شبکه‌های عصبی مصنوعی
یک ابزار قوی
برای تحلیل
سازه‌های پرالمان

علی کاوه
استاد
حسین یوسف پور
کارشناس ارشد
دانشکده عمران
دانشگاه علم و صنعت ایران

مقدمه

علم و تکنولوژی استفاده از شبکه‌های عصبی همراه با پیشرفت تکنولوژی هوش مصنوعی اهمیت ویژه‌ای پیدا کرده است و هدف و انگیزهٔ اصلی مطالعهٔ شبکه‌های عصبی دستیابی به مکانیزمهای فرآیند درک و شناخت و انتقال آنها به کامپیوتر و ماشینهای مصنوعی بدون هوش به منظور دریافت و پردازش اطلاعات است. امروزه، محققان به فکر شبیه‌سازی مغز افتاده‌اند تا از این راه بتوانند تواناییهای انسان را به کامپیوتر منتقل کنند. عملکرد مغز انسان با توجه به میلیونها سال تکامل، می‌تواند به عنوان کامل‌ترین و کارآمدترین الگو برای تشخیص وقایع پیرامون خود در نظر گرفته شود. لذا، دانشمندان و محققان در تلاشند تا با درک اصول و مکانیزمهای محاسباتی مغز انسان ـ که عملکردهای بسیار سریع و دقیق را در پی دارد در سیستم‌های مصنوعی بازسازی کنند.

مغز انسان به طور تقریبی از 1.05×10^{10} سلول عصبی تشکیل شده است که هر کدام از این سلولها را نرون می‌گویند. هر یک از این سلولهابا 10^4 سلول دیگر ارتباط دارد و از آنها علایم دریافت می‌کند و اگر مجموع علایم وارده به یک سلول از یک حد آستانه بگذرد، خود سلول به سلولهای دیگر علایم را ارسال می‌دارد. همکاری گسترده وموازی این سلولها، ساده‌ترین توجیه برای عملکرد مغز انسان در تشخیص وقایع محیط بیرونی است.

شبکه‌های عصبی مصنوعی به عنوان یک زمینهٔ تحقیقاتی بسیار فعال حاصل همکاری محققان چند زمینه تحقیقاتی از قبیل روان‌شناسی، بیولوژی، کامپیوتر و سازه است.

یکی از ویژگیهای مهم این شبکه‌ها، پردازش اطلاعات به صورت الگو است. استفاده از پردازش موازی و گسترده نیز توانایی یادگیری و شناخت وقفی را به این شبکه‌ها داده است.

امروزه، مدلهای اجزاء محدود پرالمان و پیچیده هستند و بهترین روش برای حل آنها استفاده از زیر مدلها و روش پردازش موازی است. در این روش، برای هر سازه متناسب با مقادیر ورودی و خروجی و ابعاد آن، یک ساختمان شبکهٔ عصبی انتخاب و سپس شبکهٔ مزبور با استفاده از الگوهای آموزشی پرورش داده می‌شود و این امر زمانی پایان یافته تلقی می‌گردد که مقادیر خطای بدون بعد شبکه و گره‌ها (V و V_G) به سمت عدد یک میل کند. با وجود اینکه پرورش بیش از حد شبکه مقدار V را کاهش می‌دهد ولی مقدار V_G (مقادیر تقریبی در گره‌ها) را افزایش می‌دهد.

کامپیوترهای مورد استفاده برای پیاده کردن شبکه‌های عصبی به کامپیوترهای نسل پنجم معروف هستند که با توجه به در دسترس نبودن این نوع کامپیوترها، نرم‌افزارهایی مخصوص تهیه شده است که مدل‌های شبکه‌های عصبی را روی PCهای شخصی پردازش می‌کند. در این نرم‌افزارها اغلب از روش Back Propagation یا BP برای تعلیم شبکه‌های مستقیم استفاده می‌شود.

با توجه به پیشرفت فعالیتهای کامپیوتری و قابلیت انعطاف ـ پذیری و بالا بودن دقت روش شبکه‌های عصبی، انتظار می‌رود شبکه‌های عصبی قوی‌ترین وسیلهٔ محاسباتی باشد که در قرن بیست و یکم جایگزین روشهای با دقت متوسط و کم گردد.

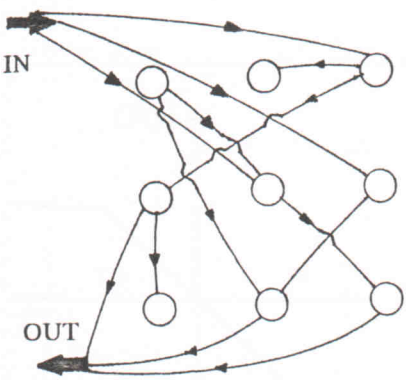
دراین مقاله، ابتدا مفاهیم اساسی مربوط به شبکه‌های عصبی بیان و در قسمت دوم با ارائه مثالهای ساده و کاربردی به تشریح چگونگی استفاده از شبکه‌های عصبی پرداخته شده است. کاربرد مطالب این مقاله

منحصر به رشتهٔ سازه نبوده بلکه در دیگر رشته‌های مهندسی و علوم پایه از جمله رشته‌های ریاضی، فیزیک، شیمی، مکانیک و کامپیوتر نیز قابل استفاده می‌باشد.

