شبکه‌های عصبی مصنوعی

درس ۱۵

یادگیری پیوندی

Associative Learning

کاظم فولادی قلعه
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

http://courses.fouladi.ir/nn

Associative Learning

یادگیری پیوندی

ASSOCIATIVE LEARNING

یادگیری شبکه‌هایی که تاکنون دیدیم، به شیوه‌ی بانظارت بود:
داده‌ها نیاز به target داشتند.

در این فصل مجموعه‌ای از قواعد ساده که امکان یادگیری بدون نظارت را فراهم می‌کنند، معرفی می‌شود.

این قواعد به شبکه توانایی می‌دهند تا پیونددهی (association) میان الگوهایی که
به طور مکرر با هم رخ می‌دهند را یاد بگیرد.

پس از یادگیری، شبکه می‌تواند وظایف مفیدی
چون بازشناسی الگو (recognition) و بازفراخوانی (recall) را انجام دهد.

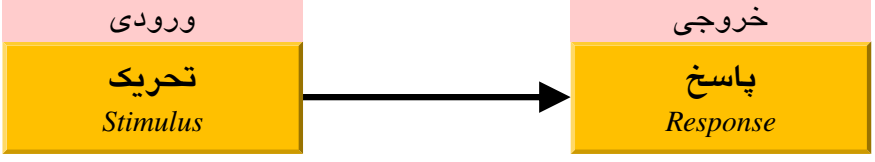
در این فصل: بررسی شکل بدون نظارت یادگیری هبی و دیگر قواعد یادگیری پیوندی مربوط به آن
کارهای: اندرسون، کوهونن، گراسبرگ و ...

پیونددهی

ASSOCIATION

پیونددهی *Association*	ایجاد هرنوع پیوند میان یک ورودی و خروجی سیستم (مثلاً وقتی یک الگوی A به سیستم ارائه می‌شود، سیستم با الگوی B پاسخ می‌دهد.)

وقتی الگوها به‌وسیله‌ی یک association پیوند داده می‌شوند،
از الگوی ورودی با عنوان **تحریک** و از الگوی خروجی با عنوان **پاسخ** یاد می‌شود.



association : شکل‌دهنده‌ی مبانی مکتب رفتارگرا در روان‌شناسی
- **ایوان پائولوف** (شرطی‌سازی کلاسیک [سگ – زنگ])
- **اسکینر** (تأثیرگذارترین طرفدار این مکتب) (شرطی‌سازی ابزاری [موش – میله])
- **دونالد هب** (تبیین بیولوژیکی بعضی از این رفتارها)

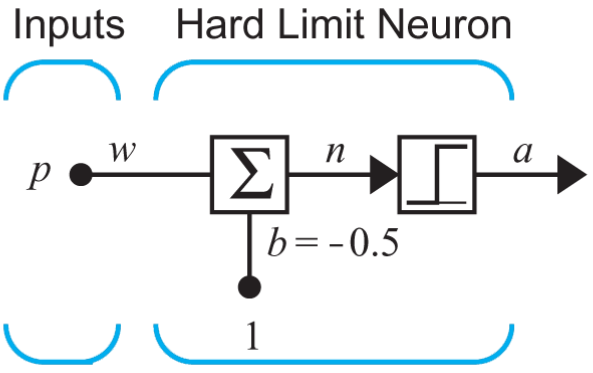
شبکه‌های عصبی مصنوعی

یادگیری پیوندی

۱

شبکه‌ی پیوندی ساده

Simple Associative Network

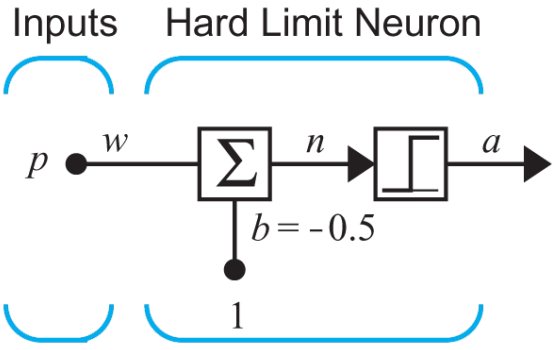


$$a = hardlim(wp + b) = hardlim(wp - 0.5)$$

$$p = \begin{cases} 1, & \text{stimulus} \\ 0, & \text{no stimulus} \end{cases} \qquad a = \begin{cases} 1, & \text{response} \\ 0, & \text{no response} \end{cases}$$

شبکه‌ی پیوندی ساده

SIMPLE ASSOCIATIVE NETWORK

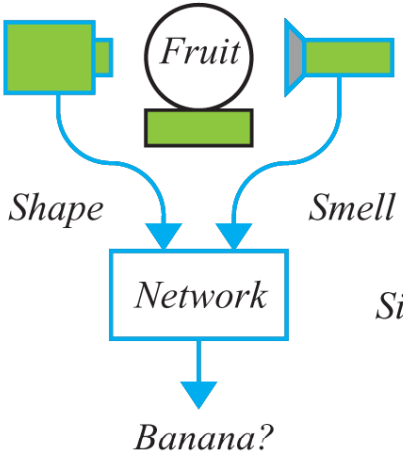


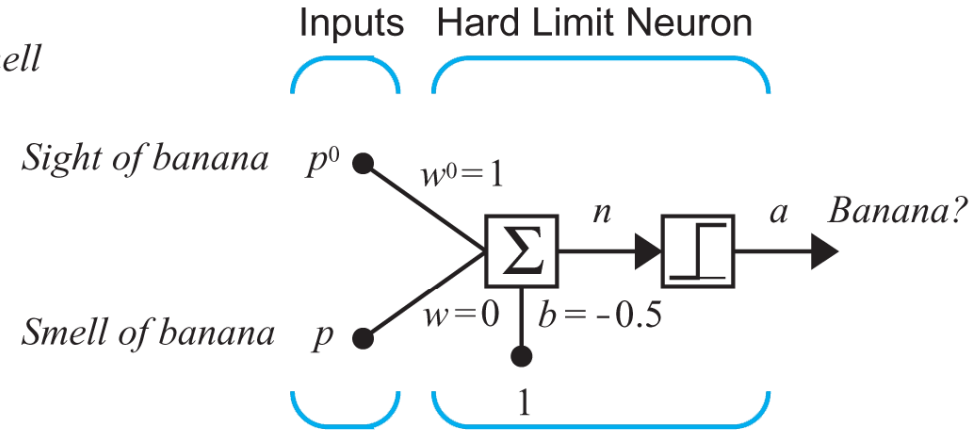
$$a = hardlim(wp + b) = hardlim(wp - 0.5)$$

$$p = \begin{cases} 1, & \text{stimulus} \\ 0, & \text{no stimulus} \end{cases} \qquad a = \begin{cases} 1, & \text{response} \\ 0, & \text{no response} \end{cases}$$

وجود پیوند بین $p = 1$ و $a = 1$ توسط مقدار w دیکته می‌شود ← $w > -b$

Banana Associator





$$a = hardlim(w^0 p^0 + w p + b)$$

Unconditioned Stimulus

$$p^0 = \begin{cases} 1, & \text{shape detected} \\ 0, & \text{shape not detected} \end{cases}$$

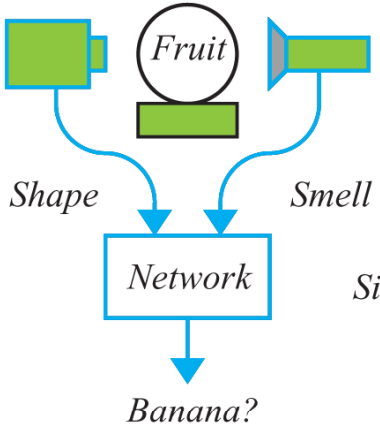
Conditioned Stimulus

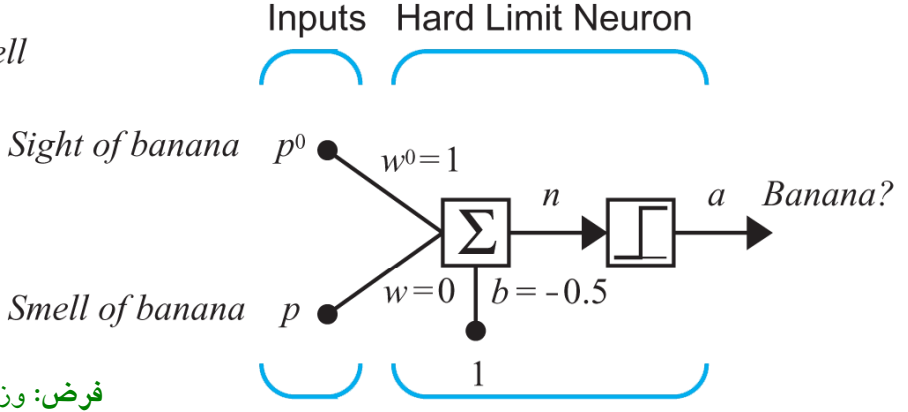
$$p = \begin{cases} 1, & \text{smell detected} \\ 0, & \text{smell not detected} \end{cases}$$

شبکه‌ی پیوندی ساده

مثال

BANANA ASSOCIATOR





فرض: وزن وابسته به p_0 ثابت است
اما وزن وابسته به p با قاعده‌ی یادگیری تنظیم می‌شود.

$$a = hardlim(w^0 p^0 + w p + b)$$

Unconditioned Stimulus

$$p^0 = \begin{cases} 1, & \text{shape detected} \\ 0, & \text{shape not detected} \end{cases}$$

تحریک شرطی نشده

Conditioned Stimulus

$$p = \begin{cases} 1, & \text{smell detected} \\ 0, & \text{smell not detected} \end{cases}$$

تحریک شرطی شده

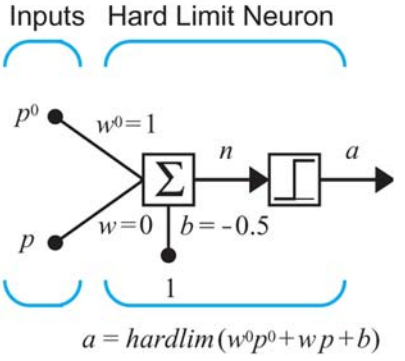
دو مجموعه‌ی ورودی

Prepared by Kazim Fouladi | Fall 2017 | 2nd Edition
کاظم فولادی

شبکه‌ی پیوندی ساده

تحریک

STIMULUS



تحریک شرطی‌شده
Conditioned Stimulus

p

مثل به‌صدا درآمدن زنگ در آزمایش پائولوف
(زنگ به‌همراه غذا به طور مکرر به صدا درمی‌آید:
⇐ بزاق سگ به صدای زنگ شرطی می‌شود.)

