تاریخ دریافت: ۸۸/۲/۲۲
تاریخ پذیرش: ۸۸/۵/۲۰

آشنایی با روش‌های رایانش نرم در مدل‌سازی پدیده‌ها

دکتر جمشید صالحی صدقیاتی*
سید امیر رضا ابطحی**

چکیده

هدف از تدوین این مقاله معرفی رایانش نرم و روش‌های مختلف آن جهت مدل‌سازی پدیده‌ها می باشد. رایانش نرم[1] به مجموعه در حال تکاملی از روش‌هایی گفته می‌شود که جهت ساخت سیستم‌های هوشمندی که همانند انسان استدلال می‌کنند و قابلیت مواجهه با عدم قطعیت در مسائل را دارند، مورد استفاده قرار می‌گیرند.

در این مقاله در ابتدا به توضیح شبکه‌های عصبی در رویکرد رایانش نرم پرداخته می‌شود، سپس روش‌های دیگر این رویکرد از قبیل الگوریتم ژنتیک و کلاً مبحث یادگیری ماشینی توضیح داده خواهد شد. از آنجائیک هدف اصلی در ساخت مدل کشف دانش از آن می‌باشد، در ادامه روش‌های مختلف کشف دانش و قانون از شبکه‌های عصبی بیان خواهد شد.

واژگان کلیدی: رایانش نرم، کشف دانش، یادگیری ماشینی، بهره هوش ماشینی، شبکه عصبی، الگوریتم ژنتیک

---

* دانشیار، گروه مدیریت صنعتی دانشگاه علامه طباطبایی
** کارشناسی ارشد مدیریت صنعتی از دانشگاه آزاد اسلامی واحد علوم و تحقیقات، تهران

1- Soft Computing

مقدمه

رایانش نرم مجموعه در حال تکامل از روش‌هایی است که جهت ساخت سیستم‌های هوشمندی که همانند انسان استدلال می‌کنند و قابلیت مواجهه با عدم قطعیت در مسائل را دارند، به کار می‌روند. استفاده از این رویکرد منجر به ایجاد سیستم‌هایی با بهره هوشی ماشینی[1] بالا شده است. (*Bonissone,2007:16*)

به نظر می‌رسد که روش‌های رایانش نرم بسیار موفق‌تر از روش‌های کلاسیک در مدل‌سازی، استدلال و تکنیک‌های جستجو در دامنه گسترده‌ای از مسائل عمل می‌کنند. مشخصات مسائلی که رویکردهای تحلیل سنتی در حل آنها ضعیف هستند را می‌توان به صورت زیر دسته‌بندی کرد:

- **مسائل مدل‌سازی:** معمولاً مسائل موجود در دنیای واقعی به اندازه کافی تعریف شده نیستند و اطلاعات تجربی به صورت الگوهای ورودی- خروجی، بخش‌هایی از رفتار مسأله را نمایش می‌دهند. مدل‌های دقیق و صریح ریاضی برای چنین مسائلی یا اصولاً وجود ندارند یا تولید آنها پرهزینه و با محدودیت همراه است. به علاوه این مدل‌ها در مواردی رفتارهای غیرخطی از خود نشان می‌دهند که ابزارهای مدل‌سازی ریاضی سنتی در مدل‌سازی آنها با محدودیت‌هایی همراه هستند.
- **فضای حل گسترده:** مسائل دارای فضای حل گسترده، معمولاً با تکنیک‌های قطعی جستجوی[2] بهترین راه‌حل، قابل حل نیستند. زمان و تلاش‌های لازم برای محاسبه بسیار زیاد است و تکنیک های قطعی جستجو معمولاً در یافتن گزینه‌های بهینه محلی چندان موفق نیستند.
- **کسب دانش:** دانش متخصصان در مورد یک مسأله، معمولاً مبهم[3]، همراه با تعاریف غیردقیق و تخمین‌های غیرقطعی است. بنابراین، دسته‌بندی‌های قابل درک و تعاریف شفاف عمدتاً در دسترس نیستند. همچنین، در برخی موارد نیاز

1- Machine Intelligence Quotient- MIQ
2- Deterministic Search Techniques
3- Fuzzy

است که دانش به طور مستقیم از داده‌های مسأله و بدون مداخله انسان به دست آید.

رایانش نرم شامل مجموعه‌ای از روش‌ها با قابلیت مواجهه با مسائل دارای مشخصات ذکر شده در بالا جهت ایجاد سیستم‌های هوشمند قابل اطمینان و قابل پیش‌بینی با هزینه پائین می‌باشد. زیر مجموعه‌هایی از رایانش نرم شامل موارد متعددی از قبیل تئوری مجموعه فازی[1]، شبکه‌های عصبی، روش‌های بهینه‌سازی تصادفی (مانند الگوریتم ژنتیک) و تکنیک‌های یادگیری ماشینی می‌باشد.

رایانش نرم با ترکیب این روش‌ها که در حل مسائل مکمل یکدیگر هستند، ساختار و چارچوبی برای ساخت سیستم‌های هوشمند ترکیبی ایجاد می‌نماید. از نقاط قوت کلیدی این روش‌ها می‌توان به موارد زیر اشاره کرد:

- تئوری فازی، توانایی به دست آوردن قواعد اگر- آنگاه فازی[2] از دانش غیردقیق ارائه شده از طرف متخصصان را می‌دهد.
- شبکه‌های عصبی با استفاده از قابلیت مدل‌سازی غیرخطی خود، اجازه یادگیری و انطباق رفتارهای سیستم‌های غیرخطی را می‌دهد.
- الگوریتم ژنتیک یک جستجوی سرتاسری سیستماتیک را در فضای جواب ارائه می‌کند و می‌تواند با شروع از جواب‌های اولیه تصادفی به جواب‌های بهتر برسد.
- روش‌های یادگیری ماشینی ابزار مهم و مناسبی برای کسب دانش به صورت خودکار می‌باشند. قابلیت‌های ذکر شده به روش‌های رایانش نرم توانایی مواجهه موفقیت‌آمیز با مسائل متعدد دنیای واقعی در فرایندهای پیچیده سازمانی، رباتیک، فضانوردی، کنترل فرایند و تولید را می‌دهد (Dote and Ovaska, 2001:8).

در بخش‌های بعدی این مقاله توضیحاتی کلی در مورد روش‌های مختلف مورد استفاده در رایانش نرم جهت آشنایی بهتر با این مبحث آورده می‌شود.

1- Fuzzy Set Theory
2- Fuzzy if-then Rules

الگوریتم ژنتیک

الگوریتم ژنتیک[1] که برای اولین بار توسط جان هلند ارائه شد، شامل یک سری از روش‌های جستجوی تصادفی جهت استفاده در مسائل بهینه‌سازی می‌باشد. اساس این روش، در اصول ژنتیک و فرایندهای تکامل در طبیعت ریشه دارد. به عنوان یکی از ابزارهای بهینه‌سازی چند منظوره، الگوریتم ژنتیک نسبت به روش‌های بهینه‌یابی متداول دارای مزایای فراوانی است. یکی از مشخصه‌های منحصر به فرد الگوریتم ژنتیک این است که به اطلاعات مشتق و گرادیان توابع برای پیدا کردن جهت جستجوی نقطه بهینه، نیازی نیست. این مزیت، استفاده از الگوریتم ژنتیک را به عنوان یک ابزار منعطف برای بهینه کردن تعداد زیادی از توابع هدف در دنیای واقعی، که مشتق‌پذیر نیستند یا محاسبه گرادیان آنها بسیار وقت‌گیر و پرهزینه است، امکان‌پذیر می‌نماید.

- روش الگوریتم ژنتیک در بهینه‌یابی

برای یک مسأله جواب یا فضای جستجو شامل تمام جواب‌های شدنی می‌شود. به هر نقطه در این فضا، یک کروموزوم گفته می‌شود، که دارای یک مقدار برازش است و معمولاً برابر با مقدار تابع هدف محاسبه شده در آن نقطه می‌باشد. الگوریتم ژنتیک، یک جامعه از کروموزوم‌ها به وجود می‌آورد، که به صورت متناوب در جهت تولید یک برازش بهتر تکامل پیدا می‌کند.

نسل بعدی با استفاده از عمل‌گرهای ژنتیک از قبیل جهش[2] و ترکیب[3] با استفاده از جامعه کنونی به وجود می‌آید. این کروموزوم‌ها با یک مقدار برازش بالاتر در خلق جامعه جدید مورد استفاده قرار می‌گیرند. این موضوع تضمین می‌کند که کروموزوم‌های موفق، ژن‌های خوب خود را به نسل بعدی انتقال دهند. جامعه مورد بحث مستمراً در جهت برازش بهتر، تکامل پیدا می‌کند و الگوریتم بعد از چندین نسل به بهترین کروموزوم می‌رسد.

