

## فهرست مطالب:

4	پیش گفتار
5	<b>فصل اول: مقدمه</b>
7	1-1 انواع سیستم های هوشمند
7	1-1-1 سیستم های خیره
8	2-1-1 سیستم های فازی
9	3-1-1 شبکه های عصبی مصنوعی
10	<b>فصل دوم: شبکه های عصبی مصنوعی</b>
11	1-2 چگونگی پیدایش
14	2-2 ساختار شبکه عصبی
16	2-2-1 اتوابع تحریک
21	3-2 توانایی های شبکه عصبی
21	1-3-2 قابلیت یادگیری
22	2-3-2 قابلیت تعمیم دهی
22	3-3-2 مقاوم بودن شبکه در مقابل خطا
23	4-2 تاریخچه شبکه های عصبی
25	5-2 آموزش شبکه های عصبی
25	1-5-2 آموزش با ناظر
26	2-5-2 آموزش بدون ناظر
27	<b>فصل سوم: انواع شبکه های عصبی</b>
28	1-3 شبکه های عصبی پرسپترون
30	1-1-3 شبکه های پرسپترون چند لایه
31	2-1-3 قانون یادگیری پس انتشار خطا
35	2-3 شبکه آدلاین
37	3-3 شبکه عصبی انتشار متقابل
37	1-3-3 آموزش لایه اول
38	2-3-3 آموزش لایه دوم
40	4-3 شبکه RBF
	5-3 مدل عصبی عادلای و پارک
	<b>فصل چهارم: کاربرد شبکه عصبی در طراحی سازه</b>
	1-4 مقدمه
	2-4 بهینه سازی سازه ها با استفاده از شبکه ها

	3-4 پیش بینی مقاومت نهائی برش تیرهای عمیق بتن مسلح دو سر ساده
	4-4 بهینه سازی تیرهای فولادی سرد نورد
	5-4 بهینه سازی دالهای بتن آرمه
	1-5-4 مدل دینامیکی عصبی عادلای و پارک
	6-4 سازه های فولادی سرد نورد فضاکار
	7-4 بهینه سازی سازه های فولادی با وجود عدم قطعیت
	8-4 کاربرد شبکه عصبی در سازه های بتنی
	9-4 مدل های محاسبات عصبی در آنالیز و طراحی سازه ها
	10-4 مدلسازی مدل اجزاء محدود پل راه آهن
	<b>فصل پنجم: نتیجه گیری</b>
	فهرست منابع

## پیش‌گفتار:

از زمانی که در دهه 70 استفاده از کامپیوترهای محاسبه گر جای خود را در مهندسی و محاسبات عددی باز کرده اند، تحول عظیمی در روند مسائل مهندسی سازه بخصوص در طراحی های طولانی و زمان بر به وجود آمده است. با این حال هنوز برخی از مسائل در طراحی وجود دارند که به علت پیچیدگی اطلاعات و یا نبود تئوری مدونی، حتی به کمک کامپیوتر نیز برای رسیدن به جواب، زمان طولانی صرف می شود. شبکه های عصبی مصنوعی ابزار قدرتمندی هستند که قادرند یک رابطه عمومی بین اطلاعات حجیم و پیچیده ناشی از آزمایشات و مثالهای تجربی بدست آورند که در حل مسائل مشابه بدون صرف زمان زیادی مفید خواهند بود.

در این نوشتار سعی شده است تا اندکی از کارایی ها و توانایی های این شبکه های مصنوعی معرفی شود. سپس در مورد مسائلی از طراحی سازه که توسط شبکه های عصبی مدل سازی و حل شده اند، بحث خواهد شد.

## فصل اول

### مقدمه

---

در سالیان اخیر شاهد حرکت مستمر از تحقیقات تئوری و کاربردی به خصوص در زمینه پردازش اطلاعات، برای مسائلی که برای آنها راه حل دقیقی موجود نیست و یا به راحتی قابل حل نیستند، بوده ایم. با توجه به این امر، علاقه فزاینده ای در توسعه تئوریک سیستم های دینامیکی هوشمند که مبتنی بر داده های تجربی هستند، ایجاد شده است. شبکه های عصبی مصنوعی جزء این دسته از سیستم های دینامیکی قرار دارند که با پردازش روی داده های تجربی، دانش و یا قانون نهفته در ورای داده ها را به ساختار شبکه منتقل می کنند. به همین خاطر به این سیستم ها، سیستم های هوشمند می گویند چرا که بر اساس محاسبات روی داده های عددی یا مثال ها، قوانینی کلی را استخراج کرده و یاد گرفته اند. [1]

در نیم قرن اخیر، به کارگیری سیستم های هوشمند در بسیاری از رشته ها رشد چشمگیری داشته است و در زندگی روزمره از وسایلی که به نوعی در آنها یک سیستم هوشمند ایجاد شده است استفاده می گردد. در خیابان در پشت چراغ راهنمایی نوشته شده این تقاطع توسط سیستم کنترل هوشمند کنترل می شود. بعضی از لوازم خانگی مانند ماشین های لباسشویی و تلویزیون ها بر پایه منطق فازی کار می کنند. (در این سیستم ها از مدل های مشابه الگوی رفتار بشر، چگونگی تصمیم گیری و تجربه انسان و شبکه های اعصاب و یا ساختار کروموزومهای سلول ها و همچنین استدلال احتمالی در حل مسائل استفاده می شود). در بعضی موارد، وقتی سخن از سیستم های هوشمند به میان می آید تصویری رویاگونه از سیستمی شبیه و حتی کاملتر از انسان در ذهن ایجاد می شود. البته در حال حاضر این مطلب رویایی بیش نیست و هوشمندی در محدوده قوانین و ضوابطی است که برای سیستم تعریف می شود. هر چه این قوانین و ضوابط کاملتر شود، رفتار سیستم به رفتار انسان نزدیکتر می شود. کارایی سیستم های هوشمند زمانی جلوه بیشتری دارد که امکان اشتباه در انسان افزایش می یابد. درحالت هایی که پارامترهای دخیل در تصمیم گیری زیاد هستند یا ابهامات مسایل زیاد است این کارایی آشکار می شود.

[یک سیستم هوشمند مانند یک جعبه سیاه است که بر اساس ورودی ها یک سری خروجی دارد. نوع سیستم هوشمند استفاده شده نوع عملیات انجام شده در داخل جعبه سیاه را مشخص می کند]. تحقیقات در زمینه سیستم های هوشمند منجر به یافتن سیستم های هوشمندی مانند سیستم های خبره، شبکه های عصبی مصنوعی و منطق فازی شده است.

## 1-1 انواع سیستم های هوشمند:

### 1-1-1 سیستم های خبره (Expert Systems):

در یک تعریف کوتاه می توانیم بگوییم سیستم های خبره، برنامه های کامپیوتری پیچیده ای هستند که می توانند مانند انسان در چگونگی انجام کار تصمیم گیری کنند. در سیستم های خبره تجربه و تخصص یک شخص خبره برای حل یک مسأله به صورت یک سری گزاره های شرطی تعریف می شود و سپس این گزاره های شرطی (قوانین) در مورد یک مسأله مشخص یک پایگاه معرفت به وجود می آورند. یکی از مشکلات، چگونگی کسب معرفت است که در این راستا سه مشکل عمده وجود دارد:

1. کسب شناخت به روش های دستی هزینه و وقت زیادی را صرف می کند.
2. بعد از کسب شناخت برای یک مسأله مشخص، آن را نمی توان به حل مسأله دیگر تعمیم داد.
3. با توجه به این که هر متخصص برای حل مسائل با راه حل های نامشخص روش خاص خودش را دارد، نمی توان روش صحیح حل این مسائل را شناخت.

این مشکلات باعث شد سیستم های یادگیری ماشینی (Machine Learning -Systems) طراحی شوند که قادرند معرفت یک متخصص را به طور خودکار از روی چندین مثال یاد بگیرند. به عبارت بهتر قبل از استفاده از یک سیستم یادگیر ماشینی، متخصص رشته مورد نظر باید آن را آموزش دهد. این سیستم در طی یک زمان مشخص به چگونگی تشخیص یک متخصص نگاه می کند و با تغییر ضرایب ساختار خود سعی می کند مانند او تصمیم بگیرد و بعد از این مرحله، سیستم برای پاسخگویی و استفاده یک شخص غیر متخصص آماده است.

### 1-1-2 سیستم های فازی (Fuzzy systems):

نظریه مجموعه های فازی در سال 1965 ارائه شده است. تفاوت عمده بین (مجموعه های کلاسیک) و مجموعه های فازی در مفهوم درجه عضویت اعضای مجموعه است. درجه عضویت می تواند هر عدد دلخواه در فاصله صفر و یک باشد. به عبارت دیگر هر شیء می تواند کاملاً عضو یک مجموعه فازی باشد، کاملاً عضو آن نباشد و یا تقریباً عضو آن باشد که با درجه عضویت آن شیء در مجموعه ها بیان می شود. کلمه Fuzzy یا مبهم برای توصیف مجموعه هایی به کار رفته است که میزان و یا درجه عضویت اعضای آن مبهم است استفاده از مجموعه های فازی در روند تحلیل تصمیم گیری در محیط هایی که اطلاعات موجود در آن ناقص، مبهم و یا گنگ هستند بسیار کار آمد است. نتیجه حاصل شده در این روش «فاکتور قطعیت» نام دارد که در روند تصمیم گیری محاسبه می شود و معین کننده درجه درستی تصمیم گرفته شده است. فاکتور قطعیت می تواند دارای مقادیری در محدوده های گوناگون باشد، ولی معمولاً پس از تعیین، آن را در محدوده صفر و یک Normalize می کنند.

### 1-1-3 شبکه های عصبی مصنوعی (Artificial Neural Networks):

شبکه های عصبی یکی دیگر از سیستم های هوشمند هستند که جای خود را در اواخر دهه 80 در کاربردهای علمی و مهندسی باز کرده اند. این شبکه ها ابزار قدرتمندی برای پردازش اطلاعات نامنظم و پیچیده ای هستند که تئوری محکم و دقیقی نمی توان برای آنها به کار برد. این ابزار قادرند بین اطلاعاتی که از آزمایشات تجربی و حل مثالهای متعدد بدست آمده است از طریق درون یابی ارتباط نهفته ای تعیین کنند که در حل مسائل مشابه دیگر مورد استفاده قرار می گیرد.

## فصل دوم

### شبکه های عصبی مصنوعی

---

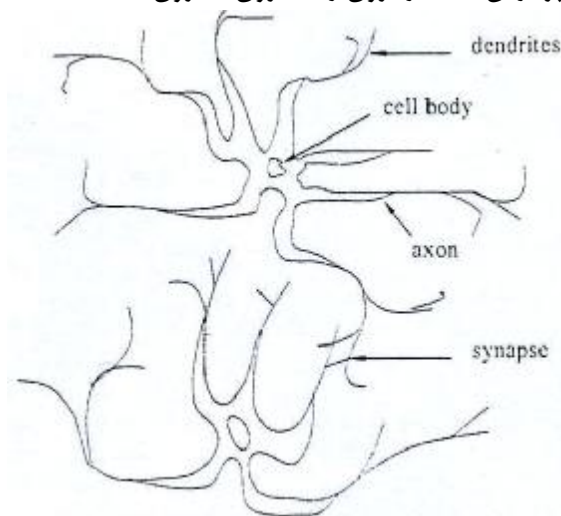
## 2-1 چگونگی پیدایش:

تحقیقات و علاقه مندی به شبکه های عصبی از زمانی شروع شد که مغز به عنوان یک سیستم دینامیکی با ساختار موازی و پردازشگری کاملاً مغایر با پردازشگرهای متداول شناخته شد. نگرش نوین در مورد کارکرد مغز نتیجه تفکراتی بود که در اوایل قرن بیستم توسط رامون سگال در مورد ساختار مغز به عنوان اجتماعی از اجزای محاسباتی کوچک به نام نرون شکل گرفت.

مغز به عنوان یک سیستم پردازشگر اطلاعات با ساختار موازی، از 100 تریلیون ( $10^{11}$ ) نرون به هم مرتبط با تعداد  $10^{16}$  ارتباط تشکیل شده است. نرون ها ساده ترین واحد ساختاری سیستم های عصبی هستند. بافتی که عصب نامیده می شود اجتماعی از نرون ها هستند که اطلاعات و پیام ها را از یک قسمت بدن به قسمت دیگر منتقل می کنند. این پیامها از نوع ایمپالس های الکترو شیمیایی هستند. [1]

میلیون ها نرون در بدن وجود دارند. حتی ساده ترین کارهای روزمره انسان از قبیل پلک زدن تنها از طریق همکاری همه جانبه این نرون ها میسر است. بیشترین تعداد نرون ها در مغز و باقی در نخاع و سیستم های عصبی جانبی تمرکز یافته اند. گرچه همه نرون ها کارکرد یکسانی دارند، ولی اندازه و شکل آنها بستگی به محل استقرار آنها در سیستم عصبی دارد. با وجود این همه تنوع، بیشتر نرون ها از سه قسمت اساسی تشکیل شده اند:

1. بدنه سلول (سوما): که شامل هسته و قسمت های حفاظتی دیگر است.
2. دندریت ها: که اطلاعات را از نرون های قبلی دریافت می کنند.
3. آکسون: اطلاعات پردازش شده در نرون را به نرون یا نرون های بعدی منتقل می کند.



شکل (1) نمایی از نرونهای عصبی مغز را نشان می دهند.

دندریت ها به عنوان مناطق دریافت سیگنال های الکتریکی، شبکه هایی تشکیل یافته از فیبرهای سلولی هستند که دارای سطح نامنظم و شاخه های انشعابی بی شمار می باشند. به همین علت آنها را شبکه های دریافتی درخت گونه می نامند. دندریت ها سیگنال های الکتریکی را دریافت کرده، به هسته سلول منتقل می کنند. بدنه سلول، انرژی لازم را برای فعالیت نرون فراهم می کند. نرون ورودیهایی را که به طریق خاصی جمع می شوند پذیرا می گردد و چنانچه ورودیها به اندازه کافی بزرگ باشند و از آستانه تحریک مشخص نرون فراتر روند، نرون فعال شده و در غیر این صورت غیر فعال باقی می ماند. [3]

یک خاصیت بسیار مهم نرون های بیولوژیکی این است که پس از پتانسیل تحریک اولیه دوباره به وضع سکون باز می گردند و آماده دریافت پتانسیل های تحریک بعدی می شوند.



عمل نرون برروی سیگنال های دریافتی با یک عمل ساده جمع و مقایسه با یک سطح آستانه مدل می گردد. اکسون بر خلاف دندریت ها از سطحی هموارتر و تعداد شاخه های کمتری بر خوردار می باشند. اکسون طول بیشتری دارد و سیگنال الکتروشیمیایی دریافتی از هسته سلول را به نرون های دیگر منتقل می کند. محل تلاقی یک اکسون از یک سلول به دندریت های سلول دیگر را سیناپس می گویند. سیناپس ها واحد های ساختاری کوچکی هستند که ارتباط بین نرون ها را برقرار می سازند. سیناپس ها انواع مختلفی دارند که از مهمترین آنها سیناپس های شیمیایی هستند.

---

## 2-2 ساختار شبکه عصبی:

در شبکه های عصبی مصنوعی نیز بلوکهای ساختاری و یا نرونها، دستگانههای محاسباتی خیلی ساده ای هستند و ارتباط بین نرونها عملکرد شبکه را تعیین می کند و هدف از آموزش شبکه های عصبی مصنوعی تعیین ارتباط مناسب، جهت حل مسائل مختلف است. در واقع شبکه های عصبی مصنوعی نیز با ایده گرفتن از رفتار شبکه عصبی بیولوژیکی شکل یافته است.

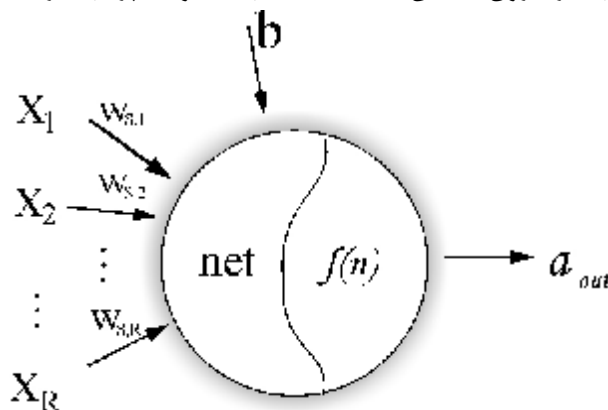
در این شبکه ها نیز عموماً لایه های موازی نرونها را می بینیم که نرونهای هر لایه نقش خاصی را ایفا می کنند. معمولاً ورودیهایی که از خارج به شبکه عصبی داده می شوند، به دسته ای از نرونها که در یک لایه مرتب شده اند، اعمال می شود. این ورودیها با وزنه های خاص جمع شده به تابع تحریک هر نرون فرستاده می شوند که این اقدام به پردازش جمع وزن دار ورودیهای اعمال شده به نرون می کند و خروجی نرون را به لایه های بعدی و نهایتاً به لایه خروجی می فرستد که جواب شبکه نسبت به ورودیهای اعمال شده به آن است.

اگر چه نرونهای بیولوژیکی از نرونهای مصنوعی که توسط مدارهای الکترونیکی ساخته می شوند، بسیار کندتر هستند (یک میلیون بار)، اما عملکرد مغز خیلی سریعتر از عملکرد یک کامپیوتر معمولی است. علت اصلی این پدیده بیشتر به خاطر ساختار کاملاً موازی نرونهای عصبی مغز است و این بدان معنی است که همه نرونها معمولاً به طور همزمان کار می کنند و پاسخ می دهند.

شبکه های عصبی مصنوعی با وجود اینکه با سیستم عصبی طبیعی قابل مقایسه نیستند، ویژگیهایی دارند که آنها را در بعضی از کاربرد ها مانند تفکیک الگو، رباتیک، کنترل و به طور کلی در هر جا که نیاز به یادگیری یک نگاشت خطی و یا غیر خطی باشد ممتاز می نماید. [1]

به بیان دیگر شبکه های عصبی مصنوعی، در واقع نوعی سیستم پردازش اطلاعات هستند که از تعمیم یافتن مدلهای ریاضی شبکه های عصبی انسان بر مبنای فرضیات زیر توسعه یافته اند:

1. داده پردازشی در اجزای ساده ای به نام نرون صورت می گیرد.
2. اطلاعات از طریق ارتباط بین نرونها منتقل می شود.
3. هر ارتباط دارای وزن مخصوص به خویش است.
4. هر نرون برای محاسبه خروجی اش، یک تابع تحریک دارد که به مجموع وزن دار ورودیهایش اعمال می شود.
5. خروجی هر پردازشگر (نرون) ممکن است به تعداد زیادی واحد پردازشگر دیگر منتقل شود. [3]



شکل (2) مدل ریاضی یک نرون را نشان می دهد.

این مدل ریاضی تعریف شده که مشابه دستگاه عصبی مغز کار می کند ساختار یک شبکه عصبی مصنوعی را تشکیل می دهد .

پس یک شبکه عصبی تشکیل شده از چندین نرون عصبی است که به هنگام نیاز فعال شده و محاسبات روی آن انجام می گیرد . به عبارت دیگر این نرونها تشکیل شده از گره ها و پاره خطهای جهت داری هستند که فرآیند حل مسئله در مسیر آنها صورت می گیرد .

