بسم الله الرحمن الرحيم

دانشگاه صنعتی شاهرود

دانشکده عمران و معماری

عنوان پایان نامه :

بهینه یابی خرپا با بکار گیری شبکه های عصبی مصنوعی

این پایان نامه جهت دریافت درجه کارشناسی ارشد

در مهندسی عمران گرایش سازه می باشد .

استاد راهنما :

دکتر محسن طالب زاده

استاد مشاور :

دکتر وحید رضا کلات جاری

دانشجو:

مهدی جامکلو

با تشکر فراوان از استاد محترم آقای

دکتر محسن طالب زاده و

کلیه کسانی که مرا در تهیه و تنظیم این مهم

یاری نمودند .

خلاصه

در این پایان نامه پس از معرفی شبکه های عصبی مصنوعی و چگـونگی شـکل گیـری آنهـا ، بـه تشریح محاسبات عصبی پرداخته ایم . مزیت محاسبات عصبی سادگی و در عـین حـال کارآمـدی آنها میباشد که از محاسبات بیولوژیکی الگو برداری شده است و هدف ، اهمیت دادن به این شکل محاسبات و شناسایی توانمندی آنها در مهندسی عمران می باشد .

برای بهینه یابی خرپا توسط شبکه عصبی ، دو شبکه عصبی انتشار برگشتی و انتشار متقابل ارائـه شده اند که برای استفاده ابتدا باید شبکه عصبی را جهت یادگیری حل مساله مورد نظر، آمـوزش داد . با استفاده از الگوریتم آموزش بهمراه زوج های آموزشی ، شبکه را آمـوزش داده تـا شـرایط پاسخگویی به تحلیل خرپا و مساله بهینه یابی راداشته باشـد سـپس بهینـه سـازی خرپـا توسـط شبکه عصبی انتشار متقابل انجام شده و پاسخ شبکه عـصبی (روش تقریبـی) بـا پاسـخ حاصـل از روش دقیق (کمپلکس) مقایسه گردیده است .

کلمات کلیدی : شبکه عصبی – بهینه یابی – خرپا

فهرست مطالب

عنوان مطالب — صفحه

تشکروقدردانی ---------- أ

خلاصه فارسی ---------- ب

فهرست مطالب ---------- ت

فصل اول – مقدمه

۱-۱ شبکه های عصبی بیولوژیکی ---------- ۲

۱-۲ شبکه های عصبی مصنوعی ---------- ۳

۱-۳ ساختار شبکه های عصبی مصنوعی ---------- ۷

۱-۴ شبکه های تک لایه و چند لایه ---------- ۸

۱-۵ تاریخچه شبکه های عصبی ---------- ۹

فصل دوم – محاسبات عصبی

۲-۱ محاسبات در شبکه های عصبی مصنوعی ---------- ۱۴

۲-۲ تعریف اصطلاحات مهم در شبکه های عصبی مصنوعی ---------- ۱۵

۲-۳ طراحی شبکه عصبی مصنوعی ---------- ۱۵

۲-۴ آموزش شبکه های عصبی ---------- ۱۷

۲-۵ توابع تحریک متداول مورد استفاده در شبکه های عصبی ---------- ۱۸

۲-۶ بایاس وآستانه تحریک ---------- ۲۱

فصل سوم – شبکه های عصبی انتشار برگشتی (پیش خور)

و انتشار متقابل

۳-۱ ساختار شبکه عصبی انتشار برگشتی ---------- ۲۴

۳-۲ الگوریتم آموزش شبکه ---------- ۲۵

۳-۳ شیوه تنظیم وزن ها ---------- ۲۷

۳-۴ مراحل تنظیم وزن ها و بایاس ها ---------- ۳۱

۳-۵ شبکه عصبی انتشار متقابل(CPN) ---------- ۳۸

۳-۶ آموزش شبکه عصبی انتشار متقابل ---------- ۴۰

۳-۷ مراحل آموزش شبکه عصبی انتشار متقابل ---------- ۴۵

فصل چهارم – مقدمه ای بر بهینه سازی

۴-۱ کلیات ---------- ۴۸

۴-۲ روش بهینه سازی ---------- ۵۰

۴-۳ روش های بهینه سازی مقید ---------- ۵۳

۴-۴ بهینه سازی هندسی ---------- ۶۰

فصل پنجم – بهینه یابی سه خرپا توسط شبکه عصبی انتشار متقابل

۵-۱ مقدمه ---------- ۶۳

۵-۲ بهینه یابی خرپا با استفاده از شبکه عصبی انتشار متقابل ---------- ۶۶

۵-۳ بهینه یابی یک خرپای نامعین ---------- ۶۹

۵-۴ بهینه یابی یک خرپای معین ---------- ۷۹

۵-۵ بهینه یابی خرپای متقارن ۲۵ عضوی ---------- ۸۵

نتیجه گیری و پیشنهادات

نتیجه گیری ---------- ۹۲

پیشنهادات ---------- ۹۳

فهرست منابع و مأخذ

فهرست منابع و مأخذ ---------- ۹۴

پیوست

پیوست شماره یک ---------- ۹۶

پیوست شماره دو ---------- ۱۰۷

پیوست شماره سه ---------- ۱۱۷

فصل اول

مقدمه

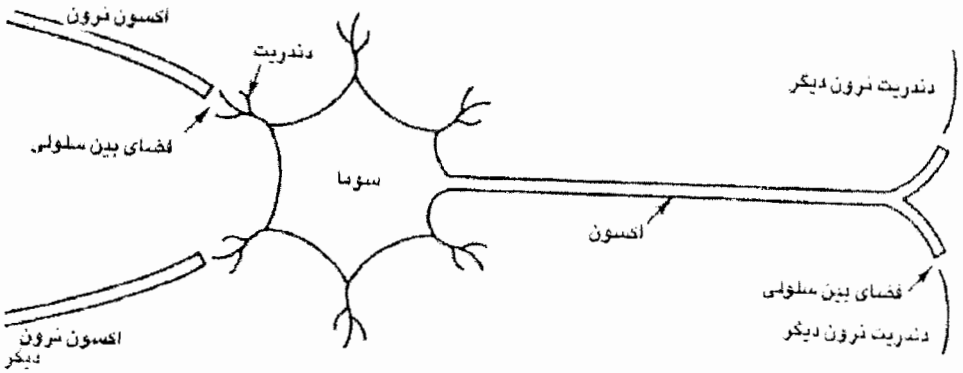
۱-۱ شبکه های عصبی بیولوژ یکی

یک نرون بیولوژیکی دارای سه جزء دندریت[۱] ، سوما[۲] و اکسون[۳] است که شناخت آنها در درک مفاهیم شبکه های عصبی مصنوعی ازاهمیت خاصی برخوردار می باشد . تعداد زیادی دندریت علائم را از سایر نرون ها دریافت می دارند . علائم ، پالس های الکتریکی هستند که از طریق واکنش های شیمیایی فضای بین سلولی انتقال می یابند . انتقال دهنده های شیمیایی ، علائم دریافتی را اصلاح می نمایند . سوما (بدنه واحد پردازشگر) ، علائم ورودی را جمع می نماید اگر مجموع ورودی ها از یک حد آستانه فراتر رود ، در آن صورت پردازشگر فعال می شود ، بدین معنی که از طریق اکسون علائمی را به سلول بعدی انتقال می دهد . تصور می گردد که یک واحد در یک لحظه زمانی یا فعال است یا بالعکس غیر فعال باقی می ماند بنابراین وضعیت آنرا می توان با سیستم دودویی مشخص نمود . اما فرکانس علائم خروجی متغیر بوده و دارای مقادیر متفاوت است .

انتقال علائم از یک نرون بواسطه نابرابر بودن تمرکز یون ها در ابتدا وانتها ی بدنه اکسون صورت می گیرد . یونهای غالب در انتقال پیام شامل سدیم ، پتاسیم و کلرید می شوند . یک نرون بیولوژیکی در شکل (۱-۱) همراه با اکسون ودو نرون دیگر (که نرون نشان داده شده از آنها اطلاعات دریافت می کند) باضافه دندریت دو نرون دیگر (که از نرون نشان داده شده اطلاعات دریافت می کند) نشان داده شده است .

---

۱ . Dendrite
۲ . Soma
۳ . Axon



شکل (۱-۱) : نرون بیو لوژیکی

یکی دیگر از خصوصیات مهمی که شبکه های عصبی بیولوژیکی تحمل خطا است . دو نمونه زیر مثالهایی از توانایی شبکه های بیولوژیکی عصبی در تحمل خطا هستند . نمونه اول تمیز اطلاعات جدید از اطلاعات قبلی است . بعنوان مثال ما قادر هستیم از روی عکس شخصی ، تشخیص دهیم که او را تاکنون ندیده ایم و یا اینکه شخصی را که برای مدت زمان طولانی ندیده ایم ، شناسایی نماییم . نمونه دوم تحمل ضایعه در شبکه عصبی است . انسانها با شبکه عصبی مشتمل بر یکصد بیلیون نرون عصبی متولد می گردند اکثر این نرون ها در مغز هستند و اکثر آنها پس از مرگشان جانشین نمی گردند . اما علیرغم از دست دادن متوالی نرون ها ما بطور دائم آموزش می بینیم و سایر نرون ها می توانند در بسیاری از موارد آموزش دیده و جای نرون های از بین رفته را پر نمایند .

۱-۲ شبکه های عصبی مصنوعی

شبکه های عصبی مصنوعی که امروزه در کاربردهای فراوانی ارزش بالای خود را نشان داده اند ، بر اساس مدل بیولوژیکی مغز جانوران بوجود آمده اند . در مغز انسان حدود $۱۰^{۱۰}$ واحد سازنده ،

بنام نرون[1] وجود دارد و هر یک از این نرونها به 10^4 نرون دیگر اتصال دارد و هرکدام دارای کارکردی به صورت ورودی یا خروجی است که مغز انسان را به ماهیچه ها مرتبط می سازد و یا علایم را از ارگانهای حساس به مغز انتقال می دهد . شبکه های عصبی مصنوعی ممکن است از چند نرون تا چند هزار نرون تشکیل شده باشند و اندازه شبکه بستگی به پیچیدگی مسئله دارد. نرون ورودی هایی را که به طریق خاصی جمع می شوند ، پذیرا می گردد . اگر چنانچه این ورودیها به اندازه کافی بزرگ باشند ، در آن صورت نرون فعال شده و اگر فعالیت نرون کمتر از آستانه[2] از پیش تعیین شده باشد در آن صورت نرون غیر فعال باقی می ماند . در یک لحظه بخصوص ورودی های جمع شده در یک نرون به یک تابع تحریک که یک خروجی مشخصی را محاسبه می کند ، انتقال می یابند . سپس علایم خروجی به لایه دیگری از نرون ها و یا به خروجی شبکه فرستاده می شوند . [۱]

شبکه های عصبی مصنوعی بواقع یک سیستم داده پردازی اطلاعات است و دارای خصوصیات اجرایی خاصی همانند شبکه های عصبی جانوران می باشد . شبکه های عصبی مصنوعی از تعمیم یافتن مدلهای ریاضی شبکه های عصبی جانوران بر اساس فرضیات زیر توسعه یافته اند :

۱- داده پردازی اطلاعات در اجزا ء ساده بنام نرون صورت می گیرد .

۲- اطلاعات بین نرون ها از طریق ارتباطات بین آنها رد و بدل می شود .

۳- هر یک از این رابط ها دارای مشخصه ای (وزن) مختص به خود هستند که در اطلاعات انتقال یافته از یک نرون به نرون دیگر ضرب می گردند .

۴- هر یک از نرونها برای محاسبه خروجی خود ، یک تابع تحریک[3] را که معمولا" غیر خطی

---

۱ . Neuron
۲ . Threshold
۳. Activation Function

است به ورودیهایش (جمع اطلاعات وزن دار شده) اعمال می نماید .

خصوصیات زیر مشخص کننده یک شبکه عصبی مصنوعی هستند :

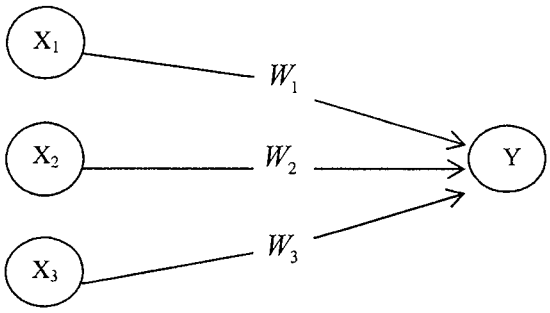
۱- طریقه ارتباط بین نرون ها (ساختار شبکه)

۲- تعیین مقادیر وزن های رابط نرون ها (روش آموزش شبکه)

۳- تابع تحریک هر نرون

هر شبکه عصبی مشتمل بر مجموعه ای از اجزا ء کوچک داده پردازی بنام نرون ، واحد و یا گره می باشد و هر نرون به نرون های دیگر از طریق یک رابط جهت دار که دارای وزن مختص به خود است ، مرتبط می گردد .

وزنها ، نمایش دهنده اطلاعات مورد نیاز شبکه برای حل یک مسئله هستند . شبکه های عصبی را می توان در موارد گوناگونی از جمله ذخیره کردن و بازبینی داده ها ، گروه بندی شکل ها ، انجام یک نگاشت کلی از یک مجموعه ورودی به یک مجموعه خروجی و بهینه سازی و تعیین جواب با وجود قیود متعدد بکار گرفت . در یک شبکه عصبی هر نرون وضعیت مشخصی دارد که بستگی به ورودیهایی دارد که دریافت نموده است . عموما" هر نرون پاسخ خود را به نرون یا نرون های دیگر ارسال می دارد . بعنوان مثال نرون Y نشان داده شده در شکل (۱-۲) داده هایی را از نرون های X_1، X_2 و X_3 دریافت می نماید که به ترتیب با x_1 ، x_2 و x_3 نمایش داده می شوند . این داده ها را پاسخ تحریک (جواب خروجی) نرون ها می نامند .



شکل (۱-۲) : نرون مصنوعی ساده

وزن رابط های بین نرون های X_1 ، X_2 و X_3 و نرون Y به ترتیب W_1، W_2و W_3 می باشد . ورودی خالص y_{in} به نرون Y برابر مجموع داده های وزن دار شده از نرون های X_1، X_2 و X_3 است (رابطه ۱-۱) .

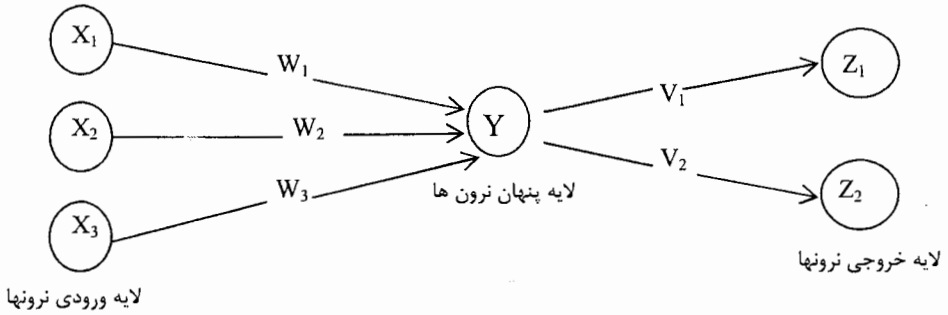
$$y_{in} = W_1.x_1 + W_2.x_2 + W_3.x_3 \qquad (۱-۱)$$

پاسخ تحریک y مربوط به نرون Y به صورت تابعی از ورودی خالص این نرون بیان می گردد $y = F(y_{in})$ بطوریکه از تابع تحریک سیگموئیدی (رابطه (۱-۲)) برای تعیین پاسخ تحریک ، y استفاده میشود .

$$F(y_{in}) = 1/(1+\exp(-y_{in})) \qquad (۱-۲)$$

اکنون فراتر رفته و تصور می نماییم که نرون Y به نرون های Z_1و Z_2 به ترتیب با وزن های v_1 و v_2مرتبط گردد . همانطوریکه در شکل (۱-۳) نشان داده شده است نرون Y خروجی y را به هر یک از این نرون ها ارسال میدارد ، اما عموما" مقادیر دریافتی نرون های Z_1و Z_2 متفاوت خواهد بود ، زیرا خروجی y با توجه به مقادیر V_1 و V_2 وزن دار می شود . در یک نمونه کلی تر شبکه عصبی ، پاسخ تحریک z_1 و z_2 مربوط به نرون های Z_1و Z_2 به ورودیهای آنها از چندین

نرون بستگی دارد نه فقط یک نرون .



شکل (۱-۳) : یک ساختار ساده شبکه عصبی

یک شبکه عصبی حداقل دارای یک لایه ورودی نرون ها و یک لایه خروجی نرون ها است و همچنین می تواند دارای لایه های پنهان یا میانی نرون ها باشد . وجود لایه های پنهان در شبکه توأم با تابع تحریک غیر خطی موجب افزایش توانایی آن می شود بطوریکه بسیاری از مسائل قابل حل با آن ، با شبکه هایی که تنها شامل لایه های ورودی و خروجی نرون ها هستند قابل حل نمی باشند . در مقابل ، آموزش[1] شبکه حاوی لایه های میانی و تعیین مقادیر بهینه وزنها مشکل است .

۱-۳ ساختار شبکه های عصبی مصنوعی

عموما" نرون های یک لایه بطور یکسان عمل می کنند و پارامتر اصلی در تعیین رفتار یک پردازشگر(نرون) عبارتند از :

الف : تابع تحریک پردازشگر (نرون)

ب : رابط های وزن داری که اطلاعات از طریق آنها بین نرون ها رد و بدل می شود .

[1] – بحث آموزش شبکه در فصل دوم تشریح خواهد شد .

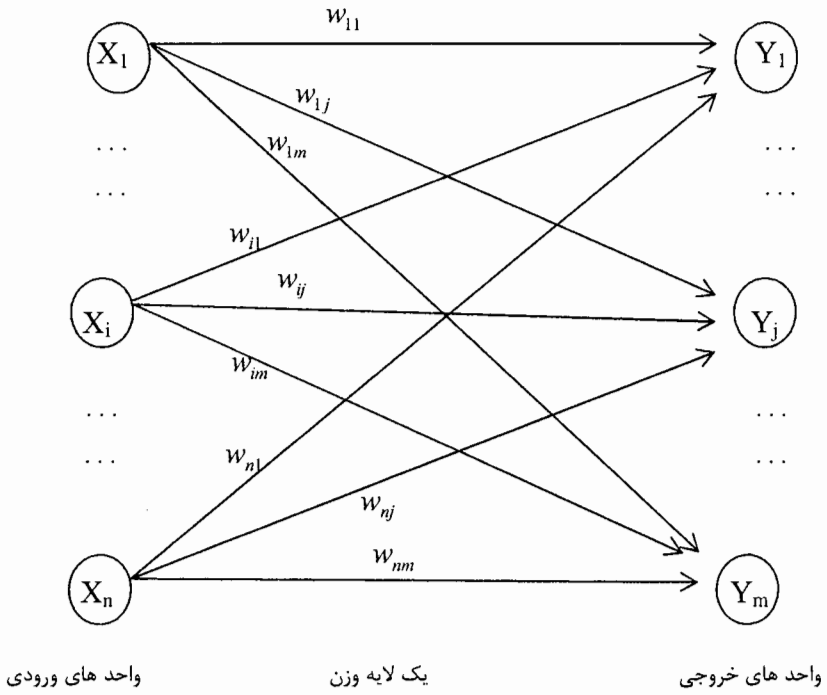
در شبکه های عصبی معمولا" در هر لایه از نرون ها تابع تحریک و همچنین طریقه ارتباط آنها به سایرنرون ها با هم یکسان است . در شبکه های عصبی نرون های یک لایه یا بطور کامل مرتبط هستند و یا اصلا" ارتباط ندارد . اگر هر نرون از یک لایه به نرونی از لایه دیگر مرتبط باشد ، می توان گفت نرون های لایه اول به نرونهای لایه دوم مر تبط شده است . ترتیب نرون ها در لایه ها و طریقه ارتباط در داخل لایه ها وبین لایه ها ، ساختار شبکه نامیده می شود .

اغلب شبکه های عصبی به صورت یک یا چند لایه گروه بندی می شوند . در تعیین تعداد لایه ها ، لایه نرون های ورودی بعنوان لایه محسوب نمی گردد ، زیرا هیچ گونه محاسبه ای در آن انجام نمی گیرد . بعبارت دیگر تعداد لایه های یک شبکه برابر با تعداد ارتباط های وزن دار بین لایه ها می باشد . این نامگذاری بدان جهت رعایت می شود که وزن ها دریک شبکه حامل اطلاعات بسیار مهمی هستند . شبکه نشان داده شده در شکل (۱-۳) شامل دو لایه وزن دار ، لایه نرون پنهان و لایه نرون های خروجی ، می باشد .

۱-۴ شبکه های تک لایه و چند لایه

یک شبکه تک لایه ای شامل یک لایه با ارتباط های وزن دار است . نرون های لایه ورودی علائم را از خارج شبکه دریافت می دارند و نرون های خروجی پاسخ شبکه را تحویل می دهند .

بعنوان مثال نمونه ای از شبکه های تک لایه در شکل (۱-۴) نشان داده شده است که نرون های لایه ورودی به طور کامل به واحدهای خروجی مرتبط گردیده اند اما هیچگونه ارتباطی بین نرون های لایه ورودی وجود ندارد . در ضمن نرون های لایه خروجی نیزبه یکدیگر مرتبط نگردیده اند .



شکل (۱-۴) : شبکه عصبی تک لایه

یک شبکه چند لایه شامل یک و یا بیش از یک لایه از نرون های پنهان بین نرون های لایه ورودی و خروجی می باشد . عموما" مجموعه ای از وزن ها بین دو لایه مجاور هم وجود دارد . شبکه های چند لایه در مقایسه با شبکه های یک لایه قادر به حل مسائل پیچیده تری هستند اما آموزش اینگونه شبکه ها مشکل تر می باشد . با این حال در بسیاری از موارد با آموزش ، شبکه قادر به حل مسائلی می گردد که از عهده شبکه های تک لایه ای خارج است .

۱-۵ تاریخچه شبکه های عصبی

بعضی از پیش زمینه های شبکه های عصبی را می توان به اوایل قرن بیستم و اواخر قرن نوزدهم برگرداند . در این دوره کارهای اساسی در فیزیک ، روانشناسی و نروفیزیولوژی توسط

دانشمندانی چون هرمان فون هلمهلتز[۱] ، ارنست ماخ[۲] و ایوان پاولف[۳] صورت پذیرفت . این کارهای اولیه عموما" بر تئوریهای کلی یادگیری ، بینـایی و شـرطی تاکیـد داشـته انـد و اصـلا" بـه روی مدلهای مشخص ریاضی عملکرد نرونها اشاره ای نداشته اند .

دیدگاه جدید شبکه های عصبی در دهه چهل قرن بیستم شروع شد ، زمانیکه مک کلـوث[۴] و والتر پیتز[۵] نشان داداند که شبکه های عصبی در اصل می تواننـد هـر تـابع حـسابی و منطقـی را محاسبه نمایند . کار این افراد را می توان نقطه شروع حوزه علمی شـبکه هـای عـصبی مـصنوعی نامید و این موضوع با دونالد هب[۶] ادامه یافت . کسی که عمل شرط گـذاری مکانیـسمی را جهـت یادگیری نرونهای بیولوژیکی ارائه داد .

نخستین کاربرد عملی شبکه های عصبی در اواخر دهـه قـرن بیـستم مطـرح شـد ، زمانیکـه روزنبلات در سال ۱۹۵۸ شبکه پرسپترون را معرفـی نمـود . روزنـبلات[۷] و همکـارانش شـبکه ای ساختند که قادر بود الگوها را از هم شناسایی نماید . در همین زمان بود کـه ویـدروهف در سـال ۱۹۶۰ شبکه عصبی تطبیقی آدلاین را با قانون یادگیری جدید مطرح نمود کـه از لحـاظ سـاختار شبیه شبکه پرسپترون می باشد .

هر دوی این شبکه ها ، پرسپترون و آدلاین ، دارای این محدودیت ذاتی بودند که توانایی طبقه بندی الگوهایی را داشته اند که بطور خطی از هم متمایز باشند . ویدرو و روزنبلات هـر دو از ایـن امر آگاه بودند . چون آنها قانون یادگیری را برای شبکه های عصبی تک لایه مطرح نمـوده بودنـد

---

۱. Hermann von helmholtz
۲. Ernst mach
۳. Ivan pavlov
۴. Mc culloch
۵. Walter pitts
۶. Donald hebb
۷. Rosenblat

که توانایی محدودی جهت تخمین توابع داشتند . هر چند آنها توانستند شبکه های چند لایه را مطرح نمایند لیکن نتوانستند الگوریتمهای یادگیری شبکه های تک لایه شان را بهبود بخشند .

پیشرفت شبکه های عصبی تا دهه هفتاد قرن بیستم ادامه یافت . در ۱۹۷۲ کوهونن[۱] و اندرسون[۲] بطور مستقل و بدون اطلاع از هم شبکه های عصبی جدید را معرفی نمودند که قادر بودند به عنوان عناصر ذخیره ساز عمل نمایند . گروسبرگ[۳] در این دهه روی شبکه های خود سازمان ده فعالیت می کرد . علاقه به کار کردن روی شبکه های عصبی در دهه ۶۰ قرن بیستم در قیاس با دهه هشتاد بعلت عدم بروز ایده های جدید و نبود کامپیوترهای سریع جهت پیاده سازی کمرنگ می نمود . لیکن در خلال دهه هشتاد ، رشد تکنولوژی میکروپروسسورها روند صعودی طی می کرد و تحقیقات روی شبکه های عصبی فزونی یافت و ایده های بسیار جدیدی مطرح شدند . ایده های نو و تکنولوژی بالا برای رونسانس دوباره در شبکه های عصبی کافی بنظر می رسید . در این زایش دوباره شبکه های عصبی دو نگرش جدید قابل تامل می باشند . استفاده از مکانیسم تصادفی جهت توضیح عملکرد یک طبقه وسیع از شبکه های برگشتی که می توان آنها را جهت ذخیره سازی اطلاعات استفاده نمود . این ایده توسط هاپفیلد[۱] فیزیکدان آمریکایی در سال ۱۹۸۲ مطرح شد . هاپفیلد تکنیک هایی را از مکانیک آماری استفاده کرد تا خواص ذخیره سازی و بهینه سازی شبکه ها را آنالیز کند . دومین ایده مهم که کلید توسعه شبکه های عصبی در دهه هشتاد شد ، الگوریتم پس انتشار خطا[۲] می باشد که توسط دیوید راملهارت و جیمز مکلند در سال ۱۹۸۶ مطرح گردید . با بروز این دو ایده شبکه های عصبی متحول شدند . شبکه

---

۱. Kohonen
۲. Anderson
۳. Grossberg
۱. Hopfield
۲. Back propagation

های عصبی در هر دو جهت توسعه تئوریک و عملی در حال رشد می باشند . آنچه که باید اذعـان داشت این است که روند رشد شبکه های عصبی آهسته و مطمئن نبوده است . دوره هایی بوده که خیلی سریع رشد نموده و دوره هایی هم بوده است که رشد سریعی از خـود نـشان نـداده اسـت . بیشتر پیشرفت ها در شبکه های عصبی به ساختار نوین و روشهای یادگیری جدیـد مربـوط مـی شود . [۲]

دوره های بعد با این کشف که اضافه کردن لایه های پنهان دیگر به شبکه ، توانایی شبکه را در پاسخگویی به مسائل افزایش می دهد ، آغاز شد . این مورد ، باعث ایجاد علاقه مجدد قابل توجهی نسبت به شبکه های عصبی مصنوعی (مخصوصا"ساختارهای پیش خور چند لایه) ، که تا به امروز ادامه دارد ، گردید . سالهای اخیر شاهد دریایی از تغییرات در محتوا و روشهای تحقیق در شـبکه های عصبی مصنوعی بوده است اکنون متداول آنست که جهت توسعه شبکه های عصبی مصنوعی بیشتر بر نظریه های اثبات شده کنونی و تجربیات عملی تکیه شود . [۳]

درحال حاضر شبکه های عصبی مصنوعی به عنوان یک ابزار علمی درحل مسائل جایگاه مهمی دارند . انتظار می رود با کسب اطلاعـات بیـشتر از چگـونگی عملکـرد مغـز و نرونهـای بیولوژیـک پیشرفت های مهمی در شبکه های عصبی مصنوعی صورت پذیرد .

فصل دوم

محاسبات عصبی

۲-۱ محاسبات در شبکه های عصبی مصنوعی

شبکه های عصبی مصنوعی (ANN)[1] از تعدادی واحد (نرون های مصنوعی) که در داخل شبکه به هم وصل شده اند ، تشکیل شده اند . هر واحد دارای یک مشخصه ورودی – خروجی می باشد و محاسبات را اجرا می کند . خروجی هر واحد ، با توجه به مشخصه ورودی – خروجی آن ، اتصالات درونیش به سایر واحدها و احتمالا" ورودی های خارجی تعیین می گردد . عمل یا عملکرد کلی شبکه های عصبی مصنوعی ، توسط توپولوژی (شکل ارتباط نرون ها با هم و موقعیت لایه ها) شبکه ، خصوصیات نرون منفرد ، روش یادگیری و داده های آموزش معین میشود .

هر یک از واحدهای منفرد شبکه عصبی مصنوعی ایفا کننده عملکرد موضعی می باشند و شبکه کلی با اتصالات درونی واحد ها ، عملی مطابق آن شبکه را نمایش می دهد . تحلیل این عملیات ، بدون آموزش یا آزمایشهای نمونه ، اغلب دشواراست .

یک معیار کلیدی یادگیری، اصلاح الگوی ارتباط عناصر درونی شبکه بر اساس تابعی از داده های آموزشی است به عبارت دیگر دانش سیستم ، تجربه یا آموزش به شکل اتصالات داخلی شبکه ذخیره می گردد . شبکه های عصبی باید توانایی ذخیره اطلاعات را داشته باشند بعبارت دیگر ، آنها باید آموزش پذیر باشند . شبکه های عصبی به شکل مورد انتظار آموزش می یابند تا بعدا" درزمانی که الگوی جدیدی به منظور تشخیص یا طبقه بندی به آنها عرضه شد ، همواره رفتاری صحیح ارائه دهند . بنابراین هدف در مرحله آموزش شبکه ، گسترش یک ساختار درونی است که شبکه را قادر می سازد تا الگوهای جدید و مشابه را به روش صحیحی مشخص یا طبقه بندی کند . شبکه عصبی می تواند یک سیستم دینامیکی باشد و حالات آن مثلا" خروجی های

---

[1] . Artificial Neural Network

هر واحد و شدت اتصالات درونی در پاسخ به ورودی های خارجی یا یـک حالـت گـذرا ، بـا زمـان تغییرکند .

۲-۲ تعریف اصطلاحات مهم در شبکه های عصبی مصنوعی

فهرست کوتاهی از اصطلاحات مهم در شبکه های عصبی مصنوعی با ارائه تعریف مختصر بـه شرح زیر است .

شبکه : ادغامی از موجودیت هایی است که در داخل به هم متصل هستند .

الگوریتم : یک روش یا رویه به منظور رسیدن به یک هدف با راه حل است .

ساختار : تشکیلات سخت افزاری یا نرم افزاری است .

طبقه بندی : قابلیت نسبت دادن ورودی اعمالی به یک طبقه است .

شاخص : عناصری هستند که یک ویژگی از یک شی یا موقعیتی را مشخص می کنند .

تعمیم : توانایی شبکه در مواجهه و پاسخگویی به مثالهای بیشتر است .