1- Genetic Algorithm
2- Mutation
3- Cross-over

یادگیری ماشینی

توانایی یادگیری از مثال‌های مختلف و ساختن مدلی از جهان پیرامون، اساس هوش بیولوژیکی است. این مدل از جهان پیرامون، عموماً ضمنی و مفهومی است و به انسان قابلیت انطباق با تغییرات مستمر محیطی را می‌دهد که این یک موضوع ضروری برای ادامه حیات می‌باشد. هدف هوش مصنوعی[1] ساختن ماشین‌ها یا برنامه‌هایی است که توانایی یادگیری، انطباق با شرایط جدید مسأله و ارائه کردن چیزی شبیه به هوش انسانی را داشته باشند. بنابراین، یادگیری یک نکته کلیدی در کاربردهای عملی هوش مصنوعی است. یادگیری ماشینی به دنبال مطالعه روش‌هایی است که به وسیله آنها برنامه‌هایی برای کامپیوترها نوشته شود تا آنها قابلیت یادگیری را به دست آورند. تا به امروز تعداد زیادی از این الگوریتم‌ها و برنامه‌های کامپیوتری نوشته شده و به طور موفقیت‌آمیزی در قسمت‌های مختلف کاربردی که نیاز به یادگیری در آنها وجود دارد به کار برده شده است. از جمله این مسائل کاربردی می‌توان به تشخیص صدا، بازی‌های کامپیوتری هوشمند، معاینه و تشخیص پزشکی، پیش‌بینی در مسائل مالی و کنترل صنعتی اشاره نمود.

سیستم یادگیری با مجموعه‌ای از مثال‌ها که به فرمت قابل تشخیص برای ماشین در آمده‌اند به نام مجموعه آموزش، تغذیه می‌شود. سپس سیستم یک مدل یا فرضیه برای انجام وظیفه‌اش (تشخیص الگو، دسته‌بندی، پیش‌بینی و ...) تولید می‌کند. حال بر این اساس که مدل بتواند مثال‌ها را طوری تعمیم دهد که اگر با داده‌ای خارج از مجموعه آموزش روبرو شد پاسخ صحیحی ارائه کند، می‌توان آن را ارزیابی کرد. بنابراین دقت پیش‌بینی یا تعمیم، یک معیار مهم در ارزیابی و انتخاب طرح‌های مختلف یادگیری ماشینی می‌باشد. معیار «قابلیت درک» در یک مدل دقیق به ما اجازه می‌دهد که دیدگاه درستی از دامنه مسأله به دست آوریم. این معیار به معنای میزان سهولت درک مدل به وسیله کاربر انسانی است و به اعتبارسنجی[2] و پالایش و کسب دانش کمک

1- Artificial Intelligence
2- Validation

می‌کند. یادگیری استنتاجی با تمرکز بر قابلیت درک مدل[1] یک مفهوم اساسی در گسترش و اجرای مفهوم کسب دانش از پایگاه داده‌ها و داده کاوی می‌باشد.
اگر چه تعدد انتخاب میان روش‌های مختلف یادگیری ماشینی وجود دارد، اما می‌توان این روش‌ها را براساس مفاهیمی چون دقت پیش‌بینی، قابلیت درک و سهولت اجرا، برای مسائل مختلف ارزیابی کرد. در دو بخش بعدی دو تئوری مهم یادگیری ماشینی، یعنی استنتاج درخت تصمیم و استنتاج خصیصه‌گرا[2] توضیح داده می‌شود. شبکه عصبی که یکی از ابزارهای عملی برای یادگیری ماشینی می‌باشد و در این تحقیق مورد استفاده قرار گرفته است در یک بخش جداگانه توضیح داده خواهد شد.

- استنتاج درخت تصمیم

درخت تصمیم معروف‌ترین الگوریتم یادگیری ماشینی است. درخت تصمیم فرضیه‌های یادگیری یا مقدار تابع هدف را با ساختار درختی منحصر به فردی ارائه می‌کند. این الگوریتم جهت ایجاد قابلیت درک بهتر می‌تواند به راحتی به صورت قواعد اگر- آنگاه در آید. درخت‌های تصمیم در جاهای زیادی از مسائل معاینه و تشخیص پزشکی گرفته تا تعیین میزان ریسک در سرمایه‌گذاری مورد استفاده قرار گرفته‌اند. مثال ارائه شده در جدول ۱ داده‌های آموزش برای مفهوم هدف که انجام بازی تنیس است را نمایش می‌دهد. این درخت تصمیم می‌خواهد مشخص کند که شنبه صبح برای انجام بازی تنیس مناسب است یا خیر.
هر گره، در این درخت به وسیله یک بیضی یک منطق موجود در مسأله را مشخص می‌کند. درخت، یک گره ریشه به نام وضع هوا، با سه مقدار ممکن- آفتابی، ابری و بارانی- دارد. هر کدام از شاخه‌های خروجی از یک گره با یکی از مقادیر ممکن ذکر شده مرتبط می‌شود. بنابراین، گره ریشه دارای سه شاخه است. درخت دارای مجموعه‌ای از گره‌های برگ می‌باشد که نشان‌دهنده خروجی تصمیم درخت می‌باشد. (در اینجا تصمیم این است که بازی تنیس انجام شود یا خیر). گره‌های برگ دارای شاخه نیستند. یک نمونه از طبقه‌بندی به وسیله پیمایش درخت با شروع از گره ریشه تا یک گره برگ

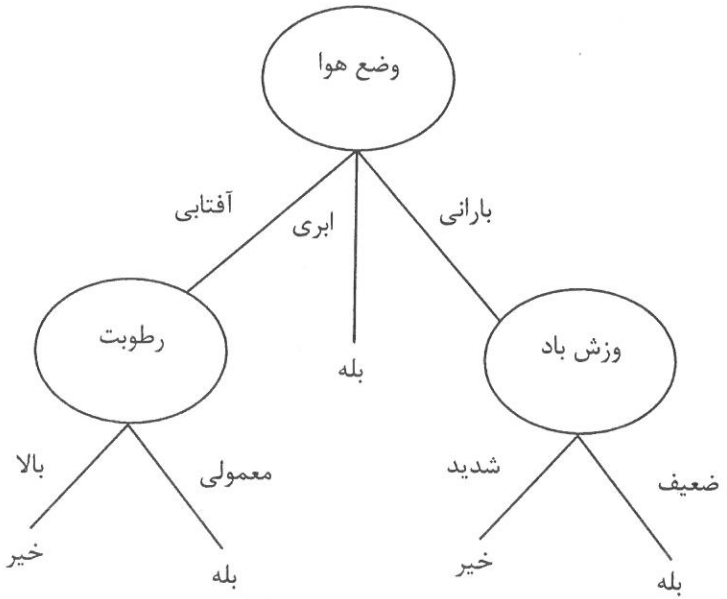
1- Model Comprehensibility
2- Attribute Oriented Induction

صورت می‌پذیرد. یک مثال نمونه (وضع هوا= آفتابی، رطوبت= بالا) با پیمایش زیر درخت چپ ریشه همانند شکل ۱ به دست می‌آید.

وضع هوا	دمای هوا	رطوبت	وزش باد	انجام بازی تنیس
آفتابی	گرم	بالا	ضعیف	خیر
آفتابی	گرم	بالا	شدید	خیر
ابری	گرم	بالا	ضعیف	بله
بارانی	متوسط	بالا	ضعیف	بله
بارانی	خنک	معمولی	ضعیف	بله
بارانی	خنک	معمولی	شدید	خیر
ابری	خنک	معمولی	شدید	بله
آفتابی	متوسط	بالا	ضعیف	خیر
آفتابی	خنک	معمولی	ضعیف	بله
بارانی	متوسط	معمولی	ضعیف	بله
آفتابی	متوسط	معمولی	شدید	بله
ابری	متوسط	بالا	شدید	بله
ابری	بالا	معمولی	ضعیف	بله
بارانی	متوسط	بالا	شدید	خیر

جدول ۲: مجموعه آموزش برای مفهوم بازی تنیس (Quinlan, 1996:81)

درخت پیش‌بینی می‌کند که مفهوم هدف- انجام بازی تنیس= خیر- نشانگر نامناسب بودن شرایط جوی جهت انجام بازی تنیس است. همچنین دیده می‌شود که از مشخصه دمای هوا در ساختن درخت تصمیم استفاده نشده است که نشانگر بی‌اهمیت بودن آن در فرایند تصمیم‌گیری است.



شکل ۱: نمایش درخت تصمیم بازی تنیس (Quinlan, 1996:81)

در دو دهه اخیر تعداد زیادی از روش‌های استنتاج درخت تصمیم، با شرایط و قابلیت‌های مختلف ایجاد شده‌اند. الگوریتم *ID3* یک الگوریتم محوری در اکثر الگوریتم‌های تولید شده است.

این الگوریتم درخت تصمیم را با یک روش بالا به پائین به وسیله تفکیک بازگشتی در هر گره پیمایش می‌کند. تعیین یک مشخصه برای شروع پیمایش، یک مسأله مهم در استنتاج درخت تصمیم است. الگوریتم *ID3* از یک مشخصه آماری به نام بهره اطلاعات[1] جهت انتخاب مشخصه‌های مورد نظر برای ساخت درخت استفاده می‌کند.

