

Невронни Мрежи

проф. дн В. Младенов

3. Асоциативно обучение и мрежи на Hebb

НМ за асоциация на образи

- Невронните мрежи за асоциация служат за асоциации на образи, които могат да бъдат сходни, противоположни, достатъчно близки или между тях може да има определени връзки.
- Информацията за образите се съхранява в невронната мрежа.
- При подаване на входен вектор (recall), съответният изходен вектор трябва да е образът, който е асоцииран с подадения входен вектор.
- Асоциациите биват два типа – хетероасоциации и автоасоциации.
- При хетероасоциациите се асоциират различни обекти и тогава размерностите на входните и изходните вектори са различни.
- При автоасоциациите се асоциират различни части от едни и същи образи и при тях размерностите на входните и изходните вектори са еднакви.

НМ за асоциация на образи

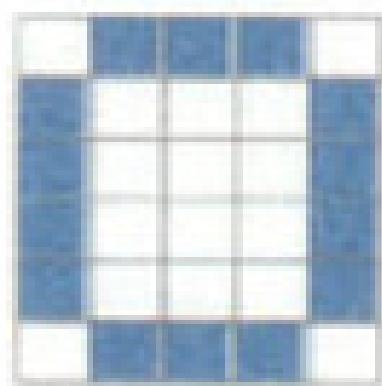
- Асоцираните образи са от типа
 - по подобия
 - по противоположности
 - по съседство (spatial)
 - по близки последователности (temporal)
- Асоциативно подаване на входен вектор (recall)
 - извикване на асоциирани образи (evoke)
 - recall на даден образ само по дадени части от него
 - evoke/recall на непълни/зашумени образи
- Асоциациите са два вида. За два образа s и t
 - хетеро-асоциация – връзка между два отделни образа
 - автоасоциация – ‘свързване’ на част от даден образ с други негови части (или парчета вместо части)

НМ за асоциация на образи

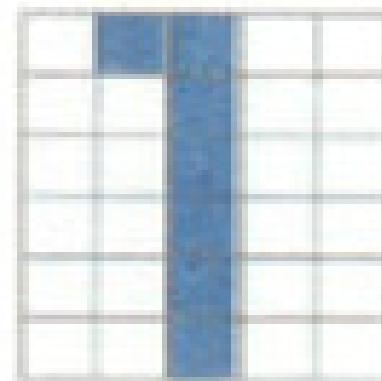
- Разгледаните до сега НМ се обучават ‘с учител’ (supervised learning process)
- За разлика от тях при НМ за асоциация настройката на теглата може да се извършва с или без ‘учител’ (supervised или unsupervised).
- Архитектурите на невронните мрежи за асоциации на образи могат да бъдат еднослойни или многослойни (при двупосочните асоциации).
- Алгоритмите за обучение на такива мрежи се базират на правилото на Hebb или градиентни методи и техни разновидности.
- Анализът на такива мрежи включва анализ на сходимостта на алгоритмите и анализ на капацитетите на мрежите (колко образи могат да бъдат запомнени и съхранени).
- Невронните мрежи за асоциации на образи могат да бъдат използвани като асоциативни памети. При автоассоциативните памети желаните изходни вектори в някои случаи могат да съвпадат с входните, а в други случаи да не съвпадат с тях.

Асоциативни памети

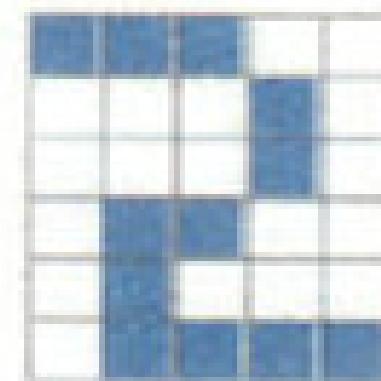
Прави се опит да се запомнят входните последователности
Желания изход се задава с входните вектори. **Входно-**
изходните вектори (образи) са $\{\mathbf{s}(1), \mathbf{t}(1)\}, \{\mathbf{s}(2), \mathbf{t}(2)\}, \dots, \{\mathbf{s}(P), \mathbf{t}(P)\}$



