

تشخیص بیماری سرطان سینه با استفاده از شبکه عصبی پرسپترون چند لایه مبتنی بر ماشین با قابلیت یادگیری فوق‌العاده

عاطفه سرلک^{1*}، سید مهدی هاشمی²، زهرا عصارزاده³

1- دانشجوی کارشناسی ارشد مهندسی برق-کنترل مؤسسه آموزش عالی پیام گلپایگان

2- عضو هیئت علمی مؤسسه آموزش عالی پیام گلپایگان

3- مدرس مؤسسه آموزش عالی پیام گلپایگان

چکیده

سرطان سینه یکی از شایعترین انواع سرطان است که در مراحل اولیه بیماری، قابل درمان است. تشخیص زود هنگام و درمان مؤثر، تنها راه نجات و کاهش مرگ و میر ناشی از سرطان سینه است. طبقه‌بندی دقیق از تومور سرطان سینه، یک کار مهم در تشخیص پزشکی است. در حوزه یادگیری ماشین، روش‌های بسیاری برای طبقه‌بندی ارائه شده است. در این مقاله از شبکه عصبی پرسپترون چند لایه مبتنی بر ماشین با قابلیت یادگیری فوق‌العاده استفاده شده است. این روش نسبت به الگوریتم‌های مبتنی بر گرادینان، الگوریتمی سریعتر، با دقت و قابلیت تعمیم فوق‌العاده در طبقه‌بندی می‌باشد. در این الگوریتم، پارامترهای لایه پنهان به صورت تصادفی انتخاب می‌شوند و وزن‌های لایه خروجی به صورت تحلیلی از یک سیستم خطی و با استفاده از معکوس تعمیم‌یافته مور پنروز به دست می‌آید. نتایج شبیه‌سازی‌ها از برتری قابل توجه الگوریتم پیشنهادی نسبت به الگوریتم پس انتشار خطا از نظر سرعت یادگیری و قابلیت تعمیم حکایت می‌نماید.

کلمات کلیدی: شبکه عصبی، ماشین با قابلیت یادگیری فوق‌العاده، طبقه بندی

1. مقدمه

یکی از روش‌هایی که به صورت گسترده برای حل مسئله طبقه‌بندی² مورد استفاده قرار می‌گیرد، الگوریتم پس انتشار خطا (BP)³ می‌باشد. الگوریتم BP، یکی از رایج‌ترین الگوریتم‌ها جهت آموزش شبکه‌های عصبی چند لایه پیش‌خور می‌باشد.

عمومیت یافتن الگوریتم BP، بخاطر سادگی و کاربردهای موفقیت آمیزش در حل مسائل فنی-مهندسی می‌باشد.

* atefehsarlak66@gmail.com

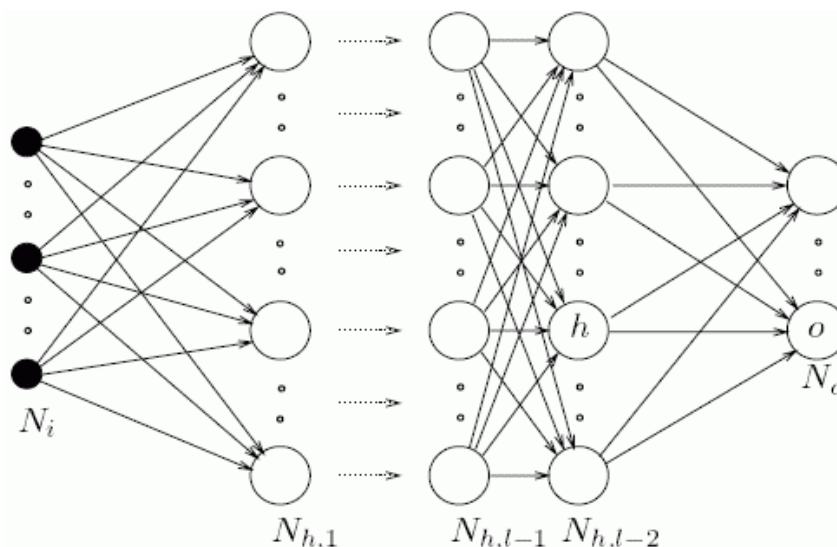
²- classification

³- Back-Propagation (BP)

