A Model Based on Deep Learning for Predicting Travel Mode Choice

4 authors:

Daisik Nam
Inha University
20 PUBLICATIONS 139 CITATIONS
SEE PROFILE

Hyunmyung Kim
Myongji University
43 PUBLICATIONS 376 CITATIONS
SEE PROFILE

Jaewoo Cho
Hansung University
2 PUBLICATIONS 24 CITATIONS
SEE PROFILE

R. Jayakrishnan
University of California, Irvine
118 PUBLICATIONS 3,106 CITATIONS
SEE PROFILE

A Model Based on Deep Learning for Predicting Travel Mode Choice

Daisik Nam*
PhD Student
Department of Civil and Environmental Engineering
Institute of Transportation Studies
4070 Anteater Instruction and Research Building (AIRB)
University of California
Irvine, CA 92697
(949) 533-2768
nds1027@gmail.com

Hyunmyung Kim
Associate Professor
Department of Transportation Engineering
San 38-2 Nam-dong, Cheoin-gu, Yongin-si, Gyeonggi-do, South Korea
Myong Ji University
(82) 10-2598-4172
khclsy@gmail.com

Jaewoo Cho
PhD Candidate
Department of Planning Policy and Design
Social Ecology
300 Social Ecology I
University of California
Irvine, CA 92697

R., Jayakrishnan
Professor
Department of Civil and Environmental Engineering
4014 Anteater Instruction and Research Building (AIRB)
Institute of Transportation Studies
University of California, Irvine
Irvine, CA 92697
(949) 209-7302
rjayakri@uci.edu

* Corresponding Author

Words: 5,488 words
Figures: 5 = 1,250 words
Tables: 3 = 750 words
Total = Words (5,488) + Figures & Tables (2,000) = 7,488 words
Submission Date: Aug 01 2016

ABSTARCT

Recognizing the limitations of previous travel mode choice models such as random utility models and multi-layer perceptron neural network models, this study develops a framework using a deep neural network with deep learning schemes, to predict travelers' mode choice behavior. Deep neural networks and deep learning are relatively newer models, applied mostly so far to pattern recognition, image/voice processing, and for big data analytics. We develop such a choice model with a structure that is appropriate for the travel mode choice problem, and demonstrate the success of the model using an available dataset. The research also develops an important component of the model that takes into account the inherent characteristics of choice models that all individuals have different choice alternatives, an aspect not considered in the neural network models of the past that led to poorer performance. The proposed model is compared against existing mode choice models. The results prove that the new model clearly outperforms the previous mode choice models.

KEYWORDS: Travel mode choice model, Deep learning, Deep neural network, Multi-layer perceptron model, Random Utility model.

1. INTRODUCTION

The advantage of Artificial Intelligence (AI) is 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. We will compare the proposed model to existing travel choice models. Random utility models (RUMs) are traditionally used to predict travelers' choice, but such models typically have strong assumptions along with limitations in their accuracy. RUMs assume that individuals select the alternative giving maximum utility and that an individual's utility can be calculated with linear combinations of deterministic elements and unseen errors. Based on the random utility model, researchers proposed various types of discrete choice models (logit model, nested logit model, cross–nested logit model, and probit model) in order to better capture individuals' choice behaviors (1,2,3,4,5,6 and 7). The random utility models however have inherent limitations since an individual's choice is not as simple as in the assumptions. Mode choice models play an active and pivotal role in a transportation planning process. In transportation planning models, a mode choice model is necessary to predict the demands for each mode of travel (such as autos, bus, rail, and walk). In addition, the market share ratio (modal split ratio), which can be inferred from the choice model, is used as a key index by policy makers. Thus, the quality of the behavior representation affects not only the reliability of the transportation models, but also the success of the transportation policies.

With various large datasets currently becoming available in the transportation domain, developing novel algorithms is necessary for improving prediction accuracy. Several researchers and companies are extending their efforts towards the machine learning area, which has shown success in various tasks related to big data (8). We might be able to obtain information from various sources such as digital maps, real-time transportation information, users' profiles, and users' activity logs from smart-phone based applications. Several companies, such as Google, Microsoft, and Amazon, are now able to acquire their users' personal preferences. Furthermore, with the development of transportation network services such as by Uber, Lyft, Zip Car, etc., and applications such as Google directions, predicting travel choices at the individual level becomes even more important. Traditional transportation models, however, cannot benefit from such large datasets because they have mainly focused on inferring population behaviors from canonical sample data sets.

Deep learning is one promising area in Big Data Analytics. It is commonly applied to computer vision, pedestrian detection, language modeling, picture classification, and speech recognition (9,10, and 11). However, in the transportation field, to the best of the authors' knowledge, a deep learning-based mode choice model is yet to be developed.

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, 13, 14, and 15). Multi-layer perceptron models (MLP) were the most commonly used AI scheme for mode choice models. From several experiments, Carvalho et al. (12) concluded that it is hard to say that MLP captures travelers' choice behavior better than a logit model, and that AI brings no advantages in computing times for optimization either. Hensher and Ton (13) compared the performance of Nested Logit Models (NLM) against MLP in modeling commuter mode choice. They found that although ANN forecasts individuals' choices better than NLM, NLM predicts market share ratios slightly better. Cantarella and Luca (15) examined a multilayer feed-forward 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 that show AI models to outperform RUMs such as NLM and Cross Nested Logit model (CNLM) in predicting travel mode choice (16 ,17 and 18). In summary, the results from past research are somewhat mixed in comparing the performance of traditional discrete choice models and neural network models.

