بسم الله الرحمن الرحيم

فهرست مطالب

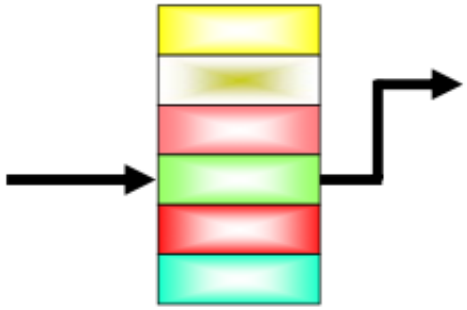
- انواع شبکه عصبی بازگشتی
- شبکه های عصبی دو سویه
- شبکه های عصبی رقابتی
  - شبکه بیشینه
  - شبکه همینگ
  - شبکه کلاه مکزیکی

انواع شبکه عصبی بازگشتی

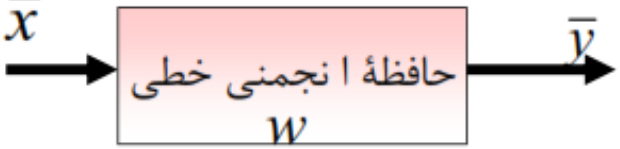
- شبکه های انجمنی
- شبکه هاپفیلد
- ماشین بولتزمان
- شبکه های لایه ای

حافظهٔ انجمنی (تداعی کننده)

Associative Memory



در یک حافظه معمولی کامپیوتر، آدرس داده شده و محتوی خوانده می‌شود.



در یک حافظه انجمنی:

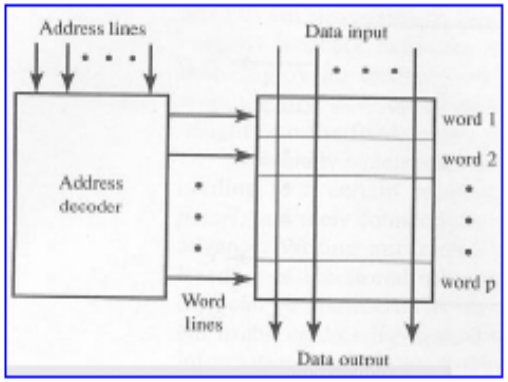
$\bar{x}$: تحریک (stimulus)

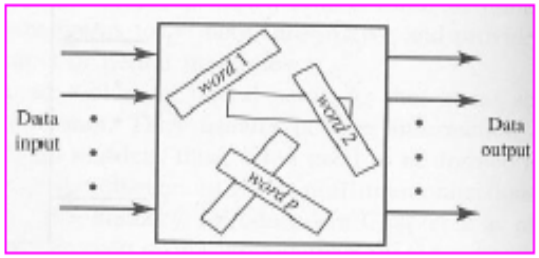
$\bar{y}$: پاسخ (response)

w : ماتریس حافظه در حافظهٔ انجمنی خطی

رابطهٔ ورودی خروجی در یک حافظهٔ انجمنی خطی: $\bar{y} = \bar{x}w$

- حافظهٔ انسان مواردی مانند افکار، احساسات و ... را که شبیه، متضاد، مجاور یا با توالی نزدیکی اتفاق می‌افتند، به هم پیوند می‌دهد
- مثال
  - ارتباط بین عکس یک فرد و خود او
  - ارتباط بین یک نت موسیقی و آهنگ آن





- ذخیره‌سازی و بازیابی داده‌ها براساس محتوا و نه بر اساس آدرس ذخیره‌سازی

شبکه‌های حافظهٔ انجمنی (Associative Memory)

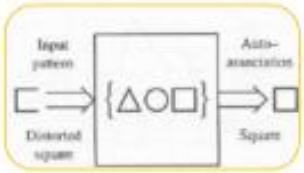
- مدل ساده شده‌ای از حافظهٔ انسان
- شبکه‌هایی برای ذخیره‌سازی الگوها و پیوند بین آنها

