3rd International Conference on
Management and Humanistic Science Research
5 July 2018 - Tehran University
سومین همایش بین المللی پژوهش های مدیریت و علوم انسانی
۱۴ تیر ۱۳۹۷ دانشگاه تهران

انواع شبکه های عصبی و کاربرد آنها

شاهین بیرانوند
دانشجوی دکتری، دانشکده مهندسی، دانشگاه آزاد اسلامی، واحد نجف آباد، نجف آباد، ایران
Shahin254@gmail.com

کیانا صحرائیان
دانشجوی دکتری، دانشکده مهندسی، دانشگاه آزاد اسلامی، واحد نجف آباد، نجف آباد، ایران
k.sahraian@gmail.com

چکیده

شبکه‌های عصبی مصنوعی که امروزه در کاربردهای فراوانی ارزش بالای خود را نشان داده‌اند، بر اساس مدل بیولوژیکی مغز جانوران بوجود آمده‌اند. این شبکه‌ها به واقع یک سیستم داده پردازی اطلاعات است که دارای خصوصیات اجرائی خاصی همانند شبکه‌های عصبی جانوری می‌باشد که از تعمیم یافتن مدل‌های ریاضی آن ها به وجود آمده‌اند. شبکه های عصبی مصنوعی ، شاخه ای از هوش مصنوعی است که یک روش مناسب برای تشخیص الگوهای ناشناخته در داده می باشد که یکی از کاربردهای آن پیش بینی است. برای پیش بینی با استفاده از شبکه عصبی مصنوعی ، دو برنامه وجود دارد یکی مرحله آموزش می باشد و دیگری پیش بینی بر اساس داده آموزش داده شده است . همچنین از الگوریتم پس انتشار برای آموزش و از شبکه عصبی مصنوعی feedforward برای پیش بینی تقاضا استفاده می شود. در این پژوهش در نظر است مروری بر شبکه های عصبی مصنوعی، کارکرد و روش اجرای آن داشته باشیم.

واژگان کلیدی: شبکه عصبی، پیش بینی، الگوریتم پس انتشار

مقدمه

شبکه‌های عصبی مصنوعی سیستم‌های دینامیکی هوشمند مدل-آزاد مبتنی بر داده‌های تجربی هستند که نیاز به برقراری هیچ پذیره‌ای ندارند و با پردازش روی داده‌های تجربی، دانش یا قانون نهفته در ورای داده‌ها را به ساختار شبکه منتقل می‌کنند. شبکه‌های عصبی مصنوعی بر اساس محاسبات روی داده‌های عددی یا مثال‌ها قوانین کلی را فرا می‌گیرند و در مدل‌سازی ساختار نرو-سیناپتیکی مغز بشر می‌کوشند (کوین ، ۱۹۹۹).

در واقع در محاسبات نرونی تلاش می‌شود تا از روش کار مغز تقلید شود که یکی از چندین روش آموزش ماشینی است. شبکه‌های عصبی پتانسیل ایجاد ویژگی‌هایی دارند که انسان برای حل مسائل سخت مانند شبیه‌سازی منطقی، تکنیک‌های آنالیتیکال سیستم‌های خبره و تکنولوژی‌های نرم افزاری بکار می‌برد. در واقع سیستم شبکه عصبی مصنوعی الهام گرفته شده از مغز و سیستم شبکه عصبی انسان می‌باشد و مانند مغز انسان از تعداد زیادی نورون تشکیل شده است. این شبکه‌ها مانند مغز انسان دارای قابلیت یادگیری، حفظ کردن و ایجاد ارتباط مابین داده‌ها را می‌باشند. پیاده‌سازی ویژگی‌های شگفت‌انگیز مغز در یک سیستم مصنوعی (سیستم دینامیکی ساخته دست بشر) همیشه وسوسه‌انگیز و مطلوب بوده است. محققینی که طی سال‌ها در این زمینه فعالیت کرده‌اند بسیارند؛ لیکن نتیجه این تلاش‌ها، صرف نظر از یافته‌های ارزشمند، باور هر چه بیشتر این اصل بوده است که مغز بشر دست نیافتنی است. این شبکه‌ها در علوم بسیاری از جمله فیزیک، مهندسی برق، مهندسی محیط زیست، مهندسی شیمی، علوم کامپیوتر، رباتیک و ... کاربرد دارند (استوارت و همکاران، ۲۰۰۳).

۱- تشابهات و انتظارات

با تمام اغراق‌ها در مورد شبکه‌های عصبی مصنوعی، این شبکه‌ها اصلا سعی در حفظ پیچیدگی مغز ندارند. اکنون موضوع یاد شده را تحت دو عنوان تشابهات و انتظارات مورد بررسی قرار می‌دهیم.

۲- جداول، شکل ها و نمودارها

۱-۲- تشابهات

۱) بلوک‌های ساختاری در هر دو شبکهٔ مصنوعی و بیولوژیکی، دستگاه‌های محاسباتی خیلی ساده‌ای هستند و علاوه بر این، نرون‌های مصنوعی از سادگی بیشتری برخوردار می‌باشند.

۲) ارتباط های بین نرون ها، عملکرد شبکه را تعیین می کند.

اگر چه نرون‌های بیولوژیکی از نرون‌های مصنوعی که توسط مدارهای الکتریکی ساخته می‌شوند، بسیار کندتر هستند (یک میلیون بار)، اما عملکرد مغز، خیلی سریعتر از عملکرد یک کامپیوتر معمولی است. علت این پدیده، بیشتر بخاطر ساختار کاملا موازی نرون‌ها می‌باشد و این یعنی این که «همه نرون‌ها معمولا بطور همزمان کار می‌کنند و پاسخ می‌دهند». شبکه‌های عصبی مصنوعی هم دارای ساختار موازی هستند. اگر چه بیشتر شبکه‌های عصبی مصنوعی هم اکنون توسط کامپیوترهای سری پیاده سازی می‌شوند، اما ساختار موازی شبکه‌های عصبی، این امکان را فراهم می‌آورد که بطور سخت افزاری، توسط پردازشگرهای موازی، سیستم‌های نوری و تکنولوژی VLSI[1] پیاده سازی شوند (محمودی و همکاران، ۲۰۱۰).

[1] Very Large Scaled Integration

۲-۲- انتظارات

شبکه‌های عصبی مصنوعی با وجود این که با سیستم عصبی طبیعی قابل مقایسه نیستند ویژگی‌هایی دارند که آنها را در بعضی از کاربردها مانند تفکیک الگو، رباتیک، کنترل، و به طور کلی در هر جا که نیاز به یادگیری یک نگاشت خطی و یا غیر خطی باشد، ممتاز می‌نمایند. این ویژگی‌ها به شرح زیر هستند:

۲-۲-۱- قابلیت یادگیری

استخراج نتایج تحلیلی از یک نگاشت غیر خطی که با چند مثال مشخص شده، کار ساده‌ای نیست. زیرا نرون، یک دستگاه غیر خطی است و در نتیجه یک شبکه عصبی که از اجتماع این نرون‌ها تشکیل می‌شود نیز یک سیستم کاملا پیچیده و غیر خطی خواهد بود. به علاوه، خاصیت غیر خطی عناصر پردازش، در کل شبکه توزیع می‌گردد. پیاده سازی این نتایج با یک الگوریتم معمولی و بدون قابلیت یادگیری، نیاز به دقت و مراقبت زیادی دارد. در چنین حالتی سیستمی که بتواند خود این رابطه را استخراج کند بسیار سودمند به نظر می‌رسد. خصوصا افزودن مثال‌های احتمالی در آینده به یک سیستم با قابلیت یادگیری، به مراتب آسانتر از انجام آن در یک سیستم بدون چنین قابلیتی است، چرا که در سیستم اخیر، افزودن یک مثال جدید به منزله تعویض کلیه کارهای انجام شده قبلی است (اویلینو[۲] ۲۰۰۳).

قابلیت یادگیری یعنی توانایی تنظیم پارامترهای شبکه (وزن‌های سیناپتیکی) در مسیر زمان که محیط شبکه تغییر می‌کند و شبکه شرایط جدید را تجربه می‌کند، با این هدف که اگر شبکه برای یک وضعیت خاص آموزش دید و تغییر کوچکی در شرایط محیطی آن (وضعیت خاص) رخ داد، شبکه بتواند با آموزش مختصر برای شرایط جدید نیز کارآمد باشد. دیگر این که اطلاعات در شبکه‌های عصبی در سیناپس‌ها ذخیره می‌گردد و هر نرون در شبکه، به صورت بالقوه از کل فعالیت سایر نرون‌ها متأثر می‌شود. در نتیجه، اطلاعات از نوع مجزا از هم نبوده بلکه متأثر از کل شبکه می‌باشد (ژانگ[۳] ۲۰۰۲).

