

بهینه سازی توپولوژی شبکه های عصبی با رویکرد نظریه بازی ها

پویا خانی^۱، محمد ایزدی^{۲*}

^۱ دانشجوی کارشناسی ارشد گرایش هوش مصنوعی، دانشکده علوم و مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

^۲ استادیار، دانشکده علوم و مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

چکیده

شبکه های عصبی مصنوعی^۱ در بسیاری از حوزه های مختلف از خود اثر مثبت به سزایی نشان داده اند لذا انتخاب یک معماری برای این شبکه ها یک مسئله بسیار حیاتی و مهم است و تاثیر زیادی بر روی کارایی آن ها میگذارد که اغلب به صورت دستی انجام می شود. به عنوان یک راه حل ساده و سرراست، اغلب شبکه های تماما متصل^۲ انتخاب میشوند، اما قسمت زیادی از انتهای شبکه اضافی هستند و در بهترین حالت آن قسمت ها برای نتیجه گیری های بعدی، نامربوط محسوب میشوند و در بدترین حالت یک معماری با پارامتر های زیاد، مانع بهینه بودن شبکه و همچنین نفوذپذیر شدن نسبت به داده های خصمانه^۳ میشود. قدم اول در حذف این مولفه های اضافی، شناختن آن ها است که نیازمند محاسبه سهم مشارکت هر مولفه به تنهایی در کارایی کل شبکه است. در پژوهش های قبلی، روش های هیوریستیک^۴ مبتنی بر توزیع وزن نتایج مثبتی از خود نشان داده اند اما درک تئوری خوبی از خود نشان نداده اند و همچنین نتایج آن ها ایده آل نیستند. بنابراین در این مقاله، برای تشخیص یک مولفه مهم از غیرمهم در شبکه عصبی مصنوعی، از یک معیار مبتنی بر نظریه بازی ها به نام مقدار شیپلی^۵ استفاده میکنیم. ابتدا برای هر شبکه عصبی یک بازی ائتلافی^۶ معادل طراحی میکنیم که در آن نورو^۷ ها با هم دیگر ائتلاف هایی تشکیل میدهند و مقدار شیپلی میانگین سهم مشارکت هر نورون در آن ائتلاف را نشان میدهد. با اینکار ما میتوانیم نورون های با مقدار شیپلی کم که نشان دهنده سهم مشارکت کم و در نتیجه بی اهمیت بودن آن در شبکه است را حذف کنیم و به این ترتیب میتوانیم با تکرار این روند، شبکه عصبی مصنوعی را تا حد امکان ساده کنیم. در این مقاله دو روش بالا-پایین^۸ مبتنی بر نظریه بازی ها و با استفاده از مقدار شیپلی ارائه میشود که روش اول روش ساده تر و مختص مسئله های محدود است و روش دوم روش کامل تری نسبت به روش اول است که می توان آن را حتی بر روی مسائل طبقه بندی تصاویر و متون استفاده کرد.

کلمات کلیدی: شبکه عصبی، مقدار شیپلی، بهینه سازی توپولوژی^۹، هرس^{۱۰} شبکه عصبی

۱- مقدمه

تعداد نورون موجود در هر لایه است. رویکرد بعدی، رویکرد بالا-پایین است که در ابتدا با یک معماری بزرگ شروع میکند و به تدریج مولفه های کم اثر^{۲۰} را حذف میکند و در نتیجه به تدریج معماری کوچک تر و به عبارتی هرس شده میشود. وقتی نورون های موجود در یک شبکه عصبی را به عنوان عضو هایی ببینیم که با همدیگر به رقابت و مشارکت میپردازند، نظریه بازی ها میتواند یک رویکرد تئوری خوب برای انتخاب مهم ترین بازیگر در این بازی، ارائه دهد. به صورت دقیق تر، بازی های ائتلافی (یا همان بازی های مشارکتی^{۲۱}) در این حوزه به ما این دید را میدهد که گروهی از نورون ها را به عنوان یک ائتلاف در رقابت با گروه دیگری از نورون ها به عنوان ائتلاف دیگر، ببینیم. در این مقاله ابتدا یک رویکرد بالا-پایین که بر خلاف روش های پیشین، بر مبنای نرخ خطا یا وزن بین نورون ها کار نمیکند، بلکه از یک ایده مبتنی بر نظریه بازی ها به نام مقدار شیپلی استفاده میکند. مقدار شیپلی اثرگذاری هر مولفه در کارایی کل شبکه را محاسبه میکند. البته به دلیل اینکه محاسبه مقدار شیپلی نیازمند تشکیل دادن تمام ائتلاف های ممکن است و این کار را سخت میکند، ما از یک پروسه نمونه برداری^{۲۲} استفاده میکنیم که به وسیله آن میتوانیم مقدار شیپلی را تقریب بزنیم. در نهایت نشان میدهم که روش جدید این مقاله، از لحاظ بهینگی و کارایی بهتر از روش های مبتنی بر هیوریستیک (روش های پیشین) است. از آنجایی که روش اول گفته شده در این مقاله، فقط در دامنه کوچکی از مسائل کاربرد دارد و از نوع تک-استراتژی^{۲۳} است، در ادامه روش پیشرفته تر و کامل تری

معماری یک شبکه عصبی مصنوعی تاثیر بسیار زیادی بر روی کارایی آن میگذارد [۱،۲،۳]. طراحی ساختار یک شبکه عصبی مصنوعی، کار سختی است که نیازمند دانش حرفه ای و تجربه زیاد است لذا پیدا کردن ساختار بهینه یک شبکه عصبی چند لایه، یک موضوع حل نشده است (ان پی کامل^{۱۱} است)، که افراد اغلب از روش آزمون و خطا با این مسئله برخورد میکنند. یعنی اینکه کاربر توپولوژی های مختلفی را امتحان میکند، آن شبکه را آموزش^{۱۲} میدهد و در نهایت خطاهای آموزش و آزمون^{۱۳} را محاسبه میکند. در حالی که این رویکرد بسیار ساده است، در عین حال بسیار از لحاظ پیچیدگی زمانی^{۱۴} غیر قابل قبول است. همچنین بنا به نظر کاربر شبکه و با هیوریستیک های مختلف، نتایج متفاوت در رابطه با تعداد نورون های موجود در لایه یا لایه های مخفی^{۱۵} بدست می آید. رویکرد دیگری که افراد زیادی در راستای انتخاب بهینه ساختار شبکه عصبی مصنوعی انتخاب میکنند، انتخاب لایه های تماما متصل است که به غیر از کارایی بالا و پارامترپذیری زیاد شبکه، مشکلاتی از قبیل ریسک بالای بیش-برازش^{۱۶} داده های آموزشی و همچنین آسیب پذیری بیشتر نسبت به مثال های خصمانه را همراه خود دارد.

