گزارش پروژه داده کاوی – پیاده سازی MLP

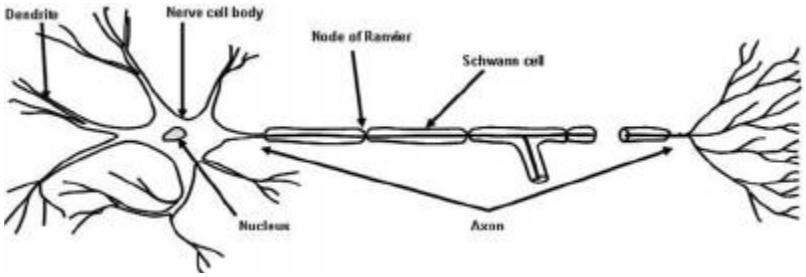
امیرحسین ابراهیمی 9813666013

شبکه‌های عصبی مصنوعی

نگاه مدرن به شبکه‌های عصبی در دهه‌ی 1940 و با شروع به کار وارن مک کلاچ و والتر پیتز آغاز شد. فعالیت آن‌ها در این زمینه را می‌توان به‌عنوان مبدأ علم شبکه‌های عصبی مصنوعی در نظر گرفت. می‌توان توسعه شبکه‌های عصبی را حدود 50 سال پیش در نظر گرفت راه مک کلاچ و پیتز توسط دونالد هب ادامه یافت. او ساز و کاری برای یادگیری در نرون‌های عصبی پیشنهاد کرد. اولین کاربرد عملی شبکه‌های عصبی در اواخر دهه‌ی 1950 شکل گرفت. در این سال‌ها شبکه‌های پرسپترون و قواعد یادگیری آن توسط فرانک روزنبلات و همکارانش ابداع شد. در همان زمان برنارد ویدرو و تد هوف یک الگوریتم یادگیری جدید را مطرح ساختند و از آن برای آموزش یک شبکه‌ی عصبی خطی انطباقی استفاده نمودند. علاقه به شبکه‌های عصبی را اواخر دهه‌ی 60 به خاطر کمبود ایده‌های جدید و نبود کامپیوترهای قدرتمند دچار وقفه شده بود. در طول دهه‌ی 1980 هردوی این موانع از پیش‌رو برداشته شد و تحقیقات در زمینه‌ی شبکه‌های عصبی جان تازهـای گرفت. اولین مفهوم استفاده از مکانیسم‌های آماری برای تشریح عملکرد کلاس خاصی از شبکه‌های بازگشتی بود که می‌توانستند به‌عنوان حافظه‌ی انجمنی مورد استفاده قرار گیرند، توسط جان‌هاپفیلد مطرح شد. این پیشرفت‌های جدید جان تازه‌ای به علم شبکه‌های عصبی بخشید. در 10 سال اخیر بیش از هزاران مقاله در این زمینه نوشته شده است و شبکه‌های عصبی دارای کاربرد بسیار زیادی شده‌اند و هر روزه زمزمه‌ی تازه‌ای در مورد یک تئوری جدید و با یک کاربرد جدید به گوش می‌رسد.

نرون طبیعی

مغز آدمی شامل بیش از ده بیلیون نرون می‌باشد که هرکدام از آن‌ها به‌طور متوسط به چندین هزار نرون دیگر متصل می‌باشد. این اتصالات تحت عنوان سیناپس شناخته می‌شوند. مغز انسان شامل حدود 60 تریلیون از این پیوندها می‌باشد. نرون‌ها دو نوع می‌باشند. نرون‌های داخلی مغز که در فاصله‌های حدود 100 میکرون به یکدیگر متصل‌اند و نرون‌های خارجی که قسمت‌های مختلف مغز را به یکدیگر و مغز را به ماهیچه‌ها و اعضای حسی را به مغز متصل می‌کنند. نرون‌ها در واقع المان‌های پردازش و بسیار ساده می‌باشند. هر نرون شامل یک سوما که بدنه‌ی نرون می‌باشد، یک آکسون و چند دندریت می‌باشد. شکل زیر مشخصات اصلی یک نرون بیولوژیک را نشان می‌دهد.



مشخصات اصلی یک نرون بیولوژیک

نرون‌ها دریافت‌هایی را از سایر نرون‌ها و دندریت‌ها دارند. وقتی‌که مقدار سیگنال ورودی بیشتر از یک حد آستانه‌ی خاص گردید، نرون به اصطلاح فعال می‌شود. در واقع یک واکنش شیمیایی حادث شده و یک پالس الکتریکی که پتانسیل فعالیت نامیده می‌شود به آکسون (خروجی نرون) فرستاده می‌شود. از آنجا و از طریق سیناپس‌هایی که متصل به نرون هستند، به دندریت‌های سایر نرون‌ها منتقل می‌گردد. در واقع این تماس به‌صورت اتصال مستقیم نیست بلکه از طریق ماده‌ای شیمیایی موقتی صورت می‌گیرد. سیناپس پس از آن که پتانسیل آن از طریق پتانسیل‌های فعالیت دریافتی از طریق آکسون به‌اندازهی کافی افزایش یافته، از خود ماده‌ی شیمیایی به نام منتقل‌کننده‌ی عصبی ترشح می‌کند. برای این ترشح ممکن است به دریافت بیش از یک پتانسیل فعالیت نیاز باشد. منتقل‌کننده‌ی عصبی ترشح شده در شکاف بین آکسون و دندریت پخش می‌شود و باعث می‌گردد که دروازه‌های موجود در دندریت‌ها فعال شده و باز شوند و بدین‌صورت یون‌های شارژ شده وارد دندریت شوند. این جریان یون است که باعث می‌شود پتانسیل دندریت افزایش یافته و باعث یک پالس ولتاژ در دندریت شود که پس از آن منتقل شده و وارد بدن نرون دیگر می‌شود. هر دندریت ممکن است تحت تأثیر تعداد زیادی سیناپس باشد و بدین‌صورت اتصالات داخلی زیادی را ممکن می‌سازد. در اتصالات سیناپسی تعداد دروازه‌های بازشده بستگی به مقدار منتقل‌کنندهی عصبی آزاد شده دارد. شبکه‌های عصبی مصنوعی فعلی هرگز به پیچیدگی مغز انسان نیستند اما به‌هرحال دو شباهت اساسی بین شبکه‌های عصبی زنده و مصنوعی وجود دارد. شباهت اول در این است که ساختار هردو از یک ابزار محاسباتی ساده با به هم‌پیوستگی بسیار بالا تشکیل شده‌اند. شباهت دوم این است که در هر دو مورد، اتصالات بین نرون‌ها، تعیین کنندهی تابع شبکه می‌باشند.

ساختار شبکه‌های عصبی مصنوعی

شبکه‌های عصبی مصنوعی به‌عنوان تعمیم مدل‌های ریاضی می‌باشد که بر اساس مفروضاتی بنا شده است :

1- پردازش اطلاعات در عناصری به نام سلول‌های عصبی (نرون) رخ می‌دهد.
2- سیگنال‌های بین نرون‌ها بر روی لینک‌های اتصال منتقل می‌شوند.
3- هر لینک اتصال، همراه با یک وزنی است، که در یک شبکه عصبی معمولی، در سیگنال منتقل شده ضرب می‌شود.
4- هر نورون، یک تابع فعال‌سازی (معمولاً غیرخطی)، را به ورودی خالص آن، (مجموع سیگنال‌های ورودی وزندار)، برای تعیین سیگنال خروجی، اعمال می‌کند.

در حالت کلی در شبکه‌های عصبی، سه لایه‌ی نرونی وجود دارد:

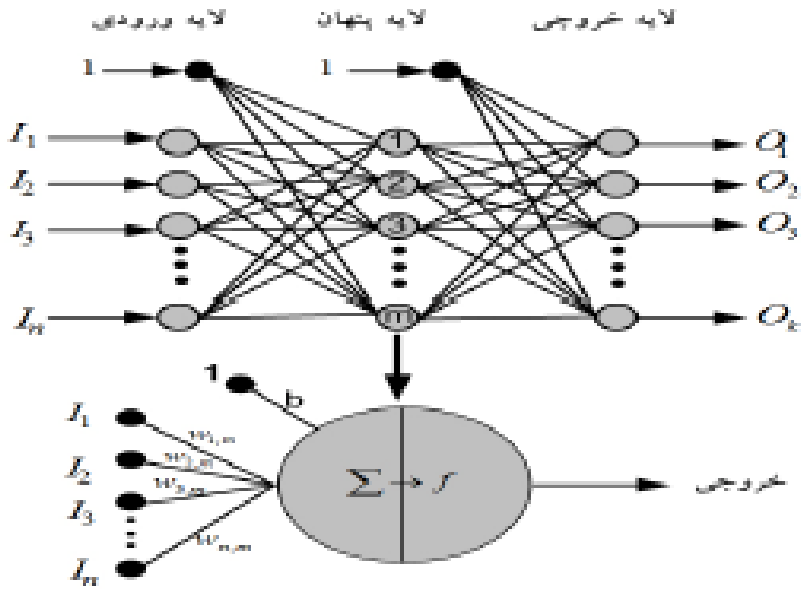
الف) لایه‌ی ورودی: دریافت اطلاعات خامی که به شبکه تغذیه شده است.

ب) لایه‌های پنهان: عملکرد این لایه به‌وسیله‌ی ورودی‌ها و وزن ارتباط بین آن‌ها و لایه‌های پنهان تعیین می‌شود. وزن‌های بین واحدهای ورودی و پنهان تعیین می‌کند که چه وقت یک واحد پنهان باید فعال شود.

پ) لایه خروجی: عملکرد واحد خروجی بسته به، فعالیت واحد پنهان و وزن ارتباط بین واحدپنهان و خروجی می‌باشد.