Since 2006, when Hinton proposed several techniques for deep learning structures, configuring deep networks with more than three layers have shown widespread success in training neural networks. In the past, using more than 2 layers and a large number of perceptrons in neural networks generally proved to be unsuccessful (9). We find that previous AI based travel mode choice models can also be regarded as using shallow network structures with a single layer and a small number of perceptrons. This was however mainly because a high number of layers would result in an over-fitting problem, as many realized, in addition to the computing time being considerable. As is known, overfitting would make the estimated model fit closely to the sample training set, but perform poorly on a real dataset. With the evolution of deep learning theory, these problems can be tackled with several newer techniques: efficient initialization, stochastic gradient descent method, regularization, dropout, etc. In addition, parallel processors such as the Graphics Processing Unit (GPU) can boost the computing speed, which has resulted in significant increase the application of DL significantly.

This study proposes Deep Neural Network (DNN), one among a family of deep learning algorithms, to predict travelers' mode choice behavior. The next section explains the experimental data used in the study. Section 3 briefly describes previous mode choice models such as random utility models and MLPs, and introduces DNN. In the following section, we describe how we construct our mode choice models. The performance of each model is evaluated in the fifth section. Finally, we will close our paper with conclusions and future research.

2. Experimental Data

To explore the performance of the Deep Neural Network model, we utilized data from a mode choice survey of long-distance travel. Abay (19) 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, 19, 20, 21, and 22) The SP survey data is available on the Biogeme website with discrete choice estimation packages (20). 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 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)

The total number of individual observations is 6,768. Prior machine learning research partitioned their observations into two subsets: learning set and test set (12,15,17, and 18) or three subsets: learning set, validation set, and test set (16). Our research separates the observations into two subsets. The total number of the sample is equal to a sum of the learning set and the test set. The learning set is independent from the test set, and they have no common observations. The learning set is used to train a 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 learning set could induce the over-fitting problem, 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 learning set could have high explanatory power for the observation set. But allocating high learning rate is undesirable in that a small test set could also be biased. This research will examine the effect of learning rate to these stated problems by changing the rate from zero to one. Furthermore, we will also delve into the prediction power relationship between learning set and test set.

3. Discrete Choice Model descriptions

3.1 Random Utility Model

Random Utility Models (RUMs) are used for our comparative analysis. The main assumption of RUM is that the error term of each alternative should be independent. Considering that rail and Swiss Metro (SM) in the choice set could be mutually relevant, we used the Nested Logit model and, which set a choice hierarchy. A Cross Nested Logit model (CNL) is used because both rail and SM are public transportation options with mixed interactions that a Nested Logit model would not capture. Small (2) initially proposed CNL, which many researchers theoretically analyzed (20).

Our research refers to Bierlaire's nested structure (6). With the two nests, a choice is determined by the combination of the existing mode and the mode's attributes. The utility function of a nested logit model has shared unobserved attributes (errors). Eq (1) indicates the utility function for the NL and CNL.

$$\boldsymbol{U_{em}^n = V_e^n + V_m^n + V_{em}^n + \varepsilon_e^n + \varepsilon_{em}^n} \qquad (1)$$

where

e $=$ *existing mode*
m $=$ *travel mode* $(Car, Rail, SM)$
V_e^n $=$ *Deterministic component of the utility of existing*
V_m^n $=$ *Deterministic component of the utility of mod*e
V_{em}^n $=$ *Deterministic component of the common utility among existing and mode*
ε_e^n $=$ *error term for ex□sting or non exististing* $-$ *assumed Gumbel*$(0, u_e)$
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 Eq (2). 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, $\mathrm{u_e}$). Mathematically, the error term on the upper nest ($\varepsilon_{\mathrm{e}}^{\mathrm{n}}$) does not involve the lower level.

$$\boldsymbol{U_{em}^n = V_m^n + \varepsilon_e^n + \varepsilon_{em}^n} \qquad (2)$$

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 Eq (3). The detailed information about NL and CNL for the data set is explicitly explained in (20).

$$\boldsymbol{Pr^n(e, m|C_n) = Pr^n(e|E)Pr^n(m|e)} \qquad (3)$$

3.2 Artificial Neural Network and Multi-Layer Perceptron Models

Artificial Neural Networks (ANN) is a learning algorithm that imitates the human neural system. An ANN consists of multiple nodes, called neurons, that communicates through synapses. Typically, there are three sets of nodes: input nodes, intermediate nodes, and output nodes and each type of node plays unique roles. Input nodes receive input information, output nodes yield output signals, and intermediate nodes receive signals from input nodes, and manipulate the information to give results to output nodes. 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 multi-layer perception (MLP) model.

MLPs typically use backpropagation, which starts with randomly weighted synapses and trains them with input and output values. A simplest type of MLP is a feed forward 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, and their greater learning and prediction power is the most significant. In addition, MLP employs transfer functions that modify input signals and pass them to nodes in the next intermediate layer, using weights and biases. Eq (4) presents the specific transfer function. It should be noted that there are various types of functions such as sigmoid, tanh, and ReLU, to estimate parameters, and their thresholds are shown in Figure 1. ReLU is a rectified linear unit that Nair and Hinton proposed in 2010 (23). 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 (24).

$$\mathbf{Y} = \boldsymbol{f}\left(\sum_{i=1}^{n}(\boldsymbol{W}_i\boldsymbol{Z}_i + \boldsymbol{\beta})\right) \tag{4}$$

