sustainability

Article

Deep Neural Network Design for Modeling Individual-Level Travel Mode Choice Behavior

Daisik Nam [1,*] **and Jaewoo Cho** [2]

[1] Department of Civil and Environmental Engineering, Institute of Transportation Studies, University of California, Irvine, CA 92603, USA
[2] College of Social Science, Hansung University, Seoul 02876, Korea; jaewoocho@hansung.ac.kr
* Correspondence: daisikn@uci.edu

Received: 11 August 2020; Accepted: 9 September 2020; Published: 11 September 2020

Abstract: Individual-level modeling is an essential requirement for effective deployment of smart urban mobility applications. Mode choice behavior is also a core feature in transportation planning models, which are used for analyzing future policies and sustainable plans such as greenhouse gas emissions reduction plans. Specifically, an agent-based model requires an individual level choice behavior, mode choice being one such example. However, traditional utility-based discrete choice models, such as logit models, are limited to aggregated behavior analysis. This paper develops a model employing a deep neural network structure that is applicable to the travel mode choice problem. This paper uses deep learning algorithms to highlight an individual-level mode choice behavior model, which leads us to take into account the inherent characteristics of choice models that all individuals have different choice options, an aspect not considered in the neural network models of the past that have led to poorer performance. Comparative analysis with existing behavior models indicates that the proposed model outperforms traditional discrete choice models in terms of prediction accuracy for both individual and aggregated behavior.

Keywords: discrete choice model; deep neural network; mode choice behavior; smart urban mobility; individual-level choice prediction; agent-based model; random utility model; logit model

1. Introduction

The capabilities of artificial intelligence (AI) are recognized in various fields. This paper aims to implement the concept of deep learning (DL) algorithms, one branch of the AI family, for a prediction model of travel mode choice. Random utility models (RUMs) and discrete choice models derived from [1–7] are traditionally used to predict travelers' choice, which is an essential component of transportation planning models. However, RUMs typically have strong underlying assumptions and limitations to their accuracy. RUMs assume that individuals select the alternative which has maximum utility and that an individual's utility can be calculated using linear combinations of deterministic elements and unseen errors.

Deep learning is a promising approach in many academic fields. It is commonly applied to computer vision, pedestrian detection, language modeling, picture classification, and speech recognition [8–11]. However, to the best of the authors' knowledge, the use of deep learning-based mode choice models in transportation planning remains in its infancy.

AI approaches have been used to analyze travel behaviors or predict traffic conditions since the late 1990s, but the performance of such approaches was often disputed. Some researchers argued that AI models do not guarantee improvements compared to traditional models [12–15]. Multilayer perceptron models (MLP) were the most commonly used AI schemes for mode choice models. From several experiments, Carvalho et al. [12] stated that it was not clear if MLP offered any advantages (computational or otherwise) over a logit model in terms of capturing travelers' choice behavior. Hensher and Ton [13] compared the performance of nested logit models (NLM) with MLP in modeling commuter mode choice. They found that although artificial neural network (ANN) forecasts individuals' choices better than NLM, NLM predicts market share ratios slightly better. Cantarella and Luca [15] examined a multilayer feedforward neural network model by comparing it with random utility models (RUMs), but they could not determine which model was better, as two case studies showed different results. There are also some recent studies [16–21] that show AI models to outperform RUMs such as NLM and cross-nested logit model (CNLM) in predicting travel mode choice. In summary, the results from past research based on AI approaches are somewhat mixed in comparing the performance of traditional discrete choice models and neural network models.

This disappointing performance is due to the inherent shortcomings of AI approaches. One of the major reasons is an overfitting problem that makes the estimated model fit too closely to the sample training data while performing poorly on a real dataset. In addition, lack of computing power often results in too many costs in terms of physical time and energy. However, these problems have been largely addressed recently, especially since Hinton proposed novel techniques for deep learning structures [22]. Followed by his pioneering study, some other improvements have been applied in order to avoid overfitting and to allow more complexity, depth, and accuracy to neural network models, such as Rectified Linear Unit (ReLU) [23] and dropout [24]. During this period, there has been a dramatic increase in computational power based on graphics processing units (GPU).

Despite these developments, scholars in the transportation discipline have paid less attention to apply these improved methodologies to their research. To fill this gap, this study proposes an application of deep neural networks (DNN), a member of the family of deep learning algorithms, to predict travelers' mode choice behavior. The next section briefly describes previous mode choice models such as random utility models and MLPs, and introduces DNN. Section 3 describes how we construct our mode choice models. Section 4 explains the experimental data and evaluation methods used in the study. The performance of each model is evaluated in the Section 5. Finally, we close our paper with conclusions and remarks on future research.

2. Choice Model Description

2.1. Random Utility Model

A mode choice model estimates the probability that a traveler will select a certain travel mode for his/her travel. The most popular choice model is the random utility model (RUM), which is an umbrella term for the family of logit, nested logit, and cross-nested logit models. The main assumption of RUM is that the probability of selecting an alternative is based on a utility of a certain travel alternative. A utility consists of observable variables (X) and an unobserved component (ε), as shown in Equation (1). McFadden [25] describes the estimation method of a vector of parameters β' by the maximum likelihood method with the assumption that the error term of each alternative is independent (independence of irrelevant alternatives, IIA).

$$U_m^n = \beta' X + \varepsilon \tag{1}$$

Travel mode choice problems have an inherent limitation in that some of the alternatives are not independent. A well-known extreme case is a traveler who commutes either by car or by a bus. Even adding a color attribute (blue and red) in the bus alternatives brings unreasonable results. A detailed explanation is addressed in [26]. One of the common approaches for this limitation is nested logit. A nested logit model structure has a choice hierarchy. An upper level estimates the choice probability of 'Auto' and 'Bus,' then applies the conditional probability of the color of buses. The cross-nested logit model (CNL) is also used when alternatives have mixed interactions that a nested logit model would not capture. Small [2] initially proposed CNL, which many researchers have theoretically analyzed thereafter [27].

The availability of alternatives in the choice set for each traveler is also an important feature in a mode choice model. For example, a non-vehicle owning person should be considered unavailable to a car option [28]. Cascetta and Papolar [29] propose a method to consider the availability of options to the utility function of traditional discrete choice models, which are now commonly applied in logit, nested logit, and cross-nested logit models [27,30–32].

2.2. Artificial Neural Network and Multilayer Perceptron Models

Artificial neural network (ANN) is a learning algorithm that imitates the human neural system. An ANN consists of multiple nodes, called neurons, that communicate through synapses. Typically, there are three sets of nodes: Input nodes, intermediate (hidden) nodes, and output nodes, and each node category plays a different role. Input nodes receive input information, output nodes yield output signals, and intermediate nodes receive signals from input nodes, and manipulate those signals to give results to output nodes. Input nodes in a mode choice model are connected to the independent variables such as travel time and cost—the output nodes associate with the probability of alternatives, implying that the number of output nodes should be the same as the number of alternatives. An ANN model can have multiple intermediate layers that contain sets of intermediate nodes, and if there exist more than two layers, we call it a multilayer perception (MLP) model. If it only has one intermediate layer with the same number of nodes with the independent variable, the model could be designed to have a logit model structure with a linear utility function. By increasing the number of the intermediate layers, a model can consider the non-linearity of input and output relationship, which will be further described in Section 3.1. A logit model also can incorporate the non-linearity characteristics; however, a model should manually define the nonlinear function—such as log, exponential—for an independent variable, which MLP automatically characterizes.

MLPs typically use backpropagation, starting with randomly weighted synapses, and trains them with input and output values. The simplest type of MLP is a feedforward network in which the signals move in only one direction, from the input nodes, via hidden layers, to the output nodes. MLPs have some advantages compared to simple perception models, with respect to their greater learning and prediction power. In addition, MLP employs transfer functions that modify input signals and pass them to nodes in the next intermediate layer, using weights and biases. Equation (2) presents a specific transfer function. It should be noted that there are various types of functions such as sigmoid, tanh, and ReLU, to estimate parameters. Figure 1 shows the input and output relationship of each function. A sigmoid function is a generalized form of a logistic curve, which outputs the conditional probability of each alternative as logit model, so past ANN models for the travel mode choice prediction have employed a sigmoid function [13,15].

ReLU is a rectified linear unit that Nair and Hinton proposed in 2010 [23], which has become a popular activation function in recent research. One advantage of this non-saturated function is that it speeds up the convergence of optimization. The other advantage is in tackling the vanishing gradient problem [33]. The detailed advantages will be addressed in the next session.

$$Y = f\left(\sum_{i=1}^{n}(W_i Z_i + \beta)\right) \tag{2}$$

where

n = number of input signals to a node

W = weights

Z = inputs

β = bias term

Another characteristic of an MLP model is that there are training processes to estimate parameters that minimize the cost function. There are several ways to train multilayer perceptron models. MLP finds each layer's parameter using backpropagation methods and optimization techniques.

