



# یک روش جدید مسیریابی در سیستمهای هدایت اتومبیل با استفاده از شبکه های عصبی

محمد رضا میبدی

دانشکده مهندسی کامپیوتر دانشگاه امیر کبیر تهران

Email: mmeybodi@aut.ac.ir

مهدی هاشم زاده

دانشکده مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد اهر

Email: meh\_hashemzadeh@yahoo.com

نقشه های دیجیتالی کارا و ارسال دستورات بصورت صوتی و تصویری پیشرفت چشمگیری داشته اند [1]. همچنین تا بحال بسیاری از مشکلات و مسایل مبهم مربوط به این سیستمها مثل تعیین موقعیت دقیق اتومبیل، جمع آوری اطلاعات کامل نقشه های دیجیتالی و راهنمایی سریع و قابل اطمینان رانندگان از سوی این سیستمها حل شده است. ولی هنوز هم مشکلاتی در مورد الگوریتمهای مسیریابی و محاسبه مسیر بهینه وجود دارد.

مسئله انتخاب مسیر در سیستمهای هدایت اتومبیل، مسئله جستجوی مسیر بهینه از نقطه مبدا تا نقطه مقصد بر روی نقشه تعریف شده است [2]. شاید از این تعریف اینطور به نظر برسد که پیدا کردن مسیر بهینه همان کوتاهترین مسیر بین مبدا و مقصد است. و معمولا بلافاصله سراغ الگوریتمهای استاندارد که برای جستجوی گرافها وجود دارند میروند. در حالیکه اگر هدف این باشد که این سیستمها مثل یک راننده ای عمل کنند که تمامی مسیرهای شهر را می شناسد و از محدودیتهای راهنمایی و رانندگی همه خیابانها اطلاع دارد ( مثل گردش به چپ یا راست ممنوع، دور زدن ممنوع و ...) و همزمان اطلاعات مربوط به حجم ترافیک راهها نیز به وی مخابره شود، الگوریتمهای جستجوی گراف به تنهایی قادر به حل مسئله نخواهند بود. البته برخی تکنیکها را در این رابطه بکار برده اند و پیشرفتهایی حاصل شده است. و برخی روشها و الگوریتمها نیز مطرح شده اند که با توجه به مشکلاتی که در ادامه بحث می کنیم اکثرا می توانند جنبه تئوری داشته باشند و معمولا به سختی می توان آنها را پیاده سازی نمود.

برای مواجه نشدن با برخی مشکلات و مسائل مطرح در این سیستمها ما ایده ای را در [3] برای عمل مسیر یابی پیشنهاد نمودیم. در روش مطرح شده از قدرت یادگیری شبکه های عصبی استفاده کرده و تمامی شرایط ممکنه را که در یک شهر می تواند برای مسیریابی درخواست شود را از قبل پیش بینی کرده و جواب آنها را نیز پیدا می کنیم، و شبکه عصبی را آموزش می دهیم. و پس از آموزش شبکه بهنگام نیاز از آن استفاده می کنیم. در واقع هم صورت مسئله و هم

**چکیده:** در این مقاله روشی برای پیدا کردن مسیر بهینه در سیستمهای هدایت اتومبیل مطرح گردیده است. در روش پیشنهادی از قدرت یادگیری و سرعت فوق العاده شبکه های عصبی بهره گرفته ایم. روش پیشنهادی مشکلات و محدودیتهای الگوریتمهای استاندارد را که برای این سیستمها مطرح شده اند، را ندارد. در این متد علاوه بر پیدا کردن مسیر بهینه بین مبدا و مقصد، موارد دیگری مثل در نظر داشتن شرایط ترافیکی راهها، محدودیتهای راهنمایی و رانندگی و تطبیق مسیر اتخاذ شده بر اساس درخواستهای رانندگان را براحتی می توان پیاده سازی کرد. همچنین این روش مسیریابی برای شهرهایی که نقشه راههای آنها از ساختار قاعده مند و استاندارد برخوردار نیستند، می تواند بسیار مفید باشد. و با توجه به ویژگیهای خاص خیابانها در کشور ایران، این روش می تواند در پیاده سازی سیستمهای هدایت اتومبیل در کشورمان مورد استفاده قرار گیرد. از آنجائیکه تا بحال در این سیستمها از شبکه های عصبی استفاده نشده است، لذا ایده مطرح شده می تواند سرآغاز تحقیقات جدید در این سیستمها باشد.

**واژه های کلیدی:** سیستمهای هدایت اتومبیل، شبکه های عصبی، الگوریتمهای مسیریابی، نقشه دیجیتالی

## ۱- مقدمه

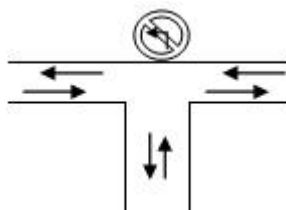
سیستمهای هدایت اتومبیل دستگاههایی هستند که بر روی اتومبیل نصب میشوند و با استفاده از سیگنالهای GPS و نقشه دیجیتالی موقعیت اتومبیل را نمایش داده و در پیدا کردن مسیر مناسب بین مبدا تا مقصد به راننده کمک می کنند. تقریبا ده سال است که جمع آوری اطلاعات مربوط به میزان ترافیک راهها و مخابره آنها به وسایل نقلیه در اکثر کشورها انجام می شود. سیستمهایی که به این اطلاعات دسترسی دارند، می توانند این اطلاعات را نیز در عمل مسیریابی خود دخالت دهند. و مسیری که برای رسیدن به مقصد انتخاب می شود، از میزان ترافیک پائینی برخوردار باشد.

