Efficient Automated Deep Learning for Time Series Forecasting

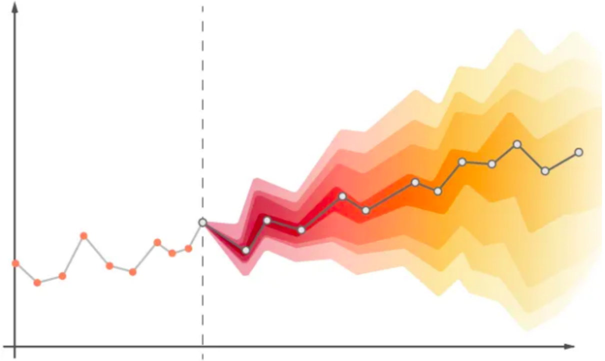
Marco Zanotti

University Milano-Bicocca

Contents

1. The TSF Problem

Time series forecasting (TSF) is the task of predicting future values of a given sequence based on previously observed values.



The TSF problem may be essentially identified by the following aspects:

- **Prediction objective**: point forecasting vs probabilistic forecasting
- **Forecast horizon**: short-term vs long-term forecasting
- **Input-Output dimension**: univariate vs multivariate forecasting
- **Forecasting task**: single-step vs multi-step forecasting

Nowadays, the TSF problem is usually faced with **Deep Learning models** (RNN, LSTM). However, the design of these models is a challenging task due to:

- **model architecture**: the choice of the model architecture is crucial for the performance of the model
- **hyperparameters**: the selection of the optimal hyperparameters is also extremely relevant for the model's performance
- **computational cost**: the search for the best model architecture and hyperparameters is resource expensive

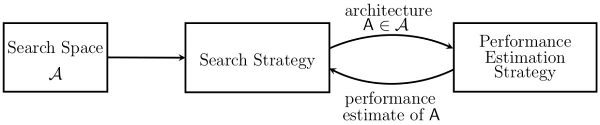
For this reason, the authors proposed a new open source framework for **Automated Deep Learning for TSF** that:

- uses **NAS** to search over a variety of state-of-the-art TSF architectures
- adopts **BO** to optimize the hyperparameters of the models (with a CASH approach)
- explores **multi-fidelity optimization** to reduce the computational cost of the search process.

2. NAS & HPO

Neural Architecture Search (NAS) is a technique that automates the design of neural network architectures and it is based on three main components:

- **Search Space** defines which architectures can be considered
- **Search Strategy** details how to explore the search space, dealing with the exploration-exploitation trade-off
- **Performance Estimation Strategy** is the process to estimate the architecture performance on test data



Hyperparameter Optimization (HPO) is the process of finding the best set of hyperparameters for a given model.

Bayesian Optimization (BO) is a **sample efficient method** for HPO that is based on two main components:

- a **probabilistic surrogate model** to approximate the objective function (usually a Gaussian Process or a Tree-based model)
- an **acquisition function** to deal with the trade-off between exploration and exploitation (e.g., Expected Improvement, Confidence Bounds, etc.)

The **Combined Algorithm Selection and Hyperparameter** (CASH) approach consists of a sequential learning process that first selects the most promising algorithms and then optimizes for their optimal hyperparameter configurations.

In the context of DL models, a CASH approach:

- uses NAS to search over a variety of architectures (the search process is usually performed via BO with Random Forest as surrogate model, since the search space is conditional and high-dimensional)
- adopts BO to optimize the hyperparameters of the models

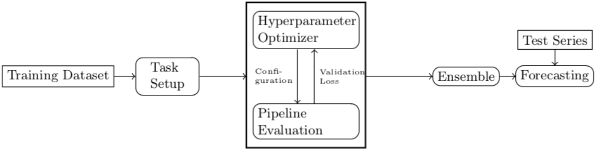
3. AutoDL for TSF

The application of NAS to the TSF problem is a relatively new research field.

The Forecasting Pipeline

The proposed framework is based on a **forecasting pipeline** that:

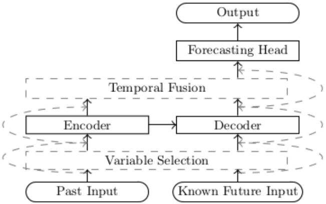
- first, it automatically **prepares the data** and splits each sequence into training, validation, and test sets
- then, the optimizer **searches for desirable architectures and hyperparameters** from the search space
- finally, a **weighted ensemble** of the top k selected configurations is used to evaluate the final predictions on the test set



The Search Space

SOTA architectures for TSF can be decomposed into three parts:

The **encoder** processes the input sequence (past target values) and embeds them into a latent space.
The **decoder** processes the latent embedding and future features.
The **forecasting head** is the output layer and takes the decoder's output to generate a sequence of scalar values.



The Search Space

Based on encoder-decoder architectures, the final search space contains many TSF algorithms, such as NBEATS, DeepAR, TFT.

Encoder		Decoder	auto-regressive	Architecture Class
Flat Encoder	MLP	MLP	No	Feed Forward Network
	N-BEATS	N-BEATS	No	N-BEATS [46]
Seq. Encoder	RNN/Transformer	RNN/Transformer	Yes	Seq2Seq [9]
			No	TFT [38]
		MLP	Yes	DeepAR [50]
			No	MQ-RNN [57]
	TCN	MLP	Yes	DeepAR [50]/WaveNet [45]
			No	MQ-CNN [57]

Network encoder may be sequential or flat.

Forecasting architecture may be autoregressive or not.

Hyperparameter Optimization

The hyperparameters of the models are optimized via a **CASH approach**.

The loss function (usually MAE or MASE) is optimized on the validation set via **BO**, using **Random Forest** as surrogate model.

Multi-fidelity optimization is used to reduce the computational cost of the

4. Conclusions

- The authors proposed a framework for AutoDL in the context of TSF to efficiently search for the best model architectures and hyperparameters
- The search space is based on the most recent SOTA TSF algorithms formulated as an encoder-decoder architecture
- Empirical results show that the proposed framework is able to outperform the SOTA methods on a variety of datasets
- The optimal choice of budget type is dataset specific and the most important hyperparameters are related to the optimizer of the neural network and its learning rate

Bibliografy

Deng D., Florian K., Hutter F., Bischl B. & Lindauer M., 2022, 'Efficient Automated Deep Learning for Time Series Forecasting', arXiv.

Elsken T., Metzen J. H. & Hutter F., 2019, 'Neural Architecture Search: A Survey', Journal of Machine Learning Research, 20 1-21

He X., Zhao K. & Chu X., 2021, 'AuoML: A Survey of the State-of-the-Art', Knowledge-Based Systems, 212

Meisenbacher S., et al., 2022, 'Review of Automated Time Series Forecasting Pipelines', WIREs Data Mining and Knowledge Discovery

Thank you!