Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

ResLogit: A residual neural network logit model for data-driven choice modelling

Melvin Wong [a,*], Bilal Farooq [b]

[a] École Polytechnique Fédérale de Lausanne (EPFL), School of Architecture, Civil and Environmental Engineering (ENAC), Transport and Mobility Laboratory, Switzerland

[b] Ryerson University, Laboratory of Innovations in Transportation (LiTrans), Canada

ARTICLE INFO

Keywords:
Residual logit
Deep learning
Data-driven discrete choice modelling
Machine learning
Non-linear utility

ABSTRACT

This paper presents a novel deep learning-based travel behaviour choice model. Our proposed Residual Logit (ResLogit) model formulation seamlessly integrates a Deep Neural Network (DNN) architecture into a multinomial logit model. Recently, DNN models such as the Multi-layer Perceptron (MLP) and the Recurrent Neural Network (RNN) have shown remarkable success in modelling complex and noisy behavioural data. However, econometric studies have argued that machine learning techniques are a 'black-box' and difficult to interpret for use in the choice analysis. We develop a data-driven choice model that extends the systematic utility function to incorporate non-linear cross-effects using a series of residual layers and using skipped connections to handle model identifiability in estimating a large number of parameters. The model structure accounts for cross-effects and choice heterogeneity arising from substitution, interactions with non-chosen alternatives and other effects in a non-linear manner. We describe the formulation, model estimation, interpretability and examine the relative performance and econometric implications of our proposed model. We present an illustrative example of the model on a classic red/blue bus choice scenario example. For a real-world application, we use a travel mode choice dataset to analyze the model characteristics compared to traditional neural networks and Logit formulations. Our findings show that our ResLogit approach significantly outperforms MLP models while providing similar interpretability as a Multinomial Logit model.

1. Introduction

Enhancing discrete choice models with neural nets and deep learning optimization algorithms is an active domain of research that has shown promising results (Sifringer et al., 2020; Borysov et al., 2019; Wong and Farooq, 2020). In recent years, experimental use cases of deep learning methods in discrete choice modelling have been explored such as automatic utility discovery (Sifringer et al., 2020), variational inference optimization (Bansal et al., 2019) and remapping explanatory variables into transferrable embeddings for travel behaviour modelling (Pereira, 2019). This paper provides a perspective of how a *residual neural network* formulation accounts for unobserved choice heterogeneity in discrete choice models. While the proposed model we have developed has its roots in the Mother Logit model, it is not a Random Utility Maximization (RUM) consistent model. Likewise, many non-RUM compatible models are used in discrete choice modelling that is still very useful (Hess et al., 2018).

[*] Corresponding author.
E-mail addresses: melvin.wong@epfl.ch (M. Wong), bilal.farooq@ryerson.ca (B. Farooq).

https://doi.org/10.1016/j.trc.2021.103050
Received 11 August 2020; Received in revised form 20 January 2021; Accepted 15 February 2021
Available online 17 March 2021

The increase in popularity of DNNs can be attributed to the general notion that these novel modelling strategies emulate behavioural actions and behaviour formation through neurological adaptations observed in the human brain (Bengio et al., 2015). This is referred to as 'biological plausibility' in deep learning literature and is an efficient way of generating and representing decision models (Friston and Stephan, 2007). The similarity between behaviour theory and DNN has led to many interesting and useful applications in travel behaviour modelling and travel demand forecasting (Cantarella and de Luca, 2005; Lee et al., 2018; Wong et al., 2018; Wang and Zhao, 2019). Intuitively, DNNs are made up of several layers of linear and non-linear operations, called activation functions, which enable the feasibility of estimation from noisy and complex data.

However, machine learning methods have their drawbacks. Even though these methods are increasingly being studied in travel mode choice prediction ever since a decade ago (Karlaftis and Vlahogianni, 2011), their usefulness has been limited to prediction tasks, lacking the explainability of models. Prediction accuracy as a comparison tool has been primarily used in early research in machine learning for travel behaviour modelling work and found to be that neural networks appear to lack consistency with economic principles (Hensher and Ton, 2000). It is argued that DNN may not be suitable for econometric interpretation, and would lead to incorrect assumptions of the stochastic nature of decision-making behaviour. More recent studies have compared the performance of discrete choice and machine learning in prediction. Variable importance analysis has shown that, in most cases, DNNs outperform discrete choice models (Omrani et al., 2013; Hagenauer and Helbich, 2017; Wang and Ross, 2018).

It has been observed in machine learning models that increasing the number of layers beyond a specific limit would degrade the model due to overfitting, unreachable optimal solutions, and model identification problems (Glorot et al., 2011; He et al., 2016). Even in cases showing DNNs producing more accurate predictions[1] than discrete choice models, the structural formulations are not consistent across studies. Another problem with DNNs, although less of immediate concern, is the inconsistency in meta-learning hyperparameter selection, data-leakage and illogically estimated parameters (Hillel et al., 2019). Although not covered in this study's scope, we can address these problems with regularization techniques such as Gradient Batch Normalization or Dropout, or adaptive gradient search such as Adam or AdaGrad (Kingma and Ba, 2014). Moreover, the applicability of machine learning algorithms has not yet been justified in behavioural modelling applications and economic analysis beyond ad-hoc decision tree[2] learning approaches, which are not robust and based on greedy heuristics that do not generalize well from training data (Witten et al., 2016; Brathwaite et al., 2017). Lastly, training and optimizing a multi-layered discrete choice model to capture variations in taste heterogeneity have not yet provided the expected benefits beyond few "shallow" layers (Wang and Zhao, 2019).

This paper proposes a tractable method of incorporating a *data-driven neural network architecture* into a random utility choice model. We seek to improve choice modelling methodologies by incorporating algorithms that work well for deep learning and can be used in choice modelling while performing post-estimation welfare analysis. It extends the systematic utility function to include attributes of other alternatives in potentially non-linear ways to relax the independent and identically distributed (IID) assumptions. The model structure is similar to the existing Mother Logit family of models that incorporate relaxation of the independence of irrelevant alternatives (IIA) property to account for correlation between the IID error terms and the observed explanatory variables (McFadden et al., 1977; Timmermans et al., 1992). Our strategy is inspired by the concept of Residual Neural Networks (*ResNet*) in deep learning literature – adding skip connections between layers allows gradient backpropagation across multiple layers to address the vanishing gradient problem (Bengio et al., 2015). Recent studies have shown that this strategy significantly improves the learning algorithm in deep neural network architecture with marginal or no loss in performance (Witten et al., 2016; He et al., 2016). We show that we can easily adapt the ResNet approach for discrete choice models, and it has similarities to the Mother Logit utility formulation. Our proposed methodology provides the utility function with a generic Deep Learning method of correcting for choice heterogeneity in the model using a residual function in the model formulation. This allows one to leverage deep learning algorithms to estimate new choice models. We define this new choice model structure as a *ResLogit* model.

This paper aims to present a practical implementation of neural networks in choice modelling research that leverages the strengths of deep learning. While this paper deals on the consistency with utility maximization methods, we acknowledge that there are other numerous methods in deep learning literature for optimization through regularization, hyperparameter search, meta-algorithms that are comparable in performance to our ResLogit implementation. This study focuses on the methodological benefits of deep learning in discrete choice analysis. Our work contributes to the use of deep learning methodology in travel behaviour modelling. It has since been highly relevant in today's context of data-driven modelling and use of Big Data for choice and behaviour modelling. In summary, the main contributions of this work are:

- We present the specification of the ResLogit model that uses a residual DNN error correction component in the choice utility in the form of a *data-driven* choice model.
- We present the desirable effects of the ResLogit that enables parameter estimation tractability and interpretability due to the skipped connections between neural network layers and allows for econometric β-parameters to be estimated consistently.
- We analyze the role of residuals in econometric behaviour models and improve previous attempts to integrating deep learning methods in discrete choice applications.

[1] Assuming discrete classification probabilities.

[2] Note: Methods used to select the subset of features in a decision tree results in categories that are sometimes arbitrary. Tree splitting rules are ultimately ad-hoc heuristics. However, comparative selection methods may still be useful if used to inform analysts about which metrics to use in specific choice scenarios.

This paper is organized as follows: Section 2 provides a primer of neural networks and an overview of discrete choice models. Section 3 presents the specification of our proposed ResLogit model. Section 4 demonstrates our formulation on a classic red-bus, blue-bus example. Section 5 evaluates the methodology on a real-world travel dataset and discusses the results. Finally, Section 6 concludes our work and discusses future implications of incorporating deep learning techniques in discrete choice modelling.

2. Background

Logit models have traditionally been used to analyze relationships between observed behaviour and attributes associated with the choices and decision maker's characteristics (Ben-Akiva and Lerman, 1985; Ben-Akiva and Boccara, 1995). This framework has proved successful for decades because of its parsimonious, tractable, and flexible model formulation for representing rational behaviour assumptions. It assumes that the underlying decision processes are unknown from the observer, and decision-makers select their preferred choice by ranking all potential alternatives and choosing the alternative with the maximum utility through Random Utility Maximization (RUM) theory. The modeller is assumed to have incomplete information about the decision-maker's behaviour, and the model will have to account for some uncertainty.

An important feature of the Logit model is the IIA property, which is an outcome of the assumption that the error terms of the alternatives in an MNL model are IID (McFaddden, 1978). When the error terms are correlated, strict IID assumption may lead to an incorrect forecast and model misspecification. The Logit model imposes a random error term representing behavioural uncertainty and account for the lack of information presented to the analyst. This random error term is assumed to be uncorrelated to the attributes of the alternatives. Extensions to the Logit models such as Nested Logit and Mixed Logit have been developed to account for the error correlation when the assumption does not hold.

2.1. *Representation of non-linearity and cross-effects in choice utilities*

Model misspecification may arise when the error terms are correlated with non-chosen alternatives. Various studies in discrete choice modelling have accounted for heterogeneity across choice alternatives and decision-makers by incorporating attributes of non-chosen alternatives known as *cross-effects*. The assumption is that the included additional function conditions for part of the error term correlate with the non-chosen alternatives. There are several approaches to dealing with similarities and cross-effects between alternatives (Schuessler and Axhausen, 2007):

- Segmentation into nests or classes,
- Analyzing the variance-covariance structure, and
- Incorporating similarity factors into the deterministic part of the utility.

The first group consists of extensions to the MNL model such as the Nested Logit model to partially relax the IID assumption by segmenting alternatives into subsets. They are similar within each group (correlated) but independent between groups (non-correlated). These models specify the correlation between alternatives by allowing attribute coefficients to vary between observations, class segments or individuals. Although this model formulation works well with simple stated preference choice scenarios where the analyst can control the survey questions and options, cognitive bias formed during the behaviour learning process, e.g. anchoring effects, are not fully captured (Tversky and Kahneman, 1981). For instance, when a traveller makes a mode choice decision, there is a tendency to rely heavily on the information that they learn. The learning process may also evolve, resulting in Spatio-temporal heterogeneity.

The second group consists of the Generalized Extreme Value (GEV) model family (e.g. Mixed Logit), and Probit models which allow for different (co-)variances among the error term in the utility function (McFadden et al., 1977; Daganzo et al., 1977). Multivariate distributed random error terms are introduced into the utility to capture potentially any correlation structure. This assumption works well with simple behavioural models and allows for tractable estimation. It does not necessarily reflect observed behaviour accurately with arbitrarily defined error distribution for more complex behavioural models. However, we can also derive individual-specific estimates from the individual's conditional distribution based on their choices (Hensher and Greene, 2003). Identification and computation of a large number of random distribution are still problematic in conventional discrete choice applications. Recent research efforts have also focused on Mixed Logit estimation using optimization techniques primarily used in machine learning. In particular, Bayesian variational inference optimization methods have shown to be promising (Bansal et al., 2020).

