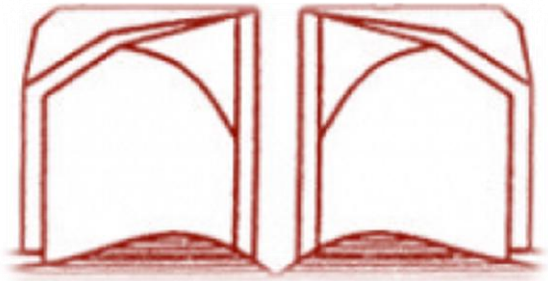


# شبکه‌های عصبی مصنوعی

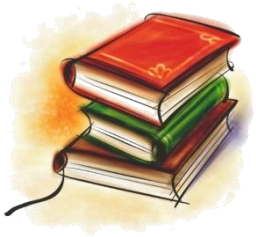
## شبکه‌های هب. پرسپترون و آدالاین


هادی ویسی

veisi@sharif.edu

دانشگاه تهران – دانشکده علوم و فنون نوین


نیم‌سال دوم ۱۳۹۲-۱۳۹۱



# فهرست

# تعریف (یادآوری)

- **بازشناسی الگو (Pattern Recognition)**
  - **پیوند الگو** (Pattern Association)
    - پیوند دادن الگوی ورودی با یک الگوی خروجی
      - ورودی: تصویر چهره یک فرد — خروجی: مشخصات و خصوصیات وی
  - **دسته‌بندی یا طبقه‌بندی الگو** (Pattern Classification)
    - حالت ساده (دو دسته): هر الگوی ورودی (یک بردار) عضو یک دسته است یا نه
    - حالت کلی (n دسته): هر الگو (بردار ورودی)، به یکی از n دسته تعلق دارد

# جداسازی خطی . . .

- مسئله دسته‌بندی ساده با شبکه عصبی
  - الگوی ورودی عضو دسته مورد نظر باشد، پاسخ «بله» و اگر ورودی عضو آن دسته نباشد، پاسخ «خیر»

$$y\_in = b + \sum_i x_i w_i$$

  - تابع فعال‌سازی پله‌ای

$$f(y\_in) = \begin{cases} 1 & if \quad y\_in \geq 0 \\ -1 & if \quad y\_in < 0 \end{cases}$$

  - مرز تصمیم‌گیری ( Decision Boundary) = مرز بین ناحیه‌ای که در آن y_in>0 و ناحیه‌ای که در آن y_in<0 است

$$b + \sum_i x_i w_i = 0$$

    - پاسخ این معادله یک خط، یک صفحه و یا یک ابرصفحه است
      - وابسته به تعداد واحدهاي ورودي (ابعاد بردار ورودی)

# جداسازی خطی . . .

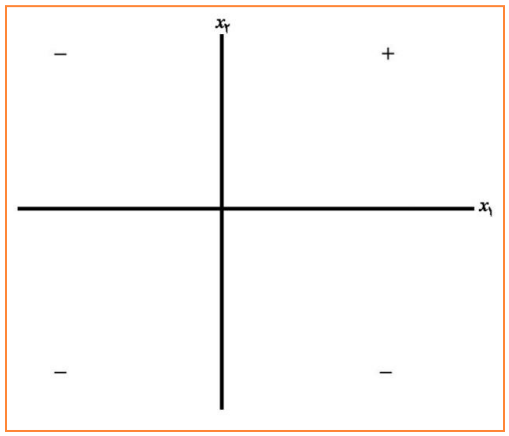
- مسئله خطی تفکیک‌پذیر (Linearly Separable)
  - حل یک مسئله توسط شبکه‌های یک لایه پس از تنظیم وزن‌ها (و بایاس)
  - تمام بردارهای ورودی آموزش که پاسخ صحیح برای آنها ۱+ (عضویت در دسته) است، در یک طرف مرز تصمیم‌گیری و تمام برادرهای ورودی آموزش که پاسخ صحیح برای آنها ۱- (عدم عضویت در دسته) است در سمت دیگر مرز تصمیم‌گیری قرار می‌گیرند

  - نشان داده شده است که شبکهٔ یک لایه فقط می‌تواند مسائلی را حل کنند که به صورت خطی تفکیک‌پذیر باشند

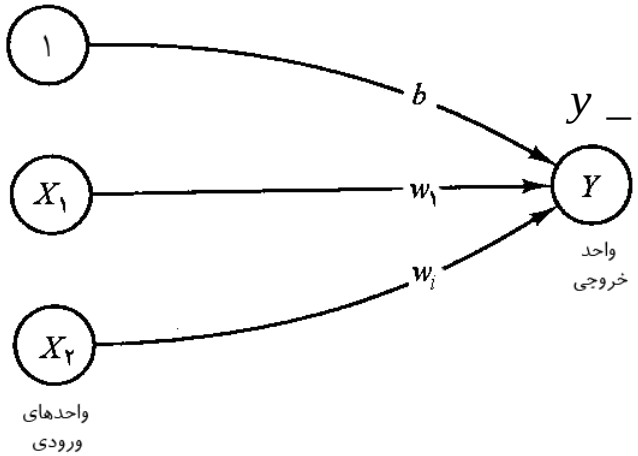
  - شبکه‌های چندلایه‌ای که از توابع فعال‌سازی خطی استفاده می‌کنند، از شبکه‌های یک لایه قوی‌تر نیستند زیرا ترکیب چند تابع خطی نیز خطی است

# جداسازی خطی . . .

| INPUT($x_1, x_2$) | OUTPUT |
|---|---|
| (1 , 1) | +1 |
| (1 , −1) | −1 |
| (−1 , 1) | −1 |
| (−1 , −1) | −1 |




## مثال: تابع AND

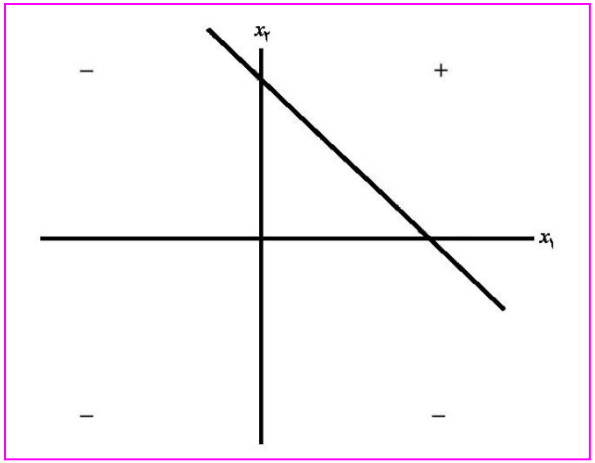



$$y\_in = 1 \times b + w_1 x_1 + w_2 x_2$$

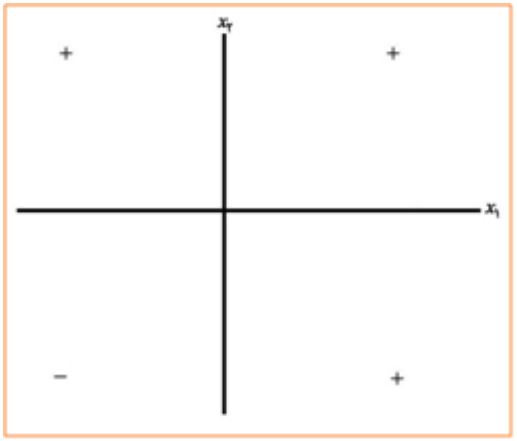
- **مرز تصمیم‌گیری** $b + w_1 x_1 + w_2 x_2 = 0$

- **پاسخ** $b = -1,\ w_1 = 1, w_2 = 1$

$$x_2 = -x_1 + 1$$

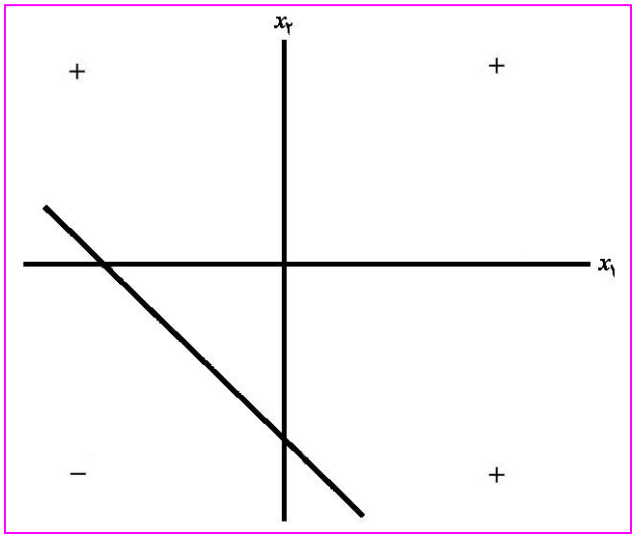


# جداسازی خطی . . .




- مثال: تابع OR

| INPUT($x_1, x_2$) | OUTPUT |
|---|---|
| (1, 1) | +1 |
| (1 , −1) | +1 |
| (−1 , 1) | +1 |
| (−1 , −1) | −1 |



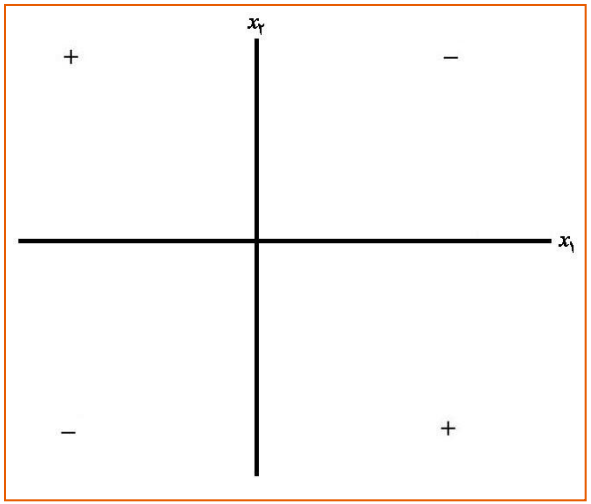

- مرز تصمیم‌گیری

$$b = 1,\ w_1 = 1, w_2 = 1$$

$$x_2 = -x_1 - 1$$

- اگر وزن بایاس وجود نداشت، مرز تصمیم‌گیری باید از مبدأ عبور می‌کرد

# جداسازی خطی




- مثال: تابع XOR

| INPUT($x_1, x_2$) | OUTPUT |
|---|---|
| (1, 1) | −1 |
| (1 , −1) | +1 |
| (−1 , 1) | +1 |
| (−1 , −1) | −1 |

- حل؟
- فضای داده‌های ورودی به صورت خطی جدایی‌پذیر نیست.
- هیچ خط مستقیم نمی‌تواند نقاط مثبت و منفی را جدا کند

