شبکه‌های عصبی مصنوعی

درس ۷

یادگیری بانظارت هبی

Supervised Hebbian Learning

کاظم فولادی قلعه
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

http://courses.fouladi.ir/nn

Supervised Hebbian Learning

قاعده‌ی هب

قاعده‌ی هب: یکی از اولین قوانین یادگیری در شبکه‌های عصبی (دونالد هب، 1949)

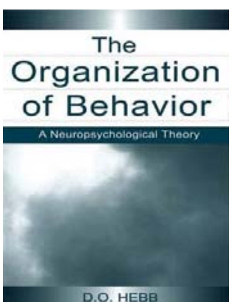
هب ابتدا می‌خواست رمان‌نویس شود و اول در ادبیات انگلیسی مدرک گرفت.
چون به درک خوبی از طبیعت انسان نیاز داشت، فروید را مطالعه کرد
و به روان‌شناسی علاقه‌مند شد.
در روان‌شناسی کارشناسی ارشد گرفت.
در 1936 از دانشگاه هاروارد دکتری گرفت.

کتاب «سازمان رفتار» (1949):
رفتار می‌تواند توسط کنش نرون‌ها توضیح داده شود.

در تقابل با نگاه اسکینر
(مکتب رفتارگرا: همبستگی بین تحریک و پاسخ + رد استفاده از فرضیه‌های فیزیولوژیک)

$\Leftarrow$ مواجهه میان فلسفه‌های «بالا به پایین» و «پایین به بالا»

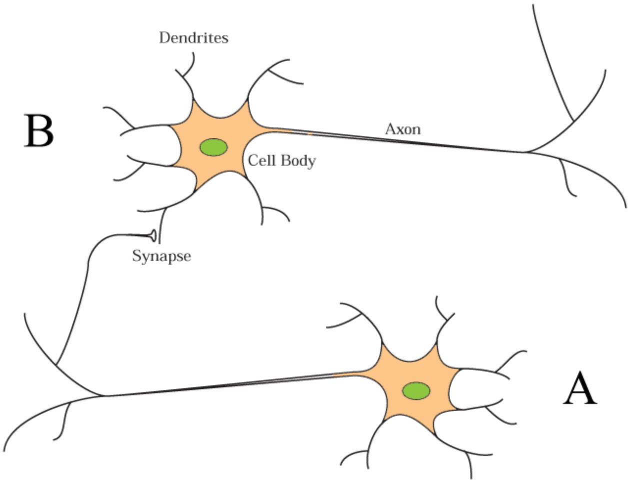

Donald O. Hebb (1904-1985)



Hebb's Postulate

"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

D. O. Hebb, 1949

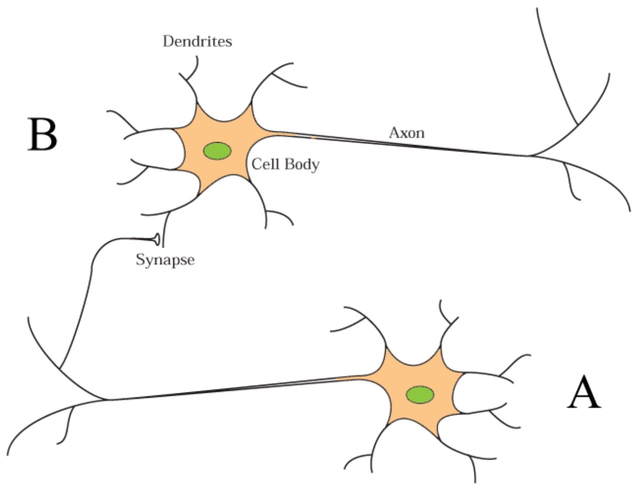


اصل موضوع هب

HEBB'S POSTULATE

وقتی یک آکسون از سلول A به‌اندازه‌ی کافی سلول B را تحریک کند
و به‌طور مکرر یا دائم در فایر کردن آن شرکت کند،
یک فرآیند رشد یا تغییر متابولیکی در یک یا هر دو سلول رخ می‌دهد،
به‌طوری‌که اثربخشی A به‌عنوان یکی از سلول‌های فایر کننده‌ی B افزایش می‌یابد.

دونالد هب، 1949



اصل موضوع هب

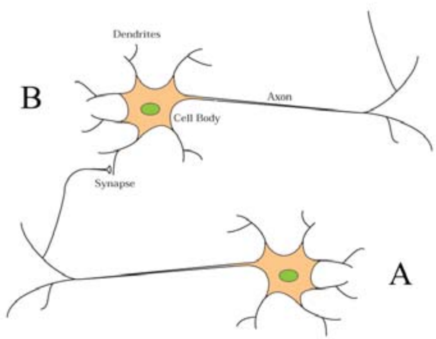
توضیح مکانیسم فیزیکی یادگیری در سطح سلولی

HEBB'S POSTULATE

اصل موضوع هب، درواقع توضیح مکانیسم فیزیکی یادگیری در سطح سلولی است.

نظریه‌ی هب، تأثیر اساسی بر پژوهش‌های جاری علوم اعصاب دارد.

پیشینه: اصل شرکت‌پذیری (ویلیام جیمز):
وقتی دو فرآیند مغزی با هم فعال شوند یا در توالی بلافاصل فعال شوند،
هر یک از این دو فرآیند میل دارد در رخداد مجدد، تحریک خود را بر دیگری منتشر کند.



