Implementation of Neural Network with a variant of Turing Machine for Traffic Flow Control

Rashmi Sehrawat
Computer Science & Engineering
B.S Anangpuria Institute of Technology and Management
Faridabad,India
Rashmi.sehrawat04@gmail.com

Honey Malviya
Computer Science & Engineering
B.S Anangpuria Institute of Technology and Management
Faridabad,India

Vanditaa Kaul
Computer Science & Engineering
B.S Anangpuria Institute of Technology and Management
Faridabad,India

***Abstract*-The conventional method of operation of a typical traffic light is to distribute the time equally for all the directions.This method causes congestion when throughput of the signal increases and is also ineffective in managing traffic flow. In this paper, we have proposed a new model for managing traffic intelligently.The model is based on Turing machine with the application of neural network. The model considers current traffic status of its own signal along with the status of its adjacent signals to determine the ratio of time slot for each signal therefore, reducing traffic congestion to a greater extent and ensuring steady flow of traffic in a wide region.**

Index Terms- Intelligent Traffic Flow Control, Turing Machine, Neural Network, Multilayer Feed-forward Neural Network, Sigmoid Activation Function.

I. INTRODUCTION

Now a day's traffic congestions are most cities growing problem despite continuing modernization of road infrastructure. Rapid increase of number of vehicles causes rapid increase of road traffic intensity. The road throughput and traffic regularity decreases, travelling grows and what is more the vehicles maintenance costs increases.

The proposed model solves the above problems.The model is designed to avoid traffic congestions and minimize the total time travel of the region. It collects the incoming traffic through various source and status of its adjacent signals and allows processing it and utilizing it to improve traffic flow. According to this model, each traffic signal is a Turing Machine, and the control system of this Turing machine is a Feed-Forward Artificial Neural Network.Turing Machine, similar to finite automation but with an unlimited and unrestricted memory is a much more accurate model of general purpose computer. Neural Network methods come under non mathematical models category of algorithms and are supposed to be efficient for solving the complex non-linear prediction of large scale systems.

In this paper a model for traffic flow control is introduced based on Turing Machine with application of Neural Networks. The experiment result proves that the introduced model is very effective despite of its complexity.

II. TURING MACHINE

In 1936, A. M.Turing proposed the Turing machine as a model of "any possible computation." Turing machine is an automation whose temporary storage is a tape. This tape is divided into cells. Each cell is capable of holding one input. A read-write head associated with the tape and can move left or right and can read/write a single symbol on each move. TM has a finite control which can be in any of a finite set of states. A move of the Turing machine is a function of the state of the finite control and the tape symbol scanned. In one move, The Turing machine may Change state, write a tape symbol in the cell and move the tape head left or right. Figure 1 shows an instance of Turing machine.

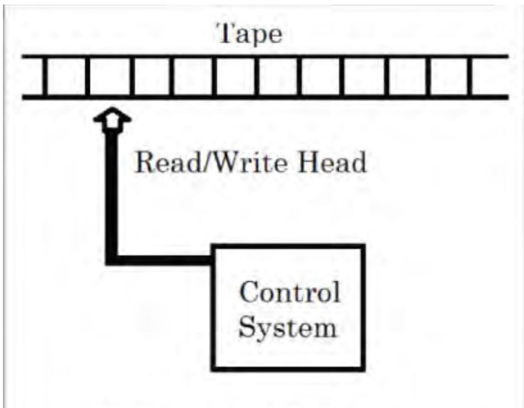

Fig1: Turing Machine

Formally, a Turning Machine is a 7-tuple, such that

TM = (Q, $\sum$, Γ, δ, q_0, B, F)

Q is finite set of states.

$\sum$ is finite set of Input symbols.

Γ is complete set of Tape symbol, $\sum$ is always a subset of Γ.

δ: (Q x Γ) $\rightarrow$ (Q x Γx{L, R}) is the transition function.

q_0 is start state, a member of Q.

B is blank symbol. This symbol is in Γ but not in $\sum$.

F is set of final or accepting states, a subset of Q.

Alternative definitions of Turing Machine abound, including versions with multiple tapes or with non-determinism. They are called variants of Turing Machine Model. The original model and its reasonable variants all have the same power - they recognize the same class of languages. The variant of Turing Machine we have used is a multi-tape Turing Machine, like an ordinary Turing Machine with several tapes. Each tape has its own head for reading and writing.