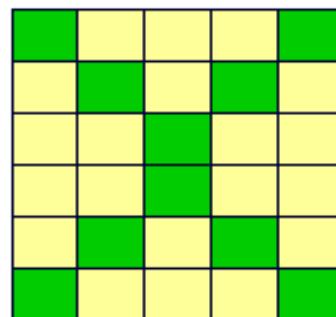
$\{\mathbf{s}(1), \mathbf{t}(1)\}$



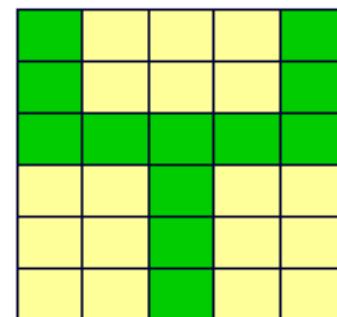
$\{\mathbf{s}(2), \mathbf{t}(2)\}$



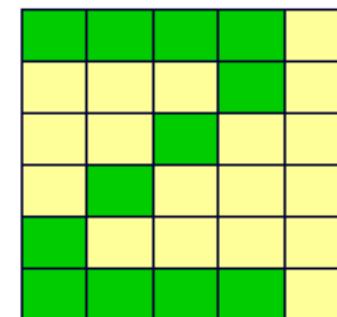
$\{\mathbf{s}(3), \mathbf{t}(3)\}$



$\{\mathbf{s}(1), \mathbf{t}(1)\}$