ویژگیهای حافظهٔ انجمنی

- حافظه‌ای گسترده (distributed) است. یعنی با معیوب شدن یک یا چند نورون، حافظه هنوز کار می‌کند.
- اطلاعات به وسیلهٔ آرایش فضایی فعالیتهای نرونی (تنظیم وزن‌های اتصالات) روی تعداد زیادی نرون، در حافظه ذخیره می‌گردد.
- این نوع حافظه نسبت به نوفه (نویز) یا اعوجاج در الگوی ورودی به میزان زیادی مقاوم است (تحمل می‌کند).

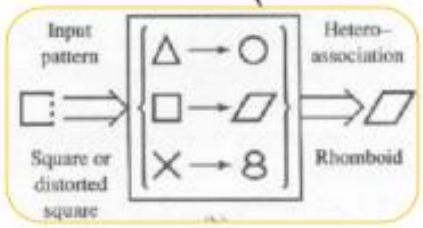
شبکه‌های حافظهٔ انجمنی (Associative Memory)

- یادگیری پیوند یک جفت بردار ورودی–خروجی s:t



- شبکه حافظه خود انجمنی (Auto-Associative Memory)
  - بردارهای ورودی و خروجی یکسان هستند.

- شبکه حافظهٔ دیگرانجمنی (Hetero-Associative Memory)
  - بردارهای ورودی و خروجی با هم متفاوت هستند.

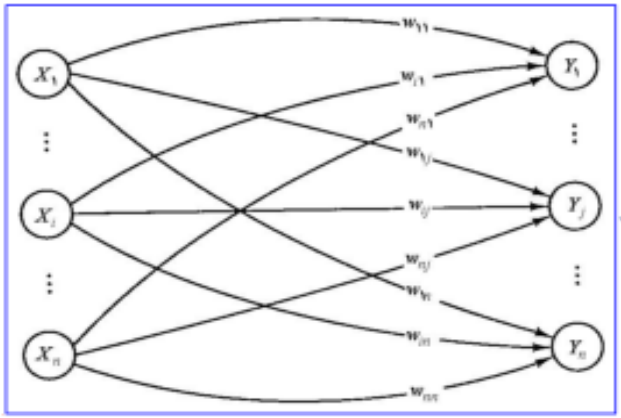


- در هر دو نوع شبکه
  - جفت الگوهای آموزش یافته را یاد می‌گیرد
  - هنگامی‌که یک الگوی ورودی به آن داده می‌شود که مشابه، اما نه یکسان با، ورودی‌های آموزش است، الگوی پاسخ مورد نظر را بازخوانی می‌کند.

آموزش (تنظیم وزن‌ها)

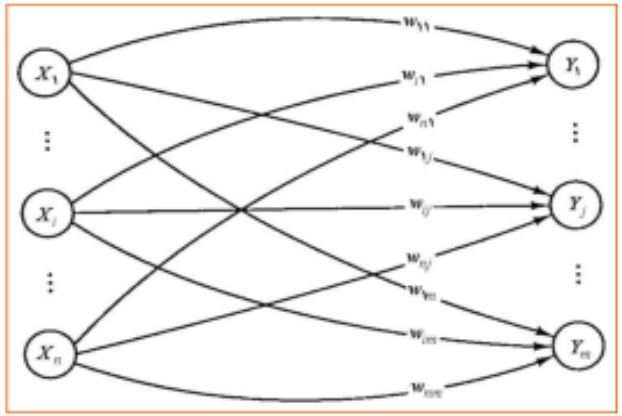
- شبکه یک لایه
  - قانون هب
  - قانون دلتا

شبکهٔ عصبی خودانجمنی



تعداد n ورودی و n خروجی

شبکهٔ عصبی دیگرانجمنی



تعداد n ورودی و m خروجی

شبکه حافظه خود انجمنی (Auto-Associative Memory)

