# جزوه درس شبکه 951

پرسپترون

کارکرد: طبقه‌بندی فضای سیگنال ورودی به دو طبقه.

if $x \in F^+ \Rightarrow Y=1$ طبقه اول

if $x \in F^- \Rightarrow Y=0$ طبقه دوم

اگر $F^+$ و $F^-$ را دو کلاس از بردارهای ورودی بنامیم، کلاس را ($F^-$، $F^+$) را مشخص می‌کند.

- مرز بین طبقه‌بندی را سطح تصمیم می‌نامند که عبارت است از: $\hat{v}=0$

decision Surface $\equiv \hat{v}=0 \quad \overset{b}{w_{p+1}} \cdot \overset{1}{x_{p+1}}$

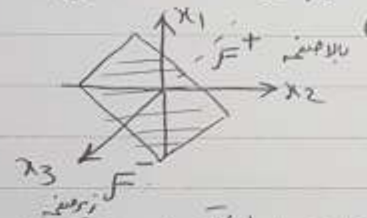
$$\Rightarrow w_1x_1+w_2x_2+\dots+w_px_p+b = W\cdot X+b=0$$

- یعنی اگر فضای ورودی واقعی ($X$) دو بعدی باشد ($X=\begin{bmatrix}x_1\\x_2\end{bmatrix}$)، صفحه‌ی تصمیم یک

$w_1x_1+w_2x_2+b=$ خط


- و اگر فضای ورودی واقعی سه بعدی باشد، صفحه‌ی تصمیم یک صفحه است (سطح است).

- و در حالت کلی یک hyperplane (فوق صفحه سطح) می‌باشد.




- اصل تعامد: در شبکه‌ی پرسپترون بردار وزن‌ها همواره بر سطح تصمیم عمود است. چه در فضای افزایش یافته ($\hat{X}$) و چه در فضای ورودی واقعی ($X$).

(با بعد $p+1$) (با بعد $p$)

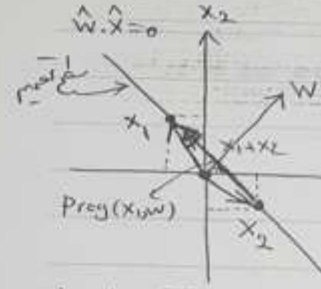
دلیل: در فضای افزایش یافته برای $\hat{X}$ داریم که $\hat{v}=\hat{W}\cdot\hat{X}=0$ در مرز تصمیم. پس $\hat{W}$ بر همه‌ی $\hat{X}$ها روی سطح تصمیم عمود است پس بر هر ترکیبی از آن‌ها نیز عمود می‌باشد یعنی بر کل سطح تصمیم عمود است.

در فضای ورودی واقعی ($p$ بعدی) نیز اگر فرض کنیم $X_1$ و $X_2$ دو بردار ورودی روی سطح تصمیم باشند با ابتدا داشته باشیم:

$$\begin{cases} W\cdot X_1+b=0 \\ W\cdot X_2+b=0 \end{cases}$$

$$\Rightarrow W\cdot(X_1-X_2)=0$$

و چون $X_1-X_2$ برداری روی سطح تصمیم است، $W$ بر این سطح، قطعاً عمود است.




- برای تعمیم موضوع فضای دو بعدی وقتی را در نظر بگیرید:
تقسیم کننده در طراحی پرسپترون، مرز تصمیم عمود بر بردار $\underline{W}$
بر مرز تصمیم عمود است و جهت را در راستای آن
در صورت با جهت مرز تصمیم بر یک نقطه آن (بایاس)
می‌توان مرز را به طور کامل مشخص کرد.

- بیان دیگر برای مرز تصمیم (سطح تصمیم) داریم که: $w \cdot x_i = -b = cte$ یعنی ضرب
داخلی $W$ در $x_i$ های روی مرز همواره ثابت است.

$|W| \cdot |Proj(x, w)| = cte$

و این تنها زمانی امکان دارد که $W$ بر مرز تصمیم عمود باشد و در نتیجه با تصویر همه‌ی نقاط روی مرز
بر بردار $W$ ثابت باشد.