n = number of input signals to a node
W = weights
Z = inputs
B = bias term

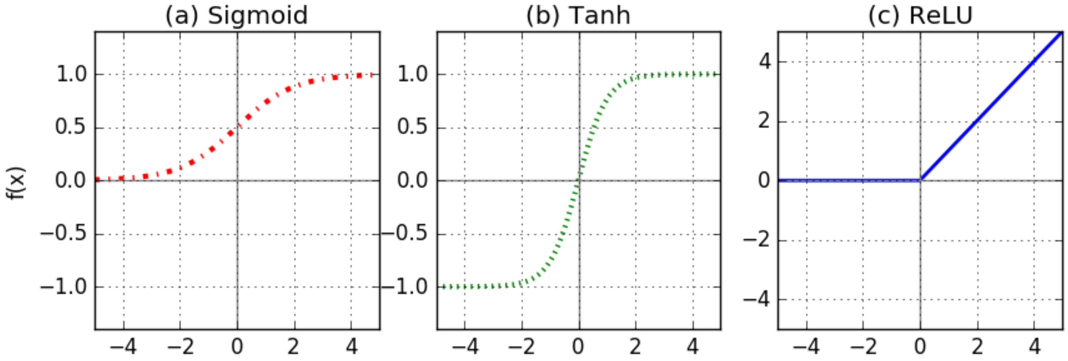


FIGURE 1 Three common transfer/activation functions

Another characteristic of an MLP is that there are training processes to estimate parameters that minimize a cost function. There are several ways to train Multiple-layer perceptrons. MLP finds each layer's parameter with a backpropagation method and optimization techniques. MLP is a nonlinear problem, so past research applies heuristics such as genetic algorithms or a gradient descent method (14 and 18). But these heuristics could be trapped in poor local optima when the learning process is initiated from a wrong point (9 and 15).

3.3. Deep Neural Network with a Function for Availability of Alternatives

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 significantly greater number of hidden (intermediate) layers than an MLP. With MLP, it has been reported that increasing the number of hidden layers produced several problems. First, more hidden layers exponentially increase required computing resources. Second, although more hidden layers and nodes contributes to model accuracy and prediction power, the local optimum or overfitting problems arise as tradeoffs. Third, processing time varies highly depending on parameter initializations. As a result, researchers concluded that adding more hidden layers bring no benefits or has even negative effects in model estimations.

To cope with these problems, in 2006, Hinton (25) suggests implementing unsupervised learning to initialize parameters and then executing supervised learning in order to estimate parameters in an efficient manner. In the following studies, Glorot and Bengio (26) devised a simplified version of the initialization process, known as Xavier initialization (after Glorot’s first name), that improves the practical application of the initialization process. The dramatic evolution of parallel computing utilizing graphical processing units (GPU) then further spurred the adoption of deep learning models.

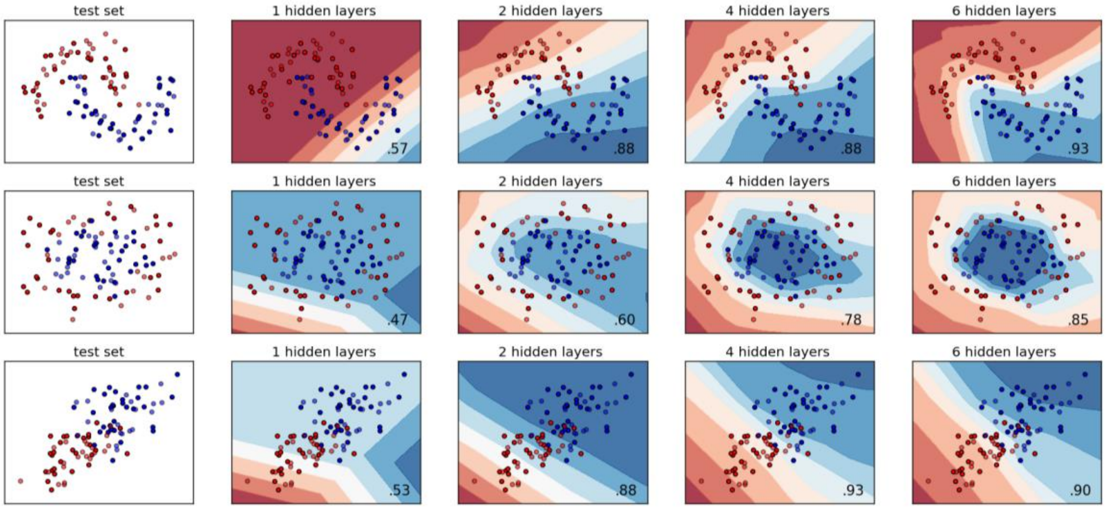


FIGURE 2 The predictive potential according to the number of hidden layers

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 sample of data sets assuming travel time (x1) and cost (x2) as the explanatory variables, as in Figure 2, with two colors showing the binary mode choices associated with each data point. In here, we use a simple neural network with two input variables (x1, x2) and two perceptrons for this data. The figure 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. The numbers in the sub graphs show the fraction of accurate predictions. One hidden layer with two perceptrons linearly divides the space. 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. By increasing the number of hidden layers, the overall predictive potential increases, since the next layer divides the space of the previous step, which could recognize more complex decision patterns. This process is confirmed 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 the overfitting problem. In Figure 3, we show our proposed structure with deep neural networks (DNN), and we now proceed to discuss the details of the component shown in the figure.

