

## تنظیم پارامترهای الگوریتم‌های یادگیری در شبکه‌های عصبی SDPTA، SCPTA، RDPTA، EBPTA با استفاده از آتاماتون‌های یادگیر و مقایسه آن با روش سعی و خطا

کامبیز بدیع  
مرکز تحقیقات مخابرات  
ایران

محمد تشنه لب  
دانشگاه صنعتی  
خواجه نصیرالدین طوسی

محمدرضا میبدی  
دانشگاه صنعتی  
امیرکبیر

علی یوسفی  
دانشگاه آزاد اسلامی  
واحد همدان

### چکیده:

یکی از ابزارهای هوش مصنوعی شبکه‌های عصبی مصنوعی است که امروزه مورد توجه بسیاری از محققان قرار گرفته است. مهمترین ویژگی یک شبکه عصبی، الگوریتم یادگیری آن می‌باشد. پارامترهای مختلفی در الگوریتم یادگیری شبکه عصبی مصنوعی می‌تواند وجود داشته باشد که تنظیم صحیح و دقیق آن می‌تواند به کارآمدی الگوریتم در روند کاربردی شبکه عصبی منجر شود. اعم این پارامترها بترتیب در شبکه‌های SDPTA، RDPTA، SCPTA، EBPTA عبارتند از c و  $\eta$  که در این تحقیق به تنظیم آن به روش آتاماتون و مقایسه آن با تنظیم به روش سعی و خطا، پرداخته شده است [1]. نتایج بدست آمده مؤید کارآمدتر بودن تنظیم پارامترهای فوق در شبکه‌های عصبی مذکور به روش آتاماتون‌های یادگیر نسبت به روش سعی و خطا می‌باشد.

**کلید واژگان:** پارامترهای شبکه‌های عصبی، آتاماتون‌های یادگیر، شبکه‌های عصبی SDPTA، RDPTA، SCPTA، EBPTA

### ۱- مقدمه

یکی از روش‌های بهینه‌سازی در مسائل مختلف مهندسی نظیر شناسایی، طبقه‌بندی و تشخیص الگو روش هوش مصنوعی است. شبکه‌های عصبی به عنوان یک روش هوشمند می‌تواند در فرآیند یادگیری یک سیستم با محیط پیرامون آن انعطاف پذیری و قابلیت تعمیم بالایی داشته باشد، مدل شبکه عصبی مصنوعی که الگوبرداری از سیستم عصبی انسان می‌باشد، به حل مسائلی که در آن پروسه یادگیری نقش اساسی در تعیین پارامترها جهت تخمین مسئله ورودی دارد می‌پردازد، یادگیری بسته به مسئله ورودی و خصوصیات محیط به دو صورت با سرپرست و بدون سرپرست می‌باشد، معماری و

- 1 - Statistical Discrete Pattern Training Algorithm
- 2 - R-Class Discrete Pattern Training Algorithm
- 3 - Statistical Category Pattern Training Algorithm
- 4 - Statistical Category Pattern Training Algorithm

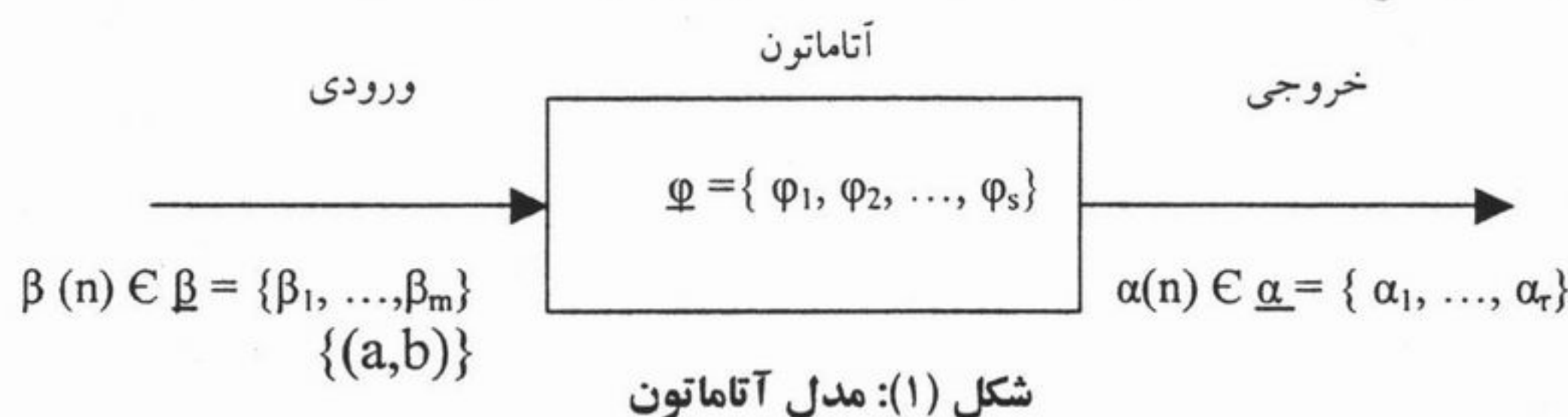
الگوریتم های مختلف آموزشی در شبکه های عصبی وجود دارد که اساساً در بیشتر این الگوریتم ها هدف بدست آوردن وزن های بین نرون ها بطور بهینه است. در تعیین وزنه های الگوریتم شبکه های عصبی فوق، تنظیم پارامتر های این شبکه ها بطور بهینه نقش اساسی دارد [2, 3, 4, 5].

## ۲- بهینه سازی

بهینه سازی کمینه یا بیشینه نمودن یک تابع هدف می باشد که معمولاً با توجه به قیودی که مشخص کننده شرایط حاکم بر مسئله می باشد انجام می پذیرد. برخلاف روشهای سنتی بهینه سازی، روش بهینه سازی هوشمند دارای ویژگی هایی در فرآیند بهینه نمودن یک مسئله مانند تطبیقی بودن مسیر جستجو، نامعین بودن قواعد و عدم نیاز به مدل سازی دقیق سیستم می باشد. در این میان شبکه های عصبی مصنوعی به عنوان یک روش بهینه سازی هوشمند و با توجه به ماهیت الگوریتم یادگیری آن دارای این ویژگی است که تا حد قابل قبولی در زمان محاسبات و عدم همگرایی به بهینه مطلق (عمومی) می تواند کارآمد باشد. بنابراین از آنجائیکه فرآیند مطلوب واقع شدن یک الگوریتم به بهینه سازی پارامترهای آن بستگی دارد. در این مقاله با روش آتاماتون به تنظیم پارامترهای فوق در شبکه های عصبی مزبور پرداخته می شود [6, 7].

## ۳- آتاماتون

یک آتاماتون، یک سیستم مجرد است که بصورت پنج تایی  $\{\phi, \underline{\alpha}, \underline{\beta}, f(0,0), H(0,0)\}$  تعریف می شود که در آن  $\phi$  مجموعه حالات داخلی،  $\underline{\beta}$  مجموعه ورودی،  $\underline{\alpha}$  مجموعه خروجی،  $F(0,0): \phi \times \underline{\beta} \rightarrow \phi$  تابع تغییر حالت بر حسب حالت و ورودی فعلی و  $H(0,0): \phi \times \underline{\beta} \rightarrow \underline{\alpha}$  تابع خروجی آتاماتون بر حسب حالت و ورودی فعلی می باشد، در صورتیکه خروجی آتاماتون تنها به حالت فعلی آن بستگی داشته باشد. با آتاماتون حالت خروجی روبرو هستیم. در این حالت تابع  $H(0,0)$  به تابع  $G(0): \phi \times \underline{\beta} \rightarrow \underline{\alpha}$  تبدیل می شود. شکل (۱) یک آتاماتون را نشان می دهد [9].



بطور رسمی مشخصات آتاماتون به صورت زیر تعریف می شود.

(۱) حالت آتاماتون در لحظه  $n$ :  $\phi(n) \in \underline{\phi} = \{\phi_1, \phi_2, \dots, \phi_s\}$

(۲) خروجی یا عمل آتاماتون در لحظه  $n$ :  $\alpha(n) \in \underline{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$

(۳) ورودی آتاماتون در لحظه  $n$  که می تواند عضوی از مجموعه محدود یا نامحدود ورودی ها باشد (رابطه ۱).

رابطه (۱):  $\beta(n) \in \underline{\beta} = \{\beta_1, \dots, \beta_m\}$  یا  $\beta(n) \in \underline{\beta} = \{(a,b)\}, a, b \in R$

(۴) تابع تبدیل حالت آتاماتون  $F(0,0)$  که حالت آتاماتون در لحظه  $n+1$  را با توجه به حالت ورودی در لحظه  $n$  تعیین

می کند (رابطه ۲)

$$\phi(n+1) = F[\phi(n), \beta(n)]$$

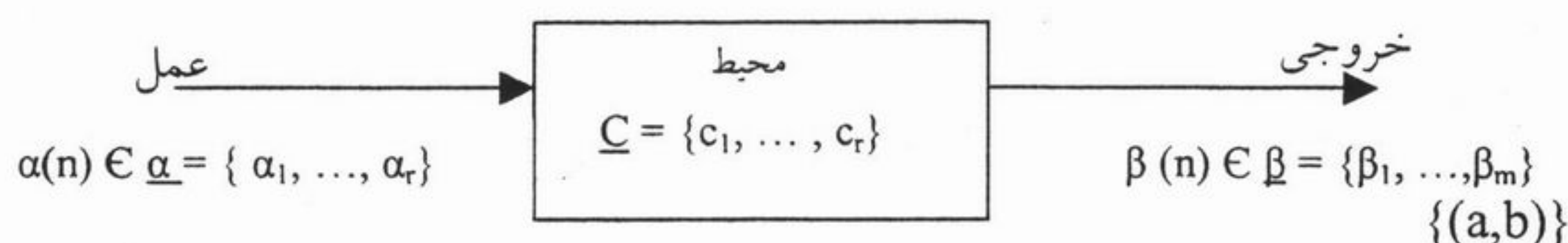
رابطه (۲):

۵) تابع خروجی  $G(0)$  آتاماتون که خروجی در لحظه  $n$  را بر حسب حالت آن در لحظه  $n$  تعیین می کند (رابطه ۳).

$$\alpha(n) = G[\phi(n)]$$

رابطه (۳):

تعاریف فوق برای آتاماتون حالت خروجی بیان شده اند، می توان نشان داد که برای هر آتاماتون با تابع خروجی  $H(0,0)$  می توان آتاماتون معادل دیگری با تابع خروجی  $G(0)$  پیدا کرد، عملکرد آتاماتون در زمانهای  $n = 0, 1, 2, \dots$  بصورت زیر می باشد. با داشتن حالت اولیه  $\phi(0)$ ، عمل  $\alpha(0)$  با توجه به تابع  $G(0)$  انتخاب و به محیط اعلام می گردد. با دریافت پاسخ  $\beta(0)$  و با استفاده از تابع  $F(0,0)$  حالت آتاماتون در لحظه  $\phi(1), n=1$  تعیین می گردد و این چرخه ادامه پیدا می کند. مشاهده می شود که حالت و عمل آتاماتون در لحظه  $n$ ، فقط به ورودی و حالت آتاماتون در لحظه  $n-1$  بستگی دارد. در این حالت یک ترتیب متناظر بین آتاماتون های یادگرو زنجیره های مارکوفی<sup>۵</sup> بوجود می آید که در تحلیل نحوه عملکرد آتاماتون می تواند مفید باشد. در صورتیکه هر دو تابع  $F$  و  $G$  قطعی<sup>۶</sup> باشد، آتاماتون قطعی است در این صورت با داشتن حالت ورودی فعلی، خروجی فعلی و حالت بعدی آتاماتون قطعاً قابل پیش بینی خواهد بود. اگر  $F$  یا  $G$  غیر قطعی<sup>۷</sup> باشند، آتاماتون غیر قطعی خواهد بود، در این صورت با داشتن حالت و ورودی فعلی آتاماتون، خروجی و حالت بعدی آتاماتون را قطعاً نمی توان تایید کرد، بلکه فقط احتمال رخداد هر یک را می توان تعیین نمود. مدل های مختلفی برای آتاماتون وجود دارد که در این مقاله فقط به نحوه عملکرد هر یک در تقابل با محیط به صورت شکل (۲) اشاره می شود.