از سال ۱۹۹۵ تا به حال سیستمهای هدایت اتومبیل پیشرفته ای به بازار عرضه شده اند. این سیستمها از نظر امکانات راهنمایی رانندگان،

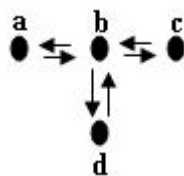
اگرچه برخی از الگوریتمهای پیشنهاد شده قادرند بصورت بلادرنگ (Real Time) پاسخ مسئله را بدست آورند، ولی در مقابل اغلب مسیرهایی که محاسبه می کنند از کیفیت پایینی برخوردارند. و حتی اگر این مسئله را هم در نظر نگیریم، برخی محدودیتها و مشکلات در عمل مسیریابی سیستمهای هدایت اتومبیل وجود دارند که باعث می شوند پیاده سازی اغلب این روشها امکانپذیر نباشد. برخی از این محدودیتها به قرار زیر هستند:

#### الف) محدودیتهای چرخشی در تقاطعها

یکی از مشکلات بارز الگوریتمهای مسیریابی سیستمهای هدایت اتومبیل امکان دخالت دادن محدودیتهای چرخشی در تقاطعهای شهر می باشد. بعنوان مثال اگر تقاطع موجود در شکل ۱.الف) را در نظر بگیریم، گراف حاصل از آن به فرم شکل ۱.ب) خواهد بود. از نظر الگوریتم دایجکسترا و  $A^*$  برای رفتن از گره  $d$  به گره  $a$  از گره  $b$  می توان عبور کرد. در حالیکه بخاطر تابلوی گردش به چپ ممنوع در تقاطع  $b$  این امر امکانپذیر نیست. پیاده سازی محدودیتهایی نظیر این که تعداد و گونه های آنها نیز بسیار زیادند در الگوریتمهایی نظیر  $A^*$  امکانپذیر نیست. و اگر هم بخواهند در مورد هر گرهی همه این محدودیتها را بررسی کنند زمان جستجو بسیار بالا خواهد رفت. که اگر این مشکل را بخواهیم در مورد راههای داخل شهرهای ایران بررسی نمائیم، مسئله بسیار حادتر خواهد بود. چرا که اکثریت تقاطعهای خیابانها در شهرهای کشورمان از محدودیتهای چرخشی برخوردارند.



الف)



ب)

شکل ۱. یک تقاطع با محدودیت چرخشی به چپ ممنوع

جواب بهینه آنها در نظر می گیریم و با آموزش دادن شبکه عصبی آنها را ذخیره می کنیم. قابل ذکر است که شبکه های عصبی بهنگام پاسخ دادن مسئله با سرعت بسیار بالایی کار می کنند. و این سرعت بالای شبکه های عصبی در سیستمهای هدایت اتومبیل که نیاز به جواب بلادرنگ دارند بسیار مفید خواهد بود. در این مقاله ایده مطرح شده قبلی خود را همراه با جزئیات پیاده سازی و نتایج تجربی حاصل از پیاده سازی عملی یک سیستم هدایت اتومبیل بر روی نقشه شهر تبریز ارائه می کنیم.

در بخش دوم مقاله مشکلات و محدودیتهای سایر الگوریتمها و روشهای ارائه شده تا حال را بررسی می کنیم. در بخش سوم روش پیشنهادی خود را مطرح کرده و تجربیات عملی بدست آمده را بیان می کنیم. در بخش چهارم مزایای طرح خود را مطرح کرده و درباره آنها بحث می کنیم.

## ۲- مشکلات و محدودیتهای الگوریتمها و روشهای مسیریابی موجود

از ابتدا اکثر تحقیقات انجام شده و الگوریتمهای ارائه شده در این زمینه، بر پایه الگوریتم دایجکسترا و الگوریتم  $A^*$  بوده است [4]. در این الگوریتمها اندازه و بزرگی گراف جستجو در زمان بدست آوردن پاسخ بسیار مهم است. و گراف راهها معمولا از سائز بزرگی برخوردارند. (تعداد گرهها و لبه های آن بسیار زیاد است). لذا اکثر تحقیقات در راستای بهینه کردن زمان پاسخ بوده است. چرا که با توجه به موقعیت متحرک اتومبیل و برای حفظ امنیت سرنشینان آن و همچنین ارسال بموقع دستورات لازم قبل از رد شدن از تقاطعهای موجود در مسیر اتخاذ شده، عمل مسیریابی باید بصورت بلادرنگ (Real Time) انجام شود.

الگوریتمهای برپایه الگوریتم  $A^*$  مثل الگوریتمهای  $RTA^*$  و  $LRTA^*$  که بصورت Real Time عمل می کنند ارائه شده است [4]. که این روشها معمولا بصورت تخمینی مسافت بین تقاطع بعدی و مقصد را محاسبه می کنند. و یک جواب نزدیک به بهینه را از نظر مسافت بدست می آورند. و یا مثلا در برخی دیگر از روشهای ارائه شده زمان محاسبات الگوریتمهای بر پایه الگوریتم دایجکسترا را با محدود کردن فضای جستجو به ناحیه ای که شرایط ترافیکی آن تغییر یافته تا حدی بهبود بخشیده اند. البته این روش در جستجوهای سراسری نمی تواند مورد استفاده قرار گیرد. الگوریتم جستجو با محدودیت زمان جستجو TCS نیز ارائه شده است [5] که یک جواب نزدیک به بهینه را را در یک زمان مشخص بدست می آورد. تکنیکهای مختلف دیگری نیز برای بالا بردن سرعت پردازش مطرح شده است [6,7]. برخی دیگر از الگوریتمهای مطرح شده برای بالا بردن سرعت عمل جستجو از جزبندی گرافها برای کاهش فضای جستجو استفاده می کنند که روشهای متعددی در این زمینه نیز ارائه شده اند [8,9].