1- Information gain

بهره اطلاعاتی، یک معیار است که میزان اثربخشی یک مشخصه را در طبقه‌بندی و تصمیم‌گیری درخت تصمیم در میان مجموعه آموزش مشخص می‌کند.
S را به عنوان مجموعه آموزش در نظر می‌گیریم. بهره اطلاعاتی *(Infogain (T))* با در نظر گرفتن مشخصه T به صورت زیر تعریف می‌شود. (*Craven , 2006:35*):

$$Infogain(T) = \inf o(S) - \inf o_T(S) \quad (۱)$$

در معادله (۱)، *info (S)*، میزان اطلاعات مورد نیاز برای طبقه‌بندی یک نمونه موجود در مجموعه S و $\inf o_T(S)$ معیار مربوطه بعد از تفکیک مجموعه S براساس مشخصه T می‌باشد. اگر مجموعه S دارای K حالت ممکن برای تفکیک برای K طبقه باشد، *info (S)* به صورت زیر تعریف می‌شود:

$$\inf o(S) = -\sum_{j=1}^{k} \frac{freq(C_j,S)}{|S|} Log_2\left(\frac{freq(C_j,S)}{|S|}\right) \quad (۲)$$

$freq(C_j,S)$ تعداد مثال‌های طبقه C_j می‌باشد. با انجام یک تفکیک براساس مشخصه T مقدار مورد انتظار برای n زیر مجموعه استنتاج شده برابر است با:

$$\inf o_T(S) = -\sum_{i=1}^{n} \frac{|S_i|}{|S|} \inf o(S_i) \quad (۳)$$

در عبارت بالا، S_i زیرمجموعه‌ای از S است.
رویه انتخاب مشخصه مورد نظر برای تفکیک به صورت بازگشتی برای همه گره‌های داخلی تکرار می‌شود. این فرآیند تا حصول یکی از شرایط زیر ادامه می‌یابد.
۱- تمامی مشخصه‌های در دسترس پیمایش شده باشند.

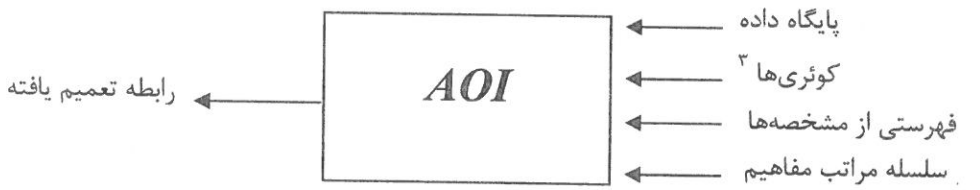
۲- نمونه‌های آموزش در گره مورد نظر به همان طبقه متعلق باشد. این گره برگ نام‌گذاری می‌شود.

- استنتاج خصیصه گرا

زمانی که مجموعه آموزش به صورت یک پایگاه داده باشد، عمل استنتاج فرضیاتی جهت توضیح داده‌ها، داده کاوی[1] نام دارد. در کاربردهای دنیای واقعی پایگاه داده‌ها به صورت بسیار گسترده‌ای جهت نگهداری و ارائه اطلاعات مورد استفاده قرار می‌گیرند که عموماً اطلاعات موجود ناکافی، غیرقطعی و ناقص هستند. نیاز روزافزون به کشف دانش از پایگاه داده‌ها منجر به گسترش سریع تکنیک‌های یادگیری ماشینی مناسب برای پایگاه داده‌ها شده است.

استنتاج خصیصه‌گرا[2]، تکنیکی برای کسب دانش از پایگاه‌های داده رابطه‌ای با استفاده از قواعد طبقه‌بندی و ویژگی‌هایی که فرضیه را توضیح می‌دهند، است. این روش یک روش مجموعه‌گرا است که زیرمجموعه داده‌ها را خصیصه به خصیصه تعمیم می‌دهد. استنتاج مشخصه‌گرا یک فرایند استنتاج داده محور است که توانایی تعمیم نتایج تا سطح انتزاع دلخواه را دارد.

این روش مفاهیم یادگیری ماشینی را مانند استقرا، تعمیم، سلسله مراتب مفاهیم و عملیات پایگاه داده‌ای را برای کشف قواعد، بهم آمیخته است. شکل ۲ ورودی‌ها و خروجی‌های روش *AOI* را نشان می‌دهد.



شکل ۲: طرح استنتاج مشخصه‌گرا (Han, 1996:399)

1- Data Mining
2- Attribute Oriented Induction- AOI
3- Queries

برای اینکه این روش بتواند روابطی را که کشف می‌کند به مسائل دیگر تعمیم دهد، نیاز دارد تا دانش کشف شده را به فرم خاصی به نام سلسله مراتب مفاهیم درآورد. این سلسله مراتب مفاهیم می‌تواند به طور صریح توسط متخصصان بیان شود یا از تحلیل داده‌ها به دست آید. *AOI* از سلسله مراتب مفاهیم جهت تولید قواعد منطقی استفاده می‌کند. در زیر مراحل اصلی الگوریتم در این روش استنتاجی آورده می‌شود:

- **درخت تکامل مفاهیم:** رابطه‌ها را به وسیله حذف چندتایی‌های برابر، تعمیم می‌دهد. برابر بودن این چندتایی‌ها را، مقادیر و مفاهیم از قبل تعریف شده مشخص می‌کنند.
- **شمارنده:** دانستن تعداد چندتایی‌هایی که در طی فرایند قبلی با یکدیگر ادغام شده‌اند، برای عمل یادگیری مهم است. یک شمارنده تولید می‌شود که نشاندهنده تعداد مفاهیم اولیه‌ای می‌باشد که به مفهوم کنونی تعمیم پیدا کرده‌اند.
- **ساده‌سازی:** رابطه تعمیم یافته با ادغام چندتایی‌های نزدیک بهم ساده می‌شود.
- **تبدیل قواعد:** از رابطه نهایی بدست آمده، یک قاعده منطقی به فرمت دلخواه کاربر تولید می شود.

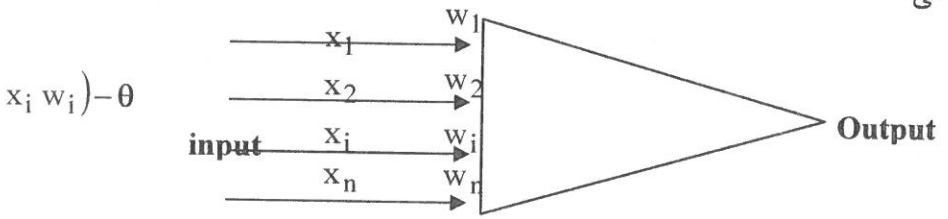
این روش توانایی تولید دو نوع قواعد استنتاجی را داراست: قواعد مشخصه یادگیری و قواعد طبقه‌بندی یادگیری که رویه‌های استنتاجی آنها مشابه هستند و تفاوت آنها در فرایند تعمیم مشخصه‌هایشان می‌باشد.

شبکه‌های عصبی مصنوعی

انگیزه ایجاد شبکه‌های عصبی در تمایلی که برای تقلید از کارکرد مغز انسان وجود دارد، ریشه داشت. شبکه عصبی یک ابزار مدل‌سازی هوشمند داده محور است که قابلیت کسب و ارائه روابط ورودی- خروجی غیرخطی سیستم‌ها را داراست. شبکه‌های عصبی کاربردهای بسیار مهمی از جمله تخمین توابع، تشخیص الگو، طبقه‌بندی، بازیابی حافظه، پیش‌بینی، بهینه‌سازی و ... دارند. شبکه‌های عصبی در محصولاتی همچون مودم، سیستم‌های پردازش تصویر، نرم‌افزارهای تشخیص صدا، داده کاوی، سیستم‌های کسب دانش و تجهیزات پزشکی و ... به کار رفته‌اند. (*Efraim et al, 2001:5*)

- محاسبات عصبی

یک شبکه عصبی از تعداد زیادی لایه‌های متشکل از چندین گره[1] تشکیل شده است. این گره‌ها بهم متصل شده‌اند و هر اتصال دارای وزن (W_i) مخصوص به خود است. وزن یک اتصال معیاری برای قدرت آن اتصال و علامت جبری آن نشان‌دهنده محرک یا بازدارنده بودن آن است. شکل ۳ یک نرون[2] دارای n ورودی را نمایش می‌دهد.



شکل ۳ - نرون با چند ورودی

این نرون دارای یک مقدار آستانه یا بایاس[3] θ است که میزان ورودی خالص جهت ایجاد یک خروجی غیر صفر را نشان می‌دهد. ورودی خالص نرون net_i به صورت زیر محاسبه می‌شود:

$$net_i = \sum w_i x_i + \theta \qquad (۴)$$

تابع انتقال، f ورودی را به خروجی تبدیل می‌کند.

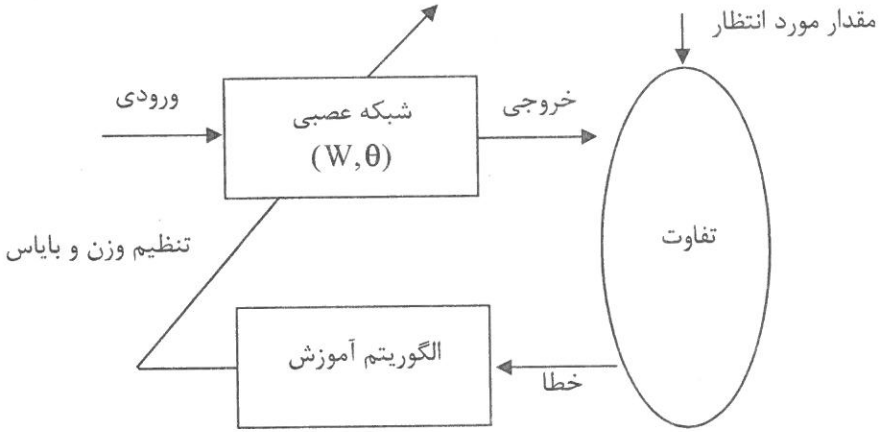
$$Output = f(net_i) \qquad (۵)$$

1- Node
2- Neuron
3- Bios

شبکه‌های عصبی دارای دو مرحله مجزای عملکردی هستند. تولید و آموزش. برخی از پارامترهای طراحی باید قبل از آموزش شبکه تعیین شوند:

- **توپولوژی[1] یا معماری سیستم:** تعداد گره‌های هر لایه و توابع انتقال مربوطه.
- **الگوریتم آموزش:** الگوریتم آموزش و معیار ارزیابی.
- **ملاحظات تعمیم پذیری:** تعداد دوره‌ها[2] لازم برای تضمین تعمیم‌پذیری مناسب و معیاری برای پایان دادن به مرحله آموزش.