یادگیری بانظارت هبی

۱

پیوندگر خطی

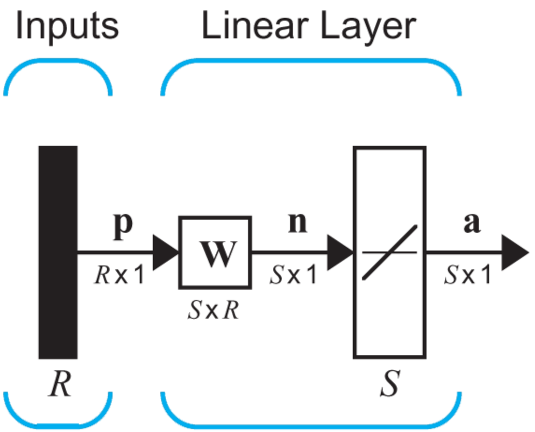
قانون یادگیری هب

قانون یادگیری هب می‌تواند در ترکیب با انواع معماری‌های شبکه‌ی عصبی استفاده شود.

برای تمرکز بر شناخت قانون یادگیری
از یک معماری بسیار ساده استفاده می‌کنیم:

پیوندگر خطی
(نمونه‌ای از یک نوع شبکه‌ی عصبی به نام: **حافظه‌ی انجمنی**)

Linear Associator



$$\mathbf{a} = \mathrm{\mathbf{purelin}}(\mathbf{Wp})$$

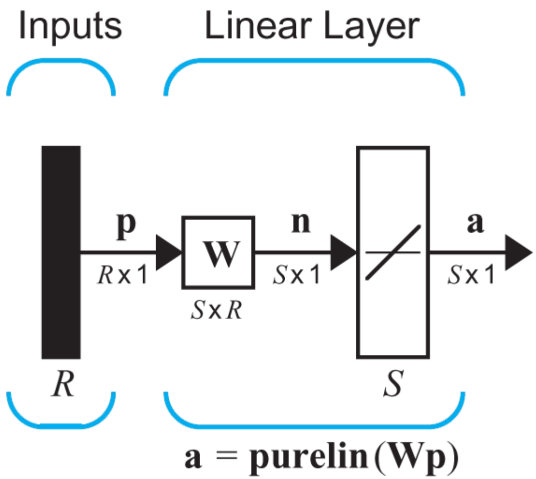
$$\mathbf{a} = \mathbf{Wp} \qquad a_i = \sum_{j=1}^{R} w_{ij} p_j$$

Training Set:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \dots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

پیوندگر خطی

LINEAR ASSOCIATOR



$$\mathbf{a} = \mathbf{W}\mathbf{p} \qquad a_i = \sum_{j=1}^{R} w_{ij} p_j$$

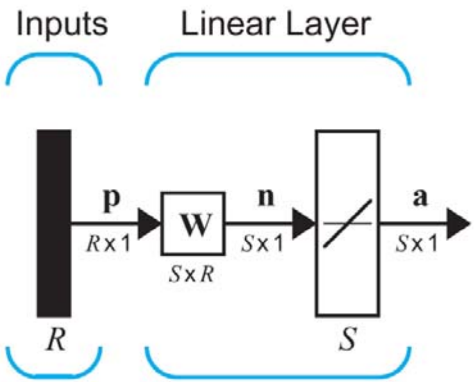
وظیفه: یادگیری Q زوج از بردارهای ورودی/خروجی پروتوتایپ

Training Set:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

پیوندگر خطی

LINEAR ASSOCIATOR



$$\mathbf{a} = \mathrm{purelin}(\mathbf{Wp})$$

تغییر اندک در ورودی $\Leftarrow$ تغییر اندک در خروجی

$$\mathbf{p} = \mathbf{p}_q \Rightarrow \mathbf{a} = \mathbf{t}_q$$

$$\mathbf{p} = \mathbf{p}_q + \delta \Rightarrow \mathbf{a} = \mathbf{t}_q + \varepsilon$$

$$\mathbf{a} = \mathbf{Wp} \qquad a_i = \sum_{j=1}^{R} w_{ij} p_j$$

Training Set:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

یادگیری بانظارت هبی

۲

قاعده‌ی هب

Hebb Rule

$$w_{ij}^{new} = w_{ij}^{old} + \alpha\, f_i(a_{iq}) g_j(p_{jq})$$

Presynaptic Signal

Postsynaptic Signal

Simplified Form:

$$w_{ij}^{new} = w_{ij}^{old} + \alpha a_{iq} p_{jq}$$

Supervised Form:

$$w_{ij}^{new} = w_{ij}^{old} + t_{iq} p_{jq}$$

Matrix Form:

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{t}_q \mathbf{p}_q^T$$

قاعده‌ی هب

HEBB RULE

قاعده‌ی هب:

اگر دو نرون در دو طرف یک سیناپس همزمان فعال شوند، قوت سیناپس افزایش می‌یابد

$\Downarrow$

اگر یک p_j مثبت یک a_i مثبت تولید کند، وزن رابطه‌ی آنها (w_{ij}) باید افزایش یابد.

$$w_{ij}^{new} = w_{ij}^{old} + \alpha\, \underbrace{f_i(a_{iq})}_{\text{Postsynaptic Signal}}\, \underbrace{g_j(p_{jq})}_{\text{Presynaptic Signal}}$$

p_{jq}	j–امین عنصر بردار $\mathbf{p}_q$
a_{iq}	i–امین عنصر بردار $\mathbf{a}_q$
α	نرخ یادگیری (مثبت)

قاعده‌ی هب

HEBB RULE

$$w_{ij}^{new} = w_{ij}^{old} + \alpha\, f_i(a_{iq})\, g_j(p_{jq})$$

Presynaptic Signal

Postsynaptic Signal

Simplified Form:

توابع f_i و g_j را خطی در نظر می‌گیریم:

$\Downarrow$

- افزایش وزن: وقتی هم‌علامت باشند.
- کاهش وزن: وقتی ناهم‌علامت باشند.

