

روشی سریع برای تعیین تعداد واحدهای لایه مخفی شبکه های چند لایه

Fast Method for Determining the Number of Hidden Units in Feedforward Neural Networks

محمد رضا میدی
عضو هیئت علمی

حمید یگی
دانشجو

دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیرکبیر
تهران - ایران

(Email : BEIGY@CE.AKU.AC.IR)

چکیده

یکی از مشکلات طراحی شبکه های عصبی، مشخص نمودن توپولوژی شبکه میباشد. و کارایی بسیاری از الگوریتم ها، از جمله الگوریتم انتشار خطا به عقب^۱، به توپولوژی شبکه وابسته میباشد. معمولاً تعداد واحدهای لایه مخفی در شبکه های چند لایه با سعی و خطا تعیین میشود. در این مقاله روشی سریع برای تعیین تعداد واحدهای لایه مخفی و آموزش این شبکه پیشنهاد شده است. در روش پیشنهادی ابتدا یک شبکه کوچک در نظر گرفته میشود و بتدریج واحدهای مخفی آموزش داده شده و به شبکه اضافه میشوند و پس از اضافه شدن هر واحد شبکه باز آموزی میشود تا خطای شبکه کاهش یابد. یکی از مزایای این روش، همزمانی آموزش شبکه و تعیین تعداد واحدهای لایه مخفی می باشد. مزیت دیگر آموزش یک واحد در هر لحظه می باشد لذا سرعت آموزش بسیار بالاتر از روش انتشار خطا به عقب خواهد بود. شبیه سازی های انجام شده توانایی های این روش را نشان میدهد.

کلمات کلیدی : تعیین تعداد واحدهای لایه مخفی ، آموزش سریع ، آموزش شبکه های چند لایه

ABSTRACT

One of the difficulties in design of neural network is determination of the topology of network and performance of many neural network learning algorithm is dependent to the topology of network. Usually the number of hidden units is determined by trial and error. In this paper, we propose a fast method for determination of hidden units in the feedforwards neural networks. In this method a small network is considered and hidden units are added one at a time to the network and the network is retrained inorder to decrease the network error. One advantage of this method is simultaneous determination of number of hidden units and weights of the network. Other advantage is its speed. Simulation results shows the feasibility of the our propose method.

KEY WORDS: Determining the Number of Hidden Units, Fast Learning, Multi-Layer Neural Network Learning

1-Error Back Propagation

۱- مقدمه

شبکه عصبی از اتصال تعدادی المان پردازش کننده تشکیل شده است که برخور دی متفاوت با محاسبه دارد. از نقطه نظر بیولوژی، شبکه عصبی مشخصاتی متناظر با مشخصات مغز انسان را عرضه می‌کند اما در مقیاسی بسیار کوچکتر. اما از نقطه نظر تئوری سیستم شبکه عصبی میتواند مدل‌های بسیار غیرخطی را بدون مدل ریاضی، مدلسازی نماید. شبکه‌های عصبی با تصحیح رفتارشان در مواجه با پاسخ محیط میتوانند یاد بگیرند.

الگوریتم‌های بسیاری برای آموزش شبکه‌های عصبی پیشنهاد شده است که هر کدام دارای مزایا و معایبی می‌باشد. یکی از الگوریتم‌های آموزش شبکه عصبی، الگوریتم انتشار خطابه عقب می‌باشد^[۱]. علیرغم کاربرد های موفق این الگوریتم، کارایی این الگوریتم به میزان زیادی به تبیولوژی شبکه مانند تعداد واحد‌های لایه مخفی و اتصالات آن وابسته است. یکی از مشکلات الگوریتم انتشار خطابه عقب، ناتوانی این الگوریتم در تعیین تبیولوژی شبکه می‌باشد. در بسیاری از روش‌ها، ابتدا یک شبکه بزرگ با اتصالات کامل در نظر گرفته می‌شود و پس از آموزش شبکه با الگوریتم انتشار خطابه عقب، اتصالات واحد‌هایی که دارای اهمیت کمتری در عملکرد شبکه می‌باشند هرس می‌شوند ولذا اندازه شبکه کوچک می‌شود. با این روش سرعت بازیابی الگوهای زیاد می‌شود اما سرعت آموزش شبکه زیاد نمی‌شود. به همین دلیل روش‌های زیادی برای ازبین بردن مشکلات این الگوریتم پیشنهاد شده است. بجای اینکه یک شبکه بزرگ بازیم و سپس آنرا هرس نماییم می‌توانیم از یک شبکه کوچک شروع کرده و آنرا رشد دهیم تا به اندازه مورد نظر برسد. برای تعیین تبیولوژی شبکه روش‌های مختلفی پیشنهاد شده است. در اکثر این روش‌ها هدف ساختن شبکه ای است که توانایی مدلسازی یک تابع بازی بری که دارای N ورودی و یک خروجی را داشته باشد. تابع فعالیت واحد‌ها در این روش‌ها تابع بله^۱ می‌باشند.