۲-۲-۲- پراکندگی اطلاعات "پردازش اطلاعات به صورت متن"

آنچه که شبکه فرا می‌گیرد (اطلاعات یا دانش)،[۴] در وزن‌های سیناپسی مستتر می‌باشد. رابطه یک به یک بین ورودی‌ها و وزن‌های سیناپتیکی وجود ندارد. می‌توان گفت که هر وزن سیناپسی مربوط به همه ورودی‌هاست ولی به هیچ یک از آنها بطور منفرد و مجزا مربوط نیست. به عبارت دیگر هر نرون در شبکه، از کل فعالیت سایر نرون‌ها متاثر می‌باشد. در نتیجه، اطلاعات به صورت متن[۵] توسط شبکه‌های عصبی پردازش می‌شوند. بر این اساس چنانچه بخشی از سلول‌های شبکه حذف شوند و یا عملکرد غلط داشته باشند باز هم احتمال رسیدن به پاسخ صحیح وجود دارد. اگر چه این احتمال برای تمام ورودی‌ها کاهش یافته برای هیچ یک از بین نرفته است(پرامانیک[۶] ۲۰۰۴).

۲-۲-۳- قابلیت تعمیم[۷]

پس از آنکه مثال‌های اولیه به شبکه آموزش داده شد، شبکه می‌تواند در مقابل یک ورودی آموزش داده نشده قرار گیرد و یک خروجی مناسب ارائه نماید. این خروجی بر اساس مکانیسم تعمیم، که همانا چیزی جز فرآیند درون‌یابی[۸] نیست به دست می‌آید. به عبارت روشنتر، شبکه، تابع را یاد می‌گیرد، الگوریتم را می‌آموزد و یا رابطه تحلیلی مناسبی را برای تعدادی نقاط در فضا به دست می‌آورد (پرامانیک، ۲۰۰۴).

---

2 . Avelino
3 . Zhang
4. Knowledge
5. Context
6 . Pramanik
7 Generalization
8. Interpolation

۲-۲-۴- پردازش موازی

هنگامی که شبکه عصبی در قالب سخت افزار پیاده می‌شود سلول‌هایی که در یک تراز قرار می‌گیرند می‌توانند بطور همزمان به ورودی‌های آن تراز پاسخ دهند. این ویژگی باعث افزایش سرعت پردازش می‌شود. در واقع در چنین سیستمی، وظیفه کلی پردازش بین پردازنده‌های کوچکتر مستقل از یکدیگر توزیع می‌گردد(استوارت و همکاران، ۲۰۰۳).

۲-۲-۵- مقاوم بودن[9]

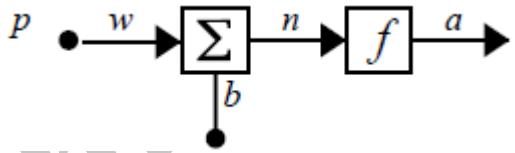
در یک شبکه عصبی هر سلول بطور مستقل عمل می‌کند و رفتار کلی شبکه، برآیند رفتارهای محلی سلول‌های متعدد است. این ویژگی باعث می‌شود تا خطاهای محلی از چشم خروجی نهایی دور بمانند. به عبارت دیگر، سلول‌ها در یک روند همکاری، خطاهای محلی یکدیگر را تصحیح می‌کنند. این خصوصیت باعث افزایش قابلیت مقاوم بودن (تحمل پذیری خطاها) در سیستم می‌گردد (استوارت و همکاران، ۲۰۰۳).

۲-۳- مدل نورون

نورون کوچکترین واحد پردازشگر اطلاعات است، که اساس عملکرد شبکه‌های عصبی را تشکیل می‌دهد (دیمیتری و جان،[10] ۲۰۱۳).

۲-۳-۱- مدل تک ورودی

شکل (۱) ساختار یک نورون تک ورودی را نشان می‌دهد. اسکالرهای p , a به ترتیب ورودی و خروجی می‌باشند.



شکل (۱): مدل نورون تک ورودی

میزان تأثیر p روی a به وسیله مقدار اسکالر w تعیین می‌شود. ورودی دیگر که مقدار ثابت است، در جمله بایاس b ضرب شده و سپس با wp جمع می‌شود، این حاصل جمع، ورودی خالص[11] n برای تابع محرک (یا تابع تبدیل) f خواهد بود. بدین ترتیب خروجی نرون با معادله زیر تعریف می‌شود (الکامل[12] و همکاران، ۲۰۰۵).

فرمول ۱
$$a = f(wp + b)$$

در مقایسه این مدل تک ورودی با یک نورون بیولوژیکی، w معادل شدت سیناپس، مجموعه جمع کننده و تابع محرک، معادل هسته سلول و a معادل سیگنال گذرنده از اکسون خواهد بود. نکته‌ای که باید به آن توجه شود اهمیت و تاثیر جمله بایاس b است. این جمله را می‌توان مانند وزنه w در نظر گرفت، با این تصور که میزان تاثیر ورودی ثابت را روی نرون منعکس می‌سازد. اهمیت جمله b به مرور توضیح داده خواهد شد (الکامل و همکاران، ۲۰۰۵).

[9] . Robustness
[10] . Dimitri and John
[11] . Net Input
[12] . Elkamel

باید توجه داشت که پارامترهای w ،b قابل تنظیم هستند و تابع محرک f نیز توسط طراح انتخاب می‌شود. بر اساس انتخاب f و نوع الگوریتم یادگیری، پارامترهای w،b ورودی و خروجی نرون با هدف خاصی مطابقت نماید(الکامل و همکاران، ۲۰۰۵).

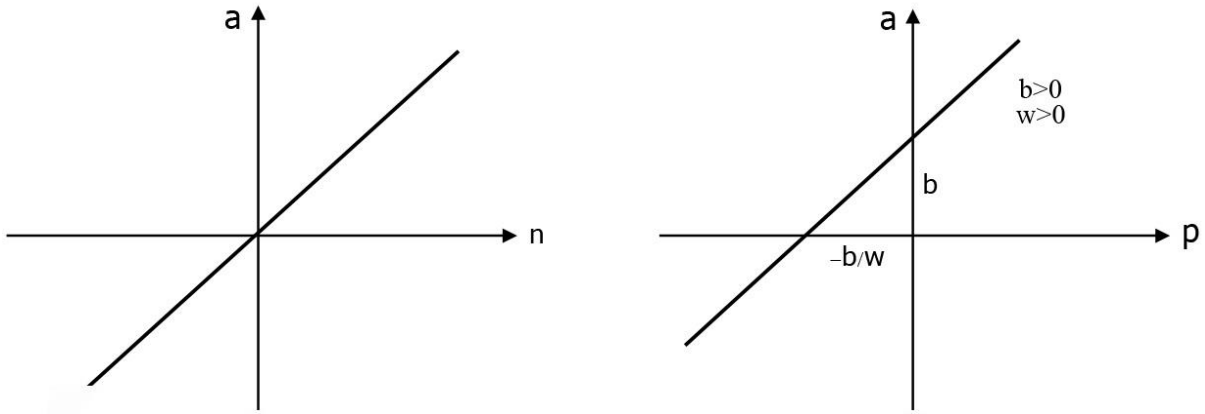
۲-۳-۲- توابع محرک

تابع محرک f می تواند خطی یا غیر خطی باشد. یک تابع محرک براساس نیاز خاص حل یک مسئله – مسئله‌ای که قرار است به وسیله شبکه عصبی حل شود- انتخاب می‌شود. در عمل تعداد محدودی از توابع محرک مورد استفاده قرار می‌گیرند، که در این جا چند مورد از مهم‌ترین آن ها اشاره می‌کنیم (لی[3] و همکاران، ۲۰۱۷).

۱-۲-۳-۲- تابع محرک خطی: خروجی این تابع برابر ورودی آن است.

$$a = f_1(n) = n$$ فرمول ۲

تابع خطی در شکل (۲) نمایش داده شده است. از نرون‌هایی با تابع تبدیل فوق، در شبکه‌های خاصی مانند آدالاین استفاده شده است. اهمیت جمله بایاس b را در شکل (۲) می‌بینیم. اگر پاسخ نرون، a، بر حسب ورودی p رسم شده باشد، جمله بایاس b موجب جابه جایی منحنی در فضای ورودی می‌گردد و به عبارتی موجب می‌گردد که نرون زیر فضایی از فضای ورودی بایاس گردد، که خود انتخاب کلمه بایاس را برای ترم b توجیه می‌کند (پرامانیک،[4] ۲۰۰۴).



