هوش محاسباتی

استاد: محمد باقر منهاج

موضوعات

- تاریخچه
- هوش مصنوعی
- چالشهای بنیادین هوش مصنوعی
- هوش محاسباتی
-

تاریخچه



- آلن تورینگ (۱۹۵۰) یکی از بحث برانگیزترین پرسش‌های فلسفی تاریخ را پرسید:

آیا ماشین می‌تواند فکر کند؟

آیا یک کامپیوتر می‌تواند بازی تقلید را با موفقیت پشت سر بگذارد؟

- تست تورینگ

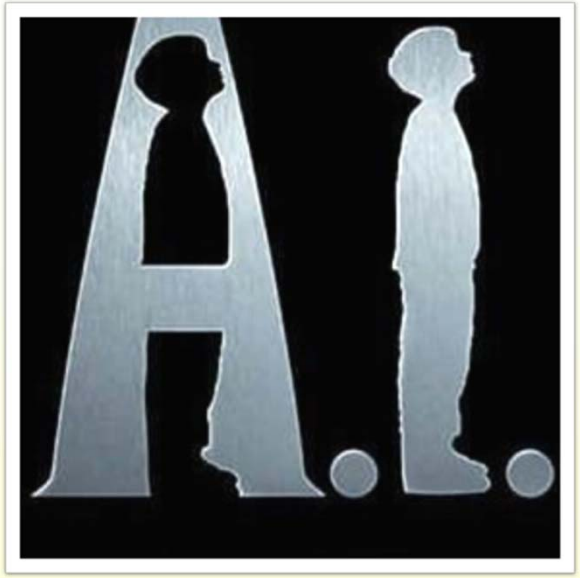
شما در یک سوی دیوار با سوی دیگر دیوار به صورت نوشتاری صحبت میکنید حال آنکه در آنسوی دیوار نه انسانی دیگر بلکه یک ماشین قرار گرفته است.

- آیا ماشین می‌تواند از انسان چنان تقلید کند که در یک آزمون محاوره‌ای نتوانیم تفاوت انسان و ماشین را تشخیص دهیم؟
- تورینگ نتوانست پاسخ قطعی این پرسش ها را پیدا کند.

تاریخچه

- در سال ۱۹۵۶ جان مک کارتی، یکی از نظریه پردازان پیشگام آن زمان، اصطلاح هوشمند مصنوعی را برای اولین بار در نخستین کنفرانسی که به این موضوع اختصاص یافته بود، به کار برد.
- دانشمندان بعدا این تاریخ را به عنوان تاریخ تولد علم هوش مصنوعی انتخاب کردند.
- تقریبا در همان زمان جان فون نیومان نظریه بازی‌ها را معرفی کرد. این نظریه نقش موثری در پیشبرد جنبه‌های نظری و علمی هوش مصنوعی داشت.
- مک‌کارتی در آن زمان معتقد بود که می‌توان کاری کرد که ماشین نیز هوشی همانند هوش انسانی داشته باشد.
- در سال ۱۹۶۸ آرتور سی کلارک، در رمان معروف خود یعنی اودیسه فضایی ۲۰۰۱ اصطلاح آزمون تورینگ را به جای بازی تقلید سر زبان‌ها انداخت.
- همه کسانی که نخستین گام‌ها را در راه معرفی هوش مصنوعی برداشتند، یک هدف را در سرداشتند و آن *رساندن سطح هوش ماشینی به سطح هوش انسانی* بود.

هوش مصنوعی



- هنوز تعریف دقیقی که مورد قبول همه دانشمندان این علم باشد برای هوش مصنوعی ارائه نشده‌است.
- هوش مصنوعی عبارت است از مطالعه این که چگونه کامپیوترها را می‌توان وادار به کارهایی کرد که در حال حاضر انسان‌ها آنها را بهتر انجام می‌دهند.
- هوش مصنوعی، شاخه‌ایست از علم کامپیوتر که ملزومات محاسباتی اعمالی همچون ادراک (Perception)، استدلال (reasoning) و یادگیری (learning) را بررسی کرده و سیستمی جهت انجام چنین اعمالی ارائه می‌دهد.
- هوش مصنوعی، مطالعه روش‌هایی است برای تبدیل کامپیوتر به ماشینی که بتواند اعمال انجام شده توسط انسان را انجام دهد

هوش مصنوعی

- هوش مصنوعی علم و مهندسی ایجاد ماشینهایی باهوش با به کارگیری از کامپیوتر و الگوگیری از درک هوش انسانی و نهایتا دستیابی به مکانیزم هوش مصنوعی در سطح هوش انسانی میباشد.
- شیوه‌ها و تکنیک‌های هوش مصنوعی، برای حل آن دسته از مسائل به وجود آمده اند که به طور سهل و آسان توسط برنامه‌نویسی تابعی (Functional programming)، یا شیوه‌های ریاضی قابل حل نبوده‌اند.
- روش‌های هوش مصنوعی به درد حوزه‌هایی می خورند که مسائل آن‌ها به‌خوبی تعریف نمی‌شوند.
- هوش مصنوعی که همواره هدف نهایی دانش رایانه بوده‌است و اکنون نیز در خدمت توسعه علوم رایانه است.

چالشهای بنیادین هوش مصنوعی

- مهمترین مشخصه هوشمندی در آزمون تورینگ توانایی پردازش و درک زبان طبیعی است.
- آیا صرف این‌که ماشینی بتواند نحوه صحبت کردن انسان را شبیه‌سازی کند، به معنی آن است که هوشمند است؟

- انتقادات دیگری نیز به آزمون تورینگ وارد می‌شود. ازجمله این‌که
  ممکن– است یک ماشین هوشمند باشد، ولی نتواند همچون انسان ارتباط برقرار کند
  دانش– پیش‌زمینه یا آرشیو ذهنی یک موجود هوشمند نقش مؤثری در هوشمندی او بازی می‌کند.

هوشمندی چیست؟

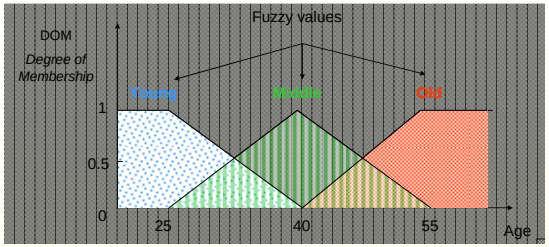
چالشهای بنیادین هوش مصنوعی

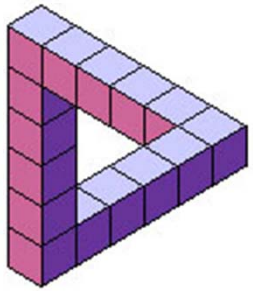
- یکی از مشهورترین انتقادات در این زمینه را فیلسوفی به نام جان سیرل (John Searle) مطرح کرده است.

- سیرل ابتدا نقد خود درباره هوش ماشینی را در ۱۹۸۰ مطرح کرد و سپس آن در مقاله کامل‌تری که در ۱۹۹۰ منتشر کرد، بسط داد.

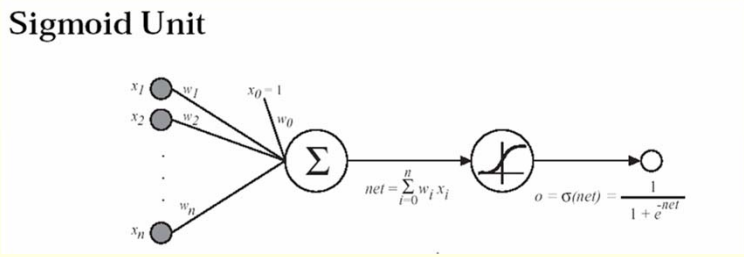
- او معتقد است بحث هوشمندی ماشین‌های غیربیولوژیک اساساًبی‌ربط است.

هوش محاسباتی



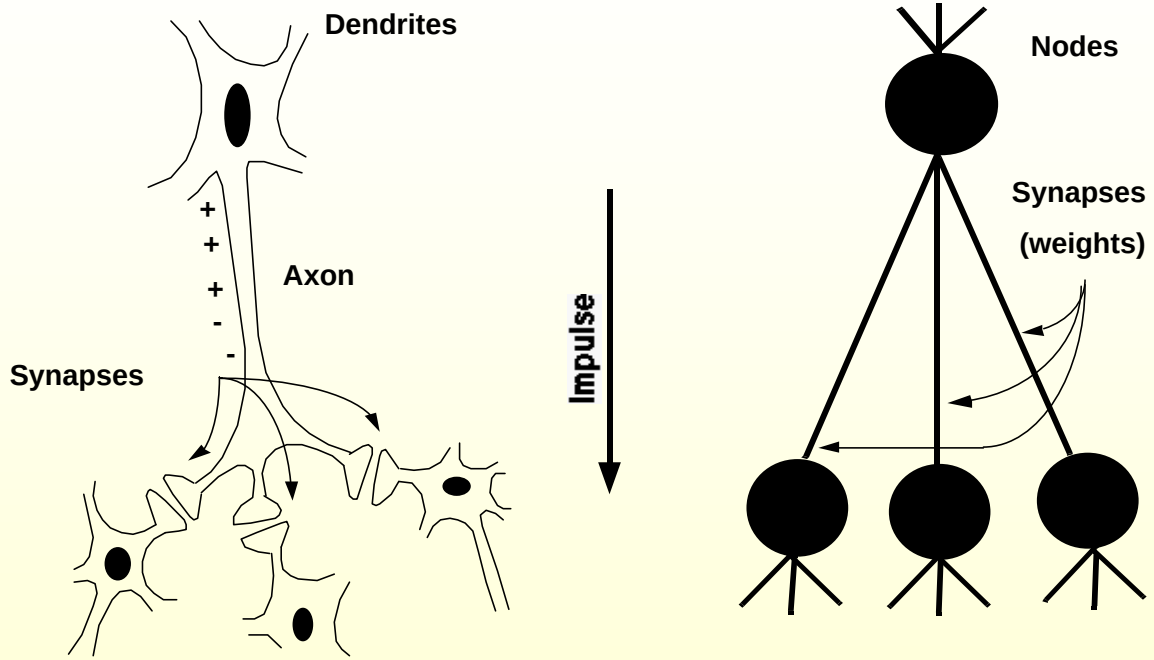


- هوش مصنوعی پیوندگرا،
- قواعد از ابتدا در اختیار سیستم قرار نمی‌گیرد، بلکه سیستم از طریق تجربه، خودش قوانین را استخراج می‌کند.



- شبکه‌های عصبی (Neural Networks) و نیز به‌کارگیری منطق فازی (Fuzzy Logic) و الگوریتم ژنیتک (GA) که با استفاده از ایده تکامل داروینی و انتخاب طبیعی پیشنهاد شده، در این دسته قرار می‌گیرند.

Neural Networks شبکه های عصبی

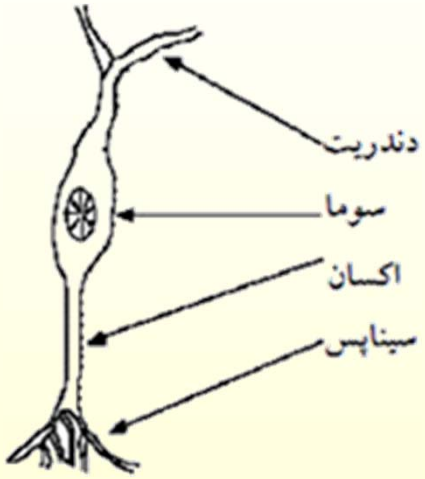


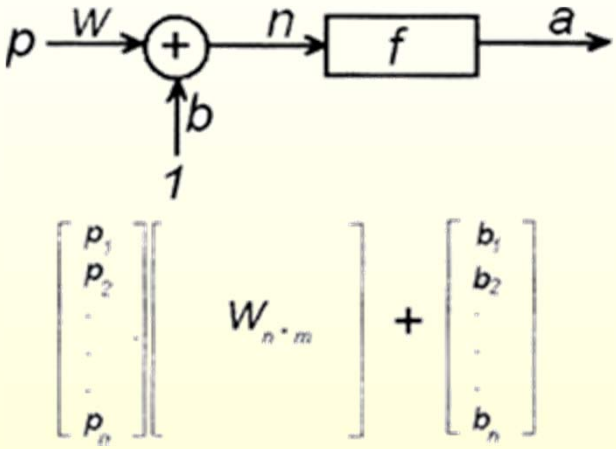
- شبکه‌های عصبی را می‌توان با اغماض زیاد، مدل‌های الکترونیکی از ساختار عصبی مغز انسان نامید.
- مکانیسم فراگیری و آموزش مغز اساساًبر تجربه استوار است.

شبکه های عصبی Neural Networks

- مدل‌های الکترونیکی شبکه‌های عصبی طبیعی نیز بر اساس همین الگو بنا شده‌اند و روش برخورد چنین مدل‌هایی با مسائل، با روش‌های محاسباتی که به‌طور معمول توسط سیستم‌های کامپیوتری در پیش گرفته شده‌اند، تفاوت دارد.
- شبکه‌های عصبی شبیه‌سازی شده یا کامپیوتری، فقط قادرند تا بخش کوچکی از خصوصیات و ویژگی‌های شبکه‌های عصبی بیولوژیک را شبیه‌سازی کنند.
- در حقیقت، هدف از ایجاد یک شبکه عصبی نرم‌افزاری، بیش از آنکه شبیه‌سازی مغز انسان باشد، ایجاد مکانیسمی برای حل مسائل مهندسی با الهام از الگوی رفتاری شبکه‌های بیولوژیک است.
- در شبکه‌های عصبی بیولوژیک، نرون‌ها در ساختاری سه بعدی به یکدیگر اتصال یافته‌اند.
- اتصالات بین نرون‌ها در شبکه‌های عصبی بیولوژیک آنقدر زیاد و پیچیده است که به هیچ وجه نمی‌توان شبکه مصنوعی مشابهی طراحی کرد.
- تکنولوژی مدارات مجتمع امروزی به ما امکان می‌دهد که شبکه‌های عصبی را در ساختارهای دو بعدی طراحی کنیم.

مدل ریاضی شبکه های عصبی





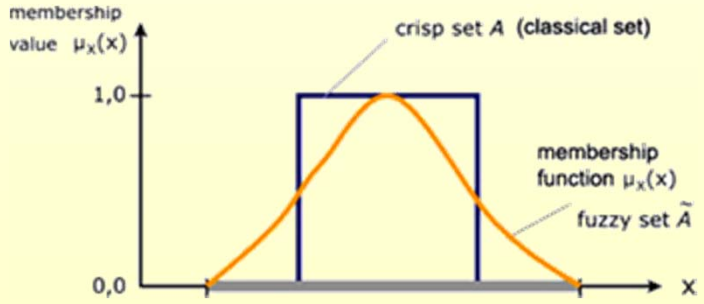
$$\begin{bmatrix} p_1 \\ p_2 \\ \cdot \\ \cdot \\ \cdot \\ p_n \end{bmatrix} \begin{bmatrix} \quad \\ W_{n \cdot m} \\ \quad \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

در جلسات آینده درس به طور مفصل بحث خواهد شد.