شبکه‌های عصبی از شمار بسیار زیادی عناصر پردازش کننده به‌هم پیوسته یا نرون تشکیل شده‌اند که برای حل یک مسئله هماهنگ با هم عمل می‌کنند. ارتباط بین این نرون‌ها عملکرد شبکه را تعیین می‌کند. در این شبکه‌ها به کمک دانش برنامه‌نویسی، ساختار داده‌ای طراحی می‌شود که می‌تواند همانند نرون عمل کند، سپس با ایجاد شبکه‌ای بین گره‌ها و اعمال یک الگوریتم آموزشی به آن، شبکه را آموزش می‌دهند. شبکه‌های عصبی را می‌توان در مسائلی که روابط دقیقی بین ورودی‌ها و خروجی‌های آن‌ها وجود ندارد و دارای رفتار پیچیده‌ای هستند استفاده نمود. شبکه عصبی با تحلیل داده‌ها، روابط پیچیده‌ی بین پارامترها را کشف کرده و جوابی با دقت قابل قبول ارائه می‌دهد. شبکه‌های عصبی مصنوعی همانند انسان با استفاده از مثال‌ها آموزش می‌بینند.

یک شبکه عصبی، ممکن است دارای بیش از یک لایه پنهان باشد. هر یک از لایه ها ی شبکه عصبی، دارای تعدادی نرون هستند. نرون ها ی لایه ها ی مختلف با هم در ارتباط بوده و عملیات پردازش داده ای در آن ها صورت می‌گیرد . شکل زیر ساختار یک شبکه عصبی مصنوعی و نحوه عملکرد یک نرون را نشان می‌دهد.



ساختار شماتیک یک شبکه عصبی سه لایه و نحوه پردازش اطلاعات در یک نورون

در یک شبکه عصبی مصنوعی، نرون mام اطلاعات ورودی خود را از طریـق گـره هـای ورودی I_i دریافت می‌کند. هر یک از این گره ها ی ورودی، قبل از اینکه وارد هسته اصلی نرون شود، وزن دار می شود. یعنی مقدار هر ورودی در $W_{i,m}$ ضرب می شود. سپس ایــن مقــادیر در بخش اول پردازشگر با هم جمع شده و مجموع کل ورودی به نرون تعیین می شــود . در برخــی موارد، یک مقدار ثابت در هر نرون بنام وزن اریب به مقدار کل ورودی افزوده می شود. مقــدار این ورودی یک و وزن آن b است. با در نظر گرفتن ایــن وزن بایــاس، مقــدار کــل ورودی بــه نرون از رابطه زیر محاسبه می شود.

$$U_m = \sum_{i=1}^{n} I_i W_{i,m} + b_m$$

در مرحله بعد، یک تابع موسوم به تابع محرک که معمولا تابعی غیرخطـــی اســـت، روی این مقدار حاصل جمع عمل کرده و مقدار خروجی نرون از رابطه زیر تعیین می گردد.

$$O_m = f(\sum_{i=1}^{n} I_i W_{i,m} + b_m)$$

هدف از به کار بردن این تابع، محدود کردن خروجی در یک بانــد مشــخّص اســت . بــه عبارت دیگر با وجود این تابع، خروجی یک نرون در مقابل ورودی ها ی بسیار کوچک یا بسیار بزرگ از حدود معینی تجاوز نمی کند.

انواع توابع محرک

متداول ترین توابع محرک شبکه ه ای عصبی مصنوعی، توابع سیگموئید، تانژانت هیپربولیک و خطی هســـتند. رابطه ریاضی هر یک از این توابع در جدول زیر آمده است :

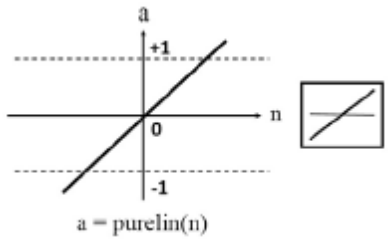
جدول 1-3. متداول ترین توابع محرک شبکه‌های عصبی

تابع محرک	محدوده تاثیر	رابطه تابع
خطی	بدون محدودیت	$f(x)=\beta x+\alpha$
سیگموئید	$0<f(x)<1$	$f(x)=\frac{1}{1+e^{-x}}$
تانژانت هیپربولیک	$-1<f(x)<1$	$f(x)=\frac{1-e^{-x}}{1+e^{-x}}$

در میان این توابع، تابع سیگموئید متداول ترین تابع مورد استفاده در شــبکه هــا ی عصــبی می باشد. دامنه این تابع از صفر تا یک است. بنابراین با استفاده از این تابع، خروجی ها ی شبکه در دامنه صفر تا یک قرار می گیرند. برای این منظور لازم است که ورودی ها ی شــبکه نیــز بــین صفر و یک قرار گرفته، یا به اصطلاح عمل نرمال سازی روی داده ها ی ورودی صورت گیـــــرد. در غیر این صورت، مقدار کل ورودی نرون بزرگ شده و در نتیجه مقدار خروجی آن بــا تــأثیر تابع محرک، همواره نزدیک یک خواهــد بــود . در ایــن حالــت تــابع خروجــی اشــباع شــده و همگرایی شبکه کند می شود. بنابراین فرآیند یادگیری بی نتیجه خواهد بود.

تابع محرک خطی

این تابع محرک در شکل زیر نشان داده شده است. نرون‌هایی که از این تابع محرک استفاده می‌کنند، برای تقریب خطی به کار می‌روند. این تابع همان مقدار ورودی را به‌عنوان خروجی بر می‌گرداند.

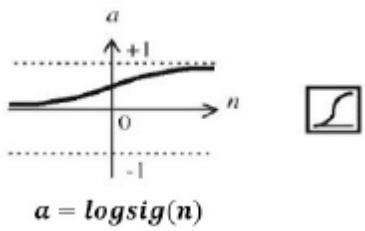


تابع محرک خطی

تابع محرک Log-sigmoid

از این تابع در شبکه‌های پس انتشار استفاده می‌شود. این تابع محرک مقادیر ورودی را در محدودهی ∞ تا ∞- دریافت کرده و بر مبنای رابطه‌ی یک مقدار خروجی بین صفر و یک تولید می‌کند. این تابع در شکل زیر نشان داده شده است.

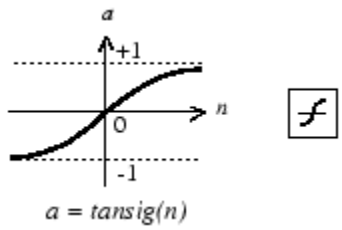
$$a = \frac{1}{1+e^{-x}}$$



شکل تابع محرک Log sigmoid

تابع محرک تانژانت هیپربولیک

از این تابع در شبکه‌های پس انتشار استفاده می‌شود. این تابع محرک مقادیر ورودی را در محدودهی ∞ تا ∞- دریافت کرده و بر مبنای رابطه‌ی یک مقدار خروجی بین منفی یک و یک تولید می‌کند. این تابع در شکل زیر نشان داده شده است.



$$a = \frac{1-e^{-x}}{1+e^{-x}}$$

تعیین تعداد لایه پنهان و گره های آن ها

لایه مخفی و تعداد گره های آن نقش بسیار موثری در شبکه های عصبی ایفا می کنند. در واقع این گره های لایه مخفی هستند که به شبکه اجازه می دهند خصوصیات الگوی نهفته در ورای داده ها را کشف نماید (ژانگ و همکاران، 1998). با این حال انتخاب تعداد لایه های مخفی و تعداد گره های این لایه، از قانونمندی خاصی پیروی نمی کند. خصوصا آنکه افزایش تعداد لایه های پنهان از یک سو باعث افزایش زمان محاسبات و آموزش شبکه شده و از سوی دیگر سبب ایجاد مشکل فرا انطباق مدل های پیش بینی می گردد.

استفاده از یک لایه مخفی برای تقریب زدن هر تابع غیرخطی کافی است (هورنیک و همکاران1، 1989) و بیشتر محققین از یک لایه مخفی در پیش بینی استفاده نموده اند. البته چون استفاده از یک لایه مخفی نیاز به تعداد نرون های زیادی دارد، سبب زیاد شدن زمان آزمایش و بدتر شدن قابلیت تعمیم شبکه می شود. ژانگ و همکاران (1998) بر این باورند که هر چند استفاده از دولایه مخفی در بعضی از موارد نتایج بهتری ایجاد می کند، با این وجود یک لایه مخفی در بیشتر مسائل پیش بینی کافی است.

در طراحی یک شبکه عصبی دو پارامتر بحرانی وجود دارد. این دو پارامتر به ترتیب اهمیت عبارتند از تعداد گره های لایه ورودی و تعداد گره های لایه مخفی (ژانگ و هو1998). نرون های لایه پنهان به شبکه اجازه می دهند الگوی غیر خطی درون داده ها را کشف نماید. بصورت تئوری محدودیتی در تعیین تعداد نرون های لایه مخفی وجود ندارد، ولی تعداد این نرون ها نیز از قانونمندی خاصی تبعیت نمی کند.

در شبکه های معمول که دارای یک لایه مخفی می باشند، محققین از روابط مختلفی برای تعیین تعداد نرون های لایه مخفی استفاده نموده اند. از جمله 2n+1 (لیپمن، 1987)، 2n (وانگ، 1991) و n (تانگ و فیشـــویک، 1993) که در آن n تعداد نورون‌های لایه ورودی می‌باشد. البته هیچکدام از روابط فوق برای تمام مسائل کارایی ندارند (ژانگ و همکاران 1998).

(کاســترا و بوید 1996) معتقدند هیچ فرمولی برای تعیین تعداد نرونهای لایه مخفی وجود ندارد، ولی استفاده از قاعده هرم دسی یکی از روش‌های پیش رو در این زمینه است)بر گرفته از آزادی، 1388(. این قاعده بصورت معادله زیر نشان داده شده است:

$$n_H = \sqrt{n_I . n_O}$$

که در آن n_H ، n_I و n_O بترتیب تعداد نرون های لایه پنهان، لایه ورودی و لایه خروجی می باشند. این محققین تعداد واقعی نرون های پنهان را بین نصف تا دو برابر تعداد بدست آمده می دانند. هر چند روابط مختلفی

برای تعیین تعداد نرون های لایه مخفی ارائه شده است، با این حال روش عمومی مورد استفاده در تعیین تعداد صحیح نرون های لایه پنهان، روش سعی و خطا می باشد (ژانگ و هو، 1998).