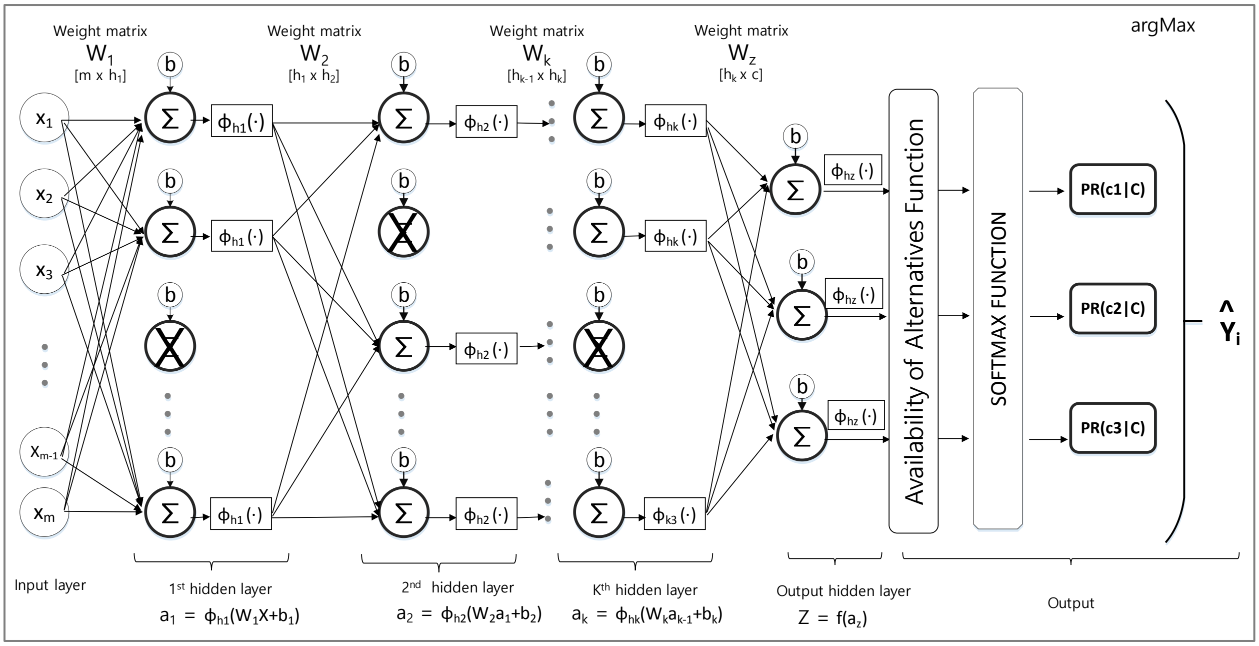


FIGURE 3 Overall structure of proposed deep neural network model

The first issue to address is overfitting. While DNNs are a powerful machine learning model, overfitting is a serious problem to deal with. It is caused by sampling noise, 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 prediction power with the training set is very high but the model cannot predict effectively with non-training data.

There are several methods to reduce overfitting, and one of the most popular tools is regularization. It provides modifications or weight penalties to a learning algorithm in order to reduce its generalization error while the training error is untouched. However, this method causes 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 the overfitting issue even in large networks.

The next item to discuss in the DNN structure in Figure 3 refers to the alternatives in the choice set of individuals. In a choice prediction model, 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 mode should be eliminated from the choice set. In conventional discrete choice models, unavailable alternatives are controlled by increasing the alternative’s utility, for a neural network, we need to develop a method to handle the issue.

Thus, we propose the Availability of Alternatives Function (AAF) to add to the DNN, shown as a vertical box in Figure 3. From the feedforward process in Eq (5) below, the dimension of the perceptron in the last hidden layer becomes same as the number of alternatives. Previous DNN models have not dealt with unavailable alternatives. Thus these alternatives could also have a value in those models, since the softmax function in the last step (Eq 7) calculates each alternative's choice probability based on the last layer's output. Thus, it is important to force the probability of unavailable alternatives to be zero, as in Eq (6). Otherwise, the model might predict unreasonable results to certain individuals. 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 probability of j becomes to zero.

$$\mathbf{a_z} = \phi_{\boldsymbol{z}}(\boldsymbol{a_k}) = \phi_{\boldsymbol{z}}(\boldsymbol{W_z}(\phi_{\boldsymbol{hk}}(\mathbf{W}_{\boldsymbol{k}}\boldsymbol{a_{k-1}} + \boldsymbol{b_k}) + \boldsymbol{b_z}) \tag{5}$$

$$\mathbf{Z} = \mathbf{AAF}(\mathbf{a_o}) = \begin{Bmatrix} \boldsymbol{z_j} = & \boldsymbol{a_z} & \boldsymbol{if\ j \in C^n} & \boldsymbol{for} \forall \boldsymbol{j, n} \\ \boldsymbol{z_j} = & -\infty & \boldsymbol{if\ j \notin C^n} & \boldsymbol{for} \forall \boldsymbol{j, n} \end{Bmatrix} \tag{6}$$

$$\mathbf{Pr}(\boldsymbol{j}|\boldsymbol{C^n}) = \boldsymbol{softmax}(\boldsymbol{Z}) = \frac{\mathbf{exp}(\boldsymbol{z_j})}{\sum_{\boldsymbol{k \in C^n}} \mathbf{exp}(\boldsymbol{z_k})} \tag{7}$$

4. Experimental Setting

In the Nested Logit model (NL) and the Cross Nested Logit model (CNL), our utility functions are configured as the utility functions in Bierlaire et al. (6). They found that both travel time and travel cost generally affect all travel modes' choice probability. The relevance of other variables depended on the modes. 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 number of seats. In our work, the multiple training sets are imported to the Biogeme software (20) 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 0.5. 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.

Instead of including all possible explanatory variables in MLPs and DNN, identifying significant variables from traditional discrete choice models, such as a nested logit model, could increase computational efficiency, since a large number of input variables to a neural network could make it too complex a network to train. DNN has a considerably complex network structure with many layers. All perceptrons in the first hidden layer are directionally linked from input variables and this affects the results. The concept of selecting input variables from a traditional model is very significant, in that we can avoid unnecessary computing processes.