#### ب) مشکلات الگوریتمهای بر اساس جزبندی گراف

اکثر الگوریتمهای مطرح شده برای مسیریابی در سیستمهای هدایت اتومبیل بنحوی از روشهای جزبندی گراف استفاده می کنند. به این ترتیب که گراف راهها را به چندین جز (ناحیه) تقسیم می کنند و عمل جستجو را فقط در داخل ناحیه ها انجام می دهند و برای انتقال بین ناحیه ها از لینکهای ثابت و از پیش تعریف شده استفاده می کنند. که این لینکهای ثابت را معمولا آزاد راهها را در نظر می گیرند. و برای گذر از یک منطقه شهر به منطقه ای دیگر همیشه از این راههای ثابت استفاده می کنند. این الگوریتمها دارای دو مشکل اساسی هستند. نخست اینکه اگر قرار باشد همه خودروها از این سیستمها استفاده نمایند بعد از گذشت مدت زمان بسیار کوتاهی این لینکهای ثابت دچار ازدحام خواهند شد. دوم اینکه معمولا بخاطر یکپارچه بودن ساختارهای محیطهای شهری، عملا امکان جزبندی کامل گراف راههای شهر وجود ندارد. چرا که گره ها معمولا بصورت یکنواخت در همه جای شهر توزیع شده اند. لذا خود جزبندی گراف راههای شهر یکی از مسائل پیچیده این سیستمها خواهد بود. و اگر بخواهیم از این روشها در شهرهای ایران استفاده نمائیم، معمولا بخاطر قرار گرفتن بافت قدیمی و بافت جدید خیابانهای شهرها در داخل یکدیگر و عدم وجود یک استاندارد مشخص، تقریبا غیر ممکن خواهد بود.

#### ج) عدم توانایی الگوریتمهای ارائه شده برای پیاده سازی جستجوی تطبیقی

در الگوریتمهای ارائه شده تا به امروز نمی توان براساس درخواستهای مختلف رانندگان عمل مسیریابی را انجام داد. در دنیای واقعی معمولا رانندگان علاوه بر رسیدن به مقصد پارامترهای دیگری را نیز در نظر می گیرند. بعنوان مثال در طی مسیر می خواهند از مکانهای خاصی مثل پمپ بنزین، بانک، رستوران، ... عبور کنند. و یا از خیابانهایی عبور کنند که اگر چه مسیر طولانی دارند ولی از آرامش کافی برای رانندگی برخوردارند. و موارد مختلف دیگر. که پیاده سازی و در نظر گرفتن این پارامترها تقریبا در الگوریتمهای موجود امکانپذیر نیست. اما در دنیای واقعی اگر این پارامترها را در مسیریابی خود دخالت ندهیم، در واقع این سیستمها کارایی مناسبی را نخواهند داشت.

#### د) مشکلات دخالت دادن اطلاعات وضعیت ترافیکی راهها

الگوریتمهایی که تا بحال برای در نظر داشتن شرایط ترافیکی راهها ارائه شده اند، اکثرا جنبه تئوری داشته اند و پیاده سازی عملی آنها معمولا امکانپذیر نیست. و برخی از آنها که پیاده سازی هم شده اند از توانایی کافی برخوردار نیستند و برخی مشکلات را دارا هستند. یکی از مشکلات اصلی آنان در نظر گرفتن شرایط ترافیکی برای همه لینکها است. که در عمل تنها در مورد برخی از خیابانها اطلاعات ترافیکی مخابره می شود. الگوریتمهای مطرح شده بهنگام تغییر شرایط ترافیکی

یک خیابان، معمولا بالاجبار همه لینکها را برای یافتن لینک مورد نظر جستجو می کنند. و این مسئله زمان عمل جستجو را که در این سیستمها از اهمیت بالائی برخوردار است بسیار بالا خواهد برد.

#### ۳- روش پیشنهادی برای مسیریابی با استفاده از شبکه

##### های عصبی

با توجه به مشکلات مطرح شده در بخش دوم و همچنین اگر بخواهیم این سیستمها واقعا در عمل مورد استفاده قرار بگیرند. و ایده هایی که مطرح می شوند صرفا جنبه تئوری نداشته باشند. و حتی راننده ای که برای اولین بار وارد یک شهر شده است بمانند رانندگان ساکن آن شهر بتواند در خیابانها رانندگی کند، بهترین راه حل آموزش سیستم برای جواب دادن به هر نوع درخواستی می باشد. به این ترتیب که تمامی درخواستهای ممکنه را برای پیدا کردن مسیر بهینه بین هر دو تقاطع موجود در گراف راهها را در نظر می گیریم و با توجه به شرایط ترافیکی موجود و پارامترهای مختلف در نظر گرفته شده جواب بهینه را نیز معین می کنیم. و این حالتها را بصورت الگوهای ورودی و خروجی متناظر و در شکل بردارهای ورودی و خروجی برای آموزش شبکه عصبی استفاده می کنیم. و از قدرت فوق العاده یادگیری و پاسخ سریع شبکه های عصبی بهره می گیریم و این الگوها را به شبکه آموزش می دهیم. در واقع ما تمامی حالات ممکن برای درخواست مسیر و جواب آنها در سیستم ذخیره می نمائیم و در موقع نیاز استفاده می کنیم. شاید اینطور به نظر برسد که چرا از یک پایگاه داده برای اینکار استفاده نکنیم. پاسخ اینکه با توجه به تعداد بسیار زیاد حالات ممکنه نیاز به یک پایگاه داده بزرگ است که با توجه به محدودیتهای سخت افزاری و بویژه ظرفیت کم حافظه این سیستمها امکان پیاده سازی آن وجود ندارد. و حتی در صورت امکان نیز سرعت پاسخ شبکه های عصبی در مقایسه با هر نوع پایگاه داده ای بسیار زیاد می باشد. البته در ادامه جزئیات بیشتری در این رابطه مطرح خواهیم کرد.

