

Capítulo 7

ADAM, BAM y SDM

Este capítulo tiene como propósito incluir tres modelos de memoria asociativa que no son esenciales para entender el desarrollo de los modelos de memorias asociativas, pero que comparten una característica común: son extensiones de modelos conocidos.

El ADAM (Advanced Distributed Associative Memory) de Austin se puede considerar como una extensión de las redes asociativas de Willshaw; la BAM (Bidirectional Associative Memory) de Kosko, que surgió en pleno auge de los intentos realizados por algunos grupos de investigación con miras a minimizar una de las desventajas de la memoria Hopfield, se puede considerar como una extensión ingeniosa de ésta última; y finalmente, la SDM (Sparse Distributed Memory) de Kanerva está basada en la estructura de una RAM usual, y puede considerarse como una generalización de los conceptos involucrados en una RAM

7.1. Advanced Distributed Associative Memory

El modelo de memoria asociativa ADAM (*Advanced Distributed Associative Memory*) fue propuesto por Jim Austin de la *University of York*, como tema de tesis doctoral, en 1986. Austin creó ADAM con miras a mejorar la capacidad de las *redes asociativas* de Willshaw *et. al.*; para ello, diseñó un arreglo de dos *redes asociativas* a las que agregó un sistema de *códigos de clase*, como se describirá a continuación.

El desarrollo de esta parte está basado en la libre adaptación de la información contenida en las referencias (Austin, 1987). (Kennedy, Austin,

& Cass. 1995), (Austin, Buckle, Kennedy, Moulds, Pack. & Turner, 1997) (Turner. & Austin. 1997).

Para describir el modelo, consideremos $A = \{0, 1\}$ y el conjunto fundamental

$$\{(x^\mu, y^\mu) \mid x^\mu \in A^n \text{ y } y^\mu \in A^m, \text{ donde } \mu = 1, 2, \dots, p\}$$

ADAM es un sistema de entrada y salida que al operar acepta como entrada un patrón binario $x^\mu \in A^n$, y produce como salida un patrón $y^\mu \in A^m$.

En ambas fases, aprendizaje y recuperación, para cada pareja de patrones de entrada y salida se requiere un código de clase, el cual es un patrón $z^\mu \in A^k$, donde k es un número entero positivo; cada código de clase contiene τ componentes con valor 1 y $k - \tau$ componentes con valor 0, donde τ es un número entero positivo fijo con la condición $\tau < k$ (esto significa que habrá $\frac{k!}{\tau!(k-\tau)!}$ diferentes códigos de clase disponibles).

Antes de iniciar las fases de aprendizaje y recuperación, es preciso llevar a cabo dos pasos previos:

1. Se escoge el valor $\tau < k$ para crear los códigos de clase
2. Se crean dos matrices nulas (llenas de valores 0): $\mathbf{P} = [p_{ij}]_{k \times n}$ y $\mathbf{Q} = [q_{ij}]_{m \times k}$

En la fase de aprendizaje de ADAM, para cada $\mu = 1, 2, \dots, p$ se realiza lo siguiente:

1. Se propone un código de clase z^μ (el cual contiene τ componentes con valor 1)
2. Se actualiza la matriz \mathbf{P} de acuerdo con el siguiente esquema:

	x_1^μ	x_2^μ	\dots	x_j^μ	\dots	x_n^μ
z_1^μ	p_{11}	p_{12}	\dots	p_{1j}	\dots	p_{1n}
z_2^μ	p_{21}	p_{22}	\dots	p_{2j}	\dots	p_{2n}
\vdots	\vdots	\vdots		\vdots		\vdots
z_i^μ	p_{i1}	p_{i2}	\dots	p_{ij}	\dots	p_{in}
\vdots	\vdots	\vdots		\vdots		\vdots
z_k^μ	p_{k1}	p_{k2}	\dots	p_{kj}	\dots	p_{kn}

donde la regla para actualizar los componentes p_{ij} es:

$$p_{ij} = \begin{cases} 1 & \text{si } z_i^\mu = 1 = x_j^\mu \\ \text{valor anterior} & \text{en otro caso} \end{cases}$$

3. Se actualiza la matriz Q de acuerdo con el siguiente esquema:

	z_1^μ	z_2^μ	\dots	z_j^μ	\dots	z_k^μ
y_1^μ	q_{11}	q_{12}	\dots	q_{1j}	\dots	q_{1k}
y_2^μ	q_{21}	q_{22}	\dots	q_{2j}	\dots	q_{2k}
\vdots	\vdots	\vdots		\vdots		\vdots
y_i^μ	q_{i1}	q_{i2}	\dots	q_{ij}	\dots	q_{ik}
\vdots	\vdots	\vdots		\vdots		\vdots
y_m^μ	q_{m1}	q_{m2}	\dots	q_{mj}	\dots	q_{mk}

donde la regla para actualizar los componentes q_{ij} es:

$$q_{ij} = \begin{cases} 1 & \text{si } y_i^\mu = 1 = z_j^\mu \\ \text{valor anterior} & \text{en otro caso} \end{cases}$$

Al final de la fase de aprendizaje se tendrán las dos redes asociativas de Willshaw P y Q .