# شبکهٔ هب ...

- اولین (و ساده‌ترین) قانون یادگیری برای شبکهٔ عصبی
- ایده اصلی یادگیری هب
  - یادگیری با تغییر استحکامات سیناپس‌های نرون‌ها (وزن‌های شبکه‌های عصبی) است
  - اگر دو نرون متصل به هم به طور هم‌زمان «فعال» باشند، وزن بین آنها باید افزایش یابد
  - هب دربارهٔ نرون‌هایی که به طور هم‌زمان برانگیخته نمی‌شوند، چیزی نمی‌گوید
    - یادگیری قوی‌تر = اگر دو نرون به طور هم‌زمان «غیرفعال» باشند، وزن‌ها افزایش یابد

- شبکهٔ هب یک لایه است
- به‌روز شدن (Update) وزن‌ها

$$w_i(new) = w_i(old) + x_i y$$

خروجی

وزن جدید

وزن قبلی

ورودی

- برای داده دودویی، اگر ورودی یا خروجی (یا هر دو) «غیرفعال» باشند، یادگیری صورت نمی‌گیرد

# شبکهٔ هب: الگوریتم . . .

- مرحله ۰ – به تمام وزن‌ها مقدار اولیه صفر بدهید $w_i = 0 \quad (i = 1, \ldots, n)$
- مرحله ۱ – برای هر بردار آموزش ورودی و خروجی هدف، $s:t$، مراحل ۲ تا ۴ را انجام بده
- مرحله ۲ – فعال‌سازی‌های واحدهای ورودی را تعیین کن $x_i = s_i \quad (i = 1, \ldots, n)$
- مرحله ۳ – برای واحد خروجی فعال‌سازی را تعیین کن $y = t$
- مرحله ۴ – وزن‌ها و بایاس را به‌روز کن

$$w_i(new) = w_i(old) + x_i y \quad (i = 1, \ldots, n)$$

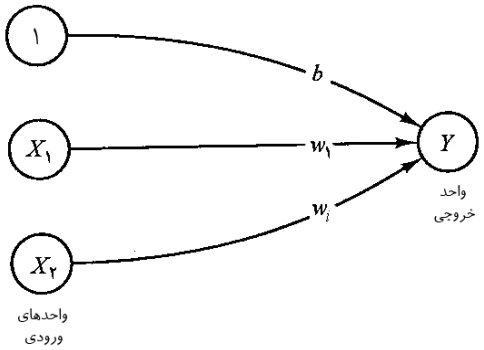
$$b(new) = b(old) + y$$

$$\mathbf{w}(new) = \mathbf{w}(old) + \mathbf{x}.y \quad \overset{\Delta w = x.y}{\Rightarrow} \quad \mathbf{w}(new) = \mathbf{w}(old) + \Delta \mathbf{w}$$

داده‌های آموزشی فقط یک بار به شبکه نشان داده شده و آموزش به اتمام می‌رسد

# شبکهٔ هب: کاربرد . . .

| INPUT | | | TARGET |
|---|---|---|---|
| $(x_1$ | $x_2$ | 1) | |
| (1 | 1 | 1) | 1 |
| (1 | 0 | 1) | 0 |
| (0 | 1 | 1) | 0 |
| (0 | 0 | 1) | 0 |




- تابع AND با ورودی‌ها و هدف‌های دودویی . . .
  - تغییر وزن

$$\Delta w_1 = x_1 t, \quad \Delta w_2 = x_2 t, \quad \Delta b = 1.t = t$$

$$\mathbf{w}(new) = \mathbf{w}(old) + \Delta\mathbf{w}$$

  - برای ورودی اول

$$x_1 = 1, \quad x_2 = 1, \quad b = 1, \quad t = 1$$

| INPUT | | | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|
| $(x_1$ | $x_2$ | 1) | $t$ | $(\Delta w_1$ | $\Delta w_2$ | $\Delta b)$ | $(w_1$ | $w_2$ | $b)$ |
| | | | | | | | (0 | 0 | 0) |
| (1 | 1 | 1) | 1 | (1 | 1 | 1) | (1 | 1 | 1) |

مقدار اولیه

$$x_2 = -x_1 - 1$$

# شبکهٔ هب: کاربرد . . .

- تابع AND با ورودی‌ها و هدف‌های دودویی
  - برای دومین، سومین و چهارمین ورودی

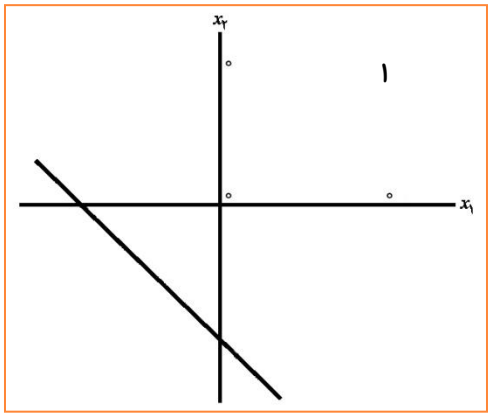
| INPUT | | | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $t$ | $\Delta w_1$ | $\Delta w_2$ | $b$ | $w_1$ | $w_2$ | $b$ |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

یادگیری رخ نمی‌دهد
وزن‌ها تغییر نمی‌کند

الگوهایی با مقدار هدف صفر یا «غیرفعال»

استفاده از نمایش دودویی

پاسخ نادرست




$$x_2 = -x_1 - 1$$

# شبکهٔ هب: کاربرد . . .

- تابع AND با ورودی‌های دودویی و مقادیر هدف دوقطبی

| INPUT | | | TARGET |
|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $t$ |
| 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | −1 |
| 0 | 1 | 1 | −1 |
| 0 | 0 | 1 | −1 |

- اولین ورودی

| INPUT | | | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

$$x_2 = -x_1 - 1$$

- ارائه دومین، سومین و چهارمین

| INPUT | | | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $t$ | $\Delta w_1$ | $\Delta w_2$ | $b$ | $w_1$ | $w_2$ | $b$ |
| 1 | 0 | 1 | −1 | −1 | 0 | −1 | 0 | 1 | 0 |
| 0 | 1 | 1 | −1 | 0 | −1 | −1 | 0 | 0 | −1 |
| 0 | 0 | 1 | −1 | 0 | 0 | −1 | 0 | 0 | −2 |

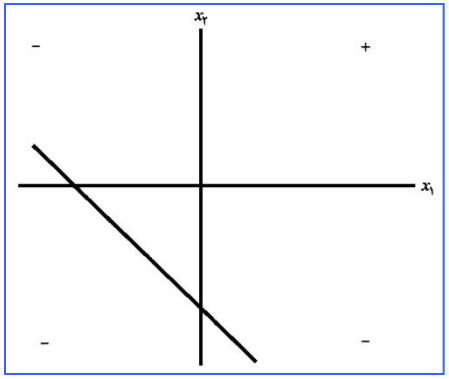
# شبکهٔ هب: کاربرد . . .

| INPUT | | | TARGET |
|---|---|---|---|
| $(x_1$ | $x_2$ | $1)$ | $t$ |
| $(1$ | $1$ | $1)$ | $1$ |
| $(1$ | $-1$ | $1)$ | $-1$ |
| $(-1$ | $1$ | $1)$ | $-1$ |
| $(-1$ | $-1$ | $1)$ | $-1$ |

- تابع AND برای ورودی‌ها و هدف‌های دوقطبی . . .
  - اولین ورودی

| INPUT | | | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|
| $(x_1$ | $x_2$ | $1)$ | $t$ | $(\Delta w_1$ | $\Delta w_2$ | $\Delta b)$ | $(w_1$ | $w_2$ | $b)$ |
| | | | | | | | $(0$ | $0$ | $0)$ |
| $(1$ | $1$ | $1)$ | $1$ | $(1$ | $1$ | $1)$ | $(1$ | $1$ | $1)$ |

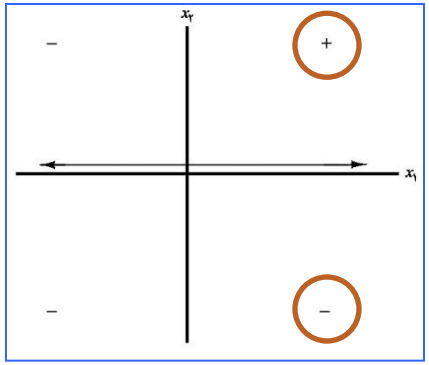
$$x_2 = -x_1 - 1$$




| INPUT | | | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|
| $(x_1$ | $x_2$ | $1)$ | $t$ | $(\Delta w_1$ | $\Delta w_2$ | $\Delta b)$ | $(w_1$ | $w_2$ | $b)$ |
| | | | | | | | $(1$ | $1$ | $1)$ |
| $(1$ | $-1$ | $1)$ | $-1$ | $(-1$ | $1$ | $-1)$ | $(0$ | $2$ | $0)$ |

$$x_2 = 0$$

  - دومین ورودی



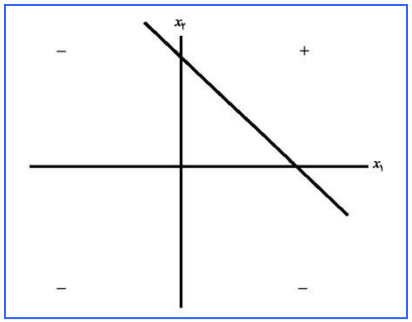

پاسخ درست برای
دو نمونه آموزش

# شبکهٔ هب: کاربرد . . .

- تابع AND برای ورودی‌ها و هدف‌های دوقطبی
  - سومین ورودی

| INPUT | | | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|
| $(x_1$ | $x_2$ | $1)$ | $t$ | $(\Delta w_1$ | $\Delta w_2$ | $\Delta b)$ | $(w_1$ | $w_2$ | $b)$ |
| | | | | | | | $(0$ | $2$ | $0)$ |
| $(-1$ | $1$ | $1)$ | $-1$ | $(1$ | $-1$ | $-1)$ | $(1$ | $1$ | $-1)$ |

