

# آموزش شبکه‌های عصبی چند لایه با $\square$ به کارگیری الگوریتم PSO

علی کاشفی کاویانی، سید علی پورموسوی کانی، علی جهانبانی اردکانی

دانشگاه صنعتی امیرکبیر، دانشکده مهندسی برق، قطب علمی قدرت

ali\_jahanbani@aut.ac.ir

چکیده - هدف از آموزش شبکه‌های عصبی، یافتن اندازه‌وزنها و بایاس‌ها به نحوی است که خطای داده‌های آموزش را به حداقل ممکن برساند. لذا آموزش شبکه‌های عصبی را می‌توان در بردارنده یک مسأله بهینه‌سازی دانست که هدف از آن بهینه‌سازی ضرایب وزنی و بایاس‌ها جهت دستیابی به حداقل خطای آموزش می‌باشد. در روش‌های مرسوم جهت آموزش شبکه‌های عصبی از الگوریتم پس‌انتشار و دیگر روش‌های گرادیانی استفاده می‌شود. این روش‌ها در مواردی که شکل تابع غیرخطی و پیچیده باشد، ضعف و ناکارآمدی خود را نشان می‌دهند. در این مقاله، آموزش شبکه عصبی با استفاده از الگوریتم PSO انجام شده است. به کارگیری الگوریتم PSO در آموزش شبکه‌های عصبی و مقایسه صورت گرفته با الگوریتم آموزشی پس انتشار، نشان می‌دهد که در مسائل پیچیده توأم با فرآیندهای غیرخطی، استفاده از الگوریتم جدید کارآیی بالاتری را به همراه دارد. در انتها نتایج پیاده‌سازی الگوریتم ارائه شده بر روی دو مسأله و مقایسه آن با روش آموزشی پس انتشار آورده شده است.

کلیدواژه- شبکه‌های عصبی چند لایه، الگوریتم PSO، الگوریتم پس‌انتشار، کمینه‌سازی مربعات خطا.

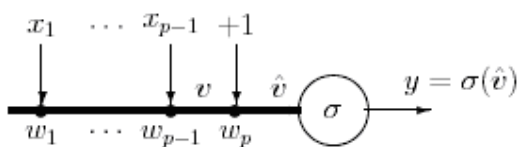
## 1- مقدمه

داده‌های ورودی و خروجی را از طریق ضرایب وزنی، بایاس‌ها و توابع اعمالی به خروجی‌های هر لایه، شکل بدهند. الگوریتم‌های آموزشی متنوعی جهت آموزش شبکه‌های عصبی به کار گرفته شده است. از مهم‌ترین الگوریتم‌های آموزشی می‌توان از الگوریتم پس‌انتشار (Back Propagation) نام برد. در الگوریتم پس‌انتشار در هر مرحله مقدار خروجی محاسبه شده جدید، با مقدار واقعی مقایسه شده، و با توجه به خطای بدست آمده به اصلاح وزن‌ها و بایاس‌های شبکه پرداخته می‌شود. به نحوی که در انتهای هر تکرار اندازه خطای حاصله کمتر از میزان بدست آمده در تکرار قبلی باشد. اساس این کمینه‌سازی، حرکت بر روی بردار گرادیان تابع مربعات خطای شبکه می‌باشد، که این بردار نیز به نوبه خود بوسیله مشتق‌گیری زنجیره‌ای از تابع خطا نسبت به تک تک پارامترهای شبکه بدست می‌آید.

در [3] نشان داده شده است که سیستم‌های فازی و شبکه‌های عصبی قابلیت بسیار مناسبی برای استفاده در مسائل پیش‌بینی دارند. اگر چه استفاده از الگوریتم

