

مهندسی شبکه های عصبی توسط اتوماتانهای یادگیر: تعیین اندازه مطلوب برای شبکه های عصبی سه لایه

محمد رضا میبدی

دانشیار دانشکده مهندسی کامپیوتر - دانشگاه صنعتی امیر کبیر

حمید بیگی

دانشجوی دکترای دانشکده مهندسی کامپیوتر - دانشگاه صنعتی امیر کبیر

(تاریخ دریافت ۷۹/۳/۲۵، تاریخ تصویب ۷۹/۱۰/۳)

چکیده

هدف از مهندسی شبکه های عصبی بررسی معایب و مزایای شبکه های عصبی مصنوعی و آرایه روشهایی برای بهبود کارایی آنهاست. یکی از موضوعات مورد بحث در مهندسی شبکه های عصبی چندلایه، یافتن ساختار مناسب (نزدیک به بهینه) برای حل مسئله می باشد. معیار و نحوه انتخاب اندازه شبکه عصبی برای یک مسئله خاص هنوز شناخته شده نیست. در روشهای کلاسیک، طراح شبکه در ابتدای آموزش ساختاری را برای شبکه تعیین و سپس شبکه را آموزش میدهد و ساختار تعیین شده در حین آموزش ثابت نگاه داشته میشود. تعیین اندازه شبکه عصبی (تعداد واحدها و لایه های مخفی) تاثیر عمده ای بر روی کارایی شبکه های عصبی خواهد داشت. طراحی یک شبکه با ساختار بهینه یک مسئله ذاتا مشکل است. بهمین جهت بیشتر الگوریتم های آرایه شده برای تعیین ساختار شبکه های عصبی، الگوریتم های تقریبی هستند. این الگوریتم ها را میتوان به پنج گروه عمده الگوریتم های هرس، الگوریتم های سازنده، الگوریتم های ترکیبی، الگوریتم های تکاملی و الگوریتم های بر اساس اتوماتانهای یادگیر تقسیم کرد. تنها الگوریتم گزارش شده بر اساس اتوماتان یادگیر، الگوریتم بقا نام دارد که توسط بیگی و میبدی آرایه گردیده است. این الگوریتم از یک اتوماتان یادگیر مهاجرت اشیا (بعنوان یک ابزار جستجوی عمومی) و الگوریتم یادگیری انتشار خطا به عقب استفاده میکند و در ضمن آموزش، تعداد واحدهای مخفی یک شبکه سه لایه را تعیین مینماید. در این مقاله سه الگوریتم بر اساس اتوماتانهای یادگیر مهاجرت اشیا و الگوریتم یادگیری انتشار خطا به عقب آرایه شده است که ساختار مناسب شبکه عصبی سه لایه (ساختاریکه دارای اندازه کوچک، پیچیدگی آموزش کم و قدرت تعمیم بالا باشد) را برای یک مجموعه از الگوهای آموزشی تعیین می نماید. اغلب روشهای گزارش شده برای تعیین ساختار شبکه از الگوریتم های کوهنوردی استفاده میکنند و مشکل گرفتاری در حداقل محلی را دارند. در روش پیشنهادی در این مقاله بدلیل استفاده از روشهای جستجوی عمومی، امکان گرفتاری در حداقل های محلی فضای ساختارها کاهش می یابد. الگوریتم های آرایه شده روی مسائل متنوعی از جمله شناسایی اعداد فارسی و شناسایی اعداد انگلیسی آزمایش شده اند. از طریق شبیه سازی نشان داده شده است که شبکه های تولید شده توسط الگوریتم های پیشنهادی دارای کارایی بالایی هستند. همچنین الگوریتم های پیشنهادی با الگوریتم هرس کارنین مقایسه گردیده است. نتایج الگوریتم های پیشنهادی هم از لحاظ ساختار تولید شده، هم ز نظر زمان مورد نیاز برای تعیین ساختار شبکه عصبی، بسیار بهتر از الگوریتم هرس کارنین عمل کرده اند.

واژه های کلیدی: مهندسی شبکه های عصبی، شبکه های عصبی چند لایه، ساختار شبکه های عصبی، اتوماتانهای یادگیر

مهاجرت اشیا

مقدمه

شبکه را آموزش میدهد. ساختار تعیین شده در حین آموزش ثابت نگاه داشته میشود. تعیین تعداد واحدهای لایه مخفی و وزنهای آن تاثیر بسزایی بر روی کارایی شبکه دارد. یک شبکه با اندازه کوچک، خروجی دقیق تولید نمیکند و شبکه ای با اندازه بزرگ، بسیار کند و پرهزینه خواهد بود و برای تعمیم مجموعه آموزشی، نیاز به مجموعه آموزشی بزرگی خواهد

هدف از مهندسی شبکه های عصبی بررسی معایب و مزایای شبکه های عصبی مصنوعی و آرایه روشهایی برای بهبود کارایی آنهاست. یکی از موضوعات مورد بحث در شبکه های عصبی چندلایه یافتن ساختار مناسب (نزدیک به بهینه) برای حل مسئله می باشد. در روشهای کلاسیک، طراح شبکه در ابتدای آموزش ساختاری را برای شبکه تعیین و سپس

داشت. طراحی یک شبکه با ساختار بهینه یک مسئله NP-Hard است [۱]. بهمین جهت بیشتر الگوریتمهای ارائه شده برای تعیین ساختار شبکه‌های عصبی، الگوریتمهای تقریبی هستند. این الگوریتم‌ها قبل، درحین یا بعد از یادگیری ساختار مناسب برای شبکه را تعیین می نمایند. بعضی از این الگوریتم‌ها از اطلاعات محلی و بعضی دیگر از اطلاعات عمومی برای یافتن ساختار مناسب شبکه استفاده میکنند. این الگوریتم‌ها را میتوان به پنج گروه عمده زیر تقسیم کرد.

الف) الگوریتم‌های هرس^۱: این الگوریتم‌ها از یک شبکه بزرگ شروع نموده و بتدریج درحین آموزش یا بعد از آن واحد‌ها یا وزنهای اضافی را از شبکه هرس می کنند. در الگوریتم‌های هرس نیاز است که تعداد واحد‌های مخفی را در ابتدای آموزش مشخص نماییم. البته این مشکل حادی را ایجاد نمیکند زیرا کران بالای تعداد واحد‌های مخفی برای یک مسئله معین و برای بعضی از انواع شبکه‌ها مشخص است [۲]-[۹].

ب) الگوریتم‌های سازنده^۲: این الگوریتم‌ها با یک شبکه کوچک شروع به آموزش و بتدریج درحین آموزش شبکه، واحد یا لایه مخفی به شبکه میافزایند. این الگوریتم‌ها معمولاً شبکه‌های کوچک تولید میکنند که دارای پیچیدگی آموزش بالا هستند [۱۰]-[۱۸].

ج) الگوریتم‌های ترکیبی: این گروه از الگوریتم‌ها، از ترکیبی از الگوریتم‌های سازنده و الگوریتم‌های هرس برای تعیین ساختار شبکه استفاده میکنند. در این الگوریتم‌ها برای رسیدن به شبکه مطلوب میتوان وزن، واحد یا لایه مخفی را کم یا زیاد نمود [۱۹]-[۲۰].

د) الگوریتم‌های تکاملی^۳: در این الگوریتم‌ها تعیین ساختار بهینه برای شبکه از طریق جستجو در فضای ساختارها انجام میگردد. هر نقطه از این فضا نماینده یک ساختار شبکه است. الگوریتم جستجو با استفاده از یک معیار کارایی مانند حداقل خطا و یا پیچیدگی آموزش بدنبال مناسبترین ساختار میباشد [۲۱]-[۲۵].

ه) الگوریتم‌های بر اساس اتوماتانهای یادگیر^۴: تنها الگوریتم گزارش شده که بر اساس اتوماتان‌های یادگیر

طراحی شده است الگوریتم بقا نام دارد که توسط پیگی و میبیدی [۲۶] [۲۷] ارائه گردیده است. این الگوریتم از یک اتوماتان یادگیر مهاجرت اشیا^۵ بعنوان یک ابزار جستجوی عمومی و الگوریتم یادگیری انتشار خطابه عقب [۲۸] استفاده میکند و در ضمن آموزش، ساختار مناسبی برای شبکه عصبی سه لایه (ساختار یکه دارای اندازه کوچک، پیچیدگی آموزش کم و قدرت تعمیم بالا باشد) تعیین مینماید. در الگوریتم بقا، آموزش از یک شبکه عصبی سه لایه بزرگ شروع شده و اتوماتان یادگیر با افزودن و کاستن واحدهای مخفی، تعداد واحدهای لایه مخفی این شبکه را تعیین میکند. اغلب روشهای گزارش شده برای تعیین ساختار شبکه از الگوریتم‌های کوهنوردی استفاده میکنند [۲۱] و مشکل گرفتاری در حداقل محلی را دارند. در الگوریتم بقا بدلیل استفاده از روشهای جستجوی عمومی (اتوماتانهای یادگیر)، امکان گرفتاری در حداقل محلی فضای ساختارها کاهش مییابد. الگوریتم بقا بعد از این مقاله بطور خلاصه شرح داده خواهد شد.

برای اطلاعات بیشتر در مورد الگوریتم‌های فوق می‌توانید به مرجع [۲۹] مراجعه نمایید.

در این مقاله سه الگوریتم بر اساس اتوماتانهای یادگیر مهاجرت اشیا و الگوریتم یادگیری انتشار خطا به عقب ارائه شده است که ساختار مناسب برای شبکه عصبی سه لایه را برای یک مجموعه از الگوهای آموزشی تعیین می نماید. الگوریتم‌های ارائه شده دارای دو قسمت میباشند: تعیین تعداد واحدهای مخفی مورد نیاز و تعیین تعداد وزنهای مورد نیاز شبکه. در الگوریتم ۱، برای تعیین تعداد واحد‌های مخفی از الگوریتم بقا استفاده گردیده است. الگوریتم بقا تعداد واحد‌های لایه میانی شبکه را که قدرت تعمیم قابل قبولی را ارائه و قادر به مدلسازی مسئله گردد تعیین می کند. برای تعیین تعداد وزنهای الگوریتم جدیدی با استفاده از اتوماتان یادگیری مهاجرت اشیا طراحی گردیده است. این الگوریتم با حذف وزنهایی که دارای اثر کمتری هستند نه تنها باعث کاهش پیچیدگی شبکه خواهد شد بلکه افزایش تعمیم شبکه را هم بدنبال خواهد داشت. الگوریتم تعیین وزنهای با خاموش و روشن کردن وزنهای سعی در پیدا کردن وزنهای مناسب مینماید. در دو الگوریتم دیگر

افزایش تعداد لبه های موجود در مرز نواحی محدب تولید شده توسط این لایه میگردد. بنابر این تعداد لبه های موجود در مرز هر ناحیه حداکثر برابر با تعداد واحد های لایه مخفی اول است. بسیاری از این ابرصفحه ها تقریباً موازی و نزدیک بهم هستند و اطلاعات تقریباً یکسانی در وزنه های آنها ذخیره شده است که می توانند براحتی حذف شوند. اگر یک لایه دیگر به شبکه اضافه شود واحد های لایه جدید با استفاده از نواحی محدب تولید شده توسط واحد های لایه مخفی دوم، ترکیبی از این نواحی را (اشتراک، اجتماع و ...) تولید میکنند که الزاماً محدب نیستند و میتوانند هر شکل دلخواهی به خود بگیرند. پیچیدگی شکل ناحیه های تولید شده به تعداد واحد های موجود در هر لایه مخفی از شبکه بستگی دارد. بنابراین با استفاده از یک شبکه عصبی چهار لایه (با احتساب لایه ورودی) میتوان ناحیه هایی با هر شکلی را تولید نمود. با توجه به اینکه مرز این نواحی همان تابع تمایز میباشد، در نتیجه تعداد لایه های مخفی و تعداد واحد در هر لایه مخفی از شبکه، شکل تابع تمایز را مشخص میکند. اگر تابع فعالیت واحدها، تابع پله باشد مرز اتصال بین دوا بر صفحه در هر ناحیه شکسته میگردد مانند رئوس مربع یا مثلث. برای رفع شکستگی در مرز نواحی میتوان از تابع سیگموئید برای تابع فعالیت واحدها استفاده کرد. استفاده از تابع سیگموئید برای تابع فعالیت واحدها باعث هموار شدن مرز نواحی تولید شده میگردد هر چند شکل ناحیه ها ثابت باقی میماند [۳۰]. تعداد لایه های مخفی، تعداد واحدهای مخفی در هر لایه و نوع اتصال وزنها، شکل نواحی تولید شده توسط شبکه عصبی را مشخص میکنند. اگر تعداد واحدهای مخفی زیاد باشد تعداد ابرصفحه های موجود در مرز تشکیل دهنده نواحی زیادتر میشوند. وزنه های متصل به یک واحد مخفی وابستگی ابر صفحه تولید شده را به ویژگیهای الگوهای ورودی تعیین میکنند و یا به عبارتی دیگر تعداد وزنه های ورودی به هر واحد مخفی، ابعاد ابر صفحه تولید شده توسط این واحد را مشخص میکند.

معیارهای کارایی شبکه

در این قسمت رابطه پیچیدگی آموزش و قدرت

هدف فقط تعیین تعداد وزنه های لایه مخفی شبکه است و تعداد واحدهای لایه مخفی را از طریق تعیین تعداد وزنه های شبکه مشخص می گردد. این دو الگوریتم با بررسی های محلی یا عمومی وزنه های لایه مخفی سعی در پیدا کردن ساختار مناسب برای شبکه را دارند. بخشهای بعدی مقاله بصورت زیر سازماندهی شده است. رابطه بین ساختار شبکه های عصبی چند لایه و شکل تابع تمایز در بخش ۲ مورد مطالعه قرار گرفته است. این مطالعه در بررسی الگوریتمهای ارائه شده در این مقاله مورد استفاده قرار خواهد گرفت. سپس در بخش ۳ معیارهای کارایی شبکه را بررسی مینماییم. مقدمه ای بر اتوماتانهای یادگیر و الگوریتم بقا در بخش ۴ آمده است. الگوریتمهای پیشنهادی در بخش ۵ بیان شده اند. نتایج آزمایشها برای مسائل مختلف در بخش ۶ و در پایان نتیجه گیری آورده شده است.

رابطه بین ساختار شبکه عصبی و شکل تابع تمایز

یکی از روشهای تعیین تابع تمایز، استفاده از شبکه عصبی چند لایه است. یک شبکه عصبی چند لایه قادر است تا هر نگاشت غیرخطی (تابع تمایز) را از روی نمونه های آموزشی تقریب بزند. کیفیت دسته بندی، تعداد و شکل تابع تمایز تعیین شده توسط این روش بستگی به تعداد لایه های مخفی، تعداد واحد هادر هر لایه مخفی، شکل تابع فعالیت واحدها، وزنها و غیره دارد.

در یک شبکه عصبی چند لایه با n ورودی $X = \{x_1, x_2, \dots, x_n\}$ را در که واحدهای آن دارای تابع فعالیت $f(\cdot)$ باشند هر واحد در لایه مخفی اول یک ابر صفحه با معادله $P_k = f(\sum_{i=1}^n x_i w_{ik} + b_k)$ تولید میکنند که w_{ik} وزن اتصال بین واحد ورودی x_i و واحد مخفی شماره k مقدار آستانه این واحد میباشد. واحد های موجود در لایه مخفی دوم با استفاده از ابرصفحه های تولید شده توسط واحد های لایه مخفی اول، نواحی محدبی تولید میکنند که ممکن است باز یا بسته باشند مانند ناحیه داخل مربع یا مثلث یا ناحیه بین دو خط موازی. اضافه نمودن واحد به لایه مخفی اول سبب

تعمیم شبکه با ساختار شبکه های عصبی را مورد بررسی قرار میدهم.

پیچیدگی آموزش شبکه: پیچیدگی آموزش شبکه در واقع همان پیچیدگی زمانی الگوریتم آموزش میباشد که بستگی به کاربرد مورد نظر، ساختار شبکه عصبی و الگوریتم یادگیری دارد. پیدا نمودن وزنه های مناسب برای شبکه برای یک مسئله مشخص یک مسئله ذاتا مشکل میباشد و در گروه مسایل NP-Complete قرار دارد [۳۱]. بهمین دلیل برای تعیین وزنه های شبکه بمنظور حصول به یک نگاشت صحیح، الگوریتم های تقریبی پیشنهاد شده است که الگوریتم یادگیری انتشار خطا به عقب که براساس جستجو در امتداد گرادیان بنا شده است یک نمونه از آنها میباشد. الگوریتم های جستجو در امتداد گرادیان دارای زمان اجرایی از مرتبه نمایی نیستند اما بسیار کند عمل میکنند. یک دلیل برای کندی این الگوریتمها شکل خاص تابع مورد جستجو (مجموع مربعات خطا) میباشد. بنابراین پیچیدگی آموزش یک شبکه به شکل سطح خطا وابسته است که خود به ساختار شبکه عصبی بستگی دارد. هنوز شکل سطح خطا برای انواع شبکه ها بررسی نشده است. تنها نتیجه ای که در این رابطه در دسترس است برای شبکه های سه لایه میباشد. اثبات شده است که اگر تعداد واحدهای مخفی در یک شبکه عصبی سه لایه به اندازه یک واحد کمتر از تعداد داده های آموزشی انتخاب شود سطح خطای این شبکه بدون حداقل محلی خواهد بود [۳۲]. متأسفانه از این نتیجه در عمل نمیتوان استفاده نمود زیرا شبکه تولید شده بسیار بزرگ و دارای قدرت تعمیم بسیار کم خواهد بود.

قدرت تعمیم شبکه: قدرت تعمیم شبکه معیار است که مشخص میکند یک شبکه پس از اتمام دوره آموزش روی داده هایی که در مجموعه آموزشی موجود نبوده اند چگونه عمل میکند. هرچه تعداد پاسخهای مناسب شبکه به اینگونه ورودیها بیشتر باشد قدرت تعمیم شبکه بیشتر خواهد بود. قدرت تعمیم یک شبکه برای یک مسئله به سه عامل اندازه و کارایی مجموعه آموزشی، معماری و ساختار شبکه و پیچیدگی فیزیکی مسئله بستگی دارد. از بین سه عامل فوق طراح هیچ نوع کنترلی روی عامل پیچیدگی فیزیکی

مسئله ندارد. بنابراین میتوان قدرت تعمیم شبکه را از دو نقطه نظر مورد بررسی قرار داد.

۱. ساختار شبکه را ثابت در نظر گرفته و با تعیین اندازه مناسب برای مجموعه آموزشی قدرت تعمیم قابل قبولی را برای شبکه ایجاد کرد.

۲. اندازه مجموعه آموزشی را ثابت در نظر گرفته و بدنبال ساختار مناسب (تعداد لایه ها، تعداد واحدهای هر لایه، محدودیت روی مقدار وزنها و غیره) برای حصول به قدرت تعمیم قابل قبول بود.

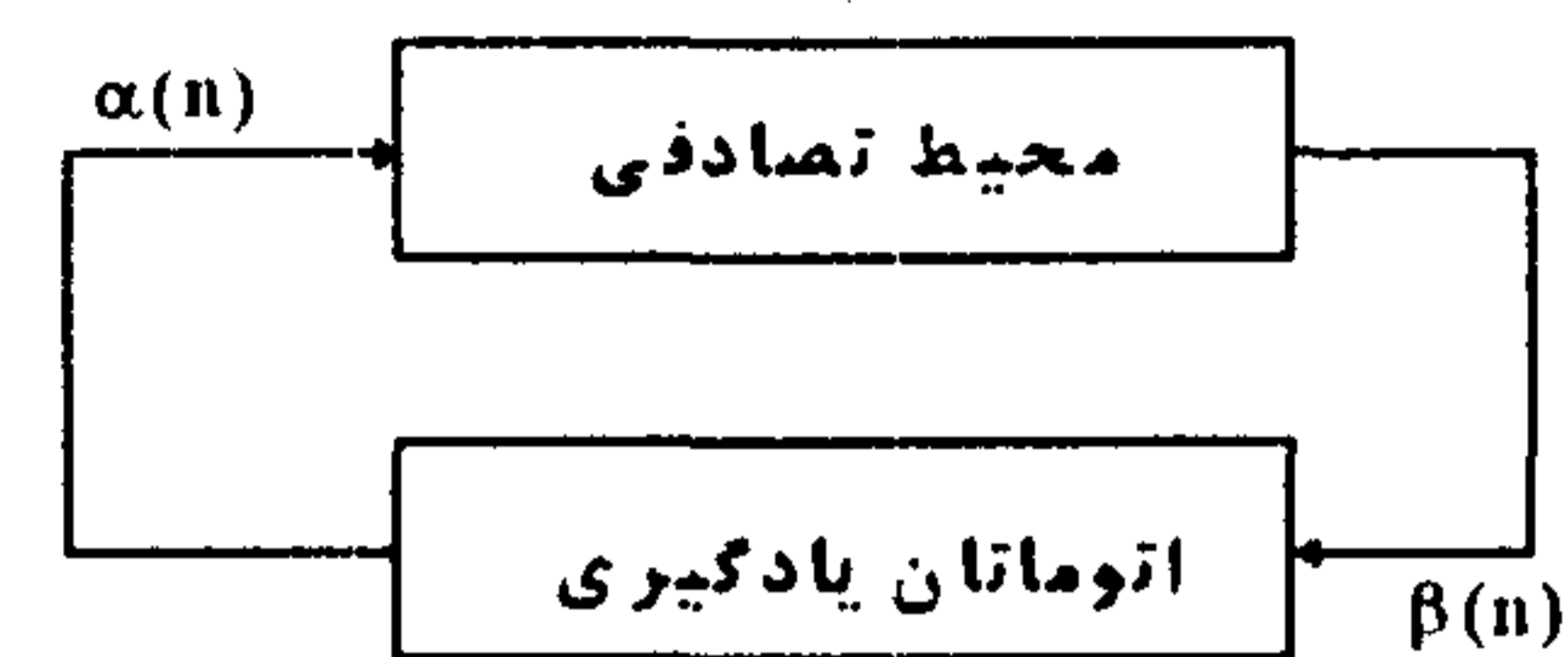
در این مقاله مسئله را از نقطه نظر دوم بررسی میکنیم و سعی میکنیم برای یک مجموعه آموزشی مشخص یک ساختار مناسب برای شبکه را پیدا کنیم.