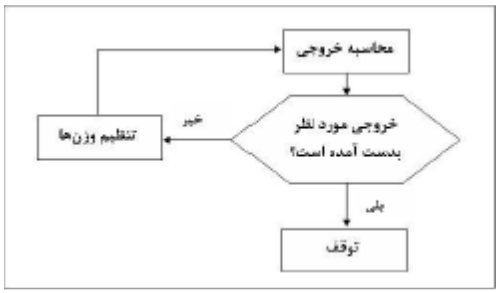
انتخاب تابع محرک در هر لایه

تابع فعال سازی بر اساس نوع مساله، بوسیله ی طراح شبکه انتخاب می گردد. برای نمونه زمانی که ارزش های خروجی مسئله تنها صفر و یک باشند، باید از تابع آستانه ای دو مقداره یا تابع خطی مثبت استفاده نمود. چرا که این توابع تنها مقادیر صفر و یک را نتیجه می دهند. البته قواعدی نیز برای انتخاب تابع فعال سازی وجود دارد. از جمله بهتر است در مسایل دسته بندی از توابع محرک زیگموئدی و در پیش بینی از تابع فعال سازی تانژانت هیپربولیکی استفاده شود. البته این قاعده همیشه برقرار نمی باشد، چرا که تابع فعال سازی نقش عمده ای در کارکرد شبکه دارد. در حالت عادی هر شبکه می تواند توابع فعال سازی متفاوتی برای هر گره در یک لایه و حتی لایه های متفاوت داشته باشد. با این حال، در اکثر مطالعات از یک نوع تابع فعال سازی برای گره های هر لایه استفاده شده است (ژانگ و هو، 1998).

راملهارت و همکاران (1995) در مطالعه خود مبانی تئوریکی برتری توابع خطی را در لایه خروجی شبکه های پیشخور مورد بررسی قرار دادند. این محققین نشان دادند که استفاده از توابع محرک خطی در لایه خروجی شبکه های پیشخور جهت پیش بینی، مناسب تر از سایر توابع هستند. در طراحی شبکه عصبی یکی از اصول کلیدی، اصل امساک است. بر اساس این اصل باید سعی نمود از ساده ترین مدل شبکه عصبی، که قادر است داده ها را آموزش دهد، استفاده نمود. بصورت تئوری اگر مساله ای توسط شبکه ای خاص قابل حل باشد، توسط شبکه های با اندازه بزرگتر نیز قابل حل است. ولی شبکه های کوچکتر که بتوانند نتایج مورد نظر را با دقت دلخواه تولید کنند، برتری دارند. اصل امساک بویژه در مورد تعداد لایه های پنهان و همچنین نرون های درون این لایه مصداق دارد (منهاج، 1377).

آموزش (یادگیری) در شبکه‌های عصبی مصنوعی

معمولاً در ادبیات شبکه های عصبی، به جای اصطلاح تخمین ضرایب، از واژه آموزش یا یادگیری استفاده می‌شود. در حقیقت سیستمهای یادگیر، سیستم هایی هستند که می توانند رفتارشان را جهت دستیابی به هدف و مقصدی خاص بر اساس مشاهده عملکردشان، بهبود بخشند. اگر مقاصد و اهداف بطور کامل تعریف شده باشند، دیگر نیازی به فرآیند یادگیری نیست. در واقع زمانی به فرآیند یادگیری نیاز می باشد که اطلاعات کامل در مورد اهداف موجود نباشد و یا جائیکه به علت عدم جود قطعیت، سیستمی با پارامترهای ثابت نمی تواند بطور کامل عمل نماید (منهاج،1377). برای یاد دادن انجام کاری خاص به شبکه می بایست شبکه را آموزش داد. برای این منظور از رویه ای که وزن ها و بایاس ها شبکه را اصلاح می نماید، استفاده می‌گردد. این رویه ، قاعده یادگیری یا الگوریتم آموزش نامیده می شود. بطور دقیق یک فرآیند یادگیری شامل سه قسمت است. در ابتدا خروجی های شبکه محاسبه می شوند. سپس خروجی ها با اهداف مورد نظر مقایسه شده و در انتها وزنها و بایاس ها تنظیم و در نهایت این فرآیند تا رسیدن به هدف خاص مسئله تکرار می شود. برای شروع فرآیند یادگیری وزنها بصورت تصادفی و یا توسط قواعدی خاص مقدار دهی می شوند. پس از مقدار دهی اولیه، خروجی شبکه محاسبه و میزان اختلاف آن با خروجی واقعی مقایسه می گردد. این اختلاف در طی فرآیند آموزش به کمک اصلاحاتی پی در پی، کاهش داده می شود. این فرآیند را می توان بصورت شکل زیر نمایش داد.



فرآیند آموزش شبکه عصبی مصنوعی

به طور خلاصه فرآیند یادگیری را می توان شامل سه مرحله دانست. در مرحله اول سیستم یادگیرند توسط محیط تحریک می شود. سپس قانون یادگیری با رجوع به نتیجه تحریک پارامترهای سیستم یادگیری تغییر می کند. در نهایت نیز سیستم یادگیری بدلیل تغییراتی که در ساختار داخلی آن اتفاق افتاده، پاسخ مناسبتری به محیط می دهد. اما این فرآیند یادگیری تا زمانی ادامه می یابد که یکی از شرایط زیر حاصل گردد:

1- رسیدن تعداد تکرارها به سقف معین (Max Epochs).

2- کوچکتر شدن مقدار تابع عملکرد شبکه از مقدار هدف معین (Goal).
3- گذشتن زمان آموزش از زمان معین (Max Time).
4- کوچکتر شدن گرادیان تابع خطا از میزان حداقل تعین شده (Min Grad).

انواع شبکه‌های عصبی از نظر شیوه یادگیری

شبکه ها ی عصبی مصنوعی بر اساس شیوه یادگیری به سه دسته تقسیم می شوند:

آموزش با وزن ها ی ثابت: در این روش وزن ها یک بار محاسبه شده و به هنگام نمی شوند. این شبکه‌ها در بهینه سازی و فشرده سازی اطلاعات و نیز بازیابی الگوها کاربرد دارند.

آموزش بدون نظارت: در این شبکه ها خروجی هدفی وجود ندارد تا بر اساس مقدار خطای حاصل از مقایسه خروجی شبکه با آن، وزن ها اصلاح شوند. بنابراین شبکه الگوهای آموزشی خود را از طریق ورودی ها یش دریافت کرده و به شکل دلخواه آن ها را در دسته ها ی مختلفی قرار می دهد. هنگامی که شبکه، یک سری ورودی را دریافت می کند، با کشف رابطه همبستگی و آماری بین ورودی ها ی مختلف، وزن ها را اصلاح کرده و پاسخی در خروجی ظاهر می شود که نشان دهنده طبقه ای است که آن ورودی بدان تعلق دارد.

آموزش با نظارت: در این گونه شبکه‌ها، به ازای هر دسته از الگوهای آموزشی ورودی، خروجی متناظر نیز به شبکه داده می شود و تغییر وزن ها تا موقعی که اختلاف بین مقدار خروجی شبکه و خروجی هدف حداقل شود، ادامه می یابد.

در محاسبه خطا به روش مجموع مربعات خطا، اگرورودی ها ی شبکه با x_i و خروجی های واقعی یا هدف متناظر با این ورودی ها با y_i و نیز مقادیر خروجی شبکه به ازای این ورودی ها با z_i نمایش داده شوند، برای تعداد N داده ی ورودی، خطای الگوی آموزش با تعداد تکرار الگوی آموزش(E) از رابطه زیر بدست می آید.

$$E = \sum_{j=1}^{p} \sum_{i=1}^{n} (z_i - y_i)^2$$

انواع شبکه های عصبی از نظر ساختار و پردازش اطلاعات

بر اساس جهت ورود اطلاعات به شبکه، ساختار و چگونگی پردازش شبکه، شبکه ها ی عصبی مصنوعی به انواع مختلفی تقسیم می شوند که مهم ترین آن ها عبارتند از:

شبکه های عصبی پیشرو (FFNN): ارتباط بین لایه‌های مختلف این شبکه‌ها یک طرفه و رو به جلو است. به عبارت دیگر در این نوع شبکه ها، روند پردازش از گره‌های ورودی شروع شده و لایه به لایه پیش می‌رود تا در گره های خروجی به پایان برسد.

شبکه ها ی بازگشتی(RNN): در این نوع شبکه ها مقدار خروجی هر نرون تابعی از نرون ها ی لایه قبلی و فعلی می باشد. یعنی بازخوردی از خروجی ها ی مرحله قبل به ورودی ها ی زمان فعلی شبکه افزوده می شود.

شبکه ها ی عصبی با توابع پایه شعاعی(RBF): این نوع شبکه ها از نوع پیشرو بوده و دارای سه لایه هستند. توابع محرک این نوع شبکه ها به صورت توابع پایه شعاعی همراه با مرکز و عرض خاصی می باشند. برخلاف شبکه ها ی پرسپترون چند لایه که خروجی هر نرون در لایه میانی به عنوان ورودی تابع محرک در نظر گرفته می شود، در شبکه ها ی پایه شعاعی فاصله بین هر الگو با بردار مرکز هر نرون در لایه میانی، به عنوان ورودی تابع محرک محاسبه می شود. بیشترین مقدار این توابع که معمولا توابع گوسی هستند، در مرکز نرون ها ی لایه میانی بوده و با دور شدن از مرکز، این مقدار کاهش می یابد. از آنجایی که شبکه ها ی تابع پایه شعاعی می توانند الگوها را روی دوایر هم مرکز توزیع نموده و بردارهای مختلف با فواصل یکسان را در یک دسته قرار دهند، در نتیجه این شبکه ها توانایی مناسبی در حل مسائل دسته بندی دارند.

شبکه ها ی عصبی پس انتشار زمانی(TBP): برخلاف شبکه های پیشرو که برای اتصال بین دو نرون در لایه‌های متوالی از یک وزن واحد استفاده می‌کند، در این نوع شبکه‌ها ، رشته ای از وزن‌های ورودی فعلی و قبلی نرون، به عنوان وزن فعلی نرون ها در نظر گرفته می شود. بنابراین میزان فعالیت یک نرون، تابع ورودی‌های قبلی آن و وزن‌های قبلی مربوط به آن ها است.