The third group consists of models that include an explicit measure of similarity among alternatives in the utility function. This group include hybrid choice models and the integrated choice and latent variable (ICLV) family of models. Most notably, the Mother Logit model introduced by McFadden (1975) represents a generalization of the conventional MNL model, but not necessarily RUM consistent, by allowing for the existence of cross-effects and other substitutions (reference dependence, decoy, anchoring bias, regret, etc.) in the utility to relax the IID assumption (Timmermans et al., 1992).

The Mother Logit formulation can approximate any discrete choice model in which the alternative's scale value is a function of all attributes of all choices (Timmermans et al., 1992). Other choice model development such as the Random Regret Minimization (RRM) model (Chorus, 2010) which include terms from foregone alternatives, can be reformulated as a Mother Logit model (Mai et al., 2017). The RRM model bases the assumption that one or more alternatives outperform the desired choice. This is translated into an anticipated regret function, and the analyst can formulate the non-linear utility as a function of attribute cross-effects of all the alternatives in the deterministic component and a random error term. Mai et al. (2017) also presented a case of a Recursive Logit (RL) model based on the Mother Logit formulation. Mai et al. (2017) formulated the RL model utility functions as a route choice problem, which computes the sum of the outgoing link utility and the expected maximum utility to the destination node, accounting for these cross-effects in the link utility functions. When links overlap between different feasible route choice alternatives, the non-linear RL utility of a given route choice would include attributes from other route alternatives.

Cross-effect represents the utility correction measure of similarity or dissimilarity across all attributes of all alternatives (Timmermans et al., 1992). A negative cross-effect indicates that an IIA model overestimates the utility of the alternative due to correlated attributes and alternatives (e.g. the red/blue bus problem (McFadden, 1973)). Likewise, a positive cross-effect indicates that the utility is underestimated and a positive bias correction is required to account for the choice heterogeneity. The Mother Logit formulation implies that the model violates RUM regularity conditions (Timmermans et al., 1992). Nevertheless, such model flexibility can accommodate behavioural anomalies incompatible with RUM based models (Hess et al., 2018).

2.2. Generalized approach to capture non-linearity and cross-effects in discrete choice models using DNNs

Passive data collected from sensors, devices and infrastructure that track decision making actions over time can reveal learning behaviour and trends of the decision-makers. The general approach of representing decision-making uncertainties and learning processes as probabilistic error terms may be sufficient in obtaining satisfactory approximations. However, it is often difficult to identify the source of heterogeneity due to the complex interactions between influences from various attributes of non-chosen alternatives over a long period of interaction. Furthermore, it provides no useful indication of selecting the error term mixing distribution or how many mixing distributions are required to reach an acceptable estimation of the decision-making behaviour (McFadden and Train, 2000).

Combining DNNs and discrete choice modelling strengths have been explored in the past several years (Borysov et al., 2019; Bansal et al., 2019; Pereira, 2019; Wong and Farooq, 2020; Badu-Marfo et al., 2020). These new hybrid models are designed to capture learning behaviour and trends from large datasets, independent from the subjective bias induced from stated preference survey questionnaires. The decision making learning algorithm is assumed to contain non-linear cross-effects, which results in complex error distributions and a non-linear utility function. In practical choice modelling applications, the learning algorithm's process updates the model is unknown to the modeller. Therefore it is said to be a 'black-box' model (Breiman, 2001). Non-linear activation functions in DNNs are assumed to represent taste variations and random heterogeneity in the choice model. For instance, a non-compensatory decision protocol distribution is often used to generalize decision rules in discrete choice, rather than to define fixed assumptions about the error distribution (Vythoulkas and Koutsopoulos, 2003).

Although neural networks have proved popular in recent years with their simple design and implementation, they rely on hyperparameter search or meta-learning process which cannot be intuitively interpreted from a micro-economic perspective. Hyperparameters are the learning algorithm parameters that specify the learning procedure: L_1 and L_2 penalties, gradient step size, decay or initialization conditions. In some situations, hyperparameter tuning[3] can yield state-of-the-art performance. Lipton (2018) gave the hypothesis on lack of model interpretability by identifying that most machine learning-based systems may achieve high accuracy despite failing to explain where the source of the difference lie. The MLP model is seen as a 'black-box' model and will not be able to identify the beta parameters associated with the independent explanatory variables. Model identifiability may be problematic as there can be multiple model specification defined by the same set of parameters.

2.3. General formulation of a neural network model

We explain the necessary notations and formulation of an MLP network, the *ResNet* architecture, and how we can integrate the residual functions into a choice model, which follows a logically consistent extension of traditional MNL that relaxes the IIA property.

Each neuron in an MLP is a basic processing unit that performs a non-linear transform on the input (Lee et al., 2018). The goal is to approximate some function $y = f^*(\mathbf{V})$ with $y = f(\mathbf{V};\theta)$, where the input $\mathbf{V}$ is a linearized function of a vector of observed variables $\mathbf{x}$ and a vector of estimated parameters $\boldsymbol{\beta}$, denoted as $\mathbf{V} = f(\boldsymbol{\beta},\mathbf{x})$. The function $f(\mathbf{V};\theta)$ is a map of the linear components $\mathbf{V}$ to a vector of discrete choice probabilities y. θ is the neural network parameters that result in the best approximation of f^*. During the training process, the model is estimated by a batched gradient descent algorithm given an objective function, i.e. maximum likelihood estimation[4] [5]. The MLP architecture can be represented mathematically as a series of chain functions:

$$\begin{aligned}
\mathbf{h}^{(1)} &= f^{(1)}(\mathbf{V})\\
\mathbf{h}^{(2)} &= f^{(2)}\left(\mathbf{h}^{(1)}\right)\\
&\dots\\
\mathbf{h}^{(M)} &= f^{(M)}\left(\mathbf{h}^{(M-1)}\right)\\
y &= \mathit{softmax}\left(\mathbf{h}^{(M)}\right)
\end{aligned} \tag{1}$$

[3] hyperparameter tuning refers to the specification of the *learning algorithm*, not the model parameters, e.g. β parameters.

[4] Batched gradient descent is most used in deep learning optimization. For most machine learning problems, the data size is too large for quasi-Netwon methods such as BFGS/L-BFGS algorithm to perform in *comparable time*. Furthermore, computing in batches allows for parallelized computation on GPUs.

[5] In general, the *no free lunch theorem* in optimization states that no one solution works best for all problems

where $f^{(1)}, f^{(2)}, \ldots, f^{(M)}$ are the activation functions of the DNN and M gives the depth of the model. For example, a 3-layer DNN results in the general form $f(\mathbf{V}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{V})))$. $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \ldots, \mathbf{h}^{(M)}$ are the intermediary non-linear output of each m^{th} activation function and the final layer is a *softmax* function[6], and the output results in a vector of discrete probabilities associated with each choice. The choice of activation functions is loosely guided by neuroscience observations and 'biological plausibility', which refers to the similarity between the behaviour theory and signal transmission in the nervous system (Goodfellow et al., 2016). The activation function can be linear or non-linear. For example, using a sigmoid function: $f(\mathbf{V}) = (1 + e^{-\mathbf{V}})^{-1}$ results in a probabilistic output between 0 and 1. In general, most DNN architectures suffers from non-identifiability due to the nature of the chain of non-linear activation functions – the change in β parameter associated with the explanatory variable cannot be mapped directly to the output probabilities.

The naïve intuition is that the MLP can learn increasingly complex features by adding more layers, and each layer returns an "improved" approximation of f^*. On the contrary, research has shown that the number of layers representing a perfect model does not follow an asymptotic limit. Still, it deteriorates as one increases the number of layers (Srivastava et al., 2015; He et al., 2016), contradicting the assumption that DNNs provides greater flexibility than conventional discrete choice models. Observations in discrete choice literature affirm this technical limitation of using multiple deep layers to improve modelling accuracy (Alwosheel et al., 2018; Lee et al., 2018).

2.4. Formulating the neural network as a dynamical system

The *ResNet* architecture was proposed by He et al. (2016) to overcome the limitations of the MLP model. We can interpret the model as a discretization of a dynamical system that exploits the use of identity shortcuts to enable the flow of information across layers without causing model degradation from repeated non-linear transformations (He et al., 2016). From an optimization perspective, the hypothesis is that it is easier to optimize "a small change to the input rather than improving the entire layer of inputs at once" (He et al., 2016). This approach potentially provides an attractive possibility for modellers to retain the econometric variables and allows the neural network function to approximate the underlying error variance from a choice modelling perspective. Furthermore, it has been proven that the *ResNet* model architecture has no critical points other than the global minimum (Hardt and Ma, 2016).

The *ResNet* model $y = f(\mathbf{V})$ is defined as the following series of functions:

$$\begin{aligned}
\mathbf{h}^{(1)} &= f^{(1)}\left(\mathbf{V}\right) + \mathbf{V} \\
\mathbf{h}^{(2)} &= f^{(2)}\left(\mathbf{h}^{(1)}\right) + \mathbf{h}^{(1)} \\
&\ldots \\
\mathbf{h}^{(M)} &= f^{(M)}\left(\mathbf{h}^{(M-1)}\right) + \mathbf{h}^{(M-1)} \\
y &= softmax\left(\mathbf{h}^{(M)}\right)
\end{aligned} \tag{2}$$

The *ResNet* uses a skip connection mechanism (Eq. 2) to the gradient to propagate through the layers, preventing the vanishing gradient problem (He et al., 2016). The last line of Eq. 2 transforms the output of the final intermediate layer to a vector of probabilities using the *softmax* function[7]. We can further generalize the *ResNet* blocks as a series of recursive functions:

$$\mathbf{h}^{(m)} = f^{(m)}\left(\mathbf{h}^{(m-1)}; \theta^{(m)}\right) + \mathbf{h}^{(m-1)}, \mathbf{h}^{(0)} = \mathbf{V}, \quad \text{for } m = 1, \ldots, M \tag{3}$$

where $\mathbf{h}^{(0)}$ is the input after the initial linearization of the utility and $\mathbf{h}^{(M)}$ is the output map before the *softmax* function. Approximating the parameters of the neural network $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(M)}$ is equivalent to solving for a series of linear discrete optimal control problem $U_m = f(V_m; \theta_m) + \varepsilon_m$. We can also interpret $\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(M)}$ as a series of non-linear utility components that capture the cross-effects induced by similarity or overlap with the non-chosen alternatives. If $f^{(m)}$ in Eq. 3 is large, it indicates the presence of cross-effects on the output probability. If this value is close to zero for all m (non-linear cross-effects not present), the model would collapse to a Logit model.

3. Specification of the ResLogit choice model

Our proposed ResLogit choice model improves discrete choice estimation by incorporating a neural network based on the recent *ResNet* architecture. Fig. 1 shows a comparison between an MNL, MLP and the proposed ResLogit model as a simplified graphical model. The general framework of our ResLogit architecture is that it is much more efficient to model the unobserved heterogeneity using a neural network rather than applying a neural network to the entire utility. Sifringer et al. (2020) applied a similar concept for a Learning MNL (LMNL) model, although using a fully connected neural network as a linear addition to the utility plus an unobserved error component. This ad-hoc approach divided the explanatory variables into two groups, where one was used in the systematic linear utility and the other group in the neural network capturing the average effects. In general, we specify the utility function as a sum of the deterministic component of observed characteristics and a neural network component that captures the unobserved heterogeneity in the choice process. Our approach's advantage is that the skip allows for a greater chance of identifiability in the estimation of each

[6] This softmax function is equivalent to a conditional Logit in discrete choice problems.

