

# ارزیابی الگوریتم‌های آموزش بر عملکرد شبکه عصبی چندلایه

## در مکان‌یابی شبکه‌های حسگر بی‌سیم

سید صابر بنی‌هاشمیان<sup>۱</sup>، فضل‌اله ادیب‌نیا<sup>۲</sup>، مهدی آقا صرام<sup>۳</sup>

<sup>۱</sup> گروه مهندسی کامپیوتر، دانشگاه یزد، یزد، s.banihashemian@stu.yazd.ac.ir

<sup>۲</sup> گروه مهندسی کامپیوتر، دانشگاه یزد، یزد، fadib@yazd.ac.ir

<sup>۳</sup> گروه مهندسی کامپیوتر، دانشگاه یزد، یزد، mehdi.sarram@yazd.ac.ir

### چکیده

مکان‌یابی در شبکه حسگر بی‌سیم یکی از مسائلی است که در بسیاری از الگوریتم‌های کاربردی این شبکه مورد استفاده قرار می‌گیرد. در این شبکه‌ها، الگوریتم‌های مکان‌یابی جدا از محدوده به دلیل هزینه اندک و عدم نیاز به تجهیزات خاص مورد توجه قرار گرفته است. از طرفی در سالهای اخیر از الگوریتم‌های محاسبات نرم برای حل این مسئله استفاده شده است. یکی از الگوریتم‌های محاسبات نرم استفاده شده شبکه عصبی است. شبکه عصبی با الگوریتم‌های آموزش مختلف، دارای کارایی متفاوت است. در این مقاله به بررسی تاثیر این الگوریتم‌های گرادیان توام بر روی ۵ معیار مختلف میانگین، حداقل، حداکثر و انحراف معیار خطای مکان‌یابی و همچنین مدت زمان سپری شده جهت آموزش پرداخته شده است. نتایج بدست آمده نشان می‌دهد از چهار الگوریتم گرادیان توام بررسی شده، الگوریتم گرادیان توام اندازه‌شده از نظر میانگین، حداقل، حداکثر و انحراف معیار کمترین خطای مکان‌یابی را داشته و همچنین مدت زمان کمتری جهت آموزش نیاز دارد.

### واژه‌های کلیدی

شبکه حسگر بی‌سیم، مکان‌یابی، شبکه عصبی، الگوریتم‌های آموزش.

### ۱- مقدمه

الگوریتم‌های مکان‌یابی خود به دو دسته مبتنی بر محدوده<sup>۴</sup> و جدا از محدوده<sup>۵</sup> تقسیم می‌گردند. الگوریتم‌های مبتنی بر محدوده از اطلاعات مسافت که با استفاده از ابزاری همچون شدت سیگنال دریافتی<sup>۶</sup>، زمان رسیدن<sup>۷</sup>، اختلاف زمانی رسیدن<sup>۸</sup> و زاویه رسیدن<sup>۹</sup> استفاده می‌نمایند [۵][۶]. ایراد این روش این است که گرها نیازمند بکارگیری تجهیزات خاص و گرانی هستند تا بتوانند مقدار فاصله را بدست بیاورند.

الگوریتم‌های مکان‌یابی جدا از محدوده برخلاف الگوریتم‌های مبتنی بر محدوده تنها از اطلاعات اتصال جهت مکان‌یابی استفاده می‌نمایند [۷]. در بکارگیری این الگوریتم‌ها نیازی به تجهیز گرهای حسگر با ابزار

شبکه‌های حسگر بی‌سیم از تعدادی گر حسگر با توانایی محدود از نظر توان محاسباتی، توان ارتباطی، حافظه و ذخیره انرژی تشکیل شده است [۳]. کاربرد شبکه‌های حسگر با اطلاعات مکانی بهبود یافته و بسیار از الگوریتم‌های مورد استفاده از این اطلاعات استفاده می‌نمایند. به عنوان مثال، در کاربردهای ره‌گیری اشیا<sup>۱</sup> و تشخیص رخداد [۴]، اطلاعات مربوط به مکان وقوع رخداد باید در دسترس باشد. در این شبکه‌ها، به دلیل هزینه بالای تجهیز گرهای حسگر با ابزار تعیین مختصات همچون GPS، از الگوریتم‌های مکان‌یابی استفاده می‌گردد. در این روش از تعدادی گر با مکان مشخص به نام لنگر<sup>۲</sup> یا دیده‌بان<sup>۳</sup> استفاده می‌گردد که سایر گرهای حسگر مکان خود را با استفاده از اطلاعات مکان این گرها تخمین می‌زنند.