منطق فازی Fuzzy logic



- تئوری مجموعه‌های فازی و منطق فازی را اولین بار پرفسور لطفی‌زاده در سال ۱۹۶۵ معرفی نمود.
- منطق فازی به سیستم‌هایی اشاره دارد که به جای مقادیر "درست" و "نادرست" که در محیط‌های دیجیتال طبیعی‌ترند، می‌توانند با سطوح متغیر قطعیت کار کنند.
- تئوری مجموعه‌های فازی مفهوم عضویت باینری عناصر را بسط می‌دهد و عضویت درجه‌بندی شده را مطرح می‌کند.



منطق فازی Fuzzy logic

- جالب‌ترین کاربرد منطق فازی، تفسیری است که این علم از ساختار تصمیم‌گیری‌های موجودات هوشمند، و در راس آن‌ها، هوش انسانی به دست می‌دهد.
- شاید یکی از جالب‌ترین کاربردهای منطق فازی هوش مصنوعی در بازی‌های رایانه‌ای و جلوه‌های ویژه سینمایی باشد.
- منطق فازی در هوشمند ساختن روبات‌های سخت‌افزاری نیز کاربردهای زیادی دارد.

در جلسات آینده درس به طور مفصل بحث خواهد شد.

الگوریتم‌های ژنتیک GA

- الگوریتم‌های ژنتیک از اصول انتخاب طبیعی داروین برای یافتن فرمول بهینه جهت پیش‌بینی یا تطبیق الگو استفاده می‌کنند.
- در حقیقت بدین روش می توانیم در فضای حالت مسئله حرکتی سریع‌تر برای یافتن جواب‌های احتمالی داشته باشیم؛ یعنی می توانیم با عدم بسط دادن کلیه حالات، به جواب‌های مورد نظر برسیم.
- الگوریتم‌های ژنتیک الگوریتم‌هایی هستند که دارای قدرت بسیار زیادی در یافتن جواب مسئله هستند، اما باید توجه داشت که شاید بتوان کاربرد اصلی این الگوریتم ها را در مسائلی در نظر گرفت که دارای فضای حالت بسیار بزرگ هستند و عملاًبررسی همه حالت‌ها برای انسان در زمان‌های نرمال (در حد عمر بشر) ممکن نیست.
- از طرفی باید توجه داشت که حتماً بین حالات مختلف مسئله باید دارای پیوستگی مناسب و منطقی باشیم.

- در جلسات آینده درس به طور مفصل بحث خواهد شد.

کاربردهای شبکه های عصبی در پزشکی

- شبکه های عصبی برای مشکلات بیو مدیکال برنامه های کاربردی گسترده ای در آینده ارائه خواهند کرد و در حال حاضر به طور موفقیت آمیز در زمینه های مختلف زیر به کار رفته اند.
- سیستمهای تشخیص
- آنالیز های شیمی-پزشکی
- آنالیز تصویر
- پیشرفتهای دارویی

سیستمهای تشخیص

- به طور عادی برای کشف سرطان و مشکلات قلبی کاربرد دارند. مزایای استفاده از ANN ها این است که تحت تاثیر عوامل خستگی ،شرایط کاری و موقعیت های عاطفی قرار نمیگیرند.

- تشخیص سرطان سینه
- بررسی سیگنال EEG برای تشخیص مراحل خواب
- تشخیص بیماری سل
- تشخیص هوشمند بیماری دیابت

آنالیز های شیمی-پزشکی

- در پزشکی برای آزمایش خون،
- نمونه ادرار ،
- سطوح شیارهای گلوکز در دیابتیک ها،
- تعیین سطوح در مایعات بدن ،
- تعیین شرایط پاتولوژی مثل سل

آنالیز تصویر

- در آنالیز تصاویر پزشکی با کیفیت های متفاوت و متنوع به کار گرفته میشود. برنامه های کاربردی در این زمینه شامل
- کشف تومور در Veltra-Sonogram ،
- دسته بندی X-Ray های سینه ،
- دسته بندی بافت و ماهیچه در MRI
- تعیین شکل استخوان بندی از عکسهای X-Ray
- تعیین بلوغ مغزی

پیشرفتهای دارویی و مدلسازی سیستم قلبی عروقی

- توسعه داروها برای درمان سرطان و ایدز
- پروسه مدل کردن بیومولکول ها

- مدل کردن آزمایشی سیستم قلبی- عروقی انسان. تشخیص می تواند به وسیله ساختن یک مدل از سیستم قلبی- عروقی یک شخص منحصر به فرد و مقایسه آن با داده های پزشکی که از بیمار گرفته شده بدست آید.
- مزایای چنین سیستم هایی میتواند چک شدن سریع و بدون درد بیماری های قلبی باشد. بنابراین بیماری در مراحل اولیه تشخیص داده میشود. البته در این سیستم نیاز به پزشکان رد نمیشود.
- آمیزش سنسورها در این روش ما را قادر میسازد تا روابط پیچیده میان مقادیر سنسورهای مجزا یاد گرفته شود (در صورت تحلیل شدن به صورت مجزا از بین خواهند رفت)

بویایی الکترونیکی

- به طور آزمایشی برای پیاده سازی بویایی الکترونیکی استفاده شده اند.
- بویایی الکترونیکی ظرفیت زیادی برای برنامه های کاربردی در پزشکی از راه دور دارد. بویایی الکترونیکی میتواند رایحه را در محیط های جراحی متحرک تشخیص دهد. این بوی تشخیص داده شده به صورت الکترونیکی به قسمت دیگر منتقل میشود تا به وسیله سیستم های باز تولید بو بازسازی شود. با توجه به این که حس بویایی اهمیت زیادی برای جراحی دارد بویایی از راه دور حایز اهمیت است.

شناخت الگو

- پاتولوژی یک تکنیک تصویرگری است که با طبیعت بیماری ها (تغییرات ساختاری و عملکردی در بافت ها) در ارتباط میباشد. احتیاج آن به رنگ و کیفیت، استفاده از تکنولوژی عکسهای دیجیتالی را برای اجرا دشوار میسازد.
- شناخت الگو یک ایده برای دسته بندی دادهای ورودی به کلاس های قابل شناسایی بوسیله خصوصیت مهم داده است. که این مشخصات این طرح از جزئیات بی ربط گرفته میشود.
- دلیل استفاده از شبکه های عصبی در شناخت الگو به خاطر توانایی آنها در یادگیری و ذخیره دانش است.

پزشک نمونه

- یک برنامه کاربردی که در اواسط سال ۱۹۸۰ تولید شد" یک نمونه پزشک "نامیده شد.
- به عنوان یک کمک اتوماتیک برای حافظه از شبکه های عصبی برای ذخیره تعداد زیادی از پروندهای پزشکی ،که هر کدام از آنها شامل اطلاعاتی از علائم ،تشخیص ها و درمان برای یک مورد بخصوص بود، استفاده شد.
- بعد از آموزش، شبکه میتوانست با یک مجموعه از علائم بهترین تشخیص و درمان را ارائه دهد.

کاربردهای سیستمهای فازی در پزشکی

- استخراج قوانین فازی از ثبت های پلتیموگرافی برای طبقه بندی خواب نوزاد
- ارائه فیلتر جدیدی مبنتی بر کنترل فازی برای بهسازی تصاویر MRI
- تشخیص بافت های سه بعدی غیر عادی در تصاویر پزشکی (CT Scanو MRI) توسط منطق فازی
- جداساز میکروکالسیفیکاسیونهای مشکوک در تصاویر دیجیتال ماموگرافی بوسیله کاربرد منطق فازی
- سیستمهای فازی در بیو انفورماتیک
- فازی ژنتیک

سیستمهای فازی در بیو انفورماتیک

- مطالعه تفاوت های بین پلی نوکلئیدها
- آنالیز اطلاعات تجربی با استفاده از تئوری انطباقی رزونانس فازی
- تنظیم توالی ها بر پایه الگوریتم برنامه نویسی دینامیک در قالب فازی
- ترتیب گذاری DNA با استفاده از سیستم ژنتیکی فازی
- جمع آوری ژن ها بوسیله اطلاعات از ریز آرایه ها
- تخمین محل قرارگیری پروتئین های زیر سلولی از ترکیبات دی پپتیدی با استفاده از الگوریتم فازی k-nearst neighbors
- شبیه سازی ویژگی های پیچیده تحت تاثیر با ژن بوسیله تاثیر فازی در جمعیت
- نسبت دادن مقادیر عضویت گروهی به ژن ها با استفاده از متود c-mean فازی

سیستمهای فازی ژنتیک

- در سالهای اخیر، دانشمندان زیادی استفاده از الگوریتمهای ژنتیک را بعنوان وسیله ای برای طراحی سیستمهای فازی کشف کردند. سیستمهای فازی ژنتیک بر روی محاسبات تکاملی و منطق فازی بحث می کند. برای دانشمندان و مهندسان از نظر تحقیق و بررسی بر روی کاربردهای موجود در حوزه سیستم فازی و الگوریتم ژنتیک زمینه های گسترده ای موجود است

جلسه دوم

شبکه های عصبی

استاد: محمد باقر منهاج

موضوعات

- **فصل اول**
  - مقدمه
  - معنای شبکه های عصبی
  - انگیزه های بیولوژیکی
  - تشابهات و انتظارات
  - تاریخچه و کاربرد
- **فصل دوم**
  - مدل ریاضی نرون
  - مدل تک ورودی و چند ورودی
  - ساختار شبکه های عصبی: شبکه های یک و چند لایه

فصل ۱

مقدمات

مقدمه

- وجود مسائلی بدون راه حل یا به سختی قابل حل
  - حرکت از تحقیقات صرفا تئوری به تحقیقات کاربردی
- توسعه تئوریک سیستمهای دینامیکی هوشمند مدل آزاد مبتنی بر داده های تجربی

  شبکه های عصبی جزو این دسته از سیستمهای دینامیکی
- استخراج دانش نهفته در داده ها
- به این سیستم‌ها هوشمند گویند ، چرا که بر اساس محاسبات روی داده های عددی یا مثال ها ، قوانین کلی را فرا می گیرند. این سیستم ها در مدلسازی ساختار نرو-سیناپتیکی مغز بشر می کوشند.
- عدم کفایت دانش موجود بشر از فیزیولوژی عصبی
  - مغز بشر دست نیافتنی است
- هنگام صحبت از شبکه های عصبی مصنوعی تعیین حدود انتظارات، امکانات و شباهتها ضروری است

معنای شبکه های عصبی

- مغز سیستمی پیچیده دارای پردازش اطلاعات با ساختار موازی
- ۲ درصد وزن بدن، مصرف بیش از ۲۰ درصد کل اکسیژن بدن
- جمع آوری و محاسبه حجم عظیمی از اطلاعات و سیگنالها هنگام فهم این مطالب در جریان است
- محاسبات مغز در ساختاری کاملا مغایر با ساختار کامپیوترهای امروزی
- مغز اجتماعی از نرونها
- هر نرون بیولوژیکی دارای
  - پیچیدگی یک میکروپروسسور
  - سرعت محاسباتی به مراتب کمتر از میکروپروسسور
  - امکان ذخیره اطلاعات در خود و ارتباطات میان نرونها
- یادگیری در واقع ایجاد ارتباطات جدید میان نرونها و تنظیم ارتباطات موجود

معنای شبکه های عصبی

- نرونهای شبکه مصنوعی
  - جزیی از یک برنامه کامپیوتری یا تراشه های نیمه هادی
  - سرعتی بسیار بالاتر از نرون بیولوژیکی (۱,۰۰۰,۰۰۰ برابر)
  - ولی فقط دارای کسری از توانایی بالای نرونهای بیولوژیکی

چرا؟؟؟؟

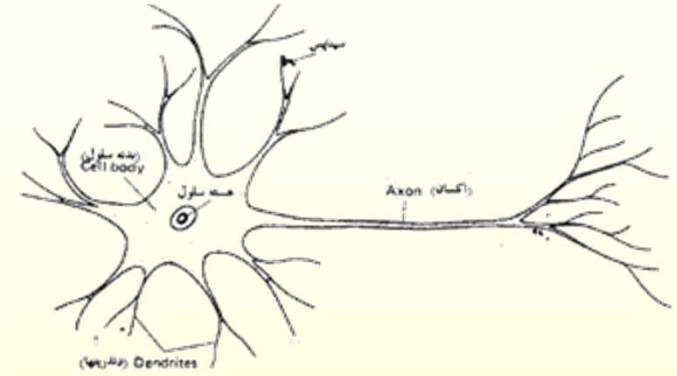
- در این درس
  - آشنایی با شبکه ای کوچک از نرونهای مصنوعی ساده که برای حل مسائل پیچیده آموزش پذیر هستند
  - مسئله پیچیده = یادگیری نگاشتها
  - شبکه های عصبی تقریب زنهای جهانی توابع هستند.

انگیزه های بیولوژیکی

- مغز دارای 10^{11} نرون با 10^{16} ارتباط میان آنها
- تعداد ارتباطات به مراتب کمتراز تعداد

یالهای گراف مرتبه کامل

- اجزای نرون
- نرونهای حسی، محرک و ارتباطی
- نکات مهم در محاسبات نرونی
  - ترتیب نرونها و شدت سیناپسهای میان نرونها
  - ساختارهای عصبی در خلال یادگیری به وجود آمده و از بین می روند



شکل ۱-۱. نواحی اصلی یک سلول عصبی بیولوژیک

تشابهات

- بلوکهای ساختاری در هر دو شبکه مصنوعی و بیولوژیکی دستگاههای محاسباتی خیلی ساده ای هستند.
  - نرونهای مصنوعی از سادگی بیشتری برخوردارند.
- ارتباطهای بین نورونها عملکرد شبکه را تعیین می کند.
- دارای ساختار موازی

هدف اصلی این درس تعیین ارتباطهای مناسب جهت حل مسائل مشخص می باشد.

انتظارات

۱) قابلیت یادگیری

- شبکه عصبی سیستم کاملا پیچیده و غیرخطی
- توزیع خاصیت پردازش غیرخطی در کل شبکه
- چنین سیستمی بدون قابلیت یادگیری با افزوده شدن یک مثال جدید توانایی خود را از دست می دهد.

- قابلیت یادگیری یعنی توانایی تنظیم پارامترهای شبکه (وزنهای سیناپتیکی) در مسیر زمان که محیط شبکه تغییر می کند و شبکه شرایط جدید را تجربه می کند، با این هدف که شبکه بتواند با آموزش مختصر برای شرایط جدید نیز کارآمد باشد.

انتظارات

۲) پراکندگی اطلاعات

- پردازش اطلاعات به صورت متن
- هر نرون متاثر از فعالیت سایر نرونها (هر وزن مربوط به همه ورودیها)
- اگر بخشی از نرونهای شبکه حذف شوند یا عملکرد غلط داشته باشند, باز هم احتمال رسیدن به پاسخ صحیح وجود دارد.

۳) قابلیت تعمیم

- آموزش شبکه با مثالهای اولیه
- ارائه خروجی مناسب در مقابل ورودیهای آموزش داده نشده
- فرایند درونیابی
- شبکه تابع را یاد می گیرد، الگوریتم را می آموزد و یا رابطه تحلیلی مناسبی برای نقاطی در فضا به دست می آورد.