The mechanisms of both MLP and DNN are similar. Both structures have hidden layers, neurons, activation functions for each hidden layer, a number of epochs for training, and a learning rate. An epoch, as is well-known, is a single pass through the whole training set. We set the learning rate at 0.001. The cost function we set for these models is a cross-entropy function (Eq 8), which can be derived from the maximum likelihood principle (27). The last hidden layer of both types of 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 ReLU function inherently generates a zero value when an input value is less than zero. From our preliminary experience, we found that ReLU could not provide each alternative's choice probability. This is a serious drawback since calculating a market share ratio from the model is an important purpose of a choice model. It does not mean that applying ReLU function is not suitable to the travel choice model because ReLU function is experimentally known to increase the prediction accuracy (28 and 29), and is efficient in optimization process (9). Details of our neural network structures are discussed in the next paragraphs.

$$\boldsymbol{C} = -\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{n}^{N}\sum_{j\in C^n}\left[\boldsymbol{y}_j^n \ln \mathbf{Pr}(\boldsymbol{j}|\,\boldsymbol{C}^n) + (\mathbf{1}-\boldsymbol{y}_j^n)\ln(\mathbf{1}-\mathbf{Pr}(\boldsymbol{j}|\,\boldsymbol{C}^n))\right] \qquad (8)$$

For the purpose of comparisons MLP with our model. our research first examines a single hidden layer MLP with a sigmoid function (MLP-S). This structure is generally used in the past MLP research. We refer to Hensher and Ton's travel choice model (13) for MLP-S. We used the same 30-perceptrons and 1,000 epochs for MLP-S.

Secondly, we test multiple hidden layers structures to understand the importance of deep learning techniques and to come up with the structure of DNN model. Note that a multiple hidden layers structure also can be called a deep network structure but we do not call it a DL since it does not implement any deep learning techniques. We constructed this multi-layer 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. This is to analyze the advantage of using the ReLU activation function. Each hidden layer has 200 neurons. 2000-epochs are used. This will be explained in the next paragraph.

The next model we developed is a deep neural network (DNN) that uses an RRRS structure, based on the MLP results that showed the RRRS hidden layer structure to outperform the SSSS structure. This DNN model is called DNN(RRRS). Finally, we constructed our proposed model (DNN-A) with AAF (Availability of Alternatives Function) as in Eq (6), and as shown above in Figure 3. A 2000-epoch training scheme is selected after we observed that too many epochs induce the over-fitting problem, as indicated in Figure 4 for DNN-A. At a certain epoch number, the log likelihood of the test set becomes maximum, whereas that of training set keeps increasing as the iterations keeps processing, which is over-fitting.

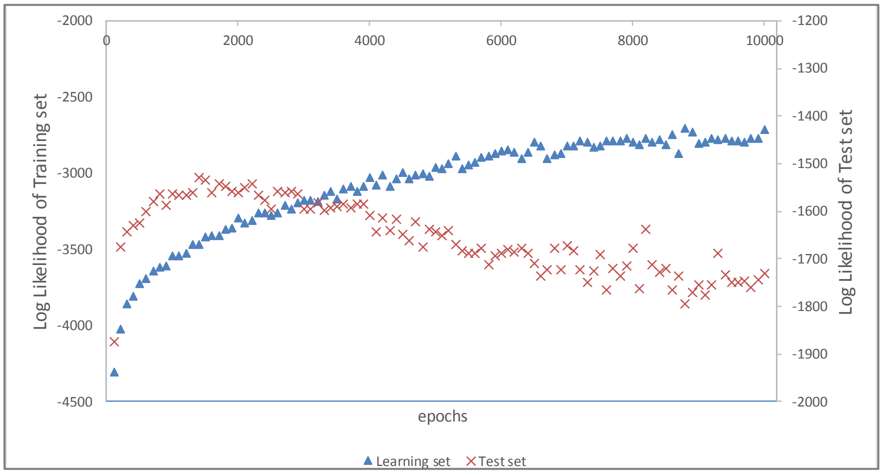
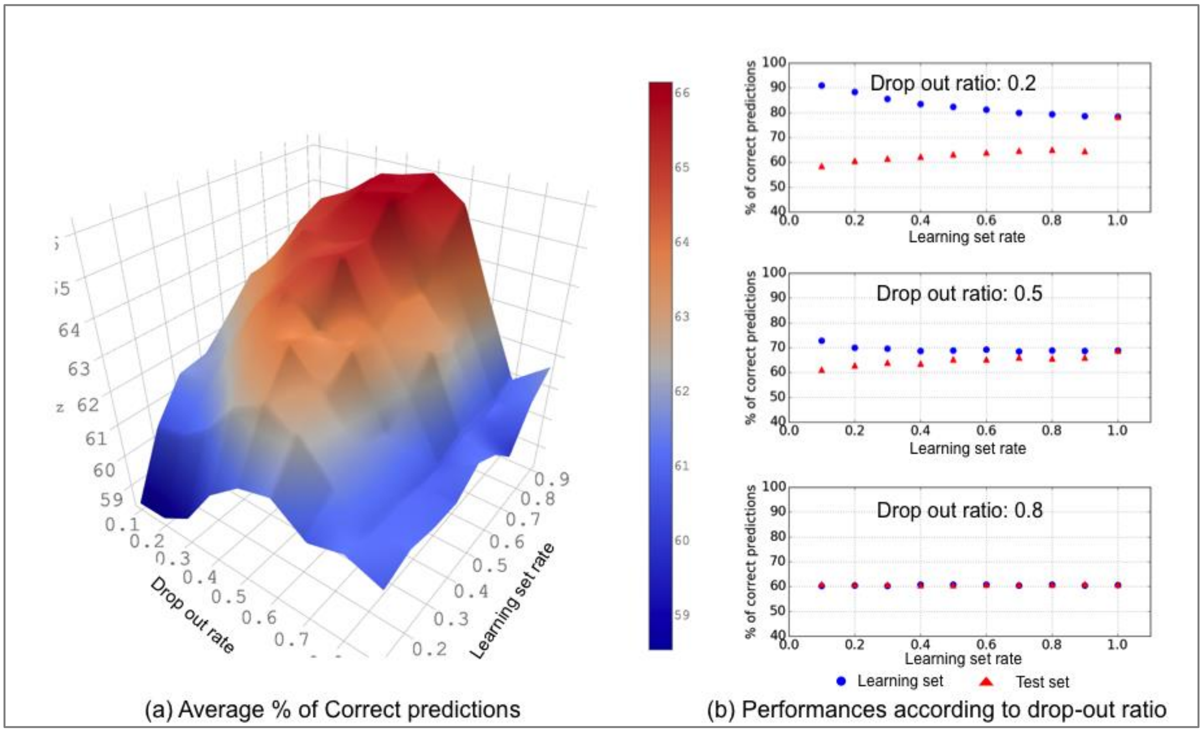


FIGURE 4 Log-likelihood comparison between training and test by the number of epoch

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).