<sup>۴</sup> Range-based

<sup>۵</sup> rang-free

<sup>۶</sup> Received Signal Strength Indicator

<sup>۷</sup> Time of Arrival

<sup>۸</sup> Time Difference of Arrival

<sup>۹</sup> Angle of Arrival

<sup>۱</sup> object tracking

<sup>۲</sup> anchor

<sup>۳</sup> beacon

پشتیبان سلسله‌مراتبی برای تخمین مکان گره‌ها پیشنهاد شده است. مولفین نشان داده‌اند که پیچیدگی آموزش در طرح پیشنهادی آنها نسبت به LSVM از  $O(n^3)$  به  $O(n^2)$  کاهش می‌یابد که  $n$  اندازه نمونه آموزش است.

در [۱۴] دو روش SVR و MLSVR برای مکان‌یابی پیشنهاد شده است. در ابتدا نسبت میان فاصله‌های جغرافیایی مین گره‌های حسگر بدست آمده و سپس توسط kernelized regression مدل می‌شود. در پایان گره‌ها مکان خود را محاسبه می‌کنند. در [۱۵] مکان‌یابی به عنوان یک مسئله رگرسیون دیده شده و با استفاده از SVR که یکی از روش‌های معروف یادگیری ماسین است، حل می‌گردد. در [۱۶] نویسندگان یک الگوریتم چندپرشه و جدا از محدوده به نام KRR-ML پیشنهاد نموده‌اند که از روش کرنل برای محاسبه میزان شباهت میان تعداد پرش و فاصله فیزیکی استفاده می‌نماید. همچنین، هر گره حسگر مکانش را با استفاده از این نگاشت پیدا می‌کند.

در [۱۷] یک تحقیق جامع در باره احتمال ادغام کردن مسائل اصلی محاسبات نرم برای حل مسئله مکان‌یابی انجام داده اند. در [۸] یک الگوریتم مکان‌یابی به نام MLPNN پیشنهاد شده است که از شبکه عصبی چندلایه مبتنی بر گرادیان توام استفاده می‌نماید. در این روش کوتاهترین تعداد پرش از گره دیده‌بان تا گره‌های دیده‌بان دیگر به عنوان نمونه آموزش و اطلاعات مکانی متناظر به عنوان کلاس گره در محور X و Y استفاده می‌شود. سپس شبکه عصبی آموزش دیده شده می‌تواند برای مکان‌یابی گره حسگر با مکان نامشخص استفاده گردد. در [۱] نویسندگان یک الگوریتم مکان‌یابی با کمک شبکه حسگر بی‌سیم مبتنی بر Levenberg-Marquard پیشنهاد داده اسند که نسبت به کار [۸] عملکرد بهتری از نظر فضای ذخیره سازی و دقت مکان‌یابی داشته است. در [۲] نویسندگان کاری دیگر بر اساس PCA و شبکه عصبی پیشنهاد نموده اند که نتایج نشان می‌دهد طرح پیشنهادی نسبت به طرح مشابه عملکرد بهتری دارد. در [۱۸] یک الگوریتم مکان‌یابی مبتنی بر محدوده با استفاده از شبکه عصبی پیشنهاد شده است.

### ۳- الگوریتم مکان‌یابی

در این بخش الگوریتم مکان‌یابی شرح داده شده و روش‌های آموزش بررسی شده است. در این مقاله فرض شده است گره‌ها به صورت تصادفی در محیط بکارگرفته شده‌اند. همچنین فاصله ارتباطی گره‌ها یکسان در نظر گرفته شده است.