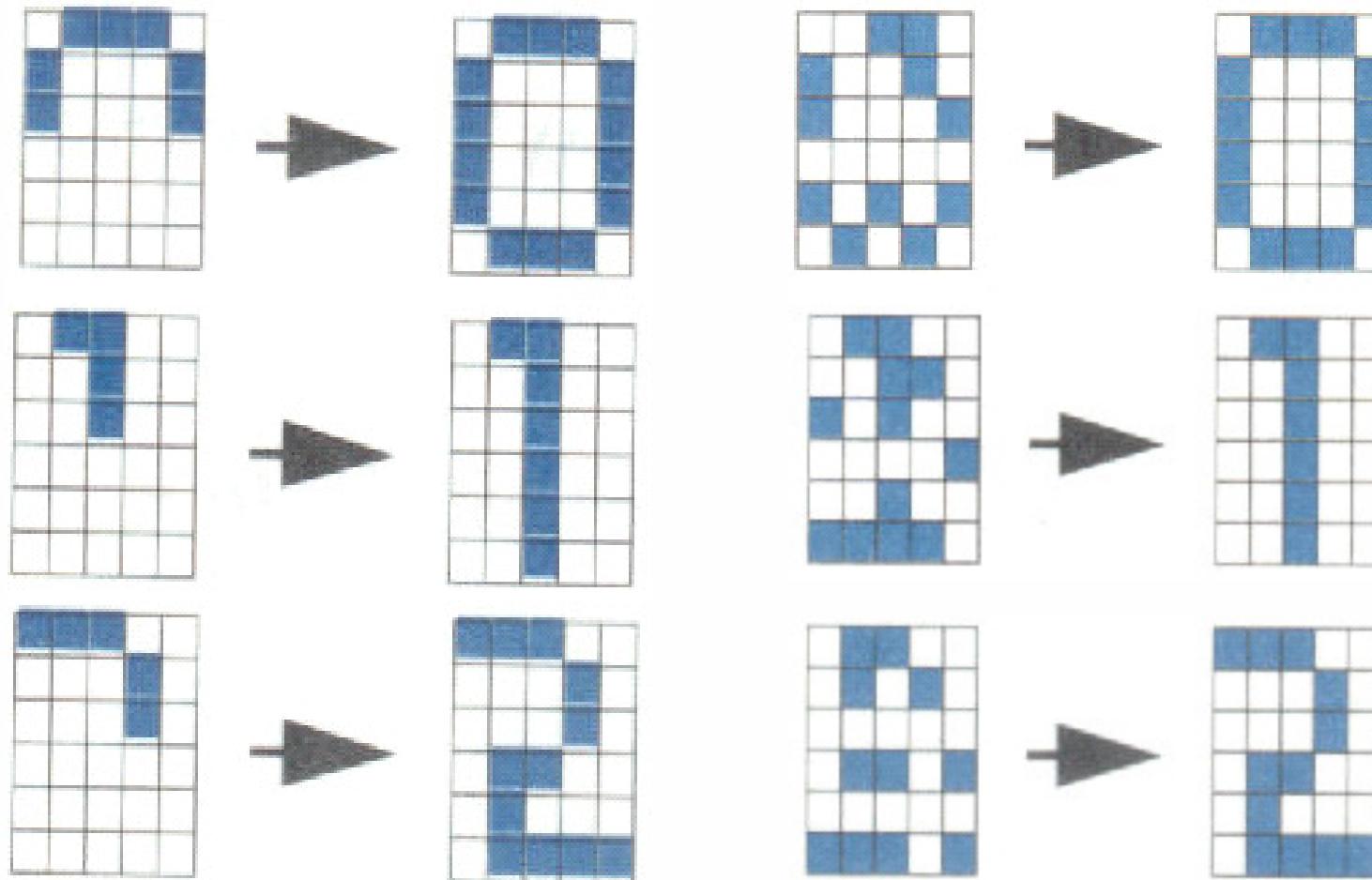
$\{\mathbf{s}(2), \mathbf{t}(2)\}$



$\{\mathbf{s}(3), \mathbf{t}(3)\}$

Асоциативни памети

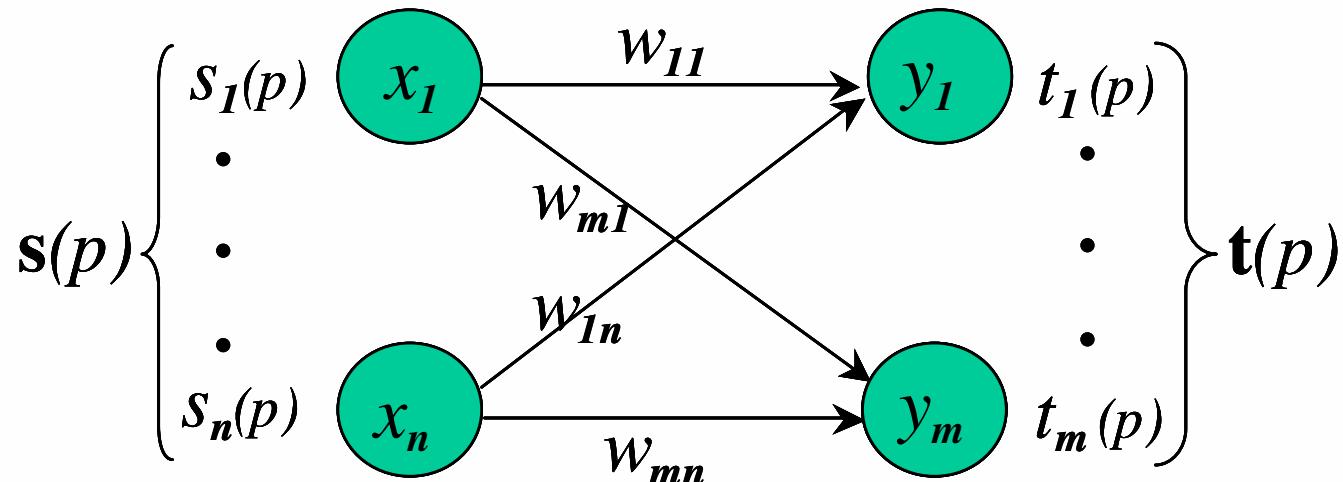
Пример за автоасоциации, когато изходните образи не съвпадат изцяло с входните.