رویکرد های خوب دیگری برای این کار نیز وجود دارد. یکی از آن ها رویکرد پایین-بالا^{۱۷} است که با یک معماری کوچک اولیه شروع میکند و به تدریج مولفه های (مثل لایه ها، نورون ها و یا وزن ها^{۱۸}) بیشتری اضافه میشوند. یک مثال برای رویکرد پایین-بالا، جستجوی توری^{۱۹} [۴] برای کاوش تعداد لایه های مخفی و

از رویکرد استفاده از مقدار شیپلی را ارائه میدهم که توسط آن میتوانیم دسته بندی تصاویر و متون را نیز انجام دهیم و همچنین از نوع چنداستراتژی^{۲۴} است.

۲- پژوهش های پیشین

الگوریتم پایین-بالا بر مبنای اعتبارسنجی متقابل^{۲۵} [۵]، شبکه را با تعداد کمی از نورون ها در لایه مخفی ایجاد میکند و سپس به صورت متوالی بر اساس دقت^{۲۶} مورد نیاز کاربر، نورون های جدید را اضافه میکند. برای تعریف کردن شرط پایان یافتن این پروسه، الگوریتم از زیرمجموعه ای داده ای آموزشی در دسترس برای اعتبارسنجی متقابل، استفاده میکند. پس نورون های جدید در صورتی اضافه میشوند که دقت شبکه را روی داده های آموزشی و اعتبارسنجی متقابل، افزایش دهند.

الگوریتم تازه به دوران رسیده^{۲۷} [۶] نمونه دیگری از رویکرد پایین-بالا مخصوص ساخت شبکه های طبقه بندی کننده^{۲۸} است. این الگوریتم ابتدا با یک جفت نورون شروع کرده و جفت های جدید را به شرطی اضافه میکند که شبکه روی داده های آموزشی، خطا ایجاد کند.

الگوریتم همبستگی آبخاری^{۲۹} [۷] فرض میکند که گره های مخفی^{۳۰} هم به گره های موجود در لایه ورودی و هم به تمام گره های مخفی بعدی، متصل است. عملکرد آن ها این است که خطاهای باقیمانده خروجی را تصحیح کنند. در این الگوریتم فقط مقدار بزرگی همبستگی مهم است. برای حداکثر کردن همبستگی، گره های مخفی جدید اضافه شده و وزن های ارتباطی آن ها تنظیم میشود. در [۸] یک مرور جامع از تکنیک های پایین-بالا گفته شده است.

الگوریتم آسیب مغزی بهینه^{۳۱} [۹] نمونه ای از رویکرد بالا-پایین است که برای مصالحه بین پیچیدگی شبکه و خطاهای آموزشی از ایده تئوری اطلاعات^{۳۲} و اطلاعات مشتق دوم^{۳۳} استفاده میکند.

الگوریتم جراح مغز بهینه^{۳۴} [۱۰] نسخه پیشرفته تر الگوریتم قبلی است که از روش های مبتنی بر مقدار بزرگی^{۳۵} بهتر عمل میکند.

الگوریتم اسکلت بندی^{۳۶} [۱۱] از اطلاعات شبکه برای ارزیابی عملکرد یا مرتبط بودن^{۳۶} نورون های موجود در شبکه استفاده میکند. سپس بر اساس این مقدار مرتبط بودن، مولفه هایی که اطلاعات اضافی انتقال میدهند را هرس میکند و به عبارتی یک اسکلت نهایی ایجاد میکند.

الگوریتم کمینه سازی تکاملی شبکه^{۳۷} [۱۲] یک الگوریتم ژنتیک استاندارد است که یک قدم اضافی دارد و آن حذف کردن وزن های کوچک است. این الگوریتم برای کنترل کننده های عصبی کاملاً بازگشتی^{۳۸} طراحی شده است.

مقدار شیپلی برای انتخاب ویژگی^{۳۹} نیز مورد استفاده قرار میگیرد. الگوریتم/انتخاب مشارکت^{۴۰} [۱۳، ۱۴] بر اساس تحلیل شیپلی چندآشفته^{۴۱} است و متوالیای بی فایده بودن یک ویژگی را توسط حذف رو به جلو^{۴۲} یا رو به عقب^{۴۳} تخمین میزند.

در [۱۵] یک رویکرد مکمل گفته شد که تعامل های بین دو نورون در یک شبکه عصبی مصنوعی، به عنوان یک بازی غیرمشارکتی دیده میشود.

۳- روش پیشنهادی اول

یک بازی در نظریه بازی ها، متشکل از مجموعه ای از بازیگرها، اعمال ممکن و توابع سودمندی به ازای هر بازیگر است. تابع سودمندی به هر خروجی تصمیم-مشترک^{۴۴}، یک مقدار عددی نسبت میدهد.

راه حل های مبتنی بر نظریه بازی زیادی وجود دارد که ایده اصلی آن ها، تقسیم کردن سود یک ائتلاف به اعضای درون آن ائتلاف، برای بازی هایی که سود انتقال-پذیر^{۴۵} دارند، است. یکی از این راه حل ها، مقدار شیپلی [۱۶] است که تلاش میکند یک تقسیم بندی منصفانه بر اساس سهم مشارکت هر بازیگر در آن ائتلاف، ارائه دهد.

برای یک ائتلاف S ، مقدار کل سودی که میتواند بین اعضای آن ائتلاف تقسیم شود را $v(S)$ مینامیم. در این حالت مقدار شیپلی برای یک عضو، در اصل میانگین پیشرفت بخاطر حضور او در ائتلافی است که آن عضو در آن حضور دارد [۱۷، ۱۸].

برای اینکه مقدار شیپلی را تعریف کنیم، فرض میکنیم A مجموعه ای از بازیگر ها باشد و مرتب شده اعضای A بر اساس حروف الفبا را $\pi(A)$ مینامیم. برای یک بازیگر به نام a ، $P(\pi, a)$ مجموعه ای از بازیگر های درون A هستند که در ترتیب π قبل از a ظاهر شده اند. در نتیجه مقدار شیپلی بازیگر a میشود [۱۹]:

$$\phi(a) = \frac{1}{|A|!} \sum_{\pi} [v(P(\pi, a) \cup \{a\}) - v(P(\pi, a))] \quad (1)$$

مقدار شیپلی یکتا است و همیشه وجود دارد اما محاسبه مقدار دقیق آن یک مسئله آن پی کامل است.

با توجه به طبیعت توزیع شده ی محاسبه ی عصبی، هر مولفه شبکه (یک ارتباط یا نورون) یک میزان مشارکت معینی در خروجی نهایی شبکه دارد. این مشارکت میتواند بحرانی باشد، یعنی بدون آن به میزان زیادی از کارایی شبکه از دست میرود. همچنین میتواند اضافی باشد، یعنی با نبودن آن مولفه، مابقی مولفه های جبران میکنند و کارایی از بین نمیرود. این معیار ها دقیقاً چیزی است که مقدار شیپلی تخمین میزند: میانگین مشارکت حاشیه ای^{۴۶} هر بازیگر در سود هر ائتلافی که در آن حضور دارد.

در روش پیشنهادی اول، ابتدا با یک شبکه عصبی پیچیده شروع کرده و با حذف یک مولفه یا گروهی از مولفه ها که مشارکت کمتری در بدست آوردن دقت شبکه میکنند، شبکه را بهینه سازی میکنیم.

