Development of Various Artificial Neural Network Car-Following Models with Converted Data Sets by A Self-Organization Neural Network

Mitsuru TANAKA [a]

[a] *Rummel, Klepper & Kahl, LLP, Wilmington, DE, 19801, USA*
[a] *E-mail: mitsurude@msn.com*

Abstract: Four car-following models with artificial neural network (ANN) structure were developed with various input variables in the car-following behavior. A four-layer ANN structure was set up and a genetic algorithm (GA) and back-propagation methodology were utilized for determining the synaptic weights in the models, however the models sometimes had a difficulty in learning such enormous number of raw data points. Therefore, a methodology of data point conversion was developed with, Kohonen Feature Map (KFM), a self-organization neural network model. In order to evaluate the ANN models, the General Motors' (GM) model was also calibrated. This paper concluded that the ANN models were successfully developed with KFM data conversion without deteriorating the original data quality. In comparing the results among the four ANN models, it was implied that the accelerations of the following vehicle and leading vehicle can also become key input variables for improving the modeling of car-following behavior.

Keywords: Car-Following Model, Artificial Neural Networks (ANN), Back-propagation, Kohonen Feature Map (KFM), Genetic Algorithm (GA), Data Sampling

1. INTRODUCTION

Various microscopic car-following models have been developed to represent vehicular movements in the microscopic level of traffic flow in the past half century. The car-following models have generally been modeled based on experiments observing variables of two vehicles called a leading vehicle, LV, and a following vehicle, FV. It started with a very simple mathematical model, that utilized a linear relationship between spacing between a leader and a follower, and a desirable speed of the follower. Since then, the car-following models have become increasingly complex, with extra input variables and specific predefined rules in newer proposed models. Developing such a complicated car-following model is normally time-consuming and sometimes there is difficulty in calibration with the predefined rules. The models and rules are predetermined by the modelers, however they may not accurately represent human driving behavior as there may be some key factors missing from the natural driving behavior in the models. In order to better present driving behavior, it may be more effective to have a car-following model learn the human driving behavior through the car-following traffic variables from the observed data sets. The artificial neural network (ANN) model has such a structure for building an input-output system throughout the learning process from training data sets.

Mathematical models of ANN have been developed from the structure and functional aspects of biological neural networks. ANNs are adaptive systems that change structures based on external or internal information that flows through the network during the learning phase. Therefore, ANNs have been often used for non-linear statistical data modeling. Car-following behavior is a human behavior and a driver learns from experience utilizing the external information surrounding the driving vehicle. It is expected that the ANN structure has great potential to model car-following behavior. There are some shortcomings of using ANNs and one of them is the long computation time during the training process due to its heavy repeating process in the data learning algorithms. However, in the past several decades, the computer technologies have been significantly improved and computation time for processing complicated computer algorithms has significantly shortened. Thus, the computation time of heavy repeating processes for learning data sets in ANN has significantly reduced as the computer technology has improved. Therefore, ANN model structure was applied for building car-following models in this paper. Multiple ANN car-following models were developed with different input variables and they were compared each other for evaluating the importance of the model input variables among them. The proposed ANN models were also compared to a well-known existing car-following model in order to evaluate how the ANN structures are valid and feasible to apply as car-following models in this study.

2. BACKGROUND

2.1 Car-Following Models

Microscopic car-following models began with a simple linear relationship between driving speed and distance headway with the vehicle running in front of the subject vehicle introduced by Pipes (1953). Forbes and Zagorsk (1958) introduced the same car-following model, but used speed and time headway instead of space headway. After these speed-distance models, researchers at General Motors Company (GM) established a series of car-following models called stimulus-response system. In this car-following system, stimulus is represented by relative speed between a leader and a follower and response is represented by acceleration of the follower as shown in equation (1).

$$\ddot{x}_n(t+T) = \lambda\left[\dot{x}_{n-1}(t) - \dot{x}_n(t)\right] \tag{1}$$

where,

t : time,
T : reaction time,
$\ddot{x}_n(t+T)$: acceleration of the n^{th} vehicle at time $t+T$,
$\dot{x}_n(t)$: speed of the n^{th} vehicle at time t,
λ : sensitivity factor

GM researchers also developed generalized car-following models by considering that the sensitivity factor in equation (1) can be as simple as a constant value, but also can be a complicated function of headway and speed of the vehicles. This generalized car-following equation is called the GM model shown as equation (2).

$$\ddot{x}_n(t+T) = \alpha \frac{\left[\dot{x}_n(t+T)\right]^m}{\left[x_{n-1}(t) - x_n(t)\right]^l}\left[\dot{x}_{n-1}(t) - \dot{x}_n(t)\right] \tag{2}$$

where,

$x_n(t)$: position of the n^{th} vehicle at time t,
α , l, m : model parameters

It is known that equation (2) can represent many of car-following models introduced in past. For example, the equation becomes Pipes' and Forbes' models when l = 0 and m = 0. In the case where l = 1 and m = 0, the equation becomes the GM's third model, and it becomes the GM's fourth model with l = 1 and m = 1. This well-known GM's car-following model is also famous for making its connection to the macroscopic traffic stream models.

The GM model has been well-known as representing the stimulus-response system in car-following behavior. Other than the GM models, there are many other existing car-following models with various modeling aspects. Bracksone and McDonald (1999) classified the existing car-following models into several groups; GM Model, Collision Avoidance Model, Linear Model, Psychophysical Action Point Model and Fuzzy Logic Based Model. In addition to these model groups, several researchers recently started developing neural network driver models (Chong and Medina, 2011; Metthew and Ravishankar, 2012; Panwai and Dia, 2007; Colombaroni and Fusco, 2011). They are all modeled with representing unique and specific driving behavior aspects. Although there are several candidates for model comparison, the GM model was selected for model comparison in this study due to its long history and name recognition.

3. CAR-FOLLOWING DATA SETS

The car-following data was recorded in a test track at the Civil Engineering Research Institute of Hokkaido, Japan in October 2000. The test track consisted of two 1.2 km straight sections with two 0.3 km semicircular curves. To eliminate any geometry effects from the curves, only the data on the straight sections were used. The location and speed of each vehicle were recorded at 0.1 second intervals throughout the experiment. RTK GPS (Real-Time Kinematic GPS) was equipped on each of the ten test vehicles and the trajectories of the vehicles were recorded. Gurusinghe et al. (2002) examined the accuracy of the experimental data and confirmed these data sets are sufficiently accurate.