در این بررسی توجه به این نکته حائز اهمیت است که در صورتیکه تعداد واحدهای مخفی و تعداد وزنه های شبکه کم باشد مرز نواحی تولید شده توسط شبکه هموار بوده و بهمین دلیل شبکه دارای قدرت تعمیم بالا خواهد بود (شبکه ویژگیهای مجموعه آموزشی را یاد میگیرد). در صورت بالا بودن تعداد واحدهای لایع مخفی و تعداد وزنه های لایه مخفی مرز نواحی تولید شده توسط شبکه غیر هموار و دارای شکستگی هایی بوده و بهمین دلیل شبکه دارای قدرت تعمیم پایین می باشد (چنین شبکه ای مشابه یک جدول عمل می کند و تنها الگوهای آموزشی را بخاطر می سپارد). لازم بذکر است که تعداد زیاد وزنه ها تنها باعث کاهش قدرت تعمیم شبکه میگردد بلکه زمان آموزش را افزایش می دهد.

اتوماتانهای یادگیر

یادگیری در اتوماتانهای یادگیر، انتخاب اقدام بهینه از میان مجموعه اقدامهای مجاز میباشد. این اقدام روی یک محیط تصادفی اعمال میشود و محیط به این اقدام اتوماتان بوسیله یک پاسخ از مجموعه پاسخهای مجاز جواب میدهد. پاسخ محیط بصورت آماری به اقدام اتوماتان وابسته است. اصطلاح محیط شامل اجتماع تمام شرایط خارجی و تاثیرات آنها روی عملکرد اتوماتان میباشد.

یک محیط بصورت یکسه تایی (α, C, β) نشان داده میشود. مجموعه $\alpha = \{\alpha_1, \dots, \alpha_p\}$ مجموعه ورودیها، مجموعه $C = \{c_1, \dots, c_p\}$ مجموعه احتمالات (c_i) احتمال شکست اقدام α_i است) و مجموعه $\beta = \{\beta_1, \beta_2\}$ خروجی دودویی محیط میباشد [۳۳]. اتصال یک اتوماتان با محیط در شکل (۱) نشان داده شده است.



شکل ۱: اتصال اتوماتان یادگیری با محیط.

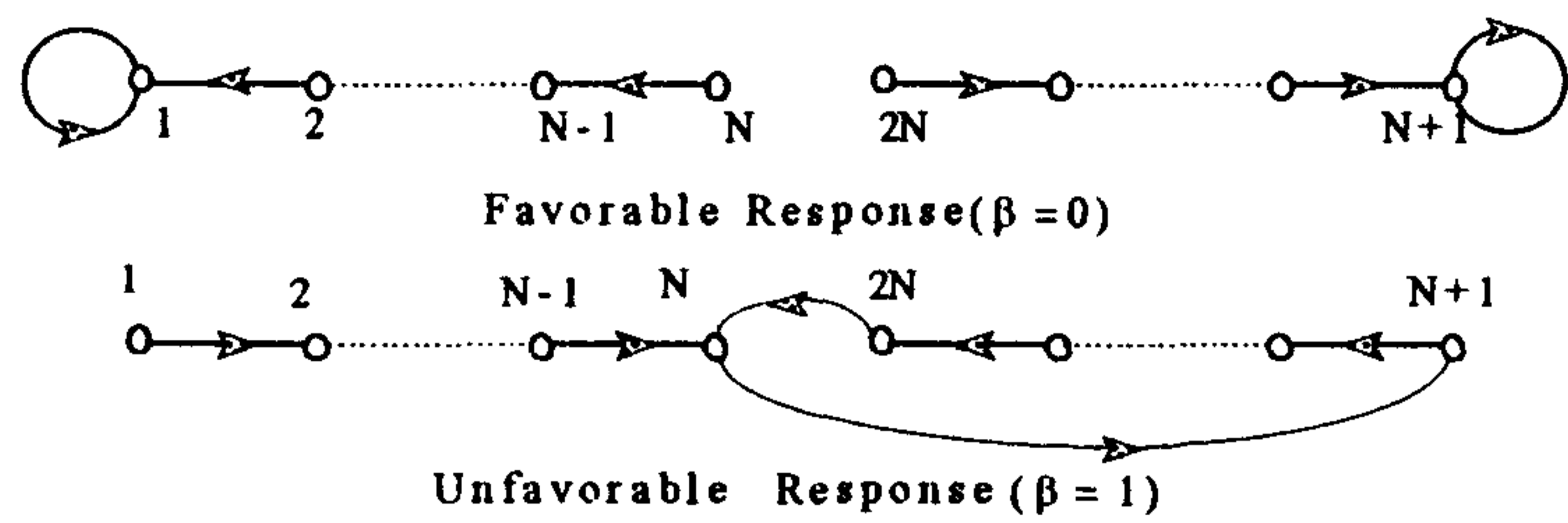
اتوماتانهای یادگیر به دو خانواده اتوماتانهای یادگیر با ساختار ثابت^۷ و اتوماتانهای یادگیر با ساختار متغیر^۸ دسته بندی میشوند. اتوماتانهای *Krylov* و *Krinsky, Tsetline* مثالهایی از خانواده اتوماتانهای یادگیر با ساختار ثابت هستند. یک اتوماتان یادگیر با ساختار ثابت را میتوان با یک پنج تایی $(\alpha, \Phi, \beta, F, G)$ نشان داد. که $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_p\}$ مجموعه اقدامهای مجاز برای اتوماتان یادگیر، $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_s\}$ مجموعه وضعیتهای اتوماتان یادگیر، $\beta = \{0, 1\}$ مجموعه ورودیهای اتوماتان (در این مجموعه یک نمایانگر شکست^۹ و صفر نمایانگر موفقیت^{۱۰} میباشد)، تابع نگاشت وضعیت ها و $G: \Phi \rightarrow \alpha$ تابع نگاشت خروجی میباشد.

اقدام اتوماتان بعنوان ورودی به محیط داده میشود و محیط بعد از اعمال اقدام داده شده توسط اتوماتان یک پاسخ تصادفی که میتواند موفق یا ناموفق باشد را تولید میکند که بعنوان ورودی به اتوماتان داده میشود. اتوماتان با توجه به پاسخ محیط اقدام مربوطه را جریمه میکند و یا به آن پاداش میدهد. اگر احتمال تغییر وضعیت ها در اتوماتان ثابت باشد آنرا اتوماتان یادگیری با ساختار ثابت و در غیر اینصورت آنرا اتوماتان یادگیری با ساختار متغیر مینامند. برای اطلاعات بیشتر در مورد اتوماتانهای یادگیر به مراجع [۳۳]-[۳۶] مراجعه نمایید. در صورتیکه از اتوماتانهای یادگیر برای دسته بندی یا افراز اشیاء استفاده شوند علیرغم سرعت همگرایی خوب، تعداد اقدام های اتوماتان بسیار زیاد خواهد بود [۳۷]. برای کاهش تعداد اقدام های اتوماتان، اتوماتان مهاجرت اشیاء توسط *اومن*^{۱۱} و *ما*^{۱۲} پیشنهاد شده

است [۳۷]. تعداد خروجی های این اتوماتان بمراتب کمتر از اتوماتانهای قبلی است ولی در عوض سرعت همگرایی آن پایین تر میباشد.

اتوماتان مهاجرت اشیاء: اتوماتان مهاجرت اشیاء توسط پنج تایی $(\alpha, \Phi, \beta, F, G)$ نشان داده میشود [۳۷]. که $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_p\}$ مجموعه اقدامهای مجاز، $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_s\}$ مجموعه وضعیتهای، $\beta = \{0, 1\}$ مجموعه ورودیها، تابع $F: \Phi \times \beta \rightarrow \Phi$ تابع نگاشت وضعیت ها و $G: \Phi \rightarrow \alpha$ تابع نگاشت خروجی اتوماتان میباشد. این نوع اتوماتان برای دسته بندی اشیاء [۳۷]، انتساب حروف به کلیدها [۳۸]، افراز گراف^{۱۳} [۳۹]، تناظر گراف [۴۰] و تعیین تعداد واحدهای مخفی شبکه های عصبی [۲۶] [۲۷] مورداستفاده قرار گرفته است. در این اتوماتان هر اقدام یک دسته از الگوها را نشان میدهد. در اتوماتانهای یادگیر با ساختار ثابت پاسخ محیط به اتوماتان سبب تغییر وضعیت اتوماتان میگردد در صورتیکه در اتوماتان مهاجرت اشیاء، اشیا به وضعیت های اتوماتان انتساب داده میشوند و پاسخ محیط به اتوماتان سبب گردش اشیا در بین وضعیت های اتوماتان میگردد. از طریق این گردش دسته بندی اشیا انجام میگردد. اگر شی w_i در اقدام شماره r اتوماتان مهاجرت اشیا قرار داشته باشد این شی متعلق به دسته شماره r است. برای اقدام α_k مجموعه وضعیت $\{\Phi_{(k-1)N+1}, \dots, \Phi_{kN}\}$ در نظر گرفته میشود که N عمق حافظه را نشان میدهد. بدون از دست دادن عمومیت بحث میتوان $\Phi_{(k-1)N+1}$ را داخلی ترین وضعیت، و Φ_{kN} را خارجی ترین وضعیت این اقدام در نظر گرفت. اگر دو شی w_i و w_j بترتیب در وضعیت های $\Phi_{(k-1)N+m}$ و $\Phi_{(k-1)N+1}$ (برای $m > 1$) قرار داشته باشند در اینصورت احتمال تعلق شی w_i به این دسته از احتمال تعلق شی w_j بیشتر است. بنابر این برای اقدام α_k وضعیت $\Phi_{(k-1)N+1}$ وضعیت با بیشترین احتمال و وضعیت Φ_{kN} وضعیت با کمترین احتمال نامیده میشود.

اتوماتان یادگیر تعیین تعداد واحدهای لایه مخفی^{۱۴}: در این قسمت اتوماتان یادگیر تعیین تعداد واحدهای مخفی که از نوع مهاجرت اشیا میباشد و اساس الگوریتم بقا را تشکیل میدهد بطور خلاصه شرح داده میشود [۲۶] [۲۷]. وظیفه این اتوماتان دسته بندی واحدهای لایه مخفی یک شبکه



شکل ۲: اتوماتان HULA (2, N, 1).

$G: \Phi \rightarrow \alpha$ تابع نگاشت خروجی می باشد. این تابع تصمیم میگیرد که به ازای هر وضعیت، اتوماتان چه اقدامی را انجام دهد. اگر واحد H_i متعلق به مجموعه وضعیتهای $\{\Phi_1, \Phi_2, \dots, \Phi_N\}$ باشد این واحد روشن در نظر گرفته میشود. اگر واحد در وضعیت Φ_1 قرار داشته باشد مناسبترین واحد است و بیشترین اهمیت را دارا می باشد. اگر واحد در وضعیت Φ_N قرار داشته باشد دارای کمترین اهمیت می باشد. اقدام شماره دو نیز به همین صورت می باشد. اگر واحد متعلق به مجموعه وضعیت های $\{\Phi_{N+1}, \dots, \Phi_{2N}\}$ باشد واحد را خاموش در نظر میگیریم. اگر واحد در وضعیت Φ_{N+1} قرار داشته باشد بیشترین اهمیت را دارد و اگر در وضعیت Φ_{2N} قرار داشته باشد دارای کمترین اهمیت می باشد.

الگوریتم های پیشنهادی

این قسمت ابتدا به شرح مختصری از الگوریتم بقا می پردازد و سپس الگوریتمهای جدیدی بر اساس اتوماتانهای یادگیر مهاجرت اشیا ارائه میکند.

الگوریتم بقا از یک اتوماتان یادگیر تعیین تعداد واحد های لایه مخفی (بعنوان یک ابزار جستجوی عمومی) و الگوریتم یادگیری انتشار خطا به عقب استفاده میکند و در ضمن آموزش، تعداد واحدهای مخفی مورد نیاز برای یک شبکه عصبی سه لایه را تعیین مینماید. در الگوریتم بقا، آموزش از یک شبکه بزرگ شروع شده و اتوماتان یادگیر با افزودن و کاستن واحدهای مخفی، تعداد واحدهای لایه مخفی یک شبکه عصبی سه لایه را تعیین میکند. در این الگوریتم، مناسب بودن یا نامناسب بودن یک واحد مخفی بر اساس واریانس فعالیت (انرژی مصرف شده) آن واحد برای الگوهای مختلف می باشد. در الگوریتمهای پیشنهاد شده در این مقاله از دو اتوماتان استفاده میگردد. در روی یکی از این اتوماتانها وزنها و در روی اتوماتان دیگر واحدهای مخفی شبکه قرار

عصبی سه لایه به دو دسته واحد های مناسب و واحد های نامناسب است. اتوماتان یادگیر تعیین تعداد واحدهای لایه مخفی بصورت یک شش تایی $(\alpha, H, \Phi, \beta, F, G)$ نشان داده میشود که در آن $\alpha = \{\alpha_1, \alpha_2\}$ مجموعه اقدام های مجاز برای اتوماتان یادگیر است. این اتوماتان دو اقدام دارد که اقدام شماره یک آن اقدام مناسب یا واحد های روشن نام دارد. واحد هایی که در وضعیت های این اقدام قرار دارند برای آموزش شبکه عصبی مورد استفاده قرار میگیرند. اقدام شماره دو آن اقدام نامناسب یا واحد های خاموش نام دارد. واحدهایی که در وضعیت های مربوط به این اقدام قرار میگیرند برای آموزش شبکه عصبی مورد استفاده قرار نمی گیرند. $H = \{H_1, H_2, \dots, H_n\}$ مجموعه واحدهای مخفی موجود در وضعیت های اتوماتان می باشد. اگر واحد H_i در اقدام شماره یک اتوماتان ظاهر شود این واحد بعنوان واحد مخفی مناسب (روشن) و در غیر این صورت این واحد بعنوان واحد مخفی نامناسب (خاموش) در نظر گرفته میشود. $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_{2N}\}$ مجموعه وضعیتها و N عمق حافظه برای اتوماتان می باشد. مجموعه وضعیتهای این اتوماتان به دو زیرمجموعه $\{\Phi_1, \dots, \Phi_N\}$ و $\{\Phi_{N+1}, \dots, \Phi_{2N}\}$ افزایش میشود و بر اساس اینکه واحدهای مخفی در چه وضعیتی قرار داشته باشند، دسته بندی میشوند. بر این اساس واحدهای روشن با مجموعه $ON = \{H_i \mid 1 \leq \text{State}(H_i) \leq N\}$ و واحدهای خاموش با مجموعه $OFF = \{H_i \mid N+1 \leq \text{State}(H_i) \leq 2N\}$ نشان داده میشوند و $\beta = \{0, 1\}$ نشان دهنده وضعیتی است که واحد H_i در آن قرار دارد. در این مجموعه ورودیهای اتوماتان می باشد. در این مجموعه یک نمایانگر شکست و صفر نمایانگر موفقیت می باشد. $F: \Phi \times \beta \rightarrow \Phi$ تابع نگاشت وضعیت ها می باشد. این تابع از روی وضعیت فعلی و ورودی اتوماتان وضعیت بعدی آنرا تولید می نماید. در واقع این تابع چگونگی گردش واحد های مخفی را در وضعیت های اتوماتان مشخص میکند. شرح کارکرد این تابع در [۲۶] [۲۷] آمده است. برای سهولت نمایش در ارائه مطالب، اتوماتان یادگیر تعیین تعداد واحدهای لایه مخفی با K اقدام، عمق حافظه N و انتساب M واحد مخفی توسط $HULA(K, N, M)$ نشان داده میشود که گراف تغییر حالت برای اتوماتان $HULA(2, N, 1)$ در شکل (۲) نشان داده شده است.

پس از اتمام مرحله اول، مرحله دوم الگوریتم که تعیین تعداد وزنه‌های لازم برای لایه مخفی شبکه میباشد شروع میشود. این مرحله از وزنه‌های متصل به واحد های تعیین شده در مرحله اول الگوریتم استفاده میکند و تنها طریقه اتصال وزنه‌های بین لایه ورودی و لایه مخفی را تعیین مینماید. در این مرحله اتوماتان تعیین واحد های لایه مخفی خاموش و اتوماتان تعیین تعداد وزنه‌های لایه مخفی شبکه روشن میگردد. در ابتدا همه وزنه‌های موجود در شبکه (وزنه‌های متصل به واحد های مخفی روشن) در وضعیت Φ_1 قرار میگیرند. به همه وزنه‌های روشن مدتی اجازه داده میشود تا در آموزش شبکه شرکت نمایند. وزنه‌هایی که عملکرد آنها در این مدت مناسب نیست جریمه میشوند، وزنه‌هایی که عملکرد آنها خیلی خوب است پاداش میگیرند، و وزنه‌هایی که در مورد آنها نمیتوان تصمیم گیری قطعی نمود بدون تغییر وضعیت باقی میمانند. برای تشخیص خوب یا بد بودن عملکرد یک، وزن از قدرت انتشار این وزن (قدر مطلق مقدار وزن) استفاده میشود که توسط دو قانون مکاشفه ای زیر بیان میشود.

عملکرد خوب یک وزن: وزنی دارای عملکرد خوب میباشد که سیگنال وارد شده به این وزن بمیزان زیادی تضعیف نگردد و یا به عبارتی دیگر قدر مطلق آن کوچک نباشد. این بدان معنی است که اطلاعات ذخیره شده در این وزن مهم میباشد.

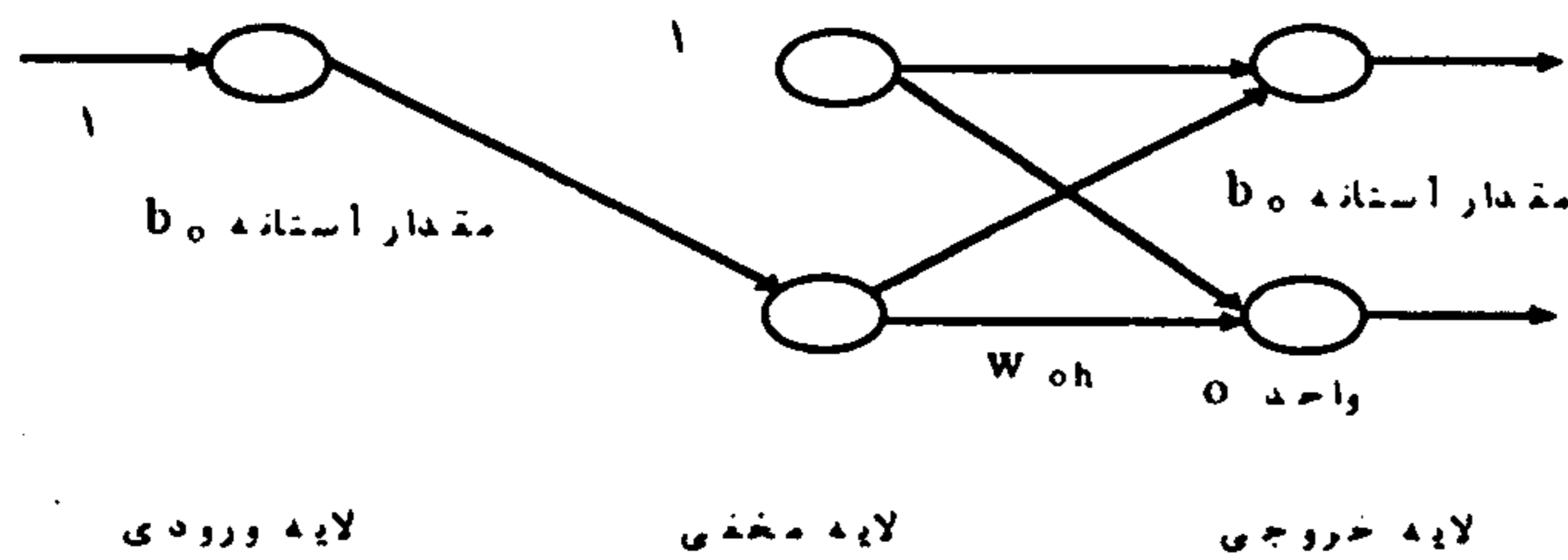
عملکرد بد یک وزن: وزنی دارای عملکرد بد است اگر سیگنال وارد شده به این وزن بمیزان زیادی تضعیف گردد و یا به عبارتی دیگر قدر مطلق آن کوچک باشد. این بدان معنی است که اطلاعات ذخیره شده در این وزن واحد مهم نمی باشد.

تشخیص نحوه عملکرد یک وزن روشن: اگر طی یک دوره آموزش شبکه توسط الگوریتم یادگیری انتشار خطا به عقب، قدر مطلق مقدار یک وزن از یک مقدار آستانه ای کمتر باشد این وزن بد و اگر قدر مطلق مقدار آن از مقدار آستانه ای دیگر بیشتر باشد وزن را خوب مینامیم. برای تعیین مقدار آستانه ای از واریانس قدر مطلق مقدار وزنها استفاده میشود. بدلیل اینکه در مرحله اول این الگوریتم، الگوریتم بقا تعاد واحدهای مخفی شبکه را تعیین کرده است و در اینجا تنها هدف تعیین اتصال وزنه‌های بین لایه ورودی و لایه مخفی

میگیرند. برای سادگی درارایه مطالب، دو اتوماتان موجود در الگوریتم بصورت $HULA(2, N_1, H)$ - $HWLA(2, N_2, W)$ نشان داده میشوند که $HULA(2, N_1, H)$ اتوماتان تعیین تعداد واحدهای لایه مخفی است که دارای دو اقدام و عمق حافظه N_1 است و در روی وضعیتهای آن H واحد مخفی قرار دارد و $HWLA(2, N_2, W)$ اتوماتان تعیین تعداد وزنه‌های لایه مخفی^{۱۵} است که دارای دو اقدام و عمق حافظه N_2 است که روی وضعیتهای آن مجموعه وزن $W = \{w_1, w_2, w_3, \dots\}$ قرار دارد. گراف تغییر حالت برای اتوماتان تعیین تعداد وزنه‌های لایه مخفی شبکه در شکل (۲) نشان داده شده است. همچنین برای تنظیم فعالیت اتوماتانها، الگوریتم‌هایی بنام الگوریتم‌های زمانبندی^{۱۶} معرفی شده است. وظیفه این الگوریتم‌ها خاموش یا روشن کردن این دو اتوماتان است. انواع مختلفی از الگوریتم‌های زمانبندی میتواند مورد استفاده قرار گیرد که در این مقاله تنها به دو مورد اشاره شده است. لازم به ذکر است که الگوریتم های زمانبندی مختلف منجر به کارایی های متفاوت میگرددند. حال به شرح الگوریتم‌های پیشنهادی میپردازیم.