MLP is a nonlinear problem, and previous research efforts have focused on heuristics such as genetic algorithms or a gradient descent method [14,18]. But these heuristics could be trapped in poor local optima when the learning process is initiated from a wrong starting point [9,15]

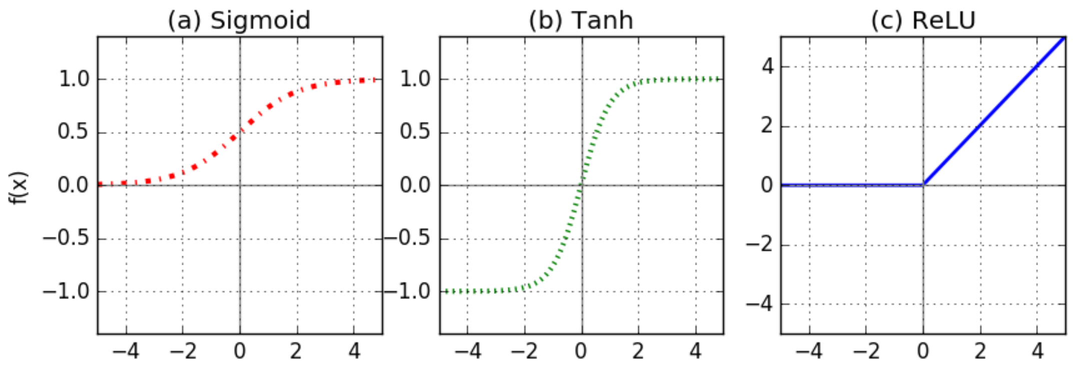

Figure 1. Three common transfer/activation functions.

3. Deep Neural Network (DNN) with a Function for Availability of Alternatives

3.1. Deep Neural Network Structure Design

Deep neural network (DNN) is a class of ANN and its structure is essentially similar to MLP. The difference with MLP is that a DNN has a significantly greater number of hidden (intermediate) layers than an MLP. With MLP, it has been reported that increasing the number of hidden layers leads to several problems. Firstly, more hidden layers exponentially increases the required computing resources. Secondly, although more hidden layers and nodes contribute to model accuracy and prediction power, the local optimum or overfitting problems arise as tradeoffs. Lastly, processing time varies highly depending on parameter initializations. Therefore, many researchers in the past have concluded that adding more hidden layers brings no benefits or even adverse effects in model estimations.

To cope with these problems, in 2006, Hinton [22] suggested implementing unsupervised learning to initialize parameters and then executing supervised learning in order to estimate parameters efficiently. Subsequently, Glorot and Bengio [34] devised a simplified version of the initialization process known as Xavier initialization (after Glorot's first name), which improves the initialization process. The dramatic evolution of parallel computing by utilizing graphical processing units (GPU) further spurred the adoption of deep learning models.

The number of hidden layers and perceptrons in the neural network characterize the complex relationship between input variables and outputs. To illustrate this, we synthesize three small samples of data sets, assuming travel time (*x1*) and cost (*x2*) as the explanatory variables, as in Figure 2—the two colors, red (Auto) and blue (Train), showing the binary mode choices associated with each data point. It is noteworthy that the synthesized data is only designed for examples by using the Scikit-learn dataset for classification (moons, circles, and linearly separable) [35]. For better understanding the role of hidden layers, we use a simple neural network with two input variables (x1, x2) and two perceptrons for this data. Figure 2 then graphically indicates with the same color regions where the neural network would predict the choices in a classification problem, for different numbers of hidden layers. A region colored blue represents highly likely to be transit. The color transitions to red as the probability of transit decreases. The numbers in the subgraphs show the fraction of accurate predictions for a test set.

One hidden layer with two perceptrons linearly divides the space (Figure 2(1-a,2-a,3-a). If we assume that the travel mode decision is based on complex combinations between time and cost, one hidden layer might not explain the travelers' mode choice behavior well. Increasing the number of hidden layers improves the overall predictive potential since the next layer divides the space of the previous step, which could unearth more complex decision-making patterns. This process is illustrated in Figure 2. As we increase the number of hidden layers, the decision areas are formed in similar patterns as in the original data sets. However, too many hidden layers could reduce the prediction potential because of overfitting issue.

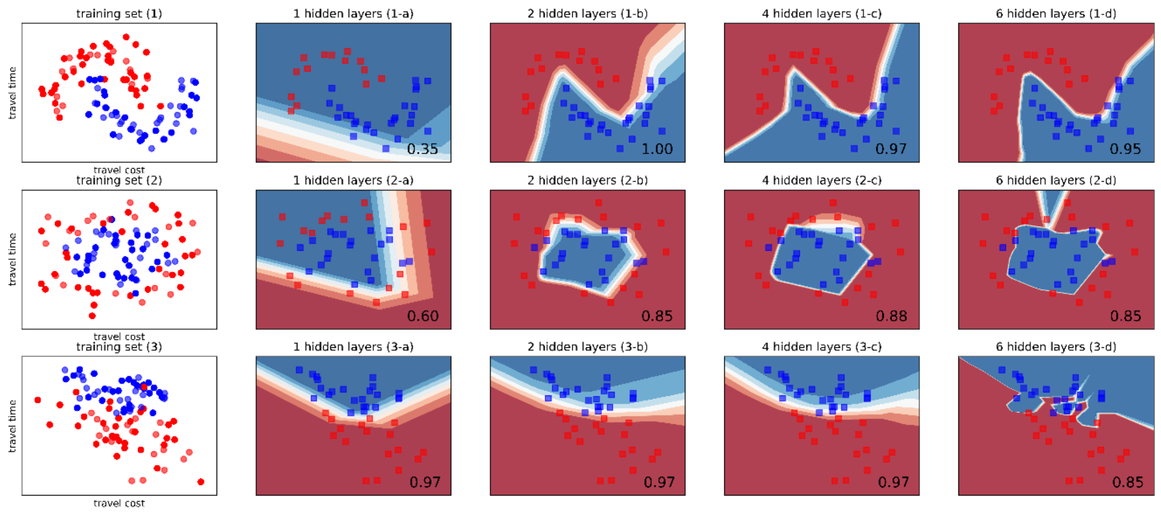

Figure 2. The predictive potential according to the number of hidden layers.

The first issue to address is overfitting. While DNN is a powerful machine learning model, just like in any statistical method the danger of overfitting always exists. This happens when the model captures noises of the dataset, which could exist in the training set but not in real data, although they have the same distribution. The obvious evidence of overfitting is when the model has a high prediction power only with the training set and fails to replicate its performance on non-training data. Figure 2(1-d,2-d,3-d) are examples of overfitting. The number in the figure indicates the accuracy of a test set of a trained model. Too many hidden layers provide worse predictions for test sets.

One of the most popular techniques to prevent overfitting is regularization. It provides modifications or weight penalties to a learning algorithm in order to reduce its generalization error while leaving the training error untouched. However, this method incurs prohibitively expensive computing costs with a large neural network. Alternatively, the "dropout" method resolves both the overfitting and computational efficiency issues. The term "dropout" means that it drops random nodes of each layer, and generates a thinned network per each training process, as marked "X" on perceptrons in Figure 3. With dropout, we can easily handle overfitting even in large networks. Furthermore, the ReLU function reduces overfitting because it regularizes an activated value to zero if a value from a perceptron is less than a certain value (zero). In other words, the ReLU function inherently generates a zero value when an input value is less than zero.

However, from our initial experience, we found that configuring a network with full ReLU functions could not provide each alternative's choice probability. This is a serious drawback since calculating the market share ratio from the model is an important purpose of a choice model. It does not mean that applying the ReLU function is not suitable for the travel choice model because ReLU function is experimentally known to increase prediction accuracy [36,37], and is efficient in the optimization process [9]. Thus, the last hidden layer of the proposed neural networks is connected to the sigmoid activation function for our travel choice model. This enables us to attain each alternative's choice possibility, which is a similar output as from RUMs.

