Available online to ejournal.pnc.ac.id

Journal of Innovation Information Technology and Application

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

Traffic Image Analysis with Stacked Denoising Autoencoder Neural Network

Daehyon Kim

[1] *Department of Culture and Tourism Management, Chonnam National University, Yeosu, Korea*

email:[1] *daehyon@chonnam.ac.kr*

ARTICLE INFO

Article history:
Received 19 November 2023
Revised 28 December 2023
Accepted 28 December 2023
Available online 30 December 2023

Keywords:
Neural Network
Stacked Denoising Autoencoding
Deep Belief Network
Backpropagation

IEEE style in citing this article: [citation Heading]
D. Kim, " Traffic Image Analysis with Stacked Denoising Autoencoder Neural Network," *Journal of Innovation Information Technology and Application*, vol. 5, no. 2, pp. 183-192, 2023.

ABSTRACT

This study aims to explore major neural network models - Stacked Denoising Autoencoder (SDA), Deep Belief Network (DBN), and Backpropagation - that have recently garnered attention and propose the most suitable and reliable artificial neural network model for real-time road traffic information collection. To enhance the reliability of experimental results, numerous experiments were conducted in this study under identical conditions (such as parameter values and network configuration) by setting different initial values for the weight vector. The results of the experiments were statistically validated to conclude. The research results showed that the SDA model exhibited the most superior performance, while the accuracy of the DBN was somewhat lower compared to the SDA model. On the other hand, the Backpropagation model demonstrated a relatively low predictive accuracy compared to both models, particularly showing a significant influence of the initial values.

1. INTRODUCTION

Intelligent Transportation Systems (ITS) integrate advanced technologies such as electronics, control systems, and communication into conventional components of transportation systems, including roads, vehicles, and signal systems. ITS aims to enhance the efficiency of transportation facilities and increase safety, representing the next generation of transportation systems. Autonomous driving, a topic of recent global interest, represents the ultimate goal to be achieved within Intelligent Transportation Systems (ITS). To realize this goal, numerous research efforts and investments are being concentrated worldwide. Real-time and accurate information collection regarding road and traffic conditions is crucial for the intelligence of roads and autonomous driving. Various technologies have evolved to collect real-time traffic information such as traffic volume, speed, vehicle trajectory tracking, congestion measurement, and sudden event detection on roads. Currently, the widely adopted technology both domestically and internationally is loop detectors. However, due to several drawbacks of loop detectors, there is a growing demand for alternative detection systems. The disadvantages of loop detectors include frequent system errors due to road surface damage caused by contact and burial in the road surface, as well as difficulties in installation and maintenance. A range of alternative detection technologies, including video detectors, microwave detectors, and infrared detectors, are under consideration. Among these technologies, video detectors are recognized as the most effective alternative. In particular, the video processing technology that automatically extracts the necessary traffic information from road videos collected in real-time through

**) Corresponding Author : daehyon@chonnam.ac.kr*

video cameras is considered a crucial technology not only for autonomous driving but also for establishing intelligent transportation systems [1].

Despite various research efforts utilizing image processing technology for real-time traffic information collection, developing an algorithm that can perceive the complex environment of roads in real-time perfectly is challenging due to factors such as shadows, occlusion, and image noise. To maximize the reliability of automatic traffic information collection in such diverse and complex visual data, effective algorithms based on machine learning are being employed, with a particular focus on artificial neural network models [2][3][4][5][6]. With the increasing interest in artificial neural network models, various forms of these models have been developed, and among them, the backpropagation model [7] has garnered considerable attention. In addition, in 2008, Hinton and Salakhutdinov [8] developed deep learning artificial neural networks called Deep Belief Networks (DBN), which are actively used in various fields to this day. Furthermore, Vincent et al. [9] introduced the Stacked Denoising Autoencoder (SDA) as a form of constructing deep networks, suggesting the potential superiority of SDA over DBN in their research results. Especially in recent years, Stacked Denoising Autoencoder (SDA) has been actively utilized in various fields [10][11][12] [13].

Numerous artificial neural network models currently used in various fields have been proposed, and the predictive capabilities of each model can significantly vary based on factors such as data characteristics and network architecture. In previous research on artificial neural network models utilizing traffic images [1], the predictive reliability of models like Backpropagation and DBN was validated. However, there has been a lack of research on the application of SDA models, which share a similar deep learning structure with DBN. Despite the effective evaluation of SDA models as artificial neural network models in deep network structures, there is insufficient research on their applicability in pattern recognition using traffic images. Therefore, this study aims to analyze the prediction ability and reliability of the SDA model suitable for collecting real-time traffic information through road image recognition, a core technology for implementing the Intelligent Transportation System (ITS). In addition, this study seeks to identify and present the characteristics of the SDA model through comparison with existing artificial neural network models. The results of this study could serve as a benchmark for evaluating the applicability of SDA artificial neural network models in various fields, including traffic engineering and planning, beyond the automatic analysis of traffic images for Intelligent Transport Systems.