علیرغم، موفقیت‌های کلی الگوریتم BP در یادگیری شبکه‌های عصبی چندلایه پیش‌خور، این الگوریتم ممکن است، به نقاط مینیمم محلی در فضای پارامتر، همگرا شود. بنابراین زمانی که الگوریتم BP همگرا می‌شود، نمی‌توان مطمئن شد که به یک جواب بهینه رسیده باشیم. و از طرفی سرعت همگرایی الگوریتم BP، خیلی کند است. از این گذشته، همگرایی الگوریتم BP، به انتخاب مقادیر اولیه وزن‌های شبکه، بردارهای بایاس و پارامترها موجود در الگوریتم، مانند نرخ یادگیری، وابسته است و گاهی ممکن است برای دستیابی به عملکردی بهتر، به تعداد زیاد تکرار مراحل یادگیری نیاز باشد. شبکه‌های عصبی مبتنی بر ماشین با قابلیت یادگیری فوق‌العاده (ELM)⁴، بر خلاف الگوریتم BP، نیازی به تنظیم پارامترهای لایه پنهان (وزن و بایاس) ندارد و این پارامترها به‌طور تصادفی انتخاب می‌شوند. هدف در این روش رسیدن به کوچک‌ترین خطای آموزش و کوچک‌ترین نرم وزن خروجی است. این الگوریتم، عملکرد کلی خوبی را نتیجه می‌دهد و بسیار سریع اجرا می‌شود و مشکل کمینه محلی ندارد و حداقل مداخله انسانی را نسبت به الگوریتم BP دارد. ساختار ادامه مقاله به شرح زیر است: در بخش دوم، ساختار شبکه‌های عصبی پیش‌خور و مدل ریاضی آن‌ها بیان می‌شود، در بخش سوم الگوریتم ELM و نتایج حاصل از شبیه‌سازی با الگوریتم BP و ELM بیان می‌شود و در بخش پایانی نتیجه‌گیری و پیشنهادات ارائه می‌شود.

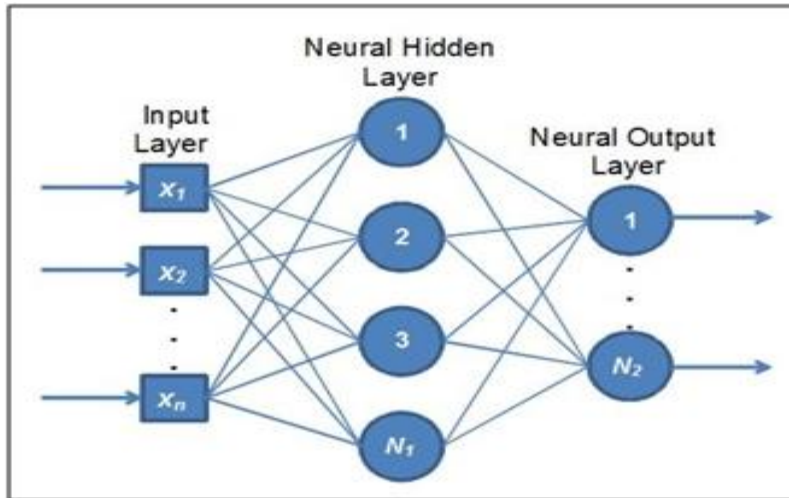
2. شبکه عصبی پیش‌خور و مدل ریاضی آن

یک شبکه عصبی پیش‌خور شامل یک لایه ورودی در یافت محرک از محیط خارجی، یک یا چند لایه مخفی و یک لایه خروجی جهت ارسال خروجی شبکه به محیط می‌باشد. از شبکه‌های عصبی پیش‌خور به‌طور وسیعی در زمینه‌های متنوعی از قبیل طبقه‌بندی الگوها، پردازش تصویر، تقریب توابع و ... استفاده شده است. در شکل (1)، یک شبکه عصبی پیش‌خور و در شکل (2) یک شبکه‌ی عصبی پیش‌خور با یک لایه مخفی نشان داده شده است.