مثال) در یک پرسپترون، بردار وزن‌ها عبارت است از: برآورد بایاس است همیشه

$\hat{W} = [1 \quad -1 \quad 2]$ $\underset{\omega_3 x_3}{2 \quad 1}$

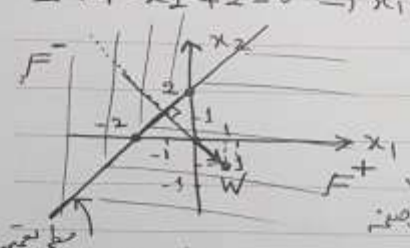
- معادله‌ی سطح تصمیم عبارت است از:

$\omega_1 x_1 + \omega_2 x_3 + b = 0$

$\Rightarrow x_1 - x_2 + 2 = 0 \Rightarrow x_1 - x_2 = -2$

و بردار $W$ در همین فضای ۲ بعدی:

$W = [1, -1]$




$Y=1$
این طرف صفحه

مثلاً مبدأ (۰،۰): $\hat{v} = x_1 - x_2 + 2 = 2 \Rightarrow y = hardlim(2) = 1$ $(F^+)$ کلاس

$\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix} : \hat{v} = 1 \Rightarrow y = 1$ ; $\begin{bmatrix} 0 \\ -2 \end{bmatrix} \Rightarrow y = 1$; چون: $\hat{v} = 4$ ✓

$\begin{bmatrix} -2 \\ 2 \end{bmatrix} : \hat{v} = -2 \Rightarrow y = 0$ (کلاس $F^-$)

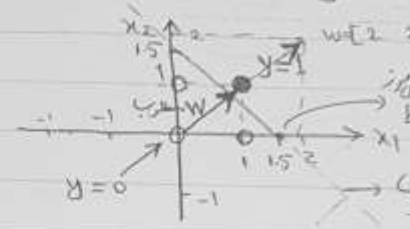
- آموزش پرسپترون ← دستی: یعنی با مشاهده و رسم مرز یا سطح تصمیم به وسیله‌ی طراح
خارجی و در نتیجه تعیین وزن‌های شبکه.
= همین تعیین وزن‌های شبکه
خودکار: صرفاً با ارائه‌ی الگوها و مثال‌های آموزشی به شبکه، خود
شبکه وزن‌ها را پیدا می‌کند.

Subject:
Year: ۹۵ Month: ۸ Date: ۱۰ ۱۷

آموزش دستی :

مثال : گیت AND منطقی دو ورودی : $Y = AND(x_1, x_2)$

| $x_1$ | $x_2$ | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |




$\begin{cases} \circ \equiv y=0 \\ \bullet \equiv y=1 \end{cases}$

— قرارداد : ○ معادل $y=0$ و ● معادل وقتی $y=1$ می باشد.

با این مرز انتخابی (سطح تقسیم انتخابی) داریم :

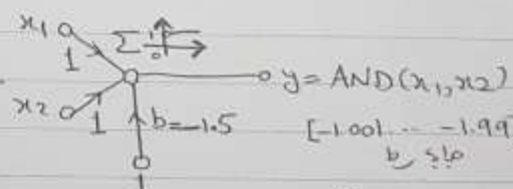
$$W = [1 \quad 1] \rightarrow \hat{W} = [1 \quad 1 \quad b]$$

$$\Rightarrow \hat{v} = x_1 + x_2 + b = 0 \Rightarrow 1.5 + 0 + b = 0 \Rightarrow b = -1.5$$

$\begin{bmatrix}1.5\\0\end{bmatrix}$ ← نقطه انتخابی

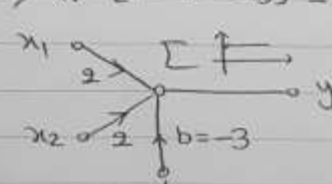
$$\Rightarrow \hat{W} = [1 \quad 1 \quad -1.5] \rightarrow$$ وزن ها مشخص شد ⇒ مسئله حل شد.

(Perc. AND.)




— با رعایت اصل تشابه می‌توان : $W = [2 \; 2]$ ; $2x_1 + 2x_2 + b = 0$

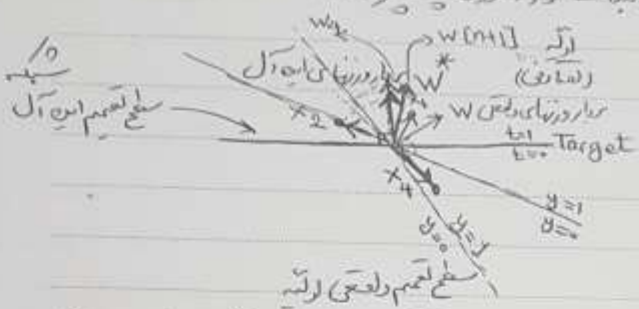
$W = \begin{bmatrix}2\\2\end{bmatrix} \Rightarrow b = -3 \Rightarrow \hat{W} = [2 \quad 2 \quad -3] = 2\hat{W}$ قبلی




— پس اندازه‌ی W اصلاً اهمیت ندارد. (از نظر ریاضیات)

— توجه : در طراحی پرسپترون (تک نورونی) صرفاً جهت W اهمیت دارد نه اندازه‌ی آن (اصل تشابه).

