Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 184 (2021) 532–540

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Ambient Systems, Networks and Technologies (ANT)
March 23 - 26, 2021, Warsaw, Poland

Applying transfer learning and various ANN architectures to predict transportation mode choice in Amsterdam

Ruurd Buijs[a], Thomas Koch[b,*], Elenna Dugundji[a,b]

[a] *Vrije Universiteit Amsterdam, De Boelelaan 1111, Amsterdam 1081 HV, The Netherlands*
[b] *Centrum Wiskunde en Informatica, Science Park 123, Amsterdam 1098 XG, The Netherlands*

Abstract

For long, statistical models have been used for transportation mode choice analysis, due to their ability to extract economic information from the model parameters. Recently, the application of Artificial Neural Nets to predict transportation mode choice is gaining ground, partly due to efforts that have led to an improved interpretability of this class of models. In this development, various innovations have been suggested concerning Neural Net architecture and hyperparameter tuning. Building on this, this paper investigates 3 similar Neural Net architectures to be applied to data from an Amsterdam case study. This data has been collected in 3 waves. Between the first and second collection period, the public transportation network in Amsterdam changed. A transfer learning approach is suggested to improve models that were trained on a single wave of data. Based on the test loss of the models from the transfer learning experiments, we conclude that this is a promising technique to use in this context, since it has shown to improve model performance.

Keywords: Transportation mode choice; neural nets; public transportation network change; travel behaviour; transfer learning;

Within the field of study of transportation mode choice analysis, the dominant approach has long been to apply conventional statistical logit models to infer economic information and predict the mode choice of individuals. Recently, there has been a surge of exploring how machine learning techniques an contribute to this field of study. This research has mainly revolved around Artificial Neural Networks (ANN) or Deep Neural Networks (DNN). This study will investigate several Neural Net architectures, and will investigate transfer learning applied to data concerning a case study revolving tracking data of Amsterdam trips that has been collected in the light of a major transportation network change in the Amsterdam region. In this paper, we will first give an overview of the literature on this topic. Next, the data set is introduced shortly, after which the applied methods regarding the Neural Net architectures and transfer learning experiments are discussed. Results and findings will be presented and discussed, after which some suggestions are given for further research directions.

[*] Corresponding author. Tel.: +31-20-592-4132
E-mail address: koch@cwi.nl

10.1016/j.procs.2021.03.067

1. Literature review

Over the past decades, transportation mode choice problems have been studied extensively. Various models have been developed to predict mode choice for individuals, as well as to study the market share of different available modes. In the 1970s, econometric models for transportation mode choice analysis with multiple (> 2) alternatives have been developed in order to infer information about market share and relative importance of different socio-economic factors relevant to transportation mode choice, based on the revealed or stated preference of individuals in a population sample. Relying on the assumption of rational agents and the concept of random utility maximization, McFadden [9] proposed a conditional logit model. A version of this model has later been applied to San Francisco survey data [8]. These early econometric models mainly are tailored to the use of revealed preference (RP) data, i.e. using observations of travel behaviour exerted by individuals in a population to identify what mode is chosen, making certain assumptions regarding the characteristics of alternatives available in real-time.

Building on these foundations (An extensive overview of conventional statistical techniques that can be applied to the revealed preference type of problem is presented by Ben Akiva and Lerman [2]), models were presented that take into account stated preference (SP) as well. In these models, part of the data is gathered in a population survey reflecting on hypothetical scenarios, with alternatives that may not hold a one-to-one correspondence to real-life available transportation modes and their respective characteristics. This approach was put forward by Morikawa [10]. In this work, a combination of RP and SP was used to obtain an enriched model for mode choice. The model was then applied to survey data regarding transportation mode choice in the Netherlands. Hensher [5] gives a good overview of developments regarding the use of SP data combined with RP data.

In the late 1990s, some studies were carried out in the broader field of market share research and consumer choice using Artificial Neural Networks (ANNs) ([1], [15]). A hybrid approach has also been considered [3]. Despite the advantages in predictive power of the Neural Net models, a main drawback of this class of models remained the lack of interpretability. Especially when one is interested in the importance of factors driving mode choice, interpretability is a desirable, and maybe imperative property for the model that is used.