شبکه‌های عصبی، الگوریتم‌های غیرخطی محاسباتی برای پردازش تصویر، سیگنال و داده‌های عددی هستند. در طول چند دهه گذشته، از شبکه‌های عصبی مصنوعی به طور گسترده‌ای در زمینه‌های مختلف استفاده شده است [1]. چندین مشخصه شبکه‌های عصبی مانند دینامیک درونی شبکه‌های عصبی در پیش‌بینی، تغییرات خطای اطلاعات و عدم نیاز به اطلاعات اضافی بر داده‌های ورودی، استفاده از شبکه عصبی را در بسیاری از مسائل مهندسی جذاب می‌کند. یکی از ساختارهای شبکه عصبی، ساختار پرسپترون چند لایه است. می‌توان یک پرسپترون چند لایه را به وسیله توابع غیر خطی به گونه‌ای آموزش داد که بتواند هر تابع قابل اندازه‌گیری را تقریب زده و پیش‌بینی کند. این فرآیند در حالی صورت می‌گیرد که شبکه اطلاعاتی درباره توزیع ورودی‌ها نیاز ندارد [2]. شبکه‌های عصبی با استفاده از مجموعه داده‌های واقعی ورودی و خروجی، الگوریتم‌های آموزشی را به کار می‌گیرند تا ارتباطات پنهانی میان

یک پرسپترون با ورودی بایاس نشان داده شده است.



شکل (1): یک پرسپترون با ورودی بایاس

یک پرسپترون چندلایه نشان‌دهنده یک ارتباط غیر خطی بین بردار ورودی‌ها و بردار خروجی‌ها می‌باشد. این کار از طریق اتصال نرون‌های هر گره در لایه‌های قبلی و بعدی انجام می‌شود [8]. خروجی نرون‌ها در ضرایب وزنی ضرب می‌شود و به تابع غیر خطی فعال‌سازی به عنوان ورودی داده می‌شود. در مرحله آموزش، به پرسپترون اطلاعات آموزش داده می‌شود. سپس وزن‌های شبکه به گونه‌ای تنظیم می‌شوند که خطای بین خروجی پیش‌بینی شده و هدف کمینه گردد و یا اینکه تعداد دفعات آموزش به مقدار حداکثر از پیش تعیین شده برسد. سپس به منظور سنجش صحت آموزش انجام شده، یک سری ورودیهای تجربه نشده به شبکه اعمال می‌شود. این ورودیها، باید متفاوت از ورودیهای بکار گرفته شده برای آموزش شبکه باشند. عموماً آموزش شبکه‌های عصبی بسیار پیچیده و یک مسأله بهینه‌سازی با تعداد متغیر زیاد است.

### 3- الگوریتم اجتماع ذرات

الگوریتم اجتماع ذرات یک الگوریتم بهینه‌سازی تقلیدی از رفتارهای جوامع جانوری در پردازش دانش جامعه است. این الگوریتم از دو زمینه ریشه گرفته است. نخست زندگی مصنوعی (مانند دسته پرندهگان، ماهی‌ها) و دوم محاسبات تکاملی [9]. مبنای توسعه الگوریتم PSO این است که، جواب‌های ممکن در یک مسأله بهینه‌سازی به صورت پرندهگان بدون حجم و خصوصیات کیفی در نظر گرفته می‌شوند که از آنان به عنوان ذرات یاد می‌شود، این پرندهگان در یک فضای n-بعدی پرواز کرده و مسیر حرکت خود در فضای جستجو را بر اساس تجارب گذشته خود و همسایگان‌شان تغییر می‌دهند.

در دسته‌ای متشکل از N جزء، موقعیت جزء  $i$ ام تحت اثر یک بردار مکانی n-بعدی مطابق معادله (1) قرار دارد.

$$X_i = (x_{i1}, x_{i2}, \dots, x_{in})^T \in S \quad (1)$$

پس‌انتشار در این سال‌ها برای آموزش شبکه‌های عصبی بسیار رواج یافته است، اما استفاده از این روش در برخی موارد منجر به بروز مشکلاتی می‌گردد. این موانع شامل سرعت پایین همگرایی در روند آموزش و همگرایی زودرس در مینیمم‌های محلی می‌باشد. در [4] چندین الگوریتم برای بهینه‌سازی آموزش به منظور غلبه بر موانع اشاره شده، ارائه شده است. امروزه، کاربرد الگوریتم‌های جستجوگر تکاملی برای آموزش شبکه‌های عصبی به جای روش‌های آموزشی مرسوم، مورد توجه محققان قرار گرفته است. در [5] برای محاسبه ضرایب شبکه عصبی از الگوریتم ژنتیک استفاده شده است. همچنین در [6] از روش Bayesian برای آموزش شبکه عصبی استفاده شده است.