#### ۱-۳ جزئیات روش پیشنهادی

الگوی ورودی (بردار ورودی) شبکه عصبی شامل مولفه هایی برای مشخص کردن نقاط مبدا و مقصد بعلاوه مولفه هایی برای مشخص نمودن وضعیت ترافیکی خیابانهایی که شرایط ترافیکی آنها مخابره می شود و همچنین هر پارامتر دیگری که بخواهیم در عمل مسیریابی دخالت دهیم، خواهد بود. بعنوان مثال نقشه شکل ۲ را در نظر بگیرید. در این نقشه شش تقاطع، سیزده خیابان، یک تابلوی گردش به چپ ممنوع، یک پمپ بنزین وجود دارد. فرض کنیم خیابانهای ۲ و ۳ و ۴ و ۵ دارای سیستم مخابره وضعیت ترافیکی باشند.

برای نمایش تقاطعهای مبدا و مقصد شماره آنها را بصورت بایبولار (۱+ و ۱-) نمایش می دهیم. و هر لینکی که دارای ازدحام ترافیکی باشد مولفه مربوط به آن لینک را در بردار ورودی با مقدار ۱+ و

خواهد بود و خیابانهایی که در مسیر نیستند مقدارشان ۱- خواهد بود. مثلا برای حالت مطرح شده فوق اگر جواب مسئله را بخواهیم خیابانهایی ۶ و ۱۰ و ۱۱ در نظر بگیریم، بردار خروجی متناظر با بردار ورودی فوق بصورت  $Y[-1,-1,-1,-1,-1,+1,-1,-1,+1,+1,-1,-1]$  خواهد بود.

بعنوان مثالی دیگر حالتی را در نظر بگیریم که مبدا تقاطع ۴ و مقصد تقاطع ۶ باشد و لینکهای ۵ و ۲ از ترافیک بالا برخوردار باشند. با توجه به محدودیت "چرخش به چپ ممنوع" که در تقاطع ۵ وجود دارد مسیر جواب را لینکهای ۹ و ۵ تعیین می کنیم. بنابراین الگوهای ورودی و خروجی بصورت زیر خواهند بود.

$$X[+1,-1,-1,+1,+1,-1,+1,-1,-1,+1,-1]$$

$$Y[-1,-1,-1,-1,+1,-1,-1,-1,+1,-1,-1,-1]$$

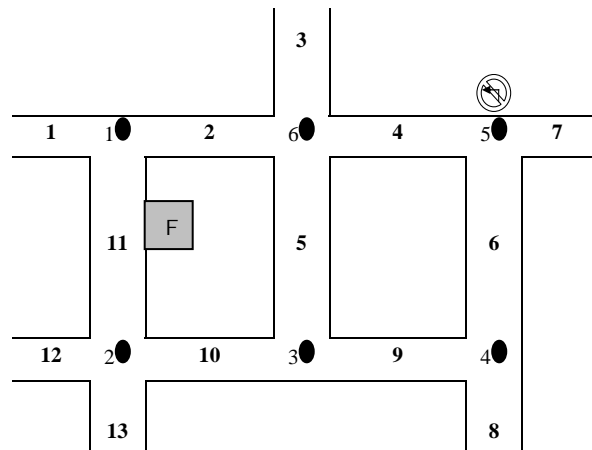
به این ترتیب تمامی حالات ممکن را در کل نقشه راهها تولید کرده و پاسخ مناسب را نیز تعیین می کنیم و بصورت بردارهای ورودی و خروجی متناظر (Training Vectors) برای آموزش شبکه عصبی ذخیره می نماییم. (در نمونه پیاده سازی شده یک برنامه کامپیوتری برای انجام این عمل ایجاد کردیم. که کاربر فقط اسم نقاط مبدا و مقصد و لیست خیابانهایی دچار ازدحام را مشاهده میکند و سپس خیابانهایی مسیر جواب را وارد می کند. و این برنامه کامپیوتری است که بردارهای ورودی و خروجی را تولید می کند در بخش ۳،۳ توضیحات بیشتری برای پیاده سازی این روش ارائه شده است.) بعد از تولید کلیه حالات و بردارهای خروجی متناظر آنها، شبکه عصبی را آموزش می دهیم. ( در بخش ۳،۳ توضیحات کافی را در مورد شبکه عصبی استفاده شده ارائه می دهیم.) پس از آموزش شبکه عصبی تنها چیزی که نیاز است در سیستم هدایت اتومبیل ذخیره کنیم، ماتریس وزنی حاصل از آموزش شبکه عصبی می باشد. به هنگام کار سیستم الگوی ورودی بر اساس درخواست راننده تشکیل شده و به شبکه داده می شود و شبکه با سرعت بسیار بالایی خروجی متناظر آموزش دیده خود را بر می گرداند. شکل ۳ و شکل ۴ مراحل انجام کار را بصورت مرحله به مرحله نشان می دهند.

