ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Interpreting Deep Learning Models for Ordinal Problems

José P. Amorim[1,2], Inês Domingues[1,2], Pedro H. Abreu[1] and João Santos[2] *

[1] University of Coimbra - CISUC - Department of Informatics Engineering
Pólo II, Pinhal de Marrocos, 3030-290 Coimbra - Portugal

[2] IPO-Porto Research Center - CI-IPOP
Rua Dr. António Bernardino de Almeida, 4200-072 Porto - Portugal

Abstract. Machine learning algorithms have evolved by exchanging simplicity and interpretability for accuracy, which prevents their adoption in critical tasks such as healthcare. Progress can be made by improving interpretability of complex models while preserving performance. This work introduces an extension of interpretable mimic learning which teaches interpretable models to mimic predictions of complex deep neural networks, not only on binary problems but also in ordinal settings. The results show that the mimic models have comparative performance to Deep Neural Network models, with the advantage of being interpretable.

1 Introduction

The effectiveness of human resources can be enhanced by machines, not only by lowering costs, but also by reducing errors related to tiredness and other human factors. In spite of machines being powerful at classification tasks such as image recognition [1] and time series classification [2], the produced models can at times be complex and hard to interpret. The black-box nature of these techniques prevents their use in several contexts such as banking and healthcare, where practitioners often prefer to be able to understand the model behavior and predictions in detriment to performance.

Interpretable mimic learning [3] has drawn inspiration from model compression [4] to reduce this trade-off. Model **compression** consists of approximating a function learned by a slow and complex model with a faster and simpler model with comparable performance [4]. Although first used with the goal of compressing the knowledge learned by a model ensemble into a shallow neural network, it can also be used to mimic other complex models such as deep neural networks.

Ba and Caruana [5] demonstrated, using **mimic** learning, a variant of model compression, that shallow neural networks could, in principle, learn as accurate functions as the ones learned by deep nets, but current training algorithms would not allow it. This was generalized by **distillation** [6], which works by using a transfer set to train the complex model with cross-entropy and *softmax* with high temperature (T) and using these soft predictions to train the distilled model also with high temperature. At test time, the distilled model is used

*This article is a result of the project NORTE-01-0145-FEDER-000027, supported by Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF).

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

with temperature 1. They demonstrated that the use of *logit* is a special case of distillation when T is high compared with the magnitude of the *logits*.

While the motivation of the the above model compression approaches was the reduction of the required storage and computational power at test time, by teaching interpretable models we can obtain another advantage, **interpretability**. Recent work on interpretable mimic learning [3] showed that it is possible to extract knowledge from deep learning models and use it to produce simpler and more intepretable models, without decreasing performance. They presented two different training pipelines, one in which the soft labels of the deep model are directly used to train the mimic model (i.e. Feed-forward Networks and Gated Recurrent Units), replacing the labels of the training set. In the second pipeline, the activations of the last hidden layer, X_{nn}, are used to train a helper classifier, and the soft predictions of the helper classifier are used to train the mimic model. But the chosen mimic model's interpretability is limited to feature importance and selection of representative decision rules, and this approach was intended for binary classification, leaving ordinal classification, still to be explored.

Standard classification algorithms for unordered problems can be used for classification of ordinal problems but with loss of information. Regression techniques have also been used, by transforming the nominal classes into numeric values and returning the predicted labels back to discrete values to obtain the final predicted labels [7]. The weakness of this methodology is that the real distances between classes are, in a typical ordinal problem, unknown and context-dependent.

To this end, an ordinal mimic learning approach was proposed, extending interpretable mimic learning [3] for ordinal classification, producing interpretable models which mimic the predictions of complex neural networks. The contribution of the present work is a new framework for ordinal mimic learning validated on 19 datasets.