شبکه‌های عصبی از یک دیدگاه شباهت زیادی به روشهای گالرکین و سری‌های فوریهٔ مورد استفاده در روشهای دستی دارند ولی از آنجا که در این روش پیدا کردن ضرایب وزنی با سرعت بالایی انجام می‌گیرد، لذا علاوه بر اینکه از مزایای مربوط به دقت روشهای اجزاء محدود برخوردارند، سرعت بالایی را نیز دارا می‌باشند.

تعریف یک شبکهٔ عصبی

یک گراف از تعداد زیادی گره و پاره خطهای جهت‌دار به نام عضو که گره‌ها را به هم ارتباط می‌دهند تشکیل شده است. شبکهٔ عصبی نیز یک سیستم پردازشگر موازی است که به شکل یک گراف جهت‌دار بوده و خصوصیات و تعاریف زیر را می‌توان در مورد آن بیان نمود:



شکل ۱ـ یک ساختار کلی از شبکه‌های عصبی

۱ـ گره‌های گراف مزبور را پردازنده‌های عصبی می‌نامیم.

۲ـ رابط‌های گراف مزبور را اتصالات شبکه نامیده و هر اتصال دارای یک وزن مخصوص به خود بوده که روی علایم عبوری تأثیر می‌گذارد.

۳ـ برای ورودی نرونها محدودیتی وجود ندارد و هر نرون می‌تواند به تعداد دلخواه ورودی قبول کند. همچنین، علایم خروجی هر نرون را می‌توان به انشعابات متعددی تقسیم کرد.

۴ـ نرونها دارای حافظهٔ محلی هستند که وزن اتصالات مربوط به گذشته را در آن نگهداری می‌کنند.

۵ـ هر نرون دارای یک تابع انتقالی است. تابع انتقالی از مقادیر ذخیره شده در حافظهٔ محلی و سیگنال ورودی به نرون استفاده و سیگنال خروجی نرون را محاسبه می‌کند.

۶ـ تغییر وزنهای اتصالات شبکه مبتنی بر قانون یادگیری است.

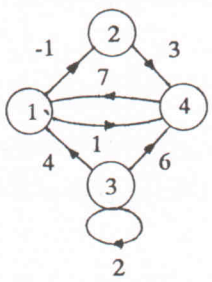
۷ـ تبادل علایم بین دنیای خارج و شبکهٔ عصبی از طریق نرونهای حسی انجام می‌گیرد و نرونهای پاسخ دهنده، اطلاعات را بیرون می‌دهند. بین نرونهای ورودی و خروجی، نرونهای پنهان یا مخفی نیز قرار دارد.

اتصالات یک شبکهٔ عصبی

اتصال بین هر دو نرون با یک وزن صورت می‌گیرد که این وزن میزان همکاری بین دو نرون را مشخص می‌کند. وزنهای فوق قابل تغییر بوده و در طی مرحلهٔ یادگیری شبکه تنظیم می‌شود. بهترین روش برای نشان دادن اتصالات بین نرونها ـ همانند تئوری گرافها ـ استفاده از روش ماتریسی است. ماتریس وزن برای یک شبکه که از n نرون تشکیل شده، به صورت زیر نشان داده می‌شود:

$$W=[W_{ij}]\ \ i,j\ =\ ۱, ۲, ۳, \ldots, n$$

در این فرمول، Wij وزن اتصال بین خروجی نرون jام و ورودی نرون iام است.



$$W=\begin{bmatrix} 0 & 0 & 4 & 7 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 3 & 6 & 0 \end{bmatrix}$$

شکل ۲ـ نمونه‌ای از شبکه عصبی به همراه ماتریس وزن آن

ورودیهای محرک و بازدارنده

در صورتی که یک ورودی با وزن مثبت اثر گذاشته باشد آن ورودی را محرک می‌نامند و چنانچه وزن آن به صورت منفی تأثیر کند آن را ورودی بازدارنده نامند. در برخی از شبکه‌ها پیشنهاد می‌کنند که برای هر نوع از ورودیهای محرک و بازدارنده، ماتریسهای جداگانه‌ای تشکیل شود. بدیهی است که ماتریس کل برابر مجموع ماتریسهای محرک و بازدارنده خواهد بود.