گره هایی که در لایه ورودی هستند نرون هایی هستند که هیچ عملیاتی بر روی آنها صورت نمی گیرد و در محاسبه تعداد لایه ها نیز قرار نمی گیرند . گره های لایه خروجی نرون های پاسخ دهنده هستند که پاسخ حل مسئله در آنها نمایان می شود . بین نرون های ورودی و خروجی نیز نرون های پنهان قرار دارند . برای حل یک مسئله با چندین ورودی و خروجی از یک نرون به تنهایی نمی توان استفاده کرد . در این حال باید از چندین نرون به صورت موازی بهره جست که بتوانند بردارهای ورودی را همزمان پردازش کنند و به بردار خروجی لایه آخر منتقل کنند .

درون هر نرون یک وزن به خصوصی است که با اعمال آن به مقادیر ورودی به نرون روی آنان تاثیر گذاشته و بردارهای وزن دار را به توابع تحریک هدایت می کنند . ممکن است نیاز باشد یک بردار علاوه بر آن که وزن آن تغییر کند در فضای برداری جابجا شود که این عمل با اضافه کردن یک بایاس به ماتریس وزن دار صورت می گیرد.

حال مقادیر وزن دار به توابع تحریک برده می شوند و تابع خروجی مقدار اصلی را به خود می گیرد و با توجه به اینکه آیا جواب مناسب بدست آمده است یا خیر این مقادیر با بردار هدف مقایسه می شوند و در صورت اختلاف با بردار هدف به عقب برگشت داده می شوند تا وزن های مناسب تری برای آن بردارها انتخاب گردد .

## 2-2-1 توابع تحریک:

اصولا وقتی مقادیر ورودی به نرون با اعمال ماتریس وزن  $W$  وزن دار شدند ، مقادیر وزن دار توسط توابع تحریک مقدار واقعی خروجی نرون را به خود می گیرند :

$$n = W \times P + b \quad (1)$$

$$a_{out} = f(n) \quad (2)$$

$P$  : بردار ورودی

$a_{out}$  : بردار خروجی

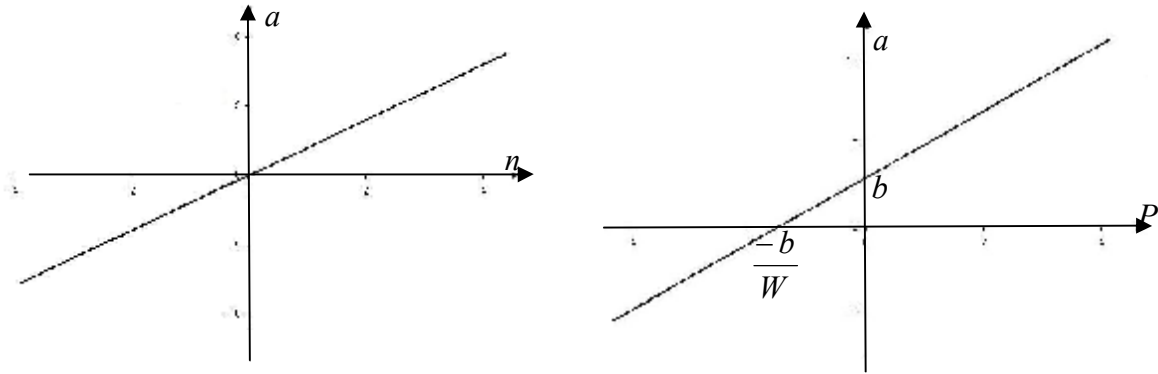
$f$  : تابع تحریک

تابع تحریک بر اساس نیاز خاص حل مسئله انتخاب می شود که در عمل تعداد محدودی از این توابع مورد استفاده قرار می گیرند . توابعی که به خصوص در کاربردهای مهندسی مورد استفاده بیشتری قرار می گیرند ، عبارتند از توابع تحریک خطی ، تابع حدی دو مقداره ( باینری ) ، تابع محرک زیگموئیدی و یا از توابع محرک تانژانت هیپربولیک که در زیر نشان داده شده اند.

1. تابع تحریک خطی: این تابع به صورت زیر تعریف می شود:

$$f(n) = n \quad (3)$$

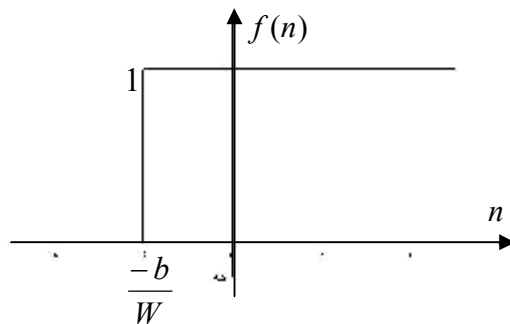
این بدان معنی است که خروجی این تابع برابر ورودی آن است. این تابع بیشتر در شبکه های آدلاین مورد استفاده قرار می گیرند. باید توجه شود که جمله بایاس  $b$  که در معادله فوق نشان داده شد موجب جابجایی منحنی در فضای ورودی می گردد که اهمیت آن را در شکل زیر می بینید.



شکل (3) یک تابع خطی را نشان می دهد

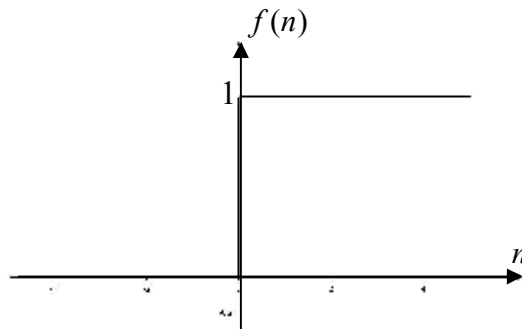
2. **تابع حدی دو مقداری:** خروجی این تابع معمولاً (0 یا 1) و یا (+1 یا -1) است اگر مقدار ورودی وزن دار بزرگتر از  $\frac{-b}{W}$  باشد مقدار تابع +1 و در غیر این صورت مقدار آن -1 و یا صفر (بسته به خواسته مسأله) می باشد. این تابع چون مقادیر ورودی را بین دو مقدار محدود می کنند در مسائلی به کار برده می شوند که بخواهیم داده ها را به طور خطی از هم جدا کنیم (مانند تابع متحرک خطی)

$$f(n) = \begin{cases} 0 & n < \frac{-b}{w} \\ 1 & n > \frac{-b}{w} \end{cases} \quad (4)$$



شکل (4) نمودار یک تابع باینری را نشان می دهد

$$f(n) = \begin{cases} 0 & n < 0 \\ 1 & n > 0 \end{cases} \quad (5)$$

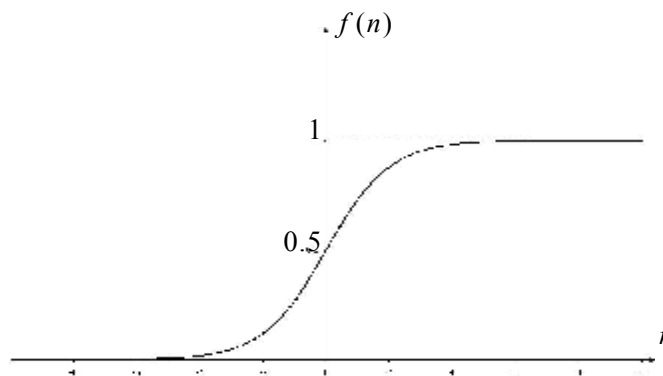


شکل (5) نمودار یک تابع باینری با عرض از مبدا صفر را نشان می دهد

3. **تابع محرک سیگموئیدی**: این تابع نیز مقادیر ورودی را بین صفر و یک محدود می کند با این تفاوت که میزان تغییرات آن غیر خطی است. مزیت این تابع در عملکرد آن با ورودیهای بزرگ و یا کوچک است.

هنگامی که مقادیر ورودی به سمت اعداد منفی بزرگ میل کند مقدار صفر و هنگامی که مقادیر ورودی به سمت صفر میل کند مقدار تابع به سمت 0.5 و هنگامی که مقادیر ورودی به سمت مقادیر مثبت بزرگ میل کند مقدار تابع به سمت یک میل خواهد کرد. فرم این تابع مطابق شکل زیر بوده و به صورت رابطه (6) تعریف می شود.

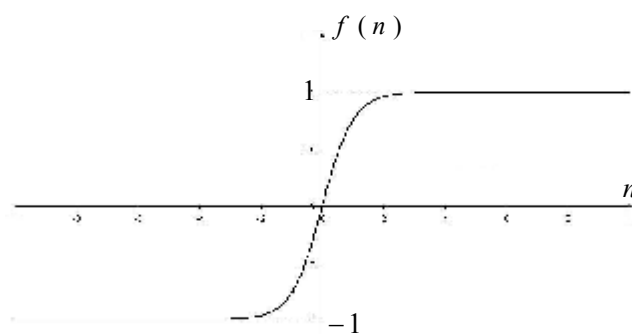
$$f(n) = sig(n) = \frac{1}{1 + e^{-n}} \quad (6)$$



شکل (6) نمودار یک تابع سیگموئیدی را نشان می دهد

4. **تابع تحریک تانژانت هیپربولیک**: تابع تحریک دیگر که اغلب توسط بیولوژیستها به عنوان مدل ریاضی از تحریک سلول عصبی مورد استفاده قرار می گیرد، تابع تانژانت هیپربولیک است که از نظر شکل شبیه تابع منطقی است:

$$f(n) = \tanh(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (7)$$



شکل (7) نمودار یک تابع تانژانت هیپربولیک را نشان می دهد

## 2-3-3 توانایی های شبکه عصبی:

### 2-3-3-1 قابلیت یادگیری:

استخراج نتایج تحلیلی از یک نگاشت غیر خطی که با چند مثال مشخص شده است، کار ساده ای نیست. نرون یک دستگاه غیر خطی است و در نتیجه یک شبکه عصبی که از اجتماع این نرونها تشکیل می شود نیز یک سیستم کاملاً پیچیده و غیر خطی خواهد بود. به علاوه خاصیت غیر خطی عناصر پردازش در کل شبکه توزیع می گردد. پیاده سازی یک سیستم با یک نگاشت غیر خطی و یا خطی پیچیده به وسیله یک الگوریتم معمولی و بدون قابلیت یادگیری نیاز به دقت و مراقبت زیادی دارد. در چنین حالتی سیستمی که بتواند خود این رابطه را استخراج کند، بسیار سودمند به نظر می رسد. خصوصاً افزودن مثالهای احتمالی در آینده به یک سیستم با قابلیت یادگیری، به مراتب آسان تر از انجام آن در یک سیستم بدون چنین قابلیت است. قابلیت یادگیری یعنی توانایی تنظیم پارامترهای شبکه (وزنهای سیناپسی)، هنگامی که محیط شبکه تغییر می کند و شبکه شرایط جدید را تجربه می کند. با این هدف که اگر شبکه برای یک وضعیت خاص آموزش دید و تغییر کوچکی در شرایط محیطی آن (وضعیت خاص) رخ داد، شبکه بتواند با آموزش مختصر برای شرایط جدید کارآمد باشد. دیگر اینکه اطلاعات در شبکه های عصبی در سیناپس ها ذخیره می گردد و هر نرون در شبکه به صورت بالقوه از کل فعالیت سایر نرونها متأثر می شود. در نتیجه اطلاعات از نوع مجزا از هم نبوده بلکه متأثر از کل شبکه است.

آنچه که شبکه عصبی فرا می گیرد (اطلاعات و یا دانش)، در وزنهای سیناپسی مستتر می باشد. رابطه یک به یک بین ورودیها و وزنهای سیناپسی وجود ندارد. می توان گفت که هر وزن سیناپسی مربوط به همه ورودیهاست ولی به هیچ یک از آنها به طور منفرد و مجزا مربوط نیست. به عبارت دیگر هر نرون در شبکه، از کل فعالیت سایر نرونها متأثر می باشد. در نتیجه اطلاعات توسط شبکه های عصبی پردازش می شوند. بر این اساس، چنانچه بخشی از سلولهای شبکه حذف شوند و یا عملکرد غلط داشته باشند باز هم احتمال رسیدن به پاسخ صحیح وجود دارد. اگر چه این احتمال برای تمام ورودیها کاهش یافته، ولی برای هیچ یک از بین نرفته است.

### 2-3-3-2 قابلیت تعمیم دهی:

پس از این که مثالهای اولیه به شبکه آموزش داده شد، شبکه می تواند در مقابل یک ورودی آموزش داده نشده قرار گیرد و یک خروجی مناسب ارائه نماید. این خروجی بر اساس یک مکانیزم تعمیم که همان درون یابی است، بدست می آید. به عبارت روشنتر، شبکه تابع نگاشت بین دسته های ورودی و خروجی را بدون این که دقیقاً بداند چیست، یاد می گیرد و الگوریتم را می آموزد و یا رابطه تحلیلی مناسب را برای تعدادی نقاط در فضا بدست می آورد. [1]

### 2-3-3-3 مقاوم بودن شبکه در مقابل خطا:

در یک شبکه عصبی مصنوعی هر سلول به طور مستقل عمل می کند و رفتار کلی شبکه برآیند رفتارهای محلی سلولهای متعدد است. این ویژگی باعث می شود تا خطاهای محلی از چشم خروجی نهایی دور بماند. به عبارت دیگر سلولها در یک روند همکاری، خطاهای محلی یکدیگر را تصحیح می کنند. این خصوصیت باعث افزایش قابلیت مقاوم بودن (تحمل پذیری خطاها) در سیستم می گردد. [2]

## 2-4 تاریخچه شبکه های عصبی مصنوعی:

الف- شروع شبکه عصبی در سال 1943 میلادی توسط مک کلاخ (Mc Culloch) و پیتز (Pitts) صورت گرفت و اولین شبکه عصبی با بکارگیری چند نرون ساده تهیه گردید که قدرت محاسباتی قابل توجهی داشت

ب- در سال 1969 میلادی هب (Hebb) اولین قانون آموزش برای شبکه های عصبی را مطرح نمود. وی پیشنهاد نمود که اگر دو نرون همزمان فعال باشند در آن صورت باید مقدار اثر ارتباط بین آنها افزایش یابد.

ج- دهه های 50 و 60 میلادی سالهای طلایی شبکه های عصبی می باشند. در محدوده سالهای 1958 تا 1962 روزن بلات [ Rosen Blat ] گروه بزرگی از شبکه های عصبی به نام پرسپترون را معرفی نمود. قانون آموزش این شبکه ها یک روش تکراری اصلاح وزن است که بسیار قوی تر از قانون هب عمل می کند

در سال 1960 میلادی شبکه آدالاین (Adaline) توسط ویدرو (Widrow) ارائه گردید. کاربردهای متعدد این شبکه در تشخیص تصویر، آنرا به ابزاری مورد علاقه در کنترل تبدیل کرده است.

د- در سال های 1970 میلادی تحقیقات فراگیری توسط کوهنن (Kohonen) در مورد شبکه هایی با حافظه مشارکتی انجام شد. این شبکه در تشخیص صدا به کار گرفته شد. اندرسن (Anderson) نیز تحقیقاتی در مورد شبکه پیشنهادی کوهنن انجام داد و راهبرد این تحقیقات، شبکه های عصبی است که امروزه در تشخیص طبی به کار می روند.

گراسبرگ (Grossberg) 1970 نیز تحقیقات متعددی ارائه کرد که اغلب روابط ریاضی حاکم بر شبکه های عصبی را مطرح می کند. ایشان به اتفاق کارپنتر (Carpenter) تئوری مربوط به شبکه های عصبی را که قادر به مرتب کردن و طبقه بندی نمودن اطلاعات ورودی هستند را بنا نموده اند.

ه- پیشرفت شبکه های عصبی در دهه 80 میلادی تکامل یافت. اصولاً شبکه های تک لایه کارایی چندانی در حل مسائلی که نگاشت فضای ورودی به خروجی آنها توابع غیر خطی طلب می کرد نداشتند و این دلیل عمده عدم پیشرفت شبکه های عصبی بود. اما با ابداع روش انتشار برگشتی که مستقلاً توسط پارکر (Parker) و لوکان (Lecun) ارائه گردید، تحولی در شبکه های عصبی صورت گرفت. از سایر کسانی که در پیشبرد این علم سهیم بوده اند، هاپفیلد (برنده جایزه نوبل فیزیک) بود. او شبکه های عصبی را که بر اساس وزن ثابت عمل می نمایند برای اولین بار معرفی نمود. این شبکه ها به صورت حافظه مشارکتی عمل نموده و امکان حل مسائلی با قیدهای اولیه توسط آنها میسر می گردد.

یکی از دلایل عمده عدم توجه به شبکه های عصبی در محدوده سالهای 1970 میلادی به واسطه محدودیت های موجود در شبکه عصبی تک لایه بود. پیدایش شبکه های عصبی چند لایه توسط رومل هارت (Rumelhart) در سال 1986 میلادی، نقش مهمی در احیای شبکه های عصبی برای حل محدوده وسیعی از مسائل ایفا نمودند. شبکه عصبی ارائه شده بر اساس روش کاهش گرادیان همانند یک بهینه ساز عمل نموده و مجموع مجزورات خطا در نرون های خروجی شبکه را به حداقل می رساند.



## 2-5 آموزش شبکه های عصبی:

هدف استفاده از شبکه عصبی مصنوعی برای حل یک مسئله ، بدست آوردن خروجی مناسب با توجه به داده های ورودی است . حال اینکه مقدار خروجی بدست آمده چقدر با مقدار واقعی آن اختلاف دارد ، بستگی به آموزش شبکه و انتخاب وزنها و بایاسهای مناسب برای لایه ها دارد نحوه یادگیری در شبکه های عصبی به دو صورت انجام می گیرد .

2-5-1 **آموزش با ناظر:** معمولا برای آموزش شبکه مجموعه زوج نیروهای ورودی و خروجی به آن اعمال می شود . هر کدام از این دسته های ورودی و خروجی را می توان به صورت یک بردار در نظر گرفت .

سپس شبکه با اعمال تنظیم وزندهای شبکه آموزش می بیند . طی آموزش شبکه ، وزندهای شبکه به تدریج به مقادیری همگرا می شوند که به ازای آنها با اعمال یک بردار ورودی ، بردار خروجی دلخواه تولید می گردد . این گونه آموزش را آموزش با ناظر می نامیم .

برخی شبکه های عصبی آموزش می بینند که بردارهای ورودی را گروه بندی کنند در این صورت اگر بردار ورودی به گروه A تعلق داشته باشد خروجی برابر یک و در غیر این صورت مقدار صفر به خود می گیرد . در مواردی دیگر با ارائه بردار ورودی ، پاسخ شبکه بر اساس شبکه آموزش دیده به صورت نزدیکترین بردار خروجی مطلوب بدست می آید .

شبکه های چند لایه قادرند که با اعمال نگاشتهای غیر خطی از بردارهای ورودی با n بعد ، بردارهای خروجی m بعدی را نتیجه دهند .

2-5-2 **آموزش بدون ناظر:** در این نوع آموزش فقط بردارهای ورودی به شبکه داده می شود و جواب مطلوب بر شبکه اعمال نمی شود . در این حالت شبکه رابطه بین ورودیها را پیدا می کند و گروه بندی کرده در خروجی کد می کند و از اقلام یادگیری بدون ناظر، می توان از قانون یادگیری شبکه های عصبی رقابتی نام برد. در این نوع شبکه ها خروجیها با هم به رقابت پرداخته تا کدامیک موفق به قرارگیری در الگوی مناسب و تعیین شده در شبکه قرار بگیرند . به این گونه الگوریتم یادگیری ، الگوریتم خود سازمانده نیز می گویند .

به علت سرعت قابل بوجهی که این نوع شبکه در هنگام آموزش نسبت به آموزش با ناظر دارند این شبکه ها را برای حل مسائل عملی متمایز کرده است .