شکل (۲): مدل محیط تصادفی

#### ۴- شبکه عصبی با الگوریتم SDPTA

در این شبکه پارامتر  $C$ ، عددی است در بازه  $[0,1]$  به عنوان پارامتر تنظیم کننده وزن های شبکه در الگوریتم یادگیری بکار می رود. کاربرد این شبکه بیشتر در طبقه بندی الگوهای بکار می رود که تابع تحریک آن در شبکه عصبی، تابع گسسته  $\text{sgn}$  می باشد.

ویژگی این الگوریتم و عملکرد آن در تنظیم پارامتر  $C$  برای محیط شبکه SDPTA در شبه کدهای شکل های (۳) و (۴) به دو روش سعی و خطا و آتاماتون آمده است [1].

5 - Markov - Chain

6 -Deterministic

7 - Non - Deterministic

Procedure Adaptparameter (c) – SDPTA (random)

//Given are P training pairs  $\{x_1, d_1, x_2, d_2, \dots, x_p, d_p\}$ ,

Where  $x_i$  is  $(n \times 1)$ ,  $d_i$  is  $(R \times 1)$ ,  $i=1, 2, \dots, P$ . Note that augmented input vectors are used:

$y_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix}$ , for  $i=1, 2, \dots, P$  and K denotes the training step; p denotes the step counter within the training cycle

Step1: Initialize parameter(C) // C is positive random value between(0,1]

Step2: Initialize the weights to small random values

Initialize  $p \leftarrow 1$ ,  $E \leftarrow 0$ ,  $k \leftarrow 1$ ,  $N \leftarrow \text{MaxIteration}$

Step3: Call (Training cycle) //  $y \leftarrow y_p$ ,  $d \leftarrow d_p$ ,  $O \leftarrow \text{sgn}(w^T y)$

Step4: Call (Weight update) //  $W \leftarrow W + \frac{1}{2} (c) (d-o) y$

Step5: Call (Error Cycle) //  $E \leftarrow E + \frac{1}{2} (d-o)^2$

Step6: If  $p < P$  Then  $p \leftarrow p+1$ ,  $k \leftarrow k+1$ , Goto Step3  
End If

Step7: If  $E > 0$  Then  $E \leftarrow 0$ ,  $p \leftarrow 1$ , Goto step3  
End If

End.

شکل (۳): شبه کد تنظیم پارامتر c در شبکه عصبی SDPTA به روش سعی و خطا

Procedure Adaptparameter (c) – SDPTA (Automata)

//Given are P training pairs  $\{x_1, d_1, x_2, d_2, \dots, x_p, d_p\}$ ,

Where  $x_i$  is  $(n \times 1)$ ,  $d_i$  is  $(R \times 1)$ ,  $i=1, 2, \dots, P$ . Note that augmented input vectors are used:

$y_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix}$ , for  $i=1, 2, \dots, P$

and K denotes the training step; p denotes the step counter within the training cycle

Note:// Select Action(i) from Automata with CDF random generation if Automata is Nondeterministic.

Initialize the weights to small random values

Initialize the parameter (c) for automation Automata

Initialize  $p \leftarrow 1$ ,  $E \leftarrow 0$ ,  $k \leftarrow 1$ ,  $N \leftarrow \text{MaxIteration}$

Repeat

For all Training Patterns (x, d) in the training set Do

Call (Training cycle) //  $y \leftarrow y_p$ ,  $d \leftarrow d_p$ ,  $O \leftarrow \text{sgn}(w^T y)$

Call (Weight update) //  $W \leftarrow W + \frac{1}{2} \text{Parameter}(c) (d-o) y$

Call (Error Cycle) //  $E \leftarrow E + \frac{1}{2} (d-o)^2$

If  $E = 0$  then  $C = \text{Call Automata}(0)$

Else  $C = \text{Call Automata}(1)$

End If

End For

Inc (k)

Until  $k > N$

End.

شکل (۴): شبه کد تنظیم پارامتر c در شبکه عصبی SDPTA به روش آتاماتون

#### ۴-۱- نتایج شبیه سازی در شبکه SDPTA

طبق الگوریتم های فوق آزمایش هایی روی مثالهای AND و OR در محیط شبکه عصبی SDPTA جهت تنظیم پارامتر C به هر دو روش انجام شده است. در روش آتاماتون بازه تغییرات پارامتر C به ۱۰ عمل با عمق ۵ طبق الگوریتم فوق به محیط شبکه عصبی اعمال شده است که در آتاماتون های مختلف نتایج مختلفی مطابق شکل (۵) حاصل شد. برای بررسی و مقایسه بهتر روشهای سعی و خطا و آتاماتون، بهترین پاسخ آتاماتون با روش سعی و خطا در شکل (۶) آمده است.

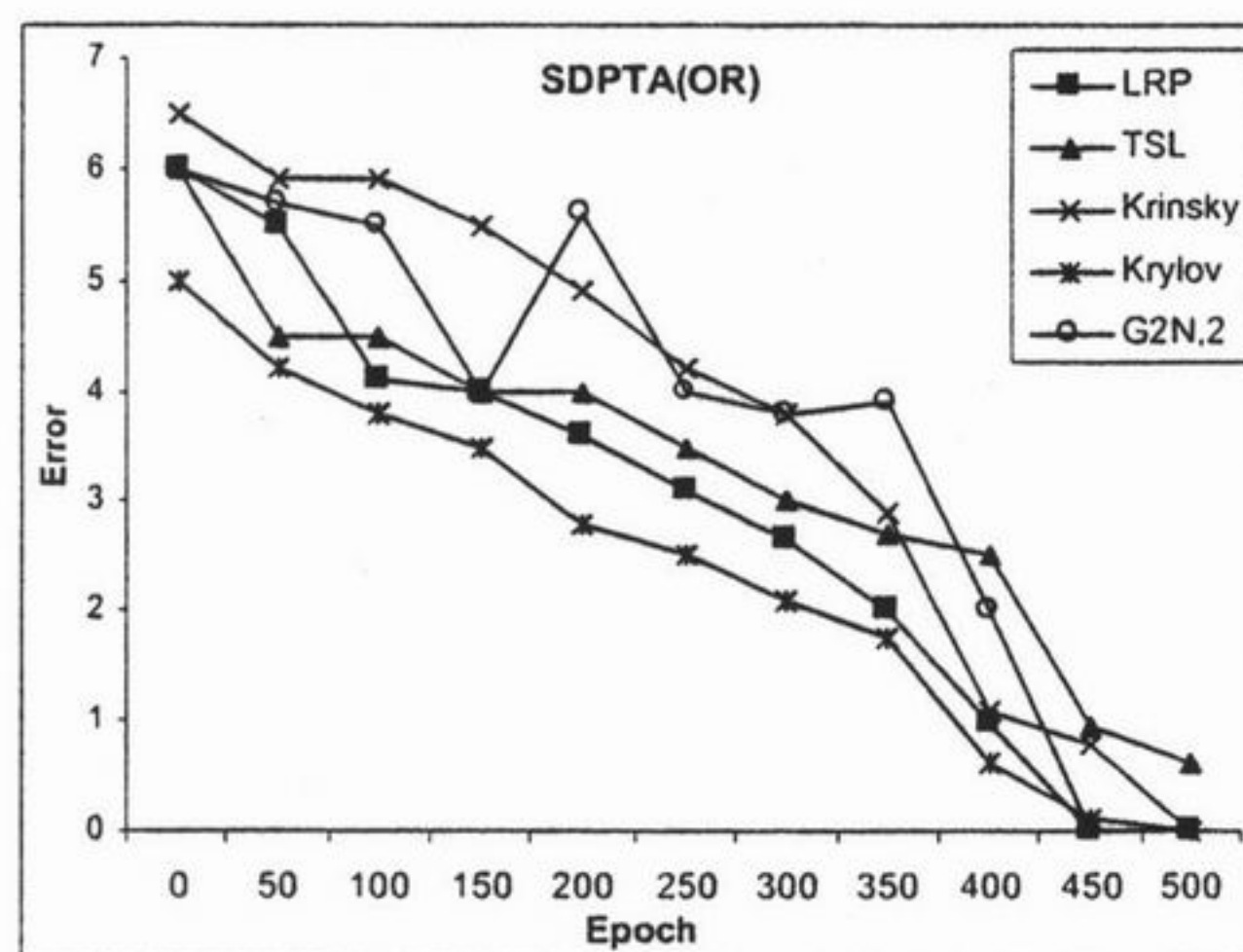
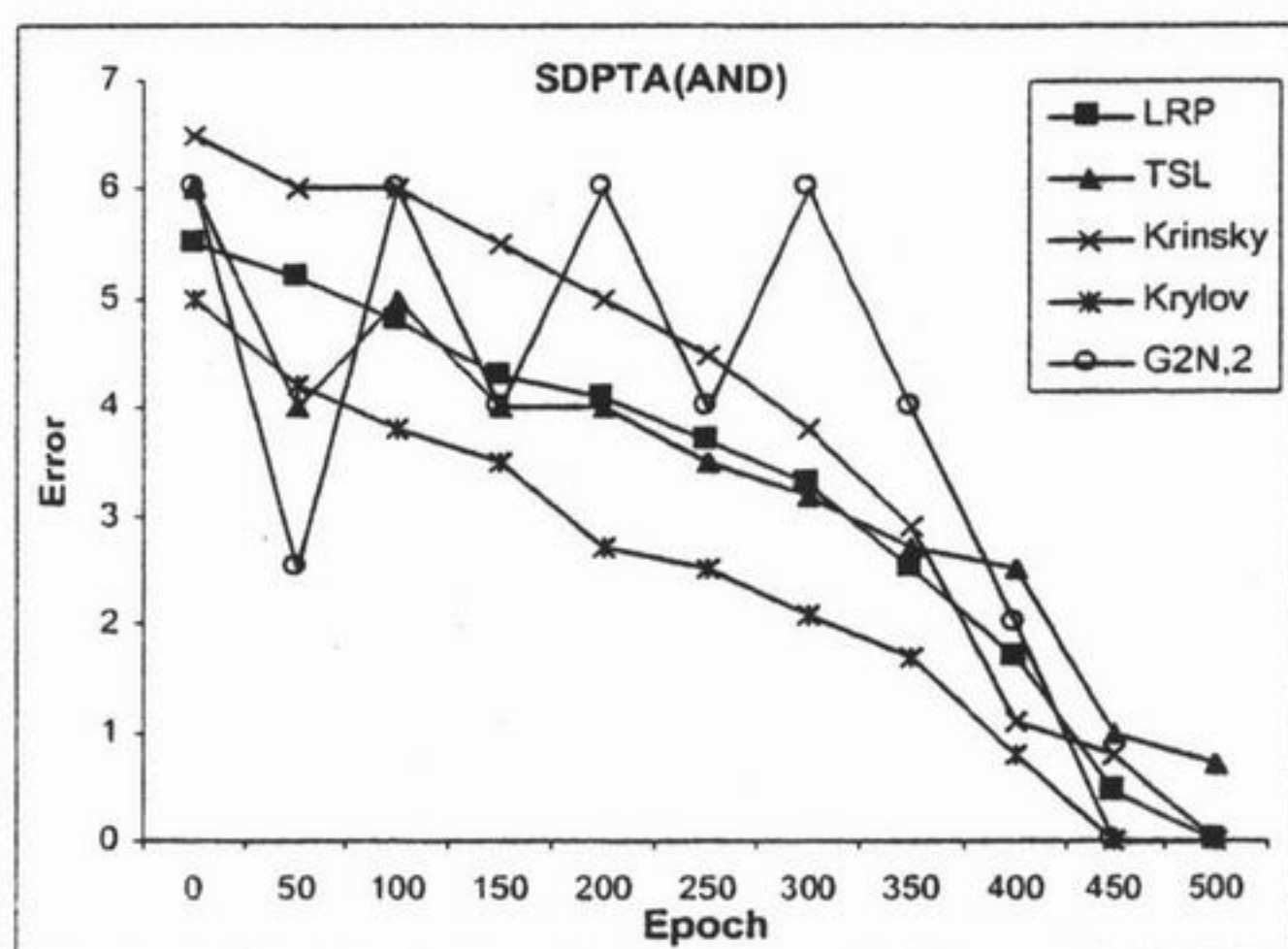
#### ۵- شبکه عصبی با الگوریتم SCPTA