- نکته: در واقع پرسپترون برای هر کارکرد، به تعداد طراحان ممکن جواب وجود خواهد داشت. (یا پاسخ منحصر به فرد نیست.)
- آموزش خودکار، روش رزنبلات در آموزش پرسپترون:




سطح تقسیم واقعی (ارثیه)

- بردار وزنهای ایده آل و واقعی (و در نتیجه مرزهای تقسیم متناظر آن ها) کل فضای ورودی را به چهار ناحیه مطابق شکل رو به رو تقسیم می کنند. ($\underbrace{y=1, t=1}_{\text{ناحیه I}}$ ، $\underbrace{y=0, t=1}_{\text{ناحیه II}}$ ، $\underbrace{t=0, y=0}_{\text{ناحیه III}}$ و $\underbrace{t=0, y=1}_{\text{ناحیه IV}}$)

* فرض کنید برای الگوی آموزشی در ناحیه I قرار داشته باشیم: و می خواهیم وزن های قدم قبل (n) را براساس آن اصلاح کنیم.

$X_1: \begin{cases} t=1 & \text{مطلوب} \\ y=1 & \text{واقعی} \end{cases}$

I $X_1: \begin{cases} t=1 \\ y=1 \end{cases}$ : $\underbrace{\overset{\text{قدم بعدی}}{W[n+1]} = \overset{\text{قدم قبل}}{W[n]} + 0}$

بدیهی است در این صورت نیازی به تغییر وزن ها نمی باشد.

* اما اگر برای یک الگوی آموزشی دیگر در ناحیه II باشیم.

II $X_2: \begin{cases} t=1 \\ y=0 \end{cases}$

II $X_2: \begin{cases} t=1 \\ y=0 \end{cases}$ : $\underbrace{W[n+1] = W[n] + X_2}$ → پس W به سمت $W^*$ می رود

اما می خواهیم y=1 باشد

* و برای نواحی III و IV داریم:

III $X_3: \begin{cases} t=0 \\ y=0 \end{cases}$ : $\underbrace{W[n+1] = W[n] + 0}$

IV: $X_4: \begin{cases} t=0 \\ y=1 \end{cases}$ : $\underbrace{W[n+1] = W[n] - X_4}_{\text{تا سمت } W^* \text{ حرکت نماید}}$

Subject:
Year: ۹۵ Month: ۸ Date: ۱۰ / ۱۹

- بر این اساس الگوریتم آموزش روزنبلات را می‌توان به صورت زیر بیان کرد:

فرض کنید تعدادی الگوهای آموزشی $(X_i, t_i)$ را اختیار داریم،
$i=1,2,\ldots$

۱- انتخاب وزن‌های شبکه به صورت اختیاری یا تصادفی و اعمال آن به شبکه.

۲- الگوهای آموزشی را یک به یک و به ترتیب به شبکه اعمال کرده و خروجی‌ها آن را ($y_i$) را به ازای هر یک از آنها بدست می‌آوریم.
خروجی شبکه را با خروجی مطلوب الگوی آموزشی مقایسه کرده و بر اساس ناحیه‌ی چهارگانه‌ی حاصل از $t=0,1$ و $y=0,1$ مطابق رابطه‌ی (*)، وزن‌های شبکه را اصلاح می‌کنیم.

۳- اگر هیچ تغییری در بردار وزن‌ها در اثر اعمال کلیه‌ی الگوهای آموزشی اتفاق نیفتد، پایان الگوریتم. در غیر این صورت، برگشت به مرحله ۲ الگوریتم و تکرار آن.

