Inteligencia Artificial
Unidad 3
Redes Neuronales Recurrentes

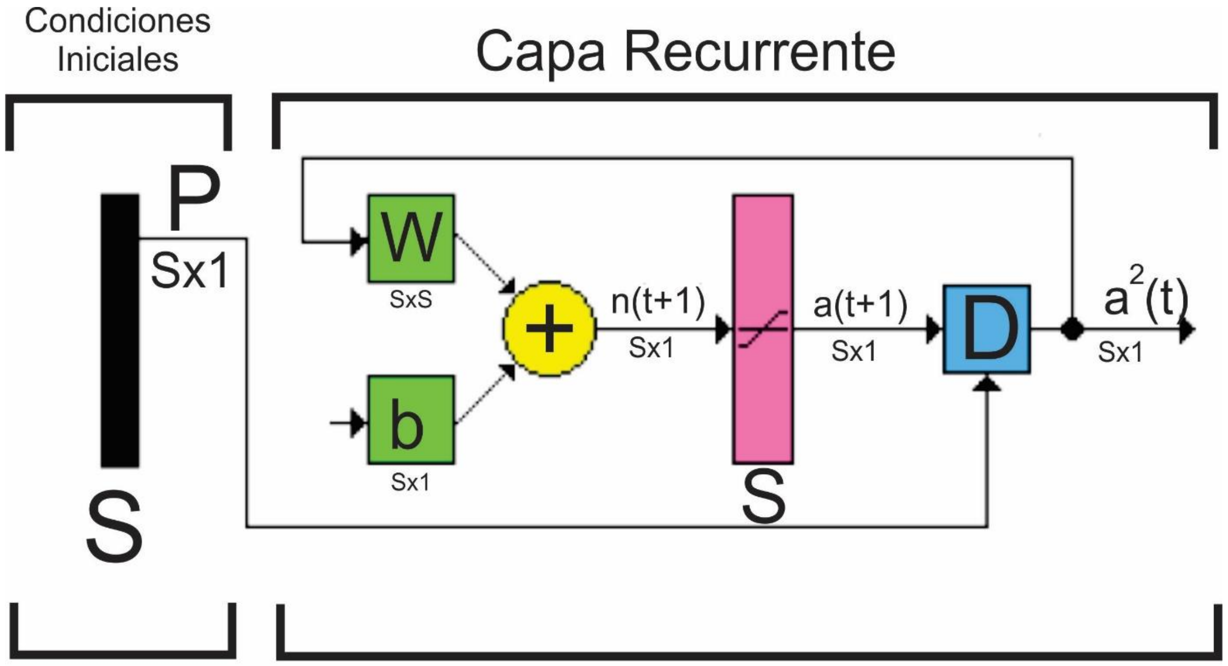
REDES HOPFIELD

- Hugo David Calderón
- Heider Sanchez Enriquez
- Willy Ugarte

Redes Recurrentes

Generalidades

Las Redes Recurrentes contienen una realimentación hacia atrás o retroalimentación, es decir algunas de sus salidas son conectadas a sus entradas, el entrenamiento de una red neuronal recurrente debe prolongarse para cada paso temporal.



Redes de Hopfield

Generalidades

Una red de Hopfield es una forma de red neuronal artificial recurrente inventada por John Hopfield.

La red de Hopfield es una de las redes unicapas de neuronas más importantes cuyas salidas son números binarios; y ha influido en el desarrollo de multitud de redes posteriores.

Cada neurona de la red se encuentra conectada con todas las demás, pero no consigo mismo.

Es una red autoasociativa no lineal que fue desarrollada por Hopfield en 1982, basándose en los modelos de redes de McCulloch y Pitts

Redes de Hopfield

Generalidades

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.

Redes de 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.

