IV－22

# Description of Macroscopic Relationships Among Traffic Flow Variables Using Neural Network Models

M.TANAKA, Student Member, Hokkaido Univ.
T.NAKATSUJI, Member of JSCE, Hokkaido Univ.
T.KAKU, Member of JSCE, Hokkaido Univ.

## 1. INTRODUCTION

For many decades, various mathematical formulations that describe the relationships between density, flow, and speed have been proposed including multi-regime models. Previously, the best mathematical curve was determined by trying several different formulas and applying regression analyses. In these processes, one has to specify in advance which mathematical formula should be adopted and where it should be shifted to another in a multi-regime model.

But neural network models have some promising abilities to accurately represent non-linear behaviors and to self-organize automatically. This paper aims to present a procedure to describe the macroscopic relationships between traffic flow variables using some neural network models.

## 2. MACROSCOPIC TRAFFIC FLOW VARIABLES

### 2.1 Nonlinear Regression Model

There are many characteristics curves proposed so far describing the relationship among density and speed. In this study we used the formula derived from the car-following theory [1) ] where k denotes density (veh/km), v speed (km/h), $k_j$ jam density, and $v_f$ free speed. And parameters, l and m, respectively have the determined range. Substituting Eq.(1) into the relationship q=kv, we can obtain the other relationships among density, flow, and speed.

$$v = v_f\left[1-\left(\frac{k}{k_j}\right)^{l-1}\right]^{\frac{1}{1-m}} \quad (l>0,\ m>1) \tag{1}$$

### 2.2 Neural Network Models

1) A Multilayer Neural Network Model

Figure 1 shows a multilayer neural network model for describing the macroscopic relationships between traffic variables. In this paper it consists of three layers, an input layer, an intermediate layer , an output layer. Each neuron in the layers mutually connected to neurons in the adjacent layers. The strength of the connections is called synaptic weight. First of all, we initialize synaptic weights randomly. We input the normalized control variable into the input layer, such as k in the case of k-v curve. And we transmit the input signals in sequence from the input layer to the output layer while repeating the neural operations. Then the output layer produces the normalized objective variable, such as v in the case of k-v curve. Next, we adjust the synaptic weights so that the error between the output signals and the target signals is minimized. This is the method called back-propagation(BP)[2)], to adjust in sequence synaptic weights from the output layer to the input layer. In this paper, we set observed data input variable or target signal and obtain objective variable.

2) Kohonen Feature Map

The Kohonen-Feature-Map(KFM)[3)] 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. By applying this property to the selection of observed data, we can convert the data set to one with fewer, more

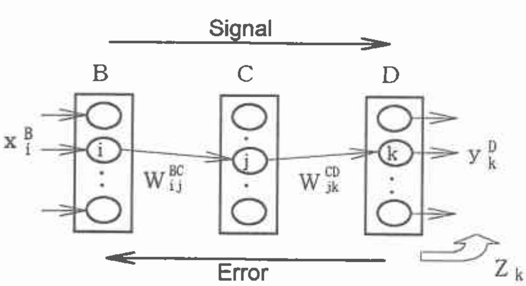


Figure 1. Basic Structure of Multilayer Neural Network.

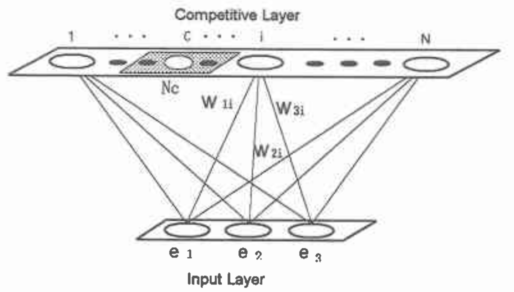


Figure 2. Basic Structure of Kohonen Feature Map Model.

integrated data points. Figure 2 depicts the basic network structure of a KFM model. It consists of two layers, an input layer and a competitive layer. We employed a KFM model whose structure is one dimensional in competitive layer. The two layers are fully interconnected from the first layer to the second. The interconnections(synaptic weights) are adjusted in a self-organizing manner without any target signals. Then after training, we can obtain fewer converted data. The synaptic weight vector consequently represents those input patterns that resemble each other. This is what we call the integration of observed data.

