MiniProject 4: Reproducibility in Machine Learning

Jonathan Halimi, Thien Pham, Rafid Saif

December 8, 2023

1 Abstract

This paper aims to reproduce the results obtained through the Deep Neural Decision Trees (DNDT) model as published by Yongxin Yang, Irene Garcia Morillo, and Timothy M. Hospedales in 2018. We reproduced the same architecture and trained the model on a number of tabular datasets. We performed experiments to test the claims made by the authors of the paper regarding the DNDT performance. We found that DT yielded better performance, higher test accuracies compared to DNDT and NN, which are similar to what the original paper reported. We also validated the active cut-points analysis. Our main goal was to reproduce the results obtained in the paper as a way to evaluate it, however, we found the information presented in the paper to be insufficient to completely reproduce these results.

Contents

1 Abstract 1

2 Introduction 1

3 Scope of reproducibility 1

4 Methodology 1
4.1 Model descriptions . . . 1
4.2 Dataset . . . 1
4.3 Hyperparameters . . . 2
4.4 Experimental setup and code . . . 2

5 Results 2
5.1 Accuracy . . . 2
5.2 Active cut-points . . . 2
5.3 Tree structure . . . 3

6 Results reproducing original paper 4

7 Discussion and Conclusion 4

8 Statement of Contributions 4

2 Introduction

This report aims to reproduce the results obtained through the DNDT architecture used on multiple datasets, as published in the paper ”Deep Neural Decision Trees” by Yongxin Yang, Irene Garcia Morillo, Timothy M. Hospedales in 2018 [3]. The publication of this paper is a step towards interpretability in machine learning because it presented Deep Neural Decision Trees (DNDT) – tree models realized by neural networks. The paper claimed a DNDT is intrinsically interpretable; and can be easily implemented in Neural Network (NN) toolkits, and trained with gradient descent rather than greedy splitting. This means DNDT would allow interpretable models to be trained much faster than conventional Decision Trees (DT). The paper evaluates DNDT on several tabular datasets, verifies efficacy, and investigates similarities and differences between DNDT and DTs.

3 Scope of reproducibility

In the paper, the authors highlight the important of predictive models, especially in cases where ethics are involved. DNDT has achieved excellent performance in many areas, however, lack of interpretability prevents this family of black-box models from being used in applications. Therefore, the paper emphases the importance of understanding how each factor contributes to the prediction by following the structure of the tree and check exactly how a prediction is made. The report claims DNDT can be easily implemented in a few lines of code in any NN software framework. All in all, we decided to test two specific claims in their report:

- Claim 1: DNDT has better performance than NNs for certain tabular datasets, while providing an interpretable decision tree
- Claim 2: Compared to conventional DTs, DNDT is simpler to implement, simultaneously searches tree structure and parameters with SGD, and is easily GPU accelerated

4 Methodology

4.1 Model descriptions

The DNDT is based on a combination of the architecture of traditional Neural Networks and Decision Trees. Essentially we are using gradient descent instead of greedy splitting methods when going down the tree. Given the former we are able to use GPU acceleration fairly easily which makes this model potentially more suited for larger models. Intrinsically we would be able to map out the tree unlike the black box from neural networks. This makes it a potentially attractive model for domains that require interpretability of results from tabular data.

We decided to also study the effects of normalization on the results given that the model has some NN properties and might benefit. The paper itself does not mention whether normalization was used or not.

4.2 Dataset

In our mini-study we are using the following sample datasets from the article. The datasets were taken from Kaggle and might not be the same as the ones used in the paper. The architecture of each model is detailed below.

- Iris: consists of three iris species with 50 samples each as well as some properties of each flower [7].
- Car Evaluation: consists of multiple structures: acceptability, price, buying price, maintenance cost, technical characteristics, comfort, number of doors, capacity, luggage boot size, and safety [2].
- Titanic: consists of social class (first class, second class, third class, crew member), age (adult or child), sex, and whether or not the person survived [8].
- Breast Cancer Wisconsin: consists of features from a digitized image of a fine needle aspirate (FNA) of a breast mass describing characteristics of the cell nuclei present in the image [1].
- Pima Indian Diabetes: consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on [4].
- Haberman’s Survival: consists of age of patient at time of operation (numerical), patient’s year of operation, number of positive axillary nodes detected (numerical), survival status (class attribute)[6].

Dataset	**#inst.**	**#feat.**	**#cl.**
Iris	150	4	3
Haberman's Survival	306	3	2
Car Evaluation	1728	6	4
Titanic	714	10	2
Breast Cancer Wisconsin	683	30	2
Pima Indian Diabetes	768	8	2

Table 1: Summary of Datasets

4.3 Hyperparameters

We trained the datasets using the three algorithms: DNDT, DT, and NN following the architecture design of the authors by following their descriptions. We used the same hyperparameters, except temperature parameter which we used in order to test their hypotheses. The effect of temperature was not explored in this paper.