در این شبکه پارامتر  $\eta$  که بعنوان پارامتر تصحیح کننده وزن در ساختار شبکه بکار می رود، نیاز به تنظیم دارد. این شبکه برای طبقه بندی الگوهایی که ساختار پیوسته ای دارند بسیار مناسب است. تابع تحریک این شبکه از نوع تابع sigmoid (رابطه ۴) می باشد [1, 7].

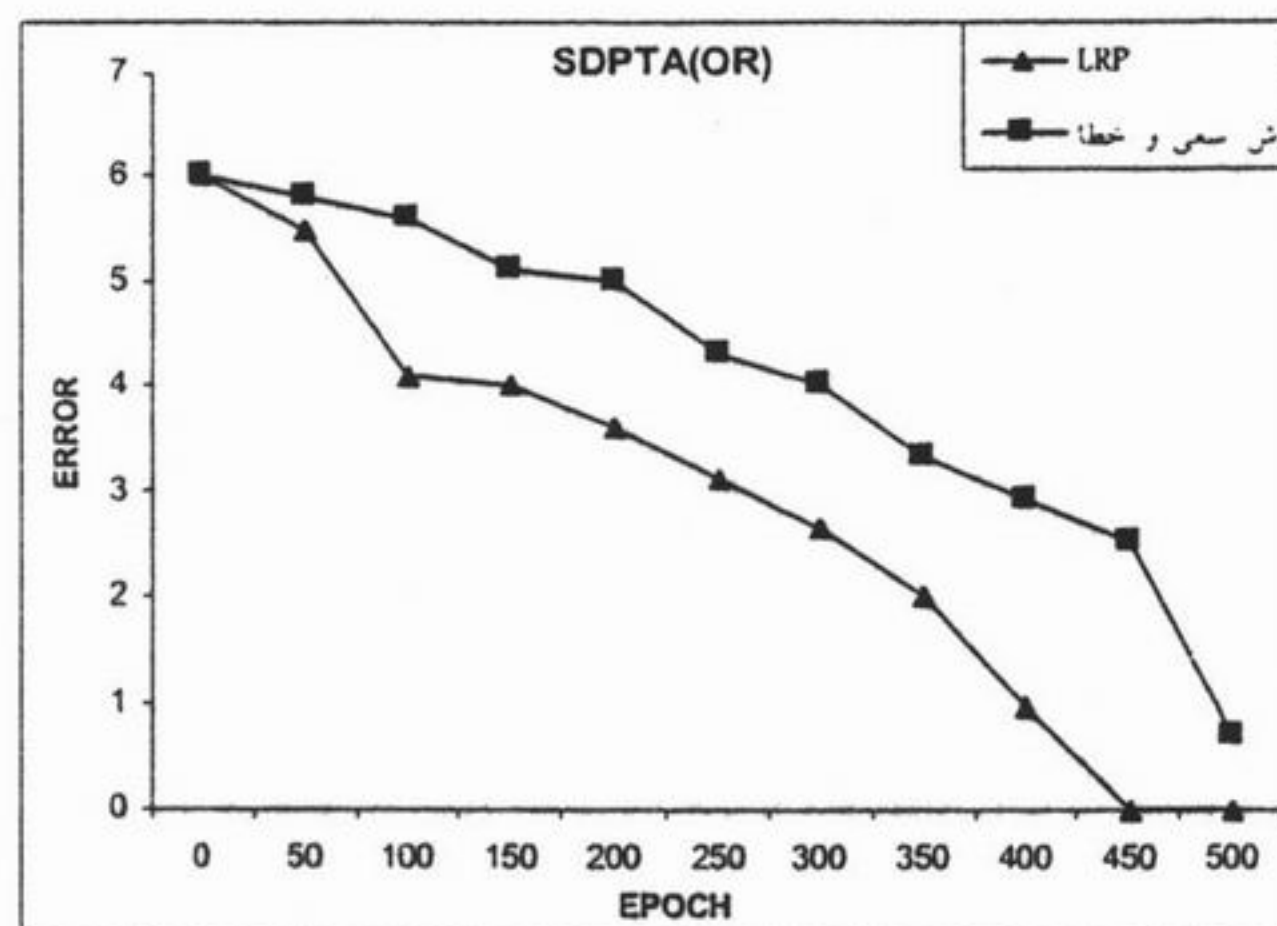
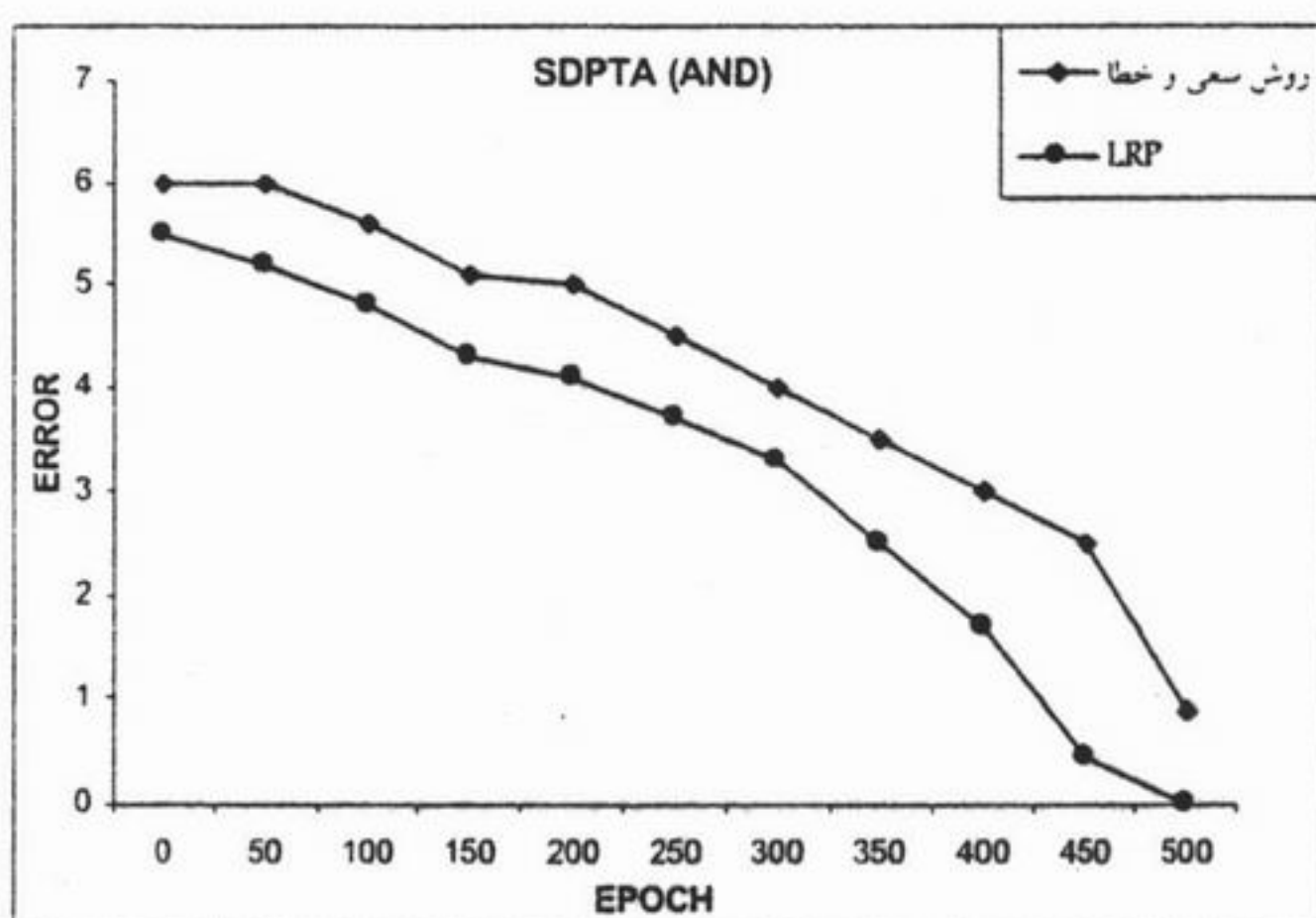
$$\frac{2}{1 + \exp(-\lambda_{net})} - 1$$

رابطه (۴):

ممکن است خطا همواره به صفر نرسد. لذا معمولاً یادگیری شبکه تا جایی ادامه می یابد که خطای شبکه از یک خطای ماکزیمم همواره کمتر باشد.