The perceptrons of the last layer are connected to the softmax function to convert the signal of a perceptron into a probability of each category j in a choice set of individual n (C^n). The dimension of the perceptrons is the same as the number of total choice mode set. The softmax (Equation (3)) is a generalization of the method of forecasting the probability of a logit model. Then k_j can be interpreted as the nonlinear utility function for the stochastic choice function, the nonlinearity is characterized by the structure of the deep neural network.

$$\Pr\left(j \middle| C^n\right) = softmax\left(j \in C^n,\ K\right) = \frac{\exp\left(k_j\right)}{\sum_{m \in C^n} \exp(k_m)} \quad (3)$$

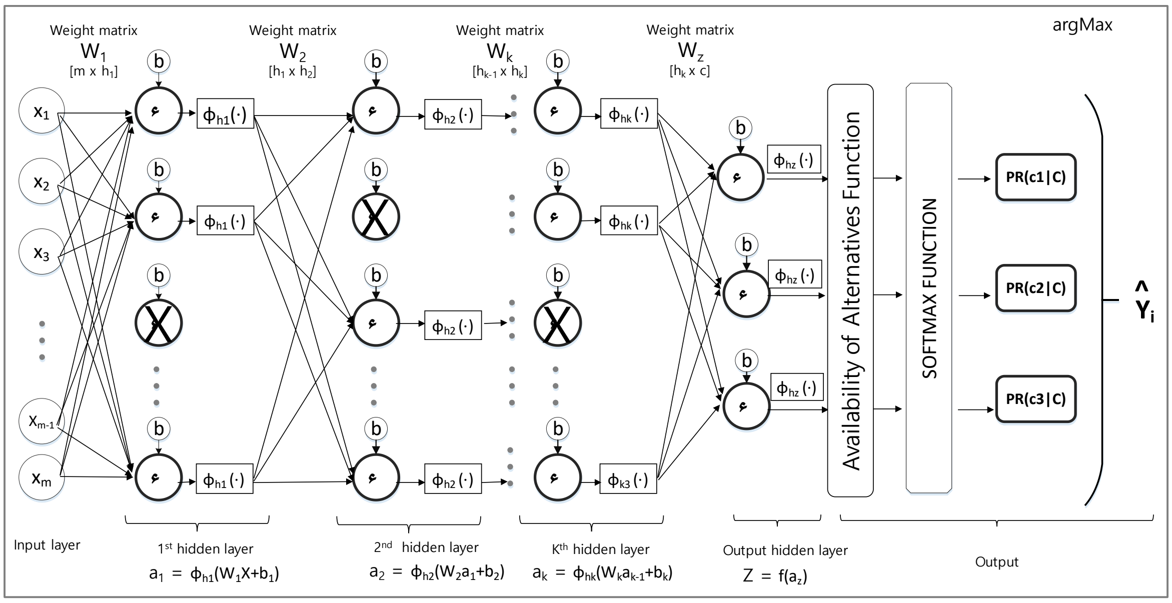

Figure 3. The overall structure of the proposed deep neural network model.

The cost function we set for these models is a cross-entropy function (Equation (4)), which is derived from the canonical form of log-likelihood function for random utility models [38].

$$C = -\frac{1}{N} \sum_{n}^{N} \sum_{j \in C^n} \left[y_j^n \ln \Pr\left(j \middle| C^n\right) + \left(1 - y_j^n\right) \ln\left(1 - \Pr\left(j \middle| C^n\right)\right) \right] \quad (4)$$

We designed a deep neural network (DNN) for a model for travel mode choice behavior, as shown in Figure 3. From our experiments, we found that a deep neural network with four layers functions with a combination of ReLU and a sigmoid function that performs well for the given dataset. We call it DNN(RRRS), DNN with activation functions (ReLU, ReLU, ReLU and Sigmoid). The first three layers include 80 perceptrons with ReLU activate functions. The last layer consists of three perceptrons, which are the same number of choice sets, with the sigmoid function. An input layer is connected to the first hidden layer. The normalization of input data helps the training process in the fast convergence of a gradient-based optimization [39]. The proposed model applies Min-Max normalization that transforms the minimum value to 0 and the maximum value to 1.

In addition, deep neural networks require fine-tuning of hyperparameters: Initialization, the number of epochs for training, and a learning rate. As part of the deep learning used in our work, we initialize the neural networks with Xavier's initialization technique, as in Glorot and Bengio (26).

An epoch, as is well-known, is a single pass through the whole training set. As shown in Figure 4, the log-likelihood of the test set becomes maximum at a certain epoch number, whereas that of the training set keeps increasing as the iterations keep processing, which is over-fitting.

Another feature of deep learning is an optimizer for the training process. We applied Adam (adaptive moment estimation) optimizer [40], which is an algorithm for gradient-based optimization with the stochastic objective function. Adam requires hyperparameters such as alpha (learning rate), beta 1, beta 2, and epsilon. A larger value of alpha facilitates faster initial training.

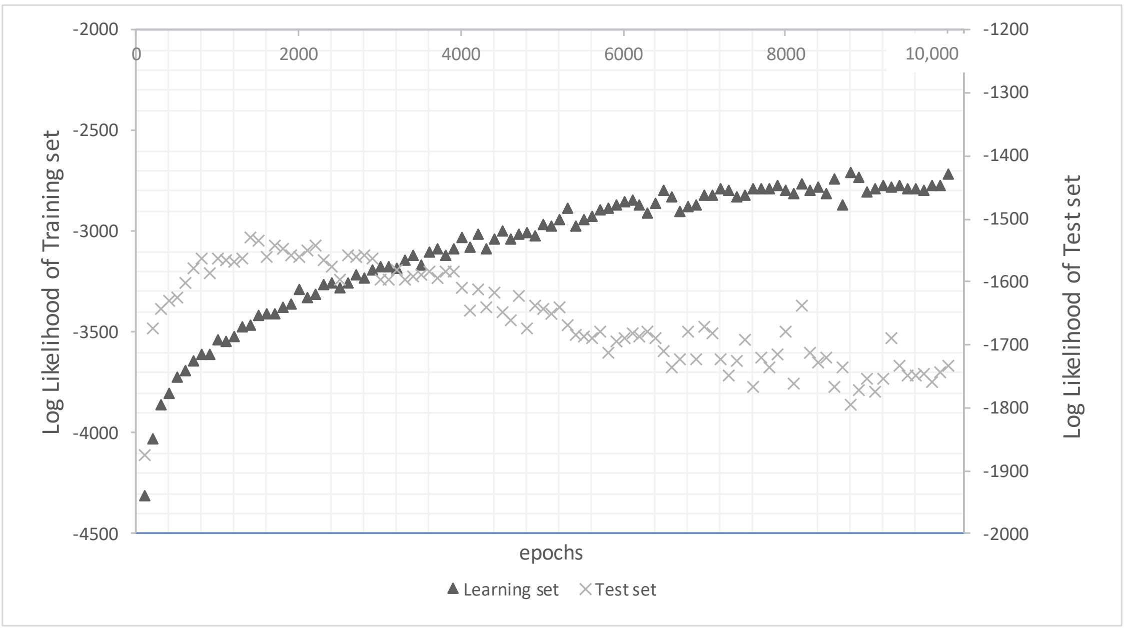

Figure 4. Log-likelihood of training and test set with respect to the number of epoch.

A certain ratio of neurons in each network is randomly dropped out to prevent overfitting. This ratio and the training set ratio are important factors in the model's performance. To determine the appropriate dropout ratio and training set ratio, a sensitivity analysis of the network's performance is conducted. Figure 5a shows an example of the sensitivity analysis for our data set that will be described in Section 4. The percentage of correct prediction is calculated by applying the trained model to the test data set. In each combination, the average value of the performance index is calculated from 10 replica sets. When the dropout ratio is below 0.3, there are critical gaps between the trained model performance and test set performance. When the dropout ratio is too high, the accuracy is too low. At a dropout ratio of 0.6, the model is very stable, with a training set ratio of 0.5 (Figure 5b), which means that the small size of the learning data set could also explain overall observations without an over-fitting issue.

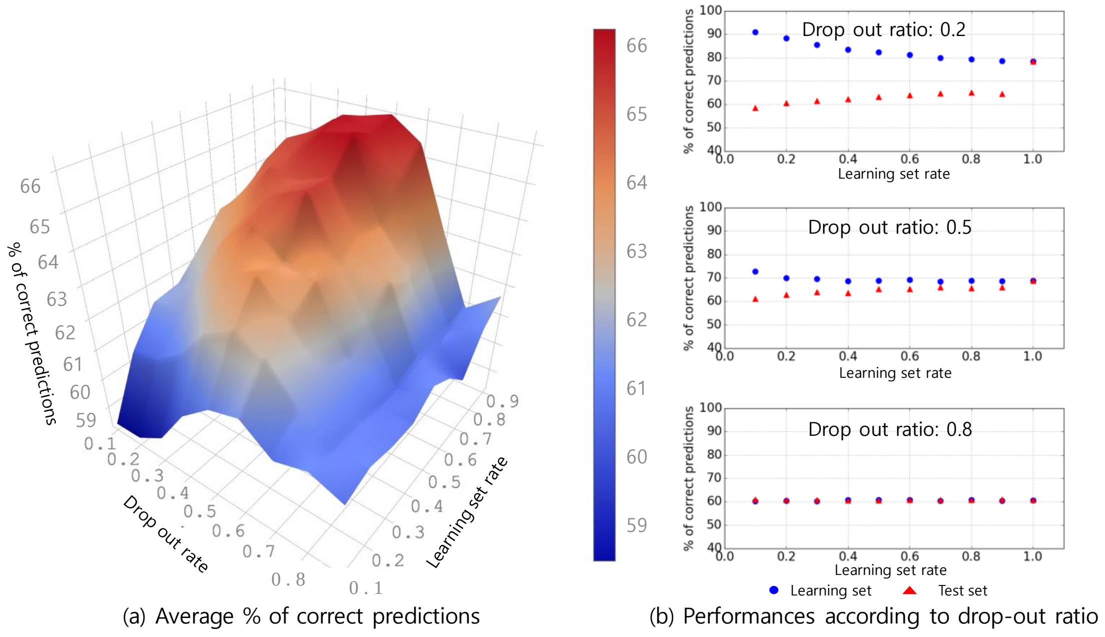

Figure 5. Sensitivity analysis with drop out ration and training set rate.