تحریک شرطی‌نشده
Conditioned Stimulus

p^0

مثل نمایش غذا به سگ در آزمایش پائولوف
(ترشح بزاق ابتدائاً فقط در اثر این مورد است:
غریزی، بدون نیاز به یادگیری)

شبکه‌های عصبی مصنوعی

یادگیری پیوندی

۲

قاعده‌ی هب بدون نظارت

Unsupervised Hebb Rule

$$w_{ij}(\mathrm{q}) = w_{ij}(q-1) + \alpha a_i(q) p_j(q)$$

Vector Form:

$$\mathbf{W}(q) = \mathbf{W}(q-1) + \alpha \mathbf{a}(q)\mathbf{p}^T(q)$$

Training Sequence:

$$\mathbf{p}(1), \mathbf{p}(2), \ldots, \mathbf{p}(Q)$$

قاعده‌ی هب بدون نظارت

UNSUPERVISED HEBB RULE

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q)$$

قاعده‌ی هب بدون نظارت، یک **یادگیری محلی (local learning)** است:
تنها از سیگنال‌های موجود در لایه‌ای که وزن‌هایش باید به‌هنگام شود استفاده می‌کند (برخلاف پس‌انتشار).

Vector Form:

$$\mathbf{W}(q) = \mathbf{W}(q-1) + \alpha \mathbf{a}(q)\mathbf{p}^T(q)$$

نرخ یادگیری α مشخص می‌کند که چه تعداد تحریک و پاسخ باید با هم رخ دهند تا پیونددهی (association) انجام شود.

Training Sequence:

تأکید بر طبیعت دنباله‌ی زمانی ورودی‌ها

$$\mathbf{p}(1), \mathbf{p}(2), \ldots, \mathbf{p}(Q)$$

چیزهایی که با هم به‌طور همزمان رخ می‌دهند، به هم پیونددهی می‌شوند.
اگر بعداً یکی از آنها رخ دهد، خروجی مشابهی تولید می‌شود.

Banana Recognition Example

Initial Weights:

$$w^0 = 1, w(0) = 0$$

Training Sequence:

$$\{p^0(1) = 0, p(1) = 1\}, \{p^0(2) = 1, p(2) = 1\}, \ldots$$

$$\alpha = 1$$

$$w(q) = w(q-1) + a(q)p(q)$$

First Iteration (sight fails):

$$\begin{aligned} a(1) &= hardlim(w^0 p^0(1) + w(0)p(1) - 0.5) \\ &= hardlim(1 \cdot 0 + 0 \cdot 1 - 0.5) = 0 \quad \text{(no response)} \end{aligned}$$

$$w(1) = w(0) + a(1)p(1) = 0 + 0 \cdot 1 = 0$$

قاعده‌ی هب بدون نظارت

مثال: بازشناسی موز (۱ از ۲)

UNSUPERVISED HEBB RULE

Initial Weights:

$$w^0 = 1, w(0) = 0$$

مقداری کوچک‌تر از $-b$ به w داده می‌شود.

مقداری بزرگ‌تر از $-b$ به w_0 داده می‌شود.

Training Sequence:

شبکه ابتدا به شکل موز پاسخ می‌دهد (نه بوی موز).

$$\{p^0(1) = 0, p(1) = 1\}, \{p^0(2) = 1, p(2) = 1\}, \ldots$$

نمونه‌ها به شبکه نمایش داده می‌شوند.

$$\alpha = 1$$

$$w(q) = w(q-1) + a(q)p(q)$$

* حسگر بو همیشه درست کار می‌کند، اما حسگر شکل یکی در میان درست کار می‌کند.

First Iteration (sight fails):

بازشناسی با شکل ✗
بازشناسی با بو ✓

$$a(1) = hardlim(w^0 p^0(1) + w(0)p(1) - 0.5)$$
$$= hardlim(1 \cdot 0 + 0 \cdot 1 - 0.5) = 0 \quad \text{(no response)}$$

$$w(1) = w(0) + a(1)p(1) = 0 + 0 \cdot 1 = 0$$

Example

Second Iteration (sight works):

$$a(2) = hardlim(w^0 p^0(2) + w(1)p(2) - 0.5)$$
$$= hardlim(1 \cdot 1 + 0 \cdot 1 - 0.5) = 1 \quad \text{(banana)}$$

$$w(2) = w(1) + a(2)p(2) = 0 + 1 \cdot 1 = 1$$

Third Iteration (sight fails):

$$a(3) = hardlim(w^0 p^0(3) + w(2)p(3) - 0.5)$$
$$= hardlim(1 \cdot 0 + 1 \cdot 1 - 0.5) = 1 \quad \text{(banana)}$$

$$w(3) = w(2) + a(3)p(3) = 1 + 1 \cdot 1 = 2$$

Banana will now be detected if either sensor works.

قاعده‌ی هب بدون نظارت

مثال: بازشناسی موز (۲ از ۲)

UNSUPERVISED HEBB RULE

بازشناسی با شکل ✓
بازشناسی با بو ✓

Second Iteration (sight works):

$$a(2) = hardlim(w^0 p^0(2) + w(1)p(2) - 0.5)$$
$$= hardlim(1 \cdot 1 + 0 \cdot 1 - 0.5) = 1 \quad \text{(banana)}$$

$$w(2) = w(1) + a(2)p(2) = 0 + 1 \cdot 1 = 1$$

بازشناسی با شکل ✗
بازشناسی با بو ✓

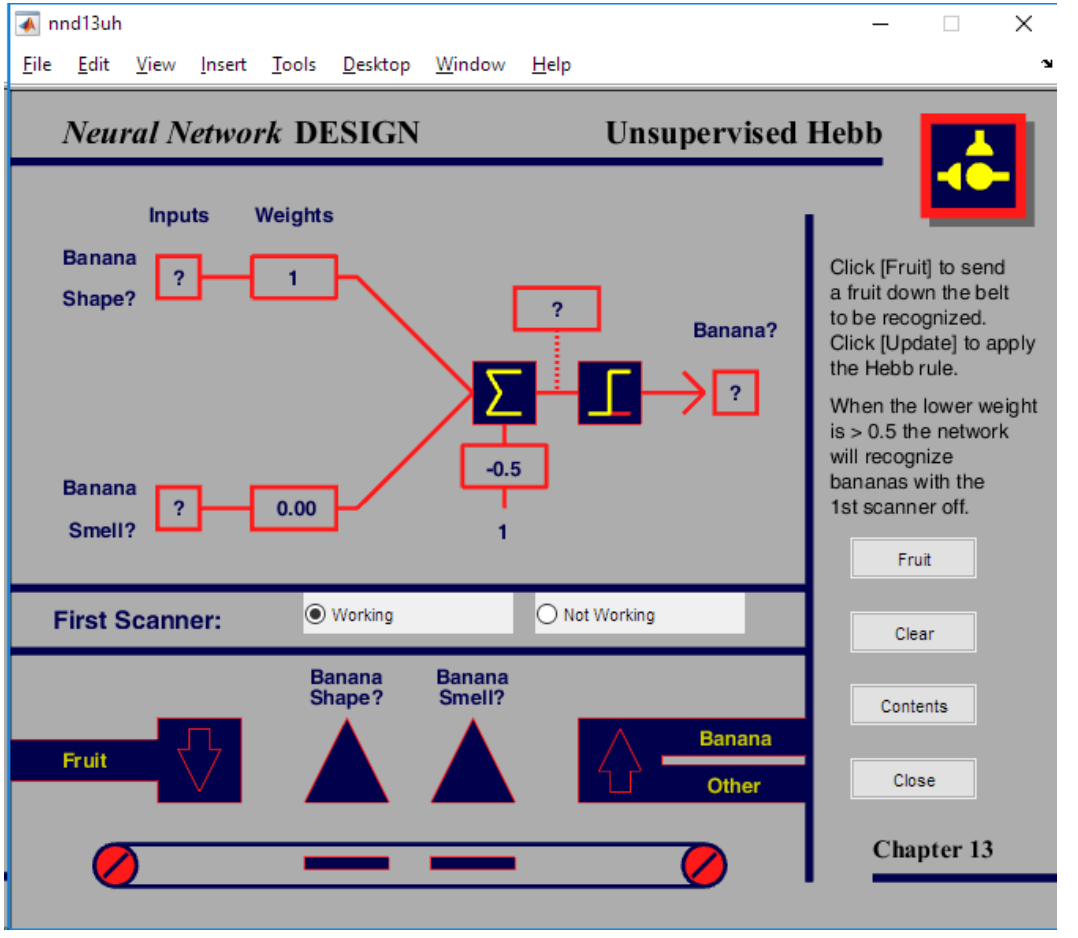
Third Iteration (sight fails):

$$a(3) = hardlim(w^0 p^0(3) + w(2)p(3) - 0.5)$$
$$= hardlim(1 \cdot 0 + 1 \cdot 1 - 0.5) = 1 \quad \text{(banana)}$$

$$w(3) = w(2) + a(3)p(3) = 1 + 1 \cdot 1 = 2$$

Banana will now be detected if either sensor works.

اکنون، هر کدام از حسگرها کار کنند، موز تشخیص داده می‌شود.


nnd13uh
File Edit View Insert Tools Desktop Window Help
Neural Network DESIGN
Unsupervised Hebb
Inputs
Weights
Banana Shape?
?
1
Banana Smell?
?
0.00
?
Banana?
?
-0.5
1
Click [Fruit] to send a fruit down the belt to be recognized. Click [Update] to apply the Hebb rule.
When the lower weight is > 0.5 the network will recognize bananas with the 1st scanner off.
First Scanner:
Working
Not Working
Fruit
Clear
Contents
Close
Banana Shape?
Banana Smell?
Fruit
Banana
Other
Chapter 13

```
>> nnd13uh
```

Problems with Hebb Rule

- Weights can become arbitrarily large
- There is no mechanism for weights to decrease

قاعده‌ی هب بدون نظارت

مشکلات

PROBLEMS WITH HEBB RULE

Weights can become arbitrarily large

با نشان دادن ورودی‌ها و به‌هنگام‌سازی وزن‌ها، وزن w بزرگ و بزرگ‌تر می‌شود.
(تفاوت با سیستم بیولوژیکی: سیناپس‌ها حد دارند.)

There is no mechanism for weights to decrease

مکانیسمی برای کاهش وزن وجود ندارد:
اگر ورودی/خروجی که به سیستم نشان داده می‌شود نویز داشته باشد،
هر وزنی بزرگ و بزرگ‌تر می‌شود (اگرچه آرام و کند) تا شبکه به هر تحریکی پاسخ دهد.

