وزارت علوم، تحقیقات و فناوری

دانشگاه دامغان

دانشکده ریاضی و علوم کامپیوتر

پایان‌نامه کارشناسی ارشد
ریاضی کاربردی (گرایش تحقیق در عملیات)

شبکه عصبی بازگشتی برای حل رده‌ای از مسائل بهینه‌سازی نامحدب

توسط:

احمد ذوالفقاریساقی

استادان راهنما:

دکتر سهراب عفتی

دکتر الهه ظهوریان‌آزاد

بهمن ۱۳۹۱

بسم الله الرحمن الرحيم

به نام خدا

شبکه عصبی بازگشتی برای حل رده‌ای از مسائل بهینه‌سازی نامحدب

توسط:
احمد ذوالفقاریساقی

پایان‌نامه

ارائه شده به تحصیلات تکمیلی دانشگاه به عنوان بخشی از فعالیت‌های تحصیلی لازم
برای اخذ درجه کارشناسی ارشد

در رشته
ریاضی کاربردی (گرایش تحقیق در عملیات)
از دانشگاه دامغان

ارزیابی و تأیید شده توسط کمیته پایان‌نامه با درجه: **بسیار خوب**

دکتر سهراب عفتی دانشیار گروه ریاضی کاربردی-کنترل و بهینه‌سازی دانشکده علوم ریاضی دانشگاه فردوسی مشهد (استاد راهنما)

دکتر الهه ظهوریان آزاد استادیار گروه ریاضی کاربردی-احتمال دانشکده ریاضی و علوم کامپیوتر دانشگاه دامغان (استاد راهنما)

دکتر علی عباسی ملایی استادیار گروه ریاضی کاربردی-تحقیق در عملیات دانشکده ریاضی و علوم کامپیوتر دانشگاه دامغان (داور اول)

دکتر حنیف حیدری استادیار گروه ریاضی کاربردی- بهینه‌سازی دانشکده ریاضی و علوم کامپیوتر دانشگاه دامغان (داور دوم)

دکتر محمد ابری استادیار گروه ریاضی محض-توپولوژی دانشکده ریاضی و علوم کامپیوتر دانشگاه دامغان (نماینده تحصیلات تکمیلی)

بهمن ۱۳۹۱

تقدیم به

پدر مهربان

مادر دلسوز

و همسر عزیزم

و به تمام آزادمردانی که نیک می‌اندیشند و عقل و منطق را پیشه خود نموده و جز رضای الهی و پیشرفت و سعادت جامعه، هدفی ندارند

دانشمندان، بزرگان، و جوانمردانی که جان و مال خود را در حفظ و اعتلای این مرز و بوم فدا نموده و می‌نمایند.

«اللّهم عجّل لولیّک الفرج»

سپاسگزاری

سپاس بی‌کران پروردگار یکتا را که هستی‌مان بخشید و به طریق علم و دانش رهنمونمان شد و به همنشینی رهروان علم و دانش مفتخرمان نمود و خوشه چینی از علم و معرفت را روزی‌مان ساخت. و سلام و درود بر محمّد و خاندان پاک او، طاهران معصوم، هم آنان که وجودمان وامدار وجودشان است؛ و نفرین پیوسته بر دشمنان ایشان تا روز رستاخیز...

بدون شک جایگاه و منزلت معلم، اجلّ از آن است که در مقام قدردانی از زحمات بی شائبه ی او، با زبان قاصر و دست ناتوان، چیزی بنگاریم. بر حسب وظیفه از پدر و مادر عزیزم، این دو معلم بزرگوار که همواره بر کوتاهی و درشتی من، قلم عفو کشیده و کریمانه از کنار غفلت هایم گذشته اند و در تمام عرصه های زندگی یار و یاوری بی چشم داشت برای من بوده اند

از استاد با کمالات و شایسته، جناب آقای دکتر سهراب عفتی که در کمال سعه صدر، با حسن خلق و فروتنی، از هیچ کمکی در این عرصه بر من دریغ ننمودند و زحمت راهنمایی این رساله را بر عهده گرفتند

از استاد دلسوز و صبور سرکار خانم دکتر الهه ظهوریان آزاد که بدون مساعدت ایشان، این پروژه به نتیجه مطلوب نمی رسید

و از اساتید فرزانه جناب آقای دکتر عباسی ملایی و دکتر حنیف حیدری که زحمت داوری این رساله را متقبل شدند کمال تشکر و قدردانی را دارم.

در پایان از همسر مهربانم به پاس عاطفه ی سرشار و گرمای امید بخش وجودش، که در تمام طول تحصیل همراه و همگام من بوده است تشکر می کنم. همچنین از دوست عزیزم آقای مهدی رنجبر که در آماده سازی این اثر مرا یاری نمودند صمیمانه سپاسگزارم.

چکیده

شبکه عصبی بازگشتی برای حل رده‌ای از مسائل بهینه‌سازی نامحدب

به وسیله‌ی:
احمد ذوالفقاریساقی

در این پایان‌نامه یک شبکه عصبی برای حل مسائل بهینه‌سازی نامحدب پیشنهاد می‌دهیم. ابتدا مسأله بهینه‌سازی نامحدب را در همسایگی جواب بهینه، با روش توان p (جزیی) به یک مسأله بهینه‌سازی محدب تبدیل می‌کنیم. سپس یک سیستم دینامیکی برای حل مسأله تغییریافته ارائه می‌دهیم. در این‌جا نشان می‌دهیم تناظر یک به یک بین نقاط تعادل سیستم دینامیکی و جواب بهینه مسأله معادل یا جواب بهینه مسأله اصلی وجود دارد. سرانجام چند مثال برای نشان دادن عملکرد شبکه عصبی می‌آوریم.

