Applying Adaptive Network-Based Fuzzy Inference System to Predict Travel Time in Highways for Intelligent Transportation Systems

Rouhollah Maghsoudi1, Behzad Moshiri2

1) Department of Computer, Mahmudabad Branch, Islamic Azad University, Mahmudabad, Iran
2) Control and Intelligent Processing Center of Excellent, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

rcemaghsoudy@gmail.com; moshiri@ut.ac.ir

Received: 2016/06/09; Accepted: 2016/08/15

Abstract

Travel time is a good criterion in analyzing transportation systems. Advanced systems of collecting traffic data (e.g. loop detectors, video cameras ...) are now collecting and storing daily status of traffic throughout the world. There are two ways to calculate travel time: direct measurement, and prediction. Several classic statistical ways have been used to predict travel time, but when nonlinear nature is focused, developing a proper model with multiple linear will be a failure. This means that when data have a nonlinear inherent, using of linear methods such as some statistics methods will not be benefit and will not generate appropriate results. Meanwhile, ANN and ANFIS are nonlinear tools. Intelligent systems approaches such as artificial neural networks (ANN) and recently neuro-fuzzy have successfully appeared in prediction. In most applications of ANN, multilayer perceptron (MLP) is applied which is trained by the algorithm of back propagation error. The main problem of this approach is that it is hard to interpret the knowledge in the trained networks. Applying neuro-fuzzy approach, information saved in trained networks will be defined within a fuzzy data base. The aim of present research is to offer a strong neuro-fuzzy network and apply it to predict travel time and compare its results with methods like ANN and AIMSUN. Our results indicate that means for neuro-fuzzy prediction remarkably decrease the error criteria of predicted travel time. This research proves the possibility of applying adaptive neuro-fuzzy inference system in predicting travel time, and reveals that it can make very successful analysis on traffic data. To study credibility of prediction results, AIMSUN traffic simulation software as an expert analyst, was applied and freeway travel time was studied and calculated by simulation.

Keywords: Predicting Travel Time, Intelligent Transportation Systems, AIMSUN, ANFIS, Artificial Neural Networks (ANN)

1. Introduction

In line with implementing advanced passenger information systems and (highways and urban) traffic management systems in recent years, it has been increasingly important for both traffic analysts and passengers to measure traffic parameters accurately and numerically. These parameters are travel time, traffic flow, speed, density and so on. Travel time is a great factor among traffic data to describe this status, and broadly used in congestion measurement and path finding systems by traffic control centers in order to
control and set traffic time. Besides traffic managers, the travel time information can be helpful to other people. It helps passengers to select proper paths, and assists laboratories and all pickup and delivery companies to decrease delivery expenses and increase certainty. Estimating travel time depends directly on vehicle speed, traffic flow and road occupancy coefficient and traffic events. These features have made accurate and optimum prediction of travel time so complicated and hard. Right now, traffic managers and passengers plan for travel time and management according to present traffic situation, and follow less the predicted traffic situation in future. The ability to predict travel time accurately in highways and freeways is a basic component for many applications of ITS as advanced traffic management systems, adaptive ramp metering, and passenger information. The predicted travel time is advised to passengers through mass media (for example, giving information through SMS, radio waves, or information boards on freeways). A very important output from this prediction process is to deliver some options of travel path to passengers to prevent congestion as much as possible. During the last two decades, the need to predict traffic parameters has led surveys to develop and create numerous algorithms in this field [2]. Yan Ying Li stated the main aspects of predicting traffic components in a table. One aspect is the resources of data collection including loop detectors or similar sensors, automatic vehicle identification (AVI), data combined by loop detector, and GPS equipped device. Prediction techniques are time series models, macroscopic and microscopic simulation (by AIMSUN), parametrical models, artificial intelligence techniques such as neural networks, fuzzy logic (here we used a very strong neuro-fuzzy combination called ANFIS). One resource of collecting data from roads is loop detectors. This technology has been used since 1970 for identification of vehicles.