En la fase de recuperación se considera un patrón de entrada $\tilde{x}^\omega \in A^n$, donde $\omega \in \{1, 2, \dots, p\}$ (si el ruido es cero, se cumple que $\tilde{x}^\omega = x^\omega$ pertenece al conjunto fundamental de entrada) y se llevan a cabo las siguientes acciones:

1. Se realiza la operación $\mathbf{P} \cdot \tilde{\mathbf{x}}^\omega$; es decir,

$$(\mathbf{P} \cdot \tilde{\mathbf{x}}^\omega)_i = \sum_{j=1}^n p_{ij} \tilde{x}_j^\omega, \forall i \in \{1, 2, \dots, k\}$$

2. Se calcula el correspondiente código de clase \mathbf{z}^ω a través de la umbralización:

$$z_i^\omega = \begin{cases} 1 & \text{si } \sum_{j=1}^n p_{ij} \tilde{x}_j^\omega = \bigvee_{h=1}^k \left[\sum_{j=1}^n p_{hj} \tilde{x}_j^\omega \right] \\ 0 & \text{en otro caso} \end{cases}$$

3. Ya conocido el código de clase \mathbf{z}^ω , se realiza la operación $\mathbf{Q} \cdot \mathbf{z}^\omega$, decir,

$$(\mathbf{Q} \cdot \mathbf{z}^\omega)_i = \sum_{j=1}^k q_{ij} z_j, \forall i \in \{1, 2, \dots, m\}$$

Se espera que el vector $\mathbf{Q} \cdot \mathbf{z}^\omega$ sea precisamente el patrón fundamental de salida \mathbf{y}^ω .

7.2. Bidirectional Associative Memory

Pasada la euforia que despertó la aparición de la memoria Hopfield los albores de la década de los ochenta, y como parte de las productivas consecuencias que vivió el campo de las memorias asociativas, los investigadores hurgaron en la posibilidad de buscar maneras de subsanar la clara desventaja de la autoasociatividad; es decir, se inició la búsqueda de modelos heteroasociativos, que dieron lugar a la creación de memorias que aceptasen conjuntos fundamentales como el siguiente:

$$\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mathbf{x}^\mu \in A^n \text{ y } \mathbf{y}^\mu \in A^m, \text{ donde } \mu = 1, 2, \dots, p\}$$

Bart Kosko, investigador de la *University of Southern California*, propuso una respuesta: la hoy famosa BAM (*Bidirectional Associative Memory*)

Similarmente a como Austin ensambló dos redes asociativas de Willshaw para diseñar su ADAM, Kosko ideó un ingenioso arreglo de dos memorias

Hopfield, y demostró que este diseño, al que llamó BAM, es capaz de asociar patrones de manera heteroasociativa.

A pesar de que en el bagaje editorial y en internet pululan ensayos, escritos, monografías y opiniones sobre la BAM de Kosko, el contenido de esta sección se basa en sólo tres fuentes: el artículo original (Kosko, 1988), un libro de texto (Kosko, 1992) y una excelente obra colectiva sobre el tema de las memorias asociativas (Hassoun, 1993).

Al igual que Hopfield descubrió que su memoria asociativa funcionaba mejor con patrones bipolares que con patrones binarios, Kosko mostró que la BAM se comporta mejor con patrones bipolares que como lo hace con patrones binarios, y por ello diseñó la BAM para que aceptase el conjunto fundamental

$$\{(x^\mu, y^\mu) \mid x^\mu \in A^n \text{ y } y^\mu \in A^m, \text{ donde } \mu = 1, 2, \dots, p\}, \text{ donde } A = \{-1, 1\}$$

Es decir, la BAM es un sistema de entrada y salida que al operar acepta como entrada un patrón bipolar $x^\mu \in A^n$, y produce como salida un patrón $y^\mu \in A^m$.

