

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 بیگی
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 واحد های لایه مخفی در شبکه های چند لایه با سعی و خطا تعیین میشود. در این مقاله روشی سریع برای تعیین تعداد واحد های لایه مخفی و آموزش این شبکه پیشنهاد شده است. در روش پیشنهادی ابتدا یک شبکه کوچک در نظر گرفته میشود و بتدریج واحد های مخفی آموزش داده شده و به شبکه اضافه میشوند و پس از اضافه شدن هر واحد شبکه بازآموزی میشود تا خطای شبکه کاهش یابد. یکی از مزایای این روش ، همزمانی آموزش شبکه و تعیین تعداد واحد های 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 واحد های 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 in order 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 برخوردی 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 مدل های بسیار غیر خطی را بدون مدل ریاضی، مدل سازی نماید. شبکه های عصبی با تصحیح رفتارشان در مواجه با پاسخ محیط میتوانند یاد بگیرند.

الگوریتم های بسیاری برای آموزش شبکه های عصبی پیشنهاد شده است که هر کدام دارای مزایا و معایبی می باشد. یکی از الگوریتم های آموزش شبکه عصبی، الگوریتم انتشار خطا به عقب می باشد [۱]. علیرغم کاربرد های موفق این الگوریتم، کارایی این الگوریتم به میزان زیادی به توپولوژی شبکه مانند تعداد واحد های لایه مخفی و اتصالات آن وابسته است. یکی از مشکلات الگوریتم انتشار خطا به عقب، ناتوانی این الگوریتم در تعیین توپولوژی شبکه میباشد. در بسیاری از روش ها، ابتدا یک شبکه بزرگ با اتصالات کامل در نظر گرفته میشود و پس از آموزش شبکه با الگوریتم انتشار خطا به عقب، اتصالات و واحد هایی که دارای اهمیت کمتری در عملکرد شبکه می باشند هرس می شوند و لذا اندازه شبکه کوچک میشود. با این روش سرعت بازیابی الگوها زیاد می شود اما سرعت آموزش شبکه زیاد نمی شود. به همین دلیل روش های زیادی برای از بین بردن مشکلات این الگوریتم پیشنهاد شده است. بجای اینکه یک شبکه بزرگ بسازیم و سپس آنرا هرس نماییم می توانیم از یک شبکه کوچک شروع کرده و آنرا رشد دهیم تا به اندازه مورد نظر برسد. برای تعیین توپولوژی شبکه روش های مختلفی پیشنهاد شده است. در اکثر این روشها هدف ساختن شبکه ای است که توانایی مدل سازی یک تابع باینری که دارای N ورودی و یک خروجی را داشته باشد. تابع فعالیت واحد ها در این روشها تابع پله^۱ می باشند.

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 می شود. واحد های مخفی طوری به شبکه اضافه می شوند 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 می شوند و این کار تا زمانی ادامه داده می شود که تمام الگوهای باقیمانده دارای خروجی یکسانی باشند و با یک واحد مخفی اضافی می توانند از هم جدا شوند. هر واحد توسط قانون پرسپترون آموزش داده می شود.

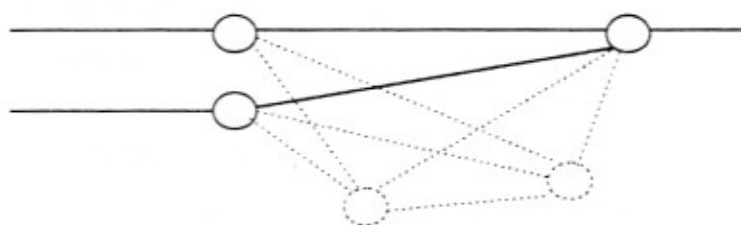
Frean [۳] الگوریتم Upstart را برای تولید ساختار شبکه عصبی پیشنهاد کرد. این روش فقط برای واحد های روشن/خاموش^۲ مناسب می باشد و ابتدا شبکه با یک واحد خروجی که تمام ورودیها به آن متصل می شوند و بدون واحدهای مخفی آموزش داده می شود. اگر جواب صحیح نباشد دو واحد مخفی در لایه ماقبل آخر اضافه می شود و آموزش تکرار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

1- Hard Limiter

2- On / Off Units

Sirat و Nadal [۴] بطور مستقل راه یکسانی را برای تقسیم فضای ورودی پیشنهاد نمودند. در این ساختار هر واحد که برای تصحیح عملکرد واحد دیگری بکار برده می شود بصورت درخت تصمیم گیری دودویی در شبکه قرار می گیرد و توسط الگوریتمی مشابه پیمایش درخت های دودویی آموزش می بیند. Nadal و Mezard [۵] الگوریتم موزایک را 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 میکنند 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. واحدهای لایه ورودی بطور مستقیم به واحدهای لایه خروجی متصل می شوند. واحدهای 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 واحدهای ورودی و واحدهای 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 واحدهای ورودی و خروجی وبدون واحدهای مخفی شروع می شود. الگوریتمی از نوع "نزول در خلاف جهت گرادیان" برای آموزش شبکه استفاده می شود. این الگوریتم بدلیل اینکه نگاشت خطی است بسیار سریع کار می کند. واحدهای مخفی، دانه دانه به 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 وزنه های 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) بازآموزی 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 خطا به 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.