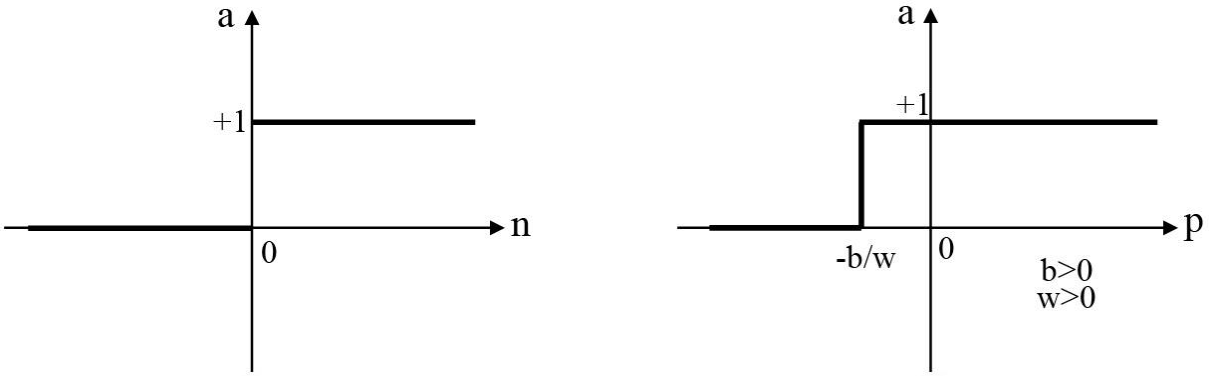
شکل (۲) تابع محرک خطی

۲-۳-۲-۲تابع محرک آستانه ای دو مقداره حدی

این تابع در شکل (۳) نشان داده شده است. همان گونه که مشاهده می‌شود، مقدار خروجی صفر یا است. اگر آرگومان n کوچکتر از $\frac{-b}{w}$ باشد، مقدار تابع ، صفر است و در غیر این صورت خروجی نرون برابر خواهد شد. عموما تابع محرک، دامنه خروجی نرون را محدود می‌سازد و به همین علت آن را تابع محدود ساز[5]نیز می‌نامند. خروجی نرون معمولا برای اینگونه توابع، در بازه متناهی [۱ , ۰] یا [۱, ۱-] قرار دارد، که در حالت اخیر تابع را تابع محرک آستانه‌ای دو مقداره حدی گویند(پرامانیک، ۲۰۰۴).

---

[1] .Li
[1] .Pramanik
[1] Hard Limitter or Squash
[3]
[4]
[5]



شکل (۳) تابع محرک آستانه‌ای دو مقداره حدی

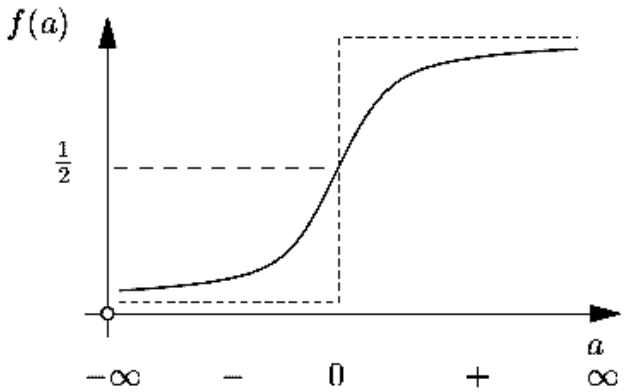
۳-۲-۳-۲تابع محرک سیگموئید

این تابع با فرمول کلی زیر بیان می‌شود:

(فرمول ۳)

$$a = f_s(n) = \frac{1}{1+e^{-cn}}, \quad c > 0$$

شکل این تابع به ازاء $C=1$ در تصویر (-) رسم شده است. مقدار C وسعت ناحیه خطی بودن تابع را تعیین می کند. مثلا اگر C خیلی بزرگ باشد، شکل منحنی به تابع محرک آستانه‌ای دو مقداره حدی نزدیکتر می‌شود. این تابع در شکل‌های عصبی مورد استفاده زیادی دارد که به عنوان مثال، می‌توان به شبکه‌های عصبی چند لایه با قانون یادگیری پس انتشار خطا، اشاره کرد.

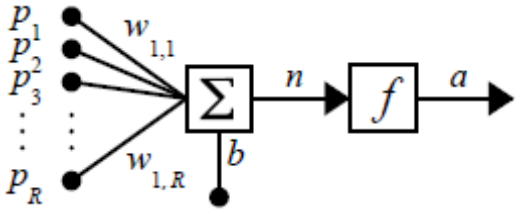


شکل (۴): تابع محرک سیگموئید

۳-۲-۳- مدل چند ورودی

عموما یک نرون بیش از یک ورودی دارد. شکل (۵) یک مدل نرون با R ورودی را ارائه می‌دهد. بردار ورودی با $\underline{P}$ نمایش داده می‌شود. اسکالرهای $p_i(i=1,2,...R)$ عناصر بردار $\underline{P}$ هستند. مجموعه سیناپس‌های $w_{1,i}$، عناصر ماتریس وزن W را تشکیل می دهند. در این حالت W یک بردار سطری با عناصر $w_{1,i}, j=1,...R$ است. هر عنصر از بردار ورودی $\underline{P}$ در عنصر متناظر از W

ضرب می‌شود. نرون، یک جمله بایاس b دارد که با حاصل ضرب ماتریس وزن W با بردار ورودی $\underline{P}$ جمع می‌شود(اسیجو[۶] و همکاران، ۲۰۰۹).



شکل (۵): مدل چند ورودی یک نرون

ورودی خالص n، مطابق فرمول زیر محاسبه می شود:

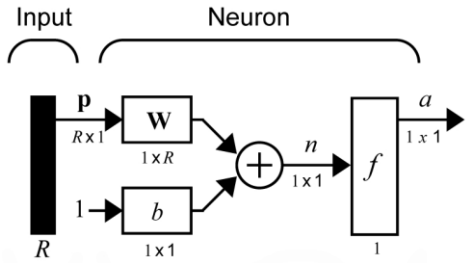
$$n = \sum_{i=1}^{R} p_i w_{1,i} + b = Wp + b$$ (فرمول ۴)

که در آن:

$$\underline{p} = [p_1, p_2, ..., p_R]^T, W = [w_{1,1}, ..., w_{1,R}]$$ (فرمول ۵)

شبکه‌های عصبی را می‌توان عموما به شکل ماتریسی مدل نمود. در انتخاب اندیس‌ها قرارداد خاصی به کار رفته است، که باید بیشتر درباره آن توضیح دهیم. دومین اندیس در نمایش ماتریس‌ها، مبدأ سیگنال ورودی نرون را نشان می‌دهد و اندیس اول به شماره خود نرون اشاره خواهد داشت. در این حالت چون فقط یک نرون داریم، ماتریس وزن W، به یک بردار سطری تبدیل می‌شود. مثلا عنصر $W_{1,2}$ ، شدت سیناپس دومین عنصر ورودی به نرون اول را نمایندگی می‌کند. در حالت کلی هر سطر ماتریس W متناظر با یک نرون است.