Hebb Rule with Decay

$$\mathbf{W}(q) = \mathbf{W}(q-1) + \alpha\mathbf{a}(q)\mathbf{p}^T(q) - \gamma\mathbf{W}(q-1)$$

$$\mathbf{W}(q) = (1-\gamma)\mathbf{W}(q-1) + \alpha\mathbf{a}(q)\mathbf{p}^T(q)$$

This keeps the weight matrix from growing without bound, which can be demonstrated by setting both a_i and p_j to 1:

$$w_{ij}^{max} = (1-\gamma)w_{ij}^{max} + \alpha a_i p_j$$

$$w_{ij}^{max} = (1-\gamma)w_{ij}^{max} + \alpha$$

$$w_{ij}^{max} = \frac{\alpha}{\gamma}$$

قاعده‌ی هب بدون نظارت

قاعده‌ی هب با زوال

HEBB RULE WITH DECAY

$$\mathbf{W}(q) = \mathbf{W}(q-1) + \alpha \mathbf{a}(q)\mathbf{p}^T(q) - \gamma \mathbf{W}(q-1)$$

$$\mathbf{W}(q) = (1-\gamma)\mathbf{W}(q-1) + \alpha \mathbf{a}(q)\mathbf{p}^T(q)$$

نرخ زوال

$0 < \gamma < 1$

$\gamma \to 0$: قاعده‌ی اصلی

$\gamma \to 1$: فراموش کردن سریع ورودی‌های قدیمی و تنها به‌خاطرآوردن الگوهای اخیر

This keeps the weight matrix from growing without bound, which can be demonstrated by setting both a_i and p_j to 1:

جمله‌ی دارای نرخ زوال، از رشد بدون کران ماتریس وزن‌ها جلوگیری می‌کند. $a_i = p_j = 1$

$$w_{ij}^{max} = (1-\gamma) w_{ij}^{max} + \alpha a_i p_j$$

$$w_{ij}^{max} = (1-\gamma) w_{ij}^{max} + \alpha$$

$$w_{ij}^{max} = \frac{\alpha}{\gamma}$$

Example: Banana Associator

$$\alpha = 1 \qquad \gamma = 0.1$$

First Iteration (sight fails):

$$\begin{aligned} a(1) &= hardlim(w^0 p^0(1) + w(0)p(1) - 0.5) \\ &= hardlim(1 \cdot 0 + 0 \cdot 1 - 0.5) = 0 \quad \text{(no response)} \end{aligned}$$

$$w(1) = w(0) + a(1)p(1) - 0.1w(0) = 0 + 0 \cdot 1 - 0.1(0) = 0$$

Second Iteration (sight works):

$$\begin{aligned} a(2) &= hardlim(w^0 p^0(2) + w(1)p(2) - 0.5) \\ &= hardlim(1 \cdot 1 + 0 \cdot 1 - 0.5) = 1 \quad \text{(banana)} \end{aligned}$$

$$w(2) = w(1) + a(2)p(2) - 0.1w(1) = 0 + 1 \cdot 1 - 0.1(0) = 1$$

قاعده‌ی هب بدون نظارت

قاعده‌ی هب با زوال: مثال (۱ از ۲)

EXAMPLE: BANANA ASSOCIATOR

$$\alpha = 1 \qquad \gamma = 0.1$$

First Iteration (sight fails):

$$\begin{aligned} a(1) &= hardlim(w^0 p^0(1) + w(0)p(1) - 0.5) \\ &= hardlim(1 \cdot 0 + 0 \cdot 1 - 0.5) = 0 \quad \text{(no response)} \end{aligned}$$

$$w(1) = w(0) + a(1)p(1) - 0.1w(0) = 0 + 0 \cdot 1 - 0.1(0) = 0$$

Second Iteration (sight works):

$$\begin{aligned} a(2) &= hardlim(w^0 p^0(2) + w(1)p(2) - 0.5) \\ &= hardlim(1 \cdot 1 + 0 \cdot 1 - 0.5) = 1 \quad \text{(banana)} \end{aligned}$$

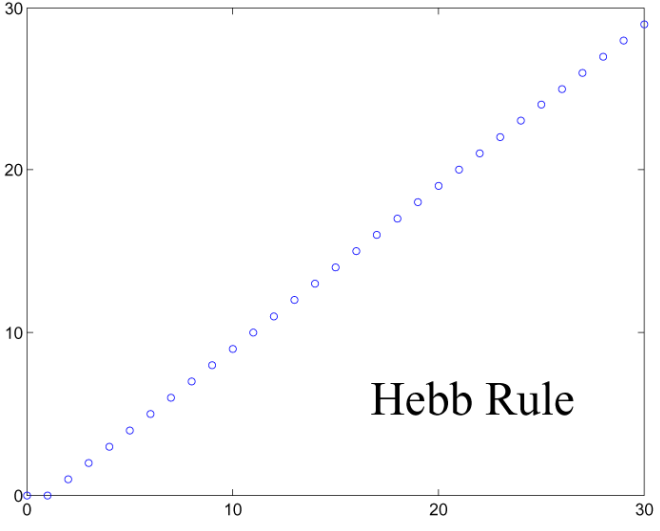
$$w(2) = w(1) + a(2)p(2) - 0.1w(1) = 0 + 1 \cdot 1 - 0.1(0) = 1$$

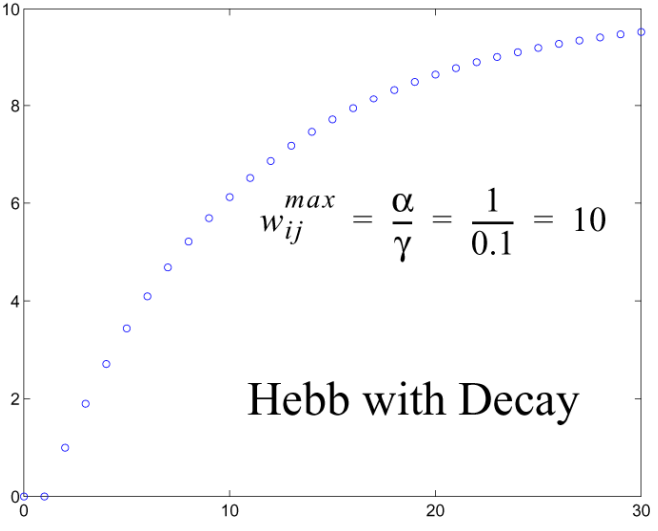
Example

Third Iteration (sight fails):

$$a(3) = hardlim(w^0 p^0(3) + w(2)p(3) - 0.5)$$
$$= hardlim(1 \cdot 0 + 1 \cdot 1 - 0.5) = 1 \quad \text{(banana)}$$

$$w(3) = w(2) + a(3)p(3) - 0.1w(3) = 1 + 1 \cdot 1 - 0.1(1) = 1.9$$





قاعده‌ی هب بدون نظارت

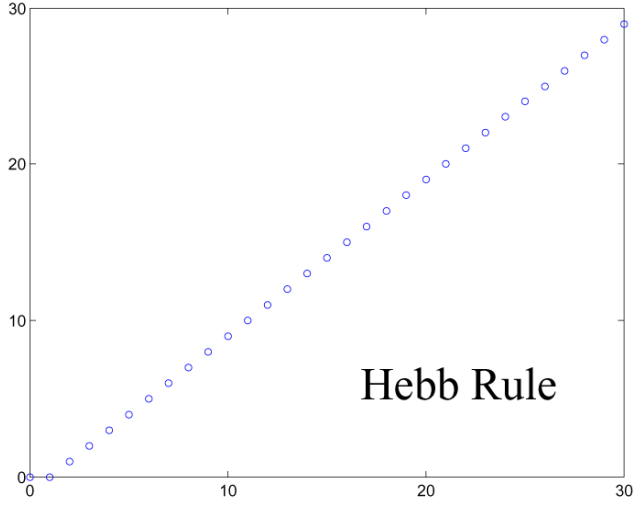
قاعده‌ی هب با زوال: مثال (۲ از ۲)

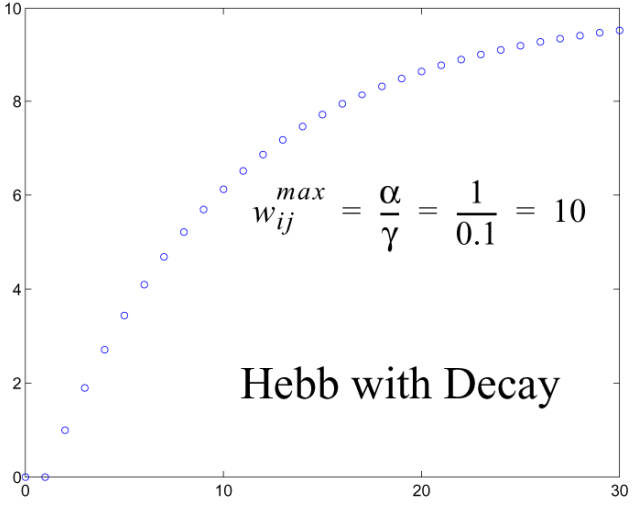
EXAMPLE: BANANA ASSOCIATOR

Third Iteration (sight fails):

$$a(3) = hardlim(w^0 p^0(3) + w(2)p(3) - 0.5)$$
$$= hardlim(1 \cdot 0 + 1 \cdot 1 - 0.5) = 1 \quad \text{(banana)}$$

$$w(3) = w(2) + a(3)p(3) - 0.1w(3) = 1 + 1 \cdot 1 - 0.1(1) = 1.9$$





Problem of Hebb with Decay

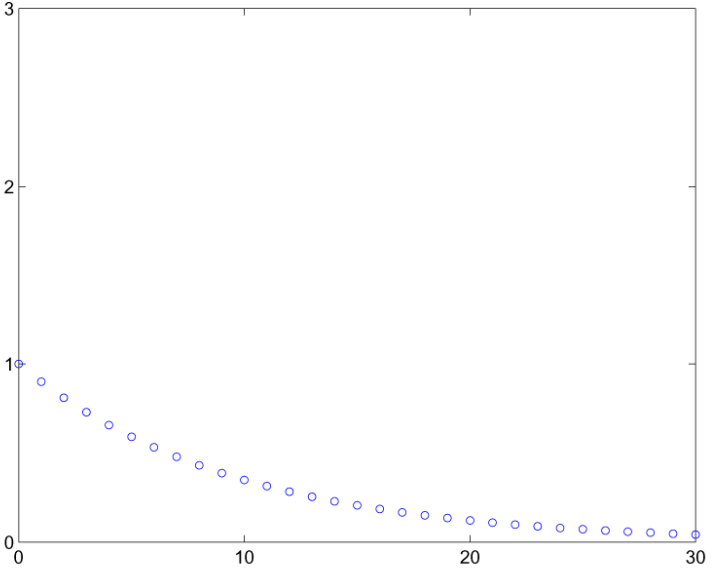
- Associations will decay away if stimuli are not occasionally presented.

If $a_i = 0$, then

$$w_{ij}(q) = (1-\gamma)w_{ij}(q-1)$$

If $\gamma = 0$, this becomes

$$w_{ij}(q) = (0.9)w_{ij}(q-1)$$



Therefore the weight decays by 10% at each iteration where there is no stimulus.

قاعده‌ی هب بدون نظارت

قاعده‌ی هب با زوال: مشکل

PROBLEM OF HEBB WITH DECAY

پیونددهی‌ها رو به زوال می‌روند اگر تحریک‌ها به‌صورت گاه‌به‌گاه ارائه نشوند.

- Associations will decay away if stimuli are not occasionally presented.

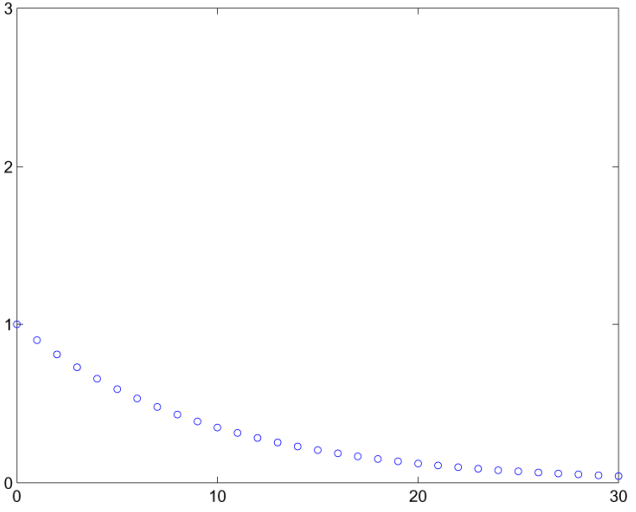
بدون تقویت، پیونددهی‌ها رو به زوال می‌روند.