## فصل سوم

### انواع شبکه های عصبی

---

## 3-1 شبکه های عصبی پرسپترون:

یکی از مهمترین شبکه های عصبی که در مهندسی جایگاه خاصی دارد و از زمره کاربردی ترین شبکه ها به شمار می رود، شبکه های پرسپترون هستند. این شبکه ها قادرند با انتخاب مناسب تعداد لایه ها و سلولهای عصبی، یک نگاشت غیر خطی را به دقت انجام دهند که این همان خواسته بسیاری از مسائل فنی مهندسی است که می تواند توسط این شبکه ها مطرح شود. پرسپترون یک شبکه غیر بازگشتی است که از یک الگوریتم آموزش با ناظر بهره می گیرد. بنابراین، دسته های آموزشی آن شامل مجموعه ای از بردارهای ورودی همراه با بردارهای هدف دلخواهشان است. در این شبکه بردارهای ورودی شامل حدود پیوسته ای از مقادیر است اما بردارهای هدف شامل اعداد باینری یعنی صفر و یک هستند که بعد از آموزش تولید می شوند.

$$a_{out} = f(W \times P + b) \quad (8)$$

$$f\left(\begin{bmatrix} w_{11} & w_{12} & \dots & w_{1R} \\ w_{21} & \cdot & & \vdots \\ \vdots & & \cdot & \vdots \\ w_{1R} & \dots & \dots & w_{SR} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_r \end{bmatrix}\right) = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_S \end{bmatrix} \quad (9)$$

برای آموزش شبکه پرسپترون تک لایه (SLPR) Single Layer Perceptron زوج بردارهای آموزشی شامل بردارهای ورودی و بردارهای هدف به شبکه داده می شود. سپس ماتریس وزن با درایه های کوچکی برای همه وزنهای شبکه به صورت تصادفی فرض می شود. بردار ورودی در ماتریس وزن ضرب شده و تابع تحریک حدی باینری (Sign) به این بردارهای وزن دار اعمال می شود.

$$a_j = \begin{cases} 1 & \sum (WP + b) > \theta_j \\ 0 & \sum (WP + b) \leq \theta_j \end{cases} \quad (10)$$

$\theta_j$ : مقدار حدی مربوط به سلول عصبی  $j$  است.

اکنون می توانیم مقدار خطا را برای هر سلول عصبی را به صورت اختلاف خروجی عملی و خروجی هدف محاسبه نماییم.

$$error_i = t_i - a_i \quad (11)$$

با داشتن مقدار خطا وزنها اصلاح می گردند و این عمل اصلاح وزنها آنقدر تکرار می شود تا مقدار خطای شبکه به حد قابل قبول برسد.

$$W_{ij}(K+1) = W_{ij}(k) + \alpha \times P_i \times error_j \quad (12)$$

که در این رابطه:

$\alpha$ : مقدار ثابت مثبت کوچکتر از یک است که نرخ یادگیری را تعیین می کند.

$P_i$ : بردار داده های ورودی است.

الگوریتم ارائه شده برای شبکه های پرسپترون تک لایه ارائه شده است. شبکه های تک لایه از این مشکل اساسی برخوردارند که تنها توانایی حل آن دسته از مسائل طبقه بندی را دارند که به طور خطی از هم مستقلند و از حل بسیاری از مسائل و طبقه بندی الگوهایی که در فضای ورودی به طور خطی از هم جداناپذیرند ناتوان می باشد.

این ناتوانی ها توسط الگوریتم یادگیری پس از انتشار خطا (BP) برای شبکه های عصبی پرسپترون چند لایه مرتفع می شود.

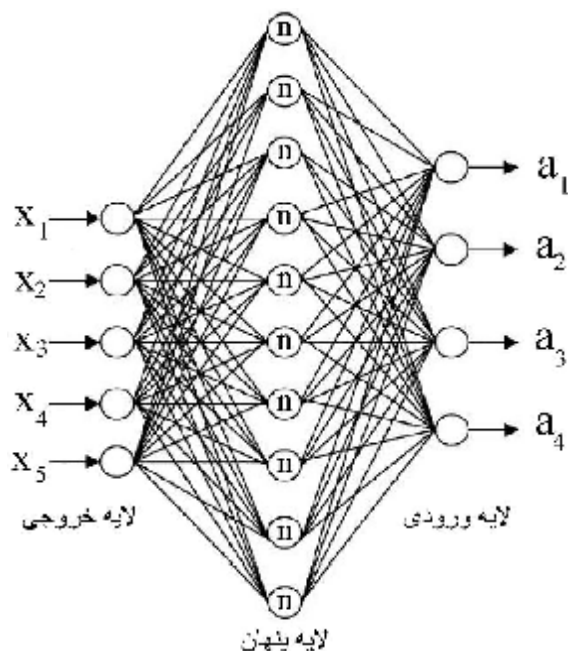
### 3-1-1 شبکه های پرسپترون چند لایه (Multi Layer Perceptron):

در شبکه های چند لایه، لایه ها به ترتیب به هم متصل می شوند به گونه ای که خروجی های لایه اول، ورودیهای لایه دوم و به همین ترتیب تا آخر که خروجی های لایه آخر خروجیهای اصلی و پاسخ واقعی شبکه را تشکیل می دهند. به عبارتی دیگر جریان سیگنال شبکه در یک مسیر پیش خور صورت می گیرد که از لایه ورودی شروع شده و به لایه خروجی ختم می گردد.

عموماً در شبکه های چند لایه پرسپترون (MLP) دو نوع سیگنال استفاده می شود که با هم متفاوتند. دسته اول سیگنالهای تابعی که بر اساس ورودیهای هر نرون و پارامترهای وزن و تابع محرک نظیرش محاسبه می شوند و دسته دوم سیگنال های خطا که با برگشت از لایه خروجی و منشعب شدن به لایه های پنهان دیگر محاسبه می شوند.

تعداد نرونهای لایه پنهان بستگی به نظر طراح شبکه دارد و با سعی و خطا بدست می آید. در صورت کافی نبودن تعداد نرونها، شبکه قادر نخواهد بود نداشت دقیقی بین بردارهای ورودی و خروجی ایجاد کند.

در خروجی هر نرون از شبکه MLP یک تابع خطی قرار دارد و فرایند یاد گیری در تمام نرونها و لایه ها صورت می گیرد. همه وزنها و بایاسهایی که در شبکه قرار دارند در طول فرایند یادگیری قابل تغییر هستند.



شکل (8) شمای کلی یک شبکه عصبی دو لایه (با یک لایه پنهان) را نشان می دهد.

## 3-1-2 قانون یادگیری پس انتشار خطا (BP):

الگوریتم پس انتشار خطا، الگوریتم یادگیری است که برای شبکه های چند لایه پرسپترون به کار می رود. در این روش از آموزش با ناظر استفاده می شود بدین صورت که بردار خروجی مطلوب (target) به عنوان ناظر در نظر گرفته می شود و مقدار خروجی شبکه ( $a_{out}$ ) در طول فرایند یادگیری شبکه به بردار مطلوب نزدیک می گردد.

قانون پس انتشار خطا (BP) از دو مسیر اصلی تشکیل می شود. مسیر اول به مسیر رفت موسوم است. در این مسیر بردار ورودی به شبکه چند لایه پرسپترون (MLP) اعمال می شود و تأثیراتش از طریق لایه میانی یا همان لایه های پنهان به لایه های خروجی انتشار می یابد و توابع تحریک روی تک تک نرونهای هر لایه عمل می کنند. در طول این مسیر پارامترهای شبکه ثابتند و بدون تغییر باقی می مانند. این مسیر با معادلات زیر بیان می شود:

$$a^{i+1}(K) = f^{i+1}(W^{i+1}(K) \times a^i + b^{i+1}(K)) \quad i = 0, 1, \dots, L-1 \quad (13)$$

$i$ : شماره لایه در حال محاسبه می باشد که  $a^i$  به عنوان خروجی لایه قبلی برای محاسبات لایه  $i+1$  ام به عنوان ورودی عمل می کند.

در مسیر دوم یا همان مسیر برگشت، بر عکس مسیر رفت پارامترهای شبکه تغییر و تنظیم می گردند. این تغییرات بر اساس پروسه اصلاح خطایی که سیگنال خطا در لایه خروجی تشکیل داده است انجام می شوند. همان طور که می دانید، بردار خطا، اختلاف بین بردار پاسخ مطلوب و پاسخ واقعی شبکه می باشد که بعد از محاسبه در مسیر برگشت از لایه خروجی به سمت لایه های پیشین و از طریق لایه های شبکه در کل شبکه توزیع می گردد.

هنگامی که بردار خروجی با بردار مطلوب مقایسه می شود مقدار خطا در خروجی نرون  $n$  ام از لایه آخر برای  $K$  امین الگو به صورت زیر تعریف می شود:

$$e_n(K) = t_n(K) - a_n(K) \quad (14)$$

بنابر این می توانیم مقدار لحظه ای خطا را برای نرون  $n$  ام از لایه خروجی، به صورت  $e_n^2(K)$  تعریف کنیم. همین طور میزان خطای شبکه را با شاخص زیر که برابر مجموع مربعات خطا است، مشاهده کنیم:

$$\hat{F}(K) = \sum_{j=1}^n e_j^2(K) \quad (15)$$

و برای این که بدانیم، تا چه حد شبکه MLP آموزش دیده است و تابع  $\hat{F}_{avr}$  را که تقریبی از  $\hat{F}$  می باشد، تعریف می کنیم:

$$\hat{F}_{avr}(K) = \frac{1}{n} \sum_{j=1}^n \hat{F}(j) \quad (16)$$

در شبکه های چند لایه پرسپترون، بر خلاف شبکه های تک لایه آن، مدل هر نرون دارای یک تابع تحریک غیر خطی مشتق پذیر باید باشد زیرا برای محاسبه خطا از مشتق تابع محرک استفاده می شود. از آنجا که مشتق توابع زیگموئیدی و تانژانت هیپربولیک به راحتی به دست می آیند می توان در MLP از این توابع استفاده کرد.

$$f_{sig}(n) = \frac{1}{1 + e^{-n}} \quad (17)$$

$$f'_{sig}(n) = f(n) \cdot (1 - f(n)) \quad (18)$$

$$f_{\tanh}(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (19)$$

$$f'_{\tanh}(n) = 1 - f^2(n) \quad (20)$$

مشتق تابع محرک در  $\frac{\partial \hat{F}(K)}{\partial W_n(K)}$  و  $\frac{\partial \hat{F}(K)}{\partial b_n(K)}$  نمایان می شود به قسمی که با داشتن پارامترهای شبکه در هر لایه  $i$  می توانیم بنویسیم:

$$\frac{\partial \hat{F}(K)}{\partial W^i(K)} = \frac{\partial \hat{F}(K)}{\partial e(K)} \cdot \frac{\partial e(K)}{\partial a^i(K)} \cdot \frac{\partial a^i(K)}{\partial n^i(K)} \cdot \frac{\partial n^i(K)}{\partial W^i(K)} \quad (21)$$

$$\frac{\partial \hat{F}(K)}{\partial b^i(K)} = \frac{\partial \hat{F}(K)}{\partial e(K)} \cdot \frac{\partial e(K)}{\partial a^i(K)} \cdot \frac{\partial a^i(K)}{\partial n^i(K)} \cdot \frac{\partial n^i(K)}{\partial b^i(K)} \quad (22)$$

که در آن:

$$\frac{\partial a^i(K)}{\partial n^i(K)} = f'(n^i(K)) \quad (23)$$

$$\frac{\partial \hat{F}(K)}{\partial e(K)} = 2e(K) \quad (24)$$

$$\frac{\partial e(K)}{\partial a^i(K)} = -1 \quad (25)$$

$$\frac{\partial n^i(K)}{\partial b^i(K)} = 1 \quad \text{و} \quad \frac{\partial n^i(K)}{\partial W^i(K)} = a^{i-1}(K) \quad (26)$$

حال پارامترهای شبکه MLP را می توانیم در هر لایه و برای هر نرون از آن لایه اصلاح کنیم:

$$W_n(K+1) = W_n(k) - \alpha \times \frac{\partial \hat{F}(K)}{\partial W_n(K)} = W_n(K) - \alpha \times \Delta W_n(K) \quad (1-27)$$

$$b_n(K+1) = b_n(k) - \alpha \times \frac{\partial \hat{F}(K)}{\partial b_n(K)} = b_n(K) - \alpha \times \Delta b_n(K) \quad (2-27)$$

نحوه ارائه داده های یادگیری برای آموزش شبکه مهم است. ترتیب ارائه نمونه ها به شبکه بایستی طوری باشد که شبکه از امکان انتخاب و آموزش برابری نسبت به نمونه ها برخوردار باشد و بدین جهت باید نمونه ها را به صورت تصادفی به شبکه اعمال کنیم.

### 3-2 شبکه آدلاین (Adaline):

ویدرو در سال 1960 شبکه آدلاین را با قانون یادگیری ویدرو-هوف معروف به قانون LMS (Least Mean Square) مطرح کرد .

شبکه آدلاین بسیار شبیه شبکه پرسپترون تک لایه است که توضیح آن داده شد ، با این تفاوت که بر خلاف پرسپترون تک لایه که دارای تابع تحریک باینری است (دو مقداره بدون شیب) ، تابع تحریک آدلاین یک تابع تحریک خطی می باشد .

همانطور که گفته شد که شبکه تک لایه پرسپترون فقط الگوهای خطی مستقل از هم را طبقه بندی می کنند ، شبکه آدلاین نیز این محدودیت را دارد و فقط در مسائلی به کار می رود که به طور خطی از هم متمایزند . قانون LMS که پارامترهای شبکه آدلاین را تنظیم می کند ، همانند قانون SLPR و حتی قوی تر از آن یک خط مرزی برای جداسازی الگوهای یاد گیری قرار می دهد وهمگرایی به یک جواب را در صورت وجود تضمین می کند . این کارایی ، کاربرد شبکه آدلاین در مسائل مهندسی را بیش از شبکه تک لایه پرسپترون افزایش داده است . [1]

شبکه آدلاین به گونه ای عمل می کند که رابطه های خطی بین بردارهای ورودی و بردارهای خروجی را آموزش می بیند . پس کارایی آن تنها در بعضی مسائل دیده می شود [1] و به دلیل اینکه فقط توانایی جداسازی خطی را دارد ، در مهندسی عمران از این روش کمتر استفاده می شود .

### 3-3 شبکه عصبی انتشار متقابل (CPN) Counter Propagation Network:

این شبکه همانند یک جدول مرجع عمل می کند. بدین معنی که ارتباط بین لایه های وسط و لایه خروجی وزن هایی به خود می گیرند که مشخص کننده مقدار خروجی شبکه در ازای هر یک از ورودی های شبکه است. آموزش این شبکه با زوج های آموزشی صورت می گیرد که این بردارها می توانند به صورت پیوسته بین (0 و 1) تغییر کنند. این شبکه مانند سایر شبکه ها در وهله اول برای منظور خاصی طراحی می شود سپس برای حل مسائل مشابه مورد استفاده قرار می گیرد.

جدول مرجع در این شبکه بدین صورت عمل می کند که ورودی  $X$  با تمامی بردارهای وزن  $W$  مقایسه شده و نزدیکترین بردار  $W_i$  انتخاب می شود سپس شبکه بردار خروجی مربوط به این بردار را به عنوان خروجی ارسال می کند. آموزش شبکه در دو مرحله صورت می گیرد. در مرحله اول لایه رقابتی موسوم به کوهونن، وزن های لایه رقابتی بدون هیچ گونه ناظری اصلاح می شوند به طوریکه بردارهای شبیه هم نرون خاص خود را فعال می کنند. سپس لایه واسطه ای (گراسبرگ) خروجی مناسب مربوط به بردار وزن  $W$  را ایجاد می کند.

#### 3-3-1 آموزش لایه اول (رقابتی یا کوهونن):

یک بردار ورودی برای آموزش به این لایه اعمال می شود و سپس حاصلضرب آن بردار در بردار وزن مربوط به هر نرون لایه رقابتی محاسبه می شود و بزرگترین مقدار آن انتخاب شده و وزن آن اصلاح می گردد. در واقع بردار وزن طوری اصلاح می گردد که شبیه ترین بردار به بردار ورودی شود و جهت تغییر به گونه ای است که اختلاف بین بردار وزن و ورودی حداقل شود.

معادله اصلاح وزن به صورت زیر است:

$$W_{ij}(new) = W_{ij}(old) + \alpha(X_i - W_{ij}(old)) \quad (29)$$

که در آن  $\alpha$  ضریب سرعت آموزش در مرحله آموزش شبکه متغیر است.

در لایه کوهونن بردارهای ورودی گروه بندی می شوند که این گروه بندی بر اساس فاصله اقلیدسی صورت می گیرد که به صورت زیر تعریف می شود و معیار نزدیکی و همگرایی بردار وزن به بردار ورودی است.

$$D = \sum_{i=1}^n |X_i - W_i|$$

معمولاً تعداد زوج های آموزشی زیاد هستند و بسیاری از بردارهای ورودی شبیه هم هستند. پس باید شبکه به صورتی آموزش ببیند که بردارهای مشابه یک نرون را در لایه رقابتی فعال کنند.

#### 3-3-2 آموزش لایه دوم (واسطه ای یا گراسبرگ):

آموزش لایه دوم با نظارت صورت می گیرد. در این لایه خروجی های لایه رقابتی به عنوان ورودی اعمال شده و خروجی های لایه واسطه ای با یک محاسبه معمولی حاصل می گردند. در واقع این لایه خروجی های طبقه بندی شده لایه رقابتی را بر خروجی واقعی نگاشت می کنند.

میزان اصلاح وزن ( $V$ ) متناسب با اختلاف بین وزن و خروجی واقعی لایه دوم است. یعنی:

$$V_{jk}(new) = V_{jk}(old) + b(Y_k - V_{jk}(old))Z_j \quad (30)$$

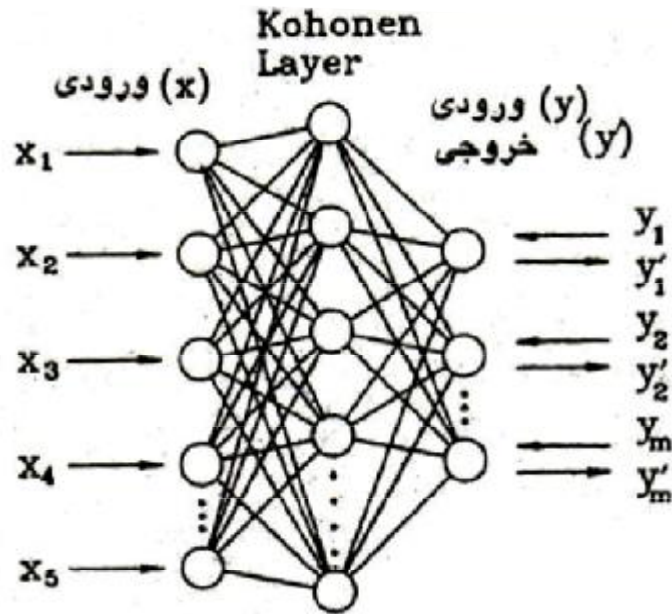
به طوریکه:

$Z_j$ : خروجی لایه رقابتی  $j$  است.



$y_k$ : مولفه  $k$  از بردار خروجی واقعی است .

مقدار  $b$  را معمولا 0/1 انتخاب نموده و در طول پیشرفت آموزش شبکه مقدار آن را کاهش می دهند .

