  de  $\bar{x}$ :

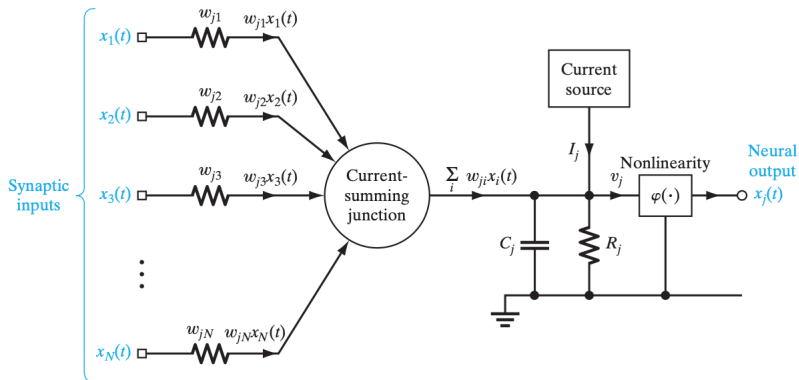
- $V(x)$  tiene derivadas parciales continuas respecto a las componentes del vector de estados  $x$ .
- $V(\bar{x}) = 0$ .
- $V(x) > 0$  si  $x \in U - \bar{x}$ .

En cuanto a las derivadas, tenemos dos opciones:

- $\frac{d}{dt} V(x) \leq 0$  si  $x \in U - \bar{x}$ .
- $\frac{d}{dt} V(x) < 0$  si  $x \in U - \bar{x}$ .



# Redes Neuronales



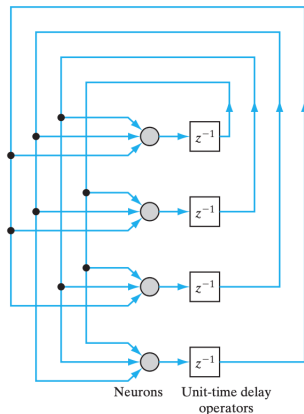
**Figura:** En la figura se puede ver el **modelo aditivo** de la  $j$ -ésima neurona de una red neuronal. Como se puede apreciar, esta es una implementación analógica de una neurona. Imagen tomada de [2].

Recurriendo a las *leyes de Kirchoff*, tenemos que

$$C_j \frac{d}{dt} v_j(t) + \frac{v_j(t)}{R_j} = \sum_{i=1}^N w_{ji} x_i(t) + I_j$$
$$C_j \frac{d}{dt} v_j(t) = -\frac{v_j(t)}{R_j} + \sum_{i=1}^N w_{ji} x_i(t) + I_j.$$

Nótese que  $x_j(t) = \phi(v_j(t))$ .

# Redes Neuronales



**Figura:** Arquitectura de una red neuronal de Hopfield de cuatro neuronas. Nótese que la salida de una neurona no se realimenta a si misma. Imagen tomada de [2].

La red de Hopfield es un tipo de red neuronal recurrente que se comporta como una *memoria asociativa*. Esta red funciona bajo tres supuestos:

- La matriz de pesos sinápticos es simétrica:  $w_{ji} = w_{ij}$ .
- Cada neurona tiene una función de activación propia  $\phi_j$ .
- La inversa de la función de activación existe:  $v = \phi^{-1}(x)$ .

Por otro lado, utilizando el modelo aditivo, la función de energía de la red de Hopfield se define como

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ji} x_i x_j + \sum_{j=1}^N \frac{1}{R_j} \int_0^{x_j} \phi^{-1}(x) dx - \sum_{j=1}^N I_j x_j.$$

Se puede demostrar que la derivada de la energía respecto al tiempo es igual a

$$\frac{dE}{dt} = - \sum_{j=1}^N C_j \left( \frac{dx_j}{dt} \right)^2 \left( \frac{d}{dx_j} \phi^{-1}(x_j) \right).$$

Si  $\phi^{-1}$  es una función monótona y creciente,  $\frac{d}{dx_j} \phi^{-1}(x_j) \geq 0$  para todo  $j$ . Por lo tanto,

$$\frac{dE}{dt} \leq 0 \text{ para todo } t.$$

Lo anterior implica que la energía se puede considerar como una función de Lyapunov. Dado esto, para un punto de equilibrio  $\bar{x}$  de la red de Hopfield, según el teorema I, este es un punto estable.