یکی از این الگوریتم‌های جستجوی تکاملی الگوریتم PSO است که می‌توان از آن به عنوان یک جایگزین برای آموزش شبکه‌های عصبی استفاده کرد.

الگوریتم PSO الهام گرفته از رفتار اجتماعی برخی گونه‌های جانوری است که به شکل شبیه‌سازی ساده شده‌ای از این رفتارها نخستین بار در [7] ارائه شد. از جمله مزایای این الگوریتم نسبت به الگوریتم‌های تکامل تدریجی نظیر GA می‌توان به پیاده‌سازی آسان، پارامترهای کم الگوریتم و همچنین سرعت همگرایی بالای آن اشاره کرد. این الگوریتم تاکنون در مسائل مهندسی فراوانی نظیر بهینه‌سازی توابع کنترل سیستم‌های فازی، برنامه‌نویسی NC و زمینه‌های دیگری که الگوریتم‌های GA در آن به کار گرفته شده است، با موفقیت مورد استفاده قرار گرفته است.

در بخش دوم این مقاله، به معرفی اجمالی پرسپترون چند لایه پرداخته شده است. سپس در بخش سوم، توضیحاتی درباره الگوریتم PSO و روند پیاده‌سازی آن ارائه شده است. در بخش چهارم مقاله، بکارگیری الگوریتم PSO در آموزش شبکه‌های عصبی عنوان شده است. در بخش پنجم این مقاله، نتایج حاصل از پیاده‌سازی شبکه عصبی بر روی دو مسأله و مقایسه دو الگوریتم آموزشی پس‌انتشار و PSO آورده شده است.

### 2- پرسپترون چند لایه

مفهوم پرسپترون در سال 1943 توسط McCulloch و Pitts به عنوان یک نورون مصنوعی ارائه شد. در شکل (1)

که در آن S فضای جستجو است.

$\omega$ : وزن اینرسی است.

$C_1$ : پارامتری مثبت به نام پارامتر شناختی است.

$C_2$ : پارامتری مثبت به نام پارامتر اجتماعی است.

استفاده از پارامتر وزن اینرسی باعث می‌شود که مصالحه‌ای بین توانایی اکتشاف سراسری و محلی دسته ایجاد گردد. وزن اینرسی بزرگ، محرکی برای اکتشاف در سراسر ناحیه (حرکت به سمت مناطقی از فضای جستجو که پیشتر تجربه نشده‌اند) بوده، در حالیکه یک وزن کمتر مشوقی برای اکتشاف در نواحی محلی می‌باشد. در واقع وزن کمتر باعث می‌شود که جستجو در مناطقی که در گذشته تجربه شده‌اند، با دقت بیشتری ادامه پیدا نماید. انتخاب اندازه مناسب برای  $\omega$ ، متضمن برقراری تعادل مطلوب بین توانایی اکتشاف محلی و سراسری بوده و در نتیجه باعث افزایش کارایی الگوریتم می‌گردد. نتایج تجربی نشان می‌دهند که انتخاب مقادیر بزرگ برای  $\omega$  در آغاز جستجو، باعث می‌شود که اولویت اکتشافات سراسری نسبت به اکتشافات محلی بالاتر رود، و با کاهش تدریجی  $\omega$  جستجو در فضاهای محلی با جدیت بیشتری دنبال گردد. در نتیجه مقدار  $\omega$  در ابتدای جستجو برابر 1 انتخاب می‌گردد و تدریجاً به صفر میل می‌کند [10].

با توجه به قابلیت الگوریتم PSO در یافتن جواب بهینه سراسری با احتمال بسیار بالا و نرخ همگرایی بالا، از این الگوریتم برای آموزش شبکه‌های عصبی استفاده شده است.

