Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis

Alwosheel, Ahmad; van Cranenburgh, Sander; Chorus, Caspar G.

DOI
10.1016/j.jocm.2018.07.002

Publication date
2018

Document Version
Final published version

Published in
Journal of Choice Modelling

Citation (APA)
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. *Journal of Choice Modelling*, *28*, 167-182. https://doi.org/10.1016/j.jocm.2018.07.002

Journal of Choice Modelling 28 (2018) 167–182

Contents lists available at ScienceDirect

Journal of Choice Modelling

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

Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis

Ahmad Alwosheel*, Sander van Cranenburgh, Caspar G. Chorus

Transport and Logistics Group, Department of Engineering Systems and Services, University of Technology, Delft, The Netherlands

A B S T R A C T

Artificial Neural Networks (ANNs) are increasingly used for discrete choice analysis. But, at present, it is unknown what sample size requirements are appropriate when using ANNs in this particular context. This paper fills this knowledge gap: we empirically establish a rule-of-thumb for ANN-based discrete choice analysis based on analyses of synthetic and real data. To investigate the effect of complexity of the data generating process on the minimum required sample size, we conduct extensive Monte Carlo analyses using a series of different model specifications with different levels of model complexity, including RUM and RRM models, with and without random taste parameters. Based on our analyses we advise to use a minimum sample size of fifty times the number of weights in the ANN; it should be noted, that the number of weights is generally much larger than the number of parameters in a discrete choice model. This rule-of-thumb is considerably more conservative than the rule-of-thumb that is most often used in the ANN community, which advises to use at least ten times the number of weights.

1. Introduction

Artificial Neural Networks (ANNs) are receiving an increasing interest from the choice modelling community to analyse choice behaviour in a variety of contexts (e.g., Hagenauer and Helbich, 2017; Hensher and Ton, 2000; Mohammadian and Miller, 2002; Van Cranenburgh and Alwosheel, 2017). This recent and profound increase in interest is due to 1) a range of recent innovations in ANN research – leading to improved performance; 2) the availability of "click-n'play" software to work with ANNs; 3) a rapid increase in computational resources, and 4) the increasing volumes and diversity of data which is at the disposal of choice modellers; this latter aspect being the core focus of the current special issue in the Journal of Choice Modelling.

To successfully train ('estimate' in choice modellers' parlance) and use ANNs, the dataset (on which the ANN is trained) needs to be sufficiently large (i.e., consist of a sufficient number of observations). In the ANNs literature such data requirements have extensively been studied (Anthony and Bartlett, 2009; Bartlett and Maass, 2003; Haussler, 1992a), leading to a series of theoretical results regarding lower bounds in terms of data size for a variety of ANNs architectures. However, these results rely on a number of assumptions which are very hard to work with in real life applications (Abu-Mostafa et al., 2012; Haussler, 1992b). As such, despite that these theoretical results are out there and perhaps because of the fact that in machine learning contexts ample of data are usually available, the ANN community – of scholars and practitioners alike – works with simple rules-of-thumb. In general, these rules-of-thumb are a factor of certain characteristics of the prediction problem. One rule-of-thumb is that the sample size needs to be at least a factor 50 to 1000 times the number of prediction classes (which, in the choice modelling context, is the choice set size) (Cho et al., 2015; Cireşan et al., 2012). Another rule-of-thumb is that the sample size needs to be at least a factor 10 to 100 times the number of

* Corresponding author.
E-mail address: a.s.alwosheel@tudelft.nl (A. Alwosheel).

https://doi.org/10.1016/j.jocm.2018.07.002
Received 15 January 2018; Received in revised form 10 July 2018; Accepted 11 July 2018
Available online 12 July 2018
1755-5345/

the features (which, in the choice modelling context, is the number of attributes) (Jain and Chandrasekaran, 1982; Kavzoglu and Mather, 2003; Raudys and Jain, 1991).[1] However, the most widely used rule-of-thumb is that the sample size needs to be at least a factor 10 times the number of weights in the network (Abu-Mostafa, 1995; Baum and Haussler, 1989; Haykin, 2009).

Despite the increasing number of applications of ANNs to analyse choice behaviour (see papers cited above, and references cited therein), to the best of the authors' knowledge no study has yet investigated the size of the data that is actually required for meaningful and reliable discrete choice analysis using ANNs. Despite the fact that emerging datasets used for discrete choice analysis tend to be relatively large, many datasets used by choice modellers typically contain somewhere between a couple of hundred and a couple of thousand observations – which is considerably smaller than those sample sizes typically used in the machine learning community. Therefore, it is important to establish what dataset sizes are in fact needed for reliable ANN-based choice modelling efforts, and whether or not conventional dataset sizes used in our community are sufficient in that regard. More specifically, it is important to establish whether the widely used rule-of-thumb to use at least 10 times the number of weights of the network also applies in the context of discrete choice analysis. A related knowledge gap addressed in this paper concerns the effect of the complexity of the data generation process (i.e., the choice model) on the required sample size. Intuitively, it is expected that the more complex (e.g., non-linear) the data generating process is, the more (choice) observations will be needed for the ANN to reliably represent the underlying DGP; but no concrete results are available as of now.[2]

This paper aims to fill the above mentioned knowledge gaps, and as such help pave the way for further and more effective deployment of ANNs for discrete choice analysis, by 1) testing whether the 'factor 10' rule-of-thumb which is used in most ANN-applications is appropriate in a discrete choice context (and if the answer is 'no', by proposing a new rule-of-thumb); and by 2) studying the relation between the complexity of the choice model's DGP and the size of the dataset that is required for meaningful, reliable discrete choice analysis using ANNs.

To achieve these two contributions to the literature, the remainder of this paper is organised as follows: Section 2 gives a brief theoretical overview of ANNs' sample size requirements, and reviews a selected number of recent applications of ANNs for discrete choice analysis. Section 3 presents a series of Monte Carlo experiments, designed to derive sample size requirements for ANN-based discrete choice analysis. Section 4 provides a cross-validation of obtained preliminary results, in the context of real empirical data. Finally, section 5 draws conclusions and presents potential directions for future research.

2. Sample size requirements for Artificial Neural Networks – theoretical considerations

ANNs are a class of machine learning algorithms that are inspired by the biological neural system. They are well-known for being highly effective in solving complex classification and regression problems (Bishop, 1995). In the context of discrete choice modelling, various comparison studies between ANNs and choice models have been conducted. For example, Hensher and Ton (2000) found that the prediction performance of ANNs is similar to a nested logit model in the context of commuter mode choice. In contrast, Mohammadian and Miller (2002) concluded that ANNs predictive power outperforms the nested logit model in the context of household automobile choice. A similar conclusion was reported by Cantarella and de Luca (2005), who trained two ANNs with different architectures to model travel mode choices. This conclusion is also confirmed by a recent study by Hagenauer and Helbich (2017), who compared many machine learning tools (including ANNs) and Multinomial Logit (MNL) to model travel mode choice.

An ANN consists of an input layer of neurons, one or more hidden layers, and a final layer of output neurons. The analyst needs to decide upon several factors such as the number of hidden layers, number of neurons at each layers, and the activation functions (see Appendix for more details and a more elaborate introduction to ANNs). Different choices of these factors result in ANNs with different levels of complexity. For example, adding more neurons to a particular hidden layer increases the capacity of the network because it has more degrees of freedom (i.e., a higher number of parameters in the network). However, it is crucial for the analyst to choose the factors so that ANN complexity is in line with the complexity of the underlying data generating process (DGP) of the problem at hand.

2.1. ANN complexity adjustment

The objective of an ANN's training process is to produce a model that approximates the underlying data generating process (DGP) based on previous observations (so-called training data) (see Appendix for more information). A successful approximation of the underlying process implies that the trained network is generalisable, meaning that it maintains a consistent performance in the available data used for training and on future data generated by the same DGP. Importantly, an ANN may fail to deliver such performance consistency if the network is excessively complex compared to the underlying data generating process. In this case, ANN performs very well on the training data, but fails to maintain a similarly strong performance on different data generated by the same DGP, which are used for validation purposes (so-called validation data). This issue is known as overfitting. Another issue that may impact the extent to which a trained ANN's is generalisable is known as underfitting, which means that the ANN is too simple compared to the underlying DGP. As a result, it performs poorly on both training and validation data. In this case, the ANN cannot

[1] Considering the fact that emerging data sets tend to be high dimensional, much effort has been devoted to optimising the data requirements by selecting the most relevant features (Blum and Langley, 1997; Ribeiro et al., 2015). Note that deep neural networks (i.e., deep learning) methods are able to process raw data and automate the feature learning step (see Goodfellow et al. (2016) for overview).

