

Inteligencia Artificial

Unidad 2: Redes Neuronales

TEMA 3: Algoritmos de IA Moderna-I

Módulo 5: Redes Recurrentes Hopfield

Unidad 2

Redes Neuronales

TEMA 3: Algoritmos de IA Moderna-I

Sesión 20

MÓDULO 5: Redes Recurrentes Hopfield



Contenido

1. ¿Qué son las redes recurrentes Hopfield?
2. Estructura y funcionamiento
3. ¿Cómo aprenden las redes recurrentes Hopfield?
4. Ejemplos de Redes Hopfield
5. Redes Hopfield: ventajas y desventajas



Preguntas

1. ¿Qué son las Redes Recurrentes Hopfield?

Historia - Redes Recurrentes Hopfield

- Las **memorias asociativas** son uno de los primeros modelos neuronales artificiales que se remontan a las décadas de 1960 y 1970.
- Las más conocidas son las **redes Hopfield**, presentadas por John Hopfield en 1982.
- Como su nombre indica, el propósito principal de las redes de memoria asociativa es asociar una entrada con su patrón más similar.
- En otras palabras, el propósito es almacenar y recuperar patrones.

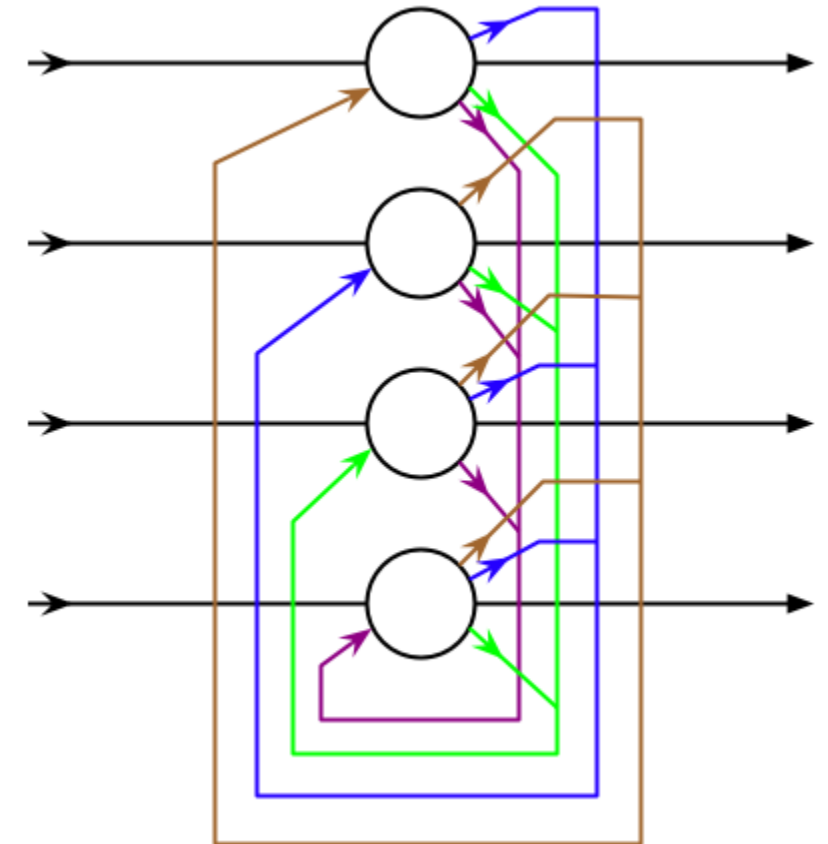


Profesor John Joseph Hopfield

1. ¿Qué son las Redes Recurrentes Hopfield?

Generalidades - Redes Recurrentes Hopfield

- **Hopfield Neural Network (HNN)** es una red neuronal UNICAPA con características cíclicas y recursivas, combinada con sistemas de almacenamiento y binarios.
- Para una red neuronal Hopfield, la clave es determinar su peso en condiciones estables.
- Las redes neuronales de Hopfield se dividen en tipos discretos y continuos. La principal diferencia radica en la función de activación.
- La Red de Hopfield es una red recurrente, es decir, existe realimentación entre las neuronas. De esta forma, al introducir un patrón de entrada, la información se propaga hacia adelante y hacia atrás, produciéndose una dinámica. En algún momento, la evolución se detendrá en algún estado estable.