Two different types of data were recorded in the experiment: one for car-following conditions without stops and the other for the vehicle movements from start to stop in short distances. The car-following data portion has been used for the studies of several researchers such as Gurusinghe et al. (2002), Ranjitkar et al. (2003; 2004; 2005), and Tanaka et al. (2008)

There were a total of ten drivers in the car-following platoon. In this study, the ten drivers were identified as D01 through D10 from head to tail of the platoon. The driver of the first vehicle in the platoon, D01, initiated several speed patterns in the experiment. The data sets for D02 through D10 were used for fitting car-following models. There were a total of 47 data sets consisting of 21 car-following data sets and 26 start & stop data sets used in this analysis. The car-following data sets were set to start at 40-50 km/h, have some speed fluctuations, and end at the same initial speed within the straight sections. The start & stop speed data sets were designed to accelerate up to an expected optimum speed of 40-50 km/h from a stop condition, travel with speed fluctuations, and decelerate to complete stop. The length of the data collection period was 4342.3 seconds with 43,423 data points for each of ten vehicles in these data sets. The reaction time for each driver's response had to be preset in training for use in calibrating the car-following models in this study. The reaction times for the individual drivers were referred to the previous study by Ranjitkar et al. (2003). The reaction time approximately ranged between 1.3 to 1.6 seconds among the drivers D02 through D10.

4. METHODOLOGY

As mentioned in the previous paragraph, the raw car-following data included more than forty-two thousand data points for each vehicle. In the back-propagation training process, the widely-spread, enormous data can sometimes cause overlearning and/or divergence problems. Therefore, data sampling and data conversion to fewer points were conducted for a smooth learning process during back-propagation training. Assuming the data points are not fluctuated in a short time interval, simple data sampling was conducted at every 10 to 30, which corresponds to 1.0 to 3.0 second intervals. Similarly to the other methodology to reduce the number of training data points, data conversion was performed through a self-organization ANN model called Kohonen Feature Map (KFM).

In this paper, two kinds of ANN models were utilized for the analysis. One is a self-organization ANN model and the other is a multilayer ANN model for back-propagation training process. The first ANN model, KFM, was used for transferring the raw data points to fewer data points for smooth back-propagation data training in the multilayer ANN model. The multilayer ANN model was then used as the main frame of the proposed ANN car-following models for representing human driving behaviors.

Four ANN car-following models were developed with various input variables for this study. Initial synaptic weights in the proposed ANN models were first calibrated with a GA model. After the synaptic weight initialization by GA, the back-propagation process was performed for training the synaptic weights further by learning the converted data sets throughout KFM. The numbers of input data variables varied from two to five depending on the ANN models.

In order to compare the performance of the ANN car-following models, the GM's car-following model was calibrated to fit to the same data sets. The GA model was again utilized for finding the best constants in calibrating the GM models. The ANN and GM models were evaluated with the R^2 value in order to measure how they fit to the raw data sets.

4.1 Genetic Algorithm (GA)

The Genetic Algorithm (GA) program was used for two purposes in this study: 1) to calibrate the initial weights of the ANN car-following models, and 2) to find the best fit coefficients of GM's car-following model shown in equation (2). The GA program used in this study was the Genecop III program proposed by Z. Michalewicz (1992). It is characterized by the operation by floating-point number, the flexibility in dealing with constraints and boundaries, and the systematic applications of mutations and crossovers. The GA's objective function was set to minimize the average of the sum squared error between estimated and observed data values.

$$J(e) = Minimize \frac{1}{N}\sum_{i=1}^{N}\left(y_{i,data} - y_{i,est}\right)^2 \tag{3}$$

where,

$J(e)$: objective function,
$y_{i,data}$: measured value for the output variable in *i*th input data,
$y_{i,est}$: estimated value for the output variable in *i*th input data,
N : number of data points