فرم ساده‌شده (قاعده‌ی بدون نظارت)

❋ به target نیازی ندارد.

$$w_{ij}^{new} = w_{ij}^{old} + \alpha a_{iq} p_{jq}$$

Supervised Form:

قرار دادن t به‌جای a:

❋ شبکه چه کاری باید انجام دهد

(به‌جای شبکه چه کاری انجام می‌دهد)

فرم با ناظر

❋ برای سادگی: $\alpha = 1$

$$w_{ij}^{new} = w_{ij}^{old} + t_{iq} p_{jq}$$

Matrix Form:

فرم با ناظر (شکل ماتریسی)

❋ برای سادگی: $\alpha = 1$

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{t}_q \mathbf{p}_q^T$$

Batch Operation

$$\mathbf{W} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \cdots + \mathbf{t}_Q\mathbf{p}_Q^T = \sum_{q=1}^{Q} \mathbf{t}_q\mathbf{p}_q^T$$

(Zero Initial Weights)

Matrix Form:

$$\mathbf{W} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \ldots & \mathbf{t}_Q \end{bmatrix} \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_Q^T \end{bmatrix} = \mathbf{T}\mathbf{P}^T$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_Q \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \ldots & \mathbf{t}_Q \end{bmatrix}$$

عملیات دسته‌ای

Batch Operation

در ابتدا ماتریس وزن صفر قرار داده می‌شود.

$$\mathbf{W} = \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \cdots + \mathbf{t}_Q\mathbf{p}_Q^T = \sum_{q=1}^{Q} \mathbf{t}_q\mathbf{p}_q^T$$

(Zero Initial Weights)

هر زوج $(\mathbf{p},\mathbf{t})$ یک بار به شبکه نشان داده می‌شود.

Matrix Form:

$$\mathbf{W} = \begin{bmatrix}\mathbf{t}_1 & \mathbf{t}_2 & \ldots & \mathbf{t}_Q\end{bmatrix}\begin{bmatrix}\mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_Q^T\end{bmatrix} = \mathbf{T}\mathbf{P}^T$$

$$\mathbf{P} = \begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_Q\end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix}\mathbf{t}_1 & \mathbf{t}_2 & \ldots & \mathbf{t}_Q\end{bmatrix}$$

Performance Analysis

$$\mathbf{a} = \mathbf{W}\mathbf{p}_k = \left(\sum_{q=1}^{Q} \mathbf{t}_q \mathbf{p}_q^T\right)\mathbf{p}_k = \sum_{q=1}^{Q} \mathbf{t}_q (\mathbf{p}_q^T \mathbf{p}_k)$$

Case I, input patterns are orthogonal.

$$\begin{aligned} (\mathbf{p}_q^T \mathbf{p}_k) &= 1 \qquad q = k \\ &= 0 \qquad q \neq k \end{aligned}$$

Therefore the network output equals the target:

$$\mathbf{a} = \mathbf{W}\mathbf{p}_k = \mathbf{t}_k$$

Case II, input patterns are normalized, but not orthogonal.

$$\mathbf{a} = \mathbf{W}\mathbf{p}_k = \mathbf{t}_k + \underbrace{\boxed{\sum_{q \neq k} \mathbf{t}_q (\mathbf{p}_q^T \mathbf{p}_k)}}_{\text{Error}}$$

تحلیل کارآیی

PERFORMANCE ANALYSIS

$$\mathbf{a} = \mathbf{W}\mathbf{p}_k = \left(\sum_{q=1}^{Q} \mathbf{t}_q \mathbf{p}_q^T\right)\mathbf{p}_k = \sum_{q=1}^{Q} \mathbf{t}_q (\mathbf{p}_q^T \mathbf{p}_k)$$

حالت یک) الگوهای ورودی متعامد هستند

Case I, input patterns are orthogonal.

$$(\mathbf{p}_q^T \mathbf{p}_k) = 1 \qquad q = k$$
$$= 0 \qquad q \neq k$$

Therefore the network output equals the target:

$$\mathbf{a} = \mathbf{W}\mathbf{p}_k = \mathbf{t}_k$$

خروجی درست برای حالت متعامد:

حالت دو) الگوهای ورودی نرمال‌شده، اما غیر متعامد هستند

Case II, input patterns are normalized, but not orthogonal.

$$\mathbf{a} = \mathbf{W}\mathbf{p}_k = \mathbf{t}_k + \boxed{\sum_{q \neq k} \mathbf{t}_q (\mathbf{p}_q^T \mathbf{p}_k)}$$

Error

میزان خطا وابسته به میزان همبستگی بین الگوهای ورودی پروتوتایپ است.