الگوریتم ۱: این الگوریتم دارای دو مرحله میباشد: مرحله تعیین تعداد واحدهای مورد نیاز لایه مخفی و مرحله تعیین تعداد وزنه‌های مورد نیاز لایه مخفی شبکه. برای تعیین تعداد واحدهای لایه مخفی از الگوریتم بقا استفاده گردیده است. الگوریتم بقا حداقل تعداد واحدهای لایه میانی شبکه عصبی سه لایه که قدرت تعمیم قابل قبولی از رایه و قادر به مدل سازی مسئله گردد را تعیین میکند. در الگوریتم بقا، در ابتدا همه واحدهای مخفی روشن بوده و در وضعیت Φ_1 قرار میگیرند و به همه واحد های روشن مدتی اجازه داده میشود تا در آموزش شبکه شرکت نمایند. واحد هایی که عملکرد آنها در این مدت مناسب نبوده جریمه میشوند، واحد هایی که عملکرد آنها خیلی خوب بوده پاداش میگیرند و واحدهایی که در مورد آنها نمیتوان تصمیم گیری قطعی نمود بدون تغییر وضعیت باقی میمانند. برای تشخیص خوب یا بد بودن عملکرد یک واحد از متوسط انرژی مصرف شده آن واحد استفاده میشود. چگونگی تغییر فعالیت یک واحد به ازای مجموعه الگوهای آموزشی، انرژی مصرف شده آن واحد نامیده میشود.

خروجی (b_o) قرار می‌دهیم و وزنه‌های b_{in} و W_{oh} و واحد مخفی h حذف می‌گردند.



شکل ۴: حذف وزن بایاس یک واحد.

تشخیص نحوه عملکرد یک وزن خاموش: بدلیل اینکه در این مرحله اتوماتان تعیین تعداد واحدهای مخفی خاموش است بنابراین هیچکدام از واحدهای مخفی تغییر وضعیت نمی‌دهند و تنها وزنه‌های خاموش متصل به واحدهای روشن بررسی می‌گردند. یک وزن خاموش در آموزش شبکه شرکت نمی‌کند و بنابراین مقدار آنرا از آخرین زمان روشن بودن آن محاسبه می‌کنیم. اگر یک وزن برای یک مدت طولانی خاموش باشد از اهمیت و در نتیجه از مقدار آن کاسته می‌شود. بنابراین مقدار یک وزن خاموش بصورت زیر محاسبه می‌شود.

$$W_{hk}(n+1) = W_{hk}(n) \lambda_d$$

که ثابت $0 < \lambda_d < 1$ ضریب کاهش مقدار وزن نامیده می‌شود و n شاخص زمان را نشان می‌دهد. به این ترتیب مقدار یک وزن خاموش بتدریج کم می‌شود. واریانس وزنه‌های خاموش بصورت زیر محاسبه می‌شود

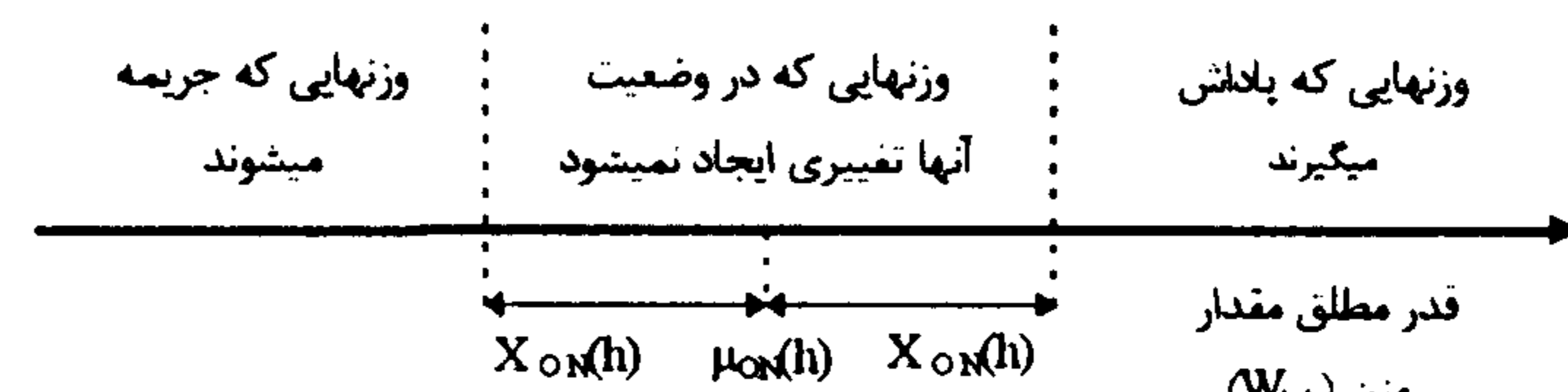
$$\sigma_h^{(OFF)} = \sqrt{\frac{\sum_{W_{hk} \in OFF_w(h)} (|W_{hk}| - \mu_{OFF}(h))^2}{|OFF_w(h)|}} \quad h \in ON$$

که در آن W_{hk} وزن اتصال بین واحد ورودی k به واحد مخفی h و $OFF_w(h)$ مجموعه وزنه‌های ورودی خاموش متصل به واحد مخفی روشن h و $\mu_{OFF}(h)$ میانگین قدرمطلق مقدار وزنه‌های خاموش متصل به واحد مخفی روشن h می‌باشد. پس از محاسبه واریانس وزنه‌های متصل به واحدهای روشن، وزنه‌های خاموشی که قدرمطلق مقدار آنها از یک مقدار آستانه‌ای کمتر باشد پاداش می‌گیرند و وزنه‌های خاموشی که قدرمطلق مقدار آنها از یک مقدار آستانه‌ای دیگر بیشتر باشد جریمه می‌شوند و وزنه‌های خاموشی که قدرمطلق مقدار آنها بین دو مقدار آستانه‌ای فوق قرار می‌گیرند تغییری در وضعیت آنها ایجاد نمی‌شود. (شکل ۵)

است لذا برای محاسبه واریانس قدرمطلق مقدار وزنه‌ها، واریانس بصورت مجزا و برای وزنه‌های متصل به هر یک از واحدهای مخفی روشن و بصورت زیر محاسبه می‌گردد.

$$\sigma_h^{(ON)} = \sqrt{\frac{\sum_{W_{hk} \in ON_w(h)} (|W_{hk}| - \mu_{ON}(h))^2}{|ON_w(h)|}} \quad h \in ON$$

که در آن W_{hk} وزن اتصال بین واحد ورودی k به واحد مخفی h و $ON_w(h)$ مجموعه واحدهای روشن، $\mu_{ON}(h)$ میانگین قدرمطلق مقدار وزنه‌های روشن متصل به واحد مخفی h می‌باشد. پس از محاسبه واریانس وزنه‌های متصل به واحدهای روشن، وزنه‌های روشن که قدرمطلق مقدار آنها از یک مقدار آستانه‌ای کمتر باشد جریمه می‌شوند و وزنه‌های روشن که قدرمطلق مقدار آنها از یک مقدار آستانه‌ای دیگر بیشتر باشد پاداش می‌گیرند و وزنه‌های روشن که قدرمطلق مقدار آنها بین دو مقدار آستانه‌ای فوق قرار می‌گیرند تغییری در وضعیت آنها ایجاد نمی‌شود. (شکل ۳)

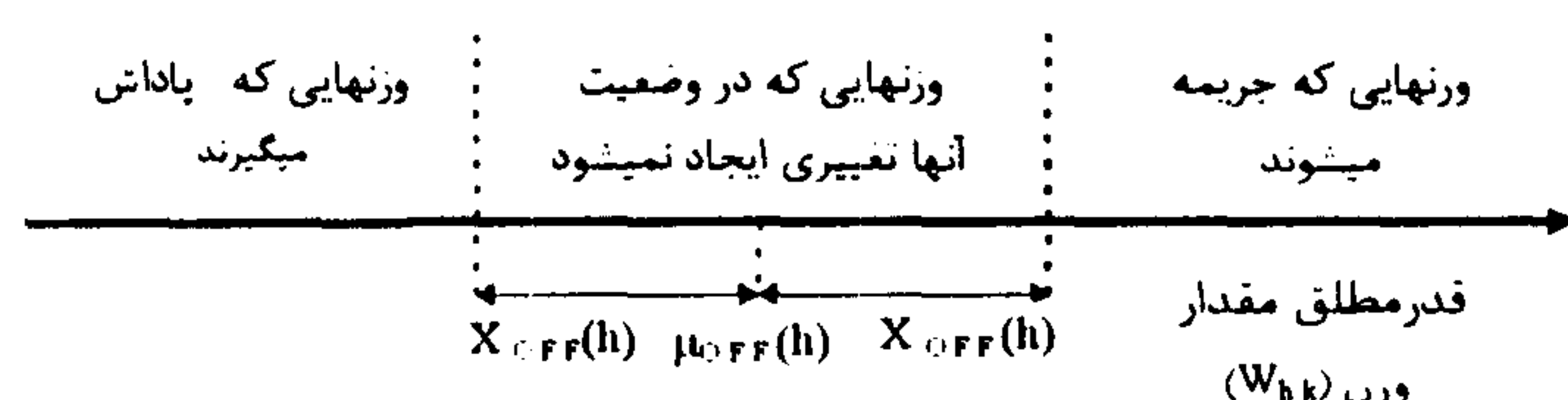


شکل ۳: تشخیص مقدار آستانه وزنه‌های روشن.

در شکل فوق پهنای روشنی نامیده می‌شود و بصورت زیر محاسبه می‌گردد.

$$X_{ON}(h) = \lambda_{ON} \frac{|OFF_w(h)|}{\#INPUT + 1} \sigma_h^{(ON)}$$

ثابت $\lambda_{ON} > 0$ ضریب پهنای روشنی، $\#INPUT$ تعداد واحدهای لایه ورودی و $OFF_w(h)$ مجموعه وزنه‌های خاموش متصل به واحد مخفی روشن h است. وزنه‌هایی که قدرمطلق مقدار آنها از $\mu_{ON}(h) - X_{ON}(h)$ کمتر باشد جریمه می‌شوند. وزنه‌هایی را که قدرمطلق مقدار آنها از $\mu_{ON}(h) + X_{ON}(h)$ بیشتر باشند پاداش می‌گیرند و وزنه‌هایی که قدرمطلق مقدار آنها در فاصله $[\mu_{ON}(h) - X_{ON}(h), \mu_{ON}(h) + X_{ON}(h)]$ قرار دارند بدون تغییر وضعیت باقی می‌مانند. اگر یک واحد مخفی (مطابق شکل ۴) دارای تنها یک وزن آستانه باشد این واحد و وزن آنرا می‌توان حذف کرد و اثر خروجی این واحد را در مقدار آستانه واحدهای لایه خروجی اعمال نمود. در این حالت مقدار $b_o + W_{oh} * f(b_{in})$ را بعنوان مقدار آستانه واحد



شکل ۵: تشخیص مقدار آستانه وزنهایی خاموش.

ورودی یک واحد مخفی، آن وزن باکلیه وزنهایی ورودی واحدهای روشن موجود در لایه مخفی شبکه مقایسه میگردد. در این الگوریتم حذف یک واحد بطور غیر مستقیم از طریق حذف وزنهایی آن انجام خواهد گرفت در صورتیکه در الگوریتمهای ۱ و ۲ برای تعیین اهمیت یک وزن، مقدار آن وزن با مقدار وزنهایی ورودی همان واحد مقایسه میگردد. در این الگوریتم، در ابتدا همه وزنهایی موجود در شبکه در وضعیت Φ_1 اتوماتان HWLA قرار میگیرند و به همه وزنهایی روشن مدتی اجازه داده میشود تا در آموزش شبکه شرکت نمایند. وزنهایی که عملکرد آنها در این مدت مناسب نبوده جریمه میشوند، وزنهایی که عملکرد آنها خوب بوده پاداش میگیرند و وزنهایی که در مورد آنها نمیتوان تصمیم گیری قطعی نمود بدون تغییر وضعیت باقی میمانند. برای تشخیص خوب یا بد بودن عملکرد یک وزن از دو قانون مکاشفه ای بیان شده در الگوریتم ۱ استفاده شده است.

تذکره ۱: در بسیاری از کاربردها ممکن است یک واحد مخفی مسئول پردازش قسمتی از الگوی ورودی باشد. مثلاً در شناسایی نوری حروف، یک واحد مخفی ممکن است پردازش ناحیه کوچکی از تصویر حرف را عهده دار باشد. اگر قسمتی از تصویر برای همه حروف یکسان باشد در فرایند شناسایی بی تاثیر خواهد بود. در این شرایط از شبکه های عصبی با اتصالات محلی استفاده میشود. این نوع شبکه ها ممکن است دارای تعداد واحدهای مخفی بیشتری نسبت به شبکه های کاملاً متصل باشد اما دارای تعداد وزنهایی کمتر و قدرت تعمیم بیشتری هستند زیرا خروجی هر واحد مخفی به تعداد متغیر کمتری وابسته است و هموار تر است و در نتیجه قدرت تعمیم آن بیشتر است [۴۱]. این مسئله در الگوریتم ۲ (وقتی که تعداد واحدهای مخفی حدس زده شده زیاد باشد) و الگوریتم ۳ مد نظر میباشد. در این حالت وزنهایی از لایه مخفی شبکه حذف میگردند که ورودی برای آن وزنهادر بیشترالگوهای ورودی یکسان باشد. نتایج آزمایشات که در قسمت بعد به آن میپردازیم نشان می دهند که هر سه الگوریتم ارائه شده، شبکه هایی با اتصالات محلی تولید میکنند. نتایج آزمایشات همچنین نشان میدهند که تعداد واحدهای مخفی که الگوریتم شماره ۱ تعیین میکند

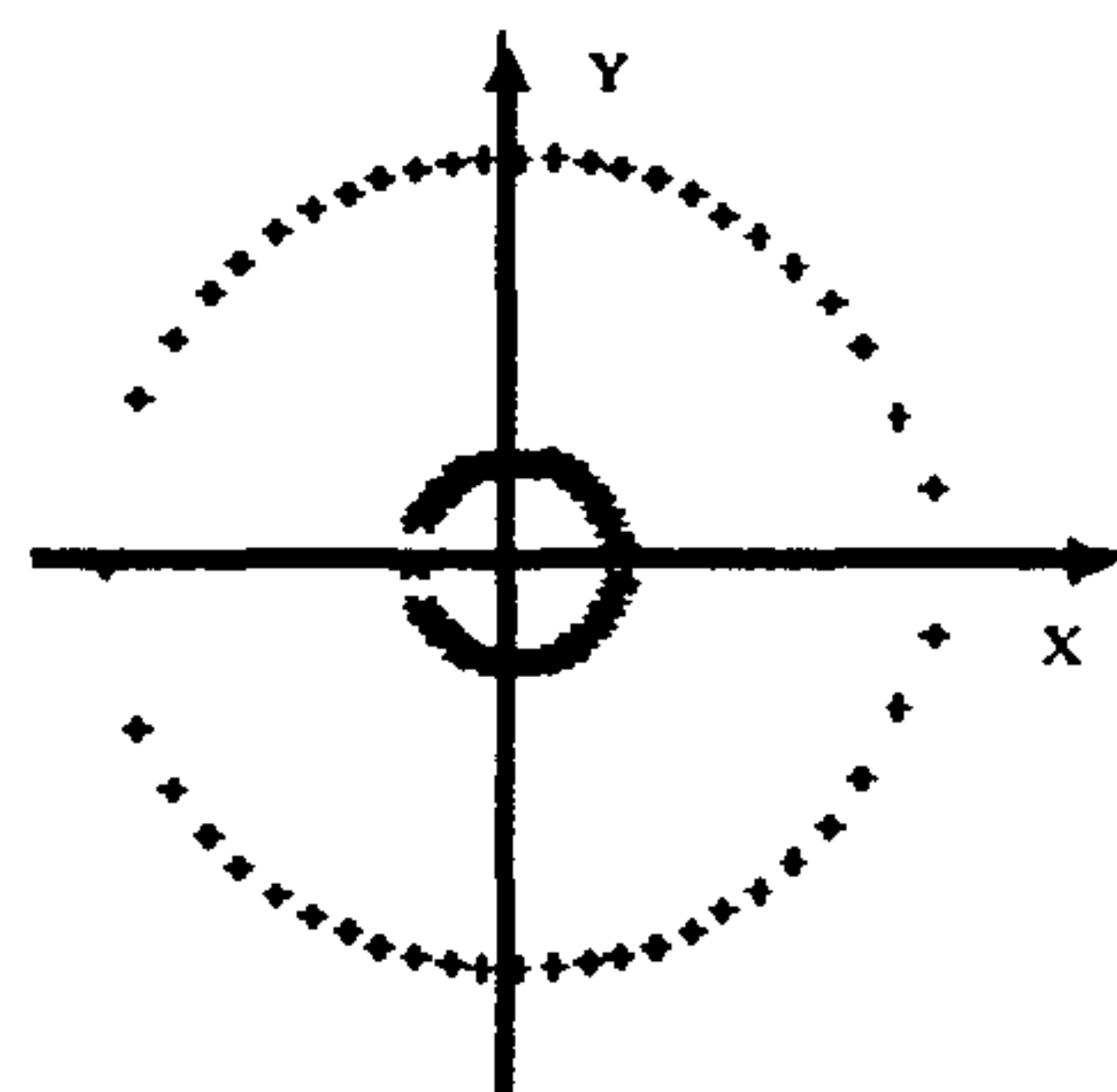
در شکل ۵ پهنای خاموشی نامیده میشود و بصورت زیر محاسبه میگردد.

$$X_{OFF}(h) = \lambda_{OFF} \frac{|ON_w(h)|}{\#INPUT + 1} \sigma_h(OFF)$$

که ثابت $\lambda_{OFF} > 0$ ضریب پهنای خاموشی و $\#INPUT$ تعداد واحدهای لایه ورودی است. وزنهایی که قدر مطلق مقدار آنها از $\mu_{OFF}(h) - X_{OFF}(h)$ کمتر باشد پاداش میگیرند. وزنهایی که قدر مطلق مقدار آنها از $\mu_{OFF}(h) + X_{OFF}(h)$ بیشتر باشند جریمه میشوند. وزنهایی که قدر مطلق مقدار آنها در فاصله $[\mu_{OFF}(h) - X_{OFF}(h), \mu_{OFF}(h) + X_{OFF}(h)]$ قرار دارد وضعیت آنها تغییر نمیکند. زمانیکه یک وزن از حالت خاموش به حالت روشن میرود مقدار آن بصورت تصادفی مقداردهی میگردد. در الگوریتم ۱، الگوریتم زمانبندی در ابتدا اتوماتان تعیین تعداد واحدهای لایه مخفی را روشن و اتوماتان تعیین تعداد وزنهایی لایه مخفی را خاموش میکند. پس از تعیین تقریبی تعداد واحدهای لایه مخفی شبکه، اتوماتان تعیین تعداد واحدهای لایه مخفی خاموش و اتوماتان تعیین تعداد وزنهایی لایه مخفی شبکه را روشن میکند. در بعضی از مواقع و با استفاده از اطلاعات موجود درباره مسئله، تعداد تقریبی واحدهای لایه مخفی را میتوان از قبل تعیین نمود و بهمین دلیل مرحله تعیین تعداد واحدهای لایه مخفی مورد نیاز نخواهد بود. در الگوریتم ۲ این مسئله مد نظر گرفته شده است.