#### 4- بکارگیری PSO در آموزش شبکه‌های عصبی

متغیرهای بهینه‌سازی در آموزش یک شبکه عصبی شامل وزن‌ها و بایاس‌های مربوط به شبکه می‌شود. اگر لایه  $n$ ام از یک شبکه فرضی متشکل از  $R$  ورودی و  $M$  نرون باشد، آنگاه ماتریس وزن‌ها ( $W^n$ ) و بایاس‌های ( $B^n$ ) این لایه را می‌توان با رابطه (8) بصورت زیر نمایش داد:

$$W^n = \begin{bmatrix} (w_1^n)^T \\ (w_2^n)^T \\ \vdots \\ (w_M^n)^T \end{bmatrix}, \quad B^n = \begin{bmatrix} b_1^n \\ b_2^n \\ \vdots \\ b_M^n \end{bmatrix} \quad (8)$$

این جزء همچنین دارای یک بردار سرعت به صورت معادله (2) می‌باشد.

$$V_i = (v_{i1}, v_{i2}, \dots, v_{in})^T \in S \quad (2)$$

بهترین موقعیت قبلی بدست آمده برای جز  $i$ ام، با استفاده از معادله (3) نمایش داده می‌شود.

$$P_i = (p_{i1}, p_{i2}, \dots, p_{in})^T \quad (3)$$

در نهایت موقعیت جدید اجزای دسته با استفاده از معادلات (4) و (5) به دست می‌آید.

$$V_i(t+1) = V_i(t) + c_1 \eta (P_i(t) - X_i(t)) + c_2 \eta (P_g(t) - X_i(t)) \quad (4)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (5)$$

در این معادلات:

$g$ : شاخص به کار رفته برای ذره‌ای است که بهترین موقعیت را دارد.

$t$ : نمایانگر تعداد تکرار می‌باشد.

$c$ : ثابت شتاب.

$r_1$  و  $r_2$ : اعدادی تصادفی در بازه [0 1] می‌باشد.

سرعت هر یک از ذرات دارای مقدار حداکثری است که توسط کاربر تعیین می‌شود. این عامل باعث می‌شود که اندازه سرعت دسته کنترل شود و از انفجار دسته جلوگیری شود. اگر چه الگوریتم PSO قادر است ناحیه جواب بهینه را بسیار سریع پیدا کند، ولی با رسیدن به این ناحیه، سرعت همگرایی آن به شدت کاهش می‌یابد. برای رفع این مشکل، روابط (4) و (5) به صورت زیر تصحیح می‌گردد.

$$V_i(t+1) = \omega V_i(t) + c_1 \eta (P_i(t) - X_i(t)) + c_2 \eta (P_g(t) - X_i(t)) \quad (6)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (7)$$

در معادلات بالا داریم:

که در آن،  $w_m^n = [w_{m,1}^n \ w_{m,2}^n \ \dots \ w_{m,R}^n]^T$  بردار وزنهایی است که نرون  $m$  از لایه  $M$  را به ورودیهای همان لایه مربوط می‌کند. بردار پارامترهای این لایه را نیز می‌توان با رابطه (9) نشان داد:

$$X^n = \begin{bmatrix} w_1^n \\ \vdots \\ w_M^n \\ b_1^n \\ \vdots \\ b_M^n \end{bmatrix} \quad (9)$$

به همین ترتیب، به ازای هر لایه، ماتریس‌های وزن و بایاس بردار پارامترهای مربوطه تعریف می‌شود. با زیر هم قرار دادن بردار پارامترهای همه لایه‌های شبکه، بردار متغیرهای بهینه‌سازی مورد نظر تشکیل می‌شود. نهایتاً برای یک شبکه  $L$  لایه بردار متغیرهای  $X$  را می‌توان از رابطه (10) بدست آورد:

$$X = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^L \end{bmatrix} \quad (10)$$

در واقع این بردار همان بردار موقعیت ذکر شده در رابطه (1) است که مقدار بهینه آرگومانهای آن با استفاده از الگوریتم PSO محاسبه خواهد شد.