شکل 1- ساختار یک شبکه عصبی پیش‌خور

⁴-Extreme Learning Machine (ELM)



شکل ۲- شبکه عصبی پیش‌خور با یک لایه مخفی

یک شبکه عصبی پیش‌خور با یک لایه مخفی (SLFN) با \tilde{N} نرون در لایه مخفی و تابع تحریک $g(x)$ ، برای N نمونه متمایز دلخواه (x_i, t_i) که در آن $x_i \in \mathbb{R}^n$ ، $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ و $t_i \in \mathbb{R}^m$ و $t_i = \{t_{i1}, t_{i2}, \dots, t_{in}\}$ به شکل رابطه (1) دارد:

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(x_i) = \sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j, \quad j=1, 2, \dots, N \quad (1)$$

که در آن $w_i = \{w_{i1}, w_{i2}, \dots, w_{in}\}$ و برابر بردار وزن متصل به i امین گره پنهان و گره ورودی است و $\beta_i = \{\beta_{i1}, \beta_{i2}, \dots, \beta_{im}\}$ بردار وزن متصل به i امین گره پنهان و گره خروجی می‌باشد و b_i بایاس i امین گره پنهان و $w_i \cdot x_j$ نشان‌دهنده ضرب داخلی بین w_i و x_j می‌باشد. تابع فعال‌سازی گره خروجی به صورت خطی در نظر گرفته شده است و SLFN با \tilde{N} نرون مخفی با تابع فعال ساز $g(x)$ می‌تواند N مشاهدات مجزا را با خطای صفر یاد بگیرد [1، 2، 3] یعنی:

$$\sum_{j=1}^{\tilde{N}} \|o_j - t_j\| = 0 \quad (2)$$

N معادلات بالا را می‌توان به صورت فشرده به صورت زیر بیان نمود:

$$H\beta = T \quad (3)$$

که در آن

$$H(w_1, \dots, w_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \quad (4)$$

و داریم:

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}$$

H ماتریس خروجی لایه پنهان است و هر ستون از ماتریس H ، خروجی i امین گرهی پنهان با توجه به ورودی‌های x_1, x_2, \dots, x_N است [4].

اگر تابع فعال ساز $g(x)$ بینهایت بار مشتق پذیر باشد، می‌توان ثابت کرد که تعداد گره‌های لایه پنهان مورد نیاز $\tilde{N} \leq N$ می‌باشد [4]. داریم:

قضیه 1-2: یک $SLFN$ استاندارد با N گره مخفی و تابع فعال ساز $g: R \rightarrow R$ که در هر بازه ای بی نهایت مشتق پذیر است، برای N نمونه متمایز دلخواه (x_i, t_i) که $x_i \in R^n$ و $t_i \in R^m$ برای هر w_i و b_i که به صورت تصادفی در هر بازه‌ای از R^n و R ، با توجه به تابع توزیع احتمال پیوسته با احتمال یک، ماتریس خروجی لایه پنهان H معکوس پذیر است [5] و داریم $\|H\beta - T\| = 0$

قضیه 2-2: با توجه به هر مقدار کوچک و $\epsilon > 0$ تابع تحریک $g: R \rightarrow R$ که بی نهایت بار در هر بازه‌ای مشتق پذیر است $\tilde{N} \leq N$ وجود دارد به طوری که برای N نمونه مجزای (x_i, t_i) که $x_i \in R^n$ و $t_i \in R^m$ برای هر w_i و b_i که به صورت تصادفی در هر بازه‌ای از R^n و R ، با توجه به تابع توزیع احتمال پیوسته با احتمال یک، انتخاب می‌شوند، داریم [5]:

$$\|H_N^* \tilde{N} \beta_{\tilde{N}^* m} - T_{N^* m}\| < \epsilon \quad (5)$$

روش‌های متفاوتی برای آموزش شبکه‌های عصبی پیش‌خور وجود دارد. از قانون یادگیری پس انتشار خطا (BP)، برای آموزش شبکه‌های عصبی چند لایه پیش‌خور که عموماً شبکه‌های چند لایه پرسپترون (MLP) هم نامیده می‌شود، استفاده می‌کنند. به عبارتی توپولوژی شبکه‌های چند لایه پرسپترون (MLP)⁵، با قانون یادگیری پس انتشار خطا تکمیل می‌شود. قانون یادگیری پس انتشار خطا در سال 1974 توسط وریاس مطرح شد که یک شبکه پرسپترون چند لایه البته با قوانین نیرومند آموزشی بود. در این الگوریتم، هدف، پیدا کردن مقادیری از w_i و b_i و β است به طوری که:

$$\|H(w_1, \dots, w_{\tilde{N}}, b_1, \dots, b_{\tilde{N}})\beta - T\| = \min \|H(w_1, \dots, w_{\tilde{N}}, b_1, \dots, b_{\tilde{N}})\beta - T\| \quad (6)$$

که این معادل حداقل کردن تابع هزینه یا شاخص عملکردی زیر است:

$$E = \sum_{j=1}^N \left(\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) - t_j \right)^2 \quad (7)$$

شاخص عملکردی تابعی از پارامترهای شبکه است.

به‌طور خلاصه، فرایند پس انتشار خطا از دو مسیر اصلی تشکیل می‌شود. 1- مسیر رفت که در این مسیر، یک الگوی آموزشی به شبکه اعمال می‌شود و تأثیرات آن از طریق لایه‌های میانی به لایه خروجی انتشار می‌یابد تا اینکه نهایتاً خروجی واقعی شبکه MLP به دست می‌آید. در این مسیر، پارامترهای شبکه (ماتریس‌های وزن و بردارهای بایاس)، ثابت و بدون تغییر در نظر گرفته می‌شوند و 2- مسیر برگشت، در این مسیر، برعکس مسیر رفت، پارامترهای شبکه MLP تغییر و تنظیم می‌گردند. این تنظیمات بر اساس قانون یادگیری اصلاح خطا انجام می‌گیرد. سیگنال خطا، در لایه خروجی شبکه تشکیل می‌گردد. بردار خطا برابر با اختلاف بین پاسخ مطلوب و پاسخ واقعی شبکه می‌باشد. مقدار خطا، پس از محاسبه، در مسیر برگشت از لایه خروجی و از طریق لایه‌های شبکه به سمت پاسخ مطلوب حرکت کند.

⁵- Multi-Layer Perceptron

در شبکه‌های MLP ، هر نرون دارای یک تابع تحریک غیرخطی است که از ویژگی مشتق‌پذیری برخوردار است. در این حالت، ارتباط بین پارامترهای شبکه و سیگنال خطا، کاملاً پیچیده و غیرخطی می‌باشد، بنابراین مشتقات جزئی نسبت به پارامترهای شبکه به راحتی قابل محاسبه نیستند. جهت محاسبه مشتقات از قانون زنجیره ای معمول در جبر استفاده می‌شود. تنظیم پارامترهای شبکه، مطابق با سیگنال‌های خطا که بر اساس ارائه هر الگو به شبکه محاسبه می‌شود، صورت می‌گیرد.

پارامترهای وزن و بایاس با رابطه (8) و (9) تنظیم می‌شوند:

$$W_{ji}^L(K+1) = W_{ji}^L(K) - \alpha \frac{\delta F}{\delta W_{ji}^L(K)} \quad (8)$$