2 Ordinal Interpretable Mimic Learning

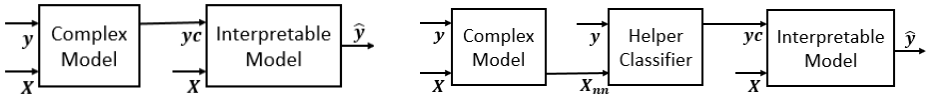
Ordinal interpretable mimic learning extends interpretable mimic learning [3], generalizing the two pipelines for binary classification, to problems with ordinal classes. By combining the two training pipelines from [3] with two ordinal approaches, Multiclass and Frank&Hall [8], we can obtain interpretable models that mimic complex models.

We propose four architectures that combine the pipelines in [3] and two ordinal approaches, Multiclass and Frank&Hall [8]. In Pipeline 1 (Figures 1a and 2a), we train the complex model(s) (e.g. feed-forward neural networks) using the training set $\{X, y\}$, composed of the original features X and the targets y, obtaining the soft predictions of the training set, yc. An interpretable model is then trained to mimic the complex model, using as input $\{X, yc\}$.

In Pipeline 2 (Figures 1b and 2b), the activations of the last hidden layer of the complex model(s), X_{nn}, are used in combination with the original targets y, to train Helper Classifier. We then take the soft predictions of the Helper

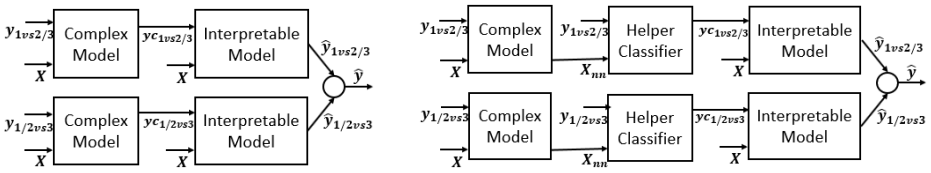
ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

Classifier, yc, and the original features, X, and train the interpretable model.



(a) Pipeline 1 (b) Pipeline 2

Fig. 1: Illustration of the Multiclass Mimic Learning approach



(a) Pipeline 1 (b) Pipeline 2

Fig. 2: Illustration of the Frank&Hall Mimic Learning approach for a 3-class problem

In both pipelines, at testing time, the classification of unseen samples is performed using only the mimic interpretable model(s). In the Multiclass approach (Figure 1), only one K-class classifier is trained and the soft predictions are weighted according with the class label. By weighting the soft predictions, the K class probabilities are combined into one numeric using the following equation:

$$yc = \sum_{k=1}^{K} [k * Pr(V = V_k)] \tag{1}$$

In the Frank&Hall architecture (Figure 2), the K-class classification problem is divided into $K-1$ classification problems. Each classifier i learns to differentiate classes $C_1, \ldots, C_i$ from classes $C_{i+1}, \ldots, C_K$. For each binary problem, we train a complex model, a Helper Classifier (in case of the pipeline 2), and an Interpretable Model. The predictions of each interpretable model are combined so that, for the case of a 3 class problem, if the two models agree with value -1 the result is class **1**, if they agree with value 1 than the result is class **3**, otherwise the result is class **2**.

So far, we have considered as a complex model, a neural network with k neurons on the output layer. For pipeline 1, this can be generalized to any multi-class classifier capable of producing class probabilities[1]. In the case of pipeline 2, the use of the activations of the last hidden layer of the complex model, X_{nn}, restricts it to neural networks.

[1]When using multi-class classifiers with outputs in the range [1,k], the output can be used directly with no need to apply Equation 1.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

3 Experimental setup

The ordinal datasets include the ones used in [9][2], which were used for benchmark different ordinal approaches, as well as two heathcare datasets described in [10, 11], where the use of the ordinal nature of the response to cancer treatment could improve its prediction. Feature selection was made using Neighborhood Component Analysis [12] or ReliefF [13], which are filter methods (independent from the classification method) and suitable for multi-class classification.