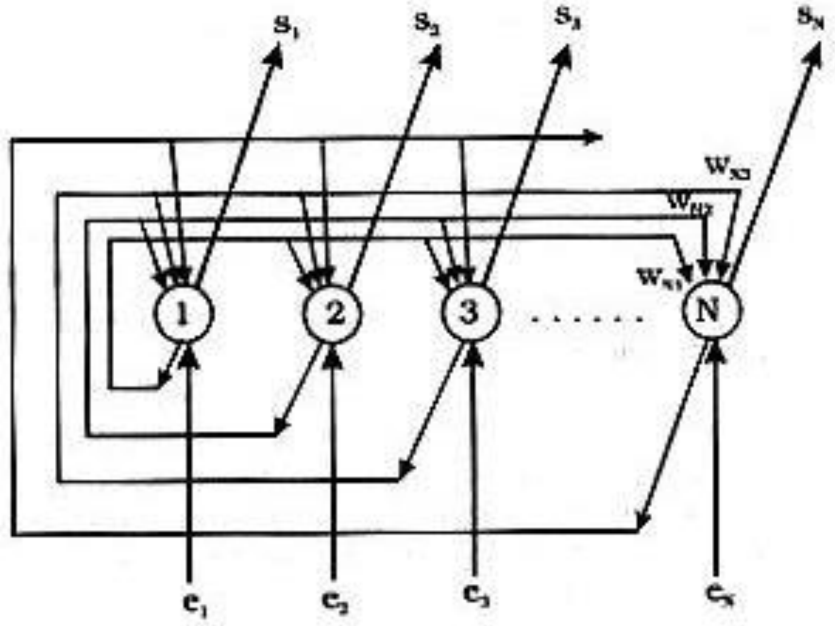
La red de Hopfield es una red monocapa, esto es, de una sola capa. Aunque también se puede mostrar como una red bicapa de dos capas, la primera capa sería una capa de sensores y la segunda capa será la capa donde se realiza el procesamiento.

Redes de Hopfield

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



Redes de Hopfield

Son pasos del funcionamiento de la red

1. Se establece el patrón de entrada en la capa de entrada.
2. Se actualizan las neuronas de la capa de procesamiento.
3. Si han cambiado el estado de la red o hemos realizado ya el número máximo de iteraciones, paramos.
4. Si no volvemos al paso 2.

Ejemplo: Si entrenamos una red con cinco unidades 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.

Redes de Hopfield

Algoritmo de entrenamiento

Calcule los valores de los pesos que conectan a los nodos, utilizando la siguiente fórmula.

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

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:

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

Esta fórmula se conoce como el producto externo de un vector renglón consigo mismo.

Redes de Hopfield

Algoritmo de evaluación de la red Hopfield.

- Inicialice la red con un patrón de entrada:

$$Ui\ (0) = Xi \quad 0 \leq i \leq n-1$$

Donde n es el número de nodos en la red

- Itere hasta converger siguiendo la siguiente fórmula:

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

Donde F es una función escalón definida como:

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

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

Redes de Hopfield

Función de Energía

El conjunto total del sistema puede venir representado por una función denominada Función de Energía de la siguiente manera:

$$E_i = -\frac{1}{2} h_i x_i$$

$$E(x) = \sum_i E_i = -\sum_i \frac{1}{2} h_i x_i = -\frac{1}{2} \sum_i w_{ij} x_i x_j$$

Memoria asociativa

- El entrenamiento de una red de Hopfield consiste en reducir la energía de los estados que la red debe “recordar”. Esto convierte a la red en un sistema de memoria direccionable, es decir, la red “recordará” un estado si se le da solo parte de dicho estado.

Redes de Hopfield

Ventajas

Prácticamente 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.

Redes de Hopfield

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, con lo que la tarea de asociación entra 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. Según Hopfield, el número de clases a aprender no puede ser mayor del 15% del número de nodos en la red.

La red se vuelve inestable si los patrones se parecen entre sí.

Redes de Hopfield

Aplicaciones

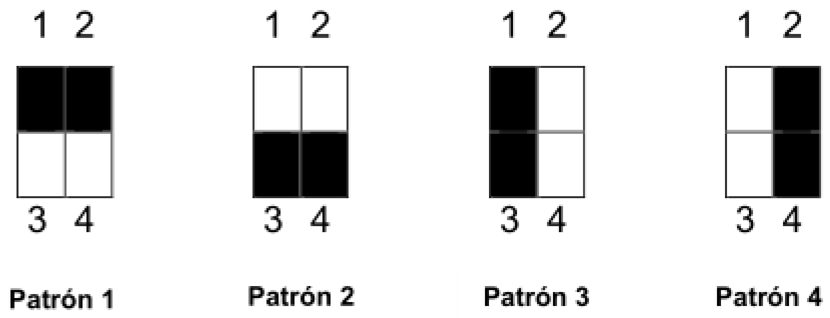
En cuanto a las aplicaciones más conocidas de este modelo destacan las relacionadas con el reconocimiento de patrones (de imágenes y de voz), el control de motores y sobre todo la resolución de problemas de optimización.