Later, Vythoulkas and Koutsopoulos [11] have applied a model based on fuzzy set theory, that adopted a neural net-like structure on a transportation mode choice problem. The case at hand was the same as the one discussed by Morikawa in his work [10]. More recently, efforts have been made to solve the problem regarding ANNs or Deep Neural Networks (DNNs) being hard to interpret. Wang and Zhao have proposed methods to extract quantities equivalent to those commonly obtained when using the conventional RUM-logit model [14]. Alongside these findings, an increasing number of studies is aimed at various architectural innovations in DNN models that can be applied to the field of transportation mode choice. Among these are interesting architectural findings by Wang, Mo and Zhao [12] allowing only links between nodes corresponding to the same alternative in the first layers. In this way, a means of regularization is applied to the DNN, that based on its architecture can be interpreted as having a specific utility function for each available alternative. Other findings focus on the combination of stated and revealed preference data, which could also be achieved by using RUM-logit models ([10],[5]). For this, a Multitask Learning Deep Neural Network (MTLDNN) is used [13]. Lastly, since analysis in transportation mode choice usually involve categorical variables related to socio-demographic data of the population sample, the suggestion to model these in using entity embedding [7] is a relevant and important contribution to the recent literature.

2. Case Study and data collection and generation

The data set that is used from this study is based on actual trip data from the area of Amsterdam, the Netherlands, collected from participating users in three separate waves. The interesting fact about these waves is that between the first and second wave of data collection, a new metro line opened, and a restructuring of the public transportation network in and around Amsterdam took place in which the new metro line forms a spine. The collected trip data (regarding users' revealed preference) was used to generate choice sets by computing the fastest possible route from the origin to the destination by using different modes. These generated choice sets have been used as input to the discussed models to make predictions. The actual recorded trip data has not been used for this, in order to prevent introducing unwanted bias. This means that the attributes in the choice set for the mode that has been chosen do not necessarily coincide with the recorded trip data. To account for this, identifying for example round trips and indirect

Table 1. Relevant alternative-specific variables that were collected and/or generated for the choice set. The last 4 variables are only relevant to the alternative ’car’.

groupid	A unique identifier referring to a single trip from origin to destination; generated trips corresponding to this trip have the same groupid and also refer to a specific person
strata	Categorical variable that indicates the transportation mode of a (generated or actual) trip: 1 for walking 2 for traveling by car 3 for traveling by bicycle 4 for traveling by public transportation with use of metro and train 5 for traveling by public transportation with use of train; without use of metro 6 for traveling by public transportation with use of metro; without use of train 7 for traveling by public transportation without use of metro and without use of train
transfers	Number of transfers on the public transportation part of this trip
duration	Total duration of trip
bicycle duration	Total duration of trip that is traversed on bicycle
car duration	Total duration of trip that is traversed by car
walk duration	Total duration of trip that is traversed on foot
waiting_time	Total time spent waiting on public transportation if applicable
duration_of_stay	Total time spent at the destination (estimated by using time between different trip records)
parking_cost_0	For trips with destination in the municipality of Amsterdam: Parking tariff (converted to hourly rate) at the expected time of arrival at destination.
parking_cost_60	For trips with destination in the municipality of Amsterdam: Parking tariff (converted to hourly rate) that applies 60 minutes after the expected time of arrival at destination.
parking_cost_180	For trips with destination in the municipality of Amsterdam: Parking tariff (converted to hourly rate) that applies 180 minutes after the expected time of arrival at destination.

trips, which would show a huge difference in recorded and generated data. As these types of trips do not give reliable information about the preferred mode, they have been discarded. Alongside the generated variables concerning the alternatives at hand, socio-demogrphic user data was also available for each of the users having trip records.

