

Universidad Veracruzana



Ingeniería de Software

Reporte Técnico

Paradigmas de Programación

Axel Gustavo Peña Sánchez

17 de marzo de 2022

Prof. Adolfo Centeno Tellez

# Introducción

Este reporte técnico nos habla sobre la red de Hopfield, utilizada para reconocer distintos patrones, nos muestra que es y cómo funciona. En principio comenzaremos por entender realmente:

## ¿Que es una Red Neuronal?

Una red neuronal es un modelo de computación cuya estructura de capas se asemeja a la estructura interconectada de las neuronas en el cerebro, con capas de nodos conectados. Una red neuronal puede aprender de los datos, de manera que se puede entrenar para que reconozca patrones, clasifique datos y pronostique eventos futuros.

Las redes neuronales descomponen las entradas en capas de abstracción. Se pueden entrenar con muchos ejemplos para que reconozcan patrones de voz o en imágenes, por ejemplo, igual que el cerebro humano. Su comportamiento está definido por la forma en que se conectan sus elementos individuales, así como por la importancia (o ponderación) de dichas conexiones. Estas ponderaciones se ajustan automáticamente durante el entrenamiento de acuerdo con una regla de aprendizaje especificada hasta que la red neuronal lleva a cabo la tarea deseada correctamente.

Las redes neuronales resultan especialmente adecuadas para llevar a cabo el reconocimiento de patrones a fin de identificar y clasificar objetos o señales en sistemas de voz, visión y control. También se pueden emplear para el modelado y la predicción de series temporales.

Algunos ejemplos de la aplicación de una red neuronal son:

- Las compañías eléctricas pronostican la carga de sus redes con precisión para garantizar la fiabilidad y optimizar la eficiencia de los generadores eléctricos que utilizan.
- Los cajeros automáticos pueden aceptar depósitos bancarios de forma fiable mediante la lectura del número de cuenta y del importe del depósito en un cheque.
- Los patólogos confían en aplicaciones de detección de cáncer como guía a la hora de clasificar los tumores como benignos o malignos en función de la uniformidad del tamaño de las células, el grosor de la masa, la mitosis y otros factores.

# Desarrollo

## Red neuronal de Hopfield

Es Modelo construido en 1982 por el físico norteamericano J. Hopfield, descrito en el artículo titulado “Redes neuronales y sistemas físicos con habilidades computacionales colectivas emergentes”, éste modelo fue construido con el número suficiente de simplificaciones como para extraer analíticamente información sobre las características relevantes del sistema, conservando las ideas fundamentales de las redes construidas en el pasado y presentando una serie de funciones básicas de los sistemas neuronales reales. Además, Hopfield supo establecer un paralelismo lo cual ha permitido aplicar todo un conjunto de técnicas bien conocidas en este campo y, con ello, producir un avance en la comprensión del funcionamiento de las redes neuronales.

Con su aporte, Hopfield redescubrió el mundo casi olvidado de las redes auto asociativas, caracterizadas por una nueva arquitectura y un nuevo funcionamiento, a las que se tuvo que añadir otro tipo de reglas de aprendizaje. Las consecuencias fueron redes con un comportamiento diferente a las diseñadas con estructura progresiva hacia adelante.

Debido a la arquitectura y al funcionamiento, la red de Hopfield se puede incluir dentro de las redes de neuronas recurrentes, pues cada neurona está conectada con todas las demás y además se produce un procesamiento temporal de los patrones. Lo que diferencia a esta red de las demás redes recurrentes es que actúa a la manera de una memoria asociativa. El concepto de memoria asociativa es bastante intuitivo: se trata simplemente de asociar dos patrones. Dentro de este concepto definiremos diferentes tipos de memorias asociativas:

**Memoria heteroasociativa:** Establece una correspondencia  $F$  entre dos vectores  $\vec{x}$  e  $\vec{y}$  de tal manera que  $F(\vec{x}_i) = \vec{y}_i$ , y si un  $\vec{x}$  arbitrario está más próximo a  $\vec{x}_i$  que a cualquier otro  $\vec{x}_j$ , entonces  $F(\vec{x}) = \vec{y}_i$ . En esta definición, el estar más próximo quiere decir con respecto a la *Distancia de Hamming*.

