Detection of Covariate Interactions by Deep Neural Network Models

Yijun Shao, PhD
Biomedical Informatics Center, George Washington University, Washington, DC, USA, yshao@gwu.edu

Ali Ahmed, MD
Washington DC VA Medical Center, Washington, DC, USA, ali.ahmed@va.gov

Qing Zeng-Treitler, PhD
Biomedical Informatics Center, George Washington University, Washington, DC, USA, zengq@gwu.edu

ABSTRACT

Deep neural networks (DNNs) are increasing popular in many areas including healthcare, but they are difficult to explain. There have been approaches for explaining DNN models which focus on measuring the individual variable effects on the outcome. In this study we first develop a method for measuring the interactions between covariates in the DNN models, and then we assess the ability of a DNN model in detecting the interaction effects using simulation data.

CCS CONCEPTS

• Applied computing • Life and medical sciences • Health informatics

KEYWORDS

deep neural network, explanation, interaction effect

1 Introduction

Deep neural networks (DNNs) have attracted much attention in recent years due to their superior performance in a variety of tasks such as computer vision, speech recognition and board game playing [1-3]. The application of DNNs to medicine and healthcare quickly followed and has shown some success [4]. Despite their increasing popularity, DNN models are very difficult to explain compared to traditional statistical models such as linear regression, which is hindering their adoption in many areas including healthcare. Therefore, there has been much interest in explaining DNN models.

Approaches for explaining DNN models include Attention Mechanism [5], Local Interpretable Model-agnostic Explanations (LIME) [6] and Layer-wise Relevance Propagation (LRP) [7]. In a similar spirit as LIME, we have developed a method called impact scores to explain the DNN models [8]. Impact scores can measure how individual variables impact on the outcome. It is natural to ask whether DNN models can capture interaction effects of two variables and how to measure the interaction effects.

Covariate interactions are commonly observed in healthcare studies. For example, smoking and exposure to asbestos have an interaction effect on predicting the risk of lung cancer [9]. Covariate interactions have been well studied in statistics in the context of regression [10], but are less well studied in the context of DNN models. Intrator et. al. used graphical tools for detecting and studying the interactions in neural network models [11]. In this study, we developed a quantitative method to measure covariate interactions and assess the ability of a DNN model in detecting the interaction effects using simulation data. Compared to real-world data, simulation data have the advantage that the underlying relationship is known, which allows one to evaluate the computed interaction effects against those defined in the underlying relationship.

2 Methods

The simulation data are generated as follows. First, we use 100 variables $x_1, x_2, \dots, x_{100}$ as predictors and a binary variable z for two outcomes represented by values 0 and 1 respectively. Among the 100 variables the first 50 are binary variables taking values 0/1, and the second 50 are continuous variables taking values between 0 and 1. This setting is to resemble the real situation of patient data: some variables such as gender, diagnoses, procedures are usually treated as binary variables, while the other variables such as age, vital signs, lab results are usually treated as continuous variables. Although the continuous variables may have different numerical value ranges in their original form, they can always be normalized to range from 0 to 1.

For each variable x_i, we define a reference value representing the baseline status such as "not having a disease" or "not taking a medication". For convenience, we choose 0 as the reference value for all the variables.

We experiment with a nonlinear relationship between the outcome variable z and the predictors $x_1, x_2, \dots, x_{100}$. Specifically, randomly sample 3 subsets $\{i_n\}_{n=1}^{20}$, $\{j_n\}_{n=1}^{20}$ and $\{k_n\}_{n=1}^{20}$ of the index set $[1, 2, \dots, 100]$ with $\{j_n\} \cap \{k_n\} = \emptyset$, and then define a nonlinear relationship by

$$\text{logit}(p) = y = \beta_0 + \sum_{i=1}^{100} \beta_i x_i + \sum_{n=1}^{20} \gamma_n x_{i_n}^2 + \sum_{n=1}^{20} \theta_n x_{j_n} x_{k_n}$$

where $p = \text{Prob}(z = 1 | x_i; \beta_i, \gamma_n, \theta_n)$, and $\text{logit}(p) = \ln \frac{p}{1-p}$ is the logit function. This relationship has 20 square terms and 20 cross-product terms in addition to 100 linear terms. The coefficients β_i's and γ_n's are randomly generated from a uniform distribution with range -1 to 1, and θ_n's are randomly generated from a uniform distribution with range -10 to 10. The cross-product terms $\theta_n x_{j_n} x_{k_n}$ are the interaction terms and θ_n are the interaction coefficients. Note that for any variable not occurring in the interaction terms, its interaction with any other variable is exactly 0.