A loop detector can collect traffic flow, road occupancy coefficient, and speed in its area. There are many loop detectors installed on roads now including urban, highway and freeway. Afterwards, various models have been developed to predict travel time according to loop detectors data. Su Hai-bin (2007) called these models as point measurement methods on his articles [4]. The other way is direct measurement of travel time in which video cameras are installed in departure and destination of a path segment. Vehicle plate number is saved once it passes a station and identifies by an image processing software. If the number saved by departure station camera matches the saved number with destination camera, time travel can be calculated and saved. The average of travel time in a same time span is deemed as travel time in that segment [3]. Exception researchers that applied ANN to predict travel time [4,9,10,11,12,13], some investigators used from fuzzy inference systems and neuro-fuzzy approach in predicting travel time and analyzing traffic states [5,6,7,8,23,24,25]. Our work has two differentiations with last similar researches. First is that ANFIS has not used in travel time prediction until now.

And second is that our training algorithm for ANFIS is hybrid (combination of least square and back propagation) that explained in following sections. The rest of the article is organized as follows: in section 2, prediction methods such as ANN and ANFIS have been analyzed, and then traffic simulation software (AIMSUN) is described. In section 3, case study and data collection area is expressed. And finally Simulation results for prediction methods are presented in Section 4. Section 5 concludes the paper.
2. Prediction Methods and Instruments, and Error Criterion

Neuro-fuzzy network

A neuro-fuzzy network consists of two main parts: a) neural network, and b) fuzzy system. The combination of these two segments leads to a neuro-fuzzy network. Here, we describe the concepts of a neuro-fuzzy network.

2.1 Basics of Artificial Neural Networks

The structure of ANN was inspired by neuron cells of body. Each neuron consists of cellular body, axon, and dendrite. These three parts form a nerve cell with the ability of sending and receiving information. The motif of modeling a biological nerve net has led to an ANN. A simple structure of an artificial nerve networks is shown in figure 1.

Input data enter the network under parameter (p). Then they are crossed with a series of synaptic weights, and added to bios (b). These values (weights and bios) are first random. Let me describe the random term. Neural networks are intelligent methods. If to be concentrated to figure 1, weights as effective parameters multiplied with inputs and added to biases. These parameters (means weights and biases) generate in ANN in [0, 1] (usually) randomly and then during training process reach to their optimal values. Not only in this model of ANN but also in most models of ANN, initial values such as weights are stochastic. This statement is inherently in ANN, Of course if these values aren’t appropriate, can be optimized with evolutionary algorithms such as GA\(^1\), PSO\(^2\) and then start ANN training process with more appropriate values. In our paper, we started with random values that led to desired results.

Next, the found value is delivered to transmission or activation function (f) with an output shown as (a). The output of each layer is in fact an input for next layer. ANNs include several active layers with various neurons in each. Finally, the ultimate output is compared with a desirable value and the found error rate is spread through variables w and b of the network. The training process continues until error criterion reaches near zero. After accomplishment of network training, it can be used as a functional and applied model [8, 9, and 17].

---

\(^{1}\) Genetic Algorithms

\(^{2}\) Particle Swarm Optimization
2.2 Fuzzy System

Each fuzzy system consists of three main parts: 1) fuzzifier, 2) fuzzy knowledge base, and 3) defuzzifier. First, input data change into fuzzy state from definite state (crisp) after passing fuzzy maker. The process is done by membership function. Then, fuzzy parameters enter fuzzy data base. It includes fuzzy rules base, and inference engine. In fuzzy rules base, fuzzy assumptions-related rules are described. Next, analysis and reasoning are performed by inference engine. Among fuzzy inference engines, Sugeno and Mamdani are popular and applied ones. Operator of each of these engines can be minimum or product. Equivalents (1) and (2) indicate Sugeno model with product and minimum operators.

\[
\text{output}(x) = \frac{\sum_{l=1}^{M} F_l(x) \left( \tilde{\mu}_{A_l} (x_i) \right)}{\sum_{l=1}^{M} \tilde{\mu}_{A_l} (x_i)}
\]  

