UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Stanford University

HyperFast: Instant Classification for Tabular Data

David Bonet, Daniel Mas Montserrat, Xavier Giró-i-Nieto, Alexander G. Ioannidis

TRL@NeurIPS 2023

Motivation

Traditional machine learning (ML), GBDTs & deep learning (DL) tabular methods

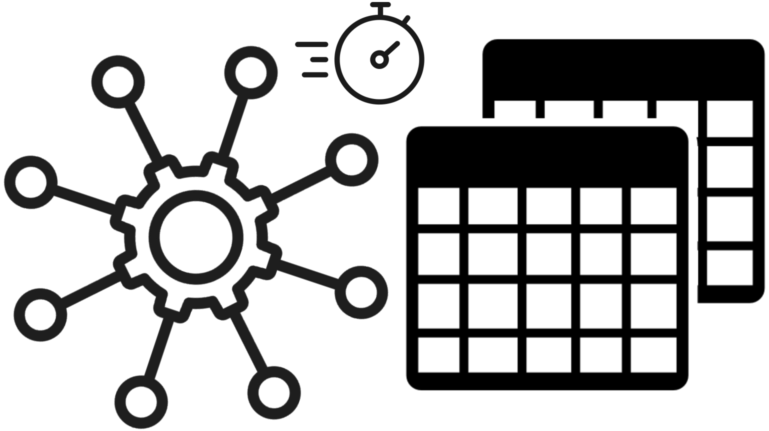
Time-consuming & computationally expensive training + hyperparameter tuning

TabPFN [1]: pre-trained Transformer for instant classification on small tabular datasets

Efficient method but very limited in dataset size.

Need for a tabular framework that satisfies:

1. Rapid model deployment
2. Single framework for multi-task/diverse datasets
3. Scalability to real world large-scale and high-dimensional datasets
   (e.g. biomedical and genomic datasets)



[1] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In The Eleventh International Conference on Learning Representations, 2023.

Hypernetwork

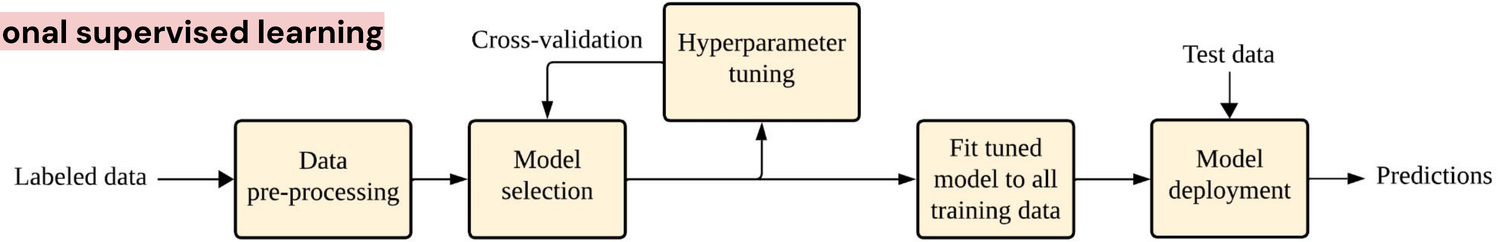
Idea: Train a high-capacity meta-model to encode task information in the prediction of the weights of a smaller model
How: Meta-train a **hypernetwork** on a wide collection of datasets

↳ Learns to generate the weights of a smaller model that performs the actual predictions

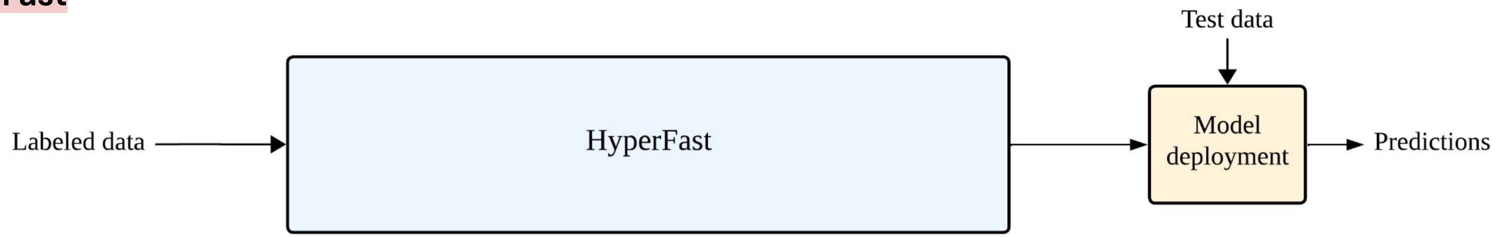
- The hypernetwork is trained just once.

- Once trained, many lightweight models generated by the hypernetwork can be used for deployment in different applications.

