



# شبکه های عصبی

استاد: محمد باقر منهج



## فصل ۸

# شبکه های عصبی چند لایه پیشخور و یادگیری پس انتشار خطا



## • فصل هشتم

- مقدمه
- شبکه پرسپترون چند لایه
- حل چند مسئله
- الگوریتم پس انتشار خطا (BP)
- تقریب توابع
- کاربردهای شبکه عصبی چند لایه



## مقدمه

- در شبکه های MLP

- هر نرون دارای تابع تحریک غیرخطی است.

- یادگیری پس انتشار خطا (BP):

- برای آموزش شبکه های عصبی چند لایه پرسپترون (MLP)

- **تعمیمی از الگوریتم LMS**، لذا تقریبی از SD و در چارچوب یادگیری عملکردی است.

- مبتنی بر قانون یادگیری **اصلاح خطا**

- اختلاف LMS و BP

- نحوه محاسبه و استفاده از مشتقات

- برخلاف آدالاین، بردار خطا تابعی غیرخطی از پارامترهای شبکه بوده و مشتقاتش به سادگی قابل محاسبه نیست.



- قانون BP از دو مسیر اصلی تشکیل شده

- مسیر رفت (سیگنالهای تابعی) :

- حرکت سیگنالها از چپ به راست

- اعمال بردار ورودی به شبکه و مشاهده اثرات در لایه خروجی از طریق انتشار در لایه های میانی

- در این مسیر، پارامترهای شبکه ثابت و بدون تغییر می مانند.

- مسیر برگشت (سیگنالهای خطا) :

- حرکت سیگنالها از راست به چپ

- توزیع خطا در کل شبکه پس از محاسبه در خروجی از طریق لایه های میانی (پس انتشار خطا)

- تنظیم پارامترهای شبکه طبق قانون اصلاح خطا



## تاریخچه

- دیدیم که شبکه های تک لایه فقط برای طبقه بندی مسائلی که به طور خطی از هم مستقلند، کارایی دارد. ولی **۲۵ سال** تعمیم الگوریتم LMS برای شبکه های چند لایه به طول انجامید.
- ۱۹۵۹ قانون SLPR
- ۱۹۶۰ قانون LMS
- نخستین توصیف BP: پاول وریز ۱۹۷۴
- ۱۹۸۵ راملهارت، هینتون، پارکر: الگوریتم BP در شبکه های عصبی
- شبکه MLP با قانون یادگیری BP دارای **بیشترین کاربرد** در حل مسائل فنی - مهندسی است.



## شبکه پرسپترون چند لایه

- معرفی کلی در فصل دوم

$$\underline{a}^0 = \underline{p}$$

$$\underline{a}^{l+1} = \underline{F}^{l+1} (W^{l+1} \underline{a}^l + \underline{b}^{l+1}), \quad l = 0, 1, \dots, L-1$$

– L تعداد لایه های شبکه

- نمایشی برای بیان ساختار شبکه چند لایه

– R تعداد ورودیها و  $S^i$  تعداد نورونها در لایه i ام

$$(R - S^1 - S^2 - S^3)$$

- هر نرون در شبکه MLP دو محاسبه بر عهده دارد:

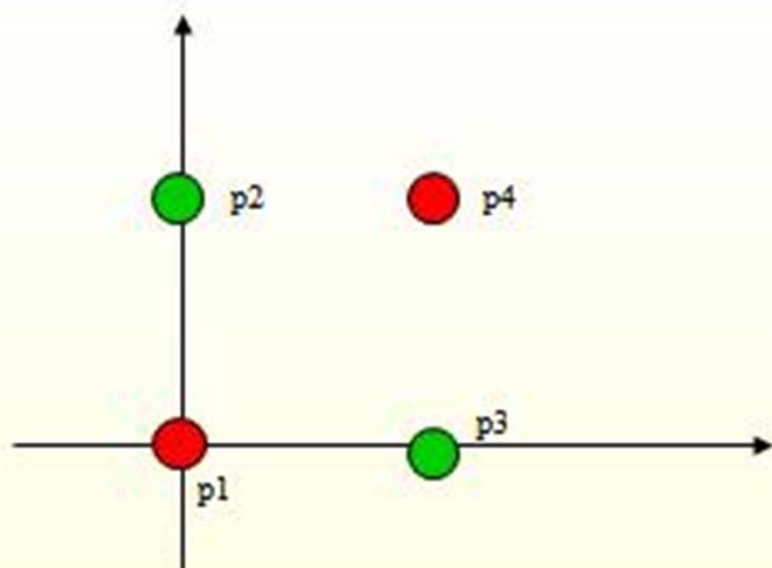
– محاسبه سیگنال تابعی

– محاسبه تخمین لحظه ای از گرادیان خطا نسبت به پارامترهای ورودی نرون



## حل چند مسئله شناسایی الگو

### • مثال ۱) الگوی xor



$$(\underline{p}^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, t^1 = 0), (\underline{p}^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, t^2 = 1)$$

$$(\underline{p}^3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, t^3 = 1), (\underline{p}^4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, t^4 = 0)$$

### • شبکه با ساختار (۲-۲-۱)

- خط مرزی نرون اول به نحوی که الگوی p1 زیر آن و بقیه بالای آن باشند.
- خط مرزی نرون دوم به نحوی که الگوی p4 زیر آن و بقیه بالای آن باشند.
- خروجیهای لایه اول به ترتیب برابرند با [0;1]، [1;1]، [1;1] و [1;0].
- اگر لایه دوم به نوعی عمل AND را انجام دهد، مسئله حل خواهد شد. یعنی فقط در حالتی که تمام ورودیها ۱ است، خروجی مثبت شود.





## شناسایی الگو

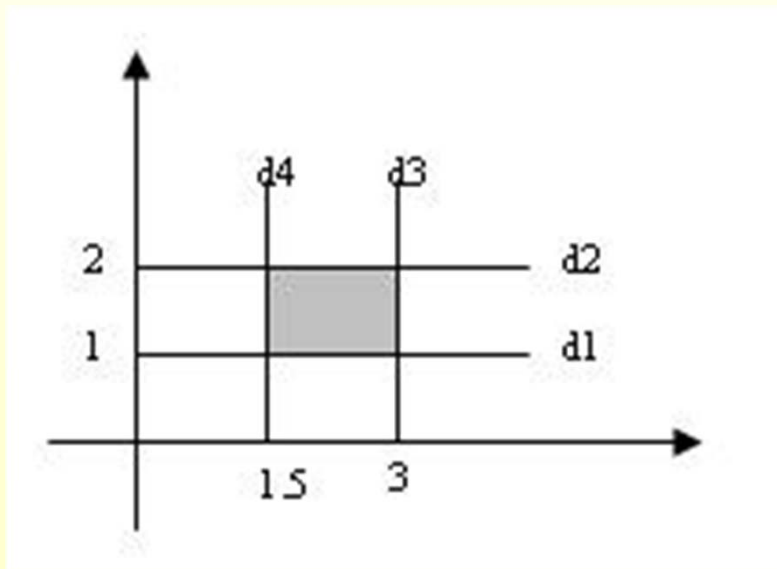
- یکی از (بینهایت) جوابهای مسئله به این قرار است:

$$W^1 = \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix}, \underline{b}^1 = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$$

$$W^2 = \begin{bmatrix} 1 & 1 \end{bmatrix}, b^2 = -1.5$$

- مثال ۲:

خروجی شبکه برای بخش خاکستری ۱  
و برای بقیه جاها صفر باشد.





## شناسایی الگو

- ناحیه مذکور توسط ۴ خط بیان می شود ← ساختار شبکه (۱-۴-۲)
- روش تعیین وزن‌ها مثل قبل
  - لایه اول: معادلات خطوط
  - لایه دوم: انجام عمل AND
- جواب مطلوب:

$$W^1 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ -1 & 0 \\ 1 & 0 \end{bmatrix}, \underline{b}^1 = \begin{bmatrix} -1 \\ 2 \\ 3 \\ -1.5 \end{bmatrix}$$

$$W^2 = [1 \ 1 \ 1 \ 1], b^2 = -3.5$$

- تمرین: درستی پاسخها تست شود.