Асоциативни памети

- При примера за автоасоциации, когато изходните образи не съвпадат изцяло с входните такива случаи се дължат на липсващи данни или наличие на шум във входните образи.
- Нека невронната мрежа за асоциация на образи е обучена с последователност от P входно-изходни образи
 $\{\mathbf{s}(1), \mathbf{t}(1)\}, \{\mathbf{s}(2), \mathbf{t}(2)\}, \dots, \{\mathbf{s}(P), \mathbf{t}(P)\}$
- Ако на входа на мрежата бъде подаден входен вектор
 $\mathbf{x} = \mathbf{s}(p), p=1,2,\dots,P$, съответният изходен вектор трябва да бъде
 $\mathbf{y} = \mathbf{t}(p), p=1,2,\dots,P$.

Пример за еднослойна невронна мрежа за асоциации на образи и структура →



Асоциативни памети

- Ако поради липсващи данни или шум във входните данни подаденият входен вектор е близък до $\mathbf{s}(p)$, то изходният вектор трябва да бъде близък до $\mathbf{t}(p)$.
- Целта при тези мрежи е „да накараме мрежата“ да запомни Р на брой прототипни входно-изходни двойки (образа) и при подавана на входни данни отговарящи (или близки, подобни) на даден прототип да се получават изходни данни отговарящи на съответния подходящ изходен клас или:

Целта на обучението е да бъдат настроени теглата w_{ij} така, че ако $\mathbf{x} = \mathbf{s}(p)$, то $\mathbf{y} = \mathbf{t}(p)$, $p=1,2,\dots,P$. Изходният вектор \mathbf{y} се изчислява с формулата $\mathbf{y}=\mathbf{f}(\mathbf{W} \bullet \mathbf{x})$, или покомпонентно $y_i=f(\mathbf{W}_i \bullet \mathbf{x})$, $i=1,2,\dots,m$, където \mathbf{x} е входният вектор, \mathbf{W} е матрицата с теглата на мрежата, \mathbf{W}_i е i -тият вектор на тази матрица, $f(\bullet)$ е активационната функция, а $\mathbf{f}(\bullet)$ е m -мерна векторна функция, компонентите на която са активационните функции на изходните наврони. Активационната функция обикновено е от релеен тип с гранични стойности 0 и 1 или -1 и 1, в зависимост от това дали се работи с бинарни {0, 1} или биполярни {-1, 1} данни.

Правило на Hebb

Правилото на Hebb гласи:

Ако активирането на неврон k (генериране на изходен сигнал) е свързано с активирането на неврон j , то силата на връзката между тях се увеличава, а ако активирането на неврон k не е свързано с активирането на неврон j , то силата на връзката между тези неврони отслабва.

С други думи - корелираните активации усилват връзките, а некорелираните ги отслабват.

Правило на Hebb

- Последователният подход за прилагане на алгоритъма свързан с правилото на Hebb, при бинарни или биполярни данни, се базира на изчисление на донастройка на теглата

$$\Delta w_{ij}(p) \quad i=1,2,\dots,m, \quad j=1,2,\dots,n$$

за всяка двойка входно-изходени вектори

$$\{\mathbf{s}(p), \mathbf{t}(p)\}, \quad p=1,2,\dots,P, \quad \Delta w_{ij}(p) = t_i(p)s_j(p), \quad i=1,2,\dots,m, \\ j=1,2,\dots,n.$$

- По този начин $\Delta w_{ij}(p)$ е положително число, ако $t_i(p)$ и $s_j(p)$ едновременно са 1 при бинарни данни или са с еднакви знаци при биполярни данни.
- Ако началната стойност теглата е нула, то на база на всички входно-изходни вектори се получава $\mathbf{W}=\{w_{ij}\}$

$$w_{ij} = \sum_{p=1}^P s_j(p)t_i(p), \quad i=1,2,\dots,m, \quad j=1,2,\dots,n$$

Правило на Hebb

Формулите могат да бъдат записани по-компактно във векторна форма, като матрицата за донастройка на теглата при използване на p -ти входно-изходен вектор е