واژه های کلیدی: بهینه‌سازی نامحدب، شبکه عصبی، مدل توان p، مدل توان p جزیی، معادله تصویر

فهرست مطالب

فهرست مطالب ه

فهرست شکل‌ها ز

۱ مقدمه‌ای بر شبکه‌های عصبی ۱

۱-۱ انسان و کامپیوتر ۱

۱-۲ مفهوم شبکه ۳

۱-۳ شبکه عصبی مصنوعی ۴

۱-۴ الگوریتم پس‌انتشار خطا ۹

۱-۵ تاریخچه و کاربرد شبکه‌های عصبی مصنوعی در حل مسائل بهینه‌سازی ۱۵

۲ پیش‌نیازها ۱۸

۲-۱ تعاریف اولیه ۱۸

۲-۲ سیستم‌های دینامیکی ۲۱

۲-۳ نامساوی وردشی ۲۵

۳ مباحثی در مورد مسائل برنامه‌ریزی غیرخطی ۳۳

۳-۱ مقدمه ۳۳

۳-۲ شرایط بهینگی مسائل برنامه‌ریزی نامقید ۳۳

۳-۳ توابع جریمه ۳۶

۳-۴ بهینه‌سازی مقید ۳۹

۳-۵ روش‌های تکراری ۴۴

۴ حل مسائل بهینه‌سازی نامحدب با شبکه‌های عصبی ۴۹

۴-۱ مقدمه ۴۹

۴-۲ شکاف دوگانی صفر برای رده‌ای از مسائل بهینه‌سازی نامحدب ۵۰

۴-۳ حل مسائل بهینه‌سازی نامحدب با شبکه عصبی تصویر ۶۲

۴-۴ ارائه شبکه عصبی برای حل مسائل بهینه‌سازی نامحدب ۶۶

۴-۵ نتایج عددی ۶۹

۴-۶ نتیجه‌گیری ۷۴

مراجع ۷۵

واژه‌نامه فارسی به انگلیسی ۷۸

واژه‌نامه انگلیسی به فارسی ۸۰

فهرست شکل‌ها

۱-۱ ساختار یک نورون ۵
۱-۲ مدل ریاضی ساده‌شده سلول عصبی ۶
۱-۳ پرسپترون ۳ لایه با اتصالات کامل ۷
۱-۴ رفتار تابع سیگموئید ۸
۱-۵ پرسپترون دو لایه ۱۱

۲-۱ پایداری به مفهوم لیاپانوف ۲۳
۲-۲ پایداری مجانبی ۲۴

۳-۱ شکاف دوگانی ۴۱

۴-۱ تشریح هندسی مسأله اولیه و دوگان ۵۳
۴-۲ منحنی‌های $w(y)$ و $\hat{w}(y^{۲})$ ۵۸
۴-۳ عدم همگرایی شبکه عصبی تصویر به نقطه بهینه مسأله (۴۰.۴) ۷۰
۴-۴ همگرایی شبکه عصبی تصویر به نقطه بهینه مسأله (۴۱.۴) ۷۰
۴-۵ همگرایی شبکه عصبی برای مسأله (۴۱.۴)با نقاط آغازین مختلف ۷۱
۴-۶ همگرایی شبکه عصبی برای مسأله (۴۲.۴)با نقاط آغازین مختلف ۷۱
۴-۷ عدم همگرایی شبکه عصبی به نقطه بهینه مسأله (۴۳.۴) ۷۲
۴-۸ همگرایی شبکه عصبی برای مسأله (۴۴.۴) ۷۳