Example

Banana Apple Normalized Prototype Patterns

$$\mathbf{p}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} \quad \mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \quad \left\{ \mathbf{p}_1 = \begin{bmatrix} -0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix}, \mathbf{t}_1 = \begin{bmatrix} -1 \end{bmatrix} \right\} \quad \left\{ \mathbf{p}_2 = \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix} 1 \end{bmatrix} \right\}$$

Weight Matrix (Hebb Rule):

$$\mathbf{W} = \mathbf{T}\mathbf{P}^T = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} -0.5774 & 0.5774 & -0.5774 \\ 0.5774 & 0.5774 & -0.5774 \end{bmatrix} = \begin{bmatrix} 1.1548 & 0 & 0 \end{bmatrix}$$

Tests:

Banana $$\mathbf{W}\mathbf{p}_1 = \begin{bmatrix} 1.1548 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix} = \begin{bmatrix} -0.6668 \end{bmatrix}$$

Apple $$\mathbf{W}\mathbf{p}_2 = \begin{bmatrix} 0 & 1.1548 & 0 \end{bmatrix} \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix} = \begin{bmatrix} 0.6668 \end{bmatrix}$$

تحلیل کارآیی

مثال

PERFORMANCE ANALYSIS

Banana Apple Normalized Prototype Patterns

$$\mathbf{p}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} \quad \mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \quad \left\{ \mathbf{p}_1 = \begin{bmatrix} -0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix}, \mathbf{t}_1 = \begin{bmatrix} -1 \end{bmatrix} \right\} \quad \left\{ \mathbf{p}_2 = \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix} 1 \end{bmatrix} \right\}$$

Weight Matrix (Hebb Rule):

$$\mathbf{W} = \mathbf{T}\mathbf{P}^T = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} -0.5774 & 0.5774 & -0.5774 \\ 0.5774 & 0.5774 & -0.5774 \end{bmatrix} = \begin{bmatrix} 1.1548 & 0 & 0 \end{bmatrix}$$

Tests:

Banana $$\mathbf{W}\mathbf{p}_1 = \begin{bmatrix} 1.1548 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix} = \begin{bmatrix} -0.6668 \end{bmatrix}$$

خروجی نزدیک به تارگت است، اما خطا دارد.

Apple $$\mathbf{W}\mathbf{p}_2 = \begin{bmatrix} 0 & 1.1548 & 0 \end{bmatrix} \begin{bmatrix} 0.5774 \\ 0.5774 \\ -0.5774 \end{bmatrix} = \begin{bmatrix} 0.6668 \end{bmatrix}$$

روال‌هایی برای کاهش خطا وقتی که الگوهای پروتوتایپ متعامد نباشند وجود دارد (مانند قاعده‌ی شبه وارون)

یادگیری بانظارت هبی

۳

قاعده‌ی شبه وارون

Pseudoinverse Rule - (1)

Performance Index: $\mathbf{W}\mathbf{p}_q = \mathbf{t}_q \qquad q = 1, 2, \ldots, Q$

$$F(\mathbf{W}) = \sum_{q=1}^{Q} \|\mathbf{t}_q - \mathbf{W}\mathbf{p}_q\|^2$$

Matrix Form: $\mathbf{W}\mathbf{P} = \mathbf{T}$

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \cdots & \mathbf{t}_Q \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_Q \end{bmatrix}$$

$$F(\mathbf{W}) = \|\mathbf{T} - \mathbf{W}\mathbf{P}\|^2 = \|\mathbf{E}\|^2$$

$$\|\mathbf{E}\|^2 = \sum_i \sum_j e_{ij}^2$$

قاعده‌ی شبه وارون

(۱ از ۲)

PSEUDOINVERSE RULE

وظیفه‌ی شبکه:

Performance Index: $\mathbf{W}\mathbf{p}_q = \mathbf{t}_q \qquad q = 1, 2, \ldots, Q$

ورودی‌های متعامد با قاعده‌ی هب
$\Leftarrow$ F(**W**) = 0

$$F(\mathbf{W}) = \sum_{q=1}^{Q} \|\mathbf{t}_q - \mathbf{W}\mathbf{p}_q\|^2$$

در حالت کلی:
هدف می‌نیمم کردن F(**W**) است

$$\mathbf{W} = \arg\min_{\mathbf{W}} F(\mathbf{W})$$

Matrix Form: $\mathbf{W}\mathbf{P} = \mathbf{T}$

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 & \cdots & \mathbf{t}_Q \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_Q \end{bmatrix}$$

$$F(\mathbf{W}) = \|\mathbf{T} - \mathbf{W}\mathbf{P}\|^2 = \|\mathbf{E}\|^2$$

$$\|\mathbf{E}\|^2 = \sum_i \sum_j e_{ij}^2$$

نرم ماتریسی: مجموع مربع همه‌ی درایه‌ها

Pseudoinverse Rule - (2)

$$\mathbf{W}\mathbf{P} = \mathbf{T}$$

Minimize:

$$F(\mathbf{W}) = \|\mathbf{T} - \mathbf{W}\mathbf{P}\|^2 = \|\mathbf{E}\|^2$$

If an inverse exists for **P**, $F(\mathbf{W})$ can be made zero:

$$\mathbf{W} = \mathbf{T}\mathbf{P}^{-1}$$

When an inverse does not exist $F(\mathbf{W})$ can be minimized using the pseudoinverse:

$$\mathbf{W} = \mathbf{T}\mathbf{P}^{+}$$

$$\mathbf{P}^{+} = (\mathbf{P}^{T}\mathbf{P})^{-1}\mathbf{P}^{T}$$

قاعده‌ی شبه وارون

(۱ از ۲)

PSEUDOINVERSE RULE

$$\mathbf{WP} = \mathbf{T}$$

Minimize:

$$F(\mathbf{W}) = \|\mathbf{T} - \mathbf{WP}\|^2 = \|\mathbf{E}\|^2$$

If an inverse exists for **P**, $F(\mathbf{W})$ can be made zero:

$$\mathbf{W} = \mathbf{TP}^{-1}$$

در صورت وجود وارون **P**
(معمولاً **P** مربعی نیست و وارون دقیق ندارد!)