[2] Note that ANNs are capable of approximating any measurable function, given that sufficient processing neurons are available at the hidden layer and sufficient data is available for training (this property is known as Universal Approximation Theorem (Cybenko, 1989; Hornik et al., 1989)).

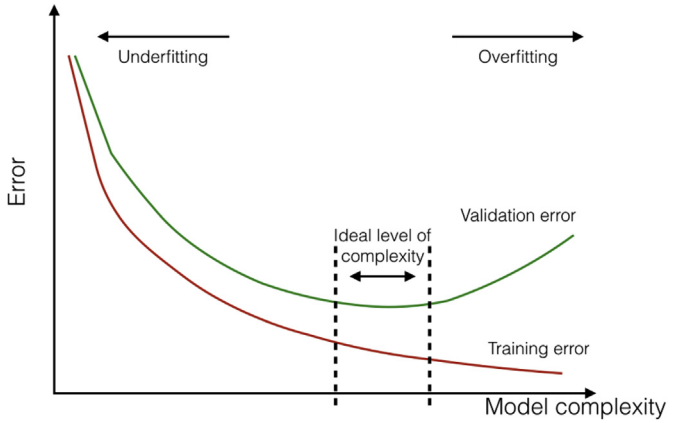


Fig. 1. A conceptual representation of the relationship between model complexity and performance. Low model complexity (compared to the underlying DGP) is represented on the left hand side: here, models perform poorly on both training and future data, as they impose too simplistic assumptions on the DGP. In contrast, very complex models are represented on the right hand side. These models perform well on the available data, but fail to obtain a similarly strong performance on validation data generated by the same DGP. The ideal level of complexity is found in the range where the validation error is low, and divergence between training and validation error (thus the vertical distance between the red and green lines) is small. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

accurately capture the relation (embodied in the DGP) between input and observed choices. In sum, it is essential for the analyst to consider the relation between complexity and performance (in the ANN-community, this relation is usually framed as a bias-variance dilemma). The above-described concepts of under- and overfitting a learning machine are shown on Fig. 1.

In this study, the ANN complexity is adjusted by adding/removing hidden neurons. For example, if the underlying DGP is complex, an ANN with very few hidden neurons (in the extreme case: only one hidden neuron) will underfit this DGP. In contrast, using large number of hidden neurons will lead to overfitting. A common approach to test for under- and overfitting is to randomly separate the available data into three subsets: one each for training, validation and testing (Ripley, 2007; Shalev-Shwartz and Ben-David, 2014). Various ANNs with different levels of complexity (i.e., different number of hidden neurons) are estimated using the training set. Then, the performance of each of the estimated ANNs is evaluated on the validation set. The network that has the best performance with respect to the validation set is selected, as its complexity falls in the ideal level of complexity range shown in Fig. 1. Subsequently, to provide an unbiased evaluation of the selected network, ANN performance is further evaluated on the testing set. If the ANN also performs well on the testing data, the analyst can be confident that the network has successfully learned the underlying DGP. The error returned by the selected network on the testing set is an approximation of the so-called generalisation error, which is the key error for assessing an ANN's learning capability, because having an ANN with low generalisation error implies that the underlying data generating process has been well approximated (Abu-Mostafa et al., 2012).[3] A pseudocode of the above-described processes can be found below.

Pseudocode 1: ANN complexity adjustment and testing

Input:
Training set, validation set, testing set, three-layers ANN
Step 1: Initialisation
M ANNs with different number of hidden neurons (different level of complexity)
Step 2: ANN performance evaluation
For m=1,2,…,M
Train ANN
Measure the performance on validation set
Choose the best performing ANN (as it has the optimum level of complexity)
Measure the performance on testing set
If satisfactory performance is obtained on testing test, ANN generalises
Output:
ANN with optimum level of complexity

[3] In some cases, training the ANN and adjusting its complexity may not result in a low generalisation error, which means that the ANN has failed to approximate the underlying DGP to a sufficient extent. One possible reason of this outcome is that the used data are insufficient in size; i.e., when trained on a very small – relative to the number of nodes in the network – dataset, the ANN may end up memorising observations rather than learning the underlying DGP. In this case, it is recommended to use larger datasets. Another possible reason behind a low generalisation error is that the data quality may be poor, e.g., there may be many outliers in the data. A remedy for this is to implement pre-processing techniques in order to limit the randomness of the data.

2.2. Theoretical measure of sample size requirements

Although this paper is intended to develop an *empirical* study of sample size requirements for ANN-powered discrete choice analysis, it is nonetheless useful to provide a brief background on theoretical contributions of the problem to the ANN-literature, and show the limited potential for practical application of these theories. As alluded to above, it is clear that the more complex the ANN, the more parameters the network consumes. And, the more parameters it consumes, the more data are needed for training the network. This intuitive relation has motivated scholars to estimate the appropriate training data size needed for reliable ANN (see papers cited in the introduction). To theoretically derive sample size requirements, a quantitative measure of ANN complexity is needed, which can be obtained from statistical learning theory. In particular, Vapnik and Chervonenkis (2015) provide a measure (known as the VC dimension) for the complexity of learning models such as ANNs. The quantification of model complexity using the VC dimension allows the statistical learning theory to provide quantitative predictions regarding the sample size requirements. The most significant outcomes in this regard are that the discrepancy between training and generalisation error is bounded from above by a quantity that grows as the model's VC dimension grows, and shrinks as the number of training examples increases (Goodfellow et al., 2016). However, despite that these outcomes provide a rigorous mathematical framework for studying data requirements, they have led to hardly any application in practice due to the prohibitive difficulty of meaningfully quantifying the VC dimension for complex learning models such as ANNs (Anthony and Bartlett, 2009; Blumer et al., 1989; Haussler, 1992a). Therefore, scientists and practitioners alike tend to follow rules-of-thumb when measuring the VC dimension for ANNs, which is then used for estimating the required sample size. The dominant rules can be summarised as follows: 1) the VC dimension of ANNs is approximately the same as the number of weights (Abu-Mostafa, 1995); 2) the sample size required to train the ANN is roughly 10 times the VC dimension (Baum and Haussler, 1989; Haykin, 2009). In sum, the size of the data that is required for meaningful and reliable ANNs is approximately 10 times the number of weights in the network (Abu-Mostafa, 1995; Baum and Haussler, 1989; Haykin, 2009).

Before we move on to the core of our paper, being the derivation and testing of rules-of-thumb for sample sizes in the context of ANN-based discrete choice analysis, we would like to note the following: ever since the introduction of ANNs, but especially in recent years (e.g., Castelvecchi, 2016), there has been debate about the 'black-box'-nature of ANNs. Indeed, compared to conventional choice models whose estimation results can be directly and meaningfully interpreted in terms of attribute-weights, elasticities and the like, the interpretability of a trained ANN's weights is very limited. Although progress is being made in this regard (see Van Cranenburgh and Alwosheel (2017) for an example in a choice modelling context), it remains the case that the use of trained ANNs is currently mostly limited to forecasting, with less to offer in terms of learning about behavioural processes. We consider attempts to 'open the black box' of ANNs and to deploy them for behavioural analyses, as very important directions for further research. However, in the present paper we do not focus on this aspect, nor do we wish to make claims about the (dis-)advantages of ANNs compared to conventional choice models. Our work in this paper is motivated by the increasing use of ANNs for discrete choice analysis, which in our view makes it important to know what sample size requirements apply in this context.

3. Sample size requirements – Monte Carlo experiments

In this section, we aim to put the 'factor 10' rule-of-thumb for sample size requirements to the test in a discrete choice analysis context, and to acquire insights into the relation between the complexity of the DGP (i.e., the choice model) and the model's sample size requirements. To do this, we conduct a series of Monte Carlo experiments, in which the true DGP varies in degrees of complexity which are observable and manageable by the analyst. Furthermore, besides studying the complexity of the DGP we also investigate the effect of random noise in the DGP (which is reflected in variations in parameter sizes, causing variation in rho-square) on sample size requirements.

