

# شبکه های عصبی بازگشتی

# RECURRENT NEURAL NETWORKS

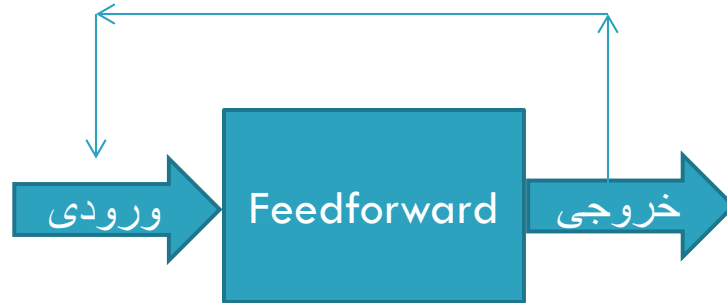
استاد : جناب آقای دکتر عبادتی  
ارائه کننده : علیرضا معتمدی

درس یادگیری ماشین

# شبکه های عصبی بازگشتی

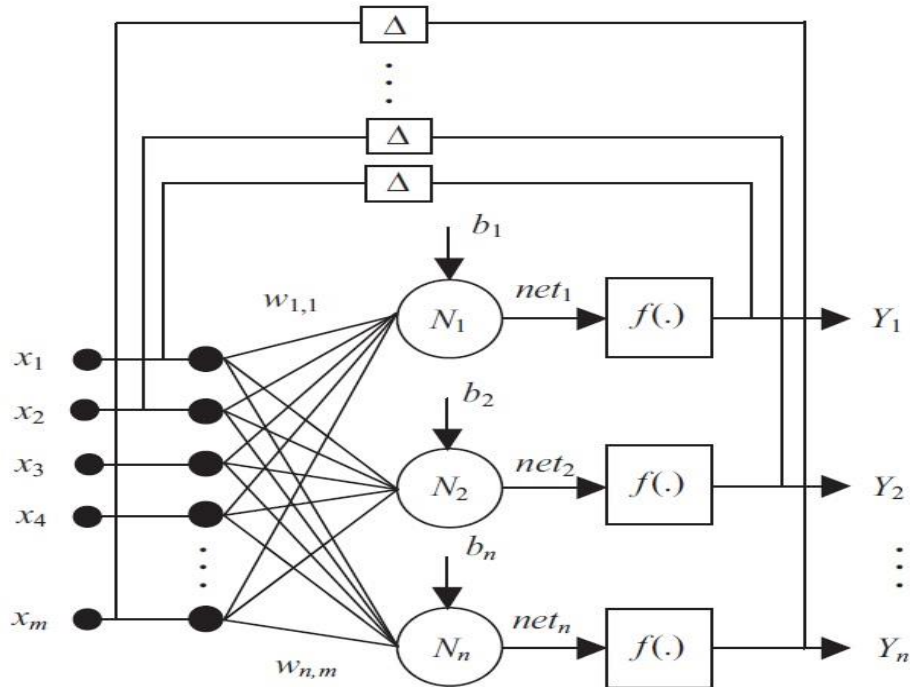
2

- شبکه بازگشتی از شبکه پیش خور (feedforward) با اتصال خروجی نورونها به ورودی خود به دست می آید.



# شبکه های عصبی بازگشتی

3



- ماهیت این حلقه بسته فعال کردن کنترل خروجی از طریق خروجیها است.
- اگر خروجی حاضر  $O(t)$  باشد، خروجی که بعد از کنترل بدست می آید  $O(t + \Delta)$  است.

# ویژگی شبکه های عصبی بازگشتی

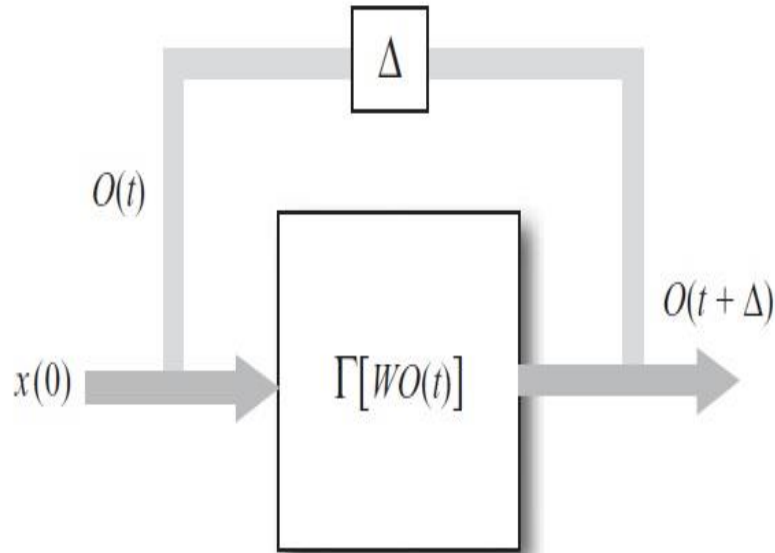
- یکی از مهمترین مشغله ها **مدل پویای سیستم**، می باشد که چگونگی تغییر سیستم از حالتی به حالت دیگر را نشان می دهد، شبکه های پیش خور نمی توانند چنین ویژگی را از خود بروز دهند حال آنکه شبکه های برگشتی توانایی انجام این کار را دارند.
- اگرچه شبکه های پس انتشار (backpropagation) بخوبی کار میکنند ولی اغلب مواقع زمان زیادی را صرف دستیابی به مجموعه ای معتبر از اوزان میکند. راه حلی که برای رفع زمان طولانی آموزش پیشنهاد میشود بکارگیری شبکه های بازگشتی است که در آنها از یادگیری هببین (Hebbian) به کار میرود و این یادگیری توسط یک محاسبات ساده قابل انجام است، لذا این شبکه ها در زمان اندک و با سرعت بالا قابل آموزش میباشند. (مانند هایفیلد)

# کاربرد شبکه های عصبی بازگشتی

- شبکه های بازگشتی شبکه هایی هستند که برای داده های **سری های زمانی** استفاده میشوند، خروجی شبکه در زمان  $t$  ورودی زمان  $t+1$  خواهد بود.
- برای مثال فرض کنید که میخواهیم با استفاده از شاخص های سهام  $x(t)$  در هر روز متوسط قیمت سهام برای روز بعد  $y(t+1)$  را پیش بینی کنیم. با داشتن یک سری از این اطلاعات، یکی از راههای بسیار ساده استفاده از شبکه تک سوپیه و استفاده از  $x(t)$  ها برای پیش بینی  $y(t+1)$  هاست. یکی از مشکلات این راه این است که مقدار  $y(t+1)$  فقط با توجه به  $x(t)$  پیش بینی میشود و هیچ تاثیری از مقادیر قبلی  $x$  نخواهد داشت. در حالی که این تاثیر بسیار حیاتی است، یک راه حل اساسی این مشکل استفاده از شبکه های بازگشتی است.

# طرز کار شبکه های عصبی بازگشتی

6



- ورودی  $X(0)$  فقط برای مقداردهی اولیه به شبکه نیاز است به طوری که  $O(0)=X(0)$ . سپس ورودی حذف میشود و سیستم برای  $t>0$  به صورت خود مختار خواهد بود.
- بنابراین یک حالت خاص این بازخورد پیکربندی صحیح سیستم است به این صورت که  $X(t)=X(0)$  و بعد از آن هیچ ورودی برای شبکه آماده نمیشود و یا برای  $t>0$  ورودی لازم نیست. ما می توانیم شبکه عصبی را برای یک زمان گسسته به صورت زیر بنویسیم:

$$O(k+1) = \Gamma(WO(k)) \quad \text{for } k=1,2,3,\dots$$

# طرز کار شبکه های عصبی بازگشتی

7

- شبکه در حالت شروع یک مرتبه در لحظه ۰ با  $x(0)$  مقدار دهی اولیه میشود و بعد از آن در حالات انتقال  $O(k)$  برای  $k=1,2,3,\dots$  تا زمانی که یک حالت تعادل پیدا کند ادامه می یابد.
- این حالت تعادل در اغلب اوقات **مجذوب کننده (attractor)** نامیده میشود.
- دنباله این حالتها به طور کلی غیرقطعی است و علاوه بر این، اغلب بسیاری از حالت های تعادلی که به طور بالقوه می توان توسط شبکه به آنها رسید وجود دارد.