یک مدل خلاصه شده نرون چند ورودی را می‌توان به شکل ۶ نشان داد:



شکل (۶): فرم ساده شده نرون با R ورودی

همان گونه که در شکل فوق مشاهده می‌شود، بردار ورودی، $\underline{P}$ با یک ستون عمودی در سمت چپ نمایش داده می‌شود ابعاد $\underline{P}$ در زیر متغیر $\underline{P}$ با $R \times 1$ مشخص شده است. این نماد نشان دهنده آن است که بردار ورودی $\underline{P}$ دارای R عنصر است. بردار $\underline{P}$ در یک ماتریس W با R ستون، ضرب می شود. مقدار ثابت 1 به عنوان یک ورودی به نرون اعمال شده و در جمله اسکالر بایاس b ضرب

[1] . Assidjo

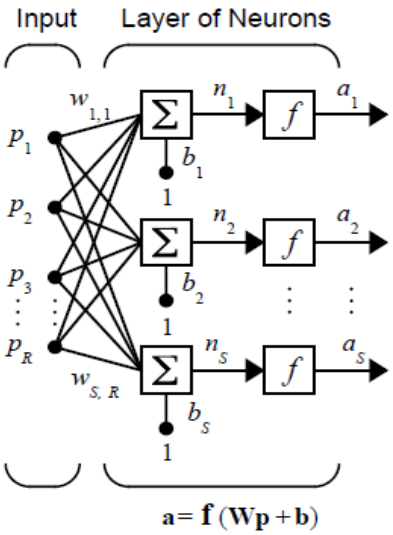
می شود. ورودی خالص n، به تابع تبدیل f اعمال شده و خروجی مربوطه به وجود می‌آید. a عملا نشان دهندهٔ خروجی شبکه تک نرونی است و در این حالت یک اسکار با ابعاد 1×1 است که در شکل مشخص شده است (اولینو[1]۲۰۰۳).

۲-۴- ساختار شبکه های عصبی

باید توجه داشت که معمولا نرونی با ورودی‌های زیاد نیز، به تنهایی برای حل مسائل فنی‌مهندسی کفایت نمی‌کند. مثلا برای مدلسازی نگاشت‌هایی که دارای دو خروجی هستند احتیاج به دو نرون داریم، که بطور موازی عمل کنند. در این حالت یک لایه خواهیم داشت که از اجتماع چند نرون تشکیل شده است.

۱-۴-۲- شبکه تک لایه

یک شبکه تک لایه با S نرون در شکل نشان داده شده است. شکل مذکور را می‌توان به فرم فشرده تصویر(۷) هم نمایش داد. ورودی شبکه با بردار $\underline{P}$ و خروجی آن با بردار $\underline{a}$ نشان داده شده است. باید توجه داشت که هر یک از ورودی‌ها به همه نرون‌ها متصل شده است. ماتریس W نیز در این حالت دارای S سطر و R ستون می‌باشد. همان گونه که در شکل مشاهده می‌شود، لایه‌ها شامل ماتریس وزن، جمع کننده‌ها، بردار بایاس $\underline{b}$ (دارای S عنصر) و تابع تبدیل f هستند(ژانگ، ۲۰۰۲).



شکل (۷): شبکه تک لایه با S نرون

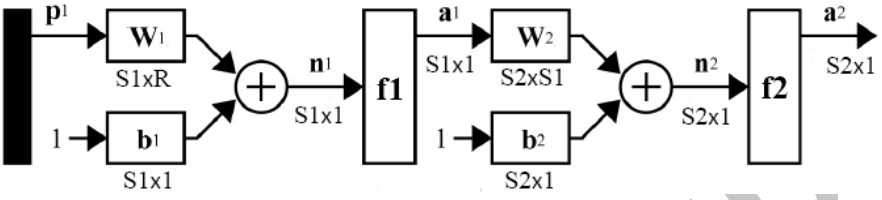
۲-۴-۲- شبکه های چند لایه

در شبکه‌های تک لایه، بردار ورودی ($\underline{p}$) توسط نرون‌های لایه (عناصر محاسباتی)، طبق رابطه $\underline{a} = \underline{f}(W\underline{p} + \underline{b})$ به بردار خروجی مرتبط می‌شود. این شبکه شکل ساده‌ای از شبکه‌های پیشخور[2]می‌باشد.

در این بخش، ایده شبکه‌های پیشخور را به شبکه‌های چند لایه تعمیم می‌دهیم. هر لایه ماتریس وزن W، بردار بایاس $\underline{b}$ ، بردار ورودی خالص $\underline{n}$ و بردار خروجی مختص خود را دارد. جهت تمایز متغیرهای فوق و این که کدام متغیر به کدام لایه تعلق دارد، نیاز داریم که علامت دیگری را هم به متغیرهای فوق تخصیص دهیم. از این رو از بالانویس برای مشخص نمودن لایه استفاده می‌کنیم، بنابراین ماتریس وزن را برای لایه اول با W^1 مشخص می‌نماییم. با به خاطر سپردن این نماد، یک شبکه پیشخور دو لایه

[1] . Avelino
[2] . Feedforward

را می توان به شکل (۸) ترسیم نمود. همان گونه که از شکل پیداست، تعداد R ورودی و تعداد S^1 نرون در لایه اول و تعداد S^2 نرون در لایه دوم در شبکه چند لایه پیشخور موجود است (ژانگ، ۲۰۰۲).



شکل (۸): شبکه پیش‌خور دو لایه

۳-۴-۲- لایه خروجی

لایه‌ای که خروجی آن، نهایی شبکه باشد، به لایه خروجی موسوم است. لایه‌های دیگر به لایه‌های میانی موسومند. شبکه موجود در شکل (-) فرم ساده شده یک شبکه عصبی پیشخور را با دو لایه میانی نمایش می‌دهد. در این جا لایه خروجی با ماتریس وزن W^3 ، بردار بایاس $\underline{b}^3$ و تابع محرک f^3 مشخص می شود. لایه میانی اول با ماتریس وزن W^1، بردار بایاس $\underline{b}^2$ و تابع محرک f^1 و لایه میانی دوم با ماتریس وزن W^2، بردار بایاس $\underline{b}^2$ و تابع محرک f^2 مشخص می‌شوند (ژانگ، ۲۰۰۲).

شبکه‌های عصبی چند لایه نسبت به شبکه‌های عصبی تک لایه دارای توانایی بیشتری هستند. همچنین شبکه‌های عصبی پیشخور دو لایه با توابع زیگموئید در لایه اول، قادرند هر تابعی را با دقت دلخواه تقریب بزنند، در حالی که شبکه‌های عصبی تک لایه از چنین توانائی برخوردار نیستند.

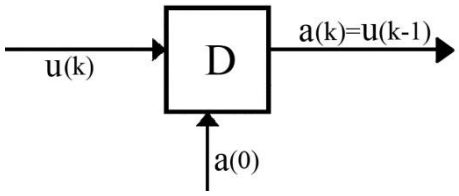
به نظر می‌رسد که تعداد درجات آزادی (مثلا تعداد ورودی‌ها، خروجی‌ها و نرون‌های هر لایه) برای طراحی یک شبکه چند لایه پیشخور زیاد باشد. ولی باید توجه داشت که تعداد ورودی‌های شبکه (R) و تعداد خروجی‌های شبکه (S)، بر اساس مسئله خاص که شبکه قرار است حل کند، به دست می‌آیند. به عبارت دیگر این دو پارامتر جزء پارامترهای آزاد طراح نیستند، بلکه انتخاب آنها بستگی دارد به مسئله‌ای که در حال بررسی است. مثلا اگر می‌خواهیم سیستمی را که دارای چهار ورودی و سه خروجی است، توسط شبکه‌های عصبی تقریب بزنیم، باید یک شبکه عصبی انتخاب کنیم که دارای چهار ورودی ($R=4$) و سه خروجی ($S=3$) باشد (به عبارتی تعداد نرون‌های لایه خروجی برابر باشد). علاوه بر این، ویژگی‌های خروجی مطلوب شبکه نیز، در انتخاب نوع تابع محرک لایه خروجی تاثیر دارد. مثلا اگر خروجی شبکه تنها یا - باشد، بهتر است تابع محرک لایه خروجی، تابع متقارن آستانه‌ای دو مقداره حدی باشد. بنابراین می‌توان گفت که ساختار شبکه‌های عصبی تک لایه پیشخور را، کاملا از روی ویژگی‌های مسئله مورد بررسی، می‌توان مشخص نمود(ژانگ، ۲۰۰۲).

در مورد شبکه عصبی چند لایه، مطالب فوق صادق نیست. یعنی از روی ویژگی‌های مسئله مورد بررسی، نمی‌توان اطلاعاتی در مورد تعداد نرون‌های لایه میانی کسب نمود. در واقع تعداد اندکی از مسائل وجود دارد که برای آنها با توجه به مسئله خاص، امکان پیش‌بینی تعداد نرون‌های لایه میانی وجود داشته باشد.