2. NEURAL NETWORK MODELS

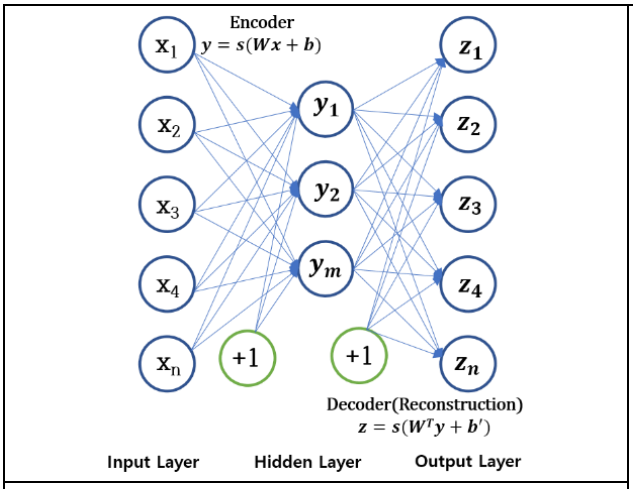
2.1. Stacked Denoising Autoencoder

The Stacked Denoising Autoencoder (SDA) was proposed by Pascal Vincent et al. in 2010 [9] and is an extension of the stacked autoencoder [14]. The core idea is to enhance the training and learning of more robust feature representations by adding noise through each layer of the encoder input. From a structural perspective, the SDA is comprised of multiple layers of an unsupervised denoising autoencoder network and layers of a supervised backpropagation neural network [13]. The learning process of the Stacked Denoising Autoencoder (SDA) consists of two stages: unsupervised learning and supervised learning.

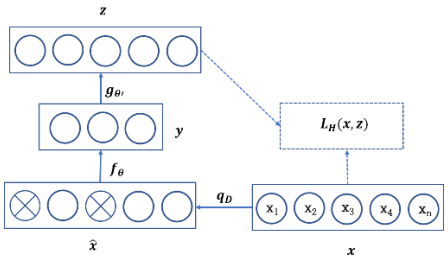
Step 1: Unlabeled samples are employed for the greedy layer-wise training of the denoising autoencoder. In this process, initial data is fed into the first layer of the denoising autoencoder for unsupervised training, and then the parameters $W^{(1)}$ of the first hidden layer are obtained. In each subsequent step, the previously trained layers up to k-1 are used as input to train the k^{th} layer and acquire the parameters $W^{(k)}$. The weights obtained from the training of each layer serve as the initialization weights for the final deep network.

Step 2: Utilizing the initial weight values established in Step 1, a backpropagation neural network is employed for supervised learning with labeled data.

Figure 1 illustrates the basic structure of an autoencoder and a denoising autoencoder. Denoising autoencoders, as shown in Figure 2[15], can be stacked to form a deep network (stacked denoising autoencoder) [9].



(A) Encoding and decoding process of an autoencoder



(B) Denoising autoencoder architecture [9]. An example x is stochastically corrupted (via q_D) to $\hat{x}$. The autoencoder then maps it to y (via encoder f_θ) and attempts to reconstruct x via decoder g_θ', producing reconstruction z. Reconstruction error is measured by loss $L_H(x, z)$.