- بردارهای ورودی آموزش و خروجی هدف یکسان هستند
- آموزش در این شبکه‌ها = ذخیره‌سازی بردارها

- بازیابی بردار ورودی (حتی در حالت خراب شدن یا نویزی شدن) در صورت شبیه بودن به یکی از بردارهای ذخیره شده

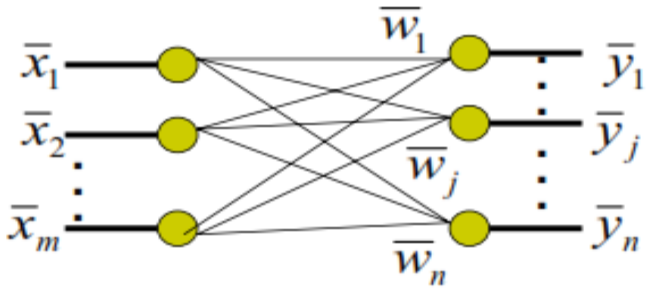
شبکه حافظهٔ دیگرانجمنی (Hetero-Associative Memory)

- الگوهای ورودی و خروجی با هم متفاوتند
- آموزش (تنظیم وزن‌ها)
  - قانون هب
  - قانون دلتا

شبکه خودانجمن خطی بازگشتی (Recurrent Linear Autoassociator)

- هر نرون به تمام نرون‌های دیگر (و خودش) متصل است
- تابع فعال‌سازی هر واحد تابع همانی است = شبکه خودانجمن خطی
- آموزش با قانون هب (ماتریس متقارن وزن‌ها)
- در نسخه اولیه، وزن‌های قطر اصلی صفر نیست
  - در نسخه جدید، وزن‌های قطر اصلی صفر شده اما تاثیر زیادی در کارایی شبکه نداشته است

- مدل حافظهٔ انجمنی خطی: در این حافظه عملاً هر ورودی، خروجی مربوطه را تداعی (انجمن) می‌کند.