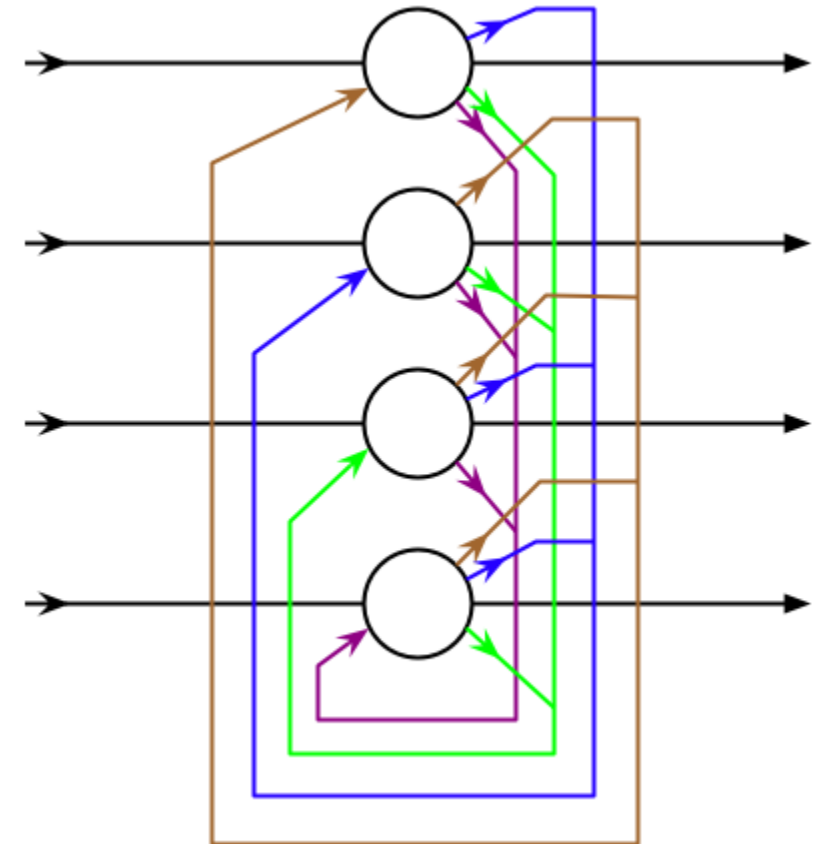


Una red Hopfield con cuatro neuronas

1. ¿Qué son las Redes Recurrentes Hopfield?

Como Memoria Asociativa

- Las redes de Hopfield se usan como sistemas de memoria asociativa con unidades binarias.
- Están diseñadas para converger a un mínimo local, pero la convergencia a uno de los patrones almacenados no está garantizada.
- La red de Hopfield es una red monocapa (unicapa), esto es, de una sola capa. Aunque también se puede mostrar como una red bicapa (de dos capas), la primera capa sería una capa de sensores y la segunda capa será la capa donde se realiza el procesamiento.

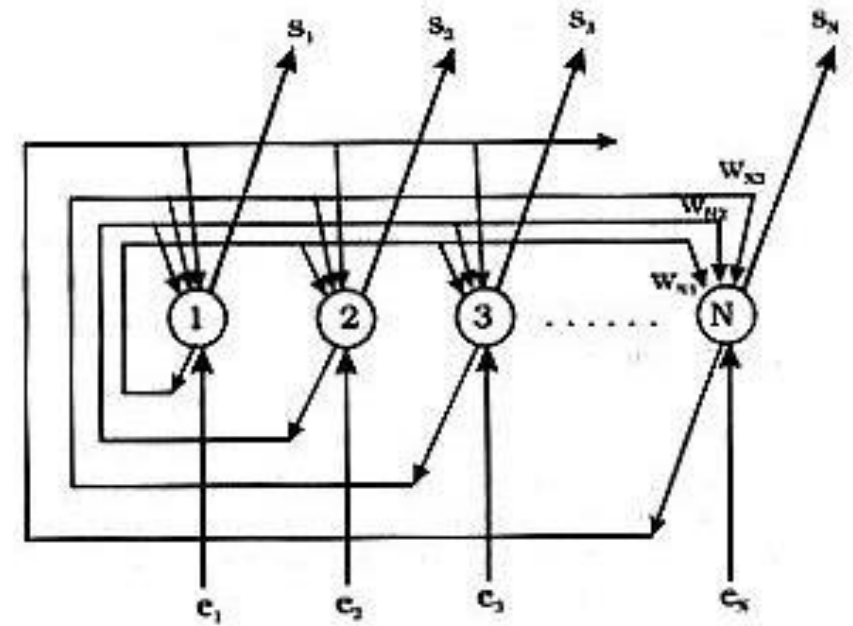


Una red Hopfield con cuatro neuronas

1. ¿Qué son las Redes Recurrentes Hopfield?