[7] For consistency with literature, we denote *softmax* in the context of neural networks, and Logit in the context of discrete choice. However, both functions are mathematically equivalent

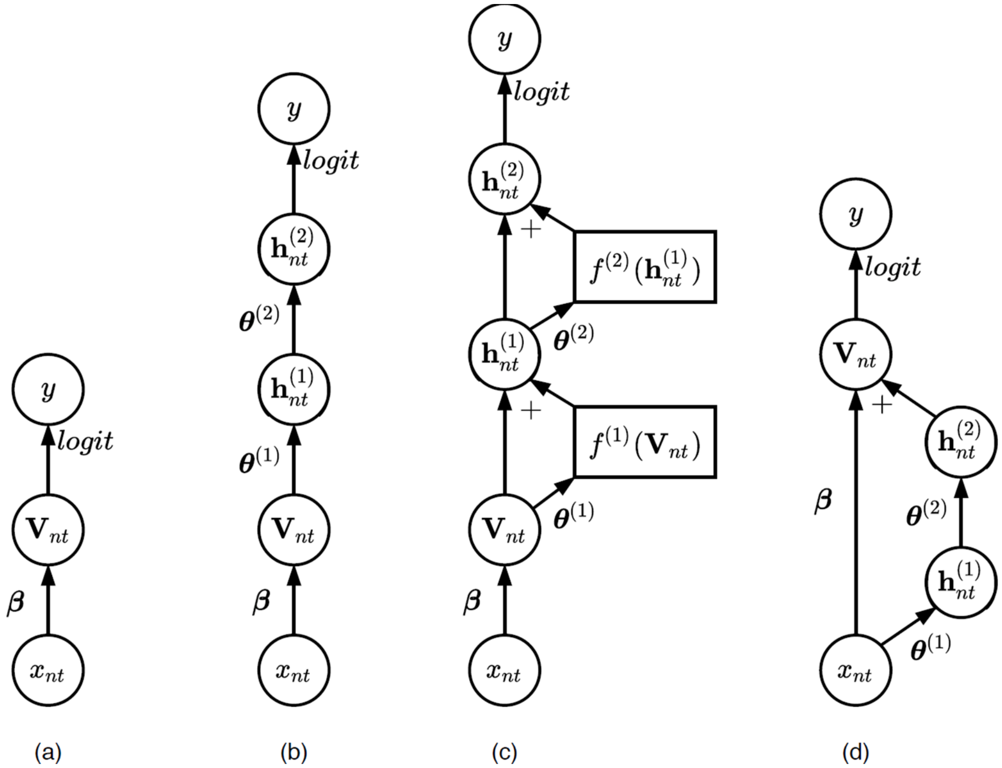


Fig. 1. Simplified graphical model. (a) A Multinomial Logit model. (b) A MLP network with 2 hidden layers. (c) The proposed ResLogit model with 2 residual layers. Here we show the models expressed as symbolic operators that compute each step from the input x_{nt} to the output probabilities y. The graph operator $+$ compute $h^{(m)} = h^{(m-1)} + f(h^{(m-1)})$. We omit the ASC variables for brevity. (d) Representation of the LMNL model used in Sifringer et al. (2020).

layer of the neural network. In contrast, the L-MNL model would still be vulnerable to the vanishing gradient problem.

A utility U_{int} is defined by a deterministic component V_{int} and a random error component ε_{int}:

$$U_{int} = V_{int} + \varepsilon_{int} \tag{4}$$

The deterministic component is a linear function of a vector of attributes x_{nt} of a single alternative with a vector of estimated parameters $\boldsymbol{\beta}$. The most general expression of the Logit model, the Mother Logit model, introduces a random variable g_{int} in the utility that is a function of all attributes of all choices.[8] Note, in some cases, the random variable g *replaces* the deterministic part V_{int} (Hess et al., 2018). Our ResLogit model's utility takes the general expression of the Mother Logit model as the output of the residual component. The utility U_{int} of individual n selecting choice i in a choice task t, from a choice set of J alternatives with the residual component term is as follows:

$$U_{int} = V_{int} + g_{int} + \varepsilon_{int} \tag{5}$$

The utility is a linear function of the systematic observed component V_{int}, the residual component g_{int}, and an extreme value distributed error term ε_{int} representing the remaining unobserved errors not captured in the neural network. $\mathbf{V}_{nt}$ is a $J \times 1$ vector of utilities v_{jnt} associated with each individual n for choice task t:

$$\mathbf{V}_{nt} = \begin{bmatrix} V_{1nt} \\ V_{1nt} \\ \vdots \\ V_{jnt} \end{bmatrix}_{J\times 1} \tag{6}$$

and $\mathbf{g}_{nt}$ is a $J \times 1$ vector of residual components g_{jnt} associated with the respective utility j that contains all attributes from all alternatives:

[8] Note to readers that the subscript i refers to the index of the alternative in this section and the following sections. It does not refer to g_{int} having only attributes from the i^{th} alternative. We represent a function that depends solely on attributes from the alternative with an uppercase notation (e.g. V).

$$\mathbf{g}_{nt} = \begin{bmatrix} g_{1nt} \\ g_{1nt} \\ \vdots \\ g_{jnt} \end{bmatrix}_{J\times 1} \tag{7}$$

Eq. 5 would lead to the choice probability $y_i = f_i(\mathbf{V}, \mathbf{g})$ for $i \in 1, \ldots, J$:

$$P\left(i\right) = y_i = \frac{\exp(V_{int} + g_{int})}{\sum_{j\in\{1,\ldots,J\}} \exp\left(V_{jnt} + g_{jnt}\right)} \forall i \in \left\{1, \ldots, J\right\} \tag{8}$$

where:

$$\mathbf{g}_{nt} = -\sum_{m=1}^{M} \ln\left(1 + \exp\left(\theta^{(m)}\mathbf{h}_{nt}^{(m-1)}\right)\right) \tag{9}$$

$$\mathbf{h}_{nt}^{(0)} = \mathbf{V}_{nt} \tag{10}$$

For any block m:

$$\mathbf{h}_{nt}^{(m)} = \mathbf{h}_{nt}^{(m-1)} - \sum_{m'=1}^{m} \ln\left(1 + \exp\left(\theta^{(m')}\mathbf{h}_{nt}^{(m'-1)}\right), \quad \text{for} \quad m = 1, \ldots, M \tag{11}$$

and $\theta^{(m)}$ is a $J \times J$ matrix of residual parameters:

$$\theta^{(m)} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1j'} \\ c_{11} & c_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ c_{j1} & \cdots & \cdots & c_{jj'} \end{bmatrix}_{J\times J} \quad \text{for} \quad m = 1, \ldots, M \tag{12}$$

where $c_{jj'}$ is the parameter matrix element for the j^{th} row and j'^{th} column, and $\mathbf{h}_{nt}^{(m)}$ is a $J \times 1$ vector of non-linear utility components for the m^{th} residual layer:

$$\mathbf{h}_{nt}^{(m)} = \begin{bmatrix} h_{1nt}^{(m)} \\ h_{2nt}^{(m)} \\ \vdots \\ h_{jnt}^{(m)} \end{bmatrix}_{J\times 1} \quad \text{for} \quad m = 1, \ldots, M \tag{13}$$

The parameter matrices are defined such that the dimension of the residual output $\mathbf{g}_{nt}$ matches the dimension of $\mathbf{V}_{nt}$ for an element-wise additive operation. We can have several intermediate neural network layers of varying sizes within each residual layer, which is one of the conveniences of the neural network architecture. $\theta^{(m)}$ serves as the similarity or cross-effect factors to the utility function. The chosen alternative's utility is increased or decreased by its degree of similarity with other non-chosen alternatives by this factor. The MNL perspective corresponds to shifting the vector of utilities by $\mathbf{g}_{nt}$. If the cross-effect factors are zero, i.e. $\theta^{(m)} = 0$ for all m, then the utility surplus is shifted by 0 and falls back to an MNL model.

Another observation is that the choice probability is conditional on the expectation of the output of the residual terms:

$$\mathbf{Q}_{nt}^{(m)} = \frac{1}{1 + \exp\left(\theta^{(m)}\mathbf{h}_{nt}^{(m-1)}\right)}, \quad \text{s.t.} \quad \mathbf{Q}_{nt}^{(m)} \geqslant 0, \quad \text{for} \quad m = 1, \ldots, M \tag{14}$$

and if we assume that $\mathbf{Q}_{nt}^{(m)} = \{Q_{jnt}^{(m)}\}$ for $j \in \{1, \ldots, J\}$ is a vector of probabilities, we can rewrite the ResLogit formulation in Eq. 8 as a conditional choice probability:

$$P\left(i\right) = y_i = \frac{\left(\prod_m Q_{int}^{(m)}\right) \exp\left(V_{int}\right)}{\sum_{j\in\{1,\ldots,J\}} \left(\prod_m Q_{jnt}^{(m)}\right) \exp\left(V_{jnt}\right)}, \forall i \in \left\{1, \ldots, J\right\} \tag{15}$$

The residual component (Eq. 9) derives from entropy, or expected surplus function of the respective residual layers and the corresponding logsum term is the result of the log of the Logit probability denominator. Behaviour modelling uses entropy to measure the variation or accessibility of a specific choice (Erlander, 2010). For example, Mattsson and Weibull (2002) characterized such formulation as maximization of the sum of the expected utility and a weighted entropy. Anas (1983) postulated that the entropy principle in choice models correspond to how much information-seeking behaviour is used to find the "best" utility specification.

Fosgerau et al. (2017) and Matějka and McKay (2015) also illustrated the affinity to generalized bounded rationality and the duality between discrete choice and rational inattention behaviour. Consequently, information cost acts as a barrier between prior beliefs and the decision making actions, which results in choice heterogeneity. An agent optimizes his or her desired outcome by minimizing this information cost (Matějka and McKay, 2015). Our ResLogit model aims to extend this concept by allowing for a data-driven surplus expression in the utility function (Presented in Eq. 5) to emulate the decision-makers' learning process.

3.1. *Depth of the neural network*

Increasing the depth of the neural network increases the number of additive residual terms in the utility function. The residual layers represent the underlying unobserved behaviour distribution that is not captured by the explanatory variables. This mathematical formulation allows the model to reflect individual taste heterogeneities in the non-linear residual function. Unlike a typical MLP model or the recently developed Learning-MNL model (Sifringer et al., 2020), training a ResLogit model does not suffer from the vanishing gradient problem. This eliminates the singularities caused by model non-identifiability. This property's key implication on choice modelling is that we can operationalize the learning behaviour as a function in the utility while retaining the same econometric parameters in the structural equation.

3.2. *Estimation approach*

The estimation procedure is a data-driven first-order stochastic gradient descent SGD learning algorithm, and we evaluate the performance on an out-of-sample validation set. In data-driven optimization, we are maximizing a performance measure (e.g. out-of-sample performance) by indirectly maximizing a different surrogate objective function (e.g. maximizing log-likelihood of the training data). We typically assume that the out-of-sample dataset is independent and identically distributed from the training dataset. In contrast, pure optimization of discrete choice models directly maximizes the likelihood objective function, which is a goal of itself. This method of estimating a large number of parameters has been proven efficient in machine learning. In some cases, a surrogate objective function approach may result in a faster and better solution (Goodfellow et al., 2016). Other pre-conditioning methods or extensions can also be implemented into the surrogate objective function allowing it to reach multiple local optimum points and provide a regulating effect. For example, these pre-conditioning includes adding momentum, adaptive learning rate methods or gradient noise normalization, see Ruder (2016) for an overview of such methods. Another important difference is that the final convergence criteria are based on the performance measure, not the surrogate objective function within data-driven optimization. This approach enables the algorithm to terminate when overfitting begins to occur (early-stopping criteria). The estimation reaches convergence when the objective function no longer improves.

For this reason, a data-driven approach is more suitable in estimating our ResLogit model since a pure optimization approach will run into model non-identifiability issues due to a large number of estimated parameters.