3.1. Data

Table 1 presents an overview of the (synthetic) DGPs used in this section, including their parameterisations. All data sets consist of three alternatives with two generic attributes: X_1 and X_2. Each data set consists of 1000 hypothetical respondents. Each decision-maker is confronted with $T = 10$ choice tasks. Attribute levels are generated using a random number generator drawing values between zero and one. To create the synthetic observations for the Random Utility Maximisation (RUM) Multinomial Logit (MNL) DGPs, the total utility of each alternative is computed and the highest utility alternative is assumed to chosen. Similarly, for the Random Regret Minimisation (RRM) DGPs, the total regret is computed for each alternative and the minimum regret alternative is assumed to be chosen; note that we use the Pure RRM (P-RRM) model introduced in Van Cranenburgh, Guevara, and Chorus (2015), which provides the strongest possible level of regret aversion which can be attained in an RRM framework. For the Panel Mixed Logit (ML) DGPs, each respondent is assigned one draw from the associated normal distribution for each β.

3.2. ANN complexity adjustment process

In this sub-section, we present an example of how ANN complexity is adjusted in practice, following the ANN training procedure as explained in the Appendix. To avoid repetition, we only present the case for dataset A1; the same procedure applies for the other cases as well. Initially, data are randomly divided into three parts: 70% for training, 15% for validation and 15% for testing.[4] Several

[4] Note that it is possible to end-up with suboptimal ANN performance due to drawing a biased or skewed subsets. One proposed remedy for such issue is to use the so-called k-fold cross validation method. In our data, we did not find different results when using the k-fold cross validation method for ANN complexity adjustment purposes.

Table 1
Data Generating Processes and their specifications.

Data no.	DGP	Model specification	Parameterisation
A1	RUM-MNL	$V_{in} = \sum_m \beta_m x_{imn}$	$\beta_1 = -4.3$ $\beta_2 = -6.45$ $\rho^2 = 0.50$
A2	RUM-MNL	$V_{in} = \sum_m \beta_m x_{imn}$	$\beta_1 = -2.85$ $\beta_2 = -4.28$ $\rho^2 = 0.35$
A3	RUM-MNL	$V_{in} = \sum_m \beta_m x_{imn}$	$\beta_1 = -1.84$ $\beta_2 = -2.75$ $\rho^2 = 0.20$
B1	RUM-ML	$V_{in} = \sum_m \beta_m x_{imn}$	$\beta_1 \sim N(-4.44, 1)$ $\beta_2 \sim N(-6.66, 1)$ $\rho^2 = 0.50$
B2	RUM-ML	$V_{in} = \sum_m \beta_m x_{imn}$	$\beta_1 \sim N(-3.07, 1)$ $\beta_2 \sim N(-4.61, 1)$ $\rho^2 = 0.35$
B3	RUM-ML	$V_{in} = \sum_m \beta_m x_{imn}$	$\beta_1 \sim N(-2.02, 1)$ $\beta_2 \sim N(-3.02, 1)$ $\rho^2 = 0.20$
C1	P-RRM-MNL	$R_{in} = \sum_m \beta_m \tilde{x}_{imn} where\ \tilde{x}_{imn} = \sum_{j \neq i} \max(0,\ x_{jmn} - x_{imn})$	$\beta_1 = -2.88$ $\beta_2 = -4.32$ $\rho^2 = 0.50$
C2	P-RRM-MNL	$R_{in} = \sum_m \beta_m \tilde{x}_{imn} where\ \tilde{x}_{imn} = \sum_{j \neq i} \max(0,\ x_{jmn} - x_{imn})$	$\beta_1 = -1.83$ $\beta_2 = -2.74$ $\rho^2 = 0.35$
C3	P-RRM-MNL	$R_{in} = \sum_m \beta_m \tilde{x}_{imn} where\ \tilde{x}_{imn} = \sum_{j \neq i} \max(0,\ x_{jmn} - x_{imn})$	$\beta_1 = -1.13$ $\beta_2 = -1.69$ $\rho^2 = 0.20$

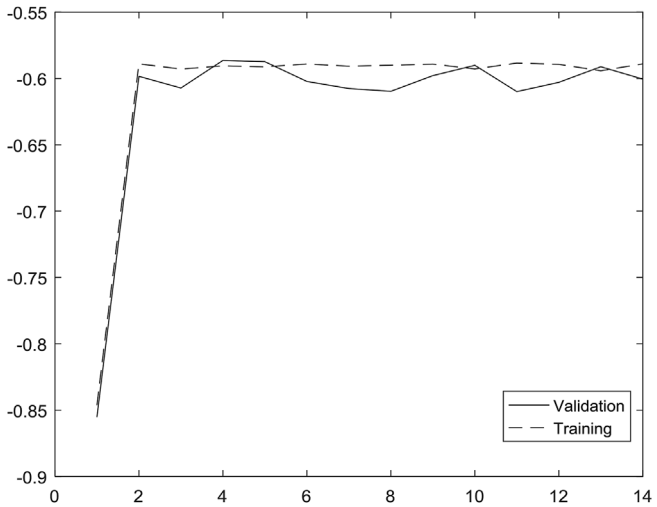


Fig. 2. Number of ANN hidden neurons vs average Log-Likelihood values for RUM-MNL data.

ANNs with different levels of complexity are subsequently created. These ANNs are then trained on the training data. Fig. 2 shows the relationship between ANN complexity (i.e., the number of hidden neurons) and the Log-Likelihoods (averaged across observations) obtained on both the training and validation set. The network that provides the best performance on the validation data is then selected. Fig. 2 shows that four hidden neurons provide the best performance (on the validation set). Using more than four hidden neurons does not affect the resulting Log-Likelihood, implying that ANN has learned the input/output relationship with four neurons.[5]

When complexity of the underlying DGP is increased, ANNs with more hidden neurons are needed. For example, our analysis

[5] According to Occam's razor principle, an explanation of a set of data should be limited to the bare minimum that is consistent with the data. In Fig. 2, increasing the complexity does not result in better performance. Therefore, the simplest model that describe data is preferred, which in this case is an ANN with four neurons.

shows that the optimum number of hidden neurons for the more non-linear and 'complex' (as it involves a series of max-operations and pairwise comparisons in the regret function) RRM-MNL data is eight, constituting a doubling compared to a linear-in-parameters RUM model (see Table 2 for results for all DGPs).[6] Once the network that provides the best performance is obtained, the number of weights in the network can be observed accordingly.

3.3. Resulting ANN sample size requirements

To assess, in the context of the testing data, whether the ANN has been trained on a number of choice observations that is large enough to enable a sufficiently accurately learning of the underlying DGP, several approaches have been introduced in different contexts and applications (Cho et al., 2015; Figueroa et al., 2012; Mukherjee et al., 2003; Sung et al., 2016). In our study we determine the sample size required for accurately learning of the underlying DGP based on the learning curve. More specifically, we inspect the gradient of the learning curve – which represents the size in improvement of the ANN's prediction performance as more training data sets are used. The intuition behind an ANN's learning curve is straightforward: as more observations are used to train the network, a better prediction performance is obtained until the learning curve reaches a saturation point where its learning rate slows and its gradient starts to approach zero, implying that the size of the training dataset has been sufficient for the ANN to learn the DGP (Cortes et al., 1994; Kohavi, 1995). We consider the ANN to have successfully learned the underlying DGP if the gradient of the learning curve is less than 10^{-5}.

Furthermore, given that –in this subsection– we deal with synthetic data, we can also inspect the deviation between the prediction performance of the ANN and the best possible prediction performance (note that no model is capable to outperform the true DGP). Hence, in the context of synthetic data there is a theoretical and observable upper limit of the prediction performance, which is embedded in the true DGP. So, as a cross-check of the learning curve gradient criterion mentioned above, we inspect the difference between the ANN prediction performance and the theoretical upper limit.

Fig. 4 presents the ANN learning curves for the data sets described in Table 1. For each data set, it shows the impact of training data size (on the x-axis) on the ANN prediction performance (using metrics presented in Appendix. 3) on testing data (y-axis). We present results for both the Log-Likelihood-based measure (top-panels) and the Hit-Rate-measure (bottom-panels); note that while the Hit-Rate measure is popular in the ANN-community, but is only occasionally used in the field of choice modelling, in generally not being recommended).[7] For each data set, we fitted a power function of the form $y = ax^b + c$. Based on this fitted function the gradient is determined.