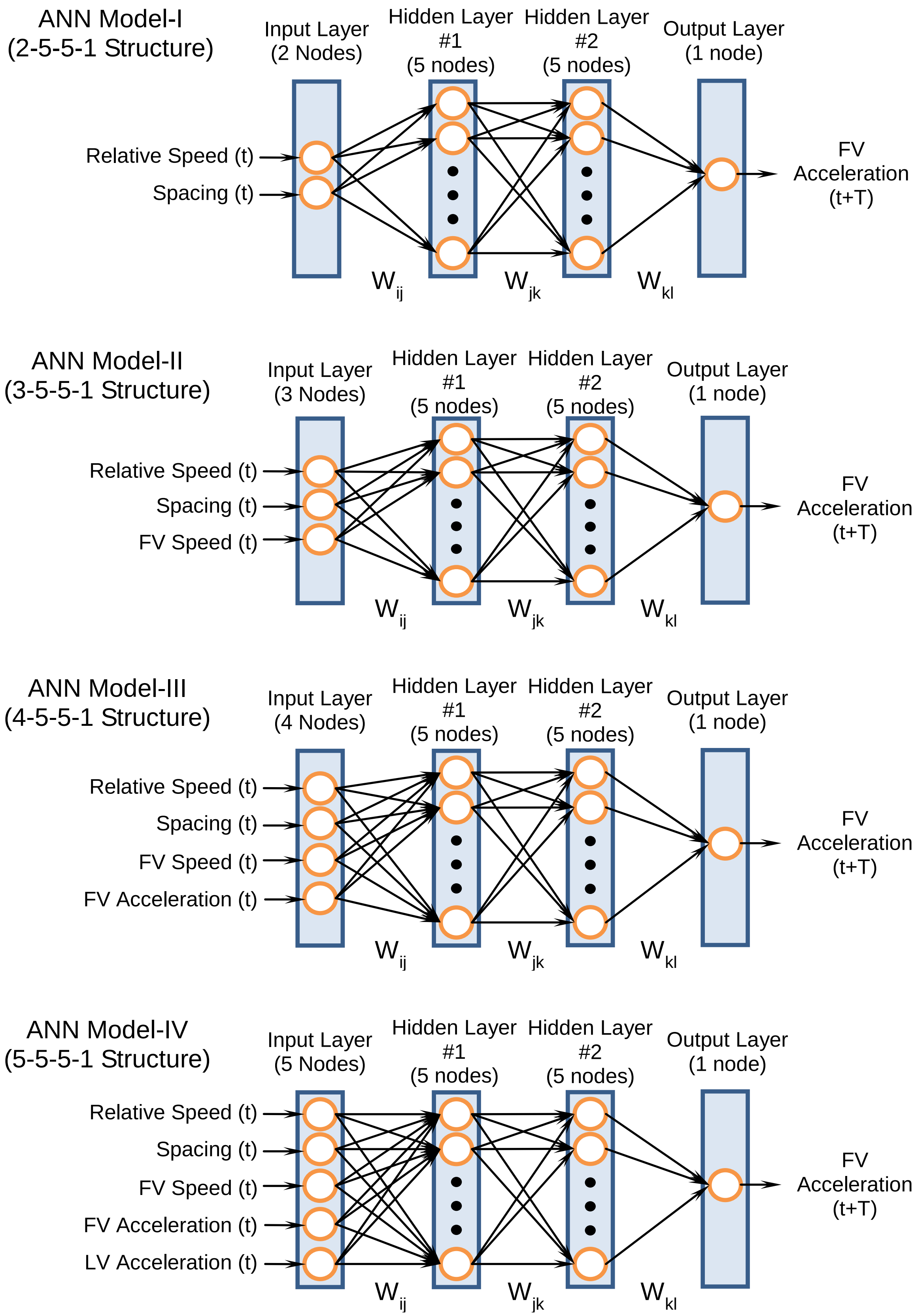


Figure 1. The Mutilayer ANN Car-Following Model Structures

Table 1. Input and Output Variables for the ANN and GM Car-Following Models

	# of Input and Output Variables	Input Variable	Output Variable
ANN Model-I	2 Inputs 1 Output	Relative Speed (t), $\dot{x}_{n-1}(t)-\dot{x}_n(t)$ Spacing (t), $x_{n-1}(t)-x_n(t)-L_n$	FV Acceleration (t+T), $\ddot{x}_n(t+T)$
ANN Model-II	3 Inputs 1 Output	Relative Speed (t), $\dot{x}_{n-1}(t)-\dot{x}_n(t)$ Spacing (t), $x_{n-1}(t)-x_n(t)-L_n$ FV Speed (t), $\dot{x}_n(t)$	FV Acceleration (t+T), $\ddot{x}_n(t+T)$
ANN Model-III	4 Inputs 1 Output	Relative Speed (t), $\dot{x}_{n-1}(t)-\dot{x}_n(t)$ Spacing (t), $x_{n-1}(t)-x_n(t)-L_n$ FV Speed (t), $\dot{x}_n(t)$ FV Acceleration (t), $\ddot{x}_n(t)$	FV Acceleration (t+T), $\ddot{x}_n(t+T)$
ANN Model-IV	5 Inputs 1 Output	Relative Speed (t), $\dot{x}_{n-1}(t)-\dot{x}_n(t)$ Spacing (t), $x_{n-1}(t)-x_n(t)-L_n$ FV Speed (t), $\dot{x}_n(t)$ FV Acceleration (t), $\ddot{x}_n(t)$ LV Acceleration (t), $\ddot{x}_{n-1}(t)$	FV Acceleration (t+T), $\ddot{x}_n(t+T)$
GM Model	3 Inputs 1 Output	Relative Speed (t), $\dot{x}_{n-1}(t)-\dot{x}_n(t)$ Headway (t), $x_{n-1}(t)-x_n(t)$ FV Speed (t+T), $\dot{x}_n(t+T)$	FV Acceleration (t+T), $\ddot{x}_n(t+T)$

Note: FV denotes a following vehicle (n^{th} vehicle), LV denotes a leading vehicle ($(n\text{-}1)^{th}$ vehicle), $x_n(t)$ is the position of the n^{th} vehicle at time t, T is a reaction time of the n^{th} vehicle, and L_n is the vehicle length of the n^{th} vehicle.

4.2 Multilayer Artificial Neural Network (ANN) Models

The main structure of the ANN models was developed for training the car-following data sets with back-propagation methodology. Figure 1 shows the structures of the multilayer ANN car-following models used in this study. All of the multilayer ANN models consist of four layers: an input layer, two intermediate hidden layers, and an output layer. Each layer has one or more neurons which are connected to the neurons on adjacent layers with some connection strengths called synaptic weights. The normalized input variable, between 0 and 1, was entered into the input layer. The input signals were transmitted in sequence from the input layer to the output layer while neural operations were repeated. The output layer produces the normalized objective variable. The forward signal process is completed with this computation sequence from the input layer to the output layer.

After the forward signal process, the synaptic weights were adjusted so that the error between the output signals and the target signals was minimized. The back-propagation method (Wasserman, 1989) produces the adjustments of synaptic weights in each layer. The synaptic weights are adjusted by the momentum method to smooth the adjustment and urge the convergence. The forward and back-propagation processes were repeated until the error between the target and the output was reduced within a tolerance limit or until a preset number of iterations was reached.

Four ANN models with different input variables were developed in this study. They are identified as ANN Model-I, ANN Model-II, ANN Model-III and ANN Model-IV throughout this work. The descriptions of the layer and node structure as well as the input and output variables for these ANN car-following models are listed in Table 1.

4.3 Kohonen Feature Map (KFM): A Self-Organization Neural Network Model

The KFM model is a two-layered neural network that can organize a topological map from a random starting point. It has the ability to classify input patterns into several output patterns. Successful data conversions were reported in non-linear regression analysis by Nakatsuji *et al.* (1995). Figure 2 depicts the basic network structure of a KFM model. In this study, a two dimensional structure was used for the analysis. It consists of two layers: an input layer and a competitive layer. The interconnections (synaptic weights) are adjusted in a self-organizing manner without any target signals. The self-organization process used by the KFM model is briefly explained in the following paragraphs. An input pattern to the KFM is denoted here as

$$E = \left[e_1, e_2, \ldots e_k \ldots, e_n \right] \tag{4}$$

where,

E : input vector

e_k : kth input variable in the input layer

The observed car-following variables are adopted as the input signals, the input layer consists between three (n=3 for ANN Model-I case) and six neurons (n=6 for ANN Model-IV case). The number of neurons in the competitive layer can be specified arbitrarily. The weights from the input neurons to a single neuron in the competitive layer are denoted as

$$W_{ij} = \left[w_{1ij}, w_{2ij}, \ldots . w_{kij}, \ldots, w_{nij} \right] \tag{5}$$

where,

W_{ij} : synaptic weight vector for the neuron in ith row and jth column in the competitive layer,

w_{kij} : synaptic weight value between kth neuron in the input layer and the neuron in ith row and jth column in the competitive layer,

n : number of input variables in the input layer

The first step in the adjustment of synaptic weights is to find a winning neuron c in the competitive layer whose weight vector matches most closely to each input vector E. The matching value is defined by the distance between vectors E and W_{ij}:

$$\sqrt{\sum_k \left(e_k - w_{kij} \right)^2} \tag{6}$$

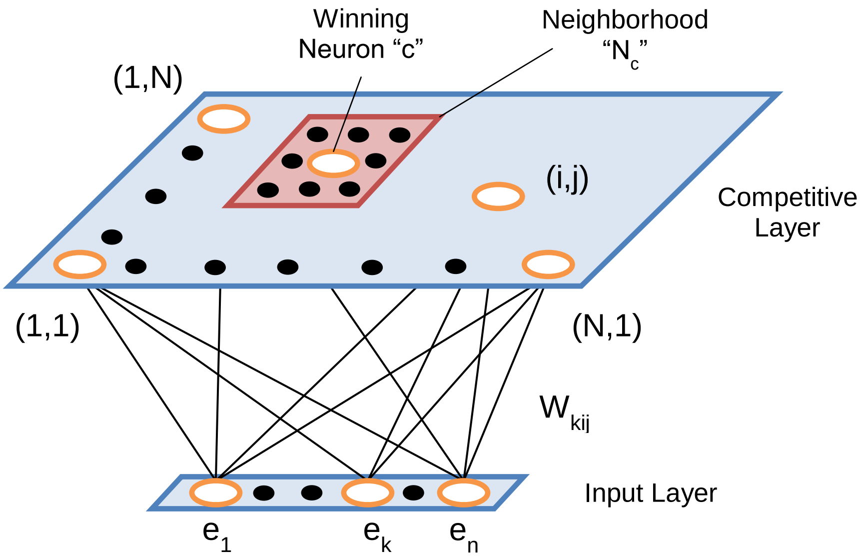


Figure 2. Two Dimensional KFM Model Structure

The neuron with the lowest matching value wins the competition. After the winning neuron c is identified, weights are updated for all neurons that are in the neighborhood N_c of the winning neuron. The adjustment is

$$\Delta W_{kij} = \begin{cases} \beta(e_k - w_{kij}) & if \quad i, j \in N_c \\ 0 & otherwise \end{cases} \tag{7}$$

where,

ΔW_{kij} : synaptic weight adjustment between kth neuron in the input layer and the neuron in ith row and jth column in the competitive layer,

