TabPFN:
A Transformer That Solves Small Tabular Classification Problems In A Second

НИС МОП, 08.02.2023

Bayesian prediction

- Given: $D = (X_{train}, y_{train}), x_{test}$
- Given: set of hypotheses Φ
- Posterior predictive distribution (PPD):
  - Find: $p(y_{test}|D, x_{test})$
- Marginalize PPD over the posterior distribution of ϕ:

$$p(y_{test}|D, x_{test}) = \int_{\Phi} p(y_{test}|\phi, x_{test})p(\phi|D)d\phi$$

- If we know sampling procedure $p(D|\phi)$ and prior $p(\phi)$:

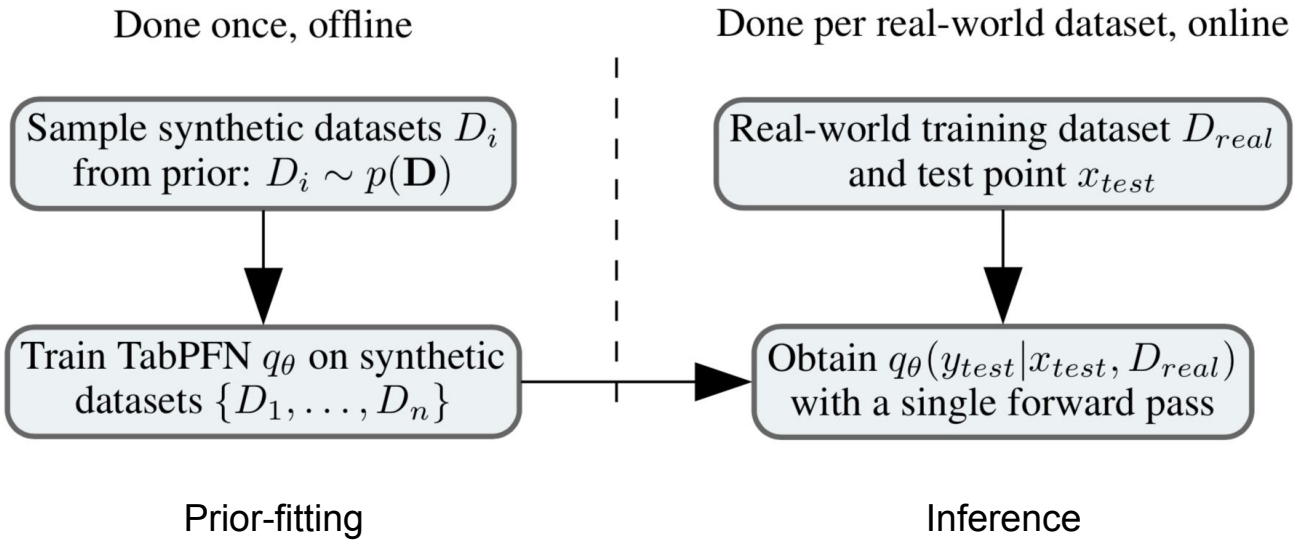
$$p(y_{test}|D, x_{test}) \propto \int_{\Phi} p(y_{test}|\phi, x_{test})p(|D|\phi)p(\phi)d\phi$$

Prior-Data Fitted Network: Prior-fitting

- Prior-Date Fitted Network = PFN
- PFN is a transformer trained to approximate the PPD.
- How to make it learn Φ?
- Prior-fitting:
  - Choose hypothesis with prior $p(\phi)$, generate dataset with $p(D|\phi)$
  - Fit datasets to “learn the prior”
- Authors introduce TabPFN, which works on tabular data.

