Deep Learning Neural Networks for Automatic Vehicle Incident Detection

Daehyon Kim¹⁾

Abstract

Recently, traffic Incidents have a significant negative impact on traffic flow safety. To realize a more accurate and faster incident detection mechanism, various types of AVID(Automatic Vehicle Incident Detection) algorithms have been proposed and ANNs (Artificial Neural Networks) are currently known to have the potential to achieve significantly good performance in terms of detection and false alarm rates, as well as operational improvements. Moreover, recently DNNs (Deep Neural Networks) have received huge attention due to their excellent performance on a wide range of recognition and classification tasks. In this study, DNNs (Deep Neural Networks) have been used to explore the performance of learning algorithm for automatic incident detection on freeways. The incident detection performance of the DNNs was compared with a general Back-Propagation (BP) model and the best network topology has been proposed for each model.

Keywords : AVID (Automatic Vehicle Incident Detection), Artificial Neural Networks (ANNs), DNNs (Deep Neural Networks), Back-Propagation (BP) Model, Network Topology

1. Introduction

Road traffic incidents disrupt the traffic flow with significant cost and cause secondary accidents. To diminish the impacts of incidents, a traffic management center needs to quickly detect and remove the incidents. In this vein, quick and accurate automatic incident detection has been one of the very important goals in transportation engineering and research for many years.

Many algorithms for automatic incident detection using roadway-based point data have been developed over the last three decades[1-3]. Until recently, artificial neural networks (ANNs) have been one of the most attractive algorithms for automatic incident detection on freeways and shown improved performance in terms of DR (Detection Rate) and FAR (False Alarm Rate) [4-7]. ANNs could provide faster and more reliable incident detection and fault-tolerant

Received(July 31, 2018), Review Result(1st: August 13, 2018, 2nd: September 4, 2018), Accepted(September 10, 2018)

1) (Professor, Corresponding Author) Dept. Ocean Civil Engineering, Chonnam National University, Yeosu-Si, Jeollanam-Do, Korea
email: daehyon@chonnam.ac.kr

operation, while being easy to implement for real-time detection systems.

Even though a large number of neural network models and learning mechanisms have been proposed, the multilayer feedforward network using Back-Propagation (BP) has been the most popular neural network model and also has been frequently used to provide powerful tools for freeway automatic incident detection[8][9]. Recently, however, Deep Neural Networks (DNNs) have been achieving state-of-the-art performance on a variety of pattern recognition classification tasks[10], and have shown remarkable performance in extracting and representing high-level abstractions in complex data[11].

The primary objective of the study is to explore the performance of DNNs, which has been used successfully in many applications, and investigate whether this model is applicable to freeway automatic incident detection (AID) using roadway-based point data. The performance of deep neural networks (DNNs) model will be compared to the performance of Back-Propagation (BP) model which is the best-known neural network model on test data sets via experiments with real world incident and incident-free data.

2. Back-Propagation (BP) and Deep Neural Networks (DNNs) Model

2.1 Back-Propagation (BP) Model

The multilayer feedforward using Back-Propagation (BP) learning algorithm, usually called simply Back-Propagation (BP), is one of the most popular neural networks and is widely applied in various problems[12]. In the standard Back-Propagation (BP) model, the Flat-Spots problem posed by Fahlman[13] presents a severe limitation on the convergence speed of the network. In order to solve the Flat-Spots problem and increase the speed of convergence, Fahlman[13] proposed an advanced Back-Propagation (BP) model called Quickprop. It has also been shown that the BPMP (Back-Propagation with Momentum & Prime-offset) which is added the Prime-offset parameter to the Back-Propagation (BP) with Momentum can be more efficient than other Back-Propagation (BP) models in terms of prediction accuracy and computing cost for training[14]. To reduce computing costs for training and obtain better predictive performance, the BPMP model was used in this study in place of the Back-Propagation (BP) model.

The algorithm of the BPMP (Back-Propagation with Momentum & Prime-offset) model can be summarized as follows[14]:

Step 1: Randomize initial weights and set up the input and output vectors

Step 2: Calculate output value for each unit of the network by using

$$t_k^l = f_p^l(Net_{pk}^l), \text{ where } Net_{pk}^l = \sum_{j=1}^{K_l-1} w_{kj}^l I_{pj}^l + \theta_k^l \quad (1)$$

In Equation (1), I_{pj}^l are inputs to the k th unit in the layer l , w is the number of synaptic weights in the network, θ_k^l is a bias term, K_l is the number of l layer units, p is a training pattern and $f()$ is an activation function.