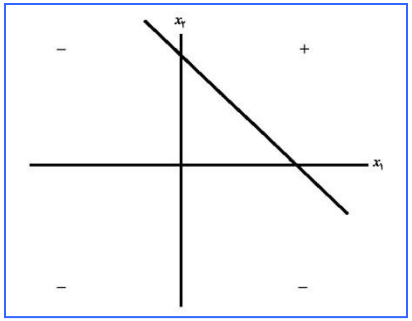
$$x_2 = -x_1 + 1$$




  - چهارمین ورودی

| INPUT | | | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|
| $(x_1$ | $x_2$ | $1)$ | $t$ | $(\Delta w_1$ | $\Delta w_2$ | $\Delta b)$ | $(w_1$ | $w_2$ | $b)$ |
| | | | | | | | $(1$ | $1$ | $-1)$ |
| $(-1$ | $-1$ | $1)$ | $-1$ | $(1$ | $1$ | $-1)$ | $(2$ | $2$ | $-2)$ |

$$x_2 = -x_1 + 1$$



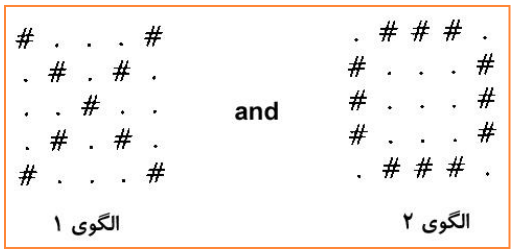

# شبکهٔ هب: نمایش داده‌ها

- **شکل نمایش داده‌ها می‌تواند مسئله قابل‌حل را به مسئله‌ای غیرقابل‌حل تبدیل کند**
  - **در قانون هب بسیار موثر است**
    - برای برخی الگوها منجر به جواب درست نمی‌شود، ممکن است برای نمایش متفاوتی از همان الگوها پاسخ درستی را نتیجه دهد

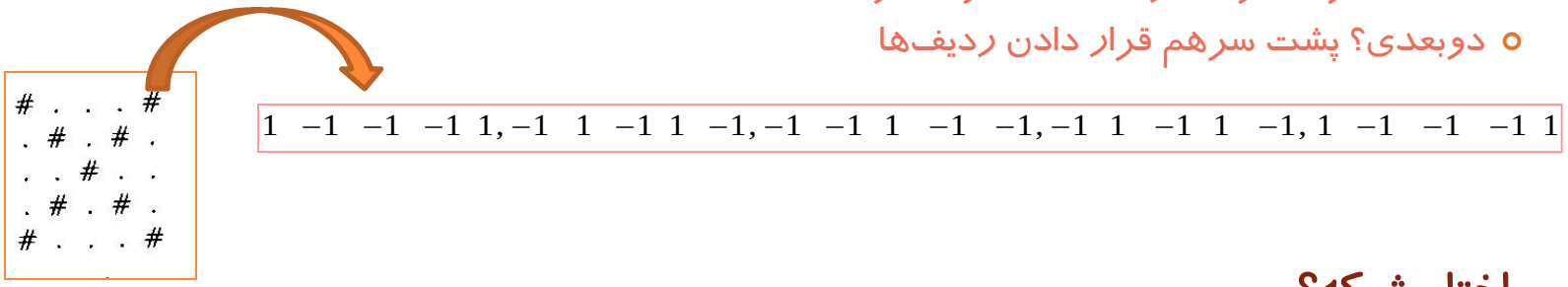
- **نمایش دوقطبی بهتر از نمایش دودویی است**
  - **افزایش قابلیت تعمیم شبکه**
  - **امکان تمایز داده‌های گم‌شده** (Missing Data) **از داده‌های اشتباه** (Mistaken Data)
    - مقادیر گم‌شده = «۰»
    - اشتباهات = قرینه مقدار ورودی از ۱+ به ۱-، یا برعکس

# شبکهٔ هب: مثال . . .

- بازشناسی نویسه (کاراکتر)- الگوهای ورودی دوبعدی . . .
  - یک شبکه هب برای تشخیص الگوی «X» از الگوی «O»



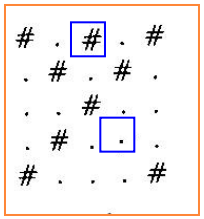

  - یک مسئلهٔ دسته‌بندی الگو؟
    - دسته «X» =خروجی مورد نظر و الگوی «O» = خروجی «غیرX»
    - روش دیگر؟
  - تبدیل الگوهای «O» و «X» به بردارهای ورودی؟
    - مقدار ۱ برای هر «#» و مقدار ۱- برای هر «۰»
    - دوبعدی؟ پشت سرهم قرار دادن ردیف‌ها

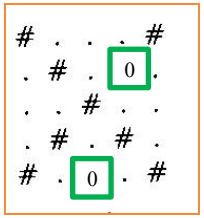



  - ساختار شبکه؟
    - تعداد نرون‌های ورودی = برابر با تعداد ابعاد بردار الگوها=۲۵

# شبکهٔ هب: مثال . . .

- بازشناسی نویسه (کاراکتر)- الگوهای ورودی دوبعدی . . .
  - قابلیت تعمیم شبکه
    - تولید پاسخ منطقی شبکه برای الگوهای ورودی شبیه الگوهای آموزش اما نه کاملاً یکسان با آنها




  - دو نوع تغییر در الگوی ورودی
    - «اشتباهات در داده‌ها»
      - علامت يك يا چند مؤلفة بردار ورودي قرينه شده و از 1 به 1- ، يا برعكس ، تغيير يافته است.
    - «داده‌های گم‌شده»
      - يك يا چند مؤلفة بردار ورودي به جاي مقدار 1 يا 1- مقدار صفر دارند

- شبکه در برخورد با داده‌های گم‌شده عملکرد بهتری در مقایسه با اشتباهات دارد
  - در مورد داده‌های ورودی، «بهتر است که حدس نزنیم»!!

# شبکه پرسپترون ...

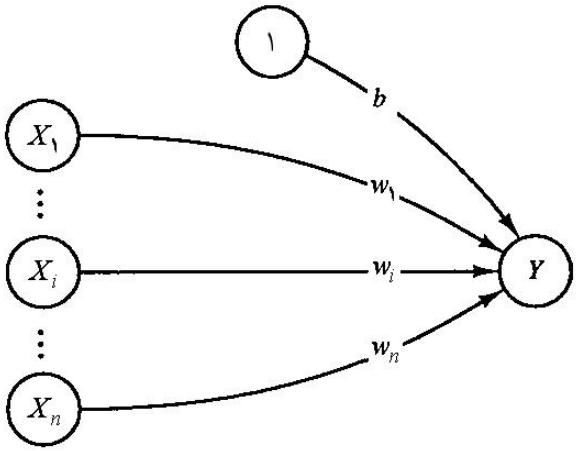
- جزو معروف‌ترین شبکه‌های عصبی است
  - حالت چند لایه آنها از پرکاربردترین شبکه‌های عصبی هستند
  - بیشترین اثرگذاری بر شبکه‌های عصبی اولیه
  - روزنبلات در سال ۱۹۶۲ و مینسکی و پاپرت در سال‌های ۱۹۶۹ و ۱۹۸۸

- قانون یادگیری قوی‌تر نسبت به قانون هب
  - یادگیری همراه با تکرار
    - در قانون هب، فقط یک بار (بدون تکرار) داده‌های آموزش به شبکه داده می‌شد
  - یادگیری پرسپترون شبیه قانون هب، تفاوت عمده: وزن‌ها فقط زمانی تغییر می‌کند که پاسخ شبکه به ازای آن ورودی دارای خطا باشد
    - خطا = خروجی محاسبه شده توسط شبکه با مقدار هدف یکی نباشد

# شبکه پرسپترون: ساختار . . .

- ساختاراولیه
  - سه لایه نرون (واحدهای حسی، واحدهای پیونددهنده، و واحد پاسخ)
    - فقط وزن‌های بین لایه‌های دوم و سوم آموزش داده می‌شود
    - خروجی واحدهای پیونددهنده به واحدهای پاسخ یک بردار دودویی است
    - عملاً یک شبکه یک لایه است
  - مدل تقریبی شبکیه چشم

- ساختار برای دسته‌بندی الگو
  - متعلق بودن به دسته با پاسخ ۱+
  - متعلق نبودن با پاسخ ۱-



# شبکه پرسپترون: الگوریتم . . .

- مرحلهٔ ۰ – مقداردهی اولیه به وزن‌ها و بایاس (مقدار صفر)

  تعیین نرخ یادگیری $0<\alpha\leq 1$ (مقدار ۱)
- مرحلهٔ ۱ – تا زمانی‌که شرایط توقف برقرار نیست، مراحل ۲ تا ۶ را انجام دهید
- مرحلهٔ ۲ – انجام مراحل ۳ تا ۵ برای هر جفت داده آموزش $s:t$
- مرحلهٔ ۳ – فعال‌سازی‌های واحدهای ورودی را مشخص کنید: $x_i = s_i$
- مرحلهٔ ۴ – پاسخ واحد خروجی را محاسبه کنید:

الگوریتم تکراری

$$y\_in = b + \sum_i x_i w_i$$

$$y = \begin{cases} 1 & if \;\; y\_in > \theta \\ 0 & if \;\; -\theta \leq y\_in \leq \theta \\ -1 & if \;\; y\_in < -\theta \end{cases}$$

جانشینی بایاس و آستانه؟

۱ = تعلق به دسته　　۱- = عدم تعلق به دسته　　۰ = ناحیه عدم تصمیم‌گیری $(2\theta)$

# شبکه پرسپترون: الگوریتم . . .

- مرحلهٔ ۵- اگر خطایی رخ داده است، وزن‌ها و بایاس را به‌روز کنید.