If $a_i = 0$, then

$$w_{ij}(q) = (1-\gamma)w_{ij}(q-1)$$

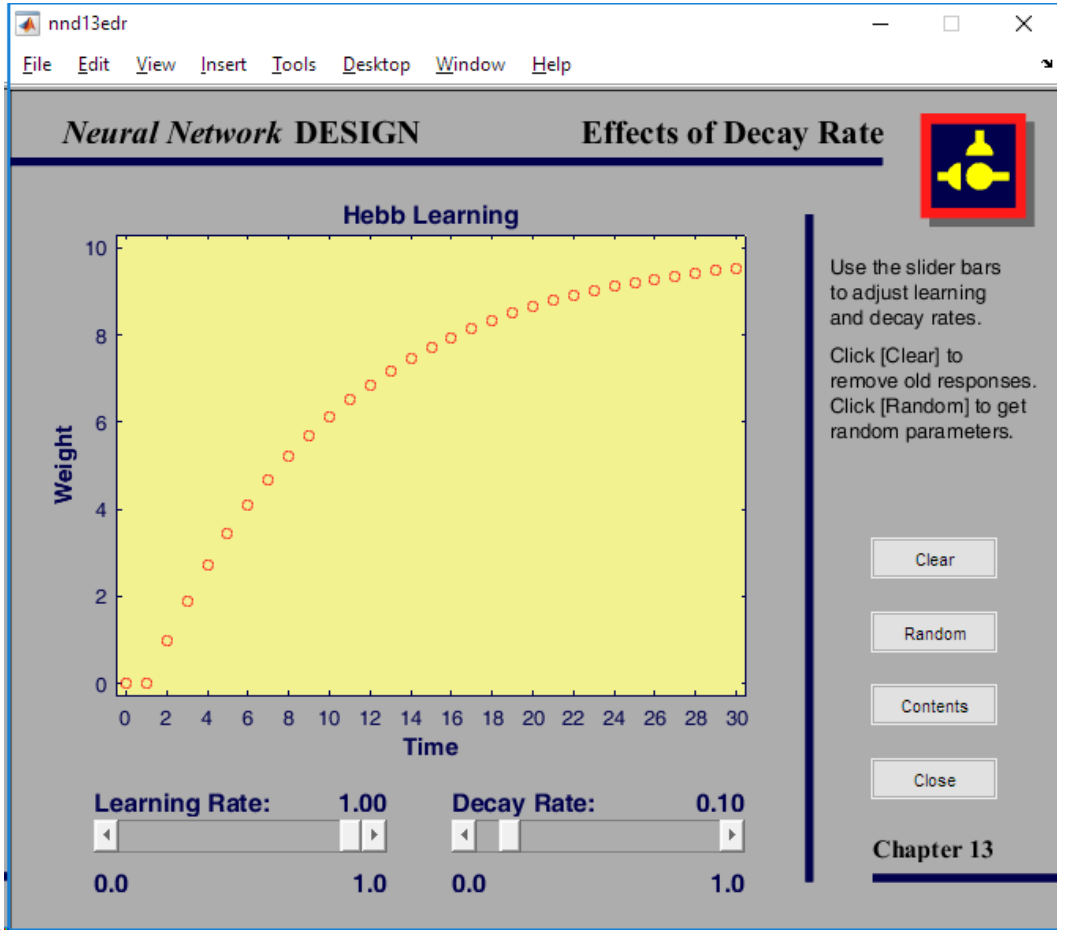
If $\gamma = 0.1$, this becomes

$$w_{ij}(q) = (0.9)w_{ij}(q-1)$$



Therefore the weight decays by 10% at each iteration where there is no stimulus.

پس وقتی هیچ تحریکی وجود ندارد، وزن در هر تکرار ۱۰ درصد کم می‌شود.


nnd13edr
File Edit View Insert Tools Desktop Window Help
Neural Network DESIGN
Effects of Decay Rate
Hebb Learning
Weight
Time
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
0 2 4 6 8 10
Use the slider bars to adjust learning and decay rates.
Click [Clear] to remove old responses. Click [Random] to get random parameters.
Clear
Random
Contents
Close
Learning Rate: 1.00
0.0 1.0
Decay Rate: 0.10
0.0 1.0
Chapter 13

```
>> nnd13edr
```

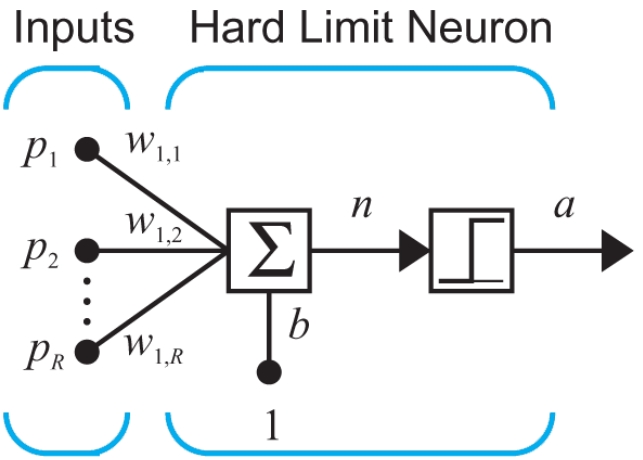
شبکه‌های عصبی مصنوعی

یادگیری پیوندی

۳

قاعده‌ی Instar

Instar (Recognition Network)

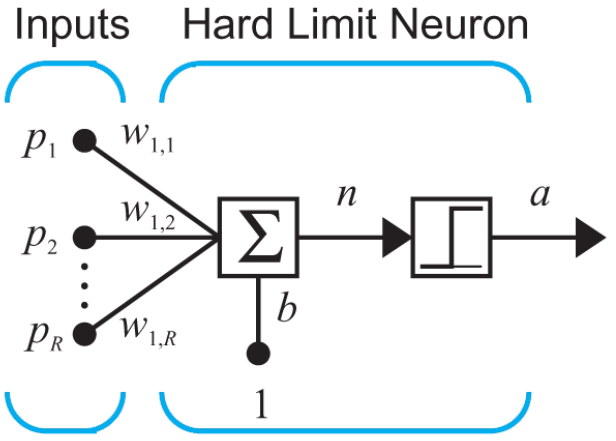


$$a = hardlim(\mathbf{W}\mathbf{p} + b)$$

Instar: شبکه‌ی بازشناسی

INSTAR (RECOGNITION NETWORK)

Instar: یک شبکه‌ی ساده که قادر به بازشناسی الگو است.
این شبکه شبیه پرسپترون است اما با تحلیلی متفاوت:
مرز تصمیم به‌طور مستقیم در نظر گرفته نمی‌شود.



$$a = hardlim(\mathbf{W}\mathbf{p} + b)$$

Instar Operation

$$a = hardlim(\mathbf{W}\mathbf{p} + b) = hardlim({}_1\mathbf{w}^T\mathbf{p} + b)$$

The instar will be active when

$${}_1\mathbf{w}^T\mathbf{p} \geq -b$$

or

$${}_1\mathbf{w}^T\mathbf{p} = \|{}_1\mathbf{w}\|\|\mathbf{p}\|\cos\theta \geq -b$$

For normalized vectors, the largest inner product occurs when the angle between the weight vector and the input vector is zero -- the input vector is equal to the weight vector.

The rows of a weight matrix represent patterns to be recognized.

Instar: شبکه‌ی بازشناسی

عملیات

INSTAR OPERATION

$$a = hardlim(\mathbf{W}\mathbf{p} + b) = hardlim({}_1\mathbf{w}^T\mathbf{p} + b)$$

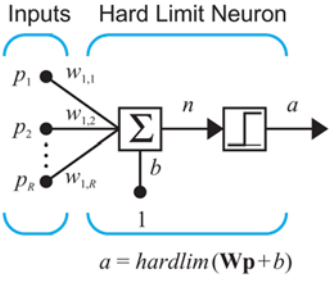
Instar فعال می‌شود وقتی که $\mathbf{p}$ به ${}_1\mathbf{w}$ نزدیک باشد (میزان نزدیکی با بایاس b مشخص می‌شود).

The instar will be active when

$${}_1\mathbf{w}^T\mathbf{p} \geq -b$$

or

$${}_1\mathbf{w}^T\mathbf{p} = \|{}_1\mathbf{w}\|\|\mathbf{p}\|\cos\theta \geq -b$$



For normalized vectors, the largest inner product occurs when the angle between the weight vector and the input vector is zero -- the input vector is equal to the weight vector.

The rows of a weight matrix represent patterns to be recognized.

سطرهای ماتریس وزن، الگوهایی که باید بازشناسی شوند را بازنمایی می‌کنند.

Vector Recognition

If we set

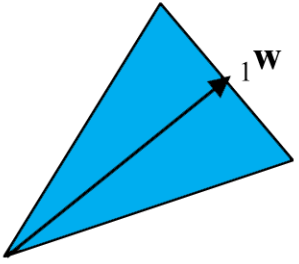
$$b = -\|_1\mathbf{w}\|\|\mathbf{p}\|$$

the instar will only be active when $\theta = 0$.

If we set

$$b > -\|_1\mathbf{w}\|\|\mathbf{p}\|$$

the instar will be active for a range of angles.



As b is increased, the more patterns there will be (over a wider range of θ) which will activate the instar.

Instar: شبکه‌ی بازشناسی

بازشناسی بردار

VECTOR RECOGNITION

If we set

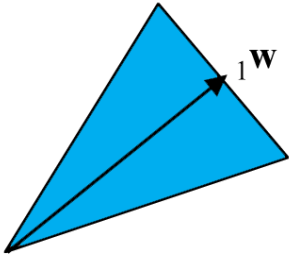
$$b = -\|_1\mathbf{w}\|\|\mathbf{p}\|$$

the instar will only be active when $\theta = 0$.

If we set

$$b > -\|_1\mathbf{w}\|\|\mathbf{p}\|$$

the instar will be active for a range of angles.



As b is increased, the more patterns there will be (over a wider range of θ) which will activate the instar.

Instar Rule

Hebb with Decay

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q)$$

Modify so that learning and forgetting will only occur when the neuron is active - Instar Rule:

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q) - \gamma a_i(q) w_{ij}(q-1)$$

or

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q)(p_j(q) - w_{ij}(q-1))$$

Vector Form:

$$_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha a_i(q)(\mathbf{p}(q) - {}_i\mathbf{w}(q-1))$$

Instar: شبکه‌ی بازشناسی

قاعده‌ی Instar

INSTAR RULE

مشکل قاعده‌ی هب با زوال این بود که لازم می‌داشت تحریک‌ها تکرار شوند، در غیر این صورت پیونددهی از دست می‌رفت.