When an inverse does not exist $F(\mathbf{W})$ can be minimized using the pseudoinverse:

$$\mathbf{W} = \mathbf{TP}^{+}$$

$$\mathbf{P}^{+} = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T$$

به‌شرط مستقل خطی بودن ستون‌های **P**

Relationship to the Hebb Rule

Hebb Rule

$$\mathbf{W} = \mathbf{T}\mathbf{P}^T$$

Pseudoinverse Rule

$$\mathbf{W} = \mathbf{T}\mathbf{P}^+$$

$$\mathbf{P}^+ = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T$$

If the prototype patterns are orthonormal:

$$\mathbf{P}^T\mathbf{P} = \mathbf{I}$$

$$\mathbf{P}^+ = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T = \mathbf{P}^T$$

قاعده‌ی شبه وارون

ارتباط با قاعده‌ی هب

Relationship to the Hebb Rule

Hebb Rule

$$\mathbf{W} = \mathbf{T}\mathbf{P}^T$$

Pseudoinverse Rule

$$\mathbf{W} = \mathbf{T}\mathbf{P}^+$$

$$\mathbf{P}^+ = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T$$

If the prototype patterns are orthonormal:

اگر الگوهای پروتوتایپ متعامد یکه باشند:

$$\mathbf{P}^T\mathbf{P} = \mathbf{I}$$

$$\mathbf{P}^+ = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T = \mathbf{P}^T$$

Example

$$\left\{\mathbf{p}_1 = \begin{bmatrix}-1\\1\\-1\end{bmatrix}, \mathbf{t}_1 = \begin{bmatrix}-1\end{bmatrix}\right\} \quad \left\{\mathbf{p}_2 = \begin{bmatrix}1\\1\\-1\end{bmatrix}, \mathbf{t}_2 = \begin{bmatrix}1\end{bmatrix}\right\} \qquad \mathbf{W} = \mathbf{T}\mathbf{P}^{+} = \begin{bmatrix}-1 & 1\end{bmatrix}\left(\begin{bmatrix}-1 & 1\\1 & 1\\-1 & -1\end{bmatrix}\right)^{+}$$

$$\mathbf{P}^{+} = (\mathbf{P}^{T}\mathbf{P})^{-1}\mathbf{P}^{T} = \begin{bmatrix}3 & 1\\1 & 3\end{bmatrix}^{-1}\begin{bmatrix}-1 & 1 & -1\\1 & 1 & -1\end{bmatrix} = \begin{bmatrix}-0.5 & 0.25 & -0.25\\0.5 & 0.25 & -0.25\end{bmatrix}$$

$$\mathbf{W} = \mathbf{T}\mathbf{P}^{+} = \begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}-0.5 & 0.25 & -0.25\\0.5 & 0.25 & -0.25\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}$$

$$\mathbf{W}\mathbf{p}_1 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}-1\\1\\-1\end{bmatrix} = \begin{bmatrix}-1\end{bmatrix} \qquad \mathbf{W}\mathbf{p}_2 = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}1\\1\\-1\end{bmatrix} = \begin{bmatrix}1\end{bmatrix}$$

قاعده‌ی شبه وارون

ارتباط با قاعده‌ی هب : مثال

Relationship to the Hebb Rule

$$\left\{\mathbf{p}_1=\begin{bmatrix}-1\\1\\-1\end{bmatrix},\mathbf{t}_1=\begin{bmatrix}-1\end{bmatrix}\right\}\quad\left\{\mathbf{p}_2=\begin{bmatrix}1\\1\\-1\end{bmatrix},\mathbf{t}_2=\begin{bmatrix}1\end{bmatrix}\right\}\qquad \mathbf{W}=\mathbf{T}\mathbf{P}^{+}=\begin{bmatrix}-1 & 1\end{bmatrix}\left(\begin{bmatrix}-1 & 1\\1 & 1\\-1 & -1\end{bmatrix}\right)^{+}$$

$$\mathbf{P}^{+}=(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T=\begin{bmatrix}3 & 1\\1 & 3\end{bmatrix}^{-1}\begin{bmatrix}-1 & 1 & -1\\1 & 1 & -1\end{bmatrix}=\begin{bmatrix}-0.5 & 0.25 & -0.25\\0.5 & 0.25 & -0.25\end{bmatrix}$$

$$\mathbf{W}=\mathbf{T}\mathbf{P}^{+}=\begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}-0.5 & 0.25 & -0.25\\0.5 & 0.25 & -0.25\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\end{bmatrix}$$

$$\mathbf{W}\mathbf{p}_1=\begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}-1\\1\\-1\end{bmatrix}=\begin{bmatrix}-1\end{bmatrix}\qquad\qquad \mathbf{W}\mathbf{p}_2=\begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}1\\1\\-1\end{bmatrix}=\begin{bmatrix}1\end{bmatrix}$$

شبه وارون، خروجی دقیق ارائه کرده است.