Step 3: For the output layer, $l = L$, calculate the values of weight changes by using

$$\Delta_p w_{kj}^L = \eta \delta_{pk}^L I_{pj}^L, \text{ where } \delta_{pk}^L = (y_{pk} - Out_{pk}^L)(f_k^L(Net_{pk}^L))' \quad (2)$$

In Equation (2), y_{pk} is the desired output.

Step 4: For the hidden layers, $l = 1, \dots, L-1$, calculate the values of weight changes using

$$\Delta_p w_{ju}^l = \eta \delta_{pj}^l I_{ij}^l, \text{ where } \delta_{pj}^l = (f_k^L(Net_{pk}^L))' \sum_{k=1}^{K_{l+1}} \delta_{pk}^{l+1} w_{kj}^{l+1} \quad (3)$$

Step 5: Update weights on the output layers by

$$w_{kj}^L(t+1) = w_{kj}^L(t) + \Delta_p w_{kj}^L \quad (4)$$

Step 6: Update synaptic weights on hidden layers by

$$w_{ju}^l(t+1) = w_{ju}^l(t) + \Delta_p w_{ju}^l, l = 1, \dots, L-1 \text{ for } l = 1, \dots, L-1 \quad (5)$$

Step 7: Repeat previous steps, until the average squared error computed over the entire training data set is at an acceptably small value. The error for the output units is calculated by

$$Err = \sum_{k=1}^K (y_{pk} - Out_{pk}^L)^2 \quad (6)$$

2.2 Deep Neural Networks(DNNs) Model

Hinton and Salakhutdinov[15] developed Deep Belief Networks (DBNs) which are composed of several layers of Restricted Boltzmann Machines (RBMs). The Deep Belief Networks (DBNs) are types of deep neural networks and require multiple hidden layers with high number of hidden units to learn and extract features from raw data. Compared with traditional shallow neural networks, DBNs can use unlabeled data to pre-train a multi-layer generative model in unsupervised learning with Restricted Boltzmann Machines (RBMs), which can support better generalization, and they have been recently successfully applied to various fields[16][17]. The training of the DBNs is defined by two steps: pre-training and fine-tuning. The pre-training will train each Restricted Boltzmann Machine (RBM) independently, such that the output of the lower RBM is provided as input for the next higher-level RBM and so forth. The fine-tuning process will be performed by Back-Propagation (BP)[18] and the algorithm of Restricted Boltzmann Machines (RBMs) can be summarized as follows[19][20]:

Step 1: Conduct following process for $n = 1 \cdots N$ (number of data samples)

- A. set visible states to n th data sample, i.e., $v^{(n,0)} = x^{(n)}$ where $x^{(n)}$ is data sample
- B. compute hidden probability $q_j^{(n,0)}$ by using Eq. (7) for all j (number of hidden units)

$$q_j: p(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij}), \text{ where } \sigma(x) = 1/(1 + \exp(-x)) \quad (7)$$

- C. sample hidden state $h_j^{(n,0)} \in \{0, 1\}$ from $q_j^{(n,0)}$ for all j .

Step 2: Calculate reconstructed visible probability for all (number of visible units) by using Eq. (8)

$$p_i: p(v_i = 1|h) = \sigma(a_i + \sum_j h_j w_{ij}), \text{ where } \sigma(x) = 1/(1 + \exp(-x)) \quad (8)$$

Step 3: Compute hidden probability by using Eq. (7) for all (number of hidden units)

Step 4: Compute expectations over data distribution for all i and j by using following equation, Eq. (9)

$$q_j \rangle_{data} = \frac{1}{N} \sum_n v_i^{(n,0)} q_j^{(n,0)}, \langle v_i \rangle_{data} = \frac{1}{N} \sum_n v_i^{(n,0)}, \langle q_j \rangle_{data} = \frac{1}{N} \sum_n q_j^{(n,0)} \quad (9)$$

Step 5: Compute expectations over reconstructions for all i and j by using following equation.

$$\langle v_i q_j \rangle_{recon} = \frac{1}{N} \sum_n v_i^{(n,1)} q_j^{(n,1)}, \langle v_i \rangle_{recon} = \frac{1}{N} \sum_n v_i^{(n,1)}, \langle q_j \rangle_{recon} = \frac{1}{N} \sum_n q_j^{(n,1)} \quad (10)$$