Note that each data point represents the performance of ANNs trained using k-folds cross validation method, to avoid presenting the result of a particular manifestation of the randomness in the data generating process (Abu-Mostafa et al., 2012). The notion of k-folds cross validation methodology is to partition the data into k equal sized subsamples. A single subsample is then used for testing and the remaining $(k-1)$ are used for training. This process is repeated k times, where each of the k subsamples used only once for testing. The resulted ANN performances are averaged and reported. Also, note that to reflect the difference in levels of noise represented in the underlying DGPs, the ANN performance is normalised with respect to the associated theoretical upper limit. For the Log-Likelihood (LL) measure, ANN prediction performance is normalised as follows:

$$1 - \frac{LL_{ANN}}{LL_{max}} \tag{1}$$

And for the Hit-Rate (classification accuracy) measure, the following normalisation applies:

$$\frac{HitRate_{ANN}}{HitRate_{max}} \tag{2}$$

Note that two vertical lines represent the data requirements according to: 1) the factor 10 requirement that is the widely adopted rule-of-thumb in the ANN-community (i.e., the data required for ANN training is 10 times the number of weights in the network); 2) the sample size requirement according to the criterion of successful learning mentioned above. For all cases, the difference between the theoretical upper limit and the ANN prediction performance (that has been trained on data of the proposed size) is less than 10%, indicating the strong prediction performance achieved by the ANN. To facilitate inspection of the figures, we only draw this second vertical line for the least noisy DGP within a particular category of DGPs. Results are summarised in Table 2.

Finally, to put our findings in yet more perspective, we also compare the results obtained using the learning curve approach with a recently proposed methodology for big data applications (not focusing on discrete choice analysis-contexts) known as the Critical Sampling Size (CSS) heuristic (Ribeiro et al., 2015; Sung et al., 2016). The CSS heuristic method aims to find the absolute minimal number of observations required to ensure that a learning machine meets a desirable performance (Silva et al., 2017). The first step of

[6] Note that different ANN structures (i.e., different number of hidden layers, different activation functions) have been also implemented for this study. We found that adding more hidden layers did not improve prediction performance. Also, we found that using different activation functions for shallow ANN did not result in a different prediction performance. As such, due to space limitations and for the ease of communication, we choose to focus in this study on the single hidden-layer ANN.

[7] Particularly in a Marketing context, Hit-Rates are often used to assess a choice model's empirical performance (e.g., Huber and Train, 2001; Kalwani et al., 1994; Neelamegham and Jain, 1999). However, its use has been criticized for failing to accurately represent the probabilistic nature of choice models (e.g., Train, 2009). In this paper, we do not wish to express a strong opinion on this matter, but we do note that the mainstream in choice modelling attaches far more importance to likelihood-based measures of model performance than 'correct classification'-based metrics.

the CSS heuristic method is to partition the data into k clusters. Then, m randomly sampled data-points are selected from each cluster (m is initially set to be fairly small) to form a training data set of size mk. If the performance of the trained ANN (on a separate testing dataset) exceeds a pre-defined threshold value T, then the training data size is considered sufficient for the ANN. Otherwise, the process of sampling is repeated with larger value of m, until a satisfactory performance is achieved. For a more extensive description of this method see Silva et al. (2017). In the context of this study, we set T to be 2% less than the ANN prediction performance (in terms of Log-Likelihood measure) when it has access to the whole dataset.

The CSS heuristic method is executed 50 times. Fig. 3 shows a histogram of the frequency of sample size requirements across 50 runs for dataset A1, which follows a seemingly normal distribution. We can notice that once the ANN has access to a sample size of 2,000, more than two thirds of the 50 runs obtained a performance that exceed the defined threshold T. Further, around half of the 50 runs provide a satisfactory performance with a training data of size 2000 (see Fig. 3). In this case, we report that the ANN data requirements is 2000. Results for all datasets are shown in Table 2.

Table 2
ANN data requirement for synthetic data.

DGP	Rho-square	Hidden nodes	Number of ANN parameters	Data requirement based on 'factor 10' rule of thumb	Data requirement based on the learning curve gradient method	Factor implied by the learning curve gradient method	Factor implied by the CSS heuristic method
(A1) RUM-MNL	0.50	4	43	430	2200	54	47
(A2) RUM-MNL	0.35	4	43	430	2000	47	42
(A3) RUM-MNL	0.20	4	43	430	2000	47	38
(B1) RUM-ML	0.50	5	53	530	2600	50	46
(B2) RUM-ML	0.35	5	53	530	2200	42	42
(B3) RUM-ML	0.20	5	53	530	1800	34	34
(C1) P-RRM-MNL	0.50	8	83	830	3000	37	37
(C2) P-RRM-MNL	0.35	8	83	830	2400	29	32
(C3) P-RRM-MNL	0.20	8	83	830	1800	22	27

3.4. Interpretation of results, and discussion

Based on these results, we are able to establish a number of important observations: first, looking at the ANN learning curves, it is directly seen that for all decision rules the training data size requirement imposed by the 'factor 10' rule of thumb is not conservative enough, especially when considering the Log-Likelihood-based measure of evaluation (which is used considerably more often in the choice modelling field than the Hit-Rate). Clearly, ANN performance significantly enhances as the network has access to larger training dataset, i.e., beyond the size which is advised by the 'factor 10' rule-of-thumb. Table 2 shows the factor (i.e., the ratio between required number of training observations and the number of weights in the network) which is implied when one considers the proposed requirements; it varies, across DGPs, between 22 and 54. Furthermore, the factors obtained using the CSS heuristic methodology are within the same range (see Table 2, last column). Therefore, to be on the safe side, these results – based on synthetic data – suggest the following rule-of-thumb when using ANNs to analyse discrete choice data and when considering Log-Likelihood-

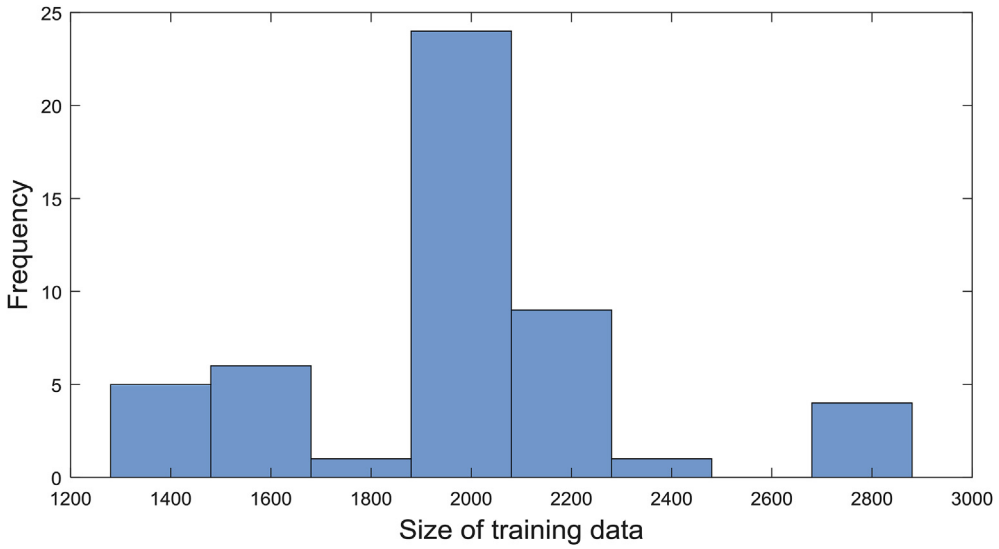


Fig. 3. Frequency of sample size requirements using heuristic CSS method.