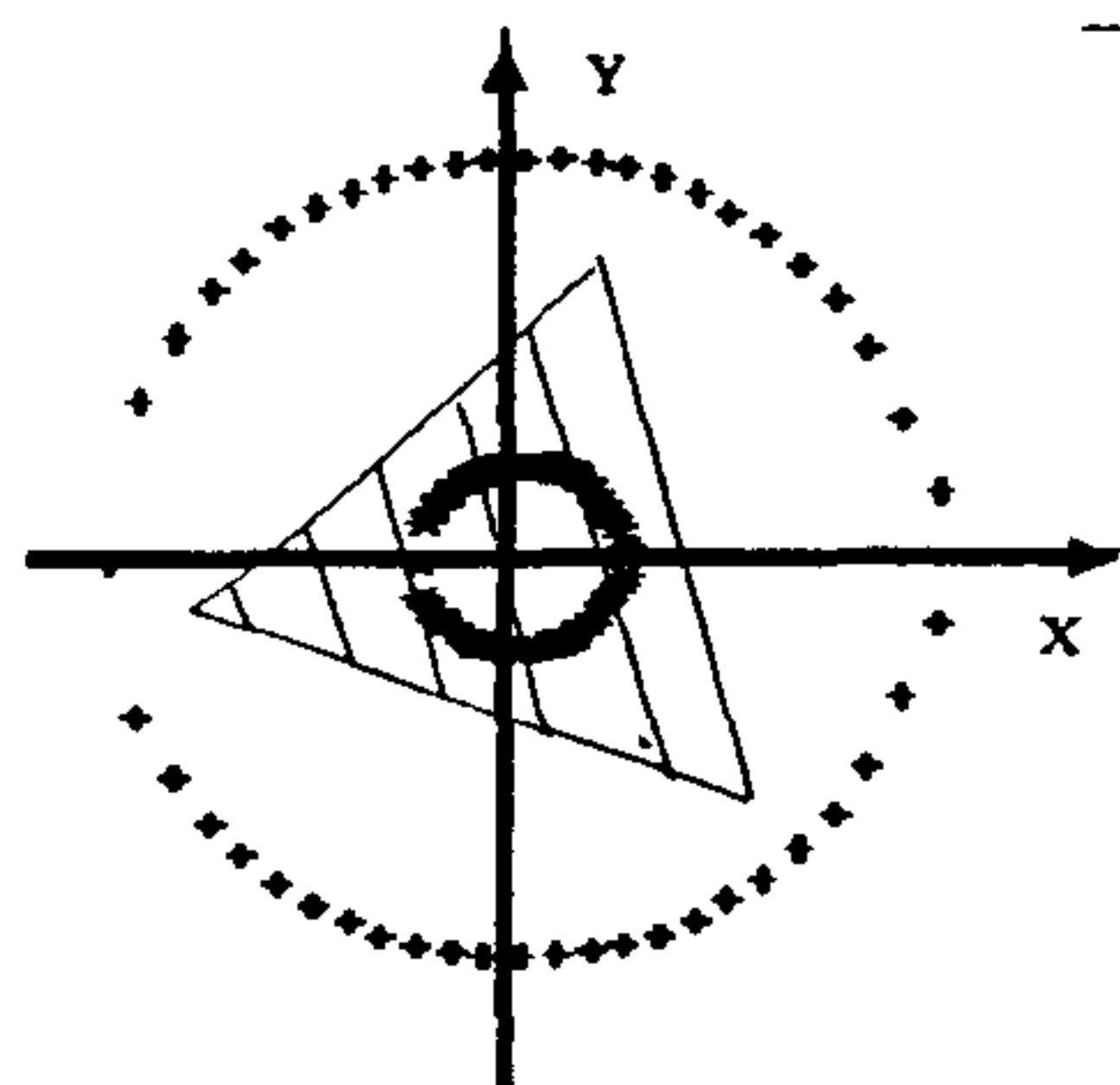
الگوریتم ۲: در صورتیکه تعداد تقریبی واحدهای لایه مخفی از قبل مشخص باشد میتوانیم از شکل خلاصه تری از الگوریتم ۱ استفاده نماییم و ساختار شبکه را تعیین کنیم. الگوریتم ۲ حالت خاصی از الگوریتم ۱ است که در آن اتوماتان تعیین تعداد واحدهای لایه مخفی همیشه خاموش است. معیار تشخیص خوب یا بد بودن عملکرد یک وزن و همچنین نحوه جریمه کردن و پاداش دادن به یک وزن همانند الگوریتم ۱ است. الگوریتم ۳: این الگوریتم در یک مرحله تعداد وزنهایی را تعیین میکند. در این الگوریتم برای تعیین اهمیت هر وزن

نمونه می‌باشد که ۱۰۰ نمونه از آنها برای مجموعه آموزشی و از ۱۰۰ نمونه دیگر برای نمونه آموزشی استفاده گردیده است.



شکل ۸: الگوهای مثال ۲.

این دسته‌ها بصورت خطی از هم جدایی پذیر نیستند و برای جداسازی آنها یک شبکه عصبی سه لایه با حداقل ۳ واحد مخفی و ۹ اتصال مورد نیاز می‌باشد نواحی تولید شده توسط چنین شبکه‌ای در شکل (۹) نشان داده شده است.



شکل ۹: خط‌های جدا کننده الگوهای مثال ۲.

ج) مسئله تشخیص اعداد چاپی فارسی: در این مسئله تشخیص اعداد چاپی فارسی که در شکل (۱۰) مد نظر می‌باشد روی یک صفحه کاغذ چاپ شده اند و از طریق یک دستگاه پوششگر و با قدرت تفکیک ۳۰۰ نقطه در اینچ نمونه برداری گردیده است. از هر حرف ۱۷ نمونه وجود دارد که از ۱۶ نمونه برای مجموعه آموزشی و از ۱ نمونه دیگر برای مجموعه آزمایشی استفاده شده است.

۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹

شکل ۱۰: اعداد چاپی فارسی.

پس از جداسازی حروف در تصویر، ثابتهای گشتاوری M_1 تا M_7 استخراج شده است و این ۷ ویژگی ورودیهای شبکه عصبی را تشکیل می‌دهند [۴۳].

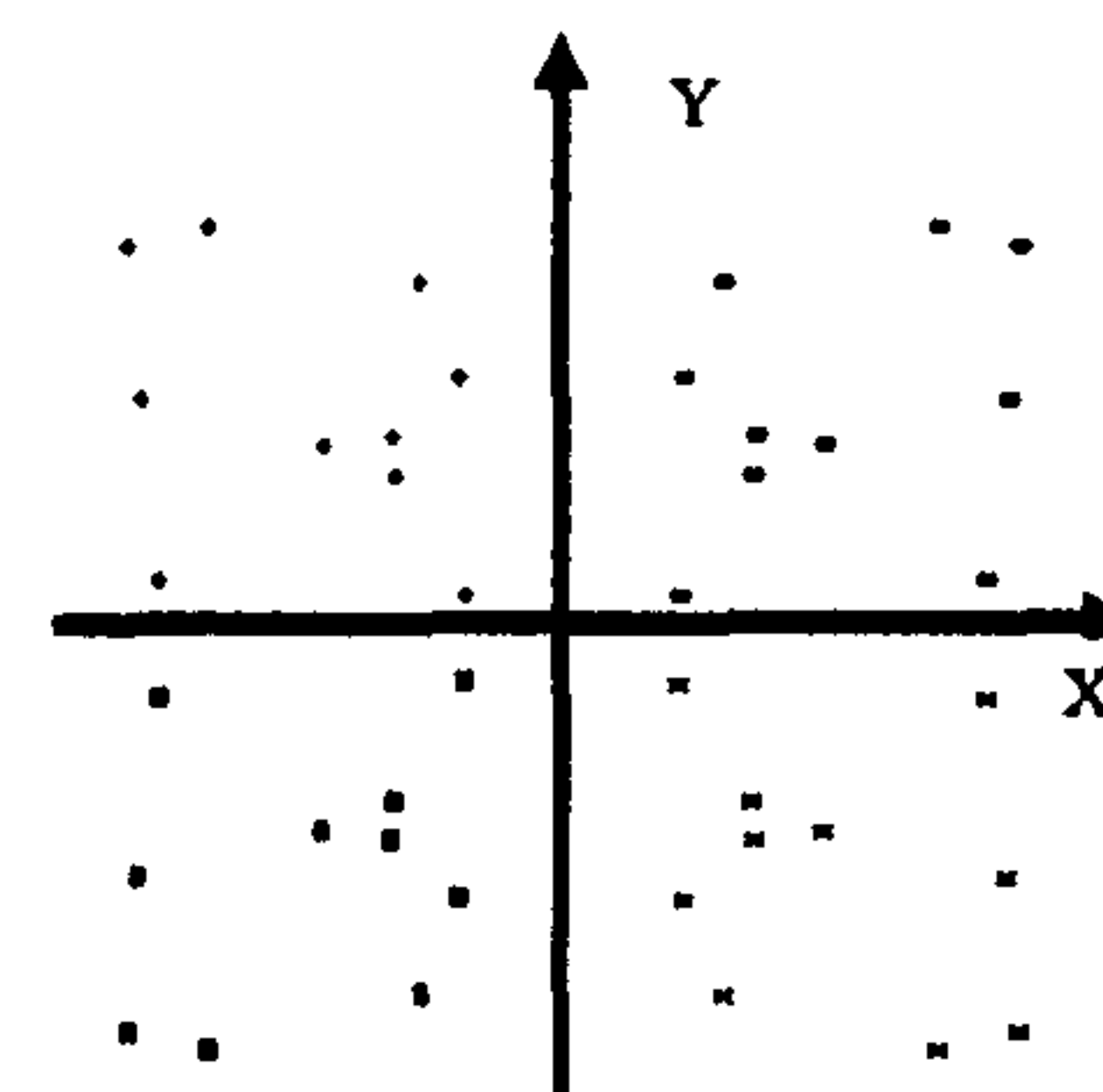
د) مسئله تشخیص اعداد لاتین: در این مسئله ده عدد وجود دارد که هر کدام از آنها توسط یک ماتریس 8×8 از نقطه‌های سیاه و سفید نشان داده میشود [۴۴]. الگوهای آموزشی برای این مسئله در شکل (۱۱) نشان داده شده است.

بمراتب کمتر از واحدهای مخفی تعیین شده توسط الگوریتم های ۲ و ۳ می‌باشد.

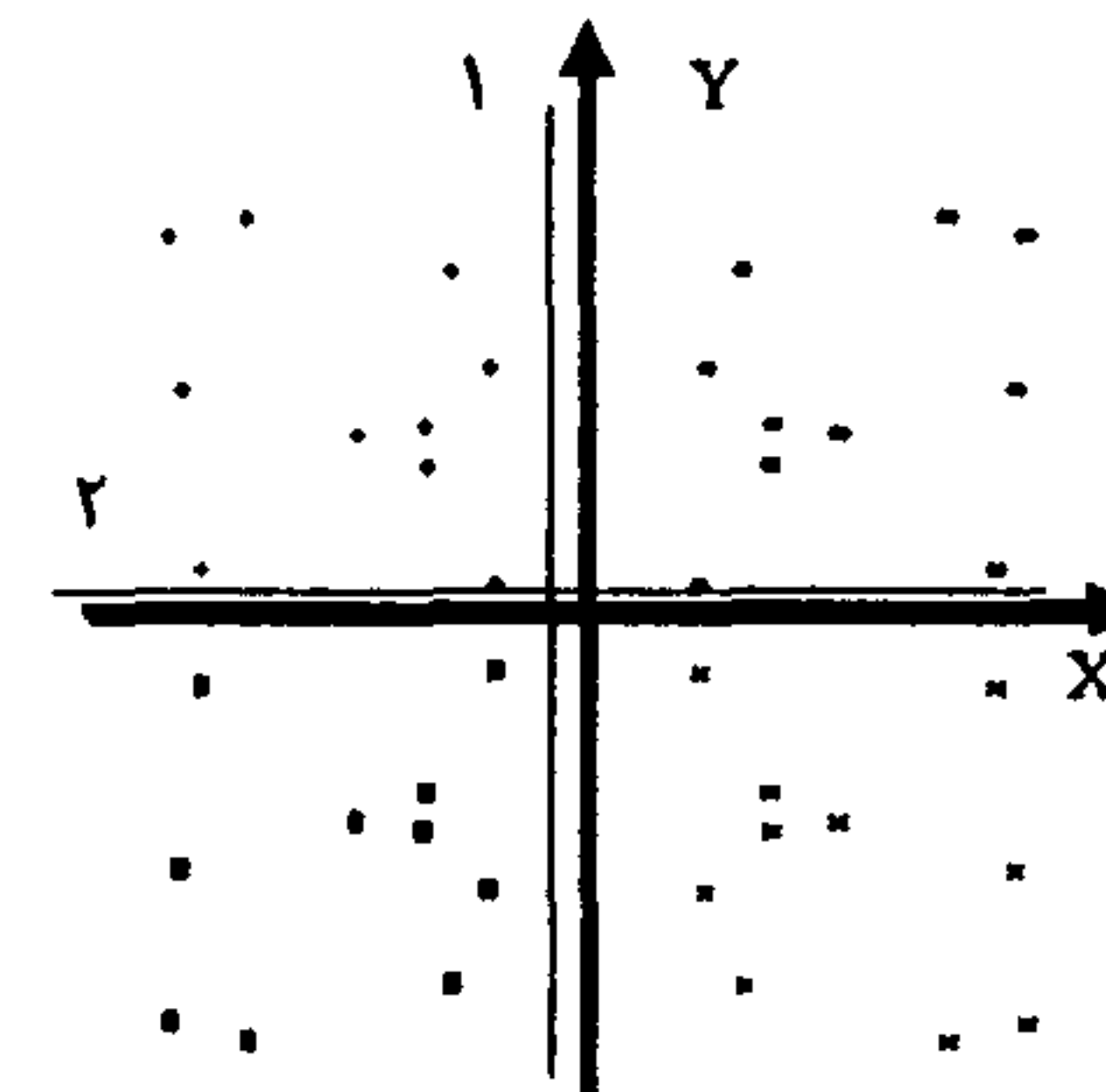
نتایج شبیه سازیها

در این قسمت چهار مسئله نمونه که الگوریتمهای پیشنهادی دی روی آنها پیاده سازی شده است معرفی می‌گردد و سپس نتایج آزمایشات ارایه میشود.

الف) دسته بندی چهار دسته از الگوها: در این مسئله چهار دسته الگو (که در شکل (۶) با علائم "+", "*", "x" و "# نشان داده شده اند) وجود دارد که هر الگو توسط دو ویژگی x و y نمایش داده می‌شوند و این دو ویژگی ورودیهای شبکه را تشکیل میدهند [۴۲]. هر دسته از این الگوها دارای ۱۵ نمونه هستند که ۱۰ نمونه از آنها برای مجموعه آموزشی و از ۵ نمونه دیگر برای مجموعه آزمایشی استفاده گردیده است. این دسته‌ها بصورت خطی از هم جدایی پذیر نیستند و برای جدا سازی آنها یک شبکه عصبی سه لایه با حداقل دو واحد مخفی و ۴ اتصال مورد نیاز می‌باشد نواحی تولید شده توسط چنین شبکه‌ای در شکل (۵) نشان داده شده است.



شکل ۶: توزیع الگوها در صفحه.



شکل ۷: خط‌های تولید شده توسط واحد های مخفی.

ب) دسته بندی دو دسته از الگوها: در این مسئله دو دسته الگو وجود دارد که در شکل (۸) توسط علائم "*" و "+" نشان داده شده اند و بترتیب روی دایره‌هایی به شعاع های 0.5 و 2 توزیع شده اند. الگوها توسط دو ویژگی x و y نمایش داده میشوند و این دو ویژگی ورودیهای شبکه عصبی را تشکیل میدهند [۴۱]. هر دسته از این الگوها دارای ۲۰۰

ب) ساختار شبکه پس از پایان مرحله تعیین وزنها

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۴	۸	۱۰۰	۱۰۰
۲	۶	۹	۱۰۰	۱۰۰
۳	۵	۱۰	۱۰۰	۱۰۰
۴	۶	۱۱	۱۰۰	۱۰۰
۵	۴	۸	۱۰۰	۱۰۰
۶	۴	۵	۱۰۰	۱۰۰
متوسط	۴/۸	۸/۵	۱۰۰	۱۰۰

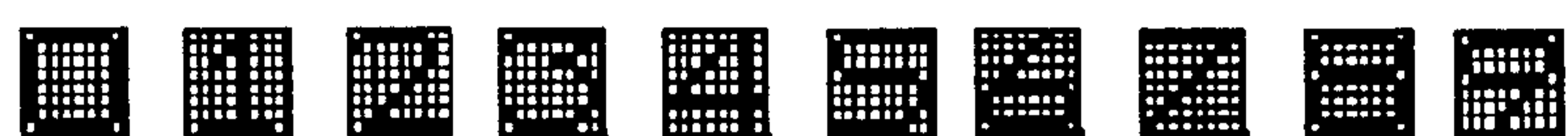
جدول ۲: نتیجه الگوریتم ۱ برای مسئله ۲.

الف) ساختار شبکه پس از پایان مرحله تعیین واحدهای مخفی

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۷	۲۱	۱۰۰	۱۰۰
۲	۱۲	۳۶	۱۰۰	۱۰۰
۳	۸	۲۴	۱۰۰	۱۰۰
۴	۵	۱۵	۱۰۰	۱۰۰
۵	۸	۲۴	۱۰۰	۱۰۰
۶	۴	۱۲	۱۰۰	۱۰۰
متوسط	۷/۳	۲۲	۱۰۰	۱۰۰

ب) ساختار شبکه پس از پایان مرحله تعیین وزنها

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۴	۸	۱۰۰	۱۰۰
۲	۶	۹	۱۰۰	۱۰۰
۳	۵	۱۰	۱۰۰	۱۰۰
۴	۴	۸	۱۰۰	۱۰۰
۵	۵	۹	۱۰۰	۱۰۰
۶	۴	۵	۱۰۰	۱۰۰
متوسط	۴/۶	۸/۱۶	۱۰۰	۱۰۰



شکل ۱۱: الگوهای آموزشی برای مسئله تشخیص اعداد لاتین.

در ادامه نتایج شبیه‌سازیهای انجام گرفته شده برای مسئله‌های فوق آمده است.

الف) الگوریتم ۱: این الگوریتم همانطور که قبل اشاره شد از دو مرحله تعیین تعداد واحدهای لایه مخفی و تعیین تعداد وزنه‌های لایه مخفی تشکیل شده است. الگوریتم ۱ روی مسائل فوق پیاده‌سازی شده و نتایج آزمایش‌ها در جداول (۱) تا (۴) آورده شده است. قسمت (الف) جداول (۱) تا (۴) شبکه تولید شده توسط الگوریتم بقا و قسمت (ب) این جداول نتیجه تولید شده بعد از مرحله دوم الگوریتم ۱ را نشان می‌دهد لازم بذکر است که در تمامی آزمایش‌های انجام گرفته توسط الگوریتم ۱، مجموعه الگوهای آموزش ۱۰۰۰۰ بار به شبکه داده شده است.

دسته‌بندی چهار دسته از الگوها: الگوریتم ۱ روی ۶ شبکه با وزنه‌های اولیه تصادفی آزمایش شده و نتایج آن در جدول (۱) نشان داده شده است. برای تمامی این شبکه‌ها از اتوماتان $HULA(2, 7, 39)-HWLA(2, 7, N_w)$ استفاده گردیده است. N_w تعداد وزنه‌های شبکه در پایان الگوریتم بقا می‌باشد. لازم بذکر است که برای انجام آزمایش‌های مختلف، نرخ یادگیری برابر ۰/۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۱۲، و ضریب پهنای خاموشی برابر ۱۲ در نظر گرفته شده است.

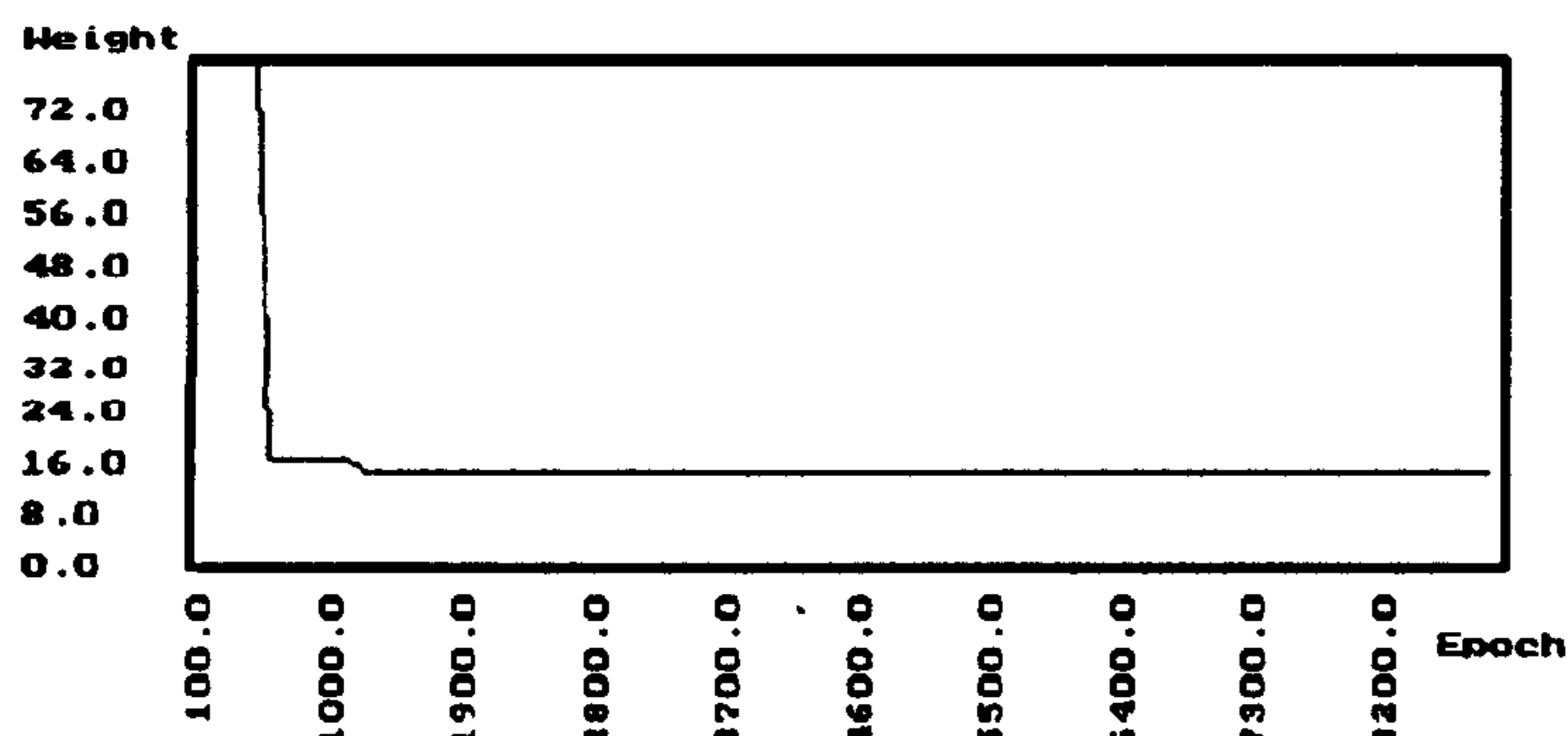
جدول ۱: نتیجه الگوریتم ۱ برای مسئله ۱.

الف) ساختار شبکه پس از پایان مرحله تعیین واحدهای مخفی

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۷	۲۱	۱۰۰	۱۰۰
۲	۱۲	۳۶	۱۰۰	۱۰۰
۳	۸	۲۴	۱۰۰	۱۰۰
۴	۱۱	۳۳	۱۰۰	۱۰۰
۵	۵	۱۵	۱۰۰	۱۰۰
۶	۴	۱۲	۱۰۰	۱۰۰
متوسط	۷/۸	۲۳/۵	۱۰۰	۱۰۰

مسئله تشخیص اعداد چاپی فارسی: الگوریتم ۱ روی ۶ شبکه با وزنهای اولیه تصادفی آزمایش شده و نتایج آن در جدول (۳) نشان داده شده است. برای تمامی این شبکه‌ها از اتوماتان $HULA(2, 7, 60)-HWLA(2, 7, N_w)$ استفاده شده است. N_w تعداد وزنهای شبکه در پایان الگوریتم بقا می‌باشد. برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۴، ضریب گشتاور برابر ۰/۵، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۳۰، و ضریب پهنای خاموشی برابر ۳۰ در نظر گرفته شده است.