# یادگیری در شبکه های عصبی

- هنگامی که یک شبکه عصبی در فاز آموزش است باید سه عامل را مد نظر قرار داد:
- ورودیهایی که از یک مجموعه آموزشی انتخاب میشوند بگونه ای باشند که پاسخ مطلوب سیستم برای این ورودیها مشخص باشد.
- پاسخ واقعی ایجاد شده با پاسخ مطلوب مقایسه شود و خطای موجود (انحراف پاسخ واقعی از پاسخ مطلوب) محاسبه شود.
- وزن ها به گونه ای اصلاح شود که خطا کاهش یابد.
- این گونه یادگیری را یادگیری نظارتی می نامند.



# یادگیری در شبکه های عصبی

- از آنجایی که به کار گیری پس انتشار (backpropagation) در عمل با موفقیت‌های چشمگیری همراه بوده است، شبکه های پیش خور (feedforward) در حوزه شبکه های عصبی بر دیگر شبکه ها غلبه یافتند اند.
- شبکه های برگشتی به این حد موفق نبوده اند و دلیل عمده آن پیچیدگی بسیار زیاد و زمان بر بودن مکانیز آموزش این دسته از شبکه ها است. دو شیوه اصلی برای حل این مشکل وجود دارند:
- آموزش از طریق پس انتشار زمانی : در این شیوه از الگوریتم ژنتیک برای بدست آوردن مجموعه وزنها استفاده میشود.
- یادگیری برگشتی بلادرنگ : از برگشت جزئی استفاده میکند و لذا در آن می توان از پس انتشار استاندارد استفاده کرد.

# یادگیری در شبکه های عصبی

10

- در ادامه به بررسی چند روش یادگیری در شبکه های عصبی می پردازیم:
  - یادگیری هببین Hebbian Learning
  - قاعده دلتا Delta Rule
  - یادگیری بازگشتی بی درنگ

# یادگیری هبیین

- بیشترین کوشش ها برای یافتن مکانیزم تعدیل وزن ها بر اساس نظرات دونالد هب میباشد بدین جهت آن را تحت عنوان شیوه یادگیری هبیین می نامند.
- **قاعده هب :** در صورتی که یک خروجی فعال باشد و در همان حال ورودی مربوط به یکی از وزنها نیز فعال باشد، آنگاه وزن متناظر با این ورودی را افزایش دهید.
- از لحاظ عددی این قاعده را میتوان به چندین شیوه تفسیر نمود، یکی از راهها این است که فرض کنیم ورودی ها و خروجی ها از نوع مقادیر دودویی باشند. در این صورت وزنها را در ابتدا صفر قرار میدهیم و سپس مقدار آنها را با به کار گیری مجموعه آموزشی به صورت روبرو محاسبه میکنیم:

$$w_i = \sum_{p=1}^p x_{ip} \cdot y_p$$

# یادگیری هبیین

- به کارگیری مقادیر دودویی ۰ و ۱ همواره نتایج خوبی را در استفاده از این مکانیزیم به هنگام کردن وزنها بدست نمی دهد.
- برای رفع این مشکل عدد ۰ را به ۱- و عدد ۱ را به ۱+ نگاشت میکنیم، با این کار در محاسبه وزنها امکان کاهش یک وزن نیز به وجود می آید.

# ویژگی یادگیری هبیین

- یادگیری هبیین همواره در شبکه موفقیت آمیز نیست.
- این شیوه تنها زمانی موفقیت آمیز است که ورودی ها **متعامد** یا **مستقل خطی** باشند، این به این دلیل است که یادگیری هبیین، **مقادیر واقعی خروجی را در نظر نمی گیرد** و فقط با مقادیر مطلوب خروجیها کار میکند. در صورتی که وزن ها به مقادیری که به میزان خطای بین خروجی واقعی و خروجی مطلوب است، بستگی دارند.
- محدودیت ذکر شده ما را به سمت قاعده جدید یادگیری (قاعده دلتا) می برد.

# یادگیری دلتا

- در قاعده دلتا، تعدیلاتی که در مورد وزنها صورت میگیرد به گونه ای است که اختلاف بین خروجی مطلوب و خروجی واقعی را کم کند.
- در این روش با محاسبه گرادیان خطا سعی در تعدیل اوزان داریم.
- ویدرو و هوف بر اساس فرضیات عنوان شده، شیوه ای برای تعدیل وزنها ایجاد نمودند، که این شیوه با نام های قاعده ویدرو و هوف، قاعده کمترین مربع میانگین LMS و یا قاعده دلتا نیز نام می برند.
- نام کمترین مربعات خطا LMS به این دلیل است که شیوه ویدرو-هوف برای بهنگام کردن وزنها، مربع خطای بین خروجی مطلوب و خروجی واقعی را کمینه میسازد.

# ویژگی یادگیری دلتا

- هزینه به کار گیری قاعده دلتا را میتوان به صورت تعداد تکرارهای مورد نیاز تا حصول همگرایی به راه حل در نظر گرفت.
- تعداد این تکرارها را نمی توان پیش بینی کرد.
- تضمینی برای حصول همگرایی در مورد همه مسائل وجود ندارد.
- پیچیدگی محاسباتی این روش زیاد است.

# یادگیری بازگشتی بی درنگ

- در یادگیری بازگشتی بی درنگ مانند روش دلتا به محاسبه گرادیان خطا می پردازیم، تا اختلاف بین خروجی مطلوب و خروجی واقعی را کم کنیم.
- تفاوت یادگیری بازگشتی بی درنگ با قاعده دلتا در این است که گرادیان خطا را در هر گام زمانی محاسبه میکند، از این جهت برای آموزشهای لحظه ای مناسب است.
- این روش دارای هزینه محاسباتی بالایی است از این جهت در مواردی که شبکه کوچک باشد مورد استفاده قرار میگیرد.



# انواع شبکه های عصبی بازگشتی

انواع مختلفی از شبکه های بازگشت کننده مطرح شده است:

- شبکه بازگشتی ساده
  - شبکه المان (Elman network)
  - شبکه جردن (Jordan network)
- شبکه انتقال بازگشتی (Recursive Transition Network)
- شبکه هاپفیلد (Hopfield network)
- شبکه هامینگ (Hamming Network)
- حافظه انجمنی دو سویه (Bidirectional Associative Memory)

# شبکه بازگشتی ساده

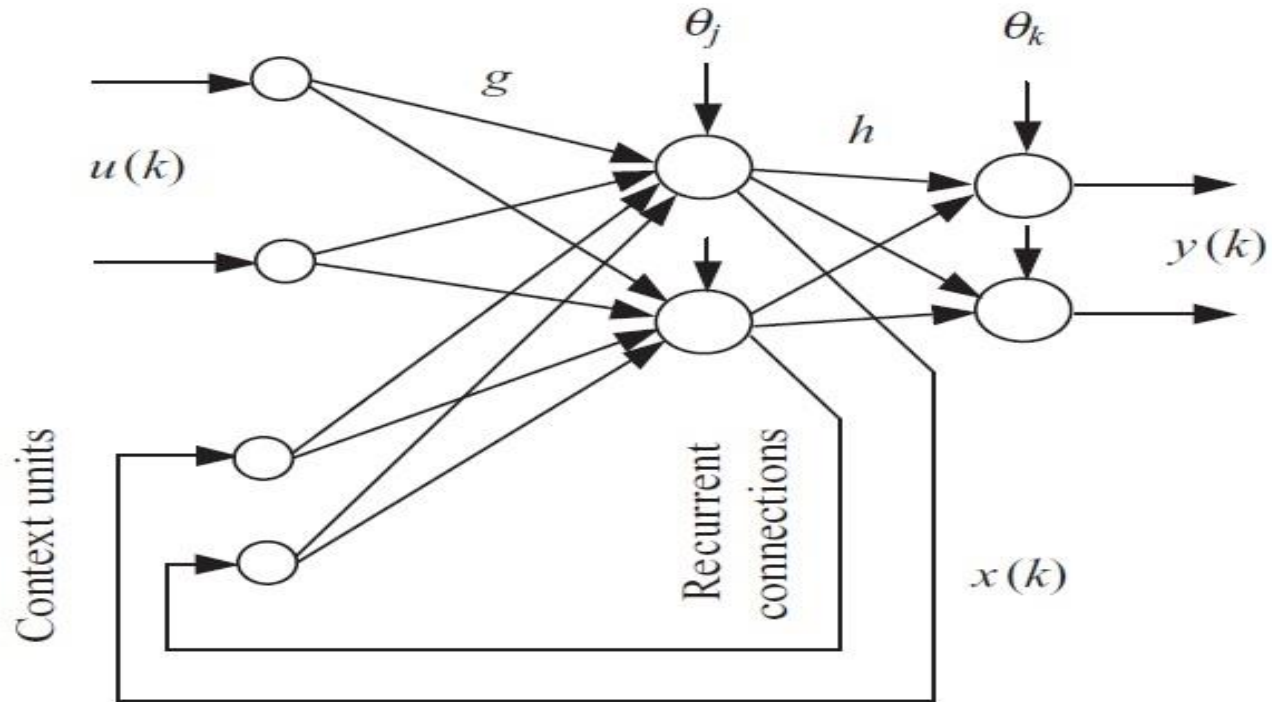
- شبکه بازگشتی ساده اولین بار توسط Jeff Elman مطرح شد.
- شبکه های بازگشتی ساده شبیه به شبکه های استاندارد feedforward میباشند. به این معنی که اطلاعات در تمام طول مسیر شبکه در هر گام رو به جلو منتشر می شوند.
- طرز کار این شبکه ها به این صورت است که حالت بعدی را توسط حالت موجود و وضعیت ذخیره شده در نورونهای پنهان پیش بینی میکند.