A certain ratio of neurons in each network is randomly dropped out to prevent overfitting. This ratio, and the learning set ratio are important factors in the model's performance. To determine the appropriate dropout ratio and learning set ratio, a sensitivity analysis of the network's performance is conducted with the results shown in Figure 5 (a). The percentage of correct prediction is calculated by applying the trained model to 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 0.5, the model is very stable, with a learning set ratio of 0.5 (Figure 5 (b)), which means that a small size of learning data set could also explain overall observations without an over-fitting issue. Finally, a deep neural network with a dropout ratio of 0.5 and a learning set ratio of 0.7 is implemented to the dataset and the results are compared with previously discussed models in the next section.



(a) Average % of Correct predictions

(b) Performances according to drop-out ratio

FIGURE 5 Sensitivity analysis with drop out ration and learning set rate

To estimate the weighted matrix of each layer, we applied a stochastic gradient descent algorithm (SGD), which is known to be effective for parallelizing over multiple processors such as GPUs. TensorFlow (8) with python 2.7 is used to analyze both MLP and DNN by fully utilizing a Parallel GPU computing environment. GPU has 2048 processors boosting the optimization speeds. The tests are conducted under Ubuntu 14.04 with Intel I5 3.30 GHz quad core CPU and 16GB memory. Average training time of 0.7 learning set rate with this condition is 46.24 seconds. When GPU is not employed, training the model takes 451.13 seconds, on average.

5. Comparison of DNN and Against Other Mode Choice Models

NL, CNL, MLPs, and DNN are compared to DNN-A, the proposed model. Same explanatory variables (input data) are imported to all models to set a fair testing environment. Randomly divided data sets are used to reduce the chance for a biased sample, which could cause spurious results. Furthermore, 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, we will randomly divide the observations into 10 training and test sets. The same common replica sets are used to each model's estimation and validation.

In mode choice research, Log-likelihood, rate of correct predictions, and RMSE are commonly used to evaluate model performance. The model's overall performance is measured with the Log-likelihood and the rate of correct prediction. Log likelihood (Eq 10) indicates how probabilistically well fitted the model is, to the data. The Log-likelihood is a log-sum of the selected alternative's probability. This index is usually applied in a logit model family.

$$\mathbf{L} = \prod_{n=1}^{N}\prod_{\forall m\in C^n} \mathbf{Pr}(m|C^n)^{y_m^n} \tag{9}$$

$$\mathbf{LL} = \mathbf{lnL} = \sum_{n}^{N}\sum_{\forall m\in C^n} y_m^n \mathbf{Pr}(m|C^n) \tag{10}$$

In contrasts, the rate of correct prediction, a common index in the machine learning area, is somewhat different, in that it measures how the model correctly predicts each individual's choice. This approach assumes that an individual selects an alternative having maximum probability among alternatives, as shown in Eq (11,12).

$$\%\ of\ corrected\ prediction = \frac{1}{N}\sum_{n}^{N}\sum_{j\in C_n} y_j^n \tag{11}$$

$$\begin{Bmatrix} y_j^n = 1 \ \ if\ j = argmax(Pr(j|C^n)) \\ y_j^n = 0 \ \ if\ j \neq argmax(Pr(j|C^n)) \end{Bmatrix} \tag{12}$$

In addition, estimating market share ratio or mode share ratio is more interesting for policy makers and planners (30). The market share ratio of N individuals, the aggregate proportion choosing *Cj*, can also be calculated in two aspects: One is the average value of the choice probability for alternative *j* of the individuals (Eq 13). The other is the average of the predictions for alternative *j* across the individuals (Eq 14).

$$Market\ share\ (Prob\ sum) of\ j = MS_j(Prob\ sum) = \frac{1}{N}\sum_{n}^{N} Pr(j|C^n) \tag{13}$$

$$Market\ share\ (arg\ Max) = MS_j(arg\ Max) = \frac{1}{N}\sum_{n}^{N} y_j^n \tag{14}$$

Root mean square error (RMSE) is also used to measure how the models explain the market share ratio. Note that we evaluate each model by using multiple replica sets in Eq (15).

$$RMSE = \frac{1}{R}\frac{1}{C}\sqrt{\sum_{r}^{R}\sum_{c}^{C}\left(\widehat{MS}_c^r - MS_c^r\right)^2} \tag{15}$$

Overall, the proposed DNN-A predicts both aggregate behavior and individuals' behavior well, as indicated in Table 1. In terms of Log-likelihood (LL), DNN-A has the highest value. LL of test set is -1557.94. When we consider that even cross-nested logit (CNL) improves the LL only to -1567.50 from the -1570.99 shown by nested logit (NL), the result of DNN-A is outstanding. Furthermore, DNN-A's correct prediction rate is 66.10, which is higher than other models, meaning that the DNN-A forecasts individuals' choice more accurately as well. 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. When the number of layers increases with sigmoid functions (MLP-SSS), the model severely suffers from the over-fitting problem, the LL and the percentage of correct prediction on the Learning set are noticeably high, but the results on the test data are poor. MLP-RRRS does not have the over-fitting problem but the model's performance is still worse. These results demonstrate strongly that designing deep structures without deep learning techniques is undesirable. It is also clear that applying the newly-developed AAF (Availability of Alternatives Function) has stepped-up the DNN's performance even further, since the LL on the test dataset increases from -1783.00 to -1576.00 and the correct prediction percentage also rises to 65.57 from 61.83.