After the simulation data are generated, we train a DNN model on the data. The DNN has an architecture as follows. It has an input layer of 100 nodes, an output layer with 1 node, and 10 hidden fully-connected layers whose numbers of nodes are alternatingly 70 and 50, starting with 70 for the first hidden layer and ending with 50 for the last hidden layer. Their nonlinear activation functions are all the rectified linear unit (ReLU) function $f(x) = \max(0, x)$. The nonlinear activation function for the output layer is the sigmoid function $\sigma(u) = 1/(1 + e^{-u})$ so that the DNN outputs a number p between 0 and 1 representing the predicted probability of the outcome being 1.

We randomly divide the set of simulation data into 3 subsets: training (60%), validation (20%) and testing (20%), and then we train the DNN model on the training set. The weights of the DNN are initialized with randomly generated small numbers, and updated using the mini-batched stochastic gradient decent method with Nesterov momentum. The mini-batch size is100, learning rate is 0.001 and momentum is 0.9. To avoid over-fitting to the training data, we adopted the strategy of early stopping. Specifically, after each epoch of training, the trained DNN model is applied to the validation set to measure the area under curve (AUC) on it. When the validation AUC reaches a peak point such that there is no improvement over the following 10 epochs, we take the DNN model with the peak validation AUC as the final model. This effectively makes the training stop at the epoch producing the peak AUC.

With the final DNN model, we calculate interaction scores as follows. Let $p = F(x_1, \dots, x_{100})$ be the final DNN model, and let $f = \text{logit} \circ F$, so that $\text{logit}(p) = f(x_1, \dots, x_{100})$. For two variables x_i and x_j, we define the interaction score between them at two levels – the individual (instance) level and the population level. The individual-level interaction score on an instance is defined as:

$$\frac{f(\dots, x_i^c, \dots, x_j^c, \dots) - f(\dots, x_i^r, \dots, x_j^c, \dots) - f(\dots, x_i^c, \dots, x_j^r, \dots) + f(\dots, x_i^r, \dots, x_j^r, \dots)}{(x_i^c - x_i^r)(x_j^c - x_j^r)}$$

where x_i^c(resp. x_j^c) and x_i^r(resp. x_j^r) are, respectively, the current and reference values of the variable x_i(resp. x_j) on the instance, and "..." represents the current values of all other variables than x_i and x_j. Note that the individual interaction score is only defined on those instances such that $x_i^c \neq x_i^r$ and $x_j^c \neq x_j^r$.

The population-level interaction score is defined simply as the mean of all the individual-level interaction scores on all the instances in the training set such that those scores are defined. Thus the instances on which the interaction score is undefined are excluded from the calculation of mean. Note that when interaction scores are applied to the nonlinear relationship defined for generating the simulation data, the individual/population-level interaction scores are exactly θ_n for the 20 pairs of variables x_{j_n} and x_{k_n}, and are zero for the rest pairs of variables. This shows the validity of the above definitions.

The population-level interaction scores calculated using the DNN model, which we call the "predicted values", are compared to the population-level interaction scores calculated on the nonlinear relationship for the simulation data, which we call the "true values" or "ground truth". The closeness between the predicted values and true values reflects the capability of the DNN model in detecting the interaction effects in the nonlinear relationship. We evaluate the "closeness" with several metrics. First we calculate the mean absolute error (MAE) which measures the difference in an absolute sense. For two sequence of values a_i and b_i, $i = 1, \dots, n$, the MAE is defined as $\frac{1}{n}\sum_{i=1}^{n}|a_i - b_i|$. We also calculate the mean absolute predicted value (MAPV), and the mean of the absolute true value (MATV). The ratios MAE/MATV and MAE/MAPV measure the difference between the predicted values and true values in a relative sense. These metrics are calculated on 2 sets of variable pairs separately: 1. the randomly selected 20 pairs (x_{j_n}, x_{k_n}), whose true values are θ_n; 2. the rest of the variable pairs excluding the 20, whose true values are all zero.

We also calculate the Pearson's correlation and Spearman's rank correlation, which measures the agreement in relative values and in relative ranks, respectively. Lastly, we calculate the sign agreement. It is the proportion of the predicted values which have the same sign as the corresponding true values. Different signs of interaction scores indicate the different interaction types. The correlations and sign agreements are only calculated on the 20 pairs of variables since they are not applicable on the rest of the 4930 pairs.