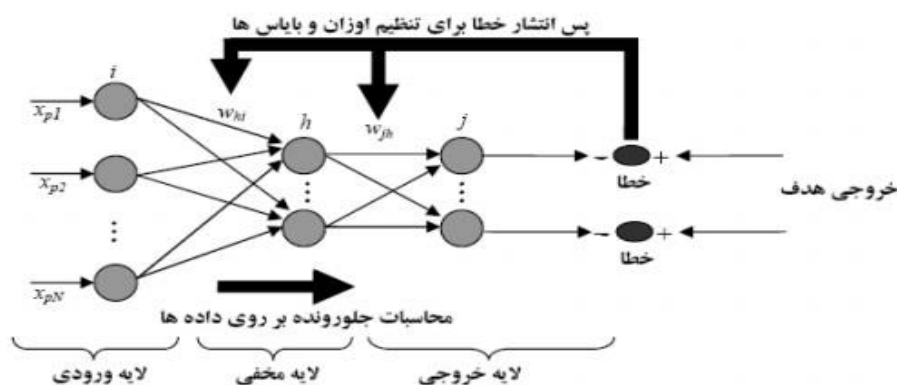
$$b_j^L(K+1) = b_j^L(K) - \alpha \frac{\delta F}{\delta b_j^L(K)} \quad (9)$$

به طوری که W_{ji}^L و b_j^L پارامترهای وزن و بایاس نرون j ام در لایه L ام است. α ، نرخ یادگیری و F ، میانگین مربعات خطا می‌باشد.

$$\frac{\delta F(k)}{\delta W_{ji}^L(k)} = S_j^L(k) a_i^{L-1}(k) \quad (10)$$

$$S_j^L(k) = \frac{\delta F(k)}{\delta n_j^L} = \left[\sum_{m=1}^{S^{L+1}} S_m^{L+1}(k) W_{mj}^{L+1}(k) \right] f^L(N_j^L(k)) \quad (11)$$

به طوریکه $S_j^L(K)$ ، حساسیت رفتار شبکه در لایه L ام است. الگوریتم BP ، هر نرون را با توجه به میزان اثرش در بردار خطا تشویق یا تنبیه می‌کند.



شکل 3- مراحل الگوریتم BP

در یک بررسی اجمالی الگوریتم‌های مبتنی بر گرادیان و بخصوص الگوریتم پس‌انتشار خطا، نقاط ضعف ذیل قابل مشاهده می‌باشد:

1- سرعت همگرایی الگوریتم BP آهسته است.

همانطور که می‌دانیم، تغییرات ایجاد شده در پارامترهای شبکه (ماتریس‌های وزن و بردارهای بایاس)، پس از هر مرحله تکرار الگوریتم BP، به اندازه $\alpha \nabla F(x(k))$ است، به طوریکه F ، شاخص اجرایی، x پارامترهای شبکه و α نرخ یادگیری است. از این رو، هر قدر نرخ یادگیری α ، کوچکتر انتخاب گردد، تغییرات ایجاد شده در پارامترهای شبکه، پس از هر مرحله تکرار الگوریتم BP، کوچکتر خواهد بود، که این خود منجر به هموار گشتن مسیر حرکت پارامترها به سمت مقادیر بهینه در فضای پارامترها می‌گردد. این مسئله موجب کندتر گشتن الگوریتم BP می‌گردد. بر عکس با افزایش α ، اگرچه نرخ یادگیری و سرعت یادگیری الگوریتم BP افزایش می‌یابد، لیکن تغییرات قابل توجهی در پارامترهای شبکه از هر تکرار به تکرار بعد ایجاد می‌گردد، که گاهی اوقات موجب ناپایداری و نوسانی شدن شبکه می‌شود که به اصطلاح می‌گویند پارامترهای شبکه واگرا شده‌اند.

2- احتمال به دام افتادن شبکه در نقاط مینیمم محلی وجود دارد. در شبکه‌های MLP، میانگین مجذور خطا، در حالت کلی خیلی پیچیده است و از تعداد زیادی نقطه اکسترمم در فضای پارامترهای شبکه برخوردار می‌باشد. بنابراین الگوریتم پس‌انتشار خطا با شروع از روی یک سری شرایط اولیه پارامترهای شبکه، به نقطه مینیمم سراسری و با شروع از یک مجموعه شرایط اولیه دیگر به نقاط کمینه محلی در فضای پارامترها همگرا می‌گردد، بنابراین زمانی که الگوریتم BP همگرا می‌شود، نمی‌توان مطمئن شد که به یک جواب بهینه رسیده باشیم.

