Redes de Hopfield I

Modelos de la computación (Redes Recurrentes y Autónomas)

Francisco Fernández Navarro

Departamento de Lenguajes y Ciencias de la Computación
Área: Ciencias de la Computación e Inteligencia Artificial

Índice de contenidos

Redes de Hopfield

- Las Redes de Hopfield son un tipo de red neuronal recurrente.
- Desarrolladas por John Hopfield en 1982.
- Utilizadas para resolver problemas de optimización y reconocimiento de patrones.
- Modelan la memoria asociativa y pueden recuperar patrones completos a partir de fragmentos.

Comparativa red feedfordward versus red recurrente

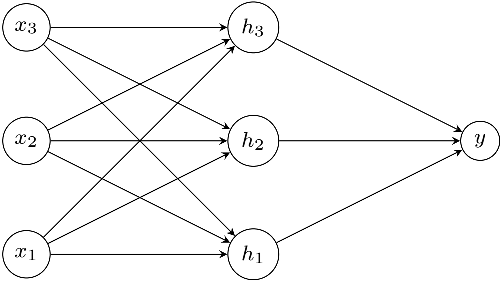


Figure: Red Neuronal Feedforward

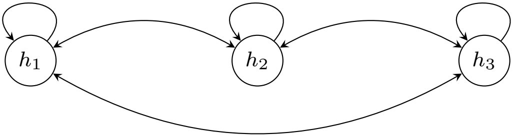


Figure: Red Neuronal Recurrente

Índice de contenidos

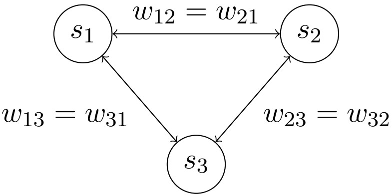
Modelo funcional

Definición

- Una red de Hopfield bipolar está constituida por K unidades de proceso bipolares completamente conectadas entre sí
- La matriz de pesos sinápticos: matriz simétrica ($w_{ij} = w_{ji}, i, j \in \{1, \dots, K\}$) con los elementos de la diagonal principal iguales a cero ($w_{kk} = 0, k \in \{1, \dots, K\}$, es decir, sin auto conexión)

Representación redes de Hopfield

Red de Hopfield con 3 nodos, s_1, s_2 y s_3, con matrices de pesos simétricos y sin auto conexión $w_{11} = w_{22} = w_{33} = 0$.



Índice de contenidos

Evolución de la red de Hopfield

Inicialización

Valores iniciales para $s_k(0) \in \{-1, 1\}$, donde la k indica el número de neurona, $k \in \{1, \ldots, K\}$, y el 0 señala que es la fase inicial de la evolución.

Actualización pesos, $i \in \{1, \ldots, I\}$

La salida de la unidad k-ésima en la iteración $i + 1$ es:

$$s_k(i+1) = \begin{cases} 1 & \text{si} \quad \sum_{j=1}^{K} w_{kj} s_j(i) \geq \theta_k \\ -1 & \text{si} \quad \sum_{j=1}^{K} w_{kj} s_j(i) < \theta_k \end{cases} \tag{1}$$

Condición de parada

Después de actualizar todas las unidades de proceso ninguna de ellas cambia su estado anterior, $s_k(i+1) = s_k(i), k \in \{1, \ldots, K\}$.

Tipos de actualizaciones en redes de Hopfield

- **Secuencial**: Vamos actualizando las neuronas de una en una empezando con una aleatoria.
- **Paralelo**: Actualizamos todas las neuronas a la vez.

Índice de contenidos

Objetivo del problema de optimización

Se pretende alcanzar concordancia entre los estados de las unidades de proceso según las conexiones sinápticas:
Pensemos en el potencial sináptico de la neurona i en su relación con la neurona j, es decir, la conexión w_{ij}

- Si $w_{ij} > 0$ y $s_j = 1$, el potencial crecerá, y el estado de s_i tenderá a ser 1
- Si $w_{ij} > 0$ y $s_j = -1$, el potencial decrecerá, y el estado de s_i tenderá a ser -1
- Si $w_{ij} < 0$ y $s_j = 1$, el potencial decrecerá, y el estado de s_i tenderá a ser -1
- Si $w_{ij} < 0$ y $s_j = -1$, el potencial crecerá, y el estado de s_i tenderá a ser 1

Los pesos sinápticos de la red, w_{ij} miden la correlación entre la unidad de proceso i y la unidad j

Problema de optimización

En la práctica la red busca maximizar el producto $w_{ij} \cdot s_i \cdot s_j$, por lo que el problema de optimización podría definirse como:

$$\max_{\mathbf{w} \in \mathbb{R}^{K \times K}} \quad \sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij} s_i s_j,$$

donde $\mathbf{w} \in \mathbb{R}^{K \times K}$ es la matriz de pesos sinápticos. Como cada conexión se ha contado dos veces, realmente nuestro objetivo será:

$$\max_{\mathbf{w} \in \mathbb{R}^{K \times K}} \quad \frac{1}{2} \sum_{i=1}^{K} \sum_{j=1}^{K} w_{ij} s_i s_j.$$

Problema de optimización

¿Que papel juegan entonces los sesgos?