Como Unidad Procesadora

- Las unidades procesadoras de la red Hopfield están completamente interconectadas, cada unidad está conectada con todas las demás unidades.
- Esta topología convierte a la **red Hopfield** en una **red recursiva** ya que la salida de cada unidad está realimentada con las entradas de las demás unidades.
- Además los pesos asignados a ambas conexiones tienen el mismo valor.



1. ¿Qué son las Redes Recurrentes Hopfield?

Utilidad – Redes Hopfield

- Las Redes Hopfield tiene una amplia gama de aplicaciones en inteligencia artificial como:
 - ❖ Aprendizaje automático (Memoria asociativa)
 - ❖ Reconocimiento de patrones (de imágenes y de voz)
 - ❖ Cálculo optimizado: resolución de problemas de optimización (problema del viajero, resolución de ecuaciones, proceso de señales de analógicas a digitales) , etc.

Las Redes Hopfield generalmente se utilizan para realizar tareas de optimización y asociación automática.

- Se entrenan utilizando el aprendizaje no supervisado.

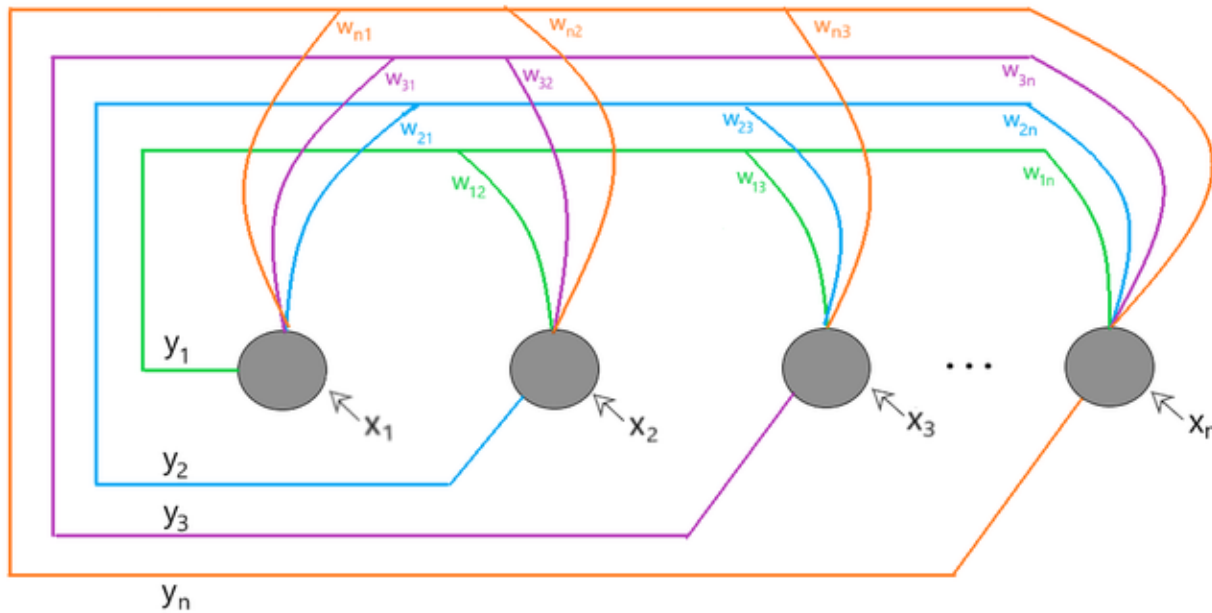
Pero, ¿Qué son y para qué se utilizan estas redes?



2. Estructura y funcionamiento

Estructura de una Red Hopfield

(red neuronal recurrente unicapa discreta)



Elementos:

$[x_1, x_2, \dots, x_n] \rightarrow$ Entrada a las n neuronas dadas.

$[y_1, y_2, \dots, y_n] \rightarrow$ Salida obtenida de las n neuronas dadas.