#### ۳-۱- مکان‌یابی با کمک شبکه عصبی

برای انجام مکان‌یابی مشابه [۱] و [۸]، ابتدا گره‌های دیده‌بان اطلاعات مکان خود را در شبکه حسگر ارسال می‌نمایند. هر گره حسگر با دریافت این اطلاعات، در صورت عدم وجود اطلاعات، مکان گره دیده‌بان و تعداد پرش تا آن گره را ذخیره کرده و سپس به گره حسگر بعدی ارسال می‌نماید. در صورتی دریافت اطلاعات مکان از یک گره دیده‌بان و وجود

اندازه‌گیری فاصله نبوده و تجهیزات خاصی نیاز ندارد. به دلیل سادگی و کم هزینه بودن پیاده سازی، این الگوریتم‌ها بسیار مورد توجه قرار گرفته‌اند. این گره‌ها اطلاعات اتصال و یا تعداد پرش از گره‌های دیده‌بان را بدست آورده و مکان‌شان را با کمک این اطلاعات به صورت تخمینی بدست می‌آورند. در سال‌های اخیر استفاده از تکنیک‌های هوش محاسباتی همچون استفاده از شبکه عصبی در شبکه حسگر بی‌سیم افزایش یافته است. این روش‌ها هم در الگوریتم‌های مکان‌یابی مبتنی بر محدوده و هم در الگوریتم‌های مکان‌یابی جدا از محدوده استفاده شده است. اطلاعات مکانی برای آموزش این شبکه‌ها استفاده شده و سپس مدل شبکه عصبی برای مکان‌یابی گره‌های حسگر ارسال می‌گردد.

در شبکه‌های عصبی الگوریتم‌های آموزش بر روی دقت و سرعت آموزش تاثیر گذارند. در این مقاله ما یک شبکه عصبی چندلایه را برای استفاده در مکان‌یابی پیاده سازی شده است. این شبکه عصبی همانند [۱] و [۸] با استفاده از اطلاعات پرش میان دیده‌بان‌ها آموزش یافته و سپس به گره‌های حسگر ارسال می‌شود. گره‌های حسگر با استفاده از این اطلاعات مکان خود را محاسبه می‌نمایند. تمرکز اصلی این مقاله ارزیابی تاثیر الگوریتم‌های آموزش بر دقت مکان‌یابی و مقایسه میان زمان آموزش آنها است. در ادامه، کارهای مرتبط گذشته شرح داده خواهد شد. در بخش ۳، الگوریتم پیاده‌سازی شرح داده شده و مروری بر الگوریتم‌های مکان‌یابی خواهیم داشت. در بخش چهارم، به شبیه سازی و مقایسه الگوریتم پیشنهادی خواهیم پرداخت. در نهایت، نتیجه گیری در بخش پنجم ارائه خواهد گردید.

### ۲- کارهای گذشته

اخیرا تعدادی از مقالات از روش‌های هوش محاسباتی در مسئله مکان‌یابی در شبکه حسگر بی‌سیم استفاده نموده‌اند. در [۹] یک الگوریتم مکان‌یابی توزیع شده به نام LSVM پیشنهاد شده که مبتنی بر ماشین‌های بردار پشتیبان است. این الگوریتم از اطلاعات مکانی دیده‌بان‌ها به عنوان داده آموزش استفاده در فرایند یادگیری استفاده می‌نماید. برای مکان‌یابی، محیط بکارگیری به نواحی جغرافیایی تقسیم شده و هر گره حسگر یک کلاس از این نواحی اختصاص داده می‌شود. تعدادی بردار پشتیبان برای داده آموزش داده شده محاسبه می‌شود. مجموعه بردار پشتیبان برای کلاس‌های مختلف محاسبه شده و سپس به گره‌های حسگر ارسال می‌گردد.