آموزش : تمرین دادن (آمادگی) شبکه توسط زوج بردارهای ورودی و خروجی نمونه است .

توپولوژی : چگونگی ارتباطات بین لایه ها و نرون های داخلی شبکه است .

۲-۳ طراحی شبکه عصبی مصنوعی

یک رهیافت مهندسی برای حل مسائل ، ترکیب همه متغیر ها و اطلاعات مناسب مساله به گونه ای ساختار یافته به منظور فرموله کردن یک راه حل است . به طور نمونه ، فرایند کلاسیک مهندسی طراحی شامل ، بکارگیری اصولی از قواعد علمی و ریاضی به منظور طرح سیستمی که

با یک مجموعه مشخصات سر وکار دارد ، می باشد. از این جهت ممکن است طراحی شامل قضاوت ، بینش و احتمالا" تکرار باشد . فرایند آموزش به صورت نمونه شامل برخی روشهای تعلیم دادن است تا در موقعی که سیستم با مشخصاتی مواجه می گردد ، آنرا به انجام رفتار هایی وادار سازد . اغلب اوقات کاملا" این تعلیم دهی (فرایند یادگیری) شامل تصحیح یا سازگاری پارامتر های سیستم است برای اینکه در تکرار یا آزمایش بعدی پاسخ سیستم به آنچه مطلوب است ، نزدیک باشد .

مهندسی عصبی ، تعیین اجزای مربوط به راه حل شبکه عصبی مصنوعی ، شامل طرح کلی شبکه، توپولوژی های شبکه عصبی مصنوعی ، پارامتر های هر واحد و یک روند مرحله به مرحله آموزش (یادگیری) را جایگزین طرح های مهندسی کلاسیک می کند .

در کاربردهای واقعی ، طراحی شبکه عصبی مصنوعی ، کاری مشکل و معمولا" همراه با تکرار و اثرات متقابل است . گر چه فراهم کردن یک روش الگوریتمی جامع وفراگیر غیر ممکن است ، اما مراحل وابسته وساختار یافته که درزیرآمده است ، انعکاس نمونه تلاش ها و کار هایی است که در این زمینه انجام شده است .

پارامتر های مهم طراحی شبکه های عصبی مصنوعی را می توان به شرح زیر بیان کرد :

۱- ساختار اتصالات درونی – توپولوژی شبکه – ساختار کلی شبکه

۲- خصوصیات یک واحد که ممکن است در درون شبکه وبین قسمت های فرعی شبکه مانند لایه ها ، متفاوت باشد .

۳- مراحل آموزش شبکه

۴- مجموعه های تست وآموزش

۵- ورودی – خروجی و پیش و پس پردازش

۲-۴ آموزش شبکه های عصبی

بطورکلی آموزش شبکه های عصبی به دو طریق صورت می گیرد ، آموزش با معلم و آموزش بدون معلم . همچنین شبکه های عصبی وجود دارند که در آنها وزن رابط های بین نرونها دارای مقادیر مشخصی هستند و نیاز به آموزش شبکه به طریق تکراری نیست .

آموزش اکثر شبکه های عصبی با استفاده از مجموعه زوج بردارهای نمونه صورت می گیرد بطوریکه به هر بردار ورودی یک بردار خروجی مشخص نسبت داده می شود . با ارائه این مجموعه بردارها به شبکه ، وزن رابط های بین نرون ها بر اساس الگوریتم یادگیری شبکه اصلاح می گردند که البته این گونه آموزش را آموزش با معلم می نامند .

گاهی شبکه های عصبی طوری طرا حی می شوند که فقط بردارهای (الگوهای) ورودی را گروه بندی و مشخص نمایند که یک بردار ورودی به یک گروه داده شده تعلق دارد یا ندارد . در اینگونه شبکه ها اگر بردار ورودی به این گروه متعلق باشد خروجی عدد یک است و اگر بردار به این گروه تعلق نداشته باشد خروجی عدد منفی یک است . اینگونه شبکه ها ، با بکار گیری الگوریتم هایی که به صورت آموزش با معلم تربیت می شوند . در مسائلی که گروه بندی بردارهای ورودی مد نظر باشد ، برای مسائل ساده از شبکه های تک لایه و برای مسائل پیچیده از شبکه های چند لایه استفاده می گردد .

در بسیاری از موارد پاسخ شبکه بصورت بلی و یا خیر نیست بلکه یک بردار (الگو) می باشد . یعنی پس از اینکه شبکه برای مجموعه ای از زوج های آموزشی ورودی – خروجی آموزش داده

شد ، با ارائه یک بردار ورودی پاسخ شبکه نزدیکترین بردار به بردار ورودی خواهد بود . شبکه های چند لایه قادر به انجام نگاشت های غیر خطی از یک فضای n بعدی به یک فضای m بعدی هستند . شبکه عصبی انتشار برگشتی[1] در این گروه شبکه ها طبقه بندی می شوند .

در آموزش بدون معلم ، بردارهای ورودی به شبکه ارائه گردیده بدون این که بردارهای خروجی مربوطه به شبکه داده شوند . وزن های شبکه به صورتی اصلاح می شوند که بردارهای ورودی مشابه در یک گروه بندی قرار گیرند . شبکه عصبی انتشار متقابل دراین گروه از شبکه ها می تواند قرارگیرد .

۲-۵ توابع تحریک متداول مورد استفاده در شبکه های عصبی

عملیات اصلی شبکه عصبی مشتمل بر جمع ورودی های وزن دار شده واعمال تابع تحریک به این مجموع برای تعیین خروجی شبکه می باشد . برای نرون های ورودی این تابع یک تابع واحد[2] است ، بدین معنی که خروجی از نرون برابر ورودی آن می باشد .

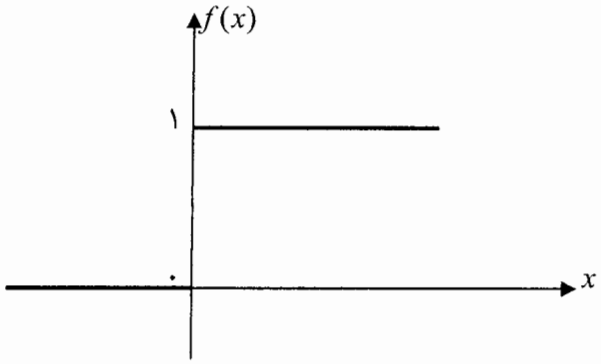
عموما" برای تمامی نرون های یک لایه تابع تحریک یکسان انتخاب می گردد ، اگر چه چنین شرطی لزوما" برقرار نیست . در اکثر موارد ، یک تابع تحریک غیر خطی بکار گرفته می شود . با در نظر گرفتن توانایی های محدود شبکه های تک لایه ای ، برای اینکه حداکثر استفاده از شبکه های چند لایه انجام گیرد نیاز به تابع تحریک غیر خطی می باشد زیرا عبور اطلاعات از دو یا چند لایه که در آنها نرون ها با تابع تحریک خطی عمل می نمایند به همان نتیجه ای منتهی می گردد که با یک شبکه یک لایه میسر است و همانگونه که می دانیم ترکیب توابع خطی

---

[1] . Back propagation
[2] . Identity function

همواره تابعی خطی خواهد بود .

تابع تحریک پله ای اغلب در شبکه های تک لایه استفاده می شود تا ورودی یک نـرون را بـه خروجی دو تایی مثبت (۰ یا ۱) و یا دوتایی مثبت و منفی (۱ یا ۱-) تبدیل نماید شکل (۲-۱).



شکل (۲-۱) : تابع تحریک پله ای

تابع تحریک سیگموئیدی (منحنی S شکل) :

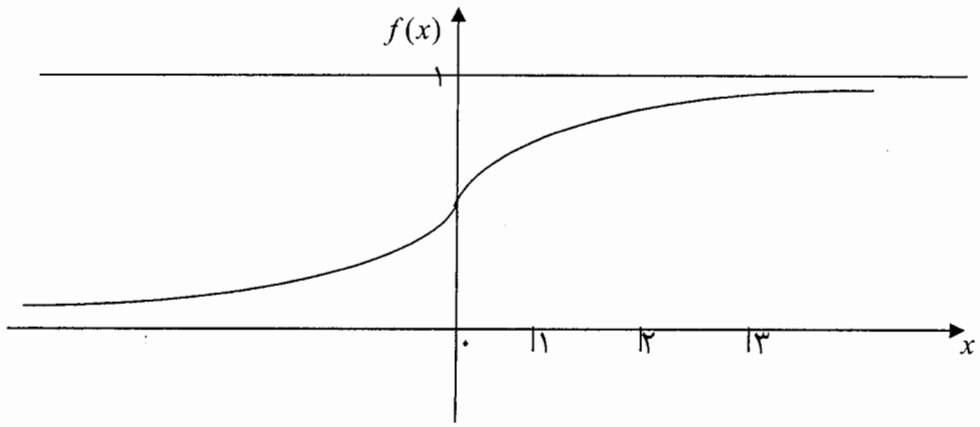
عمدتا" در شبکه عصبی انتشار برگشتی بکار گرفته می شود . علت اصلی استفاده از این تابع ارتباط ساده بین مقدار تابع در یک نقطه و مقدار مشتق تابع در آن نقطه است که درآموزش ایـن شبکه برای به حداقل رساندن خطا بکارگرفته می شود .تابع تحریک سـیگموئیدی دارای پاسـخی در محدوده صفر تا یک و همچنین می تواند پاسخی در محدوده یک و منفی یـک داشـته باشـد . این تابع در شکل (۲-۲) نشان داده شده است .

معادلات (۲-۱) و (۲-۲) تابع تحریک و مشتق آن را نمایش می دهد که دارای پاسخ در محـدوده صفر و یک هستند .

$$f(x) = 1/(1+\exp(-ax)) \qquad (۲-۱)$$

$$f'(x) = a.f(x).[1 - f(x)] \qquad (۲-۲)$$

بطوریکه با اعمال پارامتر a شیب منحنی تغییر می کند .



شکل (۲-۲) : تابع تحریک سیگموئیدی

معادلات (۲-۳) و (۲-۴) تابع تحریک و مشتق آن را نمایش می دهند که دارای پاسخ در محدوده یک و منفی یک هستند .

$$g(x) = (1 - \exp(-ax))/(1 + \exp(-ax)) \qquad (۲-۳)$$

$$g'(x) = (a/2).[1 + g(x)].[1 - g(x)] \qquad (۲-۴)$$

این تابع تحریک بطور قابل ملاحظه ای به تابع هیپربولیک نزدیک است . تابع هیپربولیک ورودی به نرون را به محدوده ای بین منفی یک ویک نگاشت می کند و اغلب بجای تابع تحریک (۲-۳) در شبکه های عصبی بکار گرفته می شود . این دو تابع در شرایط $a = 1$ مقایسه می گردند .

$$g(x) = (1 - \exp(-x))/(1 + \exp(-x)) \qquad (۲-۵)$$

وتابع هیپربولیک (رابطه (۲-۶))

$$h(x) = (1 - \exp(-2x))/(1 + \exp(-2x)) \qquad (۲-۶)$$

مشتق این تابع برابر است با :

$$h'(x) = [1 + h(x)].[1 - h(x)] \qquad (۲-۷)$$

۲-۶ بایاس[۱] و آستانه تحریک

بایاس هر نرون بمانند وزنی با مقدار مشخص که از نرون مفروض با پاسخ ثابت یک بر آن تحمیل شود ، عمل می کند . افزایش بایاس موجب افزایش ورودی نرون می گردد . بعنوان مثال در یک تابع تحریک پله ای چنانچه بایاس در نظر گرفته شده باشد پاسخ تابع تحریک به صورت زیر می باشد :

$$f(net) = \begin{cases} 1 & if \quad net \geq 0 \\ \\ -1 & if \quad net < 0 \end{cases} \qquad (۲-۸)$$

بطوریکه

$$net = b + \sum_i x_i.w_i \qquad (۲-۹)$$

(b مقدار بایاس می باشد) .

البته می توان بایاس را در نظر نگرفت اما در مقابل باید یک مقدار ثابت آستانه ای مثل θ همیشه در نرون در نظر گرفته شود . در آن صورت پاسخ تابع تحریک بصورت زیر خواهد بود :

$$f(net) = \begin{cases} 1 & if \quad net \geq \theta \\ \\ -1 & if \quad net < \theta \end{cases} \qquad (۲-۱۰)$$

و برای این حالت داریم :

---

۱. Bias

$$net = \sum_i x_i . w_i \quad . \qquad (۲-۱۱)$$

فصل سوم

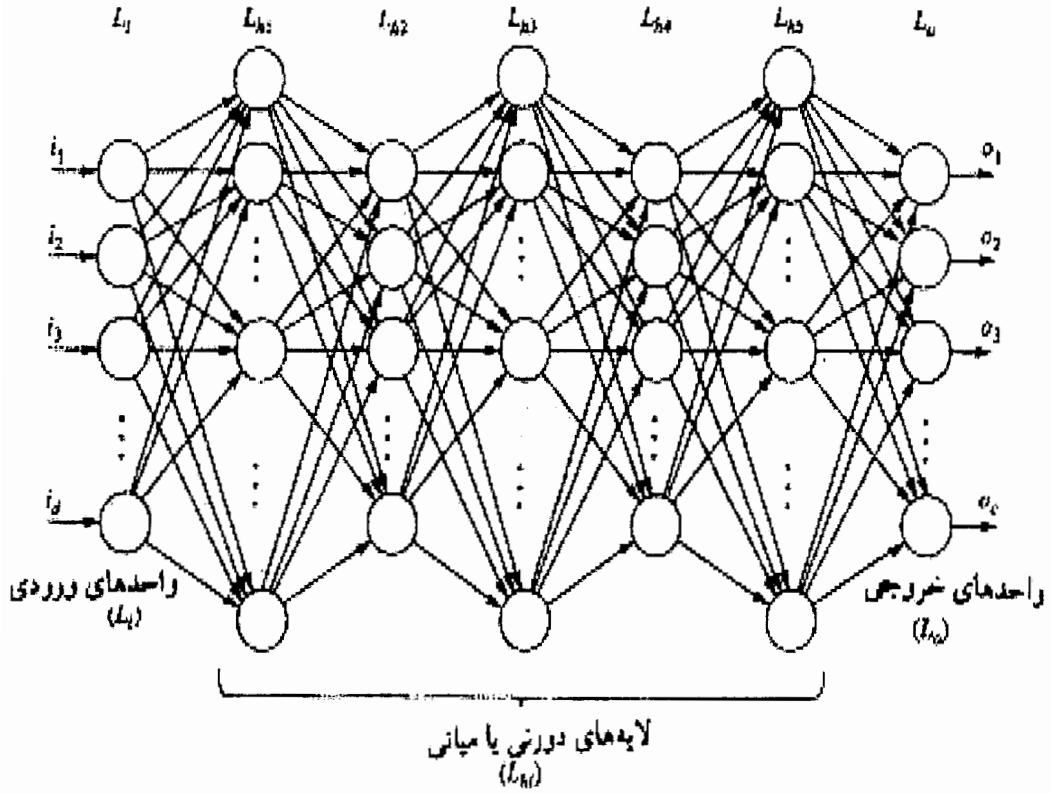
شبکه های عصبی انتشار برگشتی (پیش خور)

و

انتشار متقابل

۳-۱ ساختار شبکه عصبی انتشار برگشتی

شبکه های انتشار برگشتی ، از سلسله مراتب واحد های پردازشگر ساخته شده است که در یک مجموعه متشکل از دو یا چند مجموعه منحصر بفرد از نرون ها یا لایه ها ، ساز مان دهی شده اند . لایه اول یا لایه ورودی ، بعنوان محل نگهداری ورودی های اعمال شده به شبکه بکار می رود . لایه آخر یا لایه خروجی ، مکانیست که در آن نگاشت کلی ورودی شبکه قابل دستیابی است . بین این دو لایه ، صفر تا چند لایه درونی (مخفی) می تواند قرار داشته باشد و داخل این لایه های درونی است که نگاشت یا محاسبات عصبی انجام می گیرد . پیوند ها یا وزن ها ، هر واحد در یک لایه را تنها به آنهایی که در لایه بعدی قرار دارند ، وصل می کنند . در ضمن این اتصالات جهت دارند که در آنها خروجی یک واحد ، که با مقدار وزن اتصال مقیاس دهی می شود ، پیش خورمی شود تا نقشی را در تحریک واحدهای لایه بعدی ایفا کند . شکل (۳-۱) یک شبکه انتشار برگشتی نمونه را نشان می دهد . همانطور که نشان داده شده ، شبکه شامل یک لایه از d واحد ورودی ، c واحد خروجی و تعداد متغیری از لایه های درونی یا مخفی از واحد ها می باشد . نقش لایه ورودی به نوعی ساختگی می باشد که در آن واحدها تنها برای نگهداری مقادیر ورودی و توزیع این مقادیر بین واحدهای لایه بعدی بکار می روند . بعبارت دیگر واحدهای لایه ورودی ، نگاشت یا تبدیل جداگانه ای بر اطلاعات ورودی انجام نمی دهند .



شکل (۳-۱) : ساختار شبکه عصبی انتشار برگشتی

در شکل (۳- ۱) اطلاعاتی که درون شبکه جاری می شوند ، محدود به جریان از لایه ای بـه لایـه دیگر ، از ورودی به سمت خروجی می باشند . هر لایه ، بر مبنای ورودی اش یک بردار خروجی را محاسبه می کند و این اطلاعات را به لایه بعدی منتشر می کنـد . بنـابر ایـن از دیـد سـاختاری ، شبکه انتشاربرگشتی پردازش موازی را درون هرلایه مجاز میداند ولی جریان اطلاعات بین لایه ها لزوما" ترتیبی می باشد .

۳-۲ الگوریتم آموزش شبکه

هنگامی که یک ساختار شبکه ای مناسب انتخاب می شود، بیشترین تلاش در زمینه طراحی یک شبکه عصبی مصنوعی برای یک کاربردخاص، شامل طراحی یک شیوه آموزشـی معقـول مـی

باشد . در حال حاضر متداولترین الگوریتم آموزش شبکه ها قانون دلتای تعمیم یافته مبتنی بر پس انتشار می باشد . این قانون مبتنی بر گرادیان نزولی[1] است و تابع تحریک آنها غیر خطی می باشد .

قانون دلتای تعمیم یافته یک قانون یادگیری ضربی برای یک شبکه عصبی انتشار برگشتی چند لایه می باشد که از گرادیان نزولی ، برای دست یافتن به آموزش یا یادگیری با تصحیح خطا استفاده می کند . وزن های شبکه به گونه ای تنظیم می شوند که خطایی را که بر مبنای مقدار اختلاف بین خروجی مطلوب و خروجی واقعی شبکه حاصل شده است ، به حداقل رسانند . فرایند آموزش شبکه می تواند مشتمل بر مراحل زیر باشد :

۱- مقدار دهی اولیه به تمام وزن های واحد های شبکه

۲- اعمال یک بردار ورودی (محرک) به شبکه

۳- انتشار بردار ورودی برای تعیین تمام خروجی های واحد ها

۴- مقایسه پاسخ های واحد ها در لایه خروجی با پاسخ مطلوب یا هدف

۵- محاسبه وانتشار مقدار حساسیت خطا به سمت عقب از طریق شبکه و بعنوان مبنای تصحیح وزن شبکه

۶- به حداقل رساندن خطای کلی در هر مرحله با تنظیم وزن واحد ها

برای این منظور علامت های زیر مورد استفاده قرار می گیرد :

X : بردار ورودی $\qquad X = (x_1, x_2, ..., x_n)$

---

[1] – گرادیان نزولی ، روشی تکراری جهت پیدا نمودن نقطه مینیمم در توابع می باشد که رابطه عمومی آن برای این منظور ، به ترتیبی است که می آید $x(k+1) = x(k) - a.f'(x(k))$ ، a عددی مثبت کوچک بوده و با حدس زدن $x(0)$ و محاسبه مشتق تابع در این نقطه ، سیکل تکراری رابطه شروع می شود تا موقعی که همگرایی حاصل گردد .

T : بردار خروجی مطلوب $T = (t_1, t_2, ..., t_m)$

W_{ij} : وزن های شبکه (وزن بین نرون های لایه i و j)

Y : بردار پاسخ واقعی متناظر با بردار ورودی $Y = (y_1, y_2, ..., y_m)$

برای تنظیم وزن های شبکه ΔW_{ij} ، یک خطای E و گرادیان $\partial E/\partial W_{ij}$ را تعریف و محاسبه می کنیم . برای تنظیم وزن ، ΔW_{ij} متناسب با $-\partial E/\partial W_{ij}$ قرار داده می شود . برای آموزش شبکه عصبی انتشار برگشتی (پیش خور) و تنظیم وزن ها از یک مجموعه زوج های مرتب بردار های ورودی و خروجی متناظر با آن استفاده می شود .

۳-۳ شیوه تنظیم وزن ها

در ابتدا روشی برای تنظیم وزن ها بر اساس اختلاف بین خروجی مطلوب و پاسخ واقعی متناظر با بردار ورودی از قبل تعیین شده ، به شکل حاصلضربی زیر نیاز داریم :

$$\Delta W_{ij} = a.(t_j - y_j).x_i \qquad (۳-۱)$$

که در آن a یک پارامتر تنظیم یا نرخ یادگیری است و خطای jامین واحد خروجی به صورت زیر می باشد :

$$e_j = t_j - y_j \qquad (۳-۲)$$

برای ورودی غیر صفر ومثبت ، اگر خروجی مطلوب واحد jام لازم کمتر از مقدار واقعی بدست آمده یعنی y_j باشد ، اتصال داخلی بین ورودی i و واحد خروجی j با افزایش W_{ij} تقویت می شود . هر چند باید توجه کنید که اثر این تغییر را روی پاسخ شبکه به دیگر زوج الگو ها در فرایند آموزش در نظر نگرفته ایم. برای فرمول بندی نتیجه حاصل ازمطالب فوق یک بردار

خطای خروجی را برای یک زوج خاص به صورت زیر تعریف می کنیم :

$$e_k = t_k - y_k \qquad (۳-۳)$$

البته باید توجه داشت که t_k و y_k مربوط به بردار ورودی kام می باشند .

حال یک مقدار اسکالر از این بردارخروجی بر اساس نمونه kام آموزش ، با E_k نشان داده شده و به صورت زیر تعریف می شود :

$$E_k = 0.5\|e_k\|^2 = 0.5(e_k)^T.(e_k) \qquad (۳-۴)$$

برای تشریح اصولی الگوریتم های آموزشی قانون دلتای تعمیم یافته با استفاده از قانون زنجیری ابتدا رابطه (۳-۵) در نظر گرفته می شود .

$$\partial E_k/\partial W_{ij} = \partial E_k/\partial y_k \times \partial y_k/\partial net_k \times \partial net_k/\partial W_{ij} \qquad (۳-۵)$$

در حالی که از معادلات قبلی داریم :

$$E_k = 0.5\|t_k - y_k\|^2 \quad \text{یا} \qquad (۳-۶)$$

$$E = \sum_k E_k = 0.5\sum_k \|t_k - y_k\|^2 \qquad (۳-۷)$$

یک تابع تحریک برای واحد خروجی jام به شکل کلی زیر در نظر می گیریم :

$$y_j = f(W_{ij}, X) \qquad (۳-۸)$$

که در آن تابع f یک تابع غیر نزولی و مشتق پذیر نسبت به متغیرهای مستقلش است . توابع خروجی اکیدا" خطی بعنوان مثال $f(x) = k.x$ قابل قبولند ولی باعث بوجود آمدن نقایص دیگری می شوند . بعنوان مثال ، تابع f حداقل درون یک لایه ثابت است . و این بدون تغییر بودن ممکن است در توانمندی شبکه اخلال ایجاد کند .

محرک نرون مصنوعی برای واحد j از جمع خطی وزن دار ورودی های واحد j بدست می آید

یعنی :

$$net_j = \sum_i W_{ji}.x_i \qquad (۹-۳)$$

یا در حالتی که یک ورودی با بایاس اضافه شود داریم :

$$net_j = \sum_i W_{ji}.x_i + bias_j \qquad (۱۰-۳)$$

معادله فوق نشان دهنده یک مدل از یک واحد عصبی است که در آن بایاس مستقیما" به net_j اضافه شده است که در این موارد آموزش بایاس نیز باید انجام شود .

در ادامه بحث با در نظر توابع شبه خطی (تابعی غیر نزولی و مشتق پذیر) در شبکه عصبی ، روابط مربوط به اصلاح وزن های شبکه و تقلیل خطا را بازنگری می کنیم و به منظور تعمیم نتایج خود برای لایه مخفی ، یک شبکه عصبی را در نظر می گیریم که شامل لایه مخفی ، لایه خروجی و ورودی باشد .

از رابطه (۵-۳) داریم :

$$\partial E/\partial W_{ji} = \partial E/\partial net_j \times \partial net_j/\partial W_{ji} \qquad (۱۱-۳)$$

که E معیار خطا تعریف شده بود و برای net_j داریم :

$$net_j = \sum_i W_{ji}.O_i \qquad (۱۲-۳)$$

بطوریکه برای واحدها در لایه میانی و خروجی $O_i = y_i$ و برای واحدها در لایه ورودی $O_i = x_i$است . با تعریف حساسیت خطای الگو روی محرک خالص واحد jام به صورت زیر داریم :

$$\delta j = -\partial E/\partial net_j \qquad (۱۳-۳)$$

که با بکار گیری رابطه (۳-۵) رابطه حساسیت خطا نسبت به وزن به صورت زیر بدست می آید :

$$\partial E/\partial W_{ji} = \partial E/\partial net_j \times \partial net_j/\partial W_{ji} = -\delta j \times \partial net_j/\partial W_{ji} \qquad (۱۴-۳)$$

و از قبل داریم :

$$net_j = \sum_i W_{ji}.O_i \qquad (۱۵-۳)$$

پس رابطه حساسیت خطا نسبت به وزن در پایان (البته در حالت کلی) به صورت زیر خواهد بود :

$$\partial E/\partial W_{ji} = -\delta j \times \partial net_j/\partial W_{ji} = -\delta j \times O_i \qquad (۱۶-۳)$$

که یک رویه مشخص تصحیح وزن تکراری با استفاده از یک نمونه آموزشی دلخواه است و در عمل بیشتر به صورت رابطه (۳-۱۷) تصحیح می شود .

$$\Delta W_{ji} = -a.(\partial E/\partial W_{ji}) \qquad (۱۷-۳)$$

که در رابطه فوق a یک مقدار ثابت است که به نرخ یادگیری معروف می باشد .

نرخ یادگیری معین می کند که چه مقدار حساسیت خطای حساب شده نسبت به تغییر وزن، برای تصحیح وزن استفاده خواهد شد . بهترین نرخ یادگیری وابسته به مشخصات سطح خطای E بر حسب W_{ji} ، می باشد . اگر سطح خطا به سرعت تغییر کند ، گرادیانی راکه تنها بر اساس اطلاعات محلی محاسبه شده است ، نشانه ضعیفی از مسیر صحیح حقیقی را ارائه می دهد . در این حالت ، یک نرخ یادگیری کوچک تر مطلوب است . از طرفی اگر سطح خطا بکندی تغییرکند ،یک نرخ یادگیری بزرگتر ، رسیدن به همگرایی را سرعت می بخشد . هر چند این رابطه بر اساس آگاهی از شکل خطاست که بندرت دردسترس می باشد . یک قانون کلی نرخ یادگیری ،استفاده از بزرگترین نرخ یادگیری است که مؤثر باشد و در ضمن ایجاد نوسان نکند . بنابراین برای حرکت در جهت مخالف گرادیان ، تصحیح وزن اینچنین خواهد بود :

$$\Delta W_{ji} = a.\delta j.O_i \qquad (۱۸-۳)$$

$$\partial E/\partial net_j = \partial E/\partial y_j \times \partial y_j/\partial net_j \qquad (۱۹-۳)$$

و این رابطه همچنان یک قانون تصحیح ضرب است . برای مثال اگر یک مشخصه سیگموییدی انتخاب شده باشدکه در آن

$$y_j = F(net_j) = 1/(1+\exp(-net_j)) \qquad (۲۰-۳)$$

است مشتق خواسته شده در معادلات تصحیح وزن ، بصورت رابطه (۳-۲۱) است .

$$\partial F/\partial net_j = F'(net_j) = y_j.(1-y_j) \qquad (۲۱-۳)$$

بنابراین در مورد واحدهای خروجی ، با استفاده از معادلات (۳-۷) ، (۳-۱۳) و (۳-۱۹) داریم :

$$\delta j = (t_j - y_j).F'(net_j) \qquad \text{(برای یک لایه خاص)} \qquad (۲۲-۳)$$

و تصحیح وزن براساس نمونه برای چند واحد خروجی از معادله (۳-۱۸) و روابط تشریح شده در بالا به صورت زیردرمی آید:

$$\Delta W_{ji} = a.(t_j - y_j).F'(net_j).Oi \qquad (۲۳-۳)$$

روش تصحیح وزن از معادله (۳-۲۳) یک شیوه حاصلضربی است .

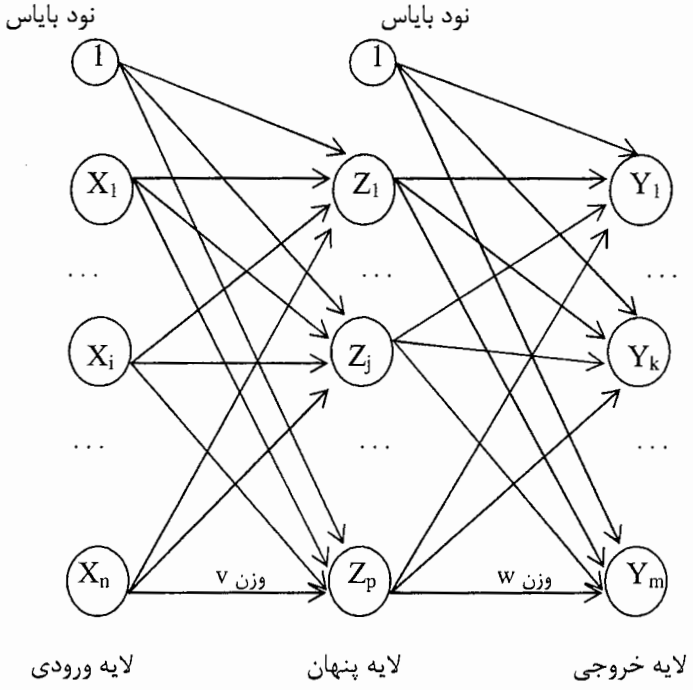
مقدار دهی اولیه مناسب وزن ها و مستولی شدن بر حداقل های نسبی ، بهره ور ساختن هر چه بیشتر آموزش بر اساس قانون دلتای تعمیم یافته را در پی خواهد داشت . [۱]

۳-۴ مراحل تنظیم وزن ها و بایاس ها

یک شبکه چند لایه با یک لایه پنهان (لایه Z) در شکل (۳-۲) نشان داده شده است . لایه

خروجی (لایه Y) و لایه پنهان ممکن است بایاس داشته باشد . بایاس نرون هـای لایـه خروجـی (Yk) با Wok و بایاس نرون های لایه Zj با Voj نمایش داده شده است .

بررسی های انجام شده [۱] نشان داده است که اعمال نمودن بایاس بـه تـابع تحریـک بـسیار مفید است . همچنین می توان اعمال نمودن بایاس را با در نظر گرفتن یک نرون اضافه کـه نـرون بایاس نامیده می شود ، انجام داد . هیچ گونه ورودی بـه ایـن نـرون وجـود نـدارد و خروجـی آن همیشه برابریک است این نرون به تمام نرونهای لایه بعدی مرتبط گردیده و همزمان با سایر نرون ها آموزش می بیند . در شکل (۳-۲) فقط جریان داده ها از ورودی به خروجـی نـشان داده شـده است . با برآورد خطا و با انتشار معکوس از خروجی بـه طـرف ورودی ، عکـس عمـل مرحلـه اول صورت می گیرد .