Figure 1. Autoencoder and denoising autoencoder

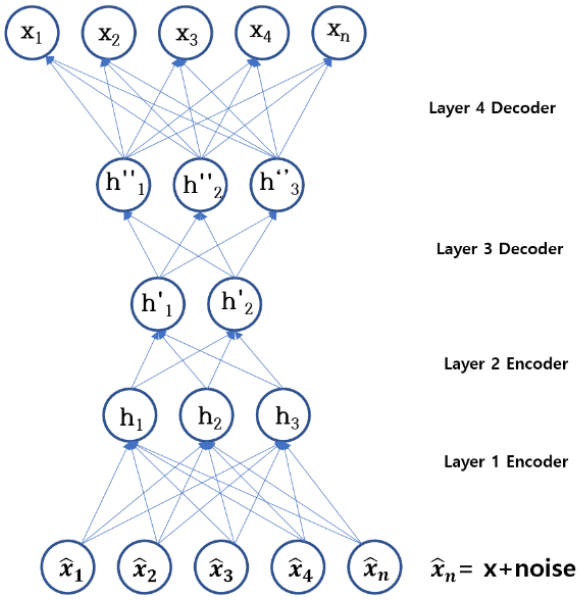


Figure 2. Stacking denoising autoencoders

The Stacked Denoising Autoencoder (SDA) is composed of multiple denoising autoencoders (DA) stacked on top of each other. In other words, the SDA is obtained by layering DAs on one another, with the hidden layer inputs of the lower-level DA derived from the hidden layer outputs of the higher-level DA [15]. The Denoising Autoencoder (DA) is a refined version of the Autoencoder designed to mitigate the risk of the network learning identity features. Specifically, when an autoencoder is excessively large, it may merely memorize the data, resulting in an output identical to the input without achieving meaningful representation learning or dimensionality reduction. Denoising autoencoders address this issue by deliberately introducing noise, corruption, or masking certain input values, thereby enhancing their ability to extract valuable features and avoid mere data memorization. The Autoencoder is an unsupervised neural network architecture that is divided into an encoding phase and a decoding phase for a given input dataset. In the learning process of Autoencoder, which belongs to the unsupervised learning domain, the goal is to reconstruct input data without labels. In the encoding phase, the model learns the properties of the input data, and in the decoding phase, the goal of the autoencoder model is to reproduce the correct data (input) using the learned features.

2.2. Deep Belief Network

Deep learning is a machine learning approach that stands in contrast to shallow learning. It focuses on learning models with deep structures within the data [16]. Deep Belief Networks (DBNs) [17] require multiple hidden layers with a high number of hidden units to learn and extract features from raw data. In contrast to Backpropagation, DBNs can utilize unlabeled data for pre-training a multi-layer generative model in unsupervised learning, employing Restricted Boltzmann Machines (RBMs) [18]. The training of DBNs comprises two steps: pre-training and fine-tuning. During pre-training, each Restricted Boltzmann Machine (RBM) is trained independently. The output of the lower RBM serves as input for the next higher-level RBM, and this process continues. The fine-tuning process is then performed using Backpropagation [19]. The algorithm for Restricted Boltzmann Machines (RBMs) can be summarized as follows [18][19].

Step 1: Conduct following process for n =1…N (number of data samples)
(a) Set visible states to nth data sample, i.e. $v^{(n.0)} = x^{(n)}$, where $x^{(n)}$is n^{th} data sample
(b) Compute hidden probability $q_j^{(n.0)}$ by using Eq. (1) for all j (number of hidden units)

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

(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 $p_i^{(n.1)}$for all i (number of visible units) by using following Eq. (2)

$$p_i: p(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij}), \text{ where } \sigma(x) = {}^{1}/_{(1 + exp(-x))} \quad (2)$$

Step 3: Compute hidden probability $q_j^{(n.1)}$ by using Eq. (1) for all j (number of hidden units)
Step 4: Compute expectations over data distribution for all i and j by using Eq. (3)

$$\langle v_i q_j \rangle_{data} = \frac{1}{N}\sum_n v_i^{(n.0)} q_j^{(n.0)},\ \langle v_i \rangle_{data} = \frac{1}{N}\sum_n v_i^{(n.0)},\quad \langle q_j \rangle_{data} = \frac{1}{N}\sum_n q_j^{(n.0)} \quad (3)$$

Step 5: Compute expectations over reconstructions for all i and j by using Eq. (4)

$$\langle v_i q_j \rangle_{recon} = \frac{1}{N}\sum_n v_i^{(n.1)} q_j^{(n.1)},\ \langle v_i \rangle_{recon} = \frac{1}{N}\sum_n v_i^{(n.1)},\quad \langle q_j \rangle_{recon} = \frac{1}{N}\sum_n q_j^{(n.1)} \quad (4)$$

Step 6: Compute changes in weights and biases for all i and j by using Eq. (5)

$$\Delta w_{ij} = \varepsilon\langle v_i q_j \rangle_{data} - \langle v_i q_j \rangle_{recon},\ \Delta a_i = \varepsilon\langle v_i \rangle_{data} - \langle v_i \rangle_{recon},\ \Delta b_j = \varepsilon\langle q_j \rangle_{data} - \langle q_j \rangle_{recon} \quad (5)$$

, where $\varepsilon$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. (6)

$$w_{ij} = w_{ij} + \Delta w_{ij},\ a_i = a_i + \Delta a_i,\ b_j = b_j + \Delta b_j \quad (6)$$