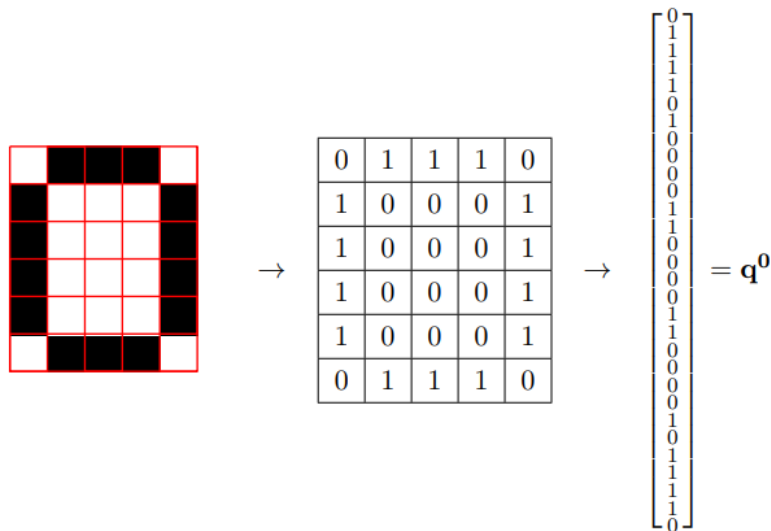
**Memoria asociativa interpoladora:** Establece una correspondencia  $F$  entre  $\vec{x}$  e  $\vec{y}$  de tal manera que  $F(\vec{x}_i) = \vec{y}_i$ , pero si el vector de entrada difiere de uno de los ejemplares en el vector  $\vec{d}$ , de tal modo que  $\vec{x} = \vec{x}_i + \vec{d}$ , entonces  $F(\vec{x}) = \vec{y}_i + \vec{d}$ .

**Memoria autoasociativa:** Presupone que  $\vec{y} = \vec{x}$  y establece una correspondencia  $F$  de  $\vec{x}$  en  $\vec{x}$  tal que  $F(\vec{x}) = \vec{x}$ , y si algún  $\vec{x}_i$  arbitrario está más próximo a  $\vec{x}$  que a cualquier otro vector  $\vec{x}_j$ , entonces  $F(\vec{x}_i) = \vec{x}$ .

## ¿Cómo funciona?

Se trata de una red auto asociativa. Por tanto, informaciones diferentes (patrones) pueden ser almacenadas en la red, como si de una memoria se tratase, durante la etapa de aprendizaje. Posteriormente, cuando se presenta una entrada a la red, esta evoluciona hasta generar una salida que coincidirá con la que corresponde a esa entrada, o bien la más parecida si la entrada está distorsionada o incompleta.

La información que recibe la red debe haber sido previamente codificada y representada en forma de vector (como una configuración binaria o como un conjunto de valores reales dependiendo de si la red es discreta o continua) con tantas componentes como neuronas (N) tenga la red. Cada neurona recibe un elemento del vector. Centrándonos en una sola neurona el funcionamiento sería el siguiente:



1. Recibe como entrada la salida de cada una de las otras neuronas (por las conexiones laterales). Estos valores de salida, inicialmente coinciden con las entradas del vector, multiplicadas por los pesos de las conexiones correspondientes. La suma de todos estos valores constituirá el valor de entrada neta de la neurona a la que hay que aplicarle la función de transferencia obteniéndose el valor de salida correspondiente. En el instante inicial ( $t=0$ ) la información de entrada es ( $e_1, e_2, \dots, e_N$ )

$$si(t=0) = ei \text{ 1 menor o igual a } i \text{ menor o igual a } N$$

$$si(t+1) = f(\text{Sum } wij \text{ } sj(t) - 0i) \text{ 1 menor o igual a } i \text{ menor o igual a } N$$

2. Este proceso continúa hasta que las salidas de las neuronas se estabilizan, alcanzan la convergencia, durante algunas iteraciones.

$$\text{si}(t+1) = \text{si}(t)$$

$$w_{ij} = \sum_{k=1}^p x_j(k)x_i(k) \quad \forall i \neq j$$

Esta ecuación cumple la regla de Hebb, pues si  $x_i(k)$  y  $x_j(k)$  son iguales el peso de la conexión de  $i$  a  $j$  se incrementa en una unidad, mientras que si son distintas se reducen en la misma cantidad.

## Fase de recuperación

Sea  $x = (x_1, x_2, \dots, x_n)$  un patrón de prueba diferente de los patrones almacenados en la fase anterior. Dicho patrón representa generalmente una versión de uno de los vectores  $x(k)$  almacenados pero con información incompleta o al que se le ha incluido ruido. En esta fase la red de Hopfield recuperará el patrón almacenado más próximo a  $x$ . El procedimiento seguido será el siguiente:

