DEEP NEURAL DECISION TREES

Yongxin Yang, Irene Garcia Morillo ,Timothy M. Hospedales

https://arxiv.org/pdf/1806.06988.pdf

Summary:

Deep neural networks are some of the most powerful models when it comes to predictive analysis. They usually provide very high accuracy of prediction. They have the ability to process and learn highly perceptual data such as images and audio, something that not many models are good at. However, the major drawback of any NN is the fact that it is not very interpretable.

Tree based learning models are highly popular and are extensively used in processing tabular data. Ensembles of trees are extremely powerful models in their own right. Tree based models , by design are highly interpretable .

Strong accuracy of prediction and the abilty to learn perceptual data are very important characteristics of a model. However, interpretabilty of a model can prove to be very important as well, especially in certain cases. Any case which requires to bring out factors and decisions affect the outcome of prediction, requires the model to be interpretable. Cases in the field of business are the prime examples. Cases which involve ethical

considerations to a big extent,such as in the fields of medicine or law, require the model to be interpretable as well, as they might necessitate explanations.

This paper proposes a model which tries to incorporate the working of a Deep Nueral network with the interpretabilty of a decision tree. In overview, any set of weights of the network , corresponds to a decision tree.

The properties of the model are such:

1. No alternative optimization techniques for parameter learning, all learning is done through a single pass of back propogation.
2. A differential binning function is applied which can split a single node into multiple leaves and not neccesarily a binary split.
3. The model is designed specifically for interpretabilty, especially on tabular data.

Usually , convential decision trees use recursive greedy methods to learn. The model proposed by this work proposes a different and more optimal method. It searches for the structure and parameters of the tree using Stochastic gradient descent.

<u>Soft binning function:-</u>

assumptions: 1) x is a continous variable
2) the number of intervals is n+1

Requires n cut points , which are essentially learnable parameters.

Represented by : [β1,β2,...,βn]

A single layered neural network is constructed with softmax as its activation.

π= softmax((wx+b)/τ)

W is taken as a constant, [1,2,...n+1]

b= [0,−β1,−β1−β2,...,−β1−β2⋯−βn].

τ is the temperature factor.

The a above nn produces a nearly one-hot encoded binning of x.

<u>Kronecker Product</u>

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}.$$

<u>Making predictions:-</u>

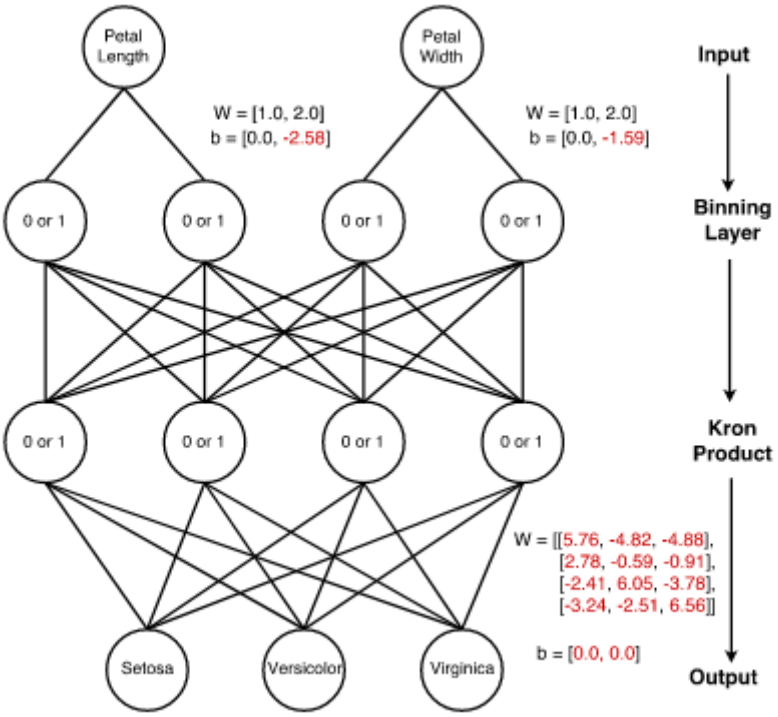
Each feature of the data set is binned and the one hot vector Z is arrived at using the kronecker product.

$z = f_1(x_1) \otimes f_2(x_2) \otimes \cdots \otimes f_D(x_D)$.
Where xi represents a feature.

Z represents the index of the leaf in which the data instance x is present.

Training:-

Since all steps of our forward propogations are differentiable, all parameters (Fig. 2, red) can now be trained with Stochastic Gradient Descent.



Outcomes:-

DNDT, decision trees and neural networks were trained on the same datasets and their outcomes were compared.

Although no one model was dominantly better, the overall performance of decision tree model proved to be the best. This is because it best suited the data ,tabular and low dimensional, data on which DNDT, which is modelled as a nn was not expected to fare very well.

It was found that the feature selection of both the DNDT and DT models were similar.

Basically , it showed that DNDT was performing considerably close to that of a DT but not better. It could be due to the nature of data.
It provides better performance on tabular data as compared to an NN along with more interpretabilty.

Other points to note:-

DNDT is scalable with increase in number of instances, due to mini-batch training.

DNDT is not scalable with increase in number of features due to the use of the kronecker product.This is one of its major drawbacks.