انتظارات

۴) پردازش موازی

- پاسخ همزمان سلولهای قرار گرفته در یک تراز به ورودیهای آن تراز
- افزایش سرعت پردازش
- توزیع وظیفه پردازش بین پردازنده های کوچکتر مستقل

۵) مقاوم بودن

- رفتار مستقل هر سلول
- تصحیح خطاهای محلی در یک روند همکاری
- افزایش قابلیت مقاوم بودن (تحمل پذیری خطا) در سیستم

تاریخچه

- دیدگاه جدید شبکه های عصبی در دهه ۴۰ با استفاده از یک مدل منطقی توسط مک کلوث و والتر پیتز انجام شد.
- دونالد هب، ادامه راه پاولف در شرط گذاری کلاسیک به عنوان خواص نرونها
- در ۱۹۵۸ شبکه پرسپترون توسط روزنبلات معرفی گردید. این شبکه قادر به شناسایی الگوها بود.
- در ۱۹۶۰ توسط ویدرو (دانشگاه استنفورد) شبکه عصبی تطبیقی خطی آدالاین با قانون یادگیری جدید بوجود آمد که از لحاظ ساختار شبیه شبکه پرسپترون بود.
- در ۱۹۶۹ مینسکی و پاپرت کتابی نوشتند که محدودیتهای سیستمهای تک لایه و چند لایه پرسپترون را تشریح کردند
- در ۱۹۷۲، تئو کوهونن و جیمز اندرسون به طور مستقل شبکه های عصبی جدیدی با کارکرد عناصر ذخیره ساز مطرح شدند.

تاریخچه

- دهه ۸۰ رشد تکنولوژی میکروپروسسورها و روند صعودی تحقیقات روی شبکه عصبی
- ایده شبکه های بازگشتی با جان هاپفیلد ۱۹۸۲
- الگوریتم پس انتشار خطا توسط راملهارت و مکلند ۱۹۸۶
-

جایگاه شبکه های عصبی در آینده؟؟؟؟؟

کاربردها

- طبقه بندی، شناسایی و تشخیص الگو
- پردازش سیگنال
- پیش بینی سریهای زمانی
- مدلسازی و کنترل
- بهینه سازی
- سیستمهای خبره و فازی
- مسائل مالی و بیمه
- ساخت وسایل صنعتی و امور حمل ونقل
- پزشکی

کاربردهای شبکه های عصبی در پزشکی

- شبکه های عصبی برای مشکلات بیو مدیکال برنامه های کاربردی گسترده ای در آینده ارائه خواهند کرد و در حال حاضر به طور موفقیت آمیز در زمینه های مختلف زیر به کار رفته اند.
- سیستمهای تشخیص
- آنالیز های شیمی-پزشکی
- آنالیز تصویر
- پیشرفتهای دارویی

سیستمهای تشخیص

- به طور عادی برای کشف سرطان و مشکلات قلبی کاربرد دارند. مزایای استفاده از ANN ها این است که تحت تاثیر عوامل خستگی ،شرایط کاری و موقعیت های عاطفی قرار نمیگیرند.

- تشخیص سرطان سینه
- بررسی سیگنال EEG برای تشخیص مراحل خواب
- تشخیص بیماری سل
- تشخیص هوشمند بیماری دیابت

آنالیز های شیمی-پزشکی

- در پزشکی برای آزمایش خون،
- نمونه ادرار ،
- سطوح شیارهای گلوکز در دیابتیک ها،
- تعیین سطوح در مایعات بدن ،
- تعیین شرایط پاتولوژی مثل سل

آنالیز تصویر

- در آنالیز تصاویر پزشکی با کیفیت های متفاوت و متنوع به کار گرفته میشود. برنامه های کاربردی در این زمینه شامل
- کشف تومور در Veltra-Sonogram ،
- دسته بندی X-Ray های سینه ،
- دسته بندی بافت و ماهیچه در MRI
- تعیین شکل استخوان بندی از عکسهای X-Ray
- تعیین بلوغ مغزی

پیشرفتهای دارویی و مدلسازی سیستم قلبی عروقی

- توسعه داروها برای درمان سرطان و ایدز
- پروسه مدل کردن بیومولکول ها

- مدل کردن آزمایشی سیستم قلبی- عروقی انسان. تشخیص می تواند به وسیله ساختن یک مدل از سیستم قلبی- عروقی یک شخص منحصر به فرد و مقایسه آن با داده های پزشکی که از بیمار گرفته شده بدست آید.
- مزایای چنین سیستم هایی میتواند چک شدن سریع و بدون درد بیماری های قلبی باشد. بنابراین بیماری در مراحل اولیه تشخیص داده میشود. البته در این سیستم نیاز به پزشکان رد نمیشود.
- آمیزش سنسورها در این روش ما را قادر میسازد تا روابط پیچیده میان مقادیر سنسورهای مجزا یاد گرفته شود (در صورت تحلیل شدن به صورت مجزا از بین خواهند رفت)

بویایی الکترونیکی

- به طور آزمایشی برای پیاده سازی بویایی الکترونیکی استفاده شده اند.
- بویایی الکترونیکی ظرفیت زیادی برای برنامه های کاربردی در پزشکی از راه دور دارد. بویایی الکترونیکی میتواند رایحه را در محیط های جراحی متحرک تشخیص دهد. این بوی تشخیص داده شده به صورت الکترونیکی به قسمت دیگر منتقل میشود تا به وسیله سیستم های باز تولید بو بازسازی شود. با توجه به این که حس بویایی اهمیت زیادی برای جراحی دارد بویایی از راه دور حایز اهمیت است.

شناخت الگو

- پاتولوژی یک تکنیک تصویرگری است که با طبیعت بیماری ها (تغییرات ساختاری و عملکردی در بافت ها) در ارتباط میباشد. احتیاج آن به رنگ و کیفیت، استفاده از تکنولوژی عکسهای دیجیتالی را برای اجرا دشوار میسازد.
- شناخت الگو یک ایده برای دسته بندی دادهای ورودی به کلاس های قابل شناسایی بوسیله خصوصیت مهم داده است. که این مشخصات این طرح از جزئیات بی ربط گرفته میشود.
- دلیل استفاده از شبکه های عصبی در شناخت الگو به خاطر توانایی آنها در یادگیری و ذخیره دانش است.

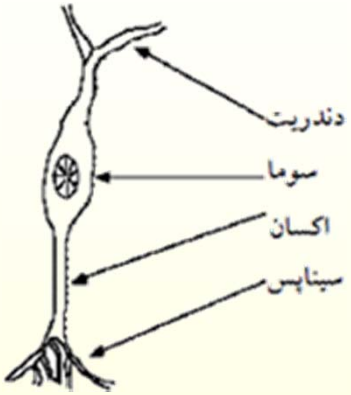
پزشک نمونه

- یک برنامه کاربردی که در اواسط سال ۱۹۸۰ تولید شد" یک نمونه پزشک "نامیده شد.
- به عنوان یک کمک اتوماتیک برای حافظه از شبکه های عصبی برای ذخیره تعداد زیادی از پروندهای پزشکی ،که هر کدام از آنها شامل اطلاعاتی از علائم ،تشخیص ها و درمان برای یک مورد بخصوص بود، استفاده شد.
- بعد از آموزش، شبکه میتوانست با یک مجموعه از علائم بهترین تشخیص و درمان را ارائه دهد.

فصل ۲

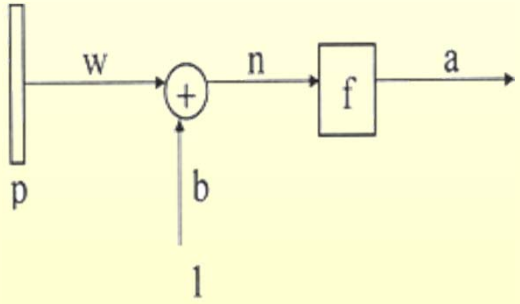
مدل ریاضی نرون

مدل تک ورودی



- w شدت سیناپس
- ورودی خالص n
- تابع تحریک f
- a سیگنال گذرنده از اکسون

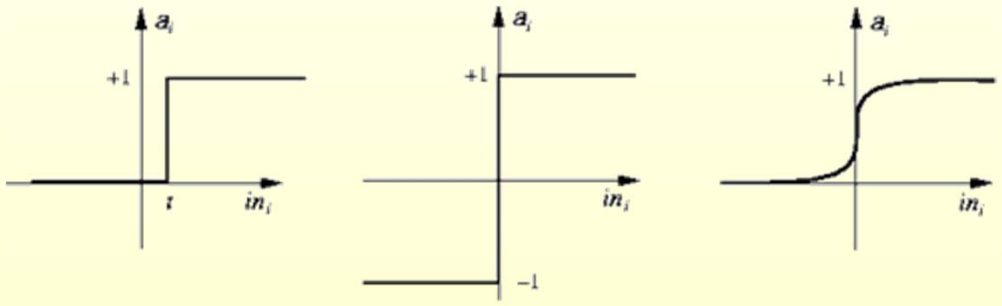
$$a = f(WP + b)$$



شکل (۲-۱): مدل نرون تک ورودی

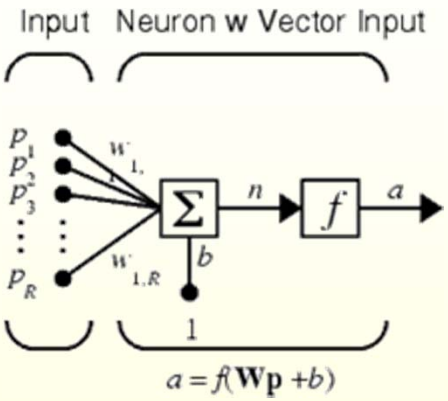
برخی توابع تحریک مرسوم نرون مصنوعی

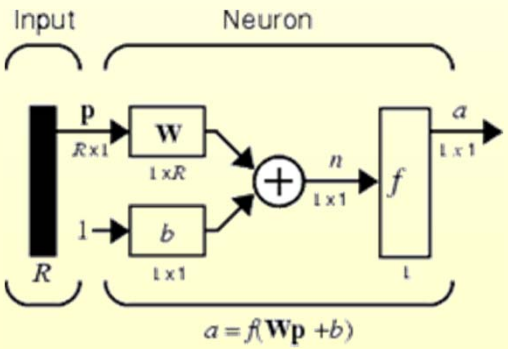
- خطی
- دو مقداره حدی
- زیگموئیدی



(a) Step function (b) Sign function (c) Sigmoid function

مدل چند ورودی





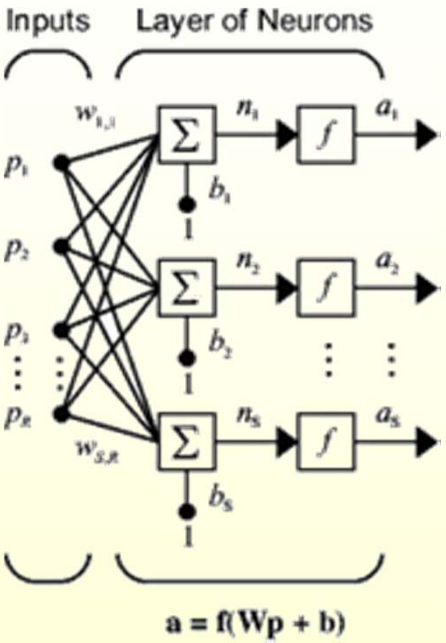
- مدل کردن به فرم ماتریسی
- اندیس اول شماره خود نرون
- اندیس دوم نشان دهنده مبدا سیگنال ورودی نرون
- مثلا W_{12} شدت سیناپس دومین عنصر ورودی به نرون اول را نشان میدهد.
- W در حالت کلی، هر سطر ماتریس با یک نرون است.

$$\underline{p}=[p_1, p_2, ..., p_R]^T$$

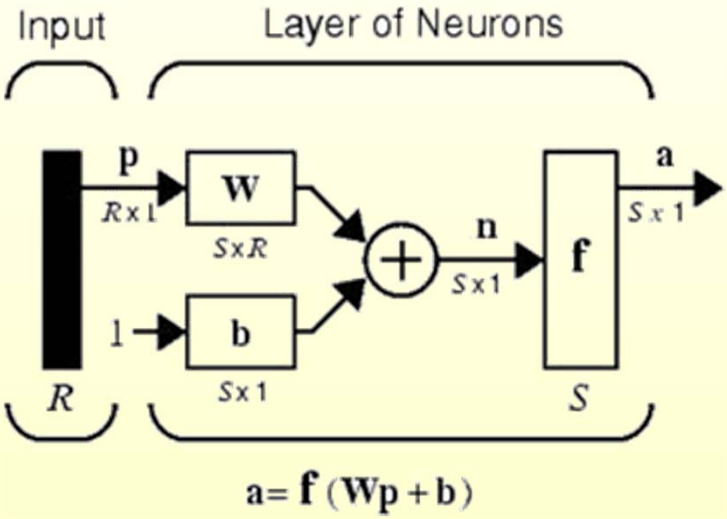
$$W=[w_{11}, w_{12}, ..., w_{1R}]$$

ساختار شبکه های عصبی

- **شبکه تک لایه با S نرون و R ورودی**
  - مدل شبکه و فرم ماتریسی آن
  - ماتریس W دارای S سطر و R ستون