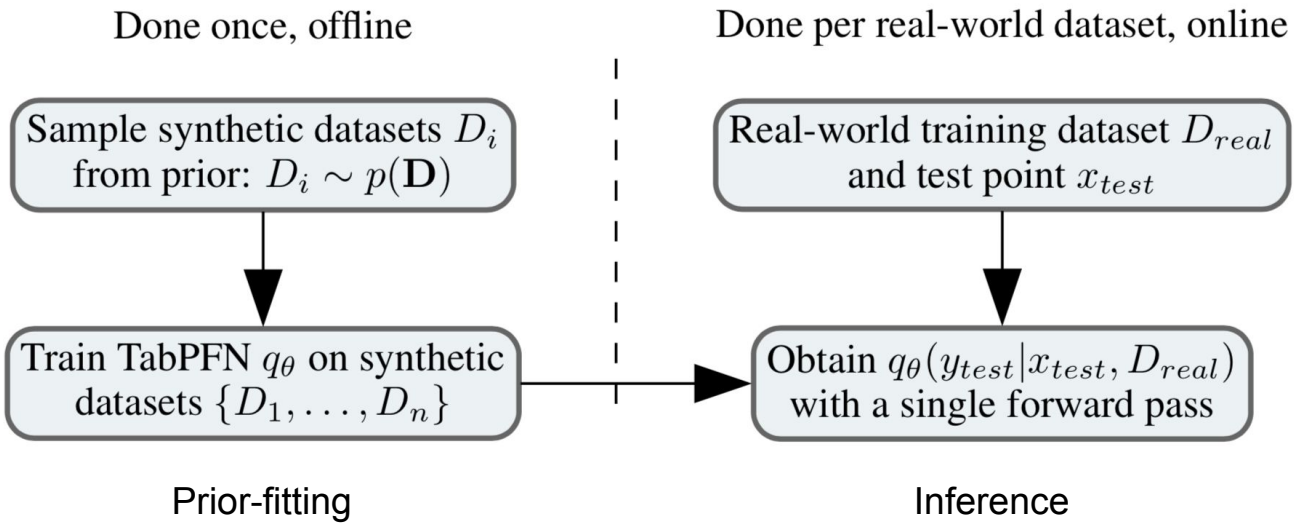
Prior-Data Fitted Network

- Authors introduce TabPFN, which works on tabular data.



Prior-Data Fitted Network

- The real-world datasets are fitted in a single forward pass
- Or in several forward passes with permutations:
  - shuffled feature columns and permuted class labels.



Which prior to use?

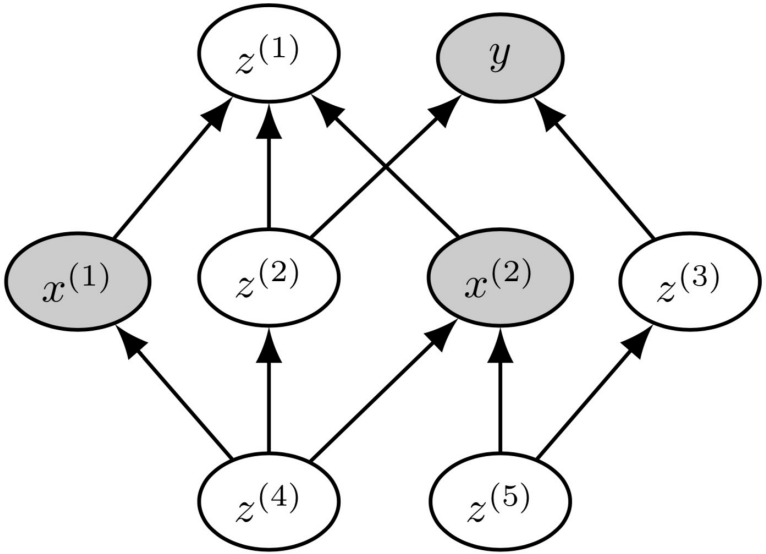
- Models for dataset generation:
  - Structural Causal Models
  - Bayesian Neural Networks