به‌روز کردن مشروط وزن‌ها

اگر $y \neq t$ است، آنگاه:

$$w_i(new) = w_i(old) + \alpha x_i t$$
$$b(new) = b(old) + \alpha t$$

در غیراین‌صورت:

$$w_i(new) = w_i(old)$$
$$b(new) = b(old)$$

- مرحلهٔ ۶- شرایط توقف را آزمایش کنید:
  - اگر در مرحلهٔ ۲ هیچ وزنی تغییر نکرد، الگوریتم را متوقف کنید، در غیراین‌صورت ادامه دهید.

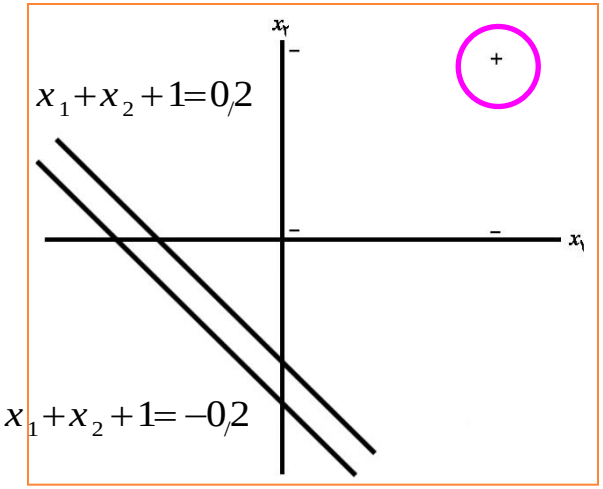
**خطا = برابر نبودن پاسخ شبکه و مقدار هدف**

**نرخ یادگیری**

# شبکه پرسپترون: کاربرد . . .

- تابع AND با ورودی‌های دودویی و هدف‌های دوقطبی . . .
  - وزن‌های اولیه و بایاس را صفر؛ نرخ اولیه یادگیری = ۱؛ آستانه =۰.۲
  - ارائه ورودی اول

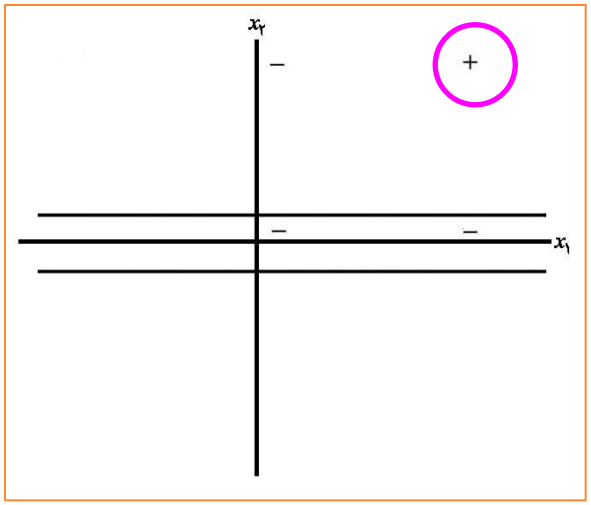
| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $y\_in$ | $y$ | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | | | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |



# شبکه پرسپترون: کاربرد . . .

- تابع AND با ورودی‌های دودویی و هدف‌های دوقطبی . . .
  - ارائه دومین ورودی

| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $y\_in$ | $y$ | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | | | 1 | 1 | 1 |
| 1 | 0 | 1 | 2 | 1 | −1 | −1 | 0 | −1 | 0 | 1 | 0 |




$x_2 = 0.2$

$x_2 = -0.2$

# شبکه پرسپترون: کاربرد . . .

- تابع AND با ورودی‌های دودویی و هدف‌های دوقطبی . . .
  - ارائه سومین ورودی

| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $y\_in$ | $y$ | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | | | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | −1 | 0 | −1 | −1 | 0 | 0 | −1 |

  - ارائه چهارمین ورودی
    - با توجه به برابر بودن پاسخ شبکه و مقدار هدف، وزن‌ها تغییری نمی‌کنند

| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $y\_in$ | $y$ | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | | | 0 | 0 | −1 |
| 0 | 0 | 1 | −1 | −1 | −1 | 0 | 0 | 0 | 0 | 0 | −1 |

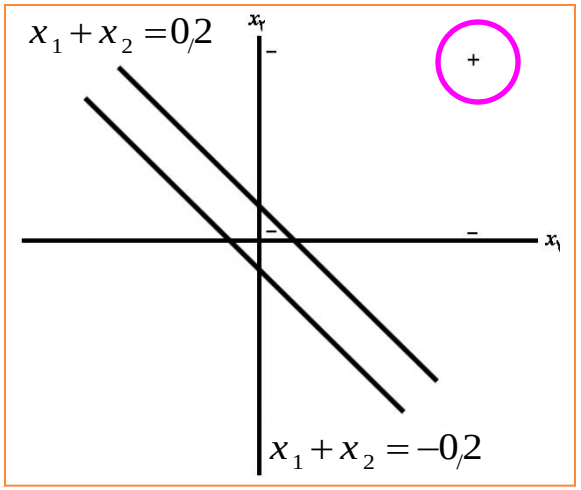
کامل شدن اولین دور آموزش (Epoch)

# شبکه پرسپترون: کاربرد . . .

- تابع AND با ورودی‌های دودویی و هدف‌های دوقطبی . . .
  - نیاز به تکرار؟؟ صحیح نبودن پاسخ برای اولین الگوی ورودی
  - تکراری بودن فرآیند آموزش (Iterative)

- دومین دور آموزش –ارائه اولین ورودی

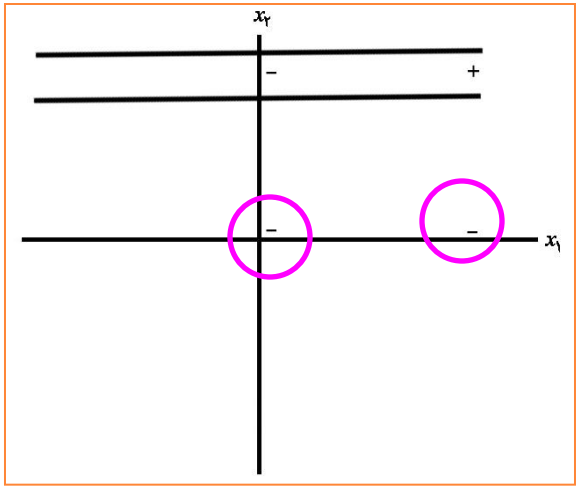
| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $y\_in$ | $y$ | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | | | 0 | 0 | –1 |
| 1 | 1 | 1 | –1 | –1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |



# شبکه پرسپترون: کاربرد . . .

- تابع AND با ورودی‌های دودویی و هدف‌های دوقطبی . . .
  - دومین دور آموزش –ارائه دومین ورودی

| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $(x_1$ | $x_2$ | $1)$ | $y\_in$ | $y$ | $t$ | $(\Delta w_1$ | $\Delta w_2$ | $\Delta b)$ | $(w_1$ | $w_2$ | $b)$ |
| | | | | | | | | | $(1$ | $1$ | $0)$ |
| $(1$ | $0$ | $1)$ | $1$ | $1$ | $-1$ | $(-1$ | $0$ | $-1)$ | $(0$ | $1$ | $-1)$ |




$x_2 - 1 = 0.2$

$x_2 - 1 = -0.2$

# شبکه پرسپترون: کاربرد . . .

- تابع AND با ورودی‌های دودویی و هدف‌های دوقطبی . . .
  - دومین دور آموزش –ارائه سومین ورودی
    - پاسخ برای تمام ورودی‌ها منفی

| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $y\_in$ | $y$ | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | | | 0 | 1 | –1 |
| 0 | 1 | 1 | 0 | 0 | –1 | 0 | –1 | –1 | 0 | 0 | –2 |

  - دومین دور آموزش –ارائه چهارمین ورودی
    - پاسخ برای تمام ورودی‌ها منفی

| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $y\_in$ | $y$ | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | | | 0 | 0 | –2 |
| 0 | 0 | 1 | –2 | –1 | –1 | 0 | 0 | 0 | 0 | 0 | –2 |

کامل شدن دومین دور آموزش ( Epoch)

# شبکه پرسپترون: کاربرد . . .

- تابع AND با ورودی‌های دودویی و هدف‌های دوقطبی . . .
  - سومین دور آموزش

| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $y\_in$ | $y$ | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | | | 0 | 0 | −2 |
| 1 | 1 | 1 | −2 | −1 | 1 | 1 | 1 | 1 | 1 | 1 | −1 |
| 1 | 0 | 1 | 0 | 0 | −1 | −1 | 0 | −1 | 0 | 1 | −2 |
| 0 | 1 | 1 | −1 | −1 | −1 | 0 | 0 | 0 | 0 | 1 | −2 |
| 0 | 0 | 1 | −2 | −1 | −1 | 0 | 0 | 0 | 0 | 1 | −2 |

  - چهارمین دور آموزش

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | −1 | −1 | 1 | 1 | 1 | 1 | 1 | 2 | −1 |
| 1 | 0 | 1 | 0 | 0 | −1 | −1 | 0 | −1 | 0 | 2 | −2 |
| 0 | 1 | 1 | 0 | 0 | −1 | 0 | −1 | −1 | 0 | 1 | −3 |
| 0 | 0 | 1 | −3 | −1 | −1 | 0 | 0 | 0 | 0 | 1 | −3 |