شکل (۱۲) منحنی تغییرات تعداد وزنهای لایه مخفی شبکه را بر اساس تکرار ارایه الگو ۱۷ برای شبکه شماره (۱) جدول (۳-ب) نشان می‌دهد.



شکل ۱۲: تغییرات تعداد وزنهای لایه مخفی شبکه بر اساس تکرار ارایه الگو.

جدول ۴: نتیجه الگوریتم ۱ برای مسئله ۴.

الف) ساختار شبکه پس از پایان مرحله تعیین واحدهای مخفی

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۲	۷۸۰	۱۰۰	۱۰۰
۲	۷	۴۵۵	۱۰۰	۱۰۰
۳	۹	۵۸۵	۱۰۰	۱۰۰
۴	۱۳	۸۴۵	۱۰۰	۱۰۰
۵	۱۱	۷۱۵	۱۰۰	۱۰۰
۶	۱۳	۸۴۵	۱۰۰	۱۰۰
متوسط	۱۰/۸۳	۷۰۴/۱۶	۱۰۰	۱۰۰

مسئله تشخیص اعداد لاتین: الگوریتم ۱ روی ۶ شبکه با وزنهای اولیه تصادفی آزمایش شده‌اند و نتایج آن در جدول (۴) نشان داده شده است. برای انجام آزمایش‌های انجام شده

دسته‌بندی دودسته از الگوها: الگوریتم ۱ روی ۶ شبکه با وزنهای اولیه تصادفی آزمایش شده و نتایج آن در جدول (۲) نشان داده شده است. برای انجام این آزمایش‌ها از اتوماتان $HULA(2, 7, 60)-HWLA(2, 7, N_w)$ استفاده شده است. N_w تعداد وزنهای شبکه در پایان الگوریتم بقا می‌باشد. برای انجام این آزمایش‌ها، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۱۰۰، و ضریب پهنای خاموشی برابر ۱۰۰ در نظر گرفته شده است.

جدول ۳: نتیجه الگوریتم ۱ برای مسئله ۳.

الف) ساختار شبکه پس از پایان مرحله تعیین واحدهای مخفی

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۰	۸۰	۹۹/۳۷۵	۱۰۰
۲	۸	۶۴	۹۶/۸۷۵	۱۰۰
۳	۸	۶۴	۹۶/۸۷۵	۱۰۰
۴	۷	۵۶	۸۷/۷۵	۹۰
۵	۱۱	۸۸	۹۹/۳۷۵	۱۰۰
۶	۷	۵۶	۸۳/۷۵	۸۰
متوسط	۸/۵	۶۸	۹۴	۹۵

ب) ساختار شبکه پس از پایان مرحله تعیین وزنهای

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۸	۱۵	۹۹/۳۷۵	۱۰۰
۲	۸	۲۶	۹۵/۰	۱۰۰
۳	۸	۲۶	۹۵/۶۲۵	۱۰۰
۴	۷	۱۸	۹۶/۸۷۵	۱۰۰
۵	۹	۱۸	۹۹/۳۷۵	۱۰۰
۶	۷	۱۸	۹۶/۸۷۵	۱۰۰
متوسط	۷/۸۳۳	۲۰/۱۶	۹۷/۱۸۷	۱۰۰

کاهش نسبت متوسط تعداد وزن بر واحد مخفی، ابر صفحه های تولید شده توسط واحدهای مخفی بسیارهوار میگردند. نکته جالب دیگری که از جداول فوق میتوان نتیجه گرفت تعداد متوسط وزنها متصل به هر واحد مخفی میباشد. در مسئله ۱ و ۲ همه واحدهای مخفی کمتر از ۳ وزن ورودی دارند و در مسئله ۱ بیشتر خطوط تولید شده توسط واحدهای مخفی موازی یکی از دو محور می باشد و تعداد کمی از آنها خطوط مورب هستند که در دسته بندی الگوها نقش ندارند. در مسئله ۳ هر واحد مخفی تقریباً ۳ وزن دارد یعنی هر واحد مخفی بطور متوسط بیشتر از نیمی از ویژگیهایی در نظر گرفته شده برای شناسایی استفاده نمیکند. در مورد مسئله ۴ نسبت متوسط تعداد وزن بر واحد مخفی برای آزمایش های مختلف، متفاوت می باشد. شبکه های تولید شده توسط الگوریتم ۱ این واقعیت را نشان می دهند که بیشتر ویژگیهای مربوط به الگوها (ورودیهای شبکه عصبی) برای شناسایی مورد استفاده قرار نمی گیرند.

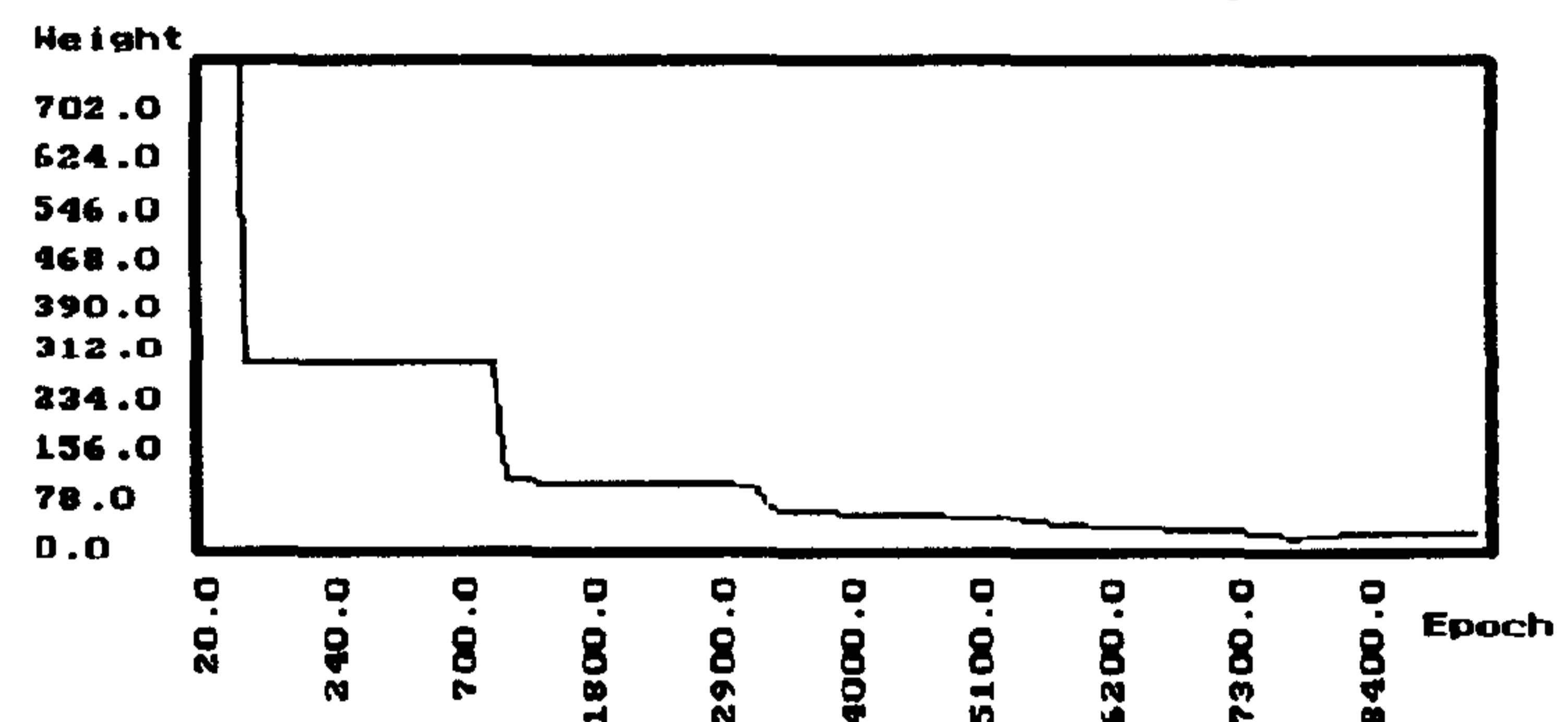
ب) الگوریتم ۲: الگوریتم ۲ شکل خلاصه تری از الگوریتم ۱ است که در آن تعداد واحدهای مخفی بصورت تقریبی و در ابتدا توسط طراح شبکه مشخص میگردد. این الگوریتم فقط طریقه اتصال بین لایه ورودی و لایه مخفی را تعیین میکند. در این الگوریتم تعیین تعداد واحدهای لایه مخفی بطور غیر مستقیم و از طریق تعیین تعداد وزنها لایه مخفی شبکه انجام میگردد. این الگوریتم نیز مانند الگوریتم ۱ شبکه ای با اتصالات محلی تولید میکند اگر تعداد واحدهای مخفی اولیه برای دو الگوریتم یکسان باشد تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۲ بمراتب بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ میباشد این بدان دلیل است که حذف واحدهای لایه مخفی بطور غیر مستقیم از طریق حذف وزنها لایه مخفی انجام می گیرد. الگوریتم ۲ روی چهار مسئله تعریف شده پیاده سازی شده و نتایج آزمایشات در جداول (۵) تا (۸) آورده شده است. برای آزمایشاتی که نتایج آنها در قسمت (الف) جداول (۵) تا (۸) آمده است تعداد واحدهای مخفی اولیه نسبت به تعداد واحدهای اولیه الگوریتم ۱ کوچک در نظر گرفته شده است. برای آزمایشاتی که نتایج آنها در

از اتوماتان HULA(2, 20, 60)-HWLA (2, 20, N_w) استفاده گردیده است. N_w تعداد وزنها شبکه در پایان الگوریتم بقا میباشد. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۰/۱ و ضریب پهنای خاموشی برابر ۰/۱ در نظر گرفته شده است.

ب) ساختار شبکه پس از پایان مرحله تعیین وزنها

شبکه	واحد مخفی	تعداد وزنها لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۲	۲۴	۱۰۰	۱۰۰
۲	۷	۲۷	۱۰۰	۱۰۰
۳	۹	۵۱	۱۰۰	۹۰
۴	۱۳	۵۶	۱۰۰	۱۰۰
۵	۱۱	۵۸	۱۰۰	۱۰۰
۶	۱۳	۵۶	۱۰۰	۱۰۰
متوسط	۱۰/۸۳	۴۵/۳۳	۱۰۰	۹۸/۳۳

شکل (۱۳) منحنی تغییرات تعداد وزنها لایه مخفی شبکه را براساس تکرار آرایه الگو برای شبکه شماره ۱ جدول (۴-ب) نشان میدهد.



شکل ۱۳: تغییرات تعداد وزنها لایه مخفی شبکه براساس تکرار آرایه الگو.

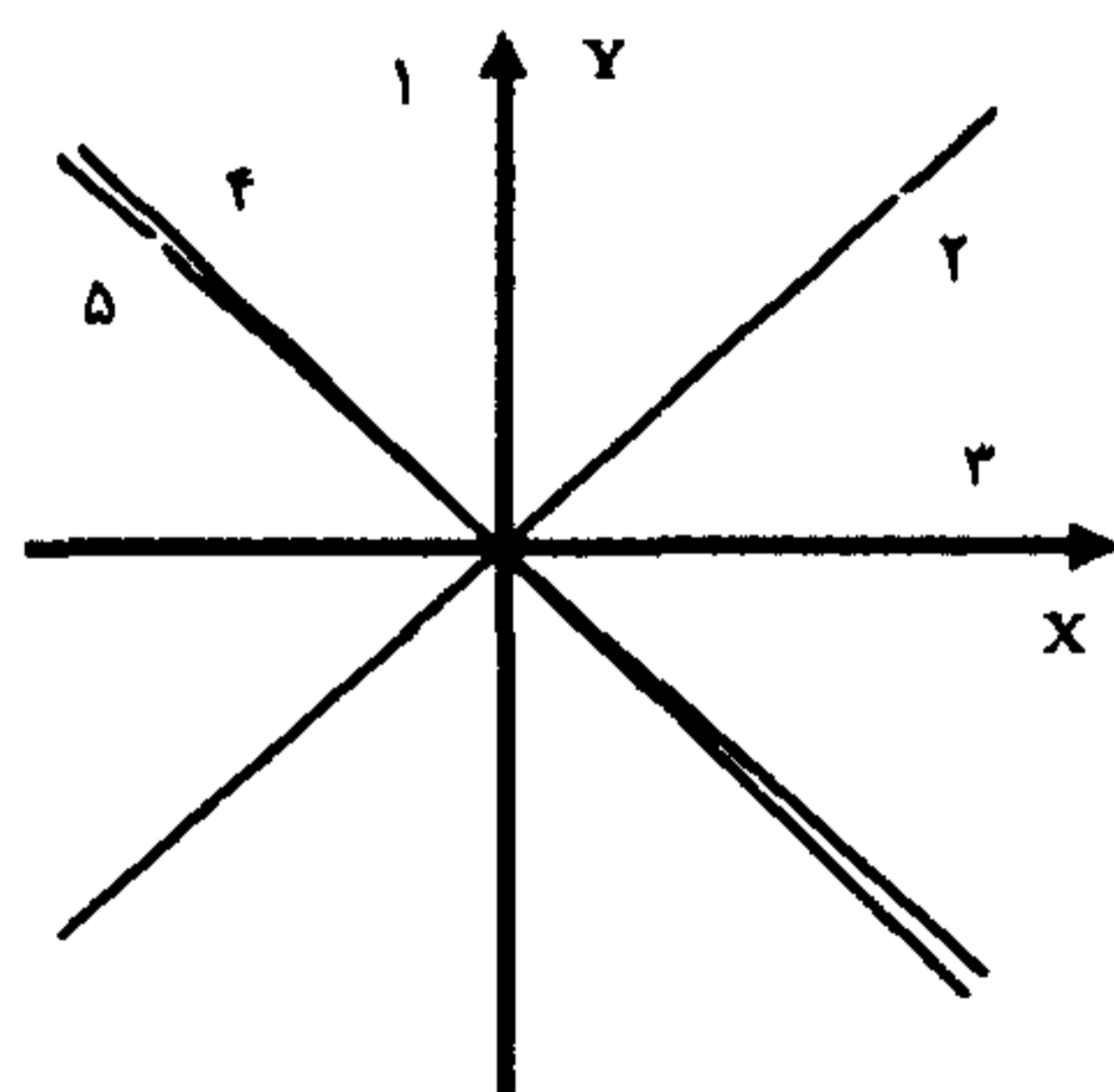
همانطور که در جداول فوق مشخص است تعداد وزنها پس از مرحله تعیین تعداد وزنها لایه مخفی بسیار کمتر از تعداد وزنها پس از مرحله تعیین واحدهای لایه مخفی میباشد و همچنین قدرت تعمیم شبکه در پایان مرحله تعیین وزنها نیز در بیشتر موارد بیشتر از قدرت تعمیم شبکه در پایان مرحله تعیین واحدها است. این بدان دلیل است که با

اتوماتان HWLA (2, 7, 117) استفاده شده است. نرخ یادگیری برابر ۰/۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن ۰/۹، ضریب پهنای روشنی برابر ۱۲، و ضریب پهنای خاموشی برابر ۱۲ می‌باشد.

با مقایسه جداول (۱) و (۵) میتوان به نکات جالب توجه ای اشاره نمود. اول آنکه متوسط تعداد وزن ورودی بر هر واحد مخفی در الگوریتم ۱ و الگوریتم ۲ تقریباً مساوی است. دوم آنکه هیچکدام از شبیه‌سازها، شبکه با ساختار بهینه (شبکه سه لایه با دو واحد مخفی و چهار وزن بین لایه ورودی و لایه مخفی که در ابتدای مقاله بدان اشاره شد) را تولید نکرده‌اند. علت این امر آنستکه تعداد زیادی از واحدهای مخفی خروجیهای مشابه‌ای را تولید میکنند و بهمین دلیل تعدادی از واحدهای مخفی زائد هستند. برای مثال شبکه شماره ۱ را در جدول (۵-الف) در نظر بگیرید. ماتریس وزنه‌های لایه مخفی در پایان اجرای الگوریتم این شبکه به صورت زیر میباشد.

$$W = \begin{bmatrix} 0 & 0 & -2.44 \\ 0 & 3.124 & -2.68 \\ 0 & -1.69 & 0 \\ 0 & -3.01 & -2.69 \\ 0 & -1.62 & -1.16 \end{bmatrix}$$

در ماتریس فوق عنصر w_{j0} مقدار آستانه واحد مخفی شماره j ، عنصر w_{jm} وزن اتصال واحد مخفی شماره j به واحد ورودی m میباشد. در این شبکه واحد ورودی ۱ ویژگی x و واحد ورودی ۲ ویژگی y را نشان میدهند. خطوط تولید شده توسط واحدهای مخفی این شبکه در شکل (۱۴) نشان داده شده است.



شکل ۱۴: خطوط تولید شده توسط واحدهای مخفی شبکه ۱.

در شکل (۲۱) خط شماره ۲ توسط واحد مخفی شماره ۲ تولید شده است (خطوط با شماره‌های ۱ و ۳ بترتیب روی محورهای X و Y قرار گرفته‌اند). با توجه به شکل فوق

قسمت (ب) جداول (۵) تا (۸) آمده است تعداد واحدهای مخفی اولیه الگوریتم ۲ مساوی تعداد واحدهای مخفی اولیه الگوریتم ۱ انتخاب شده است. لازم بذکر است که در تمامی آزمایشهای زیر ۱۰۰۰۰ بار مجموعه آموزشی به شبکه ارائه شده است.

جدول ۵: نتیجه شبیه‌سازی الگوریتم ۲ برای مثال ۱. الف) نتیجه الگوریتم ۲ برای مثال ۱ برای اتوماتان HWLA (2, 7, 18)

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۴	۵	۱۰۰	۱۰۰
۲	۶	۷	۱۰۰	۱۰۰
۳	۶	۷	۱۰۰	۱۰۰
۴	۵	۶	۱۰۰	۱۰۰
۵	۶	۶	۱۰۰	۱۰۰
۶	۴	۵	۱۰۰	۱۰۰
متوسط	۵/۸۳	۶	۱۰۰	۱۰۰

ب) نتیجه الگوریتم ۲ برای مثال ۱ برای اتوماتان HWLA (2, 7, 117)

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۲۶	۴۹	۱۰۰	۱۰۰
۲	۱۴	۱۸	۱۰۰	۱۰۰
۳	۱۸	۱۹	۱۰۰	۱۰۰
۴	۱۵	۲۰	۱۰۰	۱۰۰
۵	۱۹	۳۵	۱۰۰	۱۰۰
۶	۲۲	۴۲	۹۵	۱۰۰
متوسط	۱۹	۳۰/۵	۱۰۰	۹۹/۱۶

دسته‌بندی چهار دسته از الگوها: الگوریتم ۲ روی ۶ شبکه با وزنه‌های اولیه تصادفی آزمایش شده‌اند و نتایج آن در جدول (۵) نشان داده شده است. برای تمامی این شبکه‌ها در جدول (۵-الف) از اتوماتان HWLA (2, 7, 18) و در جدول (۵-ب) از

وزنهای اولیه تصادفی آزمایش شده‌اند و نتایج آن در جدول (۶) نشان داده شده است. برای تمامی این شبکه‌ها در قسمت (الف) از اتوماتان (HWLA (2, 7, 18) و در قسمت (ب) از اتوماتان (HWLA (2, 7, 180) استفاده شده است. نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۱۵، و ضریب پهنای خاموشی برابر ۱۵ در نظر گرفته شده است.

با توجه به جداول (۲) و (۶) میتوان به نکات زیر اشاره نمود. اول آنکه، نسبت متوسط تعداد وزن برواحد مخفی در دو جدول مذکور تقریباً یکسان است. این نسبت در جداول (۲-ب) و (۶-الف) حدود دو و در جدول (۶-ب) مساوی ۲ است. یعنی شبکه های آموزش داده شده در جدول (۶-ب) از یکی از وزنهای ورودی خود (وزن آستانه، وزن متصل به ورودی X و یا وزن متصل به ورودی Y) برای شناسایی دسته ها از یکدیگر استفاده نمیکند. با توجه به ناحیه رسم شده در شکل (۹) که دودسته الگوها را از هم جدا میسازد میتوان نتیجه گرفت که این خطوط یا موازی یکی از دو محور (معادلات خطوط بفرم $Y = C$ و یا $X = C$) و یا مورب (معادلات خطوط بفرم $Y = mX + C$) هستند. در صورتیکه معادله خط بصورت $X = C$ باشد، واحد مخفی نیاز به دو وزن (وزن آستانه و وزن متصل به ورودی X) دارد و در حالتیکه معادله خط بصورت $Y = C$ است واحد مخفی نیاز به دو وزن (وزن آستانه و وزن متصل به ورودی Y) دارد و در حالتیکه معادله خط بصورت $Y = mX + C$ است واحد مخفی نیاز به سه وزن ورودی (وزن آستانه، وزن متصل به ورودی X و وزن متصل به ورودی Y) دارد. با توجه به آزمایشهای انجام شده، واحدهای مخفی تعیین شده عموماً یکی از دو خط $X = C$ یا $Y = C$ را نشان می دهند. ناحیه های تولید شده مطابق شکل (۱۵) خواهد بود. شبکه ای که این نواحی را تولید می کند بایستی حداقل چهار واحد مخفی داشته باشد. این ساختار از ساختار بهینه (شبکه عصبی سه لایه با ۳ واحد مخفی و ۸ اتصال) یک واحد مخفی بیشتر دارد اما تعداد وزنهای این دو شبکه یکسان است. اما شبکه های تولید شده دارای تعداد واحدهای بیشتری هستند که هر کدام از آنها یکی از چهار خط زیر را تولید

خطوط با شماره های ۴ و ۵ بسیار نزدیک بهم هستند. این بدان معنی است که اطلاعات تقریباً یکسانی در وزنهای آنها ذخیره شده است (این دو واحد دارای واریانس فعالیت بالا و وزنهای با مقادیر بزرگ میباشند) وبهین دلیل میتوان یکی از این دو واحد را حذف نمود. در نتیجه شبکه هرس شده دارای ۴ واحد مخفی و ۶ اتصال خواهد بود. شبکه حاصله هنوز دارای دو واحد زائد با شماره های ۲ و ۴ می باشد که الگوریتم پیشنهادی با معیار بکار برده شده قادر به حذف آنها نشده است.