## 3. TRAFFIC DATA

Observed Data

The observed data used here come from Metropolitan Expressway in Tokyo. The data were collected on the Yokohane Line between Tokyo Haneda Airport and Yokohama in October 1993. Supersonic traffic detectors are installed each of the two directional lanes every 300 meters and traffic data on flow, occupancy, and average speed are compiled every 1 minute. We chose the time periods that include extensive traffic situations, ranging from free-flow to congested states in the daytime on weekdays. In this analysis, we used time occupancy directly rather than converting it to density. Assuming that density is proportional to time occupancy, we introduce a proportional constant and replace density k and $k_j$ with occupancy and maximum occupancy, respectively.

## 4. THE METHOD OF TRAINING AND ITS RESULT

4.1 A Regression with Nonlinear Formula

In two-dimensional analyses, the nonlinear formula(Eq.(1)) has some unknown parameters. To determine these parameters, we used Box's Complex algorithm [4)]. The algorithm was the method so as to minimize or maximize.

4.2 Training

Now we explain how we describe the relationships among traffic flow variables. First, using the KFM model in the three dimensional space, we convert original observed data to fewer points of more integrated data. Then we project them on each two-dimensional plane for two dimensional analyses. Next, using a multilayer neural network model, we build up the input-output relationships between the control and the state variables. The completion of training by the back-propagation method brings a stable regression between them.

1) Kohonen Feature Map

To convert observed data to sets of integrated data, we prepare a KFM model consisting of an input layer with three neurons and a competitive layer with neurons that correspond to the number of integrated data points. Prior to the training, all observed data are normalized. After having given a set of observed data to the input layer in Figure 2, we iterate the training until the change of synaptic weights becomes sufficiently small. Finally, we can obtain a stable formation of integrated data. In this study, we converted 120 points of observed data into 20 points integrated data. Figure 3 shows how original observed data are integrated as the training proceeds. For simplicity, the evolution process is projected on the occupancy-speed plane, and for convenience, it is enlarged to the real scale.

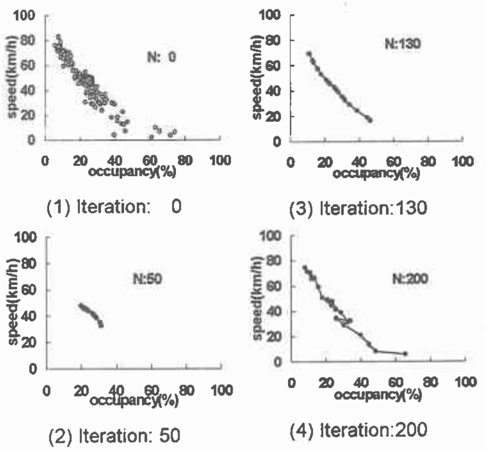


Figure 3. Evolution into Integrated Data by Kohonen Feature Mapping

2) Multilayer Neural Network

As mentioned before, we prepare a multilayer neural network with a neuron in the input layer and a neuron in the output layer for two-dimensional analyses. On the other hand, we use a network with two neurons in the input layer and a neuron in the output layer for three dimensional analyses. We adjust the synaptic weights by the back-propagation method. In this study we adopted a training procedure which is a little different from the usual one. Here, we adjust synaptic weights thoroughly for an input pattern until the error between the output signal and the target signal becomes sufficiently small.

The neural network method has another characteristic, in that we can change the smoothness of regression curves by changing the threshold for judging the convergence. In general, a large threshold would provide a smooth curve and a small one a complicated one.