3.2.1. *Objective function and parameter updates*

The set of optimal parameters θ and $\boldsymbol{\beta}$ are estimated by maximizing the log-likelihood, where the log-likelihood is as follows:

$$LL\left(\theta,\boldsymbol{\beta}\right)=\sum_{n=1}^{N}\ln P\left(i_n|\mathbf{x}_n;\theta,\boldsymbol{\beta}\right). \tag{16}$$

The mini-batch SGD algorithm performs the following update rule on each iteration t:

$$\theta_{t+1}=\theta_t-\eta_t\nabla_\theta\mathcal{J}_{\mathcal{B}}(\theta,\boldsymbol{\beta}), \tag{17}$$

$$\boldsymbol{\beta}_{t+1}=\boldsymbol{\beta}_t-\eta_t\nabla_{\boldsymbol{\beta}}\mathcal{J}_{\mathcal{B}}(\theta,\boldsymbol{\beta}), \tag{18}$$

where:

$$\nabla_\theta\mathcal{J}_{\mathcal{B}}\left(\theta,\boldsymbol{\beta}\right)=\frac{1}{K}\sum_{n'\in\mathcal{B}}\nabla_\theta LL_{n'}\left(\theta,\boldsymbol{\beta}\right), \tag{19}$$

$$\nabla_{\boldsymbol{\beta}}\mathcal{J}_{\mathcal{B}}\left(\theta,\boldsymbol{\beta}\right)=\frac{1}{K}\sum_{n'\in\mathcal{B}}\nabla_{\boldsymbol{\beta}} LL_{n'}\left(\theta,\boldsymbol{\beta}\right), \tag{20}$$

and K is the batch size, $\mathcal{B}$ is a batch of observations sampled from $\mathbf{x}_n, n'$ denotes the observation in the batch and η_t is the learning rate. We can regard $\nabla\mathcal{J}_{\mathcal{B}}(\theta,\boldsymbol{\beta})$ as a noisy estimate of the true gradient $\nabla LL(\theta,\boldsymbol{\beta})$. We sample from the training set and adjust the $\boldsymbol{\beta}$ and θ parameters to reduce the training error, then we monitor the error in the validation by sampling from the validation dataset. The goal of the optimization is to reduce the validation error while also reducing the difference between the training and validation error. This can also be achieved by taking the model at the maximum log-likelihood of the validation dataset with an assumption that the estimation on the training dataset is asymptotic as the number of iterations on the samples $N\rightarrow\infty$. The derivatives of the estimated parameters is computed using backpropagation (Goodfellow et al., 2016). Given the ResLogit formulation and taking the backpropagation from the output log-likelihood, the derivative of the log-likelihood with respect to $\boldsymbol{\beta}$ is:

$$\frac{\partial LL}{\partial \boldsymbol{\beta}} = \frac{\partial LL}{\partial \mathbf{V}}\frac{\partial \mathbf{V}}{\partial \boldsymbol{\beta}} + \frac{\partial LL}{\partial \mathbf{h}^{(m)}}\frac{\partial \mathbf{h}^{(m)}}{\partial \boldsymbol{\beta}} + \frac{\partial LL}{\partial \mathbf{h}^{(m-1)}}\frac{\partial \mathbf{h}^{(m-1)}}{\partial \boldsymbol{\beta}} + \ldots + \frac{\partial LL}{\partial \mathbf{h}^{(1)}}\frac{\partial \mathbf{h}^{(1)}}{\partial \boldsymbol{\beta}} \tag{21}$$

The gradient formulation is shown in Eq. 21 that by the nature of the residual connections, each derivative of the residual layers is independently computed. This prevents the phenomena known as vanishing gradient. If any of the gradients is computed to be zero, it does not affect the total backpropagated value and the $\boldsymbol{\beta}$ parameters can still be updated. This allows the ResLogit to converge to an optimal MNL solution, even with non-identifiable residual layers. In contrast, with a fully connected MLP model, the gradient formulation is a result of a chain rule:

$$\frac{\partial LL}{\partial \boldsymbol{\beta}} = \frac{\partial LL}{\partial \mathbf{h}^{(m)}} \times \frac{\partial \mathbf{h}^{(m)}}{\partial \mathbf{h}^{(m-1)}} \times \ldots \times \frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{V}} \times \frac{\partial \mathbf{V}}{\partial \boldsymbol{\beta}} \tag{22}$$

In Eq. 22, if any of the intermediate derivatives are zero, then the total derivative is zero, and the model fails to learn and update $\boldsymbol{\beta}$, resulting in model non-identifiability. The number of parameters used is relative to the number of alternatives in the choice set. Each element in the matrix corresponds to the cross-effects of other alternatives on the chosen alternative. The diagonal elements in the matrix are the cross-effects with itself, i.e. a scale factor adjustment. If this residual matrix is an identity matrix, that means that there are no cross-effects induced between alternatives (IIA holds), and the model collapses into a standard MNL model.

4. Red/Blue bus theoretical example

We show an example of how a simple nesting structure can be obtained using the ResLogit formulation in a hypothetical scenario. Let us consider the red/blue bus problem. The red/blue bus problem is a classic example of IIA property violation in choice models. The problem arises in the assumption that the error terms for the red and blue bus options are independent, but they are correlated and share similar decision attributes in reality. This means that the change in utility for a red bus will influence the change in utility of the blue bus. To relax this assumption, choice modellers often use a Nested Logit model to relax the IIA assumption by adding a conditional probability term or logsum term. The choice scenarios are summarized in Table 1.

4.1. Scenario description

In the first scenario (Scenario 1), assuming that we have a vector of 2 choices in a choice task *t*. $\mathbf{V} : \{V_{car}, V_{bus}\}$, where each alternative has the same utility $V_{car} = 1, V_{bus} = 1$ Under strict IID assumptions, the probability of choosing either bus or car is, therefore, $P_{car} = P_{bus} = 0.5$.

In the second scenario (Scenario 2), suppose that now we have a red bus (V_{red_bus}) and blue bus (V_{blue_bus}) option in place of V_{bus},$\mathbf{V} = \{V_{car}, V_{red\_bus}, V_{blue\_bus}\}$. The utility of each alternative does not change, and all 3 alternatives have the same utility: $V_{car} = 1, V_{red} = 1$, $V_{blue} = 1$. Assuming the choice task is IID, the probabilities for the respective alternative should result in: $P_{car} = 0.5, P_{red\_bus} = 0.25$, and $P_{blue\_bus} = 0.25$. The probability of *car* choice does not change when we add a new mode to the choice set. However, the actual probabilities when estimated by an MNL model would result in: $P_{car} = 0.33, P_{red\_bus} = 0.33$, and $P_{blue\_bus} = 0.33$, which does not seem plausible and violates IIA property conditions.

In the third scenario (Scenario 3), under our proposed ResLogit model, the correlation between the red and blue bus is corrected by a residual vector $\mathbf{g}$, with residual parameter matrix $\theta^{(1)}$. Using a 1-layer ResLogit model and a residual vector function defined by $\mathbf{g} = -\ln(1 + \exp(\theta^{(1)}\mathbf{V}))$, we simulate a choice scenario with alternatives *car*, *red bus*, *blue bus*.

Table 1
Illustration of red/blue bus choice scenario showing the effect of residual correction factors of a 1-layer model.

Choice	V_i	g_i	$\exp(V_i + g_i)$	$P(i)$
Scenario 1				
car	1	–	2.718	0.5
bus	1	–	2.718	0.5
Scenario 2				
car	1	–	2.718	0.33
red bus	1	–	2.718	0.33
blue bus	1	–	2.718	0.33
Scenario 3 (competing car/bus)				
car	1	−0.127	2.394	0.468
red bus	1	−0.693	1.359	0.265
blue bus	1	−0.693	1.359	0.265
Scenario 3 (non-competing car/bus)				
car	1	−0.693	1.359	0.482
red bus	1	−1.313	0.731	0.259
blue bus	1	−1.313	0.731	0.259

We assume at a value of 1 represents a positive cross-effect and a -1 value denotes a negative cross-effect and 0 value represents no cross-effects (IIA property holds). The negative value of cross-effects between the car and bus option may suggest that the alternatives are competing options (e.g. buses and cars sharing the same road segment). we assign a value of $\{1\}$ to elements $c_{32}^{(1)}$ and $c_{23}^{(1)}$ and a value of $\{-1\}$ to elements $c_{12}^{(1)}, c_{21}^{(1)}, c_{13}^{(1)}$ and $c_{31}^{(1)}$:

$$\theta^{(1)} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}. \tag{23}$$

Given a 3×1 vector of utilities $\mathbf{V} = [1 \quad 1 \quad 1]^\top$, the residual vector $\mathbf{g}$ is:

$$\mathbf{g} = -\ln\left(1 + \exp\left(\theta^{(1)}\mathbf{V}\right)\right), \tag{24}$$

$$= -\ln\left(1 + \exp\left(\begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\right)\right), \tag{25}$$

$$= \begin{bmatrix} -0.127 \\ -0.693 \\ -0.693 \end{bmatrix}, \tag{26}$$

giving the choice probabilities as:

$$P\left(i\right) = \frac{\exp(V_i + g_i)}{\sum_{j \in C} \exp\left(V_j + g_j\right)} \quad \text{for} \quad i \in car, redbus, bluebus \tag{27}$$

$$P(car) = 0.468; P(red\ bus) = 0.265; P(blue\ bus) = 0.265;$$

The probabilities in Eq. 27 show that with an addition of the residual matrix to account for the cross-effects, we have moved the choice probabilities of the car and red/blue bus options toward the true IIA conditions without changing the underlying utilities.

Now, if we assume no cross-effects between the car and bus alternatives (both car and buses are not sharing the same road segment), we update Eq. 23 with values of $\{0\}$ for parameters $c_{12}^{(1)}, c_{21}^{(1)}, c_{13}^{(1)}$ and $c_{31}^{(1)}$:

$$\theta^{(1)} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{28}$$

The resulting residual vector would be:

$$\mathbf{g} = \begin{bmatrix} -0.693 \\ -1.313 \\ -1.313 \end{bmatrix}, \tag{29}$$

giving the choice probabilities as:

$$P\left(i\right) = \frac{\exp(V_i + g_i)}{\sum_{j \in C} \exp\left(V_j + g_j\right)}, \quad \text{for} \quad i \in car, redbus, bluebus \tag{30}$$

$$P(car) = 0.482; P(red\ bus) = 0.259; P(blue\ bus) = 0.259;$$

In principle, the nests between the car and the bus options are not pre-specified *a priori* by the modeller. The parameter matrix is estimated from data and defines the nesting structure or error term correlation of the choice alternatives. The first observation of the hypothetical example shown above is that with a logical assumption of positive ($c_{bus,bus} = 1$) cross-effect residual parameter between the two bus alternatives and zero ($c_{bus,car} = 0$) cross-effect residual parameter between the car and bus alternatives would result in a nesting structure which reflects the relaxed IID assumption probabilities. The second observation stems from the correlations between error terms of competing alternatives. If the residual parameters are negative, it accounts for competing alternatives (e.g. buses and cars share the same road segment from the origin to destination), resulting in a slightly different outcome than a non-compete scenario.

5. Case study

This study evaluates our proposed ResLogit model's effects and performance in three criteria: model depth, model degradation, and model predictive performance compared to an MLP neural network. We also evaluate the residual effects on econometric parameters by comparing the beta and standard error values with a baseline MNL model without the residual layers.

We evaluate the ResLogit model's performance using individualized characteristics and attributes in a revealed preference (RP) travel survey dataset using out-of-sample accuracy at the minimum validation loss point on the validation curve. We computed the accuracy using a 30% hold-out validation set from our dataset. We compared the model degradation effects between our ResLogit and a vanilla MLP model with identical model hyperparameters to address the adverse impact of model degradation from increasing layers. We showed the effects of increasing layers in the ResLogit model and the MLP model on estimation accuracy and model identifiability.