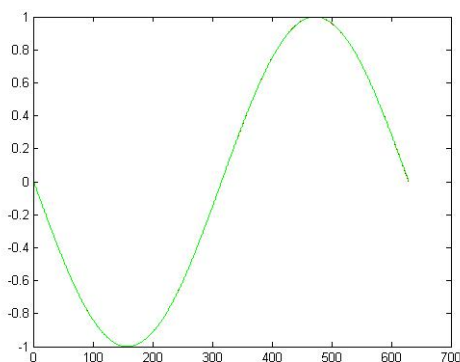
روند کار بدین ترتیب است که ابتدا  $N$  بردار موقعیت  $X_i$  ( $N$ ،  $\dots$ ،  $2$ ،  $1=i$ )، که  $N$  برابر با تعداد اعضای دسته می‌باشد، بصورت تصادفی تولید می‌گردد. جمعیت دسته نیز معمولاً 4 تا 5 برابر تعداد متغیرهای بهینه‌سازی انتخاب می‌شود. شبکه عصبی به ازای پارامترهایی برابر با متغیرهای این بردارها اجرا شده و خطای بدست آمده از هر اجرا به عنوان میزان برازندگی بردار متغیرهای آن شبکه در نظر گرفته می‌شود. در این مرحله بردارهای  $P_i$  و  $P_g$  با توجه به برازندگیهای بدست آمده محاسبه گردیده، و  $N$  بردار موقعیت جدید با استفاده از روابط (6) و (7) تولید می‌شود. این روند آنقدر تکرار می‌شود تا همگرایی نهایی حاصل گردد. منظور از همگرایی نهایی رسیدن به بردار موقعیت بهینه، به نحوی است که خطای آموزش به ازای آن کمینه گردد. در اجرای

الگوریتم PSO، ضرایب  $c_1$  و  $c_2$  برابر با 2 انتخاب شده [10] و ضریب اینرسی ( $w$ ) نیز در ابتدا برابر با 1 در نظر گرفته شده که در طول تکرارها تدریجاً به سمت صفر میل می‌کند.

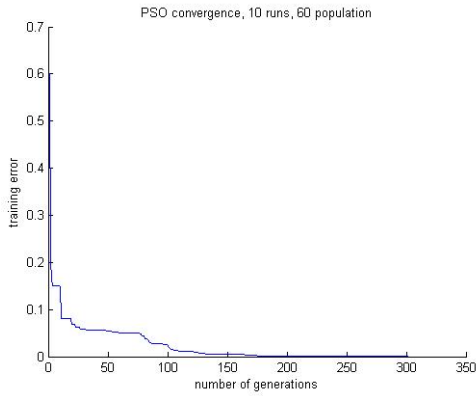
## 5- نتایج شبیه‌سازی و مقایسه

در این مقاله برای نشان دادن کارایی الگوریتم آموزشی جدید دو مثال ارائه شده است و طی آن شبکه عصبی آموزش یافته با الگوریتم آموزشی PSO با شبکه عصبی آموزش یافته با الگوریتم پسانتشار مقایسه می‌شود. در مسأله اول آموزش و تست کارایی شبکه با تابع سینوسی صورت می‌گیرد. سپس از داده‌های مربوط به سرعت باد به عنوان یک مسأله پیچیده‌تر برای آموزش و تست کارایی شبکه به کار گرفته می‌شود. داده‌های باد ارائه شده مربوط به منطقه‌ای در کشور دانمارک می‌باشد. فرکانس نمونه‌برداری از باد 2/5 ثانیه است.