فصل ۱

مقدمه‌ای بر شبکه‌های عصبی

۱-۱ انسان و کامپیوتر

در ابتدا این سوال را مطرح می‌کنیم که چرا همیشه با تردید سؤال می‌شود که آیا ممکن است که کامپیوترها از انسان باهوش‌تر باشند؟ آیا به این دلیل نیست که ما انسان هستیم ونمی‌خواهیم بپذیریم که توده‌ای از سیلیسیم و فلز می‌تواند آنچه را که ما نمی‌توانیم انجام دهیم، انجام دهد؟ و یا بدان دلیل است که شیوه عمل کامپیوترها با ما متفاوت است؟ برای مثال محاسبه جمع چند صد عدد هشت یا نه رقمی برای یک کامپیوتر عمل پیش پا افتاده‌ای است در جایی که ماهرترین افراد نیز به سختی از عهده آن بر می‌آیند. آیا این امر کامپیوترها را از ما باهوش‌تر می‌کند؟ شاید جواب اولیه ما بتواند این باشد. اینک موضوع دیدن را در نظر بگیرید، این عمل کاملاً ساده به‌نظر می‌رسد. به شیء نگاه می‌کنیم، می‌بینیم که آن چیست و به آن نامی اطلاق می‌کنیم و احتمالاً کاری با آن انجام می‌دهیم. مثلاً اگر آن شئ یک فنجان قهوه باشد ما آن را به همان صورت تشخیص می‌دهیم و آن را می‌نوشیم و اگر آن شئ توپ فوتبالی باشد که به سرعت به طرف ما می‌آید ما تصمیم به عملی پیچیده‌تر می‌گیریم. ولی می‌بینیم که کامپیوترها در انجام کارهای بسیار ساده در زمینه دیدن نیز درمانده‌اند. آن‌ها ممکن است اشیاء ساده را تشخیص دهند ولی کنترل کردن یک دست کامپیوتری برای برداشتن یک فنجان قهوه به فنون بسیار پیچیده‌ای نیاز دارد. شاید مسأله این است که چون کامپیوترها می‌توانند بعضی کارها را که ما آن‌ها را در مدت زمان قابل ملاحظه‌ای انجام می‌دهیم همچون جمع کردن اعداد، در کمترین زمان انجام دهند و یا می‌توانند نام‌ها و آدرس‌ها را ماه‌ها بعد به درستی به یاد بیاورند، از آن‌ها انتظار می‌رود که در سایر زمینه‌ها نیز چنین عمل کنند. بدین علت آن‌گاه که نمی‌توانند انتظارات ما را برآورده کنند مأیوس می‌شویم. این در واقع هدفی است که دست‌اندرکاران هوش‌مصنوعی دنبال می‌کنند، لیکن آنها پس از

گذشت سی و پنج سال تحقیقات گسترده نمی‌توانند این ادعا را داشته باشند که به چنین کامپیوترهای هوشمندی دست یافته‌اند. هدف هوش‌مصنوعی را می‌توان در این جمله خلاصه کرد که می‌خواهد در نهایت به کامپیوترهایی دست یابد که بسیار قوی‌تر از انسان باشند. چرا کامپیوترها نمی‌توانند کارهایی را که ما انجام می‌دهیم انجام دهند؟ یکی از دلایل را می‌توان در نحوه ساختار آنها جستجو کرد. به طور منطقی می‌توان انتظار داشت که سیستم‌هایی با ساختار مشابه عملکرد مشابهی داشته باشند. چنانچه به درون یک کامپیوتر نگاه کنیم تعدادی تراشه الکترونیکی حاوی مدارهای مینیاتوری خواهیم دید که با انواع مقاومت‌ها و سایر قطعات الکترونیکی در صفحات مدار جای داده شده‌اند.

اگر به درون مغز نگاه کنیم به هیچ‌وجه چنین ساختاری را مشاهده نخواهیم کرد. بررسی اولیه ما چیزی جز مجموعه گره خورده از ماده‌ای خاکستری رنگ نشان نمی‌دهد. بررسی بیشتر نشان می‌دهد که مغز از اجزای ریزی تشکیل شده است. لیکن این اجزا به شیوه‌ای بی‌نهایت پیچیده مرتب شده‌اند و هر جزء به هزاران جزء دیگر متصل است. شاید این تفاوت در شیوه ساختار، علت اصلی اختلاف بین مغز و کامپیوتر باشد. کامپیوترها طوری طراحی شده‌اند که یک عمل را بعد از عملی دیگر با سرعت بسیار زیاد انجام دهند. لیکن مغز ما با تعداد اجزای بیشتر اما با سرعت بسیار کم‌تر کار می‌کند. در حالی که سرعت عملیات در کامپیوترها به میلیون‌ها محاسبه در ثانیه بالغ می‌شود، سرعت عملیات در مغز تقریباً بیشتر از ده بار در ثانیه نمی‌باشد. ولی مغز در یک لحظه با تعداد زیادی اجزا به طور هم‌زمان کار می‌کند، کاری که از عهده کامپیوترها برنمی‌آید. کامپیوتر ماشینی سریع اما پیاپی کار است در حالی که مغز شدیداً ساختاری موازی دارد.

۱-۱-۱ محاسبات در مغز انسان

مغز انسان پیچیده‌ترین ساختار شناخته شده در دنیا است. مغز شامل بیش از ۱۰ میلیارد سلول عصبی یا نورون [۱] می‌باشد و به طور متوسط هر نورون به وسیله ۱۰۰۰۰ سیناپس به نورون‌های دیگر متصل می‌شود. شبکه نورون‌های مغز، یک سیستم پردازش اطلاعات موازی عظیم را تشکیل می‌دهند که با اصل عملکرد کامپیوترهای متداول در تناقض است زیرا کامپیوترهای موجود، پردازشگرهایی هستند که دستورات را به صورت سری اجرا می‌کنند. مغز قابلیت‌های قابل ملاحظه‌ای دارد که از آن جمله می‌توان به موارد زیر اشاره کرد :

- کارایی آن تحت آسیب‌های جزیی، کمی کاهش می‌یابد اما در مقابل برنامه‌ها و سیستم‌های مهندسی شکننده هستند، بدین معنی که اگر قسمت‌های دلخواهی از آن‌ها را پاک کنند یا از بین ببرند، مانند آن است که تمام آن‌ها را پاک یا از بین برده‌اند.