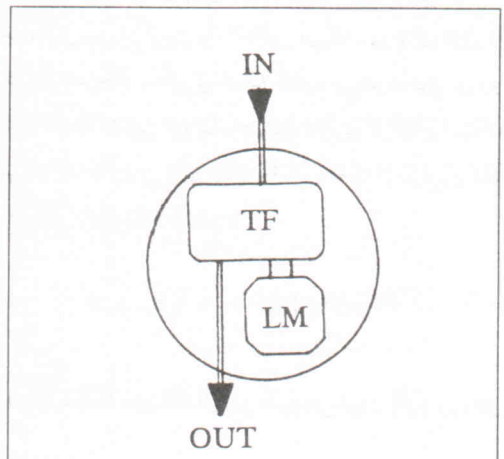
$[W] = [W_E] + [W_I]$ W_E: نشانگر ورودی محرک

W_I: نشانگر ورودی بازدارنده

بررسی ساختمانی یک سلول عصبی

هر پردازنده (سلول) عصبی از یک تابع انتقالی و یک حافظهٔ محلی تشکیل شده و ورودیهای مربوط به تابع انتقال، شامل ورودی، حافظهٔ محلی و پردازندهٔ آن است.

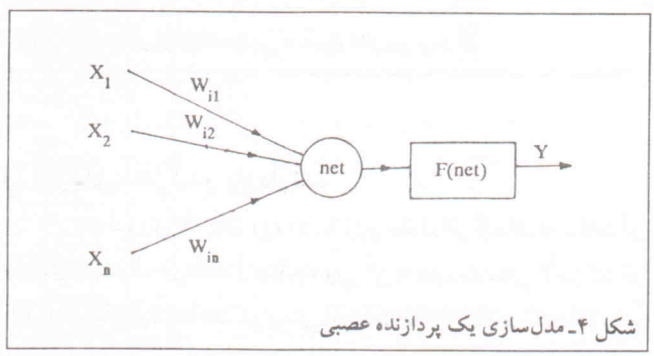
خروجی تابع انتقالی نیز به دو دسته تقسیم می‌شود. دستهٔ اول خروجی مربوط به کل پردازنده (y) و دستهٔ دوّم اطلاعات تازه پردازش شده‌ای که باید دوباره در حافظهٔ محلی ذخیره گردد.



شکل ۳ـ ساختار یک پردازنده عصبی

بررسی تابع انتقال

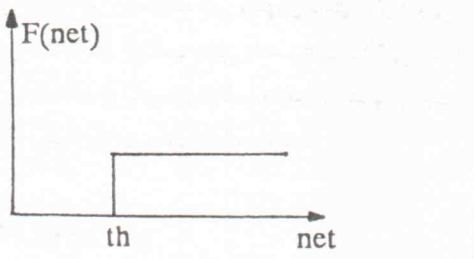
برای محاسبهٔ خروجی پردازنده توابع انتقالی مختلفی را می‌توان بکار گرفت. در شکل زیر، مدل ساده‌ای برای آشنایی در نظر گرفته شده است.



شکل ۴ـ مدل‌سازی یک پردازنده عصبی

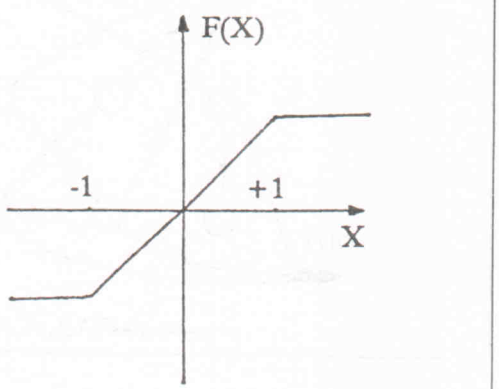
$$net(i) = \sum_{j=1}^{n} W_{ij}\ X_j \qquad y = F(net(i))$$

در یک شبکهٔ بیولوژیکی، وظیفهٔ یک نرون جمع‌آوری علایم ورودی و مقایسهٔ آن با یک سطح آستانه است. مدل‌های ریاضی مشابهی نیز برای نرونهای مصنوعی تهیه شده که از آن جمله می‌توان به حالتهای زیر اشاره نمود:



شکل ۵ـ تابع غیرخطی محدود کنندهٔ سخت

در شبکهٔ مزبور برای خروجی نرون هر مقداری را می‌توان بین [-1, 1] منظور داشت. بنابراین، یک شبکهٔ n نرونی در داخل فوق (ابر) مکعب $[-1, 1]^n$ قرار می‌گیرد.



شکل ۶ـ تابع انتقال خطی

روند تربیت در شبکه‌های عصبی

اصول مهم تربیت در شبکه‌های عصبی شامل موارد زیر است:

۱ـ در این روند با تغییر و اصلاح پارامترهای شبکه، خطای پردازش توسط شبکه مرتباً کاهش می‌یابد تا در نهایت شبکه بتواند وظیفهٔ خود را روی سیگنال ورودی به انجام رساند.

۲ـ اصول و قوانین تربیت در شبکه‌ها همان اصولی است که انسان برای یادگیری از آن استفاده می‌کند. به این معنی است که انسان پس از ارتکاب اشتباه به گونه‌ای در صدد اصلاح خود برمی‌آید که دیگر آن اشتباه تکرار نشود.

تقسیم بندی قوانین یادگیری بر اساس نحوهٔ اعمال یادگیری به شبکه

از این دیدگاه می‌توان نحوهٔ اعمال یادگیری را به دو دسته تقسیم‌بندی نمود. در زیر مراحل عملیاتی مربوط به آنها توضیح داده شده است.