## 4.3 RESULTS

1) Two-Dimensional Relationships

We compare two methods, an analytical one by non-linear equations, and one using artificial intelligence through neural network models. First, we compare the methods using the traffic data observed at the Detector Station 1201. Figure 4 shows three regression curves about occupancy-speed curve. The white circles in Figure 4 present 120 points of original observed data and black ones present 20 points of data integrated by KFM model. And the correlation coefficients about occupancy-speed curve are shown in Table 1. Figure 4 (2) and (3) let us recognize that the neural network methods are more suitable than the non-linear equation to represent discontinuous data. It needs neither to divide the whole region into several ones nor to introduce an individual function for each region. We can also recognize that the neural network methods are better than the non-linear equation in comparing with these correlation coefficients. And it is needless to say that the neural method with the KFM model is more efficient in the computation than that without the KFM model. Then we can compare the correlation coefficients on occupancy-flow curve or flow-speed curve as shown in the table 2 or 3, respectively. Here also as shown in table 2 and 3, the neural method gives better correlation coefficients than the non-linear equation. In actually, the neural method represent the non-linear curves, as shown in Figure 4 (4) and (5).

2) Three-Dimensional Relationships

By introducing a multilayer neural network model that inputs traffic flow and occupancy, and outputs average speed, we can describe a three-dimensional relationship among those variables. We prepare the multilayer network, taking flow and occupancy as the control variables and average speed as the state variable, and iterate the training operations by the back-

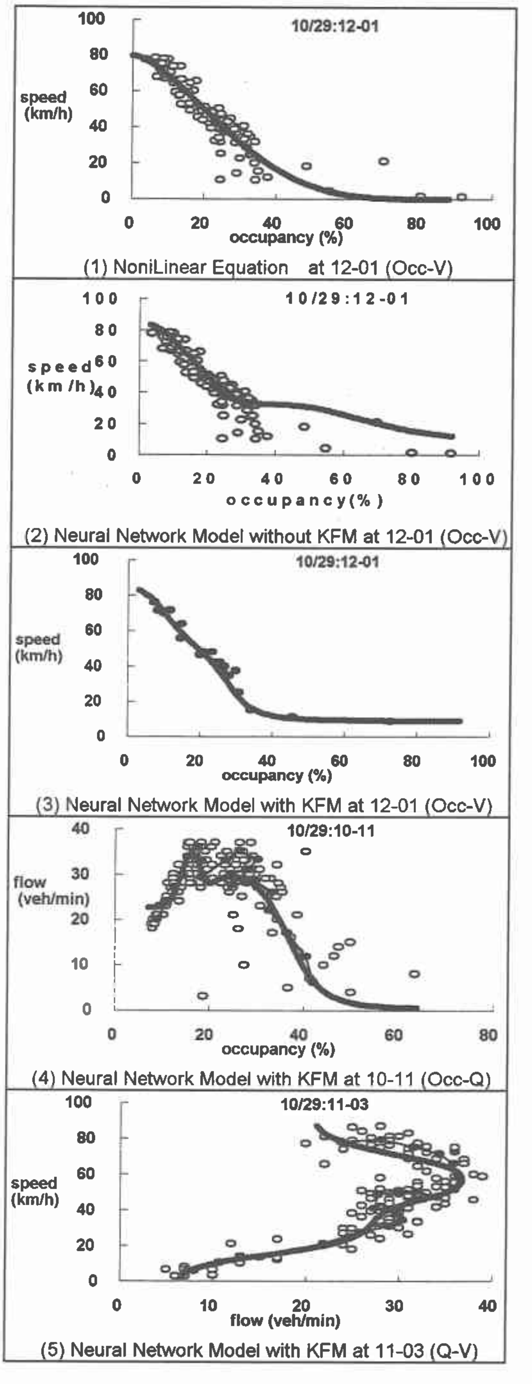


Figure 4. Comparison of Neural Network models with Non-Linear Equation on Occupancy-Speed Curve and representation by Neural Network models