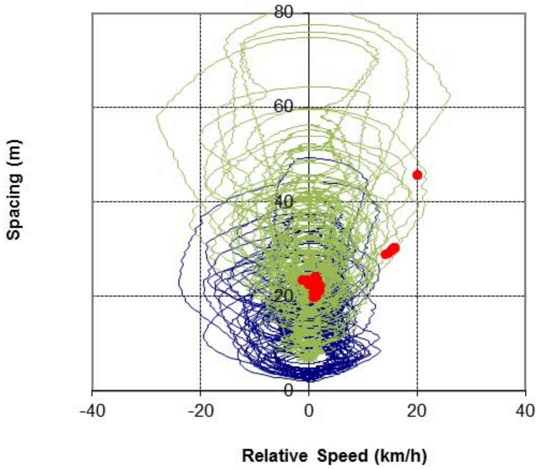
β : learning rate,

c : winning neuron in the competitive layer

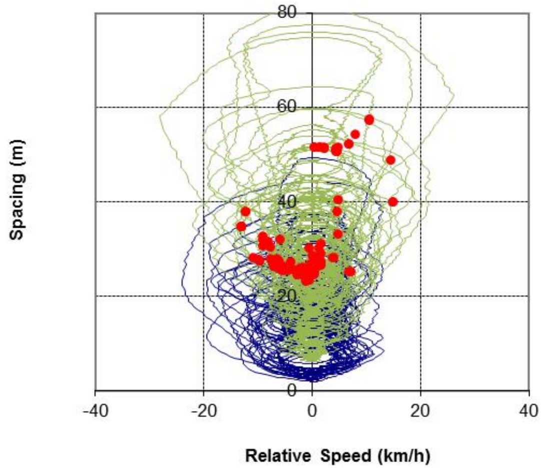
N_c : neighborhood of the winning neuron in the competitive layer

β is the learning rate, which is decreased over a span of many iterations. This adjustment results in the winning neuron becoming more likely to win the competition when the same similar input pattern is presented subsequently. In other words, the synaptic weight vector W_{ij} consequently represents those input patterns that resemble each other. This is called the integration of observed data. More details can be found in the work by Dayhoff (1990).

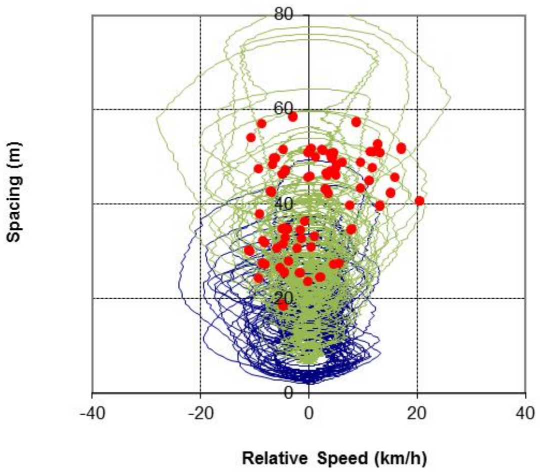
Figure 3 shows an example of the KFM network development used in this study. Only two variables of relative speed and spacing were plotted on the figures for visualizing the developments as the percentage of competitive iteration increases. Blue and green lines were plotted from the raw data of start & stop and car-following conditions, respectively, and the red points were the converted data points by KFM in the figures. At the beginning, the converted points were close to each other in a small area in the middle of the data spread as seen in Figure 3(a). Then the converted points start spreading out as the iteration percentage increased as shown in Figures 3(b) and 3(c). Finally, all the converted points were well distributed covering the raw data plot area as seen in Figure 3(d). The converted points were not well spreading out during the first half of iteration, but they actively start spreading out in the second half of iteration. This is because the converted points were pulled by many original data points with large winning neuron neighborhoods in the KFM structure and they tend to stay in the middle of majority of the data points rather than start spreading out. However, in the second half of the iteration, the winning neuron neighborhoods become smaller and smaller when the iteration percentage increases, then the converted points were pulled by fewer and fewer data points and started settling at the middle points for the fewer data points.



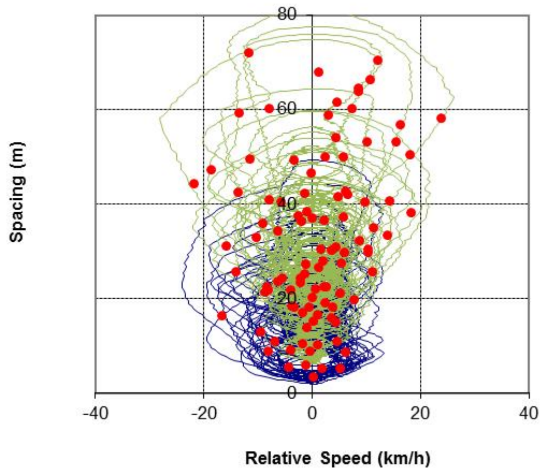
(a) Iteration = 5 (5% Completed)



(b) Iteration = 50 (50% Completed)



(c) Iteration = 70 (70% Completed)



(d) Iteration = 100 (100% Completed)

