Deep Learning For Tabular Data

BILICI, M. Şafak
safakk.bilici.2112@gmail.com

UYSAL, Enes S.
enessadi@gmail.com

Contents

1 Deficiencies And Efficiencies Of Black Box Models

Using non-black box models for tabular data is preffered in industry. The main question is "why?". They representionally efficient for decision manifolds with approximately hyperplane boundaries which are common in tabular data. They are highly interpretable in their basic form and ensemble form: tracking decision nodes is easy, they even can be visualized. Lastly, they are fast to train [1].

On the other hand, there are several benefits for using "black box" models for tabular data: Alleviating the need for feature engineering, which is currently a key aspect in tree-based tabular data learning methods. Learning from streaming data and perhaps most importantly. Lastly, data-efficient domain adaptation [1].

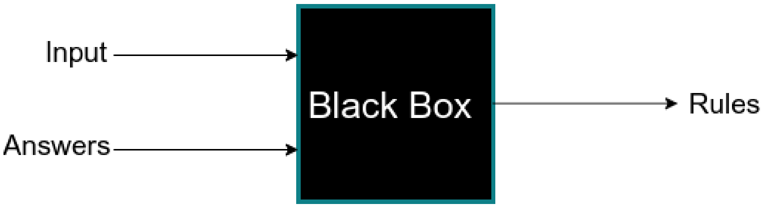
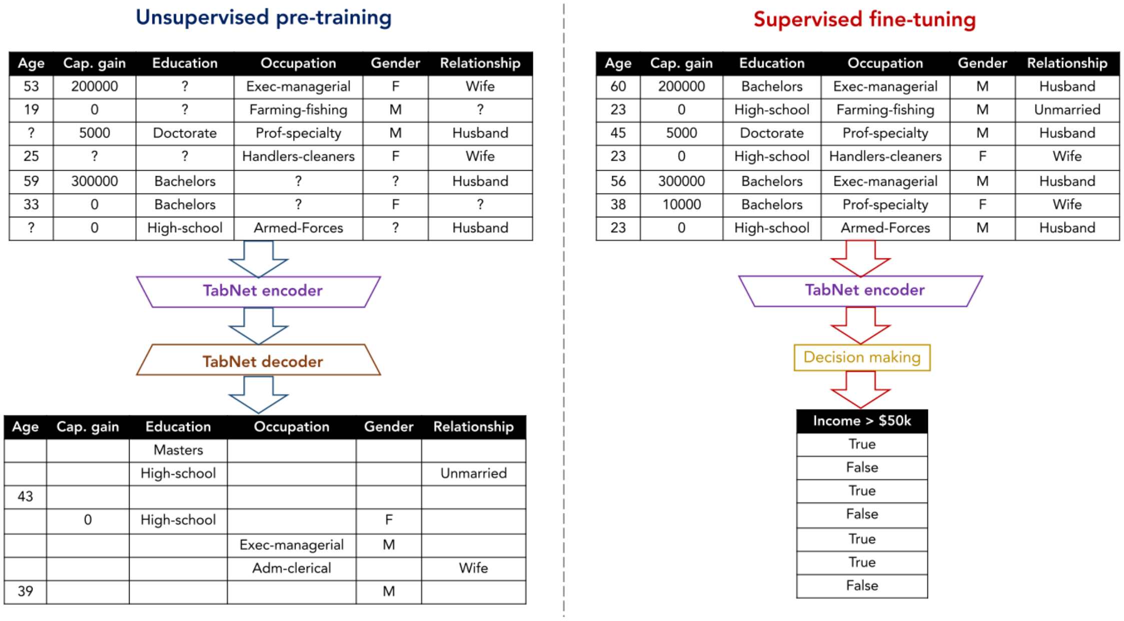


Figure 1: Black Box

2 TabNet: Attentive Interpretable Tabular Learning

TabNet [1] is proposed by Sercan Ö. Arık and Tomas Pfiste. TabNet is a Deep Learning model for tabular data. It has various advantages:

- It uses raw tabular data without any preprocessing or feature engineering.
- It uses sequential attention to choose which features to reason from at each decision step. This enables interpretability.
- Besides its supervised fashion, it can be unsupervised pre-training method by predicting masked features.