$$\Delta \mathbf{W}(p) = \mathbf{t}(p) \cdot \mathbf{s}^T(p) = \begin{bmatrix} t_1 \\ t_m \end{bmatrix} \begin{bmatrix} s_1, \dots, s_n \end{bmatrix} = \begin{bmatrix} t_1 s_1, \dots, t_1 s_n \\ t_2 s_1, \dots, t_2 s_n \\ \vdots \\ t_m s_1, \dots, t_m s_n \end{bmatrix} = \begin{bmatrix} \Delta w_{11}, \dots, \Delta w_{1n} \\ \vdots \\ \Delta w_{m1}, \dots, \Delta w_{mn} \end{bmatrix}$$

Матрицата \mathbf{W} с теглата на мрежата може да бъде изчислена директно след използването на всички P входно-изходни вектори $\mathbf{W} = \sum_{p=1}^P \mathbf{t}(p) \cdot \mathbf{s}^T(p)$

Претеглената сума на входа на активационната функция на i -ти неврон, $i=1,2,\dots,m$ е

$$y_{in_i} = \sum_{j=1}^n w_{ij} x_j$$

Правило на Hebb

Алгоритъмът включва 3 вложени цикъка p, i, j

$p = 1$ до P

/* за всяка двойка данни от обучаващата извадка */

$i = 1$ to m

/* за всеки ред във \mathbf{W} */

$j = 1$ to n

/* за всеки елемент j в ред i */

При работа с бинарни данни изходният сигнал на неврона се определя на база на активационната функция:

$$y_i = f(y_{in_i}) = \begin{cases} 1, & \text{ако } y_{in_i} \geq 0 \\ 0, & \text{ако } y_{in_i} < 0 \end{cases}$$

Правило на Hebb

Нека на входа на мрежата е подаден k -тия вектор (recall на k -тия образ). Тогава векторът от сигналите на входовете на активационните функции на невроните е:

$$\begin{aligned} \mathbf{W}\mathbf{s}(k) &= \left[\sum_{p=1}^P \mathbf{t}(p) \mathbf{s}^T(p) \right] \mathbf{s}(k) = \sum_{p=1}^P \mathbf{t}(p) \cdot \mathbf{s}^T(p) \cdot \mathbf{s}(k) = \\ &= \mathbf{t}(k) \cdot \mathbf{s}^T(k) \cdot \mathbf{s}(k) + \sum_{\substack{p=1 \\ p \neq k}}^P \mathbf{t}(p) \cdot (\mathbf{s}^T(p) \cdot \mathbf{s}(k)) = \\ &= \|\mathbf{s}(k)\|^2 \mathbf{t}(k) + \sum_{\substack{p=1 \\ p \neq k}}^P \mathbf{t}(p) \cdot (\mathbf{s}^T(p) \cdot \mathbf{s}(k)) , \end{aligned}$$

компонентът $\sum_{\substack{p=1 \\ p \neq k}}^P (\mathbf{s}^T(p) \cdot \mathbf{s}(k)) \cdot \mathbf{t}(p)$ се нарича смесване (cross-talk)

Правило на Hebb

- Векторът с изходните сигнали на невроните е

$$\mathbf{y}(k) = \mathbf{f}(\mathbf{W}\mathbf{s}(k))$$

- Ако векторът на смесването е нулев, то $\mathbf{y}(k)=\mathbf{t}(k)$ тъй като $\|\mathbf{s}(k)\|^2$ е положителна константа.

В този случай е реализирана желаната асоциация на входния вектор $\mathbf{s}(k)$ с изходния вектор $\mathbf{t}(k)$.

- Ако векторът на смесването не е нулев, то във всеки от входните сигнали за активационните функции на невроните има информация не само от подадения k -ти образ, а и от останалите образи ($p=1,2,\dots,P$, $p \neq k$). Поради това в общия случай векторът с изходните сигнали на невроните няма да съвпада с желания изходен вектор $\mathbf{t}(k)$

$$\mathbf{y}(k) = \mathbf{f}(\mathbf{W}\mathbf{s}(k)) \neq \mathbf{t}(k)$$

Правило на Hebb

От $y(k) = \mathbf{f}(\mathbf{W}\mathbf{s}(k)) \neq \mathbf{t}(k)$ следва, че при наличие на смесване в общия случай не е гарантирана асоциация на входните със съответните изходни вектори.