# شبکه المان

- شبکه المان از سه لایه شبکه پس انتشار (backpropagation) علاوه بر این از یک اتصال بازخورد از خروجی لایه پنهان به ورودی آن تشکیل شده است. این بازخورد اجازه می دهد تا شبکه های المان برای یادگیری و شناختن و تولید الگوهای موقتی و همچنین الگوهای فضایی توانایی پیدا کند.
- یک لایه اضافی از سلول های عصبی که کپی سلولهای فعال حال حاضر در لایه پنهان است ، و پس از به تاخیر انداختن یک واحد زمان، این مقادیر را مجدداً به عنوان ورودی های اضافی به سلول های عصبی لایه های پنهان داده میشود.

# شبكة المان

20



# شبکه المان

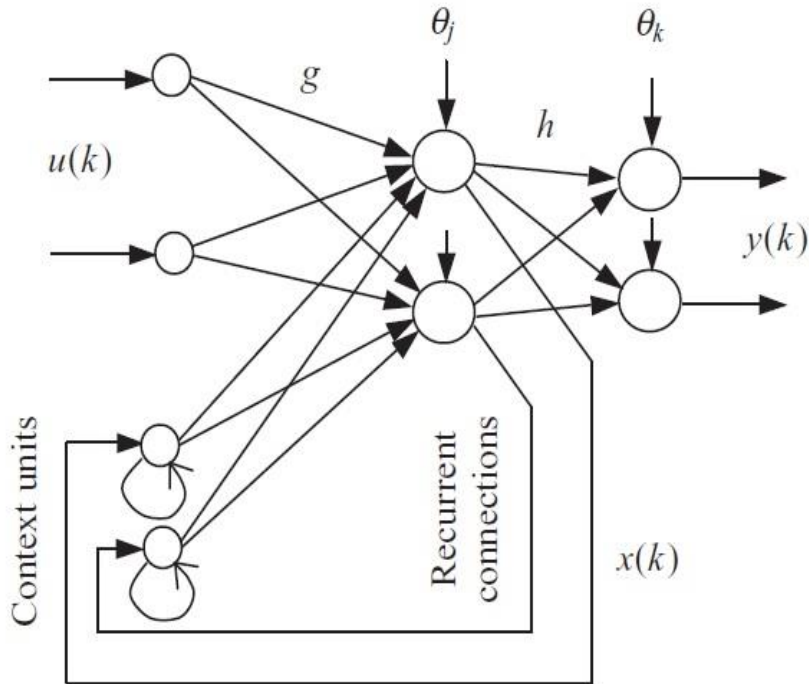
- لایه اول شامل دو گروه مختلف از سلول های عصبی است. یک گروه از نورون ها ورودی خارجی و گروهی از سلول های عصبی ورودی داخلی دریافت میکنند، این گروه **واحد مفهومی (context unit)** نیز نامیده می شود.
- **ورودی به واحد مفهومی خروجی از سلول های عصبی لایه پنهان می باشد.** خروجی از تمام واحد مفهومی و سلول های عصبی خارجی ورودی به سلول های عصبی پنهان را تغذیه می کند.
- **واحد مفهومی نیز به عنوان واحد حافظه** به عنوان ذخیره خروجی های قبلی از سلول های عصبی پنهان استفاده میشوند.

# شبکه المان

- اگر چه، به لحاظ نظری، یک شبکه المان با تمام ارتباطات بازخورد از لایه های پنهان به لایه مفهومی یک مجموعه را تشگیل میدهند میتوان آنها را به صورت قراردادی **n مرتبه ای** نامید، که  $n$  تعداد واحد مفهومی را نشان دهد.
- آنها را نمی توان با استفاده از الگوریتم backpropagation استاندارد آموزش داد.
- با معرفی اتصالات خود بازخورد به واحد مفهومی شبکه اصلی المان و در نتیجه افزایش ظرفیت حافظه آن، میتوان به اعمال الگوریتم استاندارد BP که برای آموزش شبکه است را بکار برد.

# شبکه المان تغییر یافته

23



- ایده معرفی اتصالات خود بازخورد برای واحد مفهومی از شبکه جردن اقتباس شده است.
- مقدار وزن خود ارتباط بین ۰ و ۱ قبل از شروع آموزش تنظیم میشود و ثابت است.

# الگوریتم آموزش شبکه المان

- در هر دوره، ورودی برای شبکه آماده شده و خروجی آن محاسبه میشود و با هدف مقایسه شده و یک توالی خطا تولید میشود.
- در هر مرحله زمان، خطایی که در پشت آن پخش میشود برای پیدا کردن شیب اشتباهات برای هر وزن و میزان اثر گذاری آن استفاده میشود.
- این گرادینان بدست آمده در واقع تقریبی از سهم وزن و میزان اثر گذاری خطاها از طریق پیوندهای مکرر نادیده گرفته میشوند و یک مقدار انتخاب شده از طرف کاربر برای شیب استفاده می شود و از آن برای به روز رسانی وزن ها با تابع آموزش پس انتشار بکار میرود.



# الگوریتم آموزش شبکه المان

- با توجه به استفاده از گرادینان برای تقریب خطا، شبکه المان ممکن است از انواع دیگر شبکه **قابل اعتماد کمتری** داشته باشد.
- برای شبکه المان در یادگیری مشکل آن است نیاز دارد نورون در لایه مخفی مقدار درستی بگیرند، در واقع برای حل این مشکل به انواع دیگر شبکه نیاز است. بنابراین، آموزش شبکه المان باید با **تعداد مناسبی از سلول های عصبی پنهان** شروع شود.

# مزایا و معایب شبکه المان

## ■ مزایا

- این شبکه قادر به تخمین هر تابعی با تعداد ناپیوستگی های محدود است.
- امکان به کار گیری پس-انتشار برای آموزش آن وجود دارد.

## ■ معایب

- در تعداد ناپیوستگیهای نامحدود قادر به تخمین تابع نیست.
- این شبکه به اندازه سایر شبکه ها قابل اعتماد نمی باشد، زیرا آموزش در این شبکه ها براساس شیب خطا صورت میگیرد.
- برای رسیدن به جواب نیاز به تعداد بیشتری نورون نسبت به سایر شبکه ها دارند.

# کاربردها شبکه المان

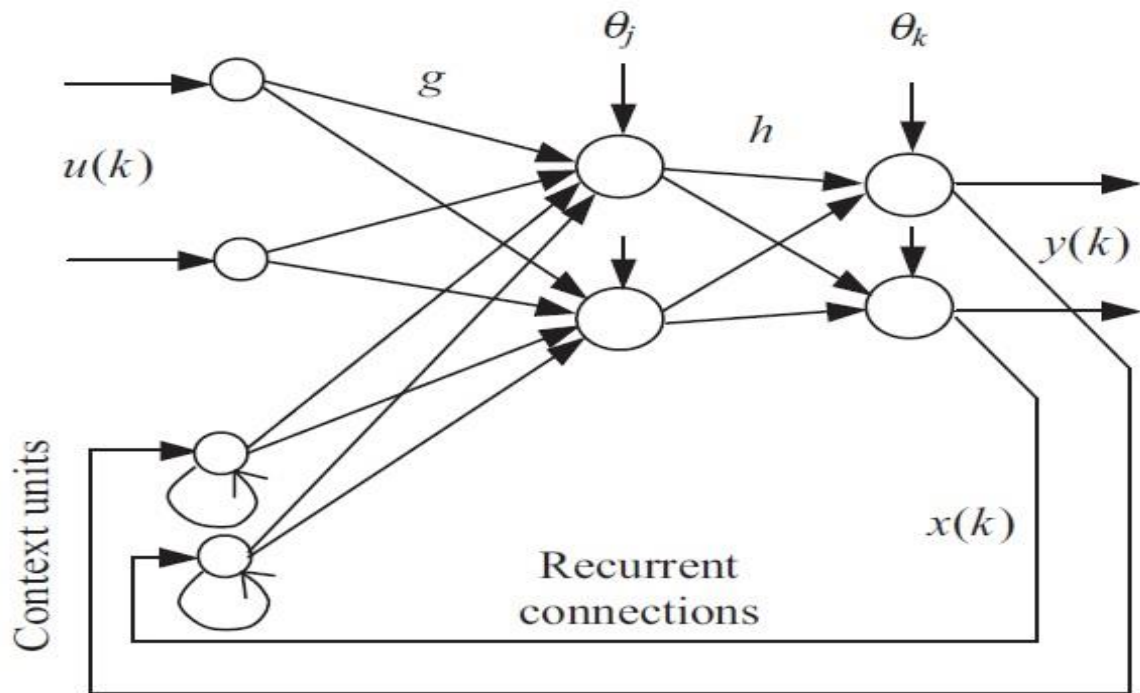
- این شبکه اساسا برای باز شناسی صدا ایجاد شده بود.
- در تشخیص سیستم و پیش بینی کوتاه مدت که برای برنامه ریزی حرکت رباطها به کار گرفته شد.
- در مدل سازی پویای سیستم ها کاربرد دارد.