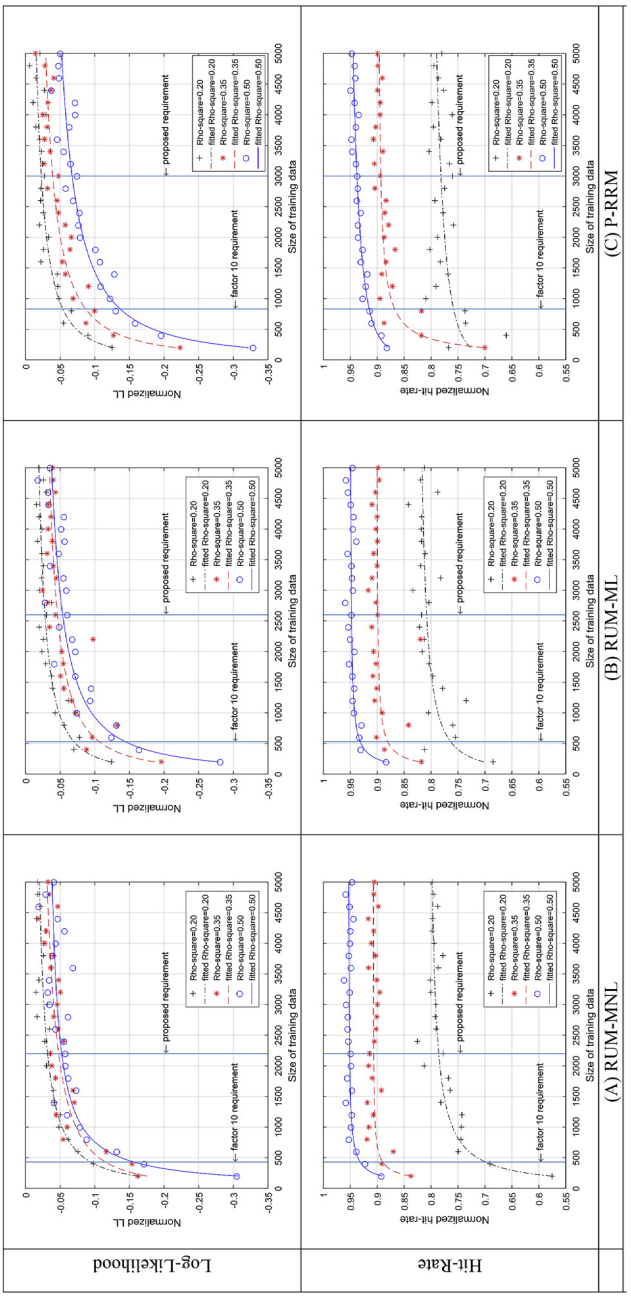


Fig. 4. ANN prediction performance on synthetic datasets.

Table 3
Description of real data sets.

	Reference	Number of choice observations	Number of alternatives in the choice set	Number of attributes per alternative
Data set 1	(Bierlaire et al., 2001)	9036	3	2
Data set 2	(Chorus and Bierlaire, 2013)	3510	3	4
Data set 3	(Hague Consulting Group, 1998)	17787	2	2

based measures as the appropriate standard for model evaluation: the number of observations in a training dataset needs to be at least 50 times larger than the number of weights in the network to enable a sufficient performance.

Another (and at first sight possibly counterintuitive) point worth noting concerns the effect of the level of noise in the DGP on ANN sample size requirements. Our analysis shows that the ANN requires more training observations, as the DGP becomes less noisy. One likely interpretation of this finding is that as the level of noise in the DGP decreases, the data contains more information worth learning by the ANN. Hence, the network requires more effort (i.e., more data for training) to extract the information.

In sum, from the Monte Carlo experiments we learn that the complexity and the 'noisiness' of the underlying DGP both have an impact on the minimum number of observations required to train an ANN. A 'factor 50' rule of thumb seems to be appropriate and the commonly used 'factor 10' rule of thumb seems too 'optimistic' (especially when using a Log-Likelihood-based metric for model evaluation as is the standard in the choice modelling community). Finally, these results on synthetic data suggest that ANN data requirements are by and large within the range of common dataset sizes used in choice modelling.

4. Sample size requirements – real data

In this section, we aim to extend our analysis of ANNs data requirements for choice modelling, beyond synthetic data towards several real data sets that have been extensively reported in the choice modelling literature. A brief description of the used data sets can be found in Table 3. To assess whether the ANN has been trained on a sufficient amount of data, the criterion reported in subsection 3.3 is used. That is, we consider an ANN to have sufficiently accurately learned the underlying DGP once the gradient of the learning curve is less than 10^{-5}. Note that, unlike the Monte Carlo experiments, the true DGP is obviously unknown for these datasets, and consequently the theoretical upper limit prediction performance – which we used to cross-check the derived sample size in the previous section – cannot be determined in this context.

Fig. 5 shows the ANN learning curves for each of the data sets described in Table 3. As in previous plots, for each data set, the impact of training data size (depicted on the *x*-axis) on two aspects of the ANNs prediction performance is shown: average Log-Likelihood and classification accuracy (Hit-Rate). Note that two vertical lines represent the data requirements according to: 1) the 'factor 10' rule-of-thumb commonly used in the ANN literature, and 2) the data requirements based on the proposed learning curve gradient criterion. Note also that smaller subsets of the full dataset were obtained by randomly removing observations from the mother-dataset. Finally, the sample size requirements obtained using learning curve gradient are compared with those obtained using the CSS heuristic method. A summary of results is shown in Table 4.

The results confirm the insufficiency of the data requirements based on the 'factor 10' rule of thumb: for all data sets, Fig. 5 shows a clear pattern of attaining better predictive performance when the network is trained on larger data sets. Based on the learning curve gradient condition, dataset sizes implying a factor of 27 to 31 times the number of weights in the network appear to be sufficient. Further, the factors obtained using the CSS heuristic method are within the same range (see Table 4, last two columns).

5. Conclusions and recommendations

This study contributes to the rapidly growing literature which focuses on using artificial intelligence (machine learning) techniques for discrete choice analysis, by investigating the size of datasets which is required for reliable representation of discrete choice models using Artificial Neural Networks (ANNs). In particular, using synthetic datasets, we study the sample size that is required for Data Generating Processes with different levels of complexity and 'noisiness'. In addition, we analyse dataset size requirements for ANN-based discrete choice analysis, based on several real data sets that have been used in the literature. For each data set, the complexity of the ANNs (which ultimately determines the required sample size) is optimised using validation methods commonly used in the artificial intelligence (machine learning) community. Using the concept of a learning curve, we are able to establish the number of observations that an ANN needs to obtain a reliable and strong predictive performance on out-of-sample data. Based on our analyses, we are able to draw the following conclusions and recommendations concerning data requirements for ANN-based discrete choice analysis.

First: data requirements based on the 'factor 10' rule-of-thumb which is widely-adopted in the ANN literature appear to be insufficient if one wants to evaluate model performance in terms of Log-Likelihood-based measures (as is the norm in most of the

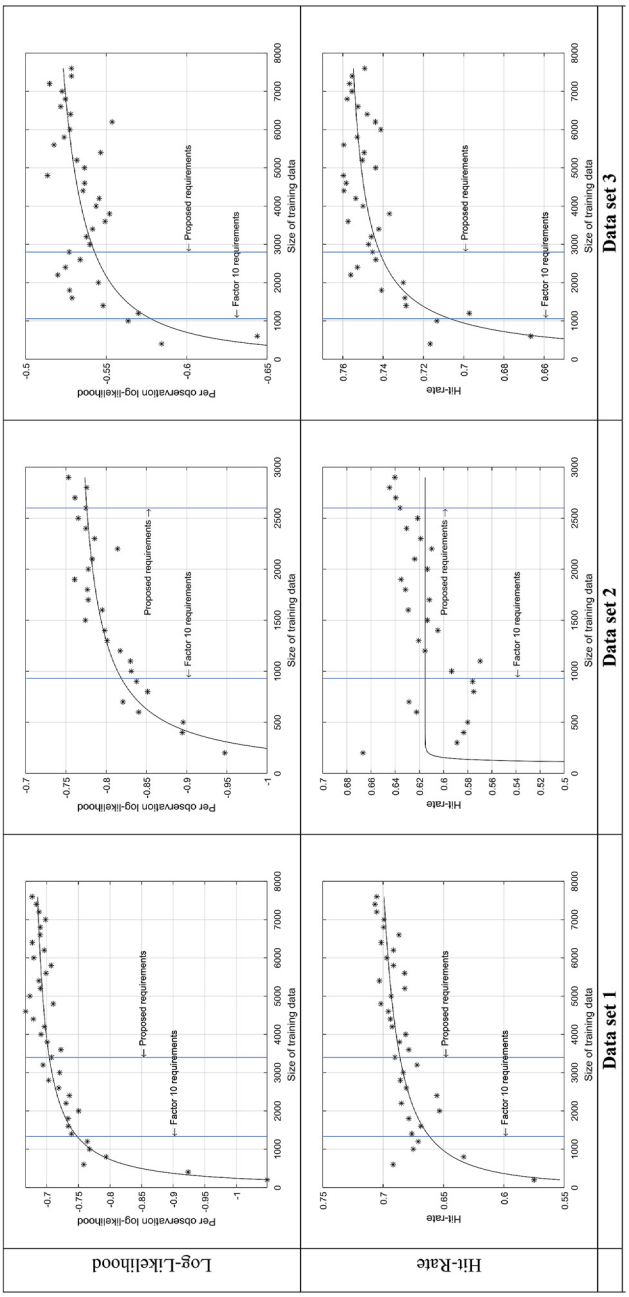


Fig. 5. ANN prediction performance for real data.