Ако входните вектори $\mathbf{s}(p)$, $p=1,2,\dots,P$ са ортогонални т.е.

$$\mathbf{s}^T(p) \cdot \mathbf{s}(k) = 0, \quad \forall p, k$$

то тогава векторът на смесването ще бъде нулев и тогава всеки входен вектор ще бъде асоцииран коректно с желания изходен вектор.

При прилагане на правилото на Hebb ако броят на образите P , които се асоциират е голям, то матрицата с теглата на мрежата \mathbf{W} става подобна на единичната матрица и тогава се загубват коригиращите и свойства (изхода е точно равен на входа). Поради това много често се прибягва до замяна на диагоналните елементи на \mathbf{W} с 0.

Примери за хетероасоциативност

- При бинарни двойки образци $s:t$ с $n = 4$ и $m = 2$
- Входният сигнал на активационната функция на i -ти неврон, $i=1,2,\dots,m$ е

$$y_{in_i} = \sum_j w_{ij} x_j$$

- Изходният сигнал на неврона се определя на база на праговата активационната функция

$$y_i = f(y_{in_i}) = \begin{cases} 1, & \text{ако } y_{in_i} \geq 0 \\ 0, & \text{ако } y_{in_i} < 0 \end{cases}$$

- Матрицата W с теглата на мрежата може да бъде изчислена директно след използването на всички P входно-изходни вектори $W = \sum_{p=1}^P t(p) \cdot s^T(p)$.

Примери за хетероасоциативност

	$s(p)$	$t(p)$
p=1	[1 0 0 0] ^T	[1 0] ^T
p=2	[1 1 0 0] ^T	[1 0] ^T
p=3	[0 0 0 1] ^T	[0 1] ^T
p=4	[0 0 1 1] ^T	[0 1] ^T

Обучаваща извадка:

$$W(P) = t(1) \cdot s^T(1) + t(2) \cdot s^T(2) + t(3) \cdot s^T(3) + t(4) \cdot s^T(4) = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

Извикване:

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$s=[0 1 0 0]^T$
подобие с $s(1)$

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$s=[0 1 1 0]^T$
няма достатъчно
подобие до всеки
от класовете

Примери за автоасоциативност

Нека е дадена невронна мрежа за автоасоциация на входния вектор (образ) $s=[1 \ 1 \ 1 \ -1]^T$.

В случая $P=1$ и $t=s=[1 \ 1 \ 1 \ -1]^T$.

Разглежданата невронна мрежа е с $n=4$ входни сигнали и $m=4$ изходни сигнали (неврони). Като се използва правилото на Hebb за матрицата с теглата на мрежата се получава

$$W = t \cdot s^T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \cdot [1 \ 1 \ 1 \ -1] = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

В случая се използват **биполярни данни** и затова активационната функция на неврона е

$$y_i = f(y_{in_i}) = \begin{cases} 1, & \text{ако } y_{in_i} \geq 0 \\ -1, & \text{ако } y_{in_i} < 0 \end{cases}$$

Примери за автоасоциативност

При подаване на вектора, който трябва да се асоциира, изходният вектор е равен точно на входния, т.е. реализирана е автоасоциация:

$$W \cdot [1 \ 1 \ 1 \ -1]^T = [4 \ 4 \ 4 \ -4]^T \rightarrow [1 \ 1 \ 1 \ -1]^T.$$

Ако поради наличие на шум входният вектор за мрежата е $[-1 \ 1 \ 1 \ -1]^T$, то изходният образ е равен също на образа, който трябва да се асоциира:

$$W \cdot [-1 \ 1 \ 1 \ -1]^T = [2 \ 2 \ 2 \ -2]^T \rightarrow [1 \ 1 \ 1 \ -1]^T.$$