# شبکه جردن

- شبکه های عصبی جردن مانند شبکه های اصلاح شده المان دارای سه لایه است.
- **اتصالات اصلی بازخورد از لایه خروجی به لایه مفهومی** گرفته شده است.
- به لحاظ نظری که یک شبکه جردن قادر به پشتیبانی از سیستم های دینامیکی دلخواه نیست. بنابراین، با اضافه کردن قابلیت اتصال به بازخورد از لایه های پنهان به لایه مفهومی، شبیه به یک شبکه المان است، که شبکه جردن در آن تغییری داده است.
- شبکه جردن را می توان با استفاده از الگوریتم استاندارد BP به مدل سیستم های مختلف پویا آموزش داد.
- اگر الگوریتم استاندارد BP استفاده شود همانند شبکه المان تغییر ارزش وزن های ارتباط بازخورد توسط کاربر ثابت در نظر گرفته میشود.

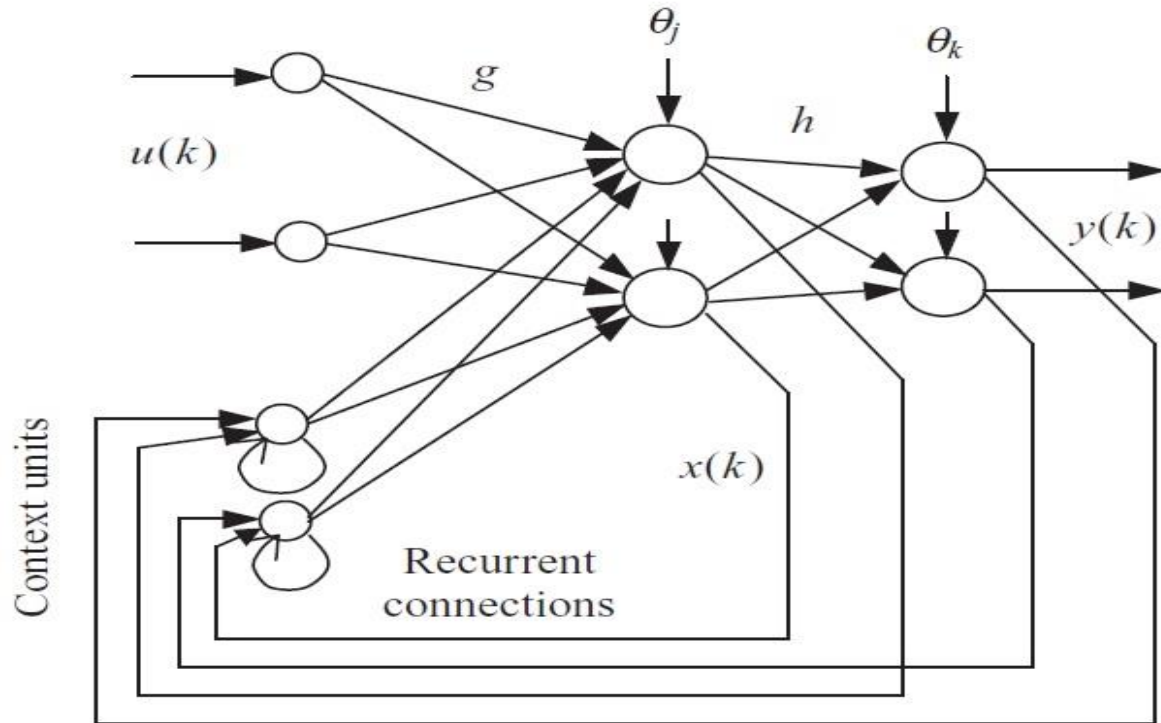
# شبکه جردن

29



# شبکه جردن تغییر یافته

30



# شبکه انتقال بازگشتی

- شبکه انتقال بازگشتی یک گراف جهت دار است.
- یک شبکه انتقال بازگشتی از عناصر زیر ساخته میشود:
  - گره ها Nodes
  - بردارهای انتقال
- این شبکه باید یک حالت شروع داشته باشد.
- میتوان چند حالت خاتمه داشت.

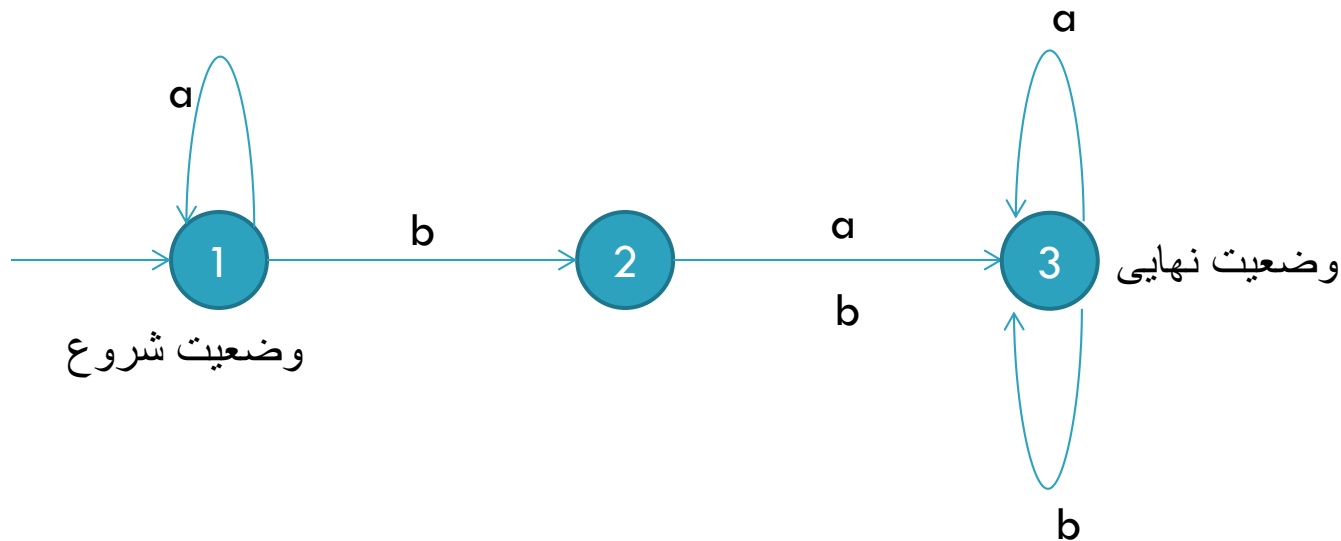
# شبکه انتقال بازگشتی

- اگر بتوان مسیری از حالت ابتدایی به حالت نهایی با خواندن کل رشته ورودی پیدا کرد آنگاه می توان نتیجه گرفت که رشته معتبر است.
- هر عبارتی که توسط این شبکه معتبر شناخته میشود، می تواند شامل عبارات معتبر کوچکتری باشد، از این جهت این گونه شبکه ها را بازگشتی گویند.



# شبکه انتقال بازگشتی

33



# کاربردهای شبکه انتقال بازگشتی

- این شبکه کاربردهایی در :
  - زبان های برنامه نویسی
  - زبان های طبیعی
  - تجزیه تحلیل واژگان
  - نمایش قوانین یک گرامر مستقل از متن

# مزایا و معایب شبکه انتقال بازگشتی

35

## ■ مزایا

- شمردن الگوهای تکراری در زبان
- کاربرد زیادی در زبان طبیعی دارد.