Structural Causal Models

- Generate a causal graph
- Each node n has a value:

$$f(Parents(n), \epsilon)$$

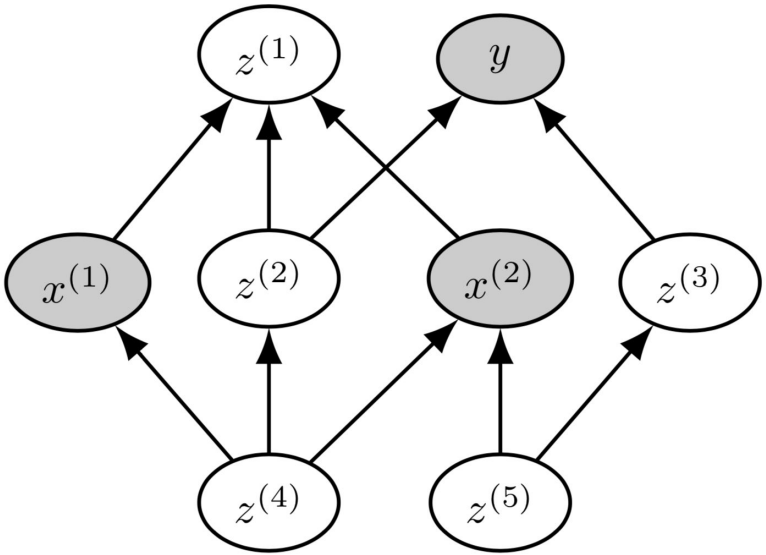
- where:
  - ϵ is random noise
  - f is a deterministic function
- Generate a dataset based on a graph (small, up to 1024 samples).



Example of a SCM graph

Structural Causal Models

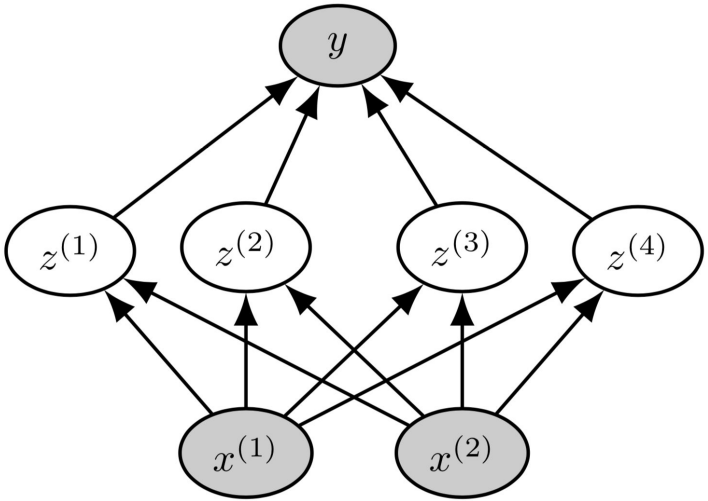
- It is actually regression.
- To turn *y* into a class label:
  - sample number of classes
  - sample boundaries from *y* values
- E.g., for 3 classes:
  - $(-\infty, 0], (0, 1], (1, +\infty)$