در [۱۰] صمدیان و همکاران، یک الگوریتم مکان‌یابی با استفاده از ماشین بردار پشتیبان احتمالی پیشنهاد شده است. نویسندگان نشان داده‌اند که ماشین بردار پشتیبان احتمالی از ماشین بردار پشتیبان متداول دقیق‌تر عمل می‌کند. در [۱۱] یک روش مکان‌یابی برای شبکه‌های حسگر بی‌سیم سیار پیشنهاد شده است. مولفین از ماشین بردار پشتیبان برای تخمین مکان گره‌های حسگر با استفاده از اطلاعات مکانی استفاده نموده‌اند. همچنین در [۱۲] یک الگوریتم مکان‌یابی سه-بعدی برای شبکه حسگر بی‌سیم مبتنی بر LSVM پیشنهاد شده است. در [۱۳] ماشین‌های بردار

محور  $Y$  است. گره سردسته از تعداد پرش یک گره دیده‌بان تا گره‌های دیده‌بان دیگر به عنوان ویژگی‌های نمونه‌های یادگیری و از کلاس مکانی گره دیده‌بان به عنوان مقدار خروجی نمونه‌های یادگیری استفاده می‌نماید. سپس گره سردسته مدل شبکه عصبی ایجاد شده شامل وزن‌ها و بایاس‌ها را به گره‌های حسگر ارسال می‌کند. گره‌های حسگر با اطلاعات پرش ذخیره شده و به وسیله شبکه عصبی کلاس مکانی خود را تخمین زده و به موقعیت جغرافیایی نگاشت می‌نمایند.

### ۲-۳- الگوریتم گرادیان توام

یک در الگوریتم انتشار به عقب<sup>۱۰</sup>، وزن‌ها در جهت پرشیب ترین نزول<sup>۱۱</sup> تنظیم می‌گردند. در این جهت، اگرچه تابع با سرعت بیشتری همراه با منفی گرادیان کاهش می‌یابد، اما لزوماً موجب سریعترین هم‌گرایی نمی‌شود. در الگوریتم‌های گرادیان توام، جستجوی همراه با جهت‌های توام انجام می‌شود که معمولاً باعث هم‌گرایی سریعتر از جهت‌های پرشیب ترین نزول می‌شود. همه الگوریتم‌های گرادیان توام با جستجو در جهت پرشیب ترین نزول در اولین تکرار آغاز می‌کنند.

$$P_0 = -g_0 \quad (۴)$$

سپس یک جستجوی خطی برای تعیین فاصله بهینه برای حرکت در جهت جستجوی کنونی انجام می‌شود:

$$w_{k+1} = w_k + \alpha_k P_k \quad (۵)$$

سپس در جستجوی بعدی، جهت طوری تعیین می‌گردد که توام با جهت جستجوی قبلی باشد. در همه الگوریتم‌های گرادیان توام، روش تعیین جهت جستجوی جدید، ادغام جهت پرشیب ترین نزول با جهت جستجوی قبلی است:

$$p_k = -g_k + \beta_k P_{k-1} \quad (۶)$$

در روش Fletcher-Reeves [۱۹]، ثابت  $\beta_k$  از طریق نسبت مربع نرم<sup>۱۲</sup> گرادیان جاری به نسبت مربع نرم گرادیان قبلی محاسبه به صورت ذیل محاسبه می‌گردد.

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \quad (۷)$$

در روش Polak-Ribière [۱۹]، ثابت  $\beta_k$  از طریق ضرب داخلی تغییرات گرادیان قبلی با گرادیان جدید تقسیم بر مربع نرم گرادیان قبلی به صورت ذیل محاسبه می‌گردد:

$$\beta_k = \frac{\Delta g_{k-1}^T g_k}{g_{k-1}^T g_{k-1}} \quad (۸)$$

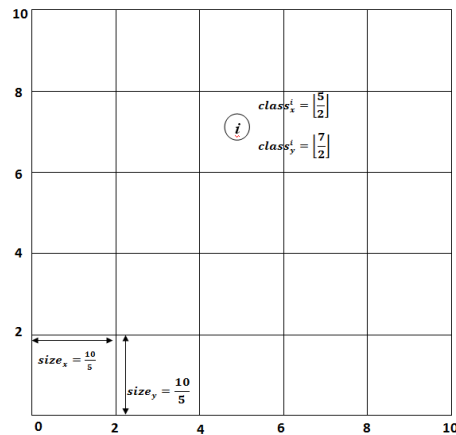
اطلاعات مربوطه در گره حسگر، تعداد پرش بسته دریافتی با تعداد پرش ذخیره شده تا آن گره دیده‌بان مقایسه می‌شود. در صورتی که مقدار کمتری داشت، تعداد پرش جدید جایگزین اطلاعات ذخیره شده قبلی شده و گره حسگر مقدار یک به تعداد پرش بسته دریافتی اضافه کرده و دوباره آن را به گره‌های همسایه خود به جلو ارسال می‌کند. گره‌های سردسته با جمع‌آوری اطلاعات تعداد پرش از تمامی گره‌های دیده‌بان دیگر، اطلاعات مربوط به فاصله تعداد پرش با سایر دیده‌بان‌ها را به همراه مکان خود به یک گره دیده‌بان سردسته ارسال می‌نماید. گره دیده‌بان سردسته، ابتدا کلاس مکانی گره‌های دیده‌بان را محاسبه می‌نماید. برای این کار محیط بکارگیری به صورت مجازی به  $N_x$  و  $N_y$  خانه در محور  $X$  و  $Y$  تقسیم می‌گردد. به عبارت دیگر اندازه هر خانه در صورتی که ابعاد محیط بکارگیری  $[D_x, D_y]$  باشد برابر خواهد بود با:

$$size_x = \frac{D_x}{N_x}, \quad size_y = \frac{D_y}{N_y} \quad (۱)$$

مکان یک گره حسگر  $i$  با موقعیت  $loc^i = (loc_x^i, loc_y^i)$  با استفاده از (۲) به کلاس مکانی در محور  $X$  و  $Y$  نگاشت می‌شود:

$$class_x^i = \left\lfloor \frac{loc_x^i}{N_x} \right\rfloor, \quad class_y^i = \left\lfloor \frac{loc_y^i}{N_y} \right\rfloor \quad (۲)$$

به عنوان مثال گره حسگر  $i$  با موقعیت  $loc^i = (5, 7)$ ، محیط بکارگیری با ابعاد  $[D_x = 10, D_y = 10]$  و با پارامتر  $N_x = 5$  و  $N_y = 5$  به کلاس مکانی  $class_x^i = 2, class_y^i = 3$  همانند شکل ۱ نگاشت می‌گردد:



شکل ۱- نگاشت مکان فیزیکی گره حسگر  $i$  به کلاس مکانی

گره سردسته با استفاده از اطلاعات مکانی جمع‌آوری شده، به آموزش شبکه عصبی می‌پردازد. اطلاعات مکانی جمع‌آوری شده در زیر نشان داده شده است:

(۳)

$$\begin{bmatrix} h(S_1, S_1) & h(S_1, S_2) & \dots & h(S_1, S_k) & : & class_x^1 & class_y^1 \\ h(S_2, S_1) & h(S_2, S_2) & \dots & h(S_2, S_k) & : & class_x^2 & class_y^2 \\ \vdots & \vdots & \dots & \dots & \dots & \vdots & \vdots \\ h(S_i, S_1) & h(S_i, S_2) & \dots & h(S_i, S_k) & : & class_x^i & class_y^i \\ \vdots & \vdots & \dots & \dots & \dots & \vdots & \vdots \\ h(S_k, S_1) & h(S_k, S_2) & \dots & h(S_k, S_k) & : & class_x^k & class_y^k \end{bmatrix}$$

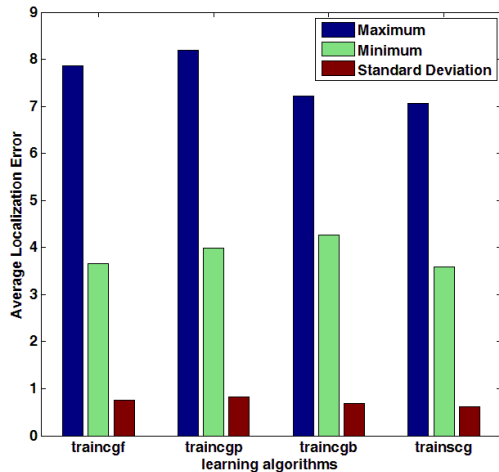
که در آن  $h(S_i, S_j)$  تعداد پرش از گره دیده‌بان  $i$  تا گره دیده‌بان  $j$  و  $class_x^i$  و  $class_y^i$  کلاس موقعیت مکانی گره  $i$  به ترتیب در محور  $X$  و