$w_{i,j} \rightarrow$ Peso asociado con la conexión entre la i -ésima y la j -ésima neurona.

- ❖ La red Hopfield es una red neuronal cíclica con conexiones de retroalimentación de salida a entrada.
- ❖ Al estar completamente conectada, la salida de cada neurona es una entrada para todas las demás neuronas, pero no para sí misma.
- ❖ Los pesos asociados con esta red son de naturaleza simétrica y tienen las siguientes propiedades.

$$w_{i,j} = w_{j,i}$$

$$w_{i,i} = 0$$

2. Estructura y funcionamiento

Tipos de Redes Hopfield

Red Hopfield discreta

- Es una red neuronal completamente interconectada donde cada unidad está conectada a todas las demás unidades.
- Se comporta de manera **discreta**, es decir, da una salida distinta finita, generalmente de dos tipos:
 - Binario (0/1)
 - Bipolar (-1/1)
- Los pesos asociados con esta red son de naturaleza simétrica y tienen las siguientes propiedades:

$$W_{i,j} = W_{j,i}$$

$$W_{i,i} = 0$$

Red Hopfield continua

- Es una red neuronal completamente interconectada donde cada unidad está conectada a todas las demás unidades.
- En lugar de obtener salidas binarias / bipolares, podemos obtener **valores que se encuentran entre 0 y 1**.
- La salida de este tipo de red Hopfield se define como:

$$v_i = g(u_i)$$

Donde:

v_i = salida de la red de campo de salto continuo

u_i = actividad interna de un nodo en la red de campo de salto continuo.

2. Estructura y funcionamiento

Funcionamiento de una Red Hopfield

1. Se establece el patrón de entrada en la capa de entrada.
2. Se actualizan las neuronas de la capa de procesamiento.
3. Mientras no haya cambiado el estado de la red o hayamos alcanzado el número máximo de iteraciones.
 4. Volvemos al paso 2.

Ejemplo:

Si entrenamos una red con cinco unidades (neuronas) para que el estado (1, 0, 1, 0, 1) sea un mínimo de energía, y le damos a la red el estado (1, 0, 0, 0, 1) esta convergerá a (1, 0, 1, 0, 1).

Así, la red estará adecuadamente capacitada cuando la energía de los estados que la red debe recordar son mínimos locales.

3. ¿Cómo aprenden las redes recurrentes Hopfield?

ALGORITMO DE ENTRENAMIENTO

- Se calculan los valores de los pesos que conectan a los nodos, utilizando la siguiente fórmula.

$$t_{ij} = \begin{cases} \sum_{s=0}^{m-1} x_{is}x_{js} & \text{si } i \neq j \\ 0 & \text{si } i = j \end{cases}$$



Donde:

- t_{ij} es el peso que va de la neurona i a la neurona j , y es el valor del i -ésimo elemento de la s -ésima clase
- m es el número de clases que se desean aprender

- En notación matricial:

$$T = \sum_i x_i^T x_i, t_{ii} = 0$$



A esta fórmula se le conoce como el **producto externo de un vector renglón consigo mismo**.

3. ¿Cómo aprenden las redes recurrentes Hopfield?

ALGORITMO DE EVALUACION

- Inicializar la red con un patrón de entrada:

$$U_i(0) = X_i \quad 0 \leq i \leq n-1$$



Donde **n** es el número de nodos en la red

- Iterar hasta converger siguiendo la siguiente fórmula:

$$U_j(t+1) = F\left(\sum_{i=0}^{n-1} t_{ij} U_i(t)\right) \quad 0 \leq j \leq n-1$$



Donde F es una función escalón definida como:

$$F(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si } x < 0 \\ U_i(t) & \text{si } x = 0 \text{ (sin cambio)} \end{cases}$$

- Cuando la red converge, su salida representa al patrón más se parecido al patrón de entrada dado.

4. Ejemplos de Redes Hopfield

Enunciado #1

- a) Crear una red de Hopfield discreta con representación bipolar del **vector de entrada** cuando $[1 \ 1 \ 1 \ -1]$ o $[1 \ 1 \ 1 \ 0]$ (en caso de una representación binaria) se almacene en la red.
- b) Probar la red de Hopfield con entradas faltantes en el primer y segundo componente del vector almacenado (es decir, $[0 \ 0 \ 1 \ 0]$).