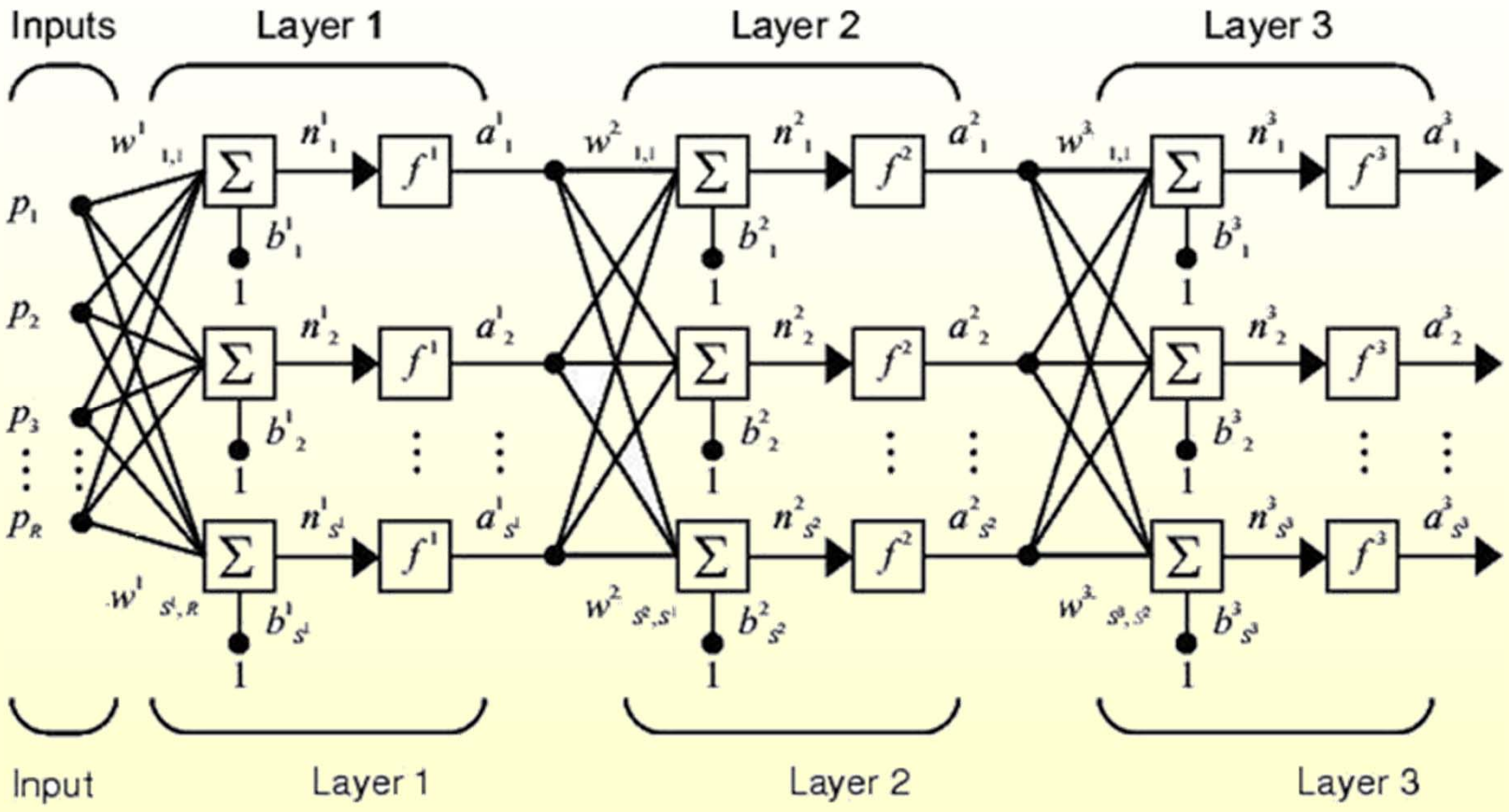
$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ & & & \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$



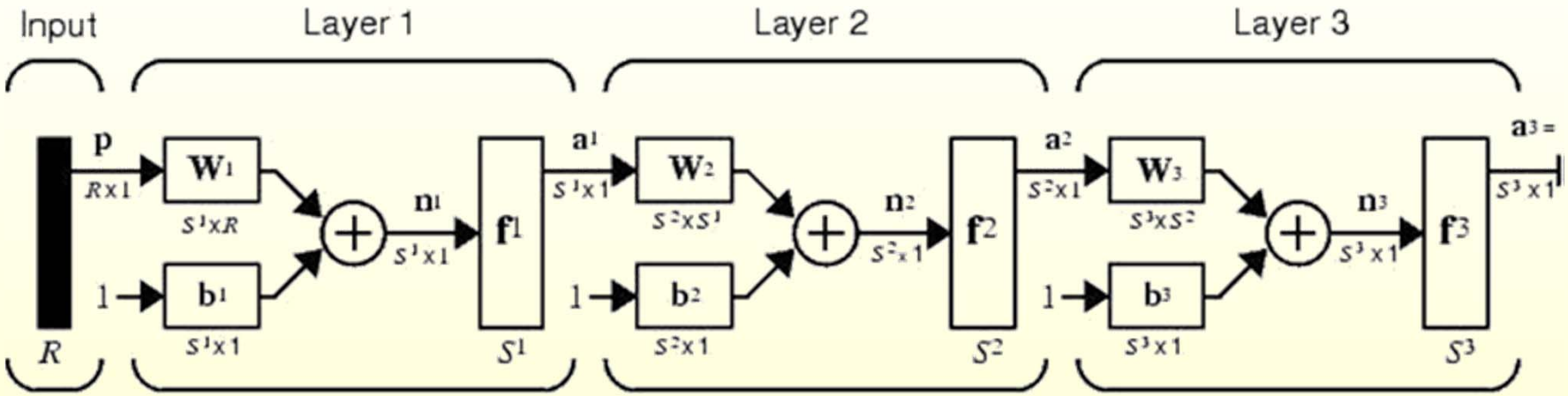
شبکه های چند لایه

- در شبکه تک لایه بردار ورودی توسط نرونهای لایه با رابطه $a=f(W\underline{p}+b)$ به بردار خروجی مرتبط می شوند.
- در شبکه های چند لایه، خروجی هر لایه به عنوان بردار ورودی برای لایه بعدی محسوب می شود.
- W^1، W^2 ،... به ترتیب ماتریس وزن لایه اول، دوم و .... را نشان می دهد.
- S^1، S^2 ،... به ترتیب تعداد نرونهای لایه اول، دوم و .... را نشان می دهد.
- f^1، f^2 ،... به ترتیب توابع تحریک لایه اول، دوم و .... را نشان می دهد.
- $\underline{n}^1$، $\underline{n}^2$ ،... به ترتیب بردار ورودی خالص لایه اول، دوم و .... را نشان می دهد.
- $\underline{a}^1$، $\underline{a}^2$ ،... به ترتیب بردار خروجی لایه اول، دوم و .... را نشان می دهد.
- $\underline{b}^1$، $\underline{b}^2$ ،... به ترتیب بردار بایاس لایه اول، دوم و .... را نشان می دهد.

مثال: یک شبکه پیشخور ۳ لایه



مثال: یک شبکه پیشخور ۳ لایه



$$a^1 = f^1(W^1 p + b^1) \qquad a^2 = f^2(W^2 a^1 + b^2) \qquad a^3 = f^3(W^3 a^2 + b^3)$$

$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3) = y$$

چند سوال

۱) تعداد عناصر بردار ورودی (R) چگونه مشخص می شود؟

۲) یک نرون چه نوع تابع ورودی-خروجی را نمایندگی می کند؟

۳) آیا می توان بردار ورودی P را به عنوان یک لایه در نظر گرفت؟

۴) آیا تعداد عناصر بردار ورودی R و تعداد نرونهای S می توانند با هم برابر نباشند؟

۵) آیا همه نرونهای موجود در یک لایه باید دارای توابع تحریک یکسان باشند؟

۶) آیا می توان در لایه های مختلف، توابع تبدیل مختلف داشت؟

شبکه های عصبی

استاد: محمد باقر منهاج

موضوعات

- **فصل دوم (ادامه)**
  - شبکه های پسخور یا برگشتی
  - مسائل حل شده فصل ۲ (خوانده شود و بحث در کلاس حل تمرین)
- **فصل سوم: مسئله تشریحی شناسایی الگو**
  - شناسایی الگو
  - معرفی سه شبکه های نمونه: پرسپترون، همینگ و هاپفیلد
  - استفاده از آن در حل مسئله فوق

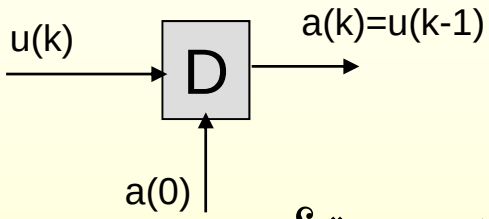
فصل ۳

مسئله تشریحی

(شناسایی الگو)

شبکه های پسخور یا برگشتی (ادامه فصل ۲)

- تفاوت با شبکه های پیشخور:
  - وجود حداقل یک سیگنال برگشتی از یک نرون به همان نرون یا نرونهای همان لایه و یا لایه های قبل
- استفاده از بلوک تاخیر D (در شبکه های گسسته)



- سوال: بلوک متناظر با عنصر تاخیر در شبکه های پیوسته چیست؟
- نکته:

شبکه های پسخور از توانایی بالقوه بیشتری نسبت به شبکه های پیشخور برخوردارند و بهتر می توانند رفتار مربوط به ویژگیهای زمانی سیستمها را نشان دهند. (چرا؟)

- در فصل سوم کمی بیشتر با این شبکه ها آشنا می شویم. توضیحات مفصلتر در فصول ۵ و ۶ کتاب آمده است.

شناسایی الگو

- فاکتور مهم در طراحی سیستمهای اطلاعاتی
- موضوع مشترک تحقیقاتی برای اکثر رشته ها از زبان شناسی تا رشته های مهندسی
- ساختار کلی یک فرایند شناسایی الگو و تصمیم گیری:
  - چیزی یا موضوعی اتفاق افتاده است
  - اتفاق، توسط سیگنالی نمایندگی می شود.
  - سیگنال مشاهده شده با سیگنال ارسالی یکسان نیست.
  - اتخاذ تصمیم بر مبنای سیگنال مشاهده شده (نه ارسالی)
- مثال: (فرد بیمار)
  - اتفاق: بیماری
  - مشاهده کننده : پزشک
  - سیگنالها : تستهای آزمایشگاهی یا علائم بیماری
  - سیگنال آمیخته با نویز: اشکالات آزمایشی (چه در وسایل و چه در بدن فرد بیمار)
  - تشخیص بیماری بر مبنای سیگنال آمیخته با نویز

شناسایی الگو

- شناسایی الگو چیزی جز جدا سازی داده ها یا الگو های ورودی بین دستجات مختلف نیست.
- مراحل طراحی یک سیستم شناسایی الگو

– کد گذاری: بیان ویژگیهای شی مورد نظر (که از طریق اندازه گیری به دست می آیند)، توسط بردارهای ورودی

بیان هندسی: تخصیص یک نقطه در فضای چند بعدی اقلیدسی به هر شی

– استخراج شاخص: استخراج مشخصه های مهم از روی بردار های ورودی و کاهش ابعاد الگوها

عناصری از بردارهای مشخصه که در تمامی طبقات مشترکند، حذف می شوند.

– طبقه بندی الگوها: روندی برای تصمیم گیری بهینه اتوماتیک

تخصیص الگوهای ورودی به یکی از طبقاتی که فضای اقلیدسی برای تصمیم گیری به تعداد متناهی از آنها تقسیم شده است

روشهای کلاسیک و شبکه های عصبی

- روشهای کلاسیک
  - باید تمامی الگوهای یادگیری قبل از حل مسئله طبقه بندی الگو در اختیار باشند.
  - یعنی، ابتدا باید یک مدل ریاضی از مشاهدات داشت تا پس از ارزیابی مدلها بر اساس داده های واقعی طراحی انجام شود.
- شبکه های عصبی
  - مستقیما با داده های واقعی کار می کنند.
  - لذا به طراحیهای مدل آزاد یا تخمین زننده های جهانی مدل آزاد موسومند.
- سوال:

اطلاعاتی که شبکه های عصبی بر مبنای آن عمل تصمیم گیری را انجام می دهد، چگونه و در کجا ذخیره می گردد؟

صورت مسئله

- هدف: جداسازی سه نوع میوه سیب، پرتقال و گلابی در یک انبار به طور خودکار
- سنسورهای موجود برای ثبت ویژگی میوه ها
  - شکل : گرد ← ۱، بیضوی ← ۱-
  - زبری و صافی سطح : صاف ← ۱، زبر ← ۱-
  - وزن : کمتر از ۲۰۰ گرم ← ۱، بیشتر از ۲۰۰ گرم ← ۱-
- کد گذاری

نشان دادن سیب، پرتقال و گلابی به ترتیب با بردارهای $\underline{p}^1$، $\underline{p}^2$، $\underline{p}^3$ به عنوان بردارهای مرجع (مولفه بردارها به ترتیب شکل، زبری و وزن می باشد)

$$\underline{p}^1=\begin{bmatrix}1\\1\\1\end{bmatrix}\quad \underline{p}^2=\begin{bmatrix}1\\-1\\1\end{bmatrix}\quad \underline{p}^3=\begin{bmatrix}-1\\1\\1\end{bmatrix}$$

صورت مسئله

- استخراج شاخص

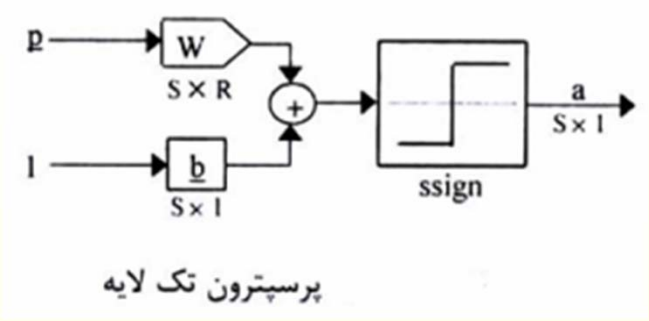
عنصر سوم در همه طبقات مشترک است. لذا با حذف این مولفه به بردارهای دو بعدی می رسیم.

$$\underline{p}^1=\begin{bmatrix}1\\1\end{bmatrix} \quad \underline{p}^2=\begin{bmatrix}1\\-1\end{bmatrix} \quad \underline{p}^3=\begin{bmatrix}-1\\1\end{bmatrix}$$

- یک مسئله ساده شناسایی الگوی باینری
  - دریافت یک بردار ورودی ۲ بعدی و تصمیم گیری درباره نوع میوه
  - معرفی سه شبکه عصبی پرسپترون (پیشخور)، همینگ (رقابتی) و هاپفیلد (حافظه انجمنی بازگشتی) و استفاده از آنها در حل مسئله

پرسپترون تک لایه

با تابع تحریک آستانه ای دو مقداره متقارن



پرسپترون تک لایه

- ساختار در فصل پیش دیده شد.
- توانایی در تفکیک الگوهای ورودی
  - جداسازی الگوهای خطی
  - مرز جدا ساز نواحی در حالت تک نرونی

یک فوق صفحه و برای حالت دو بعدی یک خط راست است.

$$b+\sum_{n} w_n p_n = 0$$

- در حالت کلی برای هر نقطه دلخواه q روی ناحیه مرزی داریم:

$$< \underline{q}, \underline{w} >= -b$$

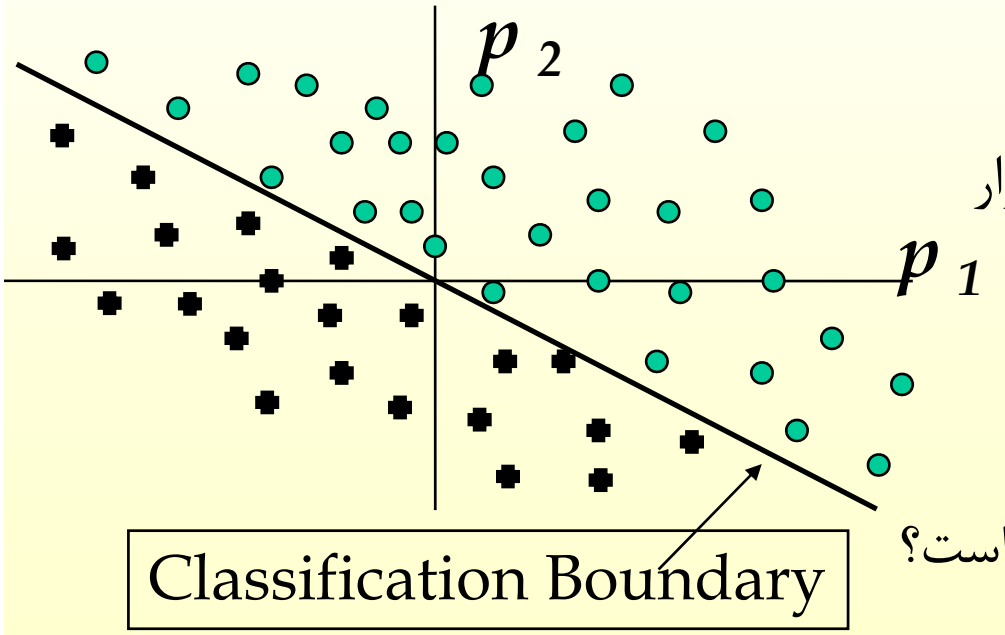
- $< \underline{a},\underline{b} >$ نماد ضرب داخلی دو بردار بوده و همواره تصویر $\underline{b}$ روی $\underline{a}$ مضربی از ضرب داخلی دو بردار است.