شکل (۵): نتایج شبیه سازی AND و OR در شبکه SDPTA با استفاده از الگوریتم های یادگیر



شکل (۶): مقایسه بهترین پاسخ آتاماتون های یادگیر با روش سعی و خطا در مثال های AND و OR

## ۵-۱- نتایج شبیه سازی در شبکه SCPTA

طبق الگوریتمهای شکل های (۷) و (۸) به تنظیم پارامتر  $\eta$  به روش سعی و خطا و آتاماتون در شبکه SCPTA پرداخته و نتایج بدست آمده را با یکدیگر مقایسه می کنیم. نتایج شبیه سازی AND و OR در شبکه عصبی SCPTA که بر روی الگوریتم های مختلف آتاماتونهای یادگیر (Krylov, Krinsky, LRP, TsetLine,  $G_{2N,2}$ ) انجام شده است، در شکل (۹) نشان داده شده است. بهترین پاسخ آتاماتون های یادگیر با روش سعی و خطا در شکل (۱۰) مقایسه شده است. در تنظیم پارامتر  $\eta$  به روش سعی و خطا مشاهده می شود که  $\eta$  با مقدار  $0.3$  نسبت به سایر مقادیر، خطای شبکه عصبی SCPTA را به مراتب کمتر می کند. که البته این تنظیم بسیار زمانگیر است. در شکل (۹) نمودار رسم شده SCPTA با مقدار  $\eta = 0.3$  مشاهده می شود که پس از ۲۰۰۰ تکرار مقدار خطا به  $0.65$  کاهش یافته است. سایر مقادیر  $\eta$  نتایج خوبی در کاهش خطا نداشتند.

Procedure Adaptparameter ( $\eta$ ) – SCPTA (random)

//Given are P training pairs  $\{x_1, d_1, x_2, d_2, \dots, x_p, d_p\}$ ,

Where  $x_i$  is  $(n*1)$ ,  $d_i$  is  $(R*1)$ ,  $i=1,2,\dots,P$ . Note that augmented input vectors are used:

$y_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix}$ , for  $i=1,2,\dots,P$  and K denotes the training step; p denotes the step counter within the training cycle

Step1: Initialize parameter( $\eta$ ) //  $\eta$  is positive random value between  $(0,1]$

$\lambda \leftarrow 1$

Chosen  $E_{MAX} > 0$

Step2: Initialize the weights to small random values

Initialize  $p \leftarrow 1, E \leftarrow 0, k \leftarrow 1$

Step3: Call (Training cycle) //  $y \leftarrow y_p, d \leftarrow d_p, O \leftarrow \text{sgn}(w^T y)$

Step4: Call (Weight update) //  $W \leftarrow W + \frac{1}{2} \eta (d-o)(1-o^2) y$

Step5: Call (Error Cycle) //  $E \leftarrow E + \frac{1}{2} (d-o)^2$

Step6: If  $p < P$  Then  $p \leftarrow p+1, k \leftarrow k+1$ , Goto Step3

End If

Step7: If  $E \geq E_{MAX}$  Then  $E \leftarrow 0, p \leftarrow 1$ , Goto step3

End If

End.

شکل (۷): شبه کد تنظیم پارامتر  $\eta$  در شبکه عصبی SCPTA به روش سعی و خطا

Procedure Adaptparameter ( $\eta$ ) – SCPTA (Automata)

//Given are P training pairs  $\{x_1, d_1, x_2, d_2, \dots, x_p, d_p\}$ ,

Where  $x_i$  is  $(n*1)$ ,  $d_i$  is  $(R*1)$ ,  $i=1,2,\dots,P$ . note that augmented input vectors are used:

$y_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix}$ , for  $i=1,2,\dots,P$  and K denotes the training step; p denotes the step counter within the training cycle

Note:// Select Action(i) from Automata with CDF random generation if Automata is nondeterministic.

Step1: Initialize parameter( $\eta$ ) //  $\eta$  is positive random value between  $(0,1]$

$\lambda \leftarrow 1$

$P \leftarrow$  number of Pattern

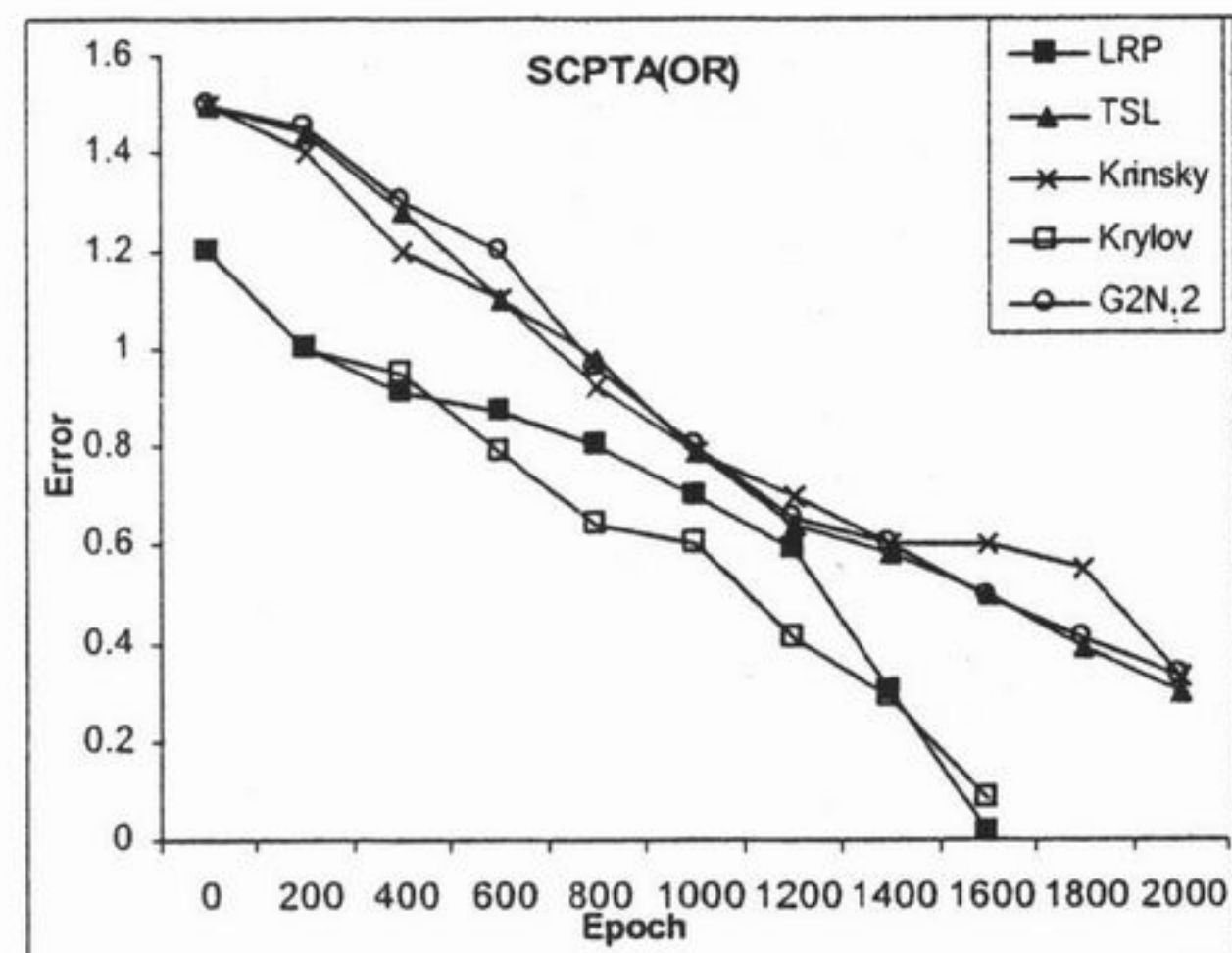
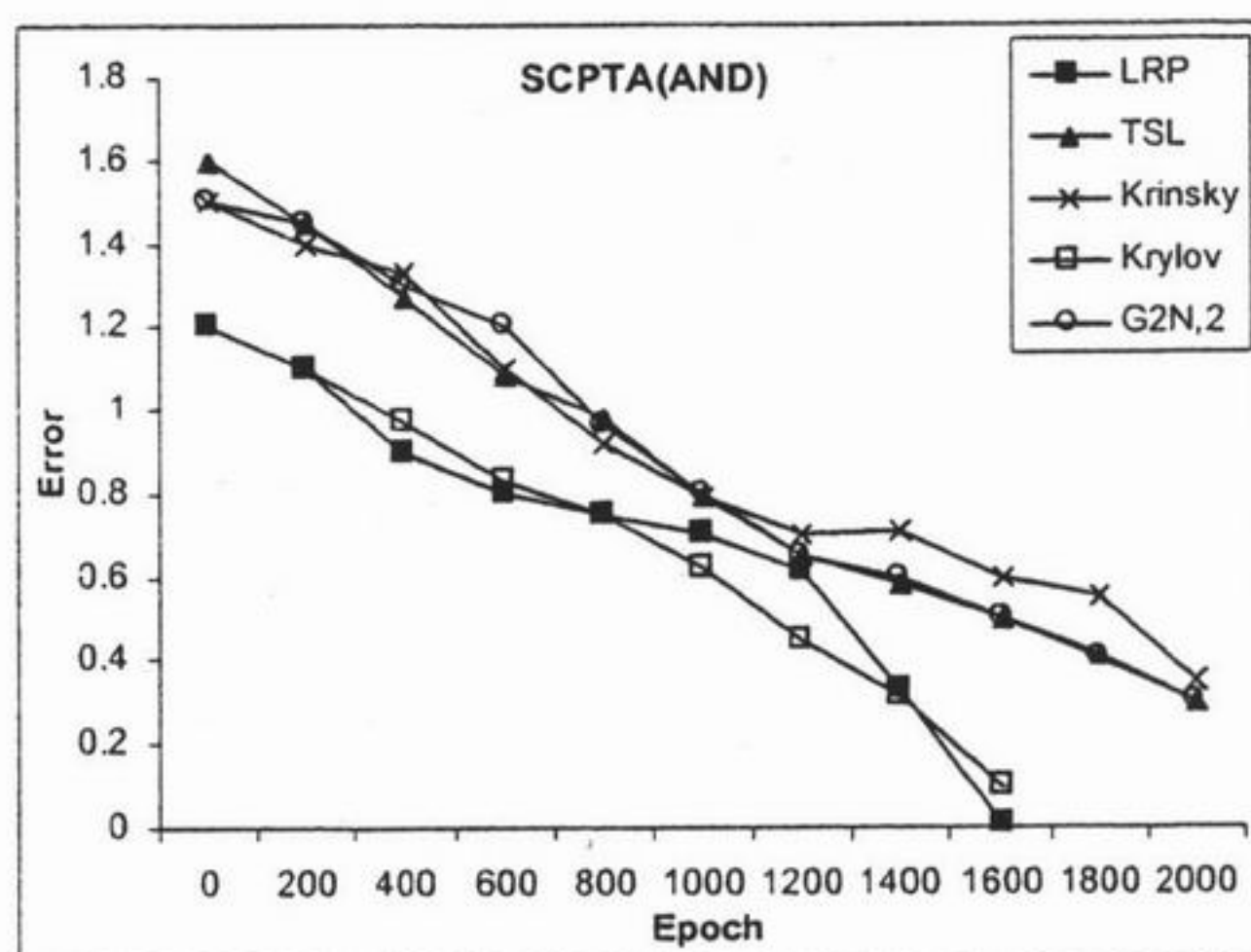
Chosen  $E_{MAX} > 0$