### ۳-۲ شبکه های عصبی پس انتشار

#### (Backpropagation Neural Networks)

یکی از شبکه های عصبی که توانایی بسیار بالایی برای ذخیره و یادگیری الگوها دارد، شبکه های عصبی پس انتشار می باشند. در این بخش توضیحاتی را در مورد این شبکه ها ارائه می کنیم. الگوریتم پس انتشار یک متد یادگیری می باشد. این شبکه های عصبی جز شبکه های چند لایه هستند که در اکثر کاربردهای امروزی استفاده می شوند. در الگوریتم پس انتشار دو مسیر محاسباتی موجود است. مسیر اول به مسیر رفت یا Feedforward موسوم است و مسیر دوم، مسیر برگشت یا Backward می باشد. در مسیر رفت پارامترهای شبکه در

درغیر اینصورت با مقدار ۱- نمایش می دهیم. و در صورت درخواست پمپ بنزین، مولفه مربوط به آنرا در بردار ورودی ۱+ می گذاریم. با توجه به اینکه کل نقشه شش تقاطع دارد لذا سه بیت برای نمایش نقاط مبدا و مقصد کافی خواهد بود. در جدول ۱ نمایش باینری و بایپولار تقاطعها را مشاهده می کنید.



شکل ۲. یک نقشه نمونه

جدول ۱. نمایش باینری و بایپولار تقاطعها

تقاطع	باینری	بایپولار
1	001	-1-11
2	010	-11-1
3	011	-111
4	100	1-1-1
5	101	1-11
6	110	11-1

الگوی ورودی (بردار ورودی) شبکه عصبی دارای یازده مولفه خواهد بود. سه مولفه اول برای مشخص کردن نقطه مبدا، سه مولفه دوم برای مشخص کردن نقطه مقصد، مولفه های ۷ و ۸ و ۹ و ۱۰ م به ترتیب برای مشخص کردن وضعیت ترافیکی خیابانهایی ۲ و ۳ و ۴ و ۵ می باشند و مولفه یازدهم برای مشخص کردن درخواست پمپ بنزین در نظر گرفته می شود.

بعنوان مثال حالت طی مسیر از تقاطع شماره ۵ به تقاطع شماره ۱ را با درخواست وجود پمپ بنزین در مسیر راه و همچنین ترافیک بالای خیابانهایی شماره ۳ و ۴ بصورت بردار

$X[+1,-1,+1,-1,-1,+1,-1,+1,+1,-1,+1]$  خواهد بود. و الگوی خروجی (بردار خروجی) شامل مولفه هایی به تعداد کل خیابانها خواهد بود. که در این نقشه سیزده تا خیابان وجود دارد و مولفه مربوط به هر خیابان که در مسیر اتخاذ شده (مسیر جواب) قرار داشته باشد، ۱+ و

توضیحات کافی در مورد ساختار این شبکه ها و همچنین الگوریتم آموزش پس انتشار در [10] ارائه شده است. پیشرفتهای و تغییرات زیادی در رابطه با شبکه های پس انتشار حاصل شده است. و روشهای مختلفی جهت بالا بردن سرعت آموزش و دقت این شبکه ها ارائه شده اند. برخی از این پیشرفتهای حاصله را که می توان در کاربرد پیشنهادی استفاده کرد در [11,12,13,14,15,16] ارائه شده اند.

### ۳-۳ تجربیات عملی

در یک تجربه عملی روش پیشنهادی را بر روی اطلاعات بر گرفته شده از نقشه شهر تبریز پیاده سازی کردیم. نرم افزاری را پیاده سازی نمودیم که عملیات بدست آوردن گراف راهها را بسادگی می توانیم توسط آن انجام دهیم. پس از ذخیره کردن داده های مربوط به تقاطعها و خیابانها و سایر اماکن، مراحل تولید درخواستهای تصادفی رانندگان از سیستم و همچنین آموزش شبکه عصبی سیستم را به شرحی که در ادامه آمده است انجام دادیم. در شکلهای ۵ و ۶ نمونه هایی از امکانات سیستم هدایت اتومبیل پیاده سازی شده مذکور را مشاهده می نمایید.



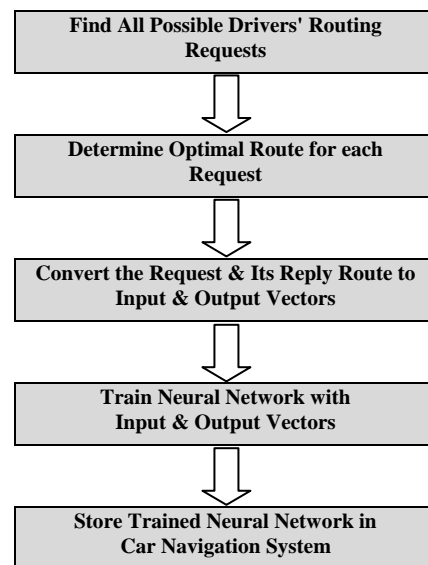
شکل ۵. امکانات سیستم هدایت اتومبیل پیاده سازی شده



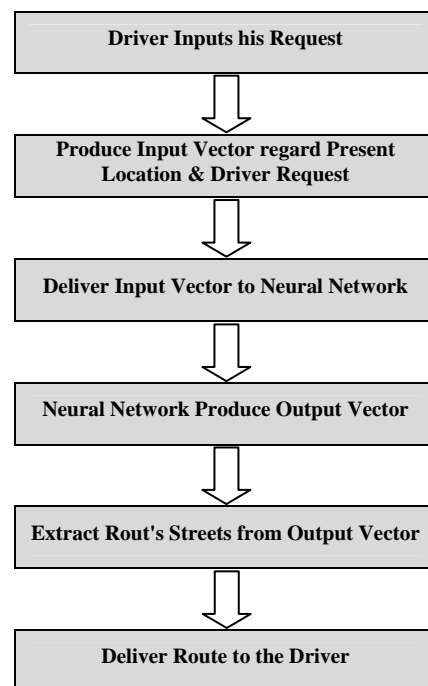
شکل ۶. صفحه نمایش سیستم هدایت اتومبیل پیاده سازی شده

خلال محاسبات تغییر نمی یابند و توابع محرک، روی تک تک نرونها عمل می کند.