۵-۲- شبکه‌های پسخور یا برگشتی[۱۹]

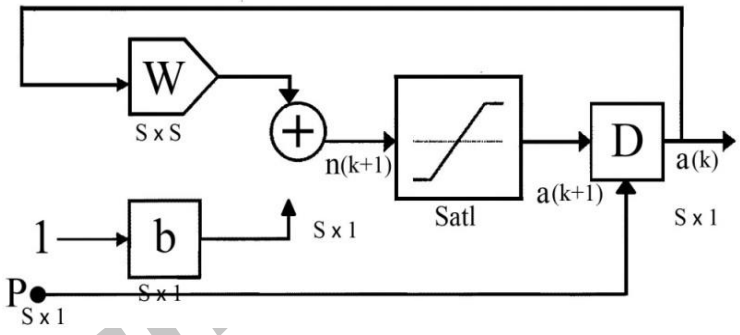
تفاوت شبکه‌های پسخور با شبکه‌های پیشخور این است که در شبکه‌های پسخور، حداقل یک سیگنال برگشتی از یک نرون به همان نرون یا نرون‌های همان لایه و یا لایه قبل وجود دارد. جهت نمایش این گونه شبکه‌ها، بلوک ساده زیر تعریف می‌کنیم:

1 Recurrent or Feedback



شکل (۹): بلوک تاخیر زمانی

شکل فوق معرف تأخیر زمانی یک مرحله‌ای است. به عبارت واضح‌تر خروجی بلوک (a) در لحظه زمانی k، برابر ورودی بلوک در یک واحد زمانی عقبتر (یعنی مقدار a در لحظه $k-1$) است. واضح است که برای بدست آوردن رابطه ورودی و خروجی بلوک فوق، باید مقدار اولیه خروجی در لحظه نخست (صفر) معلوم باشد. این مقدار اولیه با $a(0)$ در شکل فوق نشان داده شده است. اکنون می‌توانیم شبکه‌های پسخور را معرفی کنیم. یک نوع متداول از شبکه پسخور گسسته در حوزه زمان در شکل (۱۰) ترسیم شده است (ایوانا[2] و همکاران، ۲۰۰۷).



شکل (۱۰): شبکه پسخور

برای این شکل خاص، بردار ورودی p، نشان دهنده شرایط اولیه شبکه است (به عبارتی $a(0)=p$). رفتار شبکه به وسیله معادله زیر بیان می‌شود:

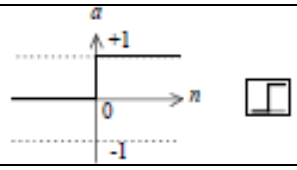
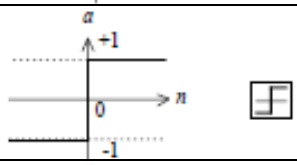
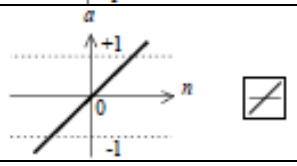
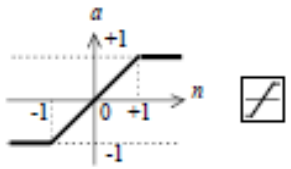
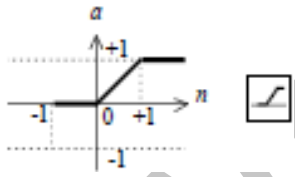
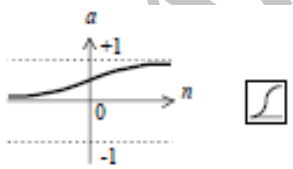
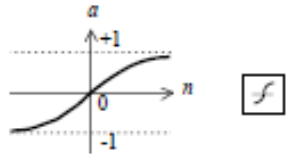
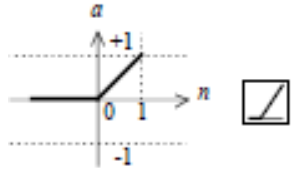
$$\begin{cases} a(k) = Satl(W\,a(k-1) + b), k = 1,2 \\ a(0) = p \end{cases}$$

(فرمول ۶)

شبکه‌های پسخور از توانایی بالقوه بیشتری نسبت به شبکه‌های پیشخور برخوردارند و بهتر می‌توانند رفتار مربوطه به ویژگی‌های زمانی سیستم‌ها را نشان دهند.

[2] . Ivana

جدول (۱): توابع محرک با علائم قراردادی (استوارت و همکاران،)

ردیف	نام	تعریف تابع	علائم قراردادی
۱	آستانه‌ای دو مقداره	$a=0 \quad , \quad n\langle 0$ $a=1 \quad , \quad n \geq 0$	
۲	آستانه‌ای دو مقداره متقارن	$a=-1 \quad , \quad n\langle 0$ $a=1 \quad , \quad n \geq 0$	
۳	خطی	$a=n$	
۴	آستانه‌ای خطی متقارن	$a=-1 \quad , \quad n\langle -1$ $a=n \quad , \quad -1 \leq n \leq 1$ $a=1 \quad , \quad n\rangle 1$	
۵	آستانه‌ای خطی	$a=0 \quad , \quad n\langle 0$ $a=n \quad , \quad 0 \leq n \leq 1$ $a=1 \quad , \quad n\rangle 1$	
۶	سیگموئیدی	$a=\frac{1}{1+e^{-n}}$	
۷	تانژانت هیپربولیکی	$a=\frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$	
۸	خطی مثبت	$a=0 \quad , \quad n\langle 0$ $a=n \quad , \quad n \geq 0$	

۶-۲- فرآیند یادگیری

سیستم‌های یادگیری سیستم‌هایی هستند که صرفاً با مشاهده عملکردشان، می‌توانند رفتارهایشان را جهت دستیابی به هدف و مقصدی خاص بهبود بخشند. اگر مقاصد و اهداف بطور کامل تعریف شده باشند آنگاه دیگر به فرآیند یادگیری احتیاجی نیست. زمانی که به پروسه یادگیری نیاز است که اطلاعات کامل در مورد اهداف موجود نباشد، جایی که می‌دانیم به علت عدم قطعیت در شرایط محیطی، سیستمی که دارای خواص و پارامترهای ثابت باشد نمی‌تواند بطور کامل عمل کند (اولینو، ۲۰۰۳).

رفتار سیستم‌های یادگیر[۲۱] توسط الگوریتم‌های بازگشتی بیان می‌شود. به همین خاطر به این الگوریتم‌ها قوانین یادگیری می‌گویند و عموماً تقریبی است از هدف خاص که مقصود پروسه یادگیری می‌باشد بهینه گردد و این کار تنها راه جبران نمودن نقصان اطلاعات اولیه می‌باشد.

به پروسه یادگیری نیاز است چون اطلاعات (ارتباط ورودی و خروجی) کاملاً مشخص نیستند. می دانیم که تجربه ها در مسیر زمان حاصل می شوند و به عبارت دیگر هیچ کس آینده خود را تجربه نکرده است. میزان یادگیری ما به درجه کامل بودن اطلاعات قبلی ما بستگی دارد. در حالت کلی دو نوع یادگیری موجود است: یادگیری با ناظر[۲۲] و یادگیری بدون ناظر.[۲۳]

در یادگیری با ناظر بر این است که در هر مرحله تکرار الگوریتم یادگیری، جواب مطلوب سیستم یادگیرنده از قبل، آماده است، و به عبارتی الگوریتم یادگیری به جواب واقعی و مطلوب دسترسی دارد. مثلاً فرض کنید سیستم یادگیر می خواهد نگارش $y = x^2$ را بیاموزد. اگر به سیستم مقدار / را بدهیم جواب مطلوب / را به دست می دهد. بطور کل جوابی را که سیستم یادگیری با وضعیت فعلی پارامترهایش می دهد به عنوان جواب واقعی در نظر می گیریم. در این جا الگوریتم یاد گیرنده که پارامترهای سیستم یاد گیرنده را تنظیم می کند، هم به جواب مطلوب / و هم به جواب واقعی دسترسی دارد. به عبارتی به خطای یادگیری که همان خطای بین مقدار مطلوب و مقدار واقعی می باشد دسترسی خواهد داشت(استوارت و همکاران، ۲۰۰۳).