(1)

\[
\text{output}(x) = \frac{\sum_{l=1}^{M} F_l(x) \left( \min_{i=1}^{n} \mu_{A_l} (x_i) \right)}{2 \sum_{l=1}^{M} \min_{i=1}^{n} \mu_{A_l} (x_i) a}
\]  

(2)

\[
F_l(x) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n \quad \text{(First order)}
\]  

(3)
Equations 1, 2 and 3 are almost equivalent to figure 3. Inputs pass membership functions($\mu A_i(x_i)$) and product together($\prod_{i=1}^{n} \mu A_i(x_i)$) and normalized($\frac{\prod_{i=1}^{n} \mu A_i(x_i)}{\sum_{i=1}^{n} \mu A_i(x_i)}$) and then CONCLUSION PART of a rule is generated ($\sum_{i=1}^{M} F_i(x)$) (as you know, each rule in fuzzy systems has two parts: ANTECEDENT(if) PART & CONCLUSION(then) PART. for example: Rule1: if x is $A_1$ and y is $B_1$, then $f_1 = p_1x + q_1y + r_1$) and finally finds ultimate output of Fuzzy Inference System by summation of pre-output signals($\sum_{i=1}^{M} F_i(x)$). Description of how equations obtained, was explained with more details in page 8-10.

Where $F_i(x)$ is the first order of $i^{th}$ rule. So output value of fuzzy system is transmitted to defuzzifier, and changes into crisp value. Figure 2 shows a simple fuzzy system [15].

**2.3 Adaptive Network-Based Fuzzy Inference System (ANFIS)**

Diversity of rules in fuzzy system and their equal membership functions increasingly depend on our knowledge about system. Right now, there is no systemic way to change the human knowledge into a fuzzy system. By combining ANN learning ability with logical function of fuzzy systems, we can establish neuro-fuzzy networks and use them to determine fuzzy system parameters such as membership functions and rules members. ANFIS applies a logical system named fuzzy logic to calculate hidden uncertainties in data, and then an accurate writing is made on its basis.

It is performed by input fuzzification through membership functions which map a curved relation of input value with (0, 1) span. The parameters attending the input are acquirable like output membership functions with the help of an algorithm such as back propagation or the least squares. Then unlike MLP where weights are updated, in ANFIS the fuzzy lingual rules or if-then statements are specified for training. Figure 3 shows general structure of ANFIS.

Selecting a fuzzy inference system (FIS) is the main problem in planning ANFIS system for modeling a certain goal system. Various kinds of FIS are presented in articles and each one of them is only defined by parameters of following section. The FIS following section is a linear equation and its parameters can be estimated by a simple error of the least squares method.
For example, imagine a FIS with two $x, y$ inputs and a $z$ output. The following typical rules with fuzzy if-then rules can be applied to define a first order Sugeno fuzzy model [7, 22].

Rule1: if $x$ is $A_1$ and $y$ is $B_1$, then $f_1 = p_1x + q_1y + \eta_1$ \hspace{1cm} (4)

Rule2: if $x$ is $A_2$ and $y$ is $B_2$, then $f_2 = p_2x + q_2y + r_2$ \hspace{1cm} (5)

Whereas $A_1$, $A_2$, $B_1$, $B_2$ are membership functions of $x$ and $y$ inputs respectively. $A_1$, $A_2$, $B_1$, $B_2$ are fuzzy sets that to be considered as values for inputs $x$, $y$. this means $\mu_{A_i}(x)$ and $\mu_{B_i}(y)$. $\mu_{A_i}(x)$ is membership degree of input $x$ to set $A_i$ and also for $y$. $A_1$, $A_2$, $B_1$, $B_2$ are membership functions that can be divided into some categories such as Gaussian, Triangle, Trapezoid, General Bell Formed. In below, for example $A_1$ and $B_1$ are trapezoid. $A_2$ is Gaussian and $B_2$ is gbell.
In fact, \( A_i \) and \( B_i \) convert crisp values to fuzzy kind. \( p_1, q_1, r_1, p_2, q_2, \) and \( r_2 \) are output function parameters. An ANFIS-like architecture is shown in figure 3. Each layer nodes have similar functionality. ANFIS operation is as follows:

**Layer 1:** Every node in this layer produces the membership degree of an input parameter, i.e., it defines fuzzy partitions on input space.

\[
op^1_i = \mu_A(x) \quad \text{for} \quad i = 1, 2 \]
\[
op^1_i = \mu_B(y) \quad \text{for} \quad i = 3, 4 \quad \tag{6}\]

Whereas \( x \) or \( y \) is the input to each node. \( A_i \) or \( B_{i-2} \) is a fuzzy set participating with this node and is defined by form of membership functions. They can be any appropriate, continuous, and piecewise differentiable function such as Gaussian, general bell formed, triangle and trapezoid.