شبکه‌های عصبی مصنوعی

یادگیری بانظارت هبی

۴

کاربرد

حافظه‌ی خود-پیوندی

(حافظه‌ی خود-انجمنی)

AUTOASSOCIATIVE MEMORY

در حافظه‌ی خود-پیوندی، خروجی مطلوب همان ورودی است:

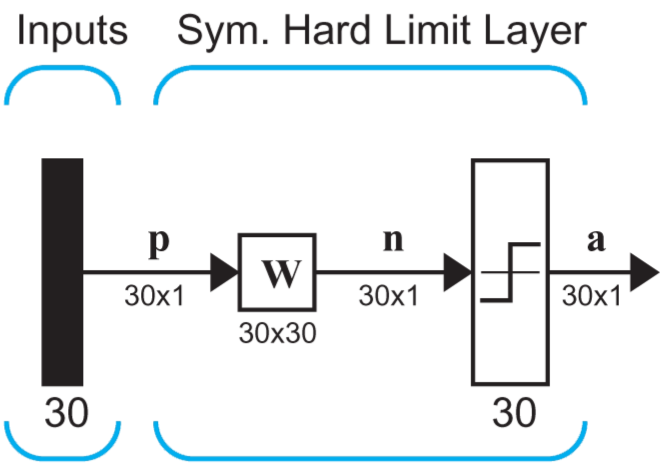
$$\mathbf{t}_q = \mathbf{p}_q$$

- شبکه، برای ذخیره‌سازی الگوها استفاده می‌شود.
- سپس این الگوها را به‌خاطر می‌آورد
  حتی وقتی که الگوهای خراب به‌عنوان ورودی به شبکه داده می‌شود.

Autoassociative Memory

$\mathbf{p}_1, \mathbf{t}_1 \quad \mathbf{p}_2, \mathbf{t}_2 \quad \mathbf{p}_3, \mathbf{t}_3$

$$\mathbf{p}_1 = \begin{bmatrix} -1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & \ldots & 1 & -1 \end{bmatrix}^T$$



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

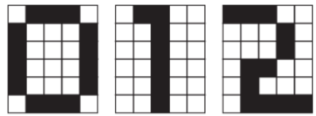
$$\mathbf{W} = \mathbf{p}_1\mathbf{p}_1^T + \mathbf{p}_2\mathbf{p}_2^T + \mathbf{p}_3\mathbf{p}_3^T$$

حافظه‌ی خود-پیوندی

(حافظه‌ی خود-انجمنی): مثال

Autoassociative Memory

الگوهایی که می‌خواهیم آنها را ذخیره کنیم:

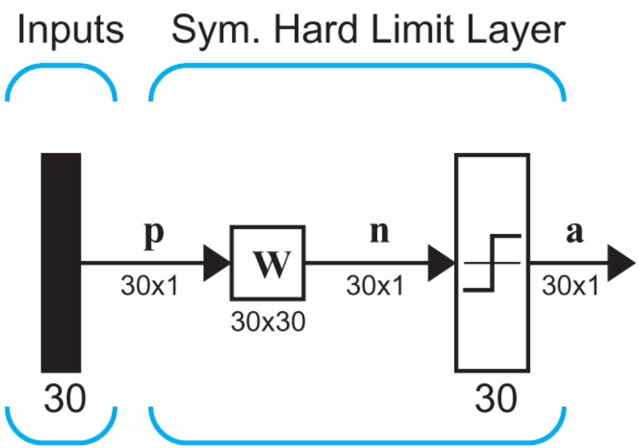


$\mathbf{p}_1,\mathbf{t}_1 \quad \mathbf{p}_2,\mathbf{t}_2 \quad \mathbf{p}_3,\mathbf{t}_3$

گرید 6×5
سفید (-1)، سیاه (1)

بسط ستون به ستون ⇓

$$\mathbf{p}_1 = \begin{bmatrix} -1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & \ldots & 1 & -1 \end{bmatrix}^T$$



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

چون تنها دو مقدار مجاز داریم.

$$\mathbf{t}_q = \mathbf{p}_q$$

$$\Downarrow$$

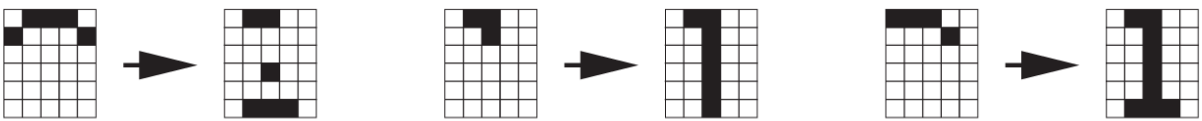
$$\mathbf{W} = \mathbf{p}_1\mathbf{p}_1^T + \mathbf{p}_2\mathbf{p}_2^T + \mathbf{p}_3\mathbf{p}_3^T$$

محاسبه‌ی وزن‌ها با قاعده‌ی هب

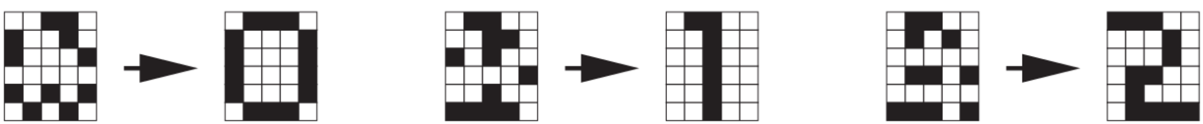
Tests

50% Occluded

67% Occluded



Noisy Patterns (7 pixels)



حافظه‌ی خود-پیوندی

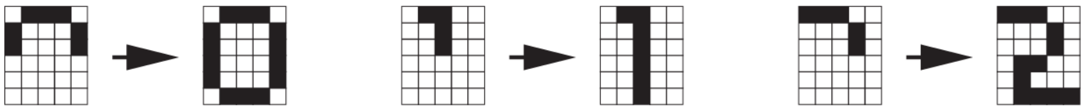
(حافظه‌ی خود-انجمنی): مثال (تست)