معیار ارزیابی کارایی، دقت در طبقه بندی و رگرسیون است. معمولاً آنالیز ائتلاف های مبتنی بر نظریه بازی روی بازی های نرمالیز شده اعمال میشود، که در آن ها ارزش یک ائتلاف یک نفره 0، ارزش ائتلاف شامل همه بازیگر ها 1 و ارزش هر ائتلاف دیگری عددی در بازه [0,1] است. بنابراین ما به یک معیار ارزیابی کارایی نیاز داریم که آن هم بین [0,1] خروجی دهد بر خلاف مثلاً خطای میانگین مربع^{۴۹}.

در این روش پیشنهادی، ما از یک معیار ترکیبی برای دقت استفاده میکنیم:

$$a = \frac{r^2 + e}{2} \quad (2)$$

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \cdot \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (3)$$

در این رابطه \bar{X} و \bar{Y} میانگین مقدار بردار های X و Y و n تعداد عنصر های درون بردار های X و Y است. بنابراین r^2 ضریب تعیین^{۴۷} است.

e معیار اندازه گیری خطا است که در مسائل طبقه بندی، از درصد نمونه هایی که درست طبقه بندی شده اند و در مسائل رگرسیون، از خطای میانگین مربع نرمال شده^{۴۸} استفاده میکند:

$$e = 1 - NRMSE = 1 - \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2}}{X_{max} - X_{min}} \quad (4)$$

همانطور که گفته شد، محاسبه دقیق مقدار شیپلی، ممکن نیست و تقریب آن باید محاسبه شود زیرا برای محاسبه مقدار شیپلی، فقط یک افراز اعضا به ائتلاف های خاصی مد نظر نیست و هر ائتلاف ممکن است بررسی شود که اینکار بسیار سخت و پیچیده است.

برای یک مولفه شبکه خاص، یک تعداد مناسب m از ائتلاف های تصادفی تولید و دقت شبکه را با حضور و بدون حضور آن مولفه حساب میکنیم. این مقدار اختلاف همان عبارت $[v(P(\pi, a) \cup \{a\}) - v(P(\pi, a))]$ در رابطه (۱) است.

در نهایت از تمام این اختلاف ها به ازای هر ائتلاف، میانگین گرفته میشود.

از نظر آماری هرچه تعداد m را بزرگ تر در نظر بگیریم، مقدار تقریب زده شده به مقدار واقعی شیپلی نزدیک تر میشود. لذا با توجه به میزان منابعی که در اختیار داریم باید مقدار m را در نظر بگیریم.

پس از اینکه مقدار شیپلی برای تمام مولفه های شبکه اندازه گیری شد، مرحله ساده سازی آغاز میشود که در آن مولفه هایی که مقدار شیپلی آن ها از یک حد آستانه θ کمتر باشد، حذف میشوند. لذا با تغییر دادن مقدار θ میتوانیم کاری کنیم که در هر مرحله ۱ مولفه حذف شود و یا اینکه یک مجموعه ای از مولفه ها حذف شوند.

از آنجایی که کارایی شبکه با هر حذف کمتر میشود، یک پروسه بهینه سازی از نوع تپه نوردی تکاملی^{۴۹} بعد از هر مرحله ساده سازی، بر روی شبکه اعمال میشود. به این صورت که از حالت اولیه (یا حالت فعلی در مراحل بعد)، به تعداد b همسایه از آن با عمل جهش^{۵۰} تولید میشود. یک همسایه در اصل یک شبکه دیگر است که با تغییر دادن حداکثر $\pm 20\%$ به یکی از مقادیر وزن های اتصال^{۵۱} یا مقادیر نورون، بدست می آید. حال همسایه با بیشترین دقت به عنوان حالت فعلی بعدی انتخاب میشود و این روال به تعداد m نسل تکرار میشود.

اگر بعد از بهینه سازی شبکه، دقت شبکه از یک حد کمینه p کمتر شد، حالت فعلی قبلی بازگردانده میشود و فاز بهینه سازی خاتمه می یابد.

شبکه اولیه توسط الگوریتم انتشار رو به عقب^{۵۲} [۲۲-۲۰] آموزش داده میشود، اما مهم است که اشاره کنیم به راحتی اینکار را با روش های ترکیبی نیز میتوان انجام داد. تابع فعالسازی^{۵۳} نورون ها از نوع تانژانت هذلولی^{۵۴} در نظر گرفته شد که عموماً

از آن در شبکه های چند لایه استفاده میشود، گرچه روش پیشنهادی گفته شده با هر تابع فعالسازی دیگری نیز میتواند کار کند.

۳-۱- مطالعات موردی روش اول

در این قسمت قصد داریم کارایی شبکه های عصبی آموزش دیده مختلف را برای مسائل طبقه بندی و رگرسیون بررسی کنیم.

در تمام موارد ما از مقادیر زیر برای پارامتر های مدل مورد نظرمان استفاده میکنیم: تعداد ائتلاف ها $m = \min(|A|, 250)$ ، حداقل دقت مورد انتظار شبکه $p = 0.9$ ، ماکسیمم حد آستانه مقدار شیپلی تقریب زده شده برا تصمیم گیری در رابطه با حذف کردن یا نکردن مولفه مورد نظر از شبکه $\theta = \theta_s$ است که در آن $\theta_s \in \{0, 0.1, 0.25\}$ و a_s همان میانگین مقادیر شیپلی تمام مولفه های شبکه است. توجه شود که اگر مقدار شیپلی هیچ مولفه ای کمتر از این حد آستانه نبود، مولفه ای که کمترین مقدار شیپلی را دارد، حذف میشود.

برای فاز بهینه سازی، تعداد همسایه ها $b = 50$ و تعداد نسل ها $g = 250$ است. با افزایش این مقادیر، به خصوص g ، نتایج بهتری و زمان اجرای بیشتری کسب میکنیم.

ابتدا در مورد مسئله کلاسیک XOR کارایی روش پیشنهادی را بررسی میکنیم. شبکه با ۲ ورودی و ۵ نورون در لایه مخفی و ۱ خروجی کار خود را آغاز میکند. جدول ۱، چهار مرحله مورد نیاز برای ساده سازی این شبکه را نشان میدهد.

در هر مرحله، نورون با کمترین مشارکت در کارایی کل شبکه و طبیعتاً با کمترین مقدار شیپلی با پس زمینه خاکستری مشخص شده است. این مثال اجرایی نشان میدهد که تابع XOR به سادگی توسط یک شبکه عصبی با ۲ نورون در لایه مخفی اش میتواند یادگیری شود.