Rujan و Golea و Marchand [۲] الگوریتمی برای ساختن شبکه چند لایه با یک لایه مخفی را پیشنهاد نمودند که در هر لحظه یک واحد به شبکه اضافه می‌شود. واحد‌های مخفی طوری به شبکه اضافه می‌شوند تا بتوانند یک یا بیشتر الگوی ورودی را از هم جدا نمایند. با اضافه نمودن هر واحد یک یا بیشتر الگو را از هم جدا می‌شوند و این کار تا زمانی ادامه داده می‌شود که تمام الگوهای باقیمانده دارای خروجی یکسانی باشند و با یک واحد مخفی اضافی می‌توانند از هم جدا شوند. هر واحد توسط قانون پرسپترون آموزش داده می‌شود.

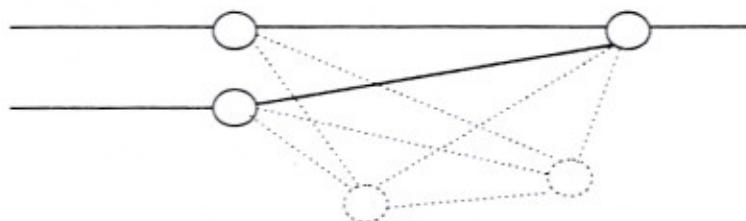
Frean [۳] الگوریتم Upstart را برای تولید ساختار شبکه عصبی پیشنهاد کرد. این روش فقط برای واحد های روشن/خاموش^۲ مناسب می‌باشد و ابتدا شبکه با یک واحد خروجی که تمام ورودیها به آن متصل می‌شوندو بدون واحد‌های مخفی آموزش داده می‌شود. اگر جواب صحیح نباشد و واحد مخفی در لایه ماقبل آخر اضافه می‌شود و آموزش تکرار می‌شود. در صورت غلط بودن جواب چهار واحد مخفی در لایه قبل از لایه مخفی قبل قرار داده می‌شود و این کار آنقدر ادامه پیدا می‌کند تاخطابه حد قابل قبولی برسد.

1- Hard Limitter

2- On / Off Units

Nadal و Sirat [۴] بطور مستقل راه یکسانی را برای تقسیم فضای ورودی پیشنهاد نمودند. در این ساختار هر واحد که برای تصحیح عملکرد واحد دیگری بکاربرده می شود بصورت درخت تصمیم گیری دودبی در شبکه قرار می گیرد و توسط الگوریتم مشابه پیمایش درخت های دودبی آموزش می بیند. Nadal و Mezard [۵] الگوریتم موزاییک را برای برای تولید شبکه عصبی چند لایه استفاده نمودند. در این روش در اولین لایه مخفی تعداد واحد ها زیاد است که بتدریج در لایه های مخفی کاهش پیدا میکند و در نهایت در لایه خروجی به یک واحد میرسد.

اسکات فالمن [۶] شبکه Cascade Correlation پیشنهاد نمود. این شبکه فقط از تعدادی واحد ورودی و خروجی تشکیل شده است. واحد های لایه ورودی بطور مستقیم به واحد های لایه خروجی متصل می شوند. واحد های لایه مخفی بصورت یک لایه سازمان نیافته اند بلکه بصورت پشت سر هم قرار می گیرند. هر واحد مخفی ورودی هایش را از خروجی های واحد های ورودی و واحد های مخفی قبل از خود دریافت می کند. یک نمونه از این شبکه با دو واحد ورودی و دو واحد مخفی (که بصورت خط چین نمایش داده شده است) و یک واحد خروجی در شکل زیر نشان داده شده است.