AUTOASSOCIATIVE MEMORY

به شبکه، ورودی‌های خراب نشان می‌دهیم:

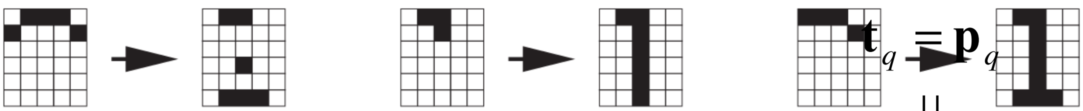
۵۰ درصد پوشیدگی (1/2)

50% Occluded



۶۷ درصد پوشیدگی (2/3)

67% Occluded

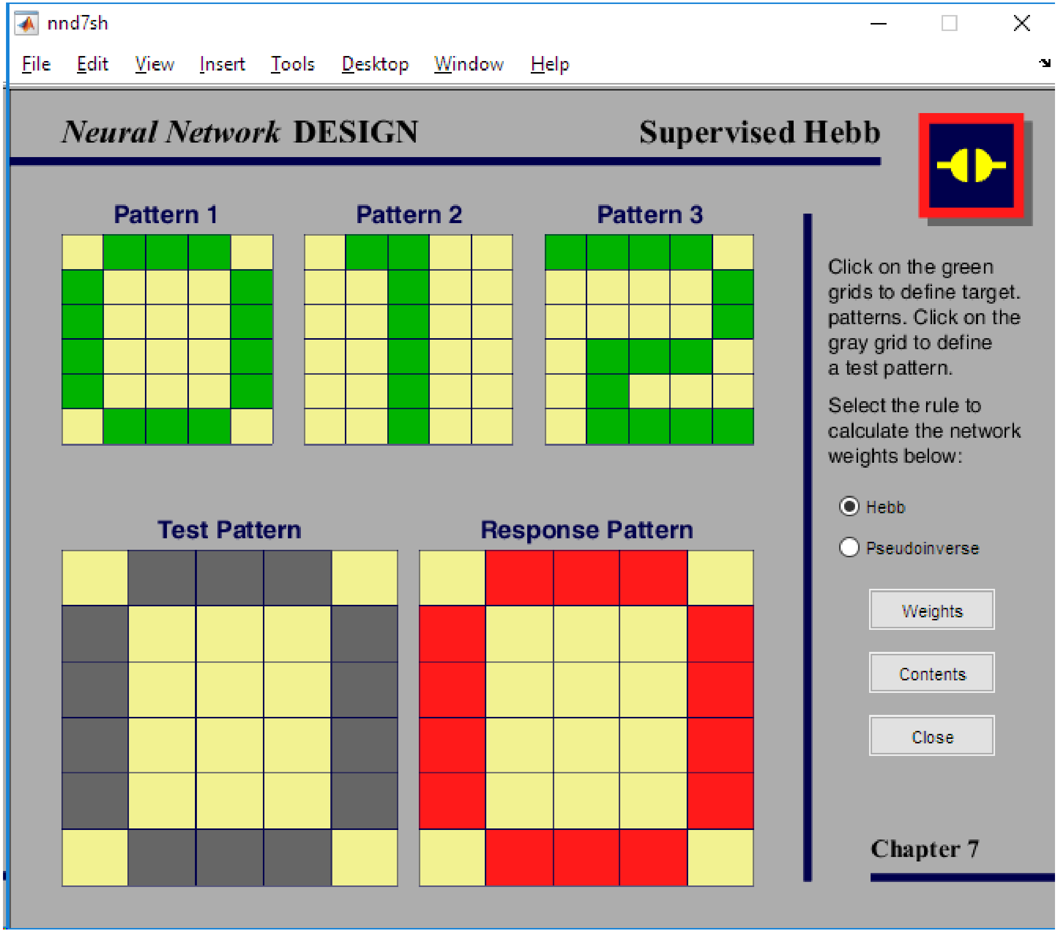


$$\mathbf{t}_q = \mathbf{p}_q$$

$$\Downarrow$$

الگوی نویزی (۷ پیکسل)

Noisy Patterns (7 pixels)



MATLAB

```
>> nnd7sh
```

شبکه‌های عصبی مصنوعی

یادگیری بانظارت هبی

۵

شکل‌های گوناگون یادگیری هبی

Variations of Hebbian Learning

Basic Rule: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{t}_q \mathbf{p}_q^T$$

Learning Rate: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \mathbf{t}_q \mathbf{p}_q^T$$

Smoothing: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \mathbf{t}_q \mathbf{p}_q^T - \gamma \mathbf{W}^{old} = (1 - \gamma)\mathbf{W}^{old} + \alpha \mathbf{t}_q \mathbf{p}_q^T$$

Delta Rule: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha(\mathbf{t}_q - \mathbf{a}_q)\mathbf{p}_q^T$$

Unsupervised: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \mathbf{a}_q \mathbf{p}_q^T$$

انواع یادگیری هبی

Variations of Hebbian Learning

Basic Rule: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{t}_q \mathbf{p}_q^T$$

Learning Rate: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \mathbf{t}_q \mathbf{p}_q^T$$

Smoothing: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \mathbf{t}_q \mathbf{p}_q^T - \gamma \mathbf{W}^{old} = (1-\gamma)\mathbf{W}^{old} + \alpha \mathbf{t}_q \mathbf{p}_q^T$$