جدول ۱- مراحل ساده سازی برای مسئله XOR

مرحله ۱	نورون ۱: مقدار شیپلی = ۰/۲۴۸۸ نورون ۲: مقدار شیپلی = ۰/۰۰۰۰ نورون ۳: مقدار شیپلی = ۰/۰۴۱۶ نورون ۴: مقدار شیپلی = ۰/۱۷۲۴ نورون ۵: مقدار شیپلی = ۰/۲۳۲۹ حذف نورون ۲ دقت شبکه قبل از بهینه سازی: ۰/۹۵۹۹ آزمایشی دقت شبکه بعد از بهینه سازی: ۱/۰۰۰۰ آزمایشی
مرحله ۲	نورون ۱: مقدار شیپلی = ۰/۱۹۷۷ نورون ۳: مقدار شیپلی = ۰/۰۶۵۶ نورون ۴: مقدار شیپلی = ۰/۲۱۰۷ نورون ۵: مقدار شیپلی = ۰/۰۷۳۲ حذف نورون ۳ دقت شبکه قبل از بهینه سازی: ۰/۹۹۸۸ دقت شبکه بعد از بهینه سازی: ۱/۰۰۰۰
مرحله ۳	نورون ۱: مقدار شیپلی = ۰/۲۶۸۵ نورون ۴: مقدار شیپلی = ۰/۲۶۷۱ نورون ۵: مقدار شیپلی = ۰/۰۹۷۱ حذف نورون ۵ دقت شبکه قبل از بهینه سازی: ۰/۹۹۰۷ دقت شبکه بعد از بهینه سازی: ۱/۰۰۰۰
مرحله ۴	نورون ۱: مقدار شیپلی = ۰/۲۹۸۶ نورون ۴: مقدار شیپلی = ۰/۲۶۵۴ حذف نورون ۴ دقت شبکه قبل از بهینه سازی: ۰/۵۴۱۶ دقت شبکه بعد از بهینه سازی: ۰/۵۴۱۷

$p = 0.9$ مورد نیاز است. در این مثال، چون با یک مسئله رگرسیون روبرو هستیم، دقت بدست آمده دیگر به ۱ نمیرسد. اختلاف در زمان اجرا در این مورد بسیار زیاد است زیرا پس از هر مرحله ساده سازی، فاز بهینه سازی با زمان اجرای زیاد اجرا میشود.

جدول ۳- نتایج ساده سازی برای مسئله بهره‌وری انرژی

$\theta_s = 0$			$\theta_s = 0.1$			$\theta_s = 0.25$		
h	a_{train}	a_{test}	h	a_{train}	a_{test}	h	a_{train}	a_{test}
۲۰	۰/۸۵۴	۰/۸۴۸	۲۰	۰/۸۵۴	۰/۸۴۸	۲۰	۰/۸۵۴	۰/۸۴۸
۱۹	۰/۹۴۷	۰/۹۴۲	۱۰	۰/۹۴۲	۰/۹۳۸	۹	۰/۹۳۰	۰/۹۲۴
۱۸	۰/۹۴۷	۰/۹۳۷	۴	۰/۹۲۸	۰/۹۲۴	۵	۰/۹۲۳	۰/۹۱۹
۱۷	۰/۹۵۵	۰/۹۵۴						
۱۶	۰/۹۵۳	۰/۹۴۹						
۱۵	۰/۹۴۵	۰/۹۳۹						
۱۴	۰/۹۴۹	۰/۹۴۳						
۱۳	۰/۹۵۹	۰/۹۵۶						
۱۲	۰/۹۵	۰/۹۶۲						
۱۱	۰/۹۵۶	۰/۹۵۴						
۱۰	۰/۹۵۰	۰/۹۴۶						
۹	۰/۹۴۰	۰/۹۳۵						
۸	۰/۹۴۹	۰/۹۴۶						
۷	۰/۹۴۸	۰/۹۴۵						
۶	۰/۹۴۸	۰/۹۴۶						
۵	۰/۹۴۸	۰/۹۴۶						
۴	۰/۹۱۹	۰/۹۰۸						

برگشت به مرحله قبل و خاتمه الگوریتم، نوروں های باقیمانده: ۴، ۱

در ادامه، ما این روش را بر ۲ مسئله طبقه‌بندی و ۲ مسئله رگرسیون اعمال میکنیم. از آنجا که هدف اصلی هرس کردن یک شبکه عصبی، جدای از زمان پاسخ شبکه، کم کردن احتمال بیش‌برازش و افزایش قابلیت تعمیم^{۵۵} است، لذا ما کارایی شبکه را هم روی داده های آموزشی و هم آزمایشی ارزیابی میکنیم. در هر مسئله داده های کلی به صورت تصادفی به ۶۷٪ آموزشی و ۳۳٪ آزمایشی تقسیم میشود.

مسئله طبقه بندی *Iris* [۲۳] یکی از معروف ترین مسائل طبقه بندی مورد مطالعه است [۲۴]. مجموعه داده^{۵۶} شامل ۳ کلاس و هر کلاس ۵۰ نمونه و هر نمونه حاوی ۴ صفت عددی است. ابتدا با یک شبکه آموزش دیده با ۲۰ نوروں شروع میکنیم. جدول ۲ مراحل ساده سازی این شبکه را نشان می‌دهد. توجه شود h همان تعداد نوروں های مخفی، a_{train} و a_{test} دقت های محاسبه شده شبکه روی داده های آموزشی و آزمایشی است. هرچه مقدار آستانه θ_s افزایش پیدا کند، مراحل ساده سازی کمتر میشود. وقتی 0 است، h در هر مرحله کاهش پیدا میکند. وقتی بزرگ تر میشود ($\theta_s = 0.25$)، تعداد مراحل ساده سازی تقریباً نصف میشود در حالی که کارایی شبکه روی مجموعه آموزشی یکسان ارزیابی میشود و قابلیت تعمیم هم حتی بهتر هم میشود.

از آنجایی که مسئله *Iris* یک مسئله طبقه‌بندی است، e در رابطه (۲) مقدار درصد طبقه‌بندی درست نمونه‌ها را نشان میدهد (۱۰۰٪)، اما ضریب تعیین همچنان ۱ است، و در نتیجه دقت بدست آمده، تنها با وجود ۲ نوروں در لایه مخفی، یک تطابق کامل را برای مجموعه آموزشی نشان میدهد.

جدول ۲- نتایج ساده سازی برای مسئله Iris

$\theta_s = 0$			$\theta_s = 0.1$			$\theta_s = 0.25$		
h	a_{train}	a_{test}	h	a_{train}	a_{test}	h	a_{train}	a_{test}
۲۰	۰/۹۷۵	۰/۹۶۰	۲۰	۰/۹۷۵	۰/۹۶۰	۲۰	۰/۹۷۵	۰/۹۶۰
۱۹	۱/۰۰۰	۰/۸۹۴	۱۵	۱/۰۰۰	۰/۸۹۴	۱۲	۱/۰۰۰	۰/۸۹۴
۱۸	۱/۰۰۰	۰/۸۶۱	۱۳	۱/۰۰۰	۰/۸۵۲	۱۱	۱/۰۰۰	۰/۸۹۴
۱۷	۱/۰۰۰	۰/۸۷۸	۱۲	۱/۰۰۰	۰/۸۹۴	۱۰	۱/۰۰۰	۰/۸۷۸
۱۶	۱/۰۰۰	۰/۸۸۱	۱۱	۰/۹۴۳	۰/۸۳۶	۸	۰/۹۵۲	۰/۸۶۷
۱۵	۱/۰۰۰	۰/۸۹۴	۱۰	۱/۰۰۰	۰/۸۷۸	۶	۰/۹۵۲	۰/۸۵۳
۱۴	۱/۰۰۰	۰/۸۷۴	۹	۰/۹۲۴	۰/۷۷۲	۵	۰/۹۵۳	۰/۸۷۷
۱۳	۱/۰۰۰	۰/۸۷۸	۷	۰/۹۵۱	۰/۸۱۸	۴	۰/۹۵۵	۰/۸۵۸
۱۲	۱/۰۰۰	۰/۸۷۰	۶	۰/۹۵۸	۰/۸۱۲	۳	۰/۹۶۰	۰/۸۷۷
۱۱	۱/۰۰۰	۰/۸۷۹	۵	۰/۹۶۰	۰/۸۵۴	۲	۱/۰۰۰	۰/۸۹۴
۱۰	۱/۰۰۰	۰/۸۳۱	۴	۰/۹۶۳	۰/۸۰۵			
۹	۱/۰۰۰	۰/۹۲۸	۳	۰/۹۵۵	۰/۸۶۳			
۸	۱/۰۰۰	۰/۸۶۱						
۷	۱/۰۰۰	۰/۸۴۴						
۶	۱/۰۰۰	۰/۸۶۱						
۵	۱/۰۰۰	۰/۸۹۶						
۴	۱/۰۰۰	۰/۸۶۱						
۳	۱/۰۰۰	۰/۸۲۸						
۲	۱/۰۰۰	۰/۸۲۸						