Figure 3. Development of Converted Data Points by KFM

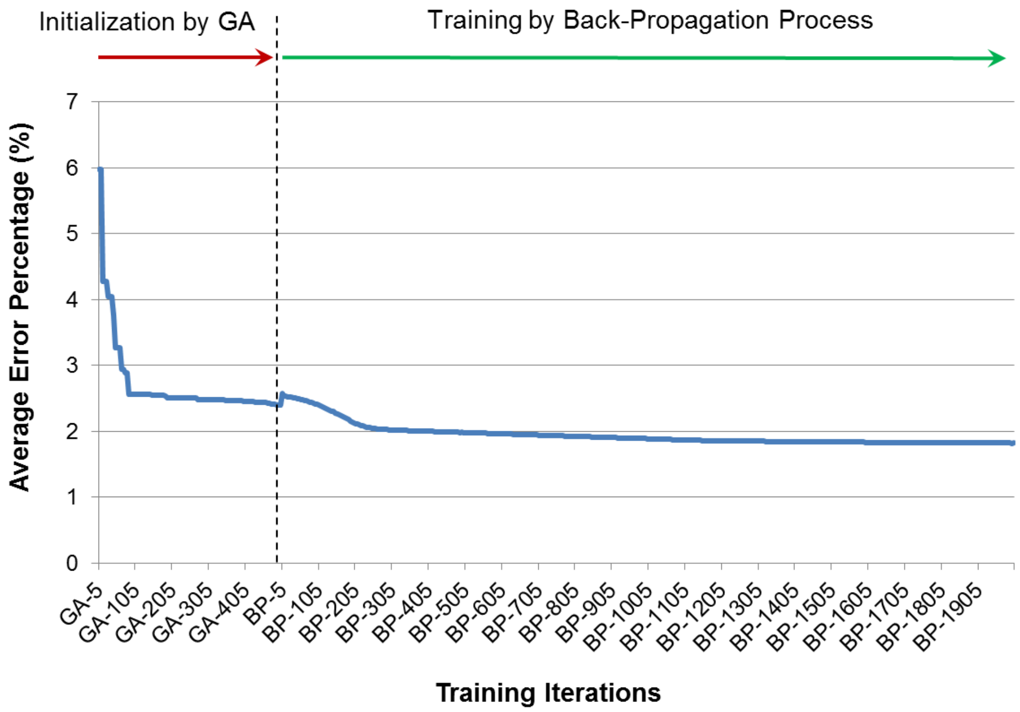


Figure 4. The Multilayer ANN Learning Curve by GA and Back-Propagation

5. ANALYSIS RESULTS

5.1 Determining Optimum Numbers of Sample Point by KFM and Simple Sampling Methodology (SSP)

Since the back-propagation network sometimes causes a problem in the learning of numerous, widely-spread data points, the effort of reducing the number of training data points was first attempted with the data set for one driver. At the beginning, only ANN Model-IV was trained with several different numbers of sample points to find out the optimum number of sampling points for the back-propagation process. Two kinds of methodologies were used for reducing number of data points. One methodology was to convert the raw data to fewer data points by KFM as introduced earlier. The raw data points were over 42,000 for each driver. The data points were converted by KFM down to 100, 225, 400, 625, and 900 points from the enormous number of raw data points. The other sampling methodology was simply sampling at every certain number of raw data points in the continuous car-following data sets. Sampled data sets at every 10, 20, and 30 raw data points were examined. Assuming the car-following data points are continuous and that there are no sudden changes in a short time period, it was expected that the sampled points with this simple sampling methodology (SSP) can represent the raw data very closely if the time interval between sampling is short enough. The ANN Model-IV was trained through the back-propagation methodology with the synaptic weight initialization by GA. An example of the learning curve encountered with these two methodologies is shown in Figure 4. As seen in the figure, the average error percentage was significantly reduced by GA at the beginning, then the back-propagation process takes over and starts reducing the error further for fine tuning through many iterations. In this combined error reduction methodology, often a bump occurs in the learning curve and the error increases temporarily before it decreases when the back-propagation process takes over. This is perhaps because the vector of reducing the error in the back-propagation process is not always the same as GA's error reduction vector.

Table 2. The R^2 Values after Training ANN Model-IV with Different Sampling Data Sets (Only with Driver D02 Raw Data Set)

Data Conversion Method	Number of Samples for Training	With Initialization of 100 GA Iterations			With Initialization of 500 GA Iterations		
		GA Only (R^2 to Samples)	GA & BP Training (R^2 to Samples)	**GA & BP Training (R^2 to Raw Data Sets)**	GA Only (R^2 to Samples)	GA & BP Training (R^2 to Samples)	**GA & BP Training (R^2 to Raw Data Sets)**
SSP-010	4281	0.767	0.845	**0.845**	0.811	0.842	**0.842**
SSP-020	2140	0.734	0.848	**0.837**	0.805	0.863	**0.853**
SSP-030	1427	0.770	0.862	**0.851**	0.790	0.861	**0.843**
KFM-900	900	0.829	0.891	**0.833**	0.836	0.887	**0.840**
KFM-625	625	0.823	0.913	**0.855**	0.836	0.910	**0.852**
KFM-400	400	0.767	0.919	**0.856**	0.849	0.919	**0.858**
KFM-225	225	0.851	0.935	**0.843**	0.895	0.945	**0.850**
KFM-100	100	0.923	0.980	**0.832**	0.953	0.977	**0.834**

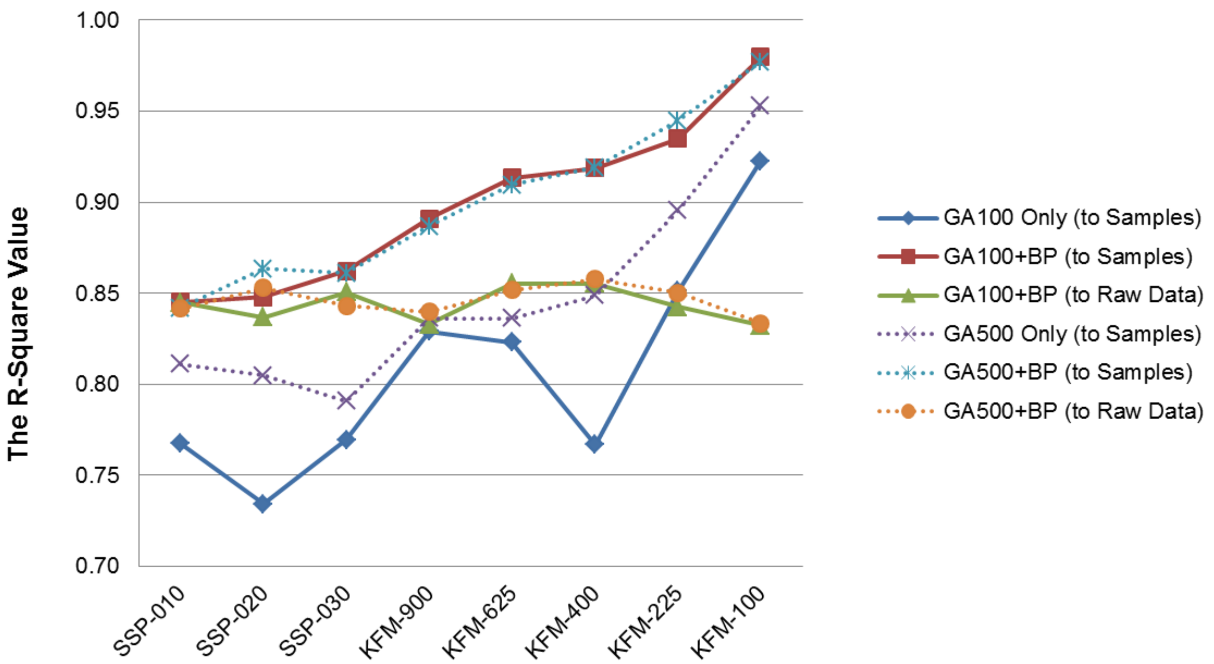


Figure 5. The R^2 Values after Training ANN Model-IV with Different Sampling Data Sets (Only with Driver D02 Raw Data Set)