<sup>10</sup> bachpropagation

<sup>11</sup> steepest descent direction

<sup>12</sup> norm

تعداد کل گره‌ها با مقدار ۴۰۰ در نظر گرفته شده است. نتایج بدست آمده در جدول ۱ نشان داده شده است. همانگونه که مشاهده می‌گردد در روش گرادیان توام اندازه‌شده در شبیه‌سازی با تعداد ۲۰، ۶۰ و ۸۰ گره دیده‌بان عملکرد بهتری از نظر میانگین خطای مکان‌یابی داشته و در تعداد گره دیده‌بان ۴۰ و ۱۰۰ در مرتبه دوم از نظر میانگین خطای مکان‌یابی قرار دارد. در حالت شبکه با ۴۰ گره دیده‌بان، روش Polak-Ribière و در حالت شبکه به ۱۰۰ گره دیده‌بان، روش Powell-Beale Restarts بهتر عمل می‌کند.



شکل ۲- حداکثر، حداقل و انحراف معیار خطای مکان‌یابی برای الگوریتم‌های مختلف آموزش (بر حسب متر)

الگوریتم آموزش	traincgb	traincgp	traincgb	trainscg
۲۰ گره دیده‌بان و ۳۸۰ گره حسگر	11.62	11.84	11.59	<b>11.34</b>
۴۰ گره دیده‌بان و ۳۶۰ گره حسگر	6.77	<b>6.63</b>	6.97	6.72
۶۰ گره دیده‌بان و ۳۴۰ گره حسگر	5.19	5.25	5.05	<b>5.01</b>
۸۰ گره دیده‌بان و ۳۲۰ گره حسگر	4.40	4.54	4.42	<b>4.39</b>
۱۰۰ گره دیده‌بان و ۳۰۰ گره حسگر	4.03	4.03	<b>3.81</b>	3.97

همچنین از نظر زمان سپری شده جهت آموزش در جدول ۲ نشان داده شده است. زمان سپری شده عبارت است از زمان شروع آموزش الگوریتم تا زمانی که فرایند آموزش شبکه عصبی تکمیل گردد. نتایج بدست آمده نشان می‌دهد که روش گرادیان توام اندازه‌شده عملکرد بهتری نسبت به سایر روش‌های مقایسه شده دارد.

در Powell-Beale Restarts [۲۰]، در صورتی جهت جستجو به منفی گرادیان بازنشانی می‌گردد که شرط ذیل برقرار باشد:

$$|g_{k-1}^T g_k| \geq 0.2 \|g_k\|^2 \quad (۹)$$

گرادیان توام اندازه شده<sup>۱۳</sup> [۲۱] به گونه‌ای طراحی شده است که از جستجوی خطی زمان بر اجتناب نماید. این الگوریتم از اطلاعات مرتبه دوم شبکه عصبی استفاده می‌نماید اما تنها نیاز به حافظه  $O(N)$  دارد که  $N$  تعداد وزن‌ها در شبکه است.

#### ۴- ارزیابی

برای ارزیابی میانگین خطای شبیه سازی برای همه گره‌های حسگر، با کمک فرمول (۱) به صورت زیر محاسبه می‌گردد:

$$ALE = \frac{\sum_{i=1}^{SP} \sqrt{(x_{i,loc-true} - x_{i,loc-estm})^2 + (y_{i,loc-true} - y_{i,loc-estm})^2}}{SP} \quad (۱۰)$$