Suppose a general bell form function as a membership one. \( \op^1_i \) output can be calculated as follows:

\[
op^1_i = \mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \quad \tag{7}\]
\[
\mu_{B_{i-2}}(y) = \frac{1}{1 + \left| \frac{y - c_i}{a_i} \right|^{2b_i}} \quad \tag{8}\]

Where \( \{a_i, b_i, c_i\} \) are a series of parameters that change the form of membership function. This change can have minimum value of 0 and maximum value of 1.

**Layer 2:** Each node products input signals of this layer to each other. It is shown by \( \pi \). \( \op^1_i \) which is the fire power of a base, is found as follows:

\[
op^2_i = w_i = \mu_A(x) \cdot \mu_B(y), \quad i = 1, 2 \quad \tag{9}\]

More generally speaking, this layer is operation of \( T\)-norm operators such as product or minimum.

**Layer 3:** With \( I^s \) node in this layer (shown by \( N \)), normalization of input signals is performed:

\[
op^3_i = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad \tag{10}\]

And it is evaluation of each rule left side.

**Layer 4:** \( I^s \) node in this layer finds the participation rate of each rule in model output with the below function:

\[
op^4_i = \bar{w}_i \cdot f_i = \bar{w}_i \cdot (p_1 x + q_1 y + r_1) \quad \tag{11}\]

Where \( \bar{w} \) is layer 3 output, and \( \{p_i, q_i, r_i\} \) is parameters set [5,16,22].

**Layer 5:** this layer finds ultimate output of ANFIS by itself in the following way:
2.4 Adjusting Parameters and Training ANFIS

Effective parameters on neuro-fuzzy ANFIS network structure are \{a_i, b_i, c_i\} which define the nature and form of membership functions and the parameters of following section i.e. \(p_i, q_i, r_i\) which define general output of the system. The learning algorithm which is used in the present study for ANFIS is a hybrid algorithm, which is a combination of gradient descent and the least-squares method. More specifically, in the forward pass of the hybrid learning algorithm, node outputs go forward until layer 4 and the consequent parameters are identified by the least-squares method. In the backward pass, the error signals propagate backwards and the premise parameters are updated by gradient descent. The consequent parameters are optimized under the condition that the premise parameters are fixed. The main benefit of the hybrid approach is that it converges much faster since it reduces the search space dimensions of the original pure Back-propagation method used in neural networks. The overall output can be expressed as a linear combination of the consequent parameters.

One of the factors of ANFIS architecture design is the number of membership functions considered for each input. Various parameters for planning ANFIS are shown on following table.

<table>
<thead>
<tr>
<th>Training algorithm</th>
<th>Time rules number</th>
<th>Type of membership function</th>
<th>Number of function</th>
<th>Number of Nonlinear parameters</th>
<th>Number of linear parameters</th>
<th>Number of network nodes</th>
<th>Number of outputs</th>
<th>Number of inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid</td>
<td>64 or 125</td>
<td>Gaussian</td>
<td>4 or 5</td>
<td>30</td>
<td>500</td>
<td>286</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

These parameters are features of any fuzzy model specially ANFIS. For simulation of ANFIS, we use from GENFIS tool in MATLAB. Some parameters are generated by means of GENFIS automatically (number of non-linear parameters, number of linear parameters …) but some of them can be handled by programmer user (training algorithm, type and number of membership functions \[MF\], number of inputs and outputs that is depended to problem …). These parameters values are effective in ANFIS performance.