5.1. Data and model description

We used the 2016 *Mtl Trajet* RP dataset collected from the user's smartphone data on a mobile application (Yazdizadeh et al., 2019). A list of explanatory variables and the choice set used for this mode choice prediction analysis are shown in Table 2. The respondents' travel diary includes mode choice, activity choice, trip attributes (e.g. trip length, start/end time, location) and GPS trajectories. The travel survey was conducted over four months, from September to December 2016. In total, there were 60,365 unique trips made during the period. To evaluate out-of-sample performance, we divide the dataset into two sets using a 70:30 training/validation split ($N_{training} = 42,256$ samples, $N_{validation} = 18,109$ samples). We developed the model estimation algorithm using open-source deep learning libraries in Python. The code for our experiments is available on our Github page.[9]

We iterated over the experiment by varying the depth of the ResLogit and MLP neural network using 2, 4, 8 and 16 hidden layers ($M = \{2,4,8,16\}$). Note that our study only shows a relative comparison between the models with a similar number of layers and neural network hyperparameters. The objective of this experiment is to show the effectiveness of the ResLogit approach as a way of incorporating deep learning methods into discrete choice models over a conventional MLP neural network.

This experiment considers three specific objectives:

1. Effects of the number of residual layers on the model β parameters.
2. Model validation accuracy and maximum log-likelihood estimation comparison.
3. Comparison of estimated β parameters between the ResLogit model and MNL model.

The model estimation process begins with a baseline MNL estimation. Next, the MLP models were estimated (4 models, one each for 2, 4, 8 and 16 hidden layers), and labelled as MLP-2, MLP-4, MLP-8 and MLP-16, respectively. We performed the same training process on the ResLogit models (RL-2, RL-4, RL-8, RL-16). For the learning algorithm, we used the mini-batch SGD learning algorithm with a mini-batch size of 64 (i.e. gradient is computed over a sample of 64 observations from the training dataset) to train our models. For the learning algorithm, we applied an RMSprop optimization step (Goodfellow et al., 2016). The ResLogit model residual parameters are initialized using an identity matrix. Once the models have been trained, we take the best-specified model at the minimum validation loss point and compute the prediction accuracy using the validation dataset's model parameter values.

5.2. Analysis of model results

Figs. 2 and 3 reports the validation results of the MNL and ResLogit models with a baseline comparison to a MNL model (red line). A condensed version of the estimated β parameters of the MNL and ResLogit models are presented in Table 3, which we showed the comparison between our best estimated ResLogit structure (RL-16) and the MNL model. Fig. 4 shows the parameters of the first four residual layers.

5.2.1. Performance measure on out-of-sample data

Fig. 2 shows the validation curves of the model log-likelihood. The x-axis represents the iteration step, and the y-axis reports the log-likelihood. The MNL curve indicates the baseline performance where no augmentation to the utility or model. The plot on the left shows the comparison between the MNL and MLP models. This result indicates that the MLP model performs *worse* than the MNL model. The only change between the MLP and ResLogit experiments is the model structure. Therefore the improvement is most likely only attributed to the change in model structure, and not other hyperparameters[10]. MLP-2 also took twice as long to reach the maximum log-likelihood (400 vs 200 iterations on the MNL model). The MLP models (MLP-4, MLP-8 and MLP-16) produced significantly noisier output in the backpropagation step in SGD, which causes the "spikes" seen on the left plot. There were also identifiability problems with MLP-4, MLP-8 and MLP-16 models. Since the MLP-4, MLP-8 and MLP-16 models were misspecified, they could not reach the same performance log-likelihood compared to the MNL models. This result showed that adding neural network layers does not guarantee better performance and a simple MNL could potentially outperform a DNN, which is in line with our initial hypothesis.

[9] https://github.com/LiTrans/reslogit-example.

[10] It is also plausible that an MLP will do better or equivalent to a Logit model and sometimes an MLP can perform worse than a Logit model (on this particular class of problem, for example). This can be explained by the "No Free Lunch" theorem (Kawaguchi et al., 2017): "If an algorithm performs well on a certain class of problems, then it necessarily pays for that with degraded performance on the set of all remaining problems." (Wolpert and Macready, 1997, Theorem 1).

Table 2
Descriptive variables of the dataset.

variable	description	type	mean	std dev
weekend	trip on weekend	dummy variable	0.205	0.001
hour_8_10	trip between 8 am to 10 am	dummy variable	0.163	0.0015
hour_11_13	trip between 11am to 1 pm	dummy variable	0.147	0.001
hour_14_16	trip between 2 pm to 4 pm	dummy variable	0.209	0.002
hour_17_19	trip between 5 pm to 7 pm	dummy variable	0.249	0.002
hour_20_22	trip between 8 pm to 10 pm	dummy variable	0.095	0.001
hour_23_1	trip between 11 pm to 1 am	dummy variable	0.03	6e−4
hour_2_4	trip between 2 am to 4 am	dummy variable	0.006	3e−4
hour_5_7	trip between 5 am to 7 am	dummy variable	0.101	0.005
num_coord	number of trajectory links	continuous	109.8	131.23
trip_dist	trip distance (km)	continuous	8.366	10.42
trip_duration	trip duration (min)	continuous	24.04	20.97
trip_avgspeed	trip average speed (km/h)	continuous	22.503	18.815
activity	trip activity type:{1: education, 2: health, 3: leisure, 4: meal, 5: errands, 6: shopping 7: home, 8: work, 9: meeting}	categorical		
choice alternatives	1: Auto, 2: Bike, 3: Public Transit, 4: Walk, 5:Auto + Transit, 6: Other mode, 7: Other combination			

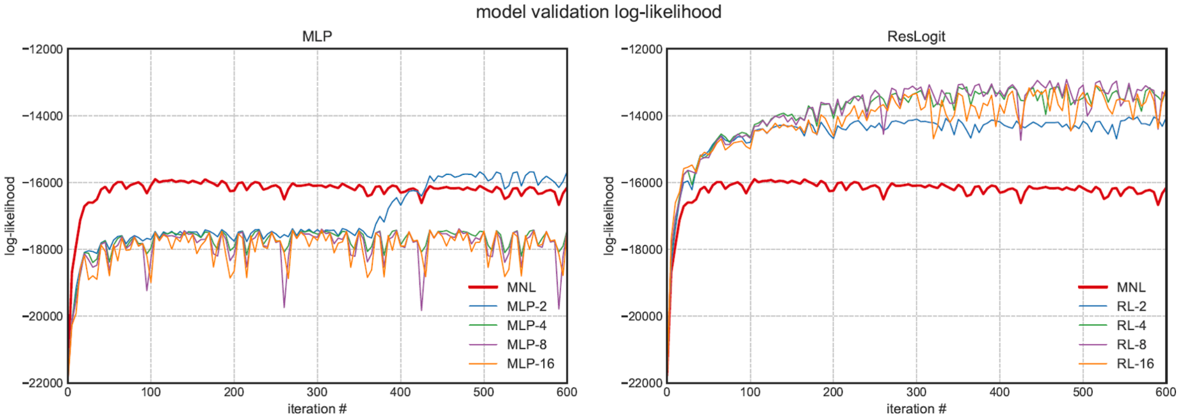


Fig. 2. Validation log-likelihood results of the model estimation.

We observed that as we increase the depth of the ResLogit models (Fig. 2, right), the log-likelihood remains consistent and outperforms the baseline MNL. Although we are using the same number of parameters and the same learning algorithm, the ResLogit method generated correctly specified models while the MLP models were misspecified. Model specification test is handled by out-of-sample validation analysis and econometric interpretation of beta parameters (explained in the following sections). We note that we did not implement any other forms of regularization for experiment consistency, e.g. L_1, L_2 regularizer or Dropout techniques. An alternative approach to model selection for more complex data where there are many unknown variables is to use a statistical measure such as the Akaike Information Criterion (AIC). The AIC statistic calculated for the MNL, MLP-16 and RL-16 models is 32566, 34902 and 28086 respectively.

Fig. 3 shows the validation error curves for both models. The error is defined as (1 - *mean prediction accuracy*) where the *mean prediction accuracy* is:

$$\mathcal{L}_n\left(i, i^*\right) = \begin{bmatrix} 1 i = i^* \\ 0 i \neq i^* \end{bmatrix} i, i^* \in \mathcal{D}_{validation} \quad (31)$$

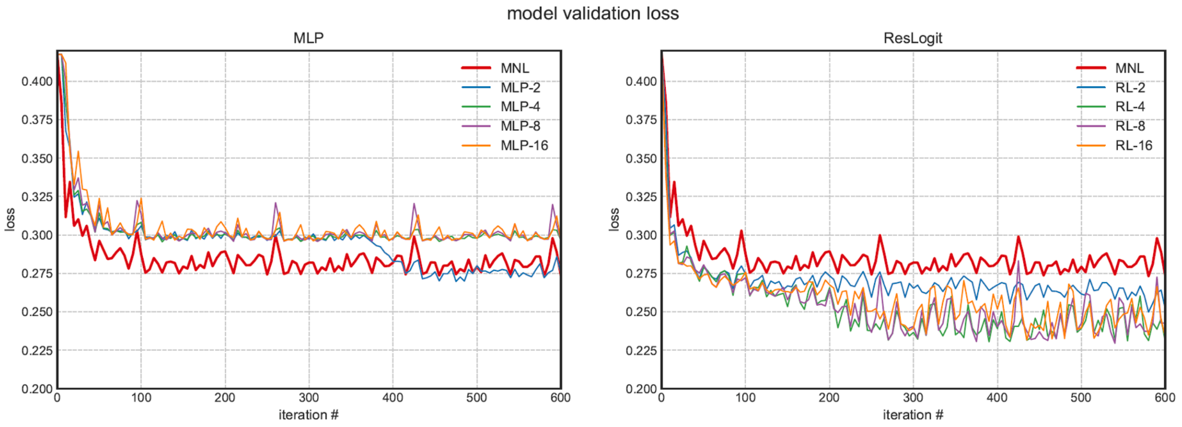


Fig. 3. Validation loss comparison between the MLP models and the ResLogit models.

Table 3
Comparison of a subset of parameter estimates between MNL and ResLogit model.