پرسپترون تک لایه

- تذکر ۱: خط مرزی همواره عمود بر بردار وزن بوده و محل قرار گرفتن مرز توسط بردار بایاس تعیین می شود.
- تذکر ۲: تصویر بردار سازنده هر نقطه روی خط مرزی بر بردار وزن، برابر با مضربی از منفی جمله بایاس (-b) است. (چرا؟)
- نتیجه: برای تمام نقاطی که بالای (با توجه به جهت بردار وزن) خط مرزی قرار دارند، ورودی خالص نرون مثبت است و برای تمام نقاط زیر آن منفی می باشد.
- سوال

در شکل روبرو ورودی خالص نرون چگونه است؟

آیا پاسخ یکتاست؟



پرسپترون تک لایه

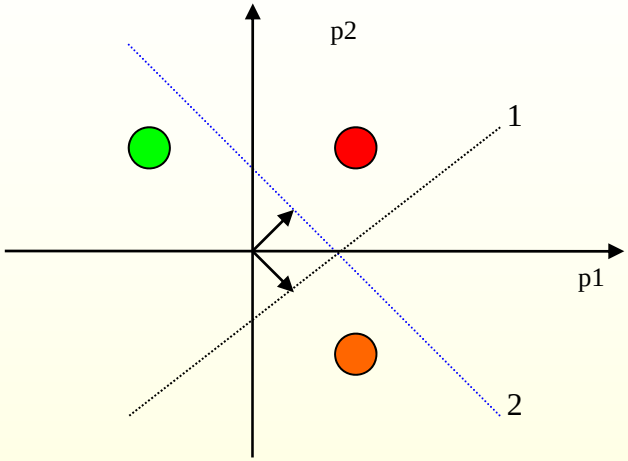
- در حالتی که شبکه دارای s نرون میانی است، ماتریس وزن دارای s سطر بوده و هر سطر یک خط (فوق صفحه) مرزی را تعیین می کند.

- سوال

حداکثر تعداد نواحی که به طور خطی از هم تفکیک پذیرند برای پرسپترون با s نرون میانی و بردار ورودی دارای n عنصر، چه مقدار است؟

- مثال تشریحی شناسایی الگو
- چون دارای ۳ گروه هستیم، از ۲ نرون میانی استفاده می کنیم (چرا؟). ماتری وزن W نیز ۲*۲ خواهد بود.
- با توجه به شکل مشخص است که جوابهای بیشماری وجود دارد. یکی از آنها نمایش داده شده است

مثال تشریحی با پرسپترون



- W(1,:)=[1,-1]
- W(2,:)=[1 1]
- b=[-1;-1]
- به این ترتیب ناحیه قرمز (مربوط به سیب) دارای خروجی a=[-1;1]،

ناحیه نارنجی (مربوط به پرتقال) خروجی a=[1;-1]

ناحیه سبز (مربوط به گلابی) خروجی a=[-1;-1]

می باشد.

تمرین ۱: درستی این بردارهای وزنی و بایاسها چک شود.

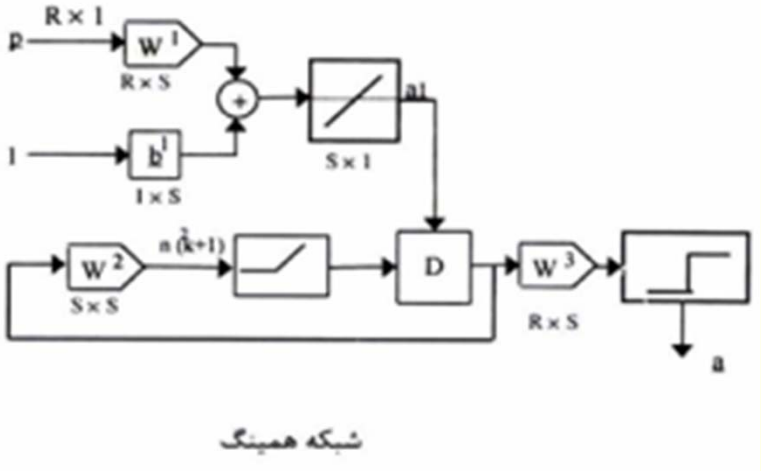
تمرین ۲: شبکه ورودی پرتقال بیضوی را چگونه ارزیابی می کند؟ چرا؟

سوال

۱) آیا می توان همواره دسته ها را در فضای ورودی توسط پرسپترون تک لایه مجزا کرد؟

۲)آیا روند به کار رفته برای یافتن وزنهای شبکه پرسپترون تک لایه برای مسائل با ابعاد وسیعتر نیز قابل کاربرد است؟

شبکه همینگ



شبکه همینگ

- دارای هر دو ساختار پیشخور و پسخور
- دارای ۳ لایه
- برای شناسایی الگوهای باینری
- تشخیص اینکه کدام الگوی مرجع بیشترین نزدیکی را به الگوی ورودی دارد.

- اگر به شبکه ورودی $\underline{p}^i$ (یکی از الگوهای مرجع) اعمال شود، خروجی نیز $\underline{p}^i$ خواهد بود.
- در مواردی که به شبکه بردار ورودی اختیاری (مثلا پرتقال بیضوی) اعمال می شود انتظار بر این است که الگوی مرجعی که بیشترین نزدیکی را با ورودی دارد، در خروجی ظاهر شود.
- دو پروسه محاسباتی: ذخیره سازی الگوها و بازیابی اطلاعات انجام می گیرد.

شبکه همینگ : لایه اول

- شبکه پیشخور با تابع تبدیل خطی
- همبستگی یا ضرب داخلی بین بردارهای مرجع با بردار ورودی را محاسبه می کند.

بردارهای مرجع الگوهایی هستند که قصد شناسایی آنها را داریم.

$$W^1 = \begin{bmatrix} (\underline{p}^1)^T \\ (\underline{p}^2) \\ (\underline{p}^3)^T \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}$$

- ذخیره سازی الگوهای مرجع در لایه اول
  - تعداد نورونهای این لایه برابر تعداد الگوها
- R تعداد عناصر بردار ورودی
- به نوعی فاصله همینگ سنجیده می شود.

$$\underline{b}^1 = [R, R, R]^T$$

$$p_H^j = 1/2 \sum_{i=1}^{R} | \underline{p}_i^j - \underline{p}_i |$$

شبکه همینگ : لایه اول

- عناصر بردار خروجی لایه اول مقادیری بین 0

- نرون با بزرگترین مقدار خروجی متناظر با الگوی مرجعی خواهد بود که بیشترین شباهت (کمترین فاصله همینگ) را با الگوی ورودی دارد. چرا؟

- در مرحله بعد باید خروجی ماکزیمم از بقیه جدا گردد. این عمل توسط لایه برگشتی انجام می شود.

شبکه همینگ : لایه دوم

- **لایه برگشتی (لایه WTA) (لایه رقابتی)**

$$\underline{a}^2(0) = \underline{a}^1$$
$$\underline{a}^2(k+1) = posl(W^2\underline{a}^2(k))$$

$$posl(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$w_{ij}^2 = \begin{cases} 1 & i = j \\ -\varepsilon & i \neq j \end{cases}, 0 < \varepsilon < 1$$

- انتخاب ماتریس W^2

$$\varepsilon = \frac{1}{s-1}$$

- برای رسیدن به سرعت همگرایی بیشتر معمولا داریم:
- دلیل انتخاب ماتریس W^2 به شرح فوق، در بخش اثبات همگرایی شبکه همینگ در فصل ۳ آمده است.
- تمرین: تحقیق کنید که خروجی لایه دوم به ورودی سیب، برابر [2;0;0] می باشد.

شبکه همینگ : لایه سوم

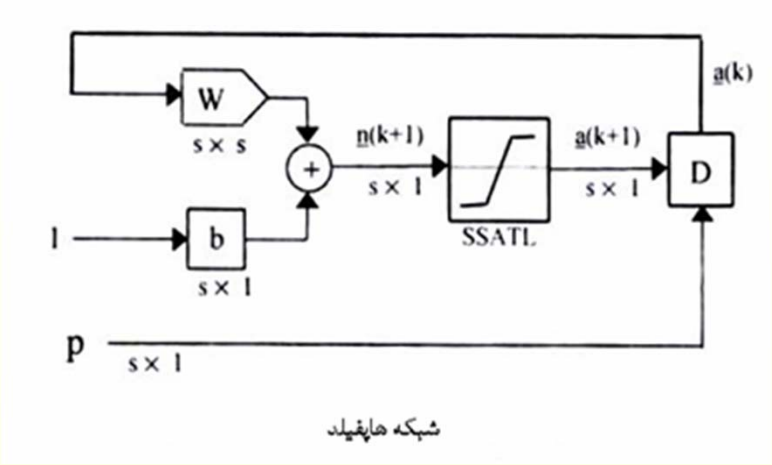
- شبکه پیشخور با تابع تحریک آستانه متقارن دو مقداره
- پس از همگرایی لایه دوم، بردار مرجع ذخیره شده در خروجی شبکه ظاهر می گردد. بازیابی اطلاعات
- انتخاب وزنهای لایه سوم:

$$w_{ij}^{3} = p_{i}^{j}$$

  - P_j^i عنصر i ام از الگوی مرجع j ام می باشد.
- مثلا اگر ورودی سیب باشد، خروجی شبکه الگوی مربوط به سیب یعنی [1;1] خواهد بود. بررسی شود.
- توجه:

حتما باید لایه دوم همگرا شده باشد

شبکه هاپفیلد



شبکه هاپفیلد

- شبکه تک لایه بازگشتی
- عملکرد همه نرونها مشابه یکدیگر
- عدم وجود نرون ورودی یا خروجی
- مقادیر اولیه نرونها توسط ورودی تعیین شده و سپس شبکه با تکرار به یکی از الگوهای مرجع همگرا می شود.

$$\underline{a}(k+1) = ssatl(w\underline{a}(k) + \underline{b})$$

$$\underline{a}(0) = \underline{p}$$

$$ssatl(x) = \begin{cases} -1 & x < -1 \\ x & -1 \le x \le 1 \\ 1 & x > 1 \end{cases}$$

شبکه هاپفیلد

- انتظار داریم که اگر در لحظه 0 به شبکه الگوی سیب [1;1] اعمال شود، شبکه به سمت [1;1] همگرا گردد.
- یک انتخاب خوب برای W ,b عبارتست از:

$$W = \begin{bmatrix} 1.2 & 0 \\ 0 & 1.5 \end{bmatrix} \quad \underline{b} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

- مسلما برای مسائل مشکلتر نیاز به روند طراحی وزنها می باشد
- تمرین: پاسخ شبکه به ورودیهای سیب، پرتقال و گلابی بررسی شود.
- سوال: پاسخ شبکه به ورودی پرتقال بیضوی چیست؟
  - عیب شبکه هاپفیلد : همگرایی به الگویی غیر از الگوهای مرجع

مهمترین ویژگیها

- شبکه پرسپترون جواب را کد می کند.

- شبکه همینگ به پاسخ مناسب منتهی می شود و همواره به یکی از الگوهای مرجع همگرا می شود، و الگوی دارای بیشترین تشابه به ورودی را بر می گزینند.

- شبکه هاپفیلد به پاسخس همگرا می شود که ممکن است جزو الگوهای ذخیره شده نباشد.

مهمترین نکات

- انتخاب مناسب پارامترها
  - انواع الگوریتمهای یادگیری برای تنظیم پارامترها از تفاوتهای مهم انواع شبکه ها با یکدیگر است

- انتخاب ساختار مناسب شبکه های عصبی
  - چه زمانی باید از ساختارهای پیشخور یا پسخور استفاده کرد؟
  - چگونه می توان فهمید شبکه های پسخور نهایتا همگرا می شوند؟
  - پایداری کدامیک از شبکه های پیشخور یا پسخور راحتتر تضمین می شود؟
  - آیا می توان وزنها و بایاسهای شبکه همینگ را ، وقتی که الگوهای مرجع را نمیدانیم، آموزش داد؟

شبکه های عصبی

استاد: محمد باقر منهاج

موضوعات

- **فصل چهارم**
  - فرایند یادگیری
  - شبکه های عصبی به عنوان سیستمهای دینامیکی آموزش پذیر
  - یادگیری شبکه
  - انواع یادگیری: با ناظر، بدون ناظر و تشدیدی
  - قانون یادگیری پرسپترون

فصل ۴

پرسپترون تک لایه

فرایند یادگیری

"مار گزیده از ریسمان سیاه و سفید می ترسد"

"از علل مشابه انتظار عواقب مشابه را داریم"

- سیستمهای یادگیر صرفا با مشاهده عملکردشان رفتار خود را جهت رسیدن به هدفی خاص بهبود می بخشند.
- اگر مقاصد و اهداف به طور کامل تعریف شده باشد، دیگر احتیاجی به فرایند یادگیری نیست.
  - یادگیری به علت عدم قطعیت در شرایط محیطی لازم می گردد.
- رفتار سیستمهای یادگیر توسط الگوریتمهای بازگشتی بیان می شود.
- با این الگوریتمها شاخص اجرایی مشخص شده ای بهینه می گردد.
- در حالت کلی دو نوع یادگیری داریم: با ناظر - بدون ناظر

یادگیری با ناظر

- در این حالت در هر تکرار الگوریتم یادگیری جواب مطلوب سیستم یادگیرنده از قبل آماده است.
  - به عبارت دیگر الگوریتم به جواب مطلوب دسترسی دارد.

- مثلا اگر هدف یادگیری تابع $y=x^2$ باشد، پاسخ مطلوب به ورودی ۰.۵ برابر ۰.۲۵ می باشد.
- به طور کل جوابی را که سیستم یادگیر با وضعیت فعلی پارامترهایش می دهد، جواب واقعی در نظر می گیریم.

بنابر این الگوریتم هم به جواب مطلوب و هم به جواب واقعی دسترسی دارد. یعنی خطای یادگیری که تفاوت این دو مقدار می باشد، در دسترس است.