Example of a SCM graph

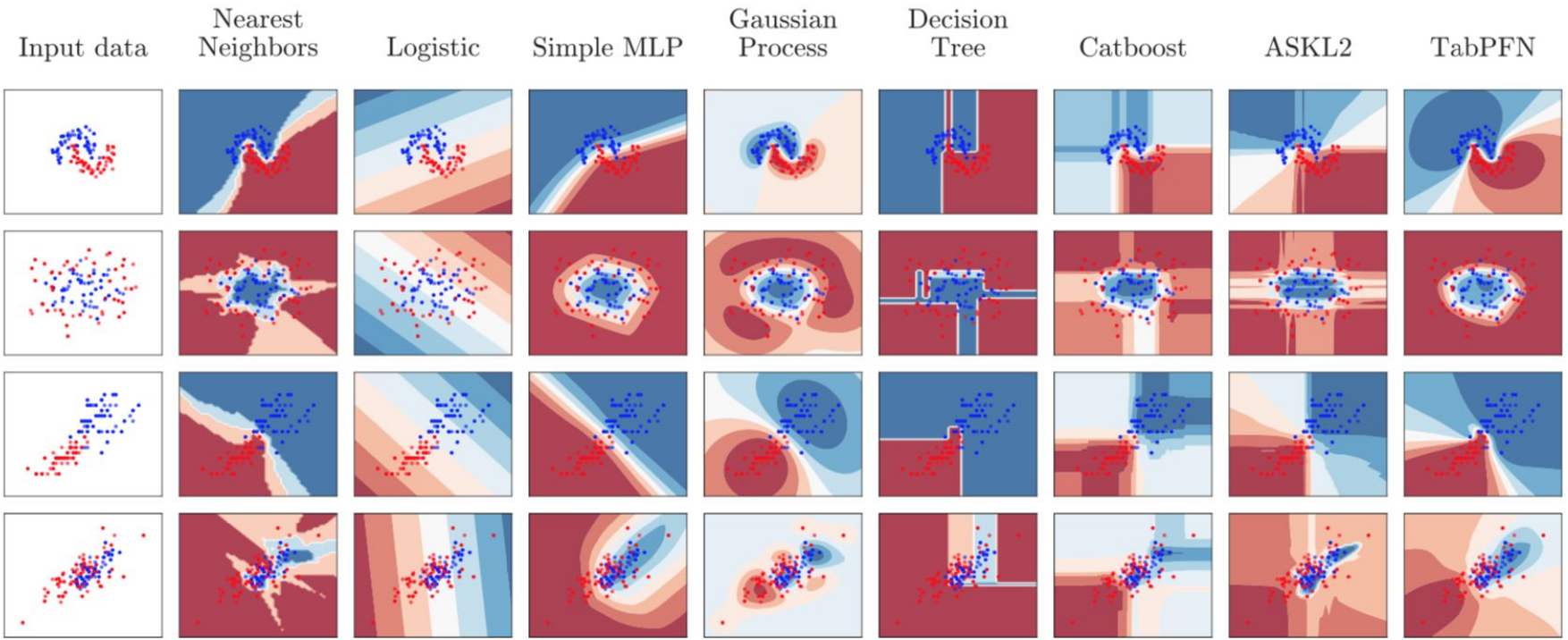
Bayesian Neural Networks

- Generate an input and a small NN.
- Weights of the BNN are distributions rather than fixed numbers.