Table 1. Comparative results of predictive potential between DNN-A with other models

	Log likelihood		% of Correct predictions	
Model	Learning	Test	Learning	Test
Nested Logit	-3658.88 ± 24.85	-1570.99 ± 25.64	63.95 ± 0.27	63.72 ± 0.85
CNL	-3642.42 ± 29.01	-1567.50 ± 23.07	63.77 ± 0.39	63.16 ± 1.07
MLP(S)	-3998.87 ± 164.56	-1760.30 ± 66.91	62.53 ± 1.77	61.77 ± 2.01
MLP(SSSS)	-528.86 ± 57.23	-4924.52 ± 293.07	94.91 ± 0.67	59.70 ± 1.42
MLP(RRRS)	-4192.41 ± 209.14	-1783.00 ± 73.65	61.64 ± 1.19	61.83 ± 1.70
DNN(RRRS)	-3370.12 ± 45.64	-1576.00 ± 47.67	68.76 ± 0.73	65.57 ± 1.23
DNN-A(RRRS)	-3326.37 ± 28.06	-1557.94 ± 14.11	68.57 ± 0.51	66.10 ± 0.93

The proposed model has also reproduced the aggregate market share ratio well. The detailed predictive power for each travel mode is indicated in Table 2 and 3. When each mode's market share ratio is aggregated from the individuals' mode choice probabilities, all models are likely to predict the actual shares. But when we calculate the market share ratio based on the predicted choices, selecting the choice giving maximum probability among alternatives, the RUMs under-estimate the market share ratio of Rail. This is because RUMs tend to underscore the alternative that has low frequencies of the observation. In the similar way, RUMs overestimates the ratio of SM. This implies that RUMs might be problematic when RUMs predict each individual's choice using the maximum probability alternative. This problem is recognized in the past, as Shalaby (31) 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. So aggregated predicted mode choice ratio could be a good index to measure the goodness-of-fit. The DNN results show that underestimating or overestimating of specific alternatives are significantly reduced.

Table 2. Comparison of the aggregate (market share) predictive potential between DNN-A with other models

	Aggregated Probability sum of each mode			Aggregated Predicted mode ratio		
Model	Car	Rail	SM	Car	Rail	SM
Nested Logit	26.66±0.45	12.91±0.24	60.44±0.46	9.09±1.89	3.08±0.40	87.83±1.88
CNL	26.54±0.43	13.00±0.21	60.45±0.47	5.32±2.29	3.96±0.53	90.72±2.07
MLP(S)	25.59±0.61	13.81±0.61	60.90±0.59	17.69±4.36	5.88±4.01	76.43±4.59
MLP(SSSS)	25.26±0.66	13.01±0.61	61.72±0.45	24.81±0.77	11.71±1.34	63.47±1.76
MLP(RRRS)	25.70±1.01	13.77±0.86	60.53±1.46	13.87±6.01	5.87±3.36	80.26±8.29
DNN(RRRS)	25.06±0.61	13.87±0.29	61.06±0.64	19.14±1.36	9.69±1.17	71.17±1.79
DNN(RRRS-A)	25.21±0.24	13.88±0.47	60.92±0.55	18.87±0.83	10.15±0.86	70.97±1.11
Observations	Car : 25.2 Rail : 13.8 SM: 61.0					

The RMSEs of DNN-A are lowest at 0.425 and 7.18 respectively, which are lower than for the other models, meaning that DNN-A closely forecasts the market share ratios of all travel modes. When we calculate Rail's choice ratio based on individuals' predicted choices, the RUMs significantly underestimate. The RMSEs of predicted choice ratios are highest at 19.56 for NL and 21.48 for CNL. This is because RUMs tend to fit to the alternative having highest frequency. MLPs also have this limitation. The two DNN models overcome this problem also via deep learning techniques such as deep hidden layers, initialization, stochastic gradient descent method, and dropouts.

Table 3. Comparison of the aggregate (market share) predictive potential between DNN-A with other models (RMSE)

Model	Aggregated Probability sum of each mode	Aggregated Predicted mode ratio
Nested Logit	1.099	19.156
CNL	1.025	21.478
MLP(S)	0.578	11.658
MLP(SSSS)	0.830	2.278
MLP(RRRS)	1.149	14.908
DNN(RRRS)	0.518	7.369
DNN-A(RRRS)	0.425	7.188

6. CONCLUSION

This paper proposes a deep neural network model for travel choice prediction. Deep learning is positively recognized in various fields of study for its ability to represent any form of functions and its strong prediction potential. Test implementation of deep neural networks shows that it outperforms the previous discrete choice models, such as nested logit and cross-nested logit models, as well as the simpler multi-layer 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 in 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, and this has caused poor performance, which we correct in this paper. 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 learning set ratio. By examining the characteristics of each such detail, our research finds proper 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 the proposed model with other models. Experiments prove the superiority of our model. In terms of overall predictive potential, the proposed model has the highest Log-likelihood and the percent of correct predictions among models. Both versions of deep neural network that we studied showed high predictive potential, but the version that incorporates a function for availability of alternatives shows better model accuracy. Market share (Mode choice ratio) is also predicted well by the proposed model and its RMSEs are the lowest among the models against which we compared it. Market share ratio is estimated in both average probabilities of each mode and predicted choices' ratio. Whereas random utility models suffer from under-estimation for the relatively less used modes (Rail in our study case) and over-estimation for the most preferred mode (Swiss Metro), the deep neural networks estimate the mode choice closer to the observed mode choice.

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 cause biases since people are generally generous to it in such a survey. Thus, future studies with our models will need to 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 any statistical interpretations such as odds ratio, elasticity and sensitivity. As there is no easy counter-argument, the authors consider that the trade-off is on the accuracy of the model, which our study amply demonstrates. Alternative suggestions can be considered, such as in Mohammadian and Miller (14) on a simulation approach to assess the model output changes by changing input values. We leave such further work also for the future.