مراحل ساخت یک شبکه عصبی مصنوعی

برای ساخت یک مدل شبکه عصبی مصنوعی مراحل کلی زیر طی می شود.

مرحله اول – تعیین ساختار شبکه: منظور از تعیین ساختار یا معماری شبکه ی عصبی، تعیین نوع شبکه، تعیین بهینه ی تعداد لایه ها و گره‌های شبکه و تعیین توابع انتقالی و پایه به روش سعی و خطا برای دست یابی به یک جواب مناسب است.

مرحله دوم - آموزش شبکه: منظور از آموزش شبکه عصبی، اصلاح وزن ها ی ارتباطی بین لایه ها و نیز وزن ها ی بایاس شبکه برای نمونه ها ی متعدد است. آموزش شبکه زمانی کامل می شود که خطای مدل یا اختلاف بین مقادیر خروجی شبکه و مقادیر هدف حداقل شود. برای رسیدن به این هدف، داده ها ی آموزشی مربوط به الگوی مورد نظر چند بار به شبکه داده می شوند تا شبکه در هر بار با استفاده از آن ها وزن ها ی خود را اصلاح کند. با تکرار زیاد این کار، وزن ها به گونه ای اصلاح می شوند که شبکه قادر است در برابر داده های ورودی غیر آموزشی، خروجی قابل قبول ارائه دهد.

مرحله سوم - آزمون شبکه: پس از تکمیل فرآیند آموزش شبکه و تصحیح وزن‌های آن، شبکه با استفاده از یک دسته داده ها ی با خروجی معلوم مورد ارزیابی قرار می‌گیرد . در صورتی که آزمون شبکه موفقیت آمیز باشد، می توان از آن در تعیین مقادیر نامعلوم استفاده نمود.

الگوریتم‌های آموزش شبکه های عصبی مصنوعی

الگوریتم آموزش، ابزاری ریاضی است که توسط آن، شیوه و سرعت آموزش شبکه ها ی عصبی برای رسیدن به یک حالت ماندگار تعیین می شود. هدف از آموزش شبکه ها ی عصبی تنظیم ضرایب وزنی شبکه برای رسیدن به یک خطای حداقل است. آموزش شبکه با یک تابع خطا یا تابع انرژی که بر اساس مقادیر اولیه وزن ها و آستانه ها انتخاب شده است، آغاز می گردد و با حداقل شدن خطا، این وزن ها و آستانه ها به حالت ماندگار می رسند . تابع خطا یک تابع درجه دوم است که اختلاف بین مقادیر خروجی شبکه و مقادیر مطلوب را برای یک مجموعه آموزشی محاسبه می کند . انتخاب تابع خطا و الگوریتم بهینه سازی در آموزش شبکه ها ی عصبی برای رسیدن به جواب بهینه، بسیار مهم می باشد . روش ها ی مختلفی برای آموزش شبکه ها ی عصبی مصنوعی وجود دارند که متداول تر ین آن ها قانون دلتا ، الگوریتم بولتزمن، و الگوریتم پس انتشار خطا هستند. به سبب توانایی مناسب الگوریتم پس انتشار خطا در مدل سازی و کاربرد گسترده آن، در این بخش توضیح مختصری درباره آن ارائه می‌گردد.

الگوریتم آموزش پس انتشار خطا

این الگوریتم بر اساس قانون یادگیری اصلاح خطا عمل می کند. در این روش، آمـــــوزش شبکه با وزن ها ی تصادفی اولیه آغاز می شود. سپس با مقایسه مقادیر خروجی مـدل بـا مقـادیر مطلوب، خطای مدل محاسبه می شود و با برگشت به داخل شـبکه در جهـت عکـس یعنـی از خروجی به ورودی، مقادیر وزن ها اصلاح می شوند. این عمـل تـا حـداقل شـدن خطـا تکـرار می‌شود.

شبکه ها ی عصبی پس انتشار خطا دو دسته اند: شبکه ها ی پس انتشار پیش خور(FFBP) و شبکه های پس انتشار پیشرو (CFBP). در شبکه ها ی پس انتشار پیش خـــور محاســـبات از ورودی به خروجی انجام شده و سپس مقادیر خطا محاسبه شده به لایه ها ی قبل انتشار می یابد. بنابراین در این شبکه ها، مسیر حرکت اطلاعات یک طرفه بوده و از یک لایه به لایه بعدی است. اما در شبکه ها ی پس انتشار پیشرو، چون نرون ها ی هر لایه به همه لایه ها ی قبـل متصـل انـد، بنـابراین مسیر حرکت اطلاعات دو طرفه بوده و خروجی هر نـرون در یـک لحظـه خـاص، عـلاوه بـر سیگنال ها ی دریافتی از نرون ها ی لایه قبل، تابعی از همان نرون در یک لحظه قبل اسـت . آموزش پس انتشار خطا خود به چند روش مختلف صـورت مـی‌گیـرد کـه متـداول تـر ین ایـن روش ها عبارتند از:

الگوریتم پس انتشار لونبرگ- مارکوات(LM):

الگوریتمLM، فرم اصلاح شده الگوریتم کلاسیک نیوتن برای یافتن حـل بهینـه مسـائل حداقل سازی است. این الگوریتم از حداقل مجموع مربعـات خطـا (SSE) بـرای بهینـه سـازی غیرخطی وزن ها و بایاس‌های شبکه استفاده می‌کند. از آنجایی که این الگوریتم بـرای آمـوزش شبکه ها ی عصبی، از طریق توزیع محاسبات و فضـای مـورد نیـاز، بـه صـورت مـوازی عمـل می کند، بنابراین از سرعت و دقت مناسبی برخـوردار اسـت . بـه همـین سـبب، ایـن روش بـه صورت گسترده کاربرد دارد.

الگوریتم تنظیم بیزی(BR):

این الگوریتم به جای مجموع مربعات خطا از یک تابع هـدف شـامل مجمـوع مربعـات خطا و یک تابع جریمه استفاده می کند که این تابع جریمه به طور خودکار تنظیم می شود.

الگوریتم گرادیان مزدوج(SCG):

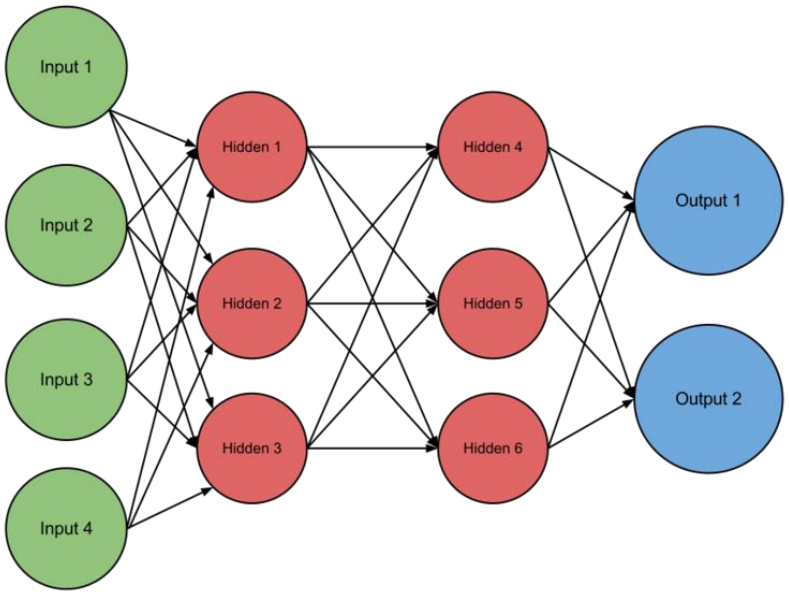
این الگوریتم پیچیده، ترکیبی از مـدل بکـار رفتـه در الگـوریتم لـونبرگ - مـارکوات و گرادیان مزدوج است و برای تعدیل وزن ها از مشتق دوم تابع تبدیل استفاده می کند.

طبقه بندی شبکه‌های عصبی

شبکه های عصبی بر اساس جهت جریان یافتن و پردازش اطلاعات به دو دسته اصلی تقسیم می‌شوند:

شبکه‌های پیش‌خور

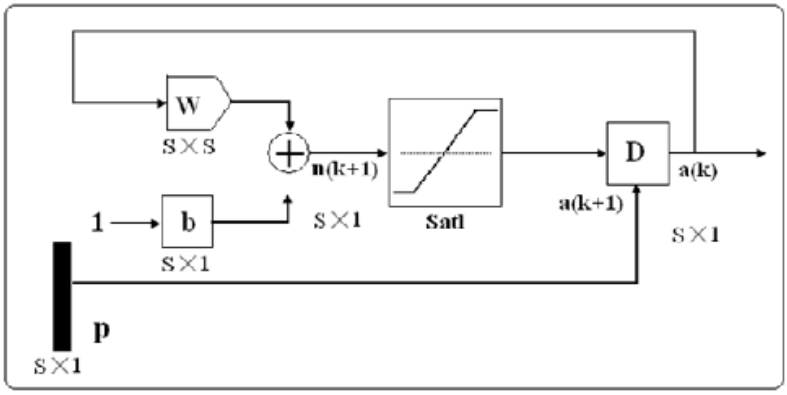
یک شـــبکه پیش‌خور با یک لایه ورودی آغاز و یک لایه خروجی ختم می‌شـــود. میان لایه ورودی و خروجی می‌توان چندین لایه پنهان وجود داشته باشد. لایه های ورودی و خروجی، تنها لایه‌های شبکه عصبی مصنوعی هســـتند که با محیط در ارتباط هســـتند. لایه ورودی مقادیر ورودی به شـــکه را از محیط یا همان کاربر دریافت کرده و لایه خروجی، پاسـخ سـیسـتم شـبکه عصـبی را به محیط یا کاربر برمی‌گرداند. دلیل نامگذاری لایه‌های پنهان، به دلیل عدم دسـترسـی مسـتقیم از محیط به این لایه‌ها و برعکس می‌باشـند. در یک شـبکه پیشـخور نرون‌های هر لایه، اطلاعات را تنها از نرون‌های لایه پیشین دریافت می‌کنند. پس از پردازش، شکل شکل پردازش شـــده را تنها به نورون‌های لایه بعدی، تحویل می‌دهند. به عبارت دیگر، در این نوع از شـــبکه‌ها اطلاعات تنها در یک جهت حرکت می‌کنند. البته لازم به توضـــیح اســـت که بکارگیری شـــبکه‌های پیش خور برای مدلســـازی فرآیندهای پویا متداول‌تر است.