شکل (۳-۲) : شبکه عصبی انتشار برگشتی همراه با لایه پنهان

شبکه نشان داده شده در شکل (۳-۲) یک لایه پنهان دارد ولی مطالب مطرح شده برای شبکه های عصبی با چند لایه پنهان نیز صادق است . بطور کلی آموزش اینگونه شبکه ها درسه مرحله دادن اطلاعات به ورودی ، برآورد میزان خطا باتوجه به خروجی درخواست شده و برگشت از خروجی به ورودی و اصلاح وزن ها ، صورت می گیرد .

در مرحله حرکت از ورودی به خروجی ، هر نرون ورودی Xi علائمی را دریافت نموده و این علایم را به نرون های لایه پنهان Zj انتقال می دهد . ورودی به هر یک از نرون های لایه پنهان براساس رابطه (۳-۲۴) محاسبه می شود و سپس پاسخ لایه پنهان به لایه خروجی انتقال می یابد .

$$net = V_1.x_1 + V_2.x_2 + ... + V_n.x_n \qquad (۲۴-۳)$$

که در آن x و V به ترتیب مقدار ورودی و وزن لایه ورودی هستند . هر نرون خروجی Yk دارای یک خروجی y_k برای ورودی داده شده خواهد بود .

به هنگام آموزش شبکه ، برای هر نرون خروجی ، مقدار تحریک (y_k) محاسبه شده آن نرون را با مقدار هدف یعنی t_k ، مقایسه نموده و میزان خطای مربوط به آن را محاسبه می نماییم . سپس بر اساس این خطا ضریب δ_k ($k = 1,...,m$) محاسبه می گردد . δ_k خطای ایجاد شده درنرون خروجی Yk را بین تمام نرونهای لایه قبل (نرونهای لایه پنهان که به نرون Yk مرتبط هستند) توزیع می شود . این ضریب همچنین در اصلاح وزن های موجود بین لایه پنهان و لایه خروجی بکار گرفته می شود .

بااستفاده ازروش مشابه ، ضریب δ_j ($j = 1,...,p$) برای هر نرون Zj از لایه پنهان محاسبه می شود . هر چند نیازی به برگشت وارونه خطا به نرون های ورودی نمی باشد ، δ_j در اصلاح

وزن های بین لایه پنهان و لایه ورودی بکار گرفته خواهد شد . بعد از اینکه تمام ضرایب δ تعیین گردیدند ، وزن های تمام لایه ها به طور همزمان اصلاح می گردند . اصلاح وزن Wjk (از نرون Zj به نرون Yk لایه خروجی) بر اساس ضریب δ_k و پاسخ تحریک z_j از نرون Zj لایه پنهان صورت می گیرد . همینطور اصلاح وزن Vij (از نرون Xi لایه ورودی به نرون Zj لایه پنهان) بر اساس ضریب δ_j و پاسخ تحریک x_i نرون ورودی انجام می شود .

برای یک شبکه عصبی انتشار برگشتی عموما" متغیر هایی بشرح زیر مد نظر قرار می گیرد :

X : بردار ورودی $\qquad X = (x_1,...,x_i,...,x_n)$

T : بردار هدف (مطلوب) $\qquad T = (t_1,...,t_k,...,t_m)$

δ_k: بخشی از اصلاح خطا که موجب تغییر وزن Wjk می گردد . این خطا مربوط به نرون خروجی Yk است بعلاوه اطلاعات مربوط به خطای Yk با برگشت معکوس به نرون های لایه میانی (پنهان) انتقال می یابد .

δ_j: بخشی ازاصلاح خطا که موجب تغییر وزن Vij می گردد . این خطایی است که دربرگشت معکوس از لایه خروجی به لایه پنهان Zj انتقال یافته است .

Xi : نرون i ام لایه ورودی می باشد .

Voj : بایاس نرون j ام لایه پنهان می باشد .

$Zinj$: ورودی به نرون Zj می باشد که توسط رابطه (۳-۲۵) محاسبه می شود :

$$Zinj = Voj + \sum_i Xi.Vij \qquad (۲۵-۳)$$

خروجی (پاسخ تحریک) نرون Zj را با z_j نمایش می دهند وبرابر است با :

$$z_j = f(Zinj) \qquad (۲۶-۳)$$

Wok : بایاس نرون k ام لایه خروجی می باشد .

$Yink$: ورودی به نرون خروجی Yk می باشد که توسط رابطه (۳-۲۷) محاسبه می شود :

$$Yink = Wok + \sum_j z_j . Wjk \qquad (۲۷-۳)$$

خروجی (پاسخ تحریک) نرون Yk را با y_k نمایش داده می شود و برابر است با :

$$y_k = f(Yink) \qquad (۲۸-۳)$$

همانطوریکه مطرح شد در این شبکه برای لایه میانی (پنهان) و لایه خروجی احتیاج به یک تابع تحریک داریم (که در بحث فوق با f نمایش داده شد) ممکن است تابع تحریک در هر دو لایه متفاوت یا مشابه باشد معمولا" برای هر دو لایه در این نمونه ، تابع سیگموئیدی بکارگرفته میشود یعنی :

$$f(x) = 1/(1 + \exp(-ax)) \qquad (۲۹-۳)$$

$$f'(x) = a.f(x).[1 - f(x)] \qquad (۳۰-۳)$$

مراحل آموزش در شبکه عصبی انتشار برگشتی پس از انتخاب مقادیر تصادفی کوچک برای وزنها و بایاس ها برای هر زوج آموزشی را می توان به شرح زیر خلاصه نمود :

گام اول : هر نرون ورودی ($Xi, i = 1,...,n$) ورودیهای x_i را دریافت نموده و به لایه بعدی (لایه پنهان) انتقال می دهد .

گام دوم : در هر نرون لایه پنهان ($Zj, j = 1,...,p$) علائم ورودی وزن دار شده باهم جمع می شوند :

$$Zinj = Voj + \sum_{i=1}^{n} x_i . Vij \qquad (۳۱-۳)$$

که در آن Voj بایاس نرون j ام لایه پنهان و Vij وزن مربوط به نرون Xi لایه ورودی به نـرون Zj لایه پنهان است .

با بکارگیری تابع تحریک علائم خروجی نرون تعیین می گردد:

$$z_j = f(Zinj) \qquad (۳۲-۳)$$

سپس این مقادیر (z_j) به نرون های لایه بعدی (خروجی) انتقال می یابد .

گام سوم : در هر نرون خروجی ($Yk, k = 1,...,m$) ورودیهای وزن دار شده بـاهم جمـع میشوند :

$$Yink = Wok + \sum_{j=1}^{p} z_j . W_{jk} \qquad (۳۳-۳)$$

که در آن Wok بایاس نرون k ام لایه خروجی و Wjk وزن مربوط به نرون Zj لایه پنهـان بـه نرون Yk لایه خروجی است .

با بکارگیری تابع تحریک علائم خروجی نرون تعیین می گردد یعنی :

$$y_k = f(Yink) \qquad (۳۴-۳)$$

مرحله برگشت خطا در شبکه شامل گامهای زیر است .

گام چهارم : هر یک از نرون های لایه خروجی ($Yk, k = 1,...,m$) مقادیری را با توجـه بـه ورودیهای داده شده دریافت می کنند . در اینجا مقدار خطا به صورت زیر قابل محاسبه می باشد :

$$\delta_k = (t_k - y_k).f'(Yink) \qquad (۳۵-۳)$$

مقدار δ_k در اصلاح وزن Wjk بکارگرفته خواهد شد .

$$\Delta Wjk = a.\delta_k . z_j \qquad (۳۶-۳)$$

محاسبه مقدار اصلاح وزن بایاس که بعدا" دراصلاح Wok بکار گرفته می شود :

$$\Delta Wok = a.\delta_k \qquad (۳۷-۳)$$

δ_k به نرون لایه پایین تر فرستاده می شود (لایه پنهان).

گام پنجم : هر نرون لایه پنهان ($Zj, j = 1,..., p$) ورودی های δ ارسالی از لایه بالاتر باهم جمع می گردند .

$$\delta_{inj} = \sum_{k=1}^{m} Wjk.\delta_k \qquad (۳۸-۳)$$

با ضریب در مشتق تابع تحریک عبارت خطا محاسبه می گردد .

$$\delta_j = \delta_{inj}.f'(Zinj) \qquad (۳۹-۳)$$

با استفاده از δ_j ، اصلاح وزن Vij بصورت زیر محاسبه می گردد :

$$\Delta Vij = a.\delta_j.x_i \qquad (۴۰-۳)$$

و محاسبه اصلاح وزن بایاس که برای اصلاح Voj بکار می رود :

$$\Delta Voj = a.\delta_j \qquad (۴۱-۳)$$

گام ششم : (اصلاح وزن ها وبایاس ها) برای هر نرون خروجی ($Yk, k = 1,..., m$) وزنها وبایاس ها اصلاح می گردند .

$$Wjk(new) = Wjk(old) + \Delta Wjk \qquad (\ j = 1,...p\ ) \qquad (۴۲-۳)$$

$$Wok(new) = Wok(old) + \Delta Wok \qquad (\ k = 1,...,m\ ) \qquad (۴۳-۳)$$

که (new) و (old) به ترتیب هم ارز جدید و قدیم است .

برای هر نرون لایه پنهان ($Zj, j = 1,..., p$) وزن ها وبایاس ها اصلاح می گردند .

$$Vij(new) = Vij(old) + \Delta Vij \qquad (\ i = 1,...,n\ ) \qquad (۴۴-۳)$$

$$Voj(new) = Voj(old) + \Delta Voj \qquad (\ j = 1,..., p\ ) \qquad (۴۵-۳)$$

گام هفتم : در صورت همگرایی وزن ها و بایاس های شبکه ، آموزش پایان می یابد وگرنه تا حصول همگرایی ، مجددا" گامها تکرار می شود .

۳-۵ شبکه عصبی انتشار متقابل (CPN)[۱]

با ترکیب لایه رقابتی و لایه واسطه ای (انتقالی یا خروجی) درکنار لایه ورودی ، شبکه جدیدی بنام شبکه انتشار متقابل حاصل می گردد و هر یک از لایه ها وظیفه خاصی بعهده دارند بطوریکه وظیفه لایه رقابتی طبقه بندی بردارهای مشابه و لایه واسطه ای تدارک پاسخ مناسب هم است . این شبکه همانند یک جدول مرجع عمل می نماید (شکل (۳-۳)) . بدین معنی که ارتباط های بین لایه ها ، وزن هایی را بخود می گیرند که مشخص کننده مقدار خروجی شبکه در ازاء متغیرهای ورودی به شبکه می باشد . آموزش شبکه با استفاده از زوج های آموزشی صورت می گیرد . ساختار این نوع شبکه به صورتی است که حتی اگر بردارورودی فاقد بعضی از متغیر ها باشد شبکه توان پاسخ گویی مناسب به ورودی ها را دارا خواهد بود .

V_1	W_1
...	...
V_i	W_i
...	...
V_n	W_n

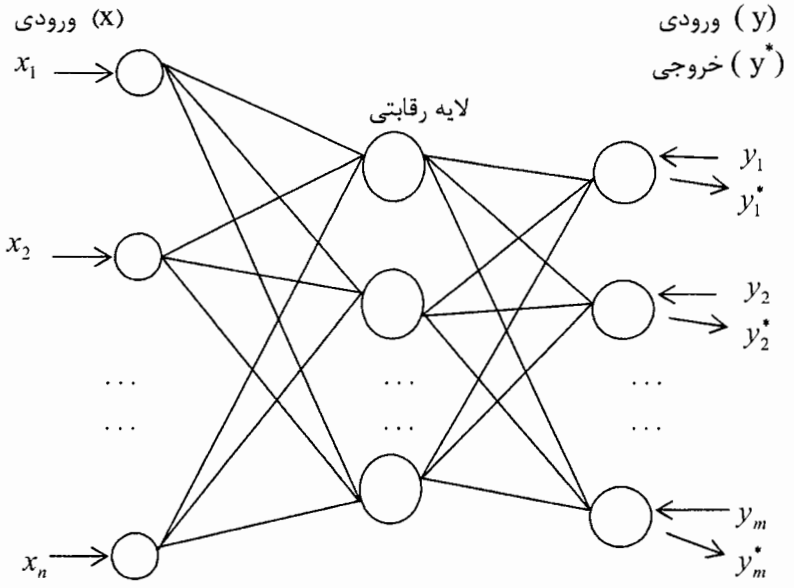
بردار ورودی X

بردار خروجی W_i

شکل (۳-۳) : نمایی از جدول مرجع برای نمایش عملکرد شبکه انتشار متقابل

۱ . Counter Propagation Neural Network

ظاهر شبکه عصبی انتشار متقابل مشابه شبکه انتشار برگشتی است و تفاوت ایندو در عملکرد پردازشگرهای لایه های آن می باشد . همانطوریکه در شکل (۳-۴) ملاحظه می شود ، نرون ها یا پردازشگرهای لایه ورودی از طریق رابط های وزن دار Vij به نرون های لایه رقابتی متصل میگردند . به همین ترتیب نرون های لایه رقابتی توسط رابط های وزن دار Wjk به نرون های لایه خروجی مربوط می شوند . زیر نویس i نشان دهنده شماره نرون های ورودی ، j شماره نرون های لایه رقابتی و k شماره نرون های لایه خروجی می باشد . همانند سایر شبکه ها این نوع شبکه در مرحله اول برای منظور خاصی تربیت می شود سپس می توان برای یافتن پاسخ مساله مکررا" از آن استفاده کرد .



شکل (۳-۴) : تصویر کلی از ساختار شبکه عصبی انتشار متقابل

۳-۶ آموزش شبکه عصبی انتشار متقابل

آموزش این شبکه در دو مرحله صورت می گیرد . در مرحله اول لایه رقابتی آموزش داده می شود و این لایه بردار های ورودی را که مشابه هم می باشنددر گروه خاصی از نرون ها طبقه بندی می کند . این عمل با اصلاح وزن های لایه رقابتی صورت مـی گیـرد بطوریکـه بردارهـای ورودی مشابه هم موجب فعال شدن آن نرون خاص می گردند سپس در مرحله بعد وزن های لایه واسطه ای طوری تنظیم می شوند که در برابر بردار های ورودی پاسخی مناسب ارائه کنند .

لایه رقابتی گروه بندی بردارهای ورودی را بر اساس یک ا لگوریتم خود سازمان ده و بـدون هیچ گونه نظارتی انجام می دهد و به همین دلیل پیش بینی اینکه کدام یک از نـرون هـای لایـه رقابتی برای یک بردار ورودی فعال می گردند بسیار مشکل و البته غیر ضروری می باشـد . فقـط لازم است که در مرحله آموزش ، گروه بندی بردارهای ورودی غیر مشابه تضمین گردد .

یکی از مسائلی که در مورد بردارهای ورودی به شبکه اهمیت بسیار دارد این است که انـدازه بردارها به گونه ای باشد که پس از ضرب در وزن های لایه اول مجموع قابل قبـولی ایجـاد کنـد . یک روش معمول یکه کردن این بردار هاست . برای این منظور باید مؤلفه بردار ورودی را به طول بردار تقسیم کرد . بدین ترتیب می توان اطمینان حاصل نمود که جهت بردار هـا بـیش از انـدازه آنها در تعیین پاسخ شبکه مؤثر می باشد . همانطوریکه می دانیم طول متفاوت بردارهای ورودی ، آنها را صرفا" در یک ضریب ثابت متفاوت می سازد و یک طبقه بندی جدید ایجاد نمی کند زیـرا عمل وزنهای شبکه و جمع خالص وزن ها یک عملگر خطی است .

برای آموزش لایه اول شبکه عصبی انتشار متقابل یک بردار ورودی به شـبکه اعمـال شـده و سپس حاصلضرب عددی آن با بردار وزن مربوط به هر نـرون لایـه رقـابتی محاسـبه مـی گـردد .

پردازشگری که بزرگترین مقدار ورودی را داشته باشد برنده بوده و وزن آن اصلاح می گـردد لازم بذکر است که بزرگترین مقدارعددی مربوط به حاصل ضرب داخلی بردار ها وقتی حاصل می شود که تفاضل دو بردار کم بوده یا دو بردار تقریبا" مشابه هم باشند . در واقع حاصـل ضـرب عـددی برای محاسبه ورودی به نرون ، معیاری از شباهت بـردار ورودی بـه بـردار وزن اسـت. البتـه روش ترتیب شامل انتخاب یکی ازنرون های لایه اول می باشد بطوریکه بردار وزن آن شبیه ترین بـردار به بردار ورودی ما باشد سپس سعی می کنیم تا این شباهت بـا اصـلاح وزن تقویـت گـردد . لازم بذکر است که این نوع آموزش بدون نظارت صورت می گیرد ، بدین معنـی کـه شـبکه خـودگروه بندی بردار های ورودی را بر اساس پردازشگر با حداکثر ورودی ها انجام می دهد . معادله اصـلاح وزن این لایه به صورت رابطه (۳-۴۶) است :

$$Vj(new) = Vj(old) + a.(Xi - Vj(old)) \quad (۴۶-۳)$$

[1]

$Vj(new)$: مقادیر جدید وزن های مربوط به رابط های ورودی X و نرون برنده

$Vj(old)$: مقادیر قبلی این وزن ها

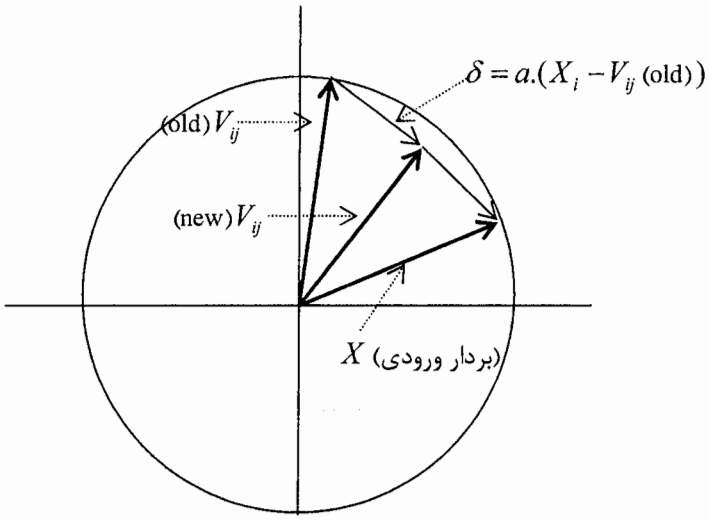
Xi : بردار ورودی

a : ضریب سرعت آموزش که در مرحله آموزش شبکه می تواند متغیر باشد .

پس می توان گفت که هر وزن مربوط به نرون برنده لایه رقـابتی متناسـب بـا اخـتلاف بـین خودش و بردار های ورودی X که به آن انتقال می یابد ، اصلاح می گردد . جهت تغییر بگونه ای است که اختلاف بین بردار وزن و بردار ورودی حداقل گردد . در شکل (۳-۵) این اصلاح وزن به صورت هندسی در فضای دو بعدی نشان داده شده است .

---

۱. (new) و (old) بترتیب هم ارز جدید و قدیم میباشند .



شکل (۳-۵) : تمایل بردار وزن به سوی بردار ورودی

در ابتدا بردار $X - V(old)$ را بدست می آوریم یعنی با وصل نمودن انتهای بردار V به انتهای بردار X بردار $X - V(old)$ ساخته می شود . سپس این بردار با ضرب دریک عدد مثبت کوچکتر از واحد مثل a ، کوتاهتر می گردد . در واقع یک تغییر به اندازه δ صورت می گیرد . در نهایت بردار وزن جدید یعنی $V(new)$ با وصل نمودن مبدا به انتهای بردار δ ایجاد می شود . از این تعبیر هندسی ملاحظه می گردد که اثر آموزش به واقع دوران بردار وزن به سوی بردار ورودی می باشد . متغیر a که ضریب سرعت آموزش می باشد معمولا" از ۰,۷ شروع شده و به هنگام آموزش بتدریج کاهش می یابد . به این ترتیب شبکه در ابتدا با طی گامهای بزرگتر، با سرعت بیشتری تربیت می شود و تدریجا" با کوچک کردن این گام ها همگرایی شبکه تضمین می شود .[۴].

لازم بذکر است یکه کردن بردارهای ورودی می تواند از ایجاد مشکلاتی چون همگرا شدن خروجی ها به یک بردار خاص و همچنین از کار افتادن شبکه جلوگیری کند . بهر حال بکار گیری تدابیر مناسب جهت ورود بردار های ورودی می تواند از خروجیهای غیر منطقی جلوگیری کند .

اگر یک بردار ورودی به هر نرون لایه رقابتی (لایه اول) اختصاص داده می شد ، می توانستیم این لایه را با یک اصلاح وزن آموزش دهیم . در این حالت وزن های نرون های برنده برابر با مؤلفه های بردار ورودی X می بودند معمولا" تعداد زوج های آموزشی زیادند و بسیاری از بردارهای ورودی هم مشابه می باشند در نتیجه شبکه باید به صورتی آموزش ببیند که بردار های مشابه منتهی به فعال شدن یک نرون لایه رقابتی گردد . دراین شرایط وزن های چنین پردازشگری میانگین بردار های ورودی است که این نرون را فعال نموده است . انتخاب یک مقدار کوچک برای a موجب کاهش اثرات هر گام آموزش و در نهایت باعث می شود مقادیر نهایی وزن ها ، میانگینی از بردارهای ورودی که با آن آموزش داده شده ، باشد . به واقع وزن های مختص یک نرون مقداری را بخود می گیرند که نزدیک مرکز بردارهای ورودی باشد که این نرون برای آنها برنده شده است .

در ادامه به آموزش مربوط به لایه دوم شبکه عصبی انتشار متقابل می پردازیم . با اعمال یک بردار ورودی ، خروجی های لایه رقابتی بدست می آیند . خروجی های لایه دوم با یک محاسبه معمولی حاصل می گردند سپس وزن های رابط هایی که به نرون های غیر صفر لایه اول متصل شده اند ، اصلاح می شوند . میزان اصلاح وزن متناسب با اختلاف بین وزن و خروجی واقعی لایه دوم می باشد یعنی :

$$Wk(new) = Wk(old) + b.(Yk - Wk(old)).Zj \qquad (۴۷-۳)$$

$Wk(new)$: مقادیر جدید یا اصلاح شده وزن های مربوط به لایه دوم

$Wk(old)$: مقادیر قبلی وزن ها ی مربوط به لایه دوم

b : ضریب سرعت یادگیری برای لایه دوم

Zj : بردار خروجی j ام لایه رقابتی (لایه اول)

Yk : بردار خروجی واقعی k ام

معمولا" مقدار b را ۰,۱ انتخاب نموده و همچنان که آموزش شبکه پیش می رود مقدار آن را کاهش می دهند . [۴]

آموزش شبکه در مرحله ای که مربوط به آموزش لایه دوم شبکه می باشد به صورت آموزش با نظارت صورت می گیرد ، یعنی الگوریتم دارای یک خروجی مشخص است که برای آن آموزش داده می شود . عمل خود سازمان ده و طبقه بندی بدون نظارت لایه رقابتی (لایه اول) خروجی هایی تولید می کند که توسط لایه دوم بر خروجی خواسته شده نگاشت می گردند .

در شبکه های عصبی انتشار متقابل ماتریس وزن ها بر اساس الگوریتم تعریف شده ، محاسبه میگردند و نیازی به معرفی این ماتریس ها در شروع آموزش نمی باشد . برای افزایش دقت در خروجی ها می توان اندازه شبکه را افزایش داد . سرعت آموزش شبکه سریع می باشد ، بدین معنی که با ارائه زوج های آموزشی در چند تکرار محدودآموزش شبکه تکمیل می گردد . [۴]

در صورت احتیاج به ماتریس های اولیه ، باید دقت نمود که ارائه ماتریس نامناسب اولیه ، شبکه را هنگام آموزش به سمت مقادیر بهینه و مناسب همگرا نخواهد کرد و بهمین علت باید در هنگام تهیه مقادیر اولیه حساسیت ویژه ای روی آنها نشان داد . بنا برتوصیه های محققان این رشته ، تاکید شده است که اولیه سازی وزنها باید با مقادیر تصادفی کوچک و غیرمتقارن صورت گیرد . [۴]

در شبکه های عصبی انتشار متقابل تعبیر هندسی گروه بندی نرون های لایه رقابتی بر اساس فاصله اقلیدسی یا حاصلضرب عددی ، مشکلاتی بهمراه دارد که برای رفع آنها ، اندازه بردار

تفاضل تعریف شده در رابطه (۳-۴۸) مطرح می گردد .

$$Di = \|Vi - Xi\| \qquad (۴۸-۳)$$

در رابطه (۳-۴۸) معیار نزدیک بودن بردار ورودی X به بردار V ، اندازه بردار ($Vi - Xi$) می باشد که با Di نمایش داده شده است . با انتخاب یک مقدار اختیاری برای D ، تعداد نرون های لایه رقابتی بصورت پویا محاسبه می شوند . گزینش مقدار D بستگی به نوع مسئله دارد انتخاب مقدار کوچک برای D منجر به گسترش شبکه می شود .

۳-۷ مراحل آموزش شبکه عصبی انتشار متقابل

تربیت شبکه عصبی انتشار متقابل شامل چندین گام بشرح زیر می باشد .

گام اول : با در نظر گرفتن مقدار $p = 1$ و همچنین بردار $Vp = X1$ آموزش شروع میشود . لازم به توضیح است که p در نهایت همان تعداد نرون های لایه رقابتی خواهد شد .

گام دوم : برای هر زوج از زوج های آموزشی $j = 2,...,m$ گام های سوم تا پنجم انجام گیرد .

گام سوم : برای هر نرون لایه رقابتی پارامتر Di که $i = 1,...,p$ محاسبه گردد .

$$Di = \|Xj - Vi\| \qquad (۴۹-۳)$$

(اندازه بردار تفاضل Xj و Vi)

گام چهارم : محاسبه i بطوریکه Di حداقل گردد .

گام پنجم : اگر Di محاسبه شده در مرحله چهارم از پارامتر D [۱] بزرگتر باشد در آن صورت باید $p = p + 1$ قرار دهیم و بردار $Vp = Xj$ و در غیر این صورت یعنی $D > D_i$

---

[۱] . D پارامتر گروه بندی است که توسط کاربر تعریف می شود و همانطور که قبلا" توضیح داده شد ، با توجه به نوع مسئله این مقدار می تواند متفاوت باشد .

خواهیم داشت:

$$Vi(new) = Vi(old) + a.(Xj - Vi) \qquad (۵۰-۳)$$

گام ششم: برای هر زوج از مجموعه زوج های آموزشی $j = 1,...,m$ با قرار دادن $N = 1$ گام های هفت وهشت اجرا می گردد .

گام هفتم : برای هر بردارورودی آموزشی X پارامتر Di که کوچکتر از D است ، محاسبه گردد.

گام هشتم : بردار $Wi = Yj$ و سپس $N = N + 1$ ، اگر $N > p$ باشد آنگاه :

$$Wi(new) = (1 - b).Wi(old) + b.Yj \qquad (۵۱-۳)$$

گام نهم : ضرایب سرعت آموزش a و b بترتیب برای هر زوج آموزشی X و Y ، کاهش یافته و سپس مراحل آموزش از ابتدا دوباره تکرار می شود تا همگرایی حاصل گردد .

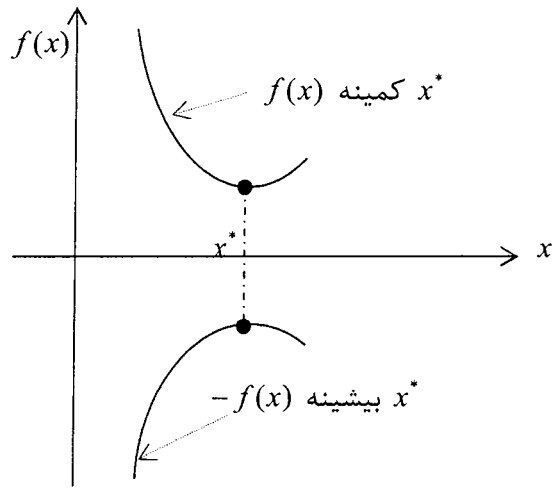
در پایان لازم بذکر است که آموزش شبکه عصبی انتشار متقابل در مقایسه با شبکه عصبی انتشار برگشتی سریع تر می باشد و این یک مزیت برای شبکه عصبی انتشار متقابل می باشد و در مسائلی که احتیاج به آموزش سریع می باشد ، توانمندی این نوع شبکه بیشتر نمایان می شود .

فصل چهارم

مقدمه ای بر بهینه سازی

۴-۱ کلیات

دستیابی به بهترین نتیجه در شرایط داده شده را بهینه سازی می گویند . در طراحی ، ساخت و نگهداری هر سیستم مهندسی ، مهندسان باید تصمیمات مدیریتی بسیاری را در چند مرحله بگیرند . هدف نهایی چنین تصمیماتی ، کمینه کردن تلاش لازم و یا بیشینه کردن سود مورد نظر است . تلاش لازم یا سود مورد نظر در هر وضعیت عملی را می توان به صورت تابعی از متغیرهای تصمیم مشخص بیان کرد . بنابراین می توان بهینه سازی را بعنوان فرایند یافتن شرایطی که مقدار بیشینه یا کمینه یک تابع را بدست می آورد ، تعریف کرد . از شکل (۴-۱) می بینیم که اگر نقطه x^* منطبق بر مقدار کمینه تابع $f(x)$ باشد ، این نقطه بر مقدار بیشینه تابع $-f(x)$ هم منطبق است . بنابراین بهینه سازی را می توان به معنای کمینه سازی در نظر گرفت زیرا بیشینه یک تابع را می توان با جستجوی کمینه منفی آن تابع پیدا کرد .



شکل (۴-۱) :کمینه $f(x)$ ، بیشینه $-f(x)$ است .

برای حل همه مسائل بهینه سازی روش یگانه ای وجود ندارد به همین دلیل برای حل گونه های مختلف بهینه سازی ، روشهای بهینه سازی مختلفی بسط یافته اند . روش های جستجوی بهینه را با عنوان روش های برنامه ریزی ریاضی هم می شناسند ، که عموما" به صورت بخشی از تحقیق در عملیات مطالعه می شوند . تحقیق در عملیات شاخه ای از ریاضیات است که به کاربرد روش های عملی در مسائل تصمیم گیری و رسیدن به بهترین پاسخ ، یا پاسخ بهینه می پردازد [۵].

روش های برنامه ریزی ریاضی ، در یافتن کمینه تابعی از چند متغیر ، با توجه به مجموعه ای از قیدها مورد استفاده قرار می گیرند . درمسائلی که با مجموعه ای از متغیر های تصادفی با توزیع احتمال مشخص توصیف می شوند از روش های فرایند تصادفی استفاده می شود . روش های تصادفی امکان تحلیل داده های آزمایشی و ساختن مدل های تجربی را برای دستیابی به دقیق ترین ارائه وضعیت های فیزیکی فراهم می سازند .