DATOS

Vector de entrada:

$x = [1 \ 1 \ 1 \ -1]$ (bipolar)

$x = [1 \ 1 \ 1 \ 0]$ (binaria)

Solución

Paso #1: Se establece el patrón de entrada en la capa de entrada.

Inicializamos la matriz de peso ($W_{i,j}$) como:

$$W_{i,j} = T = \sum_i x_i^T x_i, t_{ii} = 0$$

$$W_{i,j} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} [1 \ 1 \ 1 \ -1]$$

$$W_{i,j} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

Para patrones bipolares

La matriz de peso sin conexión propia es :

$W_{i,j} = 0$ para todo $i = j$

$$W_{i,j} = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

4. Ejemplos de Redes Hopfield

Enunciado #1

- a) Crear una red de Hopfield discreta con representación bipolar del **vector de entrada** cuando $[1 \ 1 \ 1 \ -1]$ o $[1 \ 1 \ 1 \ 0]$ (en caso de una representación binaria) se almacene en la red.
- b) Probar la red de Hopfield con entradas faltantes en el primer y segundo componente del vector almacenado (es decir, $[0 \ 0 \ 1 \ 0]$).

DATOS

Vector de entrada:

$x = [1 \ 1 \ 1 \ -1]$ (bipolar)

$x = [1 \ 1 \ 1 \ 0]$ (binaria)

a) Matriz de peso inicializada:

$$W_{i,j} = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

b) Probar con $x = [0 \ 0 \ 1 \ 0]$

$y_i = x = [0 \ 0 \ 1 \ 0]$ ($[y_1, y_2, y_3, y_4]$)

Paso #2: Se actualizan las neuronas de la capa de procesamiento.

- Elegir la unidad y_i (el orden no importa) para actualizar su activación.
- Tomar la i -ésima columna de la matriz de peso para el cálculo (haremos los siguientes pasos para todos los valores de y_i y comprobaremos si hay convergencia o no)

Aplicamos la función de activación:

$$y_{in_1} = x_1 + \sum_{j=1}^4 [y_j w_{j1}] = 0 + [0 \ 0 \ 1 \ 0] \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} = 0 + 1 = 1 \quad \rightarrow \quad y_1 > 0 ?$$

$y_1 = 1$

Reemplazamos y_1 que es 0 por 1 $\rightarrow y = [1010]$

$x = [1110]$ } $y = x$ Convergenencia?

NO = Para la próxima unidad y_i , tomaremos el valor actualizado es decir, $y = [1 \ 0 \ 1 \ 0]$

4. Ejemplos de Redes Hopfield

Enunciado #1

- a) Crear una red de Hopfield discreta con representación bipolar del **vector de entrada** cuando $[1 \ 1 \ 1 \ -1]$ o $[1 \ 1 \ 1 \ 0]$ (en caso de una representación binaria) se almacene en la red.
- b) Probar la red de Hopfield con entradas faltantes en el primer y segundo componente del vector almacenado (es decir, $[0 \ 0 \ 1 \ 0]$).

DATOS

Vector de entrada:

$x = [1 \ 1 \ 1 \ -1]$ (bipolar)

$x = [1 \ 1 \ 1 \ 0]$ (binaria)

a) Matriz de peso inicializada:

$$W_{i,j} = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

b) Probar con $x = [1010]$:

$y_i = x = [1 \ 0 \ 1 \ 0]$ ($[y_1, y_2, y_3, y_4]$)

Paso #2.1: Se actualizan las neuronas de la capa de procesamiento.

Aplicamos la función de activación:

$$y_{in3} = x_3 + \sum_{j=1}^4 [y_j w_{j3}] = 1 + \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \end{bmatrix} = 1 + 1 = y_3 \quad \begin{matrix} y_3 > 0 ? \\ y_3 = 1 \end{matrix}$$

No reemplazamos y_3 ya que es 1 $\rightarrow y = [1010]$

$x = [1110]$

$y = x$ Convergencia?

NO = Para la próxima unidad y_i , tomaremos el valor actualizado es decir, $y = [1 \ 0 \ 1 \ 0]$

4. Ejemplos de Redes Hopfield

Enunciado #1

- a) Crear una red de Hopfield discreta con representación bipolar del **vector de entrada** cuando $[1 \ 1 \ 1 \ -1]$ o $[1 \ 1 \ 1 \ 0]$ (en caso de una representación binaria) se almacene en la red.
- b) Probar la red de Hopfield con entradas faltantes en el primer y segundo componente del vector almacenado (es decir, $[0 \ 0 \ 1 \ 0]$).