## ■ معایب

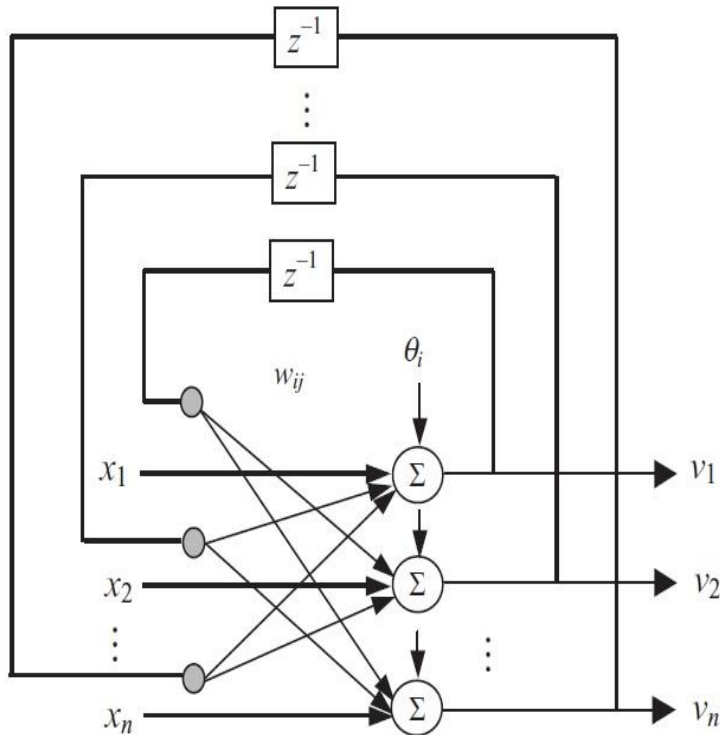
- در بعضی موارد استفاده از آنها دچار محدودیت میشود.

# شبکه هایفیلد

- شبکه هایفیلد یک شبکه بازگشت کننده است که توسط Hopfield در سال ۱۹۸۲ ارائه شده، که دارای خواص خودکار انجمنی (auto-associative) است.
- این شبکه به طور کامل متصل است، اما هیچ ارتباطی از خود نوروں به خودش وجود ندارد.
- شبکه به ذخیره یک یا چند **بردار هدف ثابت** می پردازد. این بردار ثبات میتواند به عنوان **خاطرات** که شبکه به یاد می آورد زمانی که با بردار مشابه فراخوانی میشود استفاده شود.

# شبکه هایفیلد

37



- این  $n$  سلول عصبی از داشتن مقادیر آستانه  $\theta_i$  تشکیل میشوند.
- ورودی بازخورد به نورون  $i$ ام برابر است با مجموع وزن از نورون خروجی  $V_j$ ، که در آن  $j=1,2,\dots,n$  است.
- ما میتوانیم ورودی کل  $net_i$  به عنوان نورون  $i$ ام بیان کنیم:  

$$net_i = \sum_{j=1}^n W_{ij} \cdot V_j + X_i \text{ for } j \neq i, i = 1, 2, \dots, n$$
 که در آن  $X_i$  ورودی خارجی به نورون  $i$ ام است.

# شبکه هایفیلد

- ماتریس  $W$ ، به نام ماتریس اتصال،  
ماتریس  $n \times n$  تعریف شده است :

$$W = \begin{bmatrix} 0 & w_{12} & \cdots & w_{1n} \\ w_{21} & 0 & \cdots & w_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{n1} & w_{n2} & \cdots & 0 \end{bmatrix}$$

- این ماتریس یک ماتریس متقارن است و  
عناصر قطر اصلی آن صفر هستند به این  
معنی که هیچ ارتباطی از خود نوروں به  
خودش وجود ندارد.

# شبکه هایفیلد

- لازم به ذکر است که یک شبکه Hopfield ممکن است در یک حالت پیوسته و یا گسسته بسته به مدل نرون مورد استفاده قرار گیرد.
- بیشترین کاربرد شبکه هایفیلد در مسائل **بهینه سازی** است.
- این شبکه میتواند با یک یا چند بردار ورودی که معرف حالات آغازین میباشد آموزش ببیند، بعد از اینکه حالات آغازین به شبکه اعمال شدند، شبکه خروجی خود را تولید میکند و این خروجی به ورودی پسخورد میشود، این فرآیند آنقدر تکرار میشود تا خروجی به حالت پایای خود برسد.

# طرز کار شبکه هایفیلد

- زمانی که یک الگوی ورودی به شبکه نشان داده میشود، خروجی نهایی یا مطلوب بلافاصله تولید نمیشود بلکه در خلال مسیری که شبکه طی میکند تا به یک حالت تعادل برسد، خروجی تغییراتی میکند.
- منظور از حالت تعادل حالتی است که تغییراتی بیشتر صورت نگیرد.



# طرز کار شبکه هایفیلد

- پاسخ یک عصب منفرد در شبکه بصورت زیر محاسبه میشود:

$$y_j = 1 \text{ if } \sum_{i=1, i \neq j}^n w_{ij} \cdot x_i > T_j$$

$$y_j = 0 \text{ if } \sum_{i=1, i \neq j}^n w_{ij} \cdot x_i < T_j$$

- این بدین معنا است که برای  $j$  امین نورون، ورودیهای تمام نورونهای دیگر وزندهی شده، با یکدیگر جمع میشوند.
- خروجی هر نورون از یک تابع محدود کننده سخت (Hard Limit) استفاده میکند.

# طرز کار شبکه هایفیلد

- حال وزن ها بصورت یک ماتریس  $[W]$  محاسبه میشوند، این فاز آموزش شبکه در مقایسه با شبکه های feedforward بسیار کوتاه است، زیرا در این فاز تنها یک محاسبه انجام میشود. (یادگیری هبیین)
- زمانی که وزن ها مرتب شدند، شبکه الگوهای ورودی را بخاطر می سپارد و الگوها بصورت حالت های تعادل ذخیره میشوند.

# طریقه محاسبه وزن ها شبکه هایفیلد

- برای محاسبه وزن ها از فرمول زیر استفاده میشود:

$$[W] = [X]^t[X] - [P]$$

- در این فرمول  $[X]$  ماتریس الگوی ورودی است که با اعداد 1- و 1+ کد دهی شده است.

- ماتریس  $P$  ماتریس یکانی است که  $P$  در آن تعداد الگوهای موجود در مجموعه آموزشی است که در آن ضرب شده است. ماتریس یکانی بدین جهت معرفی میشود که تضمین شود هیچ اتصالی از یک خروجی نوروں به ورودی خودش وجود نداشته باشد.

# مثالی برای تعیین وزن های شبکه هایپرفیلد

44

■ به عنوان مثال اگر الگو های زیر را داشته باشیم :

$$X_1 = [0 \ 0 \ 1 \ 1] , X_2 = [0 \ 1 \ 0 \ 1]$$

■ معمولاً ورودیهای هایپرفیلد ۰ و ۱ هستند که آنها را به ۱- و ۱+ تبدیل میکنیم:

$$X_1 = [-1 \ -1 \ +1 \ +1] , X_2 = [-1 \ +1 \ -1 \ +1]$$

■ در اینجا  $p=2$  است زیرا ۲ الگو معرفی کردیم.

# مثالی برای تعیین وزن های شبکه هایپرفیلد

45

$$[W] = [X]^t [X] - [P]$$

-1	-1	*	-1	-1	+1	+1	-	2	0	0	0	=	0	0	0	-2
-1	+1		-1	-1	+1	+1		0	2	0	0		0	0	-2	0
+1	-1		-1	+1	-1	+1		0	0	2	0		0	-2	0	0
+1	+1		-1	+1	-1	+1		0	0	0	2		-2	0	0	0

# تعیین تعداد عصبهای شبکه هایفیلد

- فرض کنید میخواهیم شبکه که طراحی میکنیم در ۲ حالت به تعادل برسد، ممکن است شبکه در ۳ حالت به تعادل برسد که یکی از این حالتها مطلوب نمی باشد و ممکن است شبکه خروجی نامطلوب یا دارای نویز تولید کند.
- یکی از راههای جلوگیری از حالات تعادل نامطلوب افزایش تعداد عصبها است.
- آنگونه که هایفیلد می گوید، برای  $n$  عصب تنها میتوان در حدود  $P=0.15n$  الگو ذخیره کرد.
- طبق نظر هایفیلد برای ذخیره نمودن ۲ الگو ۱۴ عصب مورد نیاز است. لذا مسئله ای با ۱۴ ورودی و کمتر ارزش حل کردن ندارد.