مجموعه داده بهره‌وری انرژی^{۵۷} [۲۵] شامل ۷۶۸ نمونه و ۸ صفت ورودی است و هدف آن پیش‌بینی ۲ پاسخ با مقدار واقعی^{۵۸} است: بار گرمایشی^{۵۹} و بار سرمایشی^{۶۰}. جدول ۳ نشان می‌دهد که این مسئله از مسئله قبلی پیچیده تر است چون پس از ساده سازی شبکه، وجود ۴ نوروں برای بدست آوردن حداقل دقت

مسئله یون‌کره^{۶۱} [۲۶] از نوع مسئله های طبقه‌بندی دودویی است. این مجموعه-داده شامل ۳۵۱ نمونه با ۳۴ صفت است.

جدول ۴ نتایج بدست آمده در ساده سازی شبکه مورد نظر برای این مسئله را نشان میدهد. همان‌طور که مشاهده می‌شود، زیاد کردن مقدار θ_s نتیجه بدست آمده را بسیار بهبود می‌بخشد (هر تعداد نوروں مخفی را از ۷ به ۲ کاهش میدهد و هم دقت شبکه نهایی را بین ۳ حالت، ماکسیمم می‌کند).

جدول ۴- نتایج ساده سازی برای مسئله یون‌کره

$\theta_s = 0$			$\theta_s = 0.1$			$\theta_s = 0.25$		
h	a_{train}	a_{test}	h	a_{train}	a_{test}	h	a_{train}	a_{test}
۲۰	۰/۹۳۶	۰/۷۲۸	۲۰	۰/۹۳۶	۰/۷۲۸	۲۰	۰/۹۳۶	۰/۷۲۸
۱۹	۱/۰۰۰	۰/۷۱۲	۱۰	۱/۰۰۰	۰/۷۹۲	۷	۱/۰۰۰	۰/۷۳۳
۱۸	۱/۰۰۰	۰/۷۵۵	۸	۱/۰۰۰	۰/۸۶۸	۵	۱/۰۰۰	۰/۶۹۶
۱۷	۱/۰۰۰	۰/۷۵۵	۷	۱/۰۰۰	۰/۸۰۱	۴	۱/۰۰۰	۰/۶۸۹
۱۶	۱/۰۰۰	۰/۷۱۳	۶	۱/۰۰۰	۰/۷۶۱	۳	۰/۹۷۸	۰/۷۵۵
۱۵	۱/۰۰۰	۰/۷۰۶	۵	۱/۰۰۰	۰/۷۳۶	۲	۰/۹۶۷	۰/۷۷۷
۱۴	۱/۰۰۰	۰/۷۱۱	۴	۰/۹۷۸	۰/۷۷۹			
۱۳	۱/۰۰۰	۰/۶۸۹	۳	۰/۹۳۰	۰/۸۲۵			
۱۲	۱/۰۰۰	۰/۷۷۶	۲	۰/۹۴۶	۰/۸۴۸			
۱۱	۱/۰۰۰	۰/۷۱۲						
۱۰	۱/۰۰۰	۰/۷۱۵						
۹	۱/۰۰۰	۰/۶۹۶						
۸	۰/۹۴۷	۰/۷۷۸						
۷	۰/۹۴۷	۰/۷۵۷						

مجموعه داده هیدرودینامیک قایق بادبانی^{۶۲} [۲۷] یک مسئله رگرسیون با ۶ صفت ورودی است. جدول ۵ نتایج این مورد را نشان می دهد. همان طور که مشاهده میشود تمام θ_s ها منجر به دقت و پیچیدگی شبکه یکسان می شوند.

نتیجه می گیریم که این مسئله با ۱ نورون مخفی حل می شود.

جدول ۵- نتایج ساده سازی برای مسئله هیدرودینامیک قایق بادبانی

$\theta_s = 0$			$\theta_s = 0.1$			$\theta_s = 0.25$		
h	a_{train}	a_{test}	h	a_{train}	a_{test}	h	a_{train}	a_{test}
۲۰	۰/۹۳۶	۰/۷۲۸	۲۰	۰/۹۳۶	۰/۷۲۸	۲۰	۰/۹۳۶	۰/۷۲۸
۱۹	۱/۰۰۰	۰/۷۱۲	۱۰	۱/۰۰۰	۰/۷۹۲	۷	۱/۰۰۰	۰/۷۳۳
۱۸	۱/۰۰۰	۰/۷۵۵	۸	۱/۰۰۰	۰/۸۶۸	۵	۱/۰۰۰	۰/۶۹۶
۱۷	۱/۰۰۰	۰/۷۵۵	۷	۱/۰۰۰	۰/۸۰۱	۴	۱/۰۰۰	۰/۶۸۹
۱۶	۱/۰۰۰	۰/۷۱۳	۶	۱/۰۰۰	۰/۷۶۱	۳	۰/۹۷۸	۰/۷۵۵
۱۵	۱/۰۰۰	۰/۷۰۶	۵	۱/۰۰۰	۰/۷۳۶	۲	۰/۹۶۷	۰/۷۷۷
۱۴	۱/۰۰۰	۰/۷۱۱	۴	۰/۹۷۸	۰/۷۷۹			
۱۳	۱/۰۰۰	۰/۶۸۹	۳	۰/۹۳۰	۰/۸۲۵			
۱۲	۱/۰۰۰	۰/۷۷۶	۲	۰/۹۴۶	۰/۸۴۸			
۱۱	۱/۰۰۰	۰/۷۱۲						
۱۰	۱/۰۰۰	۰/۷۱۵						
۹	۱/۰۰۰	۰/۶۹۶						
۸	۰/۹۴۷	۰/۷۷۸						
۷	۰/۹۴۷	۰/۷۵۷						

که در رابطه قبل، $\omega_S = |S|! (n - |S| - 1)!$ و همچنین قسمت دوم رابطه (۵) توسط تعداد Γ جایگشت تصادفی Π^R تقریب زده میشود:

$$\varphi_v^{\Pi^R}(i) = \frac{1}{r} \sum_{\pi \in \Pi^R} (v(P_i^\pi \cup \{i\}) - v(P_i^\pi)) \quad (7)$$

حال به طراحی بازی میپردازیم. ایده اینکه به هر بازیگر یک مقدار نسبت دهیم و تابعی داشته باشیم که به هر ائتلاف که مجموعه ای از بازیگر ها هستند، یک مقدار سود نسبت دهد، میتواند روی شبکه های عصبی پیاده شود. برای اینکار ابتدا باید مجموعه بازیگر های U و تابع سودمندی V تعریف شود.