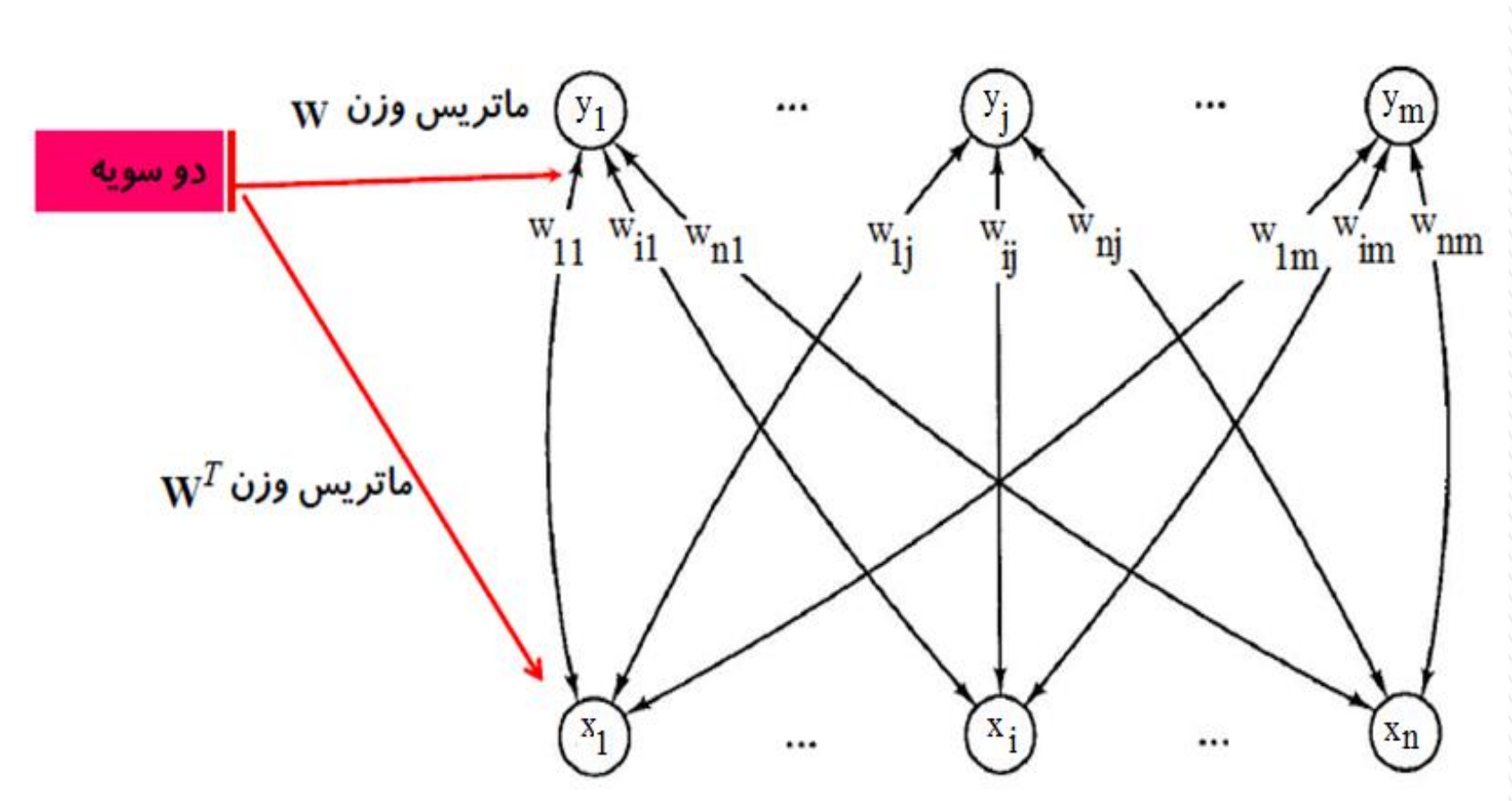
شبکه خودانجمن بازگشتی

- حالت گسترش یافته‌ای از شبکهٔ خودانجمنی
- پاسخ شبکه به یک الگوی ورودی خاص دوباره به عنوان ورودی به شبکه داده می‌شود.
  - در مواردی، شبکه به سیگنال ورودی‌ای با الگوی هدف ذخیره شده فوراً پاسخ صحیح نمی‌دهد، هر چند پاسخ داده شده به اندازهٔ کافی به یک الگوی ذخیره شده شبیه باشد.
  - دادن پاسخ اول به عنوان ورودی مجدد به شبکه، شانس صحیح پاسخ دادن شبکه را افزایش می‌دهد.
  - اثربخشی در مقابله با داده‌های نویزی

Bidirectional Associative Memory (BAM)

شبکه های عصبی دو سویه

حافظهٔ انجمنی دوسویه (BAM: Bidirectional Associative Memory)



مشخصات شبکه عصبی BAM

- ذخیره مجموعه‌ای از پیوند الگوها با جمع زدن ماتریس‌های همبستگی دوقطبی
- ساختار شبکه متشکل از دو لایه نرون است اما مسیرهای اتصال وزن‌دار و جهت‌دار به صورت دوسویه به هم وصل می‌شوند (تفاوت با دیگر شبکه‌های انجمنی)
- فرستادن و بازگرداندن سیگنال بین دو لایه تا زمان رسیدن تمام نرون‌ها به نقطه تعادل
- دریافت ورودی از هر دو لایه توسط شبکه

شبکه عصبی BAM

- الگوریتم یادگیری (تعیین وزن‌ها)
  - استفاده از قانون هب

$$w_{ij}(new) = w_{ij}(old) + s_i\, t_j \;; \quad i.e.,\ \Delta w_{ij} = s_i\, t_j$$