2.3. Backpropagation

Backpropagation is one of the most popular neural networks and is widely applied to various problems [7]. For the backpropagation model, Backpropagation with Momentum and Prime-offset [4] was used in this study to improve the convergence speed and to obtain better prediction performance. The algorithm of the BPMP (Back-Propagation with Momentum & Prime-offset) model can be summarized as follows [4][20]:

Step 1: Randomize initial weights and set up the input and output vectors
Step 2: Calculate the output value for each unit of the network by using

$$Out_{pk}^{l} = f_{pk}^{l}(Net_{pk}^{l}) \text{ where } Net_{pk}^{l} = \sum_{j=1}^{K_{l-1}} w_{kj}^{l} In_{pj}^{l} + \theta_k^{l} \quad (7)$$

In Equation (1), $In_{pj}^{l}$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

DOI: doi.org/10.35970/jinita.v5i2.2133

$$\Delta_p w_{kj}^L = \eta \delta_{pk}^L In_{pj}^L \quad \text{where } \delta_{pk}^L = \left(y_{pk} - Out_{pk}^L\right)\left(f_k^L(Net_{pk}^L)\right)' + \text{Prime- offset} \tag{8}$$

In Equation (8), $y_{pk}$is the desired output.

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

$$\Delta_p w_{ju}^l = \eta \delta_{pj}^l In_{ij}^l \ , \quad \text{where} \quad \delta_{pj}^l = \left(f_j^l(Net_{pj}^l)\right)' \sum_{k=1}^{K_{l+1}} \delta_{pk}^{l+1} w_{kj}^{l+1} \tag{9}$$

Step 5: Update weights on the output layers by

$$w_{kj}^L(t+1) = w_{kj}^L(t) + \Delta_p w_{kj}^L \tag{10}$$

Step 6: Update synaptic weights on hidden layers by

$$w_{ju}^l(t+1) = w_{ju}^l(t) + \Delta_p w_{ju}^l \quad \text{for } l = 1,..., L\text{-}1 \tag{11}$$

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

$$Err_p = \sum_{k=1}^{K_L} (y_{pk} - Out_{pk}^L)^2 \tag{12}$$

3. EXPERIMENTAL DESIGN

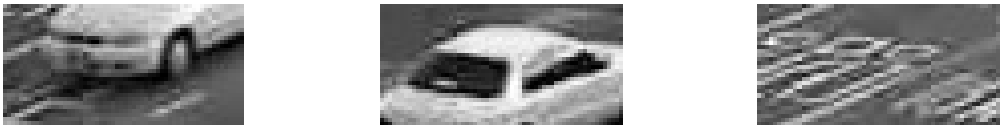
3.1. Experimental data and network configuration

In this study, the same experimental data used by Kim [4][5][20] for traffic scene analysis was employed. In this study, artificial neural networks were applied for pattern recognition of various forms appearing in the Region of Interest (ROI) of road images. The experimental task for training and testing involves recognizing three different patterns within the region of interest, each characterized as follows:

Pattern A: The front or rear bumper of a vehicle appears in the ROI

Pattern B: Appears in the ROI of the vehicle roof image

Pattern C: No part of the vehicle is visible, and only the road lane markings are present in the ROI



Pattern A Pattern B Pattern C

Figure 3. Samples of experimental data for learning and testing

In this study, 230 data sets (Pattern A: 100, Pattern B: 100, Pattern C: 30) were used for the training of the artificial neural network model, and for the reliability validation of the model after training, 700 data points (Pattern A: 300, Pattern B: 300, Pattern C: 100) were utilized.

The vectors used in this study are grayscale images of size 15 x 30 pixels, so the input units are 450. For the output vector, three units were employed to recognize three different patterns; i.e, y_1 = [1 0 0], y_2 = [0 1 0], and y_3 = [0 0 1].

The number of hidden layers and hidden neurons greatly affects training time, so it is necessary to limit them for experiments. Network architecture affects the learning and prediction capabilities of neural network models, but there is currently no theoretical method for constructing an optimal network. Therefore, in this study, considering training time, the maximum number of neurons was limited to 3,000 and the number of hidden layers was limited to 4. The networks built for the experiments in this study were compared in terms of prediction performance across five configurations. The network construction was initially based on the model proposed by Hinton and Salakhutdinov [7], with variations introduced such as halving the number of hidden neurons, reducing the number of hidden layers, and adding one more hidden layer. The experiment included a total of five scenarios. The network configuration used in the experiments of this study is as follows; i.e. 500-500-2000(3hidden layers - Hinton and Salakhutdinov's network configuration), 250-250-1000(3hidden layers), 500-2000(2hidden layers), 500-500(2hidden layers), 500-500-500-500(4hidden layers). The first experiment employed the network proposed by Hinton and

DOI: doi.org/10.35970/jinita.v5i2.2133

Salakhutdinov [7], while the remaining four were constructed within the range of 1,000 to 3,000 neurons, considering the training time and characteristics of the training data.