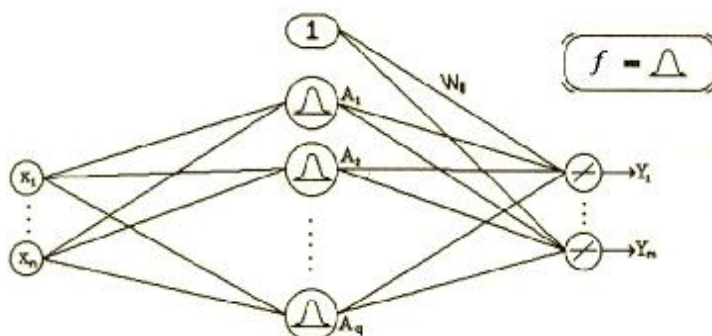


شکل (9) توپولوژی شبکه انتشار متقابل

### 3-4 شبکه RBF (Radial Basis Function)

بر خلاف شبکه MLP که دارای توابع تحریک سیگموئید یا تانژانت هیپربولیک می باشد، در شبکه RBF تابع تحریک یک تابع غیرخطی گوسی است.

در این شبکه ارتباط بین نرون های لایه های ورودی و پنهان به سادگی MLP نمی باشد. نرون های لایه مخفی، واحدهای چند بعدی هستند و بعد این نرون ها برابر تعداد ورودی شبکه است. به طور ساده می توان گفت، به علت شکل خاص تابع گوسی، هر نرون لایه مخفی هنگامی دارای خروجی بزرگتر است که بردار ورودی شبکه، به مرکز تابع گوسی آن نرون نزدیکتر باشد و با افزایش فاصله بردار ورودی از مرکز این تابع غیرخطی، خروجی نرون نیز کاهش می یابد. این فاصله را می توانیم با معیار فاصله اقلیدسی تعریف کنیم. ساختار ساده این شبکه ها در شکل زیر نمایش داده شده است:

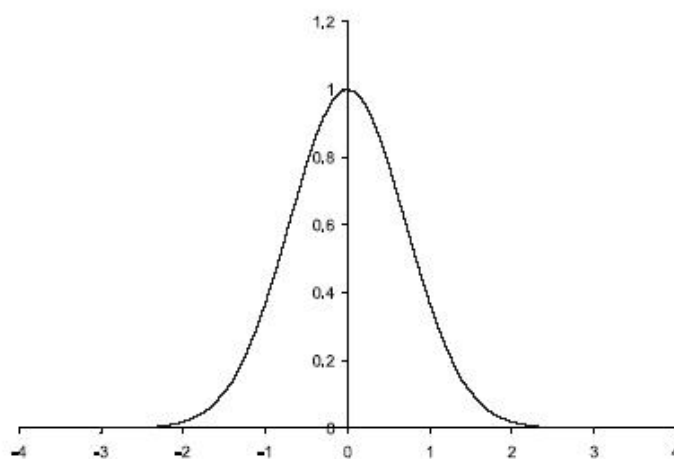


شکل (10) ساختار شبکه RBF با تابع محرک گوسی

یک تابع گوسی به صورت زیر تعریف می شود:

$$f_n = e^{-\left(\frac{X_i - c_{ij}}{d_{ij}}\right)^2} \quad (31)$$

که در آن  $X_i$ ،  $i$  امین درایه بردار ورودی است.  $c_{ij}$  مرکز تقارن و  $d_{ij}$  عرض  $i$  امین تابع گوسی از  $j$  امین نرون مخفی می باشد.



شکل (11) نمودار تابع گوسی را نشان می دهد

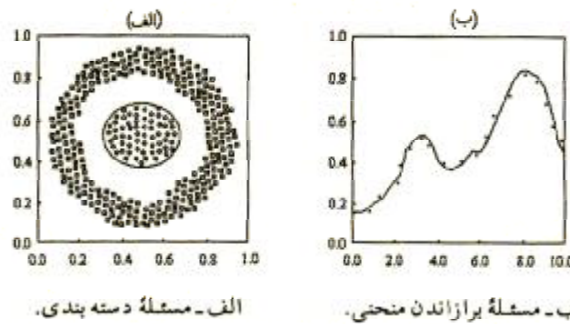
خروجی نرون  $j$  ام از لایه مخفی، از حاصل ضرب خروجی توابع گوسی مربوط بدست می آید:

$$A_j = \prod_{i=1}^N f_n \quad (32)$$

خروجی شبکه RBF نیز، تابع خطی از خروجی نرون های لایه مخفی است.

$$Y_k = \sum_j W_{kj} \cdot A_j \quad (33)$$

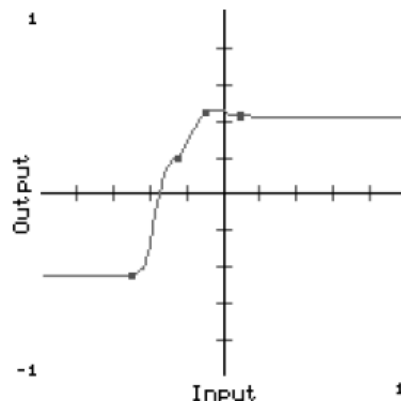
شبکه RBF، مانند شبکه MLP قابلیت پردازش هر نوع نگاشت غیر خطی بین اطلاعات ورودی و خروجی را دارد تفاوت این دو در اختلاف بین توابع تحریک آنها است، به طوری که خروجی نرون های لایه مخفی در MLP در محدوده گسترده تری از ورودی ها غیر صفر است ولی در شبکه های RBF نرون در محدوده کوچکی دارای خروجی غیر صفر هستند و به صورت محلی عمل می کنند. بنابراین شبکه های RBF اغلب در مسائل طبقه بندی نقاط بهتر از MLP عمل می کند در صورتیکه شبکه MLP در برارزش منحنی مناسب تر است.



شکل (12)

تابع دیگری به نام گوسی اصلاح شده در شبکه های RBF استفاده می شود. این شبکه ها Normalized RBF نامیده می شوند که نتایج بهتری نسبت به RBF ارائه می دهند. Normalized RBF نیز مانند شبکه RBF عمل می کند با این تفاوت که تابع تحریک گوسی آن توسط فعالیت کلی  $m$  نرون لایه پنهان نرمال شده است. یک تابع گوسی اصلاح شده به صورت زیر تعریف می شود:

$$Gn_i(\bar{x}) = \frac{\exp\left(-\frac{\sum_{j=1}^n (x_j - w_{ij})^2}{2\sigma^2}\right)}{\sum_{k=1}^m Gn_k(\bar{x})} \quad (34)$$



شکل (13) نمودار یک تابع گوسی نرمال شده را نشان می دهد.

### 3-5 مدل دینامیکی عصبی عادل و پارک :

این شبکه عصبی توسط آقای عادل و پارک مخصوص مسائل بهینه سازی در سازه ها ارائه شده است که ساختار آن شامل یک لایه از متغیر ها است که در آن متغیرهای طراحی داده می شود و نیز شامل یک لایه قید ها است که از تابع های تساوی ها و نامساوی های مطرح در بهینه سازی تشکیل شده است .

در این شبکه برای هر یک از گره های لایه متغیر 5 لینک وجود دارد :

1. یک لینک تحریک که گره  $i$  ام از لایه متغیر را به گره  $j$  ام از لایه قید وصل می کند و گره  $j$  ام را

فعال می کند .  $\longrightarrow$

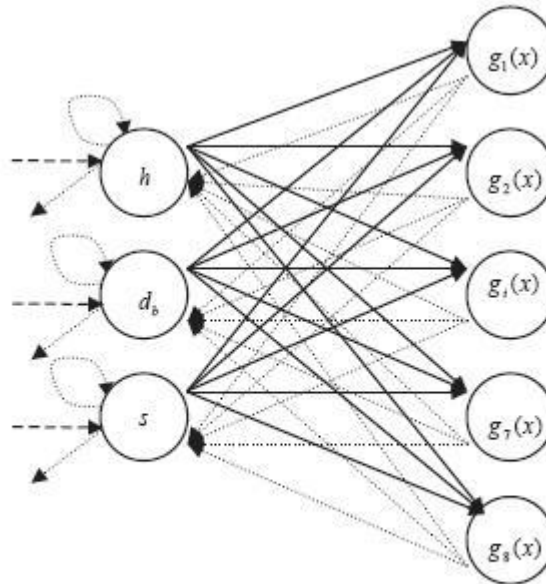
2. یک لینک ورودی که شامل مقدار طراحی در  $i$  امین گره در مرحله  $n-1$  ام می باشد .  $\dashrightarrow$

3. یک لینک خروجی که شامل مقدار طراحی در  $i$  امین گره در مرحله  $n$  ام می باشد .  $\dashleftarrow$

4. یک اتصال بازگشتی بازدارنده که شامل گرادیان تابع محرک مربوط به گره  $i$  ام می باشد .  $\dashleftarrow$

5. یک لینک باز دارنده که گره  $j$  ام از لایه قید را به گره  $i$  ام از لایه متغیر وصل می کند که شامل

گرادیان گره  $j$  ام مرتبط با هر یک از گره های لایه متغیرها است .



Variable layer

Constraint layer

شکل (14) مدل دینامیکی آقای عادل را نشان می دهد

در یک شبکه دینامیکی ، هدف کمینه کردن تابع مورد نظر (به عنوان مثال وزن سازه  $W$ ) می باشد. به طور کلی در این شبکه ها داریم:

$$\text{Minimize } F(X) \quad ()$$

$$g_j(X) \leq 0 \quad j = 1, M \quad ()$$

$$h_k(X) = 0 \quad k = 1, L \quad ()$$

که در آن  $g_j(X)$  ،  $j$  امین نامساوی لایه قید ها و  $h_k(X)$  ،  $k$  امین تساوی لایه قید ها است که در انتهای همگرایی و آموزش شبکه این قیدها باید ارضا شوند.  $M$  تعداد نامساوی های قیدهای بهینه سازی و  $L$  تعداد معادلات تساوی قیدهای بهینه سازی است.  $F$  نیز تابع تحریک اصلی شبکه است که در یک مسأله بهینه سازی می تواند وزن  $W$  باشد.

در یک مسأله بهینه سازی می توان قید های زیر را در نظر گرفت:

$$\begin{aligned} \text{Minimize } W &= \sum_{i=1}^N \rho_i A_i L_i && 0 \\ \sigma^L &< \sigma < \sigma^u && 0 \\ d^L &\leq d \leq d^U && 0 \\ A^L &\leq A \leq A^U && 0 \end{aligned}$$

که  $\sigma$  و  $d$  و  $A$  به ترتیب تنش، تغییر مکان نقاط گرهی و سطح مقطع تک تک اعضا می باشند. در طول آموزش یک شبکه یک تابع خطا (Penalty Function) تعریف شده است که به صورت زیر نشان داده شده است:

$$P(X) = \sum_{j=1}^M \{\max[0, g_j(X)]\}^2 + \sum_{k=1}^L [h_k(X)]^2 \quad 0$$

هنگامی که یک یا چند قید ارضا نشود این اختلاف به صورت مربعات خطا در تابع خطا خود را نشان می دهد. به منظور کاهش تابع خطا یک تابع انرژی با نام تابع لیاپونوف (Lyapunov Function) تعریف می شود:

$$V(X, r_n) = F(X) + \frac{r_n}{2} \left\{ \sum_{j=1}^M [g_j^+(X)]^2 + \sum_{k=1}^L [h_k(X)]^2 \right\} \quad 0$$

که در آن  $g_j^+(X) = \max\{0, g_j(X)\}$  و  $r_n$  پارامتر خطا است که در هر مرحله از تکرار فرایند آموزش از رابطه زیر بدست می آید:

$$r_n = r_0 + \frac{n}{\alpha} \quad 0$$

اگر  $\mathcal{L}(X, r_n)$  کوچکتر یا مساوی صفر باشد تابع  $V$  یا به عبارتی جواب مسأله به حالت پایدار می رسد.

$$\frac{dV}{dt} = \mathcal{L} = \left\{ \frac{\partial F(X)}{\partial X} + r_n \left[ \sum_{j=1}^M g_j^+(X) \frac{\partial g_j}{\partial X} + \sum_{k=1}^L h_k(X) \frac{\partial h_k(X)}{\partial X} \right] \right\} \left( \frac{dX}{dt} \right)$$

بدین ترتیب معادلات دیفرانسیل تعیین کننده مدل دینامیکی عصبی برای بهینه سازی سازه ها به صورت زیر تعریف می شود:

$$\mathcal{L} = -\nabla F(X) - r_n \left[ \sum_{j=1}^M g_j^+(X) \nabla g_j(X) + \sum_{k=1}^L h_k(X) \nabla h_k(X) \right]$$

در نقطه تعادل سیستم باید  $\mathcal{L} = 0$  باشد که از آنجا مقادیر طراحی با محاسبه انتگرال زیر بدست می آید:

$$X = \int \mathcal{L} dt$$

## 3-6 نحوه ارائه داده ها:

در مسائل طراحی نحوه ارائه داده های ورودی و خروجی به شبکه اهمیت زیادی دارد زیرا داده های طراحی مربوط به پارامتر های یک سازه ابعاد بزرگی دارند ، بنابر این باید در این مقادیر تغییراتی داد تا از واگرایی شبکه جلوگیری شود. برای این که یک شبکه عصبی راحت تر و سریع تر آموزش ببیند لازم است که بعضی از داده هایی که بیرون از محدوده معمول داده ها هستند از آموزش شبکه کنار گذاشته شوند.

کاهش ابعاد مسأله نیز یکی از نکاتی است که مقاله به آن توجه داشته است. کاهش مقادیر آموزشی یکی از روشهای بسیار مفید برای همگرایی سریع شبکه به شمار می رود.

Swingler به منظور کاهش ابعاد داده های آموزشی پیشنهادی داده است که ممکن است گاهی اوقات مفید باشد. او استفاده از یک شبکه با تعداد زیادی از ورودیها و تعداد کمی از نرونهای لایه پنهان را پیشنهاد کرده است. اگر وزن شبکه از ابتدا کوچک و به صورت تصادفی فرض شوند ، در مقابل متغیرهایی که از اهمیت زیادی برخوردار نیستند تغییرات کوچکی خواهند داشت. همچنین این متغیرهای کم اهمیت می توانند در صورت نیاز پاک شوند [8].

نکته دیگری که باید مورد توجه قرار داد نحوه برخورد توابع تحریک زیگموئید و تانژانت هیپربولیک نسبت به مقادیر ورودی به این توابع است . چنانچه عدد ورودی به آنها نسبتاً بزرگ باشد ، این توابع نمی توانند به خوبی عمل کنند زیرا حاصل این مقادیر بزرگ به صفر ، 1 و یا -1 همگرا می شود و شبکه در مقابل جداسازی و شناخت نمونه ها از یکدیگر دچار مشکل می شود. با افزایش مقدار ورودی تابع متمایز ساختن خروجی آن نیاز به دقت بالایی دارد. بنابر این بهتر است داده های ورودی به شبکه بین اعداد  $[-1, +1]$  محدود شوند تا با این کار تابع سیگموئید نسبت به داده های ورودی حساس تر شود بنابر این سرعت آموزش افزایش می یابد. برای رهایی از این مشکل باید اطلاعات ورودی را کوچک سازی کنیم. این کار به سادگی با استفاده از قاعده زیر انجام می گیرد. این کوچک سازی اعداد می تواند خطی و یا غیر خطی باشد که معمول ترین آنها تابع خطی زیر است که داده ها را بین  $(-1+1)$  محدود می کند [3].

$$S = \frac{V - V_{\min}}{V_{\max} - V_{\min}} \quad (35)$$

که در آن  $V_{\max}$  ،  $V_{\min}$  به ترتیب کوچکترین و بزرگترین مقادیر ورودی در محدوده مقدار واقعی آنها می باشند و S عدد نرمال شده به ازای هر عدد واقعی V می باشد.

استفاده از این روش ممکن است مقادیری مساوی یکدیگر و یا خیلی نزدیک به (0 یا 1) بدست آید. آقای Eastbrook فرمول دیگری به منظور نرمال سازی داده ها ارائه کرده است که به صورت زیر بیان می شود [8]:

$$A = (V/10^n)^{1/2} + c \quad (36)$$

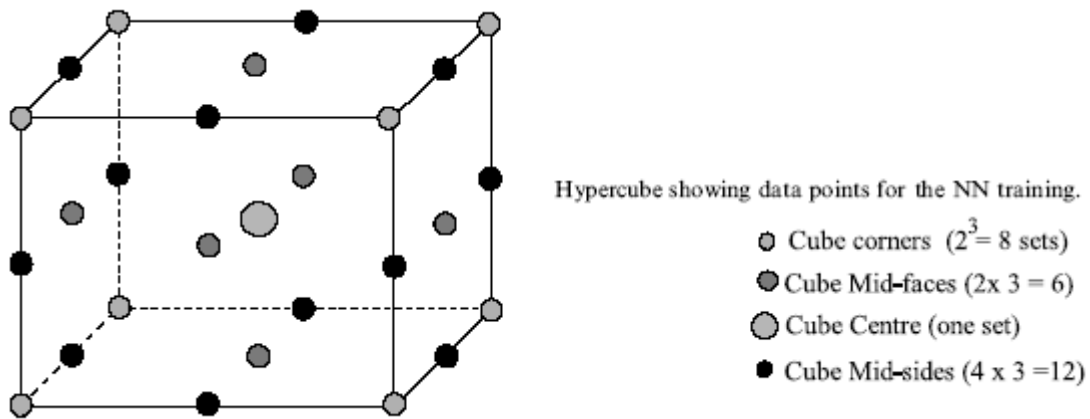
که در آن c ثابتی بین  $(-0/25)$  و  $(+0/25)$  می باشد ، برای اطمینان از این که داده ها بین  $(0/8)$  و  $(0/2)$  بدست آمده باشند. n نیز تعداد رقمهای یک متغیر ورودی است ( به عنوان مثال به ازای  $V = 234.5$  ،  $n = 3$  می باشد ).

نکته مهم دیگری که باید در مورد شبکه عصبی در نظر گرفته شود این است که یک شبکه نمی تواند برون یابی کند . یعنی هنگام آزمایش داده ای خارج از محدوده آموزش شبکه باشد ، شبکه قادر نخواهد بود پاسخ قابل قبولی در ازای این داده از خود نشان دهد. پس باید داده هایی که به یک شبکه آموزش داده می شوند ، محدوده داده هایی را که در آینده برای حل مسائل استفاده خواهند شد در بر بگیرند.

بدین منظور آقای Jenkins یک مدل مکعبی برای انتخاب داده های آموزشی برای مدل عصبی خود استفاده کرده است که این مدل برای مسائل کوچک نتایج قابل قبولی داده است.

وی دریافت که اگر داده های ارائه شده مربوط به گوشه های یک مکعب و همچنین مرکز وجوه و مرکز مکعب را در بر بگیرند ، شبکه قادر خواهد بود در مقابل اعداد تصادفی داخل مکعب جواب قابل قبولی منتشر کند و داده های این نقاط برای آموزش شبکه کافی خواهند بود.

Eastbrook نیز نقاط مربوط به یالهای مکعب را به داده های مورد نیاز برای آموزش اضافه کرده است [8].



شکل (15) مدل مکعبی Jenkins را نشان می دهد.

## فصل چهارم

### کاربرد شبکه عصبی در طراحی سازه

---



## 4-1 مقدمه:

شبکه های عصبی مختلفی به منظور طراحی سازه به کار گرفته شده اند. از جمله مهمترین آنها شبکه چند لایه پرسپترون با الگوریتم یادگیری پس انتشار خطا است. شبکه های دیگری بدین منظور توسط محققین مختلف پیشنهاد شده است که شبکه انتشار متقابل و مدل دینامیکی آقای عادل را می توان نام برد. آقای Eastbrook در مقاله ای که در سال 2001 ارائه کرده است [8]، استفاده از شبکه های چند لایه پرسپترون، Radial Basis Function (RBF) و Normalized Radial Basis Function (NRBF) را به منظور طراحی سازه پیشنهاد کرده است. سپس مقایسه ای بین آنها از نظر اینکه کدام یک از سرعت آموزش بهتر و کدام یک هنگام استفاده راحتتر است انجام داده است.