۱ـ یادگیری با نظارت: در این حالت شبکه به صورت یک سیستم ورودی ـ خروجی عمل می‌کند و با دریافت بردار ورودی X بردار خروجی Y را تولید می‌کند. در این روش:

a – دنباله‌ای از زوجهای (x_k, d_k) .... (x_2, d_2) و (x_1, d_1) به شبکه داده می‌شود.

b ـ در زوجهای مزبور مؤلفه اول بردار ورودی به شبکه و مؤلفه دوم بردار خروجی مورد نظر است.

c – بعد از اعمال ورودی x_i به شبکه، خروجی آن باید برابر d_i باشد، لیکن مقدار خروجی با تقریب برابر Y_i است؛ لذا از اختلاف بین Y_i و d_i برای تربیت شبکه و اصلاح وزنها استفاده می‌گردد.

d ـ بردار d_i را بردار مربی می‌گویند.

۲ـ یادگیری بدون نظارت: در این روش؛

a ـ بردار مربی یا خروجی مطلوب شبکه وجود ندارد.

b ـ قوانینی که در این دسته قرار می‌گیرند در پاسخ به ورودی شبکه و کشف همبستگی بین ورودیهای مختلف، وزنهای شبکه را اصلاح و باعث تفکیک کلاسها از همدیگر می‌گردد.

c ـ الگوهای ورودی که دارای ویژگیهای مشترکی هستند در یک کلاس قرار دارند.

d ـ شبکه قادر است این ویژگیهای مشترک را با استفاده از الگوهای ورودی کشف کند.

تقسیم بندی قوانین یادگیری بر اساس چگونگی یادگیری

بر اساس نوع یادگیری، تقسیم بندی زیر را می‌توان داشت:

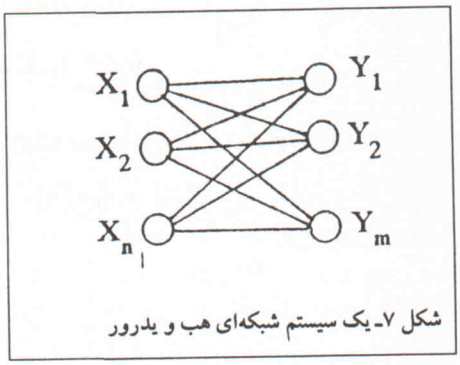
۱ ـ یادگیری هم واقعه

۲ ـ یادگیری بر اساس عملکرد

۳ ـ یادگیری رقابتی

۴ ـ یادگیری در شبکه‌های برگشتی (یک نوع از شبکهٔ انتسابی)

۱ ـ یادگیری هم واقعه یا قانون یادگیری هب: شبکه‌ای به شکل زیر را که حاوی n نرون در لایهٔ ورودی و m نرون در لایهٔ خروجی است، مد نظر قرار می‌دهیم:



شکل ۷ـ یک سیستم شبکه‌ای هب و یدرور

تابع انتقال همانی در نظر گرفته می‌شود: $f(x) = x$

هدف اصلی از تربیت شبکه، آموزش وزنهای (x_L,d_L) .... و (x_2, d_2) و (x_1, d_1) به شبکه است. برای این منظور، مقادیر مربوط به وزنهای اولیه را برابر صفر قرار داده و بر اساس رابطهٔ زیر وزنها را تغییر می‌دهیم:

$$(W_{ij})_{new} = (W_{ij})_{old} + d_{ki} \times X_{kj}$$

در رابطهٔ فوق:

$(wij)_{old}$: وزن اتصال بین نرون jام در لایهٔ ورودی ونرون iام در لایهٔ خروجی

d_{ki} : iامین مؤلفه بردارهای ستونی d_k

X_{kj} : jامین مؤلفهٔ بردارهای ستونی X_k

می‌باشد.

به این ترتیب، فرم ماتریسی معادلهٔ فوق به شرح زیر خواهد بود:

$$(W)_{new} = (W_{ij})_{old} + d_k x_k^T$$

ماتریس وزن شبکه در پایان تربیت شکل زیر را خواهد داشت:

$$W = d_1 x_1^T + d_2 x_2^T + \ldots d_L x_L^T$$

انتظاری که در پایان تربیت از شبکه می‌رود این است که اگر یکی از بردارهای تربیتی در ورودی شبکهٔ x_k قرار گیرد، خروجی شبکه برابر d_k باشد و به عبارتی:

$$Y_k = wx_k \quad ; \quad k = 1,2, \ldots, L$$

و این تنها در صورتی میسر خواهد بود که مجموعهٔ بردارهای $x_1,x_2,\ldots,x_L$ حالت اورتو نرمال داشته باشد.