Step2: Initialize the weights to small random values

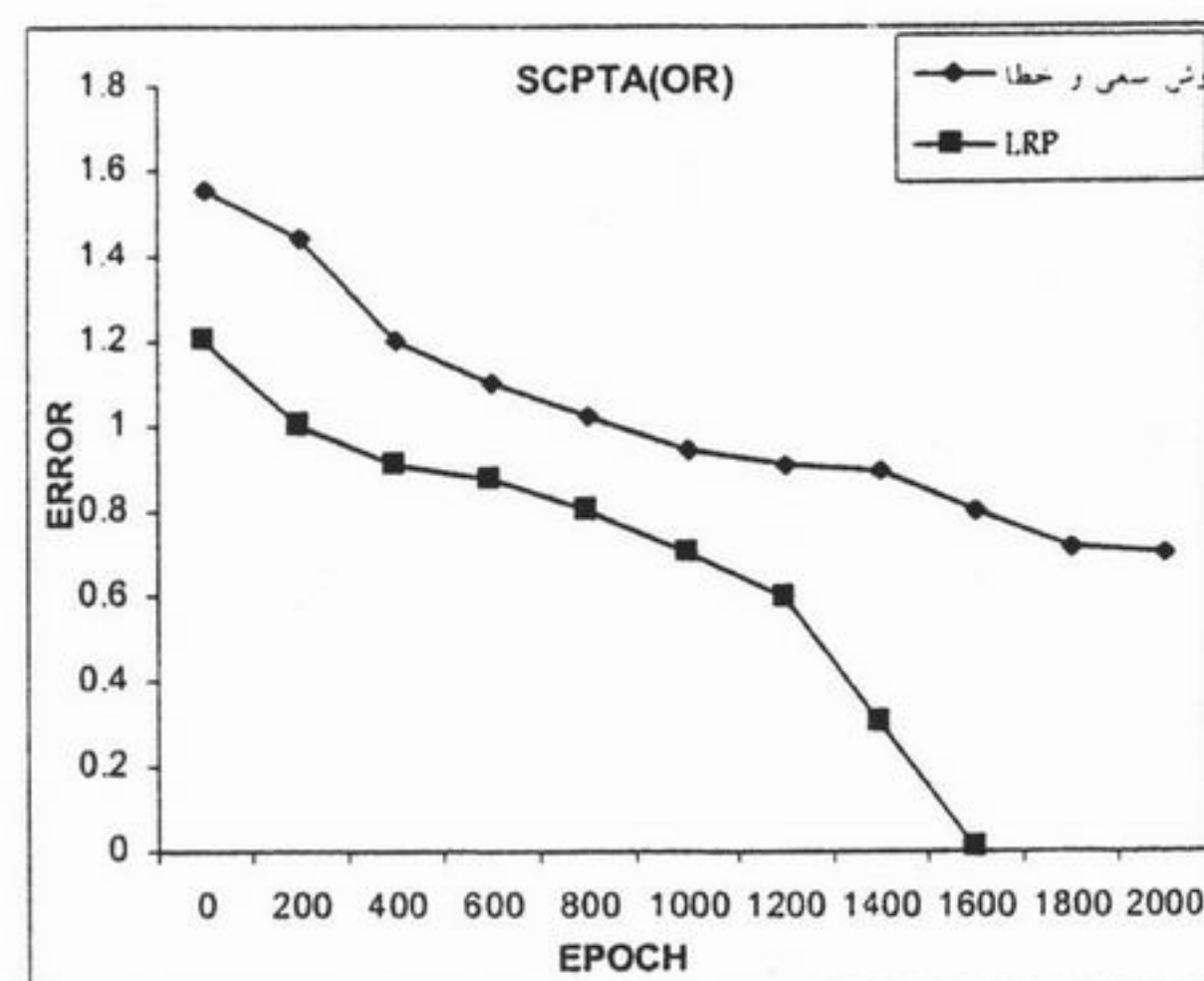
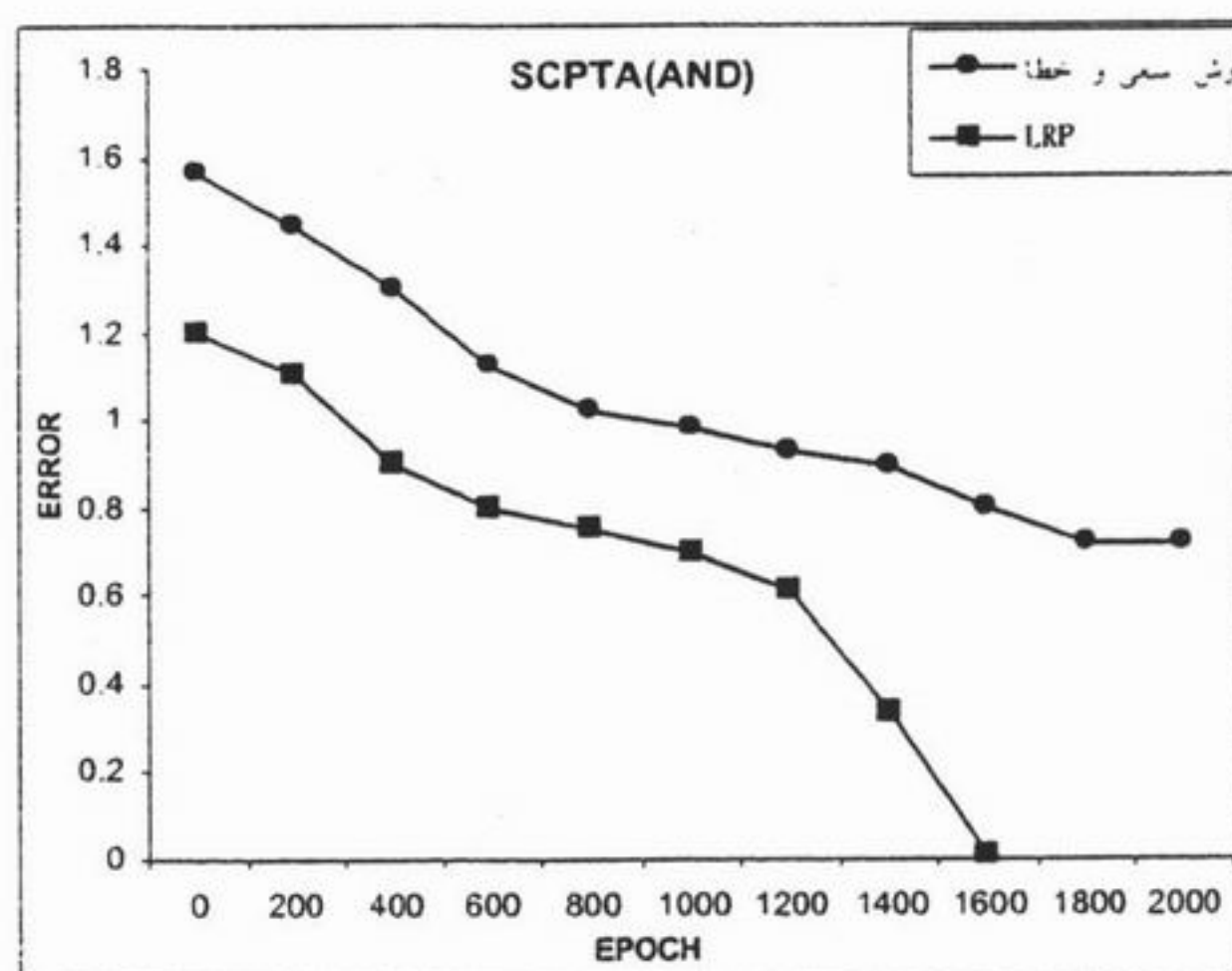
Initialize  $p \leftarrow 1, E \leftarrow 0, k \leftarrow 1, \text{flag} \leftarrow \text{false}$

Step3: Call (Training cycle) //  $y \leftarrow y_p, d \leftarrow d_p, O \leftarrow \text{sgn}(w^T y)$   
 Step4: Call (Weight update) //  $W \leftarrow W + \frac{1}{2} (\eta) (d-o)(1-o^2) y$   
 Step5: Call (Error Cycle) //  $E \leftarrow E + \frac{1}{2} (d-o)^2$   
 Step6: If  $p < P$  Then  $p \leftarrow p+1, k \leftarrow k+1$ , Goto Step3  
 End If  
 Step7: If  $E \leq E_{\max}$  and flag Then Goto Step8  
 Else  
 If  $E \leq E_{\max}$  and not flag Then  
 Flag  $\leftarrow$  True  
 Else Flag  $\leftarrow$  False  
 If  $E \geq E_{\max}$  Then  $E \leftarrow 0, p \leftarrow 1, \eta = \text{Automata}(1)$  //  $\beta=1$ , Goto step3  
 End If  
 Else  $\eta = \text{Automata}(0)$  //  $\beta=0$ , Goto step3  
 Step8: Exit  
 End.

شکل (۸): شبه کد تنظیم پارامتر  $\eta$  در شبکه عصبی SCPTA به روش آتاماتون



شکل (۹): نتایج شبیه سازی AND و OR در شبکه SCPTA با استفاده از آتاماتون های یادگیر



شکل (۱۰): مقایسه بهترین پاسخ آتاماتون های یادگیر با روش سعی و خطا در مثال AND و OR

## ۶- الگوریتم شبکه عصبی RDPTA

این الگوریتم که برای  $p$  الگوی ورودی  $x_p$  که از  $p$  سیگنال هدف استفاده می کند، الگوهای ورودی را به  $R$  کلاس طبقه بندی می کند. بردارهای ورودی را برای  $p$  الگو بصورت زیر تنظیم می شود [1, 7].

$$y_i = \begin{bmatrix} x_i \\ -1 \end{bmatrix} \quad i=1, 2, \dots, P$$

این شبکه بیشتر در مواقعی استفاده می شود که بخواهیم  $p$  الگو را در  $R$  کلاس طبقه بندی نماییم. در الگوریتم فوق،  $k$  تعداد مراحل یادگیری و  $p$  به الگویی که در حال یادگیری شدن است، اشاره دارد. الگوریتم RDPTA با کمی اختلاف مشابه الگوریتم SDPTA بوده و شبه کد آن در دو حالت سعی و خطا و آتاماتون مطابق شکل های (۱۱) و (۱۲) می باشد.