# تفاوت بین شبکه های برگشتی و شبکه هایپفیلد

- در یک شبکه هایپفیلد، نورون ها به طور غیر همزمان بهنگام می شوند اما در شبکه های برگشتی، بهنگام سازی نورون ها همزمان صورت می گیرد و این به این معنی است که تمامی نورون ها هم زمان فعال می شوند.
- آموزش در شبکه های هایپفیلد مستلزم انجام یک محاسبه ریاضی است (یادگیری هبیین) حال آنکه در شبکه های برگشتی، آموزش پیچیده تر است و درگیر تعداد زیادی تکرار است.

# مثالی برای شبکه هاپفیلد در MATLAB

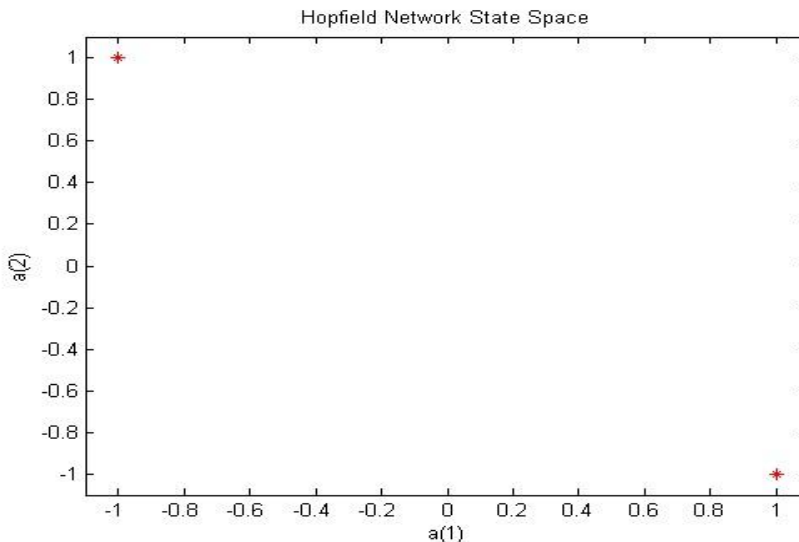
48

- فرض کنید قصد طراحی شبکه هی با دو نقطه تعادل را داریم، بردار هدف را بصورت زیر تعریف میکنیم:

```
>> T = [+1 -1;-1 +1]
```

```
T =
```

```
    1    -1  
   -1     1
```





# مثالی برای شبکه هایفیلد در MATLAB

49

■ حال شبکه را برای بردار هدف ایجاد می کنیم

```
net = newhop(T);
```

■ سپس با شبیه سازی عملکرد شبکه را برای دو نقطه تعریف شده آزمایش میکنیم.

```
[Y,Pf,Af] = net([],[],T);
```

Y

Y =

1    -1

-1    1

# مثالی برای شبکه هایفیلد در MATLAB

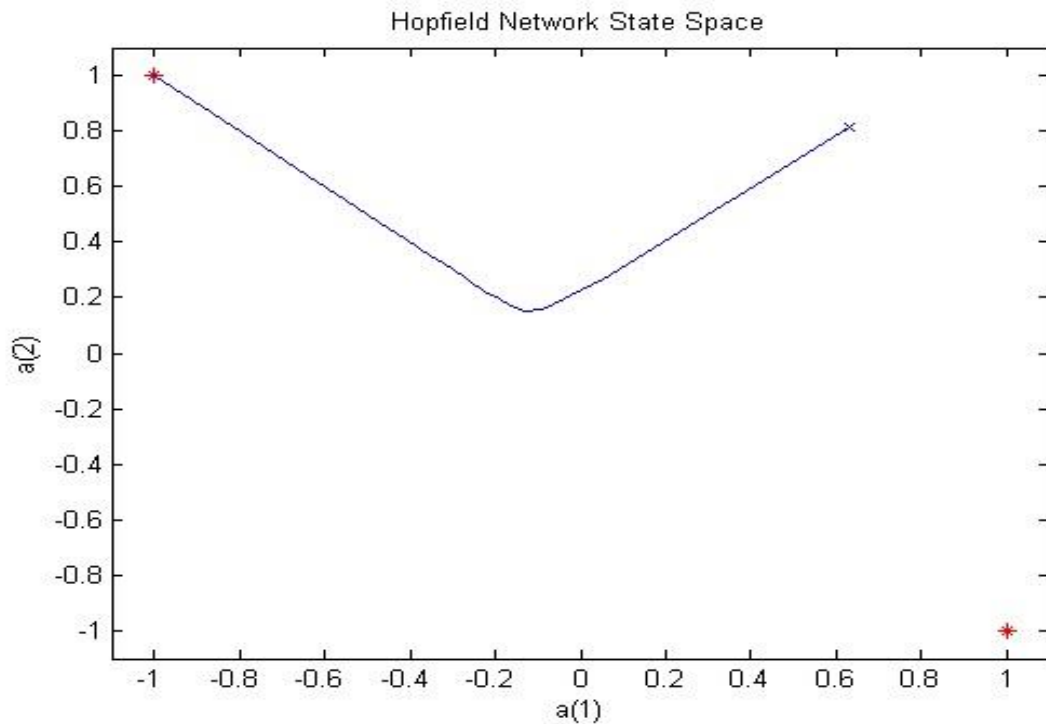
50

- حال یک نقطه تصادفی تعریف می کنیم و عملیات شبکه را در ۲۰ مرحله که به سمت متعادل شدن در یکی از نقاط هدف تعریف شده پیش میرود را روی شکل مبینیم.

```
a = {rand(2,1)};  
[y,Pf,Af] = net({20},{},a);  
record = [cell2mat(a) cell2mat(y)];  
start = cell2mat(a);  
hold on  
plot(start(1,1),start(2,1),'bx',record(1,:),record(2,:))
```

# مثالی برای شبکه هاپفیلد در MATLAB

51



# مثالی برای شبکه هایفیلد در MATLAB

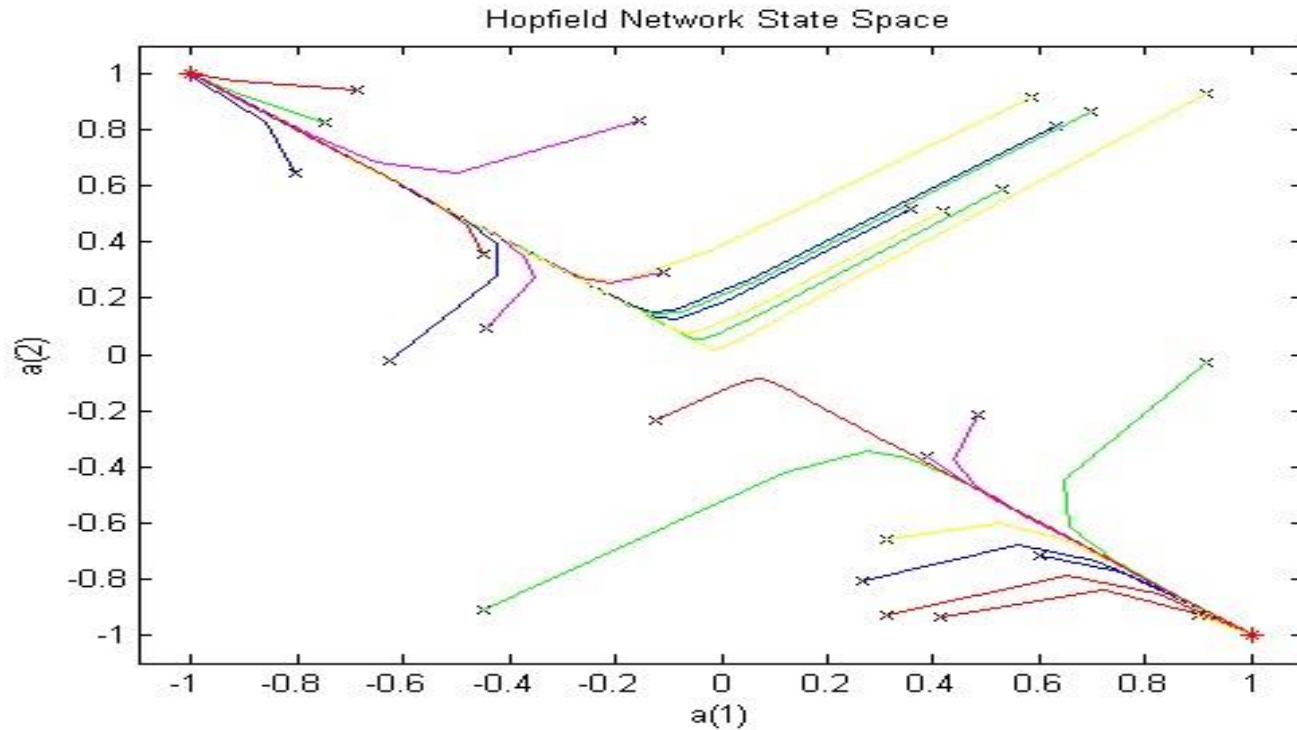
52

این عملیات را برای ۲۵ نقطه دیگر نیز انجام می دهیم تا نشان دهیم که هر نقطه تصادفی به سمت نقطه هدفی که به آن شبیه تر است همگرا میشود:

```
color = 'rgbmy';  
for i=1:25  
    a = {rand(2,1)};  
    [y,Pf,Af] = net({20},{},a);  
    record=[cell2mat(a) cell2mat(y)];  
    start=cell2mat(a);  
    plot(start(1,1),start(2,1),'kx',record(1,:),record(2,:),color(mod(i,5)+1))  
end
```