## شبکه‌های عصبی - جلسه‌ی ششم - ۹۵/۸/۱۷

الگوریتم آموزش روزنبلات برای پرسپترون:

۱- شروع با وزن‌های اولیه‌ی صفر یا تصادفی

۲- اعمال الگوی آموزشی و تعیین خروجی y:

$\text{if: } y=t \Rightarrow w[n]=w[n-1]$

$\text{if: } t=1, y=0 \Rightarrow w[n]=w[n-1]+X \; ; \; X_i \Rightarrow \begin{cases} t=1 \\ y=0 \end{cases}$

$\text{if: } X_i \Rightarrow \begin{cases} t=0 \\ y=1 \end{cases} \Rightarrow w[n]=w[n-1]-X$

۴- اگر که برای همه‌ی الگوها $w[n]=w[n-1]$ ← پایان و در غیر این صورت برو به ۲.

- اگر توقف کنیم:

خطا $\delta = t-y$

$\text{if } \delta=0 \; (y=t) \Rightarrow w[n]=w[n-1]+(0)\times X$

ناحیه دوم $\text{if } \delta=t-y=1 \Rightarrow w[n]=w[n-1]+(1)\times X$

ناحیه سوم $\text{if } \delta=t-y=-1 \Rightarrow w[n]=w[n-1]+(-1)\times X$

- بنابراین قانون آموزش پرسپترون را می‌توان بطور فشرده و مختصر بشکل زیر نوشت:

$$w[n] = w[n-1] + \delta . x^T$$ برای ضرب درایه‌ای ماتریسی شبکه

که در آن: $\delta = t - y$

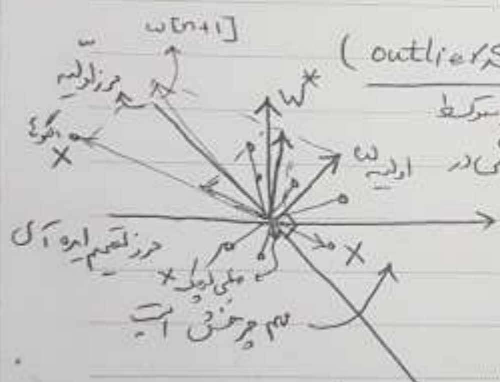
- اصلاح الگوریتم:

۱- اضافه کردن نرخ آموزش (learning rate) به قانون آموزش جهت جلوگیری از افتادن نوسانی یا ناپایدار شدن رفتار الگوریتم، یعنی:

$$w[n] = w[n-1] + \eta . \delta . x$$

که در آن نرخ آموزش عددی بین صفر و یک: $0 < \eta \le 1$ است.

اگر $\eta$ کوچک انتخاب شود احتمال ناپایداری و نوسانی شدن الگوریتم کم، ولی سرعت آن نیز کمتر خواهد بود و برعکس، با افزایش $\eta$




۲- برای حل مشکل الگوی آموزشی پرت افتاده (outliers)

یعنی الگوی آموزشی با اندازه‌ی بسیار متفاوت از سایر اندازه‌ی الگوها، پیشنهاد می‌شود که الگوهای آموزشی در ابتدا نرمالیزه شوند، یعنی:

$$x_i \to \frac{x_i}{\|x_i\|}, \quad i = 1, 2, \dots, P$$ : تعداد الگوها

$$\Rightarrow \|x_i\| = 1$$

به این ترتیب قانون آموزش پرسپترون بشکل زیر اصلاح می‌شود:

$$w[n] = w[n-1] + \eta . \delta . \frac{x}{\|x\|}$$

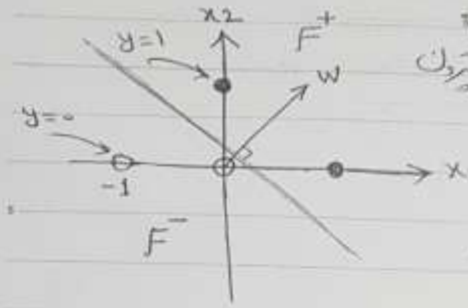
یا:

$$w[n] = w[n-1] + \frac{\eta}{\|x\|} . \delta . x$$

که در آن $\|A\|$، نرم یا اندازه‌ی بردار A را نشان می‌دهد.

یعنی عملاً نرخ آموزش ثابت نیست چراکه $\eta / \|x\|$ بستگی به x دارد. یعنی نرخ آموزش بطور وفقی خود را کنترل می‌نماید.

- یعنی عملاً نرخ آموزش به صورت وفقی، خود را با اندازه‌ی الگوها تنظیم می‌کند. برای الگوی بزرگ مقداری کوچک و برای الگوی کوچک، نرخ آموزش تقویت می‌شود.




- یک مثال ساده برای نحوه‌ی عملکرد الگوریتم:
برای الگوهای آموزشی شکل مقابل، یک پرسپترون طراحی کنید.