4.4 Experimental setup and code

We attempted both TensorFlow and PyTorch, as provided on Github (see [5]), however due to TensorFlow version incompatibility, we decided to work with PyTorch.

5 Results

5.1 Accuracy

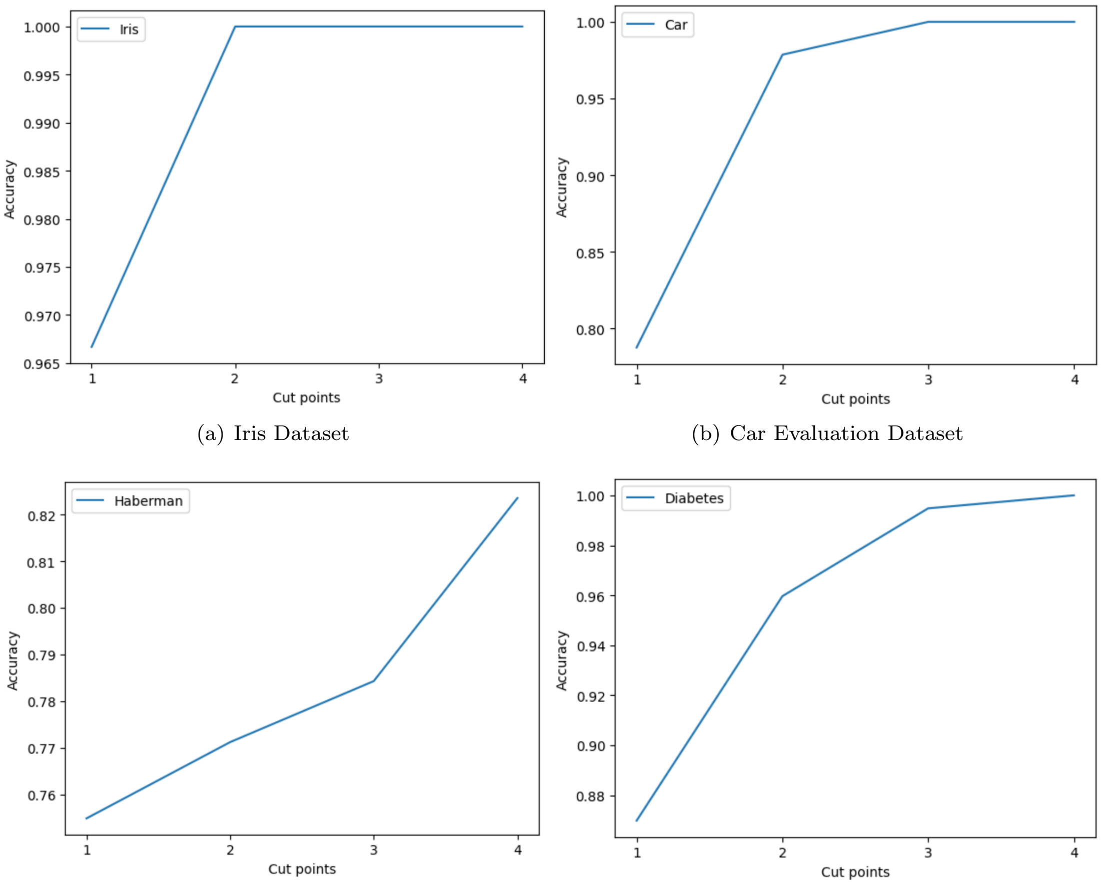
Table 2: Accuracy results of each model

Dataset	**DNDT**	**DNDT**	**DNDT**	**DNDT**	**DT**	**NN**
	(un-normalized)	**(normalized)**	**(un-normalized**	**(normalized**		
			& Split)	**& Split)**		
Iris	96	96	**100**	96.7	96.7	80
Car Evaluation	77.4	78.8	76	78.3	**98**	68.8
Titanic	63.6	63.6	61.9	61.9	**100**	61.9
Breast Cancer Wisconsin	93.3	95.8	94.7	93.9	91.2	**98.2**
Pima Indian Diabetes	68.6	**87**	63.6	74.7	70.1	75.3
Haberman's Survival	73.5	**75.5**	72.6	74.2	74.2	72.6

The reported accuracy results in Table 2 highlight the comparative performance of different models, with bolded numbers indicating superior outcomes. The findings reveal that both DNDT and DT exhibit better performance than NN, with each securing two wins over six comparisons. Notably, normalized DNDT demonstrates superior performance compared to its non-normalized counterpart. However, a nuanced analysis, considering that the DNDT model underwent testing across different experiments while only one instance of the DT model was tested, leads to the conclusion that DT outperforms DNDT, which aligns with findings in the paper. This superiority is attributed to the predominantly tabular nature of the datasets and the relatively low feature dimension.