Table 4
ANN data requirements: real data.

Data set	Hidden nodes	Number of ANN parameters	Data requirement based on 'factor 10' rule of thumb	Data requirement based on the learning curve gradient method	Factor implied by the learning curve gradient method	Factor implied by the CSS heuristic method
Data set 1	10	133	1330	3400	31	28
Data set 2	5	93	930	2600	28	25
Data set 3	8	106	1060	2800	27	36

choice modelling community). Based on inspecting results for synthetic and real data sets, and to be conservative, we propose to use a 'factor 50' rule of thumb (i.e., the number of observations needs to be at least 50 times the number of adjustable parameters in the network; where it should be noted that the number of adjustable parameters in an ANN is generally much higher than the number of parameters in a corresponding choice model). If one aims to evaluate model performance on terms of Hit-Rate – or: correctly classified – based metrics, smaller data sets may be used (which may explain the popularity of this rule-of-thumb in the machine learning literature which generally uses Hit-Rates for model evaluation). But also in that case, our analyses suggest that the 'factor 10' rule-of-thumb appears somewhat too 'optimistic'. Second, as an important side result we find that the ANN requires more data as the complexity of the DGP increases and its noisiness decreases. Third, our analysis shows that ANN sample size requirements are roughly within the range of most data set sizes encountered in the field of choice modelling. This finding suggests that indeed there is ample opportunity for using ANNs to analyse discrete choice data, also on existing data sets but particularly so on emerging 'Big-'datasets. Note that these conclusions are derived from shallow ANNs trained using back-propagation approach. We acknowledge that there are various types of ANNs models (i.e., different network structure, activation functions, etc.) that we haven't examined in this study. However, this provides an avenue for further research in the near future.

As a final note, we wish to re-emphasise that the required sample size for ANN-based (discrete choice) analysis depends on the complexity (i.e., number of neurons) of the ANN. Since the complexity of the ANN cannot be determined in advance – see section 2 for a description of the iterative procedure used to determine the optimal number of neurons – this implies that sample size requirements can only be determined after 'estimation'. Three approaches are suggested in this regard: first, the analyst may indeed determine ex post if the sample used for training the ANN has in fact been large enough. Second, the analyst may use a prior study to determine the optimal number of neurons in the ANN, and based on that choose the sample size for the core study.[8] Third, the analyst may build on past work reported in the literature to ex ante guess the likely number of neurons needed in the ANN, and work from there. Note that this approach is quite similar to common practice in classical choice modelling, where minimum sample sizes needed to obtain significant parameters can only be determined ex post, or based on prior parameters which can be based on literature or on pilot studies.[9]

Statement of contribution

Artificial Neural Networks (ANNs) are increasingly being used to analyse choice behaviour. For these problems, it is important to establish the amount of data required to ensure that a network provides a reliable and meaningful discrete choice analysis. This paper is the first to do so.

It contributes to the literature by establishing a new rule-of-thumb for the sample size requirements when using ANNs-based choice behaviour analysis. It does so by studying the ANNs performance for several synthetic datasets based on data generating processes with different levels of complexity, as well as on several real datasets. To capture whether the network has been estimated on sufficient sample size, we propose to use the learning rate which depicts the improvement in the ANN performance when increasingly large datasets are used. As the learning rate approaches zero, it indicates that no further improvement in performance of the ANN is achieved when more data are used, implying that the associated dataset size is of sufficient size.

Based on our analysis we establish a new rule-of-thumb for ANN-based discrete choice analysis which is considerably more conservative than the dominant rule of the ANN-literature: we advise a minimum sample size of fifty (as opposed to ten, the widely reported rule in ANNs literature) times the number of estimable parameters in the ANN. We also establish that more complex and less noisy data generating imply a need for larger datasets.

Acknowledgments

The authors would like to thank King Abdulaziz City for Science and Technology (KACST) for supporting this work.

[8] For highly complex problems (e.g., image processing problems), deep networks are commonly used. Due to the large number of weights and the computing power required for training them, it is commonly practiced to use pre-trained networks, where the structure (i.e., number of hidden layers, and number of neurons in each layers) and the weights' values are used (see for example Vedaldi and Lenc, 2015). The network is then trained on the newly presented data.

[9] Sample size requirements have been investigated for Stated Preference (SP) and Revealed Preference (RP) data. Just like machine learning practitioners, SP practitioners have developed several rules-of-thumb. For example, McFadden (1984) proposed that a sample size of thirty responses per alternative. Another widely used rule, which is a mirror-image of the developed rule in this study, is to have at least 30 times the number of adjustable parameters (see Rose and Bliemer (2013) for overview). For RP data, Hensher et al. (2005) proposed to have a minimum sample sizes of 50 decision maker choosing each alternative.

Appendix A. Artificial Neural Networks – An overview

ANNs consist of highly interconnected processing elements, called neurons, which communicate together to perform a learning task, such as classification, based on a set of observations. Fig. A1 shows the layout of the neuron structure.

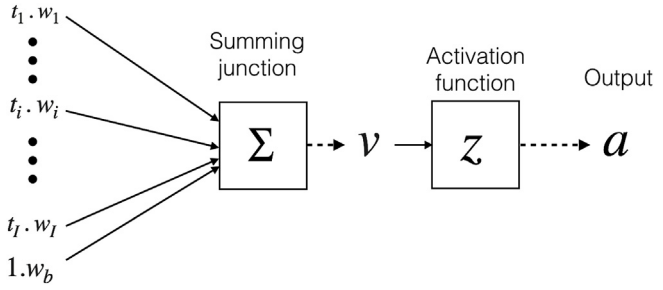


Fig. A1. A neuron layout.

Each neuron in the network receives inputs (t_i) multiplied by estimable parameters known as weights (w_i). The weighted inputs are accumulated and added to a constant (called bias, denoted b) to form a single input v for a pre-defined processing function known as activation function $z(.)$. The bias has the effect of increasing or decreasing the net input of the activation function by a constant value, which increases the ANNs flexibility (Haykin, 2009). The activation function $z(.)$ generates one output a that is fanned out to other neurons. The output a can be described as follows:

$a = z(v) = z(\sum_{i=1}^{I} w_i * t_i + w_b)$, where w_b is the weight associated with the bias.

The neurons are connected together to form a network (Bishop, 2006; LeCun et al., 2015). A widely used ANN structure consists of layers of neurons connected successively, known as multi-layer perceptron (MLP) structure. Typically, the first (input) layer and the output layer depend on the problem at hand. More specifically, input layer neurons represent the independent variables. In the context of choice modelling, these are the alternatives' attributes, characteristics of decision-makers, and contextual factors. The output layer, in a discrete choice context, consists of neurons that provide choice probabilities P for each alternative. Layers in-between are called hidden layers because their inputs and outputs are connected to other neurons and are therefore 'invisible' to the analyst. For illustrative purposes, consider the following hypothetical situation: a person can travel using one of three modes: bus, train, or car; two attributes (travel cost "TC" and travel time "TT") are associated with each alternative. Fig. A2 shows this typical choice situation in a three-layer MLP network with four hidden neurons.

Neurons at the hidden and output layers are represented by circles in Fig. A2, while input and bias neurons are represented by squares. This is to emphasise that the neurons at the hidden and output layers are processing units, meaning that they receive inputs t and return outputs a according to predefined activation function $z(.)$, as illustrated in Fig. A1. Input neurons pass the input signals to the next layer. In Fig. A2, the ANN has a total of 7 processing units.

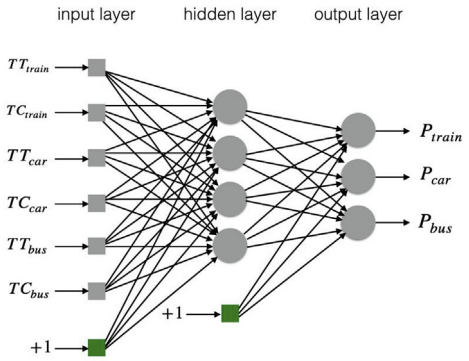


Fig. A2. Three-layers Artificial Neural Network.

