Gaining interpretability: Deep neural networks as teachers for decision trees

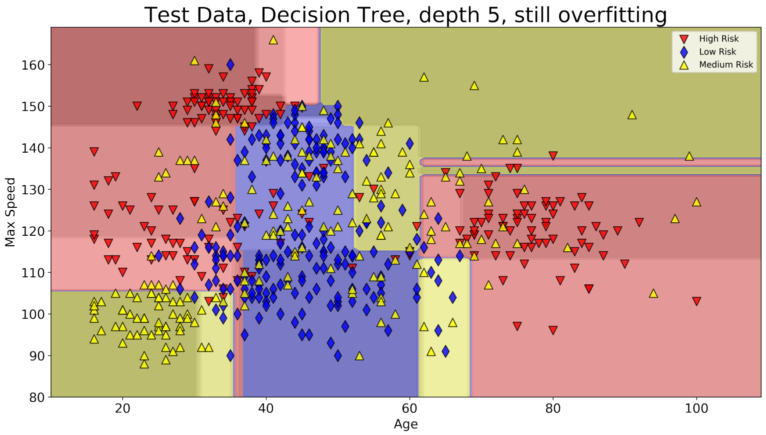
Anonymous authors
Paper under double-blind review

Abstract

Deep neural networks have outstanding performance and flexibility and can be regularized to generalize well on pretty much any data set. However, without additional work, they are black boxes and how they come to conclusions is not transparent or comprehensible. But exactly this right to explanation is well established by Europe's GDPR, United States' credit score, and many other real world applications. Additionally, interpretability helps debugging and evaluating the performance of a model. On the opposite side, decision trees can be much more comprehensi- ble, and can be trained either towards high understandability (simple tree) or high accuracy (complex tree). Unfortunately, unlike neural networks they tend to over- fit when trained on real world data and are hard to regularize. In this contribution I will show how training decision trees on data generated by a neural network gives us a dial to be tuned between predictive power on one side and interpretability and stability on the other side.

1 Motivation and Problem Statement

Decision trees are one type of machine learning model that allow for interpretation given the tree has low complexity. Unfortunately decision trees tend to overfit when trained on real world data. Real world data often comes from a combination of distributions that largely differ in density of samples. Some parts may be covered by a lot of samples, others just by few. So small variations in training data have a high impact on the tree.



Test Data, Decision Tree, depth 5, still overfitting

Figure 1: Decision Boundaries by regularized decision tree.

Figure 1 shows the classification results of a decision tree for our example use case where data looks scattered in a way we just described it. From two variables we want to lean the risk class of a driver getting into a car accident given the age of the driver and the top speed of the car driven. You see the test data plottet in the foreground while the decision boundaries are plottet as the background. Darker background colors indicate higher probabilities of the prediction. Even though we apply strong regularization, there are parts of the decision tree that are simply do complicate. Thus the decision tree overfits in a way that does not allow for a good interpretation story.

So the problem at hand can be stated like this: can we keep the interpretability of a decision tree on one side and mix it with the generalization power of a deep neural network on the other side?

2 Approach

We start with a deep neural network as our black box model and train it to high accuracy and generalization as shown in figure 2. In the next step to make our model interpretable we replace it with a regularized decision tree as a global surrogate model. We use the black box model to generate a new training data set by feeding in an equidistant grid of samples over the domains of our input values and use the predictions as our new target variable. This is used as the new training data to approximate the predictions of a black box model.

It is important that we do not propose to use the deep neural network for prediction afterwards, but we only use it as an intermediate means to come up with a decision tree which can all by itself be made interpretable. We thus do not need to pay too much attention to making our deep learning model compatible with a decision tree. We are however aware of means to such regularisations as proposed by (Schaaf et al., 2019) and also some work done by the author himself.