یادگیری بدون ناظر

- در این حالت جواب مطلوب برای سیستم یادگیرنده موجود نیست.
- یعنی : عدم دسترسی به خطای یادگیری
- سوال
  - چگونه می توان یک سیستم یادگیرنده را آموزش داد، اگر ندانیم که این سیستم قرار است چه کاری انجام دهد؟
- خواهیم دید که بیشتر الگوریتمهای بدون ناظر عمل خوشه بندی را انجام می دهند.
  - یعنی می آموزند که الگوهای ورودی را به تعداد متناهی از گروهها تقسیم کنند.
- دقت شود که در این حالت فرد معلم یا طراح است که مقصد نهایی را معلوم میکند.

یادگیری بدون معلم مفهوم غلطی است

خلاصه فرایند یادگیری

- سیستم یادگیرنده توسط محیط تحریک شود.
- قانون یادگیری با رجوع به نتیجه تحریک، پارامترهای سیستم یادگیری را تغییر دهد.
- سیستم یادگیرنده به خاطر تغییراتی که در ساختار داخلی آن اتفاق افتاده است، پاسخ مناسبتری به محیط بدهد.

شبکه های عصبی: سیستمهای دینامیکی آموزش پذیر

- شبکه عصبی توانایی تجربه اندوزی از گذشته و بهبود رفتار خود را داراست
  - بهبود یادگیری در طول زمان باید بر اساس معیاری سنجیده شود.

- معیار بهبود، هدف یادگیری را مدل می کند.
- قانون یادگیری، روندی است که در آن ماتریس وزنها و بردارهای بایاس شبکه عصبی تنظیم میشوند.
- این قوانین توسط روابط بازگشتی و عموما به صورت معادلات تفاضلی بیان می شوند.
- نوع یادگیری در چگونگی این روند دخالت دارد.

معادله یادگیری در حالت کلی

- یک نرون با یک بردار پارامتر $\underline{w}$ و بردار ورودی $\underline{p}$ قابل نمایش است.

$$\underline{w} = [w_1, w_2, ..., w_R, b]^T$$
$$\underline{p} = [p_1, p_2, ..., p_R, 1]^T$$

- هر نرون توانایی تنظیم بردار پارامتر خود بر اساس ورودی و یک سیگنال معلم را داراست
  - در یادگیری با ناظر: سیگنال معلم همان سیگنال خطا است
  - در یادگیری بدون ناظر: سیگنال معلم تغییر بردار حالت خود نرون است
- قانون کلی یادگیری برای یک نرون
  - حالت پیوسته: $\dot{\underline{w}}(t) = -\alpha w(t) + \eta.l.\underline{p}(t)$
  - حالت گسسته: $\underline{w}(k+1) = (1-\alpha)w(k) + \eta.l.\underline{p}(k)$

معادله یادگیری در حالت کلی

- α نشان دهنده تاثیر از گذشته و η نرخ یادگیری می باشد.
- L سیگنال یادگیری بوده و در حالت کلی تابعی از p, w, t (برای حالت با ناظر) می باشد.
- خواهیم دید که همه انواع قوانین یادگیری شبکه در این فرم کلی قرار دارند.
  - مثلا در قانون یادگیری پرسپترون $\alpha = 0$ بوده و سیگنال معلم همان سیگنال خطا و برابر l=t-a است.

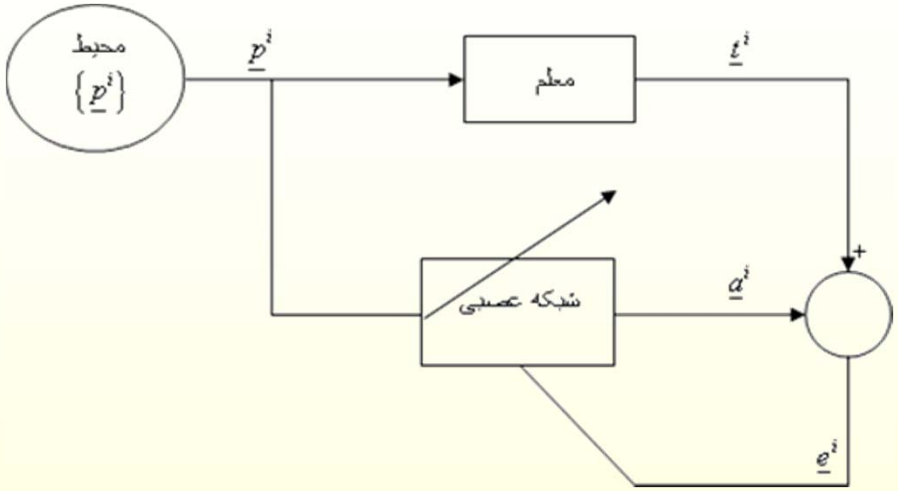
- برای هر نرون شبکه عصبی نیز داریم:
  - حالت پیوسته:
  - حالت گسسته:

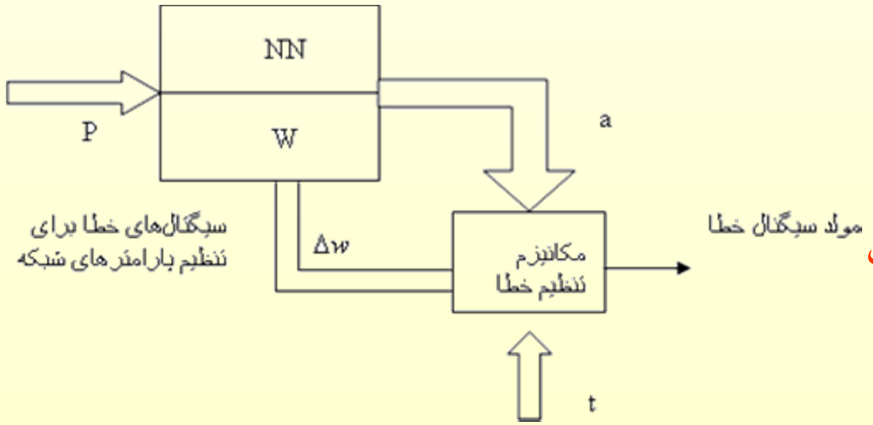
$$\dot{w}_{ij}(t) = -\alpha w_{ij}(t) + \Delta w_{ij}(t)$$

$$w_{ij}(k+1) = (1-\alpha) w_{ij}(k) + \Delta w_{ij}(k)$$

- $\Delta w_{ij}(k)$ ترم اصلاحی می باشد.

یادگیری با ناظر در شبکه عصبی





- زوج داده های یادگیری $(\underline{p}^i , \underline{t}^i)$ $i=1,2,\ldots,L$
- t پاسخ مطلوب و a خروجی واقعی شبکه
- سیگنالهای خطا پارامترهای شبکه را به نحوی تنظیم می کنند که پاسخ شبکه به سمت پاسخ مطلوب حرکت کند.
- LMS و پس انتشار خطا از انواع با ناظر بوده که در فصلهای ۷ و ۸ آمده است.
- سوال: تفاوت اصلی در یک سیستم فیدبک دارای یادگیری با ناظر و سیستمهای تطبیقی چیست؟

یادگیری تشدیدی (تقویتی) در شبکه عصبی

اشکال یادگیری با ناظر: بدون معلم نمی تواند مواضع جدیدی را که توسط داده های تجربی پوشانده نشده است، یاد بگیرد.

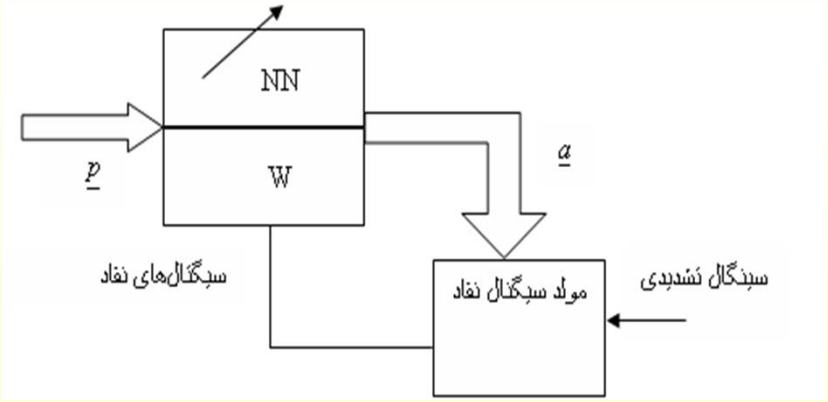
- یادگیری با ناظر: offline , online
- یادگیری تشدیدی: online

- در یادگیری offline به کمک داده های یادگیری طراحی شبکه عصبی و آموزش آن انجام می شود و پس از آن به عنوان یک سیستم استاتیکی عمل می کند. ولی در حالت online مثل یک سیستم دینامیکی همواره در حال انجام کار است.

- یادگیری تشدیدی نوع خاصی از یادگیری با ناظر است زیرا همچنان رفتار شبکه ارزیابی می شود.

یادگیری تشدیدی (تقویتی) در شبکه عصبی

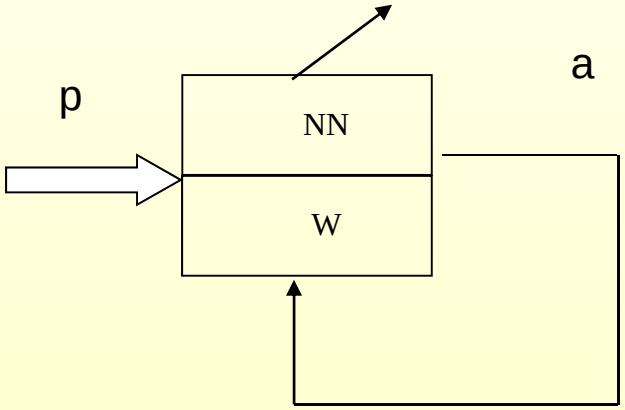
- در یادگیری تشدیدی معمولا مقادیر خروجی مطلوب در دسترس نیست و در آن به شبکه اعلام میشود که مثلا عملکرد آن ۵۰٪ درست است یا اصلا مطلوب نیست.



- در حالت حاد با یک بیت میتوان اعلام کرد که خروجی درست است یا خیر.
- سیگنال برگشتی از نوع نقادی است در صورتیکه در یادگیری با ناظر جنبه دستوری دارد (چون t معلوم است).
- سیگنال نقاد پارامترهای شبکه را تنظیم میکند با این امید که سیگنال نقاد بهتری در آینده جهت تنظیم پارامترها تولید شود.

یادگیری بدون ناظر در شبکه عصبی

- یادگیری خود سازمانده
- در این نوع یادگیری، هیچ سیگنالی که اطلاعاتی در مورد مطلوبیت جواب شبکه به خود شبکه وارد نماید موجود نیست.
- شبکه در اینجا بایستی تنها با ارائهٔ صرف بردارهای ورودی، ارتباطات موجود بین الگوهای ورودی را پیدا کرده و در خروجی شبکه کُد نماید.



قانون یادگیری پرسپترون تک لایه (SLPR)

- از نوع یادگیری با ناظر
  - خطای یادگیری در دسترس
- برای شبکه ای با یک نرون میانی و دو ورودی دیدیم:
  - معادل است با خطی که فضای ورودی را به ۲ بخش تقسیم میکند.
  - خط دارای ضریب زاویه $-w_1/w_2$ بوده و بر بردار وزن $W=[w_1\ w_2]$ عمود است.
  - برای تمامی نقاط بالای خط مرزی (با توجه به جهت بردار وزن) $Wp+b>0$ و برای تمامی نقاط پایین خط مرزی $Wp+b<0$ می باشد.
- در صورت افزایش تعداد نرونهای میانی و $R=2$، تعداد نواحی قابل تفکیک توسط خطوط متقاطع در فضای R^2 افزایش می یابد.

SLPR

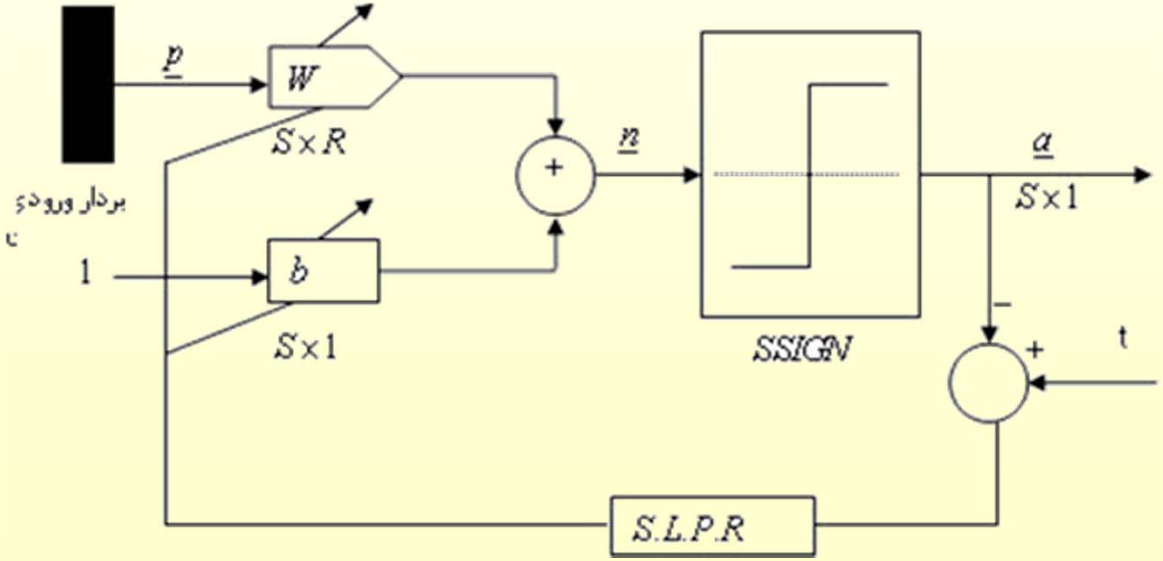
$$V^n = V^n + \frac{1}{2}\underline{e}\,\underline{q}^T \quad \Rightarrow$$

$$W^n = W^n + \frac{1}{2}\underline{e}\,\underline{p}^T \quad \Leftrightarrow \quad W(k+1) = W(k) + \frac{1}{2}\underline{e}(k)\,\underline{p}^T$$

$$\underline{b}^n = \underline{b}^n + \frac{1}{2}\underline{e} \quad \Leftrightarrow \quad \underline{b}(k+1) = \underline{b}(k) + \frac{1}{2}\underline{e}(k)$$

- بب

$$\underline{e} = \underline{t} - \underline{a}\,,\ \left(\underline{t}, \underline{a} \in \{-1,1\}^S\right)$$



شبکه های عصبی

استاد: محمد باقر منهاج

موضوعات

- **فصل هفتم**
  - مقدمه
  - مبانی بهینه سازی و نقاط بهینه
  - روشهای می نیمم سازی
  - یادگیری ویدرو-هوف و شبکه آدالاین
  - الگوریتم LMS
  - کاربرد شبکه آدالاین در فیترهای تطبیقی

فصل ۷

شبکه های آدالاین و یادگیری LMS

مقدمه

- شبکه آدالاین با قانون یادگیری ویدرو – هوف (معروف به قانون LMS) در سال ۱۹۶۰ و بعد از شبکه پرسپترون با قانون یادگیری SLPR به وجود آمد.
- شبکه آدالاین شبیه پرسپترون است ولی با تابع تبدیل خطی (به جای آستانه دو مقداره)
- محدودیت شبکه های پرسپترون و آدالاین:
  - فقط توانایی طبقه بندی الگوهایی را دارند که به طور خطی از هم جداپذیرند.
- قانون SLPR:
  - همگرایی به یک جواب را در صورت وجود تضمین میکند.
  - نسبت به نویز خیلی حساس است، زیرا خط مرزی خیلی نزدیک الگوهای یادگیری است.
- قانون LMS:
  - خیلی قویتر از SLPR
  - دارای کاربردهای فراوان در مهندسی و به خصوص پردازش سیگنال

مقدمه

- شبکه های آدالاین
  - مناسب برای تقریب خطی یک تابع یا اجرای عمل شناسایی الگو
  - دارای منحنی سطح خطای اجرایی سهموی
  - دارای مزیت برخورداری از یک نقطه مینیمم
  - یادگیری از نوع با ناظر
  - پارامترهای شبکه به نحوی تنظیم می شوند که شاخص اجرایی میانگین مربعات خطا بهینه شود.
  - قانون یادگیری LMS تقریبی از الگوریتم بیشترین شیب در حداقل کردن شاخص اجرایی
  - حتی در صورت عدم وجود جواب (در این حالت قانون SLPR همگرا نخواهد شد) ، باز به جایی که میانگین مربعات خطا حداقل است، همگرا می شود.
  - نسبت به نویز کمتر حساس

مبانی بهینه سازی و نقاط بهینه

- پایه همه تکنیکهای یادگیری از نوع عملکردی
- در یادگیری عملکردی پارامترهای شبکه به نحوی تنظیم می شوند که عملکرد شبکه بهینه شود.