مجموعه بازیگر ها که همان مولفه های شبکه عصبی مصنوعی هستند، در قسمت قبل توضیح داده شد و دیگر در این قسمت آن ها را توضیح نمیدهیم. برای تابع سودمندی این قسمت، هر معیار ارزیابی یا مقدار خطای شبکه میتواند مورد استفاده قرار گیرد. مقادیر خطا مثل خطای آموزش^{۶۵} محدود به یک بازه نیست که برای آنالیز های آینده این روش، مناسب نیست. معیار ارزیابی ای مثل دقت، محدود است و ما در این روش پیشنهادی، از دقت آنتروپی متقابل^{۶۶} استفاده میکنیم. معیار دقت به عنوان معیاری برای ارزیابی یک شبکه عصبی مصنوعی، برای ساخت مقادیر سودمندی در یک بازی ائتلافی، مورد استفاده قرار میگیرد.

برای یک معیار ارزیابی m مانند دقت، مقدار سودمندی از رابطه زیر بدست می آید:

$$v(S) = m(S) - m(\emptyset), \quad (8)$$

که در این رابطه $S \subseteq U$ و $m(T)$ یعنی معیار ارزیابی شبکه ای که فقط بازیگر های درون T در آن حضور دارند. معمولا $m(\emptyset)$ حداقل به اندازه مقدار مورد انتظار نایو^{۶۷} در مسائل طبقه بندی یا رگرسیون است. بنابراین میتوان آن را بزرگ تر از صفر در نظر گرفت. یعنی برای مثال برای یک مسئله طبقه بندی با k کلاس، داریم $m(\emptyset) > \frac{1}{k}$ که یعنی مدل ارزیابی شده باید بهتر از حدس زدن تصادفی^{۶۸} باشد.

این تعریف میتواند مقادیر منفی هم تولید کند. در حقیقت به جای اینکه تابع سودمندی در بازه $[0,1]$ باشد، در بازه $[-1,1]$ است [۲۸]. طبق چیزی که در [۲] گفته شده است: " مفهوم علامت مقدار سودمندی واضح است، برای مقادیر مثبت، معیار مربوطه باید به عنوان سود در نظر گرفته شود و برای مقادیر منفی به عنوان هزینه ".

در این بخش ما ۳ استراتژی مختلف را برای هرس کردن شبکه و بدست آوردن شبکه کوچک تر (رویکرد بالا-پایین) بررسی میکنیم: تصادفی محور^{۶۹}، وزن محور^{۷۰} و مبتنی بر مقدار شیپلی. خلاصه ای از این ۳ روش در جدول ۶ نشان داده شده است. توجه شود استراتژی هرس کردنی که در روش پیشنهادی اول گفته شد، تغییر یافته ی روش $SVbottom(p)$ است.

استراتژی هایی که تعداد ثابتی بازیگر را برای هرس کردن در یک مرحله، انتخاب میکنند را استراتژی های غیرپویا میگوییم که اغلب بازیگر های کمی را در مراحل اولیه و بازیگر های زیادی را در مراحل آخر هرس میکنند.

۴- روش پیشنهادی دوم

در این روش، ادبیات متفاوتی را نسبت به روش اول بیان میکنیم. ابتدا مقدار شیپلی را برای هر مولفه (بازیگر در بازی ائتلافی) تعریف میکنیم.

اگر U مجموعه ای از n بازیگر (مولفه شبکه) باشد و $\mathcal{P}(U)$ مجموعه قدرت^{۶۳} آن باشد، و اگر $v: \mathcal{P}(U) \rightarrow \mathbb{R}$ تابع سودمندی باشد که سود $v(S)$ را به هر زیرمجموعه $S \subseteq U$ از بازیگر ها بدهد، مقدار شیپلی برای بازیگر i میشود:

$$\begin{aligned} \varphi_v(i) &= \frac{1}{n!} \sum_{S \subseteq U \setminus \{i\}} (|S|! (n - |S| - 1)!) \cdot (v(S) - v(S - i)) \\ &= \frac{1}{|U|!} \sum_{\pi \in \Pi} (v(P_i^\pi \cup \{i\}) - v(P_i^\pi)), \end{aligned} \quad (5)$$

که Π مجموعه تمام جایگشت های U و $P_i^\pi = \{j \in U : \pi(j) < \pi(i)\}$ است. بخاطر پیچیدگی محاسباتی نمایی رابطه بالا، مقدار شیپلی در قسمت اول رابطه (۵) میتواند توسط روش مونت کارلو^{۶۴} با زیرمجموعه های تصادفی $R \subset U$ تقریب زده شود:

$$\varphi_v^R(i) = \frac{1}{\sum_{S \in R} \omega_S} \sum_{S \in R} \omega_S \cdot (v(S) - v(S - i)), \quad (6)$$

نرم وزن از رابطه زیر بدست می آید:

که در این رابطه E_n^{in} یعنی مجموعه اتصال های ورودی به نورون n و $(W_{j,n})$ یعنی وزن اتصال های از نورون j به نورون n .

۱-۴- مطالعات موردی روش دوم

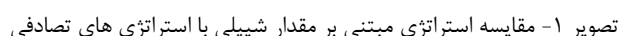
۱- هرس $MNIST$: به صورت یک رویکرد بالا-پایین متوالی مبتنی بر استراتژی های گوناگون از جمله استراتژی مبتنی بر مقدار شیپلی، مدال میشود.

۲- ارزیابی هرس: ارزیابی هرس مبتنی بر مقدار شیپلی است، به این صورت که نتایج بدست آمده را با نتایج بدست آمده از انتخاب تصادفی از مدل های مبتنی بر جستجوی توری مقایسه میکند.

۳- هرس *20newsgroup*: هرس مدل های بزرگتر *20newsgroups* برای مقایسه با بینش^{۷۲} های قبلی و اثبات مقیاس^{۷۳} است.

تصاویر ۱ و ۲ مراحل هرس بر مبنای استراتژی های مبتنی بر مقدار شیپلی را در مقایسه با استراتژی های دیگر را نشان می دهد. توجه شود در این تصاویر، تعداد دقیق مراحل یک استراتژی مشخص نیست. مراحل دقیق آن را جدول ۷ می توانید مشاهده کنید. تقریباً تمام استراتژی ها برای ۱۲ بازیگر یا بیشتر، نتایج یکسانی تولید کرده اند.