Procedure Adaptparameter (c) – RDPTA (random)

// Given are  $P$  training pairs  $\{x_1, d_1, x_2, d_2, \dots, x_p, d_p\}$ ,

Where  $x_i$  is  $(n \times 1)$ ,  $d_i$  is  $(R \times 1)$ ,  $i=1, 2, \dots, P$ . note that augmented input vectors are used:

$y_i = \begin{bmatrix} x_i \\ -1 \end{bmatrix}$ , for  $i=1, 2, \dots, P$  and  $K$  denotes the training step;  $p$  denotes the step counter within the training cycle

Step1: Initialize parameter(C) // C is positive random value between(0,1]

Step2: Initialize the weights to small random values

Initialize  $p \leftarrow 1, E \leftarrow 0, k \leftarrow 1, E \leftarrow 0$

Step3: Call (Training cycle) //  $y \leftarrow y_p, d \leftarrow d_p$

$O_i \leftarrow \text{sgn}(w_i^T y)$  //  $i:1..R$

Step4: Call (Weights update) //  $W_i \leftarrow W_i + \frac{1}{2} C (d_i - o_i) y$

Step5: Call (Error Cycle) //  $E \leftarrow E + \frac{1}{2} (d_i - o_i)^2$

Step6: If  $p < P$  Then  $p \leftarrow p+1, k \leftarrow k+1$ , Goto Step3

End If

Else Goto Step7

Step7: If  $E > 0$  Then  $E \leftarrow 0, p \leftarrow 1$ , Goto step3

End If

Else Exit

End.

شکل (۱۱): شبه کد تنظیم پارامتر  $C$  در شبکه عصبی RDPTA به روش سعی و خطا

Procedure Adaptparameter (c) – RDPTA (Automata)

// Given are  $P$  training pairs  $\{x_1, d_1, x_2, d_2, \dots, x_p, d_p\}$ ,

Where  $x_i$  is  $(n \times 1)$ ,  $d_i$  is  $(R \times 1)$ ,  $i=1, 2, \dots, P$ . note that augmented input vectors are used:

$$y_i = \begin{bmatrix} x_i \\ -1 \end{bmatrix}, \text{ for } i=1, 2, \dots, P$$

and  $K$  denotes the training step;  $p$  denotes the step counter within the training cycle and  $N$  denotes the MaxIteration

Note:// Select Action(i) from Automata with CDF random generation if Automata is Nondeterministic.

Step1: Initialize parameter(C) // C is positive random value between(0,1]

Step2: Initialize the weights to small random values

Initialize  $p \leftarrow 1, E \leftarrow 0, k \leftarrow 1, E \leftarrow 0$

Repeat

```

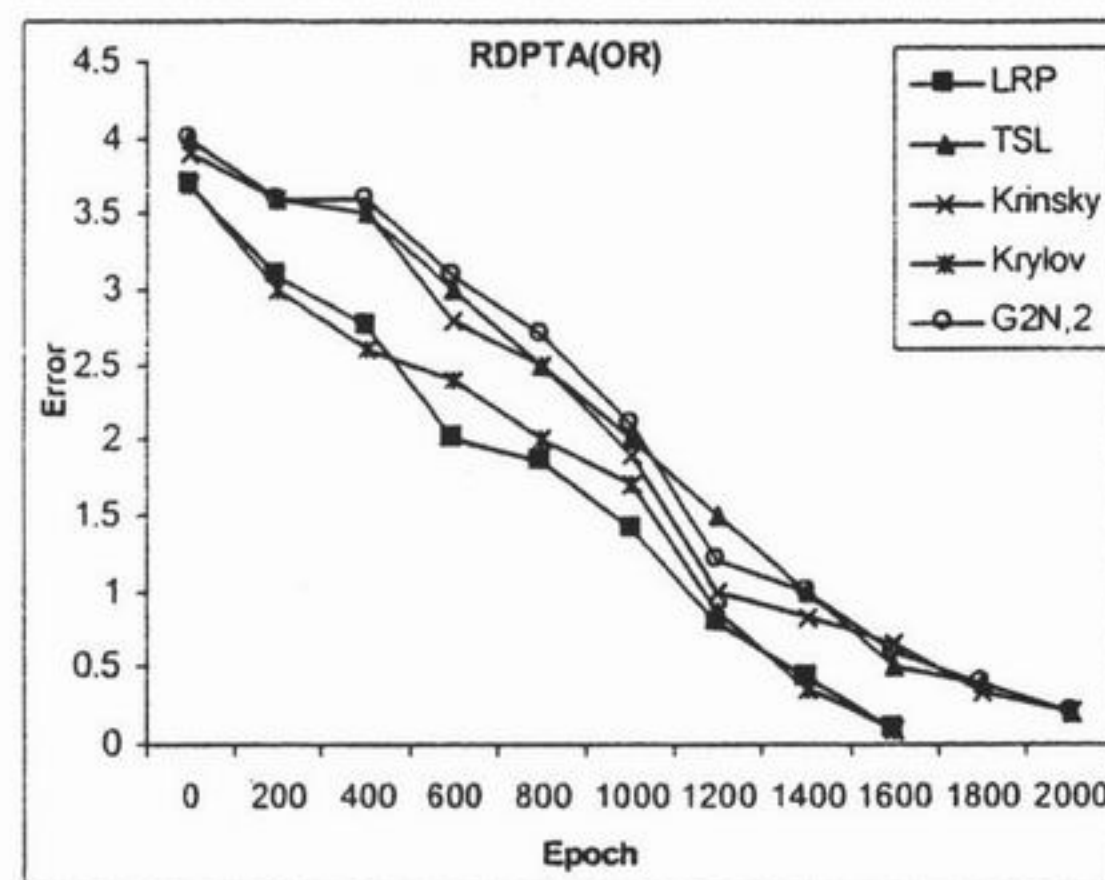
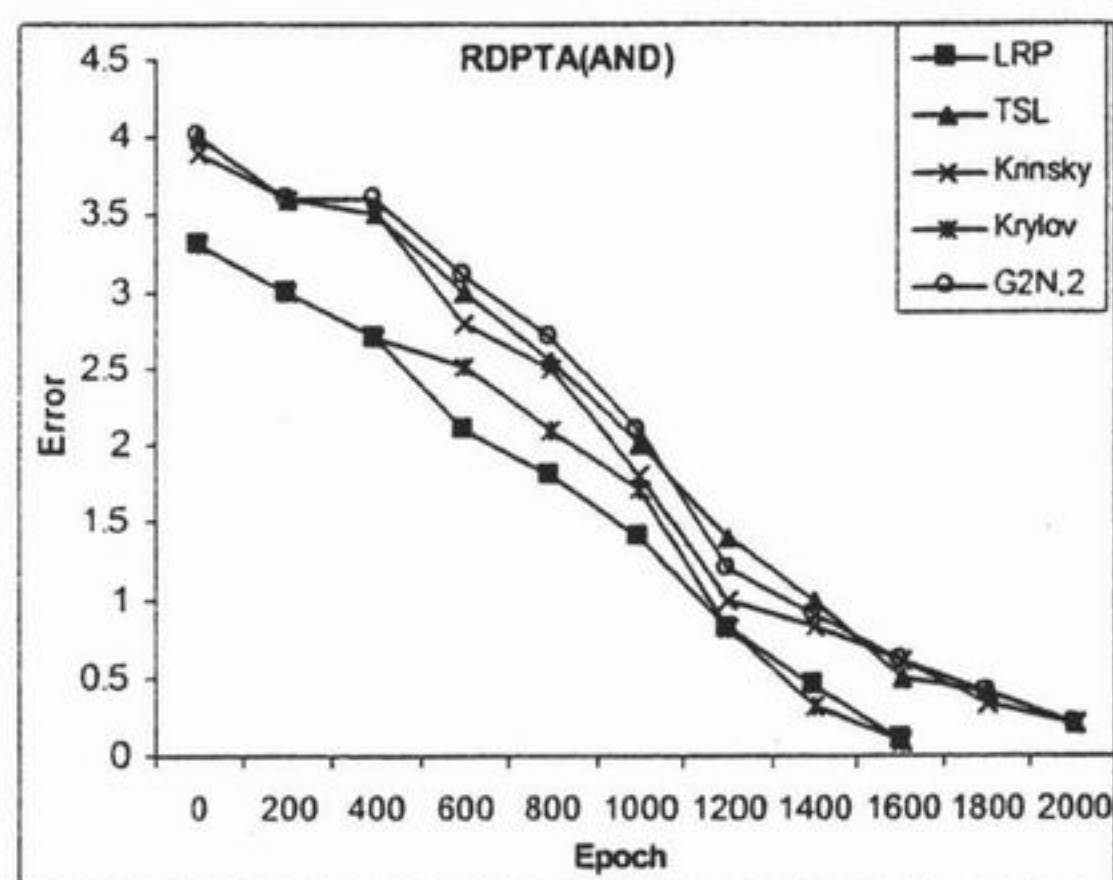
Step3: Call (Training cycle) //  $y \leftarrow y_p, d \leftarrow d_p$ 
       $O_i \leftarrow \text{sgn}(w_i^T y)$  //  $i: 1..R$ 
Step4: Call (Weights update) //  $W_i \leftarrow W_i + \frac{1}{2} C (d_i - o_i) y$ 
Step5: Call (Error Cycle) //  $E \leftarrow E + \frac{1}{2} (d_i - o_i)^2$ 
Step6: If  $<P$  Then  $p \leftarrow p+1, k \leftarrow k+1$ , Goto Step3
      End If
      Else Goto Step7
Step7: If  $E > 0$  Then  $E \leftarrow 0, p \leftarrow 1, C \leftarrow \text{Automata}(\beta)$  //  $\beta=1$ 
      End If
      Else  $C \leftarrow \text{Automata}(\beta)$  //  $\beta=0$ 
Until  $N \geq \text{MaxIteration}$ 
End.