For instance, selecting (or creating) a good training algorithm can reduce error curve considerably. Our training method is combined of Least-square and Back-propagation. Or \[MF\] type could be varied according to the type of problem. Usually user can achieve appropriate type of \[MF\] with several runs. Gaussian has often performed better than other type. Or the number of rules is equivalent to \(\text{number of MF} \times \text{number of inputs}\). In this paper, we have three inputs and four or five function. Therefore 64(\(4^3\)) or 125(\(5^3\)) rules are generated. As mentioned in abstract, Difference between ANN and Neuro-fuzzy (such as ANFIS) is reasoning or inference strength. Reasoning is based on rules. If rules number becomes few, then we will have weak reasoning and if becomes much then we will have a time consuming and slow reasoning. So rules number must be stabilized.
2.5 Credibility of Prediction Results by Traffic Simulation Software

Traffic simulation is vastly applied in transportation engineering. Some reasons to apply these models are economic and proper productivity, safe nature of simulation, vast applicable areas such as planning transportation systems, traffic flow operation, and evaluating management options. Basically, when pure mathematical models are not applicable for total system because of complicated relation between parts of a system and various factors of creating uncertainty, understanding and implementing the simple relations among the parts as a system can be a useful solution in analyzing these complicated systems. Our purpose is describing capability and benefit of simulation in several fields of engineering, for example: transportation and traffic systems. The best benefit of simulation is that engineers can analyze projects before constructing or operating and find strength or weakness points and relationship between different components of projects.

![Figure 5. A typical picture of a traffic simulator [20]](image)

Each simulation software has two type of outputs: statistical and graphical ones. Statistical outputs show quantitatively the probable function of the system within evaluation factors such as travel time, total travel time, average travel speed, number of vehicles passed the network, and total passed area, and so on. Graphical output and animation shows the events occurred in the system including formation of queue, congestion, and blockade in the area to the expertise.

Concerning software and hardware progresses during recent decades, applying simulation models in planning urban transportation has also been developed. Various traffic software such as CORSIM, SYNCHRO, SimTraffic, Vissim, Paramics, and GETRAM/AIMSUN are applied now and each has some advantages and disadvantages. Finding the results in existing simulation software’s is done by statistical rules. A number of cars enter in such software by using random digits and by tracing each vehicle, the
rates of several parameters are found. Then the output results are determined by averaging and using statistical rules. Random state is well understood in simulation results, but unfortunately they are not considered in the time of traffic simulation. The common way to avoid incorrect results in simulation is using various simulations (or replications) and averaging the considered criteria. Although the number of required replications is not defined accurately, the research, after ten operations or replications were averaged out so that less responses with error would be produced.

Simulation is broadly applied in traffic engineering. Appearance of new ideas in management, traffic control, and increased subjects in this area, are main reasons for daily increasing use of simulation software. The user of traffic simulation software presents the way of setting, features of network elements, and required information for traffic demand to the model within a scenario. Although the software has numerous positive capabilities, it should be noted that such models are not substitutes for planning, capacity estimating, and demand modeling patterns, but they are their protectors. Generally the simulation software output is available spatially and timely on a piece or segment level or in a certain time span. Contrary to these results, is equally extracted on total network or in total simulation time [19, 20]. This means that Output of traffic simulation software is available both in level of a segment and a special time range. In other word, output is extractable from two approaches: space and time.

2.5.1 AIMSUN Traffic Simulation Software

AIMSUN is integral simulation software for modeling traffic and transportation systems. It is the only software in the field of traffic engineering which presents the following transportation models in 3 levels within a software. Devices to allocate static traffic, new Mesoscopic simulator, Microscopic simulation software are various usage of AIMSUN which are used at universities and transportation research centers in order to improve road and transportation infra-structures. The cases are:
- Analyzing the effect of infrastructural projects
- Environmental researches
- Optimizing lights control program
- Managing city and road traffic
- Planning pedestrians network and supporting systems for public transportation management
- Analyzing security
- Evaluating intelligent transportation systems
- Developing new transportation algorithms and models

Although simulation process facilitates modeling and analyzing traffic flow in network, the requirements to use this strong device should also be considered. One of the requirements is validation of each simulation. The other requirements of a successful simulation is adapting applied treatment models with user’s treatment from pedestrian’s network within studied area. Therefore, by making statistics in highways, the simulation results with true statistics shall be calibrated, i.e. two main steps must be considered in the process of traffic simulation.

a) Validating: The simulation results reveal the true function of the model.

b) Calibrating the model: The model indicates true function of the system for which it has been made.