در نگاه اول به نظر می‌رسد که یادگیری بدون ناظر بی فایده و غیر ممکن باشد، با طرح این سوال که "چگونه می توانیم یک سیستم یاد گیرنده را آموزش دهیم اگر ندانیم که این سیستم چه کاری قرار است انجام دهد؟" مثلاً در مسئله گروه بندی، هر گروه باید با یک ویژگی مشخص شود. به نظر نمی آید که یک سیستم یادگیر به تنهایی بتواند مشخصه‌هایی را که موجب گروه‌بندی می‌شوند بیابد و ویژگی‌های دیگر را نادیده انگارد. به عبارتی دیگر سیستم آموزش پذیر نمی‌تواند حدس بزند که کدام گروه و یا طبقه مورد نظر فردی است که مسئله الگوبندی را ارزیابی می‌کند. خواهیم دید که بیشتر این الگوریتم‌های بدون ناظر عمل خوشه بندی[۲۴] را انجام می‌دهند. آنها می‌آموزند که الگوهای ورودی را به تعداد متناهی از گروه ها تقسیم بندی کنند. باید توجه داشته باشیم که در این حالت فرد طراح یا معلم است که هدف و مقصد نهایی، یعنی چیزی را که باید به آن رسید مشخص می‌کند. و به عبارتی یادگیری بدون معلم مفهوم نادرستی است که عموماً به جای یادگیری بدون ناظر مورد استفاده قرار می‌گیرد. خلاصه در فرآیند یادگیری سه مورد زیر باید به ترتیب انجام شوند(محمودی و همکاران، ۲۰۱۰).

- سیستم یادگیرنده توسط محیط تحریک شود.
- قانون یادگیری با رجوع به نتیجه تحریک پارامتر های سیستم یادگیری را تغییر دهد.

۳) سیستم یادگیرنده به خاطر تغییراتی که در ساختار داخلی آن اتفاق افتاده است، پاسخ مناسبتری به محیط بدهد.

---

1 2 Learning Systems
2 2 Supervised Learning
3 2 Unsupervised Learning
4 2 Clustering

۱-۶-۲- معادله یادگیری در حالت کلی

دیدیم که یک نرون با یک بردار $W=[w_1,w_2,...w_R,b]^T$ و بردار ورودی $p=[p_1,...,p_R,1]^T$ متناظر است. هر نرون توانایی دارد که بردار وزن خود را بر اساس بردار ورودی و یک سیگنال دیگر موسوم به سیگنال معلم (در حالت کلی) یا سیگنال خطا (در یادگیری با ناظر) تغییر و بهبود دهد. در یادگیری بدون ناظر سیگنال معلم تغییر بردار حالت[۵] خود نرون است.

در حالت، کلی محیط یا منبع اطلاعاتی نرون را می‌توانیم به صورت زیر مشخص کنیم:

برای یادگیری با ناظر $\{(P,t,prob(p,t))\}$

برای یادگیری بدون ناظر $\{(P,prob(p,s))\}$

که در آن p بردار ورودی، t جواب مطلوب، prob تابع توضیع احتمال و s بردار حالت نرون است. بطور هندسی می‌توانیم شکل ساده (۱۱) را داشته باشیم.

قانون کلی یادگیری را برای یک نرون می‌توان به صورت زیر نوشت:

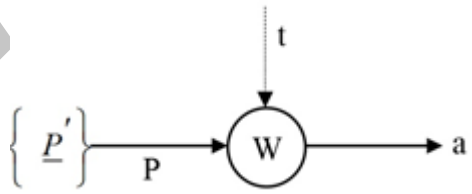
حالت پیوسته:

$$w=-\alpha w(t)+\eta.l.p(t) \quad \text{(فرمول ۷)}$$

حالت گسسته:

$$w(k+l)=(j-\alpha)w(k)+\eta.l.p(k) \quad \text{(فرمول ۸)}$$

که در این فرمول برادر وزن w متناسب با حاصل ضرب برادر ورودی p در سیگنال یادگیری l تغییر می کند و همزمان بطور خیلی کند کاهش می یابد. η,α نیز مقادیر ثابت مثبت کوچکتر از l هستند. در حالت کلی سیگنال یادگیری تابعی از t,w,p (برای حالت یادگیری با ناظر) است. انواع قوانین یادگیری که ساختار کلی آنها گفته شد از قبیل همینگ، رقابتی، پس انتشار خطا و غیره در فصول مختلف بطور کامل بحث خواهند شد. در ادامه خواهیم دید که قانون یادگیری پرسپترون همان قانون فوق برای حالت گسسته است، جایی که $\alpha=o$ و سیگنال یادگیری برابر با سیگنال خطا $l=t-a$ است.



شکل (۱۱): مدل یادگیری نرون

۲-۶-۲- یادگیری شبکه

تمامی توانایی‌های حیرت انگیز شبکه‌های عصبی بیولوژیکی انسان، بطور مادرزادی ساخته نشده‌اند. آن بخش از توانایی‌های انسان از قبیل شناسایی، محاسبه، تفکرات تجریدی، مهارت‌های فردی و هنری در طول زمان در انسان تکوین می‌یابند و از بدو تولد در انسان نهاده نشده‌اند. یک روش یادگیری در طول زمان، شبکه عصبی را طوری تنظیم می‌کند که بر اساس اطلاعات جدید و مشاهده عملکرد فعلی، رفتار خود را بهبود بخشد، مثلاً چهره جدیدی را در ذهن بسپارد بدون این که چهره‌های قبلی را از یاد ببرد (استوارت و همکاران، ۲۰۰۳).

[2] State

در دهه چهل و بخصوص دهه پنجاه از قرن بیستم محققان تلاش می‌کردند که ساختار شبکه‌های عصبی را با قانون یادگیری کامل کنند. قانون یادگیری یعنی این که پارامترهای شبکه بر اساس ارائه الگوها تنظیم شود. سرانجام در سال فرانک روزنبلات استاد روانشناسی دانشگاه کرنل آمریکا از طریق آنالیز ریاضی و شبیه‌سازی کامپیوتری، نشان داد که نوعی از شبکه‌های عصبی موسوم به پرسپترون را می توان برای حل مسئله طبقه بندی الگوها آموزش داد. نرونها در این شبکه شبیه نرون‌هایی بودند که مک کلوث و پیتز ارائه داده بودند(اولینو و همکاران، ۲۰۰۳).

کار مهم روزنبلات در معرفی این شبکه نبود، بلکه ابداع قانون یادگیری برای شبکه پرسپترون بود. او قانون یادگیری ساده و اتوماتیکی را مطرح نمود که قادر بود از هر شرط اولیه که شبکه انتخاب می‌کرد به جواب مسئله (در صورت وجود) همگرا شود. در اواسط دهه شصت روزنبلات علاقه‌مند به توصیف بیولوژیکی پروسه فراگیری (تاثیر متقابل نرون شناسی و محاسبات مصنوعی نرونی) شد. بطوری که در سال او تلاش می کرد تا توضیح دهد چگونه رشته ای از تجربه‌های حسی پس از دوره‌ای از زندگی می‌توانند به یاد آورده شوند. متاسفانه شبکه پرسپترون روزنبلات از حل بسیاری از مسائل و طبقه بندی الگوهایی که در فضای ورود بطور خطی از هم جداناپذیر ناتوان بود. در دهه هشتاد قرن بیستم این محدودیت‌ها توسط الگوریتم یادگیری جدیدی مرسوم به پس انتشار خطا، برای شبکه‌های عصبی پرسپترون چند لایه مرتفع شد(محمودی و همکاران، ۲۰۱۰).

در این جا برای یادگیری یک نرون، محیط نرون ثابت در نظر گرفته شده است. اما زمانی که یک شبکه عصبی رفتار خود را به شکل اشتراکی، نه بطور مستقل آن گونه که برای یک تک نرون دیدیم تغییر و بهبود می‌بخشد، هر نرون بردار وزن‌های متناظر خود را مطابق با قانون یادگیری خاص خودش تغییر می‌دهد. محیط منبع اطلاعاتی هر نرون در این حالت دیگر ثابت نیست، بلکه با تغییر وزن‌های نرون‌های دیگر تغییر می‌کند، چون محیط منبع اطلاعاتی یک نرون قائم به ذات نیست، بلکه وابسته به رفتارهای نرون‌های دیگر در شبکه است. بنابراین معادلات زیر را می‌توانیم برای نرون‌های یک شبکه بنویسیم:

حالت پیوسته:

$$w_{ij} = \alpha w_{ij}(t) + \Delta w_{ij}(t)$$

(فرمول ۹)

حالت گسسته:

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

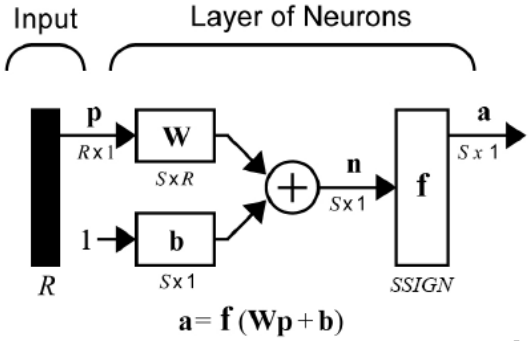
(فرمول ۱۰)

که در آن w_{ij} وزن ترم اصلاحی می‌باشد. سیناپس است که j امین عنصر بردار ورودی را به نرون i ام متصل می‌کند و Δw_{ij} ترم اصلاحی می‌باشد.