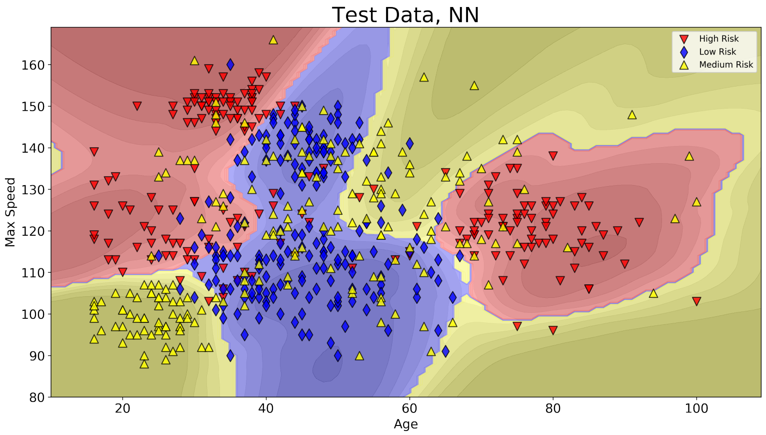


Figure 2: Decision Boundaries drawn by deep neural network, 72% accuracy on test and training data.

As it turns out the only thing that matters for our approach is that the network is properly regularized, but not so much how this is achieved. I ended up using ”Self-Normalizing Neural Networks” as proposed by (Klambauer et al., 2017) in combination with standard L1-regularization on the activation level. Decision Boundaries of a model trained that way are shown in figure 2.

3 Results

In figure 3 you can see the predictions of the resulting surrogate decision tree that has been tuned for interpretability. Setting the maximum depth of the tree gives us a dial between a model as accurate as the original black box model or as interpretable as the model you are seing. So, practically the findings of our work do not back up (Rudin, 2018) claim that there is no trade-off between accuracy and interpretability. Even more this work is based on the contrary belief. To me understanding this is largely due to the unsual definition of accuracy she uses. I, however, follow her in her suggestion that explainable models that do not replace the black box models are useless at best. This work also contradicts in the judgement that instability of an interpretable model is a good thing. We will show how our approach makes models more stable even though there is room for improvement.

The decision tree that could replicate the blackbox model by 100% has a maximum depth of 12, while the one shown works for interpretation has three levels only and significantly less accuracy

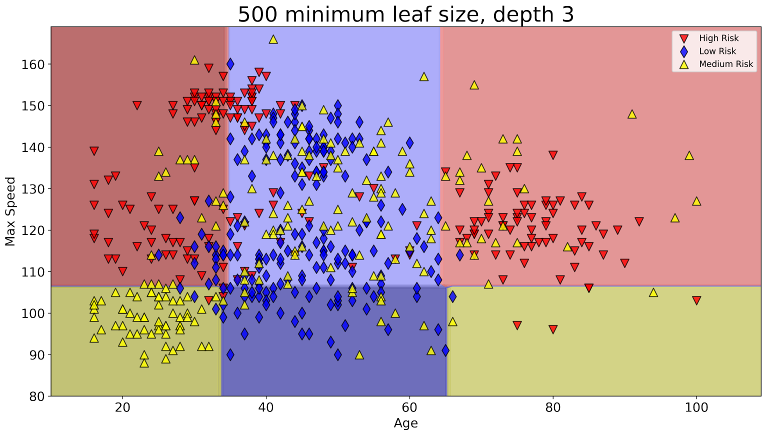
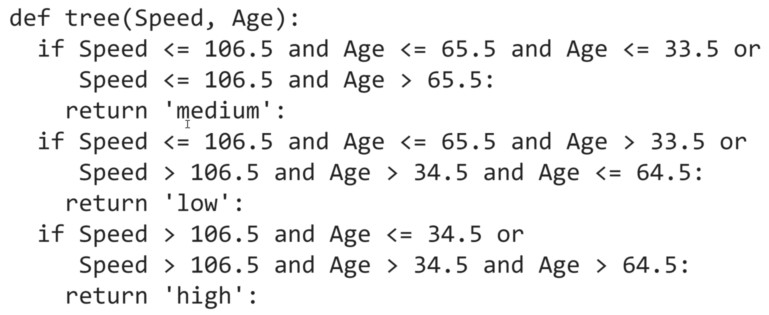


Figure 3: Decision Boundaries by shallow surrogate decision tree, still 64% accuracy.

(64% vs 72%). In the surrogate model we observe the same amount of overfitting as in the blank box neural network (namely none) as the surrogate decision tree overfits on what the black box models presents it - which is already regularized. This also positively impacts the stability of our decision tree.