Step 6: Compute changes in weights and biases for all i and j by using following equation.

$$\Delta w_{ij} = \varepsilon (v_i q_j \rangle_{data} - \langle v_i q_j \rangle_{recon}), \Delta a_i = \varepsilon (\langle v_i \rangle_{data} - \langle v_i \rangle_{recon}), \Delta b_j = \varepsilon (\langle q_j \rangle_{data} - \langle q_j \rangle_{recon}) \quad (11)$$

, where ε is learning rate, a_i is bias of i^{th} visible unit and b_j is bias of j^{th} hidden unit.

Step 7: Apply changes in weights and biases for all i and j by using Eq. (12)

$$w_{ij} = w_{ij} + \Delta w_{ij}, a_i = a_i + \Delta a_i, b_j = b_j + \Delta b_j \quad (12)$$

3. Experiments and Results

3.1 Input and Output Data Sets

This study used the incident and incident-free data which have been used in Kim et al.[21]. The traffic data were collected from video image processors placed on one of expressways in Seoul, Korea.

The video image processor installed on the freeway uses loop emulation, which sets the detection zone and provides three traffic parameters; traffic volume, speed and occupancy, for every lane of the expressway. The detectors were spaced at distances of about 500 meters for freeway basic sections and about 250 meters for tunnels. Only 31 incident data with a relatively accurate occurrence time of the event were used in the experiments in this study. In 31 incident data, parameter value changes between the upstream and downstream detectors can be distinguished at the time the incident occurs. In addition, 200 non-event data were used in

the experiment.

All the test data set in order to properly evaluate the neural network model must not be included in the training set. In order to satisfy this requirement and overcome a lack of incident data, the following method was used: A total 31 incident and 200 incident-free data sets were randomly split into two subsets, Data sets I and II - one for training, the other for testing. In the first experiment, learning was performed on Data set I, with the network performance evaluated on Data set II. In the second experiment, learning was performed on Data set II, with model performance evaluated on Data set I. This cross-validation check may improve the reliability of the experimental results[21].

The experimental data include the traffic volume, speed and occupancy on each lane of three stations at 30-second intervals. The cyclic traffic volume, speed and occupancy collected at three detector stations, including one more upstream station as well as the upstream and the downstream detector stations where the incident occurred, have been used in this study. In addition, the input vectors use each parameter data of every lane during four time cycles, i.e. the current cycle, t , and three previous cycles, $(t - 1)$, $(t - 2)$, and $(t - 3)$. Therefore, input vectors, x , consist of 72 units for 3 lanes, 3 stations, 3 traffic parameters and 4 times intervals [21]. On the other hand, the output vectors were defined as two different cases, with two units, i.e. $y1=[0 \ 1]$, $y2=[1 \ 0]$, for an incident and incident-free, respectively.

3.2 Network Topology

For the application of Back-Propagation (BP) and Deep Network Networks (DNNs), network topology should be selected prior to the experiment. In order to compare the algorithm performance of Back-Propagation (BP) and Deep Belief Networks (DBNs) precisely, this study used three types of network topology; i.e. 72(input units) - 144(hidden units) - 2(output units), 72(input units) - 144(hidden units) - 288(hidden units) - 2(output units), and 72(input units) - 144(hidden units) - 288(hidden units) - 576(hidden units) - 2(output units). The number of hidden units was determined with the following rule; $2x$ (1st hidden layer) - $4x$ (2nd hidden layer) - $8x$ (3rd hidden layer), where x is the number of input units.

3.3 Implementation and Results

In order to evaluate and compare the performance of two neural network models, Back-Propagation (BP) and Deep Brief Network, 30 trials were implemented on the same

network with the same value for parameters but different initial weights initialized to random values. The test results were statistically analyzed by ANOVA using a post-hoc test - Tukey HSD (honest significant difference) - at a significance level of $P = 0.05$, in order to assess the statistical significance.

In this study, two MOEs (measures of effectiveness) were used for incident detection, DR (Detection Rate) and FAR (False Alarm Rate). DR (Detection Rate) is defined as the number of incidents detected by the algorithm divided by the total number of incidents known to have occurred during the recording time and FAR (False Alarm Rate) is defined as the number of incident-free intervals that give false alarms divided by the total number of incident-free intervals.