Description of the variant of Turing Machine used

In our model, all the traffic signals are a Turing Machine.Assuming that each traffic signal has *N* adjacent traffic signals in *N* directions, each Turing Machine has *2N+1* tape. N tapes for incoming traffic and other N for status of the adjacent traffic signal and a tape for its own status.The machine can be in one of the possible states depending on the traffic signal throughput and magnitude of remaining traffic.The symbols on the tapes are dynamically written based on the vehicle perception in every direction and status symbols of the adjacent machine.The input symbol on the tapes represents the magnitude of the incoming traffic in every direction and status of the adjacent machine.Table 1 shows the symbols used to represent the magnitude of traffic on a road.

Magnitude of Traffic	Symbol
Less than 5%	A
5% - 10%	B
11% - 15%	C
16% - 20%	D
21% - 25%	E
26% - 30%	F
31% - 35%	G
36% - 40%	H
Greater than 40%	X

Table 1

The control system implements neural network which is discussed following this section. The heads of the machine are read/write. All the Turing machines are connected to their adjacent Turing machines to share their current state. Each machine processes inputs from the tapes written dynamically on them to provide ratio of time slot for each direction and share its current status with the adjacent machines. The transition function is dynamically evaluated based on the inputs from the input tapes and following operations are triggered.

1. Neural network based control system determine the ratio of time slots for each direction.
2. Next state of the machine is determined based on the remaining traffic after the calculated time slot finishes.
3. Current state of the machine is shared with the adjacent machines.
4. Machine then waits for perception and start again.

III. NEURAL NETWORK

Multilayer Feed-forward Neural Networks (MFNNs) are acyclic networks that implement input-output mappings. The basic multilayer Feed-forward Neural Network approximates output data by transforming input data, and consists of an input layer, one or more hidden layers, and an output layer. Each layer contains nodes or neurons.

Interconnections within the network are such that neurons in layer *i* are connected to neurons in layer *i+1*, that is, each neuron in layer in *i* is connected to every neuron in the adjacent layer *i+1*. Each interconnection has a scalar weight associated with it that is adjusted during the training phase. The number of hidden layers determines the number of layers of weights found in the network. In the most frequently invoked MFNN model, each non-input neuron calculates its output by applying a sigmoid function (such as hyperbolic tangent) to its net weighted input. Figure 2 shows the architecture of a two-layer MFNN.

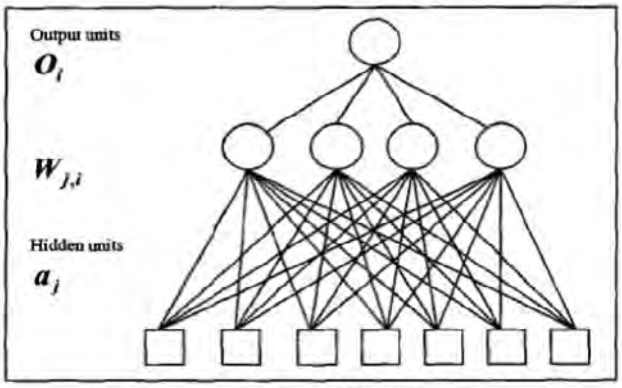

Fig. 2 A two Layer Feed-forward Neural Network

The nature of the problem being solved and the complexity of the function being approximated determine the number of input and output neurons, as well as the number of hidden layers.

An Artificial neural network is typically defined by three types of parameters:

1. The interconnection pattern between different layers of neurons.
2. The learning process for updating the weights of the interconnections.
3. The activation function weighted input to its output activation.

Mathematically, a neuron's network function is defined as a composition of other functions, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. These can be interpreted in two ways.The two views are largely equivalent.

The first view is the function view: the input is a p-dimensional vector which is transformed into a q-dimensional vector, which is then transformed into r-dimensional vector and so on and is finally transformed into output. This view is most commonly encountered in the context of optimization.

The second view is the probabilistic view: the random variable *F* depends upon the random variable *G*, which depends on the random variable *H*, and so on. Therefore the second last random variable in the dependency depends on the random variable *X*. this view is most commonly encountered in the context of graphical models.