**مثال 1:** مسأله مورد نظر یک دال بتن آرمه می باشد که هدف از حل آن پیش بینی پارامترهای طراحی بهینه با استفاده از شبکه های عصبی است. در این مقاله تنها پارامتری که مورد بررسی قرار گرفته است، ارتفاع بهینه دال است. زیرا ارتفاع بهینه یک دال در میان پارامترهای طراحی بسیار مهمتر است و همچنین در هزینه تمام شده تأثیر زیادی دارد.

با در نظر گرفتن مدل مکعبی Jenkins داده های آموزشی به صورت زیر تعریف می شوند. داده های ورودی ابعاد طول و عرض دال بتنی و همچنین مقدار بارگذاری در ابعاد مختلف دال می باشند. داده خروجی نیز عمق بهینه دال بتنی است. داده های ورودی به گونه ای انتخاب شده اند که در نقاط بحرانی یک مکعب قرار می گیرند تا داده هایی که در آینده برای آزمایش شبکه داده خواهد شد در داخل این مجموعه قرار داشته باشند و جواب مناسبی از شبکه بگیرند. اطلاعات این مثال در قالب جدول زیر ارائه شده اند. اگر ابعاد بزرگ یک دال ونسبت دال بزرگ به دال کوچک و همچنین بارگذاری را به صورت ابعاد یک نمودار دکارتی در نظر بگیرید یک مکعب ساخته خواهد شد. محدوده داده های ورودی را به صورت زیر در نظر بگیرید:

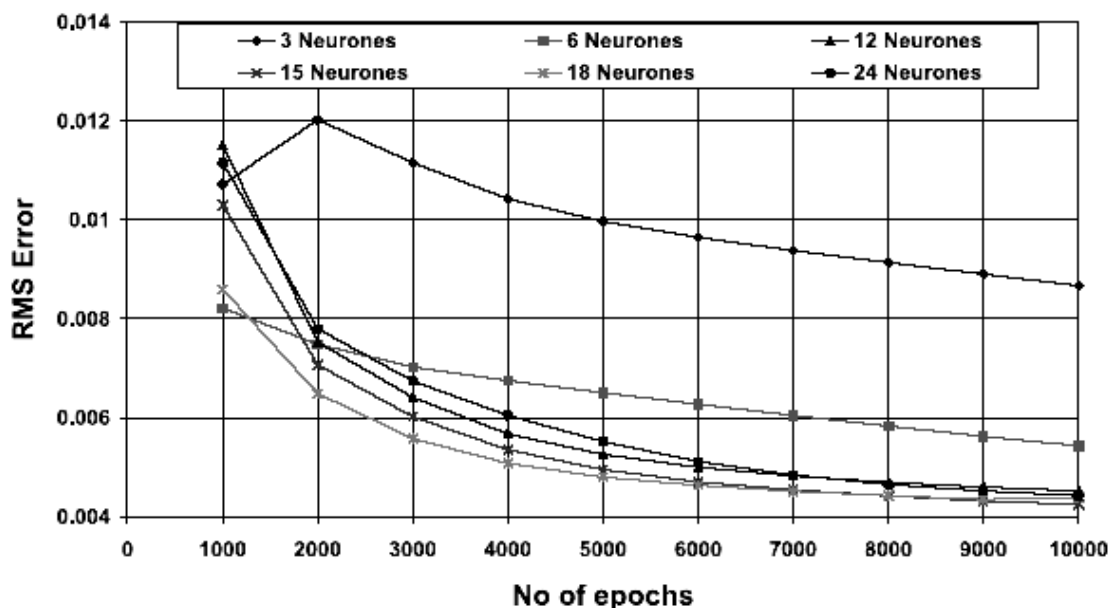
- دهانه کوتاه دال: (4-12m) که در قالب 9 قسمت ارائه می شود.
  - نسبت دهانه بزرگ به دهانه کوچک: (1-2) که دارای 6 قسمت می باشد.
  - بارگذاری اعمال شده: ( $2/5-7/5 KN/m^2$ ) که منجر به ارائه 11 بار متفاوت می شود.
- از این 594 داده موجود تنها نقاطی برای آموزش در نظر گرفته شد که مربوط به نقاط حساس مکعب مذکور باشند. بنابراین این 27 داده کافی است تا تمام داده های مسأله را در خود جای دهد.

Position	Short span (m)	Long span (m)	Design load (kN/m <sup>2</sup> )
Corners	4	4	10.636
	4	8	10.636
	12	12	19.54
	12	24	19.54
	4	4	19.14
	4	8	19.14
	12	12	28.548
	12	24	28.548
Mid faces	8	12	15.004
	8	8	19.34
	8	12	23.676
	8	16	19.34
	4	6	14.972
	12	18	24.044
Centre	8	12	19.34
Mid sides	4	6	10.636
	4	4	14.972
	4	6	19.14
	4	8	14.972
	12	18	19.54
	12	12	24.044
	12	18	28.548
	12	24	24.044
	8	8	15.004
	8	8	23.676
	8	16	23.676
	8	16	15.004

**Training data**

جدول (1) نحوه آموزش داده های ورودی را نشان می دهد

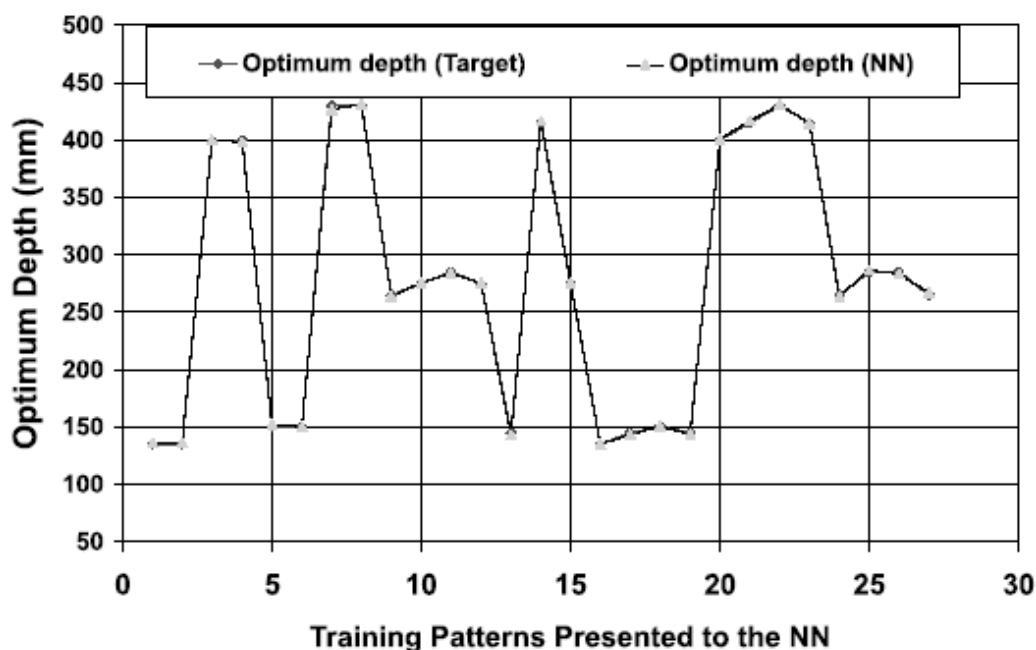
از آموزش و آزمایش شبکه های مختلف برای این مثال این نتیجه بدست آمد که شبکه NRBF خیلی ساده تر و سریعتر از شبکه MLP آموزش دیده است به طوری که یک شبکه MLP در طول 30000 الی 40000 مرحله سعی و خطا آموزش دید در حالی که NRBF فقط با 50 مرحله سعی و خطا موفق به آموزش شد. همچنین نتایج RBF نسبت به دو شبکه دیگر بسیار ضعیف می باشد، یعنی روش خوبی نیست. در مقاله مذکور فقط نتایج شبکه MLP نمایش داده شده است. شبکه عصبی MLP از یک لایه پنهان استفاده کرده است که نتایج مسأله را با تعداد نرون های مختلف در لایه پنهان مشاهده می کنید.



شکل (16) نمودار میانگین مربعات خطا در طول آموزش شبکه نشان می دهد.

همانطور که در شکل مشاهده می شود در این شبکه 15 نرون در لایه پنهان برای یک شبکه MLP برای این مسأله مناسب می باشد. به منظور بدست آمدن داده های آموزشی برای محاسبه عمق بهینه دال از قید ها و محدودیت های زیر استفاده شده است:

- هزینه حداقل
  - نسبت طول دهانه به عمق مؤثر بر اساس آیین نامه BS8110
  - حد اکثر مقدار فولاد (4% کل مقطع) که بر اساس BS8110 شرح داده شده است.
- نتایج عددی را در نمودار زیر مشاهده می کنید. در این نمودار نتایج بدست آمده توسط شبکه MLP نشان داده شده است. حداکثر خطای موجود در شبکه حدود 3% می باشد که نتیجه قابل قبولی است.



شکل (17) نتایج عمق بهینه دال بتن آرمه را از شبکه عصبی با مقادیر هدف مقایسه کرده است.

#### 4-2 بهینه سازی سازه ها با استفاده از شبکه ها:

در مقاله ای که آقای ایرانمنش و دکتر کاوه ارائه کرده اند ، شبکه های عصبی برای پردازش اطلاعات مربوط به بهینه سازی سازه ها تربیت شده اند . در این مقاله دو شبکه عصبی آموزش دیده اند . یکی برای قیود و دیگری برای مشتقات قیود و بهینه سازی سازه ها با به کارگیری این شبکه های عصبی مصنوعی انجام می گیرد [5].

هدف اصلی روش بهینه سازی این مقاله به کارگیری شبکه های عصبی در بهینه یابی سازه ها و کاهش زمان محاسبات در مقایسه با روش های متداول است .

به منظور ایجاد شرایط بهینه سازی برای یک سازه باید قیود مربوط به تنش ها و جابجایی ها تعیین شود و محاسبه مشتقات قیود همگی در مرحله اول منتهی به تعیین مشتق جابجایی ها نسبت به متغیرهای طرح می باشد زیرا اکثر قیود مستقیماً به جابجایی ها مرتبط می گردند .

پس از اینکه مشتقات جابجایی ها نسبت به متغیر های طرح تعیین گردید با داشتن این اطلاعات از مختصات محلی امکان محاسبه مشتق نیروها و در نهایت تنش ها نسبت به متغیرهای طرح فراهم می شود . به عنوان مثال برای یک عضو از خرپا قیدهایی زیر اعمال می شود.

$$g_1 = \frac{\sigma}{\sigma^+} - 1 \quad 0$$

$$g_2 = \frac{\sigma}{\sigma^-} - 1 \quad 0$$

که  $\sigma^+, \sigma^-$  به ترتیب محدوده مجاز تنشهای کششی و فشاری هستند

$$\sigma = \frac{P}{A} \quad 0$$

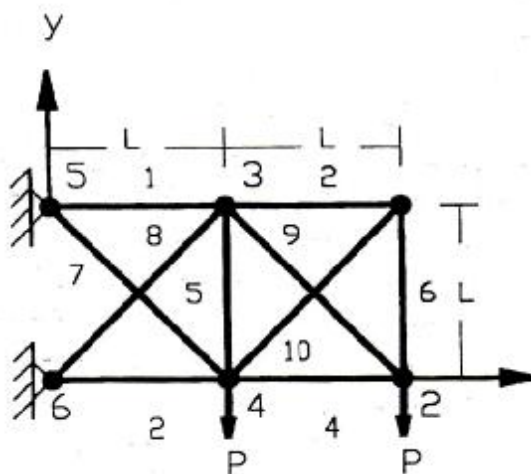
$$P = KU = \frac{EA}{L}U \quad 0$$

$$\frac{\partial P}{\partial X} = \frac{P}{A} + \frac{EA}{L} \times \frac{\partial U}{\partial X} \quad 0$$

$$\frac{\partial \sigma}{\partial X} = \frac{\partial}{\partial X} \frac{P}{A} = \frac{1}{A} \frac{\partial P}{\partial X} - \frac{P}{A^2} \quad 0$$

اجرای بهینه سازی در سه مرحله صورت می گیرد . در اولین مرحله متغیرهای طراحی (به عنوان مثال سطح مقطع عضو) به صورت تصادفی ایجاد می گردند و سپس قیود و مشتقات قیود مربوط به هر مجموعه از متغیرهای تصادفی ایجاد شده محاسبه می گردند . در مرحله دوم ، دو شبکه عصبی انتشار متقابل بر اساس اطلاعات قیود و مشتقات آن آموزش داده می شوند و ماتریس های وزن به طور خود کار در مرحله آموزش محاسبه می گردند . در مرحله نهایی به منظور بهینه سازی با استفاده از شبکه های عصبی آموزش داده شده بدین صورت انجام می گیرد که قیود و مشتقات مربوط در مدت زمان کوتاهی از طریق شبکه عصبی در اختیار بهینه ساز قرار می گیرد [5].

**مثال 2:** در این جا خر پای 10 عضوی با هدف کمینه کردن وزن سازه مورد بررسی قرار می گیرد. این سازه توسط محققان متعددی در بهینه سازی به کار گرفته شده است.



شکل (17) نمونه خرابی 10 عضوی را نشان می دهد

با در نظر داشتن سطح مقطع اعضا به عنوان متغیرهای طرح  $X_{10}, X_9, X_8, X_7, X_6, X_5, X_4, X_3, X_2, X_1$  و OBJ به عنوان تابع هدف برای حل دقیق داریم:

$$OBJ = \sum_{i=1}^{NE} \rho_i A_i L_i$$

$$OBJ = 2.4705(X_1 + X_2 + X_3 + X_4 + X_5 + X_6) + 3.4938(X_7 + X_8 + X_9 + X_{10})$$

بنابر این جواب بهینه عبارت است از:  $OBJ = 2291kg$  به طوری که سطح مقطع بهینه اعضا عبارتند از:  
 $A = (167.4, 0.1, 160.4, 109.4, 0.1, 0.1, 57.7, 141.2, 147.2, 0.1)$

به منظور آموزش شبکه عصبی سطح مقطع اعضا به طور اتفاقی در فاصله  $A \pm 0.2A$  آموزش داده می شود. نتایج حاصل را به ازای تعداد زوجهای مختلف جهت آموزش در جدول (2) مشاهده می کنید. همانطور که در جدول مشاهده می شود حد اکثر خطای ایجاد شده از 4/6% تجاوز نمی کند [5].

در بررسی دیگری فضای آموزشی از  $A \pm 0.2A$  به  $A \pm 0.4A$  افزایش داده شد تا آثار آن بر مسائل بهینه سازی مشخص گردد. نتایج این بررسی در جدول (3) آمده است. مشاهده می شود که خطای حاصل افزایش یافته است، اما با افزایش تعداد زوجهای آموزشی این مقدار خطا کاهش یافته است

در بررسی دیگری آثار تغییرات تعداد نرون های لایه رقابتی مورد بررسی قرار گرفت. برای شرایط 1000 زوج آموزشی، پارامتر D (فاصله اقلیدسی) که وظیفه گروه بندی نرون های لایه رقابتی را دارد [3] (در بخش 3-1-3 توضیح داده شده است)،  $D=2.7$  (به ازای 20 نرون رقابتی) بسیار نزدیک به جواب بود. با افزایش مقدار D پاسخ از جواب مطلوب فاصله می گیرد.

جدول (1) پنج بهینه سازه با استفاده از شبکه های عصبی هنگامی که تعداد زوج های آموزشی افزایش می یابد

NEURO OPTIMIZER	شبکه های عصبی بر اساس زوج های آموزشی مختلف آموزش داده شده اند					
	مقیارهای طرح	جواب دقیق	200 N = 158	300 N = 206	400 N = 247	500 N = 298
$X_1$	۱۴۷/۴	۱۷۸/۶	۱۷۴/۸	۱۶۱/۶	۱۷۲/۴	۱۶۵/۶
$X_2$	-/۱	-/۱	-/۱	-/۱	-/۱	-/۱
$X_3$	۱۶۰/۴	۱۷۰/۶	۱۶۶/۸	۱۶۱/۶	۱۶۵/۴	۱۶۷/۴
$X_4$	۱۰۹/۴	۱۰۹/۴	۱۰۵/۸	۱۰۰/۴	۱۰۴/۴	۱۰۴/۴
$X_5$	-/۱	-/۱	-/۱	-/۱	-/۱	-/۱
$X_6$	-/۱	-/۱	-/۱	-/۱	-/۱	-/۱
$X_7$	۵۷/۷	۵۳/۶	۵۳	۴۸/۵	۵۱/۶	۵۳/۴
$X_8$	۱۴۱/۴	۱۳۸/۸	۱۳۳/۴	۱۳۵/۹	۱۳۱/۴	۱۳۳/۹
$X_9$	۱۳۷/۴	۱۳۵/۸	۱۳۰/۴	۱۳۲/۹	۱۳۸/۴	۱۳۰/۹
$X_{10}$	-/۱	-/۱	-/۱	-/۱	-/۱	-/۱
OBJ (kgs)	۲۲۹۱	۲۲۲۰	۲۲۴۵	۲۱۴۱	۲۲۱۷	۲۲۵۲
E	-/۰	۲/۱	۳/۲	۴/۶	۲/۶	۳/۱

$D = 1/5$  پارامتر گروه بندی کننده نرون های لایه رقابتی  
 $E$  = میانگین خطا  
 $N$  = تعداد نرون های لایه رقابتی که توسط شبکه های عصبی ایجاد شده  
 $A \pm 0.2A$  = فضای جستجو

جدول (2) نتایج مثال خرابی 10 عضوی

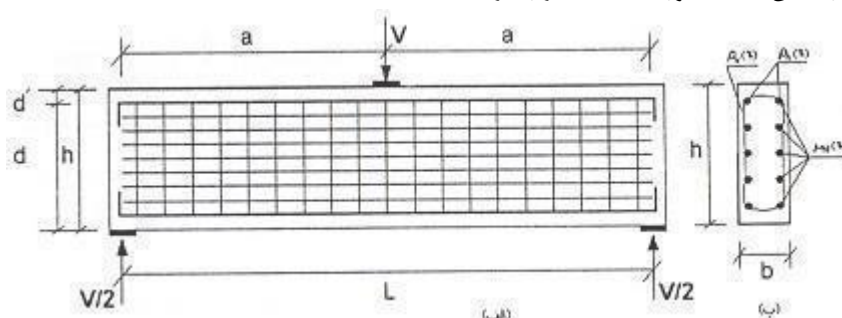
جدول (3) نتایج خرابی 10 عضوی با افزایش فضای آموزشی

### 3-4 پیش بینی مقاومت نهائی برش تیرهای عمیق بتن مسلح دوسر ساده:

این مقاله توسط آقای امیدی نسب در سال 1382 به کمک دکتر عربزاده ارائه شده است. در این مقاله برای آموزش شبکه از داده های آزمایشگاهی استفاده شده است و نتایج آن را با آیین نامه ها و روشهای مختلف جهت تعیین مقاومت نهایی برش مقایسه کرده است [7].

تیرهای عمیق دارای نسبت دهانه به عمق کمتر از 5 می باشند و بر خلاف تیرهای معمولی، رفتار این نوع تیرها در محدود ارتجاعی نیز به وسیله توزیع کرنش غیر خطی تعیین می شود. عموماً به خاطر ابعاد و رفتار خاص این نوع تیرها، رفتار برشی آنها کنترل می شود.