وزن‌ها و بایاس‌ها در طول دوره آموزش تغییر می‌کنند. شبکه از داده‌های مسأله، جهت تخصیص مقدار به این پارامترها استفاده می‌کند. وجه تمایز شبکه‌های عصبی در قابلیت تطابق آن‌ها، با توجه به طراحی جریان اطلاعات منحصر به فردش که در شکل ۴ نمایش داده شده است می‌باشد.



شکل ۴ جریان اطلاعات برای مرحله آموزش (Principe et al, 2007:46)

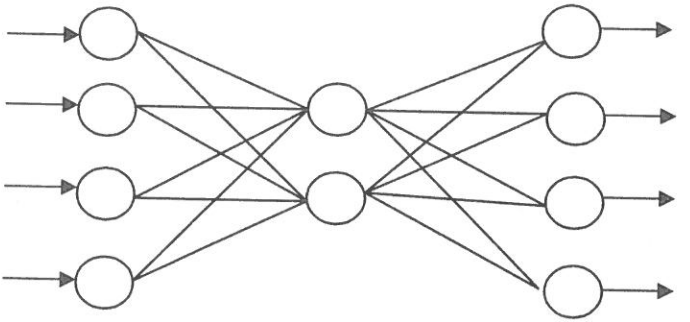
1- Topology
2- Epochs

- پرسپترون چند لایه

در پاسخ به عدم توانایی حل مسأله‌های جدایی‌ناپذیر خطی که یکی از نقاط ضعف پرسپترون بود، در ابتدا تصور می‌شد که شاید راه حل مشکل استفاده از تعداد بیشتری از یک پرسپترون باشد. در نگاه اول مشکلی به نظر نمی‌آمد، اما سپس مشخص شد که این ترکیب پرسپترون‌ها، هرگز قادر به فراگیری نیستند. بنابراین پرسپترون جدید بصورت لایه‌ای منظم شدند و به آن‌ها پرسپترون چند لایه گفته شد.(البرزی، ۱۳۸۶، ص۱۵)

پرسپترون چند لایه [1] شامل مجموعه‌ای از لایه‌های متشکل از چندین پرسپترون می‌باشد. شکل ۵ یک پرسپترون چند لایه را نمایش می‌دهد.

هر ورودی به یک گره در لایه ورودی متصل می‌شود. لایه پنهان ورودی را به یک فضای میانی که منطقه ورودی برای لایه خروجی است هدایت می‌کند.

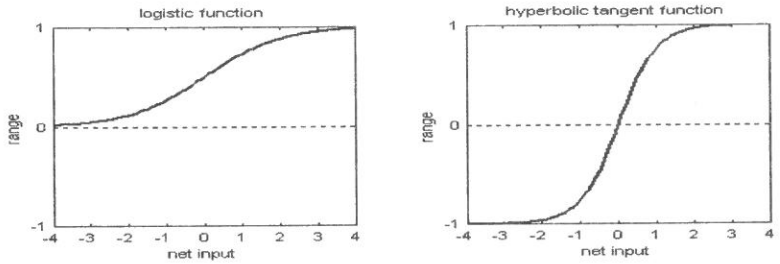


شکل ۵ پرسپترون چند لایه (البرزی،۱۳۸۶، صفحه ۶۳)

لایه خروجی، خروجی یا پاسخ را ارائه می‌دهد. گره خروجی همان طور که در شکل ۵ دیده می‌شود، اجازه می‌دهد که بتوان خروجی یا پاسخ را از متغیرهای ورودی، تشخیص داد. وقتی در پرسپترون‌های چند لایه از توابع انتقال غیرخطی استفاده شود، می‌توان از آنها برای تخمین توابع استفاده کرد. دو تابع تبدیل بسیار پراستفاده در شبکه‌های

1- Multi-layer Perceptron- MLP

عصبی تابع لجستیک[1] و تابع تانژانت هایپربولیک[2] می‌باشند و در این تحقیق مورد استفاده قرار گرفته‌اند. تفاوت این دو تابع در برد مقدار خروجی آنها همان طور که در شکل ۶ نشان داده شده است می‌باشد.



شکل ۶: تابع‌های انتقال لجستیک و تانژانت هایپربولیک (Hagan et al, 2006:36)

تابع لجستیک دارای برد خروجی [۱ و ۰] است و مقدار آن a_i به صورت زیر محاسبه می‌شود:

$$a_i = \frac{1}{1+e^{-\text{net input}}} \qquad (۶)$$

تابع تانژانت هایپربولیک دارای برد خروجی [۱ و ۱-] است و مقدار آن به صورت زیر است:

$$a_i = \frac{e^{\text{net input}} - e^{-\text{net input}}}{e^{-\text{net input}} + e^{-\text{net input}}} \qquad (۷)$$

- آموزش شبکه عصبی

مرحله آموزش در شبکه‌های عصبی پاسخی به این سؤال می‌دهد که آیا مجموعه‌ای از پارامترهای شبکه (وزن‌ها و بایاس‌ها) وجود دارد که به آن توانایی تبدیل مجموعه الگوهای ورودی به خروجی‌های مورد انتظار را بدهد؟ اگر هست، این مقادیر چه هستند.

1- Logistic Function
2- Hyperbolic Tangent

پراستفاده‌ترین الگوریتمی که برای آموزش شبکه عصبی استفاده می‌شود الگوریتم پس انتشار[1] می‌باشد. واژه «پس‌انتشار» به انتشار خطا به سمت عقب در شبکه‌های عصبی اشاره دارد. سیستم آموزش، وزن‌ها و بایاس‌های شبکه را طوری تنظیم می‌کند که تابع هزینه می‌نیمم شود. تعداد زیادی تابع هزینه وجود دارد، اما تابع مناسب برای مسائل پیش‌بینی تابع کراس آنتروپی[2] است.

$$E = \sum_{p}\sum_{i} t_{pi} \ln (y_{pi}) + (1 - t_{pi}) \ln (1 - y_{pi}) \qquad (۸)$$

در معادله (۸)، E میزان کراس آنتروپی، P تعداد الگوهای آموزش و i تعداد طبقه‌ها را نشان می‌دهند. y_{pi} احتمال تعلق الگوی ورودی به طبقه iام است و t_{pi} هدف دربازه [۱ و ۰] است. خروجی شبکه، احتمال تعلق الگوی ورودی به یک طبقه خاص است.

تابع هزینه، E، باید می‌نیمم شود که $\frac{\partial E}{\partial W}$ آن محاسبه می‌شود. با مشتق گرفتن از E، تنظیم وزن‌ها به یک مسأله بهینه‌یابی تبدیل می‌شود. الگوریتم پس انتشار از فرمول زیر برای بروز کردن وزن‌ها استفاده می‌کند.

$$\text{Weight Update}, \Delta W_{ij} = -\eta \frac{\partial E}{\partial W} \qquad (۹)$$

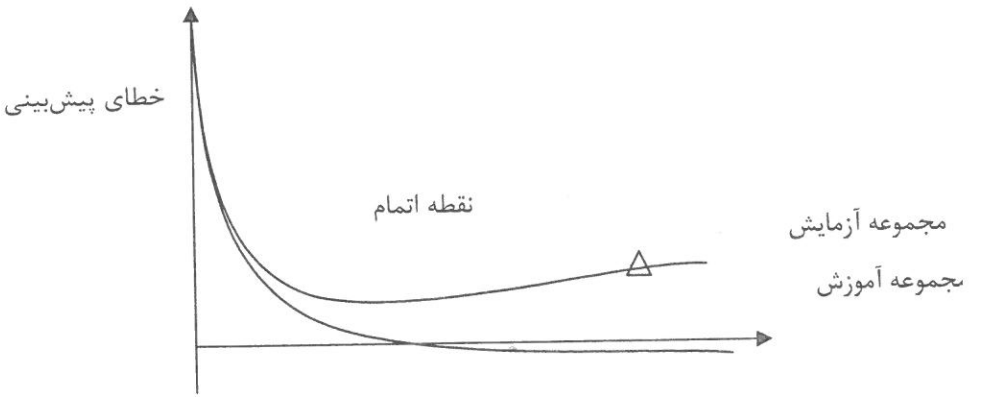
در این معادله، W_{ij}، وزن شاخه‌ی متصل کننده گره i به گره j است. η نرخ یادگیری را نشان می‌دهد و $\frac{\partial E}{\partial W}$، مشتق جزئی خطا E را بر حسب W_{ij} نمایش می‌دهد. در مرحله مقداردهی اولیه، مقادیر تصادفی به عنوان وزنهای شبکه در نظر گرفته می‌شوند و الگوریتم آموزش، این وزن‌ها را طبق رویه گفته شده تغییر می‌دهد. برای بهینه‌سازی

1- Backpropagation
2- Cross- entropy

تابع E روش‌های جایگزین زیادی وجود دارد از جمله روش گرادیان توأم، آموزش مومنتوم[1] و روش‌های دیگر. الگوریتم‌های جستجوی تصادفی نظیر الگوریتم ژنتیک نیز جهت جلوگیری از همگرایی به بهینه محلی و عدم یافتن بهینه مطلق، استفاده می‌شوند. *(Hagan et al, 1996: 57)*

- تعمیم‌پذیری

مجموعه‌ای از دوتایی‌های ورودی- خروجی مورد انتظار که به مجموعه آموزش معروف است جهت آموزش شبکه عصبی به کار می‌روند. مجموعه آزمایش نیز حاوی مثال‌هایی است که در دوره آموزش به شبکه ارائه نشده‌اند و برای ارزیابی قدرت تعمیم شبکه مورد استفاده قرار می‌گیرند. کارایی شبکه عصبی با الگوریتم آموزش پس انتشار زمانی افزایش می‌یابد که از چرخه‌های آموزش بیشتری در آن استفاده شده باشد. اگر چه مطابق شکل ۷، خطای پیش‌بینی در ابتدا با افزایش تعداد چرخه‌های آموزش کمتر می‌شود اما بعد از آن شروع به زیادتر شدن می‌کند.