Neural Network used in the control system

In this model, we have used a three layer Feed-forward neural network. It consists of an input layer, an output layer and two hidden layer. The input layer contains eight nodes, two hidden layers contain four nodes each and output layer contains two nodes. Fig 3 shows a model of neural network.

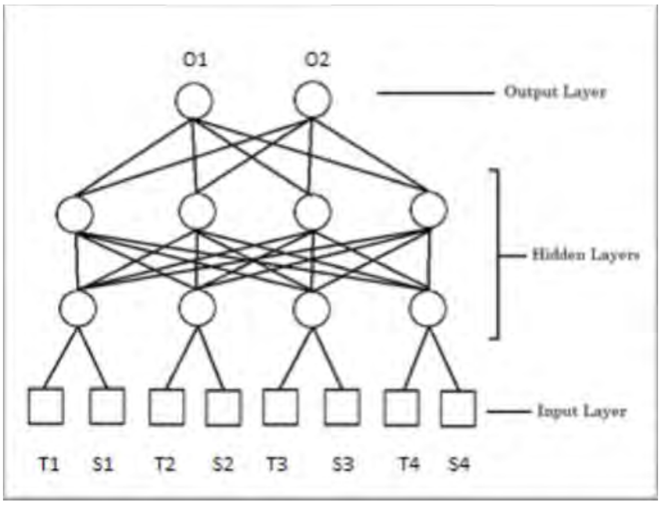

Fig3. 3 layer Feed-forward neural network

The eight input nodes in the network correspond to eight tapes of the Turing machine. The symbols read from the tapes represent the magnitude of traffic in corresponding directions and status of the adjacent traffic signal in that direction.The output O_1 is mapped to the current status of the signal and output O_2 give the ratio of time slot. For activation function, unipolar sigmoid function is used. A sigmoid function can be expressed as

$$f(x)=\frac{1}{1+e^{-x}}$$

The sigmoid activation function only returns positive values in range [0, 1].

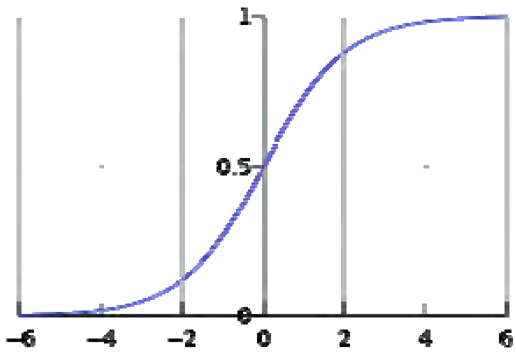

Fig 4. Sigmoidal Activation Function

IV. PROPOSED WORK

This model is simulated on a virtual traffic environment. The environmental behaviour is randomized and all the possible scenarios are simulated. Overall traffic congestion and average time travel is calculated to evaluate the model's parameter such as weights. The simulation of the proposed Turing machine with neural network as its control system is tested against different scenarios. The model is compared with conventional method under same simulated environment conditions and other related models of neural network. Three different locations L_1, $L_2$and L_3 consisting five, nine and 15 signals are simulated. The topologies used to in all the location are quite symmetric for providing simplicity to the simulation model.