Paso 1. Se inicializan los estados de las  $n$  neuronas de la red utilizando el patrón

$$x: s_i(0) = x_i \text{ para } i = 1, 2, \dots, n$$

Paso 2. Se espera a que la red alcance un estado estable o punto fijo, entendiendo por tal aquel en el que el estado de los elementos de la red permanezca invariante en el tiempo:

$$s_i(t+1) = s_i(t) \quad \forall i = 1, 2, \dots, n$$

## Red Neuronal en Matlab

Esta red neuronal consiste en reconocer patrones de signos de puntuación, el tamaño de la matriz utilizada es de 9x12, los patrones a reconocer son los siguientes:

```
[0 0 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 0
0 1 1 0 0 0 0 1 1
0 1 1 0 0 0 0 1 1
0 1 1 0 0 0 0 1 1
0 0 0 0 0 0 1 1 0
0 0 0 0 0 1 1 0 0
0 0 0 0 1 1 0 0 0
0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0 0
0 0 0 1 1 0 0 0 0];
```

Signo de interrogación

```
[1 1 1 1 1 1 1 1 0
1 0 0 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 1 0
1 1 1 1 1 1 1 1 0];
```

Corchete

```
[0 0 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 0
0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 1 1 0
0 0 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 0];
```

Llave

```
[1 1 1 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0
0 1 1 1 0 0 0 0 0
0 1 1 1 0 0 0 0 0
0 0 1 1 1 0 0 0 0
0 0 1 1 1 0 0 0 0
0 0 0 1 1 1 0 0 0
0 0 0 1 1 1 0 0 0
0 0 0 0 1 1 1 0 0
0 0 0 0 1 1 1 0 0
0 0 0 0 1 1 1 0 0
0 0 0 0 0 1 1 1 0
0 0 0 0 0 1 1 1 0];
```

Diagonal

```
[0 1 1 1 0 0 1 1 1
1 1 1 0 0 1 1 1 0
1 1 1 0 0 1 1 1 0
1 1 1 0 0 1 1 1 0
1 1 1 0 0 1 1 1 0
1 1 1 0 0 1 1 1 0
1 1 1 0 0 1 1 1 0
1 1 1 0 0 1 1 1 0
1 1 1 0 0 1 1 1 0
0 1 1 1 0 0 1 1 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0];
```

Comillas

```
[0 0 0 1 1 1 0 0 0
0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 1 0 0
0 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 0 0 0
0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 0 0 0
0 0 1 1 1 0 0 0 0
0 0 1 0 0 0 0 0 0];
```

Punto y coma

Convertidos en un solo vector con los valores 1 y -1, los patrones se buscaran de la siguiente manera:

```
Signo de interrogacion = [-1 -1 1 1 1 1 1 -1 -1 -1 1 1 1 1 1 1 -1 -1 1 1 -1 -1 -1 1 1 -1 1 1
-1 -1 -1 -1 1 1 -1 1 1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 1
1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 1 1
-1 -1 -1];
```

```
Corchete = [1 1 1 1 1 1 1 1 -1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 1 -1 -1 -1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1
-1 -1 1 -1 -1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1 1 1 -1];
```

Llave = [ -1 -1 1 1 1 1 1 1 -1 -1 -1 1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 1 1  
 -1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 1 1 -1  
 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 1 1 1 1 1 1 -1 -1 -1 1 1 1 1 1 -1];

Diagonal = [ 1 1 1 -1 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 1 1 -1 -1  
 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 1 1 1 -1 -1  
 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 1 1 1 -1];

Comillas = [ -1 1 1 1 -1 -1 1 1 1 1 1 1 -1 -1 1 1 1 -1 1 1 1 -1 -1 1 1 1 -1 1 1 1 -1 -1 1 1 1 -1 1  
 1 1 -1 -1 1 1 1 -1 1 1 1 -1 -1 1 1 1 -1 1 1 1 -1 -1 1 1 1 -1 1 1 1 -1 -1 1 1 1 -1 -1 1 1 1  
 -1 -1 1 1 1 -1 -1 1 1 1 -1];

Punto y coma = [ -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 1 -1 -1 -1 -1 1 1 1  
 1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 1 1 1 1 1 -1  
 -1 -1 -1 1 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1];



# Conclusión

Finalmente, para concluir con este reporte, yo considero que este tema en general es de suma importancia, debido a que una red neuronal puede ser aplicada para muchas cosas e investigaciones o problemas complejos que pueden solucionarse con esto, aunque debemos dejar claro que no todos los problemas con un alto nivel de complejidad podrían resolverse, debemos tomar en cuenta todos los factores que influyen al momento de utilizar una red neuronal o llevar a cabo una investigación mediante patrones, esto nos puede ser muy útil para encontrar secuencias en alguna cadena, en el campo de la medicina podría resultar muy útil trabajar con redes neuronales que aprendan o memoricen imágenes de alguna radiografía o rayos x, esto puede sonar muy complejo pero actualmente ya existe, lo que quiere decir que la aplicación de las redes neuronales va aumentando gracias a los avances tecnológicos y principalmente para facilitar este tipo de análisis o investigaciones.