3- همگرایی الگوریتم BP، به یقین مقادیر اولیه پارامترهای شبکه عصبی MLP وابسته است، بطوری که یک انتخاب خوب می‌تواند کمک بزرگی در همگرایی سریعتر الگوریتم BP فراهم آورد. برعکس انتخاب اولیه نادرست پارامترهای شبکه MLP، منجر به گیر افتادن شبکه در نقاط مینیمم محلی در فضای برداری پارامترهای شبکه می‌گردد که این خود منجر به این می‌شود که شبکه خیلی زودتر از معمول به موضعی بیفتد که منحنی یادگیری شبکه برای تعداد بزرگی از دفعات تکرار، تغییر نکند. اگر چه تمام الگوریتم‌های یادگیری تمایل به رسیدن به حداقل خطای آموزشی را دارند ولی با این حال، بسیاری از این الگوریتم‌ها نمی‌توانند به این حداقل برسند و دلیل آن می‌تواند گرفتار شدن در کمینه محلی و یا بی نهایت شدن تکرار مراحل آموزشی باشد که معمولاً در برنامه‌های کاربردی این موارد، مجاز نیستند. با توجه به مشکلات ذکر شده، الگوریتم یادگیری جدیدی را برای آموزش شبکه‌های عصبی پیش‌خور با نام شبکه عصبی‌رسترون چند لایه مبتنی بر ماشین با قابلیت یادگیری فوق‌العاده (ELM)، معرفی می‌کنیم.

3. شبکه‌های عصبی مبتنی بر ماشین با قابلیت یادگیری فوق‌العاده

3-1. الگوریتم ELM

این الگوریتم برای اولین بار توسط هوانگ و همکارانش مطرح شد [4]. از این الگوریتم در تقریب توابع و رگرسیون، طبقه‌بندی، خوشه‌بندی و... استفاده می‌شود و همچنین در زمینه‌های مختلف از جمله پردازش تصویر، پزشکی، مدل سازی سیستم و پیش‌بینی، تشخیص چهره و اثر انگشت و... کاربرد دارد [5,6,7,8,9,10]. الگوریتم پیشنهادی شامل مراحل زیر است:

مجموعه‌ی آموزشی $\mathcal{X} = \{(x_i, t_i) \mid x_i \in \mathbb{R}^n \text{ و } t_i \in \mathbb{R}^m, i=1,2,\dots,N\}$ را به صورت \mathcal{X} در نظر می‌گیریم. تابع فعال ساز لایه پنهان را $g(x)$ و تعداد نرون‌های لایه مخفی را \tilde{N} در نظر می‌گیریم، و تابع فعال ساز خروجی یک تابع خطی در نظر گرفته شده است.

مرحله 1: وزن‌های ورودی w_i و بایاس نرون‌های لایه مخفی b_i برای $i=1,2,\dots,N$ به طور تصادفی انتخاب می‌شوند.
 مرحله 2: ماتریس خروجی لایه مخفی (H) را محاسبه می‌کنیم:

$$H(w_1, \dots, w_{\bar{N}}, b_1, \dots, b_{\bar{N}}, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\bar{N}} \cdot x_1 + b_{\bar{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\bar{N}} \cdot x_N + b_{\bar{N}}) \end{bmatrix}_{N \times \bar{N}} \quad (12)$$

