DATA SCIENCE FOR EMBEDDED MACHINE LEARNING

VITESCO TECHNOLOGIES & MY ROLE

Subsidiary of Continental, made autonomous in 2019, Vitesco is a company offering technological solutions for automobiles.

i joined the company's data & analytics department in september 2023 as an apprentice data scientist.

My role is to use artificial intelligence to find solutions for temperature prediction in combustion and electric engines.

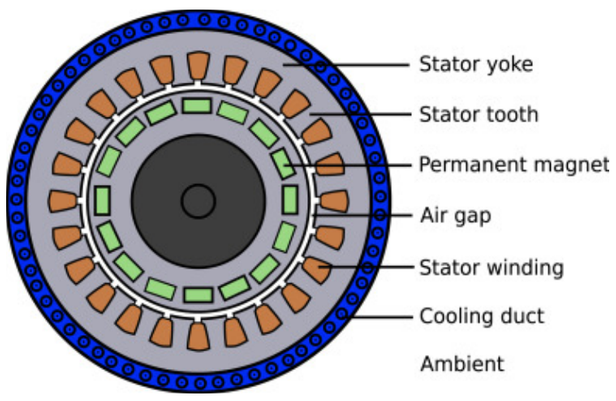
CONTEXT

With the transition to electricity in the automotive sector, the need for temperature prediction in engines is becoming increasingly important, both in terms of user comfort and safety.

The problem: some areas of the engine are inaccessible to temperature sensors, and these can represent a real cost.

Example: permanent magnets are not accessible. However, knowing their temperature is essential for motor life.

Consequences: magnetic instability, demagnetization, etc.

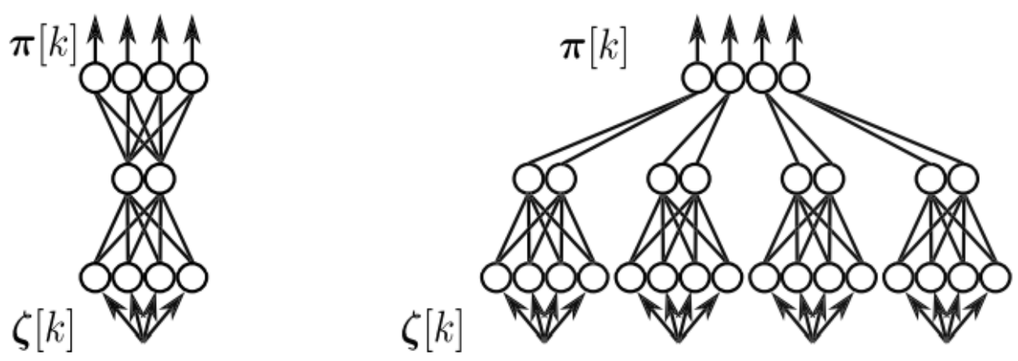


THE IDEA & THE PURPOSE

A new type of neural network came to light in 2019: thermal neural networks (tnn). inspired by lptn's (lumped parameters model) and ude's (universal differential equations), it has demonstrated impressive results, particularly for temperature forecasting.

How to process ? :

- Start From the general structure of LPTN's (system of equations)
- Note that the latter can be discretized with a first-order euler method
- Replace some of its terms by classical neural networks π and ζ (ann) :



The aim is therefore to parameterize this TNN as well as possible, so as to replace motor sensors with neural networks like this one.

HOW TO SET UP THE TNN ?

The python code for this neural network was supplied with the article by the german university behind the discovery.

the tnn adapts very well to the university database, but the aim is to adapt it to corporate databases.