شکل ۱: شکل یک شبکه Cascade Correlation

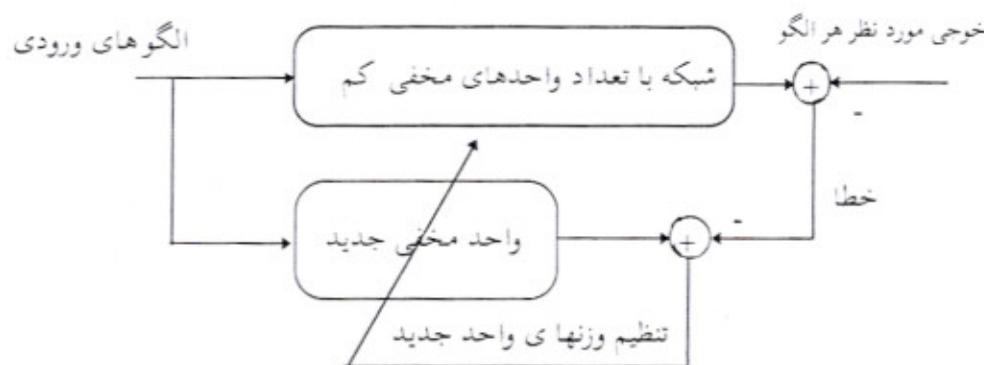
آموزش شبکه در ابتدا فقط با واحد های ورودی و خروجی و بدون واحد های مخفی شروع می شود. الگوریتمی از نوع "نزول در خلاف جهت گرادیان"^۱ برای آموزش شبکه استفاده می شود. این الگوریتم بدلیل اینکه نگاشت خطی است بسیار سریع کار می کند. واحد های مخفی، دانه دانه په شبکه اضافه می شوند و پس از اضافه کردن یک واحد مخفی به شبکه، شبکه بازآموزی می گردد. آموزش شبکه در دو مرحله صورت می پذیرد. در مرحله اول فقط واحد مخفی جدید قبل از اینکه به شبکه اضافه شود آموزش داده می شود تا حتی المقدور بتواند خطای شبکه را در مرحله فعلی جبران نماید. در مرحله دوم این واحد در شبکه نصب شده اما وزنهای آن ثابت در نظر گرفته می شود و آموزش تنها در لایه خروجی صورت می پذیرد و اتصالاتی که به واحد خروجی متصل می شوند (که شامل این اتصال جدید نیز می باشد) باز آموزی می گردد تا خطای شبکه کم شود. بدلیل اینکه واحد جدید خطای شبکه را در مرحله قبل محاسبه می کند بنابراین خطای نهایی شبکه کاهش پیدا می کند. این کار آنقدر ادامه پیدا می کند تا خطای آن حد مطلوبی برسد. در این الگوریتم مسئله اصلی آموزش واحد جدید بوده که وظیفه آن جبران خطای شبکه می باشد. این کار با حداکثر نمودن کوواریانس خروجی واحد مخفی نسبت به خطای شبکه انجام می شود.

سپس از روش نزول در خلاف جهت گرادیان برای پیدا کردن وزن های بهینه استفاده می گردد زیرا می خواهیم مقدار کوواریانس را حداقل نماییم. متاسفانه در تابع کوواریانس تعدادی حداقل محلی وجود دارد و در نتیجه لازم است که یک مجموعه از واحدهای لایه مخفی، بجای یک واحد تنها آموزش بینند این واحدها با وزن های متفاوتی شروع بکار می کند و در انتهای آموزش از واحدهای فوق، واحدی را که دارای بیشترین مقدار کوواریانس می باشد به شبکه اضافه می شود. این شبکه بسیار سریع آموزش می بیند اما مشکل این شبکه نولیدنگاشت باله های تبیین ورودی و خروجی می باشد. یکی از دلایل تبیین نگاشت، اثر مستقیم ورودی ها روی خروجی می باشد. در این مقاله روشی برای تعیین تعداد واحدهای لایه مخفی پیشنهاد شده است که مشکلات فوق را ندارد.

۲- روش پیشنهادی

روش پیشنهاد شده برایه رشد دادن یک شبکه کوچک و رسیدن به شبکه مورد نظر می باشد. طبق این روش، یک شبکه سه لایه بصورت زیر ساخته می شود در مرحله اول یک واحد در لایه مخفی قرار داده می شود و سپس این شبکه برای الگوهای ورودی، توسط الگوریتم انتشار خطای عقب آموزش داده می شود. شرط پایان می تواند تعداد تکرار ارائه الگوها به شبکه باشد. در حین آموزش شبکه، خطای شبکه برای هر الگو ذخیره می گردد و در مرحله دوم الگوریتم از آن استفاده می شود. در پایان مرحله اول، اگر خطای خروجی شبکه به میزان کافی کم نشده باشد وارد مرحله دوم الگوریتم شده و در این مرحله یک واحد مخفی دیگر به شبکه اضافه می کنیم. فرض می شود که وزن اتصال بین واحد مخفی جدید و واحد خروجی یک باشد.