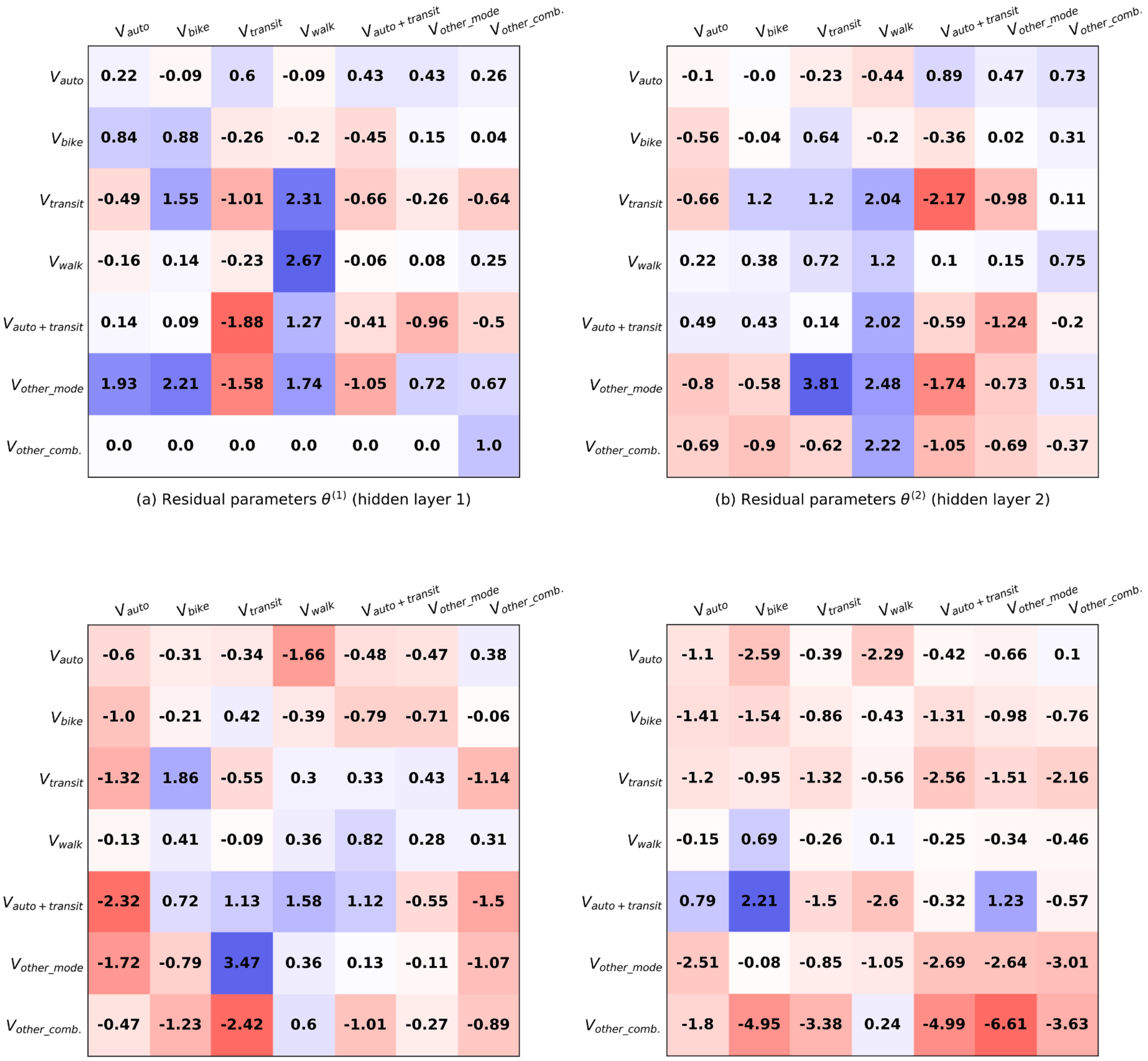
		MNL			ResLogit (16-layer)		
Parameter (β_{mj})	Choice	parameter	std. err.	rob. std. err.	parameter	std. err.	rob. std. err.
weekend	auto	−0.057*	0.036	0.386	0.045*	0.006	1.157
	bike	−0.990*	0.081	7.335	−0.448*	0.063	7.566
	transit	−0.751*	0.042	1.569	−0.090*	0.007	0.089
hour_8_10	walk	−0.841*	0.070	7.986	−1.459	0.013	0.063
	auto + transit	−2.273*	0.121	15.005	1.162	0.032	0.230
hour_11_13	bike	−0.854*	0.073	47.886	−1.210*	0.071	15.565
	auto + transit	−2.540*	0.217	48.866	1.618	0.039	0.359
hour_17_19	auto	0.058*	0.029	0.186	−0.586	0.004	0.001
hour_20_22	bike	−1.271*	0.092	16.937	−0.943*	0.085	15.009
trip_dist	auto	0.354	0.007	0.002	−0.113	0.001	0.000
	transit	0.297	0.008	0.002	0.817	0.001	0.000
	walk	−2.197	0.028	0.387	−0.257	0.004	0.001
trip_time	auto	−0.627	0.005	0.000	−0.397	0.001	0.000
	transit	0.870	0.005	0.000	0.303	0.001	0.000
	walk	0.863	0.009	0.007	−0.752	0.002	0.000
trip_aspeed	auto	0.988	0.005	0.001	−0.024	0.001	0.000
	walk	−1.738	0.014	0.058	−1.900	0.002	0.000
act_edu	auto	−1.357*	0.080	10.697	−0.187	0.011	0.055
	walk	−0.067*	0.086	22.325	−0.871	0.029	0.558
act_home	auto	−0.119*	0.026	0.151	0.340	0.003	0.001
	bike	−1.048*	0.044	3.217	−0.705*	0.039	1.477
	transit	0.109*	0.027	0.093	0.764	0.004	0.001
act_work	auto	−0.055*	0.027	0.115	0.276	0.003	0.003
	transit	−0.011*	0.028	0.096	0.631	0.004	0.004
	auto + transit	−1.853*	0.073	4.028	0.851	0.028	0.114
act_meeting	bike	−2.776*	0.259	154.812	−1.803*	0.174	106.564
log-likelihood		−16145			−13121		
sample size		42,255			42,255		
# of estimated parameters		138			922		
max. validation accuracy		72.01%			76.73%		

[*] Not statistically significant at p-value < 0.05.

$$mean\ prediction\ accuracy = \frac{1}{N_{validation}} \sum_{n=1}^{N} \mathcal{L}_n\left(i, i^*\right) \tag{32}$$

where i is the actual choice, i^* is the predicted choice, $\mathcal{L}_n(i, i^*)$ is the 0–1 loss function and $\mathcal{D}_{validation}$ is the validation dataset.

The stability of convergence shows no strong overfitting bias during the estimation process. On the MLP curves on the left plot, the model with the smallest error is the one with the least number of hidden layers but only after iteration 400, with the MNL model coming in as the second-lowest error. We can see that the error reaches a saturation point around 0.3 for MLP-2 with a negligible decrease at MLP-4 to MLP-16. This makes sense because the non-linear structure of the multi-layered neural network will be susceptible to the vanishing gradient problem observed in this figure. The results are more profound when we compare the MLP with the



(a) Residual parameters $\theta^{(1)}$ (hidden layer 1)

(b) Residual parameters $\theta^{(2)}$ (hidden layer 2)

(c) Residual parameters $\theta^{(3)}$ (hidden layer 3)

(d) Residual parameters $\theta^{(4)}$ (hidden layer 4)

Fig. 4. First 4 layers of weight matrices from the ResLogit model.

ResLogit model (Fig. 3, right). In the MLP plot, we observe that the learning gets trapped in a locally optimal point. The difference is minimal with two layers, which we expected, but a more pronounced difference between the MLP and ResLogit model when the number of layers increases. On the right plot of Fig. 3 the loss gets progressively smaller as we increase the number of residual layers, which is consistent and follows a logical pattern. Even with an RL-2, the error drops significantly faster, and the model achieved lower error than the MNL model as soon as the estimation starts. This means that neural networks are best suited to capture the error distribution rather than using it as a transformative operator on the explanatory variables.

5.3. Model coefficient estimates

Table 3 presents the coefficient estimates, standard errors and robust standard errors for the observed explanatory variables for the MNL and RL-16 model. The parameter estimates indicate the individuals' exhibited preferences for each attribute for each alternative. The results show that individuals reacted towards a stronger preference for transit when the trip time is longer in the ResLogit model, relative to the MNL model. Individuals also prefer a longer route for transit compared to auto according to the ResLogit model. In contrast, the MNL estimates show that individuals prefer a longer route when taking auto over transit. There are specific indicators which are captured in ResLogit and not in the MNL models. For instance, on weekends, people in Montreal use their car more to do shopping, recreation, visit their parents in the suburbs, go to cottage, etc. Therefore, ResLogit is giving us a positive sign for car over the weekend compared to other modes. Another example is that during morning rush hour (8–10), people commute and there is a higher chance that they take auto + transit (due to the availability of a large amount of parking at stations) to reach their office. This fact is captured only by ResLogit.

Standard errors can be calculated through the Fisher Information Matrix, requiring only the Hessian of the log-likelihood which assumes a correctly specified model. Additionally, the correct specification assumption can be relaxed by computing the robust sandwich estimator. We calculate the standard errors as a function of the negative inverse of the Hessian matrix $\mathcal{H}$, which gives the variance-covariance matrix of β, assuming those estimates are normally distributed. This value gives the Cramer-Rao bound:

$$\widehat{\Sigma}_{\beta}^{CR} = -\widehat{\mathcal{H}}^{-1} \tag{33}$$

The Hessian matrix is the second-order derivative of the log-likelihood with respect to the model parameters. Then, taking the diagonal of the square root of that variance-covariance matrix normalized by the size of the dataset, we obtain the standard errors. The robust standard error $\widehat{\Sigma}_{\beta}^{Rob.}$ is calculated by:

$$\widehat{\Sigma}_{\beta}^{Rob.} = \left(-\widehat{\mathcal{H}}^{-1}\right)\widehat{B}\left(-\widehat{\mathcal{H}}^{-1}\right) \tag{34}$$

where $\widehat{B} = \sum_{n=1}^{N}\left(\frac{\partial LL_n}{\partial\beta}\right)\left(\frac{\partial LL_n}{\partial\beta}\right)^{\top}$ In terms of the coefficient significance value, the ResLogit parameters have more parameters with a nominal p-value < 0.05 compared to the MNL model.

The standard error and robust standard error estimates show that the ResLogit estimates are more reliable than the MNL model. For the extreme cases, the parameter estimates for trip distance for walking showed the smallest value compared to other modes for both models as expected, indicating that the results are consistent. The robust standard errors also show that some parameters are not significant, for instance, *meeting activity-bike* has a high standard error when accounting for model misspecification. This is logical as travelling by bicycle is not usually common. The estimates for *hour (20–22)-bike* also indicate that this parameter is not a significant parameter, we can say that the hours between 8 and 10 pm does not impact the preference of *bike* mode.

We caution the readers that we can give no general guarantees to the precision of the standard errors or the asymptotic behaviour of the model fit for heavily biased models (Goeman et al., 2018), such as L1 or L2 regularization used in neural networks and other machine learning methods. Our ResLogit formulation reduces this bias in the model through the addition of residual layers to account for the systematic errors. Therefore, the robust standard errors that we report are reliable, but only provide an approximation of model specification correctness and the variance of the estimates.

5.4. Analyzing cross-effects from the residual matrices

The cross-effects of non-chosen alternatives are reflected in Fig. 4. The figure shows the parameters of the first four residual layers of RL-16. The matrices' values correspond to the level of dependency between the utility of one alternative with the utility function of the second alternative, and vice versa. As explained in Section 4, this matrix defines the underlying error term correlations between the choice alternatives. For example, the positive value of transit-bike in Fig. 4 (a) is 1.55. This means that the attributes of transit mode positively influence individuals who choose bike mode, increasing the utility of transit influences the increase in mode share for the bike. However, the reverse may not be identical. The value for bike-transit in Fig. 4 (a) is −0.26, indicating that increasing the utility of bike (e.g. more bike infrastructure), decreases the mode share for transit. We may relate this observation to the shared infrastructure between auto and bike. The non-zero values indicate the existence of non-linear cross-effects in the stated choices. This analysis provides an estimate of the cross-effect influence between modes of travel. Nonetheless, this experiment has shown how the ResLogit formulation uses the residual function to enhance model performance.

5.5. Elasticity analysis

The point aggregate elasticity of $P_n(i)$ with respect to input x_n is given by the following equation:

$$E_{x_n}\left(i\right) = \frac{dP_n(i)}{dx_n}\frac{x_n}{P(i)} \tag{35}$$

The elasticity measures the impact of increasing or decreasing a variable on the demand of the respective choice. In this case we use *trip_dist* as the variable and we measure the impact of market share on the *auto*, *bike*, *transit* and *walk* choices. Similarly we compute the arc elasticities of $P_n(i)$ with respect to $\widehat{x}_n$ when we change the *trip_dist* by Δx_n where $\widehat{x}_n = x_n + \Delta x_n$. Table 4 shows the point elasticities obtained from the MNL, MLP and ResLogit model (16 layers). The ResLogit model show expected signs similar to the MNL model. *Walk* mode show a smaller increase in trip distance than the MNL model, while *Transit* mode shows a more significant impact from trip distance in the ResLogit model compared to the MNL model. For the MLP model, *transit* mode show a negative sign compared to the MNL model. Surprisingly, the ResLogit model shows a different sign in *Auto* mode. Indeed, for *Auto* mode, one should expect negative elasticity.