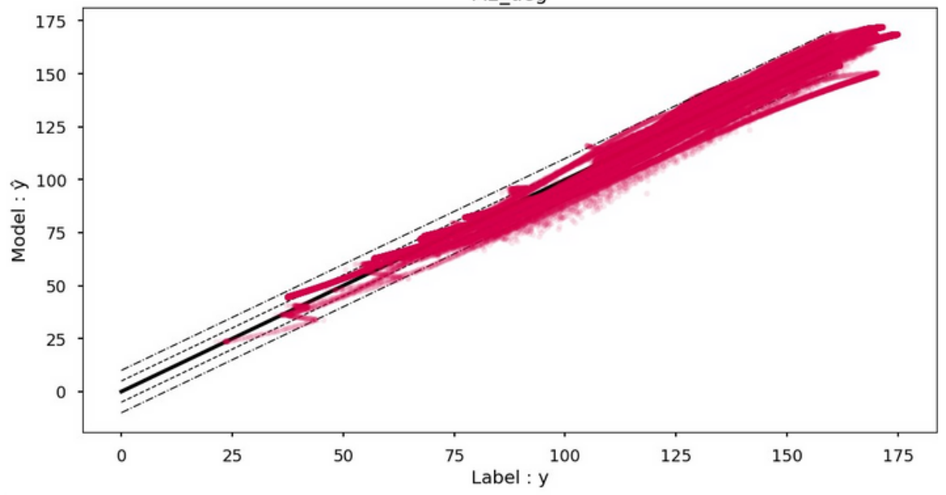
For example, I retrieved the database for the EMR4 electric motor my team is working on, and then set up the tnn as best I could. the aim of this database is to predict rotor temperature:

- Modification of input variables, creation of new relevant variables
- Modifications to π and ζ ANN parameters: activation functions, optimizer, number and complexity of layers, etc.
- Addition of relevant training performance visualization techniques: histograms, target-based prediction, training curve display

All these modifications have enabled me to achieve better results than those provided by default, but there's still a long way to go before I can embed it in a vehicle.

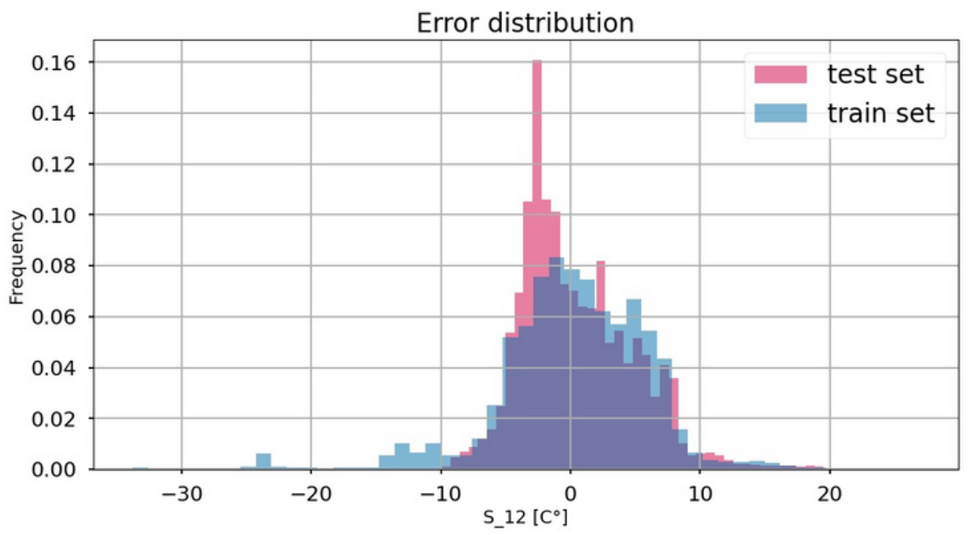
RESULTS

My tnn settings gave me the following result on the test set



This graph shows the temperature predicted by the tnn as a function of the target temperature. the closer the data (in red) are to the bisector, the more successful the prediction.

Here are the distribution errors for the training and test sets:



On the test set, the error is in the range [-10,20], which can be still be greatly improved.

WHAT NEXT ?

Continue to parameterize the tnn, in order to find the setting that minimizes the error, and then plan to embed it.