Traditional supervised learning



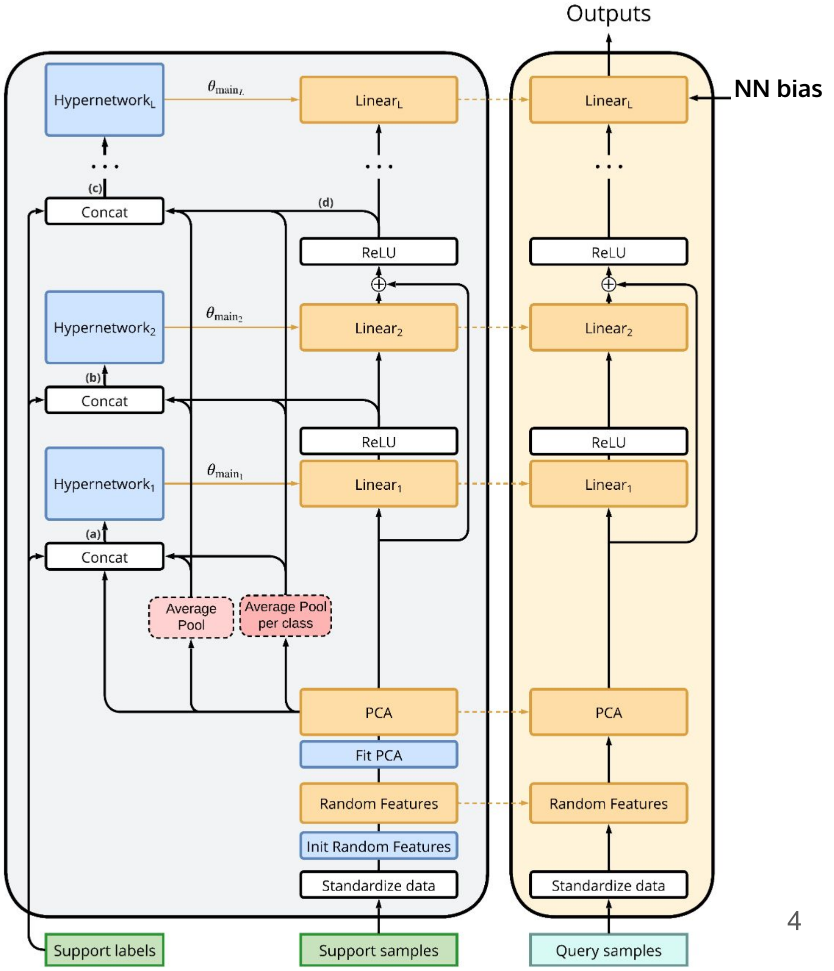
HyperFast



HyperFast

Architecture design

- **Random Features + PCA:** Initial transformation layers (low-rank approximation of Kernel PCA)
  - ✔ **Variable & high input dimensionality**
  - ✔ **Feature permutation invariance**
- **Hypernetwork modules:** Weight generation modules
  - ✔ **Sample permutation-invariant** pooling operations
- **Nearest Neighbor bias:** Retrieval-based component
  - ○ Add learnable parameters to the classification layer bias
- **Loss:** Cross-entropy loss of the (generated) main network evaluated on query samples.
  - ○ All trainable hypernetwork parameters are learnt end-to-end.