```

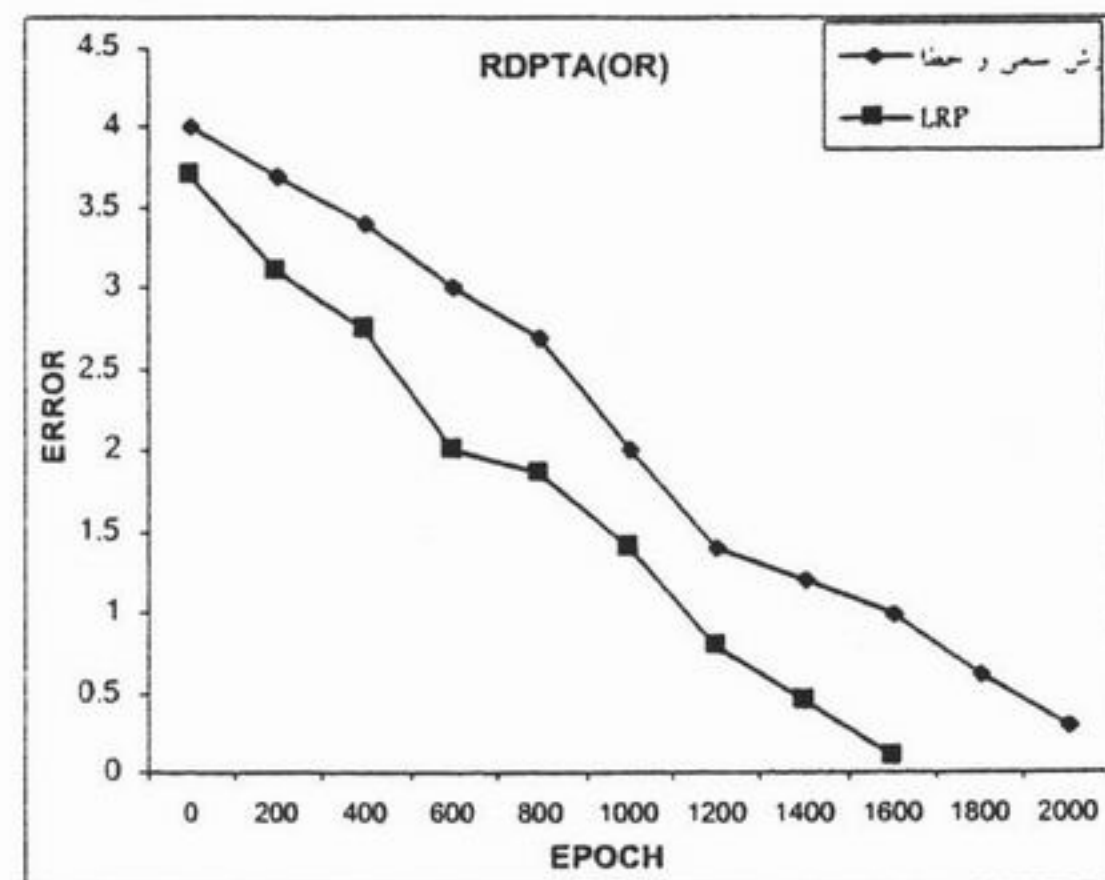
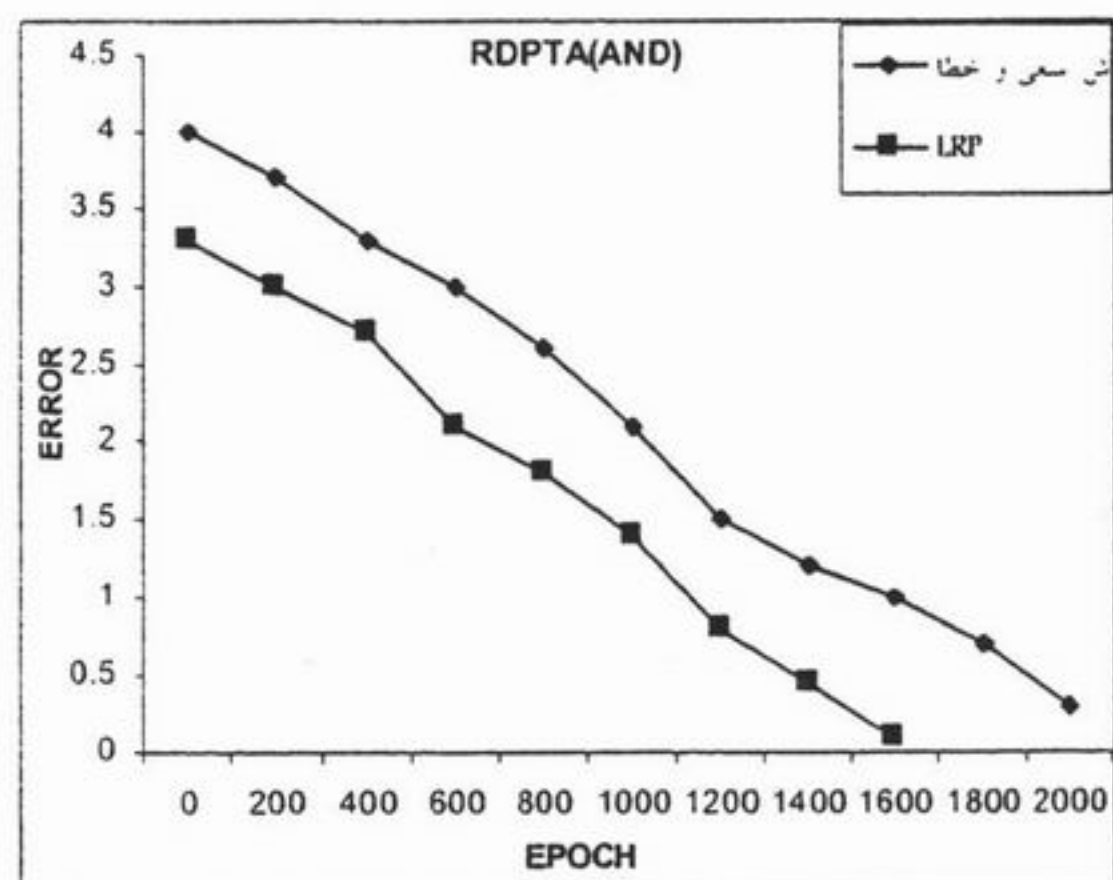
شکل (۱۲): شبه کد تنظیم پارامتر  $c$  در شبکه عصبی RDPTA به روش آتاماتون

### ۱-۶- نتایج شبیه سازی در شبکه عصبی RDPTA

نتایج شبیه سازی AND و OR در شبکه عصبی RDPTA که بر روی الگوریتم های مختلف آتاماتونهای یادگیر آتاماتون های یادگیر انجام شده است، در شکل (۱۳) نشان داده شده است. بهترین پاسخ آتاماتون های یادگیر با روش سعی و خطا در شکل (۱۴) مقایسه شده است.



شکل (۱۳): نتایج شبیه سازی AND و OR در الگوریتم RDPTA با استفاده از آتاماتون های یادگیر



شکل (۱۴): مقایسه بهترین پاسخ آتاماتون های یادگیر با روش سعی و خطا در مثال AND و OR

## ۷- شبکه عصبی با الگوریتم EBPTA

الگوریتم شبکه مزبور مطابق شبکه عصبی پرسپترون چند لایه با الگوریتم پس انتشار خطا می باشد. در تنظیم پارامتر  $\eta$  به روش سعی و خطا و آتاماتون از شبه کدهای شکل های (۱۵) و (۱۶) استفاده شده است [1, 7, 8, 9, 10].

## ۷-۱- نتایج شبیه سازی در شبکه EBPTA

مشاهده می شود که  $\eta$  با مقدار ۰/۳ نسبت به سایر مقادیر، خطای شبکه عصبی EBPTA را به مراتب کمتر می کند که البته این تنظیم بسیار زمانگیر است. در شکل (۱۷) نمودار رسم شده EBPTA با مقدار  $\eta = 0/3$  مشاهده می شود که پس از ۱۶۰۰ تکرار مقدار خطا به ۰/۱ کاهش یافته است. شکل (۱۸) مقایسه تنظیم پارامتر  $\eta$  به روش سعی و خطا و آتاماتون در مثال های XOR و NOR می باشد.

Procedure Adaptparameter ( $\eta$ ) – EBPTA (random)  
 //Given are P training pairs  $\{z_1, d_1, z_2, d_2, \dots, z_p, d_p\}$ ,  
 Where  $z_i$  is  $(I \times 1)$ ,  $d_i$  is  $(K \times 1)$ ,  $i=1, 2, \dots, P$ . Note that  $I$ 'th component of each  $z_i$  is of value  $-1$  since input vectors have been augmented. Zise  $J-1$  of the hidden layer having outputs  $y$  is selected. Note that  $J$ 'th component of  $y$  is of value  $-1$ , since hidden layer outputs have also been augmented:  $y$  is  $(J \times 1)$  and  $o$  is  $(k \times 1)$ .  
 Step1: Initialize parameter( $\eta$ ) //  $\eta$  is positive random value between  $(0, 1]$   
 $P \leftarrow$  number of Pattern, Chosen  $E_{MAX} > 0$   
 Initialize the weights ( $W_{k \times j}, V_{j \times i}$ ) with small random values, Initialize  $p \leftarrow 1, E \leftarrow 0, k \leftarrow 1$   
 Step2: Call (Training cycle) //  $z \leftarrow z_p, d \leftarrow d_p, y_j \leftarrow f(V_j^T z)$  for  $j=1..J$   
 //  $o_k \leftarrow f(W_k^T y)$  for  $k=1..K$   
 Step3: Call (Error Cycle) //  $E \leftarrow E + \frac{1}{2} (d_k - o_k)^2$  for  $k=1..K$   
 Step4: Call (Error signal vectors) //  $\delta_{ok} = \frac{1}{2} (d_k - o_k)^2 (1 - o_k^2)$  for  $k=1..K$   

$$\delta_{y_j} = \frac{1}{2} (1 - Y_j^2) \sum_{k=1}^K \delta_{ok} W_{kj} \quad \text{for } j = 1..J$$
  
 Step5: Call (output layer Weights adjusted) //  $W_{kj} \leftarrow W_{kj} + \eta \delta_{ok} y_j$  for  $k=1..K$  and  $j=1..J$   
 Step6: Call (hidden layer Weights adjusted) //  $V_{kj} \leftarrow V_{kj} + \eta \delta_{y_j} z_j$  for  $j=1..J$  and  $i=1..I$   
 Step7: If  $p < P$  Then  $p \leftarrow p+1, q \leftarrow q+1$ , Goto Step2 End If  
 Else Goto Step8  
 Step8: If  $E < E_{max}$  Then Exit  
 Else  
 If  $E > E_{max}$  Then  $E \leftarrow 0, p \leftarrow 1$ , Goto step2  
 End.

شکل (۱۵): شبه کد تنظیم پارامتر  $\eta$  در شبکه عصبی EBPTA به روش سعی و خطا

Procedure Adaptparameter ( $\eta$ ) – EBPTA (Automata)  
 //Given are P training pairs  $\{z_1, d_1, z_2, d_2, \dots, z_p, d_p\}$ ,  
 Where  $z_i$  is  $(I \times 1)$ ,  $d_i$  is  $(K \times 1)$ ,  $i=1, 2, \dots, P$ . Note that  $I$ 'th component of each  $z_i$  is of value  $-1$  since input vectors have been augmented. Zise  $J-1$  of the hidden layer having outputs  $y$  is selected. Note that  $J$ 'th component of  $y$  is of value  $-1$ , since hidden layer outputs have also been augmented:  $y$  is  $(J \times 1)$  and  $o$  is  $(k \times 1)$ .