# مثالی برای شبکه هاپفیلد در MATLAB

53

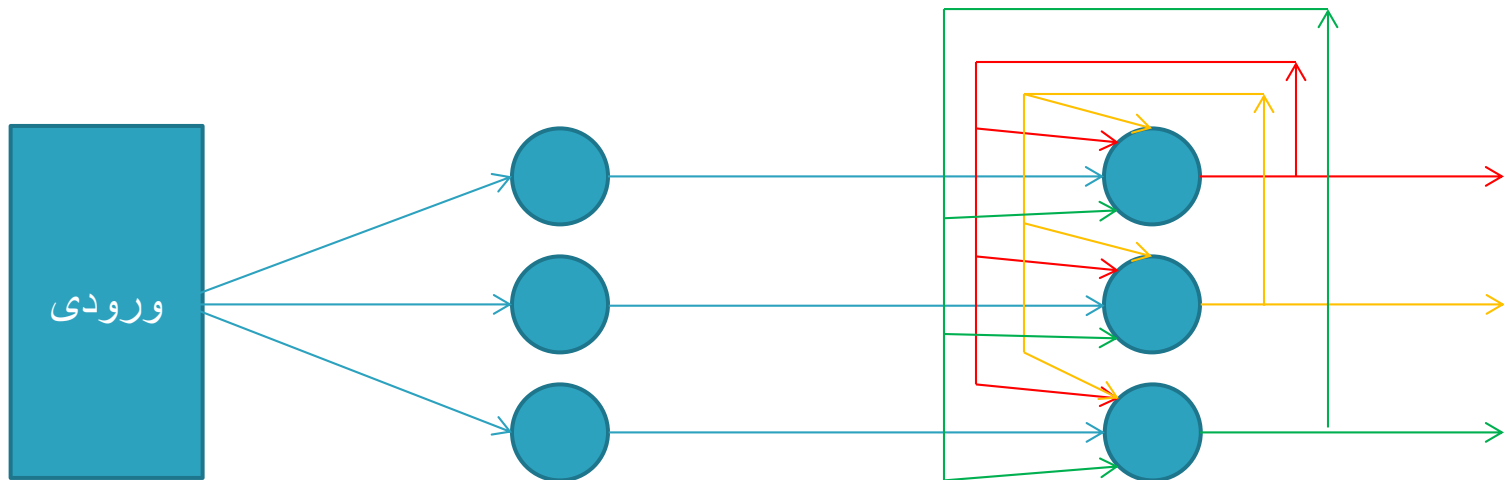


# شبکه عصبی بازگشتی هامینگ

- گونه ای تغییر یافته از شبکه هایفیلد، شبکه همنگ است.
- شبکه همنگ شبکه ای دو لایه است، که لایه نخست برای محاسبه یک امتیاز به کار برده میشود و لایه دوم آن برای انتخاب بیشینه به کار میرود.
- لایه دوم اساساً یک شبکه هایفیلد است با این تفاوت که بازخور هر نورون به خودش مجاز است.

# شبکه عصبی بازگشتی هاپینگ

55



# طرز کار شبکه عصبی بازگشتی هامینگ

56

- هر یک از نورون های موجود در لایه نخست یک بیشینه پاسخ  $n$  را برای یک الگو مشخص و یک مقدار کمتر از این بیشینه را برای سایر الگوها تولید میکند. بنابراین می توان گفت هر یک از نورون ها یک الگو را نشان میدهد.
- در صورتی که  $p$  الگو وجود داشته باشد  $p$  نورون در هر دو لایه وجود خواهد داشت.
- زمانی که یک الگو جدید به لایه نخست نشان داده می شود، خروجی هر یک از نورون های این لایه، معیاری برای سنجش فاصله هامینگ از الگوی موجود در مجموعه آموزشی خواهد بود.



# طرز کار شبکه عصبی بازگشتی هامینگ

57

- فاصله هامینگ بین دو الگو معادل تعداد بیت‌هایی می باشد که بین دو الگو متفاوت هستند.
- هدف ای شبکه تعیین الگوی ذخیره شده ای است که دارای بیشترین شباهت با الگوی ورودی است، بنابراین کار لایه دوم انتخاب نوروونی است که دارای خروجی بیشینه باشد.
- اگر خروجی نوروون  $z_m$  دارای مقدار بیشینه باشد، یعنی الگوی  $z_m$  دارای بیشترین تشابه با ورودی است.

# ویژگی شبکه عصبی بازگشتی هامینگ

58

- شبکه هامینگ شبکه عجیبی به نظر می رسد زیرا تابعی را اجرا می کند که می توان خیلی راحت تر توسط یک الگوریتم متعارف اجرا نمود. با این اوصاف این شبکه دو ویژگی شبکه عصبی را به نمایش می گذارد:
- نخستین ویژگی در لایه نخست می باشد جایی که توانایی یک نورون برای بدست آوردن پاسخ یک خروجی بزرگ به یک الگوی ورودی، توسط ذخیره کردن یک کپی از الگوی به شکل وزنه‌های این الگو، بدست می آید.
- ویژگی دوم در لایه دوم نهفته است، جایی که نورون ها برای حذف یکدیگر باهم رقابت میکنند، و نتیجه به صورت یک تابع برنده همه را می برد بدست می آید.
- هر دو این ویژگی ها به طور گسترده در شبکه های **خود سازمانده** به کار می روند.

# حافظه انجمنی دوسویه BAM

- یکی از راههای فکر کردن در مورد یک شبکه عصبی، در نظر گرفتن آن بصورت حافظه ای است که در آن اطلاعات به مانند یک Ram به شکل صریح و بی واسطه ذخیره نمی شود بلکه اطلاعات در سراسر شبکه توزیع میگردد. الگوهای ورودی را می توان بصورت آدرسهایی که خروجی مطلوب را تولید میکنند، در نظر گرفت.
- آنچه برای رسیدن به این هدف باید ایجاد شود، یک حافظه انجمنی دو سویه است، که بتواند تعدادی از الگوهای ورودی متفاوت را به یک الگو خروجی مرتبط سازد.