[۱] Neuron

- مغز می‌تواند از تجربه‌ها آموزش ببیند و تحت همین آموزش می‌تواند قسمت‌های آسیب دیده را ترمیم کند.
- مغز می‌تواند حجم عظیمی از محاسبات موازی را به طور موثری انجام دهد. به عنوان مثال درک دیداری پیچیده در کمتر از ۱۰۰ میلی ثانیه اتفاق می‌افتد و در این زمان کوتاه حدود ۷ الی ۸ مرحله پردازش انجام می‌شود.
- مغز توسط هوش(یا خود آگاهی) ما حمایت می‌شود و هنوز هیچ‌کس چگونگی این مسأله را نمی‌داند.

به عنوان شاخه‌ای از هوش‌مصنوعی، شبکه‌های عصبی تلاش می‌کنند که کامپیوترهایی با قابلیت‌های خیلی نزدیک به قابلیت‌های مغز تولید کنند که این امر از طریق پیروی از وضعیت‌های مشابه از پردازش اطلاعات در مغز حاصل می‌شود.

۱-۲ مفهوم شبکه

یکی از روش‌های کارامد در حل مسائل پیچیده، شکستن آن به زیرمسائل ساده‌تر است که هر کدام از این زیربخش‌ها به نحو ساده‌تری قابل درک و توصیف باشند. در حقیقت یک شبکه، مجموعه‌ای از این ساختارهای ساده است که در کنار یکدیگر سیستم پیچیده نهایی را توصیف می‌کنند. شبکه‌ها انواع مختلفی دارند اما همگی آن‌ها از دو مؤلفه تشکیل می‌شوند:

۱. مجموعه‌ای از گره‌ها؛ هر گره در حقیقت واحد محاسباتی شبکه است که ورودی‌ها را گرفته و بر روی آن پردازش انجام می‌دهد تا خروجی بدست آید. پردازش انجام شده توسط گره می‌تواند از ساده‌ترین نوع پردازش‌ها نظیر جمع کردن ورودی‌ها تا پیچیده‌ترین محاسبات را شامل شود. در حالت خاص، یک گره می‌تواند خود، شامل یک شبکه دیگر باشد.

۲. اتصالات بین گره‌ها؛ این اتصالات نحوه گذر اطلاعات بین گره‌ها را مشخص می‌کند. در حالت کلی اتصالات می‌توانند تک‌سویه یا دوسویه باشند.

تعامل بین گره‌ها از طریق این اتصالات سبب بروز یک رفتار کلی از سوی شبکه می‌گردد که چنین رفتاری به تنهایی در هیچ یک از اجزای شبکه دیده نمی‌شود. جامع بودن این رفتار کلی بر عملکرد گره‌ها سبب تبدیل شبکه به یک ابزار توانمند می‌شود. به عبارت دیگر، مجموعه ساده‌ای از اجزا وقتی در قالب یک شبکه باشند می‌توانند رفتاری از خود بروز دهند که هیچ یک از آن اجزا به تنهایی قادر به بروز چنین مشخصه‌ای نخواهد بود.

۱-۳ شبکه عصبی مصنوعی

آن‌چنان که بیان شد انواع مختلفی از شبکه‌ها وجود دارد. در این بین شبکه‌ای وجود دارد که گره را به عنوان یک نورون مصنوعی در نظر می‌گیرد. در اصطلاح، این چنین شبکه‌ای را شبکه عصبی مصنوعی[۲] یا به اختصار ANN می‌نامند.

شبکه‌های عصبی نوعی مدل‌سازی ساده انگارانه از سیستم‌های عصبی واقعی هستند که کاربرد فراوانی در حل مسائل مختلف در علوم دارند. حوزه کاربرد این شبکه‌ها آن چنان گسترده است که از کاربردهای طبقه‌بندی گرفته تا کاربردهایی نظیر درون‌یابی، تخمین، آشکارسازی و ... را شامل می‌شود. شاید مهمترین مزیت این شبکه‌ها، توانمندی در کنار سهولت استفاده از آن‌ها باشد.

نورون عنصر اصلی مغز است و به تنهایی مانند یک واحد پردازش منطقی عمل می‌کند. نورون‌ها دو نوع هستند. نورون‌های داخل مغز که در فاصله‌های حدود ۱۰۰ میکرون به یکدیگر متصل هستند و نورون‌های خارجی که قسمت‌های مختلف مغز را به یکدیگر و مغز را به ماهیچه‌ها و اعضای حسی را به مغز متصل می‌کنند. نحوه عملیات نورون بسیار پیچیده است. هر نورون ورودی‌های متعددی دارد که با یکدیگر به طریقی جمع می‌شوند. اگر در یک لحظه تعداد ورودی‌های فعال نورون به حد کفایت برسد، نورون نیز فعال شده و آتش می‌کند. در غیر این صورت نورون به صورت غیر فعال و آرام باقی می‌ماند. نمایشی از ویژگی‌های عمده نورون در شکل ۱-۱ آورده شده است. بدنه نورون سوما [۳] نامیده می‌شود. به سوما رشته‌های نامنظم طولانی متصل است که به آن‌ها دندریت [۴] می‌گویند. قطر این رشته‌ها اغلب از یک میکرون نازکتر است و اشکال شاخه‌ای پیچیده‌ای دارند. دندریت‌ها نقش اتصالاتی دارند که ورودی‌ها را به نورون‌ها می‌رسانند. یکی از عناصر عصبی متصل به هسته نورون آکسون [۵] نامیده می‌شود. این عنصر بر خلاف دندریت از نظر الکتریکی فعال می‌باشد و به عنوان خروجی نورون عمل می‌کند. آکسون‌ها همیشه بر روی خروجی سلول‌ها مشاهده می‌شوند، ولی اغلب در ارتباط‌های بین نورونی غایب‌اند. در این مواقع خروجی‌ها و ورودی‌ها هر دو بر روی دندریت‌ها واقع می‌شوند.