Примери за автоасоциативност

При подаване на входен вектор с липсващи данни (кодирани с 0)

$$[0 \ 0 \ 1 \ -1]^T,$$

който е сходен с желания вектор, то и в този случай изходния вектор е точно равен на асоциирания вектор:
 $W.[0 \ 0 \ 1 \ -1]^T = [2 \ 2 \ 2 \ -2]^T \rightarrow [1 \ 1 \ 1 \ -1]^T.$

Ако е подаден входният вектор $[-1 \ -1 \ 1 \ -1]^T$,
който се различава с два от елементите си от желания вектор, то той не може да бъде асоцииран правилно:

$$W.[-1 \ -1 \ 1 \ -1]^T = [0 \ 0 \ 0 \ 0]^T$$

Примери за автоасоциативност

От начина на прилагане на правилото на Hebb лесно се забелязва, че ако броят на образите P , които се асоциират е голям, то матрицата с теглата на мрежата W става подобна на единичната матрица и тогава се загубват коригиращите и свойства (изхода е точно равен на входа). Поради това много често се прибягва до замяна на диагоналните елементи на W с нули.

Тогава новата матрица с теглата от предния пример е

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

Примери за автоасоциативност

В този случай при подаване на вектора $[1 \ 1 \ 1 \ -1]^T$, който трябва да се асоциира, изходният образ е равен точно на входния, т.е. асоциацията е реализирана и при този избор на матрицата с теглата.

При подаване на зашумения образ $[-1 \ 1 \ 1 \ -1]^T$ и на образа с липсващите данни $[0 \ 0 \ 1 \ -1]^T$ се реализира асоциация, докато при подаване на образа с преобладаващ шум в данните $[-1 \ -1 \ 1 \ -1]^T$, се получава грешна класификация:

$$W \cdot [-1 \ -1 \ 1 \ -1]^T = [1 \ 2 \ -1 \ 1]^T \rightarrow [1 \ 1 \ -1 \ 1]^T .$$

Примери за автоасоциативност

$$\mathbf{W} = \mathbf{W}_0 = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

$$\mathbf{W} \cdot [1 \ 1 \ 1 \ -1]^T = [3 \ 3 \ 3 \ -3]^T \rightarrow [1 \ 1 \ 1 \ -1]^T$$

$$\mathbf{W} \cdot [-1 \ 1 \ 1 \ -1]^T = [3 \ 1 \ 1 \ -1]^T \rightarrow [1 \ 1 \ 1 \ -1]^T$$

$$\mathbf{W} \cdot [0 \ 0 \ 1 \ -1]^T = [2 \ 2 \ 1 \ -1]^T \rightarrow [1 \ 1 \ 1 \ -1]^T$$

$$\mathbf{W} \cdot [-1 \ -1 \ 1 \ -1]^T = [1 \ 2 \ -1 \ 1]^T \rightarrow [1 \ 1 \ -1 \ 1]^T$$

Примери за автоасоциативност

Нека в дадена невронна мрежа трябва да бъдат съхранени (асоциирани) два вектора, които не са ортогонални
 $[1 \ -1 \ -1 \ 1]^T$ и $[1 \ 1 \ -1 \ 1]^T$

След прилагане на правилото на Hebb и нулиране на диагоналните елементи матрицата с теглата на мрежата е

$$\mathbf{W} = \begin{bmatrix} 0 & 0 & -2 & 2 \\ 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & -2 \\ 2 & 0 & -2 & 0 \end{bmatrix}$$

За случая лесно може да се провери, че асоциацията за първия вектор $[1 \ -1 \ -1 \ 1]^T$ е грешна:

$$\begin{bmatrix} 0 & 0 & -2 & 2 \\ 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & -2 \\ 2 & 0 & -2 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ -4 \\ 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

Примери за автоасоциативност

Нека в дадена невронна мрежа трябва да бъдат съхранени (асоциирани) три вектора, които са ортогонални

$$[1 \ 1 \ -1 \ -1]^T, [-1 \ 1 \ 1 \ -1]^T, [-1 \ 1 \ -1 \ 1]^T.$$

След прилагане на правилото на Hebb и нулиране на диагоналните елементи за матрицата с теглата на мрежата се получава

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

Лесно може да бъде проверено, че асоциацията за трите вектора е правилна. Ако бъде добавен четвърти вектор, който е ортогонален на предишните три, матрицата с теглата на мрежата е нулева, т.е. свойството *памет* на мрежата се нарушава.