# شبکه پرسپترون: کاربرد . . .

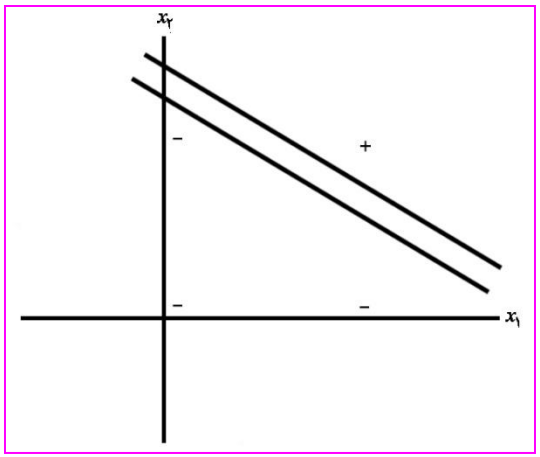
- تابع AND با ورودی‌های دودویی و هدف‌های دوقطبی . . .
  - پنجمین، ششمین، .... دور آموزش

$$\begin{array}{cccccccccc}(1 & 1 & 1) & 0 & 0 & 1 & (1 & 1 & 1) & (3 & 3 & -3)\\ (1 & 0 & 1) & 0 & 0 & -1 & (-1 & 0 & -1) & (2 & 3 & -4)\\ (0 & 1 & 1) & -1 & -1 & -1 & (0 & 0 & 0) & (2 & 3 & -4)\\ (0 & 0 & 1) & -4 & -1 & -1 & (0 & 0 & 0) & (2 & 3 & -4)\end{array}$$

  - نهمین دور آموزش

$$\begin{array}{cccccccccc}(1 & 1 & 1) & 1 & 1 & 1 & (0 & 0 & 0) & (2 & 3 & -4)\\ (1 & 0 & 1) & -2 & -1 & -1 & (0 & 0 & 0) & (2 & 3 & -4)\\ (0 & 1 & 1) & -1 & -1 & -1 & (0 & 0 & 0) & (2 & 3 & -4)\\ (0 & 0 & 1) & -4 & -1 & -1 & (0 & 0 & 0) & (2 & 3 & -4)\end{array}$$

  - دهمین دور آموزش
    - عدم تغییر وزن‌ها = توقف الگوریتم
    - همگرایی وزن‌ها




$$\begin{cases} 2x_1+3x_2-4>0{,}2 \Rightarrow x_2=-\dfrac{2}{3}x_1+\dfrac{7}{5} \\ 2x_1+3x_2-4<-0{,}2 \Rightarrow x_2=-\dfrac{2}{3}x_1+\dfrac{19}{15} \end{cases}$$

# شبکه پرسپترون: کاربرد

- تابع AND با ورودی‌ها و هدف‌های دوقطبی
  - آستانه، بایاس و وزن‌های اولیهٔ برابر با صفر؛ نرخ یادگیری برابر با ۱
  - دور اول آموزش

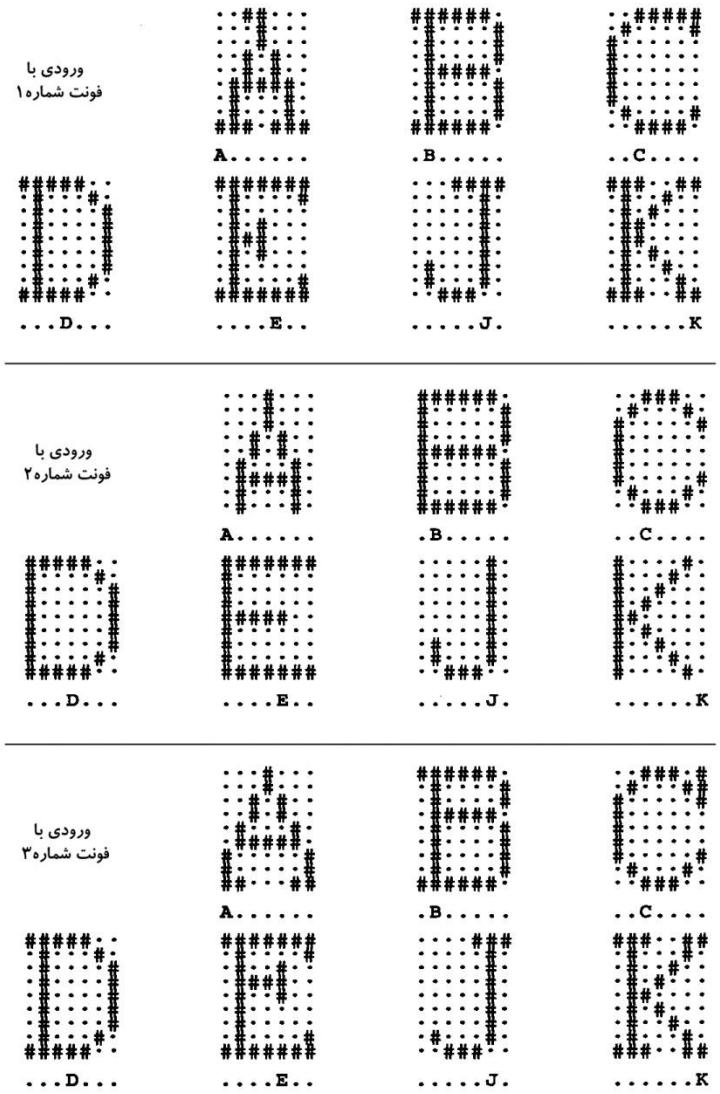
| INPUT | | | NET | OUT | TARGET | WEIGHT CHANGES | | | WEIGHTS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | $x_2$ | 1 | $y\_in$ | $y$ | $t$ | $\Delta w_1$ | $\Delta w_2$ | $\Delta b$ | $w_1$ | $w_2$ | $b$ |
| | | | | | | | | | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | −1 | 1 | 1 | 1 | −1 | −1 | 1 | −1 | 0 | 2 | 0 |
| −1 | 1 | 1 | 2 | 1 | −1 | 1 | −1 | −1 | 1 | 1 | −1 |
| −1 | −1 | 1 | −3 | −1 | −1 | 0 | 0 | 0 | 1 | 1 | −1 |

  - دور دوم آموزش

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | −1 |
| 1 | −1 | 1 | −1 | −1 | −1 | 0 | 0 | 0 | 1 | 1 | −1 |
| −1 | 1 | 1 | −1 | −1 | −1 | 0 | 0 | 0 | 1 | 1 | −1 |
| −1 | −1 | 1 | −3 | −1 | −1 | 0 | 0 | 0 | 1 | 1 | −1 |

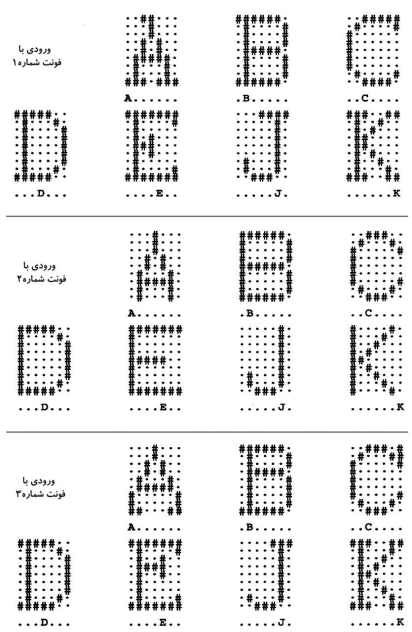
  - بهبود نتایج با تغییر نمایش دودویی به دوقطبی

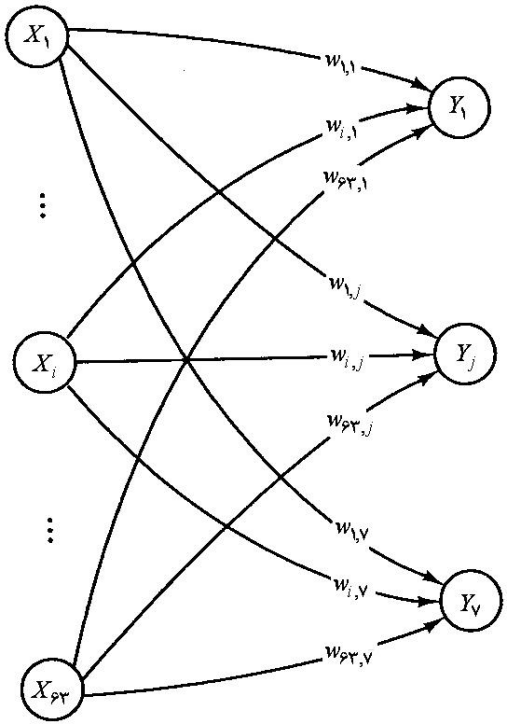
# شبکه پرسپترون: مثال . . .