مدل ساده شبکه پیش‌خور با دو لایه پنهان

شبکه‌های برگشتی

در شـــبکه‌های برگشـــتی تبادل اطلاعات میان گره‌ها دو طرفه اســـت. این نوع شـــبکه‌ها به ویژه در مدلســـازی فرآیندهای متغیر در زمان و مکان می‌تواند گزینه مناسبی باشد.



شبکه عصبی پس‌خور

انواع شبکه‌های عصبی مصنوعی

شبکه‌های عصبی مصنوعی از جنبه‌های مختلف قابلیت طبقه بندی دارند. با استفاده از تعداد لایه‌ها، شبکه‌های عصبی را می‌توان به شرح زیر دسته بندی کرد:

- شبکه یک لایه مانند شبکه عصبی هاپفیلد
- شبکه دولایه مانند شبکه عصبی پرسپترون
- شبکه چند لایه مانند شبکه عصبی پرسپترون چند لایه

شبکه‌های عصبی از لحاظ ساختار سلولی نیز به دسته‌هایی تقسیم می‌شوند. برخی از این شبکه‌ها را در زیر معرفی می‌کنیم.

شبکه‌های همینگ (Heming Networks)

شبکه‌های همینگ جزو شبکه‌های رقابتی می‌باشد. این شبکه‌ها برای حل مسائل شناسایی الگوهای باینری (الگوهای برداری که عناصرشان فقط دو مقداره می‌باشد) طراحی شده‌اند.

شبکه‌های هاپفیلد (Hopfield Networks)

شبکه‌های هاپفیلد جزو شبکه‌های حافظه انجمنی می‌باشند. این شبکه‌ها را می‌توان به دو نوع گسسته و پیوسته تقسیم کرد. از این نوع شبکه‌ها در مسائل بهینه‌سازی می‌توان استفاده کرد. نکته جالب توجه در مورد این شبکه‌ها عدم نیاز به الگوریتم یادگیری می‌باشد.

شبکه‌های پرسپترون (Perceptron Networks)

شبکه‌های عصبی پرسپترون، به ویژه پرسپترون چند لایه، در زمره پر کاربردترین شبکه‌های عصبی می‌باشند. این شبکه‌ها جزو شبکه‌های عصبی پیش‌خور می‌باشند که قادرند با انتخاب مناسب تعداد لایه‌ها و نرون‌ها، یک نگاشت غیر خطی را با دقت دلخواه انجام دهند. این همان چیزی هست که در بسیاری از مسائل مهندسی به عنوان راه حل اصلی مطرح می‌شود. در اغلب شبکه‌ها اطلاعات ورودی مساله را در لایه ورودی دریافت می‌کند. متغیرهای ورودی بر خروجی شبکه تاثیری دارد، در لایه خروجی متغیر با متغیرهایی قرار می‌گیرند که حاصل پیش‌بینی شبکه و خروجی مدل هستند.

تعداد لایه‌های مخفی و تعداد نرون‌های هر لایه مخفی، معمولاً با سعی و خطا تعیین می‌شود. نرون‌های لایه‌های مختلف به وسیله اتصالاتی به همدیگر متصل هستند و هر اتصال دارای یک وزن سیناپتیک است که وزن و قدرت اتصال آن دو نرون را مشخص می‌کند. یک شبکه عصبی مصنوعی با سه لایه پیش‌خور را می‌توان در مسائل گستردهای نظیر ذخیره و بازیابی اطلاعات، طبقه‌بندی الگوها، نگاشت از الگو یا فضای ورودی به الگو یا فضای خروجی، گروه‌بندی الگوهای مشابه یا پیدا کردن جواب مسائل بهینه‌یابی بکار برد.

شبکه‌های پرسپترون چند لایه (MLP)

شبکه‌های بزرگ و پیچیده‌تر معمولاً قابلیت‌ها و توانایی‌های محاسباتی بیشتر را ارائه می‌کنند، اگر چه شبکه‌ها در هر ساختار قابل تصوری شده‌اند. مرتب کردن سلول عصبی در لایه‌ها از ساختار لایه‌بندی شده بعضی از قسمت‌های مغز الگو برداری گردیده است. ثابت شده است که شبکه‌های چند لایه قابلیت و توانایی‌های فراتر از شبکه‌های تک لایه دارند و در سال‌های اخیر، الگوریتم‌های آموزشی برای آنها توسعه و بسط داده شده‌اند. شبکه‌های چند لایه ممکن است از تک لایه‌هایی که به شکل آبشار دنبال هم قرار گرفته‌اند، شکل بگیرد. خروجی یک لایه، ورودی لایه بعدی را مهیا می‌کند

شبکه پرسپترون چند لایه، شبکه‌ای پویا همراه با آموزش با نظارت بوده و از الگوریتم ها ی مختلفی نظیر الگوریتم پس انتشار خطا برای آموزش، استفاده می کند. این شبکه دارای یک لایه ورودی، یک یا چند لایه پنهان و یک لایه خروجی است . در این شبکه نرون ها ی درون اولین لایه پنهان، ورودی ها ی خود را با اعمال یک سری ضرایب وزنی از گره ها ی لایه ورودی دریافت کرده و آن ها را با هم جمع می کنند. این مقدار در حین خروج از نرون، تحت تأثیر تابع محرک قرار گرفته و برای نرون های لایه بعدی، ورودی محسوب می شود. این عمل تا رسیدن به لایه خروجی ادامه می یابد.

شیوه یادگیری پرسپترون چندلایه به صورت زیر خلاصه می شود:

1- داده های ورودی به مدل داده می شوند.
2- ضرایب وزنی به صورت تصادفی تعیین می شوند.
3- مقدار خطا با مقایسه مقدار خروجی شبکه با مقدار مطلوب محاسبه می شود. برای کاهش خطا ضرایب وزنی تغییر می کنند.
4- به گام دوم برگشته و تا رسیدن به خطای قابل چشم‌پوشی این مراحل تکرار می شود.

پارامترهای مورد نیاز برای مدل سازی با استفاده از شبکه ها ی عصبی پرسپترون چند لایه عبارتند از:

الف - تعداد پارامترهای ورودی: تعداد و نوع پارامترهای ورودی یا همان الگوی ورودی بر اساس میزان تأثیرگذاری این پارامترها بر پارامتر خروجی تعیین می شود.

ب - تعداد پارامترهای خروجی: این پارامتر وابسته به نوع مسئله مورد بررسی می باشد. معمولاً تعداد خروجی‌ها یک یا دو تاست.

ج - تعداد لایه های مخفی شبکه: به طور معمول شبکه ها ی پرسپترون چند لایه دارای ساختاری متشکل از 1 یا 2 لایه مخفی هستند. تعداد بهینه این لایه ها به روش سعی و خطا تا دست یابی به نتیجه مطلوب تعیین می گردد.

د - تعداد نرون ها ی لایه ها ی میانی: رابطه خاصی برای محاسبه تعداد نرون ها ی لایه یا لایه های مخفی وجود ندارد و معمولا برای این کار از روش سعی و خطا استفاده می شود تا جایی که شبکه دارای کمترین خطا شود. در بیشتر موارد، افزایش تعداد نرون ها ی میانی برای رسیدن به میزان دقت مطلوب مؤثر است. اما در صورتی که تعداد نرون ها ی لایه میانی از یک حدی فراتر رود، سبب ایجاد پدیده ورآموزی1 می شود که در این حالت نتیجه آموزش مطلوب ولی نتیجه آزمون نامطلوب خواهد بود.

ه – تعداد تکرار آموزش: یکی از معیارهای مهم در آموزش شبکه های پرسپترون چند لایه، تعیین تعداد تکرار فرآیند آموزش به منظور دست یابی به وزن های بهینه اســـت . در حالـــت کلی هر چه تعداد تکرار بیشتر شود، خطای مدل سازی کاهش می یابد. اما هنگامی که این تعداد از یک مقدار بیشتر شود، خطای دسته آزمایشی نیز افزایش می یابد. ایـن پـارامتر نیـز بـه روش سعی و خطا تعیین می شود.

شبکه‌های توابع پایه شعاعی (Radial Basis Function Networks)

یک شـبکه RBF را می‌توان به صـورت یک شـبکه سـه لایه در نظر گرفت که درآن لایه پنهان عمل تبدیل غیر خطی ثابتی را انجام می‌دهد بدون اینکه وزن‌ها و پارامترهای آن تغییر یابند. این لایه (لایه پنهان) شامل یک تعداد نرون و یک بردار پارامتر است که مرکز (Center) نامیده می‌شود، که می‌تواند بردار وزن لایه پنهان در نظر گرفته شود. شبکه RBF به علت خواص تقریب غیرخطی‌شان قادر به نگاشت مدل‌های پیچیده می‌باشند. به منظور استفاده از شبکه‌های RBF باید توابع در لایه‌های مخفی تعداد نرون‌های برازش کننده و الگوریتم آموزش دهنده برای پیدا کردن پارامترهای شبکه تعریف شوند. در این شبکه‌ها ورودی موثر هر نرون در لایه پنهان فاصله دکارتی بردارهای ورودی و بردارهای وزن م تهی به نرون مربوط در لایه پنهان می‌باشد. برای نرون فاصله دکارتی بین مرکز و بردار ورودی شبکه محاسبه می‌شود و از یک تابع غیر خطی گذرانده می‌شود که خروجی نرون‌ها در لایه پنهان می‌باشد، سپس لایه خروجی که ترکیب این نتایج می‌باشد شکل می‌گیرد.