HyperFast

Meta-training stage

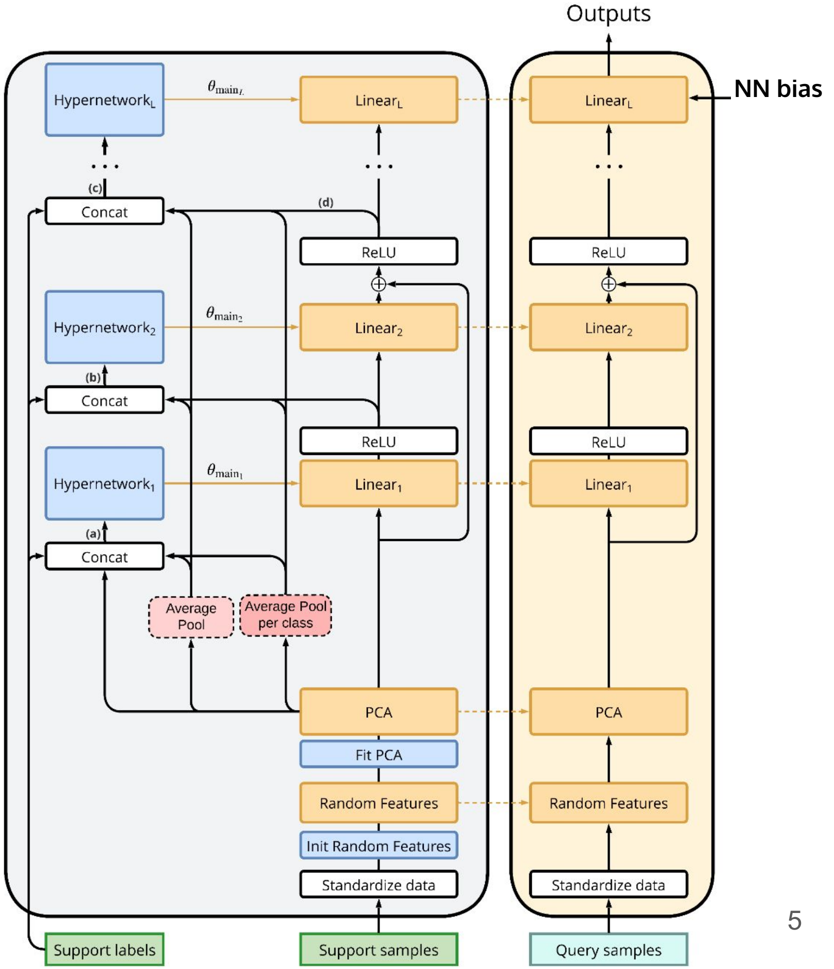
A wide variety of tasks are sampled

$$\mathcal{S}_t \subseteq d_{\text{train}},\ \mathcal{Q}_t \subseteq d_{\text{test}},\ t \in \mathcal{T}_{\text{meta-train}},\ d \in \mathcal{D}_{\text{meta-train}}$$

- Hypernetwork parameters are learnt
- Main model parameters are inferred

Meta-testing stage (inference)

1. Take labeled datapoints from an unseen dataset
2. Generate weights for the main model in a single forward pass
3. (Optional) Ensemble and/or optimize generated main networks
4. Do inference with the main model on the new unseen datapoints



Results

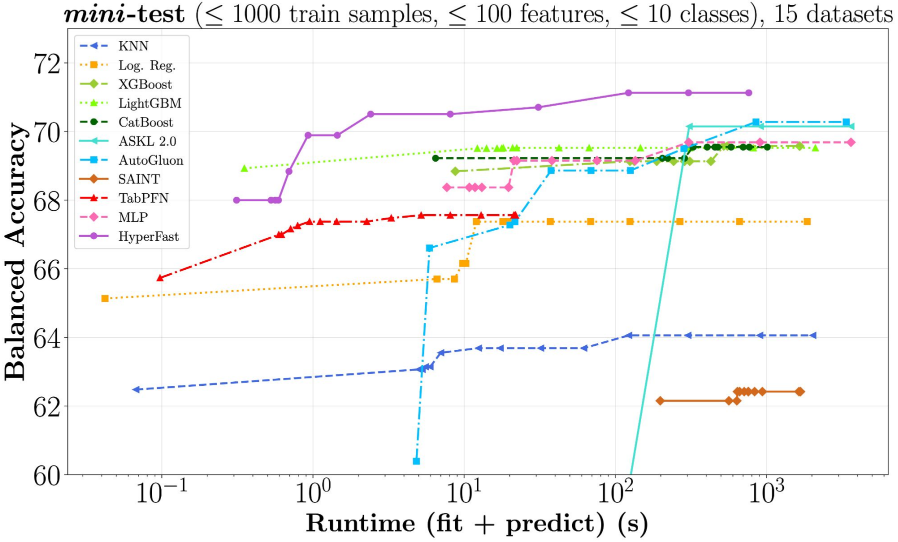
Evaluation on 15 tabular datasets from the OpenML-CC18 suite and genomic datasets.

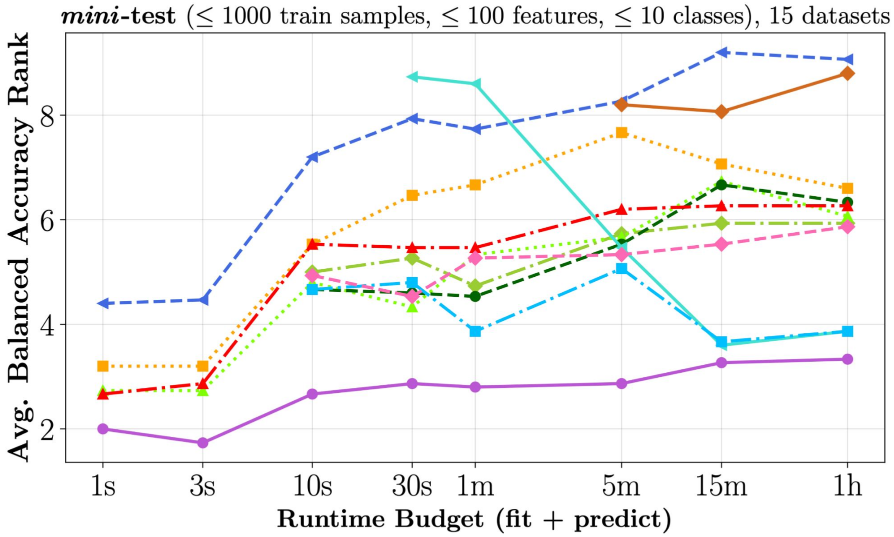
Big test: Original version of the datasets