یک نورون مصنوعی در حقیقت مدلی محاسباتی است که از نورون‌های عصبی واقعی انسان، الهام گرفته است. نورون‌های طبیعی، ورودی خود را از طریق سیناپس‌ها دریافت می‌کنند. این سیناپس‌ها برروی دندریت‌ها یا غشاء عصب قرار دارند. در یک عصب واقعی، دندریت‌ها دامنه پالس‌های دریافتی را تغییر می‌دهند که نوع این تغییر در طول زمان یکسان نمی‌ماند و در اصطلاح، توسط عصب یاد گرفته

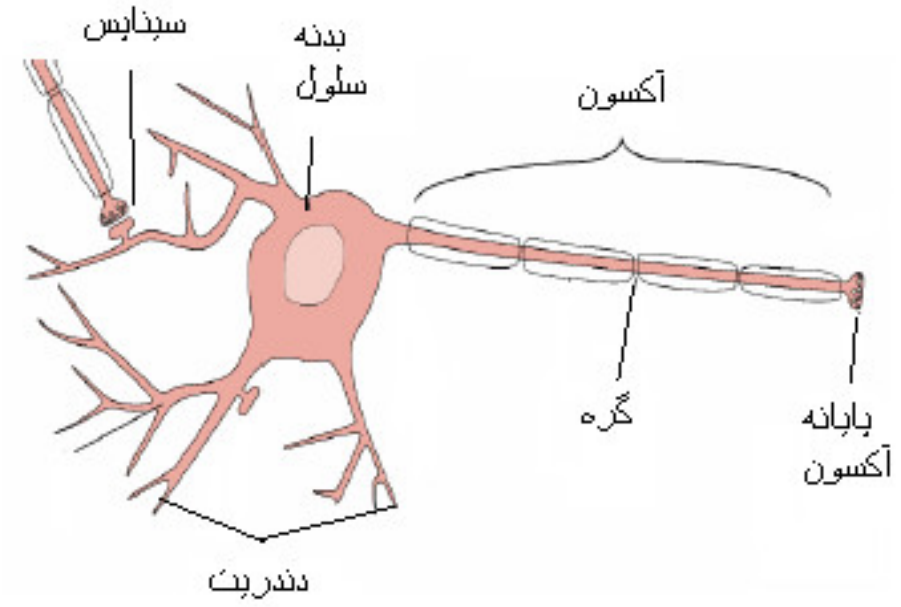
---

[۲] Artificial Neural Network

[۳] Soma

[۴] Dendrite

[۵] Axon

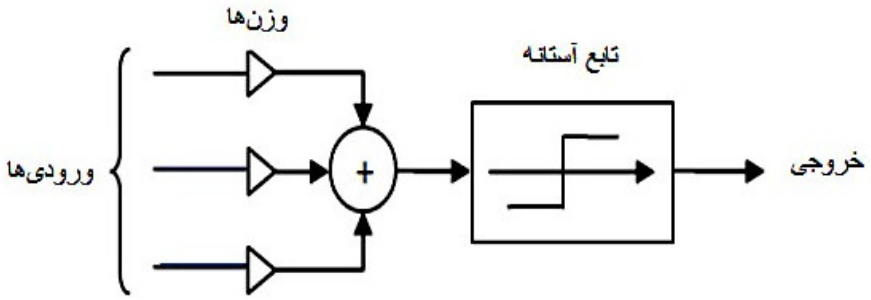


شکل ۱-۱: ساختار یک نورون

می‌شود. اگر سیگنال دریافتی به حد کافی قوی باشد (از یك مقدار آستانه بیشتر شود)، عصب فعال شده و سیگنالی را در طول آکسون منتشر می‌کند. این سیگنال نیز به نوبه خود می‌تواند به یك سیناپس دیگر وارد شده و سایر اعصاب را تحریك کند.

به هنگام مدل کردن اعصاب، از پیچیدگی‌های آن‌ها صرف‌نظر می‌شود و تنها به مفاهیم پایه‌ای بها داده می‌شود، چرا که در غیر این‌صورت رویکرد مدل‌سازی بسیار دشوار خواهد شد. در یك نگاه ساده، مدل یك عصب باید شامل ورودی‌هایی باشد که در نقش سیناپس انجام وظیفه کنند. این ورودی‌ها در وزن‌هایی ضرب می‌شوند تا قدرت سیگنال را تعیین کنند. نهایتاً یك عملگر ریاضی تصمیم‌گیری می‌کند که آیا نورون فعال شود یا خیر و اگر جواب مثبت باشد، میزان خروجی را مشخص می‌سازد. بنابراین شبکه عصبی مصنوعی با استفاده از مدل ساده شده عصب واقعی به پردازش اطلاعات می‌پردازد.