۳-۶-۲- قانون یادگیری پرسپترون (SLPR[2])

در این نوع یادگیری محرک، پاسخ درست، ورودی و خروجی مطلوب، الگو و این که الگو به چه طبقه‌ای تعلق دارد در اختیار می‌باشند. در این نوع یادگیری‌ها عملاً خطای یادگیری در اختیار است و از این رو در عمل مناسب است که در هر مرحله، از خطای یادگیری در اختیار است و از این رو در عمل مناسب است که در هر مرحله، از خطای یادگیری جهت تنظیم پارامترهای شبکه طوری استفاده شود که در صورت اعمال دوباره همان ورودی خطای یادگیری کمتر باشد. قانون یادگیری SLPR فقط برای شبکه عصبی تک لایه ای، متشکل از نرون‌های با تابع تبدیل از نوع آستانه حدی دو مقداره SSIGN یا (Hard Limiter) به وجود آمده‌اند. شکل زیر یک شبکه پرسپترون تک لایه را نشان می‌دهد .

[2] Single Layer Perceptron Rule



شکل (۱۲): پرسپترون تک لایه

برای این که تصویر روشنتری از شبکه پرسپترون تک لایه فوق داشته باشیم، حالتی را در نظر می‌گیریم که در آن S=1 و R=2 است. این نرون با معادله زیر تبیین می‌شود:

$$W_i p_i + w_2 p_2 + b = 0$$ **(فرمول ۱۱)**

۲-۶-۴- الگوریتم پس انتشار[۲۷] خطا

این الگوریتم که در سال ۱۹۸۶ توسط روملهارت و مک‌کلیلاند پیشنهاد گردید، در شبکه‌های عصبی پیشرو (Feed-Forward) مورد استفاده قرار می‌گیرد. پیشرو بودن به این معناست که نرون‌های مصنوعی در لایه‌های متوالی قرار گرفته‌اند و خروجی (سیگنال) خود را رو به جلو می‌فرستند. واژه پس‌انتشار نیز به معنای این است که خطاها به سمت عقب در شبکه تغذیه می‌شوند تا وزن‌ها را اصلاح کنند و پس از آن، مجدداً ورودی مسیر پیشرو خود تا خروجی را تکرار کند. روش پس‌انتشار خطا از روش‌های با ناظر است؛ به این مفهوم که نمونه‌های ورودی برچسب خورده‌اند و خروجی مورد انتظار هر یک از آنها از پیش دانسته است. لذا خروجی شبکه با این خروجی‌های ایده‌آل مقایسه شده و خطای شبکه محاسبه می‌گردد. در این الگوریتم ابتدا فرض بر این است که وزن‌های شبکه به طور تصادفی انتخاب شده‌اند. در هر گام خروجی شبکه محاسبه شده و بر حسب میزان اختلاف آن با خروجی مطلوب، وزن‌ها تصحیح می‌گردند تا در نهایت این خطا مینیمم شود. در الگوریتم پس‌انتشار خطا، تابع تحریک هر عصب به صورت جمع وزن‌دار ورودی‌های مربوط به آن عصب درنظر گرفته می‌شود. بدین ترتیب با فرض این که w وزن‌های متناظر بین لایه ورودی و لایه بعد باشد می‌توان نوشت (لوکاس[۲۸] و همکاران، ۲۰۱۵).

$$n_j(\bar{p},\bar{w}) = \sum_{i=0}^{n} p_i w_{j,i}$$ **(فرمول ۱۲)**

به وضوح می‌توان دید که خروجی تابع تحریک عصب فقط به ورودی و وزن‌های متناظر بستگی دارد. با فرض اینکه تابع خروجی، سیگموئید باشد می‌توان خروجی نرون j ام را به صورت زیر نوشت:

$$a_j(\bar{p},\bar{w}) = sgm(n_j(\bar{p},\bar{w})) = \frac{1}{1+e^{n_j(\bar{p},\bar{w})}}$$ **(فرمول ۱۳)**

تابع سیگموئید به ازای اعداد منفی بزرگ، بسیار نزدیک به صفر است و برای اعداد مثبت بزرگ، مقداری بسیار نزدیک به ۱ دارد و در این بین به طور هموار تغییر می کند به نحوی که در $x = 0$ دقیقاً از حد واسط بازه [۰, ۱] یعنی / عبور می کند. همچنین با دقت در فرمول (۱۳) در می‌یابیم که خروجی فقط به مقدار تابع تحریک بستگی دارد که به نوبه خود به ورودی و وزن‌ها مرتبط

---

2 Back Propagation (BP)
2 Lukas

می‌شود. لذا برای تغییر خروجی باید وزن‌ها تغییر کنند. آنچنان که پیش از این نیز بیان شد، هدف فرآیند آموزش، رسیدن به خروجی مطلوب (یا نزدیک به مطلوب) است. بدین ترتیب ابتدا باید تابع خطای هر نرون را تعریف کنیم. این خطا از اختلاف خروجی واقعی شبکه و خروجی مورد انتظار به صورت زیر بدست می‌آید:

$$E_j(\bar{p},\bar{w},t_j)=(a_j(\bar{p},\bar{w})-t_j)^2$$ **(فرمول ۱۴)**

انتخاب مربع تفاضل بین خروجی واقعی (a_j) و خروجی مطلوب (t_j) از چندین جنبه قابل بحث است؛ اولاً با استفاده از توان دوم، مقدار خطا همواره مثبت خواهد بود؛ ثانیاً اگر اختلاف بین خروجی واقعی و مطلوب زیاد باشد، توان دوم منجر به بزرگتر شدن این عدد می‌شود و بالعکس اگر اختلاف بین خروجی واقعی و مطلوب کم باشد، توان دوم منجر به کوچکتر شدن آن می‌گردد. بر این اساس می‌توان خطای کلی شبکه را به فرم مجموع خطای تک تک عصب‌های لایه خروجی نوشت. لذا داریم (لو و ناکائی[29]، ۲۰۰۱).

$$E(\bar{p},\bar{w},\bar{t})=\sum_j E_j(\bar{p},\bar{w},t_j)=\sum_j(a_j(\bar{p},\bar{w})-t_j)^2$$ **(فرمول ۱۵)**

حال بایستی به بررسی ارتباط خطا با ورودی‌ها، وزن‌ها و خروجی‌ها بپردازیم. برای این کار روش‌های متفاوتی وجود دارد که برخی از مهمترین آنها عبارتند از:

۱) روش گرادیان شیب (Gradient Descent)
۲) روش نیوتون
۳) روش اندازه حرکت (Momentum)
۴) روش انتروپی متقابل (Cross Entropy)
۵) روش Levenberg-Marquarlt

جهت توقف تکرار الگوریتم BP از دو شاخص زیر بطور همزمان می‌توان استفاده نمود:

الف) میانگین مربعات خطا در هر سیکل یا Epoch (جمع مربعات خطا برای تمامی الگوهای یادگیری) کمتر از مقدار از پیش تعیین شده‌ای باشد و یا اینکه فرم تغییرات در پارامترهای شبکه پس از هر سیکل خیلی کوچک باشد. باید توجه داشت که هر سیکل برابر با تعداد تکرار، به اندازه تعداد نمونه‌های یادگیری می‌باشد. مثلا اگر تا داده‌های نمونه یادگیری موجود است، سیکل برابر با مرحله تکرار می‌گردد.

ب) ترم گرادیان خطا خیلی کوچک باشد؛ ترم گرادیان خطا از یک مقدار از پیش تعیین شده‌ای کوچک‌تر گردد (محمودی و همکاران، ۲۰۱۰).

۲-۷- شبکه‌های عصبی پرسپترون[30]

شبکه‌های عصبی پرسپترون، به ویژه پرسپترون چند لایه، در زمره کاربردی‌ترین شبکه‌های عصبی می‌باشند. این شبکه‌ها قادرند با انتخاب مناسب تعداد لایه‌ها و سلول‌های عصبی، که اغلب زیاد هم نیستند، یک نگاشت غیر خطی را با دقت دلخواه انجام دهند. این همان چیزی است که در بسیاری از مسائل فنی مهندسی به عنوان راه حل اصلی مطرح می‌باشد (ژانگ، ۲۰۱۰).