Mini test: Version with ≤ 1000 training examples, ≤ 100 features and ≤ 10 classes

Results

Results on small-sized datasets


***mini*-test** ($\leq$ 1000 train samples, $\leq$ 100 features, $\leq$ 10 classes), 15 datasets

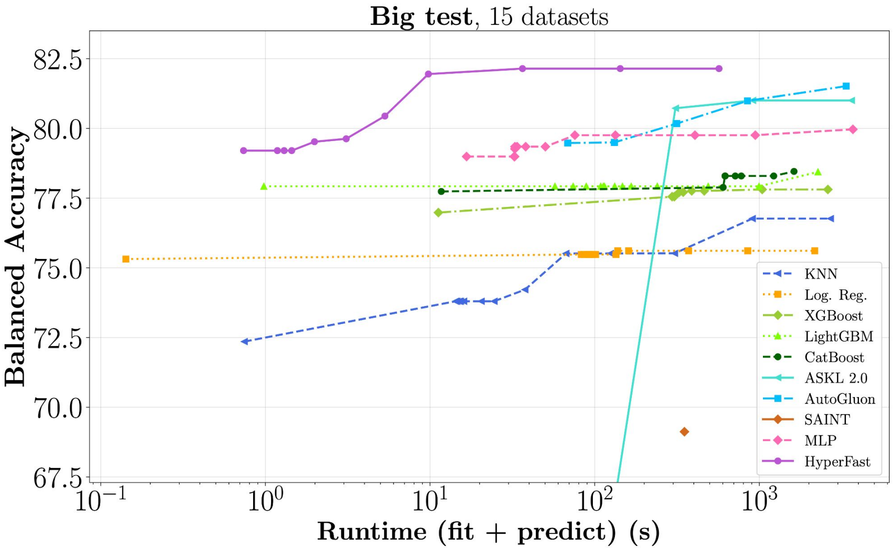

***mini*-test** ($\leq$ 1000 train samples, $\leq$ 100 features, $\leq$ 10 classes), 15 datasets

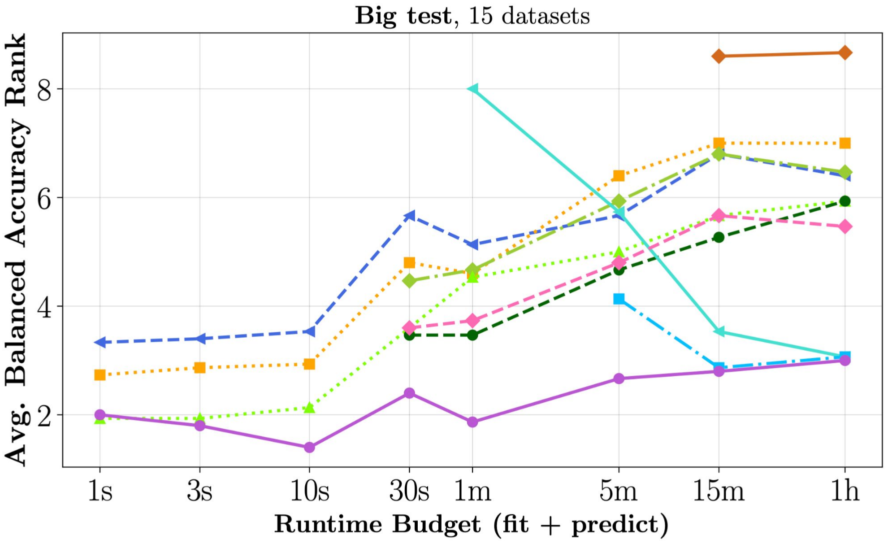
Results

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Stanford University

Results on large-scale datasets





Conclusions

HyperFast: meta-learned hypernetwork for instant classification of tabular data

- Lightweight model generation by the hypernetwork
- Adaptability to large datasets, different numbers of samples, features, and categories.

Ideal for:

- Rapid model deployment in different applications
- Computationally-constrained environments
- Fast inference priority

Easy to use **scikit-learn-like interface**:

```
from hyperfast import HyperFastClassifier

model = HyperFastClassifier()

model.fit(X_train, y_train)
predictions = model.predict(X_test)
```

AI-sandbox