که در این ماتریس، ستون i ام از ماتریس H ، نامین خروجی گره مخفی با توجه به ورودی‌های x_1, x_2, \dots, x_N می‌باشد.
 مرحله 3: ماتریس وزن لایه خروجی (β) محاسبه می‌شود که $\beta = H^+ T$ ، که در آن H^+ ماتریس شبه معکوس مور پنروز است و می‌تواند از روش متعامد سازی از روابط $H^+ = (H^T H)^{-1} H^T$ یا $H^+ = H^T (H H^T)^{-1}$ محاسبه شود [4] و $T = [t_1, t_2, \dots, t_N]^T$ است. طبق قضیه 1-2، این الگوریتم برای هر تابع فعال سازی که بینهایت مشتق پذیر باشد، کار می‌کند و طبق قضیه 2-2، تعداد نرون‌های لایه پنهان از تعداد نمونه‌های آموزشی متمایز است و تعداد آنها می‌تواند $\bar{N} \leq N$ باشد. در [1] و [2] ثابت شده که $SLFN$ ها با N گره مخفی با انتخاب تصادفی وزن و بایاس لایه مخفی می‌توانند دقیقاً N مشاهده مجزا را یاد بگیرند و در [2] ثابت شده که $SLFN$ ها با حداکثر گره‌های پنهان و با هر تابع تحریک غیر خطی می‌توانند N مشاهده مجزا را یاد بگیرند و این توابع می‌توانند شامل توابع مشتق پذیر، مشتق ناپذیر، پیوسته و ناپیوسته و ... باشد.

2-3. نتایج شبیه‌سازی

در این قسمت عملکرد الگوریتم BP و ELM در تشخیص بیماری سرطان سینه مورد سنجش و ارزیابی قرار می‌گیرد. برای الگوریتم BP تعداد نرون‌های لایه مخفی و تعداد نرون خروجی به ترتیب 8 و 1 عدد و نرخ یادگیری 0.1، زمان توقف را 2000 سیکل یا اینکه مقدار خطا کوچکتر از 0.01 در نظر گرفته شده است [11] و همچنین تعداد لایه‌های مخفی یک عدد می‌باشد. در هر دو الگوریتم از روش ارزیابی متقابل ده قسمتی ($K=10$) استفاده شده است با این روش از تمام داده‌ها در مرحله آموزش و تست استفاده شده است. در این روش، داده‌های آموزشی به K قسمت مساوی تقسیم شده، یک قسمت از K قسمت برای تست در نظر گرفته شده و $K-1$ قسمت دیگر برای آموزش استفاده می‌شوند. این فرایند K بار انجام شده و در نهایت میانگین و انحراف استاندارد نتایج گزارش شده است. الگوریتم‌ها با نرم‌افزار $MATLAB(R2013b)$ اجرا شدند. تمام ورودی‌ها در بازه $[0 \ 1]$ و تمام خروجی‌ها در بازه $[-1 \ 1]$ نرمالیزه شده‌اند. تابع فعال‌ساز یک تابع سیگموئید $g(x) = 1/(1 + \exp(-x))$ می‌باشد.

برای به دست آوردن H^+ هم روش‌های زیادی وجود دارد که ما از روش متعامد سازی استفاده می‌کنیم [4]. داده‌های آموزشی مربوط به بیماری سرطان سینه است. تعداد این داده‌های آموزشی 683 نمونه می‌باشد که در میان این نمونه‌ها، 239 نمونه‌ی بیمار و 444 نمونه‌ی سالم وجود دارد و این مجموعه داده‌ها شامل 9 ویژگی مطابق با جدول (1) می‌باشد [11].

جدول 1- مجموعه داده‌های سرطان سینه

Attribute number	Attribute	Value	Mean	Standard deviation
1	Clump thickness	1-10	4.44	2.83
2	Uniformity of cell size	1-10	3.15	3.07
3	Uniformity of cell shape	1-10	3.22	2.99
4	Marginal adhesion	1-10	2.83	2.86
5	Single epithelial cell size	1-10	2.23	2.22
6	Bare nuclei	1-10	3.54	3.64
7	Bland chromatin	1-10	3.45	2.45
8	Normal nucleoli	1-10	2.87	3.05
9	mitoses	1-10	1.60	1.73

نتایج حاصل از شبیه‌سازی الگوریتم در جدول (2) و (3) نشان داده شده است.