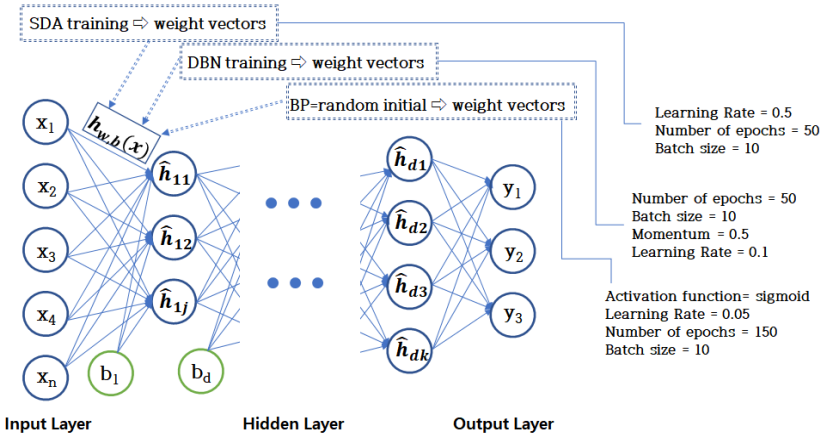


Figure 3. Learning methods and hyperparameters for each neural network model

3.2. Normalization for input vectors

The input and output vectors of neural network models and all learning algorithms must be normalized before being used in the model [21]. The most common normalization method is as shown in equation (13).

$$\overline{x_{pk}} = \frac{x_{pk} - X^p_{min}}{X^p_{max} - X^p_{min}} \tag{13}$$

where $p(p = 1, 2, \ldots, P)$ are input data sets, $X = \min(X^p_k;\ k = 1, 2, \ldots, m)$ and $X^p_{max} = \max(X^p_k;\ k = 1, 2, \ldots, m)$. In Eq. (13), X^p_{min} and X^p_{max} is the minimum and maximum values, respectively, of the input pattern and $\overline{x_{pk}}$ denotes the normalized value of the unit k of the input vector $x^p = \left(x^p_1, x^p_2, x^p_3, \ldots, x^p_m\right)$. This type of linear normalization method will use the range 0 to 1 for each input data and treat two linearly dependent data sets identically.

In particular, normalization methods play a crucial role in artificial neural network models and machine learning, exerting a significant impact on the learning capabilities and predictive accuracy of the models [21][22]. Moreover, considering the data characteristics, nonlinear models may be more suitable as a normalization model for input vectors, rather than the most common linear form [23]. Therefore, in this study, the following nonlinear model was employed as a normalization model for input vectors.

$$\overline{x_{pi}} = \frac{1}{1 + \exp\left(-x_{pi} + 128\right)} \tag{14}$$

, where $\overline{x_{pi}}$ represents the generalized value, and x_{pi} denotes the raw input value of the p-pattern data.

4. RESULTS AND DISCUSSION

In this study, neural network models such as Backpropagation, DBN, and SDA exhibit significant variations in learning ability and predictive performance based on the initialization of weights. Therefore, to appropriately compare and evaluate these models, it is essential to conduct multiple experiments using the same set of parameters while employing different initial weights on the identical network. To mitigate the impact of initial weight values, this study performed 30 trials using distinct initial weights initialized with random values between -1.0 and +1.0. The networks were trained for 150 generations on the training set, achieving remarkably low training errors and 100% recognition accuracy. To assess the statistical significance of the experimental results, an analysis of variance (ANOVA) test was conducted with a significance level of P = 0.05.

ANOVA, which stands for Analysis of Variance, is a statistical test used to analyze the difference between the means of more than two groups. A one-way ANOVA uses one independent variable, while a two-way ANOVA uses two independent variables [24].

Tables 1 to 5 show the predictive performance of the three main models used in this study, SDA, DBN, and backpropagation, across different network configurations. In this study, to eliminate the influence of initial weights, 30 trials were conducted using weight vectors with different initial weights in the same network structure.

In the network structure of 450-500-500-2000-3 (Table 1), the SDA model exhibited the highest prediction accuracy at 91.63%, while the DBN model showed a slightly lower prediction accuracy at 91.27% compared to the SDA model. The Backpropagation model demonstrated the lowest prediction accuracy at 85.61%. Although there was a subtle difference in performance between the SDA and DBN models, the ANOVA analysis revealed statistically significant differences in prediction accuracy among the three models (SDA, DBN, and Backpropagation).