Table 2 and Figure 5 summarize the R^2 values to the sampled data points and also to all raw data points calculated from outputs of the trained ANN Model-IV with the sampled data. In the table, the number after KFM, KFM-#, denotes the number of converted data points by KFM. The number after SSP, SSP-#, denotes the number of data intervals for simple sampling methodology. In all cases, the R^2 value increased with back-propagation training after GA initialization. This implies that back-propagation is successfully tuning the model further beyond GA's tuning limitation. It is noted that the number of GA iterations did not affect the R^2 values after the back-propagation training is completed. It is also recognized that there is a relationship between the number of samples and the R^2 value to the samples. As the number of samples is reduced, the R^2 value increases. This implies how well the ANN model is trained to smaller data sets. However, this phenomenon does not apply to the R^2 value to the raw data sets. Such a difference cannot be seen in the R^2 values to the raw data sets among the different sample cases. Among KFM data conversions, it is seen that KFM-400, 400 converted points, had the optimum size of samples for having the best R^2 values to the raw data point. Therefore, it was assumed that the 400 point KFM represents the best KFM data conversion with these data sets and this KFM model was used for further analysis. On the other hand, in the simple sampling methodology, SSP-020, sampling at every 20 data points, was arbitrarily chosen for the further analysis because no differences were seen in the R^2 values among the three data simple sampling cases.

Table 3. Parameter Values of the GM Models Calibrated by GA

Driver ID	*Reaction Time T (sec)*	Model Parameters			R^2
		α	l	m	
D02	1.3	0.568	0.117	0.648	**0.834**
D03	1.4	0.591	0.175	0.779	**0.858**
D04	1.4	1.051	0.085	0.795	**0.830**
D05	1.3	0.687	0.152	0.737	**0.831**
D06	1.5	0.934	0.027	0.722	**0.810**
D07	1.6	0.881	0.025	0.751	**0.845**
D08	1.5	1.431	0.207	1.042	**0.845**
D09	1.4	0.695	0.268	0.908	**0.856**
D10	1.4	1.126	0.196	0.982	**0.852**

5.2 The Four ANN Model Results

Four ANN models were trained with the two kinds of sampled data sets through two different sampling methodologies, KFM-400 and SSP-020, as mentioned in the previous paragraph. The ANN models were made for all individual drivers, D02 through D10, except the very first driver, D01, in the car-following platoon. GM models were also calibrated by GA. Table 3 summarizes the parameters and R^2 value for each driver.

Table 4 and Figure 6 summarize the R^2 values of the ANN models for each driver using SSP-020. As illustrated in Figure 1, the ANN Model-I has only two input variables while the ANN Model-IV has five input variables. As seen in the table, the R^2 values tend to increase when the number of input variables increases from the ANN Model-I to the ANN Model-IV.

The ANN Model-IV, which has the greatest number of the model input variables, had the highest R^2 value among the ANN models for the same driver. The last column in Figures 6 and 7 shows the R^2 values with the GM Model for comparison purposes. Compared to the GM Models, the ANN Model-I, the ANN Model-II and the ANN Model-III have smaller R^2 values. However, the ANN Model-IV has higher R^2 values than GM Model for most of the drivers.

Table 4. The R^2 Values of the ANN Models for Individual Drivers with the Training Data by Simple Sampling (SSP-020, Sampled Every 20 Data Point)

Driver ID	ANN Model-I	ANN Model-II	ANN Model-III	ANN Model-IV	GM Model
D02	0.714	0.718	0.782	0.837	0.834
D03	0.749	0.742	0.822	0.878	0.858
D04	0.702	0.677	0.744	0.830	0.830
D05	0.730	0.612	0.766	0.880	0.831
D06	0.672	0.684	0.785	0.883	0.810
D07	0.715	0.714	0.678	0.862	0.845
D08	0.745	0.698	0.784	0.887	0.845
D09	0.740	0.726	0.770	0.846	0.856
D10	0.734	0.632	0.796	0.841	0.852
Average	**0.714**	**0.689**	**0.770**	**0.860**	**0.840**

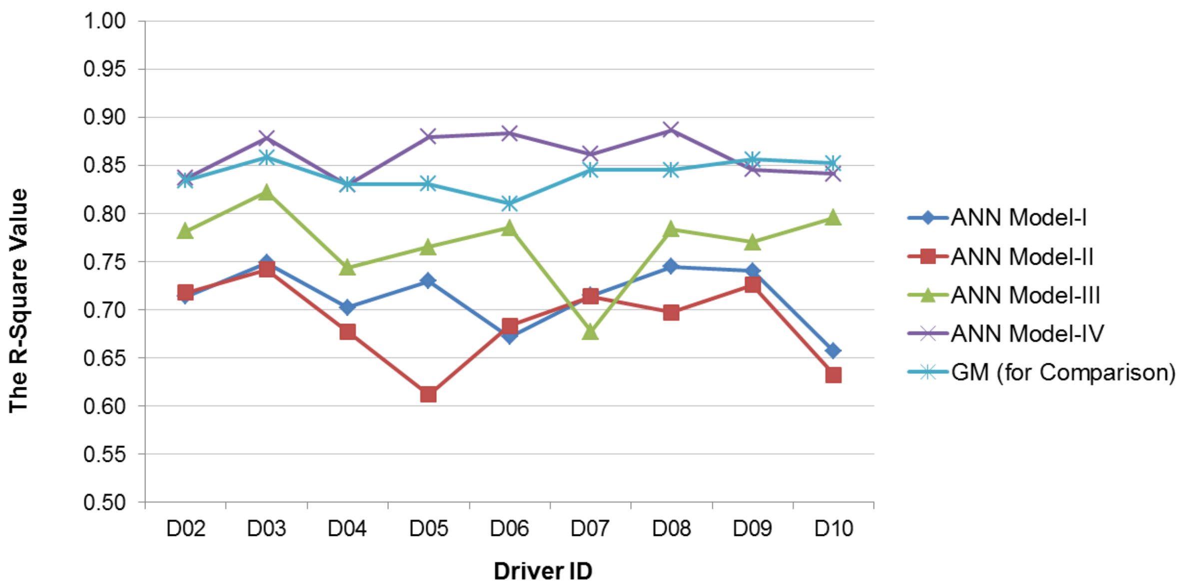


Figure 6. The R^2 Values of the ANN Models for Individual Drivers with the Training Data by Simple Sampling (SSP-020, Sampled Every 20 Data Point)

Table 5. The R^2 Values of the ANN Models for Individual Drivers with the Training Data by KFM (KFM-400, 400 Converted Sample Points)

Driver ID	ANN Model-I	ANN Model-II	ANN Model-III	ANN Model-IV	GM Model
D02	0.632	0.720	0.750	0.858	0.834
D03	0.684	0.731	0.795	0.876	0.858
D04	0.599	0.718	0.765	0.836	0.830
D05	0.696	0.703	0.780	0.876	0.831
D06	0.651	0.684	0.788	0.885	0.810
D07	0.681	0.643	0.718	0.867	0.845
D08	0.724	0.695	0.782	0.882	0.845
D09	0.710	0.723	0.772	0.835	0.856
D10	0.669	0.700	0.774	0.855	0.852
Average	**0.672**	**0.702**	**0.769**	**0.863**	**0.840**