3.2. Deep Neural Network with Available Alternative Function

The next item to discuss in the DNN structure in Figure 3 refers to the alternatives in the choice set of individuals. We constructed our proposed model (DNN-A) with AAF (availability of alternatives function) as in Equation (6). Our DNN-A structure refers to the alternatives in the choice set of individuals. In a choice prediction model, some alternatives that are unavailable to certain individuals should be addressed. Without this, an unavailable alternative can be predicted as their choice. For instance, for individuals who do not own a vehicle, that travel mode should be eliminated from the choice set. Cascetta and Papolar [29] introduced a utility function for implicit availability/perception of choice alternatives for traditional discrete choice models. In conventional discrete choice models, such as logit, nested logit, and cross-nested logit models [27], unavailable alternatives are controlled by increasing the alternative's disutility. For a neural network, we need to develop a method to handle the issue.

We propose the availability of alternatives function (AAF) for the DNN(RRRS), shown as a vertical box in Figure 3. From the feedforward process in Equation (5) below, the dimension of the perceptron in the last hidden layer becomes the same as the number of alternatives. Previous DNN models have not considered unavailable alternatives in their methodology. These alternatives could also have a non-zero value in those models, since the softmax function in the last step (Equation (4)) calculates each alternative's choice probability based on the last layer's output. It is, therefore, important to force the probability of unavailable alternatives to be zero. Otherwise, the model might predict unreasonable results for certain individuals. We refer to Cascetta and Papolar's [29] utility function, as shown in Equation (6). The logarithm transformation represents the availability of alternatives. If an alternative is available, $u_c(j)$ is 1, then the logarithm of it becomes zero. If not available, $u_c(j)$ is zero, and the perceptron value (z_j) for the alternative becomes minus infinity. A generalization is the AAF control function, as formulated in Equation (7). AAF controls the value of the output hidden layer (z_j). If the alternative j is unavailable for individual n, the z_j is altered to negative infinity, so the alternative does not affect the probability of all other choices, outcomes of AAF are directed by the softmax function, Equation (8)

$$\mathrm{a}_z^i = \phi_z\left(a_k^i\right) = \phi_z\left(W_z\left(\phi_{hk}\left(\mathrm{W}_k a_{k-1}^i + b_k\right) + b_z\right)\right) \tag{5}$$

$$\mathrm{Z} = \mathrm{AAF}(\mathrm{a_o}) = \mathrm{a}_z^i + \ln u_c(j) \tag{6}$$

$$\mathrm{Z} = \mathrm{AAF}(\mathrm{a_o}) = \left\{ \begin{array}{llll} z_j = & a_z & if\ j \in C^n & for\ \forall\ j,\ n \\ z_j = & -\infty & if\ j \notin C^n & for\ \forall\ j,\ n \end{array} \right\} \tag{7}$$

$$\Pr\left(j \middle| C^n\right) = softmax(Z) = \frac{\exp\left(z_j\right)}{\sum_{k \in C^n} \exp(z_k)} \tag{8}$$

where

a_z^i : *output value of perceptron i of the last hidden layer z*
a_k^i : *output value of perceptron i* of a value of a prior hidden layer k
ϕ_z : *activation function* of the last hidden layer z
ϕ_k : *activation function* of a prior hidden layer k
W_z : *weight matrix of hidden layer z*
W_k : *weight matrix of hidden layer k*
b_k : *bias of hidden layer k*
z_j : *the output of the last hidden layer for mode j* after AAF function
C^n : *choice set of individual n*

4. Experimental Setting

To explore the performance of the deep neural network model, we utilized data from a mode choice survey of long-distance travel. Abay [41] conducted the revealed preference (RP) and stated preference (SP) surveys to estimate the hypothetical demand for Swiss Metro, a new innovative intercity passenger transport in Switzerland. Several studies utilized this data for evaluating their proposed model [6,20,27,41–44]. The SP survey data is available on the Biogeme website with discrete choice estimation packages [27]. There are three alternative travel modes in the choice set: Car (only for car owner), rail, and Swiss Metro. Car and rail are existing modes, and Swiss Metro (SM) is a hypothetical travel mode. The dataset also contains various attributes such as travel time, travel cost, age, luggage, currently available mode, annual season pass, number of seats, and frequency [6]. Note that the travel option for car is only available for a traveler who owns a personal vehicle. Some 17.15% of individuals in the dataset cannot travel by car.

NL, CNL, MLPs, and DNN are compared to DNN-A, the proposed model. A large number of input variables to a neural network could make it too complex for a network to train. Therefore, instead of including all possible explanatory variables in MLPs and DNN, we identify significant variables from traditional discrete choice models, such as a nested logit model, to increase computational efficiency. DNN has a considerably complex network structure with many layers. All perceptrons in the first hidden layer are directionally linked from input variables, which affects the results. The concept of selecting input variables from a traditional model is very significant, in that we can avoid unnecessary computational overhead.

Table 1 shows the variables and basic statistics. The total number of individual observations is 6768. Mode share of SM is dominant at 60.43%. Mode shares of car and rail are 26.15% and 13.42%, respectively. In the nested logit model (NL) and the cross-nested logit model (CNL), our research refers to Bierlaire's utility functions and nested structure [6,7] as the base models for comparison with the proposed model. With the two nests, a choice is determined by the combination of the existing modes and the mode's attributes. The utility function of a nested logit model has shared unobserved attributes (errors). Equation (9) indicates the utility function for the NL and CNL.

$$U_{em}^{n} = V_{e}^{n} + V_{m}^{n} + V_{em}^{n} + \varepsilon_{e}^{n} + \varepsilon_{em}^{n} \tag{9}$$

where

U_{em}^{n} : utility of travel mode e, m for an individual n
e : existing travel mode (Car, Rail)
m : future travel mode (Swiss Metro)
$\mathrm{V}_{\mathrm{e}}^{\mathrm{n}}$: deterministic component of the utility of existing
$\mathrm{V}_{m}^{\mathrm{n}}$: deterministic component of the utility of Swiss Metro
$\mathrm{V}_{\mathrm{em}}^{\mathrm{n}}$: deterministic component of the common utility among existing and Swiss Metro
$\varepsilon_{\mathrm{e}}^{\mathrm{n}}$: error term for existing or future – assumed $\mathrm{Gumbel}(0, \mathrm{u_e})$
ε_{em}^{n} : *error term for combination of two nests – assumed* $Gumbel(0, u_{em})$

When we assume that there is no distinctive preference difference between existing and non-existing modes, the utility function of each mode becomes as in Equation (10). In here, $\varepsilon_{\mathrm{e}}^{\mathrm{n}}$ is assumed an independent and identical distribution (IID) Gumbel distribution with (0, $\mathrm{u_e}$). $\varepsilon_{\mathrm{em}}^{\mathrm{n}}$ is also assumed an IID Gumbel distribution with (0, $\mathrm{u_{em}}$). Mathematically, the error term on the upper nest ($\varepsilon_{\mathrm{e}}^{\mathrm{n}}$) does not involve the lower level.

$$U_{em}^{n} = V_{m}^{n} + \varepsilon_{e}^{n} + \varepsilon_{em}^{n} \tag{10}$$

In both nested structures, with the Bayes theorem, the probability of choosing the existing mode (e) and the travel mode (m) for individual n is calculated by multiplying the marginal probability and the conditional probability, as follows in Equation (11). The detailed information about NL and CNL for the data set is explicitly explained in [27].

$$Pr^{n}(e, m|C_{n}) = Pr^{n}(e\big|E)Pr^{n}(m\big|e) \tag{11}$$

We used Biogeme software for the model [27]. It is noteworthy that Biogeme also incorporates the availability of alternatives in the same way as our proposed model They found that both travel time and travel cost generally affect all travel modes' choice probability and that relevance of other variables depended on the modes. The variable 'Luggage' only influences the car choice. Frequency, annual season pass, and age are selected as additional explanatory variables for rail. The probability of selecting the Swiss Metro (SM) alternative is dependent on frequency, annual pass, and the number of seats.