که در آن  $x_{i,loc-true}$  موقعیت حقیقی گره  $i$  در محور  $x$ ،  $y_{i,loc-true}$  موقعیت حقیقی گره  $i$  در محور  $y$ ،  $x_{i,loc-estm}$  موقعیت تخمینی گره  $i$  در محور  $x$  و  $y_{i,loc-estm}$  موقعیت تخمینی گره  $i$  در محور  $y$  است. همچنین  $SP$  برابر (تعداد کل گره‌ها - گره‌های دیده بان) است. هر چه میانگین خطای شبیه سازی برای یک الگوریتم کمتر باشد، آن الگوریتم بهتر خواهد بود. به عبارت دیگر هدف یافتن الگوریتمی با میانگین خطای کمتر است. برای شبیه‌سازی، یک شبکه حسگر با ۴۰۰ گره شامل ۶۰ گره دیده‌بان و ۳۴۰ گره حسگر با مکان ناشناخته است. فاصله ارتباطی گره‌ها یکسان و ۷ متر می‌باشد. این گره‌ها در یک محیط  $60 \times 60$  به صورت تصادفی پخش شده‌اند. برای تقسیم بندی مجازی جهت مکان‌یابی به  $N_x = 30$  و  $N_y = 30$  در نظر گرفته شده‌اند. در این آموزش شبکه عصبی ۲ لایه پنهان هر کدام شامل به ترتیب ۶۴ و ۲۰ نرون وجود داشته و ۲ نرون در لایه خروجی وجود دارد. شبیه سازی در نرم‌افزار متلب انجام شده و برای روش Fletcher-Reeves از تابع traincgb، برای روش Polak-Ribière از تابع traincgp، برای روش Powell-Beale Restarts از تابع traincgb، برای روش گرادیان توام اندازه‌شده از تابع trainscg در متلب استفاده شده است. هر شبیه سازی ۵۰ بار تکرار شده است. شکل ۲ حداکثر، حداقل و انحراف معیار خطای مکان‌یابی برای الگوریتم‌های مختلف آموزش را نشان می‌دهد.

همانگونه که مشاهده می‌گردد الگوریتم گرادیان توام اندازه‌شده عملکرد بهتری از نظر میزان حداکثر، حداقل و انحراف معیار خطای مکان‌یابی نسبت به سایر الگوریتم‌های آموزش دارد.

برای ارزیابی تاثیر تعداد گره های دیده بان بر روی میانگین خطای مکان‌یابی، تعداد گره‌های دیده‌بان به ترتیب ۲۰، ۴۰، ۶۰، ۸۰ و ۱۰۰ گره از

<sup>13</sup> Scaled Conjugate Gradient

- [4] C. Y. Chong and S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," in *Proc. of IEEE*, vol. 91, no. 8, pp. 1247-1256, 2003.
- [5] L. Doherty, K. S. Pister and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1655-1663, 2001.
- [6] A. Pal, "Localization algorithms in wireless sensor networks: Current approaches and future challenges," *Network Protocols and Algorithms*, vol. 2, no. 1, pp. 45-73, 2010.
- [7] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: a survey". *Telecommunication Systems*, 52(4), 2419-2436, 2013.
- [8] A. Chatterjee. "A Fletcher-Reeves conjugate gradient neural-network-based localization algorithm for wireless sensor networks". *IEEE Transactions on Vehicular Technology*, vol. 59(2), pp. 823-830, 2010.
- [9] D. A. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 7, pp. 981-994, Jul. 2008.
- [10] R. Samadian and S. M. Noorhosseini, "Probabilistic support vector machine localization in wireless sensor networks," *ETRI J.*, vol. 33, no. 6, pp. 924-934, Dec. 2011.
- [11] S. Afzal, and H. Beigy. "A localization algorithm for large scale mobile wireless sensor networks: a learning approach". *The Journal of Supercomputing*, 69(1), 98-120, 2014.
- [12] R. Huan, Q. Chen, K. Mao, and Y. Pan, "A three-dimensional localization algorithm for wireless sensor network nodes based on SVM," in *Proc. Int. Conf. Green Circuits Syst.*, Jun. 2010, pp. 651-654.
- [13] V. Feng, and S. Y. Chang, "Determination of wireless networks parameters through parallel hierarchical support vector machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 505-512, Mar. 2012.
- [14] J. Lee, W. Chung, and E. Kim, "A new kernelized approach to wireless sensor network localization," *Inf. Sci.*, Vol. 243, pp. 20-38, 2013.
- [15] J. Lee, B. Choi, and E. Kim, "Novel range-free localization based on multidimensional support vector regression trained in the primal space". *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24(7), pp. 1099-1113, 2013.
- [16] X. Yan, A. Song, Z. Yang and W. Yang, "An improved multihop-based localization algorithm for wireless sensor network using learning approach". *Computers & Electrical Engineering*, vol. 48, pp. 247-257, 2015.
- [17] C. So-In, S. Permpol and K. Rujirakul. "Soft computing-based localizations in wireless sensor networks". *Pervasive and Mobile Computing*, vol. 29, pp. 17-37, 2015.
- [18] A. Payal, C. S. Rai and B. R. Reddy. "Analysis of some feedforward artificial neural network training algorithms for developing localization framework in wireless sensor networks". *Wireless Personal Communications*, vol. 82(4), pp. 2519-2536, 2015.
- [19] M. T. Hagan, H. B. Demuth and M. Beale. "Neural network design". Boston: PWS publishing company, 1996.
- [20] M. J. D. Powell. "Restart procedures for the conjugate gradient method". *Mathematical Programming*, vol. 12(1), pp. 241-254, 1997.
- [21] M. F. Moller. "Ascaled conjugate gradient algorithm for fast supervised learning". *Neural Networks*, vol. 6(4), pp. 525-533, 1993.