Hebb with Decay

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q)$$

قاعده را تغییر می‌دهیم تا یادگیری و فراموشی وقتی رخ دهد که نرون فعال باشد ⇐ **قاعده‌ی Instar**

Modify so that learning and forgetting will only occur when the neuron is active - Instar Rule:

جمله‌ی زوال متناسب با خروجی $a_i(q)$ اضافه می‌شود:

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q) - \gamma a_i(q) w_{ij}(q-1)$$

or

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q)(p_j(q) - w_{ij}(q-1))$$

↙ $(\alpha = \gamma)$ یعنی: مقادیر وزن جدید با همان نرخی یاد گرفته می‌شوند که مقادیر قدیمی با آن نرخ زوال پیدا کرده‌اند.

Vector Form:

$$_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha a_i(q)(\mathbf{p}(q) - {}_i\mathbf{w}(q-1))$$

Instar Rule

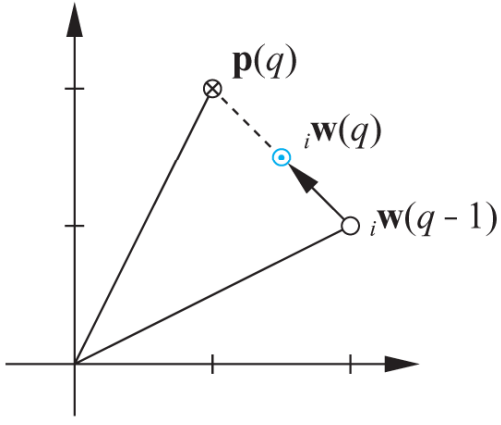
Graphical Representation

For the case where the instar is active ($a_i = 1$):

$$ {}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha(\mathbf{p}(q) - {}_i\mathbf{w}(q-1)) $$

or

$$ {}_i\mathbf{w}(q) = (1-\alpha){}_i\mathbf{w}(q-1) + \alpha\mathbf{p}(q) $$



For the case where the instar is inactive ($a_i = 0$):

$$ {}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) $$

Instar: شبکه‌ی بازشناسی

قاعده‌ی Instar: بازنمایی ترسیمی

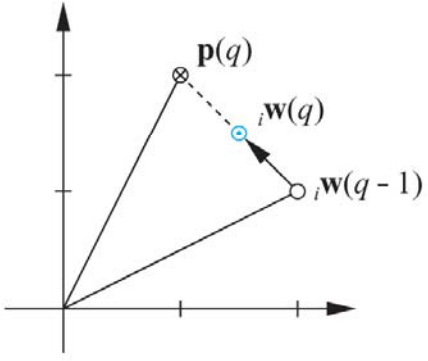
INSTAR RULE: GRAPHICAL REPRESENTATION

For the case where the instar is active ($a_i = 1$):

$$ {}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha(\mathbf{p}(q) - {}_i\mathbf{w}(q-1)) $$

or

$$ {}_i\mathbf{w}(q) = (1-\alpha){}_i\mathbf{w}(q-1) + \alpha\mathbf{p}(q) $$



کارآیی این قاعده:

وقتی شبکه فعال است،
بردار وزن به سمت بردار ورودی حرکت می‌کند:
در امتداد خط بین بردار وزن قدیمی و بردار ورودی.

فاصله‌ی حرکت وابسته به α است:
$\alpha = 0$: حرکت نداریم.
$\alpha = 1$: حداکثر حرکت: وزن جدید = بردار ورودی
$\alpha = 0.5$: حرکت به وسط بردار ورودی و وزن قدیمی

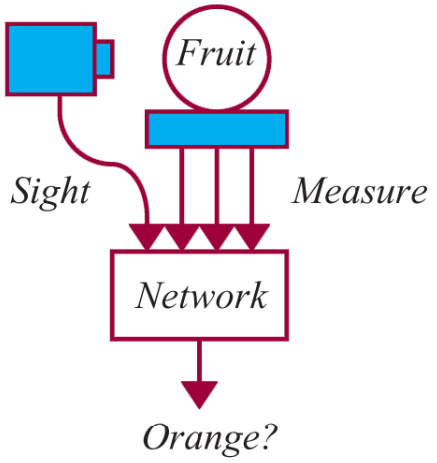
For the case where the instar is inactive ($a_i = 0$):

$$ {}_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) $$

ویژگی مفید Instar:

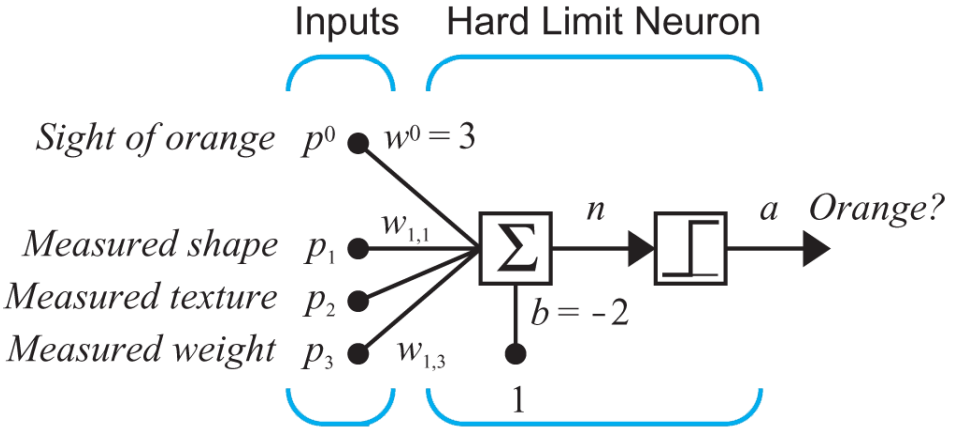
اگر بردارهای ورودی نرمال شده باشند، آن‌گاه بردار وزن $\mathbf{w}_i$ هم وقتی یک بردار خاص $\mathbf{p}$ را یاد می‌گیرد، نرمال می‌شود.
⇐ این قاعده: ۱) فراموشی را به حداقل می‌رساند، ۲) بردارهای وزن نرمال شده می‌دهد (به‌شرط بردارهای ورودی نرمال).

Example



$$p^0 = \begin{cases} 1, & \text{orange detected visually} \\ 0, & \text{orange not detected} \end{cases}$$

$$\mathbf{p} = \begin{bmatrix} shape \\ texture \\ weight \end{bmatrix}$$

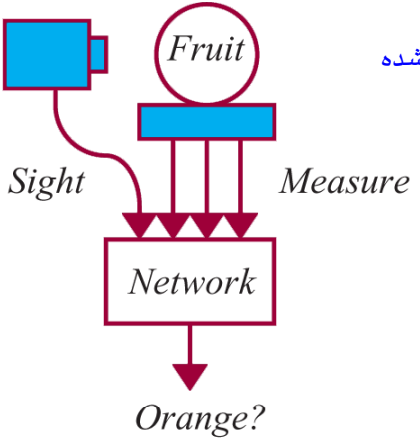


$$a = hardlim(w^0 p^0 + \mathbf{W}\mathbf{p} + b)$$

Instar: شبکه‌ی بازشناسی

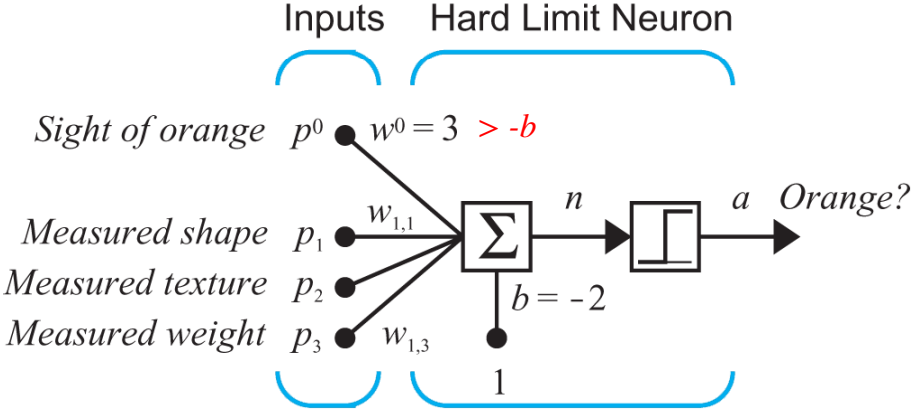
مثال (۱ از ۳)

EXAMPLE



تحریک شرطی‌نشده $p^0 = \begin{cases} 1, & \text{orange detected visually} \\ 0, & \text{orange not detected} \end{cases}$

تحریک شرطی‌شده $\mathbf{p} = \begin{bmatrix} shape \\ texture \\ weight \end{bmatrix}$ ± 1



$||\mathbf{p}|| = \sqrt{3}$ $\quad b = -2 > -||\mathbf{p}||^2 = -3$

$a = hardlim(w^0 p^0 + \mathbf{W}\mathbf{p} + b)$

Training

$$\mathbf{W}(0) = {}_1\mathbf{w}^T(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

$$\left\{ p^0(1) = 0, \mathbf{p}(1) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \right\}, \left\{ p^0(2) = 1, \mathbf{p}(2) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \right\}, \ldots$$

First Iteration (α=1):

$$a(1) = hardlim(w^0 p^0(1) + \mathbf{W}\mathbf{p}(1) - 2)$$

$$a(1) = hardlim\left(3 \cdot 0 + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - 2 \right) = 0 \quad \text{(no response)}$$

$${}_1\mathbf{w}(1) = {}_1\mathbf{w}(0) + a(1)(\mathbf{p}(1) - {}_1\mathbf{w}(0)) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 0 \left(\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Instar: شبکه‌ی بازشناسی

مثال (۲ از ۳)

EXAMPLE

$$\mathbf{W}(0) = {}_1\mathbf{w}^T(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

$$\left\{ p^0(1) = 0, \mathbf{p}(1) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \right\}, \left\{ p^0(2) = 1, \mathbf{p}(2) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \right\}, \ldots$$

First Iteration (α=1):

$$a(1) = hardlim(w^0 p^0(1) + \mathbf{W}\mathbf{p}(1) - 2)$$

$$a(1) = hardlim\left(3 \cdot 0 + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - 2 \right) = 0 \quad \text{(no response)}$$

$${}_1\mathbf{w}(1) = {}_1\mathbf{w}(0) + a(1)(\mathbf{p}(1) - {}_1\mathbf{w}(0)) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 0 \left(\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Further Training

$$a(2) = hardlim(w^0 p^0(2) + \mathbf{W}\mathbf{p}(2) - 2) = hardlim\left(3 \cdot 1 + \begin{bmatrix}0 & 0 & 0\end{bmatrix}\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 2\right) = 1 \quad \text{(orange)}$$

$${}_1\mathbf{w}(2) = {}_1\mathbf{w}(1) + a(2)(\mathbf{p}(2) - {}_1\mathbf{w}(1)) = \begin{bmatrix}0\\0\\0\end{bmatrix} + 1\left(\begin{bmatrix}1\\-1\\-1\end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix}\right) = \begin{bmatrix}1\\-1\\-1\end{bmatrix}$$

$$a(3) = hardlim(w^0 p^0(3) + \mathbf{W}\mathbf{p}(3) - 2) = hardlim\left(3 \cdot 0 + \begin{bmatrix}1 & -1 & -1\end{bmatrix}\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 2\right) = 1 \quad \text{(orange)}$$

$${}_1\mathbf{w}(3) = {}_1\mathbf{w}(2) + a(3)(\mathbf{p}(3) - {}_1\mathbf{w}(2)) = \begin{bmatrix}1\\-1\\-1\end{bmatrix} + 1\left(\begin{bmatrix}1\\-1\\-1\end{bmatrix} - \begin{bmatrix}1\\-1\\-1\end{bmatrix}\right) = \begin{bmatrix}1\\-1\\-1\end{bmatrix}$$

Orange will now be detected if either set of sensors works.