Age	Cap. gain	Education	Occupation	Gender	Relationship
53	200000	?	Exec-managerial	F	Wife
19	0	?	Farming-fishing	M	?
?	5000	Doctorate	Prof-specialty	M	Husband
25	?	?	Handlers-cleaners	F	Wife
59	300000	Bachelors	?	?	Husband
33	0	Bachelors	?	F	?
?	0	High-school	Armed-Forces	?	Husband

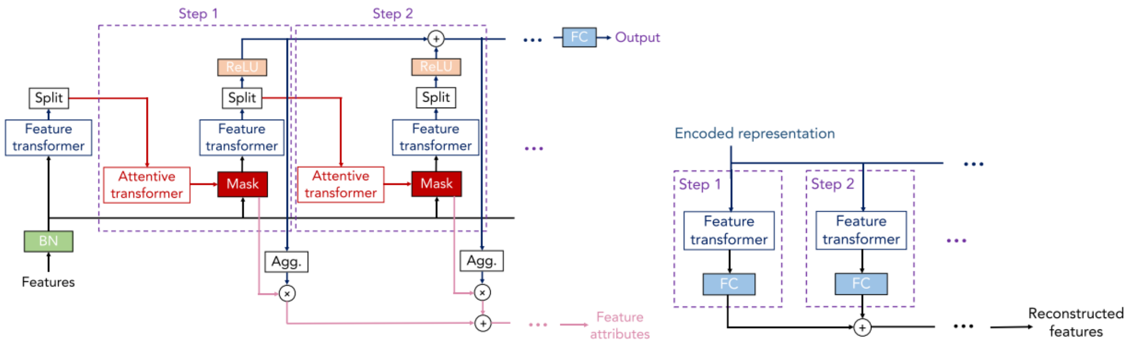
Age	Cap. gain	Education	Occupation	Gender	Relationship
		Masters			
		High-school			Unmarried
43					
	0	High-school		F	
			Exec-managerial	M	
			Adm-clerical		Wife
39				M	

Age	Cap. gain	Education	Occupation	Gender	Relationship
60	200000	Bachelors	Exec-managerial	M	Husband
23	0	High-school	Farming-fishing	M	Unmarried
45	5000	Doctorate	Prof-specialty	M	Husband
23	0	High-school	Handlers-cleaners	F	Wife
56	300000	Bachelors	Exec-managerial	M	Husband
38	10000	Bachelors	Prof-specialty	F	Wife
23	0	High-school	Armed-Forces	M	Husband

Income > $50k
True
False
True
False
True
True
False

Figure 2: TabNet's Training Procedure [1]

2.1 Model Architecture



(a) TabNet encoder architecture (b) TabNet decoder architecture

Figure 3: TabNet Architecture [1]

TabNet uses sparse instance-wise feature selection learned from data. Also, it constructs a sequential multi-step architecture, where each step contributes to a portion of the decision based

on the selected features. Lastly, it improves the learning capacity via nonlinear processing of the selected features.
TabNet does not require any normalization process. It apply batch-normalization at first level of learning.

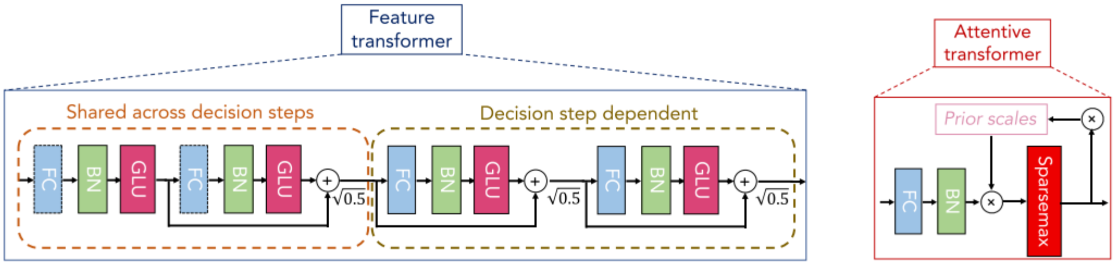


Figure 4: Feature Transformer And Attentive Transformer [1]

2.2 Learning Salient Features

TabNet uses SGD. Each minibatch has dimension of $\mathbb{R}^{\mathcal{B}\times N}$. For soft selection of salient features, TabNet employs learnable mask $\mathbf{M}[\mathbf{i}] \in \mathbb{R}^{\mathcal{B}\times N}$. This masking is multiplicative: $\mathbf{M}[\mathbf{i}] \cdot \mathbf{f}$.
This masks are obtained from TabNet's attentive transformer part by using processed features from the preceding step $\mathbf{a}[\mathbf{i}-\mathbf{1}]$:

$$\mathbf{M}[\mathbf{i}] = sparsemax(\mathbf{P}[\mathbf{i}] \cdot h_i(\mathbf{a}[\mathbf{i}-\mathbf{1}])) \tag{1}$$

sparsemax [3] is defined as

Algorithm 1 sparsemax

```
Input: $\mathbf{z}$
sort $\mathbf{z}$ as $z_{(1)} \geq z_{(2)} \geq ... \geq z_{(K)}$
find $k(\mathbf{z}) \leftarrow \max\{k \in [K] \mid 1 + k \cdot z_{(k)} > \sum_{j \leq k} z_{(j)}\}$
define $\tau(\mathbf{z}) = \frac{(\sum_{j \leq k(\mathbf{z})})-1}{k(\mathbf{z})}$
return $p_i = [z_i - \tau(\mathbf{z})]_+$
```