بهنیه سازی در مفهوم گسترده خود ، می تواند در حل مسائل مهندسی به کار گرفته شود که نمونه هایی از این مسائل به شرح زیر است :

۱- طراحی سازه های مهندسی عمران مانند خرپا ، قاب ها ، پی ها ، پلها و ... با وزن کمینه

۲- طراحی سیستمهای منابع آب با سود بیشینه

۳- طراحی پمپ ها ، توربین ها و تجهیزات انتقال حرارت با بازدهی بیشینه

۴- برنامه ریزی ، کنترل و زمان بندی بهینه تولید

۵- طراحی بهینه تجهیزات و واحدهای فرایند شیمیایی

۶- طراحی تجهیزات حمل مواد مانند نقاله ها ، کامیون ها و واگن ها با بازده بیشین

۲-۴ روش بهینه سازی

در حالت کلی یک مساله بهینه سازی یا برنامه ریزی ریاضی را می توان به صورت زیر بیان کرد :

بردار $X = (x_1,...,x_n)$ را به گونه ای بیابید که $f(X)$ به شرط قیود زیر کمینه گردد :

$$g_j(X) \le 0 \qquad j = 1,...,m \qquad (۱-۴)$$

که X یک بردار n بعدی است که بردار طراحی نامیده می شود . $f(X)$ تابع هدف و $g_j(X)$ قید مساله بهینه سازی است . لزومی ندارد که تعداد متغیرهای n و تعداد قیدهای m ارتباطی به هم داشته باشند . این مساله یک مساله بهینه سازی مقید نامیده می شود . برخی از مسائل بهینه سازی دارای قید نیستند و در حالت کلی می توان آنها را به صورت زیر بیان کرد :

بردار $X = (x_1,...,x_n)$ را به گونه ای بیابید که $f(X)$ کمینه گردد . چنین مسائلی را مسائل بهینه سازی نا مقید می نا مند .

هر سیستم یا عنصر مهندسی با مجموعه ای از کمیت ها بیان میشود که برخی از آنها به صورت متغیر هایی در فرایند تصمیم گیری ظاهر می شوند . کمیت های معینی که در خارج از مساله دارای مقدار ثابتی هستند ، پارامترهای معلوم نام دارند . همه کمیت های دیگر به صورت متغیر هایی در فرایند طراحی رفتار می کنند و متغیر های طراحی یا تصمیم نامیده می شوند این متغیر ها را با x_i و $i = 1,...,n$ است ، نشان می دهیم . مجموعه متغیرهای طراحی به صورت بردار طراحی $X = (x_1,...,x_n)$ ارائه می شوند .

یک فضای n بعدی مشخص را که هر محور مختصات آن بیانگر یک متغیر طراحی x_i و $i = 1,...,n$ است ، در نظر بگیرید . چنین فضایی را فضای متغیر های طراحی گویند . هر نقطه در

این فضای n بعدی ، یک نقطه طراحی نامیده می شود . این نقطه یک جواب امکان پذیر یا امکان ناپذیر را برای مساله طراحی ارائه می کند .

در بسیاری از مسائل عملی نمی توان متغیر های طراحی را به دلخواه انتخاب کرد ، بلکه این متغیر ها باید ویژگیهای عملی مشخص و دیگر نیازمندیها را برآورده کنند . قیدهایی را که باید به منظور تهیه یک طرح مورد قبول ، برآورده شوند ، قید های طراحی گویند . قیدهایی که محدودیتی را در رفتار و کار سیستم ارائه می کنند ، قید های رفتاری یا عملی نامیده می شوند . قیدهایی که محدودیت های فیزیکی مانند موجودیت ، قابلیت بسته بندی و قابلیت حمل را برای متغیر های طراحی ارائه می کنند ، قید های هندسی نامیده می شوند .

یک مساله بهینه سازی را که دارای قیدهای نامساوی $g_j(X) \le 0$ است را در نظر می گیریم مجموعه مقادیر X که در رابطه $g_j(X) = 0$ صدق کنند ، یک ابر صفحه را در فضای طراحی تشکیل می دهند که آن را سطح قید گویند . این ابر صفحه خود یک زیر فضای $(n-1)$ بعدی است که n ، تعداد متغیر های طراحی را نشان می دهد . سطح قید ، فضای طراحی را به دو ناحیه تقسیم می کند . در یک ناحیه $g_j(X) \le 0$ و در دیگری $g_j(X)$ بزرگتر از صفر است . بنابراین ، در ابر صفحه قرار دارند ، به طور بحرانی در قید $g_j(X)$ صدق می کنند . در حالیکه نقاط واقع در ناحیه $g_j(X) > 0$ امکان ناپذیر یا غیر قابل قبول ، و نقاطی را که در ناحیه $g_j(X) < 0$ قرار دارند امکان پذیر یا قابل قبول هستند . مجموعه همه سطوح قید $g_j(X) = 0$ ، $j = 1,...,m$ که ناحیه امکان پذیر را جدا می کنند ، سطح قید مرکب نامیده می شود .

روش های طراحی معمول ، ما را در یافتن یک طرح قابل قبول یا کافی یاری می دهند . این

نوع طراحی تنها نیازمندیهای عملی و دیگر نیازمندیهای مساله را برآورد می سازد لـیکن عمومـا" تنها قابل قبول بودن یک طرح مورد نظر نیست و هدف از بهینه سازی انتخاب بهترین طرح از بین طرح های قابل قبول موجود می باشد . بنابراین باید معیاری برای مقایسه طرحهـای قابـل قبـول مختلف و انتخاب بهترین آنها تعیین شود . چنین معیاری که طرح ، نسبت به آن بهینه می شـود را تابع مزیت و یا تابع هدف می نامند . انتخاب تابع هدف به طبیعت مساله بستگی دارد.

در طرح های مهندسی سازه اغلب تابع هدف کمینه سازی وزن یا هزینـه اسـت و در طراحـی سیستمهای مکانیکی هدف بیشینه کردن بازده مکانیکی اسـت . بنـابراین انتخـاب تـابع هـدف در بیشتر مسائل طراحی آسان به نظر می رسد . با این وجود ممکن است در حالاتی بهینه سـازی بـا یک معیار مشخص ، به نتایجی بیانجامد که نسبت به معیار دیگر رضایت بخش نباشد . برای مثال ، در سازه های نامعین استاتیکی طرحی با تنش کامل ممکن است دارای کمترین وزن و در نتیجه ارزان ترین طرح نباشد . با این حال انتخاب تابع هدف ، یکی از مهمترین تصمیمات در کل فرایند بهینه سازی است .

در بعضی حالات لازم است که بیش از یک معیار به طور همزمان برآورده شود . برای مثال در انتقال یک توان مشخص توسط یک جفت چرخ دنده ممکن است چرخ دنده ها را برای وزن کمینه و بازده بشینه طراحی کنیم . یک مساله بهینه سازی با توابع هدف چندگانه را با عنوان یک مساله برنامه ریزی با چند هدف (یا مساله برنامه ریزی آرمانی) می شناسند . با هدف های چند گانه امکان پیچیدگی مساله زیاد می شود . یک راه ساده برای حل چنین مسائلی اینست که تابع هدف را به عنوان ترکیبی خطی از توابع هدف چندگانه پیچیده در نظرگرفت . یعنی اگر $f_1(X)$ و $f_2(X)$ دو تابع هدف ممکن باشند، برای بهینه سازی یک تابع هدف جدید بصورت رابطه(۴-۲)

در نظر می گیریم :

$$f(X) = a_1.f_1(X) + a_2.f_2(X) \qquad (۲-۴)$$

که مقادیر a_1 و a_2 ثابت بوده و درجه بیانگر اهمیت یک تابع هدف نسبت به دیگری است .

۳-۴ روش های بهینه سازی مقید

برای حل مسائل بهینه سازی مقید روش های زیادی وجود دارند . همه این روش ها را می توان در دو طبقه روش های مستقیم و روش های غیر مستقیم دسته بندی کرد .

در روش های مستقیم ، قیدها به شیوه صریح بکار گرفته می شوند در حالیکه در روش های غیر مستقیم ، مساله مقید به عنوان دنباله ای از مسائل کمینه سازی نا مقید حل می شوند .

از جمله روش های مستقیم می توان به روش سیمپلکس و روش برش صفحه اشاره کرد و همچنین از جمله روش های غیر مستقیم بهینه سازی مقید می توان به روش تبدیل متغیر ها و روش تابع جریمه اشاره کرد .

الف - روش کمپلکس[۱] : باکس[۲] روش کمپلکس را برای حل مسائل کمینه سازی مقید مطرح کرد . در این روش هدف کمینه کردن تابع $f(X)$ است به شرطی که $g_j(X) \le 0$ ، $j = 1,...,m$. ایده اصلی روش کمپلکس ، تشکیل دنباله ای از شکل های هندسی است که هر یک دارای $k = n+1$ راس در یک فضای n بعدی هستند (n بعد بردار X است) .

---

[۱] – برای بهینه سازی دقیق در این تحقیق ، از این روش سود خواهیم برد .

[۲] - Box

مراحل بهینه یابی دقیق توسط روش کمپلکس[1] به شرح زیر می باشد :

مرحله اول : به تعداد k نقطه که هر نقطه شامل تمام متغیر های مسئله بهینه یابی است و در تمام قیود مسئله صدق می کند را پیدا می کنیم (k باید از تعداد متغیر ها بیشتر باشد) . درعمل تنها از یک نقطه امکان پذیر مانند X_1 استفاده می کنند و بقیه نقاط را بترتیبی که در ادامه اشاره می شود ، پیدا می کنند . نقطه بعدی بصورت تصادفی انتخاب شده و تعیین می کنیم که آیا در تمام قیدها صدق می کند یا خیر . در صورتی که تمام قیود را برآورده سازد بعنوان نقطه امکانپذیر دیگر پذیرفته می شود وگرنه این نقطه را اصلاح می کنیم ، بدین صورت که باندازه نصف فاصله آن تا مرکزیت نقاط باقیمانده به سمت مرکز حرکت می دهیم (منظور از نقاط باقیمانده ، نقاط پذیرفته شده یا امکانپذیر فعلی هستند که نقطه آغازین داده شده X_1 را شامل می شوند) . اگر نقطه اصلاح شده بازهم در برخی از قیدها صدق نکند فرایند حرکت به نصف فاصله تا مرکزیت نقاط را تا یافتن یک نقطه امکان پذیر ادامه می دهیم . با پیشروی بدین روش ، سرانجام می توانیم نقاط امکان پذیر مورد نیاز را پیدا کنیم .

مرحله دوم : تابع هدف را در هر یک از k نقطه محاسبه می کنیم . اگر نقطه (رأس) Xh منطبق بر بیشترین مقدار تابع هدف باشد ، از فرایند انعکاس برای یافتن نقطه جدید Xr به صورت زیر استفاده می کنیم :

$$Xr = (1+a).Xo - a.Xh \qquad (۳-۴)$$

در رابطه (۳-۴) مقدار a را بزرگتر از یک انتخاب می کنند و مقدار Xo نیز به صورت رابطه (۴-۴) تعریف می شود :

[1] - اطلاعات بیشتر در مورد این روش در کتابهای بهینه یابی بخصوص مرجع [۵] قابل دسترسی است .

$$Xo = 1/(k-1) * \sum_{i=1, i\neq h}^{k} Xi \qquad (۴-۴)$$

(Xo مرکزیت تمام نقاط بجز Xh می باشد)

مرحله سوم : چون مسئله مقید می باشد ، نقطه Xr باید در تمام قیود صدق کند تا به آن نقطه امکان پذیر اطلاق گردد بهرحال اگر Xr امکانپذیر باشد و همچنین اگر $f(Xr) \leq f(Xh)$ باشد به جای نقطه Xh ، نقطه Xr را قرار می دهیم و به مرحله دوم بر می گردیم . لازم به توضیح است که تابع $f(X)$ همان تابع هدف (وزن خرپا) می باشد .

درصورتیکه رابطه $f(Xr) \leq f(Xh)$ برقرار نباشد ، مقدار a را در رابطه (۴-۳) کاهش میدهیم با این تغییر یک نقطه آزمون جدید Xr بدست خواهیم آورد سپس آنرا برای برقراری رابطه $f(Xr) \leq f(Xh)$ تست می کنیم ، در هر صورت اگر رابطه قبلی صادق نبود باز هم مقدار a را کاهش می دهیم و تا جایی ادامه می دهیم که بلاخره نقطه بهبود یافته Xr شرط $f(Xr) \leq f(Xh)$ را برآورده سازد .

مرحله چهارم : در هر مرحله که نقطه منعکس شده Xr (که در مرحله قبل پیدا شده است) هر یک از قیود مسئله را نقض کند ، آن را تا رسیدن به امکان پذیری ، به نیمه راه فاصله اش تا مرکزیت حرکت می دهیم یعنی :

$$Xr = 1/2 * (Xo + Xr) \qquad (۴-۵)$$

اگر دوباره قیدی را نقض کند این روش را تا وقتیکه کمپلکس درمرکزیت محونشده است ، ادامه خواهیم داد تا نقطه امکان پذیر حاصل شود .

مرحله پنجم : هر بار که بدترین نقطه (Xh) از کمپلکس فعلی با یک نقطه جدید جایگزین میشود ، کمپلکس تعدیل می یابد و باید فرآیند را برای همگرایی تست کنیم . فرآیند را وقتی

همگرا شده فرض می کنیم که شرط زیر برقرار باشد یعنی انحراف معیار مقدارتابع به اندازه کـافی کوچک گردد :

$$\left\{ 1/k * \sum_{j=1}^{k} \left[f(\tilde{X}) - f(Xj) \right]^2 \right\}^{0.5} \leq \varepsilon \qquad (۶-۴)$$

$\tilde{X}$: مرکزیت همه k راس (نقطه) کمپلکس فعلی و ε عددی کوچـک و از پـیش تعریـف شده میباشد .

ب - روش صفحه برش : روش صفحه برش توسط چنی[۱] برای حـل مـسائل برنامـه ریـزی محدب ارائه شد . در این روش قیدها با استفاده از بسط سری تیلور خطـی مـی شـوند . بنـابراین ناحیه امکانپذیر با کرانهای خطی شده تقریب زده می شود . از آنجا کـه ناحیـه امکـان پـذیر یـک مساله محدب یک مجموعه محدب است ، همه قید های خطی شده همواره در خارج ناحیه امکان پذیر قرار می گیرند . با فرض خطی بودن تابع هدف ، می تـوانیم مـساله برنامـه ریـزی خطـی (تقریبی) حاصل را با روش سیمپلکس حل کنیم اگر جواب مساله برنامه ریـزی خطـی بـه انـدازه کافی دقیق نباشد ، قید های نافذ (فعال) را حول نقطه فعلی خطی سازی می کنیم و یک مساله خطی جدید را فرمول بندی کرده و با استفاده از روش سیمپلکس آن را حل کنیم . این روش را تا یافتن پاسخ دقیق ادامه می دهیم .

برای استفاده از روش صفحه برش باید تابع هدف خطی باشد . یک مساله بهینه سازی با یـک تابع هدف غیر خطی را همواره می توان به مساله ای با تابع هدف خطی به صورت زیر تبدیل کرد. فرض کنید مساله داده شده به صورت زیر باشد :

بردار $X = (x_1, \ldots, x_n)$ را به گونه ای بیابید که تابع $f(X)$ بدون نقض قیود زیرکمینه گردد .

---

[۱]- Cheney

$$g_j(X) \le 0 \qquad j = 1,...,m \qquad (۴-۷)$$

یک متغیر جدید مثلا" x_{n+1} را معرفی و این مساله را به یک مساله معادل که در ادامه می آید تبدیل می کند :

بردار $X = (x_1,...,x_n,x_{n+1})$ را به گونه ای بیابید که x_{n+1} را به شرط قید های زیر کمینه کند :

$$g_j(x_1,...,x_n) \le 0 \qquad j = 1,...,m \qquad (۴-۸-الف)$$

$$g_{m+1}(x_1,...,x_{n+1}) = f(x_1,...,x_n) - x_{n+1} \le 0 \qquad (۴-۸-ب)$$

با افزودن یک متغیر و یک قید ، مساله اصلی با یک تابع هدف غیر خطی رابطه (۴-۷) را به یک تابع هدف خطی ، روابطه (۴-۸) تبدیل کرده ایم . بنابراین بدون از دست دادن کلیت ، می توانیم مساله داده شده را به صورت زیر فرض کنیم .

بردار X را به گونه ای بیابید که تابع جدید خطی شده یعنی $f(X) = C^T.X = \sum_1^n c_i.x_i$ به شرط قیدهای زیر کمینه گردد .

$$g_j(X) \le 0 \qquad j = 1,...,m \qquad (۴-۹) \quad .$$

ج - روش تغییر متغیر : یکی از روشهای غیر مستقیم بهینه سازی مقید ، روش تغییر متغیر ها می باشد که تشریح خواهد شد .

اگر قید های $g_j(X)$ ، $j = 1,...,m$ توابع صریح و ساده ای از متغیر های x_i ، $i = 1,...,n$ باشند ، ممکن است بتوان متغیر های مستقل را به گونه ای تبدیل کرد که قید ها خودبخود برآورده شوند بنابراین با تغییر متغیر ها امکان تبدیل یک مساله بهینه سازی مقید به یک مساله بهینه سازی نا مقید وجود خواهد داشت . یکی از انواع قید هایی که می توان آن را

بدین روش برآورده ساخت ، حالتی است که یک متغیر از پایین و بالا با اعداد ثابت مشخصی محدود شده باشد ، یعنی :

$$l_i \le x_i \le u_i \qquad (۱۰-۴)$$

که l_i و u_i به ترتیب کران های پایین و بالا هستند . این قید ها را می توان با تبدیل متغیر x_i به صورت زیر برآورده ساخت :

$$x_i = l_i + (u_i - l_i).\sin^2 y_i \qquad (۱۱-۴)$$

بطوریکه y_i متغیر جدیدی است که می تواند دارای هر مقداری باشد .

در حالت خاصی که متغیر x_i در فاصله (۰،۱) محدود شده است ، می توان هر یک از تبدیلات زیر را به کار برد :

$$x_i = \sin^2 y_i \qquad (۱۲-۴-الف)$$

$$x_i = \cos^2 y_i \qquad (۱۲-۴-ب)$$

$$x_i = y_i^2/(1+y_i^2) \qquad (۱۲-۴-ج)$$

اگر متغیر x_i محدود به داشتن تنها مقادیر مثبت باشد ، باید تبدیل به صورت زیر داده باشد :

$$x_i = |y_i| \qquad (۱۳-۴-الف)$$

$$x_i = y_i^{\,2} \qquad (۱۳-۴-ب)$$

$$x_i = e^{y_i} \qquad (۱۳-۴-ج)$$

از طرف دیگر ، اگر متغیر ها محدود به داشتن تنها مقادیر بازه [-۱,+۱] باشد ، تبدیل به صورت زیر داده می شود :

$$x_i = \sin y_i \qquad (۱۴-۴-الف)$$

$$x_i = \cos y_i \qquad (۱۴-۴-ب)$$

$$x_i = 2y_i/(1+y_i^2) \qquad (۱۴-۴-ج)$$

پس از اعمال این تبدیل ها ، کمینه نامقید تابع هدف نسبت به متغیر های جدید y_i جستجو می شود .

د - روش تابع جریمه : یکی دیگر از روش های غیر مستقیم بهینه یابی مقید ، روش تابع جریمه می باشد . این روش مساله بهینه سازی اصلی را به فرمول دیگری تبدیل می کند به گونه ای که جواب های عددی با حل دنباله ای از مسائل کمینه سازی نا مقید جستجو می شوند .

فرض کنید مساله بهینه سازی اصلی به صورت زیر باشد :

بردار X را به گونه ای بیابید که تابع $f(X)$ را همراه با قید های زیر کمینه کند .

$$g_j(X) \le 0 \qquad j = 1,...,m \qquad (۱۵-۴)$$

این مساله با ساختن یک تابع به شکل زیر ، به یک مساله کمینه سازی نا مقید تبدیل می شود .

$$\Phi_k = \Phi(X, r_k) = f(X) + r_k . \sum_1^m G_j . [g_j(X)] \qquad (۱۶-۴)$$

که در رابطه (۱۶-۴) G_j تابعی از قید g_j و r_k یک ثابت مثبت است که پارامتر جریمه نامیده می شود .

عبارت دوم در سمت راست رابطه (۱۶-۴) را عبارت جریمه می نامند که از اهمیت ویژه ای برخوردار است . اگر کمینه سازی نا مقید تابع Φ برای دنباله ای از مقادیر پارامتر جریمه r_k و $k = ۱،۲,...$ تکرار شود ، ممکن است پاسخ آن به پاسخ مساله اصلی که در رابطه (۱۵-۴) بیان شده است ، همگرا شود . بهمین دلیل روش تابع جریمه را با عنوان روش کمینه سازی نامقید متوالی هم می نامند .

برخی از شکل های رایج تابع جریمه G_j برای فرمول بندی عبارتند از :

$$G_j = -1/g_j(X) \qquad (۱۷-۴)$$

$$G_j = \log[-g_j(X)] \qquad (۱۸-۴)$$

۴-۴ بهینه سازی هندسی

بسیاری از مطالعات و تحقیق های انجام شده در زمینه بهینه سازی سازه ها ، مربوط به طراحی بهینه سطح مقطع اعضاء سازه می شود بدون اینکه متغیر دیگری برای مساله بهینه سازی مطرح باشد یا بعبارت دیگر یک نوع متغیر آن هم از جنس سطح ، مطرح می باشد ولی در طراحی بهینه هندسی علاوه بر اینکه سطح مقطع اعضاء متغیر بوده ، نوع دیگری از متغیر ها نیز مطرح می باشند که موقعیت تعدادی از گره ها می باشد پس در این نوع بهینه سازی علاوه بر سطح مقطع باید بدنبال موقعیت مناسب گره های متغیر نیز باشیم .

بهینه سازی هندسی تاثیر زیادی در بهبود و افزایش کارایی سازه دارد و البته نباید فراموش کنیم که در این نوع بهینه سازی (بهینه یابی هندسی) محاسبات و عملیات بمراتب زیادترو پیچیده تری پیشرو خواهیم داشت .

در مسائل بهینه یابی هندسی سازه ها سطح مقطع اعضاء و مختصات گره های متغیر سازه طوری تعیین می شوند که تابع $Z = f(X,Y)$ یا وزن سازه ، حداقل شود بدون اینکه قیدی نقض گردد یعنی :

$$Z = f(X,Y) \quad \Rightarrow \quad Min \qquad (۱۹-۴)$$

$$\sigma^L \le \sigma(X,Y) \le \sigma^U \qquad (۲۰-۴)$$

$$Y^L \le Y \le Y^U \qquad (۲۱-۴)$$

$$0 \le X \qquad (۲۲-۴)$$

در روابط فوق Z وزن سازه ، f تابع وزن ، X بردار سطح مقطع اعضاء و Y بردار مختصات گره های متغیر سازه هستند . همچنین $\sigma(X,Y)$ ، σ^L و σ^U بترتیب تنش در اعضاء سازه ، محدودیت تحتانی و فوقانی تنش ها هستند . Y^L و Y^U بترتیب بردار محدودیت تحتانی و فوقانی مختصات گره ها می باشد و در نهایت اینکه سطح مقطع اعضاء مقادیر مثبت یا صفر را می تواند به خود اختصاص دهند .

در این تحقیق برای بهینه سازی دقیق خرپا از روش کمپلکس (بخش (۴-۳)) بهره برداری می شود و برای بهینه سازی تقریبی از شبکه های عصبی مصنوعی انتشار متقابل استفاده می شود نهایتا" نتایج این دو برای ارزیابی نهایی ، با هم مقایسه خواهد شد .

فصل پنجم

بهینه یابی سه خرپا توسط شبکه عصبی انتشار متقابل

۵-۱ مقدمه

در طراحی بهینه سازه ها لازم است سازه در دفعات متوا لی تحلیـل شـود زیـرا توپولـوژی و مشخصات سازه ای درپروسه بهینه سازی مرتب تغییر پیدا مـی کنـد چنانچـه تحلیـل مجـدد بـا استفاده ازروش های دقیق انجام گیرد ، محاسبات پیچیده برای بهینه سازی بطورقابل ملاحظه ای افزایش می یابد در چنین شرایطی روش های تقریبی تحلیل سازه جانشین مناسبی برای تحلیـل دقیق می باشند . دستیابی به روشی برای تحلیل تقریبی و در عین حال کارا بسیار مطلوب اسـت خصوصا" برای سیستم هایی که از لحاظ سازه ای استاندارد هستند . یکی از مشخصات بارزی کـه استفاده از شبکه های عصبی را از روشهای متداول محاسباتی متمایز می کند ، آموزش این شبکه ها برای یادگیری است . یک شبکه آموزش می بیند تا یک سری نتایج مورد انتظار را با توجه بـه اطلاعات ورودی خاص ، ایجاد نمایـد . هرگـاه شـبکه ای آمـوزش ببینـد ، مـی تـوان از آن بطـور نامحدود استفاده کرد و با استفاده از داده های ورودی ، اطلاعاتی بدست آورد که شبکه قـبلا" بـا آن مواجه نبوده است . بعبارت دیگر ، شبکه بر اساس تجربه آموزش می بیند و بر خلاف آنچـه در محاسبات متداول روی می دهد ، با وجود یک اشتباه دراطلاعات ورودی تمامی محاسـبات دچـار اشکال نمی گردد .

بدون شک تحلیل سازه را بطورمؤثر با بکار گیری ضوابط ریاضی توسعه داده شده و استفاده از رایانه میتوان انجام داد حتی برای سازه های بزرگ و پیچیده نـرم افزارهـای بـسط و توسـعه داده شده به طور گسترده در دسترس اند . مع هذا مواردی یافت می شـود کـه بکـار گیـری روشـهای تقریبی مؤثر واقع می شود و محاسبات پیچیده را تا حـدودی سـاده مـی کنـد مخـصوصا" بـرای حالتهایی که سازه دارای شکل و فرم مشخصی است ولی نیاز به تغییر پارامتر و تکرار تحلیل سازه

به دفعات ، لازم و ضروری است . در بهینه سازی سازه ها ، هر مرتبه که تغییر در یکی از پارامتر های طراحی روی می دهد ، تحلیل سازه مجددا" صورت می گیرد . در چنین مواردی بکارگیری یک شبکه عصبی آموزش دیده می تواند جایگزین مناسبی برای تحلیل دقیق سازه باشد . باید در نظر داشت که شبکه های عصبی درمقایسه باروشهای متداول ریاضی، نیازبه روابط ریاضی کمتری دارند و بنابراین توان آنها به مدل نمودن مجهولات و یا روابط نا معین در شرایط نا مشخص بیشتراست.

طراحی سازه ها در دو مرحله متفاوت انجام می شود . مرحله اول تحلیل سازه بر اساس نیرو های اعمالی صورت می گیرد و منتهی به تعیین نیروها دراجزای مختلف تشکیل دهنده سازه می گردد. در مرحله دوم با توجه به نیروهای موجود در عضوها ، مقاطع مناسب برای اجزای مختلف سازه انتخاب می گردد . در نهایت سازه دارای وزن مشخصی خواهد بود که در اکثر مواقع بهینه نیست ، بدین معنی که با توجه به نیروهای موجود در اعضاء ، مقاطع با قید حداقل شدن وزن سازه تعیین نشده اند . در علم بهینه یابی کاهش وزن سازه ، تابع هدف نامیده می شود که تحت اثر قیودی که می تواند محدوده ای از تنش ها و یا غیره باشد ، قرار می گیرد .

بهینه یابی سازه ها بدلیل طبیعت غیر خطی بودن آن احتیاج به روش های تکراری دارد . غیر خطی بودن ناشی از معادله جبری موجود در شرایط بهینه است و معادلات قیود نیز نسبت به متغیر های طراحی غیر خطی هستند . هر مرحله از تکرار روش بهینه یابی به دو گام تقسیم میگردد . درگام اول ، سازه تحلیل می گردد تا پاسخ آن نسبت به نیروهای اعمال شده بدست آید. برای تحلیل سازه ابتدا ماتریس سختی کل سازه را تشکیل می دهیم و پس از آن مقدار نیروهای داخلی اعضاء را محاسبه می کنیم که برای این منظور از حل دستگاه معادلات رابطه (۵-۱) سود

خواهیم برد .

$$[K]*\{\Delta\}=\{F\} \qquad (۱-۵)$$

که $[K]$ ماتریس سختی کل سازه ، $\{\Delta\}$ بردار تغییر مکان و $\{F\}$ بردار نیروهای وارده به خرپا است در گام دوم ، متغیر های طرح براساس ضوابط بهینه یابی اصلاح می گردند . تعداد تکرار های لازم برای همگرا شدن به سمت شرایط بهینه بستگی به تعداد عضو های سازه ونیز سطح مقطع آنها دارد زیرا در بسیاری موارد ، نیروی ایجاد شده در عضو ها نسبت به تغییر سطح مقطع عضو ها حساس می باشد . با افزایش تعداد متغیر های طراحی ، زمان محاسبه بطور قابل ملاحظه ای افزایش می یابد ودر بسیاری ازالگوریتم های بهینه یابی ، افزایش تعداد متغیر های طراحی منتهی به ناپایداری عددی می گردد . اکثر روشهای بهینه یابی دربرخورد با توابع نا پیوسته و همچنین مسائلی که غیر خطی بودن آنها شدید است بکندی همگرا می گردند وحتی در بسیاری موارد ، همگرایی صورت نمی گیرد .

در استفاده از روشهای بهینه یابی در طراحی مسائل عملی مورد نیاز ، بندرت می توان از بهینه بودن جواب در حضور قیود مفروض اطمینان حاصل کرد . دلیل عمده ، وجود جوابهای متعدد برای مسئله یا نحوه ارائه اطلاعات بطریقی است که منتهی به ناپایداری عددی یا همگرایی ضعیف در الگوریتم بهینه سازی می شود . در عمل ، بهترین روش شروع بهینه یابی استفاده از چندین بردار اولیه متغیر هاست و اگر بهینه یابی تحت شرایط مختلف به جواب یکسان رسید میتوان گفت که جواب حاصل واقعا" بهینه می باشد . در ضمن با استفاده از روابط ریاضی جنبی می توان از گیرافتادن در حداقل های نسبی اجتناب کرد .

با توجه به اینکه در مسائلی همچون بهینه یابی سازه ها بدلیل تغییر در پارامتر های طراحی ،

ناچار به تحلیل سازه بدفعات هستیم ، نیاز به یک شبکه عصبی که سرعت آموزش آن زیاد بوده بطوریکه بتوان پس از آموزش از آن بعنوان یک تحلیل کننده سریع استفاده نمود ، بسیار مشهود است . برای حل این مشکل از شبکه عصبی انتشار متقابل استفاده خواهد شد . این نوع شبکه علاوه بر داشتن سرعت زیاد در آموزش ، پاسخ در حد قابل قبول و منطقی ارائه می کند که بکارگیری آن در تحلیل سازه ها و یا بهینه یابی ، منتهی به جواب هایی نزدیک به بهینه با کاهش قابل ملاحظه محاسبات پیچیده می گردد .