۱- وزن‌های اولیه شبکه را برابر بردار تمام صفر قرار میدهیم:

$$W_0 = [0 \quad 0 \quad \underbrace{0}_{\text{بایاس}}]$$

۲- الگوهای آموزشی را به ترتیب به شبکه اعمال می‌کنیم. این الگوها عبارتند از:

$$\left(\underset{X_1}{\begin{bmatrix}-1\\0\\1\end{bmatrix}}, \underset{t_1}{0}\right), \left(\underset{X_2}{\begin{bmatrix}0\\0\\1\end{bmatrix}}, \underset{t_2}{0}\right), \left(\underset{X_3}{\begin{bmatrix}0\\1\\1\end{bmatrix}}, \underset{t_3}{1}\right), \left(\underset{X_4}{\begin{bmatrix}1\\0\\1\end{bmatrix}}, \underset{t_4}{1}\right)$$

بایاس همیشه ۱

آموزش

برای الگوی $X_1$ داریم که:

$$W \cdot X_1 = 0 \Rightarrow y = \text{hardlim}(0) = 1, \text{ اما } t = 0 \Rightarrow \delta = t - y = -1$$

$$\Rightarrow W_1 = W - X_1^T = [1 \quad 0 \quad -1] = -X_1$$

$$X_2 : y = \text{hardlim}(W_1 X_2) = \text{hardlim}(-1) = 0, t_2 = 0 \Rightarrow \delta = 0$$

$$\Rightarrow W_2 = W_1 = [1 \quad 0 \quad -1]$$

$$X_3 : y = \text{hardlim}(W_2 X_3) = \text{hardlim}(-1) = 0, t_3 = 1 \Rightarrow \delta = 1$$

$$\Rightarrow W_3 = W_2 + X_3^T = [1 \quad 0 \quad -1] + [0 \quad 1 \quad 1] = [1 \quad 1 \quad 0]$$

$$X_4 : y = \text{hardlim}(W_3 X_4) = \text{hardlim}(1) = 1, t_4 = 1 \Rightarrow \delta = 0$$

$$\Rightarrow W_4 = W_3 = [1 \quad 1 \quad 0]$$

پایان دور اول آموزش

- W ها عوض شدند چراکه با [0 0 0] ورود و با [1 1 0] خروج کردیم. پس باید تکرار
دیگر را انجام شود.

* دور دوم آموزش:

$$X_1 : W_4 \cdot X_1 = -1 \Rightarrow y = 0, t_1 = 0 \Rightarrow \delta = 0 \Rightarrow W_5 = W_4$$

Subject:
Year: ۹۰ Month: ۸ Date: ۱۷

$\delta = -1$

$X_2 : W_5 \cdot X_2 = W_4 \cdot X_2 = 0 \Rightarrow y = 1, t = 0 \Rightarrow W_6 = W_5 - X_2$
$= [1\ 1\ 0] - [0\ 0\ 1]$
$= [1\ 1\ -1]$

$\Rightarrow W_6 = [1\ 1\ -1]$

$X_3 : W_6 \cdot X_3 = 0 \Rightarrow y = 1, t = 1 \Rightarrow \delta = 0 \Rightarrow w\ \checkmark$

$X_4 : W_6 \cdot X_4 = 0 \Rightarrow y = 1, t = 1 \Rightarrow w\ \checkmark$

- پایان دور دوم آموزش $\neq$ پایان آموزش.

* آغاز دور سوم آموزش: $w = [1\ 1\ -1]$ ورودی

$X_1 : w \cdot x_1 = -2 \Rightarrow y = 0, t = 0 \Rightarrow \checkmark$

$X_2 : w \cdot x_2 = -1 \Rightarrow y = 0, t = 0 \Rightarrow \checkmark$

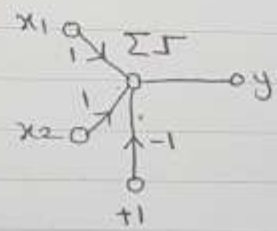
$X_3 : w \cdot x_3 = 0 \Rightarrow y = 1, t = 1 \Rightarrow \checkmark$

$X_4 = w \cdot X_4 = 0 \Rightarrow y = 1, t = 1 \Rightarrow \checkmark$

- پایان دور سوم آموزش $\equiv$ پایان کل آموزش. زیرا برای همه‌ی الگوها:

$y_i = t_i \; ; \; i = 1, 2, 3, 4.$

وزن پایانی
نهایی $w = [1\ 1\ -1]$




- برنامه‌ی کد الگوریتم آموزش پرسپترون در M:

```
function    w=percept1(x,t)
% help ...
```

`>>help percept1`

* مشخص کردن الگوهای آموزشی قبل از اجرا کردن برنامه‌ی percept1 الزامی است برای M

ستون iام $x_i = X(:,i)$

$$X = \begin{bmatrix} 1 & -1 & \cdots & -2 \\ -2 & \vdots & & \vdots \\ 0 & & & \\ 0.75 & & & 3.5 \\ -0.3 & & & -0.25 \\ p & & & \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

ورودی بایاس $\rightarrow$ 1

$x_1\ x_2\ \cdots\ x_p$

$t = [1\ 1\ 0\ \cdots\ 1]$

`>> x = [1 -1 ... -2; -2 ... ; ... ; ... 1];`

`>> t = [1 1 0 ...... ];`

...

`w = zeros(1, size(x,1))`

کمتر صفر    تعداد سطرها

Subject:

- برنامه‌ی الگوریتم پرسپترون در M :

```
function [w,j]=percept1(x,t)
% help...
% ...
n=input('number of epochs =')      % 10,000
lr=input('Learning rate =')
w=zeros(1,size(x,1));
for j=1:n
    for i=1:length(t)
        y(i)=hardlim(w*x(:,i));
        e(i)=t(i)-y(i);
        dw=dw+lr*e(i)*x(:,i)';
    end
    w=w+dw
end
end
```

شبکه و راهی که ما معرفی ... تعیین تعداد دورها معین شود

تعداد دورها    dw = zeros(1,size(x,1))

تعداد دفعات برای طرح الگوها

که سطر چندم باشد i امین ستون هر ...

ترانهاده ①

رفته میشود (۲)    حساب میشود (۱)

صفر کردن dw

تعداد کل دورهای آموزش

یک دور آموزش epoch

batch

شرط تعیین پس از پایان اعمال همه الگوها

استفاده شرط بکار ببریم → epoch

```
if e==0
    break
end
```

(منظور خروج از حلقه تا پایان آموزش (عدم تغییر wها ; e=0 )

dw==0 batch

```
>>x=[
>>t=[
[W1,R]=Percept1(x,t)
```

پایان آموزش

- انواع ممکن آموزش شبکه‌ی عصبی { - جزء به جزء یا یکی یکی incremental ، - دسته‌ای یا خوشه‌ای batch }

در حالیکه در روش incremental هر یک الگو به تنهایی باعث تغییر وزن‌های شبکه می‌شود، در روش خوشه‌ای یا دسته‌ای (batch) تا اعمال همه‌ی الگوها به شبکه وزن‌های شبکه تغییر نمی‌کند و پس از اعمال همه‌ی الگوها در حالت عادی تغییرات آن‌ها با یکدیگر ادغام شده (مثلاً جمع) و شبکه یکجا برای دور بعد آموزش تغییر می‌کند.

در حالت incr. ترتیب اعمال الگوها در الگوریتم در سرعت آن و نتیجه آن مؤثر است در حالیکه

Subject:
Year ۹۵ Month ۸ Date ۲۴ * Rosenblatt *

در روش batch، این ترتیب اهمیتی ندارد و همه ی تغییرات وزن ها به جواب یکسانی می رسند.
- مقیاس های گذشته سیگنال و مقدار انباشتن در دل incremental عمل می کنند.
شبکه ی عصبی - جلسه ی هشتم - ۹۵٫۸٫۲۴

- اثبات همگرایی آموزش پرسپترون:
* فرض ها (ساده و ساده کننده عبارات ریاضی):
فرض کنید یک بردار وزن های ایده آل ($W^*$) برای تفکیک الگوهای ورودی به دو دسته $x \in F^+$ و $x \in F^-$ وجود دارد و اندازه آن هم برابر واحد است:
به علاوه برای الگوهای آموزشی هم: $\|W^*\| = 1 = \|x_i\|_{i=1,2,\dots,p}$