  - بردارهای ورودی و خروجی $\mathbf{s}(p) : \mathbf{t}(p)\,, \quad p = 1, \ldots, P$
  - بردارهای دودویی $w_{ij} = \sum_p (2s_i(p) - 1)(2t_j(p) - 1)$
  - بردارهای دوقطبی $w_{ij} = \sum_p s_i(p) t_j(p)$
  - ماتریس وزن $\mathbf{W} = \{w_{ij}\}$

توابع فعال سازی درشبکه عصبی BAM

- **توابع فعال‌سازی برای BAM گسسته**
  - **تابع پله‌ای دودویی (برای لایه Y و X)**

$$x_i = \begin{cases} 1 & if \quad x_in_i > 0 \\ x_i & if \quad x_in_i = 0 \\ 0 & if \quad x_in_i < 0 \end{cases} \qquad y_j = \begin{cases} 1 & if \quad y_in_j > 0 \\ y_j & if \quad y_in_j = 0 \\ 0 & if \quad y_in_j < 0 \end{cases}$$

  - **تابع پله‌ای دوقطبی**

$$x_i = \begin{cases} 1 & if \quad x_in_i > \theta_i \\ x_i & if \quad x_in_i = \theta_i \\ -1 & if \quad x_in_i < \theta_i \end{cases} \qquad y_j = \begin{cases} 1 & if \quad y_in_j > \theta_j \\ y_j & if \quad y_in_j = \theta_j \\ -1 & if \quad y_in_j < \theta_j \end{cases}$$

- **تابع فعال‌سازی برای BAM پیوسته**
  - **سیگموید لجستیک با برد** $[0,1]$

$$f(x) = \frac{1}{1+\exp(-y_in_j)}$$

الگوریتم آموزش درشبکه عصبی BAM

- مرحلهٔ ۰- برای ذخیره کردن مجموعه‌ای از P بردار، وزن‌های شبکه را محاسبه کنید.
  مقدار اولیهٔ همهٔ فعال‌سازی‌ها را صفر قرار دهید.
- مرحلهٔ ۱- برای هر ورودی آزمایش، مراحل ۲ تا ۶ را انجام دهید.
- مرحلهٔ ۲ الف- الگوی ورودی x را به لایهٔ X ارائه کنید.
  (فعال‌سازی‌های لایهٔ X را برابر با الگوی ورودی فعلی تعیین کنید)
  مرحلهٔ ۲ ب- الگوی ورودی y را به لایهٔ Y ارائه کنید
- (الگوهای ورودی می‌توانند بردار صفر نیز باشند)
- مرحلهٔ ۳- تا زمانی‌که فعال‌سازی‌ها همگرا نشده‌اند، مراحل ۴ تا ۶ را انجام دهید.
- مرحلهٔ ۴- فعال‌سازی‌های واحدهای لایهٔ Y را به‌روز کنید.

  ورودی‌های شبکه را محاسبه کنید: $y_in_j = \sum_i w_{ij} x_i$

  فعال‌سازی‌ها را محاسبه کنید: $y_j = f(y_in_j)$

  سیگنال را به لایهٔ X بفرستید.
- مرحلهٔ ۵- فعال‌سازی‌های واحدهای لایهٔ را به‌روز کنید.

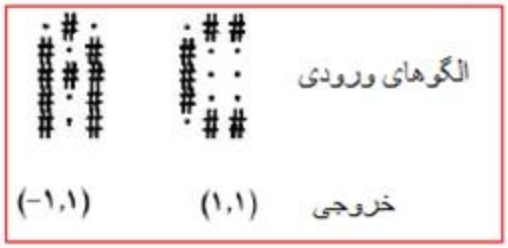
  ورودی‌های شبکه را محاسبه کنید: $x_in_i = \sum_j w_{ij} y_j$

  فعال‌سازی‌ها را محاسبه کنید: $x_i = f(x_in_i)$
- مرحلهٔ ۶- همگرایی شبکه را بررسی کنید.

  اگر بردارهای فعال‌سازی x و y به تعادل رسیده‌اند، کار را متوقف کنید.