جدول 2- مقایسه زمان آموزش در الگوریتم BP و ELM

	الگوریتم BP	الگوریتم ELM
	زمان آموزش (s)	زمان آموزش (s)
میانگین (mean)	44.9	0.0140
انحراف استاندارد (std dv)	15.481	0.0021

جدول 3- مقایسه دقت آزمایش و آموزش در الگوریتم BP و ELM

	الگوریتم BP		الگوریتم ELM	
	دقت آموزش	دقت تست	دقت آموزش	دقت تست
کمینه (min)	86.47	87.14	95.32	92.32
بیشینه (max)	90.73	89.70	96.01	95.22
میانگین (mean)	90.27	89.11	95.81	94.04
انحراف استاندارد (std dv)	1.889	1.196	0.0020	0.0095

4. نتیجه گیری و پیشنهادات

الگوریتم ELM نسبت به الگوریتم BP عملکرد کلی بهتری دارد. همان طور که در جدول (1) مشاهده می شود، روش ELM بسیار سریعتر از BP عمل کرده است. و می تواند در زمانی از ثانیه و حتی کمتر از ثانیه اجرا شود، در صورتی که الگوریتم BP، به زمان بسیار بیشتری نیاز دارد.

با توجه به جدول (2)، دقت ELM بسیار بالاتر از الگوریتم BP می باشد در مرحله تست نیز عملکردی بهتر از الگوریتم BP دارد و این نشان دهنده تعمیم پذیری و پاسخ بهتر به محرک‌های ناشناخته است. پارامترهای مورد نیاز برای آموزش شبکه تو وسط ELM، فقط نوع تابع تحریک و تعداد نرون لایه مخفی می باشد، در صورتی که در الگوریتم BP، علاوه بر این دو پارامتر، نرخ یادگیری و پارامترهای وزن و بایاس نیز باید تنظیم شوند. پس تعداد پارامترهای قابل تنظیم برای الگوریتم ELM، کمتر از BP است و در نتیجه حداقل مداخله انسانی را دارد. در این الگوریتم از هر تابع فعال سازی (از جمله توابع مشتق پذیر، مشتق ناپذیر، پیوسته و یا ناپیوسته) می توان استفاده کرد در صورتی که در BP تابع فعال ساز باید مشتق پذیر باشد. در BP استفاده از نرخ یادگیری نامناسب باعث ایجاد بیش برآزش⁶ و یا واگرایی الگوریتم می شود، در صورتی که در ELM مشکلاتی از این قبیل وجود ندارد.

استفاده از یک الگوریتم بهینه سازی برای به دست آوردن وزن و بایاس لایه مخفی و یا استفاده از یکی از روش های انتخاب ویژگی جهت انتخاب ویژگی های بهتر در تشخیص بیماری سرطان سینه می تواند به عنوان کار آینده انتخاب شود.

5. مراجع

1. G.-B. Huang, Learning capability and storage capacity of two hidden-layer feed forward networks, *IEEE Trans. Neural Networks* 14 (2) (2003) 274–281.
2. S. Tamura, M. Tateishi, Capabilities of a four-layered feed forward neural network: four layers versus three, *IEEE Trans. Neural Networks* 8 (2) (1997) 251–255.
3. G.-B. Huang, H.A. Babri, Upper bounds on the number of hidden neurons in feed forward networks with arbitrary bounded nonlinear activation functions, *IEEE Trans. Neural Networks* 9 (1) (1998) 224–229
4. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feed forward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, 2004, pp. 985-990.
5. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, pp. 489-501, 2006.
6. G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, pp. 513-529, 2012.

⁶ Overfitting

7. G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: a review," *Neural Networks*, vol. 61, pp. 32-48, 2015.
8. G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, pp. 107-122, 2011.
9. G.-B. Huang, X. Ding, and H. Zhou, "Optimization Method Based Extreme Learning Machine for Classification," *Neurocomputing*, vol. 74, 2010, pp. 155–163.
10. H. Zhu, E. C. C. Tsang, X.-Z. Wang, and R. Aamir Raza Ashfaq, "Monotonic classification extreme learning machine," *Neurocomputing*, vol. 225, pp. 205-213, 2/15/ 2017.

11. A. Bhardwaj and A. Tiwari, "Breast cancer diagnosis using genetically optimized neural network model," *Expert Systems with Applications*, vol. 42, pp. 4611-4620, 2015.