۵-۲ بهینه یابی خرپا با استفاده از شبکه عصبی انتشار متقابل

برای طراحی بهینه خرپا ، ابتدا بهینه یابی با استفاده از یک روش دقیق صورت می گیرد و سپس نتایج حاصله با پاسخ شبکه عصبی مقایسه می گردد . تابع هدف در بهینه سازی خرپا ، وزن خرپا است و متغیر های طرح عبارتند از سطح مقطع اعضاء و مختصات نقاط گرهی متغیر می باشند. تابع هدف در اینگونه سازه ها را می توان با رابطه زیر نمایش داد :

$$W = \sum_{i=1}^{NE} \gamma \cdot A_i \cdot L_i \qquad (۵-۲)$$

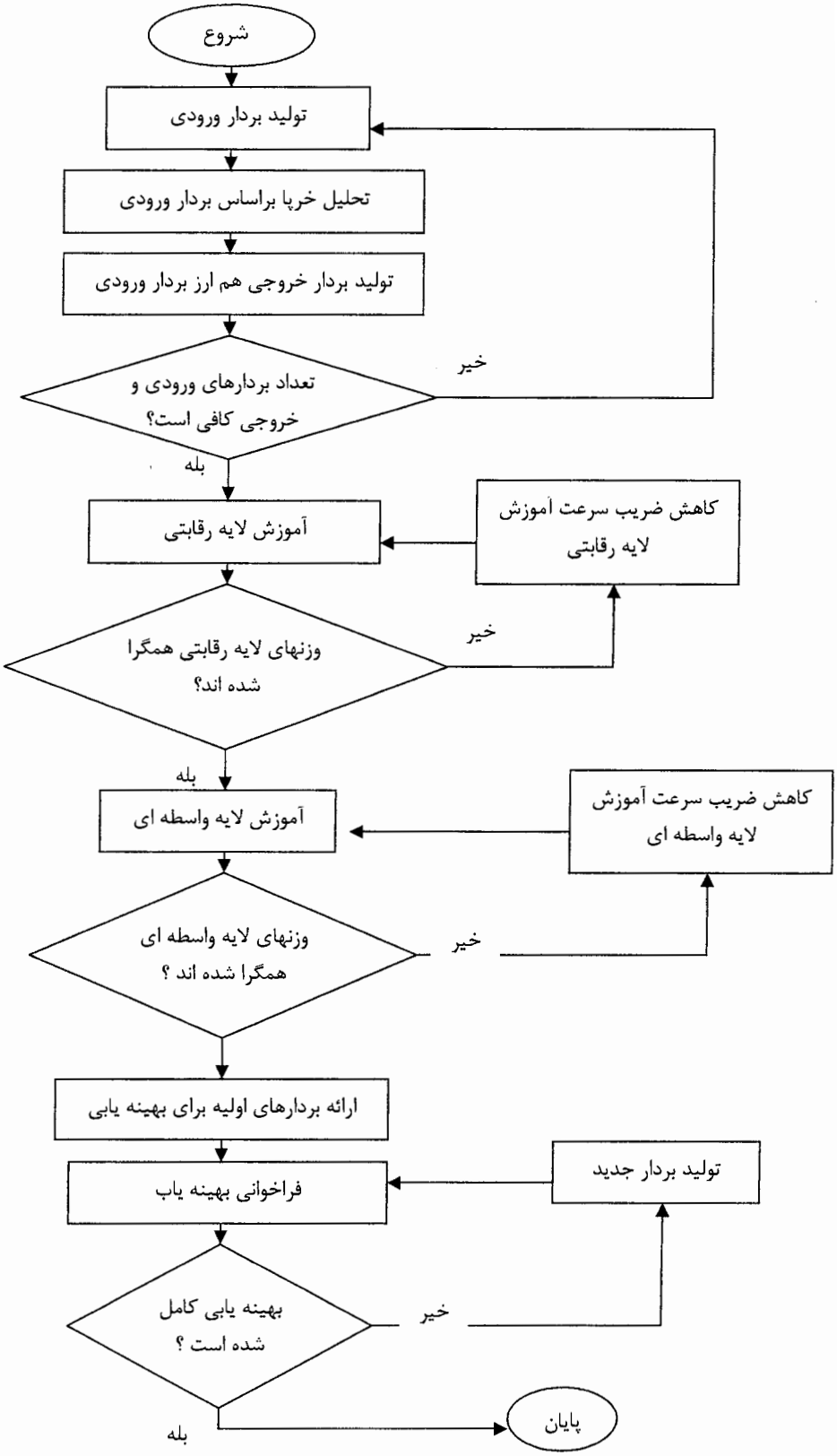
که در آن W وزن سازه ، γ وزن مخصوص مصالح سازه ، A_i سطح مقطع عضو iام و L_i طول عضو iام است . البته این کمینه سازی به شرطی باید انجام گردد که هیچ کدام از قیود مسئله نقض نگردد . قید های مربوط به کمینه سازی وزن خرپا در این تحقیق عبارتند از تجاوزنکردن تنش ها از یک مقدار ثابت و همچنین مختصات گره های متغیر باید در یک محدوده خاص باشد .

ابتدا با استفاده از روشهای مستقیم بهینه یابی خرپا ، خرپای مورد نظر را بهینه سازی می نمائیم که در این تحقیق از روش کمپلکس (فصل چهارم) جهت بهینه یابی دقیق استفاده شده

است . سپس توسط شبکه عصبی انتشار متقابل ، بهینه یابی خرپا انجام خواهد شد تا دقت این روش در مقایسه با روش کمپلکس (روش دقیق) تعیین شود .

برای استفاده ازشبکه عصبی انتشارمتقابل در بهینه یابی خرپا ، ابتدا این شبکه را برای تحلیل سازه آماده می کنیم . برای این منظور ابتدا بردار ورودی به شبکه عصبی انتشار متقابل شامل تمام متغیر های بهینه یابی خرپا از قبیل سطح مقطع اعضاء و مختصات نقاط گرهی و بردارخروجی شامل تنش درتمام اعضای خرپا تعیین می گردند سپس آموزش و تربیت شبکه جهت پاسخگویی ، آغاز می گردد . برای آموزش شبکه ابتدا باید زوج های آموزشی را به شبکه اعمال کنیم . تعداد این زوج ها برای هر مساله می تواند متفاوت باشد . تهیه و اعمال زوج های آموزشی به ابتکار طراح انجام می شود بعنوان مثال در مورد خرپاها ، سطح مقطع عضوها و مختصات نقاط گرهی ، در محدوده ای که استفاده کننده مشخص می کند به طور تصادفی یا تحت یک ضابطه خاص تولید می گردد و همزمان با این فرآیند ، تنش در اعضاء خرپا محاسبه می گردد . پس از تهیه زوج های آموزشی ، تربیت شبکه توسط الگوریتم آموزش شبکه عصبی انتشارمتقابل که در بخش (۳-۷) تشریح گردید ، انجام می گیرد .

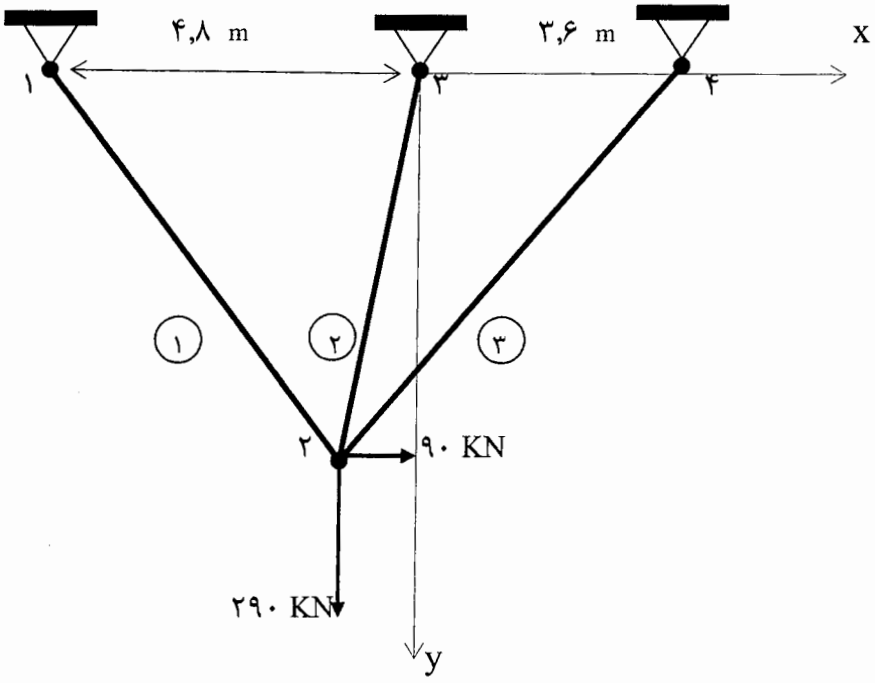
بطور خلاصه ساختار بهینه یابی خرپا توسط شبکه عصبی انتشار متقابل را می توان به صورت نشان داده شده در شکل (۵-۱) نمایش داد .



شکل (۵-۱) : ساختار کلی برنامه بهینه یابی خرپا با شبکه عصبی انتشار متقابل

۵-۳ بهینه یابی یک خرپای نا معین

در این قسمت یک خرپای نا معین سه عضوی مورد بررسی قرار می گیرد که جزئیات آن بهمراه بارهای وارده در شکل (۵-۲) نشان داده شده است . مکان هندسی سه گره منتهی به تکیه گاه ها ثابت و موقعیت گره ۲ محل اتصال سه عضو به یکدیگر ، متغیر می باشد البته محدوده تغییرات موقعیت گره ۲ در ادامه بیان خواهد شد .



شکل (۵-۲) : خرپای نا معین سه عضوی

با درنظرگرفتن سطح مقطع عضو ها و مختصات نقاط گرهی مشخص که قابل تغییرند ، بعنوان متغیر های مساله ، تابع هدف (وزن خرپا) را بصورت زیر می توان نوشت :

$$OBJ=\gamma\left[X_1\sqrt{(4.8+X_4)^2+X_5^2}+X_2\sqrt{X_4^2+X_5^2}+X_3\sqrt{(3.6-X_4)^2+X_5^2}\right] \qquad (۳-۵)$$

در این رابطه X_1 ، X_2 ، X_3 ، X_4 و X_5 بترتیب سطح مقطع عضوشماره یک ، دو ، سه و مختصات x و y گره ۲ هستند .

مقادیر X_1 ، X_2 و X_3 (سطح مقطع اعضاء) همگی بزرگتر یا مساوی صفر هستند و مقادیر X_4 و X_5 (موقعیت گره دوم) باید در محدوده ای که در زیر مشخص شده است واقع شوند و برای حل این مثال تنش مجاز (Mpa) ۱۳۰ در نظر گرفته شده است .

$$-2.0 \le X_4(m) \le 2.0 \quad \text{و} \quad 3.5 \le X_5(m) \le 5.5$$

بهینه سازی دقیق مساله تشریح شده در بالا با استفاده از روش کمپلکس انجام می شود که قبلا" در فصل چهارم بطور کامل این روش تشریح شده است و متن کامل برنامه تدارک دیده شده برای این منظور (بهینه یابی دقیق) در پیوست شماره یک آمده است که جواب دقیق برابر است با $OBJ = ۸۷۷۸\gamma$ (γ وزن مخصوص مصالح خرپاست) .

برای بهینه یابی خرپا با استفاده از شبکه عصبی انتشار متقابل ابتدا زوج بردارهای آموزشی را تعیین می کنیم . مقادیر X_1 ، X_2 ، X_3 ، X_4 و X_5 مولفه های بردار ورودی هستند که آنها را از روابطی که در ادامه اشاره خواهد شد ، بدست می آوریم . در ابتدا فضای جستجوی متغیر ها (مولفه های بردار ورودی) وسیع انتخاب می شود و در ادامه پس از چند مرحله سعی ، این فضا محدود به فضای جستجوی $D_4 = D^* \pm 0.4D^*$ محدود می شود بطوریکه D^* پاسخ دقیق متغیر های بهینه سازی است . در نهایت در فضای جستجوی D_4 ، متغیر ها از روابط زیر بدست می آیند .

$$X_1 = 1.9 + K * 0.025 \qquad (۴-۵)$$

$$X_2 = 11 + K * 0.15 \qquad (۵-۵)$$

$$X_3 = 0.03 + K * 0.004 \qquad (۶-۵)$$

$$X_4 = 55 + K * 0.72 \qquad (۷-۵)$$

$$X_5 = 350 + K * 1.4 \qquad (۸-۵)$$

K عدد صحیح است که از صفر تا N تغییر می کند بطوریکه N تعداد زوج های آموزشی که در نظر است تولید شود ، می باشد . بدین صورت که اشاره شد بردارهای ورودی آموزشی آماده خواهند شد . برای محاسبه مولفه های بردار خروجی ابتدا ماتریس سختی (متقارن) خرپای شکل (۵-۲) را تشکیل می دهیم این ماتریس بصورت ماتریس زیر خواهد بود :

$$[K] = \begin{bmatrix} k_{11} & & & & & & & \\ k_{21} & k_{22} & & & & sym & & \\ k_{31} & k_{32} & k_{33} & & & & & \\ k_{41} & k_{42} & k_{43} & k_{44} & & & & \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & & & \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} & & \\ k_{71} & k_{72} & k_{73} & k_{74} & k_{75} & k_{76} & k_{77} & \\ k_{81} & k_{82} & k_{83} & k_{84} & k_{85} & k_{86} & k_{87} & k_{88} \end{bmatrix} \qquad (۹-۵)$$

و با تعریف

$$a = X_1 \Big/ ((4.8 + X_4)^2 + X_5{}^2)^{3/2} \quad ،$$

$$b = X_2 \Big/ ((X_4{}^2 + X_5{}^2)^{3/2} \quad \text{و} \quad c = X_3 \Big/ ((3.6 - X_4)^2 + X_5{}^2)^{3/2}$$

مقادیر مولفه های ماتریس سختی برابر خواهند بود با :

$$k_{11} = (4.8 + X_4)^2 * a$$

$$k_{21} = -X_5 * (4.8 + X_4)^2 * a$$

$$k_{22} = X_5{}^2 * a$$

$$k_{31} = -k_{11} \quad ، \quad k_{32} = -k_{11}$$

$$k_{33} = k_{11} + X_4{}^2 * b + (3.6 - X_4)^2 * c$$

$$k_{41} = k_{21} \quad ، \quad k_{42} = -k_{22}$$

$$k_{43} = k_{21} - X_5 * X_4 * b$$

$$k_{44} = X_5{}^2 * (a + b + c)$$

$$k_{51} = k_{52} = 0$$

$$k_{53} = -X_4{}^2 * b$$

$$k_{54} = X_5 * X_4 * b$$

$$k_{61} = k_{62} = 0 \quad ، \quad k_{55} = -k_{53}$$

$$k_{64} = -X_5{}^2 * b \quad ، \quad k_{63} = k_{54}$$

$$k_{65} = -k_{54} \quad ، \quad k_{66} = -k_{64}$$

$$k_{73} = -(3.6 - X_4)^2 * c \quad ، \quad k_{71} = k_{72} = 0$$

$$k_{74} = -X_5 * (3.6 - X_4) * c \quad ، \quad k_{75} = k_{76} = 0$$

$$k_{81} = k_{82} = k_{85} = k_{86} = 0 \quad ، \quad k_{77} = -k_{73}$$

$$k_{84} = -X_5{}^2 * c \quad ، \quad k_{83} = k_{74}$$

$$k_{88} = -k_{84} \quad و \quad k_{87} = -k_{83} \quad .$$

پس از بدست آوردن ماتریس سختی هم ارز بردار ورودی ، مقدار نیروهای داخلی اعضاء از حل دستگاه معادلات زیر محاسبه می شود :

$$[K] * \{\Delta\} = \{F\} \qquad (۱۰-۵)$$

که $[K]$ ماتریس سختی معادل بردار ورودی است که در بالا تشریح گردید و $\{\Delta\}$ بردار تغییر مکان بوده و به صورت زیر تعریف می شود (رابطه (۵-۱۱)) :

$$\{\Delta\}=\begin{Bmatrix}u_1\\v_1\\u_2\\v_2\\u_3\\v_3\\u_4\\v_4\end{Bmatrix}=\begin{Bmatrix}0\\0\\u_2\\v_2\\0\\0\\0\\0\end{Bmatrix} \qquad (۱۱-۵)$$

$\{F\}$ بردار نیروهای وارده به خرپا است که بصورت زیر است :

$$\{F\}=\begin{Bmatrix}f_{1x}\\f_{1y}\\f_{2x}\\f_{2y}\\f_{3x}\\f_{3y}\\f_{4x}\\f_{4y}\end{Bmatrix}=\begin{Bmatrix}f_{1x}\\f_{1y}\\90000\\225000\\f_{3x}\\f_{3y}\\f_{4x}\\f_{4y}\end{Bmatrix} \qquad (۱۲-۵)$$

که در آن f_{ix} نیروی وارده به گره i در امتداد x و f_{iy} نیروی وارده به گره i در امتداد y است . همانگونه که تشریح شد پس از حل دستگاه معادلات ، مقدار نیروهای داخلی اعضاء خرپا بدست می آید که از تقسیم نیروی داخلی اعضاء به سطح مقطع عضو مقدار تنش یا همان مولفه بردار خروجی حاصل می شود که بصورت زیر است (مجددا" یادآوری می شود که این بردار خروجی ، متناظر بردار ورودی می باشد)

$$\sigma_1 = E.a/X_1 .\sqrt{\left[-(4.8+X_4)^2.u_2+X_5.(4.8+X_4)^2.v_2\right]^2+\left[X_5(4.8+X_4).u_2-X_5^2.v_2\right]^2}$$

$$\sigma_2 = E.b/X_2 .\sqrt{\left[-X_4^2.u_2 + X_4.X_5.v_2\right]^2 + \left[X_4.X_5.u_2 - X_5^2.v_2\right]^2}$$

$$\sigma_3 = E.c/X_3 .\sqrt{\left[(3.6-X_4)^2.u_2 + X_5.(3.6-X_4).v_2\right]^2 + \left[X_5.(3.6-X_4).u_2 + X_5^2.v_2\right]^2}$$

σ_1 ، σ_2 و σ_3 مولفه های بردار خروجی اند که بترتیب عبارتند از تنش در عضو اول ، دوم و سوم می باشند . یادآوری می شود که تمامی این محاسبات پس از معرفی سازه و فضای جستجو به برنامه بهینه یابی توسط شبکه عصبی ، خودبخود محاسبه می شوند (متن کامل برنامه در پیوست شماره دو آمده است) . تعداد ۱۰۰ زوج آموزشی برای مثال تحت بررسی تدارک دیده شده است که این زوج ها جهت تربیت شبکه و در نهایت برای بهینه یابی مورد استفاده قرار خواهند گرفت .

قبل از بهینه یابی ، پاسخ شبکه عصبی را برای بردار های ورودی که قبلا" با آنها مواجه نبوده را مورد ارزیابی قرار می دهیم که نتایج کلی آن در جداول (۵-۱) ، (۵-۲) ، (۵-۳) و (۵-۴) آمده است و همچنین بهینه یابی خرپا با استفاده از شبکه عصبی انتشار متقابل در چهار فضای جستجوی متفاوت $D_1 = D^* \pm 0.1D^*$ (D^* پاسخ دقیق متغیر ها ی خرپا می باشد) ، $D_2 = D^* \pm 0.2D^*$ ، $D_3 = D^* \pm 0.3D^*$ و $D_4 = D^* \pm 0.4D^*$ انجام شده است که نتایج کلی آنها در جدول (۵-۵) آمده است .

جدول (۵-۱) : پاسخ شبکه نسبت به اولین بردار ورودی که شبکه قبلا" با آن مواجه نبوده است .

متغیر های بردار ورودی	مولفه های اولین بردار ورودی	مولفه های بردارخروجی(دقیق) N/cm^2	مولفه های بردار خروجی (پاسخ شبکه) N/cm^2	خطا (درصد) E٪
$X_1(cm^2)$	۳	۱۴۲۳۰	۱۶۲۵۴	۱۴,۲
$X_2(cm^2)$	۱۸	۱۱۶۵۰	۱۱۵۹۸	۰,۴۴
$X_3(cm^2)$	۰,۰۵	-۱۰۴۱۸	-۱۳۱۳۸	۲۶,۱
$X_4(cm)$	۹۵	-----	-----	-----
$X_5(cm)$	۳۶۰	-----	-----	-----
				[۱] $\overline{E} = ۱۳,۶٪$

جدول (۵-۲) : پاسخ شبکه نسبت به دومین بردار ورودی که شبکه قبلا" با آن مواجه نبوده است .

متغیر های بردار ورودی	مولفه های دومین بردار ورودی	مولفه های بردارخروجی(دقیق) N/cm^2	مولفه های بردار خروجی (پاسخ شبکه) N/cm^2	خطا (درصد) E٪
$X_1(cm^2)$	۲,۹	۱۶۴۷۶	۱۶۵۲۶	۰,۳
$X_2(cm^2)$	۱۷	۱۲۱۱۷	۱۱۷۱۰	۳,۳۵
$X_3(cm^2)$	۰,۰۵	-۱۳۲۸۲	-۱۳۵۱۵	۱,۷۵
$X_4(cm)$	۸۸	------	------	-----
$X_5(cm)$	۳۵۸	-----	-----	-----
				$\overline{E} = ۱,۸٪$

[۱] - $\overline{E}$ ؛ میانگین حسابی خطا ها می باشد .

جدول (۵-۳) : پاسخ شبکه نسبت به سومین بردار ورودی که شبکه قبلا" با آن مواجه نبوده است

متغیر های بردار ورودی	مولفه های چهارمین بردار ورودی	مولفه های بردارخروجی(دقیق) N/cm^2	مولفه های بردار خروجی (پاسخ شبکه) N/cm^2	خطا (درصد) ٪ E
$X_1(cm^2)$	۳,۲	۱۲۹۰۰	۱۴۲۶۲	۱۰,۵۵
$X_2(cm^2)$	۲۰	۱۰۵۲۵	۱۰۷۵۳	۲,۱۶
$X_3(cm^2)$	۰,۰۵۵	-۹۱۳۵	-۱۰۳۵۷	۱۳,۴
$X_4(cm)$	۱۰۰	-----	-----	-----
$X_5(cm)$	۳۷۰	-----	-----	-----
				$\overline{E}=۸,۷٪$

جدول (۵-۴) : پاسخ شبکه نسبت به چهارمین بردار ورودی که شبکه قبلا" با آن مواجه نبوده است .

متغیر های بردار ورودی	مولفه های سومین بردار ورودی	مولفه های بردارخروجی(دقیق) N/cm^2	مولفه های بردار خروجی (پاسخ شبکه) N/cm^2	خطا (درصد) E٪
$X_1(cm^2)$	۲,۸	۱۶۷۷۷	۱۶۶۳۷	۰,۸۳
$X_2(cm^2)$	۱۹	۱۰۸۸۰	۱۱۷۵۶	۸,۰۵
$X_3(cm^2)$	۰,۰۴۵	-۱۵۴۷۳	-۱۳۶۶۹	۱۱,۶۶
$X_4(cm)$	۸۸	-----	-----	-----
$X_5(cm)$	۳۵۵	-----	-----	-----
				$\overline{E}=۶,۸٪$

جدول (۵-۵) : نتایج کلی بهینه یابی خرپای مثال اول

بهینه سازی خرپای مثال اول		فضای جستجوی D_1		فضای جستجوی D_2		فضای جستجوی D_3		فضای جستجوی D_4	
متغیرهای طرح	پاسخ دقیق[1]	پاسخ شبکه	خطا E٪	پاسخ شبکه	خطا E٪	پاسخ شبکه	خطا E٪	پاسخ شبکه	خطا E٪
$X_1(cm^2)$	۳,۱۶	۳,۳۸	۷	۳,۷	۱۷	۳,۳۳	۵,۴	۳,۹۹	۲۶
$X_2(cm^2)$	۱۸,۲۵	۱۷,۱۶	۶	۱۸,۰۱	۱,۳	۱۸,۵۱	۱,۴	۱۷,۸	۲,۴
$X_3(cm^2)$	۰,۰۵	۰,۰۵	۰	۰,۰۵۹	۱۸	۰,۰۶	۲۰	۰,۰۵	۰
$X_4(cm)$	۹۰,۱	۸۸,۷۴	۱,۵	۸۴,۴	۶,۳	۸۹,۲۷	۰,۹	۸۰,۱۲	۱۱
$X_5(cm)$	۳۵۲,۲	۳۵۵,۲	۰,۸	۳۵۵,۵	۰,۹۳	۳۵۶,۶	۱,۲	۳۵۰,۴	۰,۵
OBJ	۸۷۷۸ γ	۹۱۶۹ γ	۴,۵	۹۷۵۱ γ	۱۱,۰۸	۹۶۴۷ γ	۹,۹	۹۷۶۱ γ	۱۱,۲

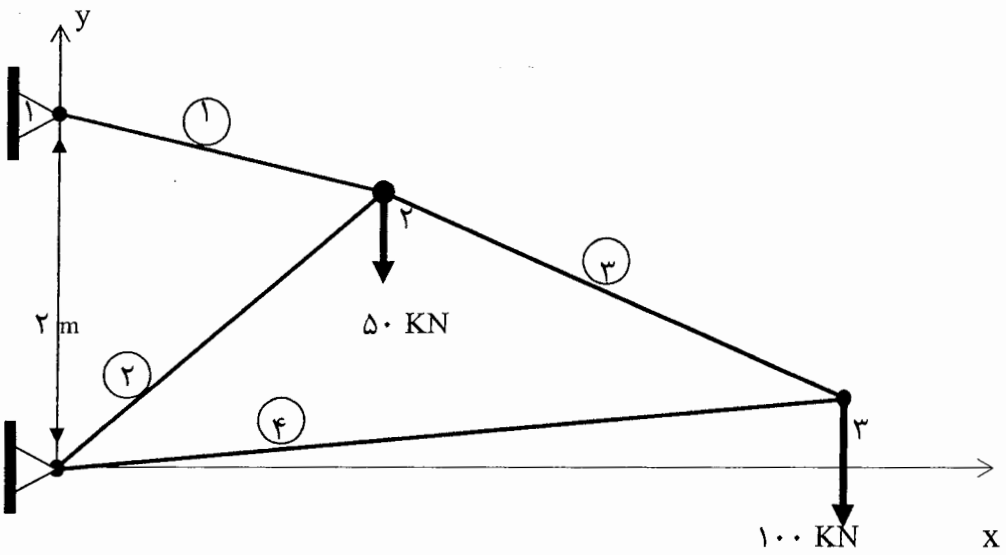
[1] - پاسخ دقیق از روش کمپلکس (بهینه سازی مستقیم) بدست آمده است .

از نتایج مربوط به بهینه سازی خرپای مثال اول پیداست که با تغییر دادن فضای جستجوی متغیرها ، دقت شبکه تا حدودی تغییر می یابد . تغییرات بوجود آمده به این دلیل است که با تغییر دادن فضای جستجو برای شبکه ، تصمیم گیری شبکه عصبی بر اساس زوج های آموزشی جدیدی که در فضای جستجوی متفاوت تدارک دیده شده اند ، صورت می گیرد و با توجه به کوچکی و بزرگی دامنه جستجو دقت شبکه عصبی تحت تاثیر قرار خواهد گرفت . این نکته کلی را باید در نظر داشت که اگر فضای جستجوی متغیر ها گسترش یابد ، یعنی متغیر ها در یک بازه بزرگتر تغییر کنند ، دقت شبکه های عصبی کاهش خواهد یافت و درنتیجه پاسخ حاصل از بهینه سازی با خطای بیشتری همراه خواهد بود پس برای بهبود نتایج بهینه سازی بهتر است که بازه جستجو را کوچکتر انتخاب کرد .

در حالت کلی برای بهبود نتایج شبکه عصبی ، ابتدا فضای جستجو را می توان بزرگ در نظر گرفت و پس از یافتن پاسخ بهینه توسط شبکه عصبی ، مجددا" در محدوده جواب های بدست آمده اما با انتخاب فضای جستجوی کوچکتر ، مساله را بهینه سازی کرده و به جواب های با دقت بیشتری رسید . متاسفانه نمی توان معین کرد که آیا پاسخ بهینه سازی بر اساس شبکه های عصبی بزرگتر یا کوچکتر از جواب واقعی هست یا خیر ؟ .

۵-۴ بهینه یابی یک خرپای معین

درمثال دوم یک خرپای چهار عضوی که درشکل (۵-۳) نشان داده شده است تحت بررسی قرار می گیرد . نیرو های اعمالی و موقعیت عضو ها در شکل (۵-۳) نمایش داده شده است .



شکل (۵-۳) : شکل کلی خرپا با در نظر گرفتن موقعیت متغیر برای گره دوم و سوم

برای حل این مثال تنش مجاز (Mpa) ۱۳۰ در نظر گرفته شده است و مکان هندسی تکیه گاه ها ثابت بوده و موقعیت گره دوم و سوم در خرپا متغیر بوده و در یک محدوده مشخص شده تغییر می کند .

با انتخاب سطح مقطع اعضاء و مختصات نقاط گرهی که قابل تغییراند ، بعنوان متغیر های مساله ، تابع هدف بصورت رابطه (۵-۱۳) در خواهد آمد :

$$OBJ = \gamma.\Big[X_1.\sqrt{X_5{}^2 + (2 - X_6{}^2)} + X_2.\sqrt{X_5{}^2 + X_6{}^2} + X_3.\sqrt{(X_7 - X_5)^2 + (X_6 - X_8)^2} + X_4.\sqrt{X_7^2 + X_8^2}\ \Big] \qquad (۱۳-۵)$$

در رابطه (۵-۱۳) X_1 ، X_2 ، X_3 ، X_4 ، X_5 ، X_6 ، X_7 و X_8 بترتیب سطح مقطع عضو شماره یک ، دو ، سه ، چهار (همگی بزرگتر یا مساوی صفراند) و مختصات x وy گره دوم و مختصات x و y گره سوم می باشند . قیدهای هندسی مربوط به موقعیت گره های دوم و سوم در رابطه (۵-۱۴) مشخص شده است .

$$\begin{Bmatrix} 80 \\ 50 \\ 150 \\ -100 \end{Bmatrix} \le \begin{Bmatrix} X_5 = x_2(cm) \\ X_6 = y_2(cm) \\ X_7 = x_3(cm) \\ X_8 = y_3(cm) \end{Bmatrix} \le \begin{Bmatrix} 250 \\ 200 \\ 350 \\ 100 \end{Bmatrix} \qquad (۱۴-۵)$$

همانند مثال قبل برای بهینه یابی خرپا با استفاده از شبکه عصبی انتشار متقابل ، ابتدا ۱۰۰ زوج آموزشی را جهت تربیت شبکه آماده می کنیم . برای تدارک بردار های ورودی آموزشی در فضای جستجوی $D_4 = D^* \pm 0.4D^*$ (D^* پاسخ دقیق متغیر ها ی خرپا می باشد) ، از روابطی که در ادامه می آید ، سود خواهیم برد .

$$X_1 = 10.82 + K * 0.15 \qquad (۱۵-۵)$$

$$X_2 = 4.152 + K * 0.056 \qquad (۱۶-۵)$$

$$X_3 = 9.4 + K * 0.125 \qquad (۱۷-۵)$$

$$X_4 = 3.03 + K * 0.041 \qquad (۱۸-۵)$$

$$X_5 = 80.0 + K * 1.086 \qquad (۱۹-۵)$$

$$X_6 = 56.0 + K * 0.75 \qquad (۲۰-۵)$$

$$X_7 = 150.0 + K * 0.6 \qquad (۲۱-۵)$$

$$X_8 = 45.0 + K * 0.55 \qquad (۲۲-۵)$$

مقدار K عدد صحیح بوده واز صفر تا صد (تعداد زوجهای آموزشی) تغییر می کند . بدین طریق بردار های ورودی آموزشی آماده اند سپس با استفاده از روابط سختی مربوط به خرپا ، بازای هر بردار ورودی ، بردار خروجی (مولفه های آن تنش در اعضاء خرپا می باشد) متناظر بدست می آید این کار خود بخود توسط برنامه (پیوست شماره سه) انجام می شود .

پس از مشخص شدن متغیر ها و پارامتر های دیگر مساله بهینه یابی ، این مساله را به برنامه معرفی می کنیم تا بهینه سازی را در چهار فضای جستجوی $D_1 = D^* \pm 0.1D^*$ (D^* پاسخ دقیق متغیرهای خرپا میباشد) ، $D_2 = D^* \pm 0.2D^*$ ، $D_3 = D^* \pm 0.3D^*$ و $D_4 = D^* \pm 0.4D^*$ انجام دهد .

