Enhancing Discrete Choice Models with Neural Networks

Brian Sifringer, Virginie Lurkin, Alexandre Alahi

March 1, 2018

Extended Abstract

Deep learning has been revisiting many fields for the past few years such as signal processing, computer vision, finance and many more. Its ability to learn a non-linear mapping function from observed data to a desired output is second to none. However, in many fields, it comes with the drawback of being a black-box. When studying demand in travel applications, health care programs or market produce for example, it is of utmost importance we understand what are the key parameters in the decision-making process of the clients. This is why researchers have been using Discrete Choice Modeling (DCM), for they are specifically designed to capture in detail the underlying behavioral mechanisms at the foundation of this decision-making process.
In this paper, we aim at bringing the predictive strength of Neural Networks, a powerful machine learning-based technique, to the field of DCM without compromising interpretability of these choice models. We start by matching the mathematical derivation of the multinomial logit model (MNL) to its neural network equivalent. This allows us to write DCM problems in modern machine learning libraries and opens the way for our novel hybrid approach: we suggest to add a term arising from a dense neural network (DNN) in the utility function of DCM such that:

$$U_{in} = ASC_i + \beta_1 \cdot x_{1in} + \beta_2 \cdot x_{2in} + ... + NN_{in} + \epsilon_{in} \quad (1)$$

where ASC stands for alternative specific constant, β_j are the learned parameters, x_{jin} the features relative to individual n and alternative i, ϵ_{in} the random term in DCM and lastly NN_{in} our new feature arising from a neural network.

Our work suggests different models and input approaches applied to a very simple discrete choice example, where data was gathered through a survey given to the swiss population for a new transportation mode, the swissmetro. The model takes into account both time and cost features as well as ASCs for the 3 following choices: train, car and swissmetro. The dataset is composed of many more unused features such as income, age, purpose of travel and more. Since we make use of neural networks, we have split the data into training and testing sets to assure no overfitting. We gather results from three different models :

1. Multinomial Logit (no added term, $U_{in} = ASC_i + \beta_1 \cdot x_{1in} + ... + \beta_d \cdot x_{1dn} + \epsilon_{in}$)
2. Dense Neural Network (equivalent to $U_{in} = NN_{in}$)
3. Hybrid model (see equation 1)

and make use of two different input methods:

a. DNN components receive the same input as the utility functions

b. DNN components receive all unused features from the original DCM

This gives us a total of 6 different combinations. For simplicity, we will refer to the different models by their number (1-3) followed by their input methods 'a' or 'b'. Since the multinomial logit model does not have any DNN parts, 1-a and 1-b are equivalent and will be simply referred to as 1 .

The pure neural networks, 2-a and 2-b, allow us to show the predictive strength of these densely connected models and demonstrate up to 27% increase in accuracy compared to the simple multinomial logit. However, due to these networks' nature, all statistics and insight on the used features are lost.
The hybrid model 3-a (Fig. 1) will unfortunately also lose most of its original DCM parameters' significance. The DNN component overruns the multinomial logit part of the model when trying to maximize the likelihood. However, when training the model,

we may fix the original β parameters to their minima found in 1 and only allow the DNN component to train its weights. The utility functions will have the new term increase the final likelihood of about 25% and keep the desired minima for its original parameters. In this hybrid case, it is interesting to add a linear variable multiplying the new term such that $U_{in} = ... + \beta_{NN} \cdot NN_{in} + ...$ and then allow all variables β_j to find their new minima. Doing this shows us effectively how the parameters have lost their significance to the new term obtained by the DNN component. It is interesting to note that this may be a way to understand what features are most important in the neural network black box, or in the opposite, which features are the least linear in the MNL model.
Lastly, we have the hybrid model 3-b (Fig. 2) which not only increases likelihood up to 30%, but it also keeps the strong parameters significance of the original DCM model[1]. Furthermore, we have reasons to believe the added term with DNN fits very well in DCM theory when using this last method. Indeed, all utility functions are defined with a random utility term ϵ, capturing all unknown or unused features of the model which may appear in the thinking process of an individual. If our selected features are too few, this unknown term may be too important giving us a model both weak at predicting and sometimes having parameters far from their true value. We believe that this ϵ term may be estimated with the data-driven approach of deep learning, not only increasing the accuracy as discussed above, but also allowing the original parameters to find a truer minimum *(ongoing)*.