Having a model like this also allows for generation of if clauses similar to what humans would write as business logic. Figure 4 shows one of many ways to turn a shallow decision tree into code. It would be just as easy to create code having nested if statements instead of one combined condition for each possible prediction. All the generated code blocks are equivalent in power as our decision tree and could used to completely replace it. You could even bring such a code based ”model” into production.



```
def tree(Speed, Age):
  if Speed <= 106.5 and Age <= 65.5 and Age <= 33.5 or
     Speed <= 106.5 and Age > 65.5:
    return 'medium':
  if Speed <= 106.5 and Age <= 65.5 and Age > 33.5 or
     Speed > 106.5 and Age > 34.5 and Age <= 64.5:
    return 'low':
  if Speed > 106.5 and Age <= 34.5 or
     Speed > 106.5 and Age > 34.5 and Age > 64.5:
    return 'high':
```

Figure 4: Business logic rules automatically generated from surrogate model.

4 Conclusions

The best known way of interpreting a prediction made by a decision tree is to look at the path chosen as shown in figure 5. The information provided for this example already is quite complex, but matches what you see in figure 3. Young drivers (20) with relatively fast cars (110) tend to have more accidents. This is what you can directly read off of the plot. Similar thougs can be made for the other 6 leaf nodes of the tree, e.g. people within a certain range of age are unlikely to have a lot of accidents regardless of the max speed of their cars. Speaking variables, low complexity and shallowness of the tree are a precondition to interpretation, though.

Practical issues arise around exactly this area of regularizing the tree to low complexity. Next to good accuracy you would also want stability of the tree. Decision trees are high variance, which means the parameters are very sensitive to small changes in the input. Since it is hard to impossible

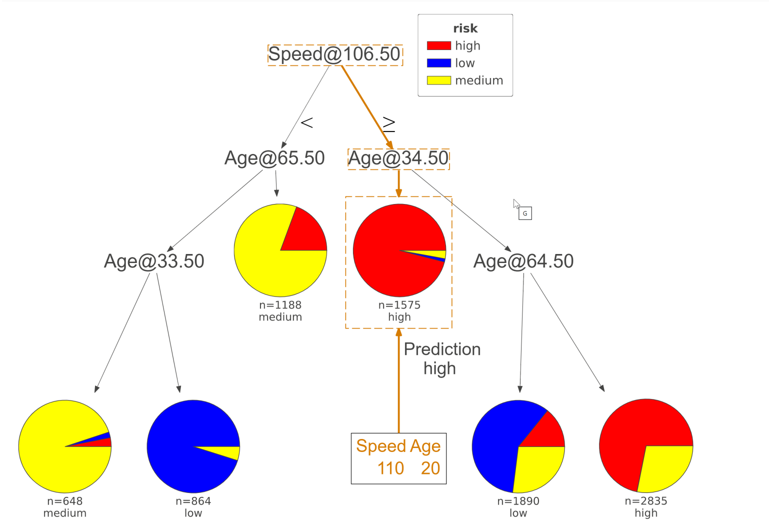


Figure 5: Prediction path featuring all kinds of information for interpretation.

to make training of neural networks totally deterministic each training run will generate slightly differnt input data for the decision tree potentially leading to drastic changes in their split points and even overall structure. This is undeseriable as it makes interpretation much harder. Best results so far arise from manual experiments restricting both the depth and minimum leaf size which results in stable results for this use case, but there is no evidence this will be the case for other use cases as well. Special measures to stabilize trees are proposed in (Arsov et al., 2019) and (Last et al., 2002).

ACKNOWLEDGMENTS

Thanks to Terence Parr for advising on how to regularize my decision trees and providing me with the `https://github.com/parrt/dtreeviz` tool used to plot figure 5. Thanks to Mikio Braun for helping me to write this short paper.

REFERENCES

Nino Arsov, Martin Pavlovski, and Ljupco Kocarev. Stability of decision trees and logistic regression, 2019.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks, 2017.

Mark Last, Oded Maimon, and Einat Minkov. Improving stability of decision trees. *IJPRAI*, 16: 145–159, 03 2002. doi: 10.1142/S0218001402001599.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, 2018.

Nina Schaaf, Marco F. Huber, and Johannes Maucher. Enhancing decision tree based interpretation of deep neural networks through l1-orthogonal regularization, 2019.