Table 1. Variables used in experiments and basic statistics.

		Frequency(trips)			Ratio		
		Car	Rail	SM	Car	Rail	SM
General	Mode share	1770	908	4090	26.15%	13.42%	60.43%

Variable Name				Mean			Standard deviation		
		Explanation	Data type	Car	Rail	SM	Car	Rail	SM
Mode-specific	ASC	Constant	Continuous						
	TT	Travel time (minutes)	Continuous	123.2	166.1	84.5	91.7	69.8	47.1
	Cost	Travel cost (CHF)	Continuous	78.7	490.9	641.1	55.9	1062.6	1411.7
	Freq	Frequency (minutes)	Integer		30.0	20.0		0.0	8.2
	Seats	Seat configuration	Binary			0.1			0.3

	Variable Name	Explanation	Data type	Categorical variable	Frequency
General	GA	Annual pass	Binary	no	5868
				yes	900
	Age	Age in class	Category	less than and equal to 24	423
				from 24 to 39	1944
				from 39 to 54;	2763
				from 54 to 65;	1197
				over 65	432
				unknown	9
	Luggage	Pieces of luggage	Integer	none	2727
				one piece	3852
				several pieces	189

Our proposed method, DNN-A (RRRS) is compared with other neural network models. Table 2 contains the details of other models. We first examine a single hidden layer MLP with a sigmoid function (MLP-S). This structure has been generally used in previous MLP research. We refer to Hensher and Ton's travel choice model [13] for MLP-S. We used the same 30 perceptrons and 1000 epochs for MLP-S.

Secondly, we test multiple hidden layer structures to understand the importance of deep learning techniques and to devise the structure of DNN model. Note that a multiple hidden layer structure can also be called a deep network structure, but we do not call it 'deep learning' since it does not implement any deep learning techniques. For comparisons, we constructed this multilayer perceptron (MLP) neural network with four hidden layers functioning with sigmoid functions and call it MLP(SSSS). This MLP(SSSS) was first compared with an MLP(RRRS) network where the first three layers have ReLU functions and the last layer has a sigmoid function to explore the benefit of the ReLU function. This is to analyze the advantage of using the ReLU activation function. Each hidden layer has 110 neurons, and 500 epochs are used. The next model we compare is a deep neural network (DNN) that uses an RRRS structure, based on our preliminary MLP results that showed the RRRS hidden layer structure to outperform the SSSS structure. This DNN model is called DNN(RRRS).

Hyperparameters for the models were selected from our experiments, as described in Table 2. A 500-epoch training scheme was selected after we observed that too many epochs induce the over-fitting problem, as exemplified in Figure 4. At a certain epoch number, the log-likelihood of the test set becomes maximum, whereas that of the training set keeps increasing as the iterations keep processing, which is over-fitting. We set the dropout ratio of the proposed neural network to be 0.55. We set the learning rate at 0.001 for the Adam optimizer, which is the recommended value by [40]. We also follow their recommendations for other hyperparameters.

Table 2. The details about neural network models for the comparative analysis.

	Hidden Layer		**Optimizer**			**Training Strategy**			
Model	The number of layers	The number of perceptrons in a layer	Optimizer	Learning rate	Initializer	Drop out ratio	Epoch	Batch size	Training set ratio
MLP-S	1	30	SGD	0.100	uniform random	N/A	1000	100	0.7
MLP (SSSS)	4	110	SGD	0.100	uniform random	N/A	500	100	0.7
MLP (RRRS)	4	110	SGD	0.100	uniform random	N/A	500	100	0.7
DNN (RRRS)	4	110	Adam	0.001	Xavier	0.55	500	100	0.7
DNN-A (RRRS)	4	110	Adam	0.001	Xavier	0.55	500	100	0.7

For the evaluation of the models, randomly divided data sets are used to reduce the occurrence of a biased sample, which could cause spurious results. This is because the mechanism of DNN has various random terms. Although the results of DNN are reliable because the estimation process generally converges, each output of DNN could be slightly different across multiple trials. To reduce this problem, prior machine learning research has proposed the partitioning of their observations into two subsets: A training set and test set [12,15,17,18], or three subsets: Training set, validation set, and test set [16].

We apply the cross-validation technique by randomly dividing the observations into 30 training and test sets. The same common replica sets are used for each model's estimation and validation. The total number of the sample is equal to the sum of the training set and the test set. The training set is independent of the test set and they have no common observations. The training set is used to train the model by pairing the input with the selected alternatives, which can be regarded as "supervised learning." The degree to which the trained model explains the travelers' choice behavior is evaluated by applying the model to the test set. A small training set could induce overfitting since a small number of data is unlikely to represent the behavior of all participants. Inferring the behavior of entire participants from a high portion of the training set could have high explanatory power for the observation set. However, allocating a high learning rate is undesirable in that a small test set could also be biased. In this research we have also examined the effect of the learning rate to these stated problems by changing the rate from 0 to 1, and we selected a training set ratio of 0.7 on the dataset

Keras and TensorFlow [8] with Python 3.6 are used to analyze both MLPs and DNNs by fully utilizing a parallel GPU computing environment. The GPU has 2048 processors boosting the optimization speeds. The tests are conducted under Windows 10 with Intel I9-9900k 3.60 GHz quad-core CPU and 64 GB memory (Redmond, WA, USA). The average training time of 0.7 training set rate in this scenario is 22.60 s. When GPU is not employed, training the model takes 265.13 s, on average

5. Comparisons of DNN and Against Other Mode Choice Models

5.1. Performance Measurements for the Comparisons

In mode choice research, log-likelihood, rate of correct predictions, and a confusion matrix are commonly used to evaluate model performance. The model's overall performance is measured with the log-likelihood (LL) and the rate of correct prediction. Log-likelihood (Equation (12)) indicates how probabilistically well fitted the model is, to the data. The log-likelihood is the log-sum of the selected alternative's probability (likelihood, L), Equation (13). This index is usually applied in a logit model family.

$$\mathbf{L} = \prod_{n=1}^{N} \prod\nolimits_{\forall m \in C^n} \Pr\left(m \middle| C^n\right)^{y_m^n} \tag{12}$$

$$\mathbf{LL} = \mathbf{lnL} = \sum\nolimits_n^N \sum\nolimits_{\forall m \in C^n} y_m^n \mathbf{Pr}\left(m \middle| C^n\right) \tag{13}$$

In contrast, agent-based modeling requires an individual-level choice; thus, the rate of correct prediction (Equation (14)) can be a critical measurement for the evaluation of a choice model. This index is common in the machine learning area in that it measures how the model correctly predicts each individual's choice.

$$\%\ of\ corrected\ prediction = \frac{1}{N} \sum\nolimits_n^N \sum\nolimits_{j \in C_n} y_j^n \tag{14}$$

We can consider two types of choice predictions from given probabilities. The first approach is the argmax max function, which is popular in machine learning research, assuming that an individual selects an alternative having maximum probability among alternatives, as shown in Equation (15). For example, if there are three choices and probabilities are $Pr^n = (0.40,\ 0.45,\ 0.15)$, then the argmax function converts it to $y^n = (0,\ 1,\ 0)$, meaning that the second option ($Pr\left(j \middle| C^n\right) = 0.4$) is selected.

$$\left\{ \begin{array}{c} y_j^n = 1\ if\ j = argmax\left(Pr\left(j \middle| C^n\right)\right) \\ y_j^n = 0\ if\ j \neq argmax\left(Pr\left(j \middle| C^n\right)\right) \end{array} \right\} \tag{15}$$

However, the argmax function is likely to bias to an alternative that has a higher probability than others irrespective of magnitude. For instance, if all individuals have the same probability, as a tuple $Pr^n = (0.40,\ 0.45,\ 0.15)$, the argmax function predicts that every individual selects the sole alternative. Then, the mode share of the second mode becomes 100%, which cannot explain the real-world observations. Thus, actual agent-based models such as activity-based regional transportation models (i.e., The San Diego Association of Governments (SANDAG) [45] and the California Statewide Travel Demand Model (CSTDM) [46]) apply the Monte-Carlo simulation method (MCS), as shown in Equation (16) and Equation (17). τ_j is a randomly generated number for alternative j and used to choose an alternative between two consecutive alternatives with relative cumulated probabilities (Equation (16)). For example, the tuple is converted to a cumulative curve (0.40, 0.85, 1.00), and generates a random number between 0 and 1. If the number is drawn at 0.30, the first travel mode will be selected, $y^n = (1,\ 0,\ 0)$

$$\sum_{i=1}^{j-1} \Pr(i) < \tau_j \leq \sum_{i=1}^{j} \Pr(i) \tag{16}$$

$$\left\{ \begin{array}{c} y_j^n = 1\ if\ j = \tau_j) \\ y_j^n = 0\ if\ j \neq \tau_j) \end{array} \right\} \tag{17}$$

Additionally, estimating the mode share ratio is more interesting for policymakers and planners [47]. The market share ratio of N individuals, the aggregate proportion choosing Cj, can also be calculated in two ways: An average of the predictions for alternative j across the individuals for argmax (Equation (18)) and Monte Carlo simulation (Equation (19)).

$$Mode\ share\ (arg\ Max) = MS_j(arg\ Max) = \frac{1}{N} \sum_n^N y_j^n \tag{18}$$

$$Mode\ share\ (Monte\ Carlo\ Simulation) = MS_j(MC) = \frac{1}{N}\sum_{n}^{N} y_j^n \tag{19}$$

A confusion matrix, also called an error matrix, can provide a summary of prediction accuracy for each travel mode. A confusion matrix consists of n by n cells. Each cell represents the total number of individuals who selected (row)/are predicted (column) kth travel mode. A value in diagonal cells represents the number of correct predictions, and other cells are for false predictions. Additionally, we can also compare an estimated mode share with observations.