As explained above, one repetition in a random process can not represent a statistical society. Experience has shown that in simulating a certain model, sometimes thirty
repetitions or more reach us 19% error, and sometimes in the same model we achieve 2% error with ten repetitions. Hence, a fixed number of repetitions can not be deemed as out task basis, and in all simulations we should study the validation of simulation output results [19].

2.6 Error Criterion

To evaluate the performance of prediction made by above models, an error criterion called mean square error was used. As you know, we have some error measurements such as MSE, RME, MAE, MARE, etc. we selected MSE because it has power 2 (is nonlinear) and has more accuracy. Our second reason is its utilization in many journal papers.

\[ \text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (T_i - \hat{T}_i)^2 \]  \hspace{1cm} (13)

Where \( T_i \) and \( \hat{T}_i \) is ith desired and predicted output respectively and \( n \) is the total number of pairs (inputs–outputs) of data in the training set.

3. Data Collection and Area Under Survey

Developing a model to predicate requires appropriate data collection. These data are divided into two groups: Model development data, and Model validation data. The former is used to make a special model for suggested methods. The latter is used to confirm accuracy of model yield. Speed, occupation coefficient, and flow are the main parameters in this survey, 70% of which are used for development, and 30% for testing and confirming accuracy of network yield. Among Minnesota state highways, the 494 freeway was chosen for survey. Minnesota road paths are equipped with numerous detectors with 1 km distance from each other. They are able to collect traffic data including speed, occupation coefficient, and number of vehicles which are collected during 24 hours by sampling rate of one minute. The specific places and detectors distance are shown in figure 5. Data about travel time are found by vehicle speed and certain distance between detectors. The collected data are for January 1\textsuperscript{st} and 2\textsuperscript{nd} 2009. For more accurate prediction, the richer data had better be used. Travel demand is usually the most 6-9 a.m and 4-7 p.m. The peak time is 8-9 a.m and 6-7 p.m. Figure 2 and 3 show travel time diagrams in rush hours. The prediction process was done at over demanding hours.

Data from 6-8 a.m and 8-9 a.m are used for training and testing sets respectively. Simulation with AIMSUN was performed hourly (8-9 a.m), and travel time was extracted as a simulation output parameter [21].
4. Simulations Results

Based on neuro-fuzzy network parameters (of course, must be said that the best values of these parameters were obtained from different runs of ANFIS) mentioned on section 3, travel time on freeway 494 was predicted and their results were compared. Desired travel time in below figures is real travel time that we had in Minnesota dataset beforehand. In other word, ANN and ANFIS training method is supervised. This means that we have correct outputs previously. Our produced output compare with desired and then is tried to minimize error with learning algorithms. The curves of figures 7 show that ANFIS follow desired curve more accurate and has low chattering rather than MLP and AIMSUN. Results of table 2 indicate $\text{mse}$ of 2 prediction models per test data for each one of 7 sections. These errors reveal that ANFIS decreases $\text{mse}$ significantly compared with MLP and AIMSUN.
(a) Predicted travel time for station 702

(b) Predicted travel time for station 703

(c) Predicted travel time for station 704
Applying Adaptive Network-Based …

R. Maghsoudi, B. Moshiri

Figure 7. The comparison of predicted travel times using several predicting methods.
In below tables, several prediction methods has compared together from two aspects, accuracy and time.