[Table 1] and [Fig. 1] show the results of the two different models, Back-Propagation (BP) and Deep Brief Network. They also show how performance varies with topology in two neural network models, Back-Propagation (BP) and Deep Brief Network.

The predicted error rate in the table represents the average recognition error rate for different experiments using a new random initial weight (a total of 30 trials) in the test set (the number of error/the number of test sets). As shown in the table, the variance of Back-Propagation (BP) model is large in some network architectures, such as 72-144-288-2 and 72-144-288-576-2 network architectures. This large variance was caused by a large number of learning failures during the experiment. There were 4 and 12 learning failures for the four-layer network 72-144-288-2 and the five-layer network 72-144-288-576-2, respectively. However, there was no learning failure in the Deep Brief network model. This means that the Deep Brief Network model is effective for complex network architectures and the Back-Propagation (BP) model can only be applied to simple network architectures. Also, the experimental results showed that the detection rate ranged from 72.16% to 74.61% and the FAR (False Alarm Rate) from 1.47% to 1.99%. Two Back-Propagation (BP) networks, BP (72-144-288-2) and BP (72-144-288-576-2) with learning failures, were excluded from the analysis.

According to the ANOVA test results, there is no statistically significant difference in the four models, BP (72-144-2), DBN (72-144-2), DBN (72-144-288-2), and DBN (72-144-288-576-2), using input vectors. The ANOVA indicates that there is no significant difference in network performance between the DBN (72-144-288-2) model and the DBN (72-144-288-576-2) model.

Moreover, the variance of DBNs models was very small; i.e. the variance of DBN (72-144-288-2) and DBN (72-144-288-576-2) are 0.03 and 0.0 respectively. This implies that the performance of DBNs models is less affected by the initial weights. The influence of initial weights is one of unsolved issues of general Back-Propagation (BP) model and DBNs models

can be effective in solving the initial weight problem.

[Table 1] Prediction Performance on Different Architecture and Learning Method

Network Topology		72-144-2		72-144-288-2		72-144-288-576-2	
Neural Network model		BP	DBN	BP*	DBN	BP**	DBN
Incident Data	Prediction Error Rate	8.63/31	7.87/31	8.03/31	8.0/31	23.93/31	8.07/31
	Variance	0.309195	0.326437	2.24023	0.0	90.2023	0.064368
Incident - Free Data	Prediction Error Rate	3.23/200	3.97/200	15.73/200	3.0/200	43.97/200	2.93/200
	Variance	0.46092	0.033333	1137.444	0.068966	3862.378	0.064368
Total	Prediction Error Rate	11.86/231	11.84/231	23.76/231	11.0/231	67.9/231	11.0/231
	Variance	0.671264	0.281609	1044.254	0.068966	2783.61	0.0
	Groupings	A	A	B	A	C	A
False Alarm Rate (%)		1.62	1.99	7.87	1.50	21.99	1.47
Detection Rate (%)		72.16	74.61	74.10	74.19	22.81	73.97

Note: *There are 4 learning failures in the experiment. **There are 12 learning failures in the experiment.

(A) False Alarm Rate

4. Conclusions

Reliable and accurate auto accident detection is one of the most important goals in transportation research and so far artificial neural networks (ANNs) have been successfully applied to automatic accident detection.

(B) Detection Rate

[Fig. 1] Performance of Back-Propagation (BP) and Deep Brief Networks (DBNs)

Although Back-Propagation (BP) has been the most common neural network model so far, recent research has shown that Deep Neural Networks (DNNs) can be used more reliably in various pattern recognition classifications.

In this study, Deep Neural Networks (DNNs) were used for automated incident detection using road-based point data such as volume, speed, and occupancy. Deep Neural Networks (DNNs) and Back-Propagation (BP) model were compared using two MOEs (measures of effectiveness) for incident detection, DR (Detection Rate) and FAR (False Alarm Rate). Experimental results show that DNN model is superior to BP in network configuration and initialization of input vectors. DBN (Deep Brief Network), which is one of Deep Neural Networks, can be an effective model to solve the initial weight problems that occur in Back-Propagation (BP) models and to improve the stability of predictive performance.

Back-Propagation (BP) models are not suitable for complex network architectures (over 3 hidden layers). However, the Deep Brief Network (DBN) is suitable for complex network architectures with multiple hidden layers.