A.1. ANN specifications

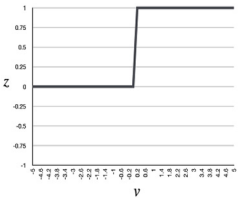
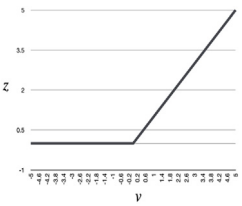
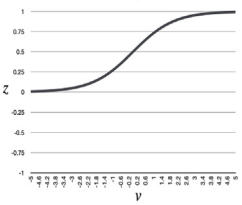
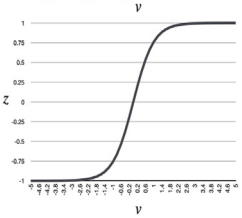
For a complete MLP structure, three elements need to be defined:

1) Number of hidden layers: a commonly used structure is three-layers MLP: input, output and one hidden layer. A key property of this structure lies in the ability to approximate, with arbitrary level of precision, any measurable function given that a sufficient number of processing neurons are available at the hidden layer; this property is known as the Universal Approximation Theorem (UAT) (Cybenko, 1989; Hornik et al., 1989). The three-layer MLP structure is considered and discussed in more detail further on.
2) Number of neurons for the hidden layer(s): the UAT holds true only if a sufficient number of hidden neurons are available.

Intuitively, ANNs with more hidden neurons have more free parameters (w) and are therefore capable of learning more complex functions.

3) Activation function $z(.)$: As mentioned before, each neuron processes its input via a pre-defined activation function. Neurons at the same layer usually employ identical functions. Examples of commonly used functions in the hidden layers are presented in Table. A1. In the analyses presented in the remainder of this paper, a tangent sigmoidal function has been employed at the hidden layer neurons, as it has been shown to lead to fast training times (LeCun et al., 2012). For the output layer, a so-called softmax function is used (which is essentially a logit) to ensure that the sum of the choice probabilities equals one.

Table A1
Activation functions.

	Activation Function Name	Function	Plot
a	Step Function	$z = \begin{cases} 0 & v \leq 0 \\ 1 & v > 0 \end{cases}$	
b	Rectifier Linear Unit (ReLU) Function	$z = max(0, v)$	
c	Sigmoid Function	$z = \frac{1}{1+\exp(-v)}$	
d	Tangent Sigmoidal Function	$z = \tanh(v)$	

An analyst sets these three elements according to the desired objective of the modelling effort. For example, adding two or more hidden layers serves to create a deep learning network, which has been shown to lead to breakthrough results in fields such as image classification (e.g., Krizhevsky et al., 2012). In this paper, for reasons of ease of communication and without loss of generic applicability, we limit our focus to the so-called shallow network version of the ANN (i.e., an ANN with single hidden layer). The complexity of such network is adjusted by adding or removing neurons at the hidden layer (called hidden neuron). It is crucial for learning to adjust the number of hidden neurons so that ANN complexity matches the problem at hand (i.e., the underlying DGP). An example that shows how to adjust the number of hidden neurons is presented in subsection 2.1.

A.2. ANN Training

In the discrete choice modelling context, the process of finding values of the model's parameters (w) is known as estimation. In this study, we comply with the language of machine learning community and call it training. The choice data used for training the ANN consists of a set of observations $S = ((\mathbf{x}_1, \boldsymbol{y}_1), (\mathbf{x}_2, \boldsymbol{y}_2), ..., (\mathbf{x}_n, \boldsymbol{y}_n), ..., (\mathbf{x}_N, \boldsymbol{y}_N))$. Each nth observation s_n contains a vector of independent variables $\mathbf{x}_n$ that represent the attributes and a K-dimensional vector of dependent variables $\boldsymbol{y}_n$ that represent the observed choice (i.e., zeros for the non-chosen alternatives, and a one for the chosen alternative); K being the size of the choice set. Since choices are mutually exclusive (i.e., only one alternative can be chosen from the choice set), from a machine learning perspective this is considered a classification problem.

The central goal of ANN training is to model the underlying data generating process (DGP) that has led to the current set of observations, so that the best possible prediction for future observations is achieved (Bishop, 1995). While to estimate the parameter of a choice model the likelihood function is maximised, for ANN training an equivalent so-called error function $J(\mathbf{w})$ is minimised. We define $\mathbf{w}$ as a vector that contains the ANN estimable parameters w. Assuming the data consist of N choice observations across K alternatives, the error function is defined as follows:

$$J(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk}\ln(P_{nk}) \tag{A1}$$

Where y_{nk} is an indicator which denotes whether alternative k is chosen in observation n, and P_{nk} is the choice probability predicted by the ANN, which is a function of $\mathbf{w}$ and $\mathbf{x}$. To avoid unnecessary semantic confusion, the function in Equation ($A1$) is called negative Log-Likelihood function in the rest of this document.

By training the ANN, the analyst's objective is to find the weight vector $\mathbf{w}$ such that $J(\mathbf{w})$ is minimised, by means of searching through the parameters' space in successive steps. At each step, $J(\mathbf{w})$ is decreased by adjusting the parameters in $\mathbf{w}$. The well-known gradient descent approach is the most widely applied algorithm for this purpose. In short, this process of training an ANN can be described as follows: first, the weights' values w are randomly initialised. The input neurons' values (taken from the training data) are propagated to the output layer through the hidden layer, this process is called forward propagation. Then, the output neurons' values (i.e., choice probabilities) are compared with the observed choices to compute the function $J(\mathbf{w})$ described in Equation ($A1$). The optimisation mechanism is then conducted by propagating J backward to the input layers through the hidden layer. To adjust the weights, the backward propagation process includes taking the partial derivative of the error J with respect to the weights, called the gradient vector $\mathbf{g}$. Along with a learning rate value η, $\mathbf{w}$ values are re-adjusted as follows:

$$\mathbf{w}_{p+1} = \mathbf{w}_p + \eta_p \mathbf{g}_p \tag{A2}$$

Where p represents a step index. The learning rate η determines how fast the learning algorithm is moving toward the optimum $\mathbf{w}$. If η is very large, there is a relatively high possibility to never obtain the optimum $\mathbf{w}$ due to overshooting. In contrast, using a very small η increases the learning time substantially. One commonly used way to overcome this problem is to use adaptive learning rates, iteratively determined during training.

The process of error (forward and backward) propagation is repeated iteratively until a pre-specified stopping criterion is achieved. This training mechanism is known as back-propagation, and constitutes the most popular approach to train neural networks (Rumelhart et al., 1988). However, it should be noted that moving toward a local minimum is one of the widely reported risks associated with this back-propagation approach (Iyer and Rhinehart, 1999; Park et al., 1996). As such, it is always recommended to train the network more than once to minimise the probability of ending up with a sub-optimal trained network. A pseudocode of the ANN training can be found below, and for comprehensive description of ANNs training interested readers are referred to Bishop (2006).

Pseudocode A1: ANN training

Step 1: Initialisation
Set $\mathbf{w}$ values to random numbers
Step 2: Forward propagation
Propagate the input neuron values $\mathbf{x}$ to output neuron through hidden neurons
Calculate the ANN output neuron values (ANN probabilities)
Calculate the error function (Equation ($A1$))
Step 3: Backward propagation
Calculate the gradient $\mathbf{g}$ for the network neurons
Update $\mathbf{w}$ values
Increase iteration p by one
Go back to step 2 and repeat the process until the selected error criterion is satisfied
Step 4: Repeat (recommended)
Go back to step 1, repeat the whole process to minimise the probability of ending up with a sub-optimal ANN

A.3. Performance metrics for classification

In this section, we define the metrics that are used to evaluate the performance of a trained ANN. The first metric is equivalent to the negative Log-Likelihood measure, presented earlier in Equation ($A1$). More specifically, we modify it slightly to obtain an average (across observations) Log-Likelihood measure (see Table A2). Another metric which is commonly used in the ANN-literature is the classification accuracy measure. This so-called Hit-Rate is computed as follows: the ANN assigns probabilities P to each output neuron (see Fig. A2). The classifier output (denoted by $\hat{y}$) is set to one for the alternative which has the highest probability and zero for all others. In case of two choice situation, the classifier thus assigns 1 (i.e., $\widehat{y_1} = 1$) for the first alternative and zero (i.e., $\widehat{y_2} = 0$)

for the second one if the predicted probability of choosing the first alternative is greater than 0.5. To measure the classification accuracy, we calculate the percentage of the correctly classified observations. A mathematical representation of the used metrics is shown in Table A2.

