Transparent Explainable Logic Layers Supplementary

Alessio Ragno[a,*], **Marc Plantevit**[b], **Celine Robardet**[c] and **Roberto Capobianco**[d]

[a]Sapienza University of Rome, Rome, Italy
[b]EPITA Research Laboratory (LRE), FR-94276, Le Kremlin-Bicêtre, France
[c]INSA Lyon, CNRS, LIRIS UMR 5205, F-69621 Villeurbanne, France
[d]Sony AI

Model Debugging and OOD Identification

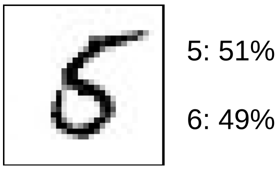

Figure 1: Example of indistinct classification. We report an image from the MNIST dataset representing a digit whose value is uncertain to the concept-bottleneck model. This results in activating both the rules for the even and odd classes, which can be used as a sign for detecting out-of-distribution samples.

The analysis of logical rules provides a direct insight into the model, enabling the examination of bias or inconsistencies in the learned decision process. In this case study, we employ the MNIST dataset to demonstrate how our approach validates the model's correctness and identifies anomalies. Upon analyzing TELL models trained on MNIST, we observe that all models have learned the following rules:

$$\begin{aligned} Even &\iff (0) \vee (2) \vee (4) \vee (6) \vee (8) \\ Odd &\iff (1) \vee (3) \vee (5) \vee (7) \vee (9) \end{aligned}$$

However, the correctness of these rules does not account for the model's imperfections in accuracy. Therefore, we investigate to identify a set of samples that are mispredicted. For instance, Figure 1 presents an example where the concept encoder struggles to differentiate between a 5 and a 6. This ambiguity can lead to a model predicting more than one class as active or, conversely, predicting none of the classes. This unusual behavior allows for data inconsistency detection, and when combined with the explanation rule, we can gain insights into its underlying reasons. We find that this application is highly relevant when combined with concept-bottleneck models, as it is aligned with their intrinsic characteristic of providing test-time interventions [1].

[*] Corresponding Author. Email: ragno@diag.uniroma1.it.

Training hyperparameters

Table 1: Hyperparameters of the models.

Dataset	Batch size	λ	Learning rate	Pruning quantile
MIMIC-II	64	0.01	0.01	0.95
V-DEM	128	0.01	0.01	0.90
MNIST E/O	256	0.01	0.01	-
CUB	128	0.01	0.01	0.95
CUB w/ PIP-Net	128	0.01	0.01	0.99

Table 1 reports the hyperparameters for the TELL models over the various datasets.

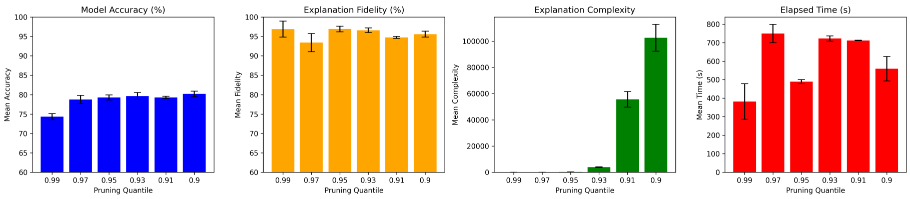

Figure 2: Effects of pruning on the models' performances on the MIMIC-II dataset. We report the model accuracy, fidelity, explanation extraction time, and explanation complexity against several values of the pruning quantile.

The role of pruning

Here, we show the effects of changing the pruning values on the model's performances. We report in Figure 2 the plots of different metrics to evaluate how they are affected by pruning. Overall, we observe that the fidelity and the extraction time are not directly correlated with the variation of the pruning parameter. On the other hand, varying pruning highly affects the model's accuracy score and the explanation complexity. Indeed, by decreasing the pruning quantile, we observe that the model scores better accuracy scores at the cost of producing more complex rules. We deduce that choosing the correct pruning parameter is highly dependent on the task and the end-user. Indeed, in some cases, it might be more useful to have a higher readability of the rules, while in other cases, we might want to optimize the model's accuracy.

References

[1] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020.