سپس واحد مخفی جدید را برای الگوهای ورودی آموزش می دهیم در حین آموزش این واحد مخفی، خروجی مورد نظر الگوهای خطا شبکه برای آن الگومی باشد. در واقع این واحد جدید یاد میگیرد تا خطای شبکه با واحد های قبلی جبران نماید. شرط پایان نیز همانند حالت قبل می تواند تعداد تکرار ارائه الگوها به شبکه باشد. مدل آموزش برای واحد جدید بصورت زیر می باشد.



شکل ۲: مدل آموزش برای واحد مخفی جدید

پس از آموزش واحد مخفی جدید آن واحد را در شبکه نصب می کنیم و تنها اتصالاتی را که به واحد خروجی متصل می شوند بهنگام می شود. اگر خطای شبکه از حد قابل قبولی بایین تر نباشد مجدداً یک واحد مخفی دیگر به شبکه اضافه می کنیم و مرحله دوم را تکرار می نماییم. الگوریتم زمانی خاتمه می یابد که خطای شبکه برای اضافه شدن دو واحد مخفی آخر تغییر زیادی ننماید یعنی اضافه شدن آخرین واحد مخفی بهبود قابل ملاحظه ای در عملکرد شبکه ایجاد نمی کند. اگر خطای شبکه در پایان الگوریتم در حد انتظار نباشد می توانیم وزن های فعلی شبکه را بعنوان وزن اولیه برای الگوریتم انتشار خطای عقب در نظر گرفته و وزن های شبکه را با این روش بصورت دقیق تر تنظیم نمود. الگوریتم آموزش توسط روش پیشنهادی در شکل زیر نشان داده شده است.

Procedure LearningMethod

```

Set no. of hidden units in the neural network to 1.
repeat
    for all training pattern (X,T) in the set of patterns do
        Call FeedForwad
        Call SaveErrorForEachPattern (E)
        Call ComputeGradient
    End for
    Call UpdateWeights
until Some termination condition is satisified.
If Error is not below of specific value Then
    repeat
        get a new hidden unit
        repeat
            for all training pattern (X, E) do
                Train the new hidden unit
            End for
        until Some termination condition is satisified.
        Install this new hidden unit in the neural network.
        repeat
            for all training pattern (X,T) in the set of patterns do
                Train the output node
            End for
        until Some termination condition is satisfied.
    until Addition of the new hidden unit do not enhancement the error of network
// This part is optional
If Error is not below of specific value Then
    Call BackPropagation
End if
End if
End procedure

```

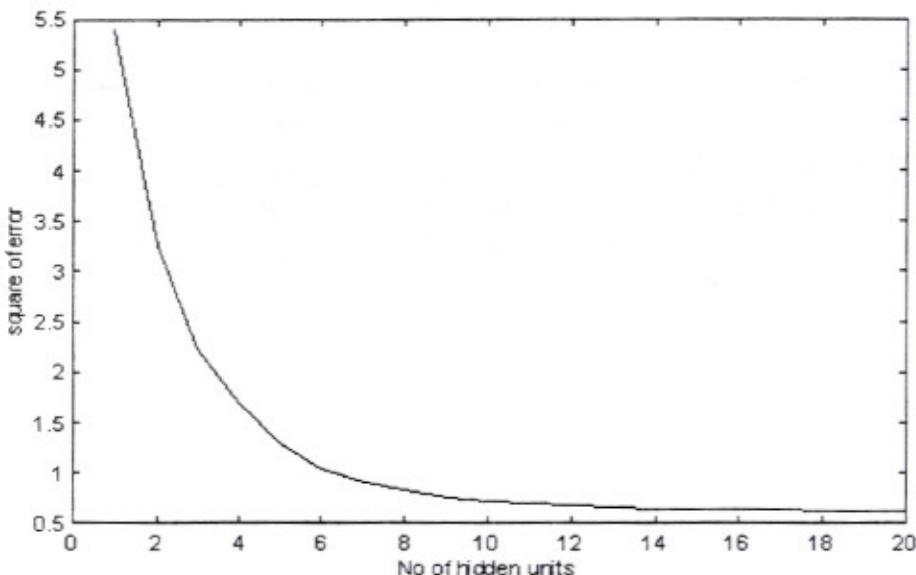
شکل ۲ : الگوریتم آموزش شبکه عصبی

با استفاده از این روش نه تنها تعداد واحد های لابه مخفی را می توان بدلست آورد بلکه افزایش بسیار چشمگیری در سرعت نیز حاصل می شود.

۳- نتایج شبیه سازی

نفیب تابع $F(x) = \text{Sin}(x)$: هدف نفیب تابع بیان شده در یک دوره تناوب می باشد و بیست نمونه آموزشی در فاصله $[0, 2\pi]$ بصورت تصادفی و با توزیع یکنواخت انتخاب شده اند. این شبکه دارای یک واحد ورودی و یک واحد خروجی می باشد.