و قدم اول بردار وزن صفر باشد: $W[0] = \phi ,\ \eta = 1$
و هم چنین: یعنی الگوها از مرز تقسیم پرسپترون عبور نمی کنند: $|W^* \cdot x_i| \geq \varepsilon > 0$
یعنی الگوها روی مرز تقسیم پرسپترون قرار ندارند.
- کسر زیر را در نظر بگیرید:
$$\Phi(n) = \frac{W^* \cdot W[n]}{\|W[n]\|}$$
که n در آن تعداد مراحلی که تغییر (آموزش در مرحله تغییر) بردار وزن های W.
- تحلیل صورت کسر $\Phi$:
$$\underbrace{W^* \cdot W[n+1]}_{\text{قانون آموزش}} = W^* \cdot (W[n] + \delta \cdot x)$$
$$= W^* \cdot W[n] + \delta \cdot W^* \cdot x$$
اما در صورت وقوع تغییر در بردار وزن ها (W): $\delta = \pm 1$
اگر $\delta = 1 \Rightarrow \begin{cases} t = 1 \\ y = 0 \end{cases} \xRightarrow{\text{فراز}} W^* \cdot x \geq \varepsilon > 0 \Rightarrow \delta \cdot W^* \cdot x \geq \varepsilon > 0$
$\delta \triangleq t - y$
مطلوب
واقعی
اگر $\delta = -1 \Rightarrow \begin{cases} t = 0 \\ y = 1 \end{cases} \xRightarrow{\text{تناظر}} \underset{(=\hat{v})}{W^* \cdot x} < 0 \Rightarrow \delta \cdot W^* \cdot x \geq \varepsilon > 0$
- بنابراین صورت کسر $\Phi$ در هر مرحله با هر تغییر بردار وزن ها در آموزش و ضربات پرسپترون، افزایش می یابد.
$$W^* \cdot W[1] = \underbrace{W^* \cdot W[0]}_{0} + \underbrace{\delta \cdot W^* \cdot x}_{\geq \varepsilon} \geq \varepsilon$$

$$w^* \cdot w[2] = \underbrace{w^* \cdot w[1]}_{\geqslant \varepsilon} + \underbrace{\delta \cdot w^* \cdot x}_{\geqslant \varepsilon} \geqslant 2\varepsilon$$

n بار تغییر

$$w^* \cdot w[n] = \dots\dots\dots \geqslant n\varepsilon$$

- تحلیل مخرج کسر: ②

$$\Rightarrow \|w[n+1]\|^2 = w[n+1] \cdot w[n+1] = (w[n] + \delta \cdot x) \cdot (w[n+1] + \delta \cdot x)$$

$$= \|w[n]\|^2 + \delta \cdot w[n] \cdot x + \delta \cdot x \cdot w[n] + \underbrace{\delta^2 \cdot \|x\|^2}_{1}$$

$$= \|w[n]\|^2 + 1 + \underbrace{2\delta \cdot w[n] \cdot x}$$

اما درصورت وقوع تغییر در W بالتبع $\delta = \pm 1$ است:

اگر $\delta = +1 \Rightarrow \begin{cases} t = 1 \\ y = 0 \end{cases} \Rightarrow w[n] \cdot x < 0 \Rightarrow \underbrace{2\delta \cdot w[n] \cdot x < 0}$

زیرا درحالتی که خروجی [illegible] y صفر باشد

اگر $\delta = -1 \Rightarrow \begin{cases} t = 0 \\ y = 1 \end{cases} \Rightarrow w[n] \cdot x \geqslant 0 \Rightarrow \underbrace{2\delta \cdot w[n] \cdot x \leqslant 0}$

- بنابراین درهرحال، عبارت $\underbrace{2\delta \cdot w[n] \cdot x < 0}$ (برای هر تغییر براساس الگوریتم روزنبلات).

پس:

$$\|w[1]\|^2 = 1$$

$$\|w[2]\|^2 = 1 + 1 + \overset{<0}{c} \leqslant 2$$

$$\|w[3]\|^2 = \underbrace{\|w[2]\|^2}_{\leqslant 2} + 1 + \underbrace{c'}_{\leqslant 0} \leqslant 3$$

$$\vdots$$

$$\Rightarrow \|w[n]\|^2 \leqslant n$$

- بنابراین خواهیم داشت:

$$1 \leftarrow \|w^*\| \cdot \|w[n]\| \cos(w^*, w[n])$$

$$1 \geqslant G(n) = \frac{w^* \cdot w[n]}{\|w[n]\|} \geqslant \frac{n\varepsilon}{\sqrt{n}} = \sqrt{n} \cdot \varepsilon > -1$$

اما توجه نمائید که:

$G(n) = \cos(\ ,\ ) \Longrightarrow$ کراندار است

پس n (تعداد دفعات تغییر) محدود کراندار است ـ پس اثبات شد.