- قدم اول: تعریف عملکرد و تعیین شاخص عملکرد. (معمولا سطح اجرایی میانگین مربعات خطا)
- قدم دوم: جستجو در فضای پارامترهای شبکه برای تنظیم آنها به طوریکه معیار اجرایی عملکرد کاهش یابد.

مبانی بهینه سازی: بسط تیلور و تقریب توابع

- بسط تیلور تابع اسکالر $F(\underline{x})$ با متغیر برداری $\underline{x}$

$$F(\underline{x}) = F(\underline{x}^*) + \nabla_{\underline{x}} F^T(\underline{x}^*)(\underline{x} - \underline{x}^*) + \frac{1}{2}(\underline{x} - \underline{x}^*)^T \nabla^2_{\underline{x}} F(\underline{x}^*)(\underline{x} - \underline{x}^*) + \dots$$

- بردار گرادیان و ماتریس هسیان تابع F به ترتیب برابرند با:

$$\nabla_{\underline{x}} F(\underline{x}) = \left[\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \dots, \frac{\partial F}{\partial x_n}\right]^T \qquad \nabla^2_{\underline{x}} F(\underline{x}) = \begin{bmatrix} \frac{\partial^2 F(\underline{x})}{\partial^2 x_1} & \frac{\partial^2 F(\underline{x})}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 F(\underline{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F(\underline{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 F(\underline{x})}{\partial^2 x_2} & \dots & \frac{\partial^2 F(\underline{x})}{\partial x_2 \partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial^2 F(\underline{x})}{\partial x_n \partial x_1} & \frac{\partial^2 F(\underline{x})}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 F(\underline{x})}{\partial^2 x_n} \end{bmatrix}$$

مبانی بهینه سازی: مشتقات برداری جهت دار

- مشتق تابع F در مسیر بردار p :

$$D_{\underline{p}}F(\underline{x}) = \frac{\underline{p}^T \nabla_{\underline{x}} F(\underline{x})}{\|\underline{p}\|}$$

- مشتق دوم تابع F در مسیر بردار p :

$$D^2_{\underline{p}}F(\underline{x}) = \frac{\underline{p}^T \nabla^2_{\underline{x}} F(\underline{x}) \underline{p}}{\|\underline{p}\|^2}$$

- سوال:

در چه مسیرهایی مقدار مشتق برداری ماکزیمم، مینیمم یا صفر خواهد بود؟

مبانی بهینه سازی: شرایط لازم برای نقاط بهینه

- برای اینکه x^* نقطه مینیمم تابع F باشد
  - شرط درجه اول : $\nabla_{\underline{x}} F(\underline{x}^*) = 0$

    نقاط ایستای تابع F
  - شرط درجه دوم: (ماتریس مثبت معین) $\nabla^2_{\underline{x}} F(\underline{x}^*) > 0$
- شرط لازم برای نقطه مینیمم x^* :

$$\nabla_{\underline{x}} F(\underline{x}^*) = 0 \qquad \nabla^2_{\underline{x}} F(\underline{x}^*) \geq 0$$

- شرط لازم و کافی برای نقطه مینیمم x^* :

$$\nabla_{\underline{x}} F(\underline{x}^*) = 0 \qquad \nabla^2_{\underline{x}} F(\underline{x}^*) > 0$$

- شرط کافی برای اینکه x^* زین اسبی باشد، ماتریس هسیان نامعین باشد.

مبانی بهینه سازی: توابع درجه دوم

$$F(\underline{x}) = \frac{1}{2}\underline{x}^T A\underline{x} + \underline{b}^T\underline{x} + c \quad , A^T = A$$

- متقارن بودن A الزامی نیست زیرا هر ماتریس نامتقارن را می توان به متقارن تبدیل کرد. چگونه؟

$$\nabla_{\underline{x}} F(\underline{x}) = \underline{b} + A\underline{x}$$

$$\nabla^2_{\underline{x}} F(\underline{x}) = A$$

- شاخص عملکردی میانگین مربعات خطا از نوع درجه دوم می باشد.
- تمامی توابع با مشتقات پیوسته مرتبه دوم حول یک ناحیه به اندازه کافی کوچک مثل تابع درجه دوم عمل می کند. چرا؟
- بیشتر الگوریتمهای بهینه سازی با تابع عملکرد مرتبه دوم پس از تعداد محدودی تکرار به نقطه بهینه خواهند رسید.

مبانی بهینه سازی: توابع درجه دوم و ساختار ویژه

- در توابع درجه دوم ساختار ویژه (مقادیر و بردارهای ویژه) ماتریس هسیان نقش مهمی در مینیمم کردن توابع دارند.
- به علاوه مقادیر و بردارهای ویژه دارای تعابیر فیزیکی نیز هستند.

تمرین:

ارتباط مقادیر و بردارهای ویژه ماتریس هسیان در مینیمم سازی چیست؟ با توجه به نقشی که آنها در مثبت معین یا منفی معین کردن یک ماتریس دارند، توضیح دهید.

(در بخش ۷-۲-۸ کتاب آمده است.)

روند مینیمم سازی: الگوریتم کلی

- روند بازگشتی: تخمین جدید از روی تخمین فعلی قابل محاسبه است

$$\underline{x}(k+1) = \underline{x}(k) + \alpha(k)\underline{p}(k)$$

- X(k) تخمین فعلی نقطه مینیمم تابع F، $\alpha(k)$ نرخ یادگیری و p(k) بردار جستجو است.

- در روشهای مختلف مینیمم سازی، بردار جستجو متفاوت است.
- p(k) به گونه ای تعیین می شود که مقدار تابع F در هر مرحله کاهش یابد.

- بردار جستجو از روی اطلاعات گرادیان و هسیان تابع محاسبه می شود.

روند مینیمم سازی: روش بیشترین نزول (SD)

- دیدیم که اگر مسیر p(k) در خلاف جهت گرادیان باشد، مقدار مشتق جهتی کمترین میزان ممکن را خواهد داشت.
- در اینجا نیز با توجه به بسط تیلور تابع F حول x(k) به سادگی دیده می شود که اگر بردار جستجو در خلاف جهت گرادیان انتخاب شود، بیشترین تنزل در مقدار تابع F را خواهیم داشت.
- الگوریتم بیشترین نزول برابر است با:

$$\underline{x}(k+1) = \underline{x}(k) - \alpha(k)\nabla F(\underline{x}(k))$$

- روش تعیین نرخ یادگیری $\alpha(k)$:
  - در نظر گرفتن یک مقدار ثابت. مثلا ۰.۰۵ یا $\alpha(k) = 2/k$
  - یافتن $\alpha(k)$ در هر مرحله به گونه ای که تابع F(x(k+1)) نسبت به $\alpha(k)$ حداقل شود. نرخ یادگیری بهینه

نکات مربوط به الگوریتم SD

- ثابت کنید نرخ یادگیری بهینه در الگوریتم SD برای توابع درجه دوم عبارتست از:
(بخش ۷-۳-۳ مطالعه شود)

$$\alpha_k = \frac{g_k{}^T g_k}{g_k{}^T A g_k}$$

- مراحل متوالی الگوریتم SD بر هم عمودند.

شبکه های عصبی

استاد: محمد باقر منهاج

موضوعات

- **فصل هفتم**
  - مقدمه √
  - مبانی بهینه سازی و نقاط بهینه √
  - روشهای می نیمم سازی √
  - یادگیری ویدرو-هوف و شبکه آدالاین
  - الگوریتم LMS
  - کاربرد شبکه آدالاین در فیترهای تطبیقی

مثالی از الگوریتم SD

- تابع $F(\underline{x}) = x_1^{\,2} + 9x_2^{\,2}$

$$\underline{x} = [x_1 \quad x_2]^T$$

$$\underline{x}_0 = [0.5 \quad 0.5]^T$$

با نقطه حداقل [0 0]

$$\nabla F(\underline{x}_0) = \begin{bmatrix} 2x_1 \\ 18x_2 \end{bmatrix}_{\underline{x}=\underline{x}_0} = [1 \quad 9]^T$$

$$\underline{x}_1 = \underline{x}_0 - \alpha g_0 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} - 0.05 \begin{bmatrix} 1 \\ 9 \end{bmatrix} = \begin{bmatrix} 0.45 \\ 0.05 \end{bmatrix}$$

$$\underline{x}_2 = \underline{x}_1 - \alpha g_1 = \begin{bmatrix} 0.45 \\ 0.05 \end{bmatrix} - 0.05 \begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix} = \begin{bmatrix} 0.405 \\ 0.005 \end{bmatrix}$$

نکات

- با تغییر نرخ یادگیری شکل مسیر حرکت تغییر می کند.
- اگر نرخ یادگیری زیاد شود سرعت نزدیک شدن به نقطه بهینه زیادتر شده ولی میزان نوسانات بیشتر خواهد شد.
- اگر میزان نرخ یادگیری از حدی بیشتر شود، الگوریتم واگرا می گردد. به این حد نرخ یادگیری پایدار گویند.
- برای توابع درجه دوم ثابت می شود که نرخ یادگیری پایدار برابر است با

$$\alpha < \frac{2}{\lambda_{\max}(A)}$$

A ماتریس هسیان تابع F

- محاسبه نرخ یادگیری پایدار در مثال قبل:

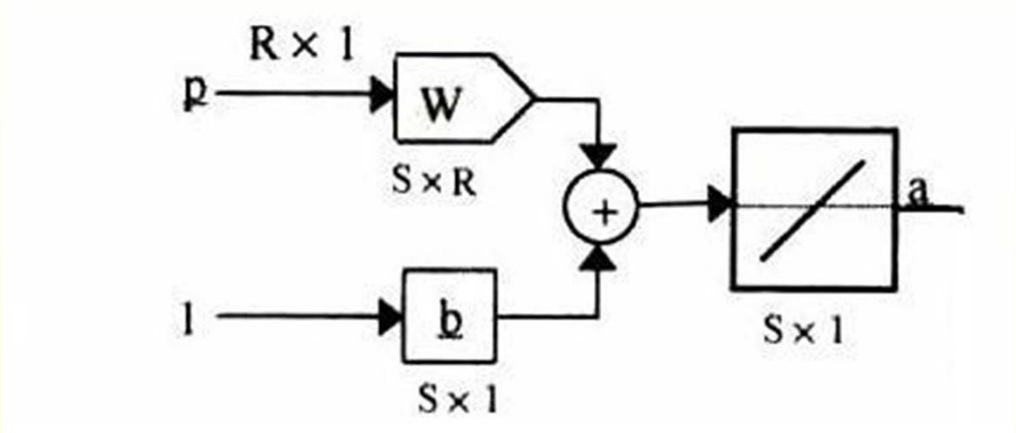
$$A = \begin{bmatrix} 2 & 0 \\ 0 & 18 \end{bmatrix}$$

$$\alpha < \frac{2}{\lambda_{\max}(A)} = \frac{2}{18} = 0.11$$

یادگیری LMS یا ویدرو–هوف

شبکه آدالاین

- شبکه آدالاین با قانون یادگیری ویدرو – هوف (معروف به قانون LMS) در سال ۱۹۶۰ و بعد از شبکه پرسپترون با قانون یادگیری SLPR به وجود آمد.
- شبکه آدالاین شبیه پرسپترون است ولی با تابع تبدیل خطی (به جای آستانه دو مقداره)



$$\underline{a} = W\,\underline{p} + \underline{b}$$

معادلات ویدرو-هوف در حالت تک نرون

- بازنویسی معادله خروجی با فرمول بندی جدید

$$\underline{x} = \begin{bmatrix} W_{1,1} \\ \vdots \\ W_{1,r} \\ b \end{bmatrix} = \begin{bmatrix} W_1^T \\ b \end{bmatrix} \quad \underline{q} = \begin{bmatrix} \underline{p} \\ 1 \end{bmatrix}$$

$$a = \underline{x}^T \underline{q}$$

- تابع هزینه میانگین مربعات خطا

$$e = t - a = t - \underline{x}^T \underline{q}$$

$$F(\underline{x}) = E(e^2) = \underline{x}^T R \underline{x} - 2\underline{x}^T \underline{d} + c$$

- تعیین نقطه ایستا از روی گرادیان تابع

$$\nabla F(\underline{x}) = -2\underline{d} + 2R\underline{x}$$

$$\nabla F(\underline{x}) = 0 \Rightarrow \underline{x}^* = R^{-1}\underline{d}$$

توجه

- اگر بتوانیم معکوس ماتریس را محاسبه کنیم نیازی به الگوریتمهای مینیمم سازی نداریم.
- اگر نخواهیم (یا نتوانیم) معکوس R را محاسبه کنیم، الگوریتم مینیمم سازی SD را می توان به کار برد.
  - در این حالت نیاز به محاسبه گرادیان تابع داریم.
- در حالت کلی مطلوب یا مناسب نیست که بردار d و ماتریس R محاسبه شوند. لذا تقریبی از الگوریتم SD یا همان LMS استفاده می شود.
  - استفاده از گرادیان لحظه ای به جای گرادیان واقعی

الگوریتم LMS

- در واقع استفاده از خطای لحظه ای به عنوان شاخص عملکرد

$$\hat{F}(\underline{x}) = e^2(k) = (t(k) - a(k))^2$$

$$\nabla\hat{F}(\underline{x}) \equiv \hat{\nabla}F(\underline{x}) = \nabla e^2(k)$$