داده های آموزشی و آزمایشی مورد نیاز شبکه از داده های آزمایشگاهی که پارامترهای آن در شکل دیده می شود، جمع آوری شده است. این داده ها شامل 82 نمونه تیر عمیق است که 90% آن برای آموزش و حدود 10% آن برای آزمایش شبکه مورد استفاده قرار گرفته است.



شکل (18) پارامترهای اساسی تیرها

که در آن  $L$  طول دهانه موثر،  $b$  عرض،  $a$  دهانه برش،  $h$  عمق تیر،  $d$  عمق موثر تیر،  $\rho_v$  و  $\rho_h$  به ترتیب نسبت آرماتور فولادی افقی و عرضی و  $V$  مقاومت نهایی برشی تیر عمیق است. در شبکه طراحی شده، داده های ورودی و خروجی به صورت زیر می باشد:

$$(L, a, b, d, f'_c, f_{yh}, f_{yv}, \rho_h, \rho_{hl}, E_s, V)$$

که در آن  $f_{yh}$  و  $f_{yv}$  به ترتیب مقاومت تسلیم فولادهای عرضی و طولی و  $\rho_{hl}$  نسبت مجموع آرماتورهای فولادی افقی است. 11 داده اول بردار ورودی و  $V$  بردار خروجی شبکه را تشکیل می دهند. در این مقاله از یک نوع الگوریتم اصلاح شده انتشار برگشتی به نام Resilient- Propagation استفاده شده است که فقط از علامت مشتق برای تعیین جهت وزن استفاده شده است و مقدار مشتق هیچ اثری در وزن ندارد. [7]، شبکه شامل دو لایه ورودی و خروجی و یک لایه میانی (پنهان) می باشد که لایه ورودی شامل 11 نرون، لایه میانی شامل 5 نرون و لایه خروجی شامل یک نرون می باشد. یک تابع تحریک تانژانت هیپربولیک برای لایه میانی و یک تابع تحریک خطی برای پردازش لایه خروجی در نظر گرفته شده است.

بعد از آموزش و آزمایش شبکه، مؤلف نتایج آن را با روشهایی آیین نامه ای و تحلیلی مختلف برای پیش بینی مقاومت برش تیرهای عمیق مقایسه کرده است.

میانگین نسبت مقاومت واقعی به پیش بینی شده برای داده آزمایش در روش ACI برابر 0/64، در CIRIA، 0/66، در روش تعادل نیرو 0/82، در روش خریایی 0/85 و برای شبکه عصبی 1/04 بدست آمده است. با توجه به اینکه داده های آموزش داده شده بین محدوده خاصی قرار دارند و شبکه عصبی برون

یابی نمی کند ، این شبکه آموزش دیده محدوده های دیگر را تحت پوشش قرار نمی دهند و بدیهی است با افزایش محدوده های آموزشی این شبکه بهتر جواب می دهد .

---



## 4-4 بهینه سازی تیرهای فولادی سرد نورد:

دکتر حجت عادلی تلاشهای فراوانی در زمینه شبکه های عصبی مصنوعی و کاربرد آن در مهندسی سازه و طراحی انجام داده است [9].

در سال 1997 ایشان مقاله ای ارائه کرده است که کاربرد شبکه های عصبی در طراحی بهینه تیرهای سرد نورد شده فولادی را نشان می دهد .

داده های آموزشی از روابط طراحی بدست آمده از آیین نامه AISI به دو روش (ASD) روش طراحی تنش مجاز و (LRFD) طراحی به روش ضریب بار تشکیل شده است و همچنین برای آموزش شبکه عصبی از مدل دینامیکی عصبی آقای عادلی و پارک استفاده شده است .

سازه های مختلف که مورد آزمایش و بارگذاری قرار گرفته اند شامل سه مدل مختلف و معمول I شکل ، Z شکل و کلاهی شکل می باشند.

هدف استفاده از شبکه های عصبی بدست آوردن مقطع بهینه طراحی برای نیمرخ هایی است که از مقاطع فولادی سرد نورد ساخته شده و در سازه استفاده می شوند .

علت اصلی استفاده از مقاطع سرد نورد ، قابلیت انعطاف آنها است از این نظر که این مقاطع را در شکلها و اندازه های دلخواه در اختیار طراح قرار می دهد .

سوآلی که در مورد این مقاطع به وجود می آید این است که چه مقطعی انتخاب شود تا اقتصادی ترین مقطع برای طراحی چنین سازه های فولادی بدست آید .

تا آن موقع آقای Salmon از یک نوع روش بهینه سازی در سال 1974 برای بدست آوردن عمق بهینه سازه فولادی سرد نورد استفاده کرده بود که این روش دارای محدودیت هایی بود و هیچ منحنی همگرایی نمی توانست ارائه دهد و روش مورد استفاده نمی توانست تضمین کند که یک طراحی بهینه صورت گرفته است .

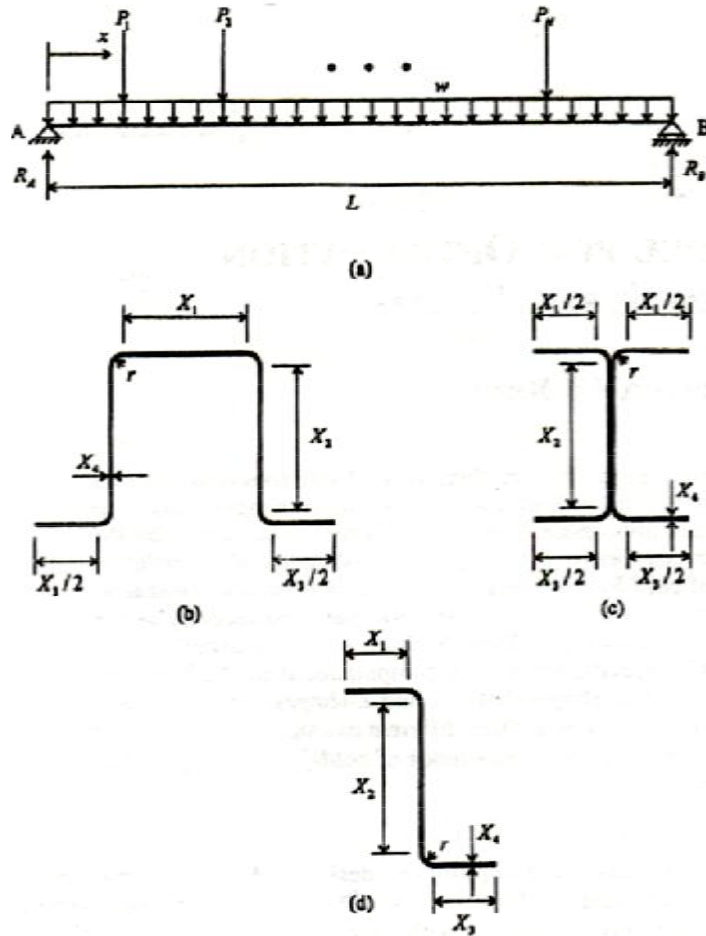
در سال 1995 دکتر عادلی و آقای پارک شبکه عصبی دینامیکی برای این منظور مدل کردند که قادر است مسائل بهینه سازی برای شرایط ویژه و پیچیده غیر خطی را حل کند . این مدل برای اولین بار برای حل مسائل بهینه سازی خطی توسط همین افراد ساخته شده بود . سپس این مدل برای مسائل بزرگ غیر خطی بهینه سازی به کار گرفته شد و از قیدهای آیین نامه AISC برای روشهای ASD (تنش مجاز) و LRFD (روش ضریب بار) برای طراحی سازه فلزی نرم استفاده شده است .

معمولاً الگوریتم های بهینه سازی با بزرگ شدن مقیاس و پیچیده شدن مسأله نتیجه مناسبی نمی دهند ، در صورتی که با مدل بهینه سازی جدید آقای عادلی می توان برای سازه های بزرگ با هزاران عضو پیچیده نتایج قابل قبولی از طراحی بهینه بدست آورد. [9]

در مقاله ارائه شده هدف بدست آوردن وزن بهینه در طراحی تیرهای سرد نورد شده با تکیه گاه ساده می باشد که بر اساس آیین نامه AISI به دو روش ASD و LRFD توصیه شده است . بار گذاری نیز یا به صورت یکنواخت یا به صورت متمرکز فرض شده است و همچنین می توان تقویت کننده هایی را در طول تیر فرض کرد .

قید هایی که برای آموزش شبکه در نظر گرفته شده مربوط به ابعاد مقطع تیر و همچنین نحوه بارگذاری و تقویت تیر می باشد و داده خروجی همان وزن مطلوب سازه است .

ابعاد مقطع تیر در سه حالت I ، hat و Z شکل به صورت زیر تعریف می شوند :



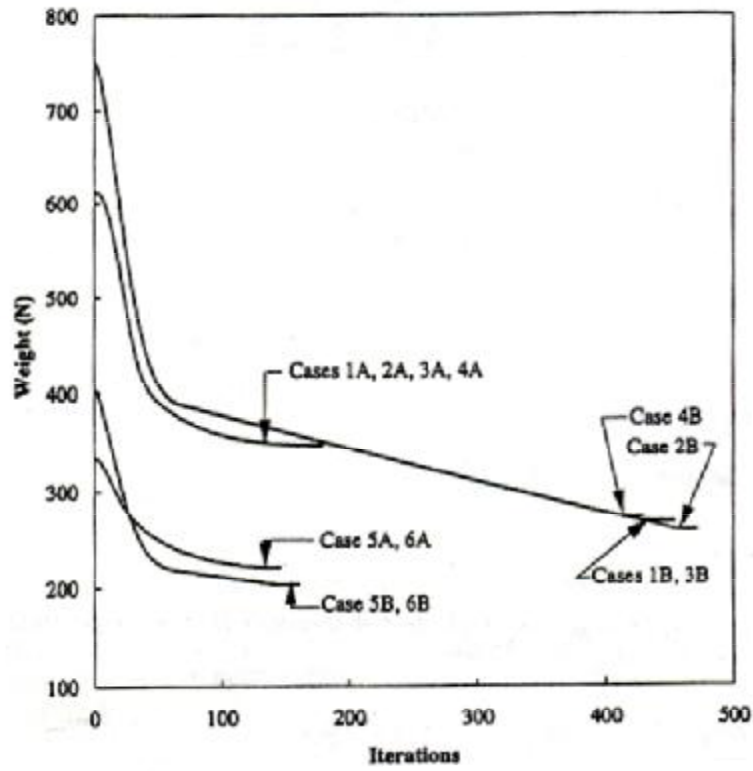
شکل (19) پارامترهای بهینه سازی فولاد های سرد نورد برای سه حالت مختلف I-, Z-, hat شکل

**مثال 3:** که ارائه شده است یک تیر دو سر ساده با طول  $L=3m$  که به صورت بار گسترده بار گذاری شده است و بار مرده و زنده به ترتیب  $1kN/m$  و  $5kN/m$  می باشد، در نظر گرفته شده است. دو فرض اولیه از متغیر های مسأله برای طراحی بهینه در نظر گرفته شده است:  $A\{20,20,100,6\} mm$  و  $B\{20,20,150,6\} mm$  که نتایج بدست آمده را در قالب جدول می بینید.

Optimum Solutions for Example

Case No. (1)	Shape (2)	Specification (3)	$X_1$ [mm (in.)] (4)	$X_2$ [mm (in.)] (5)	$X_3$ [mm (in.)] (6)	$X_4$ [mm (in.)] (7)	$W$ [N (lb)] (8)
Initial Design A			20 (0.8)	20 (0.8)	100 (3.9)	6.0 (0.24)	611.3 (137.4)
1	Hat	ASD	31 (1.2)	31 (1.2)	119 (4.7)	3.0 (0.12)	347.0 (78.0)
2	Hat	LRFD	31 (1.2)	31 (1.2)	119 (4.7)	3.0 (0.12)	347.0 (78.0)
3	I	ASD	31 (1.2)	31 (1.2)	119 (4.7)	3.0 (0.12)	347.0 (78.0)
4	I	LRFD	31 (1.2)	31 (1.2)	119 (4.7)	3.0 (0.12)	347.0 (78.0)
5	Z	ASD	40 (1.6)	40 (1.6)	132 (5.2)	3.1 (0.12)	221.1 (49.7)
6	Z	LRFD	40 (1.6)	40 (1.6)	132 (5.2)	3.1 (0.12)	221.1 (49.7)
Initial Design B			20 (0.8)	20 (0.8)	150 (5.9)	6.0 (0.24)	749.9 (168.6)
1	Hat	ASD	36 (1.4)	36 (1.4)	143 (5.6)	2.1 (0.08)	266.6 (59.9)
2	Hat	LRFD	36 (1.4)	36 (1.4)	143 (5.6)	2.0 (0.08)	258.5 (58.1)
3	I	ASD	36 (1.4)	36 (1.4)	143 (5.6)	2.1 (0.08)	266.6 (59.9)
4	I	LRFD	36 (1.4)	36 (1.4)	143 (5.6)	2.1 (0.08)	272.3 (61.2)
5	Z	ASD	36 (1.4)	36 (1.4)	143 (5.6)	2.8 (0.11)	203.2 (45.7)
6	Z	LRFD	36 (1.4)	36 (1.4)	143 (5.6)	2.8 (0.11)	203.2 (45.7)

جدول (2) نتایج مثال تیر دو سر ساده



شکل (20) مقادیر وزن بهینه مثال تیر دو سر ساده را در طول سعی و خطا نشان می دهد

## 4-5 بهینه سازی دالهای بتن آرمه:

بهینه سازی دالهای بتن آرمه نیز توسط مدل دینامیکی عصبی آقای عادل در سال 2005 مدل سازی شده است. در روند بهینه سازی سازه ها برای یافتن کاربرد گسترده در مهندسی باید این سازه ها را که در روند طراحی از قیدهای آیین نامه ACI استفاده می شود از نظر اقتصادی بهینه سازی کرد. به طور کلی در مثال های معمول و آکادمیک از بهینه یابی وزن سازه در بهینه سازی استفاده می شود. اما در کارهای بتنی معمولاً هزینه بتن، میلگرد و قالب بندی ملاک بهینه سازی می باشد [13].

کارهای اولیه ای که در مورد بهینه سازی دالهای بتن آرمه صورت گرفته بود روی یک دال بتنی یک طرفه تحت خمش خالص بر اساس آیین نامه ACI انجام شد. هزینه یک دال را با مجموع هزینه بتن و فولاد آن فرمول بندی شده است و فرمول ساده ای برای تابع هزینه آن بدست آمده است. یا در یک فرمول بندی ساده دیگر برای یافتن ضخامت بهینه در یک دال یک طرفه با بارگذاری گسترده و با تکیه گاه ساده ارائه شد و در آن فرضیات دیگر را ساده سازی شد.

در سال 1985، آقای Neilsen یک روش برای بهینه سازی هزینه سازه های بتنی پوسته ای ارائه کرد که در آن نیروهای عمودی نیز به سیستم اعمال می شد و در این روش از قیدهای ACI استفاده نشد.

بنابر این در روشهای ارائه شده گذشته بهینه سازی دارای محدودیتهایی بود. بعضی از مقاله ها روی هزینه فولاد تأکید داشتند و برخی روی هزینه بتن تأکید می کردند. در این مقاله ها برای تکیه گاه دالها محدودیت قائل شده بود به نحوی که فقط از تکیه گاه ساده در دو طرف و گیردار در طرف دیگر استفاده شده بود و دالها فقط یک طرفه و تنها برای یک دهانه فرموله شده بود. بعضی از مقاله ها نیز برای راحتی کار از قیدهای آیین نامه ACI استفاده نکردند [13].

در مقاله مذکور ابتدا یک فرمول بندی کلی برای بهینه سازی هزینه دالهای بتن آرمه یک یا چند دهانه با وضعیت تکیه گاه متغیر (تکیه گاه ساده، یک انتها گیردار، دو انتها گیردار و یا یک سر آزاد) استفاده شده است که از قید های آیین نامه ACI (1999) برای طراحی استفاده شده است: ضخامت دال، قطر فولاد و فاصله آرماتور بندی [13].

به عنوان مثال در یک دال بتن آرمه نامساوی های زیر به عنوان قید در نظر گرفته می شود.

$$1) M_u \leq \phi \cdot M_n \quad \phi = 0.9$$

$$2) V_u \leq \phi \cdot V_c \quad \phi = 0.85$$

$$3) \rho < \rho_{\max}$$

$$4) A_{s \min} = \begin{cases} 0.0020 * bh & \text{For Steel Grades 40 and 50} \\ 0.0018 * bh & \text{For Steel Grades 60} \end{cases}$$

$$5) S \leq \min(3h, 18")$$

$$6) S \geq \max(d_b, 1")$$

این روش در دو مرحله صورت می گیرد. ابتدا مدل دینامیکی عصبی آقای عادل و پارک برای تعیین و حل بهینه با فرض متغیرهای پیوسته استفاده شده است. سپس یک فرمول بندی برای بهینه سازی یک سازه با فرض نامعلوم و فرض متغیرهای مختلف در مسأله بدست می آید.

#### مثال 4:

در ادامه 4 مثال دال یک طرفه با شرایط تکیه گاهی مختلف حل شده است. در این مثال ها ضریب همگرایی شبکه  $\alpha = 0.001$  فرض شده است.

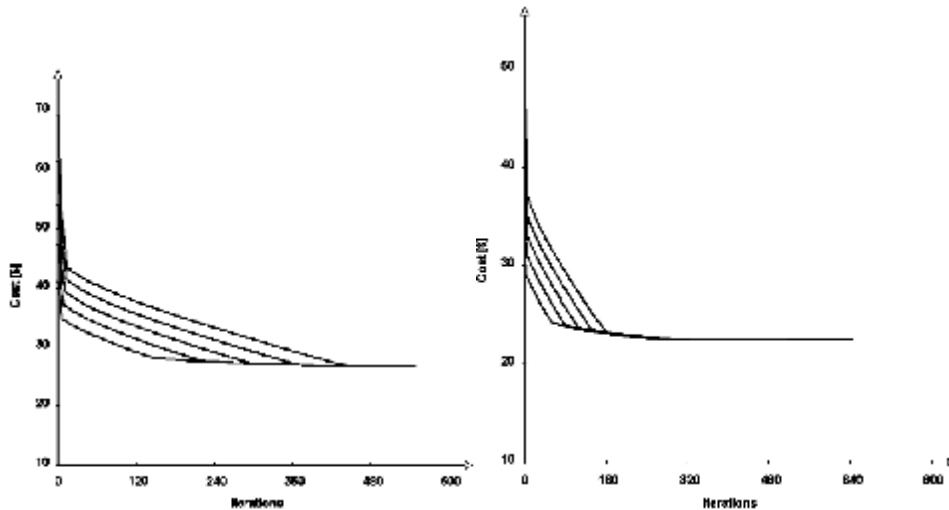
مثال اول یک دال دو طرف تکیه گاه ساده، مثال دوم از یک طرف تکیه گاه ساده و از طرف دیگر انتها پیوسته است، در مثال سوم تکیه گاه از دو طرف پیوسته است (که قسمتی از یک دال چند دهانه بتن آرمه است)، در مثال چهارم نیز یک تکیه گاه گیردار وجود دارد. نتایج و مقدار همگرایی مسأله را در قالب نمودار مشاهده می کنید [13].