- بازشناسی نویسه . . .
  - تشخیص حرف A از حرف غیر A
  - ۳ نوع فونت
    - ۳ نوع A= خروجی شبکه معادل با تعلق به دسته
    - ۱۸ نویسه غیر A= خروجی معادل با عدم تعلق به دسته
  - تعمیم برای سایر نویسه‌ها؟؟
    - یک شبکه جداگانه برای هر نویسه
    - شبکه‌ای با چندین (به تعداد نویسه‌ها) خروجی

# شبکه پرسپترون: مثال . . .




- بازشناسی نویسه . . .
  - نمایش دوقطبی
  - ۶۳ واحد ورودی (هر کدام یک پیکسل)
  - ۷ واحد خروجی (هر کدام یک نویسه)
    - بردار خروجی برای نمونه‌های A

$$(A.....) \Rightarrow (1,-1,-1,-1,-1,-1,-1)$$

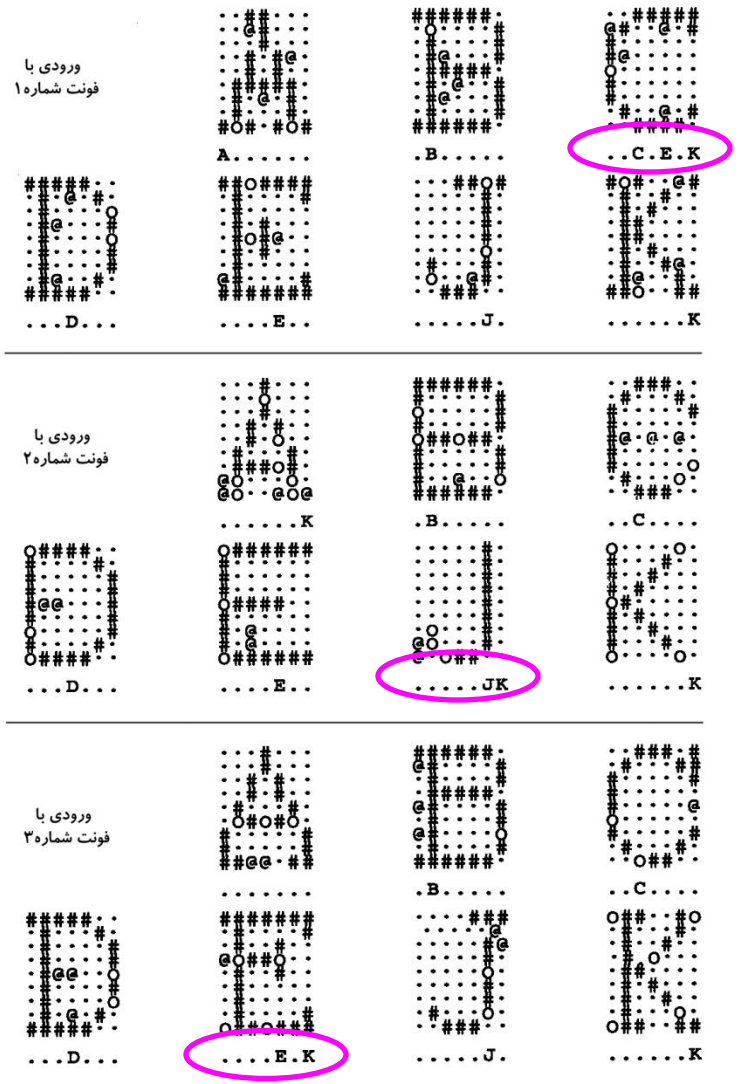
    - بردار خروجی برای نمونه‌های B

$$(.B....) \Rightarrow (-1,1,-1,-1,-1,-1,-1)$$

- دسته‌بندی درست داده‌های آموزش داده شده

# شبکه پرسپترون: مثال

- بازشناسی نویسه . . .
  - ارزیابی شبکه با ورودی‌های نویزی
    - بردارهای ورودی شبیه به بردارهای آموزش
      - و نه دقیقاً مانند آنها
    - جایگزینی پیکسل «غیرفعال» با «فعال» = @
    - جایگزینی پیکسل «فعال» با «غیرفعال» = O



# شبکه پرسپترون: همگرایی قانون یادگیری

- **قضیه**
  - اگر بردار وزن $\mathbf{w}^*$ وجود داشته باشد به‌طوری‌که برای تمام p‌ها داشته باشیم:

$$f(x(p).\ \mathbf{w}^*) = t(p)$$

**آنگاه برای هر بردار اولیه $\mathbf{w}$، قانون یادگیری پرسپترون به بردار وزنی نزدیک می‌شود (نه الزاماً منحصر به فرد و نه الزاماً $\mathbf{w}^*$) که برای تمام الگوهای آموزش پاسخ صحیحی می‌دهد و این کار در مراحلی با تعداد متناهی انجام می‌شود.**

- $p$= تعداد بردارهای ورودی آموزش
- $x(p)$ = بردارهای ورودی آموزش
- $t(p)$= مقدار هدف معادل بردارهای ورودی آموزش (دوقطبی)
- $f$ = تابع فعال‌سازی خروجی

# شبکه پرسپترون: نکات تکمیلی

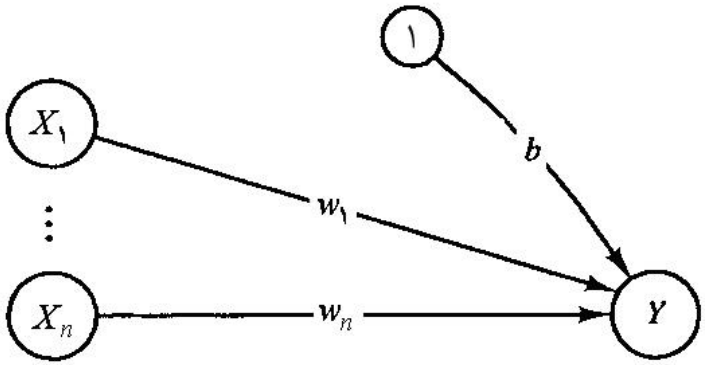
- مقداردهی نرخ یادگیری
  - مقدار ثابت غیرمنفی
  - مقدار $1/\|\mathbf{x}\|$ ؛ تا تغییر وزن یک بردار واحد باشد
  - مقدار $(\mathbf{x}.\mathbf{w}) / \|\mathbf{x}\|^2$
- مقادیر اولیه وزن‌ها
  - مقدار ثابت صفر
  - یک الگوی آموزش اختیاری
  - مقادیر تصادفی کوچکی

# شبکه آدالاین ...

- آدالاین = نرون خط وفقی(ADAptive LInear Neuron)
  - توسط ویدور و هاف در سال ۱۹۶۰
  - دارای قانون یادگیری متفاوت با هب و پرسپترون
  - قانون یادگیری= قانون دلتا = قانون میانگین مربعات کمینه (LMS) = قانون ویدرو-هاف
    - میانگین مربعات خطای بین مقدار خروجی شبکه و مقدار هدف در هر مرحله از آموزش کاهش یابد
  - استفاده از فعال‌سازی‌های دوقطبی برای سیگنال‌های وروی و خروجی
  - تابع فعال‌سازی خروجی = تابع همانی

# شبکه آدالاین: ساختار

- ساختار مشابه با سایر شبکه‌های قبلی
  - چند ورودی
  - بایاس = ورودی برابر با ۱




- قابلیت توسعه به حالت چندلایه = شبکهٔ مادالاین

# شبکه آدالاین: الگوریتم . . .

- مرحلهٔ ۰- مقداردهی اولیه به وزن‌ها (مقادیر تصادفی کوچک)
  مقداردهی به نرخ یادگیری
- مرحلهٔ ۱- تا زمانی‌که شرایط توقف برقرار نیست، مراحل ۲ تا ۶ را انجام دهید.
- مرحلهٔ ۲- برای هر جفت آموزش دوقطبی $s:t$ مراحل ۳ تا ۵ را انجام دهید.
- مرحلهٔ ۳- فعال‌سازی‌های واحدهای ورودی را مشخص کنید: $x_i = s_i \qquad i = 1,\ldots,n$
- مرحلهٔ ۴- مقدار ورودی شبکه را به واحد خروجی محاسبه کنید: $y\_in = b + \sum_i x_i w_i$
- مرحلهٔ ۵- مقادیر وزن‌ها و بایاس را به‌روز کنید:
$$\begin{cases} b(new) = b(old) + \alpha.(t - y\_in) \\ w_i(new) = w_i(old) + \alpha.(t - y\_in).x_i \end{cases}$$
- مرحلهٔ ۶- شرایط توقف را آزمایش کنید:
  اگر بزرگ‌ترین تغییر وزنی که در مرحلهٔ ۲ رخ داده است از یک مقدار کوچک کم‌تر باشد، الگوریتم را متوقف کنید، وگرنه ادامه دهید.

# شبکه آدالاین: الگوریتم

- **تفاوت یادگیری آدالاین با یادگیری پرسپترون و هب**
  - تغییر وزن‌ها متناسب با میزان تفاوت پاسخ شبکه به یک ورودی و مقدار هدف متناظر این ورودی است.
  - دربرگیرنده مفهوم خطا (که در یادگیری پرسپترون نیز وجود دارد)

- **نرخ یادگیری**
  - تاثیر بر سرعت و روند همگرایی الگوریتم
  - نیاز به اختصاص مقدار مناسب
  - دارای کران بالا از نظر تئوری
  - روش: ابتدا مقدار را کوچک فرض کرده (مثلاً ۰.۱) و به مرور مقدار آن را بزرگ کنیم
  - اگر مقدار خیلی بزرگی باشد، فرآیند یادگیری همگرا نخواهد بود
  - اگر مقدار بسیار کوچکی باشد، یادگیری بسیار کند می‌شود

# شبکه آدالاین: کاربرد . . .

- تابع AND: ورودی‌های دودویی، هدف‌های دوقطبی
  - شبکه بعد از آموزش

| $x_1$ | $x_2$ | $t$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | −1 |
| 0 | 1 | −1 |
| 0 | 0 | −1 |

$$w_1 = 1 \quad w_2 = 1 \quad w_0 = -\frac{3}{2}$$

$$x_1 + x_2 - \frac{3}{2} = 0$$

  - مربعات خطا برای چهار الگوی آموزش با این وزن‌ها = ۱

$$e = E\{(\hat{t} - t)^2\} = \sum_{p=1}^{4} [\{x_1(p).w_1 + x_2(p).w_2 + w_0\} - t(p)]^2$$