With sparse probabilistic normalization step of sparsemax provides

$$\sum_{j=1}^{N} \mathbf{M}[\mathbf{i}]_{b,j} = 1 \tag{2}$$

The term h_i in (1) is a trainable function, shown in Figure 5: fully connected layer followed by BN. The prior scale $\mathbf{P}[\mathbf{i}]$ defined as:

$$\mathbf{P}[\mathbf{i}] = \prod_{j=1}^{i} (\gamma - \mathbf{M}[\mathbf{j}]) \tag{3}$$

where γ is relaxation parameter. $\gamma = 1$ means a feature is enforced to be used only at one decision step and as γ increases, more flexibility is provided to use a feature at multiple decision steps. $\mathbf{P}[\mathbf{0}]$ is initialized as all ones, $1^{\mathcal{B}\times N}$, without any prior on the masked features. To further control the sparsity of the selected features, TabNet proposes a sparsity regularization which is added to overall loss:

$$L_{sparse} = \sum_{i=1}^{N_{steps}} \sum_{b=1}^{\mathcal{B}} \sum_{j=1}^{N} \frac{-\mathbf{M}_{b,j} \cdot \log(\mathbf{M}_{b,j} + \varepsilon)}{N_{steps} \cdot \mathcal{B}} \tag{4}$$

2.3 Attention Transformer

From Sebastien Fischman's talk:

- Instance-wise feature selection
- Built-in explainability derived from masks
- Efficient Learning Capacity

TabNet's feature selection masks can shed light on the selected features at each step. If $\mathbf{M}_{b,j}[i] = 0$, then j^{th} feature of b^{th} sample should have no contribution to the decision. This procedure provides interpretability. The masks can be averaged and give instance-wise interpretation for the model's output.

2.4 Feature Transformer Block

From Sebastien Fischman's talk:

- Masic MLP block with Gated Linear Unit Activation $GLU(x) = \sigma(x) \cdot x$
- GLU control what information will be passed to the following layer.
- Shared layers across different steps.

3 TabTransformer: Tabular Data Modeling Using Contextual Embeddings

TabTransformer [2] is proposed by Xin Huang and his colleagues. The TabTransformer is built upon self-attention based Transformers. The Transformer layers transform the embed- dings of categorical features into robust contextual embed- dings to achieve higher prediction accuracy [2].

TabTransformer applies a sequence of multi-head attention-based Transformer layers on parametric embeddings to transform them into contextual embeddings, bridging the performance gap between baseline MLP and GBDT models.

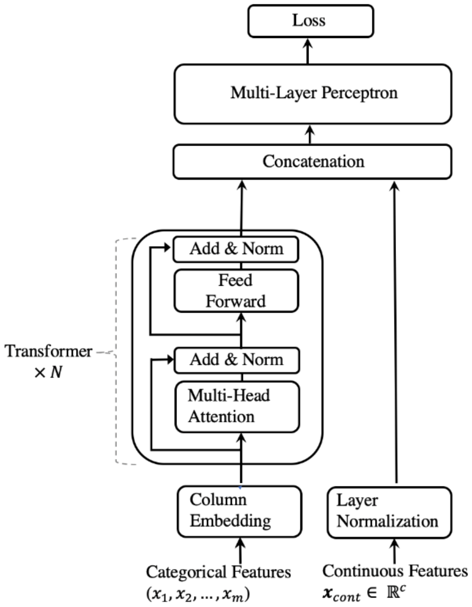


Figure 5: TabTransformer [2]

4 Take A Look

- TaBERT, Yin et al., 2021
- TABBIE, lida et al., 2021
- TAPAS, Herzig et al., 2020
- VIME, Yoon et al., 2020
- Neural Oblivious Decision Ensembles, Povov et al., 2020
- Gradient Boosting Neural Networks, Badirli et al., 2020
- DCN V2, Wang et al., 2020

References

[1] Sercan O. Arik and Tomas Pfister. *TabNet: Attentive Interpretable Tabular Learning*. 2020. arXiv: 1908.07442 [cs.LG].

[2] Xin Huang et al. *TabTransformer: Tabular Data Modeling Using Contextual Embeddings*. 2020. arXiv: 2012.06678 [cs.LG].

[3] André F. T. Martins and Ramón Fernandez Astudillo. *From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification*. 2016. arXiv: 1602.02068 [cs.CL].