شکل ۷، تعداد چرخه‌های آموزشی (Principe et al, 2007:174)

1- Momentum

به این پدیده، آموزش بیش از حد[1] گفته می‌شود و منجر به عدم تعمیم نتایج به صورت صحیح در شبکه می‌شود. یک راه‌حل برای این مسأله تقسیم مجموعه آموزش به دو مجموعه آموزش و اعتبارسنجی[2] است. بعد از تعداد مشخصی تکرار، خطای مجموعه اعتبارسنجی اندازه‌گیری می‌شود. زمانی که این خطا شروع به افزایش می‌کند، فرایند آموزش متوقف می‌شود. به این روش کراس ولیدیشن[3] گفته می‌شود.

استخراج دانش از شبکه‌های عصبی مصنوعی

بعد از آن که شبکه عصبی با توپولوژی خاص خود، ساخته و آموزش داده شد، مرحله بعدی مدل‌سازی فرایند، استخراج دانش از شبکه عصبی است. این دانش در قالب مقادیری برای وزن‌ها و بایاس‌ها در شبکه نهفته است و برای درک فرایند مدل‌سازی جهت پیش‌بینی و استخراج دانش، تحلیل این وزن‌ها و نحوه تأثیرگذاری ورودی بر خروجی ضروری است. هم‌چنین متغیرهای ورودی عموماً تأثیرات متقابلی روی همدیگر دارند. این تأثیرات متقابل بین ورودی‌ها روی خروجی یا پاسخ نهایی تأثیرگذار است، که شناسایی این تأثیرات متقابل نیز برای محقق مهم است. در بخش بعدی برخی از روش‌های مهم مورد استفاده در تعیین میزان تأثیر متغیرهای ورودی بر روی خروجی معرفی می‌شود.

- روش مشتق جزئی

روش مشتق جزئی شامل محاسبه مشتقات جزئی متغیرهای خروجی براساس مقادیر ورودی می‌باشد. دو نتیجه‌ای که می‌توان به وسیله این روش کسب کرد به این صورت است:

- شرحی از تغییرات خروجی‌ها براساس تغییرات جزئی در ورودی‌ها.
- طبقه‌بندی از تغییرات نسبی که هر متغیر بر روی خروجی شبکه دارد.

مشتق جزئی خروجی تولید شده توسط شبکه نسبت به ورودی محاسبه می‌شود تا بتوان شرحی از تغییراتی که در خروجی با تغییرات جزئی در متغیر ورودی به وجود می‌آید را به دست آورد. برای ساختار شبکه‌ای شامل یک لایه پنهان با n_h نرون و n_i ورودی با

1- Overtraining
2- Validation
3- Cross Validation

یک متغیر خروجی، مشتق جزئی متغیر پاسخ y_j برحسب ورودی x_j به این صورت است:

$$d_{ji} = S_j \sum_{h=1}^{nh} W_{ho}\ I_{hj}(1-I_{hj})\ W_{ih} \qquad (۱۰)$$

در این معادله S_j مشتق جزئی مقدار نرون خروجی نهایی بر حسب ورودی است. I_{hj} پاسخ نرون پنهان h ام W_{ho} ، W_{ih} وزن‌های بین نرون خروجی و h^{th} نرون پنهان و بین نرون ورودی i ام و نرون پنهان h^{th} می‌باشد.

بعد از انجام مراحل ذکر شده می‌توان نمودارهایی برحسب مشتقات جزئی محاسبه شده مربوط به متغیرهای ورودی رسم کرد که می‌تواند بصورت عینی میزان تاثیر هرکدام از متغیرهای ورودی و خروجی شبکه را نمایش دهد. معنی یکی از این نمودارها می‌تواند این باشد که: اگر مشتق جزئی منفی باشد، با افزایش متغیر ورودی مقدار متغیر خروجی کاهش پیدا می‌کند. برعکس، اگر مشتق جزئی مثبت باشد، متغیر خروجی در اثر افزایش مقدار متغیر ورودی مورد نظر، افزایش می‌یابد و رابطه آنها مستقیم است. نتیجه بعدی روش مشتقات جزئی تعیین میزان تاثیر هر کدام از ورودی‌ها بر خروجی شبکه است با این فرمول:

$$SSD_i = \sum_{j=1}^{n} (d_{ji})^2 \qquad (۱۱)$$

برای هر ورودی یک مقدار *SSD* محاسبه می‌شود. مقدار *SSD* طبقه‌بندی متغیرها براساس میزان تاثیر افزایش آنها بر روی متغیر خروجی در مدل را امکان‌پذیر می‌سازد. یک متغیر ورودی با مقدار بالای *SSD*، تاثیر بیشتری بر متغیر خروجی خواهد داشت.

- روش پر تورب

روش پرتورب[1] مربوط بـه اخـتلاف در متغیرهـای ورودی اسـت. در ایـن روش عمـداً در متغیرهای ورودی اختلال ایجاد می‌شود تا تأثیر آنها بر متغیـر خروجـی محاسـبه شـود. روش این الگوریتم به این صورت است که مقادیر یک متغیر ورودی خاص تغییر می‌کند، در حالیکه مقادیر سایر متغیرها ورودی ثابت نگـه داشـته مـی‌شـود. متغیـری کـه تـاثیر بیشتری بر خروجی شبکه داشته باشـد بـه عنـوان مهمتـرین متغیـر در مـدل شـناخته می‌شود.

میانگین مربعات خطا[2] با ایجاد اختلال بیشتر در ورودی انتخاب شـده بـالاتر مـی‌رود. در این‌جا هدف، محاسـبه میـزان تـاثیر تغییـرات کوچـک در هریـک از ورودی‌هـا بـر روی خروجی شبکه عصبی است. طبقه‌بندی متغیرهای ورودی براساس میزان اهمیـت آنهـا و تغییری که بر روی خروجی ایجاد می‌کند صورت می‌پذیرد.

- آنالیز حساسیت

آنـالیز حـساسیت[3]، تـاثیرات متقابـل میـان متغیرهـای ورودی و خروجـی در شـبکه را استخراج می‌کند. این روش اطلاعات مربوط به تاثیر متغیرهای ورودی را بـر روی پاسـخ شبکه در طول دوره آموزش جمع‌آوری می‌کند و این موضوع نـشاندهنده میـزان تـاثیر و اهمیت هر کدام از کانال‌های ورودی می‌باشد.

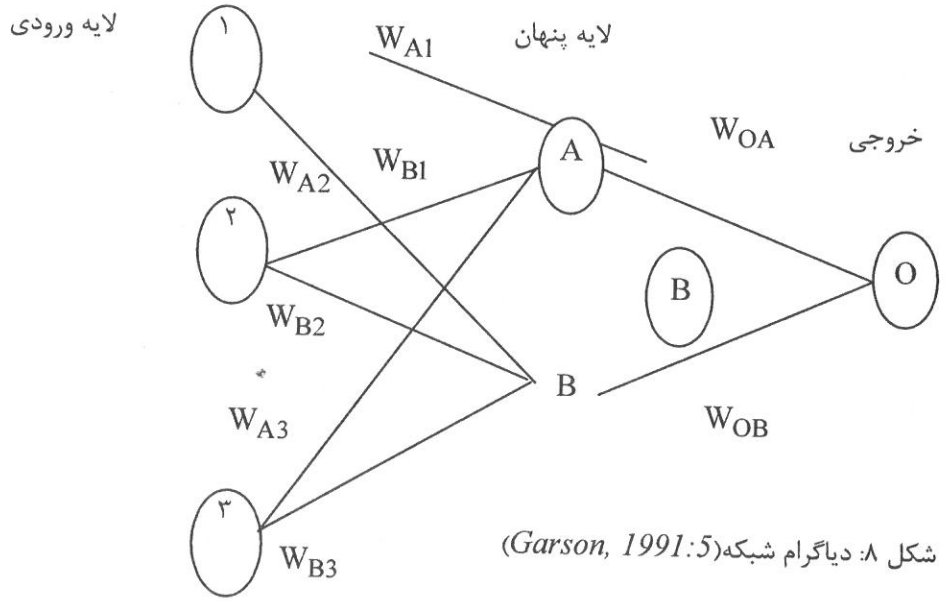
فضای ورودی به وسیله حذف کانال‌های ورودی بی‌اهمیت هـرس مـی‌شـود و در نتیجـه اندازه شبکه و به تبع آن پیچیدگی آن کاهش می‌یابد که این خود باعث کـاهش دفعـات لازم برای آموزش شبکه می‌شود. آنالیز حساسیت، به درک تـاثیر هریـک از ورودیهـا بـر خروجی تولید شده نیز کمک می‌کند. مقدار یکی از متغیرهـای ورودی در حـالی تغییـر می‌کند که بقیه متغیرها روی مقدار متوسط آنها، ثابت نگـه داشـته مـی‌شـوند. آمـوزش شبکه در طول این عملیات، جهـت اطمینـان از عـدم تغییـر وزن‌هـا متوقـف مـی‌شـود. پیچیدگی روش آنالیز حساسیت در صورت افزایش تعداد متغیرهای ورودی که قرار است تغییر کنند، بسیار زیاد می‌شود. پاسخ شبکه بر اساس مقدار تغییرات یـک متغیـر ورودی

1- Perturb Method
2- Mean Square Error - MSE
3- Sensitivity Analysis

خاص، ثبت می‌شود تا طیفی از خروجی‌های شبکه براساس تغییر در مقدار یک متغیر ورودی به وجود آید. این اطلاعات به درک تاثیرات نسبی هرکدام از متغییرهای ورودی بر روی خروجی کمک می‌کند و به سادگی می‌تواند به نمودار تبدیل شود.