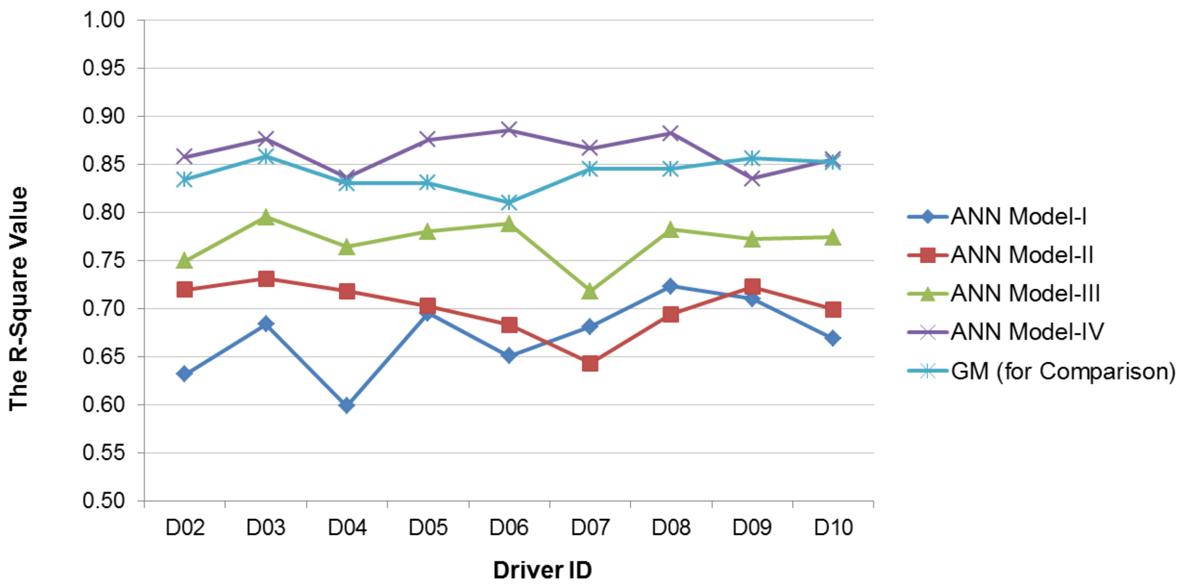


Figure 7. The R^2 Values of the ANN Models for Individual Drivers with the Training Data by KFM (KFM-400, 400 Converted Sample Points)

Finally, Table 5 and Figure 7 summarize the R^2 values of the ANN models for each driver with the training data converted by KFM-400. Similar to the R^2 values in Table 4 and Figure 6, the ANN model has the same relationship between the R^2 value and the number of input variables among the ANN models with KFM converted data sets. Comparing the R^2 values of the ANN Model-III and Model-IV in Figure 8, almost no difference can be seen between the simply sampled data and KFM converted data. This verifies that KFM data conversion was successful for sampling the data sets without data deterioration. Figure 9 shows the comparison of observed acceleration rate and the model outputs for one start & stop data set. It is seen that both the ANN Model-IV with KFM-400 well fit the observed data. However, the GM model slightly overestimates in the low range of acceleration and deceleration rates.