The best performance in terms of FAR (False Alarm Rate) has been achieved in Deep Brief Network (DBN) with an architecture of 72-144-288-576-2. The best performance in terms of DR (Detection Rate) has been achieved in Deep Brief Network (DBN) models with an architecture of 72-144-2.

References

- [1] H. J. Payne, S. C. Tignor, Freeway incident detection algorithms based on decision trees with states, *Transportation Research Record*, **(1978)**, Vol.682, pp.30-37.
- [2] B. N. Persaud, F. L. Hall, Catastrophe theory and patterns in 30-second freeway traffic data - implications for incident detection, *Transportation Research Part A*, **(1989)**, Vol.23, No.2, pp.103-113.
- [3] Y. J. Stephanedes, A. P. Chassiakos, Freeway incident detection through filtering, *Transportation Research Part C*, **(1993)**, Vol.1, pp.219-233.
- [4] R. L. Cheu, S. G. Ritchie, Automated Detection of Lane-Blocking Freeway Incidents using Artificial Neural Networks, *Transportation Research Part C*, **(1995)**, Vol.3, No.6, pp.371-388.
- [5] S. Ishak, H. Al-Deek, Performance of automatic ANN-based incident detection on freeways, *Journal of Transportation Engineering*, **(1999)**, Vol.125, No.4, pp.281-290.
- [6] Q. Wang, Traffic incident detection based on Artificial Neural Network, *IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, **(2011)**, May 27-29; Xian, China, pp.657-659.
- [7] J. Lu, S. Chen, W. Wang, B. Ran, Automatic traffic incident detection based on nFOIL, *Expert Systems with Applications*, **(2012)**, Vol.39, pp.6547-6556.
- [8] L. Yu, L. Yu, J. Wang, L. Yu, Y. Qi, H. Wen, Back-Propagation Neural Network for Traffic Incident Detection Based on Fusion of Loop Detector and Probe Vehicle Data, *ICNC Fourth International Conference on Natural Computation*, **(2008)**, August 25-27; Jinan, China, Vol.3, pp.116-120.
- [9] Z. Zhang, X. Lin, B. Hu, Algorithm design of traffic incident automatic detection based on mobile detection, *IEEE International Conference on Service Operations, Logistics, and Informatics (SOLI)*, **(2011)**, July 10; Beijing, China, pp.331-335.
- [10] A. Nguyen, J. Yosinski, J. Clune, Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images, In *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, **(2015)**, pp.1-20.
- [11] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **(2015)**, Vol.521, No.7553, pp.436-444.
- [12] D. E. Rumelhart, J. L. McClelland, PDP Research Group (Eds), *Parallel Distributed Processing*, Cambridge, Massachusetts: MIT Press, **(1986)**
- [13] S. E. Fahlman, An empirical study of learning speed in backpropagation networks, Technical Report CMU-CS-88-162, Canegie Mellon University, **(1988)**
- [14] D. Kim, Standard and Advanced Backpropagation models for image processing application in traffic engineering, *ITS Journal*, **(2002)**, Vol.7, No.3-4, pp.199-211.
- [15] G. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, **(2006)**, Vol.313, No.5786, pp.504-507.
- [16] A. Mohamed, Dahl, G. Hinton, Acoustic modeling using deep belief networks, *IEEE Transactions on*

Audio, Speech, and Language Processing, **(2012)**, Vol.20, No.1, pp.14-22.

- [17] Y. Ding, S. Chen, J. Xu, Application of Deep Belief Networks for Opcode Based Malware Detection, 2016 International Joint Conference on Neural Networks (IJCNN), **(2016)**, July 24-29; Vancouver, Canada, pp.3901-3908.
- [18] G. Hinton, S. Osindero, Y. W. Teh, A fast learning algorithm for deep belief nets, Neural Computation, **(2006)**, Vol.18, No.7, pp.1527-1554.
- [19] G. Hinton, A Practical Guide to Training Restricted Boltzmann Machines, UTM TR 2010 - 003, University of Toronto, **(2010)**
- [20] T. Amaral, Experiments with a restricted Boltzmann machine, NNIG, Technical Report No. 1/2012, Instituto de Engenharia Biomédica (INEB), Portugal, **(2012)**
- [21] D. Kim, S. Lee, S. Cho, Input Vector Normalization Methods in Support Vector Machines for Automatic Incident Detection, Transportation Planning and Technology, **(2007)**, Vol.30, No.6, pp.593-608.