La fase de aprendizaje para la BAM genera dos matrices diferentes, M y W, pero en cada caso es similar a la fase de aprendizaje de la memoria Hopfield y del *Linear Associator*, recordando que en la memoria Hopfield hay una diferencia relacionada con la diagonal principal (en ceros); la BAM no presenta esta restricción.

1. Para cada una de las p asociaciones (x^μ, y^μ) se encuentra la matriz $y^\mu \cdot (x^\mu)^t$ de dimensiones $m \times n$:

$$y^\mu \cdot (x^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_n^\mu \end{pmatrix} \cdot (x_1^\mu, x_2^\mu, \dots, x_n^\mu) = \begin{pmatrix} y_1^\mu x_1^\mu & y_1^\mu x_2^\mu & \dots & y_1^\mu x_i^\mu & \dots & y_1^\mu x_n^\mu \\ y_2^\mu x_1^\mu & y_2^\mu x_2^\mu & \dots & y_2^\mu x_i^\mu & \dots & y_2^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu x_1^\mu & y_i^\mu x_2^\mu & \dots & y_i^\mu x_i^\mu & \dots & y_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_n^\mu x_1^\mu & y_n^\mu x_2^\mu & \dots & y_n^\mu x_i^\mu & \dots & y_n^\mu x_n^\mu \end{pmatrix}$$

2. Se suman la p matrices $y^\mu \cdot (x^\mu)^t$ para obtener la matriz M :

$$M = \sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t = [m_{ij}]_{m \times n}$$

3. La matriz W de $n \times m$ se define como la transpuesta de M ; es decir:

$$\begin{aligned} W &= (M)^t \\ &= \left[\sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t \right]^t \\ &= \sum_{\mu=1}^p [y^\mu \cdot (x^\mu)^t]^t \\ &= \sum_{\mu=1}^p (x^\mu) \cdot (y^\mu)^t \\ &= [w_{ij}]_{n \times m} \end{aligned}$$

El paso 3 es la aportación de Kosko: la ingeniosa idea de que la matriz W sea la transpuesta de M .

La fase de recuperación de la BAM, para cada uno de los dos tipos de patrones x^μ, y^μ es idéntica a la fase de recuperación de la memoria Hopfield. En la BAM se requiere verificar convergencia en ambos casos, para x^μ y para y^μ .

Representemos el estado de la BAM en el tiempo t por $[x(t), y(t)]$; entonces $[x_i(t), y_i(t)]$ representa los valores de x_i y y_i en el tiempo t , y $[x_i(t+1), y_i(t+1)]$ los valores en el tiempo siguiente ($t+1$).

Dado un par de vectores columna de entrada (\tilde{x}, \tilde{y}) , la fase de recuperación consta de cuatro pasos:

1. Para $t = 0$, se hace $x(t) = \tilde{x}$ y $y(t) = \tilde{y}$; es decir, $x_i(0) = \tilde{x}_i, \forall i \in \{1, 2, 3, \dots, n\}$ y $y_i(0) = \tilde{y}_i, \forall i \in \{1, 2, 3, \dots, m\}$

2. $\forall i \in \{1, 2, 3, \dots, m\}$ se calcula $y_i(t+1)$ de acuerdo con la condición siguiente:

$$y_i(t+1) = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij}x_j(t) > 0 \\ y_i(t) & \text{si } \sum_{j=1}^n m_{ij}x_j(t) = 0 \\ -1 & \text{si } \sum_{j=1}^n m_{ij}x_j(t) < 0 \end{cases}$$

3. $\forall i \in \{1, 2, 3, \dots, n\}$ se calcula $x_i(t+1)$ de acuerdo con la condición siguiente:

$$x_i(t+1) = \begin{cases} 1 & \text{si } \sum_{j=1}^m w_{ij}y_j(t) > 0 \\ x_i(t) & \text{si } \sum_{j=1}^m w_{ij}y_j(t) = 0 \\ -1 & \text{si } \sum_{j=1}^m w_{ij}y_j(t) < 0 \end{cases}$$

4. Se realizan las siguientes comparaciones: $x_i(t+1)$ con $x_i(t) \forall i \in \{1, 2, 3, \dots, n\}$ y $y_i(t+1)$ con $y_i(t) \forall i \in \{1, 2, 3, \dots, m\}$. Si se cumplen ambas condiciones $\mathbf{x}(t+1) = \mathbf{x}(t)$ y $\mathbf{y}(t+1) = \mathbf{y}(t)$, el proceso termina y la pareja de vectores recuperados es $[\mathbf{x}(0), \mathbf{y}(0)] = (\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$. De otro modo, el proceso continúa de la siguiente manera: los pasos 2, 3 y 4 se iteran tantas veces como sea necesario hasta llegar a un valor $t = \tau$ para el cual se cumplan las dos condiciones: $x_i(\tau+1) = x_i(\tau) \forall i \in \{1, 2, 3, \dots, n\}$ y $y_i(\tau+1) = y_i(\tau) \forall i \in \{1, 2, 3, \dots, m\}$; el proceso termina y la pareja de patrones recuperados es $[\mathbf{x}(\tau), \mathbf{y}(\tau)]$.