On the other hand, in the network architecture of 450-250-250-1000-3(Table 2), the DBN model demonstrated an outstanding prediction accuracy of 91.68%. Following closely, the SDA model recorded a prediction accuracy of 90.89%, while the Backpropagation model exhibited the lowest performance at 86.21%. In this network configuration, statistical analysis also concluded that there are statistically significant differences in prediction accuracy among the three models (SDA, DBN, and Backpropagation).

Table 1. Performance of models on the network architecture of 450-500-500-2000-3

Category		Neural Network Models		
		SDA	**DBN**	**Backpropagation**
Training RMSE		**0.0022**	**0.0052**	**0.0010**
Prediction Errors	Pattern A	12.07	16.67	22.87
	Pattern B	39.40	42.47	57.27
	Pattern C	7.13	1.97	20.60
Number of total errors		58.60	61.10	100.73
Prediction error rate (%)		8.37	8.73	14.39
Prediction accuracy (%)		91.63	91.27	85.61
Statistical analysis	Std. Deviation	5.164	3.827	13.851
	Groups	1st	2nd	3rd
	P-value (F-value)	SDA-DBN: 0.037 (4.539), DBN-BP: 9.155E-22(228.201)		

Table 2. Performance of models on the network architecture of 450-250-250-1000-3

Category		Neural Network Models		
		SDA	**DBN**	**Backpropagation**
Training RMSE		**0.0020**	**0.0066**	**0.0015**
Prediction Errors	Pattern A	11.40	16.47	23.47
	Pattern B	41.23	39.07	63.57
	Pattern C	11.17	2.73	9.50
Number of total errors		63.80	58.27	96.53
Prediction error rate (%)		9.11	8.32	13.79
Prediction accuracy (%)		90.89	91.68	86.21
Statistical analysis	Std. Deviation	7.155	6.918	9.336
	Groups	2nd	1st	3rd
	P-value (F-value)	SDA-DBN: 0.003 (9.273), SDA-BP: 6.031E-22 (232.332)		

Table 3. Performance of models on the network architecture of 450-500-2000-3

Category		Neural Network Models		
		SDA	**DBN**	**Backpropagation**
Training RMSE		**0.0018**	**0.0058**	**0.0034**
Prediction Errors	Pattern A	11.63	12.10	22.50
	Pattern B	41.93	47.53	66.27
	Pattern C	6.07	1.80	10.10
Number of total errors		59.63	61.43	98.87
Prediction error rate (%)		8.52	8.78	14.12
Prediction accuracy (%)		91.48	91.22	85.88
Statistical analysis	Std. Deviation	5.893	7.328	11.184
	Groups	1st	1st	3rd
	P-value (F-value)	SDA-DBN:0.299 (1.099), DBN-BP: 4.563E-22(235.126)		

DOI: doi.org/10.35970/jinita.v5i2.2133

In addition, for the network architecture of 450-500-2000-3(Table 3), the SDA model exhibited a slightly superior prediction accuracy of 91.48% compared to the DBN model with an accuracy of 91.22%. However, according to statistical validation, there is no significant difference in performance between the two models. On the contrary, both models (SDA, DBN) exhibited significantly superior performance compared to the Backpropagation model (accuracy 85.88%), and this difference was statistically significant.

Additionally, for the network architecture of 450-500-500-3(Table 4), the DBN model showed slightly better prediction accuracy at 91.19% compared to the SDA model (accuracy of 91.67%). However, statistical analysis showed that there was no difference between the two models. Nevertheless, similar to other network configurations, both models (SDA, DBN) performed significantly better than the backpropagation model (accuracy 86.08%).

Finally, in the network architecture of 450-500-500-500-500-3 (Table 5), similar to the network architecture of 450-500-500-2000-3, the SDA model exhibited the highest performance with an accuracy of 91.36%. The DBN model recorded a prediction accuracy of 90.79%, while the Backpropagation model showed the lowest prediction accuracy at 86.51%. The statistical analysis results also indicated that the SDA model outperformed both the DBN and Backpropagation models.