# شبکه آدالاین: کاربرد . . .

- تابع AND: ورودی‌ها و هدف‌های دوقطبی
  - شبکه بعد از آموزش

$$w_1 = \frac{1}{2} \quad w_2 = \frac{1}{2} \quad w_0 = -\frac{1}{2}$$

$$\frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{1}{2} = 0$$

  - جواب یکسان با الگوریتم پرسپترون

# شبکه آدالاین: کاربرد

- تابع AND NOT: ورودی‌ها و هدف‌های دوقطبی

| $x_1$ | $x_2$ | $t$ |
|---|---|---|
| 1 | 1 | −1 |
| 1 | −1 | 1 |
| −1 | 1 | −1 |
| −1 | −1 | −1 |

$$w_1 = \frac{1}{2} \quad w_2 = -\frac{1}{2} \quad w_0 = -\frac{1}{2}$$

- تابع OR: ورودی‌ها و هدف‌های دوقطبی

| $x_1$ | $x_2$ | $t$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | −1 | 1 |
| −1 | 1 | 1 |
| −1 | −1 | −1 |

$$w_1 = \frac{1}{2} \quad w_2 = \frac{1}{2} \quad w_0 = \frac{1}{2}$$

# شبکه آدالاین: قانون یادگیری

## قانون دلتا

- کمینه کردن خطای بین خروجی شبکه و مقدار هدف متناظر
- خطا = مربعات تفاضل

$$E = (t - y\_in)^2$$

$$y\_in = \sum_{i=1}^{n} x_i w_i$$

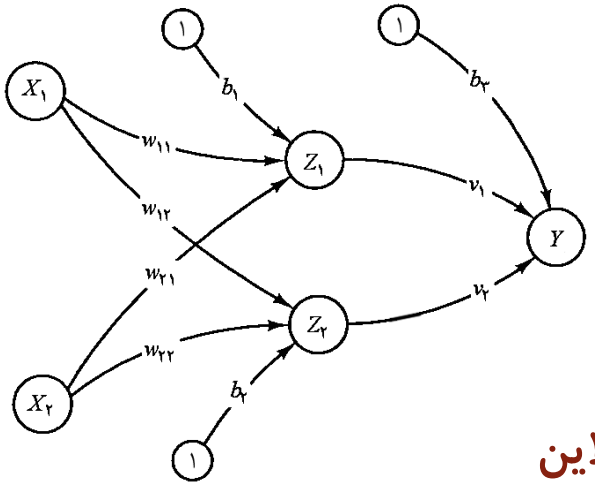
- گرادیان تابع خطا= مشتق‌های جزئی خطا نسبت به هر یک از وزن‌ها
- گرادیان بیانگر جهت سریع‌ترین رشد خطا
- جهت مخالف گرادیان = سریع‌ترین کاهش خطا

$$-\frac{\partial E}{\partial w_I} = -2(t - y\_in)\frac{\partial y\_in}{\partial w_I} = -2(t - y\_in)x_I$$

$$\Delta w_I = \alpha(t - y\_in)x_I$$

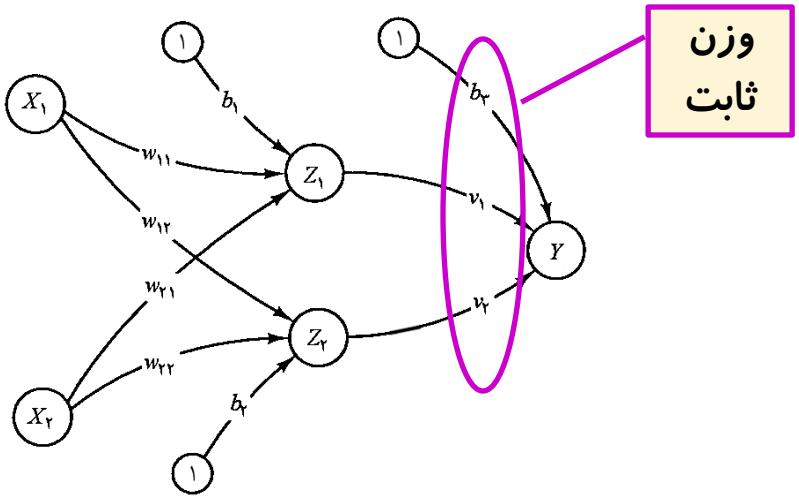
# شبکه مادالاین . . .

- حالت چند لایه آدالاین
  - ترکیب چندین واحد آدالاین در یک شبکهٔ یک لایه با هم = عدم تغییر در فرآیند آموزش
  - برای بیش از یک لایه = نیاز به آموزش متفاوت
  - شبکه چند لایه = افزایش قابلیت‌های محاسباتی شبکه = حل مسائل پیچیده‌تر
- ساختار




- الگوریتم
  - MRI = شیوهٔ اصلی آموزش مادالاین
    - آموزش وزن‌های لایه اول و ثابت گرفتن وزن‌های لایه دوم (محاسبه به صورت شهودی)
  - MRII: روشی دیگر

# شبکه مادالاین: آموزش با MRI . . .




- تعیین وزن ثابت
  - به صورت شهودی
  - شبیه سازی تابع OR با Y (y=z1 OR z2)

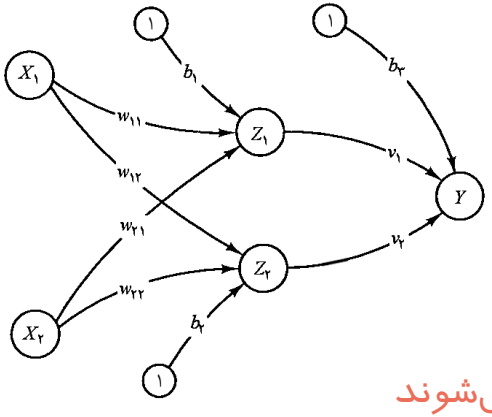
$$v_1 = \frac{1}{2} \qquad v_2 = \frac{1}{2} \qquad b_3 = \frac{1}{2}$$

  - می‌تواند AND هم باشد

- تابع فعال‌سازی

$$f(x) = \begin{cases} 1 & if\ x \geq 0; \\ -1 & if\ x < 0. \end{cases}$$

# شبکه مادالاین: آموزش با MRI . . .




- مرحلهٔ ۰ – مقداردهی اولیه وزن‌ها

  وزن‌های v1 و v2 و بایاس b3 به صورت شهودی (مشابه قبل) تعیین می‌شوند

  وزن‌های اولیه آدالاین با مقادیر کوچک تصادفی

  مقداردهی نرخ یادگیری مانند الگوریتم آدالاین (یک مقدار کوچک)

- مرحلهٔ ۱ – تا زمانی‌که شرایط توقف برقرار نیست، مراحل ۲ تا ۸ را انجام دهید.
- مرحلهٔ ۲ – برای هر جفت آموزش دوقطبی $s:t$ مراحل ۳ تا ۷ را انجام دهید.
- مرحلهٔ ۳ – فعال‌سازی‌های واحدهای ورودی را تعیین کنید: $x_i = s_i$
- مرحلهٔ ۴ – ورودی شبکه به هر واحد آدالاین مخفی را محاسبه کنید:

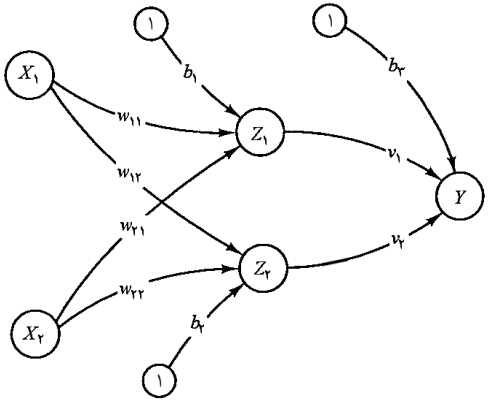
$$z\_in_1 = b_1 + x_1 w_{11} + x_2 w_{21} \qquad z\_in_2 = b_2 + x_1 w_{12} + x_2 w_{22}$$

- مرحلهٔ ۵ – خروجی هر واحد آدالاین مخفی را تعیین کنید: $z_1 = f(z\_in_1) \quad z_2 = f(z\_in_2)$
- مرحلهٔ ۶ – خروجی شبکه را تعیین کنید: $y\_in = b_3 + z_1 v_1 + z_2 v_2 \qquad y = f(y\_in)$
- مرحلهٔ ۷ – خطا را تعیین کنید و وزن‌ها را به‌روز کنید:



H. Veisi (veisi@sharif.edu)

# شبکه مادالاین: آموزش با MRI




- **مرحلهٔ ۷- خطا را تعیین کنید و وزن‌ها را به‌روز کنید:**
  - اگر y=t است، به‌روز کردن وزن اجرا نمی‌شود،
  - در غیراین‌صورت:
  - اگر t=1 بود، وزن‌های روی $Z_J$ را به‌روز کنید، (واحدی که ورودی شبکه آن به صفر نزدیک‌تر است)

$$b_J(new) = b_J(old) + \alpha(1 - z\_in_J) \qquad w_{iJ}(new) = w_{iJ}(old) + \alpha(1 - z\_in_J)x_i$$

  - اگر t=-1 بود، وزن‌های روی تمام واحدهای $Z_k$ که ورودی شبکه مثبت دارند را به‌روز کنید،

$$b_k(new) = b_k(old) + \alpha(1 - z\_in_k) \qquad w_{ik}(new) = w_{ik}(old) + \alpha(1 - z\_in_k)x_i$$

به خاطر OR بودن خروجی

- **مرحلهٔ ۸- شرایط توقف را آزمایش کنید.**
  - اگر
    - تغییر وزن‌ها متوقف شده است یا به سطح قابل قبولی رسیده است یا
    - به‌روز شدن وزن‌ها در مرحلهٔ 2 به سقف تعداد تکرارهای مشخص شده رسیده است
  - کار را متوقف کنید،
  - در غیراین‌صورت ادامه دهید.