The variables that were generated or recorded are displayed in tables 1 and 2. For more background information about the context of the data and the generation of the features for the alternative modes, please see [4]. We based the features regarding parking tariffs on publicly available data from the public governmental body regulating road traffic in the Netherlands (Rijksdienst voor het Wegverkeer).[1]

3. Methodology

3.1. Partitioning data

As mentioned in the description of the data, tracking data from users have been collected in 3 waves, 1 wave being collected before the introduction of the North-South metro line, 1 wave shortly after, and a final wave about 1 year after the introduction. For this study, the 3 periods of data collection have been treated to correspond to 3 different, yet similar classification problems. The generated choice sets for the 3 waves of data have thus been treated and

[1] https://opendata.rdw.nl/browse?category=Parkeren

Table 2. Relevant (socio-demographic variables) that were collected for each user in the data set

userid	Identifier referring to a specific person in the data set. Each userid spans one or more trips, and can thus correspond to multiple groupids. Since the mapping is one-to-many, the userid can easily be identified based on the groupid of the trip.
geslacht	Sex of the user (Categorical variable)
aantal auto's in huishouden	Number of cars at the disposal of the user's household (Integer)
gezinscyclus	Provides information about the composition of the user's household (Categorical variable)
leeftijd	Age category of user (Categorical variable)
opleidingsniveau	Highest level of education of the user (Categorical variable)
hoogst voltooide opleiding	Highest level of education of the user (Categorical variable)
netto maandinkomen persoon	Provides income information (Categorical variable)
herkomst 1e generatie	Country of origin of user (Categorical variable)
herkomst 2e generatie	Country of origin of the parents of the user (Categorical variable)

Table 3. Number and percentage of choice sets in each partition for each of the 3 waves

	Wave 1	Wave 2	Wave 3
# of users train set	243 (49.7%)		
# of observations train set	10,672 (50.1%)	9,254 (49.3%)	7,862(48.8%)
# of users validation set	105 (21.5%)		
# of observations validation set	4,281 (20.1%)	3,877 (20.7%)	3,156 (19.6%)
# of users test set	141 (28.8%)		
# of observations test set	6,334 (29.8%)	5,644 (30.1%)	5,079 (31.6%)

partitioned separately. For each of the 3 waves, the data was split into a training, validation and testing partition. To prevent any bias induced by return trips or repetitive behaviour, this split was conducted such that trips corresponding to a single user would all be part of either the train, validation or test set. The sets were sampled according the procedure in algorithm 1. The final distribution of the number of trips over the different partitions of the three waves is given in table 3. All Neural Net models discussed in this paper were trained based on training data for one of the waves

Algorithm 1: Data partitioning

```
while train set of first data wave contains less than 50% of trip records of first data wave do
    draw random user from list of users;
    add trips of the corresponding user to train sets for all waves of data;
    remove user from list of users;
end
while validation set of first data wave contains less than 20% of trip records of first data wave do
    draw random user from list of users;
    add trips of the corresponding user to validation sets for all waves of data;
    remove user from list of users;
end
add trips corresponding to remaining users to test sets for all waves of data;
```

.

3.2. Artificial Neural Nets (ANNs): Background

Artificial Neural Nets (ANNs), and more specifically Deep Neural Nets (DNNs) is a Machine Learning technique that is widely used for simple or complex tasks involving classification or prediction. Originally, the method was developed to capture complex relations and make decisions in a way that resembles the neuronal structure of the human brain. A Neural Net typically consists of multiple layers, each consisting of multiple nodes. Information is passed on from layer to layer in a sequential fashion. Nodes get information from nodes in the previous layer: The outputs from latter nodes are multiplied with trainable weights and then added up. An activation function is applied to the result before the output is being passed through to nodes in the next layer. Some activation functions that may be used are the ReLU (Rectified Linear Unit), sigmoid and tanh functions. The weights of the Neural Net are trained by a backpropagation process, which aims to minimize a certain loss function, a function of the model predictions and the actual values. In this process, weights are updated iteratively by evaluating the gradients of the weights with respect to the loss. Different optimization methods exist that aim to achieve this, of which Adaptive Moment Estimation (Adam) [6] is currently among the most popular alternatives. For most optimization methods, including Adam, a learning rate α needs to be specified, which indicates the step size of the weight updates, and needs to be chosen carefully, since it impacts the speed of the learning as well as the convergence. Another important parameter in the training process is the batch size, indicating how many data entries are passed through the model before the weights are updated.