با توجه به این توضیحات، می‌توان مدل ساده‌ای برای توصیف یك نورون (یك گره در شبکه عصبی مصنوعی) پیشنهاد کرد. این مدل در شکل ۱-۲ نشان داده شده است. جدای از ساده‌سازی‌های اعمال شده، تفاوت اصلی این مدل با واقعیت در این است که در شبکه واقعی، ورودی‌ها سیگنال‌های زمانی هستند حال آن که ورودی‌های این مدل، اعداد حقیقی هستند. در مدل ارائه شده در شکل ۱-۲ تنوع بسیاری وجود دارد. از جمله این که وزن‌های یك شبکه عصبی، که مقدار خروجی را منتقل می‌کنند، می‌توانند مثبت یا منفی باشند. از طرفی، توابع مورد استفاده برای آستانه‌گذاری می‌توانند بسیار متنوع



شکل ۱-۲: مدل ریاضی ساده‌شده سلول عصبی

باشند. از جمله مشهورترین این توابع، می‌توان به توابعی نظیر **سیگموئید** [۶] ، **آرکتانژانت**[۷] و **آرکسینوس**[۸] اشاره کرد.

این توابع باید پیوسته، هموار و مشتق‌پذیر باشند. همچنین تعداد گره‌های ورودی می‌تواند متغیر باشد. البته زیاد شدن تعداد این گره‌ها، به وضوح تعیین وزن‌ها را با مشکل روبرو می‌کند. لذا باید به دنبال روش‌هایی برای حل این موضوع باشیم. روند تعیین وزن‌های بهینه و تنظیم مقادیر آنها عمدتاً به صورت بازگشتی انجام می‌شود. بدین منظور شبکه را با استفاده از قواعد و داده‌ها آموزش داده و با استفاده از قابلیت یادگیری شبکه، الگوریتم‌های متنوعی پیشنهاد می‌گردد که همگی سعی در نزدیك کردن خروجی تولید شده توسط شبکه به خروجی ایده‌ال و مورد انتظار دارند.

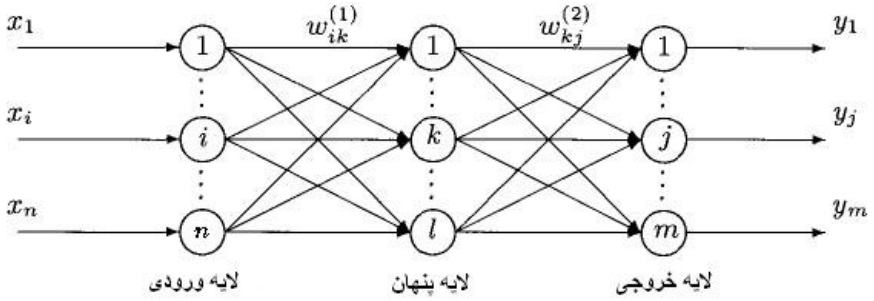
هرچند نحوه مدل کردن نورون اساسی‌ترین جزء در کارایی شبکه عصبی است، اما نحوه برقراری اتصالات و چیدمان (توپولوژی) شبکه نیز فاکتور بسیار مهم و اثرگذاری است. باید توجه داشت که چیدمان مغز انسان آنقدر پیچیده است که نمی‌توان از آن به عنوان مدلی برای اعمال در شبکه عصبی استفاده نمود، چرا که مدلی که ما استفاده می‌کنیم، یك مدل ساده شده است، در حالی‌که چیدمان مغز از المانهای بسیار زیادی استفاده می‌کند. یکی از ساده‌ترین و در عین حال کارامدترین چیدمان‌های پیشنهادی برای استفاده در مدل‌سازی عصب‌های واقعی،مدل پرسپترون چندلایه [۹] یا به اختصار MLP، و به معنای مدل هوشمند چند لایه است که از یك لایه ورودی، یك یا چند لایه پنهان و یك لایه خروجی تشکیل یافته است. در این ساختار، تمام نورون‌های یك لایه به تمام نورون‌های لایه بعد متصل هستند. این چیدمان اصطلاحاً یك شبکه با اتصالات کامل را تشکیل می‌دهد. شکل ۱-۳ شمای یك شبکه

[۶] sigmoid

[۷] arctan

[۸] arcsin

[۹] Multi layer perceptron



شکل ۱-۳: پرسپترون ۳ لایه با اتصالات کامل

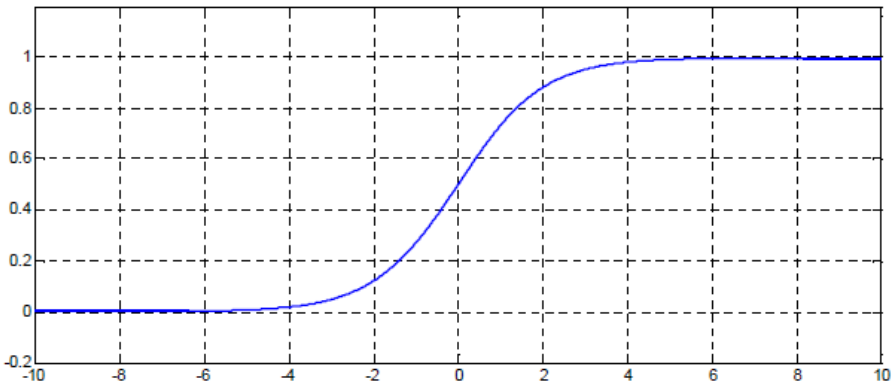
پرسپترون سه لایه را نشان می‌دهد. به سادگی می‌توان استنباط نمود که تعداد نورون‌های هر لایه، مستقل از تعداد نورون‌های دیگر لایه‌ها می‌باشد. توجه به این نکته حائز اهمیت است که در شکل ۱-۳، هر دایره تجمیع شده عمل جمع و آستانه‌گذاری (عبور از تابع غیرخطی سیگموئید) است. در حقیقت هر دایره در این شکل، مدلی است از جمع کننده و بلوك آستانه‌گذاری نشان داده شده در شکل ۱-۲، که به منظور سهولت نمایش به این فرم نشان داده شده است. با توجه به شکل، خروجی عصب j-ام (در لایه آخر) را می‌توان به صورت زیر نشان داد:

$$y_j = sgm\left(\sum_{k=1}^{l} sgm\left(\sum_{i=1}^{n} x_i w_{ik}^{(1)}\right) w_{kj}^{(2)}\right) \qquad (1.1)$$

که در آن، $w^{(1)}$و $w^{(2)}$ به ترتیب نشان‌دهنده وزن‌های لایه پنهان و لایه خروجی بوده و منظور از x_i ورودی عصب i-ام است. sgm نیز تابع سیگموئید است که به صورت زیر تعریف می‌گردد:

$$sgm(x) = \frac{1}{1 + e^{-x}} \qquad (2.1)$$

شکل ۱-۴ تابع سیگموئید را نشان می‌دهد. می‌توان دید که این تابع همواره رفتاری بین ۰ و ۱ دارد. به طور کلی شبکه‌های عصبی مصنوعی از لحاظ یادگیری بر دو دسته اند: شبکه‌های **با وزن ثابت** و شبکه‌های **با وزن متغیر** (شبکه‌های یادگیرنده). خود شبکه‌های یادگیرنده نیز به دو دسته **با سرپرست** و **بدون سرپرست** تقسیم می‌شوند. در شبکه‌های با سرپرست، در فاز آموزش از نمونه‌هایی استفاده می‌گردد که خروجی ایده‌ال متناظر با آن‌ها از پیش مشخص است. به عبارت دیگر در این‌گونه شبکه‌ها، نمونه‌های داده ورودی، برچسب دارند. در شبکه‌های بی‌سرپرست، بر اساس یك معیار (مثلاً فاصله) و بر اساس نوعی رقابت، خروجی مورد نظر در کلاس جداگانه قرار می‌گیرد. با توجه به این که شبکه عصبی، مدل ساده شده اعصاب بدن است، درست به مانند آن‌ها قابلیت یادگیری دارد. به عبارت دیگر، شبکه با استفاده از اطلاعاتی که از ورودی و توسط سرپرست خود دریافت می‌کند، قادر به فراگیری روند موجود در الگوهاست. لذا به طور مشابه با انسان، روند یادگیری در شبکه عصبی نیز از مدل‌های



شکل ۱-۴: رفتار تابع سیگموئید

انسانی الهام گرفته است، بدین صورت که مثال‌های بسیاری را به دفعات بایستی به شبکه ارائه نمود تا بتواند با تغییر وزن‌های شبکه، خروجی موردنظر را دنبال کند. ارائه نمونه داده‌های ورودی به شبکه عصبی به دو روش امکان پذیر است:

۱. **روش ارائه یک‌جا:** در این روش، تمام نمونه‌ها به شبکه ارائه می‌گردند و در آخر، خطای شبکه نسبت به کل نمونه‌ها محاسبه شده و وزن‌ها بر اساس آن خطا تغییر می‌کنند. در مرحله بعد، مجدداً تمام داده‌ها یك بار دیگر به شبکه ارائه شده و روند فوق نظیر به نظیر انجام می‌پذیرد تا نهایتاً خطا به سطح قابل قبولی برسد. مسلماً این روش پیچیده و زمان بر بوده و نیاز به حافظه زیادی دارد. همچنین امکان گیرکردن الگوریتم در مینیمم‌های محلی وجود دارد.

۲. **روش ارائه الگو:** در این روش، در هربار نمونه‌ها به صورت تك‌تك به شبکه داده شده و خطای متناظر با همان داده بلافاصله محاسبه شده و بر اساس آن، وزن‌های شبکه تغییر می‌کنند. سپس نمونه بعدی به شبکه ارائه شده و روند بالا مشابهاً انجام می‌پذیرد. چون در این روش، در هر گام، اصلاح وزن‌ها بر اساس هر نمونه انجام می‌پذیرد، الگوریتم همگرایی خوبی داشته و با توجه به ماهیت تصادفی موجود در ارائه تکی داده‌ها، خطر مینیمم‌های محلی منتفی است.

به منظور آموزش شبکه و اصلاح وزن‌ها تا رسیدن به یك خطای معنادار، روش‌های بسیار زیادی وجود دارد. یکی از مشهورترین این روش‌ها، الگوریتم پس‌انتشار خطا[۱۰] است که در ادامه توضیح داده می‌شود.