Instar: شبکه‌ی بازشناسی

مثال (۳ از ۳)

EXAMPLE

$$a(2) = hardlim(w^0 p^0(2) + \mathbf{W}\mathbf{p}(2) - 2) = hardlim\left(3 \cdot 1 + \begin{bmatrix}0 & 0 & 0\end{bmatrix}\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 2\right) = 1 \quad \text{(orange)}$$

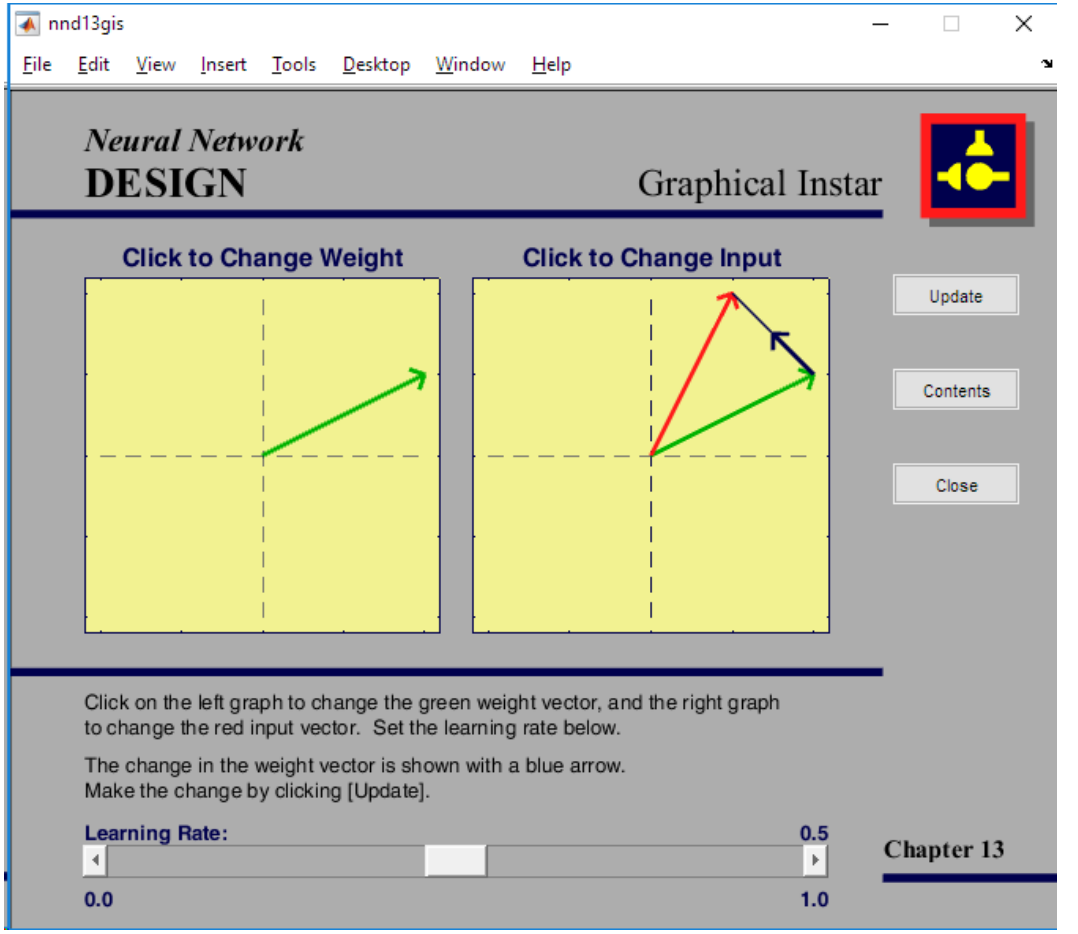
$${}_1\mathbf{w}(2) = {}_1\mathbf{w}(1) + a(2)(\mathbf{p}(2) - {}_1\mathbf{w}(1)) = \begin{bmatrix}0\\0\\0\end{bmatrix} + 1\left(\begin{bmatrix}1\\-1\\-1\end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix}\right) = \begin{bmatrix}1\\-1\\-1\end{bmatrix}$$

$$a(3) = hardlim(w^0 p^0(3) + \mathbf{W}\mathbf{p}(3) - 2) = hardlim\left(3 \cdot 0 + \begin{bmatrix}1 & -1 & -1\end{bmatrix}\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 2\right) = 1 \quad \text{(orange)}$$

$${}_1\mathbf{w}(3) = {}_1\mathbf{w}(2) + a(3)(\mathbf{p}(3) - {}_1\mathbf{w}(2)) = \begin{bmatrix}1\\-1\\-1\end{bmatrix} + 1\left(\begin{bmatrix}1\\-1\\-1\end{bmatrix} - \begin{bmatrix}1\\-1\\-1\end{bmatrix}\right) = \begin{bmatrix}1\\-1\\-1\end{bmatrix}$$

Orange will now be detected if either set of sensors works.

اکنون، هر کدام از حسگرها کار کنند، پرتقال تشخیص داده می‌شود.



MATLAB

```
>> nnd13gis
>> nnd13is
```

Kohonen Rule

$$ {}_1\mathbf{w}(q) = {}_1\mathbf{w}(q-1) + \alpha(\mathbf{p}(q) - {}_1\mathbf{w}(q-1)), \quad \text{for } i \in X(q) $$

Learning occurs when the neuron's index i is a member of the set $X(q)$. We will see in Chapter 14 that this can be used to train all neurons in a given neighborhood.

قاعده‌ی کوهونن

KOHONEN RULE

یادگیری متناسب با خروجی نرون $a_i(q)$ نیست؛ به‌جای آن از $X(q)$ استفاده می‌شود: اگر اندیس نرون i عضوی از $X(q)$ باشد.

$$_i\mathbf{w}(q) = {}_i\mathbf{w}(q-1) + \alpha(\mathbf{p}(q) - {}_i\mathbf{w}(q-1)), \quad \text{for } i \in X(q)$$

یک لایه از نرون‌ها با تابع انتقال دومقداری صفر/یک:
قاعده‌ی کوهونن با قاعده‌ی Instar معادل می‌شود، اگر:
$X(q) \equiv$ مجموعه‌ی همه‌ی اندیس‌های i که برای آنها داشته باشیم $a_i(q) = 1$

Learning occurs when the neuron's index i is a member of the set $X(q)$. We will see in Chapter 16 that this can be used to train all neurons in a given neighborhood.

قاعده‌ی کوهونن در حالت کلی می‌تواند با دیگر تعاریف $X(q)$ هم کار کند.
مناسب برای آموزش شبکه‌هایی چون SOM

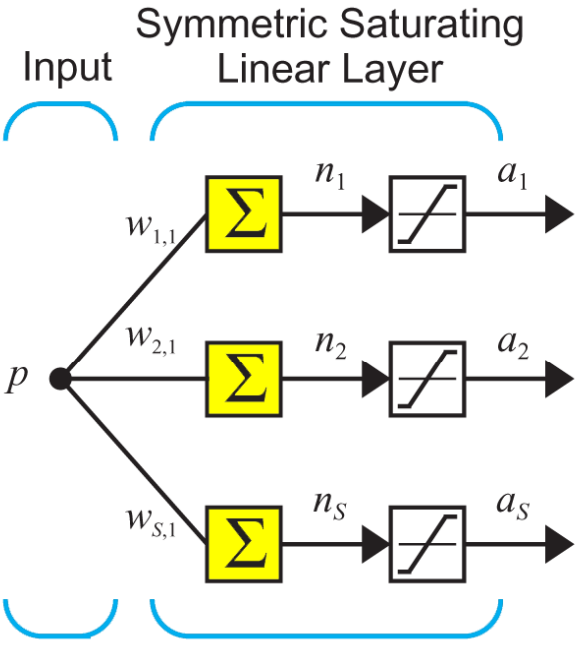
شبکه‌های عصبی مصنوعی

یادگیری پیوندی

۴

قاعده‌ی Outstar

Outstar (Recall Network)

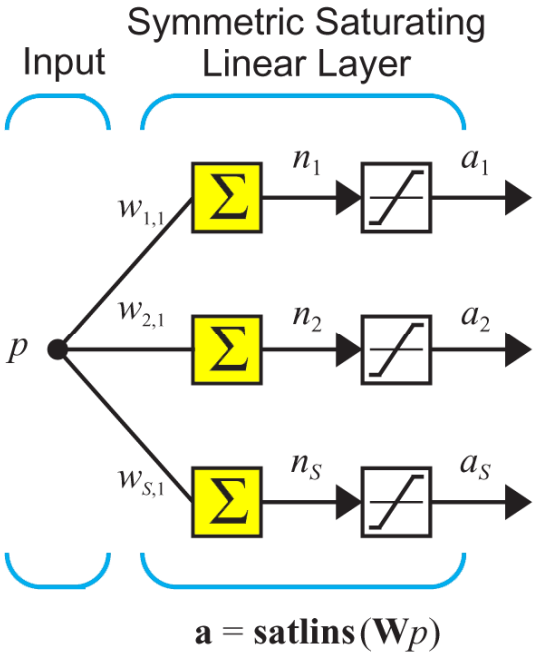


$$\mathbf{a} = \mathbf{satlins}(\mathbf{W}p)$$

Outstar: شبکه‌ی بازفراخوانی

OUTSTAR (RECALL NETWORK)

Outstar شبکه‌ای است که برای یادآوری یک بردار با مقادیر 1– و 1 به‌کار می‌رود.



Outstar: scalar input → vector output

Outstar Operation

Suppose we want the outstar to recall a certain pattern **a*** whenever the input $p = 1$ is presented to the network. Let

$$\mathbf{W} = \mathbf{a}^*$$

Then, when $p = 1$

$$\mathbf{a} = \mathbf{satlins}(\mathbf{W}p) = \mathbf{satlins}(\mathbf{a}^* \cdot 1) = \mathbf{a}^*$$

and the pattern is correctly recalled.

The columns of a weight matrix represent patterns to be recalled.

Outstar: شبکه‌ی بازفراخوانی

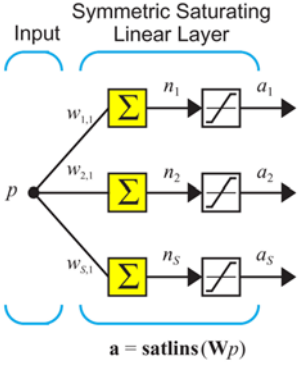
عملیات

OUTSTAR OPERATION

فرض می‌کنیم می‌خواهیم Outstar یک الگوی مشخص $\mathbf{a}^*$ را به‌خاطر بیاورد وقتی ورودی $\mathbf{p} = 1$ به شبکه نشان داده می‌شود.