آماده‌سازی داده‌ها

جهت مدل سازی با شبکه عصبی MLP، تقسیم بندی داده ها برای مراحل آموزش (Train)، آزمایش (Test) به این صورت انجام گرفت که از 80 درصد داده مربوط به صورت تصادفی برای آموزش شبکه عصبی و 20 درصد باقیمانده برای آزمایش شبکه عصبی استفاده گردید. مجموعه داده‌های مورد استفاده در این تحقیق از بخش دیتاست‌های یادگیری ماشین دانشگاه کالیفرنیا آمریکا تهیه شده است و در پایگاه داده UCI[1] قابل دسترسی است. این مجموعه داده شامل 116 ردیف است که هر کدام 9 ویژگی دارند. 64 بیمار مبتلا به سرطان سینه و 52 بیمار سالم وجود دارد. اطلاعات مربوط به ویژگی‌های این مجموعه داده عبارتند از: سن[۲]، شاخص توده بدنی[۳]، گلوکز[۴]، انسولین[۵]، مدل ارزیابی همواستاتیک[۶]، لپتین[۷]، ادیپونکتین[۸]، رزیستین[۹]، پروتئین کموتاکسی مونوسیت یک[۱۰].

متغیرهای ورودی به مدل‌های شبکه عصبی مصنوعی (MLP) در جدول زیر ارائه شده است.

متغیرهای ورودی و خروجی مدل های MLP

	نام متغیر
متغیرهای ورودی مدل	**Age**
	BMI
	Glucose
	Insulin
	HOMA
	Leptin
	Adiponectin
	Resistin
	MCP.1
متغیرهای خروجی مدل	**Labels**

[1] https://archive.ics.uci.edu/ml/datasets.php
[2] Age
[3] BMI
[4] Glucose
[5] Insulin
[6] homeostatic model assessment (HOMA)
[7] Leptin
[8] Adiponectin
[9] Resistin
[10] MCP.1

تقسیم‌بندی داده‌ها

در آموزش ماشین (Machine Learning) معمولاً داده‌ها را به دو قسمت تفکیک می‌کنند. مجموعه داده‌های آموزش و آزمایش. در این تحقیق از 80 درصد از مجموعه داده‌ها به عنوان داده‌های آموزش و 20 درصد باقی‌مانده به عنوان داده‌های آزمایش استفاده شده است.

داده‌های آموزشی (Training set): از این بخش از داده‌ها به منظور ایجاد و آموزش مدل‌ها و الگوریتم‌های مختلف یادگیری ماشین و برآورد پارامترهای آن استفاده می‌شود.

داده‌های آزمایشی (Test set): این قسمت از داده‌ها برای بررسی کارایی مدل‌ها و الگوریتم‌های مختلف یادگیری ماشین که در مرحله قبل آموزش دیده‌اند، استفاده می‌شود. اهمیت این بخش از داده‌ها در این نکته است که این مشاهدات شامل مقدارهای متغیرهای مستقل (xها) و پاسخی (y) هستند که در آموزش مدل‌های یادگیری ماشین به کار نرفته، ولی امکان مقایسه مقدار پیش‌بینی شده توسط مدل‌های یادگیری ماشین را با مقدار واقعی به ما می‌دهند؛ البته توجه داریم که این داده‌ها مدل را تحت تأثیر قرار نداده‌اند؛ پس در تعیین پارامترهای مدل نقشی نداشته و فقط برای ارزیابی مدل‌های یادگیری ماشین به کار می‌روند.

با توجه به تفکیکی که برای این دو گروه داده در نظر گرفته شد، مدل‌سازی فقط بر اساس بخش داده‌های آموزشی خواهد بود، ولی در روش اعتبارسنجی متقابل[11] که از این به بعد آن را به اختصار «CV» می‌نامیم، طی یک فرآیند تکرارشونده، قسمت داده‌های آموزشی (Training set) که به منظور مدل‌سازی به کار می‌رود، خود به دو بخش تفکیک می‌شود. در هر بار تکرار فرآیند CV، بخشی از داده‌ها برای آموزش و بخشی دیگر برای اعتبارسنجی[12] مدل به کار می‌رود. به این ترتیب این فرآیند یک روش بازنمونه‌گیری به منظور برآورد خطای مدل محسوب می‌شود.

باید توجه داشت که داده‌های آزمایشی در فرایند CV ممکن است در تکرار بعدی به عنوان داده‌های آموزشی به کار روند، در نتیجه، ماهیت آن‌ها با داده‌هایی که در قسمت قبل به عنوان داده‌های آزمایشی (Test set) معرفی شد، متفاوت است. شکل زیر به درک ماهیت داده‌های تست در فرآیند CV کمک می‌کند. مشخص است که داده‌های اعتبارسنجی بخشی از داده‌های آموزشی هستند و داده‌های آزمایشی نیز به عنوان بخشی مجزا از داده‌هایی آموزشی فرض شده‌اند. مراحل تکرار فرآیند CV نیز در تصویر به خوبی دیده می‌شود.

---

11. Cross Validation (CV)
12. Validation

نکته دیگری که در شکل زیر مشخص است، مکمل بودن مجموعه داده‌های آموزشی و اعتبارسنجی است. با انتخاب بخشی از داده‌ها برای انجام فرایند CV، بقیه داده‌ها برای آموزش به کار گرفته می‌شوند. در هر مرحله از فرایند CV، مدل به دست آمده توسط داده‌های آزمایشی برای پیش‌بینی داده‌های CV به کار گرفته و «خطا» (Error) یا «دقت» (Accuracy) حاصل از برازش مدل روی داده‌های CV محاسبه می‌شود. معمولاً میانگین این خطاها (دقت‌ها) به عنوان خطای (دقت) کلی مدل در نظر گرفته می‌شود؛ البته بهتر است انحراف معیار خطاها (دقت‌ها) نیز گزارش شود. به این ترتیب با توجه به تعداد پارامترهای مختلف (پیچیدگی مدل)، می‌توان مدل‌های متفاوتی تولید و خطای برآورد آن‌ها را به کمک روش CV اندازه‌گیری کرد. در انتها مدلی را به عنوان مدل مناسب انتخاب خواهیم کرد که دارای کمترین برآورد خطا باشد.

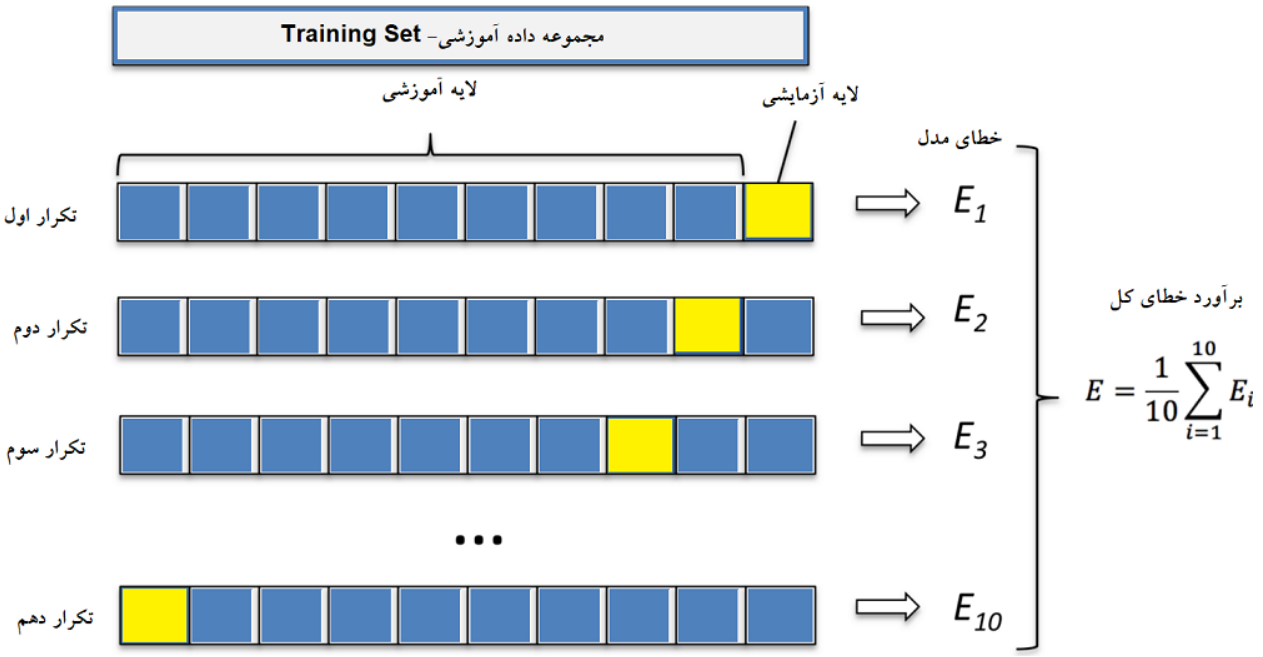
روش اعتبارسنجی متقابل (Cross Validation)

روش اعتبار سنجی متقابل K-Fold Cross Validation

بر اساس شیوه و روش انتخاب مجموعه داده‌های اعتبارسنجی، گونه‌های مختلفی از روش‌های CV معرفی شده‌اند که در اینجا به آن نمی‌پردازیم. روش اعتبار سنجی متقابلی که در این تحقیق برای آموزش مدل‌های یادگیری ماشین استفاده شده است، روش اعتبارسنجی متقابل K لایه‌ای[۱۳] می‌باشد. اگر مجموعه داده‌های آموزشی را به‌طور

13. K-Fold Cross Validation

تصادفی به k زیرنمونه یا لایه[14] با حجم یکسان تفکیک کنیم، می‌توان در هر مرحله از فرایند CV، تعداد k-1 از این لایه‌ها را به عنوان مجموعه داده آموزشی و یکی را به عنوان مجموعه داده اعتبارسنجی در نظر گرفت. شکل زیر، مراحل روش k-Fold را به خوبی نشان می‌دهد. مشخص است که با انتخاب k=10، تعداد تکرارهای فرآیند CV برابر با ۱۰ خواهد بود و دستیابی به مدل مناسب به سرعت امکان‌پذیر می‌شود. در این تحقیق، تعداد لایه‌ها یا فولدها برابر با 4 (K=4)، در نظر گرفته شده است.