3.3. Neural network architecture

An important design choice when working with Neural Nets is the architecture of the model, which entails the number of layers in the model, the number of nodes in each layer and how the nodes in the layers are connected. If all nodes are connected to each node in the previous layer, the network is fully connected, meaning it will have the highest complexity. Sometimes, refining the architecture, only allowing for some of these connections, can help the model to become a more simple and accurate representation of reality, and therefore easier to train or to interpret.

As discussed earlier, the input data consists of features that are specific to the available alternatives, denoted x_{ik}, where $i \in \{1, 2, ..., N\}$ is an index used for numbering the instances, and $k \in \{1, 2, ..., 7\}$ indicates the alternative index . Zero padding was used in case an alternative was not available for a certain choice instance. Socio-demographic attributes are available for each user as well, and are denoted z_i, $i \in \{1, 2, ..., N\}$ again being an instance numbering index.

Building on the work of Wang, Mo and Zhao [12], 3 different architectures have been considered: The 'default architecture', consisting of a separate trainable subnet for each alternative k using input x_{ik}, the 'link4' architecture, which starts with a jointly, fully connected subnet for all public transportation alternatives, which later differentiates into individual subnets after the alternative-specific subnets are exposed to the individual specific data (features based on input z_i), and the 'merge4' architecture, which uses $K = 7$ alternative-specific input vectors, but only differentiates between 4 different modes: All public transportation input is jointly fed to a fully connected subnet, leading to a single utility value and choice probability for choosing Public Transportation. The latter hence is a slightly simplified model. All 3 architectures use fully connected subnets for handling the alternative-specific inputs x_{ik} and the decision-maker inputs z_i. The categorical variables in z_i are first fed into an entity embedding layer (as suggested by Ma and Zhang [7]). Below is an overview of the layers that are used in the 3 different architectures:

- An input layer consisting of inputs x_{ik}, $k = 1, 2, ...7$ and z_i.
- An entity embedding layer mapping the categorical features in z_i.
- m_1 layers consisting of nodes with ReLu activation, forming fully connected subnets of width n for the (separate) inputs x_{ik} and z_i.
- m_2 layers consisting of nodes with ReLu activation, forming fully connected subnets of width n for the (separate) inputs x_{ik}, to each of which the features from the subnet corresponding to z_i are added in layer m_1.
- A softmax layer transforming the output of layer $m_1 + m_2$, which can be interpreted as utilities u_{ik} of the available modes, to estimated choice probabilities $P(y_i = k)$ for the alternative modes. This transformation, given utilities u_{ik} is computed by the softmax function, corrected for the availability of alternatives. Let A_i denote the set of available alternatives in choice instance i, and let y_i denote the chosen alternative for instance i . Then $P(y_i = k) = \frac{e^{u_{ik}}}{\sum_{j \in A_i} e^{u_{ij}}}$ if $k \in A_i$ and $P(y_i = k) = 0$ otherwise.

A graphical representation of the different model architectures that were used is shown in figure 1.