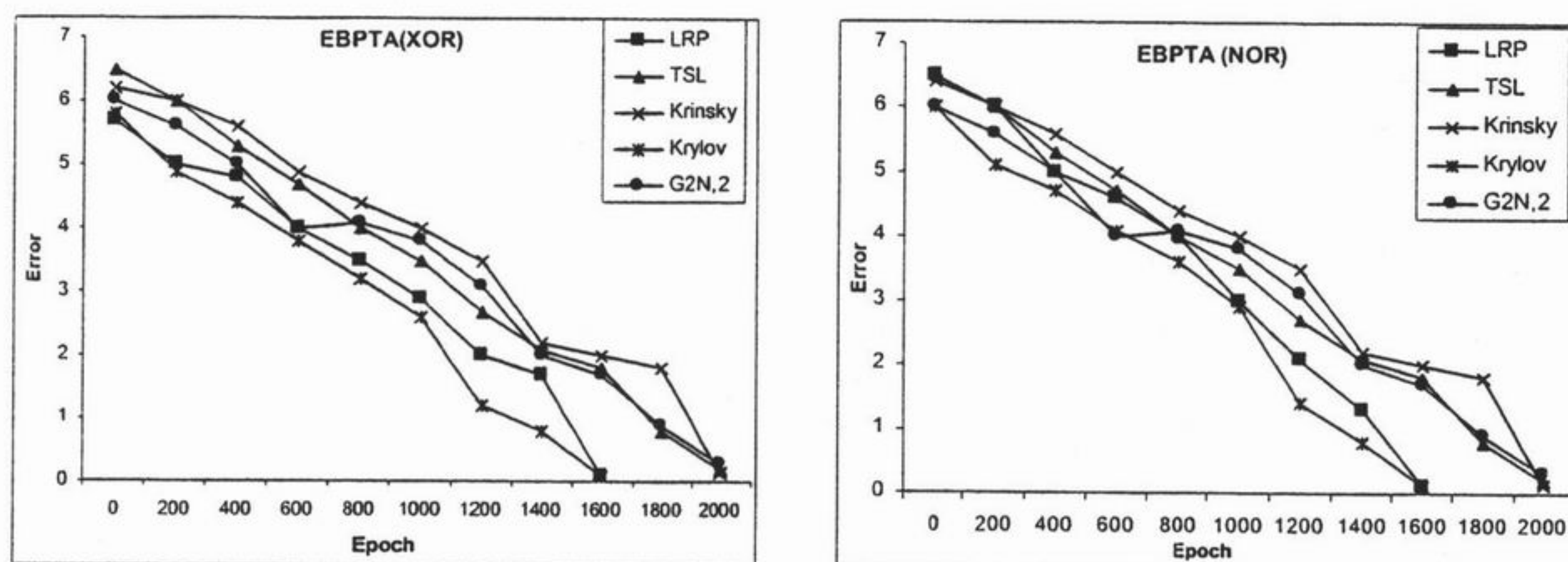
Note:// Select Action( $i$ ) from Automata with CDF random generation if Automata is Nondeterministic.

Step1: Initialize parameter( $\eta$ ) //  $\eta$  is positive random value between  $(0, 1]$

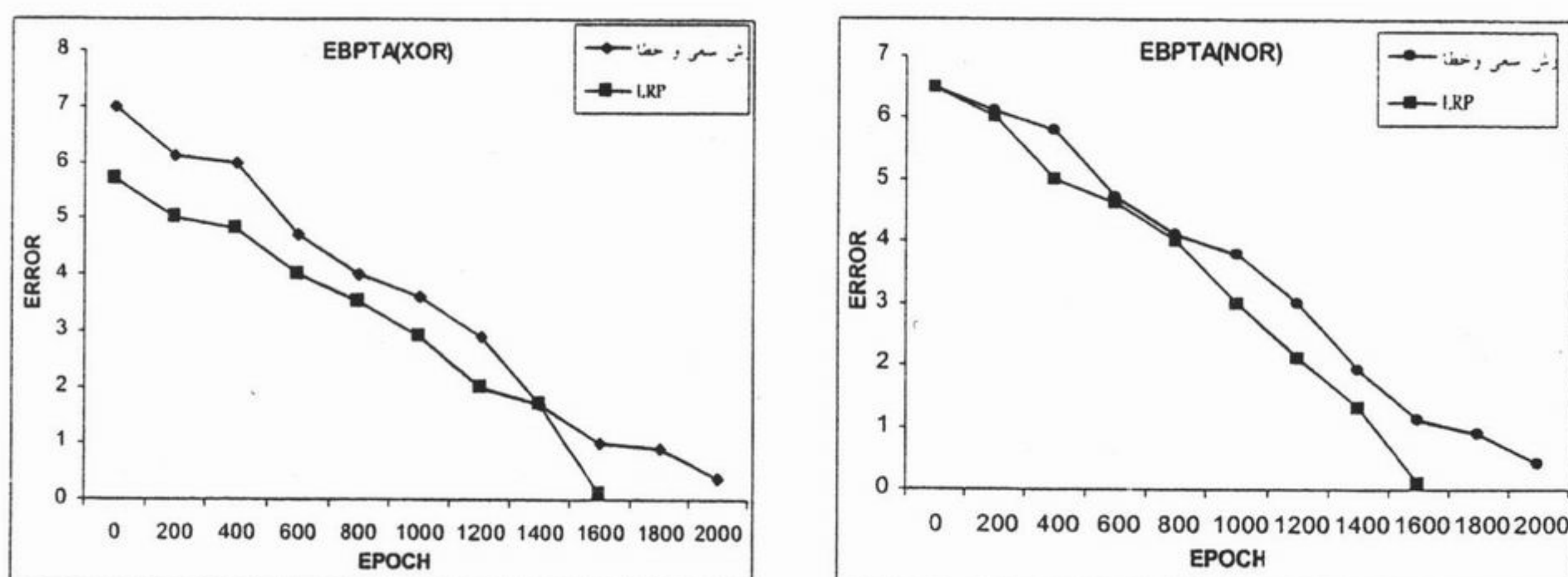
$P \leftarrow$  number of Pattern  
 Chosen  $E_{MAX} > 0$   
 Initialize the weights ( $W_{k*j}, V_{j*1}$ ) with small random values  
 Initialize  $p \leftarrow 1, E \leftarrow 0, k \leftarrow 1$   
 Step2: Call (Training cycle) //  $z \leftarrow z_p, d \leftarrow d_p, y_j \leftarrow f(V_j^T z)$  for  $j:1..J, o_k \leftarrow f(W_k^T y)$  for  $k:1..K$   
 Step3: Call (Error Cycle) //  $E \leftarrow E + \frac{1}{2} (d_k - o_k)^2$  for  $k:1..K$   
 Step4: Call (Error signal vectors) //  $\delta_{ok} = \frac{1}{2} (d_k - o_k)^2 (1 - o_k^2)$  for  $k:1..K$   

$$\delta_{yj} = \frac{1}{2} (1 - Y_j^2) \sum_{k=1}^K \delta_{ok} W_{kj} \quad \text{for } j:1..J$$
  
 Step5: Call (output layer Weights adjusted) //  $W_{kj} \leftarrow W_{kj} + \eta \delta_{ok} y_j$  for  $k=1..K$  and  $j=1..J$   
 Step6: Call (hidden layer Weights adjusted) //  $V_{kj} \leftarrow V_{kj} + \eta \delta_{yj} z_j$  for  $j=1..J$  and  $i=1..I$   
 Step7: If  $p < P$  Then  $p \leftarrow p+1, q \leftarrow q+1$ , Goto Step2 End If  
 Else Goto Step8  
 Step8: If  $E < E_{max}$  and  $p < P$  Then  
 $\eta \leftarrow \text{Automata}(\beta)$  //  $\beta=0$ , Goto step2  
 End If  
 Else  
 If  $E > E_{max}$  Then  $\eta \leftarrow \text{Automata}(\beta)$  //  $\beta=1$ ,  $E \leftarrow 0$ ,  $p \leftarrow 1$ , Goto step2 End If  
 Else If  $E \leftarrow 0$  and  $p < P$  Then Exit  
 End.

شکل (۱۶): شبه کد تنظیم پارامتر  $\eta$  در شبکه عصبی EBPTA به روش آتاماتون



شکل (۱۷): نتایج شبیه سازی XOR و NOR در الگوریتم EBPTA با استفاده از آتاماتون های یادگیر



شکل (۱۸): مقایسه بهترین پاسخ آتاماتون های یادگیر با روش سعی و خطا در مثال های XOR و NOR

## ۸- نتیجه گیری

در این مقاله از انواع آتاماتونها (Krylov, Krinsky, LRP, TsetLine,  $G_{2N,2}$ ) جهت تنظیم پارامترهای الگوریتم شبکه های عصبی استفاده شده است. با مقایسه نتایج بدست آمده به روش آتاماتون و روش سعی و خطا در تنظیم پارامتر شبکه های فوق می توان چنین استنباط نمود که روش آتاماتون نسبت به روش سعی و خطا، بهتر و بهینه تر عمل می کند و به لحاظ زمانی در تنظیم پارامتر سریعتر است.

## ۹- مراجع

- [1] E. D. Rumelhart, E. G. Hinton and J. R. Williams, "Learning Representations By Backpropagation Errors," *Nature*, vol. 323, pp. 533-536, 1986.
- J. M. Zurada, Introduction to artificial neural systems, west publishing Company, 1974.
- [2] A. Blum, Neural Networks In C++, USA, John willey & Sons, 1992.
- [3] S. Haykin, Neural Networks: A comprehensive Foundation, USA, Prentice Hall, 1994.
- [4] Y. Pao, Adaptive Pattern Recognition And Neural Networks, USA, Addison -Wesley, 1989.
- [5] T. Master, Practical Neural Networks Recipes in C++, UK, Academic Press, 1993.
- [6] J. Hertz, A. Krogh and R.G. Palmer, Introduction To The Theory Of Neural Computation, USA, Addison-Wesley, 1991.
- [7] A. K. Jain, J. Mao and K. M. Mohiuddin, "Artificial Neural Networks: A Tutorial," *Computer*, pp. 31-44, Mar. 1996.
- [8] K. S. Narendra, M. A. Thathachar, Learning Automata An Introduction , prentice Hall, 1989.
- [9] H. Beigy, M. R. Meybodi, "New Classes OF Learning Automata Based Schemes For Adaptaion Of Backpropagation Algorithm Parameters," *Iranian Journal Of Science & Technology, Transaction B*, Vol. 25, No. B3, Printed in Islamic Republic of Iran, 2001.
- [10] H. Beigy, M. R. Meybodi, "Backpropagation Algorithm Adaptation Parameters Using Learning Automata," *International Journal of neural Systems*, vol. 11, pp.219-228, 2001.