همانطور که از نتایج ارائه شده استنتاج می‌شود، آموزش به روش پسانتشار در مورد توابعی که دارای معادلات ریاضی مشخص هستند یا دارای گسستگی نیستند، به خوبی عمل می‌کند. لذا در شبیه‌سازی انجام شده برای تابع سینوسی، روش پسانتشار نتایج بهتری نسبت به روش PSO نشان داده است. در شکل (2) خروجی آموزش به وسیله روش پسانتشار و در شکل (3) خروجی آموزش به وسیله روش PSO نشان داده شده است. نتایج برای تعداد دوره تناوب بیشتر در شکل‌های بعدی نمایش داده شده است. (شکل 4 و 5)

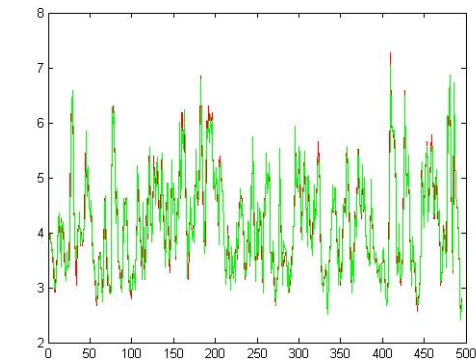


شکل(2): آموزش به روش پسانتشار، تابع سینوسی در یک پروید. همانگونه که انتظار می‌رود، روش پسانتشار کاملاً توانسته، تابع هدف را تخمین بزند. اما خطای الگوریتم PSO در

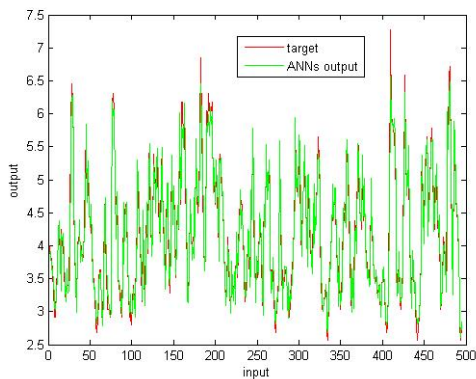


شکل (6): روند همگرایی جواب‌ها در الگوریتم PSO

اما در مورد داده‌های مربوط به باد، همان گونه که انتظار می‌رفت، نتایج مربوط به خروجی پس از اعمال روش PSO بهتر از خروجی حاصل از روش پس‌انتشار است. این امر به خاطر وجود تغییرات ناگهانی با شیب بسیار تند در داده‌های باد است که باعث می‌شود مسأله آموزش داری پیچیدگی خاصی شود. نتایج برای شبیه‌سازی با استفاده از داده‌های باد در شکل (7) و شکل (8) ارائه شده است. همچنین نتایج مربوط به خطاهای دو روش در جدول (1) ارائه شده است.

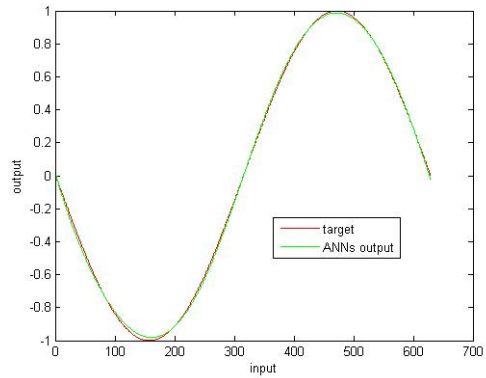


شکل (7): خروجی شبکه با آموزش به روش پس‌انتشار

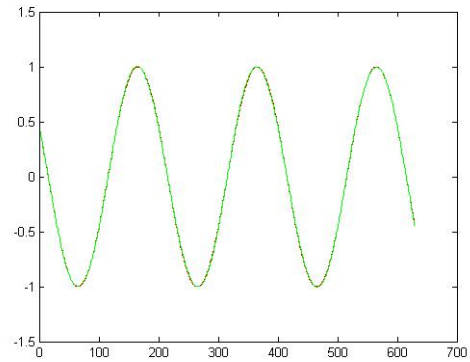


شکل (8): خروجی شبکه با آموزش با الگوریتم PSO

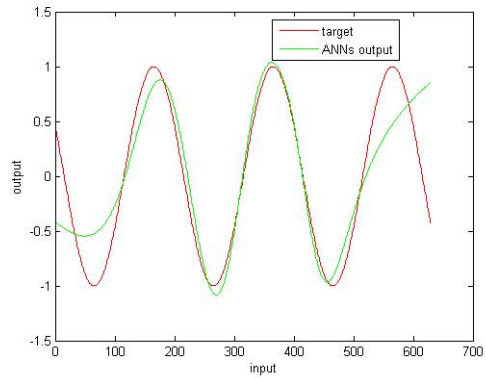
آموزش شبکه‌های عصبی و پیش‌بینی بیشتر از الگوریتم آموزشی پس‌انتشار می‌باشد.