بر اساس تعریف، در حالت اورتونرمال داریم:

$$x_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \qquad i, j = 1, 2, \ldots, L$$

$$\begin{aligned} Y_K &= w \times x_k = [d_1 x_1^T + d_2 x_L^T + \ldots + d_L x_L^T]\, x_k \\ &= d_1 x_1^T x_k + \ldots + d_k x_k^T + d_L x_L^T x_k \\ &= d_k (x_k^T x_k) \\ &= d_k \end{aligned}$$

در عین حال، اورتونرمال بودن با محدودیتهایی همراه است:

ـ اولاً تعداد بردارهای (L) باید کوچکتر یامساوی با بعد فضای ورودی (n) باشد. به دیگر سخن، در یک فضای n بعدی مقدار ماکزیمم تعداد بردار متعامد برابر n است.

ـ اندازه تمام بردارها نیز باید برابر واحد باشد.

حال اگر بردارهای X_K برهم عمود نباشند آنگاه شبکه در تخمین d_k دچار خطا خواهد شد. به این ترتیب، با فرض واحد بودن اندازه بردارها می‌توان نوشت:

$$\begin{aligned} E_k &= d_k - Y_k = d_k - \{d_1 (x_1^T x_k) + \cdots + d_k (x_k^T x_k) + \ldots + d_L (x_L^T x_k)\} \\ &= d_k - d_k (x_k^T x_k) - \sum_{j \neq k} d_j (x_j^T x_k) \\ &= d_k - d_k - \sum_{j \neq k} d_j (x_j^T x_k) \end{aligned}$$

در این حالت، باید وزنهای شبکه به گونه‌ای انتخاب شود که متوسط مربعات خطای شبکهٔ ورودی روی بردارهای $x_1, \ldots, x_L$ مینیمم شود. این کمیت به صورت زیر تعریف می‌شود:

$$F(w) = \frac{1}{L} \sum_{k=1}^{L} | d_k - w x_k |^2$$

دو تعریف زیر را مد نظر قرار داده و یک روش ماتریسی برای به دست آوردن ماتریس وزن ارائه می‌دهیم:

۱ ـ شبه معکوس ماتریس حقیقی $B_{p \times q}$ ماتریس حقیقی $B^+_{q \times p}$ است که دارای خواص زیر می‌باشد:

$$BB^+B = B, \quad B^+BB^+ = B^+, \quad BB^+ = (BB^+)^T, \quad B^+B = (B^+B)^T$$

۲ ـ مربع نرم ماتریس، برابر مجموع مربعات تمام درایه‌های ماتریس $B_{p \times q}$ است:

$$\| B \|^2 = \sum_{i=1}^{p} \sum_{j=1}^{q} b_{ij}^2$$

به این ترتیب:

$$F(w) = \frac{1}{L} \| d - wx \|^2$$

$$x = (x_1, \ldots, x_l) \qquad d = (d_1, \ldots, d_l)$$

و X و d ماتریسهای $n \times L$ و $m \times L$ هستند که ستونهای آنها را بردارهای x_k و d_k تشکیل می‌دهند.

ماتریس تریس (trace) که با tr(S) نشان می‌دهیم به شرح زیر تعریف می‌شود:

$$s = (wx - d)(wx - d)^T \rightarrow F(w) = \frac{1}{L} tr(s) \qquad (*)$$

رابطهٔ فوق با استفاده از روابط جبری اثبات می‌شود. همچنین طبق تعریف، tr(S) عبارت است از مجموع درایه‌های واقع بر قطر اصلی.

با استفاده از دو اتحاد زیر:

$$B^+BB^T = B^T \;, \quad BB^T(B^+)^T = B$$

$$\rightarrow S = (dx^+ - w)(xx^T)(dx^+ - w)^T + d(I - x^+x)\, d^T$$

به ازای $w = dx^t$ جملهٔ ماتریسی اول همواره صفر خواهد بود.

با توجه به اینکه جملهٔ دوم مستقل از W است، درایه‌های ماتریسی S به کمینه مقدار خود می‌رسد و در نتیجه tr(s) طبق رابطهٔ (*) F(W)، نیز کمینه خواهد بود و لذا:

هنگامی که مجموعهٔ بردارهای ورودی اورتونرمال نباشد وزنهای بهینه درایه‌های ماتریس $W^* = dx^+$ خواهد بود.

۲ ـ یادگیری بر اساس عملکرد: در اینجا می‌خواهیم تابع ارزشیابی را که دارای شکل زیر است، کمینه کنیم:

$$F(w) = \lim_{N \to \infty} \frac{1}{n} \sum_{k=1}^{N} [d_k - Y_k]^2$$

$$Y_k = W \times x_k = W^T . x_k$$

$$x_k = (1, x_{1k}, x_{2k}, \ldots, x_{nk})^T$$

$$W = (w_0, w_1, w_{2i}, \ldots, w_n)^T$$

در صورتی که بردارهای x_i به صورت تصادفی و با تابع چگالی f(x)

انتخاب شده باشند، آنگاه رابطهٔ بالا به فرم آماری زیر تبدیل می‌شود:

$$F(w) = E[(d_k - Y_k)^2] = \int_x (d_k - y_k)^2 F(x)dx$$
$$= E[(d_k - Y_k)^2]$$
$$= E[(d_k - w^T \times x_k)^2]$$
$$= E[d_x^2 - 2d_k w_{x2}^T + w_{x_k}^T w_k^T w]$$
$$= E[d_k^2] - 2w^T E[d_k x_k] + w^T E[x_k x_k^T] w$$
$$= p - 2w_q^T + w_R^T w$$

که در رابطهٔ مزبور:

P همبستگی خروجی، q بردار همبستگی متقابل خروجی و ورودی، و R ماتریس همبستگی ورودیها می‌باشند.