مثال فوق توسط الگوریتم بیان شده آموزش داده شد و منحنی مریع خطای شبکه بر اساس تعداد واحد های مخفی در شکل ۴ نشان داده شده است.



شکل ۴ : مریع خطای شبکه بر اساس تعداد واحد های مخفی

مطابق شکل ۴ ، زمانی که تعداد واحد های مخفی از ده واحد بیشتر می شود در خطای شبکه کاهش چندانی مشاهده نمی شود لذا تعداد واحد های مخفی حدود ده واحد برای شبکه مورد نظر کافی می باشد. این الگوریتم برای مسائل متنوعی پیاده سازی شده و نتایج مطلوبی بدست آمده است [۷].

۴- نتیجه گیری

در این مقاله روشی سریع برای تعیین تعداد واحد های مخفی شبکه عصبی پیشنهاد شده است. در این روش بطور همزمان آموزش شبکه و تعیین واحد های مخفی انجام می شود. این روش دارای مزایای بسیاری می باشد که عبارتند از: سرعت یادگیری بسیار بیشتر از الگوریتم یادگیری انتشار خطأ به عقب می باشد زیرا در هر لحظه فقط یک واحد آموزش می بیند. مزیت دیگر آن کم بودن احتمال گرفتار شدن در حداقل های محلی می باشد زیرا ابتدا

در یک فضای بسیار کوچک به جستجوی نقطه حداقل عمومی پرداخته می شود. حداقل تعداد واحد های لازم برای آموزش نیز از مزایای دیگر این روش می باشد.

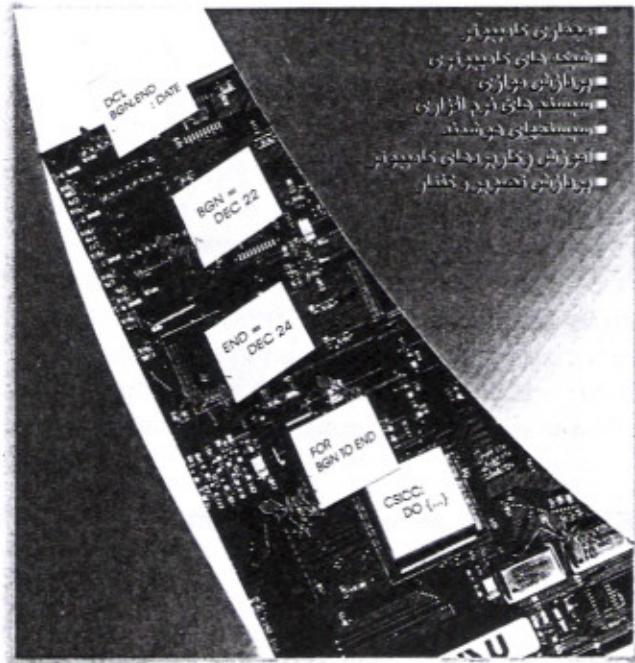
۵ - مراجع

- [1] Rumelhart, D. E., Hinton, G. E., and Williams, R. J.,(1986), "Learning internal representations by error propagation", In Parallel distributed processing, Cambridge, MA: MIT Press.
- [2] M. Marchand, M. Golea, and R. Rujan ,(1990), "A Convergence Theorem for Sequential Learning in Two-Layer Perceptrons", Europhysics Letters 11, pp. 487-492.
- [3] M. Frean, (1990), "The Upstart: A Method for Constructing and Training Feedforward Neural Networks", Neural Computation, pp. 198-209.
- [4] J. A. Sirat and J. P. Nadal, (1990), "Neural Trees: A New Tool for Classification", Preprint, Laboratories d'Electronique, Philips, Limeil Brevannes, France.
- [5] M. Mezard and J. P. Nadal, (1989), "Learning in Feedforward Neural Networks: The Tiling Algorithm", Journal of Physics, pp. 1285-1296.
- [6] S. E. Fahlman and C. Lebier, (1990), "The Cascade -Correlation Learning Architecture", Advances in Neural Information Processing System II, pp. 524-532.
- [7] H. Beigy and M. R. Meybodi,(1997), "A Fast Method for Determining the Number of Hidden Units in Feedforward Neural Networks", Technical Report, Computer Eng. Department, Amirkabir university of Technology, Tehran, Iran.



PROCEEDING OF THE THIRD INTERNATIONAL ANNUAL COMPUTER SOCIETY OF IRAN COMPUTER CONFERENCE

CSICC'97



COMPUTER CONFERENCE

Iranian Society of Science And Technology Engineering Department

23-25 Dec. 1997