همانند مثال قبل ، پاسخ شبکه عصبی را در برابر بردارهای ورودی که شبکه قبلا" با آنها مواجه نبوده را مورد ارزیابی قرار می دهیم که نتایج آن در جداول (۵-۶) ، (۵-۷) ، (۵-۸) و (۵-۹) خلاصه شده است و همچنین بهینه یابی خرپا با استفاده از شبکه عصبی انتشار متقابل در چهار فضای جستجوی متفاوت $D_1 = D^* \pm 0.1D^*$ (D^* پاسخ دقیق متغیر ها ی خرپا می‌باشد) ، $D_2 = D^* \pm 0.2D^*$ ، $D_3 = D^* \pm 0.3D^*$ و $D_4 = D^* \pm 0.4D^*$ انجام شده است که نتایج کلی آنها در جدول (۵-۱۰) آمده است (متن کامل برنامه بهینه یابی برای این مثال در پیوست شماره سه آمده است) .

جدول (۵-۶) : پاسخ شبکه نسبت به اولین بردار ورودی که شبکه قبلا" با آن مواجه نبوده است .

متغیر های بردار ورودی	مولفه های دومین بردار ورودی	مولفه های بردارخروجی(دقیق) N/cm^2	مولفه های بردار خروجی (پاسخ شبکه) N/cm^2	خطا (درصد) E٪
$X_1(cm^2)$	۱۸,۵	۷۷۸۸,۹	۸۱۶۰	۴,۷
$X_2(cm^2)$	۶,۵	-۳۳۱۹,۱	-۵۲۸۰	۵۹
$X_3(cm^2)$	۱۷	۶۳۲۹,۱	۶۷۰۰	۵,۸
$X_4(cm^2)$	۵	-۲۰۴۹۹	-۱۸۳۵۰	۱۰,۴
$X_5(cm)$	۱۳۰	-----	-----	-----
$X_6(cm)$	۹۰	-----	-----	-----
$X_7(cm)$	۱۵۵	-----	-----	-----
$X_8(cm)$	۷۵	-----	-----	-----
				$\overline{E}=۲۰٪$

جدول (۵-۷) : پاسخ شبکه نسبت به دومین بردار ورودی که شبکه قبلا" با آن مواجه نبوده است .

متغیر های بردار ورودی	مولفه های دومین بردار ورودی	مولفه های بردارخروجی(دقیق) N/cm^2	مولفه های بردار خروجی (پاسخ شبکه) N/cm^2	خطا (درصد) E٪
$X_1(cm^2)$	۱۶,۲	۸۵۰۴	۷۸۵۱	۷,۷
$X_2(cm^2)$	۶,۲	-۸۵۸۵	-۶۷۰۶	۲۲
$X_3(cm^2)$	۱۴	۶۵۸۷	۶۲۳۸	۵,۳
$X_4(cm^2)$	۴,۵	-۱۶۴۲۵	-۱۶۲۸۳	۰,۸۶
$X_5(cm)$	۱۳۵	-----	-----	-----
$X_6(cm)$	۹۵	-----	-----	-----
$X_7(cm)$	۱۵۰	-----	-----	-----
$X_8(cm)$	۸۰	-----	-----	-----
				$\overline{E}=۸,۹٪$

جدول (۵-۸) : پاسخ شبکه نسبت به سومین بردار ورودی که شبکه قبلا" با آن مواجه نبوده است .

متغیر های بردار ورودی	مولفه های سومین بردار ورودی	مولفه های بردارخروجی(دقیق) N/cm^2	مولفه های بردار خروجی (پاسخ شبکه) N/cm^2	خطا (درصد) E٪
$X_1(cm^2)$	۱۹,۸	۷۳۳۵	۸۸۹۱	۲۱,۲
$X_2(cm^2)$	۷,۶	-۱۸۶۰	-۲۱۰۰	۱۳
$X_3(cm^2)$	۱۷	۶۴۹۰	۷۸۴۵	۲۰,۸
$X_4(cm^2)$	۵	-۲۰۵۲۲	-۲۲۹۱۰	۱۱,۶
$X_5(cm)$	۱۲۱	-----	-----	-----
$X_6(cm)$	۸۵	-----	-----	-----
$X_7(cm)$	۱۵۰	-----	-----	-----
$X_8(cm)$	۶۷	-----	-----	-----
				$\overline{E} = ۱۶,۶٪$

جدول (۵-۹) : پاسخ شبکه نسبت به چهارمین بردار ورودی که شبکه قبلا" با آن مواجه نبوده است .

متغیر های بردار ورودی	مولفه های چهارمین بردار ورودی	مولفه های بردارخروجی(دقیق) N/cm^2	مولفه های بردار خروجی (پاسخ شبکه) N/cm^2	خطا (درصد) E٪
$X_1(cm^2)$	۱۹	۷۳۷۲	۸۱۶۰	۱۰,۷
$X_2(cm^2)$	۶,۸	-۴۸۴۴	-۵۲۸۰,۳	۹
$X_3(cm^2)$	۱۷,۲	۵۸۶۲	۶۷۰۰	۱۲,۵
$X_4(cm^2)$	۵	-۱۸۰۷۰,۷	-۱۸۳۴۰	۱,۵
$X_5(cm)$	۱۲۵	-----	-----	-----
$X_6(cm)$	۹۵	-----	-----	-----
$X_7(cm)$	۱۵۲	-----	-----	-----
$X_8(cm)$	۷۵	-----	-----	-----
				$\overline{E} = ۸,۴٪$

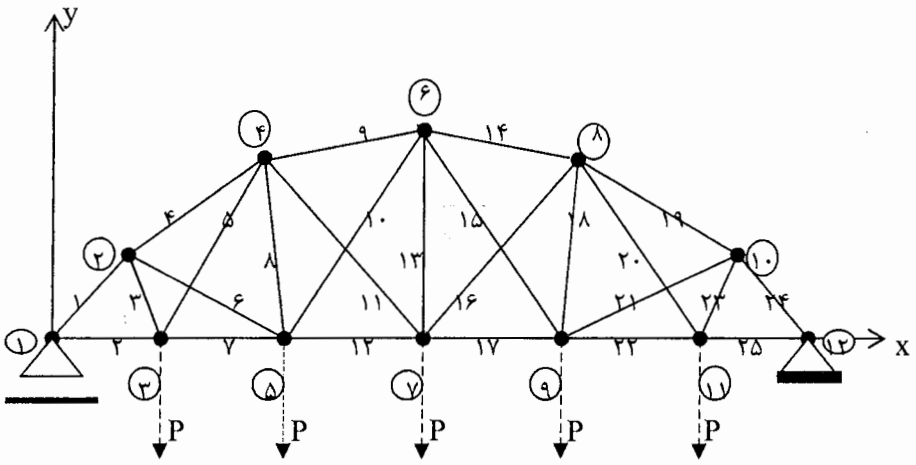
جدول (۵-۱۰) : نتایج کلی بهینه یابی خرپای مثال دوم

بهینه سازی خرپای مثال دوم		فضای جستجوی D_1		فضای جستجوی D_2		فضای جستجوی D_3		فضای جستجوی D_4	
متغیرهای طرح	پاسخ دقیق[1]	پاسخ شبکه	خطا E٪	پاسخ شبکه	خطا E٪	پاسخ شبکه	خطا E٪	پاسخ شبکه	خطا E٪
$X_1(cm^2)$	۱۸,۰۲	۱۸,۸	۴,۳	۱۹,۰۸	۵,۹	۱۸,۵۸	۳,۱	۱۹,۱۱	۶
$X_2(cm^2)$	۶,۹	۶,۶	۴,۳	۶,۸۶	۰,۶	۷,۳۹	۷,۱	۷,۶۵	۱۰
$X_3(cm^2)$	۱۵,۶۴	۱۵,۸	۱,۱	۱۶,۲۸	۴,۱	۱۶,۴۴	۵,۱	۱۵,۹۴	۱,۹
$X_4(cm^2)$	۵,۰۵	۵,۳۲	۵,۳	۵,۳۶	۶,۱	۵,۴۶	۸,۱	۵,۵	۸,۹
$X_5(cm)$	۱۳۴,۷۵	۱۲۸,۷	۴,۵	۱۱۹,۶	۱۱,۲	۱۱۱,۵	۱۷,۲	۱۰۹,۳۶	۱۹
$X_6(cm)$	۹۳,۵	۹۷,۳	۴,۱	۹۲,۳	۱,۳	۸۷,۵۷	۶,۳	۹۷	۳,۷
$X_7(cm)$	۱۵۰,۳	۱۵۶,۵	۴,۱	۱۵۴	۲,۵	۱۵۳,۵۴	۲,۱۵	۱۵۳,۷	۲,۳
$X_8(cm)$	۷۵,۰۴	۷۹,۳	۵,۷	۷۲,۶	۳,۲	۶۳,۲۱	۱۵,۷	۶۲,۵۴	۱۷
OBJ	۵۴۵۶ γ	۵۶۰۶ γ	۲,۷	۵۶۲۶ γ	۳,۱	۵۷۵۹ γ	۵,۵۵	۵۸۱۵ γ	۶,۵

[1] . پاسخ دقیق از روش کمپلکس (بهینه سازی مستقیم) بدست می آید .

۵-۵ بهینه یابی خرپای متقارن ۲۵ عضوی

در این مثال یک خرپای ۲۵ عضوی که تحت اثر بارهای قائم قرار دارد در شکل (۵-۴) مشاهده



شکل (۵-۴) : خرپای ۲۵ عضوی

می شود ، مورد بررسی قرار می‌گیرد . در شکل شماره گره ها در داخل دایره و شماره اعضاء روی آنها ذکر شده است . در این مثال موقعیت گره های ۱ ، ۳ ، ۵ ، ۷ ، ۹ ، ۱۱ و ۱۲ ثابت بوده و بفاصله یکسان و برابر ۵۰۰ سانتیمتر از هم قرار گرفته اند و موقعیت گره های ۲ ، ۴ ، ۶ ، ۸ و ۱۰ متغیر بوده و البته مکان هندسی این نقاط در یک فاصله معین تغییر می کند که در ادامه بیان خواهد شد . بارهای وارده (P) به گره های ۳ ، ۵ ، ۷ ، ۹ و ۱۱ که در شکل (۵-۴) نشان داده شده اند که مقدار آنها باهم یکسان و برابر ۳۰۰۰ KN می باشد .

تابع هدف در این مثال وزن کل خرپاست که بصورت رابطه (۵-۲۳) بیان شده است :

$$OBJ = \sum_{1}^{25} \gamma . A_i . L_i \qquad (۲۳-۵)$$

که در آن OBJ وزن سازه (تابع هدف) ، γ وزن مخصوص مصالح سازه ، A_i سطح مقطع عضو iام و L_i طول عضو iام است .

قیدهای حاکم بر این مثال دو نوع می باشند یکی قیدهای مربوط به تنش در اعضاء و دیگری قیدهای هندسی مر بوط به گره های متغیر می باشند که در زیر به آنها اشاره شده است .

قید تنش در اعضاء خر پا در رابطه (۵-۲۴) بیان شده است .

$$-130 \le \sigma_i(Mpa) \le 130 \qquad i = ۱،۲، \ldots ، ۲۵ \qquad (i \text{ شماره عضو می باشد}) \qquad (۵-۲۴)$$

وقید های هندسی گره های متغیر نیز به صورت زیر می باشند :

$$\begin{Bmatrix} 250 \\ 50 \\ 650 \\ 200 \\ 400 \end{Bmatrix} \le \begin{Bmatrix} x_2(cm) \\ y_2(cm) \\ x_4(cm) \\ y_4(cm) \\ y_6(cm) \end{Bmatrix} \le \begin{Bmatrix} 750 \\ 300 \\ 1200 \\ 750 \\ 750 \end{Bmatrix} \qquad (۵-۲۵)$$

در رابطه (۵-۲۵) مقادیر x_2 و y_2 مختصات گره ۲ و x_4 و y_4 مختصات گره ۴ بوده و y_6 ارتفاع گره ۶ می باشد . البته با توجه به تقارن خرپا محدودیت های هندسی گره های ۸ و ۱۰ تابعی از محدودیت های گره های ۲ و ۴ می باشند یعنی :

$$\begin{Bmatrix} 2250 \\ 50 \\ 1800 \\ 200 \end{Bmatrix} \le \begin{Bmatrix} x_{10}(cm) = 3000 - x_2 \\ y_{10}(cm) = y_2 \\ x_8(cm) = 3000 - x_4 \\ y_8(cm) = y_4 \end{Bmatrix} \le \begin{Bmatrix} 2750 \\ 300 \\ 2350 \\ 750 \end{Bmatrix} \qquad (۵-۲۶)$$

در ضمن با ید سطح مقطع اعضاء بزرگتر مساوی صفر باشند یعنی :

$$0 \le A_i \qquad i = ۱،۲ ، \ldots ، ۲۵ \qquad (۵-۲۷)$$

.

پس از مشخص شدن اطلاعات اولیه بهینه سازی خرپا ، ابتدا بهینه سازی خرپا را با استفاده از روش دقیق انجام می دهیم سپس مساله را به برنامه بهینه یابی توسط شبکه عصبی ، معرفی کرده تا بهینه سازی را در فضای جستجوی $D_1 = D^* \pm 0.1D^*$ و $D_3 = D^* \pm 0.3D^*$

انجام دهد . نتایج کلی بهینه سازی دقیق ، تقریبی و پاسخ شبکه عصبی در برابر بردارهای ورودی که شبکه قبلا" با آنها مواجه نبوده در جداول (۵-۱۱) و (۵-۱۲) آمده است .

جدول (۵-۱۱) : نتایج کلی خرپای ۲۵ عضوی

بهینه سازی خرپای مثال آخر		فضای جستجوی D_1		فضای جستجوی D_3	
متغیر ها	پاسخ دقیق	پاسخ شبکه	خطا E٪	پاسخ شبکه	خطا E٪
$X_1(cm^2)$	۲۰۱۵,۲	۲۴۰۰	۱۹,۱	۲۵۰۳,۳	۲۴,۲
$X_2(cm^2)$	۱۸۲۰,۳	۲۰۵۰	۱۲,۶	۲۱۵۲,۵	۱۸,۲
$X_3(cm^2)$	۱۶۳,۴	۱۵۰	۸,۲	۱۶۱,۳	۱,۳
$X_4(cm^2)$	۱۵۱۲,۵	۱۶۰۰	۵,۸	۱۸۰۴	۱۹,۳
$X_5(cm^2)$	۳۵۸	۳۵۰	۲,۲	۳۵۱,۴	۱,۸۴
$X_6(cm^2)$	۷,۳	۵,۰۱	۳۱,۳	۵,۳۴	۲۶,۸
$X_7(cm^2)$	۱۵۰۸,۶	۱۶۰۰	۶,۰۶	۱۸۹۱,۴	۲۵,۴
$X_8(cm^2)$	۵۶۴	۴۵۰	۲۰,۱	۴۵۶,۷	۱۹
$X_9(cm^2)$	۱۳۱۹,۷	۱۵۰۰	۱۳,۷	۱۶۰۲,۵	۲۱,۴
$X_{10}(cm^2)$	۳۱۲,۲	۲۲۰	۲۹,۵	۲۲۱,۳	۲۹,۱
$X_{11}(cm^2)$	۸۴,۴	۶۰	۲۸,۹	۶۱,۳۴	۲۷,۳
$X_{12}(cm^2)$	۱۱۰۸,۳	۸۵۰	۲۳,۳	۸۵۳,۴	۲۳
$X_{13}(cm^2)$	۱۹۷,۶	۱۳۸	۳۰,۱	۱۴۰,۵۲	۲۸,۸
$x_2(cm)$	۲۶۵,۲	۲۵۱	۵,۳۵	۲۵۲,۰۲	۴,۹۷
$y_2(cm)$	۸۵,۱	۶۰	۲۹,۵	۶۱,۲	۲۸,۱
$x_4(cm)$	۷۱۴,۲	۶۵۰	۸,۹	۶۵۳,۴	۸,۵
$y_4(cm)$	۲۶۳,۲	۲۰۰	۲۴	۲۰۲,۹	۲۳
$y_6(cm)$	۴۲۸,۱	۴۰۰	۶,۵۶	۴۰۳,۵۳	۵,۷۴
OBJ	$۱,۰۵۵\times۱۰^{۷}\gamma$	$۱,۰۸\times۱۰^{۷}\gamma$	۲,۳۷	$۱,۱۶\times۱۰^{۷}\gamma$	۹,۹۵

جدول (۵-۱۲) : پاسخ شبکه نسبت به بردار ورودی که شبکه قبلا" با آن مواجه نبوده است .

متغیر های بردار ورودی	مولفه های بردار ورودی	مولفه های بردارخروجی(دقیق) N/cm^2	مولفه های بردار خروجی (پاسخ شبکه) N/cm^2	خطا (درصد) E٪
$X_1(cm^2)$	۲۱۲۰	-۱۱۰۴۱	-۱۲۲۱۲	۱۰,۶
$X_2(cm^2)$	۱۸۲۵	۱۳۰۵۴	۱۲۸۹۵	۱,۲۲
$X_3(cm^2)$	۱۶۵	-۱۷۷۴۸	-۱۶۲۱۴	۸,۶۵
$X_4(cm^2)$	۱۵۱۵	-۱۴۹۹۳	-۱۴۸۹۲	۰,۶۷
$X_5(cm^2)$	۳۶۰	۱۴۳۶۲	۱۴۳۳۴	۰,۲
$X_6(cm^2)$	۷,۵	-۲۶۲۰	-۲۵۳۸	۳,۱۳
$X_7(cm^2)$	۱۵۱۰	۱۱۸۰۰	۱۱۷۱۵	۰,۷۲
$X_8(cm^2)$	۵۷۰	۱۹۶۶	۱۱۸۳	۳۹,۸
$X_9(cm^2)$	۱۳۳۰	-۱۴۲۰۸	-۱۳۹۳۳	۱,۹۳
$X_{10}(cm^2)$	۳۲۰	۱۰۸۰۴	۱۲۰۹۳	۱۱,۹
$X_{11}(cm^2)$	۹۰	-۱۲۱۳	-۲۳۴۶	۹۳,۴
$X_{12}(cm^2)$	۱۱۱۵	۱۴۳۳۷	۱۳۸۰۸	۳,۶۸
$X_{13}(cm^2)$	۲۰۰	۱۵۳۴۹	۱۵۵۹۳	۱,۵۹
$x_2(cm)$	۲۷۰	-----	-----	-----
$y_2(cm)$	۸۵	-----	-----	-----
$x_4(cm)$	۷۱۵	-----	-----	-----
$y_4(cm)$	۲۶۵	-----	-----	-----
$y_6(cm)$	۴۲۵	-----	-----	-----
				$\overline{E} = ۹,۸۶٪$

در جدول (۵-۱۱) و (۵-۱۲) مقادیر X_i برای $i=1,2,...,13$ سطح مقطع عضو iام بوده و سطح مقطع اعضاء دیگر یعنی X_{14} ، X_{15} ، ... و X_{25} بدلیل تقارن بترتیب برابر است با X_9 ، X_{10} ، X_{11} ، X_{12} ، X_8 ، X_4 ، X_5 ، X_6 ، X_7 ، X_3 ، X_1 و X_2 .

نتیجه گیری و پیشنهادات

نتیجه گیری :

در این تحقیق سعی شده است تا توانمندیهای شبکه های عصبی در بهینه یابی خرپا معرفی گردد . همانطورکه تشریح گردید بهینه سازی خرپا مستلزم تحلیل خرپا بدفعات است . در این تحقیق برای تحلیل خرپا از شبکه عصبی انتشار متقابل استفاده شده است که ابتدا توسط یکسری زوج بردار های ورودی - خروجی تربیت می شود تا مراحل تحلیل خرپا را یاد بگیرد . پس از پایان یافتن تربیت شبکه جهت تحلیل خرپا ، در تمامی مراحل بهینه سازی خرپا که احتیاج به نتایج تحلیل است ، از شبکه عصبی آموزش دیده استفاده می کنیم . نتایج بدست آمده از بهینه سازی خرپا توسط شبکه عصبی نشان می دهد استفاده از شبکه عصبی برای بهینه سازی از دقت قابل قبولی برخوردار است و همچنین در ادامه نشان داده شده است که که چگونه فضای جستجو (دامنه متغیرها) نتایج بهینه سازی را تحت تاثیر قرار می دهد . هر چقدر این فضا برای بهینه سازی خرپا توسط شبکه محدودتر باشد ، نتایج بهینه سازی خرپا با خطای کمتری همراه است و بالعکس . مزیت این روش بیشتر در مواجهه با مسائل با مقیاس بزرگ که نیاز به محاسبات و زمان قابل توجه رایانه ای دارد نمایان می شود . بنابراین توسعه و بکار گیری روش های محاسباتی که منتهی به پاسخ خوب وقابل اعتماد شوند و درعین حال محاسبات وزمان را به صورت قابل ملاحظه ای کاهش می دهند ، مورد توجه قرار گرفته است .

پیشنهادات :

برای توسعه ، شفاف سازی و اصلاح قوانین محاسبات عصبی با کارایی مناسب ، تحقیق در موارد زیر به علاقه مندان توصیه می شود :

۱- بکار گیری شبکه های عصبی در تحلیل ، طراحی و بهینه سازی قاب ها

۲- مطالعه بروی ساختار های جدید شبکه های عصبی مصنوعی

۳- ارز یابی تغییر پارامتر های شبکه عصبی از جمله تعداد زوج های آموزشی و پارامتر گروه بندی بر عملکرد شبکه

۴- مطالعه بروی افزایش حوزه های یادگیری شبکه عصبی جهت پاسخگویی به سوالات بیشتر و عکس العمل شبکه در برابر حوزه های یاد گیری جدید

۵- مطالعه بر روی ترکیب شبکه های عصبی مصنوعی با الگوریتم های ژنتیک و معیار های دیگر همچون معیار فازی

۶- بهینه یابی خرپا با شبکه های عصبی مصنوعی با در نظر گرفتن قیود دیگر از جمله محدودیت جابجایی

۹- بکارگیری شبکه های عصبی مصنوعی در طراحی اولیه یا مقدماتی سازه ها برای تعیین ابعاد اولیه اعضاء سازه

فهرست منابع و مأخذ

۱- رابرت جی . شالکف ، شبکه های عصبی مصنوعی، مترجمان : محمود جورابیان ، طناز زارع و امید استوار ، چاپ اول ، دانشگاه شهید چمران اهواز ، ۱۳۸۲

۲- منهاج ، محمد باقر ، هوش مصنوعی و محاسبات عصبی ، چاپ دوم ، جلد اول ، تهران ، مرکز نشر پرفسور حسابی ، ۱۳۷۹

۳- راسل ، استوارت جاناتان ، هوش مصنوعی : رهیافتی نوین ، مترجمان : رامین رهنمون ، آناهیتا هماوندی ، انتشارات ناقوس ، ۱۳۸۳

۴- کاوه ، علی ، شبکه های عصبی مصنوعی در بهینه سازی سازه ها ، چاپ اول ، تهران ، مرکز تحقیقات ساختمان و مسکن ، ۱۳۷۸

۵- رائو ، اس . اس ، بهینه سازی (تئوری و کاربرد) ، ترجمه : سید محمد شهیدی پور ، دانشگاه فردوسی (مشهد) ، ۱۳۷۳

۶- نشریه استقلال ، شماره دو ، اسفند ۱۳۸۰ ، صص ۱۲۱-۱۰۷

۷- فصل نامه مهندسی صنایع ، شماره سی ، بهار ۸۱ ، صص ۶۶-۵۹

۸- مجله بین المللی علوم مهندسی ، جلد ۱۲ ، شماره دو ، تابستان ۱۳۸۰ ، صص ۱۶-۱

۹- Prof . Uri Kirsch , Structural Optimization (Fundamentals and Applications) , Springer – Verlag , Berlin 1993