Table 4. Performance of models on the network architecture of 450-500-500-3

Category		Neural Network Models		
		SDA	**DBN**	**Backpropagation**
Training RMSE		**0.0021**	**0.0059**	**0.0016**
Prediction Errors	Pattern A	12.67	14.13	23.40
	Pattern B	39.03	40.93	61.47
	Pattern C	9.97	3.23	12.57
Number of total errors		61.67	58.30	97.43
Prediction error rate (%)		8.81	8.33	13.92
Prediction accuracy (%)		91.19	91.67	86.08
Statistical analysis	Std. Deviation	7.617	6.271	7.868
	Groups	1st	1st	3rd
	P-value (F-value)	SDA-DBN: 0.067(3.493), SDA-BP: 2.790E-25 (319.994)		

Table 5. Performance of models on the network architecture of 450-500-500-500-500-3

Category		Neural Network Models		
		SDA	**DBN**	**Backpropagation**
Training RMSE		**0.0013**	**0.0049**	**6.0669e-004**
Prediction Errors	Pattern A	12.63	19.57	24.17
	Pattern B	40.70	43.87	61.33
	Pattern C	7.17	1.07	8.93
Number of total errors		60.50	64.50	94.43
Prediction error rate (%)		8.64	9.21	13.49
Prediction accuracy (%)		91.36	90.79	86.51
Statistical analysis	Std. Deviation	6.776	6.107	7.968
	Groups	1st	2nd	3rd
	P-value (F-value)	SDA-DBN: 0.020 (5.769), DBN-BP: 2.323E-23(266.698)		

The difference in prediction performance between SDA, DBP, and Backpropagation models is the initial weight used for training. The SDA model and DBP model are characterized by using unsupervised neural network models of Denoising Autoencoder and Restricted Boltzmann Machines, respectively, before using supervised learning models such as the Backpropagation model. On the other hand, the Backpropagation model is performed directly through a supervised learning model without prior training of an unsupervised neural network model. Therefore, as can be seen from the results of this study, the hybrid model combining unsupervised and supervised showed excellent prediction performance, and the same results will be obtained in other similar experiments. Meanwhile, in the comparison between SDA and DBP, which are hybrid forms, In the experiments of this study, SDA showed slightly better results than DBN, but there is a possibility that different results may be obtained with other data or network structures. Therefore, to determine the superiority or inferiority of the prediction performance of the two models (SDA, DBP), it is essential to conduct follow-up research through numerous additional experiments (various network structures, number of neurons, initial values of weight vectors, etc.).

5. CONCLUSION

The purpose of this study is to establish the reliability of a traffic video analysis system using an optimal artificial neural network model. In this study, experiments were conducted using road traffic images collected in the field, and the results of this study can be used not only for road traffic image analysis but also for vacancy detection in parking lots. Therefore, the findings of this study are essential for the implementation of intelligent transportation systems and can also contribute to enhancing the reliability of real-time parking information systems currently deployed in various cities.

In particular, this study compared the performance of three core artificial neural network models widely used in various fields: SDA, DBN, and Backpropagation. Summarizing the analysis results obtained through various experiments in this study, the following points are noteworthy. Firstly, within the constructed network in this study, the SDA model exhibited the highest predictive power. However, in some alternative network configurations, DBN showed relatively better predictive performance, indicating that the predictive abilities of each model may vary depending on the network structure. Secondly, both representative deep learning algorithms, SDA and DBN, demonstrated significantly better performance than the traditional Backpropagation model. Thirdly, it was observed that all three models (SDA, DBN, and Backpropagation) are highly influenced by the initial values of the weight vectors. Therefore, to properly evaluate the models while excluding the influence of these initial values, it is necessary to conduct multiple experiments with different initial values under the same conditions to draw meaningful conclusions.

With the increasing interest in artificial neural networks these days, many researchers are applying neural network models to various fields. However, most studies draw conclusions based on a single experiment. As demonstrated in this study, since all models are significantly influenced by initial values, it is crucial to derive conclusions based on statistical analysis through multiple experiments to ensure the reliability of the results. In this research, to minimize the impact of changes in the initial weight vector value, the same conditions (network structure, hyperparameters, etc.) were run 30 times, and the results were statistically validated. However, more experiments are needed to obtain more accurate results.

In this study, statistical methods were introduced to increase the reliability of the results. However, the 30 trials of experiments conducted in this study are only the minimum standard for statistical verification, and additional experiments are needed to ensure greater reliability. Therefore, limiting the number of experiments to 30 under the same conditions is considered a limitation and weakness of this study.

In future follow-up studies, the objective is to enhance the reliability of the results obtained in this study and conduct additional experiments with a more diverse range of network structures (varying hidden layers and hidden neuron numbers) for a more accurate comparison of the three models. Moreover, there is a plan to increase the number of experiments from the current 30 to over 100, aiming to further enhance the reliability of the results.