# شبکه مادالاین: آموزش با MRII . . .

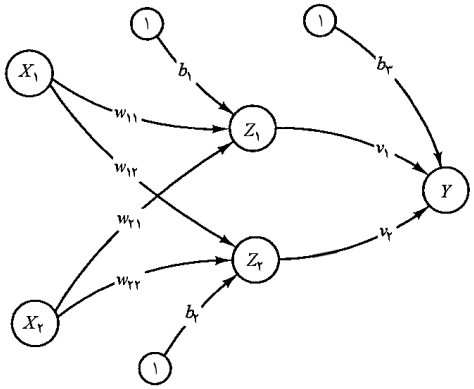
- مرحلهٔ ۰- مقداردهی اولیه وزن‌ها و نرخ یادگیری
- مرحلهٔ ۱- تا زمانی‌که شرایط توقف برقرار نیست، مراحل ۲ تا ۸ را انجام دهید.
- مرحلهٔ ۲- برای هر جفت آموزش دوقطبی $s:t$، مراحل ۳ تا ۷ را انجام دهید.
- مرحلهٔ ۳ تا ۶- خروجی شبکه را همانند الگوریتم MRI محاسبه کنید.
- مرحلهٔ ۷- خطا را مشخص کنید و در صورت لزوم وزن‌ها را به‌روز کنید:
  - اگر $t \neq y$، مراحل ۷-الف و ۷-ب را برای هر واحد مخفی که ورودی شبکهٔ آن به اندازه کافی به صفر نزدیک است (مثلاً بین مقادیر ۰.۲۵ و ۰.۲۵-)، انجام دهید. با واحدی شروع کنید که ورودی شبکه آن به صفر نزدیک‌تر است، سپس آن را برای دومین واحد نزدیک به صفر تکرار نموده و به همین ترتیب ..
  - مرحلهٔ ۷-الف- خروجی واحد را تغییر دهید (۱+ به ۱-، یا برعکس).
  - مرحلهٔ ۷-ب- پاسخ شبکه را دوباره محاسبه کنید.

    اگر خطا کاهش یافته است، وزن‌های این واحد را تنظیم کنید. از مقدار خروجی جدید به عنوان مقدار هدف استفاده شود و قانون دلتا به کار گرفته شود
- مرحلهٔ ۸- شرایط توقف را آزمایش کنید.
  - اگر تغییر وزن‌ها متوقف شده یا میزان تغییرات به سطح قابل قبولی رسیده است و یا اگر به سقف تعداد مشخص شده برای تکرارهای به‌روز شدن وزن در مرحلهٔ ۲ رسیده باشد، کار را متوقف کنید،



H. Veisi (veisi@sharif.edu)

# شبکه مادالاین: کاربرد . . .




| $x_1$ | $x_2$ | $t$ |
|---|---|---|
| 1 | 1 | −1 |
| 1 | −1 | 1 |
| −1 | 1 | 1 |
| −1 | −1 | −1 |

- مادالاین برای تابع XOR...
  - مرحلهٔ ۰- وزن‌های اولیه Z1 و Z2 با مقادیر تصادفی کوچک
    وزن‌های Y تابع OR را می‌سازند، مقادیر این وزن‌ها

$$x_1 \ \mathbf{XOR}\ x_2 \leftrightarrow (x_1 \ \mathbf{AND NOT}\ x_2)\ \mathbf{OR}\ (x_2 \ \mathbf{AND NOT}\ x_1)$$

$$v_1 = \frac{1}{2} \qquad v_2 = \frac{1}{2} \qquad b_3 = \frac{1}{2}$$

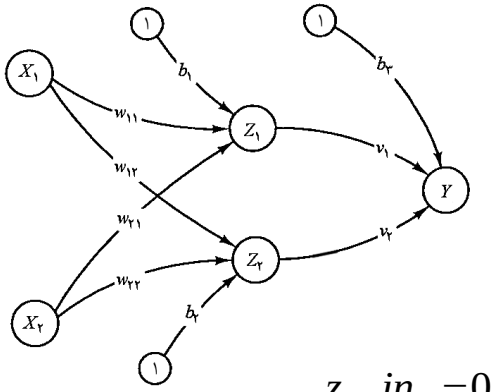
نرخ یادگیری را نیز برابر ۰.۵ قرار می‌دهیم

| Weights into $Z_1$ | | | Weights into $Z_2$ | | | Weights into Y | | |
|---|---|---|---|---|---|---|---|---|
| $w_{11}$ | $w_{21}$ | $b_1$ | $w_{12}$ | $w_{22}$ | $b_2$ | $v_1$ | $v_2$ | $b_3$ |
| 0,05 | 0,2 | 0,3 | 0,1 | 0,2 | 0,15 | 0,5 | 0,5 | 0,5 |

- مرحلهٔ ۱- آموزش را آغاز کنید.
- مرحلهٔ ۲- برای جفت آموزشی اول، $(1,1) : -1$:

# شبکه مادالاین: کاربرد . . .

| $w_{11}$ | $w_{21}$ | $b_1$ | $w_{12}$ | $w_{22}$ | $b_2$ | $v_1$ | $v_2$ | $b_3$ |
|---|---|---|---|---|---|---|---|---|
| 0/05 | 0/2 | 0/3 | 0/1 | 0/2 | 0/15 | 0/5 | 0/5 | 0/5 |




## مادالاین برای تابع XOR . . .

- مرحلهٔ ۲- برای جفت آموزشی اول، $(1,1):-1$ :
- مرحلهٔ ۳- $x_1=1,\ x_2=1$
- مرحلهٔ ۴- $z\_in_1=0/3+0/05+0/2=0/55$ $\quad z\_in_2=0/15+0/1+0/2=0/45$
- مرحلهٔ ۵- $z_1=1,\ z_2=1$
- مرحلهٔ ۶- $y\_in=0/5+0/5+0/5 \Rightarrow y=1$
- مرحلهٔ ۷- $t \neq y$ پس خطایی پیش آمده است.

چون $t=-1$، و هر دو واحد Z ورودی مثبت دارند، به‌روز کردن وزن‌های واحد Z1

$$b_1(new)=b_1(old)+\alpha(-1-z\_in_1)=0/3+(0/5)(-1/55)=-0/475$$
$$w_{11}(new)=w_{11}(old)+\alpha(-1-z\_in_1)x_1=0/5+(0/5)(-1/55)=-0/725$$
$$w_{21}(new)=w_{21}(old)+\alpha(-1-z\_in_1)x_2=0/2+(0/5)(-1/55)=-0/575$$

واحد Z2

$$b_2(new)=b_2(old)+\alpha(-1-z\_in_2)=0/15+(0/5)(-1/45)=-0/575$$
$$w_{12}(new)=w_{12}(old)+\alpha(-1-z\_in_2)x_1=0/1+(0/5)(-1/45)=-0/625$$
$$w_{21}(new)=w_{21}(old)+\alpha(-1-z\_in_1)x_2=0/2+(0/5)(-1/45)=-0/525$$

ادامه به همین ترتیب
وزن‌های نهایی در تکرار چهارم

| | |
|---|---|
| $w_{11}=-0/73$ | $w_{12}=1/27$ |
| $w_{21}=1/53$ | $w_{22}=-1/33$ |
| $b_1=-0/99$ | $b_2=-1/09$ |

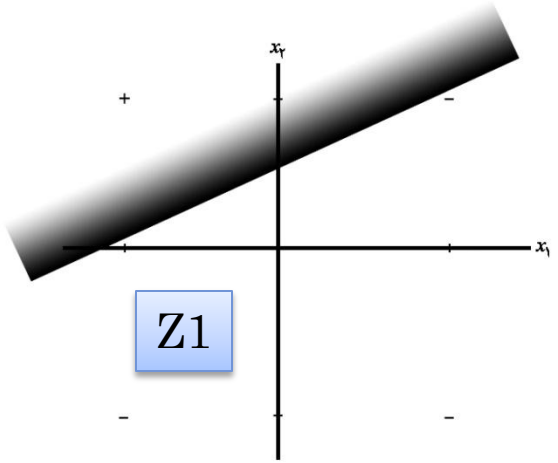
# شبکه مادالاین: کاربرد

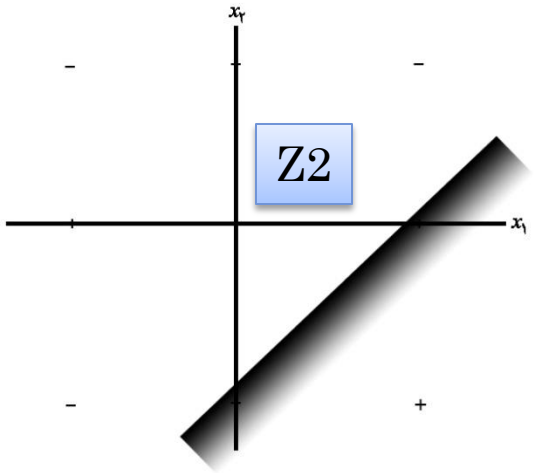
- مادالاین برای تابع XOR
  - نمایش هندسی پاسخ
  - پاسخ مثبت = اجتماع ناحیه‌هایی که هر یک از واحدهای مخفی یک پاسخ مثبت دارند
  - مرز تصمیم‌گیری برای واحد مخفی Z1

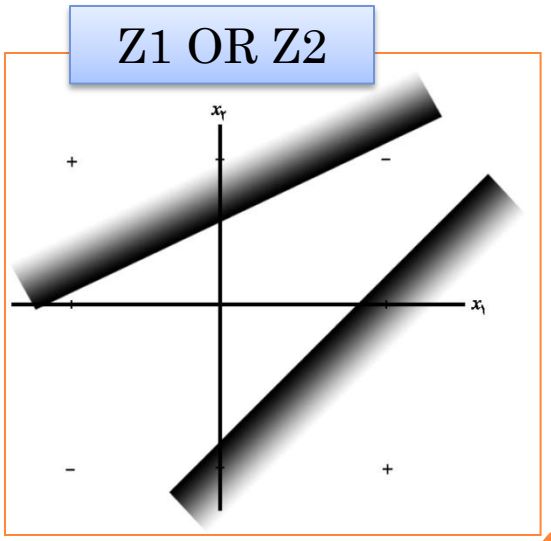
$$x_2 = -\frac{w_{11}}{w_{21}}x_1 - \frac{b_1}{w_{21}} = \frac{0{,}73}{1{,}53}x_1 + \frac{0{,}99}{1{,}53} = 0{,}48x_1 + 0{,}65$$

  - مرز تصمیم‌گیری برای واحد مخفی Z2

$$x_2 = -\frac{w_{12}}{w_{22}}x_1 - \frac{b_2}{w_{22}} = \frac{1{,}27}{1{,}33}x_1 + \frac{-1{,}09}{1{,}33} = 0{,}96x_1 - 0{,}82$$



# تمرین شماره ۱

- ارائه شده بر روی وب سایت
  - سوالات نظری + پیاده‌سازی
  - تاریخ تحویل