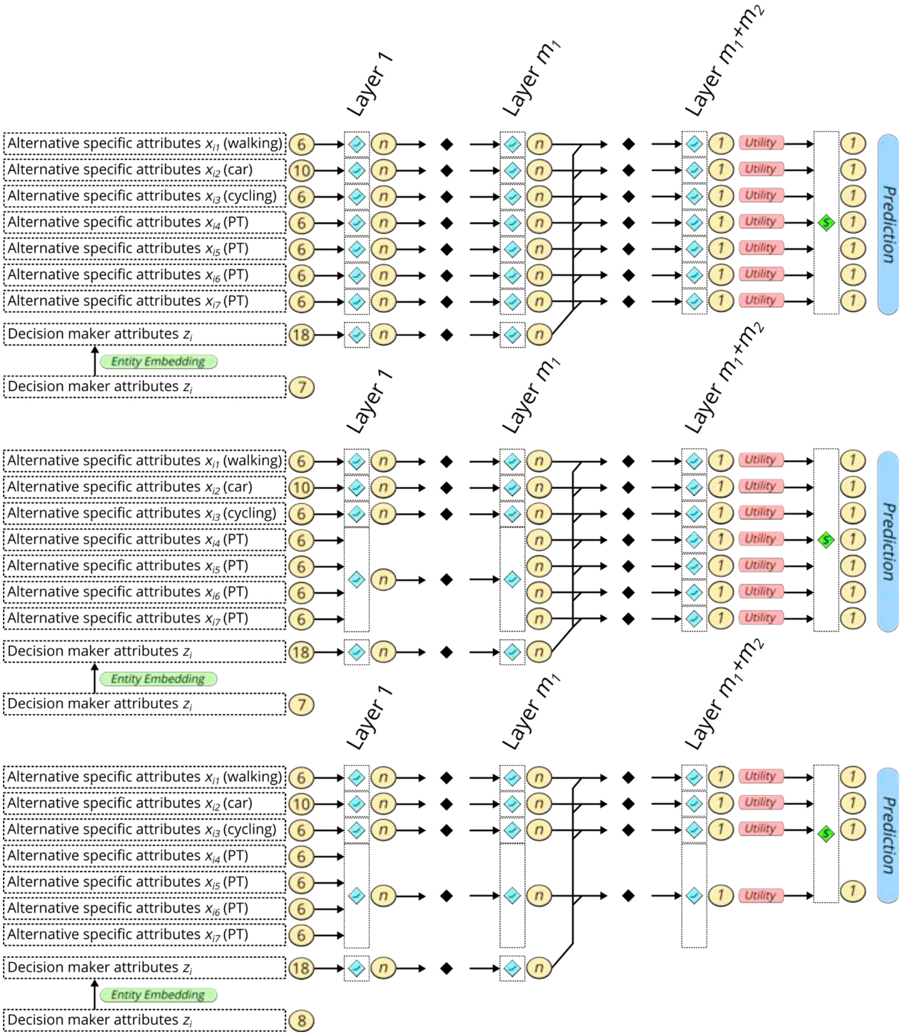
Layer 1
Layer m_1
Layer m_1+m_2
Alternative specific attributes x_{i1} (walking)
6
Alternative specific attributes x_{i2} (car)
10
Alternative specific attributes x_{i3} (cycling)
6
Alternative specific attributes x_{i4} (PT)
6
Alternative specific attributes x_{i5} (PT)
6
Alternative specific attributes x_{i6} (PT)
6
Alternative specific attributes x_{i7} (PT)
6
Decision maker attributes z_i
18
Entity Embedding
Decision maker attributes z_i
7
n
1
Utility
S
Prediction
Layer 1
Layer m_1
Layer m_1+m_2
Alternative specific attributes x_{i1} (walking)
6
Alternative specific attributes x_{i2} (car)
10
Alternative specific attributes x_{i3} (cycling)
6
Alternative specific attributes x_{i4} (PT)
6
Alternative specific attributes x_{i5} (PT)
6
Alternative specific attributes x_{i6} (PT)
6
Alternative specific attributes x_{i7} (PT)
6
Decision maker attributes z_i
18
Entity Embedding
Decision maker attributes z_i
7
n
1
Utility
S
Prediction
Layer 1
Layer m_1
Layer m_1+m_2
Alternative specific attributes x_{i1} (walking)
6
Alternative specific attributes x_{i2} (car)
10
Alternative specific attributes x_{i3} (cycling)
6
Alternative specific attributes x_{i4} (PT)
6
Alternative specific attributes x_{i5} (PT)
6
Alternative specific attributes x_{i6} (PT)
6
Alternative specific attributes x_{i7} (PT)
6
Decision maker attributes z_i
18
Entity Embedding
Decision maker attributes z_i
8
n
1
Utility
S
Prediction

Fig. 1. Three different Neural Network architectures: 'Default' architecture (upper), where a different subnet is allocated to each alternative; 'Link4' architecture (center), where the attributes related to all public transport alternatives are first jointly guided through a fully connected subnet, which after the blending with the decision-maker attributes differentiates to a single subnet for each alternative; and 'Merge4' architecture (lower) which essentially treats Public transportation as a single alternative with a single subnet, albeit using the input of each of the available Public transportation alternatives. m_1, m_2 and n are tunable hyperparameters relevant to the network architecture: m_1 and m_2 relate to the depth of the network, whereas n relates to the width of the individual subnets. The total number of nodes in each subnet is shown by layer in the yellow circles.