روش اعتبار سنجی متقابل K-Fold Cross Validation

نرمال سازی داده‌ها

قبل از شروع مدل‌سازی ابتدا بایستی ورودی‌ها و در بعضی از موارد خروجی‌ها را نیز نرمال کرد زیرا وارد کردن داده ها به صورت خام باعث کاهش سرعت و دقت شبکه های عصبی می شود.

برای نرمال کردن داده های ورودی از فرمول زیر استفاده می کنیم، این فرمول داده ها را در بازه a و b نرمال می‌کند.

14. Fold

$$XN = a + \frac{X - X_{\min}}{X_{\max} - X_{\min}} \times (b - a)$$

در این رابطه $X_{\min}$، $X_{\max}$، XN به ترتیب مقدار مینیمم و ماکزیمم داده های ورودی و داده نرمالایز شده است. همچنین a و b نیز به ترتیب برابر با حد پایین و بالای بازه مورد نظر برای نرمالیزه کردن می‌باشد که در اینجا به ترتیب برابر با 0 و 1 می‌باشند.

معیارهای ارزیابی و اعتبارسنجی

در این تحقیق، به منظور ارزیابی کارآیی مدل‌ها، از معیارهای معتبر به شرح زیر استفاده شده است.

نرخ طبقه بندی صحیح (Correct Classification Rate)

نرخ طبقه بندی صحیح، برای مدل های طبقه بندی، به نسبت ردیف هایی که به درستی طبقه بندی شده‌اند به تعداد کل ردیف‌ها در دیتاست گفته می‌شود.

به عنوان مثال، یک نرخ طبقه بندی 0/82 به این معنی است که 82٪ از ردیف‌های مجموعه داده های آموزش به درستی بر اساس مدل طبقه بندی شده اند.

ماتریس کانفیوژن (Confusion Matrix)

در بحث «دسته‌بندی» (Classification) یک «مجموعه داده» (Data Set) با استفاده از روش‌های دسته‌بندی، هدف دستیابی به بالاترین دقت ممکن در دسته‌بندی و تشخیص دسته‌ها است. در برخی از مسائل، تشخیص صحیح نمونه‌های مربوط به یکی از دسته‌ها برای ما اهمیت بیشتری دارد. به عنوان مثال، تحقیقی را در نظر بگیرید که در آن، هدف شناسایی افراد مبتلا به یک نوع خاص از یک بیماری خطرناک است. فرض کنید برای افرادی که مبتلا به این بیماری هستند، خطر مرگ وجود دارد و جهت رفع این خطر، نیاز به دریافت نوعی داروی خاص دارند. در این شرایط، تشخیص درست بیماران دارای اهمیت بسیار زیادی است.

به این معنا که خطا در تشخیص افراد سالم قابل چشم پوشی است اما برای شناسایی افراد بیمار نمی‌توان این احتمال را به جان خرید. به عبارت دیگر، انتظار ما تشخیص تمام افراد بیمار است، بدون جا انداختن، حتی اگر فرد سالمی به اشتباه جز افراد بیمار دسته‌بندی شود. در چنین مواقعی، که دقت تشخیص یک دسته در مقایسه با دقت تشخیص کلی، اهمیت بیشتری دارد، مفهوم «ماتریس درهم‌ریختگی» (Confusion Matrix)، به کمک ما می‌آید.

بر اساس مثالی که پیش‌تر بیان شد، فرض کنید تعلق به دسته افراد بیمار را مثبت بودن (Positive) و عدم تعلق به این دسته را منفی بودن (Negative) در نظر بگیریم. هر نمونه یا فردی در واقعیت، متعلق به یکی از کلاسهای مثبت یا منفی است و از سوی دیگر، از هر الگوریتمی که برای دسته‌بندی داده‌ها استفاده شود، در نهایت هر نمونه عضو یکی از این دو «دسته» (Class) دسته‌بندی خواهد شد. بنابراین برای هر نمونه داده، یکی از چهار حالتی که در ادامه بیان شده، ممکن است اتفاق بیفتد.

- نمونه عضو دسته مثبت باشد و عضو همین کلاس تشخیص داده شود (مثبت صحیح یا True Positive)
- نمونه عضو کلاس مثبت باشد و عضو کلاس منفی تشخیص داده شود (منفی کاذب یا False Negative)
- نمونه عضو کلاس منفی باشد و عضو همین کلاس تشخیص داده شود (منفی صحیح یا True Negative)
- و در نهایت، نمونه عضو کلاس منفی باشد و عضو کلاس مثبت تشخیص داده شود (مثبت کاذب یا False Positive)

پس از اجرای الگوریتم دسته‌بندی، با توجه به توضیحات و تعاریف ذکر شده، می‌توان عملکرد یک طبقه‌بند را به کمک جدولی به شکل زیر بررسی کرد.

		برچسب پیش‌بینی شده	
		مثبت	منفی
برچسب شناخته شده	مثبت	TP	FN
	منفی	FP	TN

این جدول را اصطلاحا ماتریس درهم ریختگی می‌گویند. جدول یا ماتریس درهم ریختگی، نتایج حاصل از طبقه‌بندی را بر اساس اطلاعات واقعی موجود، نمایش می‌دهد. حال بر اساس این مقادیر می‌توان معیارهای مختلف ارزیابی دسته بند و اندازه‌گیری دقت را تعریف کرد. پارامتر دقت (Accuracy)، متداول‌ترین، اساسی‌ترین

و ساده‌ترین معیار اندازه‌گیری کیفیت یک دسته‌بند است و عبارت است از میزان تشخیص صحیح دسته‌بند در مجموع دو دسته. این پارامتر در واقع نشان‌گر میزان الگوهایی است که درست تشخیص داده شده‌اند و بر اساس ماتریس ارائه شده در بالا، به شکل زیر فرموله و تعریف می‌شود:

$$\text{Accuracy} = (TP+TN) / (TP+FN+FP+TN)$$

البته، پارامتر دقت معمولا به صورت درصد بیان می‌شود. اما پارامترهای دیگری نیز علاوه بر معیار دقت وجود دارند که می‌توان به سادگی از این ماتریس استخراج کرد. یکی از متداول‌ترین آن‌ها، معیار حساسیت (Sensitivity) است که آن را «نرخ پاسخ‌های مثبت درست» (True Positive Rate) نیز می‌گویند. حساسیت به معنی نسبتی از موارد مثبت است که آزمایش آن‌ها را به درستی به عنوان نمونه مثبت تشخیص داده است. این پارامتر به صورت زیر محاسبه می‌شود:

$$\text{Sensitivity (TPR)} = TP / (TP+FN)$$

در واقع، «حساسیت» همان معیار بحث شده در مورد مثال بالا است. معیاری که مشخص می‌کند دسته‌بند، به چه اندازه در تشخیص تمام افراد مبتلا به بیماری موفق بوده‌است. همانگونه که از رابطه فوق مشخص است، تعداد افراد سالمی که توسط دسته‌بند به اشتباه به عنوان فرد بیمار تشخیص داده شده‌اند، هیچ تاثیری در محاسبه این پارامتر ندارد و در واقع زمانی که پژوهشگر از این پارامتر به عنوان پارامتر ارزیابی برای دسته‌بند خود استفاده می‌کند، هدفش دستیابی به نهایت دقت در تشخیص نمونه‌های کلاس مثبت است.

در نقطه مقابل این پارامتر، ممکن است در مواقعی دقت تشخیص کلاس منفی حائز اهمیت باشد. از متداول‌ترین پارامترها که معمولا در کنار حساسیت بررسی می‌شود، پارامتر خاصیت (Specificity)، است که به آن «نرخ پاسخ‌های منفی درست» (True Negative Rate) نیز می‌گویند. خاصیت به معنی نسبتی از موارد منفی است که آزمایش آن‌ها را به درستی به عنوان نمونه منفی تشخیص داده است. این پارامتر به صورت زیر محاسبه می‌شود:

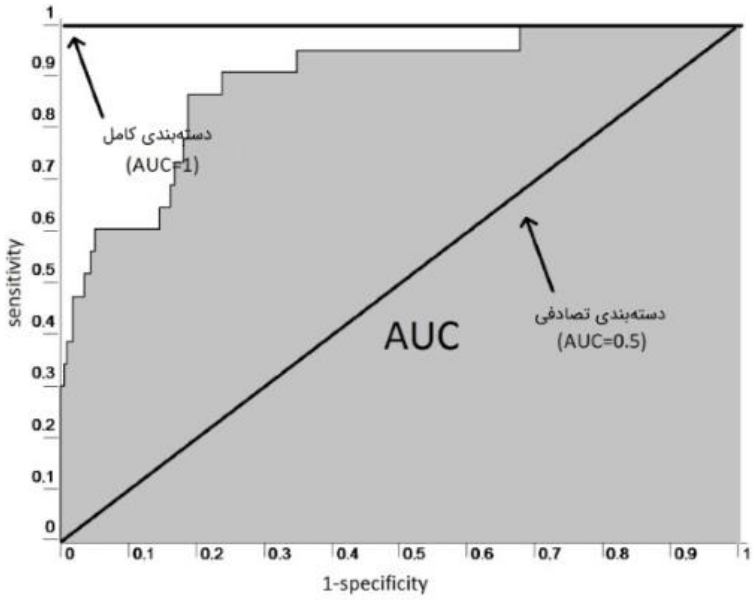
$$\text{Specificity (TNR)} = TN / (TN+FP)$$

این دو پارامتر (حساسیت و خاصیت) نیز مشابه معیار دقت، معمولا به صورت درصد بیان می‌شوند. واضح است که پیش‌بینی عالی، پیش‌بینی است که مقادیر Sensitivity و Specificity مربوط به آن، هر دو صد درصد باشند؛ اما احتمال وقوع این اتفاق در واقعیت بسیار کم است و همیشه یک حداقل خطایی وجود دارد. پارامترهای حساسیت و خاصیت، بنابر ماهیتی که دارند همواره در رقابت با یکدیگر هستند. یعنی افزایش یکی با کاهش

دیگری همراه است و برعکس. همین وضعیت منجر به تولید ابزاری دیگر برای ارزیابی کیفیت دسته‌بندها شده است.

