

# ارائه روشی نو مبتنی بر اتوماتان های یادگیر برای تطبیق پارامترهای الگوریتم یادگیری<sup>1</sup> VLR<sup>2</sup> برای آموزش شبکه های عصبی<sup>2</sup>

محمد باقر منهاج  
دانشیار

دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر

بهبود مشعوفی  
دانشجوی دکتری

دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر

محمد رضا میبدی  
دانشیار

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

سید احمد معتمدی  
دانشیار

دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر

## چکیده

الگوریتم BP<sup>3</sup> برای طبق و سیعی از کاربردها مورد استفاده قرار گرفته است. یکی از مشکلات عدمه این الگوریتم، پائین بودن نرخ همگرایی آن می باشد. دلیل عدمه، اشباع شدن تابع فعالیت می باشد. بمحض اینکه خروجی یک واحد در ناحیه اشباع قرار می گیرد، گردایان متناظر دارای مقدار کوچکی خواهد بود. اگر از نرخ یادگیری ثابت کوچکی استفاده شود این مسئله می تواند باعث پائین آمدن سرعت همگرایی شود. یک راه حل برای افزایش سرعت همگرایی، استفاده از نرخ یادگیری بزرگ می باشد. متاسفانه این روش در مواقعي که در نقطه ای از سطح خطای قرار داریم که شبکه تندی دارد بسیار خطرناک بوده و می تواند باعث نوسان و واگرایی شود. لذا نیاز به الگوریتم هایی داریم که بتوانند بطور بیان نرخ همگرایی را بر اساس تغییرات مقادیر گردایان تغییر دهد. در این مقاله روش های مختلف تغییر دینامیک نرخ یادگیری مورد بررسی قرار گرفته است. الگوریتم نرخ یادگیری متغیر (VLR) و الگوریتم های تطبیق نرخ یادگیری مبتنی بر اتوماتان های یادگیر توضیح داده و بر روی مسائل مختلف پیاده و با هم مقایسه شده اند. نظر به اینکه در روش VLR پارامترهای مورد استفاده در این روش تاثیر بسزایی در عملکرد آن دارد لذا در این مقاله از اتوماتان یادگیر برای تنظیم پارامترهای آن استفاده کرده ایم. در الگوریتم جدید که تحت عنوان الگوریتم نرخ یادگیری متغیر تطبیقی<sup>4</sup> (AVLR) نامگذاری کرده ایم، پارامترهای الگوریتم VLR بطور دینامیک توسط اتوماتان یادگیر بر اساس تغییرات خطای تغییر داده می شود. نتایج مشابه سازی ها بر روی مسائل مختلف نشان می دهد الگوریتم پیشنهادی نسبت به روش های دیگر از سرعت همگرایی بالایی برخوردار می باشد.

## کلمات کلیدی

شبکه های عصبی چند لایه، الگوریتم پس انتشار خطای، نرخ یادگیری متغیر، اتوماتان های یادگیر، نرخ یادگیری متغیر.

## Introducing A Novel Learning Automata Based Method For Adapting VLR Learning Algorithm Parameters For Learning MLP Neural Networks

B. Mashoufi  
Ph. D. Student  
Electrical Engineering Department,  
Amirkabir University of Technology

M. B. Menhaj  
Associate Professor  
Electrical Engineering Department,  
Amirkabir University of Technology

S.A. Motamed  
Associate Professor  
Electrical Engineering Department,  
Amirkabir University of Technology

M. R. Meybodi  
Associate Professor Computer  
Engineering Department, Amirkabir  
University of Technology

## Abstract

BP algorithm has been used for wide range of applications. One of the most important limitations of this algorithm, is the low rate of convergence. The important reason behind this, is the saturation property of its activation functions. Once the output of a unit lies in the saturation area, the corresponding decent gradient would take a very small value. This will result in very little progress in the weight adjustment, if one takes a fixed small learning rate parameter. To avoid this undesired phenomenon, one may consider a relative large learning rate. Unfortunately this would be dangerous, because it may take the algorithm diverges, especially when the weight adjustment happens to fall into the surface regions with a large steepness. So, we require algorithms capable of tuning dynamically learning rate according to changes of gradient values. In this paper, different methods of dynamic changing of learning rate has been considered. Variable Learning Rate (VLR) algorithm and learning automata based learning rate adaptation algorithms are considered and compared with each other. Because the VLR parameters have important influence in its performance, so we use learning automata for adjusting them. In the proposed algorithm called Adaptive Variable Learning Rate (AVLR) algorithm, VLR parameters are adapted dynamically by learning automata according to error changes. Simulation results on various problems highlight better the merit of the proposed AVLR.