REFERENCES

[1] D. Kim, “Acquiring Real-Time Traffic Information Using Deep Learning Neural Networks,” Asia-Pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology, vol. 5, no. 1, pp. 435-444, 2016. DOI: 10.35873/ajmahs.2016.6.5.042

[2] H. Cui, G. Y., N. Liu, M. Xu, and H. Song, "Convolutional neural network for recognizing highway traffic congestion," Journal of Intelligent Transportation Systems, vol. 24, no. 3, pp. 279-289, 2020. DOI: 10.1080/15472450.2020.1742121

[3] H. Nguyen, "Deep Neural Network-based Detection of Road," International Journal of Advanced Computer Science and Applications, vol. 14, no. 9, pp.307-314, 2023.

[4] D. Kim, "Standard and Advanced Backpropagation Models for Image Processing Application in Traffic Engineering," Journal of Intelligent Transportation Systems, vol. 7, no.3-4, pp.199-211, 2002. DOI: 10.1080/714040816

[5] D. Kim, "Pre-processing of inputs to a neural network model for better performance in traffic scene analysis," Civil Engineering and Environmental Systems, vol. 27, no.1, pp. 23-31, 2010. DOI: 10.1080/10286600802252719

[6] Y. Gao et al., "A novel image-based convolutional neural network approach for traffic congestion estimation," Expert Systems with Applications, vol. 180, 2021, 115037. DOI: 10.1016/j.eswa.2021.115037

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," In D.E. Rumelhart, J.L. McClelland and the PDP Research Group, eds. Parallel distributed processing, Cambridge, MA: MIT Press, 1986.

[8] G. E. Hinton, R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504-507, 2006.

[9] P. Vincent et al., "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," Journal of Machine Learning Research, vol. 11, no. 110, pp. 3371-3408, 2010.

[10] L. Wang, Z. Zhang, and J. Chen, “Short-Term Electricity Price Forecasting with Stacked Denoising Autoencoders,” IEEE Transactions on Power Systems, vol. 32, no. 4, pp. 2673-2681, 2016. DOI: 10.1109/TPWRS.2016.2628873

[11] O.M. Saad et al., “Automatic Arrival Time Detection for Earthquakes Based on Stacked Denoising Autoencoder,” IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 11, pp. 1687-1691, 2018. DOI: 10.1109/LGRS.2018.2861218
[12] W. Lin et al., “A Deep Neural Collaborative Filtering based Service Recommendation Method with Multi-Source Data for Smart Cloud-Edge Collaboration Applications,” Tsinghua Science and Technology, 2023. DOI: 10.26599/TST.2023.9010050
[13] P. Singh, A. Sharma, S. Maiya, “Automated atrial fibrillation classification based on denoising stacked autoencoder and optimized deep network,” Expert Systems with Applications, vol. 233, 120975, 2023. DOI: 10.1016/j.eswa.2023.120975
[14] Y. Bengio et al., "Greedy layer-wise training of deep networks," International conference on neural information processing systems, MIT Press, pp. 153-160, 2006.
[15] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 2016. DOI: 10.1109/ICDMW.2016.0041
[16] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," MSC thesis, Technical University of Denmark, 2012.
[17] G. E. Hinton, S. Osindero, Y. W. The, "A fast learning algorithm for deep belief nets," Neural Comput, vol. 18, no. 7, pp. 1527-1554, 2006.
[18] Mohamed, G. Dahl, G. Hinton, "Acoustic modeling using deep belief networks," IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 14-22, 2012.
[19] D. Kim, "Normalization of Input Vectors in Deep Belief Networks (DBNs) for Automatic Incident Detection," Asia-Pacific Journal of Convergent Research Interchange, vol. 4, no. 4, pp. 61-70, 2018. DOI: 10.14257/apjcri.2018.12.07
[20] D. Kim, "Improving prediction performance of neural networks in pattern classification," Int. J. Comput. Math., Vol. 82, no. 4, pp. 391-399, 2005. DOI: 10.1080/0020716042000301806
[21] D. Kim, "Normalization methods for input and output vectors in backpropagation neural networks," Int. J. Comput. Math., vol. 71, no. 2, pp. 161-171, 1999. DOI: 10.1080/00207169908804800
[22] D. Kim, "Prediction performance of support vector machines on input vector normalization methods," Int. J. Comput. Math., vol. 81, no. 5, pp. 547-554, 2004. DOI: 10.1080/00207160410001684325
[23] D. Kim, "Nonlinear Normalization Model to Improve the Performance of Neural Networks," Asia-Pacific Journal of Convergent Research Interchange, vol. 6, no. 11, pp. 183-192, 2020. DOI: 10.47116/apjcri.2020.11.16
[24] R. Bevans, "One-way ANOVA | When and How to Use It (With Examples) (scribbr.com)," June 22, 2023. https://www.scribbr.com/statistics/one-way-anova/