5.2 Active cut-points

In this section, we focus on analysing the effects of cut-points on four datasets; Car Evaluation, Pima, Iris, and Haberman's. We set the number of cut points per feature between 1 to 4. As illustrated in Figure 1, all datasets follow the same trend, as the number of cut-points increases, it becomes more stable after a certain value, which is similar to the paper results. We found the Iris dataset to be identical to the paper. The Haberman's Survival dataset takes more cut points, after 4, while the Pima Indian Diabetes Datasets provided better accuracy after 2 cut-points, as compared to 4 in the paper. The results reassure that large DNDTs do not over-fit the training data, even without explicit regularisation.



(a) Iris Dataset

(b) Car Evaluation Dataset

(c) Haberman's Survival Dataset

(d) Pima Indian Diabetes Dataset

Figure 1: Number of cut points per feature

5.3 Tree structure

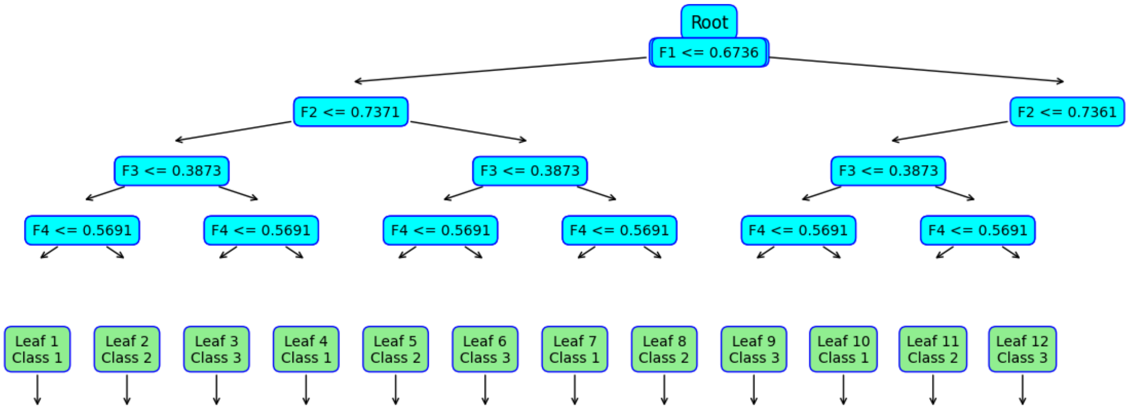


Figure 2: DT view of Iris Dataset

Here we plotted the DNDT from the Iris dataset after having normalized and split the data. The structure and methodology of the tree can be outputted, hence we can interpret the process with much more each compared to a neural network. This ability is potentially valuable to sectors that require interpretability.

6 Results reproducing original paper

- Claim 1: If we compare a normalized and split DNDT we notice that it is more accurate than our NN. Additionally given that the model is a tree, we have a clear way to interpret the process in which the classification is made. This is a clear advantage to NN which is widely considered a 'Black Box'
- Claim 2: We were able to confirm this claim. Although it was quite clear from the start that it would be true. Given the design of DNDTs, we can quickly see that GPU acceleration would be easy to utilize through libraries such as Pytorch and TensorFlow.

7 Discussion and Conclusion

In this mini-project, we explored reproducibility in Machine Learning to validate results, ensuring scientific rigor, and promoting transparency. During our experiment, we found several issues associated with the paper. Notably,

- Inconstancy in code on Github for both TensorFlow and PyTorch models. The temperature was not used in the PyTorch model while it was used for the TensorFlow model. This produced inconsistent results.
- The paper did not study the effects of temperature on the accuracy.
- The parameters and design of the NN used were not mentioned, and alternative architectures were not explored.
- There was no mention of the normalization and test/train split used for the DNDT. Additionally, in the code provided, there was no split or normalization implemented.

Overall, our goal of reproducibility of a scientific work was successful. experiments validated both Claims, as seen in Section 6 above. We were able to interpret and follow the process described in the paper and attempt to reach the same conclusions.

8 Statement of Contributions

- Jonathan Halimi - Implemented DNDT, DT, and wrote this report
- Thien Pham - Wrote this report
- Rafid Saif - Implemented the Neural Network, plots, and wrote this report

References

[1] *Breast Cancer Dataset*. URL: https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data.

[2] *Car Evaluation Dataset*. URL: https://www.kaggle.com/datasets/elikplim/car-evaluation-data-set.

[3] *Deep Neural Decision Trees (2018)*. URL: https://arxiv.org/pdf/1806.06988v1.pdf.

[4] *Diabetes Dataset*. URL: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.

[5] *Github Code*. URL: https://github.com/wOOL/DNDT.

[6] *Haberman's Survival Dataset*. URL: https://www.kaggle.com/datasets/gilsousa/habermans-survival-data-set.

[7] *Iris Dataset*. URL: https://www.kaggle.com/datasets/uciml/iris.

[8] *Titanic Dataset*. URL: https://www.kaggle.com/datasets/brendan45774/test-file.