F(w) یک تابع درجه دوم است که در فضای وزنها یک سهموی می‌باشد. برای حرکت به طرف مینیمم قسمت سهموی با بیشترین سرعت از هر نقطه در روی آن باید در خلاف جهت گرادیان حرکت نمود که بعد از رسیدن به گرادیان صفر خواهد بود:

$$\nabla F(w) = \nabla(P - 2W_q^T + W^T RW)$$
$$= -2q + 2RW = 0 \rightarrow w^* = R^{-1}q$$

اما محاسبهٔ خود R و q نیز کار آسانی نمی‌باشد.

در اینجا برای رسیدن به وزن بهینه از الگوریتم ویدرو استفاده می‌کنیم. در روش فوق برای رسیدن به مینیمم سهموی باید گرادیان را برای هر نقطه روی تابع F(w) حساب یا تخمین زد:

$$\nabla F(w) = \nabla\left\{\lim \frac{1}{N}\sum_{k=1}^{N}(d_k - Y_k)^2\right\}$$
$$= \left\{\lim \frac{1}{N}\sum_{k=1}^{N}\nabla(d_k - Y_k)^2\right\}$$
$$= \left\{\lim \frac{1}{N}\sum_{k=1}^{N}2(d_k - Y_k)\nabla(-Y_k)\right\}$$
$$= \left\{\lim \frac{1}{N}\sum_{k=1}^{N}2(d_k - Y_k)*(-X_k)\right\}$$
$$= \left\{-\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}2\varepsilon_k X_k = -2E[\varepsilon_k X_k]\right\}$$

که در آن ε_k خطای شبکه در پاسخ به ورودی X_k می‌باشد و لذا: تخمین گرادیان منجر به گرفتن معدل از تعداد زیادی $\varepsilon_k X_k$ و ضرب آنها در (-2) می‌شود. ویدر و وهف ثابت کردند که اگر از مقدار لحظه‌ای $\varepsilon_k X_k$ به جای متوسط آنها نیز استفاده شود، باز وزن به سمت وزن بهینه میل خواهد نمود. قانون فوق را قانون دلتا یا الگوریتم LMS نیز می‌نامند که به صورت زیر ارائه می‌شود:

$$\hat{\nabla} = -2\varepsilon_k x_k$$
$$w_{k+1} = w_k + \alpha(-\hat{\nabla}) = w_k + 2\varepsilon_k x_k$$

α به میزان یادگیری موسوم بوده و سرعت همگرایی و پایداری الگوریتم را کنترل می‌کند. برای مقدار مزبور باید موارد زیر را مورد توجه داشت:

- α یک عدد کوچکتر از عکس بزرگترین مقدار ویژه ماتریس R است.
- با توجه به اینکه در این قانون نیازی به محاسبه R نیست، لذا پیدا کردن حد بالایی، برای α کار ساده‌ای نیست و مقدار α با روش سعی و خطا به دست می‌آید.
- اگر مقدار α خیلی کوچک انتخاب شود، مدت زمان همگرایی خیلی طولانی خواهد بود و اگر α خیلی بزرگ باشد، وزن شبکه همگرا نشده و عملاً الگوریتم ناپایدار خواهد بود.

برای مثال، $0.005 < \alpha < 0.5$ انتخاب و مقدار $\alpha = 0.005$ به عنوان حدس اولیه انتخاب می‌شود.

یادگیری در شبکه‌های مستقیم

این نوع شبکه‌ها، قسمتهای زیر را دارا هستند:

۱ـ پردازنده‌های شبکه به چند لایه مختلف تقسیم می‌شوند.

۲ـ حداقل تعداد لایه‌ها برابر ۲ است.

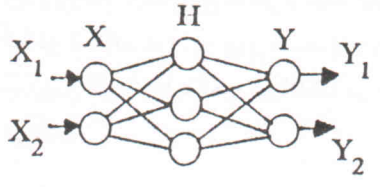
۳ـ پردازنده‌های هر لایه فقط مجاز به دریافت سیگنال از پردازندهٔ لایهٔ ماقبل خود است و سیگنال خروجی این پردازنده نیز به پردازنده‌های بعدی اعمال می‌شود.

۴ـ اولین لایه را لایهٔ ورودی می‌نامند. این لایه سیگنال ورودی را از دنیای خارج شبکه گرفته و بین پردازنده‌های لایهٔ بعد توزیع می‌کند و هیچ گونه پردازشی روی سیگنال ورودی انجام نمی‌دهد.

۵ـ لایهٔ آخر لایهٔ خروجی نام دارد، این لایه سینگال خروجی شبکه را که همان پاسخ به ورودی‌ها می‌باشد در اختیار دنیای خارج شبکه قرار می‌دهد.