5.2. Results

For nested logit model (NL) and cross-nested logit model (CNL), multiple training sets are imported to the Biogeme software [27], and the estimated models are tested on both the test sets and the total observation set. In all trials in both models, the input variables, except for “number of seats” and “constant for car,” are statistically significant since the *p*-value of each variable is lower than 0.05. The two variables just mentioned are insignificant over half the training sets, which make us drop them from the input variable set. All estimated models have a pseudo ρ^2 less than 0.260 except for one case. Thus, we concluded that the models are indicative of a good model fit.

Table 3 is a summary of log-likelihood comparisons. A cell in the table includes a mean and standard deviation of 30 runs. Overall, the proposed DNN-A and DNN predict both aggregate behavior and individuals’ behavior well, as indicated in Table 3. In terms of log-likelihood (LL) of test set, DNN-A and DNN have the highest average value with −1329.74 and −1327.29, respectively. The difference of performance is statistically insignificant, note the standard deviation of two models. When we consider that even cross-nested logit CNL improves the LL only to −1567.56 from the −1571.95 shown by NL, the result of both DNN and DNN-A are outstanding. Note that MLP’s performance is far worse than the random utility models (RUMs). This finding is a consistent result with previous research, as mentioned above. These results strongly demonstrate that designing deep structures without deep learning techniques is undesirable.

Table 3. Comparative results of predictive potential between the proposed model and other models.

	Log-likelihood	
Model	**Learning**	**Test**
Nested Logit	−3659.88 ± 25.85	−1571.95 ± 26.64
CNL	−3643.49 ± 29.51	−1567.56 ± 23.68
MLP(S)	−3719.89 ± 32.97	−1607.86 ± 24.40
MLP(SSSS)	−3929.01 ± 43.54	−1691.41 ± 25.45
MLP(RRRS)	−2601.23 ± 561.76	−1610.94 ± 234.11
DNN(RRRS)	−2817.07 ± 29.22	−1327.29 ± 28.76
DNN-A(RRRS)	−2831.86 ± 34.99	−1329.74 ± 30.97

Table 4 shows the individual-level prediction accuracy of both the training sets and the test sets. It is not likely for all models to have overfitting, except for MLP(RRRS). This model has large accuracy gaps between the training set (75.36%) and test set (67.76%) for argmax assumption 68.65%, and 62.31% for the Monte Carlo simulation assumption. This method only applies the ReLU function and does not apply other deep learning techniques such as dropout, initialization, and optimizer. DNN-A and DNN’s correct prediction rates of argmax are 71.84% and 72.08%, respectively, which are higher than the rates obtained from other models, meaning that the DNN-A forecasts individuals’ choice more accurately as well. The accuracy rates of DNN-A and DNN are higher than the value of 70.3% obtained from TasteNet [21] that also utilizes the Swiss Metro dataset. Again, neural networks without DNN techniques could not improve the accuracy of predictions. As shown in the results of MLP(SSSS), even higher complexity with the increased number of multiple layers makes a model worse. This implies that deep learning techniques enhance neural network models significantly. The table also summarizes the results of the function for individual choice assumptions (argmax and Monte Carlo simulation). The correct prediction percentage of Monte Carlo Simulation also rises to 62.8% (DNN-A) from 52.1% (CNL). Although the accuracy is lower than argmax behavior assumption, our further analysis on the detailed results of argmax assumption recommends Monte Carlo simulation approach for applications of individual-level prediction. The next paragraph will address the detailed arguments in terms of travel mode share.

Table 4. Comparisons of % of accurate predictions with respect to individual choice function.

	Argmax		**Monte Carlo Simulation**	
Model	Learning	Test	Learning	Test
Nested Logit	63.95 ± 0.25	63.72 ± 0.80	51.84 ± 0.59	52.52 ± 1.19
CNL	63.77 ± 0.37	63.16 ± 1.01	51.84 ± 0.52	52.12 ± 0.86
MLP(S)	63.65 ± 0.49	63.52 ± 0.92	52.36 ± 0.96	52.05 ± 1.31
MLP(SSSS)	60.61 ± 0.70	60.63 ± 0.92	48.85 ± 1.07	48.61 ± 1.30
MLP(RRRS)	75.36 ± 5.72	67.76 ± 5.23	68.65 ± 4.68	62.31 ± 4.20
DNN(RRRS)	75.49 ± 0.52	72.08 ± 0.71	65.32 ± 0.72	62.28 ± 1.27
DNN-A(RRRS)	75.47 ± 0.86	71.84 ± 0.81	65.13 ± 0.73	62.80 ± 1.26

The detailed predictive characteristics for each travel mode using confusion matrices are shown in Tables 5 and 6. Table 5 shows the average mode share of the 30 test sets having 2031 trips, based on the argmax behavior assumption, selecting the choice giving maximum probability among alternatives. CNL (a) underestimates the mode share of rail at only 4%. This is because RUMs tend to underscore the alternative that has low frequencies of the observation. Similarly, the argmax function with CLN overestimates the ratio of SM (90.7%). The average values of observed mode shares are rail: 13.6%, SM: 60.4%, and car: 26.0%. A naïve predictor might have at least 60.4% accuracy if a model predicts the probability of all individuals' SM choice to be highest among alternatives. The prediction accuracy of CNL is slightly higher at approximately 63.3%.

DNN and DNN-A with the argmax assumption have higher accuracy than CNL. DNN's mode share is (rail: 10.7%, SM: 67.4%, car: 21.9%). Furthermore, the results of DNN-A and DNN show that underestimating or overestimating of specific alternatives are significantly reduced. For example, the argmax assumption predicts 90.7% mode share for SM. DNN-A better predicts 69.8%, which is approximately 9% higher than the observed mode share. These mode shares are closer to the observed share but there still remain gaps. This implies that this approach might be problematic when used to predict each individual's choice using the maximum probability alternative (argmax). This problem has been recognized in the past, and Shalaby [48] has argued that the percentage of correct predictions is an inappropriate measure to check the goodness-of-fit for such models. However, predicting individual choices becomes important in transportation services' marketing and agent-based travel modeling.

The Monte-Carlo simulation behavior assumption (Table 6) is likely to better predict the actual shares for all choice models. The gap between predicted SM share and observed value for DNN-A decreased to 0.8%. In addition, it is also clear that applying the proposed AAF (availability of alternatives function) has led to an increase in the DNN's performance even further in the mode share of both SM and rail. The accuracy of SM of DNN-A increases to 70.7% from 70.2% (DNN), and rail of DNN-A is 44.4 which is 1.2% higher than DNN. Those improvements are from individuals who do not own a car. The predicted mode shares of DNN-A are rail: 13.3%, SM: 61.2%, car: 25.5%, which are closer to the observed ratio than other models.

Table 5. Confusion matrices and mode share prediction accuracy (argmax).

(a) Cross Nested Logit	**Predicted mode**			**Observed**	
Mode	Rail	SM	Car	Trips	Mode share
Rail	52	216	8	276	13.6%
SM	26	1168	33	1227	60.4%
Car	3	459	66	528	26.0%
Trips (predicted)	81	1843	108	2031	100%
Mode share (predicted)	4.0%	90.7%	5.3%	100%	
Correct prediction	64.5%	63.4%	61.3%	63.3%	

(b) DNN (RRRS)	**Rail**	**SM**	**Car**	**Trips**	**Mode share**
Rail	125	136	15	276	13.6%
SM	60	1053	115	1227	60.4%
Car	6	236	286	528	26.0%
Trips (predicted)	191	1425	415	2031	100.0%
Mode share (predicted)	9.4%	70.2%	20.5%	100%	
Correct prediction	65.5%	73.9%	68.8%	72.1%	

(c) DNN-A (RRRS)	**Rail**	**SM**	**Car**	**Trips**	**Mode share**
Rail	122	138	16	276	13.6%
SM	54	1047	126	1228	60.4%
Car	5	233	290	528	26.0%
Trips (predicted)	182	1418	431	2031	100%
Mode share (predicted)	8.9%	69.8%	21.2%	100%	
Correct prediction	67.3%	73.8%	67.2%	71.8%	

Table 6. Confusion matrices and mode share prediction accuracy (Monte Carlo simulation).