# حافظه انجمنی دوسویه BAM

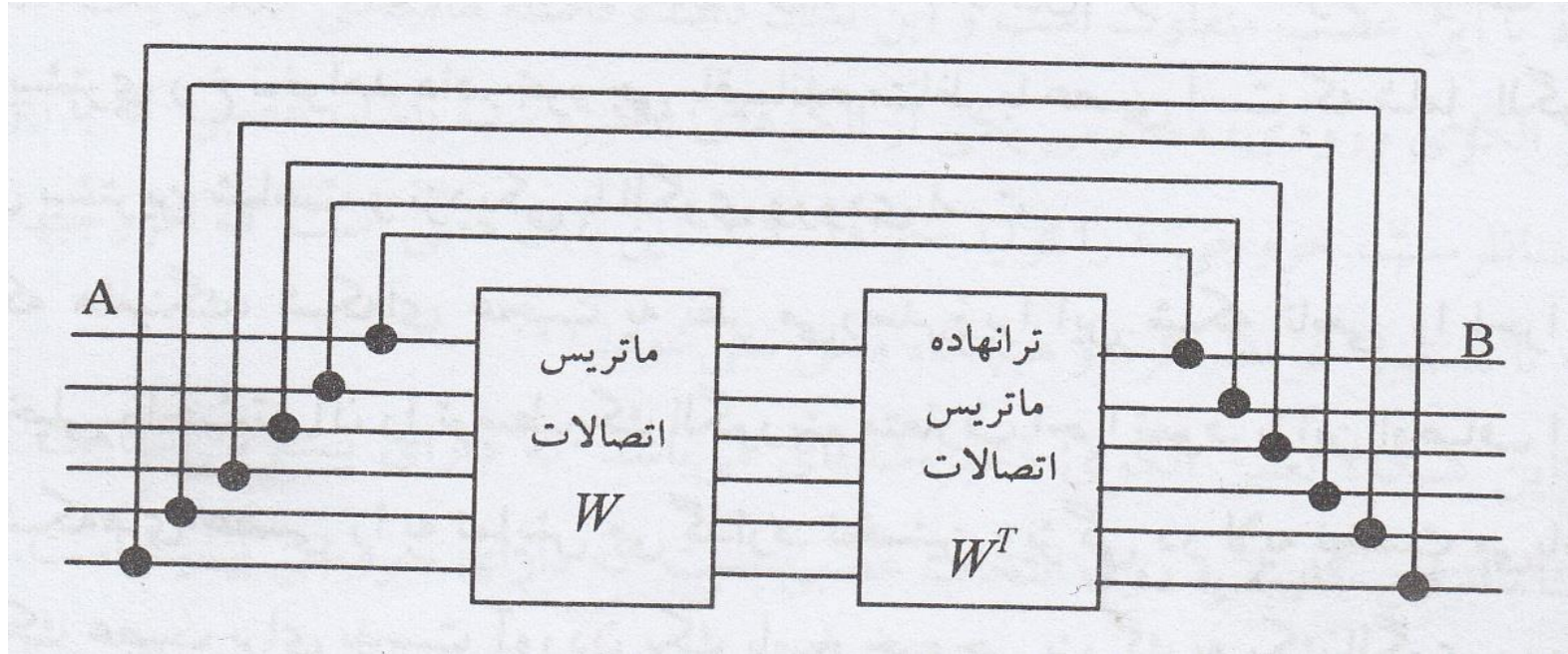
- در اینجا باید به دو نکته توجه کنیم:
- نخست آنکه الگوهایی که میخواهیم به هم ربط داده شوند می توانند هر چیزی باشند. به عنوان مثال، ورودی می تواند یک عدد باشد و خروجی نسخه صوتی از آن عدد به زبانی دلخواه باشد.
- دوم آنکه نیازی نیست برای ذخیره الگوهای خروجی، حافظه ای جداگانه تخصیص داده شود، در خلال آموزش نظارتی، الگوهای ورودی به همراه خروجی مورد نیاز به سیستم ارائه میشوند. بدین ترتیب سیستم به طور همزمان به صورت یک طبقه بند الگو و یک حافظه انجمنی عمل می کند.

# حافظه انجمنی دوسویه BAM

- یک حافظه رابط دوسویه بسیار شبیه شبکه هاپفیلد است، اما دارای دو لایه است.
- در BAM از ماتریس اتصالات استفاده می شود.
- ماتریس اتصالات به صورت وسیله برای محاسبه خروجیهایی که توسط تعدادی ورودی تولید می شود، معرفی می شود.
- نکته ای که وجود دارد این است که در BAM از ماتریس اتصالات برای محاسبه ورودیهایی که توسط مجموعه ای از خروجیها تولید میشوند نیز استفاده میشود.

# حافظه انجمنی دوسویه BAM

62



# حافظه انجمنی دوسویه BAM

63

- ماتریس اتصالات  $[W]$  از حاصلضرب ماتریس الگوی ورودی  $[X]$  در ماتریس الگوی خروجی  $[Y]$  بدست می آید (عناصر این ماتریسها به صورت مقادیر  $-1$  و  $+1$  ذخیره می شوند)

$$[W] = [X]^t \cdot [Y]$$

# طرز کار حافظه انجمنی دوسویه BAM

64

- عملکرد BAM به این شکل است که یک الگوی ورودی  $[A]$  به شبکه نشان داده می شود و این الگو پس از عبور از ماتریس اتصالات، یک الگو خروجی  $[B]$  تولید میکند. بنابراین :

$$[B(k)] = f([A(k)] \cdot [W])$$

- در این فرمول  $k$  زمان را نشان می دهد.
- ماتریسهای  $[A(k)]$  و  $[B(k)]$  به ترتیب با ماتریسهای  $[X]$  و  $[Y]$  برابرند با این تفاوت که در آنها بجای  $+1$  و  $-1$  از  $1$  و  $0$  استفاده شده است.



# طرز کار حافظه انجمنی دوسویه BAM

65

- خروجی نورنها توسط تابع  $f()$  تولید می شوند که مشابه شبکه هایفیلد، این تابع یک محدود کننده سخت است با این تفاوت که دارای حالت ویژه ای برای مقدار صفر است :

اگر عنصر متناظر با آن از ماتریس  $[A(k)][W]$  بزرگتر از ۰ باشد ؛  $b(k+1) = 1$

اگر عنصر متناظر با آن از ماتریس  $[A(k)][W]$  کوچکتر از ۰ باشد ؛  $b(k+1) = 0$

اگر عنصر متناظر با آن از ماتریس  $[A(k)][W]$  برابر با ۰ باشد ؛  $b(k+1) = b(k)$

# طرز کار حافظه انجمنی دوسویه BAM

66

- سپس خروجی یعنی  $[B]$  مجدداً از ماتریس اتصالات عبور داده می شود تا یک الگوی ورودی جدید  $[A]$  تولید شود:

$$[A(k+1)] = f([B(k)] \cdot [W]^t)$$

- الگوهای  $[A]$  و  $[B]$  متناوباً با روشی که گفته شده از ماتریس اتصالات عبور داده می شوند تا جاییکه دیگر تغییری در مقادیر  $[A]$  و  $[B]$  مشاهده نشود.

# حافظه انجمنی دوسویه BAM

67

■ مثال

$$[A] = \begin{bmatrix} A1 \\ A2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad [B] = \begin{bmatrix} B1 \\ B2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

■ با جایگزینی ۰ با ۱- و ۱ با ۱+ خواهیم داشت:

$$[X] = \begin{bmatrix} X1 \\ X2 \end{bmatrix} = \begin{bmatrix} +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & +1 & -1 & -1 & -1 \end{bmatrix}$$
$$[Y] = \begin{bmatrix} Y1 \\ Y2 \end{bmatrix} = \begin{bmatrix} +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & -1 \end{bmatrix}$$

# حافظه انجمنی دوسویه BAM

68

■ محاسبه ماتریس اتصالات :

$$[W] = [X]'[Y] = \begin{bmatrix} +1 & +1 \\ -1 & +1 \\ +1 & +1 \\ -1 & -1 \\ +1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix}$$

# حافظه انجمنی دوسویه BAM

69

■ به طور مثال برای  $[A1]$  داریم:

$$[1 \ 0 \ 1 \ 0 \ 1 \ 0] \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & -2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix} = [4 \ 2 \ -2 \ -4]$$

■ پس از به کار گیری تابع غیر خطی  $f()$  خروجی برابر  $[1 \ 1 \ 0 \ 0]$  می باشد که با  $[B1]$  یکسان است.

# حافظه انجمنی دوسویه BAM

70

■ به طور مشابه برای [B2] داریم :

$$[1 \ 0 \ 1 \ 0] \begin{bmatrix} 2 & 0 & 2 & -2 & 0 & -2 \\ 0 & -2 & 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 & -2 & 0 \\ -2 & 0 & -2 & 2 & 0 & 2 \end{bmatrix} = [2 \ 2 \ 2 \ -2 \ -2 \ -2]$$

■ پس از به کار گیری تابع غیر خطی  $f()$  خروجی برابر  $[1 \ 1 \ 1 \ 0 \ 0 \ 0]$  می باشد که با [A2] یکسان است.



# حافظه انجمنی دوسویه BAM

- با توجه به مطالبی که گفته شد، BAM مانند سایر شبکه های برگشتی ( به طور نمونه شبکه هایفیلد) الگوی ورودی را به عنوان حالت آغازین سیستم دریافت میکند و در یک حالت تعادل، که در آن الگوی خروجی مطلوب قابل دریافت است به کارش پایان می دهد.
- دستیابی به حالت تعادل، توسط عبور دادن داده ها از شبکه بصورت رو به جلو و رو به عقب انجام می گیرد و لذا گونه ای از مکانیزم بازخور به کار گرفته می شود.

# شبکه های عصبی بازگشتی

72

موفق باشید