DATOS

Vector de entrada:

$x = [1 \ 1 \ 1 \ -1]$ (bipolar)

$x = [1 \ 1 \ 1 \ 0]$ (binaria)

a) Matriz de peso inicializada:

$$W_{i,j} = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

b) Probar con $x = [1010]$:

$y_i = x = [1 \ 0 \ 1 \ 0]$ ($[y_1, y_2, y_3, y_4]$)

Paso #2.2: Se actualizan las neuronas de la capa de procesamiento.

Aplicamos la función de activación:

$$y_{in4} = x_4 + \sum_{j=1}^4 [y_j w_{j4}] = 0 + \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

$= 0 + 1(-1) + 1(-1) = -2$

$y_4 < 0 ?$
 $y_4 = 0$

No reemplazamos y_4 ya que es 0 $\rightarrow y = [1010]$
 $x = [1110]$ } $y = x$ Convergencia?

NO = Para la próxima unidad y_i , tomaremos el valor actualizado es decir, $y = [1 \ 0 \ 1 \ 0]$

4. Ejemplos de Redes Hopfield

Enunciado #1

- a) Crear una red de Hopfield discreta con representación bipolar del **vector de entrada** cuando $[1 \ 1 \ 1 \ -1]$ o $[1 \ 1 \ 1 \ 0]$ (en caso de una representación binaria) se almacene en la red.
- b) Probar la red de Hopfield con entradas faltantes en el primer y segundo componente del vector almacenado (es decir, $[0 \ 0 \ 1 \ 0]$).

DATOS

Vector de entrada:

$x = [1 \ 1 \ 1 \ -1]$ (bipolar)

$x = [1 \ 1 \ 1 \ 0]$ (binaria)

a) Matriz de peso inicializada:

$$W_{i,j} = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

b) Probar con $x = [1010]$:

$y_i = x = [1 \ 0 \ 1 \ 0]$ ($[y_1, y_2, y_3, y_4]$)

Paso #2.3: Se actualizan las neuronas de la capa de procesamiento.

Aplicamos la función de activación:

$$y_{in_2} = x_2 + \sum_{j=1}^4 [y_j w_{j2}] = 0 + [1 \ 0 \ 1 \ 0] \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \end{bmatrix}$$

$= 0 + 1 + 0 + 1 + 0 = 2$

$y_2 > 0 ?$
 $y_2 = 1$

Reemplazamos y_2 que es 0 por 1 $\rightarrow y = [1110]$

$x = [1110]$

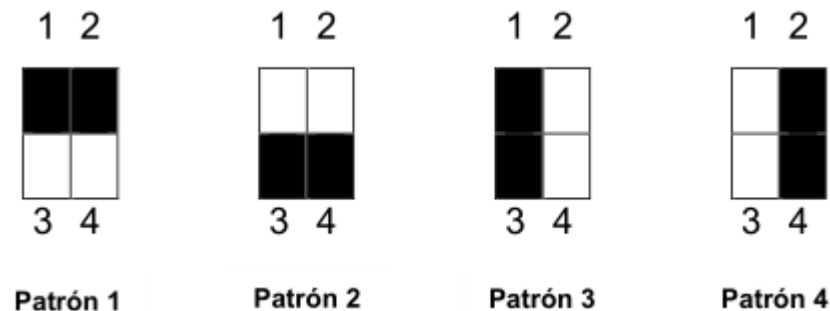
$y = x$ Convergencia?

SI = Terminamos porque hay convergencia.

4. Ejemplos de Redes Hopfield

Enunciado #2

- Se desea entrenar una Red de Hopfield que sea capaz de reconocer información (patrones) de imágenes formadas por cuatro pixeles, en una matriz de 2x2.
- En la figura siguiente se muestran ejemplos de patrones que podrían utilizarse como entradas a la red:



- Los pixeles negros podrían representarse mediante el valor binario 1, y los blancos con el valor binario -1. En este caso, las informaciones serían vectores de cuatro elementos ($N = 4$) que contienen los valores de los pixeles.
- La red, por tanto, tendría 4 neuronas para que cada una reciba el valor de un pixel.
- Aplicar el algoritmo de aprendizaje de la Red de Hopfield para los patrones 1 y 2.