جدول ۶: نتیجه شبیه سازی الگوریتم ۲ برای مثال ۲. الف) نتیجه الگوریتم ۲ برای مثال ۲ برای اتوماتان (HWLA (2, 7, 18)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۵	۱۳	۱۰۰	۱۰۰
۲	۵	۱۱	۱۰۰	۱۰۰
۳	۵	۱۳	۱۰۰	۱۰۰
۴	۵	۱۲	۱۰۰	۱۰۰
۵	۴	۱۰	۱۰۰	۱۰۰
۶	۵	۱۲	۱۰۰	۱۰۰
متوسط	۴/۸۸	۱۱/۸۳	۱۰۰	۱۰۰

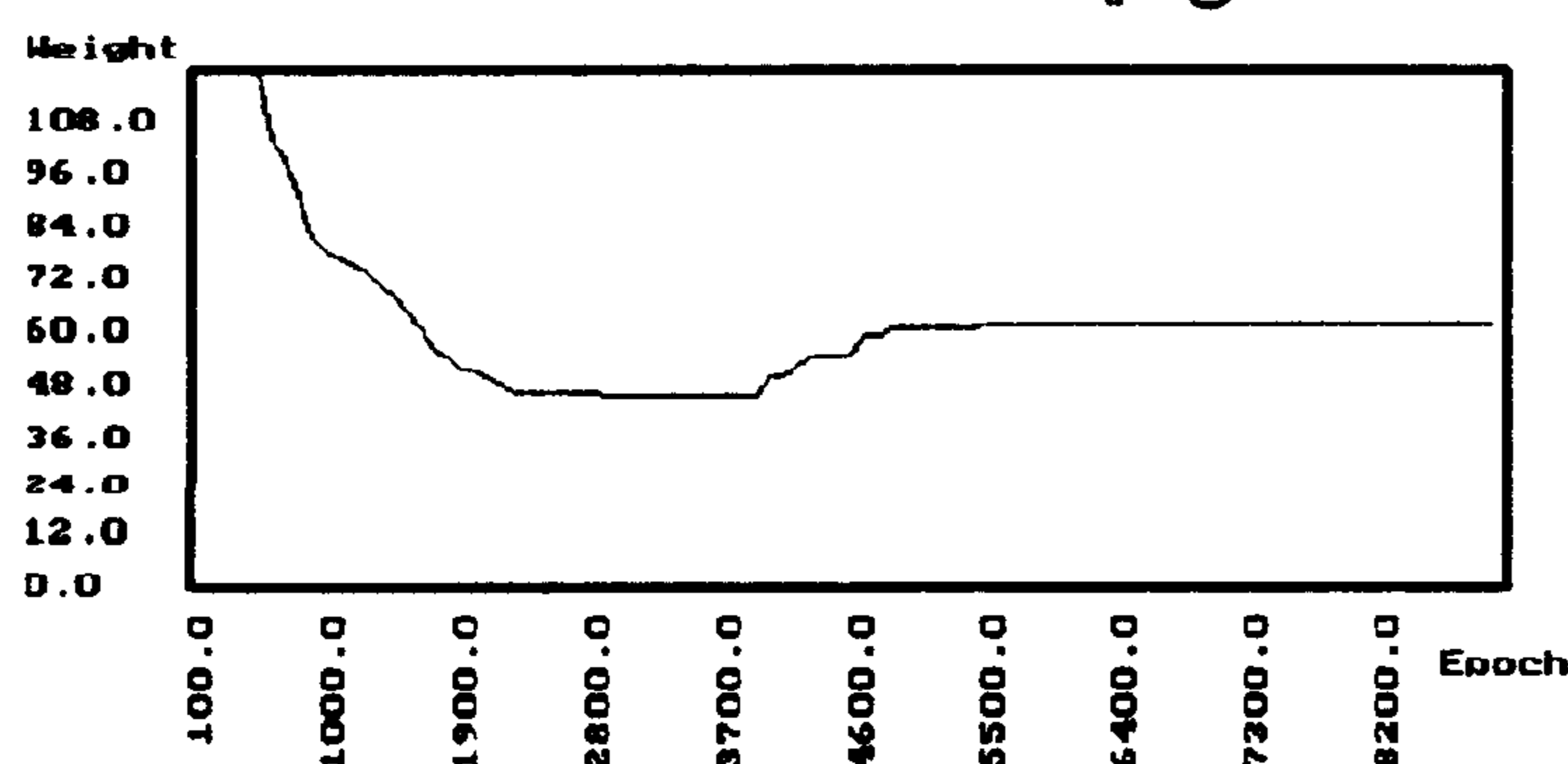
ب) نتیجه الگوریتم ۲ برای مثال ۲ برای اتوماتان (HWLA (2, 7, 180)

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۴۸	۹۶	۹۸	۹۶
۲	۳۹	۷۸	۱۰۰	۱۰۰
۳	۴۸	۹۶	۱۰۰	۱۰۰
۴	۴۴	۸۸	۱۰۰	۱۰۰
۵	۴۲	۸۴	۱۰۰	۱۰۰
۶	۳۸	۷۶	۱۰۰	۱۰۰
متوسط	۴۳/۱۶	۸۶/۳۳	۹۹/۶۶	۹۹/۳۳

دسته بندی دودسته از الگوها: الگوریتم ۲ روی ۶ شبکه با

مسئله تشخیص اعداد چاپی فارسی: الگوریتم ۲ روی ۶ شبکه با وزنهای اولیه تصادفی آزمایش شده‌اند و نتایج آن در جدول (۷) نشان داده شده است. برای تمامی این شبکه‌ها برای قسمت (الف) از اتوماتان $HWLA(2, 7, 120)$ و برای قسمت (الف) از اتوماتان $HWLA(2, 7, 480)$ استفاده شده است. برای انجام این آزمایشها، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۱۵، و ضریب پهنای خاموشی برابر ۱۵ در نظر گرفته شده است.

شکل (۱۶) منحنی تغییرات تعداد وزنهای لایه مخفی شبکه را براساس تکرار ارایه الگو برای شبکه شماره ۱ جدول (۷-الف) نشان میدهد.

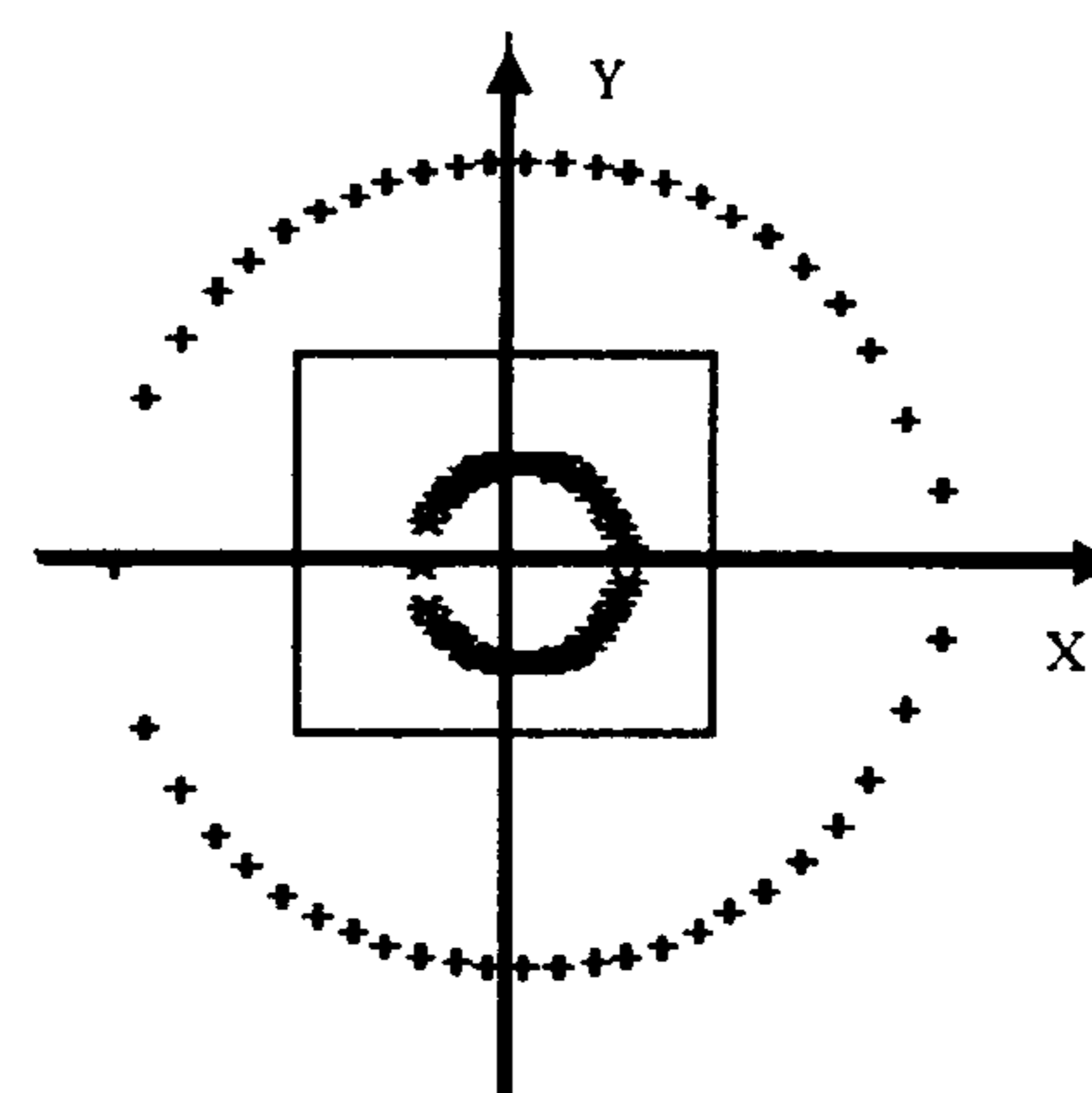


شکل ۱۶: تغییرات تعداد وزنهای لایه مخفی شبکه براساس تکرار ارایه الگو.

باتوجه به جداول (۳-ب) و جدول (۷) میتوان به نکات زیر اشاره نمود. نسبت متوسط تعداد وزن ورودی به واحد مخفی در الگوریتمهای ۱ و ۲ تقریباً مساوی هستند. بدلیل اینکه در حین کاهش تعداد وزنهای شبکه، تعدادی از واحدهای مخفی، خروجیهای مشابهی را تولید میکنند تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۲ بسیار بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ میباشد و الگوریتم قادر به حذف واحدهای مشابه نیست. اطلاعات ذخیره شده در بردار وزن واحدهایی که خروجیهای مشابه را تولید میکنند تقریباً یکسان است و میتوان همه این واحدهای مشابه بجز یکی از آنها را حذف نمود. البته این کار میتواند بعد از تعیین ساختار شبکه توسط الگوریتم دیگری انجام گیرد.

مسئله تشخیص اعداد لاتین: الگوریتم ۲ روی ۶ شبکه با وزنهای اولیه تصادفی آزمایش شده‌اند و نتایج آن در جدول (۸)

می‌کنند و الگوریتم قادر به تشخیص و حذف واحدهای تکراری نیست.



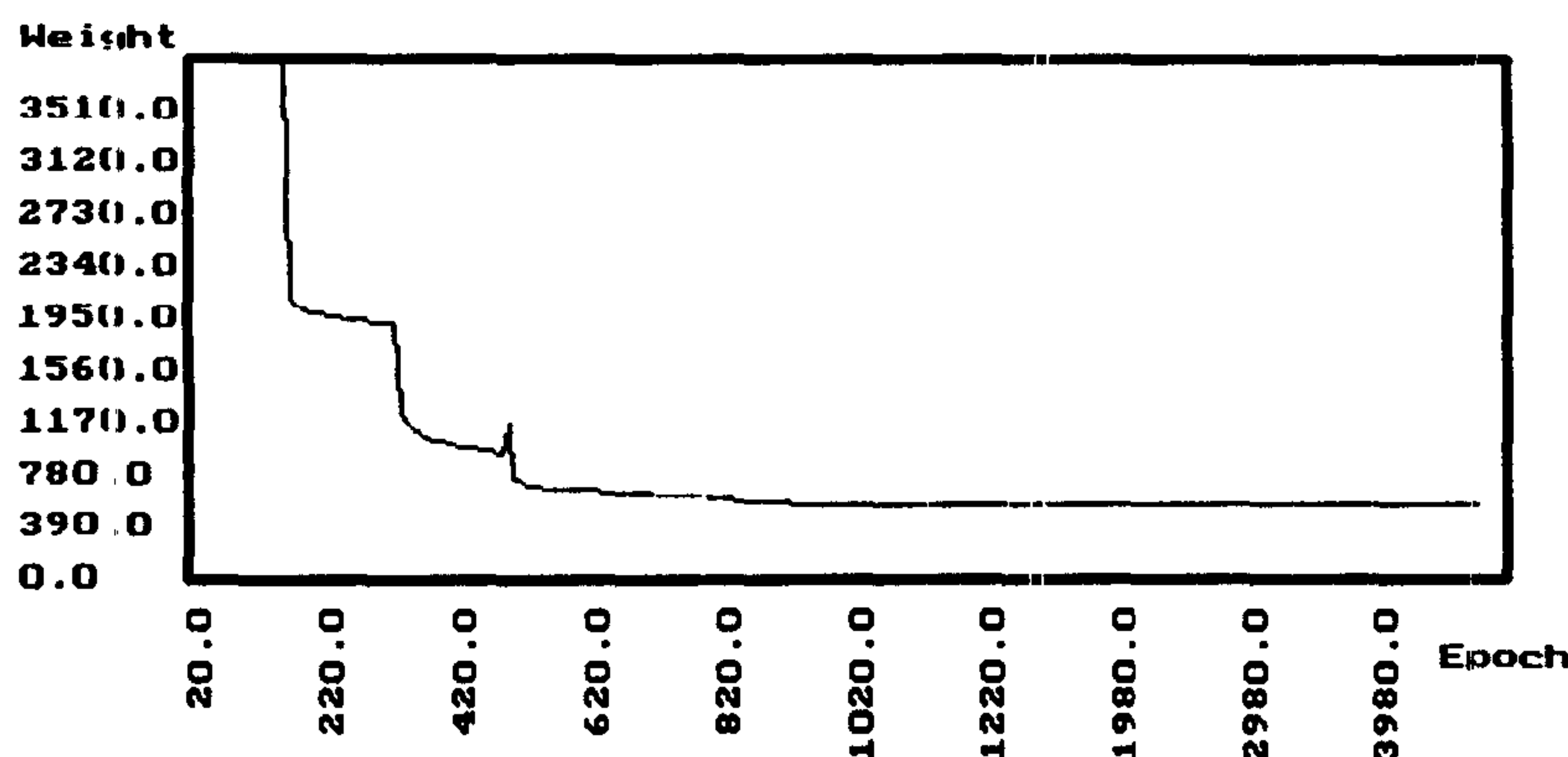
شکل ۱۵: نواحی تولید شده برای شبکه های جدول (۶-ب) برای مثال ۲.

جدول ۷: نتیجه شبیه سازی الگوریتم ۲ برای مثال ۳. الف) نتیجه الگوریتم ۲ برای مثال ۳ برای اتوماتان $HWLA(2, 7, 120)$

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۴	۶۱	۹۸/۷۵	۱۰۰
۲	۱۵	۶۲	۹۸/۱۲۵	۱۰۰
۳	۱۵	۷۰	۹۹/۳۷۵	۱۰۰
۴	۱۵	۶۱	۹۹/۳۷۵	۱۰۰
۵	۱۵	۶۰	۹۹/۳۷۵	۱۰۰
۶	۱۵	۷۴	۹۸/۷۵	۱۰۰
متوسط	۱۴/۸۳	۶۴/۶۶	۹۸/۹۵	۱۰۰

ب) نتیجه الگوریتم ۲ برای مثال ۳ برای اتوماتان $HWLA(2, 7, 480)$

شبکه	واحد مخفی	تعداد وزنهای لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۲۸	۸۲	۱۰۰	۱۰۰
۲	۳۵	۹۸	۱۰۰	۱۰۰
۳	۳۰	۶۵	۱۰۰	۱۰۰
۴	۲۵	۶۸	۹۹	۱۰۰
۵	۳۴	۹۰	۱۰۰	۱۰۰
۶	۳۰	۷۰	۱۰۰	۱۰۰
متوسط	۳۰/۳۳	۷۸/۸۳	۹۹/۸۴	۱۰۰



شکل ۱۷: تغییرات تعداد وزنه‌های لایه مخفی شبکه براساس تکرار آرایه‌الگو.

نشان داده شده است. برای تمامی این شبکه‌ها برای قسمت (الف) از اتوماتان (2, 20, 975) HWLA و برای قسمت (ب) از اتوماتان (2, 20, 3900) HWLA استفاده شده است. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۱، و ضریب پهنای خاموشی برابر ۱/۵ در نظر گرفته شده است.

جدول ۸: نتیجه شبیه سازی الگوریتم ۲ برای مثال ۴. الف) نتیجه الگوریتم ۲ برای مثال ۳ برای اتوماتان HWLA (2, 7, 875)

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۵	۵۱۲	۱۰۰	۱۰۰
۲	۱۵	۴۸۸	۱۰۰	۱۰۰
۳	۱۵	۵۲۷	۱۰۰	۱۰۰
۴	۱۵	۵۲۲	۱۰۰	۱۰۰
۵	۱۵	۵۰۷	۱۰۰	۱۰۰
۶	۱۵	۵۱۵	۱۰۰	۱۰۰
متوسط	۱۵	۵۱۱/۸۳	۱۰۰	۱۰۰

با توجه به جداول (۴-ب) و (۸) نتایج زیر را می‌توان گرفت. همانند مسائل قبل تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۲ بمراتب بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ می‌باشد. نکته دیگر اینکه شبکه‌های تولید شده توسط الگوریتم‌های ۱ و ۲ برای مسائل ۱ و ۲ تقریباً از سی درصد از ورودیهایش و برای مسئله ۳ بیشتر از نیمی از ورودیهایش و برای مسئله ۴ تقریباً از هشتاد درصد از ورودیهایش استفاده نمی‌کند. بنابراین می‌توان انتظار داشت دو الگوریتم ۱ و ۲ برای شبکه‌هایی مناسب تر هستند که دارای ورودیهای زیادی باشند.

ب) نتیجه الگوریتم ۲ برای مثال ۳ برای اتوماتان HWLA (2, 7, 3900)

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۶۰	۴۴۰	۱۰۰	۹۰
۲	۶۰	۵۳۷	۱۰۰	۱۰۰
۳	۶۰	۵۴۱	۱۰۰	۱۰۰
۴	۶۰	۵۸۲	۱۰۰	۱۰۰
۵	۶۰	۵۷۱	۱۰۰	۱۰۰
۶	۶۰	۵۳۱	۱۰۰	۱۰۰
متوسط	۶۰	۵۳۳/۶۶	۱۰۰	۹۸/۳۴

ج) الگوریتم ۳: الگوریتم ۳ بمنظور تعیین تعداد وزنه‌ها، وزنه‌ها را بصورت عمومی با در نظر گرفتن همه وزنه‌های همه واحدهای مخفی روشن مقایسه میکند و بر اساس نتایج این مقایسه‌ها عمل افزایش یا کاهش تعداد وزنه‌های لایه مخفی را انجام می‌دهد. در این الگوریتم تعیین تعداد واحدهای لایه مخفی بطور غیر مستقیم و از طریق تعیین تعداد وزنه‌های لایه مخفی شبکه انجام می‌گیرد این الگوریتم نیز مانند دو الگوریتم ۱ و ۲ شبکه‌ای با اتصالات محلی تولید میکند (تذکر ۱). در صورتیکه تعداد واحدهای مخفی اولیه برای دو الگوریتم ۱ و ۳ یکسان باشند تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۳ بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ می‌باشد زیرا در الگوریتم ۳ بصورت صریح واحدی حذف نمی‌گردد. بدلیل استفاده از مزایای آموزش شبکه‌های بزرگ، تعداد واحدهای مخفی در ابتدای آموزش شبکه توسط الگوریتم ۳ بیشتر از تعداد تعیین شده توسط الگوریتم ۱ در نظر گرفته شده

شکل (۱۷) منحنی تغییرات تعداد وزنه‌های لایه مخفی شبکه را براساس تکرار آرایه الگو برای شبکه شماره ۱ جدول (۸-ب) نشان میدهد.

جدول ۱۰: نتیجه شبیه سازی الگوریتم ۳ برای مثال ۲.

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۲۵	۸۲	۱۰۰	۱۰۰
۲	۳۵	۹۸	۱۰۰	۱۰۰
۳	۳۰	۶۵	۱۰۰	۱۰۰
۴	۲۵	۶۸	۱۰۰	۱۰۰
۵	۳۴	۹۰	۱۰۰	۱۰۰
۶	۳۷	۸۳	۱۰۰	۱۰۰
متوسط	۳۱	۸۱	۱۰۰	۱۰۰

مسئله تشخیص اعداد چاپی فارسی: الگوریتم ۳ روی ۶ شبکه با وزنه‌های اولیه تصادفی آزمایش شده‌اند و نتایج آن در جدول (۱۱) نشان داده شده است. برای تمامی این شبکه ها از اتوماتان (2, 7, 480) HWLA استفاده شده است. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۴، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۰۹، ضریب پهنای روشنی برابر ۰/۱۵، و ضریب پهنای خاموشی برابر ۰/۱۵ در نظر گرفته شده است.

جدول ۱۱: نتیجه شبیه سازی الگوریتم ۳ برای مثال ۳.