REFERENCES

1. McFadden, D. Modelling the Choice of Residential Location. *Transportation Research Record*, (673), 1978, pp. 72-77.
2. Small, K., A Discrete Choice Model for Ordered Alternatives. *Econometrica*, 55(2), 1987, pp. 409–424.
3. Vovsha, P., Cross-Nested Logit Model: An Application to Mode Choice in the Tel-AvivMetropolitan Area. *Transportation Research Record*, 1607, 1997, pp. 6–15.
4. Ben-Akiva, M. and M. Bierlaire., Discrete Choice Methods and their Applications to Short-Term Travel Decisions. In R. Hall (ed.), *Handbook of Transportation Science*, Kluwer, 1999, pp. 5–34.
5. Brownstone, David. Discrete choice modeling for transportation. *University of California Transportation Center*, 2001.
6. Bierlaire, Michel, Kay Axhausen, and Georg Abay. "The acceptance of modal innovation: The case of Swissmetro." *In Proceedings of the 1st Swiss Transportation Research Conference*. 2001.
7. Bierlaire, Michel. A theoretical analysis of the cross-nested logit model.*Annals of operations research*, 144, no. 1, 2006, pp. 287-300.
8. A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Man, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vi, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, arXiv preprint arXiv:1603.04467., 2015.
9. L. Deng and D. Yu. Deep Learning: Methods and Applications. Found. *Trends Signal Process*., vol. 7, no. 3–4, 2013, pp. 197–387.
10. Angelova, Anelia, Alex Krizhevsky, and Vincent Vanhoucke. Pedestrian detection with a large-field-of-view deep network. *In 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE*, 2015, pp. 704-711.
11. Najafabadi, Maryam M., Flavio Villanustre, Taghi M. Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data* 2, no. 1,2015.
12. De Carvalho, M. C. M., M. S. Dougherty, A. S. Fowkes, and M. R. Wardman. Forecasting travel demand: a comparison of logit and artificial neural network methods. *Journal of the Operational Research Society* 49, no. 7,1998, pp. 717-722.
13. D. a Hensher and T. T. Ton, A Comparison of the Predictive Potential of Artificial Neural Networks and Nested Logit Models for Commuter Mode Choice. *Transportation Research Part E*, vol. 36, no. 2000, pp. 155–172.
14. Mohammadian, Abolfazl, and Eric Miller. Nested logit models and artificial neural networks for predicting household automobile choices: comparison of performance. *Transportation Research Record: Journal of the Transportation Research Board* 1807, 2002, pp. 92-100.
15. Cantarella, Giulio Erberto, and Stefano de Luca. Multilayer feedforward networks for transportation mode choice analysis: An analysis and a comparison with random utility models. *Transportation Research Part C: Emerging Technologies* 13, no. 2, 2005, pp. 121-155.

16. Zhang, Yunlong, and Yuanchang Xie. Travel mode choice modeling with support vector machines. *Transportation Research Record: Journal of the Transportation Research Board* 2076, 2008, pp.141-150.
17. Omrani, Hichem, Omar Charif, Philippe Gerber, Anjali Awasthi, and Philippe Trigano. Prediction of individual travel mode with evidential neural network model. Transportation Research Record: Journal of the Transportation Research Board 2399, 2013, pp. 1-8.
18. Omrani, Hichem. Predicting Travel Mode of Individuals by Machine Learning. *Transportation Research Procedia* 10, 2015, pp. 840-849.
19. Abay, G. Nachfrageabschätzung Swissmetro: Eine stated-preference Analyse, Berichte des Nationalen Forschungsprogrammes 41 "Verkehr und Umwelt", F1, EDMZ, Bern., 1999.
20. Bierlaire, Michel. "BIOGEME: a free package for the estimation of discrete choice models." In Swiss Transport Research Conference, no. TRANSP-OR-CONF-2006-048. 2003.
21. Hess, Stephane, Michel Bierlaire, and John W. Polak. Capturing correlation and taste heterogeneity with mixed GEV models. *In Applications of simulation methods in environmental and resource economics, Springer Netherlands*, 2005, pp. 55-75.
22. S. Hess, Confounding between taste heterogeneity and error structure in discrete choice models, *In Proceedings of the European Transport Conference, Strasbourg, France* vol. 41, 2006, pp. 1–26.
23. Nair, V., & Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning*, (3), 2010, pp. 807–814. http://doi.org/10.1.1.165.6419
24 Xu, B., Wang, N., Chen, T., & Li, M. Empirical Evaluation of Rectified Activations in Convolution Network. *ICML Deep Learning Workshop*, 2015, pp. 1–5.
25 Hinton, G. E., Osindero, S., & Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 2006, pp.1527–54. http://doi.org/10.1162/neco.2006.18.7.1527
26. Glorot, Xavier, and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *In Aistats*, vol. 9, 2010, pp. 249-256.
27. Silva, Luís M., J. Marques de Sá, and Luís A. Alexandre. Data classification with multilayer perceptrons using a generalized error function. *Neural Networks* 21, no. 9, 2008, pp. 1302-1310.
28. Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. *In Proc. ICML*, vol. 30, no. 1. 2013.
29. Dahl, George E., Tara N. Sainath, and Geoffrey E. Hinton. Improving deep neural networks for LVCSR using rectified linear units and dropout. *In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE*, 2013. pp. 8609-8613.
30. de Dios Ortúzar, Juan, and Luis G. Willumsen. Modelling transport. *New Jersey: Wiley*, 1994.
31. Shalaby, Amer S., "Investigating the Role of Relative Level-of-Service Characteristics in Explaining Mode *Split* for the Work Trip", *Transportation Planning and Technology*, Vol. 22, 1998, pp. 125-148.