۶ـ لایهٔ میانی را که هیچ ارتباطی با دنیای خارج ندارد لایهٔ مخفی یا

پنهان می‌گویند.



شکل ۸ـ ساختار یک سیستم شبکه‌ای مستقیم

قوانین یادگیری در شبکه‌های مستقیم :

۱ـ در صورتی که توابع انتقال در پردازنده‌های خروجی یکی از سه حالت؛ تابع غیر خطی محدود کنندهٔ سخت، تابع غیر خطی سیگوند و تابع خطی محدود را داشته باشد خروجی مورد نظر که همان مربی شبکه نیز می‌باشد یکی از دو مقدار (۱ و ۱ـ) را به خود می‌گیرد.

۲ـ هدف کمینه کردن تابع عملکرد شبکه است.

۳ـ در صورت استفاده از تابع غیر خطی محدود کنندهٔ سخت، ساختار مذکور را پرسپترون نامند. برای تربیت این شبکه از قانون یادگیری پرسپترون که توسط رزن بلت ارائه گردیده است استفاده می‌کنند.

۴ـ خطای کوانتیزه شده برای این شبکه چنین تعریف می‌شود:

$$\tilde{e}_k = d_k - y_{2k}$$

Y_{2k} مقدار خروجی موجود و d_k مقدار خروجی است که انتظار می‌رود.

a : اگر برای الگوی ورودی k ام، مقدار e_k صفر باشد هیچ گونه اصلاحی صورت نمی‌گیرد.

b : اگر خطای کوانتیزه شده مثبت باشد آنگاه ضریبی از بردار ورودی با بردار وزن جمع می‌شود.

c : اگر خطای کوانتیزه شده منفی باشد آنگاه ضریبی از بردار ورودی از بردار وزن کم می‌شود.

با توجه به اینکه خطای کوانتیزه شده فقط مقادیر {-2,0,2} را به خود می‌گیرد، لذا:

$$w_{k+1} = w_k + \alpha \frac{\tilde{e}_k}{2} x_k$$

حالتی که تابع انتقال از نوع سیگموند باشد:

۱ـ در این حالت تابع انتقال یک تابع غیر خطی با معادلهٔ

$$y_{1k} = tanh(s_k)$$

خواهد بود.

۲ـ در اینجا هدف کمینه کردن تابع متوسط مربعات خطا و حرکت در خلاف جهت گرادیان است.

۳ـ برای سهولت کار از تقریب گرادیان مربع خطای لحظه‌ای به جای گرادیان واقعی متوسط مربعات استفاده می‌شود.

$$\tilde{e}_k = d_k - Y_{1k} = d_k - sgm(s_k) \qquad (۲۰۷)$$

$$\hat{\nabla}_k = \frac{\partial [e_k]^2}{\partial w_k} - = 2\tilde{e}_k \frac{\partial e_k}{\partial w_k} = -2\tilde{e}_k \frac{\partial}{\partial w_k} [sgm(s_k)]$$

۴ـ حال اگر از قانون مشتق توابع زنجیره‌ای استفاده کنیم:

$$\hat{\nabla}_k = -2e_k \, sgm'(s_k) \frac{\partial s_k}{\partial w_k}$$

$$s_k = x_k^T w_k \rightarrow \frac{\partial s_k}{\partial w_k} = x_k$$

$$\rightarrow \hat{\nabla}_k = -2\tilde{e}_k \, sgm'(s_k) \, x_k$$

به این ترتیب باید خلاف جهت گرادیان حرکت کرد:

$$w_{k+1} = w_k \alpha(-\hat{\nabla}_k) = w_k + 2\alpha \{\tilde{e}_k \, sgm'(s_k) \times x_k\}$$

۵ـ با جایگذاری مشتق تابع سیگموند داریم:

$$sgm'(s_k) = \frac{\partial}{\partial s_k} tanh(s_k) = 1 - tanh^2(s_k) = 1 - Y_{1k}^2$$

$$w_{k+1} = w_k + 2\alpha\tilde{e}_k (1 - Y_{1k}^2) x_k$$

۶ـ در صورتی که منحنی مربوط به سطح متوسط خطا برای سه نوع تابع انتقال رسم شود (توسط یک نرم افزار) نتایج زیر را خواهیم داشت:

a) برای رسیدن به Min، تابع خطا برای شبکه‌های خطی ساده‌ترین راه و دارای یک کمینه قطعی است.

b) شبکه‌های سیگموندی از شبکه‌های آستانه‌ای خطی آسان‌تر بوده و دارای کمینه محلی است.

c) شبکه‌های محدود کنندهٔ سخت دارای کمینه محلی است که

مانع رسیدن به جواب می‌شود.

3- حالتی که شبکه عصبی در لایهٔ خروجی m نرون دارد. در این حالت روابط ماتریسی به دست آمده از بالا، شکل زیر را به خود می‌گیرد:

$$w_{k+1} = w_k + \alpha \frac{\tilde{E}_k}{2} x_k^T$$

$$w_{k+1} = w_k + 2\alpha \tilde{E}_k x_k^T$$

که در آن:

W: یک ماتریس $m\times(n+1)$ که هر سطر آن متناظر با وزنهای یک پردازنده در لایهٔ خروجی است.