تصویر ۱ مقایسه استراتژی های $SVbucket(0.2)$ ، $random(1)$ و $random(3)$ ، نشان میدهد. تصویر ۲ هم مقایسه $SVbucket(0.2)$ با $Wbottom(1)$ و $Wbottom(3)$ نشان میدهد.



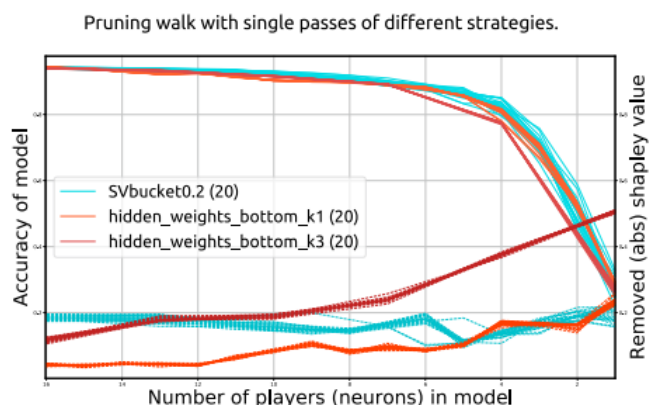
۶/۹۵	میانگین تعداد نورون برای حفظ دقت
۸	بیشترین تعداد نورون برای حفظ دقت
۶	کمترین تعداد نورون برای حفظ دقت

استراتژی SVbucket0.2

۱۲/۰۵	میانگین تعداد قدم
۱۳	بیشترین تعداد قدم
۱۲	کمترین تعداد قدم
۴۴/۱	میانگین تعداد دوره
۴۶	بیشترین تعداد دوره
۴۴	کمترین تعداد دوره
۸/۰۵	میانگین تعداد نورون برای حفظ دقت
۱۰	بیشترین تعداد نورون برای حفظ دقت
۷	کمترین تعداد نورون برای حفظ دقت

استراتژی SVbucket0.1

۱۹/۶۵	میانگین تعداد قدم
۲۱	بیشترین تعداد قدم
۱۹	کمترین تعداد قدم
۵۹/۳	میانگین تعداد دوره
۶۲	بیشترین تعداد دوره
۵۸	کمترین تعداد دوره
۶/۹۵	میانگین تعداد نورون برای حفظ دقت
۸	بیشترین تعداد نورون برای حفظ دقت
۶	کمترین تعداد نورون برای حفظ دقت



تصویر ۲- مقایسه استراتژی مبتنی بر مقدار شیلی با استراتژی های مبتنی بر وزن

۴-۱-۲- آزمایش MNIST در مقایسه با جستجوی توری

برای دو ابرپارامتر^{۷۶} : تعداد دوره های آموزش و تعداد بازیگر ها ، یک جستجوی توری تعریف و ۲۰۰ مرتبه تکرار میشود. هر خروجی آن ها، مقدار دقتی است که از شبکه تغذیه رو به جلو MNIST ای که به تعداد دوره و با تعداد بازیگر گفته شده آموزش دیده است، بدست می آید.

با مقایسه نتایج بدست آمده از جستجوی توری با نتایج بدست آمده از هرس MNIST با استراتژی های گوناگون گفته شده، مشخص است که هرس های بالا- پایین مبتنی بر مقدار شیلی، بهینه تر هستند البته در صورتی که مقادیر شیلی را بتوانیم در پیچیدگی زمانی خطی، بدست آوریم.

۴-۱-۳- آزمایش هرس مبتنی بر 20newsgroups

این آزمایش مثل آزمایش اول است با این تفاوت که مبتنی بر مجموعه داده 20newsgroups است. البته معیار های ارزیابی مستقیماً قابل مقایسه با معیار های آزمایش اول نیستند اما نتایج آماری همه استراتژی های آزمایش اول میتواند برای مقایسه با این آزمایش، بازسازی شوند.

جدول ۷- نتایج بدست آمده از ۳ آزمایش گفته شده

استراتژی SVbucket0.1

۱۹/۶۵	میانگین تعداد قدم
۲۱	بیشترین تعداد قدم
۱۹	کمترین تعداد قدم
۵۹/۳	میانگین تعداد دوره
۶۲	بیشترین تعداد دوره
۵۸	کمترین تعداد دوره

۵- ایده برای پژوهش های آینده

۷- مراجع

۱-۵- جنبه تئوری

- [1] Hochreiter, S., Schmidhuber, J., 1997. Long short- term memory. *Neural computation* 9, 1735-1780.
- [2] LeCun, Y., Bengio, Y., et al., 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 1995.
- [3] Srivastava, R.K., Greff, K., Schmidhuber, J., 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.
- [4] Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13, 281-305.

- (۱) کاهش مرتبه پیچیدگی زمانی محاسبه مقادیر شیپلی.
- (۲) مقدار شیپلی کلاسیک به ما نشان میداد که تمام بازیگر های بازی میتوانستند یک ائتلاف معنی دار تشکیل دهند. اما در شبکه های عصبی، هر ائتلافی لزوماً معنی ندارد. لذا بازی به جای فرم کلی یک بازی ائتلافی، باید مخصوص شبکه های عصبی تشکیل شود.

۲-۵- جنبه الگوریتمی

- [5] R. Setiono, "Feedforward Neural Network Construction Using Cross Validation", *Journal of Neural Computation*, vol. 13, issue 12, MIT Press Cambridge, MA, USA, 2001, pp. 2865-2877.
- [6] M. Frea, "The upstart algorithm: A method for constructing and training feed-forward neural networks", *NeuralComputation*, no. 2, 1990, pp. 198-209.
- [7] S. Fahlman, C. Lebiere, "The cascade correlation architecture", *Advances in Neural Information Processing Systems*, no. 2, Morgan Kaufman, San Mateo, CA, 1990, pp. 524-532.
- [8] I. C. G. Campbell, "Constructive Learning Techniques for Designing Neural Network Systems", in C. T. Leondes (ed.), *Constructive learning techniques for designing neural network systems*, Academic Press, 1998, pp. 91-145.
- [9] Y. Le Cun, J. S. Denker, S. A.olla, "Optimal Brain Damage", *Advances in Neural Information Processing Systems*, Morgan Kaufmann, 1990, pp. 598-605.
- [10] B. Hassibi, D. G. Stork, "Second Order Derivatives for Network Pruning: Optimal Brain Surgeon", in *Advances in Neural Information Processing Systems*, no. 5, S. J. Hanson, J. D. Cowan, C. L. Giles (eds.), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992, pp. 164-171.
- [11] M. C. Mozer, P. Smolensky, "Skeletonization: a technique for trimming the fat from a network via relevance assessment", in *Advances in neural information processing systems*, no. 1, D. S. Touretzky (ed.), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 107-115.
- [12] Z. Ganon, A. Keinan, E. Ruppin, "Neurocontroller Analysis via Evolutionary Network Minimization", *Artificial Life*, vol. 12, no. 3, 2006, pp. 435-448.
- [13] S. Cohen, E. Ruppin, G. Dror, "Feature selection based on the Shapley value", *Proceedings of the 19th international joint conference on Artificial intelligence (IJCAI'05)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005, pp. 665-670.
- [14] S. Cohen, G. Dror, E. Ruppin, "Feature Selection via Coalitional Game Theory", *Neural Computation*, vol. 19, no. 7, 2007, pp. 1939-1961.
- [15] Schuster, A., Yamaguchi, Y., 2010. Application of game theory to neuronal networks. *Advances in Artificial Intelligence* 2010, 2.
- [16] L. Shapley, "A value for n-person games", in *Contributions to the Theory of Games*, vol. 2, H. Kuhn, A. Tucker (eds.), Princeton