3.4. Hyperparameter selection

Hyperparameter tuning is an important part of the model training process in machine learning, and especially when training a Neural Net. Choosing the right hyperparameters will likely have a positive effect on the performance of the model(s) that will be obtained after the process of training and evaluation is finished. The relevant hyperparameters that can be varied when training a model in this context, are shown in the leftmost column of table 4. After some prior test runs, the hyperparameter space from which the hyperparameters for training the models are randomly sampled, has been narrowed down to a limited number of reasonable options for each of the hyperparameters. The final hyperparameter space for assessing the model architectures is shown in the rightmost column of table 4. Sparse categorical crossentropy has been used as loss function, as is appropriate to use when classifying labeled data using a Neural Network. The hyperparameters were tuned based on the loss on the validation set. Training was stopped after the loss on the validation set has not improved for 5 consecutive epochs (early stopping criterion), as to prevent overfitting.

Table 4. Variable and fixed hyperparameters for training the Neural Net models

Hyperparameter	Values to sample from
Subnet width n	$\{10, 20, 25, 50, 100\}$
Number of layers m_1 before merging alternative-specific and decision maker features	$\{2, 6, 10\}$
Number of layers m_2 after merging alternative-specific and decision maker features	$\{2, 3\}$
Batch size	$\{8, 32, 128\}$
Learning rate α	$\{10^{-5}, 10^{-4}\}$
Activation function used for all layers except for the output layer	*reLU*
Optimization method	*Adam*

3.5. Transfer learning

Since choice sets are available for 3 different data collection periods, where the underlying transportation network changed drastically between the first and second wave of collection, a technique called transfer learning can be applied to improve the performance of the Neural Net. This technique transfers information (in the form of Neural Network weights) learned from a particular data set to a Neural Net that is aimed at solving a similar problem with data that is somewhat different. For an elaborate description and example, see e.g. Yosinski et al., 2014 [16]. In order to apply transfer learning, a network is first trained regularly on a single data wave. Next, a new network is initialized, having initial weights for the first l layers identical to the previously trained network. The weights of the other layers are still initialized randomly. One can now choose to fix the weights for the first l layers, only focusing on training the weights of the final, randomly initialized layers, but a more flexible approach is to also train on the weights of the first layers, allowing for the network to re-calibrate co-adapted weights. In some runs, we fixed (part of) the pre-trained layer weights, and in others, we kept them flexible. 30 models were pre-trained on wave 1 data, of which the best model was used to transfer information about the weights to a Neural Net that should learn relations based on wave 2 data. The hyperparameter configuration used in these experiments is $n = 50$, $m_1 = 2$, $m_2 = 3$, batch size = 32 and $\alpha = 10^{-4}$. In total, we used 6 different experimental settings for transferring information from a pre-trained 5-layer model to a new model. For layers 1, 2, and 3, the weights could be 'random', 'flexible' (using information from a pre-trained model where weights are allowed to change during training) or 'fixed' (using information from a pre-trained model where weights are not allowed to change during training). The different initializations of the layers in each of the experimental settings are shown in table 5.

4. Results

4.1. Model architecture

For assessing the different model architectures, 35 random hyperparameter configurations were sampled from the hyperparameter space given in table 4. For each of these hyperparameter configurations, 3 models were trained: one for each proposed architecture. The models were then given a rank based on the validation loss or validation accuracy. The ranked performance of the models of the three different architecture types are shown in figure 2. Based on these results, it is hard to conclude which architecture suits best, although it seems that 'link4' has the highest probability of attaining a predictive accuracy > 0.62.