E_K: بردار ستون $M\times1$ با مؤلفه e_i

E_K: بردار ستونی $m\times1$ با مؤلفه $e_i\ (1- y_i^2)$

X: یک بردار ستونی $1\times(m+1)$ که همان ورودی است.

تشخیص الگو :

تشخیص الگو در حالت کلی عبارت است از انتصاب یک الگو به یکی از چند کلاس معین و از پیش تعیین شده و از لحاظ ریاضی، می‌توان تشخیص الگو را یک نگاشت از یک فضا به فضای دیگری در نظر گرفت. در این حالت، فضای اول را فضای نمایش (Representation Space) و فضای دوم را فضای تفسیر (Interpretation Space) می‌نامند.

بر این اساس خواهیم داشت:

۱ـ اعضای فضای نمایش را الگو یا ویژگیهای آنها تشکیل می‌دهند.

۲ـ در فضای تفسیر، کلاسهای معین و تعریف شده قرار دارند.

۳ـ برای اینکه یک الگو مورد نگاشت واقع شود باید به فرم مناسب که همان بردارهای فضای نمایش هستند تبدیل شوند.

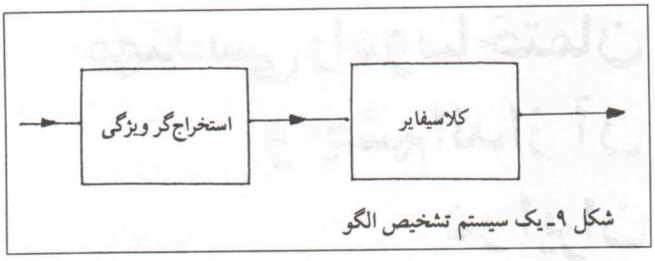
۴ـ برای نمایش هر الگو در فضای نمایش، اندازه‌گیریهای متعددی روی یک الگو صورت می‌گیرد که نتیجه هر اندازه‌گیری یک ویژگی الگو نامیده می‌شود.

۵ـ برداری را که هر مؤلفه آن یکی از ویژگیهای الگو باشد بردار ویژگی می‌نامند. این بردار در فضای نمایش قرار دارد. مراحل ۱ الی ۵ را مراحل استخراج ویژگی می‌نامند.

لازم به توضیح است که ویژگی یک الگو می‌تواند از نوع عددی یا غیر عددی و توصیفی باشد.

۶ـ بعد از مرحلهٔ استخراج ویژگی، مرحلهٔ دسته بندی شروع می‌شود. این کار توسط رده‌بند (Classifier) صورت می‌گیرد.

۷ـ توصیف ریاضی عمل رده‌بند، برقراری یک نگاشت بین فضای نمایش و فضای تفسیر است.



شکل ۹ـ یک سیستم تشخیص الگو

روشهای مختلف تشخیص الگو

دو روش برای تشخیص الگو پیشنهاد می‌شود:

۱ـ روشهای ریاضی: برای حل مسائل مربوط به تشخیص الگو، روشهای ریاضی به دو دسته تقسیم می‌شود که عبارتند از: تشخیص الگوی ساختاری و تشخیص الگوی آماری. در دسته اول خواص هندسی و توپولوژیکی الگوها برای تشخیص آنها مورد استفاده قرار می‌گیرد و در روش دوم، هر الگو به صورت یک بردار در فضای برداری نمایش داده می‌شود و هدف از تشخیص الگو تقسیم بندی فضا و تعیین ناحیه‌ای است از فضا که بردار الگو به آن ناحیه تعلق دارد.

۲ـ استفاده از شبکه‌های عصبی: همانطور که در مبحث تربیت شبکه‌ها نیز بیان گردید، یک شبکهٔ انتسابی چند لایه که با استفاده از تعداد محدودی زوج (ورودی ـ خروجی) تربیت شود، بهترین نگاشت ممکن بین این زوجها را بر اساس مینیمم متوسط مربعات خطا تقریب خواهد زد. همچنین از شبکه‌های عصبی می‌توان بعنوان رده‌بند استفاده نمود و در صورت تربیت درست شبکه می‌توان به مینیمم خطای قابل حصول دست یافت.

شبکهٔ عصبی با استفاده از زوجهای ورودی : خروجی مرزهای جدا کنندهٔ بهینه را در فضای نمایش بوجود می‌آورد. در حقیقت، این شبکه‌ها می‌توانند با استفاده از ساختار پردازش موازی خود عمل تخمین توابع چگالی (مرحلهٔ یادگیری) و استفاده از آن در تابع تصمیم گیری برای پیدا کردن مرزهای جدا کننده (مرحلهٔ تشخیص) را به طور همزمان و در کمترین زمان ممکن است انجام دهد.

ادامهٔ مطلب و کاربرد مفاهیم و روشهای ارائه شده در این مقاله در شمارهٔ آتی مجله چاپ خواهد شد.