Se espera que la pareja recuperada $[\mathbf{x}(\tau), \mathbf{y}(\tau)]$ sea una de las parejas $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ del conjunto fundamental.

7.3. Sparse Distributed Memory

En su búsqueda de un modelo para la memoria humana, el finlandés Pentti Kanerva desarrolló un tipo de memoria asociativa a la que llamó *Sparse Distributed Memory* (SDM).

La SDM puede considerarse como una variante de la RAM usual, como quedará de manifiesto en los siguientes párrafos, en donde se ha adaptado notación e información contenidas en diversas fuentes (Kanerva, 1988; Flynn, Kanerva & Bhadkamkar, 1989; Simpson, 1990; Zboril, 1997; Ludermir Carvalho, Braga, & Souto, 1999).

Para describir la SDM, consideremos $A = \{0, 1\}$ y el conjunto fundamental

$$\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mathbf{x}^\mu \in A^n \text{ y } \mathbf{y}^\mu \in A^m, \text{ donde } \mu = 1, 2, \dots, p\}$$

Esta memoria requiere de dos pasos previos a las fases de aprendizaje y recuperación:

1. Se escoge $k \in \mathbb{N}$, tal que $n \ll k < 2^n$
2. Se crea la matriz aleatoria $\mathbf{Q}_{k \times n}$ cuyas entradas q_{ij} son elementos de A escogidos aleatoriamente

La fase de aprendizaje consiste de dos etapas:

1. En esta etapa primera, para cada una de las p asociaciones $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ llevan a cabo los siguientes cuatro pasos:
 - a) Se obtiene el vector de Hamming, el cual es un vector columna \mathbf{h}^μ de dimensión k que resulta de $\mathbf{Q}_{k \times n} \otimes \mathbf{x}^\mu = \mathbf{h}^\mu$; la operación \otimes está definida de modo que el valor de la i -ésima componente

$$h_i^\mu = H(\mathbf{Q}_i, \mathbf{x}^\mu) = \sum_{j=1}^n q_{ij} \oplus x_j^\mu$$

donde $H(\mathbf{Q}_i, \mathbf{x}^\mu)$ es la distancia Hamming entre el renglón \mathbf{Q}_i de la matriz $\mathbf{Q}_{k \times n}$ y el vector columna de entrada \mathbf{x}^μ . La distancia Hamming se calcula sumando todos los ítems para los cuales la operación OR exclusiva \oplus tiene valor lógico 1. Los valores almacenados en las entradas del vector \mathbf{h}^μ son números enteros entre 1 y n ; es decir, $1 \leq h_i^\mu \leq n, \forall i \forall \mu$

- b) Se escoge el radio de Kanerva ρ , el cual es un número entero tal que $1 \leq \rho \leq \frac{n}{2}$

- c) A partir del vector de Hamming \mathbf{h}^μ y el radio de Kanerva ρ , se obtiene el vector columna de transición interna \mathbf{b}^μ , de dimensión k y con entradas en el conjunto A , cuya i -ésima componente se calcula por la expresión

$$b_i^\mu = \begin{cases} 1 & \text{si } h_i^\mu \leq n - 2\rho \\ 0 & \text{de otro modo} \end{cases}$$