Table A2
Performance metrics.

Performance metric	Function
Average Log-Likelihood function	$\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk}\ln(P_{nk})$
Classification accuracy	$\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} I$ where $I=\begin{cases}1 & if\ y_{nk}=\widehat{y_{nk}}\\ 0 & otherwise\end{cases}$

References

Abu-Mostafa, Y.S., 1995. Hints. Neural Comput. 7 (4), 639–671.
Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.-T., 2012. Learning from Data, vol. 4 AMLBook New York, NY, USA.
Anthony, M., Bartlett, P.L., 2009. Neural Network Learning: Theoretical Foundations. cambridge university press.
Bartlett, P.L., Maass, W., 2003. Vapnik-chervonenkis Dimension of Neural Nets. The Handbook of Brain Theory and Neural Networks. pp. 1188–1192.
Baum, E.B., Haussler, D., 1989. What size net gives valid generalization? In: Paper Presented at the Advances in Neural Information Processing Systems.
Bierlaire, M., Axhausen, K.W., Abay, G., 2001. The acceptance of modal innovation: the case of Swissmetro. In: Paper Presented at the Swiss Transport Research Conference.
Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford university press.
Bishop, C.M., 2006. Pattern Recognition and Machine Learning. springer.
Blum, A.L., Langley, P., 1997. Selection of relevant features and examples in machine learning. Artif. Intell. 97 (1–2), 245–271.
Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K., 1989. Learnability and the vapnik-chervonenkis dimension. J. ACM 36 (4), 929–965.
Cantarella, G.E., de Luca, S., 2005. Multilayer feedforward networks for transportation mode choice analysis: an analysis and a comparison with random utility models. Transport. Res. C Emerg. Technol. 13 (2), 121–155.
Castelvecchi, D., 2016. Can we open the black box of AI? Nat. News 538 (7623), 20.
Cho, J., Lee, K., Shin, E., Choy, G., Do, S., 2015. How Much Data Is Needed to Train a Medical Image Deep Learning System to Achieve Necessary High Accuracy? arXiv preprint arXiv:1511.06348.
Chorus, C.G., Bierlaire, M., 2013. An empirical comparison of travel choice models that capture preferences for compromise alternatives. Transportation 40 (3), 549–562.
Cireşan, D.C., Meier, U., Schmidhuber, J., 2012. Transfer learning for Latin and Chinese characters with deep neural networks. In: Paper Presented at the Neural Networks (IJCNN), the 2012 International Joint Conference on.
Cortes, C., Jackel, L.D., Solla, S.A., Vapnik, V., Denker, J.S., 1994. Learning curves: asymptotic values and rate of convergence. In: Paper Presented at the Advances in Neural Information Processing Systems.
Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Math. Control, Signals, Syst. MCSS 2 (4), 303–314.
Figueroa, R.L., Zeng-Treitler, Q., Kandula, S., Ngo, L.H., 2012. Predicting sample size required for classification performance. BMC Med. Inf. Decis. Making 12 (1), 8.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning, vol. 1 MIT press Cambridge.
Hagenauer, J., Helbich, M., 2017. A comparative study of machine learning classifiers for modeling travel mode choice. Expert Syst. Appl. 78, 273–282.
Hague Consulting Group, 1998. The Second Netherlands' Value of Time Study: Final Report. Report 6089–1 for AVV, HCG, Den Haag.
Haussler, D., 1992a. Decision theoretic generalizations of the PAC model for neural net and other learning applications. Inf. Comput. 100 (1), 78–150.
Haussler, D., 1992b. Overview of the probably approximately correct (PAC) learning framework. Inf. Comput. 100 (1), 78–150.
Haykin, S.S., 2009. Neural Networks and Learning Machines, vol. 3 (Pearson Upper Saddle River, NJ, USA).
Hensher, D.A., Rose, J.M., Greene, W.H., 2005. Applied Choice Analysis: a Primer. Cambridge University Press.
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. Transport. Res. E Logist. Transport. Rev. 36 (3), 155–172.
Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Network. 2 (5), 359–366.
Huber, J., Train, K., 2001. On the similarity of classical and Bayesian estimates of individual mean partworths. Market. Lett. 12 (3), 259–269.
Iyer, M.S., Rhinehart, R.R., 1999. A method to determine the required number of neural-network training repetitions. IEEE Trans. Neural Network. 10 (2), 427–432.
Jain, A.K., Chandrasekaran, B., 1982. 39 Dimensionality and sample size considerations in pattern recognition practice. Handb. Stat. 2, 835–855.
Kalwani, M.U., Meyer, R.J., Morrison, D.G., 1994. Benchmarks for discrete choice models. J. Market. Res. 65–75.
Kavzoglu, T., Mather, P.M., 2003. The use of backpropagating artificial neural networks in land cover classification. Int. J. Rem. Sens. 24 (23), 4907–4938.
Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Paper Presented at the Ijcai.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Paper Presented at the Advances in Neural Information Processing Systems.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.
LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.-R., 2012. Efficient Backprop *Neural Networks: Tricks of the Trade*. Springer, pp. 9–48.
McFadden, D.L., 1984. Econometric analysis of qualitative response models. Handb. Econom. 2, 1395–1457.
Mohammadian, A., Miller, E., 2002. Nested logit models and artificial neural networks for predicting household automobile choices: comparison of performance. Transport. Res. Rec.: J. Transport. Res. Board 1807, 92–100.
Mukherjee, S., Tamayo, P., Rogers, S., Rifkin, R., Engle, A., Campbell, C., 2003. Estimating dataset size requirements for classifying DNA microarray data. J. Comput. Biol. 10 (2), 119–142.
Neelamegham, R., Jain, D., 1999. Consumer choice process for experience goods: an econometric model and analysis. J. Market. Res. 373–386.
Park, Y.R., Murray, T.J., Chen, C., 1996. Predicting sun spots using a layered perceptron neural network. IEEE Trans. Neural Network. 7 (2), 501–505.
Raudys, S.J., Jain, A.K., 1991. Small sample size effects in statistical pattern recognition: recommendations for practitioners. IEEE Trans. Pattern Anal. Mach. Intell. 13 (3), 252–264.
Ribeiro, B., Sung, A.H., Suryakumar, D., Basnet, R.B., 2015. The critical feature dimension and critical sampling problems. In: Paper Presented at the ICPRAM (1).
Ripley, B.D., 2007. Pattern recognition and neural networks. Cambridge university press.
Rose, J.M., Bliemer, M.C., 2013. Sample size requirements for stated choice experiments. Transportation 40 (5), 1021–1041.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1988. Learning representations by back-propagating errors. Cognit. Model. 5 (3), 1.
Shalev-Shwartz, S., Ben-David, S., 2014. Understanding Machine Learning: from Theory to Algorithms. Cambridge university press.
Silva, J., Ribeiro, B., Sung, A.H., 2017. Finding the critical sampling of big datasets. In: Paper Presented at the Proceedings of the Computing Frontiers Conference.

Sung, A., Ribeiro, B., Liu, Q., 2016. Sampling and evaluating the big data for knowledge discovery. In: Paper Presented at the Proceedings of International Conference on Internet of Things and Big Data (IoTBD 2016). Science and Technology Publications.
Train, K.E., 2009. Discrete Choice Methods with Simulation. Cambridge university press.
Van Cranenburgh, S., Alwosheel, A., 2017. Using Artificial Neural Networks to Investigate Decision-rule Heterogeneity. (submitted for publication).
Van Cranenburgh, S., Guevara, C.A., Chorus, C.G., 2015. New insights on random regret minimization models. Transport. Res. Pol. Pract. 74, 91–109.
Vapnik, V.N., Chervonenkis, A.Y., 2015. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities *Measures of Complexity*. Springer, pp. 11–30.
Vedaldi, A., Lenc, K., 2015. Matconvnet: convolutional neural networks for matlab. In: Paper Presented at the Proceedings of the 23rd ACM International Conference on Multimedia.