We tested three different types of inputs for the mimic model, *softmax*, *double* and *neighbor*. The *softmax* is the one illustrated in Figures 1 and 2, where the input is $\{X, y_c\}$. In the *double* we train the interpretable model with the original dataset concatenated with the one with soft labels, $\{XX, yy_c\}$, represented on equation (2). For the *neighbor*, we discard the samples that the complex model classifies incorrectly, $\{XX', yy'_c\}$, represented on equation (3).

$$\{XX, yy_c\} = \left[\begin{array}{cccc|c} x_{11} & x_{12} & \dots & x_{1m} & y_1 \\ x_{21} & x_{22} & \dots & x_{2m} & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} & y_n \\ \hline x_{11} & x_{12} & \dots & x_{1m} & y_{c_1} \\ x_{21} & x_{22} & \dots & x_{2m} & y_{c_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} & y_{c_n} \end{array}\right] \tag{2}$$

$$\{XX', yy'_c\} = \left[\begin{array}{cccc|c} x_{11} & x_{12} & \dots & x_{1m} & y_1 \\ x_{21} & x_{22} & \dots & x_{2m} & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} & y_n \\ \hline x_{11} & x_{12} & \dots & x_{1m} & y'_{c_1} \\ x_{21} & x_{22} & \dots & x_{2m} & y'_{c_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n'1} & x_{n'2} & \dots & x_{n'm} & y'_{cn'} \end{array}\right], n' \leq n \tag{3}$$

Three interpretable supervised methods were selected to validate our approaches: Linear Regression, Regression Tree and Symbolic Regression; and a Feedfoward Neural Network (FNN) as our complex model.

All the above models were trained using MATLAB's (v. 9.3.0.713579) default parameters, excluding the FNN's hyperparameters which were tuned using grid search by exploring the number of hidden layers, $n_H \in \{1, 2, 3, 4, 5\}$, and the number of hidden units, $n_{HU} \in \{16, 32, 64, 128, 256, 512\}$.

4 Results

The algorithms were ranked by the average MAE, obtained using leave-one-out cross-fold validation on the healthcare datasets and over 2-folds on the other datasets, and the datasets were grouped according to the number of features, as shown in Table 1.

[2]available at http://www.uco.es/grupos/ayrna/ucobigfiles/datasets-orreview.zip

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Table 1: Top 10 ranked algorithms for datasets grouped based on number of features and ordered by approach. FS, Pl, Classif, m and AR stand for feature selection, pipeline, classifier, number of features and average rank respectively. The top 5 algorithms' ranks based on the number of features are highlighted in bold.

Approach	Pl	Input	FS	Classif	m=1-4	m=5-25	AR
Multiclass	2	Double	none	RT	**5**	**1**	3.0
Multiclass	1	Softmax	none	RT	6	**3**	4.5
Multiclass	1	Softmax	NCAreg	RT	7	**4**	5.5
Multiclass	1	Neighbor	none	RT	10	**2**	6.0
Multiclass	1	Double	none	RT	9	**5**	7.0
Multiclass	2	Neighbor	none	RT	8	6	7.0
Frank&Hall	2	Softmax	NCAreg	SR	**2**	7	4.5
Frank&Hall	1	Softmax	none	SR	**3**	8	5.5
Frank&Hall	2	Softmax	none	SR	**1**	10	5.5
Frank&Hall	2	Softmax	none	RT	**4**	9	6.5

Based on Table 1 we can see that both Frank&Hall and Multiclass approaches reach the top 10 while no interpretable model that does not use the mimic approach did. We can also see that Frank&Hall methods did better on datasets with fewer features (< 5). We believe this happens because the models perform worse with fewer features, so by predicting a mid-class when different classifiers are disagreeing the Frank&Hall method avoids making bigger mistakes.