- d) Se operan los vectores \mathbf{y}^μ y $(\mathbf{b}^\mu)^t$ para obtener una matriz \mathbf{M}^μ de dimensiones $m \times k$, de modo que ocurra lo siguiente: el valor de la ij -ésima entrada m_{ij}^μ debe ser 1 si se cumple que $b_j^\mu = 1$ y $y_i^\mu = 1$, y debe ser -1 si se cumple que $b_j^\mu = 1$ y $y_i^\mu = 0$; además, m_{ij}^μ debe ser 0 en otro caso. La expresión con que se logran obtener estos valores para la ij -ésima entrada es:

$$m_{ij}^\mu = (2y_i^\mu b_j^\mu - 1) b_j^\mu$$

2. En esta segunda etapa se suman la p matrices \mathbf{M}^μ para obtener la memoria

$$\mathbf{M} = \sum_{\mu=1}^p \mathbf{M}^\mu = [m_{ij}]_{k \times m}$$

en donde la ij -ésima componente m_{ij} se expresa así:

$$m_{ij} = \sum_{\mu=1}^p (2y_i^\mu b_j^\mu - 1) b_j^\mu$$

En la **fase de recuperación**, que también consta de dos etapas, se considera un patrón de entrada \mathbf{x}^ω , donde $\omega \in \{1, 2, \dots, p\}$ (si el ruido es cero, se cumple que $\tilde{\mathbf{x}}^\omega = \mathbf{x}^\omega$ pertenece al conjunto fundamental de entrada).

1. Esta primera etapa es en parte similar a la etapa 1 de la fase de aprendizaje y consiste de dos pasos, dado el vector columna de entrada \mathbf{x}^ω :

- a) Se realiza la operación $\mathbf{Q}_{k \times n} \otimes \tilde{\mathbf{x}}^\omega = \mathbf{h}^\omega$ para obtener el vector de Hamming \mathbf{h}^ω , de manera que:

$$h_i^\omega = H(\mathbf{Q}_i \cdot \tilde{\mathbf{x}}^\omega) = \sum_{j=1}^n q_{ij} \oplus \tilde{x}_j^\omega$$

donde $H(Q_i, \tilde{x}^\omega)$ es la distancia Hamming entre el renglón Q_i de la matriz $Q_{k \times n}$ y el vector \tilde{x}^ω . La distancia Hamming se calcula sumando todos los items para los cuales la operación OR exclusiva \oplus tiene valor lógico 1. Los valores almacenados en las entradas del vector h^ω son números enteros entre 1 y n ; es decir, $1 \leq h_i^\omega \leq n$, $\forall i$

- b) A partir del vector de Hamming h^ω y del radio de Kanerva ρ escogido en la fase de aprendizaje, se obtiene el primer vector columna de transición interna b^ω , de dimensión k y con entradas en el conjunto A , cuya i -ésima componente se calcula por la expresión $b_i^\omega = \phi(h^\omega, \rho)$, donde la función umbral ϕ de dos parámetros u y v está definida por:

$$\phi(u, v) = \begin{cases} 1 & \text{si } u \geq n - 2v \\ 0 & \text{de otro modo} \end{cases}$$

2. Esta segunda etapa consta de tres pasos, a saber:

- a) Se calculan los m umbrales que se representan en el vector umbral θ^ω de dimensión m , cuya i -ésima componente se expresa de la siguiente manera:

$$\theta_i^\omega = \frac{1}{2} \sum_{j=1}^k m_{ij}$$

- b) Se obtiene el segundo vector de transición interna t^ω de dimensión m cuya i -ésima componente se calcula así:

$$t_i^\omega = \sum_{j=1}^k m_{ij} b_j^\omega$$

- c) Finalmente se obtiene el vector columna \tilde{y}^ω de dimensión m , que es la salida correspondiente al vector de entrada \tilde{x}^ω . Para calcular el valor de la i -ésima componente \tilde{y}_i^ω se hace uso del vector umbral θ^ω y del segundo vector de transición t^ω , por medio de la siguiente expresión:

$$\tilde{y}_i^\omega = \begin{cases} 1 & \text{si } t_i^\omega \geq \theta_i^\omega \\ 0 & \text{de otro modo} \end{cases}$$

Se espera que el vector \tilde{y}^w sea precisamente el patrón fundamental de salida y^w .

7.4. Consideraciones finales del capítulo.

Actualmente, Austin es Profesor de la Universidad de York y Director del Advanced Computer Architecture Group: ha trabajado con un grupo de colegas y alumnos en un superconjunto de ADAM, el cual ha sido bautizado con el acrónimo de AURA (Advanced Uncertain Reasoning Architecture).

Los avances del grupo de Austin se pueden consultar en

www.cs.york.ac.uk/arch/nn/aura.html

Respecto de la BAM, en el año de 1998, Hongchi Shi propuso un modelo general de memorias asociativas bidireccionales en su artículo A General Model for Bidirectional Associative Memories, *TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, AUGUST 1998.

Actualmente Kanerva es el principal investigador del Stochastic Pattern Computing Group en el Swedish Institute of Computer Science, y su trabajo se enfoca al tema representación distribuida:

www.sics.se