## Keyword

Multilayer Neural Network, Backpropagation, Variable Learning Rate, Learning Automata, Adaptive Variable Learning Rate

## مقدمه

کافی کوچک باشند انجام می‌گیرد. در آغاز به تمامی وزن‌ها مقادیر تصادفی کوچک داده می‌شود.

الگوریتم BP برای رنج وسیعی از کاربردها مورد استفاده قرار گرفته است. متأسفانه نرخ همگرایی BP مخصوصاً برای شبکه‌های با بیش از یک لایه مخفی بسیار پایین می‌باشد. دلیل اصلی پائین بودن نرخ همگرایی BP اشباع شدن تابع فعالیت مورد استفاده در واحدهای مخفی و خروجی می‌باشد. بمحض اینکه خروجی یک واحد در ناحیه اشباع قرار می‌گیرد، گرادیان متناظر اگر هم خطای خروجی خیلی بزرگ باشد دارای مقدار کوچکی خواهد بود. اگر از نرخ یادگیری ثابت کوچکی استفاده شود این مسئله می‌تواند باعث نرخ پائین همگرایی شود.

بمنظور جلوگیری از این پدیده نامطلوب یک راه حل استفاده از نرخ یادگیری بزرگ می‌باشد. این راه حل بویژه در موقعي که در نقطه‌ای از سطح خطا قرار داریم که شبکه تندی دارد بسیار خطرناک بوده و می‌تواند باعث واگرایی شود.

تعدادی از محققین سعی کرده‌اند سطوح خطا شبکه‌های MLP با الگوریتم یادگیری BP را توصیف کنند [۵]-[۲]. نشان داده شده است موقعیکه شکل سطوح خطا از سطوح خطا درجه دوم خیلی فاصله داشته باشد در این صورت شامل مناطق مسطح و مناطق با شبکه تند زیادی خواهد بود. در این

مسئله آموزش پس انتشار خطا (BP) به روش دسته‌ای<sup>۱</sup> را در نظر می‌گیریم. فرض کنیم که تعداد  $p$  الگوی آموزش در دسترس می‌باشد. هدف انتخاب وزن‌های شبکه است بطوریکه مجموع مربيعات خطای زیر حداقل گردد.

$$J(W) = \frac{1}{p \cdot s_M} \sum_{s=1}^p \sum_{i=1}^{s_M} [t_{s,i} - a_{s,i}^M]^2$$

که در آن  $t_{s,i}$  و  $a_{s,i}^M$  به ترتیب  $s$  امین خروجی مطلوب و واقعی متناظر با  $s$  امین الگوی آموزش می‌باشد.  $W$  بردار وزن‌های شبکه و  $s_M$  تعداد واحدهای خروجی می‌باشد. به منظور دستیابی به بردار وزن بهینه به روش تکراری، یک الگوریتم از نوع نزولی<sup>۲</sup> برای او لین بار توسط Werbos در سال ۱۹۷۴ ارائه گردید. سپس در سال ۱۹۸۲ توسط Parker و در سال ۱۹۸۶ توسط Rumelhart توسعه داده شد. برای هر الگوی آموزش، این الگوریتم شامل دو مرحله می‌باشد. در مرحله اول، خروجی‌های واقعی شبکه بطرف جلو از لایه ورودی بطرف لایه خروجی حساب می‌شود. در مرحله دوم، گرادیان حساب شده و وزن‌ها تنظیم می‌شوند. این روند بطور تکراری برای هر الگوی آموزش تا زمانیکه تمامی سیکنال‌های خطا بین خروجی‌های مطلوب و واقعی به اندازه