If we analyze the two models' elasticities (presented in Fig. 5) assuming different scenarios where we increase or decrease the overall trip distance, for instance, willingness to change modes to travel a longer or shorter distance or construction of new transit networks. We can see that the elasticities from the ResLogit predict a non-linear change relation between trip distance and the respective mode choice. This shows a clear distinction from the MLP model, where the relationship between trip distance and mode shows a relatively linear curve. We expect that elasticity is heterogeneous and it will vary across different scenarios, given different unobserved trade-offs between mode choices, For *auto* mode, The ResLogit model predicts that with a decrease in trip distance by 50%, elasticity is positive (and negative otherwise), while increasing the trip distance will result in greater sensitivity to trip distance. *Bike* mode shows a positive elasticity when we increase the trip distance by 50% but a negative elasticity when we decrease the trip distance by about −50%. We can infer from this result that travellers are willing to switch from bikes to other modes or from other modes to bikes, considering other unobserved factors not captured in the data. This sign switching phenomenon is interesting because it indicates a heterogeneous population that will react differently while also *considering other alternatives*. This consideration of non-chosen alternatives shows that the ResLogit model behaves in line with the behavioural theory of the Mother Logit model example where attributes from non-chosen alternatives enter the utility of the chosen alternative.

5.6. *Significance of model depth and utility formulation*

The general notion is that increasing the complexity and non-linearity in the model should result in greater model fit, given the higher degree of freedom induced by the neural network. However, the MLP network model suffers from the vanishing gradient problem shown by increasing the number of layers. There is a bottleneck effect with depth $M \geqslant 4$, and the validation log-likelihood and loss no longer improved. In contrast, we do not see this detrimental effect in the ResLogit model, even at a depth of 16 layers. This study highlights how machine learning models may sometimes be worse off than a simple discrete choice model without understanding the neural network formulation structure.

5.6.1. *Behaviour interpretation*

As explained in Section 2.2, a decision-maker's learning process may be developed over time through experiences, and the agent updates his or her underlying distribution. The ResLogit model captures this effect while retaining the value function of the observed component of the utility. We can use this approach of capturing uncertainties to account for heterogeneity in the choice process arising from inconsistency within travel mode choice.

Besides the differences in optimal performance, it is also of practical interest to study the actual β parameter solution vectors and observe how they differ from a standard MNL model without accounting for learning behaviour. Table 3 shows the differences in β parameter estimates between the benchmark MNL and RL-16. These exact set of significant variables accounted for can be inferred from each reported metric's standard error. A conceptual step in discrete choice analysis is the ability to provide the basis of estimation of β (and standard error) and economic indications using data on observed choices and attributes. Here, our ResLogit approach follows the same approach as discrete choice methods. The unobserved attributes, expressed in ε in MNL models, captured the error contribution to the utility.

We observe that the ResLogit counterpart differs from the MNL model in most metrics. However, the ResLogit model's ability to "explain away" uncertainty yields greater parameter significance as reported by the lower standard error. The formulation of the ResLogit, which adds the g term, captures the cross-effects of the different mode choice alternatives to ensure that the decision is free from unobserved errors and endogeneity. Under regularity conditions, this residual component captures the unobserved error using a learning algorithm, similar to how in real-life, a traveller explores new route options or stick to habitual choices. In general, the ResLogit framework allowed for the error term to be formulated within the utility.

5.6.2. *Sensitivity analysis*

It is important to examine the differences in β parameter responses when changing the neural network size. Our emphasis of this analysis is on the β value significance and non-linear responsiveness when more residual layers are added to the choice model.

Table 5 shows a sensitivity analysis regarding the β parameters of trip time over time of departure. The table shows the variation between trip time and time of departure beta parameters for each model. The values represent the degree of variability of each time of departure dummy variable on the utility of each mode alternative. We take the ratio of $(\beta_{\text{trip time}} x_{\text{trip time}})/(\beta_{\text{departure dummy}} x_{\text{departure dummy}})$. This gives us the sensitivity of travel time over different departure time segments. If the parameters for $\beta_{\text{trip time}}$ are not influenced by variation in departure time, then the values would have a small standard deviation across departure time, and the standard deviation would give an indicator of uniformity of the trip time-sensitivity across different departure times. If the standard deviation is small, it would indicate that the trip time heterogeneity is captured in the residual component. The $\beta_{\text{trip time}}$ represents the value that is closer to the true mean. The attribute effects are shown in Table 5 represent the mean preference on each individual's utility, after controlling for taste variability. This result indicates the effects of increasing residual layers on the stability of the econometric parameters.

Table 4
Point elasticities.

	MNL	MLP	ResLogit
Choice	*trip_dist*	*trip_dist*	*trip_dist*
Auto	0.178	0.133	−0.103
Bike	−1.031	−0.128	−0.980
Transit	0.232	−0.206	0.669
Walk	−1.54	−0.207	−0.769

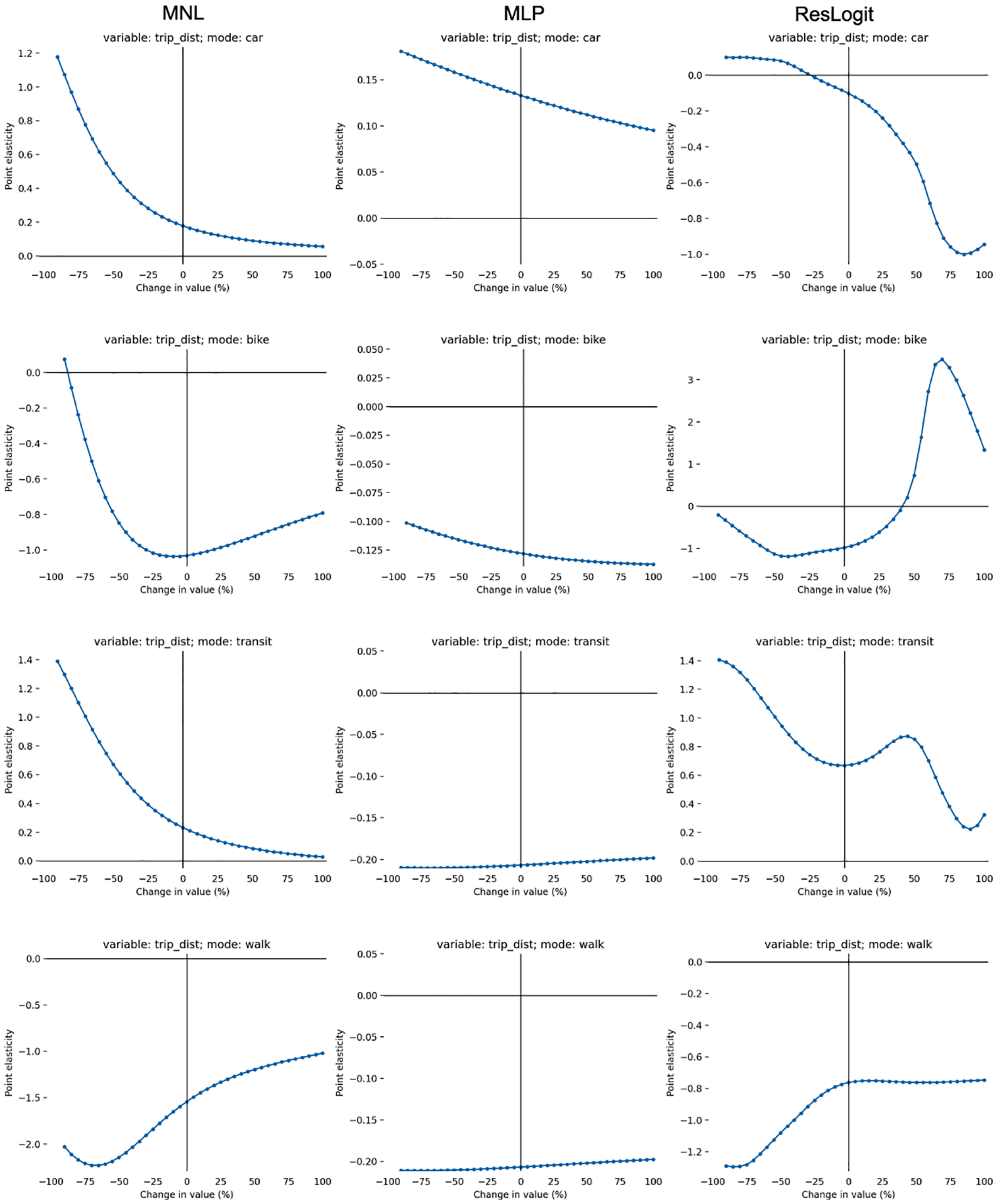


Fig. 5. Elasticity versus % increase or decrease in trip distance, comparison between models.

As expected in the MNL model, the time of departure dummy variable influences the utility and choice of mode. This is a consistent result, as we cannot represent the variation over the departure time as a single linear factor in the utility function. Modifying the MNL model by incorporating the residual layers would reduce the variability and sensitivity to time of departure. The average standard deviation of trip time versus time of departure coefficient decreases as we increase the number of layers is shown in the table. This shows how the implied heterogeneity in the utility function can be explained away through the neural network component, retaining the properties of the observed utility component. Note that this estimation does not allow us to identify the relationship between the heterogeneity of departure time and the preference of different travel modes. One can use economic indicators to estimate this effect.

Table 5
Sensitivity analysis of different travel modes over time of departure. Values show the difference in trip time parameter estimates across hourly segments.

Model	trip time/time of departure variability			
MNL	Auto	Bike	PT	Walk
hour_8_10	3.07	−0.26	26.36	−1.03
hour_11_13	−3.34	−0.24	−12.08	−5.43
hour_14_16	−2.07	−0.33	5.88	−2.61
hour_17_19	−10.81	−0.45	3.26	−1.75
hour_20_22	2.42	−0.16	−3.49	−1.31
hour_23_1	0.76	−0.14	−1.26	−0.54
hour_2_4	1.88	−0.09	−0.52	−0.43
hour_5_7	4.86	−0.16	−14.26	−0.92
stddev	4.66	0.11	11.74	1.54
RL-2				
hour_8_10	−0.18	3.63	15.37	−0.60
hour_11_13	−0.16	1.73	−5.99	−1.33
hour_14_16	−0.17	2.19	−14.66	−0.78
hour_17_19	−0.23	3.76	9.25	−0.54
hour_20_22	−0.19	1.77	−6.62	−0.83
hour_23_1	−0.29	1.65	−2.07	−0.64
hour_2_4	−0.17	1.68	−0.89	−0.94
hour_5_7	−0.17	2.29	45.38	−0.72
stddev	0.04	0.81	17.62	0.23
RL-4				
hour_8_10	0.08	0.19	0.84	−1.24
hour_11_13	0.06	0.23	1.01	−1.80
hour_14_16	0.08	0.22	1.07	−1.47
hour_17_19	0.10	0.24	1.08	−1.58
hour_20_22	0.08	0.19	0.97	−1.49
hour_23_1	0.09	0.16	0.94	−1.09
hour_2_4	0.07	0.20	1.88	−1.50
hour_5_7	0.08	0.22	0.88	−1.66
stddev	0.01	0.03	0.31	0.21
RL-8				
hour_8_10	−0.26	−1.35	0.39	−1.39
hour_11_13	−0.26	−2.89	1.42	−1.15
hour_14_16	−0.24	−1.59	0.48	−1.36
hour_17_19	−0.27	−1.58	0.41	−1.61
hour_20_22	−0.25	−2.01	0.45	−1.85
hour_23_1	−0.37	−1.46	0.34	−1.12
hour_2_4	−0.26	3.87	−2.94	−0.75
hour_5_7	−0.25	−2.14	0.38	−1.53
stddev	0.04	1.95	1.20	0.32
RL-16				
hour_8_10	0.75	−0.93	0.40	0.52
hour_11_13	0.69	−0.67	0.49	0.42
hour_14_16	0.71	−0.81	0.43	0.46
hour_17_19	0.68	−1.22	0.40	0.48
hour_20_22	0.70	−0.86	0.39	0.50
hour_23_1	0.62	−0.87	0.43	0.55
hour_2_4	0.51	−0.56	1.04	0.48
hour_5_7	0.69	−0.91	0.42	0.50
stddev	0.07	0.18	0.21	0.04

The values reported for RL-2 to RL-16 may not be entirely stable, and further investigation is needed into how the model responds to changes in hyperparameters and regularization. However, we can conclude that this experiment shows the capability of our proposed ResLogit approach, particularly in: (a) allowing for a specific analysis of the underlying distribution and (b) exploring the attributes that represent the most significant degree of heterogeneity in the model–that may present an interesting subject for future research.