Como señal externa (con valor -1) que llega a la unidad de proceso (con peso θ_i) de manera que si $\theta_i > 0$, tiende a desactivar s_i, y si $\theta_i < 0$, tiende a activar s_i

Si lo analizamos por casos de forma análoga a como hicimos con los pesos sinápticos, vemos que se persigue maximizar $\theta_i(-1)s_i$

Problema de optimización

Versión maximización

$$\max_{\mathbf{w}\in\mathbb{R}^{K\times K}} \quad \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{K} w_{ij}s_i s_j - \sum_{k=1}^{K}\theta_k s_k.$$

Versión minimización: Función de energía computacional

$$\min_{\mathbf{w}\in\mathbb{R}^{K\times K}} \quad -\frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{K} w_{ij}s_i s_j + \sum_{k=1}^{K}\theta_k s_k.$$

Función de energía computacional en las diferentes iteraciones

$$\min_{\mathbf{w}\in\mathbb{R}^{K\times K}} \quad E(i) = -\frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{K} w_{kj}s_k s_j + \sum_{k=1}^{K}\theta_k s_k,$$

para las iteraciones $0 \leq i \leq I$.

Índice de contenidos

Teorema de convergencia

Teorema

Si en la iteración $i+1$ actualizamos el estado de la unidad de proceso k según la regla de actualización anterior, manteniendo iguales los estados de las unidades de procesos restantes, entonces la función de energía decrece o permanece igual, es decir, $E(i+1) \leq E(i)$.

Demostración

Por simplicidad en la demostración usaremos la función de optimización:

$$E(i) \quad = -\frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{K} w_{kj} s_k(i) s_j(i),$$

y supondremos que queremos actualizar la neurona k-ésima, $s_k(i+1)$.

Teorema de convergencia

Demostración

Sólo habrá cambio de energia si se produce una modificación en la neurona, por lo que si $s_k(i+1) = s_k(i)$, entonces, $E(i+1) = E(i)$.
Si hay cambio de energía entonces $s_k(i+1) = -s_k(i)$, por lo que:

$$E(i+1) - E(i) = -\frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{K} w_{kj} s_k(i+1) s_j(i) + \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{K} w_{kj} s_k(i) s_j(i)$$

Teniendo en cuenta que sólo influye el térmo k-ésimo, el resto de términos se anulan porque no cambian, con lo que

$$\begin{aligned} E(i+1) - E(i) &= -\frac{1}{2}\sum_{j=1}^{K} w_{kj} s_k(i+1) s_j(i) + \frac{1}{2}\sum_{j=1}^{K} w_{kj} s_k(i) s_j(i) \\ &= \frac{1}{2}\sum_{j=1}^{K} w_{kj} s_k(i) s_j(i) + \frac{1}{2}\sum_{j=1}^{K} w_{kj} s_k(i) s_j(i), \end{aligned}$$

ya que $s_k(i+1) = -s_k(i)$.

Teorema de convergencia

Demostración

Recordemos que

$$\begin{aligned} E(i+1)-E(i) &= \frac{1}{2}\sum_{j=1}^{K} w_{kj}s_k(i)s_j(i)+\frac{1}{2}\sum_{j=1}^{K} w_{kj}s_k(i)s_j(i) \\ &= s_k(i)\sum_{j=1}^{K} w_{kj}s_j(i) \\ &= -s_k(i+1)\sum_{j=1}^{K} w_{kj}s_j(i) \end{aligned}$$

y tendiendo en cuenta que el signo de $s_k(i+1)$ y $\sum_{j=1}^{K} w_{kj}s_j(i)$ deben coincidir, entonces tenemos que $E(i+1)-E(i)<0$.

Índice de contenidos

Corolario 1: Estabilidad y Mínimo Local

Corolario 1

Si la matriz de pesos es simétrica, la red recurrente bipolar alcanza un estado estable en un número finito de pasos utilizando la regla de actualización secuencial, y dicho estado corresponde a un mínimo local de la función de energía.

Demostración

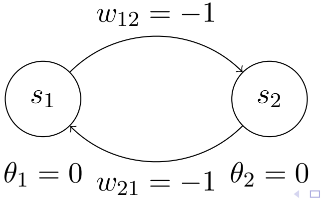
En efecto, como la red reduce la función de energía en cada iteración (o al menos no cambia) y el número posible de estados que puede alcanzar la red es finito (2^K), se tiene que estabilizar en un número finito de pasos. Además, una vez que está estabilizada, no se puede reducir la función de energía cambiando el estado de una sola unidad de proceso. Es decir, la solución encontrada es un mínimo local.