- (۱) انواع دیگر رویکرد های بالا-پایین یا پایین-بالا یا ترکیبی میتوانند با مفهوم مقدار شیپلی الگوریتم های جدیدی ارائه دهند.
- (۲) در روش های مبتنی بر هیوریستیک، الگوریتم شبیه سازی ذوب فلزات^{۷۷} به ما نشان میدهد که قدم های حریصانه ای که برای بهینه سازی شبکه برمی داریم، ممکن است منجر به گیر افتادن در نقطه بهینه محلی شود. لذا ترکیب هرس تصادفی و مقدار شیپلی، میتواند الگوریتم های جدیدی مبتنی بر هیوریستیک ایجاد کند.

۶- نتیجه گیری

مقدار شیپلی به عنوان یک مفهوم مبتنی بر نظریه بازی ها در بازی های ائتلافی، میتواند بر روی شبکه های عصبی مصنوعی اعمال شود به این صورت که مولفه های شبکه مثل نوروں ها را به عنوان بازیگر های این بازی در نظر بگیریم.

ایده اصلی این است که در دو مرحله شبکه عصبی را بهینه کنیم، اول اینکه توسط رویکرد بازی روی توپولوژی، مقادیر شیپلی مولفه های شبکه را بدست آوریم و در مرحله بعد مولفه های با مقدار شیپلی کوچک تر که نشان دهنده ی مولفه های با مشارکت کمتر و اهمیت کمتر در شبکه هستند را هرس کنیم.

با انجام این کار میتوانیم شبکه ای جدید داشته باشیم که دقت لازم را حفظ کرده و در عین حال بخاطر کم شدن مولفه های غیرضروری، ساده تر شده است.

همچنین نتیجه میگیریم که بدست آوردن مقدار شیپلی به صورت دقیق یک مسئله سخت و به عبارتی آن پی کامل است و ما میتوانیم به روش های مختلفی آن را تقریب بزنیم و از آن ها استفاده کنیم.

در این مقاله نشان دادیم که پس از اعمال هرس بر روی شبکه، میتوانیم با روش های مختلفی شبکه را برای رسیدن به دقت بیشتر، بهینه کنیم.

در آخر هم با ارائه دو روش مختلف، نشان دادیم که مقدار شیپلی میتواند در مسائل مختلف با ترکیب با روش های گوناگونی، به ما در ساده سازی یک شبکه عصبی مصنوعی، کمک کند.

- 5- Shapley value
 - 6- Coalitional games
 - 7- Neuron
 - 8- Top-down method
 - 9- Topology
 - 10- Pruning
 - 11- NP-Complete
 - 12- Train
 - 13- Test
 - 14- Time complexity
 - 15- Hidden layer
 - 16- Overfitting
 - 17- Down-top method
 - 18- Weights
 - 19- Grid search
 - 20- Low contributing
 - 21- Cooperative games
 - 22- Sampling
 - 23- Single Strategy
 - 24- Multi strategy
 - 25- Cross-validation
 - 26- Precision
 - 27- Upstart
 - 28- Classification networks
 - 29- Cascade correlation
 - 30- Hidden nodes
 - 31- Optimal brain damage
 - 32- Information theory
 - 33- Second derivation information
 - 34- optimal brain surgeon
 - 35- Magnitude value
 - 36- Skeletonization
 - 37- Evolutionary network minimization
 - 38- Fully recurrent neuro-controllers
 - 39- Feature selection
- University Press, Princeton, NJ, 1953.
- [17] A. E. Roth, "Introduction to the Shapley value", in *The Shapley value*, A. E. Roth (ed.), University of Cambridge Press, Cambridge, 1988, pp. 1-27.
- [18] S. Fatima, M. Wooldridge, N. R. Jennings, "An anytime approximation method for the inverse Shapley value problem", Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008, vol. 2, International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 935-942.
- [19] V. Conitzer, T. Sandholm, "Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains", Proceedings of the 19th National Conference on Artificial Intelligence, AAAI 2004, AAAI Press, 2004, pp. 219-225.
- [20] A. E. Bryson, Y. C. Ho, *Applied Optimal Control*, Blaisdell, New York, 1969.
- [21] P. J. Werbos, "Beyond regression: new tools for prediction and analysis in the behavioral sciences", PhD Thesis, Harvard University, Cambridge, MA, 1974.
- [22] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning internal representations by error propagation", in D. E. Rumelhart, J. L. McClelland (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, The MIT Press, Cambridge, MA, 1986, pp. 318-362.
- [23] R. A. Fisher, "The use of multiple measurements in taxonomic problems", *Annual Eugenics*, vol. 7, part II, 1936, pp. 179-188.
- [24] R. O. Duda, P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973, p. 218.
- [25] A. Tsanas, A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools", *Energy and Buildings*, vol. 49, 2012, pp. 560-567.
- [26] V. G. Sigillito, S. P. Wing, L. V. Hutton, K. B. Baker, "Classification of radar returns from the ionosphere using neural networks", *John Hopkins APL Technical Digest*, vol. 10, 1989, pp. 262-266.
- [27] I. Ortigosa, R. Lopez, J. Garcia, "A neural networks approach to residuary resistance of sailing yachts prediction", Proceedings of the International Conference on Marine Engineering MARINE 2007, 2007.
- [28] Cardin, M., Giove, S., et al., 2007. On non-monotonic Choquet integrals as aggregation functions. Technical Report.
- [29] Fatima, S.S., Wooldridge, M., Jennings, N.R., 2007. A randomized method for the shapley value for the voting game, in: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, ACM. P. 157.
- ۸- واژه نامه
- 1- Artificial neural networks
 - 2- Fully connected networks
 - 3- Adversarial data
 - 4- Heuristic

75- Epoch

76- Hyperparameter

77- Simulated annealing

40- Contribution-selection

41- Multi-perturbation

42- Forward elimination

43- Backward elimination

44- Joint decision

45- transferable utility

46- Marginal

47- Coefficient of determination

48- NMRSE: normalized root mean square error

49- Evolutionary hill climbing

50- Mutation

51- Connection Weights

52- Backward propagation

53- Activation function

54- Hyperbolic tangent

55- Generalization capability

56- Dataset

57- Energy efficiency

58- Real-value

59- Hitting load

60- Cooling load

61- Ionosphere

62- Yacht hydrodynamics

63- Powerset

64- Monte carlo

65- Training error

66- Cross-entropy

67- Naïve expectation

68- Random guessing

69- Random-based

70- Weight-based

71- Norm

72- Insight

73- Proof of scale

74- Forward-feeding