- الگوریتم گارسن

همان طور که قبلاً گفته شد، یک شبکه عصبی اطلاعات را در قالب یک سری گره‌های بهم پیوسته با وزن‌های مختلف، پردازش و تأمین می‌کند. مقدار هر کدام از متغیرهای ورودی به صورت یک وزن به شبکه اعمال می‌شود. میزان تاثیر هرکدام از این متغیرهای ورودی بر روی خروجی به مقدار و جهت وزن هر شاخه که ورودی به وسیله آن به شبکه متصل شده است بستگی دارد. وزن مثبت در اتصال به شبکه، مقدار خروجی شبکه عصبی را افزایش می‌دهد در حالیکه وزن منفی، مقدار متغیر خروجی شبکه عصبی را کاهش می‌دهد. همچنین متغیری که دارای وزن اتصال بالاتری نسبت به بقیه متغیرها است، تاثیر بیشتری بر خروجی شبکه خواهد گذاشت. نگاشت بین متغیرهای ورودی و خروجی پیش‌بینی شده به وسیله یک *MLP*، یک فرایند دو مرحله‌ای در جریان اطلاعات است که شامل انتقال وزن از ورودی به لایه پنهان و سپس از پنهان به لایه خروجی می‌شود. یک حقیقت مهم که باید به آن اشاره کرد این است که زمانیکه وزن بین لایه ورودی – لایه پنهان و لایه پنهان – لایه خروجی از نظر جهت (مثبت یا منفی) یکسان باشد، تاثیر آن بر روی خروجی شبکه مثبت خواهد بود. با بررسی جهت و مقدار این وزن‌ها و شکستن آنها براساس ارتباطات لایه‌ای که قبلاً ذکر شد می‌توان به میزان تاثیر هر کدام از ورودیها بر روی خروجی پی برد. در سال ۱۹۹۱، گارسن الگوریتمی جهت درک بهتر و راحت‌تر این تاثیرات با توجه به وزن‌ها و جهت آنها ایجاد کرد. یک مثال از کاربرد الگوریتم گارسن در یک *MLP* با دو نرون در لایه داخلی در شکل ۸ آورده شده است. (*Garson, 1991:5*)



شکل ۸: دیاگرام شبکه(*Garson, 1991:5*)

مرحله ۱: ماتریس حاوی وزن اتصال نرون‌های ورودی، پنهان و خروجی مانند جدول ۲ تشکیل می‌شود.

ورودی	پنهان A	پنهان B
ورودی ۱	$W_{A1} = -2/61$	$W_{B1} = -1/23$
ورودی ۲	$W_{A2} = 0/13$	$W_{B2} = -0/91$
ورودی ۳	$W_{A3} = -0/69$	$W_{B3} = -2/09$
خروجی	$W_{OA} = 1/11$	$W_{OB} = 0/39$

جدول ۲: ماتریس وزن‌های اتصال

مرحله ۲: تأثیر هر یک از ورودی‌ها بر روی خروجی از طریق هر نرون پنهان مانند رابطه زیر محاسبه می‌شود. جدول ۳ بیانگر این موضوع است.

$$C_{A1} = W_{A1} \times W_{OA} = -2/61 \times 1/11 = -2/90 \quad (۱۲)$$

ورودی	پنهان A	پنهان B
ورودی ۱	$C_{A1} = -2/90$	$C_{B1} = -0/48$
ورودی ۲	$C_{A2} = 0/14$	$C_{B2} = -0/35$
ورودی ۳	$C_{A3} = -0/77$	$C_{B3} = -0/82$

جدول ۳: ماتریس تأثیر نرون‌های ورودی

مرحله ۳: تقسیم تأثیر وزنی هر یک از نرون‌های ورودی روی سیگنال خروجی بر مجموع کل آنها محاسبه شده و در جدول ۴ آورده شده است.

(۱۳)

$$R_{A1} = |C_{A1}| / (|C_{A1}| + |C_{A2}| + |C_{A3}|) = 2.9/(2.9 + 0.14 + 0.77) = 0.76$$

$$S_1 = R_{A1} + R_{B1} = 0.76 + 0.29 = 1.05 \quad (۱۴)$$

ورودی	پنهان A	پنهان B	مجموع
ورودی ۱	$R_{A1} = 0.76$	$R_{B1} = 0.29$	$S_1 = 1.05$
ورودی ۲	$R_{A2} = 0.04$	$R_{B2} = 0.21$	$S_2 = 0.25$
ورودی ۳	$R_{A3} = 0.20$	$R_{B3} = 0.50$	$S_3 = 0.70$

جدول ۴: تقسیم و مجموع تأثیر وزنی نرون‌ها

مرحله ۴: اهمیت نسبی هر متغیر ورودی محاسبه می‌شود. نتایج در جدول ۵ ارائه شده است .

$$I_1 = (S_1 /(S_1 + S_2 + S_3)) \times 100 = (1.05 /(1.05 + 0.25 + 0.70)) \times 100 = 52.5\%$$

ورودی	اهمیت نسبی
ورودی ۱	$RI_1 = 52.5\%$
ورودی ۲	$RI_2 = 12.5\%$
ورودی ۳	$RI_3 = 35.0\%$

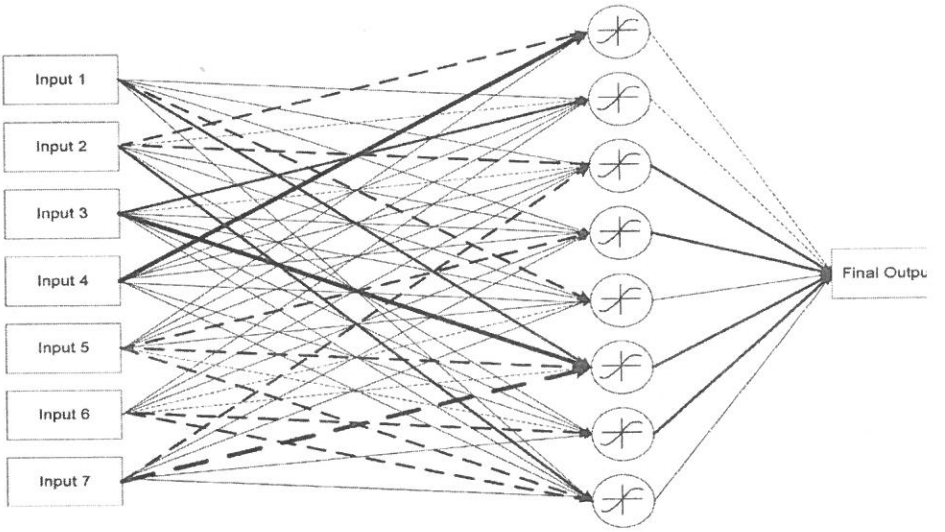
جدول ۵: اهمیت نسبی هر متغیر ورودی

- دیاگرام تفسیر شبکه

با افزایش تعداد متغیرهای ورودی در ساختار شبکه و تعداد لایه‌های نهان درک تأثیر هر یک از متغیرهای ورودی بر روی خروجی از روی مقدار و وزن اتصال آنها مشکل می‌شود. برای رفع این مشکل و ایجاد یک دیدگاه صحیح نسبت به شبکه از نظر ارتباط نرون‌ها و وزن اتصال آنها، دیاگرام تفسیر شبکه[1] ایجاد شد. (*Ozesmi,1999:16*) روش اساسی در این دیاگرام نمایش وزن‌های اتصال با استفاده از خطوطی که نرون‌ها را در هر لایه به یکدیگر متصل می‌کنند، می‌باشد. ضخامت خط، نشانده مقدار وزن در هر

1- Network Interpretation Diagram-NID

اتصال است. خطوط پیوسته نشانده علامت مثبت وزن و خط‌چین‌ها نشانده علامت منفی وزن هستند. یک دیاگرام تفسیر شبکه ساده می‌تواند به صورت شکل ۹ باشد.
زمانی که وزن‌های خطوط اتصال بین لایه ورودی- لایه پنهان و لایه پنهان- لایه خروجی هم علامت باشند، در *MLP* می‌توان با توجه به علامت، تأثیر مثبت یا منفی آن ورودی بر خروجی را حدث زد.