منحنی ROC و سطح زیر آن AUC

«منحنی مشخصه عملکرد سیستم» (Receiver Operating Characteristic | ROC)، عبارت است از منحنی که ارتباط بین دو پارامتر حساسیت و خاصیت را بیان می‌کند. چنانکه در شکل زیر مشاهده می‌کنید، محور عمودی این نمودار نشان‌دهنده نرخ مثبت صحیح (Sensitivity)، و محور افقی نشان‌دهنده مقدار نرخ مثبت غلط (One-Specificity) است. نتایج مختلف دسته‌بندی نشانگر نقاط مختلف بر روی این نمودار هستند و در نهایت یک منحنی را تشکیل می‌دهند. با توجه به شکل زیر، در بهترین حالت و با فرض طبقه‌بندی صد درصد صحیح در هر دو دسته، نقطه مربوطه عبارت است از نقطه گوشه بالای سمت چپ، یعنی نقطه $(0,1)$ و نیز با فرض دسته‌بندی به صورت تصادفی، نقطه متناظر در منحنی، یکی از نقاط موجود روی خط واصل نقطه $(0,0)$ و نقطه $(1,1)$ خواهد بود. در واقعیت، منحنی حاصل از یک دسته‌بندی، منحنی بین این دو حالت است.



مساحت زیر این نمودار (Area Under Curve)، به عنوان یک معیار برای ارزیابی عملکرد دسته‌بند مورد استفاده قرار می‌گیرد. با توجه به توضیحاتی که پیش‌تر ارائه شد، بدیهی است که در حالت ایده‌آل، مساحت زیر منحنی برابر با بیشترین مقدار خود، یعنی یک است. بنابراین، هر چه مساحت زیر نمودار به عدد یک نزدیکتر باشد، به معنای بهتر بودن عملکرد دسته‌بند است. علاوه بر دو پارامتر حساسیت و خاصیت، پارامترهای دیگری هم از ماتریس درهم‌ریختگی استخراج می‌شوند که هر یک بیان کننده مفهومی هستند و کاربردهای متفاوتی دارند.

پارامتر مهم دیگری به نام «معیار اف» (F-Measure) وجود دارد که برای ارزیابی عملکرد دسته‌بندها بسیار مورد استفاده قرار می گیرد و از ترکیب دو پارامتر حساسیت و ارزش اخباری مثبت حاصل می شود. با این توضیح که پارامتر ارزش اخباری مثبت را اصطلاحا دقت (Precision)، و حساسیت را اصطلاحا صحت (Recall) می نامند، «معیار اف» به دو صورت زیر تعریف می‌شود:

F-measure= 2 * (Recall * Precision) / (Recall + Precision)

$$\mathrm{F}_1 = 2 \cdot \frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

ماتریس درهم ریختگی، با وجود منطق و ساختار ساده ای که دارد، مفهومی قدرتمند است که در انواع تحقیقات، می‌تواند به تنهایی اطلاعاتی جامع از نحوه عملکرد دسته‌بند ارائه کند.

میانگین مربعات خطا (Mean Squared Error)

روشی برای برآورد میزان خطاست که در واقع تفاوت بین مقادیر تخمینی و آنچه تخمین زده شده، است. MSE به دو دلیل تقریباً همه جا مثبت است (صفر نیست) یک اینکه تصادفی است و دوم به این دلیل که تخمین‌گر اطلاعاتی که قابلیت تولید تخمین دقیق تری دارد را حساب نمی‌کند. پس این شاخص که مقداری همواره نامنفی دارد، هرچقدر مقدار آن به صفر نزدیکتر باشد، نشان دهنده میزان کمتر خطاست. مقدار این شاخص به صورت زیر بیان می شود:

$$MSE = \frac{1}{n} \times \sum_{i=1}^{n} [(x_{imeas} - x_{ipred})^2]$$

n , x_{ipred} , x_{imeas} به ترتیب برابر با تعداد متغیر اندازه‌گیری شده، مقدار متغیر پیش‌بینی شده و مقدار متغیر اندازه‌گیری شده می‌باشد.

مجذور میانگین مربعات خطا (Root Mean Square Error)

ریشهٔ میانگین مربعات خطا (RMSE) نیز یک تابع تناسب یا تابع هدف است و در واقع مجذور شاخص میانگین مربعات خطاست. این شاخص به عنوان معیاری از خطای مطلق بین متغیر شبیه سازی و مشاهده‌ای است. مقدار این شاخص آماری بین صفر تا بی نهایت متغیر است. هر چه مقدار این شاخص

کمتر باشد شبیه سازی بهتری صورت گرفته اســت و مقــدار بهینــهٔ آن صفر است. مقدار این شاخص به صورت زیر بیان می شود:

$$RMSE = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} [(x_{imeas} - x_{ipred})^2]}$$

n , x_{ipred} , x_{imeas} به ترتیب برابر با تعداد متغیر اندازه‌گیری شــده، مقدار متغیر پیش‌بینی شــده و مقدار متغیر اندازه‌گیری شده می‌باشد.

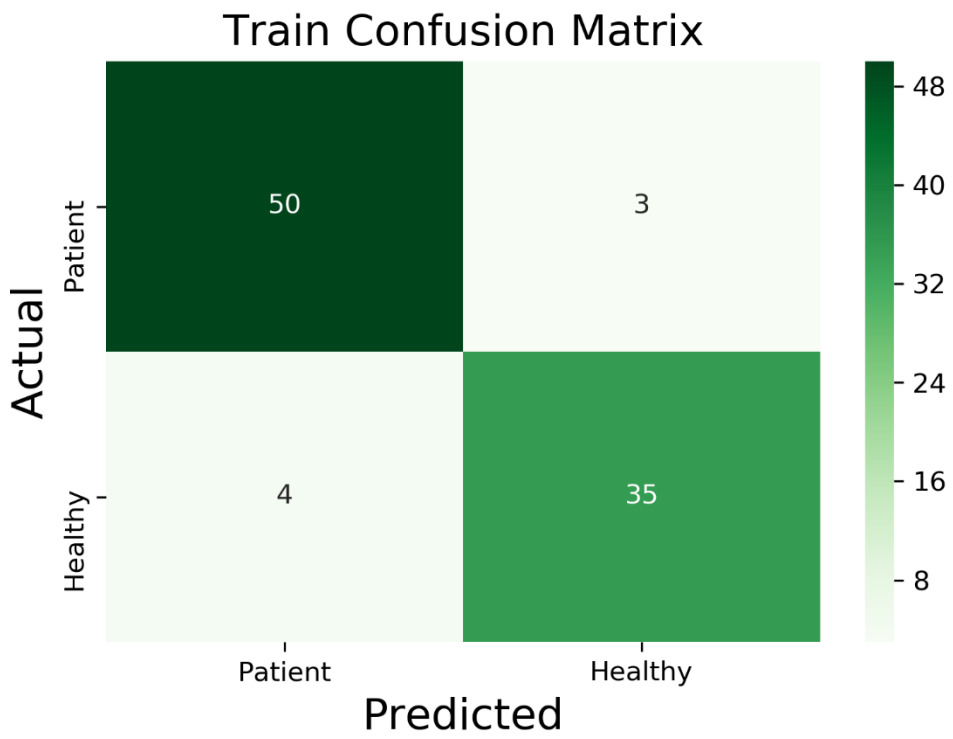
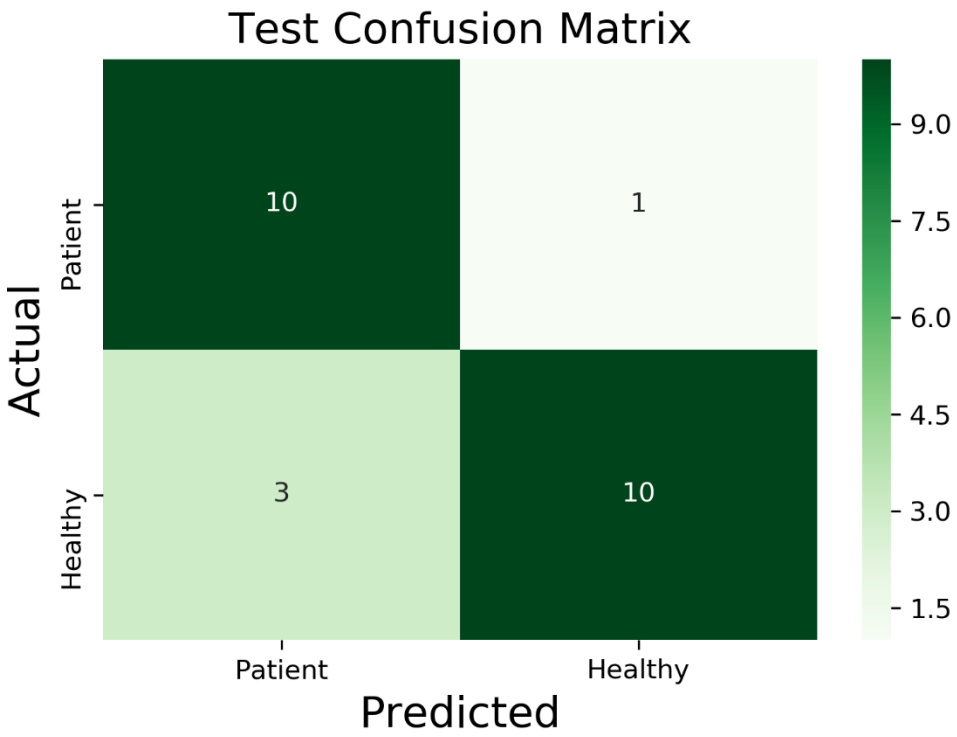
نتایج مدل‌سازی و طبقه‌بندی توسط شبکه عصبی MLP

نتایج مدل شبکه عصبی مصنوعی MLP برای طبقه بندی و تشخیص بیماری سرطان سینه

مرحله	ACC (%)	RMSE	Sensitivity	Specificity	F-score	AUC
TR	0.92	0.28	0.94	0.90	0.92	0.92
TS	0.83	0.41	0.91	0.77	0.83	0.84

در ادامه ماتریس‌های کانفیوژن برای دو حالت آموزش، آزمایش ارائه می‌شود.

- حالت آموزش (Train)



- حالت آزمایش (Test)