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۵۶	۱۵۸	۹۹/۳۷۵	۱۰۰
۲	۵۷	۱۴۴	۹۹/۳۷۵	۱۰۰
۳	۵۵	۱۴۹	۹۹/۳۷۵	۱۰۰
۴	۵۶	۱۶۱	۹۹/۳۷۵	۱۰۰
۵	۵۰	۱۱۱	۹۸/۷۵	۱۰۰
۶	۵۶	۱۶۰	۹۸/۷۵	۱۰۰
متوسط	۵۵	۱۴۷/۱۶	۹۹/۱۶	۱۰۰

شکل (۱۸) منحنی تغییرات تعداد وزنه‌های لایه مخفی شبکه را براساس تکرار ارایه الگو برای شبکه شماره ۱ جدول (۱۱) نشان میدهد.

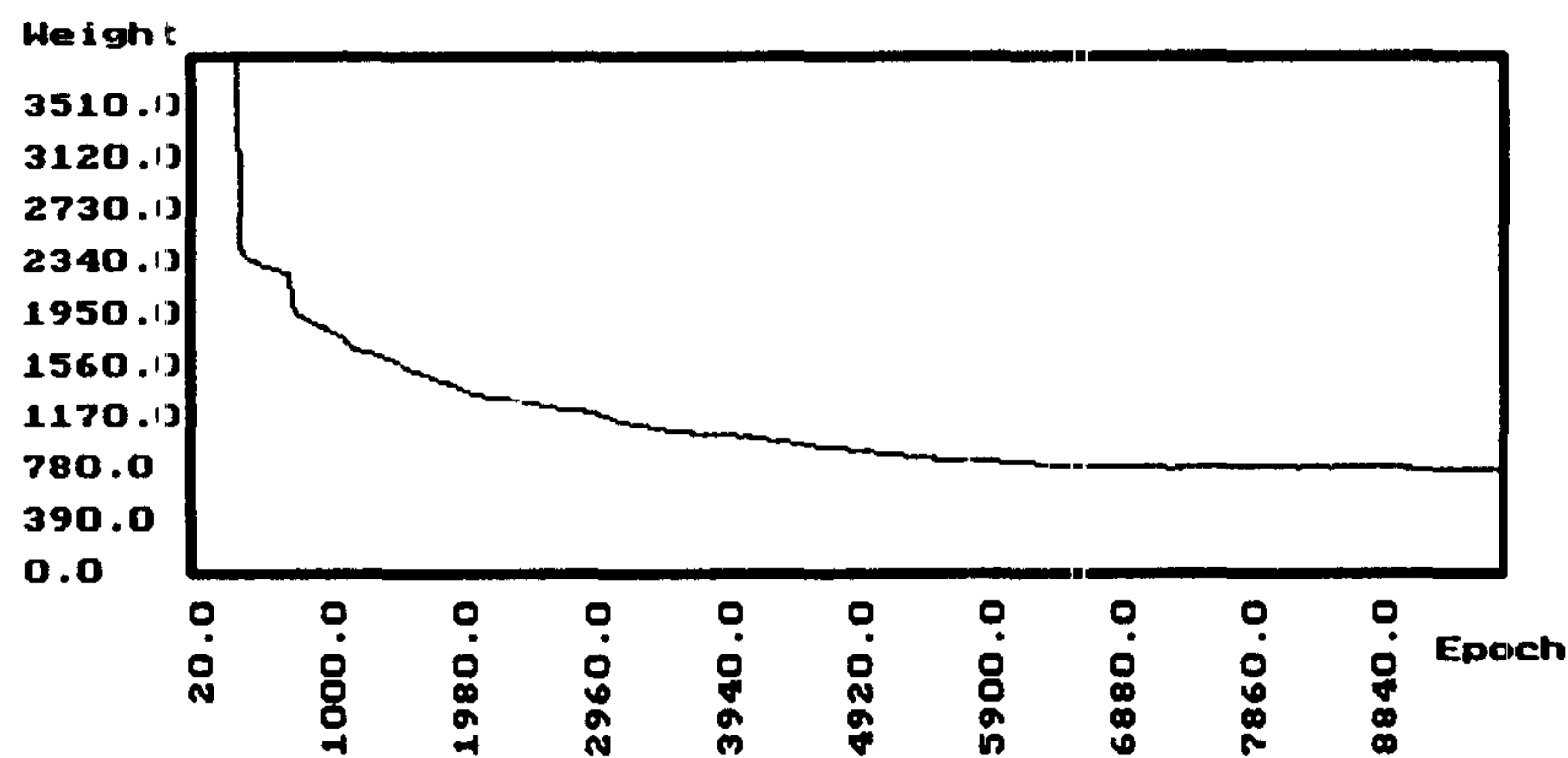
است. این الگوریتم روی مسائل نمونه پیاده سازی شده و نتایج آزمایشات در جداول (۹) تا (۱۲) آورده شده است. لازم بذکر است که در تمامی آزمایشهای زیر برای آموزش و تعیین ساختار شبکه های عصبی ۱۰۰۰۰ بار مجموعه آموزشی به شبکه ارایه شده است.

دسته بندی چهار دسته از الگوها: الگوریتم ۳ روی ۶ شبکه با وزنه‌های اولیه تصادفی آزمایش شده اند و نتایج آن در جدول (۹) نشان داده شده است. برای تمامی این شبکه ها از اتوماتان (2, 7, 117) HWLA استفاده شده است. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۴، ضریب گشتاور برابر ۰/۰۶، ضریب کاهش وزن برابر ۰/۰۹، ضریب پهنای روشنی برابر ۰/۱۵، و ضریب پهنای خاموشی برابر ۰/۱۵ در نظر گرفته شده است.

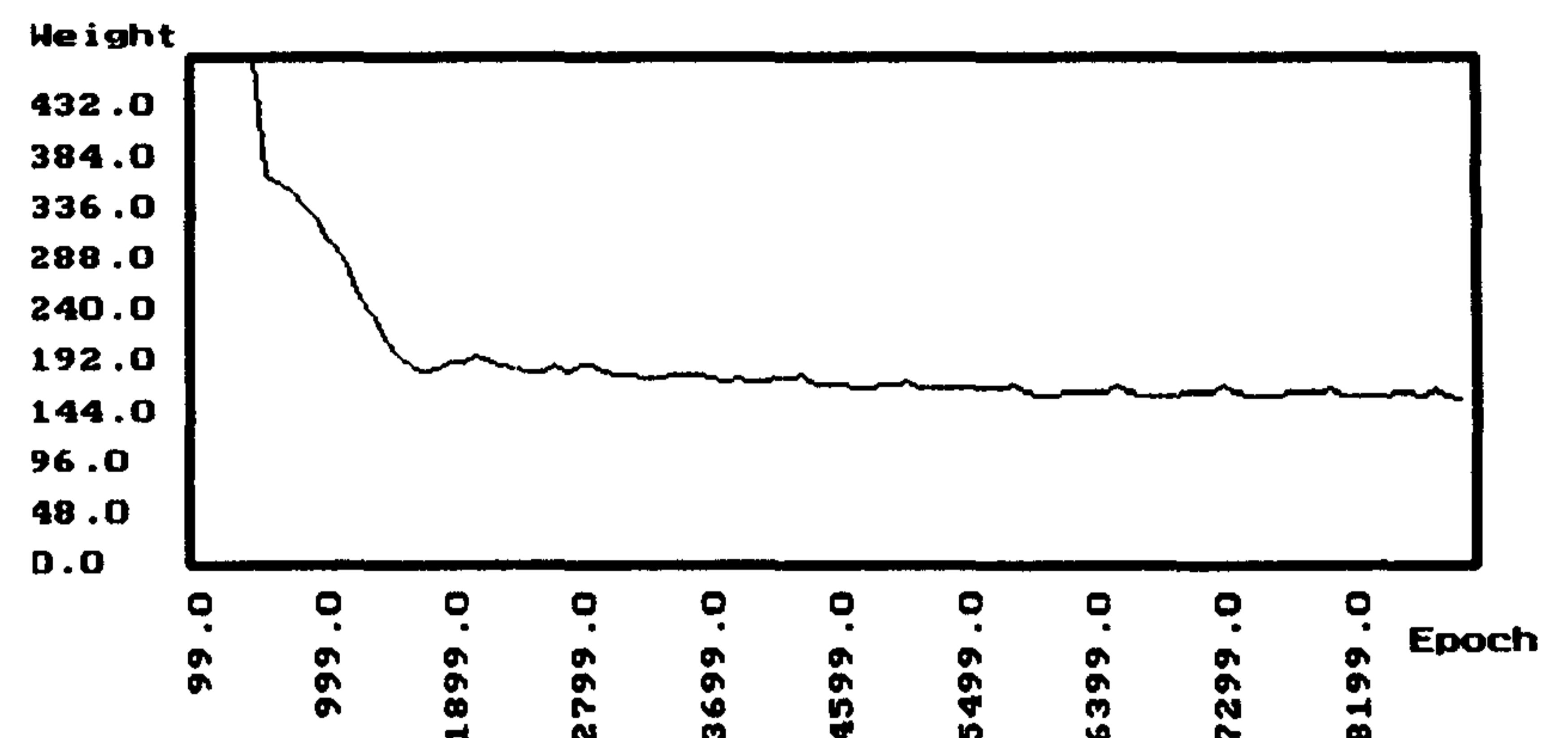
جدول ۹: نتیجه شبیه سازی الگوریتم ۳ برای مثال ۱.

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۱۷	۲۹	۱۰۰	۱۰۰
۲	۱۷	۲۹	۱۰۰	۱۰۰
۳	۱۷	۳۰	۱۰۰	۹۵
۴	۱۷	۲۹	۱۰۰	۹۷/۵
۵	۲۰	۲۷	۱۰۰	۱۰۰
۶	۲۰	۳۴	۱۰۰	۱۰۰
متوسط	۱۸	۲۹/۶۶	۱۰۰	۹۸/۷۵

دسته بندی دو دسته از الگوها: الگوریتم ۲ روی ۶ شبکه با وزنه‌های اولیه تصادفی آزمایش شده اند و نتایج آن در جدول (۱۰) نشان داده شده است. برای تمامی این شبکه‌ها از اتوماتان (2, 7, 180) HWLA استفاده شده است. لازم بذکر است که برای انجام آزمایشهای مختلف، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۰۹، ضریب پهنای روشنی برابر ۰/۲۵، و ضریب پهنای خاموشی برابر ۰/۲۵ در نظر گرفته شده است.



شکل ۱۹: تغییرات تعداد وزنه های لایه مخفی شبکه براساس تکرار ارایه الگو.



شکل ۱۸: تغییرات تعداد وزنه های لایه مخفی شبکه براساس تکرار ارایه الگو.

تذکر ۲: نتایج آزمایشات برای هر سه الگوریتم پیشنهادی نشان می دهند با افزایش تعداد واحدهای ورودی شبکه، متوسط نسبت تعداد وزنه های روشن به کل وزنه های آن واحد در ابتدای آموزش (تعداد واحدهای ورودی شبکه) برای واحدهای مخفی کاهش می یابد. با توجه به اینکه حذف یک وزن بین یک واحد ورودی (ویژگی) و یک واحد مخفی بمنزله از بین رفتن وابستگی ابر صفحه تولید شده توسط واحد مخفی به آن ویژگی می گردد (زمانیکه وزنه های متصل به یک واحد ورودی حذف گردد آن واحد ورودی (ویژگی) برای شناسایی مهم نیست و می تواند حذف شود). این بدان معنی است که این الگوریتم ها مواقعی کارایی بالایی خواهند داشت که تعداد واحدهای ورودی شبکه زیاد باشد. از این خاصیت می توان برای تعیین یک زیر مجموعه از مجموعه ویژگی های مناسب برای جداسازی الگوها استفاده نمود. بعنوان مثال، در شناسایی نوری حروف، ویژگی های گشتاورها، گشتاورهای مرکزی، گشتاورهای مقیاس شده و ثابت های گشتاوری از تصویر حروف استخراج شده از تصویر میتوانند مورد استفاده قرار گیرند. برای تعیین بهترین زیرمجموعه از مجموعه ویژگی های فوق، میتوان تمام این ویژگی ها را به شبکه ارایه نموده و از طریق الگوریتم تعیین ساختار شبکه، بهترین زیر مجموعه ویژگی ها را برای کاربرد مورد نظر انتخاب نمود.

تذکر ۳: در الگوریتم بقا (قسمت اول الگوریتم ۱)، در ابتدا واریانس فعالیت واحدهای مخفی تقریباً مشابه هستند، و بیشتر آنها مناسب تشخیص داده شده و روشن می مانند. اما بتدریج در حین آموزش، واریانس فعالیت تعدادی از

مسئله تشخیص اعداد لا تین: الگوریتم ۳ روی ۶ شبکه با وزنه های اولیه تصادفی آزمایش شده اند و نتایج آن در جدول (۱۲) نشان داده شده است. برای تمامی این شبکه ها از اتوماتان (2, 20, 3900) HWLA استفاده شده است. لازم بذکر است که برای انجام آزمایش های مختلف، نرخ یادگیری برابر ۰/۰۱، ضریب گشتاور برابر صفر، ضریب کاهش وزن برابر ۰/۹، ضریب پهنای روشنی برابر ۰/۱۲۵، و ضریب پهنای خاموشی برابر ۰/۱۲۵ در نظر گرفته شده است.

جدول ۱۲: نتیجه شبیه سازی الگوریتم ۳ برای مثال ۴.

شبکه	واحد مخفی	تعداد وزنه های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۶۰	۷۷۱	۱۰۰	۱۰۰
۲	۶۰	۸۰۲	۱۰۰	۱۰۰
۳	۶۰	۷۷۴	۱۰۰	۱۰۰
۴	۶۰	۷۸۴	۱۰۰	۱۰۰
۵	۶۰	۷۸۶	۱۰۰	۱۰۰
۶	۶۰	۷۸۷	۱۰۰	۱۰۰
متوسط	۶۰	۷۸۴	۱۰۰	۱۰۰

شکل (۱۹) منحنی تغییرات تعداد وزنه های لایه مخفی شبکه را براساس تکرار ارایه الگو برای شبکه شماره ۱ جدول (۱۲) نشان میدهد.

واحدهای مخفی از قبل مشخص باشد نتایجی مشابه الگوریتم ۱ را تولید میکند. (۳) شبکه های تولید شده توسط الگوریتم ۲ دارای کارایی پایین تری نسبت به شبکه های تولید شده توسط الگوریتم ۱ هستند. (۴) در الگوریتم ۲ در صورتیکه تعداد واحدهای مخفی اولیه نسبتا زیاد باشد، ساختار تعیین شده برای شبکه دارای تعداد واحد های مخفی زیادی خواهد بود. نتیجه (۳) بدین علت است که الگوریتم ۱، آموزش را از یک شبکه بزرگ شروع می کند و سپس در حین آموزش واحدهای مخفی را کم یا زیاد می نماید و در نتیجه از مزایای در حین آموزش شبکه های بزرگ (پیچیدگی آموزش کم، دوری از حداقل های محلی) بیشترین بهره را می برد.

تذکره ۵: با توجه به جداول (۵) تا (۸) مشاهده می گردد که نسبت متوسط تعداد وزن ورودی بر واحد مخفی در قسمت (الف) این جداول بیشتر از قسمت (ب) آنها می باشد و این تفاوت با افزایش تعداد واحدهای ورودی شبکه (تعداد ویژگیهای الگوها) افزایش می یابد. یک دلیل این افزایش این است که شبکه حداکثر کوشش خود را در حفظ وزنه های موجود می نماید. در صورتیکه شبکه های بزرگ نسبت به حذف وزن مقاوم تر هستند و بهمین دلیل این نسبت کاهش می یابد.

تذکره ۶: در الگوریتم ۳ وزنها بصورت عمومی و با همه وزنه های روشن واحدهای روشن لایه مخفی مقایسه می گردند و یک وزن ممکن است بصورت اشتباه حذف گردد زیرا ممکن است یک وزن نسبت به وزنه های ورودی همان واحد مخفی دارای مقدار بزرگی و در نتیجه برای دسته بندی مورد نیاز باشد اما یک واحد مخفی دیگر دارای وزنه های نسبتا بزرگ باشد و یکی از وزنه های این واحدهای دسته بندی لازم نباشد. مقایسه این دو وزن کوچک (وزن مناسب) و بزرگ (وزن نامناسب)، باعث حذف وزن کوچکتر گردد. در نتیجه الگوریتم ۳ با احتیاط بیشتری وزنها را حذف میکند که باعث تولید شبکه های بزرگتری می گردد.

تذکره ۷: در این قسمت الگوریتم زمانبندی دیگری برای نحوه فعالیت اتوماتانهای یادگیر تعیین تعداد واحدها و

واحدهای مخفی افزایش و تعدادی از آنها کاهش می یابد (واحدهایی که واریانس فعالیت آنها افزایش میابد واحدهایی هستند که نقش موثر و مثبتی در عملکرد شبکه دارا هستند و واحدهایی که واریانس فعالیت آنها کاهش می یابد نقش موثری در عملکرد شبکه دارا نیستند) و بدین ترتیب قسمت هایی از شبکه که فعالیت بیشتری از خود نشان میدهند بتدریج کوچکتر میشود. همین پدیده در مورد وزنها نیز اتفاق می افتد. در ابتدا، مقادیر همه وزنها تقریبا یکسان هستند (زیرا در ابتدای آموزش وزنها بصورت تصادفی، از اعداد کوچکی انتخاب و مقدار دهی می شوند) که با پیشرفت آموزش، مقادیر تعدادی از وزنها افزایش و مقادیر تعدادی از آنها کاهش می یابد. نتایج آزمایشات نشان داده است که در حین آموزش، از تعداد وزنهایی که مقدار آنها بزرگ است کاسته می شود. پدیده فوق را میتوان به این صورت نیز بیان نمود که در ابتدای آموزش همه قسمت های شبکه فعال هستند و بتدریج و با پیشرفت آموزش بخش هایی از شبکه غیر فعال میگردند و در انتها بخش کوچکی از شبکه فعال باقی می ماند. در این بخش کوچک از شبکه، واریانس فعالیت واحدها بالا و وزنها دارای مقادیر بزرگ هستند. با بررسی های انجام شده روی مغز انسان، دانشمندان به این نتیجه رسیده اند که پدیده ای مشابه پدیده فوق در مغز انسان در زمان یادگیری نیز رخ میدهد (در بررسی هایی که روی مغز انسان به انجام رسیده است مشاهده شده است که الگوی فعالیت گسترده در اوایل فراگیری و سپس کاسته شدن از این فعالیت گسترده پس از عادی شدن فعالیت، هم در مورد فعالیت های فیزیکی نظیر تایپ کردن و رانندگی کردن و امثال آن صادق است و هم در خصوص فعالیت های نظری و فکری مانند فراگرفتن اطلاعات تازه و یا خواندن مطالب جدید (نظیر دروس تازه) به منظور به خاطر سپردن آنها [روزنامه اطلاعات، تاریخ دوشنبه ۲۸ تیر ۱۳۷۸ صفحه ۳ ضمیمه]..

تذکره ۴: با توجه به نتایج شبیه سازیهای انجام گرفته برای مسائل ذکر شده میتوان نتیجه گرفت که: (۱) از بین سه الگوریتم ارائه شده، الگوریتم ۱ شبکه های کوچکتری را تولید می نماید. (۲) الگوریتم ۲ در صورتیکه تعداد تقریبی

مقدار حساسیت برای وزن حذف شده نیاز به تعداد آرایه های مجموعه آموزشی از مرتبه $O(W^2)$ دارد که W تعداد وزنه های شبکه است. مقدار حساسیت تخمین زده شده بصورت زیر محاسبه میگردد.

$$\hat{S}_{ij} = - \sum_{n=0}^{N-1} \frac{\partial E}{\partial w_{ij}} \Delta w_{ij} \frac{w_{ij}^f}{w_{ij}^f - w_{ij}^0}$$

که N تعداد مراحل یک دوره آموزش، w_{ij}^f مقدار نهایی وزن w_{ij} و w_{ij}^0 مقدار اولیه وزن w_{ij} میباشد. تمام اطلاعات مورد نیاز برای محاسبه فوق در زمان آموزش فراهم می باشد. پس از یک دوره آموزش، برای هر وزن یک مقدار تخمینی از حساسیت محاسبه میشود و وزنی که دارای کمترین حساسیت هست حذف میگردد. قابل ذکر است اگر تمام وزنه های خروجی یک واحد حذف گردند آن واحد حذف میگردد. اگر همه وزنه های ورودی یک واحد حذف شوند خروجی این واحد دارای مقدار ثابتی میشود و میتوان این واحد را حذف نمود و اثر خروجی آنرا در مقدار آستانه واحد های لایه بعد اعمال نمود (شکل ۷). لازم بذکر است که در تمامی آزمایشهای انجام گرفته با استفاده از الگوریتم کارنین ابتدا شبکه بطور کامل آموزش داده شده است بطوریکه قادر به شناسایی کامل مجموعه آموزشی باشد و سپس در مرحله هرس، پس از حذف یک وزن شبکه باز آموزی گردیده است. در مرحله هرس وزنها، ۵۰۰۰۰ بار مجموعه آموزشی به شبکه آرایه شده است و در انتها شبکه ای را که بهترین ساختار را دارد انتخاب و در جداول (۱۳) تا (۱۶) گزارش شده است.

جدول ۱۳: نتیجه شبیه سازی الگوریتم کارنین برای مثال ۱

شبکه	واحد مخفی	تعداد وزنه های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۲۸	۸۳	۱۰۰	۱۰۰
۲	۲۷	۷۹	۱۰۰	۱۰۰
۳	۲۹	۸۶	۱۰۰	۱۰۰
۴	۳۱	۹۱	۱۰۰	۱۰۰
۵	۳۳	۹۷	۱۰۰	۱۰۰
۶	۳۲	۹۴	۱۰۰	۱۰۰
متوسط	۳۰	۸۸/۳۳	۱۰۰	۱۰۰

وزنه های لایه مخفی شبکه عصبی سه لایه بیان و نتایج آن برای یک مسئله نمونه آرایه میگردد. در این الگوریتم، در حین آموزش، هر دو اتوماتان تعیین تعداد واحدها و وزنه های لایه مخفی روشن هستند و بطور همزمان سعی در پیدا نمودن ساختار مناسب شبکه عصبی را دارند. لازم بذکر است که اتوماتان تعیین تعداد وزنه های لایه مخفی در هر مرحله فقط وزنه های متصل به واحدهای مخفی روشن را بررسی می کند و با تغییر وضعیت آنها جهت یافتن ساختار مناسب حرکت می کند. این الگوریتم بهتر از الگوریتم های ۲ و ۳ و کمی بدتر از الگوریتم ۱ عمل میکند و تعداد واحدهای مخفی در شبکه های تولید شده بسیار کمتر از تعداد واحدهای مخفی تعیین شده در الگوریتم های ۲ و ۳ و مقداری بیشتر از تعداد واحدهای مخفی تعیین شده توسط الگوریتم ۱ میباشد اما نسبت متوسط تعداد وزن بر واحد مخفی در این الگوریتم بیشتر از الگوریتم های پیشنهاد شده دیگر است.