مثال

- مثال : پیوند حروف با کدهای دوقطبی ساده . . .
  - شبکه BAM گسسته با بردارهای دوقطبی)
  - نگاشت دو حرف ساده (الگوهای ۵*۳) به کدهای دوقطبی



(-1, 1, -1, 1 -1 1 1 1 1 1 -1 1 1 -1 1)

- شبکه: یک لایه ۱۵ واحدی و یک لایه ۲ واحدی
- وزن = جمع وزن‌های دو الگو

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

ادامه مثال

- لایهٔ X به Y
  - ارائه الگوی A یا C به عنوان بردار x
  - تولید پاسخ صحیح برای بردار y

INPUT PATTERN A

$$(-1\,1-1\quad 1-1\,1\quad 1\,1\,1\quad 1-1\,1\quad 1-1\,1).\ \mathbf{W}=(-14,16)\rightarrow(-1,1)$$

INPUT PATTERN C

$$(-1\,1\,1\quad 1-1-1\quad 1-1-1\quad 1-1-1\quad -1\,1\,1).\mathbf{W}=(14,18)\rightarrow(1,1)$$

- لایهٔ Y به لایهٔ X
  - ارائه بردارهای y (الگوهای دوقطبی) به عنوان ورودی
  - ماتریس وزن (لایهٔ Y به لایهٔ X) = ترانهادهٔ ماتریس وزن لایهٔ X به لایهٔ Y

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

ادامه مثال

- لایهٔ Yبه لایهٔ X
- الگوی ورودی (-1,1)

$$(-1,1).\mathbf{W}^T = (-1,1).\begin{bmatrix} 0 & 0 & 2 & 0 & 0 & -2 & 0 & -2 & -2 & 0 & 0 & -2 & -2 & 2 & 0 \\ -2 & 2 & 0 & 2 & -2 & 0 & 2 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$= (-2 \quad 2 \quad -2 \quad 2 \quad -2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad -2 \quad 2 \quad 2 \quad -2 \quad 2)$$

$$\rightarrow (-1 \quad 1 \quad -1 \quad 1 \quad -1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad -1 \quad 1 \quad 1 \quad -1 \quad 1)$$

- خروجی = A

- الگوی ورودی (1,1)

$$(1,1).\mathbf{W}^T = (1,1).\begin{bmatrix} 0 & 0 & 2 & 0 & 0 & -2 & 0 & -2 & -2 & 0 & 0 & -2 & -2 & 2 & 0 \\ -2 & 2 & 0 & 2 & -2 & 0 & 2 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$= (-2 \quad 2 \quad 2 \quad 2 \quad -2 \quad -2 \quad 2 \quad -2 \quad -2 \quad 2 \quad -2 \quad -2 \quad -2 \quad 2 \quad 2)$$

$$\rightarrow (-1 \quad 1 \quad 1 \quad 1 \quad -1 \quad -1 \quad 1 \quad -1 \quad -1 \quad 1 \quad -1 \quad -1 \quad -1 \quad 1 \quad 1)$$

- خروجی = C

شبکه عصبی BAM

- ظرفیت ذخیره‌سازی
  - کران بالای ظرفیت $\min(m,n)$
    - n- تعداد واحدهای لایهٔ X
    - m- تعداد واحدهای لایهٔ Y
  - افزایش ظرفیت تا $\min(2^n,2^m)$ در صورت در نظر گرفتن مقدار آستانه غیرصفر (مناسب)
- پاک کردن یک پیوند ذخیره شده
  - ذخیره $s:t^c$ یا $s^c:t$ در شبکه، بردار ذخیره شده $s:t$ را پاک می‌کند