شکل ۹: دیاگرام تفسیر شبکه(Ozesmi,1999:16)

استخراج قانون از شبکه‌های عصبی

گسترش موارد استفاده از شبکه‌های عصبی مصنوعی در زمینه‌های مختلف مهندسی، علمی، صنعتی، اقتصادی و پزشکی به دلیل قابلیت‌های متعدد این روش است. سه ویژگی برجسته شبکه عصبی که موجب این گستردگی استفاده شده‌اند را می‌توان این چنین بر شمرد:

- روش به نسبت صریح و واضح شبکه عصبی در شناخت ابعاد مسأله در مرحله آموزش، یکی از این ویژگی‌ها است. قدرت درک روابط خطی و غیرخطی به شبکه

عصبی کمک می‌کند تا درک بهتری نسبت به مسأله پیدا کند. این فرایند بسیار ساده‌تر از فرایندهای پیچیده یادگیری در سیستم‌های دیگر هوش مصنوعی می‌باشد.

- قالب ساده و فشرده‌ای که شبکه عصبی برای ذخیره دانش به دست آمده‌اش در دوران آموزش استفاده می‌کند نیز یکی از ویژگی‌های آن می‌باشد (وزن‌ها و بایاس‌ها). همچنین سهولت نسبی دسترسی به این دانش و استفاده راحت از آن در تحلیل‌ها نیز بسیار مناسب است.
- قدرت شبکه عصبی در مواجهه با داده‌های دارای اختلال نیز یکی از ویژگی‌های بسیار مورد توجه این روش می‌باشد. این موضوع به ایجاد مدل‌های شبکه عصبی با دقت بالا در زمانی که حتی داده‌های ورودی دارای اختلال باشند کمک می‌کند.

مزیت دیگر یک شبکه عصبی آموزش داده شده، درجه بالای دقت آن در تعمیم کار به مثال‌هایی که در دوران آموزش با آنها مواجه نشده است، می‌باشد. البته علی‌رغم ویژگی‌های ذکر شده، عدم شفافیت کافی در نحوه به دست آوردن نتایج، نقطه ضعف شبکه عصبی محسوب می‌شود. ایده اصلی یادگیری و تعمیم، توضیح فرایند و ارائه دانش کسب شده است. برای این که شبکه‌های عصبی مقبولیت گسترده‌تری به عنوان یک ابزار یادگیری و تعمیم نتایج پیدا کنند، ضروری است که به تکنیکی که بتواند یک توضیح قابل درک از نحوه رسیدن به نتایج و روش‌های آن ارائه کند، مجهز شود. برای کاربردهای شبکه عصبی در موارد حساس ایمنی و حفاظت همچون خطوط هواپیمایی، تشخیص پزشکی، انتقال انرژی، طول عمر لوله‌های انتقال گاز و ...، شبکه عصبی باید دارای سه قابلیت مهم باشد.

۱- دادن امکان ارزیابی نتایج تولید شده به وسیله شبکه عصبی برای همه مقادیر ممکن متغیرهای ورودی به کاربر.
۲- دادن قابلیت به کاربر در جهت تعریف محدودیت برای مقادیر متغیرهای ورودی تا سیستم بتواند خروجی مورد انتظار را با قابلیت اطمینان بالا تولید نماید.
۳- دادن قابلیت تعیین ساختار و اندازه بهینه شبکه عصبی به کاربر.

شبکه عصبی باید در مورد نحوه کارکرد در لایه‌های داخلی خود به اندازه کافی به کاربر توضیح دهد. این موضوع می‌تواند موجب شود که کاربر روش‌هایی که ممکن است شبکه عصبی را در ارائه پاسخ درست دچار اشتباه کنند بشناسد و در صورت امکان آنها را حذف کنند و یا متوجه شود که چه زمانی و چرا یک نتیجه بهینه نبوده است.

- عمل استخراج قانون

شبکه عصبی در طول دوره آموزش خود، دانشی را کسب کرده و آن را به این صورت نگهداری می‌کند:

- معماری و توپولوژی شبکه
- توابع انتقال
- مجموعه‌ای از پارامترهای شبکه (وزن‌ها و بایاس‌ها)

این دانش فرضیات یا مدلی که توسط شبکه یاد گرفته شده است را ارائه می‌کند. معمولاً این مدل‌ها به خاطر تغییراتی که در وزن‌ها و بایاس‌ها و تعداد نرون‌ها و لایه‌هایی که صورت می‌گیرند، به سختی قابل درک هستند. ترجمه و تفسیر این مجموعه بزرگ از پارامترها به نمادها و مفاهیم قابل درک، کار ساده‌ای نیست، زیرا نگاشت ورودی‌ها به خروجی‌ها به وسیله بخش‌های پنهان داخلی شبکه عصبی صورت می‌گیرد.

هدف اصلی روش استخراج قانون از شبکه عصبی، تبدیل دانش کسب شده از شبکه به نمادهای قابل درک توسط کاربر به صورت قوانین است. گسترش روش‌های استخراج قوانین، محققان را مجبور به ارائه معیارهایی برای ارزیابی الگوریتم‌های پیشنهادی و نحوه ارائه دانش استخراج شده توسط آنها کرده است که در زیر خلاصه شده است:

- **قابلیت درک:** به اندازه‌ای که قوانین استخراج شده برای انسان قابل درک باشد.
- **قدرت بیان:** ساختار خروجی ارائه شده به کاربر نهایی مناسب باشد. روش‌های مختلف ارائه همچون گزاره‌های ساده‌ای از قوانین، قوانین فازی، درخت‌های تصمیم و ... می‌توانند براساس نوع مسأله انتخاب شوند.
- **دقت پیش‌بینی:** شبکه قابلیت تعمیم داده‌هایی که در دوران آموزش با آن‌ها مواجه شده است را داشته باشد.
- **صحت:** انطباق انچه ارائه می‌شود با آنچه که رفتار اصلی شبکه نشان می‌دهد.

- **مقیاس‌پذیری:** قابلیت استفاده از روش در مسائل با مقیاس‌های مختلف وجود داشته باشد.
- **عمومیت:** میزانی از تغییرات، محدودیت‌ها یا معماری شبکه خاص که یک روش استخراج قوانین به کاربر تحمیل می‌کند.

- روش‌های استخراج قانون

روش‌های استخراج قانون را می‌توان براساس دیدگاهی که در الگوریتم‌های توپولوژی شبکه نهفته است به سه دسته تقسیم‌بندی کرد:

- تجزیه‌ای[1]
- آموزشی[2]
- انتخابی[3]

در روش تجزیه‌ای، قوانین از طریق هر نرون در لایه‌های پنهان و خروجی استخراج می‌شوند. این قوانین سپس ترکیب می‌شوند تا رفتار کلی شبکه را توضیح دهند. این روش به عنوان یک روش محلی جهت استخراج قانون می‌باشد که براساس معماری شبکه بنا شده است. بیشتر روش‌هایی که در این رویکرد دسته‌بندی می‌شوند از یک رویه جستجو برای یافتن زیرمجموعه‌هایی از وزن‌های ورودی که بایاس و آستانه تحریک یک نرون را تحریک می‌کنند، استفاده می‌کنند. زیرمجموعه‌های یافته شده، به قوانین گزاره‌ای تبدیل می‌شوند.

روش‌های آموزشی، قوانین را استخراج می‌کنند که ورودی‌های شبکه را مستقیماً به خروجی‌های آن نگاشت می‌کنند. یک روش می‌تواند انتخاب قوانین مفید از مجموعه قوانین تولید شده با توجه به این که چه مقداری از ورودی می‌تواند خروجی را فعال نماید باشد.

در روش انتخابی، روش‌های تجزیه‌ای و روش‌های آموزشی با یکدیگر ترکیب شده‌اند. برای مثالی از این روش مثلاً در الگوریتم ژنتیک برای استخراج قانون، به ژن‌ها، مقدار وزن بین دو لایه همجوار اختصاص داده می‌شود. سپس کروموزوم‌ها جهت ارائه راهی از

1- Decompositional
2- Pedagogical
3- Electic

یک ورودی خاص به خروجی ساخته می‌شوند. تابع برازش به عنوان محصول انتقال وزن‌ها از ورودی به خروجی محاسبه می‌گردد. الگوریتم بهترین کروموزوم را شناسایی کرده و به قوانین اگر- آنگاه[1] ترجمه می‌کند.

- تحلیل فواصل اعتبار

یکی از پراستفاده‌ترین روش‌های استخراج قانون روش تحلیل فواصل اعتبار[2] است. رویه اصلی الگوریتم فواصل اعتبار به شرح زیر است.

۱- **تولید مجموعه قانون:** جهت استخراج قانون، مرحله اول، تولید مجموعه قوانین شدنی است.

۲- **اختصاص فواصل اعتبار:** یک قانون انتخاب شده به مجموعه‌ای از فواصلی که در آنها فعال است و می‌تواند ورودی را به خروجی تبدیل کند تبدیل می‌شود.

۳- **تنظیم فواصل:** با توجه به نتایج فواصل ذکر شده تنظیم می‌شود.

۴- **تأیید اعتبار قوانین:** دو خروجی ممکن از مراحل قبلی وجود دارد.

روش تحلیل فواصل اعتبار، یا یک قانون را معتبر می‌کند، یا یک تناقض به وجود می‌آید. در صورت به وجود آمدن تناقض مراحل ۲ تا ۴ برای یک قانون دیگر تکرار می‌شود.