[۱۰] Error back propagation algorithm

۱-۴ الگوریتم پس‌انتشار خطا

این الگوریتم که در سال ۱۹۸۶ توسط روملهارت [۱۱] و مك‌کلیلاند [۱۲] پیشنهاد گردید، در شبکه‌های عصبی پیش‌سو مورد استفاده قرار می‌گیرد. پیش‌سو بودن به این معناست که نورون‌های 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 پیش روی خود تا خروجی را تکرار کند. روش پس‌انتشار خطا از روش‌های با سرپرست است، به این مفهوم که نمونه‌های ورودی برچسب خورده‌اند و خروجی مورد انتظار هر یك از آن‌ها از پیش معلوم است. لذا خروجی شبکه با این خروجی‌های ایده‌ال مقایسه شده و خطای شبکه محاسبه می‌گردد. در این الگوریتم ابتدا فرض بر این است که وزن‌های شبکه به طور تصادفی انتخاب شده‌اند. در هرگام 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 تحریك هر عصب به صورت جمع وزن‌دار ورودی‌های مربوط به آن عصب درنظر گرفته می‌شود. بدین ترتیب با فرض n ورودی و این که w وزن‌های متناظر بین لایه ورودی و لایه بعد باشد می‌توان نوشت:

$$A_j(x,w) = \sum_{i=1}^{n} x_i w_{ij} \qquad (۳.۱)$$

که در آن x_i مؤلفه i-ام x و w_{ij} وزن متناظر بین ورودی i-ام و مؤلفه j-ام لایه بعدی است. به وضوح می‌توان دید که خروجی تابع تحریك عصب فقط به ورودی و وزن‌های متناظر بستگی دارد. با فرض این‌که تابع خروجی، سیگموئید باشد می‌توان خروجی عصب j-ام را به صورت زیر نوشت:

$$O_j(x,w) = sgm\left(A_j(x,w)\right) = \frac{1}{1+e^{-A_j(x,w)}} \qquad (۴.۱)$$

همان‌گونه که در شکل ۱-۴ دیده شد، تابع سیگموئید به ازای اعداد منفی بزرگ، بسیار نزدیك به صفر است و برای اعداد مثبت بزرگ، مقداری بسیار نزدیك به ۱ دارد و در این بین به طور هموار تغییر می‌کند به نحوی که در $x = 0$ دقیقاً از حد واسط بازه $[0, 1]$ یعنی ۰/۵ عبور می‌کند. هم‌چنین با دقت در رابطه (۴.۱) در می‌یابیم که خروجی فقط به مقدار تابع تحریك بستگی دارد که به نوبه خود به ورودی و وزن‌ها مرتبط می‌شود. لذا برای تغییر خروجی باید وزن‌ها تغییر کنند. آن‌چنان که پیش از این نیز بیان شد، هدف فرایند آموزش، رسیدن به خروجی مطلوب (یا نزدیك به مطلوب) است. بدین ترتیب ابتدا باید تابع خطای هر نورون را تعریف کنیم. این خطا از اختلاف خروجی واقعی شبکه و

[۱۱] David E. Rumelhart

[۱۲] Mc Clelland

خروجی مورد انتظار به صورت زیر بدست می‌آید:

$$E_j(x, w, d_j) = (O_j(x, w) - d_j)^۲ \qquad (۵.۱)$$

انتخاب مربع تفاضل بین خروجی واقعی O_j و خروجی مطلوب d_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 کوچك‌تر شدن آن می‌گردد. بر این اساس می‌توان خطای کلی شبکه را به فرم مجموع خطای تك‌تك عصب‌های لایه خروجی نوشت، لذا داریم:

$$E(x, w, d) = \sum_{j=۱}^{m} E_j(x, w, d_j) = \sum_{j=۱}^{m} (O_j(x, w) - d_j)^۲ \qquad (۶.۱)$$

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:

- روش گرادیان شیب
- روش نیوتن
- روش اندازه حرکت
- روش آنتروپی متقابل

در این‌جا روش گرادیان شیب به دلیل این که ساده‌ترین و درعین حال پرکاربردترین روش است را مورد بررسی قرار می‌دهیم که در ادامه روابط آن بیان خواهد شد. در روش گرادیان شیب، ابتدا یك تابع هزینه درجه دوم تعریف می‌گردد که عبارت است از:

$$J(w) = \frac{۱}{n} \sum_{i=۱}^{n} J(w, i) = \frac{۱}{n} \sum_{i=۱}^{n} \left(\frac{۱}{۲} \sum_{j=۱}^{m} (O_j - d_j)^۲ \right) \qquad (۷.۱)$$

که در آن m تعداد نورون‌های لایه خروجی است. هدف نهایی الگوریتم پس‌انتشار خطا، مینیمم کردن این تابع هزینه است. بر اساس روش گرادیان شیب، با توجه به درجه ۲ و مثبت بودن تابع هزینه، فرض می‌شود این تابع رفتاری سهموی دارد. لذا برای رسیدن به مینیمم کلی آن بایستی در خلاف جهت شیب تابع حرکت کنیم. بنابراین با فرض این‌که وزن‌ها در ابتدای کار به صورت تصادفی انتخاب می‌شوند، باید شیب تابع خطا را نسبت به وزن‌ها محاسبه نموده و در جهت خلاف آن، وزن‌ها را تغییر دهیم و این روند را تا آن‌جا ادامه دهیم که به مینیمم کلی یا یك خطای قابل قبول برسیم. بنابراین میزان تغییر وزن‌ها در هرگام عبارت است از:

$$\triangle w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \qquad (۸.۱)$$