در مسیر برگشت بردارهای حساسیت از لایه آخر به لایه اول برگشت داده می شوند عبارتی دیگر در مسیر برگشت شروع کار از لایه آخر یا لایه خروجی می باشد، جائیکه بردار خطا در اختیار می باشد. سپس بردار خطا از سمت راست به چپ از لایه آخر به لایه اول توزیع می شود [10].



شکل ۳. آموزش شبکه عصبی



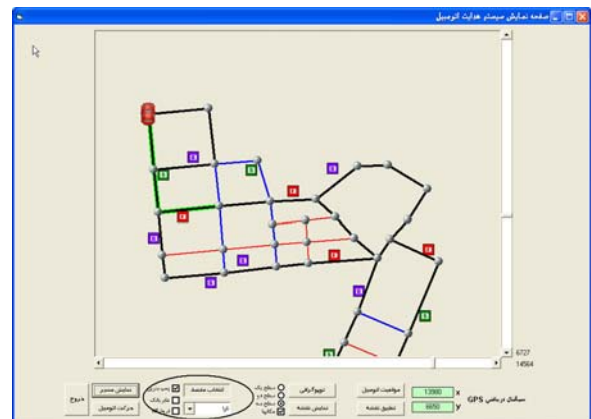
شکل ۴. نحوه کار سیستم هدایت اتومبیل

جدول ۲. نتایج براساس نرخهای یادگیری مختلف

درصد یادگیری	تعداد epoch	نرخ یادگیری
99.25%	60000	0.05
85%	60000	0.25
91%	60000	0.5

جدول ۳. نتایج براساس تعداد epochهای مختلف

درصد یادگیری	تعداد epoch	نرخ یادگیری
58%	10000	0.05
99.25%	60000	0.05
90%	70000	0.05



شکل ۷. نمایش توپوگرافی نقشه راهها در سیستم هدایت اتومبیل  
پیاده سازی شده

#### ۴- بحث و نتیجه گیری

اگر مشکلات مطرح شده در بخش دوم را در نظر بگیریم و اگر بخواهیم واقعا این سیستمها بصورت کارآمد به رانندگان در امر رانندگی کمک کنند، بناچار باید مسیرهایی را که در دنیای واقعی توسط رانندگان در قبال شرایط مختلف اتخاذ می شوند را در سیستم ذخیره نمائیم. و در موقع نیاز از آنها استفاده کنیم. مزایای روش پیشنهادی از چهار جنبه قابل بحث است.

نخست اینکه سایر الگوریتمهای استاندارد جستجو صرفا قادرند هزینه یا مسافت لینکها را در عمل جستجو در نظر گرفته و مسیر جواب را مشخص کنند. و فقط پیشرفتهای اندکی در مورد بروز کردن هزینه لینکها بهنگام تغییر شرایط ترافیکی حاصل شده است. ولی روش پیشنهادی ما اصلا از نظر تعداد پارامترها و معیارهای موجود برای عمل مسیریابی محدودیتی ندارد. و برای اضافه کردن هر معیار جدید کافی است یک مولفه به بردار ورودی اضافه بکنیم. که تنها آموزش شبکه را شاید طولانی تر بکند. که ربطی به زمان پاسخ شبکه بهنگام رانندگی ندارد. قابل ذکر است که تا بحال روشی ارائه نشده است که علاوه بر رسیدن به مقصد، بتواند پارامترهای مختلفی را نیز در نظر داشته باشد.

دوم اینکه همانطور که گفته شد با توجه به وجود محدودیتهای راهنمایی و رانندگی، که تعداد آنها نیز خیلی زیاد است، عملا پیاده سازی الگوریتمهای جستجوی گراف در دنیای واقعی امکانپذیر نیست. که روش پیشنهادی فارغ از این محدودیتهاست.

سوم سرعت بسیار بالای شبکه های عصبی بهنگام تولید پاسخ است. که در واقع اصلترین و مهمترین عامل که از شبکه های عصبی برای ذخیره کردن الگوها استفاده کردیم همین سرعت بالای آنها در تولید پاسخ می باشد.

چهارم اینکه اگر بخواهیم الگوهای ورودی و خروجی را در یک پایگاه داده ذخیره کنیم، اولاً بهنگام اجرای سیستم باید پایگاه داده در حافظه سیستم بار شود که به دلیل محدودیتهای سخت افزاری و بویژه

در نقشه استفاده شده تعداد ۱۰۰۰ تقاطع و ۲۰۰۰ خیابان را در نظر گرفتیم. ابتدا کلیه تقاطعهای شهر را با شماره های ۱ تا ۱۰۰۰ ذخیره کردیم. لذا برای نمایش تقاطعهای مبدا و مقصد تعداد ۱۰ بیت کافی می باشد ( $2^{10} = 1024$ ). خیابانهای شهر را نیز با شماره های ۱ تا ۲۰۰۰ مشخص کردیم. همچنین سه مولفه موجود بودن پمپ بنزین، بانک و فروشگاه زنجیره ای در بین مسیر مبدا تا مقصد را در نظر گرفتیم. فلذا ده مولفه از بردار ورودی را برای مشخص کردن نقطه مبدا، ده مولفه دیگر برای مشخص کردن نقطه مقصد و سه مولفه برای پمپ بنزین، بانک و فروشگاه زنجیره ای در نظر گرفته شدند. فرض کردیم که ۷۷ خیابان دارای سیستم مخابره اطلاعات ترافیکی باشند. پس بردار ورودی ما دارای ۱۰۰ مولفه خواهد بود. و بردار خروجی به تعداد کل خیابانهای شهر خواهد بود. فلذا بردار خروجی دارای ۲۰۰۰ مولفه می باشد. سپس با استفاده از برنامه کامپیوتری ۱۰۰۰۰۰ حالت مختلف را از روی اطلاعات موجود تولید کردیم. بدین ترتیب بردارهای آموزشی ما بدست آمدند.