Suppose we want the outstar to recall a certain pattern $\mathbf{a}^*$ whenever the input $p = 1$ is presented to the network. Let

$$\mathbf{W} = \mathbf{a}^*$$



Then, when $p = 1$

$$\mathbf{a} = \mathbf{satlins}(\mathbf{W}p) = \mathbf{satlins}(\mathbf{a}^* \cdot 1) = \mathbf{a}^*$$

با فرض اینکه عناصر $\mathbf{a}^*$ بین 1− و 1+ هستند.

and the pattern is correctly recalled.

The columns of a weight matrix represent patterns to be recalled.

ستون‌های ماتریس وزن، الگوهایی که باید به‌یادآورده شوند را بازنمایی می‌کنند.

Outstar Rule

For the instar rule we made the weight decay term of the Hebb rule proportional to the output of the network. For the outstar rule we make the weight decay term proportional to the input of the network.

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q) - \gamma p_j(q) w_{ij}(q-1)$$

If we make the decay rate γ equal to the learning rate α,

$$w_{ij}(q) = w_{ij}(q-1) + \alpha(a_i(q) - w_{ij}(q-1)) p_j(q)$$

Vector Form:

$$\mathbf{w}_j(q) = \mathbf{w}_j(q-1) + \alpha(\mathbf{a}(q) - \mathbf{w}_j(q-1)) p_j(q)$$

Outstar: شبکه‌ی بازفراخوانی

قاعده‌ی Outstar

OUTSTAR RULE

For the instar rule we made the weight decay term of the Hebb rule proportional to the output of the network. For the outstar rule we make the weight decay term proportional to the input of the network.

جمله‌ی زوال وزن متناسب با ورودی شبکه در نظر گرفته می‌شود:

$$w_{ij}(q) = w_{ij}(q-1) + \alpha a_i(q) p_j(q) - \gamma p_j(q) w_{ij}(q-1)$$

If we make the decay rate γ equal to the learning rate α,

$$w_{ij}(q) = w_{ij}(q-1) + \alpha(a_i(q) - w_{ij}(q-1)) p_j(q)$$

یعنی: مقادیر وزن جدید با همان نرخی یاد گرفته می‌شوند که مقادیر قدیمی با آن نرخ زوال پیدا کرده‌اند. ($\alpha = \gamma$) ↙

Vector Form:

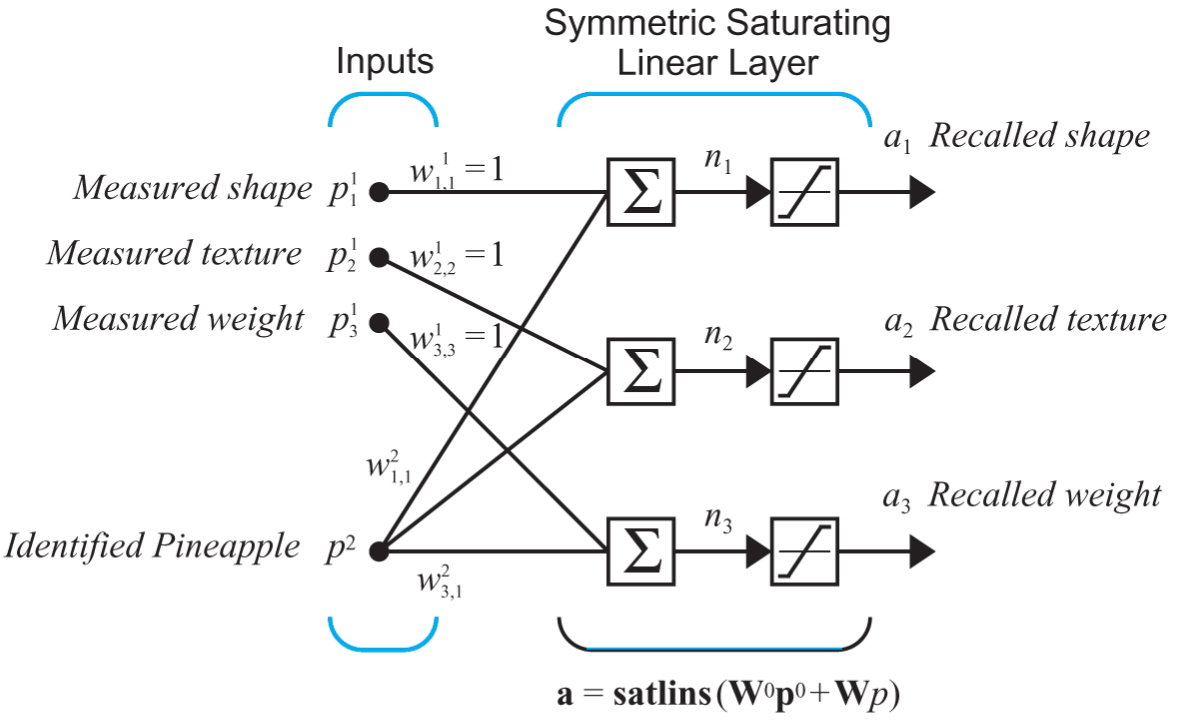
$$\mathbf{w}_j(q) = \mathbf{w}_j(q-1) + \alpha(\mathbf{a}(q) - \mathbf{w}_j(q-1)) p_j(q)$$

ستون jام $\mathbf{W}$

Outstar Rule

یادگیری زمانی اتفاق می‌افتد که p_j غیرصفر باشد (به‌جای a_i).
وقتی یادگیری صورت می‌گیرد، ستون $\mathbf{w}_j$ به‌سمت بردار خروجی حرکت می‌کند.

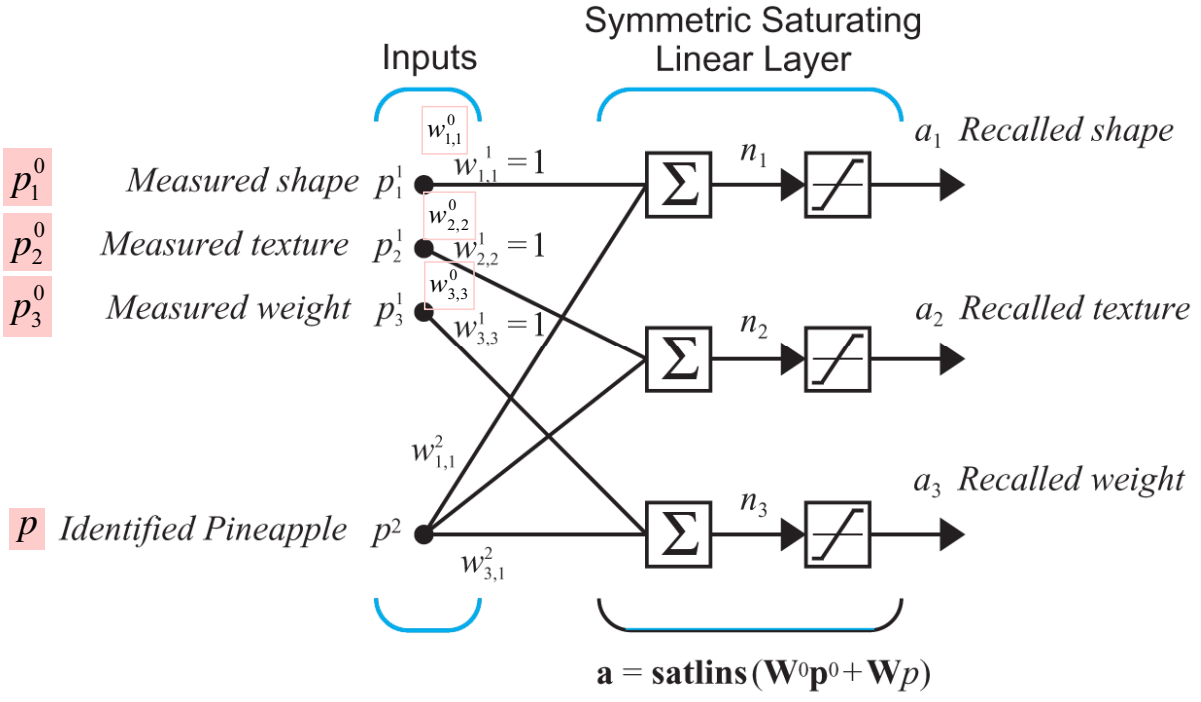
Example - Pineapple Recall



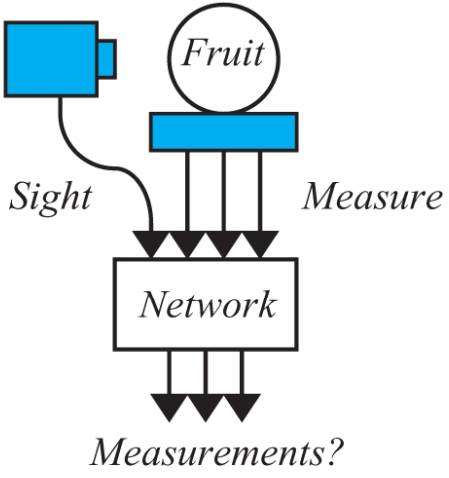
Outstar: شبکه‌ی بازفراخوانی

قاعده‌ی Outstar: مثال: بازفراخوانی آناناس (۱ از ۴)

EXAMPLE - PINEAPPLE RECALL



Definitions



$$\mathbf{a} = \mathbf{satlins}(\mathbf{W}^0\mathbf{p}^0 + \mathbf{W}p)$$

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

$$\mathbf{p}^0 = \begin{bmatrix} shape \\ texture \\ weight \end{bmatrix} \qquad \mathbf{p}^{pineapple} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$

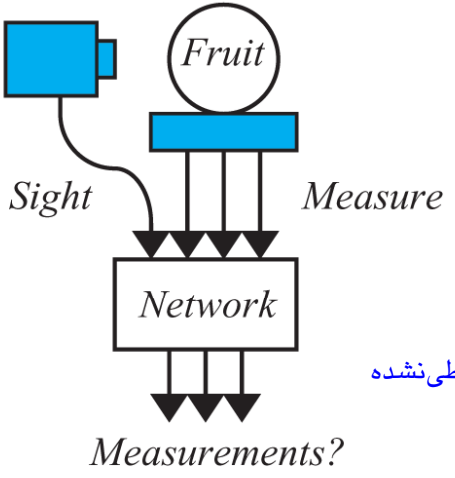
$$p = \begin{cases} 1, & \text{if a pineapple can be seen} \\ 0, & \text{otherwise} \end{cases}$$

Outstar: شبکه‌ی بازفراخوانی

قاعده‌ی Outstar: مثال: بازفراخوانی آناناس (۲ از ۴)

EXAMPLE - PINEAPPLE RECALL

$$\mathbf{a} = \mathbf{satlins}(\mathbf{W}^0\mathbf{p}^0 + \mathbf{W}p)$$



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

اندازه‌گیری
تحریک شرطی‌نشده

$$\mathbf{p}^0 = \begin{bmatrix} shape \\ texture \\ weight \end{bmatrix} \qquad \mathbf{p}^{pineapple} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$

مشاهده‌ی
تحریک شرطی

$$p = \begin{cases} 1, & \text{if a pineapple can be seen} \\ 0, & \text{otherwise} \end{cases}$$

خروجی شبکه: بازتاب اندازه‌گیری‌ها در صورتی که هر کدام از ورودی‌ها موجود بودند.