4.2. Transfer learning

In order to investigate whether transfer learning could be beneficial to apply in this context, several experiments were run. 30 models were pretrained for both wave 1 and wave 2 data, of which the best (in terms of validation loss) was retained. Next, this information was passed on to 2×30 new neural nets to be trained further on wave 2 data, by initializing and/or fixing some of the weights trained for the layers. The top 5 of these models in terms of validation loss were retained for each experiment. Models that are trained using pretrained weights from a wave 2-model form a benchmark, whereas models using pretrained weights from a wave 1-model represent the setting were information

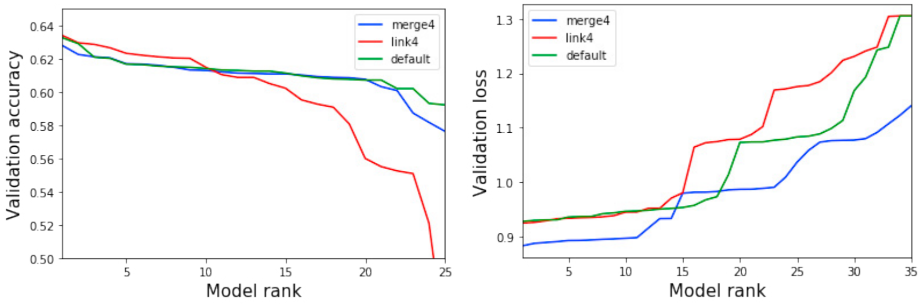

Fig. 2. Models ranked by their performance in terms of validation loss (left plot) and validation accuracy (right plot, only top 20-25 models are shown)

Table 5. Initialization configuration of the layers of the second group of models for each of the 6 transfer learning experiments

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Experiment 1	Fixed	Fixed	Fixed	Random	Random
Experiment 2	Fixed	Fixed	Flexible	Random	Random
Experiment 3	Fixed	Flexible	Flexible	Random	Random
Experiment 4	Flexible	Flexible	Flexible	Random	Random
Experiment 5	Flexible	Flexible	Random	Random	Random
Experiment 6	Flaxible	Random	Random	Random	Random

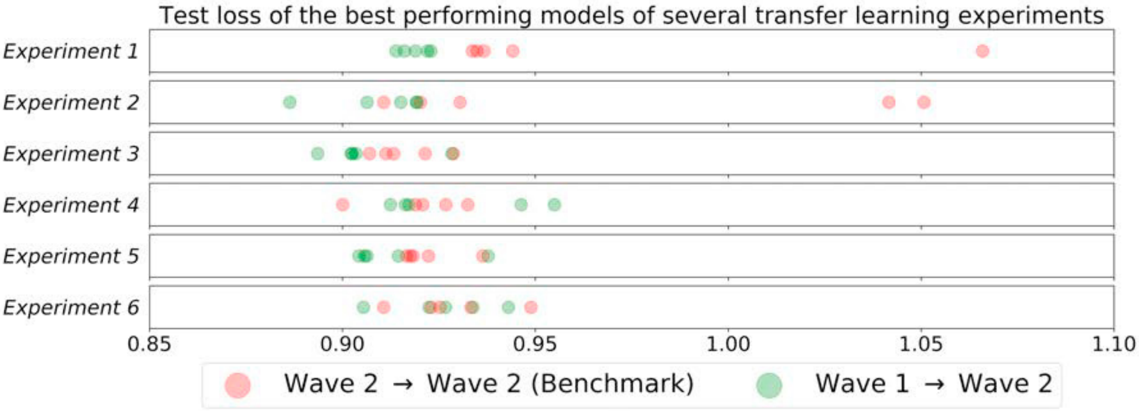

Fig. 3. Test loss of the 2 × 5 best performing models in each experiment. The green dots indicate the models to which the actual described transfer learning procedure was applied (exposed to new data in the second phase of training), whereas the red dots indicate the benchmark model (exposed to the same data in both phases of training)

was transferred. The test loss of the best 5 models from each of the experiments are shown in figure 3. Overall, the models using pretrained weights from a wave 1 model tend to outperform the benchmark models in most experiments. The most promising setting using this hyperparameter configuration appears to be the one where the weights of the first 2 layers are fixed, the weights of the third layer are flexible, and the last 2 layers are initialized randomly.