جدول ۲- مدت زمان سپری شده الگوریتم های آموزش برای تعداد مختلف گره های دیده بان				
الگوریتم آموزش	traincgf	traincgp	traincgb	traincsg
۲۰ گره دیده بان و ۳۸۰ گره حسگر	0.71	0.73	0.71	<b>0.67</b>
۴۰ گره دیده بان و ۳۶۰ گره حسگر	0.86	0.94	0.90	<b>0.84</b>
۶۰ گره دیده بان و ۳۴۰ گره حسگر	0.98	1.08	1.04	<b>0.98</b>
۸۰ گره دیده بان و ۳۲۰ گره حسگر	1.04	1.12	1.15	<b>1.01</b>
۱۰۰ گره دیده بان و ۳۰۰ گره حسگر	1.27	1.39	1.50	<b>1.26</b>

## ۵- نتیجه گیری

در این مقاله به بررسی تاثیر الگوریتم های آموزش در شبکه های عصبی بر روی دقت مکان یابی در شبکه حسگر بی سیم پرداخته شد. برای اینکار ابتدا مکان یابی با استفاده از شبکه عصبی شرح داده شده است. این شبکه عصبی تنها از اطلاعات تعداد پرش برای و اطلاعات مکانی گره های دیده بان برای آموزش استفاده می نماید. گره های حسگر با استفاده از مدل شبکه عصبی بدست آمده، قادر به تخمین مکان خود هستند. سپس مروری بر الگوریتم های آموزش گرادیان توام در شبکه عصبی انجام شده است. چهار الگوریتم گرادیان توام روش Fletcher-Reeves، روش Polak-Ribière، روش Powell-Beale Restarts و روش گرادیان توام اندازه شده بررسی شده است. سپس از طریق انجام شبیه سازی به مقایسه عملکرد این روش ها پرداخته شده است. نتایج نشان می دهد الگوریتم گرادیان توام اندازه شده از نظر میانگین، حداکثر، حداقل و انحراف معیار خطای مکان یابی نسبت به روش های دیگر عملکرد بهتری دارد. همچنین به مقایسه مدت زمان آموزش این روش ها پرداخته شد که نتایج نشان می دهد الگوریتم گرادیان توام اندازه شده از سرعت بالاتری در اجرا برخوردار بوده و مدت زمان کمتری نسبت به سایر روش های مطالعه شده جهت آموزش نیازمند است.

## مراجع

- [۱] بنی هاشمیان، سید صابر، رضائیان، مهدی، درهمی، ولی، "مکان یابی در شبکه های حسگر بی سیم با استفاده از شبکه عصبی چند لایه"، دوازدهمین کنفرانس سیستم های هوشمند، به، ایران، ۱۳۹۲.
- [۲] بنی هاشمیان، سید صابر، ادیب نیا، فضل اله. "مکان یابی در شبکه های حسگر بی سیم با استفاده از روش های هوشمند". ششمین کنفرانس فناوری اطلاعات و دانش، دانشگاه شاهرود، شاهرود، ایران، ۱۳۹۳.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, pp. 102-114, 2002.