- دقت شود که $\nabla F(\underline{x}) = E[\nabla\hat{F}(\underline{x})]$
- به کمک قاعده زنجیره ای به سادگی دیده می شود که $\nabla\hat{F}(\underline{x}) = -2e(k)\underline{q}(k)$
- قانون LMS در حالت تک نرون:

$$\underline{x}(k+1) = \underline{x}(k) + 2\alpha e(k)\underline{q}(k)$$

- قانون LMS در حالت کلی:

$$W(k+1) = W(k) + 2\alpha\underline{e}(k)(\underline{p}(k))^T$$

$$\underline{b}(k+1) = \underline{b}(k) + 2\alpha\underline{e}(k)$$

فرم دسته ای یادگیری LMS در حالت تک نرون

- شاخص اجرایی در حالت تک نرون
  - N – تعداد داده های یادگیری

$$\overline{F}(\underline{x}) = \frac{1}{N}\sum_{k=1}^{N} e^2(k)$$

- شاخص اجرایی در حالت کلی S نرون

$$\overline{F}(\underline{x}) = \frac{1}{N}\sum_{k=1}^{N} \underline{e}^T(k)\underline{e}(k)$$

- در حالت تک نرون به سادگی دیده می شود:

$$\nabla\overline{F}(\underline{x}) = \frac{-2}{N}\sum_{k=1}^{N} e(k)\underline{q}(k)$$

$$\underline{x}(k+1) = \underline{x}(k) + \frac{2\alpha}{N}\sum_{k=1}^{N} e(k)\underline{q}(k)$$

فرم دسته ای یادگیری LMS در حالت کلی

$$W(k+1)=W(k)+\frac{2\alpha}{N}\begin{bmatrix}\sum_{k=1}^{N}e_1(k)(\underline{p}(k))^T\\ \vdots\\ \sum_{k=1}^{N}e_S(k)(\underline{p}(k))^T\end{bmatrix}=W(k)+\frac{2\alpha}{N}\sum_{k=1}^{N}\underline{e}(k)(\underline{p}(k))^T$$

$$\underline{b}(k+1)=\underline{b}(k)+\frac{2\alpha}{N}\begin{bmatrix}\sum_{k=1}^{N}e_1(k)\\ \vdots\\ \sum_{k=1}^{N}e_S(k)\end{bmatrix}=\underline{b}(k)+\frac{2\alpha}{N}\sum_{k=1}^{N}\underline{e}(k)$$

- تمرین ۱: شبکه آدالاین برای مثال سیب پرتقال گلابی در هر دو حالت ترتیبی و دسته ای شبیه سازی و نتایج مقایسه شود. (بخش ۷-۶-۲- کتاب)
- تمرین ۲: مسئله XOR توسط شبکه آدالاین شود. (دقت شود که پرسپترون تک لایه قادر به حل این مسئله نیست)

بهبود هایی بر LMS (مطالعه آزاد، بخش ۷-۸)

- نرخ یادگیری متغیر با زمان
- استفاده از مومنتوم

کاربرد شبکه آدالاین در فیلترهای تطبیقی

- کاربردهای عملی فراوان در پردازش سیگنال دیجیتال
- استفاده از بلوک TDL (Tapped Delay Line) در ورودی شبکه
- حذف تطبیقی نویز (بررسی در کلاس حل تمرین)

شبکه های عصبی

استاد: محمد باقر منهاج

فصل ۸

شبکه های عصبی چند لایه پیشخور و یادگیری پس انتشار خطا

موضوعات

- **فصل هشتم**
  - مقدمه
  - شبکه پرسپترون چند لایه
  - حل چند مسئله
  - الگوریتم پس انتشار خطا (BP)
  - تقریب توابع
  - کاربردهای شبکه عصبی چند لایه

مقدمه

- در شبکه های MLP
  - هر نرون دارای تابع تحریک غیرخطی است.

- یادگیری پس انتشار خطا (BP):
  - برای آموزش شبکه های عصبی چند لایه پرسپترون (MLP)
  - تعمیمی از الگوریتم LMS ، لذا تقریبی از SD و در چارچوب یادگیری عملکردی است.
  - مبتنی بر قانون یادگیری اصلاح خطا

- اختلاف LMS و BP
  - نحوه محاسبه و استفاده از مشتقات
  - برخلاف آدالاین، بردار خطا تابعی غیرخطی از پارامترهای شبکه بوده و مشتقاتش به سادگی قابل محاسبه نیست.

مقدمه

- قانون BP از دو مسیر اصلی تشکیل شده
  - مسیر رفت (سیگنالهای تابعی) :
    - حرکت سیگنالها از چپ به راست
    - اعمال بردار ورودی به شبکه و مشاهده اثرات در لایه خروجی از طریق انتشار در لایه های میانی
    - در این مسیر، پارامترهای شبکه ثابت و بدون تغییر می مانند.
  - مسیر برگشت (سیگنالهای خطا) :
    - حرکت سیگنالها از راست به چپ
    - توزیع خطا در کل شبکه پس از محاسبه در خروجی از طریق لایه های میانی (پس انتشار خطا)
    - تنظیم پارامترهای شبکه طبق قانون اصلاح خطا

تاریخچه

- دیدیم که شبکه های تک لایه فقط برای طبقه بندی مسائلی که به طور خطی از هم مستقلند، کارایی دارد. ولی ۲۵ سال LMS
تعمیم الگوریتم برای شبکه های چند لایه
به طول انجامید.

- ۱۹۵۹ قانون SLPR
- ۱۹۶۰ قانون LMS
- نخستین توصیف BP: پاول وربز ۱۹۷۴
- ۱۹۸۵ راملهارت، هینتون، پارکر: BP

شبکه پرسپترون چند لایه

- معرفی کلی در فصل دوم

$$\underline{a}^0 = \underline{p}$$

$$\underline{a}^{l+1} = \underline{F}^{l+1}(W^{l+1}\underline{a}^l + \underline{b}^{l+1}),\ l = 0,1,...,L-1$$

– L تعداد لایه های شبکه

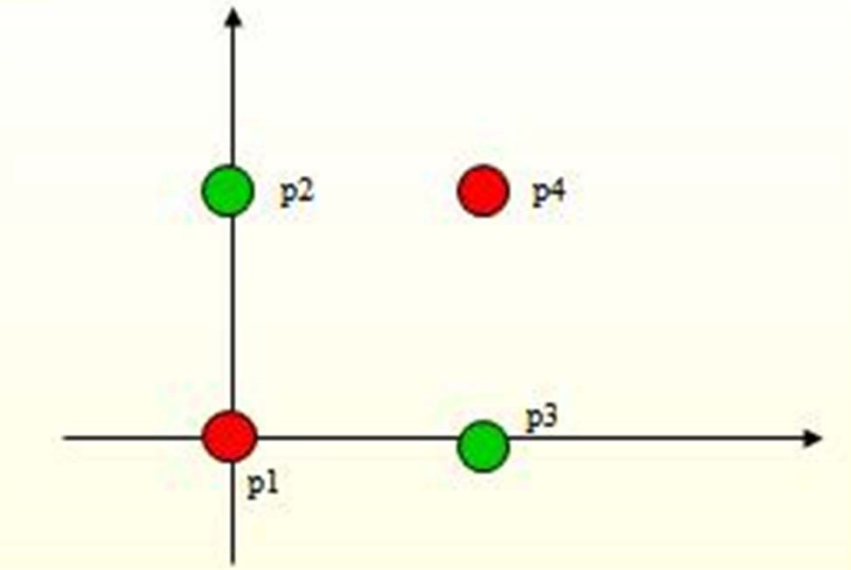
- نمایشی برای بیان ساختار شبکه چند لایه

$$(R - S^1 - S^2 - S^3)$$

– R تعداد ورودیها و S^i تعداد نورونها در لایه i ام

- هر نرون در شبکه MLP دو محاسبه بر عهده دارد:
  - محاسبه سیگنال تابعی
  - محاسبه تخمین لحظه ای از گرادیان خطا نسبت به پارامترهای ورودی نرون

حل چند مسئله شناسایی الگو



- **مثال ۱) الگوی xor**

$$(\underline{p}^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, t^1 = 0), (\underline{p}^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, t^2 = 1)$$

$$(\underline{p}^3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, t^3 = 1), (\underline{p}^4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, t^4 = 0)$$

- شبکه با ساختار (۱-۲-۲)
  - خط مرزی نرون اول به نحوی که الگوی p1 زیر ان و بقیه بالای ان باشند.
  - خط مرزی نرون دوم به نحوی که الگوی p4 زیر آن و بقیه بالای آن باشند.
  - خروجیهای لایه اول به ترتیب برابرند با [0;1]، [1;1]، [1;1] و [1;0] .
  - اگر لایه دوم به نوعی عمل AND را انجام دهد، مسئله حل خواهد شد. یعنی فقط در حالتی که تمام ورودیها ۱ است، خروجی مثبت شود.

شناسایی الگو

- یکی از (بینهایت) جوابهای مسئله به این قرار است:

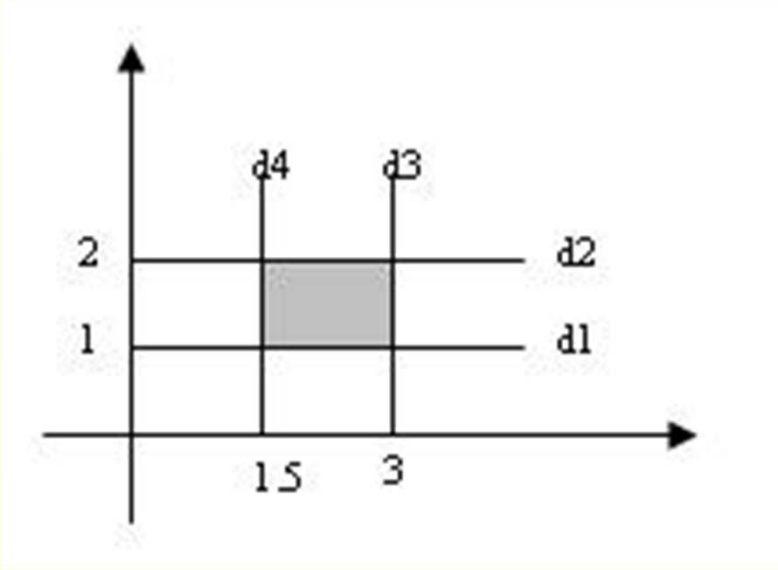
$$W^1 = \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix}, \underline{b}^1 = \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$$

$$W^2 = [1 \quad 1], b^2 = -1.5$$

- **مثال ۲:**

خروجی شبکه برای بخش خاکستری ۱

و برای بقیه جاها صفر باشد.



شناسایی الگو

- ناحیه مذکور توسط ۴ خط بیان می شود ← ساختار شبکه (۱-۴-۲)
- روش تعیین وزنها مثل قبل
  - لایه اول: معادلات خطوط
  - لایه دوم: انجام عمل AND
- جواب مطلوب:

$$W^1 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \\ -1 & 0 \\ 1 & 0 \end{bmatrix}, \underline{b}^1 = \begin{bmatrix} -1 \\ 2 \\ 3 \\ -1.5 \end{bmatrix}$$

$$W^2 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}, b^2 = -3.5$$

- تمرین: درستی پاسخها تست شود.

الگوریتم BP

- تعمیمی از الگوریتم LMS
- شاخص اجرایی مشابه LMS
  - S_L تعداد نرونها در لایه L

$$\hat{F}(k) = \sum_{j=1}^{S_L} e_j^{\,2}(k) = \underline{e}^T(k)\underline{e}(k)$$

- شاخص اجرایی تابعی از پارامترهای شبکه
- قانون یادگیری:

$$W_{ij}^l(k+1) = W_{ij}^l(k) - \alpha \frac{\partial \hat{F}(k)}{\partial W_{ij}^l(k)}$$

$$b_i^l(k+1) = b_i^l(k) - \alpha \frac{\partial \hat{F}(k)}{\partial b_i^l(k)}$$

- به سادگی محاسبه میشد ولی اینجا به دلیل توابع غیرخطی کار سخت تر است.

الگوریتم BP

- جهت محاسبه ترم اصلاحی نیاز به سیگنال خطا داریم
  - برای نرونهای خروجی (به دلیل قابل رویت بودن) سیگنال خطا در دسترس است.
  - نرونهای لایه های میانی (نرونهای مخفی) قابل رویت نبوده و سیگنال خطا برای آنها قابل اندازه گیری نمی باشد.
- توجه:

تمام نرونهای میانی در مقدار خطا در لایه خروجی سهیمند.

الگوریتم BP هر نرون را با توجه به میزان اثرش در بردار خطا تشویق یا تنبیه می کند.

خلاصه الگوریتم BP

- مسیر رفت:

$$\underline{a}^0 = \underline{p}(k)$$

$$\underline{a}^{l+1}(k) = \underline{F}^{l+1}(\underline{n}^{l+1}(k)),\ \ l = 0,1,\ldots,L-1$$

$$\underline{n}^{l+1}(k) = W^{l+1}(k)\underline{a}^l + \underline{b}^{l+1}(k)$$

$$\underline{a}(k) = \underline{a}^L(k)$$

- ملاحظه می شود که پارامترهای شبکه ثابت و بدون تغییر می مانند.
- مسیر برگشت:
  - شروع کار از لایه آخر (لایه خروجی) است.
  - بردار های حساسیت خطا از لایه آخر به لایه اول برگشت داده می شوند.
  - گرادیان محلی، نرون به نرون با الگوریتم بازگشتی محاسبه می شود.
  - در این مسیر نیز پارامترهای شبکه ثابت هستند.

خلاصه الگوریتم BP

- مسیر برگشت:

$$\underline{\delta}^{L}(k) = -2\dot{F}^{L}(\underline{n}^{L})\underline{e}(k)$$

$$\underline{\delta}^{l}(k) = \dot{F}^{l}(\underline{n}^{l})(W^{l+1})^{T}\underline{\delta}^{l+1}(k), \quad l = L-1,\ldots,1$$

$$\underline{e}(k) = \underline{t}(k) - \underline{a}(k)$$

$$\dot{F}^{l}(\underline{n}^{l}) = diag\left(\dot{f}^{l}(n_1^{l}), \dot{f}^{l}(n_2^{l}),\ldots, \dot{f}^{l}(n_{S_l}^{l})\right)$$

- تنظیم پارامترها:

$$W^{l}(k+1) = W^{l}(k) - \alpha\underline{\delta}^{l}(k)(\underline{a}^{l-1}(k))^{T}$$

$$\underline{b}^{l}(k+1) = \underline{b}^{l}(k) - \alpha\underline{\delta}^{l}(k) \quad , l = 1,2,\ldots,L$$

خلاصه الگوریتم BP

- توقف الگوریتم:
  - میانگین مربعات خطا در هر سیکل یا epoch (جمع مربعات خطا برای تمام الگوهای یادگیری) کمتر از مقدار از پیش تعیین شده ای باشد

    دقت شود که تعداد تکرارها هر سیکل برابر تعداد داده های یادگیری است.

  - نرم گرادیان خطا از مقدار از پیش تعیین شده ای کمتر باشد.

- محدودیت BP: توابع تبدیل باید مشتق پذیر باشند.
- الگوریتم BP به صورت دسته ای نیز قابل اجرا است.
-