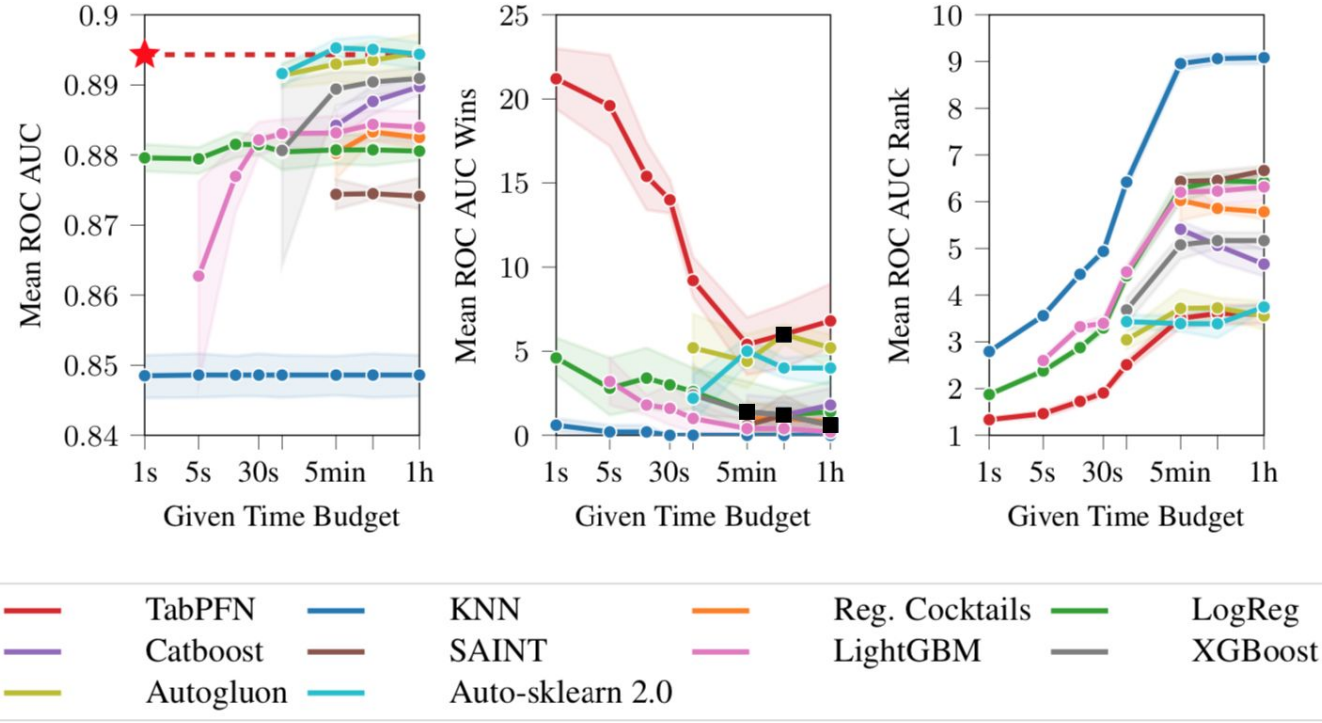
Example of a BNN

Experiments



Results on 2D datasets from sklearn: moons, circles, iris, wine

Experiments



Results on 30 small datasets (maximum 2000 samples, 100 features, 10 classes) from OpenML-CC18 suite

Experiments

	LightGBM	CatBoost	XGBoost	ASKL2.0	AutoGluon	TabPFN$_{n.e.}$	TabPFN	TabPFN + AutoGluon
Wins AUC OVO	0	0	2	2	2	4	**5**	**5**
Wins Acc.	0	2	2	3	3	0	6	**8**
Wins CE	0	1	3	1	7	1	6	**9**
M. rank AUC OVO	6.6167	4.9667	5.4167	4.05	3.7833	4.65	3.7	**2.8167**
Mean rank Acc.	6.5333	4.9833	5.1833	4.8667	3.8167	4.5333	3.6167	**2.4667**
Mean rank CE	5.7333	5.6	5.4667	5.8	2.8667	4.6167	3.5333	**2.3833**
Win/T/L AUC vs Tab..	5/4/21	9/4/17	6/5/19	10/6/14	13/4/13	4/8/18	–/–/–	15/7/8
Win/T/L Acc vs Tab..	6/0/24	9/1/20	11/0/19	11/2/17	12/0/18	6/3/21	–/–/–	19/3/8
Win/T/L CE vs TabP..	6/0/24	8/0/22	8/0/22	8/0/22	20/0/10	1/4/25	–/–/–	23/0/7
Mean AUC OVO	0.884±.012	0.89±.011	0.891±.011	0.894±.01	0.895±.01	0.891±.01	0.894±.01	**0.898**±.0097
Mean Acc.	0.815±.014	0.818±.011	0.821±.013	0.821±.016	0.83±.012	0.82±.013	0.825±.012	**0.834**±.011
Mean CE	0.782±.074	0.767±.061	0.758±.047	0.815±.06	**0.72**±.015	0.742±.021	0.732±.018	0.721±.015
Mean time (s)	3241	3718	3304	3601	3127	**0.8688** (CPU) **0.0187** (GPU)	24.57 (CPU) 0.4197 (GPU)	3152 (CPU) 3128 (GPU)

Results on 30 small datasets (maximum 2000 samples, 100 features, 10 classes)
from OpenML-CC18 suite

Sources

- https://arxiv.org/abs/2207.01848v3
- https://towardsdatascience.com/why-you-should-use-bayesian-neural-network-aaf76732c150