3 Results

The trained DNN model has a performance as follows: Training AUC = 0.978, Validation AUC = 0.950, Testing AUC = 0.945.

The population-level interaction scores are calculated based on the trained DNN model, and then compared to the population interaction scores based on the nonlinear relationship used for simulating the data. The comparison results are shown in Table 1.

Table 1. Comparison of the interaction scores detected by the DNN model with the true interaction scores in the nonlinear relationship.

	On the 20 variable pairs (x_{j_n}, x_{k_n})	On the rest 4930 variable pairs (x_i, x_j)
MAE	3.49	0.08
MATV	4.85	0
MAPV	1.36	0.08
MAE/MATV	0.72	N/A
MAE/MAPV	2.57	1
Pearson's correlation	0.86	N/A
Spearman's correlation	0.98	N/A
Sign agreement	1.0	N/A

On the randomly selected 20 variable pairs, the MATV is about 5, which is as expected since θ_n are randomly sampled from the uniform distribution ranging from -10 to 10. The MAE is about 3.5, quite large (72%) compared to MATV, suggesting that the DNN model did not capture the full extent of interactions. Although the MAPV is smaller than the MATV on the 20 pairs, it is still much larger than the MAPV on the rest 4930 pairs, showing that the DNN model can detect the interaction effects to certain degree. The perfect sign agreement on the 20 variable pairs shows that the DNN model actually detect the right interaction type, and the high Pearson's correlation and almost perfect Spearman's correlation shows that the detected interaction effects have a high agreement with the ground truth in relative values and ranks.

4 Discussions and Conclusion

In this study, we simulate data that mimic some common situations in healthcare outcome prediction, and train a DNN model on the simulation data in order to assess the capability of the DNN model in detecting the covariate interactions. The simulation uses an underlying nonlinear relationship with 20 explicit interaction terms. The coefficients of the 20 terms are set to have higher magnitude in order to make the interaction effects prominent. We define "interaction score" for the DNN model in order to calculate the interaction effects captured by the model. When the same interaction score formula is applied to the underlying relationship, the exact coefficients of the interaction terms are obtained, showing the validity of the interaction score formula. With the simulation data, we know the true values of the interaction effects, which served as the ground truth for evaluating capability of the DNN model in detecting the interactions. The results show that the DNN model can detect prominent/strong covariate interaction effects, although the detected effects are generally smaller than the true effects.

ACKNOWLEDGMENTS

This study is sponsored by GW-CN CTSA (1UL1TR001876-01) and VA HSRD (1I01HX002308-01A1).

REFERENCES

1. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. in *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*. 2012. Lake Tahoe, Nevada.
2. Xiong, W., et al., *Toward Human Parity in Conversational Speech Recognition.* IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017. **25**(11).
3. Silver, D., et al., *Mastering the game of Go with deep neural networks and tree search.* Nature, 2016. **529**(7587): p. 484-9.
4. Lee, J.G., et al., *Deep Learning in Medical Imaging: General Overview.* Korean J Radiol, 2017. **18**(4): p. 570-584.
5. Bahdanau, D., K. Cho, and Y.J.a.p.a. Bengio, *Neural machine translation by jointly learning to align and translate.* 2014.
6. Ribeiro, M.T., S. Singh, and C. Guestrin. *Why should i trust you?: Explaining the predictions of any classifier*. in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016. ACM.
7. Binder, A., et al. *Layer-wise relevance propagation for neural networks with local renormalization layers*. in *International Conference on Artificial Neural Networks*. 2016. Springer.
8. Shao, Y., et al., *Shedding Light on the Black Box: Explaining Deep Neural Network Prediction of Clinical Outcome*, in *ICHI 2019: 21st International Conference on Health Informatics*. 2019: Rome, Italy.
9. Lee, P.N., *Relation between exposure to asbestos and smoking jointly and the risk of lung cancer.* Occup Environ Med, 2001. **58**(3): p. 145-53.
10. Jaccard, J. and R. Turrisi, *Interaction Effects in Multiple Regression*. 2nd Edition ed. 2003: SAGE Publications, Inc.
11. Intrator, O. and N. Intrator, *Interpreting neural-network results: a simulation study.* Computational Statistics & Data Analysis, 2001. **37**(3): p. 373-393.