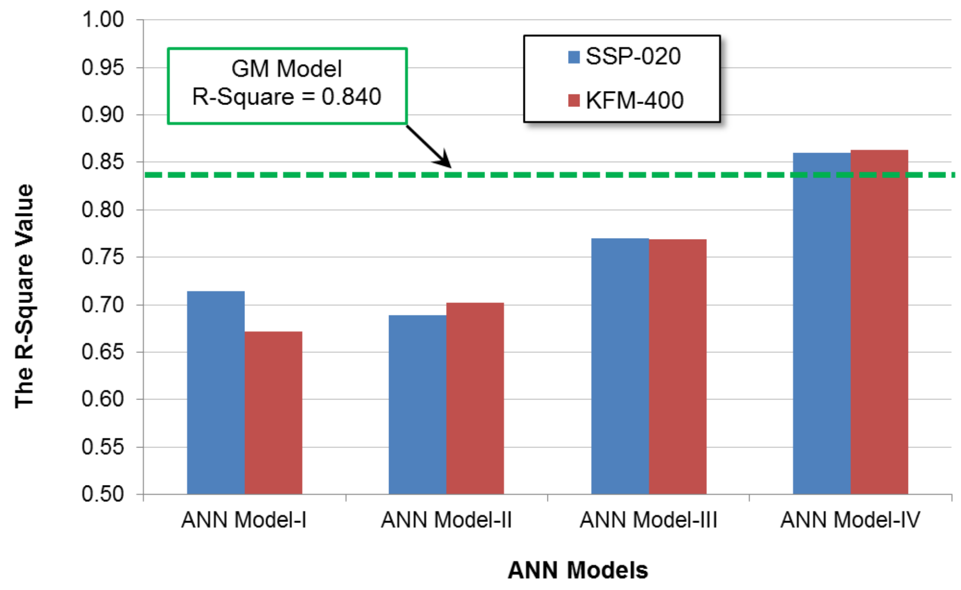


Figure 8. Average R^2 Values for the ANN Models

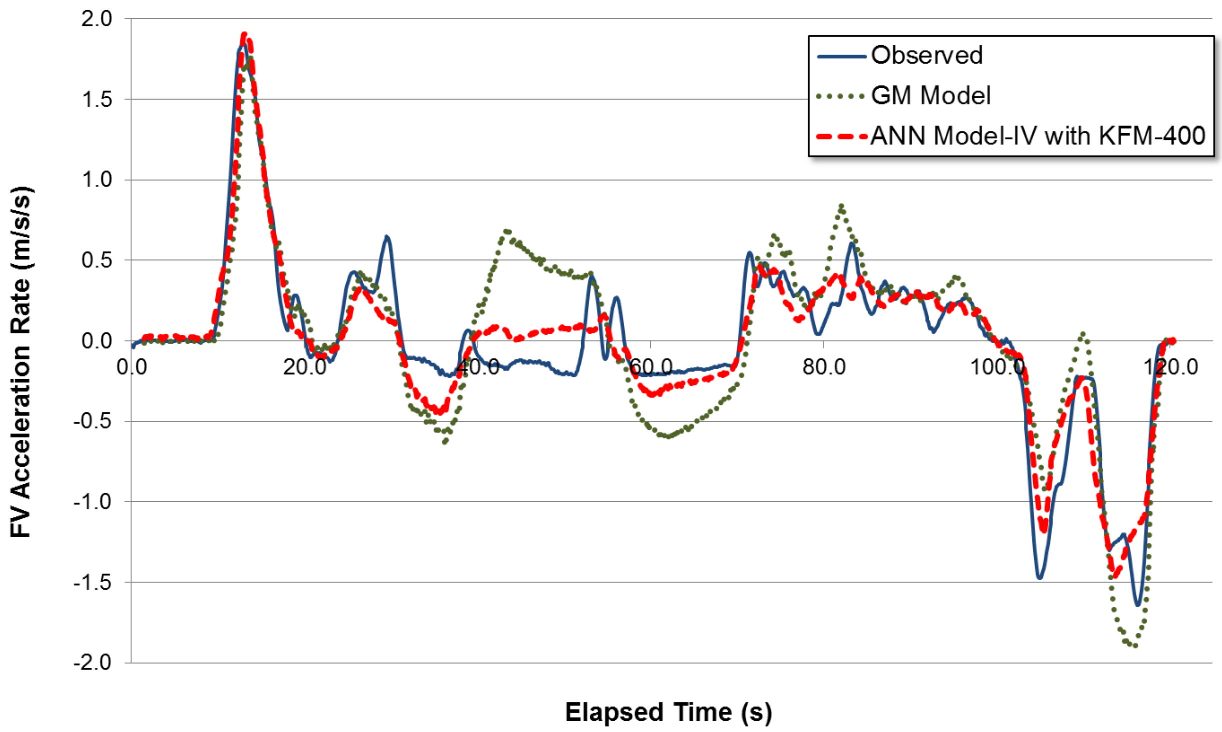


Figure 9. Comparison of Observed Acceleration Rates and Estimated Model Outputs for One Start & Stop Data Set

6. CONCLUSION

In this study, four ANN car-following models were developed and compared to the existing GM car-following model. The data conversion methodology with KFM was successful for creating the training data sets for the back-propagation process. The four ANN models were successfully trained using GA and back-propagation methodologies with the sampled data sets.

ANN Model-IV, which has five input variables, showed the highest R^2 values for all driver cases among the four ANN models. At the same time, ANN Model-I and II had similar R^2 values and showed the lowest R^2 values among the four ANN models. This phenomenon indicates that either ANN Model-II had difficulty in training from the data sets, or that the extra input variable from ANN Model-I to Model-II, FV speed at time t, has very little influence to the model output, FV acceleration at time t+T. Although the R^2 values did not increase from ANN Model-I to Model-II, they increased significantly from ANN Model-II to Model-III and from Model-III to Model-IV. This implies that the extra input variables of FV acceleration at time t and LV acceleration at time t influence the car-following behavior significantly. These input variables cannot be seen in many of the existing car-following models, however, there is a possibility that these variables are actually key inputs to improve the car-following model quality and accuracy. This may be true in the real human driving behavior. It is often seen that a driver determines the degree of acceleration not only with the relative speed, but also with the expectation of how the relative speed is going to change at the next moment. The differentiation of relative speed is the difference between FV and LV acceleration rates. The acceleration rates of both FV and LV might be very important information for a driver in order to estimate the change of the relative speed in car-following conditions.

The GM model was also calibrated to the same data sets with the GA technique and the results were compared to the ANN models. The GM model showed very high R^2 values and resulted in having better performance than the three of the ANN models: ANN Model-I, Model-II and Model-III. Only one model, ANN Model-IV, had higher R^2 values than the GM model.

The results of this study illustrate the effectiveness of the ANN car-following model with GA initialization and KFM data conversion procedures. We also provided validation of the GM car-following model in microscopic aspects with the enormous amount of observed car-following data.

REFERENCES

Brackstone, M., McDonald, M. (1999) Car-Following: A Historical Review. *Transportation Research Part F*, 2, 181-196.

Chong, L., Abbas, M., Medina A (2011) Simulation of Driver Behavior with Agent-Based Back-Propagation Neural Network. *Transportation Research Record 2249*, Transportation Research Board, Washington D.C., 44-51.

Colombaroni C., Fusco G. (2011) Artificial Neural Network Models for Car Following: Experimental Analysis and Calibration Issues. *Proceedings of 2nd International Conference on Models and Technologies for Intelligent Transportation Systems*, Leuven, Belgium, June 22-24.

Dayhoff, J. (1990), *Neural Network Architecture*, Van Nostrand Reinhold.

Forbes, T. W. Human Factor (1963) Considerations in Traffic Flow Theory. *Highway Research Board, Record 15*, Highway Research Board, Washington D.C., 60-66.

Forbes, T. W., Simpson, M. E. (1968) Driver and Vehicle Response in Freeway Deceleration Waves. *Transportation Science*, 2(1), 77-104.

Forbes, T. W., Zagorsk, H. J., Holshouser, E. L., Deterline, W. A. (1958) Measurement of Driver Reactions to Tunnel Conditions. *Highway Research Board, Proceedings*, 37, 345-357.

Grusinghe, G. S., Nakatsuji, T., Azuta, Y., Ranjitkar, P., Tanaboriboon, Y. (2002) Multiple Car Following Data Using Real Time Kinematic Global Positioning System. *Transportation Research Record 1802*, Transportation Research Board, Washington D.C., 166-180.

Mathew, T., Ravishankar, K. (2012) Neural Network Based Vehicle-Following Model for Mixed Traffic Conditions. *European Transport*, 52(1), 15.

Michaelewicz, Z. (1992), *Genetic Algorithms + Data Structures = Evolution Programs,* Springer.

Nakatsuji, T., Tanaka, M, Pourmoallem, N., Hagiwara, T., (1995) Description of Macroscopic Relationships Among Traffic Flow Variables Using Neural Network Models. *Transportation Research Record 1510*, Transportation Research Board, Washington D.C., 11-18.

Panwai, S., Dia, H (2007) Neural Agent Car-Following Models. *IEEE Transactions on Intelligent Transportation Systems*, 8(1), 60-70.

Pipes, L.A. (1953) An Operational Analysis of Traffic Dynamics. *Journal of Applied Physics*, 24(3), 274-287.

Ranjitkar, P., Nakatsuji, T., Azuta, Y., Asano, M., Kawamura, A. (2005) A Contemporary Reassessment of GM Car-Following Model Using RTK-GPS Data. *J. Infrastructure Planning and Management*, JSCE, 793(IV-68), 121-132.

Ranjitkar, P., Nakatsuji, T., Azuta, Y., Gurusinghe G. S. (2003) Stability Analysis Based on Instantaneous Driving Behavior Using Car Following Data. *Transportation Research Record 1852*, Transportation Research Board, Washington D.C., 140-151.

Ranjitkar, P., Nakatsuji, T., Kawamura (2005) A. Experimental Analysis of Car Following Dynamics and Traffic Stability. *Transportation Research Record 1934*, Transportation Research Board, Washington D.C., 22-32.

Ranjitkar, P., Nakatsuji, T., Motoki, A. (2004) Performance Evaluation of Microscopic Traffic Flow Models Using Test Track Data. *Transportation Research Record 1876*, Transportation Research Board, Washington D.C., 90-100.

Tanaka, M., Ranjtkar, P., Nakatsuji, T. (2008) Asymptotic Stability and Vehicle Safety in Dynamic Car-Following Platoon. *Transportation Research Record 2088*, Transportation Research Board, Washington D.C., 198-207.

Wasserman, P.D. (1989), *Neural Computing*, Van Nostrand Reinhold