From Table 1, we can see that Multiclass with *double* and *neighbor* perform better with more than 4 features. This is in line with the general observation that "the optimal number of features increases with increasing sample size"[3] [14], since *double* and *neighbor* are trained with augmented datasets.

Also in Table 1, we can see that every Frank&Hall approach reaching the top 10 is instantiated with *softmax*, indicating that, for Frank&Hall, *softmax* predictions do better than double and neighbor. This may be because, by using hard predictions which can only take opposing values 0,1 rather than values in between [0,1], predictions of the different binary classifiers might disagree more.

Frank&Hall ($MAE = 1.3$, $std = 1.0$) algorithms show higher mean MAE and standard deviation than Multiclass ($MAE = 0.6$, $std = 0.4$), which indicates that this simplified ordinal classification approach may not capture correctly the ordinal nature of the classes.

5 Conclusions

In this paper, an ordinal interpretable mimic learning framework was proposed to solve the performance versus interpretability trade-off in the context of ordinal problems. Results show that the interpretable models trained to mimic complex models outperform the models trained directly on the original datasets.

[3] A warning should here be made to the fact that the optimal-feature-size relative to the sample size depends not only on the classifier but also the feature-label distribution [14].

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

The main focus of our future work will be to explore strategies that bring interpretability to a local level, such as LIME [15]. It is also important to recognize that our conclusions are limited to the scope of the datasets and models used, therefore further work must be done to validate our approach on more complex problems, such as time series classification. Additionally, we will investigate the effectiveness of our approach with other ordinal classification methods such as data replication [16].

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *25th International Conference on Neural Information Processing Systems (NIPS)*, volume 1, pages 1097–1105. Curran Associates Inc., 2012.

[2] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, abs/1412.3555, 2014.

[3] Z. Che, S. Purushotham, R. Khemani, and Y. Liu. Interpretable Deep Models for ICU Outcome Prediction. *AMIA Annual Symposium proceedings*, 2016:371–380, 2016.

[4] C. Bucila, R. Caruana, and A. Niculescu-Mizil. Model compression. In *12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 535–541, 2006.

[5] L. J. Ba and R. Caruana. Do deep nets really need to be deep? In *27th International Conference on Neural Information Processing Systems (NIPS)*, volume 2, pages 2654–2662. MIT Press, 2014.

[6] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *NIPS 2014 Deep Learning Workshop*, 2015.

[7] S. Kramer, G. Widmer, B. Pfahringer, and M. De Groeve. Prediction of Ordinal Classes Using Regression Trees. *Fundamenta Informaticae XXI*, pages 1001–1013, 2001.

[8] E. Frank and M. Hall. A simple approach to ordinal classification. *European Conference on Machine Learning (ECML)*, 2167:145–156, 2001.

[9] J. C. Gámez, D. García, A. González, and R. Pérez. Ordinal classification based on the sequential covering strategy. *International Journal of Approximate Reasoning*, 76:96–110, 2016.

[10] M. A. Nogueira. *Creating Evaluation Functions for Oncological Diseases based on PET/CT*. Master thesis in biomedical engineering, University of Coimbra, 2015.

[11] M. A. Nogueira, P. H. Abreu, P. Martins, P. Machado, H. Duarte, and J. Santos. An artificial neural networks approach for assessment treatment response in oncological patients using PET/CT images. *BMC Medical Imaging*, 17(1):17–13, 2017.

[12] W. Yang, K. Wang, and W. Zuo. Neighborhood Component Feature Selection for High-Dimensional Data. *Journal of Computers*, 7(1):162–168, 2012.

[13] M. Robnik-Siknja and I. Kononeko. Theoretical and empirical analysis of RelifF and RReliefF. *Machine Learning*, 53:23–69, 2003.

[14] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8):1509–1515, 2005.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 39:1135–1144, 2016.

[16] J. S. Cardoso and J. F. P. Costa. Learning to classify ordinal data: the data replication method. *Journal of Machine Learning Research*, 8:1393–1429, 2007.