6. Conclusion

This paper has presented a data-driven deep learning-based choice model that integrates a residual neural network architecture into a Logit model structure. This paper's methodological contribution is a new model that captures the learning process using neural network model structure for accounting for cross-effects in the utility error term. We proposed an approach that combines a residual neural network with a Logit model. This study's first objective resolves the shortcomings in the integration of machine learning techniques and neural networks in discrete choice modelling. The second objective addressed the systematic error of biased model estimates in DNNs due to its lack of economic interpretability.

Unlike earlier studies that only examined the performance of machine learning algorithms and their comparisons with discrete choice models in out-of-sample predictions, this paper studies the impact of a residual function in the choice utility as a data-driven variant of the Mother Logit model. The ResLogit model proposed in this paper frames the Mother Logit model's expansion function like a neural network and the parameters within the neural network are estimated through a mini-batch stochastic gradient descent algorithm, maximizing over the out-of-sample validation set. This data-driven approach also addresses model non-identifiability issues when estimating a large number of unknown parameters. A new direction to a more flexible and general model is presented using the concept of residual modelling – mapping the error term correlation to a residual function instead of using traditional neural networks. The skipped connection structure allows each residual layer to be estimated independently without model identification problems due to exploding/vanishing gradient during backpropagation.

A classic red/blue bus IIA violation example is used, and we demonstrated our methodology on a large scale travel behaviour dataset. We examined the performance comparison with an MNL and MLP neural network across a different number of layers. The results showed that with a ResLogit model, it optimized quickly and efficiently, without degradation in model performance as the number of layers increased. In the context of model identifiability, the ResLogit model yielded a smaller standard error for each econometric model parameters than the baseline MNL model. We also demonstrated the sensitivity of trip time and time of departure variability over different model characteristics. We observed that incorporating residual layers reduced model sensitivity to cross-effects and choice heterogeneity.

Our proposed ResLogit model improved discrete choice models' capabilities in terms of performance without sacrificing model interpretability. We noted that our experiment results do not consider hyperparameter tuning or regularization steps, which may affect the reliability of our model validation results. This proof-of-concept illustrates how choice modellers can leverage on deep learning methodologies and learning algorithms to enhance the current set of tools and models for discrete choice analysis. Our future work will establish additional models and extensions to our proposed ResLogit methodology.

More work has to be done on the interpretability of the model, and how to define clear guidelines so researchers without advanced knowledge of machine learning can use these new modelling techniques. Also, more comparative studies can be done between different learning algorithms for Logit models. Further investigation is also required into the meta-learning side of deep learning in discrete choice modelling. For example, we do not know the optimal hyperparameter configuration or efficiently identify a good set of hyperparameters without a tedious iterative search.

CRediT authorship contribution statement

Melvin Wong: Conceptualization, Methodology, Investigation, Validation, Writing - original draft, Writing - review & editing. **Bilal Farooq:** Conceptualization, Methodology, Validation, Formal analysis, Writing - original draft, Writing - review & editing, Supervision, Funding acquisition.

References

Alwosheel, A., van Cranenburgh, S., Chorus, C.G., 2018. Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. J. Choice Model. 28, 167–182.

Anas, A., 1983. Discrete choice theory, information theory and the multinomial logit and gravity models. Transp. Res. Part B: Methodol. 17, 13–23.

Badu-Marfo, G., Farooq, B., Paterson, Z., 2020. Composite travel generative adversarial networks for tabular and sequential population synthesis. arXiv preprint arXiv: 2004.06838.

Bansal, P., Krueger, R., Bierlaire, M., Daziano, R.A., Rashidi, T.H., 2019. Bayesian estimation of mixed multinomial logit models: Advances and simulation-based evaluations. arXiv preprint arXiv: 1904.03647.

Bansal, P., Krueger, R., Bierlaire, M., Daziano, R.A., Rashidi, T.H., 2020. Bayesian estimation of mixed multinomial logit models: Advances and simulation-based evaluations. Transp. Res. Part B: Methodol. 131, 124–142.

Ben-Akiva, M., Boccara, B., 1995. Discrete choice models with latent choice sets. Int. J. Res. Market. 12, 9–24.

Ben-Akiva, M.E., Lerman, S.R., 1985. Discrete choice analysis: theory and application to travel demand. MIT Press, Cambridge MA.

Bengio, Y., Lee, D.H., Bornschein, J., Mesnard, T., Lin, Z., 2015. Towards biologically plausible deep learning. arXiv preprint arXiv: 1502.04156.

Borysov, S.S., Rich, J., Pereira, F.C., 2019. How to generate micro-agents? A deep generative modeling approach to population synthesis. Transp. Res. Part C: Emerg. Technol. 106, 73–97.

Brathwaite, T., Vij, A., Walker, J.L., 2017. Machine learning meets microeconomics: The case of decision trees and discrete choice. arXiv preprint arXiv: 1711.04826.

Breiman, L., 2001. Statistical modeling: The two cultures. Statist. Sci. 16, 199–231.

Cantarella, G.E., de Luca, S., 2005. Multilayer feedforward networks for transportation mode choice analysis: An analysis and a comparison with random utility models. Transp. Res. Part C: Emerg. Technol. 13, 121–155.

Chorus, C.G., 2010. A new model of random regret minimization. Eur. J. Transp. Infrastruct. Res. 10.

Daganzo, C.F., Bouthelier, F., Sheffi, Y., 1977. Multinomial probit and qualitative choice: A computationally efficient algorithm. Transp. Sci. 11, 338–358.

Erlander, S.B., 2010. Cost-minimizing choice behavior in transportation planning: a theoretical framework for logit models. Springer Science & Business Media.

Fosgerau, M., Melo, E., Palma, A.D., Shum, M., 2017. Discrete choice and rational inattention: A general equivalence result. Available at SSRN 2889048.

Friston, K.J., Stephan, K.E., 2007. Free-energy and the brain. Synthese 159, 417–458.

Glorot, X., Bordes, A., Bengio, Y., 2011. Deep sparse rectifier neural networks. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, pp. 315–323.

Goeman, J., Meijer, R., Chaturvedi, N., 2018. L1 and l2 penalized regression models. Vignette R Package Penalized.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

Hagenauer, J., Helbich, M., 2017. A comparative study of machine learning classifiers for modeling travel mode choice. Expert Syst. Appl. 78, 273–282.

Hardt, M., Ma, T., 2016. Identity matters in deep learning. arXiv preprint arXiv: 1611.04231.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the 29th IEEE conference on computer vision and pattern recognition, pp. 770–778.

Hensher, D.A., Greene, W.H., 2003. The mixed logit model: the state of practice. Transportation 30, 133–176.

Hensher, D.A., Ton, T.T., 2000. A comparison of the predictive potential of artificial neural networks and nested logit models for commuter mode choice. Transp. Res. Part E: Logist. Transp. Rev. 36, 155–172.

Hess, S., Daly, A., Batley, R., 2018. Revisiting consistency with random utility maximisation: theory and implications for practical work. Theor. Decis. 84, 181–204.

Hillel, T., Bierlaire, M., Jin, Y., 2019. A systematic review of machine learning methodologies for modelling passenger mode choice. Technical Report TRANSP-OR 191025. EPFL.

Karlaftis, M.G., Vlahogianni, E.I., 2011. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. Transp. Res. Part C: Emerg. Technol. 19, 387–399.

Kawaguchi, K., Kaelbling, L.P., Bengio, Y., 2017. Generalization in deep learning. arXiv preprint arXiv: 1710.05468.

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

Lee, D., Derrible, S., Pereira, F.C., 2018. Comparison of four types of artificial neural network and a multinomial logit model for travel mode choice modeling. Transp. Res. Rec. 2672, 101–112.

Lipton, Z.C., 2018. The mythos of model interpretability. Queue 16, 31–57.

Mai, T., Bastin, F., Frejinger, E., 2017. On the similarities between random regret minimization and mother logit: The case of recursive route choice models. J. Choice Model. 23, 21–33.

Mattsson, L.G., Weibull, J.W., 2002. Probabilistic choice and procedurally bounded rationality. Games Econ. Behav. 41, 61–78.

Matějka, F., McKay, A., 2015. Rational inattention to discrete choices: A new foundation for the multinomial logit model. Am. Econ. Rev. 105, 272–298.

McFaddden, D., 1978. Modeling the choice of residential location. Spatial Interact. Theory Plan. Models 75–96.

McFadden, D., 1973. Conditional logit analysis of qualitative choice behavior. Front. Econometr. 105–142.

McFadden, D., 1975. On independence, structure, and simultaneity in transportation demand analysis. Technical Report No. 7511. Urban Travel Demand Forecasting Project. Institute of Transportation and Traffic Engineering, University of California, Berkeley.

McFadden, D., Train, K., 2000. Mixed MNL models for discrete response. J. Appl. Econometr. 15, 447–470.

McFadden, D., Tye, W.B., Train, K., 1977. An application of diagnostic tests for the independence from irrelevant alternatives property of the multinomial logit model. Institute of Transportation Studies, University of California Berkeley.

Omrani, H., Charif, O., Gerber, P., Awasthi, A., Trigano, P., 2013. Prediction of individual travel mode with evidential neural network model. Transp. Res. Rec. 2399, 1–8.

Pereira, F.C., 2019. Rethinking travel behavior modeling representations through embeddings. arXiv preprint arXiv:1909.00154.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Schuessler, N., Axhausen, K.W., 2007. Recent developments regarding similarities in transport modelling. Swiss Transport Research Conference.

Sifringer, B., Lurkin, V., Alahi, A., 2020. Enhancing discrete choice models with representation learning. Transp. Res. Part B: Methodol. 140, 236–261.

Srivastava, R.K., Greff, K., Schmidhuber, J., 2015. Training very deep networks, in: Advances in neural information processing systems, vol. 28, pp. 2377–2385.

Timmermans, H., Borgers, A., van der Waerden, P., 1992. Mother logit analysis of substitution effects in consumer shopping destination choice. J. Bus. Res. 24, 177–189.

Tversky, A., Kahneman, D., 1981. The framing of decisions and the psychology of choice. Science 211, 453–458.

Vythoulkas, P.C., Koutsopoulos, H.N., 2003. Modeling discrete choice behavior using concepts from fuzzy set theory, approximate reasoning and neural networks. Transp. Res. Part C: Emerg. Technol. 11, 51–73.

Wang, F., Ross, C.L., 2018. Machine learning travel mode choices: Comparing the performance of an extreme gradient boosting model with a multinomial logit model. Transp. Res. Rec. 2672, 35–45.

Wang, S., Zhao, J., 2019. Multitask learning deep neural network to combine revealed and stated preference data. arXiv preprint arXiv:1901.00227.

Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. Data Mining: Practical machine learning tools and techniques, fourth ed. Morgan Kaufmann, Cambridge, MA.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evolution. Comput. 1, 67–82.

Wong, M., Farooq, B., 2020. A bi-partite generative model framework for analyzing and simulating large scale multiple discrete-continuous travel behaviour data. Transp. Res. Part C: Emerg. Technol. 110, 247–268.

Wong, M., Farooq, B., Bilodeau, G.A., 2018. Discriminative conditional restricted boltzmann machine for discrete choice and latent variable modelling. J. Choice Model. 29, 152–168.

Yazdizadeh, A., Patterson, Z., Farooq, B., 2019. Ensemble convolutional neural networks for mode inference in smartphone travel survey. IEEE Trans. Intell. Transp. Syst. 21, 2232–2239.