۲-۸- شبکه‌های پرسپترون چند لایه[31]

در شکل (۱۳) شبکه پرسپترون سه لایه ارائه گردیده است. که مدل مورد نظر در تحقیق حاضر می باشد. همان گونه که در شکل (۱۳) ملاحظه می‌شود هر نورون در هر لایه، به تمامی نورون‌های لایه قبل متصل می‌باشد. به چنین شبکه هایی، شبکه های کاملاً

2 . Lou and Nakai
3 Perceptron
3 Multy Layer Perceptron (MLP)
9
0
1

مرتبط گویند. شبکه مذکور، عملاً از به هم پیوستن سه شبکه پرسپترون تک لایه ایجاد شده است. یکی لایه خروجی و دو قسمت دیگر لایه‌های میانی نامیده می‌شوند. خروجی‌های لایه اول، بردار ورودی لایه دوم را تشکیل می‌دهند، و به همین ترتیب بردار خروجی لایه دوم، ورودی‌های لایه سوم را می‌سازند، و خروجی‌های لایه سوم، پاسخ واقعی شبکه را تشکیل می‌دهند. به عبارتی روشنتر، روند جریان سیگنالی در شبکه، در یک مسیر پیشخور صورت می‌گیرد (از چپ به راست از لایه‌ای به لایه دیگر) (لوکاس و همکاران، ۲۰۱۵).

همان گونه که قبلا گفته شد، هر لایه می‌تواند از تعدادی نرون‌های مختلف با توابع تبدیل متفاوت برخوردار باشد؛ یعنی مدل‌های نرون‌ها در لایه‌ها می‌توانند متفاوت در نظر گرفته شوند. اندیس‌های فوقانی مبین شماره لایه و اندیس‌های تحتانی مبدأ و مقصد اتصال سیناپسی را مشخص می‌کنند، بنابراین ماتریس وزن برای لایه دوم با w^2 و وزنه اتصالی بین نرون i ام از لایه اول و ورودی j ام از بردار ورودی با $w^1_{i,j}$ نمایش داده می‌شود. جهت بیان ساختار یک شبکه چند لایه، از نمایش عبارتی کوتاه زیر استفاده می‌کنیم:

$$(R-S^1-S^2-S^3)$$ **(فرمول ۱۶)**

جایی که R تعداد ورودی‌ها و S^i تعداد نرون‌ها در لایه i ام می‌باشند.

در شبکه MLP عموماً دو نوع سیگنال استفاده می‌شوند که بهتر است از هم تمیز داده شوند؛ یک نوع سیگنال‌هایی هستند که در مسیر رفت حرکت می کنند (از سمت چپ به راست شبکه) و دسته دیگر سیگنال‌هایی هستند که در مسیر برگشت حرکت می‌کنند (از سمت راست به چپ). به دسته اول سیگنال‌های تابعی و به دسته دوم سیگنال‌های خطا گویند (اژار و همکاران، ۲۰۰۹).

دلیل این نامگذاری این است که سیگنال‌های دسته نخست، بر اساس تابعی از ورودی‌های هر نرون و پارامترهای شبکه متناظرش با آن محاسبه می‌شوند و سیگنال‌های دسته دوم به خاطر منشعب شدن از سیگنال خطا و توزیع برگشت از لایه خروجی به لایه‌های دیگر شبکه به سیگنال‌های خطا موسومند، و خلاصه اینکه سیگنال‌های تابعی در مسیر رفت در شبکه‌ای از لایه‌ای به لایه دیگر توزیع می‌شود و سیگنال‌های خطا در مسیر برگشت در شبکه منتشر می‌گردند (اسیجو، ۲۰۰۹).

هر نرون در شبکه MLP دو محاسبه انجام می‌دهد. درمحاسبه اول سیگنال تابعی و در محاسبه دوم تخمین لحظه‌ای از گرادیان منحنی خطا را نسبت به پارامترهایی که ورودی نرون را به خود نرون متصل می‌کند، در اختیار قرار می‌دهد. این گرادیان‌ها جهت پخش سیگنال‌های خطا در شبکه مورد نیاز می‌باشند.

نتیجه گیری

با عنایت به اینکه شبکه های عصبی از دو ویژگی اساسی یادگیری یا نگاشت پذیری بر اساس ارائه داده های تجربی (قدرت و توانایی تعمیم پذیری) و ساختار پذیری موازی برخوردار می باشند، این شبکه ها برای سیستم های پیچیده که مدلسازی این سیستم ها یا میسر نیست و یا به سختی انجام می شود بسیار مناسب می باشند. شبکه های عصبی مصنوعی به علت استفاده از فرمول های ثابت شده با حداقل خطا، صحت و اعتبار بیشتری دارند. با توجه به کاربرد های روز افزون شبکه های عصبی تلاش های بسیاری در راستای رفع نواقص این شبکه ها انجام شده است . یکی از دلایلی که باعث محدود شدن کاربرد شبکه های عصبی می شود مرحله آموزش می باشد . برای آموزش شبکه های عصبی همواره نیاز به گروه بزرگی از اطلاعات ورودی می باشد . در عین حال تنظیم پارامتر های آموزش شبکه کاری بسیار دشوار بوده و نیاز به تجربه دارد . همچنین مشکلاتی مانند مینیمم های محلی و همگرایی به یک پاسخ نامناسب ، حفظ کردن اطلاعات ورودی به جای یادگیری آنها ، مهمتر از همه نیاز به زمان زیاد

برای مرحله آموزش از مشکلات شبکه های عصبی می باشد.

منابع

1- Assidjo, N., Malhotra, R., and Malhotra, D. (2009) A Hybrid Neural Network Approach for Batch Fermentation Simulation, Australian Journal of Basic and Applied Sciences, 3(4): pp. 3930-3936.
2- Avelino, C., Yong-Qi, C., and Xing-Hua., B. (2003) Neural networks for modelling of kinetic reaction data applicable to catalyst scale up and process control and optimisation in the frame of combinatorial catalysis, Applied Catalysis A: General 254: pp. 133–145.
3- Azhar, A., El-sayed, D., and Peter, T. (2009) Production of Polyhydroxybutyrate (PHB) Using Batch and Two-stage Batch Culture Strategies, Australian Journal of Basic and Applied Sciences 3(2): pp. 617-627.
4- Dimitri P. Bertsekas., and John N. Tsitsiklis. (2013). Neuro-Dynamic Programming, Athena Scientific, Ch. 1-3.
5- Elkamel, H., Desai, V. S., and Crook, J. N. (2005) Hybrid artificial neural network—First principle model formulation for the unsteady state simulation and analysis of a packed bed reactor for CO2 hydrogenation to methanol, Chemical Engineering Journal (115), pp. 113–120.
6- Ivana, C., Mantovaneli, L., and Hall, M. A. (2007). Hybrid neural network model for alcoholic fermentation processes with multiple stages, 2nd Mercosur Congress on Chemical Engineering.
7- K Pramanik, (2004) Use of Artificial Neural Networks for Prediction of Cell Mass and Ethanol Concentration in Batch Fermentation using Saccharomyces Cerevisiae Yeast, IE (I) Journal, (85).
8- Kevin, Gurney. (1999) an Introduction to Neural Networks, UCL Press, Ch. 1-4, 6, 8.
9- Li, Pin Tan., and Kuan, Yew Wang. (2017) A Neural Network Approach for Predicting Manufacturing Performance using Knowledge Management Metrics, Cybernetics and Systems Journal, 48(4).
10- Lukas, F., Zuzana, S., Maria, D., Beata, H. and Tatian, P. (2015) Application of Neural Network Models in Modelling Economic Time Series with Non-Constant Volatility, Business Economics and Management 2015 Conference, BEM2015, Procedia Economics and Finance, (34), pp. 600 – 607.
11- Mahmoudi, M., Najafpour, Gh., Sharifzadeh, M., and Eisazadeh, H. (2010) Kinetic model for polyhydroxybutyrate (PHB) production by Hydrogenophaga pseudoflava and verification of growth conditions, African Journal of Biotechnology, 9(21): pp. 3151-3157.
12- Stuart, J. R., and Peter, N. (2003) Artificial Intelligence: A Modern Approach, Pearson Education, Inc., Ch. 7.
13- Zhang, W.J. (2002) An artificial neural network approach to mechanism kinematic chain isomorphism identification, Mechanism and Machine Theory (37), pp. 549–551.