پیوست

پیوست شماره یک (بهینه سازی دقیق خرپای مثال اول)

```
C Nmaj=Tedade Majhola
C NBW=Max(2*ABS(I-J)+2)
C NN=Tedade Gereh
C ND=Tedede Aksolamal Tekiegahi
C NE=Tedade Ozvha
C-----------------------------------------------------------------------
dimension xx(Nmaj, 2*Nmaj+3),ss(Nmaj, 2*Nmaj+3),ff(2*Nmaj+3),
c   X(NN,2),NOC(NE,4),PM(1,2),AREA(2*Nmaj+3, 2*Nmaj+3),NU(ND),stressp(NE),
c   REACT(ND),F(2*NN),SE(4,4),S(2*NN,NBW),AL(NE,2*Nmaj+3),U(ND)
C***********************************************************************
                    Nmaj=5
                    NE=3
                    NN=4
                    ND=6
                    NL=2
                    NM=1
                    NA=3
                    e=21000000.0
c-----------------------------------------------------------------------
                    xx(1,1)=3.0
                    xx(2,1)=19.0
                    xx(3,1)=0.1
                    xx(4,1)=154.0
                    xx(5,1)=460.0
C                     ...
c-----------------------------------------------------------------------
                    xx(1,2)=3.5
                    xx(2,2)=18.1
                    xx(3,2)=0.1
                    xx(4,2)=130.0
                    xx(5,2)=440.0
C                     ...
c-----------------------------------------------------------------------
                    xx(1,3)=3.8
                    xx(2,3)=19.0
                    xx(3,3)=0.2
                    xx(4,3)=130.0
                    xx(5,3)=460.0
c-----------------------------------------------------------------------
                    xx(1,4)=3.8
                    xx(2,4)=19.5
                    xx(3,4)=0.18
                    xx(4,4)=90.0
                    xx(5,4)=470.0
c-----------------------------------------------------------------------
                    xx(1,5)=3.2
                    xx(2,5)=19.0
                    xx(3,5)=0.2
                    xx(4,5)=55.0
                    xx(5,5)=460.0
c-----------------------------------------------------------------------
                    xx(1,6)=3.2
                    xx(2,6)=18.0
                    xx(3,6)=0.25
                    xx(4,6)=19.0
                    xx(5,6)=470.0
c-----------------------------------------------------------------------
                    xx(1,7)=3.5
                    xx(2,7)=17.0
```

```
                xx(3,7)=0.2
                xx(4,7)=130.0
                xx(5,7)=400.0
c-------------------------------------------------------------------
                xx(1,8)=3.4
                xx(2,8)=18.0
                xx(3,8)=0.17
                xx(4,8)=120.0
                xx(5,8)=442.0
c-------------------------------------------------------------------
                xx(1,9)=3.0
                xx(2,9)=18.3
                xx(3,9)=0.1
                xx(4,9)=10.0
                xx(5,9)=470.0
c-------------------------------------------------------------------
                xx(1,10)=3.1
                xx(2,10)=18.5
                xx(3,10)=0.2
                xx(4,10)=15.0
                xx(5,10)=420.0
c-------------------------------------------------------------------
                xx(1,12)=0
                xx(2,12)=0
                xx(3,12)=0
                xx(4,12)=0
                xx(5,12)=0
c-------------------------------------------------------------------
                xx(1,13)=0
                xx(2,13)=0
                xx(3,13)=0
                xx(4,13)=0
                xx(5,13)=0
c-------------------------------------------------------------------
                do 30 jL=1,2*Nmaj
C*************************************************************************
C*** N= SHOMAREH OZVE , 1=SHOMARE i , 2=SHOMAREH j, 3=1 ,4=AREA ******
C***************** NOC(N,i)=SHOMAREH (i)! , NOC(N,J)=? ******************
C               N=1
                NOC(1,1)=1
                NOC(1,2)=2
                NOC(1,3)=1
                NOC(1,4)=1
C               N=2
                NOC(2,1)=3
                NOC(2,2)=2
                NOC(2,3)=1
                NOC(2,4)=2
C               N=3
                NOC(3,1)=4
                NOC(3,2)=2
               NOC(3,3)=1
                NOC(3,4)=3
C               N=4
c               NOC(4,1)=4
c               NOC(4,2)=3
c              NOC(4,3)=1
c               NOC(4,4)=4
C               N=5
C               NOC(N,1)=
C               NOC(N,2)=
C               NOC(N,3)=
C               NOC(N,4)=
C               N=6
```

```
C                 NOC(N,1)=
C                 NOC(N,2)=
C                NOC(N,3)=
C                 NOC(N,4)=
c*************************************************************************
                  DO 45 N=1,NE
                  C=2*(ABS(NOC(N,2)-NOC(N,1)))+2.0
                  IF(NBW.LT.C)THEN
                  NBW=C
                  END IF
45                CONTINUE
c*************************************************************************
                  do 44 i=1,NE
                  AREA(i,jL)=xx(i,jL)
44                CONTINUE
C*************************************************************************
C*****             DIMENSION S(NQ,NBW),F(NQ),SE(4,4) ************************
                  DO 50 I=1,NQ
                  F(NQ)=0
                  DO 55 JJ=1,NBW
                  S(I,JJ)=0
55                CONTINUE
50                CONTINUE
C*************************************************************************
C************* MOKHTASAT X=X(N,1) , Y=X(N,2) **** N=SHOMAREH GEREH********
c******************************** X(N,1)=X(gereh) , X(N,2)=Y(gereh)********
                  X(1,1)=-480.0
                  X(1,2)=0.0
                  X(2,1)=xx(4,jL)
                  X(2,2)=xx(5,jL)
                  X(3,1)=0
                  X(3,2)=0
                  X(4,1)=360.0
                  X(4,2)=0
C                 X(5,1)=
C                 X(5,2)=
C                 X(6,1)=
C                 X(6,2)=
C*************************************************************************
C********* E= BE TEDAD NM (KE MAMOLAN YEKI AST) **************************
                  E=21000000.0
                  PM(1,1)=E
                  PM(1,2)=0.0
C*************************************************************************
C************* JAHATHAII KEH TEKIEHGAH AST U(I)=0*************************
                  NU(1)=1
                  U(1)=0
                  NU(2)=2
                  U(2)=0
                  NU(3)=5
                  U(3)=0
                  NU(4)=6
                  U(4)=0
                  NU(5)=7
                  U(5)=0
                  NU(6)=8
                  U(6)=0
c                 NU(7)=8
c                 U(7)=0
C*************************************************************************
C***N=SHOMAREH GEREH , BARE VAREDEH DAR JAHATE U(I),U(I+1) ***************
                  N=?
                  F(2*N-1)=90000
                  F(2*N)=225000
```

```
c                 N=3
c                 F(2*N-1)=0
c                 F(2*N)=-100000
C                 N=3
C                 F(2*N-1)=?
C                 F(2*N)=??
C                 N=4
C                 F(2*N-1)=?
C                 F(2*N)=??
C*****************************************************************************
C***************** SAKHTI KOLI ***********************************************
                  DO 60 N=1,NE
                  I1=NOC(N,1)
                  I2=NOC(N,2)
                  I3=NOC(N,3)
                  I4=NOC(N,4)
                  X21=X(I2,1)-X(I1,1)
                  Y21=X(I2,2)-X(I1,2)
                  EL=SQRT(X21**2+Y21**2)
                  EAL=E*AREA(I4,jL)/EL
                  CS=X21/EL
                  SN=Y21/EL
                  SE(1,1)=CS*CS*EAL
                  SE(1,2)=CS*SN*EAL
                  SE(2,1)=SE(1,2)
                  SE(1,3)=-CS*CS*EAL
                  SE(3,1)=SE(1,3)
                  SE(1,4)=-CS*SN*EAL
                  SE(4,1)=SE(1,4)
                  SE(2,2)=SN*SN*EAL
                  SE(2,3)=-CS*SN*EAL
                  SE(3,2)=SE(2,3)
                  SE(2,4)=-SN*SN*EAL
                  SE(4,2)=SE(2,4)
                  SE(3,3)=CS*CS*EAL
                  SE(3,4)=CS*SN*EAL
                  SE(4,3)=SE(3,4)
                  SE(4,4)=SN*SN*EAL
                  DO 65 II=1,2
                  NRT=2*(NOC(N,II)-1)
                  DO 70 IT=1,2
                  NR=NRT+IT
                  I=2*(II-1)+IT
                  DO 75 JJ=1,2
                  NCT=2*(NOC(N,JJ)-1)
                  DO 80 JT=1,2
                  J=2*(JJ-1)+JT
                  NC=NCT+JT-NR+1
                  IF(NC.LE.0) THEN
                    GO TO 80
                  END IF
                  S(NR,NC)=S(NR,NC)+SE(I,J)
80                CONTINUE
75                CONTINUE
70                CONTINUE
65                CONTINUE
60                CONTINUE
C*****************************************************************************
C******** SHARAYET MARZII ************************************************
                  CNST=S(1,1)*10000
                  DO 85 I=1,ND
                  N=NU(I)
                  S(N,1)=S(N,1)+CNST
                  F(N)=F(N)+CNST*U(I)
```

```
85                CONTINUE
C*************************************************************************
C********** HALLE MOADELAT ***********************************************
                  N1=NQ-1
                  DO 90 K=1,N1
                  NK=NQ-K+1
                  IF(NK.GT.NBW) THEN
                    NK=NBW
                  END IF
                  DO 95 I=2,NK
                  C1=S(K,I)/S(K,1)
                  I1=K+I-1
                  DO 100 J=I,NK
                  J1=J-I+1
                  S(I1,J1)=S(I1,J1)-C1*S(K,J)
100               CONTINUE
                  F(I1)=F(I1)-C1*F(K)
95                CONTINUE
90                CONTINUE
                  F(NQ)=F(NQ)/S(NQ,1)
                  DO 105 KK=1,N1
                  K=NQ-KK
                  C1=1/S(K,1)
                  F(K)=C1*F(K)
                  NK=NQ-K+1
                  IF(NK.GT.NBW)THEN
                        NK=NBW
                  END IF
                  DO 110 J=2,NK
                  F(K)=F(K)-C1*S(K,J)*F(K+J-1)
110               CONTINUE
105               CONTINUE
C*************************************************************************
C************************** STRESS  ************************************
                  DO 115 I=1,NE
                  I1=NOC(I,1)
                  I2=NOC(I,2)
                  I3=NOC(I,3)
                  X21=X(I2,1)-X(I1,1)
                  Y21=X(I2,2)-X(I1,2)
                  EL=SQRT(X21**2+Y21**2)
                  CS=X21/EL
                  SN=Y21/EL
                  J2=2*I1
                  J1=J2-1
                  K2=2*I2
                  K1=K2-1
                  DT=(F(K1)-F(J1))*CS+(F(K2)-F(J2))*SN
                  STRESS=DT*PM(I3,1)/EL
                  STRESSP(I)=STRESS
115               CONTINUE
C*************************************************************************
c-------------------------------------------------------------------
          ss(1,jL)=stressp(1)
          ss(2,jL)=stressp(2)
          ss(3,jL)=stressp(3)
c         ss(4,jL)=stressp(4)
C         ss(4,jL)=xx(4,jL)
C         ss(5,jL)=xx(5,jL)
c         ss(7,jL)=xx(7,jL)
c         ss(8,jL)=xx(8,jL)
c**********
30        continue
c-------------------------------------------------------------------
```

```
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
35                  do 31 i=1,Nmaj
                    xx(i,2*Nmaj+1)=0
31                  continue
c-----------------------------------------------------------------------
          do 190 i=1,Nmaj
          do 195 j=1,2*Nmaj
          xx(i,2*Nmaj+1)=xx(i,j)/(2*Nmaj)+xx(i,2*Nmaj+1)
195       continue
190       continue
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
          do 200 j=1,2*Nmaj+1
                    FF(J)=0
                    DO 199 N=1,NE
                    AREA(1,j)=xx(1,j)
                    AREA(2,j)=xx(2,j)
                    AREA(3,j)=xx(3,j)
c                   AREA(4,j)=xx(4,j)
c************* MOKHTASAT X=X(N,1) , Y=X(N,2) **** N=SHOMAREH GEREH*******
                    X(1,1)=-480.0
                    X(1,2)=0
                    X(2,1)=xx(4,j)
                    X(2,2)=xx(5,j)
                    X(3,1)=0
                    X(3,2)=0
                    X(4,1)=360.0
                    X(4,2)=0
c**********
                    I1=NOC(N,1)
                    I2=NOC(N,2)
                    I3=NOC(N,3)
                    I4=NOC(N,4)
                    X21=X(I2,1)-X(I1,1)
                    Y21=X(I2,2)-X(I1,2)
                    EL=SQRT(X21**2+Y21**2)
                    AL(N,J)=AREA(I4,J)*EL
                    FF(J)=FF(J)+AL(N,J)
199                 continue
200   continue
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
          c=0
          do 205 j=1,2*Nmaj
          c=(ff(2*Nmaj+1)-ff(j))**2+c
205       continue
c*************************************************************************
          if((1.0/(2*Nmaj)*c)**0.5.lt.4.5) go to 2000
          max=0
          do 210 j=1,2*Nmaj
          if(ff(j).gt.max) then
          max=ff(j)
          h=j
          end if
210   continue
c*************************************************************************
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
                    kj=2*Nmaj+2
          do 215 i=1,Nmaj
                    xx(i,kj)=0
                    do 214 j=1,2*Nmaj
                    xx(i,kj)=xx(i,kj)+xx(i,j)
```

```
214                 continue
                    xx(i,kj)=1.0/(2*Nmaj-1)*(xx(i,kj)-xx(i,h))
215                 continue
c*************************************************************************
                    a=1.3
1600   do 220 i=1,Nmaj
     xx(i,kj+1)=(1+a)*xx(i,kj)-a*xx(i,h)
220   continue
c*************************************************************************
                    DO 225 N=1,NE
                    C=2*(ABS(NOC(N,2)-NOC(N,1)))+2.0
                    IF(NBW.LT.C)THEN
                    NBW=C
                    END IF
225                 CONTINUE
c*************************************************************************
                    do 224 i=1,NE
                    AREA(i,kj+1)=xx(i,kj+1)
224                 CONTINUE
C*************************************************************************
C*****               DIMENSION S(NQ,NBW),F(NQ),SE(4,4) ************************
                    DO 240 I=1,NQ
                    F(NQ)=0
                    DO 235 JJ=1,NBW
                    S(I,JJ)=0
235                 CONTINUE
240                 CONTINUE
C*************************************************************************
C************* MOKHTASAT X=X(N,1) , Y=X(N,2) **** N=SHOMAREH GEREH********
c******************************** X(N,1)=X(gereh) , X(N,2)=Y(gereh)********
                    X(1,1)=-480.0
                    X(1,2)=0
                    X(2,1)=xx(4,kj+1)
                    X(2,2)=xx(5,kj+1)
                    X(3,1)=0
                    X(3,2)=0
                    X(4,1)=360.0
                    X(4,2)=0
C                   X(5,1)=
C                   X(5,2)=
C                   X(6,1)=
C                   X(6,2)=
C*************************************************************************
C********* E= BE TEDAD NM (KE MAMOLAN YEKI AST) **************************
                    E=21000000.0
                    PM(1,1)=E
                    PM(1,2)=0.0
C*************************************************************************
C***N=SHOMAREH GEREH , BARE VAREDEH DAR JAHATE U(I),U(I+1) ***************
                    N=2
                    F(2*N-1)=90000
                    F(2*N)=225000
c                   N=3
c                   F(2*N-1)=0
c                   F(2*N)=-100000
C                   N=3
C                   F(2*N-1)=?
C                   F(2*N)=??
C                   N=4
C                   F(2*N-1)=?
C                   F(2*N)=??
C*************************************************************************
C***************** SAKHTI KOLI ***************************************
                    DO 250 N=1,NE
```

```
                I1=NOC(N,1)
                I2=NOC(N,2)
                I3=NOC(N,3)
                I4=NOC(N,4)
                X21=X(I2,1)-X(I1,1)
                Y21=X(I2,2)-X(I1,2)
                EL=SQRT(X21**2+Y21**2)
                EAL=E*AREA(I4,kj+1)/EL
                CS=X21/EL
                SN=Y21/EL
                SE(1,1)=CS*CS*EAL
                SE(1,2)=CS*SN*EAL
                SE(2,1)=SE(1,2)
                SE(1,3)=-CS*CS*EAL
                SE(3,1)=SE(1,3)
                SE(1,4)=-CS*SN*EAL
                SE(4,1)=SE(1,4)
                SE(2,2)=SN*SN*EAL
                SE(2,3)=-CS*SN*EAL
                SE(3,2)=SE(2,3)
                SE(2,4)=-SN*SN*EAL
                SE(4,2)=SE(2,4)
                SE(3,3)=CS*CS*EAL
                SE(3,4)=CS*SN*EAL
                SE(4,3)=SE(3,4)
                SE(4,4)=SN*SN*EAL
                DO 255 II=1,2
                NRT=2*(NOC(N,II)-1)
                DO 260 IT=1,2
                NR=NRT+IT
                I=2*(II-1)+IT
                DO 265 JJ=1,2
                NCT=2*(NOC(N,JJ)-1)
                DO 280 JT=1,2
                J=2*(JJ-1)+JT
                NC=NCT+JT-NR+1
                IF(NC.LE.0) THEN
                  GO TO 280
                END IF
                S(NR,NC)=S(NR,NC)+SE(I,J)
280             CONTINUE
265             CONTINUE
260             CONTINUE
255             CONTINUE
250             CONTINUE
C*************************************************************************
C******** SHARAYET MARZII ************************************************
                CNST=S(1,1)*10000
                DO 285 I=1,ND
                N=NU(I)
                S(N,1)=S(N,1)+CNST
                F(N)=F(N)+CNST*U(I)
285             CONTINUE
C*************************************************************************
C********** HALLE MOADELAT ***********************************************
                N1=NQ-1
                DO 290 K=1,N1
                NK=NQ-K+1
                IF(NK.GT.NBW) THEN
                 NK=NBW
                END IF
                DO 295 I=2,NK
                C1=S(K,I)/S(K,1)
                I1=K+I-1
```

```
                DO 300 J=I,NK
                J1=J-I+1
                S(I1,J1)=S(I1,J1)-C1*S(K,J)
300             CONTINUE
                F(I1)=F(I1)-C1*F(K)
295             CONTINUE
290             CONTINUE
                F(NQ)=F(NQ)/S(NQ,1)
                DO 305 KK=1,N1
                K=NQ-KK
                C1=1/S(K,1)
                F(K)=C1*F(K)
                NK=NQ-K+1
                IF(NK.GT.NBW)THEN
                        NK=NBW
                END IF
                DO 310 J=2,NK
                F(K)=F(K)-C1*S(K,J)*F(K+J-1)
310             CONTINUE
305             CONTINUE
C*************************************************************************
C************************* STRESS  ***************************************
                DO 315 I=1,NE
                I1=NOC(I,1)
                I2=NOC(I,2)
                I3=NOC(I,3)
                X21=X(I2,1)-X(I1,1)
                Y21=X(I2,2)-X(I1,2)
                EL=SQRT(X21**2+Y21**2)
                CS=X21/EL
                SN=Y21/EL
                J2=2*I1
                J1=J2-1
                K2=2*I2
                K1=K2-1
                DT=(F(K1)-F(J1))*CS+(F(K2)-F(J2))*SN
                STRESS=DT*PM(I3,1)/EL
                STRESSP(I)=STRESS
315             CONTINUE
C*************************************************************************
c-------------------------------------------------------------------
          ss(1,kj+1)=STRESSP(1)
          ss(2,kj+1)=STRESSP(2)
          ss(3,kj+1)=STRESSP(3)
c          ss(4,jL)=stressp(4)
c         ss(5,jL)=xx(5,jL)
c*************************************************************************
c*************************************************************************
                FF(kj+1)=0
                DO 320 N=1,NE
                AREA(1,kj+1)=xx(1,kj+1)
                AREA(2,kj+1)=xx(2,kj+1)
                AREA(3,kj+1)=xx(3,kj+1)
c               AREA(4,j)=xx(4,j)
                X(1,1)=-480.0
                X(1,2)=0
                X(2,1)=xx(4,kj+1)
                X(2,2)=xx(5,kj+1)
                X(3,1)=0
                X(3,2)=0
                X(4,1)=360.0
                X(4,2)=0
                I1=NOC(N,1)
                I2=NOC(N,2)
```

```
                    I3=NOC(N,3)
                    I4=NOC(N,4)
                    X21=X(I2,1)-X(I1,1)
                    Y21=X(I2,2)-X(I1,2)
                    EL=SQRT(X21**2+Y21**2)
                    AL(N,kj+1)=AREA(I4,kj+1)*EL
                    FF(kj+1)=FF(kj+1)+AL(N,kj+1)
320                 continue
                    ff(kj)=ff(kj+1)
c*************************************************************************
c*************************************************************************
c***********************    Geydhaye Masale    ************************
c******** Kontrole Taneshha !? ***********************************************
                    Narea=3
                    do 325 i=1,Narea
                    if (xx(i,kj+1).ge.0) go to 251
                       go to 261
251                 if ((ss(i,kj+1)/13000)-1.lt.0) go to 252
                            go to 261
252     if((-ss(i,kj+1)/13000)-1.lt.0) go to 253
                            go to 261
c*************************************************************************
c******** Kontrole Sharayete Hendessi !? *********************************
253     if (xx(4,kj+1).ge.-200.0) go to 254
                            go to 261
254           if (xx(4,kj+1).le.200.0)  go to 256
                            go to 261
256                 if (xx(5,kj+1).ge.350.0)  go to 257
                            go to 261
257           if (xx(5,kj+1).le.550.0) go to 325
261     do 336 k=1,Nmaj
      xx(k,kj+1)=(xx(k,kj)+xx(k,kj+1))/2.0
336     continue
c?????????????????????????????????????????/?????????????????????????///
c*************************************************************************
                    FF(kj+1)=0
                    DO 340 N=1,NE
                    j=kj+1
c*********
                    AREA(1,j)=xx(1,j)
                    AREA(2,j)=xx(2,j)
                    AREA(3,j)=xx(3,j)
c                   AREA(4,j)=xx(4,j)
                    X(1,1)=-480.0
                    X(1,2)=0
                    X(2,1)=xx(4,j)
                    X(2,2)=xx(5,j)
                    X(3,1)=0
                    X(3,2)=0
                    X(4,1)=360.0
                    X(4,2)=0
                    I1=NOC(N,1)
                    I2=NOC(N,2)
                    I3=NOC(N,3)
                    I4=NOC(N,4)
                    X21=X(I2,1)-X(I1,1)
                    Y21=X(I2,2)-X(I1,2)
                    EL=SQRT(X21**2+Y21**2)
                    AL(N,j)=AREA(I4,j)*EL
                    FF(j)=FF(j)+AL(N,j)
340                 continue
                    ff(j-1)=ff(j)
325   continue
c---------------------------------------------------------------------
```

```
                if(ff(h).gt.ff(kj)) go to 1244
                a=a/2
                go to 1600
1244            do 283 i=1,Nmaj
                xx(i,h)=xx(i,kj+1)
283             continue
                go to 35
2000    print*,xx(1,kj-1),xx(2,kj-1),xx(3,kj-1),xx(4,kj-1),xx(5,kj-1),
   c    ff(kj-1)
c3000           PRINT*,ff(12)
                end
```

پیوست شماره دو (متن کامل برنامه بهینه یابی خرپای مثال اول با شبکه عصبی)

```
C Nmaj=Tedade Majhola
C Nzoj=Tedade Kole Zojhaye Amozesh
C NBW=Max(2*ABS(I-J)+2)
C NN=Tedade Gereh
C ND=Tedede Aksolamal Tekiegahi
C NE=Tedade Ozvha
C-----------------------------------------------------------------------
dimension v(Nmaj,Nzoj),w(Nmaj,Nzoj),xx(Nmaj,Nzoj),y(Nmaj,Nzoj),
c                d(Nzoj),ss(Nmaj,Nzoj),ff(2*Nmaj+3),L(NE),SF(Nmaj,Nzoj),
c   X(NN,2),NOC(NE,4),PM(1,2),AREA(Nzoj,Nzoj),NU(ND),stressp(NE),
c   REACT(ND),F(2*NN),SE(4,4),S(2*NN,NBW),AL(NE,2*Nmaj+3),U(ND)
C***********************************************************************
          Print*,'Enter Nmaj,Nzoj,NE,NN,ND ?'
          Read (*,*)Nmaj,Nzoj,NE,NN,ND
C***********************************************************************
           E=21000000.0
           a=0.7
           b=0.1
           n=1
           p=1
C***********************************************************************
                    NQ=2*NN
                    NBW=0
c***********************************************************************
c---------------------------------- FAZAYE JOSTOJO ----------------------------------
          print*,'Enter fazaye jostojo x0(1),dxx(1), ... ,x0(Nmaj),dxx(Nmaj) ?'
          read (*,*) (x0(i),dxx(i),i=1,Nmaj)
        do 5 k=1,Nzoj
          do 1 i=1,Nmaj
            xx(i,k)=x0(i)+k*dxx(i)
1      contiue
5      continue
c-----------------------------------------------------------------------
C*** N= SHOMAREH OZVE , 1=SHOMARE i , 2=SHOMAREH j, 3=1 ,4=AREA *******
C***************** NOC(N,i)=SHOMAREH (i)! , NOC(N,J)=? ******************
          Print*,'N=shomare ozve , 1=shomare i , 2=shomare j , 3=jens , 4=area'
          Print *,'enter NOC(N,1),NOC(N,2),NOC(N,3),NOC(N,4)'
          Read (*,*) (NOC(I,1),NOC(I,2),NOC(I,3),NOC(I,4),I=1,NE)
c-----------------------------------------------------------------------
C************* MOKHTASAT X=X(N,1) , Y=X(N,2) **** N=SHOMAREH GEREH********
c******************************** X(N,1)=X(gereh) , X(N,2)=Y(gereh)********
                    X(1,1)=-480.0
                    X(1,2)=0.0
                    X(2,1)=xx(4,jL)
                    X(2,2)=xx(5,jL)
                    X(3,1)=0.0
                    X(3,2)=0.0
                    X(4,1)=360.0
                    X(4,2)=0
C                   X(5,1)=
C                   X(5,2)=
C                   X(6,1)=
C                   X(6,2)=
C***********************************************************************
C************* JAHATHAII KEH TEKIEHGAH AST U(I)=0************************
          print*,'Enter jahati keh tekiegah ast'
          print*,'NU(I)=? , U(I)=0'
          Read(*,*) (NU(i),U(i),i=1,ND)
c-----------------------------------------------------------------------
```

```
C***N=SHOMAREH GEREH , BARE VAREDEH DAR JAHATE U(I),U(I+1) ***************
      Print*,'Enter N=shomare gereh , bare varede be gereh'
      read(*,*) N , F(2*N-1) , F(2*N)
C*************************************************************************
c-------------------------------------------------------------------
c-------------------------------------------------------------------
      DO 6 I=1,Nmaj
         v(I,p)=xx(I,1)
6     CONTINUE
c-------------------------------------------------------------------
7      do 10 k=2,Nzoj
       do 15 j=1,p
       do 20 i=1,Nmaj
       y(i,j)=xx(i,k)-v(i,j)
20     continue
15     continue
c-------------------------------------------------------------------
       do 25 j=1,p
      dd=0
       do 24 i=1,Nmaj
       dd=dd+y(i,j)**2
24     continue
       d(j)=sqrt(dd)
25     continue
c-------------------------------------------------------------------
       min=d(1)
       m=1
       do 30 j=1,p
       if(d(j).lt.min) then
       min=d(j)
       m=j
      end if
30     continue
c-------------------------------------------------------------------
       if(d(m).gt.0.5) then
       p=p+1
            DO 31 I=1,Nmaj
       v(I,p)=xx(I,k)
31          CONTINUE
       else
       do 35 i=1,Nmaj
       v(i,m)=(1-a)*v(i,m)+a*xx(i,k)
35     continue
       end if
10   continue
c-------------------------------------------------------------------
c-------------------------------------------------------------------
    do 40 jL=1,Nzoj
C*************************************************************************
c*************************************************************************
            DO 45 N=1,NE
            C=2*(ABS(NOC(N,2)-NOC(N,1)))+2.0
            IF(NBW.LT.C)THEN
            NBW=C
            END IF
45          CONTINUE
c*************************************************************************
            do 44 i=1,NE
            AREA(i,jL)=xx(i,jL)
44          CONTINUE
C*************************************************************************
C*****       DIMENSION S(NQ,NBW),F(NQ),SE(4,4) ************************
            DO 50 I=1,NQ
            F(NQ)=0
```

```
                DO 55 JJ=1,NBW
                S(I,JJ)=0
55              CONTINUE
50              CONTINUE
C*************************************************************************
C********* E= BE TEDAD NM (KE MAMOLAN YEKI AST) ***************************
                E=21000000.0
                PM(1,1)=E
                PM(1,2)=0.0
C*************************************************************************
C*************************************************************************
C***************** SAKHTI KOLI ********************************************
                DO 60 N=1,NE
                I1=NOC(N,1)
                I2=NOC(N,2)
                I3=NOC(N,3)
                I4=NOC(N,4)
                X21=X(I2,1)-X(I1,1)
                Y21=X(I2,2)-X(I1,2)
                EL=SQRT(X21**2+Y21**2)
                EAL=E*AREA(I4,jL)/EL
                CS=X21/EL
                SN=Y21/EL
                SE(1,1)=CS*CS*EAL
                SE(1,2)=CS*SN*EAL
                SE(2,1)=SE(1,2)
                SE(1,3)=-CS*CS*EAL
                SE(3,1)=SE(1,3)
                SE(1,4)=-CS*SN*EAL
                SE(4,1)=SE(1,4)
                SE(2,2)=SN*SN*EAL
                SE(2,3)=-CS*SN*EAL
                SE(3,2)=SE(2,3)
                SE(2,4)=-SN*SN*EAL
                SE(4,2)=SE(2,4)
                SE(3,3)=CS*CS*EAL
                SE(3,4)=CS*SN*EAL
                SE(4,3)=SE(3,4)
                SE(4,4)=SN*SN*EAL
                DO 65 II=1,2
                NRT=2*(NOC(N,II)-1)
                DO 70 IT=1,2
                NR=NRT+IT
                I=2*(II-1)+IT
                DO 75 JJ=1,2
                NCT=2*(NOC(N,JJ)-1)
                DO 80 JT=1,2
                J=2*(JJ-1)+JT
                NC=NCT+JT-NR+1
                IF(NC.LE.0) THEN
                  GO TO 80
                END IF
                S(NR,NC)=S(NR,NC)+SE(I,J)
80              CONTINUE
75              CONTINUE
70              CONTINUE
65              CONTINUE
60              CONTINUE
C*************************************************************************
C******** SHARAYET MARZII *********************************************
                CNST=S(1,1)*10000
                DO 85 I=1,ND
                N=NU(I)
                S(N,1)=S(N,1)+CNST
```

```
                F(N)=F(N)+CNST*U(I)
85              CONTINUE
C*************************************************************************
C********** HALLE MOADELAT ***********************************************
                N1=NQ-1
                DO 90 K=1,N1
                NK=NQ-K+1
                IF(NK.GT.NBW) THEN
                NK=NBW
                END IF
                DO 95 I=2,NK
                C1=S(K,I)/S(K,1)
                I1=K+I-1
                DO 100 J=I,NK
                J1=J-I+1
                S(I1,J1)=S(I1,J1)-C1*S(K,J)
100             CONTINUE
                F(I1)=F(I1)-C1*F(K)
95              CONTINUE
90              CONTINUE
                F(NQ)=F(NQ)/S(NQ,1)
                DO 105 KK=1,N1
                K=NQ-KK
                C1=1/S(K,1)
                F(K)=C1*F(K)
                NK=NQ-K+1
                IF(NK.GT.NBW)THEN
                NK=NBW
                END IF
                DO 110 J=2,NK
                F(K)=F(K)-C1*S(K,J)*F(K+J-1)
110             CONTINUE
105             CONTINUE
C*************************************************************************
C************************** STRESS ***************************************
                DO 115 I=1,NE
                I1=NOC(I,1)
                I2=NOC(I,2)
                I3=NOC(I,3)
                X21=X(I2,1)-X(I1,1)
                Y21=X(I2,2)-X(I1,2)
                EL=SQRT(X21**2+Y21**2)
                CS=X21/EL
                SN=Y21/EL
                J2=2*I1
                J1=J2-1
                K2=2*I2
                K1=K2-1
                DT=(F(K1)-F(J1))*CS+(F(K2)-F(J2))*SN
                STRESS=DT*PM(I3,1)/EL
                STRESSP(I)=STRESS
115             CONTINUE
C*********************************************
c-------------------------------------------------------------------
          do 111 i=1,NE
           ss(i,jL)=stressp(i)
111       continue
          do 112 i=NE+1,Nmaj
           ss(i,jL)=stressp(i)
112        contiue
c*********************
40   continue
c-------------------------------------------------------------------
c*************************************************************************
```

```
      do 125 k=1,Nzoj
            do 130 j=1,p
            do 135 i=1,Nmaj
            y(i,j)=xx(i,k)-v(i,j)
135         continue
130         continue
c-------------------------------------------------------------------------
            do 140 j=1,p
            dd=0
            do 139 i=1,Nmaj
            dd=dd+y(i,j)**2
139         continue
            d(j)=sqrt(dd)
140         continue
c-------------------------------------------------------------------------
            min=d(1)
            m=1
            do 145 j=2,p
            if(d(j).lt.min) then
            min=d(j)
            m=j
            end if
145         continue
c-------------------------------------------------------------------------
      if(n.gt.p) then
                  DO 146 I=1,Nmaj
            w(I,m)=(1-b)*w(I,m)+b*ss(I,k)
146               CONTINUE
            else
                  DO 147 I=1,Nmaj
            w(I,m)=ss(I,k)
147               CONTINUE
            n=n+1
            end if
125         continue
c-------------------------------------------------------------------------
            a=a/1.05
            b=b/1.05
            if(a.gt.0.1) go to 7
c-------------------------------------------------------------------------
c-----------------------  payan amoozesh *w*  ----------------------
c-------------------------------------------------------------------------
                        xx(1,1)=4.5
                        xx(2,1)=21.0
                        xx(3,1)=0.06
                        xx(4,1)=104.162
                        xx(5,1)=420.0
c                       xx(6,1)=1.5
c                       xx(7,1)=3.5
c                       xx(8,1)=0.1
c-------------------------------------------------------------------------
                        xx(1,2)=3.5
                        xx(2,2)=19.0
                        xx(3,2)=0.025
                        xx(4,2)=80.0
                        xx(5,2)=360.0
c                       xx(6,2)=0.95
c                       xx(7,2)=3.5
c                       xx(8,2)=-0.4
c-------------------------------------------------------------------------
                        xx(1,3)=3.95
                        xx(2,3)=15.9
                        xx(3,3)=0.07
                        xx(4,3)=142.0
```

```
                    xx(5,3)=355.0
c                   xx(6,3)=1.1
c                   xx(7,3)=3.25
c                   xx(8,3)=0.1
c-------------------------------------------------------------------
                    xx(1,4)=2.9
                    xx(2,4)=26.4
                    xx(3,4)=0.07
                    xx(4,4)=90.0
                    xx(5,4)=379.0
c                   xx(6,4)=1.2
c                   xx(7,4)=2.9
c                   xx(8,4)=0.0
c-------------------------------------------------------------------
                    xx(1,5)=5.6
                    xx(2,5)=21.4
                    xx(3,5)=0.04
                    xx(4,5)=80.8
                    xx(5,5)=480.0
c                   xx(6,5)=1.2
c                   xx(7,5)=3.5
c                   xx(8,5)=-0.3
c-------------------------------------------------------------------
                    xx(1,6)=3.9
                    xx(2,6)=29.0
                    xx(3,6)=0.06
                    xx(4,6)=38.0
                    xx(5,6)=428.2
c                   xx(6,6)=0.9
c                   xx(7,6)=3.1
c                   xx(8,6)=0
c-------------------------------------------------------------------
                    xx(1,7)=4.5
                    xx(2,7)=17.9
                    xx(3,7)=0.07
                    xx(4,7)=73.0
                    xx(5,7)=460.0
c                   xx(6,7)=1.09
c                   xx(7,7)=3.3
c                   xx(8,7)=-0.03
c-------------------------------------------------------------------
                    xx(1,8)=3.0
                    xx(2,8)=17.9
                    xx(3,8)=0.05
                    xx(4,8)=58.0
                    xx(5,8)=385.0
c                   xx(6,8)=1.0
c                   xx(7,8)=3.1
c                   xx(8,8)=-0.3
c-------------------------------------------------------------------
                    xx(1,9)=2.1
                    xx(2,9)=28.0
                    xx(3,9)=0.065
                    xx(4,9)=95.0
                    xx(5,9)=390.0
c                   xx(6,9)=1.2
c                   xx(7,9)=3.6
c                   xx(8,9)=0
c-------------------------------------------------------------------
                    xx(1,10)=4.8
                    xx(2,10)=18.5
                    xx(3,10)=0.1
                    xx(4,10)=101.0
                    xx(5,10)=540.0
```

```
c                   xx(6,10)=0.8
c                   xx(7,10)=2.8
c                   xx(8,10)=0
c-------------------------------------------------------------------------
                    xx(1,11)=0
                    xx(2,11)=0
                    xx(3,11)=0
                    xx(4,11)=0
                    xx(5,11)=0
c                   xx(6,11)=1.1
c                   xx(7,11)=3.3
c                   xx(8,11)=-0.1
c-------------------------------------------------------------------------
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c-------------------------------------------------------------------------
          do 155 k=1,2*Nmaj
          do 160 j=1,p
          do 165 i=1,Nmaj
          y(i,j)=xx(i,k)-v(i,j)
165       continue
160       continue
c-------------------------------------------------------------------------
          do 170 j=1,p
          dd=0
          do 169 i=1,Nmaj
          dd=dd+y(i,j)**2
169       continue
          d(j)=sqrt(dd)
170       continue
c??????????????????????????????????????????????????????????????????????????????????????????????????????
                    sk=0
                    do 175 j=1,p
                    if(d(j).Le.5.95705) then
                    sk=sk+d(j)
                    end if
175                 continue
c                   z(j)=(sk-d(j))/(sk*d(j))
                    do 17444 i=1,Nmaj
                    sF(i,k)=0
17444               continue
                    do 1744 j=1,p
                    if (d(j).Le.5.95705) then
                    do 1743 i=1,Nmaj
                    sF(i,k)=sF(i,k)+(sk-d(j))/(sk*d(j))*w(i,j)
                    SS(I,K)=SF(I,K)
1743                continue
                    else
          min=d(1)
          m=1
          do 1705 ji=2,p
          if(d(ji).lt.min) then
          min=d(ji)
          m=ji
          end if
1705      continue
                  DO 1706 I=1,Nmaj
                 ss(I,k)=w(I,m)
1706               CONTINUE
                   end if
1744               continue
155                continue
c-------------------------------------------------------------------------
c-------------------------------------------------------------------------
185                DO 177 I=1,Nmaj
```

```
                    xx(I,2*Nmaj+1)=0
177                 CONTINUE
c*************************************************************************
c-------------------------------------------------------------------
          do 190 i=1,Nmaj
          do 195 j=1,2*Nmaj
          xx(i,2*Nmaj+1)=xx(i,j)/(2*Nmaj)+xx(i,2*Nmaj+1)
195       continue
190       continue
c-------------------------------------------------------------------
          do 200 j=1,2*Nmaj+1
                    FF(J)=0
                    DO 199 N=1,NE
                    DO 201 i=1,NE
                    AREA(i,j)=xx(i,j)
201                 continue
                    I1=NOC(N,1)
                    I2=NOC(N,2)
                    I3=NOC(N,3)
                    I4=NOC(N,4)
                    X21=X(I2,1)-X(I1,1)
                    Y21=X(I2,2)-X(I1,2)
                    EL=SQRT(X21**2+Y21**2)
                    AL(N,J)=AREA(I4,I4)*EL
                    FF(J)=FF(J)+AL(N,J)
199                 continue
200       continue
c-------------------------------------------------------------------
          c=0
          do 205 j=1,2*Nmaj
          c=(ff(2*Nmaj+1)-ff(j))**2+c
205       continue
c-------------------------------------------------------------------
          if((1.0/(2*Nmaj)*c)**0.5.lt.5.85) go to 2000
          max=0
          do 210 j=1,2*Nmaj
          if(ff(j).gt.max) then
          max=ff(j)
          h=j
          end if
210   continue
c-------------------------------------------------------------------
C******* 0.066666666666=1.0/(2*MAJ-1) ***************************************
c******* 2*MAJ+2=?  *********************************************************
                    kj=2*Nmaj+2
          do 215 i=1,Nmaj
                    xx(i,kj)=0
                    do 214 j=1,2*Nmaj
                    xx(i,kj)=xx(i,kj)+xx(i,j)
214                 continue
                    xx(i,kj)=1.0/(2*Nmaj-1)*(xx(i,kj)-xx(i,h))
215                 continue
c-------------------------------------------------------------------
          a=1.3
1600   do 220 i=1,Nmaj
     xx(i,kj+1)=(1+a)*xx(i,kj)-a*xx(i,h)
220   continue
c-------------------------------------------------------------------
c*************************************************************************
          do 225 j=1,p
          do 230 i=1,Nmaj
          y(i,j)=xx(i,kj+1)-v(i,j)
230       continue
225       continue
```

```
c-----------------------------------------------------------------------
    do 235 j=1,p
          dd=0
           do 234 i=1,Nmaj
           dd=dd+y(i,j)**2
234        continue
           d(j)=sqrt(dd)
235        continue
c-----------------------------------------------------------------------
                    sk=0
                    do 1755 j=1,p
                    if(d(j).Lt.5.9575) then
                    sk=sk+d(j)
                    end if
1755                continue
c                   z(j)=(sk-d(j))/(sk*d(j))
                    do 17434 i=1,Nmaj
                    sF(i,kj+1)=0
17434               continue
                    do 1741 j=1,p
                    if (d(j).Lt.5.9575) then
                    do 1742 i=1,Nmaj
                    sF(i,kj+1)=sF(i,kj+1)+(sk-d(j))/(sk*d(j))*w(i,j)
                    SS(I,KJ+1)=SF(I,KJ+1)
1742                continue
                    else
           min=d(1)
           m=1
           do 1707 ji=2,p
           if(d(ji).lt.min) then
           min=d(ji)
           m=ji
          end if
1707       continue
                    DO 1708 I=1,Nmaj
                    ss(I,kj+1)=w(I,m)
1708                CONTINUE
                    end if
1741                continue
C********** 2*MAJ+3=?  ************************************************
                    FF(kj+1)=0
                    J=kj+1
                    DO 242 N=1,NE
                    DO 241 i= 1,NE
                    AREA(i,j)=xx(i,j)
241                 continue
                    I1=NOC(N,1)
                    I2=NOC(N,2)
                    I3=NOC(N,3)
                    I4=NOC(N,4)
                    X21=X(I2,1)-X(I1,1)
                    Y21=X(I2,2)-X(I1,2)
                    EL=SQRT(X21**2+Y21**2)
                    AL(N,J)=AREA(I4,J)*EL
                    FF(J)=FF(J)+AL(N,J)
242                 continue
                    ff(J-1)=ff(J)
c***************************************************************************
c***************************************************************************
c**********************    Geydhaye Masale    ************************
c******** Kontrole Taneshha !? ***********************************************
                    Narea=3
                    do 325 i=1,Narea
                    if (xx(i,kj+1).ge.0) go to 251
```

```
                    go to 261
251                 if (ss(i,kj+1)/13000-1.lt.0) go to 252
                            go to 261
252                 if((-ss(i,kj+1)/13000)-1.lt.0) go to 253
                            go to 261
c-------------------------------------------------------------------
c******** Kontrole Sharayete Hendessi !? ***********************************
253         if (xx(4,kj+1).ge.-200.0) go to 254
                            go to 261
254          if (xx(4,kj+1).le.200.0)  go to 256
                            go to 261
256                 if (xx(5,kj+1).ge.350.0)  go to 257
                            go to 261
257         if (xx(5,kj+1).le.550.0) go to 325
261     do 336 k=1,Nmaj
      xx(k,kj+1)=(xx(k,kj)+xx(k,kj+1))/2.0
336     continue
c*****************************************************************************
                    FF(kj+1)=0
                    DO 237 N=1,NE
                    j=kj+1
                    DO 233 i= 1,NE
                    AREA(i,j)=xx(i,j)
233                 continue
                    I1=NOC(N,1)
                    I2=NOC(N,2)
                    I3=NOC(N,3)
                    I4=NOC(N,4)
                    X21=X(I2,1)-X(I1,1)
                    Y21=X(I2,2)-X(I1,2)
                    EL=SQRT(X21**2+Y21**2)
                    AL(N,j)=AREA(I4,j)*EL
                    FF(j)=FF(j)+AL(N,j)
237                 continue
                    ff(j-1)=ff(j)
325                 continue
c-------------------------------------------------------------------
                    if(ff(h).gt.ff(kj)) go to 275
                    a=a/2.0
                    go to 1600
275                 do 239 i=1,Nmaj
                    xx(i,h)=xx(i,kj)
239                 continue
                    go to 185
2000    print*,'x1=',xx(1,kj-1),'x2=',xx(2,kj-1),'x3=',xx(3,kj-1),'x4=',xx(4,kj-1)
C             'x5=',xx(5,kj-1),'OBJ=', ff(kj-1)
                    end
```