Delta Rule: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha(\mathbf{t}_q - \mathbf{a}_q)\mathbf{p}_q^T$$

Unsupervised: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \mathbf{a}_q \mathbf{p}_q^T$$

انواع یادگیری هبی

قاعده‌ی پایه

Variations of Hebbian Learning

Basic Rule: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{t}_q \mathbf{p}_q^T$$

مشکل: ماتریس وزن می‌تواند خیلی بزرگ شود.
(وقتی که تعداد ورودی‌های آموزشی زیاد باشد)

انواع یادگیری هبی

با نرخ یادگیری

Variations of Hebbian Learning

Learning Rate: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \mathbf{t}_q \mathbf{p}_q^T$$

نرخ یادگیری α می‌تواند میزان افزایش در وزن را محدود کند

$$\alpha < 1$$

(فصل ۱۶)

انواع یادگیری هبی

با هموارسازی (یادگیری فیلتری)

VARIATIONS OF HEBBIAN LEARNING

Smoothing: $$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \mathbf{t}_q \mathbf{p}_q^T - \gamma \mathbf{W}^{old} = (1-\gamma)\mathbf{W}^{old} + \alpha \mathbf{t}_q \mathbf{p}_q^T$$

مانند یک فیلتر هموارکننده عمل می‌کند.

γ : نرخ زوال

$0 < \gamma < 1$: ورودی‌های تازه‌تر را واضح‌تر به‌خاطر می‌آورد

$\gamma \rightarrow 0$: قاعده‌ی یادگیری پایه
$\gamma \rightarrow 1$: به‌سرعت ورودی‌های قدیمی را از خاطر می‌برد.

(فصل ۱۴)

انواع یادگیری هبی

قاعده‌ی دلتا

VARIATIONS OF HEBBIAN LEARNING

Delta Rule:

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \underbrace{(\mathbf{t}_q - \mathbf{a}_q)}_{\text{خطا}} \mathbf{p}_q^T$$

قاعده‌ی دلتا:

تنظیم وزن‌ها با هدف می‌نیمم‌سازی مربعات خطا

مزیت:

امکان به‌روزکردن وزن‌ها بعد از ارائه‌ی هر تک ورودی

(شبه وارون نیاز به همه‌ی ورودی‌ها به‌صورت یکجا دارد).

❋ وفق‌یابی با محیط در حال تغییر

(فصل ۱۰)

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

انواع یادگیری هبی

بدون نظارت

Variations of Hebbian Learning

Unsupervised:

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \alpha \mathbf{a}_q \mathbf{p}_q^T$$

به‌جای تارگت $\mathbf{t}_q$

(اصل موضوع هب، به شکل بدون ناظر اشاره دارد)

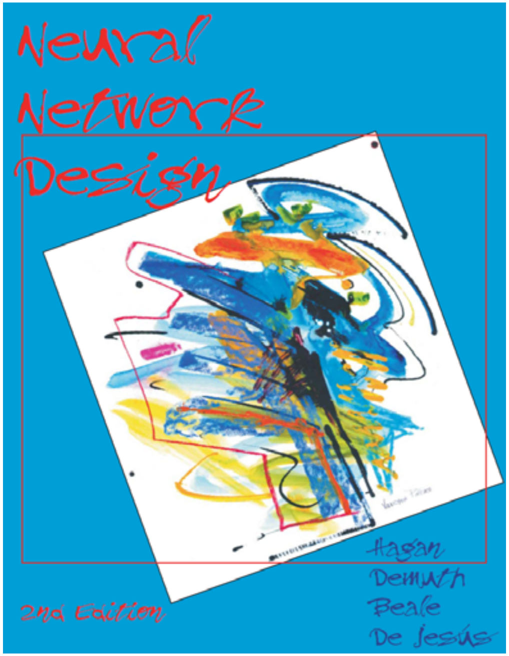
(فصل ۱۵)

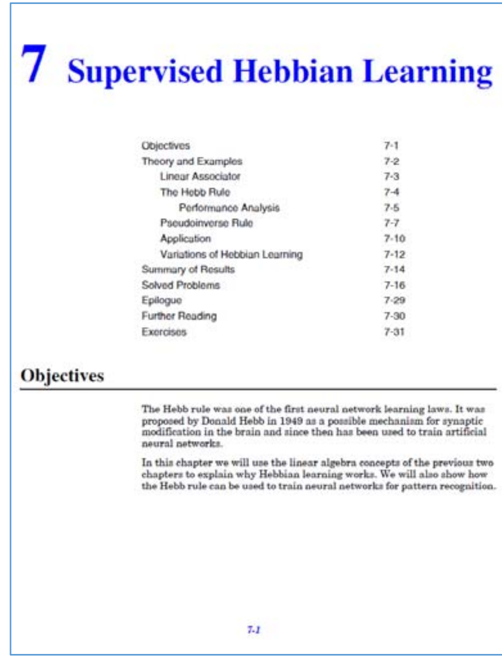
یادگیری بانظارت هبی

۶

منابع

منبع اصلی





7 Supervised Hebbian Learning

Objectives

The Hebb rule was one of the first neural network learning laws. It was proposed by Donald Hebb in 1949 as a possible mechanism for synaptic modification in the brain and since then has been used to train artificial neural networks.

In this chapter we will use the linear algebra concepts of the previous two chapters to explain why Hebbian learning works. We will also show how the Hebb rule can be used to train neural networks for pattern recognition.

Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Orlando De Jesus,
Neural Network Design,
2^{nd} Edition, Martin Hagan, 2014.
Chapter 7

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