Примери за автоасоциативност

Теорема: Една невронна мрежа за автоасоциация на образи, използваща правилото на *Hebb* и нулиране на диагоналните елементи при формиране на матрицата с теглата, може коректно да асоциира $n-1$ ортогонални биполярни вектори, където n е размерността на входните вектори.

В много случаи, след подаване на **някой** входен вектор, изходният образ на автоасоциативната мрежа **не съответства на никой от векторите**, които мрежата трябва да асоциира. В тези случаи полученият изходен вектор (образ) може отново да бъде подаден на входа на мрежата и по този начин се организира **итеративна процедура до получаване на желан изходен вектор**.

Така се реализират **итеративни автоасоциативни мрежи** и в случая състоянието на отделните неврони на дадена итерация, а оттам и на мрежата се определя от изходните сигнали на невроните.

Примери за автоасоциативност

Ако $\mathbf{s}=[1 \ 1 \ 1 \ -1]^T$ е образ, който трябва да се съхранява в мрежата. След прилагане на правилото на Hebb и нулиране на диагоналните елементи матрицата с теглата на мрежата е:

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

Нека активационната функция за всеки от четирите неврона на мрежата е

$$y_i = f(y_{in_i}) = \begin{cases} 1, & \text{ако } y_{in_i} > 0 \\ 0, & \text{ако } y_{in_i} = 0 \\ -1, & \text{ако } y_{in_i} < 0 \end{cases}$$

Примери за автоасоциативност

Ако началният вектор, който е подаден на входа на мрежата е $\mathbf{x}(0) = [1 \ 0 \ 0 \ 0]^T$, съответният изходен вектор $\mathbf{x}(1)$

$$\mathbf{W} \cdot \mathbf{x}(0) = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \mathbf{x}(1)$$

След подаване на $\mathbf{x}(1)$ на входа на мрежата се получава:

$$\mathbf{W} \cdot \mathbf{x}(1) = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 2 \\ -2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \mathbf{x}(2)$$

Примери за автоасоциативност

При следващо подаване на $\mathbf{x}(2)$ на входа на мрежата се получава:

$$\mathbf{W} \cdot \mathbf{x}(2) = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 3 \\ -3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \mathbf{x}(3)$$

Следователно за три итерации изходният образ на мрежата спира да се променя $\mathbf{x}(3)=\mathbf{x}(2)$ и е точно равен на образа, който трябва да се съхранява (асоциира). В случая номера на поредната итерация при промяна на изходният вектор на мрежата се означава в скоба.

Примери за автоасоциативност

Ако е даден началният вектор $\mathbf{x}(0)$, на всяка итерация съответният изходен вектор се изчислява с помощта на итеративната формула

$$\mathbf{x}(i+1) = \mathbf{f}(\mathbf{W} \cdot \mathbf{x}(i)) , \quad i = 0, 1, 2, \dots .$$

Итеративната процедура продължава докато за някое k

$$\mathbf{x}(k+1) = \mathbf{x}(k) .$$

Анализът на сходимостта на горната итеративна процедура е свързан с устойчивостта и областите на привличане на стационарните точки на нелинейна дискретна система, която се описва със система от уравнения

$$\mathbf{x}(i+1) = \mathbf{F}(\mathbf{x}(i)) = \mathbf{f}(\mathbf{W} \cdot \mathbf{x}(i)) , \quad i = 0, 1, 2, \dots ,$$

където $\mathbf{F}(\cdot)$ е вектор-функция, която може да се изчисли на база на вектор функцията $\mathbf{f}(\cdot)$, формирана от активационните функции на невроните от мрежата.