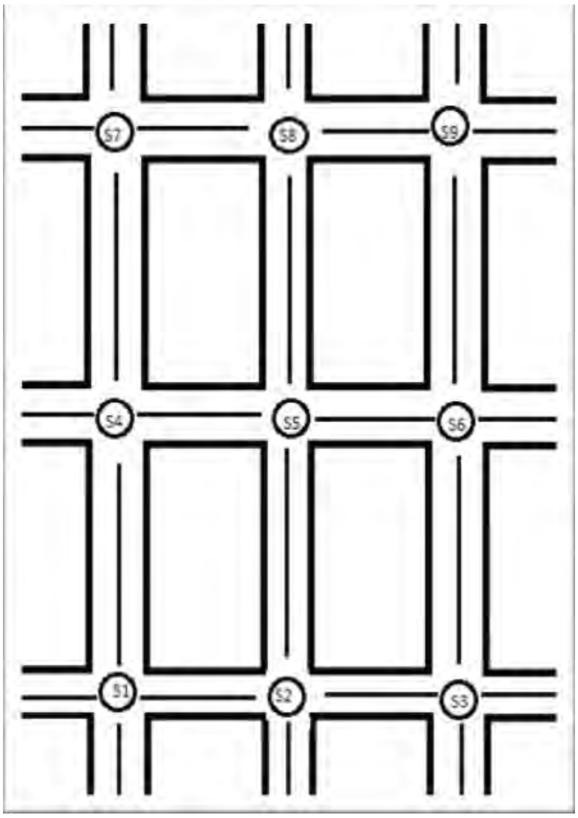

Fig 5. Virtual Traffic Environment for Location L2

In the above S_i represents a traffic signal. Each S_i is a proposed Turing machine model with neural network as its control system. These dynamic Turing machines share their status signal or more precisely, their current state with their adjacent machines. Weights of the neural networks are varied and traffic congestion and average time to travel is calculated which is used for evaluation. After a number of iterations, a set of weight is determined. This model minimized congestion and total time travel to cross the region. After these experiments, the model is then tested for locations L_1 and L_3.

The weights between the input layer and first hidden layer be denoted by W_{1a} such that it denotes the weight between first input node and first node in the hidden layer then, we can have

$$W_{1a}=W_{3b}=W_{5c}=W_{7d}$$
And
$$W_{2a}=W_{4b}=W_{6c}=W_{8d}$$

And the weights between two hidden layers are denoted as W_{ap} then, we can have

$$W_{ap} = W_{bq} = W_{cr} = W_{ds}$$
And
$$W_{aq}=W_{a\,r}=W_{as}=W_{bp}=W_{br}= W_{bs}= W_{cp}= W_{cq}= W_{cs}= {}_{Wdp}=W_{dq}= W_{dr}$$

The weights between output layer and upper hidden layers are equal. We simulated the model on location L_2 on the following set of weights.

S. No.	W_{1a}	W_{2a}
1.	0.5	0.5
2.	0.6	0.4
3.	0.7	0.3
4.	0.8	0.2
5.	0.9	0.1

Table 2

S. No.	W_{ap}	W_{aq}
1.	0.9	-0.1
2.	0.8	-0.1
3.	0.7	-0.1
4.	0.6	-0.1
5.	0.9	-0.2
6.	0.8	-0.2
7.	0.7	-0.2
8.	0.9	-0.3

Table 3

V. RESULTS

We get following results after simulating this model. The table below represents the traffic congestion in conventional and proposed model.

Location	Congestion in conventional model	Congestion in proposed model
L_1	22.53 % to 15.87%	17.83% to 5.90%
L_2	38.27% to 7.98%	19.37% to 8.90%
L_3	44.89% to 24.46%	22.56% to 7.84%

TABLE 4

The above result clearly shows that that total time travel to cover the region is reduced to a greater extent. The proposed model in better than the convention model and minimizes traffic congestion extensively.

VI. CONCLUSION

The proposed model of a Turing Machine with neural network as its control system can be used to solve a Variety of problems and provide a powerful system. The capabilities of Turing Machine and Neural Network can be used simultaneously and therefore provide us an opportunity to exploit their capacity. The proposed model in an example of it and this model can be extended and modified to solve other problems.

VII. References

[1] KishanMehrotra, Chilukuri K. Mohan, and Sanjay Ranka, *Elements of Artificial Neural Network:* MIT Press,1997.

[2] P. Webos, “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences” (Ph.D., Harvard University, 1974).

[3] Artificial Neural Networks by Dr. B. Yegnanarayana.

[4] Neural Networks – A comprehensive Foundation by Simon Haykin.

[5] Anuja Nagare, Shalini Bhatia “Traffic Flow Control Using Neural Network” in International Journal Of Applied Information System(IJAIS)-ISSN-2249-0868,Foundation Of Computer Science FCS,New York,USA,Volume1-No.2,January 2012.

[6] http://www.ijais.org/archives/volume1/number2/68-0115

[7] http://www.cs.uky.edu/~lewis/texts/theory/computability/tur-mach.pdf.

AUTHORS

First Author – Rashmi Sehrawat, Student(M.Tech), B.S.Anangpuria Institute of Technology & Management , rashmi.sehrawat04@gmail.com

Second Author – Honey Malviya, Student(B.Tech),
B.S.Anangpuria Insitute of Technology & Management.
hny.ken@gmail.com

Third Author –Vanditaa Kaul, Lecturer,B.S.Anangpuria Insitute of Technology & Management.
vanditaa.kaul@gmail.com