بنابراین $\Longrightarrow \boxed{n \leq \frac{1}{\varepsilon^2}}$

Subject:
Year 95 Month 9 Date 12.09

- روش ب: که اگر نمونه‌های الگوهای اشکال را مطلقه زیاد باشد، n می‌تواند مشکل‌ساز گردد (محاسبات دشوار). (یعنی با n↑ سخت و پیچیده)

- جعبه ابزار پرسپترون:

andnet=newp(x,t)

- در ML همیشه بایاس‌ها را جداگانه لحاظ می‌کند. یعنی ورودی ۴ ...

- تفاوت جعبه‌ی ابزار عصبی متلب با برنامه‌ی درس:

مهم * در متلب بایاس همواره بطور جداگانه لحاظ می‌شود. بعنوان مثال در ارائه‌ی الگوی آموزشی در متلب، نباید بایاس را به ماتریس x اضافه کنیم.

view(andnet)

- با ۳ نورون خروجی به عنوان مثال:

X=[1 - t - xp]

تعداد نورون خروجی

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

3×p

newp(x,T) → از روی این ماتریس‌ها شبکه را مشخص می‌کند

پس: x=[0 0, 0 1, 1 0, 1 1] → دیگر بایاس نمی‌خواهد.

t= ... t هم

andnet=newp(x,t)

view(andnet) آموزش برای مثال

andnet=train(andnet,x,t)

آموزش

تغییر پارامترهای اولیه andnet.trainparam.epochs=12

" " .lr=1;

→ دوباره اجرا کنیم

برای دیدن وزن‌ها andnet.iw{1,1}

نام شبکه=newp(x,t) — دستور تولید شبکه پرسپترون

=train(نام شبکه, x,t) — " آموزش

نام شبکه.trainparam.epoch... — " تغییرات پارامترهای آموزش

نام شبکه.iw{1,1} → (لایه ورودی) — " دیدن همه وزن‌ها

نام شبکه.Lw{2,2}

PAPCO

Subject:
Year 95 Month ۸ Date ۲۴ /۰۷

- مشاهده‌ی بایاس:

andnet.b(1)

پس وزن‌ها با بایاس نوشته داده میشود. حالا هاشا اجمال و الگوریتم حالا هانه هر یک جداگانه:

- پارامتر lr در این دستور، بی‌اثر است و هیچ تأثیری تغییر آن نخواهد داشت.

- در شبکه‌ی آموزش پرسپترون فقط بصورت inc. انجام میشود.

- " " lr جز پارامترهای آموزش تطبیقی نیست.

* فرم کامل دستور: ماتریس ورودی‌ها ماتریس تارگت‌ها

اسم شبکه = newp(x, t, 'TF', 'LF')

برای نرمالیزه کردن شبکه learnp, learnpn ← توابع فعالیت ← مثلاً پرش کنت hardlim, hardlims

← با ورودی‌های نرمالیزه

andnet=newp(x,t,'hardlim','learnp'); ← شکل کامل دستور

- پس learnpn مطلب با ورودی‌های نرمالیزه شده است:

- اما اگر learnpn ← تعداد epoch‌ها کم میشود

- با learnpn (ورودی‌های نرمالیزه شده) ← حتی بهتر است. سراغ‌تر

- محدودیت پرسپترون: مرز تفکیک خطی: فقط قادر به تفکیک مجموعه‌ی بردارهایی است که با مرزهای سطح خطی قابل جداسازی اند.

- مثلاً XOR

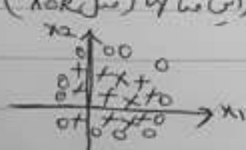
- مشاهده وزن‌ها برای یک ورودی خاص:

p=[1 0]'

XORnet(p)

XORnet(X) ← ماتریس ورودی


- اما به یک خط، نمی‌توان مرز تشخیص داد ← هیچ مرز تفکیک خطی کامل نیست هیچ مرز تفکیک خطی که در جداسازی کلاس‌های ورودی (در این مثال‌ها) (مثل XOR) نیست بنابراین پرسپترون شکست میخورد.




- اگر خطی به شکل تفکیه آن را پیدا خواهد کرد...

تمرین) طراحی پرسپترون برای: OR (دو ورودی) – OR (سه ورودی) – NAND (دو و سه ورودی) – و یک شبکه پرسپترونی با دو خروجی OR NAND

هم زمانی هم با کد الگوریتم آموزش