Table 1. Comparison of Multiple Correlation Coefficients on Occupancy-Speed Curve

| Detector Point | Non-linear Equation | Neural Network without KFM | Neural Network with KFM |
|---|---|---|---|
| 1009 | 0.94 | 0.97 | 0.97 |
| 1011 | 0.87 | 0.92 | 0.94 |
| 1103 | 0.91 | 0.94 | 0.95 |
| 1201 | 0.88 | 0.91 | 0.92 |
| 1203 | 0.92 | 0.96 | 0.97 |

Table 2. Comparison of Multiple Correlation Coefficients on Occupancy-Flow Curve

| Detector Point | Non-linear Equation | Neural Network with KFM |
|---|---|---|
| 1009 | 0.60 | 0.78 |
| 1011 | 0.47 | 0.58 |
| 1103 | 0.74 | 0.79 |
| 1201 | 0.52 | 0.66 |
| 1203 | 0.61 | 0.80 |

Table 3. Comparison of Multiple Correlation Coefficients on Flow-Speed Curve

| Detector Point | Non-linear Equation | Neural Network with KFM |
|---|---|---|
| 1009 | 0.70 | 0.83 |
| 1011 | 0.74 | 0.85 |
| 1103 | 0.79 | 0.82 |
| 1201 | 0.63 | 0.81 |
| 1203 | 0.75 | 0.81 |

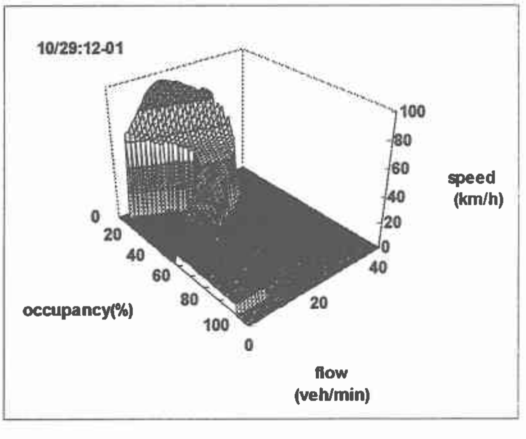


Figure 5. Three Dimensional Representation by Neural Model.

propagation method. Figure 5 shows the regression surface for the data at Detector Station 1201. Seeing such a surface, we should recognize not to need any specific functions to describe the surface.

## 5. CONCLUDING REMARKS

We applied some neural network models to description of the relationships among traffic flow variables. And we investigated the applicability of the neural network models to the regression problem. Also, we compared the results with those produced by a conventional non-linear equation. Our major finding are summarized as follows:

1) A Kohonen Feature Map method served to integrate original observed data points into fewer, more uniformly distributed data points. This integration contributes to the improvement of regression precision and computational efficiency.

2) A multilayer neural network model was effective in describing the non-linear and discontinuous relationships
among traffic flow variables. The model made it unnecessary to specify the regression curve and the transition points in advance. In addition, the multiple correlation coefficients produced by the neural model were better than those produced by a non-linear equation.

3) A multilayer neural model was applicable to a three-dimensional problem. No specific function was necessary to describe the regression surface.

In this paper, we confined the discussion to availability of neural network models. The interpretation of traffic phenomena using them is being left to future work. Moreover, we have to examine the availability of other neural network models that might be more effective than those used here.

Finally, we thank Dr. Morita and his colleagues at the Metropolitan Expressway Office in Tokyo for offering us observation data.

REFERENCES


1. A. D. MAY: *Traffic Flow Fundamentals*, Prentice Hall, 1990
2. P. D. Wasserman: *Neural Computing*, Van Nostrand Reinhold, 1989
3. J. Dayhoff: Neural *Network Architecture*, Van Nostrand Reinhold, 1990
4. J. L. Kuester and J. H. Mize: *Optimization Techniqyes with Fortran*, McGraw-Hill