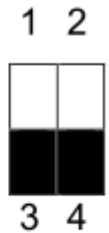
4. Ejemplos de Redes Hopfield

Enunciado #2 – SOLUCION

1. Aplicaremos para los patrones de entrada 1 y 2



Patrón 1



Patrón 2



$$E_1 = [1, 1, -1, -1] \text{ y } E_2 = [-1, -1, 1, 1]$$

2. El aprendizaje de estas dos informaciones consiste en la obtención de los pesos de la red (matriz W). Utilizaremos la fórmula:

$$W_{i,j} = T = \sum_i x_i^T x_i, t_{ii} = 0$$

Aprendizaje:

Paso #1: Se establece el patrón de entrada en la capa de entrada. Inicializamos la matriz de peso ($W_{i,j}$) como:

Para la entrada E_1 , la salida W_1 es:

$$W_1 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix}$$

Para la entrada E_2 , la salida W_2 es:

$$W_2 = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix}$$

4. Ejemplos de Redes Hopfield

Enunciado #2 - SOLUCION

Aprendizaje:

Paso #1: Continuación

Sumando W_1 y W_2 se obtiene la matriz de pesos definitiva, W :

$$W = W_1 + W_2 = \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix} \Rightarrow W = \begin{bmatrix} 0 & 2 & -2 & -2 \\ 2 & 0 & -2 & -2 \\ -2 & -2 & 0 & 2 \\ -2 & -2 & 2 & 0 \end{bmatrix}$$

- Una vez finalizada la fase de aprendizaje (entrenamiento), la red podría ser utilizada como memoria asociativa.

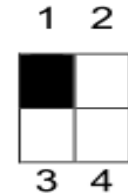
4. Ejemplos de Redes Hopfield

Enunciado #2 - SOLUCION

EVALUACION:

Paso #2:

Por ejemplo, se podría comprobar lo que ocurre con el patrón de entrada siguiente: El vector es $E = [1, -1, -1, -1]$



$$E * W = \begin{bmatrix} 1 & -1 & -1 & -1 \end{bmatrix} * \begin{bmatrix} 0 & 2 & -2 & -2 \\ 2 & 0 & -2 & -2 \\ -2 & -2 & 0 & 2 \\ -2 & -2 & 2 & 0 \end{bmatrix} \rightarrow W = \begin{bmatrix} 0 & 2 & -2 & -2 \\ -2 & 0 & 2 & 2 \\ 2 & 2 & 0 & -2 \\ 2 & 2 & -2 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 6 & -2 & -2 \end{bmatrix}$$

Si suponemos una función de activación de cada neurona de tipo escalón

$$S = [1, 1, -1, -1]$$

$E = S$ Convergencia? **NO**, seguimos iterando con el nuevo $E = [1, 1, -1, -1]$

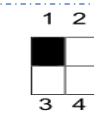
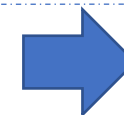
Paso #2.1:

$$E * W = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix} * \begin{bmatrix} 0 & 2 & -2 & -2 \\ 2 & 0 & -2 & -2 \\ -2 & -2 & 0 & 2 \\ -2 & -2 & 2 & 0 \end{bmatrix} \rightarrow W = \begin{bmatrix} 0 & 2 & -2 & -2 \\ 2 & 0 & -2 & -2 \\ 2 & 2 & 0 & -2 \\ 2 & 2 & -2 & 0 \end{bmatrix} = \begin{bmatrix} 6 & 6 & -6 & -6 \end{bmatrix}$$

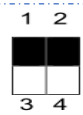
$$S = [1, 1, -1, -1]$$

$E = S$ Convergencia? **NO**, seguimos iterando con el nuevo $E = [1, 1, -1, -1]$

Observamos que se repite la salida de la primera iteración, entonces se ha llegado a una situación de estabilidad.