5. Discussion

Even though the results of the experiments are promising and supposedly indicative of a general finding towards training and tuning models, the trained models may need some more sophistication in order to extract, for example, economic information. The fact that there are few data entries where a public transportation mode is chosen, could be accounted for by allocating a higher weight to these samples when evaluating the loss function. This might lead to a different conclusion regarding the discussed model architectures. In the architecture of the network, alternatives that were not available in the choice set could still be assigned a positive utility. This would not impact the final classification, but it would impact the assigned probabilities and therefore the loss function. The model has been adjusted for this by only taking into account the estimated utilities of available alternatives when applying the softmax function, which is important in order to be able to correctly extract quantities such as elasticities in the future. Some further investigation, e.g. using comparison to a statistical model, may be needed in order to correctly interpret utilities and other quantities that may be extracted from the network. Nevertheless, the results of the transfer learning experiments are promising, and there is no reason to doubt that this technique can also be applied to more sophisticated models.

6. Conclusion and Further research

In conclusion, 3 different model architectures have been suggested to be applied to an interesting mode choice dataset, all of which use subnets for the specific alternatives to impose alternative-specific utilities in the architecture. Further research is needed to make definitive remarks about which architecture suits best in this context, although 'link4' seems to have a slightly higher probability of converging to model that gives a reliable prediction. The transfer learning experiments show promising results, since model performance was found to be improved for models predicting mode choice on the new network, if the models relied on information extracted from models predicting mode choice on the old network. The methods described could be assessed on more sophisticated models, that are suitable for economic interpretation by incorporating a weighing factor to prioritize samples of modes that are chosen less often during training to create a more balanced model. A more in-depth analysis of the transfer learning procedure could be carried out by applying the suggested procedure to models with various (sampled) hyperparameters, instead of using only a single fixed setting.

References

[1] Agrawal, D., Schorling, C., 1996. Market share forecasting: An empirical comparison of artificial neural networks and multinomial logit model. Journal of Retailing 72, 383–408.

[2] Ben-Akiva & Lerman, S., 1985. Discrete choice analysis: Theory and application to travel demand.

[3] Bentz, Y., Merunka, D., 2000. Neural networks and the multinomial logit for brand choice modelling: a hybrid approach. Journal of Forecasting 19, 177–200.

[4] Buijs, R., Koch, T., Dugundji, E., 2020. Using neural nets to predict transportation mode choice: An amsterdam case study. Procedia Computer Science 170, 115–122.

[5] Hensher, D.A., 1994. Stated preference analysis of travel choices: the state of practice. Transportation 21, 107–133.

[6] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

[7] Ma, Y., Zhang, Z., 2020. Travel mode choice prediction using deep neural networks with entity embeddings. IEEE Access 8, 64959–64970.

[8] McFadden, D., 1974. The measurement of urban travel demand. Journal of public economics 3, 303–328.

[9] McFadden, D., et al., 1973. Conditional logit analysis of qualitative choice behavior .

[10] Morikawa, T., 1989. Incorporating stated preference data in travel demand analysis. Ph.D. thesis. Massachusetts Institute of Technology.

[11] Vythoulkas, P.C., Koutsopoulos, H.N., 2003. Modeling discrete choice behavior using concepts from fuzzy set theory, approximate reasoning and neural networks. Transportation Research Part C: Emerging Technologies 11, 51–73.

[12] Wang, S., Mo, B., Zhao, J., 2020a. Deep neural networks for choice analysis: Architecture design with alternative-specific utility functions. Transportation Research Part C: Emerging Technologies 112, 234–251.

[13] Wang, S., Wang, Q., Zhao, J., 2020b. Multitask learning deep neural networks to combine revealed and stated preference data. Journal of choice modelling 37, 100236.

[14] Wang, S., Zhao, J., 2019. An Empirical Study of Using Deep Neural Network to Analyze Travel Mode Choice with Interpretable Economic Information. Technical Report.

[15] West, P.M., Brockett, P.L., Golden, L.L., 1997. A comparative analysis of neural networks and statistical methods for predicting consumer choice. Marketing Science 16, 370–391.

[16] Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks?, in: Advances in neural information processing systems, pp. 3320–3328.