(a) Cross Nested Logit	**Predicted mode**			**Observed**	
Mode	Rail	SM	Car	Trips	Mode share
Rail	82	156	38	276	13.6%
SM	147	776	305	1228	60.4%
Car	35	294	199	528	26.0%
Trips (predicted)	263	1226	542	2031	100%
Mode share (predicted)	12.9%	60.4%	26.7%	100%	
Correct prediction	31.1%	63.3%	36.8%	52.0%	

(b) DNN (RRRS)	**Rail**	**SM**	**Car**	**Trips**	**Mode share**
Rail	114	139	23	276	13.6%
SM	128	882	217	1228	60.4%
Car	23	235	270	528	26.0%
Trips (predicted)	265	1256	510	2031	100%
Mode share (predicted)	13.0%	61.8%	25.1%	100%	
Correct prediction	43.2%	70.2%	53.0%	62.4%	

(c) DNN-A (RRRS)	**Rail**	**SM**	**Car**	**Trips**	**Mode share**
Rail	120	132	24	276	13.6%
SM	128	879	221	1228	60.4%
Car	23	232	273	528	26.0%
Trips (predicted)	270	1243	517	2031	100.0%
Mode share (predicted)	13.3%	61.2%	25.5%	100%	
Correct prediction	44.4%	70.7%	52.8%	62.7%	

6. Conclusions and Discussion

This paper proposes a deep neural network model for travel choice prediction by considering individual-level prediction. Deep learning techniques are at the forefront of various fields of research and inquiry because of their ability to represent many functional forms and their strong prediction potential. Test implementation of deep neural networks shows that they outperform traditional discrete choice models such as nested logit and cross-nested logit models, as well as the simpler multilayer perceptron neural nets attempted in the past. In addition, we recognize that unavailable alternatives for certain individuals can result in unreasonable outputs from such models. Thus, our research develops a function to handle the availability of alternatives and incorporate it into the deep neural net model. Although setting the probabilities of unavailable alternatives is common in random utility models practice, existing neural network models have not implemented it, leading to poor performance. We address this issue in our study. Deep neural network structures allow for several environmental settings such as a hidden layer structure, activation functions, the number of epochs for training, the dropout ratio, and the training set ratio. By examining the characteristics of each such detail, our research finds appropriate structures and parameters to propose the successful final model.

The research used a publicly available dataset of stated preference data for Swiss Metro, and compared the performances of the proposed model with other models. Experiments show that our model yields better performance than existing models. In terms of overall predictive potential, the proposed model has the highest log-likelihood and percentage of correct predictions among models. Both versions of deep neural networks that we studied showed high predictive potential, but the version that incorporates a function for the availability of alternatives shows better model accuracy. Market share (Mode share ratio) is also predicted well by the proposed model. The mode share ratio is estimated in both average probabilities of each mode and predicted choices' ratio. Whereas random utility models suffer from underestimation for the relatively less used modes (rail in our study case) and overestimation for the most preferred mode (Swiss Metro), the deep neural networks estimate the mode choice closer to the observed mode choice.

In addition, our experiments recommend that Monte Carlo simulation approach better estimates travel mode share than an argmax assumption that machine learning studies commonly apply. This finding is important for smart mobility applications because service managers need to estimate the possible market share of each travel mode to design service and to predict individual behavior accurately at the same time. Besides, the success of agent-based modeling, which provides insights into sustainable transportation policies, depends on the accurate prediction of both aggregated behavior (mode share) and individual behavior. An agent-based model generates travelers and predicts travel modes of each traveler. Our analysis implies that a traditional mode choice model is not capable of capturing individual choice. Predictions using both argmax and Monte Carlo simulation bring poor results. Although DNN models can predict individual choice better, mode share might be worse if a modeler assumes that a traveler selects a travel option that has the best utility (probability), which is a commonly used method (argmax) in machine learning applications. In contrast, our proposed DNN-A with Monte Carlo simulation predicts reasonable mode share and provides correct predictions of individual mode choice behavior compared to other models.

This research utilized a stated preference data set that assumes that a proposed new alternative mode will affect travel mode choice behavior. This virtual alternative could introduce certain biases since people are generally sympathetic to it in such surveys. Thus, future studies with our models will need to employ revealed-preference data sets. The major argument against neural network models has always been that their input-output process is like a black box, in that it does not offer means for statistical interpretations such as odds ratio, elasticity, and sensitivity. The authors acknowledge this criticism and offer improved accuracy of these models as a tradeoff, which our study amply demonstrates. Alternative suggestions could be considered, such as the simulation approach in Mohammadian and Miller [14] to assess the model output changes by changing input values. We leave these enhancements for future work.

Author Contributions: Conceptualization, D.N. and J.C.; methodology, D.N. and J.C.; software, D.N.; validation, D.N. and J.C.; writing—original draft preparation, D.N. and J.C.; writing—review and editing, D.N. and J.C.; visualization, D.N. All authors have read and agreed to the published version of the manuscript

Funding: Author Jaewoo Cho was financially supported by Hansung University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. McFadden, D. Modeling the Choice of Residential Location. *Transp. Res. Rec.* **1978**, *673*, 72–77.
2. Small, K.A. A Discrete Choice Model for Ordered Alternatives. *Econometrica* **1987**, *55*, 409–424. [CrossRef]
3. Vovsha, P. Application of Cross-Nested Logit Model to Mode Choice in Tel Aviv, Israel, Metropolitan Area. *Transp. Res. Rec.* **1997**, *1607*, 6–15. [CrossRef]
4. Ben-Akiva, M.; Bierlaire, M. Discrete choice methods and their applications to short term travel decisions. In *Handbook of Transportation Science*; Hall, R.W., Ed.; Springer: Boston, MA, USA, 1999; Volume 23, pp. 5–33.
5. Brownstone, D.; Golob, T.F. The Effectiveness of Ridesharing Incentives: Discrete-choice Models of Commuting in Southern California. *Reg. Sci. Urban Econ.* **1992**, *22*, 5–24. [CrossRef]
6. Bierlaire, M.; Axhausen, K.W.; Abay, G. The Acceptance of Modal Innovation: The Case of Swissmetro. Presented at 1st Swiss Transport Research Conference, Ascona, Switzerland, 1–3 March 2001. Available online: https://pdfs.semanticscholar.org/17f0/3a3269e64b9c69be115a3d47923c80151472.pdf?_ga=2.231110047.1256215494.1599662572-512884126.1594177302 (accessed on 10 August 2020).
7. Bierlaire, M. A Theoretical Analysis of the Cross-Nested Logit Model. *Ann. Oper. Res.* **2006**, *144*, 287–300. [CrossRef]
8. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv* **2016**, arXiv:abs/1603.04467. Available online: https://inspirehep.net/literature/1663031 (accessed on 10 August 2020).
9. Deng, L.; Yu, D. *Deep Learning: Methods and Applications*; Foundations and Trends®in Signal Processing: Hanover, Massachusetts, MA, USA, 2013; Volume 7, pp. 197–387. Available online: https://www.microsoft.com/en-us/research/publication/deep-learning-methods-and-applications/ (accessed on 10 August 2020).
10. Angelova, A.; Krizhevsky, A.; Vanhoucke, V. Pedestrian Detection with a Large-Field-Of-View Deep Network. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, Washington, WA, USA, 26–30 May 2015; pp. 704–711.
11. Najafabadi, M.M.; Villanustre, F.; Khoshgoftaar, T.M.; Seliya, N.; Wald, R.; Muharemagic, E. Deep Learning Applications and Challenges in Big Data Analytics. *J. Big Data* **2015**, *2*, 1–21. [CrossRef]
12. De Carvalho, M.C.M.; Dougherty, M.S.; Fowkes, A.S.; Wardman, M.R. Forecasting Travel Demand: A Comparison of Logit and Artificial Neural Network Methods. *J. Oper. Res. Soc.* **1998**, *49*, 717–722. [CrossRef]
13. Hensher, D.A.; Ton, T.T. A Comparison of the Predictive Potential of Artificial Neural Networks and Nested Logit Models for Commuter Mode Choice. *Transp. Res. E-Log.* **2000**, *36*, 155–172. [CrossRef]
14. Mohammadian, A.; Miller, E.J. Nested Logit Models and Artificial Neural Networks for Predicting Household Automobile Choices: Comparison of Performance. *Transp. Res. Rec.* **2002**, *1807*, 92–100. [CrossRef]
15. Cantarella, G.E.; de Luca, S. Multilayer Feedforward Networks for Transportation Mode Choice Analysis: An Analysis and a Comparison with Random Utility Models. *Transp. Res. Part C Emerg. Technol.* **2005**, *13*, 121–155. [CrossRef]
16. Zhang, Y.L.; Xie, Y.C. Travel Mode Choice Modeling with Support Vector Machines. *Transp. Res. Rec. J. Transp. Res. Board* **2008**, *2076*, 141–150. [CrossRef]
17. Omrani, H.; Charif, O.; Gerber, P.; Awasthi, A.; Trigano, P. Prediction of Individual Travel Mode with Evidential Neural Network Model. *Transp. Res. Rec. J. Transp. Res. Board* **2013**, *2399*, 1–8. [CrossRef]
18. Omrani, H. Predicting Travel Mode of Individuals by Machine Learning. *Transp. Res. Procedia* **2015**, *10*, 840–849. [CrossRef]
19. Jing, P.; Zhao, M.X.; He, M.L.; Chen, L. Travel Mode and Travel Route Choice Behavior based on Random Regret Minimization: A systematic Review. *Sustainability* **2018**, *10*, 1185. [CrossRef]