مقایسه الگوریتم های پیشنهادی با الگوریتم هرس کارنین^{۱۷}

در این قسمت الگوریتم های پیشنهادی با الگوریتم هرس که توسط کارنین پیشنهاد گردیده است [۴۵] مقایسه میگردد. الگوریتم کارنین به شرح زیر میباشد. در این الگوریتم، ابتدا شبکه بطور کامل آموزش داده میشود و در پایان آموزش، حساسیت هر وزن (تاثیر منفی حذف یک وزن در تابع هدف شبکه) محاسبه میگردد و وزنهایی که دارای حساسیت کمتری هستند حذف میگرددند. در این الگوریتم مقدار حساسیت یک وزن (S_{ij}) بصورت زیر محاسبه میگردد.

$$S_{ij} = - \frac{E(w^f) - E(0)}{w^f - 0} w^f$$

که w^f مقدار نهایی وزن در پایان دوره آموزش، 0 مقدار وزن پس از حذف وزن w^f ، $E(w^f)$ مقدار خطای شبکه در هنگام موجود بودن وزن w^f و $E(0)$ مقدار خطا پس از حذف وزن w^f میباشد. برای محاسبه حساسیت یک وزن میتوان آن وزن را حذف نموده و مقدار خطا و در نتیجه مقدار حساسیت برای وزن حذف شده را محاسبه نمود. با توجه به اینکه محاسبه

کارنین، هرس گردیده اند. پس از حذف یک وزن در شبکه، ۵۰۰ بار مجموعه الگوهای آموزش به شبکه ارایه و شبکه بازآموزی گردیده است و نتایج آن در جدول (۱۵) آورده شده است. لازم بذکر است که در آزمایشهای جدول (۱۴) مقدار نرخ یادگیری برابر ۰/۱ و مقدار ضریب گشتاور برابر صفر در نظر گرفته شده است.

با توجه به جداول (۲-ب)، (۶-ب)، (۱۰) و (۱۴) مشاهده میشود که تعداد وزنهاى لایه مخفی تقریباً ۲۰ برابر تعداد وزنهاى تعیین شده توسط الگوریتم ۱، ۲ برابر تعداد وزنهاى تعیین شده توسط الگوریتم ۲ و بیشتر از ۲ برابر تعداد وزنهاى تعیین شده توسط الگوریتم ۳ می باشد. همچنین در شبکه‌های تولید شده توسط الگوریتم هرس کارنین تقریباً همه واحدها دارای ۳ وزن ورودی هستند. مشاهده شده است که برای مسائل ۱ و ۲ در بیشتر موارد الگوریتم کارنین، وزنها را بطریقی هرس می نماید که وزنهاى هرس شده متوالیا متعلق به یک واحد مخفی می باشند و تا حذف کامل این واحد ادامه پیدا می کند (جداول ۱۳ و ۱۴) و بهمین دلیل این الگوریتم در بیشتر موارد برای مسائل ۱ و ۲ شبکه ای با اتصالات محلی را تولید نمی نماید.

جدول ۱۵: نتیجه شبیه سازی الگوریتم کارنین برای مثال ۳.

شبکه	واحد مخفی	تعداد وزنهاى لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۵۸	۲۷۵	۹۸/۱۲۵	۹۰
۲	۵۸	۱۳۰	۹۶/۶۲۵	۱۰۰
۳	۵۹	۲۴۰	۹۶/۸۷۵	۹۰
۴	۵۶	۱۵۴	۹۶/۶۲۵	۱۰۰
۵	۶۰	۲۱۶	۹۸/۱۲۵	۱۰۰
۶	۶۰	۱۸۰	۹۹/۳۷۵	۱۰۰
متوسط	۵۸/۵	۱۹۹/۱۶	۹۷/۶۲۵	۹۶/۶۶

مسئله تشخیص اعداد چاپی فارسی: برای پیاده سازی این مسئله ۶ شبکه با ۷ واحد ورودی، ۶۰ واحد مخفی و ۱۰ واحد خروجی در نظر گرفته شده است. این شبکه ها ابتدا بطور کامل آموزش داده شده و سپس توسط الگوریتم کارنین هرس گردیده اند. پس از حذف یک وزن در شبکه،

مسئله دسته بندی چهار دسته از الگوها: برای پیاده سازی این مسئله ۶ شبکه با ۲ واحد ورودی، ۳۹ واحد مخفی و ۴ واحد خروجی در نظر گرفته شده است. این شبکه ها ابتدا بطور کامل آموزش داده شده و سپس توسط الگوریتم کارنین، هرس گردیده اند. پس از حذف وزن با کمترین حساسیت، ۱۰۰ بار مجموعه الگوهای آموزش به شبکه ارایه و شبکه بازآموزی گردیده است و نتایج آن در جدول (۱۳) آورده شده است. برای آزمایشهای انجام شده مقدار نرخ یادگیری برابر ۰/۱ و مقدار ضریب گشتاور برابر ۰/۶ می باشد. با توجه به جداول (۱-ب)، (۵-ب)، (۹) و (۱۳) مشاهده میشود که تعداد وزنهاى تعیین شده توسط الگوریتم کارنین برای لایه مخفی تقریباً ۱۰ برابر تعداد وزنهاى تعیین شده توسط الگوریتم ۱، ۲ برابر تعداد وزنهاى تعیین شده توسط الگوریتم ۲ و ۳ برابر تعداد وزنهاى تعیین شده توسط الگوریتم ۳ می باشد که این خود تاییدی بر برتری الگوریتم های پیشنهاد شده می باشد. لازم بذکر است که در شبکه‌های تولید شده توسط الگوریتم هرس کارنین تقریباً همه واحدها دارای ۳ وزن ورودی هستند این بدان معناست که این الگوریتم در این مورد شبکه ای با اتصالات محلی را تولید نمی کند.

جدول ۱۴: نتیجه شبیه سازی الگوریتم کارنین برای مثال ۲.

شبکه	واحد مخفی	تعداد وزنهاى لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۵۳	۱۵۸	۱۰۰	۱۰۰
۲	۵۱	۱۵۲	۱۰۰	۱۰۰
۳	۵۳	۱۵۷	۱۰۰	۱۰۰
۴	۴۶	۱۳۷	۱۰۰	۱۰۰
۵	۵۰	۱۵۰	۱۰۰	۱۰۰
۶	۴۹	۱۴۷	۱۰۰	۱۰۰
متوسط	۵۰/۳	۱۵۰/۱	۱۰۰	۱۰۰

مسئله دسته بندی دو دسته از الگوها: برای پیاده سازی این مسئله ۶ شبکه با ۲ واحد ورودی، ۶۰ واحد مخفی و ۲ واحد خروجی در نظر گرفته شده است. این شبکه ها ابتدا بطور کامل آموزش داده شده و سپس توسط الگوریتم

کارنن تعداد کمی از وزنه‌های لایه مخفی را حذف نموده است. در هنگام شبیه‌سازیها مشاهده گردید که با حذف اولین واحد مخفی، کارایی شبکه بشدت کاهش یافته و شبکه بهیچوجه قادر به بازآموزی نمی‌گردد. همچنین در بسیاری از شبیه‌سازیها مشاهده شده که بدون اینکه واحد مخفی حذف گردد کاهش زیادی در کارایی شبکه رخ میدهد. شبکه‌های ارایه شده در جدول (۱۶) نماینده شبکه‌های تولیدشده توسط الگوریتم کارنن قبل از شروع کاهش شدید کارایی میباشد. مثال فوق، عدم کارایی الگوریتم هرس کارنن را برای این مسئله نشان میدهد. الگوریتم‌های پیشنهادی روی مسائل متنوعی آزمایش شدند که نتایج کامل آنها در مرجع [۴۶] آمده است.

نتیجه گیری

در این مقاله سه الگوریتم براساس اتوماتانهای یادگیر مهاجرت اشیا برای تعیین ساختار شبکه‌های عصبی سه‌لایه ارایه گردیده است. این الگوریتم‌ها تعداد واحدها و تعداد وزنه‌های لایه مخفی یک شبکه عصبی سه‌لایه را تعیین میکنند. الگوریتم‌های پیشنهادی از اتوماتانهای مهاجرت اشیا و الگوریتم یادگیری انتشار خطا به عقب برای تعیین ساختار شبکه استفاده می‌کنند. الگوریتم‌های پیشنهادی، شبکه‌هایی با اتصالات محلی که دارای قدرت تعمیم بالایی هستند تولید می‌کنند. کارایی الگوریتم‌های پیشنهادی از طریق شبیه‌سازی روی مسائل مختلف نشان داده شده است. اتوماتانهای یادگیر بعنوان یک ابزار جستجوی عمومی، زمانی کارایی بالایی از خود نشان میدهد که فضایی که آنها جستجو میکنند بزرگ باشد. بهمین دلیل الگوریتم‌های پیشنهاد شده در این مقاله که مبتنی بر اتوماتانهای یادگیر طراحی شده‌اند زمانی دارای کارایی بالایی هستند که فضای جستجوی آنها (تعداد واحدهای لایه ورودی شبکه) بزرگ باشد. الگوریتم‌های پیشنهادی با الگوریتم هرس کارنن مقایسه گردیده است و برتری الگوریتم‌های پیشنهادی نسبت به این الگوریتم از طریق شبیه‌سازی نشان داده شده است.

۱۰۰ بار مجموعه الگوهای آموزش به شبکه ارایه و شبکه بازآموزی گردیده است و نتایج آن در جدول (۱۵) آورده شده است. برای آزمایشهای انجام شده مقدار نرخ یادگیری برابر ۰/۰۵ و مقدار ضریب گشتاور برابر صفر می‌باشد. با توجه به جداول (۳-ب)، (۷-ب)، (۱۱) و (۱۵) مشاهده میشود که تعداد وزنه‌های تعیین شده توسط الگوریتم کارنن برای لایه مخفی تقریباً ۱۰ برابر تعداد وزنه‌های تعیین شده توسط الگوریتم ۱، بیشتر از ۲ برابر تعداد وزنه‌های تعیین شده توسط الگوریتم ۲ و مقداری بیشتر از تعداد وزنه‌های تعیین شده توسط الگوریتم ۳ می‌باشد. نکته جالب توجه این است که در این مورد شبکه‌های تولید شده توسط الگوریتم هرس کارنن برای مسئله تشخیص اعداد چاپی فارسی و برخلاف مسائل قبل، شبکه‌هایی با اتصالات محلی هستند.

مسئله تشخیص اعداد لاتین: برای پیاده سازی این مسئله ۶ شبکه با ۶۴ واحد ورودی ۶۰ واحد مخفی و ۱۰ واحد در نظر گرفته شده است. این شبکه‌ها ابتدا بطور کامل آموزش داده شده و سپس توسط الگوریتم کارنن هرس گردیده اند. پس از حذف یک وزن در شبکه، ۵۰۰ بار مجموعه الگوهای آموزش به شبکه ارایه و شبکه بازآموزی گردیده است و نتایج آن در جدول (۱۶) آورده شده است. برای آزمایشهای جدول (۱۶) مقدار نرخ یادگیری برابر ۰/۰۱ و مقدار ضریب گشتاور برابر صفر می‌باشد.

جدول ۱۶: نتیجه شبیه سازی الگوریتم کارنن برای مثال ۴.

شبکه	واحد مخفی	تعداد وزنه‌های لایه مخفی	درصد تشخیص (آموزش)	درصد تشخیص (آزمایشی)
۱	۶۰	۳۸۴۰	۱۰۰	۱۰۰
۲	۶۰	۳۸۴۲	۱۰۰	۱۰۰
۳	۶۰	۳۸۴۰	۱۰۰	۱۰۰
۴	۶۰	۳۸۳۹	۱۰۰	۱۰۰
۵	۶۰	۳۸۵۰	۱۰۰	۱۰۰
۶	۶۰	۳۸۴۸	۱۰۰	۱۰۰
متوسط	۶۰	۳۸۴۳/۱	۱۰۰	۱۰۰

با توجه به جدول (۱۶) مشاهده میشود که الگوریتم

مراجع

- 1 - Lin, J. H. and Vitter, J. S. (1991). "Complexity results on learning by neural nets." *Machine Learning*, Vol. 6, PP. 211-230.
- 2 - Arai, M. (1993). "Bounds on the number of hidden units in binary-valued three-layer neural networks." *Neural Networks*, Vol. 6, PP. 855-860.
- 3 - Huang, S. C. and Huang, Y. F. (1991). "Bounds on the number hidden neurons in multilayer perceptrons." *IEEE Trans. on Neural Networks*, Vol. 2, No. 1, PP. 47-56.
- 4 - Tamura, S. and Tateishi, M. (1997). "Capabilities of a four-layered feedforward neural network: four layers versus three." *IEEE Trans. on Neural Networks*, Vol. 8, No. 2, PP. 251-255.
- 5 - Castellano, G., Fanelli, A. M., and Pelillo, M. (1997). "A Iterative pruning algorithm for feedforward neural networks." *IEEE Trans. on Neural Networks*, Vol. 8, No. 3, PP. 519-531.
- 6 - Kruschke, J. H. (1988). "Creating local and distributed bottlenecks in hidden layer of backpropagation networks." *Proc. of Connectionist Models, Summer School*, Eds. D. Tourestzky, G. Hinton, and T. Sejnowski, PP. 120-126.
- 7 - Kruschke, J. H. (1989). "Improving generalization in backpropagation networks." *Proc. of Int. Joint Conf. on Neural Networks*, Vol. I, PP. 443-447.
- 8 - Reed, R. (1993). Pruning Algorithms - "A survey." *IEEE Trans. on Neural Networks*, Vol. 4, No. 5, PP. 740-747.
- 9 - Sietsma, J. and Dow, R. J. F. (1991). "Creating artificial neural networks that generalize." *Neural Networks*, Vol. 4, No. 1, PP. 67-79.
- 10 - Beigy, H. and Meybodi, M. R. (1998). "A fast method for determining the number of hidden units in feedforward neural networks." *Proc. of CSIC-97*, Tehran, Iran, PP. 414-420 (In Persian).
- 11 - Fahlman, S. E. and LeBrier, C. (1990). "The cascade-correlation learning architecture." *Advances in Neural Information Processing System*, Vol. II, PP. 524-532.
- 12 - Fren, M. (1990). "The upstart: a method for constructing and training feedforward neural networks." *Neural Computation*, PP. 198-209.
- 13 - Kwok, T. Y. and Yeung, D. Y. (1997). "Constructive algorithms for structure learning in feedforward neural networks for regression problems." *IEEE Trans. on Neural Networks*, Vol. 8, No. 3 PP. 630-645.
- 14 - Marchand, M., Golea, M. and Rujan, R. (1990). "A convergence theorem for sequential learning in two-layer perceptrons." *Europhysics Letters 11*, PP. 487-492.
- 15 - Meltser, M., Shoham, M. and Manevitz, L. M. (1996). "Approximating function by neural networks: a constructive solution in the uniform norm." *Neural Networks*, Vol 9, No. 6, PP. 965-978.
- 16 - Mezard, M. and Nadal, J. P. (1989). "Learning in feedforward neural networks: the tiling algorithm." *Journal of Physics*, PP. 1285-1296.
- 17 - Sirat, J. A. and Nadal, J. P. (1990). "Neural trees: a new tool for classification, preprint." *Laboratories d'Electronique, Philips, Limeil Brevannes, France*.
- 18 - Yeung, D. Y. (1991). "Automatic determination of network size for supervised learning." *IEEE Int. Joint Conf. on Neural Networks*, PP. 158-164.
- 19 - Hirose, Y., Yamashita, K. and Hijya, S. (1991). "Back-propagation algorithm which varies the number of hidden units." *Neural Networks*, Vol. 4, No. 1, PP. 61-66.

- 20 - Nabhan, T. M. and Zomaya, A. Y. (1994). "Toward neural networks structures for function approximation." *Neural Networks*, Vol. 7, No. 1, PP. 89-99.
 - 21 - Angeline, P. J., Saunders, G. M. and Pollack, J. B. (1994). "Evolutionary algorithm that construct recurrent neural networks." *IEEE Trans. on Neural Networks*, Vol. 5, No. 1, PP. 54-65.
 - 22 - Maniezzo, V. (1994). "Genetic evolution of the topology and weight distribution of neural networks." *IEEE Trans. on Neural Networks*, Vol. 5, No. 1, PP. 39-53.
 - 23 - Schaffer, J. D., Whitley, D. and Eshelman, L. J. (1992). "Combinations of genetic algorithms and neural networks: A Survey of the state of the art." *IEEE Proc.*, COGANN-92, PP. 1-37.
 - 24 - Whitley, D. and Bogart, C. (1990). "The evolution of connectivity: pruning neural networks using genetic algorithms." *Proc. of Int. Joint Conf. on Neural Networks*, Vol. I, PP. 134.
 - 25 - Yao, X. and Liu, Y. (1997). "A new evolutionary system for evolving artificial neural networks." *IEEE Trans. on Neural Networks*, Vol. 8, No. 3, PP. 694-713.
 - 26 - Beigy, H. and Meybodi, M. R. (1999). "Optimization of topology of neural networks using learning automata." *Proc. of 4th Annual Int. Computer Society of Iran Computer Conf. CSICC-98*, Tehran, Iran, PP. 417-428 (In Persian).
 - 27 - Beigy, H. and Meybodi, M. R. (1999). "A learning automata based algorithm for determination of optimal number of hidden units in three layers feedforward neural networks, accepted for publication." *Journal of Amirkabir, Tehran, Iran* (In Persian).
 - 28 - Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). "Learning internal representations by error backpropagation." *In Parallel distributed processing*, Cambridge, MA: MIT Press.
 - 29 - Beigy, H. and Meybodi, M. R. (1998). "Optimization of topology of neural networks: a survey, technical reports." *Computer Eng.*, Dept. Amirkabir University of Technology, Tehran, Iran (In Persian).
 - 30 - Lippman, R. P. (1987). "An introduction to computing with neural nets." *IEEE ASSP Mag.*, Vol. 4, PP. 4-22.
 - 31 - Judd, J. S. (1990). *Neural network design and the complexity of learning*. Cambridge, MA: MIT Press.
 - 32 - Yu, X. H. (1992). "Can backpropagation error surface not have local minima." *IEEE Trans. on Neural Networks*, Vol. 3, No. 6, PP. 1019-1021.
 - 33 - Narendra, K. S. and Thathachar, M. A. L. (1989). *Learning automata : an introduction*. Prentice-hall, Englewood cliffs.
 - 34 - Meybodi, M. R. and Lakshmirvarhan, S. (1982). "Optimality of a general class of learning algorithm." *Information Science*, Vol. 28, PP. 1-20.
 - 35 - Meybodi, M. R. and Lakshmirvarhan, S. (1984). *On a class of learning algorithms which have a symmetric behavior under success and failure*. Springer Verlag Lecture Notes in Statistics, PP. 145-155.
 - 36 - Lakshmirvarhan, S. (1981). *Learning algorithms: theory and applications*. New York, Springer-verlag.
 - 37 - Oommen, B. J. and Ma, D. C. Y. (1988). "Deterministic learning automata solutions to the equipartitioning problem." *IEEE Trans. on Computers*, No. 37, No. 1, PP. 2-13.
 - 38 - Oommen, B. J., Valiveti, R. S. and Zgierski, J. R. (1991). "An adaptive learning solution to the keyboard optimization problem." *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 21, No. 6, PP. 1608-1618.
 - 39 - Oommen, B. J. and Croix, E. V. de St. (1996). "Graph partitioning using learning automata." *IEEE Trans. on Computers*, No. 45, No. 2, PP. 195-208.
-

- 40 - Beigy, H. and Meybodi, M. R. (2000). "Graph isomorphism using learning automata." *Proc. of 4th Annual Int. Computer Society of Iran Computer Conf. CSICC'2000*, Tehran, Iran, PP. 402-415 (In Persian).
- 41 - Hush, D. R. and Horne, B. G. (1993). "Progress in supervised neural networks." *IEEE Signal Processing Mag.*, Jan. 1993, PP. 8-39.
- 42 - Schalkoff, R. J. (1992). *Pattern recognition: statistical, structural and neural approaches*. John Wiley, New York.
- 43- Dastpak, V. (1992). *Automatic recognition of farsi printed letters*. Ms. Thesis, Computer Eng. Dept. Amirkabir University of Technology, Tehran, Iran (In Persian).
- 44 - Sperduti, A. and Starita, A. (1993). "Speed up learning and network optimization with extended backpropagation." *Neural Networks*, Vol. 6, PP. 365-383.
- 45 - Karnin, E. D. (1990). "A simple procedure for pruning backpropagation trained neural networks." *IEEE Trans. on Neural Networks*, Vol. 1, No. 2, PP. 239-242.
- 46 - Meybodi, M. R. and Beigy, H. (1999). "Neural network engineering using learning automata: determination of desired size for three layer feedforward neural network." *Technical Reports, Computer Eng. Dept. Amirkabir University of Technology, Tehran, Iran* (In Persian).

واژه های انگلیسی به ترتیب استفاده در متن

- 1 - Pruning Algorithms
- 2 - Constructive Algorithms
- 3 - Evaluonary Algorithms
- 4 - Learning Automata
- 5 - Object Migrating Automata
- 6 - Action
- 7 - Fixed Structure Learning Automata (FSLA)
- 8 - Variable Structure Learning Automata (VSLA)
- 9 - Unfavorable
- 10 - Favorable
- 11 - Ommen
- 12 - Ma
- 13 - Graph Partitioning
- 14 - Hidden Unit Learning Automata (HULA)
- 15 - Hidden Weight Learning Automata ((HWLA)
- 16 - Timming Algorithms
- 17- Karnin