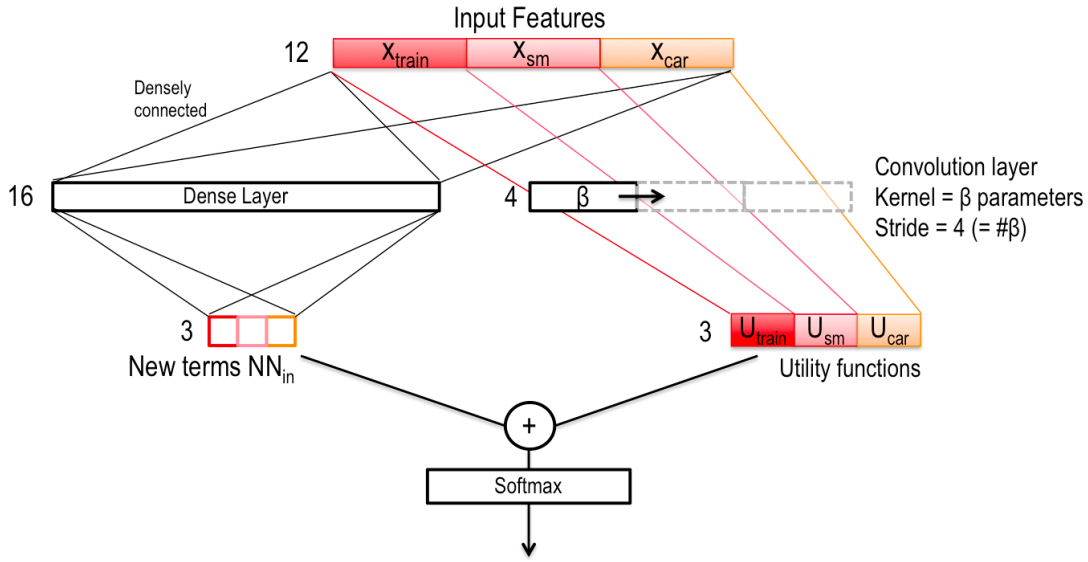

Figure 1: *Hybrid model 3-a. Written in modern machine learning libraries, one can flexibly change the model, optimizers in training and more. On the right-hand side, the weights of the kernel correspond to the β_j parameters, and applying a convolution layer to the input features gives us the same utility functions as in MNL. The left hand side is the DNN component, producing a single term for each utility function and highly increasing predictive accuracy.*

[1]It is possible to fix the original parameters to their initial minima and still appreciate high accuracy increase as done in 3-a. Ongoing research is to show that the new parameters actually reach a better value as explained afterwards

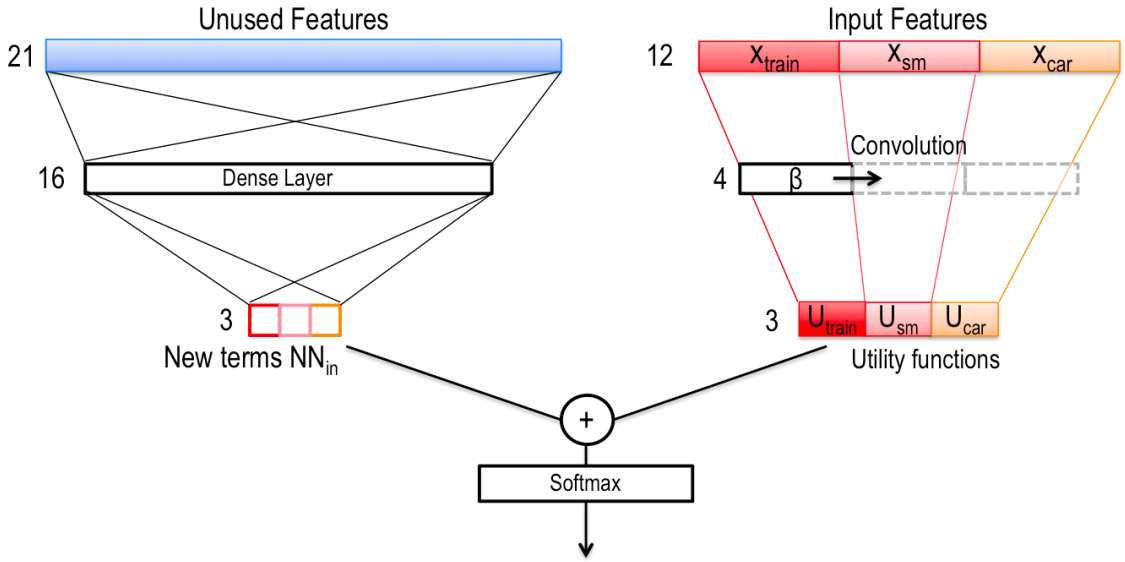

Figure 2: *Hybrid model 3-b. Our most promising model for ongoing research and with best likelihood results. Both hybrid models can be trained by fixing the DCM parameters, in other words the convolution kernel, to a previously discovered minima. This model however does not need this method to have all significant parameters. Ongoing research is to investigate if these are at a better value than with a standard MNL.*