در شبکه عصبی پس انتشار که دارای یک لایه مخفی می باشد، از تعداد ۱۰۰ نرون در لایه مخفی استفاده کردیم. فلذا ساختار شبکه عصبی ما دارای ۱۰۰ نرون در لایه ورودی، ۱۰۰ نرون در لایه مخفی و ۲۰۰۰ نرون در لایه خروجی می باشد. از پیشنهادات ارائه شده توسط Nguyen-Widrow [10] برای وزنهای آغازی استفاده کردیم. از Bipolar Sigmoid Function بعنوان Activation Function استفاده نمودیم. و یادگیری شبکه را برای ۱۰۰۰۰۰ الگوی ورودی و خروجی بر اساس نرخهای یادگیری مختلف و تعداد epochهای مختلف بررسی کردیم و برخی نتایج مطابق جداول ۳ و ۲ بدست آمد. که تعداد epochهای 60000 و نرخ یادگیری 0.05 در مورد 99.25% الگوهای تست شده جواب صحیح را تولید می کند.

برای میزان ترافیک راهها قائل شد. و برای اینکه دقت و کیفیت مسیر انتخابی را به حداکثر برسانیم، این حالت‌های مختلف و درصد‌های متفاوت میزان ترافیک راهها را به‌نگام آموزش شبکه عصبی سیستم هدایت اتومبیل در الگوهای ورودی متفاوت به سیستم وارد می‌کنیم. در نتیجه اگر سیستم مخابره وضعیت ترافیکی قادر به ارسال درصد ترافیک راهها نیز باشد، می‌توان بهترین پاسخ را با توجه به شرایط ترافیکی موجود و بر اساس درصد ترافیک خیابانهای دارای سیستم مخابره وضعیت ترافیکی توسط شبکه عصبی تولید کرد. بعنوان مثال اگر شرایط ترافیکی خیابانها را به پنج وضعیت 0% و 25% و 50% و 75% و 100% تقسیم کنیم و سیستم مخابره وضعیت ترافیکی نیز داده‌های ارسال خود را در پنج وضعیت فوق ارسال نماید، الگوهای آموزشی ما برای شبکه عصبی بصورت عناصر دوحالتی (باینری یا بایپولار) نخواهد بود. بعنوان مثال بردار

$$X[0.75, 0.0, 0.75, -1, -1, -1, +1, +1, -1, +1]$$

یک نمونه از بردارهای آموزشی شبکه عصبی می‌تواند باشد که سه مولفه اول آن بصورت نمایش بایپولار برای نقطه مبدا است، سه مولفه دوم آن برای نمایش نقطه مقصد می‌باشد و پنج مولفه آخر آن مربوط به پنج خیابان دارای سیستم مخابره وضعیت ترافیکی است که میزان ترافیک را بصورت حالت‌های پنجگانه ذکر شده ارسال می‌کند.

شبکه‌های عصبی که قادر به یادگیری و تولید پاسخ برای الگوهای نظیر بردار فوق هستند، تحت عنوان شبکه‌های عصبی فازی (Fuzzy Neural Networks) مطرح شده‌اند. و در برخی کاربردها مورد استفاده واقع شده‌اند. این شبکه‌ها از محدودیتها و مسائل بیشتری نسبت به سایر شبکه‌های عصبی برخوردار هستند. البته پیشرفتهای زیادی در این شبکه‌ها حاصل شده است و بسیاری از مسائل مربوط به آنها رفع شده است. فلذا کار بر روی این شبکه‌ها و آموزش حجم زیاد الگوهای آموزشی سیستم هدایت اتومبیل توسط این شبکه‌ها و بررسی نتایج بدست آمده می‌تواند سرآغاز تحقیقات تکمیلی در راستای روش پیشنهادی برای عمل مسیریابی توسط شبکه‌های عصبی باشد. بررسی ساختارهای مختلف شبکه‌های عصبی فازی و آزمایش قدرت یادگیری آنها برای الگوهای ورودی و سرعت آموزش شبکه‌ها و دقت و کیفیت پاسخهای تولیدی شبکه‌ها می‌تواند در بکارگیری شبکه‌های عصبی فازی در سیستم‌های هدایت اتومبیل تعیین کننده باشند.

## ۵- مراجع

- [1] C. Brenner and B. Elias, "Extracting landmarks for car navigation systems using existing GIS databases and laser scanning," ISPRS Archives, Vol. XXXIV, Part. 3/W8, Munich, Sept. 2003.
- [2] T. Nakamura, "Head start route selecting algorithm for reducing driver's waiting time," The Second World Congress on Intelligent Transport Systems, pp. 2031-2036, 1995.

کامبود حافظه و CPU که در این سیستمها وجود دارد، امکانپذیر نیست. ولی در مورد شبکه عصبی فقط یک ماتریس وزنی را در حافظه بار می‌کنیم. ثانیاً سرعت پاسخ شبکه عصبی در برابر جستجو پایگاه داده بسیار زیاد است. چرا که حجم اطلاعات در پایگاه داده بسیار زیاد خواهد بود و با افزایش یک تقاطع در شهر به اندازه تمامی حالات ممکنه در بین تقاطعها به تعداد رکوردهای پایگاه داده اضافه خواهد شد. در حالیکه در شبکه عصبی تا وقتی که تعداد تقاطعها از توان ۲ در نظر گرفته شده بالاتر نرود هیچ تاثیری در ابعاد ماتریس وزنی نخواهد داشت. مثلاً با در نظر گرفتن ۱۰ بیت برای مشخص کردن نقاط مبدا و مقصد در الگوی ورودی، تا ۱۰۲۴ تا تقاطع را می‌توانیم داشته باشیم و با اضافه شده ۱۰۲۵ امین تقاطع کافی است یک بیت به الگو اضافه کنیم. یعنی  $2^{11}$  که برای ۲۰۴۸ تا تقاطع کافی خواهد بود. و اضافه شدن یک مولفه به ماتریس وزنی تاثیر چندانی در زمان تولید پاسخ شبکه نخواهد داشت.