Corolario 1: Estabilidad y Mínimo Local

Caso pesos simétricos.

Supongamos un modelo como el que aparece debajo (pesos sinápticos simétricos, dos nodos y $\theta_1 = \theta_2 = 0$). La función de energía en la iteración i, será $E(i) = s_1(i)s_2(i)$.

- Si la red parte de la configuración (1,1) y actualizamos la primera unidad de proceso, como el potencial sináptico es -1 entonces se desactiva
- Alcanza la configuración (-1,1). La red se estabiliza en dicha configuración.
- El otro mínimo local corresponde a la configuración (1,-1)

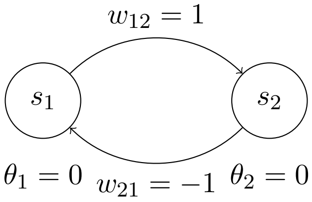


Corolario 1: Estabilidad y Mínimo Local

Caso pesos no simétricos.

Supongamos un modelo como el que aparece debajo (pesos sinápticos no simétricos, dos nodos y $\theta_1 = \theta_2 = 0$).

- La red estaría cambiando siempre los estados de sus unidades de proceso de forma cíclica y no converge a un estado estable. Intentarlo partiendo de la configuración (1, 1) (**Intentad en clase**).



Índice de contenidos

Introducción al Modelo de Hopfield Continuo

- El modelo de Hopfield es un tipo de red neuronal desarrollada por John Hopfield en 1982.
- Existen dos variantes principales: el modelo discreto y el modelo continuo.

Modelo Discreto

- Emplea neuronas binarias (0 o 1).
- Actualización sincrónica o secuencial.
- Almacena y recupera patrones de manera robusta.

Modelo Continuo

- Utiliza neuronas con valores reales en un rango continuo.
- Actualización mediante ecuaciones diferenciales.
- Apropiado para problemas de optimización continua.

Índice de contenidos

Modelo funcional red Hopfield continua

La salida de la neurona k-ésima en el instante $i+1$ quedaría definida como:

$$s_k(i+1) = s_k(i) + \Delta s_k(i),$$

donde $\Delta s_k(i)$ representa a la variación que se produce en la neurona k-ésima y dicha variación es proporcional a la siguiente ecuación (tangente hiperbólica):

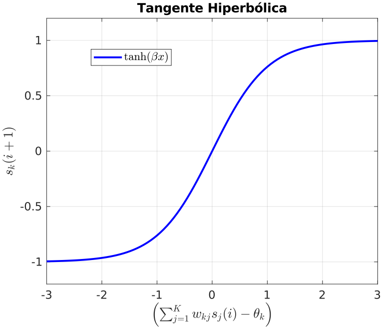
$$\begin{aligned}\Delta s_k(i) &\propto f\left(\sum_{j=1}^{K} w_{kj} s_j(i) - \theta_k\right) \\ &= \frac{\exp\left(\beta\left(\sum_{j=1}^{K} w_{kj} s_j(i) - \theta_k\right)\right) - \exp\left(-\beta\left(\sum_{j=1}^{K} w_{kj} s_j(i) - \theta_k\right)\right)}{\exp\left(\beta\left(\sum_{j=1}^{K} w_{kj} s_j(i) - \theta_k\right)\right) + \exp\left(-\beta\left(\sum_{j=1}^{K} w_{kj} s_j(i) - \theta_k\right)\right)}\end{aligned}$$

,

con $s_k(i+1) \in [-1,1], k \in \{1,\ldots,K\}, i \in \{1,\ldots,I\}$.

Modelo funcional red Hopfield continua

Representación gráfica de la tangente hiperbólica: $s_k(i) \in [-1, 1]$; si $\left(\sum_{j=1}^{K} w_{kj} s_j(i) - \theta_k\right) = 0$, entonces $\Delta s_k(i) = 0$.



Índice de contenidos

Dinámica de computación

Al igual que en el caso concreto:

$$s_k(i+1) = s_k(i) + \Delta s_k(i)$$

En este caso la variación quedaría definida como:

$$\Delta s_k(i) = \begin{cases} 0 & \text{si } s_k(i) = 1 \text{ y } f\left(\sum_{j=1}^{K} w_{kj} s_j(i) - \theta_k\right) > 0 \\ 0 & \text{si } s_k(i) = -1 \text{ y } f\left(\sum_{j=1}^{K} w_{kj} s_j(i) - \theta_k\right) < 0 \\ \eta f\left(\sum_{j=1}^{K} w_{kj} s_j(i) - \theta_k\right) & \text{si } s_k(i) \in (-1, 1) \end{cases}$$

donde $\eta > 0$ es la tasa de aprendizaje. En los dos primeros casos la variación es cero al encontrarse en los límites de la función.

Función de energía

Misma función que en el caso discreto.

¡Gracias por vuestra atención!