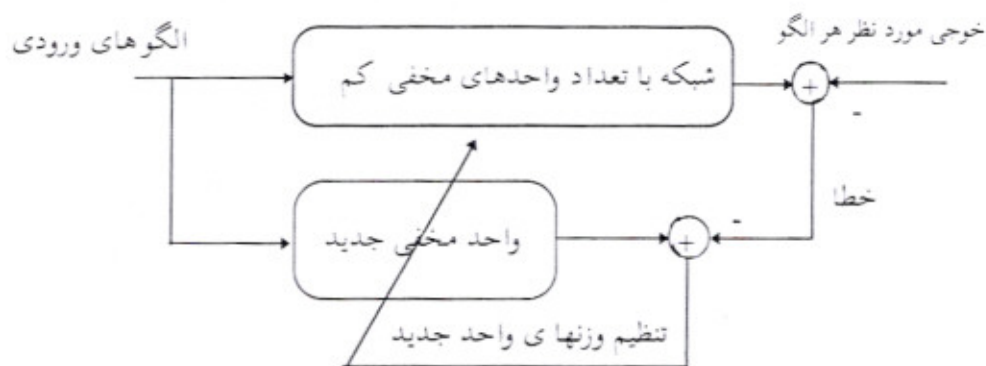
1- Gradient Descent Algorithm

سپس از روش نزول در خلاف جهت گرادینان برای پیدا کردن وزن های بهینه استفاده می گردد زیرا می خواهیم مقدار کوواریانس را حداکثر نماییم. متأسفانه 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 ببینند 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 تولیدنگاشت با لبه های تیز بین ورودی و خروجی می باشد. یکی از دلایل تیز بودن 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 بر پایه 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 خطا به 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 الگوها، خطای شبکه برای آن الگو می باشد. در واقع این واحد جدید یاد میگیرد تا خطای شبکه با واحدهای 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 پایین 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 خطا به 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 FeedForward

Call SaveErrorForEachPattern (E)

Call ComputeGradient

End for

Call UpdateWeights

until Some termination condition is satisfied.

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 satisfied.

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

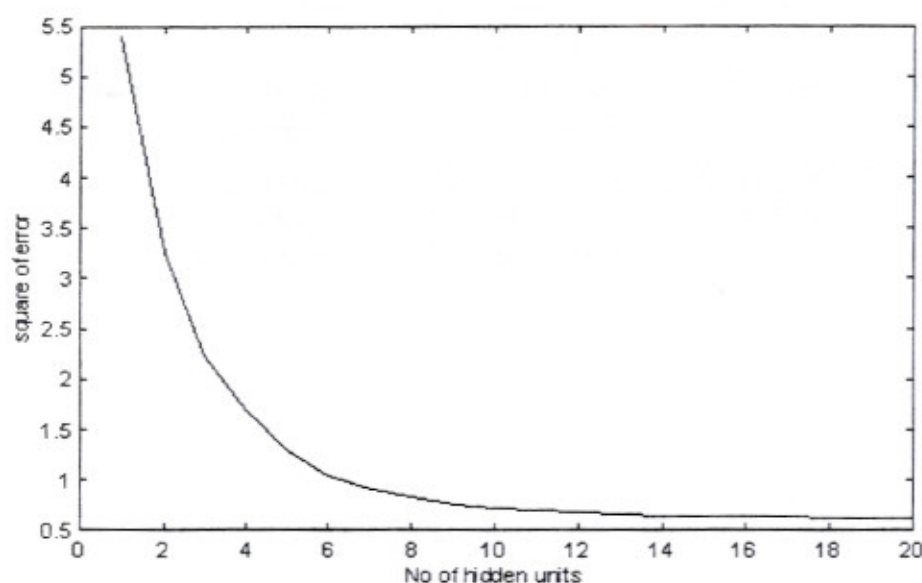
شکل ۳: الگوریتم آموزش شبکه عصبی

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

۳- نتایج شبه سازی

تقریب تابع $F(x) = \sin(X)$: هدف تقریب 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 مربع خطای شبکه بر اساس تعداد واحد های مخفی در شکل ۴ نشان داده شده است .



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

مطابق شکل ۴ ، زمانی که تعداد واحد های مخفی از ده واحد بیشتر میشود در خطای شبکه کاهش چندانی مشاهده نمی شود لذا تعداد واحدهای 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].

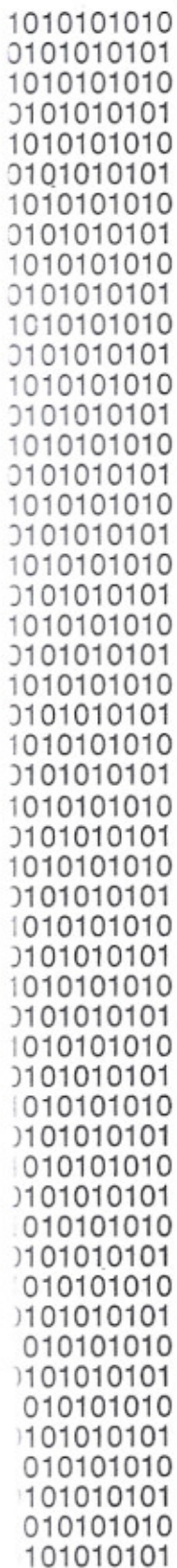
۴- نتیجه گیری

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 میشود. این روش دارای مزایای بسیاری می باشد که عبارتند از : سرعت یادگیری بسیار بیشتر از الگوریتم یادگیری انتشار خطا 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. Fren, (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. Lebie, (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.



• دیتای کاربردی
 • شبکه های کاربردی
 • پردازش های
 • سیستم های نرم افزاری
 • سیستم های
 • پردازش و کنترل دیتای کاربردی
 • پردازش های

Faculty Of Science And Technology
Engineering Department
TEHRAN - I.R.IRAN

23-25 Dec. 1997