پیوست شماره سه(متن کامل برنامه بهینه یابی خرپای مثال دوم با شبکه عصبی)

```
C Nmaj=Tedade Majhola
C Nzoj=Tedade Kole Zojhaye Amozesh
C NBW=Max(2*ABS(I-J)+2)
C NN=Tedade Gereh
C ND=Tedede Aksolamal Tekiegahi
C NE=Tedade Ozvha
C-----------------------------------------------------------------------
dimension v(Nmaj,Nzoj),w(Nmaj,Nzoj),xx(Nmaj,Nzoj),y(Nmaj,Nzoj),
c                d(Nzoj),ss(Nmaj,Nzoj),ff(2*Nmaj+3),L(NE),SF(Nmaj,Nzoj),
c   X(NN,2),NOC(NE,4),PM(1,2),AREA(Nzoj,Nzoj),NU(ND),stressp(NE),
c   REACT(ND),F(2*NN),SE(4,4),S(2*NN,NBW),AL(NE,2*Nmaj+3),U(ND)
C***********************************************************************
C***********************************************************************
          Print*,'Enter Nmaj,Nzoj,NE,NN,ND ?'
          Read (*,*)Nmaj,Nzoj,NE,NN,ND
C***********************************************************************
           E=21000000.0
           a=0.7
           b=0.1
           n=1
           p=1
C***********************************************************************
                    NQ=2*NN
                    NBW=0
c***********************************************************************
c---------------------------------- FAZAYE JOSTOJO ----------------------------------
          print*,'Enter fazaye jostojo x0(1),dxx(1), … ,x0(Nmaj),dxx(Nmaj) ?'
          read (*,*) (x0(i),dxx(i),i=1,Nmaj)
        do 5 k=1,Nzoj
          do 1 i=1,Nmaj
            xx(i,k)=x0(i)+k*dxx(i)
1      contiue
5      continue
c-----------------------------------------------------------------------
C*** N= SHOMAREH OZVE , 1=SHOMARE i , 2=SHOMAREH j, 3=1 ,4=AREA *******
C***************** NOC(N,i)=SHOMAREH (i)! , NOC(N,J)=? ******************
          Print*,'N=shomare ozve , 1=shomare i , 2=shomare j , 3=jens , 4=area'
          Print *,'enter NOC(N,1),NOC(N,2),NOC(N,3),NOC(N,4)'
          Read (*,*) (NOC(I,1),NOC(I,2),NOC(I,3),NOC(I,4),I=1,NE)
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------------------------------------------
C************* MOKHTASAT X=X(N,1) , Y=X(N,2) **** N=SHOMAREH GEREH********
c****************************** X(N,1)=X(gereh) , X(N,2)=Y(gereh)********
                    X(1,1)=0
                    X(1,2)=200.0
                    X(2,1)=xx(5,jL)
                    X(2,2)=xx(6,jL)
                    X(3,1)=xx(7,jL)
                    X(3,2)=xx(8,jL)
                    X(4,1)=0
                    X(4,2)=0
C                   X(5,1)=
C                   X(5,2)=
C                   X(6,1)=
C                   X(6,2)=
c-----------------------------------------------------------------------------------------------
C************* JAHATHAII KEH TEKIEHGAH AST U(I)=0************************
          print*,'Enter jahati keh tekiegah ast'
          print*,'NU(I)=? , U(I)=0'
          Read(*,*) (NU(i),U(i),i=1,ND)
```

```
c-----------------------------------------------------------------------
C***N=SHOMAREH GEREH , BARE VAREDEH DAR JAHATE U(I),U(I+1) ********
      Print*,'Enter N=shomare gereh , bare varede be gereh'
      read(*,*) N , F(2*N-1) , F(2*N)
C*************************************************************************
C*********************************************
c-----------------------------------------------------------------------
                  DO 6 I=1,Nmaj
                   v(I,p)=xx(I,1)
6                 CONTINUE
c-----------------------------------------------------------------------
7       do 10 k=2,Nzoj
        do 15 j=1,p
        do 20 i=1,Nmaj
        y(i,j)=xx(i,k)-v(i,j)
20      continue
15      continue
c-----------------------------------------------------------------------
        do 25 j=1,p
        dd=0
        do 24 i=1,Nmaj
        dd=dd+y(i,j)**2
24      continue
        d(j)=sqrt(dd)
25      continue
c-----------------------------------------------------------------------
        min=d(1)
        m=1
        do 30 j=1,p
        if(d(j).lt.min) then
        min=d(j)
        m=j
        end if
30      continue
c-----------------------------------------------------------------------
        if(d(m).gt.0.5) then
        p=p+1
        DO 31 I=1,Nmaj
        v(I,p)=xx(I,k)
31      CONTINUE
        else
        do 35 i=1,Nmaj
        v(i,m)=(1-a)*v(i,m)+a*xx(i,k)
35      continue
        end if
10      continue
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------
    do 40 jL=1,Nzoj
c*************************************************************************
                  DO 45 N=1,NE
                  C=2*(ABS(NOC(N,2)-NOC(N,1)))+2.0
                  IF(NBW.LT.C)THEN
                  NBW=C
                  END IF
45                CONTINUE
c*************************************************************************
                  do 44 i=1,NE
                  AREA(i,jL)=xx(i,jL)
44                CONTINUE
C*************************************************************************
C*****             DIMENSION S(NQ,NBW),F(NQ),SE(4,4) *************************
                  DO 50 I=1,NQ
                  F(NQ)=0
```

```
                DO 55 JJ=1,NBW
                S(I,JJ)=0
55              CONTINUE
50              CONTINUE
C*************************************************************************
C*************************************************************************
C********* E= BE TEDAD NM (KE MAMOLAN YEKI AST) ***************************
                E=21000000.0
                PM(1,1)=E
                PM(1,2)=0.0
C*************************************************************************
C*************************************************************************
C***************** SAKHTI KOLI ********************************************
                DO 60 N=1,NE
                I1=NOC(N,1)
                I2=NOC(N,2)
                I3=NOC(N,3)
                I4=NOC(N,4)
                X21=X(I2,1)-X(I1,1)
                Y21=X(I2,2)-X(I1,2)
                EL=SQRT(X21**2+Y21**2)
                EAL=E*AREA(I4,jL)/EL
                CS=X21/EL
                SN=Y21/EL
                SE(1,1)=CS*CS*EAL
                SE(I,2)=CS*SN*EAL
                SE(2,1)=SE(1,2)
                SE(1,3)=-CS*CS*EAL
                SE(3,1)=SE(1,3)
                SE(1,4)=-CS*SN*EAL
                SE(4,1)=SE(1,4)
                SE(2,2)=SN*SN*EAL
                SE(2,3)=-CS*SN*EAL
                SE(3,2)=SE(2,3)
                SE(2,4)=-SN*SN*EAL
                SE(4,2)=SE(2,4)
                SE(3,3)=CS*CS*EAL
                SE(3,4)=CS*SN*EAL
                SE(4,3)=SE(3,4)
                SE(4,4)=SN*SN*EAL
                DO 65 II=1,2
                NRT=2*(NOC(N,II)-1)
                DO 70 IT=1,2
                NR=NRT+IT
                I=2*(II-1)+IT
                DO 75 JJ=1,2
                NCT=2*(NOC(N,JJ)-1)
                DO 80 JT=1,2
                J=2*(JJ-1)+JT
                NC=NCT+JT-NR+1
                IF(NC.LE.0) THEN
                  GO TO 80
                END IF
                S(NR,NC)=S(NR,NC)+SE(I,J)
80              CONTINUE
75              CONTINUE
70              CONTINUE
65              CONTINUE
60              CONTINUE
C*************************************************************************
C******** SHARAYET MARZII *********************************************
                CNST=S(1,1)*10000
                DO 85 I=1,ND
                N=NU(I)
```

```
                S(N,1)=S(N,1)+CNST
                F(N)=F(N)+CNST*U(I)
85              CONTINUE
C*************************************************************************
C********** HALLE MOADELAT ***********************************************
                N1=NQ-1
                DO 90 K=1,N1
                NK=NQ-K+1
                IF(NK.GT.NBW) THEN
                 NK=NBW
                END IF
                DO 95 I=2,NK
                C1=S(K,I)/S(K,1)
                I1=K+I-1
                DO 100 J=I,NK
                J1=J-I+1
                S(I1,J1)=S(I1,J1)-C1*S(K,J)
100             CONTINUE
                F(I1)=F(I1)-C1*F(K)
95              CONTINUE
90              CONTINUE
                F(NQ)=F(NQ)/S(NQ,1)
                DO 105 KK=1,N1
                K=NQ-KK
                C1=1/S(K,1)
                F(K)=C1*F(K)
                NK=NQ-K+1
                IF(NK.GT.NBW)THEN
                        NK=NBW
                END IF
                DO 110 J=2,NK
                F(K)=F(K)-C1*S(K,J)*F(K+J-1)
110             CONTINUE
105             CONTINUE
C*************************************************************************
C************************** STRESS ***************************************
                DO 115 I=1,NE
                I1=NOC(I,1)
                I2=NOC(I,2)
                I3=NOC(I,3)
                X21=X(I2,1)-X(I1,1)
                Y21=X(I2,2)-X(I1,2)
                EL=SQRT(X21**2+Y21**2)
                CS=X21/EL
                SN=Y21/EL
                J2=2*I1
                J1=J2-1
                K2=2*I2
                K1=K2-1
                DT=(F(K1)-F(J1))*CS+(F(K2)-F(J2))*SN
                STRESS=DT*PM(I3,1)/EL
                STRESSP(I)=STRESS
115             CONTINUE
c-------------------------------------------------------------------------
        do 111 i=1,NE
         ss(i,jL)=stressp(i)
111      continue
        do 112 i=NE+1,Nmaj
         ss(i,jL)=stressp(i)
112      contiue
c*********************
c***********
40   continue
c-------------------------------------------------------------------------
```

```
c*************************************************************************
      do 125 k=1,Nzoj
            do 130 j=1,p
            do 135 i=1,Nmaj
            y(i,j)=xx(i,k)-v(i,j)
135         continue
130         continue
c-------------------------------------------------------------------------
            do 140 j=1,p
           dd=0
            do 139 i=1,Nmaj
            dd=dd+y(i,j)**2
139         continue
            d(j)=sqrt(dd)
140         continue
c-------------------------------------------------------------------------
            min=d(1)
            m=1
            do 145 j=2,p
            if(d(j).lt.min) then
            min=d(j)
            m=j
           end if
145         continue
c-------------------------------------------------------------------------
      if(n.gt.p) then
           DO 146 I=1,Nmaj
            w(I,m)=(1-b)*w(I,m)+b*ss(I,k)
146         CONTINUE
            else
           DO 147 I=1,Nmaj
            w(I,m)=ss(I,k)
147         CONTINUE
            n=n+1
            end if
125         continue
c-------------------------------------------------------------------------
            a=a/1.05
            b=b/1.05
            if(a.gt.0.1) go to 7
c-------------------------------------------------------------------------
c-------------------------  payan amoozesh *w*  ----------------------
c-------------------------------------------------------------------------
                        xx(1,1)=17.5
                        xx(2,1)=4.4
                        xx(3,1)=12.8
                        xx(4,1)=3.7
                        xx(5,1)=140.0
                        xx(6,1)=130.0
                        xx(7,1)=180.0
                        xx(8,1)=60.0
c-------------------------------------------------------------------------
                        xx(1,2)=20.5
                        xx(2,2)=5.5
                        xx(3,2)=13.6
                        xx(4,2)=4.0
                        xx(5,2)=110.0
                        xx(6,2)=85.0
                        xx(7,2)=200.0
                        xx(8,2)=55.0
c-------------------------------------------------------------------------
                       xx(1,3)=19.5
                       xx(2,3)=4.9
                       xx(3,3)=16.5
```

```
      xx(4,3)=3.5
      xx(5,3)=95.0
      xx(6,3)=120.0
      xx(7,3)=200.0
      xx(8,3)=60.0
c-----------------------------------------------------------------------
      xx(1,4)=25.0
      xx(2,4)=5.0
      xx(3,4)=17.0
      xx(4,4)=3.10
      xx(5,4)=150.0
      xx(6,4)=95.0
      xx(7,4)=205.0
      xx(8,4)=48.0
c-----------------------------------------------------------------------
      xx(1,5)=16.0
      xx(2,5)=7.0
      xx(3,5)=21.0
      xx(4,5)=3.2
      xx(5,5)=170.0
      xx(6,5)=120.0
      xx(7,5)=205.0
      xx(8,5)=60.0
c-----------------------------------------------------------------------
      xx(1,6)=18.2
      xx(2,6)=4.9
      xx(3,6)=17.5
      xx(4,6)=7.0
      xx(5,6)=100.0
      xx(6,6)=130.0
      xx(7,6)=201.0
      xx(8,6)=49.0
c-----------------------------------------------------------------------
      xx(1,7)=23.0
      xx(2,7)=4.4
      xx(3,7)=20.6
      xx(4,7)=6.5
      xx(5,7)=180.0
      xx(6,7)=109.0
      xx(7,7)=200.0
      xx(8,7)=75.0
c-----------------------------------------------------------------------
      xx(1,8)=21.0
      xx(2,8)=8.0
      xx(3,8)=15.5
      xx(4,8)=7.0
      xx(5,8)=145.0
      xx(6,8)=100.0
      xx(7,8)=210.0
      xx(8,8)=30.0
c-----------------------------------------------------------------------
      xx(1,9)=19.5
      xx(2,9)=9.4
      xx(3,9)=16.0
      xx(4,9)=3.4
      xx(5,9)=98.0
      xx(6,9)=120.0
      xx(7,9)=200.0
      xx(8,9)=100.0
c-----------------------------------------------------------------------
      xx(1,10)=24.0
      xx(2,10)=4.2
      xx(3,10)=18.0
      xx(4,10)=5.5
```

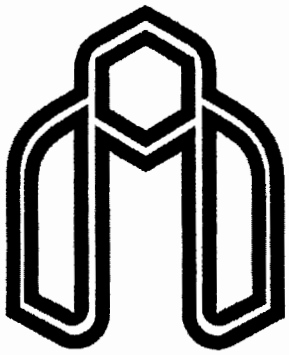
```
      xx(5,10)=105.0
      xx(6,10)=100.0
      xx(7,10)=200.0
      xx(8,10)=50.0
c-----------------------------------------------------------------------
      xx(1,11)=15.5
      xx(2,11)=8.1
      xx(3,11)=20.5
      xx(4,11)=4.0
      xx(5,11)=160.0
      xx(6,11)=130.0
      xx(7,11)=175.0
      xx(8,11)=75.0
c-----------------------------------------------------------------------
      xx(1,12)=17.7
      xx(2,12)=7.9
      xx(3,12)=17.0
      xx(4,12)=5.1
      xx(5,12)=110.0
      xx(6,12)=130.0
      xx(7,12)=200.0
      xx(8,12)=70.0
c-----------------------------------------------------------------------
      xx(1,13)=23.5
      xx(2,13)=4.2
      xx(3,13)=19.0
      xx(4,13)=5.7
      xx(5,13)=100.0
      xx(6,13)=130.0
      xx(7,13)=175.0
      xx(8,13)=95.0
c-----------------------------------------------------------------------
      xx(1,14)=23.5
      xx(2,14)=7.2
      xx(3,14)=14.11
      xx(4,14)=4.8
      xx(5,14)=180.0
      xx(6,14)=89.0
      xx(7,14)=195.0
      xx(8,14)=88.0
c-----------------------------------------------------------------------
      xx(1,15)=23.0
      xx(2,15)=6.0
      xx(3,15)=18.0
      xx(4,15)=5.4
      xx(5,15)=95.0
      xx(6,15)=120.0
      xx(7,15)=185.0
      xx(8,15)=65.0
c-----------------------------------------------------------------------
      xx(1,16)=17.0
      xx(2,16)=4.8
      xx(3,16)=19.0
      xx(4,16)=3.9
      xx(5,16)=168.0
      xx(6,16)=95.0
      xx(7,16)=200.0
      xx(8,16)=60.0
c-----------------------------------------------------------------------
      xx(1,18)=0
      xx(2,18)=0
      xx(3,18)=0
      xx(4,18)=0
      xx(5,18)=0
```

```
                    xx(6,18)=0
                    xx(7,18)=0
                    xx(8,18)=0
c-----------------------------------------------------------------------
                    xx(1,19)=0
                    xx(2,19)=0
                    xx(3,19)=0
                    xx(4,19)=0
                    xx(5,19)=0
                    xx(6,19)=0
                    xx(7,19)=0
                    xx(8,19)=0
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c-----------------------------------------------------------------------
          do 155 k=1,2*Nmaj
          do 160 j=1,p
          do 165 i=1,Nmaj
          y(i,j)=xx(i,k)-v(i,j)
165       continue
160       continue
c-----------------------------------------------------------------------
          do 170 j=1,p
          dd=0
          do 169 i=1,Nmaj
          dd=dd+y(i,j)**2
169       continue
          d(j)=sqrt(dd)
170       continue
c-----------------------------------------------------------------------
                    sk=0
                    do 175 j=1,p
                    if(d(j).Le.5.95705) then
                    sk=sk+d(j)
                    end if
175                 continue
                    do 17444 i=1,Nmaj
                    sF(i,k)=0
17444               continue
                    do 1744 j=1,p
                    if (d(j).Le.5.95705) then
                    do 1743 i=1,Nmaj
                    sF(i,k)=sF(i,k)+(sk-d(j))/(sk*d(j))*w(i,j)
                    SS(I,K)=SF(I,K)
1743                continue
                    else
                     min=d(1)
                     m=1
                     do 1705 ji=2,p
                     if(d(ji).lt.min) then
                     min=d(ji)
                     m=ji
                    end if
1705                continue
                     DO 1706 I=1,Nmaj
                     ss(I,k)=w(I,m)
1706                 CONTINUE
                     end if
1744                 continue
155                  continue
c-----------------------------------------------------------------------
185                  DO 177 I=1,Nmaj
                    xx(I,2*Nmaj+1)=0
177                  CONTINUE
c-----------------------------------------------------------------------
```

```
            do 190 i=1,Nmaj
            do 195 j=1,2*Nmaj
            xx(i,2*Nmaj+1)=xx(i,j)/(2*Nmaj)+xx(i,2*Nmaj+1)
195         continue
190         continue
c-------------------------------------------------------------------------
            do 200 j=1,2*Nmaj+1
                    FF(J)=0
                    DO 199 N=1,NE
                    DO 201 i=1,NE
                    AREA(i,j)=xx(i,j)
201                 continue
                    I1=NOC(N,1)
                    I2=NOC(N,2)
                    I3=NOC(N,3)
                    I4=NOC(N,4)
                    X21=X(I2,1)-X(I1,1)
                    Y21=X(I2,2)-X(I1,2)
                    EL=SQRT(X21**2+Y21**2)
                    AL(N,J)=AREA(I4,I4)*EL
                    FF(J)=FF(J)+AL(N,J)
199                 continue
200   continue
c-------------------------------------------------------------------------
            c=0
            do 205 j=1,2*Nmaj
            c=(ff(2*Nmaj+1)-ff(j))**2+c
205         continue
c-------------------------------------------------------------------------
            if((1.0/(2*Nmaj)*c)**0.5.lt.300.5) go to 2000
            max=0
            do 210 j=1,2*Nmaj
            if(ff(j).gt.max) then
            max=ff(j)
            h=j
            end if
210   continue
c-------------------------------------------------------------------
c******* 2*MAJ+2=?  ***************************************************
                    kj=2*Nmaj+2
            do 215 i=1,Nmaj
                    xx(i,kj)=0
                    do 214 j=I,2*Nmaj
                    xx(i,kj)=xx(i,kj)+xx(i,j)
214                 continue
                    xx(i,kj)=1.0/(2*Nmaj-1)*(xx(i,kj)-xx(i,h))
215                 continue
c-------------------------------------------------------------------------
            a=1.3
1600   do 220 i=1,Nmaj
     xx(i,kj+1)=(1+a)*xx(i,kj)-a*xx(i,h)
220   continue
c-------------------------------------------------------------------
c*************************************************************************
            do 225 j=1,p
            do 230 i=1,Nmaj
            y(i,j)=xx(i,kj+1)-v(i,j)
230         continue
225         continue
c-------------------------------------------------------------------------
            do 235 j=1,p
               dd=0
            do 234 i=1,Nmaj
            dd=dd+y(i,j)**2
```

```
234       continue
          d(j)=sqrt(dd)
235       continue
c-------------------------------------------------------------------------
                    sk=0
                    do 1755 j=1,p
                    if(d(j).Lt.5.9575) then
                    sk=sk+d(j)
                    end if
1755                continue
c                   z(j)=(sk-d(j))/(sk*d(j))
                    do 17434 i=1,Nmaj
                    sF(i,kj+1)=0
17434               continue
                    do 1741 j=1,p
                    if (d(j).Lt.5.9575) then
                    do 1742 i=1,Nmaj
                    sF(i,kj+1)=sF(i,kj+1)+(sk-d(j))/(sk*d(j))*w(i,j)
                    SS(I,KJ+1)=SF(I,KJ+1)
1742                continue
                    else
                   min=d(1)
                   m=1
                  do 1707 ji=2,p
                  if(d(ji).lt.min) then
                  min=d(ji)
                  m=ji
                  end if
1707              continue
                    DO 1708 I=1,Nmaj
                   ss(I,kj+1)=w(I,m)
1708                CONTINUE
                    end if
1741                continue
C********** 2*MAJ+3=?  ************************************************
                    FF(kj+1)=0
                    J=kj+1
                    DO 242 N=1,NE
                    DO 241 i= 1,NE
                    AREA(i,j)=xx(i,j)
241                 continue
                    I1=NOC(N,1)
                    I2=NOC(N,2)
                    I3=NOC(N,3)
                    I4=NOC(N,4)
                    X21=X(I2,1)-X(I1,1)
                    Y21=X(I2,2)-X(I1,2)
                    EL=SQRT(X21**2+Y21**2)
                    AL(N,J)=AREA(I4,J)*EL
                    FF(J)=FF(J)+AL(N,J)
242                 continue
                    ff(J-1)=ff(J)
c*********************    Geydhaye Masale    *************************
                    Narea=4
                    do 325 i=1,Narea
                    if (xx(i,kj+1).ge.0) go to 251
                       go to 261
251                 if (ss(i,kj+1)/13000-1.lt.0) go to 252
                             go to 261
252                 if((-ss(i,kj+1)/13000)-1.lt.0) go to 253
                             go to 261
c*************************************************************************
253                 if (xx(5,kj+1).ge.80.0) go to 254
                             go to 261
```

```
254                   if (xx(5,kj+1).le.250.0)  go to 256
                            go to 261
256                   if (xx(6,kj+1).ge.50.0)  go to 257
                            go to 261
257                   if (xx(6,kj+1).le.200.0) go to 258
                            go to 261
258                   if (xx(7,kj+1).ge.150.0) go to 259
                            go to 261
259                   if (xx(7,kj+1).le.350.0)  go to 267
                            go to 261
267                   if (xx(8,kj+1).ge.-100.0) go to 268
                            go to 261
268                   if (xx(8,kj+1).le.100.0)  go to 325
261      do 336 k=1,Nmaj
       xx(k,kj+1)=(xx(k,kj)+xx(k,kj+1))/2.0
336      continue
c*************************************************************************
                      FF(kj+1)=0
                      DO 237 N=1,NE
                      j=kj+1
                      DO 233 i= 1,NE
                      AREA(i,j)=xx(i,j)
233                   continue
                      I1=NOC(N,1)
                      I2=NOC(N,2)
                      I3=NOC(N,3)
                      I4=NOC(N,4)
                      X21=X(I2,1)-X(I1,1)
                      Y21=X(I2,2)-X(I1,2)
                      EL=SQRT(X21**2+Y21**2)
                      AL(N,j)=AREA(I4,j)*EL
                      FF(j)=FF(j)+AL(N,j)
237                   continue
                      ff(j-1)=ff(j)
325                  continue
c-------------------------------------------------------------------------
                      if(ff(h).gt.ff(kj)) go to 275
                      a=a/2.0
                      go to 1600
275                   do 285 i=1,Nmaj
                      xx(i,h)=xx(i,kj)
285                   continue
                      go to 185
2000     print*,'x1=',xx(1,kj-1),'x2=',xx(2,kj-1),'x3=',xx(3,kj-1),'x4=',xx(4,kj-1),
   c   'x5=',xx(5,kj-1),'x6=',xx(6,kj-1),'x7=',xx(7,kj-1),'x8=',xx(8,kj-1),'OBJ=',ff(kj-1)
                      end
```



Shahrood University Of Technology

College Of Civil Engineering

Topic :

Truss Optimization With Artificial Neural Networks Application

By :

Mehdi Jamaklou

A Thesis is submitted in partial fulfillment of the requirement for the degree of Master Of Science in Structural Engineering .

Adviser :

Dr . M . Talebzadeh

Abstract :

This thesis discuss about Artificial Neural Networks and neural computing . We use neural computing for truss analysis and truss optimization after neural network training. Neural network training was did with patterns sets to learning truss analysis and preparing to truss optimization .

Two Artificial neural network was studied in this research that one of them is Back propagation network and another one is Counterpropagation .

Counterpropagation neural network was developed , trained , tested and readied for truss analysis and optimize. The out put of Counterpropagation network evaluated with Complex exactly method out put and we were improved that neural network output is desirable .

At the end , this study presents the effort in applying neural network techniques to truss analysis and optimizes .

Key word : neural network – truss – optimization .