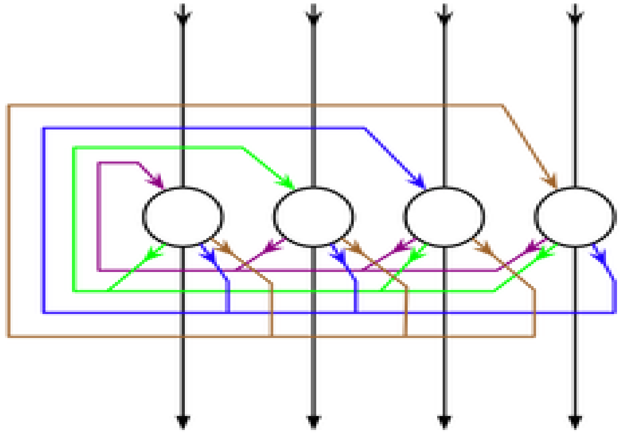
Para problemas de optimización se ha aplicado para la resolución de manipulación de grafos, por ejemplo el problema del viajante vendedor; resolución de ecuaciones, procesado de señales (conversores analógico-digitales) y de imágenes, etc.

Ejemplo

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.

Solución

Aplicaremos para los patrones de entrada 1 y 2

E 1 = [1, 1,-1,-1] y E 2 = [-1,-1, 1, 1]

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

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

Para la entrada E1, la salida W 1 es:

$$TE_1 . E_1 - I = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix}$$

Para la entrada E2, la salida W 2 es:

$$TE_2 . E_2 - I = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix}$$

Solución

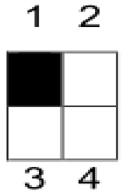
Sumando W 1 y W 2 se obtiene la matriz de pesos definitiva, W:

$$W = \sum_{k=1}^{2}[E_k^T E_k - I] = \begin{bmatrix} 0 & 2 & -2 & -2 \\ 2 & 0 & -2 & -2 \\ -2 & -2 & 0 & 2 \\ -2 & -2 & 2 & -0 \end{bmatrix}$$

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

Evaluación

Por ejemplo, ser podría comprobar lo que ocurre con el patrón de entrada siguiente:



El vector es E = [1, -1, -1, -1]

Solución

Evaluación

El vector es E = [1, -1, -1, -1]

Inicialmente, la salida corresponde a la propia información de entrada.

$$E.W = [1 \;\; -1 \;\; -1 \;\; -1]\begin{bmatrix} 0 & 2 & -2 & -2 \\ 2 & 0 & -2 & -2 \\ -2 & -2 & 0 & 2 \\ -2 & -2 & 2 & 0 \end{bmatrix} = [2 \;\; 6 \;\; -2 \;\; -2]$$

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

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

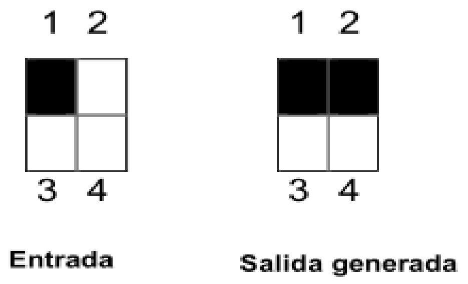
Repitiendo el proceso, tomando como entrada la salida anterior, S, tenemos:

$$E.W = [1 \;\; 1 \;\; -1 \;\; -1]\begin{bmatrix} 0 & 2 & -2 & -2 \\ 2 & 0 & -2 & -2 \\ -2 & -2 & 0 & 2 \\ -2 & -2 & 2 & 0 \end{bmatrix} = [6 \;\; 6 \;\; -6 \;\; -6]$$

Aplicando igualmente la función de activación

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

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



FIN