شکل (3): آموزش با الگوریتم PSO، تابع سینوسی در یک پریود.



شکل (4): آموزش به روش پس‌انتشار، تابع سینوسی در چند پریود.



شکل (5): آموزش با الگوریتم PSO، تابع سینوسی در چند پریود.

روند همگرایی در الگوریتم PSO در شکل (6) نشان داده شده است.

Optimal field-scale groundwater remediation using neural networks and the genetic algorithm. *Environmental Science and Technology* 29 (5), 1145–1155.

[6]: Kumar, P., Merchant, S.N., Desai, U.B., 2004. Improving performance in pulse radar detection using Bayesian regularization for neural network training. *Digital Signal Processing* 14 (5), 438–448.

[7]: Kennedy, J., 1997. The Particle swarm: Social adaptation of knowledge. In: *Proceedings of the 1997 International Conference on Evolutionary Computation*, Indianapolis, pp. 303–308.

[8]: Rumelhart, D.E., Hinton, E., Williams, J., 1986. Learning internal representation by error propagation. *Parallel Distributed Processing* 1, 318–362.

[9]: Clerc, M., Kennedy, J., 2002. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *EEE Transactions on Evolutionary Computation* 6 (1), 58–73.

[10]: Konstantinos E. Parsopoulos and Michael N. Vrahatis, "On the Computation of All Global Minimizers Through Particle Swarm Optimization", *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 8, NO. 3, JUNE 2004.

MaxAPE (%)	MAPE	Maximum Error	Mean Error	
29/4899	3/52462	1/20734	0/146675	الگوریتم پس‌انتشار
19/5605	3/42577	0/958059	0/145247	الگوریتم PSO

جدول (1): مقایسه خطاها در دو حالت.

همانطور که در جدول (1) دیده می‌شود خطای حاصل از الگوریتم PSO بهتر از الگوریتم آموزشی پس‌انتشار می‌باشد.

نتایج ارائه شده، میانگین ده بار اجرای برنامه می‌باشد.

## 6- نتیجه‌گیری

در این مقاله روشی جدید برای آموزش شبکه‌های عصبی با استفاده از الگوریتم PSO ارائه شده است. الگوریتم ارائه شده در مورد توابعی که دارای شیب تند و یا تغییرات ناگهانی هستند، به خوبی می‌تواند تابع هدف را تخمین بزند، در حالی که روش پس‌انتشار در مورد این گونه توابع عملکرد رضایت‌بخشی ندارد. البته بکارگیری الگوریتم‌های تکاملی با توجه به جستجوی تصادفی آنها در فضای جواب مسأله، خودبخود فرآیند یادگیری در شبکه‌های عصبی را با شرایط تصادفی مواجه می‌کند. اما لازم به ذکر است که الگوریتم‌های تکاملی از جمله PSO در حال تکامل هستند و استفاده از انواع ترکیبی آنها می‌تواند نتایج بهتری به همراه داشته باشد.

## 7- مراجع

[1]: Chau, K.W., Cheng, C.T., 2002. Real-time prediction of water stage with artificial neural network approach. *Lecture Notes in Artificial Intelligence* 2557, 715.

[2]: Rumelhart, D.E., Hinton, E., Williams, J., 1986. Learning internal representation by error propagation. *Parallel Distributed Processing* 1, 318–362.

[3]: Bazartseren, B., Hildebrandt, G., Holz, K.-P., 2003. Short-term water level prediction using neural networks and neuro-fuzzy approach. *neurocomputing* 55 (3–4), 439–450.

[4]: Haykin, S., 1999. *Neural Networks, A Comprehensive Foundation*. Prentice Hall, Upper Saddle River.

[5]: Rogers, L.L., Dowla, F.U., Johnson, V.M., 1995.