20. Sifringer, B.; Lurkin, V.; Alahi, A. Let Me Not Lie: Learning MultiNomial Logit. *arXiv* **2018**, arXiv:abs/1812.09747. Available online: https://www.semanticscholar.org/paper/Let-Me-Not-Lie%3A-Learning-MultiNomial-Logit-Sifringer-Lurkin/5e6fc2403c8f5152385b5c5b9d652a7d9164050d (accessed on 10 August 2020).
21. Han, Y.F.; Zegras, C.; Pereira, F.C.; Ben-Akiva, M. A Neural-embedded Choice Model: TasteNet-MNL Modeling Taste Heterogeneity with Flexibility and Interpretability. 2020. Available online: https://www.groundai.com/project/a-neural-embedded-choice-model-tastenet-mnl-modeling-taste-heterogeneity-with-flexibility-and-interpretability/1 (accessed on 10 August 2020).
22. Hinton, G.E.; Osindero, S.; Teh, Y.W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
23. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.
24. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfittin. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
25. McFadden, D. *Conditional Logit Analysis of Qualitative Choice Behavior*; Institute of Urban and Regional Development: Berkeley, CA, USA, 1972.
26. Cheng, S.; Long, J.S. Testing for IIA in the Multinomial Logit Model. *Sociol. Methods Res.* **2007**, *35*, 583–600. [CrossRef]
27. Bierlaire, M. BIOGEME: A Free Package for the Estimation of Discrete Choice Models. In Proceedings of the 3rd Swiss Transport Research Conference, Ascona, Switzerland, 19–21 March 2003. Available online: https://www.semanticscholar.org/paper/BIOGEME%3A-a-free-package-for-the-estimation-of-Bierlaire/8a1519c9dcc4f07c65c98ff3f4b42fae18ead592 (accessed on 10 August 2020).
28. Train, K. A Structured Logit Model of Auto-Ownership and Mode-Choice. *Rev. Econ. Stud.* **1980**, *47*, 357–370. [CrossRef]
29. Cascetta, E.; Papola, A. Random Utility Models with Implicit Availability/Perception of Choice Alternatives for the Simulation of Travel Demand. *Transp. Res. Part C Emerg. Technol.* **2001**, *9*, 249–263. [CrossRef]
30. San Diego. *Activity-Based Travel Model Validation for 2012*; San Diego Association of Governments (SANDAG): San Diego, CA, USA, 2016. Available online: https://www.sandag.org/uploads/publicationid/publicationid_2097_21613.pdf (accessed on 10 August 2020).
31. SACOG. 2020 Metropolitan Transportation Plan/Sustainable Communities Strategy. 2019. Available online: https://www.sacog.org/2020-metropolitan-transportation-plansustainable-communities-strategy-update (accessed on 10 August 2020).
32. Southern California Association of Governments. SCAG Regional Travel Demand Model and 2012 Model Validation. 2016. Available online: https://www.scag.ca.gov/Documents/Forms/NewDisplayForm.aspx?ID=1490 (accessed on 10 August 2020).
33. Xu, B.; Wang, N.Y.; Chen, T.Q.; Li, M. Empirical Evaluation of Rectified Activations in Convolution Network. *arXiv* **2015**, arXiv:1505.00853v2. Available online: https://paperswithcode.com/paper/empirical-evaluation-of-rectified-activations (accessed on 10 August 2020).
34. Glorot, X.; Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. *J. Mach. Learn. Res.* **2010**, *9*, 249–256.
35. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in {P}ython. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
36. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, GA, USA, 16–21 June 2013. Available online: https://www.semanticscholar.org/paper/Rectifier-Nonlinearities-Improve-Neural-Network-Maas/367f2c63a6f6a10b3b64b8729d601e69337ee3cc (accessed on 10 August 2020).
37. Dahl, G.E.; Sainath, T.N.; Hinton, G.E. Improving Deep Neural Networks for LVCSR using Rectified Linear Units and Dropout. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 8609–8613. Available online: https://www.semanticscholar.org/paper/Improving-deep-neural-networks-for-LVCSR-using-and-Dahl-Sainath/1a3c74c7b11ad5635570932577cdde2a3f7a6a5c (accessed on 10 August 2020).

38. Silva, L.M.; Marques de Sá, J.; Alexandre, L.A. Data classification with multilayer perceptrons using a generalized error function. *Neural Netw.* **2008**, *21*, 1302–1310. [CrossRef]
39. Salimans, T.; Kingma, D.P. Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. *arXiv* **2016**, arXiv:1602.07868v3, 901–909. Available online: https://www.semanticscholar.org/paper/Weight-Normalization%3A-A-Simple-Reparameterization-Salimans-Kingma/3d2c6941a9b4608ba52b328369a3352db2092ae0 (accessed on 10 August 2020).
40. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980v9, 1–15. Available online: https://www.semanticscholar.org/paper/Adam%3A-A-Method-for-Stochastic-Optimization-Kingma-Ba/a6cb366736791bcccc5c8639de5a8f9636bf87e8 (accessed on 10 August 2020).
41. Abay, G. Nachfrageabschätzung Swissmetro: Eine Stated-Preference Analyse. *EDMZ*. 1999. Available online: https://books.google.com.ph/books/about/Nachfrageabsch%C3%A4tzung_Swissmetro_eine_St.html?id=EKDloAEACAAJ&redir_esc=y (accessed on 10 August 2020).
42. Hess, S.; Bierlaire, M.; Polak, J.W. Capturing Correlation and Taste Heterogeneity with Mixed GEV Models. In *Applications of Simulation Methods in Environmental and Resource Economics*; Scarpa, R., Alberini, A., Eds.; Springer: Dordrecht, Netherlands, 2005; pp. 55–75.
43. Hess, S.; Polak, J.W.; Bierlaire, M. Confounding between Taste Heterogeneity and Error Structure in Discrete Choice Models. 2006. Available online: https://stuff.mit.edu/afs/athena/course/11/11.951/oldstuff/albacete/Other_Documents/Europe%20Transport%20Conference/advances_in_discrete_c/confounding_betwee1634.pdf (accessed on 10 August 2020).
44. Ortelli, N.; Hillel, T.; Pereira, F.C.; Bierlaire, M.; Lapparent, M. De Assisted Specification of Discrete Choice Models. 2020. Available online: https://transp-or.epfl.ch/documents/technicalReports/OrtHilPerLapBie2020.pdf (accessed on 10 August 2020).
45. SANDAG Travel Demand Model and Forecasting Documentation. Available online: https://www.sdforward.com/pdfs/RP_final/AppendixT-SANDAGTravelDemandModelDocumentation.pdf (accessed on 9 March 2020).
46. Giaimo, G.T.; Schiffer, R. California Statewide Travel Demand Model. 2014. Available online: http://onlinepubs.trb.org/onlinepubs/circulars/ec075.pdf (accessed on 10 August 2020).
47. De Dios Ortúzar, J.; Willumsen, L.G. *Modelling Transport*, 4th ed.; John wiley & sons: Hoboken, NJ, USA, 2011. Available online: https://www.wiley.com/en-us/Modelling+Transport%2C+4th+Edition-p-9781119993520 (accessed on 10 August 2020).
48. Shalaby, A.S. Investigating the Role of Relative Level-of-Service Characteristics in Explaining Mode Split for the Work Trip. *Transp. Plan. Technol.* **1998**, *22*, 125–148. [CrossRef]