Table 2. Prediction results of test data for each of sensory station

<table>
<thead>
<tr>
<th>Source Station</th>
<th>Destination Station</th>
<th>MLP Predictor</th>
<th>ANFIS Predictor</th>
<th>AIMSUN Simulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>S 702</td>
<td>S703</td>
<td>1.4295×10-6</td>
<td>1.6339×10-11</td>
<td>3.1663×10-5</td>
</tr>
<tr>
<td>S 702</td>
<td>S704</td>
<td>6.0849×10-6</td>
<td>5.7136×10-12</td>
<td>3.6348×10-5</td>
</tr>
<tr>
<td>S 704</td>
<td>S705</td>
<td>8.4721×10-6</td>
<td>2.5439×10-11</td>
<td>1.6006×10-5</td>
</tr>
<tr>
<td>S 705</td>
<td>S706</td>
<td>6.9571×10-6</td>
<td>3.5069×10-09</td>
<td>2.3194×10-5</td>
</tr>
<tr>
<td>S 707</td>
<td>S708</td>
<td>5.8682×10-6</td>
<td>4.3741×10-10</td>
<td>2.2783×10-5</td>
</tr>
</tbody>
</table>

Table 3. Comparison of three methods from time aspect

<table>
<thead>
<tr>
<th>Source Station</th>
<th>Destination Station</th>
<th>MLP Time(seconds)</th>
<th>ANFIS Time(seconds)</th>
<th>AIMSUN Simulator Time*(seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S 702</td>
<td>S703</td>
<td>31.846921</td>
<td>5.205952</td>
<td>20</td>
</tr>
<tr>
<td>S 703</td>
<td>S704</td>
<td>29.190070</td>
<td>7.978476</td>
<td>19</td>
</tr>
<tr>
<td>S 704</td>
<td>S705</td>
<td>29.293130</td>
<td>7.968780</td>
<td>20</td>
</tr>
<tr>
<td>S 705</td>
<td>S706</td>
<td>30.003509</td>
<td>7.960191</td>
<td>21</td>
</tr>
<tr>
<td>S 706</td>
<td>S707</td>
<td>29.586304</td>
<td>8.509871</td>
<td>20</td>
</tr>
<tr>
<td>S 707</td>
<td>S708</td>
<td>29.206483</td>
<td>7.968708</td>
<td>18</td>
</tr>
</tbody>
</table>

5. Discussion and Conclusion

Neuro-fuzzy network ANFIS proved its success and capability in analyzing and predicting several data. However, little works have been done in traffic data analysis using neuro-fuzzy networks. In this paper, the capability of ANFIS in predicting travel time has been examined. If you pay attend to table 2, you will observe that distance between MLP and AIMSUN errors with ANFIS errors is very much (for instance: from s702 to s703, MLP error is 1.4295×10^-6 and AIMSUN error is 3.1663×10^-5 when ANFIS error is 1.6339×10^-11). Second notice is prediction time of ANFIS that is very fewer than MLP and AIMSUN. In other words, ANFIS is a fast estimator. You see the comparison of MLP, ANFIS and AIMSUN from time approach (speed) in table 3. Character * in table 3 want to say this point that In AIMSUN, simulation time is depended to number of replications. If we want to have a good simulation, a scenario must be simulated at several replications (this means that our time will be more. in addition to, length of road, volume of input data (occupancy of road …) are effective in this time. Meanwhile, these parameters doesn’t effect on ANFIS or MLP performance). AIMSUN time has been obtained for only one replication (of course in above table). If the volume of data becomes huge, then difference of three prediction time will be much more considerably. We can compare these methods from some aspects and in this paper has been concentrated on two aspects: error criteria & time.

After numerous running, a set of ANFIS parameters have been proposed which can estimate travel time very well. Results indicate that ANFIS predictor has considerably
worked better than other predicting methods. Because it combines the two approaches, neural networks and fuzzy systems. Combining these two intelligent approaches, good reasoning is achieved in quality and quantity. In other words, both fuzzy reasoning and network calculation will be available simultaneously. This means, ANFIS is a gray-box. Perhaps this article becomes a starting point in traffic data analysis applying of an intelligent approach such as ANFIS and comparing with traffic simulation software such as AIMSUN.

References


[23] Samer Elabd, and Andreas Schlenkhoff,,: ‘ANFIS and BP Neural Network for travel time prediction’, World Academy of Science, Engineering and Technology, 57, 2009