Cost optimization results for examples 1-4

Point	$h$ (in)	$d_b$ (in)	$s$ (in)	Total cost (\$)
Example 1	6.75	0.375	6.5	26.45
Example 2	5.75	0.375	7	22.98
Example 3	4.75	0.375	7	19.93
Example 4	13.5	0.375	2.0	60.22

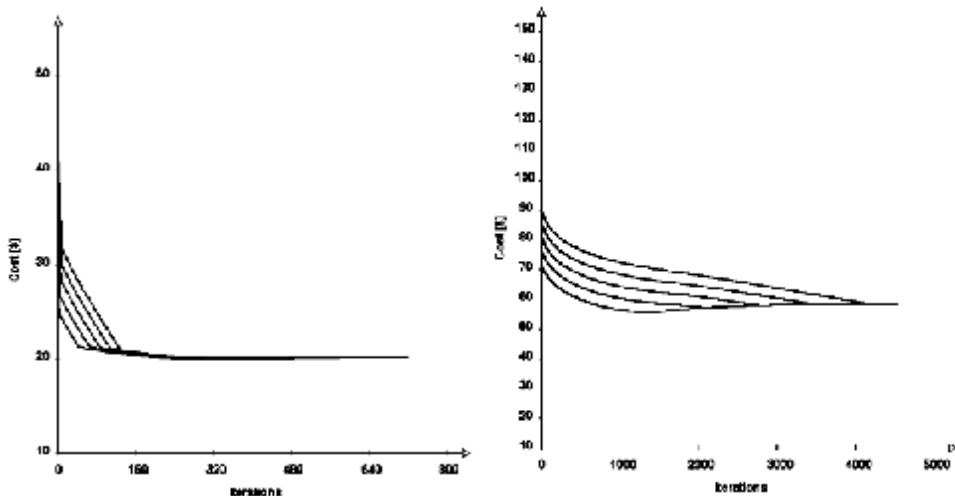
جدول (3) نتایج مثالهای 4-1

در این جدول  $h$  ارتفاع دال،  $d_b$  قطر فولاد تسلیح و  $s$  فاصله بین آرماتورها است.



شکل (22) همگرایی مثال 2

شکل (21) همگرایی مثال 1



شکل (24) همگرایی مثال 4

شکل (23) همگرایی مثال 3

شکلهای بالا همگرایی هزینه بهینه را در طول سعی و خطا نشان می دهند (با 5 نقطه شروع مختلف)

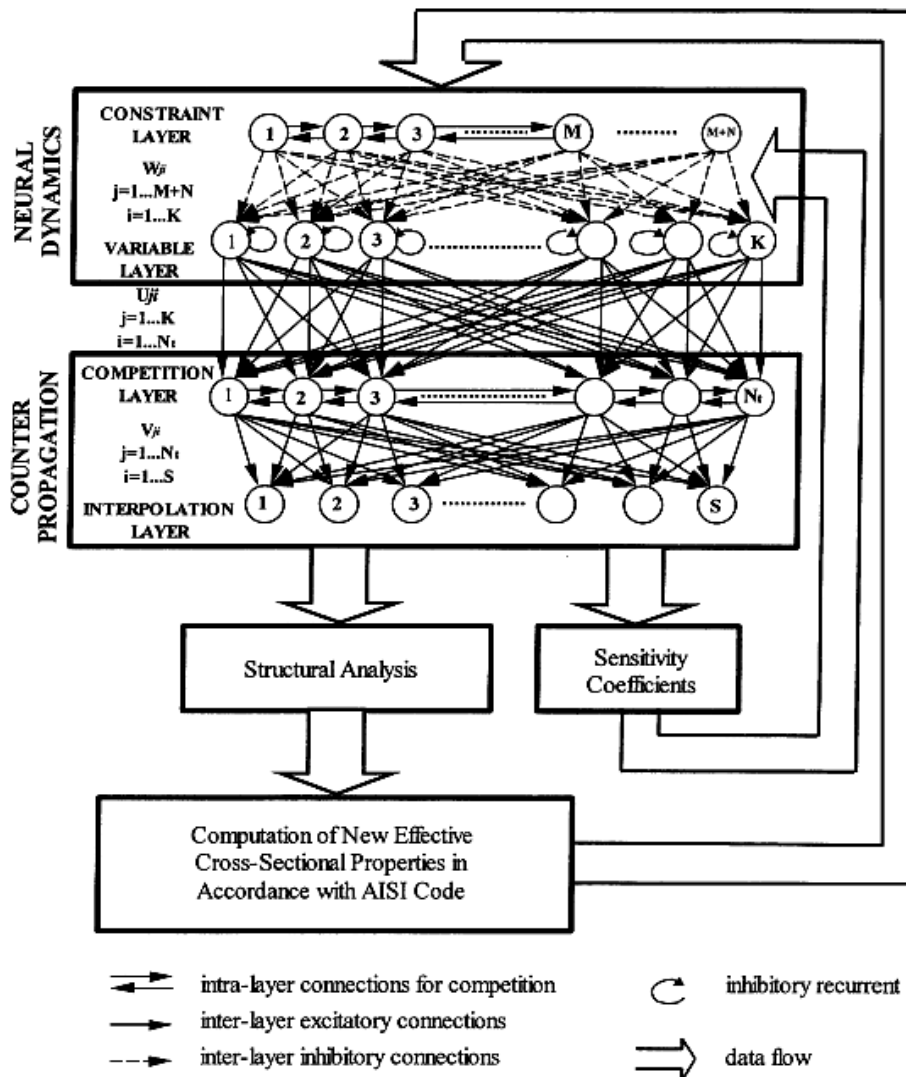
در هر مثال با 5 فرض اولیه برای هزینه بهینه در نظر گرفته شده است. همانطور که از نمودارها مشخص است این فرضیات در انتهای آموزش به یک مقدار همگرا خواهند شد. در مثال اول جواب مساله بعد از 160 مرحله سعی و خطا همگرا شد. در مثال دوم نیز بعد از 450 مرحله، در مثال سوم بعد از 4000 مرحله و در مثال آخر بعد از 150 مرحله سعی و خطا مساله همگرا شده است.

#### 4-6 سازه های فولادی سرد نورد فضاکار :

آقای عادل مقاله دیگری همراه آقای تشکری در سال 2002 ارائه کرده است که نشان می دهد بهینه سازی سازه های فضایی فولادی سرد نورد چگونه با شبکه عصبی دینامیکی مدل شده است [11]. بهینه سازی سازه های فضایی ساخته شده از فولاد سرد نورد کاری پیچیده است زیرا کاهش مساحت موثر باید برای اعضای تحت فشار محاسبه شود که این اعضا دارای توزیع تنش غیر یکنواختی هستند که منجر به کماتش پیچش این اعضای سرد نورد می شود. از طرفی بدست آوردن مقطع اقتصادی و بهینه برای تک تک اعضای سازه های فضایی با توجه به تعداد قابل ملاحظه آنها بدون در نظر گرفتن یک روند بهینه سازی مناسب دشوار خواهد بود.

در این مقاله آقای عادل از یک مدل شبکه عصبی انتشار متقابل هیبریدی استفاده کرده که می تواند طراحی بهینه (وزن بهینه) خرپاهای فضایی فلزی را انجام دهد. این طراحی با قیدهای طراحی آیین نامه AISI به روش ASD مطابقت دارد.

این مدل برای یافتن طراحی وزن بهینه خرپاهای فضایی که به عنوان سقف سازه استفاده می شود به کار گرفته شده است. این مدل دینامیکی در شکل نشان داده شده است.



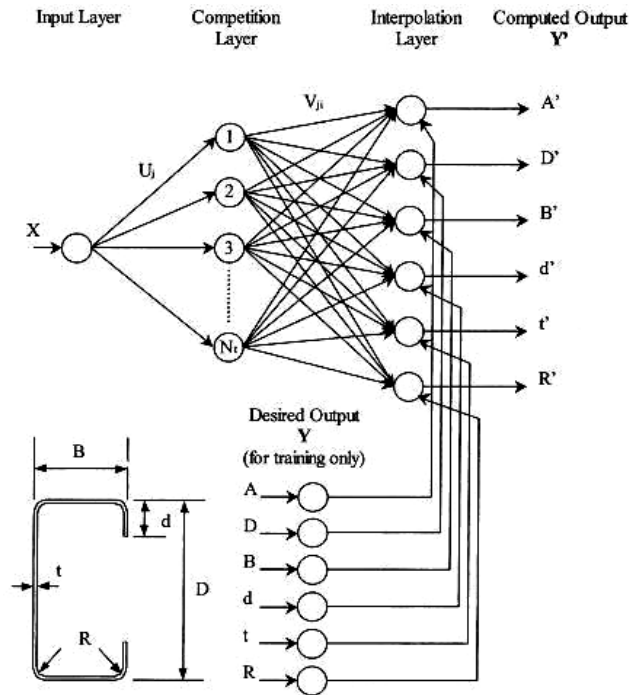
Data flow and topology of the hybrid CPN-neural dynamics model for optimization of cold-formed space trusses.

شکل (26) فلوچارت مدل دینامیکی آقای عادل را در مراحل مدل دینامیکی و انتشار متقابل نشان می دهد

در هر مرحله از عملیات بهینه سازی ، معیار خطا و حساسیت در لایه قیدها محاسبه می شود . یک شبکه انتشار متقابل مشخصات مقاطع را به طور مجزا محاسبه می کند . گره های لایه قید این مشخصات را از CPN به عنوان ورودی دریافت می کند و معیار خطا را به عنوان خروجی محاسبه می کند . لایه متغیرهای طراحی خطای وزن دار لایه قید را دریافت می کند و مقادیر طراحی را به عنوان خروجی محاسبه می کند . تعداد قیدهای تنش برابر با تعداد اعضای سازه می باشد که ممکن است به هزاران عضو برسد . برای آنکه روند طراحی سرعت قابل قبولی داشته باشد در یک لایه رقابتی در شبکه CPN در گروه های مشابه دسته بندی می شوند و در هر گروه عضوی که دارای بیشترین تنش غیر مجاز باشد برای محاسبه انتخاب می شود . هنگامی که مقادیر طراحی از لایه متغیرها به لایه رقابتی منتقل می شوند مقدار فاصله بین وزن های متغیرها و وزن های اتصال این دو لایه به همدیگر محاسبه می شود .

$$d_j = \sqrt{u_j^2 - X^2} \quad (37)$$

سپس نرونی که کمترین فاصله اقلیدسی از بردار  $X$  را داشته باشد به عنوان برنده انتخاب می شود و برای محاسبه به لایه درون یابی فرستاده می شود. سپس در لایه درون یابی مقاطع اعضا مشخصات جدیدی به خود می گیرند. این مشخصات به وسیله وزن های اختصاص یافته مربوط به اتصال هر گره از لایه رقابتی به لایه درون یابی تعیین می شوند. این روند را در شکل زیر که یک شبکه CPN پیش رو را نشان می دهد می بینید [11]:



شکل (27) قسمت پیشرفت یک شبکه انتشار متقابل را نشان می دهد.



#### 4-7 بهینه سازی سازه های فولادی با وجود عدم قطعیت :

در دو دهه اخیر به منظور طراحی بهینه سازه هایی که اطلاعات دقیق از آنها در دست نیست ، روش های آنالیزی اتفاقی و عدم قطعیت صورت گرفته است . طراحی بهینه بر اساس قابلیت اطمینان : (Reliability-Based Design Optimization) (RBDO) و طراحی بهینه قطعی (Robust Design Optimization) (RDO) دو روش مجزا از هم هستند که برای این منظور استفاده شده اند. این روش ها دارای فرمول بندی طراحی هستند که قادرند پاسخ سیستم های دارای عدم قطعیت را بدست آورند [15].

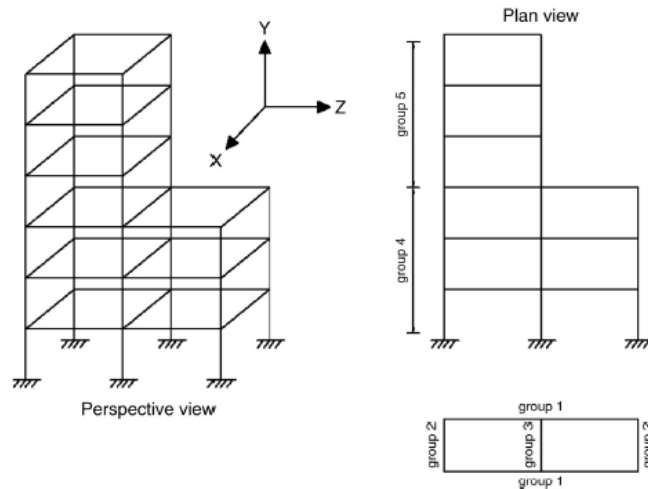
با وجود تئوری های پیشرفته در زمینه آنالیز دقیق ، هنگامی که با مسائل واقعی برخورد می کنیم . در محاسبات با مانع های جدی مواجه می شویم . روش طراحی بهینه بر اساس قابلیت اطمینان ابزار قوی برای سیستم های سازه ای با مقیاس بزرگ است که در محاسبات طراحی و آنالیز پر قدرت ظاهر می شود . در حالی که این روش های محاسباتی آنالیزی با موفقیت انجام می شوند ، هنوز نیاز است که در مقابل مسائل کاربردی ، محاسباتی غیر متجانس برای سازه های واقعی صورت بگیرد .

در مقاله ارائه شده توسط آقای Papadrakaki از یونان یک ساختمان چند طبقه با روش RBDO طراحی شده است که هدف نهایی از طراحی آن وزن بهینه ساختمان است . همچنین سازه در برابر نیروهای طبیعی مانند باد زلزله قرار گرفته است و قیدهایی که مورد آنالیز قرار می گیرند تنشها و تغییر مکان اعضا است . میزان بار ، خصوصیات مواد و شکل هندسی اعضا نیز نامعلوم و نادقیق هستند که با روش شبیه سازی Monte Carlo Simulation (MCS) محاسبه شده اند . میزان کارایی سازه در محدوده آنالیز الاستوپلاستیک محاسبه می شود [15].

به منظور استفاده از RBDO از دو ابزار شبکه عصبی و (ES) Evaluation Strategies استفاده شده است. این ابزار قادرند پاسخ سیستم یک سازه را در مقابل قیدهای احتمالی آن که ناشی از متغیرهای مختلف طراحی است پیش بینی کنند .

همچنین شبکه های عصبی آنالیز سازه را در محدوده الاستوپلاستیک پیش بینی می کنند تا جایی که رفتار سازه به مرحله فروپاشی برسد که این آنالیز جایگزین روش MCS شده است .

مثال 5: در این مقاله یک سازه فلزی نامتقارن با در نظر گرفتن مشخصات تصادفی مورد بررسی قرار گرفته شده است.



شکل (28) ساختمان 6 طبقه نامتقارن

3D six-storey frame: Characteristics of the random variables

Random variable	Probability density function	Mean value	Standard deviation
$E$	$N$	200	$0.10E$
$\sigma_y$	$N$	25.0	$0.10\sigma_y$
Design variables	$N$	$s_j$	$0.1s_j$
Loads ( $\times 10^3$ )	Log- $N$	6.4	0.20

جدول (4) متغیرهای تصادفی سازه

در طول طراحی که هدف بدست آوردن وزن حداقل از طراحی است، از یک شبکه عصبی با ساختار (7-3-1) استفاده شده است. این سازه با روشهای RBDO, RBDO-NN1, RBDO-NN2, DBO حل شده است. مقایسه زمان مورد نیاز برای آموزش و طراحی را در جدول زیر مشاهده می کنید.

3D six-storey frame: performance of the methods

Optimization procedure	ES generations	$p_f^b$	Optimum weight (kN)	Time (h)
DBO	43	$0.171 \times 10^{-0}$	727	0.05
RBDO (500 siml.)	65	$0.105 \times 10^{-2}$	869	7.6
RBDO-NN1 (500 siml.)	64	$0.105 \times 10^{-2}$	873	2.7
RBDO-NN2 (500 siml.)	65	$0.105 \times 10^{-2}$	869	3.6
RBDO (1000 siml.)	68	$0.101 \times 10^{-2}$	875	16.3
RBDO-NN1 (1000 siml.)	69	$0.101 \times 10^{-2}$	875	5.3
RBDO-NN2 <sup>a</sup>	66	$0.97 \times 10^{-3}$	881	5.0
RBDO (5000 siml.)	68	$0.101 \times 10^{-2}$	875	81.1
RBDO-NN1 (5000 siml.)	69	$0.101 \times 10^{-2}$	875	26.5
RBDO-NN2 <sup>a</sup>	66	$0.97 \times 10^{-3}$	881	5.0

<sup>a</sup> For 100 000 simulations.

<sup>b</sup> For 100 000 simulations using the NN2 scheme.

جدول (5) مقایسه روشها

#### 4-8 کاربرد شبکه عصبی در سازه های بتنی :

طراحی سازه ها یک روند تکراری است به طوری که طراح یک فرضی را در نظر می گیرد سپس بر اساس فرض خود سازه را آنالیز می کند . طراح برای طراحی سازه نیاز به روش ها و قوانین استاندارد وضع شده دارد تا سازه فرض شده خویش را با فرض اولیه خود مقایسه کند که عموماً این دو طراحی باید با هم تفاوتی نداشته باشند . در واقعیت به ندرت این دو نوع طراحی با هم یکی می شوند . بنابراین طراح باید یک فرض جدید در نظر بگیرد و سپس روند آنالیز و طراحی را دوباره ادامه دهد . این تکرار باید آنقدر ادامه یابد تا تفاوت بین طراحی فرض شده در یک مرحله و طراحی بدست آمده از آنالیز در همان مرحله ناچیز شود . این روند طولانی ما را به تکنیک های مرسوم سوق می دهند که در بسیاری از سیستم های سازه ای موفق عمل کرده اند ، اما این تکنیک ها معمولاً طولانی و پیچیده هستند ، بنابراین برای طراحی مشکلات زیادی فراهم می کنند [10] .

یکی از روش هایی که مقدار و زمان لازم برای یک پروسه طراحی را کاهش می دهد استفاده از شبکه عصبی برای طراحی است . بنابراین یک شبکه عصبی که بر اساس یک طراحی آموزش دیده باشد بهتر و سریعتر از انجام کامل یک طراحی است .

مقاله ای که توسط آقای Hadi در سال 2003 ارائه شده است [10]، در قالب معادلات ساده بهینه سازی و طراحی یک تیر بتن آرمه با تکیه گاه ساده را شرح داده است . سپس توسط این معادلات تیرهایی با اشکال متفاوت که تحت بارگذاری های مختلف قرار گرفته اند طراحی بهینه شده و برای آموزش یک شبکه عصبی استفاده شده اند .

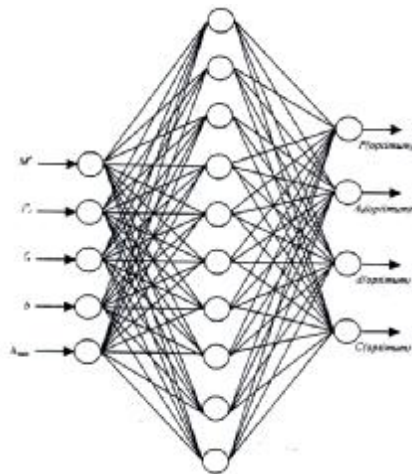
از طرفی دیگر طراحی تیرهای بتن آرمه فیبری و آموزش شبکه های عصبی مخصوص طراحی بهینه این نوع تیرها در این مقاله ارائه شده است .

برای آموزش این شبکه ها از الگوریتم آموزش پس انتشار خطا (BP) و تابع تحریک سیگموئید و نرخ آموزش 0/01 در این شبکه ها استفاده شده است.