## الگوریتم BP

$$\hat{F}(k) = \sum_{j=1}^{S_L} e_j^2(k) = \underline{e}^T(k) \underline{e}(k)$$

- تعمیمی از الگوریتم LMS
- شاخص اجرایی مشابه LMS
- $S_L$  - تعداد نرونها در لایه L
- شاخص اجرایی تابعی از پارامترهای شبکه
- قانون یادگیری:

$$W_{ij}^l(k+1) = W_{ij}^l(k) - \alpha \frac{\partial \hat{F}(k)}{\partial W_{ij}^l(k)}$$

$$b_i^l(k+1) = b_i^l(k) - \alpha \frac{\partial \hat{F}(k)}{\partial b_i^l(k)}$$

- در LMS مشتقات به سادگی محاسبه میشود ولی اینجا به دلیل توابع غیرخطی کار سخت تر است.



## الگوریتم BP

- جهت محاسبه ترم اصلاحی نیاز به سیگنال **خطا** داریم
  - برای نرونهای خروجی (به دلیل **قابل رویت** بودن) سیگنال خطا در دسترس است.
  - نرونهای لایه های میانی (نرونهای مخفی) **قابل رویت نبوده** و سیگنال خطا برای آنها قابل اندازه گیری نمی باشد.
- توجه:

تمام نرونهای میانی در مقدار خطا در لایه خروجی سهیمند.

الگوریتم BP هر نرون را با توجه به میزان اثرش در بردار خطا تشویق یا تنبیه می کند.



## خلاصه الگوریتم BP

$$\underline{a}^0 = \underline{p}(k)$$

• مسیر رفت:

$$\underline{a}^{l+1}(k) = \underline{F}^{l+1}(\underline{n}^{l+1}(k)), \quad l = 0, 1, \dots, L-1$$

$$\underline{n}^{l+1}(k) = \underline{W}^{l+1}(k)\underline{a}^l + \underline{b}^{l+1}(k)$$

$$\underline{a}(k) = \underline{a}^L(k)$$

• ملاحظه می شود که پارامترهای شبکه ثابت و بدون تغییر می مانند.

• مسیر برگشت:

- شروع کار از لایه آخر (لایه خروجی) است.
- بردارهای حساسیت خطا از لایه آخر به لایه اول برگشت داده می شوند.
- گرادیان محلی، نرون به نرون با الگوریتم بازگشتی محاسبه می شود.
- در این مسیر نیز پارامترهای شبکه ثابت هستند.



## خلاصه الگوریتم BP

- مسیر برگشت:

$$\underline{\delta}^L(k) = -2\dot{F}^L(\underline{n}^L)\underline{e}(k)$$

$$\underline{\delta}^l(k) = \dot{F}^l(\underline{n}^l)(W^{l+1})^T \underline{\delta}^{l+1}(k), \quad l = L-1, \dots, 1$$

$$\underline{e}(k) = \underline{t}(k) - \underline{a}(k)$$

$$\dot{F}^l(\underline{n}^l) = \text{diag} \left( \dot{f}^l(n_1^l), \dot{f}^l(n_2^l), \dots, \dot{f}^l(n_{s_l}^l) \right)$$

- تنظیم پارامترها:

$$W^l(k+1) = W^l(k) - \alpha \underline{\delta}^l(k)(\underline{a}^{l-1}(k))^T$$

$$\underline{b}^l(k+1) = \underline{b}^l(k) - \alpha \underline{\delta}^l(k), \quad l = 1, 2, \dots, L$$



## خلاصه الگوریتم BP

- توقف الگوریتم:

– میانگین مربعات خطا در هر سیکل یا epoch (جمع مربعات خطا برای تمام الگوهای یادگیری) کمتر از مقدار از پیش تعیین شده ای باشد  
دقت شود که تعداد تکرارها هر سیکل برابر تعداد داده های یادگیری است.

– نرم گرادیان خطا از مقدار از پیش تعیین شده ای کمتر باشد.

- محدودیت BP: توابع تبدیل باید مشتق پذیر باشند.

- الگوریتم BP به صورت دسته ای نیز قابل اجرا است.

-