Es más, si  $\frac{d}{dx_j} \phi^{-1}(x_j) > 0$ , y notando que la derivada de la energía respecto al tiempo es igual a cero si

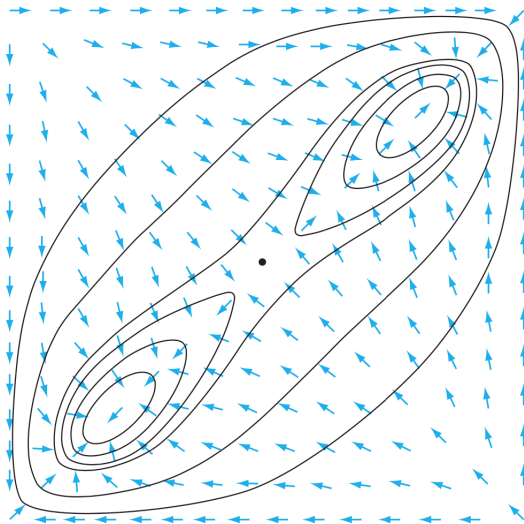
$$\frac{d}{dt} x_j(t) = 0 \text{ para todo } j,$$

entonces,

$$\frac{dE}{dt} < 0 \text{ para todo } t.$$

Por lo tanto, según el teorema II, la red de Hopfield es asintóticamente estable en el sentido de Lyapunov: los puntos de equilibrio son los mínimos de la función de energía.

## Redes Neuronales



**Figura:** Retrato de fase de una red de Hopfield de dos neuronas. Cada eje corresponde a la salida de cada neurona. Los estados estables están localizados en la esquina inferior izquierda y la esquina superior derecha. Imagen tomada de [2].

Lo que acabamos de ver es la red de Hopfield para el caso continuo, sin embargo, existe la contraparte discreta de esta red. En este caso, la función de energía se define como

$$E = -\frac{1}{2} \sum_{i,j=1, i \neq j}^N w_{ji} x_i x_j,$$

donde  $x_k = \pm 1$ .

El argumento de la función de activación de la  $j$ -ésima neurona se define como

$$v_j = \sum_{i=1}^N w_{ji} x_i + b_j,$$

donde  $b_j$  es un “bias” externo que se aplica a la  $j$ -ésima neurona.

La función de activación se define, normalmente, como

$$x_j = \text{sign}(v_j) = \begin{cases} 1, & v_j > 0, \\ -1, & v_j < 0. \end{cases}$$

Si  $v_j = 0$ , una posible convención es permitir que la neurona siga en su estado anterior, ya sea este 1 o -1.



- **Aprendizaje:** Supongamos que queremos aprender  $m$  vectores  $\{x_i\}_{i=1}^m$ , donde cada componente de estos vectores es igual a 1 o -1. Entonces, la matriz de pesos sinápticos se define como

$$W = \frac{1}{m} \sum_{i=1}^m x_i x_i^T - I,$$

donde  $I$  es la matriz identidad de tamaño  $N \times N$ .

- **Inicialización:** Sea  $\tilde{x}$  un vector que se le presenta a la red. Entonces  $x[0] = \tilde{x}$ : el estado en el tiempo cero es  $\tilde{x}$ .
- **Iterar hasta convergencia:** El estado de la red en el tiempo  $k$  se define como

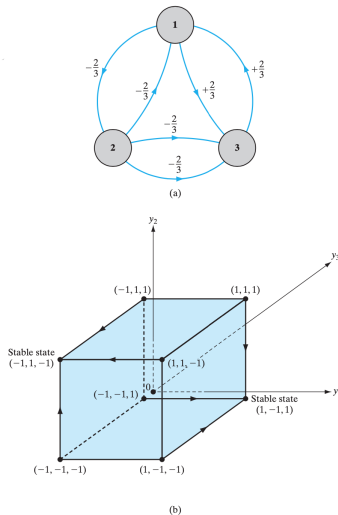
$$x[k+1] = \text{sign}(Wx[k]).$$

Repetir hasta que  $\|x[k+1] - x[k]\|_1 \simeq 0$ .

- **Retornar:** La salida de la red es la aproximación del *punto fijo*  $\bar{x}$  obtenido en el paso anterior:

$$y = \bar{x}.$$

# Redes Neuronales



**Figura:** (a): Una red de Hopfield de tres neuronas. (b): Los estados estables de la red son justamente las “memorias” aprendidas:  $x_1 = [1, -1, 1]$  y  $x_2 = [-1, 1, -1]$ . Imagen tomada de [2].

- 1 Dangeti, P. *"Statistics for machine learning,"* Packt Publishing Ltd., 2017.
- 2 Haykin, S., *"Neural networks and learning machines"*, Pearson Education India, 2010.
- 3 Stuart R., Norvig P, *"Artificial intelligence: a modern approach,"* 2002.
- 4 J.S. Bay, *Fundamentals of Linear State Space Systems*, New York: McGraw Hill International Edition, 1999.