Entrada



Salida generada

4. Ejemplos de Redes Hopfield

Enunciado #2 – SOLUCION PARA SOLO 1 PATRON DE ENTRADA

1. Aplicaremos para el patrones de entrada 1



$$E_1 = [1, 1, -1, -1] \text{ Bipolar}$$

$$E_1 = [1, 1, 0, 0] \text{ Binaria}$$

2. El aprendizaje de esta entrada consiste en la obtención de los pesos de la red (matriz W_1). Utilizaremos la fórmula:

$$W_{i,j} = T = \sum_i x_i^T x_i, t_{ii} = 0$$

Aprendizaje:

Paso #1: Se establece el patrón de entrada en la capa de entrada. Inicializamos la matriz de peso ($W_{i,j}$) como:

Para la entrada E_1 , la salida W_1 es:

$$W_1 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$



La matriz de peso sin conexión propia es :

$$W_{i,j} = 0 \text{ para todo } i = j$$

$$W_1 = \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix}$$

4. Ejemplos de Redes Hopfield

Enunciado #2 – SOLUCION PARA SOLO 1 PATRON DE ENTRADA

EVALUACION:

Paso #2: Comprobar lo que ocurre con el patrón de entrada siguiente:



El vector a probar es :

$E = [1, -1, -1, -1]$ (bipolar)

$E = [1, 0, 0, 0]$ (binario)

DATOS

Vector de entrada:

$x = [1, 1, -1, -1]$ (bipolar)

$x = [1, 1, 0, 0]$ (binaria)

a) Matriz de peso inicializada:

$$W_1 = \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix}$$

b) Probar con $x = [1000]$:

$y_i = x = [1, 0, 0, 0]$ ($[y_1, y_2, y_3, y_4]$)

Paso #2.1: Se actualizan las neuronas de la capa de procesamiento.

- Elegir la unidad y_i (el orden no importa) para actualizar su activación.
- Tomar la i -ésima columna de la matriz de peso para el cálculo (haremos los siguientes pasos para todos los valores de y_i y comprobaremos si hay convergencia o no)

Aplicamos la función de activación:

$$y_{in1} = x_1 + \sum_{j=1}^4 [y_j w_{j1}] = 1 + [1 \ 0 \ 0 \ 0] \begin{bmatrix} 0 \\ 1 \\ -1 \\ -1 \end{bmatrix} = 1 + 0 = 1 \rightarrow y_1 > 0 ?$$

$y_1 = 1$

No reemplazamos ya que y_1 que es 1 $\rightarrow y = [1000]$
 $x = [1100]$ } $y = x$ Convergencia?

NO = Para la próxima unidad y_i , tomaremos el valor actualizado es decir, $y = [1, 0, 0, 0]$

4. Ejemplos de Redes Hopfield

Enunciado #2 – OTRA SOLUCION

EVALUACION:

Paso #2: comprobar lo que ocurre con el patrón de entrada siguiente:



El vector a probar es :

$E = [1, -1, -1, -1]$ (bipolar)

$E = [1, 0, 0, 0]$ (binario)

DATOS

Vector de entrada:

$x = [1 \ 1 \ -1 \ -1]$ (bipolar)

$x = [1 \ 1 \ 0 \ 0]$ (binaria)

a) Matriz de peso inicializada:

$$W_1 = \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix}$$

b) Probar con $x = [1000]$:

$y_i = x = [1 \ 0 \ 0 \ 0]$ ($[y_1, y_2, y_3, y_4]$)

Paso #2.2: Se actualizan las neuronas de la capa de procesamiento.

Aplicamos la función de activación:

$$y_{in_2} = x_2 + \sum_{j=1}^4 [y_j w_{j2}] = 0 + [1 \ 0 \ 0 \ 0] \begin{bmatrix} 1 \\ 0 \\ -1 \\ -1 \end{bmatrix} = 0 + 1 = 1 \rightarrow y_1 > 0 ?$$

Reemplazamos y_2 que es 0 por 1 $\rightarrow y = [1100]$
 $x = [1100]$ } $y = x$ Convergencia?

SI = Terminamos porque hay convergencia.

5. Redes Hopfield: ventajas y desventajas

VENTAJAS

- No existe tiempo de entrenamiento, ya que este no es un proceso adaptativo, sino simplemente el cálculo de una matriz (T).
- Las redes de Hopfield son bastante tolerantes al ruido, cuando funcionan como memorias asociativas.

DESVENTAJAS

- Número limitado de entradas en la etapa de aprendizaje: Si se almacena demasiada información, durante su funcionamiento, la red puede converger a valores de salida diferentes de los aprendidos.
- Si los valores de salida fueran diferentes a los aprendidos, entonces, la tarea de asociación entre la información presentada y alguna de las almacenadas se realiza incorrectamente.
- El número de patrones a almacenar (o aprender) es bastante limitado comparado con el número de nodos en la red.
- La red se vuelve inestable si los patrones se parecen entre sí.

PREGUNTAS

Dudas y opiniones