در واقع در روش پیشنهادی تمامی شگردها و تکنیکها را برای بدست آوردن بهترین مسیرها در هنگام یادگیری شبکه می‌توانیم بکار ببریم که هیچ ربطی به کاربرد Real Time این سیستمها ندارد. چرا که سیستمها با استفاده از این روش پاسخ مورد نظر را در کوتاهترین زمان ممکن تولید خواهند کرد. شاید بتوان گفت این اولین بار است که بحث استفاده از شبکه‌های عصبی در سیستم‌های هدایت اتومبیل مطرح شده است. فلذا می‌تواند سرآغاز تحقیقات گسترده در این زمینه باشد. کارهای آتی در راستای بهبود دقت شبکه عصبی بکار برده شده خواهد بود. و همچنین استفاده از شبکه‌های عصبی فازی می‌تواند بعنوان ایده دیگری در این سیستمها مطرح شود که ذیلاً توضیحاتی در این زمینه آورده شده است.

## درصدهای مختلف ازدحام ترافیکی راهها در عمل مسیریابی و شبکه‌های عصبی فازی:

در روش مطرح شده برای دخالت دادن شرایط ترافیکی راهها، در بردار ورودی صرفاً قادر هستیم که تعیین کنیم یک خیابان دچار ازدحام شده است یا خیر؟ که این دو حالت را با مقادیر بایپولار +۱ و -۱ مشخص می‌کنیم. این داده‌های دودویی (بایپولار یا باینری) بهترین حالت ممکنه برای آموزش شبکه‌های عصبی می‌باشند. چرا که ساختار شبکه‌های عصبی مختلف الگوهای ورودی بصورت دودویی را با سرعت بالایی آموزش می‌بینند و همچنین تولید جوابهای صحیح نیز در این گونه از نمایش بردارها در شبکه‌های عصبی از کیفیت بالایی برخوردار است.

در دنیای واقعی به‌نگام انتخاب مسیر بهینه توسط رانندگان شاید بهترین مسیری که انتخاب می‌شود بر اساس درصد میزان ترافیک خیابانها باشد. عبارتی دیگر میزان ترافیک راهها همیشه دو حالت ازدحام یا بدون ازدحام نیست. بلکه می‌توان درصد‌های مختلفی را

- [3] Hashemzadeh, M. and Meybodi, M. R., "A fast and efficient route finding method for car navigation systems with neural networks," Proc. of the the Tenth Int. IEEE Enterprise Distributed Object Computing Conference(EDOC 2006), pp. 423-426, 2006
- [4] R. E. Korf, "Real-Time heuristic search," Artificial Intelligence, Vol. 42, No. 2-3, pp.189-211, 1990.
- [5] H. Hiraishi, H. Ohwada and F. Mizoguchi, "Intercommunicating car navigation system with dynamic route finding," Proc.of IEEE/IEEJ/JSAI Int. Conference on Intelligent Transportation Systems, pp.284-289, 1999.
- [6] I.Chabini and S.Lan, "Adaptation of the A\* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks," IEEE Trans. Intelligent Transportation Systems, Vol. 3, pp. 60-74, Mar. 2002.
- [7] E. P. F. Chan and N. Zhang, "Finding shortest paths in large network systems," Proc. of the 9th Int. Conference on Advances in Geographic Information Systems, ACM Press New York, NY, USA, 2001.
- [8] I. Flinsenberg, "Graph partitioning for route planning in car navigation systems", Proc. of the 11th IAIN World Congress, Smart Navigation Systems and Services , Berlin, Germany, Oct. 2003
- [9] K. Kim, Y. Seungwon, and K. C. Sang, "A partitioning scheme for hierarchical path finding robust to link cost update," Proc. of the 5th World Congress on Intelligent Transportation Systems, Seoul, Korea, Oct. 1998.
- [10] L. Fausett, "Fundamentals of neural networks, Architectures, Algorithms, and Applications," Prentice Hall, Upper Saddle River, New Jersey 07458, © 1994.
- [11] X. G. Wang, Z. Tang, H. Tamura, M. Ishii and W. D. Sun, "An improved backpropagation algorithm to avoid the local minima problem," Neurocomputing, Vol. 56, pp. 455-460, 2004.
- [12] K. Eom, K. Jung, H. Sirisena, "Performance improvement of backpropagation algorithm by automatic activation function gain tuning using fuzzy logic," Neurocomputing, Vol. 50, pp. 439-460, 2003.
- [13] M. Mandischer, "A comparison of evolution strategies and backpropagation for neural network training," Neurocomputing, Vol. 42, pp. 87-117, 2002.
- [14] B. Walczak, "Neural networks with robust backpropagation learning algorithm," Analytica Chimica Acta, Vol. 322, pp. 21-29, 1996.
- [15] G. D. Magoulas, M N. Vrahatis, G. S. Androulakis, "Effective backpropagation training with variable stepsize," Neural Networks, Vol. 10, No. 1, pp. 69-82, 1997.
- [16] A. V. P. Espinoza, J. B. O. Mere, F. J. M. D. Pison, A. G. Marcos, "TAO-robust backpropagation learning algorithm," Neural Networks, Vol. 18, pp. 191-204, 2005.