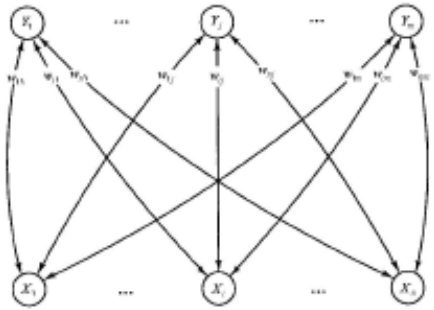
برداری که اها به ۱- و برعکس تبدیل شده است

  - ذخیره $s^c:t^c$ معادل ذخیره کردن $s:t$ است

تابع انرژی در شبکه عصبی BAM

تابع پله با آستانه صفر

- با شروع پردازش
  - ورودی x_0 به شبکه داده می‌شود ← تولید پاسخ $y_0 = sign(x_0 \cdot w)$
  - حال y_0 از سمت دیگر به شبکه داده می‌شود ← تولید ورودی برای نرون‌ها $e^t = w.y_0$
  - شبکه در حالت پایدار است اگر $x_0 = sign(e)$
  - اگر x_0 برابر با e نباشد و مقدار اندکی با آن هم زاویه داشته باشد، آنگاه مقدار $x_0 e^t$ (ضرب داخلی دو بردار) از ضرب داخلی و x_0 و هر بردار دیگر بزرگ‌تر است



- بنابراین $E = -x_0 e^T = -x_0 W y_0^T$ کمتر (علامت منفی) است
  - استفاده از آن به عنوان معیاری برای همگرایی
- تابع انرژی شبکه در تکرار iام $E(x_i, y_i) = -\frac{1}{2} x_i W y_i^T$
  - با در نظر گرفتن دو مسیر رفت (x به y) و برگشت، مقدار انرژی فوق، بدون ضریب 0.5 خواهد بود

$$E = -\frac{1}{2}(xWy^T + yW^T x^T) = -xWy^T = -\sum_{j=1}^{m}\sum_{i=1}^{n} x_i w_{ij} y_j$$

تابع انرژی در شبکه عصبی BAM

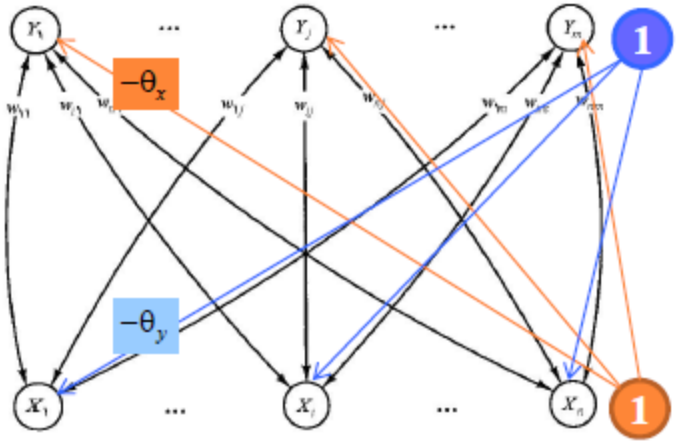
- برای تابع فعال‌سازی پله با آستانه θ
  - تغییر ماتریس وزن

$$w_{ij} = \sum_p s_i(p) t_j(p)$$

$$\mathbf{w} = \begin{bmatrix} w_{11} & \cdots & w_{1m} & -\theta_x \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & \cdots & w_{nm} & -\theta_x \\ -\theta_y & \cdots & -\theta_y & 0 \end{bmatrix}$$

آستانه نرون‌های x

آستانه نرون‌های y



- اضافه کردن نرونی با خروجی ۱ به هر دو لایه
- تابع انرژی (در تکرار iام)

$$E(\mathbf{x}_i, \mathbf{y}_i) = -\frac{1}{2}\mathbf{x}_i \mathbf{W} \mathbf{y}_i^{\mathrm{T}} + \frac{1}{2}\theta_{\mathbf{x}} \mathbf{y}_i^{\mathrm{T}} + \frac{1}{2}\mathbf{x}_i \theta_{\mathbf{y}}^{\mathrm{T}}$$