Iteration 1

$$\left\{\mathbf{p}^0(1) = \begin{bmatrix}0\\0\\0\end{bmatrix}, p(1) = 1\right\}, \left\{\mathbf{p}^0(2) = \begin{bmatrix}-1\\-1\\1\end{bmatrix}, p(2) = 1\right\}, \ldots$$

$$\alpha = 1$$

$$\mathbf{a}(1) = \mathbf{satlins}\left(\begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix}1\right) = \begin{bmatrix}0\\0\\0\end{bmatrix} \quad \text{(no response)}$$

$$\mathbf{w}_1(1) = \mathbf{w}_1(0) + (\mathbf{a}(1) - \mathbf{w}_1(0))p(1) = \begin{bmatrix}0\\0\\0\end{bmatrix} + \left(\begin{bmatrix}0\\0\\0\end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix}\right)1 = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

Outstar: شبکه‌ی بازفراخوانی

قاعده‌ی Outstar: مثال: بازفراخوانی آناناس (۳ از ۴)

EXAMPLE - PINEAPPLE RECALL

$$\left\{\mathbf{p}^0(1)=\begin{bmatrix}0\\0\\0\end{bmatrix}, p(1)=1\right\}, \left\{\mathbf{p}^0(2)=\begin{bmatrix}-1\\-1\\1\end{bmatrix}, p(2)=1\right\}, \ldots$$

$$\alpha = 1$$

اندازه‌گیری‌ها یک در میان درست هستند.

Iteration 1
تکرار ۱

$$\mathbf{a}(1) = \mathbf{satlins}\left(\begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix}1\right) = \begin{bmatrix}0\\0\\0\end{bmatrix} \quad \text{(no response)}$$

$$\mathbf{w}_1(1) = \mathbf{w}_1(0) + (\mathbf{a}(1) - \mathbf{w}_1(0))p(1) = \begin{bmatrix}0\\0\\0\end{bmatrix} + \left(\begin{bmatrix}0\\0\\0\end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix}\right)1 = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

Convergence

$$\mathbf{a}(2) = \mathbf{satlins}\left(\begin{bmatrix}-1\\-1\\1\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix}1\right) = \begin{bmatrix}-1\\-1\\1\end{bmatrix} \quad \text{(measurements given)}$$

$$\mathbf{w}_1(2) = \mathbf{w}_1(1) + (\mathbf{a}(2) - \mathbf{w}_1(1))p(2) = \begin{bmatrix}0\\0\\0\end{bmatrix} + \left(\begin{bmatrix}-1\\-1\\1\end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix}\right)1 = \begin{bmatrix}-1\\-1\\1\end{bmatrix}$$

$$\mathbf{a}(3) = \mathbf{satlins}\left(\begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}-1\\-1\\1\end{bmatrix}1\right) = \begin{bmatrix}-1\\-1\\1\end{bmatrix} \quad \text{(measurements recalled)}$$

$$\mathbf{w}_1(3) = \mathbf{w}_1(2) + (\mathbf{a}(2) - \mathbf{w}_1(2))p(2) = \begin{bmatrix}-1\\-1\\1\end{bmatrix} + \left(\begin{bmatrix}-1\\-1\\1\end{bmatrix} - \begin{bmatrix}-1\\-1\\1\end{bmatrix}\right)1 = \begin{bmatrix}-1\\-1\\1\end{bmatrix}$$

Outstar: شبکه‌ی بازفراخوانی

قاعده‌ی Outstar: مثال: بازفراخوانی آناناس (۴ از ۴): همگرایی

EXAMPLE - PINEAPPLE RECALL: CONVERGENCE

Iteration 2
تکرار ۲

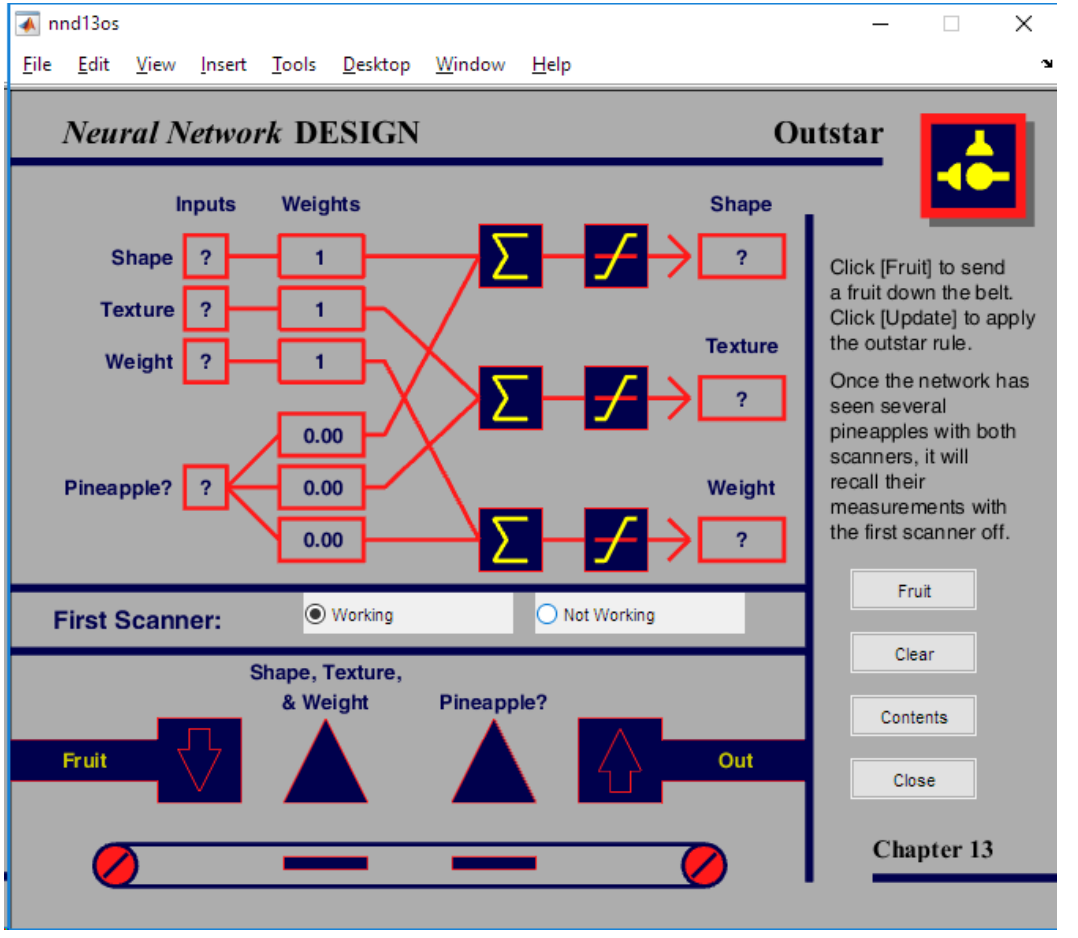
$$\mathbf{a}(2) = \mathbf{satlins}\left(\begin{bmatrix}-1\\-1\\1\end{bmatrix} + \begin{bmatrix}0\\0\\0\end{bmatrix}1\right) = \begin{bmatrix}-1\\-1\\1\end{bmatrix} \quad \text{(measurements given)}$$

$$\mathbf{w}_1(2) = \mathbf{w}_1(1) + (\mathbf{a}(2) - \mathbf{w}_1(1))p(2) = \begin{bmatrix}0\\0\\0\end{bmatrix} + \left(\begin{bmatrix}-1\\-1\\1\end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix}\right)1 = \begin{bmatrix}-1\\-1\\1\end{bmatrix}$$

Iteration 3
تکرار ۳

$$\mathbf{a}(3) = \mathbf{satlins}\left(\begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}-1\\-1\\1\end{bmatrix}1\right) = \begin{bmatrix}-1\\-1\\1\end{bmatrix} \quad \text{(measurements recalled)}$$

$$\mathbf{w}_1(3) = \mathbf{w}_1(2) + (\mathbf{a}(2) - \mathbf{w}_1(2))p(2) = \begin{bmatrix}-1\\-1\\1\end{bmatrix} + \left(\begin{bmatrix}-1\\-1\\1\end{bmatrix} - \begin{bmatrix}-1\\-1\\1\end{bmatrix}\right)1 = \begin{bmatrix}-1\\-1\\1\end{bmatrix}$$


nnd13os
File Edit View Insert Tools Desktop Window Help
Neural Network DESIGN
Outstar
Inputs
Weights
Shape
Texture
Weight
Pineapple?
Shape
Texture
Weight
Click [Fruit] to send a fruit down the belt. Click [Update] to apply the outstar rule.
Once the network has seen several pineapples with both scanners, it will recall their measurements with the first scanner off.
First Scanner:
Working
Not Working
Fruit
Clear
Contents
Close
Shape, Texture, & Weight
Pineapple?
Fruit
Out
Chapter 13

```
>> nnd13os
```

شبکه‌های ART

ART NETWORKS

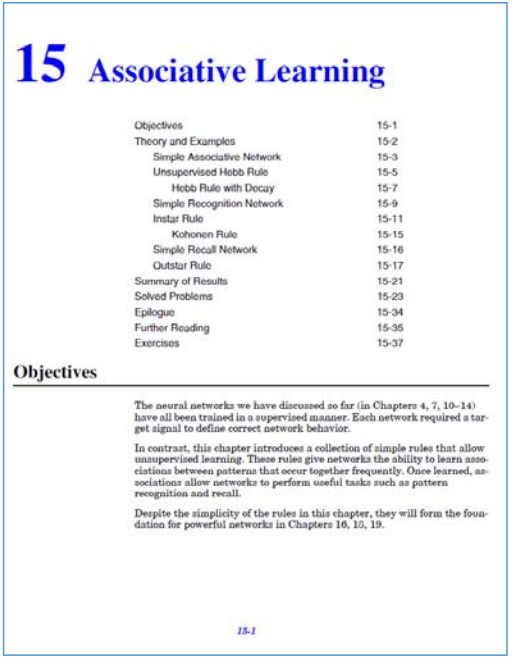
شبکه‌های ART از هر دو قاعده‌ی Instar و Outstar استفاده می‌کنند.

یادگیری پیوندی

۵

منابع

منبع اصلی



15 Associative Learning

Objectives 15-1
Theory and Examples 15-2
Simple Associative Network 15-3
Unsupervised Hebb Rule 15-5
Hebb Rule with Decay 15-7
Simple Recognition Network 15-9
Instar Rule 15-11
Kohonen Rule 15-15
Simple Recall Network 15-16
Outstar Rule 15-17
Summary of Results 15-21
Solved Problems 15-23
Epilogue 15-34
Further Reading 15-35
Exercises 15-37

Objectives

The neural networks we have discussed so far (in Chapters 4, 7, 10–14) have all been trained in a supervised manner. Each network required a target signal to define correct network behavior.

In contrast, this chapter introduces a collection of simple rules that allow unsupervised learning. These rules give networks the ability to learn associations between patterns that occur together frequently. Once learned, associations allow networks to perform useful tasks such as pattern recognition and recall.

Despite the simplicity of the rules in this chapter, they will form the foundation for powerful networks in Chapters 16, 18, 19.

15-1

Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Orlando De Jesus,
Neural Network Design,
2nd Edition, Martin Hagan, 2014.
Chapter 15

Online version can be downloaded from: http://hagan.okstate.edu/nnd.html