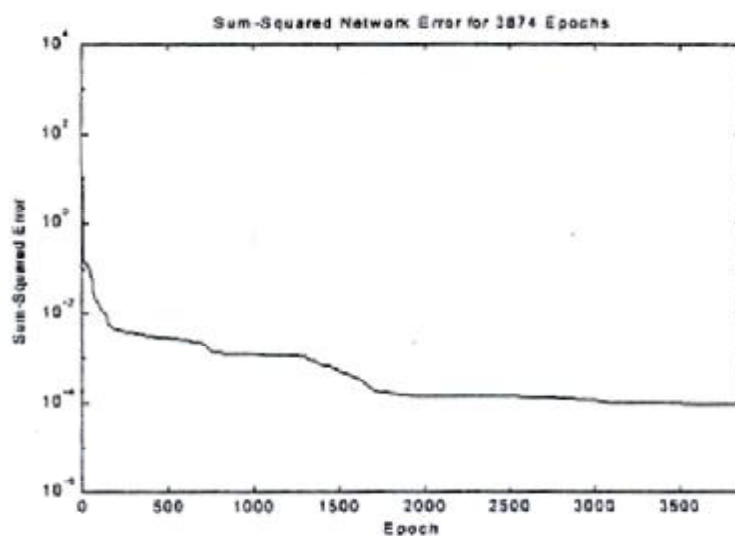
داده های ورودی که برای آموزش شبکه استفاده شده اند : ممان وارده ( $M^*$ ) ، مقاومت بتن ( $f'_c$ ) تنش جاری شدن فولاد تسلیح ( $f_y$ ) ، عرض تیر ( $b$ ) و بیشترین ارتفاع تیرها ( $h_m$ ) ، هزینه واحد بتن ( $c_c$ ) هزینه واحد فولاد تسلیح ( $c_s$ ) و هزینه واحد قالب بتن ( $C_f$ ) می باشند .

داده های خروجی برای آموزش شبکه نیز : نسبت فولاد بهینه ( $P_{optimum}$ ) ، مساحت بهینه فولاد تسلیح ( $A_{s(optimum)}$ ) ، ارتفاع موثر بهینه تیر ( $d_{optimum}$ ) و هزینه واحد بهینه تیر ( $c_{optimum}$ ) می باشند .

**مثال 6:** ابتدا در شبکه ای که آموزش داده شد از پارامترهای هزینه فولاد و بتن و قالب ها در آموزش صرف نظر شد و مقادیر آنها ثابت فرض شد . تعداد 550 نمونه برای آموزش و 50 نمونه برای آزمایش شبکه در نظر گرفته شد و تعداد نرون های ورودی شبکه 5 و تعداد نرون های لایه پنهان 10 و تعداد نرون های لایه خروجی 4 بود و خطای متوسط در هنگام آزمایش شبکه 2/71% بدست آمد .

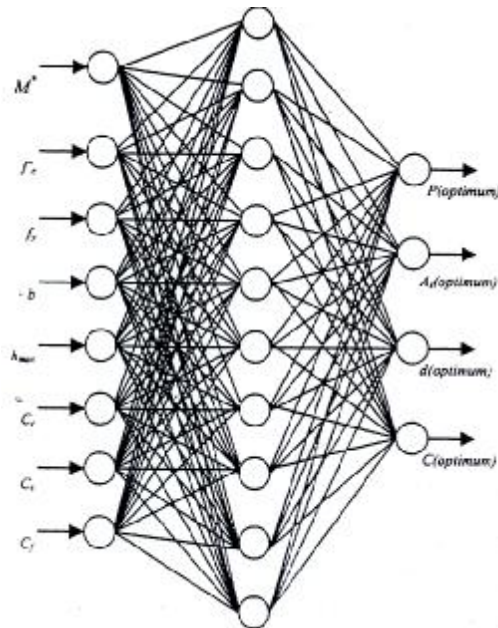


شکل (29) مدل عصبی نمونه اول را نشان می دهد



شکل (30) نمودار همگرایی میانگین مربعات خطا را در مدت آموزش نشان می دهد

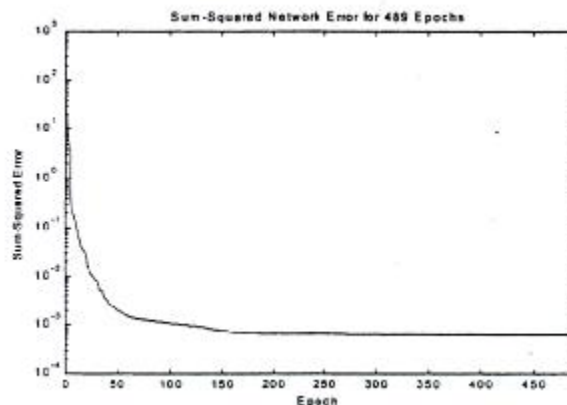
**مثال 7:** شبکه دیگر با در نظر گرفتن هزینه فولاد و بتن و قالب بتن در داده های ورودی آموزش دیده شد که متوسط هزینه ها به ترتیب فولاد و بتن و قالب ها با 10% اختلاف قیمت واقعی بدین شرح است:  $\$1075/t$ ،  $\$68/m^2$  و  $\$129/m^3$ ، این شبکه با 550 نمونه آموزش دید و با 50 نمونه مورد آزمایش قرار گرفت که منجر به خطای متوسط 1/6% شد.



شکل(31) مدل شبکه عصبی نمونه دوم را همراه با داده های هزینه اضافه شده نشان می دهد

**مثال 8:** در شبکه سوم آموزش دیده ای که به منظور طراحی بتن فیبری طراحی شده است 14 داده ورودی به منظور آموزش به شبکه داده شده است که علاوه بر ممان وارد بر مقاومت بتن و فولاد، داده های دیگر از این قرارند: برش تیر، مساحت فولاد کششی حدس زده اولیه ( $A_2$ )، طول فیبر فولادی، قطر فیبرها و نسبت مقدار فیبرهای فولادی به میزان بتن، ماکزیمم و مینیمم ارتفاع تیر، مینیمم و ماکزیمم عرض تیر، حداکثر و حداقل میزان فولاد فیبرها در بتن. شش داده خروجی شبکه که متغیرهای طراحی می باشند بدین شرحند: فولاد کشش تسلیح بهینه، عرض بهینه تیر، ارتفاع موثر تیر و فاصله بین فولادها و مقاومت نهایی خشی تیر و هزینه نهایی تیرها.

604 نمونه برای آموزش در نظر گرفته شد و با استفاده از الگوریتم آموزش پس انتشار خطا و تابع تحریک سیگموئیدی شبکه در طول 450 دور سعی و خطا آموزش یافت. این شبکه با 100 نمونه آزمایش شد که خطای متوسط 6/68% از طراحی موجود بدست آمد.



شکل(32) همگرایی مدل بتن فیبری را نشان می دهد

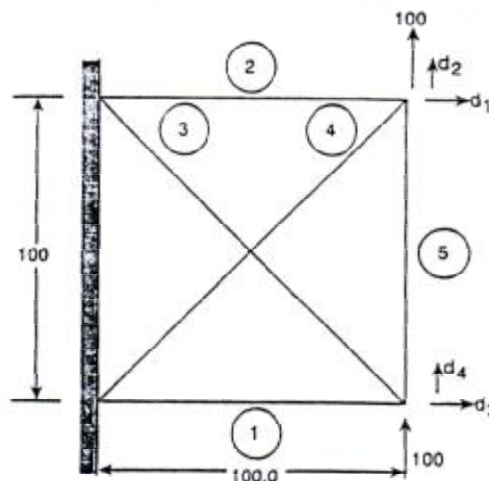
#### 9-4 مدل های محاسبات عصبی در آنالیز و طراحی سازه ها :

HAJELA و BERKE در سال 1990 مقاله ای ارائه کرده اند که در آن روابط نیرو-تغییر مکان در آنالیز استاتیکی سازه را به کمک شبکه عصبی می توان بدست آورد. از این شبکه در طراحی و آنالیز استفاده کرده اند سپس مدل به دست آمده را می توان برای طراحی بهینه عددی به کار برد [12].

آن ها سه نمونه سازه برای آموزش و آزمایش انتخاب کردند. خرپاهای 5 عضو و 10 عضو و یک سازه Wing-Box برای آنالیز و طراحی بهینه انتخاب شدند.

آزمایشی که بوسیله خرپای دارای 5 عضو انجام دادند نشان می داد که یک شبکه بدون لایه پنهان نسبت به یک شبکه با یک لایه پنهان نتایج ضعیفی ارائه می کند و زمان همگرایی آن طولانی تر است. هنگام آموزش 50 زوج آموزشی و با 1150 دور زمان آموزش، اختلافی بین 1% الی 8/7% خطا در شبکه تک لایه (بدون لایه پنهان) به وجود آمده است. ولی با تغییر ورودی ها و افزایش و تجزیه آن ها و همچنین افزایش زمان آموزش تا 50000 دور مقدار خطا به محدوده (3/9\_5/2)% خطا کاهش یافت و این خطا با افزایش ورودی ها به 0/39% نیز کاهش یافته است [12].

نتایجی که از شبکه با یک لایه پنهان بدست آمده است قابل قبول می باشد که نتایج آن را می بینید.



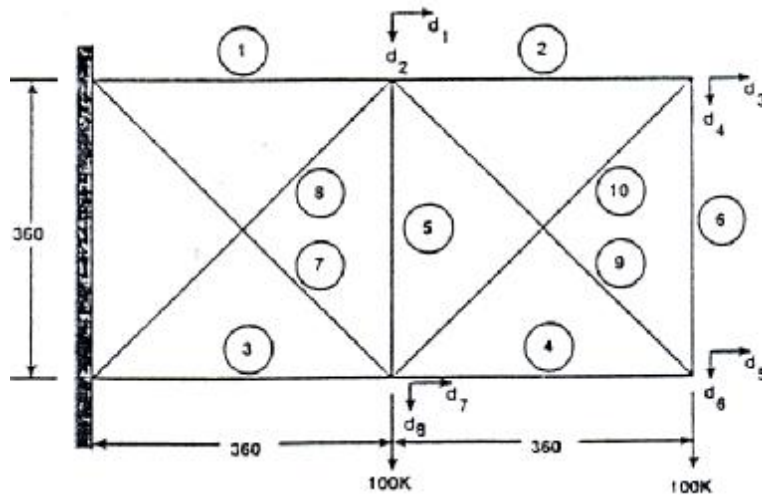
شکل(33)خرپای با 5 المان

Optimal designs for the five-bar truss using a neural network for analysis

		Design variables					Objective functions
		$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	
(3-3-4) 200 training sets, all four output displacements mapped.	Initial	1.0	1.0	1.0	1.0	1.0	58.28
	Final	2.131	2.032	2.679	2.766	1.0	128.626
(3-1-4) 500 training sets, all four output displacements mapped.	Initial	1.0	1.0	1.0	1.0	1.0	58.28
	Final	1.952	2.013	2.763	2.760	1.0	127.759
(5-7-2) 100 training sets, two vertical displacements mapped.	Initial	1.0	1.0	1.0	1.0	1.0	58.28
	Final	1.535	1.778	2.29	2.265	1.0	107.56
(5-7-2) 100 training sets, two vertical displacements scaled as constraints	Initial	1.0	1.0	1.0	1.0	1.0	58.28
	Final	1.505	1.584	2.138	2.211	1.0	102.399
	Exact Solution	1.5	1.5	2.121	2.121	1.0	100.0

جدول(6) نتایج خرپای 5 عضوی را با ساختار های متفاوت نشان می دهد

یک خرپای دیگر با 10 عضو برای آموزش شبکه استفاده شده است. هدف، بدست آوردن وزن بهینه در این سازه با وجود حداکثر جابجایی استاتیکی 2 اینچ در جهت عمودی می باشد. نتایج آزمایشگاهی به شبکه عصبی با ساختار (8-11-10) به شبکه آموزش داده شده اند در این شبکه از یک لایه پنهان با 11 نرون و 10 ورودی و 8 نرون خروجی استفاده شده است، 8 نرون خروجی معادل درجات آزادی خرپا است.



شکل (34) خرپای 10 عضوی

100 نمونه آموزشی به ازای مساحت هر یک از اعضای خرپا در محدوده  $in^2$  (0/1 و 32) به شبکه داده شده است.

محدوده داده خروجی یا همان تغییر مکان گرهی نیز بین  $in$  (-0/923 و 7/495) تغییر می کند. این شبکه با خطایی معادل 1/5% و در زمانی حدود 2/5 ساعت آموزش دیده است. شبکه های مختلف دیگری با ساختارهای مختلف برای طراحی این سازه خرپایی آموزش دیده اند که زمان آموزش و میزان خطای آن ها را در قالب یک جدول مشاهده کنید.

Network description	Range in which training sets are randomly generated in <sup>2</sup>		Training data		Constraint evaluations at known optimum		Error description
	max	min	cpu	iterations	$g_1$	$g_2$	
(10-11-8)†	32.0	0.1	2 hr 22 min	45788	1.059	1.141	$\epsilon = 0.015$
(20-8)†	32.0	0.1	40 min	28663	—	—	$\epsilon = 0.169$
(10-11-2)	32.0	0.1	2 hr 17 min	13600	1.342	1.531	$\epsilon = 0.017$
(10-6-6-2)	32.0	0.1	31.5 min	13220	0.859	0.841	$\epsilon = 0.01$
(10-6-6-2)	45.0	0.1	30.5 min	12560	0.521	0.533	$\epsilon = 0.01$
(10-6-6-2)	55.0	0.01	22 min	10210	0.391	0.416	$\epsilon = 0.01$
(10-6-6-2)	55.0	0.01	23.2 min	10209	-0.009	0.015	$\epsilon = 0.01$
(10-6-6-2)	25% of optimum	—	1.83 min	819	-0.066	-0.067	$\epsilon = 0.01$
(10-6-6-2)§	25% of optimum	—	11.1 min	1322	-0.0547	-0.0581	$\epsilon = 0.01$

جدول (7) نتایج خرپای 10 عضوی و مقایسه زمان آموزش ساختارهای شبکه عصبی

همانطور که در جدول نتایج ملاحظه می کنید، خطای موجود در شبکه حدود 0.01 بدست آمده است. همچنین زمان آموزش شبکه با ساختارهای مختلف در این جدول نشان داده شده است که با اضافه کردن یک لایه پنهان به شبکه سرعت آن به صورت قابل ملاحظه ای افزایش یافته است. با توجه به جدول وجود دو لایه پنهان با ساختار (10-6-6-2) تعداد نرون در هر لایه زمانی در حدود 22 دقیقه برای آموزش صرف کرده است.

### نتیجه گیری :

شبکه های عصبی همان طور که نشان داده شد از توانایی های زیادی در مسائل مهندسی به خصوص در طراحی سازه برخوردار است . به همین دلیل نظر طراحان را به سوی خود جلب کرده است و در آینده این شبکه ها جای خود را در مسائل مهندسی محکم خواهند کرد و به عنوان یک مرجع قابل اطمینان مورد استفاده قرار خواهند گرفت ، به شرطی که دقت زیادی در ارائه داده های طراحی شود . از این شبکه ها می توان در شناسایی خرابی سازه ، طراحی دیوارهای برشی فلزی و بتنی ، طراحی صفحات زیر ستون ، طراحی اتصالات صلب در بتن های پیش تنیده یا پس کشیده و یا در طراحی تیر ورق ها در صنعت پل سازی بهره برد .

---



### فهرست منابع:

- [1] محمد باقر منهاج (1381)، "هوش محاسباتی (جلد اول) مبانی شبکه های عصبی"، مرکز نشر دانشگاه امیر کبیر
- [2] محمد باقر منهاج، نوید سیفی پور (1377)، "هوش محاسباتی (جلد دوم) کاربرد هوش محاسباتی در کنترل"، مرکز نشر پرفسور حسابی
- [3] علی کاوه، عباس ایران منش (1378)، "شبکه های عصبی مصنوعی در بهینه سازی سازه ها"، مرکز تحقیقات ساختمان و مسکن، نشریه شماره ک-310
- [4] علی کاوه، همایون ثروتی (1379)، "شبکه های عصبی مصنوعی در تحلیل و طراحی سازه"، مرکز تحقیقات ساختمان و مسکن، نشریه شماره ک-320
- [5] علی کاوه، عباس ایران منش، "بهینه سازی سازه ها با استفاده از شبکه های عصبی مصنوعی" امیر کبیر، سال دهم، شماره 10
- [6] علی کاوه، حسینعلی بندرآبادی، "مدلسازی سیستم های سازه ای ناشناخته با استفاده از شبکه های عصبی"، تازه های ساختمان و مسکن، شماره 14
- [7] ابوالفضل عربزاده، فریدون امیدی نسب، "پیش بینی مقاومت نهایی برشی تیرهای عمیق بتن مسلح دو سر ساده با استفاده از شبکه های عصبی مصنوعی"، ششمین کنفرانس عمران اصفهان 1382
- [8] M.Y. Rafiq, G. Bugmann, D.J. Easterbrook: "Neural network design for engineering applications", computers & structures 79 (2001) 1541-1552
- [9] Hojjat Adeli and Asim Karim: "Neural network model for optimization of cold-form steel beams", ASCE, Journal of structural engineering, Nov. 1997, Vol. 123, No. 11
- [10] Muhammad N.S. Hadi: "Neural network applications in concrete structures", computes & structures 81 (2003) 373-381
- [11] A. Tashakori, H. Adeli: "Optimum design of cold-form steel space structures using neural dynamics model", Journal of constructural steel research 58 (2002) 1545-1566
- [12] P. Hajela and L. Berke: "Neurobiological computational models in structural analysis and design", Computers & Structures, Volume 41, Issue 4, 1991, Pages 657-667
- [13] Farzad Ahmadkhanlou, Hojjat Adeli: "Optimum cost design of reinforced concrete slabs using neural dynamics model", Engineering Applications of Arti.cial Intelligence 18 (2005) 65-72

- 
- [14] A. Sanad and M.P. Saka: "Prediction of ultimate shear strength of reinforced concrete deep beams using neural networks", ASCE, Journal of structural engineering, Vol. 127, No. 7, July, 2001
- [15] M. Papadrakakis, N.D. Lagaros, V. Plevris: "Design optimization of steel structures considering uncertainties", Engineering Structures, April 2005, Article in press
- [16] C. Jayatheertha, J.P.H. Webber and S.K. Morton: "Application of artificial neural networks for the optimum design of a laminated plate", Computers & Structures, 1996, Vol. 59, No. 5, 831-845
- [17] E.M.A El-Kassas, R.I. Mackie, A.I. El-Sheikh: "Using neural networks in cold-formed steel structures", Computers & Structures, 79 (2001) 1687-1696
- [18] D. Anderson, E.L. Hines, S.J. Arthur and E.L. Eiap: "Application of artificial neural networks to the prediction of minor axis steel connections", Computers & Structures, 1997, Vol. 63, No. 4, 685-692
- [19] L. Berke, S.N. Patnaik and P.L.N. Murthy: "Optimum design of aerospace structural components using neural networks", Computers & Structures, 1993 Vol. 48, 1001-1010
- [20] P. C. Pandey and S. V. Barai: "Multilayer perceptron in damage detection of bridge structures", Computers & Structures, Volume 54, Issue 4, 17 February 1995, Pages 597-608
- [21] A. Mukherjee and J. M. Deshpande: "Application of artificial neural networks in structural design expert systems", Computers & Structures, Volume 54, Issue 3, 3 February 1995, Pages 367-375
- [22] H. Adeli and Hyo Seon Park : " A neural dynamics model for structural optimization-theory", Computers & Structures, 1994, Volume 57, No.3, P.383-390
- [23] H. Adeli and Hyo Seon Park : " A neural dynamics model for structural optimization-application to plastic design of structures", Computers & Structures, 1994, Volume 57, No.3, P.391-399
-