- استفاده از درخت تصمیم جهت استخراج قانون

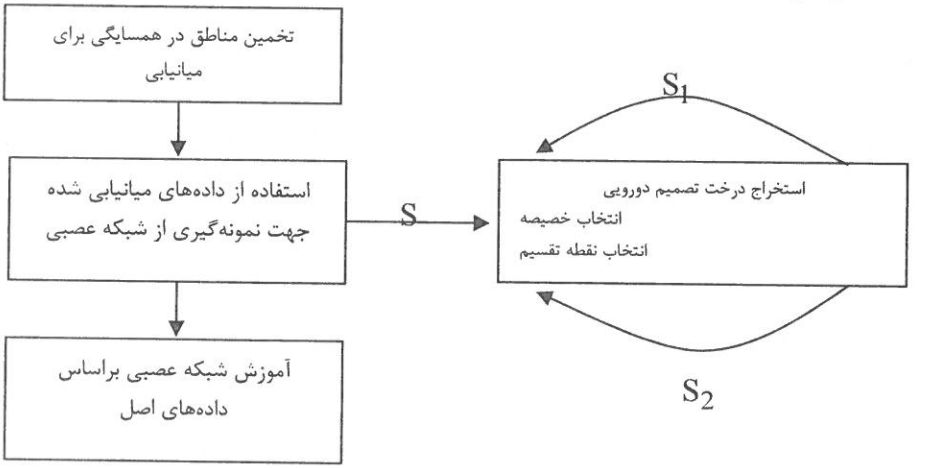
الگوریتم *ANN-DT* درخت‌های تصمیم یک متغیره را تولید می‌کند که شماتیک آنها در شکل ۱۰ ارائه شده است. همان طور که در شکل دیده می‌شود، مجموعه داده S براساس مشخصه انتخابی به دو مجموعه S_1 و S_2 تقسیم شده است. مراحل اصلی الگوریتم *ANN-DT* به قرار زیر است:

۱- **میانیابی داده‌های مرتبط:** یک مجموعه مجازی داده با نمونه‌گیری تصادفی از فضای با معنی متغیرهای ورودی تولید شده و نام‌گذاری می‌شوند.

۲- **انتخاب خصیصه:** برای خروجی‌های گسسته، از نسبت بهره اطلاعات جهت انتخاب خصیصه استفاده می‌شود.

1- If- then rules
2- Validity Interval Analysis

۳- **معیار توقف:** خصیصه انتخاب شده مجموعه داده موجود را به دو زیرمجموعه تقسیم می‌کند، با تقسیم داده‌ها به صورت بازگشتی، درخت تصمیم تولید می‌شود. برای خروجی‌های گسسته، زمانی که یک گره داخلی حاوی داده مربوط به یک طبقه خروجی شده فرایند متوقف می‌شود، برای خروجی‌های پیوسته، زمانی که واریانس استاندارد داده صفر شد، فرایند متوقف می‌شود.



شکل ۱۰: ارائه شماتیک از الگوریتم ANN-DT (Schmitz, 1999:18)

- الگوریتم ترپان

الگوریتم ترپان[1] مانند بقیه الگوریتم‌های استنتاج درخت تصمیم می‌باشد. با این تفاوت که در اینجا از یک شبکه آموزش داده شده استخراج دانش صورت می‌پذیرد و درخت تصمیم تولید شده تخمینی از شبکه می‌شود.

تفاوت اساسی میان ترپان و الگوریتم‌های درخت تصمیم دیگر، استفاده این الگوریتم از یک پیشگویی جهت ساخت کوئری‌های عضویت و ارائه طبقه‌بندی می‌باشد.

این اطلاعات در ساخت گره‌ها و برگ‌های درخت استفاده می‌شوند. عموماً، یک خصیصه برای قرار گرفتن در ریشه درخت انتخاب می‌شود. در مرحله بعد، به ازای هر مقدار

1- Trepan Algorithm

ممکن برای این خصیصه یک شاخه به آن اضافه می‌شود. فرایند اضافه کردن شاخه‌های درخت داده‌ها را به تعداد مشخصی زیرمجموعه تقسیم می‌کند، این فرایند به صورت بازگشتی بر روی هر شاخه تکرار می‌شود و تنها از داده‌هایی استفاده می‌شود که به شاخه ختم می‌شوند. اگر عدد محاسبه شده از مقدار آستانه گره کمتر شد، از پیشگویی استفاده می‌شود. اضافه کردن شاخه تا رسیدن همه ورودی‌ها به گره برگ ادامه می‌یابد. سپس این گره به عنوان یک طبقه در نظر گرفته می‌شود. طبقه‌بندی تا تمام شدن همه گره‌ها برگ ادامه می‌یابد. در شکل ۱۱ الگوریتم ترپان نشان داده شده است.

INPUT: ORACLE (), training set S, feature set F, *min_sample* parameter, stopping criteria

1. **for each** example $x \in S$
2. class label for x := ORACLE (x)
3. initialize the root of the tree, R, as a leaf node
4. construct a model M of the distribution of instances covered by node R
5. $query_instances_R$:= DRAWSAMPLE ({}, *min_sample* - $|S|$, M)
6. use S and $query_instances_R$ to determine class label for R
7. initialize *Queue* with tuple [R, S, $query_instances_R$, {}]
8. **while** *Queue* not empty and global stopping criteria not satisfied
9. remove [node N, S_N, $query_instances_N$, $constraints_N$] from head of *Queue*
10. T := CONSTRUCTTEST (F, S_N $\cup$ query $query_instances_N$)
11. make N an internal node with test T
12. **for each** outcome, t, of test T
13. make C, a new child node of N
14. $constraints_C := constraints_N \cup \{ T = t \}$
15. S_C := members of S_N with outcome t on test T
16. construct a model M of the distribution of instances covered by node C
17. $query_instances_C$:= DRAWSAMPLE ($constraints_C$, *min_sample* - $|S_C|$, M)
18. use S_C and $query_instances_C$ to determine class label for C
19. **if** local stopping criteria not satisfied **then**
20. put [C, S_C, $query_instances_C$, $constraints_C$] in *Queue*

RETURN: tree with root R

شکل ۱۱: الگوریتم ترپان (Craven et al, 2006:5)

خلاصه

به دلیل ویژگی‌هایی که برخی از مسائل در دنیای واقعی از خود نشان می‌دهند، از جمله مسائل مدل‌سازی، مسائل دارای فضای حل گسترده و همینطور مسائلی که در آن‌ها نیاز به کشف دانش از داده‌های مسأله بدون مداخله انسان وجود دارد، روش‌های کلاسیک مدل‌سازی به اندازه کافی موفقیت نداشته‌اند.

رایانش نرم با ترکیب روش‌های مختلفی از جمله تئوری مجموعه فازی، شبکه‌های عصبی، روش های بهینه‌سازی تصادفی و تکنیک‌های یادگیری ماشینی که در حل مسائل بصورت مکمل یکدیگر استفاده می‌شوند، ساختار و چارچوبی برای مواجهه موفقیت‌آمیز با مسائل دارای ویژگی‌های ذکر شده و ساخت سیستم های هوشمند ایجاد می‌کند.

در این مقاله، در ابتدا الگوریتم ژنتیک، یادگیری ماشینی و شبکه‌های عصبی مصنوعی به عنوان برخی از روش‌های رایانش نرم معرفی شده و سپس به روش‌های استخراج دانش و همینطور استخراج قانون از شبکه‌های عصبی مصنوعی پرداخته شده است.

منابع و ماخذ

۱- آر. بیل و تی. جکسون (۱۳۸۶)، **آشنایی با شبکه‌های عصبی**، ترجمه: محمود البرزی، چاپ دوم، تهران: مؤسسه انتشارات علمی دانشگاه صنعتی شریف

2- *Bonissone, P. P.,* ***Soft computing: the convergence of emerging reasoning technologies****, Soft Computing, 1(1), 2007.*

3- *Craven, M.,* ***Extracting comprehensible models from trained neural networks****, Ph.D. dissertation, University of Wisconsin, Madison, WI, 2006.*

4- *Craven, M.W., Shavlik, J. W.,* ***Extracting tree-structured representations of trained networks****, Advances in Neural Information Processing, Vol. 8, 2006.*

5- *Dote, Y., Ovaska, S. J.,* ***Industrial applications of soft computing: A review****, Proceedings of the IEEE, 89(9), 2001.*

6- *Efraim, T., Jay E. A., Liang T. P., McCarthy, R. V.,* ***Decision support systems and intelligent systems****, Prentice Hall, Upper Saddle River, NJ, 2001.*

7- *Garson, G.D.,* ***Interpreting neural network connection weights****, Artificial Intelligence Expert, 6, 1991.*

8- *Hagan, M. T., Howard B. D., Beal M,* ***Neural network design****, PWS Publishing Company, 2006.*

9- *Han, J., Fu, Y.,* ***Exploration of the power of attribute-oriented*** *induction in data*

10- ***mining in advances in knowledge discovery and data mining****, AAAI, MIT Press, Cambridge, MA, 1996.*

11- *Ozesmi, S. L., U. Ozesmi,* ***An artificial neural network approach to spatial habitat modelling with interspecific interaction****, Ecological Modelling, 116, 1999.*

12- *Principe, J. C., Euliano, E. R., Lefebvre, W. C.,* ***Neural and adaptive systems: Fundamentals through simulations with cd-rom****, John Wiley & Sons, Inc., New York, NY, 2007.*

13- *Quinlan, J. R.,* ***Induction of decision trees. Machine Learning****, Vol. 1, 1996.*

14- *Schmitz, G. P. J., Aldrich, C., Gouws, F. S.,* ***ANN-DT: An algorithm for extraction of decision trees from artificial neural networks****, IEEE Transactions on Neural Networks, Vol. 10(6), 1999.*